

1. Ma'lumotlar bazasini boshqarish tizimini tashkil etuvchilari.

Informatsion tizimlarni yaratish bo'yicha jadal harakatlar va ma'lumotlar hajmining tez sur'atlar bilan oshib borishi 60-yillar boshida maxsus "Ma'lumotlar bazasini boshqarish tizimi" (MBBT) deb ataluvchi dasturiy kompleksning yaratilishiga olib keldi.

Ma'lumotlar bazasi — biror sohaga oid o'zaro bog'langan ma'lumotlar yig'indisining disk tashuvchidagi tashkiliy jamlanmasidir. Boshqacha qilib aytganda, ma'lumotlar bazasi — bu komputer xotirasiga yozilgan ma'lum bir strukturali, o'zaro bog'langan va tartiblangan ma'lumotlar majmui bo'lib, u biror bir obyektning xususiyatini, holatini yoki obyektlar o'rtasidagi munosabatni ma'lum ma'noda tavsiflaydi.

Ma'lumotlar bazasini boshqarish tizimi (MBBT) — bu dasturiy va apparat vositalarining murakkab majmui bo'lib, ular yordamida foydalanuvchi ma'lumotlar bazasini yaratishi va shu bazadagi ma'lumotlar ustida ish yuritishi mumkin. MBBT o'z maxsus dasturlash tillariga ham ega bo'lib, bu tillarga buyruqli dasturlash tillari deyiladi. MBBTga Oracle, Clipper, Paradox, FoxPro, Access va boshqalarni misol keltirish mumkin.

MBBT asosiy xususiyatlari — bu nafaqat ma'lumotlarni kiritish va saqlashda ishlatiladigan protseduralar tarkibi bo'lmasdan, ularning strukturasini ham tasvirlaydi. Ma'lumotlarni o'zida saqlab va MBBT ostida boshqariladigan fayl oldin ma'lumotlar banki, keyinchalik esa "Ma'lumotlar bazasi" deb yuritila boshlandi.

Ma'lumotlarni boshqarish tizimi quyidagi xossalarga ega:

- fayllar to'plami mantiqiy kelishuvini quvvatlaydi;
- ma'lumotlar ustida ish yuritish tili bilan ta'minlaydi;
- har xil to'xtalishlardan keyin ma'lumotlarni qayta tiklaydi;
- bir necha foydalanuvchilarning parallel ishlashini ta'minlaydi.

2. Ma'lumot modeli tushunchasi.

MBBT istalgan foydalanuvchiga ma'lumotlarga kirishga ruxsat etadiki, ularning hech biri amaliy jihatdan quyidagilar haqida tasavvurlarga ega bo'lmaydi:

- ma'lumotlarning xotirada fizik joylashishi va ular ko'rinishi;
- so'raladigan ma'lumotlarni izlash m'axanizmi;
- bir xil ma'lumotlarga bir vaqtning o'zida ko'pchilik foydalanuvchilar tomonidan bo'ladigan so'rovlar muammosi (amaliy dasturlar bilan);
- mumkin bo'lmagan va ruxsat etilmagan o'zgarishlarni kiritishdan ma'lumotlarni himoyalashni ta'minlash usullari;
- ma'lumotlar bazasini va boshqa ko'pgina MBBT funksiyalarini aktiv holatda ta'minlash.

MBBTning bu asosiy funksiyalarining bajarilishida har xil turdagi ma'lumotlar tavsiflanadi. Albatta, ma'lumotlar bazasini loyihalashni qo'llash (predmet) sohasini tahlil qilishdan va alohida foydalanuvchilar (masalan, korxona xodimlari, ular uchun

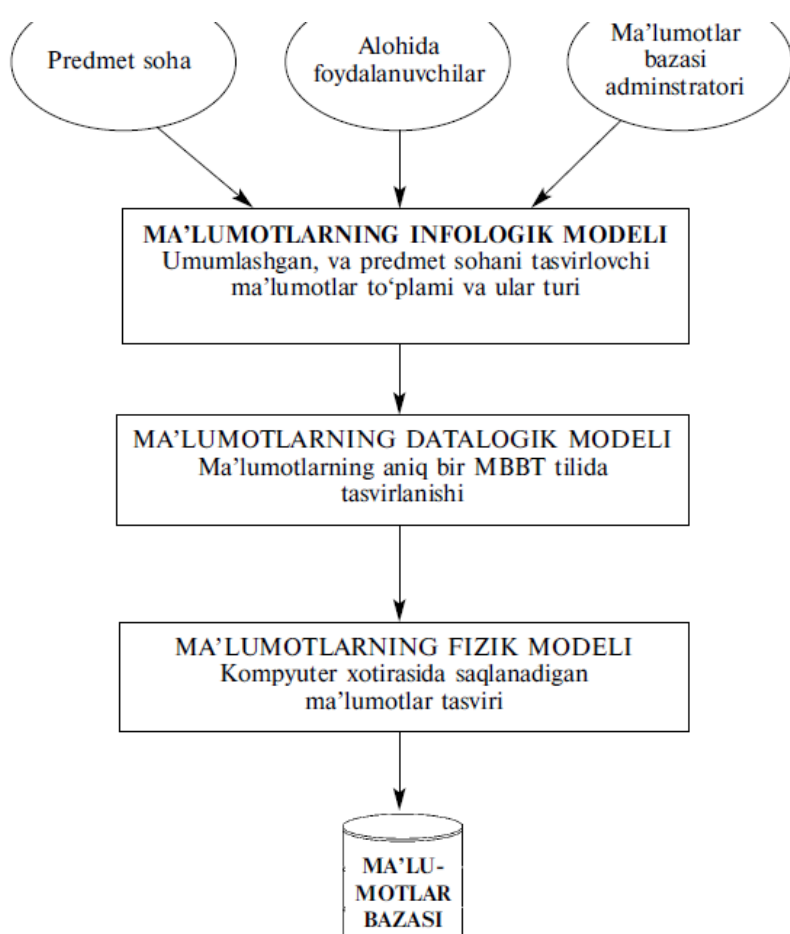
ma'lumotlar bazasi tuziladi) talablarini aniqlashdan boshlash kerak. Oldin umumlashgan holda rasmiy bo'lmagan tavsifga ega tuzilishi kerak bo'lgan ma'lumotlar bazasi tuziladi. Bu ma'lumotlar bazasini tuzish har bir foydalanuvchilardan so'rovlar natijasida olingan tasavvurlarni birlashtirilib amalga oshiriladi.

Insonlarga tushinarli bo'lgan tabiiy til, matematik formulalar, jadvallar, grafiklar va boshqa vositalar yordamida bajarilgan bu tavsif ma'lumotlar bazasini loyihalash ustida ish yuritishda ma'lumotlarning **infologik modeli** deb ataladi.

Insonlar uchun yo'naltirilgan bunday model to'laligicha ma'lumotlarni saqlash muhitining fizik parametrlariga bog'liq emas. Bu muhit, oxir-oqibatda, EHM xotirasi bo'lmasdan, balki inson xotirasi bo'lishi mumkin. Shuning uchun, infologik model birorta predmet sohasini akslantirishi uchun, real olamdagi o'zgarishlar qandaydir ta'rifni o'zgartirishni talab qilmaguncha o'zgarmasligi kerak.

Rasmda ko'rsatilgan boshqa modellar komputer uchun yo'naltirilgan hisoblanadi. Ular yordamida MBBT dasturlar va foydalanuvchilarga saqlanayotgan ma'lumotlardan foydalanish uchun imkoniyat yaratadi. Bu imkoniyat ma'lumotlarni fizik joylashishini hisobga olmasdan, balki dasturlar va foydalanuvchilar nomlari bo'yicha amalga oshiriladi. MBBT kerakli ma'lumotlarni tashqi eslab qolish qurilmasidan ma'lumotlarning **fizik modeli** bo'yicha izlaydi.

Demak, kerakli ma'lumotlardan foydalanishga ruxsat aniq bir MBBT yordamida bajariladi. Shuning uchun, ma'lumotlar modeli ushbu MBBT ma'lumotlarni tavsiflash tilida tavsiflanishi kerak bo'ladi. Ma'lumotlarning infologik modeli bo'yicha yaratiladigan



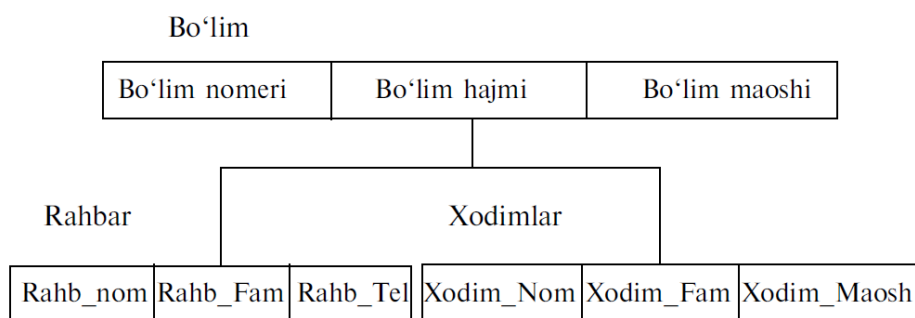
bunday tavsifga ma'lumotlarning **datalogik modeli** deyiladi.

Uch bosqichli arxitektura (infologik, datalogik va fizik bosqich) ma'lumotlarning saqlanishi unga ishlatiladigan dasturga bog'liqligini ta'minlaydi. Kerak bo'lganda saqlanayotgan ma'lumotlarni boshqa ma'lumot tashuvchilarga yozib qo'yish va (yoki) ma'lumotlarning fizik modelini o'zgartirish bilan uning fizik strukturasi qayta tashkil etish mumkin. Tizimga istalgan yangi foydalanuvchilarni (yangi ilovalarni) qo'shish mumkin. Agar datalogik model kerak bo'lsa, uni qo'shish mumkin.

3. Ierarxik (shajara) ma'lumot modeli.

Iyerarxik MB tartiblangan daraxtlar to'plamidan tuziladi. Yanada aniqrog'i, bir xil turdagi daraxtlarning bir nechta tartiblangan nusxalari to'plamidan iborat bo'ladi. Daraxt turi bitta "ildizli" tur yozuvidan va tartiblangan bitta yoki bir nechta daraxt osti turlaridan (ular har biri daraxtning turidir) tashkil topadi. Daraxt turi umuman olganda iyerarxik ravishda tashkil topgan yozuvlar turlari to'plamini tasvirlaydi.

Daraxt turiga misol:



1.2-rasm. MBning iyerarxik sxemasi.

Ma'lumotlar ustida ish yuritish:

Iyerarxik tashkil qilingan ma'lumotlar bilan ish yuritishda quyidagi operatorlarni misol tariqasida namuna qilib olish mumkin:

- MBda ko'rsatilgan daraxtni topish;
- Bir daraxtdan ikkinchisiga o'tish;
- Bitta yozuvdan boshqa daraxt ichiga kirish (masalan, bo'limdan - birinchi xodimga);
- Bitta yozuvdan ierarxiya tartibida boshqasiga o'tish;
- Yangi yozuvni ko'rsatilgan o'ringa qo'yish;
- Joriy yozuvni o'chirish;
- Butunlikni chegaralash.

Avlod va ajdodlar o'rtasidagi murojaatlar yaxlitligi avtomatik tarzda qo'llab-quvvatlanadi. Asosiy qoidalar: hech bir avlod o'z ota-onasiz mavjud bo'lmaydi. Shuni ta'kidlash kerakki, shunga o'xshash bir ierarxiyaga kirmaydigan yozuvlar orasidagi murojaatlar yaxlitligini qo'llab bo'lmaydi (bunda tashqi murojaatning misoli sifatida, Kaf nomeri maydoni tarkibiga kiruvchi Kurator yozuvi turi nusxasi bo'lishi mumkin).

Iyerarxik tizimlarda MB tasvirlaydigan forma iyerarxiyaga qo'yilgan cheklashlar asosida qo'llanadi.

4. Tarmoqli ma'lumot modeli.

Tarmoqli tizimning tipik vakili Integrated Database Management System (IDMS) tizimidir. 1970-yillar bir nechta tizimlar yaratildi, ular orasida IDMS ham bor edi. Bu tizim ko'pchilik operatsion tizimlarni boshqarishda IBM firmasi mashinalarini ishlatish uchun mo'ljallangan. Tizimning arxitekturasini Integrated Database Management System (IDMS)

dasturlash tili bo'yicha Conference on Data Systems Languages (CODASYL) tashkiloti taklifiga ko'ra tuzilgan. Bu tashkilot Kobol dasturlash tiliga mas'uldir.

Ma'lumotlarning tarmoqli strukturasi

Ma'lumotlarni tashkil qilishga bo'lgan tarmoqli yondashish iyerarxik kengaytmali bo'ladi. Iyerarxik strukturali yozuv — avlod bitta ajdodda aniq bo'lishi kerak. Ma'lumotlarning tarmoqli strukturasi esa avlodlar ajdodlarda istalgan sonda bo'lishi mumkin.

Tarmoqli MBda yozuvlar va ular orasidagi bog'lanishlar tashkil topgan, ya'ni yanada aniqrog'i MB strukturasi har bir tipdagi nusxalar to'plamidagi yozuvlar turi to'plami va berilgan aloqa turlari to'plamidagi har bir turdagi nusxalar to'plamini tashkil qiladi.

Bog'lanishlar ikki xil turdagi yozuvlar uchun aniqlanadi: ajdod va avlod. Aloqa turi nusxasi ajdod yozuv turining bitta nusxasi va avlod yozuv turi tartiblangan nusxalar to'plamidan tashkil topadi. P ajdod yozuv turi va C avlod yozuv turi bilan berilgan L bog'lanish turi uchun quyidagi ikki shart bajarilishi kerak:

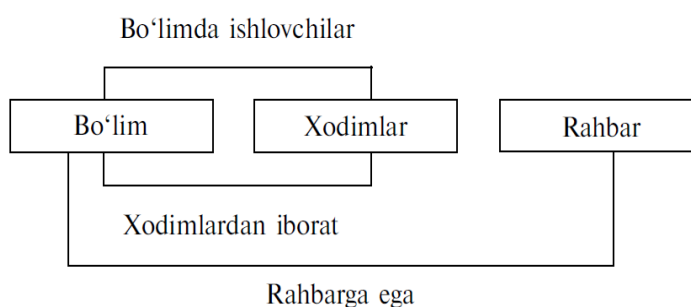
- P turning har bir nusxasi faqat bitta ko'rinishdagi L ning ajdodi hisoblanadi;
- Har bir C ko'rinishdagi nusxa bittadan ortiq bo'lmagan L ko'rinishning avlodi hisoblanadi.

MB tarmoqli sxemasiga oddiy misol:

Ma'lumotlar bilan ish yuritish

Taxminiy amallar to'plami quyidagilar bo'lishi mumkin:

- bir xil turdagi yozuvlarga ega to'plamda aniq yozuvni topish;
- ayrim bog'lanishlar yordamida ajdoddan birinchi avlodga o'tish;
- ayrim bog'lanishlar yordamida keyingi avlodga o'tish;
- ayrim bog'lanishlar bo'yicha avloddan ajdodga o'tish;
- yangi yozuv yaratish;
- yozuvni o'chirish;
- yozuvni modifikatsiyalash;
- bog'lanishni tashkil etish;
- bog'lanishni olib tashlash;
- boshqa bog'lanishni tashkil qilish va boshqalar.



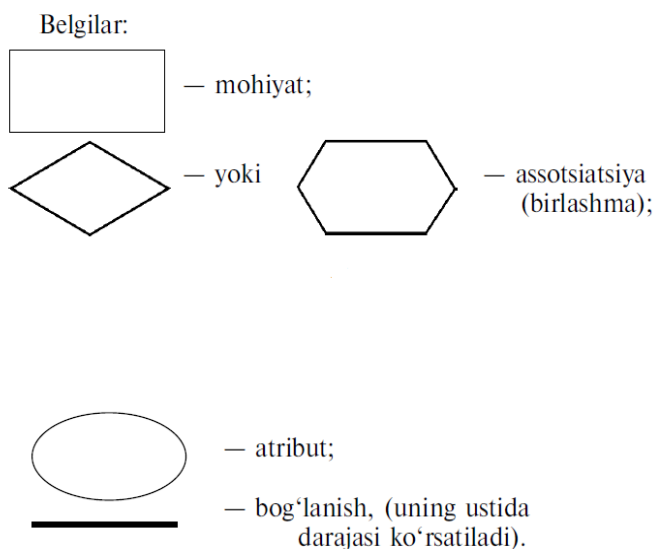
1.5-rasm. MB tarmoqli sxemasiga misol.

5. Mohiyat aloqa diagrammasi.

Hozirgi kunda axborot tizimlarini loyihalash va predmet sohasini tavsiflash uchun ko'p usullar mavjud. Shulardan biri **ER (Entity-Relation) ya'ni mohiyat-alloqa usulidir**.

Bu model 1976-yil Piter Chen tomonidan kiritilgan bo'lib, u o'ziga bir qator grafik diagrammalarni oluvchi bir necha har xil turdagi komponentlarni birlashtirgan.

Piter Chen mohiyatlar to'plami va ular orasida bog'lanish sifatida relyatsion ma'lumotlar strukturasi interpretatsiya qilishni taklif qiladi. ER modelining asosiy komponentlari mohiyat, bog'lanish va atribut hisoblanadi.



Predmet sohani mohiyat aloqa usulida tavsiflaganda quyidagi bosqichlarda ish olib boriladi:

1. Predmet sohani obyektlari aniqlanadi.
2. Obyekt sohalari (atributlari) belgilanadi va uning kalit parametri aniqlanadi. Kalit parametri obyektni identifikatsiyalaydi.
3. Obyektlar o'rtasida aloqa o'rnatiladi va ular sinflarga ajratiladi.
4. Maxsus belgilar kiritilib, mohiyat aloqa diagrammasi o'rnatiladi. Bu diagramma Predmet sohasining infologik modeli grafik tasviri hisoblanadi.

6. Relyatsion modelining asosiy tushunchalari.

Mohiyat — har xil obyekt, uning ma'lumotlari ma'lumotlar bazasida saqlanadi. Mohiyat odam, joy, samolyot, reyslar, gullar va boshqalar bo'lishi mumkin. Mohiyat turi va mohiyat nusxasi tushunchalari mavjud, ularni bir-biridan farqlash kerak. Mohiyat turi tushunchasi bir jinsli narsalar, hodisalar, shaxslar uchun qaraladi. Mohiyat nusxasiga aniq narsalar to'plami sifatida qaraladi. Masalan, mohiyat turi SHAHAR, uning nusxasi esa — TOSHKENT, MOSKVA bo'lishi mumkin.

Atribut — mohiyatni xarakterlovchi nomlar. U o'zida yagona murakkab bo'lmagan strukturani tasvirlab, mohiyat holatini xarakterlaydi. Masalan, "O'quvchi" mohiyati atributi — kod, familiya, ism, manzil, yosh va boshqalar.

Mohiyat atributlari to'plami cheksizdir. U axborot tizimlari bilan ishlaydigan foydalanuvchi talabiga va yechiladigan masalaga bog'liqdir.

Yana bir misol: AVTOMOBIL mohiyati atributi bu — tur, marka, nomer belgisi, rangi va boshqalar. Bu yerda tur bilan nusxa orasida farq mavjud. Rang atributi turi ko'p qiymatli yoki nusxali: qizil, ko'k, oq va boshqalar.

Mohiyat va atribut turlari orasida absolyut farq yo'q.

Mohiyat turi bilan aloqada faqat atribut bor. Boshqacha kontekstda atribut o'zi mustaqil mohiyat bo'lib qatnashishi ham mumkin. Masalan, avtomobil zavodi uchun rang — bu faqat ishlab chiqarish mahsuloti atributi, lok-bo'yoq fabrikasi uchun esa rang — mohiyat turidir.

Kalit — atributlarning minimal to'plami, uning qiymati yordamida kerakli mohiyat nusxasini topish mumkin. Masalan, samolyotlar uchishi uchun "Reys jadvali" mohiyati kalitlari "Reys nomeri" yoki "Jo'nash joyi", "Uchish vaqti" va "Qo'nish joyi" to'plami bo'lishi mumkin.

Aloqa — ikki yoki undan ortiq mohiyatlarning bir-biri bilan o'zaro bog'lanishidir. Agar ma'lumotlar bazasini faqat bir-biri bilan bog'lanmagan ma'lumotlar bilan to'ldirishga yo'naltirilsa, u holda uning strukturasi juda oddiy bo'ladi. Lekin, MBni tuzishning asosiy

talablaridan biri — bu boshqa mohiyat qiymati bo'yicha bitta mohiyatni izlab topish imkoniyatini ta'minlashdir. Buning uchun ular orasida aloqani aniqlashni tashkil etish zarur. Real ma'lumotlar bazasida yuzlab yoki minglab mohiyatlar bir biri bilan millionlab bog'lanish o'rnatishi kamdan kam. Bunday bog'lanishlar to'plami infologik modelning murakkabligini bildiradi.

7. Ma'lumotlarni tasvirlashda jadvallardan foydalanish.

Ma'lumotlarni relyatsion modeli asosida munosabat tushunchasi yotadi. Munosabatni ikki o'lchamli **jadvallar** yordamida tavsiflash qulay. Jadval tushunarli, ko'rimli va inson uchun oddiy. Munosabatlar to'plami ma'lumotlarni saqlash uchun ishlatilishi mumkin. Shu bilan birga ular orasidagi bog'lanishlarni modellashtirish imkonini beradi. Misol: ikki o'lchamli jadvalni ko'ramiz.

Quyidagi jadvalda:

Birinchi satr - atribut nomlari;

Ikkinchi satr - atributlarni qisqa nomlari;

Qolgan satrlar - atribut qiymatlari;

Xizmatchi nomeri	F.I.SH	Unvoni	Tugilgan yili	Bulim	Mutaxassis kodi	Mansab	Maosh
XN	FISH	UN	TY	VL	MK	MB	MSH
2518	Valiev S.P.	t.f.n.	1985	1	PM	DOTS	260000
2567	Soliev I.T.	t.f.d.	1987	2	EVM	Prof.	360000
3245	Aliev S.I.	t.f.n.	1988	1	PM	Ilm.xodi m	280000
3267	Boriev A.O.	Akad.	1982	3	ASU	Prorek.	400000

Yuqoridagi ko'rib chiqilgan ierarxik tarmoqli va boshqa ma'lumotlarni tasvirlash usullarini shunday ikki o'lchamli jadvalga keltirish mumkin. Bunday jadvallarni quyidagi xususiyatlari bo'ladi.

- jadvalni har bir ma'lumot elementi maydon hisoblanadi va takrorlanuvchi guruhlar bo'lmaydi;
- barcha ustunlar bir jinslidir;
- har bir ustunga nom tayinlangan;
- jadvalda bir xil satr ikki marta uchramaydi;
- bunday jadvalda satr va ustunlar ixtiyoriy tartibda qaraladi va ixtiyoriy ketma-ketlikda ishlatilishi mumkin.

Bunday xususiyatlarga **jadvalli munosabat** deyiladi. Bu munosabat asosida qurilgan MB **relyatsion MB** deyiladi.

Yuqoridagi jadvalni sxematik qisqartirilgan ko'rinishi (sxemasi):

Xizmatchi (xiz.nom F.I.SH, unvoni, tugilgan yili, bulim, mut.kodi. mansab, maosh).

Bunday yozuv **ma'lumotlar bazasi sxemasi** deyiladi. Shunday qilib, relyatsion MB ma'lumot elementlar to'plami asosida quriladi.

Munosabat yoki jadvalni kortejlar to'plami deb qarash mumkin. Agar jadvalda n ta ustun bo'lsa, u **n tartibli kortej**dan iborat deyiladi va munosabat ham **n -darajali** deyiladi. Har bir atribut qiymatlari to'plami **domen** deyiladi.

Munosabatda har bir kortej o'zining kalit identifikatoriga (nomiga) ega bo'lishi kerak va kalit quyidagi xususiyatlarga ega bo'ladi:

1. Kortej kalit qiymati bilan bir qiymatli ifodalanishi kerak.
2. Kalitda ortiqchalik bo'lmazligi kerak, ya'ni hech qanday atributni kalitdan olib tashlash mumkin emas.

8. Ma'lumotlar bazasida munosabatlar.

Ob'ektlarni identifikatsiyalash uchun yoki boshqacha aytganda kompyuter xotirasida yozuvlarning o'rnini aniqlashda ma'lumot elementi ishlatiladi. Bu elementni biz **kalit** deb ataymiz. Agar kalit ob'ektni bir qiymatli identifikatsiyalasa, u **birlamchi kalit** deyiladi. Aks holda **ikkilamchi kalit** deyiladi. Agar ob'ektlarni identifikatsiyalash uchun bir nechta atributlar talab etilsa, bunday kalit **tulangan kalit** deb ataladi. Agar A va B guruhdagi ob'ektlar berilgan bo'lsa, ular orasidagi quyidagi mosliklar yoki munosabatlarni o'rnatishimiz mumkin:

1. Birga – bir (1:1) munosabat. A va B ob'ektlar to'plami orasida 1:1 munosabat o'rnatilgan deyiladi, agarda A ob'ektning har bir nusxasiga B ob'ektning bitta nusxasi mos kelsa, va aksincha, B ob'ektning har bir nusxasiga A ob'ektning bitta nusxasi mos kelsa.
2. Birga – ko'p (1:n) munosabat. A va B ob'ektlar to'plamida A ob'ektning har bir nusxasiga B ob'ektning bir nechta nusxasi mos kelsa, shu bilan birga B ob'ektning har bir nusxasiga A ob'ektning bittadan ko'p bo'lmagan nusxasi mos kelsa shunday munosabat hosil bo'ladi.
3. Ko'pga – bir (n:1) munosabat A va B ob'ektlar to'plami orasida o'rnatilsa, unda A ob'ektning har bir nusxasiga B ob'ektning ko'pi bilan bitta nusxasi mos keladi. B ob'ektning nusxalari orasida shundaylari mavjudki, ularga A ob'ektning bir nechta nusxasi mos keladi.
4. Ko'pga – ko'p (m:n) munosabat. A va B ob'ektlar orasida shunday munosabat o'rnatilgan deyiladi, agarda A ob'ektning har bir nusxasiga B ob'ektni bir nechta nusxasi mos kelsa va aksincha.

9. Relyatsion hisoblash tillari.

Relyatsion MBBTda ma'lumotlar bilan ishlash uchun bir qancha tillar yaratilgan. Ba'zi hollarda bu tillarni ma'lumotlarni qism tillari deb ataladi. MB bilan ishlovchilar bu tillarda avtomatlashtirishni 3 bosqichga bo'lishadi:

- 1) Eng pastki bosqich – kortej deb ataladi. Bunda dasturchi yozuvlar yoki kartijlar bilan ishlaydi.

2) Relyatsion algebra deyiladi. Bunda foydalanuvchi munosabatlar ustida yuqori bosqichli amallar to'plamini kiritadi.

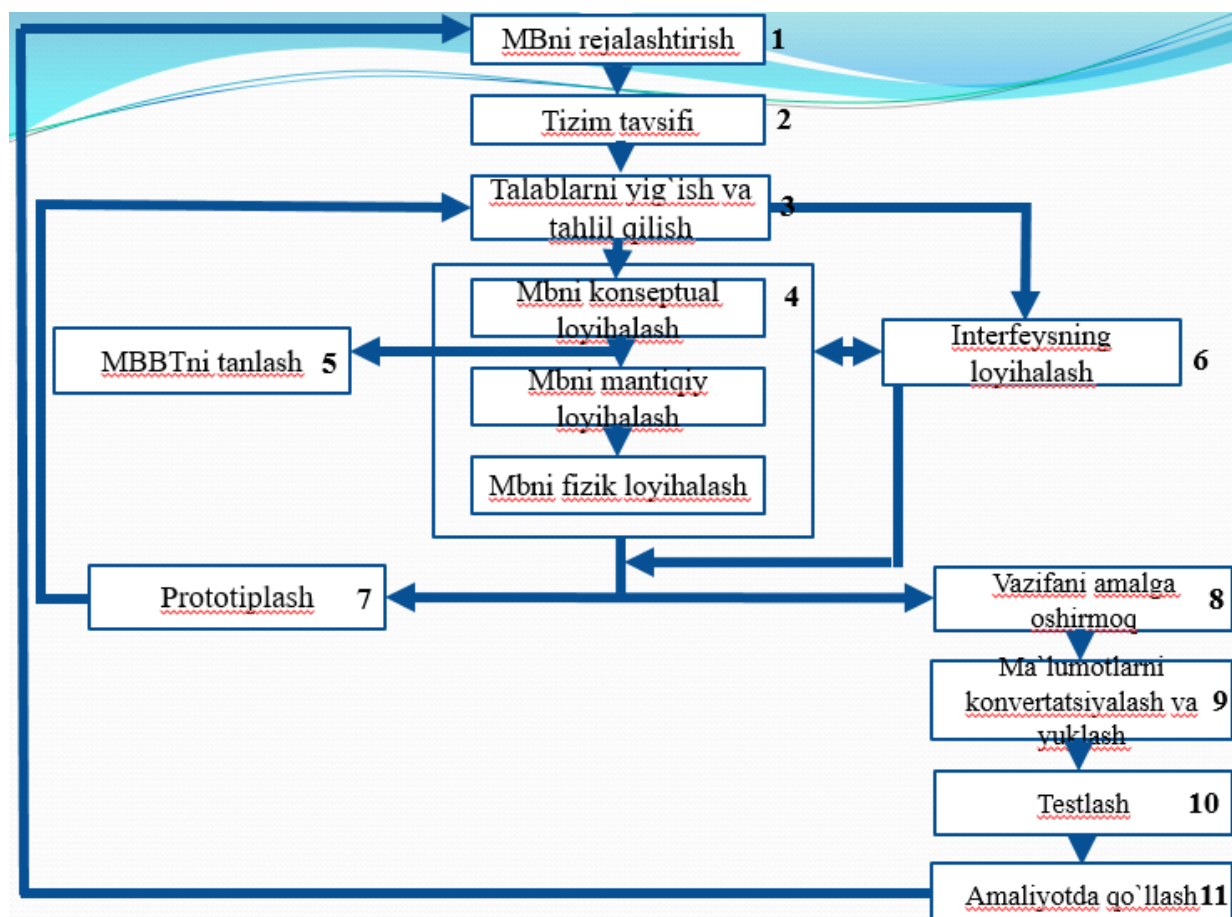
3) Eng yuqori bosqich – hisoblash bosqichi. Bunda foydalanuvchi bevosita kompyuterga maxsus tillarda murojaat qiladi va mashina bu murojaatni qabul qiladi.

Relyatsion MBBT da ma'lumotlar bilan ishlashda ishlatiladigan 2ta katta gurux tillari **relyatsion hisoblash** deyiladi. Relyatsion hisoblash predikatlarni hisoblashga asoslangan bo'lib ifodalarni yozishga mo'ljallangan qiodalar to'plamidan iboratdir. Ular yordamida biz mavjud munosabatlardan yangi munosabatlar yaratishni ta'minlaymiz. Bunday ifodalarni yozishda solishtirish amallari, mantiqiy amallar va mavjudlik kvanteri va umumiylik kvanteri ishlatiladi.

Hozirgi paytda relyatsion MBBT ni taraqqiyotida yangi til QBE tili ishlamoqda. Bu tilda relyatsion algebra va relyatsion hisoblashlarda ko'zda tutilmagan bir qancha imkoniyatlar kirgan. Bu tilni hususiyati shundan iboratki, u terminallarda ishlashga mo'ljallangan. So'rovlarni yaratish uchun maxsus ekran redaktoridan, munosabat va redaktorlaridan foydalanamiz. QBE tilida foydalanuvchi o'zi olishini mo'ljallagan natijani so'rov ko'rinishida tasvirlaydi va MBBT uni kerakli amallar ketma – ketligiga aylantirib beradi.

10. Ma'lumotlar bazasini rejalashtirish.

Ma'lumotlar bazasi tizimini yaratish hayot siklini samarali tashkil etishga imkon beradigan boshqaruv faoliyati.



Tashkillashtirishning axborot tizimi umumiyashtirilib integratsiyalanishi lozim. Ma'lumotlar bazasini rejalashtirish quyidagi boshqaruv standartlarini tashkil etishni o'z ichiga oladi:

- ma'lumotlarni qay darajada to'planganligi;
- qay darajada alohida formatga ega ekanligi;
- qandaydir kerakli hujjatlarni to'planganligi;
- qay darajada loyihalash va amalga oshirishganligi.

11. Ma'lumotlar bazasi tushunchasi. ↓ (12-savolga qarang)

12. Ma'lumotlar bazasining asosiy tushuncha va terminlari.

Hozirgi kunda inson faoliyatida ma'lumotlar bazasi (**MB**) kerakli axborotlarni saqlash va undan oqilona foydalanishda juda muhim rol o'ynamoqda. Sababi: jamiyat taraqqiyotining qaysi jabhasiga nazar solmaylik o'zimizga kerakli ma'lumotlarni olish uchun, albatta, MBga murojaat qilishga majbur bo'lamiz. Ma'lumki, MB tushunchasi fanga kirib kelgunga qadar, ma'lumotlardan turli ko'rinishda foydalanish juda qiyin edi. Dastur tuzuvchilar ma'lumotlarini shunday tashkil qilar edilarki, u faqat qaralayotgan masala uchungina o'rinli bo'lardi. Har bir yangi masalani hal qilishda ma'lumotlar qaytadan tashkil qilinar va bu yaratilgan dasturlardan foydalanishni qiyinlashtirar edi. Har qanday axborot tizimining maqsadi real muhit obyektlari haqidagi ma'lumotlarga ishlov berishdan iborat. Keng ma'noda **ma'lumotlar bazasi** - bu qandaydir bir predmet sohasidagi real muhitning aniq ob'ektlari haqidagi ma'lumotlar to'plamidir. **Predmet sohasi** deganda avtomatlashtirilgan boshqarishni tashkil qilish uchun o'rganilayotgan real muhitning ma'lum bir qismi tushiniladi. Masalan, korxona, zavod, ilmiy tekshirish instituti, oliy o'quv yurti va boshqalar. Shuni qayd qilish lozimki, **MB**ni yaratishda ikkita muhim shartni hisobga olmoq zarur:

Birinchidan, ma'lumotlar turi va ko'rinishi ularni qo'llaydigan dasturlarga bog'liq bo'lmasligi lozim, ya'ni **MB**ga yangi ma'lumotlarni kiritganda yoki ma'lumotlar turini o'zgartirganda, dasturlarni o'zgartirish talab etilmasligi lozim.

Ikkinchidan, MBdagi kerakli ma'lumotni bilish yoki izlash uchun biror programma tuzishga hojat qolmasin. Shuning uchun ham **MB**ni tashkil etishda ma'lum qonun va qoidalarga amal qilish lozim. Bundan buyon **axborot** so'zini **ma'lumot** so'zidan farqlaymiz, ya'ni **axborot** so'zini umumiy tushuncha sifatida qabul qilib, **ma'lumot** deganda aniq bir belgilangan narsa yoki hodisa sifatlarini nazarda tutamiz.

Ma'lumotlar bazasini yaratishda, foydalanuvchi axborotlarni turli belgilar bo'yicha tartiblashga va ixtiyoriy belgilar birikmasi bilan tanlanmani tez olishga intiladi. Buni faqat ma'lumotlar tizimalashtirilgan holda bajarish mumkin.

Tizimlashtirish – bu ma'lumotlarni tasvirlash usullari haqidagi kelishuvni kiritishdir. Agar ma'lumotlarni tasvirlash usuli haqida kelishuv bo'lmasa, u holda ular tizimlashtirilmagan deyiladi. Tizimlashtirilmagan ma'lumotlarga misol sifatida matn fayliga yozilgan ma'lumotlarni ko'rsatish mumkin.

Ma'lumotlar bazasidan foydalanuvchilar turli amaliy dasturlar, dasturiy vositalari, predmet sohasidagi mutaxassislar bo'lishi mumkin. Ma'lumotlar bazasining zamonaviy texnologiyasida ma'lumotlar bazasini yaratish, uni dolzarb holatda yuritishni va foydalanuvchilarga undan axborot olishini ta'minlovchi maxsus dasturiy vosita, ya'ni ma'lumotlar bazasini boshqarish tizimi yordami bilan markazlashtirilgan holda amalga oshirishni nazarda tutadi.

Ma'lumotlar bazasi – EHM xotirasiga yozilgan ma'lum bir strukturaga ega, o'zaro bog'langan va tartiblangan ma'lumotlar majmuasi bo'lib, u biror bir ob'ektning xususiyatini, holatini yoki ob'ektlar o'rtasidagi munosabatni ma'lum ma'noda ifodalaydi. MB foydalanuvchiga strukturalashtirilgan ma'lumotlarni saqlash va ishlatishda optimal qulaylikni yaratib beradi. Ma'lumki ma'lumotlarni kiritish va ularni qayta ishlash jarayoni katta hajmdagi ish bo'lib ko'p mehnat va vaqt talab qiladi. MB bilan ishlashda undagi ma'lumotlarning aniq bir strukturaga ega bo'lishi, birinchidan foydalanuvchiga ma'lumotlarni kiritish va qayta ishlash jarayonida undagi ma'lumotlarni tartiblashtirish, ikkinchidan kerakli ma'lumotlarni izlash va tez ajratib olish kabi qulayliklarni tug'diradi. MB tushunchasi fanga kirib kelgunga qadar, ma'lumotlardan turli ko'rinishlarda foydalanish juda qiyin edi. Bugungi kunda turli ko'rinishdagi ma'lumotlardan zamonaviy komputerlarda birgalikda foydalanish va ularni qayta ishlash masalasi hal qilindi. Komputerlarda saqlanadigan MB maxsus formatga ega bo'lgan muayyan tuzilmali fayl bo'lib, undagi ma'lumotlar o'zaro bog'langan va tartiblangandir. Demak, ma'lumotlar bazasi deganda ma'lum bir strukturada saqlanadigan ma'lumotlar to'plami tushuniladi. Boshqacha qilib aytganda MB - bu ma'lum bir strukturaga ega bo'lgan ma'lumotlarni o'z ichiga oluvchi maxsus formatga ega bo'lgan fayldir.

Ma'lumotlarni strukturalashtirish - bu shunchaki ma'lumotlarni tasvirlashda qandaydir moslikni kiritish usulidir. Odatda MB ma'lum bir ob'ekt sohasini ifodalaydi va uning ma'lumotlarni o'z ichiga oladi, ularni saqlaydi va foydalanuvchiga ma'lumotlarni qayta ishlashda undan foydalanish imkonini yaratib beradi.

Ma'lumotlar bazasi – bu ma'lum bir predmet sohasiga oid tizimlashtirilgan (strukturalashtirilgan) ma'lumotlarning nomlangan to'plamidir. Ma'lumotlar bazasi - axborot tizimlarining eng asosiy tarkibiy qismi hisoblanadi. Ma'lumotlar bazasidan foydalanish uchun foydalanuvchi ishini yengillashtirish maqsadida ma'lumotlar bazasini boshqarish tizimlari yaratilgan. Bu tizimlar ma'lumotlar bazasini amaliy dasturlardan ajratadi.

Ma'lumotlar bazasini boshqarish tizimi (MBBT) - bu dasturiy va apparat vositalarining murakkab majmuasi bo'lib, ular yordamida foydalanuvchi ma'lumotlar bazasini yaratish va shu bazadagi ma'lumotlar ustida ish yuritishi mumkin. Juda ko'p turdagi MBBT

mavjud. Ular o'z maxsus dasturlash tillariga ham ega bo'lib, bu tillarga SUBD buyruqli dasturlash tillari deyiladi. MBBTga Oracle, Clipper, Paradox, FoxPro, Access va boshqalarni misol keltirish mumkin.

Ma'lumotlar bazasini boshqarish tizimi – bu ma'lumotlar bazasini yaratish, ularni dolzarb holatini ta'minlash va undagi zarur axborotni topish ishlarini tashkil etish uchun mo'ljallangan dasturlar majmui va til vositasidir.

Ma'lumotlar bazasi tushunchasi maydon, yozuv, fayl (jadval) kabi elementlar bilan chambarchas bog'liq. **Maydon** - bu ma'lumotlarni mantiqiy tashkil etishni elementar birligi bo'lib, u axborotni eng kichik va bo'linmas birligi bo'lgan rekvizitga mos keladi. Maydonni tasvirlash uchun quyidagi tavsiflardan foydalaniladi.

Yozuv – bu mantiqiy bog'langan maydonlar to'plami. Yozuv tuzilishi uchun uning tarkibiga kiruvchi maydonlar tarkibi va joylashishi ketma-ketligi aniqlanib, ularni har biri ichida elementar **yozuvlarning nusxasi** deb ataladi. Yozuv ob'ektning biror bir elementi haqida to'liq ma'lumotni ifodalaydi. **Fayl (jadval)** - bu bir xil tuzilmaga ega bo'lgan yozuvning nusxalar to'plamidir. U o'zicha har bir maydonda qiymatga ega.

Har bir MB jadvali o'zining birlamchi kalitiga ega bo'lishi mumkin. **Birlamchi kalit** deganda yozuvlar qaytarilmasligini ta'minlovchi maydon (polya) yoki maydonlar guruhi tushiniladi. Birlamchi kalit sifatida ishlatiladigan maydon yoki maydonlar guruhi bir xil yozuvga ega bo'lmaslik shartini bajarishi kerak. Boshqa maydonlarda bir xil yozuvlar takrorlanishi mumkin. Shu sabab ular birlamchi kalit bo'la olmaydi. Birlamchi kalit qisqa va sonli maydonlardan tashkil topishi maqsadga muvofiqdir. MB jadvaliga birlamchi kalitni kiritishdan maqsad, jadvaldagi ma'lumotlarni izlash, tartiblashtirish va tanlab olishda qulaylikni beradi. Birlamchi kalit kiritish yoki kiritmaslik foydalanuvchi tomonidan MB jadvali strukturasi tashkil qilishda aniqlanadi.

Bosh jadval yordamida qaram jadvaldagi mos ma'lumotlarni chaqirishni ta'minlash uchun qaram jadvalda tashqi kalit tashkil qilinadi. "Bitta-ko'pga" bog'lanish holatida tashqi kalit bosh jadvalda tashkil qilinadi. Birinchi va ikkinchi kalitlarni aniqlashda MBBT avtomatik ravishda jadvalda indekslarni quradi.

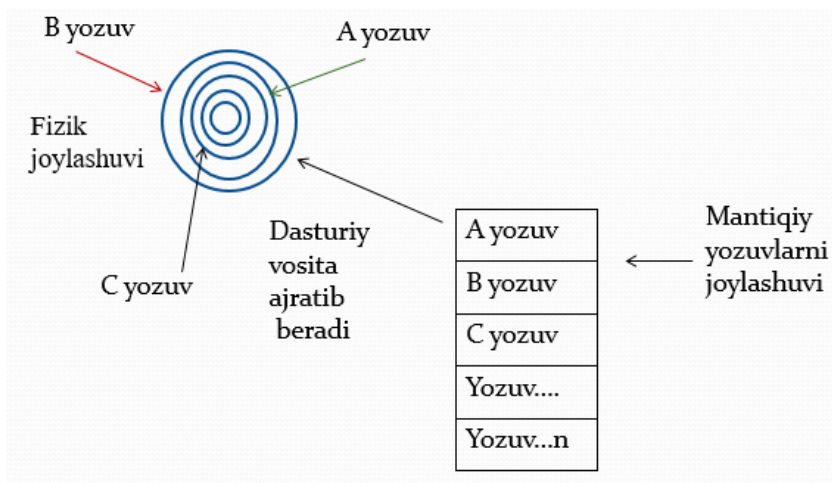
13. Ma'lumotlar bazasini sinflarga ajratish.

Ob'ektlarni sinflarga ajratish deyilganda, barcha ob'ektlar to'plamini birorta norasmiy belgi bo'yicha qism to'plamlarga ajratish tushuniladi. MBni ko'pligini hisobga olib uni sinflarga ajratish belgilari xilma-xil. Hozirgi kunda MBni quyidagi sinflari ko'p qo'llaniladi:

- MB ma'lumotlarni tasvirlash shakliga qarab: video, audio, multimedia guruhlariga ajratish mumkin.
- Video MB ma'lumotlarini ko'rinishiga qarab o'z navbatida matnli va grafik tasvirli bo'ladi.

- Matnli MB ma'lumotlarni strukturalashganiga qarab strukturalashgan, qisman strukturalashgan va strukturalashmagan MB ga bo'linadi.
- Strukturalashgan MB o'z navbatida ma'lumotlarni modeliga qarab: ierarxik, tarmoqli, relyatsion, ob'ekli relyatsion, ob'ektga yo'naltirilgan MBga bo'linadi.
- Bundan tashqari strukturalashgan MBlari statik va dinamik shuningdek, markazlashgan va taqsimlangan MBga bo'linadi.
- MBni foydalanuvchilar soniga qarab: bitta va ko'p foydalanuvchili MBga bo'linadi.
- Ma'lumotlarni saqlanishiga qarab operatsion va analitik bo'ladi.
- Iqtisodiy nuqtai nazardan, pulli va pulsiz MB ga bo'linadi.
- Murojaat qilish darajasiga qarab: ommabop va murojaati cheklangan MB ga bo'linadi.

14. Ma'lumotlarni fizik va mantiqiy tavsifi.



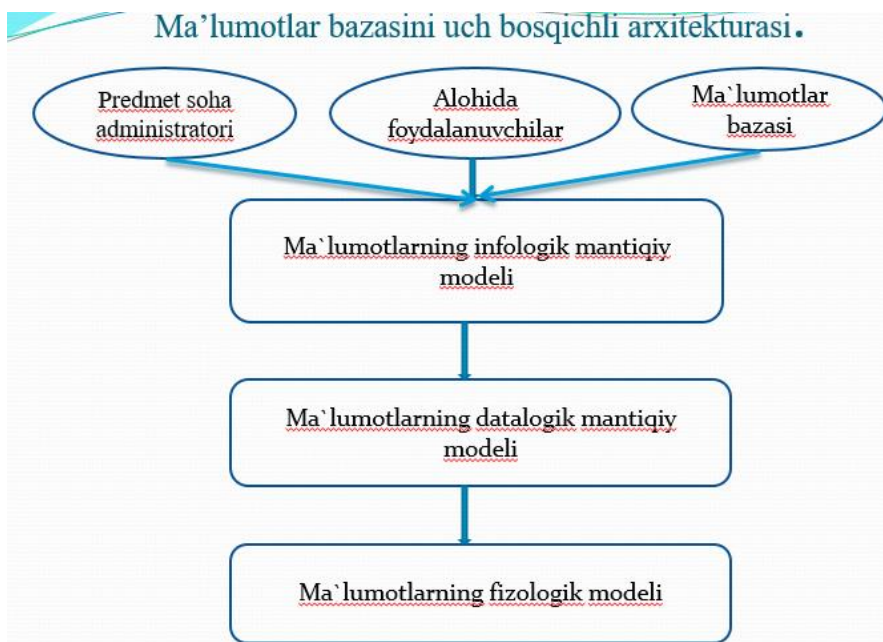
Ma'lumotlarni tavsirlash va ular orasidagi munosabat aloqalar o'rnatish 2 xil bo'ladi:

- Logik yoki mantiqiy;
- Fizik;

Fizik tasvirlashda ma'lumotlar mashinani tashqi xotirasida saqlashi bilan farqlanadi. Mantiqiy tasvirlashda esa amaliy dasturchi yoki foydalanuvchi tomonidan

ma'lumotlarni tasvirlash ko'rinishi tushuniladi.

15. Ma'lumotlar bazasini uch bosqichli arxitekturasini. (shuningdek, 2-savolga qarang)



16. Ma'lumotlar bazasini loyihalash.

Ma'lumotlarni loyihalashning uch bosqichi

- Konseptual ma'lumotlar bazasini loyihalash
- Logik ma'lumotlar bazasini loyihalash
- Fizik ma'lumotlar bazasini loyihalash

Konseptual ma'lumotlar bazasini loyihalash

- Korxonadagi barcha fizik mulohazalarni ma'lumotlar bazasi modelini qurish jarayoni.

- Ma'lumot modeli bu foydalanuvchining alohida talablariga ma'lumotdan foydalanishni yaratish.

- Konseptual model logik loyihalash bosqichi uchun axborotlarning asosi hisoblanadi.

Logik ma'lumotlar bazasini loyihalash

- Korxonada ma'lum bir ma'lumot modeli asosida ma'lumot ishlatilgan modelini qurish jarayoni, lekin muayyan MBBT va fizik bo'limlar bundan mustasno.
- Konseptual model mantiqiy ma'lumotlar bazasi modeli uchun tartibga keltirilgan va tuzilgan model hisoblanadi.

Fizik ma'lumotlar bazasini loyihalash

- Ikkilamchi saqlash bo'yicha ma'lumotlar bazasi amalga oshirishning tavsifini ishlab chiqarish jarayoni.
- Ma'lumotga samarali murojaatni tashkil qilishda bazalar o'zaro aloqasi fayl tuzilmalaridan foydalanishni tasvirlaydi. Bundan tashqari barcha o'zaro aloqadorlik chegaralari va xavfsizlik choralarini tasvirlaydi.
- Maxsus MBBTga mo'ljallangan.

Informatsion tizimlarni loyihalash jarayoni yetarlicha murakkab masala. U ma'lumotlarning infomantiqiy modelini tuzishdan, ya'ni mohiyatni identifikatsiyalashdan boshlanadi. Keyin loyihalashning logik model protsedurasining quyidagi qadamlarini bajarish kerak bo'ladi.

1. Ma'lumotlar bazasi jadvalining (asosiy jadval) har bir bog'liqsiz mohiyatini (sterjen) tasvirlash va bu asosiy (bazaviy) jadval birinchi kalitini spetsifikatsiyash lozim.

2. Har bir assotsiatsiyani ("ko'pga-ko'p" va boshqa bog'lanishlar) asos jadval ko'rinishida tasvirlash. Bu jadvalda assotsiatsiya qatnashuvchilarni identifikatsiya qilish uchun tashqi kalitni ishlatish kerak.

3. Har bir xususiyatni tashqi kalit bilan asos jadval sifatida tasvirlash lozim. Jadvalning tashqi kalitiga va uning boshlang'ich kalitiga qo'yilgan cheklashlarni spetsifikatsiyalash lozim.

4. Avvalgi punktlarda qaralgan har bir belgilanishlarni tashqi kalit bilan asos jadval sifatida tasvirlash lozim. Shu turdagi har qanday tashqi kalitlarni cheklashlarni spetsifikatsiyalash lozim.

5. Har bir xossani mohiyatni ifodalovchi asos jadvalda maydon ko'inishida tasvirlash lozim (bu xossa bilan aniqlanuvchi mohiyat tasvirlanadi).
6. Normallashtirishning qandaydir prinsiplarini bevosita buzushning oldini olish maqsadida, normallashtirish protsedurasini bajarish lozim.
7. Normallashtirish jarayonida qandaydir jadvallarni bo'laklash jarayoni sodir bo'lsa, ma'lumotlar bazasining infomantiqiy modelini takomillashtirish lozim va sanab o'tilgan qadamlarni takrorlash kerak.
8. Loyihalashtirilayotgan ma'lumotlar bazasi yaxlitligiga qo'yiladigan cheklanishlarni ko'rsatish va hosil qilingan jadval va maydonlarni qisqacha tasvirlab berish kerak (agar lozim bo'lsa).

17. Ma'lumotlar bazasini boshqarish.

Ma'lumotlarni boshqarish (administratorlash) quyidagi jarayonlarni o'z ichiga oladi.

- Rejalashtirish
- Rivojlantirish va shakllantirish standartlari
- Rivojlantirish siyosati va prosedura
- Ma'lumotlar bazasini konseptual va mantiqiy loyihalash.
- Ma'lumotlar bazasini fizik loyihalash.
- Xavfsizlik va ma'lumotlar butunligini nazorat qilish.
- Jarayonni tekshirib borish.
- Ma'lumotlar bazasini sozlash.

18. Funksional bog'lanish tushunchasi.

Relyatsion MBda ma'lumotlarni strukturasidan tashqari ularni sxematik informatsiyasiga ham e'tibor beriladi. MBni strukturasida haqidagi informatsiya munosabat sxemasi yordamida beriladi. Sxematik informatsiyalar esa atributlar orasidagi funksional bog'lanishlar orqali ifodalanadi. MB munosabatlarida atributlarni tarkibini quyidagi talablarga javob beradigan qilib guruhlash kerak:

- Atributlar orasidagi zaruriy bo'lmagan takrorlanishlar bo'lmasligi kerak.
- Atributlarni guruhlaganda ma'lumotlar takrorlanishi minimal darajada qilib ta'minlanishi kerak. Bu bevosita ma'lumotlarni tez qayta ishlash imkonini beradi. Bunga normallashtirish jarayoni yordamida erishiladi.

Normallashtirish deganda berilgan munosabatni bir necha marta oddiy va kichik munosabatlarga ajratish tushuniladi. Bu jarayonda mumkin bo'lgan barcha funksional bog'lanishlar aniqlanadi.

Normallashtirish nazariyasi jadval maydonlari orasidagi u yoki bu bog'liqlikning borligiga asoslanadi. Bunday bog'liqliklarning ikki turi aniqlangan: **funksional va ko'p qiymatli**.

Funksional bog'liqlik. Berilgan ixtiyoriy vaqtda A maydonning har bir turli qiymatiga mos ravishda albatta B maydonning har bir turli qiymatidan faqat bitta qiymati bo'lsa, jadvalning B maydoni xuddi shu jadvalning A maydoniga **funksional bog'langan** deyiladi. Aytib o'tish kerakki, A va B maydonlar tashkil qiluvchilar bo'lishi mumkin.

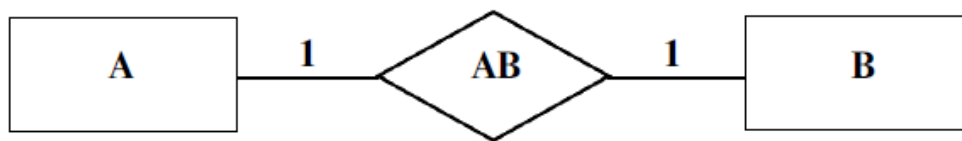
To'liq funksional bog'liqlik. B maydon agar A ga funksional bog'liq bo'lsa va A maydonning boshqa qiymatlariga bog'liq bo'lmasa, A tashkil qiluvchiga **to'liq funksional bog'liq** bo'ladi.
Ko'p qiymatli bog'liqlik. Agar A maydonning har bir qiymatiga mos ravishda B maydonning aniqlangan qiymatlari bor bo'lsa, A maydon xuddi shu jadvalning B maydonini **ko'p qiymatli** aniqlaydi.

19. Tranzitiv va funksional bog'lanishlar.

Ikki mohiyat, masalan, A va B o'rtasida to'rtta bog'lanish turi bo'lishi mumkin:

1. Birga-bir bog'lanish (1:1):

Har bir vaqt momentida A mohiyatning har bir vakiliga (nusxasiga) B mohiyatning vakili 1 yoki 0 bilan mos keladi.



2.1-rasm. Birga-bir bog'lanish.

Masalan: O'quvchi jinsi erkak yoki ayol bo'lishi mumkun.

2. Birga-ko'p bog'lanish (1:M): A mohiyatning bitta vakiliga 0,1 yoki B mohiyatning bir necha vakili mos keladi.



2.2-rasm. Birga-ko'p bog'lanish.

Masalan: O'quvchi imtihonda javob berishga 1, 2, 3, 4 yoki 5 ball olishi mumkin.

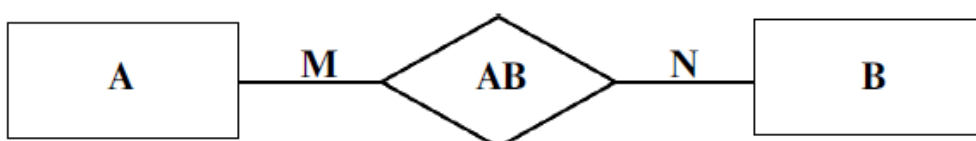
Xuddi shunday ikkala yo'nalishda ham mohiyatlar orasida aloqa bo'lishi mumkin. Yana ikkita aloqa turi mavjud.

3. Ko'pga-bir bog'lanish (M:1):



2.3-rasm. Ko'pga-bir bog'lanish.

4. Ko'pga-ko'p bog'lanish (M:N)



2.4-rasm. Ko'pga-ko'p bog'lanish.

20. Relyatsion algebra amallari.

Relyatsion algebra amallarini operandlari sifatida doimiy yoki o'zgarmas va o'zgaruvchan munosabatlar ishlatiladi. Relyatsion algebrada 5ta amal ishlatiladi:

- 1) Birlashtirish, R va S munosabatlarni birlashtirish $R \cup S$ ko'rinishida berilib, bu amalni natijasi R munosabatga tegishli bo'lgan yoki S munosabatga tegishli bo'lgan yoki ikkalasiga ham tegishli bo'lgan kartejlar to'plamidir. Bu amallarni bajarayotganda bir xil tartibda bo'lishi kerak. Natijani tartibi ham operandlar tartibiga teng bo'ladi.
- 2) Ayirma R va S munosabatlarni ayirmasi $R \setminus S$ ko'rinishida yoziladi va undagi kartejlar to'plami R munosabatga tegishli, lekin S munosabatga tegishli bo'lmagan kartejlardir. Bu amalni bajarganda ham operandlarni tartibi bir xil bo'lishi kerak.
- 3) Dekart ko'paytma. Bizda R va S munosabat berilgan bo'lsin. R munosabatni tartibi R -r va S munosabatniki S -q ga teng bo'lsin. Unda dekart ko'paytma $R \times S$ ko'rinishida yozilib, uning natijasi uzunligi $r+q$ ga teng bo'lgan kartejlar to'plamidan iborat bo'lib, bu kartejlarni birinchi r komponentasi R kartejga teng bo'ladi, qolgan q komponentasi S kartejga teng bo'ladi.
- 4) Proeksiya, R munosabatga bu amal tadbiriq etilganda, R munosabatdan ba'zi bir komponentalar olib tashlanadi. Qolganlari esa qaytadan tartiblanadi.
- 5) Seleksiya tanlash. Bu amal bajarilganda operandlar sifatida munosabat atributlari ishtirok etadi va solishtirish arifmetik amallari: $=$, \neq , \leq , \geq , $<$, $>$ va mantiqiy amallar: \vee (U), yoki (\vee), not amallari ishlatiladi.

Ma'lumot modelini rivojlanish konsepsiyasi 5 ta bosqichni ko'rsatishi mumkin:

1. 60- yillarning 2 – yarmida, bunda asosan ierarxik modellarga e'tibor berilgan;
2. 70- yillarni 1 – yarmi, tarmoqli modellar;
3. 70- yillarning 2 – yarmi, relyatsion modellar;
4. 80- yillarning 1 – yarmi, semantik modellar;
5. 80- yillarning 2 – yarmi, ob'ektga mo'ljallangan sistema.

21. Ma'lumotlar bazasida 1:M munosabat.

Agar A va B guruxdagi ob'ektlar berilgan bo'lsa, ular orasidagi quyidagi mosliklar yoki munosabatlarni o'rnatishimiz mumkin:

1. Birga – bir (1:1) munosabat. A va B ob'ektlar to'plami orasida 1:1 munosabat o'rnatilgan deyiladi, agarda A ob'ektning har bir nusxasiga B ob'ektning bitta nusxasi mos kelsa, va aksincha, B ob'ektning har bir nusxasiga A ob'ektning bitta nusxasi mos kelsa.
2. Birga – ko'p (1:n) munosabat. A va B ob'ektlar to'plamida A ob'ektning har bir nusxasiga V ob'ektning bir nechta nusxasi mos kelsa, shu bilan birga B ob'ektning har bir nusxasiga A ob'ektning bittadan ko'p bo'lmagan nusxasi mos kelsa shunday munosabat hosil bo'ladi.
3. Ko'pga – bir (n:1) munosabat A va B ob'ektlar to'plami orasida o'rnatilsa, unda A ob'ektning har bir nusxasiga B ob'ektning ko'pi bilan bitta nusxasi mos keladi. B

ob'ektning nusxalari orasida shundaylari mavjudki, ularga A ob'ektning bir nechta nusxasi mos keladi.

4. Ko'pga – ko'p ($m:n$) munosabat. A va B ob'ektlar orasida shunday munosabat o'rnatilgan deyiladi, agarda A ob'ektning har bir nusxasiga B ob'ektni bir nechta nusxasi mos kelsa va aksincha.

Ob'ektlarni tahlil qilib bo'lingandan so'ng, shu ob'ektga qo'yiladigan boshlang'ich so'rovni ishlab chiqamiz. Avtovakzalni faoliyati iqtisodiy va texnik ko'rsatgichlar bilan bog'liq bo'lganligi uchun, yo'lovchilarga informatsion ma'lumot berganligi uchun yaratayotgan informatsion sistemamiz quyidagi so'rovlarga javob berishi kerak:

1. Har bir reys uchun nechta bilek sotilganligi va umumiy sotilgan biletlarni aniqlash;
2. Reysni nomeri bo'yicha reys haqida ma'lumotlar chiqarish;
3. Marshrutlar haqidagi zarur axborotlarni chiqarish;
4. Aniq reyslar uchun qapysi haydovchilar tayinlanganligi va ular haqida ma'lumotlar olish;
5. Avtobusni texnik xarakteristikalar haqidagi ma'lumotni olish.

Ko'rib chiqilayotgan predmet sohani ob'ektlari orasida quyidagi tipdagi bog'lanishlar mavjud:

1. 1:1 – biletlar bilan reys ob'ektlari orasidagi bog'lanish (sotilgan);
2. M:1 – marshrut va reys orasidagi bog'lanish (marshrut munosabati);
3. 1:M – marshrut va bekatlar orasidagi bog'lanish (bekatlar);
4. 1:N – reys va haydovchi orasidagi bog'lanish (tayinlash);
5. M: N- haydovchi va avtobus modeli orasidagi bog'lanish (haydovchiga ruxsat berish);
6. M:1 – avtobus va avtobus modeli orasidagi bog'lanish (toifa).

Shunday qilib, ko'rilyotgan masalada asosan ob'ektlar aniqlanadi va ular orasidagi bog'lanish topiladi, sinflarga ajratiladi.

22. Ma'lumotlar bazasida M:M munosabat. ↑ (21-savolga qarang)

23. Ma'lumotlar bazasida 1:1 munosabat. ↑ (21-savolga qarang)

24. SQL tilining SELECT operatori bilan murakkab so'rovlar yaratish.

Ko'pgina amaliy masalarni yechishda ma'lum shartlar asosida axborotlarni ajratib olish talab etiladi. Masalan: STUDENT jadvalidan Petrov familiyasi talabalarni chiqarish kerak.

SELECT surname, same, FROM student WHERE surname='Petrov';

Where shartida solishtirish amallari jumladan, $=$, $<$, $<=$, $>=$, shuningdek, mantiqiy amallar "and", "or", "not" amallari ishlashi mumkin. Ular yordamida murakkab shartlar tuziladi.

Masalan: 3-kurs stipendiya oladigan talabalarni ismi familiyasini chiqaring.

SELECT name, surname FROM student WHERE kurs=3 and stipend>0;

Mantiqiy shartlarni berishda WHERE parametri tarkibida IN, BETWEEN, LIKE, IS NULL amallari ham ishlatiladi. IN, NOT parametrlari ishlatilganda tekshirilayotgan maydon

qiymati berilgan ro'yxat bilan solishtiriladi. Bu ro'yxat IN operatori o'ng tomonidan () ichida yoziladi.

1. Jadvaldan imtihon baholari «4» va «5» baholi talabalar ro'yxatini chiqaring.

SELECT * FROM exam-marks WHERE mark IN (4, 5);

Birorta ham 4 va 5 olmagan talaba haqidagi ma'lumotlar olish uchun NOT IN yoziladi.

BETWEEN amali maydon qiymatini berilgan intervalga kirganligini tekshirish uchun ishlatiladi.

2. 30 va 40 soat doirasida o'qitiladigan fanlarni chiqarish so'rovini tuzing.

SELECT* FROM subject WHERE hour BETWEEN 30 AND 40;

BETWEEN amali maydonlar sonli va simvolli bo'lganda ham ishlatiladi.

LIKE amali f-t simvolli toifadagi maydonlar uchun ishlatiladi. Bu amal maydonni satrli matnlarni LIKE dan so'ng ko'rsatilgan qism satr bilan solishtiradi.

3. Studentlar jadvalida familiyasi «M» xarfi bilan boshlanadigan talaba haqida ma'lumot chiqaring.

SELECT * FROM student WHERE surname LIKE M%;

% - foiz belgisi shu pozitsiyada ixtiyoriy simvollar ketma-ketligi kelishini anglatadi.

Bu amallarni maydonda o'tkazib yuborilgan qiymatlarni yoki noaniq qiymatlarni topishda ishlatib bo'lmaydi.

Sonli ma'lumotlarni uzunlik o'zgartirish uchun atribut amallardan foydalanamiz. Bunda quyidagi amallar ishlatiladi: «-», «+», «*», «/».

Misol: **SELECT surname, name, stipend kurs, (stipend*kurs)/2 FROM student WHERE kurs=4 and stipend > 0;**

25. Ma'lumotlar bazasini himoyalash.

Базадаги маълумотлар бузилиши ва йўқолишига қуйидагилар мисол бўлади:

қурилмалардаги бузилиш;

маълумотлар базасининг аппарат воситаларига физик таъсирлар;

табiiй офат;

ваколатли фойдаланувчиларнинг хатолари;

ваколатсиз фойдаланувчилар ва дастурларнинг ғаразли ниятдаги ҳаракатлари;

МББТ ёки операцион тизимдаги дастурий хатоликлар;

амалий дастурлардаги хатоликлар;

фойдаланувчиларнинг конфликтли сўровларини ҳамкорликда бажарилиши ва б.

Ахборот хавфсизлигининг қуйидаги учта компоненталари тушунчаси орқали аниқланади:

- *конфиденциаллик* (рухсат этилмаган киришдан ҳимоя);
- *бутунлик* (ахборотни рухсат этилмаган ўзгартиришдан ҳимоялаш);
- *кириш ҳуқуқи* (ишланувчанликни ҳимоялаш, бузилишдан ҳимоялаш, ахборот ва ресурсларни рухсат этилмаган ушлаб қолишдан ҳимоялаш).

Ахборот хавфсизлигининг сиёсатини ишлаб чиқишда, аввало ҳимоя қилинувчи объект ва унинг вазифалари аниқланади. Сўнгра душманнинг бу объектга қизиқиши даражаси, ҳужумнинг эҳтимолли турлари ва кўриладиган зарар баҳоланади.

Маълумотлар **maxfiyligi** (konfidentsialligi) buzilishiga sabab bo'luvchi taxdidlarga ixtiyoriy hisoblash tizimida saqlanayotgan yoki bir tizimdan boshqasiga uzatilayotgan ma'lumotlarni qasddan yoki tasodifan fosh etilishi kiradi. Maxfiylikni (kotsidentsiallik) buzilishiga ma'lumotlarga ruxsatsiz murojaat etishga qaratilgan g'arazli amallar, hamda dastur yoki malakasiz opreatorning tasodifiy xatosi sabab bo'ladi.

Butunlik buzilishi taxdidiga axborot tizimida qayta ishlanayotgan yoki birlamchi axborot manbasidan kiritilayotgan ma'lumotlarni qasddan yoki tasodifan o'zgartirish amali kiradi. Butunlik buzilishiga biror shaxsning o'zining g'arazli maqsadi yo'lida ma'lumotlarni qasddan o'zgartirishi, hamda dasturiy yoki apparat tizimining tasodifiy xatoligi sabab bo'lishi mumkin.

Ихтиёрий универсал компьютер тизимининг дастурий таъминоти учта асосий ташкил этувчидан иборат бўлади: операцион тизим (ОТ), тармок дастурий таъминоти (ТДТ) ва маълумотлар базасини бошқариш тизими (МББТ). Шунинг учун компьютер тизимларини бузишга бўлган барча уринишларни уч гуруҳга ажратиш мумкин:

- операцион тизим даражасидаги ҳужум;
- тармок дастурий таъминоти даражасидаги ҳужум;
- маълумотлар базасини бошқариш тизими даражасидаги ҳужум.

Маълумотлар базасининг ахборот хавфсизлигига бўлган бarcha potentsial taxdidlarni bartaraf etish amalda imkonsiz. Bundagi real masala potentsial taxdidlarni yuz berish ehtimolligini aniq tizim uchun ma'qul darajagacha kamaytirishdan iborat. Taxdidlardan himoyalanganlikning ma'qul mos darajasi qo'llanilish sohasi, ajratilgan mablag' yoki amaldagi qonunlar bilan aniqlanishi mumkin. Qoidaga ko'ra, qat'iy ierarxiyali taxdidlar daraxtini qurish imkonsiz. Shuning uchun umumiy xatar tizim komponentlari zaifliklarining yetarlicha murakkab funktsiyasidan iborat bo'ladi. Turli salbiy ta'sirlar ham yetarlicha murakkab tarzda ma'lumotlar bazasi tizimining asosiy sifat va xavfsizlik xarakteristikalariga ta'sir qiladi.

Himoya tizimini qurishdagi asosiy rioya etiluvchi prinsip baravar chidamlilik prinsipi hisoblanadi. Axborot xavfsizligini ta'minlovchi mavjud resurslarni shunday taqsimlash kerakki, bunda ixtiyoriy salbiy tashqi va ichki ta'sirlarda tizimning biror xavf-xatar ko'rsatgichi minimallashtirishi kerak. Tizimda bartaraf etuvchi choralari ko'rilmagan taxdidning mavjudligi tizimga bo'lgan boshqa salbiy ta'sirlarning oldini oluvchi samarali to'siqlarini yaratilishiga qaramasdan ko'rilgan barcha choralar kutilgan natijani bermaydi. Bundan quyidagi amaliy xulosa kelib chiqadi: taxdidlar har tomonlama mukammal o'rganilishi kerak va har bir taxdid uchun mos himoya metodi amalga oshirilishi kerak.

Axborot tizimlari xavfsizligining ma'lum darajasini ta'minlash uchun yuzaga keluvchi tahdidlarning tabiatini, tizim yoki texnologiyaning zaiflik darajasini kamayishini ta'minlovchi asosiy metodlarni, xavfsizlik darajasiga mos holda va ushbu xavfsizlik darajasini ta'minlovchi qabul qilingan mos yechim narxini anglash kerak.

Yechim qabul qiluvchi shaxslar tomonidan tahdidlar tabiatini va xavfsizlikni ta'minlovchi metodlar maqsadi va xususiyatlarini yetarli darajada tushunib yetmasligi turlicha xatoliklarga sabab bo'ladi.

Ixtiyoriy ob'ektni, shuningdek ma'lumotlar bazasi tizimlari axborot xavfsizligini har tomonlama tahlil qilish tahdidlarni klassifikatsiyalashni (sinflarga ajratishni) taqozo etadi. Ilmiy klassifikatsiyalash oldingi tahlil tajribalariga asoslanadi, mazmunan yaqin bo'lgan holatlarni mavjud klassifikator bo'limiga kiritadi. Xavfsizlikni aniqlashda tanlangan yondoshuvga bog'liq bo'lmagan holda tahdidlar va ularning manbalarini sinflash o'ziga xos ahamiyatga ega. Turli sinflarning mavjudligi tadqiqotchiga ma'lum tizim uchun jiddiy bo'lgan tahdidni o'tkazib yubormasligiga imkoniyat yaratadi.

Ma'lumotlar bazasi axborot xavfsizligini ta'minlash muammosi ko'p qirrali hisoblanadi. Ma'lumotlar bazasining o'zi — bu cheksiz turli tuman bo'lgan real dunyoning modelidir. Ma'lumotlar bazasi tizimlarini loyihalash va faoliyatini qo'llab-quvvatlash ma'lumotlarni qayta ishlashning zamonaviy dasturiy – apparat vositalarini va tashkiliy boshqaruvning yetarlicha murakkab sxema va tuzilmasini talab etadi. Kompyuter va kommunikatsiya industriyasidagi yuqori tempdagi yangiliklar inobatga olinsa, keltirilgan sinflar barcha tahdidlarni qamrab olmaganligi oson tushunib olinadi.

Маълумотлар базасини бошқариш тизими (МББТ) даражасидаги ҳужум. Катъий тузилишнинг мавжудлиги ва аниқ белгиланган амаллар МББТни химоялаш вазифасини осонлаштиради. Кўп холларда хакерлар компьютер тизимининг ОТ даражасидаги химоясини бузишни маъқул кўрадилар ва ОТ воситалари ёрдамида МББТ файлларига кириш рухсатига эга бўладилар. Бирок, агар етарли даражадаги химоя механизмларига эга бўлмаган, ёки хатолари мавжуд, ёмон тестланган МББТ версиясидан фойдаланилса, ёки МББТ администратори томонидан хавфсизлик сиёсатини аниқлашда хатоларга йўл кўйилган бўлса, у холда хакернинг МББТ даражасидаги химоядан ўтиш эҳтимоли мавжуд бўлади.



1-Расм. Ахборот хавфсизлигининг концентруаль модели

26. Ma'lumotlarni bir butunligini ta'minlash. ↑ (shuningdek 25-savolga qarang)

Butunlik (inglizcha – teginmaslik, saqlanish, bir butun) – deganda har qanday vaqtda ma'lumotlarning to'g'riligi tushuniladi. Bu maqsad ma'lum chegarada bo'lishi kerak. MBBT ma'lumotlar bazasiga kiritilayotgan har bir bo'lak qiymatning to'g'riligini tekshirish imkonini bermaydi. Masalan, kiritilayotgan 5 (hafta kun nomerini ko'rsatuvchi) qiymati haqiqatda 3 ga teng bo'lishini tekshirmaydi. Boshqa tomondan 9 qiymati aniq xato bo'ladi va MBBT bunga javob qaytaradi. Chunki bu nomer (1, 2, 3, 4, 5, 6, 7) sonlar to'plami ichida yo'q.

MBning butunligini ta'minlashni ma'lumotlarni har xil to'g'ri bo'lmagan o'zgarishlar yoki buzulishdan himoyalash deb qarash kerak. Zamonaviy MBBT butunlikni ta'minlash uchun bir qancha vositalarga ega: Butunlikni aniqlash usulini **uchta guruhga** ajratadi: **Mohiyat** bo'yicha butunlik; **Murojaat** bo'yicha butunlik; **Foydalanuvchi** aniqlaydigan butunlik.

Butunlikni aniqlash usullari:

1. Birinchi kalitda qatnashuvchi atributlarga aniqlanmagan qiymatlar qabul qulinishiga ruxsat etilmaydi.

2. Tashqi kalit qiymati quyidagilardan biri bo'lishi kerak:

- Birinchi kalit qiymatiga teng;
- To'liq aniqlanmagan ya'ni tashqi kalitda qatnashadigan har bir atribut qiymati aniqlanmagan bo'lishi kerak.

3. Har qanday aniq bir ma'lumotlar bazasi uchun qo'shimcha qoidalar spetsifikatsiyalari mavjud. Ular ishlab chiquvchilar yordamida aniqlanadi. Ko'p hollarda tekshiriladi:

- U yoki bu atributning unikalligi;
- Qiymatlar diapazoni;
- Qiymatlar to'plamining aloqadorligi.

27. Tasavvurlar va ular haqida ma'lumot. (shuningdek 43-savolga qarang)

Ma'lumotlar bazasi jadvallardan tashkil topadi. Jadvallar alohida fayl ko'rinishida, yoki birorta faylni bo'lagi bo'lishi mumkin.

Ma'lumki, **SELECT** operatori yordamida virtual jadvallar yaratish, ya'ni vaqtinchalik jadvallar yaratish mumkin. Bunday jadvallar vaqtinchalik bo'lib, yaratgan foydalanuvchi o'zi undan foydalanishi mumkin.

SQL-da tasavvur SQL bayonotining natijalar to'plamiga asoslangan virtual jadvaldir.

Tasavvurda haqiqiy jadval kabi qatorlar va ustunlar mavjud. Tasavvurdagi maydonlar - bu ma'lumotlar bazasidagi bir yoki bir nechta haqiqiy jadvallarning maydonlari.

Siz tasavvurga SQL, WHERE va JOIN funktsiyalarini qo'shishingiz va ma'lumotlarni bitta jadvaldan kelgan kabi taqdim etishingiz mumkin.

Tasavvurlar ham vaqtinchalik jadvallar bo'lib, ularga ko'p foydalanuvchilar murojaat qilishi mumkin va u ma'lumot bazasidan majburan olib tashlanguncha mavjud bo'ladi.

Tasavvurlar MB oddiy jadvallariga o'xshash bo'lib, ma'lumotlar saqlovchi fizik ob'ekt hisoblanmaydi. Tasavvurlarda ma'lumotlar jadvallardan tanlab olinadi. Tasavvurlar foydalanuvchilardan jadvallarni ba'zi ustunlarini yashirish uchun yoki ko'pincha foydalanuvchiga kerakli bo'lgan bir nechta jadvaldan bitta yaratish kerak bo'ladi.

Tasavvurlar yaratish uchun CREATE VIEW komandasi ishlatiladi.

Uni formati quyidagicha: **CREATE VIEW** «tasavur nomi» **AS** «select so'rovi»;

```
CREATE VIEW view_name AS      SELECT column1, column2,      ...      FROM table_name  
WHERE condition;
```

Tasavvur har doim eng so'nggi ma'lumotlarni ko'rsatadi! Foydalanuvchi tasavvur so'rovi yuborganda har safar SQL so'rovidan foydalanib ma'lumotlar bazasi mexanizmi ma'lumotlarni qayta yaratadi.

Tasavvurlarga ham ma'lumot baza jadvallari kabi nom beriladi. Bu nom birorta ham jadval nomi bilan bir xil bo'lmasligi kerak. AS so'zidan keyin ma'lumotlar tanlash uchun so'rov iborasi yoziladi.

Tasavvurni CREATE yoki REPLACE VIEW buyrug'i bilan yangilash mumkin.

```
CREATE OR REPLACE VIEW view_name AS      SELECT column1, column2,      ...  
FROM table_name  
WHERE condition;
```

Tasavvurlar MB jadvallar olib tashlangan kabi olib tashlanadi: **DROP VIEW** view_name;

28. Ma'lumotlar bazasini datalogik loyihalash. ↑ (2-savolga qarang)

29. Tasavvurlarni yaratish. ↑ (27-savolga qarang)

30. SQL so'rovlar tili – vazifasi, operatorlarni yozilishi.

SQL (Structured Query Language) tili strukturalashgan so'rov tili deyilib, u ma'lumotlar bazasi bilan aloqa o'rnatish uchun xizmat qiladi. SQL tili 70-yillar oxirida IBM firmasi tomonidan System R nomli MB boshqarish tizimining tajribaviy loyihasini ishlab chiqish doirasida yaratildi. Keyinchalik IBM firmasi tomonidan System R ga yaqin bo'lgan yana ikkita tizim — SQL/DS va DB2 tizimlari ishlab chiqildi. Bu tilning xalqaro standarti 1986-yili ishlab chiqildi va u 1989-yilga kelib yanada kengaytirildi, uning xalqaro standarti 1992-yil qabul qilindi. 1995-yilga kelib SQL92 standarti yangi komponentlar bilan to'ldirildi. Birinchi bo'lib SQL dan foydalanish Oracle MBBTda ishlatildi.

SQL quyidagi komponentlarni o'z ichiga oladi:

- administrativ ma'lumotlar vositasi;
- tranzaksiyalar bilan boshqarish vositasi;
- DDL (Data Definition Language) ma'lumotlarni aniqlash operatorlari;

— DML (Data Manipulation Language) ma'lumotlar ustida manipulyatsiya qilish (murakkab harakatlar bajarish) operatorlari;

— DQL (Data Query Language) ma'lumotlar bazasiga so'rovlar tili.

Administrativ ma'lumotlar — ma'lumotlar bazasini yaratish, unga kirishga ruxsat berish va parollarni o'zgartirishda kerak bo'ladi.

Tranzaksiya — bu ma'lumotlar bilan manipulyatsiya qilib ketma-ket operatsiyalar yordamida MBBTga ta'sir etishdir. Tranzaksiya to'liq bajariladi va ma'lumotlar bazasini bir butun holatdan ikkinchi bir butun holatga o'tkazadi, biror xatolik yuz bersa, ma'lumotlar bazasi boshlang'ich holatiga qaytadi.

Ma'lumotlarni aniqlash operatorlari — ma'lumotlar bazasi jadvalini tuzish, o'zgartirish yoki o'chirish, indeks tashkil qilish (ma'lumotlarni tez izlab topish vositasi), har xil foydalanuvchilar bilan ishni tashkil qilish va boshqalarni amalga oshiradi.

Ma'lumotlar ustida manipulyatsiya qilish operatorlari — ma'lumotlar bazasiga yozuvni qo'shish, o'chirish yoki yangilash imkonini beradi.

SQL tili operatorlarni erkin formatda yozishini ta'minlaydi. Buning ma'nosi, operatorlar elementlarini yozilishi ekrandan fiksirlangan joylarga bog'liq emas.

Komanda strukturasi bir qancha kalit xizmatchi so'zlar bilan beriladi, masalan:

CREATE TABLE (jadval yaratish), INSERT (qo'shish), SELECT (ajratib olish).

SQL operatori xizmatchi so'zlar va foydalanuvchi qo'llaydigan so'zlardan tashkil topadi. Xizmatchi so'zlar SQL tili doimiy qismi bo'lib, ular aniq qiymatga ega. Ularni standartda ko'rsatilganday yozish kerak va ularni bir satrdan ikkinchisiga ko'chirish uchun bo'linmaydi. Foydalanuvchi tomonidan aniqlangan so'zlar, foydalanuvchi tomonidan ma'lum sintaksis qoidalari asosida beriladi. Ular o'z navbatida ma'lumot baza ob'ektivlarini har xil nomlari iborat bo'ladi (jadval, ustun, tasvirlar, indekslar va hokazo). Operatoridan so'zlar o'rnatilgan sintaksis qoidalariga moslab joylashtiriladi. Til standartida bu ko'rsatilmagan bo'lsa ham, SQL tilining dialektlarida (ko'rinishida) matn tugallanganini bildiruvchi belgi, ko'pgina xollarda nuqtali vergul (;) ishlatiladi.

SQL operator komponentalarini ko'pchiligi registrga bog'liq emas, ya'ni ixtiyoriy har qanday katta va kichik harflar ishlatishi mumkin.

SQL tili erkin formatga ega bo'lgani uchun, SQL alohida operatorlari va ularning ketma-ketligini, alohida ajratib yozish va tekislab yozishni ishlatish mumkin. Quyidagi qoidalarga bo'ysunish talab etiladi:

-operatoridagi har bir konstruktsiya yangi satrdan boshlanishi kerak;

-har bir konstruktsiya boshlanishida tashlab ketladigan bo'sh pozitsiyalar, boshqa operator konstruktsiyalari ham bo'lishi kerak;

-agar konstruktsiya bir necha qismdan iborat bo'lsa, ularning har biri qism yangi satrlardan bo'sh o'rinlarni oldingi konstruktsiyaga nisbatan siljitib yoziladi.

31. SQL tilida foydalanish huquqi berish.

SQL muhitida har bir foydalanuvchi maxsus identifikatsion nom, murojaat identifikatoriga (ID) ega. Ma'lumotlar bazasiga yuborilgan komanda ma'lum

foydalanuvchi bilan yoki boshqacha aytganda maxsus murojaat identifikatori bilan bog'lanadi. SQL ma'lumotlar bazasida ID ruxsat – bu foydalanuvchi nomi va SQL komanda bilan bog'langan murojaat identifikatoriga ilova qiluvchi maxsus kalit so'z USER dan foydalanishi mumkin.

Registratsiya - bu kompyuter tizimiga kirish huquqini olish uchun foydalanuvchi bajarishi kerak bo'lgan protseduradir. Bu protsedura foydalanuvchi bilan qaysi murojaat ID si bog'lanishini aniqlaydi. Odatda har bir ma'lumotlar bazasidan foydalanuvchi o'zining IDsiga ega bo'lishi kerak va registratsiya jarayonida haqiqiy foydalanuvchiga aylanadi. Lekin ko'p masalalarga ega foydalanuvchilar bir necha murojaat IDlari bilan registratsiyadan o'tishlari yoki bir necha foydalanuvchi bitta murojaat ID sidan foydalanishlari mumkin.

Imtiyozlar - har bir foydalanuvchi SQL ma'lumotlar bazasida nima qilish mumkinligini ko'rsatuvchi imtiyozlarga egadir. Bu imtiyozlar vaqt o'tishi bilan o'zgarishi ya'ni eskilari o'chirilib, yangilari qo'shilishi mumkin. SQL imtiyozlar bu ob'ekt imtiyozlaridir. Bu shuni bildiradiki, foydalanuvchi berilgan komandani ma'lumotlar bazasining biror ob'ekti ustida bajarishi mumkin. Ob'ekt imtiyozlari bir vaqtning o'zida foydalanuvchilar va jadvallar bilan bog'liq. Ya'ni imtiyoz ma'lum foydalanuvchiga ko'rsatilgan jadvalda, asos jadvalda yoki tasavvurda beriladi. Ixtiyoriy turdagi jadvalni yaratgan foydalanuvchi shu jadval egasidir. Bu shuni bildiradiki foydalanuvchi bu jadvalda hamma imtiyozlarga ega va imtiyozlarini shu jadvalning boshqa foydalanuvchilariga uzatishi mumkin.

Foydalanuvchiga tayinlash mumkin bo'lgan imtiyozlar:

SELECT, INSERT, UPDATE, DELETE, INDEX, ALTER, SYNONYM;

REFERENCES - bu imtiyozga ega foydalanuvchi jadvalning ustunidan (yoki ustunlaridan) ajdod kalit sifatida foydalanuvchi tashqi kalit aniqlashi mumkin. Siz bu imtiyozni ayrim ustunlar uchun berishingiz mumkin.

DCL buyruqlari bir nechta foydalanuvchili ma'lumotlar bazasi muhitida ma'lumotlar bazasi xavfsizligini ta'minlash uchun ishlatiladi. DCL buyruqlarining ikki turi GRANT va REVOKE mavjud.

SQL GRANT - bu foydalanuvchilarga ma'lumotlar bazasi obyektlariga kirish yoki imtiyozlarni berish uchun ishlatiladigan buyruq.

GRANT privilege_name ON object_name TO {user_name | PUBLIC | role_name} [WITH GRANT OPTION];

privilege_name - bu foydalanuvchiga berilgan huquq yoki imtiyoz. Kirish huquqlarining ba'zilari HAMMA, YO'Q va TANLASH.

object_name - ma'lumotlar bazasi ob'ektining nomi, TABLE, VIEW, STORED PROC va SEQUENCE.

user_name - kirish huquqi berilgan foydalanuvchining ismi.

user_name - kirish huquqi berilgan foydalanuvchining ismi.

PUBLIC - barcha foydalanuvchilarga kirish huquqini berish uchun ishlatiladi.

ROLLAR - bu imtiyozlar to'plami bo'lib, ular guruhlangan.

GRANT OPTION - foydalanuvchi boshqa foydalanuvchilarga kirish huquqini beradi.

SQL REVOKE buyrug'i ma'lumotlar bazasi ob'ektlariga kirish huquqini yoki foydalanuvchini olib tashlaydi.

REVOKE buyrug'ining sintaksisi:

```
REVOKE privilege_name ON object_name FROM {user_name | PUBLIC | role_name}
```

Rollar: Rollar bu imtiyozlar yoki kirish huquqlarining to'plami. Ma'lumotlar bazasida ko'p foydalanuvchilar bo'lsa, ularga imtiyozlarni berish yoki bekor qilish qiyin bo'ladi. Shuning uchun, agar siz rollarni aniqlasangiz, foydalanuvchilarga imtiyozlarni berishingiz yoki bekor qilishingiz mumkin, shu bilan avtomatik ravishda imtiyozlar beriladi yoki bekor qilinadi. Rollarni yaratishingiz yoki Oracle tomonidan oldindan belgilangan tizim rollarini ishlatishingiz mumkin.

SQL Mexanizm foydalanuvchilarga bu imtiyozlarni **GRANT** komandasi yordamida beradi. GRANT komandasining 4 formati mavjud bo'lib, ulardan biri konkret ob'ekt ustidan, konkret foydalanuvchilarga konkret imtiyozlar berish bo'lib, quyidagi ko'rinishga ega: GRANT privileges ON tablename TO userid, ... [WITH GRANT OPTION];

Bu erda

- privileges – tayinlanayotgan imtiyozlar ro'yxati,
- tablename - jadval nomi,
- userid – imtiyozlar olgan foydalanuvchilar ro'yxati.

Masalan: GRANT SELECT, INSERT ON Orders TO Adrian, Diane;

Ma'lum foydalanuvchilarga imtiyozlarni SQL Central da ikki usul bilan tayinlash mumkin. Birinchidan, Users & Groups papkasini tanlash va ma'lum foydalanuvchi xossalarini ro'yxatdan chaqirish (sichqoncha o'ng klavishasini bosish va menyu Properties punktini tanlash). So'ngra Permissions qo'shimcha sahifasida kerakli jadvalni tanlab imtiyozni o'rnatish. Ikkinchidan, Tables yoki Views papkasida ma'lum jadval yoki tasavvur xossalar oynasini chaqirish, so'ngra Permissions qo'shimcha sahifasiga o'tish va GRANT tugmasi yordamida kerakli foydalanuvchini tanlab, imtiyozni o'rnatish.

GRANT UPDATE (City, Comm) ON Salespeople TO Diane; - bu Diane ga Salepeople jadvalining City va Comm ustunlari qiymatlarini o'zgartirish huquqini beradi.

32. SQL so'rovlar tili, ma'lumotlar bilan ishlash (manipulyatsiyalash).

SQL tilida jadvalga ma'lumotlar kiritish uchun, ularni o'zgartirish va olib tashlash uchun ma'lumotlar bilan manipulatsiya qilish (DML) tilining uchta komandasi mo'ljallangan. Bular INSERT (qo'shish), UPDATE (yangilash), DELETE (o'chirish) komandalaridir.

INSERT komandasi (buyrug'i) jadvalga yangi satr qo'shishni amalga oshiradi. Sodda holda uning ko'rinishi quyidagicha: **INSERT INTO** <jadval nomi> **VALUES** (<qiymat>, <qiymat>); Bunday yozuvda VALUES kalit so'zidan keyin qavs ichida ko'rsatilgan qiymatlar jadvaldagi yangi qo'shilgan satrning maydonlariga kiritiladi. Kiritish jadvalini CREATE TABLE operatori bilan yaratilish vaqtidagi ustunlarni ko'rsatilgan tartibda amalga oshiriladi. Masalan, STUDENT jadvaliga yangi satrni qo'shishni quyidagicha amalga oshirish mumkin:

INSET INTO Student **VALUES** (110, 'Ivanov', 'Aleksandr', 200, 3, 'Moskva', '6/10/1979', 15);

Ba'zi hollarda maydonlarning qiymatini CREATE TABLE komandasida berilgan tartibda boshqa tartibda kiritish zaruriyati paydo bo'lsa yoki qiymatlarni ba'zi bir ustunlarga kiritish talab etilsa, INSERT komandasining quyidagi ko'rinishi ishlatiladi:

INSERT INTO Student (Student_ID, CITY, SURNAME, NAME) **VALUES** (101, 'Moskva', 'Ivanov', 'Sasha');

INSERT komandasi yordamida, bir jadvaldan qiymat tanlab olib uni boshqa jadvalga joylashtirish mumkin. Masalan: **INSERT INTO** Student1 **SELECT * FROM** Student **WHERE** CITY= 'Moskva';

Jadvaldagi satrlarni olib tashlash uchun **DELETE** komandasi ishlatiladi. Quyidagi ifoda EXAM_MARKS1 jadvalidan barcha satrlarni olib tashlaydi. **DELETE FROM** EXAM_MARKS1; Buning natijasida jadval bo'sh bo'lib qoladi (bundan so'ng uni **DROP TABLE** komandasi bilan o'chirish mumkin). Jadvalda birorta shartni qanoatlantiradigan bir nechta satrni o'chirish uchun **WHERE** parametridan foydalanish mumkin: **DELETE FROM** marks **WHERE** kurs=4;

UPDATE komandasi jadval satrlari yoki mavjud satrni ba'zi bir yoki barcha maydonlarining qiymatini o'zgartirish imkonini beradi:

UPDATE universitet **SET** rating=200 **WHERE** city= 'Moskva';

UPDATE komandasi faqat 1 ta ustun emas, balki ustunlar to'plamini o'zgartirish imkonini beradi. Qiymatlarni modifikatsiya (takomillashtirilgan) qilinishi zarur bo'lgan aniq ustunlarni ko'rsatish uchun, **SET** parametri ishlatiladi.

UPDATE Subject1 **SET** name='Oliy matematika', hour=36, semestr=1 **WHERE** id=43;

UPDATE komandasining **SET** parametrida skalar ifodalarni ishlatish mumkin. Skalar ifodada maydonga qiymatning o'zgarish usulini ko'rsatadi: **UPDATE** universitet **SET** rating=rating*2; **SET** predikat hisoblanmaydi, shuning uchun unda **NULL** qiymatni ko'rsatish mumkin:

UPDATE University1 **SET** RATING=NULL **WHERE** CITY= 'Moskva';

INSERT operatorini qism so'rovi bilan ishlatish 1 ta jadvalga birdaniga bir nechta satr yuklash imkonini beradi. **VALUES** ishlatuvchi **INSERT** operatori 1 ta satr qo'shsa, **INSERT**li qism so'rov jadvaliga qism so'rov boshqa jadvaldan qancha satr ajratsa shuncha satr jadvalga qo'shadi. Bu holda qism so'rov bilan olinayotgan ustunlar soni va toifasi bo'yicha ma'lumotlari qo'shilayotgan jadvaldagi ustun soni va toifasiga mos kelishi kerak.

INSERT INTO STUDENT1 **SELECT * FROM** STUDENT **WHERE** CITY= 'Moskva';

STUDENT1 jadvaliga Moskvada o'qiyotgan barcha talabalar haqidagi ma'lumotlarni qo'shish uchun **WHERE** parametrida mos qism so'rov ishlatish mumkin. Masalan:

INSERT INTO STUDENT1 **SELECT * FROM** STUDENT **WHERE** UNIV_ID **IN** (**SELECT** UNIV_ID **FROM** UNIVERSITY **WHERE** CITY= 'Moskva');

33. SQL tilining agregat funksiyalari.

SQL **agregat (guruh)** funksiyalari ustun qiymatlaridan bitta natija qiymatini olish uchun qo'llaniladi. Sodda qilib aytganda, ushbu funktsiyalardan foydalanib, ustun qiymatlari to'plamidan bitta qiymatni olamiz. Masalan: barcha qiymatlarning yig'indisi, minimal, maksimal yoki o'rtacha qiymat, shuningdek jadvalning qatorlarini hisoblash.

Agregat funksiyalar jadvaldan yig'ilgan axborotlarni olish uchun xizmat qiladi. Bu funksiyalari jadvaldagi satrlar guruhi bilan amal bajarib, bitta natija chiqaradi. Agregat funksiyalar uchun quyidagi amallarni ishlatamiz.

1. Count - jadvaldagi satrlar sonini aniqlab beradi.
2. SUM - ko'rsatilgan maydon qiymatlarini yig'indisini hisoblaydi.
3. AVG - tanlab olingan maydon qiymatlarini o'rta arifmetigini hisoblaydi.
4. MAX (MIN) - tanlab olingan maydon qiymatlarini eng kattasini (kichigini) topib beradi.

SELECT so'rovida agregat funksiyalar maydon nomlari kabi ishlatiladi. Maydon nomlari funksiyalar argumentlari sifatida keladi.

SELECT SUM (amt) FROM Orders;

Agregat funksiyasini ishlatishda kalit so'zlardan ham foydalanish mumkin. Ana shunday kalit so'zlardan biri GROUP BY kalit so'zidir. Bu kalit so'z maxsus maydonlarda boshqa maydon terminlaridan foydalangan holda agregat funksiyalarini ishlatish imkoniyatini beradi. Bu SELECT operatorida agregat funksiya va maydon nomlarini bitta SELECT buyrug'ida ifodalash imkoniyatini yaratadi. Masalan, siz har bir sotuvchining eng qimmat qiymatda sotgan mahsulotini ko'rmoqchisiz. Buni siz har bir sotuvchini alohida tanlab (SALESPEOPLE jadvalidan), uning sotgan mahsulotlari ichidan eng qimmatini topish yo'li bilan ham aniqlashingiz mumkin. Ammo GROUP BY kalit so'zi buni bitta SELECT operatorida birlashtirib, yengil bajarish imkonini beradi:

SELECT snum, MAX (amt) FROM Orders GROUP BY snum;

GROUP BY funksiyasini bir nechta maydonlar ishtirokida ham ishlatsa bo'ladi. Masalan, siz har bir sotuvchining har bir kundagi eng ko'p savdo summasini ko'rmoqchi bo'lsangiz, quyidagicha so'rov berishingiz mumkin:

SELECT snum, odate, MAX (amt) FROM Orders GROUP BY snum, odate;

SQL tilida agregat funksiyalardan tashqari yana bir qancha funksiyalar ham ishlatiladi va ular quyidagicha sinflashtiriladi:

- matematik funksiyalar;
- qatorli funksiyalar;
- sana-vaqt funksiyalari;
- tizimni konfiguratsiyalash funksiyalari;
- tizim funksiyalari;
- xavfsizlik tizimi funksiyalari;
- ma'lumotlarni boshqarish funksiyalari;
- statistika funksiyalari.

SQL tilida **matematik funksiyalaridan** quyidagilar ishlatiladi:

ABS (son) – sonning absolyut qiymatini hisoblaydi;

ISNUMERIC (ifoda) – ifodaning sonli turga mansubligini aniqlaydi;

SING (son) – sonning ishorasini aniqlaydi;

RAND (butun son) – 0 va 1 oralig'ida verguli siljiydigan favqulotdagi sonni hisoblaydi;

ROUND (son, aniqlik) – ko'rsatilgan aniqlikda sonni yaxlitlaydi.

POWER (son, daraja) – sonni ko'rsatilgan darajaga ko'taradi;
SQRT (son) – sondan kvadrat ildiz chiqaradi;
SIN (burchak) – radianlarda ifodalangan burchak sinusini hisoblaydi;
COS (burchak) – radianlarda ifodalangan burchak kosinusini hisoblaydi;
EXP (son) – sonning eksponentasini hisoblaydi;
LOG (son) – sonning natural logarifmini hisoblaydi;

Qatorlar bilan ishlash funksiyalari:

LEN (qator) – qatordagi belgilar sonini hisoblaydi;
LTRIM (qator) – qator boshidagi bo'sh joylar (probel) larni o'chiradi;
RTLIM (qator) - qator oxiridagi bo'sh joylar (probel) larni o'chiradi;
LEFT (qator, miqdor) – qatorning chap tomonidan, ko'rsatilgan miqdordagi belgilarni qaytaradi;
RIGHT (qator, miqdor) - qatorning o'ng tomonidan, ko'rsatilgan miqdordagi belgilarni qaytaradi;
LOWER (qator) – ko'rsatilgan qator belgilarini kichik harflarga o'tkazadi;
UPPER (qator) - ko'rsatilgan qator belgilarini katta harflarga o'tkazadi;
STR (son) – son qiymatini belgi formatiga konvertirlashni amalga oshiradi;
SUBSTRING (qator, indeks, uzunlik) – qator uchun ko'rsatilgan uzunlikdagi qator qismini, aniqlashtirilgan indeksdan boshlab qaytaradi.

Tizim funksiyalari:

GETDATE () –tizim faol sanasini qaytaradi;
ISDATE (qator) – qatorning ko'rsatilgan sana formatiga moslini aniqlaydi;
DAY (sana) – ko'rsatilgan sana raqamini qaytaradi;
MONTH (sana) – ko'rsatilgan sanadagi oy raqamini qaytaradi;
YEAR (sana) – ko'rsatilgan yil raqamini qaytaradi;

Xavfsizlik tizimi funksiyalari:

SQL muhitida foydalanuvchilarning ma'lumotlarga nisbatan xavfsizligini ta'minlash uchun bir qancha funksiyalar mavjud. Shulardan asosiylarini ko'rib o'tamiz.

SQL Serverda xavfsizlikni o'rnatish uchun quyidagi funksiyalar ishlatiladi:

- GRANT - obyekt uchun bajarilishi mumkin bo'lgan vazifalarni, buyruq orqali bajarishga ruxsat berishni o'rnatadi;
- REVOKE – obyekt va buyruqlar uchun o'rnatilgan imtiyozlarni bekor qiladi;
- DENY – obyekt bilan har qanday amal bajarishni taqiqlaydi (shuni aytish kerakki, REVOKE buyrug'i faqat imtiyozlarni olib tashlayi).

Foydalanuvchiga ruxsat olish funksiyalarini quyidagi buyruqlar amalga oshiradi:

CREATE DATABASE — ma'lumotlar bazasini yaratish huquqi;
CREATE DEFAULT — jadval ustunlari uchun standart qiymatlarni o'rnatish huquqi;
CREATE PROCEDURE — saqlanadigan protseduralarni yaratish huquqi;
CREATE ROLE — jadval ustunlari uchun qoidalarni o'rnatish huquqi;
CREATE TABLE — javdallarni yaratish huquqi;
CREATE VIEW — obyekt tasvirlarini yaratish huquqi;
BACKUP DATABASE — zahira nusxalarni yaratish huquqi;

BACKUP TRANSACTION — tranzaksiyalar zahira nusxasini yaratish huquqi.

34. SQL tilining SELECT operatori.

SELECT (tanlash) SQL tilining eng muhim va ko'p ishlatiladigan operatori hisoblanadi. U ma'lumotlar bazasi jadvalidan axborotlarni tanlab olish uchun mo'ljallangan.

SELECT operatori sodda holda quyidagi ko'rinishda yoziladi.

SELECT [DISTINCT] <atributlar ro'yxati>

FROM <jadvallar ro'yxati>

[WHERE <tanlash sharti>]

[ORDER BY < atributlar ro'yxati >]

[GROUP BY < atributlar ro'yxati >]

[HAVING <shart>]

[UNION <ON SELECT operatorli ifoda>];

Bu yerda kvadrat qavslarda yozilgan elementlar har doim ham yozilishi shart emas.

SELECT xizmatchi so'zi ma'lumot bazasidan axborotni tanlab olish operatori yozilganini anglatadi. **SELECT** so'zidan keyin bir-biridan vergul bilan ajratilib so'ralayotgan maydon nomlari (atributlar ro'yxati) yoziladi. **SELECT** so'rov operatorini zarur xizmatchi so'zi **FROM** hisoblanadi. **FROM** so'zidan keyin axborot olinayotgan jadval nomlari bir biridan vergul bilan ajratilib yoziladi.

Masalan: **SELECT** Name, Surname **FROM** STUDENT;

SELECT operatorini yaratishdagi birinchi qoida, **SELECT** ifodasi o'z ichiga albatta, hech bo'lmaganda bitta, **FROM** ifodasini olishi kerak. Qolgan ifodalar kerak hollarda ishlatiladi. Ikkinchi qoida, **SELECT** kalit so'zidan so'ng chiqarilishi lozim bo'lgan ustunlar ro'yxati ko'rsatilishi lozim. Uchinchi qoida, **FROM** kalit so'zidan so'ng so'rovni bajarish uchun kerakli jadvallar nomi yoziladi.

WHERE kalit so'zidan so'ng kerakli ma'lumotlarni izlash sharti yoziladi. **GROUP BY** ifodasi guruhlar asosida natijaviy so'rovlarni yaratishga imkon beradi. **HAVING** ifodasi **GROUP BY** ifodasi bilan birgalikda ishlatilib, unda guruhlarni qaytarish sharti yoziladi. **ORDER BY** ifodasi natijaviy ma'lumotlarni olishda ularni qanday tartiblash yo'nalishini bildiradi. SQL tili muhitida har bir foydalanuvchi maxsus identifikatsiyali nom va murojaat identifikatoriga (ID) ega bo'ladi. Ma'lumotlar bazasiga berilgan buyruq ma'lum foydalanuvchi bilan yoki boshqacha aytganda maxsus murojaat identifikatori bilan bog'lanadi. SQL ma'lumotlar bazasidagi ID ruxsat – bu foydalanuvchi nomi va SQL buyrug'i bilan bog'langan murojaat identifikatoriga ilova qiluvchi maxsus kalit so'z **USER** dir.

35. Taqsimlangan ma'lumotlar bazasini qayta ishlash.

МБ нинг ривожланиш концепцияси ва ахборот ресурсларини компьютер тармоқларига интеграциялашнинг тўхтовсиз ривожланиши тақсимланган маълумотлар базаси тизимларини (ТМБ, англ. Distributed DataBase - DDB) пайдо бўлишига сабаб бўлди.

Тақсимланган маълумотлар базаси (ТМБ) деганда бир-бири билан мантиқан боғланган ва физик жиҳатдан компьютер тармоғининг турли тугунларида жойлашган МБ тушунилади.

ТМБ таркибига компьютер тармоғининг турли тугунларида жойлашган ва турлича МББТ томонидан бошқарилиши мумкин бўлган бир нечта МБ нинг фрагментларини бириктириб олади. Фойдаланувчилар ва амалий дастурлар учун ТМБ одатдаги локал МБ каби кўринади. Бу маънода «тақсимланган» атамаси МБ ни ички характеристикасини эмас, балки ташкил этиш усулини билдиради.

ТМБ тизимининг **асосий вазифаси** компьютер тармоғида тақсимланган маълумотларга мустақил мурожатларни таъминлаб бериш ва мурожатларни бошқаришдан иборат.

ТМББТ – бу дастурий комплекс (МББТ) бўлиб, у ТМБни бошқариш ва фойдаланувчига тақсимланишни шаффоф холда кўрсатади. ТМБ шаффофлиги - бу фойдаланувчи ТМБни худди марказлашган локал МБ шаклида кўришини таъминлашдир.

ТМББТнинг асосий вазифаси эса компьютер тармоғининг тугунларидаги локал МБларни интеграциялаш воситаларини таъминлаш ва бунда компьютер тармоғининг турли тугунларида жойлашган фойдаланувчилар барча МБлар билан ягона МБ каби ишлашини таъминлашдан иборат.

Бунда қуйидагилар таъминланиши лозим:

- тизимдан фойдаланишни соддалиги.
- тармоқдаги бузилишларда автоном ишлаш имкониятини таъминлаш.
- самарадорликнинг юқори даражасини таъминлаш.

ТМБ асоси иккита асосий ғоя мавжуд:

- ташкилий ва физик жиҳатдан тақсимланган кўп сонли фойдаланувчилар бир вақтнинг ўзида умумий маълумотлар (умумий МБ) билан ишлай олади;
- мантиқан ва физик тақсимланган маълумотлар (жадваллар, ёзувлар ва хатто майдонлар) биргаликда ўзаро мос ягона бутун – умумий МБ ни ташкил этади.

ТМБ ни қуришнинг асосий 3 та принципи:

- Фойдаланувчилар учун маълумотлар жойлашувининг тиниқлиги, шаффофлиги (кўринмаслиги, одатдаги МБ каби кўриниши керак);
- Фойдаланувчиларнинг бир – биридан изоляцияланганлиги, ажратилганлиги;
- синхронизация ва ихтиёрий вақт моментда маълумотлар ҳолатининг келишилганлиги.

Сўровларни тақсимлаб қайта ишлаш.

Тизим маълумотларни бирлаштириш (қўшиш) усулларини автоматик аниқлаши керак. DDB ни бу хоссаси SQL тилидаги одатдаги сўров доирасида тақсимланган

маълумотлар базаси устида танлаш амалини бажариш имконияти билан изохланади.

Тақсимланган транзакцияларни қайта ишлаш.

Тақсимланган транзакцияларни қайта ишлаш протоколи транзакцияларни қуйидаги тўртта хоссасига риоя этилишини таъминлаши керак: атомарлик, мослашганлик, изоляцияланганлик ва давомийлик. DDB ни бу хоссаси маълумотларнинг бутунлигини бузмаган холда янгилаш (INSERT, UPDATE, DELETE) амалларини бажариш имконияти билан изохланади. Бу хосса тақсимланган транзакцияларни бажаришнинг амалдаги стандарти бўлиб қолган транзакцияларни фикциялашни (two-phase commit protocol) икки фазага протоколини қўллаш орқали таъминланади. Уни қўлланилиши тақсимланган (глобал) транзакция доирасида бир нечта тугунларда маълумотларни келишилган холда ўзгартирилишини кафолатлайди.

36. SQL tilining SELECT operatori parametrlari. (24 va 34-savollarga qarang)

37. Ma'lumotlarni administratsiyalash.

MBni administratsiyalashtirish shundan iboratki, u foydalanuvchilarga ma'lumotlar bazasi imkoniyatlaridan foydalanish, ma'lumotlarning to'laligini ta'minlash va ko'p foydalanuvchilar ilovalarini yaratish huquqini beradi. Ma'lumotlar bazasi administrator quyidagi hollarda jadvallardan foydalanishi mumkin:

- ma'lumotlar bazasi obyektlari xotirasini taqsimlashni boshqarishda;
- ma'lumotlar bazasi foydalanuvchilari uchun xotira kvotalarini o'rnatishda;
- ma'lumotlarga kirishni boshqarishda *online* yoki *offline* holatlarni hisobga olgan holda;
- ma'lumotlarni nusxalash va qayta tiklashda;
- ishlab chiqarishni ko'paytirish uchun ma'lumotlarni uskunalarga taqsimlashda.

Ma'lumotlar bazasini administratsiyalashtirish - bu uni initsializatsiyalash (o'rnatilishi), unga kirishni boshqarish, ma'lumotlar bazasi butunligini ta'minlash va hokazo. Kompyuterda ma'lumotlar bazasini tashkil qilish – kompyuterga o'rnatiladigan va yagona ma'lumotlar bazasi sifatida foydalaniladigan bir yoki bir nechta ma'lumotlar faylini tayyorlashdan iboratdir. Qancha fayldan iborat bo'lishidan va undan necha marotaba foydalanishdan qat'iy nazar ma'lumotlar bazasi bir marotaba tuziladi. Agar mavjud bo'lgan ma'lumotlar bazasida ma'lumotlar o'chirilsa, ma'lumotlar bazasini yaratish protsedurasidan foydalanish mumkin. Bunda shu nom va fizik struktura bilan yangi ma'lumotlar bazasi tuziladi.

Ma'lumotlar bazasi quyidagi operatsiyalarni o'z ichiga oladi:

- yangi ma'lumotlar fayllarini yaratish yoki oldingi fayllaridagi ma'lumotlarni o'chirish;
- ma'lumotlar bazasi bilan ishlashga va unga kirishga kerak bo'lgan strukturalarni yaratish;
- boshqaruvchi fayllarni va ma'lumotlar bazasi jurnalini initsializatsiyalash.

Yaratiladigan jurnal odatda kirish haqida batafsil ma'lumotga ega bo'ladi, shu jumladan quyidagilarni o'z ichiga olgan holda:

- ma'lumotlar bazasi fayllariga kirgan normalar;

- kirish ma'lumotlari va shu bilan bog'liq bo'lgan jadvallarning aniqlanishi;
- MBning xatolari va ishlash natijalari;
- natija statistikasi.

Ma'lumotlar bazasini yaratishdan oldin quyidagilar kerak:

- indeks va jadvallarni rejalashtirish va ularga kerakli bo'lgan hajmlarni baholash;
- ma'lumotlar bazasi himoyasini va uning jurnallari konfiguratsiyasini, egallaydigan hajmini va zahira ko'chirmalarini rejalashtirish.

Ma'lumotlar bazasi operatori ma'lumotlar bazasini ishga tushirishni va ishdan to'xtatishni amalga oshiradi.

Ma'lumotlar bazasi foydalanuvchilari uch asosiy kateqoriyaga ajratiladi: yaratuvchilar, administratorlar va foydalanuvchilar.

Yaratuvchilarning vazifasiga quyidagilar kiradi:

- ma'lumotlar bazasini loyihalashtirish va yaratish;
- ma'lumotlar bazasi strukturasini (tuzilishini) belgilangan talablarga muvofiq loyihalashtirish;
- xotira hajmi talablarini baholash;
- ma'lumotlar bazasi tuzilishining modifikatsiyasini yaratish;
- ma'lumotlar bazasidan foydalanish va yaratish jarayonida uning himoya choralarini o'rnatish va hokazo.

Ma'lumot bazalarini administratsiyalashtirish shu tizimning boshqaruvchisi tomonidan amalga oshiriladi. Bu shaxs ma'lumotlar bazasi administratori deb ataladi. Agar ma'lumotlar bazasi katta bo'lsa, bu majburiyatlarni bir nechta odamlar (guruh) – ma'lumotlar bazasi (MB) administratorlari bajarishi mumkin. Administrator majburiyatlariga quyidagilar kiradi:

- MBni o'rnatish va MBBT versiyalarini yangilash;
- diskli xotirani taqsimlash va tizimni xotiraga bo'lgan talablarini rejalashtirish;
- ma'lumotlar bazasida xotira strukturasini yaratish (jadvaliy kenglik);
- birlamchi obyektlar yaratish (jadvallar, indekslar);
- ma'lumotlar bazasi strukturasini modifikatsiyalash;
- ma'lumotlar bazasini ishlab chiqishni nazorat qilish va optimallashtirish;
- litsenziyalangan kelishuvlarga amal qilish;
- foydalanuvchilarni hisobga olish va tizim himoyasini qo'llab-quvvatlash;
- foydalanuvchilarning ma'lumotlar bazasiga kirishini boshqarish va nazorat qilish;
- zahira nusxalash va qayta tiklashni rejalashtirish;
- axborotni saqlash qurilmalaridagi arxiv ma'lumotlarini qo'llab-quvvatlash;
- zahiraviy nusxalash va qayta tiklashni bajarish.

Ayrim hollarda ma'lumotlar bazasi bitta yoki bir nechta xavfsizlik xodimlariga ega bo'lishlari kerak, ya'ni asosan yangi foydalanuvchilarni ro'yxatga olishga javob beradi, foydalanuvchilarning ma'lumotlar bazasiga kirishini boshqaradi va nazorat qiladi hamda ma'lumotlar bazasini himoya qiladi.

38. Ma'lumotlarni xavfsizligini ta'minlash. (25-savolga qarang)

39. Ma'lumotlar bazasi xavfsizligini ta'minlash. (25-savolga qarang)

40. Tranzaksiyalarni qayta ishlash.

Ma'lumotlar bilan ishlaganda ma'lumotlarni yo'qotish yoki uzilish ehtimoli bor. Bu ehtimollik MB qancha katta bo'lsa va unga beriladigan so'rovlar murakkab bo'lganda ortadi. MBBTda ma'lumotlarni maxsus himoyalash vositalari bor, lekin ba'zi hollarda SQL yordamida tashkil qilinadigan himoya usullaridan foydalanish mumkin. Bunda bir necha SQL operatorlari **tranzaksiya** deb nomlanuvchi bitta blokka birlashtiriladi. Tranzaksiyada yoki barcha operatorlar bajariladi, yoki birortasi ham bajarilmaydi. Oxirgi holda MB tranzaksiya bajarishgacha bo'lgan boshlang'ich holatga qaytariladi. Boshqacha aytganda tranzaksiyada birorta operator bajarilmasa, unda bu tranzaksiyada barcha bajarilgan operatorni barcha ishlari bekor qilinadi. Buni otkat (orqaga qaytish) deyiladi. Tranzaksiyada SQL barcha operatorlarini ishlatish mumkin. Bundan tashqari COMMIT - bajarishni yakunlash, ROLLBACK - orqaga qaytish komandalari ham ishlatiladi.

Har bir SQL tranzaksiyasi himoyalash darajasiga ega: READUNCOMMITTED, READCOMMITTED, REPEATABLEREAD yoki SERIALIZABLE. SQL tranzaksiyasi himoyalash darajalari bajarilayotgan tranzaksiyaning boshqa parallel bajarilayotgan tranzaksiyalarga ta'sir etish darajasini aniqlaydi. Tranzaksiyaning aniq darajasini o'rnatish uchun SETTRANSACTION operatoridan foydalaniladi. Jimlik qoidasiga nisbatan SERIALIZABLE tartibi o'rnatiladi.

Himoyalash darajalari tranzaksiyalarning parallel bajarilishida yuz berishi mumkin bo'lgan hodisalarni aniqlaydi. Quyidagi ko'rinishdagi hodisalar bo'lishi mumkin:

1. P1 ("Dirtyread"- "Yomon o'qish"): T1 tranzaksiya qatorini yaratadi. Keyin T2 tranzaksiya T1 COMMIT amalini bajarmasdan bu qatorni o'qiydi. Shundan so'ng T1 ROLLBACK amalini bajarsa, T2 tranzaksiya umuman mavjud bo'lmagan qatorni o'qigan bo'lib chiqadi.

2. P2 ("Non-repeatableread" - "Takrorlanmaydigan o'qish"): T1 tranzaksiya qatorni o'qiydi. Shundan so'ng T2 tranzaksiya bu buyruqlar qatorini o'zgartiradi yoki olib tashlaydi va COMMITni bajaradi. Shundan so'ng T1 shu qatorni yana o'qishga harakat qiladi, ammo bu qator birinchi holatdagi qator emas yoki olib tashlangani uchun topolmaydi.

3. P3 ("Phantom"): T1 tranzaksiya biror-bir shartni qanoatlantiradigan N qatorni o'qiydi. Shundan so'ng T2 tranzaksiya bu qatorlar ichidan bir yoki bir nechta qator shartlarini generatsiya qiladi. Agar shu ishlardan keyin T1 o'qishni qaytarsa, u butunlay boshqa qatorlarga ega bo'ladi.

MB bilan ish jarayonida ma'lumotlar butunligi muhim o'rin tutadi. Ma'lumotlar butunligi deganimizda, ma'lumotlarning to'g'riligi va mazmunan qarama-qarshi ma'noga ega emasligi tushuniladi. Masalan, "O'qituvchi" jadvalidagi har bir o'zgarish "Yuklama" jadvalida ham qayd etilishi kerak. O'qituvchining "Yuklama" jadvalida qayd etilmasligi ma'lumotlar butunligining buzilishiga olib keladi. Ko'p hollarda MBning ma'lumotlari butunligini saqlashni tashkillashtirish uchun tranzaksiyalardan foydalanishadi.

Umuman olganda tranzaksiya - bu mantiqan bo'linmaydigan ish birligi. Bu jarayonda:

- yoki tranzaksiyaga kiruvchi barcha amallar MBda aks etadi;

- yoki bu amallar umuman bajarilmaydi.

Tranzaksiyaning bu xususiyati butunlik shartining buzilmasligini ta'minlaydi.

Ko'pgina MBBTda tranzaksiyalarning ikkita modeli ishlatiladi:

1. Tranzaksiyalarning avtomatik bajarilish modeli.

2. Tranzaksiyalarning bajarilishini boshqarish modeli.

Tranzaksiyalarning avtomatik bajarilish modelida, tranzaksiya avtomatik ravishda ishga tushadi va quyidagi usullardan biri bilan tugaydi:

- COMMIT yo'riqnomasi bilan, bunda MBdagi o'zgarishlar doimiy bajariladigan bo'ladi va yangi tranzaksiya COMMIT buyrug'idan so'ng boshlanadi.

- ROLLBACK yo'riqnomasi bilan, bunda tranzaksiyada bajarilgan barcha o'zgarishlar bekor bo'ladi va yangi tranzaksiya ROLLBACK buyrug'idan so'ng boshlanadi.

Tranzaksiyalarning bajarilishini boshqarish modeli SUBD Sybase dasturida qo'llanilib, quyidagi yo'riqnomalardan foydalaniladi:

1. BEGIN TRANSACTION yo'riqnomasi, tranzaksiyaning boshlanishini bildiradi.

2. COMMIT TRANSACTION yo'riqnomasi, tranzaksiyaning muvaffaqiyatli tugaganini bildiradi. Bunda yangi tranzaksiya avtomatik ravishda ishga tushmaydi.

3. SAVE TRANSACTION yo'riqnomasi, bunda tranzaksiya ichida saqlash nuqtasi tashkillashtiradi va saqlangan saqlash nuqtasiga nom berish imkoniyati yaratiladi.

4. ROLLBACK yo'riqnomasi, bunda barcha tranzaksiyadagi amallar bekor qilinadi va MB holati tranzaksiyadan oldingi holatga qaytariladi.

Shunday qilib, tranzaksiya - bu MBga tugallangan murojaat bo'lib, quyidagi to'rtta shartning bajarilishini kafolatlaydi:

- *Bo'linmaslik* - tranzaksiya boshi va oxiriga ega bo'lgan bo'linmas blok. Bu blok yoki to'liqligicha bajariladi, yoki umuman bajarilmaydi;

- *Kelishuvchanlik* - tranzaksiya tugaganidan so'ng, hamma obyektlar kelishganlik holatini saqlab qoladi;

- *Himoyalanganlik* - har bir tranzaksiya jarayoni boshqa tranzaksiya ishiga ta'sir ko'rsatmaydi.

- *Doimiylik* - tranzaksiya jarayonida bajarilgan barcha o'zgarishlar doimiylik xarakterga ega.

41. Parallel qayta ishlash.

Ko'p foydalaniluvchi ma'lumot bazalari tashkilot va korxonalarning muhim axborot resurslari hisoblanadi. Shu bilan birga bunday resurslardan foydalanish bir qancha muammolar keltirib chiqaradi. Birinchidan, bunday ma'lumot bazasini ishlab chiqish va loyihalash murakkab, chunki bunda ko'p sondagi foydalanuvchilar talablarini hisobga olish kerak. Shu bilan birga vaqt o'tishi bilan barcha talablar o'zgaradi. Bu esa o'z navbatida ma'lumot baza strukturasini o'zgartirish zaruratiga olib keladi. Bunday strukturalarni o'zgartirish aniq rejalashtirishni va nazorat qilishni talab etadi. Chunki bir guruh foydalanuvchilar uchun bajarilgan o'zgartirishlar boshqa guruh foydalanuvchilari uchun muammolar tug'dirmasligi talab etiladi. Bundan tashqari bir necha foydalanuvchilar

so'rovlarini parallel qayta ishlashda maxsus tadbirlar qabul qilish kerak. MB tashkilot faoliyatining muhim komponenti bo'lib xizmat qiladi. Uning tarkibi buzulishi bilan axborotlarning yo'qolishi kabi nuqsonlar paydo bo'ladi. Shu bilan birga operatsion tizimlardagi takomillashtirishlar va dasturiy ta'minotning yangi versiyalari paydo bo'lishi ma'lumot bazasini takomillashtirishni talab etadi. Bunday masalalarni yechish uchun tashkilot va korxonalarda ma'lumot bazasini administratsiyalash bo'limlari tuzilgan. Bu bo'limning asosiy vazifasi ma'lumot bazasi va uni ishlatadigan ilovani yaratishni, ishlatishni va ma'lumot bazasiga xizmat ko'rsatishni amalga oshirishdan iborat. Jumladan:

- ma'lumot baza strukturasini boshqarish;
- parallel qayta ishlashni boshqarish;
- qayta ishlash bo'yicha huquq va vazifalarni (burchlarni) belgilash;
- ma'lumot baza xavfsizligini ta'minlash va h.k.

Parallel qayta ishlashni boshqarish bo'yicha asosiy tadbirlar birorta foydalanuvchini boshqasiga kutilmagan holda ta'sir qilishini hisobga olishga qaratilgan. Parallel qayta ishlash sharoitida ba'zan foydalanuvchilar ma'lumot bazasiga yakka holda foydalanish sharoitidagi natijalarni olish maqsadiga qaratilgan. Boshqa hollarda foydalanuvchilarning kutilgan tarzda bir biriga ta'siri tushuniladi. Masalan, komputer tizimiga o'z ma'lumotlarini kiritgan foydalanuvchi ma'lumot bazasidan bir qancha foydalanuvchi foydalanishiga qaramasdan bir xil natija olishi kerak. Shu bilan birga foydalanuvchi masalan, omborxona holati haqidagi axborotning oxirgi vaqtdagi hisobotini olishga harakat qilishi mumkin. Bunda shu paytda boshqa foydalanuvchi ma'lumotlarini o'zgartirayotgan bo'lsa, amalga oshirayotganligini tugallanmagan o'zgarishlarni hisobga olib ko'rishga harakat qiladi.

Hozirgi kunda barcha parallel qayta ishlash hollari uchun ma'qul bo'ladigan usullar mavjud emas. Masalan, foydalanuvchi parallel qayta ishlashni qattiq boshqarishi mumkin. Bunda u ma'lumot bazani boshqa foydalanuvchilar uchun blokirovka qilib qo'yishi mumkin, o'zining ishi tugamaguncha boshqa mijozlar ma'lumotlar bazasi bilan hech qanday ish bajara olmaydi. Bu usul ishonchli bo'lishi bilan birga qimmat hisoblanadi. Ishlab chiqarish samaradorligini oshiradigan, lekin dasturlash va amalga oshirish qiyin usullar ham mavjud. Ma'lumot bazasi bilan ishlaydigan ko'pgina ilovalarda foydalanuvchi ishini tranzaksiya ko'rinishida amalga oshiriladi. Tranzaksiya ba'zan ishni mantiqiy birligi hisoblanadi. **Tranzaksiya** - bu ma'lumot bazasi bilan bajariladigan ishlar (harakatlar) ketma-ketligi bo'lib, unda barcha ishlar to'laligicha bajariladi yoki birortasi ham bajarilmaydi. Oxirgi holda ma'lumot bazasi o'zgarishsiz qoladi. Bunday tranzaksiya ba'zan **atomar** deb ataladi. Chunki u bir butun ko'rinishda bajariladi.

Tranzaksiyani bajarish chegaralarini ko'rsatish uchun tranzaksiyani boshlashga, tranzaksiyani saqlashga va tranzaksiyani qaytarishga komanda berish zarur. Har xil MBBTda bu komandalar formasi bo'yicha farqlanadi. Agar bir vaqtda ikkita tranzaksiya bajarilsa, bu tranzaksiya **parallel tranzaksiya** deyiladi. Himoya faqat MB ma'lumotlarigagina tegishli emas. Himoyalash hisoblash tizimlarining boshqa qismlarini ham o'z ichiga oladi. Shuning uchun MB himoyasi hisoblash qurilmalariga, dasturiy ta'minotga va bevosita ma'lumotlarga tegishli bo'lishi mumkin.

42. Obyektlarga o'zgartirish kiritish.

Ma'lumotlar bazasi obyektlarini yaratish ma'lumotlarni aniqlash tili (DLL) operatorlari yordamida bajariladi. Ma'lumotlar bazasi jadvallari yaratish **CREATE TABLE** komandasi yordamida amalga oshiriladi. Bu komanda bo'sh jadval yaratadi ya'ni jadvalda satrlar bo'lmaydi. Bu jadvalga qiymatlar **INSERT** komandasi yordamida kiritiladi. **CREATE TABLE** komandasi jadval nomini va ko'rsatilgan tartibda nomlangan ustunlar to'plamini aniqlaydi. Har bir ustun uchun tip (toifa) va o'lcham aniqlanadi. Har bir yaratilgan jadval hech bo'lmaganda bitta ustunga ega bo'lishi kerak.

CREATE TABLE komanda ko'rinishida quyidagicha:

CREATE TABLE <jadval nomi> (<ustun nomi><ma'lumot toifasi> [<o'lchami>]);

CREATE TABLE xususiyati quyidagicha: SQL ishlatilayotgan ma'lumot toifalari eng kam holda ANSI standartida berilgan.

Char (character); Int (integer); Small int; Dec (decimal); Number; Float; ...

Student1 jadvalini quyidagi komanda bilan yaratish mumkin: **CREATE TABLE Student1 (student_id INTEGER, surname VARCHAR(60), name VARCHAR (60), stipend DOUBLE, kurs INTEGER, city VARCHAR(60), birthday DATE, univ_id INTEGER);**

Jadvaldagi ma'lumotlarni maydonlar bo'yicha qidirish-tanlash amalini yetarli darajada tezlatish uchun ma'lumotlarni berilgan maydon bo'yicha indeksatsiya qilish ishlatiladi. Indeksar bitta yoki bir nechta maydon bo'yicha bajarilishi mumkin.

Index komandasining ko'rinishi:

CREATE INDEX <index nomi> **ON** <jadval nomi> (<ustun nomi>, <ustun nomi>);

Bu komanda bajarilishi uchun jadval yaratilgan bo'lishi kerak va indeksda ko'rsatilgan ustunlar unda bo'lishi kerak.

Indeksni olib tashlash uchun (bunda uning nomini albatta bilish kerak) quyidagi komanda ishlatiladi:

DROP INDEX <INDEKS NOMI>;

Mavjud jadval strukturasini va parametrlarini modifikatsiyalash uchun **ALTER TABLE** komandasi ishlatiladi. Masalan, jadvalga ustunlar qo'shish uchun **ALTER TABLE** komandasi quyidagicha bo'ladi:

ALTER TABLE <jadval nomi> **ADD** (<ustun nomi> <ma'lumot tipi> <o'lchami>);

Bu komanda bo'yicha mavjud jadval satrlariga yangi ustun qo'shiladi va unga NULL qiymati yoziladi. Jadvalga bir nechta ustun ham qo'shsa bo'ladi. Ular bir-biridan vergul bilan ajratiladi.

Ustun tavsifini o'zgartirish mumkin. Ko'pincha bu ustun o'lchamlarini o'zgartirish, ulardagi qiymatlarga cheklanishlarni qo'shish yoki olib tashlash bilan bog'liq bo'ladi. Komanda bu holda quyidagicha bo'ladi:

ALTER TABLE <jadval nomi> **MODIFY** (<ustun nomi> <ma'lumot tipi> <o'lcham/aniqlik>);

Ustun xarakteristikalarini modifikatsiyalashda quyidagi cheklanishlarni hisobga olish kerak:

- ma'lumot toifalarini o'zgartirishni, faqat ustun bo'sh bo'lsa bajarish mumkin;
- to'ldirilmagan ustun uchun o'lcham / aniqlik o'zgartirishi mumkin;
- to'ldirilgan ustun uchun o'lcham / aniqlik faqat kattalashtirishi mumkin;

– NOT NULL oʻrnatilishi uchun ustunda birorta ham NULL qiymat boʻlmasligi kerak. NOT NULLni har doim bekor qilish mumkin;

– sukut bilan oʻrnatilgan qiymatni har doim oʻzgartirish mumkin.

Mavjud jadvalni olib tashlash uchun, oldin bu jadvaldan barcha maʼlumotlarni olib tashlash kerak, yaʼni uni boʻshatish kerak. Satrlarga ega boʻlgan jadvalni olib tashlab boʻlmaydi. Boʻsh jadvallarni olib tashlash quyidagi komanda bilan bajariladi: **DROP TABLE** <jadval nomi>

43. SQL tilida INDEXni oʻrni va vazifasi.

Indekslar - jadvallarning tepasida joylashgan maxsus obyektlar. Koʻrsatkichlar SELECT, DELETE va UPDATE kabi tezkor operatsiyalarni bajarishi mumkin, bu katta miqdordagi maʼlumotlarni boshqarishga imkon beradi. INDEXni jadval deb ham atash mumkin va u maʼlumotlar tuzilishiga ega. Jadval ustunlarida INDEX tuziladi. Bitta jadvalda bir yoki bir nechta INDEX jadval boʻlishi mumkin. Indekslar maʼlumotlar bazasidan maʼlumotni tezroq olish uchun ishlatiladi. Foydalanuvchilar indekslarni koʻra olmaydilar, ular qidiruv / soʻrovlarni tezlashtirish uchun ishlatiladi.

SQL INDEX nima qiladi?

SQL INDEX quyidagilarni bajaradi:

- Indekslar maʼlumotlar bazasida maʼlumotlarni juda tez topishi mumkin.
- INDEX maʼlumotlar bazasi jadvallari satrlarining katalogini tuzadi, chunki qisqa vaqtda harakat bilan satrni belgilash mumkin.
- INDEX jadvali maʼlum bir tartibda bir yoki bir nechta ustunlarning qiymatlarini tartibga soladigan maʼlumotlar bazasining tuzilishi.
- Nisbatan kichik jadvallar bilan ishlashda INDEX koʻrsatkichlarini aniqlab boʻlmaydi.
- INDEX juda koʻp turli xil qiymatlarga ega boʻlgan ustunlar uchun toʻgʻri va tez ishlashi mumkin.

Jadvalda minglab yozuvlar mavjud boʻlsa, jadvaldan bitta yoki ustunlar birikmasi uchun maʼlumot topish uzoq vaqt talab etadi. Bunday holda, agar ushbu ustunga tez-tez kiradigan indekslar yaratilsa, maʼlumotni tezda olish mumkin.

INDEX avval maʼlumotlarni tartiblaydi va keyin har bir satr uchun identifikatsiyani tayinlaydi. INDEX jadvali faqat ikkita ustunga ega, ulardan biri ustunli, ikkinchisi esa indekslangan (ustun).

Indekslangan ustun asosida maʼlumotlar bazasi jadvalidan maʼlumotlar olinsa, indeks koʻrsatgichi satrni qidiradi va ushbu pozitsiyani tezda topadi.

```
CREATE [UNIQUE] INDEX <index name> ON <table name> (<column(s)>);
```

Parameters:

Name	Description
UNIQUE	Defines the index as a unique constraint for the table and disallows any duplicate values into the indexed column or columns of the table.
index_name	Name of the index table.
table_name	Name of a base table.
column(s)	Name of the columns of the table.

INDEX va VIEWni qanday ajratish mumkin?

VIEW:

VIEW - bu ma'lumotlar ob'ekti bo'lib, unda ma'lumotlar yo'q. Uning tarkibi asosiy jadvalning natijasidir. Ular bazaviy jadval kabi ishlaydi, lekin ularda o'zlariga tegishli ma'lumotlar yo'q. VIEW jadvalga o'xshaydi, lekin umumiy mezonlar to'plami orqali o'zaro bog'langan bir yoki bir nechta jadvallarning ma'lumotlarini o'z ichiga olishi mumkin.

VIEW - bu jismoniy ob'ekt, ammo bu mantiqiy jadval. VIEW faqat bazaviy jadvallarda saqlanadigan ma'lumotlarga ishora qiladi.

VIEW ham ma'lumotlar bazasi ob'ektlaridan biridir.

VIEW dasturini jadval kabi istalgan SELECT so'rovida ishlatish mumkin.

VIEW ma'lumotlar va jadvallar bazasining xavfsizligini ta'minlaydi. Aytaylik, xato bilan stol tushirilsa, uni tiklab bo'lmaydi. lekin agar ko'rinishi pasayib ketgan bo'lsa, uni yana yaratish mumkin.

INDEX:

INDEX ham jadval. Shunday qilib, u ma'lumotlar tuzilishiga ega. Indeksar - bu ma'lumotlarning haqiqiy manzilini ko'rsatadigan ko'rsatkichlar. Jadval ustunlarida INDEX tuziladi. INDEX jadvalning bir yoki bir nechta ustunlari asosida katalog yaratadi. Bitta jadvalda bir yoki bir nechta INDEX jadval bo'lishi mumkin. INDEX bitta ustunda yoki ma'lumotlar bazasi jadvalining ustunlari kombinatsiyasida yaratilishi mumkin

```
CREATE INDEX index_name ON table_name (column1, column2, ...);
CREATE UNIQUE INDEX index_name ON table_name (column1, column2, ...);
CREATE INDEX idx_lastname ON Persons (LastName);
CREATE INDEX idx_pname ON Persons (LastName, FirstName);
DROP INDEX index_name ON table_name;
DROP INDEX table_name.index_name;
ALTER TABLE table_name
DROP INDEX index_name;
```

44. UNIQUE komandasining vazifasi.

UNIQUE cheklovi ustundagi barcha qiymatlar turlicha bo'lishini ta'minlaydi. UNIQUE va PRIMARY KEY cheklovlari ustunlar yoki ustunlar to'plamining noyobligini kafolatlaydi. PRIMARY KEY cheklovi avtomatik ravishda UNIQUE chekloviga ega. Biroq, har bir jadvalda UNIQUE cheklovlari bo'lishi mumkin, ammo bitta jadvalda faqat bitta PRIMARY KEY cheklovi

mavjud. UNIQUE cheklovi ikkita yozuvni ustunlarda bir xil qiymatlarga ega bo'lishiga to'sqinlik qiladi. Mijozlar jadvalida, masalan, ikki yoki undan ortiq odamning bir xil yoshga ega bo'lishining oldini olishni xohlashingiz mumkin. Masalan, quyidagi SQL so'rovi CUSTOMERS deb nomlangan yangi jadval yaratadi va besh ustun qo'shadi. Bu yerda AGE ustuni UNIQUE ga o'rnatiladi, shunda siz bir xil yoshdagi ikkita yozuvga ega bo'lolmaysiz.

```
CREATE TABLE CUSTOMERS (  
  ID INT NOT NULL,  
  NAME VARCHAR (20) NOT NULL,  
  AGE INT NOT NULL UNIQUE,  
  ADDRESS CHAR (25),  
  SALARY DECIMAL (18, 2),  
  PRIMARY KEY (ID) );
```

Agar mijozlar jadvali allaqachon yaratilgan bo'lsa, unda AGE ustuniga UNIQUE cheklovini qo'shish kerak.

```
ALTER TABLE CUSTOMERS MODIFY AGE INT NOT NULL UNIQUE;
```

Cheklovni bir nechta ustunlarga nomlashni qo'llab-quvvatlaydigan quyidagi sintaksisdan ham foydalanishingiz mumkin.

```
ALTER TABLE CUSTOMERS  
  ADD CONSTRAINT myUniqueConstraint UNIQUE (AGE, SALARY);
```

UNIQUE cheklovini olib tashlash uchun quyidagi SQL so'rovidan foydalaning.

```
ALTER TABLE CUSTOMERS DROP CONSTRAINT myUniqueConstraint;
```

Agar siz MySQL-dan foydalanayotgan bo'lsangiz, quyidagi sintaksisdan foydalanishingiz mumkin: **ALTER TABLE CUSTOMERS DROP INDEX myUniqueConstraint;**

45. 2normal forma va uning tashkil etuvchilari.

Normallashtirish — bu ma'lumotlarni qo'shishda, o'zgartirishda va o'chirishda eng yaxshi xususiyatlarga ega ikki yoki undan ortiq bo'laklarga jadvalni bo'lish. Normallashtirishning asosiy maqsadi ma'lumotlar bazasini olishga qaratilganki, unda har bir dalil (fakt) faqat bir joyda uchraydi, ya'ni ma'lumotlar ortib ketmaydi. Bu faqat xotiradan tejab foydalanish maqsadida qilinmay, balki saqlanayotgan ma'lumotlar orasida qarama-qarshiliklarni bartaraf qilish uchun ham kerak.

Har bir jadval relyatsion ma'lumotlar bazasida shunday shartlarni qoniqtiradiki, unga ko'ra jadvalning har bir ustun va satrining kesishish joyida har doim yagona atomar qiymat joylashadi va hech qachon ko'p miqdorda, xuddi shunday qiymatlar bo'lishi mumkin emas. Shu shartni qoniqtiruvchi har qanday jadval **normallashtirilgan** deyiladi. Umuman olganda normallashtirmagan jadvallar, ya'ni takrorlanuvchi ma'lumotlar guruhiga ega jadvallar relyatsion ma'lumotlar bazasiga kiritilmaydi.

Har qanday normallashtirilgan jadval avtomatik ravishda birinchi normal formada, qisqacha 1NF, deb hisoblanadi. Shunday qilib, umuman olganda, "normallashtirilgan" va "1NF da joylashgan" so'zlari bitta ma'noni anglatadi. Lekin amaliyotda "normallashtirilgan" atamasi ko'proq tor ma'noda — "to'liq normallashtirilgan", ya'ni loyihada normallashtirishning hech qaysi tamoyili buzilmayapti degan ma'noda ishlatiladi.

Keyingi pog'onadagi normallashtirishlarni ko'rib chiqamiz — ikkinchi normal forma (2NF), uchinchi normal forma (3NF) va hokazo. Umuman olganda, jadval 1NF da bo'lsa va undan tashqari yana bir qo'shimcha shartni qoniqtirsa, u jadval 2NFda deyiladi. Jadval 3NF da deyiladi, qachonki u 2NF da bo'lsa va yana bir qo'shimcha shartni qoniqtirsa va hokazo.

Har bir normal forma qandaydir ma'noda oldingisiga qaraganda ancha chegaralangan, lekin ma'qulroqdir. Bu shunga bog'liqlik, "N-normal forma" ega bo'lgan ba'zi yoqimsiz tomonlariga "(N+1)-normal forma" ega emas. N-chi normal formaga qarab (N+1)-normal formaga qo'yilgan qo'shimcha shartning umumiy ma'nosi ana shunday yoqimsiz tomonlarni yo'qotishdan iborat.

Normallashtirish nazariyasi jadval maydonlari orasidagi u yoki bu bog'liqlikning borligiga asoslanadi. Bunday bog'liqliklarning ikki turi aniqlangan: **funksional va ko'p qiymatli**.

Funksional bog'liqlik. Berilgan ixtiyoriy vaqtda A maydonning har bir turli qiymatiga mos ravishda albatta B maydonning har bir turli qiymatidan faqat bitta qiymati bo'lsa, jadvalning B maydoni xuddi shu jadvalning A maydoniga **funksional bog'langan** deyiladi. Aytib o'tish kerakki, A va B maydonlar tashkil qiluvchilar bo'lishi mumkin.

To'liq funksional bog'liqlik. B maydon agar A ga funksional bog'liq bo'lsa va A maydonning boshqa qiymatlariga bog'liq bo'lmasa, A tashkil qiluvchiga **to'liq funksional bog'liq** bo'ladi.

Ko'p qiymatli bog'liqlik. Agar A maydonning har bir qiymatiga mos ravishda B maydonning aniqlangan qiymatlari bor bo'lsa, A maydon xuddi shu jadvalning B maydonini **ko'p qiymatli** aniqlaydi.

Birinchi normal forma: Jadval, qachonki uning qatorlaridan hech biri o'zining istalgan maydonlarida bittadan ortiq qiymat saqlamasa va uning birorta ham kalit maydoni bo'sh bo'lmasa, birinchi normal formada (1NF) bo'ladi.

Ikkinchi normal forma: Jadval, agar u 1NF ta'rifini qanoatlantirsa va uning barcha birinchi kalitga kirmaydigan maydonlari birinchi kalit bilan to'liq funksional bog'langan bo'lsa ikkinchi normal formada (2NF) bo'ladi.

Uchinchi normal forma: Jadval, agar u 2NF ta'rifini qanoatlantirsa va uning birorta ham kalit bo'lmagan maydonlari boshqa istalgan kalit bo'lmagan maydonlariga funksional bog'liq bo'lmasa, uchinchi normal formada (3NF) bo'ladi.

Boys-Kodd normal formasi: Jadval, qachonki uning maydonlari orasidagi har qanday funksional bog'lanishdan mumkin bo'lgan kalitdan to'liq funksional bog'lanish kelib chiqsa, Boys-Kodd normal formasida (BKNF) bo'ladi.

Beshinchi normal forma: Jadval, qachonki uning har bir to'liq dekompozitsiyasining barcha proyeksiyalari mumkin bo'lgan kalitni saqlasa, beshinchi normal formada (5NF) bo'ladi.

To'rtinchi normal forma: Beshinchi normal formaning (5NF) xususiy holi bo'lib, to'liq dekompozitsiyasi ikkita proyeksiyasiga birlashmagan bo'lishi shart. 4NF va 5NFda nafaqat funksional bog'lanish, balki yana jadval maydonlari orasidagi ko'p qiymatli bog'lanish ham hisobga olinadi.

Normallashtirish protsedurasi

Normallashtirish — bu ma'lumotlarni yangilanishda, qo'shishda va o'chirishda jadvallarni bir nechtaga bo'lishdir. Normallashtirish jadvallarni barcha dekompozitsiyalari bilan ketma-ket almashtirish jarayoni bo'lib, bu almashtirish jadvallarning barcha dekompozitsiyalari

5NF da joylashmaguncha davom etadi. Amaliyotda esa jadvallarni BKNF ga keltirish kifoya va katta kafolat bilan aytish mumkinki, ular 5NF da joylashadi.

Jadvallarni BKNF ga o'tkazish protsedurasi

Bu protsedura, ixtiyoriy jadvalda yagona funksional bog'liqliklar $K \rightarrow F$ ko'rinishda bo'lishga asoslanadi, bunda K — dastlabki kalit, F — esa boshqa maydon. Shuni inobatga olish kerakki, bu dastlabki kalit ta'rifidan kelib chiqadi, ya'ni $K \rightarrow F$ bog'lanish berilgan jadvalning barcha maydonlari uchun bajariladi. "Bir joyda bir fakt" degani boshqa funksional bog'liqliklar hech qanday kuchga ega emas degan ma'noni bildiradi. Normallashtirishdan maqsad $K \rightarrow F$ ko'rinishdagi bog'lanishdan tashqari boshqa bog'lanishlardan voz kechishdir.

Agar normallashtirish vaqtida birlamchi (tashqi) kalitlar kodlarini boshlang'ich kalitlarnikiga almashtirilsa, u holda quyidagi ikki holni ko'rib chiqish kerak bo'ladi:

1. Jadval birinchi tarkibli kalit ko'rinishida bo'lsin, aytaylik: K_1, K_2 . Bundan tashqari, bu kalitning biror qismiga, masalan, K_2 funksional bog'liq, lekin to'la kalitga bog'liq bo'lmagan F maydonni o'z ichiga olsin. Bu holda K_2 va F (K_2 —birlamchi kalit)ni o'z ichiga oluvchi boshqa jadval tashkil qilish taklif etiladi va boshlang'ich jadvaldan F o'chiriladi:

$T(K_1, K_2, F)$, birlamchi kalit (K_1, K_2) , $\Phi_3 K_2 \rightarrow F$ ni $T_1(K_1, K_2)$, birlamchi kalit (K_1, K_2) va $T_2(K_2, F)$, birlamchi kalit K_2 ga almashtirilsin.

2. Jadval birlamchi (mumkin bo'lgan) K kalitga ega, bu kalit F_1 maydonning mumkin bo'lmagan kaliti, F_1 maydon esa albatta K ga funksional jihatdan bog'liq va boshqa kalitsiz F_2 maydoni esa F_1 ga funksional jihatdan bog'liq. Bu yerda yechim, aslida, avvalgining o'zi — F_1 va F_2 ni o'zida aks ettirgan boshqa jadval shakllantiriladi, boshlang'ich kalit bilan F_1 va F_2 maydonlari boshlang'ich jadvaldan olib tashlanadi: **$T(K, F_1, F_2)$, birinchi kalit K , $\Phi_3 F_1 \rightarrow F_2$ ni $T_1(K, F_1)$, birinchi kalit K , va $T_2(F_1, F_2)$, birinchi kalit F_1 ga almashtirilsin.**

Berilgan har qanday jadval uchun, ko'rib chiqilgan qoidalarga asoslangan o'zgaruvchilarni takrorlab, deyarli barcha holatlarda ko'p jadvallarni olish mumkin, ular "oxirgi" normal ko'rinishda (formada) bo'ladi va shunday qilib, $K \rightarrow F$ dan farqli bo'lgan hech qanday funksional bog'lanishga ega bo'lmaydi.

46. 1normal forma va uning tashkil etuvchilari. (45-savolga qarang)

47. 3normal forma va uning tashkil etuvchilari. (45-savolga qarang)

48. 1normal formadan 2normal formaga o'tish. (45-savolga qarang)

49. 2normal formadan 3normal formaga o'tish. (45-savolga qarang)

50. 1normal formadan 2normal formaga o'tish qanday amalga oshiriladi? Misollar keltiring. (45-savolga qarang)