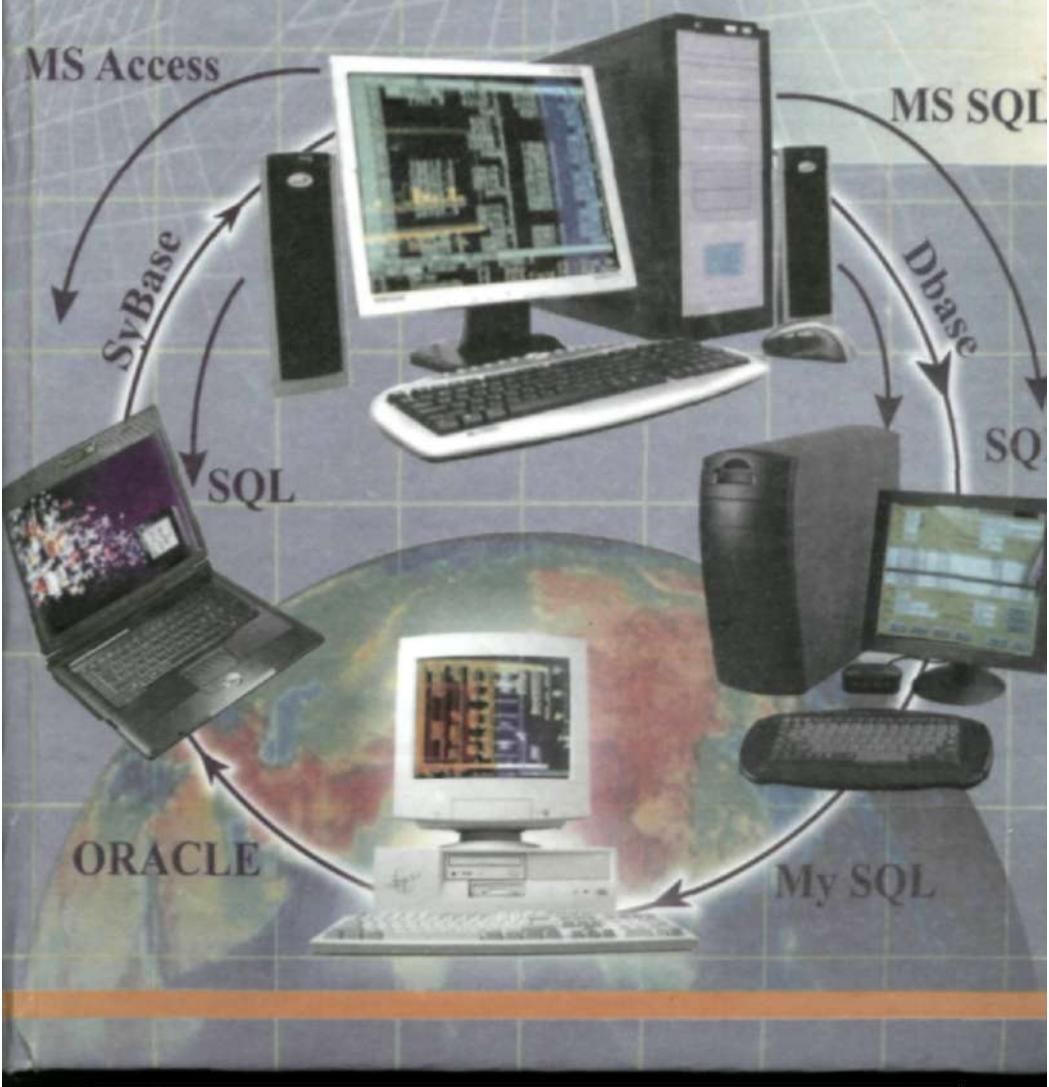


M.X. HAKIMOV, S.M. GAYNAZAROV

# BERILGANLAR BAZASINI BOSHQARISH TIZIMLARI



**O'ZBEKISTON RESPUBLIKASI OLIY VA O'RTA MAXSUS  
TA'LIM VAZIRLIGI**

**M.X. HAKIMOV, S.M. GAYNAZAROV**

# **BERILGANLAR BAZASINI BOSHQARISH TIZIMLARI**

*O'zbekiston Respublikasi Oliy va o'rta maxsus ta'lir vazirligi ismnomidagi  
5130200 - Amaliy matematika va informatika,  
5A130202 - Amaliy matematika va axborot texnologiyalari  
talabalari uchun darslik sifatida tavsiya etilgan*

**TOSHKENT - 2013**

UO'K: 51:007 (075)  
KBK 32.973.202ya73  
H-20

- H-20 M.X. Hakimov, S.M. Gaynazarov Berilganlar bazasini boshqarish tizimlari. Oliy o'quv yurtlari uchun darslik. – T.: «Fan va texnologiya», 2013, 648 bet.

**ISBN 978-9943-4263-8-2**

Darslikda berilganlar bazasi, berilganlar modellari, relatsion algebra, relatsion hisoblash, munosabatlarni normallash, berilganlarning relatsion butunligi, relatsion berilganlar bazasini loyihalash uslubiyati, berilganlarni modellashtiruvchi zamonaviy tizimlar, relatsion berilganlar bazasini yaratish, QBE va SQL tillari, berilganlar bazasini boshqarishda qo'llanilayotgan zamonaviy tizimlarning nazariy va amaliy asoslari keltirilgan. Relatsion modellarning turlarini tashkil etuvchi «mohiyat-alloqa» (ER) va semantik obyekt modellar o'zaro taqqoslangan. Relatsion berilganlar bazasining boshqarish tizimlari MS Access va MS SQL Server ning amaliy asoslari «Talabalar o'quv loyihasi» misolida «Mijoz-server» aloqasi orqali ham tushuntirilgan va obyektlar bilan ishlash ko'rsatilgan.

Oliy o'quv yurtlарининг 5130200 - «Amaliy matematika va informatika» yo'naliшining bakalavriat hamda 5A130202 - «Amaliy matematika va axborot texnologiyalari» mutaxassisligining magistrantlari uchun. Kompyuter texnologiyalari va informatika sohasidagi barcha yo'naliш va mutaxassislik talabalari ham foydalanishlari mumkin.

UO'K: 51:007 (075)  
KBK 32.973.202ya73

*Mas'ut muharrir:*

**A.A. Xoljigitov** – TATU Samarqand filiali direktori f.-m.f.d., professor

*Taqrizchilar:*

**SH.A. Nazirov** – Toshkent Axborot texnologiyalari universiteti

«Informatika» kafedrasining mudiri f.-m.f.d., professor;

**R.D. Aleyev** – Mirzo Ulug'bek nomidagi O'zbekiston Milliy universiteti «Hisoblash texnologiyalari va matematik modellashtirish» kafedrasining mudiri f.-m.f.d., professor.

**ISBN 978-9943-4263-8-2**

© «Fan va texnologiya» nashriyoti, 2013.

---

## MUQADDIMA

EHM tuzilishining takomillashib borishi bilan parallel ravishda berilganlar taqdimoti uchun axborot tuzilishi ham parallel ravishda rivojlanib bordi. Berilganlarni tasvirlash uchun: massiv, ro'yxatli tuzilishlar, iyerarxik tuzilishlar, abstrakt tuzilishlar va fayllar kabi tavsiflash usullari yaratildi. Lekin berilganlarni bu tavsiflash usullari u bilan ishlovchi programmaga bog'liq bo'ladi. Hattoki fayllar tizimi ham, foydalanuvchilar uchun alohida zaruriy amallarni bajargan xolos, har bir programma o'zining alohida berilganlarini aniqlab, ana shu berilganlarnigina boshqargan. Fayl tizimi qo'lda bajariladigan hisob-kitoblariga nisbatan ancha ilg'or texnologiya bo'lgan bo'lsa ham, undan foydalanishda ancha muammolar yuzaga kelgan. Bu muammolarning eng assosiysi berilganlarning otiqchaligi va ularning programmalarga bog'liqligidan iborat bo'lgan.

Ana shu muammoni hal qilish maqsadida 1960- yillarda berilganlarning iyerarxik tuzilishi yaratildi. Ko'pgina izlanishlar natijasida bu tuzilish, o'z talablariga ega bo'lgan aniq izchil modelga aylandi. 1968-yilda esa iyerarxik model asosida ishlovchi birinchi IMS (Information Management System) tizimi yaratildi. Deyarli parallel ravishda shu davrida tarmoq modeli yaratildi. Berilganlar bazasini Jossqarish tizimlar (BBRT) sohasidagi jiddiyesti tafqiqatishilardan chiri. Chetda, Berilganlar, xelqarligida, tizimiga modeli asosida ishlovchi IDS (Integrated Data Store) tizimi yaratildi.

Lekin, E.F. Kodd tononidan yaratilgan (1970 y.) relatsion model o'zining soddaligi, matematik nuqtayi nazardan jiddiyligi va amaliyotda foydaliligi bilan tezda olim va mutaxassislarning e'tiborini qaratdi. 1979-yilda E.F. Kodd kengaytirilgan relatsion modelni taklif etdi. Bu modelda berilganlar bazasining axborot va berilganlarga ishlov berish qismlari yetarli darajada aniqlashtirildi. Berilganlarga ishlov berish masalasining niyoyatda jiddiyligini hisobga olgan holda, model muallifi «berilganlarning tuzilishi, ularga ishlov berish usullari va qoidalarisiz, fiziologiyasiz anatomiyaga o'xshaydi», degan.

Oxirgi 40 yilda relatsion berilganlar bazasi (BB) tarkiblari idagi turli turdag'i berilganlarning tuzilishi va ular orasidagi bog'lanishlar hamda BBB tizimlarining nazariyasida bir qator juda muhim natijalar olindi. Olingan natjalarni informatikaning bugungi darjasida munosib hissasi bor, deb hisoblash mumkin. BB axborot tizimlarining yadrosiga aylandi va tash-

kilotlarning axborotga bo'lgan qarashlarini va ish usullarini tubdan o'zgartirdi. Xususan, BB texnologiyalarining rivojlanishi foydalanish uchun sodda, lekin juda keng imkoniyatlari bo'lgan boshqarish tizimlarining yaratilishiga olib keldi. Shu sababli, foydalanuvchilarning juda katta qismiga berilganlar bazasi bilan ishlashga imkoniyat yaratildi. Lekin, berilganlar bazasini yaratish sodda tuyulganligi natijasida, samarali ishlovchi tizimlarni loyihalash usullari xususida yetarli bilimga ega bo'lmagan holda, foydalanuvchilarning o'zları mustaqil ravishda BB va ilovalarni yaratma boshladilar. Bu esa, albatta juda ko'p holatlarda tizimlarning samarasizligiga, kasbiy qiyinchiliklarga olib keldi. Natijada, oxirgi foydalanuvchilarning qoniqmasligi programma ta'minotni rivojlantirish masalalarida ayrim to'siqlarni yuzaga keltirdi.

Shu nuqtayi nazardan, talabalar bilan uzoq yillar ishslash davomida berilganlar bazasini loyihalash masalasiga alohida va chuqur e'tibor qaratish zarurligiga ikki tomonlama amin bo'ldik. Birinchisi, talabalar orasida ham yuqorida keltirilgan vaziyatlar sababchi bo'lsa, ikkinchisi zamonaviy berilganlar bazasini modellashtiruvchi tizimlarning tobora ortib borayotgani, ya'ni loyihalash bosqichlarining nazariy jihatdan tan olinib amaliyatga tatbiq etilgani bo'ldi. Loyihalash konseptual, mantiqiy va fizik bosqichlardan iborat. Birinchi bosqichda, hech qanday fizik tavsiflarga bog'liq bo'lmagan, konseptual model yaratiladi. Ikkinci mantiqiy bosqichda, relatsion tizimda amalga oshirib bo'lmaydigan elementlarni olib tashlagan holda, konseptual modelga tuzatishlar kiritiladi. Uchinchi bosqichda mantiqiy model aniq maqsadli BBBT muhitini hisobga olgan holda, fizik modelga aksantiriladi.

Mualliflar – professorlar S.H.A. Nazirov va R.D. Aloyevlarga ushbu darslikni mazmunli muhokama va taqriz qilganliklari, A.A. Xolijigitovga mas'ul muharrir sifatida maskahatlari uchun chuqur minnatdorchilik izhor qiladilar.

Darslikdan 5330100 – Axborot tizimlarining matematik va ta'minoti, 5330200 – Informatika va axborot texnologiyalari (tarmoqlar bo'yicha), 5A330201 – Kompyuter tizimlari va ularning dasturiy ta'minoti (tarmoqlar bo'yicha), 5A330202 – Axborot va multimedya texnologiyalari (qo'llanish sohasi bo'yicha), 5A330203 – Amaliy informatika, 5A330204 - Axborot tizimlari (tarmoqlar bo'yicha), 5330300 – Axborot xavfsizligi, 5110700 – Informatika o'qitish metodikasi, 5111000 – Kasb ta'limi (yo'nalishlar bo'yicha), 5220200 – Axborotlashtirish va kutubxonashunoslik (sohalar bo'yicha), 5A220201 – Axborotlashtirish va kutubxonashunoslik (sohalar bo'yicha), 5A220202 – Axborot xizmatlari texnologiyalari (faoliyat turlari bo'yicha) talabalari, tadqiqotchi-izlanuvchilar ham foydalanishlari mumkin.

---

## KIRISH

Ushbu fanning maqsadi berilganlar bazasi boshqarish tizimlari asosida faoliyat yurituvchi berilganlar bazasi va u bilan ishlovchi muhitni yaratishning nazariy va eng zamонавиј amaliy texnologiyalarini o'rganishdan iborat.

Bu maqsad esa berilganlar bazasining nazariy va amaliy asoslari oldiga ilmiy izlanishlarni o'tkazish; haqiqiy obyektlar ustida ma'lumotlarni to'plash va berilganlarni modellashtrish; axborot tizimlari tarkibida faoliyat yurituvchi va unga xizmat ko'rsatuvchi berilganlar bazalarini yaratish, ularga xizmat ko'rsatish va boshqaruв jarayonidagi zatur qarorlarni qabul qilish imkoniyatini yaratish; turli hajmlardagi mustaqil faoliyat yurituvchi berilgan bazalarini yaratish kabi asosiy masalalarni yechishni qo'yadi.

Ushbu vazifalarni yechish bir tomonidan fanning turli sohalari, boshqarish, xalq xo'jaligi va boshqa sohalardagi yo'nalishlar bilan uzviy ravishda bog'lanishni taqozo etsa, ikkinchi tomonidan mustaqil ravishda eng zamонавиј yangi axborot texnologiyalarini o'rganish va joriy qilishni talab etadi.

Shu nuqtayi nazardan zamонавиј axborot texnologiyalarini o'rganish, ularni qo'llash, ulardan foydalangan holda zamонавиј berilgan bazalarini yaratish, boshqarish, ishlov berish va izlash masalalarini o'rganish, informatika va uning yo'nalishlari bo'yicha yuqori malakali kadrlarni yetkazish bu fan oldiga qo'yilgan vazifadir.

Nima uchun ma'lumotlar yoki axborotlar bazasi emas, berilganlar bazasi? Berilganlarning o'zi nima? Nima uchun aynan berilganlar? Baza nima? Nima uchun berilganlarni boshqarish kerak? Boshqarish tizimi berilganlar bazasini qanday boshqaradi?

Ana shu savollarga javob bersak, ushbu darslikning tub mohiyatini ochib bergen va uning «Axborot asri»dagi boshqaruв tizimlarida tutgan o'rni naqadar muhimligini anglagan bo'lamiz.

Axborotlar bazasi iborasi kamroq uchrasa-da, lekin keyingi davrda ma'lumotlar bazasi iborasi juda ko'p ishlatalmoqda. Hattoki, 40 yildan ortiq «Berilganlar bazasining boshqarish tizimi» deb atalib kelayotgan fanning nomini «Ma'lumotlar bazasini boshqarish tizimi», «Ma'lumotlar banki» va boshqa nomlarga o'zgartirib yozilmoqda. Shuning uchun quyida biz ushbu atamalarning farqini aniqlab beramiz.

*Ma'lumot bu – kimdir tomonidan e'lon qilingan yoki tadqiqot, tahlil yo ta'lim olish natijasida olingan ilm; berilganlardan kelib chigadigan ilm* [28]. Masalan, shifokorning qabulxonasiagi ushbu yozuvga e'tibor beraylik:

«Shifokor Xoliqov I.X.

Qabul kunlari:

dushanba	9 <sup>00</sup> – 12 <sup>00</sup>	seshanba	12 <sup>00</sup> – 15 <sup>00</sup>
chorshanba	9 <sup>00</sup> – 12 <sup>00</sup>	payshanba	12 <sup>00</sup> – 15 <sup>00</sup>
juma	9 <sup>00</sup> – 12 <sup>00</sup>	shanba	12 <sup>00</sup> – 15 <sup>00</sup>

Bu yerdagi har bir so'z va soat vaqtleri (shunday qabul qilamiz) bular berilganlar bo'lib, biz olgan ilm esa, bu *ma'lumotdir*. Ya'ni *ma'lumotni tashkil etuvchilari berilganlardan iborat*. Aslida, *berilganlar* atamasi lotin tilidan (*data*) fransuz (*date*) so'ngra, ingliz (*data*) tillariga o'tgan bo'lib, *data* – qaysidir voqeanning taqvimi yoki nimanidir (xat, hujjat va h.k.) yozilgan sanasini bildiradi [96]. *Berilganlar*, bu – o'mratilgan fakt, mantiqiy xulosa yoki da'ilni asoslovchi bo'lib, ixtiyoriy intellektual tizimni qurishning negizi hisoblanadi.

*Baza* atamasi grek tilidan (*basis*) fransuz tiliga [96], so'ngra ingliz, rus va boshqa tillarga, jumladan o'zbek tiliga kirib kelgan. Uning 7 xil *ma'nosi* bo'lib, asosiy *ma'nosi* «*nimanidir asosi*» degan mazmunni ifodalaydi.

Demak, *berilganlar bazasi* ifodasining tub *ma'nosi* – berilganlar asosi, degan mazmunni beradi. Odatta berilganlar va ularning mazmuni birgalikda qayd etiladi, chunki tabiiy tilning ularni birgalikda taqqidot qilish uchun imkoniyatlari katta. Alovida vaziyatlarda berilganlar va ularning mazmuni va berilganlar ajralgan holatda ko'riliishi mumkin, lekin bu berilganlar bilan ishlash qiyinchiliklar tug'diradi.

Ushbu darslikning asosini mualliflar tomonidan 1980–82-yillardan boshlab uzoq yillar davomida o'qitilayotgan (ToshDU-O'zMU) va doimiy ravishda to'ldirib va o'zgartirib borilgan «Berilganlar bazasini boshqarish tizimlari» fani bo'yicha o'tilgan *ma'ruzalarning asosi* yotadi. Berilgan bazasining ayrim modellari hozirgi davrda juda kam ishlatalsa ham, evolutions nuqtayi nazardan vogelikni to'g'ri talqin qilish, zamonaviy axborot tizimlaridagi berilganlar bazasining ahamiyatini tushunishda ular xususida tushunchalar bo'lishi zarur deb hisoblaymiz.

Darslikda asosiy urg'u relatcion berilganlar bazasi nazariyasi, ularni boshqarishda qo'llanilayotgan zamonaviy tizimlar va SQL tiliga berilgan. Darslikni yozishdan asosiy maqsadimiz, talabalarga relatcion berilganlar bazasining nazariyasini o'rgatish bilan birga, berilganlar bazasini tizimlarini yaratishdagi asosiy tomoniyatlarni ham o'rgatish, turli predmet sohalari uchun berilganlar bazalarining tizimlarini yaratuvchi va kafolatli faoliyat yuritishi ni ta'minlovchi mutaxassislami yetkazish, deb bildik. Uzoq yillik amaliy

tajribamiz ham, berilganlar bazasi bo'yicha mutaxassislarning ko'pgina boshqa predmet sohalardan xabardor bo'lishi kerakligini tasdiqlaydi, chunki shu sohadagi zamonaviy mutaxassis bo'lish uchun nazariy tomonidan puxta bilinga ega bo'lgan taqdirdagina boshqa predmet sohani tahlil qilish darajasiga yetish mumkin, amaliy jihatdan esa doimiy ravishda zamonaviy BBB tizimlaridagi o'zgarishlardan xabardor bo'lish talab qiladi.

Ushbu darslik yettita bo'limdan iborat bo'lib, har bir bo'limga maqsadli aniq funksional vazifa yuklatilgan.

I bo'limda boshlang'ich asosiy tushunchalar, berilganlar bazasigacha bo'lgan fayllar tizimi, fayllar tizimidan «Uch bosqichli» ANSI/SPARC arxitekturasiga o'tish, bu arxitekturaning tashkil etuvchi darajalari, axborotning uchta sohasi, kalitlar, so'rovlar, berilganlar modeli va ularning turlari, chizmalar, berilganlar orasidagi aloqalarning turlari hamda yozuvlar orasidagi bog'lanishlar yoritildi.

II bo'limda berilganlarning iyerarxik, tarmoq va relatsion turdag'i – «mohiyat-aloha (ER)» hamda semantik obyekt modellar keltirildi. Har bir modelni qanday yaratish izohlanadi; bugungi holati nazariy jihatdan to'liq yoritilib oqibati tahlil qilindi. Ushbu bo'limda asosiy e'tibor relatsion modellarga qaratildi. Relatsion modellarning turlarini tashkil etuvchi «mohiyat-aloha» va semantik obyekt modellar o'zaro taqqoslandi. Talabalar o'quv loyihasi ER-diagramma tarzida modellashtirildi. Zamonaviy axborot tizimlarining eng asosiy ta'minotchisi berilganlar bazasi bo'lgani, hamda «mohiyat-aloha» modellari (ER-diagrammalar) axborotlarni modellashtirishdagi tizimlarda ishlataluvchi UML tilida ham ishlatilgani sababli, axborot tizimlarini modellashtirish masalalari yo'nalish sifatida keltirildi. Zamonaviy axborot tizimlarini modellashtiruvchi CASE muhitlar va ularning tarkibiy qismlari keltirildi.

III bo'lim relatsion algebra, munosabatlarni normallashtirish, relatsion hisoblash va berilganlarning relatsion butunligi boblaridan iborat. Munosabatlari ustida amallarni bajarish nazariy-to'plam va maxsus relatsion amallar turli misollar orqali tushuntirildi. Munosabatlarni normallashtirish rasmiy usul bo'lib, uning oltita normal shakli mavjud. Munosabatlarni normallashtirishning hamma bosqichlari to'liq keltirildi. Relatsion hisoblash bobida kortej va domenlarni hisoblashning nazariy va amaliy asoslari yuritildi. Berilganlarning relatsion butunligi ta'minlandi va u berilganlar bazasining asosiy tamoyillaridan biri sifatida asoslandi.

IV bo'lim beshta bobdan iborat bo'lib, berilganlar bazasini yaratishdagi eng nozik masalalar, ya'ni loyihalash uslubiyatidagi loyihalash turlari hamda tranzaksiyalarni loyihalashning nazariyasi yoritilgan. Axborot tizimlarini loyihalash masalalari berilganlar bazasini loyihalashdagi konseptual, mantiqiy va fizik bosqichlar bilan umumiy kesishish nuqtalariga ega. Bu bosqich-

larning har biri «Talabalar o'quv loyihasi» misolida chuqur va atroficha yoritildi. Tranzaksiyalarga qo'yiladigan asosiy talablar, ularning bajarilish modellari, parallel bajarishdagi muammolar, tranzaksiya orasidagi ziddiyatlar keltirildi.

V bo'limda berilganlarni modellashtirishdagi zamonaviy tizimlar, ular-dagi usul va uslubiyatlardan keltirildi. Berilganlarning modelini yaratishda ishlataladigan eng zamonaviy tizimlardagi Barker usuli, IDEFIX va DATARUN uslubiyatlari, CASE vositalarning tavsifi va sinflari yoritildi. To'rtinchchi bo'limda o'tilgan nazariy bilimlarni amaliy jihatdan o'zlashtirish maqsadida Toad Data Modeler muhitida berilganlarni modellashtirish «Talabalar o'quv loyihasi» misolida keltirildi. BBBT MS Access va MS SQL Server uchun DDL Script lar yaratildi, ulardan foydalanish usullari ko'rsatildi.

VI bo'lim relatcion BBB tizimlari, QBE va SQL tillari hamda taqsimlangan berilganlar bazalariga bag'ishlandi. BBB tizimlarining arxitekturasidagi tarkibiy qismilar, ularning funksiyasi, boshqarish tizimlarining sinflari yoritildi. 22- bobda QBE tili, munosabatlar ustidagi turli so'rovlarini yaratishda QBE vositalaridan foydalanish usullari QBE to'ri orqali ko'pgina misollarda ko'rsatildi. 23-30 boblarda SQL tili, uning tarkibiy qismilari bo'lmish DDL va DML tavsiflandi, sintaktik tuzilishlari keltirildi va ko'pgina misollar yordamida munosabatlar bilan ishlash yo'riqnomalari ko'rsatildi. Taqsimlangan berilganlar bazalariga ishlov berishdagi turli arxitekturalar keltirildi, so'rovlarini optimallashtirish va qayta ishslash masalalari ko'rsatildi.

So'nggi VII bo'lim amaliyotga bag'ishlandi. Relatcion BBBT MS Access va MS SQL Server ning amaliy asoslari Talabalar o'quv loyihasi misolida keltirildi. Ikkali tizimning imkoniyatlari to'liq tavsiflandi, jadvallarni yaratish va ular bilan ishlash, shakllar va hisobotlar yaratish, turli oddiy va murakkab so'rovlar yaratish, MS Visual Basic protsedurasini yaratish, MS Access loyihasi bilan ishlash ko'rsatildi. «Mijoz-server» aloqasi yaratildi va bazalar bilan ishlash ko'rsatildi.

Har bir bob uchun qisqa xulosalar chiqarilgan hamda nazariy boblardan so'ng nazorat uchun savollar, amaliy boblardan so'ng esa nazorat uchun savollar va amaliyot uchun topshiriqlar keltirilgan. Kitob oxirida shu fan bo'yicha o'zbekcha-englizcha-ruscha atamalarning o'zbek tilidagi talqini berilgan.

Amaliyot uchun keltirilgan Toad Data Modeler muhiti, BBBT MS Access va MS SQL Server darhol o'zlashtirish qiyin bo'lsa ham, talabalar ma'lum ko'nikma, tushuncha va topshiriqlarni bajarish natijasida zaruriy amaliy bilimlarga ega bo'ladilar, degan umidda keltirildi.

Nazariy, amaliy va mustaqil ta'lim soatlarini shakllantirishda uslubiy ko'rsatmalardan foydalanish tavsiya etiladi.

Mualliflar ushbu darslik ilk bor o'zbek tilida yaratilishi, uning ahamiyati va mas'uliyatini chuqur anglagan holda, uning ustida bajariладиган ijodiy ish jarayonidagi vazifalarni aniq kelishib oлган holda olib bordilar va buni e'tirof etishni o'z vazifalari, deb biladilar. 23-30 boblar to'liq S.M. Gaynazarov tomonidan yozildi, kirish, uslubiy ko'rsatmalar va 10- bob hamkorlikda, qolgan boblar hamda muqaddima va glossariy to'liq M.X. Hakimov tomonidan yozildi.

Mualliflar 20- bob bo'yicha g'oyani amalgga oshirishdagи yordami uchun magistr L.X. Xudoyberdiyevga, kitob muqovasining dizayni bo'yicha g'oyani amalgga oshirgani uchun bakalavr A.R. Shoxnazarovga minnatdorchilik izhor qiladilar.

**Mualliflar**

## JSLUBIY KO'RSATMALAR

Bakalavriat yo'nalishlari uchun ma'ruza, amaliy mashg'ulotlarni va mustaqil ta'lif soatlarini quyidagi ikkita variant asosida shakllantirish tavsiya etiladi:

### 1-variant.

- *nazariy mashg'ulotlar uchun ma'ruza mavzulari*: 1 ÷ 7- boblar, 10, bob, 11- bobdan uchinchi normal shaklgacha bo'lgan qismlar, 12, 13, 18, 19, 20 ÷ 28- boblar;
  - *amaliy mashg'ulotlar*: 20, 22 ÷ 30- boblar, 32 ÷ 38- boblar;
  - *mustaqil ta'lif*: 8, 14 ÷ 17- hamda 19, 20-, 31- boblar asosida shakllantirish tavsiya etiladi.

### 2-variant.

- *nazariy mashg'ulotlar uchun ma'ruza mavzulari*: 1 ÷ 3- boblar, 6-, 7, boblar, 11- bobdan uchinchi normal shaklgacha bo'lgan qismlar, 12, 13, 18, 19, 20 ÷ 29 va 31- boblar;
  - *amaliy mashg'ulotlar*: 20, 22 ÷ 30- boblar, 32 ÷ 38- boblar;
  - *mustaqil ta'lif*: 4, 5, 8, 14 ÷ 17, 19, 20, 31- boblar asosida shakllantirish tavsiya etiladi

**Magistratura mutaxassisligi** uchun ma'ruza, amaliy mashg'ulotlarni va mustaqil ta'lif soatlarini quyidagicha shakllantirish tavsiya etiladi:

- *nazariy mashg'ulotlar uchun ma'ruza mavzulari*: 8, 9, 11- bobdan Boys-Kodd, to'rtinchchi va beshinchchi normal shakllar; IV va V bo'limlar, 22 ÷ 31- boblar;
- *amaliy mashg'ulotlar*: 20, 22 ÷ 38- boblar;
- *mustaqil ta'lif*: amaliy mashg'ulotlarning mavzulari, mustaqil ta'limda yaratiladigan tizim mavzulari bilan chambarchas bog'linishi va magistrant berilganlar bazasini Toad Data Modeler muhitida modelash-tirishi uchun berilganlar bazalarini loyihalash bo'yicha quyidagi namunaviy haqiqiy obyektlar bo'yicha mavzular tavsiya etiladi:

- magistratura;
- Respublika oliy ta'limi;
- Oliy ta'lif muassasasi;
- xodimlar bo'limi;
- tashkilotning moddiy boyliklari;
- respublika aholisi.

Mustaqil ta'limning maqsadi programm ta'minot bo'yicha axborotlarni boshqarish uchun zarur bo'lgan hamma tizim yaratish bosqichlarini ko'rib chiqishdan iborat bo'lib, jumladan u o'z ichiga berilganlar bazalarini loyi-halash, foydalanuvchi aloqasini ishlab chiqish, murakkab so'rovlarni yaratish va otladka qilishni oladi. Ushbu bosqichda har bir magistrant programmalashning obyektga yo'naltirilgan usullari va «mijoz-server» texnologiyalariga asosiy e'tiborni qaratadilar. O'quv jarayoni davomida har bir magistrant tanlagan mavzu bo'yicha modellashtirgan bazani Toad Data Modeler muhitidan MS Access va SQL Server ga o'tkazadi. Keyingi bosqichda «mijoz-server» texnologiyasi asosida berilganlar bazasini MS Access ga SQL Server ni bog'lagan holda, «mijoz» va/yoki «server»da yaratadi. Magistrant tanlagan mavzusi bo'yicha berilganlar bazasi tizimini yaratish jarayoni amaliy mashg'ulotlar jarayoni bilan uyg'unlashgan tarzda olib borishi tavsiya etiladi.

*Har bir magistrantga bitta mavzu biriktiriladi* va undan shu mavzu bo'yicha berilganlar bazasini yaratish talab qilinadi.

Bakalavriat talabasi ushbu fanni chuqur o'zlashtirishi va zamonaviy kadr bo'lib yetishishi uchun, atama va iboralarning ta'rifi, talqini va ma'nosini chuqur anglashi, turli predmet sohalar bo'yicha berilganlarni tahlil qilish ko'nikmasini egallashi, har bir bobdag'i nazariy savollarga javob bera olishi va amaliy mashg'ulotlar uchun keltirilgan mashqlarni mustaqil bajara olishi kerak.

Magistrantlar esa predmet sohani chuqur tahlil qila olishi va «mijoz-server» texnologiyalari asosida ishlovchi tizimlarni yaratish usullarining asosini berilgan jahasi tashkil etishini tushungar holda ushbu darslikda keltirilgan uslubiyat yondashuvlardan biri eganligini anglashlari kerak.

## I BO'LIM. BERILGANLAR BAZALARI

### 1 - bob. BERILGANLAR BAZASI

**Tayanch iboralar:** berilganlar bazasi, berilganlarning global mantiqiy taqdimoti, berilganlar modeli, konseptual model, konseptual chizma, berilganlarning tashqi tuzilishi, berilganlarning global mantiqiy tuzilishi, berilganlarning fizik tuzilishi, BB administratori, berilganlarning mantiqiy mustaqilligi, berilganlarning fizik mustaqilligi, «Uch bosqichli» arxitektura, tashqi daraja, ichki darajada, konseptual daraja, berilganlar bazasining chizmasi, tashqi chizma, ichki chizma.

#### Asosiy tushunchalar

*Berilganlar bazasi* (BB) atamasiga mutaxassislar tomonidan turlicha, lekin bir-biriga juda yaqin va chuqur ta'riflar berilgan. Shulardan biri, berilganlar bazasi – bu ma'lum predmet sohaga taalluqli bo'lgan, maxsus tuzilmaga ega berilganlar majmuasining nomi; berilganlar bazasini boshqarish tizimi (BBBT) boshqaruvida ishlaydigan, bir yoki bir qancha ilovalarni optimal tarzda ishlata digan, o'zaro bog'langan berilganlar majmuasi; berilganlarning tuzilishi uni ishlata digan programmalarga bog'liq bo'tmaydi; berilganlar shunday tuzilishga ega bo'ladiki, bunda ulami kengaytirish, o'zgartirish, olib tashlash, tanlash va boshqa imkoniyatlar bo'ladi [108]. Berilganlar bazalari tuzilish jihatidan mustaqil bo'lsa, ulami boshqaruv tizimi o'z tarkibida saqlaydi va boshqaradi.

Ta'rifdan kelib chiqqan holda, boshqaruv tizimi berilganlar bazalari bilan mustaqil ishlash va/yoki programma mahsulotlari (ilovalari) orqali ishlash imkoniyatini beradi. Berilganlar bazalaridagi har qanday o'zgartirishlar ilovalarga mutlaq ta'sir qilmaydi.

Berilganlar bazasining eng muhim jihatlaridan biri, uning doimiy ravishda o'zgarishi va ko'payishidan iborat. Xo'sh, shunga o'xshash imkoniyatlar fayllarda ham bor ediku, nega yangi maxsus yo'nalish yaratildi?

Buning sababi, fayllardagi har bir o'zgarishni amalga oshirish uchun programmani qaytadan o'zgartirish va otladka qilish kerak bo'ladi. Fayllarni kengaytirish, yana programmani o'zgartirish va otladka qilishga olib

kelar edi. Bu esa o'z navbatida berilganlarning ortiqchaligiga olib kelar edi, ya'ni bir xildagi berilganlar turli fayllarda shaklan o'zgargan yoki hattoki o'zgarmagan holda saqlangan. Vaholanki, o'tgan asrning 60- 70- yillarda berilganlarni saqlash, o'zgartirish va zaruriy qayta ishlov berish ehtiyoji kun sari o'sayotgan edi. Chunki, korxonalarни boshqarishda elektron hisoblash mashinalarining ahamiyati hamda berilganlar bilan ishlash beqiyos ravishda oshib bormoqda edi. Ana shu ehtiyojlarni qondirish uchun yangi texnologiya zarur edi. *BB ana shu talablarni qondirish uchun yangi texnologiya sifatida yaratildi.*

Bugungi kunda berilganlar bazasining tutgan o'mni nihoyatda beqiyos bo'lib, ilmiy-texnika masalalarini yechish, turli tizimlar, axborotlarni boshqarish, ularni qayta ishlash va boshqa berilganlar bilan ishlash tizimlarida, berilganlar bazasisiz ularning mustaqil faoliyat yuritishini tasavvur ham qilib bo'lmaydi.

Berilganlar bazasiga asos solgan va uning rivojlanishiga juda katta hissa qo'shgan olimlar R. Barker [4], R.F. Boys [5,49], J. Gray [7,23,27], M.M. Zluf [54-55], E.F. Kodd [10-19], R. Fagin [24-26], D.D. Chamberlin [5-8, 22], P. Chen [9], K.P. Yesvaran [22-23] va boshqalar bo'lsa, xalqaro tashkilot hamda kompaniyalardan ANSI [2], IBM [3], CODASYL [90] va h.k. bo'ldi. Berilganlar bazasining targ'iboti va ta'lim jarayonida qo'llanishiga o'zining yirik hissasini qo'shgan olimlardan birinchisi Dj. Martin [84] bo'lib, keyinchalik K. Deyt [66, 67], V.V. Kirillov, G.YU. Gromov [74], T. Konnolli, K. Begg [78], D. Kryonke [79], S.D. Kuznetsov [80-81], D. Meyer [85], M. Nagao, T. Katayama, S. Uemura [89], T.V. Olle [90], A.YU. Pushnikov [94], T. Tiori, Dj. Frey [98], D. Ulman [100], Djessri D. Ulman, Dj. Uidom [101], Dj. Xabbard [107], M.SH. Salenko [110], D. Sikritzis, F. Loxovski [111] o'zlarining munosib hissalarini qo'shdilar.

Berilganlarning mustaqilligi berilganlar bazasining asosiy xususiyatlaridan biri sifatida e'tirof etiladi. Buning mohiyati berilganlarning mustaqilligi va ulardan foydalanuvchi programma mahsulotlarning bir-biriga bog'liq emasligi, ya'ni birining o'zgarishi ikkinchisining o'zgarishiga olib kelmasligida. Yana bir jihatni, berilganlarni saqlovchi fizik qurilmalarning takomillashishi ham programma mahsulotlarining yoki berilganlarning o'zgarishiga ta'sir qilmasligi.

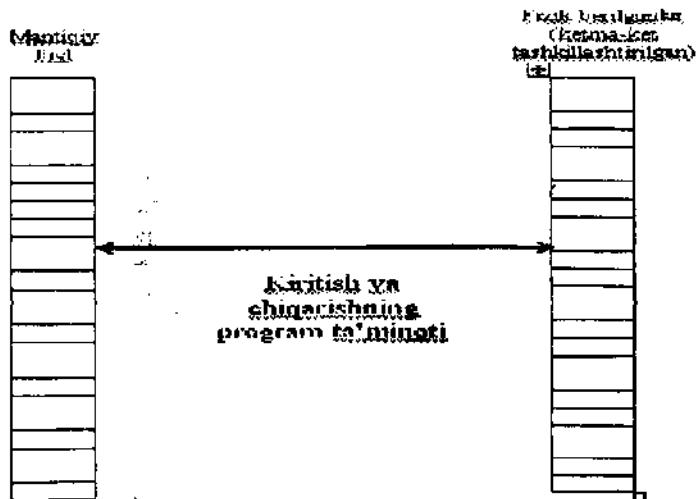
### **Berilganlar bazalari konsepsiyalarining evolutsiyasi**

BB tushunchasi o'tgan asrning 60- yillarning oxirida paydo bo'ldi. Bu davrgacha berilganlarni qayta ishlash sohasida berilganlar fayllari va berilganlar majmua (terma)lari ustida ishlar olib borilar edi. Boshqa sohalarda bo'lgani kabi, yangi tushuncha paydo bo'lishi bilan ko'pchilik program-

machilar o'z fayllarining faqat nomini o'zgartirib, xususiyatlarini o'zgartirmagan holda ishladilar. Vaholanki, berilganlar bazasining xususiyatlari – mustaqillik, ortiqchalikka yo'l qo'ymaslik, berilganlarning o'zaro bog'lig'i, ularning himoyasi va haqiqiy vaqt oralig'ida ularaga kirish imkoniyatlari bilan aloqador edi. Yangi texnologiyaning tobora kengayishi natijasida, sanab o'tilgan xususiyatlar yangi texnologiyaning asosi bo'lmish BB programma ta'minotlarida o'z aksini topa boshladи.

1.1 – 1.4-rasmarda berilganlarni saqlash usullaridagi rivojlanishning to'rtta bosqichi keltirilgan. Uchinchi avlod EHM paydo bo'lgan davrgacha (1965 y.) deyarli hamma fayllar 1.1-rasmida keltirilgan 1- bosqichdagi kabi amalga oshirilgan. Programma ta'moti asosan xotira qurilmalari uchun kiritish va chiqarish amallarini bajarishdan hamda unchalik ko'p bo'lmagan berilganlarga ishlov beruvchi yordamchi vositalardan iborat bo'lgan. Berilganlarning tuzilishi bilan amaliy programmalarini yozish paytida sodda usullar bilan, ya'ni magnit lentalarga fayllarni ketma-ket yozish yo'li bilan amalga oshirilgan. Berilganlar mustaqil bo'lmagan. Agarda berilganlarning tuzilishi yoki saqlovchi qurilma o'zgarsa, programmachi o'z programmasiga tegishli o'zgartirishlarni kiritishi, qayta komplilatsiya va ottadka qilishi zarur bo'lgan. Faylni yangilash uchun yangi faylni yozish va eskisini saqlashga majbur bo'lgan.

#### **1- BOSQICH: Berilganlarning oddiy fayllari (1960-yillardning boshi)**



1.1-rasm.

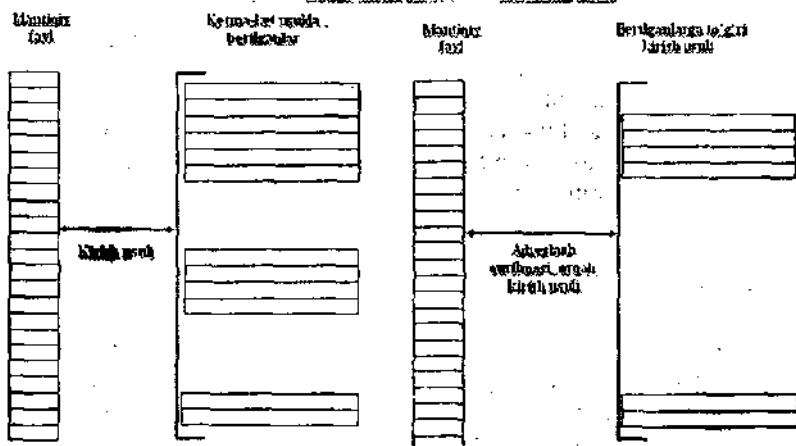
Ayrim holatlarda bir nechta ajded fayllarni saqlashga ham to'g'ri

kelgan. Fayllar faqatgina bitta masalani yechishga mo'ljallangan. Boshqa masalani o'sha berilganlar bilan faqat boshqa shaklda yechish uchun yangi fayllarni yaratishga majbur bo'lingan. Shuning uchun ham tizimdag'i fayllarda berilganlarning ortiqchalik darajasi niyoyatda yuqori bo'lgan, bir xildagi berilganlar bir nechta fayllarda saqlangan.

1- bosqichda fayliarga to'g'ri kirish huquqi, ya'nini «yozuvga» o'tish uchun ketma-ket n-1 ta yozuvni o'qimasdan, to'g'ridan-to'g'ri keraklisiga o'tish, mayjud bo'lmagan. Yozuvlarni adreslash imkoniyatidan programmani yozish jarayonida foydalilanigan. Agarda xotira qurilmasi o'zgarsa, programmaga katta o'zgartirishlarni kiritishga majbur bo'lingan. Amalda esa, albatta xotira qurilmalari o'zgarib turgan. Bu esa bitta bitni yozishga ketgan sarf-xarajatlarni yanada oshishiga olib kelgan.

2- bosqich (1.2-rasm) 1- bosqichga nisbatan fayllarning tuzilish va ularni saqlovchi qurilmalarning o'zgarishi bilan tavsiflanadi. Berilganlarga ketma-ket kirish imkoniyatidan tashqari, to'g'ridan-to'g'ri kirish huquqi yaratilgan. Berilganlarni paket, tezkor yoki haqiqiy vaqt oralig'ida ishllov berish mumkin bo'lgan. Fayllar orasidagi mantiqiy va fizik tuzilishlar turli-cha bo'lishiga qaramasdan, o'zaro aloqa sodda bo'lgan. Endi xotira qurilmalari o'zarganda ham, amaliy programmalar o'zgarmagan. Berilganlarning tuzilishi ketma-ket, indeksli ketma-ket yoki to'g'ri kirishli usullar bilan aniqlangan. Berilganlarni fayllardan ko'p kalitli qidirish usuli mavjud bo'lgan, lekin juda kam ishlatilgan.

2- BOZOICH: Fayza fayli usuli (1960-yillardan oxir)



1.2-rasm.

Berilganlami himoya qilishning sodda usullari ham paydo bo'lgan,

lekin ancha ishonchisiz bo'lgan. Faqat bitta ilova uchun berilganlarga ishlov berish va optimallashtirish usullari qo'llanilgan. Fayllarda berilganlarning ortiqchaligi hali ham ko'p bo'lgan. Berilganlarga ishlov berishning namunaviy programma ta'minoti tizimi berilganlarga kirish imkoniyatini bergan, lekin ularni boshqarish imkoniyatini hali bo'lmasagan.

XX asming 60-yillari oxiriga kelganda, tijorat sohasidagi berilganlarga ishlov berish ancha rivojlangan. Nafaqat EHM qurilmalarida fayllarni saqlash va kengayish usullarining o'zgarishidan, balki berilganlarga yangi yozuvlar va yangi o'zarlo aloqalar qo'shilganda ham, *amaliy programmalarini mustaqilligini* ta'minlash eng dolzarb masalaga aylangan.

Berilganlarni boshqarish uchun yaxshi programma ta'minoti bo'lganda, bitta masalani yechish uchun turlicha yaratilgan programmalardagi fayllarni ularni o'zgartirmagan holda yagona tarzda birlashtirish imkoniyati mavjud edi. O'tgan asrning 70-yillariga kelganda, berilganlar tizimlarini boshqaruvchi birinchi tizimlar paydo bo'la boshladi va fayllarni birlashtirish hamda programmalarning mustaqilliq masalasini yechishga harakat qilindi (1.3-rasm). BB doimiy rivojanuvchi obyekt bo'lgani uchun undan foydalanuvchi ilovalar soni tobora oshib borgan. BBga yangi yozuvlar, mavjudlariga yangi berilganlarning elementlari qo'shiladi. Yangi turdag'i so'rovlar paydo bo'lganda, tizimning samaradorligini oshirish uchun BB tuzilishi o'zgaradi. Foydalanuvchilar berilganlarga nisbatan so'rov talablarini va turlarini o'zgartiradilar.

BB tuzilishi, fayllarning tuzilishiga nisbatan tezroq o'zgaradi. Saqlanadigan berilganlar elementlari va ularni saqlash usullari doimiy ravishda o'zgarib turadi. Bir xildagi berilganlarda turlicha mantiqiy fayllarni hosil qilish mumkin, bitta to'plam berilganlarga kirish turli ilovalar orqali turli usulda bajarilishi mumkin (1.3-rasm). Bu esa berilganlarning murakkab tizimlarini yaratishga olib keladi. Lekin yaxshi yaratilgan programma ta'minoti, BB murakkab tuzilishi bilan ishlashda, programmachilarga qiyinchilik tug'dirmaydi. Berilganlar qanday tuzilishga ega bo'lmisin, programmachi o'zi yaratadigan ilovalardagi talablarga nisbatan fayllarni tuzilishini sodda ko'rinishda tasavvur qilishi kerak.

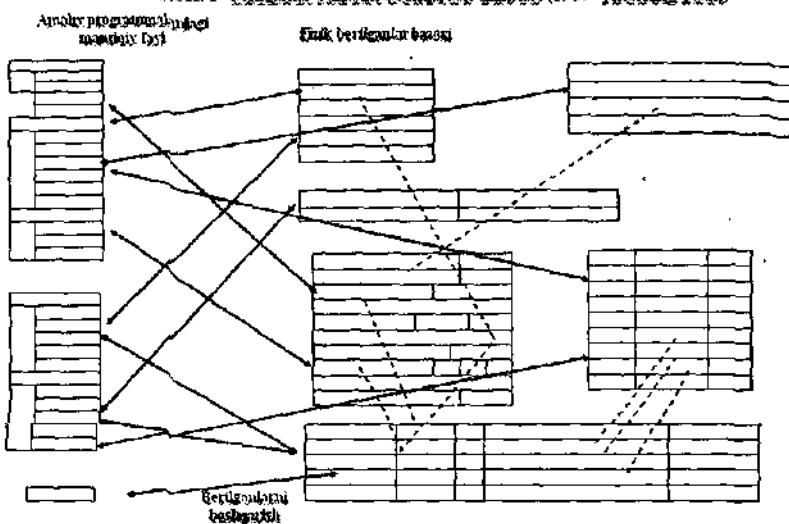
Berilganlarni saqlashning 3- bosqichida BB programma ta'minoti programmachi tasavvuridagi fayllarning tuzilishlarini (1.3-rasmdagi diagrammaning chap tomoni) berilganlarni tuzilishini saqlashning fizik qurilmalariiga akslantirish vositalariga va aksinchasiga ega bo'lgan (1.3-rasmdagi diagrammaning o'ng tomoni).

Programma ta'minoti berilganlar ortiqchaligini kamaytirish vositalari ga ega bo'ldi. Turli ilovalar uchun berilganlar elementlari umumiy tarzda tashkillashtirildi. Berilganlar ortiqchaligining yo'qligi ularning butunligini saqlashga imkon yaratdi. Ko'p kalitlar bo'yicha izlash imkoniyatlari paydo

bo'ldi. Berilganlarning murakkab tuzilishlaridan foydalanishda programmlarni murakkablashtirishga ehtiyoj juda kamaydi.

4- bosqichda BB kengayishini katta o'zgarishlarsiz amalga oshirish vositalari kiritilgan. BB administratori uchun yaratilgan vositalar tizimni boshqarish imkoniyatini yaratadi va berilganlarni saqlashni ta'minlaydi. Administrator hamma foydalanuvchilar uchun qulay bo'lgan tuzilishni tanlaydi.

3- BO'SQICH. 1- berilganlar bazasini boshqarish tizimiga (1960- yillarding hodi)



1.3-rasm.

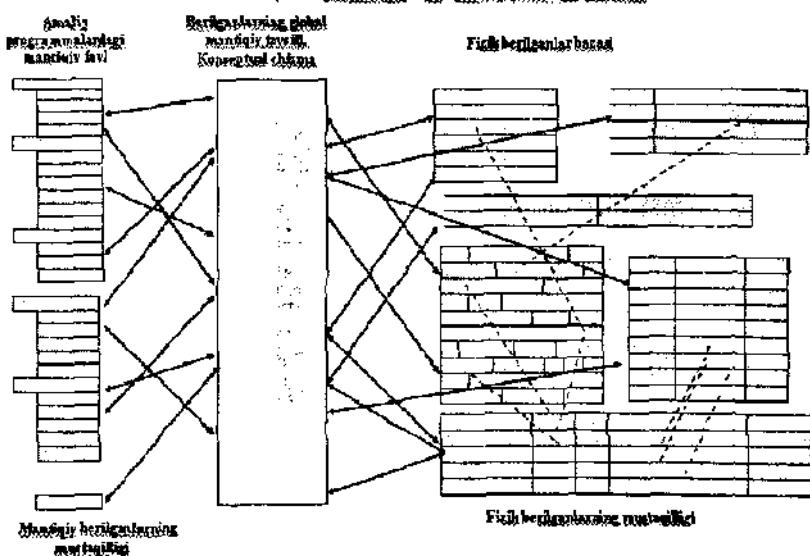
Berilganlarni boshqarishda tizim samarador protseduralar – xavfsizlik, maxsiflylik, butunlik va himoya usullari bilan ta'minlangan. Ayrim tizimlarda BBdan berilganlarni tez topish uchun teskari (invertorlangan) fayllardan foydalaniyan. BB avvaldan rejalashtirilmagan so'rovylarga ham javob berish imkoniyatlari yaratilgan. Berilganlarni ko'chirish vositalari yaratilgan. Programma ta'minotida administrator uchun berilganlarni tavsiflash tili, programmachi uchun komandalar tili va ayrim hollarda foydalanuvchilar uchun so'rovylar tillari yaratilgan.

1.4-rasmdagi BB evolutsiyasining 4- bosqichida berilganlar mustaqil ligining mantiqiy va fizik darajalarini tashkillashtirish g'oyasi ilgari surilgan. Shu rasmdagi katta to'rburchak berilganlarning umumiyligi mantiqiy tuzilishini ifodalaydi, ba'zan uni berilganlarning global mantiqiy taqdimoti deb ham atashadi. Ushbu taqdimot berilganlarning fizik va programmalar-

dagi tuzilishidan tubdan farq qilishi mumkin. Berilganlar bazasining programma ta'minoti programmadagi berilganlarning tuzilishini umumiylantiqiy tuzilishdan oladi va umumiylantiqiy tuzilishni berilganlar tuzilishining fizik taqdimotiga aylantiradi.

1.4-rasmdagi katta to'rtburchak zamonaviy BB tuzilishlarida juda muhim ahamiyatga ega. U tizimdag'i berilganlarning konseptual taqdimotini bildiradi. Ba'zida uni *berilganlar modeli* yoki *konseptual model* deb aniqlashadi, ba'zan esa *chizma* yoki *konseptual chizma* deb atashadi. Konseptual chizma yaratilganda, uning iloji boricha turg'unlikni ta'minlashiga harakat qilinadi. Berilgan xususida programmaching tavsiisi o'zgarganda, berilganlarni fizik saqlash yoki tuzilishi o'zgarganda ham, konseptual model o'zgarmasligi yoki berilganlarning yangi turlarini kiritish uchun kengayish imkoniyati bo'lishi kerak.

**4- ЁСКИЧ ЗАМОНАВИЙ БЕРИГАНЛАР БАЗАСИГА ТАБОЛЛАР**



**1.4-rasm.**

### **Berilganlar tuzilishining uch turi**

1.4-rasmda tasvirlanganidek, berilganlar tuzilishining uchta turi mavjud. Biz ularni berilganlarning *tashqi*, *global mantiqiy* va *fizik tuzilishlari* deb yuritamiz.

*Berilganlarning tashqi tuzilishi* – programmachilar tasavvuridagi taqdimotni bildiradi.

*Berilganlarning global mantiqiy tuzilishi* – umumiy mantiqiy tuzilishni yoki BB konseptual modelini bildiradi va ular asosida berilganlarning turli tashqi tuzilishlarini hosil qilish mumkin. Berilganlarni bunday umumiy mantiqiy tuzilishda tasvirlash, berilganlarning fizik tuzilishiga umuman bog'liqsiz bo'ladi. Berilganlarni bunday umumiy mantiqiy tuzilishda tasvirlash uchun, BBB tizimining programma vositalari tarkibidagi, berilganlarni tasviflash tilidan foydalaniлади.

*Fizik tuzilish* – bu berilganlarni fizik tavsifi va ularni xotira qurilmaliga joylashtirish masalalari bilan bog'liq.

Berilganlar bazasiga xizmat ko'rsatish, saqlash, himoya va nazorat qilish uchun BB *administratori* tushunchasi kiritilgan.

Birinchi yaratilgan berilganlar bazasini boshqarish (BBB) tizimlaridan foydalanish jarayonidagi tajriba shuni ko'rsatdiki, berilganlarning mustaqilligi uchun qo'shimcha yangi darajadagi programma vositalari darkor. Berilganlarning umumiy mantiqiy tuzilishi murakkab va BB kengayishi bilan u majburiy ravishda o'zgaradi. Shuning uchun berilganlarning umumiy mantiqiy tuzilishini o'zgartirishda, ulardan foydalanadigan programma mahsulotlarini o'zgartirmaslikni ta'minlash muhimdir. Shu nuqtayi nazardan berilganlar mustaqilligini ifodalash uchun ikkita darajaga ehtiyoj tug'ilgan. Bu darajalarni berilganlarning *mantiqiy* va *fizik mustaqilligi* deb atalgan.

*Berilganlarning mantiqiy mustaqilligi*, programmalarni o'zgartirmasdan berilganlarning umumiyl mantiqiy tuzilishini o'zgartirish imkoniyatini yaratadi. Bu o'zgartirishlar alo'ata programma ishlataдigan berilganlarni yo'q qilishdan iborat bo'lmasligi kerak.

*Berilganlarning fizik mustaqilligi*, berilganlarning tuzilishi va fizik joylashishi, o'zgarishi va bu o'zgarishlar na berilganlarning mantiqiy tuzilishini, na programmalarni o'zgarishiga olib kelmasligi kerak.

### **«Uch bosqichli» arxitektura**

1.4-rasmning asosiy ma'nosini saqlagan holda, chizmani ANSI/SPARC (American National Standard Institute/Standards Planning and Requirement Committee, Amerika Milliy Standartlar Instituti/Standartlarini Rejalashtirish va Talablar Qo'mitasi/) [2] tavsiya qilgan «Uch bosqichli» arxitektura ko'rinishiga aksantiramiz (1.5-rasm) va natijada berilganlarni zamonaviy tasavvurdagi darajalariga ega bo'lamiz. Bunda tashqi tuzilish – tashqi darajaga, konseptual tuzilish – konseptual darajaga, fizik tuzilish – ichki darajaga akslanadi. Foydalanuvchilar tasavvuridagi berilganlar darjasini, *tashqi daraja* (external level) deb atalsa, BBBT va operatsion tizim

berilganlarni *ichki darajada* (internal level) qabul qiladi. Oraliqdagi *konseptual daraja* (conceptual level) esa, tashqi darajadagi berilganlarni ichki darajaga akslantirish va ularning bir-biriga nisbatan mustaqilligini ta'minlashga xizmat qiladi.

«Uch bosqichli» arxitekturaning maqsadi, berilganlar bazasining foydalanuvchi tasavvuridagi taqdimotni, uning fizik tuzilishidan ajratishdan iborat. Bunday ajratishning sabablari bir nechta:

- har bir foydalanuvchi, berilganlar xususidagi o'zining tasavvurini amalga oshirishi uchun ixtiyoriy berilganlarga murojaat eta olishi va o'z tasavvurini o'zgartira olishi kerak;
- foydalanuvchilarning bazalar bilan o'zaro harakatlari berilganlarning bazada saqlanishdagi alohida xususiyatlariga bog'liq bo'lmasligi kerak (masalan, indekslar va xeshlar);
- berilganlar bazasining administratori bazadagi berilganlarning tuzilishi o'zgartira olishi va bu foydalanuvchilarning berilganlar xususidagi taqdimotiga ta'sir qilmasligi kerak;
- berilganlar bazasining ichki tuzilishi ma'lumotlarni saqlashdagi o'zgarishlarning fizik jihatlariga bog'liq bo'lmasligi kerak;
- berilganlar bazasining administratori BBning konseptual tuzilishini o'zgartirish imkoniyatiga ega bo'lishi va bu foydalanuvchilarga mutlaq ta'sir qilmasligi kerak.

### Tashqi daraja

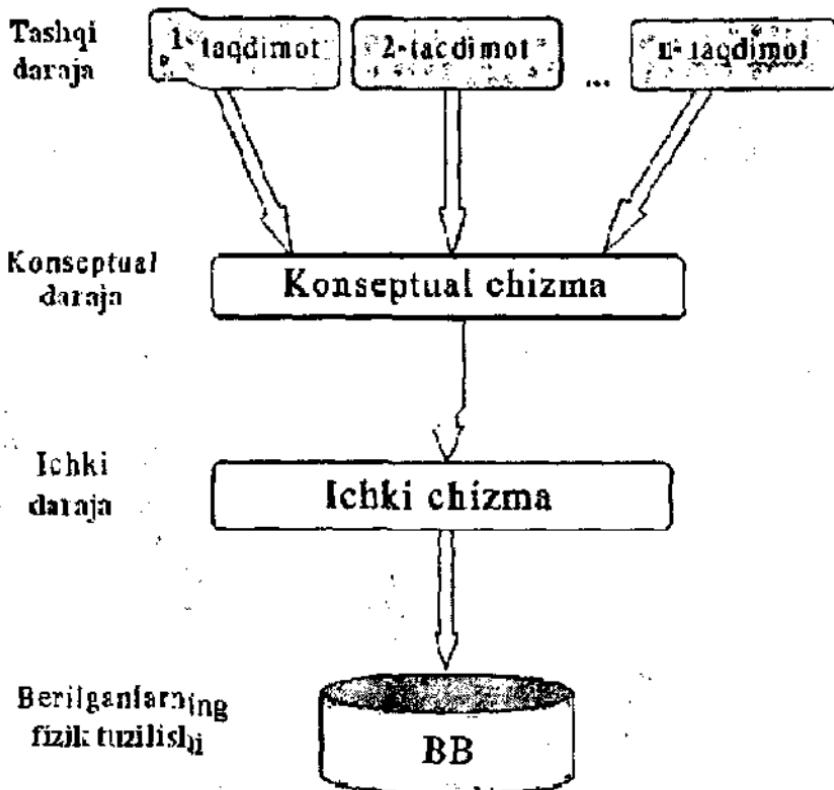
Tashqi daraja bir nechta turli berilganlar bazasining tashqi taqdimotlaridan iborat. Har bir foydalanuvchi «haqiqiy dunyo» taqdimotining o'zi uchun eng qulay shaklidan foydalanadi. Tashqi taqdimot «haqiqiy dunyo»-dagi mohiyat, atribut va aloqalarning foydalanuvchi uchun kerakliklarini ifodalaydi. Qolgan mohiyat, atribut va aloqalarning foydalanuvchi uchun keraksizlari ham berilganlar bazasida akslanishi mumkin, lekin u bularning mavjudligini bilmasligi ham mumkin.

Bundan tashqari, bir xildagi berilganlarni turli taqdimotlar har xil akslantirishi mumkin. Masalan, bitta foydalanuvchi sanani «kun, oy, yil» formatida, ikkinchisi esa «yil, oy, kun» formatida saqlashi mumkin. Ba'zi bir taqdimotlar esa hosila yoki hisoblanadigan berilganlardan iborat bo'lishi mumkin, ular zaruriy paytda yaratiladi. Masalan, yoshni saqlovchi atributga e'tibor beraylik. U yosh qiymati tarzida saqlanmaydi, agar shunday saqlanganda ularni har kuni yangilab turish kerak bo'lardi. Buning o'miga bazada tug'ilgan sana saqlanadi, yosh esa BBBT vositalari orqali hisoblanadi. Taqdimotlar, shuningdek, bir nechta obyektlarning kombinatsiyalangan yoki hosila berilganlaridan tuzilishi mumkin.

## Konseptual daraja

Bu daraja BBda saqlanadigan berilganlarning tavsiflari va ular orasidagi aloqalardan iborat bo'ladi. Bu daraja hamma berilganlar bazasining mantiqiy tuzilishini saqlaydi. Konseptual darajada quyidagi a'zolar ishtirok etadi:

- hamma mohiyatlar, ularning atributlari va aloqalari;
- berilganlarga qo'yiladigan cheklashlar;
- berilganlat xususidagi semantik ma'lumotlar;
- xavfsizlikni ta'minlash va berilganlar butunligini saqlash xususidagi ma'lumotlar.



1.5-rasm.

Konseptual daraja har bir tashqi taqdimotni qo'llab-quvvatlaydi, ya'ni foydalauvchiga zarur bo'lgan har bir berilgan shu darajada bo'tishi shart. Lekin, konseptual daraja berilganlarni saqlashi usullari to'g'risidagi hech qanday ma'lumotlarni saqlamaydi.

### **Ichki daraja**

Ichki daraja berilganlar bazasini fizik amalga oshirish tavsiflari, optimal unumdonlikka erishish va disk hajmidan iqtisodli foydalanishni ta'minlash masalarini hal qiladi. U berilganlar tuzilishining tavsifi va tashqi xotirlardagi berilganlarni saqlash uchun alohida fayllarning tuzilish tavsiflarini saqlaydi. Ushbu darajada BGBT bilan operatsion tizimning kirish usullari orasidagi o'zaro harakatlar amalga oshiriladi. Ichki darajada quyidagi ma'lumotlar saqlanadi:

- berilganlar va indekslarni saqlash uchun disk hajmining taqsimoti;
- saqlangan yozuvlarning qo'shimcha tavsiflari;
- yozuvlarni joylashishi xususidagi axborotlar;
- berilganlarni siqish va tanlangan shifrlash usullari xususidagi axborotlar.

Ichki darajadan quyida berilganlarning fizik tuzilishlari, ya'ni berilganlar bazalarining tashqi xotiradagi fizik joylashish tafsilotlari operatsion tizim nazorat qiladigan, lekin BGBT boshqaruvidagi bo'lgan masalalar hal qilinadi.

### **Chizmalar**

Berilganlar bazasining umumiyligi tavsifi *berilganlar bazasining chizmasi* deb ataladi. «Uch bosqichli» arxitekturaga nisbatan berilganlar bazasining uch turi mavjud (1.5-rasm). Eng yuqori darajada berilganlarning turli taqdimotiga mos keluvchi bir nechta *tashqi chizmalar* yoki *qism-chizmalar* mavjud. Konseptual darajadagi berilganlar bazasining tavsifini *konseptual chizma*, eng quyi darajadagisini *ichki chizma* deyishadi. *Konseptual chizma* hamma mohiyatlar, atributlar va ular orasidagi aloqalar hamda zaturiy berilganlar butunligi cheklashlarini o'z ichiga oladi. *Ichki chizma* berilganlar modelining ichki tavsifidan, ya'ni saqlanuvchi yozuvlar va berilganlar maydonlarining taqdimot usullari, indekslar va tanlangan xesh chizmalardan iborat bo'ladi.

Har bir BB uchun faqat bitta konseptual va bitta ichki chizma mavjud.

BGBT keltirilgan uchchala turdag'i chizmalar orasida moslik o'matilishiga hamda ularning orasida ziddiyatlar chiqmasligiga javobgar. Ixtiyoriy tashqi va ichki chizmalar orasidagi mosliklarni o'matish uchun BGBT kon-

septual chizmadagi ma'lumotlardan foydalanadi. Konseptual chizma ichki chizma bilan konseptual ichki akslanish orqali bog'langan. U BBB tizimiga, konseptual chizmada mantiqiylar yozuvdan iborat berilganlarni, fizik saqlash qurilmalaridan yozuv yoki yozuvlar to'plamiga akslanganlarini izlash imkoniyatini beradi. Konseptual chizma, shuningdek, obyektlar va atributlarning nomlaridagi ixtiyoriy tasovutlarni, atributlarning kelish tartiblari, ulardag'i berilganlar turlari va h.k. aniqlaydi. Har bir tashqi chizma konseptual chizma bilan konseptual tashqi akslanish orqali bog'langan. Uning yordamida BBBT foydalanuvchi taqdimatining har birining nomini konseptual chizmaning mos qismiga akslantiradi.

### Xulosa

*BBB tizimlaridan awalgi bosqich hu sayllar tizimi, ya'ni ilovalar termasi bo'lib, foydalanuvchilar uchun alohida zaruriy amallarni masalan, hisobtorni yaratishni bajargan. Har bir programma o'zining alohida berilganlarini aniqlab, ana shu berilganlarnigina boshqargan. Fayl tizimi qo'lda bajariladigan hisob-kitoblariga nishbatan ancha ilg'or texnologiya bo'lgan bo'lsa ham, undan foydalanishda ancha muammolar yuzaga kelgan. Bu muammolarning eng asosiyasi berilganlarning ortiqchaligi va ularning programmalarga hog'liqligidan iborat bo'lgan.*

*Ana shu va boshqa muammolarni hal qilish uchuni berilganlarning «Uch bosqichli» arxitekturasi yaratilgan.*

### Nazorat uchun savollar

1. Ma'lumotlarga ta'rif bering.
2. Berilganlarga ta'rif bering.
3. Baza deganda nimani tushunasiz?
4. Berilganlar bazasini to'liq ta'riflab, izoh bering.
5. Berilganlar bazasini nima boshqaradi?
6. Berilganlar bazasini o'zgarishi ilovalarga qanday ta'sir qiladi?
7. Berilganlar bazasiga ehtiyoj nima sababli paydo bo'ldi?
8. Berilganlar bazasi tushunchasi qachon va nima sababdan paydo bo'lgan?
9. Berilganlarning mustaqilligi deganda nimani tushunasiz?
10. Berilganlar bazasining evolutsiyasi nechta bosqichdan iborat?
11. Berilganlar bazasi evolutsiyasi 1-bosqichining asosiy jihatni nimadan iborat bo'lgan?

12. Berilganlar bazasi evolutsiyasi 2-bosqichining asosiy jihatni nimadan iborat bo'lgan?
13. Berilganlar bazasi evolutsiyasi 3-bosqichining asosiy jihatni nimadan iborat bo'lgan?
14. Berilganlar bazasi evolutsiyasi 4-bosqichining asosiy jihatni nimadan iborat bo'lgan?
15. Berilganlar bazasini konseptual modelini ta'riflang.
16. Berilganlarning tuzilishidagi uchta turni ta'riflang.
17. Berilganlar bazasining «Uch bosqichli» arxitekturasini tushuntirib bering.
18. Berilganlar bazasining «Uch bosqichli» arxitekturasini tashkil etuvchi darajalarini aytib bering.
19. «Uch bosqichli» arxitekturaning maqsadi nimadan iborat?
20. Taqdimot deganda nimani tushunasiz?
21. Taqdimotdan fizik tuzilishni ajratishdagi asosiy sabablarni aytib bering.
22. Berilganlar bazasi chizmasi deganda nimani tushunasiz va ular necha xil bo'ladi?
23. «Uch bosqichli» arxitekturani amalga oshiruvchi nima?

## 2 - bob. OB'YEKTLAR VA ATRIBUTLAR

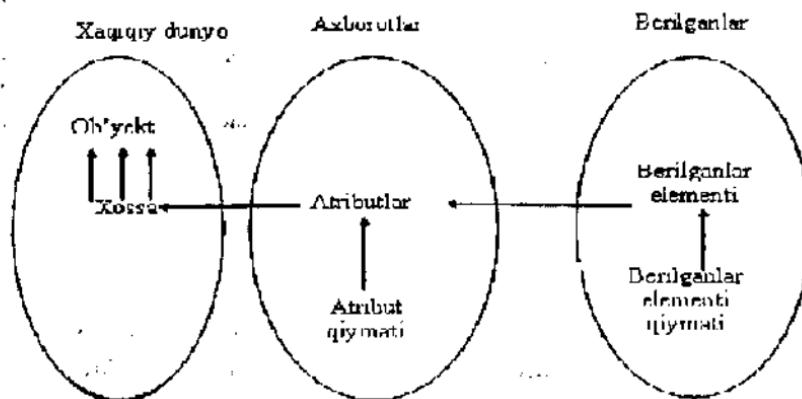
**Tayanch iboralar:** obyektlar, berilganlarga ishlov, obyektlar termasi, atribut (xususiyat), axborotning uch sohasi, yozuv, ikki o'chovli fayl, ikki o'chovli massiv, identifikator, kalit, birlamchi kalit, birtashgan kalit, (murakkab kalit, tarkibli kalit), muqobil kalit, so'rov, oddiy so'rov, murakkab so'rov.

Elementlari to'g'risidagi axborotlari saqlanuvchilar obyektlar deb ataladi [84]. Obyektlar haqiqiy (masalan, inson, mahsulot yoki yashash hудуди) va abstrakt (masalan, hodisa, fan nomi, topshiriq raqami yoki tashkilotning hisob raqami) bo'lishi mumkin. Masalan, ta'lim sohasida muktab, litsey, kollej, bakalavriat, magistratura obyektlar bo'ladi. Lekin shu obyektlarga tegishli insonlarning har birini alohida obyekt sifatida e'tirof etish mumkin.

Obyekt turli xossalarga ega bo'ladi (masalan, ism, rang yoki vaqt o'chovi) va ularni jumladan, kompyuterlar orqali turli qurilmalarida saqlash imkoniyati mavjud. Ana shu xossalarning har bir elementi berilganlardan iborat. Berilganlarga ishlov deganda, ular ustida bajariladigan izlash, qo'shish, olib tashtash, o'zgartirish algoritmlarini amalga oshiruvchi funksiyalar faoliyatining natijasini tushunamiz. Berilganlarga ishlov ko'pincha, bir xildagi xossa (xususiyat)ga ega bo'lgan obyektlarning to'plami bilan ishlanaadi. Masalan, talabalarни оlaylik, ularning har biri to'g'risidagi bir xil xususiyatlari berilganlarga saqlanadi. Bir xildagi xususiyatga ega bo'lgan obyektlarni, obyektlar termasi deb ataymiz. Biz yuqorida keltirgan tushunchalarimiz haqiqiy dunyoga tegishli. Biz har bir obyektlar xususidagi ma'lumotlarni yozuvlar tarzida ifodalaymiz. Yozuvlar esa obyektlarning xususiyatlari (atributlari) va ularning qiymatlari bilan bog'langan. Biz bundan keyin, atribut atamasining mazmuni xususiyat bo'lgani va u butun jahon bo'yicha sohaga tegishli ilmiy va amaliy adabiyotlarda qabul qilingan atama bo'lgani uchun, shu atribut atamasida to'xtaymiz. Masalan, «Talaba» obyekting atributlari familiyasi, ismi, otasining ismi, tug'ilgan sanasi, fakulteti, yo'nalishi va boshqalardan iborat bo'lishi mumkin.

## Axborotning uch sohasi

Axborot tushunchasi bilan bog'liq bo'lgan uchta soha mavjud va undagi o'zaro bog'lanishlar 2.1-rasmida keltirilgan [84]. Birinchisi, bu haqiqiy dunyo bo'lib, unda haqiqatan ham obyektlar mavjud va ularning har biri o'z xossalariiga ega. Ikkinci soha, bu axborotlarning elementlari va ularning qiyamatlari bilan bog'langan. Uchinchisi esa, berilganlarning elementlari va ularni ifodalashdagi belgilari, satrlar yoki axborot elementlарini kodlashdagi bittardan iborat. Uchinchisi sohani yanada chuqurlashtirib, berilganlarning amaliy programmachi tasavvuridagi taqdimoti, BB administratorining tasavvuridagi umumiyyatini tuzilish va berilganlarning fizik taqdimiga ajratish mumkin.



2.1-rasm.

Berilganlar elementlarining har biri o'z qiyamatiga ega bo'lishi kerak va u atributni ifodalaydi, atribut esa obyektni identifikasiyalaydi. Biz masalan, qizil, ko'k va binafsha kabi berilganlarning qiyatlarini saqlashimiz mumkin, lekin bu qiyatlar boshqa atributlar orqaligina bog'lanib biror mazmunga ega bo'lishi mumkin.

Faraz qilaylik, «*Talaba*» obyektingin atributlari faqtgina familiya, ism, otasining ismi, tug'ilgan sanasi, fakulteti, yo'nalishi va kursdan iborat bo'lsin va shu atributlarga tegishli qiyatlar majmuasini yozuv deb qabul qilamiz. Demak, N ta talaba xususidagi berilganlar N ta yozuvdan iborat va har bir yozuv zaruriy ma'lumotni beruvchi k ta berilganlar elementidan iborat. Quyida, berilgan bitta obyektni, ya'ni talabani «*Talaba*» obyektidan izlash misolida, berilganlar orasidagi bog'liqliklarni ko'tramiz.

Familiyaning o'zi yakka holda aniq bir ta'labha xususidagi hech qanday ma'lumotni bermaydi, chunki bir xildagi familiyali talabalar ko'p bo'lishi mumkin. Endi uni ism bilan birlgilikda qaraylik. Fakultetda bir xildagi familiya va ismgaga ega bo'liganlar ham bir nechta talaba bo'lishi mumkin. Agarda qaralayotgan juftlikka, endi otasining ismi qo'shilsa, bu aniq bitta talaba bo'lishi mumkin. Lekin, biz izlayotgan talaba bo'lmasi ham mumkin, chunki aynan xuddi shunday familiya, ism va otasining ismiga ega bo'ligan kamida yana bitta ta'labha bo'lishi mumkin, hattoki ikkalasi bitta kurs va guruhda o'qishi mumkin. Atributlardagi berilganlar orasidagi bunday bog'lanish juda ham kam uchraydigan, lekin bo'lishi mumkin bo'ligan holat. Atributlarning bunday bog'lanishi ancha mazmundor bo'ldi, lekin ikkita yozuv bir xildagi qiymatlarga ega bo'ldi. Demak, biz, izlagan talabamiz qaysi biri ekanligini hali aniqlay olmadik. Buning uchun mazmunni yanada boyitishimiz, ya'ni talaba xususida yana boshqa berilganlar bilan obyekt xususidagi ma'lumotlarni boyitishimiz va aniq mazmunga ega bo'lismiz mumkin.

*Aniq bir kortej elementlari to'liq va aniq mazmunga ega bo'lishi uchun, undagi atributlar sonini zaruriy atributlar bilan to'ldirish kerak, ya'ni obyekt xususidagi berilganlar qanchalik yetarli va aniq bo'lsagina, berilganlar bazasidagi berilganlar asosida to'g'ri qaror chiqarish mumkin. Chunki, berilganlar bazasidan axborot tizimlarida foydalanishning asosiy maqsadi berilgantarni ishlovi natijasida to'g'ri qaror qabul qilishdan iborat. Bu esa berilganlarning ahamiyati qanchalik muhim va har birining o'z o'rni bortigidan darak beradi hamda atribut va berilganlar orasidagi bog'lanishlarning naqodar ahamiyatlari ekanligini ko'rsatadi.*

### **Ikki o'chovli fayllar**

Kattaliklarni berilganlar elementi bilan aloqasini va berilganlar elementlarning mos obyektlarning atributlari bilan aloqasini amalga oshirishning eng oddiy usuli – ularni birlgilikda cheklangan ketma-ketlik holatida saqlashdan iborat (2.2-rasm). Ushbu rasmida berilganlar elementlarning termasi va har bir elementning qiymati keltirilgan. Har bir satrda berilganlar elementlarning aniq bir obyektga tegishli qiymatlari keltirilgan. Ushbu obyektlar termasining nomi XIZMATCHI bo'lisin. Bundan buyon obyektlar termalarining nomi va ularning atributlarini bosh harflarda yozamiz.

Bundan buyon matnda, biz doimiy ravishda *obyektlar termasi* iborasining o'miga, obyekt atamasining o'zidan foydalanamiz. Lekin zaruriy holatlarda, albatta obyektlar termasi iborasini ishlamatamiz.

Har bir ustun aniq bir atribut bilan bog'langan berilganlar elementining aniq bir turini ifodalaydi. Diagrammaning yuqori qismida to'g'ri to'rtbur-

chakdan tashqarida atributlarning nomlari yozilgan. Birinchi ustundagi berilganlar elementlari obyektlarni identifikasiyalaydi. Obyektlarni identifikatorlovchi atributning nomi XIZMATCHI\_RAQAMI. Taqdimot shaklida har bir atributdag'i qiyamalarning formatlari berilgan. Ushbu rasmdagi formatlar – N - butun sonni, A - matnli berilganni, V - ikkilik sanoq tizimidagi berilganni, raqamlar ularning o'chov birligini bildiradi, masalan:

- N6 – 6ta raqamdan iborat butun son;
- A30 – 30ta belidan iborat matn;
- B1 – 1 bitdan iborat ikkilik son.

2.2-rasmdagi satrlarni tashkil etuvchi berilganlar elementlari o'zaro bog'langan, ular yozuvlar deb ataladi. Ikkita elementdan iborat yozuv ikkilik yozuv deyiladi. Nta elementdan iborat bo'sha, Ntalik yozuv deyiladi. Fayl har xil turdag'i elementlarga ega bo'lgan yozuvlarning to'plamidan iborat. Shunday qilib, faylga berilganlar elementlarini saqlovchi, ikki o'chovli massiv deb qarash mumkin.

Atribut nomi	Birinchi kelti				Minishib kelti				Tengsizlik	Tengsizlik	Rasmi
	Xizmatchi_Raspni	Xizmatchi_Ismi	Nomi	Kodusi	Teg_niga_savchi	Nomi	Kodusi				
Tegsizchilik shaxsi	N5	A30	B1	N2	N6	N3	N2	A13	N4		
Atribut qiyamali	531300	Santsov Hefer	:	03	100362	044	73	Hsobchi	2000		
Yozuv	287.91	Chonov Ahmed	:	05	101052	172	43	Furmanek	4800		
	535502	Lakarov Zennur	0	07	090988	044	02	Lakarov	1100		
	796324	Melis Soqiq	:	11	011162	090	13	O'qibekli	1400		
	1597.8	Aliev Farhad	:	13	021272	172	43	Diroqor	1700		
	05.983	Ismurova Nozima	0	03	091120	019	73	Hsobchi	2000		
	036453	Aidov Ramon	:	08	110382	044	02	Masahatchi	1200		
	24.6.8	Shabou Vel.	:	07	071265	172	55	Muhamedis	2500		
	76.903	Sairov Anvar	:	11	011055	172	21	Memor	3200		
	372921	Kubayeva Samboer	0	03	020482	080	93	Akrisa	2100		

O'tidri  
saqlab  
olvchi

Doxan

Bezi xizmatchilarning o'z boshqa  
faylida ch'isllari oyinaydi

2.2-rasm.

2.2-rasmidagi berilganlar elementining oddiy ikki o'chovli akslanishi, ba'zan *tekis akslanish* deb atashadi. Shuning uchun ham *tekis* yoki *ikki o'chovli fayl* tushunchalari ham kiritilgan.

Ba'zi atributlarning o'zi boshqa fayllardagi obyektlarni *identifikatsiya* shakida mumkin. Masalan, 2.2-rasmidagi BO'LIM\_RAQAMI ushbu XIZMATCHI obyektdagi atributdan iborat, lekin uning qiymatlari qaysidir BO'LIM\_HUDUD «fayl»ida shu «fayl»ni identifikatsiyalashi mumkin (2.3-rasm), ya'ni xizmatchilar ishlaydigan tashkilotning bo'limlari joylashgan hududlardagi manzillardan. Masalan, tashkilot bo'lini qaysi mavze, ko'cha, uyda joylashgani xususidagi berilganlardan iborat bo'lishi mumkin.

### Kalitlar

Obyekt elementlariga murojaat qilish uchun atributlardan bittasi obyektni bir qiymatli identifikator vazifasini o'tashi kerak. XIZMATCHI obyekting identifikatori XIZMATCHI\_RAQAMI bo'lishi mumkin. TALABA obyekting identifikatori TALABA\_RAQAMI yoki REYSTING\_DAFTARI bo'lishi

Bo'lim	Mavze	Ko'cha	Uy
044	Шайхонтохур	Олмазор	45
172	Олмазор	Уста-Ширин	48
090	Чилонзор	Кальни	156
019	Шайхонтохур	Ўқчи	26

2.3-rasm.

mumkin. *Obyektni aniqlovchi atribut noyob bo'lishi shart*. Ayrim vaziyatlarda obyektdagi yozuvlarni identifikatsiyalash uchun bittadan ortiq atribut kerak bo'lishi mumkin. Masalan, samolyot reyslari xususidagi yozuvni identifikatsiyalash uchun, albatta REYS\_RAQAMI va SANA atributlari zarur. Faqatgina reysning nomeri yetarli emas, chunki shu reys bilan samolyotlar har kuni uchishi mumkin.

Yozuvni noyob ravishda aniqlovchi atributga kortejning kaliti sifatida qaraladi va u *birlamchi kalit* deb ataladi. Kalit bittadan ortiq atribut elementlaridan tuzilishi mumkin, u holda uni *birlashgan kalit, murakkab kalit yoki tarkibli kalit* deb atashadi. Birlashgan kalitlar REYS\_RAQAMI + SANA yoki (REYS\_RAQAMI, SANA) tarzda belgilanadi. Ba'zan birlamchi kalit uchun bir nechta atributlardan foydalaniladi. Masalan, aeroflot kassasini obyekt sifatida olaylik va u REYS\_RAQAMI, SANA,

MANZIL, KASSA\_TUSHUMI atributlaridan iborat bo'lsin. KASSA\_TUSHUMI atributini identifikatsiyalash uchun REYS\_RAQAMI, SANA, MANZIL atribut elementlaridan foydalanish zarur. Shuning uchun aeroflot kassasini obyekti uchun *birlamchi kalit* REYS\_RAQAMI + SANA + MANZIL bo'ladi.

Yozuvlar biron bir noyob bo'smagan xususiyatlar bilan ham bog'langan bo'lishi mumkin. Masalan, 2.2-rasnidagi XIZMATCHI obyektdida KAS-BI yoki KASB\_KODI atributlarida bir nechta takrorlanuvchi elementlar mavjud. Bu atributlarning har biri alohida kalit bo'la olmaydi. Lekin birlashgan holda, noyob *murakkab kalit* bo'la oladi va obyekt yozuvlarini bir qiymatli identifikatsiyalaydi. Agarda talaba obyektdagi familiyasi, ismi va otasining ismidan iborat uchta atributlarda aynan o'xshash qiymatlar bo'limasa, ularni ham kalit sifatida ishlash mumkin. Ana shunday kalitlar, *birlamchi kalitga muqobil kalitlar* deb ataladi.

### So'rovlar

Berilganlar bazasi Nta obyektga tegishli yuzlab ikki o'chovli fayllarni saqlashi mumkin. Lekin berilganlar bazasining asosiy maqsadi berilganlarga ishlov berishdan iborat ekanligini biz yuqorida ta'kidlagan edik. Obyektlar ustida so'rovlarni tashkil qilish uchun turli tillardan foydalanishni biz keyingi boblarda batapsil o'rganamiz. Avval, so'rovlarni tashkillashtirishda ko'nkmalar hosil qilish uchun, bu masalani mantiqiy jihatdan ko'rib chiqamiz.

So'rovlar ikki turga oddiy va murakkab so'rovlarga bo'linadi. *Oddiy so'rov* deganda, bir vaqtida bitta atributga ishlov berish tushuniladi. Bu atribut berilganlar bazasidagi bitta obyektdan yoki bir nechta obyektlardan izlanishi mumkin. Bunda faqatgina bitta yoki bir nechta kalitlardan foydalaniladi.

*Murakkab so'rov* deganda esa, bir vaqtida bir nechta atributlarga ishlov berish tushuniladi. Bu atributlar bitta yoki bir nechta obyektga tegishli bo'lishi mumkin. Murakkab so'rovlarda turli xildagi bir nechta kalitlardan foydalaniladi.

2.1-jadvalda oddiy va murakkab so'rovlarga, so'rov turi va so'rov turi ga 12ta misollar keltirilgan. Jadvalda E – obyektni, A – atributni, V – qiymatni, \* o'mida esa =, ≠, ≤, ≥, > taqqoslash belgilaridan ixtiyorisi kelishi mumkin.

Semiz qora kursivda keltirilgan savollar murakkab savollar, qolganlari esa oddiy savollar. Ushbu savollarga javoblarni topish uchun XIZMATCHI obyektdagi oxirgi ustunni olib tashlaymiz hamda ba'zi atributlarning nomini, faqat qulaylik uchun, ozgina o'zgartiramiz (2.4-rasm).

## 2.1-jadval

Nº	So'rov shahidi	So'rov turi	Misol
1.	A(E)*?	Atribut qiyomatiga aniqlashga yo'naltirilgan oddiy so'rov	287191- raqamli xodinning may oyidagi maoshi qancha bo'lgan?
2.	A(?) * V	Berilgan atributning qiyomatiga nisbatan hamma obyektlarni aniqlash	Qaysi bu xodim noyabr oyida 2500dan ortiq maosh olgan?
3.	?(E) * V	Berilgan obyekti bo'yicha ushbu qiyomatiga ega hamma atributlarni aniqlash	535502- raqamli xodinning maoshi qaysi oyldarda 2000dan kam bo'lgan?
4.	?(E)*?	Obyekti to'g'risidagi hamma ma'lumatlarni elish	287191- raqamli xodimiga tegishli hamma ma'lumotlar chiqarilsin?
5.	A(?) * ?	Ushbu atribut qiyomatiga ega bo'lgan hamma obyektlarni aniqlash	Iyul oyi uchun hamma xodimlarning maoshlari chiqarilsin?
6.	?(?) * V	Ushbu qiyomatiga ega bo'lgan hamma obyektlarning hamma atributlarni aniqlash	Her bir xodinning 4000 maosh olgan oyalar aniqlansin?
7.	A(E) * [a,b]	Arsivdagi fayllar bo'yicha ularning nomlarini aniqlashga yo'naltirilgan murakkab so'rov	535502- raqamli xodim qaysi yillarda may oyidagi maoshi 1200+3000 bo'lgan?
8.	A(?) * [a,b]	Berilgan atributning oraliq qiyomatiga nisbatan hamma obyektlarni aniqlash	Qaysi xodimlar yanuvarda 1700+2400 oraliqdagi maosh olgan?
9.	?(E) * [a,b]	Berilgan obyekti bo'yicha oraliq qiyomatiga ega hamma atributlarni aniqlash	535502- raqamli xodinning maoshi qaysi oyldarda 4000+4500 bo'lgan?
10.	?(E) * [a,b]	Obyekti to'g'risidagi hamma ma'lumatlarni arxivdagidan fayldan elish	241618- raqamli xodim 2010 yilning qaysi oyida maoshu 1500+4000 bo'lgan?
11.	A(?) * (a,b)	Ushbu atribut qiyomatiga ega bo'lgan hamma obyektlarni arxivdagidan fayldan aniqlash	2010 yil noyabr oyi uchun hamma xodimlarning 3000+5000 oraliqdagi maoshlari chiqarilsin?
12.	?(?) * (a,b)	Ushbu qiyomatiga ega bo'lgan obyektlarning arxivdagidan fayldan hamma atributlarini aniqlash	2009 va shu yil uchun har bir xodinning maoshi 3000+5000 oraliqdagi oylik aniqlansin?

MAOSH\_2010 nomi bilan XIZMATCHI obyektiga bog'liq yana bir massiv keltiramiz yoki endi aniqlik kiritib, uni deyarli jadvalga o'xshagan uchun, jadval deb atashimiz ham mumkin (2.5-rasm). Ushbu jadvalda 2.4-rasmdagi XIZMATCHI obyektidagi xizmatchilarning raqamlariga bog'langan ravishda, ana shu obyektlarning 2010-yilda olgan maoshlari keltirilgan bo'lsin. 2.1-jadvalga e'tibor bersangiz, obyektlarning 2009-yildagi maoshlari xususida ham so'rovlar mavjud. Shuning uchun MAOSH\_2009 nomli jadvalni yaratishni hujmatli talabaga qoldiramiz.

Raqami	Familiya_ism	Jinsi	Kasbi	Tug'ilgan sana	Bo'lim	Kasb_sana	Lavozim kodi
537300	Sattorov Hasan	1	03	100365	044	73	Hisobchi
287191	Choriev Ahmad	1	05	101055	172	43	Prommachi
535502	Lakaeva Zamira	0	07	090988	044	02	Laborant
796324	Meliev Sodiq	1	11	011162	090	13	O'qituvchi
159718	Aliev Farhod	1	13	021278	172	43	Duradgor
051883	Temurova Nozima	0	03	091170	044	13	Hisobchi
036453	Jalilov Ilhom	1	08	110981	044	02	Maslahatchi
241618	Shizov Vali	1	07	071265	172	55	Muhandis
761903	Salimov Anvar	1	11	011059	172	21	Me'mor
372921	Kubaeva Sanobar	0	03	020482	090	93	Aktarsa

#### 2.4-rasm.

Uchchala jadval tayyor bo'ldi, deb faraz qilgan holda, ana shu jadvallar sizning birinchi XIZMATCHI o'quv loyihangizdag'i, soddagina BB bo'lsin. Shu jadvallardan foydalangan holda, 2.1-jadvalda keltirilgan savollarga javoblarni topishni ham Sizga qoldiramiz. Ushbu mashqlarni bajarish natijasida, Sizda BB bilan ishlov jarayonidagi faqatgina berilganlarni tanlash amali qanday bajarilishi xususidagi boshlang'ich tasavvur shakllangan bo'lsa, demak, Siz BB nima ekanligini anglab yetgansiz va to'g'ri yo'nalishdasiz. Sizga keyingi boblamni o'zlashtirishda muvaffa-qiyatlar tilaymiz.

raqami	yanvar	fevral	mart	aprel	may	iyun	iyul	avgust	sentyabr	oktyabr	novabr	dekabr
537300	1200	1400	1900	2800	3400	4000	4250	5000	4500	3650	3400	3500
287191	5000	5500	6000	4500	4800	5300	6700	7000	7850	7400	7250	8000
535502	1300	1000	1200	1500	1600	1450	1500	1800	1750	2200	2400	2300
796324	3000	3500	4000	4000	4000	3500	3700	4000	3500	3400	3250	4000
159718	2200	2400	2500	2800	3000	3800	3250	3500	3600	3650	3450	3500
051883	3200	2400	2900	3800	4400	4000	4250	5000	4500	3700	4400	4200
036453	4200	4400	3900	2800	3400	3300	4300	4500	4500	4700	4400	4600
241618	5200	5550	5900	5800	5400	5300	5300	5500	5500	5700	5400	5600
761903	6200	4550	4900	4800	5400	5400	5400	6500	4700	5300	5700	5600
372921	3200	3400	2700	3700	5400	4000	4250	5000	3500	3700	4200	4500

#### 2.5-rasm.

#### Xulosa

Har qanday axborot tiziminining asosiy tuzilishida BBB tizimlari yotadi. BBB tiziminining paydo bo'lishining negizida, albatta axborotning aniq vogelikdag'i obyektning tashkil etuvchisi sifatida qaralishidir. Ana shu

*sahabdan, BBB tizimlarning paydo bo'lishi, deyarli hamma tashkilotlarning ish usularini o'zgartirishga olib keldi.*

*Hozirgi kunda BBBT ilmiy izlanishlardagi jadal rivojlanayotgan obyektlardan biri bo'lib, ko'pchilik juda muhim masalalar uchun hali qoniqarli natijalarga erishilmadi.*

### **Nazorat uchun savollar**

1. Obyektga ta'rif bering.
2. Obyekt va berilganlar orasidagi bog'lanishni ifodalang.
3. Berilganlarga ishlov deganda nimani tushunasiz?
4. Obyektlar termasi tushunchasini izohlang.
5. Axborotning uchta sohasi nimalardan iboratligini tushuntiring.
6. Obyekt bilan atribut orasidagi bog'lanishni izohlang.
7. Ikki o'chovli fayllarga misol keltiring.
8. Kalitni ta'riflang va uning turlarini aytib bering.
9. Yozuv va domenni ta'riflang.
10. So'rov deganda niinani tushunasiz?
11. Oddiy so'rovni ta'riflang.
12. Murakkab so'rovni izohlang.
13. 2.4 va 2.5-rasmidagi ma'lumotlar asosida ikkita oddiy so'rovlar tuzing va javobini keltiring.
14. 2.4 va 2.5-rasmidagi ma'lumotlar asosida ikkita murakkab so'rovlar tuzing va javobini keltiring.
15. Maosh\_2009 jadvalini yaratting hamda 2.4-rasmidagi ma'lumotlar va yaratilgan jadval asosida 3 ta oddiy so'rov tuzing.
16. Yaratilgan maosh\_2009 jadvali va 2.4-rasmidagi ma'lumotlardan foydalanib 3 ta murakkab so'rov tuzing.

---

### **3-bob. BERILGANLAR MODELI**

**Tayanch iboralar:** berilganlar modeli, kuchsiz turdag'i modellar, kuchli turdag'i modellarda, berilganlar elementi, konseptual chizma, chizmaning nusxasi, yozuvning nusxasi, aloqa, «birga-bir» aloqa, «birga-ko'p» aloqa, «ko'pga-bir» aloqa, «ko'pga-ko'p» aloqa, yozuvlar orasidagi aloqalar.

Berilganlarni o'zaro bog'liqliklarini abstrakt talqin qilish uchun ularning orasidagi munosabatlarni ko'rsata oladigan rivojlangan uslublar mavjud bo'lishi kerak. Berilganlar orasidagi munosabatlarni qo'yilgan talablarga nisbatan amalga oshirish imkoniyatini beruvchi intellektual vositani berilganlar modeli deb ataladi. Berilganlar modeli abstraksiya usuli bo'lib, u obyektdagi alohida elementlarni emas, balki yaxlit obyektni atributlari bilan ko'rish imkoniyatini beradi. Abstraksiya usuli kerakmas mayda narsalarga e'tibor qaratmasdan, obyektlarning asosiy xossalariiga e'tibor qaratishni talab qiladi. Berilganlarning modellarida abstraksiya usuli berilganlarni turkumlarga ajratilgan holatda tahlil qilish va berilganlar bazalarni to'g'ri loyihalashga xizmat qiladi. Bundan tashqari, ko'rilgan turkumlarga tayangan holatda umumiylar turkumlarni qurish imkoniyatini beradi.

Haqiqiy dunyodagi turli jabhalarni akslantiruvchi har xildagi modellar mavjud. Masalan, fizik modellar fizik xususiyatlarni tushunishga, matematik modellar matematik va maxsus belgilarni orqali dunyoning abstrakt yoki jarayonning aniq tavsifini tushunishga, iqtisodiy modellar esa ulardag'i alohida xossalarni akslantirish orqali iqtisodiyotning rivojlanishini bashorat qilishga yordam beradi. Berilganlar modeli berilganlarning ma'nosini qisman taqdim etishni va bu bilan haqiqiy dunyodagi bilimlar bilan bizni qisman ta'minlashi mumkin.

Haqiqiy dunyodagi jarayonlarni akslantiruvchi yagona modelni qurish mumkinmi, degan savolga yo'q, deb aniq javob berish mumkin. Chunki haqiqiy dunyodagi jarayonlar va bilimlar – bu ochiq tizimdan iborat. Shunday ekan, u doimiy ravishda o'zgarib turadi. Hattoki, aniq bir sohada to'plangan bilimlar orqali ham shu soha to'liq o'rganib bo'lindi, deb aytish mumkin emas. Muhimi, modeldagi bilimlar hajmi va berilganlarning ma'nosi, berilganlarni foydalinish maqsadiga mos kelishidadir.

Shu nuqtayi nazardan ham berilganlarning predmet soha bilan uzviyligi ta'minlangan holda ko'riladi. Biz ham berilganlar modeliga BB nuqtayi

nazaridan qaraymiz. Yanada aniqroq aytak *objektning nomi, uning xossasi va xossalning qiymati* (berilgan) (2.1-rasm) nuqtai nazaridan qaraymiz. Berilganlar va ular orasidagi aloqalarga shu nuqtayi nazaridan qarash, berilganlarning turli modellarini yaratishga olib kelgan. Masalan, berilganlarning iyerarxik va tarmoq modellari, ularda berilganlarning elementlari tugunlardan, ular orasidagi aloqalar esa chiziqlar orqali isfodalanadi. Biz ular xususida mos ravishda 4- va 5- boblarda batafsil to'xtab o'tamiz.

Berilganlar orasidagi aloqalarni tavsiflashdagi yana bir o'ta kuchli usul, berilganlarni turli xususiyatlarga ajratilgan holatda, ulardagi o'zaro aloqalarni o'rganishdan iborat. Talablar darajalariga nisbatan bunday modellarni ikki turga ajratishadi: kuchli va kuchsiz turlar.

*Kuchli turdag'i modellarida*, hamma berilganlar qaysidir bir turkumga kiritilgan holatda ko'rildi. Agarda berilganlarni birorta turkumga kiritib bo'lmasa, uning sun'iy usullar yordamida birorta turkumga keltiriladi. Ayrim berilgan modellari esa, mumkin bo'lgan turkumlar avvaldan aniqlangan va ular dinamik ravishda o'zgarishi mumkin emas, degan nuqtai nazar asosida quriladi.

*Kuchsiz turdag'i modellar esa*, hech bir turkumlarga bog'lanmagan holatda ko'rildi. Turkumlardan foydalantish, har bir aniq vaziyatga nisbatan maqsadga muvofiq ravishda ko'rildi. Berilganlar o'z holicha yoki boshqa berilganlar bilan o'zaro bog'liq ravishda ko'rildi. Turkumlar boshqa berilganlar bilan bir qatordagi nuqtai nazar ostida qaraladi.

*Berilganlarning kuchli turdag'i modellarining* bir qator e'tiborli tomonlari mavjud. Ular berilganlarning abstrakt xususiyatlarni yaratishda turkumlar doirasidagi atamalarga bog'liq ravishda tadqiqot o'tkazish imkoniyatini beradi, ya'ni aniq berilganlarning xususiyatlari asosidagi nazariyani qurish imkoniyatini beradi. Turkumlardagi alohida xususiyatlarni berilganlardan voris sifatida o'tadi. Bundan tashqari, nomlarni takrorlanmaydi, o'xshash obyektlar va ularning xususiyatlari mos ravishda turkum nomi va turkum xususiyatining nomiga abstraktlashadi. Masalan, turkumning nomini XIZMATCHI, turkumning xususiyatining nomini «tug'ilgan sanasi» deb nomlash, har bir <XIZMATCHI, tug'ilgan sanasi, qiymati> uchligida xizmatchi nomini takrorlashdan qochiladi.

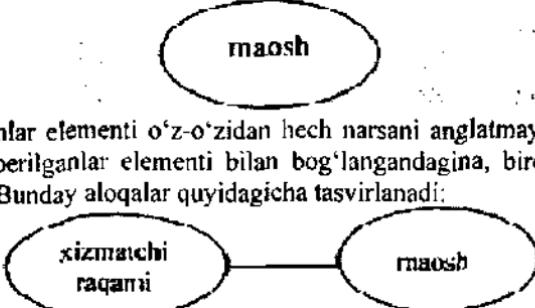
Berilganlarning kuchli turdag'i modellarining yana bir ustunligi ularning asosiy xususiyatlari bilan bevosita aniqlanadi, ya'ni hamma berilganlar qaysidir turkumga tegishli deb qaraladi. Bu usul yordamida, berilganlar o'tasidagi ziddiyatlardan qutilish mumkin, chunki ma'nosи yaqin berilganlar modelda bir-biriga yaqin deb qaraladi. Bu narsani har doim ham, kuchsiz turdag'i berilganlar modelida kuzatib bo'lmaydi. Berilganlarning bu dasili ulami umumiyl tuzilishida dalillarning turlicha

o'rinda bo'lishini ta'minlaydi. Agarda dalillar bir-biriga zid bo'lsa-yu, taqdimoti bo'yicha ham bir-biridan yiroq bo'lsa, berilganlar orasidagi ziddiyatlarni aniqlash juda murakkab.

Berilganlarning kuchli turdag'i modellaridan farqli ravishda, berilganlarning kuchsiz turdag'i modellari berilganlar va turkumlarning integratsiyasini ta'minlaydi. Ulardagi eng yuqori darajadagi imkoniyatlar predikatlarni hisoblash evaziga amalga oshiriladi. Berilganlarning ko'pgina modellarida predikatlarni hisoblash, modelning asosiy vositalari yordamida amalga oshirib bo'lunaydigan bilimlami taqdimoti uchun ishlataladi. Ushbu turdag'i modellarning ustun tonomi, berilganlarni bevosita ko'rish va predikatlar orqali yozuvlar bilan ishlash imkoniyatini yaratadi. Predikatlarni hisoblash maxsus tillar (masalan, QBE) berilganlar atributlarini ko'rib turgan holatda amalga oshiriladi va berilgan ustida bajarilgan natijalarni ham bevosita ko'rish imkoniyati mavjud. Bu BB tizimlarini yaratish yoki maxsus foydalanuvchilar uchun juda keng va qulay imkoniyatlarni ochib beradi.

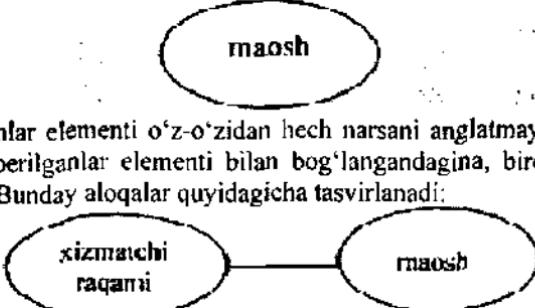
Berilganlarning relatsion modellarida ana shu kuchli va kuchsiz modellaridan birlgilikda uyg'unlashgan holatda foydalaniлади. Agarda berilganlarni kuchli modellaridan loyihalash bosqichida foydalansilsa, amaliy jihatdan esa berilganlarga bevosita ishlov berish jarayonida berilganlarning kuchsiz modellaridan foydalaniлади. Ikkala modellarning o'zaro integratsiyasi natijasida zamонавиј relatsion BBB tizimlari yaratilgan.

Berilganlar modellarida eng kichik bo'linmas berilgan, *berilganlar elementidan* iborat. Berilganlar elementini taqdim qilish uchun turli figuralardan foydalaniлади, shulardan biri ellips orqali biz berilganlar elementining turini modellashtirish jarayonida foydalananamiz. Masalan,



maosh

Berilganlar elementi o'z-o'zidan hech narsani anglatmaydi. U birorta boshqa bir berilganlar elementi bilan bog'langandagina, birorta ma'noga ega bo'ladi. Bunday aloqalar quyidagicha tasvirlanadi:



maosh

BB berilganlar elementlari va ular orasidagi aloqalardan iborat bo'ladi. Berilganlar bazasi turdag'i berilgan elementlaridan tuziladi, shuning uchun ularni tasvirlashda turli xildagi chizmalar ishlataladi. Yuqorida keltirilgan chizmada faqtgina berilganlar elementlarining mantiqiy chizmasi keltirilgan bo'lib, bunda berilganlarning kompyuterning tashqi xotirasida

qanday joylashgani xususida hech qanday ma'lumotlar berilmaydi. Masa-jan, Toshkent metrosining rasmiy chizmasida stansiyalar va ular orasidagi aloqalar tasvirlangan bo'lib, u stansiyalarning joylashishi va yer osti hamda yer ustidagi yo'llar qanday joylashgani xususida ma'lumotlar bermaydi. Biz stansiyalar orasidagi masofalar, yo'llarning qayerda burligani, qayerda yer ustidan o'tishi xususida ma'lumotlarni chizmadan ololmaymiz. Yo'llarda qanday o'zgarishlar bo'lsa, u mantiqiy chizmaga aslo ta'sir qilmaydi. Biz Toshkent metrosining chizmasini haqiqiy obyekting haqiqiy modeli sifatida qabul qilishiuniz mumkin. Berilganlar bazasidagi chizmalar kabi bu chizmalarni haqiqiy dunyodagi real obyektning akslanishi sifatida qabul qilishimiz mumkin.

### Chizmalar

Berilganlarning tuzilishini rasmiy tarzda tavsiflash zarur. Berilganlar bazasidagi berilganlarning tuzilishi xususidagi mantiqiy va fizik tavsiflaridan BBB tizimlarida foydalaniлади. Berilganlar bazasining umumiy mantiqiy tuzilishini *chizma* deb ataladi. Chizmani berilganlarning *unumiy modeli, konseptual model yoki konseptual chizma* deb atashadi. Chizma berilgan elementlarining turlarini ko'rsatadi. Chizma obyekt nomi, undagi atributlar va ular orasidagi aloqalarni ko'rsatadi. Chizmadagi tuzilishda berilganlar elementlarining qiymatlari kelishi mumkin. Masalan, aeropordagi tabloda samolyotlarning kelishi va ketishi xususidagi ma'lumotlar doimiy ravishda turadi. Lekin ularning qiymati vaqt bilan o'zgarib turadi. 2-bobdag'i 2.2-rasmni qiymatlarsiz tasavvur qilsak, uni chizma deb qarash mumkin. Agarda uni shu turishicha qabul qilsak, u *chizmaning nusxasiga* aylanadi. Demak, chizmaning nusxasida obyekt o'z qiymatlari bilan e'tirof etiladi.

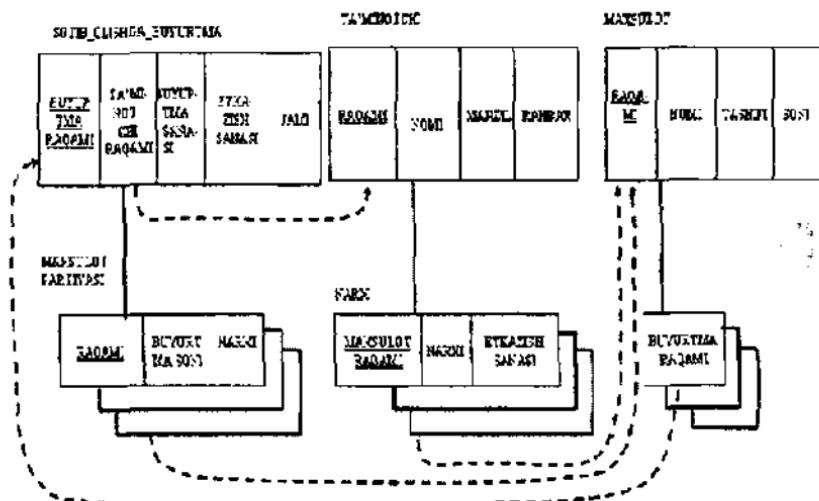
Yozuvlar bir necha turdag'i berilgan elementlarining ixtiyorij ketma-ketligidagi majmuasidan iborat. Berilganlar ketma-ketligi chizmaning nusxasi kabi o'z qiymatlariiga ega bo'lishi mumkin. Ularni *yozuvning nusxasi* deb atashadi. Yozuvlardagi har bir turning nomini atribut nomiga almash-tirib, ularni katakchalarda tasvirlasak, masalan, quyidagi TA'MINOTCHI yozuvining blokini hosil qilish mumkin. Tagiga chizilgan atribut shu yozuvning kalitini anglatadi.

Har bir yozuv Nta turli tuzilishga ega bo'lgan obyektlardan iborat. BB esa ana shunday turli yozuvlarning to'plami va ular orasidagi aloqalardan iborat. Chizmalarni tasvirlash usullari ham bir qancha. Shulardan biri 3.1 rasmda keltirilgan. Ushbu chizmada haqiqiy obyektlar, sotib olish tizimiga taalluqli yozuvlar orasidagi aloqalar keltirilgan.

**TA'MINOTCHI**

<b>RAQAMI</b>	<b>NOMI</b>	<b>MANZIL</b>	<b>RANBAR</b>
---------------	-------------	---------------	---------------

Bloklarni birlashtiruvchi uzluksiz chiziqlar bloklar orasidagi aloqalarni, uzlukli chiziqlar esa kesishma havolalarni ifodalaydi. SOTIB\_OLISHGA\_BUYURTMA yozuvi MAXSULOTLAR\_PARTIYASI yozuvi MAXSULOTLAR\_PARTIYASI yozuvi bilan, TA'MINOTCHI yozuvi NARX yozuvi bilan, MAXSULOT yozuvi esa BUYURTMA\_RAQAMI yozuvi bilan bog'langan.



3.1-rasm.

Mahsulot NOMI yoki mahsulot TASNIFI MAXSULOTLAR\_PARTIYASI yozuvida emas, balki alohidagi MAXSULOT yozuvining tarkibida kelgan. Xuddi shunday tarzda, ta'minotching NOMI va MANZILI SOTIB\_OLISHGA\_BUYURTMA yozuvida emas, balki TA'MINOTCHI yozuvining tarkibiga joylashtirilgan. Ushbu usul orgali MAXSULOTLAR\_PARTIYASI yozuvida, TA'MINOTCHI va mahsulot TASNIFI yozuvlarini takrorlashdan ozod bo'sinadi.

Chizmadagi aloqalar bizga berilganlar xususidagi qo'shimcha ma'lumotlarni, masalan, qaysi mahsulotlar qancha hajmda buyurilgan, degan

savolga javob bermaydi. SOTIB\_OLISHGA\_BUYURTMA yozuvi MAXSULOTLAR\_PARTIYASI yozuvlari orasidagi berilganlarning o'zaro aloqalari natijasida ma'lumotlar to'liq bo'ladi.

Xuddi shunday tarzda kesishma havolalar ham qo'shimcha ma'lumotlar bermaydi. SOTIB\_OLISHGA\_BUYURTMA yozuvi ta'minotchi nomini ham manzilini ham saqlamaydi. Lekin ta'minotchining raqamini saqlaganiga uchun TA'MINOTCHI yozuvidan, uning nomi va manzilini aniqlash mumkin. Uzlukli chiziqlarni chizmadan olib tashlagan bilan chizmaga hech qanday mantiqiy ta'sir qilmaydi. Kesishma havolalar orqali yozuvlar orasidagi qo'shimcha bog'lanishlarni kuzatish mumkin.

E'tibor bergan bo'lsangiz, chizmalar daliliy ravishda jadvailar orasidagi berilganlarning aloqalarini ifodalashini ko'rdingiz. Berilganlar elementlari orasidagi aloqalar turlicha bo'lishi mumkin. Berilganlar elementlari orasidagi aloqalardan oldin ba'zi berilgan modeli bilan bog'liqlikdagi ayrim tushunchalarni berib o'tamiz.

### Berilganlar elementlarining o'zaro aloqasi

*Aloqa* – konseptual darajadagi munosabatlар orasidagi bog'liqlikni nazarda tutadi. Masalan, «Xaridorlar oziq-ovqatlarni sotib olmoqdalar» degan tasdiq, «Xaridor» va «Oziq-ovqat» munosabatlari orasida aloqa borligini bildiradi va bunday munosabatlар shu aloqaning ishtirokchilarini bildiradi.

Ikkita berilganlar elementlari o'rtasida bog'lanishlarning bir necha usuli mavjud:

- «birga - bir» (1:1);
- «birga - ko'p» (1:M);
- «ko'pga - bir» (M:1);
- «ko'pga - ko'p» (M:M).

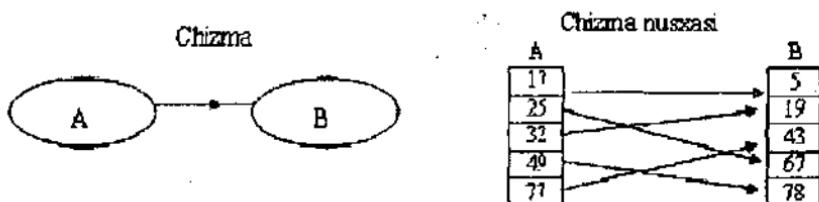
Agar A va V berilganlar elementlari bo'lsa, ular orasidagi quyidagi aloqani o'matishimiz mumkin, «birga – bir» (1:1) aloqa. A va B berilganlar elementlari orasida 1:1 aloqa o'matilgan deyiladi, agarda har bir daqiqada A berilganning qiymatiga B berilganning faqatgina bitta qiymati mos kelsa. Bunday aloqalarni milli chiziqlar bilan belgilaymiz (3.2- rasm).



3.2-rasm.

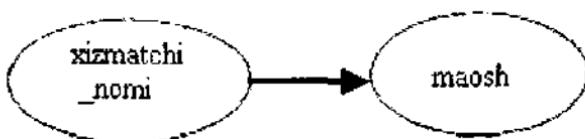
Bu A va Bni identifikatsiyalaydi, degan ma'noni beradi. Agarda A ning qiymati ma'lum bo'lsa, B ning qiymatini bilish mumkin. Matematik manтиқда [86] ham bittalik mil  $A \rightarrow B$  munosabat, A B ni identifikatsiyalaydi, degan ma'noni bildiradi.

Agarda A va Bni ikkita alohida yozuv deb qarasak, unda quyidagi 3.3-rasmdagi aloqalar o'rинli misol bo'la oladi.



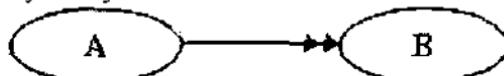
3.3-rasm.

Masalan, «xizmatchi nomi» va «maosh» elementlari orasidagi aloqani olsak. Bu aloqada bitta xizmatchiga faqat bitta maosh elementi mos kelishini ko'rishiuz mumkin (3.4-rasm).



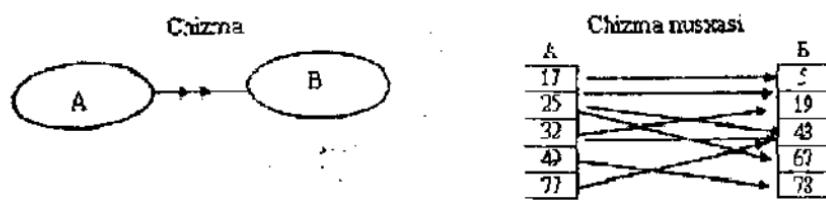
3.4-rasm.

**«Birga - ko'p» (1:M) aloqa deganda, A elementning bitta qiyamatiga B obyektning bir nechta qiyamti mos kelishi tushuniladi. Bunday aloqalarni ikkita milli chiziqlar bilan belgilaymiz (3.5-rasm). Bunday aloqalarda A Bni identifikatsiyalamaydi.**



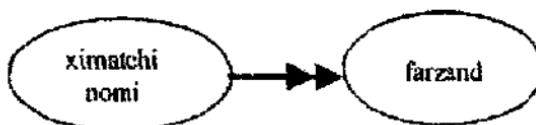
3.5-rasm.

Agarda A va Bni ikkita alohida yozuv deb qarasak, ular chizmaning nusxasidagi elementlarning o'zaro aloqasiga quyidagi 3.6-rasmida keltirilgan misol bo'ladi.



3.6-rasm.

Masalan, «birga - ko'p» aloqani «xizmatchi nomi» va «farzand» elementti orasidagi aloqada ko'rish mumkin (3.7-rasm). Har bir xizmatchiga nol va «farzand» obyektining nol va undan ko'p qiymati mos kelishi mumkin.



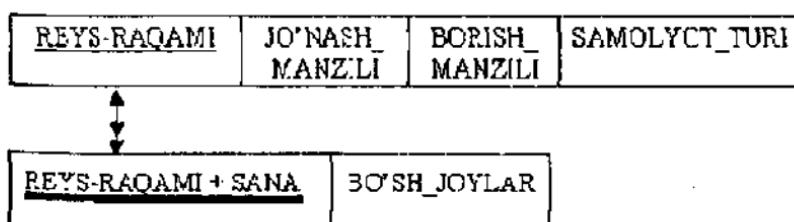
3.7-rasm.

Berilganlar elementlari orasidagi aloqalar ikkala yo'nalishda bo'lishi mumkin. Shuning uchun, «ko'pga-bir» ( $M:1$ ) aloqa va «ko'pga – ko'p» ( $M : M$ ) aloqalar ham mavjud.

Berilganlar elementlarining Nta turi orasidagi aloqalar soni  $N(N-1)$  iborat.

#### **Yozuvlar orasidagi aloqalar**

Berilganlar elementlari orasidagi aloqalar turlicha bo'lgani uchun, yozuvlarni bog'lovchi milli chiziqlar 1:1, 1:M, M:1 va M:M turlarida bo'lishi mumkin. Masalan:

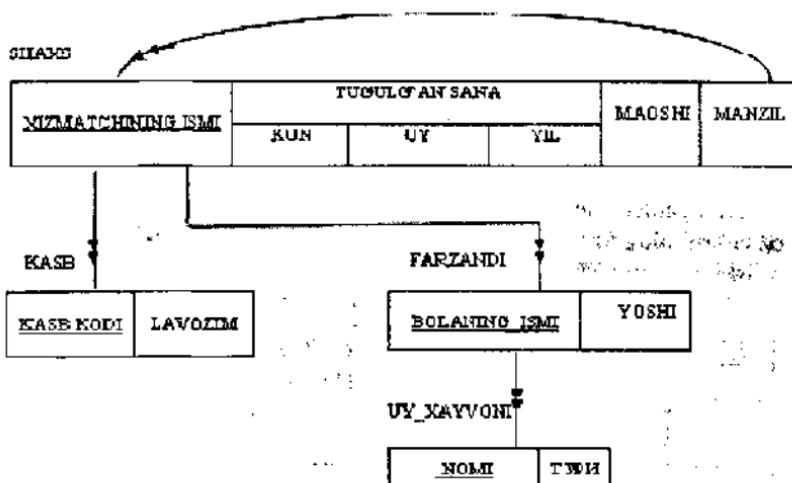


3.8-rasm.

Yozuvlar o'zaro kalitlar orqali bog'lanadi. Keltirilgan chizmada REYS-RAQAMI birlamchi kalit bo'lib, u REYS-RAQAMI + SANA birlashgan kalit bilan bog'langan va shu orqali ikkala yozuvlar orasida 1:M aloqasi o'matilgan. Berilganlar elementlarini yozuvlar orqali ifodalashda chizmadagi aloqlarning soni ancha kamayadi. Lekin hamma aloqlarni chizma orqali baribir ko'rsatib bo'lmaydi. BBB tizimlari qanday bo'lishidan qat'i nazar, chizmalar quyidagi qoidalarga rioya qilishi shart:

- har bir yozuv sodda tuzilishga, ayrim yozuvlar esa murakkab kalitlarga ega bo'lishi mumkin;
- chizmadagi yozuvlar va berilganlar elementlari o'z nomlariga ega bo'lishi;
- yozuvlardagi hech bir berilganlar elementlari va yozuv bir xil nomga ega bo'lmasligi;
- yozuvlar orasidagi aloqalar milli chiziqlar orqali ko'rsatilishi va 1:1, 1:M, M:1, M:M turlardan birida bo'lishi;
- yozuvlardagi birlamchi kalitlar alohida belgilanishi;
- yozuvlardagi boshqa kalitlarning ham aloqasi ko'rsatilishi;
- chizmalar berilganlarning xususiyatlarini ifodalagan uchun, ular iloji boricha turg'un bo'lishi.

Keltirilgan qoidalar asosidagi chizmaga misol 3.9 rasmida keltirilgan. Bunda yozuvlar orasidagi teskari aloqalar keltirilmagan, chunki ulardan foydalanilmaydi.

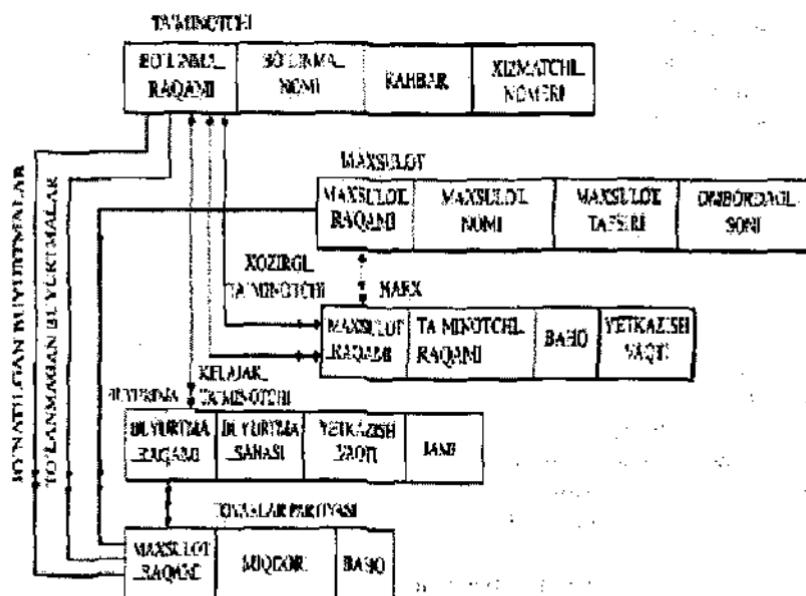


3.9-rasm.

3.10-rasmida yozuvlar orasidagi aloqalar murakkab bo'lgani uchun chizmarining aniq bo'lishi muhim ahamiyatga ega. Bunda beshta yozuvlar orasidagi sakkizta aloqa keltirilgan. Ushbu rasm 3.1-rasmga o'xshab ketadi, lekin bu chizmadagi aloqalar ancha murakkab.

Sotib olishga buyurtmalar ta'minotchilarga yetkaziladi. BUYURTMA va TOVARLAR PARTIYASI yozuvlari birlgilikda sotib olish buyurtmasini tashkil qiladi. Har bir buyurtma o'z nomeri (raqami)ga ega bir nechta mahsulotga bo'lishi mumkin. TA'MINOTCHIdan TOVARLAR\_PARTIYASiga ikkita «birga - ko'p» turidagi aloqa mavjud, bittasi «jo'natilmagan buyurtmalar» uchun, ikkinchisi «to'lanmagan buyurtmalar» uchun. Xuddi shunday ikkita «birga - ko'p» turidagi aloqa TA'MINOTCHI va NARX orasidagi yozuvlarda ham mavjud, bittasi «hozirgi ta'minotchi» uchun, ikkinchisi «bo'lajak ta'minotchi» uchun. Bu farning hammasi ikki tomonlama aloqaga ega. TA'MINOTCHI va BUYURTMA yozuvlari orasida ham ikki tomonlama aloqalar mavjud.

3.10-rasmidagi chizmadan biz berilganlarning modellariga nisbatan keyingi bobrlarda turli maqsadlarda foydalananamiz.



## Xulosa

Berilganlarning modeli muammolari eng boshlang'ich izlanishlardan boshlab nazariy tadqiqotlarning eng markazida bo'lib kelgan. Chunki tadqiqotchilar, modelning rivojlanishi turli predmet sohalarga tegishli bo'lgan berilganlarning taqdimotini BBB tizimlarini yaratilishidagi maqsadga:

- mos ravishda akslantiridi;
- o'zaro moslik darajasini aniqlaydi;
- texnologik maqsadlarga erishiladi,

deb hisoblaganlar.

Shuning uchun ham, berilganlarning turli modellari yaratilgan va haqiqotan ham, ana shu modellardagi turli nazariy asoslar, modellashdirish uchun yaratilgan turli maxsus muhitlarning yadrosiga aylangan.

Hozirgi kundagi BBB tizimlarini berilganlar modelida yaratilgan nazariy asoslersiz tasavvur etib bo'lmaydi.

## Nazorat uchun savollar

1. Berilganlar modelini ta'riflang.
2. Berilganlar modelini nechta turi mavjud.
3. Kuchli turdag'i berilganlar modelini ta'riflang.
4. Kuchsiz turdag'i berilganlar modelini ta'riflang.
5. Berilganlar modellarining kuchli va kuchsiz turlari orasidagi farqlarni ta'riflang.
6. Berilganlar elementlari orasidagi aloqalarga 3 ta misol keltiring va har birining chizmasini chizing.
7. Berilganlar bazasining umumiy mantiqiy tuzilishi nima deb ataladi?
8. Chizmaning nusxasi deganda nimani tushunasiz?
9. Yozuv va uning nusxasini ta'riflang.
10. Yozuv va uning nusxasiga misol keltiring.
11. Berilganlar bazasi va yozuv orasidagi bog'lanishga chizma ko'tinishida misol keltiring.
12. Blok deganda nimani tushunasiz?
13. Blokka ikkita misol keltiring.
14. Berilganlar elementlari orasidagi o'zaro bog'lanishlar necha xildan iborat?
15. «Birga-bir» aloqani ta'riflang va unga misol keltiring.
16. «Birga-ko'p» aloqani ta'riflang va unga misol keltiring.
17. «Ko'pga-bir» aloqani ta'riflang va unga misol keltiring.
18. «Ko'pga-ko'p» aloqani ta'riflang va unga misol keltiring.

19. Yozuvlar nima orqali bog'lanadi?
20. Chizmalarning rioya qilishi shart bo'lgan qoidalarining har birini tavsiflang.
21. 3.10-rasmdagi chizmani tavsiflang.

## **II BO'LIM. BERILGANLARNING MODELLARI**

### **4-bob. BERILGANLARNING IYERARXIK MODELI**

**Tayanch iboralar:** iyerarxik *tuzilish*, iyerarxik *model*, *tarmoq modeli*, *relatsion model*, *daraxt*, *tugun*, *boshlang'ich tugun* (*yaratuvchi*), *yaratilgan element*, *baglar*, *ildiz*, *muvozanatlashgan daraxtlar*, *muvozanatlashmagan daraxtlar*, *ikkilik daraxt*, *iyerarxik fayllar*, *bir jinsli tuzilishlar*, *chizmaning nusxasi*.

O'tgan asrning 60- yillarda oyga kosmik kemani uchirish bo'yicha AQSHda loyiha tayyorlangan. Ushbu loyihani amalga oshirish uchun o'sha davrda juda katta hajdagi berilganlarga ishlov berish yoki boshqarish uchun hech qanday tizim mavjud emas edi. Natijada, loyiha bo'yicha ish olib borayotgan North American Aviation (NAA) kompaniyasi, (hozir Rockwell International deb ataladi) GUAM (Generalized Update Access Method) nomi bilan programma ta'minoti yaratdilar. GUAM da qo'llanilgan asosiy g'oya kichik elementlardan katta tarkiblarga «pastdan-yuqoriga» chiqib, cho'qqida yagona loyihani jamlashdan iborat bo'lган. Ushbu teskari daraxtni eslatuvchi tuzilishni, *iyerarxik tuzilish* (hierarchical structure) deb ataydilar. O'tgan asrning 60-yillarining o'ttalarida IBM korporatsiyasi NAA kompaniyasiga qo'shilib, GUAM loyihasi bo'yicha hamkorlikda ish olib bordilar. Natijada, 1968-yilda jahonda birinchi marta iyerarxik tuzilish asosida ishlaydigan IMS (Information Management System) tizimi yaratildi. IBM korporatsiyasi IMS tizimining funksional imkoniyatlarini cheklab, faqat yozuvlar iyerarxiyasini boshqarish vazifasini qoldirdi, chunki saqlash qurilmalariga faqatgina ketma-ket kirish imkoniyati mavjud edi xolos. Asosan magnit lentalar tashqi xotira sifatida ishlatalar edi. Ma'lum vaqt o'tgandan keyin bu cheklash olib tashlandi. Keyinchalik iyerarxik tuzilishlarni berilganlarning *iyerarxik modeli* deb atashgan. IMS tijorat maqsadida yaratilgan birinchi BGBT bo'lishiga qaramasdan, hozirgi kungacha iyerarxik BGBT sifatida ko'pchilik katta meynfreymlarda ishlatalmoqda.

Berilganlar bazasini tavsiflashda uch xil yondashuv yoki uchta model mavjudligi ta'kidlanadi: *iyerarxik* (daraxtsimon tuzilish nazarda tutiladi), *tarmoq* (ko'pincha CODASYL tarmoq tuzilishi nazarda tutiladi) va *relatsion* (2 o'lchovli jadval asosida). CODASYL (Conference on Data Systems Languages – Berilganlarga Ishlov Tizimlaridagi Til Conference)

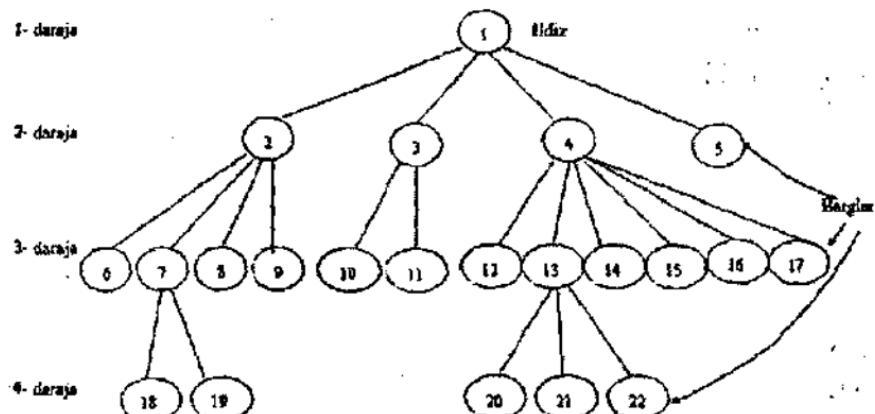
siyasi) [90] BB sohasidagi butun jahon tan olgan juda obro'li conferensiya. BB bo'yicha ishlatalidigan atamalarning deyarli hammasi ana shu conferensiya tomonidan kiritilgan.

БИР ЎЧИНГИ АДАМ

## Daraxtlar

Daraxt *tugunlar* deb ataladigan, elementlarning iyerarxik tuzilishidan iborat (4.1-rasm). Iyerarxiyaning eng yuqori darajasida faqatgina bitta tugun – ildiz bo'ladi. Izdizdan boshqa har bir tugun, o'zidan yuqoridagi darajada joylashgan, shu tugun uchun *boshlang'ich tugun* (*yaratuvchi*) deb ataladigan, boshqa tugun bilan bog'langan. Hech qaysi element bittadan ortiq boshlang'ich tugun bilan bog'lanmagan. Har bir element o'zidan quyi darajada joylashgan bitta yoki bir nechta element bilan bog'lanishi mumkin. Bu elementlar *yaratilgan elementlar* deyiladi. Shoxning uchida joylashgan, lekin yaratgan elementi bo'limganlar, *barglar* deyiladi.

4.1-rasmida 1- element ildiz, 5,6,8-12 va 14-22- elementlar *barglar*. Daraxt odatda teskari holatda, ildizi tepada va barglari pastda bo'lган shaklda tasvirlanadi. Ushbu rasmida ko'rsatilgan daraxtlarga o'xshash daraxtlar, berilganlarni mantiqiy va fizik tavsiflashda ishlataladi.



4.1-rasm. Daraxt (har bir element bittadan ortiq boshlang'ich elementga ega emas).

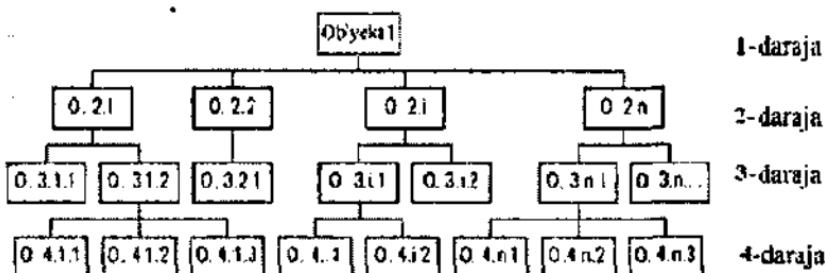
Daraxtlarga turlicha ta'riflar berilgan. Shularidan biri quyidagi daraxtning rekursiv ta'rifi. **Daraxt** – bu «cheklangan D to'plamidan iborat bo'lib, bir va undan ortiq tugunlardan tashkil topadi hamda

1) shu daraxtning ildiz deb ataladigan, bitta maxsus belgilangan tugundan,

2) ildizdan tashqari, qolgan tugunlar ketma-ket juftliklari kesishmaydigan  $D_1, D_2, \dots, D_m$  ( $m \geq 0$ ) to'plamlarida joylashadi va bularning har biri o'z navbatida daraxt bo'ladi.  $D_1, D_2, \dots, D_m$  daraxtlar shu daraxtning bo'laklari deb ataladi».

Keltirilgan tushunchalarni obyektlar sifatida ko'raylik (4.2-rasm). Bu modelda bitta asosiy obyekt bo'lib, qolgan barcha obyektlar unga bo'ysunadi va bu obyektlar iyerarxiyaning turli darajalarida joylashadi. Obyektlarning o'zaro aloqasini bir ildizli obyektdan iborat iyerarxik daraxt tashkil etadi.

Iyerarxik BB bir xildagi daraxtlarning bir nechta nusxalarining tartiblangan to'plamidan iborat bo'ladi. Ajodolar va vorislar orasidagi havolalarning butunligi avtomatik ravishda bajariladi. Asosiy qoida – birorta voris o'zining ota yoki onasisiz mavjud bo'la olmaydi (4.2-rasminga qarang).



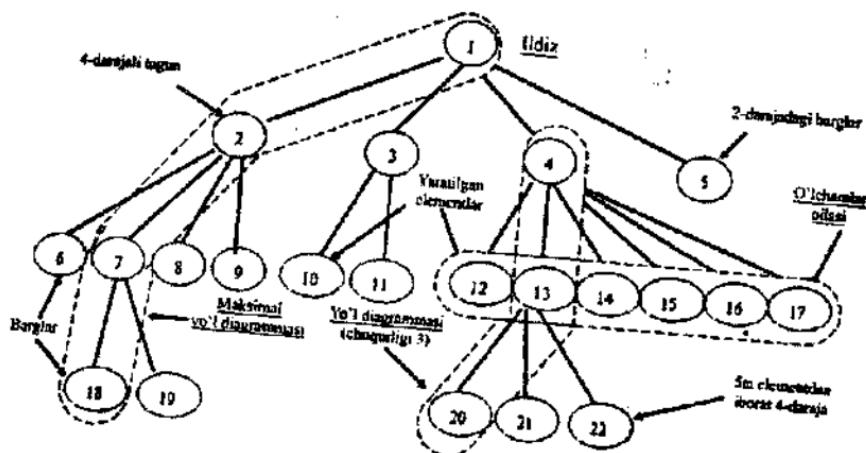
4.2-rasm. Berilganlarning iyerarxik modeli.

4.3-rasmida daraxt va uni tavsiflashda ishlataladigan atamalar keltirilgan. Daraxt diagrammasining balandligi 4 (darajalar soni), moment 22 (tugunlar soni), og'irligi 16 (barglar soni), asosi 1 (ildizlar soni).

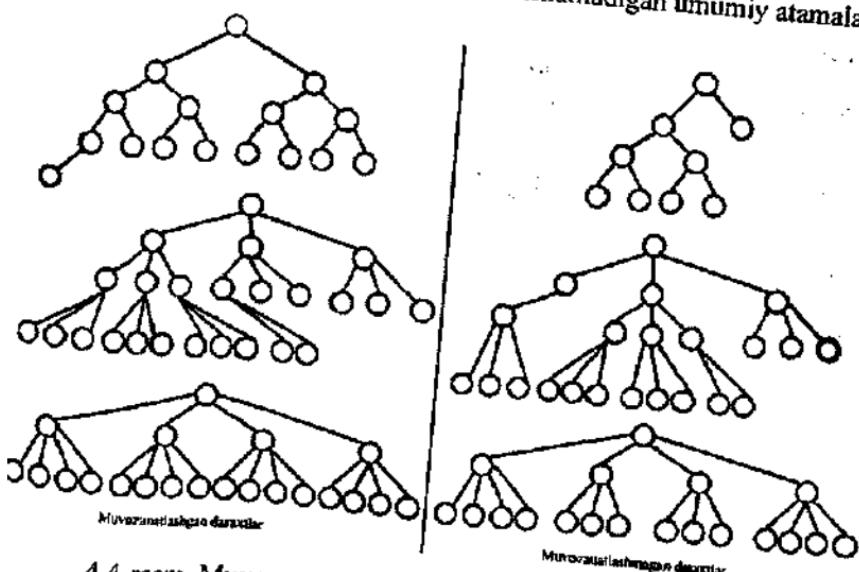
### Muvozanatlashgan va muvozanatlashmagan daraxtlar

Ba'zi hollarda *muvozanatlashgan daraxtlar* tushunchasi ishlataladi. Bunday daraxtlarda har bir tugun bir xil sondagi barglarga ega bo'lib, daraxtdagi tugunlarga yangi barglarni kiritish tepadan pastga, daraxtdagi har bir darajada esa chapdan o'ngga tomon amalga oshiriladi. 4.4-rasmida muvozanatlashgan va muvozanatlashmagan daraxtlar keltirilgan. Muvozanatlashgan daraxtlarni berilganlar sifatida fizik jihatdan kompyuterda

joylashtirish juda sodda amalga oshiriladi, chunki undagi barglar soni fiksirlangan, muvozanatlashmagan daraxtlarda esa o'zgaruvchan. Lekin berilganlarning mantiqiy tuzilishiда aksincha, ko'pincha muvozanatlashmagan daraxtlar sifatida tasvirlash talab qilinadi, buning asosiy sababi «haqiqiy dunyo»dagi obyektlar asosini muvozanatlashmagan holatga mos bo'lishidadir.

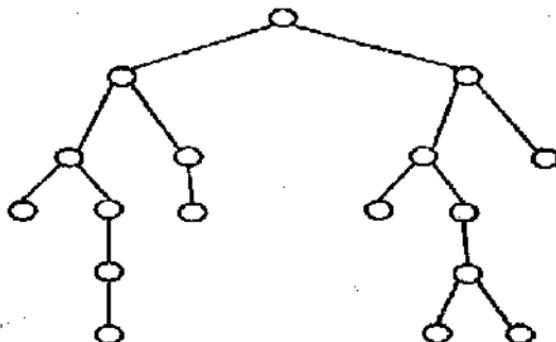


4.3-rasm. Daraxtlarni tavsiflashda ishlataladigan umumiy atamalar.



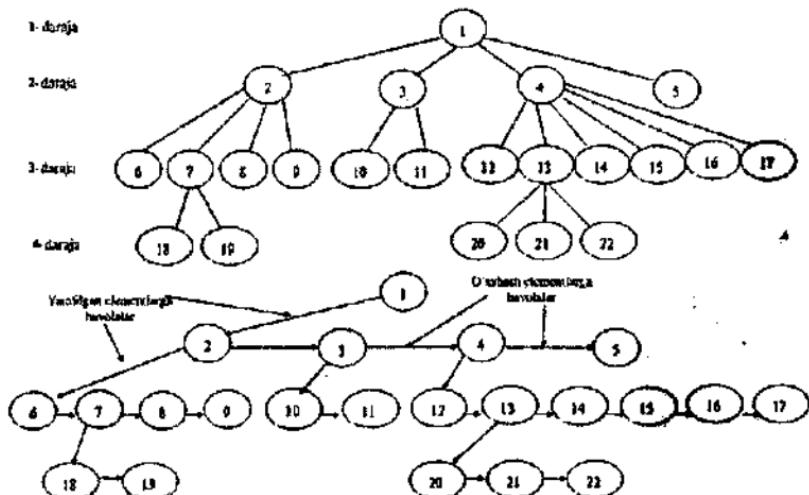
4.4-rasm. Muvozanatlashgan va muvozanatlashmagan daraxtlar.

Muvozanatlashgan daraxtlar tuzilishida bitta alohida maxsus kategoriya mavjud bo'lib, unda bitta tugunda ikkitadan ortiq barg bo'lishi mumkin emas. Bunday tuzilish *ikkilik daraxt* deyiladi. 4.5-rasmda muvozanatlashmagan ikkilik daraxt keltirilgan.

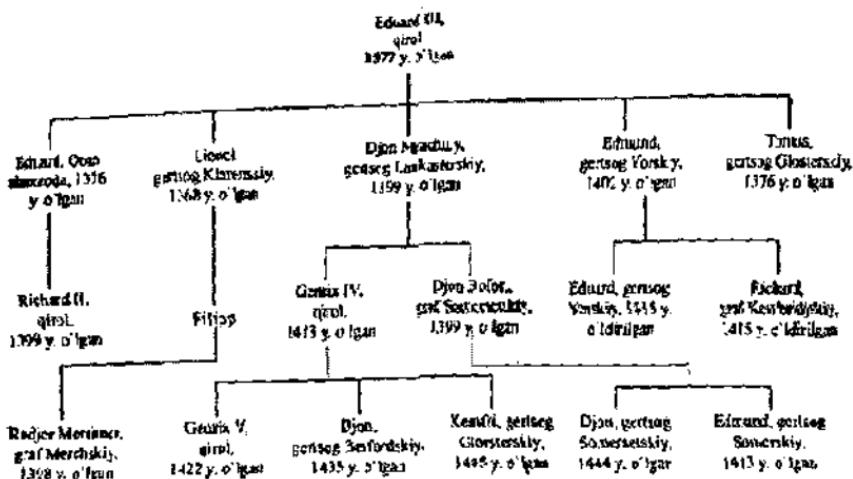


4.5-rasm. Muvozanatlashmagan ikkilik daraxt.

Daraxtdagi ixtiyoriy aloqalarni ikkilik daraxtsimon tuzilishlar sifatida keltirish mumkin (4.6-rasm). Har bir element yaratilgan va o'xshash elementlarga havolasi bo'lishi mumkin. Lekin juda kam hollarda berilganchalarining mantiqiy tuzilishini bevosita ikkilik daraxtlar sifatida keltirish mumkin.



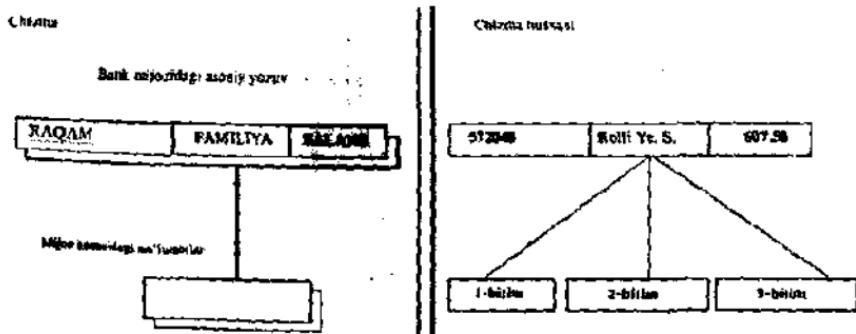
4.6-rasm. Har bir element yaratilgan va o'xshash elementlarga havola qilishi mumkin bo'lgan, ikkilik daraxtning tuzilishi.



4.7-rasm. Genealogik daraxt.

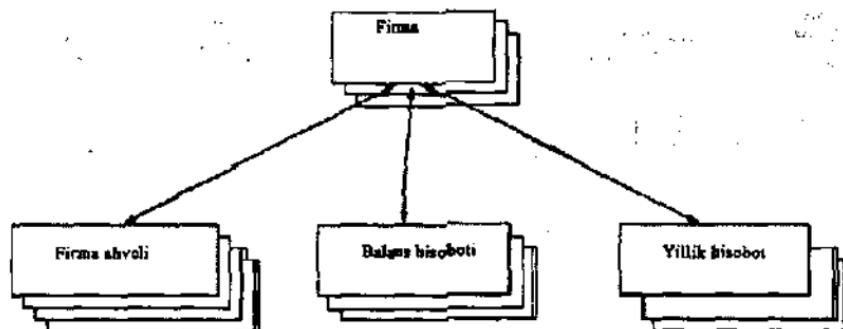
4.5 – 4.7 rasmlarda daraxtsimon tuzilishlar keltirilgan bo‘lib, tugunlar sifatida berilganlarning elementlari yoki yozuvlari akslangan. 4.7 rasmdagi genealogik daraxtning misol sifatida keltirilishiga sabab, boshlang‘ich element sifatida faqatgina (ota) bitta element olingan. Agarda ikkita element, ota va ona olinganida, bu berilganlarning murakkab tuzilishiga misol bo‘lar edi.

Berilganlarning daraxtsimon tuzilishlardan foydalanishning asosiy sababi, yaratilgan elementdan boshlang‘ich elementga oddiy o‘tish mumkinligi (ya’ni yaratilgan element uchun faqat bitta boshlang‘ich element mavjud) va teskari o‘tish murakkabsigini (ya’ni bittadan ko‘plikka) anglatadi, masalan 4.8-rasmdagi kabi. Ushbu rasmda chizma va uni oddiy ikki darajali daraxti tuzilishi sifatida amalga oshirishning nusxasi keltirilgan.



4.8-rasm. Faqatgina ikki turdag'i yozuvlardan iborat fayl.

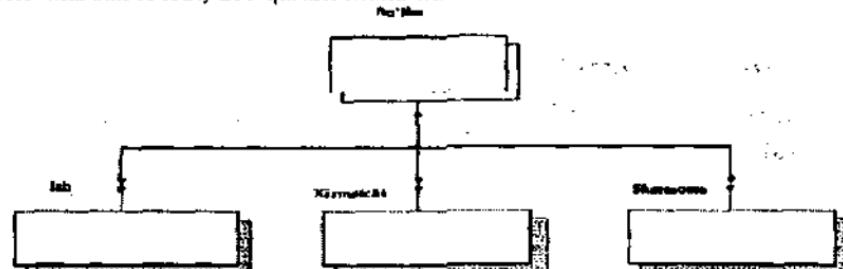
Ba'zida berilganlarning elementlari orasidagi ikki tomonlama aloqalarni 4.9-rasmidagi kabi sodda akslantrish mumkin. Bunda ikkilik tuzilish orqali, bitta obyektga tegishli, lekin har xil joy (baza)larda saqlanadigan yozuvlar keltirilgan.



4.9-rasm. Berilganlarning elementlari orasidagi ikki tomonlama aloqalar.

### Terma va iyerarxik fayl

CODASYL assotsiatsiyasining BB bo'yicha ishchi guruhi, aloqalarning mustahkamligini ko'rsatish uchun *terma* atamasidan foydalangan [90]. *Terma* – bu ikkinchi darajali daraxtlardagi yozuvlar. Bir xil turdag'i termalarning fayllari ikkinchi darajali iyerarxik fayllardan iborat. *Iyerarxik fayllar* deganda, yozuvlari daraxtsimon tuzilish sifatidagi bog'lanish tushuniladi. 4.9 va 4.10-rasmrlarda *termalar* keltirilgan. Har bir terma o'z nomiga ega. Ko'p darajali iyerarxik faylni bir qancha termalarning majmuasi sifatida qarash mumkin. 4.12-rasmni uchta, to'rtta yoki beshta termalardan iborat, deb qarash mumkin.

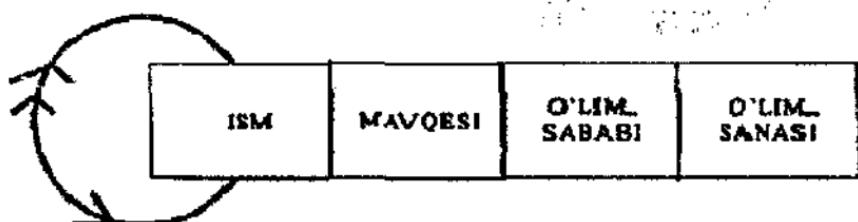


4.10-rasm. CODASYL majmuasi – 2- darajali daraxt yozuvlari.

## Bir jinsli tuzilishlar

4.7-rasmida keltirilgan genealogik daraxt tuzilishi jihatidan 4.8-rasmidagi tuzilishdan anchagini farq qiladi. Genealogik daraxtning har bir tuguni bir turdag'i yozuvlarda ifodalanishi mumkin. 4.12-rasmida esa har bir tugun turli xildagi yozuvlar bilan ifodalangan. Hamma BBBT tuzilish jihatidan bir jinsli bo'lmagan, lekin yozuv turi fiksirlangan bazalar bilan ishlaydi. Bir jinsli va bir jinsli bo'lmagan tuzilishlarga ajratish tarmoq tuzilishidagi berilganlar uchun ham muhim ahamiyatga ega.

Agarda biz 4.7-rasmda keltirilgan genealogik daraxt tuzilishiga uning mohiyatini akslantirish maqsadida yondashsak, 4.11-rasmida keltirilgan tuzilishga, ya'ni *chizmaga* ega bo'lamiz. 4.7-rasm 4.11-rasmda keltirilgan *chizmaning nusxasidan* iborat. BBni umumiyl mantiqiy tuzilishini ifodlashni, biz yuqorida *chizma* deb atagan edik. *Chizma o'zgarmaydi, lekin undog'i atributlar o'zgaradi*. Agarda chizma 4.7 rasmdagi kabi qiymatlar bilan keltirilgan bo'lsa, u *chizmaning nusxasi* deyiladi. *Chizmaning nusxasi* atamasini programmist berilganlarni tavsiflash va qiymat berish jarayonida ishlataladi. Bitta chizmadan bir nechta turli xildagi nusxalarni hosil qilish mumkin.



4.11-rasm. Daraxt tuzilishidagi halqa.

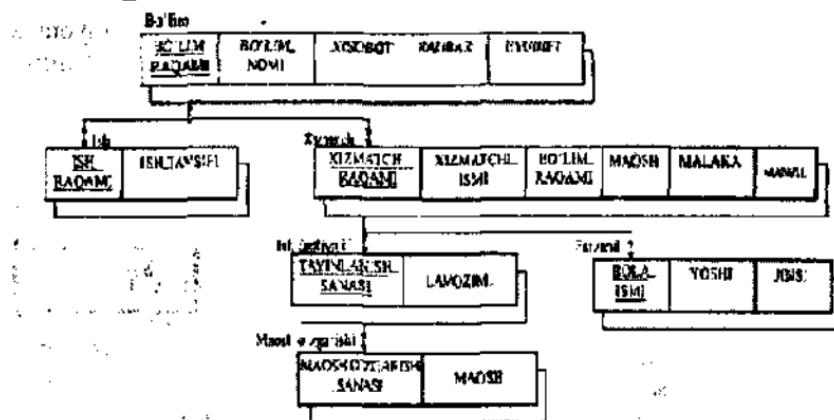
## Bog'lanishlardagi bog'liqliklar

Daraxtning quyi darajasidagi boshlang'ich yozuvlarning qiymati aniqlanmagan bo'lsa, yozuvlar to'liq aniqlangan deyilmaydi. Daraxtlar uchun yozuvlarning nomlarini kichik harflarda, qo'shtirnoq ichida kursivda yozamiz. Masalan 4.12-rasmida «*Ish faoliyati*» yozushi o'z holicha ma'noga ega emas. U nimanidir anglatishi uchun o'zining ajdodi «*Xizmatchi*» yozushi bilan bog'langan bo'lishi kerak. LAVOZIMI elementi TAYINLANISH SANASI kalitli maydon elementi bilangina identifikatsiya qilin-maydi. U XIZMATCHI\_RAQAMI va TAYINLANISH SANASI bog'langan kalitlar

orgali identifikasiya qilinadi. Demak, ushbu daraxtdagi «Xizmatchi» va «Ish faoliyati» yozuvlari orasidagi bog'lanishda bog'liqlik mavjud, deyish mumkin. Lekin boshqa tomondan, «Xizmatchi» yozuvi o'zining ajdodiga tobe emas, chunki XIZMATCHI\_RAQAMI kaliti uni to'liq identifikatsiyalaydi.

«Maosh o'zgarishi» yozuvi ham o'zining bevosita ajdodiga bog'liq emas, lekin yuqoriroq darajadagi ajdodi «Xizmatchi» yozuviga bog'liq. Chunki, XIZMATCHI\_RAQAMI va MAOSH O'ZGARISH SANASI bog'-langan kalitlar orqaligina MAOSH elementini to'g'ri identifikasiya qila olamiz.

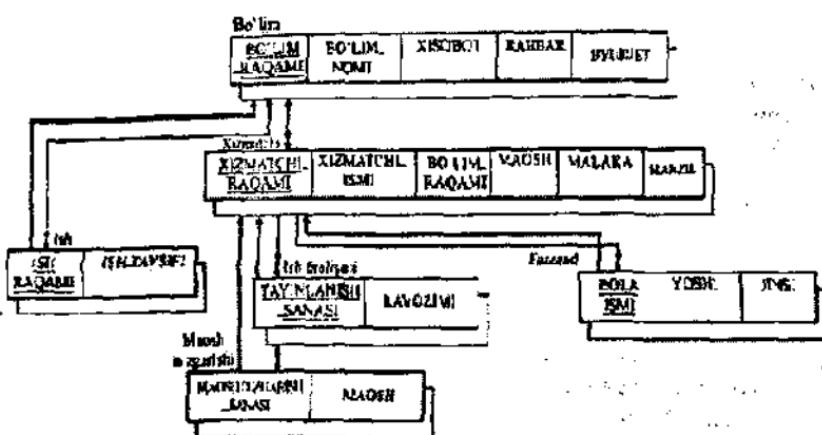
4.12-rasm bizga «Ish» yozuvi o'zining kaliti ISH\_RAQAMI orqali qanchalik to'liq identifikatsiyalanishini ko'rsata olmagan. Shunday bo'lishi mumkinki, korxonada nomerlash bo'yicha yagona tartib bo'lмаган holda, har bir bo'lim o'zi bajaradigan har bir ish bo'yicha o'zining nomerlash tartibiga ega bo'lishi mumkin. Ana shu holda, «Ish» yozuvi «Bo'lim» yozuviga tobe bo'lib qoladi. ISH\_RAQAMI va ISH\_RAQAMI birlashgan kaliti ISH\_TAVSIFI elementini to'liq identifikatsiyalaydi.



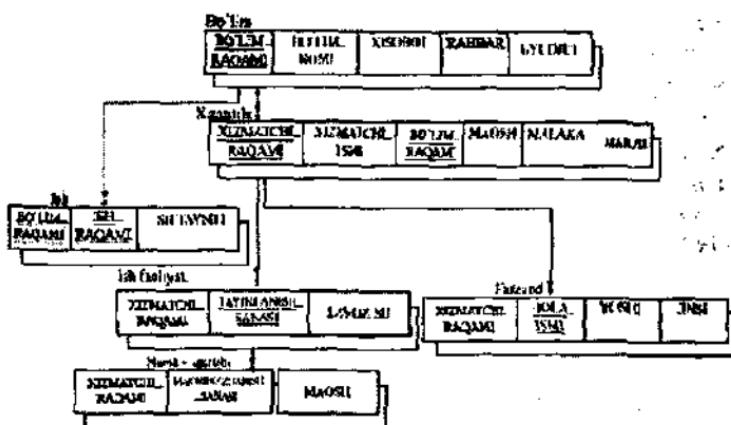
4.12-rasm. 4- darajali iyerarxik berilganlar chizmasi.

4.13-rasmida yuqorida tahtil qilingan chizmada keltirilgan yozuvlar orasidagi bog'lanishlar mili yuqoriga yo'naltirilgan yo'g'on chiziqlar orqali tasvirlangan.

Tobe yozuvlarga qo'shimcha kalitlarni kiritish, yo'nalishlardagi ana shunday bog'liqliklarni yo'qotishga va o'zini-o'zi to'liq identifikatsiyalashga olib keladi. 4.14-rasmida ana shunday kalitlar «Ish», «Ish faoliyati», «Maosh o'zgarishi» va «Farzand» yozuvlariga qo'shilgan elementlardan (semiz qora shriftlarda) iborat.



4.13-rasm.



4.14-rasm.

### Xulosa

Iyerarxik modellar daraxt tuzilishidagi diagrammalar shaklida tasvirlanadi. Daraxting elementlari munosabatlarga, aloqalar esa munosabatlar orasidagi bog'lanishlarga mos keladi. Munosabatlardagi aloqa turlari qo'yilgan asosiy chegara, ana shu aloqalarning tartiblangan bo'lishini talab qiladi, boshqacha aytganda daraxt uchlarining bir-biriga nisbatan (chap yoki o'ng tomonda) turishi ahamiyatli hisoblanadi. Bundan tashqari,

*funktional aloqalarga mos keluvchi chiziqlar, har doim daraxt ildizidan barglarga yo'naltirilgan bo'ladi. Funktional aloqaga mos keluvchi chiziqlar ajdod→avlod aloqasini ifodalaydi.*

*Eng asosiy kamchiligi ajdod aloqalarni olib tashlash mumkin, bu esa avlod berilganlarni yo'qolishiga olib keladi.*

### **Nazorat uchun savollar**

1. Berilganlarning iyerarxik modelini ta'riflang va misol keltiring.
2. Yaratuvchi va yaratilgan tugunlarni, barglarni va ildizni ta'riflang.
3. Iyerarxik modeldagi daraja nimani anglatadi?
4. Muvozanatlashgan daraxtni ta'riflang va unga misol keltiring.
5. Muvozanatlashmagan daraxtni ta'riflang va unga misol keltiring.
6. Maksimal yo'l diagrammasi nimani bildiradi?
7. Yo'l diagrammasi chuqurligini ifodalang.
8. 3 ta elementdan iborat 6-darajali iyerarxik modelning diagrammasiga misol keltiring.
9. Ikkilik daraxt deganda nimani tushunasiz?
10. Iyerarxik model nusxasiga misol keltiring.
11. Iyerarxik fayl deganda nimani tushunasiz?
12. CODASYL assotsiatsiyasi terma deb, nimani atagan?
13. Bir jinsli tuzilish deganda nimani tushunasiz?
14. Daraxt modelidagi halqaga misol keltiring.
15. 3-darajali iyerarxik haqiqiy modelga misol keltiring.
16. Iyerarxik modeldagi yozuvlar orasidagi tobe yozuvlarni chizmada ifodalang.

---

## 5-bob. BERILGANLARNING TARMOQ MODELI

**Tayanch iboralar:** CODASYL, DDL, DML, tarmoq, murakkab tarmoq tuzilishlari, oddiy tarmoq tuzilishlari, kesishishdagi berilganlar, M:M turidagi aloqalar, takrorlash, halqa.

1960-yillarning o'rtalaridagi yutuqlardan bittasi General Electric kompaniyasi yaratgan IDS (Integrated Data Store) tizimi bo'ldi. Ushbu tadqiqot ishlarining rahbari BBBT sohasidagi birinchi tadqiqotchilardan biri Charlz Bachman (Charles Bachmann) edi. IDS tizimining rivojlanishi natijasida esa BB boshqarish tizimlarining yangi *tarmoq* (network) turi yaratildi. O'sha davrdagi axborot tizimlarining rivojlanishiga bu yetarli ta'sir ko'rsatdi. Tarmoq BBBBBT berilganlar orasidagi iyeraxik tuzilishga nisbatan murakkab bo'lgan aloqalarni tasvirlash hamda berilganlar bazalarining standartini shakllantirish uchun yaratilgan edi. 1965-yilda shunday standartlarni yaratish uchun CODASYL (Conference on Data Systems Languages) konferensiya uyuشتirdi. Unda List Processing Task Force ishchi guruh tuzildi, bu guruh 1967-yilda Data Base Task Group (DBTG)ga aylandi. DBTG guruhining vazifasi, BB va berilganlarni boshqarishni yaratishdagi muhit identifikatorlarini aniqlashni shakllantirishdan iborat edi. Dastlabki hisobot natijalari 1969-yilda, to'liq naqli esa 1971-yilda e'lon qilindi. DBTG guruhining *uchta tarkibdan* iborat takliflari:

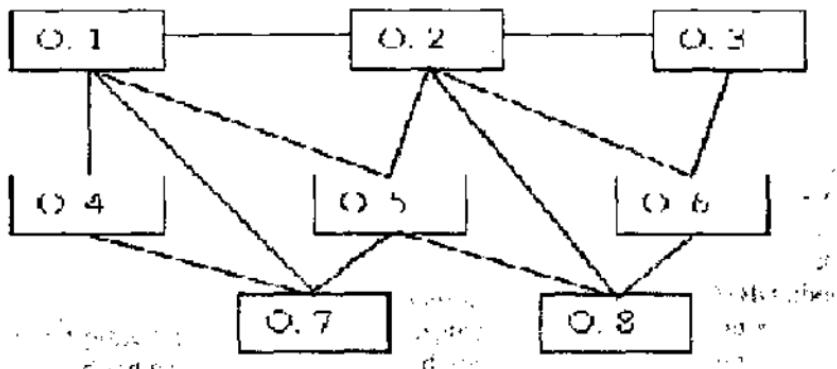
- *tarmoq chizmasi* — berilganlar bazasining to'liq mantiq tuzilishi, unga BB nomining ta'risi, har bir yozuvning turi va har bir turdag'i yozuvning tarkibidan iborat;
- *qism-chizma* — *foyдалувчи yoki ilovalar tasavvuridagi, berilganlar bazasining qismi*;
- *berilganlarni boshqarish tili* — berilganlarning tavsifi va tuzilishini aniqlaydigan hamda boshqaradigan vosita.

**DBTG** guruhи quyidagi *uchta tilni standartlashtirishni* taklif qildi:

- chizmadagi berilganlarni aniqlash tili **DDL** (Data Definition Language), berilganlar bazasining administratori berilganlarni ushbu til orqali tavsiflash imkoniyatiga ega.

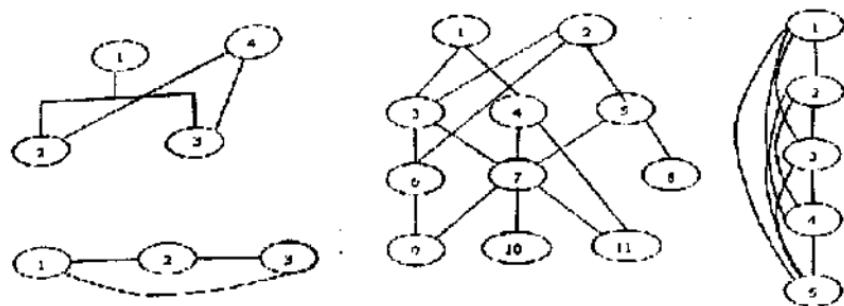
- qism-chizmadagi berilganlarni aniqlash tili **DDL**, ilovalarda berilganlar bazasining kirish zarur bo'lgan qismalarini aniqlash imkoniyatini beradi;
- berilganlarni manipulatsiyalash tili **DML** (Data Manipulation Language), berilganlarni boshqarishga mo'ljallangan.

Agarda berilganlar aloqasida yaratilgan element bittadan ortiq boshlang'ich (yaratuvchi) elementga ega bo'lsa, bunday aloqalarni daraxt-simon yoki iyerarxik tuzilish sifatida tavsiflab bo'lmaydi. Boshqacha aytganda, bir obyekt iyerarxik modelda bitta avlodga ega bo'lsa, tarmoq modelida bitta obyekt bir necha avlodga mansub bo'lishi mumkin. Berilganlarning tarmoq modelida ixtiyoriy obyekt bir payning o'zida ham yetaklovchi, ham yetaklanuvchi bo'lishi mumkin va boshqa obyektlar bilan ixtiyoriy sondagi o'zarlo aloqalarni tuzishda ishtiroy etishi mumkin. Tarmoq tuzilishidagi berilganlar, yozuvlar va ular orasidagi aloqalarning termasidan, ya'ni chizmada keltirilgan yozuvlar turlarining termasi har bir turdag'i berilganlar nusxalarining termasidan va berilgan har bir aloqa turlari termasining nusxalaridan iborat (5.1-rasm).



5.1-rasm. Tarmoq modeli chizmasi.

Keltirilgan 5.2-rasmning birinchesida har bir yaratilgan element o'zining ikkita yaratuvchisiga ega. Ikkinci misolda aloqaiarning yo'nalishi ko'rsatilmagan. Uchinchi misolda, qaysi bir tugun eng pastki bo'lmasin, uni yaratuvchisi ikkita bo'ladi. To'rtinchi misolda, 5-element to'rtta yaratuvchisiga ega. Ba'zi tarmoq tuzilishlarini tabiiy holda, darajalari to'g'risida aytish kerak. Misolda (5.2-rasm) keltirilgan tarmoq tuzilishlar mos ravishda ikki, uch, to'rt va beshta darajaga ega. Bir yoki bir nechta tugunlar bittadan ortiq boshlang'ich tugunga ega.



5.2-rasm. Tarmoq tuzilishiga misollar.

### **Oddiy va murakkab tarmoq tuzilishlari**

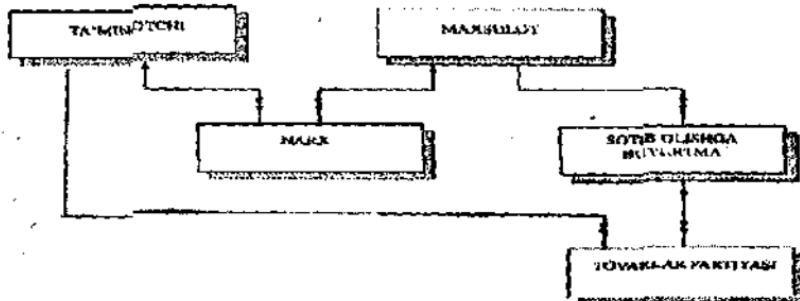
Yozuvlarning turlari orasidagi yoki berilganlar agregatlarining turlari orasidagi aloqani ko'rsatuvchi ko'pgina tarmoq tuzilishlarida, yaratuvchi va yaratilgan elementlar munosabati daraxtdagi munosabat kabi aniqlanadi, ya'ni yaratuvchi – yaratilgan murakkab, yaratilgan – yaratuvchi oddiy munosabatdan iborat.

5.3-rasmda bir jinsli bo'limgan besh turdag'i yozuvlardan iborat tarmoq tuzilishi keltirilgan. Yozuvlarning turlarini birlashtiruvchi birorta chiziq ikkala yo'nalishda ikkilangan milga ega emas. Har bir munosabat yaratuvchi-yaratilgan kabi qaralishi mumkin. SOTIB OLISHGA BUYURTMA yozuvi MAXSULOT yozuviga nisbatan yaratilgan, TOVARLAR PARTIYASI yozuviga nisbatan esa, yaratuvchi bo'ladi.

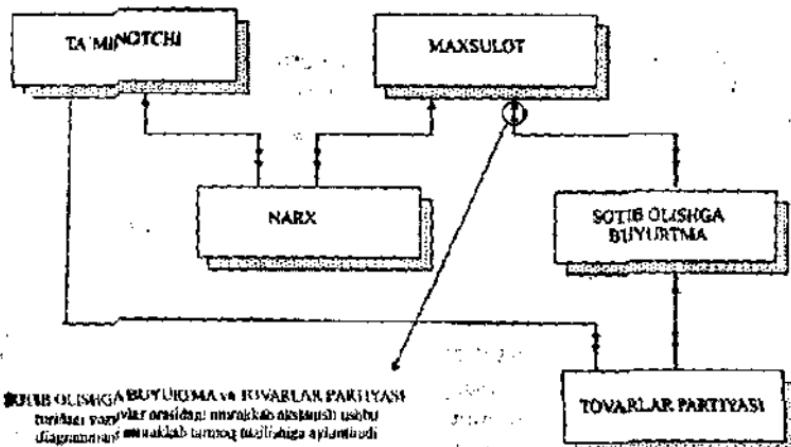
Yaratilgan-yaratuvchi munosabatlari bilan hosil bo'lgan oddiy yoki ishlatalmaydigan tuzilishlarni, ikkala yo'nalishda murakkab bo'lgan qaysidir ikkita berilganlar turlari orasidagi munosabatlardan ajrata bilish kerak. Ikkinci turdag'i tuzilishlarda chizmadaqib birorta chiziqdagi ikki tomonni ko'rsatuvchi ikkilangan millar bo'ladi. Bu turdag'i chizmani *murakkab tarmoq tuzilishlari*, agarda hech qaysi chiziq ikkita qarama-qarshi yo'nalishda ikkilik millarga ega bo'lmasa, ularni *oddiy tarmoq tuzilishlari* deb ataymiz. 5.3-rasmda oddiy tarmoq tuzilishi keltirilgan.

5.3-rasmidagi chizmani murakkab tarmoq tuzilishiga aylantirish mumkin, buning uchun SOTIB OLISHGA BUYURTMA - MAXSULOT munosabatidan foydalanish kerak, chunki bitta buyurtma bir nechta mahsulotga berilishi mumkin. 5.4-rasmda xuddi shu chizma murakkab tarmoq tuzilishi sifatida keltirilgan.

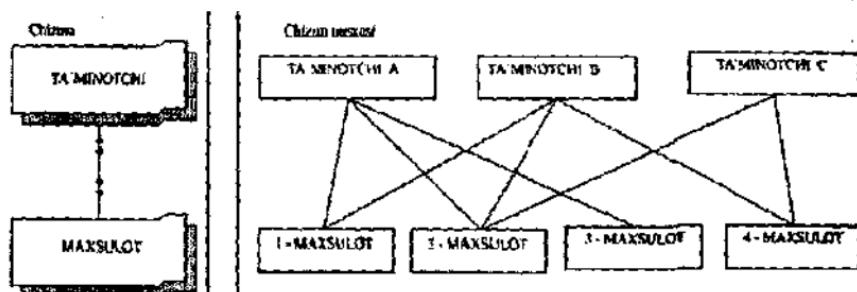
Ba'zi BBBT oddiy tarmoq tuzilishi chizmalari bilan ishlashi, lekin murakkab tarmoq tuzilishi chizmalari bilan esa ishlay olmaydi. Lekin murakkab tarmoq tuzilishini oddiy tarmoq tuzilishi chizmalariga keltirish mumkin.



5.3-rasm. Xarid tizimida ishlataladigan, beshta yozuvdan iborat tarmoq tuzilishi.



5.4-rasm. Berilganlarning murakkab tarmoq modeli.



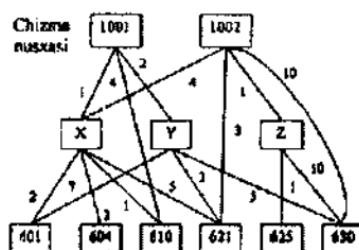
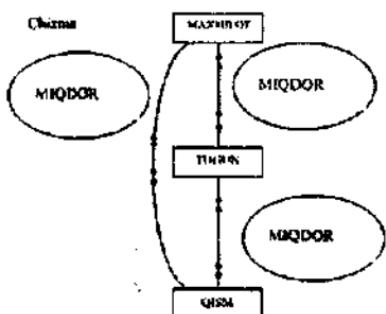
5.5-rasm. Faqat ikki turdag'i yozuvjardan iborat murakkab tarmoq tuzilishi.

Murakkab tarmoq tuzilishini yaratish uchun ikkita turdag'i yozuvlarning bo'lishi yetarli. 5.5-rasmidagi TA'MINOTCHI yozuvi bir nechta yaratilgan yozuvlarga ega bo'lishi mumkin, chunki ta'minotchi bir nechta turdag'i mahsulotni yetkazishi mumkin. MAHSULOT yozuvi ham bir nechta boshlang'ich yozuvlarga ega bo'lishi mumkin, chunki aynan shunday mahsulotni turli ta'minotchilar yetkazib berishi mumkin. Yagona munosabat yozuvlar orasidagi ikki tomonlama murakkab aloqani ta'minlaydi.

### **Kesishishdagi berilganlar**

Ayrim holatlarda berilganlarning bitta elementi berilganlarning bir nechta boshqa elementlari bilan bog'langan bo'lishi mumkin. Misol uchun, bitta mahsulot turli ta'minotchilar tomonidan turli narxlarda yetkazib berilishi mumkin. NARX berilganlar elementi, faqatgina MAXSULOT yoki faqatgina TA'MINOTCHI yozuvlari bilan birikishi mumkin emas, u bir yo'la ikkala yozuv bilan bog'lanishi kerak. Berilganlarning bunday bog'lanishi, ya'ni berilganlarning bir qancha yozuvlar bilan birikishini, *kesishishdagi berilganlar* deyishadi.

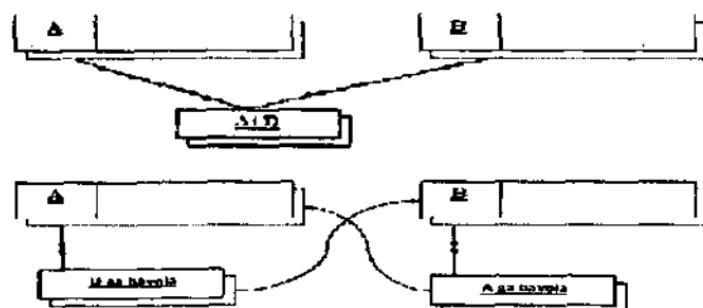
5.6-rasmida kesishishdagi berilganlarga murakkab misol keltirilgan. Korxonada ishlab chiqariladigan mahsulot tugunlar va qismlardan iborat. Korxonaning berilganlar bazasida MAXSULOT, TUGUN va QISM yozuvlari bor. Bu yozuvlar turli bog'lanishlarda turlicha bo'ladi. Ularni 5.6-rasmida ko'rsatilgan usulda bog'lash mumkin. Har bir aloqadagi son, yo tugundagi yoki mahsulotdagi qismlar sonini, yo mahsulotdagi tugunlar sonini bildiradi. Masalan, 1001 mahsulot bitta X, ikkita Y tugundan va to'rtta 610- qisdan tashkil topgan. Umumiy holda, 5.6-rasmida keltirilgan tuzilishga o'xshashi, materiallar ro'yxati yoki mahsulot identifikatsiyalanishini ko'rsatadi va mahsulotning tarkibini ishlab chiqarish maqsadlarida ifodalaydi.



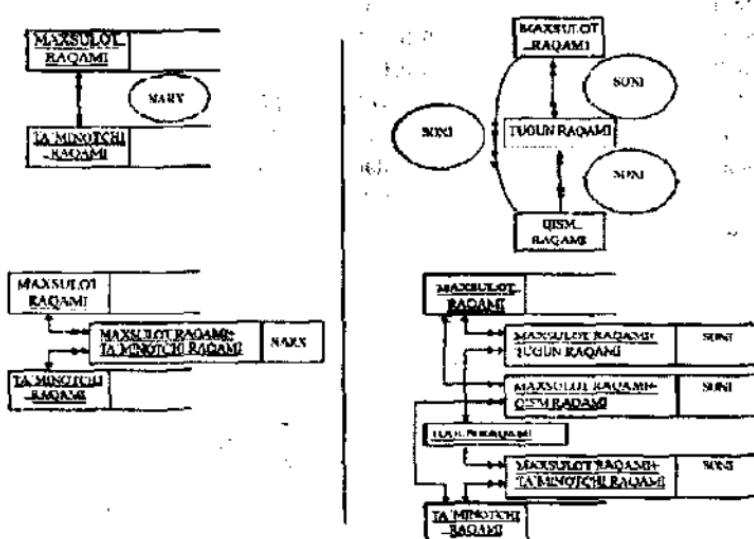
**5.6-rasm. Materiallar ro'yxati bazasi bo'yicha chizma va chizma nusxasi.**

## M:M turidagi aloqalar

M:M turidagi (qarama-qarshi yo'nalishdagi chiziqlarda ikkilangan millar orqali tasvirlangan aloqalar, ya'ni «ko'plikdan-ko'plik»ka turidagi aloqalar) yozuvlar orasidagi aloqaga tegishli, ikkita muammo mavjud. Birinchisi, ularning fizik tasvirlanishi bilan bog'liq. 5.7-rasmida yozuvlardagi kalitlari A va B bo'lgan, M:M turidagi aloqalarni qo'llashga doir ikkita usul keltirilgan. Ikkala holatda ham kesishishdagi berilganlarning haqiqiy kaliti AQB bo'ladi.



5.7-rasm. A va B kalitli yozuvlar orasida M : M turidagi akslantirishlarni amalga oshirishdagi ikkita usul.



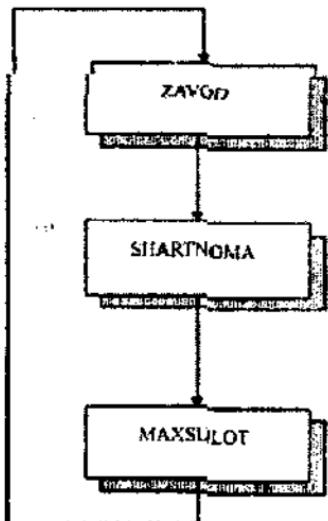
5.8-rasm. Berilganlarning kesishishiga ikkita misol.

*Ikkinci muammo shundan iboratki, M:M turidagi aloqalar mavjud bo'lganda, albatta kesishishdagi berilganlar ham bo'ladi. Agarda boshlang'ich bosqichda bular bo'lnasa ham, keyinchalik berilganlar bazasining kengayishi natijasida paydo bo'ladi. Agarda kesishishdagi berilganlar A va B kaittlarga ega yozuvlar bilan bog'liqligi tushunilsa, ular AQB bog'langan kalit orqali aynan belgilanadi.*

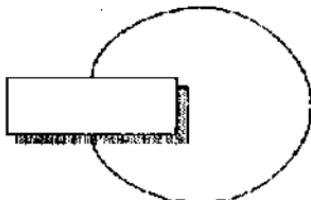
**5.8-rasmda kesishishdagi berilganlarga ikkita misol va ularni mumkin bo'lgan tasvirlash usuli keltirilgan.**

### Takrorlashlar

Ayrim tuzilishlar sikllarni o'z ichiga oladi. *Takrorlash* deganda, tuguning oldingi elementi xuddi shu vaqtida davomchi ham bo'lgan vaziyat tushuniladi. Yaratuvchi-yaratilgan munosabatlari bu vaziyatda yopiq konturni tashkil qiladi. Keltirilgan chizmaning diagrammasida (5.9-rasm) bunday kontur ikkilangan millardan tuzilgan. Masalan, zavod turli mahsulotlarni ishlab chiqaradi. Ayrim mahsulotlar pudratni birga bajaruvchi zavodlarda ishlab chiqariladi. Bitta shartnomaga bir qancha mahsulotni ishlab chiqarish bog'liq bo'lishi mumkin. Ana shu munosabatlarni tasvirlash takrorlashni tashkil etadi (5.9-rasm). Yetarlicha murakkab bo'lgan tarmoq tizimlari ba'zan juda ko'p sikllardan tashkil topadi. Lekin BBBT hammasi ham takrorlashlar bilan ishlash imkoniyatiga ega emas.



5.9-rasm. Takrorlash.



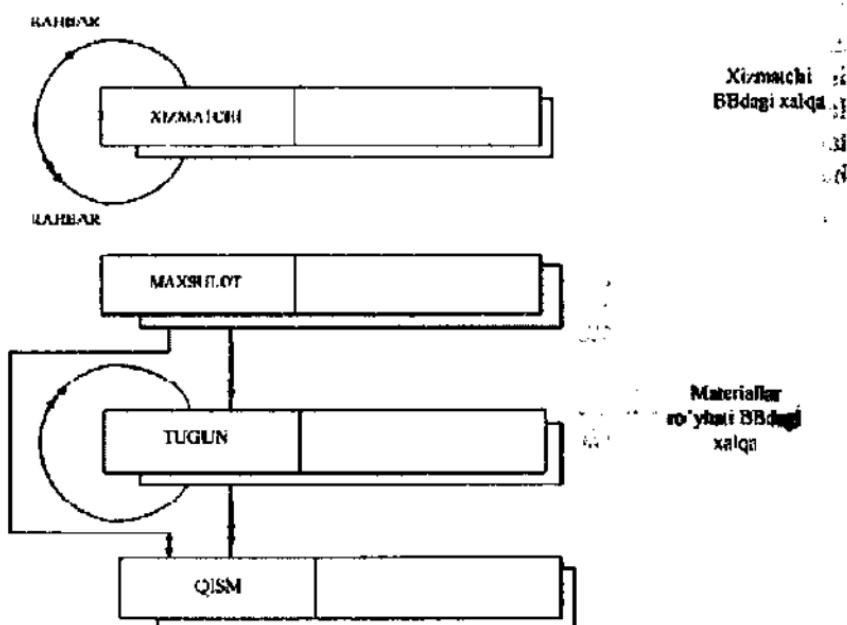
5.10-rasm. Halqa.

## Halqa

Ba'zan fayldagi yozuvlar xuddi shu fayldagi boshqa yozuvlar bilan bog'langan bo'ladi. Bunday vaziyat *halqa* deb ataladi. Halqada faqat bir tur-dagi yozuv bo'ladi, ya'ni yaratilgan yozuvning turi yaratuvchi yozuvning turi bilan bir xilda bo'ladi. 5.10-rasmida halqa misol keltirilgan.

5.11-rasmda halqalarni ishlatish mumkin bo'lган, yetarli darajada keng tarqalgan ikkita vaziyat keltirilgan. Xizmatchi faylida, ba'zi xizmatchilar orasidagi aloqalarni identifikatori keltirilgan. Materiallar ro'yxati berilganlar bazasida (5.11-rasm) qo'shimcha marrakkablik kiritilgan, ayrim tugunlarning o'zi tugunlardan iborat.

Hamma programma tizimlar ham berilganlarning xalqa turini qo'llay olmaydi. 5.11-rasmdagi ikkinchi halqa M:M turidagi aloqalarni tasvirlaydi, chunki materiallar ro'yxati berilganlar bazasining foydalanuvchilarini faqat «tugun nimadan tashkil topgan?» degan savol emas, balki «tugun qayerda ishlatalidi?» degan savol ham qiziqtiради.



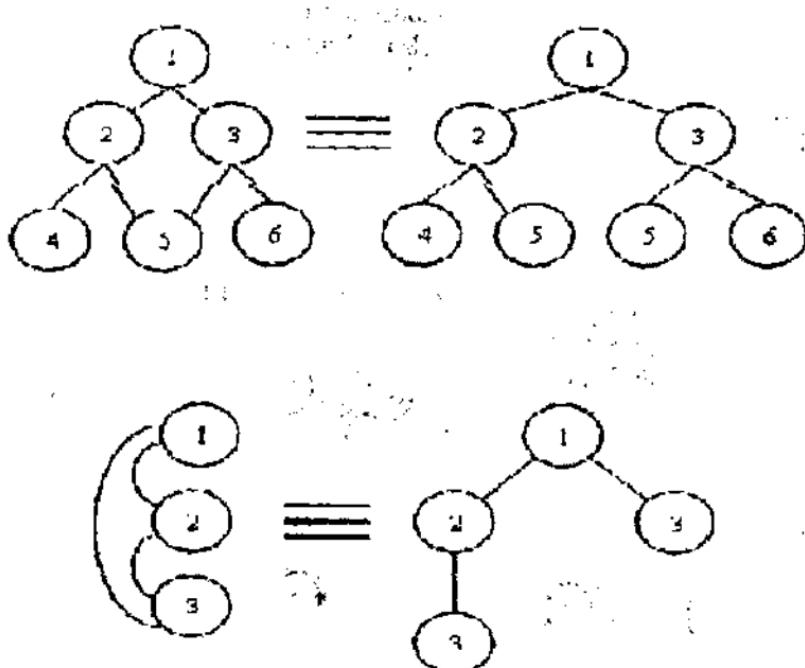
5.11-rasm. Xalqaga misollar.

5.12-rasmning chap qismida hayvonot bog'ining berilganlar bazasining chizmasi keltirilgan, unga hayvonlar orasidagi aloqalarni tasvirlovchi ikkita

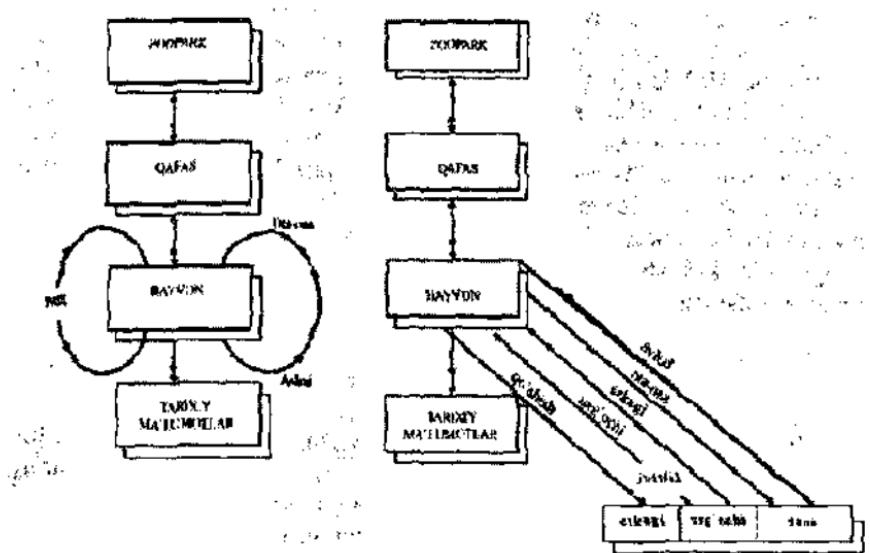
halqa kiritilgan. Bunday berilganlar bazasining zarurligi, hayvonlarning paydo bo'lish va qo'shilish jarayonlari hamda bo'lajak juftlarni tashashga yordam berish bilan bog'liq. Hayvonlar ko'pgina juftliklarni tashkil etishi mumkin, shuning uchun JUFTLIK halqasi M:M turidagi aloqalardan iborat bo'ladi. Hayvonlarning ota-onasi va ko'pgina avlodlari bo'lishi mumkin, shuning uchun OTA-ONA-AVLODLAR halqasi ham M:M aloqalardan tashkil topadi. Ba'zida qo'shish sanasini ham saqlash maqsadga muvoziq bo'ladi. Bu holda kesishishdagi berilganlar paydo bo'ladi va elementlarning yangi turini kiritish talab qilinadi. 5.12-rasmda halqadan foydalanilmaslik uchun yordamchi guruh kiritilgan.

### Tarmoq tuzilishlarni sodda ko'rinishga keltirish

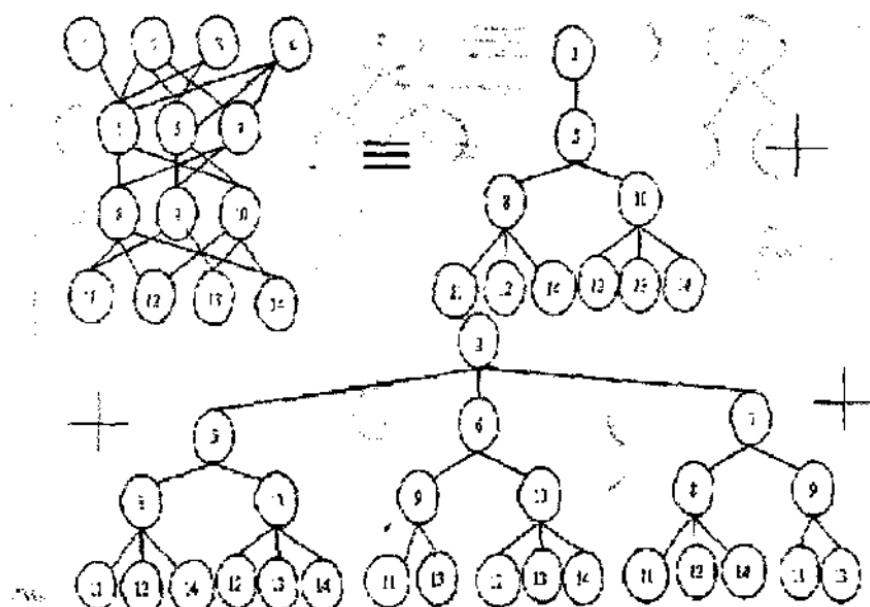
Ixtiyoriy tarmoq tuzilishi ortiqchalik orqali, sodda ko'rinishga keltirishi mumkin. 5.13 va 5.14-rasmarda oddiy tarmoq tuzilishlарини ekvivalent tarzdagи daraxtsimon tuzilishlarga keltirishga uchta misol keltirilgan. Ayrим holatlarda, paydo bo'ladigan ortiqchalik kichikdir va bu, mumkin boshqa holatlarda juda katta bo'ladi.

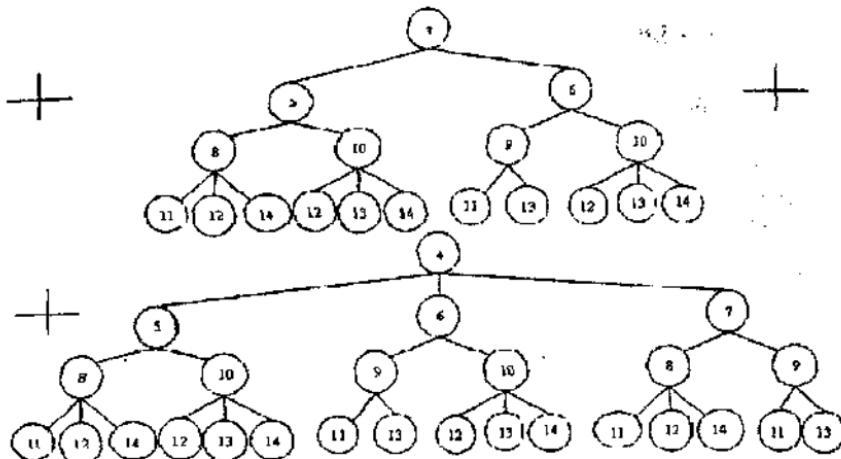


5.12-rasm. Halqani yo'qotishga misol.



5.13-rasm. Oddiy tarmoq modelini daraxt yoki ortiqcha elementli daraxtlar toplami shakliga o'tkazish.





5.14-rasm. Oddiy tarmoq modelini ortiqcha elementli daraxtlar to'plami shakliga o'tkazish.

Berilganlarning tarmoq turidagi modellarini iyerarkik ekvivalent modellarga keltirish uchun, birinchi darajadagi ajdod elementlarning har birini alohida o'ziga tegishli avlod elementlar majmuasi sifatida ifodalash kerak.

Cullinet Software, Inc. Kompaniyasining Integrated Database Management System (IDMS) tizimi tarmoq turidagi tuzilishlarga misol bo'ladı.

### Xulosa

*Yozuvlar va aloqalar tarmoq modelining asosini tashkil etadi. Yozuvlarning turlari munosabatlarning turlarini jadval ko'rinishida tasvirlashda ishlataladi. Aloqalar esa, tabiiy holda aloqalarning turlarini tasvirlash uchun ishlataladi. Aloqalar yordamida yozuvlar orasidagi bog'lanishlar tasniflanadi. Aloqalar albatta funksional bog'langan bo'lishi kerak, ya'ni ictiyoriy bog'langan munosabatlarni berilganlar bazasida amalga oshirilganda, undagi qiyamatlarning qanday bo'lischenidan qat'i nazar, ularning noyob bo'lishi talab qilinadi.*

*Shuni ham qo'shimcha qilish kerakki, relatsion modellarning mashhurligi natijasida, hozirgi kundagi relatsion bo'lmagan tizimlar maxsus ravishda relatsion aloqa bilan ta'minlanmoqdalar. Asosiy tarmoq BBBT Computer Associates firmasining IDMS tizimi, hozir CA-IDMS/SQL deb ataladi va berilganlarning relatsion taqdimotini ishlataladi.*

## **Nazorat uchun savollar**

1. Berilganlarning tarmoq modeli qachon va kim tomonidan yaratilgan?
2. DBTG guruhi taklif etgan uchta asosiy tarkiblar nimalardan iborat edi.
3. Standartlashga taklif etilgan tillar, qaysi tillardan iborat?
4. Berilganlarning tarmoq modelini ta'riflang.
5. Oddiy tarmoq modelini (tuzilishini) ta'riflang va unga misol keltiring.
6. Murakkab tarmoq modelini (tuzilishini) ta'riflang va unga misol keltiring.
7. Oddiy tarmoq modeli chizmasining nusxasiga misol keltiring.
8. Murakkab tarmoq modeli chizmasining nusxasiga misol keltiring.
9. Kesishishdagi berilganlarni ta'riflang.
10. M:M turidagi aloqalarda qanday muammolar mavjud?
11. Berilganlarning tarmoq modelidagi takrorlash, deganda nimani tushunasiz?
12. Berilganlarning tarmoq modelidagi halqa, deganda nimani tushunasiz?
13. Halqadan qutilishga misol keltiring.
14. Tarmoq modelini iyerarxik modelga o'tkazishga misol keltiring
15. Tarmoq modelini iyerarxik modelga aylantirishda nima hosil bo'ladi?

## **6-bob. RELATSION MODELLARGA KIRISH**

**Tayanch iboralar:** axborot qoidasi, kafolatli kirish, bo'sh qiymatlarni to'ldirish, onlaynli relatsion jurnal, berilganlarni boshqarish tili, kiritish, o'zgartirish, o'chirish, berilganlarning fizik mustaqilligi, ma'lumotlarning mantiqiy mustaqilligi, butunlikning mustaqilligi, taqsimot mustaqilligi, buzmaslik, munosabat, tutashuv, jadval, ikki o'chovli fayl, skalyar turlar, murakkab tur, havola turi, domen, atribut nomi, domen nomi, sarlavha, tana, atribut qiymati, relatsion berilganlar bazasi, relatsion berilganlar bazasining chizmasi, relatsion berilganlar bazasi chizmasining nusxasi, n-darajoli munosabat, munosabat quvvati (kardinalligi), ustun, maydon, mantiqiy kalit, fizik kalit, superkalit, nomzod kalit, tashqi kalit, majburiy aloqa, aloqaning darajasi.

Relatsion tizimlar birdaniga keng tarqalmadi. Berilganlar bazasining asosiy nazariy natijalari o'tgan asming 70- yillarda olingan va shu vaqtarda relatsion BBBT namunalari yaratilganga qaramay, uzoq yillar davomida bunday tizimlarning samarasini juda past, deb hisoblangan. Lekin, relatsion BB tashkil qilish va ularni boshqarishdagi usullar va algoritmlar asta-sekin to'plamib borishi natijasida, XX asming 80- yillarining o'talariga kelib, relatsion tizimlar bozordan dastlabki yaratilgan BBB tizimlarini siqib chiqardi.

Berilganlarning relatsion modeli, to'plamlar nazariyasi va predikatlar mantig'idan kelib chiqadigan matematik tamoyillarga asoslanadi. Bu tamoyillar birinchi marta berilganlarni modellash sohasida 1960- yillar oxirida o'sha paytda IBM kompaniyasi xodimi bo'lgan doktor E.F. Kodd tomonidan ishlataligan bo'lib, 1970-yilda e'lon qilingan [17].

### **Kodd qoidalari**

Doktor E.F. Kodd tomonidan 1970-yilda e'lon qilingan «Katta bo'li-nuvchan berilganlar banki uchun berilganlarning relatsion modeli» [17] nomli maqola zamonaviy relatsion BB avlodining boshida turadi deb, hisoblash mumkin. Doktor E.F. Kodd relatsion modelning 13ta qoidasini aniqlab berdi (ularni *Koddning 12ta qoidasi* deyiladi).

Koddning 12ta qoidasi:

0. Relatsion BBBT berilganlar bazasini o'zidagi relatsion imkoniyatlari orqali to'liq boshqarishga ega bo'lishi kerak.

1. Axborot qoidasi – relatsion BBdagi barcha ma'lumotlar (jumladan, jadvallar nomi va ustunlar nomi) jadvaldag'i qiymat sifatida qat'iy aniqlanishi kerak.

2. Kafolatli kirish – relatsion BBdagi har qanday ma'lumotga kirish jadval nomi, birlamchi kalit va ustunlar nomi kombinatsiyalari orqali kafolatlangan bo'lishi kerak.

3. Bo'sh qiymatlarni to'ldirish (null value) – BBBT oshkormas ravishda beriladigan qiymatlardan tashqari, ixtiyoriy domenlarga bog'liqsiz ravishda bo'sh qiymatlar bilan ishlashni (noma'lum yoki foydalanimagan qiymatlar) bilishi lozim.

4. Onlaynli relatsion jurnal – BBning tavsifi va uning mazmuni, mantiqiy darajada jadval sifatida ko'rsatilgan bo'sib, usarga BBning tilidan foydalangan holda so'rovlarni qo'llash imkoniyati yaratilishi kerak.

5. Berilganlarni boshqarish tili to'liq – kamida bitta til aniq belgilangan sintaksisga ega va keng qamrovli bo'lishi kerak. U, berilganlarni tuzilishining tavsifini, berilganlar ustida amallar bajarishni, butunlik qoidasi, mualliflik huquqi va tranzaksiyalarni qo'llab-quvvatlashi kerak.

6. Ko'rsatilganlarni (views) o'zgartirish qoidasi – nazariy yangilanishi mumkin bo'lganlar, har qanday ko'rsatilganlar tizim orqali yangilanishi lozim.

7. Kiritish, o'zgartirish, o'chirish – BBBT faqat berilganlarni tanlashni emas, balki kiritish, o'zgartirish, o'chirish amallarini ham bajarishi lozim.

8. Berilganlarning fizik mustaqilligi – ilova programmalar va maxsus programmalar, berilganlarga fizik usulda kirish va berilganlarning saqlanish tuzilmalariga, mantiqiy ta'sir qilmaydi.

9. Ma'lumotlarning mantiqiy mustaqilligi – ilova-programmalar va maxsus programmalar jadvallar tuzilishidagi o'zgarishlarga mantiqan ta'sir qilmaydi.

10. Butunlikning mustaqilligi – BB tili butunlik qoidalarini aniqlash imkoniyatiga ega bo'lishi kerak. Ular onlayn bo'limida saqlanishi va uni chetlab o'tib ma'lumotlarga o'zgartirish kiritish mumkin bo'imasligi kerak.

11. Taqsimot mustaqilligi – ilova-programmalar va maxsus programmalarga, ma'lumotlarning birinchi marta ishlatalishi yoki takroriy ishlatalishidan qat'i nazar, mantiqiy ta'sir qilmaydi.

12. Buzmaslik – quyi darajadagi tillardan foydalangan holda, BB tili orqali aniqlangan berilganlar butunligi qoidasini buzishga yo'l qo'ymaslik.

E.F. Kodd relatsion algebrani BB relatsion boshqarish tizimiga, berilganlarni bog'langan termalarga taqsimlash uchun qo'llashni taklif qildi. U o'zining BB tizimini, berilganlar termasi asosidagi konsepsiya atrofida

yaratdi [10-13,15-18]. Berilganlarning relatsion modelini yaratgani uchun 1981-yilda E.F. Kodd Hisoblash texnikasi bo'yicha Amerika assotsiat-siyasining Tyuring mukofotiga sazovor bo'ldi.

### Relatsion model tushunchalari

Relatsion modelda berilganlar termalarga ajratilib, ular jadvalning **tuzilishini** yaratadi. Jadvalning bunday tuzilishi alohida berilganlar elementlari, ya'ni *maydonlardan* iborat bo'ladi. Alohida terma yoki maydonlar guruhi *yozuv* deyiladi.

Berilganlar modeli yoki predmet shohaning konseptual tavsifi – bular BB loyihalashdagi eng abstrakt daraja.

Relatsion berilganlar bazasining nuqtayi nazariy jihatidan, relatsion modelning asosiy tamoyillarini konseptual darajada qo'yidagicha ta'riflash mumkin:

- barcha berilganlar, satr va ustunlar shaklidagi *munosabat* deb ataladigan, tartiblangan tuzilishda bo'lishi;
- hamma qiyamatlar skalyar bo'lishi. Bu, ixtiyoriy munosabatning ixtiyoriy ustun va satrlaridagi qiyamatlar yagona bo'lishini bildiradi;
- barcha amallar butun munosabat ustida bajariladi, amallarning bajarilishi natijasida yana butun munosabat hosil bo'ladi. Bu tamoyil *tutashuv* deb ataladi.

Relatsion modelning tamoyillarini shakllantirishda, doktor E.F. Kodd o'zi matematik bo'lgani uchun ham «*munosabat*» (relation) degan atamani tanladi, chunki uning fikricha, bu atama bir xildagi ma'noni anglatadi. *Munosabatning* fizik shakli ikki o'lchovli *jadval* bo'lib, u ustun va satrlardan iborat. Relatsion model deyilishining sababi, uning jadvallar orasidagi aloqalarni aniqlashida, degan noto'g'ri talqin juda keng tarqalgan. Haqiqatda esa, ushbu modelning nomi munosabat (BB jadvaljari)dan, uning asosida yotuvchilardan kelib chiqadi. Mantiqiy jihatdan BB – bu, ikki o'lchovli jadvallar va ular ustida ustunlarni tanlash va birlashtirish amallariga ega bo'lgan to'plamadir. Shu asosda E.F. Kodd quyidagicha deb yozgan «*Amatiy programmachi ham, tasodifiy foydalanuvchi ham, berilganlar bazasi – bu vaqt vaqt bilan o'zgarib turuvchi, hamma mumkin bo'lgan darajalardagi normallashgan munosabatlarning to'plamidir*».

Relatsion modellarda munosabatlar obyektlar xususidagi berilganlarni BBda saqlash uchun ishlataladi. Munosabatlar, aytilgandek jadval ko'rinishida bo'lib, undagi satrlar alohida *yozuvlarga*, ustunlar esa *atributlarga* mos keladi. Bundan tashqari atributlar ixtiyoriy ketma-ketlikda kelishi mumkin, bundan munosabatning ma'nosi o'zgarmaydi.

Relatsion modelining yana bir muqobil tushunchasi K. Deyt [66] tomonidan berilgan. K. Deyt bo'yicha relatsion model uch qismidan iborat:

- tuzilish;
- butunlik;
- ishlov qismi.

*Tuzilish qismida* qaysi obyektlarning relatsion modelda ishlatalishi tavsiflanadi.

*Butunlik qismida* relatsion modeldag'i barcha berilganlarning tegishli cheklashlari tavsiflanadi, jumladan mohiyat butunlig'i va tashqi kalitlar (havola) butunligini.

*Ishlov qismida* ikkita ekvivalent bo'lgan ishlov, ya'ni relatsion algebra va relatsion hisoblash usullari keltiriladi.

### **Tuzilish qismi**

Umuniy holda BBB tizimlaridagi berilganlarning uchta asosiy turlari mavjud:

- oddiy;
- murakkab;
- havola.

32- va 36-boblarda mos ravishda BBB tizimlari MS Access va MS SQL Server tanishganimizda berilganlarning boshqa hamma turlari xususida to'liq ma'lumot beramiz.

#### **Berilganlarni oddiy turi**

Oddiy (atomar) turlar ichki tuzilishga ega emas. Bunday turlarni *skalyar turlar* deb nomlashadi. Oddiy turlarga quyidagilar kiradi:

- mantiqiy;
- matn;
- son.

O'z navbatida son turi – butun son, haqiqiy son, sana va pullik qiymatlar turlariga bo'linishi mumkin.

*Murakkab turga* maydon turlari kiradi. Ular oddiy turlardan tayanch topgan bo'lib, ichki tuzilishga ega. Masalan, Microsoft Word hujjatlari, Microsoft Excel jadvallari, rasmlar, tovushlar, kuy va boshqa ikkijlik berilganlar bo'lishi mumkin.

*Havola turlar* boshqa berilganlarga murojaat qilish uchun ishiatiladi.

#### **Relatsion modelda ishlataladigan turlar**

Relatsion modelda oddiy, murakkab va havola turlari ishlataladi. Relatsion amallarda berilganlarning ichki tuzilishi hisobga olinmaydi. Shu nuqtayi nazardan, har qanday murakkab turdag'i berilganlarni oddiy turlarga

olib kelish mumkin. Buning uchun uni shartli ravishda yaxlit hisoblab, ichki tuzilishi hisobga olinmasligi kerak. Bunday *model postrelatsion* hisoblanadi.

### Domen tushunchasi

Relatsion modelda berilganlar turi tushunchasi bilan uni aniqlovchi domen tushunchasi bir-biriga uzviy bog'liq. Domen bu semantik atamadir. **Domen** deb, biror bir berilganlar turini qandaydir ma'noga ega bo'lgan to'plamning qismini, desak ham bo'ladi.

*Domen* quyidagi xossalarga ega:

- noyob ismga ega bo'lib, atribut nomi bilan bir xilda bo'ladi;
- biror bir oddiy tur yoki boshqa domen orqali aniqlanadi;
- domen biron bir mantiqiy shart bilan berilishi mumkin. Bu shart *shu* domenga tegishli berilganlar to'plamining terma qismini aniqlaydi;
- domen aniq bir ma'noga ega bo'lishi kerak.

Masalan, oly o'quv yurti talabasining yoshini ko'rsatuvchi domenni quyidagicha tavsiflash mumkin:

$$T_{yosh} q \{n \in N: n \geq 16 \text{ and } n \leq 60\}$$

Domenni to'plamdan farqi shundaki, u aniq bir predmet sohaning semantik atamalarini akslantiradi. Bitta to'plamning elementlari har xil ma'noga ega bo'lgan domenlarga tegishli bo'lishi mumkin. Masalan, talabaning yoshi va og'irligi umumiy holda, bir to'plamning elementlari bo'lishi mumkin, lekin bu elementlarning ma'nosini turlicha, shuning uchun ular turli domenlarga tegishli bo'ladi. Bu esa o'z navbatida, turli turdag'i va hattoki bir turdag'i domen elementlarini taqqoslashi taqiqlanadi. Domenlar berilganlar bazasini to'g'ri modellashtirishga ta'sir qiladi. Lekin, hamma domenlarga ham chegaraviy shartlarni qo'yib bo'lmaydi. Masalan, familiya domeniga.

Domen tushunchasi relatsion modellarda katta ahamiyatga ega, chunki foydalanuvchi atributlar qabul qiladigan qiymatlarning ma'nosini va manbasini unga nisbatan aniqlaydi. Natijada tizim relatsion amalni bajarganda, unda ko'proq ma'lumot bo'ladi, bu esa tizimga semantik nokorrekt amallarни bajarishiga yo'l qo'ymaydi. Masalan, agarda ikkita atributlar uchun domen tomonidan belgili satrlar aniqlangan bo'lsa ham, birida mahallaning nomi ikkinchisida ko'chaning nomi saqlansada, ularni solishtirish mantiqsizlik bo'ladi. Lekin bu masalaga boshqa tomonidan qaralsa, masalan talabani oladigan stipendiyasi miqdori va stipendiya olgan oylar soni atributlariga e'tibor beraylik. Birinchi atribut «pul» («denejno'y») turida, ikkinchisi esa butun turida. Lekin ularni ko'paytirish mumkin. Mana shu ikkita misoldan, domen tushunchasini to'liq amalga oshirishni ta'minlash, oson emas, shuning uchun ham ko'pgina relatsion BBBTda ular to'liq emas, qisman amalga oshgan.

## Munosabat, atribut, kortej va boshqa atamalar

Berilganlarning relatsion modelidagi asosiy tushunchalardan biri, bu munosabatdir. Uni ta'riflashdan oldin bir nechta ta'riflarni beramiz.

**Ta'rif 1.** Munosabat atributi bu juftlikdan iborat, ya'ni *atribut nomi : domen nomi* ( $A : D$ ).

**Ta'rif 2.** ( $D_1, D_2, \dots, D_n$ ) har xil turda bo'lishi shart bo'lmagan domenlar to'plamida aniqlangan R *munosabat ikki qismidan*, ya'ni *sarlavha va tanadan* iborat bo'ldi.

Munosabat sarlavhasi fiksirlangan atributlar sonidan iborat bo'lib, quyidagicha juftliklar ketma-ketligi shaklida yoziladi:

$$\langle A_1 : D_1 \rangle, \langle A_2 : D_2 \rangle, \dots, \langle A_n : D_n \rangle$$

Munosabat tanasi kortejlar to'plamidan iborat. Munosabatning har bir korteji *atribut ismi : atribut qiymati* juftlik to'plamidan iborat, ya'ni

$$\langle A_1 : a_1 \rangle, \langle A_2 : a_2 \rangle, \dots, \langle A_n : a_n \rangle$$

bu yerda,  $a_i \in D_i$  domenden olingan  $A_i$  atributning qiymati.

Munosabatning to'liq shaklida yozilishi

$$\langle R \rangle (\langle A_1 : D_1 \rangle, \langle A_2 : D_2 \rangle, \dots, \langle A_n : D_n \rangle)$$

yo qisqacha

$$R (A_1, A_2, \dots, A_n) \text{ yoki oddiygina } R$$

**Ta'rif 3.** Munosabatlari to'plami *relatsion berilganlar bazasi*, deb ataladi.

**Ta'rif 4.** Relatsion berilganlar *bazasining chizmasi*, bazani tashkil etuvchi munosabatlari nomidan tuzilgan to'plamdan iborat.

**Ta'rif 5.** Relatsion berilganlar *bazasi chizmasining nusxasi*, bazani tashkil etuvchi munosabatlarning tanasi (qiymati)dan tuzilgan to'plamdan iborat.

*Munosabat va oddiy jadvallar (relatsion ma'noda emas) orasidagi asosiy farqlar:*

- Munosabatda bir xildagi kortejlar bo'lishi mumkin emas, chunki munosabat tanasi to'plamidan iborat bo'lgani uchun, unda bir xildagi elementlar bo'lishi mumkin emas. Aksincha, jadvalda bir xildagi satrlar bo'lishi mumkin.
  - Munosabat kortejlari tartiblangan emas. Jadvalda esa, har bir satr o'z joyida bo'lishi kerak.
  - Sarlavhadagi atributlar tartiblangan emas. Jadvaldagi tartibi esa fiksirlangan.
  - Atributlarni barcha qiymati skalyar. Jadval yacheykasida har narsa bo'lishi mumkin.

Jadval munosabatga aylanishi uchun:

- oddiy tuzilishga ega bo'lishi, ya'ni har bir satrda ustunlar soni bir xil bo'lishi kerak;
- ikkita bir xildagi satr bo'lmasligi kerak;
- ustunda bir xil turdag'i qiymatlar bo'lishi kerak;
- berilganlar turi oddiy bo'lishi shart.

Munosabat tanasi kortejlarning to'plamidan iborat bo'lgani uchun, munosabatda n ta atribut bo'lsa, kortejlar n o'lchovli, ya'ni *munosabat n-darajali* bo'ladi. 2- darajali munosabat binar, 3- darajali ternar, n- darajalisi esa n ar, deb yuritiladi. Boshqacha qilib aystsak, atributlar soni munosabatning darajasini bildiradi. Munosabatdagi kortejlar to'plami (jadvaldagi satrlar soniga o'xshash) uning *quvvati* yoki *kardinalligi* deb ataladi.

Matematikada odadta ko'proq binar munosabatlar ishlatsa, berilganlar bazasining nazariyasida esa asosan n- darajali munosabatlar ishlataladi. Matematikada munosabatlar cheksiz to'plamlar ustida beriladi va cheksiz quvvatga ega. Berilganlar bazasida esa aksincha, munosabatlarning quvvati cheklangan, chunki saqlanadigan jadvallardagi satrlarning soni hamma vaqt cheklangan bo'ladi.

Shu nuqtayi nazardan BB relatsion modeli, turli ardag'i cheklangan munosabatlarning oldindan aniqlangan elementar berilganlar to'plami orasidagi cheklangan termalardan iborat, deyish mumkin.

BB relatsion modeli matematik mantiq ma'nosidagi chekli to'plamdan iborat. Modeldag'i munosabatlar ustida turli algebrayik amallarni bajarish mumkin. Shu nuqtayi nazardan, matematik mantiq va algebraga asoslangan holda, aniq matematik rasmiyatçılıkni talab qiladi. BB relatsion modellariga ishlov beruvchi zamonaviy tillar, ana shu talablar asosida yaratilgan (SQL, QBE, PL/SQL va h.k.).

Munosabat tuzilishining domen identifikatori va atributlar olishi mumkin bo'lgan qiymatlarining ixtiyoriy cheklashlar bilan tavsifini, ba'zan uning *sarlavhasi* (yoki mazmuni - intension) deyiladi. Odadta, yangi atributlar qo'shilmaguncha munosabatning ma'nosi o'zgarmaydi, u fiksirlangan bo'ladi. Kortejlar ma'lum vaqtidan keyin o'zgarishi mumkin.

### **Muqobil atamalar**

Relatsion modelda ishlatiladigan atamalar ba'zan chalkashliklarga olib keladi, chunki yuqorida biz keltirgan atamalardan tashqari yana boshqa bir atamalar mavjud. Unda munosabat - fayl, kortej - yozuv, atributlar - maydon deyiladi. Bu atamalar, BBBT fizik jihatdan har bir munosabatni alohida fayllarda saqlashidan kelib chiqqan. Bu atamalardan programmachi ko'proq foydalanaq. Quyidagi jadvalda ana shu muqobil atamalarni keltiramiz.

## Munosabat, atribut, kortej va boshqa atamalar

Berilganlarning relatsion modelidagi asosiy tushunchalardan biri, bu munosabatdir. Uni ta'riflashdan oldin bir nechta ta'riflarni beramiz.

**Ta'rif 1.** Munosabat atributi bu juftlikdan iborat, ya'ni *atribut nomi : domen nomi* ( $A : D$ ).

**Ta'rif 2.** ( $D_1, D_2, \dots, D_n$ ) har xil turda bo'lishi shart bo'lmagan domenlar to'plamida aniqlangan R *munosabat ikki qismdan*, ya'ni *sarlavha va tanadan* iborat bo'ladi.

Munosabat sarlavhasi fiksirlangan atributlar sonidan iborat bo'lib, quyidagicha juftliklar ketma-ketligi shaklida yoziladi:

( $A_1 : D_1, A_2 : D_2, \dots, A_n : D_n$ )

Munosabat tanasi kortejlar to'plamidan iborat. Munosabatning har bir korteji *atribut ismi : atribut qiymati* juftlik to'plamidan iborat, ya'ni

( $A_1 : a_1, A_2 : a_2, \dots, A_n : a_n$ )

bu yerda,  $a_i$  —  $D_i$  domendan olingan  $A_i$  atributning qiymati. Munosabatning to'liq shaklda yozilishi

$\langle R \rangle (A_1 : D_1, A_2 : D_2, \dots, A_n : D_n)$

yo qisqacha

$R (A_1, A_2, \dots, A_n)$  yoki oddiygina R

**Ta'rif 3.** Munosabatlar to'plami *relatsion berilganlar bazasi*, deb ataladi.

**Ta'rif 4.** Relatsion berilganlar *bazasining chizmasi*, bazani tashkil etuvchi munosabatlar nomidan tuzilgan to'plamdan iborat.

**Ta'rif 5.** Relatsion berilganlar *bazasi chizmasining nusxasi*, bazani tashkil etuvchi munosabatlarning tanasi (qiymati)dan tuzilgan to'plamdan iborat.

*Munosabat va oddiy jadvallar (relatsion ma'noda emas) orasidagi asosiy farqlari:*

- Munosabatda bir xildagi kortejlar bo'lishi mumkin emas, chunki munosabat tanasi to'plamdan iborat bo'lgani uchun, unda bir xildagi elementlar bo'lishi mumkin emas. Aksineha, jadvalda bir xildagi satrlar bo'lishi mumkin.

- Munosabat kortejlarini tartiblangan emas. Jadvalda esa, har bir satr o'z joyida bo'lishi kerak.

- Sarlavhadagi atributlar tartiblangan emas. Jadvaldagi tartibi esa fiksirlangan.

- Atributlarni barcha qiymati skalyar. Jadval yacheykasida har narsa bo'lishi mumkin.

Jadval munosabatga aylanishi uchun:

- oddiy tuzilishga ega bo'lishi, ya'ni har bir satrda ustunlar soni bir xil bo'lishi kerak;
- ikkita bir xildagi satr bo'lmasligi kerak;
- ustunda bir xil turdag'i qiymatlar bo'lishi kerak;
- berilganlar turi oddiy bo'lishi shart.

Munosabat tanasi kortejlarning to'plamidan iborat bo'lgani uchun, munosabatda n ta atribut bo'lsa, kortejlar n o'lchovli, ya'ni *munosabat n-darajali* bo'ladi. 2- darajali munosabat binar, 3- darajali ternar, n- darajalisi esa n ar, deb yuritiladi. Boshqacha qilib aytaksak, atributlar soni munosabatning darajasini bildiradi. Munosabatdagi kortejlar to'plami (jadvaldagi satrlar soniga o'xshash) uning *quvvati* yoki *kardinalligi* deb ataladi.

Matematikada odatda ko'proq binar munosabatlar ishlatsa, berilganlar bazasining nazariyasida esa asosan n- darajali munosabatlar ishlataladi. Matematikada munosabatlar cheksiz to'plamlar ustida beriladi va cheksiz quvvatga ega. Berilganlar bazasida esa aksincha, munosabatlarning quvvati cheklangan, chunki saqlanadigan jadvallardagi satrlarning soni hamma vaqt cheklangan bo'ladi.

Shu nuqtayi nazardan BB relatsion modeli, turli ardag'i cheklangan munosabatlarning oldindan aniqlangan elementar berilganlar to'plami orasidagi cheklangan termalardan iborat, deyish mumkin.

BB relatsion modeli matematik mantiq ma'nosidagi chekli to'plamdan iborat. Modeldag'i munosabatlar ustida turli algebrayik amallarni bajarish mumkin. Shu nuqtayi nazardan, matematik mantiq va algebraga asoslangan holda, aniq matematik rasmiyatichilikni talab qiladi. BB relatsien modellariga ishlov beruvchi zamonaviy tillar, ana shu talablar asosida yaratilgan (SQL, QBE, PL/SQL va h.k.).

Munosabat tuzilishining domen identifikatori va atributlar olishi mumkin bo'lgan qiymatlarining ixtiyoriy cheklashlar bilan tavsifini, ba'zan uning *sarlavhasi* (yoki mazmuni - intension) deyiladi. Odatda, yangi atributlar qo'shilmaguncha munosabatning ma'nosini o'zgarmaydi, u fiksirlangan bo'ladi. Kortejlar ma'lum vaqtadan keyin o'zgarishi mumkin.

### **Muqobil atamalar**

Relatsion modelda ishlataladigan atamalar ba'zan chalkashliklarga olib keladi, chunki yuqorida biz keltirgan atamalardan tashqari yana boshqa bir atamalar mavjud. Unda munosabat - fayl, kortej - yozuv, atributlar - maydon deyiladi. Bu atamalar, BBT fizik jihatdan har bir munosabatni alohiда fayllarda saqlashidan kelib chiqqan. Bu atamalardan programmachi ko'proq foydalanadi. Quyidagi jadvalda ana shu muqobil atamalarni keltiramiz.

Rasmiy atamalar	Foydalanuvchi atamalari	Programmachi atamalari
Munosabat	Jadval	Fayl
Kortej	Satr	Yozuv
Atribut	Ustun	Maydon

Munosabat quyidagi tavsiflarga ega:

- munosabat relatsion chizmadagi hamma nomlardan boshqacha nomga ega;
- munosabatning har bir katagi faqat bitta elementar qiymatga ega;
- har bir atribut noyob ismga ega;
- atributning qiyatlari bitta domenden olinadi;
- har bir kortej noyob hisoblanadi, ya'ni ikkita bir xildagi kortej bo'lishi mumkin emas;
- atributlarning kelish ketma-ketligi ahamiyatsiz;
- nazariy jihatdan munosabatda kortejlarning kelish ketma-ketligi ahamiyatsiz.

Lekin amaliyotda kortejlarning kelish tartibi ularga kirish samaradorligiga ta'sir qilishi mumkin. Quyidagi jadvalda relatsion atamalar keltirilgan.

### Kalitlar

Aytib o'tilganidek, munosabatda takrorlanuvchi kortejlar bo'lmasligi kerak. Shuning uchun har bir kortejni noyob tarzda bitta yoki bir nechta atributlar yordamida identifikatsiyalash imkoniyati bo'lishi kerak. Bu atributlar *kalit* deyiladi.

Amaliy ish jarayonida kortejni – *yozuv*, atributni *maydon* deyishadi. *Birlamchi kalit* – atributlar yoki ular to'plamining shunday elementidan iboratki, u yagona tarzda bitta yozuv yoki guruhnini belgilaydi. Birlamchi bo'lmagan boshqa atributlarning birlashishidan tuzilgan kalit, *tarkibili* (*birlashgan*) *kalit* bo'ladi.

Kalit BB loyihalash va uni amalga oshirish jarayonlarida turlicha ma'no kasb etadi. Loyihalash jarayonida kalit munosabatdagi satrni bir ma'noda tavsirlovchi bo'lib, bitta yoki bir nechta tushuniladi. Har bir munosabat kamida bitta kalitga ega bo'ladi, chunki har bir kortej noyob hisoblanadi. Maksimal tarzda kalit munosabatdagi hamma domenlarning birlashmasidan tuzilishi mumkin.

Loyjhani amalga oshirish jarayonida kalit atamasi boshqacha tushu-

niladi. Ko'pgina relatsion BBBT kalit sifatida ustun olinadi va shu asosda boshqarish tizimi indekslar yaratadi. Indeks berilgan ustundagi qiymatlarni tezda topishga imkon yaratadi. Bu kalitlarning noyob bo'lishi shart emas, chunki ular faqatgina tezlikni oshirishga xizmat qiladi.

Ana shu kalitlarni bir-biridan farqlash uchun *mantiqiy* va *fizik kalitlar* deb atashadi. Mantiqiy kalit bu ustunni noyob elementlarini, fizik kalit esa tezlikni oshiruvchi indeksdan iborat.

Berilgan munosabatning kortejini yagona tarzda aniqlovchi atribut yoki atributlar majmuasi *superkalit* deyiladi. Superkalit munosabatdagi har bir kortejni bir qiymatli aniqlaydi. Lekin superkalit noyob identifikatsiyalash shart bo'lmasagan qo'shimcha atributlarni ham saqlashi mumkin. Bizni noyob identifikatsiyalash shart atributlar qiziqtiradi.

Berilgan munosabatning superkaliti bo'lgan va qism to'plami bo'lmasagan superkalit, *nomzod kalit* deyiladi. Berilgan R munosabatning N nomzod kaliti ikkita xossaga ega:

- *Noyoblik*. R munosabatning har bir kortejida N nomzod kalitning qiymati shu kortejni yagona ravishda identifikatsiyalaydi.
- *Keltirib bo'lmaslik*. N nomzod kalitning mumkin bo'lgan qism to'plamidan hech biri noyoblik xususiyatiga ega emas.

Munosabat bir nechta nomzod kalitlarga ega bo'lishi mumkin. Agar kalit bir nechta atributdan iborat bo'lsa, u *tarkibli kalit* deb ataladi.

Takrorlanuvchi qismga ega munosabatni, ikki qismga ajratib, biriga shu takrorlanuvchi qismni alohida munosabatga ajratsak, biz shu munosabatni normallashtirgan bo'lamiz. Hesil bo'lgan yangi munosabat o'z nomiga ega bo'ladi (6.1-rasm). Yangi munosabat o'zining kalitiga ega bo'lishi va u, kortejlarni bir qiymatli aniqlashi kerak. BUYURTMA\_RAQAMI elementi MAXSULOTLAR\_PARTIYASI yangi munosabatida MAXSULOT\_RAQAMI elementi bilan birga *tarkibli kalit* bo'lib, kortejlarni bir qiymatli tafsiflaydi.

6.1 kabi jadvalarni chizmaslik uchun qavsdan oldin munosabat nomi va qavsdan domenlar nomi ketma-ket yozilib vergullar bilan ajratiladi. Bir-lamchi kalitlarning tagiga chiziladi. Shu usuldaggi ushbu munosabatlarning ko'rinishi quyidagicha bo'ladi:

SOTIB\_OLISHGA\_BUYURTMA (BUYURTMA RAQAMI, TA'MI-NOTCHI RAQAMI, BUYURTMA SANASI, YETKAZISH SANASI, JAMI)

MAXSULOTLAR PARTIYASI (BUYURTMA RAQAMI, MAXSULOTLAR RAQAMI, NARXI, SONI)

Nomzod kalitni identifikatsiyalash uchun ishlatalidigan atributiarning «haqiqiy dunyo»dag'i ma'nosini chuqr bilish kerak. Faqat shu asosda takrorlanuvchi qiymatlar mavjudligi haqida qaror qabul qilish mumkin. Ana

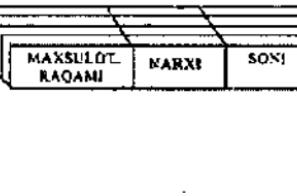
shu semantik ma'lumotga ega bo'lingandagina, atributlarning kombinasiysi munosabatning nomzod kaliti bo'ladi, deb kafolat berish mumkin.

Munosabatdag'i kortejlarni noyob identifikatsiyalash uchun tanlangan nomzod kalit, *birlamchi kalit* deyiladi. Munosabatda kortejlar nusxasi bo'l-magan uchun, uni har bir satrini noyob tarzda identifikatsiyalash mumkin, Bundan, munosabat doimiy ravishda birlamchi kalitga ega bo'lishi kelib chiqadi. Birlamchi kalit sifatida tanlanmagan nomzod kalitlar, *muqobil kalitlar* deyiladi.

Munosabatdag'i atribut yoki atributlar to'plami boshqa munosabatdag'i (yoki o'zidagi) nomzod kalitga mos kelsa, u *tashqi kalit* deb ataladi. Agarda bitta atribut bir nechta munosabatlarda ishtirok etsa, shu munosabatlarning kortejlari orasida aloqalar mavjudligini bildiradi.

#### SOTIB OLİSHGA BUYURTMA

BUYURTMA RAQAMI	TA'MINOT CHI. RAQAMI	BUYURMA SANASI	YILKAZISH SANASI	JAMI



JAMI

#### SOTIB OLİSHGA BUYURTMA

BUYURTMA RAQAMI	TA'MINOT CHI. RAQAMI	BUYURMA SANASI	YILKAZISH SANASI	JAMI

KALIT

#### MAXSULOTLAR PARTIVASI

BUYURTMA RAQAMI	MAXSU- LOT RAQAMI	NARXI	SONI

KALIT

6.1-rasm.

#### Berilganlarning relatsion modellarida chizmalar taqdimoti

Relatsion modelda ixtiyoriy sondagi normal munosabatlар bo'lishi mumkin. Relatsion chizmalarning qabul qilingan belgilash usulida – munosabat nomi va qavsda atributlar nomi ketma-ket yozilib vergullar bilan ajratiladi. Birlamchi kalitlarning tagiga chiziladi.

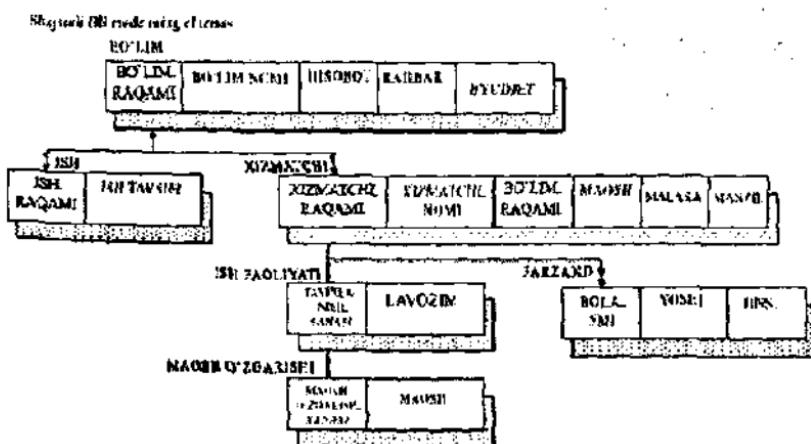
Berilganlar bazasidagi hamma relatsion chizmalar *konseptual modelni* yoki *konseptual chizmani* hosil qiladi.

Relatsion modeldagi har bir aloqa nomi, majburiyati, turi va darajasi bilan tavsiflanadi. Aloqani **fakultativ** va **majburiy** xillarga ajratishadi. Agarda bir turdag'i ma'no boshqa turdag'i ma'no bilan zaruriy bog'langan bo'lsa, u holda ular orasidagi aloqa *majburiy aloqa* deyiladi va ikkilangan chiziq bilan belgilanadi. Qolgan vaziyatlardagi aloqalar **fakultativ** aloqa-larga kiradi.

*Aloqaning darajasi* shu aloqalar bilan bog'langan munosabatlarning soni bilan aniqlanadi.

6.2-rasmida 4- darajali daraxting oltita munosabatga keltirilgan holati tasvirlangan. Ushbu daraxtsimon tuzilishni normallashgan shaklga keltirish uchun, berilganlarga kirish yo'llaridagi bog'liliklari yangi kalitlar elementlarini kiritish natijasida, yo'qolilgan. Yangi kiritilgan kalit elementlari semiz qora shriftda keltirilgan.

Bunda ayrim munosabatlarning kalitlari o'zidan yuqoridagi darajada turuvchi munosabatning kaliti bilan mos kelishi mumkin. Misol uchun, ISH FAOLIYATI munosabati (XIZMATCHI\_RAQAMI, TAYINLANISH\_SANASI) birlashgan kalitdan iborat, uning tarkibida o'zidan yuqoridagi munosabatning XIZMATCHI\_RAQAMI kaliti bor. XIZMATCHI munosabati esa faqat XIZMATCHI\_RAQAMI kalitidan iborat o'lib, bu kalit yozuvning qol-gan atributlarini bir qiymatli aniqlaydi va unga yuqori darajadagi munosabatning kalitini kiritish kerak emas.



6.2-rasm. 4- darajali daraxt.

Chizmalarda milga ega chiziqlar bilan ko'rsatilgan aloqalar, BB amalga oshirishda ilovalarda qaysi yo'l bo'yicha harakatlanish zarurligini bilsidradi. Ba'zi bir yo'naltirish vositalari hamma ana shunday yo'llar bo'yicha normallashgan tasvirlardan iborat bo'lishi kerak. XIZMATCHI munosabatidan FARZANDLAR, ISH FAOLIYATI va MAOSH O'ZGARISHI (6.2-rasm) munosabatlariiga o'tuvchi yo'llar sodda amaiga oshirilishi mumkin, chunki uchchala kortej ham XIZMATCHI\_RAQAMI elementiga ega. XIZMATCHI munosabatdan BO'LIM munosabatiga o'tish yo'lli ham amalga oshirilishi mumkin, chunki XIZMATCHI munosabati BO'LIM\_RAQAMI elementidan iborat. BO'LIM munosabatidan XIZMATCHI kortejiga o'tish ham BO'LIM\_RAQAMI elementidan foydalanishni talab qiladi. Faqat bunda ikkilamchi kalit sifatida ishlataladi

Chizmaning normallashgan shakli:

BO'LIM (BO'LIM RAQAMI, BO'LIM NOMI, HISOBOT, RAHBAR, BUDJET)

ISH (BO'LIM RAQAMI, ISH RAQAMI, ISH TAVSIFI)

XIZMATCHI (XIZMATCHI RAQAMI, XIZMATCHI NOMI, BO'LIM\_RAQAMI, MAOSH, LAVOZIMI, MANZIL)

MAOSH O'ZGARISHI (XIZMATCHI RAQAMI, MAOSH O'ZGA-RISH SANASI, MAOSH)

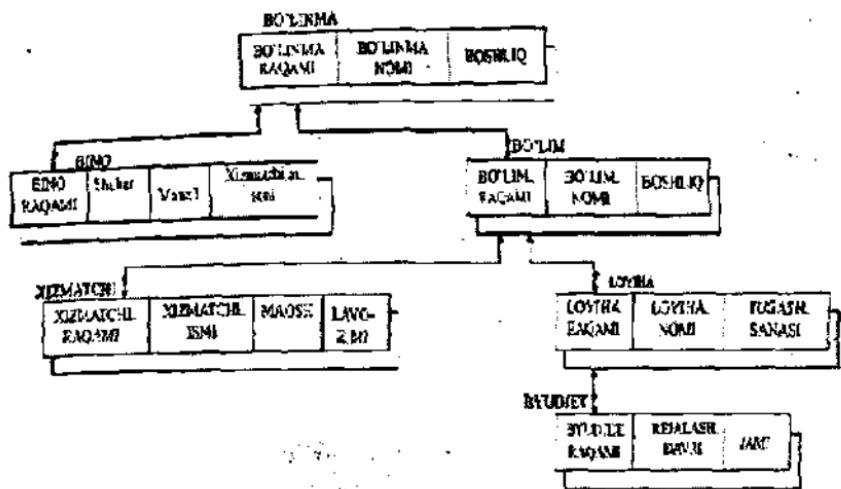
FARZANDLAR (XIZMATCHI RAQAMI, BOLA ISMI, BOLA YOSHI, JNSI)

ISH FAOLIYATI (XIZMATCHI RAQAMI, TAYINLANISH SANASI, LAVOZIMI)

XIZMATCHI munosabatini BO'LIM\_RAQAMI elementlari bo'yicha tartiblash kerak.

6.3-rasmdagi chizmani normallashtirish uchun birlamchi kalit elementlaridan foydalanish talab qilinmaydi. Har bir kalitni kortej bir xilda identifikatsiyalaydi. Lekin qo'shimcha atributlar kiritilganligining sababi, munosabatlar orasidagi aloqalarni kuzatish orqali daraxt cho'qqisiga chiqarish.

Chizmani normalashda qo'shimcha berilganlar elementi ikkita sabab bilan qo'shilishi mumkin. Birinchidan, birlamchi kalit qo'shilib hosil bo'lgan tarkibli kalit, kortejni bir qiymatli identifikatsiyalaydi. Ikkinchidan, an'anaviy chizmalardagi milli chiziqlar orgali isodalangan yo'l tafsiflash uchun ishlataladi. Ikkilangan milli chiziqlar ikkinchi darajali kalitlarga ishlav talab qilishi mumkin. Bu ishlov oddiy tartiblashdan va ikkinchi indekslardan foydalanishdan iborat bo'lishi mumkin.



### 6.3-rasm. Daraxtsimon chizma.

Chizmaning normal jashgan shakli:

BO'LINMA (BO'LINMA RAQAMI, BO'LINMA NOMI, RAHBAR)

BINO (BINO RAQAMI, BO'LINMA RAQAMI, SHAHAR, MANZIL, XIZMATCHILAR SONI)

BO'LIM -(BO'LIM RAQAMI, BO'LINMA RAQAMI, BO'LIM NOMI, RAHBAR)

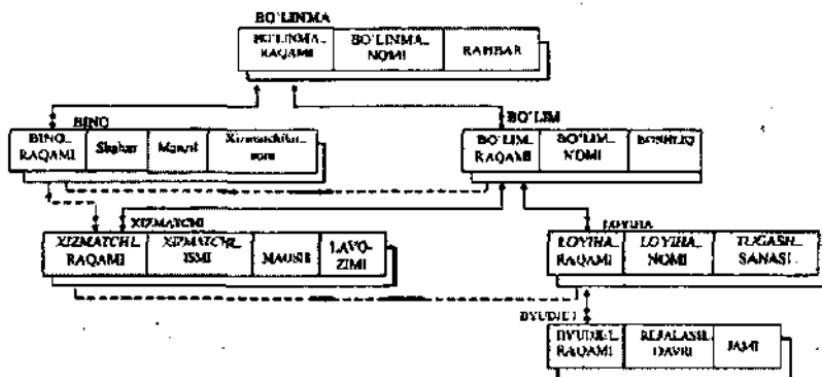
XIZMATCHI (XIZMATCHI RAQAMI, BO'LIM\_RAQAMI, XIZMATCHI ISMI, MAOSH, LAVOZIMI)

LOYIHA (LOVIHA RAQAMI, BO'LIM\_RAQAMI, LOVIHA NOMI, TUGASH SANASI)

BUDJET (BUDJET RAQAMI, LOVIHA-RAQAMI, REJALASH DAVRI, JAMI)

### Normal chizmaga qo'shimchalar

Agarda yangi ilovalar, kortejga yangi berilganlar elementlarini qo'shish talab qilsa yoki foydalanuvchining berilganlar xususidagi fikri o'zgarsa, relationson BB buni yaratilish jarayonida hisobga olishi kerak. Berilganlarni yangi elementlarini qo'shish yoki yangi munosabatni yaratish evaziga, mavjud munosabatlar orasida yangi aloqalar ham o'matilishi mumkin.



6.4-rasm. 6.3-rasmida keltirilgan daraxtsimon tuzilishga qo'shimcha aloqalarni kiritish

6.4-rasmda, 6.3- rasmda keltirilgan daraxtsimon tuzilishga qo'shimcha aloqalarni kiritish natijasida u tarmoq tuzilishidagi berilgalar modeliga aylangan. Buning uchun, XIZMATCHI va BINO munosabatlari orasidagi aloqaga BINO RAQAMI elementi XIZMATCHI munosabatiga, XIZMATCHI va LOYIHA munosabatlari orasidagi aloqaga LOYIHA RAQAMI elementi XIZMATCHI munosabatiga qo'shiladi. Demak, ikkita yangi atributlarni munosabatlarga qo'shish va yangi munosabatni yaratish natijasida qo'shimcha aloqalarga ishlov beriladi.

BINO va BO'LIM munosabatlari orasidagi aloqalar M:M turidan iborat. Bir binoda bir nechta bo'lim va bir bo'lim bir nechta binoda joylashishi mumkin. Agarda BINO munosabatiga yangi BO'LIM\_RAQAMI atributini qo'shadigan bo'lsak, bir bino uchun bir nechta bir xildagi kortejlar qo'shiladi, bu esa munosabatni ortiqchalikka olib keladi. Shuning uchun, BINO RAQAMI va BO'LIM\_RAQAMI atributlardan iborat JOYLASHISH munosabati yaratiladi. Bu berilgan bo'limning qaysi binolarda joylashishi va berilgan binoda qaysi bo'limlar joylashganini topish imkoniyatini beradi.

Umuman, relatsion BB yangi munosabatlar qo'shilganda, eski aloqalar buzilmasligini ta'minlashi kerak.

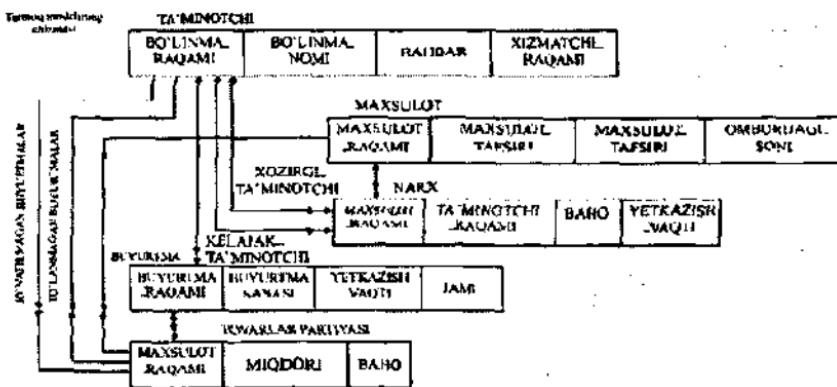
6.4-rasmda keltirilgan tarmoq tuzilishini, xuddi iyerarxik tuzilish kabi oson normallashtirish mumkin. Natijada hosil bo'lgan normallashgan chizma:

**BO'LINMA (BO'LINMA RAQAMI, BO'LINMA NOMI, RAHBAR)**  
**BINO (BINO RAQAMI, BO'LINMA RAQAMI, SHAHAR, MANZIL, XIZMATCHILAR SONI)**

**BO'LIM (BO'LIM RAQAMI, BO'LINMA RAQAMI, BO'LIM NOMI, RAHBAR)**

XIZMATCHI (XIZMATCHI RAQAMI, BO'LIM\_RAQAMI, BINO RAQAMI, LOYIHA RAQAMI, XIZMATCHI ISMI, MAOSH, LAVOZIMI)  
 LOYIHA (LOYIHA RAQAMI, BO'LIM\_RAQAMI, LOYIHA NOMI, TUGASH SANASI)  
 BUDJET (BUDJET RAQAMI, LOYIHA RAQAMI, REJALASH DAVRI JAMI)  
JOYLASHISH (BINO RAQAMI, BO'LIM RAQAMI)

6.5-rasmida tarmoq tuzilishidagi modelni normallashtirish yo'llaridan biri keltirilgan. Yuqorida aytiganidek, kortejlar ularni bir qiymatli aniqlaydigan kalitlarga ega va berilganlar elementlari tarmoq tuzilishidagi ixtiyoriy yo'llar bo'yicha yurishni ta'minlaydi. Berilganlar guruhining ayrim elementlari (6.5-rasm) ularni bog'lovchi ikkitadan aloqaga ega. TA'MINOTCHI va NARX munosabatlari orasidagi aloqa, haqiqatan ham ushbu ta'minotchi NARX munosabatida ko'rsatilgan hozirgi mahsulot yetkazuvchimi yoki kelajakdagi ta'minotchimi, degan savolga javob beradi. TA'MINOTCHI munosabati bilan TOVARLAR PARTIYASI munosabati orasidagi aloqa har bir buyurtma uchun pul to'langan va to'lanmaganligini ko'rsatadi.



6.5-rasm.

Chizmaning normallashgan shakli:

TA'MINOTCHI (TA'MINOTCHI RAQAMI, TA'MINOTCHI ISMI, TA'MINOTCHI MANZILI)

MAXSULOT (MAXSULOT RAQAMI, MAXSULOT NOMI, MAHSULOT TAFSIRI, OMBORDAGI SONI)

HOZIRGI TA'MINOTCHI (MAXSULOT RAQAMI, TA'MINOTCHI RAQAMI, BAHO, YETKAZISH VAQTI)

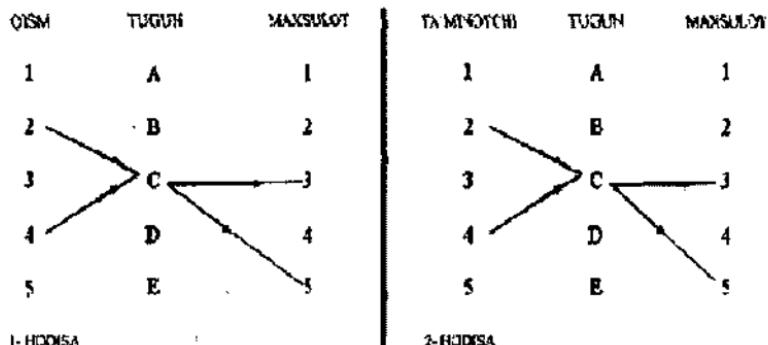
KELAJAK TA'MINOTCHI (MAXSULOT RAQAMI, TA'MINOTCHI RAQAMI, BAHO, YETKAZISH VAQTI)  
 BUYURTMA (BUYURTMA RAQAMI, TA'MINOTCHI RAQAMI, BUYURTMA SANASI, YETKAZISH VAQTI, JAMI)  
 TOVARLAR PARTIYASI (BUYURTMA RAQAMI, MAXSULOT RAQAMI, MIQDORI, BAHO)  
 TO'LAMAGAN BUYURTMALAR (TA'MINOTCHI RAQAMI, MAXSULOT RAQAMI, QO'SHIMCHA VAQT)

### Yo'naltirilgan aloqalar muammosi

Iyerarxik BBB tizimlarida iyerarxik tuzilishlarga ishlov berish mumkin emas, chunki tabiiy kengayish imkoniyati yaratilmagan, bunga 6.4-rasm misol bo'la oladi. Tarmoq tuzilishidagi BBB tizimlarida ham, berilganlar orasidagi aloqalarga chegara qo'yilgan.

6.6-rasmdagi diagrammalarda ifodalangan aloqalar 2-darajali aloqalar dan iborat, ya'ni munosabatlarning ikkita domenlari orasidagi aloqa sifatida qabul qilinadi. Bu esa yuqoriroq darajadagi munosabatlarga ishlov berish imkonini yaratadi, chunki binar (ikkita) munosabat o'rtasidagi aloqalar o'xshash tarzda amalga oshiriladi. Agarda 2-darajadagi munosabatlar milli chiziqlar orqali tasvirlansa, darajaniing oshishini diagrammada aksantirish qiyin bo'ladi, bunda chizmada yangi aloqalar paydo bo'ladi, bu esa amaliy programmalardagi mavjud aloqalarning buzilishiga sabab bo'lishi mumkin.

Chizmalarda bir blokdan ikkinchi blokka undan esa keyingisiga o'tuvchi milli chiziqlar ishlataliganda, foydalanuvchi ushbu belgilarga amal qilgan holda ikkita ikkilik munosabatlami 3- darajali munosabat, deb qabul qiladi bu esa xatoliklarga olib kelishi mumkin.



6.6-rasm. Birlashtirishdagi «pistirma».

6.6-rasmdagi 1- hodisada mil QISM 4 yozuvdan TUGUN S oraliq yozuvga, keyin yozuv MAXSULOT 5ga yo'naltirilgan. Foydalanuvchi 4-qism 5-turdagi mahsulotda ishlataladi, deb fikrlaydi va bu fikr to'g'ri bo'ladi. Endi ikkinchi ho'satni ko'ramiz. Bunda chap ustunda QISM o'miga uning TA'MINOTCHIsi turibdi. Mil TA'MINOTCHI 4 yozuvdan TUGUN S oraliq yozuvga, keyin yozuv MAXSULOT 5ga yo'naltirilgan. Foydalanuvchi 5-mahsulotni yetkazuvchilardan biri 4- ta'minotchi, deb fikrlashi mumkin. Ushbu holatda u noto'g'ri bo'lishi mumkin. 5- mahsulot tarkibiga kiruvchi S tugunni faqat 2- ta'minotchi, 3- mahsulotni faqat 4- ta'minotchi yetkazib berishi ham mumkin. Bu vaziyatni E.F. Kodd birlashtirishdagi «pistirma» deb ataydi. Bu holatga aniqlik kiritish uchun 3- darajali munosabat zarur:

1-hodisa			2-hodisa		
QISM	TUGUN	MAXSULOT	TA'MINOTCHI	TUGUN	MAXSULOT
2	C	3	2	C	5
2	C	5	4	C	3
4	C	3			
4	C	5			

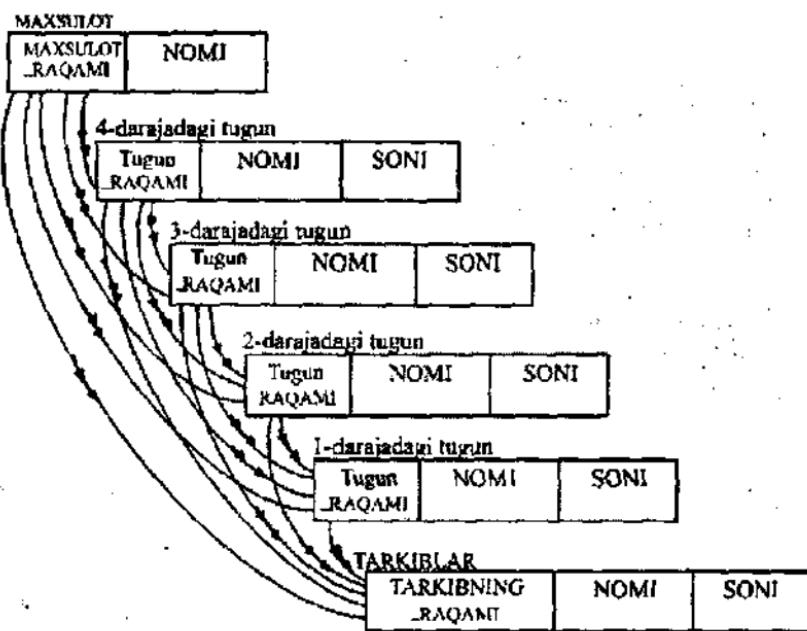
Umuman, BB mantiqiy tuzilishlarini yaratuvchi, chiziqlar va millarni ishlatishi noqulay, chalg'ish va noto'g'ri chizmalarga olib keladi. Chizmalarning relatsion tasviri esa ixcham, tushunish sodda va matematik jihatga ega. Lekin shunga qaramay, ehtiyyotsizlik bilan bajarilgan amallar keltirilgan birlashtirish «pistirma»lari kabi qiyinchiliklarni tug'dirishi mumkin.

Ko'pgina holatlarda normallashgan chizma an'anaviy chizmaga nisbatan yengilroq yaratiladi. 6.7-rasmda materiallar ro'yxati berilganlar bazasining chizmasi keltirilgan.

Korxonaning mahsuloti turli qismlar, masalan, gayka va boltlar, turli tugunlar, ya'ni motor va dvigatellardan iborat bo'lsin. Tugun esa alohida agregat yoki mayda tugunlardan tashkil topgan bo'lsin. 4- darajadagi tugun 3- darajadagi tugundan, 3- darajadagisi esa 2- darajadagi tugundan va 2-darajadagisi esa 1- darajadagi tugundan tashkil topgan bo'lsin. Materiallar ro'yxatining BB u yoki bu mahsulot yoki tugun nimadan tashkil topganini ko'rsatsin.

Normallashgan chizma ikkita munosabatdan iborat bo'ladi:

- MAXSULOT munosabatidan, uning domenlari MAXSULOT RAQAMI, NOMI va KATEGORIYA (ushbu atribut tayyor mahsulotgami, mahsulot qismigami yoki qaysi darajadagi tugunga tegishliliği xususidagi berilgan);
- TUGUNLARNI BO'SHATISH munosabatidan mahsulot nomeri, uning tarkibiy qismlarining nomeri va soni.



**6.7-rasm.** Normallashgan shakl – chizmani tasvirlashdagí soddaroq, oddiyroq va aniq tasvirlash usuli.

Normallashgan shakl:

**MAXSULOT (MAXSULOT RAQAMI, KATEGORIYA, NOMI)**

**TUGUNLARNI\_BO'SHATISH (MAXSULOT RAQAMI, TARKIB-NING RAQAMI, SONI)**

Bunday holdagi BB yaratish, mahsulotni ishlab chiqarish va kelajakda kiritiladigan konstrukturlik o'zgartirishlarni hisobga oladi. Har qanday yangi mahsulot kiritilgan konstrukturlik o'zgartirishlaridan keyin, ikkita munosabatga qo'shiluvchi yangi munosabat yaratiladi – O'ZGARTIRISH (MAXSULOT RAQAMI, O'ZGARTIRISH RAQAMI).

### Xulosa

*Relation modellarning muhimligiga asosan uchta sabab bor. Birinchidan, relatsion modellarning konstruksiyasi keng va umumiy tavsif nomaga ega bo'lib, berilganlar bazasining tuzilishini BBB tizimlariga bog'liq bo'limgan holda tavsiflash imkoniyatini beradi. Ikkinchidan, relation model deyarli hamma BBB tiziminining asosini tashkil qiladi. Uchinchidan, axborot tizimlarining «yurag»ini tashkil etuvchilar, bu*

*berilganlar bazasining relatsion modelidir.*

*Shularga asosan, relatsion model tamoyillarini tushunish zamonaviy mutaxassisining asosiy bilim «qurol»laridan biriga aylangan.*

### **Nazorat savollari:**

1. Berilganlarning relatsion modeli qachon va kim tomonidan yaratilgan?
2. E.F. Koddning relatsion modeli ta'rifi asosida nechta qoida yotadi?
3. Berilganlarning relatsion modeli fizik tarzda nimani ifodalaydi?
4. Relatsion atamasining ma'nosi nima?
5. Domen nimani anglatadi?
6. Atribut nimani anglatadi?
7. Kortej nimani anglatadi?
8. Munosabat qanday asosiy qoidalarga bo'y sunishi kerak?
9. Munosabatdagi kalit nimani anglatadi?
10. Tarkibli kalit, deganda nimani tushunasiz?
11. Fizik kalit nimani anglatadi?
12. Superkalit, deganda nimani tushunasiz?
13. Nomzod kalit, deganda nimani tushunasiz?
14. Relatsion modelni chizmasiz ko'rinishiga misol keltiring.
15. Muqobil kalit, deganda nimani tushunasiz?
16. Tashqi kalit, deganda nimani tushunasiz?
17. Relatsion modelda aloqaning nechta xili mavjud?
18. Aloqaning darajasi nimani bildiradi?
19. Relatsion modeldagи chizmalarda ifodalangan milga ega chiziqlar nimani bildiradi?
20. Relatsion model chizmasini normallashtirish uchun nima qilish kerak?
21. Relatsion modelga yangi munosabat qo'shilsa, aloqalar o'zgaradimi yoki yo'qmi?
22. Normallashgan chizmaga yana qanday qo'shimchalarni kiritish mumkin?
23. Yo'naltirilgan aloqalar muammosi nimadan iborat?
24. Yo'naltirilgan aloqalar muammosidan qutilish uchun nima qilish kerak?

---

## 7-bob. BERILGANLARNING «MOHIYAT-ALOQA» MODELI

**Tayanch iboralar:** «mohiyat-aloha» modeli, axborot qoidasi, kafolatli kirish, bo'sh qiymatlarni to'ldirish, onlaynli relatsion jurnal, berilganlarning boshqarish tili, rekursiv aloqa, zaif mohiyat, kuchli mohiyat, ajdod mohiyat, voris mohiyat, berilganlarning fizik mustaqilligi, ma'lumotlarning mantiqiy mustaqilligi, butunlikning mustaqilligi, taqsimot mustaqilligi, buzmastlik, munosabat, tutashuv, jadval, ikki o'chovli sayl, skalyar turlar, murakkab tur, havola turi, domen, atribut nomi, domen nomi, sarlavha, tana, atribut qiymati, relatsion berilganlar bazasi, relatsion berilganlar bazasining chizmasi, relatsion berilganlar bazasi chizmasining nusxasi, n- darajali munosabat, munosabat quvvati (kardinalligi), ustun, maydon, mantiqiy kalit, fizik kalit, superkalit, nomizod kalit, tashqi kalit, majburiy aloqa, aloqaning darajasi.

«Mohiyat-aloha» modeli (ER – Entity-Relationship model) berilganlar bazasini loyihalashning konseptual darajasidagi chizmalarini tavsiflash uchun ishlataladi. Bu usul 1976-yilda Peter Pin Shan Chen (Peter Pin Shan Chen) [9] tomonidan kiritilgan. Keyinchalik Chen va boshqalar tomonidan kengaytirildi va takomillashtirildi [52]. Bundan tashqari «mohiyat-aloha» modeli ko'pgina CASE vositalar tarkibiga kiritildi va ular ham o'z navbatida uning rivojlanishiga hissa qo'shdilar. Bugungi kunda «mohiyat-aloha» modelining yagona qabul qilingan standarti mavjud bo'limasa ham, umumiylar konstruksiyalarning majmuasi mavjud bo'lib, ko'pgina «mohiyat-aloha» modellarining asosida yotadi. Biz an'anaviy belgilash (tasvirlash) usullari bilan bir qatorda, loyihalash vositasi UML (Unified Model Language – yagonalashgan modellash tili) [63,88] tilining belgilaridan ham foydalanamiz. Chunki loyihalash vositalarining tarkibiga «mohiyat-aloha» modeli ham kiritilgan.

### «Mohiyat-aloha» modeli elementlari

«Mohiyat-aloha» modelining asosiy elementlari – *mohiyatlar, atributlar, identifikatorlar va aloqalardan iborat*.

**Mohiyat** – bu birorta predmet sohaga tegishli obyektdan iborat. Mohiyatlarga misol – XIZMATCHI Baxtiyor Salimov, MIJOZ 77345, BU-

YURTMA 0049 va MAXSULOT 59037. Bir xil turdag'i mohiyatlarning guruhlari mohiyatlarning sinflarini tashkil etadi. XIZMATCHI turdag'i mohiyatlarning sinfi hamma XIZMATCHI turidagi mohiyatlardan iborat. Mohiyatlarning sinflarini bosh harflar bilan belgilaymiz.

Mohiyatlar sinfi va mohiyat nusxasi tushunchalari orasidagi farqni aniqlashtirib o'taylik. *Mohiyatlar sinfi* deganda, mohiyatlarning majmuasi tushuniladi va u o'zining turi bilan tavsiflanadi. *Mohiyat nusxasi* esa, aniq bitta mohiyatni bildiradi, masalan, BUYURTMA 0049 va shu mohiyatning qiyomi bilan ifodalanadi. Demak, mohiyatlar sinfi mohiyat nusxalarining to'plamidan iborat deyishimiz mumkin.

Masalan, Talaba o'quv loyihasidagi RO'YHAT mohiyatining tarkibi – Zn (talaba reyting daftarchasining raqami), Familiya (familiyasi), Ism (ismi), O\_Ism (otasining ismi), K\_Sana (o'qishga kirgan sanasi), Fak\_N (fakultet raqami), Yon\_N (yo'nalish raqami), Kurs (kurs), Guruh\_N (guruh raqami), Pasp\_S (pasport seriyasi), Pasp\_N (pasport raqami).

Ana shu RO'YHAT mohiyatining ikkita nusxasi 7.1-rasmda keltirilgan:

Б005227	Б059022
Сангиров	Сайфиеv
Санжар	Бахтиёр
Чори ўли	Жўраевич
31.08.2006	31.08.2006
Механика-математика	Биология
AMAT	Биология
4	4
A1	B2
CA	CB
<b>5468990</b>	<b>0934073</b>

### 7.1-rasm.

Mohiyatlar atributlarga ega bo'lib, ularni xususiyat (properties)lar deb ham atashadi. Xususiyatlar mohiyatlarni tavsiflaydi. Misollar sifatida Hodimning\_ismi, tug'ilgan\_sanasi va ish\_stoji atributlarini keltirish mumkin. «Mohiyat-aloha» modelidagi berilgan sinfning hamma nusxalarini bir xil atributga ega, deb faraz qilinadi. «Mohiyat-aloha» modelining boshlang'ich ta'risi kompozit (composite) va ko'p qiy'marli (multi-valued) atributlarni o'z ichiga oladi. Kompozit atributga misol sifatida Manzil atributini olish mumkin, u Indeks, Shahar, Mavze, Ko'cha va Uy atributlar

guruhidan iborat. Ko'p qiymatli atributga misol Ustoz\_familiyası, bo'lishi mumkin, chunki har bitta talabaning bir qancha ustozi bo'ladi. *Atribut bir payida ham kompozit, ham ko'p qiymatli bo'lishi mumkin.* Masalan, Hudud\_kodi, telefon\_raqami atributlar guruhidan tuzilgan. Telefon kompozit atributi, ko'p qiymatli bo'lishi mumkin, bu esa bitta odamning bir nechta telefon raqamlarini saqlash imkoniyatini yaratadi. Ko'pgina «mohiyat-aloqa» modellarini amalga oshiruvchi vositalar bir qiymatli kompozit atributlarni inkor qiladi, ko'p qiymatli atributlarni esa mohiyatlarda aksanishini talab qiladi.

Mohiyatlarning nusxalarini *identifikator* (identifiers) – atributlarga ega bo'lib, nusxalar nomlanadi, ya'ni identifikatsiya qilinadi. Masalan, XIZMATCHI sinfi mohiyati nusxalarini Soliq\_to'lovchini\_aniqlovchi\_raqam yoki Xizmatchining\_familiyasi\_ismi\_sharifi atributlari bilan aniqlanadi. Ishga\_kirgan\_sanasi yoki Maoshi atributlari XIZMATCHI mohiyat sinfi nusxalarining identifikatorlari bo'la olmaydi, chunki ular xodimni bir qiymatli aniqlay olmaydi. Shunga\_o'xhash RO'YHAT sinfining mohiyati nusxalarini Zn yoki Familiya, Ism, O\_Ism atributlari orqali identifikatsiya qilinadi. BUYURTMA sinfining mohiyatlari esa Buyurtma\_raqami atributni orqali identifikatsiya qilinadi.

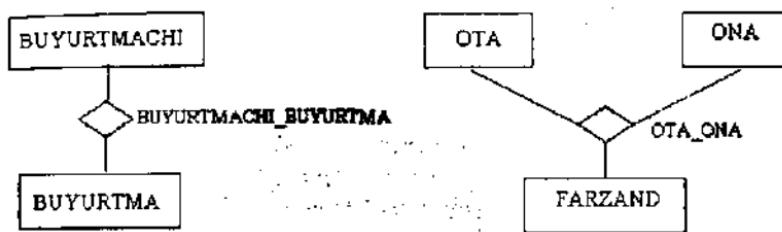
Mohiyat nusxasini identifikator bitta yoki bir nechta atributdan tashkil topishi mumkin. Identifikator *noyob* (unique) yoki *noyob bo'lmasligi* (non-unique) mumkin. Agarda identifikator noyob bo'lsa, uning qiymati mohiyat nusxasining faqat bittasini aniqlaydi. Agarda identifikator noyob bo'lmasa, uning qiymati mohiyat nusxalarining qandaydir to'plamini aniqlaydi. Masalan, Zn identifikator noyob, Familiya, Ism, O\_Ism identifikatorlari noyob emas, chunki xuddi shunday familiya, ism va otasining ismidagi ikkita va undan ortiq odam bo'lishi mumkin.

Bir nechta atributlardan iborat identifikatorlar *kompozit identifikatorlar* deyiladi. Misol, Hudud\_kodi, telefon\_raqami identifikatori.

Mohiyatlarning o'zaro munosabatlari *aloqalar* (relationships) orqali ifodalanadi. «Mohiyat-aloqa» modellarini, aloqalar sinfini va aloqalarning nusxalarini o'z ichiga oladi. *Aloqalar sinfi* (relationships classes) – mohiyatlar sinflari orasidagi o'zaro munosabatlardan, *aloqalar nusxalari* (relationships instances) esa, mohiyatlar nusxalarini orasidagi o'zaro munosabatlardan iborat. Aloqalarning ham xususiyatlari bo'lishi mumkin.

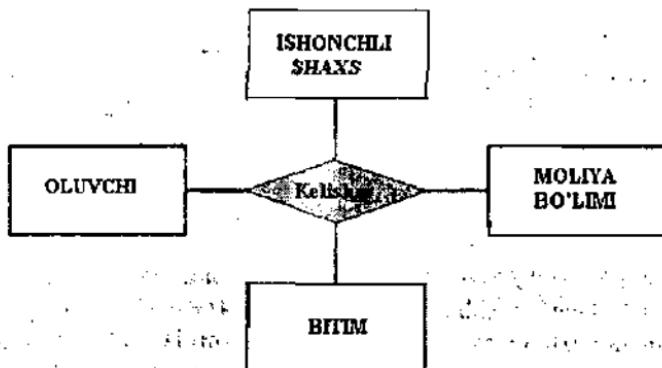
Aloqalar sinfi bir nechta mohiyatlar sinfi bilan munosabatda bo'lishi mumkin. Aloqada ishtiroy etuvchi mohiyatlar soni, *aloqa darajasini* (relationships degree) belgilaydi. 7.2-rasmida aloqa darajasi ikki va uch bo'lgan holatlar keltirilgan. BUYURTMACHI\_BUYURTMA aloqasining darajasi 2, chunki unda ikkita mohiyat sinflari BUYURTMACHI va BUYURTMA qatnashgan. OTA\_ONA aloqasining darajasi 3, chunki unda uchta

mohiyatlar sinflari OTA, ONA va FARZAND ishtirok etgan. Aloqa darajasi ikkiga teng bo'lgan holatlar juda keng tarqalgan bo'lib, ularni *binar* (ikkilik) aloqalar (binary relationships) deb ataladi. Aloqaning darajasi 3 bo'lsa, *uch tomonlama aloqa* (ternary) deb ataladi.



7.2-rasm.

To'rtinchı darajali aloqa *to'rt tomonlama aloqa* (quaternary) deb ataladi. 7.3-rasmda to'rt tomonlama kelishuv aloqasi keltirilgan. Unda to'rtta qatnashuvchi mohiyatlar, ya'ni OLUVCHI, ISHONCHLI\_SHAXS, MOLIYA\_BO'LIMI va BITIM qatnashgan. Bu aloqa bo'yicha, oluvchi ishonchli shaxs tomonidan maslahat olib va moliya bo'limi qo'llab-quvvatlagan holda, bitim tuzadi.



7.3-rasm.

### Binar aloqalarning uch turi

Binar aloqalar uch turga bo'linadi. 7.4-rasmda bitta mohiyat nusxasi boshqa turdag'i bitta mohiyat nusxasi bilan 1:1 turdag'i aloqa orqali bog'langan. TALABA-KOMPYUTER aloqasi TALABA sinfiga bitta

mohiyatni KOMPYUTER sinfidagi bitta mohiyat bilan bog'lagan. Bu bog'lanishga asosan, har bir talabaga faqat bitta kompyuter biriktirib qo'yilgan va bitta kompyuter faqat bitta talabaga tegishli, degan ma'no yotadi.



7.4-rasm. 1:1 aloqa

7.5-rasmda ikkinchi turdag'i binar aloqaga misol keltirilgan. Bu aloqa «birga-ko'p» deb ataladi. Masalan, hunarmandlikdagi USTA-SHOGIRD aloqasini olsak, USTA mohiyat sinfidagi yagona nusxa SHOGIRD mohiyat sinfidagi bir nechta nusxalar bilan bog'langan. Ushbu bog'lanishga nisbatan, ustaming Nta shogirdi bor, bu shogirdlarning hammasini ustozi ana shu bitta USTA, degan ma'no yotadi.

Bundan tashqari  $1 : N$  shakldagi yozuvning ham o'z ma'nosi bor.  $1$  raqami USTA joylashgan tomonda,  $N$  esa SHOGIRD joylashgan tomonda turibdi. Agarda teskarisi yozilganda edi, shogird bitta va uning esa bir vaqtida bir nechta ustozi bor, degan ma'no hosil bo'lar edi. Bu esa haqiqatan yiroq.



7.5-rasm. 1:N aloqa

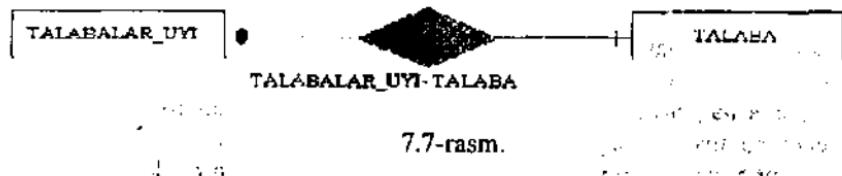
Binar aloqaning uchinchi xili  $N : M$  bo'lib, «ko'plikdan-ko'plikka» ma'nosini anglatadi. 7.6-rasmdagi TALABA-TO'GARAK aloqasi bo'lib, TALABA mohiyatlar sinfiga tegishli nusxalarni TO'GARAK mohiyatlar sinfiga tegishli nusxalar bilan bog'lagan. Ushbu aloqa, bitta talaba bir nechta to'garakka a'zo bo'lishi va bitta to'garakda bir nechta talabalar bo'lishi mumkinligini anglatadi.



## 7.6-rasm. N:M aloqa

Romb ichidagi aloqalarni sonini ifodalovchi raqamlar, har bir tomonagi mohiyatlarning maksimal nusxalarini bildiradi. Bu chegaralar *maksimal kardinal sonlar*, bunday ikkita chegaradan iborat ikki tomonli aloqa birligi *aloqaning maksimal kardinalligi* (maximum cardinality) deb ataladi. Masalan, 7.5-rasmdagi aloqaning maksimal kardinalligi 1 : N. Lekin albatta, bu turdag'i aloqlardagi kardinallik boshqacha ham bo'lishi mumkin. Masalan, **VOLEYBOL\_KOMANDA** va **O'YINCHI** mohiyatlari orasidagi aloqa 1:6, bu voleybol komandasida maydonda bir paytda oltitadan ko'p o'yinch'i bo'lishi mumkin emasligini bildiradi.

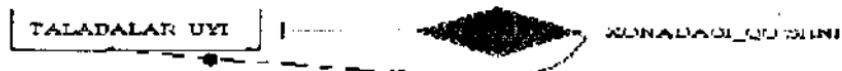
Minimal kardinallikni ko'rsatishning bir nechta usullari mavjud. Ular dan biri 7.7-rasmda keltirilgan. Aloqada qatnashishi majbur bo'lgan mohiyat tomonagi aloqaga perpendikular chiziqcha tortiladi, aloqada qatnashishi mumkin, lekin shart bo'lmasan mohiyat tomonagi aloqaga tuxumsimon yassi shakl (oval) joylashtiriladi. 7.7-rasm, **TALABALAR\_UYI** mohiyati kamida bitta **TALABA** mohiyati bilan bog'langanini, lekin **TALABA** mohiyati **TALABALAR\_UYI** mohiyati bilan bog'lanishi shart emasligini bildiradi.



7.7-rasm.

**TALABALAR\_UYI-TALABA** aloqasiga qo'yilgan to'liq cheklashlar ning ma'nosi quyidagicha, **TALABALAR\_UYI** mohiyatinining minimal kardinalligi birga va maksimal kardinalligi soni «ko'p» **TALABA** mohiyatidan iborat. **TALABA** nolga teng minimal kardinallikka va maksimal kardinalligi **TALABALAR\_UYI** mohiyatinining bitta nusxasidan iborat.

Bitta sinfga tegishli mohiyatlar orasida ham aloqalar bo'lishi mumkin. Misol uchun **TALABA** mohiyatlar sinfi uchun **XONADAGI\_QO'SHNI** aloqasini aniqlash mumkin. Bunday aloqaga misol 7.8-rasmda keltirilgan. **XONADAGI\_QO'SHNI** aloqasining nusxalari Jalil, Halim, Salim yoki boshqa ixtiyoriy talaba bo'lishi mumkin. Bitta sinfga tegishli mohiyatlar orasidagi aloqalar *rekursiv aloqalar* (recursive relationships) deb ataladi.



### 7.8-rasm.

Yuqorida keltirilgan uch xildagi binar aloqlarni ba'zan «*MAVJUD*» *aloqa turi* yoki *egalik aloqalari* (HAS-A relationships) deb atashadi. Buning sababi, bitta mohiyatning boshqa mohiyat bilan aloqasi *mavjud* (has) ligini, ya'ni aloqa borligini bildiradi. Masalan, talabaning kompyuteri bor, ustanning shogirdi bor, to'garakka yozilgan talabalar bor.

### «*Mohiyat-alоqa*» diagrammalari

Yuqoridagi 7.4 – 7.6-rasmlardagi chizmalarni yoki mohiyatlarni ular orasidagi aloqalar hamda aloqalarning maksimal kardinalligi ko'rsatilgan holda maxsus chizma tarzida ifodalash «*mohiyat-alоqa*» diagrammalari yoki *ER-diagrammalar* (entity-relationship diagrams, ER-diagrams) deb ataladi. Bu diagrammalar standartlashtirilgan, lekin juda qat'iy ravishda emas. Bu standartga asosan *mohiyatlar* – to'g'ri to'rburchaklar, *atributlar* – ellipsoidlar, *aloqalarning nomi* – romblar ichida tasvirlanadi. Ba'zan birlik aloqalar —+ tarzda, ko'plik aloqalar esa —← shaklda belgilanadi. Keyinchalik ko'pgina mualliflar ana shunday diagrammalarning o'zlarini yaratgan naqllarini taklif qildilar (Dj. Martin notatsiyalari, IDEFIX belgilari, Barker notatsiyasi va h.k.). Bundan tashqari, bir xildagi notatsiyani amalga oshirgan programmalar ham o'z imkoniyatlari bo'yicha turliha bo'lishi mumkin. Aslini olganda, «*mohiyatlar-alоqalar*» diagram-masining hamma naqllari rasm matnli tavsifdan yaxshi, deyilgan fikrga asoslanadi. Hamma ana shunday diagrammalar obyekt *mohiyatining* grafik tasviridan, ularning *atributlari* va *mohiyatlar* orasidagi aloqalardan foydalananadilar.

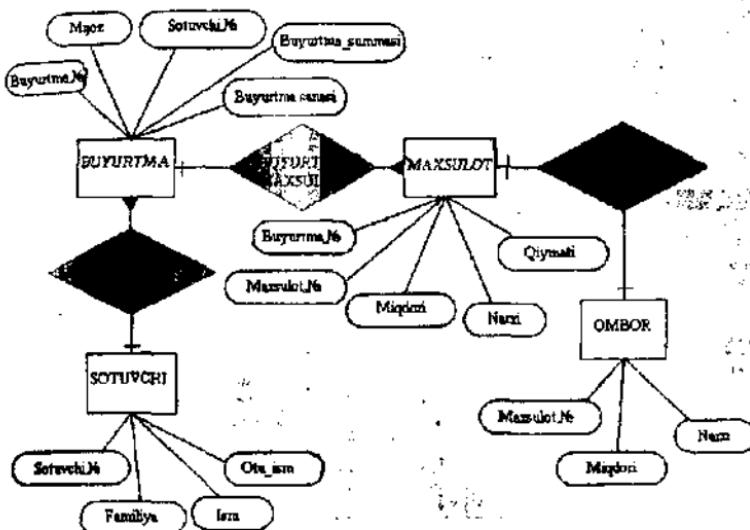
«*Mohiyat-alоqa*» diagrammasining elementlaridan tuzilgan, mahsulotning sotuvchigacha bo'lgan berilganlar bazasining modeliga misol 7.9-rasmda keltirilgan.

Undagi diagrammadan ko'rinish turibdiki, BUYURTMA va MAXSULOT mohiyatlari mos ravishda beshtadan atributga – Buyurtma\_No, Mijoz, Sotuvchi\_No, Buyurtma\_summasi, Buyurtma\_sanasi; Buyurtma\_No, Mahsulot\_No, Miqdori, Narxi, Qiymati; SOTUVCHI mohiyati to'rtta atributga – Sotuvchi\_No, Familiya, Ism, Ota\_ismi; OMBOR mohiyati esa uchta atributga ega – Maxsulot\_No, Miqdori, Narxi.

BUYURTMA-SOTUVCHI aloqasining maksimal kardinalligi N:1, BUYURTMA-MAXSULOT aloqasining maksimal kardinalligi 1:N, MAXSULOT-OMBOR aloqasining maksimal kardinalligi esa 1:1.

Agarda mohiyatning atributlari ko'p bo'lsa, tasvir ancha murakkablasib, qabul qilish qiyinlashadi. Bunday hollarda mohiyatlarga tegishli atributlar alohida ro'yxatlarda keltiriladi. Ko'pgina CASE-vositalar bunday

atributlarni ochiluvchi oynalar orqali beradi, masalan 7.10-rasmida IDEFIX standartidagi mohiyatlar orasidagi aloqalarga misol keltirilgan.



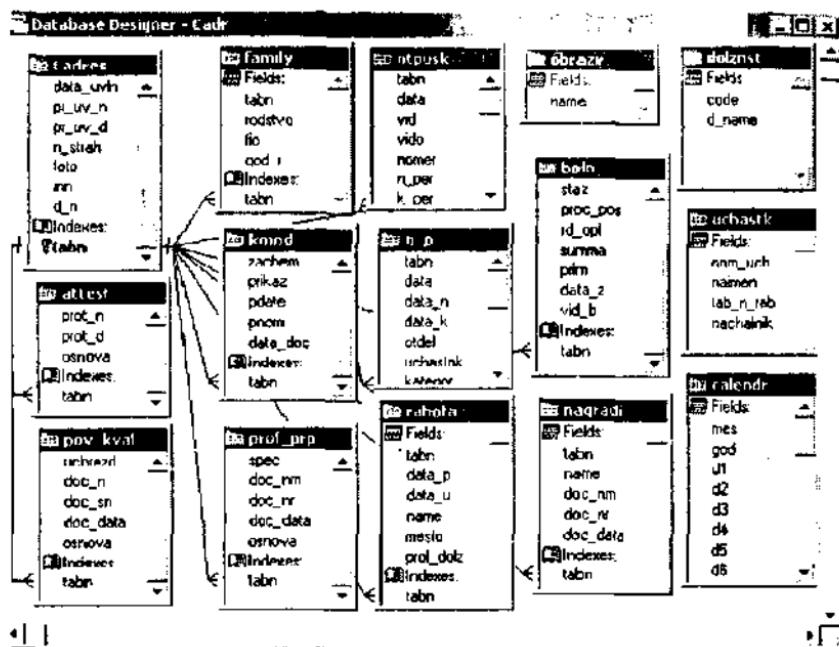
7.9-rasm. «Mohiyat-aloqa» diagrammasidagi usullardan biri.

### Zaif va kuchli mohiyatlar

«Mohiyat-aloqa» modelida mohiyatning ikkita maxsus turi aniqlangan bo‘lib, ulardan biri *zaif mohiyat* (weak entity) deb ataladi. *Zaif mohiyatlarga*, mavjudligi boshqa mohiyatlarga mantiqan bog‘liq bo‘lgan mohiyat kiradi. Mavjudligi boshqa mohiyat turiga bog‘liq bo‘lmagan mohiyat turi, *kuchli mohiyat* (strong entity) deb ataladi.

Zaif mohiyatlarga qanday mohiyatlar kirishini aniqlashtirish uchun XODIM va TOBE\_XODIM mohiyatlar sinflaridan tuzilgan berilganlar bazasini ko‘raylik.

Faraz qilaylik, o‘rnatalgan ish tartibiga mos ravishda XODIM mohiyat nusxasi TOBE\_XODIM sinfidagi hech bir mohiyat bilan aloqa o‘rnatmagan holda mavjud bo‘lishi mumkin, lekin TOBE\_XODIM mohiyati esa XODIM sinfidagi mohiyatlar bilan aloqasiz mavjud bo‘lishi mumkin emas. U holda, TOBE\_XODIM mohiyati zaif bo‘ladi. Bu esa, TOBE\_XODIM mohiyatiga tegishli bo‘lgan berilgantlar bazasida, ushbu mohiyat faqtgina XODIM sinfidagi birorta mohiyat bilan aloqasi bo‘lsagina, paydo bo‘lishi mumkinligini bildiradi.



7.10-rasm.

7.11-rasmida ana shu vaziyat keltirilgan bo'lib, zaif mohiyat burchaklari yassilangan to'rtburchakda tasvirlangan.



7.11-rasm.

Mohiyatni zaiflar toifasiga kiritish uchun, u *mantiqan* boshqa mohiyatga bog'liq bo'lishi kerak. Ayrim holatlarda USTOZ va TALABA mohiyatlar orasidagi aloqadan kelib chiqqan holda, TALABA mohiyati zaif deb hisoblanadi, chunki talabaning ustozi bo'lishi kerak va u berilganlar bazasiga faqat USTOZ sinfiga tegishli mohiyatlardan biri bilan aloqada bo'lishi kerak, deb yuritiladi. Lekin talaba fizik va mantiqiy jihatdan ham ustozi bilan bog'lanmagan, shuning uchun ham TALABA mohiyati kuchli mohiyat hisoblanadi. Talaba ish tartibi yo'l qo'ymasa ham, mantiqiy jihatdan ustozsiz ham mavjud bo'lishi mumkin.

Ushbu tasdiqni yanada yaxshi namoyish qilish uchun bir nechta misolarni ko'ramiz. Faraz qilaylik, ta'limga oid berilganlar bazasi FAN\_NOMI va REYTING\_BALI mohiyatlarini o'z ichiga olgan bo'lsin va ular orasida aloqa o'rnatilgan bo'lsin (7.12-rasm). REYTING\_BALI mohiyatining ixtiyoriy nusxasi FAN\_NOMI sinifidagi mohiyat bilan albatta aloqada bo'lishi mantiqan shart, chunki reyting bali o'z holicha mavjud bo'la olmaydi, lekin ixtiyoriy fan reyting balisiz ham mavjud. Demak, FAN\_NOMI mohiyati kuchli, REYTING\_BALI mohiyati esa zaif mohiyat.



7.12-rasm.

Endi BEMOR va RETSEPT mohiyatlari orasidagi 7.13-rasmdagagi diagrammaga e'tibor beraylik. Retsept bemorsiz mantiqan mavjud emas. RETSEPT mohiyatining minimal kardinal soni birga teng bo'lsa ham, u BEMOR mohiyatisiz mavjud bo'la olmaydi. Demak, RETSEPT zaif, BEMOR esa kuchli mohiyat.



7.13-rasm.

BUYURTMA va SOTUVCHI mohiyatlari orasidagi aloqani ko'rib chiqaylik (7.14-rasm). SOTUVCHI mohiyatining minimal kardinalligi 1, bu mantiqiy zaruratdan emas, balki ish tartibidan kelib chiqqan holda shunday bo'ladi. Buyurtmani huquqiy shaxs ham berishi mumkin. Shunday ekan, demak, BUYURTMA mohiyati SOTUVCHI mohiyatisiz ham mavjud bo'lishi mumkin. Xulosa, BUYURTMA va SOTUVCHI mohiyatlarining ikkalasi ham kuchli.



7.14-rasm.

Zaif mohiyatlarni ba'zan voris, bog'liq yoki tobe mohiyat, kuchli tur-

dagi mohiyatlarni esa *ajdod*, *ega-mohiyat* yoki *dominant mohiyat* deb atashadi.

«Mohiyat-aloqa» modelida zaif mohiyatlarning o'zi alohida turga ega, ularni *identifikatsiyalangan bog'liq mohiyailar* (ID dependent entities) deb atashadi. Bular shunday mohiyatlarki, ularning identifikatorlari boshqa mohiyatni identifikatorlarini saqlaydi. Masalan, ko'p qavatli uylardagi UY va XONADON mohiyatlari orasidagi aloqani ko'rib chiqaylik (7.15-rasm).



7.15-rasm.

UY mohiyatining identifikatori Uy\_raqami atributi bo'lsa, XONADON mohiyatining tarkibli identifikatori esa (Uy\_raqami, Xonadon\_raqami) atributlaridan iborat bo'ladi. XONADON mohiyatining identifikatori UY mohiyatini identifikatoridan (Uy\_raqami) iborat bo'lgani uchun, XONADON mohiyati UY mohiyatiga *identifikatsiyalash* bo'yicha bog'liq. Boshqacha aytganda, XONADON mohiyati ham mantiqiy, ham fizik jihatdan uysiz mavjud bo'la olmaydi.

Xuddi shunday tarzda, Talabalar uyi bilan xonalar orasidagi aloqada ham, xonalar Talabalar uyiga identifikatsiyalangan tarzda bog'liq.

### Bir va ko'p qiymatli atributlar

*Bir qiymatli atribut* deganda, ma'lum bir turdag'i mohiyat nusxasining bitta qiymatini saqlovchi atribut tushuniladi. Ko'pgina atributlar bir qiymatli bo'ladi. OLUVCHI mohiyatining har bir alohida nusxasi uchun doimiy ravishda Oluvchi\_raqami atributida yagona qiymat mavjud, masalan X004. Shuning uchun Oluvchi\_raqami atributi bir qiymatli bo'ladi.

*Ko'p qiymatli atribut* deganda, ma'lum bir turdag'i mohiyat nusxasining bir nechta qiymatini saqlovchi atribut tushuniladi. Ba'zi bir atributlar mohiyatning har bir nusxasi uchun bir nechta atributdan iborat bo'lishi mumkin. Masalan, OLUVCHI mohiyatida Telefon\_raqami atributining bir nechta raqami bo'lishi mumkin. Oluvchi\_raqami atributidagi X004 raqamli mahsulot oluvchida 233-45-55, 233-45-56 va 233-25-39 raqamli telefonlar bo'lishi mumkin. Demak, Telefon\_raqami atributi ko'p qiymatli bo'ladi. Ko'p qiymatli atributlardagi qiymatlar soni albatta ikki tomonidan chegaralangan bo'lishi shart.

## Mohiyatlarning ostki va ustki turlari

Ayrim mohiyatlarda zarur bo'lmagan atributlar ham bo'sadi, bunday mohiyatlar *ostki turlar* (subtypes) orqali ifodalanadi. Masalan, *Mijoz\_raqami*, *Mijoz\_nomi* va *To'lov\_miqdori* atributlariga ega bo'lgan MIJOZ mohiyatini ko'rib chiqaylik. Faraz qilaylik, mijoz fizik shaxs, jamiyat yoki vazirlik va bu mijozlarning turiga nisbatan ba'zi bir qo'shimcha ma'lumotlarni ko'rsatish zarur bo'lsin. Bu ma'lumotlar quyidagicha bo'lsin:

**JISMONIY\_SHAXS** – *Familiya\_Ism*, *Soliq\_to'lovchining\_shaxsiy\_raqami*, *Manzili*

**JAMIYAT** – *Nomi*, *Hisob\_raqami*, *Soliq\_to'lovchining\_shaxsiy\_raqami*, *Manzili*

**VAZIRLIK** – *Nomi*, *Hisob\_raqami*, *Soliq\_to'lovchining\_shaxsiy\_raqami*, *Manzili*.

Keltirilgan atributlarni hammasini MIJOZ mohiyati tarkibiga kiritilgandan keyingi atributlar tarkibi 7.16-rasmida keltirilgan:

*Mijoz\_raqami*  
*Mijoz\_nomi*  
*To'lov\_miqdori*  
*Familiya\_Ism*  
*Soliq\_to'lovchining\_shaxsiy\_raqami*  
*Manzili*  
*Nomi*  
*Hisob\_raqami*

7.16-rasm.

Lekin bu holda, MIJOZ mohiyatidagi ayrim atributlar keraksiz bo'lib qoladi. Masalan, jamiyat va vazirlik uchun *Familiya\_Ism* atributi kerak emas yoki fizik shaxs uchun *Nomi* va *Hisob\_raqami* atributlari kerak emas, shunday ekan, bu atributlar qiymatga ega bo'la olmaydi.

Aniq vaziyatga eng yaqin holat, bu MIJOZ mohiyatini uchta ostki turga ajratishdan iborat.

MIJOZ mohiyati tarkibi – *Mijoz\_raqami*, *Mijoz\_nomi* va *To'lov\_miqdori*;

**JISMONIY\_SHAXS** tarkibi – *Familiya\_Ism*, *Soliq\_to'lovchining\_shaxsiy\_raqami*, *Manzili*;

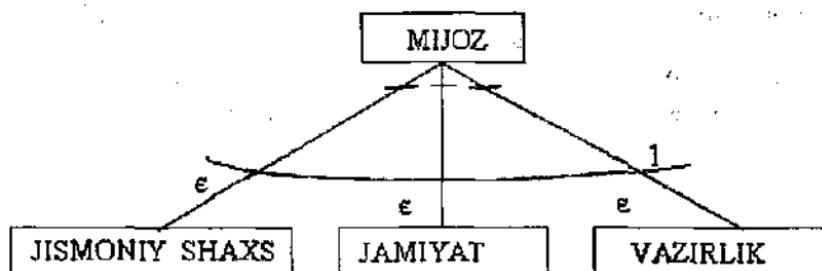
**JAMIYAT** tarkibi – *Nomi*, *Hisob\_raqami*, *Soliq\_to'lovchining\_shaxsiy\_raqami*, *Manzili*;

VAZIRLIK tarkibi – Nomi, Hisob\_raqami, Soliq\_to'lovchining\_shaxsiy\_raqami, Manzili.

7.17-rasmida JISMONIY\_SHAXS, JAMIYAT va VAZIRLIK mohiyatlari MIJOZ mohiyatining ostki turlari sifatida tasvirlangan. MIJOZ mohiyati esa JISMONIY\_SHAXS, JAMIYAT va VAZIRLIK mohiyatlari uchun *ostki tur* (supertype) hisoblanadi. Aloqa chiziqlari yonidagi e belgisi JISMONIY\_SHAXS, JAMIYAT va VAZIRLIK mohiyatlari MIJOZ mohiyatining ostki turlari ekanligini bildiradi. I raqamli egri chiziq MIJOZ mohiyati bitta va faqat bitta ostki turga tegishli ekanligini bildiradi. Bu esa, ostki turlar o'zaro bir-birini o'rmini bosadi (o'zaro almashadi) va ulardan faqat bittasi zarur, degan ma'noni beradi.

### Atribut domeni

Bir yoki bir nechta atributning qabul qilishi mumkin bo'lgan qiymatlar to'plami, atribut *domeni* deb ataladi. Har bir atribut domen deb atalmish, qiymatlar to'plami bilan bog'langan.



7.17-rasm.

Domen, atribut qabul qilishi mumkin bo'lgan, hamma qiymatlarni aniqlaydi. Domenning bu ta'rifni berilganlarning relatsion modelida keltirilgan ta'rifga o'xshash. Masalan, ko'chmas mulk obyektidagi xonalar soni Itadan 25tagacha bo'lishi mumkin desak, demak, tegishli atributning turidagi qiymatlar birdan yigirma beshgacha bo'lgan butun sondan iborat bo'ladi. Turli atributlar bitta domendenan birgalikda foydalaniishi mumkin. Domenlar, shuningdek, boshqa domenlarning kombinatsiyasidan ham tuzilishi mumkin. Berilganlarning to'liq yaratilgan modeli har bir atributning domenlarini o'z ichiga olgan holda ER-modelda tasvirlanadi.

Bo'linmas atributlar *oddiy*, bo'linuvchi atributlar esa *tarkibili* hisoblanadi. Masalan, Maosh atributi oddiy atribut, Manzil atributi esa tarkibili atribut, chunki uni bir nechta tobe atributlarga ajratish mumkin.

### Talabalar o'quv loyihasini modellashdirish

Berilganlarni modellashdirish uchun vazifani predmet sohani tahlil qilishdan boshlash kerak. Biz, predmet soha sifatida respublikamiz huddidagi ixtiyoriy oliy ta'limga muassasasini olishimiz mumkin. Chunki, yaratilgan modelimizni ixtiyoriy oliy ta'limga muassasiga amaliyatda tafbiq etish mumkin.

Oliy ta'limga ikki bosqichdan iboratligini inobatga oлган holda, biz bakalavriatda tahlil oluvchi talabalar misolida o'quv loyihamizning «mohiyat-aloqa» modelini yaratamiz. Oliy o'quv yurti tuzilishining talabalar bilan bog'liq jihatlarining eng asosiyalarini «mohiyat-aloqa» modeli tarkibiga kiritamiz.

Yuqoridagi maqsadga binoan, oliy o'quv yurtidagi bakalavriat talabari ma'lum bir yo'nalish bo'yicha ta'limga oladilar. Yo'nalishlar ro'yxati, Oliy ta'limga Davlat ta'limga standarti - Oliy ta'limga yo'nalishlari va mutaxassisliklari klassifikatoriga asosan belgilanadi. Har bir yo'nalish oliy ta'limga muassasasidagi birorta fakultet bilan bog'langan. Bakalavriatdagi talabalar 4 yil tahlil oladilar. Ta'limga olish jarayoni sakkizta semestrni tashkil qiladi. Har bir semestrda, davlat ta'limga standartiga asosan, har bir yo'nalish hamda ma'lum fanlar bo'yicha darslar o'tiladi va talabalarning reyting natijalari qayd etiladi. Har bir fan bo'yicha kamida bitta professor-o'qituvchi dars jarayonini olib boradi. Talabalar yo'nalishlar bo'yicha kurslari kamida bitta guruhi tashkil etadi. Talabalar grant yoki shartnomaga-to'lov asosida tahlil oladilar. Talabalarning o'qishga kirgan sanalari rektorning buyrug'i asosida qayd etiladi. Talabalar o'z uylarida, talabalar uyida yoki boshqa joyda yashab o'qishlari mumkin. Har bir o'quv yilining boshida, talabalarni avtomatik tarzda kursdan kursga ko'chirish va bakalavriyatni bitirgan talabalarni tizim arxiviga (maxsus bazaga) o'tkazish kerak. Birinchi kursga qabul qilingan talabalarни «Abituriyent» berilganlar bazasidan avtomatik tarzda o'tkazish kerak bo'lsin.

Ana shu to'liq ma'lumotlarni tahlil qilish asosida, biz modelimizdagи asosiy omillarni turlicha nisbiy yondoshuv asosida ajratib olamiz.

*Birinchi nisbiy yondashuv.* Biz fakultetlarni asosiy omil sifatida olsak, fakultetlar kesimida model qurishimiz kerak bo'ladi. Bu albatta eng qulay, lekin bazalarning soni birmuneha ko'p bo'ladi. Ushbu omil asosida yaratilgan modelning ustunligi, har bir fakultet dekanati faqtgina o'zining talabalari ustidagi amallarni bajarish imkoniyatiga ega bo'ladi. Oliy o'quv

yurtidagi umumiy natijalarni boshqa fakultet dekanati ko'ra olmaydi va umumiy tahlil ham qila olmaydi, faqatgina o'ziga tegishli fakultet talabalari bo'yicha o'quv yili yoki semestr natijalarini olish va tahlil qilish imkoniyatiga ega bo'ladi. Ushbu yondashuv ko'p miqdordagi talabalarga ega bo'lgan oliy o'quv yurtlari uchun yaratiladigan berilganlar bazasini boshqarish tizimining modeli, deb qabul qilsak bo'ladi.

*Ikkinci nisbiy yondashuv.* Talabalarini asosiy omil sifatida olsak, talabalar modelning markaziy qismiga aylanadi va shunga asosan fakultetlar yordamchi mexanizm sifatida ishtirok etadi. Bu yondashuvda talabalarini hammasini yagona markaziy tuzilmaga jamlash va qolgan tuzilmalarni yordamchilar sifatida qarash mumkin. Talabalar modeldagi yagona markaziy tuzilmani tashkil etgani uchun, har bir dekanat oliy o'quv yurtidagi barcha talabalar bo'yicha o'quv yili yoki semestr natijalarini olish va taqposlash imkoniyatiga ega bo'ladi.

Talabalar o'quv loyihasini modellashtirish uchun, biz ikkinchi nisbiy yondashuvni tanlagan holda, unga yuqoridagi tahlil natijasida kirishi kerak bo'lgan ma'lumotlarning ham ba'zilarini kiritmaymiz. Lekin bu torroq doiradagi «mohiyat-aloqa» modeli uni yaratish jarayoniga aslo salbiy ta'sir etmaydi.

### Mohiyatlar

«Mohiyat-aloqa» modelini qurishni nomzod mohiyatlarni aniqlashdan boshlash kerak. Hujjatlarda yoki suhbatlarda mohiyatlar odatda ot sifatida taqdim etiladi, ya'ni joy, insonlar, voqealar, asbob-uskunalar va h.k. Yuqoridagi ma'lumotlarni berilgan modeliga nisbatan, eng muhim so'z yoki ifoda nuqtayi nazaridan tahlil qilish natijasida, biz Talabalar o'quv loyihasida ishtirok etuvchi quyidagi mohiyatlar ro'yxatini tuzamiz:

- RO'YHAT
- BAHOLAR
- FAKULTETLAR
- YO'NALISH
- FANLAR.

Berilganlarni modellashtirish jarayonida fan qancha miqdorni tashkil etsa, san'at ham shu miqdorni tashkil etadi. Shuning uchun biz qabul qilgan qarorimiz, ana shunday yechimlardan birini tashkil etadi xolos. Muqobil variantlar orasidan to'g'risini tanlash uchun qo'yilgan talablarga javob beruvchisini hamda berilganlar tizimning tuzilishiga qanday ta'sir ko'rsatishini aniqlash zarur. Ba'zida mohiyatlarni atributlarini ko'rib chiqish juda foydalı bo'ladi. Masalan, agarda mohiyatlarda uni aniqlovchi atributdan tashqari hech qanday atribut ishtirok etmasa, bunday mohiyatlardan voz

kechish kerak. Shu nuqtayi nazardan hosil qilgan har bir mohiyatning atributlarini ko'rib chiqamiz.

RO'YHAT mohiyati ushbu atributlardan iborat bo'lsin: Zn (talabaning reyting daftarchasining raqami), Familiya (familiyasi), Ism (ismi), O\_Ism (otasining ismi), K\_Sana (o'qishga kirgant sanasi), Fak\_N (fakultetning raqami), Yon\_N (yo'nalishning raqami), Kurs (kurs), Guruh\_N (guruh nomeri), Pasp\_S (pasport seriyasi), Pasp\_N (pasportining nomeri). Bu atributlarning hammasi talaba xususidagi ayrim berilganlarni o'z ichiga oladi.

Biz bu erda shuni alohida ta'kidlashimiz kerakki, atributlarning soni mohiyatning ma'nosidan kelib chiqqan holda qanchalik ko'p bo'lsa, shunchalik ko'p so'rovchlari tizimda tashkillashtirish imkoniyatiga ega bo'lamiz. Bu esa tizimni amaliyotdagi tub maqsadiga erishishini ta'minlaydi.

BAHOLAR mohiyati Zn (talabaning reyting daftarchasining seriyasi va raqami), Semestr (semestr raqami), Fan\_N (fanlar raqami), Ball (ma'lum fanga qo'yilgan ball), Sana\_Kir (fan bo'yicha qo'yilgan ballni tizimga kiritilgan sanasi), Ustoz (fan bo'yicha baholagan ustozning familiyasi, ismi va sharifi) atributlaridan tuzilgan bo'lsin.

FAKULTETLAR mohiyati ikkita atributlardan – Fak\_N (fakultet raqami) va Fakultet (fakultet nomi) iborat bo'lsin. Har bir fakultet FAKULTETLAR mohiyatini hamda fakultetlar nomini yagona tarzda aniqlovchi fakultet raqami atributining qiymatlaridan iborat. Quyidagi ikkita mohiyat ham shunday vazifani bajarish uchun xizmat qiladi.

YO'NALISH mohiyati ham ikkita atributlardan iborat bo'lsin – Yon\_N (yo'nalish raqami) va Yo'nalishlar (yo'nalishlar nomi).

Oxirgi FANLAR mohiyati ham ikkita atributlardan – Fan\_N (fan raqami) va Fan\_Nomi (fan nomi) tuzilgan bo'lsin.

Munosabatning darajasi undagi atributlar soniga bog'liq. Talabalar o'quv loyihamizdag'i Ro'yhat munosabatida 11ta atribut bor, demak, uning darajasi 11ga teng. Baholar munosabatida oltita atribut bor, demak, uning darajasi oltiga teng. Qolgan Fakultetlar, Fanlar va Yo'nalishlar munosabatlarining darajalari ikkiga teng. Munosabatning darajasi uning har bir kortejida nechtadan elementni saqlashini bildiradi.

Talabalar o'quv loyihasidagi relatson chizmalar quyidagicha:

Ro'yhat (ZN, FAMILIYA, ISM, O\_ISM, K\_SANA, FAK\_N, YON\_N, KURS, GURUH\_N, PASP\_S, PASP\_N)

Baholar (ZN, SEMESTR, FAN\_N, BALL, SANA\_KIR, USTOZ)

Fakultetlar (**FAK\_N**, FAKULTET)

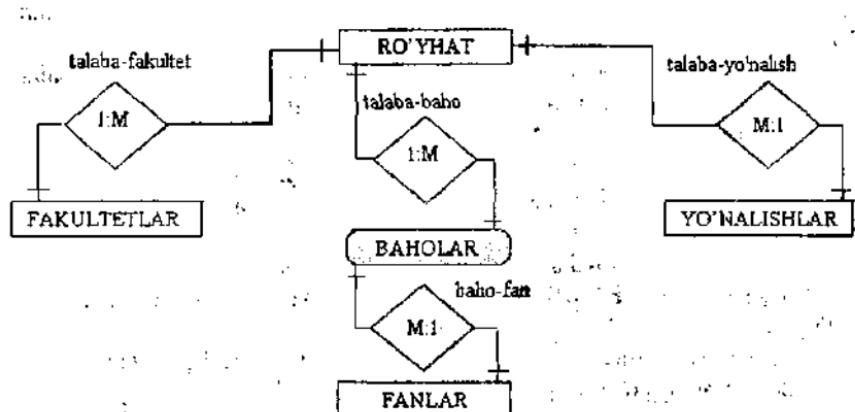
Yo'nalish (YON\_N, YO'NALISHLAR)

Fanlar (FAN\_N, FAN\_NOMI)

## Aloqalar

RO'YHAT mohiyati markaziy mohiyat sifatida ishtirok etmoqda. Shuning uchun uning boshqa mohiyatlar bilan aloqasini ko'rib chizamiz. Avval RO'YHAT va BAHOLAR mohiyatlari orasidagi aloqalarni ko'rib chiqaylik. RO'YHAT mohiyatida barcha talabalar jamlanadi. FAMILIYA atributi talabalarning familiyasidan iborat bo'ladi. Ushbu atributda turli familiyadagi talabalar yozilgan paytda nomzod kalit bo'lishi mumkin. Lekin bitta o'xshash familiya paydo bo'lsa, bu atributni nomzod kalit sifatida ishlatalib bo'lmaydi. Har bir talabaning reyting daftarchasi mavjud bo'lib, bu daftarchadagi seriya va raqamlar noyob bo'lgani uchun Zn atributining nusxasi har bir talabani bir qiymatli identifikasiyalaydi. Xuddi shunday vazifani Zn atributi BAHOLAR mohiyati uchun ham bajaradi. Lekin BAHOLAR mohiyatida Zn atributi tashqi identifikator, ya'ni talabalarning har birini RO'YHAT mohiyatidan bir qiymatli aniqlaydi. RO'YHAT va BAHOLAR mohiyatlari orasidagi aloqa 1:M bo'ladi. Bitta talaba j ta fan bo'yicha baho oladi,  $1 \leq j \leq M$  bo'ladi. RO'YHAT mohiyati RO'YHAT mohiyatiga identifikasiyalangan bog'liq turdag'i zaif mohiyat.

RO'YHAT mohiyati va FAKULTETLAR hamda YO'NALISHLAR mohiyatlari orasidagi aloqalar esa M:1 tarzda bo'ladi. Fak\_N atributi FAKULTETLAR mohiyatini, Yon\_N atributi esa YO'NALISHLAR mohiyatini noyob tarzda identifikasiyalaydi. Fak\_N atributining har bir nusxasi Fakultet atributining bir nusxasini, Yon\_N atributining har bir nusxasi Yon\_N atributidagi bir nusxani bir qiymatli aniqlaydi. Lekin RO'YHAT mohiyatidagi fakultetlar va yo'naliishlar ko'pchilikni tashkil etadi.



7.18-rasm. Talabalar o'quv joyihasining ER-diagrammasi.

BAHOLAR va FANLAR mohiyatlari orasidagi aloqalar ham keltirilgan asoslar kabi M : I tarzda bo'ladi. Fan\_N atributi, BAHOLAR va FANLAR mohiyatlari orasidagi aloqalarni modellashtiradi, ya'ni fanning raqami qaysi sanga mosligini aniqlaydi.

7.18-rasmida Talabalar o'quv loyihasining «mohiyat-aloha» modeli keltirilgan. Har bir aloqaning nomlari mohiyatlarga bog'liq ravishda berilgan. RO'YHAT mohiyati talabalar bilan bog'liq bo'lgani uchun mos ravishdagi aloqalarning nomi talabaga tegishligini bildiradi.

### Xutesa

«Mohiyat-aloha» modelida, obyektlarni identifikatorlari – soydalanuvchilar uchun juda zarur bo'lgan, mohiyatlarni aniqlandi. Bir xil turdag'i mohiyatlarni, mohiyatlarni sinfini tuzadi. Alovida olingan mohiyat, nusxa sistida ko'rildi. Mohiyatlarni atributlardan iborat bo'lib, ular obyektning xossalarni tavsiflaydi. Bir yoki bir nechta atributlar mohiyatni tashkil etadi.

Aloqalar mohiyatlarni orsidiagi o'zaro munosabatlarni aksantiradi. ER-modella aloqalar oshkor ravishda aniqlanadi – har bir aloqa nomga ega; aloqalar sinfi va aloqa nusxalarining sinflari mavjud.

«Mohiyat-aloha» modellari juda ko'p CASE vositalarning muhim qismiga aylangan. Bu programmalar ER-diagrammalarni yaratish va saqlash vositalariga ega.

Modellashni unifikatsiyalashgan (yagonalashgan) tili (UML)da «mohiyat-aloha» diagrammalarini qurishning yangi shakli taklif qilingan.

### Nazorat uchun savollar:

1. «Mohiyat-aloha» modeli nima uchun ishlataladi?
2. «Mohiyat-aloha» modeli qachon va kim tomonidan yaratilgan?
3. «Mohiyat-aloha» modeli asosiy elementlari nimalardan iborat?
4. Mohiyatlarni sinfi deganda nimani tushunasiz?
5. Mohiyat nusxasi nimani bildiradi?
6. Kompozit atribut, deganda nimani tushunasiz?
7. Ko'p qiymatli atribut, deganda nimani tushunasiz?
8. Mohiyani aniqlovchisi, deganda nimani tushunasiz?
9. Kompozit aniqlovchilar, deganda nimani tushunasiz?
10. Aloqalar sinfi, deganda nimani tushunasiz?
11. Aloqalar nusxasini ta'riflang.
12. Aloqa darajasi nimani belgilaydi.

13. Binar va ternar aloqalarni ta'riflang.
14. Binar aloqalar uch turini ta'riflang va unga misollar keltiring.
15. Romb nimani belgilaydi?
16. Kordinallik nimani bildiradi?
17. «Mohiyat-aloqa» diagrammalaridagi maxsus belgilarni ta'riflang.
18. Kuchli mohiyatni ta'riflang va unga misol keltiring.
19. Zaif mohiyatni ta'riflang va unga misol keltiring.
20. Bir qiymatli atributni ta'riflang va unga misol keltiring.
21. Ko'p qiymatli atributni ta'riflang va unga misol keltiring.
22. Mohiyatlarning ostki turlari nimani bildiradi?
23. Mohiyatlarning ustki turlari nimani bildiradi?
24. Talabalar o'quv loyihasining muqobil naqli – ER-diagrammasini yaratting.

## 8-bob. SEMANTIK OBYEKT MODEL

**Tayanch iboralar:** *semantik obyekti model, oddiy atributlar, guruhli atributlar, tarkibli atributlar, semantik obyekt atributlari, semantik obyekti diagramma, obyekti diagramma, minimal kardinal son, maksimal kardinal son, just atributlar, obyektni identifikatori, guruh identifikatori, fizik tavsif, semantik tavsif, bir qiyamli atribut, ko'p qiyamli atribut, obyekti bo'lмаган atribut, oddiy obyekti, kompozit obyekti, tarkibli obyekti, gibrild (aralash) obyekti, assotsiativ obyekti, ajdod (ustki) tur obyekti, avlod (ostki) tur obyekti, arxetur/naql obyekti, foydalanuvchi tasavvuri.*

Ushbu bobda relatsion turdag'i berilganlarni «mohiyat-alloqa» modellash usuli kabi ishlatalidigan *semantik obyekti model* (semantic object model) usulini o'rganamiz. Semantik obyekt modelni yaratish uchun *birinchi navbatda*, foydalanuvchilar so'rov qilinadi va obyekt xususida, uning xossalari va xususiyatlari to'g'risida ma'lumotlar to'planadi va tahlil qilinadi. *Ikkinci navbatda*, foydalanuvchilar tayyorlagan hisobotlar, shakllar va so'rovlari o'rganiladi va tahlil qilinadi. Natijada foydalanuvchi bergan ma'lumotlar bo'yicha obyektning boshlang'ich modeli yaratiladi. *Semantik obyekti model* usuli asosida E.F. Kodd, M. Xammer va D. Leod [11,19,53] tomonidan ishlab chiqilgan konsepsiya yotadi. Ushbu modelni yaratishdagi asosiy maqsad, berilganlarni modellashtirishda haqiqiy dunyoni yanada aniqroq tavsiflashga erishishdan iborat.

Modelning aniq shakli, uni qurish uchun foydalaniladigan tuzilmalarga bog'liq bo'ladi. Bu tuzilmalar semantik obyektlar va unga bog'liq bo'lган elementlardan iborat bo'ladi. Berilganlarni modellashtirishda ishlatalidigan «mohiyat-alloqa» va semantik obyekti model usuli bir-biriga ancha yaqin bo'lib, ikkalasi ham berilganlar bazalarini yaratuvchilar tomonidan ishlataladi va berilganlar bazalarining aniq ko'rinishdagi shakliga aylantiriladi. Ikkala usulning o'ziga yarasha ustun va kamchiliklari mavjud.

### Semantik obyektlar

Modellashtiriluvchi narsani biz 7- bobda mohiyat deb atagan edik, ushbu bobda uni *semantik obyekti* (semantic object) yoki oddiygina *obyekti* deb ataymiz. *Semantika* atamasini «ma'no, mazmun»ni anglatadi, *semantik*

obyekt deyilganda esa, obyektning foydalanuvchi tomonidan talqin qilinadigan berilganlar xususidagi tushunchalarning ma'nosini qaysidir darajada modellashirish nazarda tutiladi. Semantik obyektlar foyda-a-nuvchi qabul qiladigan narsalarni «mohiyat-aloqa» modeliga nisbatan aniqroq modelashiradi.

Mohiyatlar va obyektlar ba'zi munosabatlarda o'xshash, lekin ular orasida farqlar ham mavjud. Ularning o'xshashligi quyidagicha. Semantik obyektning rasmiy talqini quyidagidan iborat. *Semantik obyekt – bu atributlarning nomlangan majmuasi bo'lib, u yetarli darajada alohida nodirlikni tavsiflaydi*. Mohiyatlar kabi, semantik obyektlar ham sinflarga birlashadi. Obyekt sinf o'z nomiga ega bo'lib, shu orqali boshqa sinflardan farqlanadi. Bu sinflar taqdim etadigan narsalarning nomlariga mos keladi. Masalan, Talabalar o'quv loyihamizdagи berilganlar bazasida bir qancha obyekt sinflar mavjud bo'lib, shulardan biri RO'YHAT obyekt sinfi deb qarashimiz mumkin. Obyekt sinflarning nomi ham, mohiyatlar sinfining nomi kabi bosh harflar bilan yoziladi. Alohidagi semantik obyekt sinfining nusxasidan iborat bo'ladi. Masalan, «Salimov Rashid» RO'YHAT obyekt sinfining, «Mexanika-matematika» esa FAKULTETLAR obyekt sinfining nusxalaridan iborat.

Mohiyatlar kabi obyekt ham atributlar termasiga ega. Har bir atribut nodirlik xususiyatlaridan faqat bittasini tavsiflaydi. Masalan, BUYURTMA obyektning atributlari – Buyurtma №, Mijoz, Sotuvchi №, Buyurtma summasi, Buyurtma sanasi bo'lishi mumkin va har bir atribut BUYURTMA obyektidagi nodirliklardan bittasini tavsiflaydi. Atributlarning ushbu termasi ham, foydalanuvchilarining ishlashi uchun, yetarli darajadagi nodirliklarni tavsiflaydi. Dunyodagi har bir narsa juda ko'p tavsiflarga ega, ularning hammasini ham biz obyektning nodirlik xususiyatlari safida ko'rsata olmaymiz. Buning o'rniga biz foydalanuvchilarga ularning shu obyekt bo'yicha ma'lumotga bo'lgan talablarini qondirish va zaruriy funksiyalarni amalga oshirish uchun yetarli bo'lganlarini taqdim etamiz. Yetarli tafsif deganimiz, obyekt o'ziga yetarli taqdimotga ega, degan ma'noni beradi. Talabalar o'quv loyihamizdagи RO'YHAT obyektnini olsak, talabalar xususidagi Oliy o'quv yurti uchun zarur bo'lgan hamma ma'lumotlar uning atributlarida yetarli darajada o'z aksini topgan.

Obyektlar alohida nodirliklardan iborat bo'lib, foydalanuvchilar tushunchasida erkin, mustaqil va ular ustidan hisob yuritish mumkin bo'lgan narsadan iborat. Alohida nodirlik iborasining ma'nosini yaxshi tushunib olish uchun, obyektlar va obyektlarning nusxalari orasidagi farqlarni hamda 2- bobda keltirilgan ayrim misollarni eslashimiz kerak bo'ladi. RO'YHAT bu obyektning nomi, «Salimov Rashid» obyekt nusxasining qiymati. Biz obyektning alohida nodirliklari mavjud deganimizda, foydalanuvchilar

obyektning har bir nusxasi noyob va mustaqil qiymatga ega, degan fikrini nazarda tutgan bo'lamiz. Yana shuni ham ta'kidlash kerakki, obyektlarni ifodalovchi nodirliklar fizik jihatdan mavjud bo'lishi, masalan MAXSULOT obyekt sinfi kabi yoki mavjud bo'lmashigi mumkin, masalan BUYURTMA obyekt sinfi kabi. Buyurtma bu fizik narsa emas, faqatgina shartnomadagi tovarlarni yetkazish bo'yicha kelishuv. Ana shu narsa obyekt bo'lishi uchun u jismonan mavjud bo'lishi shart emas, u foydalanuvchi tasavvuridagi mustaqil tushuncha va qiymatga ega bo'lishi kerak.

### Atributlar

Semantik obyektlar o'zining xususiyatlarini tavsiflovchi atributlardan iborat bo'ladi. Atributlar uch turga bo'linadi. *Oddiy atributlar* (simple attributes) bitta elementdan iborat bo'ladi. Masalan, Buyurtma\_No, Sotuvchi\_No, Mijoz va Buyurtma\_summasi atributlari. Ikkinchisi turi, *guruhi atributlar* (group attributes) boshqa atributlar majmuasidan iborat. *Guruhi atributlarni tarkibli atributlar* deb ham atashadi. Masalan Buyurtma\_sanasi (Kun, Oy, Yil) yoki Manzil (Indeks, Shahar, Mavze, Ko'cha, Uy). Atributlarning uchinchi turi *semantik obyekt atributlari* (semantic object attributes) bo'lib, ular ikkita obyekt orasida aloqa o'matadilar. Ushbu ta'riflarga misol tariqasida *semantik obyekt diagrammani* (semantic object diagram) yoki sodda tarzda aystsak, *obyekt diagrammani* (object diagram) keltiramiz (8.1-rasm).



8.1-rasm.

Bunday diagrammalar berilgan bazalari tizimlarini yaratuvchilar tomonidan obyektlarni tavsiflash va ularning tuzilishini namoyish etish uchun ishlataladi. Obyektlar vertikal to'rtburchaklarda tasvirlanadi. Obyekt nomi yuqorida, atributlar esa ketma-ket obyekt nomidan keyin ko'rsatiladi. Oxirida shu obyekt bilan aloqada bo'ladijan semantik obyekt atributlari keltiriladi.

FAKULTET obyekti uchchala turdag'i tashkil topgan. Fakultet nomi, Dekanat, Telefon va Faks atributlari oddiy atributlar, ularning har biri bitta berilganlar elementini tavsiflaydi. Ichki\_manzil atributi – guruhli atribut, u Bino\_raqami va Qavatlar\_raqami nomli sodda atributlardan tashkil topgan. KAFEDRA, USTOZ va TALABA atributlari – bular semantik obyekt atributlari bo'lib, FAKULTET obyekti bilan bog'langan va mantiqiy jihatdan unga tegishli. Ushbu obyekt atributlarning ma'nosi shundan iboratki, foydalanuvchi birorta oliy o'quv yurtidagi fakultetni nazarda tutganda, faqatgina fakultet nomi, ichki manzili, joylashgan bino raqami, telefon va faks nomerlari emas, balki undagi kafedralar, darslar olib boruvchi professor-o'qituvchilar va ta'lim oluvchi talabalarni ham nazarda tutadi. KAFEDRA, USTOZ va TALABA ham obyekt bo'lgani uchun berilganlarning to'liq modeli ular xususidagi diagrammalarni ham saqlaydi. KAFEDRA obyekti o'zining atributlariga, USTOZ obyekti professor-o'qituvchilar xususidagi atributlarga, TALABA obyekti esa talabalar xususidagi atributlarni saqlaydi.

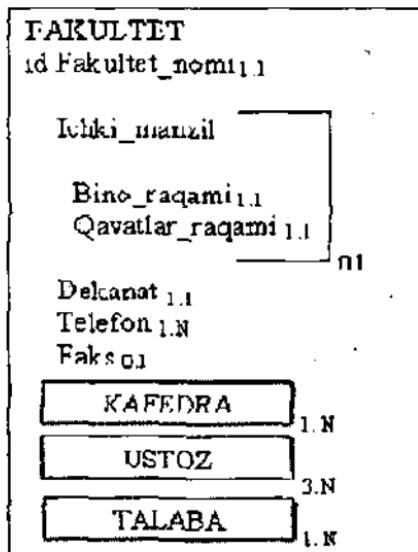
### Atributning kardinallik soni

Semantik obyektning har bir atributi maksimal va minimal kardinal songa ega. *Minimal kardinal son* obyektning yetarli bo'lishi uchun zarur bo'lgan atribut nusxalarining sonini bildiradi. Odatta, u nolga yoki birga teng bo'ladi. Agarda u nolga teng bo'lsa, atribut qiymatga ega bo'lishi shart emas, agarda u birga teng bo'lsa, atribut albatta qiymatga ega bo'lishi shart. Lekin ayrim hollarda minimal kardinallik birdan katta qiymatga ham ega bo'lishi mumkin. Masalan, VOLEYBOL\_KOMANDA obyektidagi sportchi atributining minimal kardinallik soni 6ga teng, chunki voleybol komanda tuzish uchun zarur bo'lgan sportchilar soni oltitadan kam bo'imasligi kerak.

*Maksimal kardinal son* obyektdagi atribut nusxalarining maksimal sonini bildiradi. Odatta, u 1 yoki Nga teng bo'ladi. Agarda u birga teng bo'lsa, atribut bittadan ortiq qiymatga ega bo'la olmaydi, agarda u Nga teng bo'lsa, atribut bir nechta nusxalarga ega bo'ladi, lekin nechta bo'lishi aniq ko'rsatilmaydi. Lekin ayrim vaziyatlarda maksimal kardinallik soni atributlar uchun ko'rsatiladi. Masalan, yana VOLEYBOL\_KOMANDA obyekti-

dagi Sportchi atributining maksimal kardinallik soni 20ga teng bo'lishi mumkin va bu komanda tarkibiga 20ta sportchining har birini kiritish mumkinligini bildiradi.

Atributlarning kardinalligi atributning quyi indeksi tarzida, M:N shaklda keltiriladi. Bunda M – minimal kardinal sonni, N esa maksimal kardinal sonni bildiradi. 8.2-rasmida diagrammaga e'tibor beraylik. Bu erda Fakultet\_nomi atributining minimal va maksimal kardinalligi 1ga teng, ya'ni fakultet nomi har jihatdan bitta bo'lishi mumkin, shuning uchun bu atributning bittagina nusxasi yetadi. Fakultetning Ichki\_manzil guruh atributining kardinalliklar 0..1, bu ichki manzil bo'lmashligi mumkinligini, lekin bo'lsa ham faqat bitta nusxdan iboratligini bildiradi. Guruh atributini tashkil etuvchi Bino\_raqami va Qavatlar\_raqami oddiy atributlarning kardinalligi 1..1.



8.2-rasm.

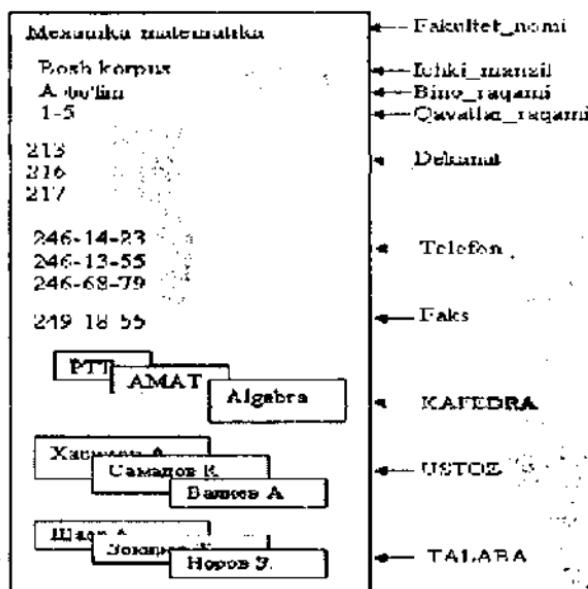
«Qanday qilib, guruhning qatnashishi shart bo'lmasa-da, uning tarkibidagi atributlarning kelishi shart bo'lishi mumkin», degan savol tug'iladi. Kardinallik soni faqat atribut va uning egasi (shu atributdan tashkil topgan guruh) orasidagi aloqalarni bildiradi. Ichki\_manzil atributidagi minimal kardinal son, shu atribut qiymatga ega bo'lishi shart emas, ya'ni fakultetning manzili bo'lishi shart emasligini anglatadi. Bino\_raqami va Qavatlar\_raqami atributlaridagi minimal kardinal sonlar, ularning Ich-

`ki_manzil` atributida ishtirok etishi shartligini bildiradi. Shunday qilib, Ich-  
`ki_manzil` atributining mavjudligi shart emas, lekin u mavjud bo'lsa, uning  
tarkibidagi `Bino_raqami` va `Qavatlar_raqami` atributlari qiymatga ega  
bo'lishi shart.

Dekanat atributining kardinalligi 1..1, bu atributning bittagina nusxasi  
yetadi. Telefon atributining kardinalligi 1..N, ya'ni fakultetda kamida bitta  
telefon bo'lishi shart, lekin ularning soni ko'p bo'lishi ham mumkinligini  
bildiradi. Faks atributining kardinalligi 0..1, ya'ni fakultetda faks bo'lmasligi  
mumkin, lekin faks bor bo'lsa bittadan ortiq bo'lishi mumkin emas, degan  
ma'no yetadi.

### Obyekt nusxalari

Obyekt diagrammalar umumiyl tuzilish to'g'risidagi tasavvurlarni hosil  
qilish uchun ishlataladi. 8.1-rasmidagi diagramma ham fakultetlarning  
umumiyl tuzilishi xususidagi tasavvurni hosil qilishga yordam beradi va bu  
diagrammani ixtiyoriy fakultetning tuzilishiga qo'llash mumkin. Obyektni  
nusxasi deganimizda, biz shu obyektdagi turli atributlarning qiyamatini  
tushunamiz.



8.3-rasm.

8.3-rasmida FAKULTET obyekting nusxasi tasvirlangan. Bunda aniq bir fakultet atributlarining qiymatlari keltirilgan. KAFEDRA, USTOZ va TALABA obyekt atributlarining bir nechtdan nusxalari mavjud. Bu obyekt atributlarning har biri to'liq obyekt hisoblanadi va shu obyekt turi bo'yicha hamma atributlarga ega. Diagrammaning chigallashtirmaslik uchun saqat obyekt atributlar nusxalarining nomlari keltirildi.

Semantik obyekt modellarda obyektlar orasida bir tomonlama aloqalar yo'q. Agarda bir obyekt o'zida boshqa obyektni saqlasa, ana shu boshqasi, o'z navbatida birinchi obyektni saqlaydi. Masalan, FAKULTET obyekti KAFEDRA obyekt atributini saqlasa, u holda KAFEDRA obyekti FAKULTET obyekt atributini saqlaydi. Bu obyekt atributlar, ba'zida *just attributlar* (paired attributes) deb ataladi, chunki ular doimo just holda uchraydi. Chunki, A obyekt V obyekt bilan aloqada bo'lsa, u holda V obyekt A obyekt bilan bog'langan bo'ladi. Obyektlar orasidagi aloqalar hech qachon bir tomonlama bo'lmaydi. Shuning uchun just obyektlar turli sohalarda ko'p uchraydi.

### Obyektning identifikatorlari

*Obyektning identifikatori* (object identifier) – bu bir yoki bir nechta obyekt atributi bo'lib, ular yordamida obyektlarning nusxalari identifikatsiya qilinadi. Bunday identifikatorlar semantik obyektni identifikatsiyalashga nomzod atributlar bo'ladi. Masalan, FAKULTET obyektni identifikatsiyalash uchun nomzod atribut sifatida Fakultet nomi yoki Fakultet raqami atributlari bo'lishi mumkin. Chunki, bu atributlarning har biri FAKULTET obyekt sinfi nusxalari uchun mumkin bo'lgan nomlar sifatida e'tirof etish mumkin. Boshqa atributlarning birortasi Ichki\_manzil, Dekanat, Telefon yoki Faks nomzod atributlar bo'la olmaydi, chunki bularni FAKULTET obyekt sinfi nusxalarining nomi sifatida e'tirof etib bo'lmaydi.

*Guruh identifikatori* (group identifier) – bu bittadan ortiq atributdan iborat identifikator. Misol sifatida (Kun, Oy, Yil) yoki Manzil (Indeks, Shahar, Mavze, Ko'cha, Uy) identifikatorlarni keltirish mumkin.

Obyektning identifikatorlari noyob yoki noyob bo'imasligi mumkin. Masalan, BUYURTMA obyekti uchun Buyurtma\_Nº noyob identifikator, TALABA obyekti esa Talaba\_Familiyasi\_Ismi esa noyob identifikator bo'la olmaydi. Chunki, familiyasi va ismini bir xil bo'lgan ikkita talaba bo'lishi mumkin. Bunday vaziyatda foydalanuvchilar Talaba\_Familiyasi\_Ismi atributi orgali bitta yoki undan ortiq talabalar guruhini identifikatsiyalash, zarur bo'lganda esa, boshqa atributlar yordamida ushbu guruhdagagi aniq elementni ko'rsatish mumkin.

Semantik obyekt diagrammalarda obyekt identifikatorlarining oldiga ID belgisi qo'yiladi. Agar identifikator noyob bo'lsa, belgining tagiga chizib ham ko'rsatish mumkin. Odatda, atribut identifikator sifatida ishlatilsa, u albatta qiymat olishi shart. Bundan tashqari odatda, har bir obyekt uchun bittadan ortiq atribut identifikatsiyalash uchun ishlatalmaydi. Shuning uchun identifikator atributlarining kardinalligi 1.1 bo'ladi va bundan biz yashirin holda foydalanamiz.

### Atribut domenlari

Atribut *domeni* – qiymatlar to'plamini tavsiflashga ishlataladi. Domen tavsifi atributning turiga beg'liq. Oddiy atributning domeni fizik va semantik tavsifdan tuziladi. *Fizik tavsif* (physical description) berilgan larning turini (masalan, son yoki satr), uzunligini va berilganlarning boshqa chegaralarini ko'rsatadi. *Semantik tavsif* (semantic description) funksiyani yoki shu atributning tayinlanishini ko'rsatadi, shu atribut boshqa aributlardan fizik tavsifi bilan farqlanadi.

Masalan, *Fakultet\_nomi* – Oliy o'quv yurti fakultetlarining nomini ko'rsatuvchi atribut domeni 20 ta belgidan oshmasligi kerak. Bu erda «20 ta belgidan oshmasligi kerak» iborasi domenning fizik tavsifi, «Oliy o'quv yurti fakultetlarining nomini ko'rsatuvchi» iborasi esa domenning semantik tavsifi. Domenning semantik tavsifi xuddi shunday uzunlikdagi atributning boshqa tavsiflaridan farqlaydi, masalan kafedraning nomi, fanning nomi va h.k.

Ayrim holatlarda oddiy atribut domenining fizik tavsifi tartiblangan raqamlardan iborat ro'yxat – aributning alohida qiymatlari termasidan iborat bo'ladi. Masalan, *Fakultet\_raqami* aributining domeni tartiblangan yoki ixtiyor yoki noyob raqamlardan iborat bo'lishi mumkin.

Guruh aributlarining domeni ham fizik va semantik tavsifga ega. Guruh aributining fizik tavsifi – guruhdagi hamma aributlar ro'yxatining tartiblangan raqamlaridan iborat bo'ladi. Guruh aributining semantik tavsifi esa – bu guruh funksiyasining yoki tayinlanishining tavsifi. *Ichki\_manzil* domenining fizik tavsifi (*Bino\_raqami*, *Qavatlar\_raqami*) ro'yxat bo'lsa, semantik tavsifi «fakultet joylashgan ichki manzil» iborasi bo'ladi.

Obyekt aributning domeni – shu obyekt nusxalarining ushu turdag'i majmuasidan iborat. Masalan 4.3-rasmdagi USTOZ obyekt aributining domeni, berilganlar bazasidagi USTOZ obyekt sinfi hamma nusxalarining to'plamidan iborat bo'ladi. KAFEDRA obyektining domeni, berilganlar bazasidagi hamma KAFEDRA obyekt sinfi nusxalarining to'plamidan iborat bo'ladi. Ma'lum ma'noda, donen aributi – bu dinamik tarzda raqamlangan ro'yxat bo'lib, shu turdag'i hamma obyekt nusxalarini saqlaydi.

## Berilganlarning semantik obyekt modellarini yaratish

Berilganlarning semantik obyekt modellarini yaratish uchun tayyorlangan shakllar hisobotlar va so'rovlar bilan tanishish hamda ularni tahlil qilish natijasida obyektning tuzilishi xususida boshlang'ich ma'lumotlar olinadi. Masalan, FAKULTET obyektning tuzilishini yaratish uchun, biz avval fakultetga bog'liq bo'lgan hamma hisobotlar, shakllar va so'rovlarni to'playmiz. Ana shundan keyin, biz FAKULTET obyektni modellashtiramiz va tizimni yaratamiz. Tizim esa bizga hisobotlar, shakllar va so'rovlar yaratish imkonini beradi.

Umuman yangi tizimni yaratish uchun esa, obyektni har tomonlama o'rghanish va uni chuqur tahlil qilish kerak. Bu jarayonda foydalanuvchilar bilan muloqotlar qilib, natijada obyekt xususida yanada ko'proq ma'lumotlar to'plash, qaysi bir obyekt atributlar ahamiyatli ekanligini aniqlashtirish kerak. Ana shular natijasida berilganlarning obyekt modeli oydinlasha boshlaydi.

### Fakultet obyekti

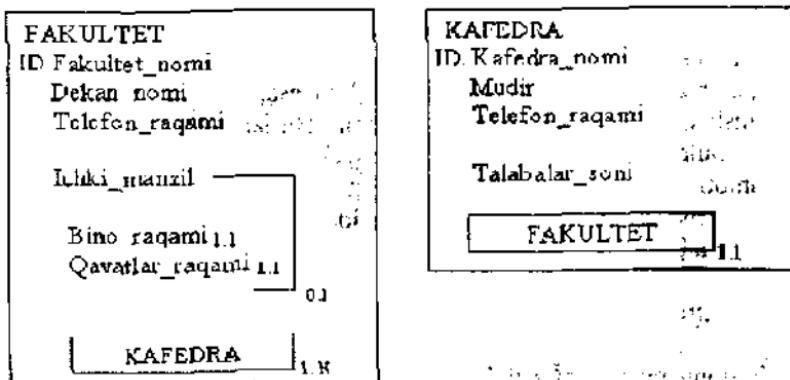
Bizning maqsadimiz tizimlar hosil qiladigan hisobotlarni o'rGANIB va ularni tahlil qilib, qaysi obyektlar va atributlar berilganlar bazasida saqlanishi kerakligini aniqlashdan iborat. Faraz qilaylik, bizda 8.4-rasmdagi fakultet, ya'ni «Geografiya» fakulteti xususidagi ma'lumotlar keltirilgan bo'lsin. Bu bitta xususiy hol bo'lib, universitet bunday hisobotlarni boshqa fakultettardan ham olgan, deb faraz qilamiz.

Geografiya fakulteti Masaliyev Anyar Vaxidovich, dekan			
Telefon: 246-29-54		Bosh bino «B» bo'shiu, 203 xona	Telki manzil:
Xatdedra	Mudir	Telefon	Talabatlar soni
Tabiiy geografiya Ilimliy geografiya	Zuxitov S.S. Chorlyev Z.J.	246-12-13 246-22-89	15 18
Geodeziya	Aliyev P.Z.	246-37-87	16
Quruqtik hidrologiyasi	Fayziyev B.A.	246-42-43	17
Demografika	Safarov S.D.	246-24-93	12
Mintaqaviy iqtisod	Yakubov O.Sh.	246-55-64	13

### 8.4-rasm.

Berilganlar modelini tuzish uchun yetarli darajadagi misollarni to'plab, ular asosida fakultet hisobotining namunasini qurish mumkin. Bu erda biz 8.4-rasmdagi hisobotni namuna sifatida qabul qilamiz.

Hisobotni tahlil qilish natijasida, biz fakultetga xos ma'lumotlarni (fakultet nomi, dekanning ismi sharifi, ichki manzil va telefon) hamda fakultetdagi har bir kafedra xususidagi ma'lumotlarni ko'ramiz. Bu bizni, berilganlar bazasi FAKULTET va KAFEDRA obyektlarini va ular orasidagi aloqalarni ham o'z ichiga olishi mumkin ekan, degan taxminga olib keladi. Bu boshlang'ich taxmin natijasida biz 8.5-rasmdagi obyekt diagrammanni hosil qilamiz.



8.5-rasm.

FAKULTET obyektidagi KAFEDRA obyekt atributining kardinalligi 1:N – bu, fakultetning kamida bitta kafedrasi bo'lishi shartligini, lekin ular bir nechta bo'lishi mumkinligini bildiradi. Bu ma'lumot aniqlash hisobotning tahlilidan kelib chiqmaydi, balki foydalanuvchilardan so'rash natijasida aniqlanadi.

Xuddi shunday tartibda, KAFEDRA obyektining tuzilishi 8.4-rasmdagi ma'lumotlar asosida quriladi. Yuqorida ta'kidlaganimizdek, obyekt attributlar doimo juft bo'lgani uchun FAKULTET obyekti KAFEDRA obyektinining ichida keltirilgan, lekin bu fakt ni ham 8.4-rasm orqali keltirib chiqarish mumkin emas. KAFEDRA attributining kardinalligini aniqlash kabi, FAKULTET obyektinining kardinalligi – 1..1, ham so'rov natijasida aniqlanadi. Buning ma'nosi, kafedra bitta va faqatgina bitta fakultet bilan bog'lanishi mumkin, degani.

### Kafedra obyekti

8.6-rasmda keltirilgan ma'lumotlar kafedra xususida bo'sib, unda shu kafedrada ishlayotgan professor-o'qituvchilar xususida ham axborot beril-

gan. E'tibor bersangiz, ushbu hisobotda kafedradagi professor o'qituvchilarning ichki manzil bo'yicha xona raqamlari keltirilgan. Bu ma'lumot 8.5-rasmidagi KAFEDRA obyektining tuzilishida keltirilmagan va bu ma'lumot-

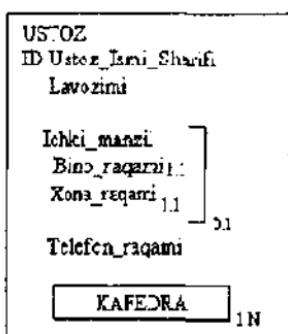
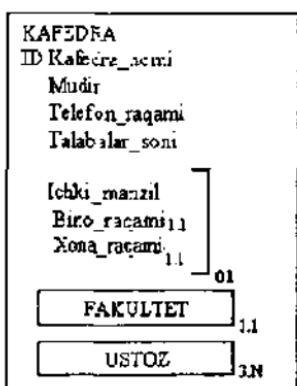
«Jumaiy geografiya» kafedrası Geografiya fakulteti			
Medir:	Choriyev Z.J.		
Telefon:			
Ichki manzil:	Bosh bino «B» bo'lim, 213 xona		
Ustoz	Lavozim	Xona	Telefon
Choriyev Z.J.	professor	Bosh bino «B» bo'lim xona 213	246-22-89
Ashurov I.X.	dotsent	Bosh bino «B» bo'lim xona 215	155-56-98
Kaxxorov A.B.	dotsent	Bosh bino «B» bo'lim xona 215	233-55-77
Maxarov T.S.	dotsent	Bosh bino «B» bo'lim xona 217	356-66-80
Kamalov S.E.	kt o'qit	Bosh bino «B» bo'lim xona 217	229-13-15

### 8.6-rasm.

ni 8.7-rasmda akslanriramiz. Bunday qo'shimchalarini berilganlar modeliga ana shunday jarayonda kiritish oddiy holat bo'lib, aniqlashtirish tizim yaratilguncha davom etadi.

#### Ustoz obyekti

8.6-rasm faqatgina KAFEDRA obyekti kiritishni emas, balki professor-o'qituvchilar xususida ham yana bir obyekt bo'lishi kerakligini ang-



### 8.7-rasm.

latadi. Shu fikrga asosan 8.7-rasmida USTOZ obyekti qo'shilgan. USTOZ obyektini identifikatori Ustoz\_Ismi\_Sharifi atributi noyob atribut emas.

8.7-rasmidagi diagrammaga asosan har bir kafedrada kamida uchta professor-o'qituvchi bo'lishi va har bir ustozi faqatgina bitta kafedrada ishlashi lozimligi tushuniladi. Shunday qilib, ish reglamentiga asosan har bir ustozi bir shiat birligida ishlaydi.

8.8-rasmida kafedraning ikkinchi hisoboti keltirilgan. Unda kafedra va talabalar xususidagi ma'lumotlar bor. Talabalarning yo'naliishi shu kafedraga bog'liq. Bitta obyekt xususida ikkita hisobotning mavjudligi, bitta narsani ikki nuqtayi nazardan qarash, deb tushunish kerak. Bu oddiy holat bo'lib, bundan tashqari kafedrani obyekt sifatida qarashimiz to'g'riligini tasdiqlaydi.

**Talabalar ro'yhati**  
**«Djitimoiy geografiya» kafedrasи**

Mudir:	Choriyev Z.J.	Telefon:	246-22-89	
Reyting daftari raqami	Talaba Ismi Sharifi	Kurs	Telefon	
IU0345678	Asomov Z.J.	3	246-28-89	
IU0348978	Ashurova I.X.	3	159-56-98	
IU0453456	Karimova A.B.	4	245-55-77	
IU0345678	Masudov T.S.	4	356-55-11	

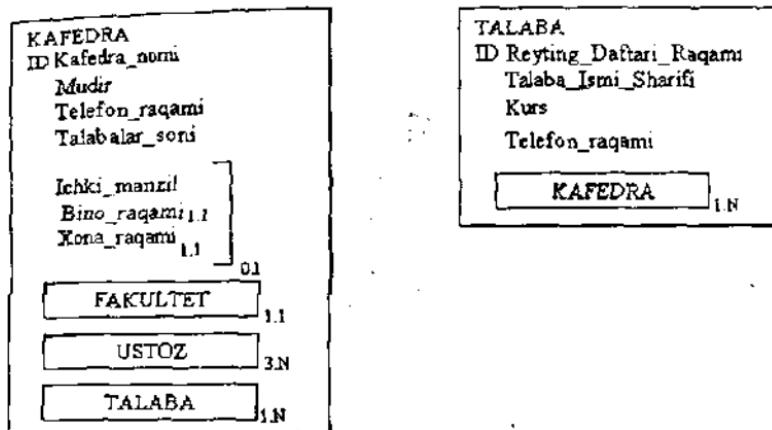
**8.8-rasm.**

**Talaba obyekti**

8.8-rasmidagi hisobot talabalar obyektni tashkil etishini tasdiqlaydi. Shuning uchun KAFEDRA obyekti TALABA va USTOZ obyektlarini o'zida saqlaydi (8.9-rasm).

TALABA obyekti 8.9-rasmida Reyting\_Daftari\_Raqami, Talaba\_Ismi\_Sharifi, Kurs va Telefon\_raqami, ya'ni 8.8-rasmidagi hisobotda ko'rsatilgan atributlarga ega. Reyting\_Daftari\_Raqami identifikatori noyob atribut va shu orqali TALABA obyekti identifikasiya qilinadi.

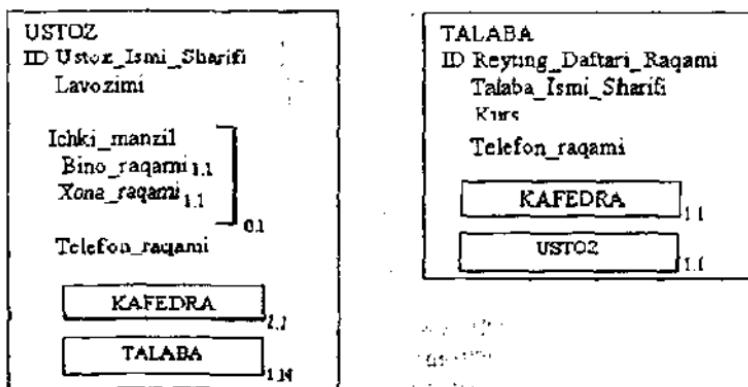
Aniqlashtirilgan USTOZ va TALABA obyektlarining diagrammasi 8.10-rasmida keltirilgan. TALABA obyektiiga e'tibor bersak, KAFEDRA obyekti ham USTOZ obyekti ham yagona qiymatga ega (ularning maksimal kardinalligi birga teng). Bundan talaba faqat bitta kafedraga biriktirilishi va faqat bitta rahbarga ega bo'lishi zarurligi kelib chiqadi.



8.9-rasm.

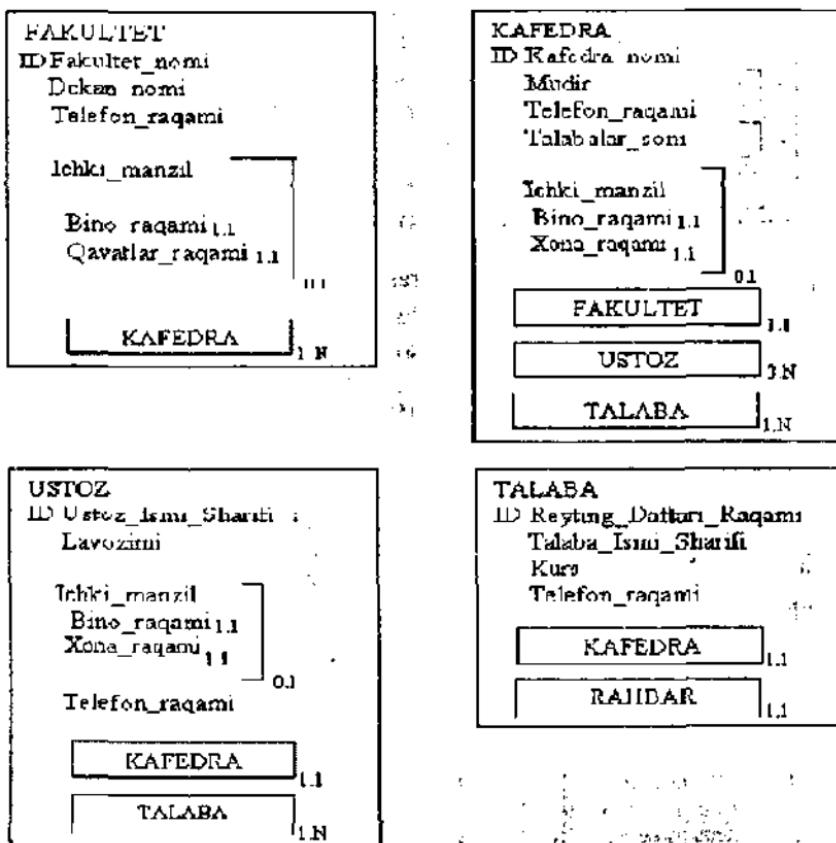
### Obyektlar tasnifi

Oliy o'quv yurti uchun obyekt diagrammalarning so'nggi o'quv vari-anti 8.11-rasmda keltirilgan. Bunga bir qancha o'zgartirishlar kiritilgan. Modeldagi aniqlikni oshirish uchun TALABA obyektdagi USTOZ atributi RAHBAR atributiga o'zgartirildi. USTOZ obyektfining nusxasi TALABA obyekti nusxasidagi atribut bilan bog'langan bo'lib, bu qaysidir ustoz emas, balki talabaning aniq rahbari bo'lib, bu atama ustoz atamasiga nishbatan aniqdir.



8.10-rasm.

Bunday aniqlik kiritish natijasida, USTOZ atributining domeni o'zgarmaydi. RAHBAR atributi USTOZ domeniga ega, USTOZ atributi ham xuddi shunday domenga ega. U avvalgidek USTOZ semantik obyekting nusxasini ifodalaydi. Nomni o'zgartirish, bu vazifani aniqlashtirishdan iborat bo'lib, USTOZ domenining TALABA semantik obyektiagi ahamiyatini belgiladi.



8.11-rasm.

8.1- va 8.2-jadvallarda berilganlar modelining tasnifi keltirilgan. Semantik obyektlar va atributlar semantik obyektlarning tasnifida aniqlangan, domenlar esa domenlar tasnifida aniqlangan. Birinchi jadval semantik obyekt diagrammalariga nisbatan axborotni muqobil taqdimotidan iborat bo'lib, uni talqini ravshan.

Obyekt nomi	Xususiyat nomi	Min. kard.	Mak. kard.	Identif. sh holati	Domen nomi
FAKULTET	Fakultet_nomi	1	1	ID	Fakultet_nomi
	Dekan_nomi	1	1		Odam_nomi
	Familiya	1	1		Familiya
	Ism	1	1		Ism
	Ota_ismi	0	1		Ota_ismi
	Telefon_raqami	0	1		Telefon_raqami
	Ichki_manzil	0	1		Ichki_manzil
	Bino_raqami	1	1		Bino_raqami
	Qavatlar_raqami	1	1		Qavatlar_raqami
	KAFEDRA	1	N		KAFEDRA
KAFEDRA	Kafedra_nomi	1	1	ID	Kafedra_nomi
	Mudir	1	1		Odam_nomi
	Familiya	1	1		Familiya
	Ism	1	1		Ism
	Ota_ismi	0	1		Ota_ismi
	Telefon_raqami	0	1		Telefon_raqami
	Talabalar_soni	0	1		Talabalar_soni
	Ichki_manzil	0	1		Ichki_manzil
	Bino_raqami	1	1		Bino_raqami
	Xona_raqami	1	1		Xona_raqami
USTOZ	FAKULTET	1	1	ID	FAKULTET
	USTOZ	3	N		USTOZ
	TALABA	1	N		TALABA
	Ustoz_Ismi_Shariifi	1	1		Odam_nomi
	Familiya	1	1		Familiya
RAHBARLIK-	Ism	1	1	ID	Ism
	Ota_ismi	0	1		Ota_ismi
	Lavozim	1	1		Lavozimi
	Ichki_manzil	0	1		Ichki_manzil
	Bino_raqami	1	1		Bino_raqami
	Xona_raqami	1	1		Xona_raqami
	Telefon_raqami	0	1		Telefon_raqami
	KAFULTET	1	1		FAKULTET
	TALABA	1	N		TALABA

DAGI_TALABA					
TALABA	Reyting_Daftari_Raqami	1	1	ID	Reyting_Daftari_Raqami
	Talaba_Ismi_Sharifi	1	1		Odam_nomi
	Familiya	1	1		Familiya
	Ism	1	1		Ism
	Ota_ismi	0	1		Ota_ismi
	Kurs	1	1		Kurs
	Telefon_raqami	0	1		Telefon_raqa-mi
	KAFEDRA	1	1		KAFEDRA
	RAHBAR	1	1		RAHBAR

Ikkinchı jadval, ya'ni domenlar jadvalida, semantik obyekti diagrammalardan olib bo'lmaydigan ma'lumotlar keltirilgan. Yuqorida ta'kidlaganimizdek, domen ham semantik, ham fizik tavsifga ega.

8.2-jadval.

Domen nomi	Turi	Semantik tavsif	Fizik tavsif
FAKULTET	Semantik obyekt	OO'YU fakultetlar-dan biri	8.1-jadvalga qarang
Fakultet_nomi	Oddiy	Fakultetning rasmiy nomi	Text 30
Manzil	Guruhli	OO'YU manzili	Text 35
Ichki_manzil	Guruhli	OO'YU dagi manzil	Bino_raqami
Bino_raqami	Oddiy	OO'YU dagi bino	Qavatlar_raqami
Qavatlar_raqami	Oddiy	Binodagi qavatlar	Text 10
KAFEDRA	Semantik obyekt	Fakuhet kafedrasi	8.1-jadvalga qarang
Kafedra_nomi	Guruhli	Kafedraning rasmiy nomi	Text 30
Odam_nomi	Oddiy	Administrator, ustoz yoki talaba familiyasi, ismi, otasining ismi	Familiya
Familiya	Oddiy	Familiyaning qo'shatuvchi. Odam_nomi guruhining qismi	Ism
Ism	Oddiy	Ismni ko'rsatuvchi	Otaning ismi
			Text 15
			Text 10

Ota_ismi	Oddiy	Odam_nomi guruhining qismi Ota ismini ko'rsa-tuvchi Odam_nomi guruhining qismi	Text 18
Talabalar_soni	Oddiy	Kafedraga tegishli talabalar soni	Integer; format 99
Xona_raqami	Oddiy	Fakultetga tegishli xona	Text 4
Telefon_raqami		Hudud kodlari bilan telefon raqami	Text 11
USTOZ	Semantik obyekt	OO'YU shtatidagi professor-o'qituvchi	8.1-jadvalga qarang
Lavozim	Oddiy	Professor-o'qituvchi shtatidagi lovozim-lardan biri	Text 10
TALABA	Semantik obyekt	OO'YU ga qabul qilingan talaba	8.1-jadvalga qarang
Reyting_Daftari_Raqami	Oddiy	Talaba reyting daftaring raqami	Text 10
Kurs	Oddiy	Talaba kursi	Integer; format 9

### Obyektlarning turlari

Biz bu yerda, yana yangi atamalarga ta'rif beramiz hamda obyektlarning yetti turi bilan qisqacha tanishib chiqamiz.

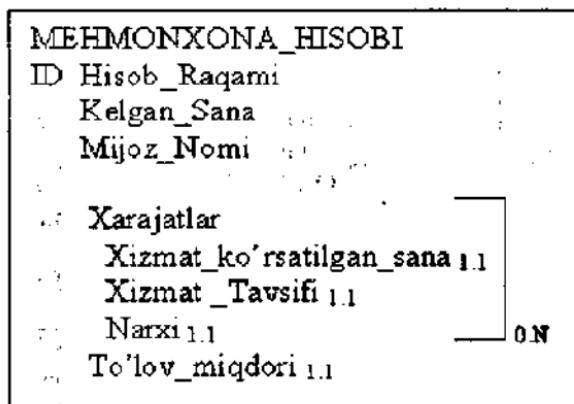
*Bir qiymatli atribut* (single-value attribute) – bu maksimal kardinalligi birga teng bo'lgan atribut. *Ko'p qiymatli atribut* (multi-value attribute) – bu maksimal kardinalligi birdan katta bo'lgan atribut. *Obyekt bo'limgan atribut* (non-object attribute) – bu oddiy yoki guruhli atribut.

*Oddiy obyekti* (simple object) – bu semantik obyekt bo'lib, faqat bir qiymatli, oddiy yoki guruhli atributlardan iborat bo'ladi. Bunga misol, 8.12-rasmdagi QURILMA oddiy obyekti.

QURILMA	
ID	Invertar_Raqami
	Tavsifi
	Sotib_olingga_sana
	Narxi

8.12-rasm.

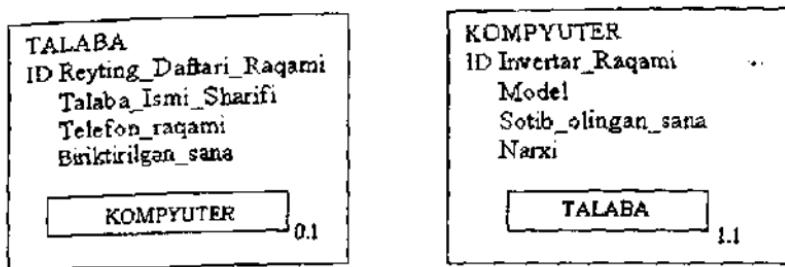
*Kompozit obyekt* (composite object) – bu semantik obyekt bo'lib, obyekt atributi bo'limgan, bitta yoki bir nechta ko'p qiymatli, oddiy yoki guruhli atributlardan iborat bo'ladi. Misol sifatida 8.13-rasmdagi MEHMONXONA\_HISOBI obyekti keltiramiz.



8.13-rasm.

Bu – Hisob\_Raqami, Kelgan\_Sana, Mijoz\_Nomi va To'lov\_miqdori atributlaridan iborat. Bundan tashqari, mijozga ko'rsatiladigan xizmatlar tavsifidan iborat, takrorlanuvchi atributlar guruhi mavjud. Har bir guruh o'z ichiga Xizmat\_ko'rsatilgan\_sana, Xizmat\_Tavsifi va Narxi atributlarini oladi. Xarajatlar atributi – bu guruhli atribut bo'lib, maksimal kardinallik soni Nta. Buning ma'nosi, ushbu guruh MEHMONXONA\_HISOBI obyekti juda ko'p marta takrorlanishi mumkinligini anglatadi.

*Tarkibli obyekt* (compound object) kamida bitta obyekt atributdan iborat bo'lishi mumkin. 8.14-rasmda TALABA va KOMPYUTER obyektlari uchun diagrammalar keltirilgan. TALABA obyektining atributlaridan birini KOMPYUTER obyekti, KOMPYUTER obyektining atributlaridan birini esa, TALABA obyekti tashkil etadi. TALABA va KOMPYUTER obyektlarining atributlari ichida obyektlar ham bo'lgani uchun, ularning ikkalasi ham tarkibli obyektlar bo'ladi. Bundan tashqari ulardagi atributlarning hech biri ko'p qiymatli bo'limgani uchun, obyektlar orasidagi aloqa turi - 1:1 bo'ladi.

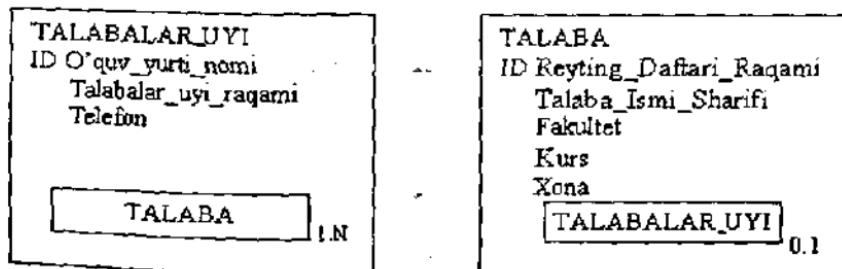


8.14-rasm.

Tarkibli obyektga yana bir misol, 8.15-rasmdagi TALABALAR\_UYI va TALABA obyektlari diagrammalari. Talaba yashasa, faqatgina bitta talabalar uyida yashashi mumkin, lekin u uyda yashamasligi ham mumkin. Demak, TALABALAR\_UYI obyekti ko'p qiymatli TALABA obyekt attributini saqlaydi, TALABA obyekti esa bir qiymatli TALABALAR\_UYI obyekt attributini saqlaydi. Shunday qilib, TALABALAR\_UYI va TALABA obyektlari orasidagi aloqa «birlikdan-ko'plikka» turiga ega, ya'nı 1:N.

Umumiy holda, F obyekt bitta yoki bir nechta Q tarkibli obyektni saqlashi, xuddi shunday Q obyekti ham bitta yoki bir nechta F tarkibli obyektni saqlashi mumkin.

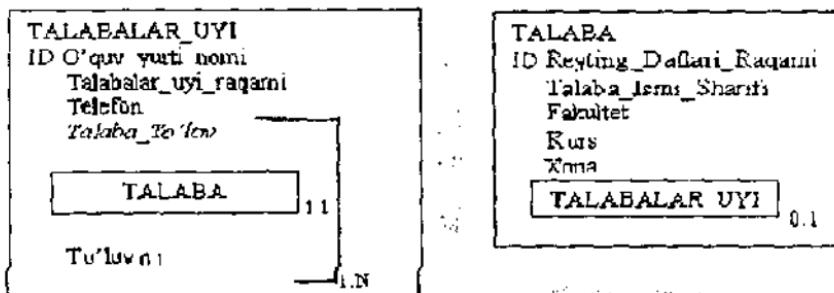
*Gibrild (aralash) obyektlar* (hybrid objects) – bu kompozit va tarkibli obyektlar kombinatsiyasidan iborat. Xususiy holda, gibrild obyekt – bu semantik obyekt bo'lib, uning tarkibida kamida bitta ko'p qiymatli guruh atributi bo'ladi va bu atribut tarkibida esa obyekt atribut joylashadi.



8.15-rasm.

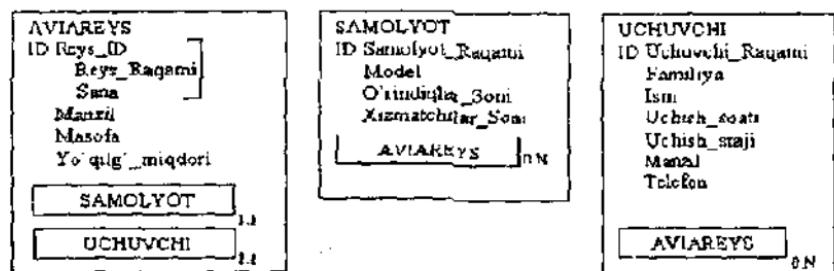
8.16-rasmda gibrild (aralash) obyekt modeliga misol keltirilgan. TALABALAR\_UYI obyekti tarkibida ko'p qiymatli *Talaba\_To'lov* guruh

atributni bo'lib, uning tarkibida esa TALABA obyekt atributini va oddiy To'lov atributini mavjud. Talaba\_To'lov guruh atributining maksimal kardinalligi N. TALABA obyekt atributini va To'lov atributining maksimal kardinalligi 1. Bu ikkala atribut TALABALAR\_UYI obyekti tarkibidagi juft atributlar hisoblanadi. Ular juft atributlar sifatida ko'p qiymatga ega bo'ladi. Har biri alohida ko'p qiymatli bo'tishi mumkin emas, chunki talabalar uyida yashash to'lovi talabalar uchun har xil bo'tishi mumkin.



8.16-rasm.

*Assotsiativ obyekti* (association object) – bu ikki va undan ortiq obyektlarni bog'lovchi va shu aloqa xususidagi berilganlarni saqlovchi obyektdir.

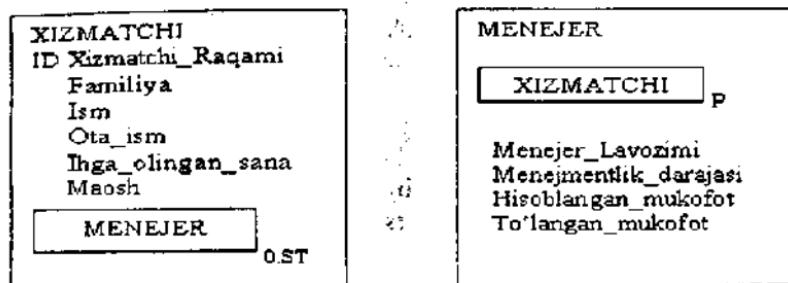


8.17-rasm.

8.17-rasmdagi diagrammada AVIAREYS assotsiativ obyekt bo'lib, SAMOLYOT va UCHUVCHI obyektlarini o'zarlo bog'lagan holda, ular orasidagi aloqaga tegishli bo'lgan berilganlarini ham saqlaydi. AVIAREYS assotsiativ obyekti bittadan SAMOLYOT va UCHUVCHI obyekt atributlarni saqlaydi, lekin ularning har biri AVIAREYS obyekt atributlarning to'plamini saqlaydi. Bunday vaziyatlar predmet sohalarda juda ko'p uchraydi, ayniqsa, ikkita obyekt orasidagi bir-biriga ikki va undan ortiq

narsalarni yuklash jarayonida. Masalan, MASALA obyekti TALABA va ADABIYOT obyektlarini bog'laydi, BUYURTMA obyekti TA'MINOT-CHI va XIZMAT\_KO'RSATUVCHI obyektlarini bog'laydi.

*Ajdod/ostki tur obyektlar* (parent/subtype object) qanday bo'lishini aniqlashtirish uchun, XIZMATCHI obyektni tahlil qilamiz. XIZMATCHI obyektning ayrim atributlari hamma xizmatchilarga, ba'zi atributlari esa faqat rahbar xizmatchilarga tegishli bo'lishi mumkin. Ana shunday vaziyatlarda, tur osti obyektlaridan foydalanish kerak bo'ladi. Masalan, 8.18-rasmidagi XIZMATCHI obyekti MENEJER ostki tur obyektni saqlasin.



8.18-rasm.

Menejerlarga tegishli hamma atributlar MENEJER obyektiga o'tkazilgan. Menejer bo'Imagan xizmatchilar XIZMATCHI obyektning bir nusxasidan iboratdu hisob, dironta ham MENEJER obyektning nusxasiga ega bo'lihaydi. Menejer bo'lgan xizmatchilar esa, XIZMATCHI va MENEJER obyektlarining bittadan nusxasiga ega bo'ladi. XIZMATCHI obyekti *ajdod obyekti* (parent object) yoki *ustki tur* (supertype object), MENEJER obyekti esa *ostki tur obyekti* (subtype object) deb ataladi. Ostki tur obyektning birinchi atributi bo'lib, R harfi bilan belgilangan, ajdod atribut hisoblanadi. Ajdod atributni belgilash majburiyidir. Ostki tur obyektni aniqlovchi atributlar, xuddi ajdod obyektidagi kabi bo'ladi. Ya'ni Xizmatchi\_Raqami atributi XIZMATCHI va MENEJER obyektlarining identifikatori bo'ladi.

Ostki tur obyekti 0.ST yoki 1.ST indekslari bilan belgilanadi. Sonlar ostki tur obyektning minimal kardinalligini ifodalaydi. Agarda minimal kardinallik 0 bo'lsa, ostki tur mavjud bo'lishi shart emasligini, 1 ga teng bo'lsa, mavjud bo'lishi shart ekanligini bildirada. ST belgisi (SubType) atribut ostki tur ekanligini hamda u «MAVJUD» turiga a'zoligini bildiradi.

*Ajdod/ostki tur obyektlarning eng muhim tavsisi, uning vorislik xususiyatiga egaligi.* Ostki tur obyekti o'z ajdodining hamma atributlarini voris sifatida qabul qiladi. Shuning uchun ham, MENEJER obyekti XIZMATCHI

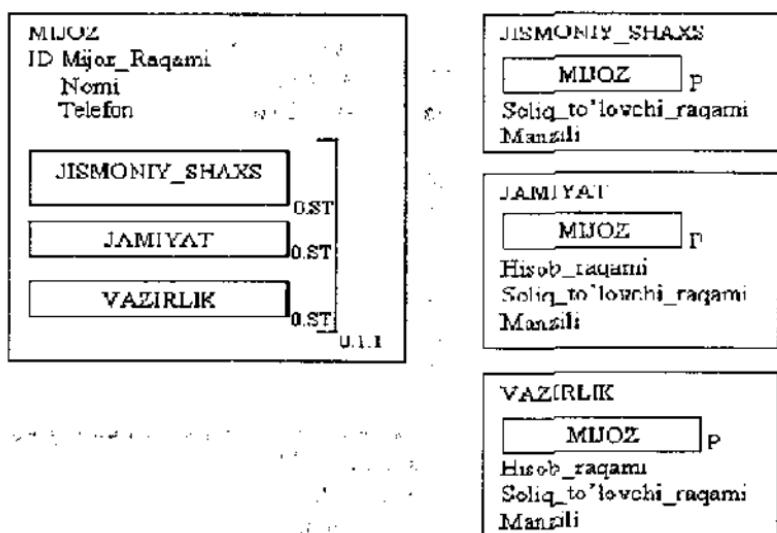
obyektning hamma atributlariga ega bo'ladi. Bundan tashqari, ajdod obyekt ham o'z avlodining, ya'ni ostki turlarning hamma atributlariga ega bo'ladi.

8.19-rasmda uch xildagi mijozlarning ostki tur guruhi tasvirlangan. Guruhning 0.1.1 indeksi, ostki tur kerak emasligini, lekin mayjud bo'lsa, guruhda faqat bitta minimal va bitta maksimal ostki tur mayjud bo'lishi kerakligini bildiradi. Shuning uchun ham ostki turlarning har birining indeksi 0.1.1, ya'ni ostki turlarning hammasi mayjud bo'lishi shart emasligini bildiradi. Agarda ostki tur kerak emas bo'lsa-da, lekin mayjudligi shart bo'lsa, ulardan uchchalasining zarurligi, guruhning indeksi 0.1.3 bo'lishi bilan ifodalananar edi.

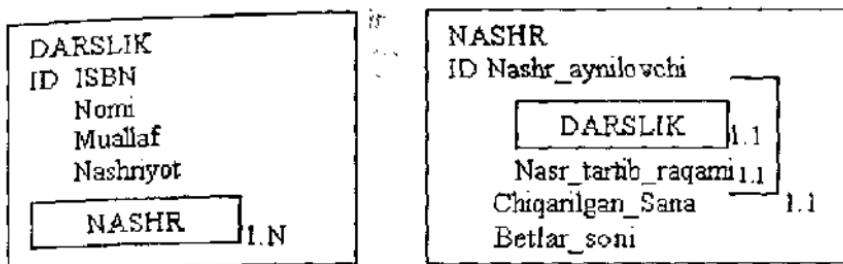
Obyektlarning oxirgi turi, bu *arxetur/naql obyektlar* (archetype/ version) bo'lib, ular boshqa semantik obyektlarni yaratadi. Bu semantik obyektlarda arxetur obyektning naqli, yaratilishi va nashr qilinishi xususidagi ma'lumotlar bo'ladi.

Masalan, 8.20-rasmdagi diagrammada DARSLIK arxetur obyekti, NASHR naql obyektni yaratadi. Ushbu modelga asosan, Nomi, Muallif va Nashriyot atributlari DARSLIK obyektiga, Nashr\_aynilovichisi, Chiqarilgan\_Sana va Betlar\_soni atributlari esa NASHR naql obyektiga tegishli.

NASHR naql obyektni aniglovchi guruh ikki qismdan iborat bo'ladi – DARSLIK va Nashr\_Tartib\_Raqami. Keltirilgan misol, naql obyekt identifikatorining namunaviy nusxasi. Identifikatorning bir qismi arxetur obyekti, ikkinchi qismi esa oddiy atribut bo'lib, arxetur naqlini identifikasiyalaydi.



8.19-rasm.



8.20-rasm.

### Semantik obyekt model va «mohiyat-aloqa» modelini taqqoslash

Semantik obyekt model va «mohiyat-aloqa» modelining o'xshashlik jihatlari bilan bir qatorda farqli tomonlari ham mavjud. O'xshashlik tomoni, ikkala model ham foydalanuvchi tomonidan beriladigan ma'lumotlarni aniqlashtirish va hujjatlashtirishdan iborat. Ikkala modelning maqsadi, foydalanuvchilar uchun zarur narsalarning tuzilishlarini va ular orasidagi aloqalarni modellashtirishdan iborat.

Ikkala modelning keskin farq qiladigan tomoni esa, ularning obyektga yo'naltirilgantigida. «Mohiyat-aloqa» modeli tayanch tushuncha sifatida *mohiyatlarni* qaraydi. Mohiyatlar va aloqlar ushbu berilgan modelida bo'linmas zarrachalar rolini o'ynaydi. Bu zarrachalar *foydalanuvchi tasavvuri*, deb atalgan tuzilishlarni tashkil qiladi. *Foydalanuvchi tasavvuri* – bu mohiyatlar kombinatsiyasidan iborat bo'lib, ularning tuzilishi semantik obyektlar tuzilishini eslatadi.

Semantik obyekt modelning tayanch tushunchasi – bu *semantik obyekt*. Berilganlar modelidagi semantik obyektlar majmuasi – bu foydalanuvchilar zarur deb hisoblagan, narsalar tuzilishining xaritasidan iborat. Ular BBB tizimlarida, foydalanuvchilar uchun ahamiyatsiz bo'lgan, yanada mayda qismlarga ajratilishi mumkin.

Semantik obyekt model nuqtayi nazaridan, «mohiyat-aloqa» modelida aniqlangan mohiyatlar mavjud emas. Ular aniq mohiyatlarning qismlari yoki bo'laklari, deb qaratadi. Foydalanuvchilar uchun daliliy ma'noga ega bo'lgan yagona mohiyatlar bu – semantik obyektlar deb yuritiladi. Boshqacha aytganimizda, semantik obyektlar «o'ziga to'q» yoki semantik tugallangan hisoblanadi.

8.21-rasmida to'rtta semantik obyektlar BUYURTMA, MIJOZ, SOTUVCHI va MAXSULOT keltirilgan. Agarda foydalanuvchi, «menga V1325 buyurtmani ko'rsating», deb aysa, u keltirilgan diagrammadagi vaziyatni tasavvur qiladi. Bu erda mijoz xususida ham ma'lumotlar mavjud.

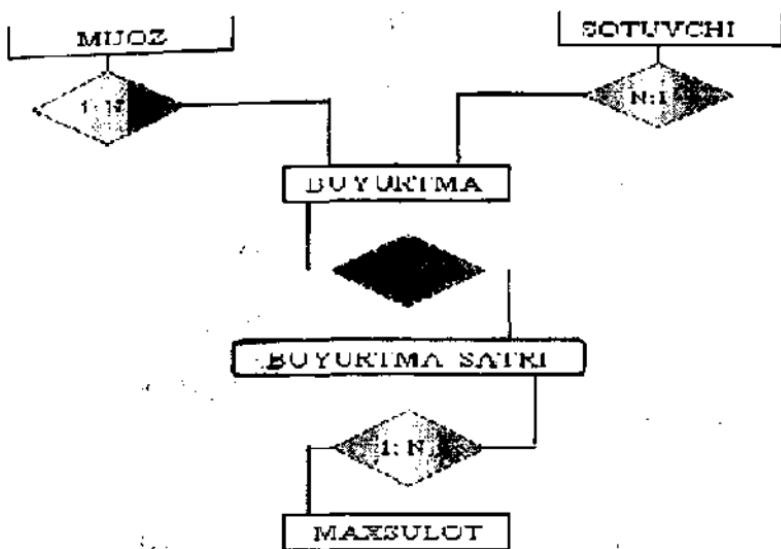
Mijoz xususidagi ma'lumotlar buyurtmaning bir qismi bo'lgani uchun, BUYURTMA obyekti MIJOZ obyektini saqlaydi.

<b>BUYURTMA</b>		<b>MIJOZ</b>	
ID Buyurtma Raqami		ID Mijoz_nomi	
Sonaq		Mijoz_nomi	
<b>MIJOZ</b>	1.1	<b>MIJOZ</b>	
<b>SOTUVCHI</b>	1.1	<b>Indeks</b>	
Buyurtma_sari		Shahar	
Mindiri		Ko'cha	
<b>MAXSULOT</b>		Uy	1.1
Muxta		<b>BUYURTMA</b>	0.N
Oraliq_natiya			
Soliq			
Jami			
<b>SOTUVCHI</b>		<b>MAXSULOT</b>	
ID Sotuvchi_rasqami		ID Maxsulot_rasqami	
Nomi		Nomi	
<b>BUYURTMA</b>	1.N	Tasvif	
		<b>BUYURTMA</b>	0.N

8.21-rasm.

8.22-rasmda xuddi shu berilganlar, ya'ni BUYURTMA, MIJOZ, SOTUVCHI va MAXSULOT mohiyatlaridan iborat «mohiyat-alloqa» modeli keltirilgan. BUYURTMA mohiyati Buyurtma\_Raqami, Sana, Oraliq\_natiya, Soliq va Jami atributlaridan iborat. Endi foydalanuvchi, «menga V1325 buyurtmani ko'rsating», deb aytsa, u umidsizlikka tushgan holda, «qolgan berilganlar qani?» degan savolni berishi mumkin. Chunki, uning tasavvuridagi BUYURTMA mohiyati alohida nodirlilikka mos kelmaydi. Bu mohiyat aniq buyurtmaning faqatgina qismini tasvirlagan bo'lib chiqadi, xolos.

Agarda foydalanuvchi «menga VX11367 mijozni ko'rsating», deb aytsa, u 8.21-rasmdagi mijozga tegishli hamma berilganlarni nazarda tutgan bo'ladi. 8.22-rasmdagi MIJOZ mohiyati Mijoz\_nomi, Indeks, Shahar, Ko'cha va Uy atributlaridan iborat. Agarda foydalanuvchi, «menga Toshev Abbas nomli mijozni ko'rsating», deb aytsa, u yana umidsizlikka tushgan bo'lardi, chunki unga faqat shu ma'lumot berilgandan keyin, «bu men so'ragan berilganlarning faqatgina bir qismi xolos», deb yana norozi bo'ladi.



8.22-rasm.

Semantik obyektlar bilan «mohiyat-aloqa» modelining yana bir farqi, semantik obyektlar mohiyatlarga nisbatan, o'zida ko'p metaberilganlarni saqlaydi. 8.21-rasmida akslangan Mijoz\_raqami atributi foydalanuvchilar tasavvurida noyob identifikator. U mos jadvalda identifikator sifatida ishlatalishi yoki ishlatilmasligi mumkin, lekin bu dalil berilganlar modeli uchun ahamiyatsiz bo'ladi. Semantik obyekt diagrammasi Manzil nomli atributlar guruhni mavjudligini ko'rsatmoqda. Bu guruh boshqa atributlarni ham o'z tarkibiga olgan holda, birlgilikda manzilni hosil qiladi. Ushbu guruhning mavjudligi shakl va hisobotlarni yaratish jarayonida katta ahamiyatga ega. Shuningdek, semantik obyekt diagramma aniq bir MAXSULOT, bir nechta BUYURTMA safiga kiritilgan bo'lishi mumkinligini ko'rsatmoqda, lekin har bir buyurtmada bu mahsulot faqat bitta satrda turishi mumkin. Bu tasdiqni ham «mohiyat-aloqa» diagrammasida akslantirishning iloji yo'q.

### Xulosa

*Semantik obyekt modellar ham, «mohiyat-aloqa» modellari kabi relatsion berilganlar modelini qurish uchun ishlataladi. Bu ikkita modelning farqi, semantik obyekt modellarni yaratishda, yaratuvchilar komandasini bevosita foydalanuvchilar bilan ishlab, ulardan obyekti bo'yicha so'rovlar o'tkazadi, ular bergen hisobotlar, shakllar va so'rovlarni tahsil qiladilar, va*

*shu asosda foydalanuvchilarning berilgan modelini yaratadilar. Keyin ana shu berilganlar modeli asosida berilgan bazasining tuzilishi yaratiladi.*

*Berilgan modelining antiq shakli, uning tuzilishida ishlataladigan antiq konstruksiyalarga bog'liq. Agarda «mohiyat-aloqa» modeli ishlataligan bo'lsa, unda yaratiladigan model mohiyatlar, aloqalar va h.k. iborat bo'ladi. Agarda semantik obyekt model ishlataligan bo'lsa, yaratiladigan model, semantik obyektilar va u bilan bog'liq bo'lgan tuzilishlardan, ya'ni biz shu bobda tavsiflagan konstruksiyalardan tuziladi.*

### **Nazorat uchun savollar**

1. Berilganlarning semantik obyekt modeli asosida nima yotadi?
2. Semantik obyektni ta'riflang.
3. Yetarli tavsif, deganda nimani tushunasiz?
4. Semantik obyekt atributlari nechta turga bo'linadi?
5. Semantik obyekt diagrammasida nimalar keltiriladi?
6. Semantik obyekt diagrammasida minimal va maksimal kardinallik qanday belgilanadi?
7. Semantik obyekt nusxalariga misollar keltiring.
8. Atribut domeni semantik obyekt modelida nimani anglatadi?
9. Berilganlarning semantik obyekt modellari qanday yaratiladi?
10. Obyektlar tasnifiqa nimalar kiradi?
11. Obyektlarning turlarini aytib bering.
12. Gibrid obyektlarni ta'riflang va unga misol keltiring.
13. Assotsiativ obyektlarni ta'riflang va unga misol keltiring.
14. Ajod va ostki tur obyektlarni ta'riflang va unga misol keltiring.
15. Arxetur naql obyektlarni ta'riflang va unga misol keltiring.
16. «Mohiyat-aloqa» modeli va semantik obyekt modellarining o'xshashligi nimada?
17. «Mohiyat-aloqa» modeli va semantik obyekt modellarining farqi nimada?

## **9-bob. AXBOROT TIZIMLARINI MODELLASHTIRISH**

**Tayanch iboralar:** *UML tili, Use-Case – diagrammalar, Package – diagrammalar, Behavior – diagrammalar, Class – diagrammalar, Deployment – diagrammalar, tuzilishli mohiyat, xulqiz mohiyat, guruhlangan mohiyat, qisqa mazmunli mohiyatlar, Rational Rose tizimi, Paradigm Plus tizimi, ARIS tizimi, Together Designer Community Yedition tizimi, joylash tirish (holatlar) diagrammasi, tarkiblar diagrammasi, sinflar diagrammasi UEMT tili.*

### **UML tili**

Axborot tizimlarini modellashdagi muhim o'rinni [56,77,83], UML (Unified Modeling Language) – yagonalashtirilgan modellash tilidan foydalananadigan uslubiyat va tizimlar egallaydi [63,88].

UML – tili murakkab axborotlarga to'liq obyektlarning tizimlarini identifikatsiyalash, namoyishlash, konstruksiyalash va hujjalash uchun ishlataladi. Hozirgi kunda, xalqaro standart sifatida ISO/IEC 19501:2005 «Information technology - Open Distributed Processing - Unified Modeling Language (UML)» ishlataladi.

### **UML tilidagi «mohiyat-aloqa» diagrammalar**

**UML modeli** o'z ichiga quyidagi jihatlarini olgan:

1. **Tuzilishli jihat** – *Use-Case ~ diagrammalar*, tijorat jarayonlari va tijorat tranzaksiyalar, ularning o'zaro aloqasi, qaramlligi va o'zaro harakatini tavsiflovchi; *Package ~ diagrammalar*, predmet soha va tashkilotning iyerarxik tuzilishini tavsiflovchi.

2. **Dinamik jihat** – *Behavior ~ diagrammalar* (Activity, Statechart, Collaboration, Sequence), tijorat jarayonlarining ishlataladigan zaxiralari va olinadigan natijalarga bog'liq ravishdagi vaqt va fazo oralig'idagi o'zaro harakatlarining yashash davridagi xulqini tavsiflovchi.

3. **Statistik jihat** – *Class – diagrammalar*, o'zaro bog'langan obyektlar majmuasini akslantiruvchi, ya'ni predmet sohaning mantiqiy tuzilishini ko'trib chiqish, uning ichki konsepsiyasи, obyektlar iyerarxiyasi va ular orasidagi statistik aloqani tekshirish; *Deployment – diagrammalar*,

boshqarmaning texnologik zaxirasini ko'rsatuvchi.

UML tilining lug'ati 3 turdag'i blokdan iborat:

- mohiyat;
- munosabat;
- diagramma.

UMLda mohiyat – bu modelning asosiy elementi hisoblanadi. Munosabat turli mohiyatlarni bog'laydi; diagrammalar mohiyatlar majmualarini qiziqishlar bo'yicha guruhlaydi.

UML da 4 turdag'i mohiyat turi mavjud:

- tuzilishli;
- xulqiy;
- guruhlangan;
- annotatsiyalangan (qisqa mazmunli).

So'nggi yillarda «mohiyat-aloha» modeli doirasida berilganlar modelini yaratish ancha osonlashdi, chunki ER-diagrammalarini qurish vositalari keng tarqalgan bir nechta CASE-vositalarning tarkibiga kiritilgan. Bunga misollar tarzida IEW, IEF, DEFT, WIN, Visio va Toad Date Modeler programma vositalarini keltirish mumkin. Bu mohiyatlarni munosabatlar bilan ham uyg'unlashtiradi, ya'ni munosabatlar orqali mohiyatlar berilganlar bazasida tavsiflenadi. Bu esa o'z navbatida berilganlar bazasini boshqarish, administratsiyalash va unga xizmat ko'rsatish masalalarining yechimini yengillashtiradi.

UML – bu modellash uslubiyati va obyektga yo'naltirilgan programma mahsulotlari va ilovalarni loyihalar uchun zarur tuzilmadan iborat. Ba'zan UML stilida yozilgan «mohiyat-aloha» diagrammalari turli berilganlarni modellashtirish tizimlarida uchrab turadi. Ana shuning uchun bu til xususidagi ba'zi bir ma'lumotlarni keltiramiz. Shuni ta'kidlashimiz kerakki, berilganlar bazasini loyihalash jarayonida UML tilidagi diagrammalardan ham xuddi ER-diagrammalaridan foydalangan kabi foydalanamiz.

### **UML tilida mohiyat va aloqalar**

7- bobdag'i 7.4–7.6-rasmarda keltirilgan aloqalarni UML tilidagi ko'rinishi mos ravishda 9.1–9.3-rasmarda keltirilgan. Har bir mohiyat mohiyatlar sinfi tarzida uch bo'lakli to'rtburchaklarda tasvirlanadi. Yuqori bo'lakda mohiyatning nomi, keyingisida mohiyatning atributlari, oxirgisida esa shu mohiyatga tegishli cheklashlar va usullar (programma protse-durulari) keltiriladi.

Aloqalar mohiyatlarni bog'lovchi to'g'ri chiziqlar, kardinallik x .. u formatda keltirilgan bo'lib, x – minimumni, u – esa maksimumni ifodalaydi. Masalan, 0 .. 1 ushbu mohiyatning mavjud bo'lishi shart emasligini,

maksimal zarur soni esa 1 ekanligini bildiradi. \* cheksiz ko'plikni bildiradi. Masalan, 1 .. \* bitta mohiyat zarurligini, lekin ishlatalishi mumkin bo'lgan-larning soni cheksiz ko'p ekanligini bildiradi.

TALABA	TALABA-KOMPYUTER	KOMPYUTER
Reyting_daftari_raqami Familiya Ism Ota_ism Fakultet Yo'naliш Kurs Cheklashlar va usullar shu erda keltiriladi	0 .. 1	Kompyuter_raqami Modeli Ishlab_chiqaruvchi Chiqgan_yili Cheklashlar va usullar shu erda keltiriladi

### 9.1-rasm.

USTA	USTA-SHOGIRD	SHOGIRD
Familiya Ism Ota_ism Sohasi Telefon Manzili Cheklashlar va usullar shu erda keltiriladi	1 .. 1	Familiya Ism Ota_ism Telefon Manzili Cheklashlar va usullar shu erda keltiriladi

### 9.2-rasm.

TALABA	TALABA-TOGARAK	TO'GARAK
Reyting_daftari_raqami Familiya Ism Ota_ism Fakultet Manzili Cheklashlar va usullar shu erda keltiriladi	0 .. *	To'garak_nom Familiya Ism Ota_ism Telefon Manzili Cheklashlar va usullar shu erda keltiriladi

### 9.3-rasm.

9.4-rasmda zaif mohiyatlarni tasvirlashga misol keltirilgan. Aloqa chizig'inining ajdod tomonidagi mohiyatga yo'naltirilgan uchi qora romb bilan tasvirlanadi. 7- bodbagi 7.13-rasmdagi RETSEPT mohiyati *zaif*, BEMOR mohiyati esa *ega* mohiyat. Hamma tobe mohiyatlarning ajdodi

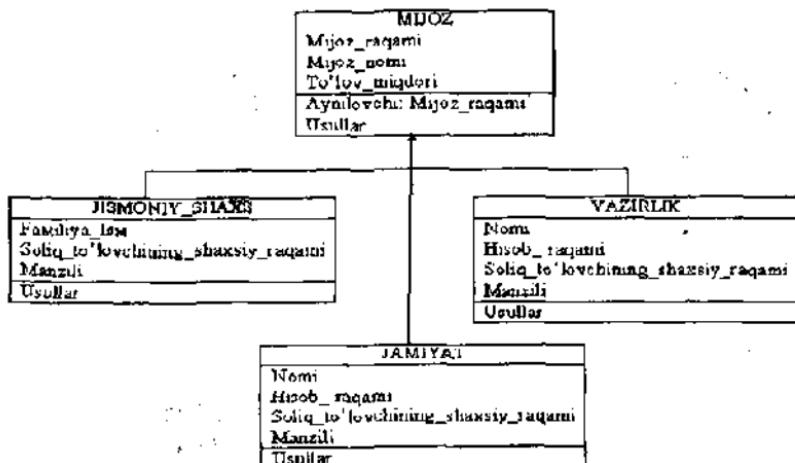
bo'ladi, shuning uchun ularning ajdodlari bilan aloqasi doimiy ravishda 1 .. 1 bo'ladi. Shu asosda, ajdod tomonidagi kardinallik 1 bo'ladi.



9.4-rasm.

9.4-rasmidagi RETSEPT zaif mohiyati BEMOR mohiyatiga identifikatsiyalangan bog'liq emas. Bu diagrammada <identifikator emas> ifodasi bilan belgilangan.

Mohiyatlarning ostki va ustki turlarini tasvirlash 9.5-rasmida keltirilgan. Ushbu rasmida MIJOZ mohiyatining ostki turlari 7- bobdag'i 7.17-rasmida keltirilgan JISMONIY - SHAXS, JAMIYAT va VAZIRLIK mohiyatlaridan iborat. Keltirilgan diagrammaga asosan har bir mijoz bitta, ikkita yoki uchta ostki turdan iborat bo'lishi mumkin. Har bir vaziyat uchun bu ma'noga ega bo'lmagan holat, chunki yuqorida ta'kidlaganimizdek mijoz faqtgina bitta ostki turdag'i mohiyat bilan bog'lanishi mumkin. Bu vaziyatni, ya'ni mohiyatlarning o'zaro bir-birini o'mini bosish imkoniyatini, UML tilida tasvirlash imkoniyati mavjud emas.



9.5-rasm.

UML tilidagi «mohiyat-alloqa» diagrammalarini berilganlar bazalarini loyihalash jarayonida, ER-diagrammalardan foydalangan kabi ishlatalish mumkin.

Berilganlarni modellashtirish jarayonini yaxshi o'zlashtirish uchun, turli xilda keltirilgan misollarni o'rganish va ER-diagrammalardan foydalangan holda mustaqil ravishda shaxsiy modellarni yaratish kerak.

### **UML tilining asosiy xususiyatlari**

*Tuzilishli mohiyat* – UML tilidagi modellardagi mavjud nomlar. Odatta, ular tizimning konseptual yoki fizik elementlariga mos keluvchi modelning statistik qismini tavsiflaydi. Tuzilishli mohiyatning 7 xil turi mavjud: Sinf, Aloqa, Kooperatsiya, O'xhash hodisa, Faol sinf, Tarkib, Tugun.

*Xulqiy mohiyat* – UML modelining dinamik tarkibi. Bu tilning fe'llari, modelning vaqt va fazodagi xulqini tavsiflaydi. Xulqiy mohiyatning faqatgina ikkita asosiy turi mavjud – *o'zaro harakatli va avtomatik*.

*Guruhlangan mohiyat* – UML modelining boshqarish qismini guruhalash uchun xizmat qiladi. Modelni ana shu bloklarga yoyish mumkin. Faqat bitta birinchi guruhlangan mohiyat mavjud bo'lib, u - paketdir.

*Qisqa mazmunli mohiyatlar* – UML modelining tushuntiruvchi qismidan iborat. Bu qisqa izoh bo'lib, qo'shimcha yozuvlar uchun mo'ljallangan, uning yordamida modelning ixtiyoriy elementiga aniqlik kiritish yoki izoh berish mumkin.

Mohiyatlarning barcha xillari UML diagrammalarida o'zining grafik tasviriga ega. UML tilida munosabatning 4ta turi mavjud:

- tobe'lik;
- assotsiatsiya;
- umumlashtirish;
- amalga oshirish.

UMLda diagramma – bu elementlar to'plamining graf shaklidagi ko'rinishi. Grafning barcha uchlari (mohiyat) va qirralari (munosabat)larni tasvirlaydi. Diagrammalarni turli nuqtalardan tizimga nazar solish uchun namoyish qilinadi. Nazariy jihatdan diagrammalar mohiyat va munosabat-larning ixtiyoriy kombinatsiyalaridan iborat. Amaliyotda axborot tizimlari arxitekturasini tashkil qiladigan 5ta eng ko'p ishlataladigan kombinatsiyalardan iborat tur ishlataladi. Shunday qilib, UMLda 9ta turdag'i diagrammalarni ajratishadi:

- sinflar diagrammalari (Class Diagrams);
- obyektlar diagrammalari (Objects Diagrams);
- o'xhashliklar diagrammalari (Use Cases Diagrams);

- ketma-ketliklar diagrammalari (Sequence Diagrams);
- kooperatsiyalar diagrammalari (Collaboration Diagrams);
- holatlar diagrammalari (State Diagrams);
- harakatlar diagrammalari (Activity Diagrams);
- tarkiblar diagrammalari (Component Diagrams);
- kengaytirishlar diagrammasi (Deployment Diagram).

UML uslubiyatiga xizmat qiladigan instrumental vositalar – Rational Rose (Rational Software), Paradigm Plus (CA/Platinum), ARIS (IDS Sheer AG), Together Designer (Borland) va boshqalar.

Rational Rose tizimi 9ta turdag'i UML diagrammalarni hosil qilish imkonini beradi - sinflar, o'xshashliklar, obyektlar, kooperatsiyalar, ketma-ketliklar, holatlar, harakatlar, tarkiblar va kengaytirishlar. UMLda diagrammalarning asosiy turi, ya'ni yadrosini sinflar diagrammalari tashkil etadi. UMLdan tashqari boshqa usullardan ham foydalanish imkoniyatlari berilgan (Booch, OMT).

Paradigm Plus tizimi OOCL (Object Oriented Change and Learning) uslubiyatiga hamda loyihalash va yaratishning tarkibli texnologiyasiga yo'naltirilgan. U turli usullar diagrammalarini qo'llash imkoniyatiga ega (UML, CLIPP, TeamFusion, Booch, OOCL, Martin/Odell, OMT, Shlaer/Mellor, Coad/Yourdon).

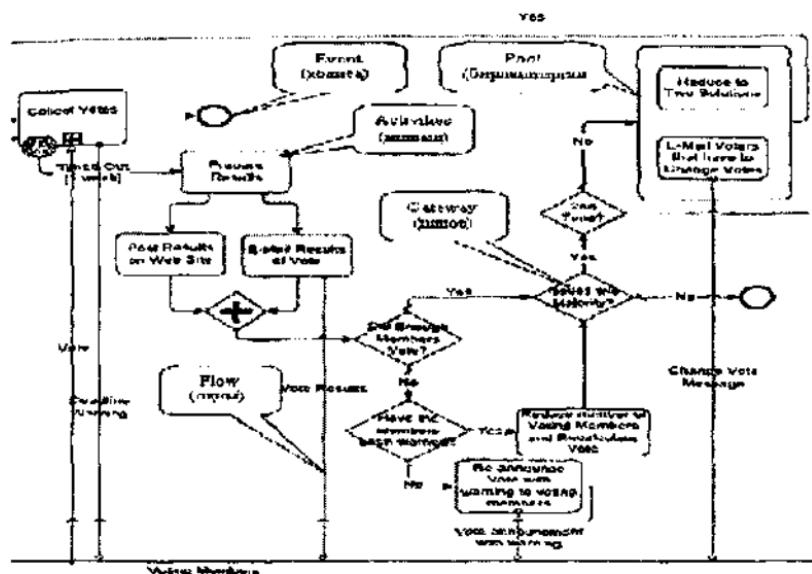
ARIS tizimi modellash va tahlilning to'rtta turini mujassamlashtirgan: jarayonlar, funksiyalar (maqsadlar bilan), berilganlar, tashkilot. Hari bir tur uchun tahlilning uchta darajasi qo'llaniladi (talab, identifikatsiyalash, joriy qilish). Tahlilning har bir darsasi modellar majmuasining turli xillaridan iborat, shu jumladan UML diagrammalaridan, SAP/R3 diagrammalaridan va boshqalardan. ARIS modelining har bir obyekti ko'pgina xususiyatlardan iborat bo'lib, ular modellarni yaratish jarayonini nazorat qilish imkonini, funksional qiymatli tahlilni bajarish shartlarini aniqlash, modellashni imita-tsiyalash, workflow-tizimlar bilan o'zaro harakatlanish va h.k. imkoniyatlarni beradi.

Together Designer Community Yedition – tizimi UML 2.0 diagrammalini yaratish vositalaridan iborat. O'xshashliklar diagrammalarini qurish, axborotlarni bir obyektdan ikkinchisiga uzatish tartibini tavsiflovchi ketma-ketliklar diagrammalarni, ishlar oqimi va obyekt holatlarining o'zgarishining tavsiflovchi harakatlar diagrammalarni va kengayishlar diagrammalarni tavsiflash imkoniyatini beradi. Zaruriy holatda, «mohiyat-aloha» diagrammalarini saqlovchi berilganlarning mantiqiy modellarini yaratishi, ular asosida esa tanlangan loyihami amalda aniq BBBT uchun berilganlarning fizik modeli generatsiya qilinadi.

UML - modellashtirish zamонави vositalarining mijoz va server programmlari turli tillardan foydalangan holda generatsiya qilinadi.

2006-yil fevralda Object Management Group (OMG) [40], Inc.- kompyuter industriyalari bo'yicha identifikatorlarni ishlab chiqish xalqaro konsersiumi - BPMI (Business Process Modeling Language)ning [41] tijorat jarayonlarni modellash tilining eng oxirgi naqlini taqdim etdi.

Bu tilning asosiy grafik elementlari quyidagi 9.6-rasmida ko'rsatilgan.



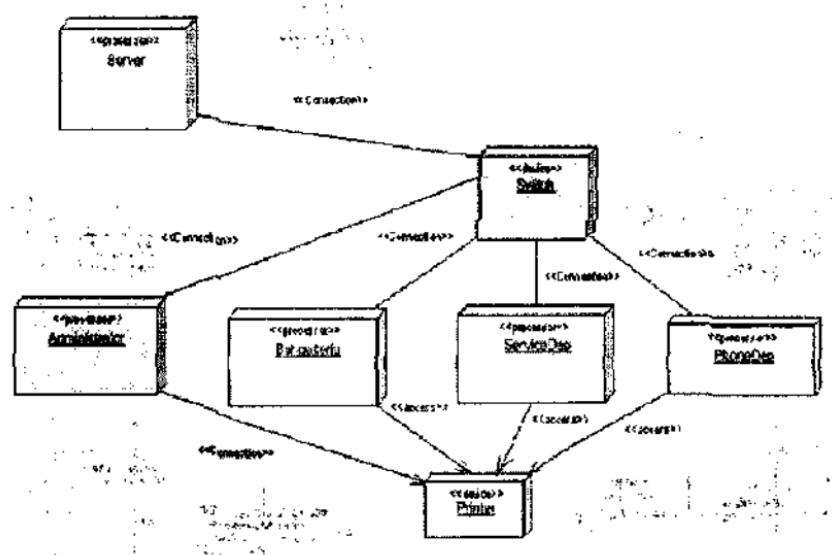
**9.6-rasm. BPML standarti bo'yicha tasvirjangan tijorat jarayoni modelining fragmenti.**

## UML tilida modellashga misollar

UML tilida axborot tizimini modellash jarayonini ham amalga oshirish mumkin. Masalan, All Fusion Data Modeling Suite paketining tarkibiga kiruvchi [83] Component Modeler muhitida joylashtirish, sinflar va tarkiblar diagrammalariga misollar keltirilmoq.

*Joylashtirish (holatlar) diagrammasi.* Joylashtirish diagrammasini qurishda uchta turdag'i shartli-grafik belgilash usulidan foydalaniladi. 9.7-rasmda «Mehmonxona» chizmasini tarmog'i keltirilgan. Tarmoq beshta kompyuterni o'zaro bog'laydi, ya'ni «server», mehmonxona administratori, hisobchi, xizmat ko'rsatish va telefonidagi so'zlashuvlarni hisobga olishi bo'limlaridagi kompyuterlarni. Mehmonxona administratorining kompyuteri

teriga printer ulangan, qolgan xizmatchilar esa ma'lumotlarni tarmoq orqali chop etishlari mumkin.



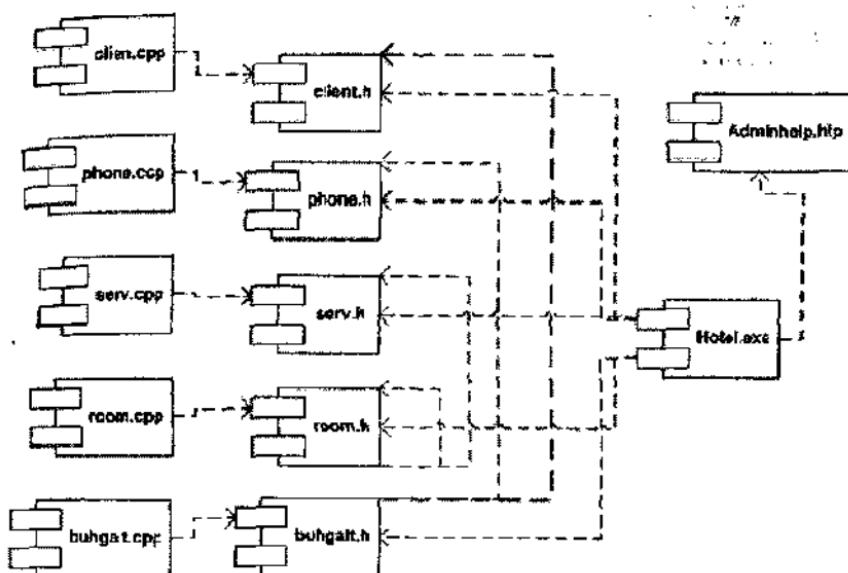
9.7-rasm. Joylashtirish diagrammasi.

*Tarkiblar diagrammasi.* Tarkiblar diagrammasi, model fizik darajada qanday ko'rinishini ifodalaydi. Diagrammada programma ta'minotidagi tarkiblar va ular orasidagi aloqaqlar ifodalanadi. Bunda bajariluvchi tarkiblar va kod kutubxonalarini ajratib ko'rsatiladi. Tarkiblar diagrammasiga misol 9.8-rasmida keltirilgan. Har bir sınıf o'zining \*.SRR kengaytirmali fayliga va sarlavhali fayliga ega bo'ladi. Shunday asnoda diagrammadagi har bir sınıf o'zining tarkiblariga o'zgartiriladi. Masalan, Client sınıfı ikkita tarkibdan iborat bo'ladi, ya'ni client.h va client.cp. Bularning ikkalasi birgalikda Client sınıfining tanasi va sarlavhasini taqdim etadi. Hotel.exe tarkibi esa, axborotlarga ishllov berish oqimini (thread of processing) bildiradi. Bizning holatda bu ishllov oqimi – bajariluvchi programmadan iborat.

*Sinflar diagrammasi.* Diagrammada to'rtta sınıf keltirilgan. Har bir sınıf o'zining usullarini harakatini tavsiflovchilariga ega (9.9-rasm). Masalan, Client sınıfining usullari:

- Search() – mijozlarni izlash uchun;
- Delete() – mijoz haqidagi berilganlarni o'chirish uchun;
- New() – yangi mojozlarni ro'yxatga kiritish uchun;
- CostRoom() – mehimonxonada yashash uchun ketgan umumiylar xarajatlarni hisoblash uchun.

Phone sinfida esa, suhbatlashish vaqtini – Time() va qaysi raqamli telefondan qo'ng'iroq bo'lganini aniqlash uchun – Number() usullaridan foydalaniлади.



9.8-rasm. Tarkiblar diagrammasi.

### UEML tili

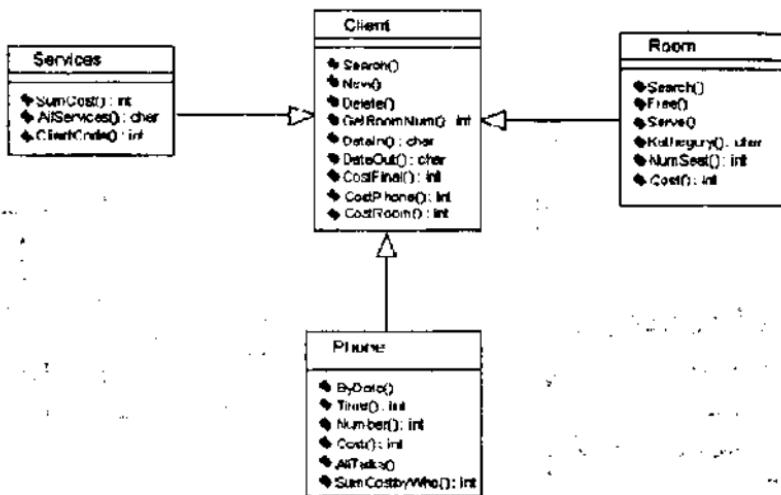
Axborot tizimlarini modellashtirishda ishlataladigan eng oxirgi yaratilgan programma mahsulotlaridan biri, bu unifikatsiyalangan modellash uchun maxsus yaratilgan UEML (Unified Enterprise Modeling Language) tilidir. UEMLnini yaratish tarmoq loyihasi boʼlib - (IST-2001-34229), Yevro Ittifoqi tomonidan moliyalashtirilgan [42].

UEML loyihasi quyidagilarni yaratishni oʼz ichiga olgan:

- tilning andozalari orqali korxonalarni modellash instrumental vositalari orqali umumiy, vizual va workflow - sinflarning programma tizimlarini;
- standartlashtirilgan, loyihalar orasida vositalarga bogʼliqsiz ravishda modellarni uzatish vositalarini;
- korxona modellarining repozitariyalarini.

ATning BB loyihalashning konseptual modeli va fizik modelini shakllantirishda instrumental CASE (Computer-Aided Software System

Engineering) – vositalardan, masalan Case Studio, SyBase Power Designer, ERWin Data Modeler va boshqalardan foydalanish mumkin.



9.9-rasm. Sinflar diagrammasi.

Ushbu tizimlar berilganlar modelini tavsiflashda IDEFIX standartini ishlataladi va SQL, VBScript, Jscript tillarida programmalar kodlarini generatsiya qilish yoki fizik modellarni aniq BBBT ko'chirishda Oracle, Microsoft SQL Server, IBM DB2, Informix, Microsoft Access texnologiyalari va boshqalar bilan ishslash imkoniyatini beradi.

### Xulosa

*Axborot tizimlarini modellashtirishda ishlataladigan UML tili «mohiyat-aloqa» diagrammalariga asoslangan bo'lib, zamonaviy Case vositalarida juda keng qo'llanilmoqda. UML tilining arxitekturasi juda keng miqyosli bo'lib, 9 xildagi diagrammalar orqali obyektlarning tizimlarini identifikatsiyalash, namoyishlash, konstruksiyalash, hujjatlash va h.k. uchun ishlatiladi.*

*UEML tili esa, UML tilining uzviy davomchisi bo'lib, yaratilgan modellarni boshqa multitlarga konvertatsiya qilish va keng tarqalgan BBB tizimlari uchun programma kodlari asosida berilganlar bazalarining jadvallari bilan ishni davom ettirish imkoniyatini yaratadi.*

## Nazorat uchun savollar

1. Axborot tizimlarini modellashtirishning asosiy maqsadi nimadan iborat?
2. UML tilidagi asosiy jihatlarini aytib bering.
3. UML tilida necha xil bloklar ifodalanadi?
4. UML tilida nechta mohiyatlarni turi mayjud?
5. UML tilida mohiyatlar va ular orasidagi aloqalarni belgilashga misol keltiring.
6. UML tilida zaif va kuchli (ega) mohiyatlar va ular orasidagi aloqalarga misol keltiring.
7. UML tilida nechta turdag'i diagrammalarni yaratish mumkin?
8. UML tilida xizmat qiluvchi instrumental vositalarni aytib bering.
9. BPML tilidagi asosiy jarayonlar nechta turdan iborat?
10. Axborot tizimlarini modellashtirishda ishlataladigan eng so'nggi til qaysi?

---

### III BO'LIM. RELATSION ALGEBRA VA NORMALLASH

#### 10-bo'b. Relatsion algebra

**Tayanch iboralar:** nomsiz munosabatlar, bir xildagi chizmalar, bir xildagi domenlar, kesishish amali, birlashish amali, ayirish amali, dekart ko'paytmasi amali, tanlash amali, nusxalash amali, bo'lisk amali, tutashish amali, tabiiy tutashish amali, yarim tutashish amali, ekvivalentlik bo'yicha tutashish,  $\Theta$ -tutashish amali, tashqi tutashish amali, tashqi chap tutashish amali, tashqi o'ng tutashish amali, tashqi to'liq tutashish amali.

Relatsion BB moslashuvchanligi munosabatlar ustida o'tkaziladigan o'zgartirishlarning yengil amalga oshishi bilan bog'liq. Buning natijasida, keyinchalik konseptual model foydalanuvchining modeli bilan bir xil bo'lmasligi mumkin. Foydalanuvchining munosabatlari chizmadagi munosabatlardan kelib chiqishi kerak. Matematik amallardan foydalangan holda, munosabatlar ustida turli o'zgartirishlarni amalga oshirish mumkin.

Munosabatlar «haqiqiy dunyo»ning qaysidir bir qismidagi haqiqiy obyektlarini akslantiradi. Dunyoning ixtiyoriy qismi esa vaqt o'tishi bilan o'zgaradi. Shuning uchun munosabatlar ham vaqt o'tishi bilan o'zgaradi, ya'ni yangi kortejlar qo'shilish, o'zgarish yoki yo'q qilinishi mumkin. Falsafiy kategoriyalar zamon va makon bilan uzviy bog'liqligiga misol bo'la oladi. Demak, vaqt oraliq'ida munosabatlar bir holatdan ikkinchi, uchinchi va h.k. holatlarga o'tadi. Lekin munosabat qaysi holatga o'tmasin, uning chizmasi o'zgarmaydi.

Shuningdek, domenlarni tartiblash axborot ma'nosida munosabatning ma'nosini o'zgartirmaydi.

Har bir munosabat o'zi aniqlangan berilganlar bazasi doirasida noyob ismga ega bo'lishi shart. Munosabatning relatsion amal natijasida hosil bo'ladigan nomini, tenglik ishorasining chap tomonida yozamiz. Agarda, relatsion amal natijasida hosil bo'lgan munosabat yana bir boshqa relatsion amalda ishtirok etsa, unga nom qo'yish shart emas. Bunday munosabatlarni deb atashadi. Nomsiz munosabatlar fizik jihatdañ berilganlar bazasining tarkibida ishtirok etmaydi, faqatgina relatsion operatorning qiymatini hisoblash davridagiga hisoblanadi xatosi.

E.F. Kodd [15] va keyinchalik boshqalar an'anaviy tarzda sakkizta relatsion amallarni ikkita guruhga ajralgan holda aniqlashadi.

### Nazariy-to'plam amallari:

- kesishish;
- birlashish;
- ayirish;
- dekart ko'paytma.

### Maxsus relatsion amallar:

- tanlash;
- nusxalash;
- birlashtirish;
- bo'lish.

Bu amallarning hammasi ham bog'liqsiz emas, chunki bu amallarning ayrimlarini boshqa relatsion amallar orqali keltirib chiqarish mumkin.

### Munosabatlar ustidagi amallar

Munosabatlar ustida bajariladigan maxsus amallardan oldin, ba'zi bir oddiy to'plam amallarini munosabatlarga tatbiq etishni o'rganishdan boshlaymiz.

To'plamlar ustida bajariladigan ayrim to'plam amallarini munosabatlarga ham qo'llash mumkin. Buning uchun munosabatlar ega bo'lishi talab qilinadi. Bu esa o'z navbatida bir xil nomdagi atributlarning *bo'lishini* talab qiladi.

Shuning uchun, misol tariqasida **R** va **S** munosabatlarni (**A,V,S**) chizmaga asosan, quyidagicha holatda qabul qilamiz (10.1-rasm) va to'plam amallarini bajarishda ana shu munosabatlardan foydalanamiz:

R	A	B	C
a <sub>1</sub>	B <sub>1</sub>	c <sub>1</sub>	
a <sub>2</sub>	B <sub>1</sub>	c <sub>2</sub>	
a <sub>3</sub>	B <sub>2</sub>	c <sub>1</sub>	
a <sub>4</sub>	B <sub>2</sub>	c <sub>2</sub>	

S	A	B	C
a <sub>1</sub>	B <sub>1</sub>	c <sub>2</sub>	
a <sub>1</sub>	B <sub>2</sub>	c <sub>2</sub>	
a <sub>2</sub>	B <sub>3</sub>	c <sub>1</sub>	
a <sub>4</sub>	B <sub>2</sub>	c <sub>2</sub>	

10.1-rasm.

### Kesishish amali

Bir xildagi sarlavhaga ega bo'lgan **R** va **S** munosabatlar uchun *kesishish* (*intersection*) amalining natijasidagi **T** = **R**  $\cap$  **S** munosabat ham ana shu sarlavhaga ega bo'ladi. **T** munosabat tanasi **R** va **S** munosabatlarning bir xildagi kortejlaridan hosil bo'ladi.

Bu erda va keyingi munosabatlar orasidagi = belgisidan shartli ravishda yangi munosabatni ifodalash ma'nosida foydalanamiz.

Kesishish amali natijasida, T munosabat quyidagicha hosil bo'ladi:

A	B	C
a <sub>1</sub>	b <sub>2</sub>	c <sub>3</sub>

### Birlashish amali

Birlashshish (union) amali R U S natijasida, T munosabat R va S munosabatga tegishli kortejlardan hosil bo'ladi, lekin ikkala munosabatda qatnashgan bir xildagi kortejlar yangi munosabatda bir marta qatnoshadi.

Birlashish amali natijasida T = R U S, T munosabat quyidagicha hosil bo'ladi:

A	B	C
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>
a <sub>2</sub>	b <sub>1</sub>	c <sub>2</sub>
a <sub>3</sub>	b <sub>2</sub>	c <sub>1</sub>
a <sub>1</sub>	b <sub>1</sub>	c <sub>2</sub>
a <sub>1</sub>	b <sub>2</sub>	c <sub>2</sub>
a <sub>2</sub>	b <sub>3</sub>	c <sub>1</sub>
a <sub>4</sub>	b <sub>2</sub>	c <sub>2</sub>

### Ayirish amali

Munosabatlarni ayirish (difference) amali R - S natijasida, T munosabat R ga tegishli, lekin S munosabatga tegishli bo'lmagan kortejlardan hosil bo'ladi.

Ayirish amali natijasida T = R - S, T munosabat quyidagicha hosil bo'ladi:

A	B	C
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>
a <sub>2</sub>	b <sub>1</sub>	c <sub>2</sub>
a <sub>3</sub>	b <sub>2</sub>	c <sub>1</sub>

### Dekart ko'paytmasi amali

Ikkita munosabatni ko'paytirish amalini *dekart ko'paytmasi (cartesian product)* amali deb yuritishadi. Yangi munosabat ( $m * n$ ) o'chovli, birinchi munosabatdagi hamma kortejlarni ( $m$  ta) ikkinchi munosabatdagi ( $n$  ta) hamma kortejlar bilan tutashuvidan, iborat bo'ladi. Ko'paytma  $x$  – belgisi bilan belgilanadi.

10.2-rasmdagi **RP** munosabati ikkita kortejdan, **SP** munosabati esa uchta kortejdan iborat. Yangi munosabati  $T = RP \times SP$  6ta kortejdan iborat bo'ladi (10.3-rasm).

**RP**

A	B
a <sub>1</sub>	B <sub>1</sub>
a <sub>2</sub>	B <sub>1</sub>

**SP**

C	D
c <sub>1</sub>	d <sub>1</sub>
c <sub>1</sub>	d <sub>2</sub>
c <sub>2</sub>	d <sub>1</sub>

10.2-rasm.

**T**

A	B	C	D
a <sub>1</sub>	B <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>
a <sub>1</sub>	B <sub>1</sub>	c <sub>1</sub>	d <sub>2</sub>
a <sub>1</sub>	B <sub>1</sub>	c <sub>2</sub>	d <sub>1</sub>
a <sub>2</sub>	B <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>
a <sub>2</sub>	B <sub>1</sub>	c <sub>1</sub>	d <sub>2</sub>
a <sub>2</sub>	B <sub>1</sub>	c <sub>2</sub>	d <sub>1</sub>

10.3-rasm.

### Tanlash amali

*Tanlash (selection)* amali munosabat ustidagi unar amaldan iborat, chunki u bitta munosabat ustida shartga asosan (predikat) bajarilgan tanlash amali naijasida olingan kortejlarni chiqaradi. Bu amal berylganlarni izlashda juda muhim ahamiyatga ega. Shuning uchun ushbu amal

ustida batafsil to'xtab o'tamiz. Tanlash amali, alohida yoki bir yo'la kortejlar ustida izlanishlar olib boradi. Bu amalning bajarilishi natijasida, qo'yilgan shart qanoatlantirilsa, R munosabatning qismi shaklida yangi munosabat tuziladi, aks holda natijaga ega bo'linmaydi. R munosabat yuqoridagi keltirilgan (A,V,S) chizmaga hamda munosabat o'sha elementlarga ega, atributlarning nomi A, V va S bo'lzin.

*Shart bo'yicha oddiy tanlov.* R munosabatdan, S atributning qiymati  $c_2$  ga teng bo'lgan kortejlar tanlansin, deyilgan shartni, shartli ravishda rasmiylashtirib, «Rdan R.C =  $c_2$  TANLANSIN» shakldagi so'rovga aylantiramiz. Natijada quyidagi munosabat hosil bo'ladi:

A	B	C
$a_2$	$b_1$	$c_2$
$a_4$	$b_2$	$c_2$

Keltirilgan shartli so'rovda (predikat shartida) biz tenglik (=) amalidan foydalandik. Yana predikatda quyidagi taqqoslash belgilaridan foydalanimiz mumkin:

- > katta
- >= katta yoki teng
- < kichik
- <= kichik yoki teng
- <> teng emas.

Ulardan atributlar ustida bo'ladigan so'rovlarni shakllantirishda foydalanish mumkin.

*Shart bo'yicha oddiy tanlov.* MUXANDIS munosabatidan (10.4-rasm) maoshi 1500 dan ko'p bo'lgan xizmatchilar tanlansin. Shartli so'rovimiz, «MUXANDISdan MUXANDIS.MAOSH>1500 TANLANSIN» shaklida bo'ladi.

### MUXANDIS

XIZMATCHI RAQAMI	F. SH.	BO'LIM. RAQAMI	MAOSH.	MANZIL.
53704	ТСШЕЕ Д.Б.	721	1200	НАВСИЙ
53703	РУЗИЕВ Е.С.	721	2300	НАВСИЙ
53791	МАШАРОПОВ М.	807	5000	ГУЛМСТОН
53800	ЦОН Е.	402	1100	КАРШИ
53803	ДАМИКОВ Д.Х.	721	1900	НАВСИЙ
53806	АСОМОВ П.Л.	402	1000	КАРШИ

So'rov natijasida olinadigan yangi munosabat quyidagicha bo'ldi:

XIZMATCHL RAQAMI	FISH	BO'LIM. RAQAMI	MAOSH	MANZIL
53703	РУЗИЕВ Е.С.	721	2300	НАВОЙИ
53791	МАШАРИПОВ М.	007	5000	ГУЛИСТОН
53805	ДАМИНОВ Д.Х.	721	1900	НАВОЙИ

Mantiqiy Bul amallari, berilganlar bazalarining jadvallari bilan ishlashda juda muhim ahamiyatga ega. Ayniqsa, asosiy uchta AND, OR va NOT orqali turli sodda (bitta atribut ustida) va tarkibli predikatlardan iborat murakkab (bir nechta atributlar ustida) so'rovlarni shakllantirish mumkin.

Mantiqiy Bul amallarining ishslash algoritmi quyidagicha:

- AND (VA) amali A AND V shakldagi, A va V ifodalarni argument sifatida olib, ularni Bul qiyamat («haqiqat» va «yo'lq'on») larga tekshiradi. Natijada, ikkala argument haqiqiy bo'lsagina, A AND V shart bajarilgan hisoblanadi;
- OR (YOKI) amali A OR V shakldagi, A va V ifodalarni argument sifatida olib, ularni kamida bittasi haqiqiyligini tekshiradi, shunda shart bajarilgan hisoblanadi;
- NOT (AKSINCHA) amali NOT A shakldagi, A ifodani argument sifatida olib, natijada olingan bul qiyamatini teskarisiga o'zgartiradi.

MUXANDIS munosabatidan (10.4-rasm) va mantiqiy Bul amallaridan quyida foydalanamiz.

Shart bo'yicha oddiy tanlov. Qarshi yoki Gulistonida bo'limlarda ishlavchi muhandislar aniqlansin. Shartli so'rov «MUXANDISdan (MUXANDIS.MANZIL = 'QARSHI') OR (MUXANDIS.MANZIL = 'GULISTON') TANLANSIN».

Natija:

XIZMATCHL RAQAMI	FISH	BO'LIM. RAQAMI	MAOSH	MANZIL
53791	МАШАРИПОВ М.	007	5000	ГУЛИСТОН
53800	ЦОЙ В.	402	1100	КАРШИ
53806	АСОМОВ П.Л.	402	1000	КАРШИ

Shart bo'yicha oddiy tanlov. Maoshi 1500dan kam xodimlar aniqlansin. Shartli so'rov «MUXANDISdan NOT (MUXANDIS.MAOSH > 1500) TANLANSIN».

Natija:

XIZMATCHL RAQAMI	FISH	BO'LIM. RAQAMI	MAOSH	MANZIL
53800	ЦОЙ В.	402	1100	КАРШИ
53806	АСОМОВ П.Л.	402	1000	КАРШИ

*Shart bo'yicha murakkab tanlov.* 721 bo'limdagi 1500 va undan ortiq maosh oladigan xizmatchilarni tanlashga shartli so'rov yozamiz. «MUXANDISdan (MUXANDIS.BO'LIM\_RAQAMI=721) AND (MUXANDIS. MAOSH>=1500) TANLANSIN».

Natija:

XIZMATCHL. RAQAMI	FISH	BO'LIM. RAQAMI	MAOSH	MANZIL
53703	РУЗИЕВ Е.С	721	2300	НАВОЙИ
53805	ДАМИНОВ Д.Х	721	1900	НАВОЙИ

*Shart bo'yicha murakkab tanlov.* Muhandislardan 1500dan ko'p, lekin 5000dan kam oylik oladiganlar tanlansin. Shartli so'rov «MUXANDISdan (MUXANDIS.MAOSH>1500) AND (MUXANDIS. MAOSH <=5000) TANLANSIN».

Natija:

XIZMATCHL. RAQAMI	FISH	BO'LIM. RAQAMI	MAOSH	MANZIL
53703	РУЗИЕВ Е.С	721	2300	НАВОЙИ
53791	МАШАРИПОВ М.	007	5000	ГУЛИСТОН
53805	ДАМИНОВ Д.Х	721	1900	НАВОЙИ

*Shart bo'yicha murakkab tanlov.* Navoiyda ishlovchi maoshi 2000dan ortiq bo'lgan, xodimlar aniqlansin. Shartli so'rov «MUXANDISdan (MUXANDIS.MAOSH >2000) AND (MUXANDIS.MANZIL = 'NAVOIY') TANLANSIN».

Natija:

XIZMATCHL. RAQAMI	FISH	BO'LIM. RAQAMI	MAOSH	MANZIL
53703	РУЗИЕВ Е.С	721	2300	НАВОЙИ

### Nusxalash amali

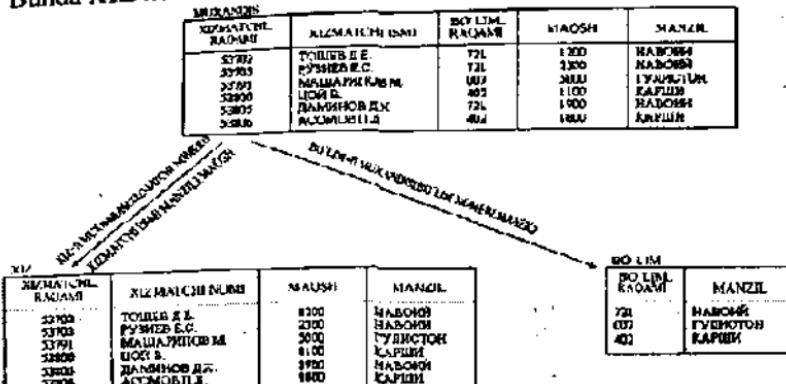
*Nusxalash (projection)* amali jadvaldan kerakli ustunlarni tanlab, yangi jadval qanday tartibda kelishini belgilash imkoniyatini beradi. Shartli ravishda **Π** bilan belgilaylik.

10.5-rasmda MUXANDIS jadvalining ikkita jadvalga ajralgani keltirilgan. Amallarning shartli ko'rinishi:

XIZ = **Π** (XIZMATCHI\_RAQAMI, XIZMATCHI\_NOMI, MAOSH, MANZIL) (MUXANDIS)

BO'LIM = **Π** (BO'LIM\_RAQAMI, MANZIL) (MUXANDIS)

Bunda XIZ nomi bilan XIZMATCHI\_RAQAMI, XIZMATCHI\_NOMI,



10.5-rasm. MUXANDIS munosabatining ikkita munosabatga ajralishi

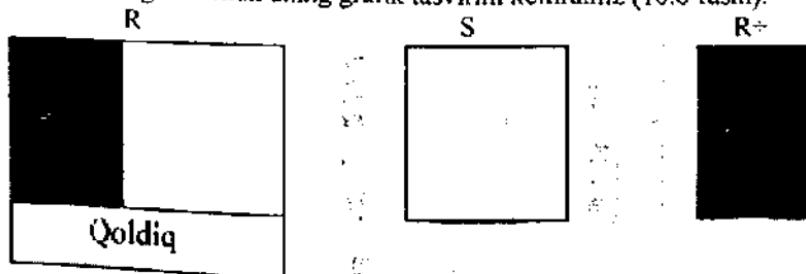
MAOSH VA MANZIL nomlaridagi berilganlar elementlaridan iborat jadval hosil bo'ldi.

Ikkinchisi jadval esa, BO'LIM nomi bilan BO'LIM\_RAQAMI va MANZIL nomlaridagi atribut elementlaridan tuziladi.

*Nusxalash* amalining natijasida, yangi jadvallar (10.5-rasm)dagi BO'LIM jadvali kabi, takrorlanadigan kortejlar bo'lmaydi.

### Bo'lish amali

*Bo'lish amali* algoritmini tushunish biroz qiyinroq. *Bo'lish amali* munosabatning «tik-yotiq» to'plamini tuzishga, ya'ni bo'linuvchi munosabatning «tik» qism to'plamini, bo'lувчи munosabatning «yotiқ» holati bilan kesilishi natijasidagi, «tik» qoldiqdan iborat. *Bo'lish amalida* qoldiq tushunchasi ishlatilgani uchun uni + bilan belgilashadi. *Bo'lish amalini* tushunish murakkab bo'lgani uchun uning grafik tasvirini keltiramiz (10.6-rasm).



10.6-rasm.

Endi munosabatlar orqali bo'lish amalini tushuntiramiz. Faraz qilaylik, umumiy  $Y_1, Y_2, \dots, Y_m$  atributlarga ega bo'lgan  $R(X_1, X_2, \dots, X_n, Y_1, Y_2, \dots, Y_m)$  va  $S(Y_1, Y_2, \dots, Y_m)$  munosabatlar berilgan va hamma  $(y_1, y_2, \dots, y_n) \in S$  kortejlar uchun  $R$  munosabatda  $(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n)$  kortej mavjud bo'lsin.  $R$  munosabati  $S$  munosabatga bo'lish deganda, sarlavhasi  $(X_1, X_2, \dots, X_n)$  va tanasi  $(x_1, x_2, \dots, x_n)$  iborat bo'lgan yangi munosabat tushuniladi. Munosabatlarni bo'lishni butun sonlarni qoldiqli bo'lish bilan, «tik» qoldiqni esa qoldiq bilan taqqoslashimiz mumkin.

*Bo'lish amali* xususida yaxshi tasavvurga ega bo'lish uchun misollarni turli munosabatlar ustida keltiramiz.

1- misol. TA'MINOTCHI munosabati va STATUS munosabatlari berilgan bo'lsin (10.7-rasm).

TA'MINOTCHI

TA'MINOTCHI RAQAMI	TA'MINOTCHI ISMI	STATUS	MANZIL
N1	Карим	30	Тошкент
N2	Салим	20	Нукус
N10	Халим	40	Нукус
N17	Халла	30	Тошкент
N18	Марса	40	Андижон

STATUS

STATUS	MANZIL
30	Тошкент

10.7-rasm.

TA'MINOTCHI + STATUS natijasida, quyidagi munosabat hosil bo'ladi:

TA'MINOTCHI RAQAMI

N1
N17

TA'MINOTCHI ISMI

Карим
Нукус

2- misol. QISM va RANG munosabatlari (10.8-rasm) quyidagi ko'rinishda bo'lsin:

QISM

QISM RAQAMI	QISM NOMI	RANGI	OQTIRISHI	MANZIL
P1	вакт	«ў»	13	Тошкент
P1	вакт	қорал	13	Ферзик
P2	Болт	кўх	18	Тошкент
P3	табак	ҳаро ранг	18	Андижон
P4	табак	кўниҳ	15	Бухоро
P5	шабба	кўх	13	Тошкент
P6	шуруп	ҳаро ранг	20	Навој
P7	вакт	сарик	21	Тошкент

RANG

RANGI	MANZIL
кўк	Тошкент
кўзла	Ферзик

10.8-rasm.

U holda, QISM + RANG natijasida, quyidagi munosabat hosil bo'ldi:

QISM RAQAMI	QISM NOMI	OG'IRLIGI
P2	боят	18
P5	шайба	13

3- misol. MANZIL munosabati (10.9-rasm) quyidagicha bo'lsin:

### MANZIL

RANGI	OG'IRLIGI	MANZIL
хўк	18	Тошкент
жаво ранг	18	Андиқкан
жаво ранг	20	Наводӣ

10.9-rasm.

U holda, munosabatlarni bo'lish QISM + MANZIL natijasida, quyidagi munosabat hosil bo'ldi:

QISM_RAQAM:	QISM_NOMI

Bo'lish amali natijasida amalga oshiriladigan so'rovlar, odatda «hamma» so'zini o'z ichiga oladi. Masalan, Toshkentda ko'k rangdagi, Farg'onada qizil rangdagi qismlarni ishlab chiqaruvchi hamma qism raqamlari, nomlari va og'irligi aniqlansin.

### Tutashish amali

*Tutashish (join)* amali dekart ko'paytmasining hosila amali hisoblanadi, chunki u dekart ko'paytmasidagi tanlash amaliga ekvivalent bo'lib, ikkita operand-munosabatlarning dekart ko'paytmasidan, ya'ni tutashish amalining predikatidagi tanlash shartini qanoatlantiruvchi kortejlardan iborat.

Ushbu holdagi algoritm asosida birlashtirishni, ya'ni munosabatlardagi tutashishga asos atributlarning elementlari bir xildagi qiymatga ega bo'lgan kortejlarining kombinatsiyali tutashishini, (*join*) deb atashadi. Bu amalning

natijasi, ya'ni yangi munosabatga bir yo'la atributlarni nusxalash amalini ham qo'llash mumkin.

*Tutashish* amalini  $R_1 \rightarrow\!\!< R_2$  belgisi bilan ifodalaymiz.

*Tutashish* amalining turlari ko'p bo'lgani va ularni bir-birdan farqlash uchun, har biri turli nomlar bilan ataladi:

- tutashish (tabiiy tutashish);
- yarim tutashish;
- ekvitutashish;
- $\Theta$  - tutashish;
- tashqi tutashish.

$R_1(A,B,C)$

A	B	C
I	1	Y
J	3	Y
K	3	Z
L	2	Y
M	1	Z
N	7	Y

$R_2(B,D,E)$

B	D	E
1	p	S
2	q	T
5	p	U
7	q	T

$R_3(E)$

E
T

10.10-rasm.

10.10-rasmdagi  $R_1(A,B,C)$ ,  $R_2(B,D,E)$  va  $R_3(E)$  munosabatlarni turli usullarda *tabiiy tutashish* natijasida 10.11-rasmdagi T munosabat turli atributlar ishtirokida keltirilgan, ya'ni bir yo'la nusxalash amali ham bajarilgan.

Munosabatlarning *tutashish* natijalari:

a)  $T(A,B,C,D,E) = R_1 \rightarrow\!\!< R_2$

A	B	C	D	E
I	1	Y	p	S
L	2	Y	q	T
M	1	Z	p	S
N	7	Y	q	T

b)  $T(A,B,D) = R_1 \rightarrow\!\!< R_2$

A	B	D
I	1	p
L	2	q
M	1	p
N	7	q

c)  $T(A,D) = R_3 \rightarrow\!\!< R_2$

A	D
I	p
L	q
M	p
N	q

d)  $T(B,D,E) = R_2 \rightarrow\!\!< R_3$

B	D	E
2	q	-
7	q	-

e)  $T(D,A,C) = R_1 \rightarrow\!\!< R_3 \rightarrow\!\!< R_2$

D	A	C
q	1	Y
q	N	Y

f)  $T(D) = R_3 \rightarrow\!\!< R_2$

D
q

10.11-rasm.

a), b) va c) misollarda  $R_1(A,B,C)$  va  $R_2(B,D,E)$  munosabatlar birinchi munosabatdag'i, lekin ikkalasi uchun bir xil nomga ega bo'lgan, umumiyl V atribut asosida tutashtirildi va talab qilingan atributlarga hosil qilindi.

d) misolda  $R_2(B,D,E)$  va  $R_3(E)$  munosabatlar  $R_2$  munosabatidagi E atribut asosida tutashtirildi va talab qilingan atributlar hosil qilindi.

e) misolda ketma-ket bajarilishi lozim bo'lgan amallar bir yo'la keltirilgan. Avval  $R_1$  va  $R_2$  munosabatlar V atribut asosida tutashtiriladi, hosil bo'lgan, ya'ni a) misolning natijasidagi YE atribut bo'yicha  $R_3$  munosabat tutashtiriladi va talab qilingan atributlar bo'yicha natija hosil qilindi.

f) misolda  $R_2$  va  $R_3$  munosabatlar YE atribut bo'yicha birlashtirilgan, talab qilingan atribut bo'yicha ikkita bir xildagi kortejlar qatnashgani uchun faqat bittasi qoladi.

10.12-rasmda aniq jadvallar misolida tutashish amali va uning natijasiga misol keltirilgan.

A117			
HISOB RAQAMI	ISMU	MANZIL	POCHTA INDEKSI
B12-27/11	ЧОРМЕДЛ. РАЙМОДЛ.	10Б51	10517
B12-28/15	ДЖАССУНГ ИДОРОВА А.	24Б51	10518
B12-29/12	ДЖАССУНГ ЧОРИ Б.	20Б 2-3	10517
B12-30/29	ДЖАССУНГ ЧОРИ Б.	40Б 2-3	10517
B12-40/51	ЧОРИ Б.	45Т 2-3	10517

B129			
HISOB RAQAMI	JINOVAL SAHASI	JINOVAL TURI	YAMOQ MUDODATI
B12-42/13	11273	ЗПРДАИ	15
B12-42/15	2326	ЧЕРКАШ	9
B12-42/21	421125	ЧЕРКАШ	3
B12-42/21	11176	ОЛАМУРЗ	БРД 3МР
B12-46/72	3378	ЧУДАЧОВ	11

A118			
HISOB RAQAMI	ISMU	KASBI	YILIGI DAROMADI
B12-42/13	РУМНОВ Р.	БОХДАНГИР	10000
B12-27/12	СЕРОВ С.	САВИ АБДИ	30000
B12-75/19	МАСИХЕВ К.	БУРАДОР	10000
B12-40/51	ЧОРИ Б.	СУЛТОНОВ	10000
B12-39/24	ХАРДОЛ.	БОГРАММЕД	10000

T (ISMU, POCHTA-INDEKSI, KASBI, JINOVAL TURI, JINOVAL SAHASI) = B129 < A117 > < A118 >

T			
ISMU	POCHTA INDEKSI	KASBI	JINOVAL SAHASI
РАЙМОДЛ. ЧОРИ Б.	10517 10517	СУЛТОНОВ БУРАДОР	2326 11176

10.12-rasm. Uchta munosabatdan bitta munosabat hosil qilish.

### Yarim tutashish amali

*Yarim tutashish* amali algoritmining asosida, ikkala munosabatdag'i umumiyl atributning teng bo'lgan elementlarini birlashtirish yotadi. Yangi munosabatdag'i atributlar soni *tutashayotgan* birinchi munosabatdag'i atri-

butlar nomi asosida yaratiladi.

10.10-rasmdag'i  $R_1(A,B,C)$ ,  $R_2(B,D,E)$  va  $R_3(E)$  foydalanib, yarim tutashish amaliga uchta misol keltiramiz:

$$A) T = R_1 > R_2$$

A	B	C
I	1	Y
L	2	Y
K	?	Y

$$B) I = R_2 > R_3$$

B	D	E
2	q	T
?	q	T

$$C) T = R_3 > R_2$$

E
T

### Ekvivalentlik bo'yicha tutashish amali

*Ekvivalentlik bo'yicha tutashish* (qisqacha, *ekvitutashish*) amalida, ikkita munosabatni turli nomdag'i atributlarining elementlari taqqoslanadi. Munosabatlarning bir xildagi qiymatga ega bo'lgan kortejlari kombinatsiyali tutashadi va yangi munosabatni hosil qiladi. Oddiy tutashish amali dan farqi, taqqoslashga asos atributlarning ikkalasi ham qatnashadi.

10.10-rasmdag'i  $R_1$  va 10.13-rasmdag'i  $R_4$  munosabatlarini mos ravishda C va E atributlari bo'yicha tutashtiramiz. Boshqacha aytganda, F predikat (C va E) bo'yicha teng qiymatli atributlami kombinatsiyali tutashtiradi.

$$R_4(D, E)$$

D	E
p	Y
q	Z
p	U
s	Z

10.13-rasm.

Misollardagi = belgisi chap va o'ng ifodalarning aynan bir xilligini ifodalaydi.

$T(A, B, C, D, E) = R_1 \sqcap \neg R_2 \sqcap F$   $R_4 = R_1[C=E]$   $R_4$  ekvitetashish natijasida hosil bo'lgan munosabat quyidagicha bo'ladi:

A	B	C	D	E
I	1	Y	p	Y
J	3	Y	p	Y
K	3	Z	q	Z
K	3	Z	s	Z
L	2	Y	p	Y
M	1	Z	s	Z
M	1	Z	q	Z
N	7	Y	p	Y

10.14-rasm.

10.14-rasmdagi kursiv bilan ajratilgan kortejlarga e'tibor bering.

#### ⊖ - tutashish amali

⊖ – *tutashish* amali, munosabatlarni *tutashish* amalining kengayti-  
rilgan varianti bo'lib, ikkita munosabatning dekart ko'paytmasidagi F predi-  
katni qondiruvchi kortejlardan iborat bo'ladi. F predikatda turli taqoslash  
belgilari hamda OR va AND ishlatalishi mumkin.

10.15-rasmdagi MA'LUMOT va OILA munosabatlarini XIZMAT-  
CHI RAQAMI va XIZMATCHI KODI atributlari bo'yicha = belgisi va  
asosida ⊖-*tutashish* natijasidagi to'rtta misolni ko'ramiz. ⌈ ⌋ shakldagi  
F predikat atributlariga turli taqoslash belgilari va BuJ amallarini qo'llash  
mumkin.

#### MA'LUMOT

XIZMATCHI RAQAMI	OQ'YU	TUGATGAN YILI	DIPLOM RAQAMI	STAJ
53702	ТДПУ	1982	КД124568	29
53703	ТДПУ	1989	КД164989	22
53791	АндрМИ	1979	УА000258	32
53800	ФарМИ	1984	АП235680	27
53805	НМИ	1978	МС456893	33
53806	ТДПУ	1981	АА000235	30

OILA

XIZMATCHI KODI	FARZAND	NABIRA
53702	4	1
53703	4	0
53791	4	6
53800	3	0
53802	0	0
53804	2	0
53805	5	10
53806	3	1

10.15-rasm.

a) MA'LUMOT [XIZMATCHI\_RAQAMI = XIZMATCHI\_KODI] OILA amali orqali hosil bo'lgan munosabat quyidagicha bo'ladi:

XIZMATCHI RAQAMI	OO'YU	TUGATGAN YILI	DIPLOM. RAQAMI	STAJ	XIZMATCHI KODI	FARZAND	NABIRA
53702	ТДПУ	1982	КД124568	29	53702	4	1
53703	ТДПУ	1989	КД164989	22	53703	4	0
53791	АнМК	1979	УАО00218	32	53791	4	6
53800	ФарМи	1984	АЛ235680	27	53800	3	0
53805	НМИ	1978	МС456893	33	53805	5	10
53806	ТДПУ	1981	АА000235	30	53806	3	1

b) MA'LUMOT [XIZMATCHI\_RAQAMI > 53800 OR XIZMATCHI\_KODI > 53800] OILA formulasi orqali hosil bo'lgan yangi munosabat esa quyidagicha bo'ladi:

XIZMATCHI RAQAMI	OO'YU	TUGATGAN YILI	DIPLOM. RAQAMI	STAJ	XIZMATCHI KODI	FARZAND	NABIRA
53805	НМИ	1978	МС456893	33	53805	5	10
53806	ТДПУ	1981	АА000235	30	53806	3	1

c) MA'LUMOT [XIZMATCHI\_RAQAMI = XIZMATCHI\_KODI OR STAJ ≥ 32] OILA formulasi orqali hosil bo'lgan yangi munosabat quyidagicha bo'ladi:

XIZMATCHI RAQAMI	OO'YU	TUGATGAN YILI	DIPLOM. RAQAMI	STAJ	XIZMATCHI KODI	FARZAND	NABIRA
53791	АнМК	1979	УАО00218	32	53791	4	6
53806	ТДПУ	1981	АА000235	30	53806	3	1

d) MA'LUMOT [XIZMATCHI\_RAQAMI = XIZMATCHI\_KODI]

> 53800 AND STAJ ≥ 32] OILA formulasi orqali hosil bo'lgan yangi munosabat quyidagicha bo'ladi:

XIZMATCHI RAQAMI	OO'YU	TUGATGAN YILI	DIPLOM. RAQAMI	STAJ	XIZMATCHI KODI	FARZAND	NABIRA
53805	MMI	1978	MC458K93	33	53805	3	10

e)  $R_1 \bowtie_{F} R_4 \equiv R_1 [C > E]$   $R_4$  formulasi orqali hosil bo'lgan yangi munosabat quyidagicha bo'ladi:

A	B	C	D	E
I	1	Y	P	U
J	3	Y	P	U
K	3	/	P	Y
K'	3	Z	P	U
L	2	Y	P	U
M	1	Z	P	Y
M'	1	Z	P	U
N	7	Y	P	U

### Tashqi tutashish

Ikkita munosabatni *tashqi tutashish* (outer join) natijasida, turlicha algoritm asosidagi yangi munosabat hosil bo'ladi. Tashqi *tutashishni*, munosabatlarning turlicha *tutashishiga* nisbatan *tashqi chap tutashish*, *tashqi o'ng tutashish* va *tashqi to'liq tutashish* deb atashimiz ham mumkin. Shunga asosan, *tashqi tutashishni* 3 xilda amalga oshirish mumkin. Mos kelsmagan qiymatlar bo'yicha olingan kortejlar **null** bilan to'ldiriladi.

1- xil. a)  $R_1 \bowtie_F R_2 \equiv R_1 [B=B] R_2$  formula asosida birinchi munosabatdagi kortejlarning hammasi hamda taqqoslanayotgan umumiy atributlarda ikkinchi munosabatning mos qiymatlari bo'lmagan kortejlari ham yangi munosabatga tutashilgan holda kiritiladi. *Tashqi chap tutashish* natijasida quyidagi yangi munosabatni hosil qilamiz:

A	B	C	B	D	E
I	1	Y	1	p	S
L	3	Y	null	null	null
J	3	Z	null	null	null
K	2	Y	2	q	T
M	1	Z	1	p	S
N	7	Y	7	q	T

b)  $R_1 \supset \triangleleft_F R_2 \equiv R_1 [B > B] R_2$  formula asosida *tashqi chap tutashish* natijasida quyidagi yangi munosabatni hosil qilamiz:

A	B	C	D	E
I	1	Y	null	null
J	3	Y	1	p
J	3	Y	2	q
K	3	Z	1	p
K	3	Z	2	q
L	2	Y	1	p
M	1	Z	null	null
N	7	Y	1	p
N	7	Y	2	q
N	7	Y	5	p

2- xil.  $R_1 \triangleright \subset_F R_2 \equiv R_1 [B=B] R_2$  ikkinchi munosabatdagи kortejlar-ning hammasи hamda taqqoslanayotgan umumiy atributlarda birinchi munosabatdagи mos qiymatlari bo'yigan kortejlar ham yangi munosabatga tutashilgan holda kiritiladi. *Tashqi o'ng tutashish* natijasida quyidagi yangi munosabatni hosil qilamiz:

A	B	C	D	E
M	1	Z	1	p
I	1	Y	1	p
L	2	Y	2	q
null	null	null	5	p
N	7	Y	7	q

3- xil.  $R_1 \supset \triangleleft_F R_2$  ikkala munosabatdagи kortejlarining hammasи hamda taqqoslanayotgan umumiy atributlar bo'yicha ikkala munosabatdagи o'zaro mos qiymatlari bo'yigan kortejlar ham yangi munosabatga tutashilgan holda kiritiladi. *Tashqi to'liq tutashish* natijasida quyidagi yangi munosabatni hosil qilamiz:

A	B	C	D	E
I	1	Y	1	S
L	3	Y	null	null
J	3	Z	null	null
K	2	Y	2	T
M	1	Z	1	S
N	7	Y	?	T
null	null	null	5	U

Quyidagi 10.1 jadvalda yuqorida keltirilgan relatsion amallarning muqobil belgilash sintaksisi keltirilgan.

10.1-jadval

Amal nomi	Amalni relatsion belgilash	Muqobil belgilash sintaksisi
Birlashish	$R \cup S$	R UNION S
Kesishish	$R \cap S$	R INTERSECT S
Ayirish	$R - S$	R MINUS S
Dekart ko'paytmasi	$R \times S$	R TIMES S
Tanlash amali	$X \Theta Y$ , X va Y R munosabatning atributlari, $\Theta$ -taq-qoslash belgilaridan biri	R WHERE F yoki R WHERE X $\Theta$ Y F predikat
Nusxalash	$G = P(X_1, X_2, \dots, X_n)$ (R)	$R[X_1, X_2, \dots, X_n]$
Bo'lish	$R \div S$	R DEVIDE BY S
Tutashish amalining umumiy shakli	$R \sqsubset\sqsupset_{F_p} S$	(R TIMES S) WHERE F F: mantiqiy ifoda $\rightarrow$ R va S munosabatning atributlari, skalyar ifoda
Tabiiy tutashish	$R \sqsubset\sqsupset_{F_p} S$	R JOIN S; R JOIN S JOIN Z ham o'rnli
Yarim tutashish	$R \triangleright_{F_p} S$	$R[X_1, X_2, \dots, X_n] \text{ JOIN } S$
Ekvitetutashish	$R \sqsubset\sqsupset_{F_p} S$	(R TIMES S) WHERE F F: $R[X=Y] S, X \in R, Y \in S$

$\Theta$ -tutashish	$R \bowtie_F S$	(R TIMES S) WHERE F F: $X \Theta Y$ yoki $R [X \Theta Y]$ S
Tashqi tutashish	$R \supset_{\Theta} F S$ , $R \bowtie_{\Theta} S$ , $R \supset_{\Theta} F S$	R LEFT OUTER JOIN S $R [X \Theta Y] S$ R RIGHT OUTER JOIN S R $[X \Theta Y] S$ R FULL OUTER JOIN S $R [X \Theta Y] S$

### Xulosə

Relatsion algebraning oddiy algebradan farqi, algebrayik o'zgaruvchilar sifatida munosabatlar ishtiroy etadi, shuning uchun ular ustida bajarilgan algebrayik va boshqa amallarning natijasi yangi munosabat bo'ladi. Munosabatning o'zi esa to'plam deb hisoblanadi.

Berilganlarning relatsion modelida relatsion tillar ishlataladi. Relatsion tillarning asosida esa relatsion algebra va relatsion hisoblash yotadi. Relatsion algebrani norasmiy ravishda yuqori darajali protseduralari til sifatida tavsiflash mumkin. U orqali, BBB tizimiga qanday qilib, berilganlar bazasidagi bitta yoki bir nechta munosabatdan, bitta yangi munosabati qurish mungkinligi ko'rsatiladi.

Biz uchun relatsion algebraning eng zaruriy tomoni, berilganlarni o'zgartirish tillaridagi asosiy amallarning natijalarini ko'rish imkoniyatini beradi.

### Nazorat uchun savollar

1. Relatsion algebraning maqsadini tushuntiring.
2. Kesishish amalini ikkita munosabat misolida tushuntiring.
3. Birlashtirish amalini ikkita munosabat misolida tushuntiring.
4. Ayirish amalini ikkita munosabat misolida tushuntiring.
5. Tanlash amalini misollar bilan tushuntiring.
6. Oddiy tanlovga misol keltiring.
7. Predikatlar bo'yicha oddiy tanlovga misol keltiring.
8. Bul amallarini izohlang.
9. Bul amallari va predikatlardan foydalangan holda oddiy tanlovga misol keltiring.
10. Bul amallari va predikatlardan foydalangan holda murakkab tanlovga misol keltiring.

11. Aloqalar, bul amallari va predikatlardan foydalangan holda murakkab tanlovg'a misol keltiring.
12. Nusxalash amalining maqsadini izohlang.
13. Nusxalash amaliga misol keltiring.
14. Dekart ko'paytmasi amaliga misol keltiring.
15. Bo'lish amalining maqsadini izohlang.
16. Bo'lish amaliga misol keltiring.
17. Birlashtirish amalining ishlashini tushuntiring.
18. Birlashtirish amaliga misollar keltiring.
19. Yarim birlashtirish amalining bajarilishini tushuntiring.
20. Yarim birlashtirish amaliga misol keltiring.
21. Ekvivalentlik bo'yicha birlashtirish amalining vazifasini **izohlang**.
22. Ekvivalentlik bo'yicha birlashtirish amaliga misol keltiring.
23. Birlashtirish amalining vazifasini izohlang.
24. Birlashtirish amaliga misol keltiring.
25. Tashqi birlashtirish amalining vazifasini izohlang.
26. Tashqi birlashtirish amaliga misol keltiring.

---

## 11-bob. MUNOSABATLARNI NORMALLASH

**Tayanch iboralar:** «*pastdan yugoriga (ko'tariluvchi)*» yondashuv, «*yugoridan pastga (tushuvchi)*» yondashuv, normallashtirish, bиринчи normal shakl, oddiy atribut, murakkab atribut, funksional bog'liqlik, to'liq funksional bog'liqlik, ikkinchi normal shakl, uchinchi normal shakl, Boys-Kodd normal shakli, to'rtinchi normal shakl, ko'p ma'noli bog'liqlik, besinchi normal shakl.

Relatsion BBB tizimlari berilganlar bazasini loyihalashdagi berilganlarning mantiqiy modellarini yaratishdagi *asosiy maqsad* berilganlarni aniq tafsifini, ular orasidagi aloqalarni va zaruriy cheginnalarni yaratishdan iborat. Bu maqsadga erishish uchun bиринчи galda, berilganlar munosabatlarining kerakli majmuasini yaratish zarur. Ana shu masalani yechish uchun *normallashtirish usuli* qo'llaniladi. BB loyihalashda normallashtirish usuli «*pastdan yugoriga (ko'tariluvchi)*» yondashuviga, ya'ni atributlar orasidagi aloqalarni o'rnatishdan boshlanadi.

BB, ya'ni ularning chizmalarini loyihalash - berilganlarni takrorlashni minimallashtirish, soddalashtirish, ularga ishlov va yangilash jarayonlarini tezlashtirish masalalarini hal qilishi zarur. BB chizmalarini *noto'g'ri* loyihalash berilganlarni mukammallashtirishda va amaliyotda turli tasodiflarga olib kelishi mumkin. Shu kabi muammolarni yechish uchun munosabatlarni normallash kerak bo'ladi.

Lekin BB bilan ishslash texnologiyasida, berilganlar zaxiralarning juda katta hajmlariga nisbatan so'rovlarni bajarish tezligini oshirish maqsadida, teskari usul - *munosabatlarni aks normallash* qo'llanishi mumkin. Bu jarayon amaliy tarzda faoliyat yuritayotgan relatsion BBBT BB ustida olib boriladi. Bu usulni «*yugoridan pastga (tushuvchi)*» yondashuv deyishimiz mumkin. Loyihalash chizmalaridagi ba'zi aloqalar qaytadaq ko'riliishi natijasida, munosabatlar bo'linishi ham mumkin.

BB doimiy ravishda o'zgarib turadi. Unga yangi ma'lumot elementlari qo'shiladi. Ular orasida yangi aloqalar yoki bog'lanishlar o'rnatiladi va ularni qayta ishslashni yangi usullari qo'llaniladi. Foydaluvchi, yaratilgan BB boshqaruvchi tizimni deyarli o'zgartirmasligiga harakat qilinadi. Bu muammoni hal qilish uchun berilganlar elementlarini asosli ravishda guruhlarga birlashtirish va ular uchun kalitlarni aniqlash zarur bo'ladi. Hozirgi

kunda BB boshqaruvchi tizimlarni yaratishda berilganlarni normal shaklda tasvirlashni taklif etadir.

### Normallashtirish bosqichlari

Berilganlarning relatsion modeli doirasida birinchi marta E.F. Kodd [12] tomonidan munosabatlarni normallash tamoyillari ishlab chiqilgan. Dastavval birinchi, ikkinchi va uchinchi normal shakllar taklif qilingan. Keyinchalik R. Boys va E.F. Kodd [16] tomonidan V. Kent [44] takliflariga asoslangan holda, uchinchi normal shaklning qat'iyroq ta'rifi berilgan va bu Boys-Kodd normal shakli (BKNSh) nomi bilan yuritiladi. Bu normal shakllarning hammasi munosabat atributlari orasidagi funksional bog'lanishlarga asoslangan. Bularidan yuqoriyoq darajadagi normal shakllar R.Fagin [24-26] tomonidan keyinroq kiritilgan (1977-1981-yillar). Bular to'rtinchi va beshinchi normal shakllardan iborat.

**Normallashtirish** – *munosabatlarning birlamchi kalitlari va mavjud aloqalari asosida, ularni tahlil qilishning rasmiy usuli*. Uning vazifasi – BB bitta chizmasini (yoki munosabatlar majmuasini) boshqa chizma bilan almashtirishdan iborat bo'lib, unda munosabatlar soddaroq va doimiy tuzilishga ega bo'ladi.

Ayrim relatsion BB loyihalarini amaiga oshirish jarayonida sifatlari boshqarish tizimini yaratish uchun birinchi normal shaklning talablarini bajarish yetarli, deb hisoblanadi. 11.1-rasmida normallashtirish bosqichlari va ularning qisqacha mazmuni kehrilgan.

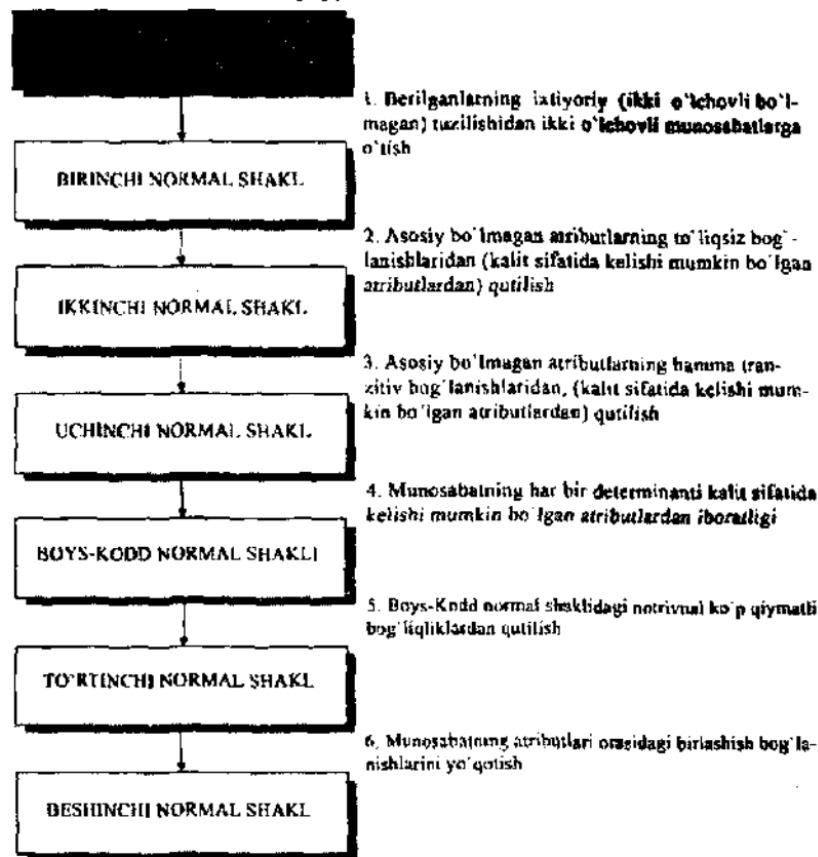
#### Birinchi normal shakl

**Normallashmagan shakl** – bu bir yoki bir nechta marta takrorlanuvchi berilganlar guruhibi saqlovchi jadvaldir. **Birinchi normal shakl (INSh)**da esa, munosabatdagi har bir satr va har bir ustunning kesishishida faqatgina bitta, yagona qiymat bo'lishi mumkin. **Birinchi normal shakl** oddiy va murakkab atributlar tushunchasi bilan bog'liq. **Oddiy atribut** – bu qiymatlari bo'linmaydigan xususiyatdir. **Murakkab atribut** esa, bitta yoki bir nechta domenlarning qiymatlarini birlashtirish natijasida hosil bo'lgan qiymattan iborat bo'ladi. Birinchi normal shaklda qaytariladigan *atributlar* yoki ularning guruhlari yo'qiladi, ya'ni oshkormas «yashiringan» xususiyatlar aniqlanadi.

**Birinchi normal shakl** – bu oddiy munosabatdir. Har qanday munosabat INShda bo'ladi. INShdag'i munosabatlarning xossalalarini yana bir bor eslatib o'tamiz:

- munosabatda bit xildagi kortejlar bo'lmaydi;
- kortejlar tartiblanmagan;

- atributlar tartiblanmagan va turli xilda nomlangan;
- barcha atributlarning qiymatlari bo'linmas.



11.1-rasm. Normallashtirish bosqichlari.

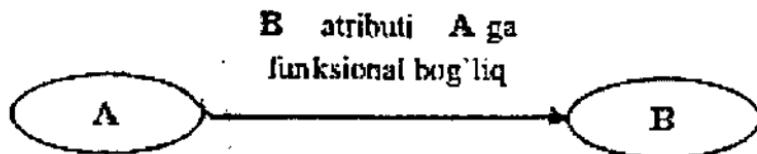
Berilganlarni mantiqiy modellash jarayonidagi birinchi bosqichda berilganlarni bitta munosabatda saqlash tavsiya qilinadi. Bu tavsiya ancha munozarali. Bu xususida biz 16- bobda batafsil to'xtab o'tamiz.

*Munosabatning hamma atributlari oddly bo'lsa, INShga keltirilgan deyiladi*, ya'ni atributning qiymatlari to'plam yoki takrorlanuvchi guruh bo'lmagan holat.

Jadvalarni INShga keltirish uchun murakkab atributlarni sodda atributlarga ajratish kerak.

## Funksional bog'lanish tushunchasi

Relatsion BBda berilganlarni tuzilishidan tashqari ularni chizmalariga ham e'tibor beriladi. BBni tuzilishi – munosabatning chizmasi yordamida beriladi. Berilganlar elementlarini qamrab oluvchi munosabatlar yaratilganda, obyektning qaysi atributlari bog'liqligini aniqlashi kerak. Shuning uchun amaliyotda munosabatlarning normal shakllari, relatsion BB nazar-iyasidagi tayanch tushuncha bo'lib, funksional bog'lanish tushunchasiga asoslanadi. Funksional bog'lanish, munosabatdagi atributlarning mazmuni bo'yicha bog'lanishni ko'rsatadi, ya'ni atributlar bir-bir bilan qanday bog'lanishi mumkinligini, hamda atributlar orasidagi funksional bu bog'lanishlar ba'zi bir atributlarga qo'yilgan turli cheklashlar orqali aniqlanadi. Masalan, R munosabatdagi A va V atributlarni ko'raylik. Agarda vaqtning har daqiqasida R munosabatdagi A atributning har bir elementiga shu munosabatdagi V atributning bittadan ortiq bo'limgan elementi mos kelsa, V atribut A atributga *funktional bog'langan* deyiladi (11.2-rasm). V atribut A atributga funksional bog'langan, degani, bu



11.2-rasm.

A atribut V atributni bir qiymatli aniqlaydi, ya'ni vaqtning qaysidir daqiqasida A atributning bitta elementining qiymati ma'lum bo'lsa, V atributning ham bitta elementining qiymatini aniqlash mumkin. Bularni, biz yuqorida A atributdan V atributga o'tuvechi bitta millik chiziqlar bilan ifodalagan edik. A atribut (atributlar to'plami)ni *determinant* deb ham atashadi. Yuqorida keltirilgan ta'rifni boshqacha shaklda ham ifodalash mumkin.

*Ta'rif.* R munosabat berilgan bo'lsin. R munosabatning ixtiyoriy vaqtidagi holatidan, uning har qanday  $r_1, r_2 \in R$  kortejlari uchun  $r_1.A = r_2.A$  ifodadan  $r_1.V = r_2.V$  kelib chiqsa, ya'ni A atributning hamma bir xil qiymatdagi kortejlari bilan V atributning ham bir xil qiymatdagi kortejlari uchun A atributning A atributlar to'plamiga *funktional bog'liq* yoki A Vni *funktional aniqlaydi* deyiladi. Lekin, B atributning berilgan qiymati uchun A atributning bir nechta qiymati mos kelishi mumkin (11.4.b rasm). Bu tahlillarni quyida isbotlaymiz.

Agarda A atributlar R munosabatning nomzod kalitlari bo'lsa, R munosabatning ixtiyeriy atributi Aga bog'liq bo'ladi.

XIZMATCHI (XIZMATCHI RAQAMI, XIZMATCHI ISMI, MAOSH, LOYIHA RAQAMI, TUGASH SANASI) munosabatini (11.3-rasmga qarang) ko'rib chiqaylik.

Quyida XIZMATCHI munosabatidagi funktional bog'lanishlar keltirilgan:

XIZMATCHI\_ISMI atributi XIZMATCHI\_RAQAMI atributiga bog'liq;

MAOSH atributi XIZMATCHI\_RAQAMI atributiga bog'liq;

LOYIHA RAQAMI atributi XIZMATCHI\_RAQAMI atributiga bog'liq.

TUGASH\_SANASI atributi XIZMATCHI\_RAQAMI yoki LOYIHA\_RAQAMI atributlariga bog'liq.

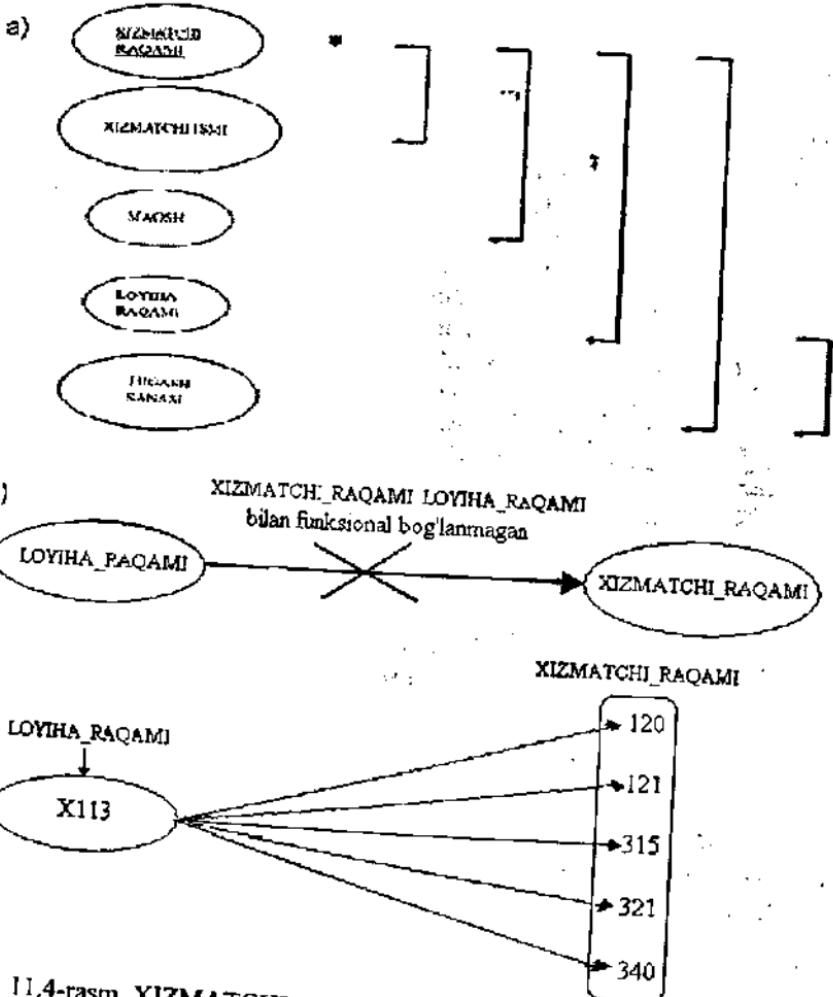
#### XIZMATCHI

XIZMATCHI_RAQAMI	XIZMATCHI_ISMI	MAOSH	LOYIHA_RAQAMI	TUGASH_SANASI
120	Махсур	2000	X113	17.07.2011
121	Балшир	1700	X113	17.07.2011
270	Тоёкор	1800	Y007	12.01.2014
373	Баҳор	3600	X113	17.07.2011
274	Зифир	3000	Z055	21.03.2012
279	Атак	2400	Y007	12.01.2014
301	Ҳайруннис	1800	Z055	21.03.2012
306	Абдулмажид	2100	Z055	21.03.2012
310	Хож нахбер	3000	Z055	21.03.2012
315	Хамит	3100	X113	17.07.2011
317	Бахтиер	2700	Y007	12.01.2014
320	Мўйисокон	1700	Y007	12.01.2014
321	Носир	2900	X113	17.07.2011
340	Рахматитон	3100	X113	17.07.2011
349	Мутторон	1900	Z055	21.03.2012
358	Тургун	2800	T043	15.09.2013
377	Тўқон	2800	T043	15.09.2013
390	Танзат	3333	Y007	12.01.2014

#### 11.3-rasm.

XIZMATCHI\_RAQAMI atributi MAOSH atributiga *funktional bog'lanagan*, chunki bir xil maoshlili kortejlarda XIZMATCHI\_RAQAMI bir xil emas, ya'ni bir nechta xizmatchi bir xildagi miqdorda maosh olishi mumkin. Xuddi shunday, XIZMATCHI\_RAQAMI atributi LOYIHA\_RAQAMI atributiga funksional bog'lanagan. Lekin LOYIHA\_RAQAMI atributi TUGASH\_SANASI atributiga funksional bog'langan. Boshqa hech

bir atribut LOVIHA\_RAQAMI atributi bilan funksional bog'lanmagan. Keltitilgan funksional bog'lanishlar 11.4.a) rasmida, 11.4 b) rasmida esa funksional bog'lanmagan atributlarga aniq misollar keltirilgan, \* belgisi bilan birlamchi yoki boshqa kalit bo'lishi mumkin bo'lgan atributlar belgilangan.



11.4-rasm. XIZMATCHI munosabatidagi funksional bog'lanishlar.

Agarda B Aga funksional bog'liq bo'lsa, A uchdan B uchga millik chiziq o'tkaziladi. Bu matematik mantiqdagi belgilashlarga mos, ya'ni

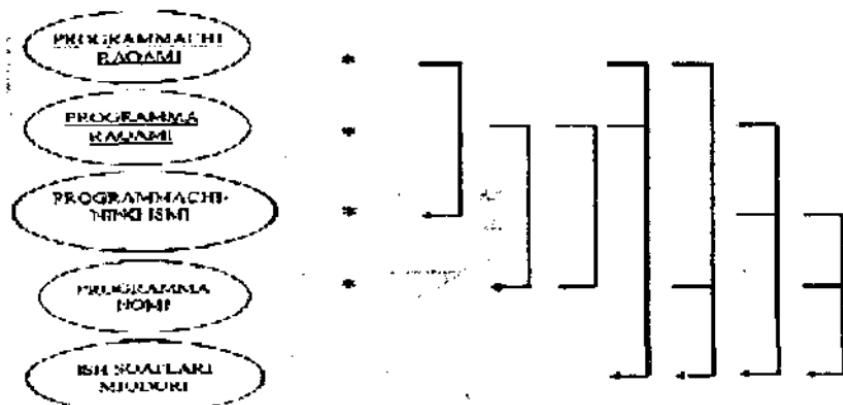
$A \rightarrow B$  belgilash, A Bni aniqlaydi degani, ya'ni A qandaydir a, qiymatni olsa, u holda B qaysidir lekin aniq  $b_k$  qiymatni oladi.

Shunday ham bo'ladiki, atribut bitta atributga emas, balki bir qancha atributlarga funksional bog'liq bo'lishi mumkin. Masalan, quyidagi programmachi faoliyatiga bog'liq munosabatni tahtil qilaylik:

PROGRAMMACHI\_FAOLIYATI (PROGRAMMACHI\_RAQAMI, PROGRAMMA\_RAQAMI, PROGRAMMACHINING\_ISMI, PROGRAMMA\_NOMI, ISH\_SOATLARI\_MIQDORI)

ISH\_SOATLARI\_MIQDORI atributi (PROGRAMMACHI\_RAQAMI, PROGRAMMA\_RAQAMI) tarkibli kalitiga yoki kalitga nomzodlardan biri bilan funksional bog'langan:

- (PROGRAMMACHI\_RAQAMI, PROGRAMMA\_NOMI)
- (PROGRAMMA\_RAQAMI, PROGRAMMA\_NOMI) yoki
- (PROGRAMMACHINING\_ISMI, PROGRAMMA\_NOMI)



11.5-rasm. PROGRAMMACHI\_FAOLIYATI munosabatidagi funksional bog'lanishlar.

Ushbu funksional bog'lanishlar 11.5-rasmida tasvirlangan. Bu erda **ham** \* belgisi bilan birlamchi (birlamchi yoki boshqa kalit bo'lishi mumkin bo'lgan) atributlar belgilangan.

### To'liq funksional bog'liqlik

Agarda R munosabatdagi V atribut shu munosabatdagi A (yoki atributlar termasi) atributning hamma elementlariga funksional bog'langan bo'lsayu, lekin A atributning birorta qismiga bog'liq bo'lmasa, V atribut A atributga to'liq funksional bog'langan deyiladi.

Masalan, PROGRAMMACHI\_FAOLIYATI munosabatida (11.5-rasmga qarang) ISH\_SOATLARI\_MIQDORI atributi (PROGRAMMACHI\_RAQAMI, PROGRAMMA\_RAQAMI) tarkibli kalitga to'liq funksional bog'langan, chunki aniq bitta programmani yaratishga ma'lum bir programmachingin sarflagan ISH\_SOATLARI\_MIQDORIni aniqlaydi. Lekin, shu tarkibli kalitni tashkil etuvchilarining hech biri alohida ISH\_SOATLARI\_MIQDORIni aniqlay olmaydi. PROGRAMMACHINING\_ISMI atributi bitta PROGRAMMACHI\_RAQAMI atributiga to'liq funksional bog'langan, xuddi shunday PROGRAMMA\_NOMI atributi PROGRAMMA\_RAQAMI atributiga to'liq funksional bog'liq.

### Ikkinchchi normal shakl

**Ikkinchchi normal shakl (2NSh)** tarkibli kalitlarga ega bo'lgan *munosabatlarga* (ikkita va undan ortiq atributlardan iborat) qo'llaniladi va funksional bog'liqlik tushunchasi bilan aloqador.

Ikkinchchi normal shakida, noyob kalitning qismiga bog'liq bo'lgan *atributlar* yo'qotiladi. Noyob kalitning ushbu qismi alohida *ma'nori* aniqlaydi.

*Munosabat 2NShda dey'iladí, agarda u INShga keltirilgan va har bir kalit bo'lmagan atribut tarkibli birlamchi kalitga to'liq funksional bog'liq bo'lgan holatda bo'tsa.*

Agarda hamma kalit bo'lishi mumkin bo'lgan (*kalitga nomzod deb atalgan*) atributlar bittadan atributga ega bo'lsa (masalan, 11.4-rasindagi kabi), u holda munosabat ikkinchi normal shakida bo'ladi, chunki birlamchi kalit bo'lmagan hamuna atributlar, kalitga nomzod atributlarga to'liq funksional bog'liq bo'ladi. Agarda inynosabatdagi kalitlar bittadan ortiq atributdan iborat bo'lsa va u birinchchi normal shakida berilgan bo'lsa, u ikkinchi normal shakida bo'imasligi ham mumkin.

Masalan 11.5-rasindagi munosabat ikkinchi normal shakida berilgan deyish mumkin, chunki ISH\_SOATLARI\_MIQDORI atributi yagona birlamchi atribut emas va kalitga nomzod har bir atributga to'liq funksional bog'liq.

Quyidagi 11.6-rasmda TA'MINOT munosabati TA'MINOT (MAX-SULOT\_RAQAMI, TA'MINOTCHI\_RAQAMI, TA'MINOTCHI\_ISMI, TA'MINOTCHI\_MA'LUMOTLARI, NARX) ikkinchi normal shakida berilmagan.

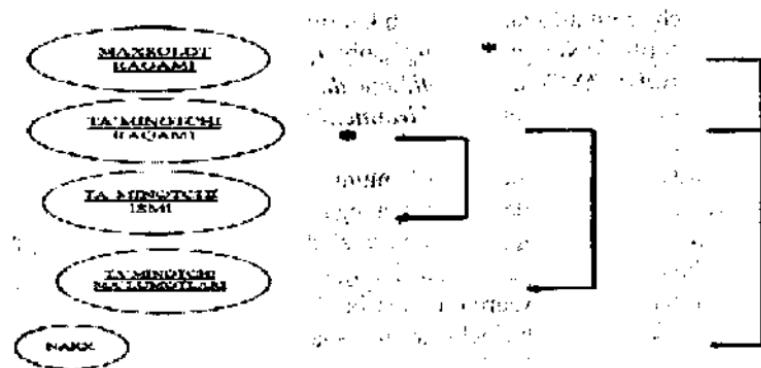
Bu munosabatda faqat bitta kalit bor. TA'MINOTCHI\_ISMI atributi kalitga kirmaydi, chunki bitta ta'minotchini turli joylarda turli nomer bilan belgilash mumkin, shuning uchun TA'MINOTCHI\_RAQAMI atributi TA'MINOTCHI\_ISMI atributining qiymati bilan aniqlanmaydi.

### TA'MINOT

MAXSULOT RAQAMI	TA'MINOTCHI RAQAMI	TA'MINOTCHI ISMI	TA'MINOTCHI MA'LUMOTLARI	NARX
1	1000	СОБИР	XZ	20
1	1900	ФАЙЗИЙ	ZX	28
1	2050	ВАЛИ	YY	22
1	1900	ХИКМАТ	ZW	30
2	3100	ЧОРИ	ZH	520
2	1000	СОБИР	XZ	90
2	2050	ВАЛИ	YY	590
3	2050	ВАЛИ	YY	1000
4	1000	СОБИР	XZ	80
4	3100	ЧОРИ	ZH	90
4	1900	ХИКМАТ	ZW	95
5	1500	ФАЙЗИЙ	ZX	160
5	1000	СОБИР	XZ	140

11.6-rasm.

11.7-rasmda TA'MINOTCHI\_ISMI va TA'MINOTCHI\_MA'LUMOTLARI atributlari birlamchi atributlar bo'lmasa ham, tarkibli kalitning qismi bo'lgan TA'MINOTCHI\_RAQAMI atributiga funksional bog'lanishlar.

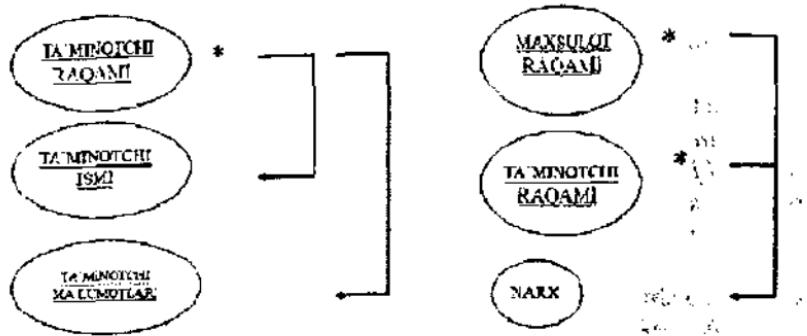


11.7-rasm. 2NShda bo'limagan munosabatdagi funksional bog'lanishlar.

Ikkinci normal shaklning shartlarini bajarilmasligi bir qator noqulayliklarga olib keladi:

- ta'minotchi qaysidir mahsulotni yetkazib bermasa, qo'shimcha ma'lumotlarni TA'MINOTCHI\_MA'LUMOTLARI atributiga qo'shib bo'lnaydi. Agarda u hech qanday mahsulot yetkazib bermasa kalit bo'lmaydi;
- agarda ta'minotchi mahsulot yetkazishni vaqtinchalik to'xtatgan bo'lsa, u holda kortejni yo'q qilish natijasida ta'minotching ushbu qiymatga tegishli hamma ma'lumotlar yo'q bo'ladi. Odatda, ulami saqlash kerak;

- qo'shimcha ma'lumotlarni kiritishda qiyinchiliklar tug'iladi, agarda ta'minotchi ko'p mahsulotlarni yetkazish majburiyatini olgan bo'lsa, qo'shimcha ma'lumotlarni kiritish berilganlar ortiqchaligiga olib keladi.
- Keltirilgan muammolardan qutilish yo'li, munosabatni ikkita munosabatga ajratishdan iborat. Shunda ular ikkinchi normal shaklga o'tadi (11.8 va 11.9-rasmlardagi kabi).



11.8-rasm.

Faqat NARX atributi to'liq tarkibli kalitga bog'liq, bo'lgani uchun ikkinchi munosabatni (11.8 va 11.9-rasmlar), qolgan hamma atributlar birinchi munosabatni tashkil etadi. Birinchi munosabatda faqat yagona kalit TA'MINOTCHI\_RAQAMI bo'ladi.

TA'MINOTCHI RAQAMI	TA'LINDORCHI ISHLI	TA'MINOTCHI MA'LUMOTLARI
100	0002	22
150	0030	22
200	0031	22
1900	0000	22
3.00	0001	22

MAXSULOT RAQAMI	TA'LINDORCHI RAQAMI	NARX
1	000	100
2	001	150
3	002	200
4	003	120
5	004	300
6	005	200
7	006	100
8	007	200
9	008	100
10	009	100

11.9-rasm. TA'MINOTCHI munosabatini ikkiga ajratish yo'li bilan 2NShni hosil qilish.

### Uchinchi normal shakl

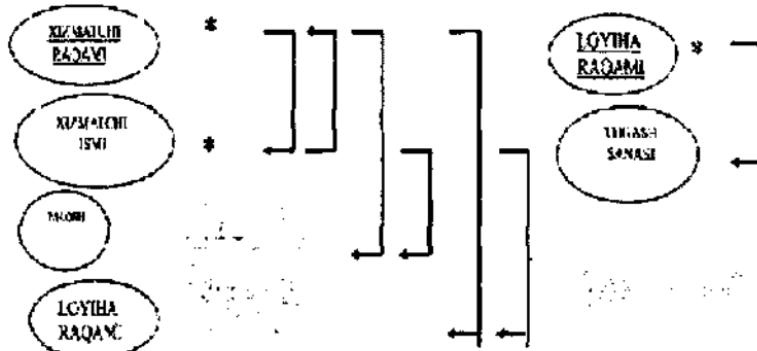
**Uchinchi normal shakl (3NSh)** tranzitiv bog'liqlik tushunchasi bilan aloqador. Faraz qilaylik, A, V, S – atributlar R munosabatga tegishli bo'lsin. Bunda A→V va V→S o'rinni bo'lib, aksinchasi o'rinsiz bo'lsin, ya'ni S Vga yoki V Aga bog'liq bo'lmasin. U holda, S Aga tranzitiv bog'liq deyiladi va A→→S tarzda ko'rsatiladi.

Uchinchi normal shaklda, noyob kalit tarkibida bo'lmanan *atributlarga* bog'liq bo'lgan *atributlar* yo'qotiladi, ya'ni munosabat ikkita munosabat-larga ajratiladi (A va V, V va S).

*Munosabat 3NShida deyiladi, agarida u 2NShga keltirilgan va birlamchi kalit bo'lmanan har bir atribut, birlamchi kalitga nisbatan tranzitiv aloqaga ega bo'lmanan holatda bo'lsa.*

Masalan, 11.3-rasmidagi XIZMATCHI munosabatida TUGASH\_SANASI atributi LOYIHA\_RAQAMI atributiga, oxirgisi esa o'z navbatida XIZMATCHI\_RAQAMI atributiga bog'langan. Shunday qilib, TUGASH\_SANASI atributi XIZMATCHI\_RAQAMI atributiga tranzitiv bog'langan. XIZMATCHI munosabatini uchinchi normal shaklga olib kelish uchun, uni ikkita munosabat-larga ajratish kerak. Ushbu holatlар 11.10 va 11.11-rasmarda keltirilgan.

Yangi, uchinchi normal shakldagi XIZMATCHI (XIZMATCHI\_RAQAMI, XIZMATCHI\_ISMI, MAOSH, LOYIHA\_RAQAMI) va LOYIHA (LOYIHA\_RAQAMI, TUGASH\_SANASI) munosabatlari hosil bo'ladi. Ushbu munosabatlarning afzalligi shundaki, loyihaning tugashi xususidagi sanalar xizmatchiga bog'liqsiz ravishda ham so'ralishi yoki aksinchasi bo'lishi mumkin. Loyihaning tugash sanasi haqiqatan ham loyihaning raqami bilan bog'liq.



11.10-rasm. XIZMATCHI munosabatini 3NShga keltirish.

XIZMATCHI XIZMATCHI- RAQAMI	XIZMATCHI- ISM	MAOSH	LOYIHA- RAQAMI
220	Мансур	2800	X113
221	Бекимбер	1700	X113
222	Тобко	1200	Y007
223	Балори	3600	X113
224	Зефар	3000	Z055
225	Ахмад	2400	Y007
226	Хайрулло	1800	Z055
227	Абдулжони	2100	Z055
228	Ходжембер	3000	Z055
229	Хамид	3100	X113
230	Бакетер	2100	Y007
231	Муроджон	1700	Y007
232	Насир	2900	X113
233	Рахматзода	3100	X113
234	Мухторали	1900	Z055
235	Тургун	2500	T043
236	Турсунов	2800	T043
237	Томат	3333	Y007

LOYIHA RAQAMI	YUGASRL SANASI
X113	17.07.2011
X113	21.03.2012
Y007	15.09.2013
X113	12.01.2014

Rasm 11.11.

### Boys-Kodd normal shakli

Hattoki, uchinchi normal shaklda bo'lgan munosabatlarda ham nonormallik bo'lishi mumkin. Bu tranzitiv bog'lanishlar bilan aloqador. Hamma tranzitiv bog'liqliklardan qutilish mumkinmi, degan tabiiy savol tug'iladi. Bu savolga ijobjiy javob beriladi. TRENER munosabatini (11.12-rasmga qarang) ko'raylik. Bu munosabatga quyidagi talablar qo'yilgan bo'lsin:

- sportchi bitta yoki bir nechta sport turiga qatnashishi mumkin;
- sportning bir turi bo'yicha bir nechta murabbiylar bor;
- har bir murabbiy faqat bitta sport turi bo'yicha murabbiylik qilishi mumkin.

Trenerlarning familiyasi, ismi va otasining ismi bir xilda emas, deb hisoblaymiz. Sportchi bir nechta sport turi bilan shug'ullanishi mumkin bo'lgani uchun SPORTCHI\_RAQAMI atributi SPORT\_TURI atributini aniqlamaydi. Bundan tashqari, sportchi bir nechta trenerda shug'ullanishi mumkin bo'lgani uchun, SPORTCHI\_RAQAMI atributini MURABBIY atributini ham aniqlamaydi. Shu asosda SPORTCHI\_RAQAMI atributni o'z-o'zidan kalit bo'la olmaydi.

(SPORTCHI\_RAQAMI, SPORT\_TURI) juftligi MURABBIY atributini, (SPORTCHI\_RAQAMI, MURABBIY) birligi esa SPORT\_TURI atributini aniqlaydi. Demak, shu juftliklardan ixtiyorisi kalit bo'lishi mumkin. Ikkita yoki undan ortiq kalit bo'lishi mumkin bo'lgan atributlarni *kalitga nomzodlar* (candidate keys), deb atagan edik. Kalitga nomzodlar safidan tanlangan kalitni esa *birlamchi kalit* (primary key) deb atagan edik.

Kalitga nomzoddardan tashqari yana bitta funksional bog'lanish mavjud. MURABBIY atributti SPORT\_TURI atributini aniqlaydi. Chunki, ixtiyorli murabbiy faqat bitta sport turidan trenerlik qiladi, uning familiyasi,

ismi sharifini bilgan holda, uning mutaxassisligini aniqlashimiz mumkin. Ana shuning uchun MURABBIY atributi determinant bo'ldi.

### TRENER

SPORTCHI RAQAMI	SPORT TURI	MURABBIY
50	күнгібозжык	Салынов М.Т.
100	волейбол	Бараев А.Ю.
130	футбол	Сафаев Д.Ж.
160	сүзаш	Алтев Б.Э.
100	сүзаш	Миниев М.С.
200	волейбол	Бараев А.Ю.
130	сүзаш	Алтев Б.Э.

11.12-rasm.

Ta'rif bo'yicha TRENER munosabati birinchi normal shaklda. Kalitsiz atributlari bo'limgani uchun, kamida bitta atributi minimum bitta kalitning tarkibiy qismida ishtirot etadi, TRENER munosabati bir yo'la ikkinchi normal shaklda ham. Tranzitiv bog'lanishlari bo'limgani uchun, bu munosabat uchinchchi normal shaklda ham. Lekin shunday bo'lishiga qaramasdan, bu munosabatda nonnormallik mavjud.

Sababi quyidagicha. Faraz qilaylik, 200 raqamli sportchi sport to'garagini tark etmoqchi. Agarda 200 raqamli sportchi to'g'risidagi kortejni olib tashlasak, A.IO. Barayevning voleybol bo'yicha murabbiyiligi to'g'risidagi ma'lumotni yo'qotamiz. Bu munosabatdan olib tashlashdagi nonnormallik.

Endi biz munosabatga I.B. Sharipovning gandbol bo'yicha murabbiyilagini kiritmoqchimiz, deytik. Buni, gandbol bo'yicha I.B. Sharipovda shug'ullanmoqchi bo'lgan kamida bitta sportchi paydo bo'limguncha, amalga oshirib bo'lmaydi! Bu, munosabatga qo'shishdagi nonnormallik.

Boys-Koddning normal shakli munosabatning hamma nomzod kalitlaridagi funksional bog'lanishlarga asoslangan. Yuqorida keltingan vaziyatlarni hal qilishda Boys-Kodd normal shaklidan foydalilanildi: *agarda munosabatning har bir determinanti kalitga nomzod bo'lsa, shu munosabat Boys-Kodd normal shaklda deyladi.*

TRENER munosabati Boys-Kodd normal shaklida emas. Uqorida keltilgan tahillilar asosida TRENER munosabatini ikkita munosabatga ajratamiz. Masalan, SPORTCHI (SPORTCHI\_RAQAMI, MURABBIY) va MURABBIY\_SPORT (MURABBIY, SPORT\_TURI) munosabatlariga ajratsak (11.13-rasm), bularda nonnormallik bo'lmaydi va ular BKNShda bo'ldi.

BKNSh funksional bog'lanishlarga aloqador nonnormalliklarga ega emas, munosabatlarni o'zgartirishda endi nonnormallik holatlari bundan

buyon yuz bermaydi deb hisoblanganda, nonormallik holati funksional bog'lanishlarga aloqasiz holatda ham bo'lishi mumkinligi aniqlandi.

SPORTCHI	MURABBIY
SPORTCHI RAQAMI	
50	Сарифов М.Т.
100	Ершев А.Ю.
130	Сафаров Д.Х.
160	Азимов Б.З.
100	Мусатов М.С.
200	Ершев А.Ю.
130	Азимов Б.З.

#### MURABBIY SPORT

MURABBIY	SPORT TURI
Сарифов М.Т.	яккучбозлик
Берасов А.Ю.	волейбол
Сафаров Д.Х.	футбол
Азимов Б.З.	сурх
Мусатов М.С.	сурх

11.13-rasm.

#### To'rtinchchi normal shakl

Nazariy tadqiqotlar natijasida, bog'lanishlarning yana bir turi – *ko'p ma'noli bog'liqliklarning* aniqlangan. Bu holat munosabatlarni loyihalash va amalga oshirishda, berilganlarning ortiqchaligiga bog'liq muammolarni keltirib chiqarishi mumkin.

Munosabatlardagi *ko'p ma'noli bog'liqliklarning* mavjudligi boshlang'ich jadvalarni 1NSHga keltirishdan kelib chiqishiga aloqador bo'lib, ba'zi bir qiymatlar majmuasi elementlarining bitta satr va bitta ustunning kesishgan joyida ketishi mumkin emasligi bilan bog'liq. Masalan, munosabatda ikkita *ko'p ma'noli* atributlar bor bo'lsa, satrlarning bir-biriga qarshi bo'limgan holatlarining yuzaga keltirish uchun, birorta atributdagi har bir qiymatni boshqa atributdagi har bir qiyamat bilan «birikma»larini takrorlash kerak. Ana shunday xildagi cheklashlar *ko'p ma'noli bog'liqliklarni* keltirib chiqaradi va berilganlarning ortiqchaligiga olib keladi.

11.14-rasindagi TALABA\_SPORTCHI munosabatini tahlil qilaylik. Ushbu munosabatda talaba sportchilar, sport yo'nalishi va sport turlari orasidagi aloqalari akslangan. Faraz qilaylik, talaba sportchilar bir nechta yo'nalishga ega bo'lishi va yana qo'shimcha ravishda qiziqqan sport turlari bilan ham shug'ullanishi mumkin bo'lsin. Bu holda yagona kalit (TALABA RAQAMI, YO'NALISH, SPORT TURI) tarqibiy qismidan iborat bo'ladi. Masalan, 50 raqamli talaba ikkita qilichbozlik va voleybol yo'nalishlarida tahsil oladi hamda qo'shimcha ravishda suzish va futbol bilan, 100 raqamli talaba esa futbol va suzish yo'nalishida hamda qo'shimcha gandbol bilan 130 talaba esa gandbol yo'nalishida tahsil oladi va voleybol bilan shug'ullanadi.

## TALABA\_SPORTCHI

TALABA RAQAMI	YO'NALISH	SPORT TURI
50	қиличбозлик	сузиш
50	волейбол	сузиш
50	қиличбозлик	футбол
50	волейбол	футбол
130	гандбол	волейбол

11.14-rasm.

TALABA RAQAMI va YO'NALISH atributlari orasida funksional aloqa yo'q, chunki talaba bir nechta yo'nalish bo'yicha shug'ullanishi mumkin. Shuning uchun TALABA RAQAMI atributining bir xildagi qiymatiga bir nechta yo'nalish mos kelishi mumkin. Bundan tashqari shu qiymatga SPORT TURI atributining bir nechta qiymatlari mos kelishi ham mumkin (11.14-rasmga qarang).

Atributlarning ana shunday bog'liqligi *ko'p ma'noli bog'liqlik* (multi-value dependency) deyiladi. Ko'p ma'noli bog'liqliklar berilganlarni o'z-gartirish vaqtida nonormallik holatlariiga olib keladi. Birinchi navbatda TALABA SPORTCHI munosabatidagi ortiqchalikka e'tibor qaratamiz. Raqami 50 bo'lgan talabaga to'rtta yozuv ajratilgan, ularning har birida yo'nalish va sport turlaridan bittadan. Agarda berilganlar ikkita kortejda saqlanganda, masalan, qilichbozlik va suzish bittada, voleybol va futbol ikkinchisida bo'lganda, bu holat foydalanuvchiga noto'g'ri ma'lumot berar edi. Foydalanuvchi, raqami 50 bo'lgan talaba qilichbozlik yo'nalishida bo'l-sagina suzish bilan, voleybol yo'nalishida o'qisagina futbol bilan shug'ullanishi mumkin, degan xulosalarga kelar edi. Lekin bu albatta noto'g'ri xulosalar. Chunki, yo'nalish va sport turlari orasida umuman hech qanday bog'liqlik yo'q. Ana shunday noto'g'ri xulosalarni chiqarmaslik uchun, qiymat «birikma»larini saqlash zarur.

Faraz qilaylik, raqami 50 bo'lgan talaba boks bilan shug'ullanmoqchi bo'lsin va shuning uchun munosabatga yangi kortej qo'shamiz (11.15-rasmga qarang). Endi, 50 raqamli talaba boks bilan faqt qilichbozlik yo'nalishida bo'lgani uchun shug'ullanadi, degan xulosa kelib chiqadi. Berilganlar to'g'ri tavsifga ega bo'lishi uchun, munosabatda raqami 50 talaba nechta yo'nalishda ta'lim olsa, shuncha kortej qo'shish kerak, ya'ni har bir yo'nalish bilan boks sport turini bog'liqligini ko'rsatish kerak. Demak, foydalanuvchi to'g'ri xulosaga kelishi uchun, yana bitta kortejni [50, voleybol, boks] munosabatga kiritish kerak (11.16-rasmga qarang).

**TALABA\_SPORTCHI**

TALABA_RAQAMI	YO'NALISH	SPORT_TURI
50	киличбозлик	бокс
50	волейбол	сузиш
50	киличбозлик	сузиш
50	волейбол	футбол
50	гандбол	футбол
130	волейбол	волейбол

11.15-rasm.

Bu munosabatni o'zgartirishdagi nonormallik, ya'ni bitta kortejni qo'shish uchun ancha ortiqcha yozuvlarni kiritish zarur bo'limoqda.

**TALABA\_SPORTCHI**

TALABA_RAQAMI	YO'NALISH	SPORT_TURI
50	киличбозлик	бокс
50	волейбол	бокс
50	киличбозлик	сузиш
50	волейбол	сузиш
50	киличбозлик	футбол
50	волейбол	футбол
130	гандбол	волейбол

11.16-rasm.

Umuman olganda, ko'p ma'noli bog'liqlik, munosabatning atributlarining soni kamida uchta bo'lganda, shundan ikkitasi ko'p ma'noli bo'lib, ularning qiymatlari faqatgina uchinchi atributga bog'liq bo'lganda yuzaga keladi. Boshqacha aytganda, R (A,B,C) munosabatda ko'p ma'noli bog'liqlik, agarda A atribut V va S atributlarni ko'p ma'noli aniqlasada, lekin V va S atributlar bir-biriga bog'liq bo'lmasdan holatda yuzaga keladi. Keltirgan misolimizda ham, TALABA\_RAQAMI atributi YO'NALISH va SPORT\_TURI atributlarini ko'p ma'noli aniqlaydi, lekin YO'NALISH va SPORT\_TURI atributlari o'zarbo'liq emas.

Ko'p ma'noli bog'liqlik >> kabi belgilanadi. TALABA\_RAQAMI >> YO'NALISH va TALABA\_RAQAMI >> SPORT\_TURI shakldagi yozuvning ma'nosi - «TALABA\_RAQAMI atributi YO'NALISH atributini, TALABA\_RAQAMI atributi SPORT\_TURI atributini ko'p ma'noli aniqlaydi». Ushbu munosabat BKNShda. Chunki u, 2NShda -- uning kaliti hamma atributlar majmuasidan iborat, 3NShda -- tranzitiv bog'liqliklar yo'tq, va

BKNShda – kalit bo‘limgan determinantlar mavjud emas. Shunga qaramasdan, u nonormallikka ega ekanligini aniqladik, agar talaba qo’shimcha yo’nalish olsa, u qancha sport turi bilan shug’ullansa, shuncha yo’nalishlarda ushbu talabaning sport turlarini ko’rsatish kerak. Agarda talaba yangi sport turiga qo’shimcha ravishda yozilsa ham xuddi shunday holat yuz beradi. Talaba yo’nalishlarning biridan voz kechsa, hamma shu yo’nalishga tegishli kortejlarni o’chirish kerak bo’ladi.

Ana shu nonormallikdan qutilish uchun, ko’p ma’noli bog’lanishlardan qutilish zarur. Buning uchun munosabatni ikkiga ajratish kerak. Ularning har birida faqatgina bittadan ko’p ma’noli bog’lanishlar saqlanishi mumkin. Natijaviy munosabatlarda nonormalliklar bo’lmaydi. Bu munosabatlar 11.17-rasmdagi TALABA\_YO’NALISH (TALABA RAQAMI, YO’NALISH) va TALABA\_SPORT\_TURI (TALABA RAQAMI, SPORT TURI).

Yuqoridagi tahlillardan kelib chiqqan holda, to’rtinchi normal shaklga ta’rif beramiz: *munosabat BKNSh bo’lsa va ko’p ma’noli trivial bog’liqliklarga ega bo’lmasa, u 4NShda bo’ladi.*

TALABA YO’NALISH

TALABA RAQAMI	YO’NALISH
50	юлашибози
50	волейбол
133	теннис

TALABA SPORT TURI

TALABA RAQAMI	SPORT TURI
50	бекс
50	тунд
50	тутон
133	волейбол

11.17-rasm.

### Beshinchchi normal shakl

Munosabati ixtiyorli usulda ikkita munosabatga ajratilganda, ular o’zlarining xossalari yo’qotmaydi va hosil bo’lgan munosabatlar yo’qotishsiz birlashtirish xossasiga ega bo’ladi. Bu esa, ularni yagona munosabatga o’z xossalari bilan qayta tiklash mumkin, degan ma’noga olib keladi. Lekin ba’zida munosabatni ikkitadan ortiq munosabatlarga ajratish zarur bo’ladi. Ana shunday juda kam uchraydigan vaziyatda yo’qotishsiz birlashtirishdagi bog’liqliki hisobga olish zaruriyatini tug’iladi. Bu vaziyat beshinchchi normal shakl yordamida amalga oshadi.

*Beshinchchi normal shakl*, ba’zida uni *toyihani birlashtiruvchi normal shakl* (Project-Join Normal Form) deb ham yuritishadi, ushbu shakldagi munosabatlarning atributlari birlashtirish natijasidagi o’zaro bog’liqlik-

*targa ega bo'lmaydi, ya'ni munosabatlarning atributlari orasidagi bir-tashish bog'ligtlarini yo'qotilgan bo'ladi.*

11.18-rasmda keltirilgan munosabatni tahlil qilamiz. Bu munosabat ko'chmas mulk obyektlarini tavsiflaydi. Zaruriy jihozlarni obyektga ta'minotchi yetkazib beradi.

a) KO'CHMAS\_MULK munosabatining mumkin bo'lgan holati

OB'EKTNOMI	JIHOZ_NOMI	TA'MINOTCHI_RAQAMI
A1	жавон	PY31
A1	музлатич	PY33
A5	жавон	PY33

b) KO'CHMAS\_MULK munosabatining mumkin bo'lgan holati

OB'EKTNOMI	JIHOZ_NOMI	TA'MINOTCHI_RAQAMI
A1	жавон	PY31
A1	музлатич	PY33
A5	жавон	PY33
A1	жавон	PY33

11.18-rasm.

Bundan tashqari, qaysidir ko'chmas mulk obyekti A<sub>n</sub> ga qandaydir k jihoz zarur bo'lsa va uni PY ta'minotchi A<sub>n</sub> obyektga yetkazgan bo'lsa, xuddu shu ta'minotchiga shu A<sub>n</sub> obyekt uchun yana boshqa jihozni yetkazib berishga buyurtma beriladi. Faraz qilamizki, ushbu misolda jihozning tavsifi jihozni bir ma'noda identifikatsiyalaydi.

11.18-rasmagi a) KO'CHMAS\_MULK munosabatiga qanday chegaralar ta'sirida ekanligini aniqlash uchun quyidagi tasdiqni ko'ramiz.

Agarda A1 obyektinga «javon» kerak (1- kortejdagi berilganlarga nisbatan), A1 obyektni jihoz bilan ta'minlovchi PY33 (2- kortejdagi berilganlar ga nisbatan), PY33 ta'minotchi «javon» bilan ta'minlovchi (3- kortejdagi berilganlarga nisbatan) bo'lsa, u holda PY33 ta'minotchi «javon» bilan A1 obyektni ta'minlashi kerak.

Ushbu tasdiq, cheklashda takrotlanish xossasi borligini va u KO'CHMAS\_MULK munosabatiga ta'sir etganiga yaqqol misol bo'ladi. Agarda shu cheklashga amal qilinsa, u holda (A1, javon, PY33) korteji KO'CH-

MAS\_MULK munosabatining hamma mumkin bo'lgan holatlarida qatnashishi zarur (11.18-rasmdagi b holat). Bu munosabatni yangilashdagi nonormallik holatlaridan biri bo'lib, KO'CHMAS\_MULK munosabati *birlashtirish bog'liqligini* saqlagan deyiladi.

*Birlashtirish bog'liqligi* bog'liqliklarning bir turi. Masalan, R munosabat A, B, C, ..., Z atributlardan iborat bo'lsin, u holda faqat va faqat R munosabatning olishi mumkin bo'lgan har bir qiymati uni nusxalarining A, B, C, ..., Z atributlardagi birlashish soniga teng bo'lsagina, R munosabat *birlashtirish bog'liqligi* shartini bajargan deyiladi.

Shunday qilib, R munosabatda *birlashtirish bog'liqligi* mavjud va shuning uchun u beshinchgi normal shaklda emas. Ushbu *birlashtirish bog'liqligidan* qutilish uchun KO'CHMAS\_MULK munosabatini dekompozitsiya qilamiz, ya'ni uchta 5NShdagi munosabatlarga OB'YEKT\_JIHOZ, JIHOZ\_TA'MINOTCHI va OB'YEKT\_TA'MINOTCHI ajratamiz (11.19-rasm).

OB'YEKT_JIHOZ	
OB'YEKT	JIHOZ NOMI
A1	жадор
A1	жудзатар
A5	жадор

JIHOZ TA'MINOTCHI	
JIHOZ NOMI	TA'MINOTCHI RAQAM
жадор	РУ31
жудзатар	РУ33
жадор	РУ33

OB'YEKT TA'MINOTCHI	
OB'YEKT NOMI	TA'MINOTCHI RAQAM
A1	РУ31
A1	РУ33
A5	РУ33

### 11.19-rasm.

Demak, *normallashtirish* – bu tashkilotdagisi berilganlarga qo'yilgan tabablar asosida munosabatlar majmularini xossalalarini shakllantirish usullaridan iborat.

BB tuzilishini modellashtirishda *normallashtirish* algoritmining bir qancha kamchiliklari bor:

- loyihalashtirilayotgan predmet sohaning hamma atributlarini bitta munosabatda bo'lishi taxmin qilinadi, bu esa tabiiy bo'lmagan holat. Chunki loyihani yaratishda, tajribali yaratuvchi aniqlangan *ma'nolarga* mos ravishda bir yo'la bir nechta *munosabat* tuzish zarurligini anglaydi. Agarda, bitta munosabatga hamma atributlar kiritilgan holatda ham, hosil bo'lgan munosabatning mag'zini chaqish murakkab kechadi;
- atributlarning to'liq ro'yxatini bir yo'la tuzish qiyin. Foydalananuvchilar bitta narsani turli nomlar bilan ataydilar va aksincha, turli narsalarni bir nomda ataydilar;
- normallash jarayonini o'tkazish uchun atributlarning bir-biriga bog'liqligini aniqlash va ajratish kerak. Bu masalani yechish ham katta

tajribani talab qiladi.

Normallashtirish rasmiy usul bo'lib, kalitlar va atributlar orasidagi mavjud funksional bog'liqliklar asosida munosabatlар tarkibini aniqlash uchun ishlataladi.

Munosabatlardagi berilganlarning ortiqchaligi berilganlarni yangilash paytida berilganlarni kiritish, yo'q qilish va o'zgartirish munosabatlarni nonnormalliklarga olib keladi.

Normallashtirishdagi eng asosiy ikkita konsepsiyaning birinchisi munosabatdagi atributlar orasidagi funksional bog'liqlikka aloqador bo'lsa, ikkinchisi atributlardagi berilganlarning ma'nolari bilan bog'liq.

Lekin shuni ta'kidlash joizki, BB bilan ishlashda tajriba ortgan sari, BB loyihalash jarayoni qanchalik murakkab va normallashtirish bu jarayonda muhim ahamiyat kasb etishini tushunish mumkin.

### Xulosha

*Normallashtirish – bu berilganlarga qo'yilgan talablarga nisbatan munosabatlар termasini aniqlangan xususiyatlar asosida yaratish usulidir. Normallashtirish rasmiy usul bo'lib, uni kalitlar va atributlar orasida mavjud bo'lgan funksional bog'lanishlar asosida munosabatlар tarkibini aniqlash uchun ishlatalish mumkin.*

Ortiqchalikka ega bo'lgan munosabatlар berilganlarni yangilashdagi nonnormallikka olib kelishi mumkin. Yangilash nonnormalligi deganda, berilganlarni kiritish, olib tashlash va o'zgartirish umallari natijasidagi nosozliklar tushuniladi.

*Normallashtirishdagi asosiy konsepsiyalardan biri funksional bog'liqlik bo'lib, u munosabatdagi atributlar orasidagi aloqalarni tavsiflaydi.*

### Nazorat uchun savollar

1. Relatsion berilganlar bazasini loyihalashdagi mantiqiy modellarni yaratishdan maqsad nima?
2. Munosabatlarni normallashtirishdan maqsad nima?
3. Munosabatlarni normallashtirishdagi «ko'tariluvchi» yondashuvni ta'riflang.
4. Munosabatlarni normallashtirishdagi «tushuvchi» yondashuvni ta'riflang.
5. Munosabatlarni normallashtirish tamoyillarini birinchi marta kim kiritgan?
6. INSh dan avval berilganlar qaysi holatda bo'ladi?

7. 1NSh ni izohlang va unga misol keltiring.
8. Munosabatlar orasidagi funksional bog'lanish tushunchasini izohlang.
9. Munosabatlar orasidagi funksional bog'lanishga 3ta misol keltiring.
10. Munosabatlar orasidagi to'liq funksional bog'lanishni izohlang.
11. 2NSh ni izohlang va unga misol keltiring.
12. 2NSh ni bajarniaslik qanday noqulayliklarga olib keladi? Misollar bilan izohlang.
13. Tranzitiv bog'liqlik tushunchasini izohlang.
14. 3NSh ni ta'riflang va unga misol keltiring.
15. 3NSh da bo'lgan, lekin normal bo'lmagan munosabatga misol keltiring.
16. Boys-Kodd normal shaklini ta'riflang.
17. Boys-Kodd normal shaklidagi munosabatga misol keltiring.
18. Ko'p ma'noli bog'liqlik, deganda nimanı tushunasiz?
19. Ko'p ma'noli bog'lanishga misol keltiring.
20. Ko'p ma'noli bog'lanishdan qutilish uchun qanday usul qo'llaniladi.
21. 4NSh ni ta'riflang va unga misol keltiring.
22. Birlashtirish bog'liqligi tushunchasini ta'riflang.
23. Birlashtirish bog'liqligidan qutilish uchun qanday usul qo'llaniladi?
24. 5NSh ni ta'riflang va unga misol keltiring.
25. Normallashtirish jarayonidagi asosiy ikkita konsepsiyanı keltiring.

## 12-bob. Relatsion hisoblash

**Tayanch iboralar:** *relatsion tillar, relatsion hisoblash, relatsion to'liq til, predikat, kortejlarni relatsion hisoblash, domentlarni relatsion hisoblash, aniqlanish sohasi, mulohaza, formula, mavjudlik kvantori, umumiylilik kvantori, erkin o'zgaruvchilar, bog'liq o'zgaruvchilar, ifodalar, xavfsiz, transformatsion tillar.*

Berilganlar modelidagi muhim jihatlardan biri bu – berilganlarni ishlov qilish mexanizmi bo'lib (yoki so'rovlar tili), u berilganlarni tanlash va yangilash imkoniyatini beradi. So'rovlar tili relatsion modellar bilan bog'langan. Shuning uchun ham relatsion modellar asosidagi berilganlarga ishlov beruvchi tillarni *relatsion tillar* deb atashadi. Relatsion tillar asosida esa, relatsion algebra va relatsion hisoblash yotadi. Norasmiy ravishda, relatsion algebrani yuqori darajadagi protsedurali til sifatida qarash mumkin va uning yordamida BBB tizimiga yangi munosabatni, bitta yoki bir nechta munosabatlardan, *qanday* qurish kerakligini bildiradi. Relatsion hisoblashga esa, norasmiy jihatdan, protsedurali bo'lgan til sifatida qarash mumkin va uning yordamida, bitta yoki bir nechta munosabatlardan hosil bo'lgan yangi munosabat *qanaga* bo'lishi kerakligi ta'siflanadi. Lekin qat'iy ravishda aytsak, relatsion algebra va relatsion hisoblash bir-biriga ekvivalent, ya'ni relatsion algebraning har bir ifodasi uchun relatsion hisoblashning bitta ifodasi mos keladi va aksincha. Relatsion algebra va relatsion hisoblash rasmiy tillar bo'lib, foydalanuvchi uchun do'stona bo'lgan tillar majmuasiga kirmaydi. Relatsion algebra va relatsion hisoblash berilganlarni boshqarish uchun ishlataladigan yuqori darajadagi tillarni yaratishga asos bo'lgan. Biz uchun ularning muhim tomoni, berilganlarga ishlovdagi tilning asosiy *amallarini* ko'rish va relatsion tillarni bir-biri bilan *solishtirish* imkoniyatining yaratilishidir.

*Relatsion hisoblash relatsion tillarning quvvatini baholashga xizmat qiladi.* Relatsion hisoblash yordamida ixtiyoriy yangi munosabatni olish imkoniyatini beruvchi til, *relatsion to'liq til* deb ataladi. Ko'pchilik relatsion so'rov tillari relatsion to'liq tillar hisoblanadi. Lekin relatsion algebra va relatsion hisoblash bilan bu tillarni taqqoslasak, ular boshqa kengroq funksional imkoniyatlarga ham ega, chunki ularda hisoblash, agregatlash va tartiblash amallarini bajaruvchi qo'shiinchalar mavjud.

Relatsion algebra ifodalarida doimiy ravishda oshkor tarzda ma'lum bir tartib va so'rovlarini amalga oshirishdagi ma'lum bir strategiya yotadi. Relatsion hisoblashda esa, so'rovni amalga oshirish bo'yicha hech qanday *amallarni ketma-keitik bajarilish tartibi berilmaydi*, relatsion hisoblash asosidagi so'rovda uni qanday amalga oshirish emas, *nima* kerakligi ko't-satiladi. Uning differensial yoki integral hisoblashlarga hech qanday aloqasi yo'q, uning nomi mantiqiy belgilashdagi *predikat* (*qo'shimcha shart*)larni *hisoblash* iborasidan otingan. Berilganlar bazasiga nisbatan u ikki shaklda mavjud. Birinchisi E.F. Kodd tomonidan taklif qilingan *kortejlarni relatsion hisoblash* shakli [19], ikkinchisi esa, M. Lakrua va A. Pirotti tomonidan taklif qilingan *domenlarni relatsion hisoblash* [46] shaklidan iborat.

Predikatlarni hisoblash nazariyasida, *predikat* deganda, o'zgaruvchiga ega bo'lgan haqiqiy funksiya tushuniladi. O'zgaruvchi qiymatga ega bo'l-ganda, funksiya ifodaga aylanadi va uni *mulohaza deb* atashadi. Ushbu mulohaza esa, haqiqiy yoki yolg'on bo'lishi mumkin. Masalan, «*Latipov Mansur biologiya fakulteti talabasi*» va «*Valiyev Sodir Latipov Mansurdan kam stipendiya oladi*» iboralari, bu mulohazalardan iborat bo'lib, ulaming har birini haqiqiy yoki yolg'on ekanligini tekshirish mumkin. Birinchi mulohazada «*biologiya fakulteti talabasi*» funksiyasi bitta o'zgaruvchiga ega («*Latipov Mansur*»), ikkinchi mulohazadagi funksiya esa ikkita o'zgaruvchiga ega («*Latipov Mansur*», «*Valiyev Sodir*»).

Agarda predikatda o'zgaruvchi ishtirok etsa, masalan «*z biologiya fakulteti talabasi*» shaklda bo'lsa, u holda ushbu o'zgaruvchining *aniqlanish sohasi* bo'lishi zarur. Agarda z o'miga uning aniqlanish sohasidan ba'zi qiymatlar qo'yilsa, berilgan mulohaza haqiqiy, boshqa qiymatlar uchun esa yolg'on bo'lishi mumkin. Masalan, aniqlanish sohasi biologiya fakultetini o'z ichiga olgan oliy o'quv yurtining barcha talabalaridan iborat bo'lsa, z o'miga «*Latipov Mansur*» qiymatini qo'ysak, u holda «*Latipov Mansur biologiya fakulteti talabasi*», degan mulohaza haqiqiy bo'ladi. Agarda ushbu oliy o'quv yurtining talabasi bo'lмаган, ixtiyoriy boshqa talabaning familiyasi va ismini qo'ysak, mulohaza yolg'on bo'ladi.

Agarda R – predikat bo'lsa, R mulohaza rost bo'ladigan z o'zgaruvchining hamma haqiqiy qiymatlar to'plamini, sharthi tarzda  $\{z \mid P(z)\}$  shaklda yozamiz.

Predikatlar, mantiqiy Bul amallari AND ( $\wedge$ ), OR(V) va NOT ( $\neg$ ) orqali birlashgan holda tarkibli predikatlarni hosil qiladi.

### Kortejlarni relatsion hisoblash

Kortejlarni relatsion hisoblashdagi masala, predikatlari haqiqiy bo'lgan kortejlarni topishdan iborat. Ushbu hisoblash kortejlardagi o'zgaruvchilarga

*asoslangan. Kortejlardagi o'zgaruvchilar deganda, aniqlanish sohasi ko'rsatilgan munosabat bo'lgan o'zgaruvchilar tushuniladi. Olishi mumkin bo'lgan qiymati faqat shu munosabatning kortejlariga tegishli bo'lgan o'zgaruvchilardan iborat. Bu erda aniqlanish sohasi deganda, qiymatning olish mumkin bo'lgan chegara emas, balki shu qiymatlar aniqlangan domen tushuniladi.*

Masalan, MUXANDIS munosabatini (10.1 rasmdagi) Z kortejidagi o'zgaruvchining aniqlanish sohasi ekanligini ko'rsatish, quyidagicha ifodalanadi:

### **MUXANDIS(Z).**

E'tiboringiz uchun, BBB tizimidagi belgilashlar bilan yagonalikka erishish uchun yana lotin harflariga o'tamiz.

Bundan tashqari, «F(Z) rost bo'ladigan, Z kortejlarning hamma to'plamini aniqlang», degan so'rovni quyidagicha yozish mumkin:

### **{Z | F(Z)}**

Bu erda F predikat formula (matematik mantiqda to'g'ri qurilgan formula – Well-Formed Formula yoki WFF) deb ataladi. Masalan, «MUXANDIS munosabatidan 1700dan ortiq maosh oladigan hamma xizmatchilarning Xizmatchi\_nomeri, Fish, Bo'lim\_nomeri, Maosh, Manzil atributlari tanlansin» so'rovini quyidagicha yozish mumkin:

### **{Z | MUXANDIS(Z) $\wedge$ Z.Maosh > 1700}**

Bu yerda, Z.Maosh ifodasi, «Maosh atributining qiymati Z korteji uchun aniqlansin», degan ma'noni anglatadi. Agarda faqat Maosh atributini tanlamoqchi bo'lsak, so'rov quyidagicha yoziladi:

### **{Z.Maosh | MUXANDIS(Z) $\wedge$ Z.Maosh > 1700}**

## **Mavjudlik va umumiylilik kvantorlari**

Predikat qo'llanilishi kerak bo'lgan munosabat nusxalarining sonini ko'rsatish uchun ikki xildagi kvantor (*shart*)lar ishlataladi. Birinchisi, mavjudlik kvantori ( $\exists$ ) yoki mavjudlik belgisi, formulada kamida bitta munosabat nusxasi haqiqiy bo'lishi uchun ishlataladi, masalan:

**{RO'YXAT(Z)  $\wedge$  ( $\exists$  W) (FAKULTETLAR(W)  $\wedge$  (W.FAK\_N = Z.FAK\_N)  $\wedge$  (W.FAKULTET = 'Biologiya'))}**

Bu ifoda quyidagi ma'noni anglatadi:

«FAKULTETLAR munosabatida shunday kortej mavjudki, undagi FAK\_N atributining qiymati, RO'YXAT munosabatining joriy Z kortejdagi FAK\_N atributining qiymati bilan bir xilda va xuddi shu paytda W kortejdagi FAKULTET atributining qiymati esa 'Biologiya'».

Ikkinchisi, umumiylig kuantori ( $\forall$ ) yoki «hamma uchun» belgisi, ifodadagi hamma munosabatlarning musxalariga tegishli bo'lishi uchun ishlataladi, masalan:

### ( $\forall W$ ) (W. FAKULTET $\neq$ 'Biologiya')

Bu ifoda, «FAKULTETLAR munosabatining barcha kortejlar uchun Fakultet atributining qiymati 'Biologiya'ga teng emas», degan ma'noni anglatadi.

Mantiqiy amallarga nisbatan, quyidagi ekvivalentlik qoidalari qo'llanilishi mumkin:

$$\begin{aligned}(\exists Z) (F(Z)) &\equiv \neg (\forall Z) (\neg (F(Z))) \\(\forall Z) (F(Z)) &\equiv \neg (\exists Z) (\neg (F(Z))) \\(\forall Z) (F_1(Z) \equiv F_2(Z)) &\equiv \neg (\exists Z) (\neg (\neg F_1(Z) \vee F_2(Z)) \vee \neg (\neg F_1(Z) \vee F_2(Z))) \\(\exists Z) (F_1(Z) \equiv F_2(Z)) &\equiv \neg (\forall Z) (\neg (\neg F_1(Z) \vee F_2(Z)) \vee \neg (\neg F_1(Z) \vee F_2(Z)))\end{aligned}$$

Shunga asosan, yuqorida keltirilgan formulani quyidagicha yozish mumkin:

### $\neg (\exists W)$ (W.Fakultet = 'Biologiya')

Ushbu ko'rinishda bu formula, Fakultet atributida 'Biologiya' yo'q (bo'lmasin), degan ma'noni anglatadi.

Agarda kortejning o'zgaruvchilari  $\exists$  va  $\forall$  kuantorlari bilan malakanmagan bo'lsa, erkin o'zgaruvchilar deyiladi, aks holda bog'liq o'zgaruvchilar deyiladi. Relatsion hisoblash qoidalariga asosan yozilgan ifodada, erkin o'zgaruvchilar faqatgina tik chiziqning {} chap tomonida turishi mumkin.

Biz quyida, 7- bobda keltirilgan Talabalar o'quv loyihasining modeliga asoslangan holda, undagi munosabatlar ustida misollar keltiramiz. Masalan, quyidagi so'rovda erkin o'zgaruvchilar Z.FAMILIYA, Z.ISM, Z.O\_ISM bo'lib, ular shu ifodani hisoblash jarayonida ketma-ket ravishda RO'YXAT va FAKULTETLAR munosabatlarining har bir korteji bilan bog'lanadi:

$$\{Z.FAMILIYA, Z.ISM, Z.O\_ISM \mid RO'YXAT(Z) \wedge (\exists W) (FAKULTETLAR (W) \wedge (W.FAN_N = Z.FAN_N) \wedge (W.FAKULTET = 'Biologiya'))\}$$

Ushbu so'rov natijasida biologiya fakultetidagi talabalarning har bir fanga bog'liq ro'yxati tuziladi.

## Ifodalar va formulalar

Relatsion hisoblashdagi har bir formulani ham to'g'ri, deb bo'lmaydi. Mumkin bo'lgan formulalar yagona va to'g'ri ma'noga ega bo'lishi shart. Kortejlarni relatsion hisoblashdagi ifodalar quyidagi umumiy shaklga ega:

$$\{Z_1.a_1, Z_2.a_2, \dots, Z_m.a_m \mid F(Z_1, Z_2, \dots, Z_m)\}$$

Bu erda  $m > n$ ,  $Z_1, Z_2, \dots, Z_n, \dots, Z_m$  – kortej o'zgaruvchilari,  $a_i$  munosabat atributlari bo'lib, unda  $Z_i$  o'zgaruvchining qiymati aniqlangan,  $F$  esa formula.

To'g'ri tuzilgan formula bitta yoki bir nechta oddiy ifodalardan iborat bo'lishi va quyidagi shakklardan biriga ega bo'lishi mumkin:

- $R(Z_i)$ , bu erda  $Z_i$  – kortej o'zgaruvchisi,  $R$  – munosabat;

•  $Z_i.a_1 * Z_j.a_2$ , bu erda  $Z_i$  va  $Z_j$  – kortej o'zgaruvchilari,  $a_1 = Z_i$  o'zgaruvchining qiymati aniqlangan munosabat atributi,  $a_2 = Z_j$  o'zgaruvchining qiymati aniqlangan munosabat atributi,  $*$  – taqqoslash amallaridan biri ( $\leq, <, \geq, >, =, \neq$ );  $a_1$  va  $a_2$ , atributlarining aniqlanish sohasi taqqoslash amallaridan birini qo'llash imkoniyatini berishi shart;

•  $Z_i.a_1 * v$ , bu erda  $Z_i$  kortej o'zgaruvchisi,  $a_1 = Z_i$  o'zgaruvchining qiymati aniqlangan munosabat atributi,  $v$  –  $a_1$  atributning aniqlanish sohasiga tegishli o'zgarmas,  $*$  – taqqoslash amallaridan biri.

Formulalar rekursiv ravishda, quyidagi qoidalar asosida, elementar ifodalardan quriladi:

- ixtiyoriy oddiy ifoda formula kabi qaratadi;

• agarda  $F_1$  va  $F_2$  ifodalar formulalar bo'lsa, u holda ularning kon'yunksiyasi ( $F_1 \wedge F_2$ ), dizyunksiyasi ( $F_1 \vee F_2$ ) va inkori ( $\neg F_1$ ) ham formula bo'ladi;

• agarda  $F$  ifodasi  $X$  erkin o'zgaruvchiga ega bo'lgan formula bo'lsa, u holda ( $\exists X$ ) ( $F$ ) va ( $\forall X$ ) ( $F$ ) ifodalar ham formula bo'ladi.

## Kortejlarni relatsion hisoblashga misollar

Quyidagi hamma misollar 7- bobdag'i Talabalar o'quv loyihasining modelidagi munosabatlarga asoslangan.

1. Ta'llim yo'nalishining raqami 25 bo'lgan 3- va 4- kurs talabalarining ro'yxatini tuzing.

$\{Z.FAMILIYA, Z.ISM, Z.O_ISM \mid ROYHAT(Z) \wedge Z.YON_N = 25 \wedge Z.KURS > 2\}$

2. Ta'lif yo'naliishi «Informatika va axborot texnologiyalari» (IAT) bo'lgan «Berilganlar bazasini boshqarish tizimlari» (BBBT) sanidan 5 baho olgan 4- kurs talabalarning ro'yxatini tuzing.
- {Z.FAMILIYA, Z.ISM, Z.O\_ISM | RO'YHAT(Z)  $\wedge$  (( $\exists$  W) ( $\exists$  R) (YO'NALISH(W)  $\wedge$  FANLAR(R)  $\wedge$  BAHOLAR(P)  $\wedge$  (Z.ZN=P.ZN)  $\wedge$  (Z.YON\_N=W.YON\_N)  $\wedge$  W.YO'NALISH-LAR=IAT'  $\wedge$  Z.KURS=4  $\wedge$  (R.FAN\_N=P.FAN\_N)  $\wedge$  R.FAN\_NOMI = 'BBBT'  $\wedge$  P.BALL > 85))}
3. Talabalarni BGBT sanidan reyting balini a'loga bahoqagan ustozlarni aniqlang.
- {R.USTOZ | BAHOLAR(P)  $\wedge$  (( $\exists$  R) (FANLAR(R)  $\wedge$  (R.FAN\_N=P.FAN\_N)  $\wedge$  R.FAN\_NOMI = 'BBBT'  $\wedge$  P.BALL > 85))}
4. Hamma a'lochi talabalarning ro'yxatini tuzing.
- {Z.FAMILIYA, Z.ISM, Z.O\_ISM | RO'YHAT(Z)  $\wedge$  (( $\exists$  R) (BAHOLAR(R)  $\wedge$  (Z.ZN=R.ZN)  $\wedge$  R.BALL > 85))}
5. Hamma qarzdor talabalamining ro'yxati tuzilsin.
- {Z.FAMILIYA, Z.ISM, Z.O\_ISM | RO'YHAT(Z)  $\wedge$  (( $\exists$  R) (BAHOLAR(R)  $\wedge$  (Z.ZN=R.ZN)  $\wedge$  R.BALL < 55))}
6. Semestr 31.01.11 tugaganini hisobga olgan holda, qarzdor bo'lib, 15 kun ichida fanlarni o'zlashtirgan hamma talabalarning ro'yxati tuzilsin.
- {Z.FAMILIYA, Z.ISM, Z.O\_ISM | RO'YHAT(Z)  $\wedge$  (( $\exists$  R) (BAHOLAR(R)  $\wedge$  (Z.ZN=R.ZN)  $\wedge$  R.SANA\_KIR > 31.01.2011  $\wedge$  R.SANA\_KIR < 16.02.2011  $\wedge$  R.BALL > 55))}
7. Har bir fakultet bo'yicha 8- semestrdagi qarzdor talabalar ro'yxati tuzilsin.
- {Z.FAMILIYA, Z.ISM, Z.O\_ISM, W.FAKULTET | RO'YHAT(Z)  $\wedge$  (( $\exists$  R) ( $\exists$  W) (BAHOLAR(R)  $\wedge$  FAKULTETLAR(W)  $\wedge$  (Z.ZN=R.ZN)  $\wedge$  (Z.FAK\_N=W.FAK\_N)  $\wedge$  R.SEMESTR = 8  $\wedge$  R.BALL < 55))}
8. «Mexanika-matematika» fakulteti bo'yicha barcha a'lochi talabalarning ro'yxatini tuzilsin.
- {Z.FAMILIYA, Z.ISM, Z.O\_ISM | RO'YHAT(Z)  $\wedge$  (( $\exists$  R) ( $\exists$  W) (BAHOLAR(R)  $\wedge$  FAKULTETLAR(W)  $\wedge$  (Z.ZN =

$R.ZN) \wedge (Z.FAK\_N = W.FAK\_N) \wedge R.BALL > 85) \wedge$   
 $(W.FAKULTET = 'mexanika-matematika'))\}$

9. Talabalarini BBBT fanidan reyting balini a'lo yoki yaxshiga baholagan ustozlarni aniqlang.

$\{R.USTOZ \mid BAHOLAR(R) \wedge ((\exists P) (FANLAR(P) \wedge$   
 $(R.FAN\_N = P.FAN\_N) \wedge P.FAN\_NOMI = 'BBBT' \wedge$   
 $((R.BALL > 85) \text{ OR } (R.BALL > 70 \wedge R.BALL < 86)))\}$

10. 3-kurs talabalarini ikkiga baholagan barcha professor-o'qituvchilarning ro'yxati tuzilsin.

$\{R.USTOZ \mid BAHOLAR(R) \wedge ((\exists Z) (RO'YXAT(Z) \wedge$   
 $(Z.ZN=R.ZN) \wedge Z.KURS = 3 \wedge R.BALL < 55))\}$

### Domenlarni relatsion hisoblash

Domenlarni relatsion hisoblashda ham o'zgaruvchilar ishlataladi, ularning qiymatlari atributlarning aniqlanish sohasidan olinadi. Domenlarni relatsion hisoblashdag'i ixtiyoriy ifoda quyidagi umumiyl shaklga ega:

$\{d_1, d_2, \dots, d_n \mid F(d_1, d_2, \dots, d_m)\}$

Bu erda  $m > n$ ,  $d_1, d_2, \dots, d_n, \dots, d_m$  – domenning aniqlanish sohasidagi o'zgaruvchilar,  $F(d_1, d_2, \dots, d_m)$  esa formula.

Formulalar bir yoki bir nechta oddiy ifodalardan iborat bo'lib, quyidagi shakklardan biriga ega bo'lishi mumkin:

- $R(d_1, d_2, \dots, d_n)$ , bu erda  $R$  –  $n$ - darajali munosabat,  $d_i$  – domen o'zgaruvchisi;

- $d_i * d_j$ , bu erda  $d_i$  va  $d_j$  – domen o'zgaruvchilar,  $*$  – taqqoslash amallaridan biri ( $\leq, <, \geq, >, =, \neq$ );  $d_i$  va  $d_j$  o'zgaruvchilarning aniqlanish sohasi taqqoslash amallaridan birini qo'llash imkoniyatini yaratishi shart;

- $d_i * v$ , bu erda  $d_i$  domen o'zgaruvchisi,  $v$  –  $d_i$  domenning aniqlanish sohasiga tegishli o'zgarmas,  $*$  – taqqoslash amallaridan biri.

Formulalar rekursiv ravishda, quyidagi qoidalar asosida, elementar ifodalardan quriladi:

- ixtiyoriy oddiy ifoda formula kabi qaraladi;

- agarda  $F_1$  va  $F_2$  ifodalar formulalar bo'lsa, u holda ularning konyunksiyasi ( $F_1 \wedge F_2$ ), dizyunksiyasi ( $F_1 \vee F_2$ ) va inkori ( $\neg F_1$ ) ham formula bo'ladi;

- agarda  $F$  ifodasi  $X$  erkin o'zgaruvchiga ega bo'lgan formula bo'lsa, u holda  $(\exists X)(F)$  va  $(\forall X)(F)$  ifodalar ham formula bo'ladi.

## Domenlarni relatsion hisoblashga misollar

Domenlarni relatsion hisoblashdagi quyidagi hamma misollar ham, *Talabalar o'quv loyihasining modelidagi munosabatlarga asoslangan*. Kortejlarni relatsion hisoblash bilan domenlarni relatsion hisoblashni taqqoslash uchun, kortejlarni relatsion hisoblashdagi hamma misollarning shartlarini o'zgartirmagan holatda quyida keltirdik.

Faqatgina, quyida keltirilgan formulalarda qisqartirish kiritilgan:

$$\exists (\exists d_1), (\exists d_2), \dots, (\exists d_n) \text{ o'miga } (\exists d_1, d_2, \dots, d_n)$$

Agar ikkita munosabatdagi domenlar bir xil nomga ega bo'lsa, ular shu domenlar bo'yicha tutashtiriladi.

1. Ta'lim yo'naliشining raqami 25 bo'lgan 3- va 4- kurs talabalarining ro'yxatini tuzing.

{FAM, IS, OI | ( $\exists$  KS, FN, YN, K, PS, RN) (RO'YHAT(FAM, IS, OI, KS, FN, YN, K, PS, RN)  $\wedge$  YN = 25  $\wedge$  K>2)}

Ushbu so'rnvi kortejlarni relatsion hisoblashdagi 1- misol bilar solishtirsak, har bir atributga (o'zgaruvchi) nomi berilganini ko'ramiz RO'YHAT(FAM, IS, OI, KS, FN, YN, K, PS, RN) sharti, domenning o'zgaruvchilari, shu kortejlarning atributlari bilan chegaralanishin kafolatlaydi. Shuning uchun RO'YXAT.YON\_N = 25 formulasi o'miga YN = 25, RO'YHAT.KURS>2 formulasi o'miga esa K>2 formulasini ishlatishimiz mumkin.

Shuningdek, mayjudiik kvantorini ishlatischdag'i farqni ham hisobga olish kerak. Birinchidan, kortejlarni relatsion hisoblashda ushbu kvantorni bironqa FAM kortejining o'zgaruvchisini  $\exists$  FAM shaklda ishlatish, shu o'zgaruvchini RO'YHAT munosabatining RO'YHAT(FAM) ifodasiida foydalanish kabi teng kuchli. Ikkinchidan, domenlarni relatsion hisoblashda FAM o'zgaruvchisi domendagi qiymatlardan biriga havola qiladi va unga qo'yilgan chegara RO'YHAT(ZN,FAM, IS, OI, KS, FAKN, YN, K, PS, RN) kabi formulada uchramaguncha, olib tashlanmaydi, undan keyin uning qiymati RO'YHAT munosabatidagi FAMILIYA domeni qiymatlari bilar chegaralanadi. Ana shu xulosalarga asosan, domenlarni relatsion hisoblashda, doimiy ravishda to'g'ri natijaga erishishimiz mumkin. Demak domenlarni relatsion hisoblashdagi formulalarni *xavfsiz* deyishimi: mumkin.

Quyidagi misollarda qisqalik uchun kvantorlar, faqatgina shartlarda ishlataligan domen o'zgaruvchilariga qo'llanilgan.

2. Ta'lim yo'nalishi «Informatika va axborot texnologiyalari» (IAT) bo'lgan «Berilganlar bazasini boshqarish tizimlari» (BBBT) fanidan 5 baho olgan 4- kurs talabalarning ro'yxatini tuzing.

{FAM, IS, OI | ( $\exists$  YON, FAN, K, BL) (RO'YHAT(ZN, FAM, IS, OI, KS, FAKN, YN, K, PS, RN)  $\wedge$  YO'NALISH(YN, YON)  $\wedge$  FANLAR(FN, FAN)  $\wedge$  BAHOLAR(ZN, S, FN, BL, SK, US)  $\wedge$  YON = 'IAT'  $\wedge$  K = 4  $\wedge$  FAN = 'BBBT'  $\wedge$  BL > 85)}

3. Talabalarni BGBT fanidan reyting balini a'loga baholagan ustozlarni aniqlang.

{US | ( $\exists$  FAN, BL) (FANLAR(FN, FAN)  $\wedge$  BAHOLAR(ZN, S, FN, BL, SK, US)  $\wedge$  FAN = 'BBBT'  $\wedge$  BL > 85)}

4. Hamma a'luchi talabalarning ro'yxatini tuzing.

{FAM, IS, OI | ( $\exists$  BL) (RO'YHAT(ZN, FAM, IS, OI, KS, FAKN, YN, K, PS, RN)  $\wedge$  BAHOLAR(ZN, S, FN, BL, SK, US)  $\wedge$  BL > 85)}

5. Hamma qarzdor talabalarning ro'yxati tuzilsin.

{FAM, IS, OI | ( $\exists$  BL) (RO'YHAT(FAM, IS, OI, KS, FN, YN, K, PS, RN)  $\wedge$  BAHOLAR(ZN, S, FN, BL, SK, US)  $\wedge$  BL < 55)}

6. Semestr 31.01.11 tugaganini hisobga olgan holda qarzdor bo'lib, 15 kun ichida fanlarni o'zlashtirgan hamma talabalarning ro'yxati tuzilsin.

{FAM, IS, OI | ( $\exists$  SK, BL) (RO'YHAT(FAM, IS, OI, KS, FN, YN, K, PS, RN)  $\wedge$  BAHOLAR(ZN, S, FN, BL, SK, US)  $\wedge$  (SK > 31.01.2011  $\wedge$  SK < 16.02.2011)  $\wedge$  BL > 55)}

7. Har bir fakultet bo'yicha 8- semestrda qarzdor talabalar ro'yxati tuzilsin.

{FAM, IS, OI, FAK | ( $\exists$  FAK, BL) (RO'YHAT(ZN, FAM, IS, OI, KS, FAKN, YN, K, PS, RN)  $\wedge$  BAHOLAR(ZN, S, FN, BL, SK, US)  $\wedge$  FAKULTETLAR(FAKN, FAK)  $\wedge$  (S = 8  $\wedge$  BL < 55))}

8. «Mexanika-matematika» fakulteti bo'yicha barcha a'lochi talabalarning ro'yxati tuzilsin.

{FAM, IS, OI | ( $\exists$  BL, FAK) (RO'YHAT (ZN, FAM, IS, OI, KS, FAKN, YN, K, PS, RN)  $\wedge$  BAHOLAR (ZN, S, FN, BL, SK, US)  $\wedge$  FAKULTETLAR (FAKN, FAK)  $\wedge$  BL > 85  $\wedge$  FAK = 'mexanika-matematika')}

9. Talabalarni BBBT fanidan reyting balini a'lo yoki yaxshiga baholagan ustozlarni aniqlang.

{US | ( $\exists$  FAN, BL) (FANLAR (FN, FAN)  $\wedge$  BAHOLAR (ZN, S, FN, BL, SK, US)  $\wedge$  FAN = 'BBBT'  $\wedge$  (BL > 85) V (BL > 70 AND BL < 86))}

10. 3-kurs talabalarini ikkiga baholagan barcha professor-o'qituvchilarning ro'yxati tuzilsin.

{US | ( $\exists$  K, BL) ((RO'YHAT (ZN, FAM, IS, OI, KS, FAKN, YN, K, PS, RN)  $\wedge$  BAHOLAR (ZN, S, FN, BL, SK, US)  $\wedge$  K = 3  $\wedge$  BL < 55)}

Yuqorida keltirilgan so'rovlarning hammasi xavfsiz. Domenlarni relationsion hisoblashda faqat ana shunday ifodalar bilan cheklanilsa, ular kortejlarni relationsion hisoblashdagi ifodalarga ekvivalent bo'ladi. Kortejlarni relationsion hisoblash esa relationsion algebraga ekvivalent. Demak, relationsion algebraning har bir ifodasiga relationsion hisoblashning bitta ekvivalent ifodasi to'g'ri keladi. Domenlarni yoki kortejlarni relationsion hisoblashdagi har bir ifodaga relationsion algebraning bitta ekvivalent ifodasi mos keladi.

Relationsion hisoblash o'zlashtirish va foydalanish nuqtayi nazaridan yetarlicha murakkab bo'lsa ham, uning protsedurasizlik xossasi hisobiga kelajagi porloq bo'lib, bu foydalanish uchun protsedurasiz boshqa yetarlicha sodda usullarni izlashga majbur qiladi. Ana shunday izlanishlar natijasida relationsion tillarning ikkita, ya'ni transformatsion va grafik sinflari paydo bo'ldi.

*Transformatsion tillar* protsedurasiz tillar sinfini tashkil etib, berilganlarni o'zgartirib zaruriy shaklga keltirish uchun munosabatlardan foydalanadi. Ushbu tillar foydalanish uchun sodda tuzilishlarni ishlataladi. Transformatsion tillarga misollar – SQUARE [5,49], SEQUEL [8,49], SEQUEL2

[6] hamda SQL [2,58,61,74,80,81]. SQL tili xususida biz 23-30 boblarda to'xtab o'tamiz.

Grafik tillar chizma yoki munosabatlarni tuzilishini grafik akslan-  
tiruvchi boshqa vositalarni taklif qiladi. Bu tilga misol tariqasida QBE  
(Query-By-Example) [54,55] tilini keltirish mumkin. Bu tilda relatsion  
algebra va relatsion hisoblashlarda ko'zda tutilmagan bir qancha imkoniyatlar  
kiringan. Bu tilni xususiyati shundan iboratki, u terminallar orqali  
ishlashga mo'ljalangan. Foydalanuvchi olmoqchi bo'lgan natijasining biror  
namunasini, maxsus ekran tahrirchisi, munosabat va taxrirlardan  
foydalanib yaratadi va BBB tizimi so'ralgan berilganlarni, talab qilingan  
shaklda chiqarib beradi. QBE tilining imkoniyatlari bilan 22- bobda tanishib  
chiqamiz.

### Xulosha

Berilganlarning relatsion modelida relatsion tillar ishlatalishi, relatsion  
tillarning asosida esa relatsion algebra va relatsion hisoblash yotishini  
ta'kidlagan edik. Relatsion algebrani norasmiy ravishda yuqori darajali  
protsedurani til sifatida tavsiflagan bo'lsak, relatsion hisoblashni esa  
norasmiy ravishda yuqori darajali protsedurasiz til sifatida tavsiflash  
mumkin. U orqali BBB tizimidan, berilganlar bazasidagi bitta yoki bir  
nechta munosabatdan, bitta yangi qurilgan munosabat qanaqa bo'lishi  
mumkinligini aniqlash imkoniyati yaratiladi.

Relatsion hisoblash relatsion tillarning tanlov quvvatini quvvatini  
baholashda ishlatalidi. Relatsion to'liq tillar, relatsion hisoblash yordamida  
hosil qilish mumkin bo'lgan ixtiyoriy munosabatni olish imkoniyatini  
yaratadi. So'rov uchun ishlataladigan ko'pchilik relatsion tillar, relatsion  
to'liq hisoblanadi, lekin ular xususan, agregatlash va tartiblash funksiylaridan  
foydalanish hisobiga, relatsion algebra va relatsion hisoblashga  
nisbatan, katta ma'nodorlik quvvatiga ega.

### Nazorat uchun savollar

1. Relatsion tillar asosida nimalar yotadi?
2. Relatsion algebra va relatsion hisoblash orasida qanday bog'lanishlar  
mavjud?
3. Relatsion hisoblash va relatsion tillar orasida qanday bog'lanishlar  
mavjud?
4. Relatsion hisoblashda so'rovlар ketma-ket kelish tartibi qanday ahamiyatga ega?
5. Relatsion hisoblash asosan nimaga bog'liq?

6. Protsedurali va protsedurasiz tillar orasidagi farq nimadaň iborat?
7. Relatsion to'liq tilni izohlab bering.
8. Relatsion to'liq tilda qanday qo'shimcha amallar mavjud?
9. Relatsion hisoblash protsedurali tilmi yoki protsedurasiz tilmi?
10. Predikat deganda nimani tushunasiz?
11. Mułohaza deganda nimani tushunasiz?
12. Kortejlarni relatsion hisoblashdagi predikat o'zgaruvchisining aniqlanish sohasini ifodalab bering.
13. Erkin va bog'liq o'zgaruvchilarni ta'riflang.
14. Relatsion hisoblashdagi formulaga misol keltiring.
15. Mavjudlik va umumiylig kvantorlarining farqini tushuntirib bering.
16. Domenlarni relatsion hisoblashdagi predikat o'zgaruvchisining aniqlanish sohasini ifodalab bering.
17. MUXANDIS munosabati asosida kortejlarni relatsion hisoblashga misollar keltiring.
18. MUXANDIS munosabati asosida domenlarni relatsion hisoblashga misollar keltiring.
19. Transformatsion tillar protsedurali tillar sinfigami yoki protsedurasiz tillar sinfiga kiradimi?
20. Transformatsion tillarga misol keltiring.
21. Grafik tillar protsedurali tillar sinfigami yoki protsedurasiz tillar sinfiga kiradimi?
22. Grafik tillarga misol keltiring.

### **13-bob. Berilganlarning relatsion butunligi**

**Tayanch iboralar:** *butunlik chegaralari, oshkor chegaralar, ichki chegaralar, berilganlarning relatsion butunligi, domen chegarasi, mohiyatlarning butunligi, semantik butunlik va havola butunligi, bo'sh qiymat, uch ma'noli mantiq, asos munosabat, ajdod, avlod, berilganlar turi, oshkormas qiymat, chegarani tekshirish, korporativ chegara, taqdimot, nazariy jihatdan yangilanmaydigan taqdimot, nazariy jihatdan yangilanadigan taqdimot, qisman yangilanadigan taqdimot, kiritish amali, yangilash amali, olib tashlash (yo'qotish) amali, RESTRICT (cheklash), CASCADE (kaskadli o'zgartirish), IGNORE (inkor etish), NULL, SET NULL (NULL berish), SET DEFAULT (oshkormas ravishda qiymat berish).*

Berilganlarga qo'yiladigan mantiqiy chegaralar, *butunlik chegaralari* deb ataladi. BBB tizimlari, berilganlar bazasini bir holatdan boshqa holatga o'tganda, berilganlarning qo'yilgan chegaralarga mosligini nazorat qilishi kerak. Chegaralardan foydalanish, predmet sohani to'g'ri akslantirish bilan ham bog'liq.

*Oshkor chegaralar* predmet sohaning semantikasi bilan bog'liq. Ular atributlarning olishi mumkin bo'lgan qiymatlar sohalarini, atributlar orasidagi munosabatlarni, ularning o'zgarish dinamikasini va h.k. taysiflaydi. *Ichki chegaralar* faqatgina berilganlar modeliga xos. Ular, berilganlarning tanlab olingan modeliga kiritilgan munosabatlarning tuzilishi, aloqalar, berilganlar majmuasining olishi mumkin bo'lgan qiymatlar to'plamiga ta'sir qiladi. Berilganlarning ichki chegaralaridagi butunlikni amalga oshirish usullari BBB tizimiga bog'liq.

Berilganlar modelini shartli ravishda ikki qismga ajratish mumkin – berilganlar bilan mumkin bo'lgan amallarni turlarini aniqlovchi boshqaruvchi qism va berilganlarning to'g'riligini kafolatlovchi butunlik chegaralarining majmuasi. Ushbu bobda biz relatsion butunlik chegaralari va berilganlarni boshqarishdagi relatsion amallarning butunlikka ta'sirini ko'rib chiqamiz.

Har bir atribut qaysidir domenga bog'liq bo'lgani uchun, munosabatdagi har bir atributning qabul qilishi mumkin bo'lgan to'plami uchun *domen chegarasi* degan tushuncha aniqlanadi. Bundan tashqari, uchta muhim *butunlikning qoidusi* beriladi, ular aslida berilganlar bazasining hamma mumkin bo'lgan holatlaridagi chegaralarni belgilaydi. Relatsion

modeldag'i bu asosiy uchta qoida *mohiyatlarning butunligi, sentantik butunlik va havola butunligi* deb ataladi. Bu uchta qoidani o'rganishdan avval bo'sh qiymatlar tushunchasini qarab chiqamiz.

### Bo'sh qiymat va uch ma'noli mantiq

*Bo'sh qiymat* atributning qiymati hozirgi vaqtida noma'lum yoki ushbu kortejda ishlatalish mumkin emasligini bildiradi. Bo'sh qiymat shartli ravishda **NULL** sifatida belgilanadi va u «noma'lum» mantiqiy kattalik sifatida qaraladi. Boshqacha aytganda, bu qaysidir kortejning aniqlanish sohasiga tegishli bo'lmaydi yoki bali hech qanday qiymat berilmagan bo'ladi. **NULL** kalit so'zi to'liq bo'lmagan yoki g'alati berilganlarni bildiradi. Lekin **NULL**, nol qiymatni ham yoki bo'sh katakchalar bilan to'ldirilgan satni ham bildirmaydi. Nol va bo'sh katakchalar qandaydir qiymatlarni bildir-sada, **NULL** qiymatning yo'qligini bildiradi. Shuning uchun **NULL** boshqa qiymatlarga nisbatan alohida xususiyatga ega.

Masalan, Talabalar o'quv loyihasidagi Ball atributi har o'quv semestri tugagandan keyin ma'lum muddat davomida ayrim kortej qiymatlari bo'sh turishi mumkin, chunki talabalar ayrim fanlarni o'z vaqtida o'zlashtirmagan bo'lishi mumkin. Agarda bu holatni **NULL** kaitli so'zsiz ifodalaydigan bo'lsak, u holda soxta qiymatni berish yoki foydalanuvchi uchun ma'nosiz tuyulgan atributni kiritish mumkin. Bundan tashqari BAHOLAR munosa-batiga yana bitta, Y (Yes) yoki N (No) qiymatli – **ha s Comment Be enSuppl led (izoh bormi) degan**, atribut qo'shish mumkin. Lekin bu ikkala yondashuv foydalanuvchini faqat chalg'itishi mumkin.

**NULL** dan foylanish amaliyotda muammolar keltirib chiqarishi mumkin, chunki ular birinchi tartibli predikatlarni hisoblash bilan bog'liq, ya'ni ikki qiymatli (Bul) mantiqqa asoslangan (qabul qilishi mumkin bo'lgan qiymatlari «haqiqat» yoki «yo'lg'on»).

**NULL** ni qo'llash, yuqori tartibli mantiqdan, masalan, uchinchi va hatto to'rtinchi tartibli [13,14,18] foydalanishni va u bilan ishlashni talab qiladi. Relatsion modellarda **NULL** dan foydalanish bahslarga sabab bo'lgan mavzu. E.F. Kodd [18] **NULL** ni relatsion modelning tarkibiy qismi sifatida qarashni taklif etgan bo'lsa, boshqa mutaxassislar esa bu yondashuvni xato deb hisoblashgan, chunki mavjud bo'lmagan ma'lumotning muammosi hali oxirigacha tushunib yetilmagan, qoniqarli yechim topilmadi, shuning uchun relatsion modelga **NULL** ni kiritishga hali vaqt bor, deb hisoblashgan. Relatsion model nazariyasining eng yirik nam-yondalaridan biri bo'lgan K. Deyt **NULL** qiymatga qat'iy qarshi chiqqan. K. Deyt **NULL**dan foydalanish natijasida kelib chiqadigan muammolarni batafsil keltirgan [66].

Lekin yetarli darajadagi nazariy asoslanmagan bo'lishiga qaramasdan, zamonaviy relatsion BBB tizimlarining deyarli hammasida NULL dan foydalananish imkoniyati yaratilgan. NULL ishtirokidagi ifodalarni taqqoslashdagi natija noma'lum bo'lishi mumkin. Masalan,  $x = \text{NULL}$  yoki  $\text{NULL} = \text{NULL}$  ifodalari uchun haqiqat ma'nosi NULL.

Shuning uchun biz, bu erda NULL bilan ishslashda zarur bo'lgan *uch ma'noli mantiqning (3VL – three-valued logic)* tavsifini keltiramiz. Uch qiymatli mantiqda mantiqiy ifodalarning haqiqatligini aniqlash uchun T – True (post) va F – False (yolg'on)dan tashqari U – Unknown (noma'lum) kiritilgan [82]. Mantiqiy ma'nodagi U – noma'lum, NULL ning xuddi o'zi.

Mantiqiy VA (AND), YOKI (OR) va YO'Q (NOT) amalidan foydalishdagi X va Y operandlarning qiymatiga nisbatan ifodalarning haqiqat ma'nosi mos ravishda quyidagi 13.1-jadvalda keltirilgan:

13.1-jadval.

X	Y	X AND Y	X OR Y	NOT X
True	True	True	True	False
True	False	False	True	False
True	Unknown	Unknown	True	False
False	True	False	True	True
False	False	False	False	True
False	Unknown	False	Unknown	True
Unknown	True	Unknown	True	Unknown
Unknown	False	False	Unknown	Unknown
Unknown	Unknown	Unknown	Unknown	Unknown

Endi relatsion butunlik chegaralarini o'rganishga o'tamiz.

### Mohiyatlar butunligi

Berilganlarning butunligini ta'minlashdagi birinchi qoida, asosiy munosabatlardagi birlamchi kalitlarga tegishli.

Bu erda *asos munosabat* deganda, konseptual chizmadagi birorta mohiyatning nomiga mos keluvchi va kortejlari fizik jihatdan berilganlar baza-sida saqlanuvchi munosabat tushuniladi. Mohiyatlar butunligi qoidasiga asosan, *asos munosabatdagi birorta birlamchi kalit atributi NULL sifatidagi qiymatsiz atributga ega bo'lishi mumkin emas*. Qoidaga asosan, birlamchi kalit – bu minimal identifikator bo'lib, u kortejlarni identifikasiyalash uchun noyob tarzda ishlataladi. Bu esa, birlamchi kalitning hech bir qism to'plami kortejlarni identifikatsiyalash uchun yetarli bo'lishi mumkin emasligini bildiradi. Agarda, birlamchi kalitning ixtiyoriy qismida

NULL kelishi mumkin deb faraz qilsak, bu kortejlarni noyob tarzda identifikatsiyalash uchun hamma atributlari qatnashishi shart emas, degan tasdiqqa olib keladi. Bu esa, birlamchi kalitning ta'rifiga zid. Masalan, RO'YHAT munosabatidagi Zn atributi birlamchi kalit bo'lgani uchun, RO'YHAT munosabatidagi Zn atributiga NULL ga ega kortejni kiritish mumkin emas. *Xuddi shunday asos munosabatidagi tarkibli birlamchi kalit atributlarining hech biriga NULL ga ega kortejni kiritish mumkin emas.*

Agarda biz bu qoidalarga diqqat bilan e'tibor bersak, uning bir qancha g'ayri oddiy xususiyatlarini ko'ramiz. Birinchidan, nimaga bu qoida birlamchi kalitga nisbatan qo'llaniladi-yu, kortejlarni xuddi shunday bir qiymat aniqlovchi nomzod kalitlarga nisbatan qo'llanilmaydi? Ikkinchidan, nimaga NULL ga ega kortejni faqat asos munosabatlarga kiritish mumkin emas? Masalan, quyidagi 13.1-rasmida keltirilgan ANIQLASH munosabati-dagi berilganlardan foydalangan holda, quyidagi so'rovni ko'raylik:

#### 11. «Kuzatish natijalariga nisbatan hamma izohlar tanlansin».

Obekt	Ijarachi	Aniqlash sanasi	Izoh
Tosh0757	IDK1201	29.11.2008	жавон
Tosh1075	IDK0107	06.09.2005	кичик
Tosh0052	IDK0177	16.05.2011	
Tosh1707	IDK5978	17.04.2009	кент ва ёрут
Tosh1075	IDK0177	30.10.2007	
Tosh1707	IDK5248	25.09.2008	тальмидашларга
Tosh0757	IDK0177	08.06.2008	
Tosh0052	IDK1116	10.10.2010	боянид шинни

#### 13.1-rasm.

Ushbu so'rovning natijasida bitta Izoh atributiga ega bo'lgan, munosabat hosil bo'ladi (13.2-rasmi). Ta'rifga binoan, bu atribut birlamchi kalit bo'lishi kerak, lekin uning qiymatlari orasida NULL ga ega kortejlar bor. Bu kortejlar Tosh0052, Tosh0757 va Tosh1075 obyektlarini ijarrachisi IJ0177 ko'rsatmasiga asosan Izoh atributiga kiritilgan. Bu munosabat asos munosabat bo'lmagan uchun, relatsion model uning birlamchi kalitida NULL turishiga ruxsat beradi.

Обект
БИОГАЗ
БИЗНЕС
КЕНТ ВА ЁРУТ
ТАЛЬМИДЛАНМАГАРА
БОЯНИД ШИННИ

#### 13.2-rasm.

Berilganlar bazasidagi munosabatlar albatta bir-biri bilan bog'langanini  
biz o'quv loyihamizda ham ko'rib o'tdik.  
Ro'yhat (ZN, FAMILIYA, ISM, O\_ISM, K\_SANA, FAK\_N,  
YON\_N, KURS, GURUH\_N, PASP\_S, PASP\_N)  
Baholar (ZN, SEMESTR, FAN\_N, BALL, SANA\_KIR, USTOZ)  
Fakultellar (FAK\_N, FAKULTET)  
Yo'nalish (YON\_N, YO'NALISHLAR)  
Fanlar (FAN\_N, FAN\_NOMI).

Bunday bog'lanishlar relatson berilganlar bazasida tashqi kalitlar  
orqali amalga oshiriladi. Bu munosabatlardagi tashqi kalitlar:

- Ro'yhat munosabatida FAK\_N va YON\_N;
- Baholar munosabatida ZN va FAN\_N.

Bu erda Ro'yhat va Fanlar munosabatlari Baholar munosabati uchun  
*ajdod*, Baholar esa *avlod* munosabat hisoblanadi. Fakultellar va Yo'nalish  
munosabatlari Ro'yhat uchun *ajdod*, u esa *avlod* munosabat hisoblanadi.

#### Mulohazalar:

- Tashqi kalit, nomzod kalit kabi oddiy va tarkibili bo'lishi mumkin.
- Tashqi kalit ajdod munosabatdagi birlamchi kalit aniqlangan domenlarda aniqlangan bo'lishi kerak.
- Tashqi kalit odatda, noyoblik xususiyatiga ega bo'lmaydi. Haqiqatan ham avlod munosabatda, ajdod munosabatdagi bitta kortejga murojaat qiluvchi, bir nechta kortej bo'lishi mumkin. Bu esa, albatta munosabatlar orasidagi «birdan-ko'pga» aloqasini beradi.
- Agarda tashqi kalit haqiqatan ham noyoblik xususiyatiga ega bo'lsa, munosabatlari orasidagi aloqa «birlikdan-birlikka» bo'ladi.
- Tashqi kalitning qaysidir nomzod kalitning tarkibiy qismi bo'lishi talab qilinmaydi.
- Tashqi kalitlarda NULL bo'lishi, faqatgina uning atributlari hech qaysi nomzod kalitning tarkibida bo'lgagan holatlarda, ruxsat etiladi.

### Semantik butunlik

*Semantik butunlik*, satrga kiritilgan berilganlar elementi shu satr uchun  
mumkin bo'lgan qiymatga ega ekanligini kafolatlaydi. Bu – kiritilgan qiymat  
domenga yoki shu ustundagi qiymatlar to'plamining mumkin bo'lgan  
elementiga tegishli bo'lishi kerak, degan ma'noni anglatadi. Masalan,  
Talabalar o'quv loyihamizdag'i BAHOLAR munosabatiga tegishli Ball  
nomli ustundagi qiymatlar faqat butun sonlar bo'lishi mumkin. Berilgan  
larning semantik butunligi, domendagi qiymatdan tashqaridagi qiymatlar  
kiritilsa, buziladi.

Semantik butunlik asosan quyidagi uchta cheklashlar usullari orqali amalga oshiriladi:

- *Berilganlar turi.* Berilganlar turi, berilganlarning qaysi qiymatlarini ustunlarda saqlash mumkinligini aniqlaydi. Masalan, berilganlarning SMALL INT turi ustunlarning qiymatlari sifatida -32767 dan 32767 gacha bo'lgan qiymatlarni kiritish mumkinligini anglatadi.
- *Oshkormas qiymat.* Oshkormas qiymat ustunga kiritish uchun oshkor qiymati tafsirlanmagan vaziyatdagi qiymatni bildiradi. Masalan, Foydalanuvchi\_Jadvali nomli jadvaldagi Foydalanuvchi\_ID ustuning qiymati, foydalanuvchining tizimga kirishi uchun zarur bo'lgan oshkormas qiymatdan iborat.
- *Chegarani tekshirish.* Chegarani tekshirishda ustundagi har bir qiymat shartni bajarish nuqtayi nazaridan tekshiriladi. Har bir qiymat berilgan shartni bajarsagina senantik butunlik saqfanadi. Masalan, BAHOLAR jadvali Bali ustunidagi qiymatlar [0,100] bo'lishi shart.

### Havola butunligi

Berilganlar butunligidagi uchinchi qoidasidagi cheklashi, tashqi kalitlarga bog'liq. *Havola butunligida*, agarda munosabatda tashqi kalit mavjud bo'lsa, tashqi kalitning qiymati asos munosabatdagi qaysidir kortejdagi nomzod k- kalit qiymatiga mos kelishi yoki tashqi kalit to'liq NULL lardan iborat bo'lishi kerak, degan shartning bajarilishi ta'minlanadi.

Masalan, Talabalar o'quv loyihamizdagi BAHOLAR munosabatiga tegishli Zn atributi tashqi kalit bo'lib, RO'YHAT asos munosabatdagi Zn atributiga havola qiladi. Tizim BAHOLAR munosabatiga, RO'YHAT asos munosabatda, reyting daftaring raqami **B000600** bo'lgan talaba xususidagi yozuv paydo bo'limguncha, **B000600** aniqlaydigan kortejga shu talabaga tegishli ma'lumotlarni kiritishga bo'lgan har qanday harakatni taqiqlashi kerak.

Yana bir misol, RO'YHAT munosabatda Fak\_n atributi tashqi atribut bo'lib, FAKULTETLAR asos munosabatidagi Fak\_n atributiga havola qiladi. Agarda, reyting daftaring raqami **B000600** bo'lgan talaba kirgan fakultet FAKULTETLAR asos munosabatida ko'rsatilmagan bo'lsa, tizim ma'lumotlarni kiritishga yana ruxsat bermaydi.

### Butunlikning korporativ chegaralari

*Korporativ chegara* deganda, foydalanuvchi yoki berilganlar bazasining administratori aniqlaydigan berilganlar butunligini qo'llab-quvvatlash bo'yicha qo'shimcha qoidalari tushuniladi.

Foydalanuvchilarning o'zları berilganiarga qo'shimcha chegaralarni o'matishi mumkin. Masalan, bir guruvida 20 tadan ortiq talaba bo'lishi mumkin bo'lmasa, buni foydalanuvchi qoida sifatida kiritishi, tizim esa bu qoidaning bajarilishini nazorat qilishi kerak. Shuni e'tirof etish kerakki, turli BBB tizimlarda relatsion butunlikni saqlash darajasi turlicha amalga oshirilgan.

### Tasavvur

1- bobda «Uch bosqichli» ANSI-SPARC arxitekturasini ko'rganinizda, tashqi taqdimot alohida foydalanuvchining tasavvuridagi berilganlar bazasining tuzilishi sifatida tavsiflangan edi. Relatsion modelda «taqdimot» (view) boshqacharoq ma'noni, yani tasavvurni bildiradi. U foydalanuvchining hamma tashqi modelini emas, balki u ishlataligani virtual yoki hosilaviy munosabatni bildiradi, ya'ni munosabat aslida mavjud bo'lmaydi, lekin u dinamik tarzda bitta yoki bir nechta berilganlar bazasidagi mavjud munosabatlar asosida hosil qilinadi. Shunday qilib, tashqi model bir paytda bir nechta konseptual darajasidagi asos munosabatlardan va shu munosabatlar asosida yaratilgan taqdimotlardan ham tuzilishi mumkin. Biz bu erda relatsion tizimlarning elementlaridan biri bo'lgan, ana shunday virtual munosabatlarni yoki *tasavvurlarni* qarab chiqamiz.

Shu paytgacha biz o'rgangan munosabatlarimizi, *asos munosabatlar* deb atadik. *Tasavvur* tushunchasini ana shu asos munosabatlar zaminida aniqlaymiz.

Qaysidir R munosabatni olish maqsadida, asos munosabatlar ustida bajarilgan bitta yoki bir nechta relatsion amallarning dinamik natijasini, *tasavvur* deb ataymiz. *Tasavvur* virtual munosabat bo'lib, u berilganlar bazasida aniq tarzda mavjud bo'lmaydi, lekin alohida foydalanuvchining talabi bilan o'sha daqiqada yaratiladi.

Foydalanuvchi nuqtayi nazeridan, *tasavvur* bu munosabatdan iborat bo'lib, u doimiy ravishda berilganlar bazasida mavjud bo'ladi va u bilan xuddi asos munosabat bilan ishlagandek ishlash mumkin. Lekin yuqorida aytiganidek, *tasavvur* har doim ham berilganlar bazasida asos munosabat kabi saqlanmaydi. Tasavvurning ichidagilar, bir yoki bir nechta asos munosabatlarga yo'naltirilgan so'rovlar natijasida aniqlanadi. Tasavvurlar ustidagi ixtiyoriy amallar, o'zini yaratilishida asos bo'lgan, munosabatlar ustidagi amallarga avtomatik tarzda aylantiriladi. Tasavvurlar dinamik xususiyatga ega bo'lib, asos munosabatdagi o'zgarishlar tasavvurdagi tegishli berilganlarni o'zgarishiga olib keladi. Agarda foydalanuvchilar tasavvurga ba'zi bir mumkin bo'lgan o'zgarishlarni kiritsalar, bu o'zgarishlar shu zahoti tegishli asos tasavvurlarga ham kiritiladi.

## Tasavvurlarning tayinlanishi

**Tasavvurlar mexanizmidan foydalanishning bir nechta sababları mavjud:**

- tasavvurlar qudratli va o'zgaruvchan himoya mexanizmiga ega bo'lib, berilganlar bazasining ayrim qismlarini ma'lum bir foydalanuvchilardan yashirish imkoniyatini beradi. Foydalanuvchi o'zi uchun kirish huquqi berilmagan tasavvurlarning qaysidir atributlar yoki kortejlar mavjudligi xususida hech qanday ma'lumot ololmaydi;
- tasavvurlar berilganlarga foydalanuvchilar kirishi uchun ushga eng qulay imkoniyatlar yaratadi, shuning uchun bir xildagi berilganlar bir paytning o'zida turli foydalanuvchilar uchun turlicha qaralishi mumkin;
- tasavvurlar asos munosabatlar ustida bajariladigan murakkab amallarni sodda lashtirish imkonini beradi. Masalan, tasavvur ikkita asos munosabat negizida aniqlangan bo'lsa, foydalanuvchi uning ustida oddiy unar amallar tanlash va nusxalashni bajarishi mumkin. Bu amallar BBB tizimining vositalari orqali avtomatik ravishda asos munosabatlarni birlashtiradi va ekvivalent amallarni bajaradi.

Tasavvurlarni loyihalashda, foydalanuvchi uchun eng qulay bo'lgan, tashqi modelning qo'llab-quvvatlash usulini tanlashi kerak.

Quyida ushbu yondashuv bo'yicha amaliyatda qo'llash bo'yicha misollar keltirilgan.

1. RO'YHAT munosabatidagi yozuvlar bilan ishlashda, foydalanuvchilar yozuvlardagi turli atributlarni tanlashdan tashqari, nomlarni va mos ravishda talabalar xususidagi boshqa ma'lumotlarni tanlashi mumkin. Shu kabi tasavvur RO'YHAT va BAHOLAR munosabatlarini birlashtirgan holda, uni nusxasidan yuqori ball olganlar tanlanadi.
2. Ko'pchilik foydalanuvchilar RO'YHAT munosabati yozuvlar bilan ishlaganda talabalarning Pasp\_S (pasport seriyasi) va Pasp\_N (pasport raqami) atributlariga kirmasligi kerak.
3. Atributlar qayta nomlanishi mumkin. Jadvallardagi atributlarning nomlarini foydalanuvchilar o'zlariga moslab, qayta yangi nomlar berishlari mumkin.
4. Talabalar o'quv loyihasidagi har bir bazaga tegishli foydalanuvchilarning kirishi uchun huquq berish mumkin.

Keltirilgan hamma misollar, berilganlarga nisbatan aniq bir mantiqiy bog'liqsizlik darajasi mavjudligini ko'rsatadi. Haqiqatan ham tasavvurlar berilganlarga nisbatan mantiqiy bog'liqsizlikni, konseptual chizmani qayta

taskilllashtirish natijasida foydalanuvchilarni himoyasini, bundan ham muhim darajasiga erishishga yordam beradi. Masalan, munosabatga yangi atribut qo'shilsa, bu atribut tasavvurda ko'rsatilmaguncha foydalanuvchilar uning mayjudligini shubba ham qilmaydilar. Agarda muvjud munosabat qayta tuzilsa yoki yangi munosabatlarga ajratilsa, uni ishlatuvchi tasavvur shunday qayta aniqlanadiki, foydalanuvchi berilganlar bilan eski formatlarda ishlash imkoniyatini yo'qotmaydi.

### Tasavvurlarni yangilash

Asos munosabatlardagi berilganlarni yangilash ular bilan bog'liq bo'lган hamma tasavvurlarda bir zumda akslanishi kerak. Aksincha, tasavvur-dagi berilganlar o'zgartirilsa, asos munosabatlarda darholda akslanishi kerak. Tasavvurlar yordamida bajariladigan o'zgartirishlarga ma'lum bir chegaralar o'matiladi. Quyida, ko'pgina mavjud bo'lgan BBB tizimlaridagi tasavvurlar orqali berilganlarni o'zgartirish mumkin bo'lgan shartlar keltirilgan:

- yakka holdagi asos munosabat negizida oddiy so'rov orqali aniqlangan va shu asos munosabatning birlamchi yoki nomizod kalitini o'zida saqlaydigan tasavvur orqali yangilash mumkin;
- bir nechta asos munosabatlar negizida yaratilgan tasavvurlarda, berilganlarni yangilash mumkin emas;
- ustida aggregatlash va guruhash amallari bajariladigan ixtiyoriy tasavvurlarda berilganlarni yangilash mumkin emas.

Relatsion tasavvurlar ushbü sinflarga ajratilgan – *nazariy jihatdan yangilanmaydigan, nazariy jihaidan yangilanadigan va qisman yangilanadigan*.

### Relatsion berilganlar bazasida havola butunligini saqlash shartlari

Tashqi kalitlarning birlamchi kalitlarga mos kelish qoidasi havola butunligi shartiga rioya qilishning asosiy qoidasi hisoblanadi. Tashqi kalitning har bir qiymati uchun ajdod munosabatda birlamchi kalitning mos qiymati mavjud bo'lishi kerak.

Havola butunligi kiritish (qo'shish), yangilash va yozuvlarni olib tashlash amallari natijasida buzilishi mumkin. Havola butunligini aniqlashda ikkita munosabat ishtirok etadi – ajdod va avlod. Ularni har biri keltirilgan amallarni bajarishi mumkin, shuning uchun ham quyidagi oltita variant havola butunligini buzilmasligiga yoki buzilishiga olib kelishi mumkin.

*Ajdod munosabat uchun:*

*Kiritish amali.* Birlamchi kalitning yangi qiymati hosil bo'ldi. Ajdod munosabatda avlod munosabatdan havolasi bo'limgan yozuvlarning mavjud bo'lishi mumkin. Bu amal havola butunligini buzmaydi.

*Yangilash amali.* Yozuvdag'i birlamchi kalitni qiymatini yangilash havola butunligining buzilishiga sababchi bo'lishi mumkin.

*Olib tashlash (yo'qotish) amali.* Yozuv olib tashlanganda birlamchi kalitning qiymati ham yo'qotiladi. Agarda avlod munosabatda qiymati olib tashlangan kalitga havola qiluvchi yozuv bo'lsa, u holda tashqi kalitfarning qiymati aniqlanmay qoladi. Yo'qotish amalining bajarilishi havola butunligini buzilishiga olib kelishi mumkin.

#### *Avlod munosabat uchun:*

*Kiritish amali.* Agarda yangi yozuv uchun tashqi kalitning qiymati aniqlanmagan bo'lsa, bu yozuvni avlod munosabatga kiritib bo'lmaydi. Kiritish amalining bajarilishi havola butunligini buzilishiga olib kelishi mumkin.

*Yangilash amali.* Avlod munosabatdag'i yozuvni yangilash paytida tashqi kalitning qiymatini bexosdan o'zgartirib yuborish mumkin. Yangilash amalining bajarilishi havola butunligini buzilishiga olib kelishi mumkin.

*Olib tashlash (yo'qotish) amali.* Avlod munosabatdag'i yozuv olib tashlansa, havola butunligi buzilmaydi.

Shunday qilib, havola butunligi quyidagi to'rtta amaldan bittasining bajarilishi natijasida buzilishi mumkin:

- ajdod munosabatdag'i yozuvlar yangilanganda;
- ajdod munosabatdag'i yozuvlar olib tashlanganda;
- avlod munosabatga yozuvlar kiritilganda;
- avlod munosabatdag'i yozuvlar yangilanganda.

### **Havola butunligini qo'llab-quvvatlashdagi asosiy strategiyalar**

Havola butunligini qo'llab-quvvatlashda asosiy ikkita strategiya mavjud.

**RESTRICT (cheklash)** – havola butunligini buzishga olib keluvchi amallarning bajarilishiga yo'l qo'ymaslik.

**CASCADE (kaskadli o'zgartirish)** – talab qilingan amalni bajarishga ruxsat berish, lekin bir-biri bilan bog'langan munosabatlarda havola butunligini buzilishiga yo'l qo'ymaslik uchun bog'langan munosabatlarda o'zgartirishlar kiritilganda havola butunligi va hamma aloqalar saqlanishi kerak. O'zgartirishlar ajdod munosabatdan boshlanib, ketma-ket ravishda avlod munosabatlarda bajariladi. Bu strategiyaning bajarilishida bitta nozik tomoni bor, avlod munosabatlarning o'zi ba'zi bir boshqa munosabatlari

uchun ajdod sifatida kelishi mumkin. Bu holatda qaysidir boshqa strategiya uchun ushu aloqani saqlash uchun bajarilishi talab qilinishi mumkin va givaning ushu aloqani saqlash uchun bajarilishi talab qilinishi mumkin va h.k. Bunday vaziyatda, ixtiyoriy darajadagi kaskad amallarni bajarish imkonini bo'lmasa, u holda birlamchi bajarilgan amallarni bekor qilib, berilganlar bazasini boshlang'ich holatini tiklash zarur. Bu murakkab strategiya hisoblanadi, lekin u ajdod va avlod munosabatlar orasidagi aloqalarni buzmaydi.

Bu strategiyalar standart bo'lgani uchun, havola butunligini saqlovchi hamma BBB tizimlariga kiritilgan.

### **Havola butunligini qo'llab-quvvatlashdagi qo'shimcha strategiyalar**

Havola butunligini qo'llab-quvvatlashdagi qo'shimcha strategiyalar quyidagilardan iborat.

*IGNORE (inkor etish)* – havola butunligini tekshirmsadan amalni bajarishga ruxsat etish. Bu holatda avlod munosabatda tashqi kalitlaming aniq bo'limagan qiymatlari paydo bo'lishi mumkin, butunlikni saqlashdagi hamma javobgarlik programmachi yoki foydalanuvchi zimmasiga yuklandi.

*SET NULL (NULL berish)* – talab qilingan amalning bajarilishiga ruxsat etiladi, lekin tashqi kalitning hamma noaniq qiymatlari NULL bilan o'zgartiriladi. Bu strategiyaning ikkita kamchiligi bor. Birinchidan, NULL dan foydalanishi uchun ruxsat kerak. Ikkinchidan, avlod munosabatdagi yozuvlarning ajdod munosabatdagi yozuvlar bilan aloqasi yo'qoladi. Amal bajarilgandan keyin, qaysi ajdod munosabatdagi yozuv bilan avlod munosabatdagi o'zgargan yozuvlar aloqa bo'lganligini aniqlab bo'lmaydi.

*SET DEFAULT (oshkormas ravishda qiymat berish)* – talab qilingan amalning bajarilishiga ruxsat etiladi, lekin tashqi kalitning hamma noaniq qiymatlarini oshkormas ravishda qabul qilingan qiymatga o'zgartiriladi. Ushbu strategiyaning foydali tomoni, NULL dan foydalanishga ruxsat bermasligi. Bunda ham, amal bajarilgandan keyin, qaysi ajdod munosabatdagi yozuv bilan avlod munosabatdagi o'zgargan yozuvlar aloqa bo'lganligini aniqlab bo'lmaydi.

### **Xulosa**

*Butunlik chegaralari* – berilganlarga qo'yiladigan mantiqiy chegaralardan iborat. BBB tizimlari, berilganlar bazasini bir holatdan boshqa holatga o'tganda, berilganlarning qo'yilgan chegaralarga mosligini nazarat qildi. Berilganlarning chegaralarini kiritish, predmet sohani to'g'ri akslantirish bilan ham bog'liq.

*Ayrim holatlarda, berilganlar bo'sh qiymatga ega bo'lib, qolishi mumkin. Ana shunday vaziyatda NULL bilan ishlash uchun zarur bo'lgan uch qiymatli mantiqdan foydalaniladi.*

*Berilganlarning relatsion butunlik chegaralari mohiyatlar butunligi, semantik butunlik va havola butunligi bilan bog'liq.*

*Havola butuntigida asosiy va qo'shimcha stragiyalar mavjud bo'lib, berilganlarni butunligini saqlashga yordam beradi.*

### Nazorat uchun savollar

1. Butunlik chegarasi deganda nimani tushunasiz?
2. Oshkor va ichki chegaralarni ta'riflang.
3. Domen chegarasi nimani aniqlaydi?
4. Berilganlarning relatsion modelidagi butunlikning asosiy uchta qoidarining asosida nimalar yotadi?
5. Berilganlarning relatsion modelidagi butunlikning birinchi qoidasining ta'riflarini keltiring.
6. Berilganlarning relatsion modelidagi butunlikning ikkinchi qoidasining ta'riflarini keltiring.
7. Berilganlarning relatsion modelidagi butunlikning uchinchi qoidasining ta'riflarini keltiring.
8. Berilganlarning relatsion modelidagi butunlikning uchta qoidasidan foydalanish zarurligini tushuntirib bering.
9. Bo'sh qiymat deganda nimani tushunasiz. Misollar keltiring.
10. Asos munosabatni ta'riflang va boshqa munosabatlardan farqini keltiring.
11. Qaysi holatda munosabatning birlamchi kaliti NULL ga ega bo'lishi mumkin?
12. Semantik butunlikni saqlashdagi «berilganlar turi» usuli, deganda nimani tushunasiz?
13. Semantik butunlikni saqlashdagi «oshkormas berilgan» usuli, deganda nimani tushunasiz?
14. Semantik butunlikni saqlashdagi «chegarani tekshirish» usuli, deganda nimani tushunasiz?
15. Havola butunligining shartini keltiring.
16. Korporativ chegara nimani bildiradi?
17. Relatsion modeldagi tasavvurning ma'nosini ochib bering.
18. Relatsion modellarda tasavvurdan foydalanishning asosiy sabablarini izohlab bering.
19. Relatsion modellardagi tasavvurlarning sinflarini aytib bering.

- 20
20. Relatsion modellardagi qaysi amallar natijasida havola butunligi buzilishi mumkin?
  21. Ajod munosabatga nisbatan amallarni bajarish natijasidagi havola butunligining holatlarini aytib bering.
  22. Avlod munosabatga nisbatan amallarni bajarish natijasidagi havola butunligining holatlarini aytib bering.
  23. Havola butunligining buzilishi mumkin bo'lgan holatlar qaysi?
  24. Havola butunligini qo'llab-quvvatlashda asosiy strategiyalarni aytib bering.
  25. Havola butunligini qo'llab-quvvatlashdagi qo'shimcha strategiyalar nimalardan iborat?

---

## **IV BO'LIM. RELATSION BERILGANLAR BAZASINI LOYIHALASH USLUBIYATI**

### **14-bob. Axborot tizimlarida berilganlar bazasi**

**Tayanch iboralar:** *axborot tizimi, konseptual model, fizik model, ichki model, tashqi model, loyihalash uslublari, IDEF oitasi, Case vostitalari, predmet soha, loyihalash uslubiyati, tashqi model, qism chizma.*

#### **Axborotlar tizimini loyihalash negizlari**

*Axborot tizimi* (AT) – programm-apparat majmua bo'lib, ma'lumotlarni saqlash va ma'lum bir sohaga taalluqli axborotlarni qayta ishlash uchun mo'ljallangan. BB – ixtiyoriy axborotlashgan tizimning eng muhim komponentasi hisoblanadi. Yaxshi yaratilgan berilganlar bazasi nafaqat muammosiz uni ishlatish imkonini yaratadi, balki unda kundalik xizmatlarni bajarish, uni mukammallashtirish, korxonaning modernizatsiyalash jarayonida, axborot oqimlari, qonunchilik va hisobot shakllari o'zgarganda, takomillashtirish imkonini yaratadi.

Berilganlar bazasining tuzilishi loyihalash jarayonida aniqlanadi. Berilganlar bazasini loyihalash jarayoni juda murakkab bo'lishi mumkin. Birorta tashkilotning axborotga bo'lgan talablarni qondirish uchun, alohida foydalanuvchilarining ehtiyojlarini qondiruvchi oddiy fayl tizimlaridan iborat ilovalarni yaratishga nisbatan, umuman boshqacha yondashuvni talab qildi. BB asosidagi tizimlarni yaratish uchun birinchi navbatda berilganlar, ana undan keyin ilovalar xususida o'ylash mumkin. Bunday yondoshuvlarining almashishini loyihalash tamoyillarining almashishi deb baholanishi mumkin. Tizim, foydalanuvchilarining so'rovlariga to'liq javob berishi uchun, berilganlar bazasini loyihalash jarayoniga juda katta e'tibor bilan munosabatda bo'lish kerak. Noto'g'ri loyihalangan BB xatoliklar tug'diradi, natijada noto'g'ri qarorlar qabul qilinadi, bu esa tashkilotni qiyin vaziyatga olib kelishi mumkin. Boshqa tomonдан, yaxshi loyihalangan BB bilan ishlovlchi tizimga aniq ma'lumotlar chiqarib berilsa, to'g'ri va samarali qarorlarni qabul qilishga zamin bo'ladi.

Afsuski, hozirgi kundagi berilganlar bazalarini loyihalash uslublari keng tarqalmagan. Ko'pchilik tashkilotlar yoki alohida programma

mahsulotlarini yaratuvchilar, juda kam holatlarda birorta uslubiyatga tayanadilar. Axborot tizimlarini yaratishdag'i ana shu kamchilik, ko'pgina muvaffaqiyatsizliklarga olib kelmoqda.

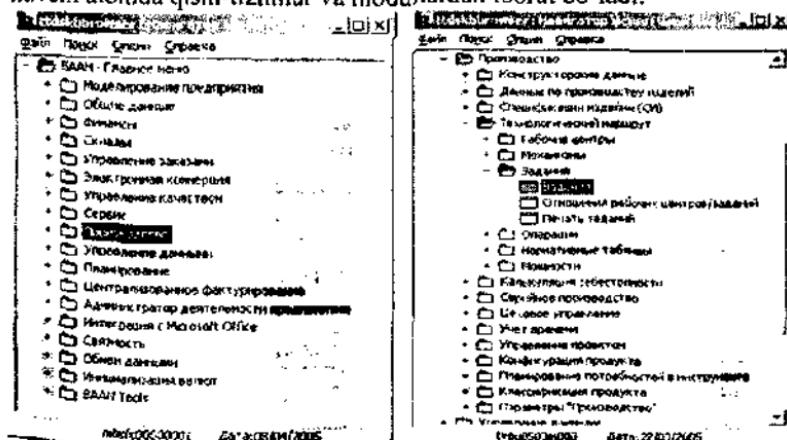
Berilganlar bazasini loyihalashdagi tuzilmali yondashuvlarning yo'qligi, yaratishga ketadigan vaqt va vositalarga yetarlichcha e'tibor bermaslikka, yaratilgan BB sanurasiz yoki ilovalarning talablariga javob bermaydigan holatlarga olib kelmoqda. Programma mahsulotining tavsisi va uni amalga oshirish bo'yicha ko'sratmalar yozilgan hujjatlar yetarlichcha yaxshi emasligi, berilganlar bazalarini kuzatishdag'i katta qiyinchiliklarni tug'dirmoqda.

Shu bilan birga, ayrim yiwik tashkilotlarda majmuaviy axborot tizimlari yaratilgan bo'lib, (korporativ tizimlar, ERP-tizimlar), ular barcha servislarni vizuallashirishga xizmat qiladi. Ularga misol qilib, SAP firmasining R/3, Oracle E – Bisness Suite, BaanERP larni keltirish mumkin. Rossiyada yaratilgan «Galaktika», «Flagman» va «Parus» tizimlari o'zini funksionallik darajasi bo'yicha ERP sinfigagi tizimlarga yaqinlashadi.

IDC analistik kompaniyasining 2008-yildagi ma'lumotlariga ko'ra, Rossiya bozorida korxonalarini boshqaruvchi integrallashgan tizimlar hajmi so'nggi 3 yilda 30,4% oshgan va 486 mln. AQSH dollariga yetgan.

Ko'pgina ERP tizimlarni har xil operatsion tizimlar va berilganlar bazasi o'matilgan serverlarga yuklash (ko'p platformali shaklli tizimlar) mumkin. Shu kabi tizimlardagi BB bir necha ming jadvallardan iborat (BaanERP 5.0c – bitta korxona bo'yicha 2500 tadan ortiq jadval mavjud).

Har qanday murakkab tizimning ishonchlligini ta'minlash uchun, u iyerarxik tizim shaklida yaratiladi. U umumiy BBdan, o'zaro harakatlanuvchi alohida qism-tizimlar va modullardan iborat bo'ladi.



14. I-rasm. BaanERP tizimining qism-tizimlari va modullari.

14.1-rasmida 5.0c naqlidagi BaanERP tizimining to'liq tarkibi (tizim administratorning menyusi) hamda «Ishlab chiqarish» tizimining qism-tizimlardan iborat modular tarkibi ko'rsatilgan.

Programmalarни yaratishni tushunish, tashkilotga taalluqli tizimni ishlab chiqish, ma'lumotlarni saqlash va ularni qayta ishlov berishni bilish har bir zamonaviy mutaxassis uchun zarur hisoblanadi.

14.2-rasmda «Flagman» Korporativ ATning ichki qism-tizimlari va modullari keltirilgan.

#### MOLIYA-IQTISODIY BOSHQARMA

Marketing	Fermis nijozdny rejaksib	Shartnomalar va o'zaro hisoblar	Moliyaviy menedzhment
-----------	-----------------------------	------------------------------------	--------------------------

#### LOGISTIKA

Savdo-satiq	Ombor hisobi	Ta'minot
-------------	--------------	----------

#### Ishlab-chiqarishni boshqarish

Ishlab-chiqarishni tehnik tayyorlash	Etiyalatlari boshqarish	Kalendar reja	Setlamlari boshqarish
Gizmatlar va servis		Tarmoq qazanchilar	

#### HODIMLARNI BOSHQARISH

Xodimlar	Moosh	Sohiqalar
----------	-------	-----------

#### HISOB-KITOB VA TAHLIL

Hisob-kitob	Moliyaviy taxil
-------------	--------------------

#### NAZORAT

Xarakatlarni boshqarish	Biznes monitoringi
-------------------------	--------------------

#### AXBOROT ZAHIRALARINI BOSHQARISH

Administrasiyalash	Xujjalay aylansib	Instrumental vestular
--------------------	-------------------	-----------------------

14.2-rasm. «Flagman» Korporativ ATning ichki qism-tizimlari va modullari.

Axborot tizimini tavsiflaganda, u ikki xil ma'noni anglatadi, deb taxmin qilinadi – operatsion ma'no bu qandaydir qayta ishlashni amalga oshiradi (programmaga o'xhash) va past darajali ma'no bu ma'lumotlarni BBda saqlagan holda, ularga o'zgartirishlar, izlash va o'qishtirish imkoniyatini beradi.

Murakkab axborot tizimlarni hosil qilishda «dekompozitsiya» usuli qo'llaniladi. Bunda tizim bir-biri bilan bog'langan ichki qismalarga bo'linadi va bu qismular iyerarxik tuzilishni tashqil etadi. Murakkab tizimlarning iyerarxik atributi guruhi lab yaratish negizi bilan yaxshi kelishadi. Bunday holda har bir loyiha ishtirokchilarining faoliyati iyerarxiyadagi o'miga qarab chegaralanadi.

Murakkab tizimlarni yaratishning eng qulay va klassik usuli bu tuzilishli loyihalash bo'lib, bunda tizimni «yuqorida pastga» usuli bo'yicha algoritmiq «dekompozitsiya» qilinadi. Xuddi shu holatda umunniy BBga ega bo'lgan yaxshi faoliyat yurituvchi tizimni qurish mumkin. Foydalanishda, kelishilgan formatlar bilan ishlash axborotni qayta ishlash tizimining barcha bo'limlarida qulaylik tug'diradi hamda hamma qism-tizimlar o'zaro optimal ishlaydi.

Ba'zi tizimlarni yaratish tarixan «pastdan-yuqoriga» usuliga asoslangan bo'lib, bunda avval alohida avtomatizatsiyalashgan ishchi o'rinnari yaratilgan, keyinchalik ularni yagona axborot tizimiga birlashtirishga harakat qilingan. Bunday usul yirik tizimlarni yaratishda muvaffaqiyatlbo'imasligi mumkin.

Axborot tizimini va uning loyihasini yaratishda quyidagilarni aniqlash lozim:

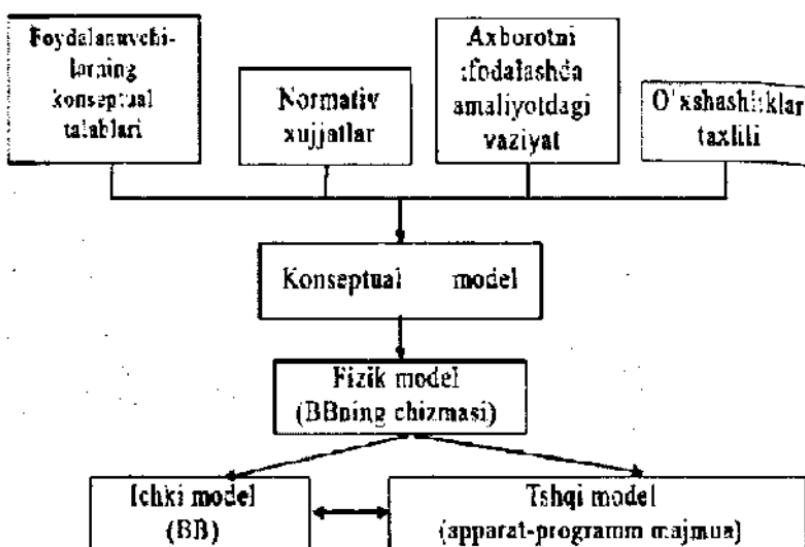
- axborot tizimining obyektlarini (konseptual modeldag'i mohiyatlarni);
- axborot tizimining atributlarini;
- obyektlarning o'zaro aloqalari, ulardag'i ichki va o'zaro axborot oqimlarining harakatini.

Bunda amaliyotda mavjud bo'lgan axborot jarayonlarini va normativ axborotlarni (qonunlarni, hukumat qarorlarini, sohadagi standartlarni) amalga oshirishda axborot hajmini aniqlash, saqlanadigan axborot formatlarini tanlash va ularni uzatish juda mubim ahamiyat kasb etadi. Agarda axborot jarayonining qayta qurishda tub o'zgarishlar bo'lmasa, u holda bu axborotni saqlash formatlarini va ularni qayta ishlashdagi jurnal, vedomost, jadval yoki boshqa biror-bir qog'ozda saqlash usullaridan ham foydalanish tavsiya etiladi.

Lekin avvaldan, axborotlarning yangi hisob tizimlariga o'tishi, saqlanishi va ishlov berilishi masalalarini bozorda mavjud bo'lgan yirik axborot kompaniyalari tomonidan yaratilgan o'xhash programma mahsulotla-

rimi, qo'yilgan masalalarga qisman yoki to'liq mos kelishi nuqtayi nazaridan tahlil etilishi lozim.

Axborot modelini shakllantirish chizmasi 14.3-rasmda keltirilgan.



14.3-rasm. Axborot modelining chizmasi.

*Konseptual model* (14.3-rasmga qarang) – axborot obyektlarini, ularning atributlari, ular orasidagi aloqani ko'rsatadi, lekin axborotlarni saqlashning fizik usullarini ko'rsatmaydi (predmet sohaning modeli, ba'zida uni mantiqiy axborot modeli deb ataladi). Axborot obyekti odatda mohiyat deb atalib, alohida obyektlar yoki voqealardan iborat bo'lib, ular haqidagi ma'lumotlarni saqlash tafab qilinadi. Mohiyatlar aniq atributlarning majmuasidan tuziladi.

*Fizik model* – axborot obyektlarining bazalaridagi hamma atributlarni (atributlarni) va ular orasidagi aloqalarni, bazalarni saqlash usulini (qaysi BBBT foydalanishni hisobga olgan holda) akslantiradi.

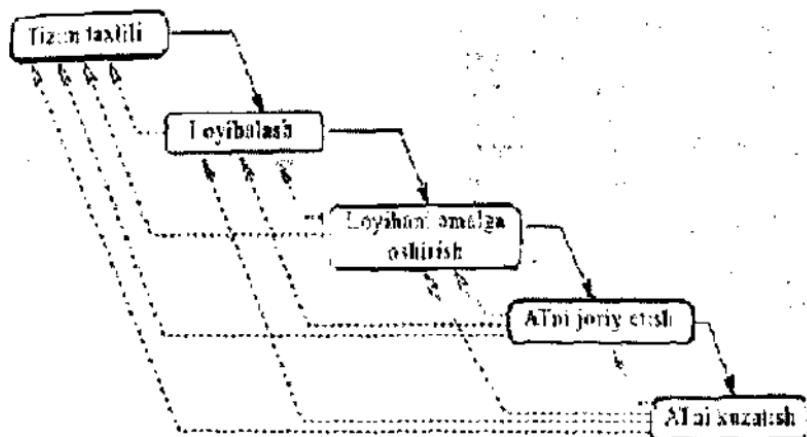
*Ichki model* – aniq fizik modelga mos keluvchi berilganlar bazasidir.

*Tashqi model* – yaratish, saqlash, tahrirlash, ma'lumotlarni o'chirish va izlash jarayonlarini ta'minlovchi hamda yechiladigan masalaning hal qilishdagi zaruriy hisoblashlar va chop qilish uchun hosil qilingan shakllardan iborat, berilganlar bazasi bilan ishlash uchun zaruriy apparat va programmlar to'plami.

Axborot tizimlarini yaratish bir nechta bosqichda olib boriladi, ularning har birida yaratilayotgan tizimning elementlari aniqlashtiriladi.

Axborot tizimini yaratishdagi yashash davrini tasvirlovchi turli usullar mavjud. Teskari aloqaga ega pog'ona usuli 14.4-rasmida keltirilgan.

Murakkab tizimning hayotiy siklini ishlab chiqish bunday holda tahlil, loyihalash, programmalash va testdan o'tkazish, joriy qilish va kuzatish bosqichlaridan iborat bo'lib, ular ketma-ket bajariladi.



14.4-rasm. ATning pog'onali yashash davrining chizmasi

Bugungi kundagi qabul qilingan normalarda AT loyihalarining ustida quyidagilar ishlashadi:

- predmet sohaga taalluqli tijorat-jarayonni o'rGANUVCHI va ulami modellashtiruvchi tijoratchi-tahlilchilar;
- yechimlarni ko'rishni, masalalar, ilovalar va ma'lumotlarni loyihashtiruvchi tizimli tahlilchi va ma'murlar;
- kodlar ilovasining mualliflari;
- sinovdan o'tkazish va sifatiga baho beruvchi mutaxassislar;
- hujjatlash mualliflari;
- distributivlar mualliflari;
- joriy qilish bo'yicha mutaxassislar.

Odatda, bu funksiyalar har xil soha mataxassislari orasida taqsimlanadi, lekin amaliyatda ba'zida bitta mutaxassis bir nechta vazifani bajarishi mumkin.

AT obyektlarini yaratishdagi loyihalash va programmalash bosqichlarida obyektga yo'naltirilgan usuldan foydalanish mumkin (vorislik, inkapsulatsiya, polimorfizm).

### Loyihalash uslublari

Murakkab tizimlarni loyihalashdagi masalalarni yechishda maxsus uslubiyat va standartlar mavjud.

Bu standartlarga IDEF (Item DEFinition, ICAM – Integrated Computer-Aided Manufacturing – XX asrning 70- yillar boshida AQSH Harbiy havo kuchlari tomonidan birinchi bo'lib ishlab chiqilgan, ishlab chiqarishni integrallashgan kompyuterlar orqali qo'llab-quvvatlash) oilasiga mansub uslublar kiradi. Ular yordamida loyihalashlarni samarali bajarish, akslantirish va murakkab tizimlarni modellarining har xil bo'limlарini tahlil qilish mumkin. Hozirgi kunda IDEF oilasiga mansub standartlarga quyidagilar kiradi:

1. IDEF0 - Function Modeling – murakkab tizimlarning funksional modellashtirish uslubiyati. IDEF0 dagi grafik til yordamida, o'rgani layotgan tizim uni yaratuvchilar va analitiklarga bir-biri bilan bog'langan funksionallar bloklar majmuasi sifatida tasvirlanadi. U SofTech, Inc. Kompaniyasining o'tgan asrning 60-yillarida yaratilgan SADT – tuzilishli tahlil va yaratish texnologiyasi asosida qurilgan. Odatda, IDEF0 yordamida modellashtirish har qanday tizimni o'rganishning birinchi bosqichi hisoblanadi.

2. IDEF1 - Information Modeling – tizim ichidagi axborot oqimini modellashtirish uslubiyati. U axborot oqimlarini akslantirish hamda ularning tuzilishlari va o'zaro aloqalarini tahlil qilish imkoniyatini beradi.

3. IDEFIX (IDEF1 Extended) - Data Modeling – relatsion turdag'i berilganlar bazasini loyihalash uslubiyati. «Mohiyat-aloha» (ERD - Yentity-Relationship Diagram) turdag'i berilganlar modeli yaratiladi.

4. IDEF2 - Simulation Model Design - tizimlarni dinamik modellash uslubiyati. Hozirgi paytda, «bo'yagan Petri tarmoqlariga» asoslangan, IDEF0 dagi statistik diagrammalar to'plamini dinamik modellarga aylantiruvchi algoritm va programmalar mavjud.

5. IDEF3 - Process Description Capture – tizimdag'i jarayonlarni hujsjatlashtirish uslubiyati, masalan korxonalardagi texnologik jarayonlarni tadqiqot qilishda. IDEF3 yordamida har bir jarayonning ssenariysi va bajariladigan amallar ketma-ketligi tasvirlanadi. IDEF3 IDEF0 uslubiyati bilan o'zaro bog'langan, ya'ni har bir funksiya (funksional blok) IDEF3 vositalari yordamida alohida jarayon tarzida tasvirlanishi mumkin.

6. IDEF4 - Object-Oriented Design – obyektga yo'naltirilgan tizimlar yaratish uslubiyati. IDEF4 – vositalari, obyektlar tuzilishi va ularning yaratish uslubiyati. IDEF4 – vositalari, obyektlar tuzilishi va ularning yaratish uslubiyati. IDEF4 – vositalari, obyektlar tuzilishi va ularning yaratish uslubiyati. IDEF4 – vositalari, obyektlar tuzilishi va ularning yaratish uslubiyati. IDEF4 – vositalari, obyektlar tuzilishi va ularning yaratish uslubiyati. IDEF4 – vositalari, obyektlar tuzilishi va ularning yaratish uslubiyati. IDEF4 – vositalari, obyektlar tuzilishi va ularning yaratish uslubiyati.

7. IDEF5 - Ontology Description Capture – murakkab tizimlarni ontologik tadqiqot qilish uslubiyati. IDEF5 uslubiyatida tizim ontologiyasi, kerakli atama va qoidalarning lug'atlari yordamida, ma'lum vaqt oraliq'ida ko'rilayotgan tizim holatining ishonchli tasdiqlarini olish mumkin. Shu tasdiqlar asosida tizimning keyingi rivojlanishi xususidagi xulosalar olinadi va u optimallashtiriladi.

8. IDEF6 - Design Rationale Capture – loyihalashda ratsional tajribadan foydalanish uslubiyati. AT loyihalashda, yangi loyihalarni yaratishda tuzilishga bog'liq xatoliklarni oldini olish uchun, ratsional tajribalarni saqlash uchun yaratilgan uslubiyat.

9. IDEF7 - Information System Auditing – axborot tizimining audit uslubiyati.

10. IDEF8 - User Interface Modeling – foydalanuvchilar aloqasini hosil qilish uslubiyati.

11. IDEF9 - Scenario-Driven IS Design – mavjud shartlar va cheklashlarni tahlil qilish, jumladan fizik, huquqiy, politexnik va ularning reininginiring jarayonida qaror qabul qilishga ta'siri.

12. IDEF10 - Implementation Architecture Modeling – bajarish arxitekturasini modellashtirish.

13. IDEF11 - Information Artifact Modeling – artefaktlarni axborot modellashtirish.

14. IDEF12 - Organization Modeling – tashkiliy modellashtirish.

15. IDEF13 - Three Schema Mapping Design – kartalarning uch chizmali dizayni.

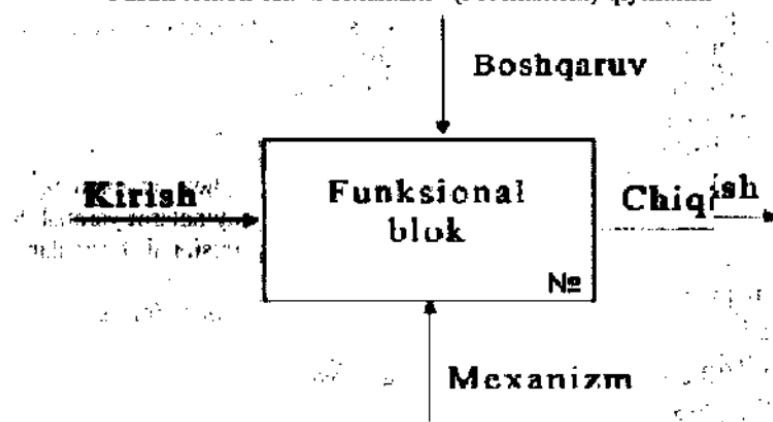
18. IDEF14 - Network Design - kompyuter tarmoqlarini modellashtirish uslubiyati. Hisoblash tarmoqlarini loyihalashda berilganlarning tarkibi, navbatlar, tarmoq tuzuvchilari, ishonchlik talablari va h.k. grafik tilda tasvirlash hamda tahlil qilish imkoniyatini yaratadi.

IDEF0 - IDEF5, IDEF9 standartlarining tafsifini saytlarda [33,35,36] ko'rish mumkin. IDEF0 - IDEF1X standartlari ATni tashkil etuvchilarini tasvirlash, ular orasidagi aloqalar va ATda berilganlar modelini qurish usullarini tavsiflaydi.

IDEF0 standartida funksionallar blokining grafik tasvirini to'g'ri to'rtburchak shaklida ko'rishimiz mumkin (14.5-rasmiga qarang). Standart shartlariga ko'ra har bir funksional blokning nomi moyillikka undovchi fe'l shaklida bo'lishi lozim (masalan, «amalni bajarmoq» emas, balki «amal

bajarilsin» kabi). Funksional blokning 4ta tomonining har biri o‘zining aniq qiymatiga ega, ya’ni:

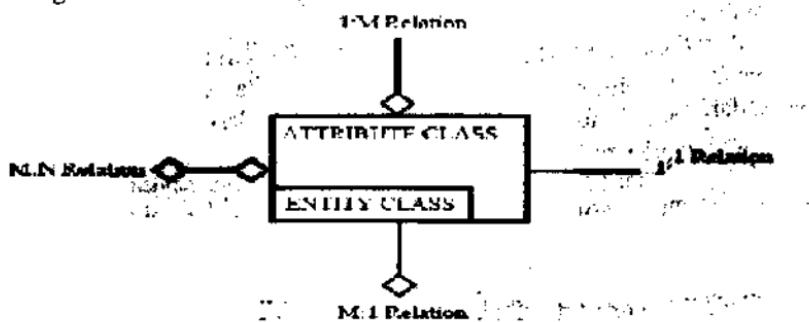
- Yuqori tomon «Boshqaruv» (Control) qiymatini oladi;
- Chap tomon «Kirish» (Input) qiymatini oladi;
- O‘ng tomon «Chiqish» (Output) qiymatini oladi;
- Pastki tomon esa «Mexanizm» (Mechanism) qiymatini oladi.



14.5-rasm. IDEF0 standartida funksional blok.

IDEF0 standartida funksional modellarni tasvirlash u’suli model-lashtirishning instrumental vositalaridan foydalanadi (CASE-vositalar - Computer-Aided Software System Engineering), masalan, Computer Associates firmasining Bpwin vositalari va h.k.

IDEF1 standartida konseptual modelning mohiyat va aloqalari 14.6-rasmidagi kabi tasvirlanadi.



14.6-rasm. IDEF1da konseptual modelning aloqa va mohiyat tasyvirlanishi.

IDEF1 usuli tahlil usuli hisoblanib, quyidagi larni tavsiflaydi:

- tashqilotlarda ma'lumotlar to'plash, saqlash va ishlav berishni;
- axborotlarni boshqarishning qoidalari va mantiqini;
- axborotni yaxshi boshqarilmasligidan kelib chiqadigan muammo-larni.

Murakkab tizimlarni modellashtirishda foydalanadigan boshqa usullar:

- DFD - «berilganlar oqimi» tahlil texnologiyasi (Data Flow Diagrams);
- Workflow – «ishlar oqimi» tahlil texnologiyasi.

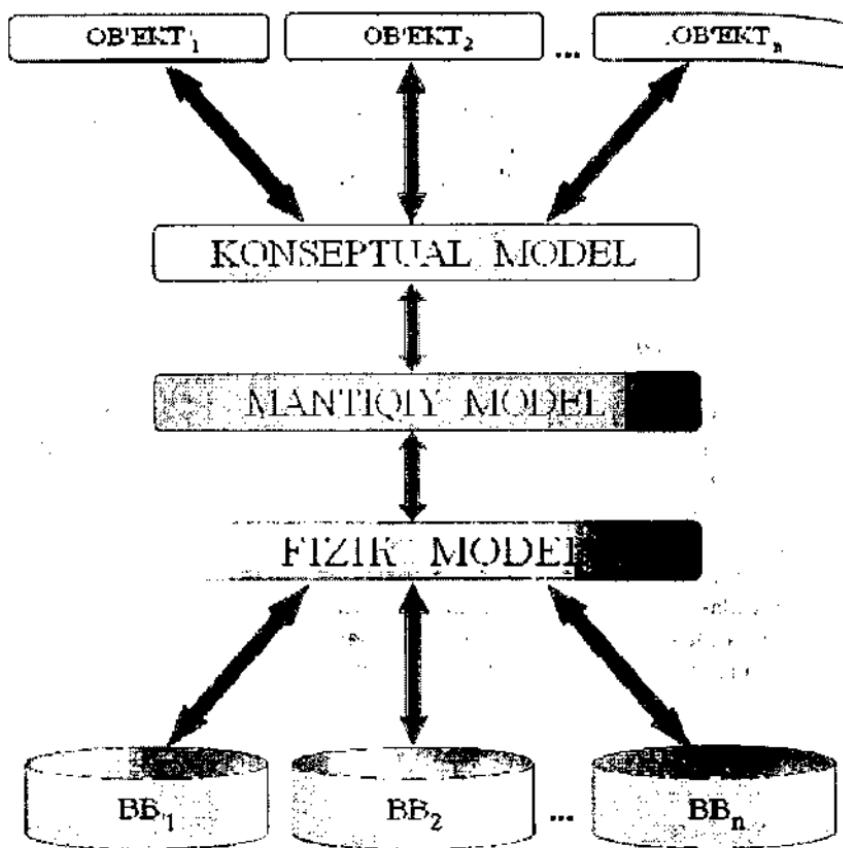
### Predmet sohani modellashtirish darajalari

Berilganlar bazasi aniq bir *predmet sohani*, lozim topilgan tadqiqot uchun zarur bo'lgan, haqiqiy dunyodagi bir qism (bir yoki bir nechta obyektlar)ni akslantriradi.

Berilganlar bazalarining yashash (mavjudlik) davridagi asosiy bosqichlardan biri, bu berilganlar bazalarini loyihalash bo'lib, undagi aniq uslubiyatlardan biri relatsion berilganlar bazalarini loyihalashdan iborat.

**Loyihalash uslubiyati** – bu aniq tuzilishga ega bo'lgan va maxsus protseduralar, texnik usullar, vositalar va hujjatlardan foydalanishni hisobga olgan hamda loyihalash jarayonini qo'llab-quvvatlash va soddalashtirishga yo'naltirilgan jarayondir. Loyihalash uslubiyati yaxlit jarayonni bir qancha davrlarga bo'lgan bo'lishni, ularning har biri esa bir qancha bosqichlardan iborat bo'lishini nazarda tutadi. Har bir bosqichda yaratuvchiga, ishlab chiqishning shu davriga bog'liq bo'lgan masalalarini yechish uchun texnik usullar majmuasi taklif qilinadi. Bundan tashqari loyihalash uslubiyati rejalash usullari, muvofiqlashtirish, boshqarish, loyihami amalga oshirish bosqichini baholash hamda berilganlar bazalariga qo'yiladigan talablar majmuasini tahlil qilish va modellashtirish uchun tuzilishli yondashuv usulini taklif qiladi, bu harakatlarni standart va tashkillashgan tartibda amalga oshirishga imkon yaratadi. Ushbu uslubiyat berilganlar bazalarini loyihalashdagi uchta bosqichni o'z ichiga oladi – konseptual, mantiqiy va fizik bosqichlar (14.7-rasm).

**Konseptual loyihalash** – eng muhim mohiyatlar turlarining tavsifi, ular orasidagi aloqalar va atributlarni o'z ichiga olgan berilganlar bazasi (obyektlar)ning konseptual tasavvurini yaratishdan iborat. Berilganlar bazasini konseptual loyihalash korxonaning berilganlarini konseptual modelini yaratishdan boshlanadi. Bu jarayonda konseptual model amalga oshirish bosqichlaridagi hech bir qism yoki bo'laklarga bog'lanmagan holda yaratiladi.



14.7-rasm.

*Mantiqiy loyihalash* – konseptual tasavvurni berilganlar bazalarining mantiqiy tuzilishlariga aylantirish, shu jumladan munosabatlarni loyihalash. Mantiqiy loyihalashdagi berilganlar bazasining tuzilishi, *chizma* deb ataladi.

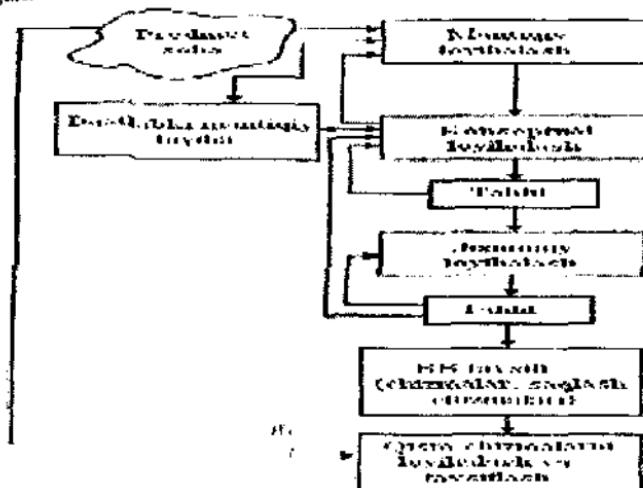
*Fizik loyihalash* – tanlangan BBB tizimidan foydalangan holda, mantiqiy modelni jadvallar yordamida fizik jihatdan berilganlar bazalariga akslantirish bo'yicha qaror qabul qilishdan iborat. Bunda kompyuter tashuvchilarida berilganlarni joylashtirish usullari aniqlanadi, berilganlarni taqsimlash, ularga kirish va indekslash usullari hisobga olinadi. Zamonaviy amaliy programma vositalarida tashkillashtirishning bu bosqichi foydaluvchining ishtiokisiz avtomatik tarzda amalga oshiriladi. Foydalanuvchi, amaliy programmalar va universal programma vositalar orqali ishlaganida,

**BBB** tizimlaridagi tushunchalarga asoslangan holda ish olib boradi. Shu nuqtayi nazaridan, loyihalashdagi eng asosiy bosqichlar bu konseptual va mantiqiy modellarni yaratisidan iborat bo'ladi.

Turli foydalanuvchilarga mantiqiy modelning turli qismi to'plamlaridan foydalanishga to'g'ri keladi. Aniq bir foydalanuvchi nuqtayi nazaridan berilganlar bazasini mantiqiy tuzilishini tavsiflovchi modelni, *tashqi model*, uning tavsifi esa *qism chizma* deyiladi. Foydalanuvchining tashqi modeli, ana shu foydalanuvchining konseptual talablarini akslantirishdan iborat.

Foydalanuvchi faqatgina qism chizmadagi berilganlarga kirishi mumkin. Bu esa ma'lum darajada berilganlarning himoyasini ta'minlaydi. Har bir qism chizma doirasida berilganlarga ishlovnning mumkin bo'lgan holatlari beriladi va bu ham qo'shimcha himoyani ta'minlaydi. Qism chizmalaridan foydalanish foydalanuvchining ishini ancha yengillashtiradi, chunki u faqatgina o'ziga tegishli berilganlar bazasining tuzilishini bilishi yetarli bo'ladi.

Loyihalash bosqichlari orasidagi o'zaro bog'lanishlar 14.8-chizmada keltirilgan.

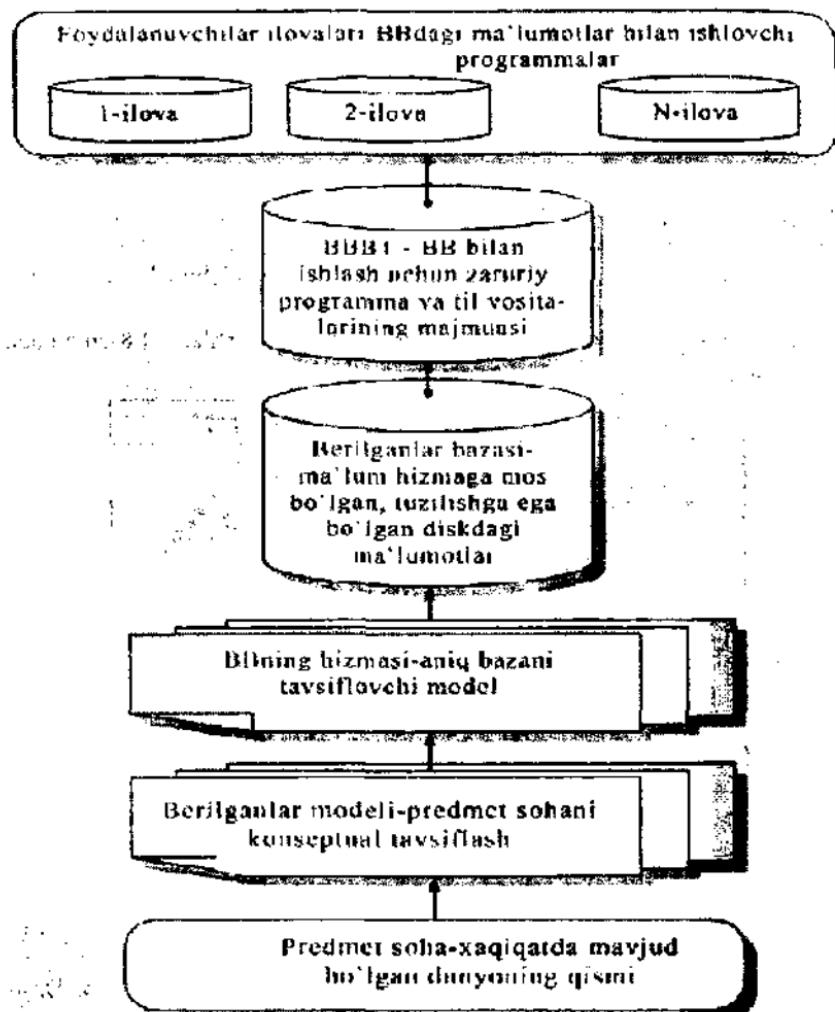


14.8-rasm.

Loyihalash jarayoni konseptual modelni qurishdan boshlanadi. Dastlabki model loyiha oldi bosqichida quriladi va loyihalashning keyingi bosqichlarida yana aniqlashtiriladi. Keyin uning asosida mantiqiy model quriladi. Fizik va tashqi modellar bulardan keyin ichtiyoriy ketma-ketlikda qurilishi mumkin, shu jumladan parallel ravishda ham. Umumiyl qurish

jarayoni davomida avvalgi bosqichlarga qaytib loyihalash jarayonlarini o'zgartirish yoki modelga aniqliktar kiritish mumkin.

14.9-rasmda BB loyihalashdagi asosiy tushunchalarining o'zaro bog'liqligi va ular bilan ishlash chizmasi keltirilgan.



14.9-rasm. BB loyihalashdagi asosiy tushunchalarining o'zaro harakati va ular bilan ishlash.

## Xulosa

Axborot tizimi – bu programma-apparat majmua bo'lib, ma'lumotlarni saqlash va ma'lum bir sohaga taalluqli axborotlarni qayta ishlash uchun mo'ljallangan. BB – ixtiyoriy axborot tizimining eng muhim tarkibiy qismi sifatidu qaraladi.

Shu nuqtayi hazardan, axborot tizimlarini loyihalash masalalari berilganlar bazasini loyihalashdagi bosqichlar bilan umumiy kesishish nuqtalariga ega. Bu kesishish nuqtalari konseptual, mantiqiy va fizik loyihalashdagi bosqichlarga borib taqaladi.

Axborot tizimlarini modellashtirishda keng foydalaniladigan xalqaro standart IDEF bo'lib, u juda ko'p Case vositalar tarkibiga kiritilgan. IDEF oиласига mansub standartlar 16 xildan iborat bo'lib, turli vazifalarni bajarishga yo'naltirilgan.

## Nazorat uchun savollar

1. Axborot tizimini ta'riflang.
2. Berilganlar bazasi va axborot tizimi orasidagi bog'lanishlarni izohlang.
3. Berilganlar bazasini loyihalash jarayonini izohlang.
4. Berilganlar bazasini loyihalashdagi kamchiliklar qanday oqibatlarga olib keladi?
5. Axborot tizimlariga misol keltiring.
6. Murakkab axborot tizimlari qaysi usul bilan yaratiladi?
7. Axborot tizimini yaratishda aniqlanishi zaruriy bo'lgan narsalar nima?
8. Axborot modelining chizmasini izohlang.
9. Konseptual va fizik modellarni izohlang.
10. Ichki va tashqi modellarni izohlang.
11. Murakkab tizimlarni loyihalashda qanday uslubiyat va standartlar ishlatalidi?
12. IDEF oиласига mansub nechta standart mavjud?
13. IDEF oиласига mansub standartlardan ixtiyoriy uchtasini aytib bering.
14. 14.5-rasmidagi IDEF0 standartini izohlang.
15. 14.6-rasmidagi IDEF1 standartini izohlang.
16. Predmet soha, deganda nimani tushunasiz?
17. Konseptual loyihalashni izohlang.
18. Mantiqiy loyihalashni izohlang.
19. Fizik loyihalashni izohlang.
20. 14.8-rasmni izohlang.
21. 14.9-rasmni izohlang.

## **15-bob. Berilganlar bazasini konseptual loyihalash**

**Tayanch iboralar:** berilganlarning lokal konseptual modeli, mohiyatlarning turlari, aloqalarning turlari, attributlarni aniqlash, attributlarning domenlarini aniqlash, modelni ortiqchaligini tekshirish, «mohiyat-aloga» diagrammalari, atribut nomi, berilganlar turi va qiymatining formati, attributning hamma taxalluslari, attributning guruhli yoki guruhstizligi, attributning hosilavivligi, attributning ko'p qiymatiligi, tranzaksiya.

### **Berilganlarning lokal konseptual modelini yaratish**

Biz ushbu bobda berilganlar bazasini bosqichma-bosqich konseptual loyihalash usuli bilan tanishib chiqamiz. Birinchi bosqichda korxonadagi turli xildagi foydalanuvchilarning berilganlar xususidagi tasavvuri asosidagi lokal konseptual modelni yaratamiz (15.1-rasm). Berilganlarning lokal konseptual modelni deb atalishining sababi, korxonaning predmet sohasini qamrab oluvchi, berilganlarning har bir tasavvuri uchun konseptual model yaratiladi. Tahsil jarayoni natijasida, foydalanuvchilaming har birining o'z nuqtayi nazari jihatidan tasavvurdagi berilganlarni umumlashtirish imkoniyatlarini hisobga olib, yagona identifikatorni belgilash mumkin bo'lgan holat izlanadi. Berilganlarga qo'yilgan talablarni toplash va ularni tahsil qilish natijasida turli tasavvurlar aniqlanadi. Masalan, Talabalar o'quv loyihamizda quyidagi tasavvurlardan foydalanish mumkin:

- o'quv ishlari bo'yicha har bir fakultetdagi dekan muovinlarining nuqtayi nazaridan kelib chiqqan holdagi RO'YHAT va BAHOLAR tasavvuri;
- oliy o'quv yurti o'quv ishlari bo'yicha prorektorning nuqtayi nazaridan kelib chiqqan holatda FAKULTETLAR va YO'NALISH tasavvuri;
- o'qitiladigan fanlar bo'yicha javobgar kafedralarning mudirlari nuqtayi nazaridan ketib chiqqan holatda FANLAR tasavvuri.

RO'YHAT tasavvuri uchun lokal konseptual modelni yaratish misolida bajariladigan ishlarni ko'rib chiqamiz. Berilganlarning har bir lokal konseptual modeli quyidagi tarkibdan iborat:

- mohiyatlarning turlari;
- aloqalarning turlari;

- atributlar va atributlarning domenlari;
- birlamchi kalitlar;
- muqobil kalitlar;
- butunlik chegarasi.

Berilganlarning konseptual modeli shu modelni yaratish jarayonida hujjatlar va berilganlar lug'ati bilan to'ldiriladi. Har bir bosqichda zarur bo'ladigan kuzatuv hujjatlari shu bosqichlarni tavsiflashda keltirilishi mumkin. Berilganlarning konseptual modelini yaratishning birinchi bosqichida quyidagi vazifalar bajarilishi kerak:

- mohiyatlarning turlarini aniqlash;
- aloqalarning turlarini aniqlash;
- atributlarni aniqlash va ularni mohiyatlarning turlari va aloqalar bilan bog'lash;
- atributlarning domenlarini aniqlash;
- nomzod va birlamchi kalit bo'ladigan atributlarni aniqlash;
- modelni ortiqchalik nuqtayi nazaridan tekshirish;
- lokal konseptual modelni aniq foydalanuvchining tranzaksiyalariga mosligini tekshirish;
- berilganlarning lokal konseptual modelini oxirgi foydalanuvchilar bilan muhokama qilish.

### **Mohiyatlarning turlarini aniqlash**

Mohiyatlarning turlarini aniqlashdan maqsad, aniq tasavvur uchun zarur mohiyatlarni belgilashdan iborat. Berilganlarning lokal konseptual modelidagi birinchi bosqichda foydalanuvchi uchun qiziqish tug'dirishi mumkin bo'lgan obyektlarni aniqlash kerak. Mohiyatlarni identifikatsiyalash usullaridan bittasi, surishtirilayotgan foydalanuvchining shu korxonadagi aniq bajaradigan vazifalar majmuasining tafsirini o'rganishdan iborat. Ana shu tafsirlardan hamma «ot» yoki «ot» bilan «sifat» birlashmalarini ajratib olish zarur (masalan, RO'YHAT tasavvuri uchun «reyting daftarchasi», «familiyasi», «ismi», «otasining ismi», «kirkagan sana», «fakultet nomi», «yo'nalish nomi», «kurs», «guruh», «pasport seriyasi», «pasport nomeri»). Keyin bu obyektlar ichidan eng muhimlari tanlanadi («fakultet nomi», «yo'nalish nomi») yoki muhim konsepsiylar tanlanadi va boshqa obyektlarni aniqlovchi «ot»lar olib tashlanadi. Masalan, «fakultet nomi» va «yo'nalish nomi» xususiyatlari alohida mohiyatlarni (FAKULTETLAR va YO'NALISH) tashkil qilgani va bu mohiyat bilan esa, FAKULTETLAR va YO'NALISH mohiyatlariiga mos ravishda kiritilgan «fakultet raqami» va «yo'nalish raqami» xususiyatlari orqali bog'langani ma'qil.

Mohiyatlarni muqobil identifikatsiyalash usulida, boshqa obyektlarga bog'liqsiz ravishda mavjud bo'lgan obyektlar izlanadi. Masalan, FANLAR yoki BAHOLAR obyektlari alohida mohiyatlarni tashkil etadi.

Ayrim hollarda, mohiyatlarni ajratish tafsirlarning qoniqarsiz ravishda taqdim etilishi natijasida qiyinlashadi. Foydalanuvchilar o'z fikrlarini bayon qilish jarayonida ko'pincha bir ma'noli bo'lmagan ta'riflami, misollarni va o'xshashliklarni ishlatalardilar. Misol uchun, umumlashtirilgan so'z o'miga bir yoki bir nechta faktlarni yoki ismlarni keltiradilar. Ba'zida xodimlarning familiyasini keltirish yoki korxonaning nomini keltirish o'miga uning bajaradigan ishlari yoki ko'rsatadigan xizmat turlari bilan chalqashtiradilar. Bunday vaziyatlarda ular xodimning lavozimi yoki u bajaradigan funksiyalarni keltiradilar (masalan, «rektor», «ustoz» yoki «dekan»).

Loyihalash jarayoni, foydalanuvchilar sinonim yoki omonimlarni ko'p ishlatishi natijasida ham murakkablashadi. Sinonim deganda, biz ma'nosi o'xshash yoki bir xilda bo'lgan so'zlarni, masalan «bo'llim» va «filial», tushunamiz. Omonim deganda esa, yozilishi bir xilda bo'lgan, lekin turlicha ma'noni anglatuvchi so'zlarni tushunamiz, masalan «o't». Bunday so'zlarni qaysi ma'noni kasb etishini biz faqatgina berilgan matnga nisbatan aniqlashimiz mumkin.

Har doim ham obyekt mohiyatmi, aloqami yoki atributmi, degan savolga aniq javob topish qiyin. Tahlil ham subyektiv jarayon bo'lgani uchun, ayrim lokal konseptual modellarni yaratuvchilar ham bitta faktga turli nuqtayi nazaridan qarashlari mumkin. Shuning uchun variantni tanlash yaratuvchining mantiqi va tajribasiga bog'liq. Berilganlar bazalarini yaratuvchilar predmet sohani, yaratilayotgan korxonadagi nuqtayi nazarlar, ularda ishlataladigan faktlar asosida, cheklagan holatda yaratishlari lozim. Shuning uchun ayrim holatlarda tafsirlarga qo'yilgan talablardan bir ma'noli mohiyatlarning noyob majmuasini aniqlab bo'lmaydi. Lekin, tahlilni loyiha tafsirlari majmuasini ketma-ket ravishda bir necha bor bajarish natijasida, albatta tizimga qo'yiladigan talablarni qondiruvchi mohiyatlarning butun majmuasi aniqlanadi.

### **Mohiyatlar turlarini hujjatlashtirish**

Har bir mohiyatlar ajratilgandan keyin unga ma'lum bir ma'noga ega bo'lgan nom berish va bu nom albatta foydalanuvchilarga tushunarli bo'lishi kerak. Tamlangan nom va mohiyatning tavsisi berilganlar lug'atiga joylashtiriladi. Agarda mohiyat foydalanuvchilarga turli nomlar bilan ma'lum bo'lsa, bu qo'shimcha nomlarni ham sinonim sifatida berilganlar lug'atiga kiritish kerak. 15.1-rasmda berilganlar lug'atining bo'lagi keltirilgan bo'lib, unda Talabalar o'quv loyihamizdag'i RO'YHAT, BAHOLAR, FA-

## KULTETLAR va YO'NALISH taşavvurlaridagi mohiyatlar tavsifi keltirilgan.

Mohiyat nomi	Tavsifi	Taxal-jus	Mohiyat nushalari turgan joy
RO'YHAT	OO'YUDagi barcha talabalar ning ro'yxati	Ankeba	Har bir talaba aniq bitta fakultetda o'qiydi
BAHOLAR	OO'YUDagi barcha talabalar ning olgan baholari	Baho	Har bir fan bo'yicha talabalar baboga ega
FAKULTETLAR	OO'YUDagi barcha fakultetlar	Fak	Har bir fakultet o'z talabalariga ega
YO'NALISH	OO'YUDagi barcha yo'nalishlar	Yo'n	Har bir yo'nalishda talabalar o'qiydi

15.1-rasm.

### Aloqalarning turlarini aniqlash

Aloqa turlarini aniqlashdan maqsad, mohiyatlar orasidagi eng muhim aloqa turlarini topishdan iborat. Yuqorida keltirganimizdek, mohiyatlarni aniqlashdagi usullardan biri foydalanuvchi talablaridagi tafsirlardan hamma «ot»larni tanlash edi. Bunda aloqalarni aniqlash uchun talablardagi tafsirlarni grammatik tahlil qilish kerak bo'ladi. Mavjud aloqalarni aniqlashda ham shunga o'xshash yondashuv talab qilinadi, lekin bu holatda «kesim»ga ega bo'lgan hamma ifodalar tanlanadi.

Masalan:

- RO'YHAT (talaba ta'lim oladi);
- BAHOLAR (talaba reyting ballarini oladi);
- FAKULTETLAR (talaba aniq bitta fakultetda o'qiydi);
- YO'NALISH (talaba bitta yo'nalish bo'yicha o'qiydi).

Tafsirlarning matni aloqalar bo'yicha ba'zi ma'lumotlarni saqlashi, korxona uchun ushbu aloqalar juda muhimligidan dalolat beradi. Shuning uchun bu aloqalar albatta yaratilayotgan modelda akslanishi kerak.

Loyihachilarni esa, mohiyatlar orasidagi loyiha talablarini qondiruvchi aloqalar qiziqtiradi. RO'YHAT mohiyati bilan FAKULTETLAR va YO'NALISH mohiyatlari orasidagi aloqalarning mavjudligini ko'rgan edik. Talaba har bir fan bo'yicha reyting ballarini oladi. Shuning uchun RO'YHAT mohiyati va BAHOLAR mohiyati orasida ham aloqalar bo'lishi zarur. Lekin fanlarning nomini alohida saqlashimiz mumkin. Shu nuqtayi nazardan, FANLAR mohiyatini yaratib, BAHOLAR mohiyati bilan aloqani «reyting daftari raqami» atributi orqali o'matishimiz mumkin.

Ko'pgina hofattarda aloqafar ikki tomonfama o'matishadi, ya'ni faqatgina ikkita mohiyatlar orasida. Lekin alohida e'tibor bilan murakkab

aloqalarning ham mavjudligini, ya'ni ikkitadan ortiq turli turdag'i mohiyatlar orasidagi aloqalarni hamda rekursiv aloqalarni (bir turdag'i mohiyatlar orasidagi aloqalarni) tekshirish zarur. Eeng muhimma hamma aloqalar aniqlanganini tekshirish zarur. Lekin yuzlab mohiyatlarga ega bo'lgan yirik loyihalarda bunday tekshirish uncha samara bermaydi. Lekin tekshirmaslik yomon oqibatlarga olib kelishi mumkin. Shuni ta'kidlash kerakki, u yoki bu qolib ketgan aloqalar keyinchalik, foydalanuvchi uchun zarur bo'lgan tranzaksiyalarni bajarish imkoniyatlarini tekshirish paytida, albatta aniqlanadi.

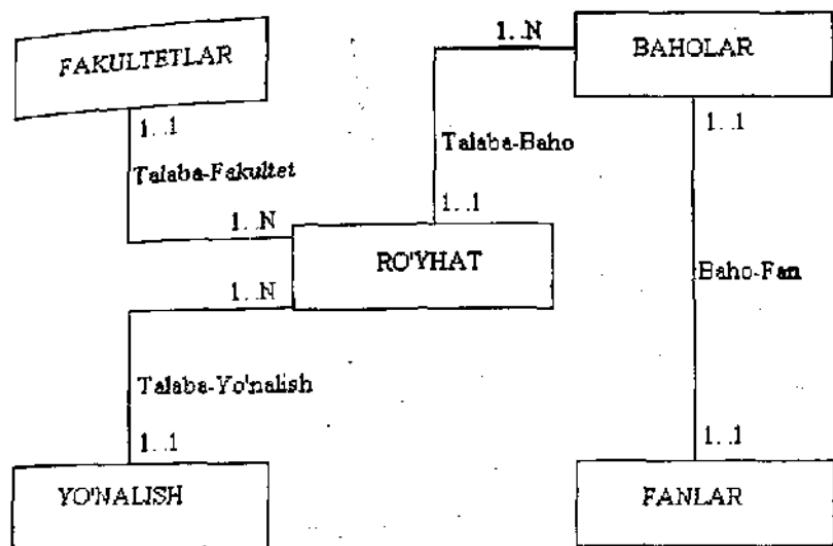
### «Mohiyat-aloqa» diagrammalaridan foydalanish

Agarda loyihalash jarayonini chizmalar orqali kuzatish imkoniyati tug'dirilsa, ish ancha yengillashadi. Mohiyatlar va ular orasidagi aloqalarni kuzatish uchun odatda, «mohiyat-aloqa» diagrammalaridan foydalilanadi. Bu hamma vaqt, ko'z ostida korxonaning modellashtirilayotgan qismining chizmasi turishini ta'minlaydi. Bunda modellashtirishning universal tili, UMLdan ham foydalanish ham yaxshi samara beradi.

Yaratilgan modelda aloqalar o'rnatilgandan keyin, ularning har birining kardinalligini aniqlash kerak. Agarda ularning aniq qiymatlari bo'lsa yoki yuqori yo quiyi chegaralari ma'lum bo'lsa, ularni albatta hujjatlashtirish kerak. Kardinallik chegaralari berilganlarning sifati va uni ta'minlashga xizmat qiladi. Bu chegaralar mohiyatlar nusxalarining berilganlar bazasi-dagi qiymatlari o'zgarganda o'rnatilgan qoidalar buzilishi yoki buzilmassisligini aniqlashga xizmat qiladi. Kardinallik chegaralarini akslantirgan model, aloqalar semantikasini aniqroq tasvirlaydi va buning natijasida predmet sohani ancha yaxshi tavsiflash imkonini beradi.

Hamma zaruriy aloqalar aniqlangandan keyin, modeldag'i har bir aloqa «haqiqiy dunyo»dagi bog'liqliklarni to'g'ri akslantirishi tekshiriladi.

Odatda, berilganlar bazasi ikki va undan ortiq mohiyatlardan yaratilgan bo'lsa, har bir mohiyat hech bo'lmaganda bitta aloqaga ega bo'lishi kerak, aks holda berilganlar bazasidagi vositalar orqali ushbu mohiyatga o'tib bo'lmaydi. Agarda bir nechta mohiyatlardan iborat berilganlar bazasida aloqasiz mohiyat mavjud bo'lsa, modelda u boshqa nomda qatnashayotgan yoki qatnashmayotganligini tekshirish kerak. Agarda bu tekshirish natija bermasa, yana talablarni qayta ko'rib chiqish va qaysidir aloqa qolib ketganligini aniqlash kerak. Bu ham natija bermasa, foydalanuvchilar bilan qayta muhokama o'tkazib, ushbu aloqasiz mohiyatdan qanday foydalanish kerakligini aniqlash zarur. 15.2-rasmida Talabalar o'quv loyihamizning ER-diagrammasining birinchi naqli keltirilgan.



15.2-rasm.

### **Atributlarni aniqlash, mohiyatlarning turлari va aloqalar bilan bog'lash**

Ushbu bochqichning maqsadi, atributlarni mos mohiyat yoki aloqa turлari bilan bog'lashdan iborat. Buning uchun yaratilayotgan berilganlar baza-sidan ajratilgan mohiyat va aloqalarni tafsiflovchi hamma berilganlarni aniqlash kerak. Bu erda ham mohiyatlarni identifikasiyalash uchun qo'llangan uslubdan foydalanamiz. Buning uchun hamma «ot» va ularni saqlovchi ifodalarni, foydalanuvchi talablari tafsiridan ajratib olamiz. Agarda «ot»lar mohiyat yoki aloqaning xossasi, sifati, identifikatori yoki xususiyatini tafsiflasa, u holda ular atribut bo'la oladi.

Mohiyat yoki aloqaning atributlarini aniqlashning soddarоq usuli foydalanuvchi talablari tafsiridan foydalanish va «Ushbu mohiyat va aloqa xususida qanday berilganni saqlash kerak?», degan savolga javob topish kerak. Ba'zi hallarda foydalanuvchilardan o'z talablarini aniqlashtirishni so'rash kerak.

Har bir atribut sodda yoki tarkibli bo'lismeni tasdiqlagan edik. Guruhli (tarkibli) atributlar sodda atributlar termasidan iborat. Ushbu bosqichda hamma oddiy atributlarni berilganlat bazasining konseptual modeli tarkibiga kiritish kerak.

Odatda atributlarning soni mohiyatlar va aloqalarning sonidan anch'a ko'p bo'lishini inobatga olgan holda, avval hamma atributlarning ro'yxatini foydalanuvchi talablari tafsiridan tayyorlash ham mumkin, bu uchinchi usul bo'ladi. Navbatdagi atribut birorta mohiyat yoki aloqa bilan bog'lansa, uni tuzilgan ro'yxatdan o'chirish kerak. Ushbu usul har bir atribut birorta mohiyat yoki aloqaning bitta turi bilan bog'lanishini kafolatlaydi. RO'YHATdan oxirgi atribut o'chirilganda, hamma identifikatsiyalangan atributlar birorta mohiyat yoki aloqa bilan bog'langan bo'ladi.

Talabalar o'quv loyihamizdag'i tasavvurlar tahlili natijasida, mohiyatlar aniqlangan atributlarning ro'yxati 15.3-rasmda keltirilgan.

Mohiyat nomi	Atributlar nomi
RO'YHAT	Zn_Famliya_Ism_C_Ism_K_Sama_Fak_N_Yon_K_Kurs_Guruh_N_Pasp_S_Pasp_H
BAHOLAR	Zn_Semestr_Fan_N_Bah_Sana_Ka_Usoz
FAKULTETLAR	Fak_N_Fakultet
YO'NALISH	Yon_N_Yonalishlar
FANLAR	Fan_N_Fan_Nomi

### 15.3-rasm.

Har bir atributning nomi o'z xususiyatidan kelib chiqqan holda, foydalanuvchi tushunadigan birorta ma'noga ega bo'lishi kerak. Har bir atribut xususida, quyidagi ma'lumotlar berilganlar lug'atida saqlanishi kerak:

- atribut nomi va uning tavisi;
- berilganlar turi va qiymatining formati (o'chami);
- atributning hamma taxalluslari;
- atributning guruhli yoki guruhsizligi, guruhli bo'lsa, uning tarkibi;
- atributning ko'p qiymatliligi;
- atributning hosilaviyligi, agarda hosilaviy bo'lsa, uni qiymatini hisoblash usuli;
- agarda atributning oshkormas ravishda qabul qiladigan qiymati bo'lsa, uning qiymati.

15.4-rasmda Talabalar o'quv loyihasidagi faqat RO'YHAT tasavvurining berilganlar lug'atidagi ko'rinishi misol tariqasida keltirilgan. Qolgan BAHOLAR, FAKULTETLAR, YO'NALISH va FANLAR tasavvurlarining berilganlar lug'atidagi ko'rinishi shu kabi yaratiladi. Ushbu vazifani mustaqil ravishda amalga oshirilg.

Mohiyat nomi	Atributlar	Tavsifi	Berilganlar turi va o'chhami	Ko'p qiyomatli	Oshkor-mas qiyomatli
RO'YHAT_Zn		Talabanining reyting da fiorchasingin raqami:	7 beligili satr	Yo'q	Yo'q
Familiya		Talabanining familyasi	15 beligili satr	Yo'q	Yo'q
Ism		Talabanining ismi	10 beligili satr	Yo'q	Yo'q
O_tan		Talabanining o'sasining ismi	15 beligili satr	Yo'q	Yo'q
K_Sana		Talabanining o'qishga kigan sanasi		Yo'q	Yo'q
Fak_N		Talaba o'qiydigan fakultetning raqami	Qisqa butun son	Yo'q	Yo'q
Von_N		Talaba o'qiydigan yo'nalishining raqami	Qisqa butun son	Yo'q	Yo'q
Kurs		Talabanining kursi	Qisqa butun son	Yo'q	Yo'q
Osolut_N		Talabanining ga'buli nomasi	Qisqa butun son	Yo'q	Yo'q
Pasp_S		Talabanining pasport seriyassi	2 beligili satr	Yo'q	Yo'q
Pasp_N		Talabanining pasportining nomasi	Butun son	Yo'q	Yo'q

15.4-rasm.

### Atributlarning domenlarini aniqlash

Atributlarning domenlarini aniqlashdan maqsad, atributlar uchun mumkin bo'lgan qiymatlar to'plamini aniqlashdan iborat. Atributlarning domenlari aniqlangandan keyin, ularning nomi va tavsifi berilganlar lug'atiga kiritiladi va bir yo'la atributlarga tegish yozuvlar ham yangilanadi, ularga har bir atribut uchun belgilangan domenlarning nomi, berilganlar turi va o'chhami o'miga yoziladi.

### Nomzod va birlamchi kalit bo'ladigan atributlarni aniqlash

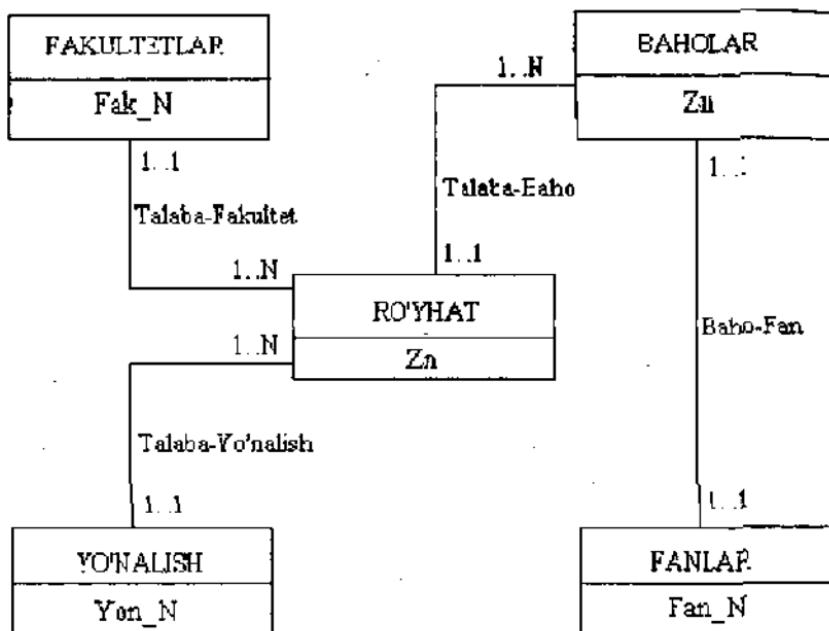
Ushbu bosqichdagi maqsad, har bir mohiyatdan hamma kalit bo'lishga nomzod atributlarni aniqlash va ularning orasidan esa, birlamchi kalitlarni tanlashdan iborat.

Kalitlikka nomzod atributlar ichidan birlamchi kalitni tanlash uchun quyidagi qoidalardan foydalanish kerak:

- kalitlikka nomzod atributlar ichidan minimal termani tanlash;
- kalitlikka nomzod atributlar ichidan qiymatining o'zgarish ehtimoli minimal bo'lganini tanlash;
- kalitlikka nomzod atributlar ichidan uzunligi eng kaltasini (satrli holat uchun) tanlash;
- kalitlikka nomzod atributlar ichidan eng kichik maksimal uzunkligisini (raqamli holatlар uchun) tanlash;
- foydalanuvchi nuqtayi nazaridan ishlatalish eng oson bo'lgan kalitlikka nomzod atributlar ichidan tanlash.

Birlamchi kalit tanlangandan so'ng, uning kuchli yoki zaif mohiyatligi o'matiladi. Agarda mohiyat uchun birlamchi kalitni tanlash mumkin bo'lgan bo'lsa, u kuchli mohiyat hisoblanadi, aksincha bo'lsa u zaif mohiyat bo'ladi. Zaif mohiyatning birlamchi kaliti, ajdod mohiyat bilan aloqasi akslantirilgandan keyingina, u bilan aloqa o'matish uchun tashqi kalit sifatida kuchli mohiyatning atributlaridan biri kiritiladi.

15.5-rasmda Talabalar o'quv loyihasidagi mohiyatlar birlamchi kalitlari bilan birga keltirilgan. Tashqi kalit BAHOLAR mohiyatida ishlataladi.



15.5-rasm.

### Modelni ortiqchalik nuqtayi nazaridan tekshirish

Bu bosqichda lokal konseptual model aniq maqsad bilan, ya'nii ortiqcha berilganlarni topish va topilgan holatda esa ulardan qutilish uchun tahlil qilinadi. Buning uchun quyidagi amallar bajariladi:

- «birga-bir» aloqalarni qayta tadqiqot qilish;
- ortiqcha aloqalardan qutilish.

Shunday bo'lishi mumkinki, mohiyatlarni aniqlash jarayonida bitta konseptual obyektni tavsiflovchi ikkita mohiyat ishlatalgan, masalan Mijoz va Klient. Ikkalasi bir maqsadda, bir ma'noni anglatadi, faqatgina bittasi

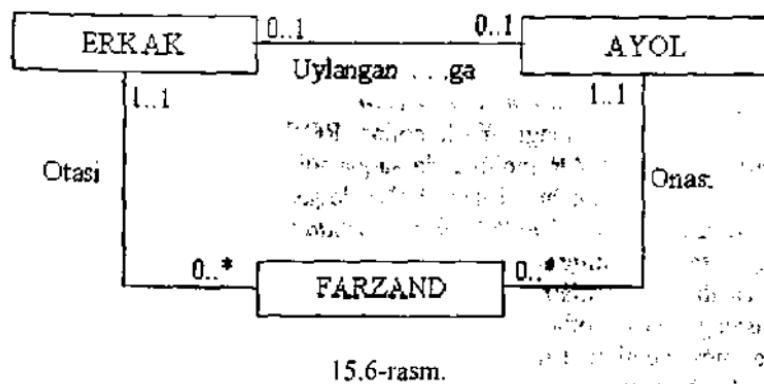
o'zbek tilida ikkinchisi esa rus tilida keltirilgan. Bunday holatlar hamda sinonimlar bilan atalgan mohiyatlar uchraganda, ularni tahlil qilib birlashish kerak. Ular uchun turli birlamchi kalitlar tanlangan bo'lsa, bittasini birlamchi, ikkinchisini muqobil kalit sifatida ishlatalish kerak.

Agar biror aloqadagi ma'lumotlarni boshqa aloqalar orqali olish mumkin bo'lsa, bu aloqa ortiqcha hisoblanadi. Yaratuvchi doimo berilgan-larning minimal modelini tuzishga harakat qiladi, shuning uchun ortiqcha aloqalar kerak emas, ular modeldan olib tashlanishi kerak. ER-diagrammada ortiqcha aloqalar oson aniqlanadi, chunki ular ikkita mohiyatni bir nechta yo'llar orqali bog'laydi. Lekin bu vaziyat har doim to'g'ri bo'lmasligi ham aloqaning ortiqchaligini bildirmasligi ham mumkin, chunki ular mohiyatlar orasidagi berilganlarning har xil bog'lanish assotsiatsiyasi (turi)ni bildirishi mumkin.

Ortiqchalikni baholashda, bu aloqalar qaysi vaqtida yuz berayotganini ham hisobga olish kerak. Masalan, ERKAK, AYOL va FARZAND mohiyatlari orasidagi aloqalarni modellashtirish jarayoniga nazar tashlaylik (15.6-rasm). ERKAK va FARZAND mohiyatlari orasida ikkita aloqa yo'li borligi aniq – birinchisi **Otasি** to'g'ridan-to'g'ri aloqa yo'li bo'lsa, ikkinchisi esa **Uylangan ...ga** va **Onasi** aloqasıdır. Shu aloqalarda, Otasi aloqa yo'li ortiqcha, deb faraz qilish mumkin, lekin bu fikr quyidagi sabablarga ko'ra noto'g'ri bo'lishi mumkin:

a) otasinining avvalgi nikohidan farzandi bo'lishi mumkin, bu holatda esa faqatgina joriy nikohi bo'yicha 1:1 aloqa modellashtirilgan;

b) otasi va onasi nikohda bo'lmasligi mumkin yoki otasi onasiga emas boshqa ayloga uylangan bo'lishi va aksincha, onasi bolaning otasi bo'lmasgan boshqa erkakka turmushga chiqqan bo'lishi mumkin.



Ikkala holatda ham, Otasi aloqa yo'lisisiz, zaruriy berilganlarni modellashtirib bo'lmaydi. Shuning uchun berilganlarning ortiqchaligiga baho

berishda, mohiyatlar orasidagi har bir aloqaning tayinlanish sababini chuqur tahlil qilish zarur. Ana shu ishlar bajarilgandan so'nggina berilganlarning lokal konseptual modelini hamma ortiqchaliklardan ozod qilish mumkin.

### **Lokal konseptual modelni aniq foydalanuvchining tranzaksiyalariga mosligini tekshirish**

Ushbu bosqichni bajarishdan maqsad, berilganlarning lokal konseptual modeli ko'rileyotgan tasavvur uchun zarur bo'lgan tranzaksiyalarni qo'llab-quvvatlashiga ishonch hosil qilish. Ko'rileyotgan predmet sohaning aniq tasavvuriga mos berilganlarning lokal konseptual modeli yaratildi, deb faraz qilaylik. Buning uchun hamma kerakli amallarni qo'llda ana shu model yordamida bajarish kerak. Agarda hamma tranzaksiyalarni ana shunday tekshirish mumkin bo'lgan bo'lsa, u holda konseptual modelning tranzaksiyalarga mosligi muvaffaqiyatli deb topiladi. Aks holat, modelda kamchiliklar borligi va bu kamchiliklarni bartaraf qilish kerakligini anglatadi. Kamchiliklar modelda birorta mohiyatlar, aloqalar yoki atributlar hisobga olinmaganidan bo'lishi mumkin.

Lokal konseptual modelni aniq foydalanuvchining tranzaksiyalariga mosligini tekshirishning ikkita usulini ko'rib chiqamiz:

- a) tranzaksiyalar tavsifi;
- b) tranzaksiyalarni bajarilish yo'llari orqali tekshirish.

*Birinchi usuldan* foydalanganda, model har bir tranzaksiya uchun mohiyatlar, aloqalar va ularning atributlari xususidagi hamma zaruriy ma'lumotlarni bera olishi tekshiriladi. Buning uchun har bir tranzaksiya uchun talablarning tavsifi tuzilishi kerak. Bu usulga misol tariqasida Talabalar o'quv loyihasidagi RO'YHAT tasavvuri talablaridan foydalana-digan tranzaksiyalarga misollar keltiramiz.

*Tranzaksiya A. Har bir talabani alfavit bo'yicha saralab, uning reyting ballari to'g'risidagi ma'lumotlar tuzilsin.*

Talabalar xususidagi ma'lumotlar RO'YHAT mohiyatida, reyting ballari esa BAHOLAR mohiyatida saqlanadi. Bu so'rov bo'yicha zaruriy ma'lumotlarni olish uchun Talaba-Baho aloqasidan foydalanish mumkin.

*Tranzaksiya V. 2008/09 o'quv yilida «Fizika» fakultetiga qabul qilingan talabalarning ro'yxati tuzilmasi.*

Talabalar va ularning o'qishga kirgan sanasi va fakulteti raqami xususidagi ma'lumotlar RO'YHAT mohiyatida, fakultetlarning o'z raqamiga mos nomlari FAKULTETLAR mohiyatida saqlanadi. Bu so'rov bo'yicha zaruriy ma'lumotlarni olish uchun Talaba-Fakultet aloqasidan foydalanish kerak.

*Tranzaksiya S. «Informatika va axborot texnologiyalari» va «Axborot xavfsizligi» yo'nalishlarida ta'lim oluvchi talabalarini ro'yxati tuzilsin.*

Talabalar va ularning yo'nalishlari xususidagi ma'lumotlar RO'YHAT mohiyatida, yo'nalishlarning o'z raqamiga mos nomlari YO'NALISH mohiyatida saqlanadi. Bu so'rov bo'yicha zaruriy ma'lumotlarni olish uchun Talaba-Yo'nalish aloqasidan foydalanish kerak.

*Ikkinci usuldan foydalanish uchun, bevosita berilganlar modeli uchun yaratilgan ER-diagrammada yo'llarni chizmada tasvirlanishi talab etiladi. Tranzaksiyalarning soni qancha ko'p bo'lsa, chizma shunchalik murakkablashib boradi, shuning uchun tranzaksiyalarni bir nechta diagrammalarda saqlashga to'g'ri keladi. Bu usul qiyinchilik tug'dirsa ham, lekin aynan shu bosqichda tranzaksiyalarni tekshirish kerak, chunki keyinchalik bu ishlarni amalga oshirish murakkablashadi va qimmatlashadi.*

### **Berilganlarning lokal konseptual modelini oxirgi foydalanuvchilar bilan muhokama qilish**

Berilganlarning lokal konseptual modelini oxirgi foydalanuvchilar bilan muhokama qilishdan maqsad, ushbu modelning foydalanuvchilar tasavvuridagi tafsirlar talablariga mos kelganini aniqlashtirishdan iborat. Konseptual model ER-diagramma sisatida tuzilishi va yaratilgan berilganlar modelini tavsiflovchi kuzatuv hujjalari ega bo'lishi kerak. Kamchiliklar o'z vaqtida, bosqichlarni qo'shimcha tahlil qilish natijasida to'g'rilanishi va hujjalashtirilishi kerak. Bu jarayon foydalanuvchilar, «berilganlar modeli predmet sohaga to'liq mos keldi», deb ta'kidlamaguncha davom etishi kerak.

### **Xulosa**

*Loyihalash uslubiyati berilganlar bazasini loyihalashdagi tuzilishi yondoshuvdan iborat bo'lib, protseduralar, texnologiyalar, vositalar va hujjalardan foydalanishni ko'zda tutadi. Berilganlar bazasini loyihalash uslubiyati uchta – konseptual, mantiqiy va fizik loyihalash bosqichlarini ketma-ket ishlatishni nazarda tutadi.*

*Berilganlar bazasini konseptual loyihalash tashkilotdagi ish taribini axborot modelini yaratish jarayonidan iborat bo'lib, u bu jarayoni amalga oshirishdagi ixtiyoriy fizik ko'rsatkichlarga bog'lanmaydi.*

*Berilganlar bazasini konseptual loyihalash tashkilotdagi berilganlarning konseptual modelini yaratishdan boshlanadi. U qanday keyinchalik qaysi BBB tizimi qo'llanishidan qat'i nazar, amalga oshirish bilan bog'liq bo'lgan har qanday omillarga bog'lanmagan holda qaratadi.*

*Ayniqsa, amaliy programmalar, ishlatalidagan programmalush tillari, tanlangan hisoblash platformasi, samaradorlik muammolari va boshqa fizik shartlarga bog'liqsiz ravishda.*

### **Nazorat uchun savollar**

1. Loyihalash uslubiyatining umumiyligi vazifasi nimadan iborat?
2. Berilganlar bazasini loyihalashdagi asosiy bosqichlar nimalardan iborat?
3. Berilganlar bazasini muvaffaqiyatli yaratish jarayonidagi asosiy omillarni ko'rsating?
4. Berilganlar bazasini loyihalash jarayonidagi oxirgi foydalanuvchilarning roli nimadan iborat?
5. Berilganlar bazasini konseptual loyihalash bosqichidagi asosiy maqsad nimadan iborat?
6. Berilganlar bazasini konseptual loyihalashdagi asosiy bosqichlar qaysilar?
7. Foydalanuvchi tafsirlaridagi talablardan kelib chiqqan holatda, mohiyatlarning turlari va aloqalari qanday aniqlanadi?
8. Foydalanuvchi tafsirlaridagi talablar asosida atributlarni aniqlash uchun qanday usul ishlataladi.
9. Atributlarning qaysi mohiyat va aloqalarga tegishli ekanligini aniqlash uchun qanday usuldan foydalilanadi?
10. Berilganlar modelidagi ortiqchalikni tekshirish usullarini misollar bilan ta'riflab bering.
11. Berilganlarning konseptual modelini tekshirish zaruriyati nima bilan bog'liq ekanligini tushuntiring.
12. Berilganlarning konseptual modelini tekshirishning birinchi usulini misollar bilan izohlang.
13. Berilganlarning konseptual modelini tekshirishning ikkinchi usulidagi asosiy qiyinchilik nimadan iborat?
14. Berilganlarning konseptual modelini loyihalash bosqichidagi hujjatlashtirishning ahamiyati nimadan iborat?

## 16-bob. Berilganlar bazasini mantiqiy loyihalash

**Tayanch iboralar:** *relatsion berilganlar bazasini mantiqiy loyihalash usuli, rekursiv aloqa, rekursiv «birga bir» aloqa, rekursiv «birga ko'p» aloqa, rekursiv «ko'plikdan ko'plikkha» aloqa, murakkab aloqa, ajdod munosabat, avlod munosabat, kuchli turdag'i mohiyatlar, zaif turdag'i mohiyatlar, burlamchi kalit, tashqi kalit, muqobil kalit, birinchi normal shakl, ikkinchi normal shakl, uchinchi normal shakl. Boys-Kodd normal shakl, to'rtinchchi normal shakl, beshinchchi normal shakl, berilganlarning butunlik chegarasi, munosabatlarning global diagrammasi, global ER-diagramma.*

Relatsion berilganlar bazasini mantiqiy loyihalash konseptual model-dagi ba'zi alohidalik xususiyatlarini yo'qotishdan boshlanadi. Masalan, bunday alohidalik xususiyatlariga relatsion modelda ko'rsatib bo'lmaydigan, «ko'plikdan ko'plikkha» aloqalari kiradi. Lekin bu amal shart bo'l-magan amal sifatida qaraladi, chunki u keyingi bosqichlarni sodda-lashtirishga xizmat qildi. Bu bosqichlarda esa, model munosabatlar maj-muasiga aylantiriladi. Lekin shunga qaramasdan, agarda mantiqiy modeldan «ko'plikdan ko'plikkha» aloqalarini yo'qotish maqsadga muvofiq bo'lmasa, bevosita bu aloqalarni munosabatlar majmuasiga akslantirish usulini ko'rsatamiz. Keyin olingen munosabatlar 11-bobda keltirilgan normallash-tirish usullari bo'yicha tekshiriladi. Shundan keyin har bir beril-ganlarning mantiqiy modeli, foydalanuvchi talab qiladigan tranzaksiyalarni qo'llab-quvvatlashi tekshiriladi.

Akslantirish va tekshirish jarayoni, har bir tasavvur uchun lokal mantiqiy model yaratilguncha, davom ettiladi. Agar konseptual modelda faqatgina bittagina tasavvur bor bo'lsa, u holda berilganlar bazasini mantiqiy loyihalash bosqichi tugatiladi. Bir nechta tasavvurlar berilgan bo'lsa, lokal mantiqiy modellarni global mantiqiy modelga birlashtirish jarayoni o'tkaziladi va bunda berilgan korxonaning barcha foydalanuv-chilarining tasavvurlariga mosligi tekshiriladi.

Berilganlar bazasini mantiqiy loyihalash bosqichi Talabalar o'quv loyihamizning avvalgi bobda yaratilgan konseptual modeli asosida o'tkaziladi.

## **Relatsion berilganlar bazasini mantiqiy loyihalash usuli**

Berilganlar bazasini mantiqiy loyihalash jarayoni, foydalanuvchilarning alohida modellari asosida umumiylar axborot modelini yaratishdan iborat bo'lib, u hech qaysi aniq foydalanadigan BBB tizimlarining alohida xususiyatlari va boshqa fizik shartlarga bog'liqsiz bo'ladi.

Biz *relatsion berilganlar bazasini mantiqiy loyihalash usulini* quyidagi ikkita bosqichga ajratilgan holatlari bo'yicha o'rghanamiz:

- foydalanuvchilarning alohida tasavvurlari bo'yicha lokal mantiqiy berilganlar modelini yaratish va tekshirish;
- global mantiqiy berilganlar modelini yaratish va tekshirish.

### **Foydalanuvchilarning alohida tasavvurlari bo'yicha lokal mantiqiy berilganlar modelini yaratish va tekshirish**

*Ushbu bosqichda yechiladigan masalalarning asosiya maqsadi*, lokal mantiqiy konseptual berilganlar modeli asosida, premet soha bo'yicha foydalanuvchilarning aniq tasavvurlarini akslantiruvchi lokal mantiqiy berilganlar modelini yaratish va shu modelni normallashtirish usullari asosida tekshirish hamda tranzaksiyalarning bajarilishini nazorat qilishdan iborat.

Ushbu bosqichda 15-bobda yaratilgan har bir lokal konseptual berilganlar modelini, relatsion chizmaning ER-diagrammasi va kuzatuv hujjatlaridan iborat, lokal mantiqiy berilganlar modeliga aylantiramiz. Bevosita relatsion modelda kelishi mumkin bo'lmagan alohida, masalan «ko'plikdan ko'plikka» kabi xususiyatlarni modeldan yo'qotamiz. Hosil bo'lgan relatsion chizmani normallashtirish qoidalari bo'yicha va chizmaning tuzilishi nuqtayi nazaridan tahlil qilamiz. Xuddi shunday mantiqiy modelni ham foydalanuvchi talablarining tafsirida keltirilgan talablarga tranzaksiyalari javob berishi nuqtayi nazaridan tekshiramiz. Keyin esa berilganlarning mantiqiy modeliga butunlik chegaralarini kiritamiz.

Ushbu bosqichning oxirida, bizda tasavvurning to'g'ri, to'liq va ziddiyatli bo'lmagan modeli hosil bo'lishi kerak.

### **Relatsion modelga mos kelmaydigan alohidalikkardan qutilish (majburiy bo'lmagan amaliyat)**

Ushbu majburiy bo'lmagan amaliyotda ko'rildigani masalalar nimolaridan iborat bo'lishi va qanday yechilishini yetkazish nuqtayi nazaridan biz ularni quyida keltiramiz:

- ikki tomonlama «ko'plikdan ko'plikka» aloqalardan qutilish;

- rekursiv «ko'plikdan ko'plikka» aloqalardan qutilish;
- murakkab aloqalardan qutilish;
- ko'p qiymatli atributlardan qutilish.

Agarda konseptual modeldag'i mohiyatlar orasida «ko'plikdan-ko'plikka» aloqasi mavjud bo'lsa, bu aloqani dekompozitsiya qilib, oraliqda joylashgan mohiyatni aniqlayniz. «Ko'plikdan ko'plikka» (\*:\*) aloqasi ikkita «birlikdan ko'plikka» (1:\*) aloqalariga ajratiladi va uchinchi aloqa ishtirokchisi sifatida yangi aniqlangan mohiyat ishtirok etadi.

*Rekursiv aloqa* alohidagi aloqalar turi bo'lib, u orqali mohiyat o'zi bilan o'zi aloqa qiladi. Bu aloqalarning uch turi mavjud:

- rekursiv «birga-bir» aloqa (1:1);
- rekursiv «birga-ko'p» aloqa (1: \*);
- rekursiv «ko'pga-ko'p» aloqa (\*: \*).

Birinchi turdag'i rekursiv aloqa tuzilishini o'zgartirmagan holda, relatson modeldag'i bitta munosabatga aylantirish mumkin. Agarda, 1: \* aloqasi «ko'plik» aloqasi tomondan aloqaning ishtirok etishini shart qilib qo'ymasa, u holda berilganlar bazasida saqlanadigan nol qiymatlarni kamaytirish maqsadida, ikkinchi munosabatni yaratish kerak. Uchinchi turda, konseptual modelda rekursiv (\*: \*) aloqasi ishtirok etayotgan bo'lsa, u holda shu aloqani dekompozitsiya qilib, oraliq mohiyat aniqlanadi.

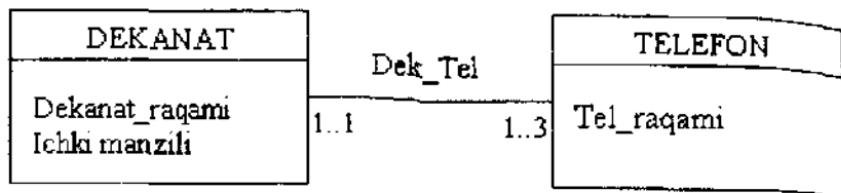
*Murakkab aloqa* deganda, uch va undan ortiq mohiyatlarning turlari ishtirokidagi aloqalar tushuniladi. Agarda berilganlarning konseptual modelida murakkab aloqa mavjud bo'lsa, uni dekompozitsiya qilib, oraliq mohiyat aniqlanadi. Murakkab aloqa esa, yangi topilgan mohiyat bilan zaruriy sondagi 1: \* aloqalarga almashtiriladi.

Ko'p qiymatli atribut, bitta mohiyatga mos bo'lган bir qancha qiymatlarni saqlaydi. Agarda berilganlarning konseptual modelida ko'p qiymatli atribut mavjud bo'lsa, shu atributni dekompozitsiya qilib, yangi mohiyatni aniqlash mumkin. Masalan, dekanatlarda ucta telefon nomeri bo'lishi mumkin, degan tasavvurni akslantirish uchun Tel\_raqami atributi DEKANAT mohiyatida ko'p qiymatli sifatida akslantirilgan bo'lsin (16.1-rasm).

DEKANAT
Dekanat raqami
Ichki manzili
Tel_raqami 1.3

16.1-rasm.

Bu ko'p qiymatli atributni olib tashlab, uning o'mniga TELEFON mohiyatini bir qiymatli va oddiy Tel\_raqami atributi (birlamchi kalit) hamda Dek\_Tel aloqasi (1:3) bilan almashtiramiz (16.2-rasm).



16.2R-rasm.

### **Lokal mantiqiy berilganlar modeli tuzilishi asosida munosabatlarni aniqlash**

Ushbu jarayonning maqsadi, lokal mantiqiy berilganlar modelida aniqlangan mohiyatlar, aloqalar va atributlarga mos berilganlar munosabatlarni aniqlashdan iborat.

Yaratiladigan munosabatlarning tuzilishini tavsiflash uchun relations BBB tizimlarida ishlataladigan DDL (Database Definition Language) - berilganlar bazasini aniqlash tilidan foydalanamiz. DDL tilida munosabatlarni tavsiflash, munosabat nomi va dumaloq qavslarga olingan oddiy atributlar nomini keltirish bilan boshlanadi. Keyin esa, munosabatning birlamchi kaliti va uning hamma nomzod va/yoki tashqi kalitlari keltiriladi. Har bir tashqi kalitning yonida u murojaat qiladigan birlamchi kalit ko'rsatiladi. Hamma hosila atributlar ularning qiymatlarini hisoblash uchun ishlataladigan atributlar bilan birga sanab o'tiladi. Ikkita munosabatlarni orasidagi aloqalar «birlamchi kalit/tashqi kalit» shaklida keltiriladi. Tashqi kalitning atribut (farji) qaysi munosabatda ekanligini aniqlashda, aloqada ishtiroy etuvchi qaysi munosabat *ajdod*, qaysi biri *avlod* ekanligi aniqlanishi kerak. O'zining birlamchi kalitining husxasini *avlodiga tashqi kalit* sifatida ishlatalishga uzatgan munosabat, *ajdod munosabat* hisoblanadi.

Quyida biz berilganlar modelida keltirilgan tuzilishlar asosida munosabatlarni yaratish usullarining tavsiflarini keltiramiz.

**Kuchli turdag'i mohiyatlar.** Berilganlar modelida har bir kuchli mohiyat uchun, shu mohiyatning hamma oddiy atributlaridan iborat munosabat yaratiladi. Agarda modelda tarkibli (guruh) atribut uchrasha, uning tarkibidagi sodda atributlar kuchli mohiyatga kiritiladi. Masalan, bizning o'quv loyihamizdagi RO'YHAT kuchli mohiyatning tuzilishi quyidagicha bo'ladi:

- RO'YHAT (Zn, Familiya, Ism, O\_Ism, K\_Sana, Fak\_N, Yon\_N, Kurs, Guruh\_N, Pasp\_S, Pasp\_N)
- Birlamchi kalit Zn

**Zaif turdag'i mohiyatlar.** Berilganlar modelida har bir zaif mohiyat uchun, shu mohiyatning hamma oddiy atributlaridan iborat munosabat yaratiladi. Zaif mohiyatning birlamchi kaliti, to'liq yoki qisman ajdod mohiyatning kalitiga bog'liq. Shuning uchun zaif mohiyatning kalitini, ajdod mohiyatdagi hamma aloqalarni aniqlash jarayoni tugamaguncha, belgilab bo'lmaydi. Lekin bizning o'quv loyihamizda, BAHOLAR zaif mohiyati quyidagi munosabatga aylanadi:

- BAHOLAR (Semestr, Fan\_N, Ball, Sana\_Kir, Ustoz)
- Birlamchi kalit hali aniqlanmagan.

**Ikki tomonlama «birga-ko'p» aloqalar.** Har bir ikki tomonlama 1:<sup>\*</sup> aloqada ishtirok etuvchilardan biri, ya'nii «birlik» tomondag'i mohiyat ajdod, «ko'plik» tomondag'i mohiyat esa, avlod sifatida aniqlanadi. Bunday aloqalarni belgilash uchun, ajdod mohiyatdagi atributning nusxasi mos ravishdagi avlod mohiyatga o'tkaziladi va bu ajdod mohiyat uchun tashqi kalit bo'ladi. Yana bir bor 16.3-rasmga e'tibor qaratamiz, Zn atributi BAHOLAR zaif mohiyati uchun birlamchi kalit, RO'YHAT mohiyati uchun esa tashqi kalit sifatida keladi.

Masalan, 16.3-rasmidagi Talaba-Baho aloqasiga e'tibor beraylik. Bu aloqa 1:<sup>\*</sup> turida, chunki har bir talaba bir nechta baholarni olishi mumkin. Ushbu misolimizda, RO'YHAT mohiyati aloqaning «birlik» tomonida, demak, u ajdod mohiyat bo'ladi. BAHOLAR mohiyati esa «ko'plik» tomonida, bu esa avlod mohiyat bo'ladi. Bu mohiyatlar orasidagi aloqa ajdod mohiyat (RO'YHAT) ning birlamchi kalitini avlod mohiyat (BAHOLAR)ga uzatish natijasida o'matiladi. Ikkala mohiyatlarning o'zaro harakati 16.3-rasmda keltirilgan. Har bir bahoning kiritilgan sanasi (Sana\_Kir) ham unga bog'liq bo'lgani uchun ham, bu atribut BAHOLAR munosabatidan joy olgan.

Za erigi BAHOLAR mohiyati 1:<sup>\*</sup> turdag'i o'sirishish uchun esa

RO'YHAT (za Familiya, O\_Ism, K\_Sana,  
Jek\_N, Yon\_N, Kurs, Guruh\_N, Pasp\_S, Pasp\_N)  
Birimchi kalit Zn

BAHOLAR (za Semestr, Ball, Sana\_Kir, Ustoz,  
Tasdiq bilan Zn, RO'YHAT (Zn) qo'shishiga jihatda

## **Ikki tomonlama «birga-bir» aloqalar**

1:1 aloqalarga mos munosabatlarni yaratishda ancha murakkablik bor, chunki bunda qaysi bir mohiyat ajdod, qaysi biri avlod mohiyat bo'lishini aniqlash uchun aloqalarning kardinalligidan foydalanib bo'lmaydi. Shuning uchun, bu aloqani ishtirokchilari bo'lgan hamma munosabatlarni bitta munosabatga aylantirish yoki ikkita munosabat yaratib, birinchi munosabatdagi birlamchi kalitning nusxasini ikkinchi munosabatga uzatish kerakmi, degan masalani hal qilish uchun aloqada ishtirok etuvchi munosabatlarning darajalarini cheklash qoidasi qo'llaniladi. Munosabatlarni yaratish jarayonida, quyidagi ishtirok etish darajalarini cheklash turlariga mos keluvchi munosabatlardan foydalaniladi:

- a) 1:1 aloqadagi ikkita munosabatning majburiy ishtiroki;
- b) 1:1 aloqadagi ikkita munosabatdan bittasining majburiy ishtiroki;
- v) 1:1 aloqadagi ikkita munosabatning majburiy bo'lmagan ishtiroki.

### **1:1 aloqadagi ikkita munosabatning majburiy ishtiroki**

Bu holda, aloqada ishtirok etuvchi mohiyatlarni bitta munosabatga birlashtirish zarur hamda ulardan birining birlamchi kalitini yangi munosabatning birlamchi kaliti, boshqa birlamchi kalit mavjud bo'lsa, uni muqobil kalit sifatida olish kerak.

### **1:1 aloqadagi ikkita munosabatdan bittasining majburiy ishtiroki**

Bunda, 1:1 aloqa uchun ishtirok etish darajalarini cheklash turlariga mos keluvchi ajdod va avlod mohiyatlarni aniqlash masalasi oson yechiladi. Aloqada ishtirok etishi shart bo'lmagan mohiyat ajdod, ishtirok etishi shart bo'lgan mohiyat esa avlod sifatida aniqlanadi. Ajdod mohiyatdagi birlamchi kalitning nusxasi avlod mohiyatni ifodalovchi munosabatga joylashtiriladi. Agar bu aloqa bitta yoki bir nechta atribut bilan aloqada bo'lsa, u holda ular birlamchi kalit bilan birga avlod munosabatga joylashtirilishi kerak.

### **1:1 aloqadagi ikkita munosabatning majburiy bo'lmagan ishtiroki**

Bu vaziyatda esa, agarda mohiyatlardan birortasining foydasi uchun ko'rilyotgan aloqa xususida birorta qo'shimcha ma'lumot bo'limasa, ixtiyoriy mohiyatdan bittasini ajdod, ikkinchisini esa avlod mohiyat sifatida belgilash mumkin.

### **Rekursiv «birga-bir» aloqa (1:1)**

Rekursiv «birga-bir» aloqalarga ham yuqorida keltirilgan ishtirok etish darajalarini cheklash qoidasi ishlataladi. Lekin bu aloqaning alohidalik turidagi qoidaning majburiylik qismida, 1:1 aloqaning ikkala tomonida bitta mohiyat turadi. Shuning uchun, bitta munosabatda birlamchi kalitning ikkita

nusxasi bo'sadi, lekin ulardan bittasi tashqi kalitga mos keladi va uning nomi, mos aloqani akslantirayotganini ko'rsatish uchun, o'zgartirilishi kerak.

Rekursiv «birga bir» aloqadagi bitta tomonning ishtirok etishi qoidasi bo'yicha, keltirilgan usul bo'yicha bitta munosabatni birlamchi kalitning ikkita nusxasi bo'yicha yaratish yoki 1:1 aloqani akslantiruvchi yangi munosabatni yaratish kerak. Yangi munosabat, birlamchi kalitlarning nusxalaridan iborat, faqatgina ikkita atributdan yaratilishi kerak. Avvalgi holadagi kabi, bu birlamchi kalitning ikkala nusxasi ham tashqi kalitlar sifatida nomi o'zgartirilgan holda kelishi va ularning har bir munosabatdagi vazifasi aniqlanishi kerak.

Rekursiv «birga bir» aloqadagi ikkala tomonning ishtirok etishi shart bo'limgan qoida bo'yicha esa, keltirilgan usulga binoan yangi munosabat yaratish kerak.

### **Ikki tomonlama «ko'pga-ko'p» aloqalar**

Har bir ikki tomonlama «ko'pga-ko'p» aloqalar uchun yangi munosabat yaratish va shu aloqalarni akslantirish hamda unga shu aloqada ishtirok etuvchi hamma atributlarni kiritish zarur. Aloqada ishtirok etuvchi mohiyatlarning birlamchi kalitlarining nusxalarini yangi munosabatga tashqi kalit sifatida ishlatalish uchun uzatilishi kerak. Ushbu tashqi kalitlar yangi munosabat uchun birlamchi kalit vazifasida aloqadagi boshqa atributlar bilan birgalikda ham kelishi mumkin. Agarda kalitning noyobligini, aloqani tashkil etuvchi bitta yoki bir nechta atributlar tashkil atsa, u holda berilganlarning konseptual modelida qaysidir mohiyat hisobga olinmagan bo'lishi mumkin. Lekin bu kamchilikni yuqorida tavsiflangan o'rgartirishlar orqali yo'qotish mumkin.

### **Aloqalarning murakkab turlari**

Har bir murakkab aloqa turi uchun ushbu aloqani akslantiruvchi munosabat yaratiladi va unga shu aloqada ko'rileyotgan hamma atributlar kiritiladi. Yangi munosabatga, tashqi kalit sifatida ishlatalish uchun murakkab aloqada ishtirok etayotgan mohiyatlarning birlamchi kalitlarining nusxalari joylashtiriladi. Aloqaning «ko'plik» tomonida ishtirok etuvchi hamma tashqi kalitlar, odadta yangi munosabatning birlamchi kalitini ham ifoda laydi, lekin bu holada birlamchi kalit tarkibida aloqadagi boshqa atributlar ham ishtirok etishi mumkin.

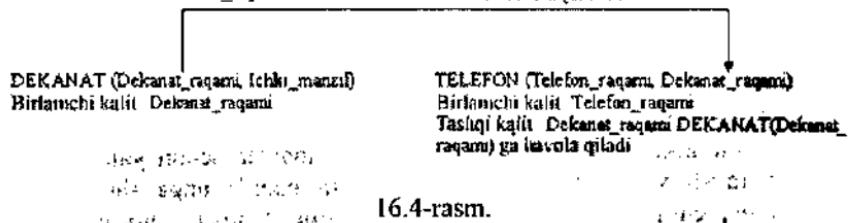
### **Ko'p qiymatli atributlar**

Mohiyatdagi har bir ko'p qiymatli atributlar uchun ko'p qiymatli atributlarga mos yangi munosabat yaratiladi va unga birlamchi kalit tashqi

kalit sifatida ishlatish uchun uzatiladi. Agar ko'p qiymatli atribut mohiyatning muqobil kaliti bo'lmasa, u holda yangi munosabatning birlamchi kaliti ko'p qiymatli atributdan va mohiyatdagi birlamchi kalitning birlashmasidan tuziladi.

Masalan, 16.1-rasmda dekanatlarda uchta telefon nomeri bo'lishi mumkin, degan tasavvumi akslantirish uchun Tel\_raqami atributi DEKANAT mohiyatida ko'p qiymatli sifatida akslantirilgan edi. Bu atributni o'zgartirish uchun DEKANAT mohiyati uchun ko'p qiymatli Tel\_raqami atributidan iborat TELEFON munosabati yaratiladi. Natijada 16.4-chizmadagi tuzilish hosil bo'ladi.

*Dekanat\_raqami* atributi TELEFON munosabatiga uzatilish



16.4-rasm.

16.1 jadvalda munosabatdagi mohiyat va aloqalarni o'zgartirish bo'yicha natijaviy ma'lumotlar keltirilgan.

16.1-jadval.

Mohiyat-aloqa	O'zgartirish uslubi
Kuchli mohiyat	Hamma oddiy atributlardan iborat munosabatni yaratish
Zaif mohiyat	Hamma oddiy atributlardan iborat munosabatni yaratish (har bir ajdod mohiyat aloqlari o'zgartirilgandan keyin birlamchi kalitni aniqlash zarur)
1:* turidagi ikki tomonlama aloqa	«Birlik» tomondagi mohiyat (ajdod)ning birlamchi kalitini, «ko'plik» tomondagi mohiyat (avlod)ga birlamchi kalit sifatida o'tkaziladi. Aloqaga tegishli hamma atributlar «ko'plik» tomonga o'tkaziladi
1:1 turdaggi ikki tomonlama aloqa	
ikki tomonning majburiy ishtiroki	Mohiyatlar bitta munosabatga birlashtiriladi
bir tomonning majburiy ishtiroki	Ishtiroki «majburiy bo'Imagan» mohiyatning birlamchi kaliti (ajdod)ga, ishtiroki «majburiy bo'lgan» mohiyat (avlod)ga tashqi kalit sifatida o'tkaziladi

ikkala tomonning majburiy bo'lmagan ishtiroki	Mohiyatlar uchun qo'shimcha ma'lumot bo'lmasa, ixtiyoriy ravishda ajdod va avlod mohiyatlar tanlanadi
Ikki tomonlarga ** turidagi murakkab aloqa	Aloqani ifodalovchi munosabat yaratiladi va unga hamma aloqalarga tegishli atributlar kiritiladi. Har bir ajdod mohiyatdagи birlamchi kalitlarning nusxalarini, tashqi kalitlar sifatida ishlatalish uchun, yangi munosabatga uzatiladi
Ko'p qiymatli atribut	Ko'p qiymatli atributdan iborat yangi munosabat yaratiladi va ajdod mohiyatning birlamchi kalitining nusxasini, tashqi kalit sifatida ishlatalish uchun yangi munosabatga uzatiladi

Ushbu bosqichdagi amallar bajarilgandan keyin, Talabalar o'quv loyihamizniug RO'YHAT tasavvuriga mos berilganlarning mantiqiy modeli asosida olingan munosabatlarni rasmiy ravishdan DDL tilida tafsiflash natijalari 16.2-jadvalda keltirilgan.

#### 16.2-jadval.

Munosabatlari	
RO'YHAT (za, Faniya, Ism_Q_Ism, K_Sana, Fak_N, Yon_E, Kurs, Guchh_N, Pasp_S, Fasp_N)	
Birlamchi kalit Za	
Tashqi kalit Fak_N FAZULCEMLAR(Fak_N) ga havola qilinadi	
Tashqi kalit Yon_N YONALISH(Yon_n) ga havola qilinadi	
Mugobil kalit Pasp_S + Fasp_N	
BAHOLARI (za, Sessess, Fan_N, Bal, Sana_Xiz, Urzor)	
Birlamchi kalit Za	
Tashqi kalit Za RO'YHAT (za)ga havola qilinadi	
Tashqi kalit Fan_N FANLAR (Fan_N) ga havola qilinadi	
Mugobil kalit Za + Fan_N	
FAKULTELLAR (Fak_N, Fakultet)	
Birlamchi (eji) Fak_N	
Mugobil kalit Fakultet	
YONALISH (Yon_K, Yonalishlar)	
Birlamchi kalit Yon_N	
Mugobil kalit Yonalishlar	
FANLAR (Fan_N, Fan_Nomi)	
Birlamchi kalit Fan_N	
Mugobil kalit Fan_Nomi	

## **Normallashtirish qoidalarini qo'llagan holda munosabatlarni tekshirish**

Munosabatlarni normallashtirish qoidalarini biz batapsil 11- bobda ko'rib chiqqan edik. Normallashtirish berilganlar modelini yaxshilash, ya'ni berilganlarni takrorlamaslikda turli cheklashlarni qondirish uchun ishlataladi. Normallashtirish jarayoni uni qo'llash natijasida olingan berilganlar modeli, korxonadagi ma'lumotlardan foydalanishni alohida xususiyatlarini akslatirishda ziddiyatlarga ega bo'lmaslik, minimal ortiqchalik va maksimal turg'unlikni ta'miqlashni kafolatlaydi.

Munosabatlarni tekshirishda, atributlarni munosabatlarda to'g'ri guruhlanganiga asosiy e'tibor qaratiladi. Biz bunda normallashtirish nazariyasining konsepsiyalardan birida, atributlar munosabatlarga guruhlanganda, bu vazifa ular orasidagi mantiqiy aloqalarga nisbatan bajarilishi kerak, deb ta'kidlangan.

Bizning ushbu bosqichdagи vazifamiz yaratilgan har bir munosabatga normallashtirish qoidalarini qo'llagan holda tekshirishimiz kerak. Normallashtirish jarayoni quyidagi asosiy vazifalardan iborat:

- munosabatni birinchi normal shakl (1NSh)ga keltirish va undan atributlarning takrorlanuvchi guruhlarini olib tashlash;
- munosabatni ikkinchi normal shakl (2NSh)ga keltirish va undan birlamchi kalitga qisman bog'liq atributlarni olib tashlash;
- munosabatni uchinchi normal shakl (3NSh)ga keltirish va undan birlamchi kalitga tranzitiv bog'langan atributlarni olib tashlash;
- munosabatni Boys-Kodd normal shakl (BKNSh)ga keltirish va undan funksional bog'liqliklardagi qolgan nonnormalliklarni olib tashlash;
- munosabatni to'rtinchi normal shakl (4NSh)ga keltirish va undan ko'p ma'noli bog'liqliklarni olib tashlash;
- munosabatni beshinchi normal shakl (5NSh)ga keltirish va undan atributlar orasidagi birlashish bog'lanishlarini olib tashlash.

Talabalar o'quv loyihamizning mantiqiy modeli asosida yaratilgan munosabatlarni, shu bosqichlarning har birida tekshirsak, u holda ziddiyatlarga ega bo'laman, minimal ortiqchalik va maksimal turg'unligi kafolatlangan munosabatlarni olamiz.

## **Foydalanuvchi tranzaksiyalarini bajarishda munosabatlardan foydalanishni tekshirish**

Ushbu bosqichning asosiy maqsadi, foydalanuvchi tasavvurlarida nazarida tutilgan tranzaksiyalarini qo'llab-quvvatlash nuqtayi nazaridan lokal

mantiqiy berilganlar modelini tekshirishdan iborat. Tranzaksiyalarning ro'yxati foydalanuvchi tafsiridagi talablarda keltirilgan bo'ladi. Tranzaksiyalarni tekshirish 15-bobda keltirilgan edi va unda lokal konseptual berilgan modeli hamma tranzaksiyalarni qo'llab-quvvatlashini tekshirishi zarurligi aytib o'tgan edik. Bu erda esa ushu tranzaksiyalarni har bir munosabat ham qo'llab-quvvatlashini tekshirishimiz kerak. Bu esa munosabatlarni yaratish jarayonida yo'l qo'yilgan xatolarni to'g'rilash imkoniyatini beradi.

Buning uchun berilganlarga kirishdagi hamma zaruriy amallar, munosabatlar va ularni birlashtiruvchi birlamchi/tashqi kalitlarning aloqa chiziqlari, ER-diagrammalar va berilganlar lug'atlari orqali qo'lida bajarilishi kerak. Agarda, ushu usul bilan hamma talab qilingan tranzaksiyalarni tekshirishga muvaffaq bo'lsak, unda berilganlarning mantiqiy modelini tekshirish tugaydi. Mabodo, qaysidir tranzaksiyani qo'lida tekshira olmasak, demak, yaratilgan berilganlar modelsida xato borligini va uni bartaraf etish kerakligini bildiradi. Bu xato yangi yaratilgan munosabatga taalluqli bo'lish ehtimoli ko'proq, shuning uchun berilganlar modelidagi yangi munosabatga tegishli tranzaksiyalar bilan bog'liq bo'lgan aloqalarni tekshirish zarur.

### Butunlik chegaralarini aniqlash

*Berilganlarning butunlik chegarasini cheklashdan maqsad*, bazani bir-biriga zid bo'lgan berilganlarning paydo bo'lishidan saqlashdan iborat. Aniq bir BBB tizimlarida buni nazorat qiluvchi alohida funksiyalarning mavjudligi yoki mavjud emasligi biz uchun bu bosqichda muhim emas. Biz loyihalashning eng yuqori bosqichi bilan shug'ullanayotganimiz uchun, berilganlarning butunligini aniqlash masalasi majburiy talab bo'lib. hech qaysi aniq amalga oshirish vositalariga bog'liq emas. Butunlik chegaralari aniqlangandan so'ng, foydalanuvchi tasavvuridagi talablarga to'liq mos bo'lgan berilganlarning lokal mantiqiy modeli yaratiladi. Biz quyida besh xildagi berilganlarning butunligini cheklovchilarni ko'ramiz:

- majburiy berilganlar;
- atributlarning domenlarini cheklash;
- mohiyatlarning butunligi;
- havolali butunlik;
- predmet sohani cheklash.

#### Majburiy berilganlar

Ba'zi atributlar har doim kelishi mumkin bo'lgan qiymatlardan birini saqlashi zarur. Boshqacha aytganda, bu atributlar bo'sh qiymatlarga ega bo'lishi mumkin emas. Masalan, har bir talabaning albatta tug'ilgan sanasi

atributi bo'lishi kerak. Bu cheklash berilganlar lug'atiga atribut xususidagi ma'lumotlar kiritilganda, yozilishi shart.

### **Atributlarning domenlarini cheklash**

Har bir atribut, o'zining kelishi mumkin bo'lgan qiymatlarining maj-muasini saqlovchi, domenlardan iborat. Masalan, Jins atributi kelishi mumkin bo'lgan qiymatdan bittasini «E» (Erkak) yoki «A» (Ayol)ni saqlashi mumkin. Shuning uchun uning domeni ikkita satrдан iborat bo'lib, har birining uzunligi bitta belgili bo'ladi va u shu qiymatlarni saqlaydi. Ushbu cheklash berilganlarning modelida, atributlarning domenlarini aniqlash davrida o'rnatiladi.

### **Mohiyatlarning butunligi**

Ixtiyoriy mohiyatning birlamchi kaliti bo'sh qiymatga ega bo'lishi mumkin emas. Masalan, RO'YHAT munosabatining har bir satri birlamchi kalitning noyob qiymatidan iborat atribut qiymatini saqlaydi, bizning holda bu Zn atributi. Bunday cheklashlar har bir mohiyatning birlamchi kalitlarini aniqlash vaqtida hisobga olinishi kerak.

### **Havola butunligi**

Tashqi kalit avlod munosabatning har bir satrini ajdod munosabatdagi xuddi shunday qiymatga ega bo'lgan nomzod kalitning satri bilan bog'laydi. *Havola butunligi*, agarda tashqi kalit qaysidir qiymatga ega bo'lsa, shu qiymat albatta ajdod munosabatdagи satrlardan birida bo'lishi zarur, degan ma'noni anglatadi. Masalan, FAKULTETLAR mohiyatidagi Fak\_N tashqi kalitining qiymati 5 bo'lsa, RO'YHAT munosabatining Fak\_N atributidagi qiymatlardan biri albatta 5 bo'lishi zarur, aks holda ro'yxatdagи talaba mavjud bo'lmasan fakultetning talabasiga aylanadi. Shuning uchun tashqi kalitlar doimiy ravishda qiymatga ega bo'lishi shartmi, degan savol tug'iladi.

Umumiy holda, agarda avlod mohiyatning aloqa ishtirok etishi majburiy bo'lsa, u qiymatga ega bo'lishi shart. Agarda avlod mohiyatning aloqada ishtirok etishi majburiy bo'lmasa, u holda bo'sh qiymatga ega bo'lishi mumkin.

Keyingi muammo havola butunligini qo'llab-quvvatlashni tashkil etuv-chilar bilan bog'liq. Buni amalga oshirish mavjudlikni cheklashlar orqali, ya'ni nomzod yoki tashqi kalitlarning har bir qiymati kiritilishi, yangilanishi yoki olib tashlanishi bilan bog'liq shartlarni aniqlash yo'li bilan hal qilinadi. BBB tizimlarida ushu muammoni hal qilishda, jadvallar orasidagi bog'lanishlarni o'matishda, «kaskad» (CASCADE) usuli ishlataladi. Buning ma'nosi, agarda bir-biri bilan bog'langan bitta jadvaldagи birorta yozuv olib

tashlansa, berilganlarning butunligi buzilmasligi uchun, qolgan jadvallar-dagi shu tashqi kaitiga bog'liq hamma yozuvlar olib tashlanadi.

Xuddi shunday, agarda birlamchi kaitning qiymati ajdod munosabatda yangilansa, «kaskad» usul ishlatsilmagan bo'lsa, havola butunligi buziladi. Agarda «kaskad» usul ishlatalgan bo'lsa, ushbu munosabat bilan bog'langan har bir avlod munosabatlarda mos qiymatlarning yangilanishi amalga oshiriladi. Odadta yangilash amali, «kaskad» usul yo'li bilan amalga oshiriladi.

### **Predmet sohani cheklash**

Predmet sohadagi yoki tijorat qoidalariagi cheklashlarni ham tahlil qilish zarur. Masalan, inohiyatlarni cheklash korxonadagi qoidalarni bilan bog'liq bo'lishi, aniq tranzaksiyalarni amalga oshirish usullari berilganlarni yangilash bilan bog'langan bo'lishi mumkin.

### **Berilganlarning lokal mantiqiy modelini foydalanuvchilar bilan muhokama qilish**

Ushbu bosqichning maqsadi, yaratilgan berilganlarning lokal mantiqiy modeli va kuzatuv hujjatlari foydalanuvchi tasavvuridagi talablarni aniq akslantirgani va modelning tavsifi yoritilganiga ishonch hosil qilishdan iborat. Buning uchun oxirgi foydalanuvchilar bilan yana bir marta yaratilgan berilganlarning lokal mantiqiy modellarini muhokama qilish zarur. Agarda berilganlar batasi faqat bitta tasavvur uchun yaratiladigan bo'lsa, u holda bevosita berilganlar bazasini fizik loyihalash bosqichiga o'tish mumkin. Lekin tasavvurlar bir nechta bo'lsa va tasavvurni integratsiyalash yondashuvi qo'llanilsa, u holda quyida keltirilgan loyihalash uslubiyatining keyingi qadamlariga o'tish mumkin.

### **Berilganlarning global mantiqiy modelini yaratish va tekshirish**

Bu bosqichdagi maqsad, har biri alohida yaratilgan lokal mantiqiy berilganlar modellarini yagona berilganlarning global mantiqiy modeliga birlashtirishdan iborat. Berilganlarning lokal mantiqiy modeli yagona global modelga birlashtirilgandan keyin yana bir marta tekshirishga zarurat tug'ilsa, u holda normallashtirish va tranzaksiyalarni tekshirish yuqorida keltirilgan qoidalarni bo'yicha amalga oshiriladi. Lekin normallashtirish, faqatgina birlashtirish jarayonida munosabatlarga o'zgartirishlar kiritilgan bo'lsagina, amalga oshiriladi. Xuddi shunday, tranzaksiyalarni ham tekshirishda, faqatgina birlashtirish jarayonida modellarga kiritilgan o'zgartirishlar bilan bog'liq bo'lgan sohalari gina tekshiriladi. Katta tizimlarda

ushbu yondashuv qo'llanilsa, talab qilingan qayta tekshirish vaqtini anchagina qisqartiradi.

Ushbu bosqichda bajariladigan vazifalar quyidagilardan iborat:

- berilganlarning lokal mantiqiy modellarini berilganlarning yagona global modeliga jamlash;
- berilganlarning global mantiqiy modelini tekshirish;
- berilganlarning global mantiqiy modelini kelajakda kengaytirish imkoniyatlarini tekshirish;
- berilganlarning global mantiqiy modelini foydalanuvchilar bilan muhokama qilish.

### **Berilganlarning lokal mantiqiy modellarini berilganlarning yagona global modeliga jamlash**

Ushbu daqiqa gacha har bir lokal mantiqiy berilganlarning modellari uchun ER-diagrammalar, relatsion chizmalar, berilganlarning lug'atlari va cheklashlardan ham iborat kuzatuv hujjatlari tayyorlangan bo'lishi kerak. Ko'rsatilgan hamma tarkiblar nonormalliklarni va lokal mantiqiy berilganlarning modellarini orasidagi farqlarni aniqlash uchun ishlataladi, bu esa lokal mantiqiy modellarni jamlashga yordam beradi.

Agarda tasavvurlar soni ikkita yoki ucta bo'lsa, jamlash jarayonida uncha katta muammolar tug'ilmaydi, lekin katta tizimlarni global mantiqiy berilganlar modellarini jamlashda tizimli yondashuvga asoslanish kerak. Ana shunday yondashuvlardan biriga [78] biz ham asoslanamiz, boshqa turli yondoshuvlar [13,14,19] keltirilgan.

Biz asoslanadigan yondashuvimizda, quyidagi harakatlarni amalga oshirish tavsiya qilinadi:

- mohiyatlar/munosabatlarning nomlari, ularda saqlanuvchilarni va nomzod kalitlarni taxlil qilish;
- aloqalar/tashqi kalitlarning nomlari va ularda saqlanuvchilarni taxlil qilish;
- lokal berilganlar modeliga mos keluvchi mohiyat/munosabatlarni birlashtirish;
- faqatgina alohidagi berilganlar modeliga xos bo'lgan mohiyat/munosabatlarni kiritish;
- alohidagi lokal berilganlar modelidagi aloqalar/tashqi kalitlarni birlashtirish;
- faqatgina alohidagi berilganlar modeliga xos bo'lgan aloqalar/ tashqi kalitlarni kiritish;
- o'tkazib yuborilgan mohiyat/munosabatlari va aloqa/tashqi kalitlarni tekshirish;

- tashqi kalitlarni tekshirish;
- cheklash butunliklarini tekshirish;
- global ER-diagramma/munosabatlar chizmalarini shakllantirish;
- hujjatlarni yangilash.

Ba'zi harakatlarni amalga oshirishda «*mohiyatlar/munosabat*» va «*kaloga/tashqi kalit*» atamalarini keltirdik. Bu loyihalash jarayonida ER-diagrammani tekshirish kerakmi yoki ER-diagrammadan olingan munosabati tekshirish kerakmi, balki ikkalasini ham amalga oshirish yaxshi bo'lar, degan ma'moni beradi. Odatda, *munosabatlarning kompozit-siyalari asosida* tekshirish yaxshi natija beradi va bunda turli xildagi ER-diagrammalar orasidagi ko'pgina munosabatlar, atributlar, aloqalar orasidagi sintaktik va semantik xatolar to'g'rilanadi.

#### **Mohiyatlar/munosabatlarning nomlari, ularda saqlanuvchilarni va nomzod kalitlarni tahlil qilish**

Bu masalani hal qilish uchun har bir mohiyat/munosabatning nomlari va ularda saqlanuvchilar va nomzod kalitlar o'zaro taqqoslanadi, masalan jadvalga yozish tarzida. Nomzod kalitlarni taqqoslash natijasida o'xshash mohiyatlar/munosabatlarni aniqlash mumkin.

#### **Aloqalar/tashqi kalitlarning nomlari va ularda saqlanuvchilarni tahlil qilish**

Bu masalani hal qilish uchun ham aloqalar va tashqi kalitlarning nomlari har bir munosabatlar uchun lokal mantiqiy berilganlar modeli asosida jadvallarga to'plangan shaklda taqqoslanadi.

Bir nomga ega bo'lgan mohiyatlar yoki aloqalar «haqiqiy dunyo»da bir konsepsiyaga, turli nomlar turli konsepsiyalarga mos keladi. Taqqoslash jarayonida ana shu omillarga e'tibor berish kerak. Shuningdek, bitta tasavvurga tegishli mohiyatlar yoki aloqalar atributlar sifatida boshqa tasavvurlarda ishlataligan bo'lishi mumkin.

#### **Lokal berilganlar modeliga mos keluvchi mohiyat/munosabatlarni birlashtirish**

Birlashtirishga mo'ljalangan modellardagi mohiyatlar/munosabatlarni nomini va ularda saqlanuvchilarni tekshirishda, mohiyatlar/munosabatlar bitta «haqiqiy obyekt»ga mos kelishi va ulami birlashtirish mumkinligi aniqlanadi. Odatda, bu vazifani bajarish uchun quyidagi masalalar yechiladi:

- bir xil nom va bir xildagi birlamchi kalitga ega mohiyatlar/ munosabatlar birlashtiriladi;

- bir xil nom, lekin turli birlamchi kalitga ega mohiyatlar/munosabatlар birlashtiriladi;
- turli xildagi nomga, lekin bir xildagi yoki turli birlamchi kalitga ega mohiyatlar/munosabatlар birlashtiriladi.

Bir xil nom va bir xildagi birlamchi kalitga ega mohiyatlar/munosabatlар «haqiqiy dunyo»dagi bitta obyektiga mos keladi va shuning uchun birlashtirilishi kerak. Birlashtirilgan mohiyatlar/munosabatlар boshlang'ich mohiyatlar/munosabatlardagi takrorlangan nusxalardan tashqari hamma atributlardan tuziladi.

Ayrim hollarda, ikkita mohiyatlar/munosabatlар bir xildagi nom va nomzod kalitlarga, lekin turli birlamchi kalitlarga ega bo'lishi mumkin. Buriday vaziyatlarda ham mohiyatlar/munosabatlар birlashtirilishi kerak. Lekin birlamchi kalitlardan birini qoldirib, ikkinchisini muqobil kalitga aylantirish kerak.

Ba'zan mohiyatlar/munosabatlар turli nomga ega bo'lishi, lekin bajara-digan vazifasi aynan bir xil bo'lishi mumkin. O'xshash mohiyatlar/ munosabatlар quyidagicha ajratib olinishi mumkin:

- nomlarning o'xshashligidan;
- ulardagи saqlanuvlardan hamda birlamchi kalitlaridan;
- aniq aloqalardagi ishtirokidan.

#### **Faqatgina alohidagi berilganlar modeliga xos bo'lgan mohiyat/ munosabatlarni kiritish**

Avvalgi bosqichda, o'xshash obyektlarni tavsiflovchi hamma mohiyat/ munosabatlarni ajratish usulini ko'rib o'tdik. Qolgan hamma mohiyatlar global modelga hech qanday o'zgarishlarsiz kiritiladi.

#### **Alohidagi lokal berilganlar modelidagi aloqalar/tashqi kalitlarni birlashtirish**

Bu bosqichda berilganlar modelidagi aloqalar/tashqi kalitlarning nomi va tayinlanishi ko'rildi. Aloqalar/tashqi kalitlarni birlashtirishdan oldin aloqalar orasidagi ziddiyatlarni bartaraf qilish kerak, xususan karralik chegaralaridagi nomosliklarni. Bu bosqichda quyidagi harakatlar amalga oshiriladi:

- bir xil nomdagи va tayinlanishi bir xil bo'lgan aloqalar/tashqi kalitlar birlashtiriladi;
- turli nomdagи, lekin tayinlanishi bir xil bo'lgan aloqalar/ tashqi kalitlar birlashtiriladi.

## Faqatgina alohidagi berilganlar modeliga xos bo'lgan aloqalar/tashqi kalitlarni kiritish

Avvalgi masalani bajarishda bir xildagi aloqalar/tashqi kalitlar topilgan bo'lishi mumkin. Qolgan hamma aloqa/tashqi kalitlar global modelga o'zgarishsiz o'tkaziladi. 16.3-jadvalda Talabalar o'quv loyihamizning global mantiqiy berilganlar modeliga kiruvchi aloqalar va/yoki tashqi kalitlar keltirilgan.

16.3-jadval.

Munosabat	Kalit	Aloqa
RO'YHAT	Zn → BAHOLAR (Zn) Fak_N → FAKULTE-LAR(Fak_N) Yon_N → YO'NALISH(Yon_N)	Talaba_Baho Talaba_Fakultet Talaba_Yonalish
BAHOLAR	Zn → RO'YHAT (Zn) Fan_N → BAHOLAR (Fan_N)	Talaba_Baho
FAKULTE-LAR	Fak_N → RO'YHAT(Fak_N)	Talaba_Fakultet
YO'NALISH	Yon_N → RO'YHAT(Yon_N)	Talaba_Yonalish
FANLAR	Fan_N → BAHOLAR (Fan_N)	Baho_Fan

## O'tkazib yuborilgan mohiyat/munosabatlari va aloqa/tashqi kalitlarni tekshirish

Mantiqiy global modelni shakkantirishdagi eng murakkab masalalardan biri, mohiyat/munosabatlari va aloqa/tashqi kalitlardan qaysi biri hech qaysi lokal berilganlar modeliga kirmaganini aniqlashdan iborat. Agarda korxonada berilganlarning korporativ modeli bo'lsa, u holda hech qaysi lokal berilganlar modeliga kirmagan mohiyatlar va aloqalar ana shu korporativ model orqali aniqlanishi mumkin.

Yanaqdir o'tkazib yuborilgan mohiyat/munosabatlari va aloqa/tashqi kalitlarni aniqlash usuli, har bir mohiyat/munosabatning atributlarini o'rganish va boshqa lokal berilganlar modelidagi mohiyat/munosabatlari havolalarini tekshirishdan iborat. Buning natijasida shunday atribut aniqlanishi mumkinki, u qaysidir bir lokal berilganlar modelidagi mohiyat/munosabatga tegishli bo'lgan holda, boshqa lokal berilganlar modelidagi mohiyat/munosabatning birlamchi, muqobil yoki kalitsiz atributiga mos kelishi mumkin.

### Tashqi kalitlarni tekshirish

Bu bosqichda mohiyatlar/munosabatlari va aloqa/tashqi kalitlar birlashtirilishi, birlamchi kalitlar o'zgartirilishi va yangi aloqalar topilishi mumkin. Avlod munosabatlardagi tashqi kalitlar to'g'riligini tekshirish zarur va kerak bo'lsa o'zgartirishlar kiritish mumkin.

Bizning Talabalar o'quv loyihamiz bitta tasavvumning amalga oshirgani uchun 16.2-jadvalimiz o'zgarishsiz qoladi.

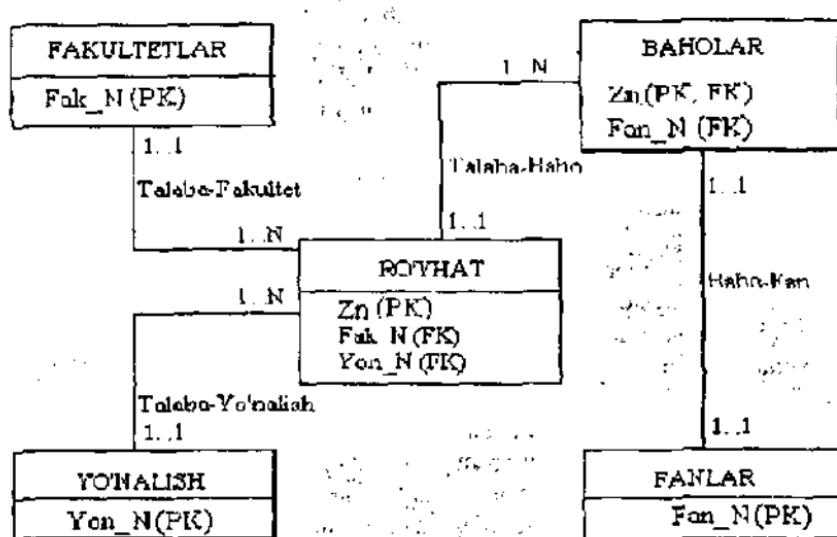
### Cheklash butunliklarini tekshirish

Bu bosqichda global mantiqiy berilganlar modelidagi butunlik chegaralari, boshlang'ich har bir tasavvurlar uchun qo'yilgan butunlik chegaralariga zid emasligi tekshiriladi. Agarda birlashtirish jarayonida qandaydir yangi aloqalar aniqlangan va yangi tashqi kalitlar yaratilgan bo'lsa, ularga mos havolali butunlik shartlari o'rnatilganini tekshirish zarur. Har qanday aniqlangan o'zgarishlarni foydalanuvchilar bilan kelishgan holda amalga oshirish kerak.

### Global ER-diagramma/munosabatlardan chizmalarini shakllantirish

Endi biz diagrammaning oxirgi naqlini shakllantirishimiz va unda hamma lokal mantiqiy berilganlar modellarini birlashtirishimiz mumkin. Agarda birlashtirish asosini munosabatlardan tashkil etgan bo'lsa, birlamchi va tashqi kalitlar ko'rsatilgan, tayyor bo'lgan diagramma *munosabatlarning global diagrammasi* deb ataladi. Agarda birlashtirish asosini lokal ER-diagrammalar tashkil qilgan bo'lsa, hosil bo'lgan diagramma *global ER-diagramma* deb ataladi.

Talabalar o'quv joyihamizdag'i *munosabatlarning global diagrammasi* 16.5-rasmida keltirilgan.



16.5-rasm.

### **Hujjatlarni yangilash**

Bu bosqichda hamma hujjatlarni berilganlarning global modelini yaratish jarayonidagi o'zgarishlarni hisobga olgan holda qayta ko'trib chiqish kerak. Hujjatlarning joriy berilganlar modeliga mos va har doim dolzarb bo'lishi juda muhimdir. Keyinchalik modelga o'zgartirishlar kiritilsa, ular bilan birga hujjatlarda ham bu o'zgarishlar akslanishi zarur. Eski ma'lumotlarning saqlanishi, keyinchalik tushunmovchiliklarga olib kelishi mumkin.

### **Berilganlarning global mantiqiy modelini tekshirish**

Global mantiqiy modelni tekshirish har bir lokal mantiqiy berilganlar modellarini tekshirish kabi tekshiriladi. Lekin bu vaziyatda modelning faqat o'zgartirish kiritilgan sohalari tekshiriladi. Bular modellarni birlashtirish jarayonidagi o'zgartirishlar bo'lishi mumkin.

### **Berilganlarning global mantiqiy modelini kelajakda kengaytirish imkoniyatlarini tekshirish**

Loyihalash jarayonidagi eng muhim jihatlardan bittasi global mantiqiy berilganlar modeliga kelajakda o'zgartirishlarni oson kiritishni ta'minlashdan iborat. Agar model faqat joriy talablarga javob bersa, uning yashash davri juda qisqa bo'ladi, yangi talablarga moslashtirish uchun ancha qayta ishlashga to'g'ri keladi. Shuning uchun yaratilgan model kengayuvchi bo'lishini ta'minlash va yangi talablarni amalga oshirish imkoniyatiga ega bo'lishi hamda foydalanuvchilar uchun minimal o'zgarishi juda muhimdir. Albatta bunday maqsadga erishish juda murakkab, chunki korxonada qanday o'zgarishlar bo'lishini hech kim bilmaydi.

Lekin ana shunday ma'lumotlar bo'lganda ham, oldindan bo'lajak talablarga tizimni moslash juda qimmatga aylanadi. Shuning uchun faqatgina kengaytirishning ayrim yo'naliishlarigina hisobga olinishi mumkin.

Shunday qilib, yaratilgan global mantiqiy berilganlar modelini minimal manfiy natijalar olinadigan tarzda kengaytirish qobiliyatini tekshirish zarur. Qanday vaziyat bo'lishidan qat'i nazar berilganlar modeliga, faqatgina foydalanuvchilarning talablariga muvofiq o'zgartirishlar kiritish mumkin.

## **Berilganlarning global mantiqiy modelini foydalanuvchilar bilan muhokama qilish**

Global mantiqiy berilganlar modeli tashkilotdagi berilganlarning tuzilishini aniq akslantirishini tekshirish, ushbu bosqichning maqsadidan iborat. Bungacha tashkilotdagi global mantiqiy berilganlar modelini yaratish jarayoni tugallangan hamda model to'liq va aniq bo'lishi kerak. Endi modelni va modelni tavsiflagan hujjatlar tashkilotdagi berilganlarning tuzilishini to'liq akslantirishiga ishonch hosil qilish uchun foydalanuvchilar bilan muhokama qilish zarur.

### **Xulosa**

*Berilganlar bazasini mantiqiy loyihalash tashkilotdagi ish tartibini axborot modelini yaratish jarayonidan iborat bo'lib, u aniq berilganlar modeli asosida bajariladi, lekin biron bir BBB tizimini va boshqa fizik chegaralarni hisobga olmagan holda ko'rildi.*

*Berilganlarning mantiqiy modeli ER-diagramma, relatsion chizma va modelni yaratish jarayonida ishlatiladigan berilganlar lug'atini o'z ichiga olgan kuzatuv hujjatlaridan iborat bo'ladi.*

*Relatsion turdag'i berilganlar bazasini mantiqiy loyihalashdagi asosiy bosqichlar – alohida foydalanuvchilarining tasavvuri asosida berilganlarning lokal mantiqiy modelini yaratish va tekshirish; hamda tashkilotdagi berilganlarning global mantiqiy modelini yaratish va tekshirishdan iborat.*

*Berilganlarning lokal mantiqiy modelini munosabatlarga aylantirish uchun hamma kuchli va kuchsiz mohiyatlarni munosabatlarga o'zgartirish kerak.*

*Berilganlarning havola butunligini qo'llash uchun mayjudlik chegarasi aniqlangan bo'lishi va unda nomzod yoki tashqi kalitlarni kiritish, yangilash yoki olib tashlash shartlari aniqlangan bo'lishi kerak.*

### **Nazorat uchun savollar**

1. Berilganlar bazasini mantiqiy loyihalash bosqichining vazifasi nimadan iborat?
2. Relatsion modelga bilan mos kelmaydigan tuzilishlarning turini tavsiflab bering va ularni konseptual modeldan olib tashlash yo'lini tushuntiring.
3. Kuchli mohiyatlarga mos munosabatlarni hosil qilish qoidasini tushuntirib bering.

- Zaif mohiyatlarga mos munosabatlarni hosil qilish qoidasini tushuntirib bering.
- Ikki tomonlama «birga bir» aloqalarga mos munosabatlarni hosil qilish qoidasini tushuntirib bering.
- Rekursiv «birga bir» aloqalarga mos munosabatlarni hosil qilish qoidasini tushuntirib bering.
- Ikki tomonlama «ko'plikdan ko'plikka» «birga bir» aloqalarga mos munosabatlarni hosil qilish qoidasini tushuntirib bering.
- Murakkab aloqalarga mos munosabatlarni hosil qilish qoidasini tushuntirib bering.
- Ko'p qiymatli atributlarga mos munosabatlarni hosil qilish qoidasini tushuntirib bering.
- Mantiqiy berilganlar modeli asosida olingan munosabatlarni va mantiqiy berilganlar modelini normallashtirish usullari bilan tekshirishni ta'riflab bering.
- Aniq tasavvur tafsiridagi talablarda ifodalangan tranzaksiyalarni mantiqiy berilganlar modeli qo'llab-quvvatlashi uchun qaysi yondashuvlardan foydalaniadi?
- Butunlikni cheklash qaysi maqsadda ishlatalidi?
- Butunlikni cheklashning beshta asosiy talablarini aytib bering.
- Agarda ajdod munosabatdagi olib tashlanishi kerak bo'lgan satrga, avlod munosabatdagi satrning havolasi bo'lsa, qanday muqobil strategiya ishlatalidi?
- Lokal mantiqiy berilganlar modelini global mantiqiy modelga birlashtirish uchun, qanday masalalar yechilishi kerak?

---

## 17-bob. Berilganlar bazasini fizik loyihalash

**Tayanch iboralar:** berilganlarni fizik loyihalash, *NULL*, *DDL* tili, hosilaviy atributlar, hisobli atributlar, *SQL*, *CREATE TABLE*, diskdag xotira, tizim zaxiralari, tezkor xotira, sahifali almashuv, svoping, prosessor, diskdagi kiritish-chiqarish, tarmoq, tranzaksiyalarni tahlil, ketma-ket fayllar, xeshlangan fayllar, xeshlangan maydon, xesh-kalitli maydon indeksi ketma-ket fayllar, muvozanallashgan takomil daraxtilar, klasterlar.

Berilganlarning global mantiqiy modelini loyihalashtirilgandan va 15- va 16 boblarda yaratilgan zaruriy kuzatuv hujjatlari tayyorlangandan so'ng, taklif eti layotgan loyihalash uslubiyatining fizik loyihalash qismiga o'tamiz. Tanlangan BBB tizimi muhitida, berilganlar bazasining mantiqiy loyihasini qanday qilib fizil loyihaiga aylantirish mumkinligi xususida batafsil tushuntirib o'tamiz. Berilganla bazasini fizik loyihalashning ko'pgina masalalari doimiy ravishda tanlangan BBB tizimining turiga bog'liq bo'lGANI uchun, ixtiyoriy berilganlar bazasi elementlarin amalga oshirishning bir nechta uslublari mayjud bo'lishi mumkin. Loyihach albatta tanlangan BBB tizimining hamma funksional imkoniyatlarini yaxshi bilishi hamda uning hamma afzallik va kamchilik tomonlarini tushunishi kerak chunki shundagina loyihachi berilganlar bazasini amalga oshirish usullari tanlashda asosli qaror qabul qilishi mumkin. Bundan tashqari, loyihachi berilgan lami joylashtirish va saqlashdagi optimal strategiyani tanlashni ham bilishi kerak.

### Berilganlar bazasini loyihalashning mantiqiy va fizik bosqichlarini taqqoslash

Berilganlarning global mantiqiy modelini loyihalashda, biz *nima* vazifalarni bajarish kerakligini o'rgandik, fizik loyihalash bosqichida esa bajarilgan vazifalarni *qanday* *amalga oshirish* kerakligini o'rganamiz. Loyiha chi kompyuter tizimida u yoki bu BBB tizimining faoliyatini aniq tasavvut eti olishi hamda maqsadli BBB tizimining funksional imkoniyatlarini yaxshi bilishi shart. Turli xildagi berilganlar bazalarining funksional imkoniyatlari bir-biridan juda ham katta farq qilgani uchun, fizik loyihalash ani tanlangan tizim bilan bog'liq. Lekin berilganlar bazasini fizik loyihalash bosqichi bir-biridan butunlay uzilgan deyish mumkin emas. Hamma va fizik, mantiqiy va ilovalarni yaratishda teskari aloqa mavjud. Masalar

tizimning unumtdorligini oshirish maqsadidagi fizik loyihalash bosqichida qabul qilingan qaror (xususan, munosabatlarni birlashtirish masalasida), berilganlarning mantiqiy modeli tuzilishiga ta'sir qilishi, bu esa o'z navbatida ilovalarni loyihalashga ta'sir qilishi mumkin.

### **Global mantiqiy berilganlar modelini BBB tizimi muhitiga ko'chirish**

Ushbu bosqichdagi asosiy maqsad, BBB tizimi muhitida amaiga oshirish mumkin bo'lgan, global mantiqiy relatson berilganlar modeli asosida tayanch funksional chizmani yaratishdan iborat. Berilganlarni fizik loyihalash bosqichidagi birinchi qadam, berilganlarni global mantiqiy loyihalash jarayonida yaratilgan munosabatlarni, BBB tizimida amalga oshirish mumkin bo'lgan shaklga keltirishdan iborat. Buning uchun birinchidan, berilganlar lug'atlari-dagi berilganlar bazasini mantiqiy loyihalash jarayonida to'plangan ma'lumotlarni tekshirish kerak. Ikkinchidan, shu ma'lumotlarni munosabatlarning asosiy loyihasini yaratishda foydalanish kerak. Bu jarayon BBB tizimi muhitining funksional imkoniyatlarini chuqur biliishni talab qiladi. Xususan, loyihachi quy'idagilarni biliishi talab qilinadi:

- asosiy munosabatlarni yaratish usullarini;
- tizim birlamchi, tashqi va muqobil kalitlarni qo'llab-quvvatlashini;
- tizim majburiy berilganlar tavsifini qo'llab-quvvatlashini, ya'ni atribut tavsifida uning NULL qiymatni olishi mumkin emasligini ko'rsatish;
- tizim domenlar tavsifini qo'llab-quvvatlashini;
- tizim butunlikni relatson cheklashni qo'llab-quvvatlashini;
- tizim predmet sohadagi cheklash tavsifini qo'llab-quvvatlashini.

### **Asosiy munosabatlarni loyihalash**

Fizik loyihalash jarayonini boshlash uchun, munosabatlarni xususidagi hamma ma'lumotlarni tahlil qilish va biliш talab qilinadi. Bu ma'lumotlar berilganlar lug'atida va DDL tilidagi munosabatlarni tavsiflash yozuvlarida bo'lishi mumkin. Berilganlarning global mantiqiy modelida aniqlangan munosabatlarning tavsisi quyidagi elementlarni o'z ichiga olgan bo'lishi mumkin:

- munosabat nomini;
- dumaloq qavsga olingan oddiy atributlar ro'yxatini;
- birlamchi (PK), tashqi (FK) va muqobil (AK) kalitlar tavsifini;
- hisobliy atributlar ro'yxatini va ularni hisoblash usullarining tavsifini;
- ixtiyoriy tashqi kalit uchun havolali butunlik talablarining tavsifini.

Har bir atribut uchun berilganlar lug'atida quyidagi ma'lumotlar bo'lishi mumkin:

- berilganlaming turi, atribut ichki tasavvurining o'chami va kelishi mumkin bo'lgan qiymatlar uchun ixtiyoriy talab qilinuvchi cheklashlar ko'rsatilgan o'z domenining tavsifi;
- oshkormas ravishda atribut olishi mumkin bo'lgan qiymat;
- ushbu atributning NULL qiymatni qabul qilishi mumkinligi.

Loyihaning asosiy munosabatlari yaratishda DDL tilidagi ifodalar ning kengaytirilgan formatlaridan foydalanamiz. Bu format orqali oshkormas va NULL qiymatni qabul qiluvchi domenlarni ko'rsatish mumkin. Misol sifatida, Talabalar o'quv loyihamizdagi RO'YHAT munosabatining tavsisini quyidagi 17.1-jadvalda keltiramiz.

### 17.1-jadval

Domain Zn:		fixed length character string,	length 8
Domain Familiya:		variable length character string,	length 15
Domain Ism:		variable length character string,	length 15
Domain O_Ism:		variable length character string,	length 15
Domain K_Sana:		date/time, in the range 00.00.0000	
Domain Fak_N:		fixed length character string,	length 2
Domain Yon_N:		integer Long	
Domain Kurs:		integer Byte	
Domain Guruh_N:		fixed length character string,	length 10
Domain Pasp_S:		fixed length character string,	length 2
Domain Pasp_N:		integer Long	
RO'YHAT			
Zn	Zn	NOT NULL	
Familiya	Familiya	NOT NULL	
Ism	Ism	NOT NULL	
O_Ism	O_Ism	NOT NULL	
K_Sana	K_Sana	NOT NULL	
Fak_N	Fak_N	NOT NULL	
Yon_N	Yon_N	NOT NULL	
Kurs	Kurs	NOT NULL	
Guruh_N	Guruh_N	NOT NULL	
Pasp_S	Pasp_S	NOT NULL	
Pasp_N	Pasp_N	NOT NULL	
PRIMARY KEY (Zn)			
FOREIGN KEY (Fak_N) REFERENCES FAKULTETLAR(Fak_N) ON UPDATE CASCADE ON DELETE SET NULL			
FOREIGN KEY (Yon_N) REFERENCES YONALISH(Yon_n) ON UPDATE CASCADE ON DELETE NO ACTION			
CANDIDATE KEY (Pasp_S) REFERENCES YONALISH(Yon_n) ON UPDATE CASCADE ON DELETE NO ACTION			
CANDIDATE KEY (Pasp_N) REFERENCES YONALISH(Yon_n) ON UPDATE CASCADE ON DELETE NO ACTION			

## **Hosilaviy berilganlarni olish usullarini yaratish**

Ushbu qadamidagi maqsad, berilganlarning global mantiqiy modeliga kiritilgan hosilaviy berilganlarni BBB tizimida taqdim etish usullarini aniqlashdan iborat. *Hosilaviy yoki hisobli atributlar* deganda, qiymati boshqa atributlarning qiymati yordamida aniqlanadigan atributlar tushuniladi. Masalan, quyidagilar hosilaviy atributlarga misol bo'ladi:

- aniq bir bo'limda ishlaydigan xizmatchilar soni;
- hamma xizmatchilar maoshining yig'indisi;
- talabaning aniq bir semestrдagi o'ttacha bahosi;
- guruhdagi talabalarning aniq bir semestrдagi o'ttacha bahosi.

Ba'zida hosilaviy atributlarni berilganlarning mantiqiy modelida emas, balki berilganlarning lug'atida tavsiflashadi. Agarda hosilaviy atribut mantiqiy modelga kiritilgan bo'lsa, u holda uni belgilash uchun atribut nomidan oldin qiya chiziqcha (/) qo'yilgan bo'ladi. Bu bosqichda hamma hosilaviy atributlarning ro'yxati tuziladi va hosilaviy atribut bazada saqlanadimi yoki zarurat tug'ilganda har gal qayta hisoblanadimi, degan savolga javob berilishi kerak. Loyihachi quyidagi holatlarni hisoblab, kerakli qarorni qabul qilishi zarur:

- hosilaviy berilganlarni saqlashga va ularni hisoblashda ishlatalgan haqiqiy berilganlar bilan muvofiqlashtirishga ketadigan qo'shimcha xarajatlar;
- agarda hosilaviy berilganlar zarurat tug'ilganda har gal qayta hisoblanadigan bo'lsa, bu jarayonga ketadigan xarajatlar.

*Unumdonlik talablarini hisobga olgan holda, bu ikki usulden arzonini tanlanadi.*

## **Predmet sohani cheklashni amalga oshirish**

Munosabatlardagi berilganlarni yangilash, predmet sohani cheklash bilan bog'liq bo'lib, yangilashga bog'liq tranzaksiyalar bajarilsa, yetarli bo'lishi mumkin. Predmet sohani cheklashni amalga oshirish BBB tizimining turi bilan bog'liq, chunki ayrim tizimlar boshqalariga nisbatan predmet sohani cheklashda ko'proq imkoniyatlar yaratadi. Agarda BBB tizimidagi SQL tili standart talablariga javob bersa, unda cheklashlarni amalga oshirish oson bajariladi. Masalan, xorijiy tillar fakultetiga tegishli har bir guruhdagi talabalar soni sakkiztadan oshmasligi kerak, degan qoida bor bo'lsin. Bu cheklashni SQL tilining CREATE TABLE yo'riqnomasi tarkibiga quyidagi konstruksiya yordamida kiritamiz:

**SONSTRAJNT**

**CHECK (NOT EXIST (SELECT RO'YHAT**

## FROM RO'YHAT.XORIJY\_TILLAR GROUP BY Guruh\_N HAVING COUNT (\*) > 8))

### Berilganlar bazasini fizik tasavvurini loyihalash

Berilganlar bazasini fizik loyihalashning eng asosiy maqsadlaridan biri berilganlarni saqlashni samarali tashkil etishdan iborat. Samaradorlikni baholash uchun bir nechta ko'rsatkichlardan foydalanish mumkin:

- *Tranzaksiyalarni bajarish unumdorligi.* Bu ko'rsatgich, berilgan vaqt oralig'ida ishlov berilgan tranzaksiyalarning sonini taqdim etadi. Ayrim tizimlarda, masalan aviachiptalarni zaxiralash xizmatida, tranzaksiyalarning bajarilishining yuqori unumdorligini ta'minlash butun tizimdan muvaffaqiyatli foydalanishdagi hal qiluvchi omil bo'lib hisoblanadi.
- *Javob vaqt.* Bitta tranzaksiyanı bajarish uchun ketgan vaqt oralig'ini ifodalaydi. Foydaluvchi nuqtasi nazaridan tizimning javob vaqtini minimallaştirish kerak. Lekin ayrim omillar tizim tezligiga ta'sir qiladi, lekin ulami yaratuvchilar nazorat qila olmaydi. Bularga misol, tizimning yuklanish darajasi yoki berilganlarni uzatishga ketadigan vaqt.
- *Diskdag'i xotira.* Bu ko'rsatkich berilganlar bazasining fayllarini joy uchun zarur bo'lgan xotira hajmi bilan bog'liq. Tizimni yaratuvchi foydalilanidigan diskning xotirasini kamaytirishga harakat qilishi kerak.

Lekin bu omillarning birortasi talabni to'liq qondira olmaydi. Shuning uchun, tizimni yaratuvchi shu omillar orasidan qoniqarli muvozanatga erishish uchun kelishtirish variantlarini izlaydi. Masalan, saqlanadigan berilganlar hajmini oshirish tizimning javob vaqtini ortishiga yoki tranzaksiyalarning bajarilish unumdorligini kamayishiga olib kelishi mumkin. Berilganlar bazasining boshlang'ich fizik loyihasini o'zgarmas narsa sifatida emas, balki tizimning samaradorlik darajasini baholash vositasi sifatida qarash kerak. Berilganlar bazasining boshlang'ich fizik loyihasi amalga oshirildigan keyin, tizim ko'rsatkichlarini kuzatib borish kerak va olingan natijalarga nisbatan ish ko'rsatkichlarini yaxshilash uchun uni sozlash kerak. Ko'pgina BBB tizimlari berilganlar bazasining administra-torlariga, tizimning faoliyati va uni sozlash bo'yicha turli maxsus programmalarni beradi.

### Tizim zaxiralari

Tizimning yuqori unumdorlikda ishlatish uchun, berilganlar bazasini fizik loyihasini yaratuvchi, apparat vositalarning quyidagi to'rtta asosiy

tarkiblari tizim unumdorligiga ta'sirini va qay tarzda o'zaro harakatlanishini bilishi surur:

- *Tezkor xotira.* Tezkor xotiradagi berilganlarga kirish, tashqi xotiradagi berilganlarga kirishdan bir necha yuz va ming marta tez bajarilishi mumkin. Umumiy holda, BBB tizimi va berilganlar baza-sining ilovalari uchun qanchalik katta tezkor xotiradan foydalanish imkoniyati yaratilsa, ilovalar shunchalik tez ishlaydi. Tajriba, tizimda doimiy ravishda tezkor xotiraning 5% bo'sh ushlab turish imkoniyatini yaratish juda foydali ekanligini ko'rsatadi. Lekin aksincha, tezkor xotirani 10% ortig'ini bo'sh ushlab turish esa, tizim samaradorligini pasaytirib yuborishi aniqlangan. Agarda hamma jarayonlarning ehtiyojini qondirish uchun tezkor xotira yetishmasa, operatsion tizim tezkor xotiraning bir qismini, ayrim jarayonlarning sahifalarini diskka yozib, bo'shatadi. Bu sahfalardagi berilganlarga murojaat bo'lsa, sahfalarni yana tezkor xotirani yuklaydi. Ana shu amal *svoping* yoki *sahifali almashuv* deb ataladi. Ayrim holatlarda tezkor xotirada ma'lum bo'sh hajmga ega bo'lish uchun, undagi sahafalar vaqtinchalik to'liqligicha diskka ko'chiriladi, keyin yana orqaga qaytariladi. Lekin svoping amalini bajarish darajasi juda yuqorilashib ketsa, bu tizimda tezkor xotira yetishmasligidan darak beradi.
- *Protsessor.* U tizimdagи boshqa apparat tarkiblarning faoliyatini boshqaradi va foydalanuvchi jarayonlariga ko'maklashib turadi. Protsessorning samarali ishlashining eng muhim sharti undan foydalanish uchun bo'ladijan raqobatchilikni oldini olishdan iborat bo'lib, bu odatda jarayonlarni kutish holatiga o'tkazish bilan kuzatiladi. Agarda operatsion tizim yoki foydalanuvchilarining programmalari protsessorga juda katta yulklama berayotgan bo'lsa, protsessor tizimning eng nozik qismiga aylanib qoladi. Bunday vaziyat sahfalari almashish jadalligi juda ortib ketganda yuzaga keladi.
- *Diskdagi kiritish-chiqarish.* Ixtiyoriy yetarli darajadagi BBB tizimlarida berilganlarni saqlash, tanlash va ishlov berish amallari diskdagi kiritish-chiqarish amallariga bog'liq. Odatda disklarni ishlab chiqaruvchilar kiritish-chiqarish amallarining bir sekunddagи sonini tavsiya qiladilar. Agarda haqiqatdagи ko'rsatgich bu sondan ortib ketsa, u holda disk tizimning nozik joyiga aylanadi. Diskdagi berilganlarning unumdorligiga berilganlarni saqlashni tashkil etish usuli muhim ahamiyat kasb etadi. Shuning uchun ham saqlanadigan berilganlarni disk vositalari orasida tekis taqsimlash taklif qilinadi. Ya'ni operatsion tizim fayllari BB fayllaridan ajratilgan, BB asosiy

fayllari indeksli fayllardan ajratilgan, tikelash jurnal berilganlar baza, sinning boshqa qismlaridan ajratilgan bo'lishi shart.

- *Tarmoq*. Tarmoqdagi trafiklar yoki tarmoqdagi manfaatlar to'qna-shuvi soni juda ortib ketsa, tarmoq tizimning nozik qismiga aylanishi mumkin.

Keltirilgan zaxira vositalarining har biri tizimdagi boshqa zaxiralarga ta'sir qilishi mumkin. Ma'lum bir zaxira ko'rsatkichlarini o'zgartirish tizim-dagi boshqa zaxira vositalarining ko'rsatkichlarini yaxshilashga olib kelishi mumkin. Masalan, tezkor xotira hajmini oshirish sahifali almashish jadalligining kamayishiga, bu esa protsessor yuklamasini kamayishiga olib kela-di. Tezkor xotiradan samali foydalanish esa, diskdag'i kiritish-chiqarish amallarini kamayishiga olib keladi.

Yuqorida keltirilgan vaziyatlarni hisobga olgan holda, berilganlar bazasini yaratishdagi harakatlarni tahlil etishga o'tamiz.

### **Tranzaksiyalar tahlili**

*Tranzaksiyalarni tahlil qilishdan maqsad*, loyihalashtiriladigan beril-ganlar bazasida bajariladigan tranzaksiyalarning funksional tavsiflari va ularning orasidan esa eng zarurlarini aniqlashdan iborat. Berilganlar bazasining yaratiladigan fizik loyihasi kerakli darajadagi samaradorlikka erishishi uchun berilganlar bazasida bajariladigan tranzaksiyalar va so'rovlar xususida eng maksimal ma'lumotlarga ega bo'lish talab qilinadi. Bizga son va sifat ko'rsatkichlari kerak bo'ladi. Har bir tranzaksiyani yaxshi loyiha-lash uchun quyidagilarni bilish talab qilinadi:

- tez-tez bajariladigan va samaradorlikka yetarlicha ta'sir ko'rsa-tuvchi tranzaksiyalarni;
- tashkilot uchun muhim ahamiyatga ega bo'lgan tranzaksiyalarni;
- bir sutka yoki hafta davomidagi berilganlar bazasining yuklamasi maksimal darajaga ko'tariladigan vaqt oraliqlarini.

Bu ma'lumotlar samarador bo'yicha muammolarini keltirib chiqarishi mumkin bo'lgan, berilganlar bazasining tarkiblarini aniqlashda ishlatalidi. Bundan tashqari, yuqori darajadagi tranzaksiyalardagi yangilash amallarida o'zgartiriladigan atributlari yoki so'rov natijasida qaytuvchi satrlar chegarasini belgilashda ishlataladigan mezonlarni aniqlash zarur. Bu ma'lumotlar fayllar tuzilishidagi eng kerakli yondashuvni va indekslarni yaratish uchun aniqlanadi.

Juda ko'p holatlarda hamma kelishi mumkin bo'lgan tranzaksiyalarni tahlil qilishning hech bir iloji yo'q, shuning uchun ham u yoki bu usul bilan eng zururlarini tanlash kerak. Tajribaga asoslangan qoida mavjud bo'lib,

qismida ba'zi bir arifmetik amallarni bajarishga asoslangan. Bunda belgili satrlar butun sonlarga aylantiriladi. Masalan, RO'YHAT munosabatining Pasp\_S atributidagi ikkala belgini raqamlarga aylantirib, keyin bunga Pasp\_N atributidagi raqamlarni qo'shish mumkin. Hosil bo'lgan yig'indi disk sahifasining adresi sifatida ishlataladi va unda berilgan yozuvlar saqlanadi.

Bu usulga muqobil bo'lgan eng sodda usul xeshlashni qoldiq bo'yicha bajarishdan iborat. Bunda MOD funksiyasidan foydalaniladi, funksiyaga maydonning qiymati uzatiladi va avvaldan berilgan butun songa bo'linadi, qoldiq esa diskdag'i adres uchun ishlataladi.

Ko'pchilik xesh-funksiyalarning kamchiligi, noyob adres hosil qilishni kafolatlama maydi, chunki xeshlangan maydonning kelishi mumkin bo'lgan qiymatlar soni, yozish mumkin bo'lgan adreslardan ancha ko'p bo'lishi mumkin.

Jadval uchun xotiraning tashkiliy tuzilishida xeshlangan fayldan foydalanish, xeshlash uchun ishlataligan qiymat bo'yicha tanlash maydonning aniq bir qiymati bo'yicha amaiga oshirilganda, ayniqsa, satrlarga kirish tasodifiy tarzda bo'lganda maqsadga muvofiq bo'ladi. Masalan, BAHOLAR jadvalining Ball atributi bo'yicha xeshlash, «Ball qiymati 77ga teng bo'lsin» kabi tanlashdan iborat bo'lsa, samarali bo'ladi. Xeshlangan fayllardan quyidagi holatlarda foydalanish tavsiya etilmaydi:

- jadvaldan satrlarni tanlash, xeshlangan maydon kalitining andozasini taqqoslash usuli bilan bajarilgan holatda. Masalan, familiyasi «A» dan (Familiya atributidan) boshlanadigan talabalar ro'yxatini aniqlansin;
- jadvaldan satrlarni tanlash, xeshlangan maydon qiymatiga kiruvchi maydonning berilgan oraliq qiymati, ustida bajarilgan holda. Masalan, BBBT fanidan bali 71 dan 95 gacha bo'lgan talabalar aniqlansin;
- jadvaldan satrlarni tanlash, xeshlangan maydon qiymatidan farqli qiymatda bajarilganda. Masalan, RO'YHAT munosabatini Zn maydoni (atributi) bo'yicha xeshlasak, hosil bo'lgan faylni Familiya atributi bo'yicha tez izlashga ishlatalib bo'lmaydi. Bu holatda satrga kirish uchun ketma-ket kirish yoki Familiya maydoni bo'yicha qo'shimcha indeks yaratish kerak;
- satrlarga kirishni faqat xeshlangan maydon qismi bo'yicha amalga oshirish kerak. Masalan, RO'YHAT jadvalini Pasp\_S va Pasp\_N atributlarining qiymatlari bo'yicha xeshlasak, u holda xeshlash mexanizmini satrni izlash uchun faqat Pasp\_S maydonining qiymati bo'yicha izlab bo'lmaydi;

- tez-tez xeshlangan maydonlarni yangilash bajariladi. Bunday amallar bajarilganda, BBB tizimida hamma satr olib tashlanishi va zarur bo'lganda, agarda xesh-funksiya yordamida yangi adres shakllangan bo'sha, yangi adres bo'yicha joylashtirilishi kerak. Shuning uchun, xeshlangan maydonlarni tez-tez yangilash samaradorlikning pasa-yishiga olib keladi.

### **Indeksli ketma-ket fayllar**

Xeshlangan fayllarga nisbatan indeksli ketma-ket fayllarning tuzilishi berilganlarni saqlash uchun ancha qulay. U berilganlarni kalitning aniq qiymatining mosligi, almashtirish andazasi, qiymatlar oralig'i va asosiy kalitning qismi bo'yicha tanlash usullarini qo'llab-quvvatlaydi. Lekin fayl indeksining tuzilishi fayl yaratilgan keyin o'zgarmaydi. Shuning uchun, indeksli ketma-ket fayllarga kirish, berilganlar o'zgargan sayin qiyinlashib boradi. Yangilashlar indeksli ketma-ket fayllarning kalitlarini buzilishiga ham olib kelishi mumkin, shuning uchun kalitlarning qiymatlaridan foydalanib, berilganlarni tanlash borgan sayin sekinlashib boradi.

### **Muvozanatlashgan takomil daraxtlar**

Xeshlangan fayllarga nisbatan muvozanatlashgan takomil daraxtlar (V'-Tree) usuli ham berilganlarni saqlash uchun ancha qulay. Bu usul berilganlarni, kalitli maydon qiymatining aniq mosligi, almashtirish andazasi, oraliqlar qiymati va berilgan kalitning qismi bo'yicha tanlash imkoniyatini yaratadi. Muvozanatlashgan takomil daraxtlar usulida indeks dinamik o'zgaruvchi bo'lib, jadvalning fayli kengaygan sari ortib boradi. Ana shu sababli, indeksli ketma-ket fayllarga nisbatan jadvaldagi berilganlar yangilanganda ham, ularga kirish samaradorligi kamaymaydi. V'-Tree tuzilishi fayllar kalit bo'yicha kirishda, hattoki berilganlar o'zgarganda ham, ularning tartibini saqlaydi. Agarda jadvaldagi ma'lumotlar tez-tez o'zgarib turmasa, bu usuldan foydalanish indeksli ketma-ket fayllarga nisbatan samarasizroq bo'lib qoladi. Sababi, indeksli ketma-ket fayllarning indeksi ketma-ket bo'lib, muvozanatlashgan takomil daraxtlar usulida esa ko'p darajalardan iborat, ya'ni tugun sahifalar fizik satrlarni emas, balki fizik satrlarga ko'rsatkichni saqlaydi.

### **Klasterlangan jadvallar**

Ba'zi bir BBB tizimlari, masalan Oracle, klasterlangan jadvallar bilan ishlaysdi. Fizik jihatdan birga saqlanuvchi bitta yoki bir nechta jadvallardan

tuzilgan guruhlarga *klaster* deb atildi. Ularda umumiy ustunlardan birgalikda foydalaniadi va ularga kirish ko'pincha bir vaqtida amalg'a oshiriladi. Agarda o'zaro bog'langan yozuvlar bir-biriga yaqin joylashgan bo'lsa, u holda diskka kirish tezligi ortadi. Jadvallarning o'zaro bog'langan ustunlari klasterda, *klaster kaliti* deb ataladi. Klaster kaliti saqatgina bish nusxada saqlanadi, shuning uchun klasterdag'i jadvallar majmuasi klasterlanmagan (alohida saqlanuvchi) jadvallarga nisbatan kam joy egallaydi.

### **Indekslarni qo'llashga tavsiyalar**

Shaxsiy kompyuterlardagi deyarli hamma BBB tizimida va xususat MS Access da doimiy fayl tuzilishi ishlataladi. BBBT MS Access indekslarni ishlataladi. Shuning uchun ham biz quyidagi tavsiyalarni BBBT MS Access ga nisbatan keltiramiz.

BBBT MS Access da birlamchi kalit avtomatik tarzda indekslanadi. Memo, Hyperlink yoki OLE Object turidagi berilganlar maydonlari indekslanmaydi. Boshqa turdag'i berilganlarning maydonlarini esa, agarda quyidagi shartlar bajarilsagina indekslash mumkin:

- maydondagi berilganlarning turlari Text, Currency yoki Date/Time bo'lsa;
- maydondagi qiymatlar izlanadi, deb faraz qilinadi;
- maydondagi qiymatlar saralanadi, deb faraz qilinadi;
- maydonda 'k' yoki 'w' qiymatlar saqlanadi, deb faraz qilinadi.

Agarda maydonda saqlanuvchi qiymatlarning asosiy qismi bir xilda bo'lsa, indeks so'rovlanming tez bajarilishiga yordam bera olmaydi.

Bundan tashqari, Microsoft korporatsiyasi quyidagilarni tavsiya etadi:

- ikkala birlashuvchi tomonlardagi maydonlarni indekslash yoki shunday maydonlar orasida aloqa o'rnatish kerakligini, chunki MS Access BBBT agarda hali indeks yaratilmagan bo'lsa, avtomatik tarzda tashqi kalit maydonida indeks yaratadi;
- birlashtirish bajariladigan maydondagi qiymatlar asosidagi yozuvlar guruhanganda, GROUP BY konstruksiyasini, shu jadvaldag'i agregatlash funksiyasi bajarilayotgan maydonga bergen kabi, berish kerak.

MS Access BBBT so'rovlanmi oddiy va murakkab predikatlar (MS Access da ifodalar deyiladi) orqali optimallashtirish imkoniyatini yaratadi. MS Access BBBT ayrim murakkab ifodalar turi uchun berilganlarga kirishning Rushmore texnologiyasini qo'llashga imkonini yaratadi. Bunda optimallashtirish ancha yuqori darajasiga erishiladi. Murakkab ifodalar oddiy ifodalarni AND yoki OR bul operatorlari bilan birlashtirish orqali shakllantiriladi. Masalan,

- a) (MUXANDIS.BO'LIM\_RAQAMI=721) AND (MUXANDIS.MAOSH>1500)
- b) (MA'LUMOT.STAJ>20) AND (OILA.FARZAND>3)

yoki ikkala ifodani birlashtirish natijasida yanada murakkab ifoda yaratiladi:

(MUXANDIS.BO'LIM\_RAQAMI=721) AND (MUXANDIS.MAOSH>1500) OR (MA'LUMOT.STAJ>20) AND (OILA.FARZAND>3)

MS Access BBB tizimida murakkab ifodalarni to'liq yoki qisman optimallash mumkin. Bu masala, ikkala sodda ifoda optimallashtiriladimi yoki bittasimi, sodda ifodalarni birlashtiruvchi operatorni hisobga olgan holda, hal qilinadi. Agarda quyidagi shartlar bajarilsa, u holda Rushmore texnologiyasi bo'yicha murakkab ifodani optimallashtirish mumkin:

- ikkita shartli ifodalarni birlashtirishda AND yoki OR operatorlari ishlatalgan bo'lsa;
- ikkala shart ham oddiy optimallashtiriluvchi ifodalardan iborat bo'lsa;
- ikkala ifoda indekslangan maydonlardan tuzilgan; maydonlar alohida indekslardan yoki bir nechta maydonlarni qamrab oluvchi tarkibli indeksdan tuzilgan bo'lsa.

### *Xulosa*

*Berilganlar bazasini fizik loyihalash shu bazani tashqi xotirada amalga oshirish bilan bog'liq bo'lgan jarayonni tavsiflashga tayyorqarlik ko'rish bilan bog'liq. Fizik loyihada asosiy munosabatlari, fayllarning tuzilishi va ularga samarali kirish usullari, butunlik chegaralari, berilganlarning xavfsizlik va himoya usullari tavsiflanishi zarur. Loyihachi maqsadli tanlangan BBB tizimining hamma imkoniyatlarini to'liq o'rganib chiqqanidan keyingina, asosiy munosabatlarni yaratishga kirishishi mumkin.*

*Berilganlar bazasini fizik loyihalashning birinchi bosqichi, global mantiqiy berilganlar bazasini, maqsadli tanlangan BBB tizimini muhitiga akslantirish masalasini amalga oshirish bilan bog'liq.*

*Ikkinci bosqich, berilganlar bazasidagi asosiy munosabatlarni saqlash uchun ishlataladigan, fayl tizimini va kirish usulini tanlashdan iborat. Buning uchun berilganlar bazasida bajariladigan tranzaksiyalarni tahlil qilish va shu tahlil natijasida eng maqbul bo'lgan fayllar tuzilishi va indekstar tanlanadi, so'ngra loyihada ko'zda tutilgan yechimlarni amalga oshirish uchun zarur bo'ladigan disk hajmi baholanadi.*

## Nazorat savollari

1. Fizik loyihalash bosqichlarida qanday vazifalar amalga oshiriladi?
2. Global mantiqiy berilganlar modelini BBB tizimi muhitiga ko'chishdan maqsad nimadan iborat?
3. Global mantiqiy berilganlar modelini loyihalash uchun nimalarni bilish talab qilinadi?
4. Asosiy munosabatlarni tavsiflashda nimalarga e'tibor berish kerak?
5. 17.1-jadval asosida Talabalar o'quv loyihamizdagi qolgan munosabatlarning tavsifini keltiring.
6. Hosilaviy atributlar, deganda nimani tushunasiz?
7. Predmet sohari cheklash, deganda nimani tushunasiz?
8. Berilganlarni fizik tasavvurini loyihalashdagi asosiy vazifalar nimalardan iborat?
9. Fizik loyihalash jarayonida tezkor xotiraga qanday talablar qo'yiladi?
10. Fizik loyihalash jarayonida protsessorga qanday talablar qo'yiladi?
11. Fizik loyihalash jarayonida diskka qanday talab qo'yiladi?
12. Fizik loyihalash jarayonida tarmoqqa qanday talablar qo'yiladi?
13. Fizik loyihalash jarayonida tranzaksiyalarni tahlil qilishdan maqsad nima?
14. «80/20» qoidasini izohlang.
15. Tranzaksiyalar va munosabatlarni bajarish yo'llari orasidagi chizmalarни tayyorlashga misol keltiring.
16. Tranzaksiyalarni bajarishda eng ko'p ishlataladigan munosabatlarni aniqlashdan maqsad nima?
17. Tranzaksiyalarni bajarishdagi berilganlarni ta'mil qilishdan maqsad nima?
18. Fayllarning tuzilishini tahlil qilishdan maqsad nima?
19. Ketma-ket tuzilishli fayllarni izohlang?
20. Xeshlangan fayllarni izohlang?
21. Indeksli ketma-ket fayllarni izohlang?
22. Muvozanatlashgan takomil daraxtlar usulidagi fayllarning tuzilishini izohlang?
23. Klasterlangan jadval, deganda nimani tushunasiz?
24. Oddiy predikatlarga misol keltiring.
25. Murakkab predikatlarga misol keltiring.

---

## **18-bob. Tranzaksiyalarni loyihalash**

**Tayanch iboralar:** tranzaksiya, tranzaksiyalarni loyihalash, tranzaksiyalar ishlatalidigan berilganlar, tranzaksiyalarning funksional tavsiflarni, tranzaksiya shakkantiradigan chiquvchi berilganlar, tantash tranzaksiyasi, o'zgartirish tranzaksiyasi, aralash tranzaksiya, COMMIT holat, ROLLBACK holat, tatomarlik holati, uyg'unlik holati, yakkatash holati, mustahkamlik holati, tranzaksiyalarni avtomatik bajarish modeli, boshqariluvchi tranzaksiyalarni bajarish modeli, takrorlanmas o'qish muammosi, fantom kiritish (tasavvurdagi satr) muammosi, tranzaksiyalarni alohida ko'rish (seriyalash) usuli, tranzaksiyalarni seriyalash, obyektlarni ushlab olish usuli, obyektlarni ozod qilish usuli.

### **Tranzaksiyalardan foydalanish maqsadi**

Tranzaksiyalar haqiqiy hayotdagi hodisalarни akslantiradi, masalan talabaning balini kiritish, yangi talabalarni qabul qilish yoki yangi ustozni ishga qabul qilish va h.k. hamma bu tranzaksiyalar berilganlar bazasiga murojaat etishi shart, chunki foydalanuvchilarning axborotlarga bo'lgan ehtiyojlarni aniq qondirish uchun, unda saqlanayotgan berilganlar doimiy ravishda haqiqiy hayotdagи voqeliklarga mos kelishi zarur.

Tranzaksiya bir nechta amallardan tashkil topishi mumkin, masalan bi hisobdan ikkinchi hisobga pul o'tkazishga o'xshashlari kabi. Lekin foydalanuvchi nuqtayi nazaridan, bu bitta yaxlit vazifadan iborat. BBB tizimin loyihalashtiruvchi nuqtayi nazaridan esa, har bir tranzaksiya berilganlar bazasini bir holatdan ikkinchi holatga o'tkazadi. BBBT hattoki berilganlar bazasida inqirozga uchragan vaziyatda ham, berilganlar bir-biriga zid bo'lmasligini ta'minlaydi. Bundan tashqari, BBBT berilganlar bazasiga tranzaksiya tomonidan kiritilgan hamma o'zgarishlar ishonchli tarzda saqlanadi. Bunda, birinchi tranzaksiya bajargan amallarda inqiroz natijasida xatoliklar paydo bo'ldi, ularni tuzatish uchun ikkinchi tranzaksiyani bajarish kerak deb fikrlashga ham asos yo'q. Agarda ayrim sababga ko'ra, tranzaksiya tugallanmay qolsa, BBBT hamma kiritilgan o'zgarishlar saqlanishini ham kafolatlaydi. Masalan, bankda hisobdan pul o'tkazish amaliyoti vaqtida, pu bir hisobdan olingan va ikkinchi hisobga kiritish vaqtida tranzaksiyada inqiroz yuz bersa, BBBT birinchi hisob raqamidagi amalni bekor qiladi.

Agarda debet va kredit amallarini boshqa-boshqa tranzaksiyalarga joylashsak, birinchi tranzaksiya o'z ishini tugatishi bilan debet hisobdag'i o'zgarishni bekor qilib bo'simaydi. Buning uchun, boshqa tranzaksiyani shu olingan pul miqdori bo'yicha kreditlashni amalga oshirishi uchun ishlatali kerak.

*Tranzaksiyalarni loyihalashdan maqsad*, hamma tranzaksiyalar tavsifini yuqori darajada antiqlash va hujjalashdan iborat bo'lib, yaratilayotgan berilganlar bazasida bajarilishini ta'minlanishdan, jumladan:

- tranzaksiyalar ishlatalidigan berilganlarni;
- tranzaksiyalarning funksional tavsiflarni;
- tranzaksiya shakllantiradigan chiquvchi berilganlarni;
- foydalanuvchi uchun tranzaksiyalarning zarurlik darajasini;
- foydalanishda vujudga kelishi mumkin bo'lgan jadallikni.

Bu ishlarning hammasini berilganlar bazasini loyihalash bosqichida manтиqiy berilgan modeli tranzaksiyalarni qo'llab-quvvatlashini hisobga olgan holda bajarilishi kerak.

Tranzaksiyalarning asosiy uch turi mavjud – tanlash, o'zgartirish va aralash tranzaksiyalar.

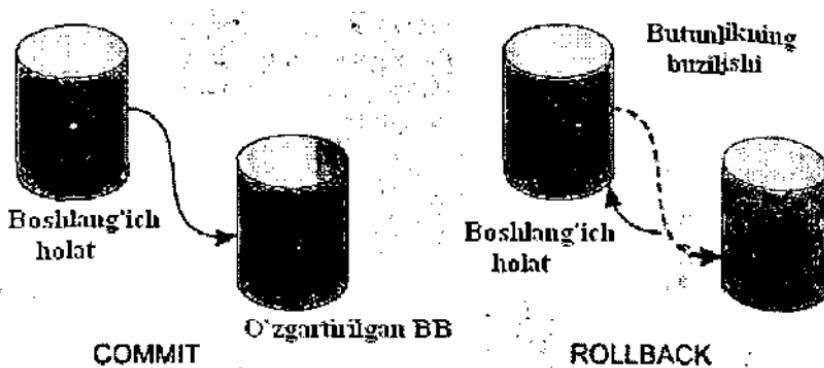
*Tanlash tranzaksiyasiga* ba'zi berilganlarni ekranda yoki hisobotda namoyon etish maqsadida ishlataladi. Tanlash tranzaksiyasiga misol, aniq bir talaba xususidagi ma'lumotlarni izlash va ularni ekranda akslantirishdan iborat.

*O'zgartirish tranzaksiyasiga* berilganlar bazasiga yangi yozuvlarni kiritish, kerak emaslarini olib tashlash yoki mavjud yozuvlarni esa yangilash uchun ishlataladi. Bunga misol, yangi talaba xususidagi ma'lumotlarni kiritish yoki talabaning yangi semestrdag'i fanlar bo'yicha ballarini kiritish.

*Aralash tranzaksiya* berilganlarni tanlash va o'zgartirishdagi hamma amallardan iborat bo'ladi.

### Tranzaksiyalar holatlari

*Tranzaksiya* – bu berilganlar bazasi ustida bajariladigan tugallangan harakatlar majmuasi bo'lib, bazani bir butun holatdan ikkinchi butun holatga o'tkazadi. Boshqacha qilib aytaksak, tranzaksiya berilganlar bazasiga kirish usullari belgilaydi. Agarda tranzaksiyaning operatorlari bajarilsa, u normal tugallanadi va berilganlar bazasi yangi butun holatga o'tadi (18.1-rasmidagi COMMIT holat). Tizimda inqiroz yuz bersa, berilganlar bazasi boshlang'ich holatga qaytadi (18.1-rasmidagi ROLLBACK holati).



18.1-rasm. Tranzaksiyani bajarilishi va qaytishi.

Tranzaksiyalarga **ACID** (Atomicity, Consistency, Isolation, Durability) nomi bilan yuritiluvchi talablar majmuasi qo'yildi.

**Atomarlik (atomicity) holati.** Tranzaksiya tugallangan harakatlar majmuasidan iborat. Tizim ularning bajarilishini «hammasi yoki hech narsa» tamoyili bo'yicha bajarilishini ta'minlaydi, ya'ni hamma harakatlar bajariladi va tranzaksiya bajarilgan hisoblanadi yoki tizimda inqiroz yuz bersa, tranzaksiya orqaga qaytadi va berilganlar bazasi boshlang'ich holatga qaytadi.

**Uyg'unlik (consistency) holati.** Tranzaksiyani bajarilishi natijasida BB bir aniq holatdan ikkinchi aniq holatga o'tadi, deb faraz qilinadi.

**Yakkalash (isolation) holati.** Tranzaksiyani bajarilish davomida berilganlar vaqtinchalik kelishilmagan holatga tushib qolishi mumkin. Bunday berilganlar boshqa tranzaksiyalarda o'zgarishlar bajarilguncha «ko'rinishi» mumkin emas. Tizim har bir tranzaksiyani alohida bajarilayotgani ta'minlayotgandek tuyuladi, ya'ni ba'zi tranzaksiyalar tugallan-gandan so'nggina qolganlari boshlanishini bildirayotgandek.

**Mustahkamlik (durability) holati.** Agarda tranzaksiya bajarilsa, uning natijalari mustahkam bo'ladi. Hamma obyektlarning yangi holati apparat yoki tizim inqirozi bo'lganda ham saqlanib qoladi.

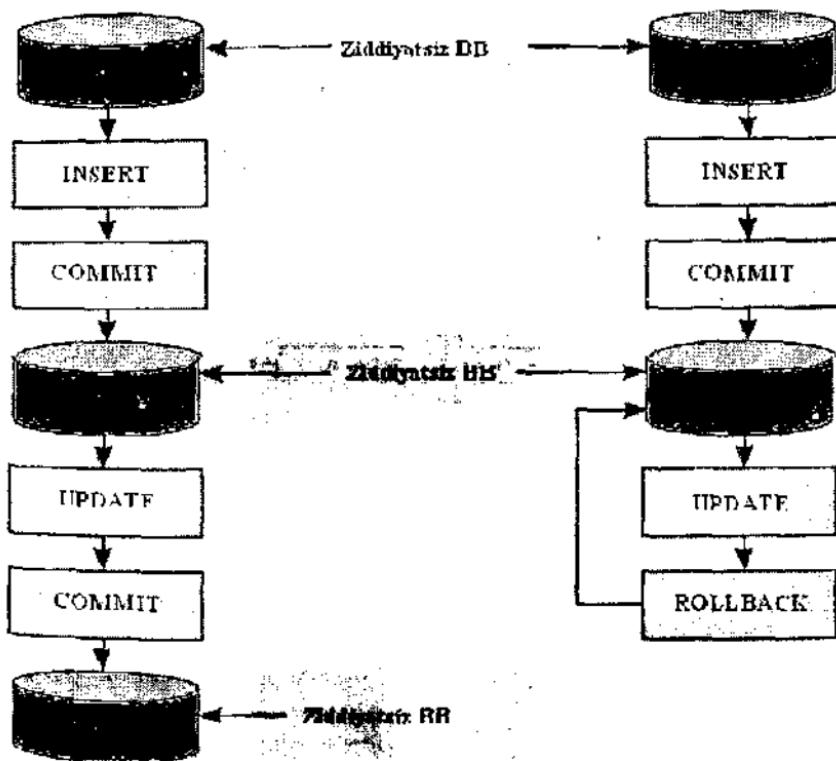
### Tranzaksiyalar modeli

Tranzaksiyalarning ko'pgina tijorat BBB tizimlarida mavjud bo'lgan ikkita modeli bilan tanishib chiqamiz. Bularidan birinchisi, tranzaksiyalarni avtomatik bajarish modeli COMMIT, ikkinchisi esa boshqariluvchi tranzaksiyalarni bajarish modeli ROLLBACK bo'lib, ikkalasi ham SQL tilining yo'riqnomalarini tashkil etadi.

*Tranzaksiyalarni avtomatik bajarish modeli.* Ushbu model DB2 BBB tizimida qabul qilingan chizma asosida yaratilgan. Tranzaksiya, foydalanuvchi yoki programma tomonidan birinchi yo'riqnomani, avtomatik tarzda bajarilishidan boshlanadi. Keyingi qadamlarda yo'riqnomalar ketma-ket, tranzaksiyalar tugamaguncha bajariladi (18.2-rasm).

Tranzaksiya quyidagi ikkita usuldan biri bilan tugallanadi:

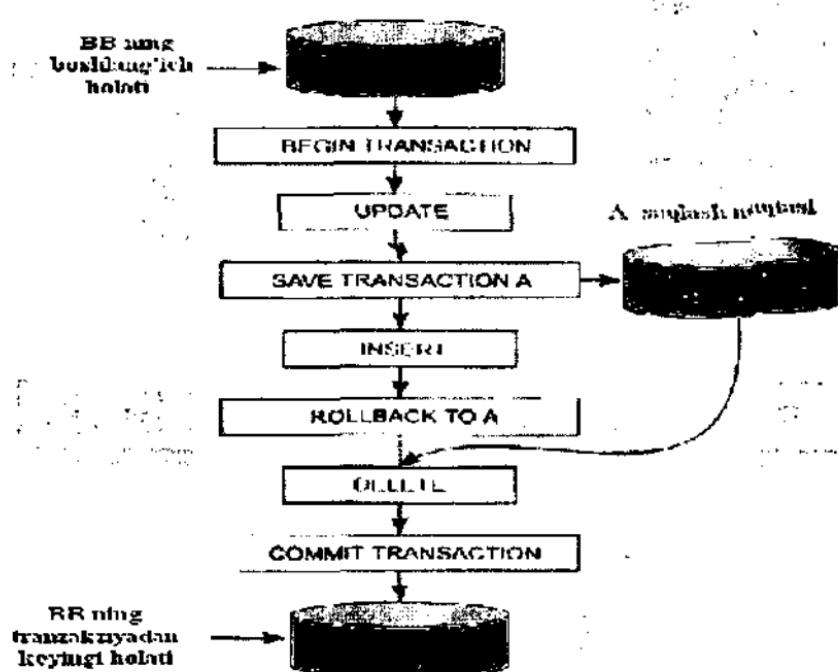
- tranzaksiyaning tugallovchi COMMIT yo'riqnomasi bo'yicha. Bunda BBga kiritilgan o'zgarishlar doimiyga aylanadi, yangi tranzaksiya esa darhol COMMIT yo'riqnomasidan keyin boshlanadi;
- joriy tranzaksiyani bajarilishini bekor qiluvchi ROLLBACK yo'riqnomasi bo'yicha. BB tranzaksiya boshlanmasdan avvalgi holatga qaytariladi, yangi tranzaksiya darholda ROLLBACK yo'riqnomasidan keyin boshlanadi.



18.2-rasm. Tranzaksiyalarning avtomatik bajarilish modeli.

*Boshqariluvchi tranzaksiyalarni bajarish modeli.* Ushbu model Sybase BBB tizimida ishlataladi. Bu tizimda Transact-SQL tilining shevalaridan biri ishlataladi. Buning uchun esa tranzaksiyalarga ishlov berishda to'rtta yo'riqnomadan foydalaniladi (18.3-rasin):

- BEGIN TRANSACTION yo'riqnomasi tranzaksiyani boshlanishi ni bildiradi (tranzaksiyaning boshlanishi oshkor ravishda betiladi);
- COMMIT TRANSACTION yo'riqnomasi tranzaksiyaning muvafqaqiyatli tugaganini bildiradi (bunda yangi tranzaksiya avtomatik tarzda boshlanmaydi);
- SAVE TRANSACTION yo'riqnomasi tranzaksiyani ichida saqlanish nuqtasini yaratadi va saqlangan holatga, yo'riqnomada ko'rsatilgan, saqlanish nuqtasining nomini beradi;
- ROLLBACK yo'riqnomasi joriy tranzaksiyani bajarishni bekor qiladi va BBni SAVE TRANSACTION yo'riqnomasi bajarilgan holatga (agarda yo'riqnomada saqlash nuqtasi – ROLLBACK TO saqlash nuqtasining nomi, ko'rsatilgan bo'sha) qaytaradi yoki tranzaksiyani boshlanishidan avvalgi holatga qaytariladi.



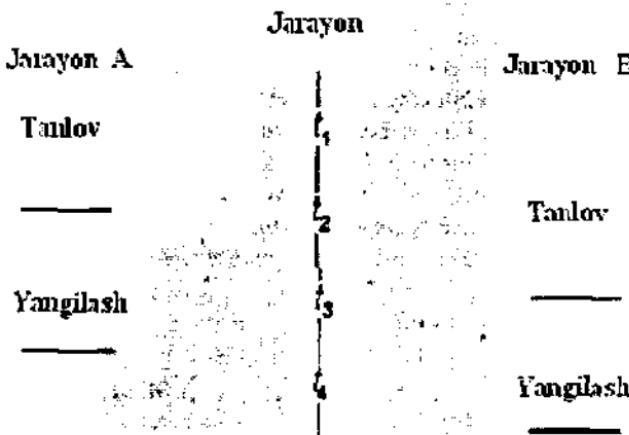
18.3-rasm. Boshqariluvchi tranzaksiyalarni bajarish modeli.

BB bilan bir nechta foydalanuvchilar birgalikda ishlaydigan vaziyatlar, BBB tizimi foydalanuvchilarning bir-biriga xalaqit bermasligini kafolatlashi kerak. Tranzaksiyalarga ishlov beruvchi vositalar foydalanuvchilarni bir-biridan yakkalash imkoniyatiga ega. Foydalanuvchilarni *yakkalashdagi birlik sifatida tranzaksiyalar olinadi*.

### **Tranzaksiyalarning parallel bajarishdag'i muammolar**

Tranzaksiyalarning parallel qanday bajarilishini tushunish uchun, berilganlar bilan parallel ishlashda yuzaga keladigan muammolarni ko'rib chiqamiz.

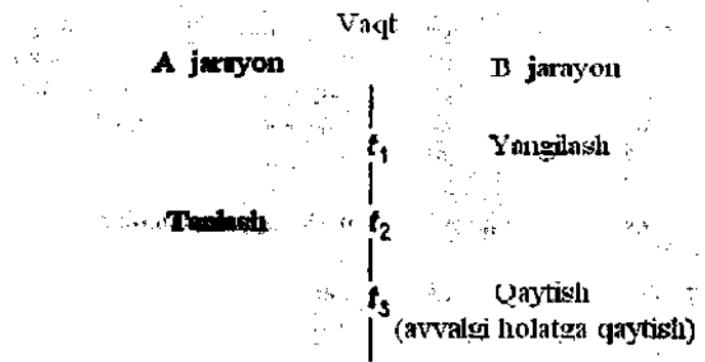
Yo'qotilgan yangilashlar muammosi shundan iboratki, agarda foydalanuvchilar parallel ravishda bir joydagi berilganlarni yangilayotgan bo'lsalar, oxirgi bajarilgan yangilanish saqlanadi. Qolganlari esa yo'qoladi (18.4-rasm).



Vaqtning  $t_3$  daqiqasidagi yangilanish yo'qoladi

18.4-rasm. Yo'qotilgan yangilash muammosi.

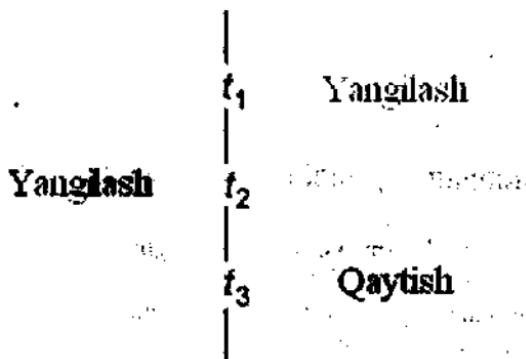
Qayd etilmagan yangilashlarga bog'liqlik muammosi shundan iboratki, A foydalanuvchi B foydalanuvchining berilganlar ustida bajargan yangilashlar natijasidagi berilganlarni, hali bu yangilashlar qayd etilmasidan avval ko'rishi mumkin. B foydalanuvchi BBni boshlang'ich holatiga qaytarishi mumkin (18.5-rasm).



18.5-rasm. Muddatidan oldin o'qish muammosi.

A foydalanuvchi xato berilganlar ustida amallarni bajarishga parakat qiladi. Bunday holatlar uchun ba'zan *vaqtidan avval o'qish* jadamasi ishlataladi.

Yana bir muammo shundan iboratki, parallel tranzaksiyani bajarish paytida, ishlov berilayotgan berilganlar to'plamining qaysidir elementlari shu paytda o'zgartirilayotgan, boshqalariga esa tuzatish bilan pog'liq bo'limgan guruhli ishlov berilayotgan (yig'indini hisoblash yok, o'rta qiymatni hisoblash) bo'lishi mumkin. Ba'zan, mavjud yozuvlarni o'zgartirish va yangi yozuvlarni kiritish vaziyatlari ajralgan holda ko'riladi. Birinchi muammo, *takrorlanmas o'qish muammosi* deb atalsa, ikkinchisi esa fantom kiritish (assevuradagi satr) muammosi deyiladi (18.6-rasm).



18.6-rasm. Fantom kiritish muammosi

Tranzaksiyalarning har birini alohida bajarishni ta'minlash uchun BBBT tranzaksiyalarni birga bajarishdagi maxsus usullardan foydalanishi zarur.

*Tranzaksiyalarni alohida ko'rish (seriyalash) usuli* – bu shunday rejalashtirish mexanizmidan iborat bo'lishi kerakki, bunda tranzaksiyalarni birgalikda bajarilishining natijasi xuddi shu tranzaksiyalarning ketma-ket bajarilishidagi natijasiga ekvivalent bo'lishi zarur. Bunday mexanizm bilan ta'minlash tranzaksiyalarni boshqarishdagi asosiy funksiya bo'lishi kerak. Tranzaksiyalarni alohida ko'rish foydalanuvchilarni alohida o'z ishlarini to'g'ri bajarishini ta'minlaydi.

Bu masalaning eng sodda yechimi, tranzaksiyalarni haqiqatan ham ketma-ket bajarilishini ta'minlashdan iborat. Lekin ayrim vaziyatlarda turli tranzaksiyalarning operatorlarini ixtiyoriy ketma-ketlikda bajarish ham mumkin bo'lib, bu vaziyatlar muammo tug'dirmaydi. Misol sifatida, faqatgina o'qish amalini bajaruvchi tranzaksiyalarni yoki BBning turli obyektlari bilan ishlovchi tranzaksiyalarni keltirish mumkin.

### Tranzaksiyalar orasidagi ziddiyatlar

Tranzaksiyalar orasida quyidagi ziddiyatlar yuzaga kelishi mumkin:

- 1- tranzaksiya o'zgartirayotgan obyekt bo'yicha ishini tugallamasdan, 2- tranzaksiyaning shu obyektni o'zgartirishga harakat qilishi (ziddiyat W-W, yozish-yozish);
- 1- tranzaksiya o'qiyotgan obyekt bo'yicha ishini tugallamasdan, 2- tranzaksiyaning shu obyektni o'zgartirishga harakat qilishi (ziddiyat R-W, o'qish-yozish);
- 1- tranzaksiya o'zgartirayotgan obyekt bo'yicha ishini tugallamasdan, 2- tranzaksiyaning shu obyektni o'qishga harakat qilishi (ziddiyat W-R, yozish-o'qish).

*Tranzaksiyalarni seriyalashni* ta'minlash uchun, *obyektlarni ushlab olish* va *ozod qilish usuli* tranzaksiya tashabbusi bilan amalga oshiriladi. Tranzaksiya obyektni ushlab oladi, natijada obyekt boshqa tranzaksiyalar uchun ularning domiga tushadi va uni faqat o'z ishini tugatgandan keyin qo'yib yuboradi.

Obyektni o'qish maqsadida ushlab olishni bir nechta tranzaksiyalar bir paytda bajarishi mumkin. Lekin, bir paytda obyektni o'qish maqsadida bitta tranzaksiyaning ushlab olishi, boshqa tranzaksiyaning esa shu obyektga yozish uchun ushlab olishini birgalikda amalga oshirish mumkin emas. Shuningdek, bir paytda obyektga yozish maqsadida bir nechta tranzaksiyalarning bitta obyektni ushlab olishiga ham ruxsat berilmaydi.

Shu asnoda, obyektlarni tranzaksiyalar tomonidan ushlab olishda asosiy ikkita tartib o'rnatilgan:

- S (Shared) – birqalikda bajariluvchi tartib, bunda o'qish amalini bajarish uchun obyekt alohida ushlab olinadi;
- X (eXclusive) – yakka bajariluvchi tartib, bunda obyekt bitta tranzaksiya tomonidan ushlab olinib, yozish, yo'qotish va o'zgartirish amallari bajariladi.

«Mijoz-server» arxitekturasiga asoslangan BBB tizimlarida eng keng tarqalgan yondashuv, BB obyektlarini ikki davrli qamalga olish bayonomasi shartlarini amalgaga oshirishdan iborat. BB obyekti ustida ixtiyoriv amalni bajarishdan oldin, tranzaksiya nomidan obyektni ushlab olishga birqalikda yoki yakka tartibda bajarishga ruxsat so'taladi. Tranzaksiyalarni bajarish ikkita qismga ajratiladi:

- o'sish davri – ushlab olishlarni to'plash (qamalni o'rnatish);
- qaytish yoki o'rnatish davri – ushlab olishdan ozod qilish (qamalni olib tashlash).

Agarda bir paytda bajarilayotgan tranzaksiyalar tegishli ushlab olishlarda – «Avval qamalga olingan obyekt ozod bo'limguncha, qaysidir tranzaksiya nomidan hech bir qamalga olish o'matilmasligi kerak», degan qoidaga amal qilsagina, tranzaksiyalarni amalgaga oshirish tizim tomonidan kafolatlanadi.

Relatsion berilganlar bazasida ushlab olish obyektlari quyidagilardan iborat bo'lishi mumkin:

- fayl – BB nuqtayi nazaridan fizik obyekt bo'lib, bir nechta munosabatlardan indekslarni saqlash joyi;
- jadval – mantiqiy obyekt bo'lib, berilgan munosabatga mos bo'lgan yozuvlar to'plamidan iborat;
- berilganlar sahifasi – fizik obyekt bo'lib, bir yoki bir nechta munosabatlardagi yozuvlarni, indeksli yoki xizmatchi ma'lumotlarni saqlaydi;
- yozuv – berilganlar bazasining sodda fizik obyekti.

Maydon darajasidagi qamalga olish amalga oshirilmagan, chunki bu qamalni boshqaruvchining ishini juda ko'paytirib yuboradi, bu esa o'z navbatida tizim unumdorligining kamayishiga olib keladi.

Qamal qilish va obyektni tanlash, qaysi amal bajarilayotgani, bunday amallarning qanchalik ko'pligi hamda ishlov berish vaqtining chegaralariga bog'liq. Masalan, yoppasiga faylga o'zgartirish kirilayotgan bo'lsa, u holda undagi har bir yozuvni qamalga olgandan ko'ra, uni alohida o'zini ochib ishlagan ma'qul.

Ushlab olish obyekti qanchalik katta bo'lsa, tizimda shunchalik ushlab olishlar soni kamayadi va vaqt ham kam sarf bo'ladi. Agarda, obyekt

sifatida sayl yoki munosabat ushlab olinsa, unda hattoki fantom kiritishlar muammosi ham hal bo'ldi.

Lekin katta obyektlarni ushlash vaqtida tranzaksiyalar orasida ziddiyatlar chiqish ehtimoli ko'payadi va parallel ishlash imkoniyati esa kamayadi. Ushlanishi kerak bo'lgan obyektni kattalashtirganimizda, biz vaziyatni atayin murakkablashtiramiz va ziddiyat bo'lishi mumkin bo'lmagan hollarda ularga duch kelamiz.

Ayrim BBB tizimlari qamal qilish chizmasining dinamik usulini qo'llaydi. Bu chizmaning algoritmi quyidagidan iborat. Tranzaksiya avval katta obyektni, masalan sahifani qamal qiladi. Agarda ushbu sahifadagi bирорта yozuvga davo qiluvchi tranzaksiya paydo bo'lsa, u holda birinchи tranzaksiya qamal qilish doirasini yozuv darajasigacha kamaytiradi.

### Xulosa

*Tranzaksiya – bu foydalanuvchi yoki amaliy programma tomonidan bajariladigan harakat yoki bir nechta keuna-ket harakatlardan iborat bo'lib, berilganlar bazasiga kirish va/yoki uni o'zgartirishga yo'naltirilgan. Tranzaksiya mantiqiy ish birligi bo'lib, berilganlar bazasini bir holatdan zid bo'lmagan ikkinchi holatga o'tkazadi.*

Tranzaksiya normal yoki ziddiyatlil tugashi mumkin. Agarda tranzaksiya bekor qilinsa, berilganlar bazasi o'zining awvalgi holatiga qaytariladi. Tranzaksiya parallel ishlash va tiklashni boshqarish birligidir.

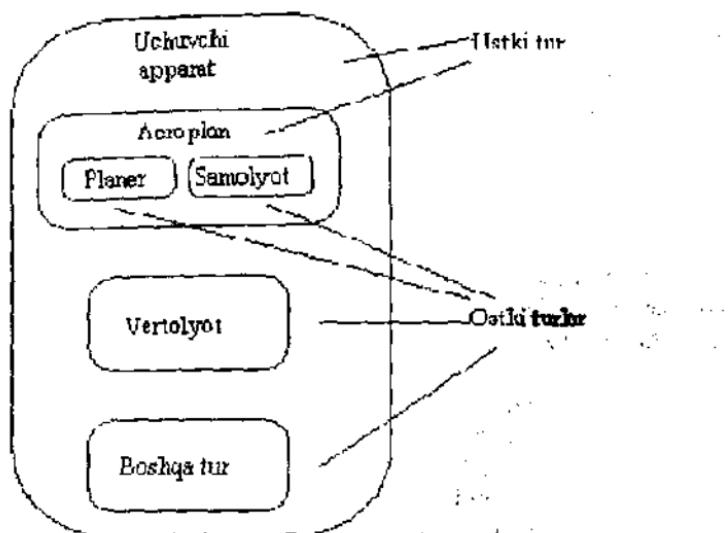
Tranzaksiya to'rtta asosiy xususiyatiga ega bo'lishi shart – atomarlik, uyg'unlik, yakkalash va mustahkamlik. Atomarlik va uyg'unlik holatlarini ta'minlash tiklash qism tizimiga yuklatilgan, yakkalash va ma'lum darajagacha mustahkamlikni ta'minlash parallel ishlashni boshqarish qism tizimiga yuklatilgan.

### Nazorat uchun savollar

1. Tranzaksiya nima?
2. Tranzaksiyadan foydalansidan maqsad nima?
3. Tranzaksiyalarning asosiy turlarini aytib bering.
4. Tanlash tranzaksiysi nima ish bajaradi?
5. O'zgartirish tranzaksiysi nima ish bajaradi?
6. Aralash tranzaksiyaga ikkita misol keltiring?
7. Tranzaksiyalar holati deganda nimani tushunasiz?
8. Tranzaksiyaning atomarlik holati deganda nimani tushunasiz?
9. Tranzaksiyaning uyg'unlik holati deganda nimani tushunasiz?
10. Tranzaksiyaning yakkalash holati deganda nimani tushunasiz?

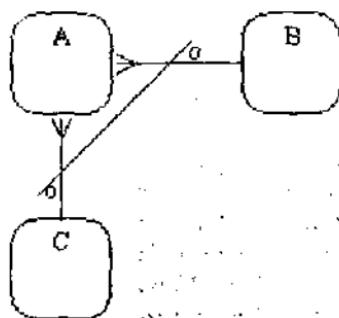
11. Tranzaksiyaning mustahkamlik holati deganda nimani tushunasiz?
12. Tranzaksiyalarning birinchi modeliga ta'rif bering.
13. Tranzaksiyalarning ikkinchi modeliga ta'rif bering.
14. Tranzaksiyalarni avtomatik bajarish modelini izohlang.
15. Boshqariluvchi tranzaksiyalar modelini izohlang.
16. Tranzaksiyalarning parallel bajarishda nechta asosiy muammolar tug'ilishi mumkin?
17. Yo'qotilgan yangilash muammosini izohlang.
18. Muddatidan oldin o'qish muammosini izohlang.
19. Fantom kiritish muammosini izohlang.
20. Tranzaksiyalar orasidagi ziddiyatlar necha xil?
21. Tranzaksiyalarni seriyalashdagi mohiyat nimadan iborat?
22. Obyektlarni tranzaksiyalar tomonidan ushlab olishda asosiy tartiblarni izohlang.
23. Relation berilganlar bazasida ushlab olish obyektlari sifatida nimalar kelishi mumkin?
24. BBB tizimlaridagi qamal qilish chizmasining dinamik usulidagi algoritmi aytib bering.

Ostki va ustki turlarni muqobil belgilash 19.8-rasmdagи kabi amalga oshiriladi. Bitta mohiyat o'xshash mohiyatlar guruhlarini umumiylashtiradi.



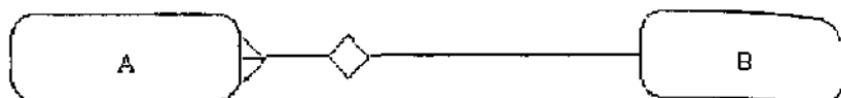
19.8-rasm.

O'zaro (bir-birini) yo'q qiluvchi aloqalarda, mohiyatning har bir nusxasi, bir-birini yo'q qiluvchi aloqalar guruhidagi faqat bitta aloqada qatnashishi mumkin. Bunday aloqa 19.9-rasmindagi kabi amalga oshiriladi.



19.9-rasm.

**Siljimas (non-transferrable) aloqalar:** mohiyatning nusxasini bitta aloqa nusxasidan ikkinchisiga ko'chirish mumkin emasligini tasvirlash 19.10-rasmdagi kabi belgilanadi.



19.10-rasm.

### IDEF1X uslubiyati

Axborot modelini loyihalashdagi eng muhim maqsad – bu mavjud bo'lgan o'r ganilayotgan predmet sohaning bir-biriga qarshi bo'l magan tuzilishlarini va bu tuzilishdagi tarkiblar orasidagi o'zaro harakatlarni yaratish. Berilganlarning konseptual modeli tushunchasi IDEF1X vositasida berilganlarni semantik modellash uslubiyati, ya'ni berilganlarni boshqa berilganlar bilan o'zaro aloqasining tasavvuridan iborat.

P. Chen yondashuviga asoslangan va T. Remey (T.Ramey) tomonidan yaratilgan IDEF1 usuli uchinchi normal shakldagi berilganlarning relatsion modeli talablariga mos javob beruvchi ekvivalent model qurish imkoniyatini yaratadi. Hozirgi paytda IDEF1 uslubiyatining takomillashgan naqli IDEF1X ishlatalmoqda. IDEF1X uslubiyati – «mohiyat-aloqa» konsepsiyasiga asoslangan berilganlarni semantik modellashdagi yondashuvlardan biri. Turli tabiatdagi tizimlarni axborot tuzilishini tahlil qilish vositasi. IDEF1X uslubiyati yordamida qurilgan axborot modeli, tizimdagi obyektlar xususidagi axborotlarning mantiqiy tuzilishini akslantiradi. IDEF1X uslubiyatida o'r ganishni soddalashtirish va avtomatizatsiyalashni amalga oshirish imkoniyatlari hisobga olingan. IDEF1X diagrammalari bir qancha keng tarqalgan CASE vositalar tarkibida ishlatalmoqda xususan, ERwin, ERStudio, Design/IDEF tizimlarida. Bu uslubiyat AQSH armiyasi uchun yaratilgan bo'lib, davlat muassasalari, moliya va ishlab chiqarish korporatsiyalarida keng ishlataladi. Shunday qilib, IDEF1X standartiga mos ravishda taqdim etilgan konseptual model, loyihalashtirilayotgan tizimdagi berilganlar bazasining mantiqiy chizmasidan iborat. IDEF1X konsepsiyasida *mohiyat* – bu modellashtirilayotgan tizimdagi alohida obyekt yoki voqeа bo'lib, xususiyatlarga ega bo'lgan to'plam - atributlardan iborat. Ushbu to'plamning alohidagi elementi «*mohiyatning nusxasi*» deyiladi.

IDEF1X standarti yordamida mohiyatlarning ikki turi – *bog'liq* va *bog'liqmas* hamda aloqalarning – *identifikator* va *identifikator bo'l maganlarini tasvirlash* mumkin. IDEF1X uslubiyatida, mohiyat identifikatorlarga bog'liq bo'lmaydi yoki mohiyatning har bir nusxasini uning boshqa moti-

yat bilan munosabatlari aniqlanmagan holatda bir qiymatli identifikasiyalash mumkin. Har bir mohiyat boshqa mohiyatlar bilan ixtiyoriy sondagi aloqalarga ega bo'lishi mumkin. Agarda mohiyatning har bir nusxasini, boshqa mohiyatlar bilan aloqasini aniqlanmagan holda, bir qiymatli identifikatsiyalash mumkin bo'lsa, u erkin (*mustaqil, bog'liqsiz*) mohiyat deyiladi.

Agarda mohiyatning nusxasini bir qiymatli identifikatsiyalash uning boshqa mohiyatlar bilan aloqasiga bog'liq bo'lsa, u *tobe (avlod) mohiyat* bo'ladi. Mohiyat uni yaratuvchi (ajdod) mohiyatdag'i aloqa orqali voris sifatida o'tgan atributlarga ega bo'lishi mumkin. Bu atributlar odatda *tashqi kalit* bo'ladi va mohiyatlar orasidagi aloqalarni tashkillashtirishda ishlatalidi. Agarda mohiyatning tashqi kaliti uning birlamchi kaliti (RK) yoki tarkibli birlamchi kalitning qismi sifatida ishlatilsa, u holda mohiyat *ajdod* mohiyatga *tobe (avlod)* bo'ladi. Masalan, Talabalar o'quv loyihamizda RO'YHAT mohiyati ajdod, BAHOLAR mohiyati esa avlod bo'ladi. Agarda tashqi kalit birlamchi kalit bo'lmasa va u birlamchi kalitning tarkibiy qismi bo'lmasa, u holda mohiyat ajdod mohiyatga nisbatan *erkin* bo'ladi.

Agarda mohiyat identifikatorlarga bog'liq bo'lsa yoki mohiyat nusxalarini bir qiymatli identifikatsiyalash boshqa mohiyat bilan o'zaro munosabatiga bog'liq bo'lsa, *identifikatorlarga bog'liq mohiyat* yoki *bog'liq mohiyat* deb ataladi. O'zaro bog'langan (*assotsiativ*) mohiyat deganda esa, ikita yoki undan ortiq mohiyat bilan bog'langan mohiyat tushuniladi. Har bir mohiyatga noyob nom va yassi chiziq («/») bilan ajratilgan mohiyat raqami yoziladi hamda blok ustida joylashtiriladi. Bu mohiyatlarni belgilashda IDEFIX uslubiyatida qabul qilingan yondashuvlar 19.11-rasmda keltirilgan.

### Bog'liq to'lmagan mohiyatlar

Mohiyat nomi/mohiyat raqami

X'matchi/25

### Bog'liq mohiyatlar

Mohiyat nomi/mohiyat raqami

Loyiha/013

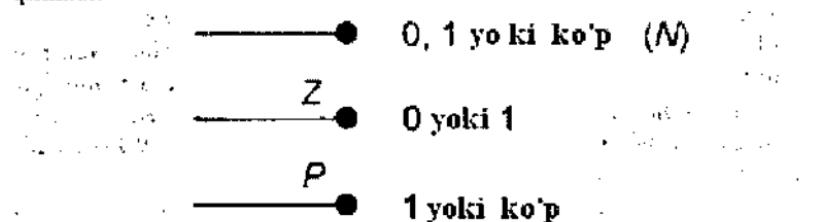
19.11-rasm.

Aloqa qo'shimcha ravishda daraja yoki quvvatni (*ajdod mohiyatning har bir nusxasi uchun mavjud bo'lgan avlod mohiyat nusxalarining soni*) ko'rsatish yo'li bilan aniqlanadi. IDEFIX uslubiyatida aloqalar quvvati quyidagicha bo'lishi mumkin:

- ajdod mohiyatning har bir nusxasiga avlod mohiyatlarning 0, 1 yoki ko'p nusxalarining mosligi;
- ajdod mohiyatning har bir nusxasiga avlod mohiyatlar 0 yoki 1 nusxasining mosligi;
- ajdod mohiyatning har bir nusxasiga avlod mohiyatning 1 yoki ko'p nusxasining mosligi;
- ajdod mohiyatning har bir nusxasiga avlod mohiyat cheklangan sondagi nusxasining mosligi (masalan, 4).

Agarda avlod mohiyat nusxasi o'zining ajdod mohiyati nusxasi bilan bir qiymatli aniqlansa, *aloqa identifikatsiyatangan*, aks holda *aloqa identifikatsiya qillinmagan* deb ataladi. Boshqacha aytganda, agarda mohiyat erkin bo'lsa, uning ajdod mohiyat bilan aloqasi *identifikator*, aks holda *identifikator bo'lmaydi*.

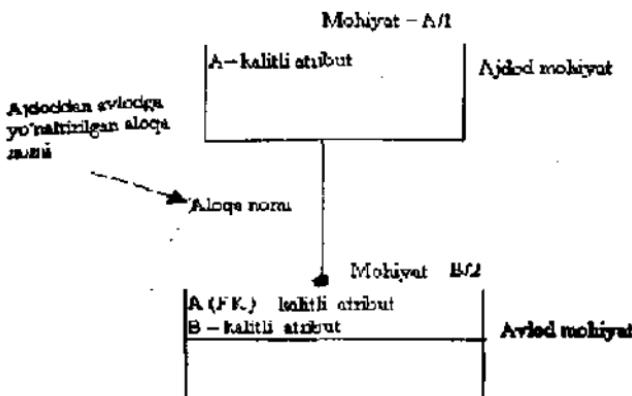
ER-diagrammada aloqa, ya'nini ajdod mohiyat va avlod mohiyat orasidagi aloqa chiziq orqali ifodalanadi. Aloqaning quvvati 19.12-rasmida kabi belgilanadi. *Oshkormas ravishdagi aloqa quvvati* Nga teng, deb qabul qilinadi.



19.12-rasm. Aloqa quvvati.

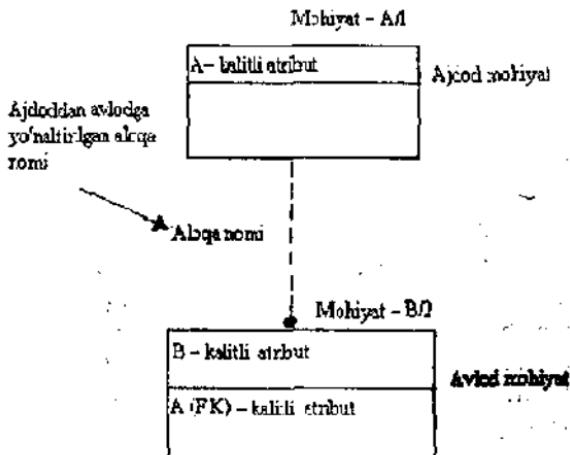
Aloqalarga nomlar qo'yildi. Aloqa quvvati ajdod mohiyat uchun avlod mohiyatning nechta nusxasi mavjudligini bildiradi. Aloqaning nomi har doim ajdod mohiyat nuqtayi nazaridan shakllanadi. Agarda *ajdod mohiyatning nomini, aloqaning nomini, quvvatning ifodasini va yaratilgan mohiyatning nomini ma'nosiga nisbatan birlashtirsak, to'g'ri gap hosil bo'ladi* (masalan, «ko'p TALABAlar – TEST – topshirmoqdalar»). Ajdod mohiyat va avlod mohiyatlar orasidagi *aloqani identifikatsiyalash uzlucksiz chiziq* orqali ifodalanadi (19.13-rasm). Bu aloqa tenglashtirilgan aloqa, deb ham yuritiladi. Avlod tomonidagi chiziqning oxirida nuqta qo'yildi. Ajdod mohiyatni belgilovchi aloqa o'zining identifikatoriga bog'liq bo'ladi. *Ajdod*

*mohiyatni belgilovchi aloqa erkin yoki o'zining identifikatoriga bog'liq bo'lishi ham mumkin, bu bog'liqlik boshqa mohiyatlar bilan aloqasi orqali aniqlanadi.*



19.13-rasm. Belgilovchi aloqa.

*Uzlukli chiziq identifikasiya qilinmaydigan aloqani bildiradi (19.14-rasm). Bu aloqani esa, tenglashtirilmaydigan aloqa deb ham yuritiladi. Identifikasiya qilinmaydigan aloqadagi ajdod mohiyat, qaysidir identifikatorli aloqada ajdod mohiyat sifatida kelmasa, u o'zining identifikatoriga bog'liq bo'lmaydi.*



19.14-rasm. Identifikasiya qilinmaydigan aloqa.

Atributlar mohiyat bloki ichida ro'yxat tarzida ifodalanadi. Birlamchi kalitni aniqlovchi atributlar ro'yxatni boshida keladi va qolgan atributlardan, gorizontal chiziq orqali ajratiladi (19.15-rasm).

### Mohiyat\_nomi/Mohiyat\_raqami

<b>atribut_nomi</b>
[atribut_nomi]
[atribut_nomi]
[atribut_nomi]
[atribut_nomi]

*Birlamchi kalit atributlari*

19.15-rasm. Atributlar va birlamchi kalitlar.

Mohiyatlar tashqi kalitlarga (Foreign Key) ham ega bo'lishi, tashqi kalitlar esa birlamchi kalitning tarkibida yoki birlamchi kalit sifatida yoki kalit bo'lмаган atribut sifatida ishlatalishi mumkin. Tashqi kalit mohiyatlar blokidagi atributlar nomlarining ro'yxatida keltiriladi va qavsdas FK harflari yoziladi (19.16-rasm).

Kalit bo'lмаган atributning  
tashqi kalit sifatida kelishi

Birlamchi kalit bo'lgan atributning  
kalit sifatida kelishi

Dalloi /12

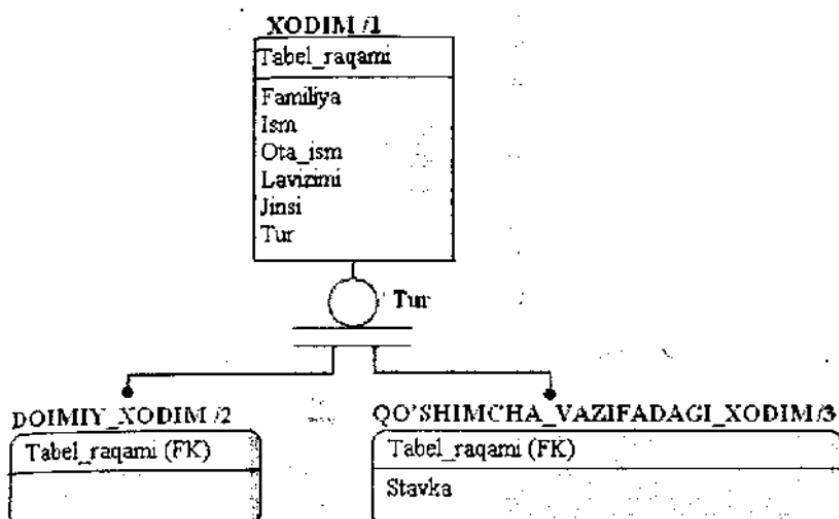
<b>Dalloi_raqami</b>
Bo'lim_raqami (FK)

Sotib\_olishga\_byyurtma/5

<b>Byyurtma_raqami (FK)</b>
Maxsulot_raqami

19.16-rasm.

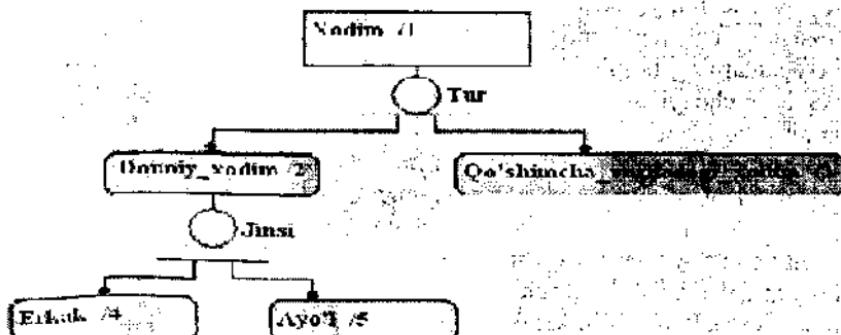
*Vorislik iyerarxiyasi*, umumiy tavsiqga ega bo'lgan mohiyatlarni birlashtirishdagi alohida turdan iborat. Masalan, doimiy xodimlar va qo'shimcha vazifada ishlochilarining xossalaridan umumlashgan (ajdod) XODIM mohiyatini shakillantirish mumkin. Maxsus ma'lumotlar tabaqalashgan DOIMIY\_XODIM va QO'SHIMCHA\_VAZIFADAGI\_XODIM avlod mohiyatlarda saqlanishi mumkin (19.17-rasm).



19.17-rasm.

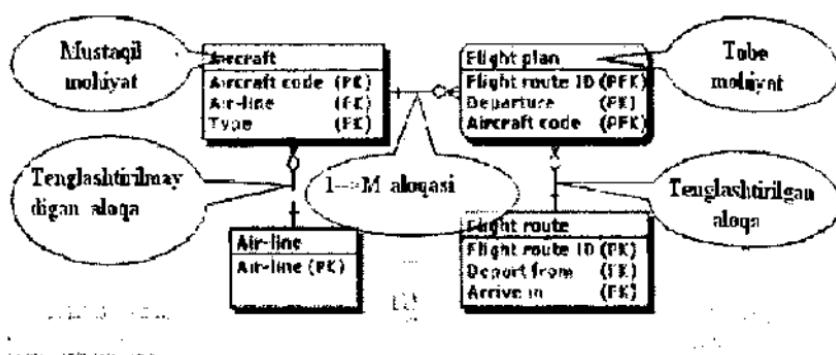
*Diskriminatör* – bu ajdoddagi maxsus atribut bo‘lib, bitta tabaqadagi mohiyatni boshqa tabaqadagi mohiyatdan ajratib ko‘rsatadi. Keltirilgan misolda, bu *tur*.

Agarda ajdodning avlodda o‘ziga o‘xshagan nusxalari mavjud bo‘lmasa, tabaqa to‘liq bo‘lmaydi. Masalan, xodim maslahatchi bo‘lishi mumkin, lekin u mohiyat sifatida iyerarxiyaga kiritilmagan bo‘lishi mumkin. To‘liq va to‘liq bo‘lmagan tabaqalar birgalikda kelishi ham mumkin (19.18-rasm).



19.18-rasm.

19.19-rasmida IDEF1X standarti bo'yicha mohiyat va aloqalarni tasvirlashga misol keltirilgan. Tashqi kalitning birlamchi kalit sifatida ishlatalishgi PFK tarzida qavsda ko'rsatiladi.



19.19-rasm.

## DATARUN uslubiyati

Berilganlarni modellashtirishni jahonda eng ko'p tarqalgan elektron uslubiyatlardan bittasi bu DATARUN [20] uslubiyatidir. Bu uslubiyatga asosan programma ta'minotining yashash sikli, ISO 12207 standartlari bo'yicha aniqlangan, asosiy jarayonlarni bajarish natijalarini bilan bog'langan. Jarayonning har bir bosqichi bajarilgan ish natijalaridan tashqari, keyingi bosqichni bajarish uchun ish rejasini tayyorlash bilan tugallanadi. Talablarni shakllantirish va rejalashtirish bosqichi o'z ichiga, loyihaning boshlang'ich hajmini baholash va narxlarini aniqlashni ham oladi. Axborot tizimini yaratish uchun talablar va iqtisodiy asoslar shakllantirilishi kerak, loyihani texnik jihatdan amalga oshirishni baholash uchun asosni yaratib beruvchi funksional modellar (tashkilot tijorat jarayonining modeli) va berilganlarning boshlang'ich konseptual modellari shakllantirilishi kerak. Bu bosqichdagi asosiy natijalar tashkilot faoliyatining modeli (jarayonlarning boshlang'ich modellari va tashkilotdagi boshlang'ich berilganlar), faoliyat yuritayotgan axborot tizimlarini birlashtirish va boshlang'ich tijorat rejaga talablarini o'z ichiga olgan, tizimga umumiy talablar shakllantirilishi kerak. Konseptual loyihalash bosqichida, birinchidan boshlang'ich berilganlar puxta va har tomonlama tahlil qilinishi, ikkinchidan berilganlarning konseptual modeli aniqlashtirilishi, ana shundan keyin tizimning arxitekturasi loyihalashtirilishi kerak. Tizimning arxitekturasi, konseptual modelni ishtiroy etishi mumkin bo'lgan ostki modellarga ajratishni nazarda tutadi. Mayjud bo'lgan axborot tizimlaridan foydalanish imkoniyati baholanadi va

ularni o'zgartirish usullari tanlanadi. Loyiha qurilgandan keyin boshlang'ich tijorat reja aniqlashtiriladi. Bu bosqichdag'i natijalar – berilganlarni konseptual modeli, tizimning arxitekturasи va aniqlashtirilgan tijorat reja bo'ladi.

Ilovalarni tafsirlash bosqichida loyihani yaratish va qismlarga ajratish davom ettiladi. Berilganlarning konseptual modeli berilganlarning relation modeliga aksantiriladi. Ilovalarning tuzilishi, ekranlar shaklidagi zaruriy aloqalar, hisobottar va paket jarayonlari ularni chaqirish martig'i bilan birga aniqlanadi. Berilganlar modeli tijorat reja va har bir jadval uchun zaruriy usullar orqali aniqlashtiriladi. Bu bosqichning oxirida ilovalarni amalga oshirish uslubi bo'yicha qaror qabul qilinadi.

Bosqich natijalariga nisbatan axborot tiziminining loyihasi quriladi va u o'z ichiga:

- axborot tizimining arxitekturasini;
- berilganlarni va funksiyalarni;
- tashqi tizimlarni;
- foydalanuvchilar bilan bo'ladigan aloqalarni;
- yaratiladigan ilovalarga talablarni (berilganlar modeli, aloqalar va funksiyalar);
- mavjud axborot tizimlarini takomillashtirishga qo'yiladigan talablarni;
- ilovalarning integratsiya qilinish talablarini olishi hamda axborot tizimi rejasining oxirgi naqli shakllantiriladi.

Yaratish, integratsiyalash va testlash bosqichlarida testlash uchun berilganlar bazasi, xususiy va majmuaviy testlar yaratilishi kerak. Loyihaga asoslangan holda, berilganlar bazasi va ilovalar yaratiladi, ular turlarga ajratiladi va testdan o'tkaziladi. Mavjud axborot tizimlari bilan aloqalar o'tnatiladi va tekshiriladi. Programma ta'minotining joriy naqli tavsiflanadi. Test natijalari bo'yicha berilganlar bazasi va ilovalar optimallashtiriladi. Ilovalar tizimga kiritiladi, tizimning tarkibidan ilovalar testdan o'tkaziladi va tizim sinovdan o'tkaziladi. Bosqichning asosiy natijalari bu – tizim tarkibidagi majmuaviy testdan o'tgan tayyor ilovalar, programma ta'minotining joriy tavsifi, tizimning sinov natijalari bo'yicha o'zgartirilgan naqli va tizimdan foydalanish uchun tayyorlangan hujjatlardan iborat bo'ladi.

Tizinni joriy qilish – uni o'matish, berilganlar bazasini va ilovalarni joriy qilish harakatlardan iborat. Asosiy natijalar, joriy qilishga tayyor va buyurtmaching programmin-apparat platformasiga o'tkazilgan tizim, kuza-tuv hujjatlari va tajriba sinovdan o'tkazilganligi xususidagi qabul qilish natijalarining dalolatnomasi.

Rivojanish bosqichi, deyarli ishlab chiqish bosqichini qayta takror-lashdan iborat.

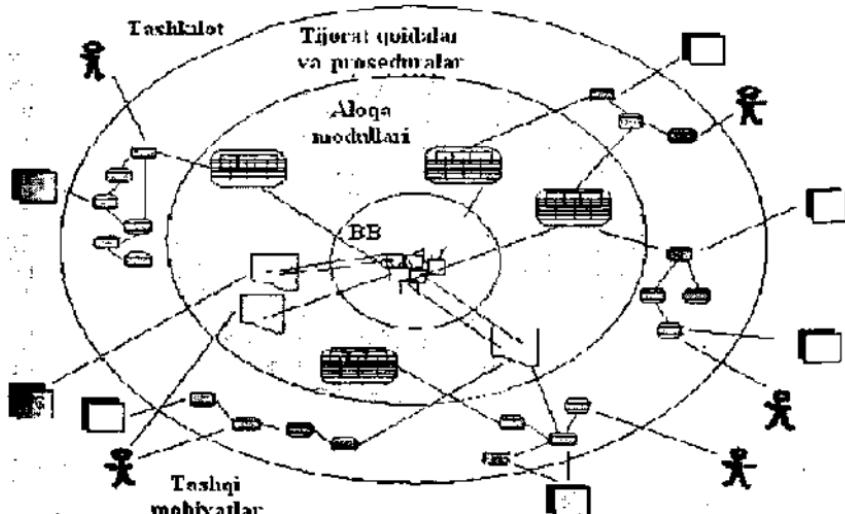
DATARUN uslubiyati ikkita model yoki ikkita tasavvurga tayanadi:

- tashkilot modeli;
- axborot tizimining modeli.

DATARUN uslubiyati tashkilot faoliyatini tavsiflashdagi tizimli yondashuvga asoslanadi. Modelarni qurish jarayonlarni tavsiflashdan boshlanadi, keyin ularidan boshlang'ich berilganlar (tashkilot o'z faoliyatida ishlataladigan, berilganlarning turg'un to'plami) ajratib olinadi. Birlamchi berilganlar mahsulot yoki xizmatlarni, bajariladigan amal (tranzaksiya)larni va iste'mol qilinuvchi zaxiralarni tavsiflashi mumkin. Birlamchi berilganlarga tashqi va ichki mohiyatni tavsiflovchilar, ya'ni xizmatchilar, mijozlar yoki agentliklar hamda qaror qilish natijasida olingan berilganlar, masalan ish grafiklari, mahsulotlar narxi kiradi.

DATARUN uslubiyatining asosiy tamoyili, «agarda birlamchi berilganlar kerakli ravishda berilganlar modelida tashkillashtirilgan bo'lsa, u holda ular axborot tizimining arxitekturasini yaratishda asosiy zamin bo'ladi», degan g'oyadan iborat. Axborot tizimining arxitekturasi turg'un bo'lishi uchun, u an'anaviy funksional modelga emas, balki asosiy ish faoliyati bilan chambarchas bog'langan va tijoratning tabiatini aniqlaydigan birlamchi berilganlarga asoslangan bo'lishi kerak.

Ixtiyoriy axborot tizimi, berilganlar bazalari bilan o'zaro harakatlanuvchi va protsessorlar orqali boshqariladigan modular majmuasidan iborat bo'ladi (19.20-rasm).



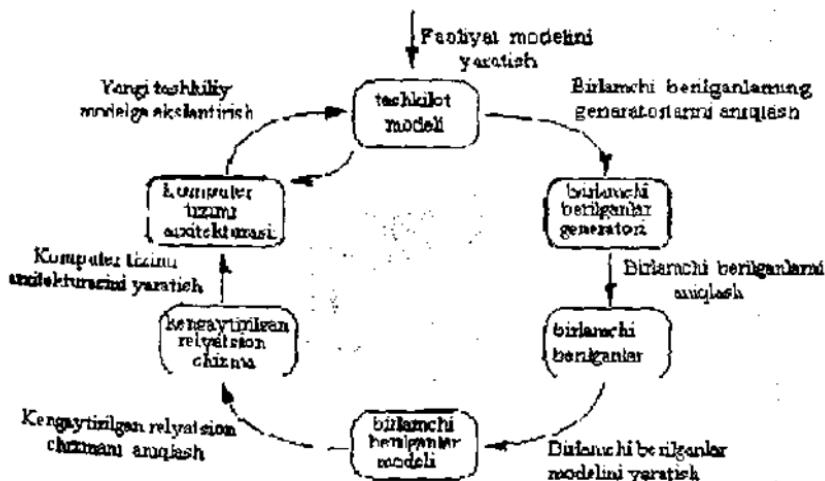
19.20-rasm. Axborot tizimining modeli.

Berilganlar bazalari va protsessorlar markazlashgan yoki taqsimlangan holatda joylashishi mumkin. Tizimdagи voqealar tashqi mohiyatlар (masalan bankomatlar yonidagi mijozlar kabi) yoki vaqtinchalik voqealar (oy yoki kvartal oxiri) orqali tashabbus ko'rsatishi mumkin. Hamma tranzaksiyalar bir yoki bir nechta berilganlar bazalari bilan o'zaro harakatlanuvchi obyektlar yoki aloqa modullari orqali amalga oshirilishi mumkin.

DATARUN uslubiyati ikkita maqsadni ko'zlaydi:

- axborot tizimini qurish mumkin bo'lgan, turg'un tuzilishni aniqlash. Bunday tuzilish berilganlar modeli bo'lib, ular tashkilotdagi fundamental jarayonlarni tavsiflovchi birlamchi berilganlardan olingan;
- berilganlar modeli asosida axborot tizimini yaratish.

Berilgarlar modellari asosida shakllantirilgan obyektlar, berilganlar bazalining obyektlaridan iborat bo'lib, odatda mijoz/server muhitidagi serverlarda joylashtiriladi. Kompyuter tizimi arxitekturasida aniqlangan aloqa obyektlari esa, odatda mijoz tomondagi kompyuterlarda joylashtiriladi. Berilganlar bazalining va turli aloqalar obyektlarining birgalikda ishlataligan tafsirlari asosidagi berilganlar modeli, axborot tizimining kuzatuvini ta'minlaydi. 19.21-rasmida axborot tizimini loyihalashdagi qadamlar ketma-ketligi keltirilgan.



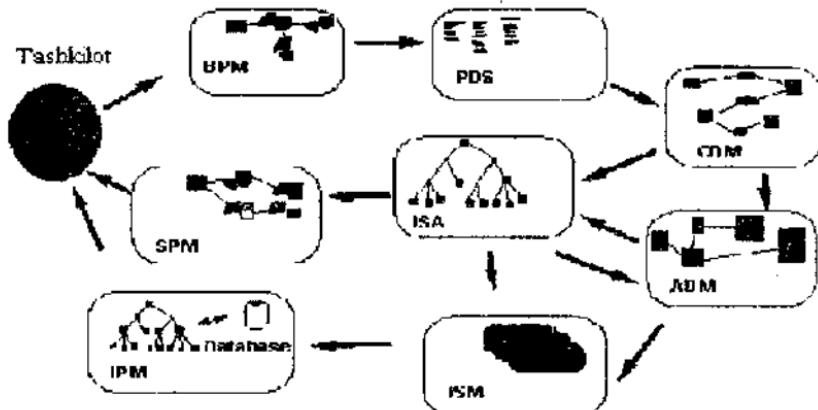
19.21-rasm. Axborot tizimini loyihalash qadamlari.

19.22-rasmida axborot tizimini ishlab chiqish jarayonida yaratiladigan modellar aniqlangan. Ularni yaratishda Silverrun CASE vositasi ishlatiladi.

Silverrun Amerikaning Somputer Systems Advisers, Inc. (CSA) firmasining mahsuloti bo'lib [91], tijorat sinfiga tegishli axborot tizimlarini tahlil qilish va loyihalash uchun ishlataladi. U alohida funksional va axborot modelari (berilganlar oqimlarining diagrammalari va «mohiyat-aloha» diagrammlari)ni qurishga asoslangan ixtiyoriy uslublar uchun qo'l keladi. Silverrun DATARUN uslubiyatiga mos ravishdagi loyihalash ishlarini amalga oshirishni avtomatizatsiyalashni ta'minlaydi. Axborot tizimi modellarni, tijorat jarayonlari modellaridan boshlab, programmaning modelini yaratishgacha bo'lgan jarayonlarni avtomatizatsiyalovchi, ketma-ket qurish orqali yaratiladi.

19.22 rasmdagi qisqartmalarning to'liq nomlari quyida keltirilgan:

- **BPM** (Business Process Model) – tijorat-jarayonlar modeli;
- **PDS** (Primary Data Structure) – birlamchi berilganlarning tuzilishi;
- **CDM** (Conceptual Data Model) – berilganlarning konseptual modeli;
- **SPM** (System Process Model) – tizimdagи jarayonlar modeli;
- **ISA** (Information System Architecture) – axborot tizimining arxitekturasи;
- **ADM** (Application Data Model) – ilovadagi berilganlar modeli;
- **IPM** (Interface Presentation Model) – aloqa taqdими modeli;
- **ISM** (Interface Specification Model) – aloqa tafsiri modeli.



19.22-rasm. DATARUN uslubiyati orqali yaratiladigan modellar.

Yaratiladigan axborot tizimi tashkilot bajaradigan funksiyalarga asoslanadi. Shuning uchun birinchi yaratiladigan model – bu *tijorat jarayonlar modeli* bo'lib, u BPM Silverrun da bajariladi. Bu model uchun BPM modulining maxsus *shartli yozma belgi* (*notatsiya*)lar ishlataladi. Tijorat

funksiyalarni tahlil va tavsirlash jarayonida asosiy axborot obyektlari aniqlanadi va berilganlarning modeldag'i oqimlar va omborlari tuzilishi sifatida hujjatlashtiriladi. Tuzilishlarni yaratish manbalari sifatida hujjatlar, mansab yo'riqnomalari, ishlab chiqarish amallarining tavsiflaridan foydalaniлади. Bu berilganlar tashkilot faoliyatida qanday mavjud bo'lsa, shunday tarzda modelda akslanadi. Ortiqchilikni normallashtirish va yo'qotish keyinchalik Silvrun ERX modulida berilganlarni konseptual modelini qurishda ishlataladi. Tijorat jarayonlar modeli yaratilgandan keyin axborot joyihaning repozitoriyisi (*meta berilgantari*)da saqlanadi.

Tashkilot faoliyatini ko'rikdan o'tkazishda birlamchi berilganlarning tuzilishlari aniqlanadi va hujjatlashtiriladi. Bu tuzilishlar, tashkilotdag'i hujjatlar, ma'lumotlar va berilganlarning tavsiflari sifatida, BPM modulining repozitoriyisiga kiritiladi. Tijorat jarayonlar modelida berilganlarning birlamchi tuzilishlari axborotlarning oqimlari va omborlari bilan bog'langan bo'ladi.

Silvertun ERX modulida birlamchi berilganlarning tuzilishi asosida berilganlarning konseptual modeli (ER-model) yaratiladi. U birlamchi berilganlardan ortiqchilik, nomlarning standartlashgani va normallashtashgani bilan farqlanadi. Bu amallar ERX modulida joylangan ekspert tizimi yordamida amalga oshiriladi. Berilganlarning konseptual modelining maqsadi, berilganlar bazasini amalga oshirish uchun foydalaniladigan ma'lumotlarni mayda tafsilotlarga e'tibor bermasdan, yaxshi normallashtashgan shaklda tavsiflashdan iborat.

Tijorat jarayonlar modeli va berilganlarning konseptual modeli asosida axborot tizimining arxitekturasi joyihalashtiriladi. Axborot tizimining arxitekturasi Silvertun BPM modulida maxsus ISA notatsiyalaridan foydalangan holda yaratiladi. Bu modelning asosiy mazmuni tizimning tarkibini va ular orasidagi tuzilmaviy harakatlarni shakllanirish. Berilganlarning konseptual modeli tizim tarkibiga kiruvchi ilovalarga mos ravishda qism-larga ajratiladi.

Ilovalarni ishlab chiqishdan oldin, berilganlar bazasining uyushinaviy berilganlar bazalarining tuzilishlari joyihalashtirilishi kerak. DATARUN uslubiyati berilganlarning relatsion modellaridan foydalanishni nazarda tutadi. Berilganlarning konseptual modeli normallashtirilgandan keyin, maxsus «ko'pri» ERX-RDM yordamida Silvrun RDM relatsion modellash moduliga o'tkaziladi. Modelni ERX formatdan RDM formatga o'zgartirish avtomatik tarzda bajariladi. Formatlar o'zgarishi natijasida relatsion model hosil bo'ladi. Bu model esa, Silvrun RDM modulida fizik jihatdan amalga oshirish uchun aniqlashtiriladi, ya'ni BBB tizimidagi berilganlarning turlari, kalitlar, indekslar, triggerlar, butunlikni chegaralash shartlari belgilanadi.

BPM modulida tizim jarayonlarining modeli yaratiladi va unda tijorat jarayonlarini qanday tarzda amalga oshirish aniqlangan bo'ladi. Bu model har bir ilova uchun alohida yaratiladi va ilovalardagi berilganlar modeli bilan uzviy bog'liq bo'ladi.

Ilovalar ekran shakllari, hisobotlar, berilganlarning ishlov protseduralari kabi aloqa obyektlaridan iborat bo'ladi. Tizimning har bir aloqasi (ekran shakllari, hisobotlar, berilganlarning ishlov protsедuralari) berilganlar bazasining qismlariga aloqador bo'ladi. Ilovalardagi berilganlarning modelida ilovalardagi aloqalarning har biri uchun berilganlar bazasining qism osti chizmalarini yaratiladi.

Aloqalarning tasavvur modeli, tizimdan oxirida foydalanuvchi tasavvuridagi aloqalar tashqi shaklini tafsifidan iborat. Bu, ekranning tashqi ko'rinishini yoki hisobot tuzilish shaklini, ilovalarni namoyishli yaratish vositalaridan biri bo'lib, u to'rtinchchi avlod tillari (4GL - Fourth Generation Languages) yordamida yaratilgan, hisobotning o'zi bo'lishi mumkin. To'rtinchchi avlod tillarining ko'pchiligi ilovalarning ishlovchi timsolini tez yaratish imkoniyatiga ega bo'lgani uchun foydalanuvchi loyihalashning boshlang'ich paytidayoq tizimning ishlovchi timsolini ko'rishi mumkin.

Ilovalar uchun relatsion modelning qism chizmalarini yaratilgandan keyin, har bir ilovaning batafsil tuzilishi ekranlar, hisobotlar, paketli ishlov protsедuralari tarzidagi chizmalar shaklidagi harakatlari loyihalanadi. Ushbu qadamda bu tuzilish, berilganlar bazalarining jadvallari va ustunlari aniqlangan, ularga ishlov berish, ekran shakllari va hisobotlari ko'rsatilgan holda, batafsil yoritiladi. Hosil bo'lgan model ilovaning batafsil hujjatlashitradi va bevosita maxsus aloqalarni programmalash uchun ishlataladi.

Keyin esa, ilovalarni yaratish vositalari yordamida tizimni fizik ishlab chiqish, ya'ni ilovalar programmalashtiriladi va ular axborot tizimiga kiritiladi.

### CASE vositalarning tavsisi va sinflari

Zamonaviy CASE vositalar axborot tizimlarini loyihalash texnologiyalarining anchasini qo'llab-quvvatlaydi, bunga berilganlarning oddiy tahlili va ularni hujjatlashtirishdan boshlab, butun programma ta'minotining yashash davrigacha bo'lgan jarayonni to'liq avtomatlashtirish bosqichlarini qamrab oladi.

Zamonaviy CASE vositalar bozorida hozirgi payda 300ga yaqin turli xillari mavjud bo'lib, ularning eng quvvatilari u yoki bu holatlarda jahondagi deyarli hamma yetakchi firmalar tomonidan ishlatalmoqda.

Odatda, CASE vositalarga programma ta'minotining yashash davridagi u yoki bu jarayonlar majmuasini automatizatsiyalovchi programma

vositalarini kiritishadi. Bu vositalar quyidagi asosiy alohida tavsiflarga ega bo'lishi kerak:

- axborot tizimini tavsiflash va hujjatlashtirish uchun juda kuchli grafik vositalarga, yaratuvchi bilan qulay aloqani va uning ijodkorlik imkoniyatlarini o'stirishni ta'minlash vositalariga;
- axborot tizimlarini yaratish jarayonini boshqarishni ta'mintovchi CASE vositalar alohida tarkiblarining integratsiya qilish imkoniyati mavjudligi;
- tashkillashtirilgan loyihamning repozitoriyalar omborlaridan foydalanishdagi maxsus yo'llari mavjudligi.

Integrallashgan CASE vositalar yoki programma ta'minotining to'liq siklini o'z ichiga olgan vositalar majmuasi, quyidagi tarkiblarni o'z ichiga oladi:

- CASE vositalarning asosi bo'lgan repozitoriyini. U loyiha naqlarini va uning alohidagi tarkiblarini saqlashni ta'minlashi, yaratuvchilar guruhga birlashib ishlaganda turli yaratuvchilardan tushgan ma'lumotlarni sinxronlash, meta berilganlarni to'liqlik va ziddiyatsizlik nuqtayi nazaridan nazorat qilishni;
- axborot tizimining modellarini tashkil etuvehi iyerarxik bog'langan diagrammalarni (DFD, ERD va h.k.) yaratish va tahrirlashni ta'minlovchi tahlil va loyihalashning grafik vositalarini;
- 4GL tillari va kodlar generatorlarini o'z ichiga olgan, ilovalarni ishlab chiqish vositalarini;
- shakliy boshqarish vositalarini;
- hujjatlashtirish vositalarini;
- testlash vositalarini;
- loyihami boshqarish vositalarini;
- reinjiniring (qayta loyihalash) vositalarini.

Hamma zamonaviy CASE vositalar asosiy turlar va toifalar bo'yicha sinflanishi mumkin. Turlar bo'yicha sinflarga ajratish yashash siklidagi u yoki bu jarayonlardagi CASE vositalarning funksional yo'nalishini aksantirishni nazarda tutadi.

CASE vositalar konseptual loyihalash bosqichini amalga oshiradi, mantiqiy va fizik loyihalashni esa, maqsadli BBB tizimi uchun avtomatik tarzda berilganlar bazasini generatsiyalash orqali amalga oshiradi. Lekin atamalar CASE vositalarda qat'iy ravishda qo'llanilmaydi. Juda ko'pgina CASE vositalarda ER modelni mantiqiy model, maqsadli berilganlar bazasining tuzilishidagi tasavvurni esa fizik model deb atashadi.

*Toifalar bo'yicha sinflanganda esa, bajariladigan funksiyalarining integrallashuv darajasi aniqlanadi. Kichik avtonom masalalarni yechuvchi*

lokal alohida vositalar, axborot tizimining ko'pgina bosqichlarini qamrab oluvechi qisman integrallashgan vositalar va axborot tizimining to'liq yashash siklini qo'llab-quvvatlovchi va umumiy repozitariy bilan bog'langan to'liq integrallashgan vositalar – *toifani tashkil etuvchilaridir*.

Bulardan tashqari CASE vositalarni yana quyidagi alomatlar bo'yicha sinflash ham mumkin:

- qo'llanilayotgan uslubiyat, tizimdagi modellar va berilganlar bazalari;
- BBB tizimlari bilan integrallashuv darajasi;
- turli yondashuvlardan foydalanish imkoniyatining mavjudligi.

Bugungi kunda CASE vositalar bozori quyidagi eng rivojlangan vositalar bilan ta'minlangan va ular doimiy ravishda o'zgarib turadi, yangilari paydo bo'ladi, eskilarining esa yangi naqllari paydo bo'ladi:

- Vantage Team Builder (Westmount I-CASE);
- Silverrun;
- ERwin+BPwin;
- S-Designor;
- CASE.Analitik;
- CASE /4/0;
- PRO-IV;
- System Architect;
- Visible Analyst Workbench;
- YeasyCASE;
- ProKit\*WORKBENCH;
- Design / IDEF;
- CASE Oracle (Designer / 2000);
- Power Designer (S-Designor);
- ERStudio.

CASE vositalarni taqqoslaganda, ishlataladigan ER modellash uslubiyatidan tashqari, yana avtomatizatsiyalashgan loyihalash funksiyalarini amalga oshirish bilan bog'liq bo'lган, quyidagi maxsus alomatlarni ham hisobga olish kerak:

- maqsadli BBB tizimini soni va ro'yxati;
- taqsimlangan berilganlar bazasini qo'llab-quvvatlashi;
- loyihalash jarayonida jamoaviy ishlash tamoyilini qo'llab-quvvatlashi (masalan, foydalanuvchi huquqlarini boshqarish, repozitariyni yuritish va h.k.);
- konseptual ER-modelni qurishni mavjud berilganlar bazasining tuzilishining tavsisi bo'yicha qayta tuzish;
- loyihalashning avtomatizatsiyalashtirilgan funksiyalari va ularni avtomatizatsiyalash darajalari;

- loyiha yechimlarining sifati va qat'iyligi (bir nechta muqobil yechimlardan tanlash, jarayonga fizik aralashuv);
- bajarilgan ishning ishonchliligi;
- loyihani hujjatlashtirish;
- tizimning ochiqligi, ya'ni boshqa vositalar bilan bog'lanish imkoniyati);
- grafik tahrirchining qulayligi;
- sonli cheklashlar (mohiyatlarning umumiy soni, ustki mohiyatlar uchun ichma-ich joylashish darajasi va h.k.);
- loyihalashtirilayotgan BBB tizimi uchun xotira hajmini avtomatik baholash imkoniyati;
- protsedurlarni avtomatik generatsiyalash imkoniyati;
- berilganlar omborlarini modellashtirish vositalarining mavjudligi;
- kompyuter zaxiralariiga talablarning mavjudligi;
- operatsion muhit;
- tizimning narxi.

CASE vositalar modelni turli darajada aniqlashtirishni ko'rsatadi:

- faqatgina mohiyatlarni va ular orasidagi aloqalarni belgilash;
- mohiyatlar + kalitlar;
- mohiyatlar + kalitlar + tashqi kalitlar;
- mohiyatlar + hamma atributlar.

Keltirilgan imkoniyatlarning mavjudligi, ayniqsa, katta va murakkab tizimlarni yaratishda juda muhim qulayliklar yaratadi.

CASE vositalarni taqqoslashdagi yana bir alomati shuki, quriladigan modelarning to'g'rilik slantjasini tekshirishdan iborat.

Modelni tekshirish vositalarining mavjudligi, e'tiborsizlik natijasida paydo bo'lgan – mohiyatning identifikatorining bo'lmasligi, obyektlarning o'zaro aloqasining mavjud emasligi, nomlarning to'g'ri qo'yilmagani, loyihalashning keyingi bosqichlari uchun ma'lumotlarning yetarli emasligi, modeldag'i ziddiyatlarning mavjudligi va h.k. xatolami tuzatishga yordam beradi. Ko'pgina CASE vositalarda, modeldag'i butunlik chegarasini berish va buning uchun programmalarni generatsiya (triggerlar, saqlanuvchi protsedralar) qilish, berilganlar bazalarini ishlatalish davrida esa bu chegaralarni tekshirish imkoniyati mavjud. Bundan tashqari CASE vositalar BBni yurituvchi programmalarni generatsiya qilish imkoniyatiga ega. CASE vositalarda modellarni boshqa tizimlarga eksport qilish va aksincha, boshqa tizimlardan import qilish imkoniyatlari mavjud.

## Xulosa

CASE vositalar modelni turli darajada aniqlashtirishni ko'rsatadi:

- saqatgina mohiyatlarni va ular orasidagi aloqalarni belgilash;
- mohiyatlар + kalitlar;
- mohiyatlар + kalitlar + tashqi kalitlar;
- mohiyatlар + hamma atributlar.

Keltirilgan imkoniyatlarning mayjudligi, ayniqsa, katta va murakkab tizimlarni yaratishda juda muhim qulayliklar yaratadi.

CASE vositalarni taqqoslashdagi yana bir alomat quriladigan modellarning to'g'rilik darajasini tekshirishdan iborat.

Hech bir lovihalashni avtomatizatsiyalash tizimi, qurilgan konseptual modelni predmet soha vogeligiga mosligining kafolatini bermaydi. Bu saqatgina tizimni yaratuvchilarning malakasiga, predmet sohani tushunishiga va uni modelda qanchalik ustalik bilan akslanirishiga bog'liq.

## Nazorat uchun savollar

1. Barker notatsiyasi nimaga asoslangan?
2. Berilganlarni modellashtirishdagi birinchi qadam nimadan iborat?
3. Berilganlarni modellashtirishdagi ikkinchi qadam nimadan iborat?
4. Barker notatsiyasida birlamchi kalit, majburiy va majburiy bo'lmagan atributlar qaysi belgilarni ifodalanganadi?
5. Barker notatsiyasida rekursiv aloqa va siljimas aloqalarni belgilab ko'rsating.
6. IDEFIX uslubiyatida berilganlar qaysi modelga tayanadi?
7. Avlod va ajod mohiyatlarga misol keltiring.
8. IDEFIX uslubiyatida aloqalar quvvati necha turda bo'lishi mumkin?
9. Identifikatsiyalangan va identifikatsiya qilinmagan aloqalarga misol keltiring.
10. Birlamchi, tashqi va birlamchi tashqi kalitlarni IDEFIX uslubiyatida asosida belgilang.
11. DATARUN uslubiyatidagi axborot tizimi loyihasi miqyosida qanday masalalar ko'rildi?
12. DATARUN uslubiyati qaysi modellarga asoslanganadi?
13. DATARUN uslubiyatining asosiy tamoyilini tushuntiring.
14. DATARUN uslubiyatida axborot arxitekturasi turg'unligini nima ta'minlaydi?
15. DATARUN uslubiyatining maqsadi nima?
16. DATARUN uslubiyatidagi axborot tizimini loyihalashdagi asosiy qadamlarni aytib bering?

17. DATARUN uslubiyati orqali nechta turdag'i modellar yaratiladi?
18. DATARUN uslubiyatidagi berilganlar qaysi modelga tayanadi?
19. CASE vositalar yordamida qanday ishlarni amalga oshirish mumkin?
20. CASE vositalarni sinflarga ajratishdagi asosiy ko'rsatkichlar nima?
21. Zamonaviy CASE vositalar bozorida ularning nechta turi mavjud?
22. CASE vositalarni taqqoslashda nimalarga e'tibor berish kerak?

---

## **20-bob. Toad Data Modeler muhitida berilganlarni modellashtirish**

**Tayanch iboralar:** *Toad Data Modeler, Oracle, Microsoft SQL Server, MySQL, PostgreSQL, DB2, Sybase ASE, Sybase SQL Anywhere, ERD, SQL, HTML, RTF yoki PDF, «to-do-ro'yhat», ProjecS, Models, Universal Data Model, Yentities, General, Yentity Properties, Attributes, Add, Integer, Yedit, Varchar, Kays, Relationship, ObjecS, Simple Model Conversion, Mandatory, Logical Model Yexplorer, Fizik Model Access, DDL Script, Generate DDL Script, Macros, MS SQL Server, Fizik Model SQL Server.*

### **Toad Data Modeler muhitining imkoniyatlari**

BBBBT juda yuqori sur'atda rivojlanmoqda va turli axborot tizimlarida keng qo'llanmoqda. Berilganlar bazalarini loyihalash va modellashtirish va yaratilgan modellarni BBB tiziiniga uzatish hamda BB ishlash uchun turli xildagi instrumental muhitlar yaratilmoqda. Biz ana shunday instrumental muhitlardan biri, ya'ni zamonaviy **Toad Data Modeler** muhiti bilan tanishamiz hamda berilganlar bazalarini loyihalash bo'yicha nazariy jihatdan olgan bilimlarimizni ushbu muhit doirasida amalga oshirish imkoniyatlaridan foydalananamiz.

*Quest Software* kompaniyasining **Toad Data Modeler** muhiti berilganlar bazasining mavjud bo'lgan tiziimlarini tahlil qilish yoki yangilarini yaratish, qo'llab-quvvatlash va hujjatlashtirishda grafik aloqalardan foydalanigan holda mutaxassislarga yordam beradi. **Toad Data Modeler** muhiti berilganlar bazasining tuzitishini mavjud bo'lgan baza asosida loyihalash, modellarni yangilash, SQL tilini yaratish va generatsiyalash, modellarni taqqoslash va o'zgartirilgan skriptlarni (programma kodlarini) generatsiya qilish hamda HTML va RTF formatlarida hisobotlarni yaratish imkoniyatlarini beradi.

### **Toad Data Modeler muhitida quyidagilarni amalga oshirish mumkin:**

- berilganlar bazasining tuzitishini namoyishli tarzda yaratish, ya'ni mantiqiy va fizik loyihalashda «mohiyat-aloqa» diagrammlarini (ERD) yaratish;
- Oracle, Microsoft SQL Server, MySQL, PostgreSQL, DB2, Sybase ASE, Sybase SQL Anywhere va boshqa BBBT uchun ERD yaratish;

- berilganlar bazasining mavjud tuzilishlarini aks loyihalash va ularni diagramma shaklida ko'rish;
- diagrammalarga mantiqiy berilganlarni qo'shish va mavjud berilganlar bazalarining tuzilishini yaxshi tavsiflash;
- ERD modelni tekshirish va xatolar, ogohlantirishlar va yordam olish;
- maqsadli berilganlar bazasining SQL tilidagi kodini avtomatik generatsiya qilish;
- HTML, RTF yoki PDF formatlarda hujjatlarni bat afsil generatsiyalash;
- o'zgartirilgan skriptlarni generatsiyalash va modellarni birlashtirish xossalardan foydalangan holda, modelni fizik mavjud bo'lgan BB bilan sinxronlashtirish (moslashtirish);
- «Naqlar menejeri» dan foydalangan holda, o'zgartirishlarni kuzatish;
- «to-do ro'yxat» dan foydalangan tarzda, modelga yoki uning qismiga tegishli bo'lgan masalalar ro'yxatini yaratish va qo'llab-quvvatlash;
- muhitni talabga binoan sozlash va h.k.

#### *Muhitning afzalliklari:*

- tizimni yaratishdaga xotalarni kamaytirish;
- tizimni yaratishdagi yuqori samaradorlikka erishish;
- berilganlar bazasining tuzilishini vizuallashtirish va **mavjud bazalarni** tuzilishini chuqur tushunish;
- avtomatik tarzda bajarilgan ishlarni tekshirish;
- o'ta bat afsil hisobotlarni HTML yoki RTF da generatsiyalash;
- mavjud berilganlar bazalari bilan ishflash;
- mavjud berilganlar bazalarining tuzilishini **zamonaviy darajada** ushlab turish.

#### *Muhitning eng asosiy tavsifnomalari:*

- berilganlarning universal (utmumiy) relatsion, mantiqiy va **fizik** modellarini yaratish;
- turli BBB tizimlarini qo'llash;
- aks loyihalash;
- SQL/DL skriptlarni generatsiyalash;
- o'zgartirilgan skriptlarni generatsiyalash;
- HTML/RTF hisobotlarni yaratish;
- **modellarni tavsiflash;**
- modellarni birlashtirish;
- modellarni taqqoslash;
- modellarni tekshirish;

- avtomatik tarzda maketlarni yaratish;
- tahrirlanuvchi shakllarni yaratish;
- nusxalardan foydalanish;
- oshkormas qiymatlardan foydalanish;
- ilovalardagi o'zgaruvchilardan foydalanish;
- makroslar va sozlanuvchi shakllarni yaratish;
- o'rnatiluvchi chizimlardan foydalanish;
- loyihalarni yaratish;
- modellarni yangilash;
- obyektlar yoki ularning guruhlarini miqyoslash;
- «to-do-ro'yxat» lardan foydalanish va h.k.

**Toad Data Modeler** muhitida berilganlar bazasini model tashtirish jarayoni juda qulay. Bu muhit yordamida ixtiyoriy BBning konseptual, mantiqiy va fizik modelini yaratish mumkin.

**Toad Data Modeler** muhitida yangi BBni yaratish, mavjud berilganlar bazasini ochish va o'zgartirish mumkin.

**Toad Data Modeler** muhitida jadvallar, so'rovlar yaratish mumkin. Yaratilgan jadvalga birlamchi va tashqi kalitlarni qo'shish mumkin.

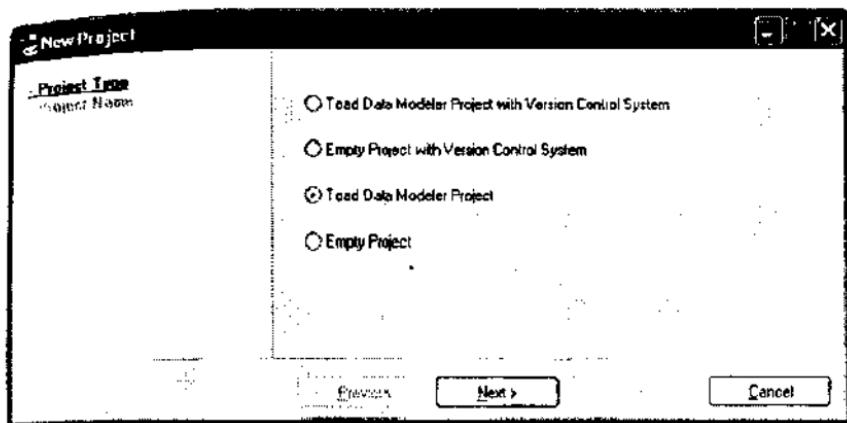
**Toad Data Modeler** muhitida keltirilgan imkoniyatlarni chuquiroq o'rganish uchun *Talabalar o'quv loyihamizni* ikki usulda modellaشتirishni amalga oshiramiz. *Birinchi usulda* berilgan bazasining konseptual (universal) relatson modelini yaratamiz va uni yuqorida keltirilgan BBB tizimlaridan biriga konvertatsiya qilishgacha bo'lgan hamma bosqichlarni ko'ramiz. *Ikkinci usulda* esa, avval mantiqiy modelni yaratamiz va so'ngra mantiqiy modelni fizik modelga o'tkazib, uni tekshirib, xatolari bo'lsa tuzatib, MS Access va MS SQL Server 2005 relatson BBB tizimlari uchun berilganlar bazasi va SQL so'rovlarini yaratib, ularning skriptlarini hosil qilamiz. So'ngra skriptlarni boshqarish tizimlariga uzatish variantlari hamida MS Access va MS SQL Server 2005 hosil bo'lgan *Talabalar o'quv loyihamizning* boshqarish tizimlariga uzatilgan berilgan bazalarining tayyor holatlarini ko'ramiz.

Yuqorida keltirilgan ikkala usulni, *Talabalar o'quv loyihasidagi* munosabatlari bo'yicha 5 ta jadvaldan iborat berilganlar bazasini **Toad Data Modeler** muhitining 4.0.6.15 naqlidan foydalangan holda amalga oshiramiz.

### Berilganlarning konseptual modelini yaratish

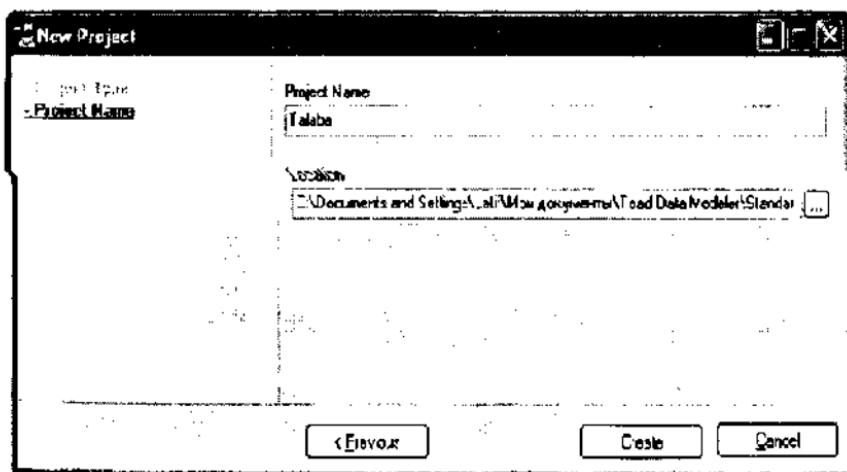
Maqsadni amalga oshirishni, berilganlarning konseptual modelini yaratishdan boshlaymiz. Buning uchun File -> New -> Projects...

buyruqlar ketma-ketligini bajaramiz. Loyihamizning turini tanlash uchun 20.1-rasmidagi holatni amalga oshiramiz:



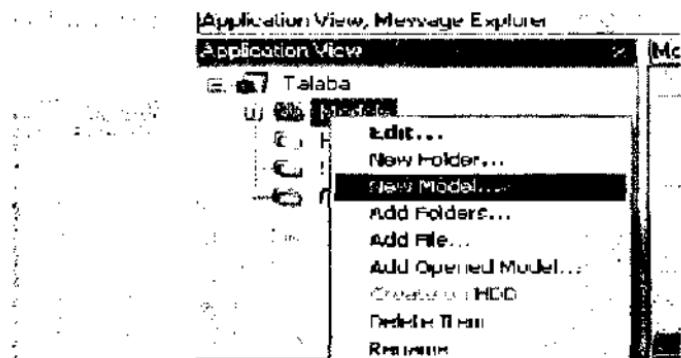
20.1-rasm.

Loyihamizga nom beramiz:



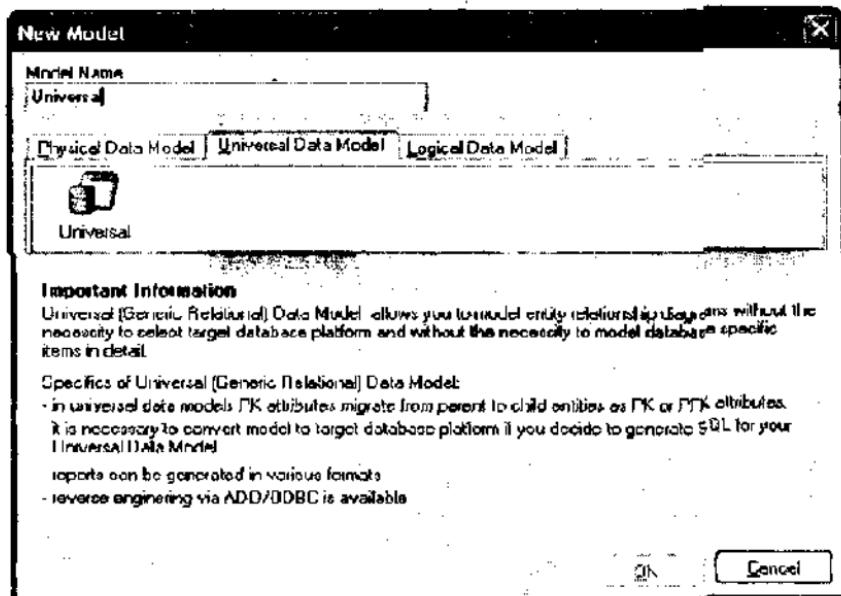
20.2-rasm.

Endi loyihamizga universal model qo'shamiz. Buning uchun yaratgan loyihamizdagi **Models** ni tanlab, o'ng tugmani bosamiz va yangi oynadan **New model...** ni tanlaymiz:



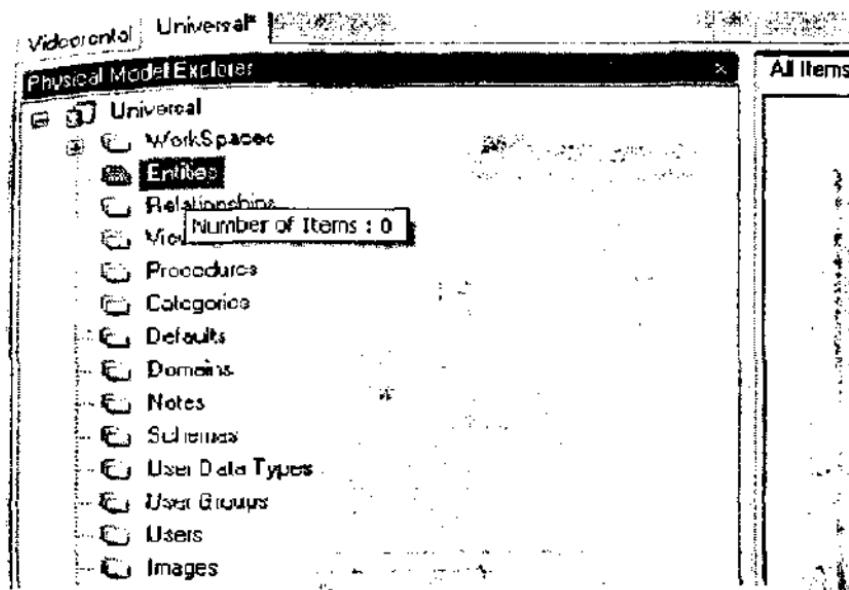
20.3-rasm.

Quyidagi oyna ochiladi. Bu oynadan **Universal Data Model** bo'limiga o'tamiz. Modelga Universal deb nom beramiz:



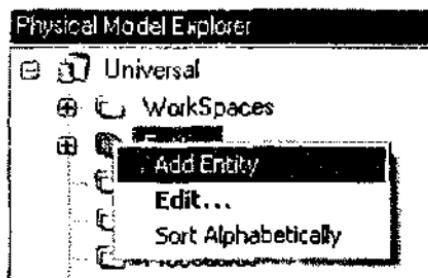
20.4-rasm.

OK tugmasini bosganimizdan so'ng, **Universal Model** Yexplorer oynasi hosil bo'ladi (20.5-rasm).



20.5-rasm.

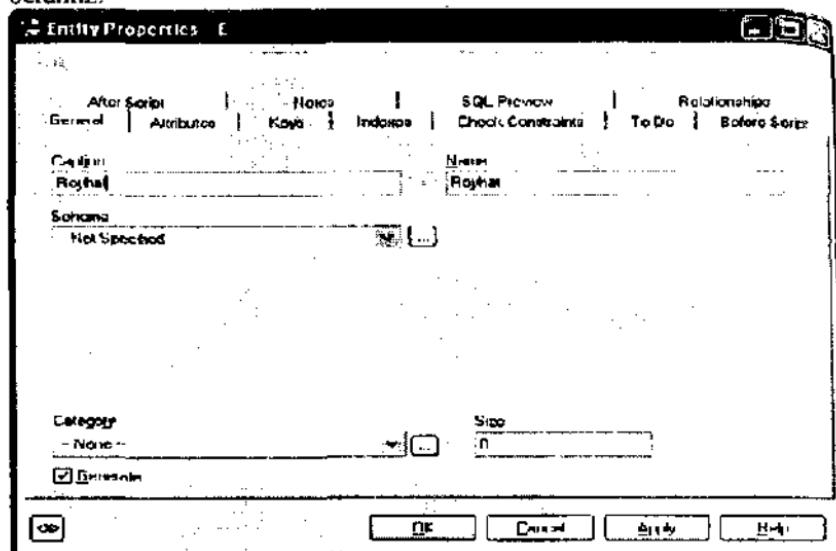
Universal modelga Universal nomini berdik, endi bu modelga *Talabalar o'quv toyihasidagi* munosabatlarni birin-ketin hosil qilamiz. Buning uchun Yentities papkasining ustiga o'ng tugmani bosib, Add Yentity buyrug'iini tanlaymiz:



20.6-rasm.

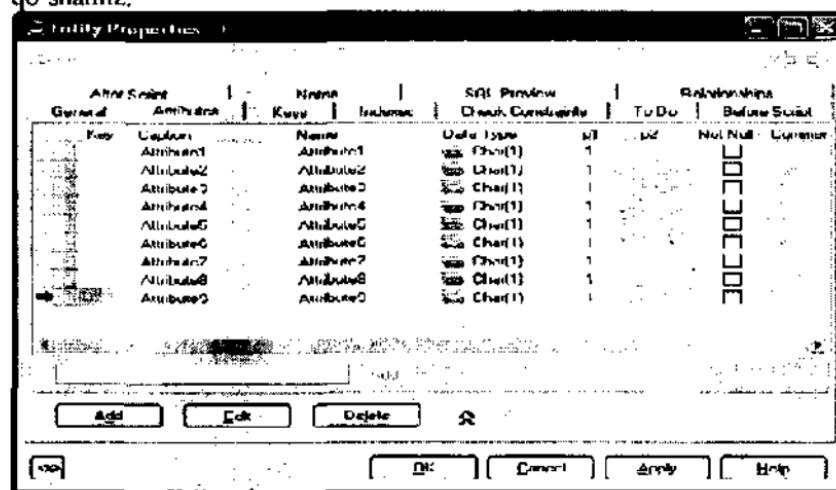
All Items oynasida avtomatik tarzda Yentity1 munosabati hosil bo'ladi. Kursorni ustiga olib kelib, 2 marotaba bosamiz. Yentity Properties

oynasi ochiladi. General bo'limida munosabatga Royhat deb noma beramiz:



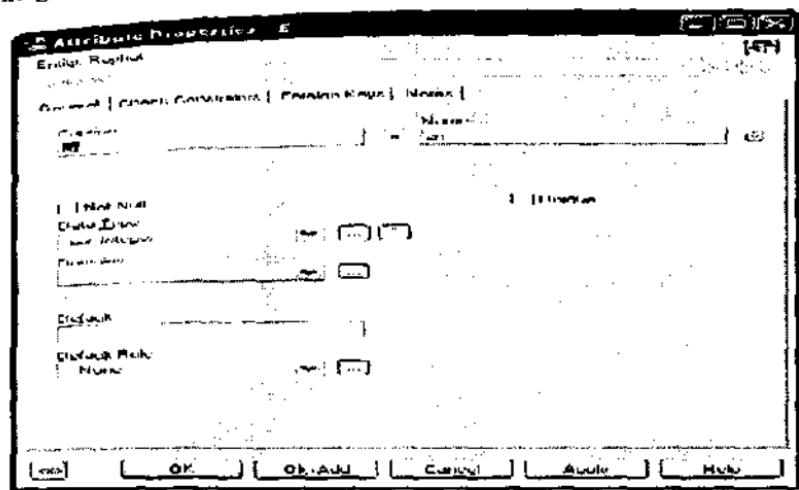
20.7-rasm.

Attributes bo'limiga o'tamiz va Add tugmasi yordamida 9 ta atribut qo'shamiz.



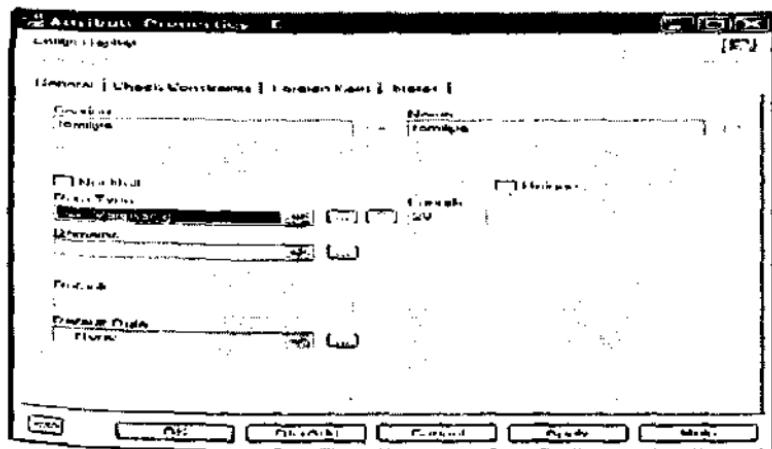
20.8-rasm.

Edit tugmasi yordamida Attribute Properties oynasini ochamiz va atributning xususiyatlarini beramiz. Atributga **Zn** deb nom beramiz, turini **Integer** deb tanlaymiz va **OK** tugmasini bosamiz.



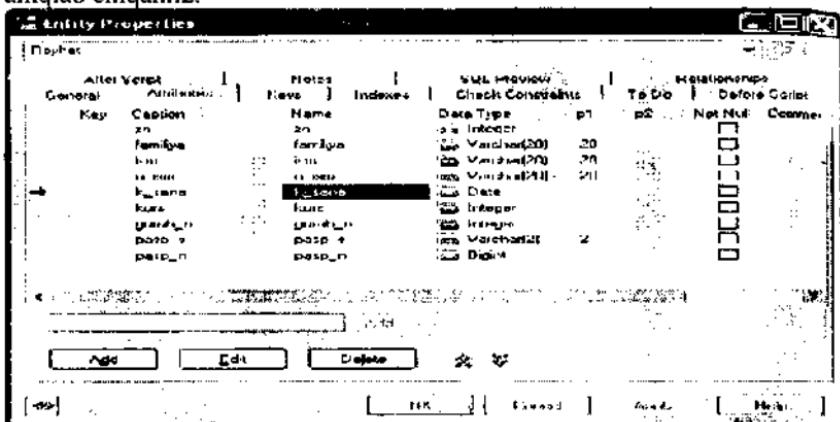
20.9-rasm.

Keyingi aribut familiya, turi **Varchar**, uzunligi **length 20** ta belgi va **OK** tugmasini bosamiz.



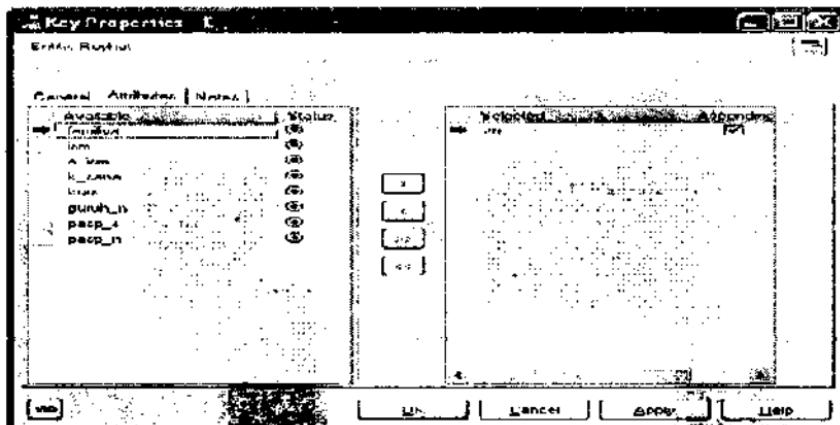
20.10-rasm.

Qolgan atributlarning nomlari, turlari va uzunliklarini ham shu taraqa aniqlab chiqamiz:



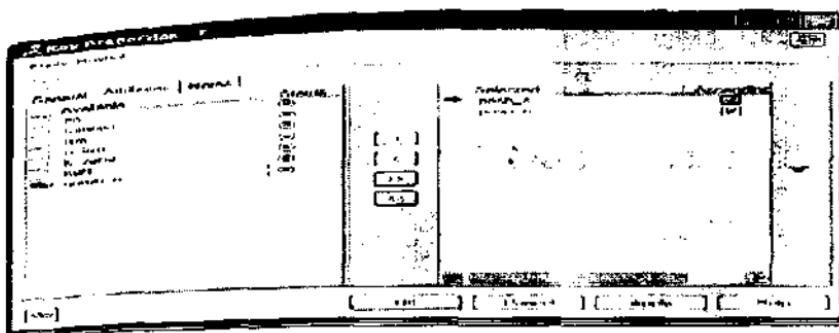
20.11-rasm.

Atributlarni aniqlab bo'lgach, kalitlarni kiritamiz. Buning uchun Kays bo'limiga o'tib, Add tugmasi yordamida 2 ta kalit qo'shamiz:  
**zn** atributi uchun **pk\_zn** birlamchi noyob kalit

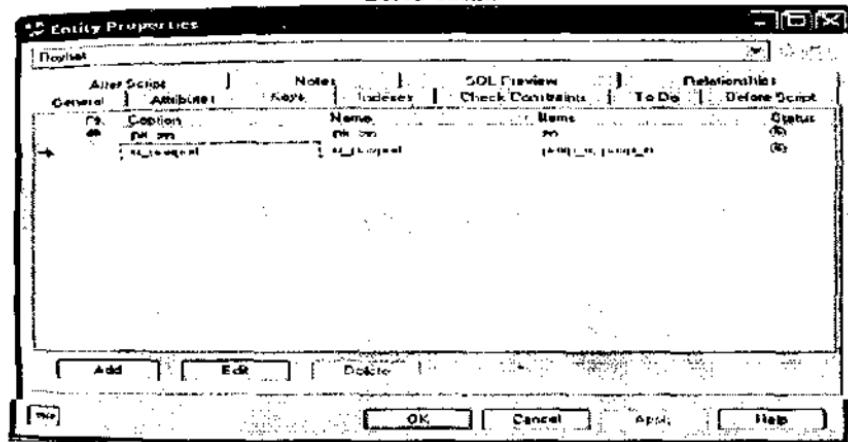


20.12-rasm.

**pasp\_s**, **pasp\_n** atributlariga tarkibli **u\_pasport** noyob kalit qo'yamiz. Buning uchun Edit tugmasini bosib **pasp\_s**, **pasp\_n** atributlarini ro'yxatning o'ng qismiga o'tkazamiz va **OK** tugmasini bosamiz.



20.13-rasm.



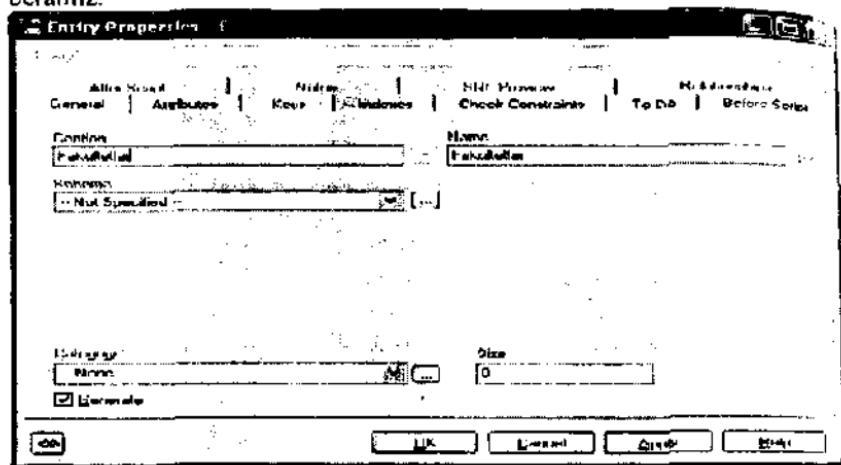
20.14-rasm.

Royhat munosabatimiz tayyor holatga keldi:

Royhat			
q_zn	Integer	NN	(PK)
familiya	Varchar(20)		
ism	Varchar(20)		
o_ism	Varchar(20)		
k_sana	Date		
kurs	Integer		
guruh_n	Integer		
pasp_s	Varchar(2)	NN	(AK1)
pasp_n	Bigint	NN	(AK1)

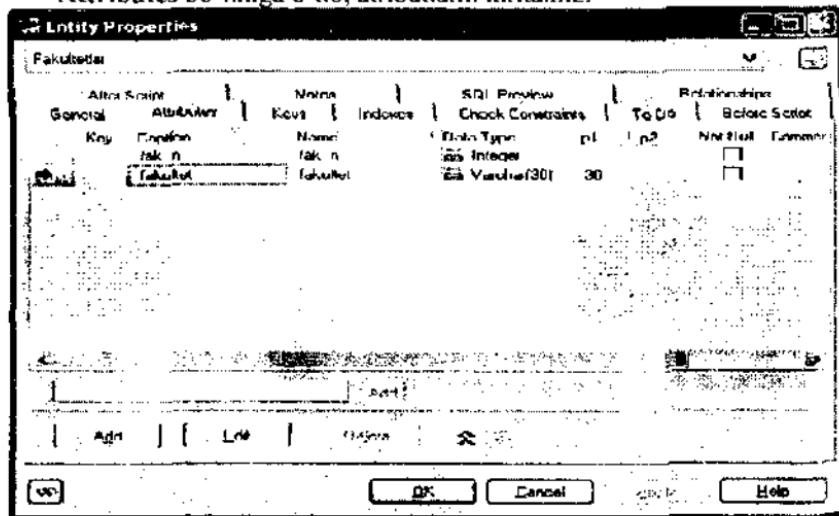
20.15-rasm.

Endi **Fakultetlar** munosabatini yaratamiz. Buning uchun Entity-papkasining ustiga o'ng tugmani bosib, **Add Entity** buyrug'ini tanfaymiz. All Items oynasida hosil bo'lgan yangi munosabatga **Fakultetlar** deb nomberamiz.



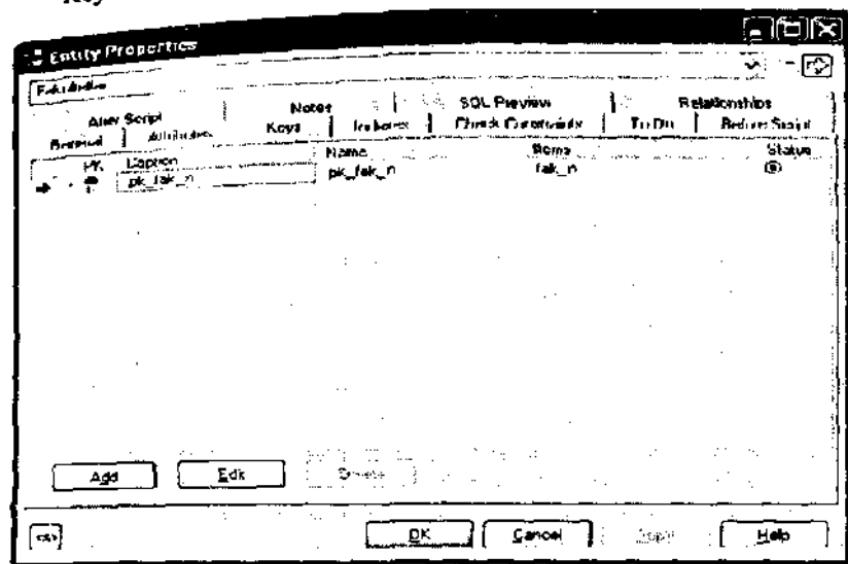
20.16-rasm.

Attributes bo'limga o'tib, atributlarni kiritamiz:



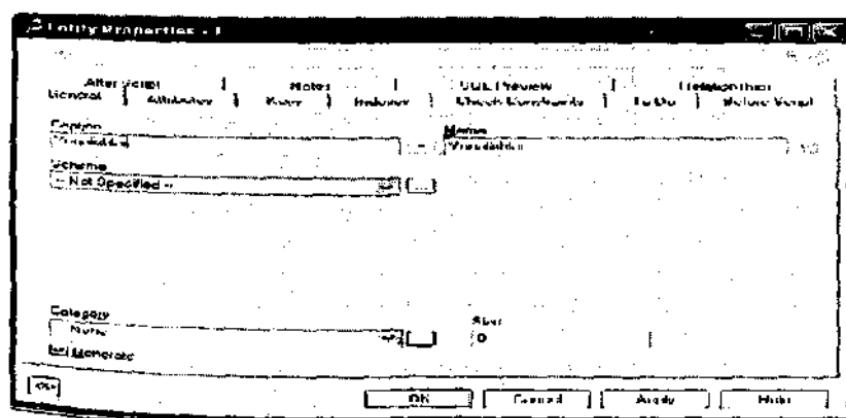
20.17-rasm.

Keys bo'limiga o'tib, pk\_fak\_n kalit qo'shamiz:



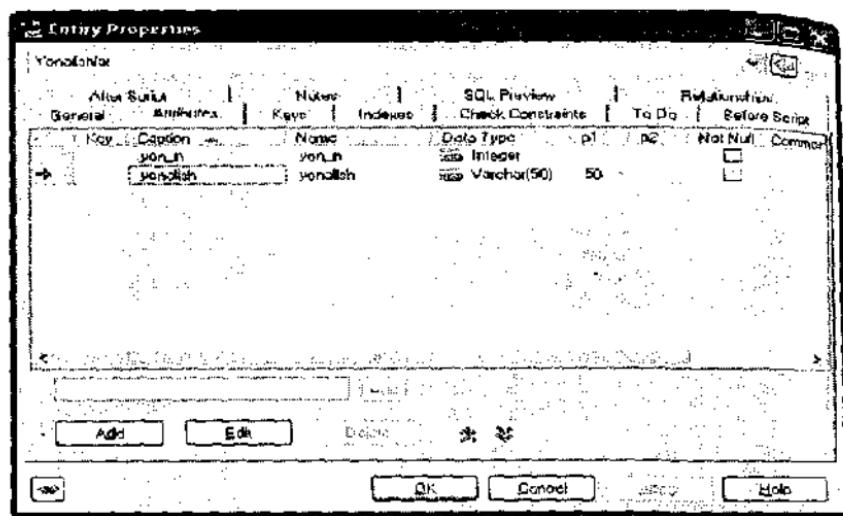
20.18-rasm.

OK tugmasini bosamiz. Fakultetlar munosabati ham tayyor.  
Endi Yo'nalishlar munosabatini yaratamiz.



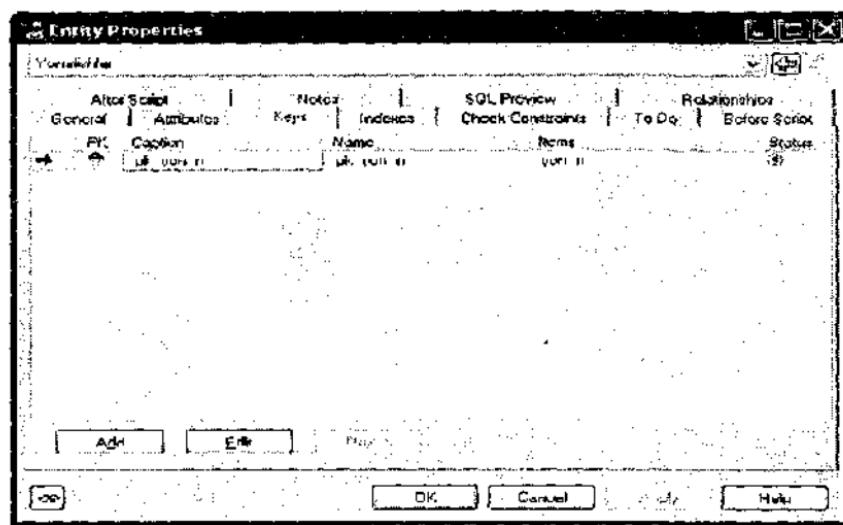
20.19-rasm.

Attributes bo'limiga o'tib, atributlarni kiritamiz:



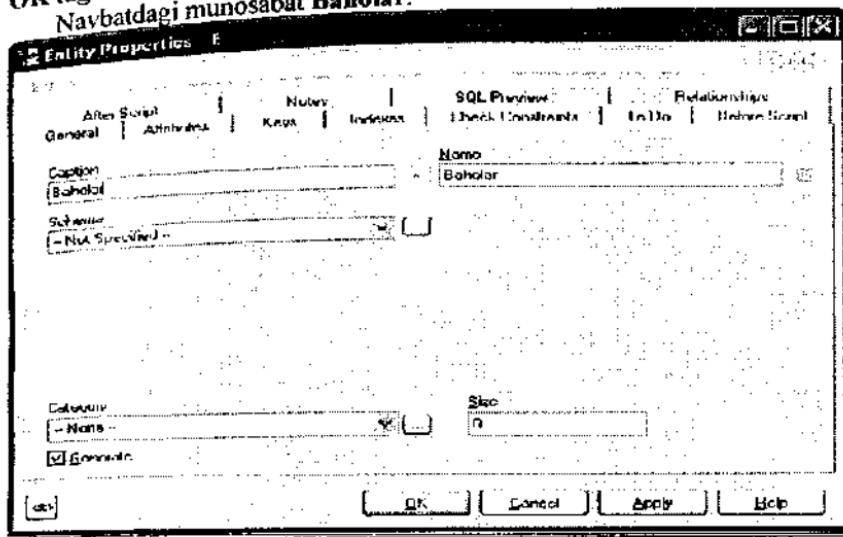
20.20-rasm.

Keys bo'limiga o'tib, pk\_yon\_n kalit qo'shamiz:



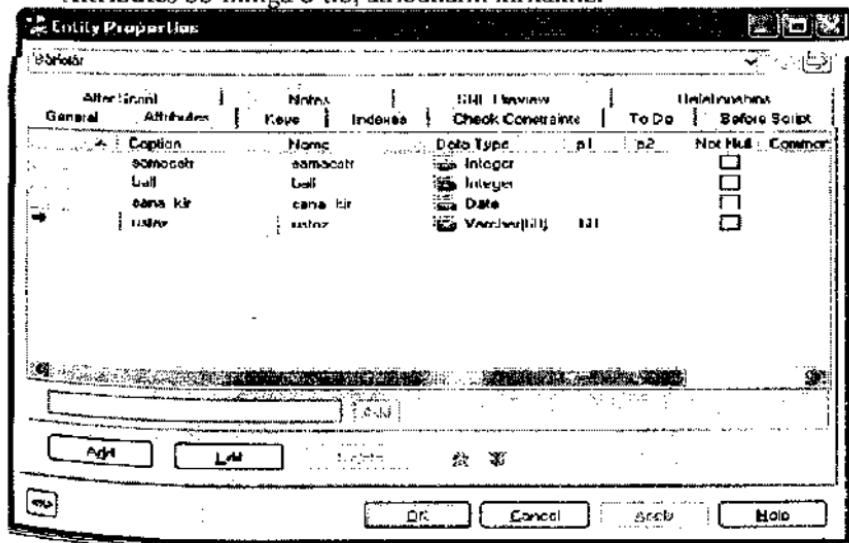
20.21-rasm.

OK tugmasini bosamiz.  
Navbatdag'i munosabat Baholar.



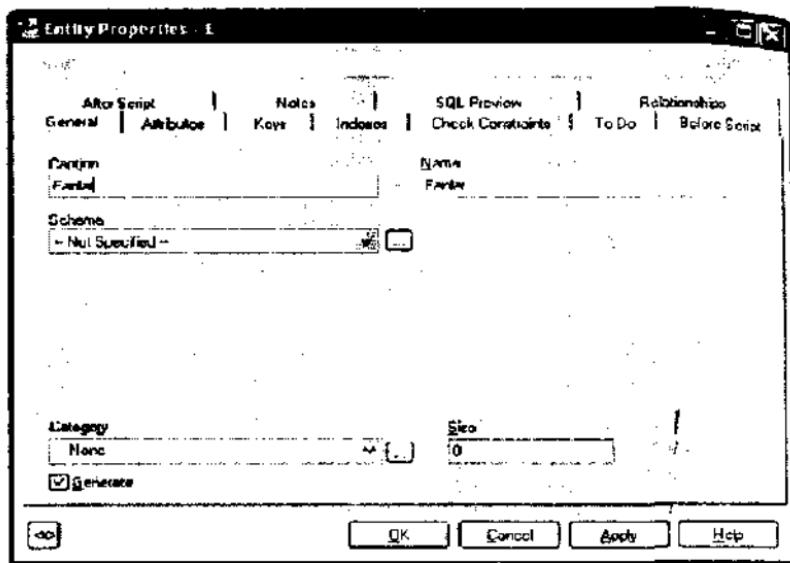
20.22-rasm.

Attributes bo'limiga o'tib, atributlarni kiritamiz:



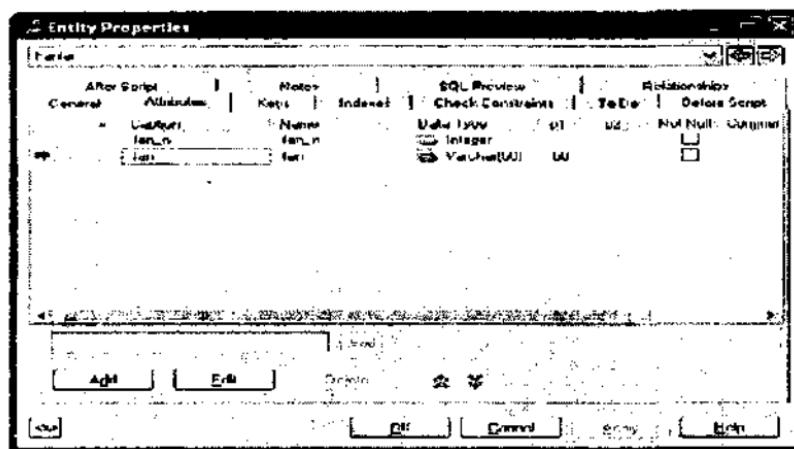
20.23-rasm.

**OK tugmasini bosamiz.**  
**So'nggi munosabat Fanlar:**

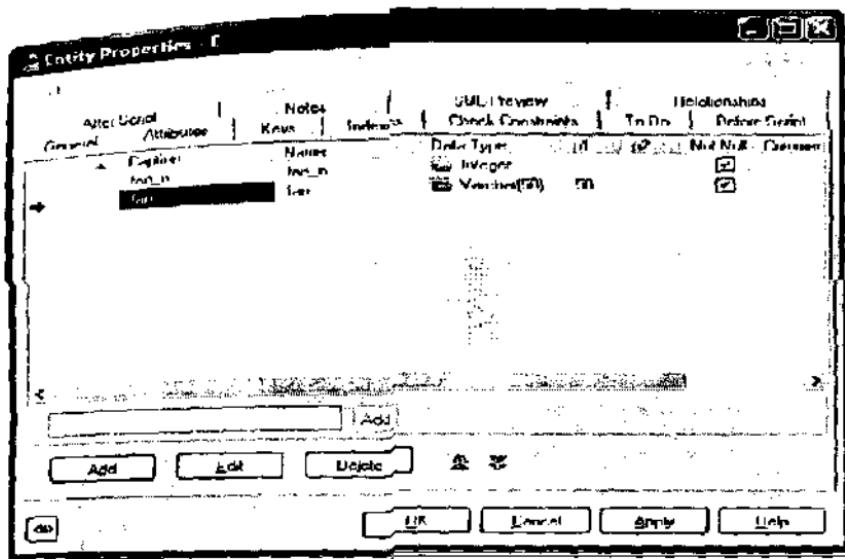


20.24-rasm.

**Attributes bo'limiga o'tib, atributlarni kiritamiz:**

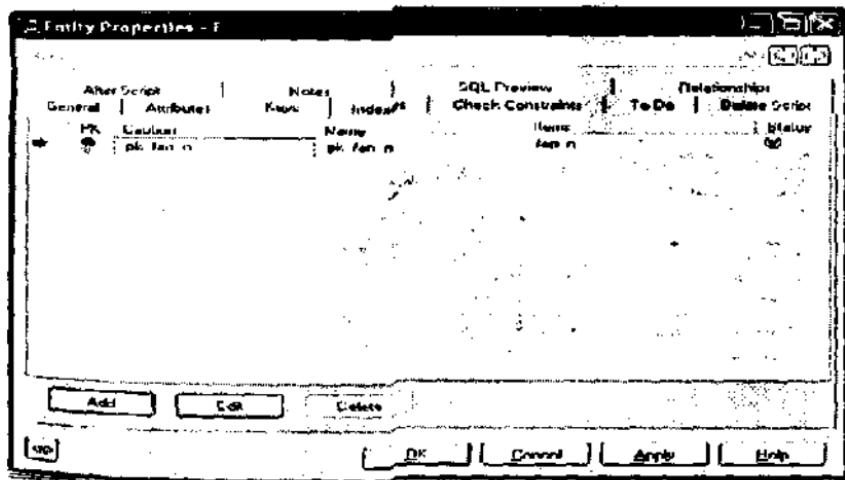


20.25-rasm.



20.26-rasm.

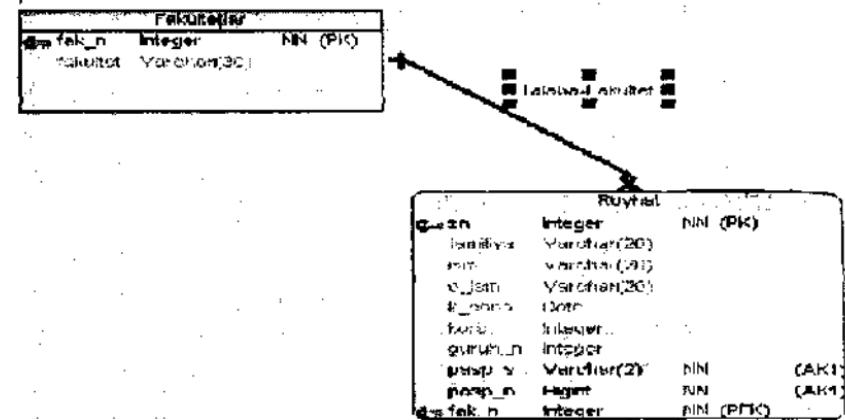
Keys bo'limiga o'tib, pk\_fan\_n keltit qo'shamiz:



20.27-rasm.

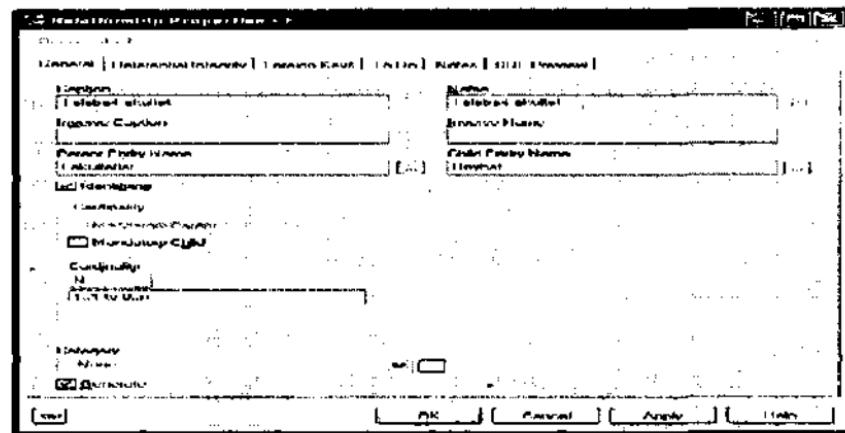
OK tugmasini bosamiz.

Barcha munosabatlarni hosil qilib oldik. Endi bu munosabatlarni tashqi kalitlar yordamida bog'lab chiqamiz. Buning uchun ObjecS menyusidan Relationship buyrug'ini tanlaymiz. Avval Fakultetlar munosabatining ustida sichqonchani bosib, keyin Royhat munosabatining ustida sichqonchani bosamiz.



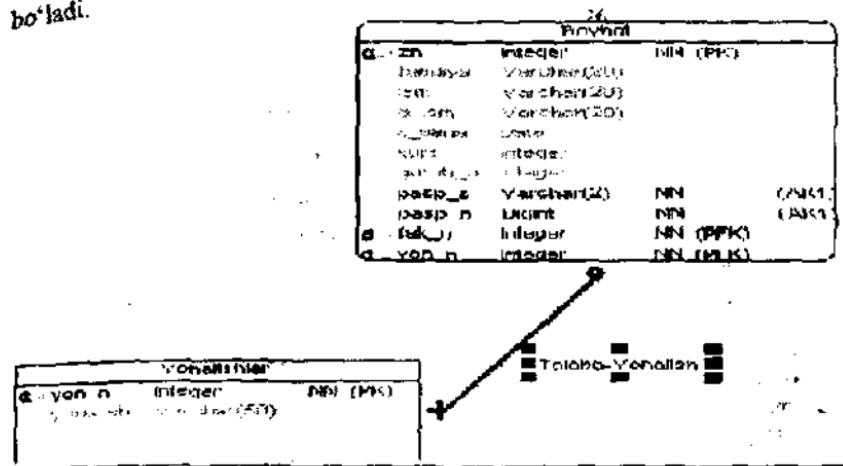
20.28-rasm.

Hosil bo'lgan chiziqning ustida sichqonchani 2 marta bosib, tashqi kalitni xususiyatlarini o'zgartirish mumkin. Bu aloqani Talaba-Fakultet deb nomlaysiz:



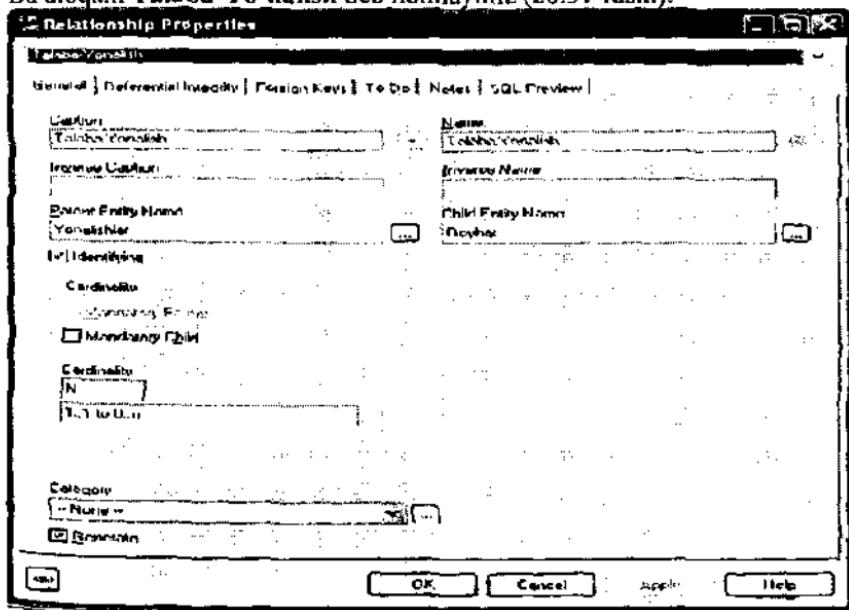
20.29-rasm.

Keyingi tashqi kalit Yo'nalishlar va Royhat munosabatlari orasida bo'ladi.



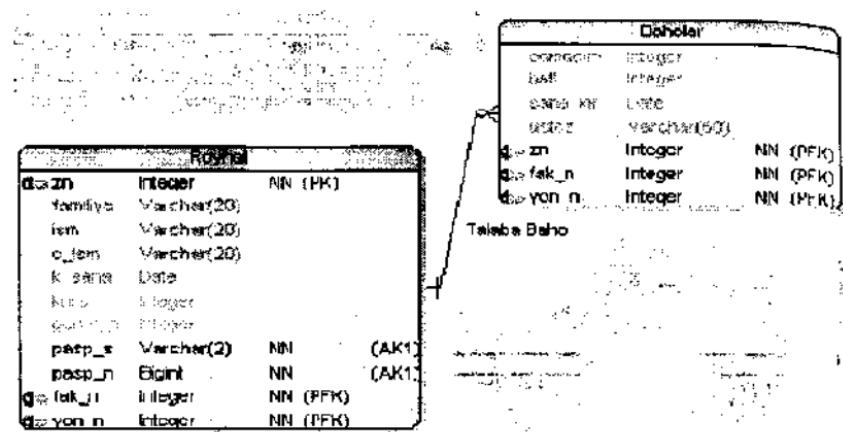
20.30-rasm.

Bu aloqani Talaba-Yo'naliish deb nomlaymiz (20.31-rasm).



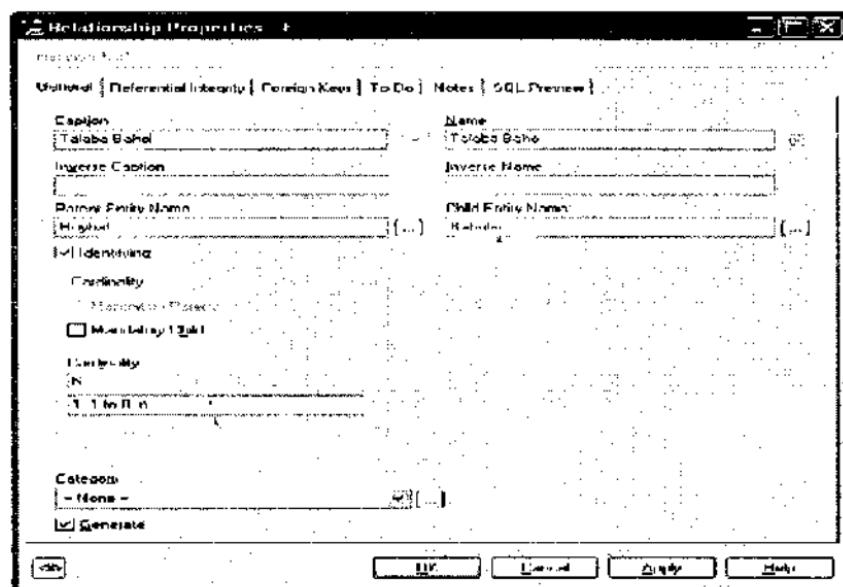
20.31-rasm.

Keyingi tashqi kalit Royhat va Baholar munosabatlari orasida bo'ldi.



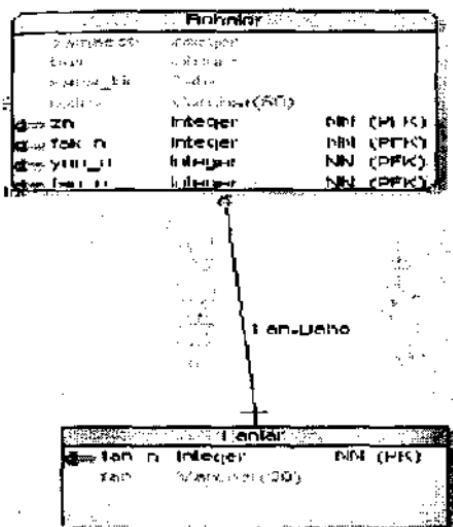
20.32-rasm.

Bu aloqani Talaba-Baho deb nomlaymiz:



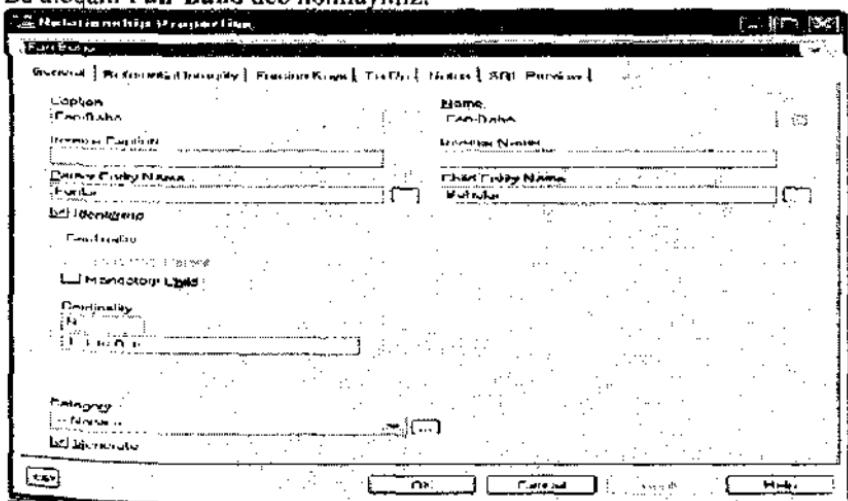
20.33-rasm.

Keyingi tashqi kalit Fanlar va Baholar munosabatlari orasida bo'ldi.



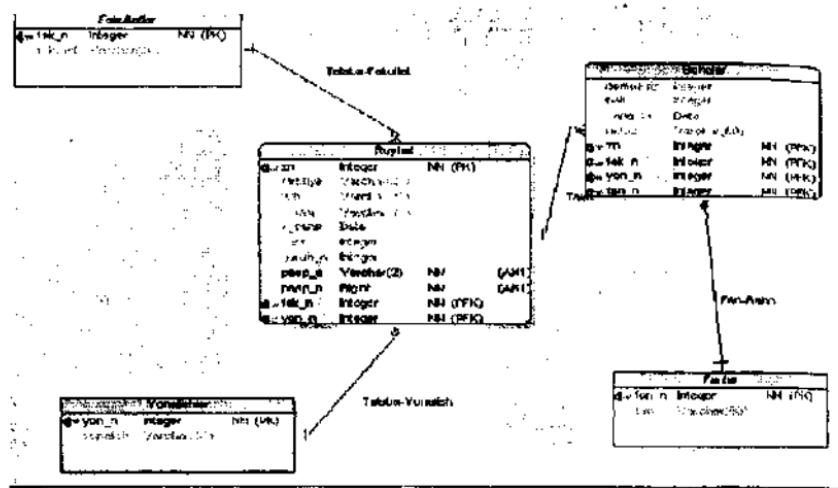
20.34-rasm.

Bu aloqani Fan-Baho deb nomlaymiz:



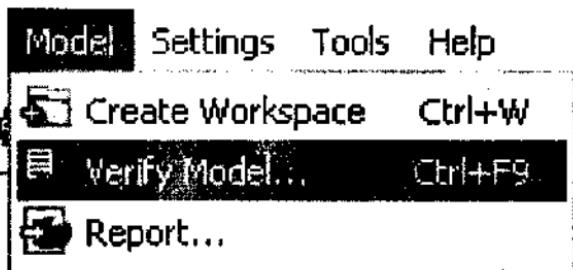
20.31-rasm.

Natijada ekranda quyidagicha chizma hosil bo'ldi:



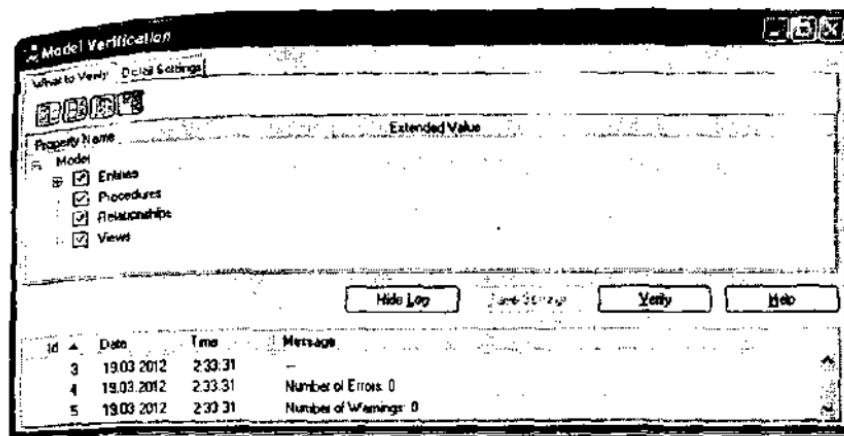
20.36-rasm.

Endi yaratgan modelimizning xatosini tekshiramiz. Buning uchun **Model** menyusidan **Verify Model...** buyrug'ini tanlaymiz.



20.37-rasm.

Ochilgan oynadagi Verify tugmasini bosamiz:



20.38-rasm.

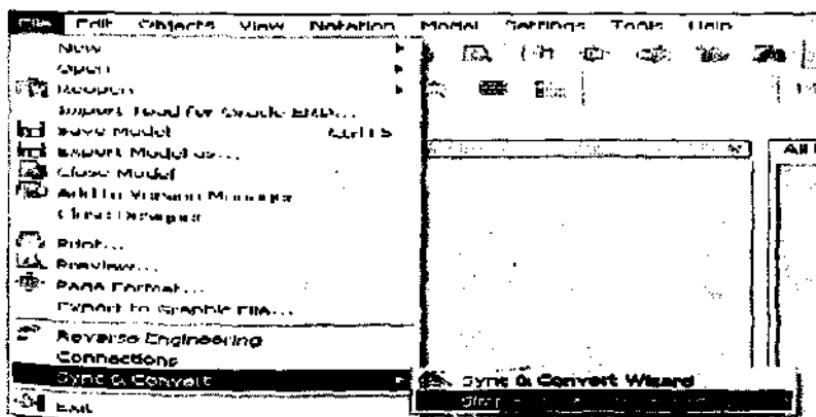
Pastdagi ro'yxatga e'tibor beramiz:

**Number of Errors: 0**

**Number of Warnings: 0**

YA'ni xato yo'q.

Universal model tayyor bo'ldi. Endi bu tayyor model ustida ixtiyoriy ishlarni amalga oshirish mumkin. Biz esa, File → Sync & Convert → Simple Model Conversion... buyruqlar ketma-ketligini bajaramiz:



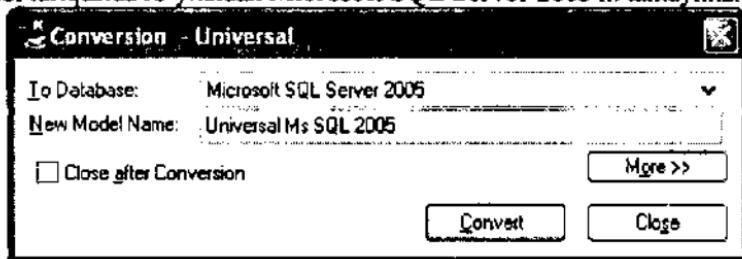
20.39-rasm.

Quyidagi ro'yxatdan kerakli BGBT tanlab konvertatsiya qilishini mumkin.

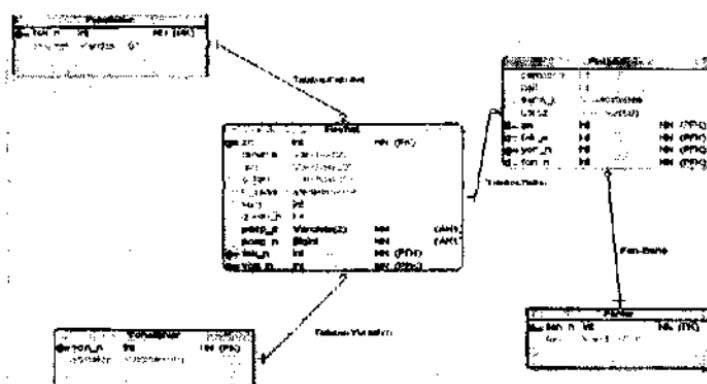


20.40-rasm.

Misol tarigasida ro'yxatdan Microsoft SQL Server 2005 ni tanlaymiz.



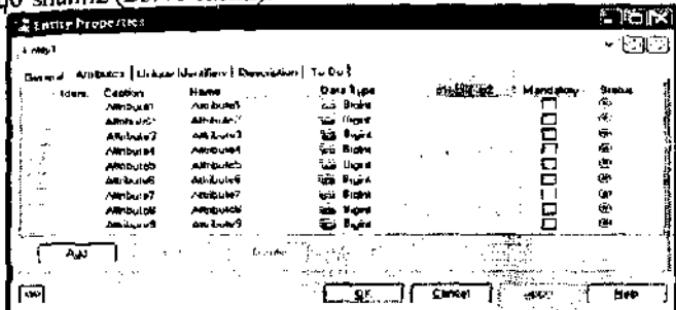
Convert tugmasini bosamiz va quyidagi fizik modelni hosil qilamiz:



## Berilganlarning mantiqiy modelini yaratish

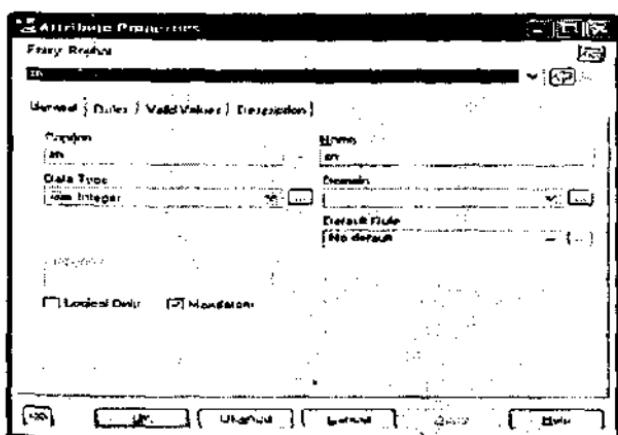
Endi loyihamizga mantiqiy modelni qo'shamiz. Buning uchun 20.3-rasmdan 20.8-rasingacha bo'lgan bosqichdag'i ishlarni takrorlaymiz:

- modelga Mantiqiy deb nom beramiz;
- Logical Model Explorer → Yentities → Add Yentity → All Items → Yentity1 → Royhat
- Attributes bo'limiga o'tamiz va Add tugmasi yordamida 9 ta atribut qo'shamiz (20.41-rasm.).



20.41-rasm.

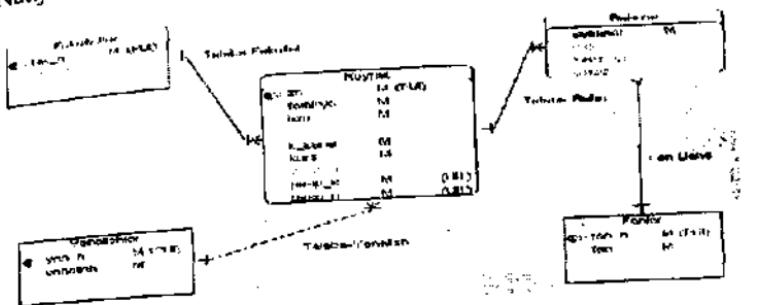
Edit tugmasi yordamida Attribute Properties oynasini ochamiz va atributning xususiyatlarini beramiz. Zn deb nom beramiz, turini Integer deb tanlaymiz, Mandatory qiymat berish shart, degan belgini kiritamiz va OK tugmasini bosamiz.



20.42-rasm.

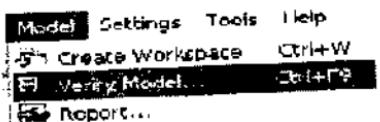
- Mantiqiy model ustida bajariladigan qolgan vazifalarni bajarish uchun 20.10-rasmdan 20.37-rasmgacha bo'lgan bosqichdagi ishlarni takrorlaymiz.
- keyingi atribut **familiya**, turi **Varchar**, uzunligi **length 20** ta berilish OK;
  - qolgan atributlarni ham shu tariqa to'ldirib chiqamiz;
  - ichki kalitlarni kiritamiz. Buning uchun **Unique Identifiers** bo'limiga o'tib, **Add** tugmasi yordamida 2 ta kalit qo'shamiz – **zn** atribut uchun **pk\_zn** 1-darajali noyob kalitni, hamda **rasp\_s**, **pasp\_n** atributlariga tarkibli **u\_pasport** noyob kalit qo'yamiz. Buning uchun **Edit** tugmasini bosib **pasp\_s**, **pasp\_n** atributlarini ro'yxatning o'ng qisimiga o'tkazamiz, OK. Royhat munosabatimiz tayyor holatga keladi;
  - **Fakultetlar** munosabatini yaratamiz. Buning uchun **Entities** papkasining ustiga o'ng tugmani bosib, **Add Entity** buyrug'ini tanlaymiz va **All Items** oynasida hosil bo'lgan yangi munosabatga **Fakultetlar** deb nom beramiz;
  - **attributes** bo'limiga o'tib, atributlarini kiritamiz;
  - **Unique Identifiers** bo'limiga o'tib, **pk\_fak\_n** kalit qo'shamiz, OK tugmasini bosamiz. **Fakultetlar** munosabati ham tayyor holatga keladi;
  - **Yo'nalishlar** munosabatini yaratamiz. **Attributes** bo'limga o'tib atributlarini kiritamiz;
  - **Unique Identifiers** bo'limiga o'tib **pk\_yon\_n** kalit qo'shamiz, OK tugmasini bosamiz. **Yo'nalishlar** munosabati ham tayyor holatga keladi;
  - navbatdagi munosabat **Baholar**. **Attributes** bo'limga o'tib atributlarini kiritamiz; OK tugmasini bosamiz. **Baholar** munosabati ham tayyor holatga keladi;
  - so'nggi munosabat **Fanlar**. **Attributes** bo'limga o'tib atributlarini kiritamiz;
  - **Unique Identifiers** bo'limiga o'tib, **pk\_fan\_n** kalit qo'shamiz, OK tugmasini bosamiz. **Fanlar** munosabati ham tayyor holatga keladi;
  - munosabatlarni tashqi kalitlar yordamida bog'lab chiqamiz. **Objects** menyusidan **Relationship** buyrug'ini tanlaymiz. Keyin **Fakultetlar** munosabatining ustiga sichqonchani bosib, **Royhat** munosabatining ustiga sichqonchani bosamiz. Hosil bo'lgan chiziqning ustiga 2 marta bosib, tashqi kalitni xususiyatlarini o'zgartirish mumkin. Bu aloqani **Talaba-Fakultet** deb nomlaymiz;
  - keyingi tashqi kalit **Yo'nalishlar** va **Royhat** munosabatlari orasida bo'ladi. Bu aloqani **Talaba-Yo'nalish** deb nomlaymiz;
  - keyingi tashqi kalit **Royhat** va **Baholar** munosabatlari orasida bo'ladi. Bu aloqani **Talaba-Baho** deb nomlaymiz;

- so'nggi aloqa **Fanlar** va **Baholar** munosabatlari orasida bo'ladi. Bu aloqani **Fan-Baho** deb nomlaymiz.
- Natijada ekranда quyidagicha chizma hosil bo'ladi:



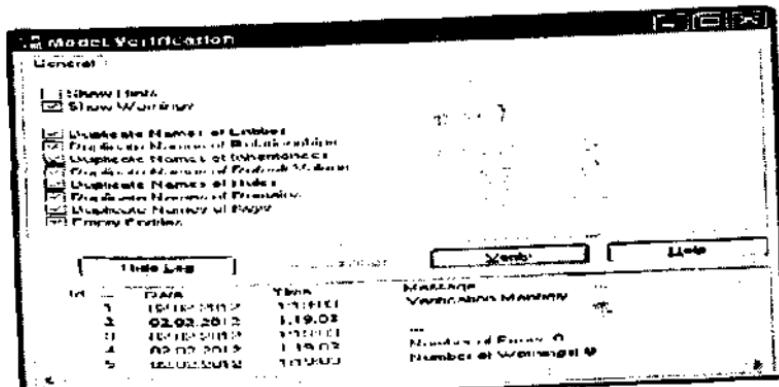
20.43-rasm.

Endi, yaratilgan mantiqiy modelimizning xatosini tekshiramiz. Buning uchun **Model** menyusidan **Verify Model...** buyrug'ini tanlaymiz.



20.44-rasm.

Ochilgan oynadagi **Verify** tugmasini bosamiz:



20.45-rasm.

20.45-rasmda keltirilgan ekran pastidagi ro'yxatga e'tibor beramiz:

**Number of Errors: 0**

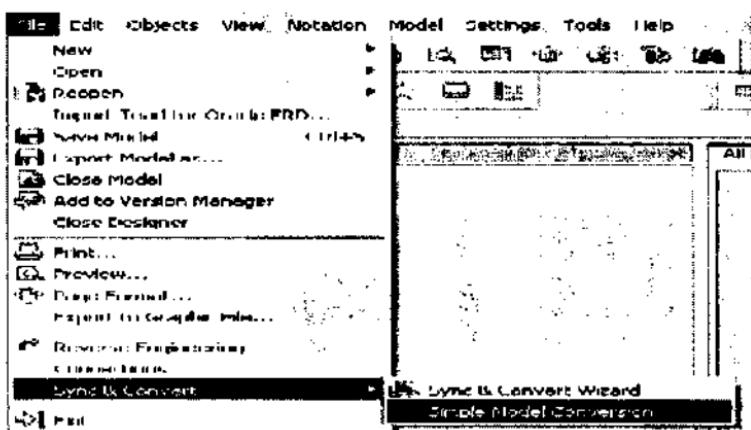
**Number of Warnings: 0**

Ya'ni xato yo'q.

Mantiqiy model tayyor bo'ldi.

### Berilganiarning fizik modellashtirish

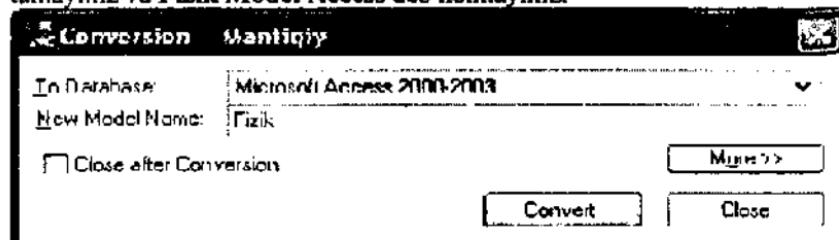
Berilganiarning fizik modellashtirish uchun mantiqiy modelni fizik ko'rinishga o'tkazamiz. Buning uchun **File -> Sync & Convert -> Simple Model Conversion...** buyruqlar ketma-ketligini bajaramiz:



20.46-rasm.

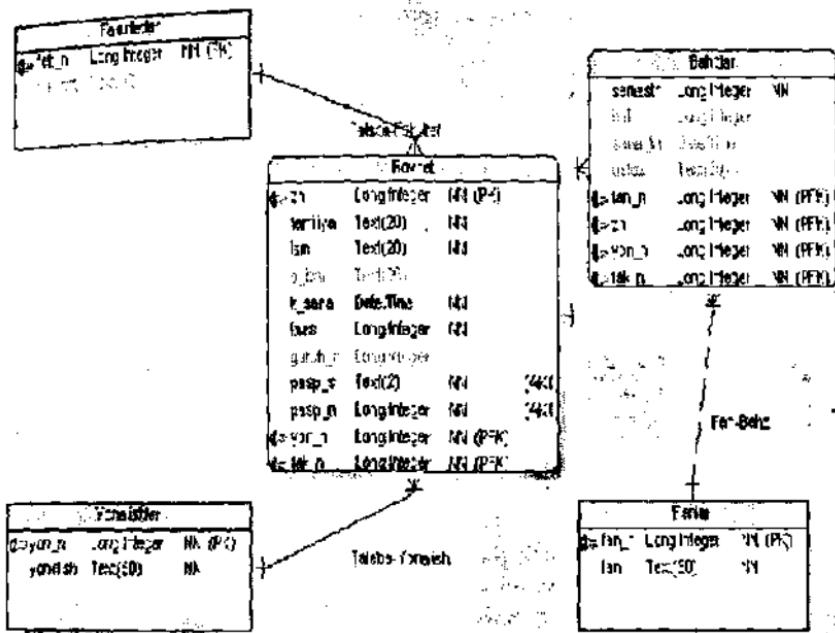
Biz yanada tushunarli bo'lishi uchun fizik modelni 2 xil ko'rinishda yaratamiz.

Ochilgan oynadan To Database ga Microsoft Access 2000-2003 ni tanlaymiz va **Fizik Model Access** deb nomlaymiz.



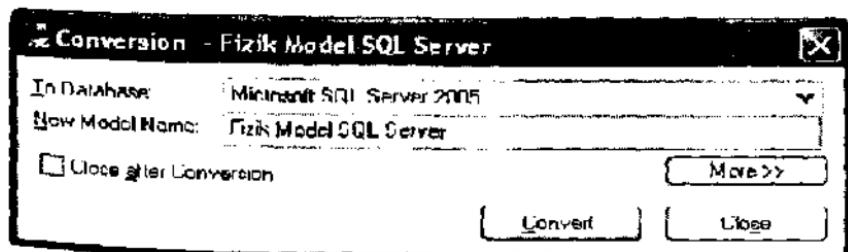
20.47-rasm.

Convert tugmasini bosamiz. Natijada MS Access uchun mos keladigan fizik model hosil bo'ladi.



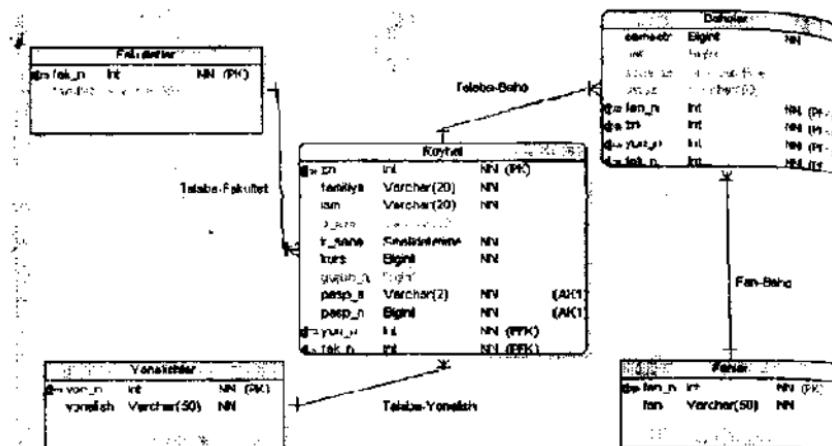
20.48-rasm.

Ochilgan oynadan To Database ga MS SQL Server 2005 ni tanlaymiz va Fizik Model SQL Server deb nomlaysiz.



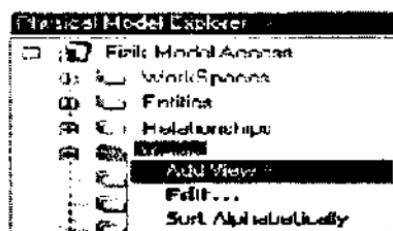
20.49-rasm.

Convert tugmasini bosamiz. Natijada Microsoft SQL Server uchun mos keladigan fizik model hosil bo'ladi.



20.50-rasm.

Endi hosil bo'lgan fizik modelga **Talaba\_v** va **Baholar\_v** nomli 2 so'rov qo'shamiz. So'rov qo'shish uchun, joriy fizik modeldagi View papkasi ustida o'ng tugmani bosib, Add View buyrug'ini tanlaymiz.



20.51-rasm.

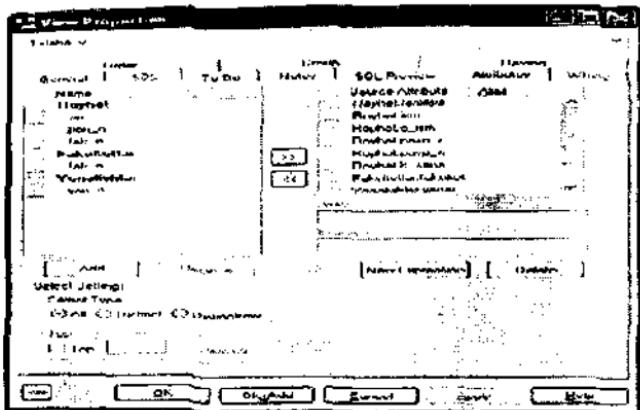
Natijada ekranida quyidagi ko'rinishdagi



20.52-rasm.

Yangi so'rov paydo bo'ladi. Bu so'rovning ustida sichqonchani 2 marta bosamiz va so'rovning xususiyatlarini to'liq ko'ssatamiz.

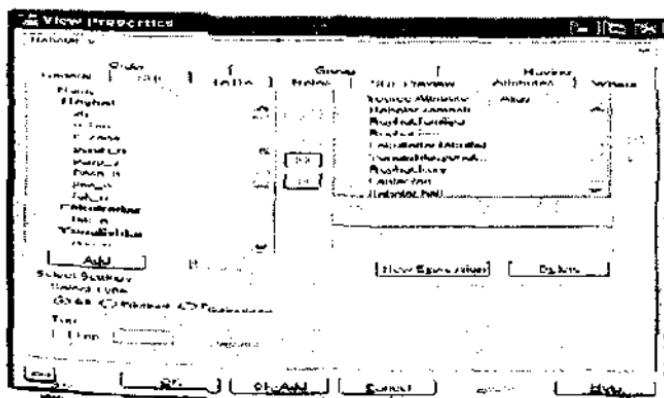
**General** bo'limida so'rovga nom beramiz. **Attributes** bo'limidan so'rov uchun kerakli atriibutlarni chap ro'yxatdan o'ng torxonadagi ro'yxaiga o'tkazamiz:



20.53-rasm.

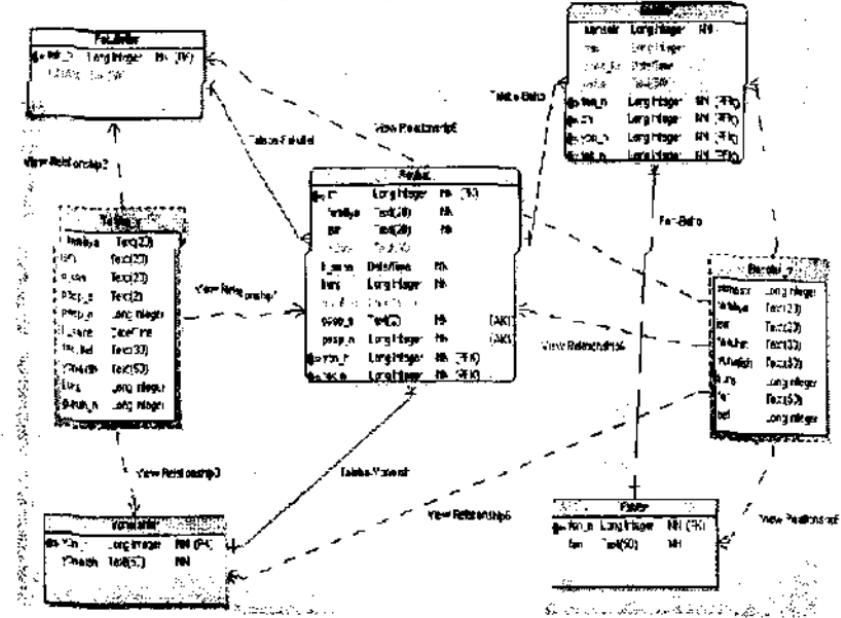
Where bo'limiga o'tib, «**Royhat.fak\_n = Fakultellar.fak\_n** and **Royhat.yon\_n = Yo'nalishlar.yon\_n**» deb yozamiz va **OK** tugmasini bosamiz.

**Baholar\_v** so'rovini ham **Talaba\_v** so'rovi singari yasaymiz:



20.54-rasm.

Natijada 20.55-rasmdagi ER-diagramma hosil bo'ldi:

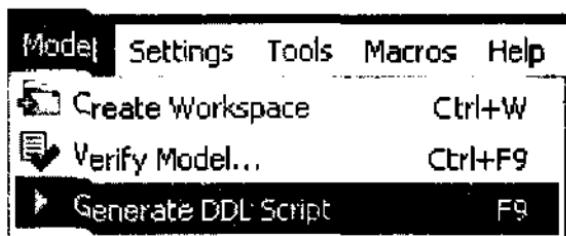


20.55-rasm.

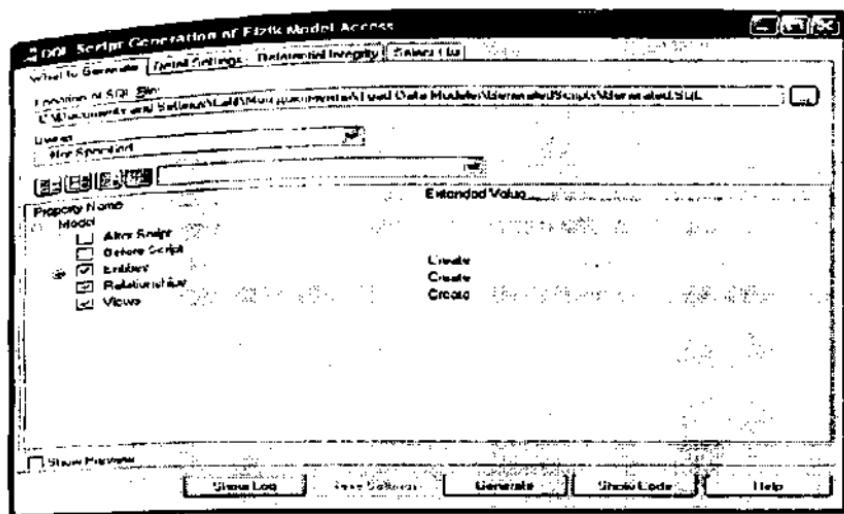
Talabalar o'quv loyihasining berilganlar bazasi tayyor holatga keldi.

#### MS Access uchun DDL Script

Endi MS Access uchun DDL Script ni hosil qilishdan iborat. Buning uchun Model menyusidan Generate DDL Script buyrug'ini tanlaymiz:



20.56-rasm.



20.57-rasm.

Natijada 20.57-rasmdagi oyna ochiladi. Bu yerdan generatsiya qilish uchun zarur bo'lgan obyektlarni belgilab, **Generate** tugmasini bosamiz.

**Fizik Model Access** ni generatsiya qilganimizda, quyidagicha **DDL Script** hosil bo'ladi:

'Created: 02.02.2012

'Modified: 04.02.2012

'Model: Fizik Model Access

'Database: MS Access 2000-2003

---

#### ==== Microsoft Access 2000/2002/2003 database creation method

---

==== 1. Create a new database in the Microsoft Access

==== 2. Create a new module

==== 3. Copy the TDM3 output SQL script into the new Microsoft Access module

==== 4. Select from main menu «Tools» item «References...» and check

==== the «Microsoft ActiveX Data ObjecS 2.x Library»

==== and «Microsoft ADO Yext. 2.x for DDL and Security»

==== and «Microsoft DAO 3.6 Object Library»

==== 5. Place your mouse cursor somewhere in the main procedure Main()

==== 6. Run the module code (Click the «Run Sub/UserForm» button or press F5)

---

```

Public con As New ADOX.Connection
Public cat As New ADOX.Catalog
Public tbl As ADOX.Table
Public idx As ADOX.Index
Public dbs As DAO.Database
Sub Main()
Set con = CurrentProject.Connection
cat.ActiveConnection = con
Set dbs = CurrentDb()
On Error GoTo ErrorHandler
    Call CreateTables
    Call CreatePrimaryKeys
    Call CreateIndexes
    Call CreateAlterKeys
    Call CreateRelations
    Call CreateRelationsDAO
    Call CreateQueries
    MsgBox «Script successfully processed!», vbInformation
    Exit Sub
ErrorHandler:
Select Case Err.Number
    Case -2147217857 'DAO: 3010
        MsgBox «Table « & tbl.Name & « already exist!», vbInformation
        Err.Clear
    Case -2147217868 'DAO: 3284
        MsgBox «Index « & idx.Name & « for table « & tbl.Name & « already
exist!», vbInformation
        Err.Clear
    Case Yelse
        MsgBox Yerr.Description, vbCritical
End Select
End Sub
' Create tables
'=====
Sub CreateTables()
Call CreateTable1 'Royhat
Call CreateTable2 'Fakultetlar
Call CreateTable3 'Yonalishlar
Call CreateTable4 'Baholar
Call CreateTable5 'Fanlar
End Sub

```

```

==== Create table Royhat =====
Sub CreateTable1()
Set tbl = New ADOX.Table
tbl.Name = «Royhat»
cat.Tables.Append tbl
Call AddFieldToTable(«Royhat», «zn», adInteger, 0, 0, «», «FALSE», ««,
««, ««, ««, ««)
Call AddFieldToTable(«Royhat», «familiya», adVarWChar, 20, 0, ««,
«FALSE», ««, ««, ««, ««)
Call AddFieldToTable(«Royhat», «ism», adVarWChar, 20, 0, ««,
«FALSE», ««, ««, ««, ««)
Call AddFieldToTable(«Royhat», «o_ism», adVarWChar, 20, 0, ««,
«TRUE», ««, ««, ««, ««, ««)
Call AddFieldToTable(«Royhat», «k_sana», adDate, 0, 0, «», «FALSE»,
««, ««, ««, ««, ««)
Call AddFieldToTable(«Royhat», «kurs», adInteger, 0, 0, «», «FALSE», ««,
««, ««, ««, ««)
Call AddFieldToTable(«Royhat», «guruh_n», adInteger, 0, 0, ««, «TRUE»,
««, ««, ««, ««, ««)
Call AddFieldToTable(«Royhat», «pasp_s», adVarWChar, 2, 0, ««,
«FALSE», ««, ««, ««, ««, ««)
Call AddFieldToTable(«Royhat», «pasp_n», adInteger, 0, 0, «», «FALSE»,
««, ««, ««, ««, ««)
Call AddFieldToTable(«Royhat», «yon_n», adInteger, 0, 0, «», «FALSE»,
««, ««, ««, ««, ««)
Call AddFieldToTable(«Royhat», «fak_n», adInteger, 0, 0, «», «FALSE»,
««, ««, ««, ««, ««)
dbs.TableDefs.Refresh
Call AddPropertyToFieldDAO( «Royhat», »zn», »Caption», »zn», dbText)
Call AddPropertyToFieldDAO( «Royhat», »familiya», »Caption»,
«familiya», dbText)
Call AddPropertyToFieldDAO( «Royhat», »ism», »Caption», »ism», dbText)
Call AddPropertyToFieldDAO («Royhat», «o_ism», «Caption», «o_ism»,
dbText)
Call AddPropertyToFieldDAO( «Royhat», »k_sana», »Caption», »k_sana»,
dbText)
Call AddPropertyToFieldDAO(«Royhat», »kurs», »Caption», »kurs», dbText)
Call AddPropertyToFieldDAO(«Royhat», »guruh_n», »Caption», »guruh_n»,
dbText)
Call AddPropertyToFieldDAO («Royhat», «pasp_s», «Caption», «pasp_s»,
dbText)

```

```

Call AddPropertyToFieldDAO(«Royhat», »pasp_n», »Caption», »pasp_n»,
dbText)
Call AddPropertyToFieldDAO( «Royhat», »yon_n», »Caption», »yon_n»,
dbText)
Call AddPropertyToFieldDAO(«Royhat», »fak_n», »Caption», »fak_n»,
dbText)
End Sub
'==== Create table Fakultetlar =====
Sub CreateTable2()
Set tbl = New ADOX.Table
tbl.Name = «Fakultetlar»
cat.Tables.Append tbl
Call AddFieldToTable(«Fakultetlar», «fak_n», adInteger, 0, 0, ««,
«FALSE», ««, ««, ««, ««)
Call AddFieldToTable(«Fakultetlar», «fakultet», adVarWChar, 30, 0, ««,
«TRUE», ««, ««, ««, ««)
dbs.TableDefs.Refresh
Call AddPropertyToFieldDAO(«Fakultetlar», »fak_n», »Caption», »fak_n»,
dbText)
Call AddPropertyToField-
DAO(«Fakultetlar», »fakultet», »Caption», »fakultet», dbText)
End Sub
'==== Create table Yo'nalishlar =====
Sub CreateTable3()
Set tbl = New ADOX.Table
tbl.Name = «Yo'nalishlar»
cat.Tables.Append tbl
Call AddFieldToTable(«Yo'nalishlar», «yon_n», adInteger, 0, 0, ««,
«FALSE», ««, ««, ««, ««)
Call AddFieldToTable(«Yo'nalishlar», «yo'nalish», adVarWChar, 50, 0,
««, «FALSE», ««, ««, ««, ««)
dbs.TableDefs.Refresh
Call AddPropertyToFieldDAO(«Yo'nalishlar», »yon_n», »Caption», »
yon_n», dbText)
Call AddPropertyToFieldDAO(«Yo'nalishlar», »yo'nalish», »Caption»,
«yo'nalish», dbText)
End Sub
'==== Create table Baholar =====
Sub CreateTable4()
Set tbl = New ADOX.Table
tbl.Name = «Baholar»

```

```

cat.Tables.Append tbl
Call AddFieldToTable(«Baholar», «semestr», adInteger, 0, 0, ««,
«FALSE», ««, ««, ««, ««, ««)
Call AddFieldToTable(«Baholar», «ball», adInteger, 0, 0, ««, «TRUE», ««,
««, ««, ««, ««)
Call AddFieldToTable(«Baholar», «sana_kir», adDate, 0, 0, ««, «TRUE»,
««, ««, ««, ««, ««)
Call AddFieldToTable(«Baholar», «ustoz», adVarWChar, 60, 0, ««,
«TRUE», ««, ««, ««, ««)
Call AddFieldToTable(«Baholar», «fan_n», adInteger, 0, 0, ««, «FALSE»,
««, ««, ««, ««, ««)
Call AddFieldToTable(«Baholar», «zn», adInteger, 0, 0, ««, «FALSE», ««,
««, ««, ««, ««, ««)
Call AddFieldToTable(«Baholar», «yon_n», adInteger, 0, 0, ««, «FALSE», ««,
««, ««, ««, ««, ««)
Call AddFieldToTable(«Baholar», «fak_n», adInteger, 0, 0, ««, «FALSE»,
««, ««, ««, ««, ««)
dbs.TableDefs.Refresh
Call AddPropertyToFieldDAO( «Baholar», »semestr», »Caption», »semestr»,
dbText)
Call AddPropertyToFieldDAO( «Baholar», »ball», »Caption», »ball», dbText)
Call AddPropertyToFieldDAO( «Baholar», »sana_kir», »Caption», »sana_kir», dbText)
Call AddPropertyToFieldDAO( «Baholar», «ustoz», «Caption», «ustoz», dbText)
Call AddPropertyToFieldDAO( «Baholar», «fan_n», «Caption», «fan_n», dbText)
Call AddPropertyToFieldDAO( «Baholar», »zn», »Caption», »zn», dbText)
Call AddPropertyToFieldDAO( «Baholar», «yon_n», «Caption», «yon_n», dbText)
Call AddPropertyToFieldDAO( «Baholar», «fak_n», «Caption», «fak_n», dbText)
End Sub
'== Create table Fanlar =====
Sub CreateTable5()
Set tbl = New ADQX.Table
tbl.Name = «Fanlar»
cat.Tables.Append tbl
Call AddFieldToTable(«Fanlar», «fan_n», adInteger, 0, 0, ««, «FALSE»,
««, ««, ««, ««, ««))

```

```

Call AddFieldToTable(«Fanlar», «fan», adVarWChar, 50, 0, «», «FALSE»,
«», «», «», «», «»)
dbs.TableDefs.Refresh
Call AddPropertyToFieldDAO( «Fanlar», «fan_n», «Caption», «fan_n»,
dbText)
Call AddPropertyToFieldDAO( «Fanlar», «fan», «Caption», «fan», dbText)
End Sub
' Create primary keys
'=====
Sub CreatePrimaryKeys()
'== Create primary key for table Royhat =====
Set idx = New ADOX.Index
Set tbl = cat.Tables(«Royhat»)
idx.Name = «pk_zn»
idx.PrimaryKey = True
idx.Unique = True
idx.IndexNulls = adIndexNullsDisallow
Call AddFieldToIndex(idx, «zn», False )
Call AddFieldToIndex(idx, «yon_n», False )
Call AddFieldToIndex(idx, «fak_n», False )
tbl.Indexes.Append idx
'== Create primary key for table Fakultetlar =====
Set idx = New ADOX.Index
Set tbl = cat.Tables(«Fakultetlar»)
idx.Name = «pk_fak_n»
idx.PrimaryKey = True
idx.Unique = True
idx.IndexNulls = adIndexNullsDisallow
Call AddFieldToIndex(idx, «fak_n», False )
tbl.Indexes.Append idx
'== Create primary key for table Yo'nalishlar =====
Set idx = New ADOX.Index
Set tbl = cat.Tables(«Yo'nalishlar»)
idx.Name = «pk_yon_n»
idx.PrimaryKey = True
idx.Unique = True
idx.IndexNulls = adIndexNullsDisallow
Call AddFieldToIndex(idx, «yon_n», False )
tbl.Indexes.Append idx
'== Create primary key for table Baholar =====
Set idx = New ADOX.Index

```

```

Set tbl = cat.Tables("«Baholar»")
idx.Name = «Unique_Identifier»
idx.PrimaryKey = True
idx.Unique = True
idx.IndexNulls = adIndexNullsDisallow
Call AddFieldToIndex(idx, «fan_n», False )
Call AddFieldToIndex(idx, «zn», False )
Call AddFieldToIndex(idx, «yon_n», False )
Call AddFieldToIndex(idx, «fak_n», False )
tbl.Indexes.Append idx
'== Create primary key for table Fanlar ==
Set idx = New ADOX.Index
Set tbl = cat.Tables("«Fanlar»")
idx.Name = «pk_fan_n»
idx.PrimaryKey = True
idx.Unique = True
idx.IndexNulls = adIndexNullsDisallow
Call AddFieldToIndex(idx, «fan_n», False )
tbl.Indexes.Append idx
End Sub
'Create indexes
=====
Sub CreateIndexes()
End Sub
' Create alternate keys (unique indexes in Microsoft Access)
=====
Sub CreateAlterKeys()
'== Create alternate key (unique index) for table Royhat ==
Set idx = New ADOX.Index
Set tbl = cat.Tables("«Royhat»")
idx.Name = «u_pasport»
idx.Unique = True
idx.IndexNulls = adIndexNullsAllow
Call AddFieldToIndex(idx, «pasp_s», False )
Call AddFieldToIndex(idx, «pasp_n», False )
tbl.Indexes.Append idx
End Sub
'Create relations
=====
Sub CreateRelations()
Dim keyFk As ADOX.Key

```

==== Create relationship between parent table Fanlar and child table Baholar

```
Set keyFk = New ADOX.Key  
Set tbl = cat.Tables("Baholar")  
keyFk.Name = "Fan-Baho"  
keyFk.Type = adKeyForeign  
keyFk.RelatedTable = "Fanlar"  
Call AddFieldToRelation(keyFk, "fan_n", "fan_n")  
tbl.Keys.Append keyFk
```

==== Create relationship between parent table Royhat and child table Baholar

```
Set keyFk = New ADOX.Key  
Set tbl = cat.Tables("Baholar")  
keyFk.Name = "Talaba-Baho"  
keyFk.Type = adKeyForeign  
keyFk.RelatedTable = "Royhat"  
Call AddFieldToRelation(keyFk, "zn", "zn")  
Call AddFieldToRelation(keyFk, "yon_n", "yon_n")  
Call AddFieldToRelation(keyFk, "fak_n", "fak_n")  
tbl.Keys.Append keyFk
```

==== Create relationship between parent table Yo'nalishlar and child table Royhat

```
Set keyFk = New ADOX.Key  
Set tbl = cat.Tables("Royhat")  
keyFk.Name = "Talaba-Yo'nalish"  
keyFk.Type = adKeyForeign  
keyFk.RelatedTable = "Yo'nalishlar"  
Call AddFieldToRelation(keyFk, "yon_n", "yon_n")  
tbl.Keys.Append keyFk
```

==== Create relationship between parent table Fakultetlar and child table Royhat

```
Set keyFk = New ADOX.Key  
Set tbl = cat.Tables("Royhat")  
keyFk.Name = "Talaba-Fakultet"  
keyFk.Type = adKeyForeign  
keyFk.RelatedTable = "Fakultetlar"  
Call AddFieldToRelation(keyFk, "fak_n", "fak_n")  
tbl.Keys.Append keyFk
```

End Sub

' Create relations (DAO)

```

Sub CreateRelationsDAO()
    Dim rel As DAO.Relation
    dbs.Relations.Refresh
End Sub
' Create queries
=====
Sub CreateQueries()
    Dim cmd As ADODB.Command
    '== Create query Talaba_v ==
    Set cmd = New ADODB.Command
    cmd.CommandText=«SELECT familiya, ism, o_ism, pasp_s, pasp_n,
    k_sana, fakultet, yo'nalish, kurs, guruh_n « + «FROM Royhat, Fakultetlar,
    Yo'nalishlar» _ + «WHERE Royhat.fak_n = Fakultetlar.fak_n and
    Royhat.yon_n = Yo'nalishlar.yon_n ; «
    cat.Views.Append «Talaba_v», cmd
    '== Create query Baholar_v ==
    Set cmd = New ADODB.Command
    cmd.CommandText=«SELECT semestr, familiya, ism, fakultet, yo'nalish,
    kurs, fan, ball « _ + »FROM Royhat, Fakultetlar, Yo'nalishlar, Baholar,
    Fanlar; «
    cat.Views.Append «Baholar_v», cmd
End Sub
' Add fields to table
=====

Sub AddFieldToTable(TableName As String, FieldName As String,
    DataType As Integer, SizePrecCol As Integer, ScaleCol As Integer,
    Attributes As String,
    Nullable As String, DefaultValue As Variant,
    AllowZeroLength As String, CompressUnicode As String,
    ValText As String, ValRule As String)
    Dim col As New ADOX.Column
    col.ParentCatalog = cat
    col.Name = FieldName
    col.Type = DataType
    If DataType = adVarWChar Then col.DefinedSize = SizePrecCol
    If DataType = adNumeric Then
        col.Precision = SizePrecCol
        col.NumericScale = ScaleCol
    End If
    If Nullable <> «« Then col.Properties(«Nullable»).Value = CBool(Nullable)
    If Attributes <> «« Then col.Properties(Attributes).Value = True

```

```

If AllowZeroLength <> «« Then col.Properties(«Jet OLEDB:Allow Zero
Length»).Value = CBool(AllowZeroLength)
If CompressUnicode <> «« Then col.Properties(«Jet OLEDB:Compressed
UNICODE Strings»).Value = CBool(CompressUnicode)
If DefaultValue <> «« Then col.Properties(«Default»).Value =
DefaultValue
If ValRule <> «« Then col.Properties(«Jet OLEDB:Column Validation
Rule»).Value = ValRule
If ValText <> «« Then col.Properties(«Jet OLEDB:Column Validation
Text»).Value = ValText
cat.Tables(TableName).Columns.Append col
End Sub
' Add DAO properties to table
'=====
Sub AddPropertyToTableDAO( TableName As String, PropertyName As
String, Value As Variant, DataType As String)
Dim prp As DAO.Property
Dim tdf As DAO.TableDef
Set tdf = dbs.TableDefs(TableName)
Set prp = tdf.CreateProperty(PropertyName, DataType, Value)
tdf.Properties.Append prp
End Sub
' Add DAO properties to field
'=====
Sub AddPropertyToFieldDAO( TableName As String, FieldName As
String, PropertyName As String, Value As Variant, DataType As String)
Dim prp As DAO.Property
Dim fld As DAO.Field
Dim tdf As DAO.TableDef
Set tdf = dbs.TableDefs(TableName)
Set fld = tdf.Fields(FieldName)
Set prp = fld.CreateProperty(PropertyName, DataType, Value)
fld.Properties.Append prp
End Sub
' Add fields to index
'=====
Sub AddFieldToIndex(idx As ADOX.Index, FieldName As String, De-
scending As Boolean )
idx.Columns.Append FieldName
If Descending = True Then idx.Columns(FieldName).SortOrder = adSort-
Descending

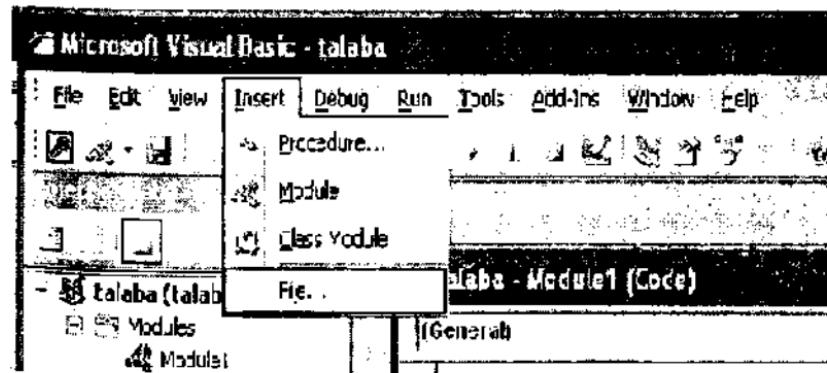
```

```

End Sub
' Add fields to relation
=====
Sub AddFieldToRelation(keyFk As ADOX.Key, PKField As String,
FKField As String)
keyFk.Columns.Append FKField
keyFk.Columns(FKField).RelatedColumn = PKField
End Sub
' Add fields to relation (DAO)
=====
Sub AddFieldToRelationDAO( rel As DAO.Relation, PKField As String,
FKField As String )
Dim fld As DAO.Field
Set fld = rel.CreateField( PKField )
fld.ForeignName = FKField
rel.Fields.Append fld
End Sub

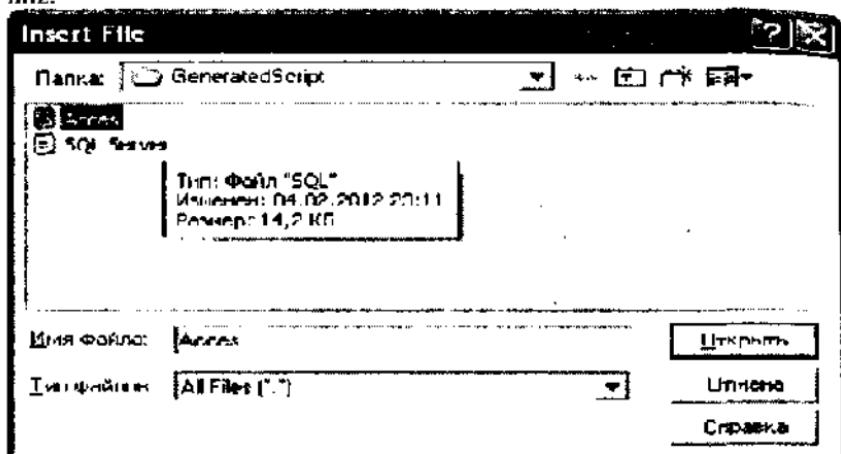
```

Endi bu hosil bo'lgan DDL Script ni MS Access 2003 da ishga tushiramiz. Buning uchun PUSK menyusidan MS Access 2003 BBB tizimiga kiramiz, **Fayl** → **Sozdat** orqali, yangi berilganlar bazasini yaratamiz. Oynaning o'ng chetida hosil bo'lgan ro'yxatdan **Novaya baza dannix** buyrug'ini tanlaymiz. Ochilgan oynada berilganlar bazasini **Talaba** nomi bilan saqlaymiz. Hosil bo'lgan Obyektlar ro'yxatidan Modular bo'limini tanlaymiz va **Sozdat** tugmasini bosamiz. **MS Visual Basic** oynasi ochiladi. Insert opsiyasidan File... buyrug'ini tanlaymiz:



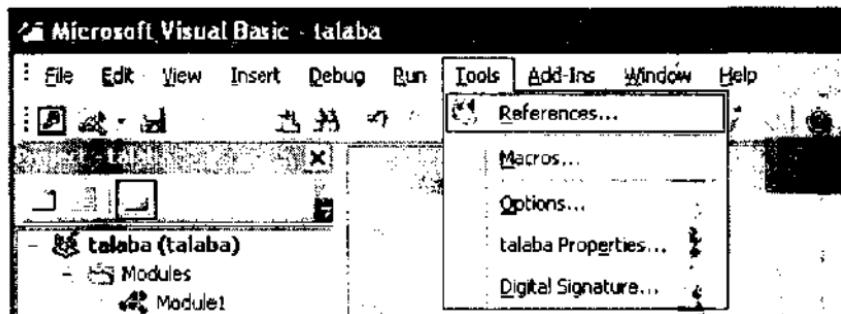
20.58-rasm.

Ochilgan oynadan MS Access uchun hosil qilingan DDL Script ni chaqiramiz:



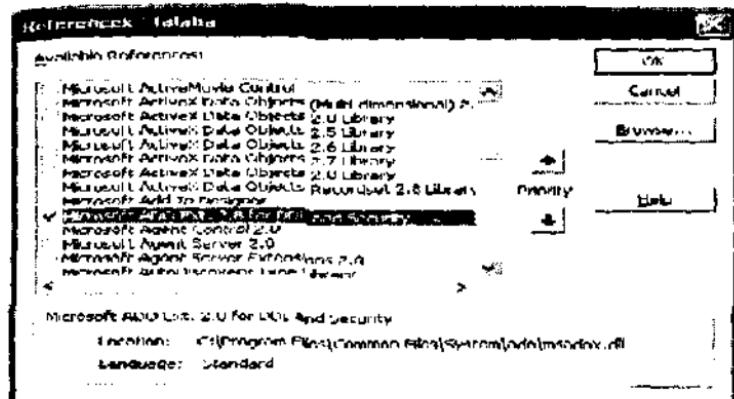
20.59R-rasm.

Agar Script ni chaqirganingizda birinchi satr boshiga «» kabi yozuv tushib qolsa, uni olib tashlash kerak. Shundan so'ng Tools menyusidan References... buyrug'ini tanlaymiz:



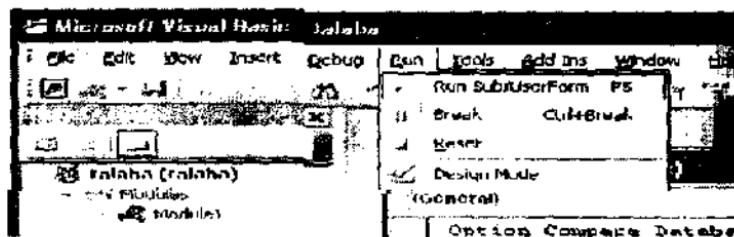
20.60-rasm.

Ochilgan oynadagi ro'yxatdan Microsoft ADO Ext. 2.8 for DDL and Security modulini yoqamiz:



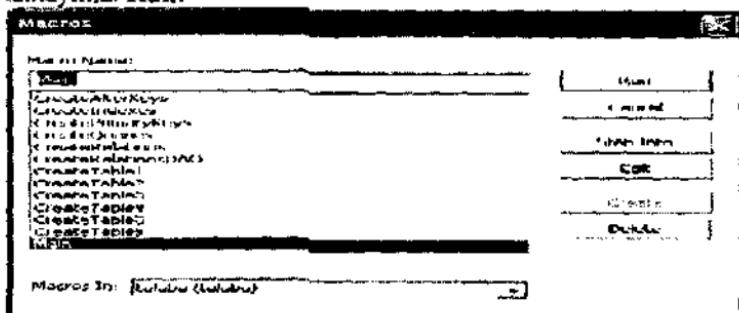
20.61-rasm.

Keyin Run buyrug'ini beramiz:

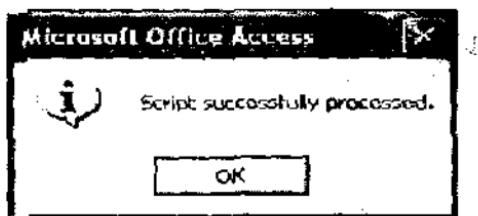


20.62-rasm.

Macros oynasidan boshlang'ich funksiyamizni, ya'nı Main funksiya-sini tanlaymiz. Run:

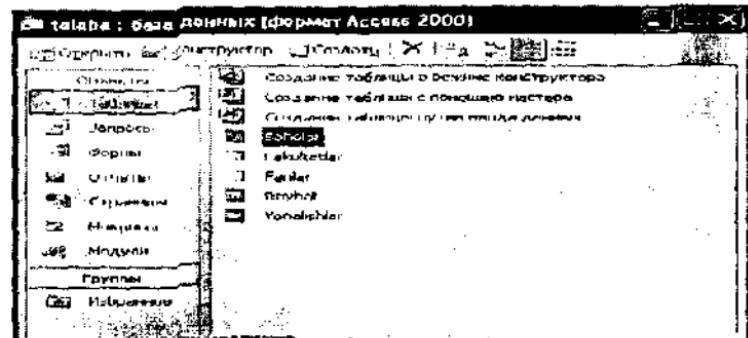


20.63-rasm.

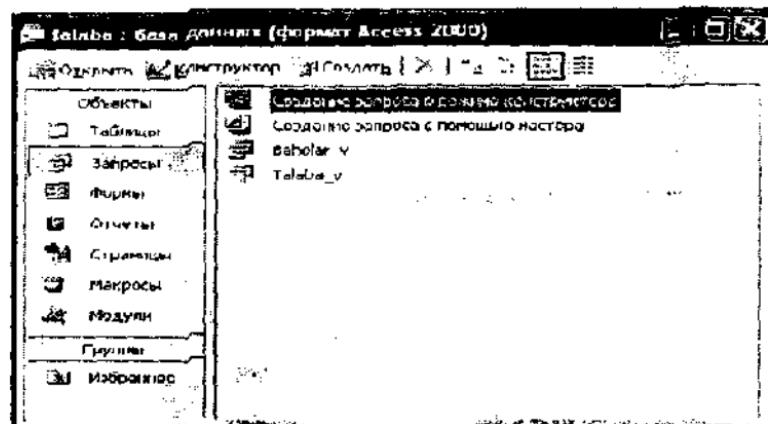


20.64-rasm.

MS Visual Basic oynasini yopamiz. Bizga kerakli jadvallar va so'rovlar hosil bo'lganini ko'ramiz:



20.65-rasm.



20.66-rasm.

Berilgan bazasi ishlash uchun tayyor holatga keltirildi.

### MS SQL Server uchun DDL Script

Fizik Model SQL Server ni generatsiya qilganimizda, quyidagiicha DDL Script hosil bo'ladi:

```
/*
Created: 02.02.2012
Modified: 04.02.2012
Model: Fizik Model SQL Server
Database: MS SQL Server 2005
*/
-- Create tables section -----
-- Table Royhat
CREATE TABLE [Royhat]
( [zn] Int NOT NULL, [familiya] Varchar(20) NOT NULL, [ism]
Varchar(20) NOT NULL,
 [o_ism] Varchar(20) NULL, [k_sana] Smalldatetime NOT NULL, [kurs]
Bigint NOT NULL,
 [guruh_n] Bigint NULL, [pasp_s] Varchar(2) NOT NULL, [pasp_n]
Bigint NOT NULL,
 [yon_n] Int NOT NULL, [fak_n] Int NOT NULL ) go
-- Add keys for table Royhat
ALTER TABLE [Royhat] ADD CONSTRAINT [pk_zn] PRIMARY KEY
(([zn],[yon_n],[fak_n]))
go ALTER TABLE [Royhat] ADD CONSTRAINT [u_pasport] UNIQUE
(([pasp_s],[pasp_n])) go
-- Table Fakultetlar
CREATE TABLE [Fakultetlar]
( [fak_n] Int NOT NULL, [fakultet] Varchar(30) NULL ) go
-- Add keys for table Fakultetlar
ALTER TABLE [Fakultetlar] ADD CONSTRAINT [pk_fak_n] PRIMARY
KEY ([fak_n]) go
-- Table Yo'nalishlar
CREATE TABLE [Yo'nalishlar]
( [yon_n] Int NOT NULL, [yo'nalish] Varchar(50) NOT NULL ) go
-- Add keys for table Yo'nalishlar
ALTER TABLE [Yo'nalishlar] ADD CONSTRAINT [pk_yon_n]
PRIMARY KEY ([yon_n]) go
-- Table Baholar
CREATE TABLE [Baholar]
```

```

([semestr] BigInt NOT NULL, [ball] BigInt NULL, [sana_kir]
Smalldatetime NULL, [ustoz] Varchar(60) NULL, [fan_n] Int NOT
NULL, [zn] Int NOT NULL, [yon_n] Int NOT NULL,
[fak_n] Int NOT NULL ) go
-- Add keys for table Baholar
ALTER TABLE [Baholar] ADD CONSTRAINT [Unique_Identifier1]
PRIMARY KEY ([fan_n],[zn],[yon_n],[fak_n]) go
-- Table Fanlar
CREATE TABLE [Fanlar]
( [fan_n] Int NOT NULL, [fan] Varchar(50) NOT NULL ) go
-- Add keys for table Fanlar
ALTER TABLE [Fanlar] ADD CONSTRAINT [pk_fan_n] PRIMARY
KEY ([fan_n]) go
-- Create views section -----
CREATE VIEW [Talaba_v] AS SELECT [familiya], [ism], [o_ism],
[pasp_s], [pasp_n], [k_sana], [fakultet], [yo'nalish], [kurs], [guruh_n]
FROM [Royhat], [Fakultetlar], [Yo'nalishlar]
WHERE Royhat.fak_n = Fakultetlar.fak_n and Royhat.yon_n =
Yo'nalishlar.yon_n go
CREATE VIEW [Baholar_v] AS SELECT [semestr], [familiya], [ism],
[fakultet], [yo'nalish], [kurs], [fan], [ball] FROM [Royhat], [Fakultetlar],
[Yo'nalishlar], [Baholar], [Fanlar] go
-- Create relationships section -----
ALTER TABLE [Baholar] ADD CONSTRAINT [Fan-Baho] FOREIGN
KEY ([fan_n]) REFERENCES [Fanlar] ([fan_n]) go
ALTER TABLE [Baholar] ADD CONSTRAINT [Talaba-Baho] FOREIGN
KEY ([zn], [yon_n], [fak_n]) REFERENCES [Royhat] ([zn], [yon_n],
[fak_n]) go
ALTER TABLE [Royhat] ADD CONSTRAINT [Talaba-Yo'nalish]
FOREIGN KEY ([yon_n]) REFERENCES [Yo'nalishlar] ([yon_n]) go
ALTER TABLE [Royhat] ADD CONSTRAINT [Talaba-Fakultet]
FOREIGN KEY ([fak_n]) REFERENCES [Fakultetlar] ([fak_n]) go

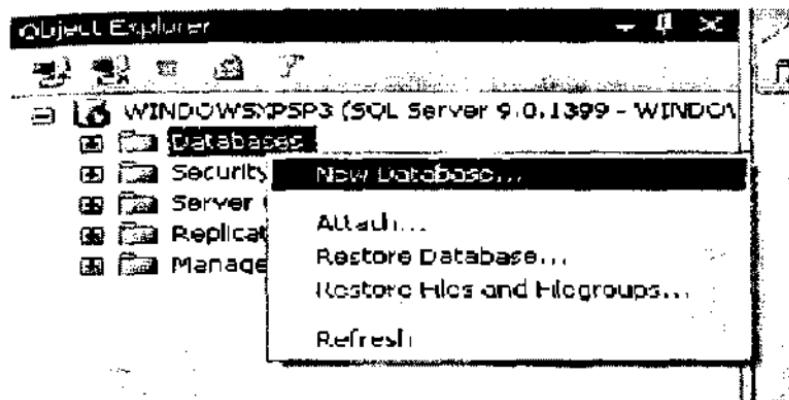
```

Endi bu hosil bo'lgan DDL Script ni MS SQL Server 2005 da ishga tushiramiz. Buning uchun PUSK menyusidan SQL Server Management Studio Yexpress programmasiga kiramiz:



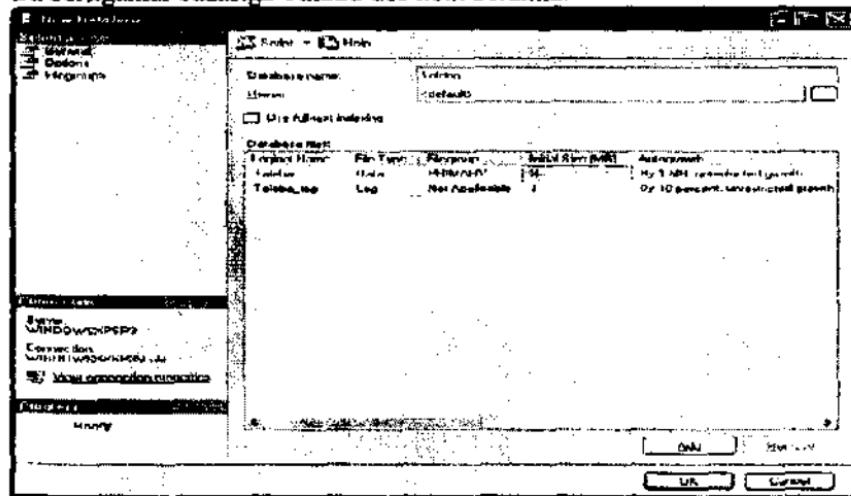
20.67-rasm.

Yangi berilganlar bazasini yaratamiz:



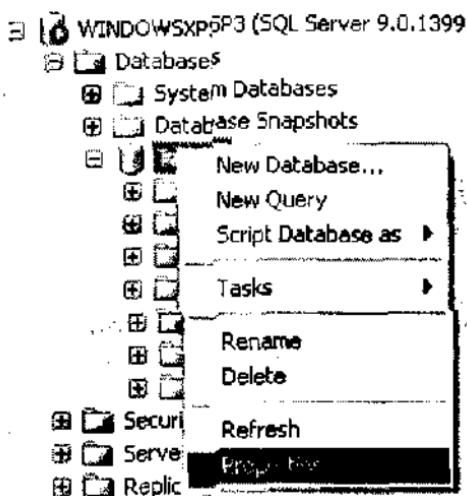
20.68-rasm.

Bu berilganlar bazasiga **Talaba** deb nom beramiz:



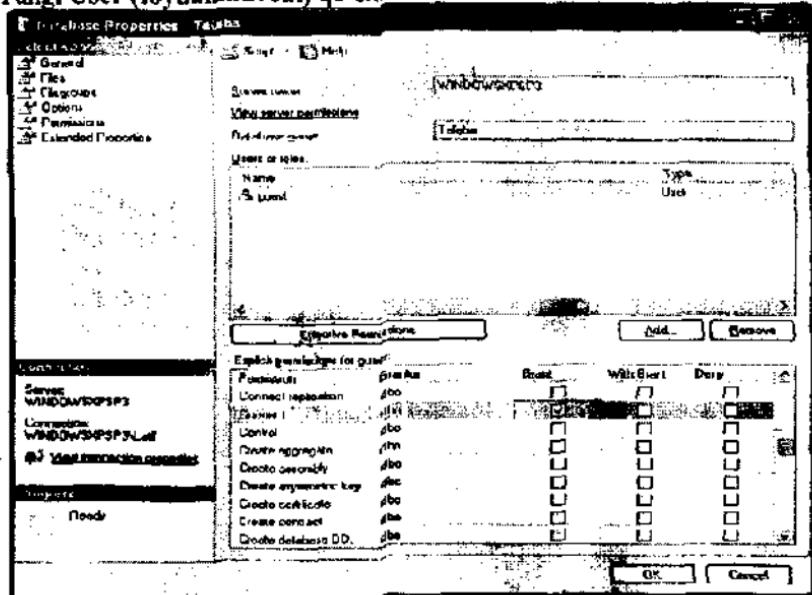
20.69-rasm.

Talaba nomli bazamizning xossasiga kiramiz



20.70-rasm.

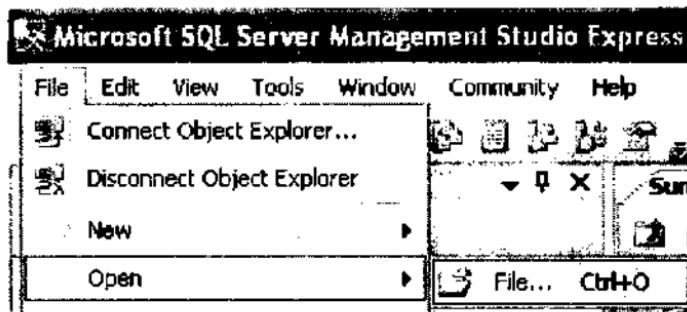
### Yangi User (soydalanuvchi) qo'shamiz



20.71-rasm.

**Connect va Yexecute buyruqlarni belgilaymiz:**

Endi Talaba deb nomlangan bazamizda **Toad Data Modeler** da yaratgan DDL Script imizni ochamiz:



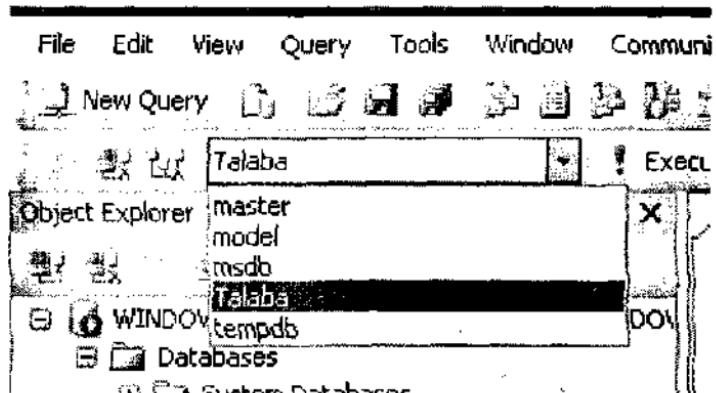
20.72-rasm.

Script imiz mana bunday ko'rinishda ochiladi:

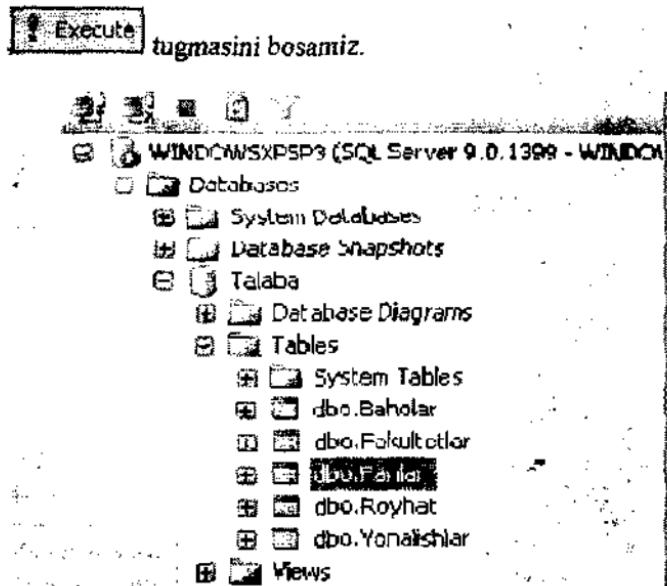
```
-- USE [master]
-- GO
-- SET ANSI_NULLS ON
-- GO
-- SET QUOTED_IDENTIFIER ON
-- GO
-- CREATE TABLE [dbo].[TBL_STUDENTS] (
--     [STUDENT_ID] INT NOT NULL,
--     [NAME] NVARCHAR(50) NOT NULL,
--     [SURNAME] NVARCHAR(50) NOT NULL,
--     [BIRTH_DATE] DATE NOT NULL,
--     [GENDER] NVARCHAR(10) NOT NULL,
--     [ADDRESS] NVARCHAR(100) NOT NULL,
--     [PHONE_NUMBER] NVARCHAR(15) NOT NULL,
--     [EMAIL] NVARCHAR(50) NOT NULL,
--     [GRADE] NVARCHAR(10) NOT NULL,
--     [CLASS] NVARCHAR(10) NOT NULL,
--     [TEACHER_ID] INT NOT NULL,
--     [REGISTRATION_DATE] DATE NOT NULL,
--     [LAST_UPDATE_DATE] DATE NOT NULL
-- );
-- GO
-- ALTER TABLE [dbo].[TBL_STUDENTS] ADD CONSTRAINT [PK_STUDENTS] PRIMARY KEY ([STUDENT_ID]);
-- GO
-- ALTER TABLE [dbo].[TBL_STUDENTS] ADD CONSTRAINT [FK_TEACHER_ID] FOREIGN KEY ([TEACHER_ID])
-- REFERENCES [dbo].[TBL_TEACHERS] ([TEACHER_ID]);
-- GO
-- CREATE INDEX [IX_STUDENTS_TEACHER_ID] ON [dbo].[TBL_STUDENTS] ([TEACHER_ID]);
```

20.73-rasm.

Quyidagi ro'yxatdan o'z bazamizni tanlaymiz:



20.74-rasm.



20.75-rasm.

Shu tariqa hamma jadvallarimiz hosil bo'lди. **Bazamiz ishlashga tayyor holatga keltirildi.**

## Xulosa

Toad Data Modeler muhitining imkoniyatlar o'rganildi va berilganlar bazasini modellashtirish Talabalar o'quv loyihasi misolida ikki usulda amalga oshirildi.

Birinchi usulda berilgan bazasining universal relation modelini yaratildi va uii aniq keltirilgan BBB tizimlaridan biriga konvertatsiya qilishgacha bo'lgan hamma bosqich amalga oshirildi.

Ikkinci usulda esa, avval mantiqiy model yaratildi va so'ngra mantiqiy model fizik modelga aylantirildi, tekshirildi, xatosiz ekanligini isbotlanib, MS Access va MS SQL Server 2005 relatsion BBB tizimlari uchun berilganlar bazasi va SQL so'rovlarini yaratildi, hamda ularning DDL skriptlari hosil qilindi. So'ngra skriptlar MS Access va MS SQL Server 2005 boshqarish tizimlariga uzatildi, MS Access va MS SQL Server 2005 boshqarish tizimlarida «Talabalar o'quv loyihasi» bo'yicha berilgan bazasi ishlash uchun tayyor holatga keltirildi.

### Nazorat uchun savollar va amaliyot uchun topshiriqlar

1. Toad Data Modeler muhitining maqsadi nimadan iborat?
2. Toad Data Modeler muhitining vazifalari nimadan iborat?
3. Qaysi relatsion BBB tizimlari uchun Toad Data Modeler muhiyi berilganlar bazasini modellashtiradi?
4. Toad Data Modeler muhitida universal, mantiqiy va fizik modellashtirishdagi o'zaro farqlar nimadan iborat?
5. Munosabatlarni modellashtirishdagi eng asosiy vazifa nimadan iborat?
6. Munosabatlar orasidagi aloqalarni o'mmatishidagi eng asosiy vazifa nimadan iborat?
7. 20.36-rasmdagi berilganlar bazasining chizmasini izohlang.
8. 20.50-rasmdagi berilganlar bazasining chizmasini izohlang.
9. 20.55-rasmdagi berilganlar bazasining chizmasini izohlang.
10. 20.36, 20.50 va 20.55-rasmdagi berilganlar bazasining chizmalari orasidagi farqlarni izohlang va nima uchun shunday ekanligini tushuntiring.
11. MS Access uchun yaratilgan DDL Script kodidagi izohlarda keltirilgan qadamlarni amalga oshiring va natijani izohlang.
12. MS SQL Server 2005 uchun yaratilgan DDL Script kodini ishlating va natijani izohlang.
13. Toad Data Modeler muhitida 7- bobdag'i 7.9-rasmida keltirilgan ER diagrammani:
  - a) universal modelini yarating va chizmani izohlang;
  - b) universal modelni fizik modelga aylantiriting;

- d) fizik modelni MS Access yoki MS SQL Server ga konvertatsiya qiling va MAXSULOT\_1 nomi bilan saqlang.
- 14. Toad Data Modeler muhitida 7- bobdagi 7.9 rasmida keltirilgan ER diagrammani:
  - a) mantiqiy modelini yarating va chizmani izohlang;
  - b) fizik modellarini yarating;
  - d) fizik modelni MS Access yoki MS SQL Server ga konvertatsiya qiling va MAXSULOT\_2 nomi bilan saqlang.
- 15. 13- va 14- topshiriqlar bo'yicha olingan chizmalmi taqqoslang va izohlang.
- 16. MAXSULOT\_1 berilganlar bazasi bo'yicha MS Access bo'yicha 3 ta murakkab so'rovlarni yarating va izohlang.
- 17. MAXSULOT\_2 berilganlar bazasi bo'yicha MS SQL Server bo'yicha 3 ta murakkab so'rovlarni yarating va izohlang.

---

## VI BO'LIM. RELATSION BERILGANLAR BAZASINI YARATISH

### 21-bo'b. Berilganlar bazasining boshqarish tizimlari

**Tayanch iboralar:** berilganlar bazasining boshqarish tizimi, berilganlarni aniqlash tili, berilganlarni manipulatsiyalash (turli amallarni bajarish, ishlov) tili, so'rovlar tili, so'rovlar protsessori, BB dispetcheri, sayllar dispetcheri, DML tilining preprotsessori, DDL tilining kompilyatori, lug'at dispetcheri, kirish huquqini nazorat modidi, komanda (buyruq)lar protsessori, butunlikni nazorat vositalari, so'rovlanri optimallashtiruvchi, tranzaksiyalar dispetcheri, rejalashtiruvchi, tiklash dispetcheri, buferlar dispetcheri, kesh dispetcheri, tizim katalogi, meta berilganlar, programmalarning berilganlardan mustaqilligi, berilganlarni abstraksiyalash, apparat ta'minot, programma ta'minoti, berilganlar, protseduralar, foydalanuvchilar, mijoz/ server, rasmiy tillar, namuna bo'yicha so'rov tili, tuzilishi so'rovlar tili, to'rtinchaviy tillar, ilovalar generatorlari, shaktlar generatori, hisobotlar generatorlari, berilganlarni grafik tasvirlash generatorlari va namoyishli generatorlar, tasavvur, berilganlar bazasini yaratuvchilar, amaliy programmachilar, oxirgi foydalanuvchilar, berilganlar administratori, berilganlar bazasining administratori, mantiqiy BB yaratuvchilar, konseptual BB loyihalash, fizik berilganlar bazasini yaratuvchi, amaliy programmachi, oddiy foydalanuvchilar, tajribali foydalanuvchilar, berilganlar bazasining xavfsizligi, parallel kirish

*Berilganlar bazasining boshqarish tizimi (BBBT)* – berilganlar bazasini yaratish, yuritish va foydalanishga mo'ljallangan programma va til vositalarining majmuasidan iborat.

BBBT – bu, foydalanuvchining amaliy programmalari va BB bilan o'zaro birqalikda ishlovchi va quyidagi imkoniyatlarga ega bo'lgan programma ta'minoti:

- foydalanuvchiga berilganlarning turlari va ularning tuzilishi, berilganlar bazasida saqlanuvchi ma'lumotlarning chegaralarini ko'rsatuvchi vositalardan iborat bo'lgan, berilganlarni aniqlash tili DDL (Data Definition Language) dan;
- DML (Data Manipulation Language) – berilganlarni manipulatsiyalash (turli amallarni bajarish, ishlov) tili kiritish, yangilash, yo'q qilish va berilganlarni bazadan olishdan. Hamma berilganlar va

ularning tavsiflarini markazlashgan holdagi bazalarda saqlash imkoniyati, DML tilini so'rovlarni tashkil etishdagi umumiy vosita sifatida foydalanish imkoniyatini beradi, uni *so'rovlar tili* (query language) deb atashadi. So'rovlar tilining mavjudligi, fayllar tizimidagi cheklangan so'rovlar bilan yoki doimiy ravishda programmalarni ko'paytirish kabi, programma ta'minotini boshqarishdagi muammo-larni bartaraf etishga olib keladi. Ana shunday protsedurasiz til bo'lgan *SQL (Structured Query Language) tili*, bugungi kundagi hamma relatsion BBBTda majburiy til sifatida ishlatalmoqda. *SQL tili* BBBTdag'i muhim ahamiyatini hisobga olgan holda biz, 23-30 boblarda u xususida to'xtab o'tamiz;

- berilganlar bazasiga nazorat ostida kirish.

Berilganlar bazasiga nazorat ostida kirish imkoniyati quyidagi **vositalar** orqali yaratib beriladi:

- foydalanuvchilar tomonidan ruxsatsiz kirishni taqiqlovchi himoya tizimi;
- berilganlarning butunligini qo'llab-quvvatlovchi, ya'ni saqlanayotgan berilganlarning qarama-qarshi holatlarini oldini olish tizimi;
- ilovalarning berilganlar bazasiga birgalikda kirish va parallel ishslash jarayonlarini nazorat qiluvchi boshqarish tizimlari;
- apparat vosita yoki programma ta'minotida inkor holatlari yuz berganda, berilganlar bazasini avvalgi ishslash mumkin bo'lgan holatlari tiklovchi tizim;
- foydalanuvchilar kirishi mumkin bo'lgan, berilganlar bazasidagi berilganlarning tavsiflarini ko'rsatish tizimi.

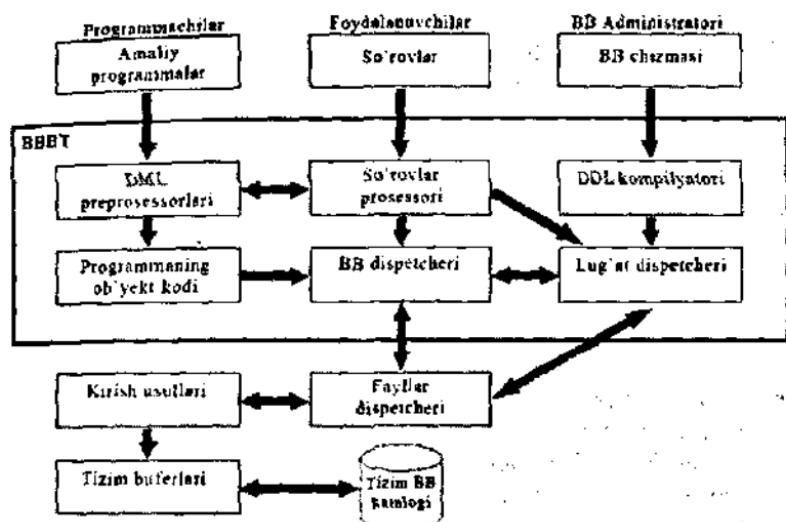
21.1-rasmda 1- bobdag'i 1.4-rasmdagi talablarga nisbatan zamonaviy BBB tizimining namunaviy varianti bo'yicha tarkibiy qismi keltirilgan.

*So'rovlar prosessori.* Bu BBBT asosiy tarkibiy qismi bo'lib, BB dispetcheri uchun quiyi darajadagi buyruqlarni hosil qiladi.

*BB dispetcheri.* Bu tarkibiy qism foydalanuvchi tomonidan ishlatalayotgan amaliy programmalar va so'rovlar bilan o'zaro harakatlanadi. BB dispetcheri so'rovlni qabul qiladi, ularni bajarish uchun tashqi va konseptual chizmalarini tekshiradi. Keyin BB dispetcheri so'rovni amalga oshirish uchun fayllar dispetcherini chaqiradi. BB dispetcherining tarkibi 21.2-rasmda keltirilgan.

*Fayllar dispetcheri* berilganlarni saqlashga mo'ljallangan fayllarni manipulatsiyalash va diskdagi bo'sh joylarni taqsimlashga javob beradi. U ichki chizmada aniqlangan tuzilishlar ro'yxati va indeksini yaratadi hamda qo'llaydi. Agarda fayllarni xeshlash ishlatilsa, fayllar dispetcheri yozuvlarining adreslarini generatsiya qilish uchun xeshlash funksiyasini chaqirish bilan ham shug'ullanadi. Lekin fayllar dispetcheri fizik jihatdan beril-

ganlarni kiritish-chiqarishni boshqarmaydi, faqtgina zaruriy so'rovlarni kerakli kirish usullariga yo'llaydi, ular berilganlarni o'qish yoki disk (yoki xesh)ga yozishni tizim buferi orqali hal qiladi.



21.1-rasm. Namunaviy BBT asosiy tarkibi.

*DML tilining preprosessori*, amaliy programmalarga joylangan DML operatorlarni, asosiy ildagi standart funksiyalarni chaqiruvchilariga o'zgartiradi. Mos kodni generatsiya qilish uchun DML tilining preprosessori so'rovlar protsessori bilan o'zaro aloqaga chiqadi.

*DDL tilining kompilyatori* DDL buyruqlarni meta berilganlardan iborat jadvallar majmuasiga olib keladi. Keyin bu jadvallar tizim katalogida saqlanadi, boshqaruvchi ma'lumotlar esa berilganlarning fayllaridagi sarlavhalarda saqlanadi.

*Lug'at dispatcheri* tizim katalogiga kirish va u bilan ishlash jarayonini boshqaradi. Tizim katalogiga kirish huquqi BBT deyarli hamma tarkiblarida bor.

Quyida BB dispatcherining tarkibiga kiruvchi asosiy programmalarning (21.2-rasmida) tashkil etuvchilari keltirilgan.

*Kirish huquqini nazorat moduli*. Ushbu modul foydalanuvchining bajarmoqchi bo'lgan amallari uchun bajarish huquqi bor yoki yo'qligini tekshiradi.

*Komanda (buyruqlar protsessori).* Foydalanuvchining kirishi huquqi tekshirilgandan keyin boshqaruv buyruqlar protsessoriga uzatiladi. U o‘z navbatida buyruqlarni ket-ket bajarishga o‘tadi.

*Butunlikni nazorat vositalari.* Butunlikni nazorat vositalari, agarda BBdag'i berilganlarni o‘zgartirish amali qo'llansa, shu amal butunlikni nazorat qilishda o‘rnatilgan chegaralarni qanoatlantirishini nazorat qiladi. Masalan, o‘rnatilgan kalitlarga nisbatan bo‘lgan talablarni.

*So‘rovlarini optimallashtiruvchi.* Ushbu modul so‘rovni bajarish strategiyasining optimal usulini tanlaydi.

*Tranzaksiyalar dispatcheri.* Bitta foydalanuvchi yoki amaliy programma tomonidan berilganlar bazasini o‘qish yoki o‘zgartirish bo‘yicha amalga oshiriladigan harakat yoki harakatlar ketma-ketligi *tranzaksiya* deb ataladi. Tranzaksiya berilganlar bazasida olib boriladigan mantiqiy ishdan iborat. U alohida programma, programmaning qism yoki alohida bitta buyruqdan (masalan, SQL tilidagi INSERT yoki UPDATE buyruqlari) iborat bo‘lishi va berilganlar bazasida bajariladigan ixtiyoriy sondagi amallarni o‘z ichiga olishi mumkin. Berilganlar bazasining administratori nuqtayi nazaridan ixtiyoriy ilovadan foydalanish bu bir qator tranzaksiyalarga olib keladi, ularning orasida esa berilganlarga ishlov beriladi. *Tranzaksiyalar dispatcheri*, tranzaksiyalarni bajarish jarayonida kelib tushadigan amallarga talab qilingan ishlov berishni amalga oshiradi.

Tranzaksiyalarning asosiy to‘rtta xossasi mayjud *ACID* (Atomicity, Consistency, Isolation, Durability – uzilmaslik, uyg‘unlik, yakkalik, tur-g‘unlik):

- *Uzilmaslik.* «Hammasi yoki hech nima» iborasiga mos bo‘lgan xossa. Ixtiyoriy tranzaksiya bo‘linmas ishlari majmuasidan iborat va bu majmua butunligicha bajariladi yoki butunlay bajarilmaydi. Uzilmaslikni ta‘minlashga javobgar *tiklash dispatcheri*.
- *Uyg‘unlik.* Har bir tranzaksiya berilganlar bazasini bir uyg‘unlik holatidan ikkinchisiga o‘tkazadi. Uyg‘unlikka javobgarlikni ta‘minlash *BBBT* va *ilovalarning yaratuvchilarga yuklanadi*. BBB tizimida uyg‘unlik berilganlar bazasidagi chizmada keltirilgan hamma, ya’ni butunlik va predmet sohadagi, cheklashlarni ta‘minlash orqali erishiladi. Ilovalardagi mantiqiy amallar natijasida yuzaga keladigan uyg‘unsizlik holatlariga programmachi javob beradi.
- *Yakkalik.* Hamma tranzaksiyalar bir-biriga bog‘liqsiz ravishda bajariladi. Boshqacha aytganda, tugallanmagan tranzaksiyaning oraliq natijalari boshqa tranzaksiyalarning ish faoliyatiga umuman ta’sir qilmasligi kerak. Yakkalikni ta‘minlashga javobgar parallel bajarish ishlarini boshqaruvchi – *rejalashtiruvchi* moduldirdir.

- *Turg'unlik.* Muvaffaqiyallli tugagan tranzaksiyalarning natijalari berilganlar bazalarida saqlanishi va kelajakdagi inkor vaziyatlarda yo'qotilmasligi kerak. Turg'unlikni ta'minlashga javobgarlik *tiklash dispatcheriga* yuklatilgan.

*Rejalahtiruvchi.* Ushbu modul BB bilan bajariladigan parallel amallarni maksimal normal holatda bajarilishini nazorat qiladi. Alovida bajariladigan tranzaksiyalardagi talab qilingan amallarni bajarish tartibini boshqaradi. Ayrim vaziyatlarda, masalan parallel bajarish protokoli blokirovkalar tizimi asosida qurilsa, rejalahtiruvchi modulni blokirovkalar dispetcheri deb atashadi.

*Tiklash dispatcheri.* Tizimda inkor yuz bergan holatlarda, BBni ishlaysidan normal holatga qaytarishni kafolatlovchi modul. Xususan, tranzaksiyalarni bajarish natijalarini fiksirlash va bekor qilishga javob beradi.

*Buferlar dispetcheri.* Bu modul berilganlarni tezkor xotiradan ikkinchi darajali xotiralarga ko'chirishga mas'ul. Masalan, vinchester yoki flesh-kaga.

Ba'zan tiklash va buferlar dispetcherlarini birgalikda *berilganlar dispetcheri*, buferlar dispetcherini esa *kesh dispetcheri* deb atashadi.

BB nafaqat tashkilotga tegishli berilganlarni, balki ularning tavsiflarini ham saqlay boshladi. Shuning uchun BB o'z-o'zini tavsiflovchi yozuvlardan iborat integrallashgan majmua, deb qarash ham mumkin. Berilganlar tavsifining majmuasini *tizim katalogi* (system catalog) yoki *berilganlar lug'ati* (data dictionary), elementlarning o'zini esa, *meta berilganlar* (meta-data), ya'ni «berilganlarning berilgani» deb atay boshlandi. BBda berilganlar o'z tavsifiga ega bo'lgani uchun ham *programmalarning berilgantardan mustaqilligi* (program-data independence) ta'minlanadi.

BBda, berilganlarning tavsiflarini ilovalardan alovida ajratish usulidagi yondashuv, zamonaviy programma ta'minotida obyektning ichki tavsifi bilan tashqi tavsifidan ham foydalanish imkoniyatlari mavjudligi bilan juda o'xshab ketadi. Obyektdan foydalanuvchilar faqat tashqi tavsifni ko'radilar va u qanday tavsiflanishi va faoliyat yuritishi bilan qiziqmaydilar. Ushbu *berilganlarni abstraksiyalash* (data abstraction) yondashuvidagi yutuqlardan biri shundan iboratki, obyektning tashqi tavsifini o'zgartirmaslik sharti bilan ichki tavsifini ixtiyoriy ravishda o'zgartirish mumkinligida hamda buning foydalanuvchilarga mutlaq ta'sir yetmaslidigadir. BBda ham berilganlarning tuzilishi ilovalardan alovida turadi va BBda saqlanadi. Masalan, yozuvga yangi maydonni qo'shish yoki yangi fayllarni yaratish ushbu BB bilan ishlovchi mavjud ilovalarga hech qanday ta'sir qilmaydi. Lekin, ilova ishlatajigan fayldan maydon o'chirilsa, u ilovaga ta'sir qiladi, shuning uchun ilovani o'zgartirishga to'g'ri keladi.

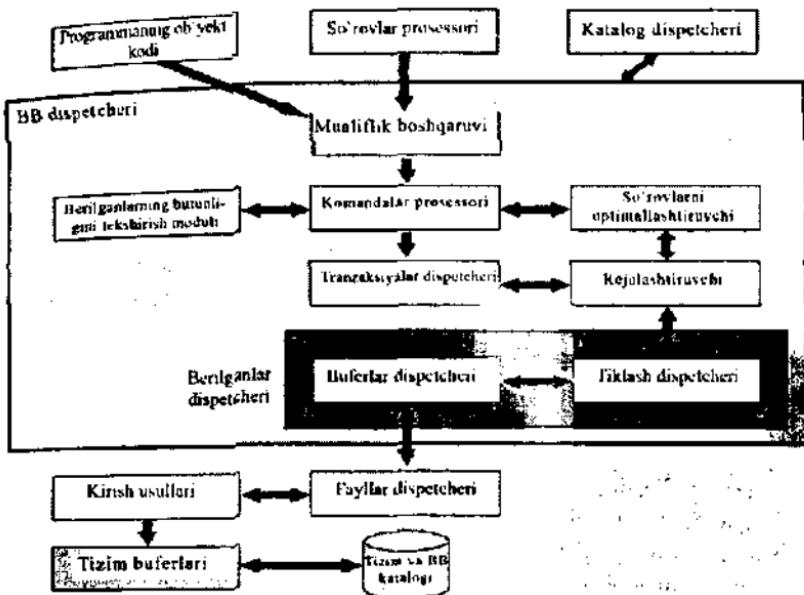
BB boshqarish muhitida asosiy beshta tarkibiy elementlarni keltirish mumkin: *apparat ta'minot*, *programma ta'minoti*, *berilganlar*, *protseduralar* va *foydalanuvchilar*.

### Apparat ta'minot

BBBT va ilovalarning ishlashi uchun *apparat ta'minot* tabiiy holda zarur. U juda katta oraliqni tashkil qilishi mumkin bittagina shaxsiy kompyuterdan yoki bir nechta o'nlab tarmoqqa ulangan kompyuterlar bo'lishi mumkin. Ishlatiladigan apparat ta'minot foydalanuvchi tashkilot talablari va BBBTning turiga bog'liq. Ayrim BBBT faqatgina aniq operatsion tizimlar va qurilmalar, boshqalari esa keng turdag'i apparat ta'minot va turli operatsion tizimlar ostida ishlashi mumkin. Odatda, BBBT minimal tarzdagi tezkor va tashqi xotira yetarli bo'ladi, lekin bu tizimning samaradorligini oshirishga kamlik qilishi mumkin. Masalan, minglab talabaga ega hamda o'ndan ortiq fakultetlari mavjud katta oliy o'quv yurtlarida markaziy kompyuter va unga ulangan har bir fakultetda bir nechta kompyuterlar bo'lishi mumkin. Markaziy kompyuterda BBga kirishni nazorat va xizmat qiluvchi *BBBT server qismi* (backend) joylashishi, qolgan kompyuterlarda esa *BBBTning mijoz qismi* (frontend) ishlashi mumkin. Bunday arxitektura *mijoz/server* deb nomlanadi. Server qismida server sifatida markaziy kompyuter, mijoz sifatida fakultetlardagi kompyuterlar qatnashadi.

### Programma ta'minoti

*Programma ta'minoti* – BBB tizimidagi programma mahsulotlari, amaliy programmalar, operatsion tizim hamda BBBT tarmoqda faoliyat yuritayotgan bo'lsa, u holda tarmoqning programma ta'minotidan ham iborat bo'ladi. Odatda ilovalar uchinchi avlod tillarida S, C++, Java, Visual Basic, Ada, Pascal yoki SQL tilining operatorlari kiritilgan uchinchi avlod tillarida yaratilmoqda. Bundan tashqari, tezda ilovalarni yaratish uchun BBBT o'zining joylangan protsedurasiz tillardan foydalanuvchi shaxsiy vositalariga ega bo'lishi mumkin. Masalan, hisobotlar generatori, shakllar, grafik tasvirlar va hattoki to'liq miqyosdagi ilovalar.



21.2-rasm. BB dispatcherining tarkibi.

### BBBBT tillari

BB bilan ishlovchi BBBTning ichki tili ikki qismdan iborat: *berilganlarni aniqlash tili DDL* va *berilganlarni manipulotsiyalash DML* tilidan iborat. DDL tili BB chizmasini aniqlashga, DML tili esa berilganlarni o'qish va yangilashga ishlataladi. Bu tillar berilganlarning til ostilari deb ataladi, chunki ularda programmalash tillariga o'xshash odadagi vositalar yo'q. Lekin ko'pgina BBB tizimlarida til osti elementlarini yuqori darajadagi tillarga joylash imkoniyati mavjud. Ana shunday tuzilmalar lug'atdan foydalanib, BBBT funksiyalarini chaqirish bilan almashitadi, keyin u kompilatsiya qilinib, obyekt kodga joylashtiriladi. Shundan so'ng programmaning kodи ishga tayyor bo'ladi.

DDL berilganlarni aniqlash tili BBning administratori yoki foydaluvchiga mohiyatlar va atributlarni tavsiflash hamda mohiyatlar orasidagi aloqalarni tavsiflash hamda butunlik va himoya chegarasini ko'rsatish imkonini yaratadi. BB chizmasi DDL tilida yozilgan aniqlashlar majmudasidan iborat bo'ladi. DDL yordamida yangi chizmani aniqlash yoki mavjud chizmani o'zgartirish mumkin. Bu tilni berilganlarni boshqarishga ishlatib bo'lmaydi. DDL operatorlari jadvallarning maxsus turlarini, ya'ni tizim kataloglarini yaratadi (21.1-rasm). Tizim kataloglatida meta berilganlar,

ya'ni BB obyektlarini tavsiflovchi berilganlar hamda ularga kirish va boshqarishni soddalashtirish imkoniyati yaratiladi. Meta berilganlar yozuvlarni, berilganlar elementlarini hamda boshqa obyektlarning BGBT uchun yoki foydalanuvchilar uchun zaruriy tavsiflaridan iborat bo'ladi. Aniq berilganlarga murojaat qilishdan oldin BGBT tizim katalogiga murojaat qiladi. Tizim katalogini belgilashda, berilganlar lug'ati va berilganlar katalogi atamalari ham ishlataladi. Nazariy jihatdan berilganlarning «Uch bosqichli» arxitekturasida tashqi chizma, konseptual chizma va ichki chizma uchun alohida tillarni ishlatalish mumkin. Amaliyotda esa, bitta umumiyl til – identifikatsiyaflash tili ishlataladi. Bu orqali kamida tashqi va konseptual chizmalar aniqlanadi.

DML tili berilganlarni boshqarish uchun ishlataladi. Berilganlarni boshqarish amallariga quyidagilar kiradi:

- BBga yangi berilganlarni kiritish;
- bazadagi berilganlarni o'zgartirish;
- bazadagi berilganlardan ma'lumotlar olish;
- bazadagi berilganlarni yo'qtish.

Shunday qilib, BGBT asosiy funksiyalaridan biri berilganlarni manipulyatsiyalash tilini qo'llab-quvvatlash va yuqorida sanab o'tilgan amallarni berilganlar bazalari ustida amalga oshirishdan iborat. Berilganlarni manipulyatsiyalash konseptual va ichki darajaga tegishli. Lekin bunda berilganlarga kirishning juda samarador usullaridan foydalaniib, ichki darajada murakkab protseduralarning quyi darajalarini aniqlash kerak. Bundan ham yuqoriroq darajalarda aksincha, foydalanishni soddalashtirish va foydalanuvchi bilan tizim orasidagi o'zaro aloqalarning samaradorligini ta'minlashga e'tibor qaratiladi.

DML tilining protsedurasiz qismi, ya'ni berilganlarni tanlash amallarini bajaruvchilar, so'rovlar tili deb ataladi. So'rovlar tilini tor doiradagi berilganlar bazasidan, berilganlarni tanlash imkoniyatini yaratuvchi, yuqori darajadagi til sifatida qarash mumkin. Shu ma'nuda «so'rov» atamasи berilganlarni so'rov tili orqali tanlash uchun ishlataladi. DML tillarida berilganlarni tanlash usullari turlicha. DML tilining ikkita turi mavjud: *protsedurali* va *protsedurasiz*. Ularning orasidagi asosiy farq shundan iboratki, protsedurali tillar natijaga erishish mumkinligini aniqlasa, protsedurasiz tillar esa natijani qanday qilib olish kerakligini ko'rsatadi. Odatda, protsedurali tillarda yozuvlarning har biri alohida ko'riladi, protsedurasiz tillar esa yozuvlarning bir qator to'plamlari bilan ishlaydi.

### Protsedurali DML tillari

Programmachi protsedurali DML tili qanday berilganlar kerakligini va ularni tanlash usulini ko'rsatadi. Bu unga berilganlarga kirishning hamma

usullarini aniqlash zarurligini bildiradi. Odatda, bunday protsedurali DML tili yozuvni tanlash, ishlov berish va olingan natijaga nisbatan keyingi yozuvni tashlash, ishlov berish va hokazodan iborat bo'ladi. Ushbu berilganlar ustidagi jarayon hamma so'ralgan yozuvlar ishlovdan o'tmaguncha, davom ettiriladi. Odatda protsedurali DML tilining operatorlari yuqori darajadagi programmalash tillarining ichiga joylanadi hamda programmalash tilining takrorlash va boshqa yozuvlarga o'tish imkoniyatlaridan foydalilaniladi. Iyerarxik va tarmoq turidagi BBB tizimlarining DML tillari odatda protsedurali bo'ladi.

### Protsedurasiz DML tillari

Protsedurasiz DML tillari berilganlarning butun to'plamini bitta tanlash operatori yordamida aniqlash yoki yangilash imkonini yaratadi. Protsedurasiz DML tillari orqali foydalanuvchi *nima* kerakligini bildiradi, xolos. Qolgan amallarning hammasini tilning o'zi bajaradi va berilganlar ustida manipulatsiya qilish imkoniyatlarini yaratadi. Bunday usul foydalanuvchidan berilganlarning tuzilishi va tanlash algoritmlarini o'zlashtirishni talab qilmaydi. Natijada foydalanuvchi berilganlarga nisbatan mustaqil ravishda ishlaydi. Protsedurasiz tillarni *rasmiy tillar* deb ham yuritiladi. Relatsion turdag'i BBB tizimi odatda berilganlarni manipulatsiyalash tillarini o'z ichiga olgan bo'ladi. Asosan bu *namuna bo'yicha so'rov tilidan QBE* (Query-by-Example) yoki *tuzilishli so'rovlar tili SQL* (Structured Query Language) iborat. Protsedurasiz DML tillarni programmalashdan yiroq bo'lganlar ham tushunishi va ishlatishi oson. Biz ushbu tillar xususida 22-30 boblarda batafsil to'xtab o'tamiz.

### 4GL tillari

4GL qisqartmasi *to'rtinchi avlod tili* (Fourth-Generation Language) iborasining ingliz tilidagi variantining yozilishidan tuzilgan. Agarda berilganlar ustida birorta amalni bajarish uchun 3GL tillarida, masalan COBOL, yuzlab satrdan iborat programmaning kodini yozish kerak bo'lsa, xuddi shu narsani amalga oshirish uchun 4GL tillarida 10-15 satr yetarli. Uchinchi avlod tillarning hammasi protsedur tillar bo'lsa, 4GL tillari protsedurasiz tillar bo'lib, bunda foydalanuvchi *nima* qilish kerakligini talab qiladi, lekin *qanday* amalga oshirish tushuntirilmaydi. To'rtinchi avlod tillarni amalga oshirish yuqori darajadagi tarkiblardan foydalanishga asoslanadi, bularni «to'rtinchi avlod vositalari» deb atashadi. Foydalanuvchi *nima* kerakligini kerakli ko'rsatmalar orqali yozadi, yuqorida aytilgan vositalar esa avtomatik tarzda ilovalarni generatsiya qiladi. To'rtinchi avlod

tillari shu tillar yechaladigan masalalarning yechish turlarini toraytirish evaziga ishlash unumtdorligini ancha yuqoriga ko'taradi. To'rtinchi avlod tillarining quyidagi turlariga alohida e'tibor qaratiladi:

- axborotni taqdim qilish tillari, masalan so'rovlar tili yoki hisobotlar generatori;
- maxsus tillar, masalan elektron jadvallar va BB tillari;
- ilovalarni yaratishda berilganlar bazasidan aniqlash, kiritish, yangilash yoki berilganlarni tanlashni ta'minlovchi ilovalar generatori;
- ilovalarning kodlarini generatsiya qila oladigan juda yuqori darajadagi tillar.

To'rtinchi avlod tillari misolida yuqorida aytib o'tgan SQL va QBE tillarini keltirish mumkin. Quyida to'rtinchi avlod tillarining ayrimlariga qisqacha izohlar berib o'tamiz.

### **Shakllar generatori**

*Shakllar generatori* muloqot vositasidan iborat bo'lib, kiritish namunalarini tezda yaratish va berilganlarni ekran shakllarida akslantirish imkoniyatini yaratadi. Shakllar generatori ekran ko'rinishining tashqi shaklini, unda saqlanuvchi berilganni aniqlash va ekrandagi o'mini aniqlashdan iborat. Uning yordamida ekran elementlarining rangini, shriftlarni turlarini o'zgartirish va h.k. amalga oshirish mumkin. Takomil-lashtirilgan shakllar generatorlari arifmetik operatorlar yoki holatini o'zgartiruvchi funksiyalar yordamida hisoblanuvchi atributlarni yaratishi, hamda kiritiladigan berilganlarni tekshirish qoidalarini berish imkoniyatiga ega.

### **Hisobotlar generatorlari**

*Hisobotlar generatorlari* BBdag'i berilganlar asosida hisobotlarni yaratish vositasidir. Uning so'rov tiliga o'xshashligi, foydalanuvchi uchun BBga so'rovni shakllantirish vositasining mavjudligi va undan hisobotdagi ma'lumotni yaratish imkoniyati borligidadir. Lekin hisobotlar generatorlari hisobotlarning tashqi ko'rinishini boshqarishga juda katta imkoniyatlar beradi. Hisobotlar generatori olinadigan natijalarning tashqi shaklini avtomatik hosil qilish yoki maxsus buyruqlar orqali hisobotning o'zingizga ma'qul naqlini yaratish imkoniyatini beradi.

Hisobotlar generatorlari asosan ikki xil bo'ladи: *berilganlarni grafik tasvirlash generatorlari* va *namoyishli generatorlar*. Birinchisida qism tillardan birida hisobot uchun zarur berilganlar va hujjatning tashqi ko'rinishi aniqlanadi. Ikkinci usul bilan esa, bu maqsadlar uchun namoyishli vosita ishlataladi.

## Berilganlarni grafik tasvirlash generatorlari

Ushbu generator berilganlarni BBdan tanlash va ularni grafik diagrammalar shaklida taqdim qilish uchun aloqalar va an'analardan foydalaniadi. Odatda ana shunday generatorlardan foydalangan holda gistogrammalar, aylana, ustun, nuqtali diagrammalar va h.k. tayyorlanadi.

## Ilovalar generatorlari

*Ilovalar generatorlari* BB bilan o'zaro harakatlanuvchi programmalarni yaratish vositalaridan iborat. Ilovalar generatorlaridan foydalangan holda, amaliy programma ta'minotning loyihalashga ketadigan vaqtini qisqartirish imkoniyati mavjud. Ilovalar generatorlari odatda, asosiy funksiyalardan iborat, ko'pchilik programmalar ishlataladigan, avvaldan yaratilgan modullardan tashkil topgan bo'ladi. Yuqori darajadagi tillarda tuzilgan bunday modullar funksiyalarining «kutubxonalarini» tashkil qiladi. Foydalanuvchi qaysi masalalarni yechishni ko'rsatadi, ilovalar generatori esa masalani qanday bajarishni hal qiladi.

## Tasavvur

Yuqorida keltirilgan BB boshqarish tizimlarining funksional imkoniyatlarning mavjudligi, uni juda ham foydali vositaga aylantiradi. Lekin foydalanuvchilarga tizimning ichki tuzilishi soddamni yoki murakkabnii, bu narsalar qiziqtirmaydi. Hattoki ayrim holatlarda, bizga kerakligidan ko'p berilganlar chiqmoqda, degan qarshiliklar ham paydo bo'ladi. Ana shunday muammolarni hal qilish uchun BBB tizimida *tasavvur* (view)larni yaratish mexanizmi mavjud bo'lib, ixtiyoriy foydalanuvchi o'ziga kerakli BB zaruriy ko'rinishini yaratadi, buni BB qism-to'plami deb qarash mumkin.

Foydalanuvchilarga faqatgina zaruriy berilganlar bilan ishlash imkoniyatidani tashqari, tasavvurning bir qator foydali tomonlari mavjud:

- qo'shimcha xavfsizlik darajasini ta'minlaydi. Tasavvur boshqa foydalanuvchilar ko'rishi kerak bo'limgan berilganlarni chiqarib tashlangan holda yaratilishi mumkin. Masalan, bo'lim boshliqlari va tegishli hisobotchilarga xodimlarning maoshlari xususidagi berilganlarni ham ko'rish imkoniyati bor bo'lgan tasavvurni yaratish. Qolgan foydalanuvchilar uchun esa maosh xususidagi berilganlar chiqarib tashlangan holdagi tasavvurni yaratish.

- BB tashqi aloqasini sozlash mexanizmining mavjudligi. Masalan, **FAMILIYA\_ISMI\_SHARIFI** o'miga qisqaroq, FISH nomi bilan ishtash.
- BB tuzilishiga o'zgartirishlar kiritilgan holatda ham, tashqi aloqani qarama-qarshiliklarsiz va o'zgarmas holatda saqlash. Masalan, yangi maydonlar qo'shilganda yoki olib tashlanganda, aloqalar o'zgar-ganda, fayllar ajratilganda, qayta tashkillashtirilganda yoki nomi o'zgartirilganda ham BB tashqi aloqalari o'zgarmaydi. Agarda qaysidir tasavvurning fayliga ishlatalmaydigan maydonlar qo'shilsa yoki olib tashlansa, ushbu tasavvurga umuman ta'sir qilmaydi. Shunday qilib, tasavvurlar programmalarining BB haqiqiy tuzilishiga risbatan to'liq mustaqilligini ta'minlaydi va bu bilan fayllar tizimi-dagi eng katta kamchilikni bartaraf etadi.

Haqiqatda, BBBT funksional imkoniyatlari aniq boshqarish turiga bog'liq. Masalan, shaxsiy kompyuterlar parallel birgalikda kirish imkoniyatlarini bermaydi, berilganlarning butunligini saqlagan holda himoya uslubini boshqarish va berilganlarni tiklash juda tor miqyosda amalga oshirildi. Lekin zamonaviy ko'p mijozlar ishlatajigan BBBT yuqorida keltirilgan hamma funksional va boshqa imkoniyatlarni beradi. Zamonaviy boshqarish tizimlari o'ta murakkab programma ta'minoti bo'lib, programma kodlarining millionlab satrlaridan va ko'pgina jildli hujjalardan iborat.

Bundan tashqari BBBT apparat va programma inqirozlarida deyarli yuz foiz ishonchilik va tayyorgarlikni ta'minlaydi. BBBT programma ta'minoti doimiy ravishda takomillashib bormoqda. Masalan, ba'zi bir ilovalarda grafika, video, tovush va h.k. saqlash talab qilinmoqda. Ana shu masalalarni hal qilish uchun BBBT doimiy ravishda rivojanishi va shuning uchun uning funksional qismi ham o'zgarib turishi mumkin.

### Berilganlar

Albatta BBBT muhitida, berilganlar tarkibdagi eng asosiy a'zo bo'lib hisoblanadi. BB berilganlardan tashqari, meta berilganlarni, ya'ni «berilgan xususidagi berilganlarni» ham saqlaydi. BB tuzilishini chizma (schema) deb atashadi.

*Talabalar* o'quv loyihasining eslaylik. Ana shu *Talabalar* BB beshta jadvallardan tuzilgan edi: **Ro'yhat, Baholar, Fakultetlar, Yo'naliш va Fanlar**. Talabalar BB chizmasi quyidagicha:

- **Ro'yhat (ZN, **FAMILIYA**, **ISM**, **O\_ISM**, **K\_SANA**, **FAK\_N**, **YON\_N**, **KURS**, **GURUH\_N**, **PASP\_S**, **PASP\_N**)**
- **Baholar (ZN, **SEMESTR**, **FAN\_N**, **BALL**, **SANA\_KIR**, **USTOZ**)**

- Fakultetlar (FAK\_N, FAKULTET)
- Yo'nalish (YON\_N, YO'NALISHLAR)
- Fanlar (FAN\_N, FAN\_NOMI)

### **Protseduralar**

*Protseduralar* yo'riqnomalar va qoidalardan iborat bo'lib, ular BB loyihalash va foydalanish jarayonlarida hisobga olinadi. BBni foydalanuvchilari va unga xizmat qiluvchilarga shu tizimdan foydalanish protseduralarining to'liq tavsifi keltirilgan hujatlarni hamda quyida keltirilgan harakatlarni bajarish qoidalardan iborat yo'riqnomalarni o'z ichiga oladi:

- BBBT ro'yxatdan o'tkazish;
- BBBT alohida vositasi yoki ilovasidan foydalanish;
- BBBT ishga tushirish yoki to'xtatish;
- BBBT zaxira musxalarini yaratish;
- apparat va programma fa'minotida inkor bo'lganda, ularga ishlov berish, ishlamay qolgan tarkibiy qism yoki programmani aniqlash va to'g'rilash hamda nosozliklar bartaraf etilgandan so'ng, berilganlar bazasini tiklash;
- jadval tuzilishini o'zgartirish, bir nechta disklarda joylashgan BBni qayta tashkil etish, tizim unumдорligini yaxshilash va ikkinchi darajali qurilmalarda saqlanadigan berilganlarni arxivlash usullari.

### **Foydalanuvchilar**

Foydalanuvchilarni to'rtta guruhlarga ajratsak to'g'ri bo'ladi: *BB* va *berilganlarning administratorlari*, *berilganlar bazasini yaratuvchilar*, *amaliy programmachilar* va «oxirida» *foydalanuvchilar*.

#### **BB va berilganlarning administratorlari**

BB va BBBT umumiy zaxira hisoblanadi, bosliqa zaxiralar qanday boshqarilsa, u ham shunday boshqarilishi kerak. Odatta berilganlarni va BBni boshqarish deganda, BBBTni hamda undagi berilganlarni boshqarish va nazorat qilish tushuniladi. *Berilganlar administratori* berilganlarni boshqarish; BBni rejalash; standartlarni, amaliy algoritmlarni va ish protseduralarini yaratish va kuzatish hamda BB konseptual va mantiqiy loyihasiga javob beradi. *Berilganlar administratori* BBning umumiy rivojlanish tamoyillarining tashkilot maqsadlariga mosligini nazorat qilgan holda, yuqori bo'g'indagi rahbarlarga maslahat va tавsiyalar beradi.

*Berilganlar bazasining administratori* (Database Administrator) berilganlar bazasining fizik jihatni amalga oshishiga javob beradi. Bunga

fizik loyihalash va loyihani amalga oshirish, xavfsizlikni va berilganlarning butunligini ta'minlash, operatsion tizimni kuzatish hamda ilovalar va oxirida foydalanuvchilarga maksimal tarzdag'i unumdonlikni ta'minlash masalalari kiradi. **Berilganlar administratoriga** nisbatan **berilganlar bazalarining administratorining** majmuriyatları ko'proq texnik sifatga ega. Unga BBBT va uning aniq vazifalarini va tizim atrofini bilish kerak emas. Ba'zi tashkilotlarda buning farqiga borilmaydi, ba'zilarida esa har birining vazifalari aniq va ravshan ajratiladi.

### **Berilganlar bazasini yaratuvchilar**

Katta hajmdagi BBni loyihalashda ikkita turga bo'lingan guruh ishtiroy etadi: mantiqiy BBni yaratuvchilar va fizik BBni yaratuvchilar.

Mantiqiy BB yaratuvchilar berilganlarni identifikatsiyalash (ya'ni mohiyat va ularning atributlarini), berilganlar orasidagi aloqalarni va saqlanadigan berilganlarga qo'yiladigan chegaralarni o'rnatadi. Mantiqiy BB yaratuvchilar, avvalami bor obyektning xossalari va uning tuzilishini, ana shundan keyin berilganlar tuzilishi va uning ish xususiyatlarini har tomonlama va chuqur tushinishi zarur. Ish xususiyatlari tizimga nisbatan talablarni shakllantirish imkoniyatini beradi. Masalan, Talabalar o'quv loyihasidagi ish xususiyatlariga misol keltiramiz:

- har bir talaba bir vaqtida faqat bitta oliy o'quv yurtida o'qishi mumkin;
- talabada faqat bitta reyting daftarchasi bo'lishi mumkin;
- talaba oliy o'quv yurtidagi kutubxonalardan foydalanishi mumkin;
- guruhlarda talabalar soni 25dan oshmasligi kerak;
- har bir talaba rejadagi fanlarni o'zlashtirishga majbur;
- ijobjiy reyting 54 balldan yuqori bo'lishi kerak;
- professor-o'qituvchilar bir vaqtida bir nechta fandan dars berishi mumkin.

Samarali ishlash uchun berilganlar bazasining mantiqiy darajasini yaratuvchi, bo'lajak tizimdan foydalanuvchilarni berilganlar bazasini yaratish jarayoniga iloji boricha vaqtliroq jalb qilishi kerak. Biz berilganlar bazasining mantiqiy darajasini yaratuvchilarni ikki bosqichga ajratgan holda ko'ramiz:

- *konseptual BB loyihalash* aniq maqsadli BBB tizimiga, ilovalarga, programmalash tillariga yoki boshqa ixtiyoriy fizik tafsiflarga umuman bog'lanmagan;
- tanlangan berilgan modelini – relatsion, tarmoq yoki iyerarxik turlarini hisobga olgan holda, *berilganlar bazasining mantiqiy darajasini loyihulash*.

*Fizik berilganlar bazasini yaratuvchi* tayyor mantiqiy berilganlar modelini olib, uni fizik amalga oshirish bilan shug'ullanadi, jumladan:

- berilganlarning mantiqiy modelini jadvallar to'plamiga akslantirish va berilganlarning butunligini chegaralash;
- BB bilan ishlash jarayonida yuqori darajadagi samaradorlikni ta'minlash uchun berilganlarni saqlashning aniq tuzilishlarini va ularga kirish usullarini tanlash;
- berilganlarni himoyasi uchun ixtiyoriy talablarni loyihalash.

Berilganlar bazasini fizik loyihalashning ko'pgina bosqichlari tarlangan BBB tizimiga bog'liq, shuning uchun ham chizmalarini amalga oshirishning turli usullari mavjud. Demak, berilganlar bazasini fizik ravishda amalga oshiruvchi mutaxassis maqsadli BBB tizimining funksional imkoniyatlarini yaxshi farqiga borishi, har bir amalga oshirish usullarini chuqur tushunishi va berilganlarni saqlashning eng yaxshi strategik yo'llini tarlay olishi kerak. Agarda berilganlar bazasini konseptual va mantiqiy loyihalash «*nima?*» degan savolga javob bersa, fizik loyihalash «*qanday?*» degan savolga javob beradi.

### **Amaliy programmachilar**

BB yaratilganidan so'ng, foydalanuvchilarga zaruriy funksional imkoniyatlarni beruvchi ilovalarni yaratish kerak. Amaliy programmachilar ana shu ish bilan shug'ullanadilar. Odatda, ular tizim analitikkleri yaratgan identifikatorlar asosida ishlaydilar. Har bir programma BBB tizimidagi BB bilan aniq ishlarni bajarish uchun kerak bo'lgan operatorlardan iborat bo'la-di, masafan berisganfani tanlashi, kiritish, o'zgartirish yoki olib tashlash amallarini bajarishi mumkin. Yuqorida ta'kidlab o'tganimizdek, bu ishlarni uchinchi yoki to'rtinchchi avlod tillari yordamida bajarish mumkin.

### **«Oxirida» foydalanuvchilar**

BB mijozlari bu bazadan eng oxirgi bosqichda va doimiy ravishda foydalanuvchi bu «*oxirida*» foydalanuvchilaridir. Ular ma'lumotlarga bo'lgan talablarni qondirish uchun berilganlar bazalari va ularni boshqarish tizimlaridan foydalanadilar. «*Oxirida*» foydalanuvchilarni, oddiyroq qilib foydalanuvchilar deb ataymiz. Foydalanuvchilarni tizimdan foydalanish usullariga nisbatan sinflarga ajratish mumkin:

*Oddiy foydalanuvchilar.* Bular hattoki BBBT borligini ham bilishmaydi. Ular BBga bajarmoqchi bo'lgan amallarni maksimal tarzda soddalashtirilgan maxsus yaratilgan ilovalar orqali murojaat etadir. Bunday foydalanuvchilar oddiy buyruqlarni kiritib yoki menyu buyruqlarini

tanlab zaruriy amallarni bajarilishiga tashabbuskorlik qiladilar. Bu degani bunday foydalanuvchilarga BBBT yoki BB xususidagi hech qanday ma'lumot qiziqtirmaydi. Masalan, supermarket kassiri mahsulotning narxini bilish va xaridor bilan hisoblashish uchun skanerdan foydalanib, undagi shtrix kodni o'qiydi. Ana shu harakat natijasida maxsus programma nafaqat shtrix kodni o'qiydi, balki shu kod asosida mahsulotning narxini berilganlar bazasidan topadi, ombordagi qolgan shu turdag'i mahsulotlarni qayd etish maydonida uning qiymatini kamaytiradi, keyin kassa apparatida narxini akslantridi.

*Tajribali foydalanuvchilar.* Bular BBBT imkoniyatlari va BB tuzilishi bilan tanish bo'lgan foydalanuvchilar. Zarur amallarni bajarish uchun ular yuqori darajadagi so'rov tillaridan foydalanshtari mumkin, masalan SQLdan. Boshqa tajribali foydalanuvchilar o'zlarining maxsus amaliy programmalarini ham yozishlari mumkin.

### **BBB tizimlarining ustunlik va kamchiliklari**

BBB tizimlarining ko'p qirrali istiqbolfi tomonlari bilan birga **kamchiliklari** ham bor. Quyida shu masalaning ikki tomonini ko'rib chiqamiz.

#### **BBB tizimlarining ustunliklari**

##### **BBB tizimlarining ustunlik tomonlari:**

- berilganlar ortiqchaligining nazorat;
- berilganlar ziddiyati;
- berilganlarni saqlash hajmi o'zgarmagan holda foydali ma'lumot-larning ko'pligi;
- berilganlarni birgalikda ishlatish;
- berilganlar butunligini saqlash;
- yuqori darajadagi xavfsizlik;
- standartlarni qo'llash;
- tizim miq'yosining o'sishi – samaradorlikni oshishi;
- ziddiyatli talablar yuzaga kelganda kelishish imkoniyatlari;
- berilganlarga kirish imkoniyatlarining o'sishi va ularning ishga tayyorligi;
- mehnat unumдорлиги ko'rsatkichlarining oshishi;
- berilganlar mustaqilligi sababli tizimni kuzatishdagi soddalashuv;
- parallel ishlashni boshqarishdagi yaxshilanish;

- zaxiraviy nusxalash va tiklash xizmatlarining rivojlanishi.

### **Berilganlar ortiqchaligining nazorati**

1- bobda aytiganidek, fayl asosidagi tizimlar tashqi xotiradan samasiz foydalanar edi. BBdan foydalanishda fayllarning integratsiyasi hisobiga bir xildagi berilganlarni bir nechta nusxalarni saqlamaslik imkoniyati yaratildi, doimiy ravishda ortiqchalikni yo'qotishga harakat qilinadi. Lekin to'liq ortiqchalikka erishilmaydi, faqat ortiqchalik darajasi kamayadi.

### **Berilganlar ziddiyati**

Berilgalarning ortiqchaligini yo'qotish yoki uning ustidan nazorat o'matish qarama-qarshi hotatlar yuz berish xavfini kamaytiradi. Agarda berilganlar elementi bazada bir nusxada saqlangan bo'lsa, uni o'zgartirishga bittagina amal kerak, o'zgargan element berilganlar bazasidan foydalanuvchilarga yetib boradi. Aksincha, agarda tizimning ruxsati bilan BBda bir nechta nusxada saqlansa, u holda tizim ushbu nusxalar orasida ziddiyatlar bo'lmasligini kuzatib turishi kerak. Lekin afsuski, zamонавиј BBB tizimlarining ko'pchiligidagi ziddiyatlar chiqmasligi avtomatik ravishida ta'minlanmagan.

### **Berilganlarni saqlash hajmi o'zgarmagan holda foydali ma'lumotlarning ko'pligi**

Tashkilotdagi ishchi berilganlarni integratsiya qilish natijasida xuddi shu berilganlardan yangi ma'lumotlarni olish mumkin. Masalan, yuqorida keltirilgan **Talabalar** o'quv loyihamizda professor-o'qituvchilarning qaysi fanlardan dars berishi xususidagi ma'lumotlar to'liq emas. Mavjud fayllarni integratsiya qilsak, maqsadga oson erishamiz.

### **Berilganlarni birqalikda ishlatish**

Fayllar alohida odamlarga yoki bo'limlarga tegishli bo'lishi mumkin. BB butun tashkilotga tegishli bo'lib, hamma ro'yxatdan o'tgan foydalanuvchilar birqalik ishlatishi mumkin. Ana shunday ravishda ishni tashkil qilishda, ko'pgina foydalanuvchilar ko'proq berilganlar bilan ishlash imkoniyatiga ega bo'ladi. Bundan tashqari doimiy ravishda BBni tuzilishini buzmasdan yangi berilganlarni qo'shish, yangi ilovalarni yaratish imkoniyati bor. Yangi ilovalarda – BBBT beradigan hamma imkoniyatlar, ya'ni berilgalarning tuzilishi va berilganlarga kirish usulini aniqlash; parallel

ishlovnii tashkil etish, nusxalash va tiklash imkoniyatlaridan shu funksiya-larni yaratmasdan ishlay olishi.

### **Berilganlar butunligini saqlash**

BB butunligi deganda, berilganlarning ziddiyatsizlik bilan saqlanishi tushuniladi. Butunlik odatda chegaralar orqali, ya'ni qarama-qarshiliklarga olib kelmaslik qoidalari buzilmasligi orqali aniqlanadi. Chegaralarni berilganlarning elementlariga bitta yozuv ichida yoki yozuvlar orasidagi aloqalarga qo'llash mumkin. Masalan, **Talabalar** o'quv loyihasida, «reyting balining maksimal darajasi 100dan oshmasligi kerak» yoki «kunduzgi bo'lindagi talabalarning yoshi 35dan oshmasligi kerak», deganimiz butunlik chegarasini bildiradi.

### **Yuqori darajadagi xavfsizlik**

Berilganlar bazasining xavfsizligi foydalanuvchilarga ruxsatsiz kirish huquqini bermaslikdan iborat. Xavfsizlik bo'yicha yetarli darajada choralar ko'rilmasa, fayllar tizimiga nisbatan integrallashgan berilganlarga kirish zaifroq. Lekin berilganlarning integratsiyasi BBning administratoriga BB tizimini talab doirasidagi xavfsizlikka olib kelishi, BBBT esa uni amalga oshirishi mumkin.

Himoya choralarini qo'llashda berilganlarga kirishning tabaqalashgan usulini qo'llash ham yaxshi chora beradi. Masalan, xodimlar o'z lavozimi va mavjesi bo'yicha tabaqalarga bo'linib, har biri o'zi bevosita bajaradigan ishlarga taalluqli fayllarga kirish huquqini berish.

### **Standartlarni qo'llash**

Berilganlarning integratsiyasi BBning administratoriga zaruriy standartlarni aniqlash va qo'llash imkoniyatini beradi. Masalan, bo'lim va tashkilot, davlat va xalqaro standartlar tizimlari orasida berilganlar yoki berilganlarni almashishda, berilganlarning formatlari, nomlari, shakllari, yangi-lash va kirish qoidalari kelishiladi.

### **Tizim miqyosining o'sishi – samaradorlikni oshishi**

Tashkilotdagi hamma berilganlarni bitta berilganlar bazasiga to'plab va ilovalar majmuasini yaratib, ancha mablag'larni iqtisod qiliш mumkin. Masalan, bir bo'limga o'zining ishlarini avtomatizatsiyalashtirish uchun ajratilgan mablag'larni boshqa bo'lindagi mablag'lar bilan birlashtirilsa,

ishlab chiqarish miqyosi oshadi va samaradorlik o'sadi. Birlashgan mablag'lar hisobiga tashkilot uchun zarur bo'lgan vositalarni sotib olish imkoniyati tug'iladi.

### Ziddiyatli talablar yuzaga kelganda kelishish imkoniyatlari

Ayrim foydalanuvchilar yoki bo'limlarning talablari boshqa bo'lim talablariga zid kelib qolishi mumkin. BB o'zining administratori tomonidan nazorat ostida bo'lgani uchun, u tashkilotdagi berilganlardan samarali foydalantish uchun loyihami o'zgartirishi va optimal yechimlarni izlashi mumkin.

### Berilganlarga kirish imkoniyatlarining o'sishi va ularning ishga tayyorligi

Integratsiya natijasida berilganlar hamma foydalanuvchilar uchun ochiq bo'ladi. Bu esa foydalanuvchilarning ish unumidorligini va tizimning bajaradigan vazifalarining samaradorligini oshiradi. Ko'pgina BBB tizimlarida hisobotlarni yaratish vositalari yoki so'rovlar tili mavjud bo'lib, foydalanuvchilarga avvaldan nazarda tutilmagan so'rovlarni kiritish va natijalarni olish imkoniyati berilgan. Masalan, **Talabalar** o'quv loyihamizdan a'lochi talabalar ro'yxatini olmoqchimiz, deylik. SQL tilida quyidagi so'rovnini kiritamiz va natijaga erishamiz:

```
SELECT [Ro'yhat].FAMILIYA, [Ro'yhat].ISM, [Ro'yhat].O_ISM,  
[Ro'yhat].FAK_N, [Ro'yhat].YON_N, [Ro'yhat].KURS,  
[Ro'yhat].GURUH_N, Baholar.BALL
```

```
FROM [Ro'yhat] INNER JOIN Baholar ON [Ro'yhat].ZN = Ba-  
holar.ZN
```

```
WHERE (((Baholar.BALL)>85))
```

```
ORDER BY [Ro'yhat].FAMILIYA;
```

### Mehnat unumidorligi ko'rsatkichlarining oshishi

BBB tizimlarida ancha standart funksiyalar kiritilgan bo'lib, programmachi bu funksiyalardan o'zi yaratadigan ilovalarda foydalanshi mumkin. BBBT asos darajasida fayllar bilan ishlovchi quyi darajadagi protseduralar bilan ta'minlaydi, bulardan odatda ilovalarda foydaliladi. Ana shu protseduralarning mavjudligi programminachini mayda ishlardan xalos qilib, ko'proq maxsus vazifalarga e'tiborini qaratishga undaydi. Ko'pchilik BBBTda to'rtinchi avlod berilganlar bazalarini yaratish vositalari ham mavjud bo'lib, ular ilovalarni yaratishni yanada osonlashtiradi.

Natijada programmachilarning mehnat unumdarligining ko'rsatkichlari oshadi.

### **Berilganlar mustaqilligi sababli tizimni kuzatishdagi soddalashuv**

Fayl tizimlarida berilganlarning tavsifi va berilganlarga mantiqiy kishish, har ilovaga joylangan, shuning uchun ham bu programmalar berilganlarga bog'liq bo'lib qoladi. Berilganlarning tuzilishidagi ozgina o'zgarish ham, hamma programmalariga ta'sir qiladi, ularni o'zgartirishga olib keladi. BBBT boshqacha yondashuv, berilganlarning tavsifi ilovalardan ajratilgan, shuning uchun ham ilovalar berilganlarning o'zgarishidan himoya qilingan. Bu xususiyatni biz yuqorida *berilganlardan mustaqillik* deb atadik. BB bilan ishlovchi programmalarning berilganlardan mustaqilligi ilovalarga xizmat ko'rsatish va ularni kuzatishni ancha sodda lashtiradi.

### **Parallel ishlashni boshqarishdagi yaxshilanish**

Ayrim fayl tizimlarida bir paytda ikkita foydalanuvchi bitta faylga kirib qolsa, ikkita so'rovlar orasida nizo chiqishi va natijada ma'lumot yo'qolishi yoki uning butunligi yo'qolishi mumkin. BBBT esa BBga parallel kirish imkoniyati yaratilgan va bunday nizolar chiqmasligi kafolatlangan.

### **Zaxiraviy nusxalash va tiklash xizmatlarining rivojlanishi**

Fayl tizimlarida apparat yoki programma ta'minotda yuz beradigan inkorlardan berilganlarning himoyasini ta'minlash foydalanuvchining zimmasidadir. Shunday bo'lishi mumkinki, har kuni kechasi zaxira nusxani olish darkor. Bunda agarda inkor yuz bersa, zaxira nusxa ishlataladi, lekin inkor vaziyat yuz berguncha analga oshirilgan ishlarning hammasini qaytadan bajarish kerak bo'ladi. Zamonaviy BBBT har qanaqangi inqiroz vaziyatlar yuzaga kelganda ham, ma'lumotlarni yo'qotish ehtimolligi juda oz.

#### **BBBT kamchiliklari**

##### **BBBtizimlarining kamchiliklari quyidagilardan iborat:**

- murakkablik;
- hajm;
- BBBT narxi;

- apparat ta'minotga qo'shimcha xarajatlar;
- o'zgartirish xarajatlari;
- unumdoorlik;
- tizim ishlamay qolgandagi jiddiy natijalar.

## Murakkablik

BBBT har bir yaxshi funksionallikni ta'minlashi uchun, programma ta'minotini ancha murakkablashtirish kerak bo'ladi. BBT hamma ustunliklaridan foydalanishi uchun, loyihami va BB yaratuvchilar, BB administratorlari hamda «oxirida» foydalanuvchilar BBTning funksional imkoniyatlarini yaxshi bilishlari kerak. Tizimning ishlash tamoyillarini tushunmaslik, noto'g'ri loyihalashga olib keladi, buning natijasida buyurtmachi juda tang vaziyatga tushib qolishi mumkin.

## Hajm

BBBT murakkabligi va funksional imkoniyatlarning borgan sari yanada kengayib borishi, bu tizimni o'ta murakkab programma mahsulotiga aylantirmoqda. Bu esa, samarali ishlash uchun, tezkor xotira hajmini o'sishiga olib keladi.

## BBBT narxi

Mavjud bo'lgan hisoblash muhiti va funksional imkoniyatlardan kelib chiqqan holda, BBT narxi juda katta oraliqda o'zgarishi mumkin. Masa'an, shaxsiy kompyuterlar uchun bunday tizimlarning narxi 100 AQSH dollari, lekin yuzlab foydalanuvchilarga xizmat qiladigan meynfreymdag'i BBT yuz minglab AQSH dollaridan iborat bo'lishi mumkin.

## Apparat ta'minotga qo'shimcha xarajatlar

Talab doirasidagi unumdoorlikka erishish uchun ancha quvvatli va faqat BBT ishlaydigan kompyuterlar talab qilinishi mumkin. Bu albatta qo'shimcha xarajatlarga olib keladi.

## O'zgartirish xarajatlari

Ayrim holatlarda BBT va qo'shimcha apparat ta'minotning narxi, yangi BBTga o'tish va ishlash uchun mavjud ilovalarni yangi tizimga moslashtirish xarajatlariiga nisbatan ancha qimmat turadi. Bu xarajatlarga

xodimlarni yangi tizimda ishlashga o'rgatish va yangi tizimni o'zgartirish va ishlatish uchun mutaxassislar xizmati haqi ham kiradi. Bularning hammasi, ba'zi tashkilotlarning BB boshqarishdagi yangi texnologiyalarga o'tishiga to'sqinlik qilmoqda.

## **Unumdorlik**

Odatda fayl tizimi ba'zi bir maxsus ilovalar uchun yaratiladi, masalan banklarga pul o'tkazish shuning uchun ularning unumdorligi katta bo'ladi. Lekin, BBBT umumiylashtirilishi tizimning zaifligini oshiradi. Hamma foydalanuvchilar va ilovalarning ishlashi BBBTga bog'liq bo'lgani uchun, uning tarkibida birorta inqiroz yuz bersa, butun tashkilot yoki bir nechta tashkilotlarning ishi to'xtab qolishi mumkin.

## **Tizim ishlamay qolgandagi jiddiy natijalar**

Vositalarning markaziylashtirilishi tizimning zaifligini oshiradi. Hamma foydalanuvchilar va ilovalarning ishlashi BBBTga bog'liq bo'lgani uchun, uning tarkibida birorta inqiroz yuz bersa, butun tashkilot yoki bir nechta tashkilotlarning ishi to'xtab qolishi mumkin.

Xulosa sifatida shunday bir shartli taqqoslashni keltiramiz. Agarda BBB tizimlarini shartli ravishda jonli mavjudotlar bilan taqqoslasak, BB zamonaviy axborotlar tizimlarida «yurak» vazifasini o'tamoqda. Lekin, jonli mavjudotdagisi yurak bilan axborot tizimlaridagi «yurak»ning eng katta farqi va ustunligi, oxirgisiining mustaqil ravishda ham faoliyat yurita olishidan iborat.

## **BBB tizimlarining tavsisi va sinflari**

1985- yillardan boshlab, ya'ni sifati kompyuterlarning keng tarqala boshlagan vaqtidan BBB tizimlarining «stolbop» naqlari – dBase, FoxBase (keyingi naqlari FoxPro va Visual FoxPro), Paradox, Access keng tarqala boshladi. Shularga o'xshash BBB tizimlaridagi jadvalarning eng tarqalgan formati .dbf bo'ldi. Bu format bilan dBase, FoxBase hamda BB bilan ishlash uchun Clipper (programmalarni qadamli kompilyator usulida yozishga mo'ljallangan) tizimlari keng foydalana boshladi. Keyinchalik bularning ayrimlari nafaqat «fayl-server» arxitekturasidagi turli operatsion tizimlar boshqaruvida tarmoqda faoliyat yurituvchi to'la qonli BBB tizimlariga aylandi, balki «mijoz-server» arxitekturasidagi BB serverlari bilan ishlash

imkoniyatini, bundan tashqari esa BB bilan ishlash uchun HTML sahifalarini yaratish va foydalananish imkoniyatlarini ham yaratdi.

Shaxsiy kompyuterlar uchun yaratilgan hamma BBB tizimlarini 3 sinfga ajratish mumkin:

- haqiqiy ma'nodagi BGBT ifodasiga to'liq mos keluvchi tizimlar. Bunday tizimlarda bazalar bilan ishlash imkoniyati faqatgina tizim kompyuterga to'liq yuklangandan so'ng yaratiladi, BB bilan ishlovchi avtonom ravishdagi programmalarni yaratish imkoniyati yo'q. Bunday tizimlarga Access, Paradox va dBBase kiradi;
- BB bilan to'g'ridan-to'g'ri ishlash imkoniyati va BB bilan operatsion tizim ostida ishlovchi foydalanuvchi programma (ilova)larini yaratish imkoniyatini ham beruvchi FoxPro tizimi;
- BB bilan operatsion tizim ostida ishlovchi foydalanuvchi programma (ilova)larini yaratish imkoniyatini ham beruvchi Clipper va Clarion tizimlari.

Shularga o'xshash BBB tizimlarining hammasi o'z tarkibida quyidagi vositalarga ega:

- berilganlar bazasini yaratuvchi va ularning tuzilishini o'zgartiruvchi; indeks fayllarni yaratuvchi;
- bazalar bilan ishlashning jadval formati yoki maydonlari satrlar bo'yicha joylashgan standart shakllarini yaratuvchi; bunda berilganlarni tahrirlash, yozuvlarni qo'shish va olib tashlash, bazadagi bir nechta jadvallar bilan ishlash, berilgan shartlar bo'yicha murakkab ifodalarni hisoblash va boshqa imkoniyatlari mavjud;
- BB bilan bog'langan maydonlarni yoki xotiradagi o'zgaruvchilarini tahrirlash imkoniyatiga ega bo'lgan ekran shakllarini ishlab chiqishdan tashqari, tugmalar shaklidagi turli boshqarish elementlari; ochiluvchi ro'yxatlar turidagi murakkab obyektlar va boshqalar;
- chop etish shakllarini generatsiyalash – qiymatlarni guruhi va umumiyligi qiymatlarni hisoblash (yig'indi, soni, o'rtachasi, maksimal, minimal va h.k.) amallarida berilganlar guruhiidan foydalinish kabi murakkab tuzilishi hisobot yaratuvchi;
- berilganlarga murakkab ishlov beruvchi programma modullarini yaratuvchi;
- juda murakkab tarzdagi so'rovlarni generatsiyalash – berilganlarni tanlashdagi murakkab shartlar bo'yicha, berilganlarni saralash va guruhash kabi turli bazalardagi berilganlardan foydalangan holdagi;
- foydalanuvchi uchun mo'ljallangan tizimlarda esa qo'shimcha ravishda, yuqorida keltirilgan hamma tarkiblami o'z ichiga oluvchi

menyu, tizim va loyiha ma'lumotnomalarini yaratuvchi hamda kompilatsiyalanuvchi bajariluvchi programma aylantiruvchi.

BBB tizimlarini tanlashdagi eng muhim omillar quyidagilardan iborat:

- operatsion tizimdagi boshqa ilovalar bilan ma'lumotlarni almashtish imkoniyatini ta'minlovchi BB formatlarining mavjudligi. Eng keng tarqalgan formatlardan biri dbf – format bo'lib, bu format bilan dBBase, FoxBase, FoxPro, Visual FoxPro, Clipper tizimlari ishlaydi. Bu formatni MS Office dagi hamma ilovalar «tushunadi». Bu bazalardagi berilganlarni Word, Yexcel va Access ga ko'chirsada bo'ladi. Clarion, Paradox va Access tizimlari esa bazalarning o'z maxsus formatlariga ega;
- berilganlarning maxfiyligi va alohidaligini ta'minlashi. Programmani yaratishga yo'naltirilmagan tizimlar Access va Paradox shu omillarni ta'minlaydi. Lekin bu omillar berilganlarni ajratilgan serverda saqlashda amalga oshirish va turli foydalanuvchilarning huquqlarini osongina cheklash mumkin.

Hamma BBBT, lokal tarmoqda ko'pgina foydalanuvchilarning bitta BB bilan ishlashini ta'minlaydi. Ba'zi tizimlar BB, ekran shakllari, hisobotlar, standart ilovalarni tezda yaratish uchun «usta» («master»), «quruvchi» («postroitel») va «ifodalar generatori» («generator vujrajeniy») kabi maxsus vositalarga ega.

Hozirgi vaqtida BBB tizimlarining, asosida berilganlar modelining ko'p o'lchovli jadvallari yotgan shakllari amalga oshirilmoqda. Bularda BB va programmalash bosqichlari obyektga yo'naltirilgan yondashuv tamoyillari asosida amalga oshiriladi.

### Xulosha

*BBB tizimlarning yaratilishi, berilgan bilan ishlashda fayl tizimi bilan bog'liq bo'lgan muammolarni hal qilish maqsadida yaratilgan. Berilganlar bazasi – bu birgalikda ishlataladigan mantiqiy bog'langan berilganlar termasidan iborat bo'lib, tashkilotning axborotga bo'lgan ehtiyojini qondirishga xizmat qiladi. BBBT – bu programma ta'minotidan iborat bo'lib, foydalanuvchilarga berilganlar bazasini aniqlash, yaratish va xizmat ko'rsatish hamda unga kirishni boshqarish imkoniyatini beradi.*

Berilganlarga kirish BBBT orqali amalga oshiriladi. Buning uchun foydalanuvchilarga, berilganlar bazasini tuzilishini aniqlashga – DDL tili yoki shu vazifani bajaruvchi modellashirish vositalari hamda berilganlarni boshqarish tili – DML xizmat qiladi.

*BBBT foydalanuvchilarning berilgan bazasiga kirishini nazorat qili-shni tashkillashtiradi. Berilganlarning xavfsizligi va butunligini ta'minlash uchun vositalar beradi, ko'pgina ilovalarning parallel ishlashi, nusxalash va tiklash uchun imkoniyatlar yaratadi, hamda foydalanuvchilar kirishi mumkin bo'lgan kataloglarni tashkillashtiradi.*

*Namunaviy BBB tizimlarida foydalanuvchilar ishlaydigan berilganlarning ko'rinishini soddalashtiruvchi tasavvurlarni yaratish mexanizmi ham mavjud.*

*BBB tizimining muhiti apparat ta'minot (kompyuterlar va chetki qurilmalar), programma ta'minoti (BBBT, opratsion tizim va ilovalar), berilganlar, protseduralar va foydalanuvchilar (berilganlar va berilganlar bazasining administratorlari, berilganlar bazasini loyihalashtiruvchilar, amaliy programmachilar va oxirgi foydalanuvchilar)dan iborat.*

### **Nazorat uchun savollar**

1. BBB tizimlariga misollar keltiring.
2. Berilgan va berilganlar bazasi atamalari orasidagi farqlarni izohlab bering.
3. BBB tizimining ta'rifini keltiring.
4. Berilganlarning bog'liqsizligi, deganda nimani tushunasiz?
5. Berilganlarning butunligi nimani ta'minlaydi?
6. Tranzaksiyani ta'riflang va asosiy xossalarni aytib bering.
7. Tasavvur deganda nimani tushunasiz?
8. Protsedurali va protsedurasiz DML tillari orasidagi farqni tushuntirib bering.
9. Berilganlarga ishlov berishdagi fayllar tizimidan foydalanishdagi yondashuvni tavsiflab bering va asosiy kamchiliklarini ko'rsating.
10. Berilganlar bazasiga asoslangan yondashuvning asosiy tavsiflarini keltiring va ularni oddiy fayllar tizimlarining tavsiflari bilan taqqoslang.
11. BBB tizimlaridagi asosiy beshta tarkibni tavsiflang. Ularning o'zaro aloqasini tushuntiring.
12. Berilganlar administratorining asosiy vazifasini tushuntiring.
13. Berilganlar bazasi administratorining asosiy vazifasini tushuntiring.
14. Berilganlar bazasini mantiqiy loyihalovchining asosiy vazifasini tushuntiring.
15. Berilganlar bazasini fizik loyihalovchining asosiy vazifasini tushuntiring.
16. Amaliy programmachi va berilganlar bazasi orasidagi bog'liqliklar nimadan iborat?

17. Foydalanuvchilarning guruhlarini sanab bering.
18. Oxirgi foydalanuvchi va berilganlar bazasi orasidagi bog'liqliklar nimadan iborat?
19. BBB tizimlarining asosiy afzallik tomonlari nimadan iborat?
20. BBB tizimlarining asosiy kamchiliklari nimadan iborat?
21. To'rtinchi avlod tillarining uchinchisi avlod tillaridan farqi nimada?
22. BBB tizimlarining asosiy turlarini ta'riflab bering.
23. BBBT qanday vositalarga ega?
24. BBB tizimlarining asosiy omillari nimalardan iborat?

## 22-bob. QBE TILI

**Tayanch iboralar:** namuna bo'yicha so'rov, namoyishli usul, tanlov so'rovi, dinamik berilganlar termasi, dinamik terma, QBE tilining to'ri, tanlov sharti, ko'p jadvallli so'rovlar, umumlashtiruvchi so'rovlar, parameetrlik so'rovlar, kesishgan so'rovlar, nusxalarini tanlashga so'rovlar, o'ziga mos bo'lmagan yozuvlarni tanlash, faol so'rovlar, jadvallarni yaratuvchi faol so'rovlar, yozuvlarni yo'qotuvchi faol so'rovlar, yozuvlarni yangilovchi faol so'rovlar, yozuvlarni qo'shuvchi faol so'rovlar, avtomatik almashtirishli so'rovlar.

Ushbu bobda biz QBE (Query-by-Example) namuna bo'yicha so'rov tilining BBBT MS Access funksional imkoniyatlaridan foydalangan holda, alohida xususiyatlari bilan tanishib chiqamiz. QBE tilida so'rovlar andazasini [54,55] qo'llashga asoslangan berilganlar bazasiga kirishni tashkil-lashtirishning namoyishli usulidan foydalanilgan. QBE tilidan foydalanish so'rovlar andazasida qiymatlarining namunalarini bergen holda, berilganlar bazasiga shu zahoti kirib, javob olish va namoyishli tarzda natijani tas-virlash imkoniyatini yaratadi.

QBE tili 1970-yillarda IBM kompaniyasi tomonidan yaratilgan bo'lib, foydalanuvchilarga berilganlar bazasidan ma'lumotlarni tanlashni amalga oshirish uchun mo'ljallangan. U foydalanuvchilar uchun shunchalik zarur tilga aylandiki, hozirgi kunda deyarli hamma BBB tizimlarida ishlab turibdi.

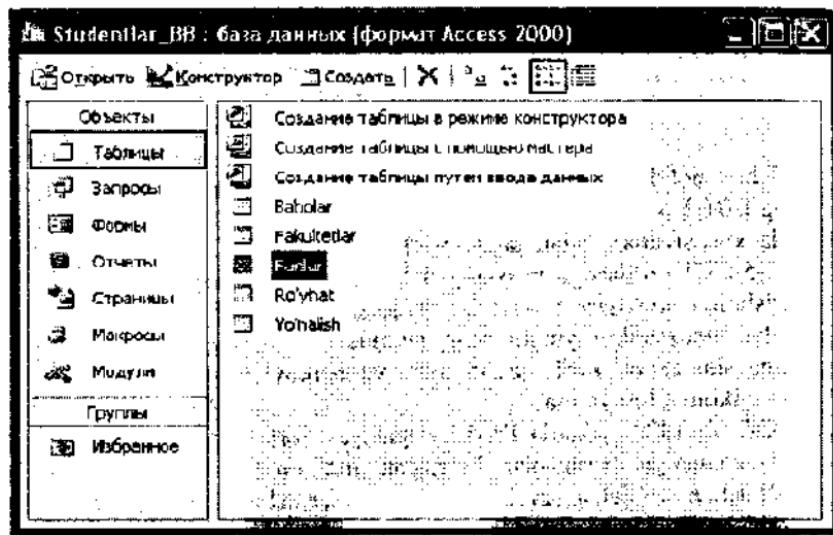
QBE tilining vositalaridan jadvallar ustida turli amallarni bajarishda, jumladan yozuvlarni kiritish va yo'qotish, maydon qiymatlarini o'zgartirish yoki yangi maydonlarni va jadvallarni yaratish imkoniyatlari mavjud. Bu imkoniyatlarining hammasini amaliy tomonidan ko'rsatish uchun Talabalar o'quv loyihamizning berilganlar bazasi chizmasidan va avvalgi boblarda keltirilgan munosabatlardan foydalanamiz.

BBBT MS Access QBE tilidan foydalanib so'rovlar yaratilganda, oshkormas ravishda ko'rsatilgan harakatlarni amalga oshirish uchun SQL tilining operatorlarida shu so'rovning ekvivalentini shakllantiradi. SQL tili relatsion BB uchun so'rovlarini bajarish, yangilash va xizmat qilish uchun keng qo'llaniladi. Biz SQL tilini keyingi boblarda o'rganamiz. Bu bobda esa, QBE tilidagi har bir misolda yaratilgan so'rov uchun, SQL tilida uning ekvivalentini Talabalar o'quv loyihamizning berilganlar bazasi misolda keltiramiz.

Bu bobni amaliy jihatdan o'zlashtirishda qiyinchilik tug'ilmasligining sababi, bu erda bajariladigan relatsion amallarning hammasini biz 10- bobda nazarij jihatdan misollar yordamida o'rganib chiqqanmiz.

### MS Access BBB tizimida so'rovlarini generatsiya qilish vositalari

MS Access BBB tizimida berilganlar bazasini yaratish yoki ochishda bazadagi hamma obyektlar bazaning oynasida akslanadi. Shuning uchun ham, Talabalar berilganlar bazasini ochganimizda oynada shu bazadagi hamma jadvallar keltiriladi (22.1-rasm).



22.1-rasm.

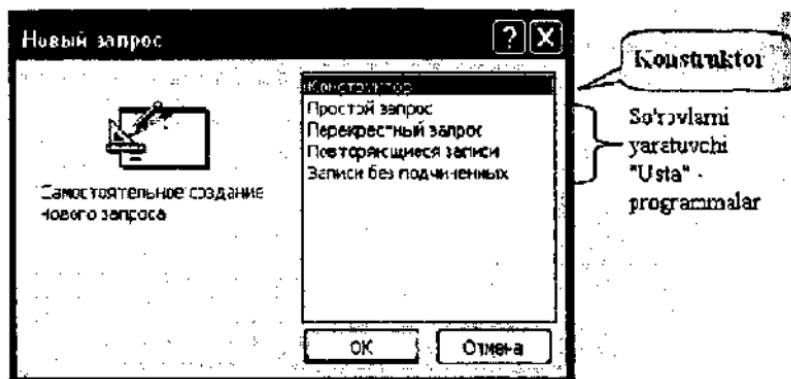
Berilganlar basasidagi ma'lumotlarga yo'naltirilgan so'rovni shunday shakllantirish kerakki, BBB tizimiga bizni qaysi berilganlar qiziqtirishini ko'rsata bilishimiz kerak. Eng ko'p ishlataladigan so'rovlardan bittasi berilganlarni tanlashga yo'naltirilgan bo'lib, uni *tanlov so'rovi* (Select Query) deb atashadi. Tanlov so'rovi bitta yoki bir nechta jadvallardagi berilganlarni ko'rish, tahlil qilish yoki o'zgartirish imkoniyatlarini beradi. Tanlov so'rovini bajarishda MS Access BBBT tanlangan berilganlarni *dinamik berilganlar termasiga* (dynaset) joylashtiradi. *Dinamik termasiga* dinamik tarzadagi tasavvur bo'lib, bir yoki bir nechta jadvallardan terib olingan berilganlarni saqlaydi. Boshqacha aytganda, dinamik berilganlar termasi doimiy yangilanuvchi obyektdan iborat.

Tanlov so'rovlaridan tashqari, MS Access BBB tizimida boshqa bir qancha so'rov turlarini ham yaratish mumkin. 22.1-jadvalda MS Access BBB tizimidagi turli so'rovlarning qisqacha tavsiflari keltirilgan.

22.1-jadval

So'rov turri	Tavsifi
Tanlov so'rovsi	Bir yoki bir nechta jadvallardan berilganlarni alomat bo'yicha tashashni aniqlash
Agregatli so'rov	Yozuvlarning aniq quruhlaridagi berilganlar ustida hisoblashni bajarish
Parametrik so'rov	So'rovning aniq qiymatlarini berish uchun avvaldan aniqlangan bitta yoki bir nechta muloqat oynalishni chiqarish. Orqali bajarish
Nusxalarni tashashga so'rov	Bitta jadvaldagi takrorlanuvchi yozuvlarni aniqlash
Mos bo'limgan yozuvlarni tashashga so'rov	Bog'langan jadvallardan bitta jadvaldagi yozuvning mos bo'limganlarini boshqa jadvaldandan aniqlash
Ketishgan so'rovlar	Katta hajmdagi berilganlarni yig'indisini hisoblash va kichik elektron jadval shaklidu taqdidi eisitishni bajarish
Avtomatik almashirishli so'rov	So'rovning bezarisilish jarayonida yangi yaratiladigan yozuvlarning aynim qiymatlar avtomatik almashiriladi
Faqi so'rovlar (jumladan yo'qotish, qo'shish, yangilash va jadvallarni yaratish)	Bitta amal yordamida bir nechta yozuvlarni o'zgartirish. O'zgartirish jadvaldagi yozuvlarni yo'qotish, qo'shish yoki yangilashdan hamda yangi jadvallarni yaratishdan iborat
SQL tilining maxsus so'rovlar (birlashtirish, uzatish, berilganlarni aniqlash va qism so'rovlar)	Ushbu so'rov, yuqorida keltirilgan so'rovarning turlarini mukammallashirish va shakl hamda hisobolaming xossalari ni aniqlashda ishlataladi. Bu so'rovlarla SQL tilining maxsus vositalardan mafsalan, birlashtirish amali, berilganlarni aniqlash operatorlari, qismi so'rovlar, hamda uzatiladigan so'rovlardan foydalaniшiz mumkin. Uzatiladigan so'rovlar SQL tilining operatorlaridan tuzilgan bo'lib, SQL Server BBBT ga uzatiladi.

Yangi so'rovni yaratish protsedurasidan avval MS Access BBBT 22.2-rasmida keltirilgan yangi muloqot oynasi New Query (yangi so'rov)ni ochadi. Ushbu oynada keltirilgan bajarish mumkin bo'lgan harakatlarni ro'y-xati yangi so'rovni yaratishni boshidan boshlash va hamma talab qilinadigan harakatlarni mustaqil bajarishni (Desing View – konstruktur), yoki qolgan harakatlarni MS Access BBBT «usta»si (Mactep) orqali bajarishni taklif qiladi. «Usta» berilganlar bazasidagi yordamchi programmalar naqlarining biridan iborat. U foydalanuvchiga bir qator savollar berib, olingen javoblar asosida kerakli bo'lgan obyektni generatsiya qiladi.



22.2-rasm.

### Tanlov so'rovini yaratishdagi QBE vositalari

Berilganlarni tanlash bo'yicha so'rovlar eng ko'p ishlatalidigan so'rov turidan iborat. Ular bir yoki bir nechta jadvallardan berilganlarni tanlab, natijalarni to'r shaklida namoyon qiladi va undagi berilganlarni (ayrim cheklashlar bilan) o'zgartirish imkoniyati yaratiladi. Jadvallardan tanlab olingan ma'lumotlar to'rda, oddiy elektron jadval kabi, satr va ustun tarzida akslanadi. Tanlov so'rovlari yozuvlarni guruhlash, yig'indilar, hisobotlar, o'rta qiymatlarni hisoblash va agregat funksiyalarning boshqa turlarini qo'llash imkoniyatini beradi.

Oddiy tanlov so'rovini yaratishda, «usta»dan foydalansak ham bo'ladi, uni 22.2-rasmdagi muloqot oynasidan chaqirish mumkin. Lekin biz oddiy tanlov so'rovini konstruktur orqali «usta»siz yaratishni ko'ramiz. Berilganlar bazasiga yo'naltirilgan so'rovni yaratishda, Select Query (tanlov so'rovi) oynasi ochiladi va ekranga muloqot oynasi chiqadi. Bizning holatda esa Talabalar o'quv loyihasidagi berilganlar bazasining jadval va so'rovlari chiqadi. Shu oyna orqali foydalananuvchi o'zini qiziqtirgan jadval va/yoki so'rovlarini ko'rsatishi kerak.

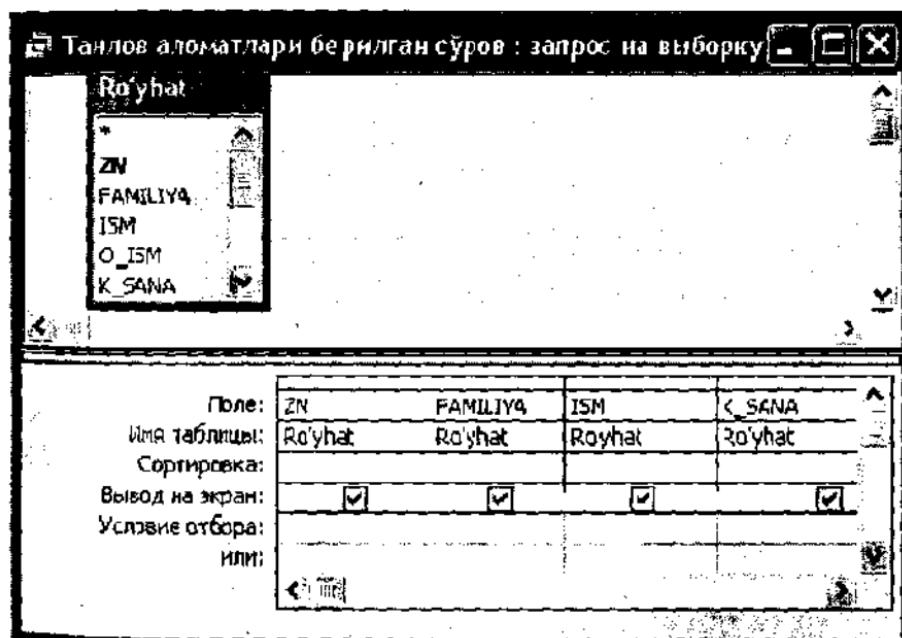
Tanlov so'rovi oynasi QBE tilining grafik vositasidir. Bizni qiziqtirgan yozuvlar namunasini aniqlash uchun, «sichqoncha» orqali kerakli obyektlarni tanlash, surish yoki ustida amallar bajarish mumkin. So'rov natijalariga kiruvchi maydan va yozuvlarni aniqlash, QBE tilining to'rda bajariladi.

*QBE tilining to'rda so'rovlar shakllantirilganida, MS Access BBBT oshkormas ravishda SQL operatorlari orqali ekvivalent so'rovni generatsiya*

qiladi. SQL operatorlarini ko'rish va tahrirlash uchun SQL oynasini ochish kerak. Bu bobda, biz ekvivalent SQL operatorlarini, QBE tilining to'rida yaratilgan yoki «usta» orqali yaratilgan har bir so'rov uchun keltiramiz. Shuni ham ta'kidlab o'tamizki, MS Access BBB tizimida generatsiya qilingan va keltirilgan SQL operatorlari SQL tilining standartiga mos kelmasligi mumkin.

### Tanlov alomatlarini berish

Tanlov alomatlarini deganda, foydalanuvchi uchun zarur berilganiarni maydon yoki yozuvlardan tanlash uchun qo'yilgan shartlar tushuniladi. Masalan, RO'YHAT jadvalidan talabaning reyting daftarining raqami, familiyasi, ismi va o'qishga kirgan sanasi ko'rsatilgan ustunlarni tanlash uchun QBE to'rida tayyorlangan so'rov 22.3-rasmida keltirilgan. Olingan QBE to'ridagi natija 22.4-rasmida, ekvivalent SQL operatorlarida generatsiyalangan so'rov esa 22.5-rasmida keltirilgan.



22.3-rasm.

Танлов аломатлари берилган сўров ...

Зачетка №	Фамилия	Исм	Кирган сана
5005227	Сангиров	Санжар	31.08.2006
5009080	Фаёзов	Али	30.08.2007
5012023	Эшонов	Бахром	01.10.2007
5012305	Зуннунов	Кодир	31.08.2007
5012345	Исақулов	Мамур	30.08.2008
5012789	Халиков	Темур	31.08.2007
5021356	Алиева	Барно	29.08.2008
5045544	Моннова	Чарос	29.08.2007
5054687	Яминов	Амин	30.08.2009
5076000	Имубаев	Восик	30.08.2009
5080809	Алиева	Аълоҳон	30.08.2007
5110354	Салимов	Шохир	31.08.2007
5123456	Тошев	Исок	31.08.2007
5205697	Хошимов	Бакир	31.08.2007
5235813	Иминов	Салим	13.09.2006

Запись: |◀| ◀ |▶| ▶|\*| на 15

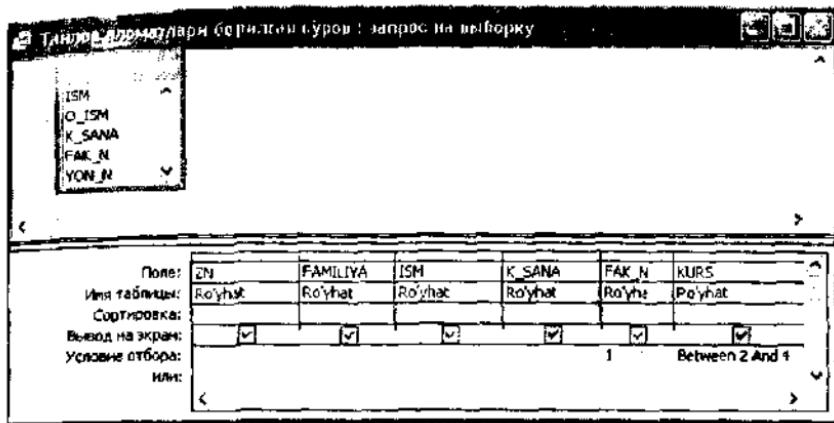
22.4-rasm.

Танлов аломатлари берилган сўров ...

```
SELECT [Ro'yhat].ZN, [Ro'yhat].FAMILIYA, [Ro'yhat].ISM,
[Ro'yhat].K_SANA
FROM [Ro'yhat];
```

22.5-rasm.

Faraz qilaylik, yuqorida keltirilgan tanlov so'rovi faqat «mexanika-matematika» fakultetining 2, 3 va 4 kurs talabalaridan iborat bo'lsin, degan qo'shimcha tanlov alomati (shart)ni kiritaylik. Buning uchun QBE to'ridagi «Tanlov sharti» (Условие отбора) satrining va Fak\_N ustunidagi kesishgan yacheykaga 1, Kurs ustuni bilan kesishgan yacheykaga esa «Between 2 And 4» ifodani kiritamiz. Ana shu keltirilgan shartlar bo'yicha QBE to'ridagi tanlov so'rovi 22.6-rasmda keltirilgan. 22.7-rasmida keltirilgan so'rov bo'yicha olingan natijaning QBE to'ridagi tasviri, 22.8-rasmida esa ekvivalent tarzda generatsiyalangan SQL tilidagi so'rov keltirilgan.



22.6-rasm.

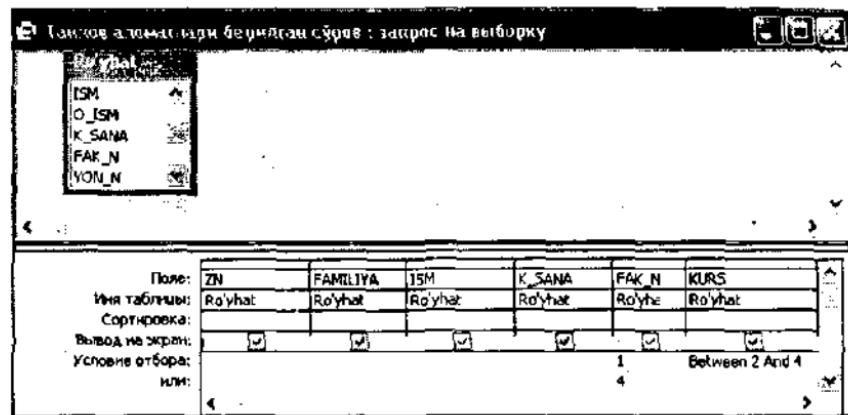
Зачетка №	Фамилия	Имя	Киргана сана	Факультет	Курс
E005227	Сангиров	Санжар	31.08.2005	мектмат	4
E012023	Эшоное	Бакром	01.10.2007	мектмат	4
E012305	Зуннунов	Кодир	31.08.2007	мектмат	4
E012739	Халиков	Темур	31.08.2007	мектмат	4
E110354	Салимов	Шокир	31.08.2007	мектмат	3
E206697	Хошимов	Бакир	31.08.2007	мектмат	2

22.7-rasm.

```
■ Танчылған ақындардың сұрояуындағы мәттердің табло
SELECT [Roýhat].ZN, [Roýhat].F4M, [Roýhat].ISN, [Roýhat].Roýhat < _SANA,
[Roýhat].F4K_N, [Roýhat].KLRS
FROM [Roýhat]
WHERE ((([Roýhat].F4C_N)=1) AND (([Roýhat].KURS) Between 2 And 4))
```

22.8-rasm.

Yuqorida keltirilgan oxirgi tanlov so'rovini «mehanika-matematika» yoki «biologiya» fakultetining 2, 3 va 4 kurs talabalaridan iborat bo'lsin, degan qo'shimcha tanlov alomatiga o'zgartiramiz. Buning uchun QBE to'ridagi «ili» satri va Fak\_N ustuni kesishgan yacheykaga 4 kiritamiz, qolgan yacheykalarni o'zgarishsiz qoldiramiz. Ana shu keltirilgan shartlar bo'yicha QBE to'ridagi tanlov so'rovi 22.9-rasmda keltirilgan. 22.10-rasmida keltirilgan so'rov bo'yicha olingan natijaning QBE to'ridagi tasviri, 22.11-rasmida esa ekvivalent tarzda generatsiyalangan SQL tilidagi so'rov keltirilgan.



22.9-rasm.

Зачетка №	Фамилия	Исм	Кирган сана	Факультет	Курс
6005227	Сангиров	Санжар	31.08.2006	мехмат	4
6012023	Эшонов	Бахром	01.10.2007	мехмат	4
6012305	Зуннунов	Кодир	31.08.2007	мехмат	4
6012345	Исақулов	Маймур	30.08.2008	биология	1
6012789	Халиков	Темур	31.08.2007	мехмат	4
6110364	Салимов	Шокир	31.08.2007	мехмат	3
6205697	Хошимов	Бакир	31.08.2007	мехмат	2
6235813	Иминов	Салим	13.09.2006	биология	4

22.10-rasm.

Тандов аломатлари берилген сурʼов : запрос на выборку

```

SELECT [Ruyshot].ZN, [Ruyshot].FAMILYA, [Ruyshot].ISM, [Ruyshot].K_SANA,
[Ruyshot].FAK_N, [Ruyshot].KURS
FROM [Ruyshot]
WHERE (((Ruyshot.FAK_N)=1) AND ((Ruyshot.KURS) Between 2 And 4)) OR
((Ruyshot.FAK_N)=4))

```

22.11-rasm.

Agarda ustun sarli turda boʼlsa, tanlanishi kerak boʼlgan qiymatlarni aniqlashda, almashtirish belgilari yoki Like operatori ishlatalishi mumkin. Bu holda tanlov berilgan qiymat boʼyicha yoki murakkabroq boʼlgan andaza boʼyicha bajariladi. Masalan, hamma talabalarning ichidan familiyasi «A» harfidan boshlanadigan talabalarni aniqlash kerak boʼlsin. Bu holda Like ‘‘A\*’’ konstruksiyasidan foydalaniлади. \* belgisi nomaʼlum sondagi belgilarni chiqarish uchun ishlataladi. Masalan, Like ‘‘\*’’ konstruksiyasini Familiya ustuni bilan «Tanlov sharti» satrining kesishgan yacheysigiga kiritsak, hamma familiya va talab qilingan ustunlarni chiqarib beradi. ? almashtirish belgisi boʼlib, bittagina nomaʼlum belgining joyini koʼrsatadi. Masalan, Like ‘‘Ali??’’ konstruksiyasi beshta harfdan iborat hamma»Ali» bilan boshlanuvchi soʼzlarni chiqarib beradi.

22.12-rasmida QBE toʼrida Like ‘‘A\*’’ konstruksiyasidan foydalanish boʼyicha, 22.13-rasmida olingan natija, 22.14-rasmida esa ekvivalent tarzda generatsiyalangan SQL tilidagi soʼrov keltirilgan.

Тандов аломатлари берилган сурʼов : запрос на выборку

ЗН Ruyshot	FAMILYA Ruyshot	ISM Ruyshot	K_SANA Ruyshot	FAK_N Ruyshot	KURS Ruyshot
1	Алиева	Барно	29.08.2006	ХИМИЯ	Between 2 And 4
2	Алиева	Аълоҳон	30.08.2007	филология	

22.12-rasm.

Тандов аломатлари берилган сурʼов : запрос на выборку

Зачетка №	Фамилия	Имя	Киргэн сана	Факультет	Курс
6021356	Алиева	Барно	29.08.2006	ХИМИЯ	2
6080809	Алиева	Аълоҳон	30.08.2007	филология	3

22.13-rasm.

Танлов автоматлари берилган сурʼов : запрос на выборку

```
SELECT [Ro\u043fhat].ZN, [Ro\u043fhat].FAMILIYA, [Ro\u043fhat].ISM, [Ro\u043fhat].K_SANA,
[Ro\u043fhat].FAK_N, [Ro\u043fhat].KURS
FROM [Ro\u043fhat]
WHERE ((([Ro\u043fhat].FAMILIYA) Like 'A%') AND ((([Ro\u043fhat].KURS) Between 1 And 4));
```

22.14-rasm.

### Ko‘p jadvalli so‘rovlarini yaratish

To‘g’ri normallashgan berilganlar bazasida bog’langan berilganlar bir nechta jadvallarda saqlanishi mumkin. Shuning uchun ham, BBB tizimida bir-biri bilan bog’langan berilganlarga ishlov berish imkoniyatining mavjud bo‘lishi juda katta ahamiyatga ega.

Bir nechta jadvallardan kerakli berilganchani tanlab olish uchun so‘rov oynasida zarur bo‘lgan jadvallar taqdim etish va QBE to‘rida tanlov alomatlari aniqlanishi kerak. Masalan, Talabalar o‘quv loyihasidan BBBT fanidan 70 dan yuqori ball olgan talabalar reyting daftari, familiyasi va ismini aniqlash kerak bo‘lsin. Natijaviy jadvalda RO‘YHAT jadvalidan talabaning reyting daftarining raqami, familiyasi, ismi; BAHOLAR jadvalidan Fan\_N va Ball ustunlari; FANLAR jadvalidan esa Fan\_Nomi ustuni tanlanadi. 22.15-rasmida so‘rov oynasida bog’langan jadvallar va so‘rovni QBE to‘rida shakllantirish keltirilgan.

Ко‘п жадвалин сурʼовлар : запрос на выборку

Поне:	ZN	FAMILIYA	ISM	FAN_NOMI	BALL
Имя таблицы:	Ro'yhat	Ro'yhat	Ro'yhat	Fanlar	Baholar
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				"667"	>70
или:					

22.15-rasm.

Кўп жадвалли сўровлар : запрос на выборку

Зачетка №	Фамилия	Имя	Фан номлари	Бало
Б012305	Зуннунов	Кидир	БББТ	71
Б012709	Халиков	Темур	БББТ	66
Б123456	Тошев	Исок	БББТ	76

Запись: [1] [2] [3] [4] [5] [6] из 3

22.16-rasm.

Кўп жадвалли сўровлар : запрос на выборку

```
SELECT [Ro'yhat].ZN, [Ro'yhat].FAMILIYA, [Ro'yhat].ISM, Fanlar.FAN_NOMI,
      Fanlar.BALI
FROM ([Ro'yhat] INNER JOIN Baholar ON [Ro'yhat].ZN = Baholar.ZN) INNER
      JOIN Fanlar ON Baholar.FAN_N = Fanlar.FAN_N
WHERE ((Fanlar.FAN_NOMI) = "БББТ") AND ((Baholar.BALI) > 70);
```

22.17-rasm.

Ko'p jadvalli so'rov (22.15, 22.16 va 22.17-rasmlar) munosabatlarni tabiiy birlashтиришга (10- bob) misol bo'ladi. Jadvaliar orasidagi o'zaro aloqalar 1:1 aloqalar mavjudligini tizim jadvallardagi maydonlar o'zaro bog'liq bo'lgani uchun avtomatik tarzda aniqlagan. 22.16-rasmida so'rov natijasida olingan jadval, 22.17-rasmida esa ekvivalent avtomatik tarzda hosil bo'lgan SQL tilidagi so'rov keltirilgan.

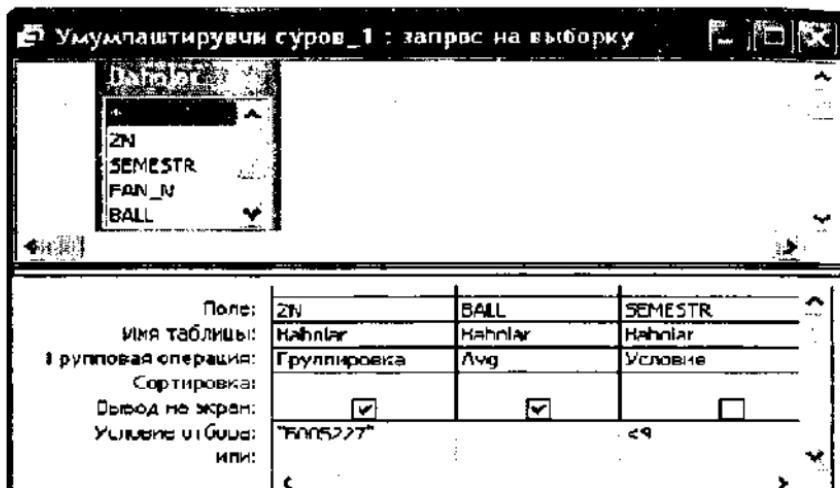
### Umumlashtiruvchi so'rovlar

Ko'p holatlarda berilganlarning u yoki bu guruhlarini umumlashtirish zaruriyat tug'iladi. Masalan, nechta talaba «Fizika» fanidan a'lo, yaxshi yoki qoniqarli baholar olgan? Bo'lim xizmatchilarining o'rtacha oyligi qancha? 2011-yilda o'qishga qabil qilingan talabalar soni qancha?

Yozuvlar guruhi ustida agregat hisoblashlarni amalga oshirish umumlashtiruvchi so'rovlar orqali amalga oshiriladi. MS Access BBBT turli umumiyy hisoblarni amalga oshirish imkoniyatini yaratgan, jumladan **Sum** – yig'indini hisoblash, **Avg** – o'rta qiymatni hisoblash, **Max** – maksimal qiymatni aniqlash, **Min** – minimal qiymatni aniqlash, **Count** – nuxsalarni sanash uchun xizmat qiladi. Bu funksiyalardan foydalanish uchun so'rov turini «Guruhli amal» (Групповая операция) turiga o'zgartirish kerak, natijada QBE to'rida shu nomdagi qo'shimcha satr paydo bo'ladi.

Masalan, 4-kursni bitirgan, reyting daftarining raqami B005227 bo'lgan talabaning hamma fanlar bo'yicha o'rtacha balini hisoblaylik. Buning uchun, BAHOLAR jadvalidan Zn, Ball va Semestr maydonlarini QBE

to'rida akslantiramiz, so'rov turidan «Групповая операция»ni tanlaymiz. Ball va «Групповая операция» kesishgan yacheykaga ro'yxatdan Avg funksiyasini tanlaymiz, keyin tanlov shartlarini kiritamiz. Shu umumilashituvchi so'rov shakllangan tasvir 22.18 rasmida keltirilgan, so'rovning natijasi 22.19-rasmida, ekvivalent avtomatik tarzda hosil bo'lgan SQL tilidagi so'rov 22.20-rasmida keltirilgan.



22.18-rasm.

Зачетка №	Avg-BALL
B0005227	59,5

22.19-rasm.

```
SELECT Baholar.ZN, Avg(Baholar.BALL) AS [Avg-BALL]
FROM Baholar
WHERE ((Baholar.SEMESTR)<9))
GROUP BY Baholar.ZN
HAVING (((Baholar.ZN) = "B0005227"));
```

22.20-rasm.

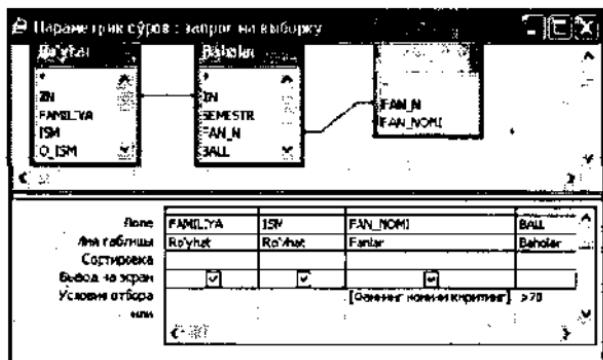
## QBE so'rovlarining murakkab turlari

MS Access BBB tizimida so'rovlarning bir qator takomillashgan turlari mavjud. Shularidan eng keng qo'llaniladiganlari quyidagilardan iborat:

- parametrik so'rovlar;
- kesishgan so'rovlar;
- nusxalarni tanlashga so'rovlar;
- o'ziga mos bo'limagan yozuvlarni tanlash.

### Parametrik so'rovlar

Parametrik so'rovlar orqali foydalanuvchi avvaldan tayyorlangan muloqot oynalari orqali so'rovning parametrlarini kiritadi. Masalan, Talabalar o'quv loyihasidan «X» fanidan 70 dan yuqori ball olgan talabaning familiyasi va ismini aniqlash kerak bo'lсин. Buning uchun, «Условие отбора» (Tanlov sharti)ga kvadrat qavsda foydalanuvchiga murojaat uchun tayyorlangan matn yoziladi. 22.21-rasmida QBE to'ridagi so'rov, 22.22 rasmda foydalanuvchi bilan bo'ladijan muloqot oynasi, 22.23-rasmida so'rov natijasi, 22.24-rasmida esa avtomatik tarzda hosil bo'lgan SQL tilidagi ekvivalent so'rov keltirilgan.



22.21-rasm.



22.22-rasm.

Изображение 22.23: Скриншот программы Microsoft Access с результатом выполнения запроса на выборку. Таблица содержит 3 строки с данными о фанах:

Фамилия	Имя	Фан номигарн	Балы
Зулупов	Кодир	БББТ	71
Халиков	Темур	БББТ	88
Гишаев	Ислак	БББТ	71

22.23-rasm.

Изображение 22.24: Скриншот программы Microsoft Access с текстом SQL-запроса, генерированного для выполнения в предыдущем окне.

```

SELECT Гаро'йнат.ФАМИЛИЯ, Гаро'йнат.ИМ,
    Гаро'йнат.ФАН_НОМ, Бахолар.ВАЛ
FROM Гаро'йнат INNER JOIN ((Ро'йнат) INNER JOIN Бахолар
ON [Ро'йнат].ЭН = Бахолар.ЭН) ON Гаро'йнат.ФАН_Н =
    Бахолар.ГАН_Н
WHERE (((Гаро'йнат.ФАН_НОМ)>=Фаннинг номини
    киртирилган) AND ((Бахолар.ВАЛ)>70));

```

22.24-rasm.

### Kesishgan so'rovlar

Kesishgan so'rovlar berilganlarga ishlov berish va ularni ixchani elektron jadval shaklida akslantirish uchun ishlataladi. Bunday shakl katta hajmda berilganlarni ixcham ko'rinishda tasvirlagan holda mavjud mayliklarni topish va taqqoslab tahlil qilish maqsadida ishlataladi. Kesishgan so'rov natijasidagi to'r berilganlarning shu daqiqadagi tasviridan iborat bo'lib, unga hech qanday o'zgartirishlar kiritish mumkin emas. Kesishgan so'rovni yaratish uchun, «Перекрестный запрос» (Kesishgan so'rov) «csta»-sidan yoki mustaqil ravishda QBE to'ridan foydalanish mumkin. Kesishgan so'rovni yaratish umumilashgan so'rovlarni yaratishga o'xshab ketadi, lekin bunda berilganlarning boshlang'ich qiymatlarini saqlaydigan qo'shimcha maydonlarni ko'rsatish talab qilinadi, chunki ular ustun va satrlarning sarlavhasi sifatida ishlataladi.

Masalan, har bir talabaning har bir fandan olgan o'rtacha bahosini aniqlash zarur bo'lsin. Buning uchun avval Talabalar o'quv toyihasining berilganlar bazasidagi RO'YHAT, FANLAR va BAHOLAR jadvallari ishtirokidagi tanlov so'rovini yaratamiz. Bu so'rovning natijasidagi to'r 22.25-rasmida keltirilgan.

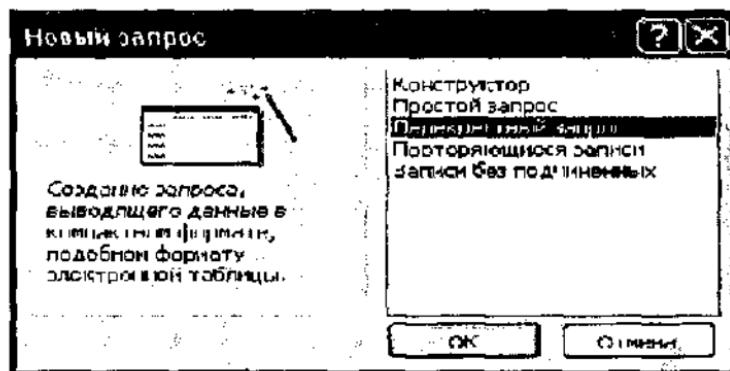
Ro'yhat Запрос 2 : запрос на выборку

Фамилия	Имя	Факультет	Балл
Эшринов	Бахром	Информатика	72
Саидиров	Сайкар	БББТ	80
Зулфуров	Кодир	БББТ	71
Халиков	Темур	БББТ	86
Сатимов	Шокир	ИИКИ	74
Санжарев	Ислак	ИИКИ	71
Хошимов	Бакир	Ахборот технологиялари	82
Файзик	Али	Информатика	70
Исақулов	Маммур	Ахборот технологиялари	87
Момилова	Чарос	Физика	92
Саидиров	Сайкар	Алгебра	69
Имубаев	Восик	Информатика	88
Иминов	Салич	Алгебра	77
Сангиров	Санжар	Математик анализ	84
Сангиров	Санжар	Математик анализ	70
Сангиров	Санжар	Математик анализ	75
Сангиров	Санжар	Математик анализ	80

Запись: [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] из 17

22.25-rasm.

Endi, ana shu tanlov so'rovini kesishish so'roviga aylantirish uchun «Перекрестный запрос» (Kesishgan so'rov) «уставидан foydalanamiz. Buning uchun 22.26 – 22.29-rasmlardagi akslanishlarni «usta»ning savollariга ketma-ket javob berishlar natijasida aniq hosil qilamiz.



22.26-rasm.

## Создание перекрестных таблиц

Выберите таблицу или запрос, поля которых необходимо вывести в перекрестной запросе.

Для включения полей из нескольких таблиц сначала создайте обычный запрос, содержащий все необходимые поля.

- Запрос: Baholar\_перекрестный
- Запрос: Baholar\_перекрестный1
- Запрос: Baholar\_перекрестный2
- Запрос: Baholar\_перекрестный3
- Запрос: Ro'yunat Запрос
- Запрос: Ro'yunat Запрос1
- Запрос: Umumiy

**Показать**

Таблицы  Запросы  Таблицы и запросы

	Заголовок1	Заголовок2	Заголовок3
		ИТОГИ	

**Отмена**

**Создан**

**Далее >**

**Назад**

22.27-rasm.

## Создание перекрестных таблиц

Какие вычисления необходимо провести для каждой суммы на перекрестном табл. итогов?

Например, можно вычислить сумму расходов для каждого вида услуг (столбец) по строкам и районам (строка).

Включите, чтобы приложение для каждой строки?

Да.

**Поля:**

**БРЛ**

**Пунктами:**

- Десетый
- Максимум
- Минимум
- Среднее значение
- Первый
- Последний
- Сумма
- Число

FAMILIYA	ISM	ГАН. НОМЕР	ГАН. НОМЕР2	ГАН. НОМЕР3
FAMILIYA1	ISM1		Среднее(BALL)	
FAMILIYA2	ISM2			
FAMILIYA3	ISM3			
FAMILIYA4	ISM4			

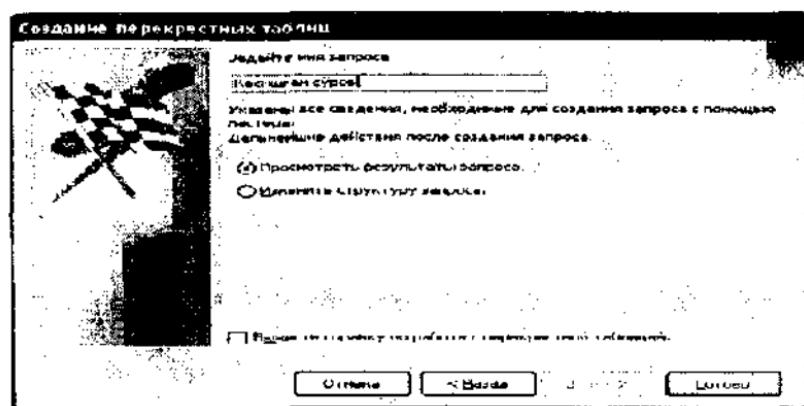
**Назад**

**Назад**

**Далее >**

**Назад**

22.28-rasm.



22.29-rasm.

Natijada 22.30-rasmdagi, tanlov so'rovi asosidagi QBE to'rida kesishgan so'rov hosil bo'ladi. Kesishgan so'rovning natijasi 22.31 rasmda, avtomatik tarzda hosil bo'lgan SQL tilidagi ekvivalent so'rov esa 22.32 rasmda keltirilgan.

← Несколько строк, перекрестный запрос					
Вывод					
<b>Вывод:</b>					
Имя собрата:	FAMILIA	ФИА	FAN_NOMI	ФАН_НОМІ	ИТОГОВОЕ ЗНАЧЕНИЕ ФАН_НОМІ
Группировка:	Большой Запрос2	Большой Запрос2	Большой Запрос2	Большой Запрос2	Большой Запрос2
Перекрестная таблица:	Группировка	Группировка	Группировка	Лог.	Сумм.
Сортировка:	Значение	Значение	Значение	Значение	Значение
Условия отбора:					

22.30-rasm.

← Итоговая таблица перекрестного запроса						
Фамилия	Имя	Итоговое значение	Лог.табл	Лог.перегрупп.	БЕБТ	Информатика
Зуников	Кодир	1			71	
Изимов	Салым	1	77			
Ишувасо	Восик	1			89	
Изакуши	Минмул	1				
Мониров	Нарс	1				82
Султанов	Шамир	1			66	
Султанов	Санжар	6	54		54	71
Ташев	Игорь				76	
Фадилов	Ази				77	
Халиков	ТӨМҮР				88	
Халикова	Ганин	1			70	
Эшшолов	Бахров				72	

22.31-rasm.

Кесиңгай сұрағы : перекрестный запрос

```
TRANSFORM Avg([Ro'yhat Запрос2].BALL) AS [Avg-BALL]
SELECT [Ro'yhat Запрос2].FAMILYA, [Ro'yhat Запрос2].ISM, Count([Ro'yhat Запрос2].BALL) AS [Итоговое значение BALL]
FROM [Ro'yhat Запрос2]
GROUP BY [Ro'yhat Запрос2].FAMILYA, [Ro'yhat Запрос2].ISM
PIVOT [Ro'yhat Запрос2].FAN_NOMI;
```

22.32-rasm.

### Nusxalarni tanlashga so'rov

Nusxalarni tanlashga so'rov natijasida jadvalda takrorlanuvchi yozuvlarni hamda jadvaldagı qaysi yozuvlar qaysi bir ustunning maydonida takrorlanganini aniqlash mumkin. Masalan, qaysi talabalar bir xil qiymatli ballarni nechta fandan olganini aniqlash zarur bo'lisin. Buning uchun, nusxalarni tanlashga so'rov «*кластер*»ning savollariga javob bergan holda, 22.33-rasmidagi QBE to'rini hosil qilamiz. So'rov natijasi 22.34-rasmida, 22.35-rasmida esa avtomatik tarzda hosil bo'lgan SQL tilidagi ekvivalent so'rov keltirilgan.

Поиск повторений для Baholar : запрос на выборку

Поле:	ZN	BALL	поле: ZN	Повторы: ZN	ZN	ЕАН	ЕАН
Что искать:	Baholar	Baholar	Baholar	Baholar	Baholar	Baholar	Baholar
Групповая операция:							
Сортировка:							
Быстро к группам:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Установка отбора:							
АИК:							

22.33-rasm.

Поиск повторений для Bahola...

ZN	BALL	Повторы
Б005227	59	?

Запись: 1 из 1

22.34-rasm.

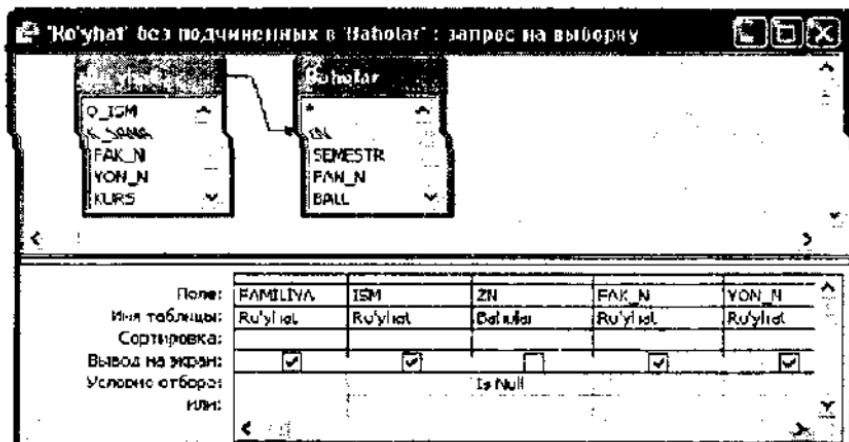
Пойск повторений для Baholar...

```
SELECT First(Baholar.ZN) AS [ZN none],
First(Baholar.BALL) AS [BALL none], Count(Baholar.ZN)
AS Повторы
FROM Baholar
GROUP BY Baholar.ZN, Baholar.BALL
HAVING (((Count(Baholar.BALL))>1));
```

22.31-rasm.

### O'ziga mos bo'limgan yozuvlarni tanlash

«Записи без подчиненных» (tobesiz yozuvlar) «usta»si orqali ko'rsatilgan jadvaldagi hamma yozuvlarning, boshqa jadvaldagi bog'lanmagan yozuvlarini topish mumkin. Masalan, RO'YHAT va BAHOLAR jadvalini bir-biriga solishtirib, birorta ham fanlardan ball oлмаган talabalarini aniqlash mumkin. «Usta» berilgan savollarga foydalanuvchining javoblari asosida so'rovni yaratadi (22.36-rasm).



22.36-rasm.

22.37-rasmdagi natijani tahlil qiladigan bo'lsak, RO'YHAT jadvalidagi uchta yozuvning birorta ham BAHOLAR jadvalidagi mos ravishda bog'langan yozuv yo'qligini ko'tramiz. Boshqacha aystsak, uchta talaba hali birorta fandan baho oлmaganini aniqladik.

**'Ro'yhat' bez podchinenных v 'Baholar': zapro...**

Фамилия	Имя	Факультет	Инфайл
Алиева	Барна	Химия	ДМАТ
Ямнина	Амин	История	Гуманист.
Ликова	Альбина	Филология	Рабок ти

Запрос: [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [57] [58] [59] [60] [61] [62] [63] [64] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75] [76] [77] [78] [79] [80] [81] [82] [83] [84] [85] [86] [87] [88] [89] [90] [91] [92] [93] [94] [95] [96] [97] [98] [99] [100] [101] [102] [103] [104] [105] [106] [107] [108] [109] [110] [111] [112] [113] [114] [115] [116] [117] [118] [119] [120] [121] [122] [123] [124] [125] [126] [127] [128] [129] [130] [131] [132] [133] [134] [135] [136] [137] [138] [139] [140] [141] [142] [143] [144] [145] [146] [147] [148] [149] [150] [151] [152] [153] [154] [155] [156] [157] [158] [159] [160] [161] [162] [163] [164] [165] [166] [167] [168] [169] [170] [171] [172] [173] [174] [175] [176] [177] [178] [179] [180] [181] [182] [183] [184] [185] [186] [187] [188] [189] [190] [191] [192] [193] [194] [195] [196] [197] [198] [199] [200] [201] [202] [203] [204] [205] [206] [207] [208] [209] [210] [211] [212] [213] [214] [215] [216] [217] [218] [219] [220] [221] [222] [223] [224] [225] [226] [227] [228] [229] [230] [231] [232] [233] [234] [235] [236] [237] [238] [239] [240] [241] [242] [243] [244] [245] [246] [247] [248] [249] [250] [251] [252] [253] [254] [255] [256] [257] [258] [259] [260] [261] [262] [263] [264] [265] [266] [267] [268] [269]

22.37-rasm.

22.38-rasmida ana shu so'rovning avtomatik tarzda hosil bo'lgan SQL tilidagi ekvivalenti keltirilgan.

**'Ro'yhat' bez podchinenных v 'Baholar': запро...**

```
SELECT [Ro'yhat].FAMILIYA, [Ro'yhat].ISM, [Ro'yhat].FAK_N,
[Ro'yhat].YON_N
FROM [Ro'yhat] LEFT JOIN Baholar ON [Ro'yhat].ZN = Baholar.ZN
WHERE (((Baholar.ZN) Is Null));
```

22.38-rasm.

O'ziga mos bo'limgan yozuvlarni tanlash so'rovi, 10-bobdag'i tashqi birlashtirish amaliga misol bo'ladi.

### Jadval maydonlarini faol so'rovlar yordamida o'zgartirish

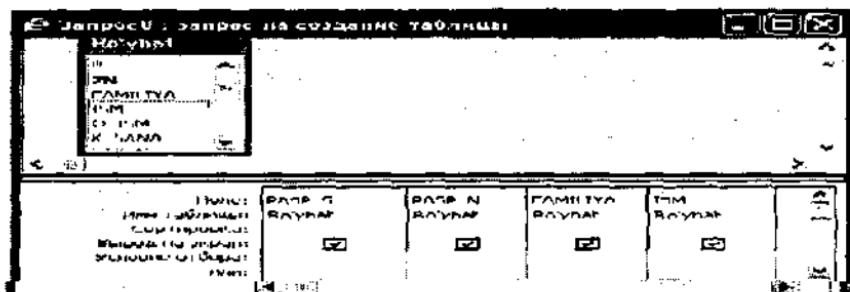
MS Access BBBT zaruriy o'zgartirishlarni so'rov natijasida olingan jadval maydonlariga ketma-ket kiritish imkoniyatini beradi. Agarda katta hajmdagi o'xshash o'zgartirishlarni kiritish zaruriyati tug'ilsa, faol so'rovlardan foydalanib, o'zgartirishlarni kiritish vaqtini ancha qisqartirish mumkin. Faol so'rov bir yo'la bir nechta yozuvlarni o'zgartirish imkoniyatini beradi. Faol so'rovlarining to'rt turi mavjud: jadvalni yaratishdagi so'rovlar, yozuvlarni yo'qotish, yangilash va qo'shishdagi so'rovlar.

#### Jadvallarni yaratishdagi faol so'rovlar

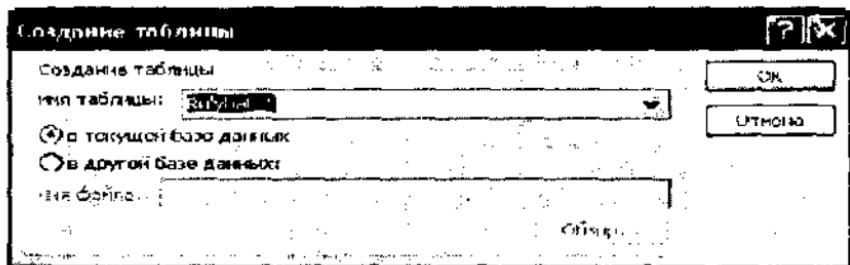
Jadvallarni yaratishga yo'naltirilgan faol so'rovlar yangi jadvallarni mavjud hamma jadvallar, bitta yoki bir nechta jadvallarning qismi asosida yaratish imkoniyatini beradi. Yangi yaratilgan jadval joriy berilganlar bazasida saqlanishi yoki boshqasiga eksport qilinishi mumkin. Yangi jad-

valdag'i berilganlar maydonlarning xususiyatlarini voris sifatida qabul qilmaydi, jumladan birlamchi kalitni aniqlanishini ham. Bu xususiyatlar foydalanuvchi tomonidan kiritilishi kerak. Yangi jadvallarni yaratish so'rovlar, ayniqsa, eski berilganlarni arxivda saqlash, berilganlarning shu dadiqadagi holatini saqlashda, ko'p jadvalli shakllar va hisobotlar bilan ishlashda ularning unumdotligini oshirishda juda foydalidir.

Faraz qilaylik, yangi RO'YHAT\_1 jadvalini RO'YHAT jadvalidagi Pasp\_S, PaspN, Familiya va Izm ustunlaridagi berilganlar asosida yaratish kerak bo'lisin. Buning uchun avval ko'rsatilgan ustunlarni RO'YHAT jadvalidan tanlab olishga so'rov tayyorlash kerak (22.39-rasm). So'ngra, «Zapros» opsiyasi orqali so'rov turini «Создание таблицы . . .» (jadvalni yaratish)ga o'zgartiramiz, natijada 22.40-rasmdagi muloqot oynasi ochiladi. Bu oyna orqali yangi jadvalining nomini kiritamiz va qaysi berilganlar bazasida saqlashni ko'rsatamiz.



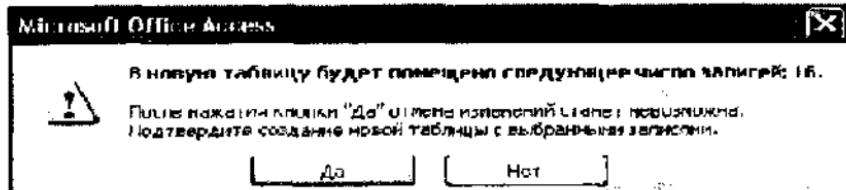
22.39-rasm.



22.40-rasm.

So'rov bajarilishini buyurganimizdan keyin, BBBB 22.41-rasmdagi kabi «yangi jadvalni yaratishni davom ettirish kerak yoki yo'qligini», aniqlash maqsadida muloqot oynasini ochadi. Agarda yangi jadval

yaratishga ruxsat etilgan bo'lsa, BBBT RO'YHAT\_1 jadvalini yaratib baza-dagi jadvallar ro'yxatiga qo'shib qo'yadi. RO'YHAT\_1 jadvalidagi berilganlar 22.41-rasmda keltirilgan. Avtomatik tarzda hosil bo'lgan SQL tilidagi ekvivalent so'rov esa 22.42-rasmda keltirilgan.



22.41-rasm.

PASPORT	NOMER	FAMILIYA	ISM
СА	5466990	Ситиков	Ситиков
СО	1234564	Файзов	Али
СА	1238/90	Эшонов	Бахром
СП	1201450	Зуинунов	Кодир
СА	2114574	Исаев	Моминур
СВ	1200233	Халиков	Темур
СН	1234567	Ахмадов	Үйерин
СА	1201451	Моинова	Чарлс
СП	1211011	Яминов	Амин
СВ	3216547	Имубабов	Васик
СА	1234567	Алиев	Абдюхан
СА	1239847	Салимов	Шокир
СА	2110101	Ташев	Ислак
СА	1770072	Халимов	Гакир
СН	1111110	Иминов	Селим
СМ	1212120	Юриев	Имин

22.42-rasm.

Запрос к запрос на создание таблицы

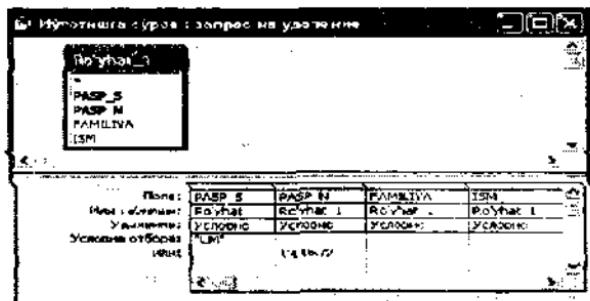
```
SELECT [Ro'yhat].PASP_S, [Ro'yhat].PASP_N, [Ro'yhat].FAMILIYA, [Ro'yhat].ISM INTO
[Ro'yhat_1]
FROM [Ro'yhat];
```

22.43-rasm.

## Yozuvlarni yo'qotishga faol so'rovlar

Yozuvlarni yo'qotishga yo'naltirilgan faol so'rovlar yozuvlar to'plamini bitta yoki bir nechta jadvallardan olib tashlashi mumkin. Bitta yo'qotish so'rovi bitta jadvaldag'i yozuvlarni yoki «birga bir» / «birga ko'p» aloqali bir nechta jadvallardagi yozuvlarni olib tashlashi mumkin. Lekin albatta yozuvlar olib tashlanganda, berilganlarning butunlik shartlari buzilmasligi kerak.

Masalar, yangi yaratilgan RO'YHAT\_1 jadvalining Pasp\_S ustunidan «SM» va PaspN ustunidan esa «1300572» qiymatlar bilan bog'langan yozuvlarni olib tashlaylik. Buning uchun avval RO'YHAT\_1 jadvalidan yozuvlarni tanlash so'rovini yaratamiz. So'ngra, «Запрос» opsiyasi orqali so'rov turini «Удаление» (yo'qotish)ga o'zgartiramiz. Hosil bo'lgan QBE to'riga 22.44-rasmidagi kabi yuqorida keltirilgan shartlarni kiritamiz va bajarishga buyruq beramiz. Bunda ham ogohlantiruvchi ma'lumot chiqadi. Agarda rozi bo'lsak, yozuvlarni olib tashlash jarayoni amalga oshiriladi. Olingan natija RO'YHAT\_1 jadvalining o'zgargan holati 22.45-rasmida keltirilgan. Avtomatik tarzda hosil bo'lgan SQL tilidagi ekvivalent so'rov esa 22.46-rasmida keltirilgan.



22.44-rasm.

PASP_S	PASP_N	FAMILIYA	ISM
1234567	1234567	Иманирова	Инур
CD	1234567	Эзизкулов	Кодир
CD	1234567	Алиева	Барна
CB	1234567	Ямнов	Алми
CA	1234567	Имубек	Болик
CA	1234567	Фархад	Ани
CA	1234567	Иманиев	Салим
CA	1234567	Монирова	Чарод
CA	1234567	Адилев	Аълоҳон
CA	1234567	Салимов	Шайх
CA	1234567	Иманиши	Нигорим
CA	2131010	Тошев	Ислон
CA	2131010	Ислакулов	Махмур
CA	5469000	Сенгиров	Сайхар

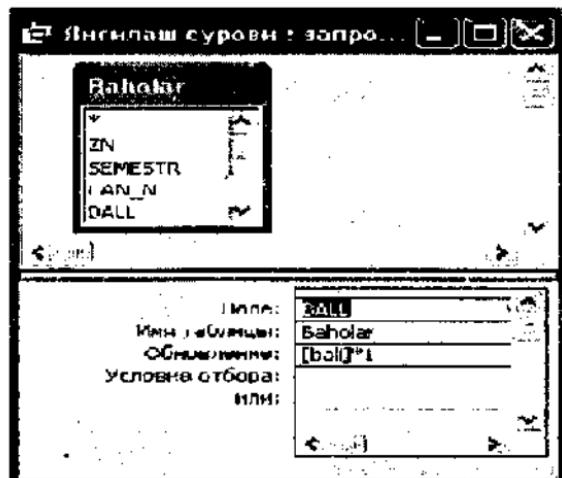
22.45-rasm.

```
DELETE [Ro'yhat_1].PASP_S, [Ro'yhat_1].PASP_N, [Ro'yhat_1].FAMILIYA, [Ro'yhat_1].ISM
FROM [Ro'yhat_1]
WHERE (([Ro'yhat_1].PASP_Si='CM')) OR ((([Ro'yhat_1].PASP_N)=1300572));
```

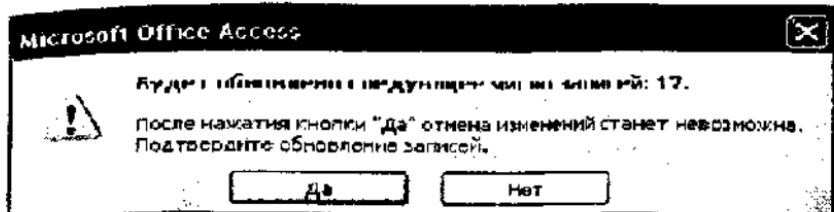
22.46-rasm.

### Yozuvlarni yangilashdagi faol so'rovlar

Berilganlarni yangilashga yo'naltirilgan faol so'rovlar bitta yoki bir nechta jadvaldagi yozuvlar to'plami ustida ishlashni ta'minlaydi. Masalan, masalaning mantiqiy tomoniga ahamiyat bermagan holda shu ustundagi berilganlarga qo'yilgan chegaralarni buzmaslik uchun quyidagi talabni faraz qilaylik, bizning o'quv loyihamizdag'i berilganlar bazasida talabalarining har bir fan bo'yicha olgan ballarini iga ko'paytirish zaruriyatini tug'ildi deylik. Buning uchun avval BAHOLAR jadvali bo'yicha berilganlarni turlashga so'rov yaratamiz. So'ngra, «Запрос» opsiyasi orqali so'rov turini «Обновление» (yangilash)ga o'zgartiramiz. Hosil bo'lgan QBE to'riga 22.47-rasmdagi kabi yuqorida keltirilgan shartlarni «Обновление» satriga **[ball]\*1** ifodasini kiritamiz va bajarishga buyruq beramiz. 22.48-rasmda keltirilgan ogohlantiruvchi ma'lumot chiqadi. Agarda rozi bo'lsak, yangilash jarayoni amalga oshiriladi. Olingan natija, BAHOLAR jadvalining o'zgargan holati 22.49-rasmda keltirilgan. Avtomatik tarzda hosil bo'lgan SQL tilidagi ekvivalent so'rov esa 22.50-rasmda keltirilgan.



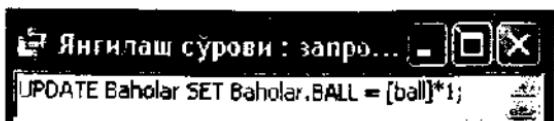
22.47-rasm.



22.48-rasm.

Зачетка №	Семестр	Фан	Группа	Сана	Учитель
E012345	2	Информатика	72	12.06.2010	Салимов С.Р.
E000000	2	Информатика	70	11.02.2010	Галиев
E076000	2	Информатика	68	01.10.2009	Абдулов Ю.Ж.
E205697	1	Акбарат ТЭХН	82	10.06.2011	Чорнос
E012345	2	Акбарат ТЭХН	87	11.02.2010	Адильев
E000000	1	Алгебра	79	18.12.2010	Олиев Г.Г.
E235813	1	Алгебра	77	13.12.2010	Судаев И.П.
E005227	8	БББТ	60	13.06.2010	Чорнос
E012305	8	БББТ	71	10.06.2011	Сакимов
E012209	8	БББТ	66	10.06.2011	Салимов

22.49-rasm.



22.50-rasm.

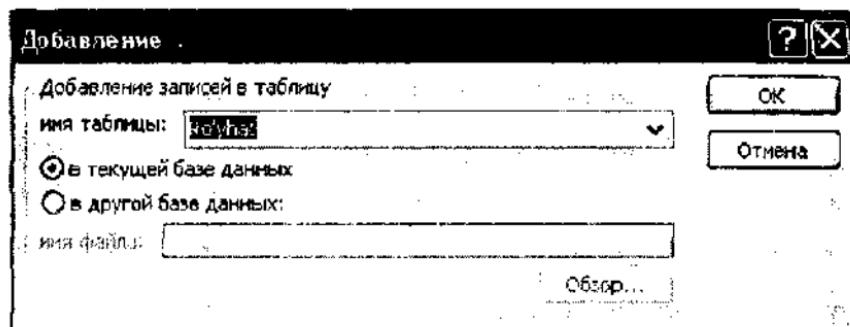
### Yozuvlarni qo'shishdagi faol so'rovlar

Yozuvlarni qo'shishga yo'naltirilgan faol so'rovlar bitta yoki bir nechta jadvallardagi yozuvlarni *bitta jadvalga qo'shish* imkoniyatini beradi. Yozuvlar, joriy yoki boshqa berilganlar bazasidagi, jadvalga qo'shiladi. Yozuvlar berilgan shartlar asosida, hattoki boshqa jadvalda ayrim maydonlar bo'limgan holda ham qo'shiladi.

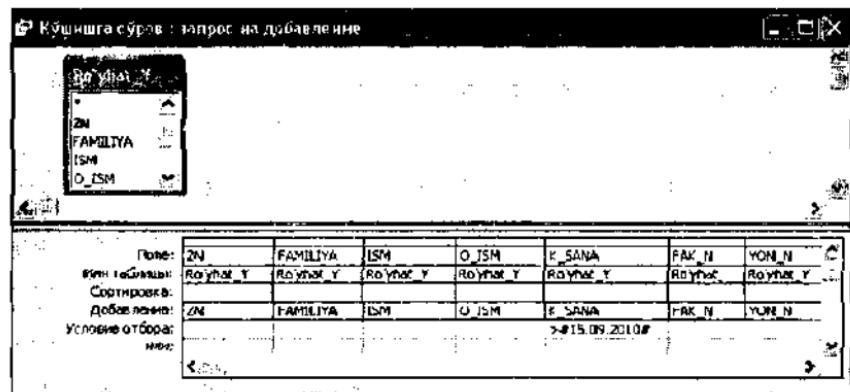
Masalan, faraz qilaylik Talabalar o'quv loyihasining berilganlar baza-sidagi RO'YHAT jadvaliga RO'YHAT\_Y jadvalidagi yozuvlarni qo'shish zarur bo'lib qoldi. RO'YHAT\_Y jadvalida faqatgina Zn, Familiya, Ism, O\_Ism, K\_Sana, Fak\_N va Yon\_N ustunlari mavjud bo'lsin. Bundan

tashqari faqatgina, qabul qilingan sanasida 2010-yil 15-sentabrdan keyingi sanadagilarni qo'shish talab qilingan bo'lsin.

Buning uchun, avval RO'YHAT\_Y jadvalidagi yozuvlarni tanlash bo'yicha so'rovni yaratamiz. So'ngra, «Запрос» opsiyasi orqali so'rov turini «Добавление» (qo'shish)ga o'zgartiramiz. Bunda 22.51-rasmidagi jadval nomi va joylashadigan berilganlar bazasini ko'rsatishni so'rangan muloqot oynasi chiqadi. 22.52-rasmda faol so'rovni bajarish uchun tayyorlangan QBE to'ri keltirilgan. Bajarishga buyruq bersak 22.53-rasmida keltirilgan ogohlantiruvchi ma'lumot chiqadi. Agarda rozi bo'lsak, yangilash jarayoni amalga oshiriladi. 22.54-rasmda RO'YHAT\_Y jadvalidagi berilganlar keltirilgan. Yangilash amalining natijasi, ya'ni o'zgargan RO'YHAT jadvali 22.55-rasmida keltirilgan. Avtomatik tarzda hosil bo'lgan SQL tilidagi ekvivalent so'rov esa 22.56-rasmida keltirilgan.



22.51-rasm.



22.52-rasm.

Microsoft Office Access

Будет добавлено 10 записей в таблицу:

После каждого ввода "Да" отмена изменений станет невозможна.  
Подтверждите добавление записей.

22.53-rasm.

Запись №	Фамилия	Имя	Старинный №	Киргиз сана	Факультет	Изменение
1	Салимбек	Санжар	Кириллическ	31.08.2010	математ	АМАТ
2	Федоров	Ситник	Алимбет	30.08.2007	география	ГЕОГРАФ
3	Зайшлов	Бахыт	Каримбек	01.10.2010	математ	АМАТ
4	Зумпалиев	Бирю	Маммуджан	31.10.2010	математ	МАТ
5	Меликумов	Магжумур	Алимбет	14.09.2010	биология	МАТЕМАТИКА
6	Жакиров	Темир	Басыкбек	31.10.2010	математ	АМАТ
7	Алиев	Бирю	Санжарбек	24.04.2010	химия	АМАТ
8	Меликов	Бирю	Хасанбек	29.08.2008	физика	ФИЗИКА

22.54-rasm.

Запись №	Фамилия	Имя	Старинный №	Киргиз сана	Факультет	Изменение	Курс	Группа	Паспорт	
1	Шипко	Балык	Кадырович	31.10.2010	математ	АМАТ	0			
2	Хадиков	Текер	Есековна	31.10.2010	химия	МАТ	0			
3	Андреев	Барис	Салымбек	29.08.2010	химия	АМАТ	0			
4	5006227	Санжар	Черекбек	31.03.2006	математ	АМАТ	4-1	СА	54689	
5	5009080	Файзулла	Алтын	30.03.2007	география	ГЕОГРАФ	4-2	СО	12345	
6	5010223	Шипко	Батыр	Салымбек	31.10.2007	математ	АМАТ	4-1	СА	12387
7	5012005	Зумпалиев	Кодир	Нажиуддин	31.03.2007	математ	АМАТ	4-1	СВ	12314
8	5012035	Исангулов	Маныз	Абубекер	30.03.2008	биология	МАТЕМАТИКА	1-1	СА	21345
9	5012789	Хадиков	Текер	Есековна	31.03.2007	математ	МАТ	4-1	СВ	12002
10	5021306	Андреев	Барис	Есековна	25.05.2006	химия	АМАТ	2-1	СВ	12345
11	5025544	Маликова	Чарис	Гильсара	29.03.2007	физика	ФИЗИКА	3-001	СА	12345
12	5052687	Яндыков	Амант	Шалымбек	30.03.2003	геология	ГЕОГРАФ	3-11	СВ	31110
13	5076000	Имубаев	Восик	Чубаков	30.03.2009	коргай физ	ФИЗИКА	4-001	СВ	31165
14	5080009	Анисов	Албеков	Есековна	30.03.2007	биология	БИОЛОГИЯ	3-001	СА	12345
15	5100254	Салимов	Эбрагим	Есековна	31.03.2007	математ	АМАТ	3-12	СА	12356
16	5111111	Чокиев	Яныч	Турдакиев	11.09.2010	физика	ФИЗИКА	1-001	СМ	12121
17	5123456	Тошев	Ислак	Еркинбек	31.03.2007	химия	МАТ	4-1	СА	21131
18	5205697	Хадиков	Бакир	Чынсызбек	31.03.2007	математ	АМАТ	2-1	СА	12005
19	5205813	Микаил	Санжар	Халимбек	13.03.2006	биология	БИОЛОГИЯ	4-012	СН	11111

22.55-rasm.

Microsoft Access... запрос на добавление

```
INSERT INTO [Rohat] (ZN, FAMILIYA, ISM_O_ISM, K_SANA, FAK_N, YON_N)
SELECT [Rohat_Y].ZN, [Rohat_Y].FAMILIYA, [Rohat_Y].ISM, [Rohat_Y].K_ISM, [Rohat_Y].K_SANA, [Rohat_Y].FAK_N,
[Rohat_Y].YON_N
FROM [Rohat_Y]
WHERE (([Rohat_Y].K_SANA)>49/(IS/2010%));
```

22.56-rasm.

RO'YHAT jadvaldag'i (22.55-rasm) qo'shilgan birinchi uchta yozuvlar, ning oxirgi to'rtta maydondagi berilganlarning bo'sh holatiga e'tiborni qarating.

### Avtomatik almashtirishli so'rovlar

Avtomatik almashtirishli so'rovlar yangi yaratilgan yozuvlarning maydonlariga avtomatik tarzda qiymatlar qo'yishni ta'minlaydi. So'rov oynasi yoki shu so'rovning natijasida hosil bo'ladigan so'rov shaklning oynasida ikkita jadvalni birlashtirish uchun kiritiladigan qiymatni MS Access BGBT avtomatik tarzda topib, foydalanuvchi kiritgan qiymatga mos qiymatni ko'r-satilgan maydonga joylashtiradi.

Masalan, RO'YHAT va FAKULTETLAR jadvallarini birlashtiruvchi Fak\_N maydonga o'zgartirish kiritilsa, bu o'zgartirishni BGBT darhol RO'YHAT jadvalidagi zarur maydonlarga avtomatik ravishda kiritadi.

Avtomatik almashtirishli so'rovlarini yaratish uchun, avval QBE to'riga orasidagi aloqa «birdan ko'pga» bo'lgan ikkita jadvalni joylashtirish kerak. Keyin esa natija to'rida joylashadigan maydonlarni ko'rsatish kerak. Fak\_N ustun (tashqi kalit)ni aloqaning ko'plik tomonida joylashgan RO'YHAT jadvalidan tanlash kerak. 22.57-rasmida avtomatik almashtirishli so'rovni bajarish uchun tayyorlangan QBE to'ri keltirilgan. 22.58-rasmida esa so'rov natijasida hosil bo'lgan to'rdagi ma'lumotlar keltirilgan. Bizning holatda, jadvaldag'i Fak\_N ustunining xususiyatlaridan biri «ro'yxatli maydon» bo'lgani uchun «Fakultet» ustunidagi har qanday o'zgarish avtomatik tarzda Fak\_Nomi ustunining o'zgarishiga olib keladi. Avtomatik tarzda hosil bo'lgan SQL tilidagi ekvivalent so'rov esa 22.59-rasmida keltirilgan.

The screenshot shows the Microsoft Access Query Builder window with the following details:

- Title Bar:** АВТОМАТИК АЛМАСТИРИШЛИ ГУРЛОТ: ЗАПРОС НА ЗАБОРНУЮ
- Left Panel (Ro'yxhat table):**
  - Fields: ZN, FAMILIYA, ISM, IO\_ISM, KLSANA, FAK\_N, YON\_N.
  - Relationships: A line connects ZN from Ro'yxhat to FAK\_N in Fakultet.
- Right Panel (Fakultet table):**
  - Fields: FAK\_N, FAK\_NOMI.
- Bottom Panel (Query Results):**
  - Fields:** ZN, FAMILIYA, ISM, FAK\_N, FAK\_NOMI.
  - Filter:** Имя таблицы: Сортировка: Овощи на складе Установлено: Быстро: Кнопки: Помощь
  - Table Structure:**

ZN	FAMILIYA	ISM	FAK_N	FAK_NOMI
Ro'yxhat	Ro'yxhat	Ro'yxhat	Ro'yxhat	Fakultet

22.57-rasm.

Автоматик алмаштиришلى сұров : запрос на ...

	Фамилия	Исм	Факультет	FAK_NOMI
	Халилов	Темур	МОЛІМӘТ	МОЛІМӘТ
	Селимов	Шоқир	МЕХМЕТ	МЕХМЕТ
	Хошимов	Цайир	МЕХМЕТ	МЕХМЕТ
7	Жанисбек	Талгүр	МОЛІМӘТ	МОЛІМӘТ
	Мынисов	Нарс	ФИЗИКА	ФИЗИКА
	Чарин	Ямин	ФИЗИКА	ФИЗИКА
	Эшінбаев	Бахром	ФИЗИКА	ФИЗИКА
	Алиева	Барин	ХИМИЯ	ХИМИЯ
	Төшев	Исек	ХИМИЯ	ХИМИЯ
	Диана	Царна	ХИМИЯ	ХИМИЯ
	Мынисбек	Магжуме	БИОЛОГИЯ	БИОЛОГИЯ

Запрос: 14 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |

22.58-rasm.

Автоматик алмаштиришلى сұров : запрос на ...

```
SELECT [Ro'yhat].ZN, [Ro'yhat].FAMILIYA, [Ro'yhat].ISM, [Ro'yhat].FAK_N,
Fakultetlar.FAK_NOMI
FROM Fakultetlar INNER JOIN [Ro'yhat] ON Fakultetlar.FAK_N =
[Ro'yhat].FAK_N;
```

22.59-rasm.

### Xulosa

Biz QBE namuna bo'yicha so'rov tilining funksional imkoniyatlaridan foydalandik, alohida xususiyatlari bilan tanishdik. QBE tilida so'rovlar andazasini qo'llashga asoslangan berilganlar bazasiga kirishni tashkil-lasshtirishning namoyishli usulidan foydalandik. QBE tilidan foydalanib, QBE to'rida so'rovlar namunalarini bergen holda, berilganlar bazasiga shu zahoti kirib, javob olish va namoyishli tarzda natijani tasvirlash imkoniyatini o'rgandik.

So'rovlarning turlari bo'yicha – tanlov, ko'p jadvalli, tanlov alomatlari berish, umumlashtiruvchi, parametrik, kesishgan, nusxalarini tashlash, o'ziga mos bo'lgagan yozuvlarni tashlash, jadvallarni yaratish, yozuvlarni yo'qotish, yozuvlarni yangilash, yozuvlarni qo'shish, avtomatik almash-tirishli so'rovlar yaratishdagい QBE vositalari va/yoki QBE to'ridan foydalangan holda, misoltar yordamida o'zlashtirdik.

## **Nazorat uchun savollar va amaliyot uchun topshiriqlar**

1. QBE tilidan nima maqsadda foydalaniladi?
2. QBE to'ri nima uchun ishlataladi?
3. So'rov «usta»si nima ish bajaradi?
4. So'rov konstruktori orqali qanday so'rovlarni amalga oshirish mumkin?
5. Talabalar o'quv loyihasi berilganlar bazasidagi hamma jadvallarni yarating.
6. Atributlar sonini qo'shish mumkin bo'lgan jadvallarga yangi atributlarni kiriting. Masalan, jinsi, tug'ilgan sanasi va h.k.
7. Jadvallar orasidagi aloqalarni 7- bobda keltirilgan chizma bo'yicha amalga oshiring.
8. Yaratilgan jadvallarga o'zingiz tahlil olayotgan Oliy o'quv yurtining 3ta yoki 4ta fakultetidagi har bir yo'nalish bo'yicha 2-3 ta talabalar xususidagi berilganlarni kiriting.
9. BAHOLAR jadvalini har bir yo'nalishdagi talabalarning 8ta semestrda o'tiladigan fanlar bo'yicha ixtiyoriy ballardagi qiymatlarini qo'yib chiqing.
10. Ushbu bobda keltirilgan hamma so'rov turlaridagi misollarni MS SQL Server yoki MS Access BBBT orqali amalga oshiring.
11. Yaratilgan jadvallar asosida QBE tilining vositalaridan foydalangan holda quyidagi tanlov so'rovlari mashqlarini amalga oshiring.
12. Kiritilgan fakultettlardagi har bir yo'nalishlar bo'yicha talabalarning ro'yxatini tanlang.
13. Fanlar bo'yicha olgan ballari [56,70] oraliq'ida yotuvchi talabalarning reyting daftarini aniqlang.
14. Kiritilgan aniq bir fan bo'yicha «yaxshi» va «a'llo» baho olgan talabalar ro'yxatini aniqlang.
15. Familiyasi «T» harfidan boshlanadigan ayol talabalarning ro'yxatini aniqlang.
16. Yoshi [17,23] oraliqda bo'lgan erkak talabalarning ro'yxatini aniqlang.
17. Pasportining seriyasi SA bo'lgan hamma talabalarning to'liq ma'lumotini chiqaring.
18. Ko'p jadvalli ixtiyoriy so'rov yarating.
19. Yig'indini hisoblash bo'yicha umumlashtiruvchi so'rov yarating.
20. O'rta qiymatni hisoblash bo'yicha umumlashtiruvchi so'rov yarating.
21. Maksimal qiymatni aniqlash va minimal qiymatni aniqlash bo'yicha umumlashtiruvchi so'rov yarating.
22. Nusxalarni sanash bo'yicha umumlashtiruvchi so'rovni yarating.
23. Ixtiyoriy parametrik so'rov yarating.

24. Ixtiyoriy kesishgan so'rov yarating.
25. Ixtiyoriy nusxalarni tanlashga so'rov yarating.
26. Ixtiyoriy o'ziga mos bo'limgan yozuvlarni tanlashga so'rov yarating.
27. Jadvalni yaratishga ixtiyoriy faol so'rov yarating.
28. Yozuvlarni yo'qotishga ixtiyoriy faol so'rov yarating.
29. Yozuvlarni yangilashga ixtiyoriy faol so'rov yarating.
30. Yozuvlarni qo'shishga ixtiyoriy faol so'rov yarating.
31. Ixtiyoriy avtomatik almashtirishli so'rov yarating.

## 23- BOB. SQL TILIGA KIRISH

**Tayanch iboralar:** *SQL tili, SQL DML tili, SQL DDL tili, so'rov, SELECT operatori, from bo'lim, where bo'limi, order by bo'limi, group by bo'limi, having bo'limi, so'rov ifodasi, so'rov qismi.*

Berilganlarni relationsion modelini ishlab chiqish natijasida SQL tili yaratilib, relationsion berilganlar bazasini namuna tiliga aylandi.

Bu tilni asosida relationsion algebra va relationsion hisob turibdi. 1987-yilda Standartlar bo'yicha xalqaro tashkiloti (ISO) tomonidan SQL tilining namunasini qabul qilindi.

### SQL tilining vazifasi

BB bilan ishlaydigan har qanday til quyidagi imkoniyatlarga ega bo'lishi kerak:

1. Berilganlar jadvalini, ularning bazaviy tuzimini yaratish va tavsiflash.
2. Berilganlar ustida ishlov amallarini bajarish, ya'ni jadvalga berilganlarni qo'shish, o'zgartirish va olib tashlash.
3. Oddiy va murakkab so'rovlarni bajarish.

Ko'rsatilgan imkoniyatlar SQL tilida amalga oshirilgan.

Jadvallardagi berilganlarni talab qilingan chiquvchi ko'rinishga o'tkazish uchun SQL tili ishlatiladi.

SQL tili ikkita asosiy tarkiblarga ega:

- DDL (Data Destination Language) tili, ya'ni berilganlarni tavsiflash tili. Bu til berilganlar bazasini tuzilishini tavsiflash va ularga murojaat qilishni boshqarish imkoniyatlarini yaratadi.
- DML (Data Manipulation Language) tili, ya'ni berilganlarga ishlov. Bu til berilganlarni yangilash va tanlash uchun ishlatiladi.

Bu tilda hisoblash jarayonini boshqarish komandalari, ya'ni IF ... THEN... ELSE, DO ... WHILE, GOTO kabi komandalar yo'q, shu sababli bu tilni ikkita usul bilan ishlatish mumkin, ya'ni muloqot (foydanuvchi bevosita SQL operatorlarini ketma-ket kiritadi) usuli yoki SQL tiliga ega bo'lgan yuqori bosqich tili orqali.

1999-yilda kiritilgan SQL3 [46,65] tili namunasida yuqorida ko'rsatilgan imkoniyatlar kiritilgan.

SQL tili noprotsedur til bo'lgani sababli, unda berilganlarni qaysi yo'l bilan olish emas, balki qanday ma'lumot kerakligini ko'rsatish kerak. SQL tilida operatorlar erkin formatda yoziladi, ya'ni fiksirlangan pozitsiyaga bog'lanmagan holda. SQL tilida operatorlar inglez tilida yozilgan kalit so'zlar bilan belgilanadi, masalan:

```
CREATE TABLE talaba(id_talaba int,
    fam varchar(20), ismi varchar(10),
    stipendiya decimal(9,2));
INSERT INTO talaba
VALUES('1','Boltayev','Boltabay', 150000);
SELECT fam FROM talaba WHERE stipendiya>100000;
```

SQL tilini administratoridan boshlab, oddiy foydalanuvchilargacha bar-cha ishlatishi mumkin.

### SQL tilining tarixi

SQL til va relatsion modelning tarixi 1970-yildan E.F. Kodd [17] tomonidan nashr etilgan maqoladan boshlangan.

- 1974 y. - D.D. Chemberlin, P. Reyner, R.F. Boys, F.B. Raymondlar SEQUEL (Structured English Quary Language) [8] va SQUARE [42] tillarning tavsifini chiq etadilar. Bu tillar SQL tilining dastlabki namunalari hisoblanadi.
- 1976 y. - Qayta ishlangan SEQUEL/2 [6] tili chiqariladi, keyinchalik til nomi SQL bo'ldi.
- 1976y. - Shu til asosida IBM kompaniyasi «System R» [3] nomidagi BBBT chiqaradi.
- 1979 y. - ORACLE kompaniyasi SQL tili asosida BBBT «ORACLE» yaratadi.
- 1981 y. - DOS tizimi uchun IBM birinchi relatsion BBBT «SQL/DS» chiqariladi. Bu tizimni shahsiy kompyuterlarda ishlatsa bo'ldi.
- 1982 y. - AQSH standartlar milliy instituti (ANSI) RDL (Relation Data Lenguage) tili bo'yicha tadqiqot ishlarni boshlagan.
- 1983 y. - IBM Shularni o'zi yangi BBBT «DATA BASE II» chiqaradi.
- 1987 y. - ISO tomonidan SQL tilini birinchi namunasi nashr etilgan.
- 1989 y. - ISO SQL ga berilganlarni butunligini saqlash funksiyalarini qo'shimcha qilib kiritadi va bu namuna SQL/89 deb nomlanadi.

1992 y. – Qayta ishlab chiqarilgan namunani ikkinchi naqli SQL/2 (SQL-92) chiqariladi.

1999 y. – Na'munani uchinchi naqli SQL3 [21,64] chiqariladi. Bu namunada berilganlarni boshqarishda obyektga yo'naltirilgan funksiyalar qo'shiladi.

Bundan tashqari tijorat berilganlar bazasini yaratuvchilar tomonidan qo'shilgan funksiyalar tilni kengaytmasiga kiritilgan.

Yuqorida keltirilgan amalga oshirilgan har bir til SQL tilini naqli deb nomlanadi.

Hozirgi vaqtida 100dan ortiq dialektlar mavjud, shu jumlaqan:

- SQL – Access [70,86] – BGBT Access tili;
- Tranzact - SQL – BGBT MS SQL Server tili [72,91,92];
- MySQL – BGBT Internet tili [38,39];
- PL / SQL – BGBT ORACLE tili [68,69].

Bu tillar namuna tilga o'zgartirishlar kiritgan va uni ancha **kengaytir-gan**.

### Ishlatilgan atamalar

SQL tilida **relatsion** modeliga nisbatan quyidagi muqobil atamalar ishlataladi:

<i>Relatsion modelda</i>	<i>SQL tilida</i>
munosabat	jadval
kortej	satr
atribut	ustun (mavdon)

Undan tashqari SQL tilida tanlash natijasida **hosil bo'lgan jadvalda bir xildagi takrorlanuvchi satrlar bo'lishi mumkin**, satrlari ni tartiblash imkoniyati bor, ustunlarni tartibi fiksirlangan.

### SQL operatorlarni yozish qoidalari

SQL tilida operatorlar erkin formatda yoziladi.

Har bir operator kalit so'zlar va foydalanuvchi tomonqan kiritilgan so'zlardan iboratdir. Kalit so'zлarni bo'lib yozish mumkin emas, ya'ni ularni namunada ko'rsatilgan bo'yicha yozish kerak. Foydaluvchilar tomondan kiritilgan so'zlar jadvallar, ustunlar, so'rovlarini va berilgan bazasi-dagi boshqa obyektlarni nomlash uchun ishlataladi. So'zлarni ketma-ketligi qoidalari bo'yicha beriladi, operator nuqta-vergul (;) bilan yaku'llanadi.

Barcha operatorlar registrga bog'lanmagan, ya'ni katta va kichik harflarni ishlatsa bo'ladi, lekin literalda bu ahamiyatga ega. Masalan, «Boltayev» va «BOLTAYEV» har xil literallardir.

Til erkin formatda yozilishi mumkin bo'lgan uchun, quyidagi kelishuvlarga tayanamiz:

1. Har bir konstruksiya yangi satrдан boshlanishi kerak.
2. Har bir konstruksiya bir xil satrni bir xil pozitsiyasidan boshlanishi kerak.
3. Agar konstruksiya bir nechta qismidan iborat bolsa, qismlar yangi satrдан boshlanib biroz o'nga surilib yozilishi kerak.

Misol :

```
SELECT fam, ismi, sharifi, jinsi, tug_il,
shaharlar.nameshahar
FROM talaba INNER JOIN shahar
    ON talaba.id_shah=shahar.id_shah
ORDER BY fam;
```

Til konstruksiyalar sintaksisini tavsiflash uchun biz Kengaytirilgan Bekus-Naur shakli (KBNSh) metatilini ishlatamiz [59]. Bu tilda quyidagi metabelgilar kiritilgan (meta bu boshqa tilni tavsiflash uchun ishlataladigan so'z):

- 1) ::= meta belgisi chap tarafda turgan meta so'zni o'ng tarafda turgan tavsifdan ajratish uchun ishlataladi, ya'ni meta so'z bu o'ngda turgan konstruksiyadan iborat.
- 2) Katta harflar kait so'zlar uchun ishlataladi.
- 3) Kichik harflar foydalanuvchi kiritgan so'zlar uchun ishlataladi.
- 4) Vertikal belgisi | bir nechta variantlardan birini tanlash kerakligini bildiradi (a | b | c).
- 5) Figurali qavslar { } ichidagi element majburiyligini bildiradi.
- 6) Kvadrat qavslar [ ] ichidagi element shart emasligini bildiradi.
- 7) Kvadrat qavslardagi vergul va 3 nuqta belgisi [,...] bundan oldin turgan element nol, bir yoki bir necha marta qaytarilishi mumkinligini bildiradi.
- 8) Burchak qavslari < > konstruksiya elementlarini belgilash (meta so'zlar) uchun ishlataladi.
- 9) Probel belgisi operatorlar shaklini qulay ko'rsatish uchun ishlataladi.
- 10) Vergul belgisi elementni ro'yxatda berish uchun ishlataladi.
- 11) Nuqta-vergul belgisi operator tamom bo'lishini bildiradi.

Berilganlar bazasi turkumini (jadvallar va bog'liqliklar) tafsiflash uchun DDL tili ishlataladi, bazani to'ldirish va undan ma'lumot olish uchun DML tili qo'llanadi. Lekin barcha BBTlarda jadvallarni yaratish, tafsiflash va o'zaro bog'lash uchun «master» (usta) deb nomlangan maxsus programma vositasi bordir, shu sababli DDL tili amalda kamdan-kam ishlataladi. Shuning uchun biz oldin DML tilini o'rganib chiqamiz.

### Berilganlar ustida qayta ishslash

Berilganlar ustidan ishlov uchun quyidagi to'rtta operator mavjud:

**SELECT** – berilganlar bazasidan ma'lumotlarni tanlash;

**INSERT** – jadvalga berilganlarni qoshish;

**UPDATE** – jadvaldagi berilganlarni yangilash;

**DELETE** – jadvaldagi berilganlarni o'chirish.

Bu operatorlar jadvallar, tasavvurlar va berilganlar bilan ishlaydi. Aniq berilganlar, ya'ni o'zgarmaslar literallar deb nomlanadi. Literallar ustun qiymatini berishda, shartli ifodalarda (predikatlar) va boshqa turli joylarda ishlataladi. SQL/89 tilida quyidagi berilganlar turlarini ishlatish mumkin (qisqa ma'lumot):

**CHARACTER** – satrli berilganlar uchun, satr o'zgarmas qo'shtimoq belgilari ichiga olinadi va satrli litera deb nomlanadi.

**NUMERIC**, **DECIMAL**, **INTEGER**, **SMALLINT** turlar aniq fiksirlangan sonlar uchun ishlataladi, masalan **NUMERIC[(<razryad\_soni>[,<masshtabi>])]** sonni aniqligini va masshtabini beradi. **DECIMAL** turi **NUMERIC** turiga ekvivalent, **INTEGER** va **SMALLINT** turlar butun sonlar uchun ishlataladi. Suzuvchi nuqtali haqiqiy sonlarni tasvirlash uchun **FLOAT**, **REAL** va **DOUBLE PRECISION** turlar kiritilgan, bu turdag'i literada darajani ifodalovchi «E» harfi ishlatalishi mumkin.

SQL/92 da namunaga yana bir nechta turlar kiritilgan bu DATE, TIME, INTERVAL, TIMESTEP, MONEY bu turlar sana, vaqt, oraliq va pul qiymatlariga bog'liq.

Barcha o'zgarmastalar literal qiymatiga kiradi, uning sintaktik ko'rinishi quyidagicha:

```
<literal qiymati> ::= <son> | <satr> | <sana> | <vaqt>
<son> ::= [+|-]{<raqam>...[.<raqam>...]}.<raqam>...}[{e|E][+|-]
<raqam>...
<satr> ::= '{<belgi>|''}...
<sana> ::= DATE 'yyyy-mm-dd'
<vaqt> ::= TIME 'hh:mm:ss[.soniya qismi]'
```

Jadval ustuniga murojaat qilish uchun quyidagi sintaksis ishlataladi:

*<ustun tasnif>*::= [{<jadval nomi> | <jadval taxallusi>}..] <ustun nomi>

Bunda jadval taxallusi jadvalga berilgan qo'shimcha sinonim nomi.

Misol:

```
INSERT INTO talaba (id_tal, fam, ismi, tug_il,
                     shahar, reyt_bal)
VALUES
(100, 'Boltayev', 'Bolta', 1960, 'Toshkent', 78.12);
```

### Tanlash operatori

Operatorlardan eng ko'p qo'llanadigan bu SELECT operatoridir. Buning yordamida yaratilgan so'rovlar orqali berilgan jadvallardan kerakli ma'lumotlarni ajratib olish mumkin. Bu operator barcha relatsion amallarni bajarishi mumkin, undan tashqari relatsion hisobga o'xshab unda faqat natijaning ko'rinishi yoziladi. Uning namunada berilgan umumiy sintaksisi quyidagicha:

*<tanlash operatori>*::= <so'rov ifoda> [<order by bo'limi>]

*<so'rov ifoda>*::= <yagona so'rov> | <so'rov ifoda>

{UNION|INTERSECT|EXCEPT} [ALL] <yagona so'rov>

*<yagona so'rov>*::= <so'rov tasnifi>

*<so'rov tasnifi>*::= <select ifodasi> <jadval ifodasi>

*<o'tkazish operatori>*::= <select ifodasi> INTO {<o'zgaruvchilar ro'yxati> | (<jadval tasnifi>)} <jadval ifodasi>

*<so'rov qismi>*::= (<so'rov ifoda>)

*<select ifodasi>*::= SELECT [ALL | DISTINCT] <select ro'yxati>

*<select ro'yxati>*::=<select elementi> [, ...]

*<select elementi>*::=\*<jadval nomi>.\*<jadval taxallusi>.\*|<skalyar ifoda> [AS <ustun psevdonimi>]

*<jadval ifodasi>*::= <from bo'limi> [<where bo'limi>] [<group by bo'limi>] [<having bo'limi>]

*<skalyar ifoda>*::= [+ | -] <skalyar elementi> [{ { + | - | \* | / | || } <skalyar elementi> }...]

*<skalyar elementi>*::=<literal qiymati> | <ustun tasnifi> | <funksiya> | <so'rov qismi> | (<skalyar ifoda>)

*<funksiya>*::=<CAST funksiya>|<agregat funksiya>

*<agregat funksiya>*::=COUNT(\*)|[SUM|AVG|MIN|MAX|COUNT] [ALL|DISTINCT] <skalyar ifoda>)

**<CAST funksiyasi>:=CAST({<literal qiymati>}|<ustun tasnifi>} AS <berilganlar turi>)**

**SELECT** kalit so'z bilan uchta sintaktik konstruksiyalarni tavsiflash mumkin: kursor tasnifi, tanlash operatori va so'rov qismi. Ular asosida <jadval ifodasi> sintaktik konstruksiya turibdi. Jadval ifodani semantikasi shundan iboratki, from, where, group by va having bo'limlarni ketma-ket qo'llash hisobiga, from bo'limda ko'rsatilgan jadvallardan yangi jadval hosil bo'ladi. Bunday jadvalni satrlari tartiblangan bo'lishi va unda takrorlangan satrlar ham uchrashi mumkin. Biz ketma-ket sintaktik konstruksiyalarni ko'rib chiqamiz.

Tanlash operatori bitta yoki bir nechta jadvaldan kerakli ma'lumotlarni natijaviy jadval ko'rinishda hosil qiladi. Bunday jadvalga hech qanday chegaralar qo'yilmaydi, bunday jadval bitta ustun bitta satr (skalyar), yoki bitta satr bir nechta ustun, yoki bo'sh ham bo'lishi mumkin. Natijaviy jadval vaqtincha quriladi va operator ishlagandan so'ng unuman aytganda o'chirilib ketadi, lekin namunaga kursor tushunchasi ham kiritilgan. Kursor jadvalni joriy satrini ko'rsatib turadigan ko'rsatkich, kursor orqali jadvalni satrlari bo'yicha navigatsiya, ya'ni har tomoniga yurish mumkin, buning uchun kursor avvalo natijaviy jadvalga bog'lanib qo'yiladi, so'ng ma'lum komandalar bilan shu jadval bo'ylab yuradi.

Oxirida kursor yopilishi kerak bo'ladi, shunda natijaviy jadval ham o'chiriladi.

### **So'rov tasnifi**

So'rov tasnifida tanlash ro'yxati beriladi bu ro'yxatda jadvallar ustunlari nomidan tashqari ulardan tuzilgan arifmetik ifodalar, satrlar va o'zgarmaslar berilishi mumkin. Natijaviy jadvalda shu ro'yxatda ko'rsatilgan ustunlar bo'ladi. So'rov tasnifi **ALL** yoki **DISTINCT** kalit so'zları uchrashi mumkin. Agar so'rov tasnifida kalit so'zi uchrasa, natijaviy jadvaldan bir xil satrlar, ya'ni takrorlanuvchi satrlar olib tashlanadi, aksincha **ALL** kalit so'zi ishlatisa yoki ikkalasi ham yozilmasa satr nusxalari saqlanib qoladi.

### **So'rovlar ifodasi**

So'rov ifodasi bu so'rov tasnifi asosida sintaktik qoidalarga ko'ra tuzilgan ifodadir. Bu ifodada faqat **UNION**, **INTERSECT**, **EXCEPT** (jadvallarni birlashtirishi, kesmasi va ayirmasi) yoki **UNION ALL** amali, bor. Bu amallarning jadval operandlariga ustunlar soni teng bo'lishligi va mos ustunlar turi bir xil bo'lishi sharti qo'yiladi. So'rov ifodasi chapdan o'nga

qarab hisoblanadi va qavslar hisobga olinadi. **UNION** amali bajarish natijasida birlashtirilgan jadval hosil bo'ladi va unda bir xil satrlar olib tashlanadi. **UNION ALL** amal natijasida, hosil bo'lgan jadvalda satr-nusxalar saqlanib qoladi. Xuddi shunday **INTERSECT** amali jadvallar kesmasini hosil qiladi va **EXCEPT** jadvallar ayirmasini hosil qiladi, bu amallar relation algebra amalini to'g'ridan-to'g'ri bajaradi .

### *Order by bo'lim*

Order by bo'limi natijaviy jadvalni kerakli tartibda ko'rsatish uchun ishlataladi, uning sintaksisi quyidagicha:

```
<order by bo'limi> ::= ORDER BY <tartib tasnifi> [, ...]
<tartib tasnifi> ::= {<tartib raqami> | <ustun tasnifi>} [ASC | DESC]
```

Bu sintaktik qoidalardan ko'rinish turibdiki, **ORDER BY** ro'yxatida so'rov ifodaru natijaviy ustunlari beriladi va har bir ustun uchun tartiblash qoidasi beriladi (**ASC** – o'sish (kelishuv bo'yicha), **DESC** – kamayish bo'yicha). Ustunlar o'z nomi bilan berish mumkin agarda:

- 1) so'rov ifodada **UNION**, **INTERSECT**, **EXCEPT** amallar uchramasa,
- 2) ustunda skalyar ifoda ishlatalmasa, aks hollarda ustunni tartib raqami beriladi.

### *Bir satrni tanlash operatori*

Bir satrni tanlash operatori SQL/92 tilining alohida operatoridir. Bu operator so'rov natijasini to'g'ridan-to'g'ri taqbiqiy programma kursorsiz uzatish imkoniyatini beradi. Shu sababli bu operatordan foydalanish natijasidagi jadvalda satrlar soni bittadan ortmasligi kerak. Operatorda **INTO** bo'limida taqbiqiy programma o'zgaruvchilari ro'yxati beriladi.

### *So'rov qismi*

Jadval ifodaga ega bo'lgan konstruksiya bu so'rov qismidir, ya'ni SQL operatorning tanlash shartiga (predikatga) kiradigan so'rovdir. SQL/92da so'rov qismi natijasidagi jadval faqat bitta ustunga ega bo'lish kerak. Shu sababli natija tasnifi, qiymatni hisoblovchi arifmetik ifodadan iborat bo'lishi shart. Shuni aytib o'tish kerakki, so'rov qismilar boshqa so'rovga kirganligi

uchun, unda o'zgarmaslar sifatida tashqi so'rovni ustunlar qiymati ishlash tilishi mumkin. Bu to'g'risida keyinroq batafsil to'xtab o'tamiz.

### Jadval ifodasi

SQL/92 namunasida jadval ifodasi from, where, group by va having bo'lmlari ketma-ket from bo'limga ko'rsatilgan jadvallar ro'yxatiga qo'llash natijasida hosil bo'ladi.

#### *From bo'limi*

From bo'limi quyidagi sintaksisiga ega:

```
<from bo'limi> ::= FROM <jadval tasnifi> [, ...]  
<jadval tasnifi> ::= {<jadval nomi> | <so'rov qismi>} [[AS]  
<jadval taxallusi>] [(<ustun nomi> [, ...])] [<ulangan jadvallar>]
```

From bo'limi bajarilish natijasida ro'yxatda berilgan jadvallarning «kengaytirilgan» dekai ko'paytmasi hosil bo'ladi. «Kengaytirilgan» atamani ma'nosi shundaki, natijaviy jadvalda bir xil satrlar bo'lishi mumkin.

Sintaksisdan ko'rinish turibdiki, jadval nomidan keyin qo'shimcha ism qo'yish mumkin va operatori boshqa bo'lmlarida shu ism orqali jadval ustunlariga murojaat qilish mumkin. Bu ancha qulayliklar yaratadi, birinchidan sinonim qisqa bo'lishi mumkin, ikkinchidan biror bir jadval ko'p marja ro'yxatda uchvashni mumkin, bu holda sinonim albattra ishlatalishi kerak.

Jadval ifodasida from bo'limi majburiydir, agarda u yagona bo'lsa, ifodani natijasi shu bo'lim natijasiga teng bo'ladi.

#### *Where bo'limi*

Agar jadval ifodasida where bo'lim mavjud bo'lsa, u keyingi navbatda hisoblanadi. Where bo'limning sintaksisi quyidagicha:

```
<where bo'lim> ::= WHERE <qidirish sharti>  
<qidirish sharti> ::= [NOT] {<predikat>|(<qidirish sharti>)} [{[OR |  
AND]} { [NOT] {<predikat>|(<qidirish sharti>)} } ...]  
<predikat> ::= <solishtirish predikati> | <interval predikati> |  
<mansublik predikati> | <qolip predikati> | <>null predikati> | <kvantor  
predikati> | <mavjudlik predikati>
```

Where bo'limini hisoblanishi quyidagicha bajariladi: faraz qilaylik R - **FROM** bo'limini hisoblash natijasi bo'lsin. U holda, tanlash sharti R

jadvalni barcha satrlariga qo'llanadi, where bo'limi natijasida R jadvalga tanlash shartini qo'llashda predikat qiymati **TRUE** ga teng bo'lган satrlar qoladi. Agar tanlash shartiga so'rov qismlari kirsa, u holda har bir satr uchun so'rov qismi qayta hisoblanib shart tekshiriladi, lekin aslida hisoblash usullari reatl BBBTda boshqacha bo'lishi mumkin.

SQL/92 namunada berilgan bazasida noma'lum qiymatlar uchrashi mumkin, ya'ni Null-qiyamat, bu qiymatni 0 bilan yoki bo'sh satr bilan adashtirmaslik kerak, Null-qiyamatga ega bo'lган har qanday arifmetik ifodani qiymati **NULL**ga teng, agar satrlar ulamasida (kontenatsiyasida) Null-qiyamat uchrasa, bunday ulamani qiymati **NULL** bo'ladi, xuddi shunday solishtirish amallarni natijasi ham **NULL**ga teng. Tanlash shartini hisoblashda uch qiymatlik mantiq (**TRUE**, **FALSE**, **NULL**) ishlataladi, ya'ni bu amallarning natijasi quyidagicha bo'ladi:

<b>TRUE AND NULL = NULL;</b>	<b>FALSE AND NULL = FALSE;</b>
<b>NULL AND TRUE = NULL;</b>	<b>NULL AND FALSE = FALSE;</b>
<b>NULL AND NULL = NULL;</b>	
<b>TRUE OR NULL = TRUE;</b>	<b>FALSE OR NULL = NULL;</b>
<b>NULL OR TRUE = TRUE;</b>	<b>NULL OR FALSE = NULL;</b>
<b>NULL OR NULL = NULL;</b>	
<b>NOT NULL = NULL</b>	

### *Group By bo'limi*

Agar jadval ifodasida group by bo'limi mavjud bo'lsa, u navbat bo'yicha hisoblanadi. Group by bo'limi sintaksisi quyidagicha:

*<group by bo'limi> ::= GROUP BY <ustun tasnifi> [, ...]*

Agar oldingi bo'limilar natijasini R deb olsak, u holda GROUP BY ishlashi natijasida R jadvali ko'rsatilgan ustunlar bo'yicha guruhlarga bo'linadi, har bir guruhda mos ustunni qiymati bir xil bo'ladi. Bunday jadval guruhlangan jadval deb nomlanadi.

### *Having bo'limi*

Having bo'limi, agar mavjud bo'lsa, oxirida hisoblanadi. Uning sintaksisi quyidagicha:

*<having bo'limi> ::= HAVING <qidirish sharti>*

Bu bo'lim jadval ifodasida faqatgina group by bo'limi mavjud bo'lsiga qatnashadi. Bu bo'lim guruhlangan jadvaldag'i satrlarni solishtirib tan-

laydi. Tanlash sharti where bo'limi shartiga o'xshaydi. Lekin ba'zi bir chegaralar ham bor. Bu chegaralarning, sababi shundaki, unda qo'yilgan shartlar alohida satrga emas, butun guruuhga tegishli bo'ladi. Shartli ifodada faqat group by bo'limda ko'rsatilgan ustunlar qatnashishi mumkin, qolgan ustunlar faqat agregat funksiyalar parametrlarida qatnashishi mumkin, bu funksiyalar ko'rsatilgan guruh bo'yicha hisoblanadi va shart ifodada qatnashishi mumkin. Xuddi shunday predikatga kirgan so'rov qismi faqat guruh ustunlarga murojaat qilishi mumkin. Having bo'limi bajarish natijasida shartni qanoatlantirgan guruhlangan jadval bo'ladi.

### Xulosa

- SQL noprotsedur til bo'lib, ingliz so'zlardan iboratdir. Uni oddiy foydalanuvchilar va professionallar ham ishlatsi mumkin. Bu til relatson berilganlar bazasini tavsiflashda va ishlovda namuna bo'lib qolgan.
- SELECT operatori SQL tilini eng muhim operatori bo'lib, so'rovni yaratishda ishlataladi.
- SELECT operatorning tanlash ro'yxatida ustunlar nomi va/yoki hisoblanuvchi maydonlar ko'rsatilishi mumkin bu ustunlar natijaviy jadvalda aks ettiriladi.
- FROM bo'limida tanlash ro'yxatiga tushgan ustunlar jadvali yoki tasavvuri ko'rsatilishi kerak.
- WHERE konstruksiyasi natijaviy jadvalgan tushadigan satrlarni tanlash uchun ishlataladi. Tanlash qo'yilgan shartni tekshirish orqali bajariladi, agar tekshirish natijasi rost bo'lsa satr qo'shiladi , aks holda qo'shilmaydi.
- ORDER BY konstruksiyasi natijaviy jadvalni satrlarini tartiblash uchun qo'llaniladi.

### Nazorat uchun savollar

1. SQL tilning ikki asosiy tarkiblarini ayting. Qanday ishlarni ular bajaradi?
2. SQL tilning afzalligi va kamchiligi nimadan iborat?
3. Tanlash operatorning har bir konstruksiyani vazifasini tushuntirib bering, konstruksiyalarga qanday chegaralar qo'yilgan?

## 24-bob. TANLASH SHARTLARI

**Tayanch iboralar:** solishtirish predikati, interval predikati, mansublik predikati, qolip predikati, null predikati, kvantor predikati, mavjudlik predikati.

SQL/92 namunasi bo'yicha tanlash shartida quyidagi predikatlar qo'l-lanishi mumkin: solishtirish predikati, between predikati, in predikati, like predikati, null predikati, kuantor predikati va exists predikati.

### *Solishtirish predikati*

Solishtirish predikatining sintaksisi quyidagicha:

<solishtirish predikati> ::= <skalyar ifoda> <solishtirish amali>  
<skalyar ifoda>  
<solishtirish amali> ::= <> | = | < | > | <= | >=

Bu erda «< >» tengsizlikni bildiradi. Solishtirish amalining o'ng va chap tomondagи arifmetik ifodalar umumiy qoidalar bo'yicha hisoblanadi va umumiy holda ifodada jadvallar ustuntarning nomi hamda o'zgarmaslar uchrashi mumkin. Skalyar ifodalarni turlari solishtirishli bo'lishi kerak. So'rov qismi saqat o'ng operand bo'lishi mumkin va u bitta qiymatdan iborat bo'lishi kerak.

Agar so'rov qismi bo'sh bo'lsa yoki biror operand NULL bo'lsa, solishtirish predikatining qiymati NULLga teng. Shuni aytib o'tish kerakki, SQL/92 namunasida NULL maxsus qiymat deb hisoblanadi va GROUP BY, DISTINCT va ORDER BY konstruksiyalarda bir guruh deb hisobga olinadi.

### *Interval predikati*

Interval predikatni sintaksisi quyidagicha:

<interval predikati> ::= <skalyar ifoda>  
[NOT] BETWEEN <skalyar ifoda> AND <skalyar ifoda>  
«x BETWEEN y AND z» natijasi «x >= y AND x <= z» natijasi bilan bir xil. «x NOT BETWEEN y AND z» natijasi «NOT (x BETWEEN y AND z)» natijasi yoki «x < y OR x > z» natijasi bilan teng.

### *Mansublik predikati*

**Mansublik predikati** quyidagi sintaktik qoidalar bilan tavsiflanadi:

$\langle \text{mansublik predikati} \rangle ::= \langle \text{skalyar ifoda} \rangle [\text{NOT}] \text{ IN}$   
 $\quad \quad \quad \langle \text{so'rov qismi} \rangle | (\langle \text{skalyar ifoda} \rangle [, ...]) \}$

Chap operand va o'ng tomondagi ro'yxat qiymatlarning turlari solishtirishli bo'lishi kerak. Predikatni qiymati **TRUE**ga teng bo'ladi, agarda chap operand o'ng operand ro'yxatga mansub bulsada, o'ng operand ro'yxati bo'sh yoki chap operand ro'yxatga mansub bo'lmasa, predikatni qiymati **FALSE** teng, aks holda **NULL** bo'ladi. Tavsif bo'yicha «x NOT IN S» teng «NOT (x IN S)»ga.

### *Qolip predikati*

**Qolip predikati** quyidagi sintaksisga ega:

$\langle \text{qolip predikati} \rangle ::= \langle \text{skalyar ifoda} \rangle [\text{NOT}] \text{ LIKE } \langle \text{qolip} \rangle$   
 $\quad \quad \quad [\text{ESCAPE } \langle \text{belgi} \rangle]$   
 $\langle \text{qolip} \rangle ::= \langle \text{satr} \rangle$

Chap operand ustuni va namunani turlari satr turi bo'lishi kerak. **ESCAPE** dan keyin bitta belgi turishi kerak.

Agar namuna ustunni qismi bo'lsa, predikat qiymati **TRUE**ga teng. Namunada maxsus belgilari ishlatalishi mumkin, ularni ma'nosi quyidagicha: tag chiziq belgisi («\_») uni o'mida har qanday belgi bo'lishi mumkinligini bildiradi, foiz belgisi («%») o'mida har qanday belgilari ketma-ketligi turish mumkinligini bildiradi, ketma-ketlik bo'sh bo'lishi mumkin. Agarda **LIKE** predikatida **ESCAPE** ishlatalib, biror x belgisi tursa, u holda namunadagi «x\_» va «x%» belgilari shu joyda aynan «\_» va «%» belgilari turishini bildiradi.

Agarda ustun qiymati yoki namuna qiymati noma'lum bo'lsa, like predikatni qiymati **NULL**ga teng.

«x NOT LIKE y ESCAPE z» predikatning qiymati «NOT x LIKE y ESCAPE z» predikat qiymatiga teng.

### *Null predikati*

**Null predikati** quyidagicha tavsiflanadi:

$\langle \text{null predikati} \rangle ::= \langle \text{skalyar ifoda} \rangle \text{ IS } [\text{NOT}] \text{ NULL}$

Bu predikat faqat ikkita qiymatga ega, agar ustunni joriy qiymati noma'lum bo'lsa, predikatni qiymati **TRUE**, aks holda **FALSE** bo'ladi. «x NOT IS NULL» predikatni qiymati «NOT x IS NULL» predikatni qiymatiga teng.

## *Kvantor predikati*

Kvantor predikatining sintaksisi quyidagicha:

*<kvantor predikati> ::= <skalyar ifoda> <solishtirish amali>  
<kvantor> <so'rov qismi>  
<kvantor> ::= ALL | SOME | ANY*

S orqali so'rov qismi hisoblash natijasini belgilaymiz x orqali predikatni chap qismidagi arifmetik ifodani hisoblash natijasini belgilaymiz. Bunda S bo'sh bo'lsa yoki «x <solishtirish amali> s» predikat qiymati har qanday Sga kiruvchi s uchun TRUE bo'lsa, u holda «x <solishtirish amali> ALL S» predikat qiymati TRUE bo'ladi. Agarda «x <solishtirish amali> s» predikat qiymati Sga kiruvchi biror bir s uchun FALSE bo'lsa, predikat «x <solishtirish amali> ALL S» FALSE bo'ladi. Qolgan holatlar uchun «x <solishtirish amali> ALL S» predikat qiymati NULL bo'ladi.

Agarda S bo'sh bo'lsa yoki «x <solishtirish amali> SOME s» predikat qiymati Sga kiruvchi har qanday s uchun FALSE bo'lsa, predikat «x <solishtirish amali> SOME S» FALSE bo'ladi. Agarda «x <solishtirish amali> s» predikat qiymati Sga kiruvchi biror bir s uchun TRUE bo'lsa, predikat «x <solishtirish amali> SOME S» TRUE bo'ladi. Qolgan holatlar uchun «x <solishtirish amali> SOME S» predikat qiymati NULL bo'ladi. ANY va SOME ekvivalentdir.

## *Mavjudlik predikati*

Mavjudlik predikati quyidagi sintaksisiga teng:

*<mavjudlik predikati> ::= [NOT] EXISTS <so'rov qismi>*

EXISTS predikatning qiymati TRUE bo'ladi, agarda so'rov qismi hisoblash natijasida bo'sh bo'limasa, ya'ni hech bo'limasa bitta satr mavjud bo'lsa, aks holda FALSE bo'ladi, NOT EXISTS aksineha.

## **SELECT operatoriga döir misollar**

### *Oddiy so'rovlar*

Bizga talaba jadvali berilgan bo'lsin va unda quyidagi ustunlar mavjud bo'lsin: **talaba(id\_talaba, familiya, ismi, sharifi, millati, jinsi, pas\_ser, pas\_nom, tug\_yil, tug\_shahar, manzil, stipendiya, og'irligi, kod\_fak, kurs)**.

1. Jadvalni barcha ustunlarini va barcha satrlarini tanlash.

Barcha talabalar to'g'risida to'liq ma'lumot chiqarilsin. Bu holda barcha ustunlar ro'yxatini berish shart emas, uni o'miga \* belgisini ishlatalish yetarli.

```
SELECT *
FROM talaba;
```

2. Aniq ustunlar va barcha satrlarni tanlash.  
Talabalarni tartib raqami, familiya, nomi va otasinining ismini ro'yxatini chiqarilsin.

```
SELECT id_talaba, familiya, ismi, sharifi
FROM talaba;
```

3. DISTINCT kalit so'zini ishlatalish.  
Talabalar kelgan shaharlar ro'yxatini chiqarilsin.

```
SELECT tug_shahar
FROM talaba;
```

Bu jadvalda har bir talabani kelgan shahri chiqadi va shaharlar qaytariladi.

```
SELECT DISTINCT tug_shahar
FROM talaba;
```

Bu jadvalda oldingi jadvaldan qaytarilgan shaharlar olib tashlanadi va natijada qaytarilmas shaharlar ro'yxatini qoladi.

4. Hisoblanuvchi maydonlar.  
Talabanining yillik stipendiyasi hisoblansin.

```
SELECT id_talaba, familiya, ismi, stipendiya*12
FROM talaba;
```

Ustunlar nominini qulay ko'rinishda o'zgartirish mumkin:  
SELECT id\_talaba as nomer, familiya, ismi,
stipendiya\*12 as yil\_stipendiya
FROM talaba;

5. Satrlarni tanlash. (WHERE bo'limi)  
a) solishtirish predikatini ishlatalish  
1989-yilda tug'ilgan talabalar ro'yxatini chiqarilsin

```
SELECT id_talaba, familiya, ismi, sharifi  
      FROM talaba  
     WHERE tug_yil=1989;
```

1989 yoki 1990-yilda tug'ilgan talabalar ro'yxati chiqarilsin

```
SELECT id_talaba, familiya, ismi, sharifi, tug_yil  
      FROM talaba  
     WHERE tug_yil=1989 OR tug_yil=1990;
```

Toshkent yoki Samarqand shahridan kelgan va millati o'zbek bo'lgan talabalarni ro'yxati chikarilsin

```
SELECT id_talaba, familiya, ismi, sharifi,  
          tug_shahar  
      FROM talaba  
     WHERE (tug_shahar='Toshkent' or tug_shahar=  
           'Samarqand') and millati='o'zbek';
```

b) interval predikatini ishlatish.

1980 yildan 1994-yilgacha tug'ilgan talabalarni ro'yxati chiqarilsin  
1- variant.

```
SELECT id_talaba, familiya, ismi, sharifi, tug_yil  
      FROM talaba  
     WHERE tug_yil>=1980 AND tug_yil<=1994;
```

2- variant.

```
SELECT id_talaba, familiya, ismi, sharifi, tug_yil  
      FROM talaba  
     WHERE tug_yil BETWEEN 1980 AND 1994;
```

1980-yildan 1994-yilgacha tug'ilgan talabalardan boshqa, hamma tala-  
balarning ro'yxati chiqarilsin.

```
SELECT id_talaba, familiya, ismi, sharifi, tug_yil  
      FROM talaba  
     WHERE tug_yil NOT BETWEEN 1980 AND 1994;
```

c) To'plamga kirish predikatini ishlatish.

Toshkent, Samarqand, Buxoro, Andijon, Urganch, Qarshi shaharlardan kelgan talabalar ro'yxati chiqarilsin.

```
SELECT id_talaba, familiya, ismi, sharifi, tug_
      shahar
   FROM talaba
 WHERE tug_shahar IN ('Toshkent', 'Samargand',
 'Buxoro', 'Andijon', 'Urganch', 'Qarshi');
```

Toshkent va Samarqand shaharlardan tashqari kelgan talabalar ro'yxatini chiqarish.

```
SELECT id_talaba, familiya, ismi, sharifi, tug_
      shahar
   FROM talaba
 WHERE tug_shahar NOT IN
 ('Toshkent', 'Samargand');
```

Bizga qo'shimcha jadval berilgan bo'lsin:

shaharlari(id\_shahar, shahar\_nomi, vil\_nomi)

Toshkent viloyatidagi shaharlardan kelgan talabalar ro'yxati chiqarilsin.

```
SELECT id_talaba, familiya, ismi, sharifi, tug_
      shahar
   FROM talaba
 WHERE tug_shahar IN (SELECT shahar_nomi
   FROM shaharlari
 WHERE vil_nomi='Toshkent viloyati');
```

Bu misolda shaharlar to'plami so'rov qism orqali berilgan.

d) Qolip predikatini ishlatalish.

Familiyasi birinchi harfi «A» uchinchisi «b» bo'lgan talabalarni ro'yxati chiqarilsin

```
SELECT id_talaba, familiya, ismi, sharifi,
   FROM talaba
 WHERE familiya LIKE 'A_b%';
```

Familiyasining birinchi harfi «A», uchinchisi «b» bo'lmagan talabalarni ro'yxati chiqarilsin

```
SELECT id_talaba, familiya, ismi, sharifi,  
      FROM talaba  
     WHERE familiya NOT LIKE 'A_b%';
```

e) Noma'lumli predikatni ishlatalish.  
Otasi nomi yo'q talabalar ro'yxatini chiqarilsin

```
SELECT id_talaba, familiya, ismi,  
      FROM talaba  
     WHERE sharifi IS NULL;
```

Toshkent shahrida yashovchi, lekin manzili noma'lum bo'lgan talabalar  
ro'yxati chiqarilsin

```
SELECT id_talaba, familiya, ismi, sharifi  
      FROM talaba  
     WHERE tug_shahar='Toshkent' AND manzil IS  
          NULL;
```

Pasport raqami mayjud bo'lgan talabalar ro'yxati chiqarilsin

```
SELECT id_talaba, familiya, ismi, sharifi,  
      pas_ser, pas_nomer  
      FROM talaba  
     WHERE pas_nomer IS NOT NULL;
```

So'rovlarni ehtiyojkorlik va diqqat bilan noma'lum qiymatlarni hisobga  
olib tuzish kerak. Masalan, bizga ikki satr berilgan bo'lsin:

1 Karimov Maqsud Abdullayevich ....

2 Usmanov Alisher ....

va bundan ikkinchì satmi tanlash lozim bo'lsin, unda quyidagi ikkita  
so'rovlar natijsi bo'sh satr beradi, chunki ikkinchi satrda **otasining** nomi  
noma'lum :

```
SELECT id_talaba, familiya, ismi, sharifi  
      FROM talaba  
     WHERE sharifi='';
```

```
SELECT id_talaba, familiya, ismi, sharifi  
      FROM talaba  
     WHERE sharifi<>'Abdullayevich';
```

Birinchi so'rovda sharifi bo'sh satr tekshirilmoqda, lekin **NULL** bu bo'sh satr emas, shu sababli ikkala satr uchun shart qiymati bajarilmaydi, ya'ni **FALSE** va **NULL** bo'ladi.

Ikkinci so'rovda tengsizlik sharti qo'yilgan bu shart birinchi satr uchun **FALSE** beradi, ikkinchi satr uchun **NULL** beradi, chunki **NULL** bilan har qanday solishtirish **NULLga** teng, shu sababli ikkala satr uchun ham shart bajarilmaydi.

To'g'ri so'rov esa quyidagicha bo'ladi:

```
SELECT id_talaba, familiya, ismi, sharifi  
FROM talaba  
WHERE sharifi IS NULL;
```

Qolgan predikatlarga tegishli misollarni so'rov qismida batatsil ko'rsatib beramiz.

### Natijani tartiblash

Umumiy holda natijaviy jadvalda satrlar tartiblanmagan. Lekin ularni tartiblash imkoniyati bor, buning uchun **ORDER BY** konstruksiyasi qo'llanadi. **ORDER BY** konstruksiyasi tartiblash ustunlari yoki ustun raqami ko'rsatiladi. Agar bunday ustunlar bir nechta bo'lsa, avval birinchi ustun bo'yicha tartiblanadi, keyin birinchi ustundagi bir xil qiyamatlar uchun ikkinchi ustun tartiblanadi va h.k. **ASC** va **DESC** kalit so'zlar orqali ustundagi berilganiarni o'sish yoki kamayish bo'yicha tartiblash mumkin. Oshkormas ravishda o'sish bo'yicha tartiblanadi, ya'ni **ASC** faraz qilinadi. **ORDER BY** konstruksiyasida jadval ifodani har qanday ustuni bo'yicha tartiblash mumkin va bu ustunni **SELECT** ro'yxatiga kirish shart emas. Sintaksidan ko'rinish turibdiki, bu konstruksiya **SELECT** operatorining oxirida turishi kerak.

Bir ustun qiymati bo'yicha tartiblash.

Talabalar ro'yxatini familiyasi bo'yicha tartiblansin.

```
SELECT familiya, ismi, sharifi  
FROM talaba  
ORDER BY familiya;
```

1. Bir nechta ustun bo'yicha tartiblash.

Talabalar ro'yxatini tug'ilgan yili kamayish va familiyasi alfavit o'sish bo'yicha tartiblansin.

```
SELECT familiya, ismi, sharifi, tug_yil
FROM talaba
ORDER BY tug_yil DESC, familiya;
```

bu misolda bir yilda tug'ilgan talabalar familiyasi bo'yicha tartiblanadi.

2. Hisoblanuvchi ustunlar bo'yicha tartiblash.

Order by guruhdag'i ustunlar ro'yxatida psevdonimlarni ishlatalish mumkin emas, agar ustun ifoda bo'lsa va shu ustun bo'yicha tartiblansa, uni yana ORDER BY ro'yxatida qaytarish kerak. Talabalar ro'yxati yillik stipendiyasi bo'yicha tartiblansin:

```
SELECT familiya, ismi, sharifi, stipendiya*12 as
      yil_stipendiya
      FROM talaba
      ORDER BY stipendiya*12;
```

### Xulosa

1. Predikatlar orqali tanlash shartlari beriladi.
2. LIKE operatori qolipni beradi.
3. Mansublik predikati to'plamga tegishligini bildiradi.
4. Mayjudlik predikati so'rov qismida satr mavjudligini **bildiradi**.

### Amaliyot uchun topshiriqlar

1. Berilganlar bazasidan barcha talabalar to'g'risida to'liq ma'lumotlar chiqarilsin.
2. Berilganlar bazasidan Toshkent shahrida turadigan talabalar to'g'risida to'liq ma'lumot berilsin.
3. 9-raqamli fakultet talabalarini familiyasi alfavit bo'yicha tartiblangan holda chiqarilsin.
4. Manzil ma'lumoti kiritilmagan talabalar ro'yxati chiqarilsin.
5. Familiyasi A harfdan boshlanib v harf bilan tugallangan talabalarning ro'yxati chiqarilsin.
6. Tug'ilgan yili 1980 va 1990-yillar oralig'ida bo'lgan talabalar ro'yxati chiqarilsin.
7. Anvar, Rustam, Alisher, To'lqin, Shavkat ismli talabalar ro'yxati chiqarilsin.

---

## 25-bob, NATIJA OLISH VA GURUHLASH

**Tayanch iboralar:** *agregat funksiya, natijani guruhash, GROUP BY konstruksiya, HAVING konstruksiya, ANY, SAME, ALL, EXISTS, NOT EXISTS, korrelatsion so'rov.*

### SQL tilida agregat (yig'uvchi) funksiyalarni ishlatish

SQL tilida beshta agregat funksiyalar mavjud:

**COUNT** – ko'rsatilgan ustundagi qiymatlar sonini hisoblaydi;

**SUM** – ko'rsatilgan ustundagi qiymatlar yig'indisini hisoblaydi;

**AVG** – ustunning o'rta qiymatini hisoblaydi;

**MIN** – ustundagi minimal qiymatni topadi;

**MAX** – ustundagi maksimal qiymatni topadi.

Bu funksiyalar guruhlangan satrlar va faqat bitta ustun bo'yicha yoki ifoda bo'yicha hisoblaydi va faqat bitta qiymatni qaytaradi.

**COUNT, MAX, MIN** har qanday qiymatlar ustidan bajarilishi mumkin, **SUM, AVG** funksiyalar faqat sonli qiymatlar ustidan bajariladi.

**COUNT(\*)** dan tashqari, barcha funksiyalar oldin nomallum qiymatlami (**NULL**) olib tashlab, keyin amal bajariladi. **COUNT(\*)** funksiyani vazifasi natijaviy jadvalda satrlar sonini hisoblashdir. Agregat funksiyalarda ustundagi bir xil qiymatlarini hisobga olmaslik uchun ustun oldida **DISTINCT** kalit so'zini qo'llash kerak. Agregat funksiyalar faqat select ro'yxatida yoki having bo'lmidagi, qo'llanishi mumkin. Agar select ro'yxatida agregat funksiyalar ishlatsilsa va group by bo'limi bo'lmasa, u holda ro'yxatda boshqa ustunlar nomi bo'lishi mumkin emas. Masalan, quyidagi so'rov xato:

```
SELECT tug_shahar, SUM(stipendiya)
      FROM talaba;
```

bu misolda tug\_shahar ustunni ishlatish xatolikka olib keladi, to'g'ri so'rov quyidagicha yoziladi:

```
SELECT tug_shahar, SUM(stipendiya)
      FROM talaba
      GROUP BY tug_shahar;
```

### Misollar:

1. Talabalar umumiylar soni aniqlansin.

```
SELECT COUNT(*) AS talaba_soni  
      FROM talaba;
```

2. Yuqori stipendiya oladigan talabalar soni aniqlansin.

```
SELECT COUNT(*) AS alo_soni  
      FROM talaba  
     WHERE stipendiya>150000;
```

3. Talabalarni o'rta stipendiya haqi ming so'mda aniqlansin:

```
SELECT AVG(stipendiya)/1000 AS urta_stip  
      FROM talaba;
```

4. Talabalar nechta shahardan kelganligi aniqlansin:

```
SELECT COUNT(DISTINCT tug_shahar) AS son  
      FROM talaba;
```

5. Toshkent shahridan talabalar soni, eng katta yoshi va ular oladigan stipendiyaning yig'indisi aniqlansin:

```
SELECT COUNT(*) AS S, MIN(tug_yil) as mint,  
SUM(stipendiya) as maosh  
      FROM talaba  
     WHERE tug_shahar='Toshkent';
```

### Natijalarni guruhlash

Yuqorida ko'rsatilgan misollarda bitta satrda umumiylar natija hosil bo'ladi, lekin ba'zi bir hollarda oraliq natijalarni hisoblash kerak bo'ladi. Bu holda group by bo'limi qo'llanilishi kerak. Group by bo'limi ishlataligan so'rovlari, guruhlovchi so'rovlari deb nomlanadi, chunki SELECT operatori ishlash natijasida hosil bo'lgan satrlar, ustunlar qiymati bir xil bo'yicha guruhlanadi va keyin har bir guruh uchun yig'uvchi satr hosil bo'ladi. Agregat funksiyada uchramagan, lekin select-ro'yxatda uchragan har qanday ustun albatta group by bo'limida qatnashishi lozim. Lekin teskarisi shart emas, ya'ni group by bo'limida uchragan ustun select ro'yxatida bo'lishi shart emas.

Agar tanlash operatororda where bo'limi tursa, avval u bajariladi va tanganlangan satrlar keyin guruhlanadi. Guruhlovchi ustunda ba'zi bir qiymatlar noma'lum bo'lsa, ular uchun alohida guruh tashkil qilinadi.

### Misollar:

1. Har bir shahar bo'yicha talabalar soni aniqlansin:

```
SELECT tug_shahar, COUNT(*) AS ts  
      FROM talaba  
     GROUP BY tug_shahar;
```

2. Talabalarning tug'ilgan shahar va tug'ilgan yili bo'yicha taqsimotj aniqlansin:

```
SELECT tug_shahar, tug_yil, COUNT(*) AS ts
FROM talaba
GROUP BY tug_shahar,tug_yil
ORDER BY tug_shahar,tug_yil;
```

Bu misolda talabalar soni shaharlar bo'yicha va har bir shaharda tug'ilgan yili bo'yicha chiqariladi.

3. Har bir shaharda tug'ilgan yigitlar soni aniqlanib, natija kamayish tartibida chiqarilsin:

```
SELECT tug_shahar, COUNT(*) AS ts
FROM talaba
WHERE jinsi='erkak'
GROUP BY tug_shahar
ORDER BY COUNT(*) DESC;
```

Yuqoridagi so'rnvi bajarish amallari quyidagicha:

1. Talaba (talaba) jadvaldan jinsi (jinsi) «erkak» bo'lgan satrlar tanlanadi.
2. Tanlangan satrlar tug'ilgan shahri (tug\_shahar) bo'yicha guruhlari wadi.
3. Har bir guruh uchun satrlar soni hisoblanadi.
4. Natijada shahar nomi va talabalar sonidan bo'lgan satrlar jadvali hosil bo'ladi.
5. Hosil bo'lgan satrlar talabalar sonining kamayishi tartibida chiqariladi.

### Guruhlarni tanlash

**HAVING** konstruksiyasi **GROUP BY** bilan birlgilikda natijaviy jadvaldagi guruhlari ichida shart bo'yicha kerakli guruhlarni tanlaydi.

**HAVING** va **WHERE** konstruksiyalarni sintaksisi bir-biriga o'xshaydi, lekin ma'nosi har xil. **WHERE** konstruksiya manba jadvaldan satrlarni tanlaydi, **HAVING** hosil bo'lgan guruhlardan tanlaydi. Amalda **HAVING** da agregat funksiyalar ishlataladi, ustunlar nomi **WHERE**da ko'rsatiladi.

Misol: Har bir shahar bo'yicha tug'ilgan yigitlar soni aniqlansin va talabalarining soni 100dan ortiq bo'lgan shaharlar chiqarilsin:

```
SELECT tug_shahar, COUNT(*) AS ts
  FROM talaba
 WHERE jinsi='erkak'
 GROUP BY tug_shahar
 HAVING COUNT(*)>100
 ORDER BY COUNT(*) DESC
```

### So'rov qismi

**SELECT** operator ichida boshqa **SELECT** operatori ishlatalishni ko'tramiz. Tashqi **SELECT** operatori ichki operatorni natijasini ishlataladi.

Ichki so'rovlardan **HAVING** va **WHERE** konstruksiyalarda joylanishi mumkin. Undan tashqari ichki so'rovlardan **INSERT**, **UPDATE** va **DELETE** operatorlarda qo'llanilishi mumkin.

So'rov qismalar uch xil bo'lishi mumkin:

1. Skalyar so'rov qismi. Bu holda so'rov qismi bir ustun va bir satr dan iborat bo'lgan yagona qiymatni qaytaradi. Bunday so'rovni **SELECT** operatorda yagona qiymat ishlataligan joyda, uning o'miga ishlatalish mumkin.
2. Satr so'rov qismi bir nechta ustunli yagona satr ko'rinishda qaytaradi. Bunday so'rov satr qiymatni o'miga ishlatalishi mumkin. Asosan predikatlarda qo'llanadi.
3. Jadval so'rov qismi. Bitta va undan yuqori ustunlardan iborat bo'lgan jadval satrlarini qaytaradi. Sintaksis bo'yicha jadval ishlatalishi mumkin bo'lgan joyda jadval so'rov qismi qo'llanilishi mumkin, masalan **IN** predikatida.

#### Misollar:

1. Skalyar so'rov qismini ishlatalish.

Bizga qo'shimcha jadval berilgan bo'lsin:

fakultet(id\_fak, name\_fak, dekan\_fak, telefon)

Karimov dekan bo'lgan fakultet talabalarining ro'yxati chiqarilsin.

```
SELECT familiya, ismi, sharifi, kurs
  FROM talaba
 WHERE kod_fak=(SELECT id_fak FROM fakultet
                  WHERE dekan_fak='Karimov')
```

Bu misolda so'rov qismi skalyar turda bo'lib, faqat yagona qiymatni beradi, masalan id\_fak=1. So'rov qismi bajarish natijasida fakultet nomeri aniqlanadi so'ng tashqi so'rov bajariladi va shart bo'yicha (kod\_fak=1) shu

nomerli satrlar tanlanadi, ya'ni yuqoridagi so'rov quyidagi so'rovga almashtiriladi:

```
SELECT familiya, ismi, sharifi, kurs  
      FROM talaba  
     WHERE kod_fak=1;
```

So'rov qismini agregat funksiyalarda ishlatish.

Stipendiya qiymati o'tta stipendiya qiymatidan kam bo'lgan talabalar ro'yxati chiqarilsin va stipendiya haqi bilan o'tta qiymat ayirmasi chiqarilsin:

```
SELECT familiya, ismi, sharifi, kurs, stipendiya  
      (SELECT AVG(stipendiya) FROM talaba) as airim  
      FROM talaba  
     WHERE stipendiya<(SELECT AVG(stipendiya)  
      FROM talaba);
```

*Shuni aytib o'tish kerakki, agregat funksiyalarni WHERE konstruksiyasida ishlatish mumkin emas, ya'ni quyidagi yozuv xato bo'ladi:*

```
WHERE stipendiya<AVG(stipendiya).
```

So'rov qismi quyidagi qoidalarga rioya qilishi kerak:

1. So'rov qismida **ORDER BY** konstruksiyani ishlatisn mumkin emas.
2. So'rov qismini **SELECT** ro'yxatida ustunlar nomi yoki ustunlardan tuzilgan ifoda bo'lishi mumkin. Tashqi so'rovda **EXISTS** kalit so'zi ishlatilsa, ichki so'rov ro'yxatida tashqi so'rovdagagi jadvalning ustunlarini ishlatish mumkin.
3. Oshkormas ravishda ustunlar nomi so'rov qismidagi **FROM** konstruksiyasida yozilgan jadvallarga tegishlidir. Tashqi so'rov ustunlariga ham murojaat qilish mumkin, bu holda oshkor kengaytirilgan ism ishlatiladi.
4. Agar so'rov qismi solishtirish amalida qatnashsa, u amalning o'ng qismida yozilishi kerak.

So'rov qismini **IN** predikatida ishlatalishga misol:

Bizga quyidagi o'zlashtirish bo'yicha qo'shimcha jadval berilgan bo'linsin:

**uzlashtirish** (id\_talaba, predmet, ball)

Karimov dekan bo'lgan fakultet talabalarining predmet bo'yicha olgan ballari ro'yxati chiqarilsin.

```

SELECT id_talaba, predmet, ball
    FROM uzlashtirish
   WHERE id_talaba IN (SELECT id_talaba
    FROM talaba
   WHERE kod_fak = (SELECT id_fak
    FROM fakultet
   WHERE dekan_fak='Karimov'))
        ORDER BY predmet,id_talaba

```

Eng ichki so'rov qismining ishlashi natijasida fakultet kodi qiymati aniqlanadi, so'ng undan yuqori so'rov qismi bir ustundan iborat bo'lgan talabalar kodi ro'yxati hosil bo'ladi va tashqi so'rov shu ro'yxatga kirgan talabalarning ballarini chiqaradi.

### **ANY (SOME) va ALL kalit so'zlarni ishlatalishi**

**ANY** va **ALL** kalit so'zlari so'rov qismlari bilan birga ishlataladi va bitta ustun bo'yicha qiymatlarni qaytaradi. Agar so'rov qism oldida **ALL** so'zi tursa va solishtirish sharti barcha so'rov qism ustunidagi qiymatlar uchun bajarilsa, shart bajarildi deb hisoblanadi. Agar so'rov qism oldida **ANY** yoki **SOME** kalit so'zlari turgan bo'lsa solishtirish sharti so'rov qism ustunidagi biror bir qiymat uchun bajarilsa, shart bajarildi deb hisoblanadi. Agarda so'rov qismini bajarish natijasida bo'sh qiymat hosil bo'lsa, unda **ALL** uchun shart bajarildi va **ANY** uchun shart bajarilmadi, deb hisoblanadi.

#### **Misollar:**

1. Talabaning og'irligi 1 raqamli fakultetdagi talabalarning birortasining og'irligidan katta bo'lgan talabalarning ro'yxatini chiqaring

#### *1-variant.*

```

SELECT familiya, ismi, sharifi, ogirligi
    FROM talaba
   WHERE ogirligi>SOME (SELECT ogirligi
    FROM talaba
   WHERE kod_fak=1);

```

#### *2-variant.*

```

SELECT familiya, ismi, sharifi, ogirligi
    FROM talaba
   WHERE ogirligi>(SELECT MIN(ogirligi)
    FROM talaba
   WHERE kod_fak=1);

```

2. Talabaning og'irligi 1 raqamli fakultetdagi talabalarning barchasini  
og'irligidan katta bo'lgan talabalarning ro'yxatini chiqarining

*1-variant.*

```
SELECT familiya, ismi, sharifi, ogirligi
      FROM talaba
 WHERE ogirligi > ALL (SELECT ogirligi
      FROM talaba
 WHERE kod_fak=1);
```

*2-variant.*

```
SELECT familiya, ismi, sharifi, ogirligi
      FROM talaba
 WHERE ogirligi > (SELECT MAX(ogirligi)
      FROM talaba
 WHERE kod_fak=1);
```

Bu misollardan ko'rinish turibdiki, bir xil masalani turli xildagi yo'llar  
bilan yechish mumkin.

### EXISTS va NOT EXISTS predikatlarni qo'llash

Bu kalit so'zlar faqat so'rov qismlari bilan birga ishlataladi. Ularni qo'llash  
natijasida ikkita mantiqiy qiymat TRUE yoki FALSE hosil bo'lishi  
mumkin. Agar so'rov qismini bajarish natijasida biror bir satt mavjud  
bo'lsa, EXISTS predikati TRUEga teng, aks holda FALSEga teng bo'ladi.  
NOT EXISTS predikati teskari natija beradi. So'rov qismida ustunlar  
sonini ahamiyati bo'lmagan sababli, uni ko'rinishi quyidagicha bo'lishi  
mumkin:

```
(SELECT * FROM ...)
```

**Misollar:**

1. Biror bir fandan yuqori ball olgan talabalarning ro'yxati chiqarilsin:

```
SELECT familiya, ismi, sharifi
      FROM talaba t
 WHERE EXIST (SELECT * FROM uzlashtirish u
      WHERE t.id_talaba=u.id_talaba AND
          ball>=86);
```

2. Barcha fanlardan yuqori ball olgan talabalarning ro'yxati chiqarilsin:

```
SELECT familiya, ismi, sharifi
  FROM talaba t
 WHERE NOT EXIST (SELECT * FROM uzlashtirish u
      WHERE t.id_talaba=u.id_talaba AND ball<86);
```

Ko'rsatilgan misollardagi ichki so'rovlarda, tashqi so'rovdan uzatilgan parametr mavjud (t.id\_talaba) bo'lib, bunday ichki so'rovlar *korrelatsion so'rov* deb nomlanadi. Bu holda tashqi so'rovni har bir satiri uchun ichki so'rov qayta hisoblanadi va tashqi parametrga teng bo'lgan satrlar tanlanadi va shu satrlar uchun boshqa shartlar qo'llanadi. Shuni aytib o'tish kerakki, korrelatsion so'rovlar hisoblashni sekinlashtirmaydi, chunki aslida so'rovni bajarilishi BBBTni ichki algoritmiga bog'liqdir. Chunki SQL tili bu deklarativ tildir, ya'ni unda qanday natija kerakligi ko'rsatiladi, natijani qaysi usul bilan olish bu BBBTga bog'liqdir.

### Xulosa

- SQL tilida beshta agregat funksiyalar aniqlangan (COUNT, SUM, AVG, MIN va MAX). Bu funksiyalarda parametr sifatida ustunning barcha elementlari bo'yicha qiymatlari olinadi va natijada bitta qiymat qaytadi.
- Agar GROUP BY konstruksiysi operatoriga kelmasa, tanlash ro'yxatida ustun nomi va agregat funksiya qatnashishi mumkin emas.
- GROUP BY konstruksiya natijaviy jadvalga guruh bo'yicha natija olishga imkon beradi.
- Guruhi ichida tanlash uchun HAVING konstruksiysi ishlataladi. HAVING konstruksiyasida agregat funksiyalarni ishlatalish mumkin.
- So'rov qismi bu boshqa so'rovni ichiga kirgan to'liq SELECT operatori. Ichki so'rov tashqi operatorning WHERE yoki HAVING konstruksiyalarida ishlatalishi mumkin. So'rov qismi ichida yana so'rov qismi bo'lishi mumkin.
- So'rov qismi uch xil bo'lishi mumkin: skalyar, satr va jadval. Skalyar so'rov qismi faqat bitta qiymat qaytaradi. Satr so'rov qismi bitta satr va bir nechta ustun qaytaradi. Jadval so'rov qismi bir yoki undan ko'p ustunlar va bir nechta satrlarni qaytaradi. Skalyar so'rov qismi bir qiymat turadigan joylarda ishlatalishi mumkin. Satr so'rov qismi predikatlarda ishlatalishi mumkin.

## **Nazorat uchun savollar va amaliyot uchun topshiriqlar**

1. Agregat funksiyalarga qanday chegaralar qo'yilgan? Agregat funksiyalar NULLni qanday qayta ishlaydi?
2. GROUP BY konstruksiyani ishlash prinsipini tushuntirib bering.
3. WHERE va HAVING konstruksiyalarni farqi nimada?
4. So'rov qismi va tutashtirishni nima farqi bor. Qaysi hollarda so'rov qismini ishlatalish imkoniyati bo'lmaydi?
5. HAVING konstruksiyasi ishlataligan so'rovga HAVINGsiz ekvivalent so'rov keltirish mumkinligini isbotlang.
6. SQL tili relations to'liqligini isbotlang.

## **Agregat funksiyalarni qo'llash va guruhash**

7. Har bir fakultetdagi talabalar soni chiqarilsin.
8. Har bir fakultetdagi talabalarni o'rta og'irligi chiqarilsin.
9. I raqamli fakultetni bir oylik stipendiyasi chiqarilsin.
10. Talabalar tug'ilgan shahridan talabalar soni 200 dan ortiq bo'lgan shaharlar chiqarilsin.

---

## **26-boʻb. JADVALLARNI BIRLASHMASI, KESISHMASI VA AYIRMASI**

**Tayanch iboralar:** UNION, INTERSECT, EXCEPT, CROSS JOIN, NATURAL JOIN, JOIN ... ON, JOIN ... USING, INNER JOIN, OUTER JOIN.

Relatsion algebrani amallarini bajarish uchun SQL tilida maxsus operatorlar kiritilgan. Bu UNION, INTERSECT va EXCEPT operatorlaridir. Ikki soʼrovni birlashtirish uchun UNION operatori, soʼrovlarni kesmasi INTERSECT operatori, ayirmasi EXCEPT operatori yordamida bajariladi.

### **Misollar:**

Bizga yana bir oʼqituvchilar boʼyicha jadval berilgan boʼlsin:

Uqituvchi(id\_uqituvchi, familiya, ismi, sharifi, millati, jinsi, pas\_ser, pas\_nom, tug\_yil, tug\_shahar, manzil, kod\_fak, unvoni, favozim).

1. I raqamli fakultetdagi talabalar yoki oʼqituvchilarning tug'ilgan shaharlari boʼyicha roʼyxati chiqarilsin:

```
SELECT DISTINCT tug_shahar
  FROM talaba
 WHERE kod_fak=1
 UNION
      SELECT DISTINCT tug_shahar
  FROM uqituvchi
 WHERE kod_fak=1
```

2. I raqamli fakultetdagi talabalar ya oʼqituvchilarning tug'ilgan shaharlari boʼyicha roʼyxati chiqarilsin:

```
SELECT DISTINCT tug_shahar
  FROM talaba
 WHERE kod_fak=1
 INTERSECT
      SELECT DISTINCT tug_shahar
  FROM uqituvchi
 WHERE kod_fak=1
```

3. I raqamli fakultetdagi talabalar tug'ilgan shaharlarda tug'ilmagan o'qituvchilar holatidagi shaharlar bo'yicha ro'yxat chiqarilsin:

```
SELECT DISTINCT tug_shahar
  FROM talaba
 WHERE kod_fak=1
    EXCEPT
SELECT DISTINCT tug_shahar
  FROM uqituvchi
 WHERE kod_fak=1
```

### Ko'p jadvalli so'rovlari

Yuqorida biz faqat bir jadvalli so'rovlarni ko'rib chiqdik. Lekin ko'pincha natijaviy jadval ustunlari bir nechta manba jadvallar ustunlaridan tashkil topadi. Buning uchun manba jadvallariga tutashtirish amalini qo'llash kerak. SQL tilida tutashtirish amali ikkita jadvalni bog'lash uchun ishlatalidi, natijaviy jadvalning satrlari manba jadvallarining satrlarini o'zaro tutashtirish orqali hosil bo'ladi. Natijada ikki jadvalni dekart ko'paytmasi hosil bo'ladi, ya'ni birinchi jadvalni har bir satrni ikkinchi jadvalning har bir satr bilan ularib, natijaviy jadval satrlarini hosil qiladi. So'ng ikki jadvaldan tanlangan ustunlardagi qiymatlari teng bo'lgan satrlar qoldiriladi. Jadvallarni tutashtirish uchun **FROM** konstruksiyasida jadvallarning nomlarini ko'rsatish yetarlidir. Undan keyin **WHERE** konstruksiyasida ustunlarni tutashtirish shartini ko'rsatish kerak. Undan tashqari **FROM** konstruksiyasida jadvallar uchun taxalluslarni ko'rsatish mumkin. Taxalluslarni bir xil nomli ustunlarni ajratish uchun aniqlovchi kengaytirilgan ustun ismlarida ishlatalish mumkin. Taxallus jadval nomi o'miga har qanday joyda ishlatalishi mumkin.

### Oddiy tutashish

Ulanishlarni ko'rsatish uchun bir nechta bir-biri bilan bog'langan jadvallarni olamiz.

*Talaba* jadvalida talaba to'g'risida anketa ma'lumotlari saqlanadi, *prikaz* jadvalida talabani joriy o'quv yilida o'qish joyi, uning harakati va boshqa o'qishga tegishli ma'lumotlari saqlanadi va yana bir nechta qoshimcha jadvallarni ko'rinishini keltiramiz:

*talaba*(id\_talaba, familiya, ismi, sharifi, id\_millati, jinsi, pas\_ser, pas\_nom, tug\_yil, manzil, og'irligi, tug\_joi, id\_tum, id\_ray, kod\_fak).

*prikaz*(id\_prikaz, id\_talaba, ykuv\_yil, id\_fak, id\_spec, tipob, tipfin, kurs, til, tip\_stip, tip\_prik)

fakultet(id\_fak, name\_fak, dekan, telefon)  
spetsialnost(id\_spec, namespec, id\_fak, tipob, shifrspec)  
millatlar(id\_millati, name\_millat)  
tumanlar(id\_tum, name\_tum)  
rayon(id\_ray, id\_tum, name\_ray)

Misollar:

1. Talabalar ro'yxati familiyasi, ismi, sharifi va tug'ilgan tumani ko'rsatilgan holda chiqarilsin:

```
SELECT familiya, ismi, sharifi, name_tum  
FROM talaba t, tumanlar tu  
WHERE t.id_tum=tu.id_tum;
```

2. Talabalar ro'yxati familiyasi, ismi, sharifi, fakultet nomi, yo'naliishi, kursi va ta'lim tili ko'rsatilgan holda chiqarilsin:

```
SELECT t.familiya, t.ismi, t.sharifi,  
f.name_fak, s.namespec, p.kurs, p.til  
FROM talaba t, prikaz p, fakultet f,  
spetsialnost s  
WHERE t.id_talaba=p.id_talaba AND  
p.id_fak=f.id_fak AND  
p.id_spec=s.id_spec AND p.ykuv_il=2011  
ORDER BY name_fak, namespec, kurs, yaz,  
familiya;
```

Ko'pincha so'rovlarda ikkita jadval ulanganda, biri ajdod jadvali bo'lib, ikkinchisi avlod jadval bo'ladi. Ajdod jadvalni tutashtirish uchun birlamchi kalit ustuni ishlataladi, avlod jadvalda unga mos bo'lgan ularish ustuni tashqi kalit hisoblanadi. Bizni misolda talaba va prikaz jadvallari tutashtirishda talaba jadvali ajdod bo'lib, prikaz jadvali avlod jadvali va id\_talaba ustuni ajdod jadvalda birlamchi kalit bo'lib, avlod jadvalda xuddi shu nomdag'i ustun tashqi kait bo'ladi va shu ustunlardagi qiymatlari teng bo'lgan satrlar orqali jadvallar ularadi.

Amalda WHERE konstruksiyasi tanlash uchun ishlatilsa, tutashtirish uchun esa JOIN maxsus konstruksiyasi ishlataladi va bu konstruksiya FROM konstruksiyasi tarkibida yoziladi.

### Jadvallarni tutashtirish sintaksisi

Jadvallarni tutashtirish sintaksisi quyidagicha bo'ladi:  
<ulangan jadval> ::= <o'zaro kesishilgan ularish> |

<tabiiy ulanish> | <predikat orqali ulanish> |  
 <ustun nomi orqali ulanish> | <birlashma ulanish>  
  
 <ulanish turi> ::= INNER | LEFT [OUTER] | RIGTH [OUTER] |  
 FULL [OUTER]  
 <o'zaro kesishilgan ulanish> ::= <jadval tasnifi> CROSS JOIN  
 <jadval tasnifi>  
  
 <tabiiy ulanish> ::= <jadval tasnifi> [NATURAL]  
 [<ulanish turi>] JOIN <jadval tasnifi>  
  
 <predikat orqali ulanish> ::= <jadval tasnifi> [<ulanish turi>]  
 JOIN <jadval tasnifi> ON <predikat>  
  
 <ustun nomi orqali ulanish> ::= <jadval tasnifi>  
 [<ulanish turi>] JOIN <jadval tasnifi>  
 USING (<ustun nomi>,...)  
  
 <birlashma ulanish> ::= <jadval tasnifi>  
 UNION JOIN <jadval tasnifi>

Ishlatilgan atamalarni izohini berib o'tamiz.

**CROSS JOIN** – O'zaro kesishilgan ulanish, bu tutashtirishda natijaviy jadval manba jadvallarni dekart ko'paytmasiga teng bo'ladi. Bunday ulanishni FROM bo'limidagi jadvallarning ro'yxati orqali ham amalga oshirish mumkin.

**NATURAL JOIN** – Tabiiy tutashtirishda, chap va o'ng jadvallarni bir xil ustun ismlari bo'yicha ulanadi, natijaviy jadvalga bir xil ismlar bir marta kiradi.

**JOIN ... ON** – Predikat orqali tutashtirishda, chap va o'ng jadvallarni ko'rsatilgan predikat yordamida ulanadi.

**JOIN ... USING** – Ustunlar nomi bo'yicha tutashtirishda, USINGda ko'rsatilgan umumiy ustunlar bo'yicha ikkita jadval ulanadi.

**OUTER - OUTER** (tashqi ulanish) kalit so'zi majburiy emas, uni yozmasa ham bo'ladi.

**INNER** – Ichki tutashtirish turi. Bu kalit so'zi orqali, ulanish turi oshkor ravishda ko'rsatilmaganda amalga oshiriladi. Tutashtirish natijasida faqat mos bo'lgan satrlar qoladi, ya'ni solishtirilgan ustunlardagi qiymati bir xil bo'lgan satrlar qoladi. Agar chap jadvalning ustunini biron bir qiymati,

o'ng jadvaldag'i solishtirilayotgan ustunning hech bir qiymatiga teng bo'lmasa, bunday satrlar olib tashlanadi.

**LEFT [OUTER]** – Chap tashqi tutashtirish turi. Bu tutashtirishda natijaviy jadvalga chap jadvalning barcha satrlari kiradi va o'ng jadvaldan esa faqat mos tushgan satrlar kiradi. Chap jadvaldag'i mos tushmagan satrlar uchun o'ng jadval ustunlaridagi qiymatlar o'miga NULL yoziladi.

**RIGHT [OUTER]** – O'ng tashqi tutashtirish turi. Bu tutashtirishda natijaviy jadvalga o'ng jadvalni barcha satrlari kiradi va chap jadvaldan faqat mos tushgan satrlar kiradi. O'ng jadvaldag'i mos tushmagan satrlar uchun chap jadval ustunlaridagi qiymatlar o'miga NULL yoziladi.

**FULL [OUTER]** – To'liq tashqi tutashtirish turi. Bu tur chap va o'ng tutashtirish turlar kombinatsiyasıdir. Bu tutashtirishda natijaviy jadvalga ikkala jadvalning barcha satrlari kiradi, shunda mos satrlar uchun ustunlar aniq qiymatlar bilan to'ldiriladi, mos kelmagan satrlar uchun ustunlardagi qiymatlar chap va o'ng ulanish qoidasi bo'yicha to'ldiriladi.

**UNION JOIN** – Tutashtirish ichki tutashtirishga teskaridir. Bu tutashtirishda natijaviy jadvalga mos tushmagan satrlar kiradi. Aniqrog'i bu tutashtirish natijasida, to'liq tashqi tutashtirish natijasida hosil bo'lgan jadvaldan mos tushgan satrlarni olib tashlashdan qolgan jadvaldan iborat.

Misol taraqasida talaba va prikaz jadvallarni tutashtirish usullarini ko'rsatamiz:

- 1) `SELECT * FROM talaba t JOIN prikaz p ON t.id_talaba=p.id_talaba`
- 2) `SELECT * FROM talaba JOIN prikaz USING id_talaba`
- 3) `SELECT * FROM talaba NATURAL JOIN prikaz`

### Ichki va tashqi tutashtirish usullari

Bizga ikkita jadval berilgan bo'lsin:

**A jadvali**

<b>Id_A</b>	<b>Name_A</b>
1	A
2	B
3	C
4	D

**B jadvali**

<b>Id_B</b>	<b>Id_A</b>	<b>Name_B</b>
1	1	A1
2	1	A2
3	1	A3
4	2	B1
5	2	B2
6	5	E1
7	5	E2

**Ichki** tutashtirish quyidagicha yoziladi:

```

SELECT A*, B.*
FROM A INNER JOIN B ON A.id_A=B.id_A

```

Shunda natijaviy jadval quyidagicha bo'ladi:

<b>Id_A</b>	<b>Name_A</b>	<b>Id_B</b>	<b>B.id_A</b>	<b>Name_B</b>
1	A	1	1	A1
1	A	2	1	A2
1	A	3	1	A3
2	B	4	2	B1
2	B	5	2	B2

Agar natijaviy jadvalga boshqa jadvalga kirmagan satrlarni qo'shish kerak bo'lsa, unda tashqi ularishni qo'llamoq kerak. Tashqi ularish uch xil turda bo'lishi mumkin: chap, o'ng va to'liq ularishlar. Chap ularish quyidagicha:

```

SELECT A*, B.*
FROM A LEFT JOIN B ON A.id_A=B.id_A

```

Shunda natijaviy jadval quyidagicha bo'ladi:

<b>Id_A</b>	<b>Name_A</b>	<b>Id_B</b>	<b>B.id_A</b>	<b>Name_B</b>
1	A	1	1	A1
1	A	2	1	A2
1	A	3	1	A3
2	B	4	2	B1
2	B	5	2	B2
3	C	NULL	NULL	NULL
3	D	NULL	NULL	NULL

O'ng ularish quyidagicha:

```

SELECT A*, B.*
FROM A RIGHT JOIN B ON A.id_A=B.id_A

```

Shunda natijaviy jadval quyidagicha bo'ladi:

<b>Id_A</b>	<b>Name_A</b>	<b>Id_B</b>	<b>B.id_A</b>	<b>Name_B</b>
1	A	1	1	A1
1	A	2	1	A2
1	A	3	1	A3
2	B	4	2	B1
2	B	5	2	B2
NULL	NULL	6	5	E1
NULL	NULL	7	5	E2

To'liq ulanish quyidagicha:

```
SELECT A*, B.*  
FROM A FULL JOIN B ON A.id_A=B.id_A
```

Shunda natijaviy jadval quyidagicha bo'ladi:

Id_A	Name_A	Id_B	B.id_A	Name_B
1	A	1	1	A1
1	A	2	1	A2
1	A	3	1	A3
2	B	4	2	B1
2	B	5	2	B2
3	C	NULL	NULL	NULL
4	D	NULL	NULL	NULL
NULL	NULL	6	5	E1
NULL	NULL	7	5	E2

Tutashtirish quyidagicha amalgam oshiriladi:

```
SELECT A*, B.*  
FROM A UNION JOIN B
```

Shunda natijaviy jadval quyidagicha bo'ladi:

Id_A	Name_A	Id_B	B.id_A	Name_B
3	C	NULL	NULL	NULL
4	D	NULL	NULL	NULL
NULL	NULL	6	5	E1
NULL	NULL	7	5	E2

### Xulosa

SQL tilida bir nechta jadvallarni bir xil ustunlar bo'yicha tutashtirish mumkin, bu orqali tuzilgan so'rovlar ko'p jadvalli murakkab so'rovlar jumlasiga kiradi.

Jadvallarni tutashtirish uch xil bo'ladi: o'zaro kesishilgan tutashtirish, ichki tutashtirish va tashqi tutashtirish.

O'zaro kesishilgan tutashtirishda birinchi jadvalning har bir satri ikkinchi jadvalning har bir satri bilan tutashtiriladi.

Ichki tutashuv natijasidagi jadvalda faqat mos tushgan satrlar qoladi.

- Tashqi tutashish uch xilga bo'linadi – tashqi chap, tashqi o'ng va to'liq tutashish.
- Tashqi chap tutashishda chap tomonda turgan jadval satrlari natijaviy jadvalga to'liq kiradi, o'ng tomondagi jadvalda mos satr mavjud bo'lmasa, unga tegishli ustunlar Null bilan to'ldiriladi.
- Tashqi o'ng tutashish natijasi tashqi chap tutashishga nisbatan aksincha bajariladi.
- To'liq tutashishda ikkala jadvalning barcha satrlari natijaviy jadvalda qatnashadi.

### **Nazorat uchun savollar va amaliyat uchun topshiriqlar**

1. Ko'p jadvallik so'rovlar nimadan iborat?
2. CROSS JOIN konstruksiysi nimani bildiradi?
3. INNER JOIN konstruksiysi qanday tutashuvlar uchun ishlataladi?
4. Qaysi konstruksiyalar orqali chap, o'ng va to'liq tutashuvlar amalga oshiriladi?
5. UNION JOIN konstruksiysi qanday tutashuvni tavsiflaydi?
6. Keltirilgan jadvaldan talabaning familiyasi, ismi sharifi, millati va tug'ilgan tumani chiqarilsin.
7. Keltirilgan jadvaldan talaba xususidagi berilganlar to'liq chiqarilsin.

## 27-bob. BERILGANLAR BAZASI USTIDAGI ISHLOVLAR

**Tayanch iboralar:** *INSERT, UPDATE, DELETE, tasavvur, INSERT INTO, VALUES, NULL.*

Jadvaldagi berilgantarni o'zgartirish uchun uchta operatorni qo'llash mumkin:

**INSERT** – jadvalga yangi berilganlarni qo'shish;

**UPDATE** – jadvaldagi qiymatlarni o'zgartirish;

**DELETE** – jadvaldagi satrlarni o'chirish.

### **Yangi berilganlarni jadvalga qo'shish**

**INSERT** operatorini ikkita shakli bor. Birinchisi shakl faqat **bitta satrni** qo'shish uchun mo'ljallangan. Bu shaklni sintaksisi quyidagicha:

*<satr qo'shish operatori> ::= INSERT INTO <jadval nomi>  
[(<ustunlar ro'yxati>)] VALUES (<qiymatlar ro'yxati>)*

Bu yerda, *<jadval nomi>* bazadagi jadval yoki yangilanuvchi tasavvur (viewer) nomi, *<ustunlar ro'yxati>* vergul bilan ajratilgan ustunlar nomidan iborat. Agarda ustunlar ko'rsatilmasa, barcha ustunlarni qiymati beriladi deb hisoblanadi, ularning ketma-ketligi jadval tuzilishidagi ketma-ketlik kabi bo'ladi. Agarda ro'yxat ko'rsatilsa, unga kirmagan ustunlarni qiymati **NULL**ga yoki jadval tuzishda berilgan oshkorras qiymatga teng bo'ladi.

*<Qiymatlar ro'yxati> <ustunlar ro'yxatiga>* quyidagicha **mos** kelishi kerak:

- ikkala ro'yxatdagi elementlar soni bir xil bo'lishi;
- qiymatlar mos ustunlarga aniq tushishi;
- berilganlar turi ustunlar turiga to'g'ri kelishi.

**Misol :**

*INSERT INTO talaba(familiya, ismi, sharifi,  
tug\_yil, jinsi)*

**VALUES ('Botirov', 'Rustam', 'Anvarovich', 1954, 'erkak')**

Operatorning ikkinchi shakli boshqa jadvaldan yoki so'rov qismidan satrlarni nusxalash uchun qo'llanadi:

**<nusxalash operatori> ::= INSERT INTO <jadval nomi> [<ustunlar ro'yxati>] <select ifodasi>**

**Misol:**

I kursga kirgan abituriyentlar talaba jadvaliga qo'shilsin

```
INSERT INTO talaba  
SELECT * FROM talabайлkurs
```

**Berilganlarni o'zgartirish**

Berilganlarni o'zgartirish uchun UPDATE operatori ishlataladi, uning sintaksisi quyidagicha:

**<o'zgartirish operatori> ::= UPDATE <jadval nomi> SET {<ustun nomi> = <qiymat>} [, ...] [<from bo'limi>] [<where bo'limi>]**

Bu yerdagi ustun nomi, o'zgartirish kiritilayotgan jadvalning ustunlaridan iborat. Qiymat bu skalyar ifoda bo'sib, yangi qiymatni bildiradi, bu ifodada o'zgarmaslardan tashqari ustunlarning nomi ham qatnashishi mumkin. Agarda **<from bo'limi>** mavjud bo'lsa, u holda qiymat ifodasida **FROM** konstruksiyasida ko'rsatilgan jadvallarning ustunlari qatnashishi mumkin. Agarda **WHERE** konstruksiysi bo'lnasa, barcha satrlarni qiymatlari o'zgaradi, aks holda tanlangan satrlar ustunlari o'zgaradi.

**Misollar:**

1. Talabalarni stipendiyasi 1,2 martaga oshirilsin:

```
UPDATE talaba  
SET stipendiya = stipendiya * 1.2
```

2. 1980-yildan oldin tug'ilgan talabalarning vazni 1,5 martaga oshirilsin:

```
UPDATE talaba  
SET ogirligi = ogirligi * 1.5  
WHERE tug_yil < 1980
```

3. O'qishga kirgan talabalarning tug'ilgan shahri talaba 1 kurs jadvaldan olinsin:

```
UPDATE talaba
    SET tug_shahri=a.tug_shahri
    FROM talaba t INNER JOIN talabalkurs a ON
        t.id_talaba=a.id_talaba
```

4. 1 raqamli fakultetdagi, 2-kurs rus guruhi talabalarining stipendiyasi hisoblansin.

```
UPDATE talaba
    SET stipendiya=70000*p.stip_turi
    FROM talaba t INNER JOIN prikaz p
    ON t.id_talaba=p.id_talaba
    WHERE p.id_fak=1 AND p.kurs=2 AND
        p.yaz='rus' AND ykuv_yil=2011
```

### Satrlarni o'chirish

Satrlarni o'chirish uchun quyidagi konstruksiya ishlataladi:

```
<o'chirish operator> ::= DELETE [FROM] <jadval nomi>
    [<from bo'limi>] [where bo'limi]
```

**WHERE** konstruksiyasida o'chirish kerak bo'lgan satrlar tanlanadi. Agar **<from bo'limi>** ko'rsatilsa, u holda shu konstruksiyada ko'rsatilgan jadvallarning satrlariga mos bo'lgan satrlar o'chiriladi.

#### Misollar:

a) 1960-yildan avval tug'ilgan talabalar o'chirilsin.

```
DELETE talaba
```

```
    WHERE tug_yil<1960
```

b) Toshkent viloyatidagi shaharlardan kelgan talabalar o'chirilsin.

```
DELETE FROM talaba
```

```
    WHERE tug_shahar IN
        (SELECT shahar_nomi
        FROM shaharlar
        WHERE vil_nomi='Tashkent viloyati');
```

c) 1-raqamli fakultetning, 4-kurs rus guruhi talabalari o'chirilsin.

```
DELETE talaba
```

```
    FROM talaba t INNER JOIN prikaz p ON
        t.id_talaba=p.id_talaba
    WHERE p.id_fak=1 AND p.kurs=4 AND p.yaz='rus'
        AND ykuv_yil=2011
```

## SQL tilini relatsion to'liqligi

SQL tilini relatsion to'liqligini isbotlash uchun relatsion amali SQL operatori orqali yozish imkoniyati mavjud bo'lishi kerak. Quyidagi operatorlarda shu imkoniyatlar keltirilgan.

### Dekart ko'paytmasi amali

Relatsion algebrada: A TIMES B

SQL operatori:

```
SELECT A.* , B.*  
      FROM A, B;  
  
yoki  
SELECT A.* , B.*  
      FROM A CROSS JOIN B;
```

### Nusxalash amali

Relatsion algebrada: A[X,Y,...,Z]

SQL operatori:

```
SELECT DISTINCT X, Y, ..., Z  
      FROM A;
```

### Tantash amali

Relatsion algebrada: A WHERE c

SQL operatori:

```
SELECT *  
      FROM A  
    WHERE c;
```

### Birlashtirish amali

Relatsion algebrada: A UNION B

SQL operatori:

```
SELECT *  
      FROM A  
UNION  
SELECT *  
      FROM B;
```

### Ayirish amali

Relatsion algebrada: A MINUS B

SQL operatori:

```
SELECT *  
      FROM A  
EXCEPT
```

```
SELECT *
FROM B;
```

### *Tutashtirish amali*

Relatsion algebrada: (A TIMES B) WHERE c

SQL operatori:

```
SELECT A.* , B.*
FROM A, B
WHERE c;
```

yoki

```
SELECT A.* , B.*
FROM A CROSS JOIN B
WHERE c;
```

Boshqa turdag'i tutashtirishlarni **biz yugorida ko'rsatganimiz.**

### *Kesishish amali*

Relatsion algebrada: A INTERSECT B

SQL operatori:

```
SELECT *
FROM A
INTERSECT
SELECT *
FROM B;
```

### *Bo'lish amali*

Relatsion algebrada: A(X,Y) DEVIDE BY B(Y)

SQL operatori:

```
SELECT DISTINCT A.X
FROM A
WHERE NOT EXISTS
(SELECT *
FROM B
WHERE NOT EXISTS
(SELECT *
FROM A A1
WHERE A1.X=A.X AND A1.Y=B.Y));
```

## Xulosə

- Agar natijaviy jadvalni uestunlari bir nechta manba jadvallardan olinsa, bu jadvallarni tutashtirish kerak, buning uchun ular FROM konstruksiyasida berilishi kerak va ularni ularni uestashda JOIN konstruksiyasi qo'llanishi kerak.
- Jadvallarni to'plam nuqtayi nazarda birlashtirish kesilish va ayrish uchun UNION, INTERSECT va EXCEPT kalit so'zlar ishlataladi.
- Jadvallar uestidan ishlovlar bajarish uchun uchta operator kiritilgan.
- INSERT operatori jadvalga yangi satrlar qo'shish uchun ishlataladi.
- UPDATE operatori satr uestunlarini qiymatlarini o'zgartirish uchun qo'llanadi.
- DELETE operatori berilgan jadvaldan satrlarni olib tashlash uchun ishlataladi.

## Nazorat uchun savollar va amaliyot uchun topshiriqlar

SQL tili relatsion to'liqligini isbotlang.

## 28-bob. BERILGANLARNI TAVSIFFLASH TILI

**Tayanch iboralar:** *Identifikator, berilganlar turi, chegaralar, domenlar, skalyar funksiyalar, CREATE DOMAIN, ALTER DOMAIN, DROP DOMAIN, CREATE ASSERTION, DROP ASSERTION, PRIMARY KEY, FOREIGN KEY, UNIQUE, CONSTRAINT, CHECK*

Berilganlarni tavsifflash tili jadvallar, chizmalar, domenlar, ko'rsatmalar va indekslar yaratish, o'chirish hamda o'zgartirish imkoniyatini beradi. Undan tashqari jadvallarga alohida chegaralarni qo'yish operatorlari ham mavjud. DDL tili quyidagi operatorlardan iborat:

- **CREATE SCHEMA** – berilganlar bazasini chizmasini tuzish;
- **DROP SCHEMA** – berilganlar bazasini chizmasini o'chirish;
- **CREATE TABLE** – jadval yaratish;
- **ALTER TABLE** – jadvalni o'zgartirish;
- **DROP TABLE** – jadvalni olib tashlash;
- **CREATE DOMAIN** – domen yaratish;
- **ALTER DOMAIN** – domenni o'zgartirish;
- **DROP DOMAIN** – domenni olib tashlash;
- **CREATE INDEX** – indeks yaratish;
- **ALTER INDEX** – indeksni o'zgartirish;
- **DROP INDEX** – indeksni olib tashlash;
- **CREATE VIEW** – tasavvurni yaratish;
- **DROP VIEW** – tasavvurni o'chirish;
- **CREATE ASSERTION** – chegarani yaratish;
- **DROP ASSERTION** – chegarani olib tashlash;
- **COMMIT** – tranzaksiyani bajarishni tugatish;
- **ROLLBACK** – tranzaksiyani bajarishni qaytarish;
- **SET TRANSACTION** – tranzaksiyaga izolyatsiya darajasini berish;
- **GRANT** – foydalanuvchiga ishlovda ustivorlikni yaratish;
- **REVOKE** – ustivorlikni man etish.

Har bir til elementar tushunchalardan boshlangani kabi, biz ham DDL tilini o'rganishni leksemalaridan boshlaymiz.

## **Identifikator**

SQL tilidagi identifikatorlar berilganlar bazasidagi obyektlarni belgilash uchun ishlataladi. Jadvallar, tasavvurlar va ustunlar nomlari identifikatorlar orqali yoziladi. Identifikator harflar, raqamlar va «\_» tag chiziq belgilaridan iborat bo'lib, bundan tashqari kirill harflarini ham ishlatish mumkin. Identifikatorlarni yozishda bir nechta cheklashlar mavjud:

- identifikatorning uzunligi 128ta harfdan ortiq bo'lishi mumkin emas;
- identifikator qisqatta harfdan oshishishni kerak;
- identifikatororda probel ishlatish mumkin emas.

Lekin SQL tilining ba'zi bir naqlari uchun bu qoidalar mos kelmasligi mumkin, masalan ASSESSda identifikator raqamdan boshlanishi ham mumkin, probellar ham bo'lishi mumkin, u holda identifikator kvadrat qavsga ([ ]) olinadi.

## **Berilganlar turlari**

ISO namunasida berilganlar turi quyidagicha **aniqlangan**:

**BOOLEAN** – mantiq turi (boolean);

**CHAR , VARCHAR** – belgilar turi (char);

**BIT, BIT VARING** – bitlar turi (bit);

**NUMERIC, DECIMAL, INTEGER, SMALLINT** – fiksirlangan nuqtali son;

**FLOAT, REAL, DOUBLE PRECISION** – suzuvchi nuqtali sonlar turi;

**DATE, TIME, TIMESTAMP** – sana va vaqt turi;

**CHLOB, BLOB** – obyekt berilganlar turi.

### **Mantiq turidagi berilganlar (BOOLEAN)**

Mantiq turidagi berilganlar, quyidagi qiymatdan iborat **TRUE, FALSE** va **NONE**.

Mantiqiy qiymatlar solishtirish va mantiq amallarida hamda ustunlar qiymati sifatida ishlatalishi mumkin. Uning sintaksisi quyidagicha:

**<mantiq turi> ::= BOOLEAN**

### **Belgilar (CHARACTER) turidagi berilganlar**

Belgilar turidagi berilganlar ketma-ket yozilgan belgilardan **iborat**. Bunda berilganlar quyidagicha berilishi mumkin:

**<Belgilar turi> ::= {CHARACTER [VARYING] | CHAR |  
VARCHAR} [<uzunligi>]**

Bu yerda, <uzunlik> ketma-ketlikdag'i maksimal uzunlikni ko'rsatadi, oshqarmas holda uzunlik birga teng. Belgilar satri fiksirlangan bo'lishi mumkin (CHAR) yoki o'zgaruvchi uzuntikda bo'lishi mumkin (VARCHAR). Agar ustun fiksirlangan satr bo'lsa, uning qiymatini kiritishdag'i bo'sh qolgan joylar probellar bilan to'ldirilmaydi, ya'ni o'zgaruvchi ustunida probellar yozilmaydi, bu esa xotira hajmini kamaytirishga olib keladi. Masalan, pasport seriyasi har doim ikkita harf bilan belgilanadi, shuning uchun bu ustunni quyidagicha tavsiflash mumkin:

**pas\_seriya CHAR(2);**

Manzil ustunini tavsiflash quyidagicha bo'ladi:

**adress VARCHAR (100)**

Satrning maksimal uzunligi tilning naqliga bog'liq. Masalan, MS SQL SERVERda satrn uzunligi har xil turlar uchun 4000 dan  $2^{31}$  gacha bo'lishi mumkin.

#### ***Berilganlarning bit (BIT) turi***

Bit turidagi berilganlar 0 va 1 raqamlardan iborat bo'lgan satr bilan aniqlanadi. Bu tur tavsifi quyidagicha:

**<bit turi> ::= BIT | VARYING [<uzunligi>]**

Misol:

**Bitstr BIT(4)**

#### ***Fiksirlangan aniq qiymatli berilganlar turi***

Fiksirlangan aniq sonlar kompyuterda aniq ko'rinishga ega, bunday sonlar raqamlardan va «.-» belgilardan iboratdir. Sonlar, kasr qism uzunligi va umumiy raqamlar soni bilan aniqlanadi. Masalan, -12.345da raqamlar umumiy soni 5ga, kasr qismi 3ga teng. Kasr qismi Oga teng bo'lgan son, butun son deb nomlanadi. Aniq sonlarni tavsiflashning bir nechta usullari bor:

**<fiksirlangan son turi> ::= {NUMERIC | DECIMAL | DEC}  
[<umumiy soni> [, <kasr qismi>]] | {INTEGER | INT} | SMALLINT**

Bu turlar kasr qism mavjudligi hamda sonni chegarasi bilan farq qiladi. **NUMERIC**, **DECIMAL** va **DEC** ekvivalentdir va bu turdag'i sonlar chegarasi  $-10^{38}$  dan  $+10^{38}$  gacha kasr qismiga ega bo'lgan sonlardan iborat. Raqamlar soni SQL naqliga bog'liq. Oshkormas ravishda kasr qismini nolga teng. **INTEGER** yoki **INT** turlar ishorali butun sonlar uchun ishlataladi, bunday sonlar xotiradan 4 bayt egallab,  $-2^{31}$  dan  $+2^{31}$  gacha chegarada bo'lishi mumkin. **SMALLINT** tur ishorali qisqa butun sonlar uchun ishlataladi, bunday sonlar xotiradan 2 bayt egallab  $-2^{15}$  dan  $+2^{15}$  gacha chegarada bo'lishi mumkin.

Misollar:

```
stipendiya DECIMAL (10,2)
tug_yil SMALLINT
```

### *Suzuvchi nuqtali haqiqiy sonlar turi*

Haqiqiy sonlar turi taxminiy qiymatga ega bo'lgan berilganlar va o'z-garmaslar uchun ishlataladi. Bunday sonlarni yozish uchun darajalik shakl ishlatalishi mumkin, masalan  $5.25E-6$ . Haqiqiy son turining sintaksisi quyidagicha:

```
<haqiqiy son turi> ::= {FLOAT | REAL | DOUBLE PRECISION}
[(<mantissa soni>)]
```

Bunda **<mantissa soni>** mantissadagi raqamlar sonidan iborat.

**FLOAT** va **REAL** xotirada 4 bayt joy egallab,  $\pm 3.4e-38$  dan  $\pm 3.4e+38$  gacha chegarada bo'lishi mumkin, **DOUBLE PRECISION** xotirada 8 bayt joy egallab,  $\pm 1.7e-308$  dan  $\pm 1.7e+308$  gacha sonlar bo'ladi.

### *Sana va vaqt turidagi berilganlar*

Sana va vaqt turi ma'lum aniqlik bilan bir paytda berilgan vaqtini aniqlaydi.

Sana va vaqtini berish uchun quyidagi sintaksis bo'yicha uchta tur mavjud:

```
<sana va vaqt> ::= DATE | {TIME | T | ME STAMP} | <aniqlik>
[WITH TIME ZONE]
```

**DATE** turi sanani saqlash uchun ishlataladi. U uchta tur ostiga **YEAR**, **MONTH**, **DAY** bo'linishi mumkin. Ular yil, oy va kunni bildiradi.

**TIME** turi vaqtini ko'rsatish uchun ishlataladi. U ham uchta tur ostida bo'linishi mumkin, bular **HOUR**, **MINUTE** va **SECOND**, mos ravishda soat, daqiqa va soniyani bildiradi.

**TIME STAMP** turi sana va vaqtini birgalikda saqlashda ishlataladi.

Soniyaning qismini ko'rsatish uchun <aniqlik> parametri beriladi, agar ustun turi **TIME** bo'ssa, oshkormas ravishda 0ga, **TIME STAMP** turi uchun aniqlik 6ga teng, ya'ni vaqt nonasekundgacha berilishi mumkin. **WITH TIME ZONE** parametri grinich vaqtiga nisbatan siljishni aniqlaydi.

Masalan, talabani tug'ilgan sanasini saqlaydigan ustunni quyidagicha tasvirlash mumkin:

Datatug **DATE**

### Skalar amallar

SQL tilida skalar amallar va funksiyalar ishlatalishi mumkin. Skalar ifodani yozishda (+, -, \* va /) amallardan tashqari quyidagi amal va funksiyalarni ishlatalish mumkin:

**Amal yoki funksiya nomi**

**Bajaradigan ishi**

<b>CHAR_LENGTH</b>	satrni uzunligini aniqlaydigan funksiya
<b>CAST</b>	bir turdan boshqa turga o'tkazadigan funksiya
<b>  </b>	ikki satrni tutashtirish amali
<b>LOWER</b>	satrdagi katta harflarni kichik harflarga o'tkazadi
<b>UPPER</b>	satrdagi kichik harflarini katta harflarga o'tkazadi
<b>LTRIM</b>	satrdagi boshlang'ich probellarni olib tashlaydi
<b>RTRIM</b>	satr oxiridagi probellarni olib tashleydi
<b>POSITION</b>	bir satrni boshqa satrdagi joyini aniqlaydi
<b>SUBSTRING</b>	satrdan satr ostini ajratish uchun ishlataladi
<b>CASE</b>	bu amal, shart bo'yicha bir nechta qiymatni tanlash uchun ishlataladi.

**CASE** amalining sintaksisi quyidagicha :

<shartli amal> ::= **CASE** { **WHEN** <shartli ifoda> **THEN**  
<skalar ifoda>} ... [**ELSE** <skalar ifoda>] **END**

<b>CURRENT_DATE</b>	joriy sanani qaytaradi
<b>CURRENT_TIME</b>	joriy vaqtini qaytaradi
<b>CURRENT_TIMESTAMP</b>	joriy sana va vaqtini qaytaradi

Shuni aytib o'tish kerakki, turli naq'larda funksiyalar va **amallar**, boshqacha bo'lishi mumkin.

## Berilganlarni butunligini saqlash usullari

SQL namunasida berilganlar butunligini saqlash funksiyalarni ko'rib chiqamiz.

Bu funksiyalar butunliklarni saqlash va himoyalash uchun berilganlarning chegaralari orqali amalga oshiriladi. Umuman olganda, beshta chegaralar mavjud:

- majburiy berilganlar;
- domen chegaralar;
- mohiyat butunligi;
- havola butunligi;
- tashkilot qo'ygan chegaralar.

### Majburiy berilganlar

Ba'zi bir ustunlarning qiymatlari bo'sh bo'lmasligi talab qilinadi. Masalan, har bir talaba familiya va ismiga ega bo'lishi kerak. Bunday chegarani berish uchun jadvalni yaratish jarayonidagi ustunni tavsiflashda **NOT NULL** kalit so'zi ko'rsatiladi. Agarda ustun uchun bu kalit so'zi berilsa, tizim bunday ustunni qiymati bo'lmasligiga yo'l qo'ymaydi. Aks holda, ustun uchun **NULL** qiymati berilsa, bunday ustunda bo'sh qiymatlар mavjud bo'lishi mumkin, oshkormas ravishda **NULL** qiymatlari bo'lishi mumkin. Masalan, ustun tavsifi quyidagicha bo'lishi mumkin:

`fan VARCHAR(30) NOT NULL`

### Domen chegaralari

Har bir ustun o'z domeniga ega, ya'ni uning qiymatlari va chegarasi aniqlangan, masalan jinsini ko'rsatadigan ustunda faqat ikkita qiymat bo'lishi mumkin, bu «erkak» va «ayol». Namunada domenni tavsiflash uchun ikki usul mavjud. Birinchi usul bo'yicha **CHECK** konstruksiyasini qo'llashni ko'ramiz. Bu konstruksiya ustun yoki jadval uchun chegara shartlarini berish uchun ishlataladi.

Bu konstruksiyaning umumiyo ko'rinishi quyidagicha:

`<check chegarasi> ::= CHECK <chevara sharti>`

Chegara shartida shu ustunni nomini ko'rsatish mumkin. Bu chegara ustunga qo'llanadi, shu sababli bu chegarani *ustun chegarasi* deb nomlashtadi.

Masalan:

jinsi **CHAR NOT NULL CHECK** jinsi IN («erkak»,»ayol»)

yoki

jinsi **CHAR NOT NULL CHECK** jinsi=«erkak» **OR** jinsi=«ayol»

Ikkinci usul bo'yicha ustun uchun alohida domen yaratiladi. Bunday operatorning sintaksi quyidagicha:

<domen operatori> ::= **CREATE DOMAIN** <domen nomi> [AS]

<domen turi> [<oshkormas qiymat>] [<check chegarasi>]

<oshkormas qiymat> ::= **DEFAULT** <literal>

Bunda yangi domen yaratiladi va unga domen nomi beriladi. Bu ismni yangi tur sifatida ishlatalish mumkin, domen turi bu tayanch turlardan biridir, oshkormas qiymat berilmashligi mumkin, bu qiymat berilmaganda chegara ustunga qo'yiladi. Chegara sharti bu domenga qo'yilgan qo'shimcha shartdir. Chegaraviy qiymatlarni berish uchun ustun nomi o'rniga **VALUE** kalit so'zi ishlataladi.

Misol:

```
CREATE DOMAIN Jjinsi AS CHAR(5)
    DEFAULT 'erkak'
    CHECK VALUE IN ('erkak', 'ayol')
```

Kiritilgan yangi turni ustunni tavsiflashda ishlatalish mumkin:

jinsi Jjinsi NOT NULL

Domenni biror bir jadval asosida yaratish mumkin masalan shaharlar(id,shahar, maydon, aboli) berilgan bo'lsa:

```
CREATE DOMAIN shaht AS CHAR(20)
    CHECK VALUE IN (SELECT shahar FROM shaharlar)
```

Domenni parametrlarini o'zgartirish uchun quyidagi konstruksiya ishlataladi:

<domenni o'zgartirish> ::= **ALTER DOMAIN** <domen nomi>

{**SET DEFAULT** <literal>} | {**DROP DEFAULT**}  
| {**ADD** [**chevara nomi**] <check chegarasi> }

Bitta operator faqat bitta parametrni o'zgartiradi. Masalan, oshkormas qiymatni olib tashlash uchun operatorni quyidagicha yozish kerak:

**ALTER DOMAIN jinsi DROP DEFAULT**

Domenni olib tashlash uchun quyidagi operator ishlataladi:

*<domen olib tashlash>* ::= **DROP DOMAIN** *<domenni nomi>*  
[ **CASCADE** | **RESTRICT** ]

**RESTRICT** kalit so'zi ko'rsatilganda, domen biror bir jadvalda qo'llangan bo'lsa, domen olib tashlanmaydi. **CASCADE** kalit so'zi ko'rsatilganda, domenni olib tashlashdan oldin, domenga bog'liq bo'lган jadvallarda quyidagilar bajariladi:

- domen turi, domenga asoslangan ustunlarga uzatiladi;
- agar domenda kelishuv bo'yicha qiymat ko'rsatilgan bo'lsa, u holda kelishuv bo'yicha qiymat mos ustunlarga qo'yiladi;
- domen chegaralari ustunlar chegaralari sisatida qoladi.

### Mohiyat butunligi

Jadvalni birlamchi kaliti har bir satrda noyob bo'sh bo'lmas qiymatga ega bo'lishi kerak. Masalan, student jadvalida bu *id\_stud* ustuni bo'ladi, chunki u hat bir satrda noyobdir. SQL tilini namunasida bunday mohiyat butunligi talabi **CREATE TABLE** yoki **ALTER TABLE** operatorlarning **PRIMARY KEY** konstruksiyasi orqali amalga oshiriladi. Uning sintaksisi quyidagicha:

*<birlamchi kalit>* ::= **PRIMARY KEY** (*<ustunlar ro'yxati>*)  
*<ustunlar ro'yxati>* ::= *<ustun nomi>* [, ...]

Masalan:

**PRIMARY KEY** (*id\_stud*)

SQL yo'riqnomasida bu konstruksiyani faqat bir marta ko'rsatish mumkin, lekin muqobil kalitlarni ham ko'rsatish imkoniyati yo'q emas. Buning uchun, **UNIQUE** kalit so'zini **NOT NULL** bilan birlgilikda ishlatisch kerak:

*<muqobil kalit>* ::= **UNIQUE** (*<ustunlar ro'yxati>*)

Masalan:

```
pas_ser CHAR(2) NOT NULL,  
pas_nom CHAR(8) NOT NULL,  
UNIQUE (pas_ser, pas_nom);
```

Bu konstruksiyani bitta jadval uchun bir nechta marta ishlatalish mumkin.

### Havola butunligi

Ma'lumki, tashqi kalitlar avlod jadvalning satrlarini ajdod jadvalni satrлari bilan bog'lash uchun ishlataladi. Havola butunligining ma'nosi shundaki, agarda tashqi kalit qiymatga ega bo'lsa, u albatta ajdod jadvaldagi mavjud bo'lgan satrga havola qilishi kerak. Masalan, student jadvalining id\_fak ustuni fakultet jadvalining mos bo'lgan qiymatiga havola qilishi kerak. Agar bu ustun student jadvalida bo'sh bo'lmasa, u albatta biror bir fakultet jadvalidagi satrga havola qilishi shart. SQL tilining namunasida tashqi kalit so'zlarni ko'rsatish uchun **CREATE TABLE** yoki **ALTER TABLE** operatorlarning **FOREIGN KEY** konstruksiyasi ishlataladi, uning formati quyidagicha:

```
<tashqi kalit qismi> ::= FOREIGN KEY (<ustunlar ro'yxati>
    REFERENCES <ajdod jadval nomi>
    {<tashqi ustunlar ro'yxati>} [MATCH {PARTIAL | FULL}])
```

```
<tashqi ustunlar ro'yxati> ::= <ustunlar ro'yxati>
```

bunda *<ustunlar ro'yxati>* – avlod jadvalning tashqi kalit sifatidagi ustunlar nomidir; *<tashqi ustunlar ro'yxati>* – ajdod jadvalning birlamchi kalit sifatidagi ustunlar nomi. Agar bu ustunlar ko'rsatilmasa, tashqi kalit ajdod jadvalning birlamchi kalitiga mos kelishi kerak. Aks holda, tashqi kalit ro'yxatidagi usturilami hamma qiymatlari NULL bo'lishi kerak yoki hammasi qiymatga ega bo'lishi kerak. Agar MATCH PARTIAL berilsa, unda barchasi bo'sh bo'lishi kerak yoki qisman bo'sh bo'lishi mumkin. Bu holda ajdod jadvaldagi bo'shmas ustun uchun mos satr mavjud bo'lishi kerak, ya'ni bir avlod satriga bir nechta ajdod satrlari to'g'ri kelib qoladi, ya'ni havola butunligi buziladi. Bunday vaziyatlarda asosan MATCH FULL q'llanadi.

Masalan:

```
FOREIGN KEY (id_fak) REFERENCES fakultet
```

Endi, talaba jadvalining id\_fak ustuniga yangi qiymat kirtsak yoki uni o'zgartirsak va bu qiymat fakultet jadvalida mavjud bo'lmasa, u holda

tizim amalni bajarishga ruxsat bermaydi. Agar ajdod jadvalda tashqi kalitga mos kelgan nomzod kalitni o'zgartirish yoki o'chirish harakatini qilsak, unda natija kiritilgan havola butunligini saqlash qoidasiga bog'liq bo'ladi.

Bu qoidalar **FOREIGN KEY** konstruksiyasida quyidagicha beriladi:

*<tashqi kalit> ::= <tashqi kalit qismi> [ON UPDATE <saqlash qoidasi>] | [ON DELETE <saqlash qoidasi>]  
<saqlash qoidasi> ::= {CASCADE | SET NULL | SET DEFAULT | NO ACTION}*

Agar ajdod jadvalida satni o'chirish (ON DELETE) harakati qilinganda, SQL tilida quyidagi holatlardan biri yuz berishi mumkin:

- **CASCADE** – ajdod jadvaldagisi satrlari o'chiriladi, avlod jadvaldagisi shu satrlarga havola qilgan satrlari ham avtomatik ravishda o'chiriladi. Vaholanki, avlod jadvali ham boshqa jadval uchun ajdod bo'lishi mumkin, u holda avlod jadvalida o'chirilgan satrlarga havola qilgan boshqa jadvalni satrлари ham o'chiriladi (kaskad shaklida);
- **SET NULL** – ajdod jadvaldagisi satrlari o'chiriladi, avlod jadvaldagisi shu satrlarga havola qilgan tashqi kalitni qiymati NULLga almashtiriladi. Bu usulni tashqi kalit NOT NULL xususiyatiga ega bo'lmagandagina qo'llash mumkin.
- **SET DEFAULT** – ajdod jadvalida satrlari o'chiriladi, avlod jadvaldagisi shu satrlarga havola qilgan tashqi kalit ustunning qiymati kelishuv bo'yicha olingan qiymatga almashtiriladi. Bu usulni tashqi kalit SET DEFAULT xususiyatiga ega bo'lsa va unda kelishuv qiymat berilgan bo'lsa, qo'llash mumkin.
- **NO ACTION** – agarda avlod jadvaldagisi tashqi kalit ajdod jadvalga biror bir satriga havola qilsa, ajdod jadvalidagi satni o'chirish man etiladi. Bu qoida kelishuv bo'yicha o'matiladi.

Xuddi shunday qoidalar berilganlarning o'zgartirishda ham qo'llanadi (ON UPDATE):

- **CASCADE** – ajdod jadvaldagisi nomzod kalitni qiymati o'zgartiriladi, avlod jadvaldagisi shu satrlarga havola qilgan satrlarning tashqi kalitdagisi qiymati ham avtomatik ravishda nomzod kalitni yangi qiymatiga o'zgaradi. Avlod jadvali ham boshqa jadval uchun ajdod bo'lishi mumkin, u holda, agarda tashqi kalit avlod jadvalni nomzod kalit tarkibiga kirgan bo'lsa, avlod jadvalida o'zgartirilgan

satrlarga havola qilgan boshqa jadvalning satrlari ham o'zgartiriladi (kaskad shaklida).

- **SET NULL** – ajdod jadvaldag'i nomzod kalitni qiymati o'zgartiriladi, avlod jadvaldag'i shu satrlarga havola qilgan tashqi kalit qiymati NULLga almashtiriladi. Bu usulni tashqi kalit NOT NULL xususiyatiga ega bo'limgandagina qo'llash mumkin.
- **SET DEFAULT** – ajdod jadvalida nomzod kalitni qiymati o'zgartiriladi, avlod jadvaldag'i shu satrlarga havola qilgan tashqi kalitni qiymati kelishuv bo'yicha olingan qiymatga almashtiriladi. Bu usulni, tashqi kalit SET DEFAULT xususiyatiga ega bo'lib, unda kelishuv qiymat berilgan bo'sagina qo'llash mumkin.
- **NO ACTION** – agarda avlod jadvaldag'i tashqi kalit ajdod jadvalga biror bir satriga havola qilsa, ajdod jadvalidagi nomzod kalitni o'zgartirish man etiladi.

Misol:

```
FOREIGN KEY (id_fak) REFERENCES fakultet  
    ON DELETE SET NULL  
    ON UPDATE CASCADE;
```

#### Tashkilot qo'ygan chegaralar

Berilganlami yangilashda tashkilot tomonidan chegaralar qo'yilishi mumkin. Shu jumladan tijorat qoidalari, ish tartibi va h.k. Bunday chegaralarni CHECK va UNIQUE konstruksiylar orqali berish mumkin. Agar chegara bevosita jadvalni tavsiflashga bog'liq bo'lsama CREATE ASSERTION operatori qo'llash mumkin, uning sintaksisi quyidagicha:

*<chegaralash operatori> ::= CREATE ASSERTION <chegaralash nomi> CHECK (<chegara sharti>)*

Masalan, predmet sohasidan kelib chiqqan holda, har bir kursda talabalar soni 50 dan oshmasligi kerak bo'lsin, bunda chegaralash operatori quyidagicha bo'ladi:

```
CREATE ASSERTION guruh CHECK(NOT EXISTS  
    (SELECT kurs  
        FROM PRIKAZ GROUP BY kurs  
        HAVING COUNT(*)>50));
```

Bakalavr ta'lif shaklidagi har bir talaba **4 yildan ortiq o'qimasligi kerak** bo'lsin:

**CREATE ASSERTION kurs**

```
CHECK (NOT EXISTS (SELECT *
FROM PRIKAZ WHERE kurs>4));
```

### Xulosa

SQL tilining namunasida berilganlarning sakkizta tayanch turi kiritilgan:

- Mantiq, belgi va bit satrlari, aniq va haqiqiy sonlar, sana va vaqt, vaqt oraliqi, ikkilik obyektlar (surat, media va boshqalar).
- SQL namunasi bo'yicha SQL tilida berilganlarni butunligini saqlab turish uchun CREATE TABLE va ALTER TABLE operatorlar ichiga maxsus konstruksiyalar kiritilgan, bularga ushbular kiradi – berilganlarni qiyatlari majburiyligi; domenlarga chegaralar; havola butunligini saqlash; mohiyat butunligini saqlash va foydalanuvchi tomonidan qo'yilgan chegaralar.
- Berilganlarni majburiyligi NOT NULL kalit so'zi bilan beriladi.
- Domen chegaralari CHECK konstruksiya bilan yoki CREATE DOMAIN operator orqali domen yaratish bilan beriladi.
- Birlamchi kalit PRIMARY KEY konstruksiya bilan aniqlanadi, tashqi qalitlar FOREIGN KEY konstruksiya orqali beriladi ularni olib tashlash yoki o'zgartirish qoidalari ON UPDATE va ON DELETE konstruksiyalar orqali beriladi.
- Foydalanuvchi chegaralari alohida CREATE ASSERTION operatori bilan berilishi mumkin.

### Nazorat uchun savollar va amaliyot uchun topshiriqlar

1. Berilganlarni butunligini saqlab turish uchun qaysi imkoniyatlar mavjud?
2. Ustunlarni majburligi qaysi konstruksiya orqali amalga oshiriladi?
3. Domenlarni chegarasi berilishini ko'rsating
4. Havola butunligini saqlash konstruksiyani keltiring.
5. Foydalanuvchi chegaralariga misol keltiring.

## 29-boB. BERILGANLAR BAZASINI YARATISH

**Tayanch iboralar:** Berilganlar bazasi, chizma, jadval, indeks, CREATE SCHEMA, DROP SCHEMA, CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE INDEX, ALTER INDEX, DROP INDEX, tasavvur, aniqlovchi tasavvur, ishlovli tasavvurlar, CREATE VIEW, DROP VIEW, CREATE ASSERTION, DROP ASSERTION.

SQL tilini namunasida berilganlar bazasini yaratish operatorlari yo'q, har bir naql o'z usullariga ega. Masalan ACCESS BBBTda har bir berilgan bazasi alohida .mdb kengaytmali faylni tashkil qiladi. MS SQL SERVERda ham .mdf kengaytmali fayl bo'ladi, undan tashqari, bu BBBTda CREATE DATABASE operatori yordamida berilganlar bazasini yaratish mumkin, masalan:

```
CREATE DATABASE TalabaS ON
    (NAME=talaba_dat
     FILENAME='c:\st\StbaseSQL.mdf'
     SIZE=10
     MAXSIZE=90
     FILEGROWTH=5)
LOG ON
    (NAME=talaba_log
     FILENAME='c:\st\StbaseSQL.ldf'
     SIZE=5
     MAXSIZE=25
     FILEGROWTH=5)
```

Berilganlar bazasini konseptual modelini tuzish uchun chizmalar ishlataladi. Chizmalar CREATE SCHEMA operator orqali beriladi, uning sintaksisi quyidagicha:

<chizma yaratish operatori> ::= CREATE SCHEMA [<chizma nomi>]

## AUTORIZATION <baza egasi>

Misol:

```
CREATE SCHEMA talaba AUTORIZATION sa
```

Bu operatordan keyin chizmaga kirgan barcha obyektlarni yaratish operatorlari keltiriladi.

Chizmani olib tashlash uchun quyidagi operator qo'llanadi:

```
<chizmani olib tashlash> ::= DROP SCHEMA [<chizma nomi>]  
[CASCADE | RESTRICT ]
```

Agar RESTRICT ko'rsatilsa, faqat chizma bo'sh bo'lsa, olib tashlanadi. CASCADE ko'rsatilsa, shu chizma bilan bog'langan barcha jadvallar va boshqa obyektlar olib tashlanadi, keyin esa chizma olib tashlanadi. Shuni aytib o'tish kerakki, ko'p BBB tizimlarida bunday operatorlar mavjud emas va bu ishni alohida aloqa orqali bajariladi.

### Jadval yaratish

Jadval yaratish uchun CREATE TABLE operatori ishlataladi, uning formati quyidagicha:

```
<jadval yaratish operatori> ::= CREATE TABLE <jadval nomi>  
({<ustun nomi> {<domen nomi> | <berilgan turi>}  
| NOT NULL | NULL } || UNIQUE || PRIMARY KEY )  
| <kelishuv bo'yicha qiymat> [<check chegarasi>] } , ...  
<chevara tavsif> ::= { [CONSTRAINT <chegaralanish nomi>]  
[<birlamchi kalit>] [<muqobil kalit> , ... ]  
[<tashqi kalit>] [CHECK (<jadval bo'yicha  
shart>)] } , ... }
```

Har bir ustun uchun uning turi yoki domeni ko'rsatiladi. Kalit so'zlar ustunning xususiyatini berish uchun quyidagicha ishlataladi:

- **NULL** – ustun qiymati bo'sh bo'lishi mumkin;
- **NOT NULL** – ustun qiymati majburiydir;
- **PRIMARY KEY** – ustun birlamchi kalit;
- **UNIQUE** – ustun qiymati noyob;
- **DEFAULT** – ustun uchun kelishuv bo'yicha qiymat berish;
- **CHECK** – ustun uchun chevara berish.

Qolgan parametrlar jadvalga tegishli, bo'lib ularga **CONSTRAINT** kalit so'zi bilan nom berish mumkin. Bu nom jadvalni o'zgartirishda ishlash mumkin.

Misol:

talaba (id\_talaba, zach\_nom,familiya, ismi, sharifi, id\_millati, jinsi, pas\_ser, pas\_nom, tug\_yil, manzil, ogirligi, tug\_joi, id\_tum, id\_ray, id\_fak) munosabatning jadvali yaratilsin:

Oldin domenlarni yaratamiz

```
CREATE DOMAIN Jjnsi CHAR(3)
```

```
CHECK VALUE IN ('erkak', 'ayol')
```

```
CREATE DOMAIN Dtug_yil INT
```

```
CHECK VALUE BETWEEN 1960 AND 1989
```

```
CREATE TABLE talaba (id_talaba INTEGER PRIMARY KEY,  
zach_nom CHAR(8) NOT NULL UNIQUE,  
familiya VARCHAR(40) NOT NULL,  
ismi VARCHAR(40) NOT NULL,  
sharifi VARCHAR(40),  
id_millati INTEGER,  
jinsi Jjnsi,  
pas_ser CHAR(2) NOT NULL,  
pas_nom CHAR(8) NOT NULL,  
tug_yil Tug_yil,  
manzil VARCHAR(255),  
ogirligi DECIMAL (6,2),  
tug_goy VARCHAR(50) DEFAULT 'Toshkent',  
id_tum INTEGER,  
id_ray INTEGER,  
id_fak INTEGER,  
stipendiya INTEGER,  
UNIQUE (pas_ser,pas_nom),  
CONSTRAINT Millat  
FOREIGN KEY (id_millati)  
REFERENCES millatlar  
ON DELETE SET NULL,  
CONSTRAINT Oblast  
FOREIGN KEY (id_obl)  
REFERENCES viloyat  
ON DELETE SET NULL,  
CONSTRAINT Rayon  
FOREIGN KEY (id_ray)  
REFERENCES tumanlar  
ON DELETE SET NULL,
```

```
CONSTRAINT Fakul
  FOREIGN KEY (id_fak)
  REFERENCES fakultet
    ON DELETE CASCADE
    ON UPDATE CASCADE)
```

### Jadval tavsifini o'zgartirish

Jadval yaratilgandan keyin uni tuzilishini o'zgartirish uchun **ALTER TABLE** operatori qo'llanadi, bu operator yordamida quyidagi ishlardan amalga oshirish mumkin:

- jadvalga yangi ustun kiritish;
- jadvaldan ustunni o'chirish;
- jadval uchun yangi chegaralanishlar kiritish;
- jadval chegaralanishlarini o'chirish;
- ustun uchun kelishuv qiymatlarini o'matish;
- kelishuv qiymatlarni o'chirish.

Bu operatorning formati quyidagicha:

```
<jadval o'zgartirish> ::= ALTER TABLE <jadval nomi>
  [ADD | COLUMN] {<ustun nomi>
    [<domen nomi> | <berilgan turi>]
    [NOT NULL | NULL | UNIQUE]
    [<kelishuv bo'yicha qiymat>] [<check
      chegarasi>] [, ...]
  } | (DROP COLUMN [<ustun nomi>
    [RESTRICT | CASCADE]] [, ...])
  | (ADD <chevara tavsifi>) [, ...]
  | DROP CONSTRAINT [<chevara nomi>]
    [RESTRICT | CASCADE]] [, ...]
  | ALTER [COLUMN] {<ustun nomi>
    [SET DEFAULT <kelishuv qiymati>
    | DROP DEFAULT]} [, ...]
```

Misol:

talaba jadvaldan zach\_nom ustunni o'chirish kerak, yangi ustun qo'shilsin kir\_yil nonili.

```
ALTER TABLE talaba
  ADD COLUMN kir_yil INT NOT NULL DEFAULT
  2011 CHECK kir_yil>2010;
ALTER TABLE talaba
  DROP COLUMN zach_nom RESTRICT;
```

## Jadvalni olib tashlash

Jadvalni olib tashlash uchun quyidagi operator ishlataladi:

<jadval olib tashlash operatori> ::= **DROP TABLE** <jadval nomi>  
[ **RESTRICT** | **CASCADE** ]

Jadvalni olib tashlashda, barcha satrlar ham yo'qotiladi. Agar **RESTRICT** kalit so'zi ko'rsatilgan bo'lsa, boshqa obyektlarda jadvalga murojaat bo'lsa (masalan tasavvurlarda), u holda jadval olib tashlanmaydi, **CASCADE** kalit so'zi berilganda esa, barcha havola qilgan obyektlar ham o'chiriladi va h.k. Shu sababli, **CASCADE** kalit so'zni ehtiyyotlik bilan ishlatmoq kerak.

Misol:

```
DROP TABLE talaba RESTRICT
```

## Indeks yaratish

Indekslar jadval satrlarini tezkor qidirish uchun ishlataladi. Agar jadvalda indekslar mavjud bo'lsa har yangi satr qo'shilganda jadval indeks bo'yicha qayta tartiblanadi shu sababli yangi satr kiritish vaqtida cho'ziladi, lekin jadvalda qiymat qidirish ancha tezlashadi. SQL tilini namunasida bunday operator yo'q, lekin barcha dialektlarda bunday operator bor. Indeks jadvalini yaratish uchun **CREATE INDEX** operatori qo'llash kerak:

<indeks yaratish operatori> ::= **CREATE [UNIQUE] INDEX**  
<indeks nomi> ON <jadval nomi>  
({<ustun nomi> [**ASC** | **DESC**] } [...])

Misol: Birlamchi kalit bo'yicha indeks yaratilsin:

```
CREATE UNIQUE INDEX indtal ON talaba  
(id_talaba)
```

Indeksni olib tashlash uchun **DROP INDEX** operatori ishlataladi:

<indeksni olib tashlash operatori> ::= **DROP INDEX** <indeks nomi>

Misol:

```
DROP INDEX indtal
```

## Tasavvurlar

*Tasavvur* – bu bitta yoki bir nechta fizik jadvallar ustidan bajarilgan amallar natijasida hosil bo'lgan natijaviy dinamik jadvaldir. Tasavvur bu so'rov natijasida yaratilgan virtual jadval hisoblanadi. Tasavvurni haqiqiy jadvaldan farqi shundaki, bunday jadval fizik mavjud emas. Uning ustunlari va satrlari haqiqiy jadvaldan olingen va tasavvur ustidan bajarilgan barcha ishlolar haqiqiy jadvallarni o'zgartiradi. BBBT tasavvurlarni tafsifini berilganlar bazasida saqlaydi. Bunday tasavvurga murojaat qilinganda BBBT ikki usul bilan ishlashi mumkin.

Birinchi usul bo'yicha tasavvurga bo'lgan har murojaatda ko'rsatilgan so'rov bajariladi va natijada vaqtincha jadval hosil bo'ladi, tasavvur ishlataligandan so'ng jadval tizim tomonidan olib tashlanadi.

Ikkinci usulda esa, tasavvur uchun alohida vaqtincha jadval tuziladi va tasavvurga kirgan jadvallarning o'zgarishining jarayonida birdaniga vaqtincha jadval ham o'zgaradi.

### Tasavvurni yaratish

Tasavvurni yaratish uchun **CREATE VIEW** operatori ishlatalidi uning sintaksisi quyidagicha:

```
<tasavvurni yaratish operatori> ::= CREATE VIEW <tasavvur nomi> [(<ustun yangi nomi> [, ...])] AS  
<ianlash operatori> [ WITH | CASCDED | LOCAL ]  
CHECK OPTION ]
```

Tasavvur **SELECT** operatori orqali tafsiflanadi. Natijaviy jadvalning ustunlariga yangi nomlar berish mumkin, bu holda ro'yxatdagi ustunlar nomining soni natijaviy jadvaldagi ustunlarning soniga teng bo'lishi kerak. Agarda natijaviy jadvalda hisoblanuvchi ustunlar yoki bir xil nomli ustunlar mavjud bo'lsa, ustunlarni yangi nomlari ko'rsatilishi shart. Tanlash operatori orqali berilgan so'rov qismi aniqlovchi so'rov, deb nomlanadi. **WITH CHECK OPTION** konstruksiyasi ko'rsatilsa, o'zgartirilgan satr aniqlovchi so'rovning **WHERE** bo'limidagi ko'rsatilgan shartni qanoatlantirmasa, manba jadvalga mos satr o'zgartirilmaydi.

#### Misol:

##### Gorizontal tasavvurni yaratish

Faqat 1 raqamli fakultet talabalarini ko'rish imkoniyatini beruvchi tasavvurni yaratish quyidagicha bo'ladi:

```
CREATE VIEW faktalaba
AS SELECT *
    FROM talaba
    WHERE kod_fak = 1;
```

Bu operatorni bajarish natijasida faktalaba nomli tasavvur yaratiladi. Bu tasavvur manba jadvalni barcha ustunlaridan iborat bo'lib, faqat 1 raqamli fakultetga tegishli satrlarni ko'rsatadi. Bunday so'rov *gorizontal tasavvur* deb nomlanadi, chunki gorizontal chiziq bilan cheklanadi. Bu tasavvurni ko'rish uchun quyidagi operatorni ishlatalish mumkin:

```
SELECT * FROM faktalaba;
```

Tasavvur orqali berilganlar bazasini himoya qilish mumkin, chunki tashkilotning har bir bo'limi faqat o'ziga tegishli ma'lumotlarni ko'rish imkoniyatiga ega bo'ladi.

#### Misol:

Vertikal tasavvurni yaratish.

Faqat, 1 raqamli fakultetning talabalarini ko'rish imkoniyati bo'lsin. Stipendiya to'g'risida ma'lumot, faqat hisob bo'limiga tegishli bo'lgani uchun tasavvurda ko'rsatilmas. Asosan talabani tartib raqami, zachenet raqami, familiyasi, ismi, otasining ismi, tug'ilgan yili va jinsi ko'rsatilsin. Bunday tasavvur quyidagicha yaratiladi:

```
CREATE VIEW faktalabal
AS SELECT id_talaba, zachenet_nom,
    familiya, ismi, sharifi, tug_yil,
    jinsi, id_fak
    FROM talaba
    WHERE id_fak = 1;
```

Bu tasavvur faydalanuvchiga faqat ma'lum ustunlardagi ma'lumotlarni ko'rishga imkoniyat beradi, boshqa ustunlar himoyalangan bo'ladi va imtiyozga ega bo'lgan foydalanuvchilarning ko'rish imkoniyati bo'ladi. Bu tasavvurni boshqacha ham berish mumkin:

```
CREATE VIEW faktalabal
AS SELECT id_talaba, zachenet_nom,
    familiya, ismi, sharifi, tug_yil, jinsi
    FROM faktalaba
```

Manba jadvallarning bir nechta ustunlari tushirilgan tasavvurlar vertikal tasavvurlar, deb nomlanadi. Agar jadvallar bilan har xil foydalanuvchilar

ishlasa, vertikal tasavvurlarni qo'llash ma'qil hisoblanadi, chunki bu holda faqat kerakli ustunlar chiqadi.

#### Misol:

- Bir nechta jadvallarni tutashtirish va guruhash tasavvuri.  
Talabani o'zlashtirish ko'rsatkichi to'g'risida ma'lumot beruvchi tasavvur yaratilsin. U fakultet raqamini, talaba raqami va o'rta balidan iborat bo'lsin.

```
CREATE VIEW talababal (fak_raqami, tal_rakami,  
orta_bal)  
AS SELECT t.id_fak, t.id_talaba, AVG(bal)  
FROM talaba t INNER JOIN uzlashtirish u  
ON t.id_talaba=u.id_talaba  
GROUP BY t.kod_fak, t.id_talaba
```

Bu tasavvurda agregat funksiyalar ishlatalganligi sababli, ustunlar ro'yxati keltirilgan. Bunday tasavvurlar *guruhashlangan yoki tutashtirilgan tasavvurlar* deb nomlanadi. Bunday tasavvurlarni qulayligi shundaki, ikki manba jadval o'rniiga bitta natijaviy jadval hosil bo'ladi va bu jadvalni boshqa so'rovlarda bitta jadval sifatida ishlash mumkin.

#### Tasavvurni olib tashlash

Tasavvurni olib tashlash uchun **DROP VIEW** operatori qo'llanadi, uning shakli quyidagicha:

```
<tasavvur olib tashlash operatori> ::= DROP VIEW <tasavvur  
nomi> [RESTRICT | CASCADE]
```

**DROP VIEW** operatorini bajarish natijasida, tasavvurning tavsifi berilganlar bazasidan olib tashlanadi. Masalan, faktalaba tasavvurni olib tashlash quyidagicha bo'ladi:

```
DROP VIEW faktalaba;
```

**RESTRICT** va **CASCADE** kalit so'zlarining ma'nosi, jadvalni olib tashlash operatorining ta'rifida keltirilgan.

#### Tasavvurni tanlash operatoriga almashtirish

Biror bir tasavvurga so'rovni bajarishini batafsil ko'rib chiqamiz, masalan quyidagi so'rov bajarilsin:

```
SELECT tal_raqami, ortabal
      FROM talababal
     WHERE fak_raqami=1
    ORDER BY tal_raqami
```

Bu so'rovni bajarish natijasida, manba tasavvuri **quyidagi ekvivalent** so'rovga almashtiriladi:

```
SELECT t.id_talaba AS tal_raqami, AVG(bal)
AS ortabal
      FROM talaba t INNER JOIN uzlashtirish u
      ON t.id_talaba=u.id_talaba
     WHERE id_fak=1
    GROUP BY t.id_fak, t.id_talaba
    ORDER BY t.id_talaba
```

### Tasavvurni qo'llashga qo'yilgan chegaralar

SQL tilining namunasida tasavvurlarni yaratishga va qo'llashga muhim chegaralar qo'yilgan. Lekin ayrim naqlarda bu chegaralar katta farq qilishi mumkin.

Tasavvurning ustuni agregat funksiyalar orqali tuzilganda, tasavvurni chaqirgan so'rovda ushbu ustun faqat SELECT va ORDER BY konstruksiya orqali ko'rsatilishi mumkin. Bu ustunni WHERE konstruksiyada yoki so'rovni agregat funksiyalarida argument sifatida ishlatish mumkin emas. Masalan, yuqorida ko'rsatilgan tasavvurni quyidagicha ishlatish mumkin emas:

```
SELECT COUNT(ortabal)
      FROM talababal;
```

Hatto, bitta ustunga agregat funksiya ikki marta qo'llanmoqda, bu mumkin emas.

Quyidagi so'rov hani xatodir, chunki agregat funksiya orqali hisoblangan ustun WHERE konstruksiyasida qo'llanmoqda:

```
SELECT *
      FROM talababal
     WHERE ortabal>71;
```

Guruhi langan tasavvurlar boshqa jadvallar yoki tasavvurlar bilan tutashtirilmasiligi kerak. Bu tutashtirish xatolikka olib keladi.

## Tasavvurdagi berilganlarni o'zgartirish

Berilganlar bazasidagi jadvallarga kiritilgan o'zgarishlar shu jadvallarga murojaat qilgan tasavvurlarda birdaniga aks etiladi. Xuddi shunday tasavvurda bajarilgan o'zgarishlar manba jadvallaridagi berilganlarni o'zgarishiga olib keladi. Lekin bu har doim ham bajarilmaydi. Masalan, talababal tasavvuriga 2 raqamli fakultet va 1000 raqamli talabanining 72 o'rtacha bali to'g'risida quyidagicha yangi satr qo'shamoqchi bo'lib,

```
INSERT INTO talababal
VALUES (2,1000,72);
```

operatorini yozsak, bunday so'rov xatolikka olib keladi, chunki ko'rsatilgan o'rta baflni hosil qilish uchun o'zlashtirish jadvaliga qo'shimcha satrlar kiritish kerak, lekin qaysi va nechta fanga ball qo'yilishi kerakligi noma'lum. Agar biz agregat funksiyanı ishlatsadan, talabani fan va balini ko'rsatsak, tasavvur quyidagicha bo'лади:

```
CREATE VIEW talabauzlash (fak_raqami,
tal_rakami,bal)
AS SELECT t.id_fak, t.id_talaba, u.bal
FROM talaba t INNER JOIN uzlashtirish u
ON t.id_talaba=u.id_talaba
```

Agar bu tasavvurga ham yangi satrni quyidagi operator bo'yicha qo'shamoqchi bo'lsak:

```
INSERT INTO talabauzlash
VALUES (2,1000,72);
```

bunday so'rov ham manba jadvalga o'zgartirish kiritish imkoniyatini bermaydi, chunki o'zlashtirish jadvalida fan nomi albatta ko'rsatilishi kerak, lekin so'rovda fan nomi yo'q.

SQL namunasidagi tasavvurda ishlov amallarini quyidagi hollarda bajarish mumkin:

- konstruksiya tavsifida **DISTINCT** kalit so'zi ishlatalmasa, ya'ni natijaviy jadvaldan qaytarilgan satrlar olib tashlanmagan bo'lsa;
- aniqlovchi so'rovdagи **SELECT** konstruksiyasidagi ustunlar ro'yxatida faqat ustun nomi (o'zgarmas, ifoda, agregat funksiyalar

ishlatilmagan holda) bo'lishi va bu nom ro'yxatda bir marta qatnashishi kerak;

- **FROM** konstruksiyasida faqat bitta jadval ko'rsatilgan bo'lsa, ya'ni tasavvur yagona jadval asosida tuzilgan bo'lsa, undan tashqari manba jadvalning o'zi ham tasavvur bo'lsa, u ham bitta jadval asosida tuzilgan bo'lishi kerak. Bu talab jadvallarni tutashtirish (**JOIN**), birlashtirish (**UNION**), kesishish (**INTERSECT**) yoki ayirish (**EXCEPT**) amallari asosida tuzilgan tasavvurlar ustidan ishlov man etilishiga olib keladi;
- **FROM** konstruksiyasida ko'rsatilgan jadvalga tegishli bo'lgan ustunlar **WHERE** konstruksiyani ichki so'rov qismining **SELECT** ro'yxatiga kirmagan bo'lsa;
- aniqlovchi so'rovda **GROUP BY** va **HAVING** konstruksiyalar bo'tmasa.

Bu holda bunday tasavvurlar ishlovli tasavvurlar deb nomlanadi.

Undan tashqari tasavvur orqali qo'shilayotgan har qanday satr manba jadvalga qo'yilgan, berilganlarni butunlik qoidasini buzmasligi kerak. Masalan, tasavvur orqali yangi satr qo'shilganda, barcha tasavvurga kirmagan ustunlar qiymati **NULL** yoki kelishuv qiymatga teng bo'ladi, shunda manba jadvalda qo'yilgan **NOT NULL** chegara buzilmasligi kerak.

Xullas, tasavvur ustidagi ishlovlarni bajarish imkoniyatiga ega bo'lish uchun, BBBT bu o'zgarishlarni manba jadvalga mos ravishda o'tkaza olishi kerak.

### **WITH CHECK OPTION** konstruksiyanı ishlatilishi

Tasavvurga aniqlovchi so'rovdagagi **WHERE** shartini qanoatlanuvchi satrlarga qo'shiladi. Agar tasavvurdagi satr biron bir ishlov natijasida o'zgartirilib, shartni qanoatlantirmasa, bunday satr tasavvurdan o'chiriladi. Xuddi shunday qo'shish yoki o'zgartirish natijasida, **WHERE** shartini qanoatlantiruvchi tasavvurga yangi satrlar qo'shilib qoladi. Qiymatlari asosida tasavvurga qo'shiluvchi yoki o'chiruvchi satrlar erkin (migratsiya qiluvchi) satrlar deb nomlanadi. Umuman olganda, satrlar migratsiyasini oldini olish uchun **WITH CHECK OPTION** konstruksiysi qo'llanadi. Bu konstruksiyada majburiy bo'lmagan holda ko'rsatilgan **LOCAL/CASCA-**

**DED** kalit so'zları tasavvurlarni kaskadi mavjud bo'lganda qo'llanadi, ya'ni biror bir tasavvur boshqa tasavvur asosida tuzilgan holda. Agar **WITH LOCAL CHECK OPTION** konstruksiya ko'rsatilsa va kaskad tasavvurda satrlar o'chirilib ketgan bo'lsa, tayanch tasavvurga qo'shilgan yoki o'zgartirilgan satrlar ham o'chiriladi. Aksincha, **WITH CASCADED CHECK OPTION** konstruksiysi ko'rsatilsa (kelishuv bo'yicha qiymat), tasavvurda yoki uning avlodida qo'shilgan yoki o'zgartirilgan satr tasavvurdan o'chirilmaydi. Shuni aytib o'tish kerakki, **WITH CHECK OPTION** konstruksiyanı faqat ishlövli tasavvurlarda qo'llash mumkin. Bizga quyidagi tasavvur berilsin:

```
CREATE VIEW faktalaba
AS SELECT *
    FROM talaba
    WHERE id_fak = 1
    WITH CHECK OPTION;
```

Endi shu tasavvurga kiruvchi 100- raqamli talabaning fakultet kodini o'zgartiramiz:

```
UPDATE faktalaba
SET id_fak=2
WHERE id_talaba=100;
```

Tasavvurga **WITH CHECK OPTION** konstruksiyası qo'shilgani sababli mos bo'lgan satr o'zgartirilmaydi, chunki satr migrantsion bo'lib qoladi. Endi shu tasavvurga yangi satr qo'shishga harakat qilib ko'ramiz: talaba(id\_talaba, zah\_nom, familiya, ismi, sharifi, id\_millati, pas\_set, pas\_nom, tug\_yil, manzil, ogirligi, id\_tum, id\_ray, id\_fak)

```
INSERT INTO faktalaba (id_talaba, zah_nom,
familiya, ismi, sharifi, tug_yil, jinsi,
id_fak
VALUES(10123,'K-01001','Dadjanov','Maxmud',
'Achilovich', 1990, 'erkak', 2);
```

Tasavvurda **WITH CHECK OPTION** konstruksiysi ko'rsatilgani uchun, bunday satr tasavvurga qo'shilmaydi, chunki bu satr jadvalga qo'shilgandan keyin, tasavvurdan shu zahoti o'chirilishi kerak (chunki id\_fak=2). Endi boshqa tasavvurga asoslangan tasavvurni ko'ramiz:

```
CREATE VIEW ostitalaba
AS SELECT *
```

```

        FROM talaba
        WHERE stipendiya>150000;
CREATE VIEW ustitalaba
AS SELECT *
        FROM ostitalaba
        WHERE stipendiya>200000;
CREATE VIEW faktalaba
AS SELECT *
        FROM ustitalaba
        WHERE id_fak = 1

```

Endi 100-raqanli talabani stipendiyasini 220000 so'mdan 180000 so'ninga o'zgartirishga harakat qilamiz:

```

UPDATE faktalaba
SET stipendiya=180000
WHERE id_talaba=100;

```

lekin bu o'zgartirish bajarilmaydi, chunki satr ustitalaba tasavvurdan yo'qoladi, lekin ostitalabadan yo'qolmaydi, agarda stipendiyani 140000 o'zgartirilsak, unda bu satr ostitalaba tasavvurdan ham yo'qoladi, natijada bu o'zgartirish bajariladi. Agar ustitalaba tasavvurga WITH CASCADED CHECK OPTION konstruksiya qo'shilsa, ko'rsatilgan stipendiya qaysi qiymat (180000 yoki 140000) bo'lishidan qat'i nazar, bu satr ustitalaba tasavvurdan olib tashlanishi kerak bo'ladi, shuning uchun bu o'zgarishlar bajarilmaydi.

### Xulosha

- SQL DDL tilining operatorlari berilganlar bazasida yangi obyektlarini yaratishga imkoniyat beradi.
- SQL tilining na'munasiga bazani yaratish operatori kiritilmagan, naqlarda buning uchun maxsus operatorlari bor.
- CREATE va DROP SCHEMA operatorlari chizmani yaratish va olib tashlashda ishlataladi.
- CREATE, ALTER va DROP TABLE operatorlari berilganlar bazasida jadvallar yaratish, shakiini o'zgartirish va olib tashlash imkoniyatini beradi.
- CREATE va DROP INDEX ko'rsatilgan jadval uchun indeks yaratish va olib tashlash uchun ishlataladi.

- Tasavvur deb, virtual jadvalga aytildi. Bu jadval tarkibida bir yoki bir nechta berilganlar bazasini jadvallari yoki boshqa tasavvurlar ustunlarining nomi, hisoblanuvchi ustunlar bo'lishi mumkin.
- Tasavvurlar CREATE VIEW operator yordamida tuziladi va DROP VIEW operatori bilan olib tashlanadi. Tasavvurga havola qilinganda vaqtincha jadval yaratiladi.
- SQL tilining namunasida tasavvurlarni yaratishga va qo'llashga chegaralar qo'yilgan.
- Tasavvurlar ustidan ishlovlar bajarish mumkin. Buning uchun ular ma'lum shartlarni qanoatlanishi kerak.
- Ishlovli tasavvurlarni o'zgartirishlarda ba'zi bir chegaralar qo'yiladi. Ularni **WITH CHECK OPTION** konstruksiyasi orqali bajarish mumkin.
- Berilganlar bazasini butunligini saqlash uchun iyerarxik holda, har bir tasavvurga **WITH CASCADED CHECK OPTION** konstruksiyasini qo'shish ma'qul bo'ladi.

### **Nazorat uchun savollar va amaliyot uchun topshiriqlar**

1. Jadvalni yaratish uchun qaysi operator ishlataladi va uni sintaksisi qanaqa?
2. Jadvalni o'zgartirish va o'chirish uchun qaysi operator ishlataladi?
3. Tasavvur bu nima? Tasavvur qaysi operator orqali yaratiladi?
4. Tasavvur ishlovli bo'lishi uchun nimalarni qanoatlantirishi lozim?

Quyidagi berilgan munosabatlar uchun butunligi hisobga olgan holda jadvallarni yaratish operatorlari keltirilsin:

1. talaba(id\_talaba, familiya, ismi, sharifi, id\_millati, jinsi, pas\_ser, pas\_nom, tug\_yil, manzil, ogirligi, tug\_joi, id\_tum, id\_ray, kod\_fak)
2. prikaz(id\_prikaz, id\_talaba, ykuv\_yil, id\_fak, id\_spec, tipob, tipfin, kurs , til,tip\_stip, tip\_prik)
3. fakultet(id\_fak, name\_fak, dekan, telefon)
4. specialnost(id\_spec, name\_spec, id\_fak, tipob, shifrspec)
5. millatlar(id\_millati, name\_millat)
6. tumanlar(id\_tum, name\_tum)
7. rayonlar(id\_ray, id\_tum, name\_ray)

Taqdimot yaratish

8. Talaba va prikaz jadvallarini tutashtirgan talaba tasavvuri yaratilsin **va bu tasavvurda familiya va ism-shariflari qo'shilgan ustun yaratilsin.**

## 30-bob. TRANZAKSIYANI QO'LLASH

**Tayanch iboralar:** *tranzaksiya, chegarani tekshirish, imtiyozlar, huquqlar, COMMIT, ROLLBACK, SET TRANSACTION, ISOLATION LEVEL, READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE, DEFERRABLE, AUTHORIZATION, REFERENCES, GRANT, REVOKE, soydalanuvchilar identifikatori.*

SQL namunasida tranzaksiya modelini tavsiflashda ikkita maxsus operatorlar qo'llanadi — **COMMIT** va **ROLLBACK**. Bir nechta ketma-ket SQL operatorlarni tiklash nuqtayi nazaridan butun bo'linmas amal tranzaksiya deb nomlanadi. Tranzaksiya berilgan bazasini butunligini saqlash uchun qo'llanadi. SQL tilining namunasida yozilishi bo'yicha, har bir operator bajarilishni boshlaganda, tranzaksiya avtomatik ravishda boshlanadi. Tranzaksiya tugamaguncha, ya'ni tranzaksiya oshkor ravishda to'xtatilmaguncha, boshqa parallel ishlagan tranzaksiyalar, kiritilgan o'zgarishlarni hisobga olmaydi. Tranzaksiyaning bajarilishi to'rt holatda to'xtatilishi mumkin:

- **COMMIT** operatorini kiritish orqali. Bu holda tranzaksiya muvaffaqiyatli tugaydi va bazaga kiritilgan o'zgarishlar doimiy saqlanib qoladi. Bundan keyin bajarilgan har qanday operator yangi tranzaksiyani boshlaydi.
- **ROLLBACK** operatorini kiritish orqali. Bu holda tranzaksiyani jarayonida qilingan barcha o'zgarishlar, qayta joyiga olib kelinadi (man etiladi) va baza tranzaksiya boshlanishdan oldingi holatga qaytadi. Tranzaksiya muvaffaqiyatsiz bajarilgan deb hisoblanadi. Bundan keyin bajarilgan har qanday operator yangi tranzaksiyani boshlaydi.
- Agarda **COMMIT** operatori oshkor ravishda kiritilmasa ham, SQL operatori programma orqali bajarilsa va programma bajarishni muvaffaqiyatlidir. Tugatsa, tranzaksiya muvaffaqiyatli bajarildi deb hisoblanadi.
- Agar SQL operatori programma orqali bajarilsa, programma bajarilishi avariyaligi tugatilsa, oxirgi tranzaksiyada qilingan o'zgarishlar qayta joyiga olib kelinadi va baza tranzaksiya boshlanishdan oldingi holatga qaytadi.

SQL tilida bir-biriga kirgan tranzaksiyalarni qo'llash man etilgan. SET TRANSACTION operatorlar orqali foydalanuvchilar tranzaksiya jarayonini boshqarishi mumkin.

Bu operatorning formati quyidagicha bo'ladi:

```
<tranzaksiya o'rnatish> ::= SET TRANSACTION
| READ ONLY | READ WRITE | [ ISOLATION LEVEL { READ
UNCOMMITTED | READ COMMITTED | REPEATABLE READ |
SERIALIZABLE } ]
```

READ ONLY kvalifikatori, tranzaksiyada berilganlarni faqat o'qish mumkinligini bildiradi. READ WRITE – tranzaksiyada ham o'qish, ham yozish mumkinligini ko'rsatadi. Kelishuv bo'yicha READ WRITE hisoblanadi (bu xolda READ UNCOMMITTED ko'rsatilmasligi kerak).

ISOLATION LEVEL kvalifikatori izolyatsiya darajasini ko'rsatadi. Bu kvalifikator tranzaksiyalar parallel bajarish jarayonida tartibni ko'rsatadi. Bajarish tartibi natija (seriyasida) quyidagi izolyatsiya darajalari ishlatalishi mumkin:

- **READ UNCOMMITTED** – tugatilmagan o'qish darjasи;
- **READ COMMITTED** – tugatilgan o'qish darjasи;
- **REPEATABLE READ** – qayta o'qish darjasи;
- **SERIALIZABLE** – tartib bo'yicha o'qish darjasи.

Agar ba'zi bir tranzaksiyalar quiyi darajaga ega bo'lib tartibsiz ishlasa, quyidagi xato holatlar bo'lishi muumkin:

- yangi fiksirlangan berilganni o'qish («fflos» o'qish);
- bir maydonдан ikki marta har xil qiymat o'qish (qaytarilmay o'qish);
- bitta shart bo'yicha ikki marta har hil saurlar sonini olish (phantom).

Bu holatlar uchun har bir izolyatsiya darjasи o'ziga mos ruxsatlar beradi:

Izolyatsiya darjasи	«fflos» o'qish	Qaytarilmay o'qish	Fantomlar
<b>READ UNCOMMITTED</b>	Ha	Ha	Ha
<b>READ COMMITTED</b>	Yo'q	Ha	Ha
<b>REPEATABLE READ</b>	Yo'q	Yo'q	Ha
<b>SERIALIZABLE</b>	Yo'q	Yo'q	Yo'q

Yangilash natijalari yo'qolmasligi kerak, shu sababli eng quyi darajali izolyatsiyadagi tranzaksiyalar bilan ehtiyoj bo'lib ishlash kerak. Agarda quyi darajali izolyatsiyada **READ UNCOMMITTED** ko'rsatilsa, kelishuv bo'yicha **READ ONLY** qabul qilinadi. To'liq havfsizlikni faqat **SERIALIZABLE** yuqori darajali izolyatsiya beradi (oshkormas qiymat).

### Berilganlarni butunligini saqlashdagi chegaralarni tekshirish

Ba'zi bir holatlarda, berilganlarni butunligini tekshirish vazifasi har bir operatordan keyin darhol bajarilmasdan, aksincha har bir tranzaksiya bajarilgandan keyin tekshiriladi. Bunday imkoniyatlar jadvalni tafsiflashda **NOT DEFERRABLE** (darhol) yoki **DEFERRABLE** (kechiktirilgan) chegaralarni tekshirish rejimlarini berish orqali amalga oshiriladi, uning sintaksisi quyidagicha:

**<tekshirish rejimi> ::= {DEFERRABLE [ INITIALLY DEFERRED | INITIALLY IMMEDIATE } | {NOT DEFERRABLE}**

Kelishuv bo'yicha **NOT DEFERRABLE** rejimi o'rnatiladi. Agar **DEFERRABLE** ko'rsatilsa, ikkita rejimni ishlatalish mumkin, darhol tekshirish yoki kechiktirilgan tekshirish, bu rejimlarni tranzaksiya bajarish jarayonida o'zgartirish mumkin. Buni **SET CONSTRAINTS** operatori orqali bajarish mumkin. Operatorning sintaksisi quyidagicha:

**<cheagaralash rejimi> ::= SET CONSTRAINTS {ALL | <constraint nom> [ ... ]} {DEFERRED | IMMEDIATE}**

Bunda **DEFERRED** har bir chegaraga yoki ko'rsatilgan chegaraga kechiktirilgan tekshirish rejimini o'rnatadi, ya'ni tranzaksiya bajarishni tugatish oldidan chegaralar tekshiriladi va shart qanoatlantirilmasa, tranzaksiya qaytariladi. **IMMEDIATE** darhol tekshirish rejimini o'rnatadi, ya'ni har bir amaldan so'ng chegaralar tekshiriladi va shart bajarilmasa, tranzaksiya qaytariladi.

Boshlang'ich holatini **INITIALLY DEFERRED** yoki **INITIALLY IMMEDIATE** bilan beriladi. Kelishuv bo'yicha **INITIALLY IMMEDIATE** rejimi o'rnatiladi.

### Berilganlar bazasiga kirish uchun ruxsat berishni boshqarish

Har bir BBB tizimida foydalanuvchilarga berilganlarga kirish ruxsatini beradigan imkoniyatlar bo'lishi kerak. SQL tilida bunday imkoniyatlarni **GRANT** va **REVOKE** operatorlari beradi. Bu operatorlar orqali berilganlar bazasidagi jadvallar himoyalanadi. Himoyalash usuli foydalanuvchilarning identifikatori va ularga berilgan egalik huquqi hamda imtiyozlar orqali bajariladi.

## Foydalanuvchilarning identifikatori va egalik huquqi

Berilgan bazasidan foydalanuvchini belgilaydigan SQL tilining oddiy identifikatori, *foydalanuvchilar identifikatori* deb nomlanadi. Har bir foydalanuvchiga berilganlar bazasini administratori tomonidan noyob identifikator beriladi va bu identifikatorni himoyalash uchun parol ham beriladi. BBT tomonidan bajarilgan har bir SQL operatori biror bir foydalanuvchi tomonidan bajariladi. Foydalanuvchining identifikatori orqali foydalanuvchiga qaysi obyektlarga havola qilish va ular ustidan qanday vazifalarni bajarish huquqi beriladi.

Har bir obyektni o'z egasi bor. Chizmani berish konstruksiyani **AUTHORIZATION** kalit so'zida shu chizmaga kirgan barcha obyektlarni egasi foydalanuvchining identifikatori orqali beriladi.

Boshida faqat obyekt egasi obyekt ustidan har qanday amallar bajarish huquqiga ega.

### Imtiyozlar

Jadvalga yoki tasavvurga nisbatan foydalanuvchi tomonidan bajarish huquqiga ega bo'lgan amallar *imtiyozlar* deb nomlanadi.

SQL namunasida quyidagi imtiyozlar aniqlangan:

- **SELECT** — jadvaldan berilganlarni tanlash huquqi;
- **INSERT** — jadvalga yangi satrlarni qo'shish huquqi;
- **UPDATE** — jadvaldagagi berilganlarni o'zgartirish huquqi;
- **DELETE** — satrlarni olib tashlash huquqi;
- **REFERENCES** — berilganlar butunligini tavsiflashda ko'rsatilgan jadval ustuniga havola qilish huquqi;
- **USAGE** — domenlarni qo'llash huquqi.

**INSERT** va **UPDATE** imtiyozlari jadvalni alohida ustunlar bilan chegaralanishi mumkin, bu holda foydalanuvchi faqat ko'rsatilgan ustunlarni o'zgartirishi mumkin. Xuddi shunday **REFERENCES** imtiyozi ham alohida ustunlarga qo'llanishi mumkin va ushbu ustunlar boshqa jadvallarning butunligini himoya qilishga ishlatalishi mumkin.

Foydalanuvchi **CREATE TABLE** operatori yordamida yangi jadval tuzganida, u avtomatik ravishda uning egasi bo'lib qoladi va barcha imtiyozlarga ega bo'ladi. Boshqa foydalanuvchilar shu yaratilgan jadvalga nisbatan dastavval hech qanday imtiyozlarga ega bo'lmaydi. Ularga jadvalga kirish ruxsatini berish uchun, jadval egasi oshkor ravishda foydalanuvchilarga kerakli huquqlarni berishi kerak. Buning uchun **GRANT** operatori qo'llanadi. **CREATE VIEW** operatori yordamida foydalanuvchi tasavvur yaratganda, u avtomatik ravishda uni egasi bo'ladi, lekin barcha huquqqa

ega bo'lmastigi mumkin. Tasavvur yaratish uchun foydalanuvchi **SELECT** va **REFERENCES** imtiyozlarga ega bo'lishi yetarli. Lekin tasavvurga nisbatan **INSERT**, **UPDATE** va **DELETE** imtiyozlarga ega bo'lishi uchun u barcha manba jadvallariga mos imtiyozlarga ega bo'lishi kerak.

### Boshqa foydalanuvchilarga imtiyozlar berish

Berilgan bazasidagi obyektlarga nisbatan ko'rsatilgan foydalanuvchi-larga imtiyozlar berish uchun GRANT operatori qo'llanadi. Bu operator jadval egasi tomonidan boshqa foydalanuvchilarga jadvalga kirish ruxsatini berish uchun ishlatalidi.

GRANT operatori quyidagi formatga ega:

```
<imtiyoz berish operatori> ::= GRANT{<imtiyozlar ro'yxati> | ALL  
PRIVILEGES} ON <ob'yekt nomi> TO {<foydalanuvchi ro'yxati> |  
PUBLIC} [WITH GRANT OPTION]  
<imtiyozlar ro'yxati> ::= <imtiyoz>[,...]  
<imtiyoz> ::= SELECT | DELETE | INSERT [<ustun nomi> [, ... ]]|  
UPDATE [<ustun nomi> [, ... ]]| REFERENCES | (<ustun nomi>  
[, ... ]) | USAGE  
<foydalanuvchi ro'yxati> ::= <foydalanuvchining identifikatori> [...]
```

Agar **ALL PRIVILEGES** ko'rsatilsa, foydalanuvchiga barcha imtiyozlar beriladi.

Bu operatordagi **PUBLIC** kalit so'zi ko'rsatilgan imtiyozlar faqatgina mavjud bo'lgan foydalanuvchilarga emas, balki barcha keyin aniqlanadigan foydalanuvchilarga ham beriladi.

Obyekt nomi bu – jadval, tasavvur, domen nomi bo'lishi mumkin.

**WITH GRANT OPTION** konstruksiyasi foydalanuvchilar ro'yxatida ko'rsatilgan barcha foydalanuvchilar ko'rsatilgan ob'ktiga nisbatan berilgan imtiyozlarini boshqa foydalanuvchilarga uzatishga imkoniyatlar beradi. Bu boshqa foydalanuvchilar o'z navbatida **WITH GRANT OPTION** konstruksiyasidan foydalaniib, boshqa foydalanuvchilarga imtiyozlar beradi va h.k. Agar bu konstruksiya ko'rsatilmasa, imtiyoz oluvchi boshqa foydalanuvchilarga huquqlarini uzata olmaydi.

**Misol:** Barcha imtiyozlarni berish.

Identifikatori Manager bo'lgan foydalanuvchiga talaba jadvali uchun barcha imtiyozlar berilsin.

```
GRANT ALL PRIVILEGES
```

```
    ON talaba  
    TO Manager WITH GRANT OPTION;
```

Natijada identifikatori Manager bo'lgan foydalanuvchi barcha imtiozlarga ega bo'ladi va u boshqa foydalanuvchilarga imtiozlar berish huquqiga ham ega bo'ladi.

Misol: Foydalanuvchilarga faqat ma'lum imtiozlar berish Personnel va Director nomli foydalanuvchilarga **SELECT** va **UPDATE** imtiozları talaba jadvalining faqat stipend ustuniga berilsin.

```
GRANT SELECT, UPDATE (stipend)  
ON talaba  
TO Personnel, Director;
```

Bu misolda, **WITH GRANT OPTION** konstruksiyasi berilmaganı uchun, Personnel va Director foydalanuvchilar o'z imtiozlarini boshqa foydalanuvchilarga uzata olmaydi.

Misol: PUBLIC turidagi barcha foydalanuvchilarga ma'lum imtiozlar berish.

Barcha foydalanuvchilarga fakultet jadval uchun **SELECT** imtiozları berilsin.

```
GRANT SELECT  
ON fakultet  
TO PUBLIC;
```

Keltirilgan misolda, PUBLIC katit so'zini ishlatalishi, barcha bor va yana keyinchalik yaratiladigan foydalanuvchilarga fakultet jadvalidan ma'lumot olish huquqiga egaligini bildiradi. **WITH GRANT OPTION** konstruksiyani ishlatish hojati yo'q, chunki barcha foydalanuvchilar ko'rsatilgan huquqlarga ega.

### **Berilgan imtiozlarini bekor qilish**

SQL tilida berilgan imtiozlarini bekor qilish uchun **REVOKE** operatori ishlataladi. Bu operator orqali ko'rsatilgan foydalanuvchiga berilgan barcha yoki ba'zi bir imtiozlar bekor qilinadi. Bu operatorni sintaksisi quyidagicha:

```
<imtiozni bekor qilish operatori>:= REVOKE | GRANT OPTION
```

```
FOR | {<imtiyozlar ro'yxati> | ALL PRIVILEGES}
ON <obyekti nomi>
FROM {<foydalanuvchilar ro'yxati> | PUBLIC}
[ RESTRICT | CASCADE ]
```

**ALL PRIVILEGES** kalit so'zi, ko'rsatilgan foydalanuvchini barcha imtiyozlarini bekor qiladi. Majburiy bo'limgagan **GRANT OPTION FOR** konstruksiyasi **GRANT** operatorida berilgan **WITH GRANT OPTION** konstruksiyasi orqali berilgan vakolatni bekor qiladi.

**RESTRICT** va **CASCADE** kalit so'zlarini ma'nosi jadvalni olib tashlash operatorining ta'sifida keltirilgan.

**Misol:** Barcha foydalanuvchilarga berilgan ma'lum imtiyozlarni bekor qilish. Barcha foydalanuvchilarni fakuktet jadvaliga berilgan **SELECT** imtiyozi bekor qilinsin:

```
REVOKE SELECT
ON fakultet
FROM PUBLIC;
```

**Misol:** Ko'rsatilgan foydalanuvchini talaba jadvaliga berilgan ma'lum bir imtiyozlari bekor qilinsin. Identifikatori Direktor bo'lgan foydalanuvchini barcha initiyozlari bekor qilinsin.

```
REVOKE ALL PRIVILEGES
ON talaba
FROM Director;
```

## Xulosa

- Tranzaksiyalarni boshqarish uchun COMMIT va ROLLBACK operatorlari ishlataladi.
- Tranzaksiya bajarish jarayonida chegaralarni darhol yoki kechiktirilgan tekshirish mumkin.
- Berilganlarga kirish uchun ruxsat berishni boshqarishi foydalanuvchilar identifikatori, ega bo'lishi huquqi va imtiyoz berish konsepsiylariga asoslangan.
- Foydalanuvchining identifikatori berilganlar bazasini administratori tomonidan beriladi. Bu identifikator har bir foydalanuvchida bo'lishi shart.

- Har bir yaratilgan obyektni egasi bo'ldi. Obyekt egasi boshqa foydalanuvchilarga shu obyektga kirish uchun vakolatlar va imtiyozlar berishi mumkin. Buning uchun **GRANT** operatori ishlataladi, keyinchalik bu imtiyozlar **REVOKE** operatori bilan olib tashlanishi mumkin.
- Imtiyozlarga quyidagi huquqlar kiradi: **USAGE**, **SELECT**, **DELETE**, **INSERT**, **UPDATE** va **REFERENCES**, shunda oxirgi uchta huquq alohida ustunlarga ham quyilishi mumkin.
- Foydalanuvchiga berilgan imtiyozlarni targatish huquqi **WITH GRANT OPTION** konstruksiya orqali berilishi mumkin. Bu imtiyoz **GRANT OPTION FOR** konstruksiyasi bilan bekor qilinishi mumkin.

### **Nazorat uchun savollar va amaliyot uchun topshiriqlar**

Izolyatsiya darajalarini keltiring.

Tranzaksiya natijasidagi xato holatlar nima dan iborat?

---

## 31-bob. TAQSIMLANGAN BERILGANLAR BAZASI

**Tayanch iboralar:** *taqsimlangan berilganlar bazasi, bir jinsli taqsimlangan berilganlar bazalari, bir jinsli bo'lmagan berilganlar bazalari, ilovalar serveri, BB serverlari, hisoblash tizimi, berilganlarning tugunlar, so'rovlarini qabul qiluvchi tugunlar, berilganlar bo'laklari va replikatsiyalar (o'ziga o'xshash tuzilishlar), gorizontal ajratish, vertikal ajratish, operatsion tizim, parallel BBB tizimlari, so'rovlararo parallellash, ichki so'rovlardagi parallellash, ichki operatsion parallellash, replikatsiya shaffofligi, bo'laklash shaffofligi, kirish shaffofligi, mukammal holdagi to'liq shaffoflik, chiziqli mijoyoslash, chiziqli tezlanish, markazlashgan BB, taqsimlangan BB, taqsimlangan berilganlar bazasini boshqarish tizimi, lokal kirish, tarmoq orgali kirish, «sayl-server» arxitekturasi, «mijoz-server» BB arxitekturasi, ajratilgan serverti arxitektura, SQL-server, «nozik» mijoz - ilovalar serveri - BB serveri arxitekturasi, berilganlar bazasining faol server arxitekturasi, «nozik» mijozli model, «semiz» mijozli model, trigger, so'rovlar ishlovi, so'rovlarini optimallashtirish, so'rov dekompozitsiyasi, lokallashtirish programmasi, relatsion algebra jarayonlari, kommunikatsion oddiylik, qidiruv fazosi, narx modeli, qidiruv strategiyasi, yarim birlaskatirish, ay so'plami, amallar tajuddalarining ay so'plami.*

*Taqsimlangan berilganlar bazasini boshqarish tizimining asosiy vazifasi hisoblash tizimining qaysidir tugunlarida joylashgan lokal berilganlar bazalarini integratsiya vositalarini ta'minlashdan iborat bo'lib, tarmoqning shu tugun nuqtasida ishlayotgan foydalanuvchi hamma berilganlar bazalariga xuddi yagona baza bilan ishlagandek kira olishi kerak.*

*Bir jinsli va bir jinsli bo'lmagan taqsimlangan berilganlar bazalari bo'lishi mumkin. Bir jinsli taqsimlangan berilganlar bazalari deganda, har bir lokal bazaning yagona BBB tizimiga ega bo'lgan holat tushuniladi.*

*Bir jinsli bo'lmagan tizimda lokal berilganlar bazalari turli berilganlar modellariiga tegishli bo'lishi mumkin. Bir jinsli bo'lmagan berilganlar bazasining tarmoq integratsiyasi juda murakkab muammodir. Ko'pgina yechimlar saqat nazariy jihatdan hal qilingan bo'lib, eng asosiy muammo, ya'ni integrallashgan tizimlar unumidorligi hali yetarli darajada emas. Bu sohadagi oraliq masala – bir jinsli bo'lmagan SQL ga yo'naltirilgan*

tizimlarning integratsiyasi, yaxshi hal qilinmoqda. Bunga SQL tilining standartlashgani ancha yordam bermoqda.

Taqsimlangan BBB tizimiغا misol tariqasida System R\* tizimini keltirish mumkin. Bu tizimda amaliy programmalarни yaratuvchilar va oxirgi foydalanuvchilar SQL tilining multipla ishlaydi. SQL tilidan foydalanish imkoniyatini System R\* tizimining berilganlarni joylashgan tugunlarining shaffofligini ta'mintashiga asoslangan. Tizim avtomatik ravishda, foydalanuvchining so'roviga ko'rsatilgan berilganlar obyektining joylashgan joriy joylarini aniqlaydi. SQL tilidagi ifodalardan tuzilgan bitta amaliy programma, tarmoqning turli tugunlarida bajarilishi mumkin. Bunda tarmoqning har bir tugunida so'rovni kompilatsiya bosqichida, so'rovni bajarish rejasining eng optimal naqli, tarmoqlangan tizimdagi berilganlarning joylashishiga nisbatan tanlanadi.

### Asosiy tushunchalar

BBB tizimlarining tiklanish davri taqsimlangan hisoblash va parallel ishlov texnologiyalarining rivojanishi davri bilan bir paytga to'g'ri kelgan. Natijada taqsimlangan BBB tizimlari va parallel BBT vujudga keldi. Aynan shu tizimlar berilganlarni qayta ishlovchi kuchli ilova yaratish uchun ustun vositalarga aylanmoqda. Taqsimlangan muhitda ishechi stansiyalarning integratsiyalashuvi sababli, undagi funksiyalarni taqsimlash samarali amalga oshirila boshlandi. *Ilovalar serveri*, deb nomlanuvchi ishechi stansiyada amaliy programmalarning bajarilishi, berilganlar bazalariga esa *BB serverlari* deb nomlanuvchi ajratilgan kompyuterlar orqali xizmat ko'rsatish shular jumlasiga kiradi. Bu esa shunday taqsimlangan arxitekturalar manbasining rivojanishiga olib keldiki, bunda tugunlar sifatida faqatgina umumiyl maqsadga xizmat qiluvchi oddiy kompyuterlar emas, balki ixtisoslashtirilgan serverlar muhim ahamiyatga ega bo'ldi.

Parallel kompyuter yoki multiprotsessorni (protsessorlar, xotira tarkiblari) o'z-o'zidan umumiyl korpus ichida tezkor tarmoq orqali bog'langan tugunlardan tashkil topgan taqsimlangan hisoblash tizimi deb, aytish mumkin. Taqsimlangan BB texnologiyasi tabiiy holda qayta qurilishi parallel BB tizimlari orqali tarqalishi mumkin.

**Taqsimlangan BB** (DDB – Distributed DataBase) – bu taqsimlangan kompyuter tarmog'ida mantiqan o'zaro bog'langan berilganlar bazalari majmuasidir. Taqsimlangan BBT, taqsimlangan berilganlar bazalarini shunday boshqarish imkonini beradigan programma tizimi kabi aniqlanadi, undagi berilganlar bazalarining taqsimlanganligi foydalanuvchilar uchun shaffof hisoblanadi. Bu arxitekturada ikkita farqlanuvchi xususiyatlarni ko'rish mumkin. Ulardan biri shundan iboratki, tizim so'rovlarini qabul

*qiluvchi tugunlar* (query site) to‘plamidan (baiki bo‘sh bo‘lgan) va bo‘sh bo‘lman berilganlarning tugunlar (data site) to‘plamidan iborat. Berilganlarning tugunları berilganlarni saqlash uchun vositalarga ega, so‘rovlarini qabul qiluvchi tugunlarda esa bunday vositalar yo‘q. So‘rovlarini qabul qilish tugunlarida faqatgina berilganlar tugunlarida saqlanuvechi berilganlarga kirish uchun foydalanuvchi aloqasini bajarilishini ta‘minlovchi programma bajariladi.

*Ikkinci xususiyat* esa shundan iboratki, bunda tugunlar mantiqan mustaqil kompyuterlar deb tasavvur qilinadi. Demak, bunday tugunlarda shaxsiy asosiy va tashqi xotira mavjud bo‘lib, shaxsiy operatsion tizim (OT) (hamma tugunlarda bo‘lishi yoki umuman bo‘lmashligi mumkin) o‘matilgan va ilovalarni yaratish imkonini yaratilgan bo‘ladi. Tugunlar multiprotsessor konfiguratsiyasiga emas, baiki kompyuter tarmog‘i orqali bog‘langan bo‘ladi.

BB berilganlar bo‘laklari va replikatsiyalar (*o‘ziga o‘xshash tuzilishlar*) asosida fizik ravishda berilganlarning tugunlariga taqsimlanadi. Relation berilganlarning chizmalarini mavjud bo‘lganda, har bir munosabat *gorizontal* va *vertical qismi* bo‘linadi.

*Gorizontal ajratish*, tanlash amali yordamida bo‘laklarga ajratish predikati qoidasiga amal qilgan holda, munosabatning har bir kortejini birorta tugunga jo‘natish orqali amalga oshiriladi. Masalan, Talabalar o‘quv loyihasidagi RO‘YHAT munosabatini har bir fakultet uchun alohida gorizontal ajratish mungkin.

*Vertical ajratishda* munosabatni nusxaflash amali yordamida bo‘laklarga ajratiladi. Masalan, RO‘YHAT munosabatini bir qismi Zn, Familiya, Ism, O\_Ism, Pasp\_S, Pasp\_N ikkinchi qismi esa K\_Sana, Fak\_N, Yon\_N, Kurs, Guruh maydanlariga ega bo‘lishi mumkin.

Berilganlarni bo‘laklash natijasida berilganlar eng ko‘p ishlataladigan joyda guruhanadi va bu qayta yo‘naltilishdagi xarajatlarni kamaytiradi va foydalanuvchi so‘rovlarida qatnashuvchi munosabatlarning o‘lehamini ham kichraytiradi. Berilganlarni bo‘laklarga ajratish, ularga kirish taysisi asosida o‘ziga o‘xshash tuzilishlarga ajratilishi mumkin. Agarda bir xil berilganlarga turli tugunlarda bajarilayotgan ilovalar orqali kirilsa, bu juda foydalib bo‘ladi, chunki berilganlarni uzlusiz ravishda tugunlararo uzatgandan ko‘ra berilganlardan tugunlarning qatorida foydalanmoq samarali hisoblanadi.

### Taqsimlangan va parallel BBBT

Taqsimlangan va parallel BBB tizimlarida aniq chegara mavjud emas. Xususiy holda, parallel BBB tizimlarining arxitekturasi quyida muhokama

qilinuvchi bir-biriga bog'liq bo'limgan, kuchsiz bog'langan taqsimlangan tizimlarda foydalanilgan zaxiralardan iborat. Parallel BBB tizimlarida yangi ko'p protsessorli arxitekturalardan foydalaniadi va shu asosda yuqori ishlab chiqarishga ega bo'lgan BB serverlari yaratiladi va uning narxi meynf-reymning ekvivalent tizimiga qaraganda ancha arzon bo'ladi.

BB programma ta'minotida ilovalar orqali berilganlar bilan shiddatli ishlovda parallelashning *uchta turi* qaraladi:

- *so'rovlararo parallelash* – turli xil tranzaksiyalarga mansub bo'lgan so'rovlar to'plamini bir vaqtning o'zida bajarilishiga asoslangan;
- *ichki so'rovlardagi parallelash* – bitta so'rovga taalluqli bo'lgan bir vaqtning o'zida bir nechta amallarning bajarilishidan iborat. Ichki so'rovlardagi parallelash ham, so'rovlararo parallelash ham o'xshash ravishda gorizontai bo'laklash orqali berilganlarni bo'lish asosida amalga oshiriladi;
- *ichki operatsion parallelash* – amallarning majmuasini berilganlar bo'laklariga qo'shimcha ravishda funksiyalarni bo'laklash usulini qo'llab, bitta amalni parallel bajarishdan iborat. To'plamlar ustidagi amallarga yo'naltirilgan BB tillari ichki operatsion parallelashni qo'llash uchun ko'pgina imkoniyatlarni ta'minlaydi.

Quyida taqsimlangan va parallel BBBT asosiy xususiyatlari sanab gan:

- *taqsimlangan/parallel BB* – bu tarmoqning turli tugunlarida saqlanuvchi fayllar «majmuasin» emas, balki aynan berilganlar jarasining o'zidir. Taqsimlangan BB va taqsimlangan fayllar tizimi orasidagi farq ana shundan iborat. *Taqsimlangan BB relation model kabi rasmiy tuzilish bilan bog'langan va ularga kirish uchun esa yagona yuqori bosqichli aloqa ishlataladi*;
- *taqsimlangan tizim BBBTning to'liq funksionalligiga egadir*. U o'z imkoniyatlari bo'yicha na taqsimlangan fayllar tizimiga, na tranzaksiyalarga o'xshamaydi. *Tranzaksiya* – bu tizimlarning faqatgina bir funksiyasıdir. Shu bilan bir qatorda, ular tranzaksiyani qayta ishlovchi tizimlarning qo'llashi shart bo'limgan so'rovlar funksiyalari va berilganlarning tarkibiy tuzilishlarini ta'minlashlari lozim;
- *taqsimlangan berilganlar* (bo'laklash va replikatsiyani hisobga olgan holda) tugunlar to'plamida foydalanuvchilar uchun ko'rinishmaydi. Bu xususiyat *shaffoflik*, deb ataladi. *Taqsimlangan/parallel BB texnologiyasi, BBni boshqarishdagi berilganlar erkintligi konsepsiyasini tarmoq orqali bog'langan kompyuterlar to'plamiga tarqatadi*. Bu shaffoflikning bir nechta turlari orqali ta'minlanadi –

- tarmoq shaffofligi* (demak, taqsimot shaffofligi), *replikatsiya shaffofligi* va *bo'laklash shaffofligi*. *Kirish shaffofligida*, foydalanuvchilar BBBning yagona mantiqiy qiyofasi bilan ish ko'radilar va taqsimlangan berilganlarga kirish xuddi berilganlarning markazlashgan holda saqlangani kabi amalga oshiriladi. *Mukammal holdagi to'liq shaffoflikda*, taqsimlangan BBB tizimlaridagi so'rovlar tili mavjud hamda uning markazlashgan BBB tizimlari uchun mavjud tildan farq qilmasligi talab qilinadi;
- shaffoflik parallel BBB tizimidan ko'ra taqsimlangan BBB tizimlari uchun tanqidiroqdir. Buning ikkita sababi bor. *Birinchidan*, parallel BBBT amalga oshiruvchi mikroprotsessorli tizimlar yagona OT boshqaruvi ostida ishlaydi. OT shunday tashkillash tirilgan bo'lishi mumkinki, BBBT ba'zi funksional tamoyillarni olishi hisobiga, shaffoflikning aniq darajasini ta'minlaydi. *Ikkinchidan*, parallel tizimlardagi programma ta'minotini yaratish parallel programmalash tillari orqali bajariladi va ular ham ba'zi shaffoflikning qaysidir darajalarini ta'minlaydi;
  - taqsimlangan BBB tizimlaridagi berilganlar va ulardagi ilovalar bitta tugun orqali amalga oshirilishi mumkin va buning natijasida berilganlarga masofadan kirish ehtiyoji yo'qoladi. Har bir tugunda kamroq ilovalar bajarilishi va berilganlar bazasining kamroq miqdori saqlanishi evaziga, berilganlarga va zaxiralarga kirish raqobatini ham qisqartirish mumkin. Va niyoyat, taqsimlangan tizimlarga xos bo'lgan parallelash, so'rovlararo va ichki so'rovlar parallelelligini amalga oshirish uchun imkoniyat yaratadi;
  - agarda foydalanuvchilarni berilganlar bazasiga kirishi, faqatgina so'rovlarni bajarilishidan iborat bo'lsa, unda so'rovlararo va ichki so'rovlar parallelelligini amalga oshirish, berilganlar bazasining maksimal qismini replikatsiyalashni talab qiladi. Lekin amaliyotda berilganlar bazasiga kirish faqatgina o'qish orqali amalga oshirilmagani uchun, berilganlarni takomillashtirish va navbatlab o'qishlar jarayonini amalga oshirish uchun taqsimlangan tranzaksiyanini qo'llash zarur;
  - yuqori unumidorlik eng asosiy maqsadlardan biri bo'lib, unga erishish uchun parallel BBBT texnologiyalari yo'naltirilgan. Odatda, buni ta'mintash uchun o'zaro bir-birini to'ldiruvchi yechimlardan birgalikda foydalaniladi. Masalan, berilganlar bazasini qo'llab-quvvatlashga yo'naltirilgan operatsion tizimlarni qo'llash, parallelash, optimizatsiyalash va yuklamalarni muvozanatlash shular jumlasiga kiradi;
  - taqsimlangan va parallel BBBT texnologiyasi ishonchlilikni oshi-

rishga ham yo'naltirilgan bo'lib, berilganlar replikatsiyasi sababli inkorni yagona nuqtalaridan voz kechiladi. Bitta tugunning inkori yoki aloqa liniyalarining ishlamay qolishi butun tizimning ishdan chiqishiga olib kelmaydi. Hattoki, agar berilganlarning bir qismidan foydalanish imkonni bo'lmagan holda ham, tizimni to'g'ri tashkillashtirish orqali foydalanuvchilar ma'lumotlarning boshqa qismlariga kirish imkoniga ega bo'ladilar. «To'g'ri tashkillashtirish» deganda, taqsimlangan tranzaksiyalar va ishonchliklarni ta'minlovchi bayonnomalarni qo'llab-quvvatlash tushuniladi;

- parallel va taqsimlangan BBBT muhitida berilganlar bazasining o'sish hajmi yoki qayta ishlovga ehtiyoji bilan bog'liq bo'lgan masalalar oson hal qilinadi. Bunday vaziyatlarda tizimlarni qayta qu'rish zarurati deyarli tug'ilmaydi. Imkoniyatlarni kengaytirish protsessor quvvatini yoki xotirani qo'shish hisobiga amalga oshiriladi;
- mukammal holatda, parallel va kamroq darajada taqsimlangan BBBT *chiziqli miqyoslash* (linear scaleup) xususiyatiga va *chiziqli tezlanish* (linear speedup) xususiyatiga ega. *Chiziqli miqyoslash* deganda, berilganlar bazasini hajmi oshishi va bir vaqtning o'zida proporsional ravishda protsessor quvvati hamda xotira hajmi oshishi orqali, unumdoorlikning o'sha darajasini saqlab qolish tushuniladi. *Chiziqli tezlanish* deganda, protsessor quvvati va xotira hajmini ortishi berilganlar bazasini oldingi hajmi saqlangan holda, uning unumdoorligi proporsional ravishda o'sishi tushuniladi. Bundan tashqari, tizimning kengayishi mavjud berilganlar bazasini minimal ravishda qayta tashkil qilishni talab qiladi;
- narxlar va unumdoorlikning o'zaro munosabatlarni, mikroprotsessorlar va ishchi stansiyalar uchun hisobga olgan holda, bitta katta kompyuter quvvatidagi tizimni amalga oshirgandan ko'ra, bir nechta katta bo'lmagan kompyuterlardan tizimni hosil qilish iqtisodiy tomonidan foydaliroqdir. Ko'pchilik tijorat taqsimlangan BBBT ham, narx/unumdoorlik munosabatlarning foydaliligi sababli, ishchi stansiyalar va mini-kompyuterlarda faoliyat yuritadi. Sababi, ko'pgina ishchi stansiyalardan foydalanishga asoslangan texnologiyalarning juda keng miq'yosda tarqalishiga, tijorat BBB tizimlarning asosan ishchi stansiyalardan foydalanuvchi lokal tarmoqlarda faoliyat yuritishi bilan bog'liq.

## **BERILGANLAR BAZALARINING SERVERLARI**

Mijozlarga xizmat ko'rsatishda kompyuter va programma vositalaridan iborat, lokal va global kompyuter tarmoqlaridagi serverlar – ishchi stansiya va/yoki boshqa serverlar keng qo'llaniladi.

Serverlarga misollar:

- hamma ishchi stansiyalardagi umumiy fayllar bazasini qo'llab-quvvatlovchi fayllar serveri;
- global tarmoq – Internetdagи axborotlami yetkazishni ta'minlovchi, internet-server;
- elektron pochta bilan ishlashni ta'minlovchi, pochta serveri;
- lokal tarmoqdan so'rovlanini qabul qilib, natijani qaytarishni ta'minlovchi, berilganlar bazalarining serveri.

*Berilganlar bazalarining serveri* deganda, «mijoz-server» arxitekturasiga asoslangan, server va mijoz qismlarini turli nisbatlarda o'z ichiga olgan, BBBT tushuniładi. Eng keng tarqalgan serverlar – Microsoft SQL Server, Oracle, IBM DB2 Universal DataBase, Informix va boshqalar. Ushbu serverlardagi bitta berilganlar bazasining hajmi million terabaytga yetishi mumkin.

### **Taqsimlangan ishlovning tayanch arxitekturalari**

Berilganlarga ishlov berish texnologiyasiga ko'ra BB markazlashtirilgan va taqsimlangan bo'limlardan iborat.

*Markazlashtigan BB* bitta hisoblash tizimining xotirasida saqlanadi. Bu hisoblash tizimi meynfreym bo'lishi mumkin – u holda, unga kirish terminallar orqali yoki shaxsiy kompyuterlar lokal tarmog'idagi fayl serveri amalga oshiradi.

*Taqsimlangan BB* bir nechta, hattoki kesishuvchi yoki bir-birini qaytaruvchi qismlardan iborat bo'lishi va hisoblash tarmog'ining turli tugunlarida saqlanishi mumkin. Bunday BB bilan ishlash *taqsimlangan berilganlar bazasini boshqarish tizimi* orqali olib boriladi.

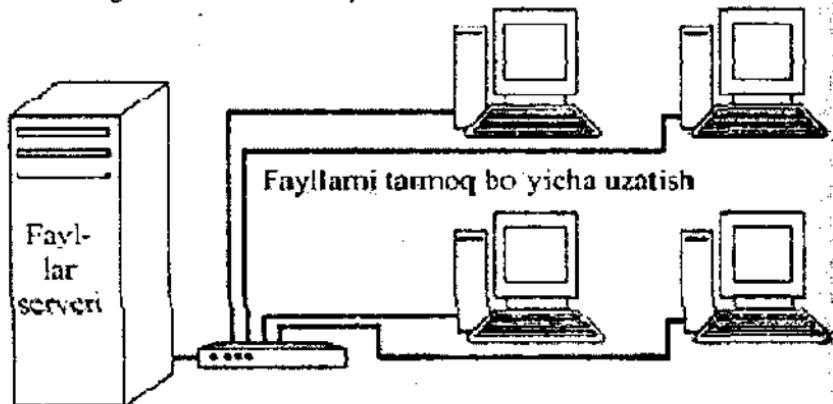
Berilganlarga kirish usuli bo'yicha BB *lokal kirish* va *tarmoq orqali kirishga* bo'linadi. Hozirgi barcha zamonaviy berilganlar bazasiga ko'p foydalanuvchili tarmoqdan kirish usulini qo'llash mumkin.

*Markazlashtigan berilganlar bazasiga tarmoqdan kirish quyidagi to'rtta turdagи arxitekuuraga ega:*

- «fayl-server»;
- «mijoz-server» BB;
- «nozik» mijoz - ilovalar serveri - BB serveri;
- berilganlar bazasining faol serveri.

## «Fayl-server» arxitekturasi

«Fayl-server» arxitekturasida, berilganlar bazasiga tarmoqdan kirish uchun kompyuterlarning biri markaziy mashina (fayllar serveri) sifatida olinadi. Alovida server uchun shu kompyuterda operatsion tizim o'mratiladi (misol uchun, Microsoft Windows Server 2008). Unda umumiy foydalaniladigan markazlashgan BB bir yoki bit nechta fayllar sifatida saqlanadi. Tarmoqdagagi qolgan barcha kompyuterlar ishchi stansiya funksiyalarini bajaradi (Microsoft Windows 2003/2007/2010 Professional yoki Microsoft Windows 2003/2007/2010 operatsion tizimlari bo'lishi mumkin). BB fayllari foydalanuvchilar so'roviga qarab ishchi stansiyalarga uzatiladi va ularda berilganlar qayta ishlashadi (31.1-rasm). Agarda berilganlar bazalarining *bitta fayliga ko'p murojaatlar bo'lsa*, axborot tizimining *samaradorligi pasayadi*. Bunday holatda, foydalanuvchilar o'zining ishchi stansiyasida lokal berilganlar bazasini hosil qilishi ham mumkin.

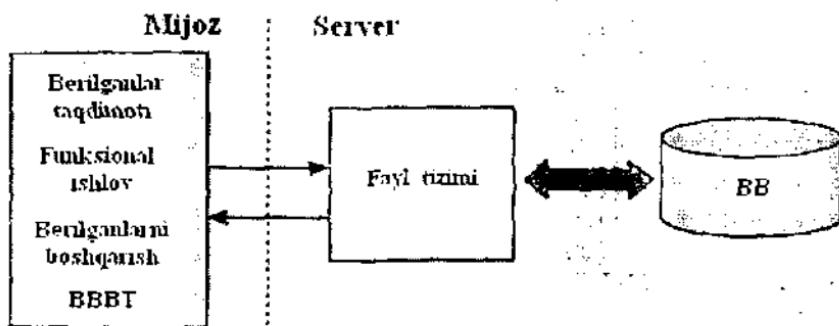


31.1-rasm. BB bilan ishlashda alovida fayl serverli lokal tarmoqda ishlash chizmasi.

«Fayl-server» arxitekturasida (31.2-rasm) berilganlarga maqsadli ishlov va boshqarish mijoz kompyuterida maxsus fayllardan iborat, BB esa server-kompyuterda saqlanadi. Bunda server tarkibi BBB tizimining vositalari bilan emas, balki masofali taqsimlangan fayllarga kirishni ta'minlovchi operatsion tizimni tarkibiy qismlari orqali ifodalanadi. Server berilganlarni saqlash, so'rovlarga xizmat ko'rsatish va olingan natijalarni foydalanuvchiga maxsus ishlov uchun jo'natadi. SQL tilidagi berilganlar bazasiga so'rov, BBT orqali kiritish-chiqarish buyruqlariga aylantiriladi va server-kompyuterning operatsion tizimida ularga ishlov beriladi.

Bunday ikki tarkibga ajratish quyidagi imkoniyatlarni yaratadi:

- berilganlar bazasi bilan bir vaqtida bir nechta amaliy program-malarning ishlashi;
- boshqarish funksiyalarini markazlashtirish, ya'ni berilganlar himoyasi va butunligi, zaxiralardan birlgilikda foydalanish;
- taqsimlangan berilganlar bazalarida so'rovlarning parallel ishlovini ta'minlash;
- tarmoq va ishchi stansiyalar zaxiralarini bo'shatish va/yoki yengil-lashtirish;
- berilganlar bazalarining maxsus serverlardan foydalanish hisobiga berilganlarni boshqarish unumdarligini oshirish.



31.2-rasm. «Fayl-server» arxitekturasi

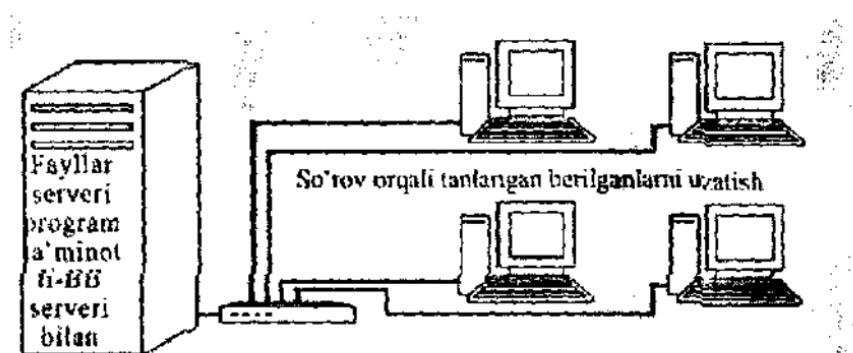
«Fayl-server» arxitekturasining kamchiliklari quyidagilardan iborat:

- tarmoqdagi va mijoz kompyuterlaridagi yuqori yuklama yuzaga keladi, chunki berilganlarni almashish fayl tizimidagi axborot birligi darajasida (fizik yozuvlar, bloklar va fayllar) bajariladi;
- berilganlarning himoyasi quyi darajada bo'ladi, chunki berilganlar bazasining fayllariga kirish server-kompyuterning operatsion tiziuning umumiyligini vositalari orqali boshqariladi;
- mijoz tomonda jamlangan funksional ishlovning tijorat qoidalari bir-biriga zid bo'lishi mumkin.

Berilganlar bazasining chizmalari uchun axborot ishlovining eng katta yig'ma vaqt sarf bo'lishi, «fayl-server» arxitekturasining nozik xususiyatlaridan bittasi hisoblanadi.

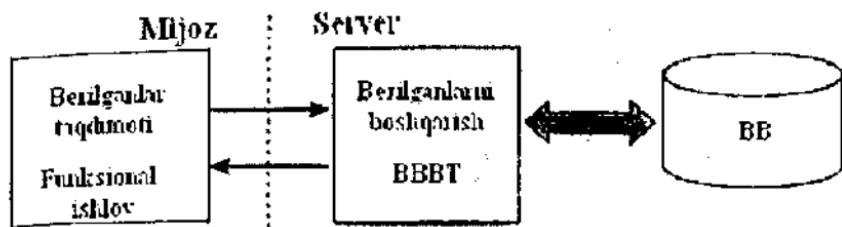
### «Mijoz-server» arxitekturasi

«Mijoz-server» arxitekturasida server OT orqali bosqariladigan ajratilgan serverda, maxsus programmalar ta'minoti esa BB serveriga o'tmatiladi (31.3-rasm). Bu arxitekturani *ajratilgan serverli arxitektura*, deb ham atashadi. Berilganlar bazasiga murojaat SQL tilida amalga oshirilgani uchun BB serverini *SQL-server*, deb ham atashadi. U hamma relatson BBBT (Oracle, Informix, MS SQL, DB2, ADABAS D, InterBase, SyBase) tomonidan qo'llab-quvvatlanadi.



31.3-rasm. BB arxitekturasida «mijoz-server» chizmasining ishlash negizi.

BBBT ikki qisimga bo'linadi: mijoz va server (31.4 rasm). BB serverning asosiy vazifasi SQL tilidan foydalanish. So'rovlar mijoz tomoni (ishehi stansiya)dan BB serveriga beriladi, so'ngra serverda qidiruv va berilantarni tanlab olish jarayoni amalga oshiriladi. Shartlarga javob beruvchi yozuvlar, serverda jamlanadi va so'rov ishllov qilingandan keyin foydalanuvchiga uzatiladi. Shu tarzda, tarmoqda ma'lumotlarni uzatish miqdori bir necha martaga kamayadi. Mijoz ilovalar stolbop BBB tizimlarida (MS Access, FoxPro, Paradox, Clipper) amalga oshirilishi mumkin. SQL-server bilan mijoz ilovalarning o'zaro harakati ODBC-drayver (Open Data Base Connectivity) orqali amalga oshiriladi. ODBC-drayver turli jinsdagi berilganlar bazalariga kirish algoritmining standartiga aylangan.



31.4-rasm. Berilganlar bazasining ajratilgan server arxitekturası.

«Mijoz-server» arxitekturasining afzalliklari quyidagilardan iborat:

- server va mijoz kompyuterlaridagi yuklama kamayadi;
- kirish-chiqishni optimallashtirish va buferlash evaziga ishlov unumdarligi oshadi va tarmoq grafigi kamayadi;
- BBT vositalari orqali berilganlarning himoyasi amalga oshirish natijasida, foydalanuvchi uchun ruxsat etilmagan harakatlarni to'xtatadi;
- server tranzaksiyalarni boshqarishni amalga oshiradi va bir vaqtda berilganlarni o'zgartirishga qilingan urinislarni to'xtatadi.

«Mijoz-server» arxitekturasining kamchiliklari esa quyidagilardan iborat:

- funksional ishlovnинг tijorat mantig'i va berilganlar tasavvuri bir qancha foydalanuvchilarning ilovalari uchun bir xilda bo'lishi mumkin, bu esa zaxiralarga bo'lgan talabni oshiradi, masalan programmlar kodi va so'rovlnarni takrorlash;
- mijoz tarafda jamlangan funksional ishlovnинг tijorat qoidalari bir-biriga zid bo'lishi mumkin.

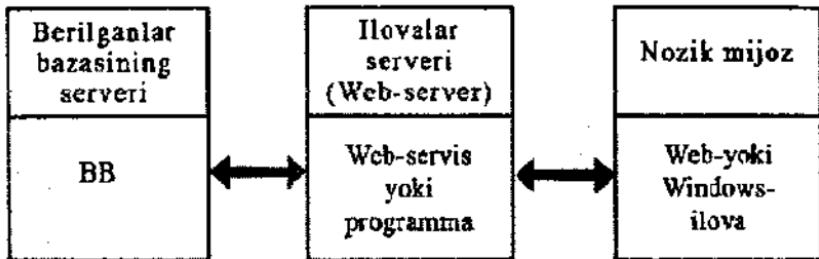
### «Uch bosqichli» arxitektura

«Uch bosqichli» arxitektura Internet va Intranet tarannoqlarida ishlaydi. Mijoz qismi («nozik» mijoz) foydalanuvchilar bilan o'zaro harakatlanuvchi qismi, Web-brauzerde HTML sahifani yoki Web-servisdagи Windows-ivilovalardan iborat. «Nozik» mijozlar eng sodda ilovalardan iborat bo'lib, faqatgina berilganlarni uzatish, lokal xeshlash, foydalanuvechi aloqa vositalarini tavsiflash, tahrirlash va oddiyagina ishlov vazifalarini ta'minlash bilan shug'ullanadi.

*Mijoz ilovalari BB serveriga to'g'ridan-to'g'ri murojaat qilmaydi, balki maxsus programma ta'minoti bo'lmish oraliq qatlam orqali ishlaydi. Bu bitta bo'g'in (oddiy «Uch bosqichli» arxitektura) yoki murakkab tuzilish bo'lishi mumkin. Maxsus programma ta'minoti bo'lmish oraliq qatlamni*

*Illovalar serveri* deyiladi, u mijozlarning so'rovlarini qabul qiladi, ularga programmalashtirilgan mantiqiy-tijorat qoidalariga asosan ishlov beradi, zarurat tug'ilganda BB serveriga mos shaklga o'zgartirib, unga uzatadi. *Illovalar serveri* sifatida Web-servis keladi (31.5 va 31.6-rasmlar). Misol uchun, Oracle Forms Server.

*BB serveri* qabul qilgan so'rovlni bajaradi va natijalarni ilovalar serveriga jo'natadi, u esa ishlov natijalarini mijozlarga qaytaradi.



31.5-rasm. «Uch bosqichli» arxitekturada BBning ishlash negizi.

Shunday qilib, BB «Uch bosqichli» arxitekturaning tarkibi quyidagi lardan tashkil topgan:

- berilganlarni uzatish, tavsiflash, tahrirlash va sodda ishlovnin ta'minlovchi «nozik» mijoz ilovalaridan;
- bitta kompyuterda yoki tarqatilgan holatda lokal tarmoqda, bir yoki bir nechta bo'g'linli oraliq qatlam hisoblanadigan – programmalar ta'minoti (*ilovalar serveri*) dan;
- BB faoliyatini qo'llab-quvvatlovchi va so'rovlargaga ishlov beruvchi – *BB serverlaridan* (Oralce, Sybase, MS SQL, InterBase va h.k.).

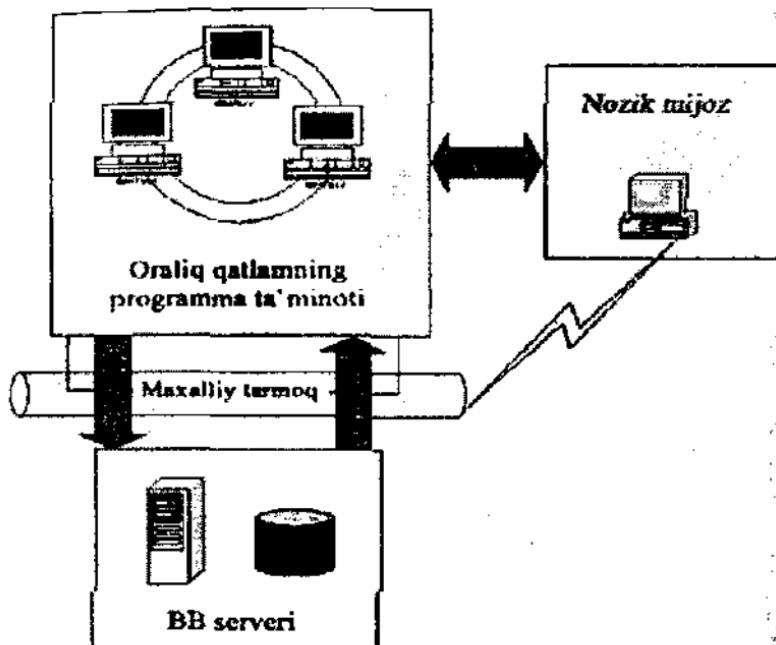
Ayrim programmalash muhitlarida (masalan, Delphi) mijoz programma ta'minoti va oraliq programma ta'minotini yaratish vositalari va tarkibili jihozlari mavjud. Ilovalar serveri BB serveri bilan aloqa qilganda, berilganlarga kirishdagi texnologiyalardan (ADO, BDE, InterBase Express, db Express va h.k.), qo'yilgan masalaga nisbatan birini tanlab foydalanadi.

Masofali mijoz ilovalarini yaratishda, maxsus tarkiblar termasidan iborat DataSnap texnologiyasidan foydalaniladi. Bu texnologiya standart transport vositalari (DCOM, HTTP, soketlar)dan foydalaniib, masofadagi mijoz ilovalarini ilovalar serveriga bog'lanishini ta'minlaydi. DataSnap tarkiblari AppServer aloqlaridan foydalaniib, ilovalar serveridagi funksiyalarga kirishni ham ta'minlaydi.

Mijoz kopyuterlari ko'pincha ojiz bo'lgani uchun, murakkab mantiqiy-tijorat masalalarini amalga oshirishda ilovalar serveri hamda foydalanuvchilarning bir jinsli so'rovlarini bajarishni optimallashtirish usullaridan foyda-

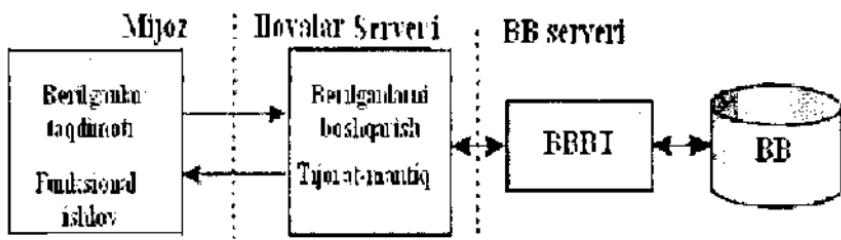
lanish, tizimni tezligini oshirishda muhim ahamiyatga ega. Masalan, BB serveri juda ham yuklangan bo'sha, ilovalar serveri mustaqil ravishda foydalanuvchi so'rovlariga ishlov berishi (ya'ni navbatga qo'yish yoki uni bekor qilish)ni BB serverini qoshimcha yuklamasdan o'zi bajarishi mumkin.

Zaxiralarga bo'lgan talablar darajasini keyinchalik kamaytirish ilovalar serverining kiritilishi hisobiga erishilgan. Ilovalar serveriga berilgani ami



36.6-rasm. BB ilovalarining «Uch bosqichli» arxitekturasi.

boshqarish programmasi tarkiblarining kattagina qismi va tijorat-mantiqning ham katta qista qismi o'tkaziladi. Bunda BB serverlari faqatgina BBBT funksiyalarini ta'minlaydi (31.7-rasm). Ilovalar serverida mijoz ishchi stanisiyalarining amaliy programmalar bilan aloqasi API (Application Programming Interface) aloqalari orqali o'rnatiladi. Mijoz qismining vazifasi ilovalar serveridagi funksiyalarni chaqirishdan iborat bo'ladi. Amaliy programmalar BB serveriga SQL so'rovlar yordamida murojaat qiladi.



31.7-rasm. Ilovalar serverining «uch bosqichli» arxitekturasi.

«Uch bosqichli» arxitekturaning afzallikkilari quyidagilardan iborat:

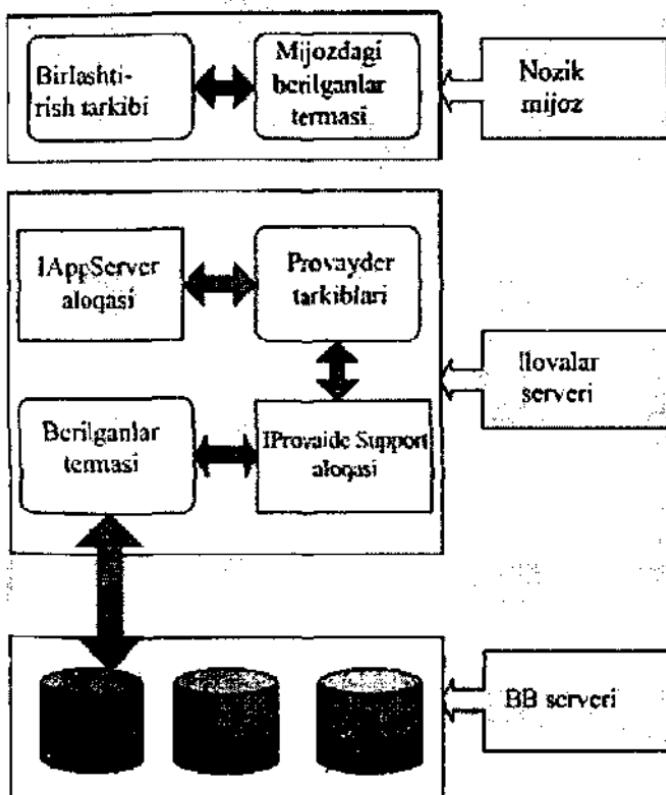
- berilganlarga ishvodagi umumiyligini funksiyalarni mijoz ilovalar to‘plamida ko‘p marotaba ishlatalishi;
- tijorat-mantiqni markazlashgan holda yuritish, ya’ni bazalarga o‘zgarishlar kiritilganda mijoz ilovalarda ularni nusxasini olmaslik;
- mijoz kompyuterlarda programma ta’mintonining berilganlarga kirishi boshqarish qismini o’tnatishni kerak emasligi;
- berilganlar bazasiga kirishni optimallashtirish (so‘rovlarini qayd etish ilovalar serverida bajariladi);
- avtonom holatda, berilganlar o‘zgartirilayotgan vaqtida, qoldirilgan berilgan bazasini yangilash imkoniyati berilganlar bazasiga, keyingi ularishda amalga oshiriladi;
- bir-birini o‘zaro almashtiruvchi, bir nechta ilovalar serveridagi programma ta’mintonini nusxalash natijasida tezlikni va ishonchlikni ortishi;
- foydalanuvchini vakolatlari tekshirishni BB serveridan ilovalar sereriga ko‘chirish;
- ilovalar serveri va BB serverining parallel ishlashi natijasida ilovalar serveri quvvatining kamayishi.

Ilovalar serverining mavjudligi tizimning xavfsizligini oshiradi, ya’ni unda foydalanuvchilarning mualliflik huquqini tekshirish va xavfsizlikka daxldor boshqa funksiyalarni ham amalga oshirish mumkin. Bundan tashqari, berilganlarni uzatishning himoyalangan kanallaridan (masalan, HTTPS) oson foydalanish mungkin.

Taqsimlangan «Uch bosqichli» arxitekturadagi ilovalarning tarkibiy qismlari 31.8- rasmida keltirilgan. Masalan, Delphi muhitida BB ilovalarini yaratishda xuddi ana shu rasmdagi tarkibiy qismlardan foydalangan holda yaratiladi.

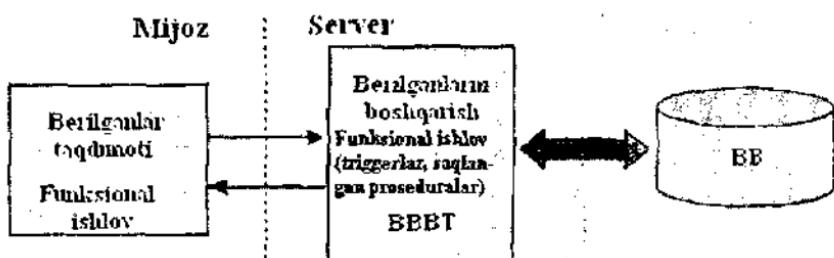
## Berilganlar bazasining «faol server» arxitekturası

*Berilganlar bazasining «faol server» arxitekturasida* (31.9-rasm), tijorat mantig'ining qarama-qarshiliklari server tomonidan nazorat qilinadi. Tijorat mantig'ining funksiyalari mijoz va server tomonlari orasida taqsimlanadi. Umumiyluq funksiyalar saqlanuvchi protseduralar sifatida tayyoranadi. Berilganlar bazasidagi hodisalarni kuzatish uchun triggerlar mexanizmi kiritiladi. Hodisa yuz bersa (odatda berilganlarni o'zgartirish) SQL operatori bajariladi yoki triggerga bog'liq saqlangan protsedura chaqiriladi.



31.8-rasm. Taqsimlangan ilovalarning «Uch bosqichli» arxitekturasidagi programma mahsulotlarining tarkibiy qismlari

Saqlanuvchi protseduralalar va triggerlar BB bilan ishlovchi ixtiyoriy mijoz ilovalar tomonidan foydalangan bo'lishi mumkin. Bu programma kodlarini nusxalashni kamaytiradi va har bir so'rovni komplatsiya qilish zaruratidan xalos etadi. Arxitekturaning kamchiligi server yuklamasining ortishi bilan bog'liq bo'ladi. Bunday arxitekturani ba'zan «nozik» mijozli model, deb atashadi. Avvalgi arxitekturalarning hammasini «semiz» mijozli model, deb atashgan, chunki unda mijoz tomonida ko'pgina funksiyalar amalgaga oshirilgan.



31.9-rasm. Berilganlar bazasining faol server arxitekturasi.

**Trigger** – bu BBBT yadro tomonidan bajariladigan programma bo'lib, ular BB jadvallarini tiklashdan (UPDATE, INSERT, DELETE) oldin va keyin bajariladi. Ya'ni triggerlar BB jadvallarini tiklashda (UPDATE, INSERT, DELETE) oldin va keyin, BBBT yadrosi tomonidan bajariladigan programmalardir. Ular berilganlar bazasini butunligini avtomatik ravishda ta'minkab beradi.

BB serveri modelini ORACLE, Sybase, Informix, Ingress, Progress kabi BBB tizimlari ta'minlaydi.

Berilganlar bazasining faol server arxitekturasining afzalliklari:

- bu BBBT asosida yaratilgan tizimlar yuqori ishlab chiqarishga ega, chunki shinalar bo'yicha faqat SQL so'rovlarini va bajarish natijalari uzatiladi;
- BBBT taqsimlangan qayta ishlashni ta'minlaydi;
- bu BBBT doirasida juda ko'p server programmalari bo'lib, ular ilovalar yaratishni va taqsimlangan tizimlar yaratishni yengil-lashtiradi.

**Kamchiligi:**

- ular oldindi sinf BBB tizimiga nisbatan qimmat, o'zlashtirish qiyin;
- ularni samarali ishlashi uchun yuqori tezlikli (shuning uchun qimmat) server va tarmoqlar talab etiladi.

## **So'rovlarini optimallashtirish va qayta ishlash**

Taqsimlangan berilganlar bazasini berilganlarni joylashtirish va ularni qidirish muhim masalalardan biridir. Berilganlarni qidirish berilganlarni tuzilishiga moslab yoki qiymatiga moslab amalga oshiriladi. 1- holda lokal berilganlar bazasida berilganlarni tuzilishi bir-biridan farq qiladi. 2- holda BB umumiyl tuzilishga ega bo'ladi, lokal berilganlar bazasining tuzilishida faqat aniq qiymatlar bilan farq qiladi. Taqsimlangan berilganlar bazasida ma'lumotlarni qidirishni quyidagi naqlari mavjud:

- foydalanuvchi eng yaqin tizim bilan o'zaro bog'langan, agar eng yaqin BBB tizimida kerakli ma'lumot bo'lmasa, unda qidirish berilganlar bazasida bajariladi;
- qidirilayotgan berilganlar tuzilishi axborot bo'yicha tugunlarda amalga oshiriladi. Bunday berilganlar tuzilishi barcha lokal tizimlarda saqlanishi lozim.

Bunda bitta tizim ajratib boshqaruvchi deb hisoblanadi va unda barcha berilgannarning tuzilishlari saqlanadi. Foydalanuvchi 1- qidirishda kerakli berilganni topolmasa, u berilgannarning tuzilishlariga murojaat qiladi.

*So'rovlar ishlovi* (query processing) – bu past darajadagi berilganlarga ishlov amalini translatsiya qilish jarayonidagi so'rovning rasmiy aniqlanishi. Zamonaviy BBBTlarini qo'llab-quvvatlovchi standart so'rovlar tili SQL hisoblanadi. *So'rovlarini optimallashtirish* (query optimization) – bu so'rovni bajarish bo'yicha muqobil to'plamdan «eng yaxshi» strategiyani tanlab olish protsedurasidir.

Odatda markazlashgan BBBT uchun bu jarayon ikkita qadamdan iborat: *so'rov dekompozitsiyasi* (query decomposition) va *so'rovni optimallashtirish*. *So'rov dekompozitsiyasi* – bu so'rovni SQL tilidan relatsion algebra ifodalariga o'tkazishdan iborat. Bunda so'rov dekompozitsiyasi semantik tahlildan o'tadi, ya'ni aniq bo'limgan so'rovlardan voz kechiladi, aniqlari esa soddalashtiriladi. Soddalashtirishda, xususan tasavvurlardan foydalanish hisobiga ortiqcha paydo bo'lgan predikatlarni hisobga olmagan holda, xavfsizlik chegaralari va semantik butunlikdan kelib chiqib amalga oshiriladi. Soddalashgan so'rov algebrayik shaklga o'tkaziladi.

Berilgan SQL so'rovda bittadan ko'p bo'lgan algebrayik ifodalar mavjud bo'lishi va ularning ba'zilari boshqalaridan «yaxshi» bo'lishi mumkin. Algebraik ifodalarning «sifati» hisoblash uchun zarur bo'lgan xarajat hajmidan kelib chiqqan holda aniqlanadi. An'anaviy protsedura shundan iboratki, avval SQL so'rovni biror ifodaga translatsiya qilinadi, so'ngra ekvivalent algebrayik o'zgartirishlar qoidasini qo'llab, toki «eng yaxshisi» topilmaguncha, so'rovni bajarish uchun zarur bo'lgan boshqa algebrayik ifoda olinadi. «Eng yaxshi» ifodani qidirishda, xarajatlar yig'indisini

*hisoblovchi narx funksiyasi ishlataladi. Bu jarayon so'rovlarini optimalkashirish deyiladi.*

Taqsimlangan BBB tizimlaridagi so'rovlar dekompozitsiyasi va optimatsiya qadamlari orasiga yana ikkita harakatlar qoshiladi, bular *berilganlarni lokallashtirish* (*data localization*) va *so'rovlarini global optimalkashirish* (*global query optimization*)dan iborat.

Berilganlarni lokal lashtirish uchun boshlang'ich ma'lumotlar sifatida so'rov dekompozitsiyasi qadamida olingan boshlang'ich algebrayik ifodalar xizmat qildi. Boshlang'ich algebrayik ifodalarda, ularning bo'laklanishi yoki taqsimlanishini hisobga olmagan holda, global munosabatlар ishlatiadi. Berilganlarni lokallashtirishdagi asosiy vazifa, berilganlarni taqsimlanishi xususidagi ma'lumotlardan foydalangan holda, so'rovda ishtirot etgan berilganlarni lokallahdan iborat. Bu qadamda, so'rovda ishtirot etuvchi bo'laklar aniqlanadi va so'rov shakli shunday o'zgartiriladiki, undagi amalgar global munosabatlarga emas, balki bo'laklarga qo'llaniladi.

Yuqorida aytib o'tilganidek, bo'laklash qoidasi *relatsion jarayonlar vosisatini ifodalaydi* (tanlash uchun gorizontal bo'laklash va nusxalash uchun vertikal bo'laklash). *Ikkinci qadamda*, taqsimlangan munosabatlар bo'laklash qoidasiga inversiyani qo'llash yo'li orgali qaytadan quriladi. Bu *lokallashtirish programmasi* deyiladi. Gorizontal (vertikal) bo'laklangan munosabatlар uchun lokallashtirish programmasi mos bo'laklarni *birlashtirishdan* (union) iborat. Shunday qilib, berilganlarni lokallashtirish qadamida har bir global munosabatlар so'ovi uni lokallashtirish programmasiga almashtiradi, keyin esa natijaviy bo'lakli so'rov soddalashtiriladi va boshqa «yaxshi» so'rovni olish maqsadida qayta tarkiblashtiriladi. Soddalashtirish va qayta tarkiblashtirish uchun dekompozitsiya qadamiidagi qoidalardan foydalangan mumkin. Dekompozitsiya qadamiidagi kabi, bo'laklash ustidagi oxirgi so'rov optimaldan ancha yiroq bo'lishi mumkin. Ushbu jarayon faqatgina «yomon» algebrayik so'rovlarini istisno qiladi.

*Uchinchi qadam* uchun boshlang'ich ma'lumotlar bo'laklangan so'rov hisoblanadi, ya'ni bo'laklash ustidagi algebrayik ifoda. Global optimalkashirishning maqsadi, optimalga yaqin bo'lgan so'rovni bajarilish strategiyasini topishdir. Optimal strategiyani topish – hisoblash qiyin bo'lgan masaladir. Taqsimlangan so'rovlar bajarilish strategiyasini, tugunlararo berilganlarni uzatish uchun foydalilanidigan *relatsion algebra jarayonlari* va *kommunikatsion oddiylik* atamalari bilan ifodalash mumkin. Oldingi qadamlarda, xususiy holda ortiqcha ifodalarni o'chirish hisobiga so'rovlar qaysidir darajada optimallashtirilgan edi. Lekin optimallashtirishni amalga oshirish fragmentlar tavsiflariga, masalan uning quvvatiga, bog'liq emas edi. Bulardan tashqari oldingi qadamlarda kommunikatsion jarayonlar hali hisobga olinmagan edi. Bitta bo'laklangan so'rovning ichida amallar tartilini

o'zgartirish yo'li bilan uni bajarilishining bir qancha ekvilavent rejalarini hosil qilish mumkin. So'rovni optimallashtirish, optimizator o'rganayotgan tadqiqot o'tkazayotgan rejalar to'plamidan «eng yaxshi» rejani qidirib topishdan iborat.

So'rovlar optimizatori odatda uch tarkibili ko'rinishda tasvirlanadi: *qidiruv fazosi, narx modeli va qidiruv strategiyasi*. *Qidiruv fazosi* – bu boshlang'ich so'rovni bajarishdagi muqobil rejalar to'plamidir. Bu rejalar bir xil natija berish ma'nosida ekvivalent, lekin alohida amallarni bajarilish tartiblari va uslublari bilan farqlanadi. *Narx modeli* – bu shu so'rovni bajarilish rejasining naxxini baholash usulidir. Aniqlikka erishish uchun narx modeli parallel hisoblashlar muhitli haqidagi aniq bilimlarga asoslanishi kerak. *Qidiruv strategiyasi* – bu qidiruv fazosini aylanib o'tish va yaxshi rejani tanlab olish usulidir. U qaysi rejani va qaysi yo'l bilan ularni tanlab olish va baholashni aniqlaydi.

Taqsimlangan muhitlardagi ko'pincha vaqt birligi bilan aniqlanadigan narx funksiyasi disk fazosi, disklar bilan almashish soni, markaziy protsesor vaqt, kommunikatsiyalar kabi hisoblash vositalari xarajatlarni baholaydi. Odatda, bular kirish-chiqish, markaziy protsessor va kommunikatsiyaarning mo'ljallangan xarajatlari yig'indisidan iborat. Taqsimlangan BBB tizimida soddalashtirilgan yondashuv qo'llaniladi, eng ahamiyatlisi sifatida faqatgina kommunikatsion xarajatlar qaratadi. Bu lokal qayta ishlashga qaraganda, berilganlarni jo'natuvchi aloqa liniyalarining o'tkazuvchanlik imkoniyatlari chegaralanganligi hisobiga yetarli darajada qimmat bo'lgan global to'rlar uchun o'rinnidir. Amallarning bajarilish tartibini aniqlash uchun boshqa tartibda amallar rejolarining narxlarini baholash zarur. So'rovlarning aniq bajarilishiga bajarilish narxini aniqlash (statik optimallashtirish) relation amallar natijalarining quvvatlarini baholash uchun formulalarga va bo'lakar statistikasiga asoslanadi. Shu tariqa, optimallashtirish paytida qabul qilingan qarorlar mayjud bo'lgan bo'laklar statistikasiga bog'liqdir.

So'rovnini optimallashtirishning asosiy ko'rsatgichi, birlashtirish amalini bajarilishining tartibiga bog'liq, chunki uning o'zgarishi bir necha barobar tezlanishga olib kelishi mumkin. Taqsimlangan amallar ketma-ketligini birlashtirishdagi optimallashtirishning tayanch usuli *yarim birlashtirish* (semijoin) amalini qo'llashdan iborat. Taqsimlangan tizimlardagi yarim birlashtirishning asosiy afzalligi bu – birlashtirishda qatnashuvchi operandar o'chami va shundan kelib chiqqan holda, kommunikatsion xarajatlarni qisqartirishdan iborat. Global optimizator ishining natijasi bo'laklar ustida kommunikatsion amallarni o'z ichiga oluvchi optimallashtirilgan algebrayik ifodalar hisoblanadi.

So'rovlarga parallel ishlov, umuman olganda taqsimlangan so'rovlar ishloviga o'xshaydi. U yuqorida muhokama qilingan ichki so'rovlar par-

Jelligining ustunligi hamda amallararo parallellash afzalliklariga tayanadi.

Ichki amallar parallelizmiga amallarning ko'p protsessorli mashina tugunlarida bir qanchasining baravar bajarilish hisobiga erishiladi. Buning uchun operandlarni avvalroq bo'lish zarur, ya'ni ulaming tugunlar bo'yicha gorizontal bo'laklash kerak. Asosiy munosabatni bo'lish ushibi berilganlar bazalarining fizik loyihalash sohasiga tegishlidir. Odatta bo'lish amali, munosabatning birlashtirish uchun xizmat qiluvchi atributiga nisbatan qaysidir xesh-funksiyani qo'llash yo'li bilan amalgalash oshiriladi. Munosabatlar saqlanuvchi tugunlar to'plami, *uy to'plami* deyiladi. *Amallar tugunlarining uy to'plami* deb, u bajariladigan tugunlar to'plamiga aytildi. Bu to'plam *uy to'plami* tugunlaridagi operandlar bilan mos tushishi kerak, chunki amallarning o'ziga tegishli operandlarga kira olishi kerak. Bu shuni ko'rsatadiki, birlashtirish kabi binar amallar uchun operandlardan birini qayta bo'lish zarurati tug'ilishi mumkin. Ba'zi bir hollarda optimizator ikkala operandlarni ham qayta taqsimlash maqsadga muvofiq deyishi mumkin.

Parallel BBBTlarida ichki amallararo parallellashni amalgalashga tatbiq qilish mumkin bo'lgan ba'zi usullar taqsimlangan berilganlar bazalarini uchun ishlab chiqilgan. Amallararo parallelash, berilganlarning umumiyoqimiga bog'liq holda yoki mustaqil ravishda bir vaqtning o'zida ikkita yoki undan ortiq amallar bajarilganda o'rini bo'ladi. Berilganlar oqimi atamasini biz konveyer ishlov usullarini amalgalashga oshiruvchi parallelash shakli orqali belgilaymiz. Mustaqil parallelashda amallar vaqt bo'yicha yoki ixtiyoriy taitibda bajariladi. Agarda amallar operandlar sifatida umumiy berilganlardan iborat bo'lmasa, u holda bog'liqsiz parallelash holati yuzaga kelishi mumkin.

### Replikatsiyalar qaydnomalari

Replikatsiyani qo'llab-quvvatlovchi taqsimlangan berilganlar bazasida berilganlarning har bir mantiqiy elementiga bir qancha fizik nusxalar mos keladi. Masalan, talabalar ballari (berilganlarning mantiqiy elementi) beshta tugunda (fizik nusxalar) saqlanishi mumkin. Bunday tizimlarda nusxalarni muvofiqlashtirish bo'yicha muammo yuzaga keladi. Tranzaksiyan yakunlanishiда berilganlarning barcha mantiqiy elementlari nusxalari o'xshash bo'lishini talab qiluvchi muvofiqlashtirishning yanada taniqli alomatlaridan biri, nusxalarning to'liq ekvivalentligi alomati hisoblanadi.

Agar replikatsiyaning shaxsifligi qo'llab-quvvatlansa, u holda berilganlarning mantiqiy elementi Zga tegishli bo'lgan o'qish-yozish amallarini tranzaksiya beradi. Replikani boshqarish qaydnomasi  $Z(z_1, z_2, \dots, z_n)$ ning fizik nusxalari ustidagi amallarda Z ustidagi amallarning akslanishiga javob beradi.

Replikalarni boshqarishning namunaviy qaydnomalari alomat bo'yicha nusxaning to'liq ekvivalenti bo'lib, ROWA (Read-Once/Write-All – bir nusxadan o'qish, barcha nusxalarga yozish) nomi bilan taniqlidir. ROWA qaydnomasi qaysidir  $Z$  [Read( $Z$ )] o'qishni qaysidir fizik  $Z_i$  [Read( $Z_i$ )] nusxadan o'qish amaliga akslantiradi. Aynan qaysi nusxadan o'qishni amalga oshirayotganligi ahamiyatga ega emas, chunki bu samaradorlik nuqtayi nazaridan hal qilinishi mumkin. Berilganlarning har bir mantiqiy elementi  $Z_g$  yozish amali berilganlarning barcha  $Z$  fizik nusxalari to'plamiga yozuv amali akslanadi. ROWA qaydnomasi sodda va to'g'ri chiziqli, lekin transaksiyani terminallash mumkin bo'lishi uchun, u berilganlarning barcha elementlari nusxalariga kirishni talab qiladi. Bitta tugunning ishdan chiqishi tranzaksiyani inqiroziga olib keladi, bu esa berilganlar bazasiga kirishni kamaytiradi.

### Xulosa

Taqsimlangan BB – bu taqsimlangan kompyuter tarmog'ida mantiqan o'zaro bog'langan berilganlar bazalari to'plamidan iborat. Taqsimlangan BBBT, taqsimlangan berilganlar hazalarini shunday boshqarish imkonini beradigan programma tizimi kabi aniqlanadi. Bundagi berilganlar bazalarning taqsimlanganligi foydalanuvchilar uchun shaffof bo'ladi. Bu arxitekturada ikkita farqlanuvchi xususiyatlar mavjud. Birinchisi, tizim so'rovlarni qabul qiluvchi tugunlar to'plamidan va bo'sh bo'lmagan berilganlarning tuguntari to'plamidan iborat.

Ikkinci xususiyatda esa, tugunlar mantiqan mustaqil kompyuterlar deb tasavvur qilinadi.

BB berilganlar bo'laktari va repliksiyalar (o'ziga o'xshash tuzilishlar) asosida fizik ravishda berilganlar tugurlarga taqsimlanadi. Relatsion berilganlarning chizmalari mavjud bo'lganda, har bir munosabat gorizonttal va vertikal qismlarga bo'linadi.

Gorizontal ajratish, tanlash amali yordamida bo'laklarga ajratish predikati qoidasiga amal qilgan holda, munosabatning har bir kortejini biorita tugunga jo'natish orqali amalga oshiriladi.

Vertikal ajratishda munosabatni nusxalash amali yordamida bo'laklarga ajratiladi.

BB programma ta'minotida ilovalar orqali berilganlar bilan shiddatlil ishlovda parallelashning uchta turi qaratildi – so'rovlararo parallelash; ichki so'rovlardagi parallelash; ichki operation parallelash.

Taqsimlangan BB «Uch bosqichli» arxitekturasi – «nozik» ilovalari, programmalar ta'minoti (ilovalar serveri) va BB serverlaridan tashkil topgan.

## Nazorat uchun savollar

1. Taqsimlangan BBB tizimining asosiy vazifasi nimadan iborat?
2. Bir jinsli taqsimlangan BB deganda nimani tushunasiz?
3. Bir jinsli bo'lмаган тақсимланған BB deganda nimani tushunasiz?
4. Illovalar serverining asosiy vazifasi nimadan iborat?
5. BB serveri deganda nimani tushunasiz?
6. Taqsimlangan BB deganda nimani tushunasiz?
7. Taqsimlangan BB shaffofligining birinchi xususiyati nimadan iborat?
8. Taqsimlangan BB shaffofligining ikkinchi xususiyati nimadan iborat?
9. Relatsion BB chizmalari mavjud bo'lsa, munosabat qanday qismrlarga bo'linishi mumkin?
10. Berilganlar bazasining programma ta'minoti bilan shiddatliz ishlovda parallelilikning qanday turi mavjud?
11. Taqsimlangan va parallel berilganlar bazasining asosiy nechta xususiyati mavjud?
12. Markazlashgan BB deganda nimani tushunasiz?
13. Markazlashgan BB arxitekturasi nechta turdan iborat va ular qanday nomlanadi?
14. «Fayl-server» arxitekturasining ishlash negiziga tushuncha bering?
15. «Fayl-server» arxitekturasining avzalliklar va kamchiliklari nimadan iborat?
16. «Mijoz-server» arxitekturasining ishlash negiziga tushuncha bering?
17. «Mijoz-server» arxitekturasining avzalliklar va kamchiliklari nimadan iborat?
18. ««Uch bosqichli»» arxitekturaning ishlash negiziga tushuncha bering?
19. ««Uch bosqichli»» arxitekturasining avzalliklar va kamchiliklari nimadan iborat?
20. «Faol server» arxitekturasining ishlash negiziga tushuncha bering?
21. Triggerni ta'riflang.
22. «Faol server» arxitekturasining avzalliklar va kamchiliklari nimadan iborat?
23. So'rovilarga ishlov, deganda nimani tushunasiz?
24. So'rovlarni optimizatsiyalash, deganda nimani tushunasiz?
25. So'rovlarni dekompozitsiyalash, deganda nimani tushunasiz?
26. So'rovilar optimizatori qanday tarkiblardan iborat?
27. Replikatsiyalarni boshqarish qaydnomasi ROWA ning ishlash tamoyilini tushuntiring.

## VII-BO'LIM. RELATSION BBB TIZIMLARI

### 32- bob. MS ACCESS BERILGANLAR BAZASINI BOSHQARISH TIZIMI

**Tayanch iboralar:** *relation BB, relation BBB tizimlari, MS Access, MS Jet database engine, MS Visual Basic, MS Visual Basic for Applications, jadval, maydon, jadval o'lchami, MEMO maydoni, OLE obyekt, jadvaldag'i indekslar soni, indeksdag'i maydonlar soni, o'rnatilgan aloqalari soni, so'rovlarda jadvallar soni, yozuvlar to'plamidagi maydonlar soni, yozuvlar to'plamining o'lchami, saralash chegarasi, WHERE, HAVING, AND, shakl, hisobot, makros, makrohuyrug, string, byte, integer, long, single, double, date, currency, boolean, admins guruhi, users guruhi, Talabalar o'quv toyihasi, berilganlar chizmasi, aloqalarni o'zgartirish.*

#### Tizimning umumiy tasnifi

MS Access tizimi MS Office ning asosiy tarkibiy qismlaridan biri hisobanib, relationsion BB bilan ishlash uchun mo'ljalangan. Mazkur BB bilan ishlash tizimining alohida jihat: berilganlar bazasining barcha axborotlari bitta faylda saqlanadi (\*.mdb). Berilganlar jadvallaridan tashqari shu faylini o'zida BB bilan ishlash uchun ilovalarni tarkibiy qismlari-ekranli shakllar, hisobotlar talab-so'rovnomalar, programmalash modullari saqlanadi.

BB bilan ishlash uchun tizim MS Jet database engine- BBni boshqarish tizimidan foydalanadi. Ushbu tizim foydalanuvchilar va tizimdag'i masalalarning berilganlarini tanlab olish va saqlash uchun xizmat qiladi. MS Jet BB yadrosiga berilganlarning dispetcheri sifatida qarash mumkin, chunki u yordamida berilganlardan foydalanishning boshqa tizimlari, jumladan MS Access va MS Visual Basic tashkil topadi. BB bilan ishlash uchun programmalash modullarini yozilish tili - MS Visual Basic for Applications (VBA) programmasi hisoblanadi.

Tizimni BB bilan ishlash jarayonidagi asosiy imkoniyat ko'rsatkichlari 32.1-jadvalda keltirilgan.

## 32.1-jadval

**MS Office Access 2003 tizimini bat afsil ko'rsatkichlari**

<b>Xususiyat (atribut)</b>	<b>Maksimal qiymati</b>
<b>BB</b>	
<b>BB faylining o'Ichami (*.mdb)</b>	2 Gbayt tizimli hujjatlar va obyektlar uchun zarur bo'lgan joyni hisobga olmaganda.
<b>BB obyektlar soni</b>	32 768
Modullar (jumladan shakllar va hisobotlar), True ma'nosiga ega modulni mavjudlik xususiyati (Has Module)	1 000
Obyektlar nomida belgilari soni	64
Maxfiy so'z (parol)da belgilari soni	14
Foydalanuvchi va guruhlari nomida belgilari soni	20
Bir vaqtning o'zida ishlovchi foydalanuvchilar soni	255
<b>Jadval</b>	
Jadval nomidagi belgilari soni	64
Maydon nomidagi belgilari soni	64
Jadvalda maydonlar soni	255
Ochiq jadvallar soni	2048 (MS Access ochuvchi ichki jadvallar tufayli aniq raqam kamroq bo'lishi mumkin)
Jadval o'Ichami	2 Gb tizimli hujjatlar va obyektlar uchun zarur bo'lgan joyni hisobga olmaganda
Matnli maydonda belgilari soni	255
MEMO maydonida belgilari soni	65 535 foydalanuvchi aloqasi orqali berilganlarni kiritganda; programma tasnif

	yordamida berilganlarni kiritganda belgilarni saqlash uchun 1 Gb
OLE obyekt maydonining o'lchami	1 Gb
Jadvaldagi indekslar soni	32
Indeksdagisi maydonlar soni	10
Xatolar haqidagi xabarnomalarda belgilari soni	255
Yozuvlarga qo'yilgan shartdagi belgilari soni	2048
Jadval yoki maydonni tavsifidagi belgilari soni	255
Yozuvlarda belgilari soni (MEMO maydonlari va OLE obyektlar maydonlaridan tashqari)	2000
Maydon xususiyatidagi belgilari soni	255
<b>So'rovlar</b>	
O'matilgan aloqalar soni	Aloqalarda ishtirok etmaydigan maydonlar yoki maydonlar birikmasi uchun no'ljallangan jadvallarda joylashgan indekslar soni hisobga olinmaganda, bitta jadval uchun 32
So'rovlardagi jadvallar soni	32
Yozuvlar to'plamidagi maydonlar soni	255
Yozuvlar to'plamining o'lchami	1 Gbayt
Saralash chegarasi	255 belgilari bir yoki bir necha maydonlarda
So'rovlaning ichma-ich qo'yilishidagi darajalar soni	50
So'rovlar blankasi katakchasiidagi belgilari soni	1024
Parametrlar so'roviidagi parametrlar uchun belgilari soni	255
WHERE yoki HAVING ifodalarida AND operatorlarining soni	99

<b>SQL yo'riqnomasida belgilar soni</b>	<b>Taxminan 64000</b>
<b>Shakl va hisobot</b>	
Yozuvdagagi belgilar soni	2048
Maydondagagi belgilar soni	65535
Shakl va hisobotni kengligi (eni)	22 dyuym (55,87 sm)
Bo'lim balandligi	22 dyuym (55,87 sm)
Barcha bo'limlar va bo'limlar sarlavhasining balandligi (konstruktor rejimida/tartibida)	200 dyuym (508 sm)
Shakl va hisobotlarning ichma-ich qo'yilishidagi darajalar soni	7
Hisobotda saralash yoki guruhlash mumkin bo'lgan maydonlar yoki so'z ifodalari soni	10
Hisobotda izohlar/eslatmalar va sarlavhalar soni	Hisobotning 1 ta sarlavhasi/ izohi; sahifaning 1 ta sarlav- hasi/izohi; guruhlar- ning 10 ta 1 ta sarlavhasi/izohi
Hisobotda nashr etiladigan sahifalar soni	65536
Shakl va hisobot mavjud bo'lgan davrda qo'shish mumkin bo'lgan bo'limlar va boshqarish elementlari soni	754
Shakl, hisobotlar yoki boshqarish elementlarining yozuvlari manbasi (RecordSource) yoki satrlar manbasi (RowSource) xususiyati sifatida SQL yo'riqnomasida ishllovchi belgilar soni (ikkisi .mdb va .adp)	32750
<b>Makros</b>	
Makrosdagi makrobuyruqlar soni	999
Shartlardagi belgilar soni	255
Izohlardagi belgilar soni	255
Makrobuyruqlar argumentidagi belgilar soni	255

32.2-jadvalda jadval maydonlariga ega bo'lishi mumkin bo'lgan berilganlar turlari to'g'risida to'liq ma'lumotlar keltirilgan.

**Tizimdag'i berilganlarning turlari**

<b>Maydonagi berilganlar turi</b>	<b>VBA da berilganlar turi</b>	<b>Foydalanish</b>	<b>O'chami</b>
Matnli	String	Unicode (belgi uchun 2 bayt) kodlash tizimida mavjud bo'lgan har qanday belgilardan tuzilgan matn	255 belgilargacha
MEMO maydoni	String	Unicode kodlash tizimdag'i matn	64000 belgilargacha
Raqamli (bayt, butun, uzun butun, oddiy suzuvchi nuqtali, ikkilangan suzuvchi nuqtali, replikatsiya kodi, haqiqiy)	Byte, Integer, Long, Single, Double	Raqamli berilganlar	1, 2, 4 yoki 8 bayt. 16 bayt faqat replikatsiya kodlari uchun (GUID)
Sana va vaqt. Sanani to'liq formati. Sanani o'rtacha formati. Sanani qisqa formati. Vaqtni uzun formati. Vaqtni o'rtacha formati. Vaqtni qisqa formati.	Date	Sana va vaqt. 31 dekabr 2005y 31-dek-04 31.12.04 23:55:59 23:55:59 23:55	8 bayt (maydon faollashtirilganda doimo sanani to'liq formatini ko'rsatadi)
Pulli/pulga oid	Currency	Valuta qiymati. Hisoblash jarayonida yaxlitlashni oldini olish uchun pullik shakldagi	8 bayt

<b>Hisoblagich</b>		formatdan foydalaniladi. Senning butun qismida 15 ta va kasr qismida 4 ta raqam bo'lishi mumkin.
<b>Mantiqiy</b>	<b>Boolean</b>	Yozuvlar qo'shilganda tartib yoki tasodifiy raqamlarni avtomatik tarzda qo'shish (1 ga ortib borishi) 4 bayt. 16 bayt faqt replikatsiya kodlari uchun (GUID)
<b>OLE obyekti maydonida</b>	<b>String</b>	Faqat 2 ta qiymatdan iborat maydonlar jumladan, Ha/Yo'q, Rost/Yolg'on, Yoqish/O'chirish. 1 bit
<b>Giperhavola</b>	<b>String</b>	OLE qaydnomasi dan foydalananuvchi programmalarda tuzilgan obyektlar (masalan, MS Word hujjatlari, MS Excel ning elektron jadvallari, rasmlar, tovushlar va boshqa ikkilik berilganlar). Obyektlar o'zaro bog'langan yoki joylangan bo'lishi mumkin. 1 Gb gacha (diskni hajmi bilan chegaralangan)
		<b>64000 ta belgilargacha</b>

O'rin  
almashtirish  
Мастери (устаси)

Ro'yxatli maydon-  
dan foydalangan  
holda, boshqa jad-  
valdan yoki so'z  
ifodalari ro'yxatidan  
qiymatlarni saralab  
olishga imkon  
beradigan maydonni  
tashkil etadi.  
Mazkur xizmat turi  
tanlanganda, ushbu  
maydonni aniqlash  
uchun Macrep ishga  
tushiriladi

MS Access tizimi foydalanuvchilarni BBga kirish huquqini chegaralish maqsadida o'zining shaxsiy imkoniyatlariiga ega.

BBga kirishni chegaralashning eng oddiy usuli bu BBni ochish uchun maxfiy so'z (parol)ni o'rnatish hisoblanadi (\*.mdb). Maxfiy so'z o'rnatilgandan so'ng, har vaqt BB ochilishidan oldin muloqot oynasi paydo bo'ladi. Xuddi shu oynaga maxfiy so'z kiritilishi lozim. Ushbu usul yetarlicha ishonchlidir (MS Access maxfiy so'zni shifrlaydi, shuning uchun BB faylini bevosita o'qishda unga kirishni iloji yo'q), lekin ushbu usul faqat BB ochilgan paytda ishlaydi. Baza ochilgandan keyin foydalanuvchi barcha hujjatlar va obyektlardan bemalel foydalanish imkoniyatiga ega bo'ladi (ushbu bo'linda quyida keltirilgan himoyaning boshqa turlari hali aniqlanmagan). Foydalanuvchilarning kichik guruhlari bilan birgalikda yoki shaxsiy kompyuterlar tomonidan foydalaniладigan BB uchun odatda maxfiy so'z (parol)ni o'rnatishning o'zi kifoya qiladi.

**BBni shifrlash mumkin.** BB shifrlanganda uning fayllari zichtanadi va uni xizmat ko'rsatuvchi programmalar yoki matn tahrirlovchi programmalar yordamida o'qish mumkin bo'lmay qoladi. BBni shifrini ochish shifrlash jarayonlarining natijalarini bekor qiladi.

Agar BBni replikatsiyasini analaga oshirish ko'zda tutilsa, BBga maxfiy so'z (parol)ni o'rnatish mumkin emas. Agar BBga parol o'rnatilgan bo'lsa, replikatsiya qilingan BB sinxronlashtirilmaydi (bir-biriga muvofiq lashtirilmaydi).

**Foydalanuvchilar darajasidagi himoya huquqlarni chegaralashda keng imkoniyatlarga ega.** Ushbu usul ko'pgina tarmoqli tizimlar foydalaniб keladigan usullar bilan o'xshash.

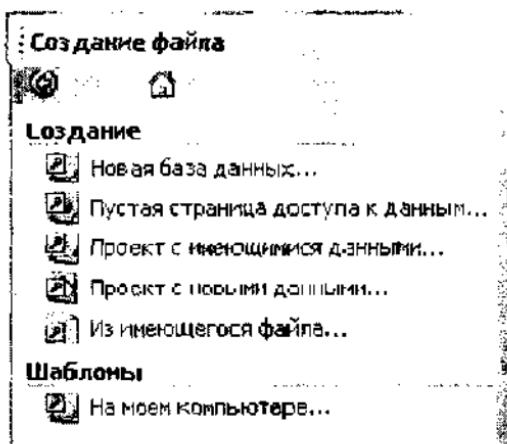
MS Access foydalanuvchi tomonidan ishga tushirilganda, foydalanuvchi o'zini tizimga tanishtirib, maxfiy so'z (parol)ni kiritishi kerak. MS Access 2ta guruhni tashkil etadi: administratorlar (**Admins guruhi**) va od-diy foydalanuvchilar (**Users guruhi**). Boshqa guruhlar va foydalanuvchilarni aniqlashga imkoniyati ham bor.

**Admins guruhi** a'zolari BBning barcha obyektlari va hujjalardan bemalol foydalanish huquqiga egadir. Boshqa guruhlar va foydalanuvchilar uchun BBning faqat alohida obyektlardan foydalanish huquqi taqqid etilishi mumkin. Foydalanuvchilar **Users** guruhi uchun bir turdag'i foydalanish huquqi tarkibiga «Berilganlarni o'qish» va «Berilganlarni yangilash» jadvallar va so'rovlar uchun, shuningdek, shakllar va hisobotlar uchun «Ochish/ishga tushirish» xizmat turlarini kiritish mumkin.

### Berilganlar bazasini yaratish va tashkil etish

BBni yaratish jarayonini avvalda ta'riflab o'tgan edik. Talabalar o'quv loyihamiz obyektini axborot tizimining BB modeli misolida ko'rib chiqamiz.

MS Access tizimida BBni yaratish jarayoni quyidagi tarzda amalga oshiriladi. Tizim ishga tushirilganda, ish tartibi-rejimini tanlash uchun mu-loqot oynasi paydo bo'ladi. (32.1 rasm.), ushbu oynachada «Yangi BB» («Новая база данных») punktini tanlash lozim.



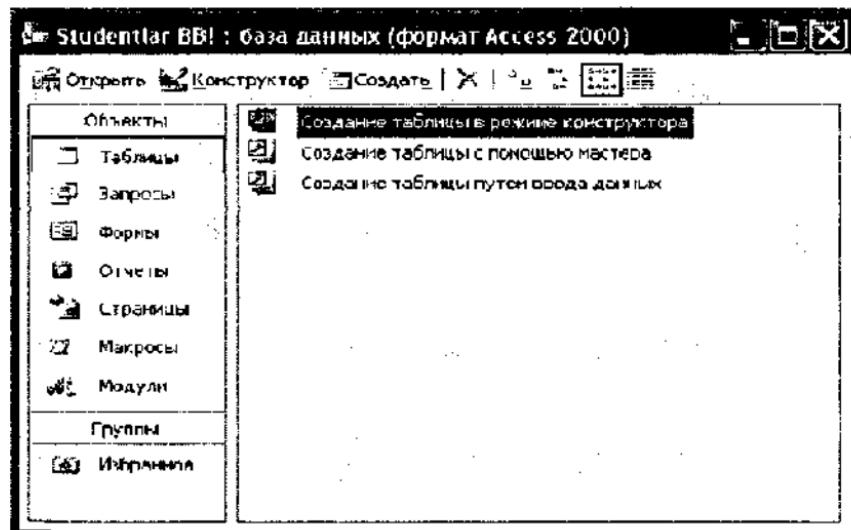
32.1-rasm. Faylini yaratish va tashkil etish.

Birinchi bo'lim tanlangandan keyin yangi bazani nomi va saqlanish yo'lli vazifasini ko'rsatadigan oyna paydo bo'ladi. BB faylini saqlash uchun

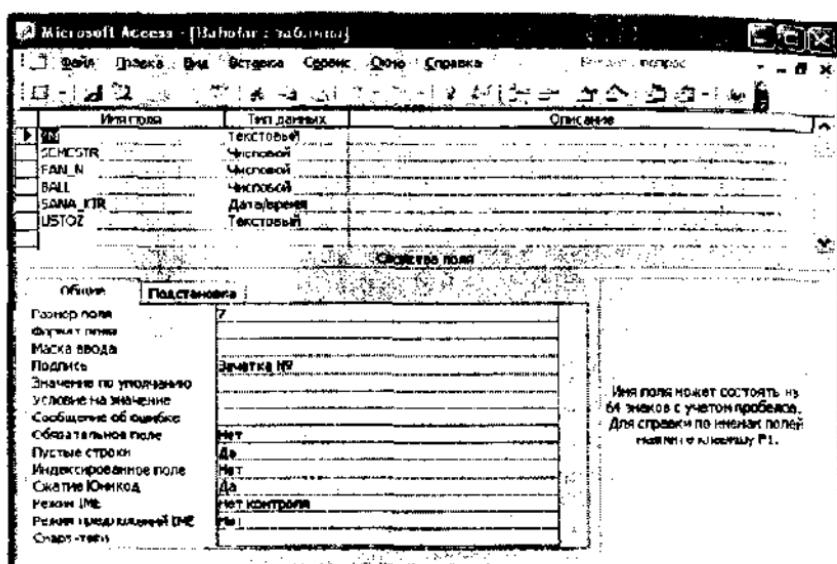
diskda papkani tanlab olamiz va bazani **Talabalar BB** deb nomlaymiz, Sozdat-Yaratish tugmasiga bosamiz, shundan keyin BB oynasi ochiladi.

Keyin oldin ta'riflangan konseptual modelga muvofiq jadvallar tuzilmasini tashkil etish zarur. Shuningdek, Case muhitlardan foydalangan holda BB jadvalini yaratish va MS Access ga import qilish mumkin, lekin quyida konstruktor yordamida jadvallar tuzish tartibi ta'riflangan. Mayjud tizimlarni yangilash va yangi jadvallar tuzish maqsadida har bir foydalanuvchi ushbu rejimni ishlata bilishi kerak.

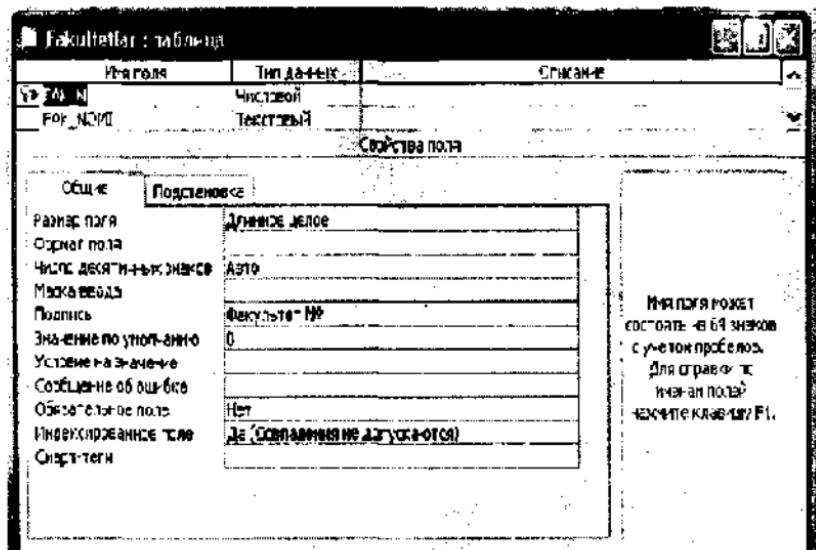
«Конструктор» rejimida **jadval tuzish** (16.2-rasm) bo'limini tanlaymiz va BBni bosh jadvali tuzilmasiga ta'rif beramiz, ya'ni har bir jadval maydonini **nomi**, turi, o'chishini belgilaymiz, shuningdek, **birlamechi kalit** (agar zarur bo'lsa), shartli belgililar bilan ifodalangan maydon va imzolarga ham nom beramiz (16.3 va 16.4-rasmlar). Maydon nomlarini lotin harflarda yozish maqsadga muvofiq, BB bilan ishlovchi programmalar va so'rovlarda foydalanish uchun qulay bo'lgan bir jumlali qisqa so'z; Maydonlar uchun imzo xizmati ekrani shakllar va hisobtlarni ishlab chiqish vazifasini yengillashtiradi.



32.2-rasm. BB oynasi.



32.3-rasm. «Konstruktur»da Baholar jadval tuzilmasining tavsifi.



32.4-rasm. «Konstruktur»da Fakultetlar jadval tuzilmasining tavsifi.

## Talabalar o'quv loyihasining tuzilishi

**Ro'yhat jadval tuzilmasi 32.3-jadvalda keltirilgan.**

32.3-jadval

### Ro'yhat jadvalining tuzilmasi

Maydon nomi	Berilganlar turi	Maydon o'chami	Shartli belgilar bilan ifodalangan (indeks) maydon	Imzo
ZN	Matnli	8	Ha (O'xshashliklarga yo'l qo'yilmaydi)	Reyting daftarchasi raqami
FAMILIYA	Matnli	15		Familiya ism
ISM	Matnli	15		Otasining ismi
O_ISM	Matnli	15		
K_CANA	Sana/vaqt	Sanani qisqa formati		Qabul qilingan sana
FAK_N	Raqamli	Uzun butun	Ha (O'xshashliklarga yo'l qo'yildi)	Fakultet nomeri
YON_N	Raqamli	Uzun butun	Ha (O'xshashliklarga yo'l qo'yildi)	Yo'nalishi
KURS	Raqamli	Bayt		Kurs
GURUH_N	Matnli	10		Guruhi
PASP_S	Matnli	2		Pasport seriyasi
PASP_N	Raqamli	Uzun butun		Pasport raqami

**ZN** maydoniga «Kalit maydon» xizmat turini kiritish kerak, chunki Reyting daftarchasi nomeri – har bir talaba uchun alohida ahamiyat kasb etadi va BB jadvaliga uni alohida tanishtiradi. Jadval tuzilmasini ta'riflashni yakunlagandan so'ng, **Сохранить-Saqlash** buyrug'ini (standart asboblar panelida, menuy – **Файл bo'limida yoki «Конструктор» oynasini yopish vaqtida**) beramiz va jadvalni - **Ro'yhat** deb nomlaymiz.

Qo'shimcha tarzda BBda fayl nomi **Fakultetlar** bo'lgan fakultetlar berilgan ma'lumotnomasini tashkil etamiz, uning tuzilmasi 32.4-jadvalda keltirilgan.

32.4-jadval

#### **Fakultetlar jadvalining tuzilmasi**

Maydon nomi	Berilganlar turi	Maydon o'chami	Shartli belgilar bilan ifodalangan (indeks) maydon	Imzo
FAK_N	Raqamli	Uzun butun	Ha (O'xshashliklarga yo'l qo'yilmaydi)	Fakultet raqami
FAK_NOMI	Matnli	50		Fakultet nomi

32.5-jadval

#### **Baholar jadvalining tuzilmasi**

Maydon nomi	Berilganlar turi	Maydon o'chami	Shartli belgilar bilan ifodalangan (indeks) maydon	Imzo
ZN	Matnli	8	Ha (O'xshashliklarga daftarchasi yo'l qo'yiladi)	Reyting raqami
SEMESTR	Raqamli	Bayt		Semestr
FAN_N	Raqamli	Butun sonlar	Ha (O'xshashliklarga yo'l qo'yiladi)	Fan/predmet
BALL	Raqamli	Butun		Baho
SANA_KIR	Sana/vaqt	Qisqa format		Sana
USTOZ	Matnli	40		ustoz

32.6-jadval

#### **Fanlar jadvalining tuzilmasi**

Maydon nomi	Berilganlar turi	Maydon o'chami	Shartli belgilar bilan ifodalangan (indeks) maydon	Imzo

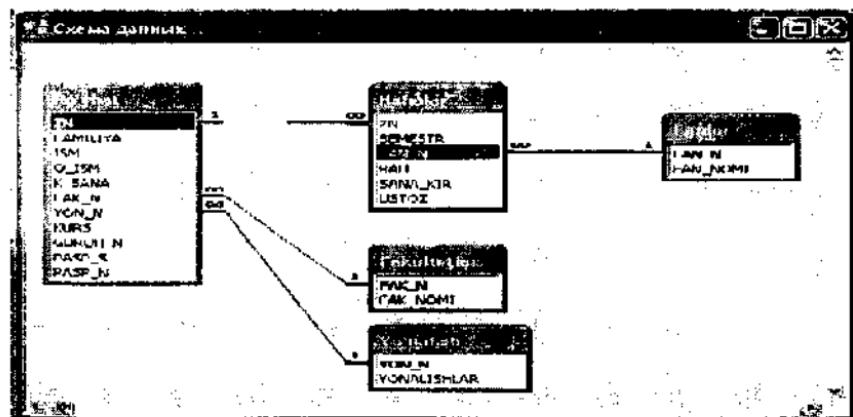
FAN_N	Raqamli	Butun sonlar	Ha (O'xshashliklarga raqamiga yo'l qo'yilmaydi)	Fan
FAN_NOMI	Matnli	50		Fan nomlari

### 32.7-jadval

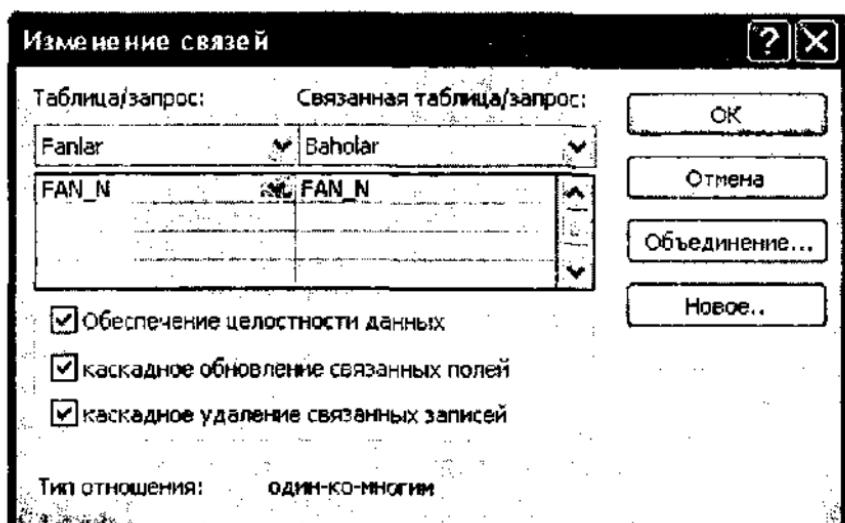
#### Yo'nalish jadvalining tuzilmasi

Maydon nomi	Berilganlar turi	Maydon o'lchami	Shartli belgilar bilan ifodalangan (indeks) maydon	Imzo
YON_N	Raqamli	Butun sonlar	Ha (O'xshashliklarga yo'l qo'yilmaydi)	Yo'nalish raqami
YONALISHLAR	Matnli	50		Yo'nalishlar

Keyin Схема данных (berilganlar chizmasi) oynasini ochib, bazada jadvallar o'rtaida birdan-ko'plikka aloqalarni keltiramiz (BB oynasi uchun kontekstli menyuda ushbu buyruq tanlanadi) hamda birlamchi kalit maydoni nomini boshqa jadvalning qo'shimcha maydoniga o'tkazamiz (32.5-rasmga qarang). Изменение связей/Aloqalarni o'zgartirish oynasida (32.6-rasmga qarang) berilganlar havola butunligi shartlariga rivoja qilinishi: bog'langan maydonlar pog'onali yangilash va bog'langan yozuvlarni pog'onali o'chirish xizmat turlarini keltiramiz.



32.5-rasm. BB chizmasi.



**32.6-rasm.** Berilganlarning butunlik shartlariga rioya qilish vazifalari.

### Xulosa

*Relatsion BBBT MS Access kichik va o'rta hajmdagi korxonalar uchun axborot tizimlari, mutaxassisning ishechi o'rinxarini yaratish uchun juda qulay tizimlardan hisoblanadi. MS Access uchun integratsiya masalalari yaxshi hal qilingan. Bunga biz 20- bobda ham guvoh bo'ldik. Hattoki, berilganlar bazasini modellashtirishga xizmat qiluvchi eng zamonaviy muhit Toad Data Modeler ham uning uchun DDL Script tarni yaratib beradi.*

*Aynan shu bobdan keyin, Toad Data Modeler muhitida qilingan ishlarni yuqorida bajargan vazifalar bilan taqqoslash va xulosa chiqarish Sizning zimmangizda. Agarda bu ishn'i hozir bajara olmasangiz, keyingi relatsion BBBT MS Access ga tegishli boblarni o'zlashtirgan keyin taqqoslashishing ham mumkin. Muhimi, albatta bu vazifani bajarish zarurligida.*

### Nazorat uchun savollar va amaliyot uchun topshiriqlar

1. MS Access 2003 BBB tizimidagi maksimal obyektlarning soni nechta va nima uchun?

2. Bazani himoya qilishdagi maksimal belgilari soni nechta?
3. Jadval nomidagi belgilari soni nechtdan oshmasligi kerak?
4. Maydonlar nomidagi belgilarning maksimal soni nechta?
5. Jadvaldagi maksimal maydonlar soni nechta?
6. MEMO maydonining hajmi qancha?
7. OLE obyekt maydonining hajmi qancha?
8. Indeksdagi maksimal maydonlar soni nechta?
9. Jadvalda nechta indeks bo'lishi mumkin?
10. Predikatdagi belgilari soni qancha bo'lishi mumkin?
11. Bitta jadvaldagi maksimal aloqalar soni qancha?
12. So'rovlarda bir yo'la nechta jadval ishtirop etishi mumkin?
13. So'rovlarda bir yo'la nechta maydonlar ishtirop etishi mumkin?
14. So'rovlardagi saralashdagi maksimal belgilari soni nechta?
15. So'rovlarning ichma-ich qo'yilishidagi darajalarning maksimal soni nechta?
16. WHERE yoki HAVING ifodalarida AND operatorlarining maksimal soni nechta?
17. Shabl va hisobotda yozuvdagi belgilarning maksimal soni nechta?
18. Shabl va hisobotlarning ichma-ich qo'yilishidagi darajalarning maksimal soni nechta?
19. Makrosdagagi makrobuyruqlarning maksimal soni nechta?
20. MS Access 2003 BBB tizimidagi berilganlarning turlarini aytib bering?
21. MEMO va OLE maydonlarining turlari qanday?
22. O'rinn almashtirish Mactepi (ustasi)dagagi maydon o'lchami qanday bo'ladi?
23. BBni himoyasi uchun qanday ishlataladi?
24. BBni himoyasi uchun qanday guruhlar ishtirop etadi?
25. 31.3–31.7-jadvallardagi tuzilishlar orqali berilganlar **bazasining** jadvallarini yaratting.
26. Har bir jadval uchun kamida 4 – 5 satrhami to'ldiring.
27. Jadvallar orasida aloqalar o'rnating.
28. Jadvallar orasidagi aloqalarni izohlang.
29. Jadvallarga berilganlar ustida – olib tashlash, o'zgartirish amallarini yangi va berilganlarni kiritish ishlarini bajaring va tizimdagi o'zgarishlarni izohlang.

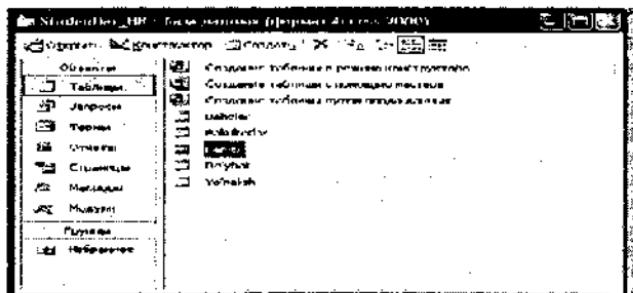
## **33-bo'lib. JADVALLAR BILAN ISHLASHNING STANDART REJIMI**

### **Jadvallar va ular bilan ishlash vositalari**

**Tayanch iboralar:** o'rin almashirishlar ustasi, qo'yish, satrlar manbasi turi, ro'yxatli maydon, suralash, ekrani shakl, avtoshakl, konstruktor, shakl, shakl maketi, xossa, tekislash, o'lcham, rang, rasmiylashtirish, shartli formattash, maket, tobe shakl, ro'yxatli maydon, chizim, tugma, chiziq, to'g'riburchak, obyektni erkin katagi, ilovalar to'plami, MS Visual Basic protsedurasini, Function protsedurasi, Sub protsedurasi, sinf moduli, shakl moduli, hisobotlar moduli, yangi sinf modulini ochish, mavjud sinf modulini ochish.

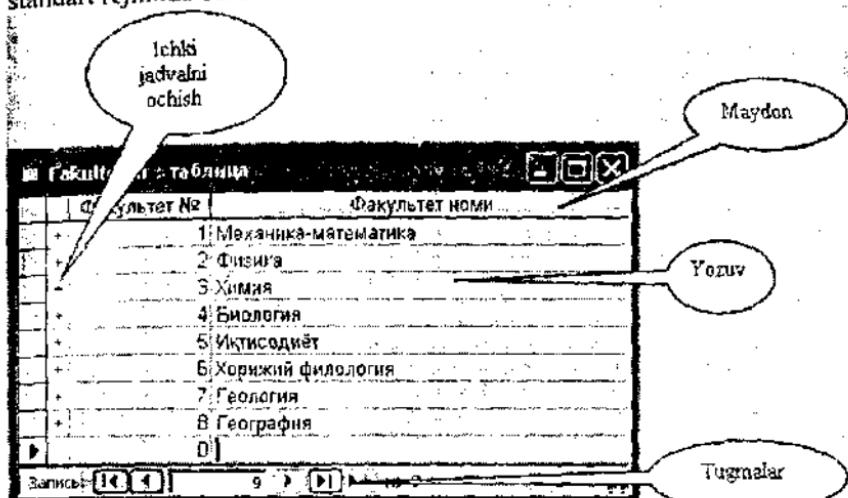
**BBni axborotlar bilan to'ldirishni berilgan ma'lumotnomalardan boshlash** maqsadga muvofiqdir – aks holda bosh jadvallarni to'ldirganda BB havola butunligini saqlash borasida nomuvofiqliklar yuzaga kelishi mumkin. Masalan, agar fakultetlar berilgan ma'lumotnomasida ko'rsatilgan har bir talaba uchun fakultet raqami bo'lmasa, oynada «**Kiritilgan ifoda mazkur maydon uchun to'g'ri kelmaydi**» degan xabar paydo bo'ladi va siz to'g'ri ifodani ketirmaguningizcha, berilganlarni saqlay olmaysiz.

BB jadvali axborotlari bilan ishlash uchun (qo'shish, tahrirlash va yozuvlami o'chirish) uni BBni «**Jadvallar**» (Tаблицы) obyekt bo'limida tanlash lozim (33.1-rasmiga qarang) va sichqoncha tugmasini ikki marta bosib ochish kerak.



**33.1-kasm. Berilganlar bazasi**

Jadval, 33.2.-rasmida ko'rsatilganidek, axborotlar bilan ishlashdagi standart rejimida ochiladi.



33.2-rasm. BB jadvali bilan ishlashni standart rejimi.

Ushbu rejimda BB jadvalini har bir yozuvi satr ko'rinishida keltiriladi, bunda satrlar maydonlar imzolari yoki ular bo'lganda, maydonlar nomi keltirilgan maydon ustunchalaridan iborat bo'ladi. Jadvalni quyi qismida \* belgili bo'sh yozuv chap ustunda mavjud bo'ladi – bu jadvaldag'i barcha yozuvlami umumiyy umidori haqida axborot va yozuvlardir. Oyna quyi qismida jadvallararo o'tish tugmalari, jadvaldag'i joriy yozuvlar va yozuvlar bo'yicha umumiyy berilganlar o'rinn oladi. Agarda jadval yozuvlarini boshqa ko'pgina yozuvlar bilan bog'liqligi bo'lsa, mazkur holatda + belgili chap ustuncha mavjud bo'ladi. Bunday paytda sichqoncha bilan + ga bosilganda, kichkina ostki jadval sifatida - kerakli fakultetning talabalari ro'yxati ochiladi. Ostki jadvalni sozlash bosh menyuni **Вставка-Qo'yish (Kiritish)** bo'limida amalga oshiriladi.

MS Access tizimiida, shuningdek, ochiladigan ro'yxatlar ichidan berilganlarni tanlab olish va ko'rsatish uchun jadvallar rejimi «Подстановка – О'rin almashtirish»da qo'shimcha foydalaniш imkoniyatlari mavjud («Satrlar manbasi tur» – jadval yoki so'rov, ifodalar yoki maydonlar ro'yxati). Ushbu rejimni **Мастер хизмат** turi yordamida tashkil etish mumkin bunda «Мастер подстановок - О'рин алмасхтислар устаси» maydon turi tanlanadi yoki «Подстановка - О'рин алмасхтис» sahifasini o'ziga ta'rif beriladi. Misolda, ushbu rejim, **Ro'yhat** jadvalini **FAK\_N** maydoni uchun keltiriladi (33.3-rasmga qarang). Kelgusida ekran shakldagi

«Подстановка» bo'limini ta'riflangan jihatlari hisoblangan maydonlar uchun avtomatik tarzda «Ro'yxatli maydon» shaklidagi obyekt yaratiladi.

**Ro'yhat: таблица**

Имя поля	Тип данных	Описание
ФИО	Текстовый	
FAMILIA	Текстовый	
ISM	Текстовый	
O_ISM	Текстовый	
K_SAMA	Дата/Время	
FAK_N	Числовой	
YON_N	Числовой	
KURS	Числовой	
GURUH_N	Текстовый	
PASP_S	Текстовый	
PASP_N	Числовой	

**Свойства поля**

Общие	Подстановка
Размер поля	3
Формат поля	
Макс. ввода	
Должность	Зачетка №
Значение по умолчанию	
Условия на значение	
Сообщение об ошибке	
Обязательное поле	Нет
Пустые строки	Да
Индексированное поле	Да (Совпадения не допускаются)
Скрытие Юникод	Да
РежимIME	Нет контроля
Режим предложенийIME	Нет
Спарт-теги	

Имя поля может состоять из 6 знаков с учетом пробелов.  
Для справки по клавишам, нажмите клавишу F1.

33.3-rasm. «Конструктор»da «Подстановка» sahifalar parametrlari.

FAK\_N maydon tuzish uchun ro'yxatli maydondan foydalanish yordamida Ro'yhxat jadvalini ko'rinishi 33.4 rasmda keltirilgan.

**Ro'yhat: табл.123**

Запись №	Фамилия	Имя	Составной №/Марка/год	Бренд	Продукт	Код	Группа	Параметры	Заданы №
1	Борис	Борис	31.10.1990 Узбекистан	AMAT		1 A1	103		1236750
2	Саломон	Халид	13.02.2011 Узбекистан	Бозсат		1 Б12	104		1111110
3				Марка и модель		0			0

Бюро  
Чинчи  
Бекасий  
Геология  
Гидрогеология  
Гидрология

33.4-rasm. «Подстановка» tejimidagi jadval.

Jadval bilan ishlash jarayonida har bir maydon uchun bittadan yozuvlarni saralab olish va tayinlangan shartga muvofiq berilganlarni ko'rsatish uchun filtr (saralash) qilish mungkin (menyu **Yozuvlar bo'limiga** qarang).

Berilganlarga qo'shimchalar kiritish, tahrir va o'chirish ishlarnini amalga oshirib bo'lgandan keyin, **Coxpanit-Saqlash** buyrug'ini berish kerak yoki jadval oynasini berkitganda paydo bo'ladigan oynada, «saqlash» yoki «o'zgarishlar yo'q» buyrug'ini tanlash kerak.

### **BB bilan ishlash uchun ekrani shakllarni yaratish**

Ekrani shakllar relatsion BB bilan bog'langan ko'p miqdordagi jadvallardan iborat bo'lgan BB bilan qulay va ko'rgazmali tarzda ishlashni tashkil etishga imkon yaratadi. Shakllarni uch xil tarzda yaratish mumkin:

- jadval yoki so'rov asosida avtoforma (avtoshakl) yordamida;
- bitta yoki bir nechta jadval yoki so'rov asosida master (usta) yordamida;
- konstruktor yordamida.

*Avtoshakl yordamida* so'rov yoki asos jadvalning hamma maydonlari va yozuvlarini chiqaruvchi shakllarni yaratish mumkin. Asos manba shakl, hisobot yoki sahifalar orqali berilganlarga kirishni ta'minlaydi. MS Access ning berilganlar bazasida yozuvlar manbasi sifatida jadval, so'rov yoki SQL yo'riqnomalari kelishi mumkin. MS Access loyihasida esa yozuvlar manbasi sifatida jadval, tasavvur, SQL yo'riqnomalari yoki saqlangan protse-dura kelishi mumkin. Agarda tanlangan yozuvlar manbasi bog'langan jadvallar yoki so'rovlardan iborat bo'lsa, u holda shaklda ham ana shu manbaning hamma maydon va yozuvlari qatnashadi.

*Macmep (usta) yordamida* bitta yoki bir nechta jadval yoki so'rov asosida shakl yaratilishida, usta yozuv manbalari, maydonlar, maketlar va talab qilinadigan formatlar xususida batafsil savollar beradi hamda olingan javoblarga nisbatan *shaklni yaratadi*.

*Konstruktor yordamida* shakl yaratishda, avval asos shakl yaratiladi va keyin u talabga asosan konstruktor uslubida o'zgartiriladi. Konstruktor uslubida berilganlar bazasining jadvallari, so'rovlari, shakllari, hisobotlari, makroslar va sahifa obyektlarining makettari akslanadi. Foydalanuvchi ushu uslubda berilganlar bazasining yangi obyektlarini yaratishi yoki mavjudlarini o'zgartirishi mumkin.

Ekrani shakllarida bitta ekranida bosh va tobe jadvallar bilan ishlashni tashkil etish mumkin, ochiladigan ro'yxatlardan foydalangan holda berilgan ma'lumotnomasi jadvallardan berilganlarni tanlash ishlarni amalga oshirish, axborotni izlash va saralash rejimlaridan foydalanish, zarur hisobottarni

printerda chiqarish va boshqa amallarni ham amalga oshirish imkoniyati mavjud.

BB oynasida ekrani shaklni tuzish uchun **Shakllar** (Формы) obyekti shakl-rejim sahisasida – **Mastep yordamida shakllar tuzish rejimini tanlaymiz**. Keyin Mastepning bir qator savollariga javob berish kerak bo'ladi:

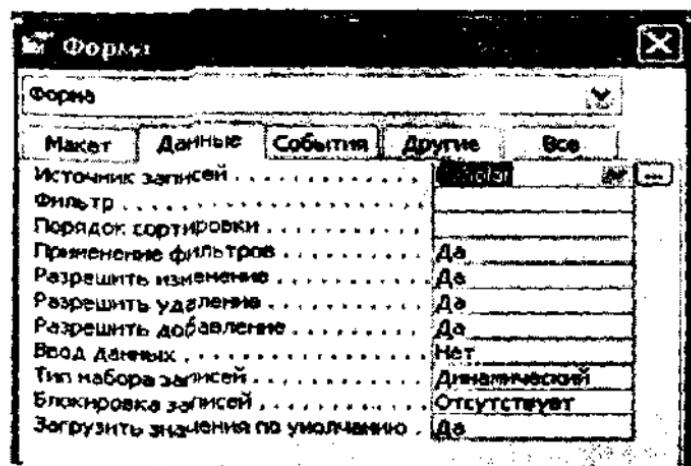
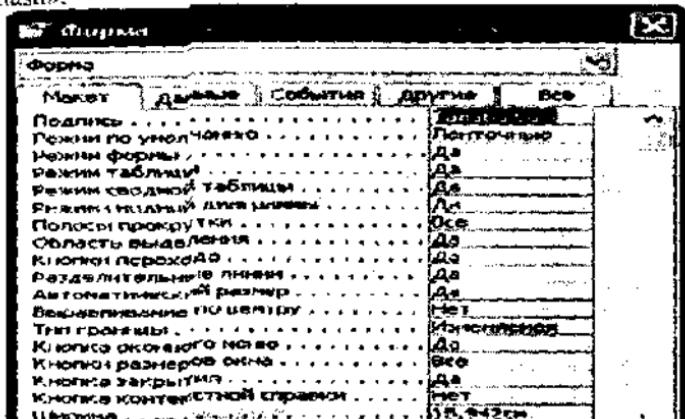
- shakllar uchun maydon tanlash – **Ro'yhat jadvalining barcha maydonlari va Baholar** jadvalining barcha maydonlarini tanlaymiz (oxirgi jadval uchun ZN maydonini ro'yxatni oxiriga joylashtiramiz; uni qo'shishimizga sabab, biz faqat bitta talabaning bahosini ekranda ko'rib ishonch hosil qilishimiz uchun kerak);
- berilganlarni namoyish etish turini tanlang – **tobe shakllarni**, ya'ni bitta shakldagi asosiy jadval va unga bog'liq shakldagi berilganlarni joylashuv tarzini tanlaymiz;
- tobe shakllarni tashqi ko'rinishini tanlang – **lentali turini** tanlaymiz;
- talab etilgan uslubni tanlang - **standart uslubni** tanlaymiz;
- shakllarga nom bering – bosh shaklga **Talaba** nomini, tobe shaklga **Talaba baho** nomini bering va aynan shu ekranni pastrog'ida **Shakl maketini o'zgartirish** punktini tanlaymiz, shundan so'ng **Gotovo** - Tayyor tugmasini bosamiz.

Olingan ekrani shakl shakllar konstruktorda ochiladi, BBni **Shakllar** sahisasida 2 ta yangi – **Talaba** va **Talaba baho** degan shakllar paydo bo'ladi. Tobe shaklni konstruktorda yanada yaxshiroq ko'rinishi uchun uning oynasini yopib, keyin konstruktorda yana **Talaba** shaklini ochish yaxshi natija beradi. Mazkur holatda u 33.5-rasmida keltirilgan shaklga ega bo'ladi.

33.5-rasm. Mastep yordamida tuzilgan ekrani shakl.

Ushbu shaklda, barcha yozuvlar to'liq ko'rinnmayapti, maydonlar joylasuvi holatini ham yaxshilasa bo'ladi.

Shaklar va uning barcha elementlari uchun **Свойства - Xossalalar** oy-nasimi o'chish mumkin, buning uchun sichqonchaning o'ng tugnachasi har qanday obyekt ustiga olib borilib chertiladi va kontekstli menyuda **Свойства - Xossalalar** so'zi tanlanadi (33.6-rasm). Bundan tashqari, kontekst menyusida obyektlar bilan ishlashni bir necha muhim imkoniyatlari mavjud, jumladan, «Выровнять-Текистлар» – obyektlar guruhini tenglash-tirish uchun foydali, «О'лчами», «Rang», «Rasmiylashtirish», «Shartli formatlash».



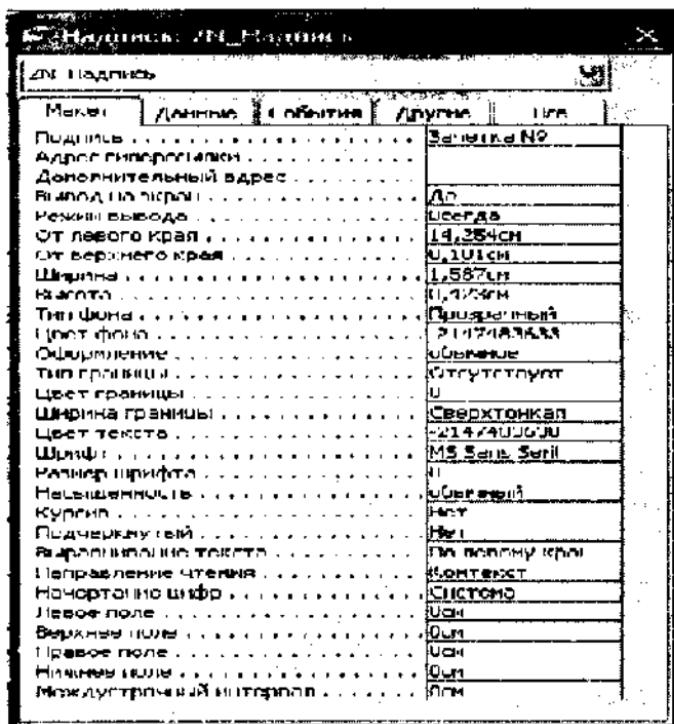
33.6-rasm. Shakl obyekti uchun xususiyatlar oynasi, maket va berilganlar sahifalari.

Hamma xususiyatlar oynada guruhlarga bo'lingan:

- **Макет** – joylashuv, shrift, rang va hokazo, obyektni tashqi ko'rinishi bilan bog'liq jihatlar;
- **Berilganlar** – ushbu bo'limda muhim jihatlar – **Berilganlar** yoki **Yozuvlar manbasi** bo'lib, obyektlarning biron bir berilganlarini tahrirlashga xizmat qiladi;
- **Voqeа-hodisalar** – aniq bir hodisalar (**Yuklash**, **Ochish**, **Yangilashdan oldin**, **Yangilashdan keyin** va h.k.) yuz berishida obyekt uchun qo'llaniladigan usullar, ya'ni protseduralar (programmalar);
- **Boshqalar** – boshqa xususiyatlar.

Ekranli shaklda quyidagi turdag'i elementlar(obyektlar) mavjud:

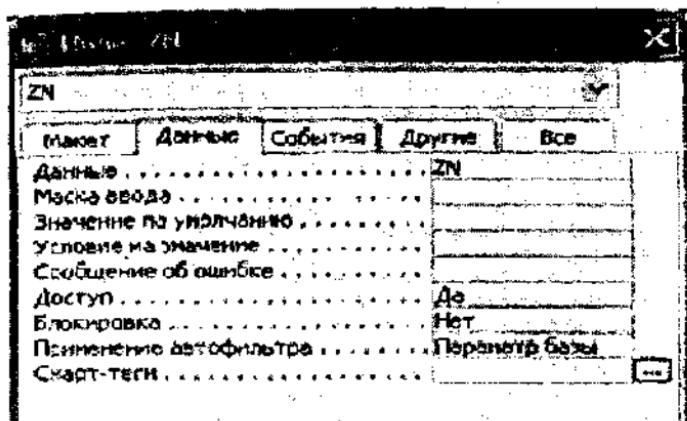
**Надпись** - **Yozuv** – odatda o'zgarmaydigan shakl ichidagi matn. Ushbu obyektni asosiy mohiyatlari xususiyatlar oynasini **Макет** sahifasida joylashigan (33.7-rasm).



33.7-rasm. Obyektni xususiyatlar oynasini **Yozuv** turi.

**Maydon** – tahrirlash maydoni, berilganlar bazasini maydoni bilan bog'liq yoki o'zgaruvchan bo'ladi. Ushbu obyektning asosiy jihat –

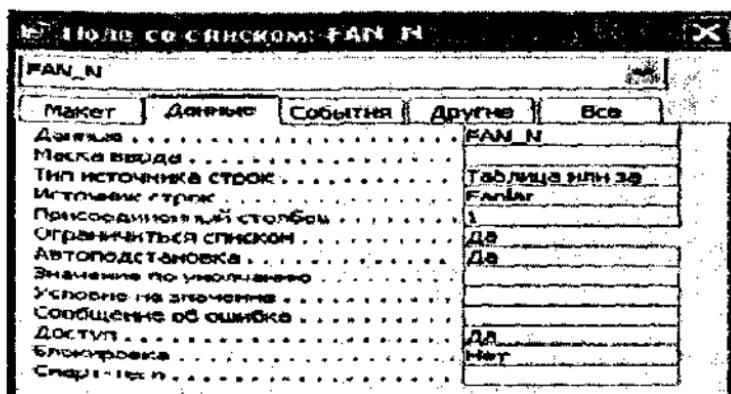
xususiyatlar oynasini **Данные-Berilganlar sahifasidagi Berilganlar** satr hisoblanadi (33.8-rasm).



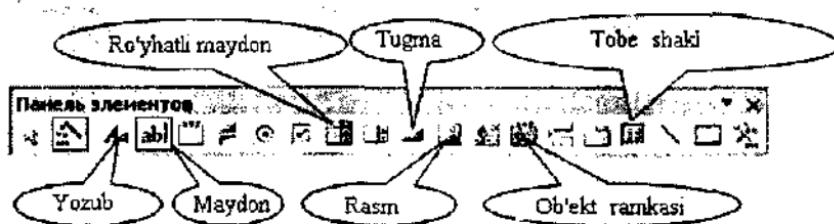
33.8-rasm. Obyektni xususiyatlar oynasini Maydon turi.

**Tobe shakl** – bosh jadval bilan bog'liq bo'lgan, berilganlar jadvali uchun ilovali shaki hisoblanadi, unda asosiy shaklda mavjud bo'lgan elementlar ham joylashishi mumkin.

**Ro'yxatli maydon** – murakkab element bo'lib, berilgan ma'lumot-noma jadvallari, ro'yxatlar va massivlar berilganlar ini ko'rsatish imkonini beradi va tanlangan ifodalarni boshqa jadvalning maydoniga olib o'tish vazifasini bajaradi (panelni 9- elementi, xususiyatlar oynasi, 33.9-rasm qarang).



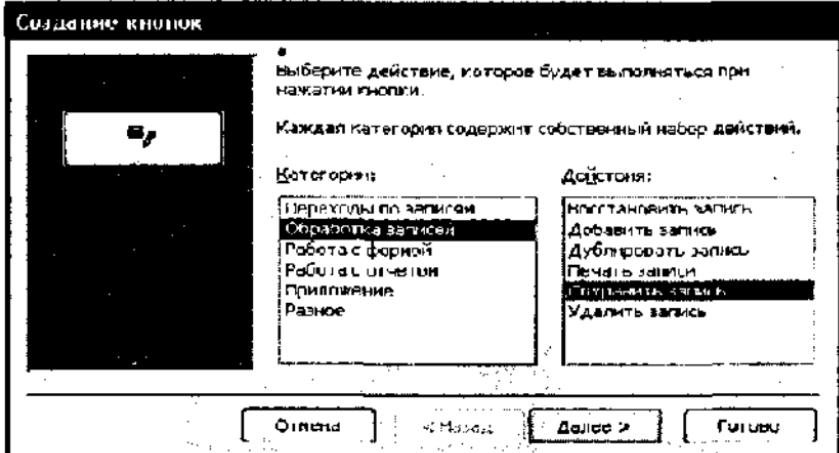
33.9-rasm. Obyektni xususiyatlar oynasidagi «Ro'yxatli maydon» turi.



33.10-rasm. Ekrani shakllarni elementlar chizimi.

Bundan tashqari, shaklga elementlar paneli tugmasidan foydalangan holda boshqa obyektlarni qo'shish mumkin (33.10-rasm). Obyekt turlarini ro'yxati quyida keltirilgan.

**Biron bir isodani o'zgartirgichlar guruhi, o'chirish tugmalari.** Tugmalar – aniq bir vazifalarni bajarish bilan bog'liq bo'lgan tugmalar va turli xildagi tugmalar guruhi (Chizimdag'i 5-7 va 11 tugmalar). Tugma obyekti tanlanganda, **Мастер** ishga tushadi, ushbu Мастер sizga jadvaldagi yozuvlarga o'tishni standart tugmalari to'plamini, yozuvlarni qayta ishlash (tiklash, ko'paytirish, nashr qilish va h.k.), shakl va hisobotlar bilan ishlash(nashr qilish, ko'rib chiqish, faylga yuborish yoki pochtadan jo'natish) ilovalar bilan ishlashni va boshqa har xil xizmat turlarini taklif etadi. Tugmalarni **Мастер** yordamida tuzish mumkin, bunda uning faoliyatini bilan bog'liq bo'lgan programmalni kod allaqachon mavjud bo'лади. (33.11-rasm).



33.11-rasm. Tugmalarni yaratish Мастері оynasi.

**Bayroqcha** – odatda mantiqiy turdag'i jadval maydoni bilan bog'liq bo'lgan maydon, bunda odatda belgi turadi, ba'zida turmaydi (**Chizimning 6-elementi**).

**Ro'yxat** – ifodalar ichidan bittasini tanlash uchun berilganlar ro'yxati (10).

Rasm – shaklga rasmni qo'yish (12).

Obyektni erkin katagi – Windows ilovalarini har qanday obyekti, bunga kerakli ilovalarni chaqirish mumkin bo'lgan ilovalarni tahrirlash kiradi (13).

Obyektni qo'shilgan ramkasi/hoshiyasi – «OLE obyekti maydoni» ko'rinishidagi jadval maydonlari bilan ishlash uchun (14).

Ilovalar to'plami – ko'p sahifali shakl (16).

Chiziq, Te'g'riburchak – rasmiylashtirish elementlari (18, 19).

Ekrani shakldan foydalangan uchun uni ishga tushirish kerak, buning uchun konstruktur oynasini yopamiz va **Talabalar** shaklida **Ochish** buyrug'ini tanlaymiz. Shaklni BB bilan ishlash jarayonidagi ko'rinishi 33.12-rasmida keltirilgan.

Lokotchi Taro	Seriya	Seriya Ustrova
1	12/05/2010/12/05/2010	
0	12/05/2010/12/05/2010	
0	12/05/2010/12/05/2010	

33.12-rasm. Yekranli shakllardan foydalangan holda BB bilan ishlash

Mazkur rejimda mayjud yozuvlarni tahrirlash talabalar ro'yxatiga yan-gilarini qo'shimcha qilish va har bir talaba uchun baholar ro'yxatini tuzish mumkin. Yozuvlarni o'chirish uchun **Mactep** yordamida asosiy shaktiga yozuvlarni qayta ishlash kategoriysi tugmasini qo'shish mumkin, bunda **Yozuvni o'chirish** xizmat turidan foydalilanadi.

Shakl bilan ishlashda maydonlarning biridan yozuvlarni tanlab olish va faqat tayinlangan shartga muvofiq keladigan berilganlarni ko'rsatish uchun

filtr (saralash) qilish mumkin (Access tizimini menyusidagi Yozuvlar bo'limiga qarang).

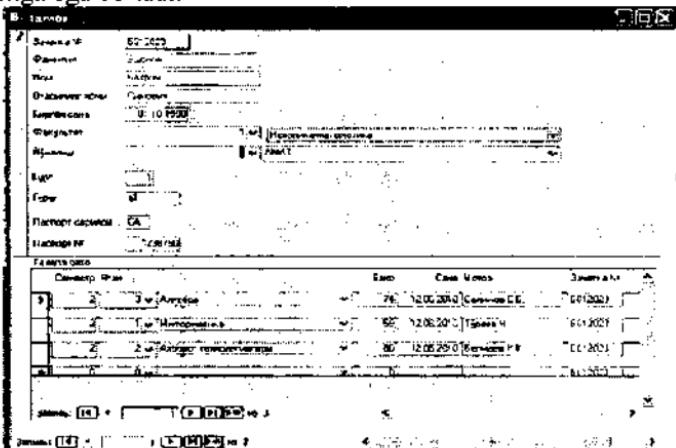
Yuqorida aytib o'tilganidek, mazkur shakl uchun BB bilan ishlash qu'layliklarini oshiruvchi ahamiyatli muhim takomillashuvlarni amalga oshirish mumkin.

Dastavval, **Констукторда** maydonlar joylashuvini o'zgartiramiz va yozuvlarni kengaytiramiz.

Keyinchalik berilgan ma'lumotnoma jadvallaridan mutaxassislik (bosj jadvalga kerakli raqamni kiritish bilan) va fan predmetini (uning raqamini berilgan nomadan olib baholar jadvaliga kiritish bilan) tanlab olish uchun obyektlar shakllariga ro'yxatli maydonni qo'shamiz. Ushbu asosiy xususiyatlar **Мастер** yordamida ta'riflanadi, ushbu **Мастер** shaklga obyektni ushbu turini qo'shish bilan avtomatik tarzda ishga tushadi.

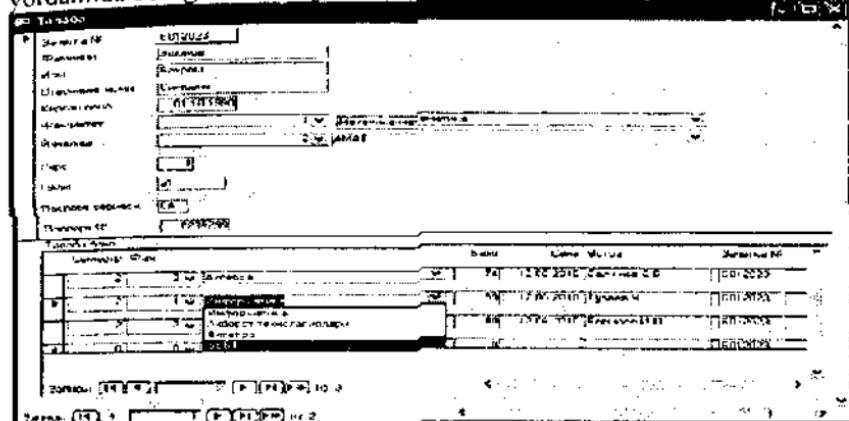
**Мастерни** birinchi bosqichida **Ro'yxatli maydon obyekti jadval va so'rovlardan olingan ifodalarni ishlataladi** degan bo'limni tanlaymiz, ikkinchi bosqichda – ro'yxatdan bizga kerakli bo'lgan berilgan ma'lumotnoma jadvalini tanlaymiz, uchinchisida – barcha maydonlarni (raqam va nom) tanlaymiz, to'rtinchi bosqichda – **kalitli ustunni yashirish bayroqchaga** belgi qo'yamiz va uni nomlash uchun maydon kengligini belgilaymiz, beshinchisida – **Сохранить в поле/Maydonda saqlash** shartini keltiramaniz va berilgan ma'lumotnoma jadvalini kalitli maydon ifodasi saqlanadigan asosiy jadval ro'yxatidan maydonni tanlaymiz. Keyin **Tayyor, Yozuv** xizmat turiga sichqoncha bilan chertamiz, ro'yxatli maydonni o'chirish mumkin.

Shundan keyin, konstruktorda ushbu shakl 33.13-rasmda keltirilgan ko'rinishga ega bo'ladi.



33.13-rasm. Shakllar konstruktorda takomillashgan ekranli shakl.

Takomillashgan shaklda BB bilan ishlash rejimida fakultetlar, mutaxassistiklar, fan nomlarini ko'tish mumkin va ularni ochiladigan ro'yxatlar yordamida berilgan nomlar jadvalidan tanlab olish mumkin. (33.14-rasm).



33.14-rasm. Ro'yxatli maydon obyektlari ishtirokida ekrani shakklardan foydalangan holda. BB bilan ishlash.

### MS Visual Basic protsedurasini yaratish

Berilganlar bazasida MS Visual Basic protsedurasidan foydalangan holda, jadvallar, shakllar, hisobotlar va so'rovlarning o'zaro harakatini boshqarish mumkin. MS Visual Basic yuqori darajadagi Basic programmalash tilining namoyishli naqli bo'lib, Microsoft korporatsii mutaxassislari tomonidan Windows ilovalarni yaratish uchun ishlab chiqilgan. Protseedralarning bir nechta turi mavjud:

- hodisalarga ishlov beruvchi;
- standart modullardagi shaxsiy **Function** va **Sub** protseduralari;
- sinf modullaridagi **Function** va **Sub** protseduralari.

*Hodisalarga ishlov beruvchi protseduralarning kodini* shakl yoki hisobotning xossalariiga joylab qo'yish mumkin.

**Function** protsedurasi qiymatni qaytaradi, undan ifodada foydalanish mumkin. Bu protseduraning tavsisi **Function** yo'riqnomasidan boshlanadi va **End Function** bilan tugallanadi.

**Sub** protsedurasi amalni bajaruvchi protsedura bo'lib, u natijani qaytarmaydi. Uning tavsisi **Sub** yo'riqnomasidan boshlanadi va **End Sub** bilan tugallanadi. Bularning ikkalasi shu bazadagi barcha protseduralar kirishi mumkin bo'lgan **standart modullarga yoki sinf modullariga** joylashtiriladi.

*Sinf moduli* yangi obyektning tavsifini saqlaydi. Shu modulda aniq-langan protseduralar obyektning xossalari va usullariga aylanadi. Bu protseduralar shakllar moduli va hisobotlar modulini saqlaydi. *Shakl moduli* aniq shakl yoki uning elementlarida yuz beradigan hodisalarga ishllov beruvchi hamma protseduralarning programmasidan iborat bo'ladi. *Hisobotlar moduli* aniq hisobot yoki uning boshqaruv elementlarida yuz beradigan hodisalarga ishllov beruvchi hamma protseduralarning kodidan iborat bo'ladi. Modul bu – tavsiflar, yo'riqnomalar va protseduralarning yagona nom bilan atalgan majmuasidir.

### Foydalanuvchi funksiyalarini yaratish

Modulni ochish uchun quyidagi ikkita harakatdan birini amalga oshiring:

A) *yangi standart modulni ochish* uchun BB oynasidan «**Modul**»ni belgilab, BB asboblar panelidan **Sozdat** tugmasini bosamiz.

*Mavjud standart modulni ochish* uchun BB oynasidan «**Modul**»ni belgilab, BB asboblar panelidan **Констуктор** tugmasini bosamiz.

Shakl moduli yoki hisobot modulini ochish uchun, shakl yoki hisobotni **Констуктор** bilan ochamiz va asboblar chizimidan «**Программа**»ni tanlaymiz.

B) Shakl yoki hisobotga bog'liq bo'limgan, *yangi sinf modulini ochish* uchun **Insert** menyusidan **Class Module** buyrug'ini tanlaymiz.

*Mavjud sinf modulini ochish* uchun BB oynasidan «**Modul**»ni belgilab, BB asboblar panelidan **Констуктор** tugmasini bosamiz.

**Function** yo'riqnomasini kiritib, funksiyani e'lon qilamiz.

Funksiya nomi va qavsda funksiyaning hamma argumentlarini kiritamiz. Masalan, **IsLoaded** funksiyasining yo'riqnomasi **strFormName** satrli argumentni bildiradi.

**Function IsLoaded (strFormName As String) As Boolean**

Visual Basic tilida funksiya bajarishi kerak bo'lgan amal yoki hisoblash programmasini yozamiz.

### Foydalanuvchining Sub protsedurasini yaratish

Shakl moduli yoki hisobot modulini ochish uchun shakl yoki hisobotni **Констуктор** bilan ochamiz va asboblar chizimidan **Programmani** tanlaymiz.

Shakl yoki hisobotga bog'liq bo'limgan, *yangi sinf modulini ochish* uchun **Insert** menyusidan **Class Module** buyrug'ini tanlaymiz.

*Mavjud sifz modulini ochish* uchun BB oynasidan «Modul»ni belgilab, BB asboblar panelidan Конструктор tugmasini bosamiz.

**Sub yo'riqnomasini kiritib, protsedurani e'lon qilamiz.**

Protsedura nomi va qavsda funksiyaning hamma argumentlarini kiritamiz. Masalan, **ShowEvent nomli Sub** protsedurasi *EventName* satrli argumentni bildiradi.

**Sub ShowEvent(EventName As String)**

Visual Basic tilida protsedura bajarishi kerak bo'lgan amallar programmasini yozamiz.

### **Hodisalarga ishlov beruvchi protsedurani yaratish**

Yuz bergan hodisaga javob beruvchi programmani ishga tushirish uchun, hodisaning xossasini shakl, hisobot yoki boshqaruv elementining qiymati (hodisalarni ishlov protsedurasi)da ko'rsatish kerak. Bu vaziyatda hodisalarni ishlov protsedurasining andazasi yaratiladi, ya'ni foydalanuvchi harakatlari, programmaning bajarilishi yoki tizim generatsiyasi natijasida sodir bo'ladigan hodisalarga avtomatik tarzda javob beruvchi protsedura. Keyin aniq bir hodisa yuz berganda, unga javob beruvchi programmani qo'shish mumkin. Hodisaning xossasi boshqarish elementining nomli atributi, shakllar, hisobollar, sahisalar yoki bo'limlardagi yuz bergan hodisaga javob qaytarish jarayonida amalga oshiriladi. Boshqaruv elementi bu – foydalanuvchining grafik aloqadagi obyekti bo'lib, maydon, bayroqcha, aylantirish chizimi yoki **tugmadan iborat**.

Shaklni Конструктор uslubida ochaniz.

Ekranga shakl, hisobot, bo'sim yoki boshqarish elementlarini shakl yoki hisobot tarzida chiqaring.

**События** (Hodisa) ilovasiga kiring.

**Protsedura** javob qaytarishi kerak bo'lgan hodisaning xossasini tanlang. Masalan, O'zgarish (Change) hodisasiga ishlov beruvchi protsedurani ochish uchun, O'zgarish (OnChange) xossasini tanlang.

**Построитель** tugmasini bosing va **Построитель** muloqot oynasiga o'ting.

Shakl modulida Sub va End Sub protsedurasiga ishlov beruvchi hodisani akslantirish uchun ikki marta «Программы» elementini bosing. Bu yo'riqnomalar hodisalarga ishlov protsedurasini e'lon qiladi yoki aniqlaydi. Microsoft Access shakl yoki hisobot modulida avtomatik tarzda har bir obyekt yoki hodisaga ishlov berish protsedurasini Private kalitli so'zdan foydalangan holda tavsiflaydi hamda shu moduldag'i boshqa protseduralaridan ana protseduraga kirish mumkinligini bildiradi.

Hodisa yuz berganda javob beruvchi programmani protseduraga kiritamiz. Masalan, ZN maydonidagi berilganlar o'zgartirilganda, signal berish uchun CompanyName\_Change hodisasiga ishlov protsedurasiga Beep yo'rqnomasini kiritamiz:

**Private Sub ZN\_Change()**

**Beep**

**End Sub**

Har gal O'zgarish (Change) hodisasi yuz berganda, hodisaga ishlov berish protsedurasi ishlaydi.

### **Shakl yoki hisobotga bog'liq bo'limgan sinf modulini yaratish**

BB oynasi yoki MS Visual Basic taxrirkhisida **Insert** menyusidan **Class Module** buyrug'ini tanlaymiz. MS Visual Basic tahrirkhisida bo'sh sinf moduli ochiladi.

Modulga zaruriy tavsiflar va protseduralarni qo'shamiz.

Asboblar chizmasida Сохранить тугмасини bosamiz va Сохранить как muloqot oynasida modulning nomini ko'rsatamiz.

### **Xulosa**

*MS Access BBB tiziminig jadvallar va ular bilan ishlash jadvalning xossasi va undan foydalangan holda, jadvalga berilganlari kiritish, foydalaniш va o'zgartirish imkoniyatlarini ko'rib chiqdik.*

*BB bilan ishlash uchun ekranli shakllarni uch xil tarzda yaratish mumkin – jadval yoki so'rov asosida avtoforma (avtoshakl) yordamida; bitta yoki bir nechta jadval yoki so'rov asosida master (usta) yordamida; konstruktor yordamida. Bularning har biri bilan ishlash imkoniyatiga ega bo'ldik. Ekrani shakllarni yaratish uchun ishlatiladigan elementlar chizimi bilan tanishdik.*

*MS Visual Basic protsedurasining – hodisalarga ishlov beruvchi; standart modullardagi shaxsiy Function va Sub; sinf modullaridagi Function va Sub – mavjudligini o'zlashtirgan holda, shu protseduralarni qanday yaratilishi va nima uchun ishlatilishini o'rgandik.*

### **Nazorat uchun savollar va amaliyot uchun topshiriqlar**

1. Talabalar o'quv loyihasidagi jadvallarni Master yordamida yarating.
2. Talabalar o'quv loyihasidagi jadvallarni Конструктор uslubidan foydalaniб yarating.

3. Talabalar o'quv loyihasidagi jadvallarni berilganlarni kiritish uslubida yaratting.
4. Jadvallarni yaratishdagi uchchala uslubiyatning o'zaro farqlarini izohlang.
5. Yaratilgan jadvallardagi berilganlarning turlari va formatlarini izohlang.
6. Jadval maydonining o'lchami va formatini izohlang.
7. Jadval maydonining «Kiritish maskasi», «imzo» va «oshkormas qiymat» xossalarini izohlang.
8. Jadval maydonining «Qiymatga qo'yilgan shart» xossasini izohlang.
9. Jadval maydonining «Xato xususida axborot» xossasini izohlang.
10. Jadval maydonining «Indeksli maydon» xossasini izohlang.
11. «Подстаковка - О'rin almashtirish» da qanday qo'shimcha imkoniyatlar bor?
12. «Фильтр (saralash)» qanday funksiyani bajaradi?
13. Avtomatik usulda shakl yaratishning mavjud imkoniyatlarini sharhlang.
14. Usta yordamida shakl yaratishdagi mavjud imkoniyatlarni sharhlang.
15. Konstruktor yordamida shakl yaratishdagi mavjud imkoniyatlarini sharhlang.
16. Shakl yaratishdagi asosiy xossalarni izohlab bering.
17. Shakl yaratishda «Макет» nima funksiyalarini bajaradi?
18. Shakl yaratishda «Berilganlar» nima funksiyalarini bajaradi?
19. Shakl yaratishda «Voqeа/bodisa» nima funksiyalarini bajaradi?
20. Shakl yaratishda «Boshqalar» nima funksiyalarini bajaradi?
21. Obyektni xususiyatlar oynasidagi «Maydon» turining vazifasi nimadan iborat?
22. Obyektni xususiyatlar oynasidagi «Ro'yxatli maydon» turining vazifasi nimadan iborat?
23. Shaklga qo'yiladigan elementlar panelini izohlang.
24. Elementlar panelidagi hamma elementlardan foydalangan holda ixtiyoriy shakl yaratting va uni izohlab bering.
25. MS Visual Basic protsedurasi nimaga ishlatalidi?
26. Protseduralarning qanday turlari mavjud?
27. Hodisalarga ishllov beruvchi protseduraning vazifasi nimadan iborat?
28. Sinf modullarining vazifasi nimadan iborat?
29. Shakl modullarining vazifasi nimadan iborat?
30. Hisobot modullarining vazifasi nimadan iborat?
31. Foydalantuvchi funksiyalarining vazifasi nimadan iborat?
32. Hodisalarga ishllov beruvchi protseduralarning vazifasi nimadan iborat?

### **34-bob. HISOBOTLARNI TUZISH**

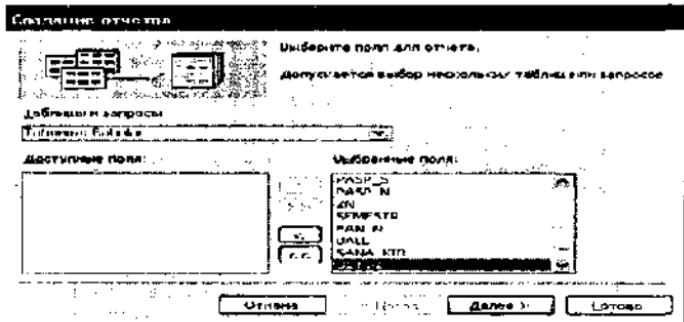
**Tayanch iboralar:** master yordamida hisobot tuzish, yuqorigi kolontitul, pastki kolontitul, guruhning sarlavhasi va ilovasi, berilganlar hududi, xulosalar hududi, yozuv, maydon, chiziq.

## Мастер yordamida hisobotlarni yaratish

Access tizimida BB axborotini ifoda etuvchi nashrli shakllar-hisobotlarni tuzish uchun bazani hisobotlar bo‘limida **Hisobotlarni Maʼterep** (*ястра*) yordamida tuzish rejimidan foydalansa bo‘ladi, bunda keyinchalik hisobotlar **Констуктор** rejimida takomillashtiriladi. Hisobotlar baza jadvallaridagi mavjud barcha axborotlar asnosida tuzilishi mumkin, lekin ko‘pincha hisobotlar uchun SQL-so‘rovidan foydalangan holda, bazadan kerakli axborotlarni saralab olish zarur bo‘ladi va shu axborot asnosida hisobot tuzish kerak bo‘ladi. Hisobotlarni muhim jihatni, bu berilganlarning guruhlash va butun hisobot va guruh uchun yakuniy natijaviy berilganlarning olish imkoniyatining mavjudligidir.

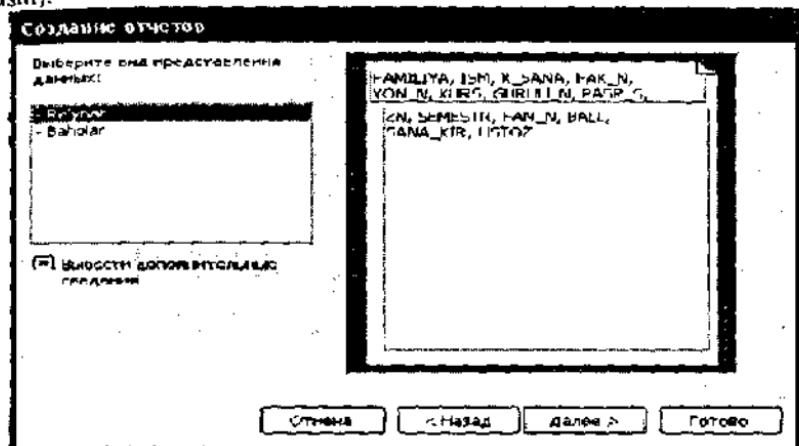
Fakultetlar, kurslar, guruhlar bo'yicha sinflangan berilganlar yordamida barcha talabalarni baholari ko'rsatilgan hisobotni tuzish vazifasini ko'ramiz.

Buning uchun Hisobotlar (Отчеты) bo'limini va Mastep yordamida hisobot tuzish rejimini tanlaymiz. Mastepni birinchi bosqichida hisobotda ko'rsatmoqchi bo'lgan bazani ajdod (bosh) jadval maydonini Ro'yhat va avlod baholar jadvali (Baholar)ni barcha maydonlarini tanlaymiz (34.1-rasmi).



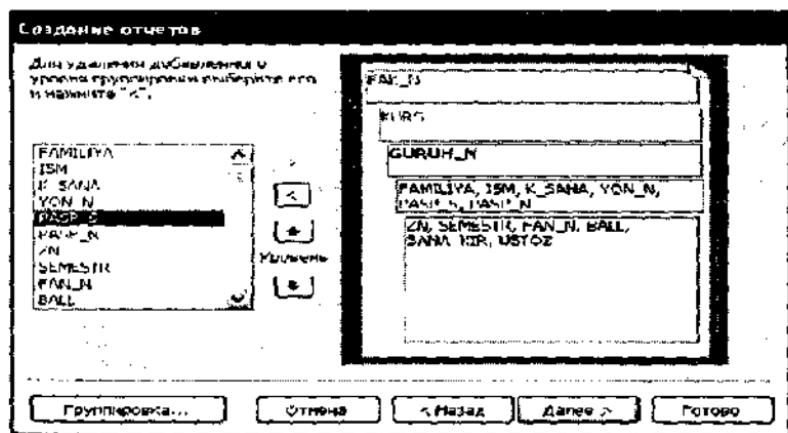
#### **34.1-rasmi. Hisobotlar Mactepida hisobotlar uchun maydonlarni taqilash.**

Ikkinci bosqichda berilganlarni namoyish etish turini tanlang - Ro'yhat jadvali ajratilib belgilanganda, birinchi variant tanlanadi (34.2-rasm).



34.2-rasm. Hisobotda atributlarni tanlash.

Uchinchi bosqichda fakultetlar, kurslar, guruhlar bo'yicha jamlangan (guruflangan) berilganlarni keltiramiz. 3 ta darajadan ortiq guruflashni Mastер bajarishga yo'l qo'yumaydi (34.3-rasm).



34.3-rasm. Hisobotda berilganlarni guruflashi.

Keyingi bosqichdag'i sarxillash (sortirovka) jarayonini keltirmaymiz.

Beshinchi bosqichda chap tarafidagi hisobot uchun mакет ko'rinishini tanlaymiz, keyingi bosqichda hisobotlar uslubi - oddiy, keyin hisobotga

**Talabalar va ularning baholari degan nom beramiz va Tayyor tugmasini bosamiz.**

**Констуктор rejimida hosil qilingan hisobot 34.4 rasmda keltirilgan.**

GURUH_N	KURS	TANSLIYA	Исполнитель	Исполнитель
Мирзалиев	100		Пастки	Пастки
Жарғиз сана	100		Пастки	Пастки
Мирзалиев	100		Пастки	Пастки

**34.4-rasm. Mastер yordamida tuzilgan hisobot.**

Olingan hisobotda 3 turdag'i obyektlar mavjud – **Maydon** – hisobotda baza yoki so'rovnomaga jadvali maydoni berilganlarini ko'rsatadi, **Yozuv** – hisobotdag'i har qanday matn va **Chiziq** – rasmiylashtirish elementi.

Констукторда hisobot alohida hududlarga ajratilgan bo'lib, u yerdagi berilganlar hisobotda bir marta aks etishi mumkin. Hisobotni sarlavhasi va ilovasi – har bir sahifani boshi va oxirida, Yuqorigi kolontitul va Pastki kolontitul – har bir guruohni boshi va oxirida, Guruhnining sarlavhasi va ilovasi – guruqlar soni ko'p bo'lishi mumkin, jadvalni har bir yozuvi uchun – **Berilganlar hududi**. Xulosalar hududi – guruqlar yoki butun hisobot uchun umumiy natijalar maydonini (yig'indi, o'itacha va h.k.) joylashtirish mumkin.

Olingan hisobotni ekranda ko'rish, printerga nashr uchun (masalan, standart instrumentlar panelida kerakli tugmalardan foydalangan holda) va MS Word yoki Excel (dastlabki ko'rish rejimi orqali)ga yuborish mumkin.

Hisobotni dastlabki ko'rish rejimidagi ko'rinishi 34.5-rasmda keltirilgan.

Талабалар ва уларнинг бахолари

## Талабалар ва уларнинг бахолари

FAK_N	1		
KURS	1		
GURUH_N	a1		
Фамилия	Зионов	Паспорт серияси	CA
Имя	Бахром	Паспорт №	1236780
Кириллица	Д1 10.1980		
Родившийся		2	
Занят - инвест. Фан		Балко	Сана Узбек
60120	2	2	80.06.2010 Бешимов И.Ю.
60120	2	1	59.06.2010 Тураев Ч.
60120	2	3	74.06.2010 Салимов С.Б.

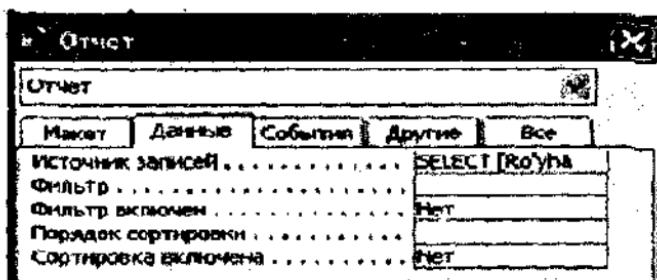
FAK_N	2		
KURS	1		
GURUH_N	Ф12		
Фамилия	Ильинов	Паспорт серияси	CH
Имя	Садик	Паспорт №	1111110
Кириллица	Д13.05.2011		
Родившийся		1	
Занят - инвест. Фан		Балко	Сана Узбек
62350	2	1	89.12.2011 Закеев Т.Т.
62350	1	3	77.12.2010 Судаков К.П.

**34.5-rasm.** Dastlabki ko'rish rejimidagi hisobot.

**Maʼtep** yordamida tuzilgan hisobot unchalik toʼliq emas va keragidan ortiq qalin hoshiyalar, yirik shriftlar bilan bejab yuborilgan. Hisobotga fakultetlar, ixtisosliklar va fan nomlarini qoʼshish va ularning imzolarini oʼzgartirish kerak. Yangi obyektlarni qoʼshishda «Elementlar paneli» yoki «Maydonlar roʼyxati» (bosh menyuda «Koʼrinish» - «Maydonlar roʼyxati») oynasidan foydalanish lozim, obyektning xususiyatlar oynasi va kontekstli menyuni oʼzgartirish uchun, sahifa parametrlarini sozlash uchun (maydon va qogʼoz oʼlchami) tizimni bosh menyusi **Файл boʼlimidagi kerakli punktdan** foydalanish lozim.

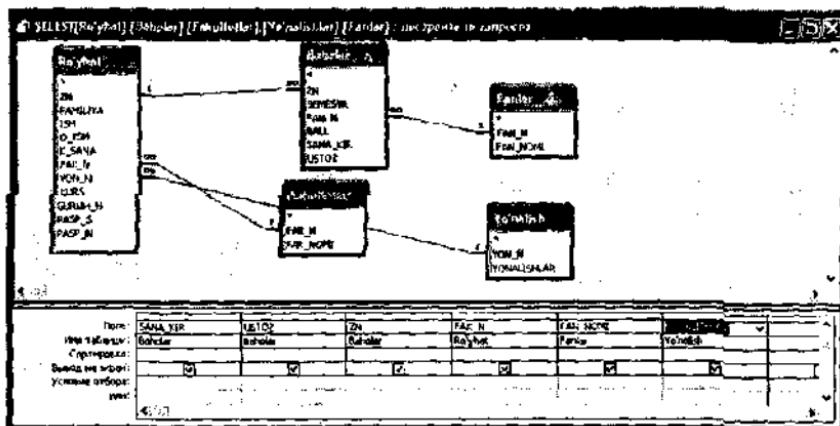
#### **Hisobotlarni normal shaklga keltirish**

Hisobotda jadvallardagi berilgantardan foydalaniш uchun ularni hisobotning xossalari oynasidagi **Vozuвлar manбasi** bo'limiga kiritish lozim (34.6-rasmga qarang).



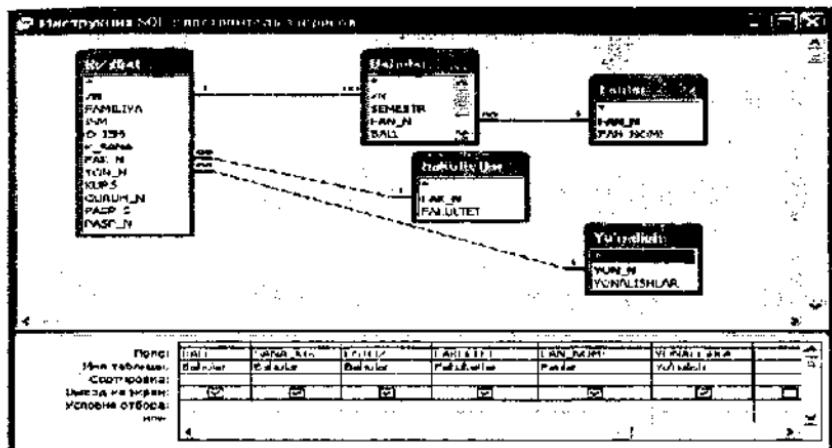
34.6-rasm. Hisobotlar xossasi oynasi, berilganlar sahifasi.

З нуqтали tasvirga ega bo'lган tugmani bosgandan keyin, so'rovlarни tashkil etuvchi oyna ochiladi, ushbu oyna kerakli jadvallar qo'shilgandan keyin 34.7-rasmida ko'rsatilgan ko'rinishiga ega bo'ladi. So'rovga jadvallarni qo'shish uchun (ekranni tashkil etuvchini yuqori qismida ko'rsatilgan) kontekstli menyudan foydalaniш mumkin.



34.7-rasm.

Shundan so'ng ekranni pastki qismida foydalilaniladigan maydonlar ro'yxatiga 3 ta – NAME\_F, NAME\_P i NAME\_S maydonlarni qo'shish lozim (ko'proq maydonlarni qo'shish ham mumkin, rasm 34.8). Keyin o'zgarishlarni saqlagan holda so'rovlarни tashkil etuvchi oynasini berkитish lozim.



34.8-rasm. «Yozuvlar manbaşı» xossasi uchun so'rovlar tashkil etuvchisi.

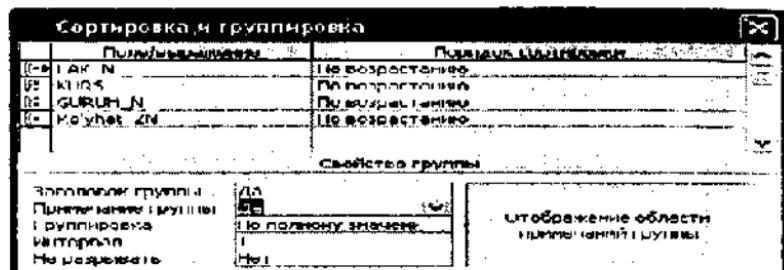
«Maydonlar ro'yxati»-«Список полей» oynasiga yangi jadvallar qo'shilgandan keyin jadvallar yangi berilganlar maydonini tuzish uchun ishlatalishi mumkin. Biz uchun berilgan ma'lumotnomalar jadvalidan maydonlar nomini tanlab olib, qo'shishimiz lozim bo'ladi. Buning uchun 3 xil usuldan foydalanish mumkin:

- Sichqoncha yordamida maydon nomini «Maydonlar ro'yxati» oynasidan Констуктордаги hisobotning kerakli joyga ko'chirib o'tkazish lozim.
- Eleminiatr panelenda «Майдон» - «Поле» tagmasini tanlab, sichqoncha bilan hisobotni yangi maydoni joylashishi kerak bo'lgan joyga chertiladi, uning asosiy xususiyatini keltirish (Свойства ойнисида) - «Данные» (ochiladigan ro'yxatdan tanlab olib) – BB jadvalini kerakli maydoni va tashqi ko'tinish bilan bog'liq bo'lgan boshqa xususiyatlari; berilganlar uchun imzo (подпись)ni o'chirib tashlashi lozim.

Hisobotni mavjud maydonidan nusxa ko'chirish (masalan, FAK\_N) va u uchun yangi xossa «Данные»ni keltirish kerak.

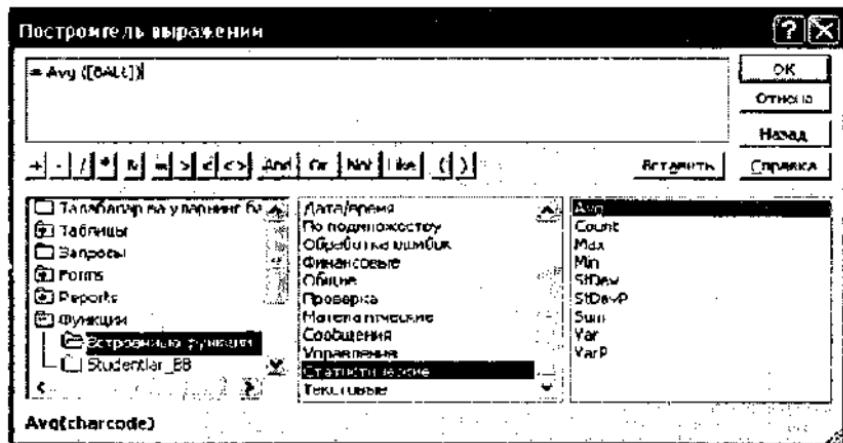
Uchinchi usul ayrim afzallikka ega – nusxa ko'chirilayotganda, mazkur hisobotni maydonlari uchun berilgan xususiyatlar bo'limi «Макет» (shrift, uning o'lchami, yorqinligi va h.k.) saqlanadi.

Bundan tashqari, «Saralash va guruhlash»-«Сортировка и группировка» oynasida (uni kontekstli menyudan yoki bosh menyuni «Ko'rinish» punktidan ochish mumkin) barcha guruhlar uchun Ilovalar mavjudligini keltiramiz (34.9-rasmga qarang).



34.9-rasm. «Saralash va guruhlash» hisoboti oynasi.

Ilovalar hududida butun hisobot va talaba, guruhlar, fakultetlar uchun o'rtacha baholarni ko'rsatadigan hisoblash maydonlarini joylashtiramiz. Hisoblash maydonlarini tashkil etishda eng osoni «berilganlar hududidan» BALL maydonini «Guruhan izohi» hududiga ko'chirishdir. Keyin ushbu obyektni xossalalar oynasiga o'tib, **берилганлар-данниые** bo'limida ko'p nuqtali tugmani bosish kerak, shundan so'ng ifodalarni tashkil etuvchi oyna ochiladi, bu erda funksiyalar ro'yuxatidan AVG (o'rtacha)ni topish va AVG (BALL) ifodasini keltirish mumkin (34.10-rasmga qarang).



34.10-rasm. Hisobotlar izohi hududi uchun hisoblash maydonlarini tuzish

Keyin hisobotga «talabaning o'rtacha bahosi» degan Yozuv joylashtiriladi, shundan so'ng ushbu maydon va yozuvni hisobotni boshqa hududiga ko'chirib olib o'tiladi. Hisobotni o'zgartirishlardan keyingi ko'rinishi 34.11 rasmda keltirilgan.

Галабалар ва уларнинг бахолари					
<input type="checkbox"/> Воситат колонтаги					
<input type="checkbox"/> Заповест групни FAK_N					
FAK_N	FAK_N				
<input type="checkbox"/> Заповест групни KURS					
KURS	KURS				
<input type="checkbox"/> Заповест групни GURUH_N					
GURUH_N	GURUH_N				
<input type="checkbox"/> Заповест групни 'Роубак' ZN					
Фамилия	FAMIL_YA	Паспорт серияси	ЗАРП_S	Паспорт №	FASF_V
Имя	СМ				
Киртти сабо	SABO				
Мундариж	YOK_N				
Зачетка №	Семестр Юрик	Бало	Сано/Учеба		
<input type="checkbox"/> Статус аз-бад					
Балоба SEMES_FAN_V		САМА	STDZ		
<input type="checkbox"/> Протектига турган FAK_N					
<input type="checkbox"/> Факультет тапабаллонинг уртача бахоси		=40%			
		GURUH_N	FAK_N		

34.11-rasm. Hisobotni Констуктордаги о'зgartirishlardan keyingi ko'rinishi.

Hisobot qog'ozga chiqarilganda 34.12-rasmida ko'rsatilgan ko'rinishiga ega bo'ladi.

## ТАЛАБАЛАР ВА БАХО

FACULTET: Механика-математикада  
Семестр/Сезон: Февраль 2008/2009

### KURS

<b>GURUH_N</b>	<b>A1</b>	Семестр/Сезон: 22.08.2008
Запись № 0005227	Ф.И.О. Салников	Группа: 10.25

Оператор: ФИО: Салников  
О дате: 00.03.2010 Четверг

Запись № 60 (2013) Ф.И.О. Жиганов Бакыт Сабирович Июльский 01.10.2007 Физика МАТ

Семестр	ФИО	База	Сем.	Учеба
2	Жиганов Бакыт	00.03.2010	Физика	10.25
1	Жиганов Бакыт	00.03.2010	Физика	10.25
2	Жиганов Бакыт	00.03.2010	Физика	10.25
1	Жиганов Бакыт	00.03.2010	Физика	10.25
2	Жиганов Бакыт	00.03.2010	Физика	10.25
1	Жиганов Бакыт	00.03.2010	Физика	10.25

<b>GURUH_N</b>	<b>A2</b>	Группа: 10.25
Запись № 5110361	Ф.И.О. Салников	Шокир Валиевич Июльский 31.08.2007 Физика МАТ

Оператор: ФИО: Салников  
О дате: 00.03.2010 Четверг

<b>GURUH_N</b>	<b>311</b>	Группа: 10.25
Запись № 6012205	Ф.И.О. Салников	Кодир Махмудуллаев Июльский 01.08.2007 Физика МАТ

Оператор: ФИО: Салников  
О дате: 00.03.2010 Суббота

Запись № 0012208 Ф.И.О. Жиганов Тимур Баситович Июльский 01.08.2007 Физика МАТ

Семестр	ФИО	База	Сем.	Учеба
2	Жиганов Тимур	00.03.2010	Физика	10.25
1	Жиганов Тимур	00.03.2010	Физика	10.25

Запись № 0012209 Ф.И.О. Жиганов Тимур Баситович Июльский 01.08.2007 Физика МАТ

Семестр	ФИО	База	Сем.	Учеба
2	Жиганов Тимур	00.03.2010	Физика	10.25
1	Жиганов Тимур	00.03.2010	Физика	10.25

Запись № 0012210 Ф.И.О. Жиганов Тимур Баситович Июльский 01.08.2007 Физика МАТ

Семестр	ФИО	База	Сем.	Учеба
2	Жиганов Тимур	00.03.2010	Физика	10.25
1	Жиганов Тимур	00.03.2010	Физика	10.25

Запись № 0012211 Ф.И.О. Жиганов Тимур Баситович Июльский 01.08.2007 Физика МАТ

Семестр	ФИО	База	Сем.	Учеба
2	Жиганов Тимур	00.03.2010	Физика	10.25
1	Жиганов Тимур	00.03.2010	Физика	10.25

Оператор: ФИО: Жиганов Тимур  
О дате: 00.03.2010

Оператор: ФИО:

34.12-rasm. Hisobotni printerdan chiqarilgan oxirgi nusxasi.

### Xulosa

*Hisobotlarni master yordamida yaratish, turli funksiyalardan foydalanish, hisobotlarning xossalariidan foydalanish, hisobotlar jadvaliar orasidagi o'zaro mantiqiy bog'lanishlarni amalgga oshirish usullari, hisobotlarni normal holga keltirish usullarini turli xil variantlarda amallyy jihatdan o'rgandik.*

*Bular bo'yicha ko'nikma hosil qilish uchun amaliy mashg'ulotlar davomida turli mashqlarini, jumladan quyida keltirilgan mashqlarni amalgga oshirish zarur.*

## **Nazorat uchun savollar va amaliyot uchun topshiriqlar**

1. Hisobotlarni qanday uslublarda tayyorlash mumkin?
2. Hisobot nima asosida tuziladi?
3. Hisobotlarni yaratishda nimalardan foydalilanadi?
4. Hisobotlarni namoyish etish turlarini izohlang.
5. Hisobot shakllaridagi obyektlarning turlarini izohlang.
6. Hisobotlarni eksport qilish mumkinmi?
7. Hisobotlarni yaratishdagi Mactep va Konstuktor uslublaridagi farqlarni izohlab bering.
8. Hisobotda jadvallardagi berilganlardan foydalanish uchun nima qilish kerak?
9. Hisobotda so'rovlardan foydalanish uchun nima nima qilish kerak?
10. «Guruhash» amalidagi asosiy vazifa nimadan iborat?
11. «Guruhash» amalida ishlataladigan funksiyalarni izohlang.
12. Talabalar o'quv loyihasidagi, masalan Ro'yhat jadvalidagi «Biologiya» fakultetida tahsil olayotgan 3 kurs talabalari bo'yicha 5 semestrdagi ixtiyoriy fan asosida jadvaldagi ma'lumotlar sonini 20 taga yetkazing.
13. Tayyorlangan Ro'yhat jadvali asosidagi hisobotni Mactep yordamida tayyorlang.
14. Tayyorlangan Ro'yhat jadvali asosidagi hisobotni Konstuktor yordamida tayyorlang.
15. Tayyorlangan hisobotlarni MS Word va Excel ga eksport qiling va natijani izohlang.
16. Berilganlarni «Saraksib» amalidalar foydalangan holda hisobot yaratting.
17. «Guruhash» dagi AVG funksiyasidan boshqa, ixtiyoriy funksiyadan foydalangan holda hisobot Mactep yordamida yarating va natijani izohlang.
18. «Guruhash» bo'yicha Matn dagi joylangan ixtiyoriy funksiyadan foydalangan holda hisobot Mactep yordamida yarating va natijani izohlang.
19. «Guruhash» dagi AVG funksiyasidan boshqa, ixtiyoriy funksiyadan foydalangan holda hisobot Konstuktor yordamida yarating va natijani izohlang.
20. «Guruhash» bo'yicha Matn dagi joylangan ixtiyoriy funksiyadan foydalangan holda hisobot Konstuktor yordamida yarating va natijani izohlang.

## **35-bob. SO'ROVLARDAN FOYDALANISH**

**Tayanch iboralar:** so'rovlar konstraktori, yozuvlar, kengaytirilgan filtr, konstruktur, master, MS Access, master yordamida so'rovlar tuzish, so'rov maketini o'zgartirish, UPDATE, DELETE, INSERT, MDE-fayl, berilganlarga kirish huquqi sahifalari, Internet Explorer, MS Access toyihasi, «mijoz/server» arxitekturasi, MS SQL Server, Toad Data Modeler, DDL Script.

### **So'rovlarni tashkillashtirish**

Avval eslatib o'tilganidek, biror bir shartlarga muvofiq tanlab olingan berilganlar bilan ishlash uchun BB yoki shakl jadvallari uchun filtr o'matish imkoniyatidan foydalanish mumkin (**Yozuvlar** menu bo'limida). Xuddi shu menu bo'limida **So'rovlarni konstruktori** oynasini ochuvchi **Kengaytirilgan filtr** bo'limi mavjud. Bundan tashqari, **So'rovlarni konstruktori** xizmatidan hisobot uchun yozuvlar manbasini ta'riflashda foydalilanigan (34.7 va 34.8-rasmrlarga qarang).

**MS Access** tizimida so'rovlar bilan ishlash hamda ularni bazada saqlash uchun maxsus bo'lim mavjud bo'lib, ushbu bo'lim **Konetyuktor** rejimida yoki **Master** yordamida yangi so'rovlarni tuzish imkonini beradi.

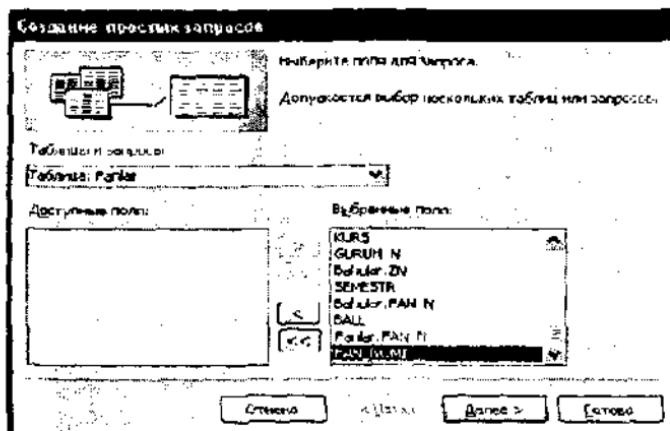
**MS Access** tizimida so'rovlarni asosiy quyidagi bir necha xili xususida qo'shimcha ravishda to'xtab o'tamiz (so'rovlarning turlari xususida biz batatsil 22- bobda to'xtab o'tgan edik):

- tayinlangan murakkab shartlar bo'yicha berilganlarni bir yoki bir necha BB jadvallaridan tanlab olish uchun so'rov, bunda berilganlar hisoblash uchun guruhi langan tarzda so'rov natijalari esa jadval shaklida namoyish etiladi yoki natijalar shakl va hisobotlarda qo'llangan holda ko'rsatiladi; so'rovlar jadvalida berilganlar tahrirlangandan keyin BB jadvalini yangilash mumkin (ba'zi bir cheklovlardan bilan);
- ikki ifodani guruhi langan orqali, natijaviy ikki ustunli jadvallarni hosil qiluvchi kesishgan so'rov, ifodalardan biri satrlar sarlavhasi, boshqasi – ustun sarlavhasi bo'ladi;
- yangi jadval tuzishga oid so'rovlar;
- berilganlarni o'zgartirishga doir so'rovlar;

- o berilganlarni yangilash – bir yoki bir necha jadvallarni yozuvlar guruhiga umumiy o'zgartirishlar kiritish buyrug'i;
- o berilganlarni qo'shish – bir yoki bir necha jadvallarni yozuvlar guruhini bir yoki bir necha jadvallarning oxiriga qo'shish buyrug'i;
- o berilganlarni o'chirish – bir yoki bir necha jadvallardan yozuvlar guruhini o'chirish buyrug'i.

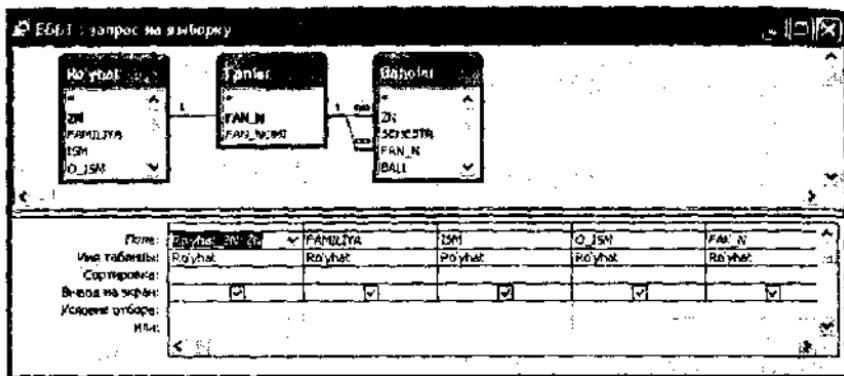
So'rovlarini hosil bo'lish tamoyilini so'rovlar masteridan foydalanish orqali osongina o'zlashtirib olish mumkin. Faraz qilaylik, biz sakkizinchи semestr natijalariga ko'ra, BBBT fanidan faqat «a'lo» baholarga ega talabalarni tanlab olishimiz kerak, so'rovni tuzish uchun bazani so'rovlar bo'limida **Мастер yordamida so'rovlar tuzish** rejimini tanlaymiz.

Birinchi bosqichda so'rovgaga kiritilishi lozim bo'lgan jadval va maydon tanlab olinadi. Maydonlarni tanlash bazani bir nechta jadvallari orqali amalga oshiriladi. Bizning misol uchun **Ro'yhat** jadvalidan, faqat **K\_CANA, PASP\_S** va **PASP\_N** tashqari, barcha maydonlarni tanlab olamiz, Baholar jadvalidan - birinchi 4 ta maydonni va Fanlar jadvalidan 2 ta maydonni tanlab olamiz (35.1-rasmi).



35.1-rasm. So'rovlar Masterida maydonlarni tanlash.

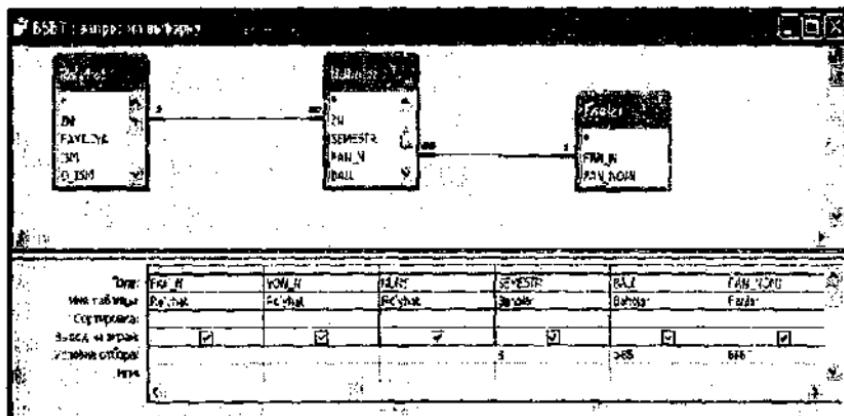
Ikkinci bosqichda («batafsil yoki natijaviy hisobot») batafsil hisobotni tanlaymiz. So'nggi 3- basqichda so'rovni BBBT deb nomlaymiz va So'rov mакетини o'zgartirish xizmat turini tanlaymiz, shundan so'ng Tayyor tugmasini bosamiz. So'rovlar konstrukturida so'rov ochiladi, uning ko'rnishi 35.2-rasmida ko'rsatilgan.



35.2-rasm. So'rovlar Konstruktori.

So'rovlar Конструкторини ўзори qismida berilganlarni tanlash va ular orasini bog'lash uchun xizmat qiladigan jadval keltirilgan, qiy'i qismida esa - maydonlarni tanlash va berilganlarni guruuhlash uchun jadvallar (agar «Guruhli faoliyatlar» satri bo'lmasa, u holda MS Access ning bosh menu-sini «Vid - Ko'rinish» bo'simida ushbu buyruqni tanlash lozim), saralash vazifalari va tanlash shartlari mavjud (22- bobdag'i QBE to'rini eslang).

Berilganlarni saralash vazifasi va talabalarni ismi sharifi bo'yicha tartib bilan joylashtirish shartlariga muvofiq so'rovlarни o'zgartiramiz. Buning uchun **FAMILIYA** maydon ustunchasida ko'payish shkalasi bo'yicha saralashni amalga oshiramiz. **SEMESTR** maydoni uchun saralash sharti =8 ni (sakkizinchi semestr), **BALL** maydoni uchun >85 shartini va **FAN\_NO-MI** maydoni uchun «BBBT» shartini keltiramiz (35.3-rasm).



35.3-rasm.

Shuningdek, siz ekranda ko'rsatishni istamagan maydonlardan belgilarni olib tashlashingiz mumkin.

So'rovni saqlaymiz va uning matnini SQL rejimida (Structured Query Language) ko'rishimiz mumkin. So'rov matni quyidagi ko'rinishda bo'ladi:

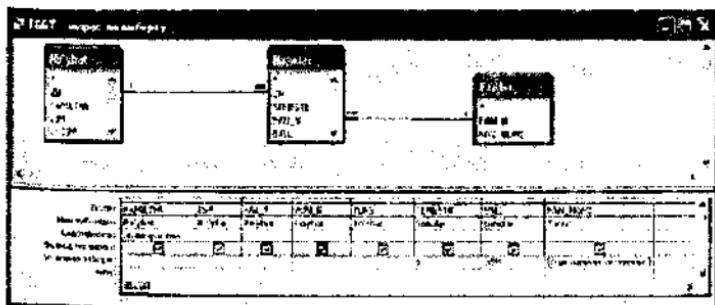
```
SELECT [Ro'yhat].FAMILIYA, [Ro'yhat].ISM,
[Ro'yhat].FAK_N, [Ro'yhat].YON_N, [Ro'yhat].KURS,
Baholar.SEMESTR, Baholar.BALL, Fanlar.FAN_NOMI
FROM [Ro'yhat] INNER JOIN (Fanlar INNER JOIN Baholar ON
Fanlar.FAN_N = Baholar.FAN_N) ON [Ro'yhat].ZN = Baholar.ZN
WHERE (((Baholar.SEMESTR)=8) AND ((Baholar.BALL)>85))
AND ((Fanlar.FAN_NOMI)=<>BBBT>))
ORDER BY [Ro'yhat].FAMILIYA
```

Конструктор oynasini yopamiz va so'rovni **Ochish** buyrug'i yoki sich-qonchani ikki marta chertish orqali amalga oshiramiz. So'rovni bajarish natijasi ekranda jadval ko'rinishida namoyon bo'ladi. (35.4-rasm). *Shuni nazarda tutish lozimki, ushbu jadvaldag'i berilganlarni tahrirlash BB jadvallaridagi axborotlarni o'zgarishiga olib keladi!*

Фамилия	Имя	Факультат	Информатика	Гуре	Семестр	Балл	Фан номи
Шарифов	Салим	Биология	БИОИНФОРМ	1	3	89,5557	
Эшонов	Дахром	Механика-математика	МАМАТ	1	3	86,5557	

35.4-rasm. So'rovni amalga oshirish natijalari.

Agarda saralash shartida kvadrat qavs ichida matn yozilsa, so'rovni amalga oshirish jarayonida ushbu parametrni kiritish uchun oyna paydo bo'ladi. Masalan, Agarda **FAN\_NOMI** maydoni uchun shartda **Fan nomini keltiring** yozuvini yozsak, turli xil predmetlar bo'yicha berilganlarni saralash uchun bir xil so'rovdan foydalanish mumkin (35.5-rasmga qarang).

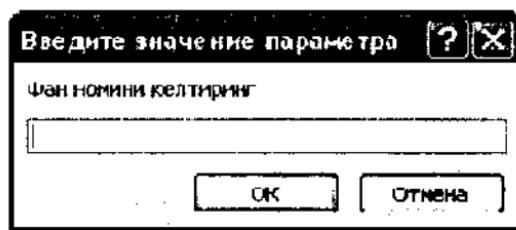


35.5-rasm.

So'rov matni quyidagi ko'rinishda bo'ladi:

```
SELECT [Ro'yhat].FAMILIYA, [Ro'yhat].ISM,  
[Ro'yhat].FAK_N, [Ro'yhat].YON_N, [Ro'yhat].KURS,  
Baholar.SEMESTR, Baholar.BALL, Fanlar.FAN_NOMI  
FROM [Ro'yhat] INNER JOIN (Fanlar INNER JOIN Baholar ON  
Fanlar.FAN_N = Baholar.FAN_N) ON [Ro'yhat].ZN =  
Baholar.ZN  
WHERE ((Baholar.SEMESTR)=8) AND ((Baholar.BALL)>55)  
AND ((Fanlar.FAN_NO - MI)=|Fan nomini keltiring|))  
ORDER BY [Ro'yhat].FAMILIYA
```

So'rovni bajarish natijasida avval quyidagi (35.6-rasm) oyna ochiladi va u «Fan nomini keltiring» deb, uning nomini kiritishni kutadi.



35.6-rasm.

Fanning nomi kiritilgandan so'ng, so'rovni bajarish natijasi ekranda 35.7-rasmndagi jadval ko'rinishida namoyon bo'ladi.

Фамилия	Имя	Факультет	Инвалид	Курс	Семестр	Балл	Фан номари
Жарилов	Кодир	Механика-математика	ИАТ	4	8	71	6667
Ичигов	Салим	Биология	Биоинформ	4	8	69	6667
Салимов	Шокир	Физика	АМАТ	4	8	66	6667
Саидиков	Санжар	Механика-математика	АМАТ	4	8	60	6667
Тошев	Ислок	Химия	ИАТ	4	8	76	6667
Халиков	Темур	Механика-математика	ИАТ	4	8	73	6667
Эмбиров	Бахром	Механика-математика	АМАТ	4	8	86	6667

35.7-rasm.

So'rovni amalgalashish natijalari yoki jadvallar berilganlarini diagramma va grafik shaklda namoyon etish mumkin.

«BBBT» (fan № = 4) fani bo'yicha talabalar guruhini o'rtacha to'plagan ballarini grafik shaklda ko'rsatadigan so'rov nomasi beramiz. Berilganlarni guruhlash uchun, yuqorida qayd etilganidek, tizimni menyu-sini «Ko'rinish» punktidagi «Guruhli amaliyotlar» satriga belgi qo'yamiz.

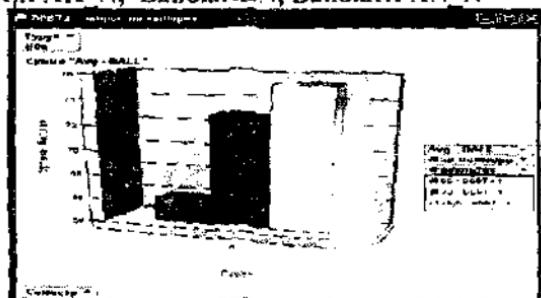
Quyidagi so'rov matniga ega bo'lamiz:

```
SELECT DISTINCTROW [Ro'yhat].GURUH_N, [Ro'yhat].FAK_N,
Fanlar.FAN_NOMI, Baholar.SEMESTR AS [First - SEMESTR],
Avg(Baholar.BALL) AS [Avg - BALL]
FROM [Ro'yhat] INNER JOIN (Fanlar INNER JOIN Baholar ON
Fanlar.FAN_N = Baholar.FAN_N) ON [Ro'yhat].ZN = Baholar.ZN
GROUP BY [Ro'yhat].GURUH_N, [Ro'yhat].FAK_N,
Fanlar.FAN_NOMI, Baholar.SEMESTR
HAVING ((([Ro'yhat].FAK_N)=<1>) AND ((Fanlar.FAN_NOMI)=
(JFan nomi)) AND ((Baholar.SEMESTR)=8))
```

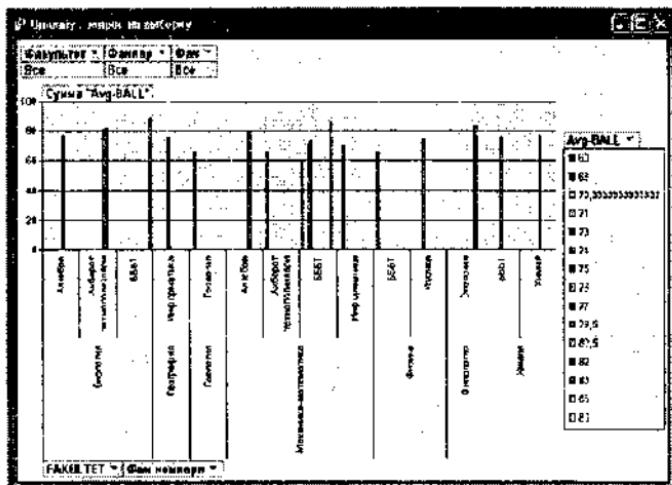
Berilganlami grafik shaklida namoyon etish uchun «Ko'rinish» menyusida Yig'ma diagramma punktini tanlaymiz, shundan keyin Diagrammalar tuzuvchini oynachasi ochiladi. Diagrammalarni rasmiylashtirish usullari MS Word yoki Excel programmalarida foydalaniладigan obyekt MS Graph diagrammasiga o'xshash 35.8-rasmida yuqorida keltirilgan so'rov uchun diagramma ko'rsatilgan.

35.9-rasmda quyidagi so'rov shakli uchun uch o'lbchamli diagramma keltirilgan:

```
SELECT DISTINCTROW Fakultetlar.FAK_N,
Fakultetlar.FAKULTET,
Fanlar.FAN_N, Fanlar.FAN_NOMI, [Ro'yhat].ZN,
[Ro'yhat].FAK_N,
Baholar.ZN AS [First-Baholar_ZN], Baholar.FAN_N AS [First-
Baholar_FAN_N], Avg(Baholar.BALL) AS [Avg-BALL]
FROM (Fakultetlar INNER JOIN [Ro'yhat] ON Fakultetlar.FAK_N =
[Ro'yhat].FAK_N) INNER JOIN (Fanlar INNER JOIN Baholar ON
Fanlar.FAN_N = Baholar.FAN_N) ON [Ro'yhat].ZN = Baholar.ZN
GROUP BY Fakultetlar.FAK_N, Fakultetlar.FAKULTET,
Fanlar.FAN_N, Fanlar.FAN_NOMI, [Ro'yhat].ZN,
[Ro'yhat].FAK_N, Baholar.ZN, Baholar.FAN_N
```



35.8-rasm. Diagramma shaklida keltirilgan berilganlarni guruhlash so'rovini amalga oshirish natijalari.



**35.9-rasm.** Uch o'chamli diagramma shaklida keltirilgan berilganlarni guruhlash so'rovini amalga oshirish natijalari.

Bitta komandanasi ishlatgan holda, boshqa shakklardagi so'rovlaridan foydalanish orqali qaysidir shartlar asosida saralangan jadval yozuvlarining to'plamidagi berilganlarni o'zgartirish (SQL buyrug'i **UPDATE – Yangilash**) yoki o'chirish (SQL buyrug'i **DELETE – O'chirish**) shuningdek, boshqa jadvallardan yozuvlar qo'shish mumkin (SQL buyrug'i **INSERT-Kiritish**) (23- bobga qarang).

### Berilganlar bazasini kompilatsiya qilish (MDE-fayl)

MS Access berilganlar bazasini MDE-fayl ko'rinishida saqlash mumkin.

Buning uchun bosh menyuda «Xizmat/Сервис» - «Xizmatchi programmalar» - «MDE-fayl tuzing» punktini tanlab, keyin faylga nom berish lozim.

Bazani MDE formatida saqlashda barcha programmalash modullari kompilatsiya etiladi, programmani barcha dastlabki matnlari o'chiriladi va zichlanadi. Visual Basic programmalari avvalgidek ishlaydi, lekin ularni ko'rish yoki o'zgartirish mumkin bo'lmaydi. Dastlabki kodlarni o'chirilishi tufayli BBni o'chami kichrayadi, ya'ni BBni tezkor ishlashini oshishiga elib keladigan xotiradan optimal foydalanish imkoniyati tug'iladi.

BB bilan ishlash tarzi o'zgarmaydi. Foydalamuvchilar avvalgidek berilganlarni yangilashlari, ekrani shakkarni ishga tushirishlari va hisobotlarni rasmiylashtirishlari mumkin bo'ladi.

**Конструктор** va **Мастер** xizmatidan foydalangan holda jadvallar tuzilmasini o'zgartirish, yangi jadvallar tuzish, jadvallarni va ular bilan bog'liq shakllarni qo'shish va olib tashlash orqali BB chizmasini o'zgartirish mumkin.

BB MDE formatida saqlanganda, baza bilan ishlash uchun ilovalar tarkiblarini tuzish bilan bog'liq bo'lgan quyidagi faoliyatlarini amalga oshirish mumkin bo'lmaydi:

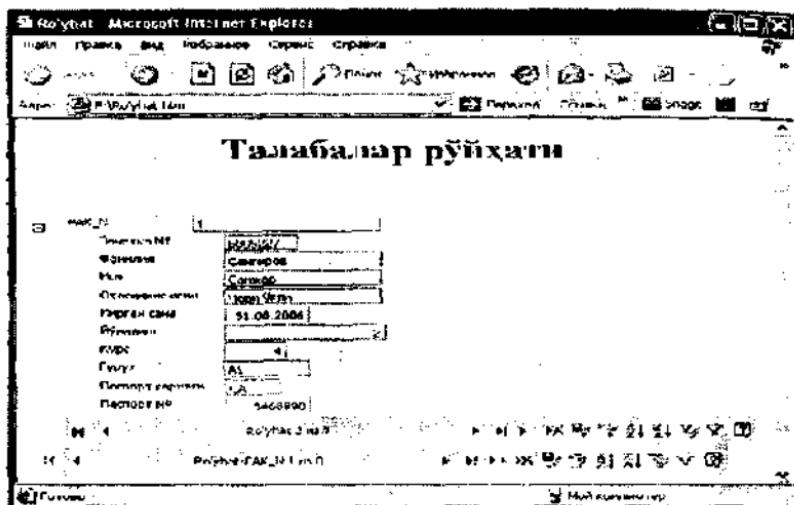
- tashkil etish, ko'rish, **Конструктор** va **Мастер** xizmatidan foydalangan holda shakllar, hisobotlar, modullarni o'zgartirish;
- programmalash modullarini tuzish va o'zgartirish;
- BB yoki obyektlar kutubxonasidagi havolalarni tuzish, o'chirish va o'zgartirish.

### Berilganlarga kirish huquqi sahifalari

Berilganlarga kirish huquqi sahifalari, shuningdek MS Access yoki MS SQL Server BBda saqlanadigan, Internet orqali yoki berilganlar mavjud bo'lgan ichki tarmoq yordamida ko'rish va ishlash uchun mo'ljallangan Web-sahifalarining maxsus shaklini namoyon etuvchi boshqa manbalarning berilganlarini o'z ichiga olishi mumkin. Berilganlarga kirish huquqi sahifalari, shuningdek, boshqa manbalarning masalan, MS Excel berilganlarini ham o'z ichiga qamrab olishi mumkin.

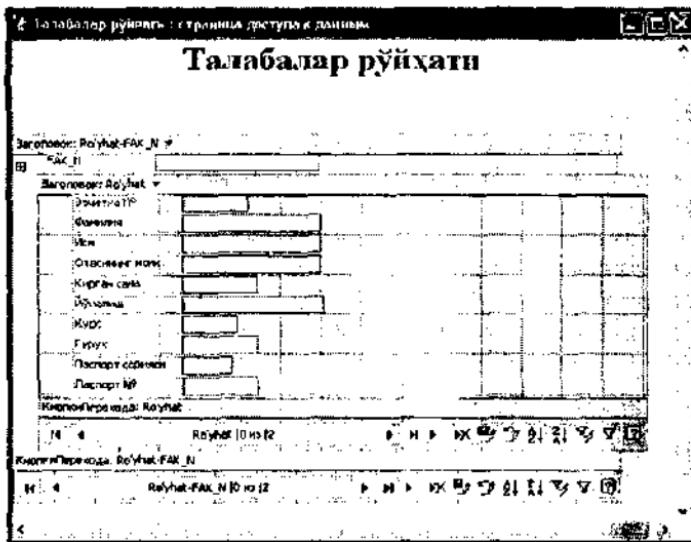
Berilganlarga kirish huquqi sahifalaridan berilganlar bilan ishlashda foydalanish ekrani shakllar bilan ishlash tarzi deyarli bir xildir: foydalanuvchi BBni jadvallarida yozuvlarni ko'rish, tahrirlash, qo'shish va olib tashlash imkoniyatiga egadir. Biroq sahifadan MS Access tizimining imkoniyatlari chegarasi tashqarisida ham foydalanish mumkin, bunda SP2 paketli MS Internet Explorer 6.01 brauzeridan yoki eng so'nggi naqlidan foydalangan holda foydalanuvchi Internet yoki intratarmoq orqali berilganlari ko'rish va yangilash imkoniyatiga ega bo'ladi. Sahifalar elektron shaklda elektron pochta yordamida tarqatilishi mumkin, qabul qiluvchilar har bir xabarni ochish vaqtida joriy berilganlarni ko'ra oladilar.

35.10-rasmida, konstruktor yordamida MS Access tizimida tashkil etilgan **Ro'yhat.htm** sahifasiga Internet Explorer orqali kirilganda, ochilgan sahifaning misoli keltirilgan. Bir yozuvdan ikkinchi yozuvga o'tish, shuningdek, yozuvlarni qo'shish, o'chirish, saqlash, saralash tanlash va berilgan ma'lumotnomani chaqirish uchun standart tugnalar to'plami joriy etilgan.



35.10-rasm. BB bilan ishlash rejimida berilganlarga kirish huquqi sahifasi.

**Конструкторда** kirish sahifasi bo'limlarga (Section) bo'linadigan guruhlardan tashkil topgan: imzo, yuqorigi kolontitul, pastki kolontitul, o'tish tugmalari keltirilgan (35.11-rasmga qarang).



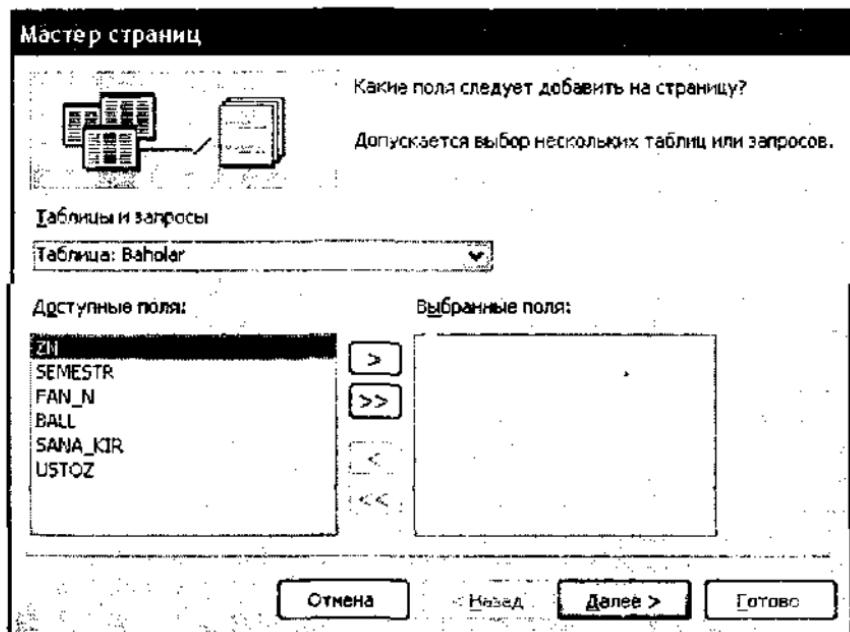
35.11-rasm. Конструкторда berilganlarga kirish huquqi sahifasi.

Berilganlarga kirish huquqi sahifasida har bir guruhlash darajasi yozuvlar manbasiga ega. Yozuvlar manbasi nomi mazkur guruhlash darajasi uchun har bir bo'lim sarlavhasini satrlarida aks ettiriladi.

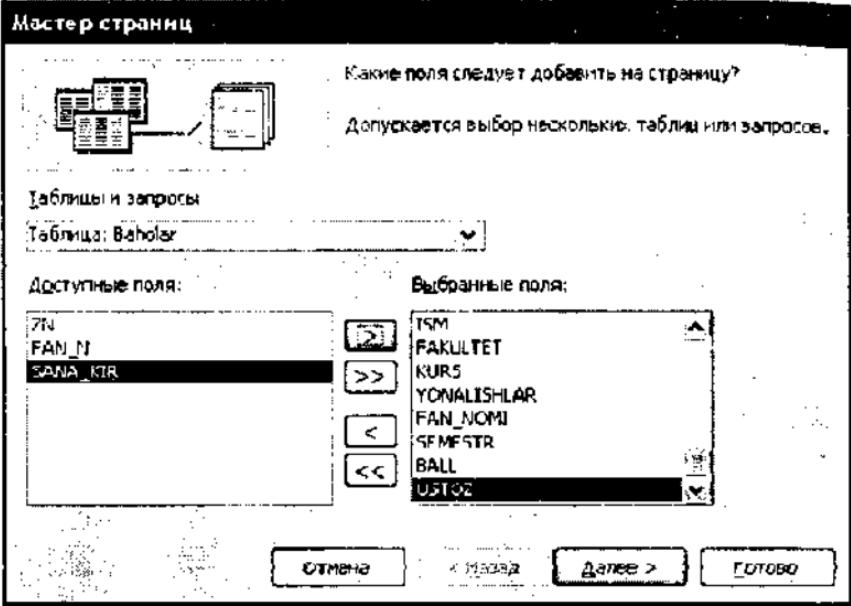
Sahifa Access ni imkoniyatlari chegarasi tashqarisida saqlanadigan alohida faylni o'zida namoyon etadi. Usibu faylni tuzishda MS Access uni BB oynasini yorliqchasiga qo'shadi. Berilganlarga kirish huquqi sahifasini tuzish shakl va hisobtlarni tuzish jarayoni bilan bir xildir: maydonlar ro'yxati, elementlar paneli va hokazolardan foydalaniadi. Shunday bo'lsada, ma'lumotlarga kirish huquqi sahifasini tuzish usullari shakl va hisobtlarni tuzishdan bir necha marta farqlanadi.

Sahifalarda guruhlash va hisoblash maydonlarini o'zgarishiga yo'l qo'yilmaydi. Agar kursorni sahifarang biron bir maydoniga qo'yish imkon bo'lmasa, sahifa yozuvlardagi o'zgarishlarni saqlamaydi.

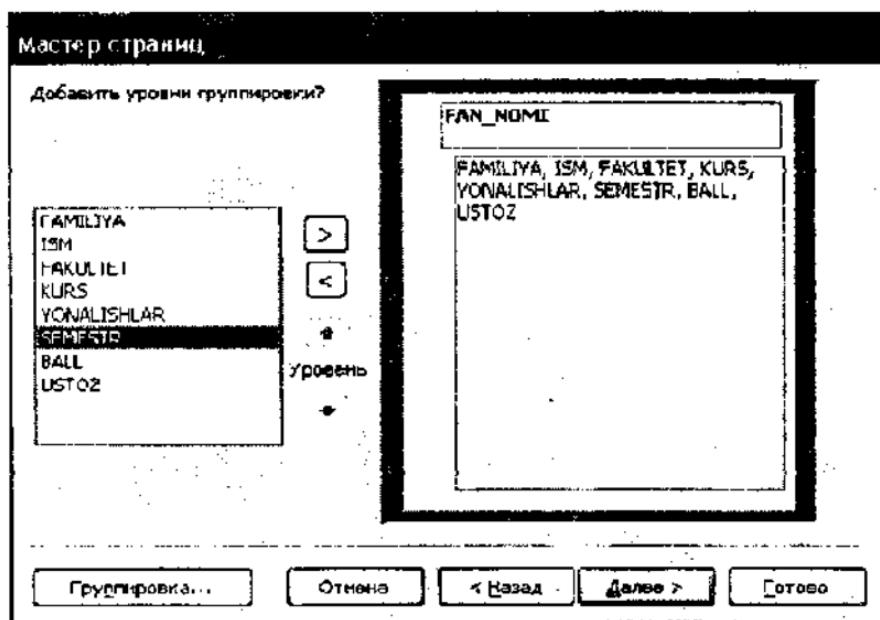
Sahifalarni master yordamida yaratish bosqichlari quyidagi 35.12 – 35.18-rasmarda keltirilgan.



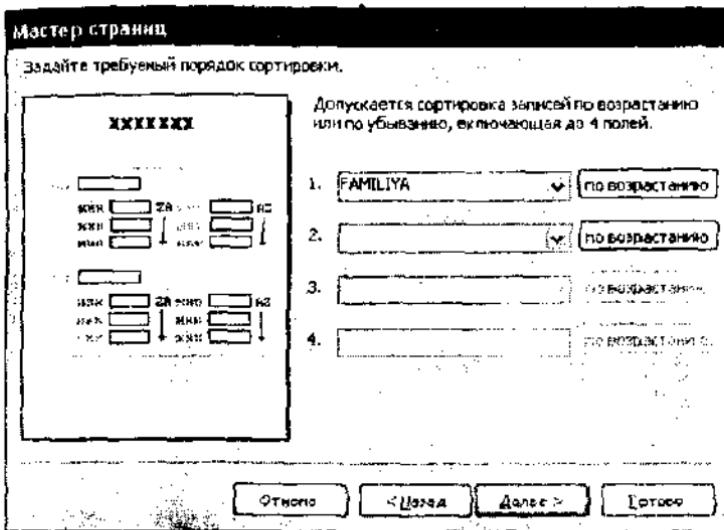
35.12-rasm.



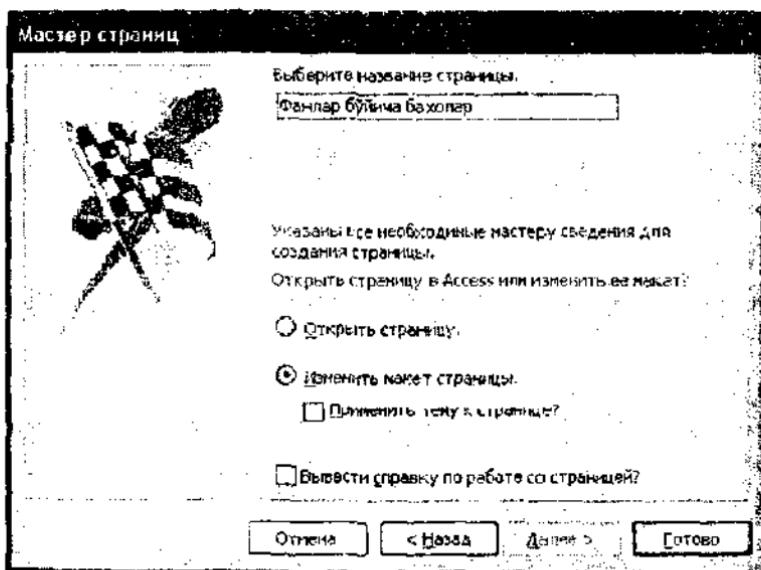
35.13-rasm.



35.14-rasm.



35.15-rasm.



35.16-rasm.

Страница2 : страница доступа к данным

## Фанлар бўйича баҳолар

Заголовок: Baholar FAN\_NOMI =

ИД: FAN\_NOMI

Заголовок: Баҳолар № =

Фондия									
ИСН									
Факультет									
Курс									
Инженерия									
Сенестр									
Бало									
Учтас - инженерия									

Кнопки перехода: Клавиши:

ИД: FAN\_NOMI [стр 1 из 2] Курс: 1 из 1 Сенестр: 1 из 1 Бало: 1 из 1 Учтас: 1 из 1

Кнопки перехода: Baholar-FAN\_NOMI

ИД: FAN\_NOMI [стр 1 из 2] Курс: 1 из 1 Сенестр: 1 из 1 Бало: 1 из 1 Учтас: 1 из 1

35.17-rasm.

Э Фанлар бўйича баҳолар

## Фанлар бўйича баҳолар

Заголовок: Baholar FAN\_NOMI =

ИД: FAN\_NOMI

Фондия									
Харбос									
Темза									
Муассин-штадион									
Курс	4								
Инженерия									
Сенестр	5								
Бало	53								
Учтас	Саломон								

Кнопки перехода: Баҳолар № =

ИД: FAN\_NOMI [стр 1 из 2] Курс: 1 из 1 Сенестр: 1 из 1 Бало: 1 из 1 Учтас: 1 из 1

35.18-rasm.

## **MS Access loyihasi (ADP)**

MS Access loyihasi (\*.adp) berilganlarini fayli hisoblanib, MS SQL Server berilganlari bazasi bilan samarali ishlash jarayonini ta'minlaydi. MS Access loyihasini qo'llash orqali fayl serveri ilovasiga o'xshab «mijoz/server» arxitekturasida ilovalarni osongina tuzish mumkin.

«Mijoz/server» arxitekturasidagi ilova shakllar va hisobotlarga asoslangan an'anaviy xulosa yoki berilganlarga kirish sahifasi foydalilanigan Internet bazasidagi xulosalar bo'lishi mumkin.

MS Access loyihasini MS SQL Server ni alohida BBga, MS SQL Server ni mahalliy BBga yoki MS SQL Server 2000 Desktop Engine (MSDE) mahalliy qurilmasiga ulash mumkin.

Loyilha faylida berilganiar va obyektlarni aniqlaydigan jadvallar va tushunchalar mavjud emas.

Loyihada BB bilan ishlash uchun berilganlarga kirish sahifasining hisobotlari, shakllari, so'rovlar, modullarini tuzish hamda ulardan foydalinish mumkin.

MS Access loyihasi bilan ishlash haqidagi ma'lumotlar MS SQL Server tizimiga bag'ishlangan keyingi boblarda keltirib o'tiladi. MS Access loyihasi bilan biz dastavval 20- bobda tanishgan va Toad Data Modeler muhitini orqali DDL Script ni MS Access ga jo'natgan edik.

## **Xulosa**

*Biz ushbu bobda, so'rovlarini tashkillashtirishning turli variantlari bilan ishladiq. Oddiy va murakkab so'rovlarini tashkil qilishni o'rgandik. Avvalgi boblarda olgan nazariy bilimlarimizni, so'rovlarini tashkillashtirish va Talabalar o'quv loyihasining berilganlar bazasidagi jadvallar yordamida amalga oshirdik.*

*SQL tili va uning DML qismi orqali so'rovlarini yaratdik, ularni QBE to'ridagi variantini ko'rdik, aksincha variantni ham amalga oshirib namoyishli natijalarini oldik.*

*Berilganlarga kirish huquqi sahfalari, shuningdek MS Access yoki MS SQL Server BBda saqlanadigan. Internet orqali yoki berilganlar mavjud bo'lgan ichki tarmoq yordamida ko'rish va ishlash uchun mo'ljallangan Web-sahifalarning maxsus shaklini namoyon etuvchi boshqa manbalarning berilganlarini o'z ichiga oladi. Bu sahifani amaliyotda sinab ko'rdik.*

## **Nazorat uchun savollar va amaliyot uchun topshiriqlar**

1. MS Access BBB tizimida so'rovlarni yaratishda asosan nechta turdan foydalaniadi?
2. MS Access BBB tizimida so'rovlarni yaratishdagi asosiy birinchi xilni tavsiflang.
3. MS Access BBB tizimida so'rovlarni yaratishdagi asosiy ikkinchi xilni tavsiflang.
4. MS Access BBB tizimida so'rovlarni yaratishdagi asosiy uchinchchi xilni tavsiflang.
5. MS Access BBB tizimida so'rovlarni yaratishdagi asosiy to'rtinchi xilni tavsiflang.
6. So'rov bir nechta jadvallar ustida amalga oshirilishi uchun eng asosiy shart nimadan iborat?
7. So'rov natijasida hosil bo'lgan jadvalni saqlash mumkinmi?
8. So'rov natijasida hosil bo'lgan jadvalni tahrirlashi mumkinmi?
9. So'rov natijasida hosil bo'lgan jadval tahrirlansa, nima bo'ladi?
10. MDE formatdagi fayllar nimani bildiradi?
11. MDE formatdagi fayllar ustida qanday amallarni bajarish mumkin?
12. MDE formatdagi fayllar ustida qanday amallarni bajarish mumkin emas?
13. Berilganlarga kirish huquqi sahifalari nimaga xizmat qiladi?
14. Berilganlarga kirish huquqi sahifalari orqali foydalanuvchilar qanday ishlarni bajarishi mumkin?
15. Konstruktor orqali berilganlarga kirish huquqi sahifasini hosil qiling va izohlang.
16. Mactep orqali berilganlarga kirish huquqi sahifasini hosil qiling va izohlang.
17. Berilganlarga kirish huquqi sahifalarida qanday imkoniyatlar mavjudligini izohlang.
18. Internet orqali berilganlarga kirish huquqi sahifalaridan foydalanish mumkinmi?
19. Internet orqali berilganlarga kirish huquqi sahifalaridan foydalanishda qanday amallarni bajarish mumkin?
20. MS Access loyihasini qo'llash orqali «mijoz/server» arxitekturasida ilovalarni yaratish mumkinmi?
21. ADP kengaytirmasi nimani bildiradi?

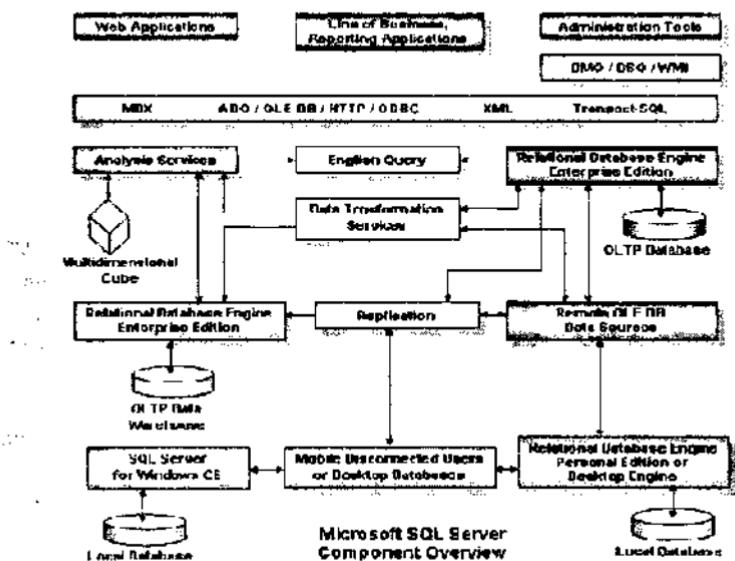
## **36-bob. MS SQL SERVER**

**Tayanch iboralar:** *MS SQL Server, NTFS, FAT, MS Windows NT, berilganlar bazasi hajni, berilganlar bazasidagi obyektlar soni, bitta kompyuterdag'i server nusxasining soni, bitta server nusxasidagi BB soni, berilganlar bazasidagi fayllar soni, baza jadvalidagi maydonlar soni, berilganlar faylining hajmi, identifikatorning uzunligi, joylangan protseduralar darajasi, kiritilgan so'rovlar darajasi, bitta baza jadvali uchun noklaster indekslar soni, bitta indeksdagi maydonlar soni, bitta indeksdagi bayt soni, bitta so'rovdag'i jadvallar soni, binary, varbinary, image, bit, shar, varchar, text, nchar, nvarchar, ntext, bigint, int, smallint, tinyint, decimal, numeric, float, real, datetime, smalldatetime, money, smallmoney, timestamp, unique identifier, sql\_variant, sysname, installatsiya, Installation Definition, Client Tools Only, Connectivity Only, autentifikatsiya, litsenziya, nomlangan nusxa, nusxa, SQL Server Service Manager.*

### **Tizimning umumiy tavsifi**

MS SQL Server - «mijoz-server» arxitekturasida BB bilan ishlaydigan eng quvvatli tizimlardan biri. Tizimning aszalliklari shundaki, server faqat MS Windows NT - NT Server 4.0, 2000 Server, Server 2003 operatsion tizimlarida ishlayotgan paytda mijoz qismi MS Windows 98 va boshqa operatsion tizimlardagi serverlar bilan kelishib ketadi. SQL Server uchun tavsija qilinadigan sayl tizimi - NTFS, lekin FAT tizimida ham ishlasa bo'ladi.

Tizim o'z tarkibida BBni tuzish uchun mablag'ga, BB bilan ishslash, berilganlarni bitta tiziñdan boshqa tizimga o'tkazish, berilganlarni qo'shimcha nusxasini olish, tranzaksiya tizimini rivojlantirish, berilganlar tizimini so'zma-so'z to'g'rilash, relatsion qism-tizimni tahlili, mijozlar murojaatlarini optimallash va bajarishga, BB obyektlarini boshqarish huquqini berish va xavfsizlik tizimiga ega. Tizim mijoz ilovalarni tuzish uchun mablag'ga ega emas. 36.1 - 36.4-jadvallarda tizimning ayrim maksimal imkoniyatlari ko'rsatilgan.



36.1-rasm. Tizim arxitekturasining asosiy tarkibiy qismi.

### Berilganlar bazasining maksimal parametrlari

36.1-jadval.

Nomi	O'chish
Berilganlar bazasi hajmi	1 048 516 Tb
Berilganlar bazasidagi obyektlar soni	2 147 483 647
Bitta kompyuterdag'i server nusxasining soni	16
Bitta server nusxasidagi BB soni	32767
Berilganlar bazasidagi fayllar soni	32767
Berilganlar bazasidagi jadvallar soni bazadagi obyektlar soni bilan chegaralangan	
Baza jadvalidagi maydonlar soni	1024
Berilganlar faylining hajmi	32 Tb
Identifikatorning uzunligi	128ta belgi
Joylangan protseduralar darajasi	32
Kiritilgan so'rovlar darajasi	32
Bitta baza jadvali uchun noklaster indekslar soni	249
Bitta indeksdagi maydonlar soni	16
Bitta indeksdagi bayt soni	800
Bitta so'rovdag'i jadvallar soni	256
Jadvalning bitta qatoridagi bayt soni	8060

**Berilganlarni simmetrik multiprotessor tartibida  
tizimning turli naqlarida ishlaydigan protsessorlarning maksimal soni  
(SMP)**

36.2-jadval.

<b>Operatsion tizim</b>	<b>MS SQL Server 2000 naqli</b>						
	<b>Enterprise Edition</b>	<b>Standard Edition</b>	<b>Personal Edition</b>	<b>Developer Edition</b>	<b>Desktop Engine</b>	<b>SQL Server CE</b>	<b>Yenterprise Evaluation Edition</b>
MS Windows 2000 DataCenter	32	4	2	32	3	-	32
Windows 2000 Advanced Server	8	4	2	8	2	-	8
Windows 2000 Server	4	4	2	4	2	-	4
Windows 2000 Professional	-	-	2	2	2	-	2
MS Windows NT®4.0 Server, Yenterprise Edition	8	8	2	8	2	-	8
Windows NT 4.0 Server	4	4	2	4	2	-	4
Windows NT 4.0 Workstation	-	-	2	2	2	-	2
MS Windows 98	-	-	1	Use Desktop Engine	1	-	-
<b>Windows CE</b>	-	-	-	-	-	<b>1</b>	-

**Fizik xotiraning maksimal soni (RAM), tizimida ishlaydigan turli naqlar (Gb)**

36.3-jadval.

Operatsion tizim	MS SQL Server 2000 naqli						
	Enterprise Edition	Standard Edition	Personal Edition	Developer Edition	Desktop Engine	SQL Server CE	Enterprise Evaluation Edition
Windows 2000 DataCenter	64	2	2	64	2	-	64
Windows 2000 Advanced Server	8	2	2	8	2	-	8
Windows 2000 Server	4	2	2	4	2	-	4
Windows 2000 Professional	-	-	2	2	2	-	2
Windows NT 4.0 Server, Yenterprise Ydition	3	2	2	3	2	-	3
Windows NT 4.0 Server	2	2	2	2	2	-	2
Windows NT 4.0 Workstation	-	-	2	2	2	-	2

**Tizimdagi berilganlarning turi**

Ma'lumotlar bazasini to'g'ri loyihaflash uchun bazadagi jadvallar maydoni uchun ishlatilsa bo'ladigan berilganlar turini bilish shart. 36.4-jadvalda MS SQL Server tizimidagi berilganlar turi guruhlarga bo'lingan shaklda ko'rsatilgan.

36.4-jadval.

Nomí	Berilganlar turining yozilishi
	Ikkilik berilganlar
<b>binary [(n)]</b>	maksimal uzunlik 8 000 bayt (n)
<b>varbinary [(n)]</b>	o'zgaruvchan uzunlik berilgan i, maksimal uzunlik 8 000 bayt (n)

<b>image</b>	maksimal uzunlik 2 147 483 647 bayt
<b>bit</b>	1 yoki 0 qiymatini qabul qiladigan berilganlar turi
<b>Ishorali berilganlar</b>	
<b>char [(n)]</b>	maksimal uzunlik 8 000 simvol (n)
<b>varchar [(n)]</b>	o'zgaruvchan uzunlik berilgan maksimal soni 8 000 simvol (n)
<b>text</b>	maksimal uzunlik 1 073 741 823 simvol
<b>Unicode kodirovkasidagi ishorali berilganlar</b>	
<b>nchar (n)</b>	maksimal uzunlik 4 000 simvol (n)
<b>nvarchar (n)</b>	o'zgaruvchan uzunlik Unicode kodirovkasidagi maksimal uzunlik 4 000 simvol (n)
<b>ntext</b>	maksimal uzunlik 1 073 741 823 simvol
<b>Sonli butun berilganlar</b>	
<b>bigint</b>	oralig'i -922 337 203 685 4775808dan 922 337 203 685 4775807gacha
<b>int</b>	oralig'i -2 147 483 648dan 2 147 483 647gacha
<b>smallint</b>	oralig'i - 32 768dan 32 767gacha
<b>tinyint</b>	oralig'i Odan 255gacha
<b>Kasr qismli sonli berilganlar</b>	
<b>decimal[(p[, s])]</b>	oralig'i $-10^{38}$ -1dan $10^{38}$ -1gacha chegaralangan belgilari soni berilganda (p – jami va s –kasr qismi), maksimal jami belgilari soni 38
<b>numeric</b>	decimal ning o'zi
<b>float [(n)]</b>	oralig'i $+2.29 \cdot 10^{-308}$ dan $+1.79 \cdot 10^{308}$ gacha
<b>real</b>	$+1.18 \cdot 10^{-38}$ dan $+3.40 \cdot 10^{38}$ gacha oraliqda 7-xonagacha aniq sonlar
<b>Sana va vaqt</b>	
<b>datetime</b>	oralig'i 1.01.1753dan 31.12.9999gacha 3.33 ms aniqligida
<b>smalldatetime</b>	oralig'i 1.01.1900dan 6.06.2079gacha 1 min. aniqligida
<b>Pul turi</b>	
<b>money</b>	oralig'i -7 203 685 477.5808dan +922 337 203 685 477.5807gacha
<b>smallmoney</b>	oralig'i -214 748.3648dan +214 748.3647gacha
<b>Maxsus turdag'i berilganlar</b>	
<b>timestamp</b>	baza uchun noyob mazmunga ega avtomatik oshadi-gan hisoblagich (binary(8) yoki varbinary(8) turlari)
<b>uniqueidentifier</b>	noyob identifikator raqamga ega tur (GUID), 16-bitli saqlanadigan ikkilik qator

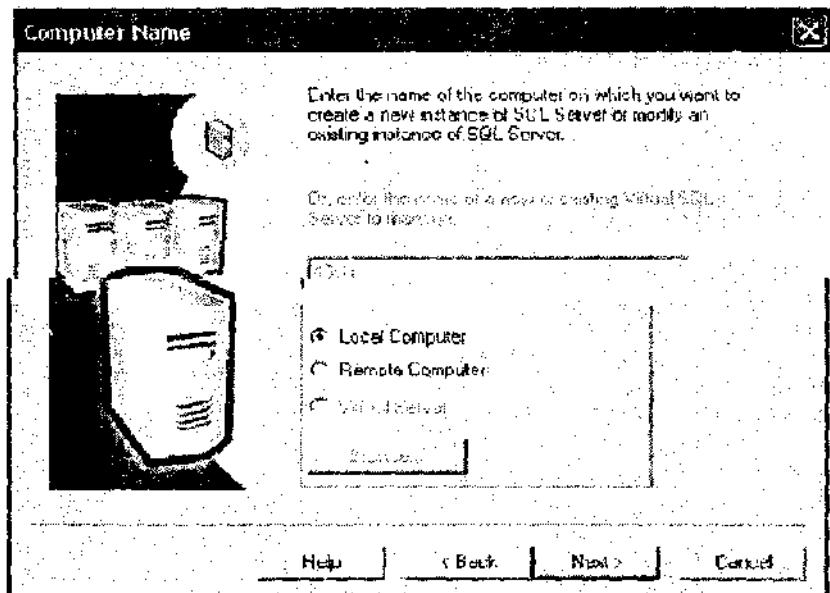
sql_variant	har xil turlarning ma'nosini saqlaydigan tur text, ntext, timestamp va sql_variant lardan tashqari.
sysname	nvarchar sinonim tur, BBdag'i obyektlar nomiga yo'naltirish uchun ishlafiladi

### Tizimning o'rnatilishi

MS SQL Server tizimning o'rnatilishi AUTORUN.EXE faylini distributiv diskdan ishga tushirish yo'li bilan bajariladi (o'z navbatida u \Sql\x86\ setup\setupsql.exe programmasini ishga tushiradi). Shu vaqtida o'rnatish Ustasi ishga tushib, qadamma-qadam tizimning qaysi parametrini o'rnatishni tanlashda tavsiya berib turadi.

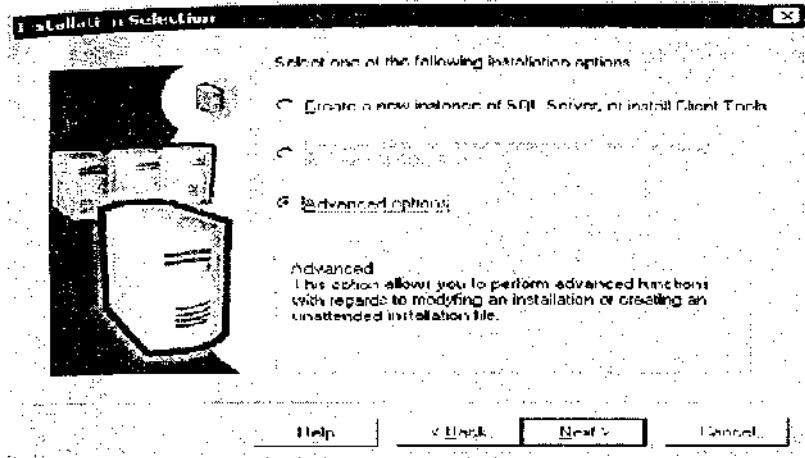
Birinchi qadam (36.2-rasm) – o'rnatish uchun kompyuter tanlash:

- lokal kompyuterda SQL Server o'rnatish;
- masofadagi kompyuterda o'rnatish;
- virtual server yaratish yoki sozlash.



36.2-rasm. Tizimni o'rnatish «Usta»sining 1-qadami.

Keyingi qadam – installatsiya turini tanlash (36.3-rasm.)



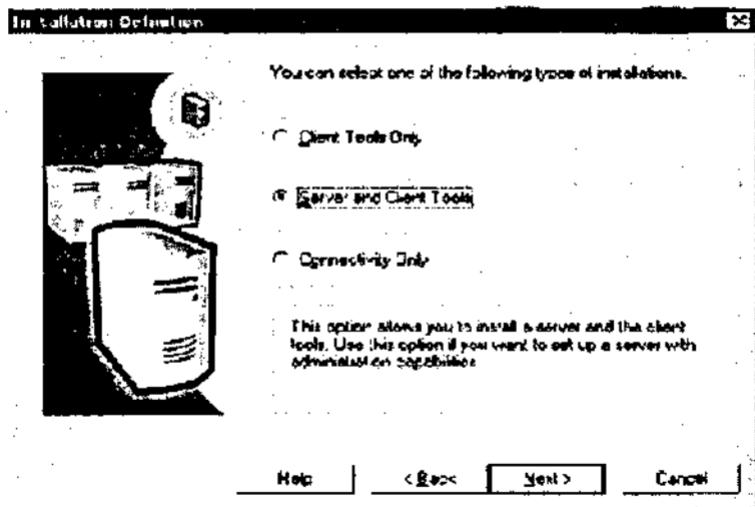
36.3-rasm. «Server» installatsiyaning turini tanlash.

Mumkin bo‘lgan variantlar:

- yangi SQL Server installini tuzish;
- mavjud installning tarkibiy qismini yangilash yoki o‘chirish;
- virtual serverni sozlash;
- kechroq bajarilishi mumkin bo‘lgan SQL Server tarkibiy qismini avtomatik o‘rnatadigan berilganli sayni tuzish.

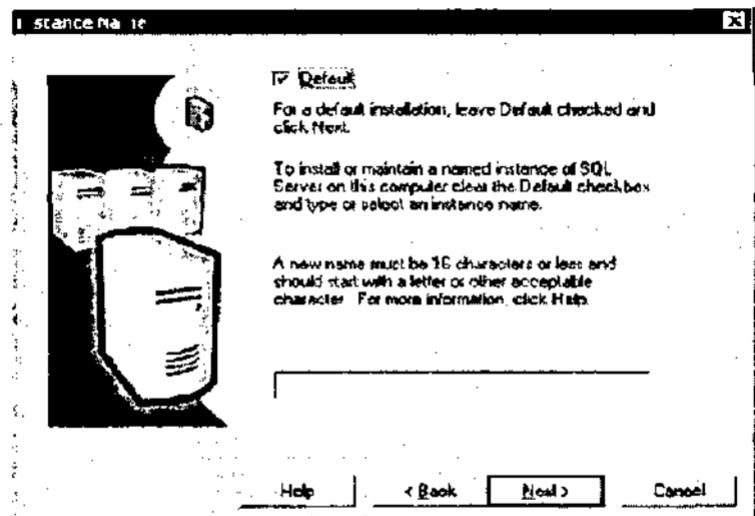
So‘ngro programma foydalanuvchining va taskikilotning nomini kirishni so‘raydi, shuningdek, litsenziyalı shartnomani qabul qilishni taklif qiladi. Shundan so‘ng Installation Definition oynasi ochiladi. Unda programma ta’mintonining o‘rnativini uchta varianti mavjud (36.4-rasm):

- Client Tools Only – tarmoq kutubxonalarining va SQL Server administrator vositalarini o‘rnativi. Bu operatsiya o‘chgan serverni boshqarish uchun ishlataladigan kompyuterlar uchun tanlanadi;
- Server and Client Tools - SQL Server to‘la o‘rnativ. Bu opsiya yashirin tarzda tanlanadi;
- Connectivity Only – tarmoq kutubxonalarining va tarkiblarining berilganlarga kirish huquqini o‘rnatish (MS Data Access Components, MDAC), lekin server administratori vositasi uchun emas. Bu opsiya SQL Server tizimi bilan birgalikda harakat qilishi kerak bo‘lgan kompyuterlar uchun o‘rnataladi, ammo SQL Server i administrator serveri uchun foydalananmaydi.



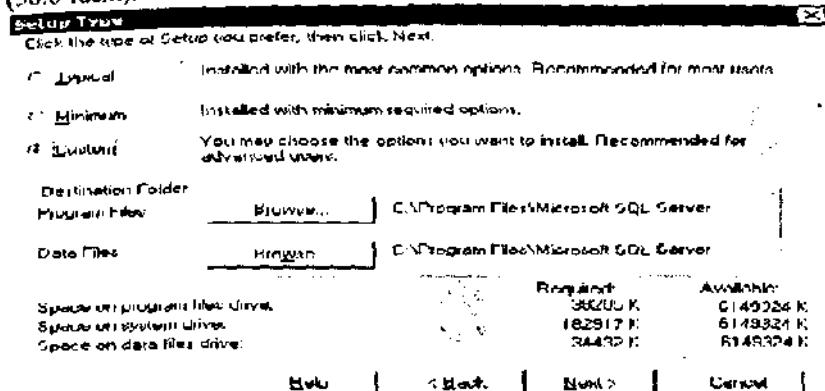
36.4-rasm. Programma vositalarini o'matish variantlari tanlovi.

So'ngra o'matuvini indamay yoki nomi yozilgan server nomini kiritish kerak (36.5-rasm).



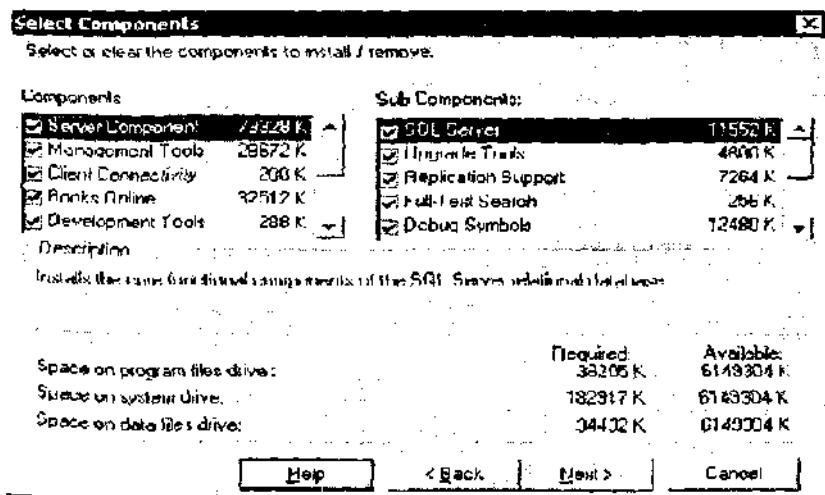
36.5-rasm. Server nomini kiritish.

So'ngra o'matuv varianti tanlanadi: o'ziga xos bo'lgan, minimal yoki foydalanuvchi o'matuv (o'matish uchun tarkiblarni tanlash imkoniyati bilan) va o'matuv uchun kompyuter diskidagi katalogni tanlash mumkin (36.6-rasm).



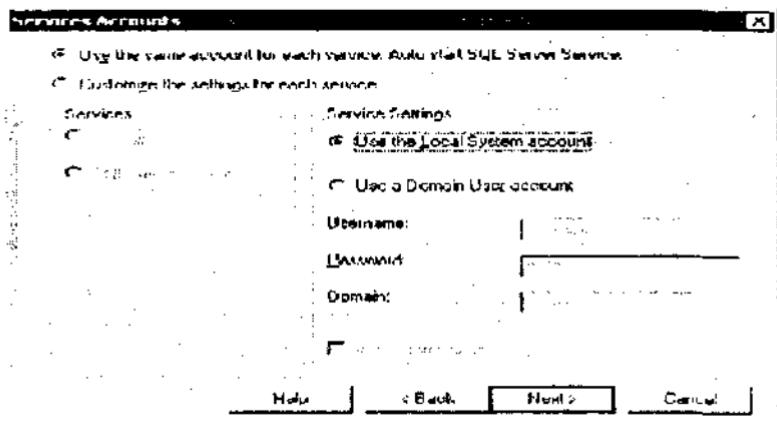
36.6-rasm. Serverni o'matish variantini tanlash.

Foydalanuvchi o'matish variantini tanlaganda, tizimning tarkiblari ro'yxati oynasi va har bir tarkibning ichidagi bo'laklar ko'rsatiladi (36.7-rasm).



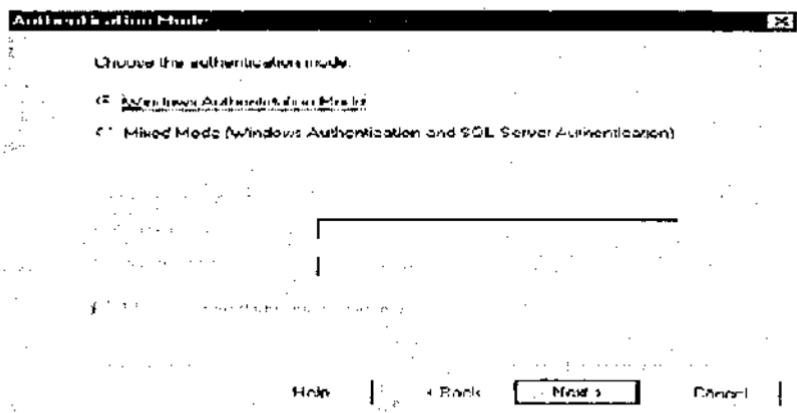
36.7-rasm. O'matuv uchun komponentlar tanlovi.

So'ngra SQL Server xizmatini ishga solish uchun qayd qilingan yozuvni kiritish kerak, bu lokal foydalanuvchi yoki tarmoq domenida SQL Server i bilan ishlash uchun qayd qilingan foydalanuvchi bo'lishi mumkin (36.8-rasm).



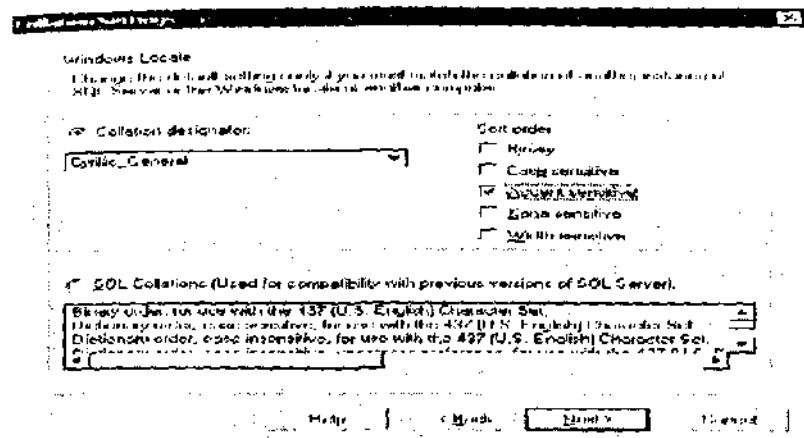
36.8-rasm. Server xizmatini ishga solish uchun qayd qilingan yozuvni kiritish.

So'ngra Windows yoki SQL Server autentifikatsiya qilish tizimi tanlanadi (36.9-rasm).



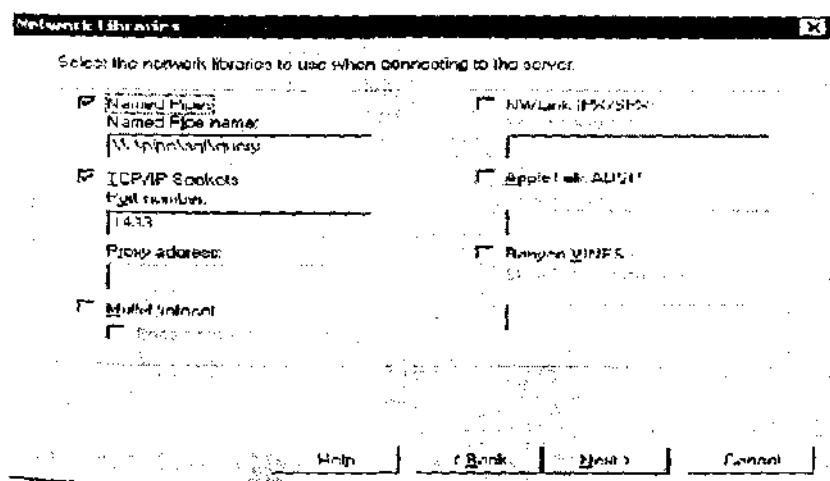
36.9-rasm. Autentifikatsiya usulini kiritish.

Keyingi oyna – kodli varaqni va berilganlarni saralash parametrlarini kiritish. Bu erda kompyuterda o'tmatilgan yoki SQL Serveri tizimi uchun alohida berilgan parametrlarini kiritish mumkin (36.10-rasm).



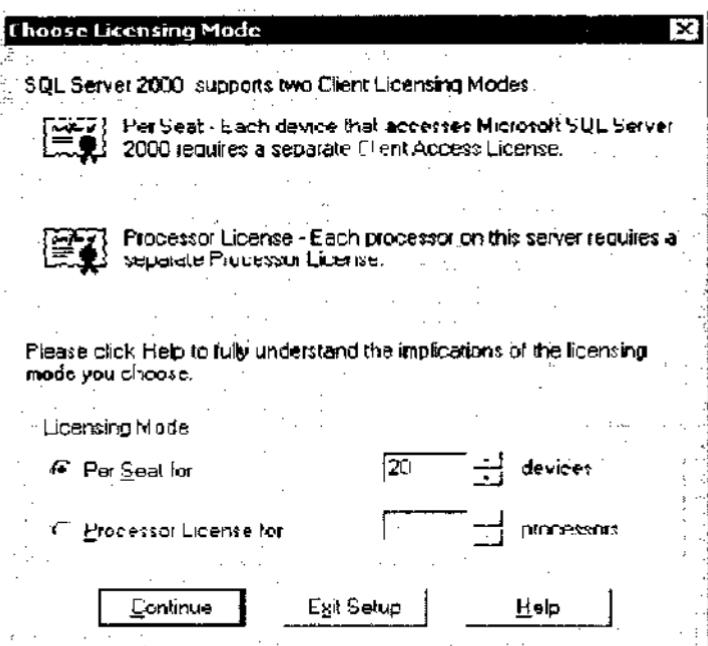
36.10-rasm. Kodli varaqni kiritish va berilganlar parametrlarini saralash.

Keyingi oyna – server tomonidan ishlatalayotgan tarmoq kutubxonalarini kiritish (36.11-rasm).



36.11-rasm.

So'ngra mijoz litsenziyasi turini va sonini kiritish so'traladi (36.12-rasm).

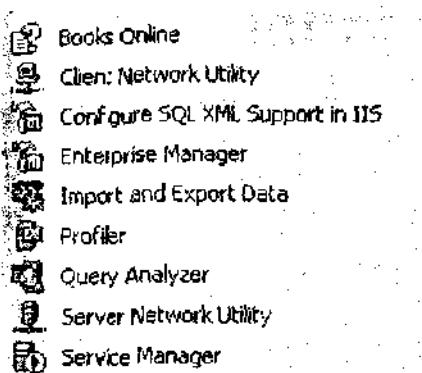


36.12-rasm. Litsenziyalash turini tanlash.

Shundan so'ng, fayllarning nusxasi ko'chirish boshlanadi va SQL Server tizimi programmalari bilan ishlash uchun Windows menyusida programma guruhini tuzish bilan tizimni o'rnatish jarayoni tugaydi.

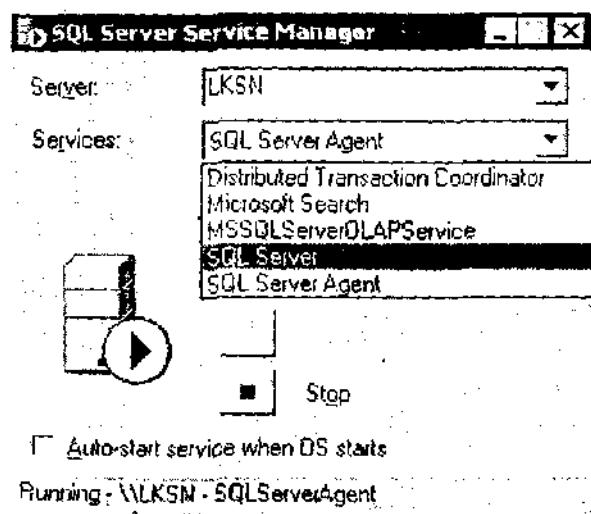
SQL Server 2000 tizimining muhim yangiliklaridan biri – bitta kompyuterda bir nechta SQL Server nusxalarini o'rnatish mumkinligi. Birinchi bo'lib o'rnatiladigan SQL Server nusxasi *standart yoki oshkorinas ishlataladigan* deyiladi; o'sha kompyuterga o'rnatilgan qolgan hamma nusxalar *nomlangan* deyiladi. Har bir nomlangan SQL Server nusxasi uchun o'z BB to'plami va foydalanuvchisi aniqlangan bo'lishi mumkin. Agar har xil kompyuterlarda bir xil SQL Server nusxasi o'rnatilsa, ulami bitta virtual serverga birlashtirish mumkin.

MS SQL Server programmalari guruhidagi installatsiyadan so'ng Developer Edition naqli uchun 36.13-rasmida ko'rsatilgan bandlar mavjud.



36.13-rasm. Tizimni o'matgandan keyingi, Windows menyusidagi programmalar guruhi.

SQL Server tizimining asosiy tarkibiy qismi Windows xizmati (Services) singari amalga oshiriladi. **SQL Server Service Manager** programmasi orqali tizimning tarkibiy qismlari bilan bog'liq xizmatni ishga solish va to'xtatishni boshqarish mumkin (36.14-rasm). Bu programmaning yorlig'i Windows ning masalalar chizmasidagi bildirish hududida paydo bo'ladi va SQL ServerAgent asosiy xizmatining ishga tushishi indikatsiyasini beradi.



36.14-rasm. Service Manager programmasi.

## Xulosa

*Relation BBBT MS SQL Server muhiti «mijoz-server» arxitekturasida BB bilan ishlaydigan eng qurvatli tizimlardan biri. Tizimning afzalliklari shundaki, server faqat MS Windows NT - NT Server 4.0, 2000 Server, Server 2003 operatsion tizimlarida ishlayotgan paytda mijoz qismi MS Windows 98 va boshqa operatsion tizimlardagi serverlar bilan kelishib ketadi.*

*MS SQL Server muhiti BB bilan ishlash, berilganlarni bitta tizimdan boshqa tizimga o'tkazish, berilganlarni qo'shimcha nusxasini olish, tranzaksiya tizimini rivojlanadirish, berilganlar tizimini so'zma-so'z to'g'rilash, relatsion qism-tizimni tahlili, mijozlar murojaatlarini optimallash va bajarishga, BB obyektlarini boshqarish huquqini berish va xayfsizlik tizimiga ega.*

*MS SQL Server tizimini o'rnatishdagi boshlang'ich ko'nikma hosil qilindi.*

### Nazorat uchun savollar va amaliyat uchun topshiriqlar

1. MS SQL Server BBB tizimining tavsiflab bering.
2. MS SQL Server BBB tizimining tarkibiy qismidagi programma mahsulotlarini tavsiflab bering.
3. MS SQL Server tizimidagi berilganlar bazasining maksimal hajmi qancha bo'lishi mumkin?
4. MS SQL Server tizimidagi maksimal obyektlarning soni nechta?
5. MS SQL Server tizimidagi bitta kompyuterdag'i server nusxasining soni qancha?
6. MS SQL Server tizimidagi bitta server nusxasidagi berilganlar bazasining maksimal soni qancha?
7. Berilganlar bazasidagi fayllarning maksimal soni qancha?
8. Baza jadvalidagi maydonlarning maksimal soni qancha?
9. Joylangan protseduralar darajasining maksimal soni qancha?
10. Bitta indeksdagi maksimal maydonlar soni qancha?
11. Bitta indeksdagi maksimal baytlar soni qancha?
12. Bitta so'rovdag'i maksimal jadvallar soni qancha?
13. Berilganlarning turlari nechta?
14. MS SQL Server ning standart nusxasi bilan nomlangan nusxasining farqi nimada?
15. Virtual server, deganda nimani tushunasiz?
16. MS SQL Server nusxalarini bitta kompyuterga o'rnatish va bajargan ishingizni izohlang.

## **37-bo'b. MS SQL SERVER TIZIMI**

**Tayanch iboralar:** master, model, msdb, tempdb, Northwind, Pubs, Security, Logins, BUILTIN\Administrators, ИМЯ ДОМЕНА/Учетная запись службы SQL Server, Database Creators, Transact-SQL, Databases, New, Tools, Diagrams, Tables, Views, Stored Procedures, Users, Roles, Defaults, User Defined Data Types, User Defined Functions, Full-Text Catalog, Column Name, Data Type, Length, Allow Nulls, Design Table, Manage Indexes/Keys, Properties, Diagram, Grid, SQL, Results, Add table.

SQL Server tizimi tarkibiga beshta xizmat kiradi, ular uchun Windows ni ishga tushirganda avtomatik yoki qo'lida ishga solish mumkin (37.1-jadval).

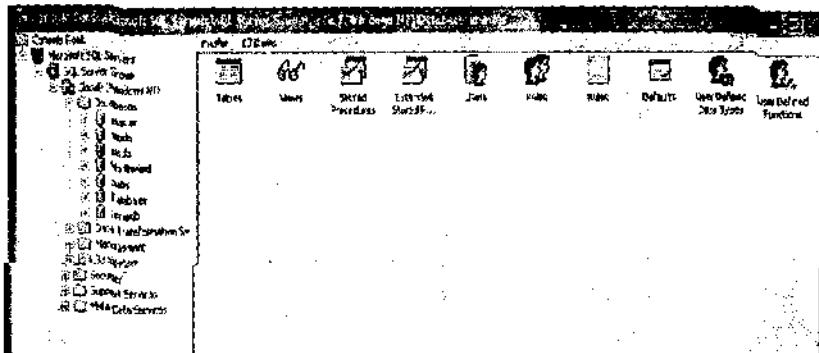
### **MS SQL Server 2000 tizimi xizmati**

**37.1-jadval.**

<b>Xizmat</b>	<b>Mu'jal</b>
MS SQL Server	SQL Server ning asosiy yadrosi, BB serverining vazifalarini amalga oshiradi
SQL Server Agent	Administrativ vazifalarini bajaradi, vazifalarini rejaga asosan bajarilishini va operatorlarni qo'llashga javob beradi. SQL Server bu programmasiz ishlay oladi, ammoye bunday holda uning imkoniyatlari chegaralaniladi
MS DTC (MS Distributed Transaction Coordinator)	Faqatqina tizimda taqsimlangan tranzaksiyalar bajarilayotgan vaziyatda zarur bo'ladi. Agar zarurat bo'limasa, o'rnatish shart emas
MS Search (MS Search)	To'liq matnli qidiruvni qo'llaydi. U kataloglarni va to'liq matnli indekslarni yuzaga keltiradi, shuningdek, qidiruvni amalga oshiradi. Agar zarurat bo'limasa, o'rnatish shart emas
MS SQL Server OLAP Service	SQL Server - MS SQL Server 2000 Analysis Services ga qoshimcha tarkib taqdim etadigan maxsus xizmat (OLAP - berilganlarni operativ analitik tayyorlash uchun server)

*MS SQL Server 2000 tizimi tarkibiga administratsiya va BB bilan ishlash bo'yicha katta imkoniyatlarga ega Yenterprise Manager (37.1-rasm) programmasi kiradi. Agar ushbu programmani ishga tushirgandan keyin serverlar ro'yxati bo'sh bo'lsa, unda kompyuterda o'rnatilgan server nusxasini ro'yxatdan o'tkazish kerak. Windows NT tizimini autentifikatsiyasini ishlatganda, ularish uchun parol so'rovi kelmaydi. Agar SQL Server tizimi autentifikatsiyasini ishlatalayotgan ho'lsa, Yenterprise Manager avtomatik ularuvni amalgal oshirish kerakmi yoki siz har bir ularishda, ism va parolning so'ralishini xohlaysizmi, deb so'raydi.*

*Servernij ro'yxatdan o'tkazgandan so'ng, unga ularish uchun server nomining chap tarafidan «+» belgisini bosish kifoya. Agar ularish omadli o'tsa, server piktogrammasidagi qizil belgi yashilga o'zgaradi.*



37.1-rasm. Enterprise Manager programmasi.

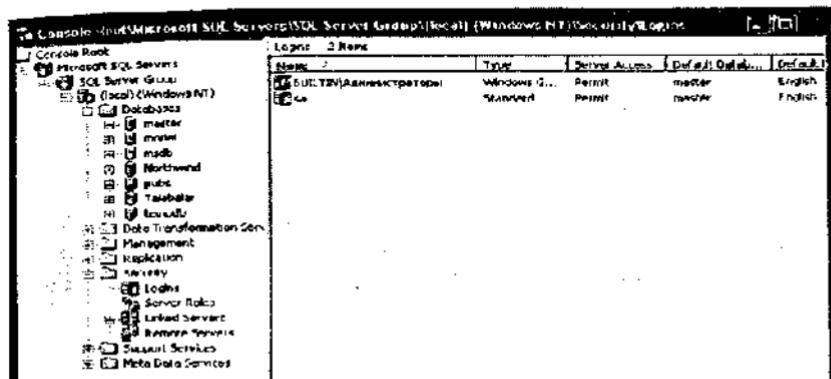
Enterprise Manager da serverning berilganlar bazasi ro'yxatini ochgach biz tizimning va oshkormas ravishda o'matilgan o'quv berilganlar bazasini ko'ramiz.

Bu keyingi oltita BB:

- **master** – serverni boshqarish uchun xizmat ko'rsatadi;
- **model** – foydalanuvchilar bazasi shabloni;
- **msdb** – vazifa va ro'yxatlar jadvalini hajarish jurnali;
- **tempdb** – vaqtinchalik jadvallar va obyektlarni saqlantishi;
- **Northwind** – foydalanuvchi bazasi misoli;
- **Pubs** – foydalanuvchi bazasi misoli.

Northwind va Pubs BB – bu o'matilishi shart bo'lmagan SQL Server o'quv BB. Mactep, model, msdb, tempdb tizimning berilganlar bazalari hisoblanishi va SQL Server ishlashi uchun kerak. Enterprise Manager programmasini o'matish paytida yaratilgan tizimni qayd qiluv yozuvini

ko'rib chiqish kerak. Buning uchun **Security** papkasida joylashgan Logins papkasini ochish kerak (37.2-rasm).



37.2-rasm. Enterprise Manager programmasidagi foydalanuvchilarni qayd qiluv yozuvi.

Serverda uchta qayd qiluvchi yozuv aniqlangan bo'lishi kerak:

- **BUILTIN\Administrators;**
- **ИМЯ ДОМЕНА/Учетная запись службы SQL Server** (agar o'matish jarayonida, *Use a domain user account*, tanlangan bo'lsa);
- **sa.**

Bu qayd qiluv yozuvlari SQL Server o'matilayotgan paytda paydo bo'ladi va katta ahamiyatga ega.

Faqatgina, Windows NT tizimi autentifikatsiyasini ishlatganda, SQL Server ni Windows NT Server yoki Windows NT Server Enterprise Edition o'matilganda, **BUILTIN\Administrators** guruhi, paydo bo'ladi. Unda, serverga administrativ kirish huquqi berilgan, Windows NT Administrators guruhining barcha a'zolari ko'rsatilgan.

Sa qayd qiluv yozuvi serverni boshqarish uchun mo'ljallangan. Sa SQL Server ni barcha o'matish jarayonida paydo bo'ladi, vaholanki Sa qayd yozuvisz serverga bog'lanish mumkin emas. Oshkormas ravishda, bu yozuv parolga ega emas. Uning uchun darhol parol berish kerak va muntazam ravishda parolni o'zgartirib turish tavsiya qilinadi. Bu qayd qiluv yozuvi uchun, SQL Server va uning barcha obyektlariga kirish huquqi bor va barcha BBda u oshkormas holda dbo laqabini oladi.

Windows 9.x ga SQL Server ni Desktop naqlini o'matganda, faqatgina sa qayd qiluv yozuvi yaratiladi.

## Berilganlar bazasini yaratish

Yangi BBni yaratish uchun foydalanuvchi, administrator huquqiga yoki **Database Creators** ahamiyatiga ega bo'lishi kerak.

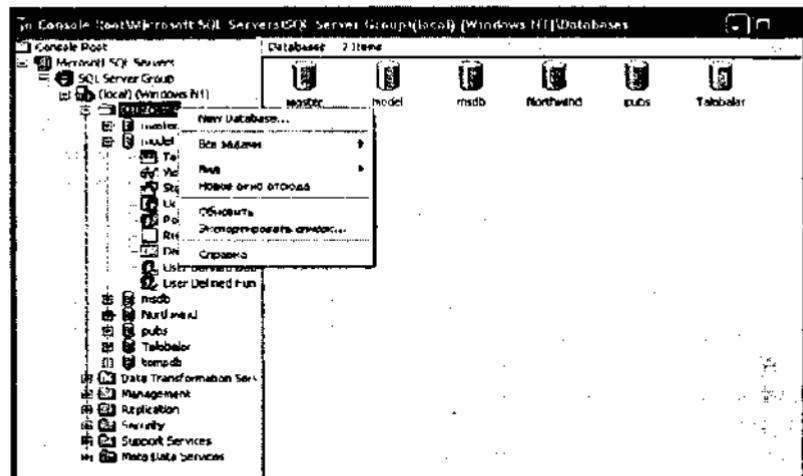
Boshqa operatsiyalar singari, BBni yaratish **SQL Server Enterprise Manager** programmasini ishlatgan holda bajarish oson.

Shuningdek, **SQL Query Analyzer** programmasini ishga tushirib, **Transact-SQL** tilida yozilgan, BBni yaratish programmasidan ham foydalananish mumkin. BBni va jadva'llarini yaratishni, BBni modellashtirish vositalarini ishlatgan holda ham bajarish mumkin, masalan 20- bobda yoritilgan **Toad Data Modeler** muhitidan foydalangan holda.

**Enterprise Manager** programmasi orqali, BBni yaratish quyidagicha bajariladi:

- shu programmaning oynasidagi (37.3-rasmiga qarang) *Databases* papkasidagi oynalar menyusidan, kontekst menyudan yoki asboblar chizimidan New komandasini tanlagan holda;
- BB server konsoli oynasining *Tools* menyusi punkti orqali chaqiriladigan, BB yaratuvchi masteridan foydalangan holda.

*Ixtiyoriy holata - BB, sayllar va tranzaksiyalar jurnali nomlarini va ularning boshlang'ich hajmini hamda ushbu sayllar hajmini avtomatik ortish qiymatini berish kerak.*



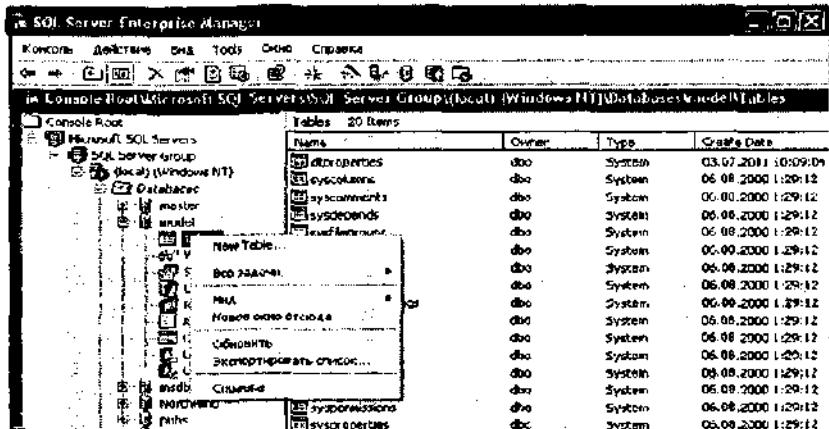
37.3-rasm. Yangi BBni yaratish.

Natijada **model** bazasi qolipida yangi baza tuziladi. Unda shu qolipning barcha obyektlar guruhi qatnashadi:

- **Diagram** – bazaning jadvallari o'rtaqidagi aloqani tasvirlaydigan chizmalar;
  - **Tables** – bazalarning jadvallari va ularning indekslari haqidagi ma'lumotlar saqlanadigan, jadvallar papkasi;
  - **Views** – bir necha jadvallardan bitta virtual jadvalga birlashtilgan, berilganlar to'plami tavsiflaridan iborat - tavsiflar papkasi;
  - **Stored Procedures** – *Transact-SQL* tilidagi protseduralar ro'yxati saqlanayotgan protseduralar;
  - **Users** – baza egasi (owner)ni va bazaga kirish huquqiga ega foydalanuvchilar haqidagi ma'lumotlar;
  - **Roles** – foydalanuvchilar guruhlari turlarining tavsifi;
  - **Defaults** – bazadagi oshkormas qiymatlar tavsifi va ularning jadvallar ustunlari bilan aloqalari;
  - **User Defined Data Types** – foydalanuvchi ishlatgan berilganlar va ularning turlari tavsifi;
  - **User Defined Functions** – foydalanuvchi funksiyalarining tavsifi;
  - **Full-Text Catalog** – to'liq matnli indekslarni saqlovchi papka.

#### **Berilganlar bazasining jadvalini yaratish**

*Enterprise Manager* programmasidagi Table papkasida New komandasini tanlash (37.4-rasm).



37.4-rasm. BBda yangi jadvalni yaratish.

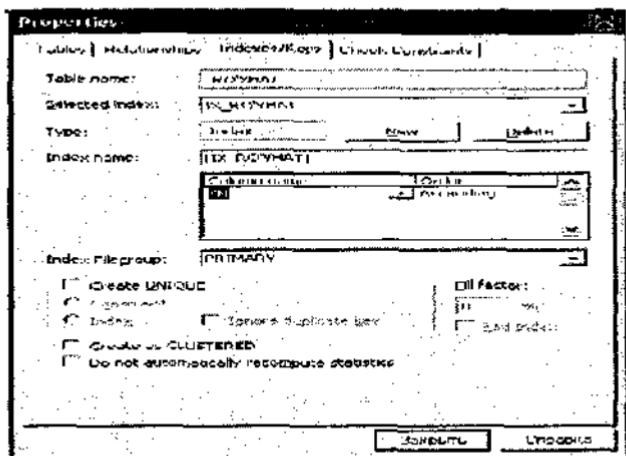
Paydo bo'lgan *New Table in <baza nomi> on <SQL server nomi>* nomli oynada jadval tuzilishini tavsiflash, ya'ni ustunlar nomi - **Column Name**, ustundagi berilganlar turi - **Data Type**, berilganlar uzumligi - **Length** va berilgan bilan to'limgan maydonni mavjudligi - **Allow Nulls** xususidagi ma'lumotlar (37.5-rasm).

37.5-rasm. Yangi jadval tuzilishi tavsifi.

**Saqlash (Soxranit)** komandasidan so'ng, jadval nomini kiritish kerak va u bazalar jadvali ro'yxatida paydo bo'ladı.

Uning tuzilishini o'zgartirish uchun keyinchalik, **Design Table** komandasini tanlash mumkin, undan so'ng yana jadval tuzilishi ta'rifni oynasi ochiladi.

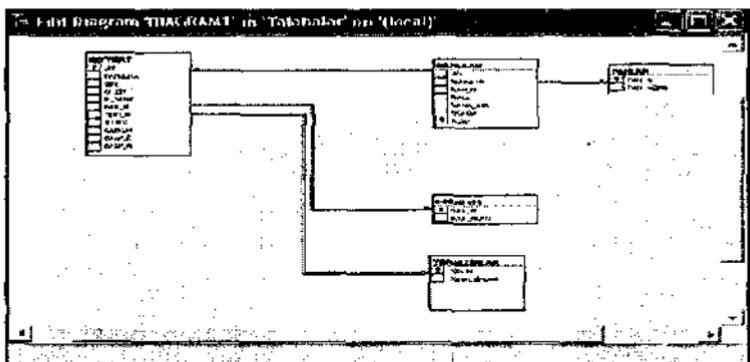
**Design Table** oynasida indekslarni yaratish uchun asboblar orasidan **Manage Indexes/Keys** tugmasini tanlash lozim, undan so'ng jadval xususiyatlari oynasi - **Properties** ochiladi, uning uchinchi sahifasida indekslarni tasvirlash kerak, ular noyob bo'lishi mumkin yoki yo'q va klasterli (diskdagi jadvalning fizik tartibi indeksga mos keladi) bo'lishi mumkin yoki yo'q (37.6-rasm).



37.6-rasm. Jadval indekslari tavsifi.

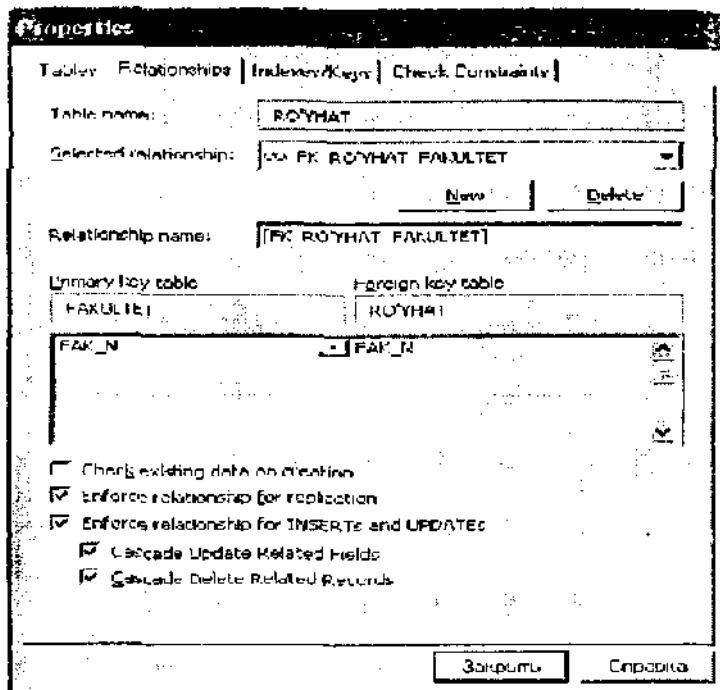
Har bir jadvalni ko‘rinishi o‘zgargandan so‘ng, berilganlarni yangilab turish imkoniyati bo‘lishi uchun o‘z noyob indeksi bo‘lishi lozim.

Bazaning barcha jadvallarini yaratgandan so‘ng, o‘sha xususiyatlar oy-nasida jadvallar o‘rtasida aloqani yaratish lozim (boshqa jadvallar bilan aloqaga ega jadvallar uchun **Properties** oynasida ikkinchi sahifasida), so‘ngra bazaning **Diagrams** papkasida jadvallar o‘rtasidagi aloqalarning grafik tasavvurini yaratish mumkin (37.7-rasm).



37.7-rasm. BB chizmasi.

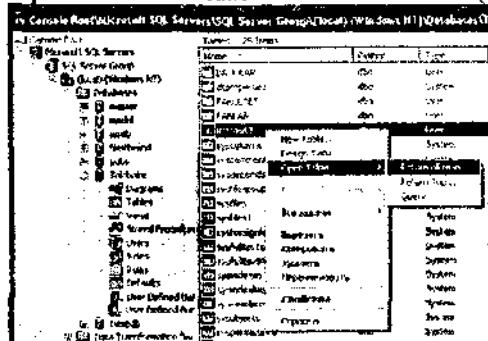
Aloqalar uchun, havolalar butunligi shartiga rioya etish talablarini, berish mumkin (37.8-rasmga qarang). Shu shartlarni **Design Table** oynasida ishlayotganda ham berish mumkin (37.6-rasm).



37.8-rasm. Jadval uchun aloqalar xususiyati oynasi.

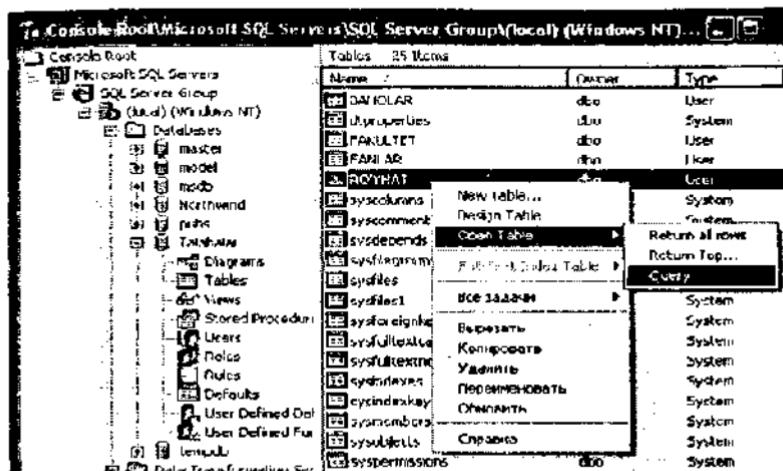
**Enterprise Manager programmasida BB xabari bilan ishlash**

Tanlangan jadvalga yangi yozuvlarni qo'shish, xabarni to'g'rilash va o'chirish uchun ***Open table*** komandasini ishlatalish mumkin (37.9-rasm).



### **37.9-rasm. Ishlash tartibini tanlash.**

Tanlangan jadvalning so'rov oynasiga o'tish 37.10-rasmida keltirilgan.



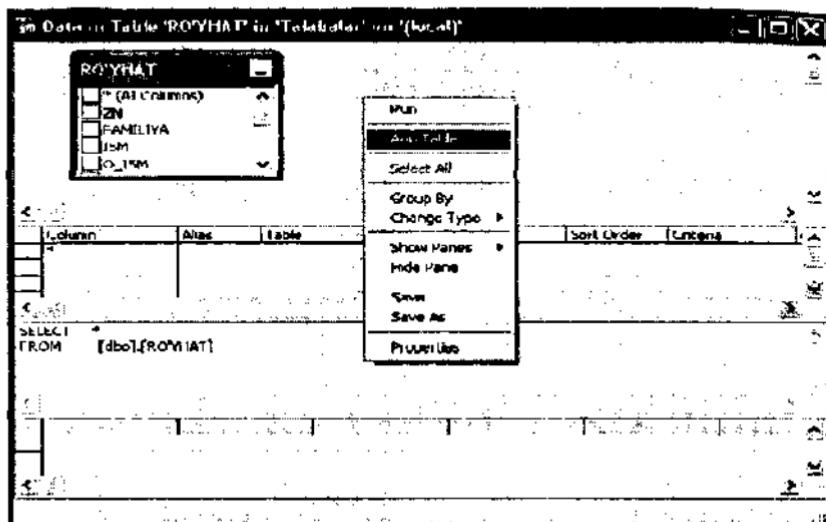
37.10-rasm.

Ochilgan so'rov oynasi to'ri qismdan iborat bo'lib, ixtiyorisini zaruriy paytada ochib – yopish mumkin. Oynaning qismlarining nomlari quyidagiicha:

- *Diagram;*
- *Grid;*
- *SQL;*
- *Results.*

Oyna qismlarining hammasi bir xil xossaga, ya'ni so'rov oynasining xossasiga ega. *Diagram* jadvallar chizmasini ko'rish, ulardan zaruriy ustunlarni tanlash imkoniyatini beradi. *Grid* – tanlangan ustunni natija qism oynasida nomini o'zgartirish yoki o'zgartirmasdan chiqarish, qaysi jadvalga tegishliliqi, natija qism oynasida ustunni ko'rsatish yoki ko'rsatmaslik, sarkillashni qanday bajarish, so'rov shartlarini berish kabi imkoniyatlarni beradi. Grid ni QBE tilining to'ri sifatida qarasak bo'ladi. *SQL* – tanlangan ustunlar bo'yicha yaratilgan ixtiyoriy so'rovlarni namoyish etishga mo'ljalangan. *Results* – so'rov natijalarini darhol kuzatish imkonini beradi.

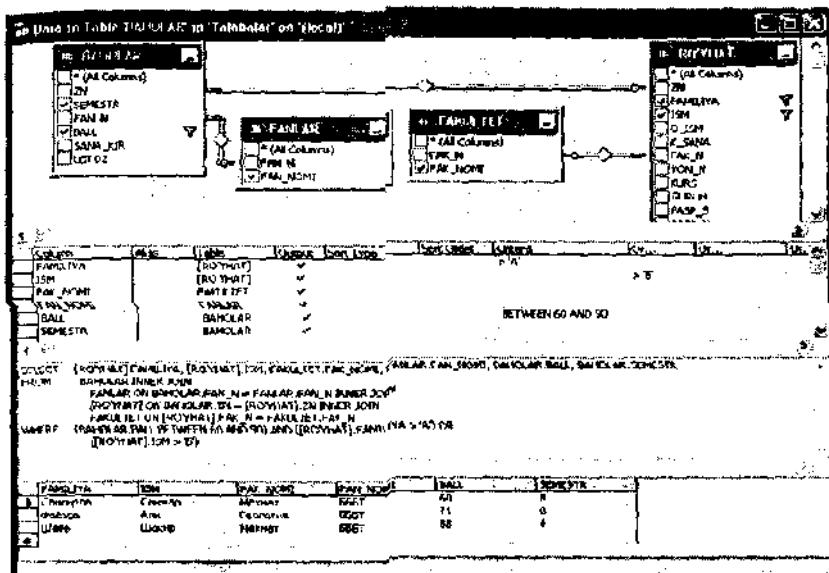
*Diagram* qism oynasi orqali sichqonchaning o'ng tugmasi bilan 37.11-rasmidagi kabi, *Add table* zaruriy jadvallarni oynaga qo'shish mumkin (37.12 – 37.13-rasmrlarga qarang).



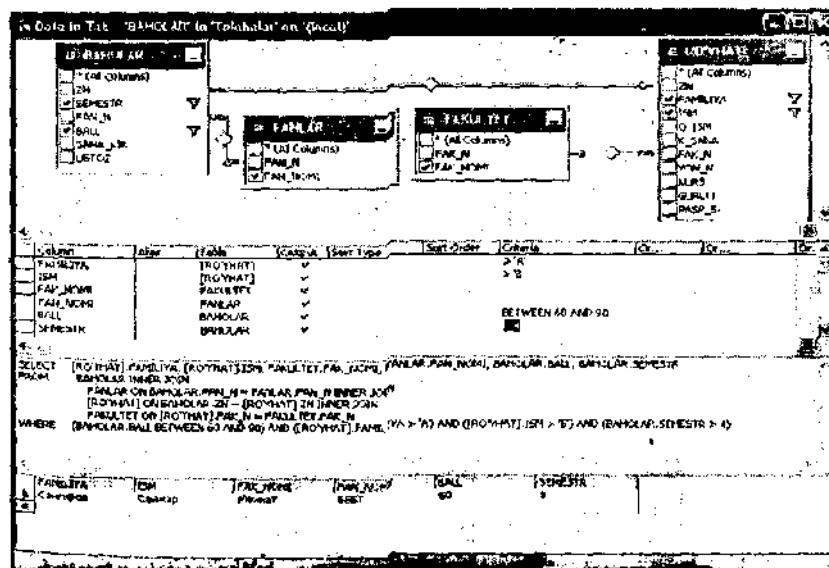
37.11-rasm.

Student ID	FNAME	LNAME	FAK_N	FAK_NAME	YONALISHLAR	YONALISHLAR_N
E005227	CAMILA	CHEN	3	FAKULTET	YONALISHLAR	4

37.12-rasm.



37.13-rasm.



#### 37-14. tasse - Setayalar konstruktör oynası

37.12 – 37.14-rasmida so'rovlar konstruktori oynasida, sodda va mukarrak so'rovlni tashkil qilishning uchta usuli keltirilgan. Tashkillashtirilgan so'rovlar avtomatik tarzda SQL tilida yoziladi va bu jarayon ko'rinishi turadi. Bu usul SQL tilini o'reganishda ham juda katta foyda beradi.

## Xulosa

*MS SQL Server dagi xizmat turlarini ta'minlovchi asosiy programmalar - SQL Server Agent, MS DTC, MS Search va MS SQL Server OLAP Service.*

*Berilganlar bazasini yaratishda, Talabalar o'quv loyihamizdan foydalangan holda, undagi jadvallarni yaratdik, ularga berilganlarni kiridik, jadval orasidagi aloqalarni va berilgan butunligini ta'mindik.*

*So'rovlar oynasi orqali bir nechta so'rovlni yaratdik va natijalarni kuzatdik.*

## Nazorat uchun savollar va amaliyat uchun topshiriqlar

1. MS SQL Server BBB tizimining qanday xizmat turlari mavjud?
2. Yenterprise Manager programmasining vazifasi nimadan iborat?
3. Mactep programmasining vazifasi nimadan iborat?
4. Model programmasining vazifasi nimadan iborat?
5. Msdb programmasining vazifasi nimadan iborat?
6. Tempdb programmasining vazifasi nimadan iborat?
7. Toad Data Modeler muhitidan foydalangan holda, MS SQL Server uchun 20- bobda keltirilgan skriptni ularshni amalga oshiring.
8. Yenterprise Manager programmasidan foydalangan holda, Talabalar o'quv loyihasining jadvallarini yaratting.
9. Model bazasi qolipidagi Diagrams va Tables obyektlarining funksiyaliga izoh bering.
10. Model bazasi qolipidagi Views va Stored Procedures obyektlarining funksiyalariga izoh bering.
11. Model bazasi qolipidagi Users va Roles obyektlarining funksiyalariga izoh bering.
12. Model bazasi qolipidagi Defaults va User Defined Data Types obyektlarining funksiyalariga izoh bering.
13. Model bazasi qolipidagi User Defined Functions va Full-Text Catalog obyektlarining funksiyalariga izoh bering.
14. Talabalar o'quv loyihasining jadvallari orasidagi aloqalarni yaratting.

15. Aloqalar uchun havola butunligi shartining talablarini qondiring va ularni izohlang.
16. So'rov oynasi qaysi qismlardan iborat?
17. So'rov oynasi qismlaridan Diagram va Grid programmalarini izohlang.
18. So'rov oynasi qismlaridan SQL va Results programmalarini izohlang.
19. QBE to'rini izohlang.
20. SQL tilining oynasini izohlang.
21. QBE to'ridan foydalangan holda, uchta so'rov tuzing.
22. SQL tilidan foydalangan holda, uchta so'rov tuzing.

## **38-bo'b. MIJOZ ILOVALARINI YARATISH**

**Tayanch iboralar:** *Transact-SQL, ADO, SQL Server, OLE DB, COM, MS Visual Studio.NET, ADO.NET, OLE DB Provider for SQL Server, ODBC, Xpath, World Wide Web Consortium, ADO API, OLE DB API, MS SQL Server va MS Access aloqasi, Create New data Source, Talabalar\_SQL, berilganlar iste'molchilar, BBga kirish huquqi vositalari, mavjud berilganlar asosidagi toyicha, MS Data Engine.*

### **Mijoz ilovalarining yaratishdagi tarkibiy qismlar**

MS® SQL Server™ 2000 tizimida BB bilan ishlaydigan asosiy til bu - **Transact-SQL**.

Bu tildagi programmalar quyidagi tizimlarni generatsiya qiladi - MS Visual C++®, MS Visual Basic®, MS Visual J++® va boshqalar. Ular esa mijoz ilovalarini yaratishda quyidagi umumiy programmalar aloqasidan (Application Programming Interface - API) foydalanadi:

- **ADO** - MS ActiveX® Data ObjecS tezda murakkab ilovalarni yaratishni qo'llab-quvvatlaydi va **SQL Server** tizimining ko'pgina tarkiblariga kirish huquqiga ega.

ADO arxitekturasi – amaliy darajasidagi aloqa bo'lib, u OLE DB, COM aloqalar kutubxonasidan foydalanadi. ADO dan foydalanadigan yaratuvchini COM aloqalarini programmalashtirish ehtiyojidan saqlaydi.

ActiveX® – bu texnologiyalar to'plamidan iborat bo'lib, programma ta'minoti tarkiblarini, ularni yaratishda ishlataligan programma tilidan qat'i nazar, tarmoq muhitida o'zaro ishlashihi ta'minlaydi.

**OLE** – obyektlarni birlashtirish va joriy qiluvchi programma mahsuloti.

**COM (Component Object Model)** – Windows texnologiyasi. Ko'pgina ilovalarga kerak bo'lgan **SQL Server** tizimi tarkiblari, MS OLE DB Provider for **SQL Server** ni ishlatganda ADO texnologiyasini qo'llaydi.

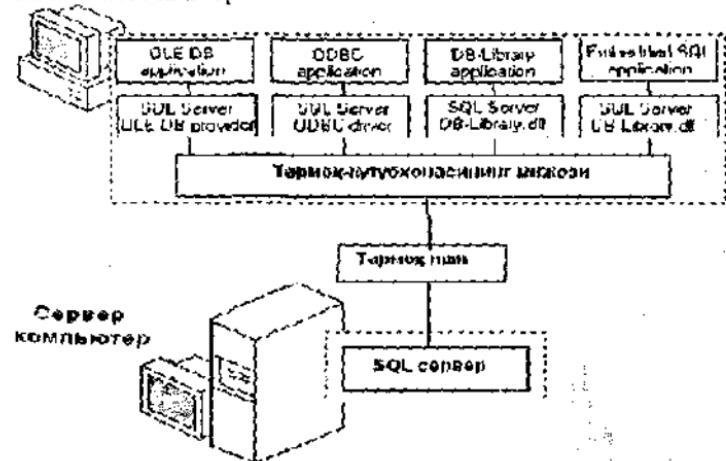
Ilovalarni MS Visual Studio.NET tizimida ishlab chiqilganda, berilganlarga kirishning ADO.NET obyekti ishlataladi, u berilganlar manbayidan ajralgan holda ishlashda yangi imkoniyatlarni yaratadi (ulanish, faqat berilganlarni olish va qayta jo'natish vaqtida bo'ladi).

- **OLE DB**, COM ga asoslangan vositalar uchun. OLE DB Provider for SQL Server, OLE DB-tafsirlariga kiritilmagan provayderning maxsus xossalari, SQL Server tarkiblariidan aloqalar va usullarni ishlataladi. Provayder tomonidan aniqlangan ko'pgina tarkiblarga ADO orqali kirish mumkin emas.
- **ODBC (Open Database Connectivity)** – Windows ilovalariga, BB drayveri o'matilgan berilganlar manbayiga murojaat qilish huquqini beradigan standart aloqa. SQL Server o'z bazalari bilan ilovalarning ishlashi uchun o'zini drayverini o'matadi.

**MS® SQL Server™ 2000** tizimida BB bilan ishlaydigan ikkinchi til - Xpath – tili bo'lib, u **W3C (World Wide Web Consortium)** standartida keltirilgan. Uning hujatlari XML-formatda bo'ladi. SQL Server tizimi bilan o'zaro aloqachilar - **ADO API**, **OLE DB API**.

Mijoz tarkiblari va serverning o'zaro ishlashi chizmasi 38.1-rasmida ko'rsatilgan.

Мижоз комп'ютер

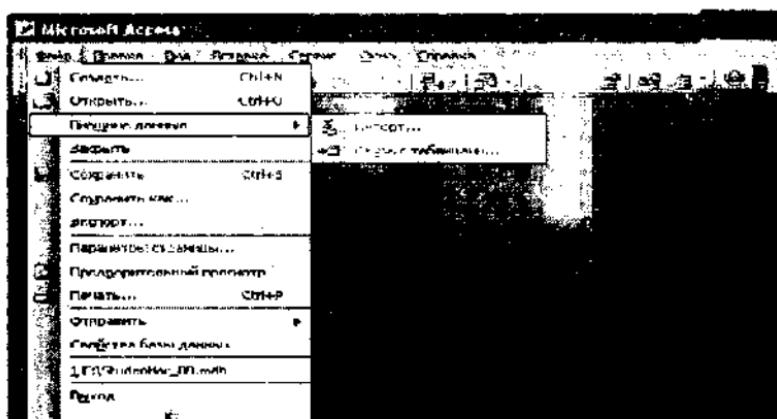


38.1-rasm. Mijoz tarkiblarining server bilan o'zaro ishlash chizmasi.

BB bilan ishlashni, ODBC (SQL Server drayveri) dan foydalaniib, MS Access yoki Visual FoxPro ishlatgari holda tashkil qilish mumkin.

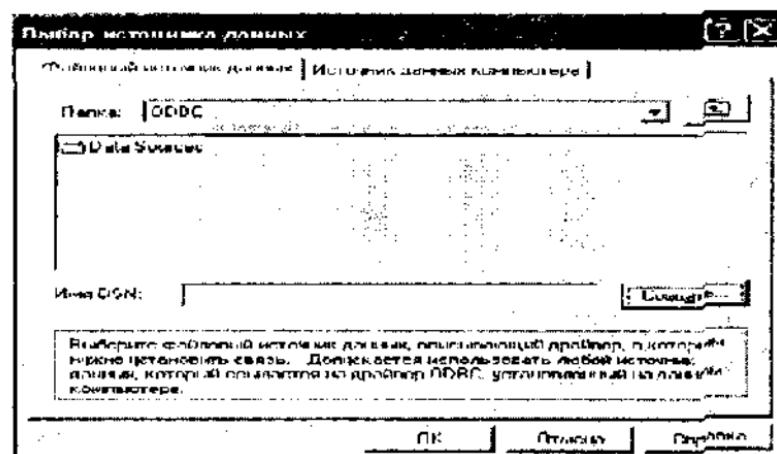
## MS SQL Server va MS Access aloqasi

MS Access BBda SQL Server tiziminining BB jadvali bilan aloqani o'tnalish mumkin. Buning uchun Fayl → Внешние данные → Связь с таблицами yoz'nalishida (bazaning kontekst menyusida) komandani tanlash kerak va SQL Server tizimidagi SQL Server drayverini tanlagan holda, undagi BB bilan yangi ularishni tuzish kerak (yoki borini tanlash kerak).



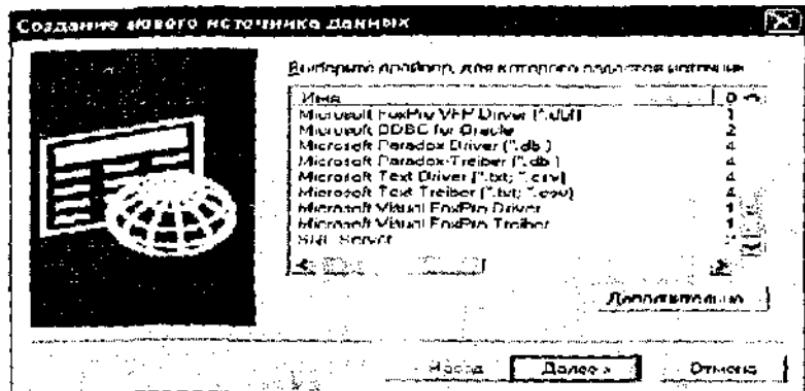
38.2-rasm.

Buning uchun aloqa uchun «fayl turi» – «ODBC» BBni tarlaymiz va payda bo'lgan oyunda «New» komandasini tanlaymiz.

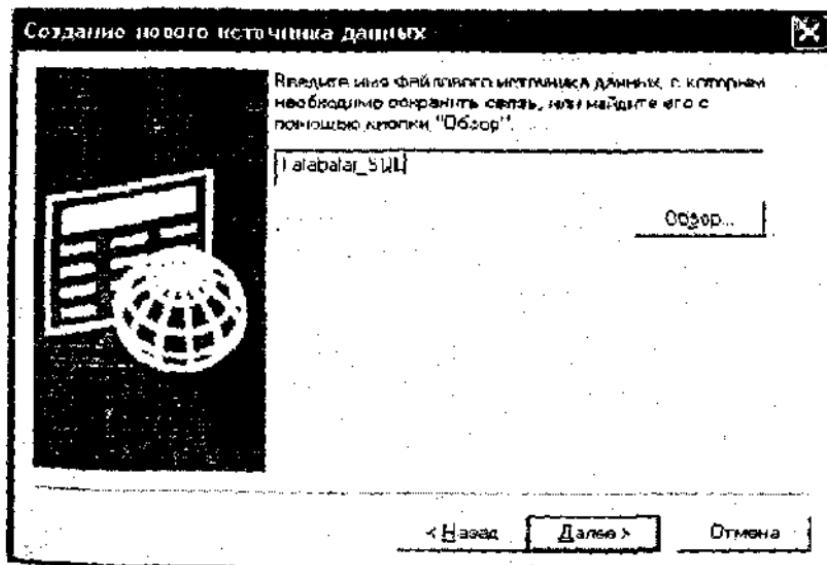


38.3-rasm.

Keyingi «Create New data Source» oynasida «SQL Server» drayverini tanlaymiz va ularish nomini kiritamiz - Talabalar SQL. Shundan so'ng «Create a New Data Source to SQL Server» oynasida SQL-server nomini kiritamiz (indamagan holda o'smatish uchun bu nom lokal) va Talabalar ularishi o'smatiladigan BB nomini ro'yuxatdan tanlaymiz.

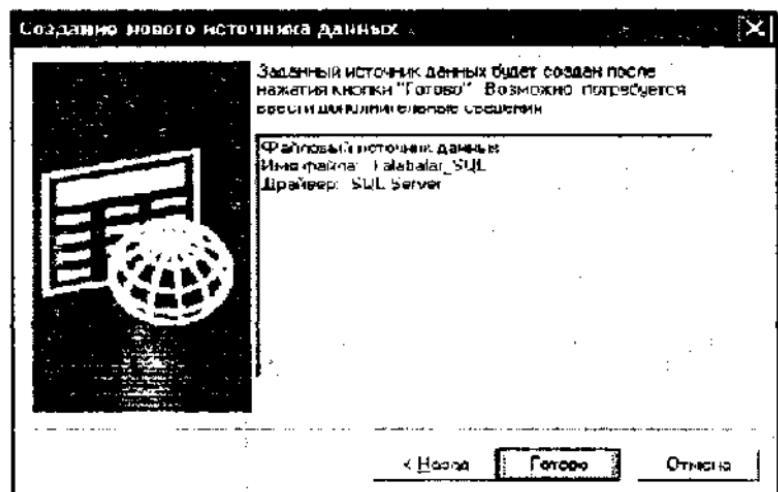


38.4-rasm.

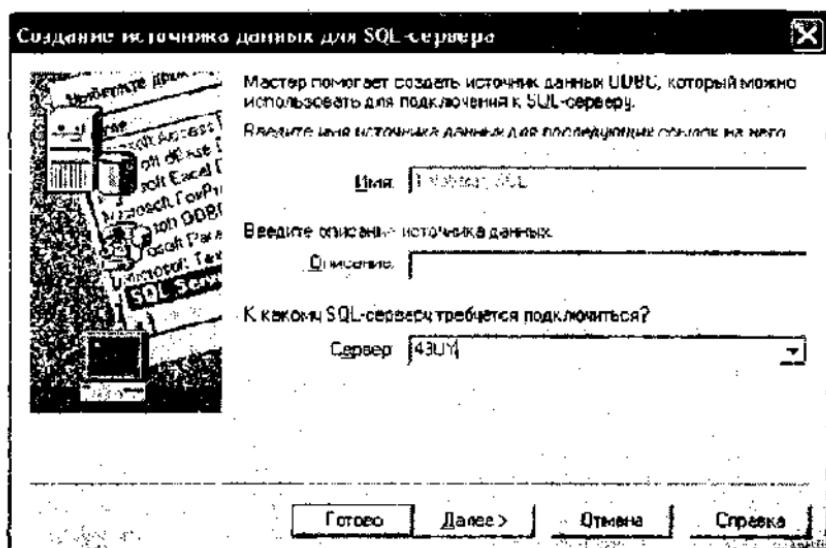


38.5-rasm.

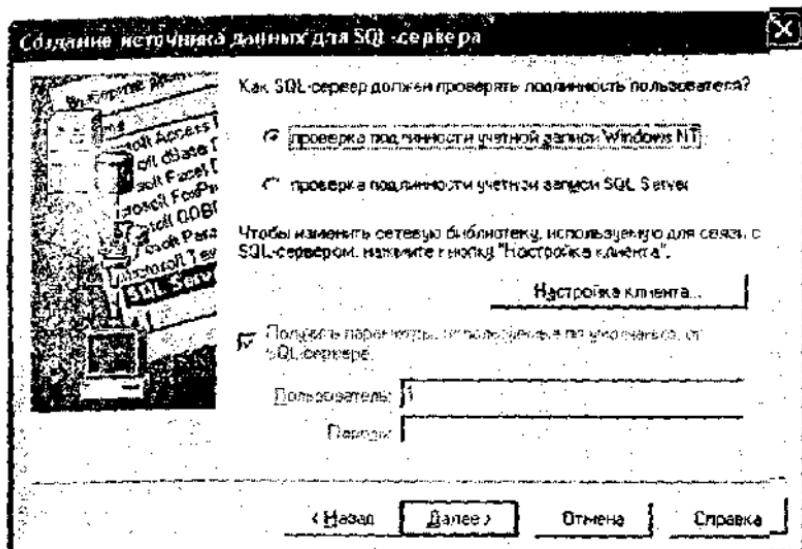
38.6 - 38.13-rasmarda ketma-ket bajarilishi lozim bo'lgan qadamlar keltirilgan.



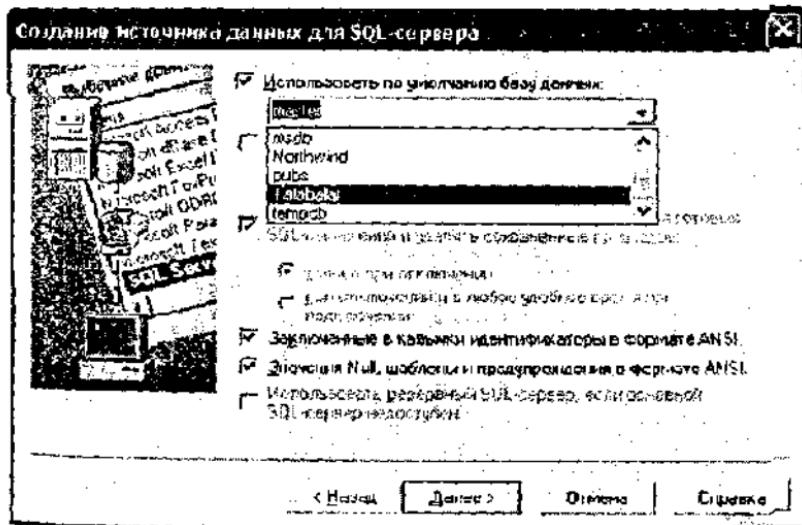
38.6-rasm.



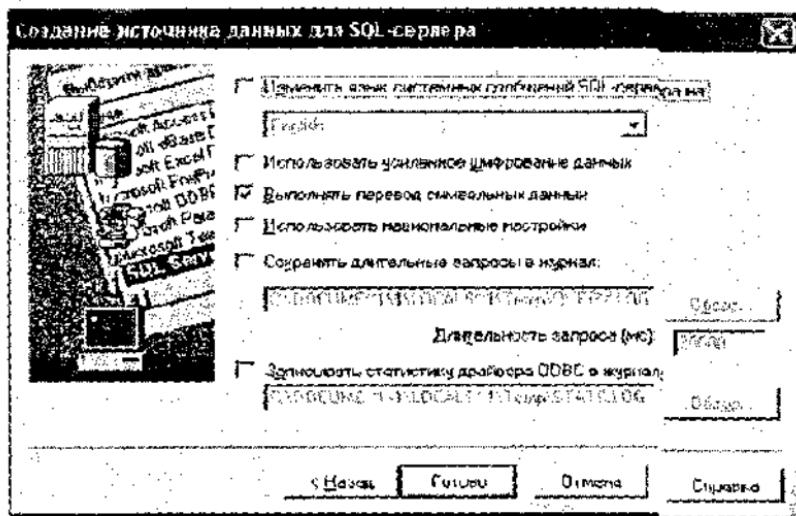
38.7-rasm.



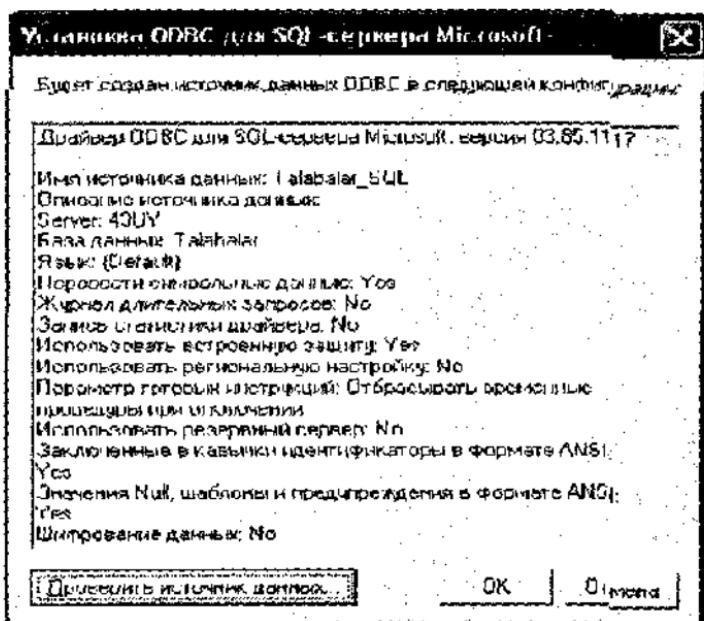
38.8-rasm.



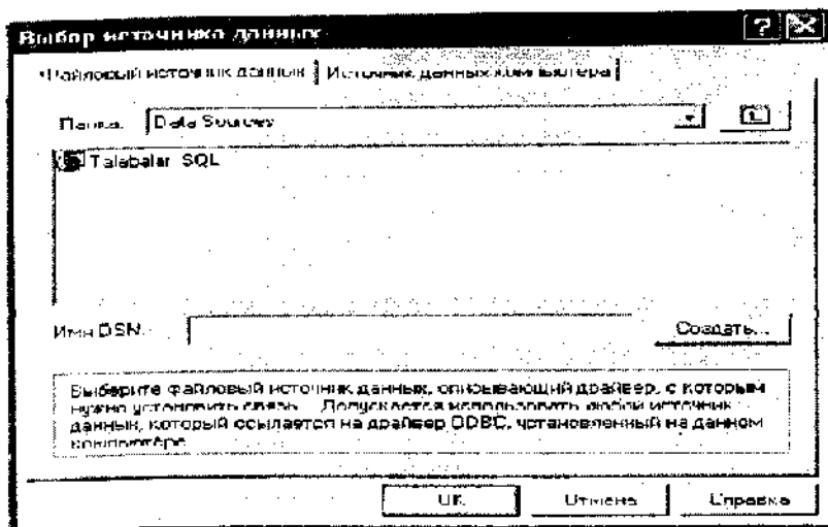
38.9-rasm.



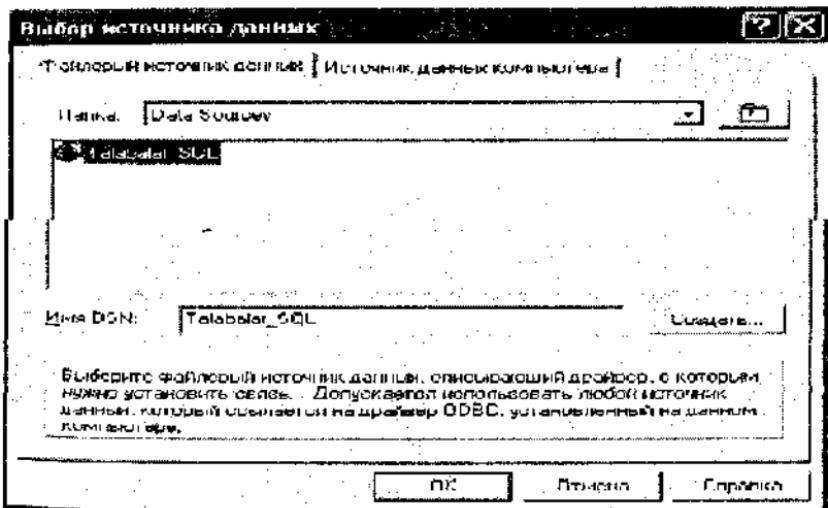
38.10-рasm.



38.11-рasm.



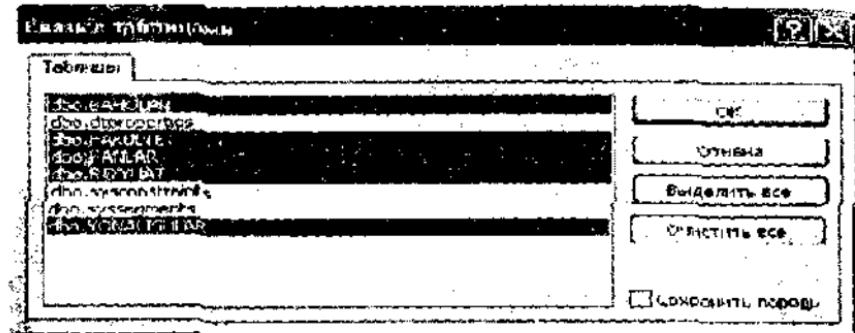
38.12-rasm.



38.13-rasm.

*So'nqgi SQL Server ODBC Data Source Test oynasida*

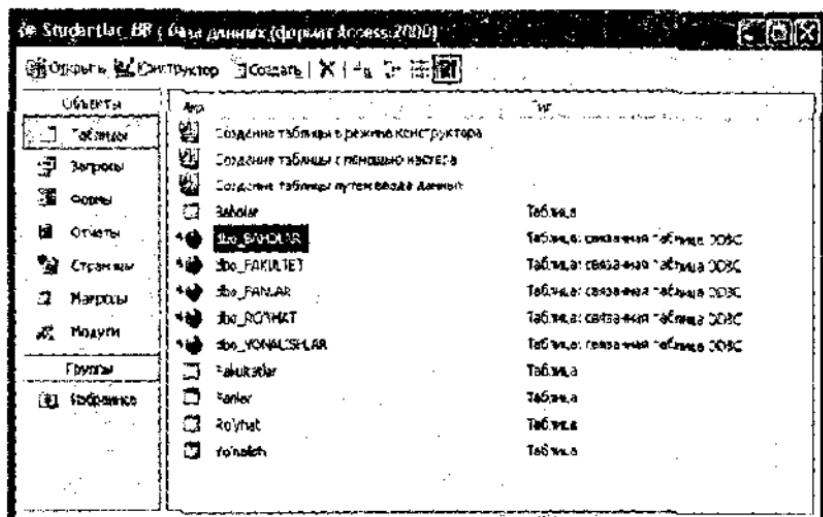
**TESS COMPLETED SUCCESSFULLY!**  
xabarini olishimiz kerak.



38.14-rasm.

### MS Access da BB bilan ishlash uchun SQL Server tizimidan foydalanish

So'ngra MS Access da ishlatish uchun SQL Server bazasi jadvalini tanlashimiz mumkin (38.15-rasm).



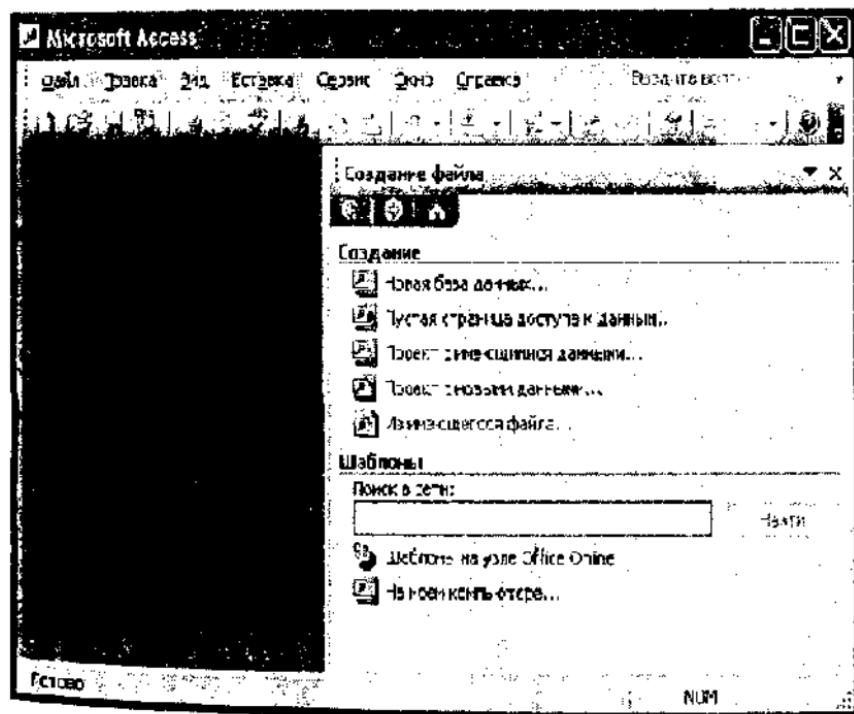
38.15-rasm. Berilganlar bazasi jadvallarini tanlash oynasi.

Natijada MS Access bazasi jadvallari ro'yxatida biz tanlangan MS SQL Server bazasi jadvallari bilan aloqani ko'tarmiz. Shunda ham, bu jadvallarda axborotni to'g'rilash, qo'shish va o'chirish imkoniyatlari mavjud. Saqlash komandasidan so'ng berilganlar SQL Server bazasiga qayta yoziladi.

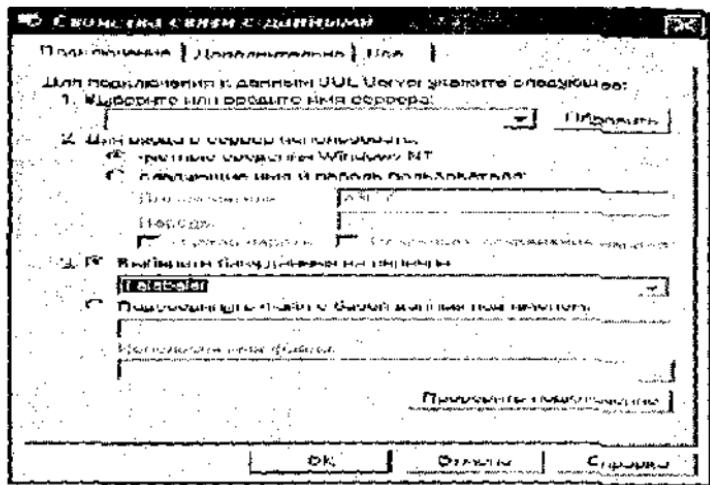
*MS SQL Server tizimi berilgantarı bilan ishlash MS Access toyihasını ishlataǵan holda uyuştirilishi mungkin.*

MS Access (\*.adp) loyihasi bu Access fayllarining yangi turi, OLE DB tarkiblari arxitekturasi yordamida MS SQL Server BBga samarali, tabiiy kirish huquqini yaratadi. OLE DB arxitekturasida berilganlarga kirish huquqi berilgan ilovalarni, *berilganlar istemotchilarini* deyiladi (masalan, MS Access 2000 yoki MS Visual Basic 6.0), berilganlarga ichki kirish huquqini ta'minlaydigan programmalar, BBga kirish huquqi vositalari deyiladi (masalan, MS OLE DB Provider SQL Server uchun yoki MS Jet 4.0 OLE DB Provider).

Access loyihasi yordamida «mijoz-server» turidagi ilovani oson yaratish mumkin. Buning uchun MS Access da **«Проект с имеющимися данными»** (mavjud berilganlar asosidagi loyiha) komandasini tanlaymiz (38.16-rasm) loyihaга nom beramiz, 38.17 rasmdagi oyna chiqadi va u bo'yicha SQL-serverdan **Talabalar** BB bilan aloqani tanlaymiz.

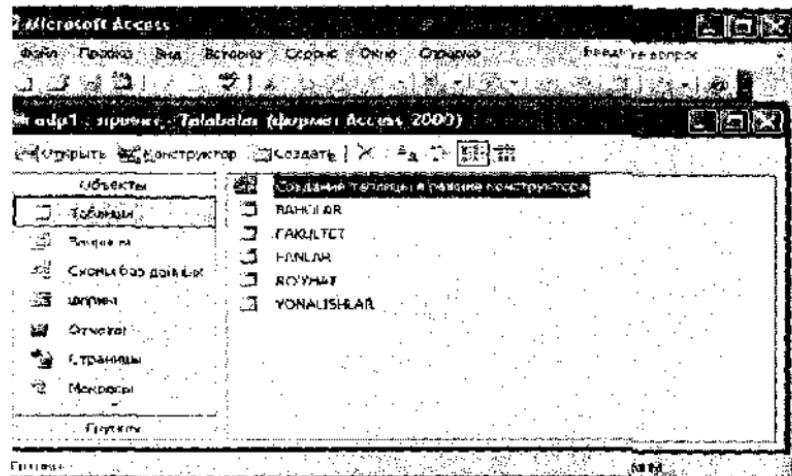


38,16-rasm.



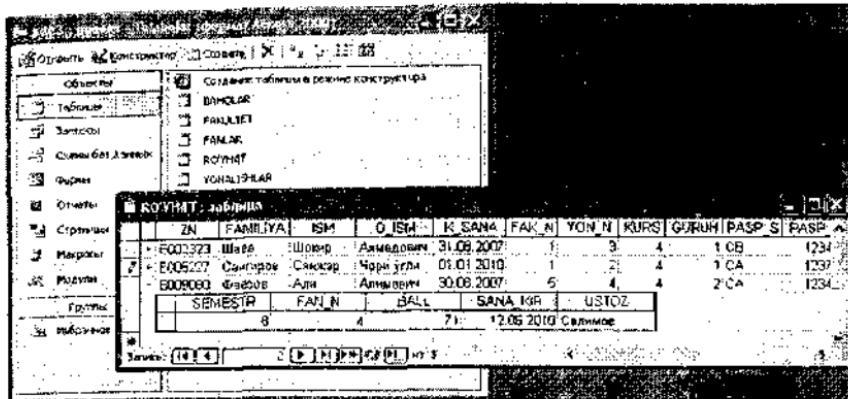
38.17-rasm. Yangi MS Access loyihasi uchun BB bilan aloqani kiritish oynasi.

Olingan loyiha oynasi (38.18 rasm) MS Access BB bynasidan tashqi itdan deyarli farq qilmaydi. Oynada yangi «BB» bo‘limi mavjud, vallar Конструктори оynasida SQL Server tizimi berilganlarining turi dalaniladi, BB shizmasi xuddi SQL Server tizimi ko‘rinishiga ega.



38.18-rasm. MS Access loyiha oynasi.

**38.19-rasmida SQL Server da yaratilgan Talabalar BBdagi RO'YHAT jadvalining MS Access ADP3 loyihasidagi Talabalar BB**



**38.19-rasm. SQL Server da yaratilgan RO'YHAT jadvali.**

MS Access loyihasi bilan ishlash Access BB bilan ishlashga juda o'xshaydi. Shakl, hisobot, berilganlarga kirish huquqi varag'i, makroslar va modular yaratish jarayoni bir xil. SQL Server BB ulangandan so'ng jadvallarni, tasavvurlarni, saqlangan protseduralar va BB chizmalarini qayta ko'rish, yaratish, o'zgartirish va o'chirish mumkin. Loyihada shakllarni, hisobtlarni va berilganlarga kirish huquqini beradigan Web-sahifalarni ishlab chiqish uchun Macrepdan foydalanish mumkin.

MS Access loyihasi yangi **MSDE (MS Data Engine)** texnologiyasini ishlatadi, u **MS SQL Server** da berilganlarni birlgilikda mahalliy saqlash imkoniyatini yaratadi. **MSDE** ni fayl serverini MS Jet BB yadrosiga muqobil «mijoz-server» arxitekturasida berilganlarga ishlov berish yadrosi deb tasvirlash mumkin. **MSDE** texnologiyasi kichik kompyuterlarda ishlatish uchun ishlab chiqilgan va optimallashtirilgan, ya'ni foydalanuvchilarning ish stansiyalari yoki ishchi guruhlarining kichik serverlari uchun.

MS Access berilganlari va obyektlarini **SQL Server** formatiga eksport qilish uchun **SQL Server** formatiga o'zgartiruvchi masterdan foydalanish mumkin. Macrep MS Access (*.mdb*) BBni yangi yoki mayjud bo'lgan **MS SQL Server** BBga yoki yangi MS Access (*.adp*) loyihasiga berilganlarni o'zgartirish, tasvirlash va BB obyektlarini ko'chirish yo'li bilan o'zgartiradi.

## **Xulosə**

*Mijoz ilovalarini yaratishda MS SQL Server ilovalarga ADO, OLE DB, ODBC, ADO API, OLE DB API texnologiyalaridan foydalananish imkoniyatini beradi. Shuning uchun ham, MS SQL Server dagi Talabalar BB bilan aloqa, ODBC dan foydalangan holda, MS Access bilan tashkil qilindi.*

### **Nazorat uchun savollar va amaliyot uchun topshiriqlar**

1. MS SQL Server BBB tizimida SQL tilining qaysi naqli ishlataladi?
2. MS SQL Server BBB tizimida SQL tili qaysi muhitlar uchun berilganlar bazalarining ilovalarini yaratishga xizmat qiladi?
3. Umumiy tarzda ishlataladigan programmalar orasidagi aloqalar uchun qaysi programma mahsulotlari ishlataladi?
4. Programmalar orasidagi aloqalar uchun ishlataladigan programmalardan ADO ni izohlang.
5. Programmalar orasidagi aloqalar uchun ishlataladigan programmalardan OLE DB ni izohlang.
6. Programmalar orasidagi aloqalar uchun ishlataladigan programmalardan ODBC ni izohlang.
7. MS SQL Server BBB tizimida BB bilan yana qaysi tilda ishlash mumkin?
8. Mijoz tarkiblарining server bilan o'zaro ishlashi chizmasini izohlang.
9. MS SQL Server va MS Access BBB tizimlari orasidagi aloqani mustaqil ravishda ta'labaslar yoki ixtiyoriy boshqa BB orqa si o'mating va izohlang.
10. MS Access loyihasi uchun MS SQL Server dagi BB bilan aloqani o'matishni amalga oshiring va uni shaxhlang.
11. MS Access loyihasi bilan ishlash, Access BB bilan ishlashdan nimasi bilan farqlanadi va o'xshaydi?
12. MS Access loyihasi qaysi texnologiyadan foydalanadi va uning asosiy vazifasi nimadan iborat?
13. MS Access loyihasida formatning qaysi kengaytirmasi ishlataladi?
14. MS Access loyihasidagi yangi hosil bo'lganlar va eski bazadagi jadvallarni o'zaro solishtirib, izohlab bering.
15. MS SQL Server va MS Access BBB tizimlari orasidagi aloqadan foydalangan holda, bazalardagi jadvallarga yangi berilgahlarni kriting va natijani izohlang.
16. MS SQL Server va MS Access BBB tizimlari orasidagi aloqadan foydalangan holda, baza jadvallaridagi berilgahlarni o'zgartiring va natijani izohlang.

- ... MS Access BBB tizimlari orasidagi aloqadan tashlang va natijani izohlang.
- 18. MS SQL Server va MS Access BBB tizimlari orasidagi aloqadan foydalangan holda, oddiy so'rovlarini tashkillashtiring va natijani izohlang.
  - 19. MS SQL Server va MS Access BBB tizimlari orasidagi aloqadan foydalangan holda, murakkab so'rovlarini tashkillashtiring va natijani izohlang.
  - 20. MS SQL Server va MS Access BBB tizimlari orasidagi aloqadan foydalangan holda, MS SQL Server dagi QBE to'ri orqali ixtiyoriy so'rovni yarating va izohlang.
  - 21. MS SQL Server va MS Access BBB tizimlari orasidagi aloqadan foydalangan holda, MS Access dagi QBE to'ri orqali ixtiyoriy so'rovni yarating va izohlang.
- ...

## GLOSSARY

**Agregat funksiya** – aggregate function – агрегатная функция – relatsion xildagi BBBT so'rovlar tiliga kiritilgan funksiya.

**Aloqa** – relation – связь – obyektlar o'rtasidagi funksional bog'lanish. Relatsion BBda jadvallar o'rtasida kalitlar orqali bog'lanishlar o'matiladi, ularning biri asosiy (parent, ajdod, ona) jadvalda birlamchi, ikkinchisi tashqi kalit - (child, avlod, bola) jadvalda o'matiladi. Odatda, tashqi kalit birinchi darajadagi kalit bo'la olmaydi va aloqaning «birdan ko'plikka» (1:M) turini tashkil qiladi. Agarda birinchi darajadagi kalit tashqi kalit bo'lsa, jadvallar orasidagi aloqa «birdan birga» (1:1) turida bo'ladi. Aloqalar xususidagi axborotlar BBda saqlanadi.

**Aloqa** – relationships – связь – konseptual darajadagi munosabatlar orasidagi bog'lanish.

**Aloqa darajasini** – relationships degree – степень связи – aloqada ishtirok etuvchi mohiyatlar soni.

**Aloqalar nusxalari** – relationships instances – экземпляр связей – mohiyatlar nusxalari orasidagi o'zaro munosabatlar.

**Aloqalar sinfi** – relationships classes – класс связей – mohiyatlar sinflari orasidagi o'zaro munosabatlar.

**Aloqaning maksimal kardinalligi** – maximum cardinality – максимальная кардинальность связей – ikkita maksimal kardinal sonlardan iborat ikki tomonli aloqa birligi. Maksimal kardinal sonlar aloqalarni belgilashdagi romb ichidagi aloqalar sonining chegarasini ifodalaydi va har bir tomondagi mohiyatlarning maksimal tusxalarini bilditadi.

**Aralash BB** – hybrid database – гибридная база данных – 1. Belgili berilganlar bilan bir qatorda grafik berilganlarni saqlovchi BB, 2. Har xil tuzilishdagi BB, masalan, iyerarkik va to'rsimon tur.

**Aralash BBBT** – hybrid database management system – гибридная СУБД – BBBT aralash turi BBBT tili va BBBT avtonom tilidan iborat shakli.

**Arxetur (naql) obyektlar** – archetype/version object – архетурный объект – boshqa semantik obyektlarni yaratuvchilar.

**Arxitektura darajasi** – architecture layer – степень архитектуры – BBBT arxitekturasidagi qatlam bo'lib, ularning funksiyalaridan birini modellashtiradi.

**Asosiy til aloqasi – host language interface – интерфейс основного языка** – BBBT tomonidan beriladigan, asosiy programmalash tili uchun aloqa.

**Assotsiativ obyekt – association object – ассоциатив объект** – bu ikki va undan ortiq obyektlarni bog'lovchi va shu aloqa xususidagi berilganlarni saqlovchi obyekt.

**Axborot mantiq modeli – information logical model – инфологическая модель** – ma'lumot obyektlarining majmuasini, ularning alomatini va obyektlar orasidagi munosabatni, predmet sohaning o'zgarish jo'shqinligini hamda foydalanuvchining axborotga muhtojlik tavsifnomasini aniqlaydigan, predmet soha modeli. Predmet sohani loyiha oldi tekshirish natijasida yaratiladi hamda berilganlar bankini texnik iqtisodiy asoslash va loyihalashning texnik topshirig'ini ishlab chiqish uchun asos bo'ladi.

**Axborot mantiqiy chizma – information logical schema – инфологическая (информационно-логическая) схема** – predmet soha obyektlari va ular orasidagi munosabatlar mazmunining tarkibini hisobga olgan holda aniq BBBTni bog'liqsiz ravishda tavsiflash. BBning konseptual modelida jadvalga mos mohiyat (entity) bo'lib, bunda atributlarning aniq termasi mavjud bo'lib, ular aniq qiymatlarni qabul qilishi mumkin (mumkin bo'lgan qiymatlar termasi - domen).

**BB administratori – database administrator – администратор баз данных** – BB xususida to'liq tasavvurga ega bo'lgan hamda uni yuritish, foydalanish va rivojlanishiga javobgar maxsus lavozimli shaxs (shaxslar guruh).

**BB bilan ishlash uchun ilova – database application – приложение для работы с базами данных** – BBning yozuvlarini saqlovchi fayllarni boshqarish hamda turli amallar (ko'rish, izlash, saralash, qayta guruhlash va h.k.)ni bajarish imkonini yaratuvchi, ilova (programma).

**BB bo'lagi – database area – фрагмент баз данных** – berilganlarni, boshqa atrofiga bog'liqsiz ravishda ochish mumkin bo'lgan, BB atrofi.

**BB chizmasi – database scheme – схема базы данных** – berilganlarning ma'lum bir modeli asosida BB tavsiflash.

**BB kaliti – database key – ключ базы данных** – BBdagi kaliining takrorlanmas nomeri.

**BB ma'lumotnomasi – database directory – справочник базы данных** – BB xususidagi asosiy ma'lumotlarni saqlovchi va BBBT tomonidan BB yozuvlariga murojaat uchun foydalilaniladigan maxsus fayl.

**Berilganlar banki – databank – банк данных** – berilganlarni markazlashgan holda saqlash va jamoa tarzda foydalanishga mo'ljalangan avtomatizatsiyalashgan axborot tizimi. Berilganlar banki tarkibiga bir yoki bir nechta BB, BB so'rovnomasi, BBBT, APIlar va so'rovlar majmuasi kiradi.

**Berilganlar bazasi (BB) – database – база данных – ma'lum predmet sohaga taalluqli bo'lgan, maxsus tuzilmaga ega berilganlar majmuasining nomi; BBBT boshqaruvida ishlaydigan, bir yoki bir qancha ilovalar optimal tarzda ishlataladigan, o'zaro bog'langan berilganlar majmuasi; berilganlarning tuzilishi uni ishlataladigan programmalarga bog'liq bo'lmaydi; berilganlar shunday tuzilishga ega bo'ladiki, bunda ularni ko'paytirish, o'zgartirish, olib tashlash va boshqa imkoniyatlar bo'ladi.**

**Berilganlar bazasiga so'rov – database request – запрос к базе данных – programma yoki foydalanuvchi tomonidan, BB yozuvlarini izlash uchun yuborilgan, axborot so'rovi.**

**Berilganlar bazasini boshqarish – database control – управление базой данных – BB yaratish va yurgizish, programma va foydalanuvchilar ning BBga kirish, izlash va so'rovlari asosida ma'lumotlarni chiqarishni boshqarishdan iborat, BBBTning asosiy funksiyasi.**

**Berilganlar bazasini boshqarish tizimi – database management system – система управления базами данных – BBni yaratish, yuritish va foydalanishga mo'ljalangan programma va til vositalarining majmuasi. Programmalar va tillar vositalari berilganlar bazasini hosil qilish va o'zgartirish, qo'shish, boshqarish, o'chirish, qidirish va ma'lumotlarni saralash, ma'lumotlarni chop etishga tayyorlab oynaga chiqarish, ma'lumotlarga egalik huquqini chegaralash hamda boshqa xizmatlarni bajaradi.**

**Berilganlar bazasini boshqarishning tarmoq tizimi – network database management system – сетевая система управления базами данных – tarmoqlangan BBni yaratish, yuritish va foydalanish uchun mo'ljalangan BBBT.**

**Berilganlar bazasini generatsiyalash – database generation – генерация баз данных – BBni yaratish jarayoni bo'lib, berilganlar majmuasini yaratish va formatlash, berilganlarni yuklash, BB so'rvonmasini yaratish, BB tashuvchilarini aniqlash funksiyalarini o'z ichiga oladi.**

**Berilganlar bazasini nusxalash – replication – репликация – berilganlar bazasining nusxasini hosil qilish, ya'ni sinxronlash jarayonida yangilanuvchi berilganlar yoki nusxalangan shakllar, hisobotlar yoki boshqa obyektlar bilan amallarni bajarish.**

**Berilganlar bazasini tashkil qilish – database organization – организация баз данных – berilganlarni mantiqiy va fizik darajalarda, BBni tashqi, konseptual, ichki chizmalari hamda ular modullarini tasvirlashning usul va vositalar majmuasi.**

**Berilganlar bazasini tiklash – database recovery – восстановление баз данных – mashinaning ko'zda tutilmagan to'xtashlari yoki programmaning xatosi natijasida yuzaga keladigan berilganlar butunligini saqlash uchun**

**zaxira nusxadan, BB tizlash.** Tizlash usul va vositalari: nusxalash, tekshiruv nuqtasidan qaytadan ishga tushirish, tizim jurnali.

**Berilganlar bazasini yuklash – database load** – загрузка базы данных – BBBT bajaradigan boshlang'ich berilganlarni, saqlanadigan tuzilishga keltirish.

**Berilganlar bazasini yuklash – database load** – загрузка базы данных – BBBT bajaradigan boshlang'ich berilganlarni, saqlanadigan tuzilishga keltirish.

**Berilganlar bazasining fayli – database file** – файл баз данных – nisbatan mustaqil bo'lgan, boshqa fayllarga bog'liqsiz o'chish mumkin bo'lgan va shu BBning boshqa fayllari bilan bog'langan BB fayli bo'lagi.

**Berilganlar bazasining jadvali – database relation** – таблица базы данных – bir xil atributli ustunlarga (maydon, fields, pole) ajratilgan va bir xil turdag'i satrlardan (yozuv, records, zapis) iborat doimiy tuzilishdir.

**Berilganlar bazasining obyekti – database object** – объект базы данных – BBdagi nomga ega va xotiradan ma'lumi joyni egallagan, ixtiyoriy obyekt. BBning relatsion modelidagi obyektlar jadva'llar va indeksladir.

**Berilganlar bazasining serveri – database server** – сервер базы данных – BBga kiritish, saqlash, himoya va ishlavni ta'minlovchi, ilova. Masalan, Microsoft SQL Server.

**Berilganlar bazasining shakli – database form** – форма баз данных – BB bilan ishlash uchun ilovaning tarkibiga kirgan xizmatchi programma bo'lib, u foydalanuvchiga berilganlarga kirish bo'yicha ko'pgina masalalarni yechishga yordam beradi, masalan, berilganlarni kiritish, ko'rish, o'zgartirish va h.k.

**Berilganlar bazasining tahrirchisi – database editor** – редактор баз данных – BBBTda programma bo'lib, BB tuzilishi va unda saqlanayotganlarni tahrirlashni ta'minlaydi.

**Berilganlar bilan ishlash – data manipulation** – манипулирование данными – BBBTda BB ustidan bajariladigan amallar; BBni o'chish va yopish, izlash, o'qish, yangilash, qo'shish, yo'qotish majmuasi.

**Berilganlar majmuu guruhi – data set group** – группа наборов данных – BBBT orqali BB yuklash jarayonida yaratiladigan bir yoki bir nechta berilganlar majmuasi.

**Berilganlar modeli – model data** – модель данных – berilganlar orasidagi munosabatlarni qo'yilgan talablarga nisbatan amalga oshirish imkoniyatini beruvchi intellektual vosita. *Kuchli turdag'i modellarda*, hamma berilganlar qaysidir bir turkumga kiritilgan holatda ko'riladi. *Kuchsiz turdag'i modellarda esa*, berilganlar hech bir turkumlarga bog'lanmagan holatda ko'rifadi.

**Berilganlar relations bazasi – relation database – реляционная база данных** – berilganlar bazasi bo'lib, bunda berilganchalar tavsiflash ma'lumotlari, berilganlarning obyektlari orasidagi munosabatlar algebrasini ko'rsatish asosida tuziladi.

**Berilganlarga ishllov – processing data – обработка данных** – berilganlar ustida bajariladigan izlash, qo'shish, olib tashla va, o'zgartirish algoritmlarini amalga oshiruvchi funksiyalar faoliyatining natijasi.

**Berilganlarning havola butunligi – referential integrity – ссылочная целостность данных** – bog'langan jadvallardagi kalitlarning qiymatlarini bir-biriga mosligini ta'minlovchi, qoidalari majnuasidir.

**Berilganlarning konseptual modeli – conceptual data model – концептуальная модель данных** – konkret BBBT atamalarida berilgan ma'lumotlar modeli bo'lib, berilganlarning to'liq to'plami va ular orasidagi aloqalarni saqlaydi. BBBT arxitekturasida, tashqi va ichki darajalar (qatlamlar) orasida turadi.

**Berilganlarning relations bazasi – relational database – реляционная база данных** – berilganlarning relations modeliga mos ravishda amalga oshirilgan BB.

**Berilganlarning tarmoq tuzilishi – network data structure – сетевая структура данных** – ixtiyorli tuguniga bittadan ortiq aloqa kirishi mumkin bo'lgan yo'naltirilgan grafdan iborat, berilganlar tuzilishi.

**Beshinchchi normal shakl (5NSh) – fifth normal form (5NF) – пятая нормальная форма (5НФ)** – ushbu shakldagi munosabatlarning atributlari birlashtirish natijasidagi o'zaro bog'liqliklarga ega bo'lmaydi, ya'ni munosabatlarning atributlari orasidagi birlashish bog'liqliklari yo'qotilgan holat.

**Binar (ikkilik) aloqa – binary relationships – бинарная связь – aloqa darjasasi ikkiga teng bo'lgan holat, ya'ni ikkita mohiyatlar orasidagi aloqa.**

**Bir jinsli bo'lmagan BB – heterogeneous database – неоднородная база данных** – boshqa BBga nisbatan, BBBT boshqa xili tomonidan yaratilgan, BB.

**Bir qiymatli atribut – single-value attribute – однозначный атрибут – ma'lum bir turdag'i mohiyat nusxasining bitta qiymatini saqlovchi atribut tushuniladi. Semantik obyektda bir qiymatli atribut deganda, maksimal kardinalligi birga teng bo'lgan atribut tushuniladi.**

**Birinchi normal shakl (1NSh) – first normal form (1NF) – первая нормальная форма (1НФ)** – munosabatning ta'rifini qondiruvchi ixtiyorli berilganlar jadvali 1NShda bo'ladi. Jadval munosabat bo'lishi uchun quyidagi qoidalarga amal qilinishi shart.

**Birlanchi kalit – primary key – первичный ключ – asosiy kalitli element bo'lib, jadval satrini noyob tarida belgilaydi.**

**Blokning nisbiy adresi – relation block address – относительный адрес блока – berilgan blok boshlanishiga nisbatan berilgan blokning adresi.**

**Boys-Kodd normal shakli (BKNSh) – Boyce-Codd normal form (BK/NF) – нормальная форма Бойса-Кодда (НФБК) – munosabatning har bir determinanti kalitga noimzod bo'lgan holat. Bu semantik obyekt bo'lib, uning tarkibida kamida bitta ko'p qiymatli guruh atributti bo'ladi va bu atribut tarkibida esa obyekt atribut joylashadi.**

**Butunlikni cheklash – integrity constraint – ограничение целостности – berilganlarga yuklanadigan mantiqiy cheklanishlarni aniqlaydigan va BBBTda berilganlarning butunligini saqlash uchun ishlataladigan, qoida.**

**Chizma – scheme – схема – berilganlar bazasining umumiyligi mantiqiy tuzilishi. Berilganlar bazasining hamma relatsion chizmalarining to'plami konseptual model yoki konseptual chizma deb ataladi.**

**Chizmaning nusxasi – scheme instances – экземпляр схемы – qiymatlar (tana) bilan keltirilgan munosabat.**

**ER-diagrammalar – ER-diagrams – ER-диаграммы – qarahg → mohiyat-aloha diagrammalari**

**Fayllarni bo'shatish – file uncoupling – развязывание файлов – BBB tizimida berilganlar bazasidagi alohida yozuvlar orasidagi avval o'matilgan aloqalarni uzish, protsedurasi.**

**Fizik kalit – physical key – физический ключ – berilganlar ustidagi ishlov tezligini oshiruvchi indeks.**

**Fizik tavsif – physical description – физический дескриптор – berilganlar turini (masalan, son yoki satr), uzunligini va berilganlarning boshqa chegaralarini ko'rsatadi.**

**Foydalaniш uslubi – usage mode – режим использования – BBB tizimida, BB sohalarini ochishda o'matiladigan va shu sohalariga boshqa avtomatizatsiyalashgan programmlarning parallel kirish imkonini aniqlovchi qoida.**

**Funksional bog'langan – functional dependency – функциональная зависимость – agarda vaqtning har daqiqasida R munosabatdagagi A atributning har bir elementiga shu munosabatdagagi V atributning bittadan ortiq bo'lmagan elementining mos kelishi, V atributning A atributga funksional bog'lanishini bildiradi.**

**Gibrid (aralash) obyekt – hybrid object – гибридный объект – bu kompozit va tarkibli obyektlar kombinatsiyasidan iborat obyekt. Xususiy holda, gibrid obyekt**

**Giperindeks – hyper index – гипериндекс – ba'zi BBBTdag'i, BBda indekslar tuzilishidagi oliy darajadagi indeks.**

**Guruh identifikatori – group identifier – групповой идентификатор** – bu bittadan ortiq atributdan iborat identifikator.

**Havola butunligi – referential integrity constrains – ограничения ссылочной целостности** – agarda munosabatda tashqi kalit mavjud bo'lsa, tashqi kalitning qiymati asos munosabatdagi qaysidir kortejdagi nomzod k- kalit qiymatiga mos kelishi yoki tashqi kalit to'liq NULL lardan iborat bo'lishi kerak, degan shartning bajarilishi ta'minlanadi.

**Hosilaviy berilganlar – secondary data – производные данные** – boshqa berilganlardan hosil bo'lgan berilganlar. Bunday berilganlarga mos, ba'zi BBBTdagi tarkibli va kesma ajomatlari kiradi.

**Ichki daraja – internal level – внутренний уровень** – berilganlarning modellashtirishda BBBT va operatsion tizimning berilganlarni qabul qila olishdagi berilganlarning ichki tuzilishi.

**Ichki model – internet model – внутренняя модель** – BBBT arxitekturasidagi quyi qatlari (fizik) darajasi bo'lib, u berilganlar tasvirini va unga kirish usullarini tashqi xotirada akslantiradi.

**Identifikator – identifier – идентификатор** – kortejlarni bir qiymatli belgilovchi. Identifikator noyob yoki noyob bo'lmashligi mumkin.

**Identifikatsiyalangan bog'liq mohiyatlar – ID dependent entities – идентифицированные зависимые сущности** – «mohiyat-aloqa» modelidagi zaif mohiyatlarning alohida turi bo'lib, ularning identifikatorlari boshqa mohiyatni identifikatorlarini saqlaydi.

**Identifikatsiyalash – matching – отождествление** – BBBTda istiyoriy identifikatorlarni BBning elementlariga bog'lash.

**Ikkinci normal shakl (2NSh) – second normal form (2NF) – вторая нормальная форма (2НФ)** – munosabat 1NShga keltirilgan va har bir kalit bo'lmasagan atribut tarkibli birlamchi kalitga to'liq funksional bog'liq bo'lgan holat.

**Iyerarxik BBBT – hierarchical database management system – иерархическая СУБД** – iyerarxik BBni yaratish, yurgizish va foydalanish uchun ishlataladigan, BBBT.

**Jadvalning kalit elementlari – regular key – ключ** – bu maydon (oddiiy kalit) yoki sarli ifoda bo'lib, u boshqa maydonlarning qiymatidan tashkil topadi (tarkibli kalit). Uning yordamida boshqa maydonlarning qiymatini, jadvalning bitta yoki bir qancha yozuvlari uchun aniqlash mumkin. Amaniyotda, kalitdan foydalanish uchun maxsus indekslar, ya'ni kalitlarning tartiblangan qiymatlari bo'yicha xizmatchi ma'lumotlar yaratiladi. Relatsion nazariya va konseptual modelda «kalit» tushunchasi munosabat yoki mohiyat atributlari uchun ishlataladi.

**Joriy holat indikatori – currency indicator – индикатор текущего состояния** – BBBTda ishlataladigan, BBdagi yozuvning joyini ko'rsatuvchi indikator.

**Kesilgan alomat – short attribute – усеченный атрибут** – BBBTda ishlataladigan, berilganlar modelidagi, berilganlar alomatining qismi. Kiritilgan qiymat domenga yoki shu ustundagi qiymatlar to'plamining mumkin bo'lgan elementiga tegishli bo'lishi kerak, degan ma'noni anglatadi.

**Ko'p ma'noli bog'liqlik – multivalue dependency – многозначная зависимость** – R munosabatdagi A atributning har bir elementiga shu munosabatdagi V atributning bittadan ortiq elementining mos kelishi.

**Ko'p qiymatli atribut – multi-value attribute – многозначный атрибут** – ma'lum bir turdag'i mohiyat nusxasining bir nechta qiymatini saqlovchi atribut tushuniladi. *Semantik obyektda ko'p qiymatli atribut deganda, maksimal kardinalligi birdan katta bo'lgan atribut tushuniladi.*

**Kodd modeli – Codd model – модель Кодда** – berilganlar relations modeliting munosabatlari ustida turli amallar bajarishga mo'ljallangan matematik asbob bo'lib, birlashtirish, kesishish, ayirish, dekart ko'paytmasi, nusxalash, munosabatlarni tutashtirish va ajratish amallarini o'z ichiga oladi. Model 1970-yilda E.F. Codd tomonidan taklif qilingan va juda katta ahamiyatga ega. Hamma relations BBBT shu model asosida ishlaydi.

**Kompozit identifikator – composite identifier – композитный идентификатор** – bir nechta atributlardan tuzilgan identifikator.

**Kompozit obyekt – composite object – композитный объект** – bu semantik obyekt bo'lib, obyekt atributi bo'lmagan, bitta yoki bir nechta ko'p qiymatli, oddiy yoki guruhli atributlardan iborat.

**Konseptual chizma – conceptual schema – концептуальная схема** – BBBT tavsiflashning, uning hamma ilovalari uchun yagona tarzda aniqlovchi chizmasi bo'lib, bunda berilganlardan soydalanish, keltirilgan BBBT dagi berilganlarni saqlash muhitiga va berilganlarga murojaat qilish uslublariga bog'liq bo'lmaydi.

**Konseptual daraja – conceptual layer – концептуальный уровень** – berilganlar banki arxitekturasida tashqi va ichki qatlam orasidagi BBBT atamalaridagi ichki darajaga bog'liq bo'lmagan predmet soha modeli. Konseptual modelda esa, birinchi darajali kalit

**Konseptual daraja – conceptual level – концептуальный уровень** – berilganlarning modellashtirishda tashqi darajadagi berilganlarni ichki darajaga akslantirish va ularning bir-biriga nisbatan mustaqilligini ta'minlashga xizmat qiluvchi oraliq daraja.

**Konseptual modellash – conceptual simulation – концептуальное моделирование** – BBBT atamalarida, berilganlar modelini yaratish va optimallash.

**Korrelatsiya o'zgaruvchisi – correlation variable – переменная корреляции** – BBBT so'rovlar tilida, munosabatlar hisobiga asoslangan munosabatlar satr bo'lib, unga boshqa so'rovlar bloklaridan murojaat qilish mumkin, sql va sequel tillarida ishlataladi.

**Kuchli mohiyat – strong entity – сильная сущность** – mavjudligi boshqa mohiyat turiga bog'liq bo'lмаган mohiyat. Kuchli mohiyat *ajdod (ega-mohiyat yoki dominant) mohiyat* deb yuritiladi.

**Majmuani tanlash – set selection – выбор набора** – tarmoqlangan BBBT ega yozuv yoki a'zo yozuvning izlanayotgan nusxasi joylashgan, majmuasi nusxasini izlash amali.

**Mantiqiy bog'lash – logical binding – логическая привязка** – BBBTda berilganlarni tashqi darajadan (qism chizma) konseptual darajaga (chizma) o'tkazish.

**Mantiqiy bog'liqsizlik – logical independence – логическая независимость** – BBBTda mantiqiy darajadagi aniqlangan berilganlarning APlarga bog'liq emasligi.

**Mantiqiy kalit – logical key – логический ключ** – munosabatdagi ustunni noyob elementlarini tasvirlovchi.

**Metaberilganlar – metadata – метаданные** – berilganlar to'g'risidagi ma'lumot. BBida saqlanayotgan berilganlar xususidagi axborot; berilganlar tavsisi va bog'lanishlar jadvali, adresli jadvallar va h.k. BBBT metaberilganlarni BB bilan ishlash jarayonida foydalanadi. Mohiyat atributlarning minimal termasi bo'lib, mohiyatlarning nusxalarini bir qiymatda belgilaydi.

**Mohiyat-aloqa diagrammalar / ER-diagrammalar – entity-relationship diagrams/ER-diagrams – диаграммы сущность-связь/ER-диаграммы** – mohiyatlarni ular orasidagi aloqalar hamda aloqalarning maksimal kardinalligi ko'rsatilgan holda maxsus chizma tarzida ifodalash.

**Munosabat – relation – отношение** – relyasion BBning nazariyasida jadvalning sininimi hisoblanib, unda satrlar *kortej* deb, ustunlar esa *atribut* deb nomlanadi.

**Munosabat – relation – отношение** – relatsion algebraning amallaridagi operandlar bo'lmish, kortejlar (satr) va alomatlar (ustunlar)dan iborat, berilganlar relatsion modelining jadvali.

**Munosabat alomati – relation attribute – атрибут отношения** – berilganlarning relatsion modelida, jadvalning munosabat ustuni. Har bir alomat o'z nomiga ega, alomatlarning aniqlanish sohasi mos domendantan iborat.

**Munosabat arligi – relation arity – ариность отношения** – berilganlarning relatsion modelidagi munosabatlarda alomatlar soni n (jadval ustunlari soni). Unar ( $n=1$ ), binar ( $n=2$ ), ternar ( $n=3$ ), n ar munosabatlarga ajratishadi.

**Munosabat chizmasi – relational schema – чизма отношения –** berilganlarning relatsion modelidagi munosabat alomatlarining ketma ketligi;  $r(A_1, A_2, \dots, A_n)$  bu erda  $r$  munosabat nomi,  $A_i$  alomat nomi,  $n$  munosabatdagi alomatlar soni.

**Munosabat ifodasi – relation expression – выражение отношения –** munosabat amali bilan bog'langan, ikkita arifmetik ifodadan tuzilgan, ifoda.

**Munosabat kaliti – relation key – ключ отношения –** asosiy kalit sifatida ishlataladigan, munosabatlarning bir yoki bir nechta alomati.

**Munosabat turi – relation type – тип отношения –** bir xil domenlar majmuasidan iborat munosabatlari to'plami. Munosabatda bir xildagi kortejlar bo'lmaydi; kortejlar tartiblanmagan; atributlar tartiblanmagan va turli xilda noimlangan; barcha atributlarning qiymatlari bo'linmas.

**Munosabatlarni majmuasi – relation set – набор отношений –** relatsion BBning tarkibi.

**Munosabatlarni obyekti – relation concatenation – объект отношений –** berilganlarning relatsion modelidagi ikkita munosabat ustidan bajariladigan nazariy to'plamli amal bo'lib, uning natijasi bиринчи va ikkinchi munosabatlardagi kortejlardan iborat yangi, o'sha ardag'i munosabat bo'ladi.

**Munosabatlarni bo'lish – relation division – деление отношения –** bo'linuvchi munosabat  $A(x,z)$  ustidan bo'luvchi munosabat  $V(z)$  bajara-digan, relatsion algebraning maxsus amali. Bu erda  $x$  va  $z$  alomatlar qism to'plami. Amal natijasida yangi  $S(z)$  munosabat hosil bo'lib, u bo'linuvchi munosabat kortejlardan iborat bo'ladi va undagi  $z$  ning qiymatlari  $z$  bo'luchining qiymatlari bilan ustma-ust tushadi.

**Munosabatlarni bog'lash – relation binding – связывание отношений –** berilganlarning relatsion modelidagi munosabatlari orasidagi aloqani, aloqa alomatlarini kiritish yo'li bilan o'matish.

**Munosabatlarning kesishishi – relation intersection – пересечение отношений –** relatsion algebraning ikkita bir xil alomatlar majmuasidan iborat munosabatlari ustidagi amallar bo'lib, uning natijasi kortejlari ikkala boshlang'ich munosabatga tegishli yangi munosabatdir.

**Munosabatlarni kengaytirish – relation extension – расширение отношения –** berilganlarning relatsion modelidagi munosabatga yangi alomatlarini qo'shish.

**Muqobil kalit – alternative key – альтернативный ключ –** birlamchi kalit sifatida tanlanmagan nomzod kalit.

**Nisbiy adres – relative address – относительный адрес –** asos adres qiymatiga nisbatan ayirmani ifodalovchi adres; asos adresiga nisbatan berilgan adres. Absolyut, nisbiy va bazali adres yordamida aniqlanadi; xotira sohasining boshlanishiga nisbatan xotira sohasining yacheyka nomeri.

**Nisbiy adreslash – relative addressing – относительная адресация –** komandaning adres qismi nisbiy adresni saqllovchi, adreslash usuli.

**Nisbiy berilganlar – relative data – относительные данные –** display uchun programmalardagi, elektron nurning joriy qiymatiga nisbatan yangi qiymati.

**Nisbiy kodlash – relative coding – относительное кодирование –** nisbiy adresli mashina komandalari ishlataligining kodlash usuli.

**Nisbiy vaqt – relative time – относительное время –** vaqtning ikkita absolyut qiymatlari orasidagi farq.

**Nisbiy xato – relative error – относительная ошибка –** xatolikning absolyut kattaligining parametrning haqiqiy qiymatiga munosabati.

**Nisbiy yo'l – relative path – относительный путь –** berilgan fayl, hujjat yoki obyektga nisbatan, boshqa fayl, hujjat yoki obyektning joyini aniqlovchi, nisbiy yo'l.

**Nomzod kalit – candidate key – потенциальный ключ –** berilgan munosabatning superkaliti bo'lgan, lekin qism to'plami bo'lmasigan superkalit.

**Normallashtirish – normalization – нормализация –** munosabatlarning birlamchi kalitlari va mavjud aloqalar asosida, ularni tahlil qilishning rasmiy usuli.

**Null – null –null –** jadvalning maydonining qiymati yo'qligini bildiradi. Bunday qiymatni jadvalni ma'lum maydonlariga berish mumkin.

**Obyekt – object – объект –** axborotlar tizimining elementi bo'lib, aniq atributlarga (properties) ega bo'ladi va ma'lum bir tarzda tashqi hodisalarga (events) e'tibor qaratadi.

**Obyekt – object – объект –** elementlari xususidagi axborotlarni saqllovchi.

**Obyekt bo'lmasigan atribut – non-object attribute – не объектный атрибут –** bu oddiy yoki guruhli atribut.

**Obyekt identifikatori – object identifier – идентификатор объекта –** bu bir yoki bir nechta obyekt atributi bo'lib, ular yordamida obyektlarning nusxalari identifikasiya qilinadi.

**Oddiy atribut – simple attributes – простой атрибут –** bo'linmas atribut.

**Oddiy obyekt – simple object – простой объект –** bu semantik obyekt bo'lib, faqat bir qiymatli, oddiy yoki guruhli atributlardan iborat.

**Ostki turdag'i mohiyatlar – subtypes entity – подтипы сущности –** mohiyatlardagi zarur bo'lmasigan atributlardan tashkil topgan mohiyat.

**Predmet soha – subject domain – предметная область –** haqiqatda faoliyat yurituvchi aniq bir tizimning, mustaqil faoliyat yurituvchi qaysidir bir qismi. Predmet soha to'liq holatda mamlikatning yoki ittifoq davlatlar iqtisodiyotini tavsiflashi mumkin, lekin amaliyotda axborotlar tizimi uchun

korxona yoki korporatsiya miqyosidagi predmet soha ko'proq ahamiyatga ega.

**Qatlamlı arxitektura – layer architecture – слойная архитектура** – berilganlarning mantiqiy va fizik bog'liqsizligini ta'minlovchi, ko'r qatlamlı BBBT arxitekturasi. Arxitekturada tashqi, konseptual va ichki darajalar, relatsion BBBT arxitekturasida qo'shimcha global konseptual darajalar ko'zda tutiladi.

**QBE (namuna bo'yicha so'rov) tili – QBE (Query-by-Example) – язык**

**QBE (запрос по образцу) – so'rovlar andazasini qo'llashga asoslangan berilganlar bazasiga kirishni tashkillashtirishning namoyishli usuli.**

**Rekursiv aloqalar – recursive relationships – рекурсивные связи** – bitta sinfga tegishli mohiyatlar orasidagi aloqalar.

**Relatsion algebra (munosabatlar algebrası) – relation algebra – реляционная алгебра (алгебра отношений)** – berilganlarning relatsion modeli ustida murakkab amallarni bajaruvchi matematik usul bo'lib, u munosabatlar ustidan bajariladigan birlashtirish, kesishish, ayirish, dekart ko'paytmasi, sarkillash, proyeksiyalash, tutashtirish va bo'lish amallarini o'z ichiga oladi. E.F. Kodd tomonidan taklif qilingan.

**Relatsion amal – relation operation – операция реляционной алгебры** – munosabat yoki berilganlarning relatsion modelining munosabatlari ustidan bajariladigan, amal. Amalning natijasi, hamma vaqt yangi munosabat bo'ladi. Relatsion algebrani nazariy to'plamli va maxsus amallarda ajratishadi.

**Relatsion berilganlar bazasi – relation database – реляционная база данных (РБД)** - BBning asosiy zamonaviy turidir. RBB jadvallardan iborat bo'lib, bu jadvallar bir-biri bilan kalitli maydon orqali bog'lanishi mumkin.

**Relatsion berilganlar bazasini boshqarish tizimi – relation database management system – реляционная система управления базами данных** – relatsion BBni yaratish, kuzatish va foydalanishga mo'ljallangan, BBBT.

**Relatsion hisob – relation calculus – реляционное исчисление** – yangi munosabatlarni boshqa munosabatlar orqali tasvirlash uchun berilganlarning relatsion modelida qo'llaniladigan, predikatlar hisobi.

**Relatsion model – relation model – реляционная модель** – qarang Kodd modeli.

**Relatsion til – relation language – реляционный язык** – relatsion algebra amallarini ishlatalishga asoslangan, berilganlarga murakkab ishlov berish tili.

**Relatsion yondashish – relation approach – реляционный подход** – BBni loyihalashda relatsion nazariyani qo'llash.

**Saqlangan protseduralar – stored procedures – сохраненные процедуры** – berilganlar saqlanadigan, hamda shu bazadagi ma'lumotlar bilan ma'lum amallarni bajarish uchun xizmat qiladigan programmalarning modullari hisoblanadi.

**Semantik butunlik – semantic integrity – семантическая целостность** – satrga kiritilgan berilganlar elementi shu satr uchun mumkin bo'lgan qiymatga ega ekanligini kafolatlash.

**Semantik obyekt – semantic object – семантический объект** – obyekting foydalanuvchi tomonidan talqin qilinadigan berilganlar xususidagi tushunchalari ma'nesini qaysidir darajada modellashtirishni nazarda tutadi. U atributlarning nomlangan majmuasi bo'lib, yetarli darajada alohida nodirlikni tavsiflaydi. Semantik obyektda oddiy atributlar bitta elementdan iborat bo'ladi. Oddiy atributning domeni fizik va semantik tavsifdan tuziladi.

**Semantik tavsif – semantic description – семантический дескриптор** – funksiyani yoki shu atributning tayinlanishini ko'rsatadi, shu atribut boshqa atributlardan fizik tavsifi bilan farqlanadi.

**Siljish – relocability – перемещаемость** – programmani yuklashning tashqi xotiranining ixtiyoriy ruxsat berilgan joyiga yuklash va programmani xotiraning bir qismidan ikkinchi qismiga siljish imkoniyati.

**Siljuvchi adres – relocatable address – перемещаемый адрес** – qiymati shu adresni saqlagan mashina programmasining siljishi vaqtida o'rnatiladigan, adres; programmaning siljish vaqtida o'zgartirilishi kerak bo'lgan adres.

**So'rov – enquiry – запрос** – BBBT yoki AYITga oddiy murojaat.

**Sozlab yuklovchi – relocating loader – настраивавший загрузчик** – obyekt yoki yuklanuvchi modul aniq adres bo'yicha joylashganda, adreslarini o'zgartiruvchi yuklovchi.

**Sozlanuvchi adres – relocatable address – настраиваемый адрес** – programmaning tezkor xotirada aniq joylashishiga nisbatan yuklash vaqtida sozlanuvchi, yuklanuvchi modul adresi.

**SQL (tuzilishli so'rovlar tili) – SQL (Structured Query Language) – язык SQL** – tuzilishli so'rovlar tili bo'lib, relatsion turdag'i BBBT bilan ishlashda BBga so'rovlarni yaratish, yangilash va boshqarish imkoniyatini beradi.

**Statistik bog'liqlik – static binding – статистическая привязка** – BBB tizimida berilganlarni bir tasvirdan boshqasiga AP bajarilgunga qadar, almashtirish. Masalan, konseptual darajadan fizik darajaga almashtirish.

**Superkalit – superkey – суперключ** – berilgan munosabatning kortejini yagona tarza aniqlovchi atribut yoki atributlar majmuasi.

**Tasavvur – view – представление** – relatsion modelda foydalanuvchining hamma tashqi modelini emas, balki u ishlataladigan virtual yoki hosilaviy munosabatni bildiradi, ya'ni tuungsabat aslida mavjud bo'lnaydi, lekin u dinamik tarzda bitta yoki bir nechta berilganlar bazasidagi mavjud munosabatlar asosida hosil qilinadi.

**Taqsimlangan BB – Distributed DataBase (DDB) – распределенная база данных** – bu taqsimlangan kompyuter tarmog'ida mantiqan o'zarbo'lgan berilganlar bazalari majmuasi.

**Taqsimlangan berilganlar banki – distributed databank – распределенный банк данных** – hududiy tarqoq mahalliy berilganlar bankining tizimi bo'lib, ular hisoblash tarmog'ining vositalari orqali bog'langan va yagona boshqarish orqali faoliyat yuritadi. Bunday tizimning asosini taqsimlangan BB va tarqatilgan BBBT tashkil qiladi.

**Taqsimlangan berilganlar bazasini boshqarish tizimi – distributed database management system – система управления распределенными базами данных** – tarkibiy qismlari mahalliy BB hisoblash tarmog'ining turli tugunlarida joylashgan, ya'ni taqsimlangan BBga foydalanuvchilarning murojaati uchun mo'ljalangan BBBT.

**Tarkibiy alomat – compound attribute – составной атрибут** – BBBT ishlataladigan berilganlar modelidagi ikki yoki undan ortiq soddarorq alomatlardan iborat alomat.

**Tarkibli atribut – group attributes – сложный атрибут** – bo'linuvchi atribut.

**Tarkibli obyekt – compound object – составной объект** – kamida bitta obyekt atributidan iborat.

**Tarqatiladigan naql – release version – распространяемая версия** – otladka va tashxis funksiyalari kiritilmagan, tashqi dizaynga ega bo'lgan, programmaning komplikatsiya qilingan naql.

**Tashqi daraja – external level – внешний уровень** – berilganlarning modellashtirishda foydalanuvchilar tasavvuridagi berilganlar darajasi.

**Tashqi kalit – foreign key – внешний ключ** – bo'yusunuvchi (tashqi, avlod) jadvalarning kalit elementi bo'lib, uning qiymatlari asosiy (ajded) jadvaldagagi birinchi darajali kalitning qiymatlari bilan ustma-ust tushadi.

**Tashqi kalit – foreign key – внешний ключ** – munosabatdagi atribut yoki atributlar to'plami boshqa munosabatdagi (yoki o'zidagi) nomzod kalitga mos kelgan holat.

**Tashqi model – external model – внешняя модель** – BBBT arxitekturasida, foydalanuvchining BB (BB qism chizmasi va uning tavsisi) xususidagi tasavvurini akslantiruvchi tashqi darajadagi berilganlar modeli (oliv darajadagi qatlami).

**Test (namoyishli) berilganlar bazasi – test database – тестовая (демонстрационная) база данных – nisbatan kichik hajmidagi, foydalanuvchilarga o'rganish imkoniyatlarini ko'rsatish va BBBT baholash uchun qo'llaniladigan, BB.**

**Tiklash – roll forward – восстановление – BBBTdagи, tizimning katta inkorlardan keyingi tiklanish qobiliyati.**

**Tiklash programmasi – recovery program – программа восстановления – BBBTning tarkibiy qismiga kiruvchi va BBni tiklashni ta'minlovchi, programma.**

**Tizim – system – система – o'zaro va tashqi atrof bilan harakatlanuvchi majmua.**

**To'rtinchи normal shakl (4NSh) – fourth normal form (4NF) – четвертая нормальная форма (4НФ) – munosabat BKNSh bo'lsa va ko'rma'noli bog'liqliklarga ega bo'lmagan holat.**

**Tranzaksiya – transaction – транзакция – 1. BBni, bir qarama-qarshi bo'lmagan holatdan ikkinchisiga o'tkazuvchi, kirish axboroti; BBni o'zgartirish so'rovi. 2. Muloqot tizimlaridagi, foydalanuvchida berilganlar qismini (axborot, so'rov) qabul qilish, unga ishlov berish va javob ma'lumotni chiqarish.**

**Tranzaksiyalar jurnali – transaction log – журнал транзакций – BBBT vositaari bilan avtomatik tarzda qo'llanuvchi, tushayotgan tranzaksiyalarni hisobga oluvchi jurnal.**

**Tranzaksiyalarni boshqarish – transaction control – управление транзакциями – parallel jarayonlarni boshqarish va berilganlarni tiklashni boshqarishdan iborat, BBBTning funksiyasi.**

**Triggerlar – triggers – триггеры – bu saqlangan protseduralar bo'lib, asosiy kalit o'zgartirilganda (pog'onali (kaskad) ravishda ma'lumotlarni o'zgartirish), asosiy jadvaldan ma'lumotlar o'chirilganda (pog'onali ravishda avlod jadvallardan ma'lumotlarni o'chirish) va avlod jadvalda ma'lumotlar o'zgartirilganda yoki qo'shilganda, berilganlarning havolali butunligi shartlarini ta'minlashdan iborat.**

**Tuzilishli so'rovlar tili – structured query language (SQL) – язык структурированных запросов – bu universal til bo'lib, relatsion turdag'i BBBT Oracle, Microsoft SQL Server, Informix, Access, DB2 va h.k. bilan ishlashda BBga so'rovlarni yaratish, yangilash va boshqarish imkoniyatini beradi. Ixtiyoriy relatsion BB SQL 92 standartini qo'llab-quvvatlaydi. SQL tili SQL operatorlari va bir qator joylangan funksiyalardan iborat.**

**Uchinchi normal shakl (3NSh) – third normal form (3NF) – третья нормальная форма (3НФ) – munosabat 2NShga keltirilgan va tranzitiv bog'liqliklari bo'lmagan holat.**

**Virtual berilganlar bazasi – virtual database – виртуальная база данных** – tasavvurdagi berilganlarning tasvirlanishi bo'lib, unga ixtiyoriy BBBT har bir integratsiya bo'ladigan BBni aylantirish mumkin.

**Halqa yo'lning nisbiy adresi – relation track address – относительный адрес дорожки** – berilganlar to'plamiga ajratilgan diskdagi halqa yo'lning boshlanishiga nisbatan, halqa yo'l nomeri.

**Zaif mohiyat – weak entity – слабая сущность** – mayjudligi boshqa mohiyatlarga mantiqan bog'liq bo'lgan mohiyat. Zaif mohiyatlar voris (bog'liq yoki tobe) mohiyat deb yuritiladi.

## FOYDALANILGAN ADABIYOTLAR

1. Allen G. Taylor, Alex Kriegel, Boris M. Trukhnov. Introduction to Database Management. – wiley, 2007, 504 c.
2. ANSI X3.135-1992, American National Standard for Information Systems - Database Language - SQL, November, 1992.
3. Astrahan M.M., System R: A Relational Approach to Data Base Management //ACM Transactions on Data Base Systems. - 1976. - VI, 97, June.
4. Barker R. CASE\*Method. Entity-Relationship Modelling. Copyright Oracle Corporation UK Limited, Addison-Wesley Publishing Co., 1990.
5. Boyce R.F., Chamberlin D.D., King W.F., Hammer M.M. Specifying Queries as Relational Expressions: The SQUARE Data Sublanguage //Communications ACM. - V.18, November, 1975, p.621-628.
6. Chamberlin D.D., Astrahan M.M., Eswaran K.P., Grifts P.P., Lorie R.A., Mel D.V., Reisner P., Veyd B.V. SEQUEL2: A unified approach to data definition, manipulation and control. IBM J. Research and Development, 20(6), 1976, p. 560-575.
7. Chamberlin D.D., Gray J.N., Traiger L.L. Views, Authorization and Locking in a Relational Data Base System //Proceedings of AFIPS National Computer Conference, Anaheim, CA, May. - 1975.
8. Chamberlin D.D., Raymond F.B. SEQUEL: A Structured English Query Language. //Proc. ACM-SIGMOD - Workshop, Ann Arbor, Michigan, May, 1974.
9. Chen P.P. The Entity-Relationship Model - Towards a Unified View of Data. ACM Transactions on Database Systems. январь 1976, c. 9-36.
10. Codd E.F. A data base sublanguage founded on the relational calculus //Proc. ACM-SIGFIDET - Workshop, San Diego, Calif., Nov. 1971, p.35-68.
11. Codd E.F. Extending the Database Relation Model to Capture More Meaning. //ACM Transaction on Database Systems. 1979.- V.4, №4. - p.397-434.
12. Codd E.F. Further Normalization of the Data base Relational Model //Data Base Systems.- N.J.: Prentice-Hall, 1972. - p.33-64.
13. Codd E.F. Missing information (applicable and inapplicable) in relational databases. ACM SIGMOD Record, 1986, Vol. 15, №4.

14. Codd E.F. More commentary on missing information in relational databases. ACM SIGMOD Record, 1987, Vol. 16, №1.
15. Codd E.F. Normalized Data Base Structure: A Brief Tutorial //Proc. of 1971 ACM-SIGFIDET Workshop on Data Description, Access and Control. - N.Y.: ACM., 1971, p.1-17.
16. Codd E.F. Recent investigations in relational data base systems //Proc. IFIP Congress, - North-Holland Pub. Co., Amsterdam, 1974, p.1017-1021.
17. Codd E.F. Relation Model of Data for Large Shared Data Banks //Comm. ACM. - 1970. - V.13, № 6, p.377-383.
18. Codd E.F. The Relational Model for Database Management Version 2, Reading, MA: Addison-Wesley, 1990
19. Codd E.F., Rustin R., ed. Relational completeness of data base sublanguages. In Data Base Systems, Courant Comput. Sci. Symp 6th, Englewood Cliffs, NJ: Prentice-Hall, 1972, p. 65-98
20. DATARUN Concepts. Computer Systems Advisers Research Ltd., 1994.
21. Donahoo Michael J., Gregory D. Speegle. SQL - Practical Guide for Developers - Morgan Kaufmann Publishers, 2005, 266 c.
22. Eswaran K.P. Chamberlin D.D. Functional specifications of a subsystem for data base integrity //Proc. Very Large Data Base Conf., Framingham, Mass., Sept. - 1975. - p. 48-68.
23. Eswaran K.P., Gray J.N., Lorie R.A., Traiger I.L. The Notions of Consistency and Predicate Locks in a Data Base System //CACM. - 1976. - V.19, №11.
24. Fagin R. Multivalued dependencies and a new normal form for relational databases. ACM Trans. Database Systems, 1977, Vol. 2, №3.
25. Fagin R. Normal forms and relational database operators. In Proc. ACM SIGMOD Int. Conf. on Management of Data, 1979, p.153-160.
26. Fagin R.A. Normal Form for Relational Databases that is based on domains and key //ACM Transactions on Database Systems, 1981, V.6, №3, p.387-415.
27. Gray J., Lorie R., Putzolu G., Traiger I. Granularity of Locks and Degrees of Consistency in a Shared Data Base //in Readings in Database Systems, Second Edition, Chapter 3, Michael Stonebraker, Ed., Morgan Kaufmann, 1994.
28. Guralnik D.B., ed., Webster's/ New World of the American Language. William Collins and World Publishing Go., New York, 1974
29. Halpin T. Selected Readings on Database Technologies and Applications - IGI Global, 2009, 565 c.

30. Heath I.J. Unacceptable File Operations in Relational Database //Proc. 1971 ACM SIGFIDET Workshop on Data Description, Access and Control. - San Diego, Calif, 1971.
31. Hector Garcia-Molina, Jeffrey D. Ullman, J. Widom. Database Systems: The Complete. Book, 2nd Edition. Prentice Hall, - 2009, 1224 c.
32. Held G.D., Stonebraker M.R., Wong E. INGRES: A Relational Data Base System //Proceedings of AFIPS National Computer Conference, Anaheim, CA, May, 1975.
33. <http://athena.troux.com/akmii/Default.aspx?WebID=249>
34. <http://www.guest.com>
35. <http://www.idef.com>
36. <http://www.idef.ru>
37. <http://www.intuit.ru>
38. <http://www.mysql.com/doc>
39. <http://www.mysql.com/documentation>
40. <http://www.omg.org>
41. <http://www.omg.org/cgi-bin/doc?dtc/2006-02-01>
42. <http://www.vpg.ru/main.mhtml?PubID=6>
43. <http://www.williamspublishing.com>
44. Kent W. A Primer of Normal Forms. IBM Technical Report TR, 02.600, December, 1973.
45. Kriegel A. Discovering SQL: A Hands-On Guide for Beginners, - Wrox, 2011
46. Lacroix M., Pirotte A. Domain-oriented relational languages. In Proc. 3rd Int. Conf, Very Large Data Bases, 1977, p.370-378.
47. Li Yan. Advanced Database Query Systems: Techniques, Applications and Technologies. - IGI Global, - 2011, 350 c.
48. Mark L. Gillenson. Fundamentals of Database Management Systems, 2nd Edition, Wiley, - 2011, 416 c.
49. Reisner P., Boyce R.F., Chamberlin D.D. Human Factors Evaluation of Two Data Base Query Languages: SQUARE and SEQUEL //Proceedings of AFIPS National Computer Conference, Anaheim, CA, May, 1975.
50. Steve Conger. Hands-On Database. Prentice Hall, - 2011, 216 c.
51. Teorey T.J., Nadeau T., Lightstone S. Database Modeling and Design: Logical Design (4 edition), Morgan Kaufmann, - 2005, 296 c.
52. Thomas A. Bruce, Designing Quality Databases with IDEF1X Information Models. New York: Dorset House Publishing, 1992.
53. Xammer M., Leod D. Database Description with SDM: A Semantic Database Model., ACM Transactions on Database Systems, сентябрь 1981, c. 351-386

54. Zloof M.M. Query-By-Example //Proceedings of AFIPS National Computer Conference, Anaheim, CA, May. - 1975.
55. Zloof M.M. Query-By-Example: A database language. IBM Systems J., 16(4), 1977, p.324-343
56. Ахмадеев И.А., Хайруллин А.Х. Базы данных. - Камский государственный политехнический институт, Кама, 2004, 237 с.
57. Бенькович Е.С., Колесов Ю.Б., Сениченков Ю.Б. Практическое моделирование динамических систем - СПб.: БХВ-Петербург, 2002
58. Боуман Д., Эмерсон С., Дарновски М. Практическое руководство по SQL. - Киев: Диалектика, 1997.
59. Бураков П.В., Петров В.Ю. Введение в системы баз данных. Учебное пособие. - Санкт-Петербург, СПбГУ ИТМО, 2010, 129 с.
60. Бэкус Дж.В. и др. Сообщение об алгоритмическом языке АЛГОЛ-60. ЖВМ и МФ, №1, 2, 1961.
61. Васкевич Д. Стратегии клиент/сервер. - Киев: Диалектика, 1997.
62. Грабер М. Справочное руководство по SQL. - М.: Лори, 1997. - 291 с.
63. Гради Буч. Язык UML. Руководство пользователя, СПб., Питер, 2002
64. Григорьев Ю.А., Ревунков Г.И. Банки данных. - М.: МГТУ им. Баумана, 2002. - 320 с.
65. Грофф Д.Р., Вайнберг П.Н. SQL: Полное руководство. - Киев: BMV, «Ирина», 2001. - 816 с.
66. Дейт К. Введение в системы баз данных //8-издание. - Киев: Диалектика, 2002. - 784 с.
67. Дейт К. Руководство по реляционной СУБД DB2. - М.: Финансы и статистика, 1988. - 320 с.
68. Дунаев В.В. Базы данных. Язык SQL, СПб.: БХВ-Петербург, - 2006, 288 с.
69. Кайт Т. Oracle для профессионалов. Пер. с англ./ - К.: ООО «ТИД DC», 2003, кн. 1 – 672 с.,
70. Кайт Т. Oracle для профессионалов. Пер. с англ./ - К.: ООО «ТИД DC», 2003, кн. 2 – 848 с.
71. Карпов Б. MS Access 2000, Санкт-Петербург – Питер, 2000, 416 с.
72. Кен Х. Профессиональное руководство по SQL Server: структура и реализация - М.: Издательский дом «Вильямс», 2006 -1056 С.
73. Кен Х. Профессиональное руководство по SQL Server: хранимые процедуры, XML, HTML - СПб. Издательский дом «Питер», 2005 – 620 С.
74. Кириллов В.В., Громов Г.Ю. Структуризованный язык запросов (SQL). Учебное пособие. –Санкт-Петербургский ГТУ, 2004.

75. Клайн К. Справочник по SQL, -М.: КУДИЦ-ОБРАЗ, 2006, 832 с.
76. Когаловский М.Р. Энциклопедия технологий баз данных. - М.: Финансы и статистика, 2003. - 800 с.
77. Колесов Ю.Б., Сениченков Ю.Б. Визуальное моделирование сложных динамических систем – СПб.: Издательство Мир и Семья & Интерлайн, 2000
78. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 3-е издание.: Пер. с англ. - М.: Издательский дом «Вильямс», 2003. - 1440 с.
79. Крёнке Д. Теория и практика построения баз данных. 8-е издание.: Пер. с англ. - М, С-П., Минск: Издательский дом «Питер», 2003. - 800 с.
80. Кузнецов С.Д. Основы баз данных. -М.: Интернет-Университет информационных технологий; БИНОМ,- 2007, 484 с.
81. Кузнецов С.Д. Введение в модель данных SQL – ИНТУИТ, БИНОМ,- 2005, 203 с.
82. Кузнецов С.Д. Неопределенная информация и трехзначная логика //СУБД. - 1997. - №5. - с.65-67.
83. Маклаков С.В. Создание информационных систем с All Fusion Modeling Suite 4.1 (Bpwin 4.1) - М.: ДИАЛОГ – МИФИ, 2002. – 224с.
84. Мартин Дж. Организация баз данных в вычислительных системах. –М., Мир, 1980. 664 с
85. Мейер М. Теория реляционных баз данных. - М.: Мир, 1987. - 608 с.
86. Мендельсон Э. Введение в математическую логику. – 3-е изд. – М.: Наука, 1984.
87. Михеева В.Д. Харитонова И.А., MS Access 2002. Наиболее полное руководство. ВНУ-Санкт-Петербург. 2003.
88. Мюллер Р. Базы данных и UML. Проектирование. – Лори, 2002, 420 с.
89. Нагао М., Катаяма Т., Уэмуро С. Структуры и базы данных. - М.: Мир, 1986, 197 с.
90. Олле Т.В. Предложения КОДАСИЛ по управлению базами данных. –М.: Финансы и статистика, 1981, 286 с.
91. Панащук С.А. Разработка информационных систем с использованием CASE-системы Silverstripe. «СУБД», 1995, №3.
92. Пирогов В. MS SQL Server 2000: управление и программирование. //СПб.: БХВ-Петербург, 2005.
93. Полякова Л.Н. Основы SQL. Интернет-университет информационных технологий – INTUIT.ru, БИНОМ. Лаборатория знаний, 2007, 224с.

94. Пушкиков А.Ю. Введение в системы управления базами данных. Учебное пособие. Башкирский Государственный Университет, 2002, 184 с.
95. Ролланд Ф. Д. Основные концепции баз данных, - М.: Издательский дом «Вильямс», 2002, 256 с.
96. Словарь иностранных слов. 18-е изд., стер. -М.: «Русский язык», 1989, 624 с.
97. Стасышин В.М. Доступ к базам данных: Учеб. пособие. - Новосибирск: НГТУ, 2001. - 94 с.
98. Тиори Т., Фрай Д. Проектирование структур баз данных. В 2 кн., - М.: Мир, 1985. Кн. 1. - 287 с.; Кн. 2. - 320 с.
99. Токмаков Г.П. Базы данных: Учеб. пособие. - Ульяновск: УлГТУ, 2010, 193 с.
100. Ульман Д. Основы систем баз данных. - М.: Финансы и статистика, 1983. - 334 с.
101. Ульман Джейфри Д., Уидом Дж. Введение в системы баз данных - М.: Лори, 2000.
102. Уорсли Дж., Дрейк Дж. SQL. - Санкт-Петербург, Питер, 2003. - 496 с.
103. Фаронов В.В. Разработка приложений для баз данных и Интернета - СПб: Питер, 2006 г - 603 с.
104. Форте Б. Освой самостоятельно SQL. - Издательский дом «Вильямс», 2005, 288 с.
105. Фронковяк Дж., Гарсиа М., Уолен Э. Руководство администратора Microsoft SQL Server 7.0. -М.: Русская редакция, 2000.
106. Фуфаев Э.В., Фуфаев Д.Э. Базы данных - М.: Издательский центр «Академия», 2007, 320 с.
107. Хаббард Д. Автоматизированное проектирование баз данных. - М.: Мир, 1984. - 296 с.
108. Ҳакимов М.Ҳ. Инглизча-русча-ўзбекча компьютер илми бўйича изоҳли лутат. -Т.: Университет, 2006, 624 б.
109. Хоменко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных. - Санкт-Петербург, Корона, 2002. - 672 с.
110. Цаленко М.Ш. Моделирование семантики в базах данных. - М.: Наука, 1988.
111. Цикритзис Д., Лоховски Ф. Модели данных. - М.: Финансы и статистика, 1985. - 344 с.

## MUNDARIJA

MUQADDIMA.....	3
KIRISH.....	5
USLUBIY KO'RSATMALAR .....	10

### I- BO'LIM. BERILGANLAR BAZALARI

1- bob. Berilganlar bazasi.....	12
Asosiy tushunchalar .....	12
Berilganlar bazalari konsepsiyalarining evolutsiyasi .....	13
Berilganlar tuzilishining uch turi .....	18
«Uch bosqichli» arxitektura .....	19
Chizmalar .....	22
Nazorat uchun savollar .....	23
2- bob. Obyektlar va atributlar .....	25
Axborotning uch sohasi .....	26
Ikki o'lcovli fayllar .....	27
Kalitlar .....	29
So'rovlar .....	30
Nazorat uchun savollar .....	33
3- bob. Berilganlar modeli.....	34
Chizmalar .....	37
Berilganlar elementlarining o'zaro aloqasi .....	39
Yozuvlar orasidagi aloqalar .....	41
Nazorat uchun savollar .....	44

### II BO'LIM. BERILGANLARNING MODELLARI

4- bob. Berilganlarning iyerarxik modeli .....	46
Daraxtlar .....	47
Muvozanatlashgan va muvozanatlashmagan daraxtlar .....	48
Terma va iyerarxik fayl .....	52
Bir jinsli tuzilishlar .....	53
Bog'lanishlardagi bog'liqliklar .....	53
Nazorat uchun savollar .....	56

<b>5- bob. Berilganlarning tarmoq modeli .....</b>	<b>57</b>
Oddiy va murakkab tarmoq tuzilishlari .....	59
Kesishishdag'i berilganlar .....	61
M:M turidagi aloqalar .....	62
Takrorlashlar .....	63
Halqa .....	64
Tarmoq tuzilishlarni sodda ko'rinishga keltirish .....	65
Nazorat uchun savollar .....	68
<b>6- bob. Relatsion modellarga kirish .....</b>	<b>69</b>
Kodd qoidalari .....	69
Relatsion model tushunchalari .....	71
Muqobil atamalar .....	75
Kalitlar .....	76
Berilganlarning relatsion modellarida chizmalar tasavvuri .....	78
Normal chizmaga qo'shimchalar .....	81
Yo'naltirilgan aloqalar muammozi .....	84
Nazorat uchun savollar .....	87
<b>7- bob. Berilganlarning «mohiyat-aloqa» modeli.....</b>	<b>88</b>
«Mohiyat-aloqa» modeli elementlari .....	88
Binar aloqalarning uch turi .....	91
«Mohiyat-aloqa» diagrammalari .....	94
Zaif va kuchli mohiyatlar .....	95
Bir va ko'p qiymatli atributlar .....	98
Mohiyatlarning ostki va ustki turlari .....	99
Atribut domeni .....	100
Talabalar o'quv loyihasini modellashtirish .....	101
Nazorat uchun savollar .....	105
<b>8- bob. Semantik obyekt model .....</b>	<b>107</b>
Semantik obyektlar .....	107
Atributlar .....	109
Atributning kardinallik soni .....	110
Obyekt nusxalari .....	112
Obyektning identifikatorlari .....	113
Atribut domenlari .....	114
Berilganlarning semantik obyekt modellarini yaratish .....	115
Obyektlar tasnifi .....	119
Obyektlarning turlari .....	123
Semantik obyekt model va «mohiyat-aloqa» modelini taqqoslash .....	129

Nazorat uchun savollar .....	132
<b>9- bob. Axborot tizimlarini modellashтирish .....</b>	<b>133</b>
UML tili .....	133
UML tilidagi «mohiyat-aloha» diagrammalari .....	133
UML tilida mohiyat va aloqalar .....	134
UML tilining asosiy xususiyatlari .....	137
UML tilida modellashга misollar .....	139
UEML tili .....	141
Nazorat uchun savollar .....	143
 <b>III BO'LIM. RELATSION ALGEBRA VA NORMALLASH</b>	
<b>10- bob. Relatsion algebra.....</b>	<b>144</b>
Munosabatlар ustidagi amallar .....	145
Dekart ko'paytmasi amali .....	147
Tanlash amali .....	147
Nusxalash amali .....	150
Bo'lish amali .....	151
Tutashish amali .....	153
Yarim tutashish amali .....	155
Ekvivalentlik bo'yicha tutashish amali .....	156
Θ - tutashish amali .....	157
Tashqi tutashish .....	159
Nazorat uchun savollar .....	162
<b>11- bob. Munosabatlarni normallash .....</b>	<b>164</b>
Normallashtirish bosqichlari .....	165
Birinchi normal shakl .....	165
Funksional bog'lanish tushunchasi .....	167
To'liq funksional bog'liqlik .....	170
Ikkinchi normal shakl .....	171
Uchinchi normal shakl .....	174
Boys-Kodd normal shakli .....	175
To'rtinchi normal shakl .....	177
Besinchi normal shakl .....	180
Nazorat uchun savollar .....	183
<b>12- bob. Relatsion hisoblash.....</b>	<b>185</b>
Kortejlarni relatsion hisoblash .....	186

Mavjudlik va umumiylik kvantorlari .....	187
Ifodalar va formulalar .....	189
Kortejlarini relatsion hisoblashga misollar .....	189
Domenlarni relatsion hisoblash .....	191
Domenlarni relatsion hisoblashga misollar .....	192
Nazorat uchun savollar .....	195

<b>13- bob. Berilganlarning relatsion butunligi .....</b>	<b>197</b>
Bo'sh qiymat va uch ma'noli mantiq .....	198
Mohiyatlar butunligi .....	199
Semantik butunlik .....	201
Havola butunligi .....	202
Butunlikning korporativ chegaralari .....	202
Tasavvur .....	203
Tasavvurlarning tayinlanishi .....	204
Tasavvurlarni yangilash .....	205
Relatsion berilganlar bazasida havola butunligini saqlash shartlari .....	205
Havola butunligini qo'llab-quvvatlashdagi asosiy strategiyalar .....	206
Havola butunligini qo'llab-quvvatlashdagi qo'shimcha strategiyalar .....	207
Nazorat uchun savollar .....	208

#### **IV BO'LIM. RELATSION BERILGANLAR BAZASINI LOYIHALASH USLUBIYATI**

<b>14- bob. Axborot tizimlarida berilganlar bazasi .....</b>	<b>210</b>
Axborotlar tizimini loyihalash negizlari .....	210
Loyihalash uslublari .....	216
Predmet sohani modellashtirish darajalari .....	219
Nazorat uchun savollar .....	223
<b>15- bob. Berilganlar bazasini konseptual loyihalash .....</b>	<b>224</b>
Berilganlarning lokal konseptual modelini yaratish .....	224
Mohiyatlarning turlarini aniqlash .....	225
Aloqalarning turlarini aniqlash .....	227
Atributlarni aniqlash va ularni mohiyatlarning turlari va aloqalar bilan bog'lash .....	229
Atributlarning domenlarini aniqlash .....	231
Nomzod va birlamchi kalit bo'ladigan atributlarni aniqlash .....	231
Modelni ortiqchalik nuqtayi nazaridan tekshirish .....	232

Lokal konseptual modelni aniq foydalanuvchining tranzaksiyalariga mosligini tekshirish.....	234
Berilganlarning lokal konseptual modelini oxirgi foydalanuvchilar bilan muhokama qilish .....	235
Nazorat uchun savollar .....	236
<b>16- bob. Berilganlar bazasini mantiqiy loyihalash.....</b>	<b>237</b>
Relatsion berilganlar bazasini mantiqiy loyihalash usuli.....	238
Foydalanuvchilarning alohidagi tasavvurlari bo'yicha lokal mantiqiy berilganlar modelini yaratish .....	238
Relatsion modelga mos kelmaydigan alohidaliklardan qutilish.....	238
Lokal mantiqiy berilganlar modeli tuzilishi asosida munosabatlar majmuasini aniqlash.....	240
Normallashtirish qoidalarini qo'llagan holda munosabatlarni tekshirish...246	246
Foydalanuvchi tranzaksiyalarini bajarishda munosabatlardan foydalanishni tekshirish .....	246
Butunlik chegaralarini aniqlash .....	247
Berilganlarning lokal mantiqiy modelini foydalanuvchilar bilan muhokama qilish .....	249
Berilganlarning global mantiqiy modelini yaratish va tekshirish.....	249
Berilganlarning lokal mantiqiy modellarini berilganlarning yagona global modeliga jamilash .....	250
Berilganlarning global mantiqiy modelini tekshirish.....	255
Berilganlarning global mantiqiy modelini kelajakda kengaytirish imkoniyatlarini tekshirish .....	255
Berilganlarning global mantiqiy modelini foydalanuvchilar bilan muhokama qilish.....	256
Nazorat uchun savollar .....	256
<b>17- bob. Berilganlar bazasini fizik loyihalash .....</b>	<b>258</b>
Berilganlar bazasini loyihalashning mantiqiy va fizik bosqichlarini taqqoslash.....	258
Global mantiqiy berilganlar modelini BBB tizimi muhitiga ko'chirish....259	259
Assosiy munosabatlarni loyihalash.....	259
Hosilaviy berilganlarni olish usullarini yaratish .....	261
Predmet sohani cheklashni amalga oshirish.....	261
Berilganlar bazasini fizik tasavvurini loyihalash .....	262
Tizim zaxiralari .....	262
Tranzaksiyalar tahlili .....	264
Tranzaksiyalar va munosabatlarni bajarish yo'llari orasidagi mos chizmalarni tayyorlash .....	265

Tranzaksiyalarni bajarishda eng ko'p ishlataladigan munosabatlarni aniqlash .....	266
Berilganlardan foydalanish tahlili .....	266
Faylning tuzilishini tanlash .....	267
Indekslami qo'llashga tavsiyalar .....	271
Nazorat savollari .....	273
<b>18- bob. Tranzaksiyalarni loyihalash .....</b>	<b>274</b>
Tranzaksiylardan foydalanish maqsadi .....	274
Tranzaksiyalar holatlari .....	275
Tranzaksiyalar modeli .....	276
Tranzaksiyalarning parallel bajarishdagi muammolar .....	279
Tranzaksiyalar orasidagi ziddiyatlar .....	281
Nazorat uchun savollar .....	283
<b>V BO'LIM. ZAMONAVIY MODELLASHTIRISH TIZIMLARI</b>	
<b>19- bob. Berilganlarni modellashtirish vositalari.....</b>	<b>285</b>
Barkerning CASE usuli .....	285
IDEF1X uslubiyati .....	290
DATARUN uslubiyati .....	296
CASE vositalarning tavsisi va sinflari .....	302
Nazorat uchun savollar .....	306
<b>20- bob. Toad Data Modeler muhitida berilganlarni modellashtirish .....</b>	<b>308</b>
Toad Data Modeler muhitining imkoniyatlari .....	308
Berilganlarning konseptual modelini yaratish .....	310
Berilganlarning mantiqiy modelini yaratish .....	331
Berilganlarning fizik modellashtirish .....	334
MS Access uchun DDL Script .....	338
MS SQL Server uchun DDL Script .....	353
Nazorat uchun savollar va amaliyot uchun topshiriqlar .....	359
<b>VI BO'LIM. RELATSION BERILGANLAR BAZASINI YARATISH</b>	
<b>21- bob. Berilganlar bazasining boshqarish tizimlari.....</b>	<b>361</b>
Apparat ta'minot .....	366
Programma ta'minoti .....	366
Berilganlar .....	372

<b>Protseduralar</b>	373
Foydalanuvchilar	373
BBB tizimlarining ustunlik va kamchiliklari	376
BBB tizimlarining tavsifi va sinflari	382
Nazorat uchun savollar	385
<b>22- bob. QBE tili</b>	387
MS Access BBB tizimida so'rovlni generatsiya qilish vositalari	388
Tanlov so'rovini yaratishdagi QBE vositalari	390
Ko'p jadvalli so'rovlni yaratish	396
Umumlashtiruvchi so'rovlar	397
QBE so'rovlnarning murakkab turlari	399
Jadval maydonlarini faol so'rovlar yordamida o'zgartirish	406
Avtomatik almashtirishli so'rovlar	414
Nazorat uchun savollar va amaliyot uchun topshiriqlar	416
<b>23- bob. SQL tiliga kirish</b>	418
SQL tilining vazifasi	418
SQL tilining tarixi	419
SQL operatorlarni yozish qoidalari	420
Berilganlar ustida qayta ishlash	422
Tanzh operatori	423
So'rovlar ifodasi	424
Bir satrni tanzh operatori	425
So'rov qismi	425
Jadval ifodasi	426
Group By bo'limi	427
Having bo'limi	427
Nazorat uchun savollar	428
<b>24- bob. Tanzh shartlari</b>	429
SELECT operatoriga doir misollar	431
Natijani tartiblash	436
Amaliyot uchun topshiriqlar	437
<b>25- bob. Natija olish va guruhash</b>	438
SQL tilida agregat (yig'uvchi) funksiyalarni ishlatish	438
Natijalami guruhash	439
Guruhashlari tanzh	440
So'rov qismi	441
ANY (SOME) va ALL kalit so'zlarni ishlatalishi	443

. EXIST va NOT EXIST predikatlarni qo'lish.....	444
Nazorat uchun savollar va amaliyot uchun topshiriqlar.....	446
<b>26- bob. Jadvallarni birlashmasi, kesishmasi va ayirmasi .....</b>	<b>447</b>
Ko'p jadvalli so'rovlar .....	448
Oddiy tutashish .....	448
Jadvallarni tutashtirish sintaksisi.....	449
Ichki va tashqi tutashtirish usullari .....	451
Nazorat uchun savollar va amaliyot uchun topshiriqlar .....	454
<b>27- bob. Berilganlar bazasi ustidagi ishlovlar .....</b>	<b>455</b>
Yangi berilganlarni jadvalga qo'shish .....	455
Berilganlarni o'zgartirish .....	456
Satrlarni o'chirish.....	457
SQL tilini relatson to'liqligi .....	458
Nazorat uchun savollar va amaliyot uchun topshiriqlar .....	460
<b>28- bob. Berilganlarni tavsiflash tili .....</b>	<b>461</b>
Identifikator .....	462
Berilganlar turlari .....	462
Skalyar amallar .....	465
Berilganlarni butunligini saqlash usullari .....	466
Majburiy berilganlar .....	466
Domen chegaralari .....	466
Mohiyat butunligi .....	468
Havola butunligi .....	469
Tashkilot qo'ygan chegaralar .....	471
Nazorat uchun savollar va amaliyot uchun topshiriqlar .....	472
<b>29- bob. Berilganlar bazasini yaratish .....</b>	<b>473</b>
Jadval yaratish .....	474
Jadval tavsifini o'zgartirish .....	476
Jadvalni olib tashlash .....	477
Indeks yaratish .....	477
Tasavvurfar .....	478
Tasavvurni yaratish .....	478
Tasavvurni olib tashlash .....	480
Tasavvurni tanlash operatoriga almashtirish .....	480
Tasavvurni qo'llashga qo'yilgan chegaralar .....	481
Tasavvurdagi berilganlarni o'zgartirish .....	482
WITH CHECK OPTION konstruksiyani ishlatalishi .....	483

Nazorat uchun savollar va amaliyot uchun topshiriqlar .....	486
<b>30- bob. Tranzaksiyani qo'llash .....</b>	<b>487</b>
Berilganlarni butunligini saqlashdagi chegaralarni tekshirish .....	489
Berilganlar bazasiga kirish uchun ruxsat berishni boshqarish .....	489
Foydalanuvchilarning identifikatori va egalik huquqi .....	490
Imtiyozlar .....	490
Boshqa foydalanuvchilarga imtiyozlar berish .....	491
Berilgan imtiyozlarini bekor qilish .....	492
Nazorat uchun savollar va amaliyot uchun topshiriqlar .....	494
<b>31- bob. Taqsimlangan berilganlar bazasi.....</b>	<b>495</b>
Asosiy tushunchalar .....	496
Taqsimlangan va parallel BBBT .....	497
Berilganlar bazalarining serverlari .....	501
Taqsimlangan ishllovning tayanch arxitekturalari .....	501
«Fayl-server» arxitekturasi .....	502
«Mijoz-server» arxitekturasi .....	504
«Uch bosqichli» arxitektura .....	505
Berilganlar bazasining «faol server» arxitekturasi .....	509
So'rovlarini optimallashtirish va qayta ishlash .....	511
Replikatsiyalar qaydnomalari .....	514
Nazorat uchun savollar .....	516
 <b>VII BO'LIM. RELATSION BBB TIZIMLARI</b>	
<b>32- bob. MS Access berilganlar bazasini boshqarish tizimi.....</b>	<b>517</b>
Tizimning unumiy tasnifi .....	517
Berilganlar bazasini yaratish va tashkil etish .....	524
Talabalar o'quv loyihasining tuzilishi .....	527
Nazorat uchun savollar va amaliyot uchun topshiriqlar .....	530
<b>33- bob. Jadvallar bilan ishslashning standart rejimi.....</b>	<b>532</b>
Jadvallar va ular bilan ishslash vositalari .....	532
BB bilan ishslash uchun ekrani shakllarni yaratish .....	535
MS Visual Basic protsedurasini yaratish .....	543
Foydalanuvchi funksiyalarini yaratish .....	544
Foydalanuvchining Sub protsedurasini yaratish .....	544
Hodisalarga ishllov beruvchi protsedurani yaratish .....	545
Nazorat uchun savollar va amaliyot uchun topshiriqlar .....	546

<b>34- bob. Hisobtlarni tuzish .....</b>	<b>548</b>
Mastep yordamida hisobtlarni yaratish .....	548
Hisobtlarni normal shaklga keltirish .....	551
Nazorat uchun savollar va amaliyot uchun topshiriqlar .....	557
<b>35- bob. So'rovlardan foydalanish .....</b>	<b>558</b>
So'rovlni tashkillashtirish .....	558
Berilganlar bazasini komplatsiya qilish (MDE-fayl) .....	564
Berilganlarga kirish huquqi sahfalari .....	568
MS Access loyihasi (ADP) .....	571
Nazorat uchun savollar va amaliyot uchun topshiriqlar .....	572
<b>36- bob. MS SQL Server .....</b>	<b>573</b>
Tizimning umumiyl tavsifi .....	573
Berilganlar bazasining maksimal parametrlari .....	574
Tizimdagi berilganlarning turi .....	576
Tizimning o'rnatilishi .....	578
Nazorat uchun savollar va amaliyot uchun topshiriqlar .....	586
<b>37- bob. MS SQL Server tizimi .....</b>	<b>587</b>
MS SQL Server 2000 tizimi xizmati .....	587
Berilganlar bazasini yaratish .....	590
Berilganlar bazasining jadvalini yaratish .....	591
Yenterprise Manager programmasida BB xabari bilan ishlash .....	594
Nazorat ochish savollar va amaliyot ochish topshiriqlar .....	598
<b>38- bob. Mijoz ilovalarini yaratish .....</b>	<b>600</b>
Mijoz ilovalarining yaratishdagi tarkibiy qismlar .....	600
MS SQL Server va MS Access aloqasi .....	602
MS Access da BB bilan ishlash uchun SQL Server tizimididan foydalanish .....	608
Nazorat uchun savollar va amaliyot uchun topshiriqlar .....	612
<b>GLOSSARIY .....</b>	<b>614</b>
<b>FOYDALANILGAN ADABIYOTLAR .....</b>	<b>630</b>

## **QAYDLAR UCHUN**

**HAKIMOV M.X., GAYNAZAROV S.M.**

# **BERILGANLAR BAZASINI BOSHQARISH TIZIMLARI**

**Toshkent – «Fan va texnologiya» – 2013**

Muharrir:	M. Hayitova
Tex. muharrir:	M. Holmuhamedov
Musavvir:	B. Nasridinov
Musahhih:	F. Ismoilova
Kompyuterda sahifalovchi:	N. Hasanova

E-mail: tipografiyacnt@mail.ru Tel: 245-57-63, 245-61-61.  
Nashr.lits. AIN №149, 14.08.09. Bosishga ruxsat etildi 21.11.2013.  
Bichimi 60x84 1/16. «Timez Uz» garniturasi. Ofset bosma usulida bosildi.  
Shartli bosma tabog'i 39,75. Nashriyot bosma tabog'i 40,5.  
Tiraji 500. Buyurtma №176.