

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

GAIPNAZAROV TAHIRITDIN SIDIKOVICH
IRGASHEVA DURDONA YAKUBDJANOVNA

MA'LUMOTLAR OMBORI

*O'zbekiston Respublikasi Oliy va o'rta maxsus ta'lif vazirligi
tomonidan kasb-hunar kollejlarining 32521900 – Axborot tizimlari
va texnologiyalari yo'nalishi o'quvchilari uchun
o'quv qo'llanma sifatida tavsiya etilgan*

O'ZBEKISTON FAYLASUFLARI MILLIY JAMIYATI NASHRIYOTI
TOSHKENT – 2017

UO‘K: 004.65(075)

KBK: 32.973 202

G-11

Gaipnazarov, Tahiritdin Sidikovich

G-11 Ma'lumotlar ombori: kasb-hunar kollejlari uchun o'quv qo'llanma/ T.S. Gaipnazarov, D.Y. Irgasheva; O'zbekiston Respublikasi Oliy va o'rta maxsusus ta'lif vazirligi. — Toshkent: «O'zbekiston faylasuflari milliy jamiyati» nashriyoti, 2017. — 224 bet.

UO‘K: 004.65(075)

KBK: 32.973 202

O'quv qo'llanmada ma'lumotlar bazasining nazariy asoslari, Microsoft Access ma'lumotlar bazasining boshqarish tizimlarida relatsion ma'lumotlar bazasini yaratish, ma'lumotlar bazasining SQL so'rov tillari, Web texnologiyalar va ma'lumotlar bazasi kabi masalalar bayon qilingan. Ushbu qo'llanma kasb-hunar kollejlari o'quvchilari uchun Ma'lumotlar ombori fanidan o'quv qo'llanma sifatida tavsiya etiladi.

Taqrizchilar: TATU, «Axborot texnologiyalari» kafedrasi mudiri,

t.f.d., prof. H.N. Zaynidinov,

TKTI, «Informatika, avtomatlashtirish va boshqaruv» kafedrasi dotsenti, t.f.n. U.T. Akramxodjayev.

ISBN 978-9943-391-56-7

© «O'zbekiston faylasuflari milliy jamiyati» nashriyoti, 2013.

© «O'zbekiston faylasuflari milliy jamiyati» nashriyoti, 2014.

© «O'zbekiston faylasuflari milliy jamiyati» nashriyoti, 2016.

© «O'zbekiston faylasuflari milliy jamiyati» nashriyoti, 2017.

KIRISH

Moddiy ishlab chiqarishda yoki aholiga xizmat ko'rsatish sohasidagi (xizmat ko'rsatayotgan) ixtiyoriy firmaning iqtisodiy faoliyatining asosiy maqsadi foyda olishga qaratilgan. Bunday maqsadga erishish ko'p qirrali masala bo'lib, u firmadagi barcha xodimlarning ijod qilish qobiliyatiga bog'liq. Qo'yilgan masalani yechish uchun ishlab chiqarish jarayonining har bir qatnashuvchisi rahbardan tortib to oddiy xodimgacha o'zini ishlab chiqarish sohasidagi o'rnida optimal (to'g'ri, eng yaxshi) qarorlar qabul qila olishi muhim o'rinni tutadi. Ixtiyoriy mutaxassisning asosiy masalasi (vazifasi) dastlabki holatdan (vaziyatdan) boshlab imkoniyati boricha optimal qarorlar qabul qilishdan iborat. Bu masalani (vazifani) samarali bajarish uchun ma'lumotlar bazasi asosida yaratiladigan kompyuter axborot tizimlarini ishlatish lozim.

Ma'lumotlar bazasini yaratish nazariyasi fani yosh sohalardan hisoblanadi. Lekin shunga qaramay bugungi kunda ma'lumotlar bazasi axborotlarni qayta ishlashning avtomatlashtirilgan tizimlari yo'nalishlari, jumladan sun'iy intellekt tizimlari, ekspert tizimlari, konstruktorlik va texnologik loyihalashning avtomatlashtirilgan tizimlari va boshqalarning asosi bo'lib hisoblanadi.

Darslik 3 qismidan iborat:

- Birinchi qism (I–III boblar), ma'lumotlar bazasini tashkil qilish va loyihalashning nazariy asoslariga bag'ishlangan.
- Ikkinchi qism (IV–VII boblar), Microsoft firmasining amaliy-dasturiy vositasi ACCESS, Visual Basic for Application, SQL texnologiyalari yordamida ma'lumotlar bazasini ishlab chiqish texnologiyasiga bag'ishlangan.
- Uchinchi qism (VIII bob), ma'lumotlar bazasi va internet texnologiyalariga bag'ishlangan.

I BOB. MA'LUMOTLAR BAZASINI LOYIHALASH ASOSLARI

1.1. Ma'lumotlar bazasi asosidagi avtomatlashtirilgan axborot tizimlari

Zamonaviy ishlab chiqarish va biznes taraqqiyotini avtomatlashtirilgan axborot tizimlarini (AAT) yaratmasdan amalga oshirib bo'lmaydi. AATning asosiy vazifasi – foydalanuvchiga optimal qarorlar qabul qilish uchun zarur bo'lgan ishonchli axborotlarni tayyorlab berishdan iborat. Hozirgi vaqtida ishlab chiqarishni boshqarish va biznesning hech bir masalasini AAT ishlatmasdan bajarib bo'lmaydi. Bularga bozorning tahlili va yangi mahsulotlarning konstruksiyasi va texnologiyasini tayyorlash; mahsulotlarni tayyorlashning texnologik jarayonlarini ishlab chiqarishni boshqarish tizimlari va boshqalar kiradi.

Bugungi kunda ixtiyoriy mutaxassisning ixtiyoriy faoliyatini birorta qaror qabul qilish tizimi deb qarash mumkin. Shuning uchun u ishonchli axborotlarga suyanishi kerak. Shunday qilib, axborot tizimlarining muhim funksiyalaridan biri boshqarish jarayonlarining axborot ta'minotini yaratishdan iborat. Bunday tizimlar boshqaruvchi axborot tizimlari deyiladi va ular odatda o'z tarkibiga katta va murakkab ma'lumotlar bazalarini oladi.

AATni tor ma'noda ba'zi bir amaliy masalalarni yechishda ishlatiladigan apparat va dasturiy vositalar majmuasi deb ham tushunamiz. Masalan, tashkilotlarda ishlaydigan kadrlarni hisobga olish va kuzatish, omborxonalar yoki omborlardagi material va tovarlarni hisobga olish va nazorat qilish, buxgalteriya masalalarini yechish va boshqalar.

AATning asosida ma'lumotlar bazasi (MB) yotadi. MB deganda, ma'lumotlarning shunday o'zaro bog'langan to'plamini tushunamiz-ki, u mashina xotirasida saqlanib, maxsus ma'lumotlar bazasini boshqarish tizimi dasturiy vositasi yordamida to'ldirilishi, o'zgartirilishi, takomillashtirilishi mumkin. Aniq ma'lumotlarni (masalani) hal qilishda inson real dunyoning u yoki bu soha bilan cheklanadi. Bunday hollarda faqat ba'zi bir obyektlarni o'rganishgina qiziqish uyg'otadi. Bunday obyektlarning majmuasi predmet soha deyiladi.

Obyekt bu ixtiyoriy predmet, hodisa, tushuncha yoki jarayondir.

Barcha obyektlar atributlar bilan xarakterlanadi. Masalan, obyekt sifatida fakultet, kutubxona, kompyuter va boshqalarni qarash mumkin. Jumladan, kompyuter obyektining atributi sifatida hisoblash tezligini, operativ xotira hajmi, o'lchamlari va boshqalarni ko'rish mumkin. Atributlarda saqlanadigan xabarlar ma'lumotlarning qiymatlari deyiladi. Masalan, operativ xotira hajmi – 128 MB, EHM hisoblash tezligi sekundiga 5 mln ta amal. Atributning shunday qiymatlari mavjudki, ular yordamida obyektlarni identifikasiyalash mumkin. Bog'langan atributlarning qiymatlarini birlashtirsak, ma'lumot yozuvlarini hosil qilamiz. Tartiblangan yozuvlarning majmuasi ma'lumot fayli deyiladi. Ma'lumotlarning nomlangan eng kichik birligi ma'lumot elementidir. U ko'pincha maydon deb aytildi va bayt hamda bitlardan tashkil topadi.

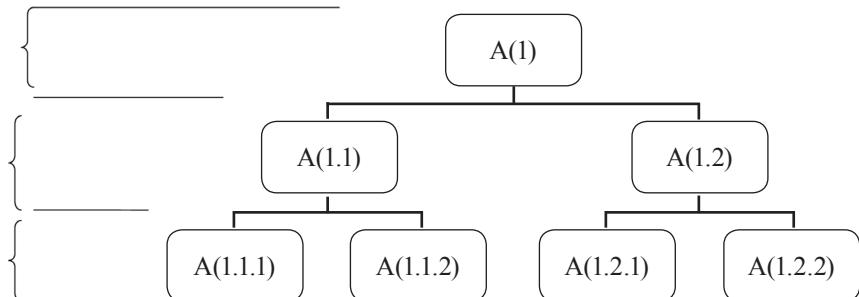
Hozirgi kunda ma'lumotlar bazasi va ma'lumotlar bazasini boshqarish tizimining bir qancha ta'riflari mavjud (hisoblash texnikasi izohli lug'ati). Jumladan: o'rganilayotgan predmet soha obyektlarining holatini va ular orasidagi munosabatlarni akslantiruvchi ma'lumotlarning majmuasi ma'lumotlar bazasi deyiladi.

Amaliy dasturlar va chekli foydalanuvchilar tomonidan ma'lumot bazasiga keladigan murojaatlarni qayta ishslash imkoniyatlarini beradigan va ma'lumot bazasining (bir butunligini) yaxlitligini ta'minlaydigan, ma'lumot baza tilida qayta ishslash vositalariga ega bo'lgan dasturiy ta'minot tizimi ma'lumot bazasining boshqarish tizimi deyiladi. Foydalanuvchilar tomonidan ma'lumotlar bazasini birgalikda ishlatish, yuritish, yaratish uchun mo'ljallangan til va dasturiy vositalar majmuasi ma'lumot bazasining boshqarish tizimi (MBBT) deyiladi.

1.2. Ma'lumotlar bazasining modellari

Odatda ma'lumotlar bazasini tashkil qilishning uchta sinfini (modelini) farqlashadi: ierarxik (shajara), tarmoqli va relatsion. Bu yerda model tushunchasi ma'lumotlarga murojaatni va saqlashni tashkil qilishni mantiq darajasida sonli va sifatli baholash imkonini beradigan struktura (tuzilma) deb qaraladi. Boshqa-cha aytganda ma'lumot modeli, bu MBni ma'lumot elementlari to'plami orasidagi bog'lanish strukturalarini tasvirlovchi umumiy sxemadir.

Ma'lumotlarning ierarxik modeli. Ma'lumotlarning ierarxik modeli ierarxik (pog'onali shajara) tuzilishga ega, ya'ni har bir element faqat bitta o'zidan yuqorida joylashgan element bilan bog'langan, lekin shu bilan birga unga o'zidan quyida joylashgan bir yoki bir nechta element murojaat qilishi mumkin. Ma'lumotlarning ierarxik modelida aniq tushunchalar: «element» (tugun), sath va bog'lanish ishlataladi. Tugunda ko'pincha birorta obyektlarning atributlari (belgilar) tavsiflanadi. Ierarxik model chizmada graf ko'rinishida tasvirlanadi, ularning har bir uchi tugun hisoblanadi (1.1.1-rasm).



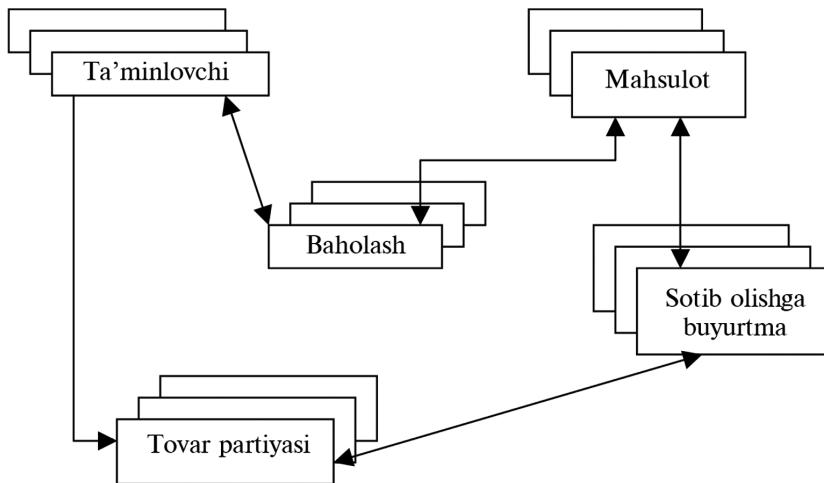
1.2.1-rasm. Ma'lumotlarning ierarxik modeli.

Bu modelning umumiyyidan xususiyga tobe tartibda joylashtirilgan elementlar majmuasidan iborat va ierarxik tuzilishli graf-daraxtni hosil qiladi deb qarash mumkin. Bunday graf boshqa hech qaysi uchlarga tobe bo'lмаган va eng yuqori (birinchi) sathda joylashgan yagona uchga ega. Birinchi sathlardagi uchlarni ma'lumot bazasidagi daraxtlar sonini aniqlaydi.

Tarmoqli ma'lumot modeli. Bu modelda ierarxik modelda ishlataladigan terminlardan foydalaniladi. «Tugun», sath va «bog'lanish»li ierarxik va tarmoqli modellarning yagona farqi shundan iboratki, tarmoqli modelning har bir ma'lumot elementi (tugun) ixtiyoriy boshqa element (tugun) bilan bog'lanishi mumkin (1.1.2-rasm).

Tarmoqli modelni tarmoqli graf ko'rinishida tasvirlash mumkin. Tarmoqli graf graf-daraxtga aylantirilishi mumkin.

Relatsion ma'lumotlar modeli. Relatsion ma'lumotlar modelining asosiy g'oyasi ixtiyoriy ma'lumotlar to'plamini (majmuasini) ikki o'lcham massiv – jadvallar ko'rinishida tasvirlashdan iborat.



1.2.2-rasm. Beshta toifali yozuvdan iborat tarmoqli model.

Eng sodda holda relatsion model bitta ikki o'lchamli jadval ko'rinishida tasvirlanadi (1.2.1-jadval).

1.2.1-jadval

Shaxsiy raqam	I.F.Sh.	Tug'ilgan sana	Kurs	Mutaxassislik
00012	Sobirov B.B.	05/12/1990	2	Tarix
00105	Soliyev B.P.	12/02/1991	3	Biologiya
00125	Alimov A.V.	14/04/1996	2	Fizika
01010	Karimov M.M.	01/05/1992	3	Pedagogika

Ma'lumotlarning relatsion modeli yoki relatsion ma'lumotlar bazasi ishlab chiqarishda va biznesda axborot tizimlarini tashkil qilish va loyihalash asosiy usul hisoblanadi.

MBni ishlatish afzalliklari:

- ixchamligi;
- axborotlar qayta ishlash tezligining oshishi;
- kam mehnat sarfi;
- har doim yangi axborot olish imkoniyati;
- ma'lumotlar ortiqchaligining kamayishi.

Bundan buyon, ma'lumotlar bazasi bilan ishlaydigan dasturni ilova (*приложение*) deb ataladi. Bitta ma'lumot bazasi bilan juda ko'p ilovalar ishlashi mumkin.

1.3. Relatsion ma'lumotlar bazasi

Relatsion ma'lumotlar bazasining taraqqiyoti 1960-yillar oxiridan boshlangan. Shu davrlarda mutaxassislar uchun oddiy bo'lgan ma'lumotlarni jadval ko'rinishida rasmiy ravishda tasvirlash imkoniyatlaridan foydalanish bo'yicha birinchi ilmiy maqolalar paydo bo'lgan. Ba'zi mutaxassislar axborotlarni bunday tasvirlash usullarini jadvalli yechimlar, boshqalari jadvalli algoritmlar deb atashgan.

Relatsion ma'lumotlar nazariyotchilari axborotlarni jadvalda tasvirlash usullarini datalogik (mantiqiy ma'lumot) modellar deb atashgan. Relatsion ma'lumotlar bazalari nazariyasining asoschisi IBM firmasining xodimi doktor E.F. Kodd hisoblanadi. U o'zining 1970-yillarda yozgan maqolalarida birinchi bo'lib relatsion ma'lumotlar modeli tushunchasini ishlatgan.

E.F. Kodd ixtiyoriy ma'lumotlar to'plamini maxsus ko'rinishdagi ikki o'lchamli jadvallar yordamida tasvirlash mumkinligini isbot qildi. Bunday maxsus ko'rinishdagi jadvallar matematikada munosabat deb atalgan. Inglizcha relatsion (munosabat, aloqa) so'zidan relatsion ma'lumot modellari nomi kelib chiqqan.

Shunday qilib, relatsion ma'lumotlar bazasini aniq bog'lanishlar bo'lgan birlashtirilgan ikki o'lchamli massiv-jadvallar ko'rinishida tasvirlangan obyektlar haqida axborotlar (ma'lumotlar)dan iborat deb qarash mumkin. Ma'lumotlar bazasini bitta jadvalda ham tashkil qilish mumkin.

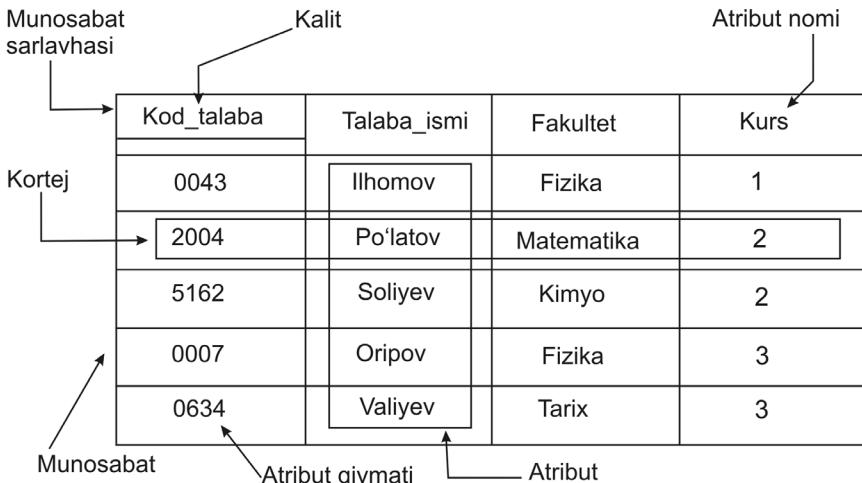
Bir necha ta'rif va tushunchalarni kiritamiz. Ma'lumotlar bazasi jadvali bir singfa tegishli obyektlar haqida axborotlarni saqlovchi ikki o'lchamli massivdir. Relatsion algebra nazariyasida ikki o'lchamli massiv (jadval) munosabat deb ataladi. Jadval quyidagi elementlardan tashkil topadi: maydon (atribut), yacheyska (katak), yozuv (kortej) (1.3.1-rasm).

Maydonda MB obyektlarini xarakterlaydigan belgilardan birining qiymati yoziladi. Jadvaldagagi maydonlar soni MB obyektlarini xarakterlaydigan belgilar soniga mos keladi. Yacheykada (katakda) obyektning bitta belgisiga mos kelgan qiymat yoziladi.

Yozuv – jadval satri. U bitta obyektga tegishli barcha belgilarning qiymatlaridan tashkil topadi. Yozuvlar soni jadvalda ma'lumotlari joylashtirilayotgan obyektlar soniga mos keladi. Ma'lumotlar bazasi nazariyasida yozuv terminiga kortej tushun-

chasi mos keladi. Kortej bu o‘zaro AND (va) munosabati bilan bog‘langan atributlar ketma-ketligidir.

Graflar nazariyasida kortej yo‘nalishli graf-daraxtning bitta tarmog‘iga mos keladi.



1.3.1-rasm. Relatsion ma’lumotlar bazasi munosabati.

Jadvalda (1.3.1-jadval) relatsion ma’lumot bazasini ishlab chiqish nazariyasi va amaliyotda ishlatiladigan terminlar keltirilgan.

1.3.1-jadval.

Fayl	Jadval	Munosabat	Mohiyat
Yozuv	Satr	Kortej	Mohiyat nusxasi
Maydon	Ustun	Atribut	Atribut

Quyida relatsion ma’lumotlar bazasining munosabatlariiga qo‘yiladigan shart va cheklanishlarni sanab o‘tamiz. Ular munosabatlarni jadval deb hisoblash imkonini beradi:

1. Bir xil birlamchi kalit bo‘lmashligi kerak, ya’ni jadvalning barcha satrlari yagona bo‘lishi kerak.

2. Jadvalning barcha satrlari bir xil tuzilishga (strukturaga) ega bo‘lishi kerak, ya’ni bir xil sondagi atributlarga nomlari mos ravishda bir-biriga to‘g‘ri kelishi kerak.

3. Jadval ustunlarining nomlari har xil, ularning qiymatlari bir jinsli bo‘lishi kerak.

4. Atribut qiymatlari atomar bo‘lishi zarur, demak munosabat tarkibida boshqa munosabat komponent bo‘lib kela olmaydi.

5. Tashqi kalitlarda murojaatning yagonaligiga rioya qilish zarur.

6. Jadvalda satrlarning kelish tartibi muhim emas, chunki u faqat satrlarga murojaat qilish tezligigagina ta'sir qiladi.

Relatsion ma'lumotlar bazasida optimal tuzilishni qurish uchun muhim tushunchalardan biri kalit yoki kalit maydon tushunchasi hisoblanadi. Kalit deb, o'zining qiymatlari bilan jadvaldagi qolgan barcha maydonlarning qiymatlarini aniqlaydigan maydonni tushunamiz. Masalan, pasport nomeri yoki soliq to'lovchining identifikatsion nomeri maydoni, ixtiyoriy fizik shaxsning xarakteristikasini bir qiymatli aniqlaydi (kadrlar bo'limi yoki korxona buxgalteriyasi uchun mos ma'lumotlar bazasi jadvallarini tuzishda).

Jadvalning kalit maydoni bitta maydondan emas, balki bir nechta maydondan tashkil qilinishi mumkin. Maydonlar to'plami jadvalning mumkin bo'lgan kaliti hisoblanadi. Bunday hollarda kalit maydon vaqt bog'liq bo'limgan ikkita shartni qanoatlantirishi kerak:

- yagonalik;
- minimallik.

Kalit maydon tarkibiga kirmagan har bir maydon jadvalning kalit bo'limgan (kalitsiz) maydoni deyiladi.

Kalitning yagonaligi, MB jadvali tarkibida ixtiyoriy vaqt momentida kalit maydonining qiymati bir xil bo'lgan ikkita har xil yozuv bo'lmasligini anglatadi.

Bu shartning bajarilishi juda muhim. Kalit maydonlarni minimallashtirish sharti, faqat tanlab olingan maydonlarning qiymatlari birgalikda ma'lumotlar bazasi jadvali yozuvlarini yagonaligi talablariga javob berishini anglatadi.

Bundan tashqari, kalitga kiruvchi maydonlardan hech birini yagonalik shartini buzmasdan olib tashlab bo'lmaydi. Ma'lumotlar bazasi jadvalida bir necha maydondan tashkil topgan kalitni yaratishda quyidagi talablar bajarilishi zarur: jadvalni kalit maydonining tarkibiga, jadvaldagi yozuvlarni o'z qiymati bilan bir xil qiymatli identifikatsiyalaydigan ma'lumotlarni kiritmaslik kerak, masalan, bir vaqtning o'zida pasport nomeri va soliq to'lovchining identifikatsion nomeridan atributlarni jadvalning kalit maydoni tarkibiga kiritmasligi kerak, chunki bu atributlarning har biri o'zi jadval yozuvlarini identifikatsiyalashi mumkin.

Kalit tarkibiga jadvalda qiymatlari takrorlanuvchi maydonlarni (yagona qiymatli bo'limgan maydonlarni) kiritmaslik kerak. Har bir jadvalda hech bo'limganda bitta kalit bo'la oладиган maydon bo'lishi kerak va u birlamchi kalit deb tanlab olinadi.

Agar jadvalda yozuvlarning bir qiymatli aniqlovchi maydonlari mavjud bo'lsa, unda bu maydonlar alternativ kalitlar sifatida qabul qilinishi mumkin. Masalan, agar birlamchi kalit sifatida soliq to'lovchining identifikatsion nomerini tanlab olsak, unda pasport nomeri alternativ kalit bo'ladi.

1.4. Relatsion ma'lumotlar bazasi jadvalini normallashtirish

Relatsion ma'lumotlar bazasini o'zaro bir-biri bilan bog'langan jadvallarning birorta to'plamidan iborat deb qarash mumkin. Bitta ma'lumot bazasi yoki bitta fayldagi jadvallar soni bir qancha omillarga bog'liq, ulardan asosiyalar quyidagilar:

- ma'lumotlar bazalaridan foydalanuvchilar tarkibi;
- axborotlar yaxlitligini ta'minlash (ayniqsa, ko'p foydalanuvchili axborot tizimlarida);
- talab etilgan xotiraning minimal hajmini va ma'lumotlarni qayta ishlashning minimal vaqtini ta'minlash.

Relatsion ma'lumotlar bazasini loyihalashda bu omillarni hisobga olish jadvallarni normallashtirish va ular o'rtasida bog'lanishlarni o'rnatish bilan amalga oshiriladi. Jadvallarni normallashtirish, ma'lumotlar bazasining bitta jadvalini yuqoridagi sanab o'tilgan talablarga to'la javob beradigan bir necha jadvalga bo'lish usulidan iborat bo'lgan jarayondir. Jadvalni normallashtirish jadval strukturasini (tizimini) oxirgi normal forma talablariга javob beradigan bo'lgunga qadar, ketma-ket o'zgartirishdan iborat.

Hammasi bo'lib normallashtirishning 6 ta normal formalari mavjud:

1. Birinchi normal forma (First Normal Form – 1HF).
2. Ikkinci normal forma (Second Normal Form – 2HF).
3. Uchinchi normal forma (Third Normal Form – 3HF).
4. Boys kodd normal forma (Brice–Codd Normal Form – BCHF).
5. To'rtinchi normal forma (Fourth Normal Form – 4HF).
6. Beshinchi normal forma (Fifth Normal Form – 5HF).

Normal formalarini tavsiflashda quyidagi tushunchalar ishlataladi:

- «Maydonlar orasida funksional bog‘lanish»;
- «Maydonlar orasida ko‘p qiymatli funksional bog‘lanish»;
- «Maydonlar orasida tranzitiv funksional bog‘lanish»;
- «Maydonlar orasida o‘zaro mustaqillik».

Agar ixtiyoriy vaqt momentida A maydonning har bir qiymatiga B maydonning barcha mumkin bo‘lgan qiymatlaridan bitta (yagona) qiymat mos kelsa, A va B maydonlar orasida funksional bog‘lanish bor deb aytildi. Masalan, soliq to‘lovchining identifikatsion nomeri bilan uning pasport nomeri orasidagi bog‘lanishini keltirish mumkin.

Agarda B maydon A murakkab maydonga funksional bog‘langan bo‘lib, uning ixtiyoriy qism to‘plamiga funksional bog‘liq bo‘lmasa, unda A murakkab maydon bilan B maydon orasida to‘la funksional bog‘lanish bor deyiladi.

Ko‘p qiymatli funksional bog‘lanish quyidagicha aniqlanadi: A maydon B maydonni ko‘p qiymatli aniqlaydi, agarda A maydonning har bir qiymatiga, B maydonning mos qiymatlariga yaxshi aniqlangan to‘plam mos kelsa. Masalan, maktabdagi o‘quvchilar o‘zlashtirish jadvalini ko‘rib chiqsak, «o‘zlashtirish» jadvali o‘z tarkibiga predmet (maydon A) va «baholar» (maydon B)ni oladi. Unda B maydon yaxshi aniqlangan to‘plamga ega. Ya’ni mumkin bo‘lgan 1, 2, 3, 4, 5 qiymatli baholar, predmet maydonining har bir qiymatiga baholar maydonining ko‘p qiymatli yaxshi aniqlangan to‘plami mavjud. A va S maydonlar orasida tranzitiv funksional bog‘lanish shu vaqtda mavjud deyiladi, agarda S maydon B maydon bilan funksional bog‘liq bo‘lsa, B maydon esa A maydonga funksional bog‘liq bo‘lib, bunda A maydon B maydonga funksional bog‘liq bo‘lmasa.

Maydon orasida o‘zaro mustaqillik quyidagicha aniqlanadi: bir nechta maydon o‘zaro mustaqil, agar ulardan birortasi ham boshqasiga funksional bog‘liq bo‘lmasa.

Funksional bog‘lanishga doir misolni ko‘rib chiqamiz. Bizga R (o‘qituvchi, fan, kafedra) munosabati (jadvali) berilgan bo‘lsin:

O‘qituvchi	Fan	Kafedra
Ilhomov	Ehtimollik nazariysi	101
Ergashev	Anorganik kimyo	103

Isayev	Organik kimyo	101
Musayev	To‘plamlar nazariyasi	103
Po‘latov	Matematik mantiq	103
Po‘latov	Matematik analiz	101

R munosabatda o‘qituvchi va kafedra atributlari orasida funksional bog‘lanish mavjud. Chunki, har bir o‘qituvchi bitta kafedraga biriktirilgan bo‘ladi. Boshqacha aytganda o‘qituvchi atributi qiymati u qaysi kafedrada ishlashini bir qiymatli aniqlash mumkin, lekin R munosabatda kafedra va o‘qituvchi orasida bog‘lanish yo‘q, chunki ayni bir kafedrada bir necha o‘qituvchilar ishlashi mumkin. Shu bilan birga o‘qituvchi va fan atributlari orasida birorta ham funksional bog‘lanish yo‘q. Chunki, ayni bir o‘qituvchi bir nechta fanni olib borishi va ayni bir fan bir nechta o‘qituvchi tomonidan olib borilishi mumkin. Shunday qilib, R (o‘qituvchi, fan, kafedra) munosabatida 2 ta funksional bog‘lanish mavjud:

- o‘qituvchi va kafedra atributlari orasida;
- fan va kafedra atributlari orasida.

Birinchi normal forma. Jadval birinchi normal formada deyiladi, agarda uning har bir maydoni bitta qiymatdan iborat (ya’ni atomlar) bo‘lsa va ixtiyoriy kalit maydon bo‘sish bo‘lmasa. Birinchi normal forma relatsion ma’lumotlar modelining asosi hisoblanadi. Relatsion ma’lumotlar modeli ta’rifida asosan, relatsion ma’lumotlar bazasini ixtiyoriy jadvali avtomatik ravishda birinchi normal formada (INF) bo‘ladi. Bunday jadvalda bir necha maydonga bo‘lish mumkin bo‘lgan maydon (belgi) bo‘lmasligi kerak. Odatda tarkibidagi axborotlar dastlab kompyuterda qayta ishlashga mo‘ljallangan jadvallar normallashmagan bo‘ladi. Masalan, universal metall qirquvchi stanoklar haqida axborotlar joylashgan jadval berilgan bo‘lsin:

T/r	Stanok modeli	Qayta ishlash eng katta diametri, mm	Shpindel aylanish chastotasi	O‘lchamlar (uzunligi x eni x balandligi)
1	1D12	12(sakkiz oltiyoqli)	112...5000(gacha) 56...63(o‘ng)	1630x740x1410

Quyidagi sabablarga ko‘ra bu jadval normallashmagan:

1. Bu jadvalda bitta maydonda bir necha qiymat yozilgan yacheykaga (kataklar) ega bo‘lgan satrlar bor: qayta ishlashning eng katta diametri – mm va shpindelning aylanish chastotasi – ayl/min.

2. Bitta maydon tashqi o'lchamlarini (uzunligi x eni x balandligi) uchta maydonga ajratish mumkin. Bunday ajratish maqsadini keyinchalik yuza va hajmini hisoblash zarurati paydo bo'lishi bilan asoslash mumkin.

Berilgan jadvalni birinchi normal formaga keltiramiz. Buning uchun quyidagilarni bajaramiz:

1) qayta ishlashni eng katta diametr va shpindelning aylanish chastotasi maydonlari yacheysida turgan qiymatlar soniga qarab bir nechta maydonlarga ajratamiz;

2) tashqi o'lchamlar maydonini (uzunligi x eni x balandligi) uchta maydonga bo'lamiz. Uzunligi, eni, balandligi;

3) bu jadvalning kalit maydoni qilib stanok modeli yoki tartib sonini olish mumkin.

Normal forma ko'rinishi quyidagi jadvalda keltirilgan:

Normal forma

T/r	Stanok modeli	Silindr qurilmashining maksimal diametri, mm	Silindrik bo'lmagan qurilmashining maksimal diametri, mm	Silindrning aylanish chastotasi (chap), ayl/min	Shpindelning aylanish chastotasi (o'ng), ayl/min	Uzunligi	Eni	Balandligi
1	1d12	12	8	112...5000	56...630	163	740	141

2-misol. Talaba jadvali berilgan.

Talaba

ISMI	SPORT	
	Turi	Razryadi
Boboyev	Shaxmat	1
Po'latov	Futbol	S.U.

Bu jadval me'yorlashmagan, chunki sport atrubuti murakkab (atomar emas).

Jadvalni me'yorashtirish uchun sport ustunini ikki ustunga ajratamiz va 1NF ko'rinishdagi jadvalga ega bo'lamiz:

Talaba

ISMI	SPORT_ turi	SPORT_ razryadi
Boboyev	Shaxmat	1
Po'latov	Futbol	S.U.

Ikkinci normal forma. Jadval ikkinchi normal formada deyiladi, agar u birinchi normal forma talablarini qanoatlantirsa va birlamchi kalitga kirmasa, uning barcha maydonlari birlamchi kalit bilan to'la funksional bog'lanish bilan bog'langan bo'lsa. Agar jadval bitta maydondan tashkil topgan oddiy birlamchi kalitga ega bo'lsa, u avtomatik ravishda ikkinchi normal formada bo'lmasligi ham mumkin. Unda bu jadvalni ikkita va undan ortiq jadvallarga shunday ajratish kerak-ki, birlamchi kalit ixtiyoriy maydon qiyamatini bir qiyamatli identifikasiyalasini. Agar jadvalda birlamchi kalitga bog'liq bo'lмаган hech bo'lмаганда bitta maydon bo'lsa, unda birlamchi kalitga qo'shimcha ustunlar qo'shish zarur. Agar bunday ustunlar topilmasa (bo'lmasa) yangi ustun qo'shish zarur.

Misol sifatida quyida keltirilgan sport seksiyasi bilan bog'liq munosabatni (jadvalni) ko'rib chiqamiz:

ISM *	SEKSIYA*	NARX
Isayev	Futbol	100
Isayev	Voleybol	120
Po'latov	Kurash	170
Soliyev	Shaxmat	200
Soliyev	Kurash	170
Fozilov	Kurash	170
Voleybol	Voleybol	120

Bu munosabatning birlamchi kaliti { ISM, SEKSIYA } maydonlari.

Demak, kalit murakkab kalit. Bu kalitning SEKSIYA maydoni (kalit qismi) bilan NARX maydoni funksional bog'langan, ya'ni SEKSIYA → NARX. Shuning uchun 1NF ko'rinishidagi bu munosabatni 2 ta 2NF ko'rinishdagi munosabatga ajratamiz:

- Seksiya_ a'zolari (ISM, SEKSIYA) va
- Seksiya_ narx (SEKSIYA, NARX).

Uchinchi normal forma. Jadval uchinchi normal formada deyiladi, agarda u ikkinchi normal forma ta'rifini qanoatlantirsa va uni birorta ham kalit bo'lмаган maydoni boshqa ixtiyoriy kalit bo'lмаган maydoniga funksional bog'liq bo'lmasa.

Boshqacha aytganda, agarda jadval ikkinchi normal formada bo'lsa va har bir kalit bo'lмаган maydon birlamchi kalitga notranzi-

tiv bog'liq bo'lsa uchinchi normal formada deyiladi. Uchinchi normal formaga qo'yilgan talabning mazmuni barcha kalit bo'lmasligi maydonlar faqat kalit maydoniga bog'liq bo'lib, o'zaro bir-biriga bog'liq bo'lmasligini anglatadi. Misol sifatida quyidagi sxemali mehmonxona munosabatini ko'rib chiqamiz:

Mehmonxona (mehmon_id, xona_turi, narx);

Bu munosabatning birlamchi kaliti mehmon_id hisoblanadi. U yagona atributdan (maydondan) tashkil topgan. Bu munosabat 2NF ko'rinishda va unda quyidagi tranzitiv bog'lanish mavjud:

$$\begin{aligned} \text{mehmon_id} &\rightarrow \text{xona_turi} \text{ va } \text{xona_turi} \text{ narx} \rightarrow \\ &\rightarrow \text{mehmon_id} \rightarrow \text{narx}; \end{aligned}$$

Bu tranzitiv bog'lanishni yo'qotish uchun berilgan Mehmonxona (mehmon_id, xona_turi, narx) munosabatini ikki munosabatga ajratamiz.

Yashash (mehmon_id, xona_turi) va to'lov_turi (xona_turi, narx);

Hosil bo'lgan munosabatlar 2NFda va unda tranzitiv bog'lanishlar yo'q.

Shunday qilib, munosabat 3NFda bo'ladi, agarda u 2NFda bo'lib, ularda tranzitiv bog'lanishlar bo'lmasa.

Amaliyotda ma'lumotlarning baza jadvallarini optimallashtirish uchinchi normal forma bilan yakunlanadi. Jadvallarni to'rtinchi va beshinchi normal forma ko'rinishlariga keltirish nazariy jihatdan qiziqish uyg'otishi mumkin. Amaliyotda bu muammoni yangi jadval yaratishga so'rovlari ishlab chiqish bilan yechiladi (hal qilinadi). Shuni qayd qilib o'tish kerak-ki, normallashtirish (me'yorlashtirish) ma'lumot baza jadvallari bilan ishlash jarayonida yangi yozuvlarni qo'shish yoki ularni olib tashlashda paydo bo'ladigan noxushlik (anomaliya)larni bartaraf qilish imkonini beradi.

1.5. Jadvallar orasidagi bog'lanishlarni loyihalash

Berilgan ma'lumot baza jadvallarni normallashtirish jarayoni axborot tizimining optimal strukturasini yaratish imkoniyatini beradi. Xotira resurslarini eng kam talab qiladigan, bu o'z navbatida axborotga murojaat qilish vaqtining eng kam bo'lishini ta'minlaydi.

Shu bilan birga berilgan bitta jadvalni bir nechta jadvalga bo'lish, axborot tizimlarini loyihalashning muhim shartlaridan birini ma'lumotlar bazasini ishlatish jarayonida axborot yaxlitligini ta'minlashni (bajarishni) taqozo qiladi.

Jadvallarning ixtiyoriy har birida qanday o‘zgartirish bajarilishi, boshqa jadvallardagi adekvat o‘zgarishlarda o‘z aksini topishi kerak. Mana shuning o‘zi ma’lumotlar bazasi yaxlitligini ta’minlash mohiyatini tashkil qiladi. Amaliyotda bu masalani amalga oshirish ma’lumotlar bazasi jadvallari orasida bog‘lanishlar o‘rnatish bilan amalga oshiriladi.

Jadvallar orasida bog‘lanishlar o‘rnatishning asosiy qoidalari quyidagilar:

1. Bog‘lanayotgan jadvallardan bosh va tobe jadvallarni tanlab olinadi.

2. Har bir jadvalda kalit maydon tanlab olinadi. Bosh jadvalning kalit maydoni birlamchi kalit deyiladi. Tobe jadvalning kalit maydoni tashqi kalit deyiladi.

3. Jadvalni bog‘layotgan maydonlarning nomlari har xil bo‘lsa ham ma’lumot toifalari bir xil bo‘lishi kerak.

4. Jadval orasida quyidagi toifa bog‘lanishlar o‘rnatiladi: «birga bir», «birga ko‘p», «ko‘pga ko‘p».

— **Birga bir bog‘lanish.** Agar bosh jadvalning aniq satriga ixtiyoriy vaqt momentida tobe jadvalning faqat bitta satri bilan bog‘lansa, birga bir bog‘lanish o‘rnatiladi.

— **Birga ko‘p bog‘lanish.** Agar bosh jadvalning aniq satriga ixtiyoriy vaqt momentida tobe jadvalning bir necha satri bilan bog‘lansa va bunda tobe jadvalning ixtiyoriy satri bosh jadvalning faqat bitta satri bog‘langan bo‘lsa, birga ko‘p bog‘lanishga ega bo‘lamiz.

— **Ko‘pga ko‘p bog‘lanish.** Agar bosh jadval aniq satriga ixtiyoriy vaqt momentida tobe jadvalning bir nechta satri bilan bog‘lansa va ayni shu vaqtida tobe jadvalning bitta satri bosh jadvalning bir nechta satri bilan bog‘langan bo‘lsa, ko‘pga ko‘p bog‘lanishga ega bo‘lamiz.

Bosh jadvalning birlamchi kalitli qiymati o‘zgarganda bog‘langan jadvallarning holatida quyidagi variantlar bo‘lishi mumkin.

Kaskadlash (cascading). Bosh jadvaldagи birlamchi kalit ma’lumotlari o‘zgarganda bog‘langan jadvalni tashqi kalitning mos ma’lumotlarida o‘zgarishi amalga oshadi. Barcha mavjud bog‘lanishlar saqlanadi.

Cheklanishlar (restrict). Birlamchi kalit qiymatini o‘zgarishiga harakat qilinganda u bilan bog‘liq tobe jadvalning bog‘langan satrlarida o‘zgarishlar rad qilinadi. Birlamchi kalit bog‘langan

jadvallarning faqat bog'lanishi o'rnatilmagan kalit qiymatlarigina o'zgarishiga ruxsat beriladi.

O'rnatish (relatsion). Birlamchi kalit ma'lumotlari o'zgartirilganda tashqi kalitga noaniq qiymat (NULL) o'rnatiladi. Bog'liq jadval satrlarining bog'liqligi haqidagi ma'lumot yo'qoladi. Agar kalit maydonning bir necha qiymati o'zgartirilsa, unda bog'liq jadvalda o'zgartirilgan kalit bilan bog'liq bo'lmanan bir nechta satr guruhlari hosil bo'ladi. Shundan so'ng qaysi satr qaysi birlamchi kalit bilan bog'langanligini aniqlash imkoniyati bo'lmaydi.

1.6. Relatsion ma'lumotlar bazasining axborot modellari

Loyihalanayotgan ma'lumotlar bazasining har xil variantlarini baholash uchun axborot modellari ishlab chiqiladi. Hozirgi kunda loyihalanayotgan ma'lumotlar bazasining optimalligini baholash uchun asosiy tamoyillar ishlab chiqilgan. Ma'lumotlar axborot modeli tizimini tavsiflashning uch bosqichini: konseptual, mantiqiy, fizik va ularga mos uch turdag'i ma'lumot turlari nazarda tutiladi.

Ma'lumotlarning konseptual modeli. Ma'lumotlar bazasining tavsifi konseptual bosqichda (konseptual model) axborot obyektlari va ular orasidagi ma'lumotlarni saqlash va tavsiflash usullarini ko'rsatmasdan (e'tibor bermasdan) tavsiflashdan iborat. Bu ta'rifda axborot obyektlari deyilganda ma'lumotlar bazasi jadvallarida xabarlari saqlanayotgan obyektlar sinfini tushunamiz. Sinf deb bir xil belgililar to'plami bilan xarakterlanadigan obyektlar majmuasiga (guruhiга) aytildi. Bitta sinfga tegishli axborot obyektlari haqidagi ma'lumotlar bir yoki bir necha jadvallarda saqlanadi. Har xil sinfga tegishli axborot obyektlari haqidagi ma'lumotlar har xil jadvallarda joylashtiriladi. Konseptual modelni ishlab chiqishdan asosiy maqsad ma'lumotlar bazasi jadvallarining optimal tarkibini topishdan iborat. Yuqorida ko'rib chiqilgan normallashtirish tamoyillari asosida ma'lumotlar bazasi jadvallari tarkibini yaratish usullari nihoyasida ma'lumotlar bazasining konseptual modelini aniqlab beradi.

Ma'lumotlar bazasini tavsiflashning mantiqiy bosqichida (mantiqiy model) ma'lumot baza jadvallari o'rtasida mantiqiy bog'lanishlar akslantiradi. Masalan: bu mantiqiy modelning ma'lumot bazasini bu strukturasi jadvallarda bo'ladigan ixtiyoriy o'zgarishlarda axborotlar butunligini ta'minlab berish imkonini beradi, degan xulosaga kelish mumkin.

Ma'lumotlarning fizik modeli. Relatsion ma'lumotlar bazasining fizik bosqichi (fizik model) axborotlarni saqlash va qayta ishlash usullarini xarakterlaydi. Ma'lumotlar bazasi va ma'lumotlar bazasini boshqarish tizimlarini ishlab chiqish nazariyasi va amaliyotida ma'lumotlarning fizik modelini qurishga ikki xil yondashish mayjud. Birinchi yondashish aniq MBBTga bog'liq bo'lmay har bir jadval ma'lumotlarining fizik xossalalarini tavsiflashga asoslanadi. Ya'ni, ma'lumot bazalari jadvallarining fizik modeli.

Fizik modellarni ishlab chiqishga ikkinchi yondashish aniq MBBTda ma'lumotlarni saqlash, tashkil qilish, arxitekturasini ishlab chiqish bilan bog'liq va ko'pincha ma'lumotlarni saqlashning fizik modeli deyiladi.

Ma'lumotlar bazasini loyihalovchi o'zi yaratayotgan axborot tizimi uchun amaliy dastur tuzilishining arxitekturasini bilmasligi mumkin, lekin har bir jadval uchun fizik modelni ishlab chiqishi kerak.

Ma'lumotlar bazasi jadvallarining fizik modeli. Ma'lumotlar bazasi jadvallarining fizik modeli jadvalning har bir maydoni xossalari tavsiflashni nazarda tutadi. Maydon xossalalarini tavsiflash uchun quyida ko'rsatilgan shaklga qarab loyiha jadvali tuzilishi zarur.

T/r	Maydon nomi	Maydon imzosi	Ma'lumot toifasi	Simvollar soni	Aniqlik	Kalit (ha)

1.6.1-rasm. Ma'lumotlar bazasi jadvalining strukturasini tavsiflash loyihasi.

Shunday qilib, ma'lumotlar bazasi jadvali loyihasining fizik modelini ishlab chiqish har bir maydonning xarakteristikalarini tavsiflashda keltiriladi. Ma'lumotlar bazasi jadvallari maydonlarning majburiy bo'lgan xarakteristikalarini keltiramiz.

Maydon nomi birorta minimal sondagi simvollardan tashkil topadi. Jadvalda har bir alohida MBBT maydon nomlarini tuzishning o'ziga xos qoidalari mayjud. Umuman olganda maydon nomini probel bilan boshlamaslik, simvollar sifatida tinish belgilari ishlatishga yo'l qo'ymaslik kerak.

Maydon imzosi – qiymati maydon yacheykasida saqlanayotgan obyekt nomi bilan identifikatsiyalanadi. Maydon imzosi jadval sarlavhasida joylashadi. Zamonaviy MBBTda maydon imzosi ni tuzishga biron-bir cheklanishlar mayjud emas.

Ma'lumot toifasi (tipi) – MBBTda ishlataladigan ma'lumot turi. Simvollar (belgilar) soni – maydon yacheykalarida saqlanishiga mo'ljallangan belgilar soni.

Aniqlik – sonli maydonlarda verguldan keyin raqamlar soni.

Kalit – ushbu maydonning kalit maydonini aniqlaydi.

Keltirilgan xossalalar tarkibi jadvallarda saqlanishi zarur bo'lgan ma'lumotlar tavsifini minimal zaruriy kattaliklaridir.

Ma'lumotlar saqlashning fizik modeli. Fizik model ma'lumotlarni kompyuter xotirasida yoki boshqa axborot saqlovchilarda joylashish usullarini, shuningdek, bu ma'lumotlarga saqlash va murojaat qilish usullarini aniqlaydi. Tarixan saqlash va murojaat qilishni birinchi tizimlari faylli strukturalar va fayllarni boshqarish tizimlari bo'lgan. MBBTning boshqarish tizimida axborotlarni saqlash uchun faylli tizimlar samarasiz hisoblanadi. Shuning uchun zamonaviy MBBTda faylli strukturalardan bevosita ma'lumotlarni tashqi tashuvchilar tashqi xotira qurilmalarida joylashtiriladi.

Savol va topshiriqlar

1. Ma'lumotlar bazasi va ma'lumotlar bazasini boshqarish tizimlariga ta'rif bering.
2. Ierarxik, tarmoqli va relatsion ma'lumotlar bazalarini tafsiflang.
3. Relatsion ma'lumotlar bazasi jadvallari elementlari: maydon, yacheyka, yozuv, kortejga ta'rif bering.
4. Kalit, kalit maydon tushunchalari nimani anglatadi?
5. Qaysi kalit maydon birlamchi kalit, qaysi biri tashqi kalit deyiladi?
6. Relatsion ma'lumotlar bazasi jadvalini normallashtirish (me'yorlashtirish) jarayoni nimadan iborat?
7. Ma'lumotlar bazasi jadvalining qaysi beshta normal formalarini bilasiz?
8. Ma'lumotlar bazasi jadvallari orasidagi quyida keltirilgan bog'lanishlarni ta'riflang: «birga bir», «birga ko'p» «ko'pga ko'p».
9. Ma'lumotlarning axborot modelida axborot tizimlarining qaysi bosqichi tavsiflari (model turlari) ko'zda tutilgan?

II BOB. MA'LUMOTLAR BAZASINI BOSHQARISH TIZIMLARINI ISHLAB CHIQISH DASTURIY MAHSULOTLAR TAHLILI

2.1. Ma'lumotlar bazasini ishlab chiqish dasturiy vositalarining rivojlanish tarixi

O'tgan asrning 60-yillarida IBM korporatsiyasi NAA (North American Aviation, hozirgi kunda Rockwell International) firi-masi bilan birgalikda ierarxik ma'lumotlar modeliga asoslangan birinchi MBBTni ishlab chiqdilar. IMS eng birinchi MBBT bo'lishiga qaramay, hozirgi kunda ko'pgina eirik meynfreymlarda ishlatilayotgan ierarxik MBBTning asosi bo'lib qolmoqda. 1960-yilning yana bir ko'rinarli yutug'i General Electric firmasi tomonidan yaratilgan IDS (Integrated Data Store) tizimining paydo bo'lishi hisoblanadi. Bu tizimning rivojlantirilishi yangi toifa-dagi ma'lumot bazasini boshqarish – tarmoqli MBBT yaratishga olib keldi. Tarmoqli MBBT ma'lumotlar orasidagi murakkab bog'lanishlarni modellashtirishga mo'ljallab yaratilgan. Bunday tizimlarni yaratish birinchi MB standartlarini ishlab chiqishga asos bo'ldi. Bunday standartlarni yaratish uchun 1965-yil CODA-SYL (Conference on Data Systems Languages) konferensiyasida ishchi guruh tuzildi.

Ierarxik va tarmoqli ma'lumotlar modeliga asoslangan MBBTlari – MBBTning birinchi avlodи hisoblanadi. Lekin bu modellarga asoslangan tizimlarning asosiy kamchiliklariga qu'yidagilar kiradi:

- oddiy so'rovlarni yaratishni bajarish uchun ham murakkab dasturlar yaratish talab etiladi;
- dasturning ma'lumotlardan mustaqilligi, minimal darajadaligi, mavjudligi;
- nazariy asoslarning yo'qligi.

1970-yil IBM firmasi xodimi E.F. Kodd yuqoridagi kamchiliklarni bartaraf qilish imkonini beradigan ma'lumotlarni relatsion modeli haqidagi maqolasini nashr qildi. Shundan so'ng shu modelga asoslangan bir qancha MBBT yaratildi.

Shu bilan birga relatsion MBBTda so'rovlар yaratish uchun strukturalashtirilgan so'rovlар tili SQL ishlab chiqildi. Keyinchalik bu til (1986-yildan boshlab) barcha relatsion MBBT uchun standart tilga aylandi.

1980-yilga kelib bir nechta relatsion ma'lumotlar bazasining boshqarish tizimlari yaratildi, masalan, DB2 yoki SQL/DS (IBM korporatsiyasi), Oracle (Oracle korporatsiyasi) va boshqalar.

Ma'lumotlar bazasi bilan ishlaydigan ilovalarning murakkablashib ketayotgani yangi uchinchi avlod MBBT yaratish ehtiyojini vujudga keltirdi. Masalan, bu tizimlar obyektga yo'naltirilgan ma'lumotlar bazasiga asoslangan, obyektga yo'naltirilgan MBBT yoki OY MBBT (Object-Oriented DBMS – OODBMS) va obyektli-relatsion MBBT yoki OP MBBT (Object-Relational DBMS – ORDBMS) kiradi.

2.2. SQL tili, MS SQL SERVER, ORACLE haqida umumiylar ma'lumotlar

Axborot tizimlarini ishlab chiqishning boshlang'ich davrlarida ma'lumotlar bilan ishlashning maxsus tillari – so'rovlar tili yaratilgan. Ular asosan ierarxik bog'langan fayllar ko'rinishidagi ma'lumotlar bilan amal bajarishga mo'ljallangan va ularga mos axborotni qidirish algoritmlariga ega bo'lgan.

Relatsion ma'lumotlar bazasining paydo bo'lishi axborotlar qidirishning tezkor algoritmlarini yaratishga asos soldi.

Jadval ko'rinishida strukturalashgan – ikki o'lchamli massivdagi axborotlarni qayta ishlash uchun XX asrning 70-yillarida IBM firmasi tomonidan maxsus so'rovlarning strukturalashgan tili – SQL yaratildi.

Hozirgi kunda SQL tili ma'lumotlarni qayta ishlash tilining xalqaro standarti hisoblanadi. SQL tili ma'lumotlar bazasining boshqarish tili (MBBT) ishlab chiqish uchun ishlataladigan dasturiy mahsulotlarning yadrosi hisoblanadi.

SQL tili matnga mo'ljallangan til va shu sababli u bilan ishlashda matn tahririyatining bo'lishi yetarli. Uning yordami ma'lumotlar baza tuzilishini va baza axborotini yangilashda keng qo'llaniladi.

MBBTdan foydalanuvchilar va yaratuvchilar orasida hozirgi kunda quyidagi dasturiy mahsulotlar eng ko'p tarqalgan:

- maxsus dasturlash tillari – Visual FoxPro, SQL, MS SQL Server;
- amaliy dasturiy tizimlar – Microsoft Access, Oracle va boshqalar.

MS SQL Server Microsoft firmasi tomonidan yaratilgan relatsion ma'lumotlar bazasining boshqarish tizimi hisoblanadi. MS SQL Server – bu dasturiy tizim asosan foydalanuvchi ilovalarini ishlab chiqish uchun emas, balki kliyent – server arxitekturasi asosida yaratilgan ko'p foydalanuvchili ma'lumotlar bazasini boshqarish uchun mo'ljallangan. Bu tizim ma'lumotlar bazasini boshqarish (ma'lumotlarni ko'paytirish, ularni parallel qayta ishlash, korxonaning lokal hisoblash tarmog'ida, internet tarmog'ida va boshqalar)da ma'lumotlarni uzatish va qabul qilib olish, turli xildagi texnik xarakteristikali apparat vositalariga ega bo'lgan mijoz kompyuterlari bilan bog'lanish imkoniyatlarini beradi. SQL Server yetarli katta hajmdagi ma'lumotlarni (terabaytgacha) qayta ishlash uchun mo'ljallangan. MS SQL SERVER yirik korxonalarning ma'lumot bazalarini boshqarish tizimi hisoblanadi. MS SQL SERVER dastur tizimi alohida xizmatlar servisi to'plamidan iborat. Ularning har biri mustaqil ishga tushadi va ma'lum doira masalalarini hal qilishga javob beradi.

MS SQL SERVERni bunday amalga oshirish MBBTning operatsion tizimi qismi sifatida ishlashni, o'zining murojaat huquqini va aynan shu vaqtida kompyuterda ishlayotgan foydalanuvchiga bog'liq bo'lmaslik imkoniyatini beradi.

MS SQL SERVERni asosiy so'rovlar tili T-SQL va ANSI SQL hisoblanadi. Hozirgi kunda MS SQL SERVERning bir necha nashrlari (versiyalari) mavjud. Bu tizimni MS SQL SERVER 2008 R2 va MS SQL SERVER 2012 nashrlari yangi va keng tarqalgan hisoblanadi.

MS SQL SERVER keyingi versiyalarida integrallashgan ishlab chiqarish muhitini yaxshilash va rivojlantirish, ma'lumotlarni tanlab olish, almashtirish va yuklash uchun yangi texnologiyalar ni qo'llash, ko'p o'lchamli ma'lumot modellarini analitik qayta ishlash vositalari va boshqalar kiritilgan.

Taqsimlangan MBBT Oracle ga 1977-yilda asos solindi. Oracle – (Oracle Database yoki Oracle RDBMS) Oracle kompaniyasi tomonidan yaratilgan obyektli-relatsion ma'lumotlar bazasini boshqarish tizimi hisoblanadi. Oracle ning asosiy xarakteristikalariga uni ishonchliligi, xavfsizligi, yuqori ishlab chiqarish samadorligi, ishlashda qulayligi kiradi. Bu tizim har xil operatsion tizim platformalari (Linux, Microsoft Windows, Solaris, IBM z/OS, Mac OS X Server va boshqalar) uchun ishlab chiqilgan.

Oracle ning asosiy funksiyalariga nazorat nuqtalarini yaratish, jurnallashtirish, arxivlashtirish va boshqalar kiradi. Bu funksiyalar ma'lumotlarning yaxlitligini, ma'lumotlar bazasini nosoz ishslash-dan himoyalash va tiklash, xatoliklarni aniqlash va boshqa masalalarni yechish imkonini beradi. Taqsimlangan MBBT kuchli himoyalanish vositalariga ega. Oracle da asosan foydalanuvchi, klaster, sinonim kabu tushunchalar ishlatiladi.

Oracle ni keng tarqalgan va ishlatilayotgan nashrlariga Oracle 11g Release 2 (2010-y), (Oracle 10g Release 2, (2005-y) kiradi.

Microsoft Access ma'lumotlar bazasini boshqarishning professional dasturi hisoblanadi. U xilma-xil axborotlarni yig'ish va tartiblash (tizimlashtirish), tanlab olingan mezonga asosan obyektlarni qidirish va tartiblash, ma'lumotlarni kiritish uchun formalar yaratish imkoniyatiga ega. Uning yordamida mavjud yozuvlar asosida juda yaxshi hisobotlar yaratish mumkin.

2.3. Microsoft Access MBBT

Microsoft Access – bu Microsoft firmasi tomonidan ishlab chiqilgan dasturlash muhitidir. U katta hajmdagi axborotlarni saqlovchi relatsion ma'lumotlar bazasini boshqarish tizimini yaratish uchun mo'ljallangan. Microsoft Access foydalanuvchiga ma'lumotlarni qayta ishslash va yaratishni avtomatlashtirish uchun hamda ma'lumotlar bilan ishslash jarayonida ularni boshqarish uchun talab qilinadigan barcha vositalarga ega. Microsoft Access tizimi ishlab chiqarishning har xil sohalarida axborot tizimlari ishlab chiqish uchun yetarli imkoniyatlarga ega. Uni ko'pincha kichik axborot tizimlarini yaratishga mo'ljallangan MBBT deyishlariga qaramasdan, unda shunday imkoniyatlar mavjudki, unda foydalanuvchilarning ko'p murakkab masalalarini, jumladan loyihalashning avtomatlashtirilgan tizimlarini ishlab chiqish yoki avtomatlashtirilgan ekspert tizimlarini ishlab chiqish mumkin. Access ishlatib yaratilgan jadvallarni 2 mln yoziylargacha saqlash mumkin.

MBBT Access quyidagi xarakteristikalarini uning ma'lumot baza asosida axborot tizimlari ishlab chiqish metodologiyalarini o'rganish uchun boshqa tizimlarga nisbatan ancha ustun qo'yadi:

- dasturlash tili imkoniyatlarini egallamagan mutaxassislarining oson o'zlashtirib olishi, axborot tizimlari loyihalash vaqtini va uni ishlab chiqish xarajatlarini kamaytiradi;

- Windows ilovalari bilan to‘g‘ri mos kelishi;
- MBga grafik va multimediya obyektlarini qo‘yib yaratish imkoniyatlari;
- Lokal (mahalliy) va global tarmoqlarda ishlash imkoniyatlarini;
- boshqa dasturiy vositalarda yaratilgan MB jadvallarini ishlatish imkoniyatlarini.

Access MBBTning muhim afzalliklaridan biri boshqa dasturiy mahsulotlardan foydalanib ishlab chiqilgan ma’lumotlar bilan ishlash imkoniyatlari, boshqa ma’lumotlar bazasiga ma’lumotni importlash yoki boshqa ma’lumot bazalaridan ma’lumotlarni eksportlash imkoniyatlarining borligi.

Access ma’lumotlariga murojaat ochiq standartni ta’minlovchi ODBS (ODBS – Open Data Base Connectivity) ixtiyoriy ma’lumotlar bazasi bilan o’zaro ta’sir qiladi.

Access MBBTda axborotlar bilan barcha amallar ma’lumotlarni o‘qish, qo‘yish, olib tashlash va boshqalar SQL tili komandalari bilan bajariladi.

Bu kitobda SQL – 2003 standart elementlari keltirilgan, ular hammasi ham mavjud MBBT tomonidan o‘zgarishsiz berilavermaydi. Shuning uchun texnik hujatlarga murojaat qilish kerak. SQL ishlab chiqilganda oddiy tabiiy tilga yaqin relatsion MB oddiy so‘rovlari tili sifatida loyihalangan.

Amaliyotda MB bilan dasturchilar tomonidan protsedura tillarida C++, Visual Basic, Java, Pascal, Delphi va boshqa tilarda yozilgan ilovalar bilan ishlaymiz. Ko‘p hollarda ilovalar maxsus vizual ishlab chiqish muhitlarida, masalan, Delphi, Microsoft Access, Visual Dbase va h.k. yaratiladi. Bunday hollarda ilova yaratuvchilariga dastur kodini yaratishga hojat yo‘q. Unisistemaning o‘zi bajaradi. Har holda dastur kodi yozish minimal bo‘ladi. Yaratilgan ilovalar qulay grafik interfeysga ega bo‘lib, ko‘pincha foydalanuvchilardan bevosita SQL tilida so‘rov yaratishni talab qilmaydi. Buni ilovalarning o‘zi bajaradi. Ilova MBga murojaat qilganda, SQL ishlatish va ishlatmaslik mumkin. SQL ma’lumotlar qo‘sish, olib tashlash, o‘zgartirishga samarador usul bo‘lsa ham, u yagona usul emas. Agar ilovada SQL tilini ishlatish mumkin bo‘lsa, undan foydalanish kerak.

Relatsion MB foydalanuvchi interfeysi ta’minlovchi ilovalar mustaqil ravishda haqiqatda mavjud bo‘lishi mumkin. Agar

qandaydir sabablarga ko‘ra bunday interfeys mavjud bo‘lmasa, bog‘lanishga yordam beradigan qandaydir ilova yordamida amalga oshirish mumkin (masalan, Borland Sql Explorer).

SQL tili deklorativ (tavsiflovchi) til hisoblanadi. Shu bilan u dastur yozadigan tildan farq qiladi. Buning ma’nosи SQL tilidagi beriladigan ifodalar nima qilish kerakligini tavsiflaydi. Ya’ni uni qanday bajarish kerakligi ko‘rsatilmaydi.

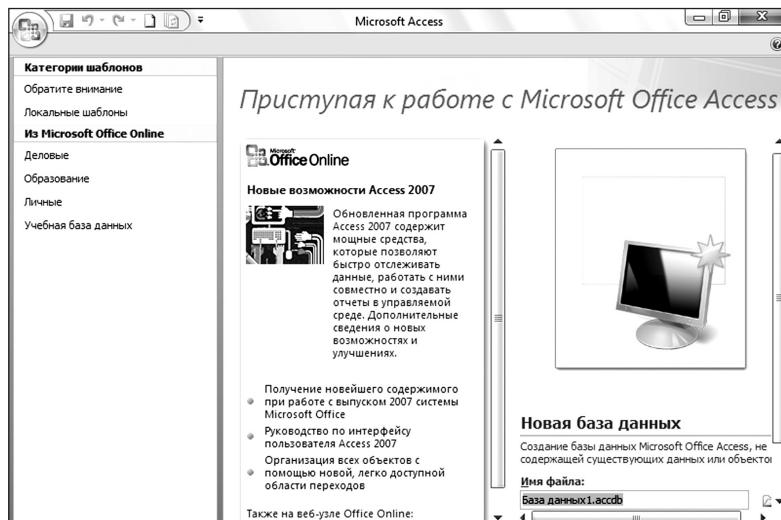
MBBTda so‘rovlар kerak bo‘lgan ma’lumotlarni bajarish kerak bo‘lgan ishlarni (funksiya, amallar) batafsil tavsiflanmaydi. Biz faqat nima olish zarurligini tavsiflaymiz. So‘rovni bajarish natijasida MBBT biz so‘ragan ma’lumotlar to‘plangan jadvalini chiqaradi.

2.4. Microsoft ACCESS vizual muhiti bilan tanishish

Microsoft Access ma’lumotlar bazasini boshqarish tizimini kaspie dasturi hisoblanadi. Uning yordamida xilma-xil axborotlarni to‘plash va tizimlashtirish, tanlab olingan mezonga asosan obyektlarni qidirish va tartiblash, ma’lumotlar kiritish uchun qulay formalar va yozuvlar asosida hisobotlar tayyorlash mumkin.

1. Yangi ma’lumotlar bazasini yaratish uchun quyidagi komandani bajarib Microsoft Access ni ishga tushiramiz:

Меню Пуск > Программы > Microsoft Office > Microsoft Office Access 2007.



2.4.1-rasm. MS Access ish oynachasi.

2. Ochilgan shablonlar darchasida «*Новая база данных*» belgisi ni bosamiz.

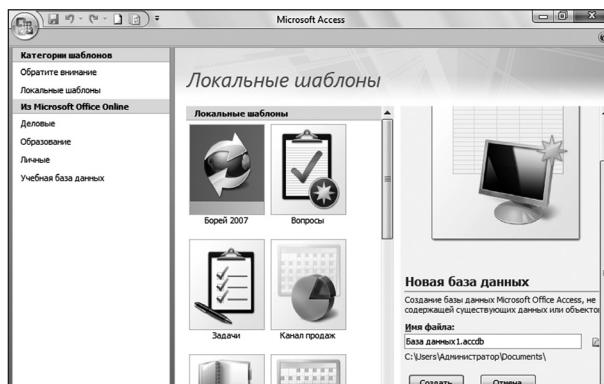
3. «*Имя файла*» maydonida yaratilayotgan bazaning nomini kiritamiz.

4. Sukut bilan ma'lumot bazasi «*Мои документы*» (My Documents) papkasida yaratiladi. Yo'lni o'zgartirish uchun belgisini bosish zarur.

5. «*Создать*» tugmasi bosiladi.

Ma'lumotlar bazasini shablonlar yordamida yaratish. Shablonlar yordamida ma'lumotlar bazasini yaratish quyidagi ketma-ketlikda olib boriladi:

- 1) MS Access yuklanadi;
- 2) ochilgan darchani «*Категории шаблонов*»da «*Локальные шаблоны*» maydonida kerakli shablonni tanlab olamiz;
- 3) «*Создать*» tugmasini bosamiz.

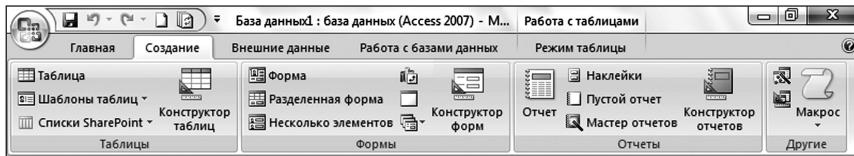


2.4.2-rasm. Shablonlar darchasi.

Yangi ma'lumot bazasi yaratilgandan so'ng «*База данных*» darchasi ochiladi:

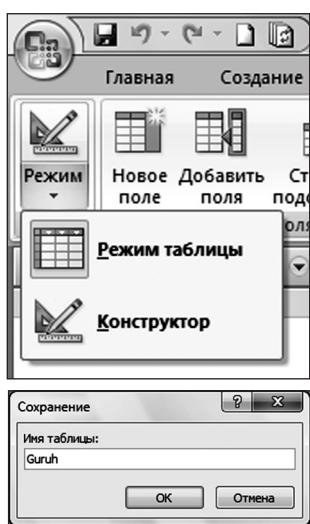
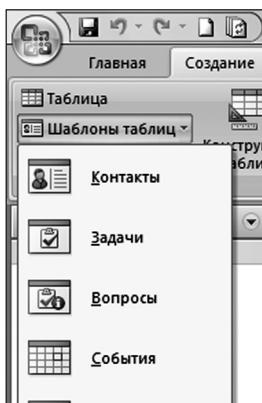
2.4.3-rasm. Jadval yarastish darchasi.

Bu darchada Access ning asosiy tasmali menyusi joylashgan. Uning ko‘rinishi quyidagi rasmda keltirilgan:



2.4.4-rasm. MS Access tasmali menyusi.

«Создание» vkladkasiga o‘tamiz. «Шаблоны таблиц»ni bosib kerakli shablon tanlanadi:

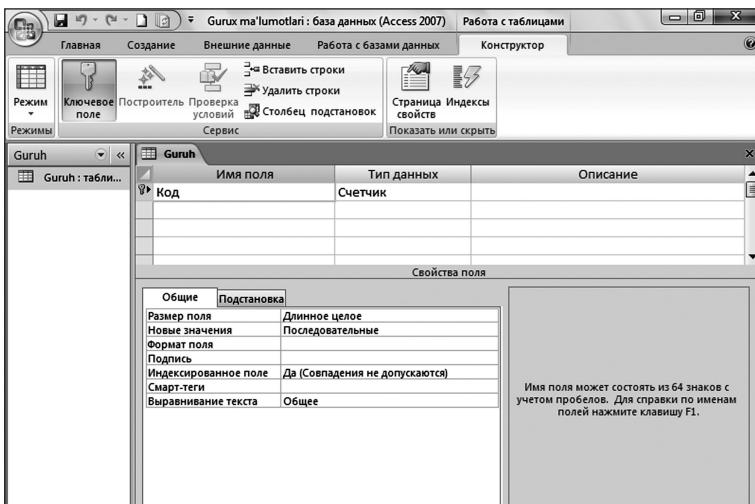


Natijada ustun nomlari tayyor bo‘lgan jadvalni olamiz va unga zarurat bo‘lsa o‘zimizning ustunlarimizni qo‘sishimiz mumkin. Buning uchun sichqonchaning o‘ng tugmasidagi «Добавить поле» ustunidan keyingi ustunda bosiladi.

Konstruktor yordamida jadval yaratish
Buning uchun:

1. «Таблица 1» vkladkasi bosiladi.
2. Konstruktor rejimiga o‘tiladi.
3. Jadvalni saqlash tavsiya etiladi.
4. Jadval nomi kiritiladi va «OK» tugmasi bosiladi.

Ochilgan «Конструктор таблиц» darchasida yangi jadval uchun maydon nomlarini kiritish mumkin. Shuningdek, mavjud jadvallarga maydon qo'shish mumkin.



2.4.5-rasm. Jadval tuzilmasi.

Savol va topshiriqlar

1. Ma'lumotlar bazasini ishlab chiqishning asosiy dasturiy vositalarini sanab o'ting va ularni tavsiflang.
2. SQL tili ma'lumotlar bazasini ishlab chiqish dasturiy tizimlarida qanday muhim rol o'yndaydi?
3. MS SQL SERVER, ORACLE MBBTlarini qisqacha tavsiflab bering.
4. Microsoft ACCESS tizimini ishlatish sohalarini aytib bering.
5. Guruh o'quvchilarining shaxsiy ma'lumotlar jadvalini tuzing.
6. O'quv fanlaridan o'quvchilarning o'zlashtirish jadvalini yaratting.
7. O'quv fanlari haqidagi ma'lumotlar jadvalini tuzing.

III BOB. MICROSOFT ACCESS VOSITALARI ASOSIDA YORDAMCHI MA'LUMOTLAR BAZASINI ISHLAB CHIQISH TEXNOLOGIYALARI

3.1. Ma'lumotlar bazasi jadvallarini ishlab chiqish texnologiyasi

Ma'lumot baza jadvallarini yaratish jarayonini quyidagi bosqichlarga bo'lish mumkin:

- ma'lumotlarning fizik modelini ishlab chiqish;
- «*Конструктор таблици*» yordamida jadvallarni yaratish;
- jadvallar o'rtaida bog'lanishlar (aloqa) o'rnatish;
- jadvallarni ma'lumotlar bilan to'ldirish.

Ma'lumotlarning fizik modelini ishlab chiqish. Microsoft ACCESS muhitida ish boshlashdan oldin, odatda MB obyektlari kerakli xarakteristikalarini tuzib olish kerak, ya'ni ma'lumotlar fizik modelini yaratib olish kerak. Buning uchun:

- obyektlarni tasvirlash belgilarining (tarkibi va maydonlari soni) nomenklaturasini o'rganish;
- jadvallar har bir maydonining xarakteristikasini o'rganish;
- natijalarni jadval ko'rinishida rasmiylashirish (3.1.1-jadval).

3.1.1-jadval.

MB maydonlarining xarakteristikalarini tasvirlash uchun jadval

MB obyektlarining belgilari tarkibi		MB maydonlarining xarakteristikalari		
Tartib raqami	Belgisi	Maydon nomi	Simvollar soni	Aniqlik
1				
2				
3				

Obyektni tasvirlash belgilar (xossalari) tarkibi va ularga mos maydon xarakteristikalari o'ylab topilgandan so'ng ACCESS muhitida jadval yaratishga o'tish mumkin. Hozirgi kunda MS ACCESSning bir nechta versiyalari mavjud bo'lib ularda amallar bajarish tartibi ko'p farq qilmaydi.

«Конструктор таблици» yordamida jadvallar yaratish. «*Конструктор таблици*» yordamida jadvallar yaratish uchun quyidagi amallarni bajarish zarur:

- kompyuterni ishga tushirish va WINDOWS hamda MS ACCESS dasturiy ta'minotlarini yuklash;
- ACCESS yuklangandan so'ng paydo bo'lgan muloqot darchasida «Файл» menyusida ikki marta sichqonchani bosish va «Создать» komandasini tanlash kerak;
- paydo bo'lgan «Создание» muloqot darchasida «База данных» tugmasi faollashtirilib, so'ngra sichqoncha bilan «Создать» tugmasini chertamiz;
- keyingi paydo bo'lgan «Файл новой базы данных» muloqot darchasida faylga nom tayinlash va ma'lumot bazasi saqlanadigan direktoriya nomi ko'rsatiladi va «Создать» tugmasi bosiladi.

Muloqot darchalar (3.1.1-rasm ... 3.1.5-rasmlarda) ko'rsatilgan.

Keyin paydo bo'lgan «База данных» muloqot darchasida «Таблица» vkladkasi faollashtiriladi va «Создать в режиме конструктора» komandasini tanlab olinadi.

Paydo bo'lgan «Конструктор таблиц» muloqot darchasida maydonlarning o'rnatilgan xarakteristikasi va tarkibiga mos qilib jadval strukturasini tuzamiz:

- maydon nomi;
- ma'lumot toifasi;
- tavsifi;
- maydon xossalari.

«Свойства поля» blokida ikkita darcha (закладки): «Общие» va «Подстановка свойства поля»dagi «Общие» albatta to'ldiriladi. «Подстановка» darchasida jadval bevosita kiritilayotgan ma'lumotlarni akslantirib turadigan qiymatlar ro'yxatini berishi mumkin. Foydalanuvchi bu holda kerakli qiymatda sichqonchani bosishi yetarlidir. Bunday maydonlar ro'yxatli maydonlar deyiladi.

Jadval maydonlariga nom berishda quyidagi tavsiyalarga e'tibor berish zarur:

- maydon nomi probel (oraliq) bilan boshlanmasligi kerak;
- maydon nomida tinish belgilari, qavs, undov belgisini ishlatalish mumkin emas;
- jadval maydon nomlarida takrorlanuvchi nomlarni ishlatalish mumkin emas;
- maydon nomlarining uzunligi 255 simvolgacha (belgiga-ch) bo'lishi mumkin. Nomlarni minimal sondagi belgilar bi-

lan berish tavsya etiladi (bu axborotlarni qidirish vaqtiga xotira hajmini minimallashtirish uchun zarur). Maydon nomlari, maydon yacheykalariga kiritilayotgan obyekt belgilarining nomlari abbreviaturasiga mazmunan yaqin bo'lishi kerak. «*Конструктор таблици*» axborot blokining satrlarini to'ldirish texnologiyasi xuddi word matn tahririyatida jadvallar bilan ishslash texnologiyasiga o'xshaydi.

Axborot blokini to'ldirishni har bir maydon uchun ketma-ket amalga oshirish kerak. Axborot blokini to'ldirishning quyidagi tartibi tavsya etiladi:

- maydon nomini kiritish;
- ma'lumot toifasini tanlash;
- «*Onucanue*» satriga maydon yacheykasiga kiritilayotgan qiymatning mazmunini tushuntiruvchi izohni kiritish kerak (keyinchalik bu izoh jadvalni to'ldirish jarayonida ekranning pastiga chiqariladi);
- maydon xossalari berish.

Sanab o'tilgan amallarni jadvalning qolgan barcha maydonlari uchun bajarish kerak. Maydon nomi yuqorida tavsiyalarga ko'ra kiritilganidan so'ng ma'lumotlar toifasi tanlab olinadi.

«*Конструктор таблици*»da ma'lumot toifalarini ro'yxatdan foydalanib tanlab olish yo'li bilan amalga oshirish qulay. Ro'yxatga quyidagi ma'lumot toifalari kiritilgan.

«*Текстовой*» – **matnli maydon toifikasi**. Bu toifadagi maydonga 255 tacha simvol (belgi) joylashtirish mumkin. Bu belgilarni ixtiyoriy belgi bo'lishi mumkin, jumladan, sonlar ham joylashtirishi mumkin. Faqat bu holda ular ustida hisoblashlar bajarib bo'lmaydi.

«*Мемо*» – bu maydon sharh maydoni deb ataladi. Bu toifadagi maydon 255 ta simvoldan uzun matnli axborot kiritish uchun mo'ljallangan (masalan, ACCESS 2000 – 65535 – simvol). Bu ma'lumot toifasi matnli maydondan shu bilan farq qiladiki, jadvalda bu maydonda ma'lumotlarning o'zi emas, balki alohida joylashgan ma'lumotlar bloklariga murojaatlar saqlanadi. Bu jadvalni qayta ishslashni yetarlicha tezlashtiradi. «*Мемо*» maydonlari kalitli yoki indeksli maydon bo'la olmaydi.

«*Числовой*» – sonli maydonda matematik hisoblashlarda qatnashadigan ma'lumotlar bazasi obyektlarini sonli qiymatlarda saqlaydi.

«Дата/время» – **sana/vaqt toifasi.** Bu toifa aniq jadvaldagi yozuvni xarakterlovchi sana va vaqtini ko'rsatish uchun ishlatiladi (masalan, mahsulot axborotga kelib tushishi sanasi yoki foydalananuvchining internet tarmog'ida ishlashni boshlashi va yakunlash sanasi ko'rsatilishi mumkin). Bu maydonda 100 dan 9999-yilgacha bo'lgan sanalarni kiritish mumkin.

«Денежный» – **pulli ma'lumot.** Bu ma'lumot toifasi sonliga o'xshash. Undan faqat kiritilayotgan sonlarning xarakteristikasi bilan farq qiladi. Sonning aniqligi verguldan keyin to'rtta raqamdan oshmaydi. Butun qismi 15 o'nli razryadlargacha bo'lishi mumkin. Sonning oxirida valuta belgilari (\$ yoki P) qo'yilishi mumkin.

«Счётчик» – bu maydonga ma'lumot baza jadvalidagi yagona (takrorlanmaydigan) yozuv raqami yoziladi. Bu maydon qiymati tiklanmaydi.

«Логический» – **mantiqiy.** Bu maydon toifasi ikkita qiymat qabul qilishi mumkin. Ular xuddi DA (ha) yoki HET (yo'q) (ДА/НЕТ, Истина/Лож, Включено/Выключено) kabi talqin qilinadi. Mantiqiy toifali maydon kalit maydon bo'la olmaydi, lekin indeks maydoni bo'lishi mumkin.

«OLE» (OLE – Obyekt). Bu toifa maydonlar yacheykasiga WINDOWS uchun ishlab chiqilgan, ilovalarga murojaatlar kiritiladi. Bular matnli, grafik yoki multimediali fayllar bo'lishi mumkin. Bu maydon yacheykalarida saqlanadigan ma'lumot hajmi faqat kompyutering disk hajmi (sohasi) bilan cheklangan.

«Гиперссылка» (Hyperlink) – **Gipermurojaat.** Bu ma'lumot toifasi maydonga gipermurojaat joylashtirish imkonini beradi. Uning yordamida MB jadvali joylashgan kompyuterdag'i yoki mahalliy tarmoqdagi ixtiyoriy kompyuterdag'i yoki internet tarmog'idagi ixtiyoriy fayl, yoki uning bo'lagiga murojaat qilish mumkin.

Gipermurojaat 3 ta qismdan tashkil topadi: faylga yo'l ko'rsatuvchi manzil; fayl ichidagi yoki matn betidagi fragment holatini ko'rsatuvchi qo'shimcha manzil; akslantiruvchi matn. Gipermurojaatning har bir qismi 2048 ta simvolgacha bo'lishi mumkin.

«Мастер подстановок» (o'rniqa qo'yish ustasi). Bu toifani tanlab olinsa, maydon yacheykalariga kiritilayotgan ma'lumotlarni qabul qilishi mumkin bo'lgan qiymatlarning fiksirlangan ro'yxatini yaratish imkoniyati bo'ladi. Ma'lumotlar nomi va toi-

fasi o'rnatilgandan keyin kursorning «*Описание*» blokining mos satriga joylashtiriladi va jadvalni to'ldirishda foydalanuvchiga to'g'ri axborot kiritish imkonini beruvchi izoh yoziladi. Umuman olganda izohni kiritish muhim, chunki, ko'p hollarda belgilangan maydonning nomi ma'lumotni to'g'ri kiritish uchun yetarli axborotga ega bo'lishi ham mumkin.

Izoh kiritilgandan so'ng «*Свойства поля*» blokidan «*Общие*» bo'limiga o'tish zarur va maydonga kerakli xossalar berilishi kerak.

«*Конструктор таблиц*»da har bir maydonga ma'lumot toifasiga bog'liq ravishda avtomatik holda (sukut bilan) aniq xossalar to'plami tayinlanadi. Jadvalni konstruksiyalash (tuzish) jarayonida bu xossalarni ma'lumotlarga qo'yiladigan aniq talablarga asosan o'zgartirish mumkin.

«*Свойства поля*», «*Общие*» axborot blokida beriladigan maydon xossalaring xarakteristikalari 3.1.2-jadvalda ko'rsatib o'tilgan. Jadvalning barcha maydonlari xarakteristikalarini tafsiflab bo'lgandan keyin «*Конструктор*» darchasi yopiladi; bunda muloqot darchalari ochiladi. Ularda jadval nomini kiritish va kaitit maydonni o'rnatish kerakligi (agar ular jadvalni konstruksiyalash vaqtida berilmagan bo'lsa) tavsiya etiladi.

Jadval nomi. Jadvalga nom berishda quyidagi tavsiyalarni hisobga olish kerak:

- jadval nomi unda joylashgan ma'lumotlarning mazmuniga mos bo'lishi kerak; jadval nomida tinish belgilari, qavs, undov belgisini ishlatalish mumkin emas;
- jadval nomi oraliq (probel) bilan boshlanishi mumkin emas;
- bitta ma'lumot faylida bir xil ismli jadvallar bo'lishi mumkin emas.

3.1.2-jadval.

Ma'lumotlar bazasi jadval maydonlarining xarakteristikalari

Maydon xossasi	Xarakteristikasi
<i>Размер поля</i>	Maydon o'lchami; ushu maydon yacheykasiga kiritilayotgan ma'lumotlarning maksimal o'lchamini o'rnatadi. Matnli (simvolli) ma'lumot maydonlari o'lchamlari 255 belidan oshib ketmasligi kerak Sonli maydonlar o'lchamlari avtomatik ravishda son toifasiga bog'liq holda o'rnatiladi:

	<p>Bayt – 0...255 butun sonlar 1 bayt Целое – 32768.... +32677 diapazondagi butun sonlar 2 bayt Длинное – 2147483648...+2147483648 diapazondagi butun sonlar. Suzuvchi nuqtali butun 6 ta raqam aniqligi bilan – sonlar diapazoni – $3.4 \cdot 10^{38}$... $+3.4 \cdot 10^{38}$ – 4 bayt Suzuvchi nuqtali butun 8 ta raqam aniqlik bilan beriladigan sonlar $-1.797 \cdot 10^{308}$... $+1.797 \cdot 10^{308}$ – 8 bayt</p>
<i>Формат поля</i>	<p>Текстовой va MEMO toifali maydonlar uchun ma'lumotlarni kiritish formatini berish mumkin, so'ngra unga mos ravishda ma'lumotlar display ekraniga chiqariladi;</p> <p>Числовой, денежный ma'lumot toifalari uchun quyidagi formatlarni tanlab olish mumkin;</p> <p>Стандартный – format, sukut bilan o'rnatiladi (mungli xonalarni ajratish, valuta belgilari bo'lmaydi, sonning o'nli raqamlari uni aniqligiga mos keladi);</p> <p>Денежный (pulli) – verguldan keyin 2 raqam o'rnatiladi va valuta belgisi chiqariladi;</p> <p>Фиксированный – eng kamida verguldan avval bitta belgi, undan keyin ikkita raqam (belgi); Mingliklar ajratish bilan verguldan keyin ikkita raqam va mingliklarni ajratuvchi;</p> <p>Процентный – son oxirida protsent belgisi chiqariladi;</p> <p>Экспоненциал – sonlar eksponensial ko'rinishda chiqariladi (masalan, $1.10 \cdot 10^3$)</p> <p>Дата/время (sana/vaqt) toifali maydonlar uchun quyidagi formatlar mayjud:</p> <p>Sananing to'la formati, sukut bilan o'rnatiladi va quyidagi ko'rinishga ega: 15.04.10.05:30:10 PM; Sananing uzun formati, masalan, Juma, 13-yanvar 2010; Sananing o'rtacha formati, 13.01.10; Sananing qisqa formati, 13.01.10; Vaqtning uzun formati, 14:33:10; Vaqtning o'rta formati, 14:33 PM; Vaqtning qisqa formati, 14:33 Mantiqiy toifa maydonlari uchun quyidagi formatlar ishlataladi: Да/Нет; Истина/Лож; Вкл/Выкл</p>
O'nli raqamlar soni (maydon aniqligi)	Числовой va Денежный toifali maydonlar uchun beriladi. Raqamlar (belgilar) soni – 0 dan 15 gacha
Kiritish niqobi	Kiritish niqobi текстовой (matnli), числовой (sonli), денежный (pulli), дата/время (sana/vaqt) maydonlarga

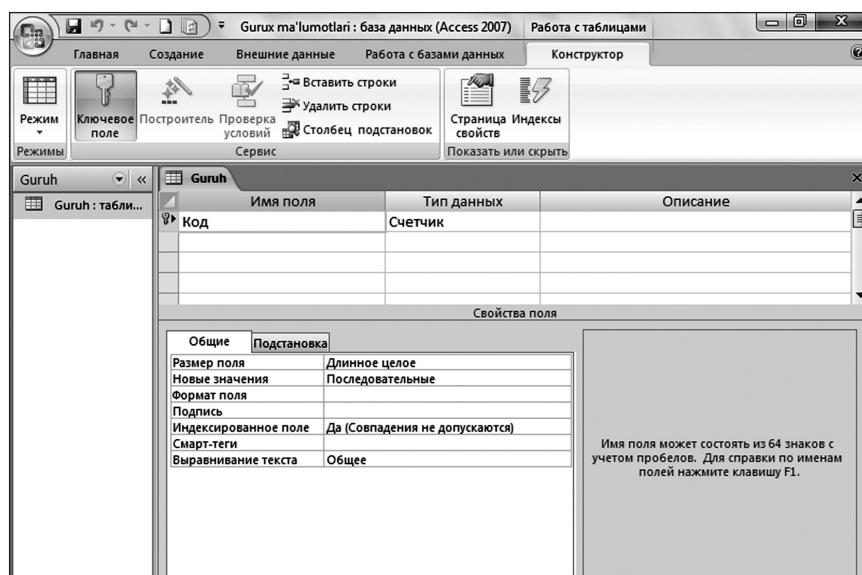
	ma'lumotlar kiritish uchun shablon (andaza) o'rnatiladi <i>Дата/время</i> (sana/vaqt) toifasidagi maydon uchun kiritish niqobi berilgan formatga mos keladi
<i>Условие на значение</i>	Kiritilayotgan ma'lumotlar qiymatlariga cheklanishlar o'rnatiladi. Masalan, sonli maydonga « <100 » shart berilsa, unda bu maydonga 100 dan katta qiymatlarni kiritib bo'lmaydi Masalan, «Moskva» or «Toshkent» or «Kiev» ko'rinishidagi shart maydonga kiritilayotgan shahar nomlari faqat Moskva, Toshkent, Kiev bo'lishi kerakligini anglatadi. Kiritilayotgan ma'lumotlarning qiymatlariga qo'yilayotgan shartlar solishtirilayotgan qiymatlar va solishtirish operatorlaridan tashkil topadi. Shart berishda ma'lum operatorlar ishlatalidi: $<$ – (kichik); $<=$ – (kichik yoki teng); $>$ – (katta); $>=$ – (katta yoki teng); $=$ – (teng); $<>$ – (teng emas); Ifodalarda mantiqiy operatorlar OR (yoki), AND (va) va solishtirish operatorlari BETWEEN, IN, LIKE qo'llanishi mumkin BETWEEN – maydonning kiritilgan qiymati, berilgan diapazon ichida joylashganini tekshiradi. Diapazonning yuqori va quyisi chegarasi AND mantiqiy operator bilan bo'linadi. Masalan, BETWEEN 20 AND 45 ifoda kiritilayotgan qiymat 20 dan 45 gacha bo'lgan intervalda joylashganini anglatadi. Bu ifodani quyidagicha yozish ham mumkin: $>50 \text{ AND } <100$; IN – kiritilayotgan qiymatini, berilgan ro'yxatdagi ixtiyoriy qiymatlardan biriga tengligini tekshiradi. Masalan, IN («Moskva» «Toshkent» «Kiev») ifoda «Moskva» or «Toshkent» or «Kiev» ifoda mos kelishini anglatadi LIKE <i>Текстовой</i> va <i>Memo</i> maydonlari berilgan simvollar shabloniga mosligi tekshiriladi. Masalan, LIKE «TEX*» ifoda kiritilgan simvollar satri «TEX» simvollar bilan boshlanishi zarurligini anglatadi
<i>Сообщение об ошибке</i>	Kiritilayotgan ma'lumot qiymatlarini berilgan shartga mos kelmaganda ekranga chiqaradigan matn
<i>Обязательное поле</i>	<i>Текстовый</i> va <i>MEMO</i> maydonlar uchun bo'sh satrlar kiritishga ruxsatni anglatadi
<i>Индексированное поле</i>	Indekslanuvchi maydon. Bu qiymatni jadvaldagi ma'lumotlarni qidirishni amalga oshirishga mo'ljallangan maydonlarga o'rnatish tavsiya etiladi, chunki indeksli maydon ma'lumotlarni qidirishni tezlashtiradi

Kalit maydon. MB jadvalidagi ma'lumotlar boshqa jadvaldagi ma'lumotlar bilan bog'lanish zarur bo'lgan hollarda o'rnatiladi. Kalit maydon jadvaldagi har bir yozuvni bir qiymatlari aniqlashi lozim. Demak, kalit maydon qiymatlari takrorlanmaydi. Agar jadvaldagi ixtiyoriy maydon qiymati barcha yozuvlarning bir qiymatini aniqlay olsa, u kalit maydon bo'lishi mumkin. Shu ma'noda jadvalning ixtiyoriy maydoni kalit maydon bo'la oladi.

Agar yozuvni berilgan bitta maydon qiymati bilan aniqlab bo'lmasa, unda bir nechta maydondan tashkil topgan kalit o'rnatish mumkin. Kalit maydon sifatida «Счётчик» toifali maydonni tanlab olish mumkin. U jadvaldagi har bir yozuvni bir qiymatli aniqlaydi.

Kalit maydon «Конструктор таблиц»да maydon xossalariini tavsiflashda yaratiladi. Buning uchun kerakli maydonni ajratish kerak va uskunalar panelida kalit ko'rinishli tugmani bosish yetarli yoki ajratilgan maydonda sichqonchaning o'ng tugmasini bosib, kontekstli menyuda «Ключевое поле» komandasini tanlanaadi.

Quyidagi rasmlarda «Конструктор таблиц» rejimida jadval strukturasining fragmentlari (qismi) keltirilgan.



3.1.1-rasm. Jadval strukturasini yaratish fragmenti.

Имя поля	Тип данных
O'QUV GURUHI	Числовой
OQITUVCHI	Текстовый

3.1.2-rasm. Jadval strukturasini yaratish fragmenti.

Имя поля	Тип данных	Добавить поле
101	Mahkamov A.	
102	Ergashev D.A.	
103	Muxitdinov H.	
104	Solieva F.T.	
105	Akbarov B.I.	
106	Xasanov J.A.	

3.1.3-rasm. Jadval strukturasini yaratish fragmenti.

Имя поля	Тип данных	Описание
TALABA_ID	Счетчик	RALIT MAYDONI
FAMILIYA	Текстовый	TALABA FAMILIYASI
ISM	Текстовый	TALABA ISMI
OTASINING ISMI	Текстовый	TALABANING SHARIFI
TIJ'UGAN YILI	Текстовый	TALABANING TIJ'UGAN YILI

Свойства поля

Общие

Размер поля: Длинное целое
Новые значения: Последовательные
Формат поля: Основной
Подпись:
Индексированное поле: Да (Совпадения не допускаются)
Смарт-теги:
Выравнивание текста: Общее

Подстановка

Тип данных определяет значения, которые можно сохранять в этом поле. Для справки по типам данных нажмите клавишу F1.

3.1.4-rasm. Jadval strukturasini yaratish fragmenti.

3.1.5-rasm. Jadval ko'rinishi.

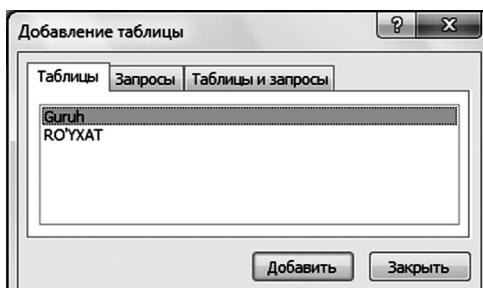
MB jadvallarning strukturalari yaratib bo'lingandan keyin, jadvallar orasida mantiqiy bog'lanishlarni o'rnatish kerak.

Jadvallar orasida bog'lanishlar o'rnatish. Jadvallar o'rtasida aloqa o'rnatish ma'lumotlar yaxlitligini ta'minlash uchun zarur. Ma'lumotlarning yaxlitligi bog'langan jadvallarda axborotni tasodifiy o'zgarishlardan himoyasini kafolatlaydi. Bog'langan jadvallarda bitta jadval bosh jadval, ikkinchisi tobe jadval hisoblanadi. Bosh jadval albatta kalit maydonga ega bo'lishi kerak.

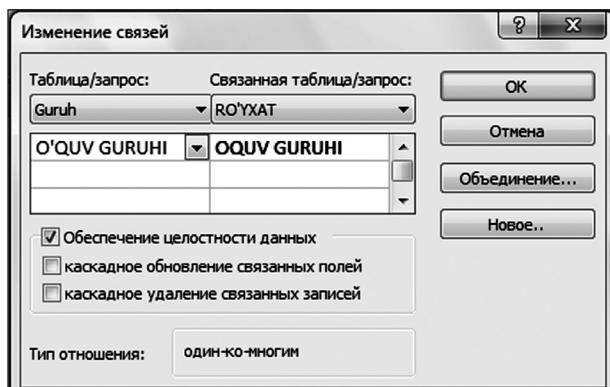
Jadvallar orasida bog'lanishlar o'rnatish uchun quyidagi amallarni bajarish kerak:

- ma'lumotlar bazasi darchasidagi uskunalar panelida «Схема данных» belgisini bosish kerak;
- ochilgan «Добавление таблицы» darchasida bosh va tobe jadvallar tanlanadi;
- bir xil maydon bo'yicha jadvallar ulanadi.

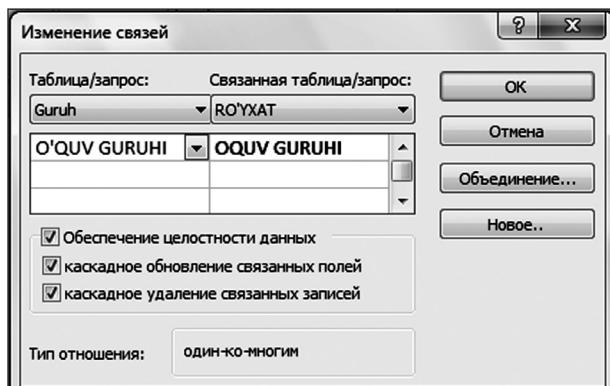
Bog'lanishlarni o'rnatish jarayonida «Обеспечение целостности данных» parametri ulanadi (ajratilgan maydonda bayroqcha o'rnatish yo'li bilan).



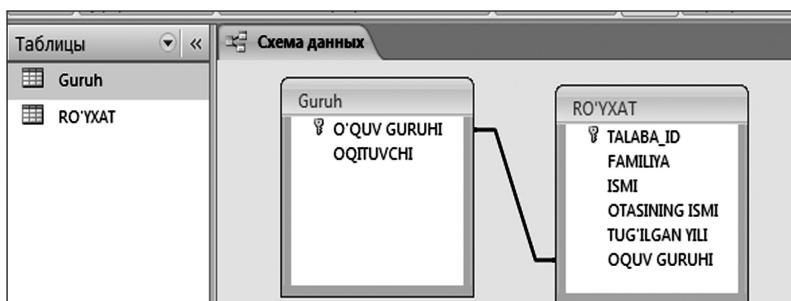
3.1.6-rasm. Jadvallarni qo'shish.



3.1.7-rasm. Jadvalni ulash darchasi.



3.1.8-rasm. Jadvalni ulash darchasi.



3.1.9-rasm. Jadvallarning bog'lanishi.

Bu parametrlar o'rnatilganda bosh jadvaldagи yozuvlar ixtiyoriy ravishda o'zgartirilishi yoki olib tashlanishiga yo'l qo'yilmaydi. Agar jadvallar orasidagi bog'lanishlar «Каскадное связанных поле»

va «*Каскадное удаление связных полей*» parametrlari o'rnatilsa (mos ravishda maydonga bayroqchalarni o'rnatish bilan), unda bosh jadvaldagi ma'lumotlarni har qanday o'zgarishi tobe jadvaldagi bog'lanishlar aniqlangan jadvaldagi ma'lumotlarning o'zgarishiga olib keladi. Bog'lanishlar o'rnatilgandan keyin jadvallarni ma'lumotlar bilan to'ldirishga o'tiladi.

Jadvallarni ma'lumotlar bilan to'ldirish. Jadvalga ma'lumotlarni kiritish texnologiyasini ikki usul bilan amalga oshirish mumkin:

- bevosita jadval yacheykalariga ma'lumotlarni kiritish yo'li bilan;
- ma'lumotlarni kiritishni forma orqali amalga oshirish yo'li bilan.

Ma'lumotlarni kiritishning birinchi usuli tanlanganda quyidagilarga amal qilish kerak:

- operatorni xatolikka yo'l qo'yish ehtimolligini kamaytirish;
- ma'lumot kirish jarayonini qulay tashkil qilish.

Agar ma'lumot baza jadvallarining maydonlari soni aytarli ko'p bo'lmasa u monitor ekraniga joylashsa, boshqa jadvallar bilan bog'liq bo'lmasa, shuningdek, yaratilayotgan tizim kommersiya maqsadlari uchun bo'lmasa, unda ma'lumotlarni kiritish uchun mos forma yaratish shart emas. Boshqa hollarda jadvalga ma'lumotlar kiritish uchun forma ishlab chiqish tavsya etiladi.

3.2. So'rovlar yaratish texnologiyasi

Ixtiyoriy axborot texnologiyasining asosiy vazifasi foydalanuvchiga zarur va ishchonchli axborotlarni tayyorlab berishdan iborat. MB jadvallarida saqlanadigan axborotlarni qayta ishlash so'rovlar yordamida amalga oshiriladi.

So'rovlar foydalanuvchi tomonidan berilgan shartlar bo'yicha jadvaldagi axborotlarni qayta ishlash va qidirish uchun mo'ljallangan birorta komandalar (buyruqlar) to'plamidir. ACCESS tizimida bajariladigan amallarga bog'liq ravishda quyidagi ko'rinishdagi so'rovlar yaratish mumkin:

- birorta amallarni bajarish (tanlov so'rovları);
- tiklash;
- qo'shish;
- olib tashlash;
- jadvallarni yaratish.

Sanab o‘tilgan so‘rovlар har biri yaratish texnologiyasi va axborotlarni tasvirlash shakli bilan farq qilishi mumkin. So‘rovlarni yaratish texnologiyasiga bog‘liq ravishda ularni doimiy va parametrik so‘rovlarga bo‘lish mumkin.

Doimiy so‘rovlarga – axborotlarni tanlab olish parametrлari uzoq vaqt davomida o‘zgarmaydigan so‘rovlар kiradi.

Parametrik so‘rovlarga – axborotlarni tanlab olish parametrлari o‘zgarib turadigan so‘rovlар kiradi.

So‘rovlarning bajarilishi natijasida dinamik jadvallar hosil bo‘ladi. Shakli bo‘yicha dinamik jadvallar ikki ko‘rinishda bo‘ladi:

- strukturalari MBda berilgan jadvallar strukturasiغا mos bo‘lgan jadvallar;

- strukturalari MBda berilgan jadvallar strukturalaridan farq qiluvchi jadvallar, ya’ni kross jadvallar.

Kross jadvalning strukturasi MBda berilgan jadvalning strukturasiдан farqi shundaki, bunday jadvallarda ustunlarning sarlavhasi (mavzusi) nomlar bilan emas, tanlangan maydon qiymatlari bilan beriladi. Bunday jadvallar alohida ko‘rinishdagi so‘rovlар – kesishuvchi so‘rovlар yordamida yaratiladi.

Kesishuvchi so‘rovlар – ayrim (alohida) maydonlarni qiymatlari bo‘yicha ma’lumotlar bir vaqtda guruhlari bilan axborotlarni tanlash uchun qo‘llaniladigan so‘rovlardir. Bunday so‘rovda ma’lumotlar kross-jadval shaklida yaratiladi.

So‘rovlarni yaratish usullari. ACCESSda foydalanuvchiga so‘rov yaratishning ikkita usuli tavsiya etiladi:

- usta yordamida konstruksiyalash – «*Конструктор*» rejimida;
- SQL rejimida dasturlash.

«*Конструктор*» rejimida so‘rovlар yaratish ustalar tizimi yordamida bajariladi. Bu holda foydalanuvchi konstruksiyalash darchasida so‘rovlар parametrлarini ko‘rsatish kerak. Bunda usta tomonidan beriladigan imkoniyatlardan foydalaniladi. ACCESS bu holda avtomatik ravishda SQL komandalaridan maxsus ketma-ketlik ko‘rinishida dastur kodini yaratadi.

SQL rejimida so‘rovni dasturlash uchun foydalanuvchi so‘rov bajaradigan barcha amallarni SQL tili komandalari yordamida tavsiflab (yozib) chiqishi kerak.

Umuman olganda, «*Конструктор*» rejimida so‘rovlarni konstruksiyalash imkoniyatlari MB jadvallaridagi axborotlarni qayta ishlashni ixtiyoriy masalasini yechish uchun yetarli.

Konstruktor rejimida so‘rovlар ishlab chiqish. ACCESS MBBT so‘rovlарини ishlab chiqish ikki usulda bajarilishi mumkin:

- yangi so‘rovni mustaqil yaratish;
- so‘rovlarni usta (master) yordamida yaratish.

Ixtiyoriy usulda MBning elementlar darchasini ochish zarur.

«Запросы окна объектов» БД tugmasi aktivlashtirilsa, MB barcha so‘rovlар joylashgan darcha ochiladi.

So‘rovlар yaratishda foydalanuvchi so‘rovlarni konstruksiyalashni quyidagi variantlarini tanlab olishi mumkin:

- konstruktor rejimida so‘rov yaratish;
- usta (*мастер*) yordamida so‘rov yaratish.

Birinchi usul tanlab olinsa «Создание запросов» darchasi ochiladi. Bunda foydalanuvchiga quyidagi so‘rovlар ishlab chiqishning rejimlari taklif etiladi:

- mustaqil ravishda so‘rovni konstruksiyalash (Konstruktor rejimida);
 - so‘rovni master yordamida konstruksiyalash rejimlari:
 - oddiy so‘rov;
 - kesiluvchi so‘rov;
 - takrorlanuvchi so‘rovlар;
 - tobesiz yozuvlar.

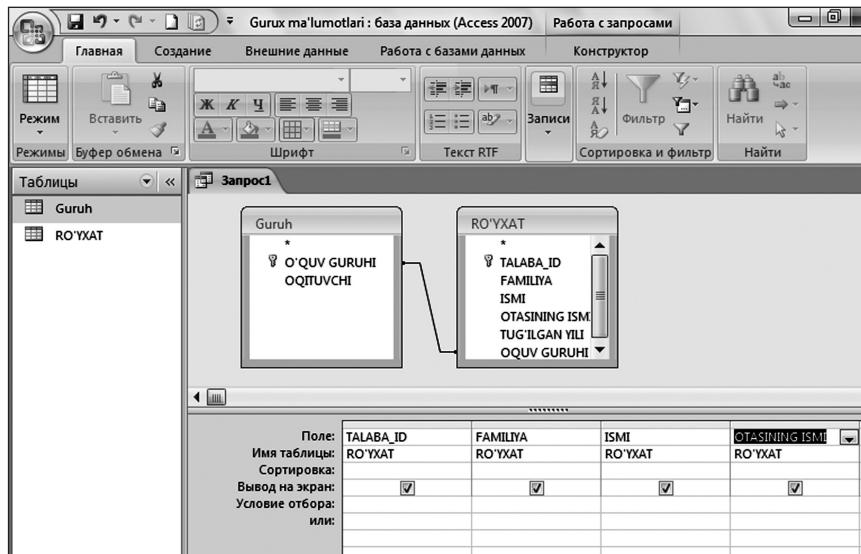
So‘rovlarni «Конструктор запросов» yordamida yaratish.

«Конструктор запросов» yordamida so‘rovlар yaratishda quyidagi amallarni bajarish zarur:

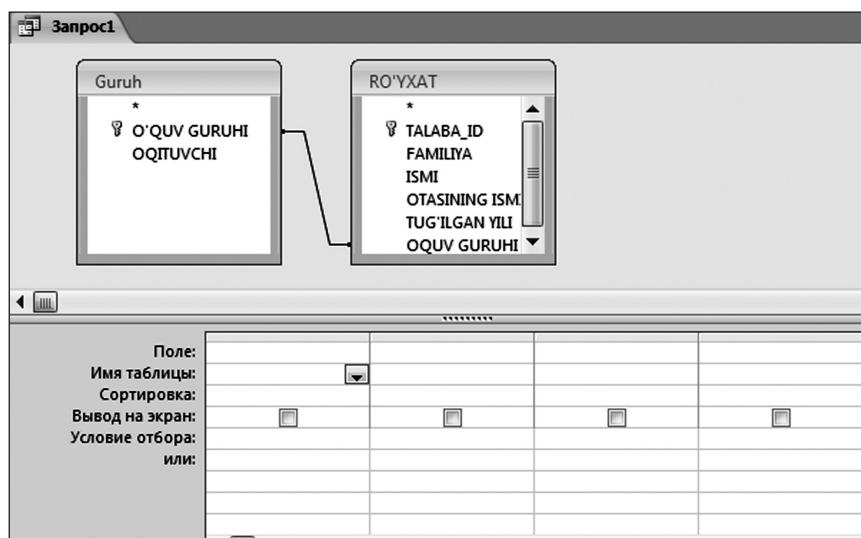
1. «Конструктор запросов»ni ochish.
2. Ochilgan «Добавление таблицы» darchasida so‘rov yaratishga asos bo‘lgan jadval yoki jadvallar tanlab olinadi (*Добавить командаси*). Tanlashni amalga oshirish uchun mos jadval ustida sichqoncha bir marta bosiladi. Bunda konstruktor darchasida jadval maydon nomlari bilan paydo bo‘ladi.
3. Jadval (jadvallar) kiritilib bo‘lgandan so‘ng «Закрыть» tugmasini bosamiz.
4. So‘rov uchun kerak bo‘lgan jadval maydonlarini «Конструктор запросов» darchasining maydonlariga kirtamiz.
5. Zarur bo‘lgan birorta maydon qiymati bo‘yicha tartiblash ketma-ketligini o‘rnatamiz (masalan, FIO maydonidagi qiymatlarni alifbo tartibida joylashtirish).

6. Maydon qiymatlarini ekranga chiqarish zaruratinini aniqlaymiz. Buning uchun pastdagi mos kvadratlarga sichqoncha bosiladi (Да(Нা)).

7. «Условия отбора» satriga jadvaldan ma'lumotlar ajratib olish maydoni bo'yicha mos qiymatlarni (ifodalarni) o'rnatamiz.



3.2.1-rasm. So'rovlar yaratish spetsifikatori.



3.2.2-rasm. «Конструктор запросов» darchasi.

Запрос1

Guruh	*	O'QUV GURUHI OQITUVCHI
RO'YXAT	*	TALABA_ID FAMILIYA ISMI OTASINING ISMI TUG'ILGAN YILI OQUV GURUHI

Поле:	TALABA_ID	FAMILIYA	
Имя таблицы:	RO'YXAT	RO'YXAT	
Сортировка:	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>		
Вывод на экран:			
Условие отбора:	или:		
или:			

3.2.3-rasm. So'rov yaratish.

Запрос1

TALABA_ID	FAMILIYA	ISMI	TUG'ILG	OQL
1 Choriev	Komil	1998	101	
2 Adilova	Nasiba	1997	101	
3 Suyunov	Xoliq	1997	102	
4 Xoliquov	Abdulla	1997	104	
5 Adilova	Nafisa	1998	105	
*	(№)			

3.2.4-rasm. So'rovni bajarish.

Запрос1

Guruh	*	O'QUV GURUHI OQITUVCHI
RO'YXAT	*	TALABA_ID FAMILIYA ISMI OTASINING ISMI TUG'ILGAN YILI OQUV GURUHI

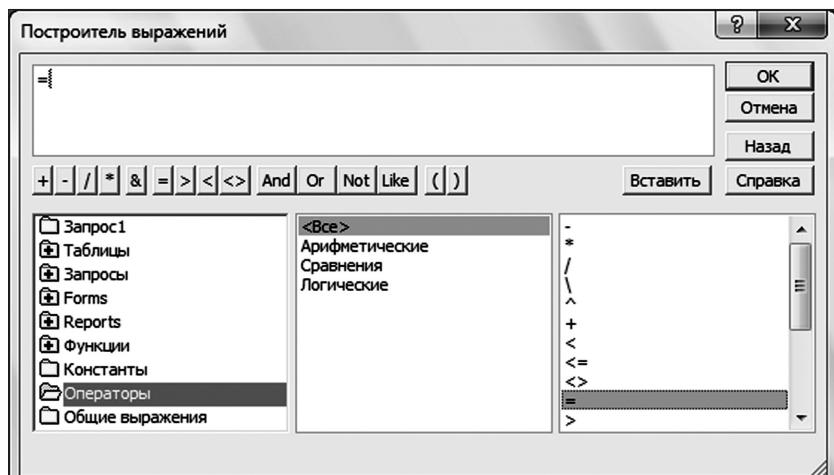
Поле:	TALABA_ID	FAMILIYA	ISMI	TUG'ILGAN YILI	OQUV GURUHI
Имя таблицы:	RO'YXAT	RO'YXAT	RO'YXAT	RO'YXAT	RO'YXAT
Сортировка:	<input checked="" type="checkbox"/>				
Вывод на экран:					
Условие отбора:	или:				
или:					

3.2.5-rasm. Shartli so'rov tuzish.

Запрос1

TALABA_ID	FAMILIYA	ISMII	TUG'ILG	OQL
1	Choriev	Komil	1998	101
2	Adilova	Nasiba	1997	101
*	(№)			

3.2.6-rasm. Shartli so‘rov natijasi.



3.2.7-rasm. Ifoda quruvchi.

Запрос1

TALABA_ID	FAMILIYA	ISMII	TUG'ILG	OQL
1	Choriev	Komil	1998	101
5	Adilova	Nafisa	1998	105
*	(№)			

3.2.8-rasm. Natija.

So‘rovni konstruksiyalash jarayonida uning bajarilishini tekshirish mumkin.

Ko‘rib chiqilgan so‘rov yaratish texnologiyasi tanlash so‘rovi deyiladi. Boshqa ko‘rinishdagi so‘rovlarni loyihalash texnologiyasi shu kabi amalga oshiriladi, faqat so‘rovni yechishdan oldin ro‘yxatdan kerakli ko‘rinishni tanlab olish kerak. So‘rovni konstruksiyalab bo‘lgandan keyin uni saqlash zarur («Конструктор» darchasida mos tugmani bosish bilan). Bunda so‘rovgan nom berishni taklif qiluvchi muloqot darchasi paydo bo‘ladi.

Ma’lumotlarni tanlash shartini tuzish qoidalari.

Ma’lumotlarni tanlash uchun shart berilganda quyidagi qoidalarga rioya qilish kerak:

1. Matnli (simvolli) maydonga qiymat kiritishdan oldin «=> belgisi qo‘yilishi kerak.

2. Kiritilayotgan simvollar satri qo‘shtirnoq ichiga olinishi kerak. Bu qoidani bajarishni unutsangiz ACCESS uni o‘zi bajaradi.

3. Tanlashning matematik shartlari tanlash munosabat amallari (=,<,>,<=,>=) bilan aniqlanadi. Bu operatorlardan tashqari maxsus solishtirish amallari (operatorlari) BETWEEN, IN, LIKE ishlataladi.

4. Sana/vaqt (Дата/время) maydonlari bo‘yicha tanlash shartlarini tuzganda, so‘rovlarni tuzishda hisoblanuvchi maydonlar kabi kiritiladigan shartlar (funksiyalar) qo‘llanilishi mumkin (3.2.1-jadval).

Sana/vaqt toifali maydonlarda tanlash shartlari yoki xossalari berish uchun ishlataladigan funksiyalar.

3.2.1-jadval.

Funksiya	Qiymat
Day	1...31 diapazonida oy sanalari tanlash shartini o‘rnatish
Month	1...12 diapazonda oylar bo‘yicha tanlash shartlari o‘rnatiladi
Year	100...9999 gacha diapazonda yillar bo‘yicha tanlash shartlarini o‘rnatish
Week day	1 dan (yakshanba) 7 (shanbagacha) hafta kunlari bo‘yicha tanlash shartlarini o‘rnatadi
Hour	Vaqt diapazoni (hafta tartibi, kvartal tartibi) bo‘yicha tanlash shartlari o‘rnatiladi va quyidagicha yoziladi: Datepart <<q>> – kvartallar bo‘yicha tanlash uchun Datepart <<ww>> – haftalar bo‘yicha tanlash uchun (q 1 dan 4 gacha qiymat qabul qiladi ww 1 dan 53 gacha qiymat qabul qiladi)

Date()	Joriy sanani tanlash sanasi sifatida o'rnatadi, masalan, <<Date()-15>> sharti, joriy sanadan 15 kunga kam (oldin) bo'lgan sanani barcha yozuvlar tanlanish kerakligini bildiradi
--------	---

Agar bir vaqtida bir nechta parametrlar bo'yicha bir necha maydon qiymatlariga javob beruvchi mantiqiy shart (AND) (va) mos keladigan yozuvlarni tanlash zarur bo'lsa, unga mos maydon qiymatlari «*Условия отбора*» satriga kiritiladi.

Agarda (OR) munosabati bilan bog'langan bir nechta qiymatlar bo'yicha yozuvlar tanlashni amalga oshirish talab etilsa, unga mos qiymatlari «*ИЛИ*» satrli yacheykalariga kiritiladi.

Odatda ma'lumotlarni tanlash shartini «*Конструктор запросов*» satriga kiritishni doimiy so'rovlarni konstruksiyalashda taklif etiladi.

Parametrik so'rovlarni konstruksiyalashda ma'lumotlarni tanlash shartlarini kiritishni forma orqali amalga oshirish tavsiya qilinadi.

Kesishuvchi so'rovlarni konstruksiyalash. Kesishuvchi so'rovlarni konstruksiyalash kesishuvchi so'rovlarni natijasi hisoblanuvchi kross-jadvallarni so'rov yoki jadvallarning sonli maydonlarini tahlil qilish uchun yaratish maqsadga muvofiqdir.

Kesishuvchi so'rovlarni yaratish ketma-ketligi yuqorida ko'r-satilgan so'rovlarni konstruksiyalash texnologiyasidan katta farq qilmaydi.

Bunday so'rovni loyihalashni boshlash uchun uni yaratish rejimi, masalan, konstruktor yoki kesishuvchi so'rovni tanlab olish zarur .

«*Конструктор*» rejimi tanlanganda, «*Запрос*» menyusida «*Перекрёстный*» tugmasida sichqonchani bosish kerak.

Kesishuvchi so'rovda uchta maydon bo'lishi kerak. Bitta maydon satrlar sarlavhasidan iborat bo'lsa, ikkinchi maydonda berilgan jadval maydoni mos qiymatlaridan tashkil topadi.

3.3. So'rovlardan hisoblashlarni avtomatlashtirish

ACCESS MBBTning muhim xossalardan biri xilma-xil turdag'i hisoblashlarni avtomatlashtirish imkoniyatining mavjudligidir. Masalan, so'mdagi tovar narxini, chet el valutasiga pulli ekvivalentga hisoblab o'tkazish jarayoni. Bunday hisoblashlarni korxonaning savdo va moliya faoliyatida bajarish zaruriyati tug'iladi.

So‘rovlari ishlatab hisoblash jarayonlari so‘rovlarini konstruksiyalashda maxsus hisoblanuvchi maydon yaratishga asoslangan.

Bu usul berilgan jadvalda hisoblanuvchi maydon nazarda tutilmagan holda ishlataladi. Bunday maydonni «*Конструктор*» rejimida so‘rovni konstruksiyalashda yaratish mumkin. Buning uchun quyidagi amallarni bajarish kerak:

- «*Конструктор*» rejimida so‘rov yaratish;
- «*Поле*» satrining oxirgi yacheykasiga kursorni o‘rnatish va sichqonchaning o‘ng tugmasini bosib kontekstli menyuni aktivlashtirish.
- Paydo bo‘lgan darchada «*Построить*» komandasini aktivlashtirish. Bu komandani bajarish natijasida «*Построитель выражений*» darchasi ochiladi. Uning ko‘rsatmalarini bajarib, hisoblash uchun ifoda quramiz.

Shuni ta’kidlab o‘tish kerakki, ixtiyoriy so‘rovning bajarilishi SQL dastur yordamida amalga oshiriladi. Bunday dastur so‘rovni konstruksiyalash jarayonida avtomatik ravishda yaratiladi. Bu dastur nimalardan iborat ekanligini tushunish uchun so‘rovni SQL rejimida ko‘rish yetarlidir («*Конструктор*» rejimida emas).

Savol va topshiriqlar

1. «*Конструктор таблицы*» qanday axborot bloklaridan tashkil topadi va ular qanday ketma-ketlikda to‘ldiriladi?
2. Maydon nomi qancha simvollardan tashkil topishi mumkin?
3. Maydon nomi probel bilan boshlanishi mumkinmi?
4. Sonli ma’lumot toifalarining pulli ma’lumot toifasidan farqi nimada?
5. Ma’lumotlar bazasi jadvallarida kalit maydon takrorlanuvchi qiymatga ega bo‘lishi mumkinmi?
6. Relatsion ma’lumotlar bazasida qaysi maydonlar bosh, qaysilari tobe hisoblanadi?
7. ACCESS MBBTda yaratiladigan so‘rov ko‘rinishlarini aytilib bering.

IV BOB. MA'LUMOTLAR BILAN ISHLASHNI AVTOMATLASHTIRISH

4.1. Formalar yordamida ma'lumotlarni kiritish va tahlil qilish

Ixtiyoriy ma'lumot bazasi asosini jadvallar va so'rovlardan tashkil qiladi. Lekin, zamonaqchilik amaliy dasturlar yoki foydalanuvchi ilovalarini yaratish, foydalanuvchining do'stona interfeysini, ya'ni foydalanuvchi va kompyuter orasidagi muloqotning samarali usulini tashkil qilishni ishlab chiqishni talab qiladi.

Ma'lumotni tashkil qilish usullaridan biri formalar ishlab chiqishdir. Microsoft Access tizimi ilova ishlab chiquvchiga kuchli muloqot formalar yaratish vositalarini beradi. Quyidagi vazifalarga mo'ljallangan muloqot formalari mavjud:

- jadvalga ma'lumotlar kiritish uchun;
- so'rovlarga axborotlarni qayta ishlash shartlarini kiritish;
- foydalanuvchi interfeysini tashkil qilish.

Jadvallarga ma'lumotlar kiritish uchun formalar axborotlarni kiritishni shunday tashkil qilishiga mo'ljallanganki, u operatorni ma'lumot kiritishda yo'l qo'yadigan xatoliklarni minimumga keltirishga yordam beradi. Bundan tashqari bunday formalar jadvalga joylashgan ma'lumotlarni tahlil qilish uchun xizmat qiladi.

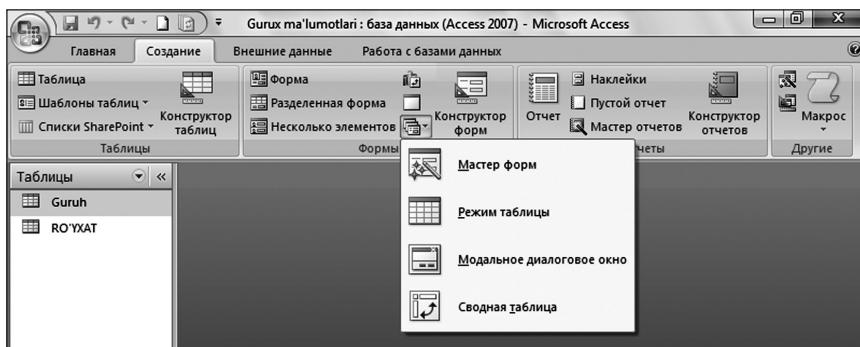
So'rovlarga axborotlarni qayta ishlash shartlarini kiritish uchun formalar, yuqoridaq kabi maqsadlarga mo'ljallangan. Bundan tashqari ular SQL tilidan foydalanmasdan so'rovlarni konstruksiyalash imkoniyatini beradi.

Foydalanuvchi interfeysini tashkil qilish uchun ishlatiladigan formalar, ishlab chiqilgan amaliy dasturni samarali rasmiylashtirish (interfeyslar yaratish) uchun mo'ljallangan. Bularga har xil vazifali formalar, zastavka formasi, menu formasi, tugmali formalar va boshqalar kiradi.

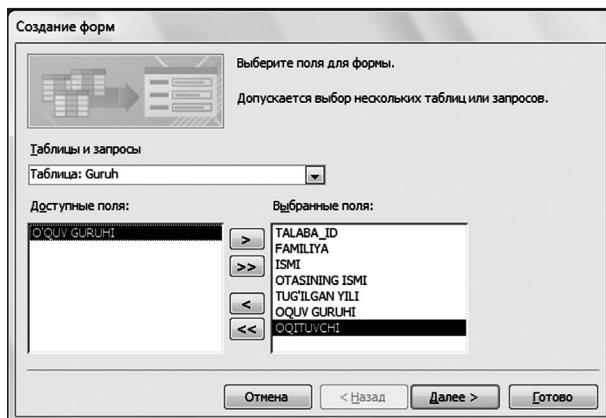
4.2. Ma'lumotlarni jadvalga kiritish uchun formani ishlab chiqish texnologiyasi

Ma'lumotlarni kiritish formasi jadvallar bilan ishslashda foydalanuvchi uchun tushunarli va qulay interfeysini ifodalaydi. Ma'lumotlarni kiritish formasi quyidagilarni ta'minlaydi:

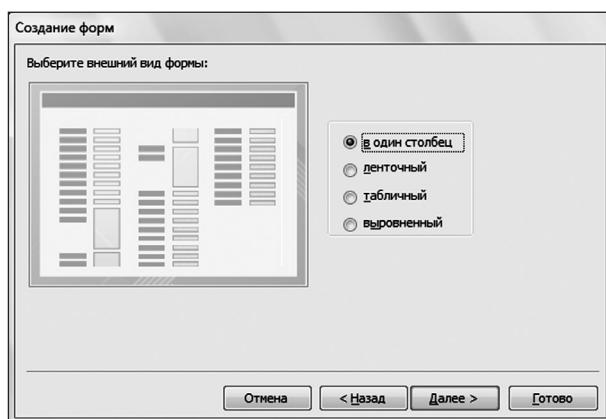
- ma'lumotlarni kiritish va qo'shish;
- ixtiyoriy yozuvni kiritish;
- ma'lumotlarni tuzatish.



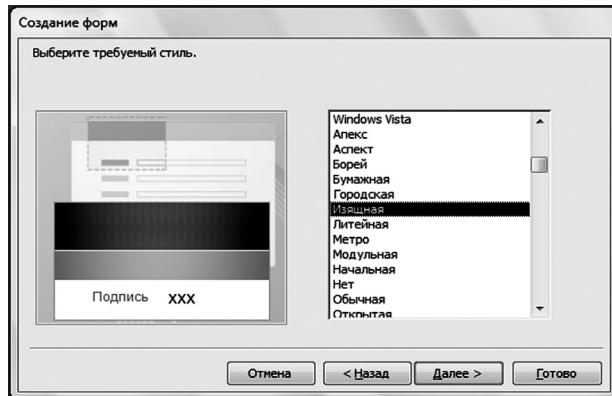
4.2.1-rasm. Forma yaratish.



4.2.2-rasm. Forma maydonlarini tanlash.



4.2.3-rasm. Forma ko'rinishini tanlash.



4.2.4-rasm. Forma uslubini tanlash.

RO'YXAT2	
TALABA_ID	1
FAMILIYA	Choriev
ISMII	Komil
OTASINING ISMI	Xasnovich
TUG'ILGAN YILI	1998
OQUV GURUHI	101

4.2.5-rasm. Forma natijasi.

Jadvalga ma'lumot kiritish uchun forma yaratish quyidagi ketma-ketlikda bajariladi.

- «База данных» darchasida «Форма» obyekti ajratiladi (formada sichqonchani bosish bilan faollashtiriladi);
- «Создать» buyrug'i (komandasasi) tanlanadi;
- paydo bo'lgan «Новая форма» muloqot darchasidagi ro'yxatdan forma yaratish uchun jadval tanlab olinadi;
- formani yaratish usuli tanlanadi.

Access tizimi ma'lumotlar bazasini ishlab chiqishda formani loyihalashning turli ko'rinish va uslublaridan foydalanish mumkin (4.2.3–4.2.5-rasmlar).

- konstruktor;
- forma ustasi;
- avtoforma: ustunda;
- avtoforma: tasmali;
- avtoforma: jadvalli;
- diagramma;
- yig‘ilgan forma.

Ma’lumotlarni kiritish formasini yaratish uchun, ko‘p hollarda formani avtomatlashtirilgan loyihalash usulini «*avtoforma: ustunda*» ishlashni tavsiya etadi. Formaning bu usuli foydalanuvchi interfeysini yaratishda eng ko‘p tarqalgan hisoblanadi.

Forma yaratishning bu usuli tanlanganda, jadvalning barcha maydonlari bitta ustunda joylashadi. Har bir maydon bitta satrda joylashadi. Maydon yozuvi jadvalda berilgan maydon nomlariga mos keladi. Bitta sahifada (ekranda) bitta yozuv ma’lumotlarini kiritish uchun maydonlar joylashadi.

Bu usul bilan olingan formani, keyinchalik «*Конструктор*» rejimida ochib, takomillashtirish mumkin («*Конструктор*» mustaqil ravishda forma yaratishda ham ishlatiladi).

«*Табличная*» formasiga jadval formasiga mos keladi. Bitta sahifada monitor ekraniga bir vaqtida bir nechta yozuv joylashsa, shuncha yozuv ko‘rinadi va har bir yozuv bitta satrni egalaydi.

4.2.1-jadval.

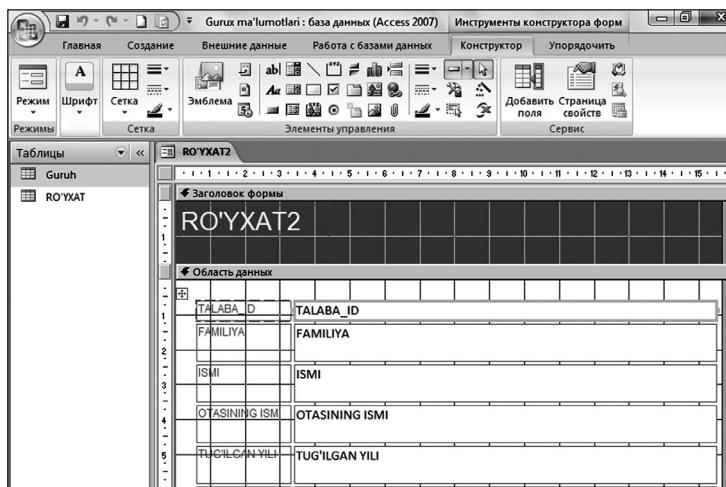
Jadval formasining ko‘rinishi

maydon 1	maydon 1	maydon 1	maydon 1

Diagramma formasini jadvaldagи yozuvlarni diagramma yoki grafik shaklida qurish uchun ishlab chiqishda tavsiya etiladi. Odatda, bunday formalar ilmiy tajriba natijalarini yoki formaning ixtiyoriy faoliyatini natijalari tahlilini qayta ishlash uchun zarur bo‘lganda yaratiladi. Bu rejimda foydalanuvchiga har xil ko‘rinishdagi grafik va diagrammalar tavsiya etiladi.

«*Сводная таблица*» formasida bir vaqtida ikkita bog‘langan jadvaldagи axborotlar keltiriladi. Bitta jadval bosh, ikkinchisi tobe hisoblanadi.

Tobe jadval bosh jadval formasiga qurilgan (kiritilgan) bo‘ladi.



4.2.6-rasm. «Конструктор форм» darchasi.

Rasmda «Конструктор форм» darchasi keltirilgan. «Конструктор форм» quyidagi bloklardan tashkil topgan:

- forma sarlavhasi;
- ma'lumotlar sohasi;
- forma izohi (*примечание*).

Bu bloklarning vazifalari ularning nomlari bilan aniqlanadi. Har bir komanda vazifasini sanab o'tamiz.

Надпись (ustama yozuv). Bu buyruq ustama yozuvlarni (matnlarni) «Конструктор форм»ning ixtiyoriy blokiga kiritish uchun mo'ljallangan. Matn kiritish uchun quyidagi amallarni bajarish kerak:

- «Надпись» tugmasida sichqonchaning chap tugmasi bilan bosiladi va uni qo'yib yubormagan holda kursorni kiritilayotgan matnni boshiga olib kelinadi;
- sichqonchaning chap tugmasini qo'yib yuboring va matnni kirititing.

Matnni kiritish texnologiyasi va uni rasmiylashtirish (bezash) Word tahririyatida matn bilan ishslash texnologiyasiga juda o'xshaydi.

ESLATMA: Ustama yozuvni bir necha satrlarda joylashtirish uchun, birinchi satr oxirida [Ctrl]+[Enter] karetkani qaytarish tugmasi bosiladi. Bu holda matn kiritilgan satrdan avtomatik ravishda keyingi satrga o'tadi, ustama yozuvning maksimal uzunligi matn satrining birinchi satri uzunligi bilan aniqlanadi.

Кнопка (tugma) – bu boshqarish elementi formalar bilan ish-lashda, boshqarish amallarini yaratish uchun mo‘ljallangan komandalar guruhini o‘z tarkibiga oladi.

Поле, поле со списком, список tugmaları – ma'lumotlar kiritishning mos maydonlarini yaratish uchun mo‘ljallangan. Ularni ishlatishga ehtiyoj so‘rovlarga ma'lumot kiritish formalarini loyi-halashda paydo bo‘ladi. Agar jadvalga ma'lumot kiritish formasi, avtomatik usullardan biri bilan ishlab chiqilgan bo‘lsa, unda maydonlar avtomatik ravishda ma'lumotlar sohasiga joylashtiriladi. «Конструктор форм» va foydalanuvchiga bu komandalarga murojaat qilish kerak bo‘lmaydi. Jadvalga ma'lumotlar kiritishning formasi mustaqil konstruksiyalashtirilsa, maydonlarni joylashtirish, jadval maydonlari ro‘yxatidan, maydonlarni sudrab o‘tish bilan amalga oshiriladi. Buning uchun «Конструктор» rejimida «ВИД» menyusi komandasini faollashtiriladi, «Список полей» komandasini tanlab olinadi va paydo bo‘lgan jadval maydonlari ro‘yxatida ketma-ket maydonlarni «Конструктор таблицы» jadvalining ma'lumotlar sohasiga ko‘chirib o‘tkazamiz. Ma'lumotlarni kiritish va joylashtirish ketma-ketligi mo‘ljallanayotgan ma'lumotlarni kiritish ketma-ketligiga mos bo‘lishi kerak.

«Группа переключателей», «выключатель», «включатель», «флажок tugmalari – mantiqiy toifali maydonlarga ma'lumotlarni kiritish uchun mo‘ljallangan.

Рисунок. Bu buyruq formaga rasmlarni joylashtirish uchun mo‘ljallangan. Rasmlarni qo‘yish texnologiyasi, Word hujjatlariga rasmlarni joylashtirish texnologiyasiga o‘xshash.

Свободная рамка объектов. OLE – obyektlar toifasidagi jadval maydonlari ma'lumotlarini akslantiradigan darcha. Jadvallar avtomatik yaratilganda, formadagi darchalar ham avtomatik yaratiladi.

Присоединенная рамка объекта (ulangan obyekt ramkasi). Boshqa faylda yoki boshqa MBda joylashgan OLE obyektni joylashtirish mumkin bo‘lgan darcha.

Разрыв строки (sohani uzish). Bu komanda agar ma'lumotlarni kiritish uchun maydonlar bitta sahifaga (ekran displayiga) joylasha olmaganda ishlatiladi.

Набор вкладок (Xatcho‘plar guruhi). Bu komanda agar ma'lumotlar kiritish uchun maydonlar bitta sahifaga (ekran displayiga) joylasha olmaganda ishlatiladi. Xatcho‘plar guruhi ishlatilganda maydonlarni birorta belgisi bo‘yicha guruhlashni amalga

oshirish va har bir guruh uchun mos xatcho‘pni yaratish tavsiya etiladi.

Formani konstruksiyalash jarayonida xatcho‘plarga maydonlarni joylashtirish, ro‘yxatdan maydonlarni ko‘chirishga asoslangan.

The screenshot shows a Microsoft Access application window. The ribbon at the top has tabs: Главная (Home), Создание (Create), Внешние данные (External Data), and Работа с базами данных (Work with Databases). Below the ribbon are various toolbars and a search bar labeled 'Найти' (Find). The main area is titled 'RO'YXAT' and contains a grid table with columns: TALABA_ID, FAMILIYA, OQUV_GURUHI, ISMI, OTASINING_ISMI, and TUG'ILGAN_YILI. The data grid shows five rows of student information:

TALABA_ID	FAMILIYA	OQUV_GURUHI	ISMI	OTASINING_ISMI	TUG'ILGAN_YILI
1	Choriev	101	Komil	Xasnovich	1998
2	Adilova	101	Nasiba	Xikmatovna	1997
3	Suyunov	102	Xoliq	Nazarovich	1997
4	Xoliqov	104	Abdulla	Ergashovich	1997
5	Adilova	105	Nafisa	Abdullaevna	1998
(№)					

4.2.7-rasm. Formani yaratish jarayoni.

Подчиненная форма (tobe forma). Bu komanda murakkab formalarni ishlab chiqishda qo‘llaniladi, murakkab formalar odatda «birga ko‘p» munosabati bilan bog‘langan jadvallar uchun yaratiladi. Bu holda bitta jadval bosh jadval, ikkinchisi tobe jadval hisoblanadi.

Shunga o‘xhash ma’lumotlar kiritib murakkab formalarni loyihalashda, formalardan birini bosh forma deb, boshqasini tobe forma deb ataymiz. Murakkab formalarni ishlab chiqish sxemasini quyidagi amallar ketma-ketligida quramiz:

- tobe jadval uchun ma’lumotlarni kiritish formasini ishlab chiqish;
- bosh jadvalga ma’lumotlarni kiritish uchun forma ishlab chiqish. Unda «Подчиненная форма» komandasini (tugmasini) ishlatib tobe formani joylashtirish sohasini hisobga olish kerak.

4.3. So‘rovga ma’lumotlar kiritish uchun forma ishlab chiqish texnologiyasi

Parametrik so‘rovlar yaratish texnologiyasi va u bilan bog‘liq ma’lumot tanlab olish shartlarini kiritish uchun formalar yaratishni ko‘rib chiqamiz.

So'rovlarga ma'lumotlar ajratish uchun maxsus formalarni ishlab chiqish zaruriyatini quyidagi faktorlar bilan asoslash mumkin:

- ma'lumotlarni ajratish shartlarini kiritish, xatoliklarini istisno yoki minimumga keltirish;

- ma'lumotlarni ajratish shartlari qiymatlarini tez-tez uzilish sharoitida so'rovlarni boshqarish zaruriyati;

- kliyent-server arxitekturasi asosida tashkil qilingan tarmoq ma'lumot bazasini ishlab chiqish.

Bunday forma va so'rovlarning ishlash texnologiyasi yuqorida ko'rib chiqilgan usullarga to'la mos keladi. O'ziga xos xususiyati ularni loyihalash ketma-ketligini tashkil qilishda namoyon bo'ladi.

Ma'lumotlarni tanlash shartlari kiritish formali so'rovlarni loyihalashni quyidagi ketma-ketlikda bajarish tavsiya etiladi:

- ma'lumotlarni tanlash shartsiz kiritish formasini ishlab chiqish;

- ma'lumotlarni tanlash shartli kiritish shaklini ishlab chiqish;

- konstrukturlar rejimida so'rovni ochish;

- mos maydon uchun «*Условие отбора*» satri yacheykasiga kursorni o'rnatiting;

- ma'lumotlarni ajratish sharti bilan formani mos maydonining qiymati bilan bog'lanishni o'rnatuvchi ifoda quriladi.

4.3.1-rasm. «Конструктор форм» darchasida forma uslubini ishlab chiqish.

Ifoda qurishning sodda usuli «Построитель выражений» ustasidan foydalanib bajarishdir. Buning uchun forma ishlab chiqilgandan keyin quyidagi amallarni bajarish kerak:

- so‘rovni «*Конструктор*» rejimida ochish;
- kursorni «*Условия отбора*» satri yacheykasiga o‘rnatish.

«*Построитель выражений*» darchasi ochilgandan keyin quyidagi amallarni bajaring:

- mos belgiga sichqonchani chertib, MB obyektini tanlab olamiz (bizning misolda – Forms) shundan so‘ng berilgan toifa obyektlarining ro‘yxati ochiladi;
- ro‘yxatdan MB obyekti elementini tanlab oling (*ноле со списком*).

Bu amallarni bajarish natijasida so‘rovdagi ma’lumotlarni ajratish sharti bilan forma maydonidagi kiritilayotgan qiyamatni bog‘lovchi ifoda yaratiladi.

Foydalanuvchi interfeysini tashkil qilish uchun formalar ishlab chiqish texnologiyasi. Ixtiyoriy ilovani ishlab chiqishning yakuniy masalasi – foydalanuvchiga qulay interfeys yaratishdan iboratdir. Foydalanuvchi interfeysi ishlab chiqishni, xuddi «фильм» ssenariyясini ishlab chiqish va uni kompyuter displayida «ekran formalarini» ketma-ket paydo bo‘lishini amalga oshirishga qiyoslash mumkin.

ACCESS tizimi o‘zidagi mavjud dasturiy vositalar, jumladan, VISUAL BASIC dasturlash tili yordamida, foydalanuvchi interfeysini minimal harakatlar bilan yaratish imkonini beradi. Bu ACCESS muhitida foydalanuvchi interfeysini yaratishda obyektga yo‘naltirilganligi bilan izohlanadi.

Bu usulning mohiyati shundan iboratki, ishlab chiquvchi obyekt xossalari tavsiflaydi, dastur esa bu xossalarga mos ravishda tasvirlangan obyektlarni yaratadi. Umuman olganda foydalanuvchi interfeysini ishlab chiqish, mavjud uskunaviy vositalar mavjudligiga qaramasdan, ko‘proq san‘at hisoblanadi. Shuning uchun kommersiya tizimlarini ishlab chiqishda dizaynerlar xizmatidan foydalanish kerak.

Ilova ssenariysi – formaning tarkibi va ketma-ketligi ishlab chiquvchi tomonidan aniq masalani yechish algoritmi bilan aniqlanadi.

Ekran formalarida ma’lumot baza obyektlarining barchasi uchun jumladan, jadvallar, so‘rovlar, hisobotlar qo‘shib hisoblaga boshqarish elementlariga ega bo‘lishi kerak.

Ma’lumot bazasi bilan ishlaganda, boshqarish elementlarini bajaradigan amallarni makroslar va dastur modullari yordamida

bajarish mumkin. Makroslar ma'lumot baza elementlari bilan ishlaydigan avtomatlashtiruvchi makrokomandalar to'plamidan iborat bo'ladi.

Modullar VISUAL BASIC tilida yozilgan dasturlardir.

Ishlab chiqilgan jadvalga ma'lumotlar kiritish formasi foydalananuvchiga quyidagi amallarni bajarish imkoniyatini beradi:

- ma'lumotlarni kiritish va to'ldirish;
- ixtiyoriy yozuvni ko'rib chiqish;
- ma'lumotlarni tuzatish.

Ma'lumotlarni kiritish texnologiyasini forma misolida qo'yib chiqamiz.

ESLATMA: Forma maydoniga axborotlarni kiritish texnologiyasi, Word tahririyatida jadvalga axborot kiritish texnologiyasiga o'xshash.

Ko'rيلayotgan forma murakkab bo'lib, ma'lumotlarni ikkita bog'langan jadvalga kiritilishga mo'ljallangan.

Foydalanuvchi interfeysi ishlab chiqish usullari va yo'llarini ko'rib chiqamiz.

Foydalanuvchi interfeysi formasini yaratish uchun jadvallar va so'rovlargacha ma'lumotlarni kiritish formalari yaratishda ishlatiladigan usul va vositalardan foydalanamiz. Shuni qayd qilib o'tish kerakki, jadval formalari ham foydalanuvchi interfeysining tarkibiy qismi hisoblanadi.

Ko'rيلayotgan holda, forma yaratishning asosiy usuli, ularni konstruktor rejimida ishlab chiqishdan iboratdir. Ma'lumot bazasi bilan ishlashni avtomatlashtirish imkonini beradigan boshqarish elementlarini yaratish texnologiyasi bilan tanishib chiqamiz.

Foydalanuvchi interfeysi formasini yaratish usullaridan eng keng tarqalgani tugmali (knopkali) formalari hisoblanadi. Bunday formalarda boshqarish obyektlaridan biri tugma hisoblanadi. Tugmani bosish birorta qandaydir amalni bajarishga olib keladi (masalan, birorta formani yoki so'rovni ochish, hisobotni bosmaga chiqarish, ilovadan chiqish va boshqalar).

Shunday qilib, foydalanuvchi interfeysi tizimini ishlashiga mos keluvchi algoritmlar asosida qandaydir ekran formalarning ketma-ket paydo bo'lishi deb qarash mumkin.

Ixtiyoriy yozuvni ko'rish va ma'lumotlarni tuzatish jadval bilan ishlash jarayonida birorta yozuvning mazmunini ko'rib chiqish vaziyatida vujudga kelishi, zarurat bo'lganda unga tuzatish-

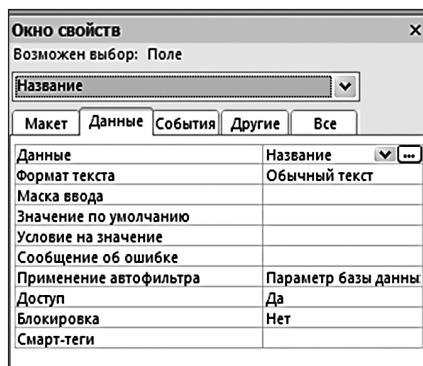
lar kiritish birorta maydon qiymatini o‘zgartirishga to‘g‘ri keladi. Buning uchun quyidagi amallar ketma-ketligini bajarish kerak:

- jadvalni ochish;
- qiymati bo‘yicha qidirilayotgan yozuvning maydon darchasiga kursorni o‘rnatamiz;

ESLATMA: Qidirilayotgan maydon nomini kiritishni bevosita «Поиск и замена» muloqot darchasida bajarish mumkin.

- «Правка» menyusida «Найти» komandasini tanlaymiz yoki uskunalar panelida mos tugmani aktivlashtiramiz;
- ochilgan «Поиск и замена» muloqot darchasida, qiziqtirayotgan yozuv maydonlaridan birining qiymatini kiritamiz;
- «Найти далее» tugmasini bosamiz.

Bajarilgan amallar natijasida ma’lumotlarni kiritish formasi darchasida qidirilayotgan yozuv maydonlarining barcha qiymatlari paydo bo‘ladi.



4.3.2-rasm. Forma xossalari darchasi.

Ma’lumotlarni tuzatish. Agar maydon qiymatlarini yangi qiymatga o‘zgartirish zaruriyati paydo bo‘lsa, «Поиск и замена» muloqot darchasida, «Замена» vkladkasi (xatcho‘p) ochiladi va maydonda ta’minot qiymatini tuzatamiz yoki o‘zgartirish bajariladi.

4.4. Makroslar yordamida ma’lumotlar bazasi obyektlarini boshqarish

Makroslar MS ACCESS buyruqlaridan iborat bo‘lgan makrokomandalar ketma-ketligidan iborat bo‘lib, yetarli ko‘p marta takrorlanuvchi amallarni bajarishni avtomatlashtirish uchun ishlatalidi.

ACCESS makrokomandalarni funksional vazifalari bo'yicha quyidagi sinflarga bo'lish mumkin:

- jadvallar, so'rovlар, formalar va hisobotlarni ochish va yopish;
- ma'lumotlarni bosmaga chiqarish;
- so'rovlarni bajarish;
- shartlarning rostligini tekshirish va makrokomandalarni boshqarish;
- qiymatlar o'rnatish;
- ma'lumotlarni qidirish;
- foydalanuvchi menyusini ko'rish va menu komandalarini bajarish;
- axborotning ekranga chiqishini boshqarish;
- boshqarilayotgan amallar haqida foydalanuvchiga xabar berish;
- obyektlarni qayta nomlash, nusxa ko'chirish, o'chirish, import va eksport qilish;
- WINDOWSning boshqa ilovalarini yuklash.

Jadvalda makroslarni yaratish uchun makrokomandalar keltirilgan.

4.1.1-jadval.

Boshqaruvchi makroslar yaratish uchun makrokomandalar ro'yxati

Makrokomanda	Vazifasi
Jadvallar, so'rovlар, formalar va hisobotlarni ochish va yopish.	
<i>Закрыть</i>	Faol (berilgan) darchani yopish
<i>Открыть Форму</i>	Formani «Таблицы», «Конструктор» yoki «Формы» rejimida ochadi
<i>Открыть Модуль</i>	Modulni «Конструктор» rejimida ochadi
<i>Открыть Запрос</i>	So'rovni «Таблицы», «Конструктор» rejimida ochadi
<i>Открыть Отчет</i>	Hisobtni «Просмотр» rejimida ochadi
<i>Открыть Таблицу</i>	Jadvalni «Таблицы», «Конструктор» rejimida ochadi
Ma'lumotlarni bosmaga chiqarish	
<i>Открыть Форму</i>	Formani «Предварительный просмотр» rejimida ochadi

<i>Открыть Отчет</i>	Hisobotni «Просмотр» rejimida ochadi
<i>Открыть Таблицу</i>	Jadvalni «Таблицы» rejimida ochadi
<i>Вывести в Формате</i>	XLS – Excel uchun yoki RTF – Word uchun
<i>Печать</i>	Ma'lumotlarni (jadvallar, so'rovni yoki hisobotni) bosmaga chiqarish
So'rovni bajarish	
<i>Открыть Запрос</i>	So'rovni ishga tushiradi va «Таблицы» rejimida tanlangan yozuvlarni chiqaradi
Шартларни ростлигина текширish ва makrokomandalarni boshqarish	
<i>Отменить Событие</i>	Hodisani (bajariladigan amalni) bekor qiladi
<i>Команда Меню</i>	Standart menu komandasini bajaradi. Foydalanuvchi menusini yaratish uchun ishlatalishi mumkin
<i>Выход</i>	ACCESSning barcha darchalarini yopadi va undan chiqishni bajaradi
<i>Запуск Программы</i>	ACCESS BASIC protsedurasini bajaradi
<i>Запуск Макроса</i>	Boshqa makrosni ishga tushiradi
<i>Остановить все Макросы</i>	Barcha makroslarni, jumladan joriy makrosni qo'shib, ishini to'xtatadi
<i>Остановить Макрос</i>	Joriy makrosni ishini to'xtatadi
Qiymatlar o'rnatish	
<i>Обновление</i>	Bu komanda faol obyektga ma'lumotlarni tiklash uchun ishlatalishi mumkin («Формы» yoki «Таблицы» rejimida)
<i>Команды Клавиатуры</i>	Almashtirish buferida bosilayotgan tugmalar ketma-ketligini saqlaydi
<i>Задать Значение</i>	Ixtiyoriy boshqarish elementi qiymatini o'zgartiradi yoki tiklaydi
<i>Применить Фильтр</i>	Forma yoki hisobotlarda chiqarilayotgan axborotlarni cheklaydi

<i>Следующая Запись</i>	Joriy yozuvdan keyin yozuvni qidirishni amalga oshiradi
<i>Найти Запись</i>	Berilgan qidirish shartini qanoatlaniruvchi yozuvni qidiradi
<i>На Запись</i>	Ko'rsatilgan nomerli, shuningdek birinchi, oxirgi, oldingi, keyingi yozuvni joriy qiladi

Foydalanuvchi menyusini ko'rish va menu komandalarini bajarish

<i>Добавить Меню</i>	Yoyiluvchi menu yaratadi
<i>Команда Меню</i>	ACCESSni standart menyularidan birining amallarini bajaradi

Ekranga chiqarishni boshqarish

<i>Вывести на Экран</i>	Makros ishlashining oraliq amallarini bajarish haqidagi axborotlarni ekranga chiqaradi
<i>На Страницу</i>	Formada yoki hisobotda berilgan sahifaga o'tadi
<i>Песочные Часы</i>	Makrosni ishslash vaqtida kursorning (ko'rsatkich) shaklini o'zgartiradi
<i>Развернуть</i>	Faol darchanening o'lchamlarini to'la o'lchamgacha oshiradi
<i>Свернуть</i>	Faol darchani belgiga yig'ishtiradi
<i>Сдвиг размера</i>	Faol darchanening o'lchamlarini o'zgartiradi va siljitadi
<i>Восстановить</i>	Obyektni uning oldingi o'lchamlarida tiklaydi
<i>Выделить Объект</i>	Berilgan obyektli darchani ajratadi
<i>Обновление</i>	So'rov bilan bog'liq boshqarish elementlar ma'lumotlarini tiklaydi. Komanda ma'lumotlarini jadval yoki formada tiklaydi
<i>Установить Сообщения</i>	Tizim xabarları yoki ogohlantirishlarni tayinlaydi (bekor qiladi)
<i>Показать все Записи</i>	Oldin qo'yilgan filtrlarga e'tibor bermay, jadvalning barcha yozuvarini ko'rsatish

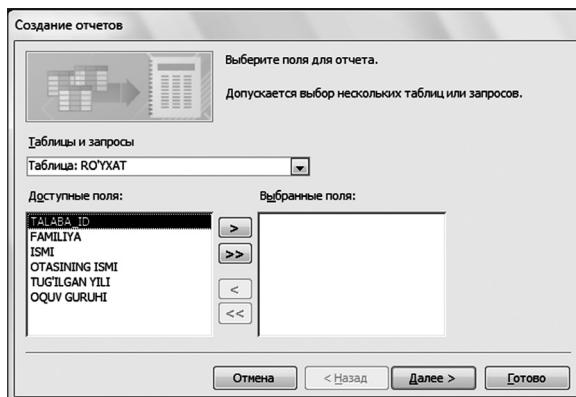
<i>Панель инструментов</i>	Ixtiyoriy standart yoki foydalanuvchi uskunalar panelini chiqaradi (olib tashlaydi)
Foydalanuvchiga xabar	
<i>Сигнал</i>	Tovushli signal beradi
<i>Сообщение</i>	Ogohlantirish yoki axborot xabarini chiqaradi
Obyektlar bilan ishlash	
<i>Копировать Объект</i>	Joriy ma'lumot bazasini ixtiyoriy obyekti – (yangi yoki eski nom bilan) ACCESSni, boshqa ma'lumot bazasiga nusxasini ko'chiradi
<i>Удалить Объект</i>	Obyektni olib tashlash (jadvalni, so'rovni, formani, hisobotni, makrosni, modulni)
<i>Вывести в Формате</i>	Obyektni Windows ilovasi formatida chiqarish
<i>Переименовать</i>	Joriy MBda obyektini qayta nomlash
<i>Отправить Объект</i>	Obyekt mazmunini faylga chiqaradi va uni ko'rsatilgan manzilga elektron pochtada uzatadi
<i>Преобразовать базу данных</i>	Obyektlarni boshqa MBdan (DBASe, Paradox, FoxPro) import yoki eksport qiladi
<i>Преобразовать Текст</i>	Ma'lumotlarni import yoki eksport qilishda qo'llaniladi. Axborotlarni matnli fayllarga uzatadi yoki ma'lumotlarni matnli fayllardan o'zgartirib beradi
<i>Преобразовать Электронную таблицу</i>	O'xhash elektron jadvallar uchun

Foydalanuvchi interfeysi formasini ishlab chiqishda, makros bajarilishi tugmani bosish bilan bog'lanadi, bunda ikki usulni qo'llash mumkin:

- makros yaratish va uni formaning mos tugmasi bilan bog'lash;

- formada tugma yaratish va uning xossalarini tavsiflashda mos makros ishlab chiqish.

Har ikki holda ham avval tugmali forma yaratish zarur.



4.4.1-rasm. Hisobotlar masteri.

Boshqaruvchi makroslarni yaratishda bajariladigan amallar ketma-ketligini quyidagi misolda ko‘rib chiqamiz. Shunday qilib, masala shunday qo‘yiladi.

«Создание баз данных» (Ma’lumotlar bazasini yaratish) tugma bosilganda, «Создание баз данных» deb nomlanuvchi muloqot formasini ochadigan (ishga tushiriladigan) makros bajarilsin. Buning uchun quyidagi amallarni bajarish zarur:

- tugmali forma «Конструктор» rejimida ochilsin;
- makros bilan bog‘lanishi zarur bo‘lgan tugma ajratilsin (faollashtirilsin). (Ko‘rilayotgan misolda bu «Создание баз данных» tugmasi);
- kontekstli menu faollashtirilsin (sichqoncha o‘ng tugmasi bilan);
- paydo bo‘lgan kontekstli menyuda «Свойства» buyrug‘i tanlansin.

Tugmaning xossalarini tavsiflovchi ochilgan darchada «Нажатие кнопки» yoki «Двойное нажатие кнопки» xossalarini tanlab olish zarur.

Xossani tanlab, bu misolda «Нажатие кнопки» makrosini ko‘rish tugmasi bosilsa, natijada makrosni tavsiflash talab (rasm) etiladigan darcha ochiladi.

Xuddi shu usul bilan ma’lumot bazasi bilan ixtiyoriy amal bajarilishi uchun tugmalar yaratiladi.

RO'YXAT1



5 октября 2013 г.

TALABA_ID	FAMILIYA	ISMİ	OTASINING ISMI	TUG'ILGAN YILI	OQUV GURUHI
1	Choriev	Komil	Xasnovich	1998	101
2	Adilova	Nasiba	Xikmatovna	1997	101
3	Suyunov	Xoliq	Nazarovich	1997	102
4	Xoliqov	Abdulla	Ergashovich	1997	104
5	Adilova	Nafisa	Abdullaevna	1998	105
5					

Страница 1 из 1

4.4.2-rasm. Hisobotlar ko'rinishi.

RO'YXAT1



=Date()

◆ Заголовок отчета

◆ Верхний колонтитул

TALABA_ID	FAMILIYA	ISMİ	OTASINING ISMI	TUG'ILGAN YILI	OQUV GURUHI
-----------	----------	------	----------------	----------------	-------------

◆ Область данных

TALABA_ID	FAMILIYA	ISMİ	OTASINING ISMI	TUG'ILGAN YILI	OQUV GURUHI
-----------	----------	------	----------------	----------------	-------------

◆ Нижний колонтитул

:Страница & [Page] из & [Pages]

◆ Примечание отчета

=Count(*)

4.4.3-rasm. Hisobotlar konstruktori masteri.

Xossalarni tavsiflash darchasi ACCESS muhitida obyektga yo'naltirilgan dasturlarning asosi bo'lib hisoblanadi. «Создание баз данных» (Ma'lumotlar bazasini yaratish) tugmasi bosilganda ochiladigan forma tasvirlangan.

Bajarilgan amallar natijasida tugma uchun makros yaratiladi va tugma bosilganda makros komandasi (yoki komandalari) bajariladi.

Bu masalani boshqacha ketma-ketlikda yechish ham mumkin, jumladan:

- tugmali forma ishlab chiqish;
- har bir tugma uchun, «Конструктор макросов» menyusidan foydalananib, makros ishlab chiqish. Bunda har bir makrosga nom (ism) tayinlash kerak;

- formani «*Конструктор*» rejimida ochish;
- tugma uchun «*Свойства*» darchasini ochish;
- mos satrda, ro'yxatlardan zarur makrosning nomini tanlash.

4.5. Foydalanuvchi menyusini ishlab chiqish

Axborot tizimida foydalanuvchi ishlashida qulay bo'lishi uchun samarali ma'lumot modelini (jadval va so'rovlarni o'zaro ta'siri va tarkiblari) yaratishdan tashqari qulay va «do'stona» foydalanuvchi interfeysi ishlab chiqish muhimdir. Foydalanuvchi interfeysi ishlab chiqish uskunalar panelini sozlash, foydalanuvchi menyusini yaratish, mutloq formalar ishlab chiqish bilan bog'liq.

Uskunalar panelini sozlash. Uskunalar panellarini, menuy satrlarini va kontekstli menyuni sozlash va yaratish uchun, shuningdek, ularning ko'rinishi va ishlashiga ta'sir qiluvchi xossalarni o'rnatish uchun muloqot darchasi ishlataladi. Uni ochish uchun «*Вид*» menyusida «*Панели инструментов*» komandasasi va «*Настстройка*» qism komandasini tanlab olish zarur.

Ochiq ma'lumotlar bazasi uchun maxsus komandalar panelini quyidagi ketma-ketlikda yarating:

- «*Вид*» menyusida «*Панели инструментов*» komandasini tanlang, so'ngra «*Настстройка*» qism komandasini tanlang;
- «*Панели инструментов*» vkladkasida «*Создать*» tugmasi bosiladi;
- «*Панели инструментов*» maydonida kerakli nomni kriting va «*OK*» tugmasini bosing;
- «*Панели инструментов*» vkladkasida «*Свойства*» tugmasini bosing;
- kerakli xossalarni o'rnatish va «*Закрыть*» tugmasini bosing.

Yangi uskunalar paneli «*Настстройка*» muloqot darchasining orqasida paydo bo'ladi. Uskunalar panelini yaratishni tugatish uchun quyidagi amallarni bajaring:

- «*Настстройка*» muloqot darchasidan tugmalar qo'shing;
- boshqa uskunalar panelidan tugmaning nusxasini oling yoki ko'chiring.

ESLATMA: Uskunalar paneliga foydalanuvchi menyusini qo'shish mumkin. Mavjud makroslarni ishga tushiradigan tugmalar bor bo'lgan uskunalar paneli avtomatik ajratiladi. Maxsus uskunalar panelini forma yoki hisobotga ulash mumkin.

Foydalanuvchi menyusini yaratish maxsus kontekstli menu quyidagi ketma-ketlikda yaratiladi:

- «*Вид*» menyusida «*Панели инструментов*» komandasasi va «*Настройка*» qism komandasi tanlanadi;
- «*Панели инструментов*» vkladkasida «*Создать*» tugmasini bosamiz;
- «*Панели инструментов*» maydonida kerakli nomni kriting va «*OK*» tugmasini bosing;
- «*Панели инструментов*» vkladkasida «*Свойства*» tugmasini bosing;
- «*Tun*» ro'yxati maydonida «*Контекстное меню*» qadamini (punktini) tanlab oling;
- «*Настройка*» bayroqchasini o'rnatiting yoki olib tashlang va «*Закрыть*» tugmasini bosing.

Kontekstli menu uskunalar paneliga qo'shiladi.

Muloqot formalari ishlab chiqish. Shu narsa oydinki, foydalanuvchi interfeyсини qandaydir muloqot formalari ketma-ketligidan iborat deb tasavvur qilish mumkin. Demak, muloqot formalarini yaratishga qadar foydalanuvchi interfeysi ssenariyasi yaratish ishlarini bajarish zarur.

Avval forma-zastavkalar yaratilishi zarur. Ularda ishlab chiqilgan tizimning asosiy vazifalarini ko'rsatish zarur. Shu formalardan foydalanuvchi o'z (foydalanuvchi) menyusini ham tashkil qilishi mumkin. Bu tugmali menu yoki silqib tushuvchi menu bo'lishi mumkin.

Silqib tushuvchi menyuni makroslari aniq ketma-ketligi ko'rinishida tashkil qilishi mumkin. «Yangi menu»ni har bir punkt uchun, keyingi bosqich qismi menyular tarkibini o'rnatuvchi mos makroslar ishlab chiqiladi.

Har bir bajarishga yuklanuvchi makrosning vazifasi ma'lumotlarni kiritish mos formasini mos ochishidan iborat. Bu usul bilan ko'p sathlilarni (bosqichlarini) loyihalash mumkin. Menyular yaratishning bu usulidan tashqari, menyulardan foydalanish uchun, komanda tugmalari ko'rinishida yaratish usuli ham keng

tarqalgan. Bundan tashqari forma maydoniga rasm ham qo‘yish mumkin.

Xulosa qilib aytganda, formalarini ishlash uchun qulay qilib yaratish kerak.

Savol va topshiriqlar

1. Microsoft ACCESS tizimida forma ko‘rinishlari va ularni konstruksiyalash usullarini aytib bering?
2. Forma konstuktori qanday bloklardan tashkil topgan?
3. Foydalanuvchi interfeysi tushunchasi nimani anglatadi?
4. Tugma (knopka)li forma nimani anglatadi?
5. Makroslar qanday sinflarga bo‘linadi?

V BOB. VISUAL BASIC FOR APPLICATIONS MUHITIDA BOSHQARUVCHI DASTURLAR ISHLAB CHIQISH

5.1. Visual Basic for Applications ning umumiylar xarakteristikasi

Visual Basic for Applications (VBA) dasturlash tili Microsoft office barcha ilovalarda ishlash uchun umumiylar uskuna bo'lib xizmat qiladi. Uning yordamida ixtiyoriy dasturlash masalalarini Microsoft Office vositalaridan ishlab chiqarish muhitida foydalanib, foydalanuvchi barcha ishlarini avtomatlashtirishdan tortib yirik ilovalar ishlab chiqishgacha hal qilishi mumkin.

VBA obyektga yo'naltirilgan til hisoblanadi. Access ilovalaridagi asosiy ish ma'lumotlar bilan ishlashga asoslanadi, shuning uchun ba'zi bir ma'lumotlar boshqarish kutubxonasini tafsiflab chiqamiz:

- DAO (Data Access objekts)
- ADO (Active Data Objects)
- JRO (Jet and Replications Objekts)

VBAning quyidagi komponentlari bilan tanishib chiqamiz:

- protsedura va funksiyalar;
- o'zgaruvchilar, konstantalar va ma'lumot toifalari;
- o'zgaruvchi va funksiyalarning ta'sir sohasi;
- boshqaruvchi konstruksiyalar tarmoqlanish va sikllar;
- sikllardan chiqish va protseduralar;
- modullar.

5.2. Protsedura va funksiyalar

VBA dasturini asosiy komponentlar protsedura va funksiyalar hisoblashda protsedura mos ravishda SUB va END SUB yoki Function va And Function operatorlari orasiga joylashtiriladi, masalan, ularning tuzilishi quyidagicha:

```
SUB <protsedura nomi> (<argument1> <argument2>.....)
<operator1>
<operator2>
.....
End Sub
Yoki
Function <funksiya nomi> (<argument1> <argument2>.....)
<operator1>
<operator2>
```

.....

<funksiya nomi>=< qaytariluvchi qiymat>

End Function

Funksiya protseduradan shu bilan farq qiladiki, uning nomi o'zgaruvchi sifatida kelishi mumkin va u funksiyaning chaqirish nuqtasiga qiymatini qaytarish uchun ishlataladi.

Yozilgan protsedura yoki funksiyani bajarish uchun uni chaqirish kerak. Argumentlar bilan berilgan protsedurani faqat boshqa protsedura yoki funksiyalardan chaqirish mumkin. Bunda uning nomini argumentlar haqiqiy qiymatlar ro'yxati bilan VBA operatorini birortasi bilan berish kerak.

Funksiyalar VBA ni alohida operatori bilan chaqirishdan tashqari uning nomini haqiqiy qiymatli argumentlar ro'yxati bilan VBA ga yozilgan dasturdagi formula yoki ifodaga to'g'ridan to'g'ri joylashtirib ham bajarish mumkin. Masalan, uning nomini Access hisobotlari va formalar, so'rovdagи hisoblanuvchi maydon formulalarga to'g'ridan to'g'ri joylashtirilishi ham mumkin. Argumentlar bo'limgan protseduralar (ular ba'zan makros buyrug'i deb ataladi) faqat boshqa protsedura va funksiyalardan chaqirilibgina qolmay, balki tezkor chaqirish tugmalar yordamida, yoyiluvchi menu komandalari yoki uskunalar paneli tugmalari yordamida ham chaqirish mumkin. Shuningdek, bunday protseduralarni har xil hodisalarini bajarish bilan bog'lash mumkin (masalan formadan tugmalarni sichqonchani bosib forma yoki hisobotlarni ochish, formasini boshqarish elementlariga ta'sir qilib va boshqalar).

Bunday protseduralar hodisani qayta ishlovchi protseduralar deyiladi. Argument qiymatlarini qabul qilib ishlaydigan protsedura yoki funksiyalar bunday xususiyatlarga ega emas. Agar chaqirilayotgan protsedura yagona nomga ega bo'lib, chaqirayotgan protsedura joylashgan modulda bo'lsa, uni chaqirish uchun nomini ko'rsatish yetarli. So'ngra argument haqiqiy qiymatlarini berishi mumkin (bunda ularni qavs ichiga olish shart emas).

Prosedurani chaqirishning yana bir usuli, call operatorini ishlatalib bajariladi. Oldin call operatori, so'ngra protsedura nomi, so'ngra qavs ichida parametrlar ro'yxati beriladi. Bunda qavslar ishlatalishi albatta shart.

Funksiyalarni chaqirishning eng ko'p ishlataladigan usuli tayinlash operatorining o'ng tomonidan funksiya nomini qavs ichida uning parametrlar bilan berilishi hisoblanadi.

Misol. CROSS RC nomini protsedurali, unga ikkita argumentni (konstanta va ifoda) uzatish bilan chaqirish:

CROSS 7,i+2

yoki

CALL CROSSRC(7,i+2)

Quyidagi misolda ikkita (left va mid) funksiyalarini chaqirish va ular qaytaradigan qiymatlarni ifoda ishlatishi ko'rsatilgan

yStr =left(y,1) & mid(y,2,1)

Prossedura funksiyalarga o'zgaruvchilarni uzatishning ikkita har xil usuli ishlatilishi mumkin: murojaat va qiymatlar bo'yicha.

O'zgaruvchi murojaat bo'yicha uzatilganda protsedura yoki funksiyada shu o'zgaruvchining xotiradagi adresi (yacheyska nomeri) uzatiladi. Chaqirayotgan protsedura haqiqiy parametr qiymatini o'zgartirishi mumkin.

Agar haqiqiy parametr qiymat bo'yicha uzatilsa, unda chaqirilayotgan protsedura yoki funksiya o'zgaruvchisining o'zini emas, shu parametr sifatida ishlatilayotgan haqiqiy parametrning faqat qiymatini oladi.

Haqiqiy parametr olingan qiymatlarini barcha o'zgarishlari (agar ular chaqirilayotgan protsedura bajarilayotgan bo'lsa) haqiqiy parametr o'zgaruvchisining qiymatiga ta'sir qilmaydi.

Protsedura yoki funksiyalarga parametrlarni uzatish usuli uning argumentlarini (formal parametrlarni) tavsiflashda ko'rsatiladi. Argument nomi oldida uzatish usulini oshkor tasvirlovchi (ByRef – murojaat bo'yicha uzatish, ByVal – qiymat bo'yicha) ko'rsatiladi. Agar parametrni oshkor uzatish usuli ko'rsatilmasa unda sukut bilan uzatishga murojaat bo'yicha deb hisoblanadi.

Misol: faraz qilaylik ikkita protsedura Main va Examples

Sub main()

A=10

B=20

C=30

Call examples(a,b,c)

Call MsgBox(a)

Call MsgBox(b)

Call MsgBox(c)

Sub Example1(x, ByVal, ByRef)

x=x+1

```
y=y+1  
z=z+1  
Call MsgBox(x)  
Call MsgBox(y)  
Call MsgBox(z)  
End Sub
```

Dastur odatda ko‘p protsedura va funksiyadan tashkil topadi va ular bitta yoki bir nechta har xil modullarda joylashishi mumkin.

Modullarda loyihalar birlashtiriladi. Bunda bitta loyihada bir nechta har xil umumiy modul yoki protsedurani ishlataladigan das-turlar joylashishi mumkin.

Bitta modulda joylashgan har bir protsedura yagona nomga ega bo‘lishi kerak; bunda loyihada bir nechta har xil modullar bo‘lishi mumkin. Agar loyihada bir xil nomli bir nechta protse-duralar bo‘lsa, unda protsedurani chaqirish uchun komanda ko‘rinishini quyidagicha yozish mumkin:

<modul nomi> < protsedura nomi>

Bunda modul nomi bir necha so‘zdan iborat bo‘lsa, unda no-mini kvadrat qavsga olib yozish kerak. Masalan, agar modul nomi < trafik protseduralar> protsedura esa <hisoblash> deb nomlansa, unda protsedura chaqirish komandasi quyidagicha bo‘ladi:

[Grafik protseduralar]. Hisoblash

Boshqa loyihalarda joylashtirilgan protseduralarni ham ish-latish mumkin. Bunda nomini aniqlashtirish uchun yana bir bosqich talab etilishi mumkin:

<loyiha nomi>.<modul nomi>.<protsedura nomi>.

5.3. O‘zgaruvchilar, konstantalar va ma’lumot toifalari

VBA tilida ham boshqa tillar kabi, vaqtinchalik qiymatlarni saqlash, parametrлarni uzatish va hisoblashlarni bajarish uchun o‘zgaruvchilar ishlataladi.

Odatda, o‘zgaruvchilarni ishlatishdan oldin ularni e’lon qilish kerak, ya’ni dasturda o‘zgaruvchi qanday nom bilan ishlatilishi, saqlashga mo‘ljallangan ma’lumot toifalari e’lon qilinadi. VBA tilida buning uchun Dim operatori ishlataladi. Uning yozilishi umumiy holda quyidagicha:

Dim <o‘zgaruvchi nomi> [AS<ma’lumot toifasi>]

VBA o‘zgaruvchilarini nomlashda quyidagilarga rioya qilinadi:

- o'zgaruvchi nomining uzunligi 255 simvoldan oshmasligi kerak;
- nomlar harf bilan boshlanadi va undan keyin harflar, raqamlar va tag chizig'i (_) ishlatalishi mumkin;
- nomlar yozilishida oraliq (probel) tinish belgilari yoki maxsus belgilarning ishlatalishi mumkin emas;
- o'zgaruvchi nomining oxiriga yana 6 ta maxsus belgi !, #, \$, %, &, @ lardan birini qo'shish mumkin.

Bu belgilar nomning qismi hisoblanmaydi. Agar dasturda bir paytda, string1\$ va string1 ishlatilsa, ular bitta satrni o'zgaruvchi nomiga murojaatni anglatadi:

- ma'lumot toifasi har xil belgi bilan aniqlangan o'zgaruvchi nomini ishlatib bo'lmaydi;
- o'zgaruvchi nomi sifatida VBA xizmatchi so'zlarni va standart obyektlarni ishlatish mumkin emas;
- VBAda e'lon qilinmagan o'zgaruvchilarni ham ishlatish mumkin.

Odatda Option Explicit operatori bilan boshlanadigan modulda VBA shu modulga tegishli barcha o'zgarishlarning e'lon qilishini talab qiladi, aks holda xatolik xabari chiqariladi. VBA ishlatiladigan ma'lumot toifalarning ro'yxati jadvalda keltirilgan.

5.3.1-jadval

VBA ma'lumot toifasi

Ma'lumot toifasi	Tavsiflash
Array	O'zgaruvchilar massivi. Massiv elementiga murojaat qilish uchun indekslar ishlataladi. Talab etiladigan xotira hajmi massiv o'lchamlariga bog'liq
Boolean	Ikkita mantiqiy qiymat true (rost) false (yolg'on) birini qabul qiladi. Talab etiladigan xotira hajmi 2 bayt
Byte	Ishorasiz son qiymati 0 dan 255 gacha. Talab etiladigan xotira hajmi 1 bayt
Currency	Pulli hisoblashlarni hisoblashda ishlataladi. Mumkin bo'lgan yaxlitlash xatoliklarini oldini olish maqsadida
Currency	O'nli nuqtadan keyin fiksirlangan sondagi ishoralar bilan pulli hisoblashlar bajarilishida ishlataladi. Mumkin bo'lgan qiymatlar diapazoni (chegarasi) – 922337203685977,5808 dan to 922337203685977,5807 gacha talab etiladigan xotira 8 bayt, sukut bilan toifa aniqlash uchun @ simvol ishlataladi

Date	Sana ma'lumotlarini saqlash uchun ishlataladi. Mumkin bo'lgan qiymatlar chegarasi (diapazoni) 1-yanvar 0100-yil to 31-dekabr 9999-yil. Talab etiladigan xotira hajmi 8 bayt
Double	Suzuvchi nuqtali ikkilangan (yuqori) aniqlikdagi sonli qiymatlar. Manfiy sonlar uchun mumkin bo'lgan qiymatlar diapazoni – 1,79769313486232E–308 dan – 4,94065645841247E – 324 musbat sonlar uchun mumkin bo'lgan diapazon 4,94065645841247E – 24 dan 1,79769313486232E308 gacha. Talab etiladigan xotira hajmi 8 bayt, ma'lumot toifasini sukut bilan aniqlash uchun # belgisi ishlataladi
Integer	Qisqa butun sonli qiymatlar uchun. Mumkin bo'lgan qiymatlar diapazoni – 32768 dan 32767 gacha talab etiladigan xotira hajmi 2 bayt, sukut bilan ma'lumot toifasini aniqlash uchun % belgi ishlataladi
Long	Uzun butun sonli qiymatlar o'zgarishi mumkin bo'lgan diapazoni – 2147483648 dan 2147483647 talab etiladigan xotira hajmi 4 bayt. Sukut bilan ma'lumot toifasini aniqlash & belgisi qo'yiladi
Object	Obyektlarga murojaatni saqlash uchun ishlataladi. Talab etiladigan xotira hajmi 4 bayt
Single	Oddiy aniqlikdagi suzuvchi nuqtali sonli qiymatlar. Manfiy sonlar qiymatlarining mumkin bo'lgan diapazoni – 3,402823 E–38 dan to – 1,401298 E – 45 gacha. Talab qilinadigan xotira hajmi 4 bayt. Sukut bilan toifani aniqlash uchun «!» simvoli ishlataladi
String	Satrli qiymatlarni saqlash uchun ishlataladi. Satr uzunligi 0 dan to 64 Kb gacha. Bitta simvolga 1 bayt xotira talab etiladi. Sukut bilan toifani aniqlash uchun «\$» simvoli ishlataladi
Variant	Har xil toifali ma'lumotlarni saqlash uchun ishlatalishi mumkin. Satr uzunligi 16 bayt plus 1 bayt satrli qiymatlarning har bir simvoli uchun. Sukut bilan toifani aniqlash ishlatilmaydi
Foydalanuvchi aniqlaydigan toifa	Foydalanuvchi aniqplaydigan ma'lumot toifasi. Ajratilgan xotira hajmi ma'lumot toifasiga bog'liq

O'zgaruvchilarni tavsiflaganda ma'lumot toifasini ko'rsatmasa ham (tushirib qoldirish) bo'ladi. Bunday hollarda o'zgaruvchi toifasi o'zgaruvchi nomi oxirgi belgisi bilan saqlanadi: bunda

belgilar @, #, \$, %, &, ! mos ravishda currency, double, integer, long, single yoki stringga mos keladi.

Masalan, \$ belgi satrli ma'lumot toifalarini aniqlovchi belgi bo'lganligi uchun text\$ nomli o'zgaruvchi avtomatik ravishda simvollar satri toifalaridagi o'zgaruvchi bo'lib hisoblanadi. Agar oxirgi belgi (simvol) yuqorida ko'rsatilib o'tilgan belgilardan birortasi ham bo'lmasa (ishlatilmasa) va toifa oshkor ko'rsatilmasa, unda bunday o'zgaruvchiga sukut bilan variant ma'lumot toifasi tayinlanadi. Bunday toifali o'zgaruvchida ma'lumotlarning ixtiyoriy toifasini saqlash mumkin.

Bitta protsedura ichida o'zgaruvchining oxirida faqat ma'lumot toifasi bilan farqlanuvchi o'zgaruvchilarni ishlatib bo'lmaydi.

Masalan, bir vaqtida Var\$ va Var% o'zgaruvchi ishlatilishiga ruxsat berilmaydi. Shuningdek, nomi oxirida toifani aniqlovchi (simvol) belgi bo'lgan o'zgaruvchilarni tavsiflovchi <as> <o'zgaruvchi toifasi> yordamida e'lon qilishga ruxsat berilmaydi.

Masalan, quyidagi e'lon qilishdan keyin xatolik haqida xabar olamiz:

Dim Var1\$ As STRING

Dim Vas2% As Integer

Protsedura yoki funksiyalar argumentlarining ma'lumot toifalarini aniqlash uchun protsedura yoki funksiyaning bevosita sarlavhasi ma'lumot toifasini tavsiflash uchun ishlatiladi. Masalan, protsedurani quyidagi bosh satrda satrli o'zgaruvchilarning parametrlari tavsiflangan.

Sub SplitStr(str1 As String, str2 As String, str3 As String).

Funksiya qaytarayotgan funksiya qiymatining toifasi funksiya bosh satrining oxirida ko'rsatiladi. Masalan:

Function FindDplitspace (str1 As string) As Integer

Funksiya qaytarayotgan qiymatni qisqa toifali o'zgaruvchi kabi tavsiflanadi. Nomlanuvchi konstantalarni ishlatishni ko'rib chiqamiz. Ularni tavsiflash uchun const operatori ishlatiladi. Bu operatorlarning yozilish sintaksisi quyidagicha:

Const <konstanta nomi> (As< ma'lumot toifasi>)=<ifoda>

Bu yerda <ifoda> bu ixtiyoriy qiymat yoki hisoblangan qiymatni konstanta sifatida ishlatadigan formula. Masalan, quyidagi operator Maxlen% butun qiymatli konstanta saqlaydi.

Const Maxlen%=30

Foydalanuvchi tomonidan aniqlanadigan konstantalardan tashqari oldindan aniqlangan tizim konstantalar ham mavjud.

Kiritilgan konstantalar nomida standart kelishuvlardan foydalanadi. Bu kelishuvlar konstantani qaysi ilova obyektlariga tegishli ekanligini aniqlashga imkon beradi. Masalan, Access obyektlariga tegishli kiritilgan konstantalar Ac old qo'shimchasi bilan boshlanadi. Excel ga tegishlilari «xl» old qo'shimchasi bilan boshlanadi va h.k. Masalan,

Doc2.OpenForm <orders> Ac Normal stlinkCriteria komanda Ac Normal komandada Access qurilgan konstantasi ishlatiladi.

Obyektlarga murojaat. Visual Basic oddiy o'zgaruvchilardan tashqari obyektga murojaatni ifodalovchi o'zgaruvchilar ham ishlatiladi. Bunday o'zgaruvchilarni ishlatish faqat dastur matnini soddalashtiriladi va qisqartiribgina qolmasdan uning ishlash tezligini yetarlicha oshiradi.

O'zgaruvchi obyektlarini ishlatish oddiy o'zgaruvchilarni ishlatishdan farqlanadi. Ishlatishdan oldin uni faqat e'lon qilibgina qolmay, unga mos obyektni Set operatori yordamida tayinlash kerak.

Bunday e'lon qilish sintaksisi va tayinlanishi quyidagicha:

Dim <o'zgaruvchi nomi> As object

Set <o'zgaruvchi nomi>=<obyektga murojaat>

Ba'zi hollarda bunday o'zgaruvchilarni e'lon qilishdan oldin aniq obyekt toifasini ko'rsatish qulay. Bunda office ni obyekt modelidan aniq ixtiyorisini ko'rsatish mumkin:

Set MyBase=DBEngine.Workspaces(0).Database(0).

Bunday e'lon qilish va tayinlashdan so'ng MyBase o'zgaruvchisini joriy ochiq ma'lumot bazasiga murojaat uchun ishlatish mumkin. Bunday murojaatlar tezroq qayta ishlaydi.

Massivlar. Massiv bir vaqtida bir toifaga tegishli bir nechta qiyamatni saqlovchi o'zgaruvchidir.

Massivning rasmiy ta'rifi bir toifali indekslangan o'zgaruvchilar majmuasidir. Massivda ishlatiladigan indekslar soni ham har xil bo'lishi mumkin. Odatda bir yoki ikki indeksli massivlar ko'proq ishlatiladi. VBA 6.0 tagacha indeksli massiv ishlatilishi mumkin.

Bitta indeksli massivlar bir o'lchamli massivlar deyiladi, ikkitalisi – ikki o'lchamli va hokazo. Massivni dasturda ishlatishdan avval uni albatta Dim operatori yordamida e'lon qilish kerak va massivda saqlanuvchi qiyatlarning toifalari ko'rsatilishi kerak.

Massivning barcha qiymatlari bitta ma'lumot toifasiga tegishli bo'lishi kerak.

Bu cheklanishni chetlab o'tish uchun massivni e'lon qilishda variant toifasini ishlatib amalga oshirish mumkin. Bunda massiv elementlari har xil toifali ma'lumot qiymatlarini qabul qilishi mumkin. Massivni e'lon qilish operatori sintaksisi

Dim< massiv nomi > (<o'lcham 1>,<o'lcham 2>,...) As <ma'lumot toifasi> qavs ichida ko'rsatilgan kattaliklar bilan massivning o'lchamlari – indekslar soni va har bir aniq indeksning maksimal qiymati beriladi.

Bunda massiv elementlarini indekslash sukut bilan 0 dan boshlanadi. Masalan, DIM ARRAY(9) AS INTEGER ARRAY butun toifali o'zgaruvchi 10 ta elementli bir o'lchamli massivini aniqlaydi.

DIM ARRAY (4,9) AS VARIANT

ARRAY nomli variant universal toifali o'zgaruvchini 50 ta (5x10) elementdan iborat ikki o'lchamli massivni aniqlaydi. Massivni e'lon qilishda indeksning faqat yuqori chegarasini emas, balki uning quyi chegarasini ham ko'rsatish mumkin, ya'ni massivli aniq indeksni o'zgartirish diapazonini ham ko'rsatish mumkin, bunda quyi chegara ixtiyoriy musbat son bo'lishi mumkin. Bunday e'lon qilishning sintaksisi

Dim <massiv nomi> (<tin 1> To <maks 1>) As <ma'lumot toifasi>

Keltirilgan misollarda massivlar fiksirlangan massivlar hisoblanadi. Ularda massivni Dim operatori bilan e'lon qilinganda elementlar soni oshkor ravishda ko'rsatiladi. Bunday massivlar statik massivlar deyiladi. VBA da dinamik massivlarni ishlatish mumkin. Dinamik massivlarni e'lon qilishda ularning o'lchamlari fiksirlanmaydi.

Dinamik massivlarning o'chamlarini aniqlash bevosita dasturni bajarish vaqtida amalga oshiriladi. Dinamik massivni Dim operatori bilan aniqlaganda Dim operatorida massiv nomidan keyin bo'sh qavslar va o'zgaruvchilar toifasi yoziladi. Indekslar soni va ularning o'zgarish diapazoni berilmaydi. Massivni ishlatishdan avval ReDim operatorini kiritish zarur. U dinamik massivni o'lchamlari va indekslarini o'zgarish diapazonini beraadi. Dinamik massivning o'lchamlari va e'lon qilish sintaksisi quyidagicha:

Dim <massiv nomi> ()As <ma'lumot toifasi>
 ReDim <massiv nomi>(<o'lcham 1>,<o'lcham 2>,...)
 Misol: Dim dArray () As Variant
 ReDim dArray (1, 2)
 dArray (0, 0) = 2
 dArray (0, 1) = 3
 k=dArray (0, 0) + dArray (0, 1)
 ReDim dArray (k)
 dArray (0) = (Строка 1)

5.4. O'zgaruvchilarning ta'sir sohasi va protseduralar

VBAda barcha protseduralar, funksiyalar, o'zgaruvchilar va konstantalar o'zining ta'sir doirasiga ega. Buning ma'nosi shuki, ularning barchasi dastur kodining faqat ma'lum qismida ishlatalishi mumkin. Ya'ni qayerda e'lon qilingan bo'lsa, u yerda ishlataladi. Masalan, agar A o'zgaruvchi Dim operatori yordamida Proc1 nomli protsedura tanasida ishlatsa, unda ish protsedura uning ta'sir doirasi hisoblanadi. Agar proc2 nomli boshqa protsedura bo'lsa, unda xuddi shu o'zgaruvchini e'lon qilmasdan ishlatib bo'lmaydi.

O'zgaruvchilarning ta'sir sohasi. O'zgaruvchilar dasturning qaysi yerida va qanday tasvirlanganiga qarab va u xotirada qancha uzoq yashashi va unga tayinlangan qiymatni saqlashi, uning ta'sir etish sohasini aniqlaydi. O'zgaruvchilar ta'sir sohasini aniqlashning uchta har xil sathi mavjud: protsedura sathi, modul sathi va loyiha sathi.

Protsedura sathida o'zgaruvchini aniqlash uchun, uning tavsifi protsedura tanasiga joylashtiriladi.

Modul sathida protsedurani aniqlash uchun va shu bilan uni shu moduldagi barcha protseduralarda birgalikda ishlatalishi mumkin bo'lishi uchun, uni tavsifini modulni e'lon qilish sekiyasiga — protsedura yoki funksiya matni oldiga joylashtirish kerak.

Bunda ta'sir sohasini oshkor tavsiflab ishlalistish ham mumkin. Bu holda **Dim** kalit so'zi o'rniiga **Private** kalit so'zi ishlataladi.

O'zgaruvchini loyiha sathida tavsiflash uchun, uni tavsifni loyiha modullaridan birini e'lon qilish sekiyasiga joylashtirish zarur. Bunda albatta **Public** kalit so'zi ishlatalishi zarur.

Shunday tavsif qilingan o‘zgaruvchini loyihaning ixtiyoriy modulida ishlatalishi mumkin. Yuqorida aytib o‘tilgan tavsiflar va ta‘riflar konstanta va massivlarga ham tegishli.

O‘zgaruvchilar uchun ularni tavsiflashning yana bir usuli bor.

Bu usul o‘zgaruvchining sathini o‘zgartirmasdan protserdura sathida tavsiflanadigan va uni ishi tugashidan keyin o‘zgaruvchini qiymatini saqlash imkonini beradigan usul hisoblanadi.

Buning uchun statik o‘zgaruvchisini statik o‘zgaruvchi deb tavsiflab ishlatalish kerak. Bunday o‘zgaruvchi unga ajratilgan xotiradan joyni va o‘zining qiymatini tavsiflagan protsedura ishlashi tugagandan so‘ng ham saqlaydi. Shunga qaramasdan statik o‘zgaruvchini boshqa protsidualarda ishlatib bo‘lmaydi, uning ishslash vaqtি o‘zgaradi, lekin ta’sir sohasi o‘zgarmaydi. Agar statik o‘zgaruvchi tavsiflangan protsedura yana qayta chaqirlisa, unda bu o‘zgaruvchi o‘zining avvalgi qiymatini saqlab qoladi.

Oldingi chaqirishda oddiy (statik bo‘lmagan) o‘zgaruvchilar har safar inisializatsiyalanadi va protseduraga kirishda bo‘sh qiymatni oladi.

Protsedura va funksiyalarning ta’sir sohasi. Protsedura va funksiyalar ta’sir sohasi faqat ikkita sathiga egadir: modul sathi va loyiha sathi.

Sukut bilan loyiha sathi ishlataladi. Loyihada protsedura yoki funksiya, ixtiyoriy boshqa protsedura yoki funksiya bilan chaqirilishi mumkin. Loyerha sathida protsedura va funksiya tavsiflashda majburiy bo‘Imagan Public kalit so‘zi ishlatalishi mumkin. Agar modul sathida ishlataladigan protsedurani tavsiflash zarur bo‘lsa, buning uchun Private kalit so‘zi ishlataladi.

Lekin protsedurani tavsiflash, uning mustaqil ishslashini taqiqlaydi. Uni faqat boshqa protseduradan chaqirish mumkin.

Protsedura yoki funksiya tavsiflashda Static kalit so‘zi ishlatalishi mumkin. U protseduraning ta’sir doirasiga hech qanday ta’sir qilmaydi, lekin shu protsedura yoki funksiya ichida tavsiflangan barcha o‘zgaruvchilarga ta’sir qiladi. Bu holda barcha mahalliy o‘zgaruvchilar static status oladi va shunga asosan protsedura ishi tugallangandan so‘ng xotirada saqlanib qoladi. Uni takror chaqirishda ular o‘zlarining avvalgi qiymatlarini saqlab qoladi:

Quyidagi misolni ko‘rib chiqamiz:

Public A1 As String

Private A2 As Integer

```

Dim A3 As Single
Sub Proc1()
    Dim A4 As Integer
    Static A5 As integer
    A1= «Satrli matn»
    A2=2
    A3=3.14
    A4=A4+4
    A5=A5+5
    MsgBox A4
    MsgBox A5
End Sub
Sub Proc1
    MsgBox A1
    MsgBox A2
    MsgBox A3
    MsgBox A4
    MsgBox A5
    Proc1
End Sub

```

Bu misolda A1 butun loyihada aniqlangan (Public kalit so‘zi ishlatalayapti). A2 va A3 o‘zgaruvchilar modul doirasida aniqlangan. A4 o‘zgaruvchi faqat Proc1 protsedura sohasida aniqlangan, A5 o‘zgaruvchi Proc1 protsedura jismida aniqlangan bo‘lsa ham u statik o‘zgaruvchi deb e’lon qilinayapti. Proc2 protsedurasi chaqirilganda quyidagi amal bajariladi: bu protseduradan Proc1 protsedurasi chaqiriladi, u beshta o‘zgaruvchilarga A1, A2, A3, A4 va A5 qiymat tayinlaydi, so‘ngra, A4 va A5 o‘zgaruvchilarning joriy qiymatini (MsgBox) muloqot darchasiga chiqaradi. Bu protsedura ishini yakunlagandan keyin Proc2 protseduradan beshta o‘zgaruvchining barcha joriy qiymatlari chiqariladi. A1:A3 o‘zgaruvchilar o‘z qiymatini saqlab qoladi, chunki ular modul doirasida e’lon qilingan, A4 va A5 o‘zgaruvchilar esa 0 (nol) qiymat (bo‘s sh qiymat)ini oladi. Chunki bu o‘zgaruvchilarning e’lon qilingan sohasi faqat Proc1 bo‘lib hisoblanadi. So‘ngra, yana bir bor Proc1 protsedura chaqiriladi va u yana ekranga A4 va A5 o‘zgaruvchilarning qiymatlarini chiqara boshlaydi. Bunda A4 o‘zgaruvchi yana 4 (to‘rt) qiymatini oladi. Chunki yana yangidan chaqirilganda bu o‘zgaruvchiga yangidan xotira ajratiladi va

u bo'sh qiymat bilan initsiallashadi (boslang'ich qiymat oladi). A4 dan farqli ravishda A5 statik o'zgaruvchi deb e'lon qilingan, u oldingi qiymatini bu protsedura tomonidan chaqirilgandagi qiymatini saqlab qoladi va natijada uning qiymati qaytadan chaqirilganda 10 ga teng bo'lib qoladi.

5.5. VBA ni boshqarish konstruksiyalari

VBA da ham boshqa dasturlash tillari kabi, dasturni boshqarish tartibini o'zgartirish imkoniyatini beradigan har xil boshqarish konstruksiyalari bor. Agar boshqarish konstruksiyalari (operatorlari) ishlatsmasa, dasturlash tilining operatorlari ketma-ket bajariladi. Birinchi operatordan tortib oxirgi operatorgacha (dasturning operatorlari birinchidan boshlab oxirigacha ketma-ket bajariladi). Lekin amaliyotni ko'pgina masalarini yechishda operatorning kelish tartibini birorta shartning bajarilishiga qarab o'zgartirish talab etiladi. Bunda ba'zi operatorlarning bajarilishini o'tkazib yuborib yoki aksincha ularni bir necha bor takror bajarishga to'g'ri keladi.

Dasturchilik amaliyoti shuni ko'rsatadiki, axborotlar qayta ishslashni ixtiyoriy algoritmlarini amalga oshirish uchun ikki ko'rinishdagi boshqarish ko'rsatmalari (buyruqlari) yetarli hisoblanadi: tarmoqlanish va sikllar.

Tarmoqlanish. Tarmoqlarni boshqarish konstruksiyalari biorsta shartni tekshirish va uning natijasi asosida u yoki bu operatorlarni boshqarish imkoniyatlarini beradi.

VBA da tarmoqlanishni tashkil qilish uchun **if** operatorining xilma-xil shakli va **select case** operatori ishlataladi.

IF operatori oddiy qisqa bitta shartni tekshirish uchun ishlatiladi. Tekshirish natijasiga qarab bitta operator yoki bir nechta operatorlik bloki bajariladi yoki o'tkazib yuboriladi. IF operatorining qisqa shakli bitta satrli yoki blokli forma shakliga ega bo'lishi mumkin. IF operatorining qisqa bir satrli shakli quyidagicha yozilishi mumkin.

If «shart» Then «operator»

IF operatorining qisqa blokli shakli quyidagi ko'rinishda bo'ladi.

If <shart> Then
<Operatorlar bloklari>

ELSE

<Operatorlar bloki2>

End If

Shart sifatida True(rost) va False (yolg'on) qiymatlar beruvchi mantiqiy ifoda yoki ixtiyoriy arifmetik ifoda ishlatalishi mumkin. Agar arifmetik ifoda ishlatilmasa, bu ifoda 0 qiymatli mantiqiy false qiymatiga ekvivalent. Agar shart false ifodalarni qaytarsa, qisqa tarmoqlanish operatori tanasini tashkil qiluvchi operator yoki operatorlar bloki ishlagmaydi.

If operatori to'la formasi quyidagi hollarda ishlataladi: ikkita har xil operatorlar bloki mavjud bo'lib, shartli tekshirish natijasida shulardan birini bajarish zarurati paydo bo'lsa, bunday if operatorini bitta satrda yozib bo'lmaydi va u doim blok shaklida yoziladi.

If <shart1> Then

<operatorlar bloki1>

Else If <shart2> Then

<operatorlar bloki2>

Else If <shart3> Then

<operatorlar bloki3>

.....

Else If <shartN> Then

<operatorlar blokiN>

Else

<operatorlar bloki_Else >

End If

Agar shart rost bo'lsa, unda Then va Else kalit so'zлari orasida joylashgan birinchi blok operatorlari bajariladi, aks holda Else va End if kalit so'zлari orasida joylashgan ikkinchi blok operatorlari bajariladi.

Ba'zi hollarda bir nechta alternativ amallar guruhi amallaridan bittasini tanlab olib amal bajarish talab qilinadi. Bunday hollarda if operatori if...then...Else operatorlar ketma-ketligini (zanjirini) ishlatalish mumkin.

If <shart1> Then

<operatorlar bloki1>

Else If <shart2> Then

<operatorlar bloki2>

Else If <shart3> Then

<operatorlar bloki3>

.....

Else If <shartN> Then

<operatorlar blokiN>

Else

<operatorlar bloki_Else >

End If

Bunday if...Then...Else if operatorlar zanjiri yaxshi moslanuv-chanlikka ega va ko‘p muammolarni yechish imkonini beradi, lekin, agar bir necha imkoniyatlardan birini tanlash hamma vaqt bitta ifodaning har xil qiymatlariga asoslangan bo‘lsa, unda maxsus shu maqsadlarga mo‘ljallangan select case tanlash operatorini ishlatish qulay. Uning ko‘rinishi quyidagicha:

Select Case <tekshirilayotgan ifoda>

Case <qiymatlar ro‘yxati1>

<operatorlar bloki1>

Case <qiymatlar ro‘yxati2>

<operatorlar bloki2>

Case <qiymatlar ro‘yxati3>

<operatorlar bloki3>

.....

Case Else

<operatorlar bloki_Else>

End Select

Tekshirilayotgan ifoda select case operatori boshlanishida hisoblanadi. Bu ifoda ixtiyoriy toifali qiymatlarni qaytarishi mumkin. Ifodalar (qiymatlar ro‘yxati) bir-biridan vergul bilan ajratilgan bir yoki bir necha ifodalardan iborat bo‘ladi. Operatorning bajarilishida shu ro‘yxat elementlaridan hech bo‘lmaganda bittasi tekshirilgan ifodaga mosligi tekshiriladi. Ifoda ro‘yxati elementlari quyidagi shakllardan biriga ega bo‘ladi.

<ifoda >

(bu holda tekshirilayotgan ifoda qiymatini berilgan ifoda qiymatiga mosligi tekshiriladi)

<ifoda 1> to <ifoda 2>

Bu holda tekshirilayotgan ifoda qiymati ko‘rsatilgan qiymatlar diapazoni mos joylashishi tekshiriladi.

If <mantiqiy operator> <ifoda>

Bu holda tekshirilayotgan ifoda ko'rsatilgan qiymat bilan mantiqiy operator yordamida solishtiriladi. Masalan: if>=20 bajarilgan hisoblanadi, agar tekshirilayotgan qiymat 20 dan kichik bo'lsa.

Agar ro'yxat elementlaridan biri tekshirilayotgan ifodaga mos kelsa, unda shunga mos operatorlar bajariladi va select case operatorining ishi yakunlanadi, ifoda ro'yxatining qolganlari tekshirilmaydi.

Agar barcha shu ro'yxatning birorta ham elementi tekshirilayotgan ro'yxatidagi elementlariga mos kelmasa, u holda Else dan keyin joylashgan (agar u bo'lsa) operatorlar guruhi bajariladi.

Sikllar. VBA siklini tashkil qilishini yetarli vositalariga ega. Ularni ikki guruhg'a bo'lish mumkin:

- DO...Loop shartli sikllar;
- FOR...NEXT sanoqli sikllar.

DO LOOP sikllari sikl tanasini tashkil qiluvchi operatorlar bloki necha marta qayta bajarishlar soni noma'lum bo'lganda ishlataladi. Bunday sikl birorta aniq shart bajarilguncha, ishini davom ettiradi.

DO...LOOP operatorining 4 ko'rinishi mayjud. Ular shartini va bu tekshirishni bajarish vaqt bilan farqlanadi.

Ularning konstruksiyalari jadvalda keltirilgan.

Konstruksiya	Tasviri
DO while <shart> <operatorlar bloki > LOOP	Bu konstruksiyada sikl ishining tugashi shartining qiymati sikl operatorlarini bajarishga qadar tekshiriladi
DO <operatorlar bloki> LOOP while <shart>	Bu konstruksiyada sikl ish tugashi shartining qiymati operatorini bajargandan keyin tekshiriladi
DO Until <shart> <Operatorlar bloki> LOOP	Bu konstruksiyaga sikl ishi tugashi shartining qiymati, operatorlar bajarilgunga qadar ko'rsatiladi
DO <operatorlar bloki> LOOP Until <shart>	Bu konstruksiyada sikl operatorining ishi tugashi shartining qiymati, operatorlar bajarilgandan keyin ko'rsatiladi

Shuningdek, sanoqli FOR...NEXT operatorining 2 ta ko'rinishi bor. Ko'p hollarda, massivlarni qayta ishlashda, shuningdek, birorta operatorlar guruhini ko'rsatilgan marta takroran

bajarilishi talab qilinganda schotchikli FOR...NEXT sikl operatoridan foydalaniladi.

Bu operator DO...LOOP siklidan farqli ravishda schotchik deb nomlanuvchi maxsus o'zgaruvchi ishlatalidi. Schotchik qiymati sikl tanasi har gal bajarilganda berilgan qiymatga ko'payadi yoki kamayadi. Bu o'zgaruvchi qiymati tayinlangan qiymatga erishganda siklning bajarilishi tugaydi. Bu ko'rinishdagi siklning sintaksisi quyidagicha (kvadrat qavslarda majburiy bo'limgan qiymatlar ko'rsatilgan):

For <schotchik> = <boshlang'ich qiymat> To <oxirgi qiymat>

[Step <orttirma>
<operatorlar bloki>
Next <schotchik>

Bu sikl konstruksiyasi bo'yicha ba'zi fikrlarni aytib o'tamiz:

- <orttirma> musbat yoki manfiy son bo'lishi mumkin. Agar manfiy orttirma ishlatalisa, sikl hech bo'limganda bir marta ishlatalishi uchun sikl oxirgi qiymat boshlang'ich qiymatdan kichik yoki unga teng bo'lishi kerak;

- sikl yakunlangandan keyin FOR...NEXT schotchik yordamida ishlataligan o'zgaruvchi qiymati oxirgi qiymatdan katta (agar orttirma musbat bo'lsa), yoki oxirgi qiymatdan kichik (agar orttirma manfiy bo'lsa) bo'ladi;

- agar boshlang'ich va oxirgi qiymatlar teng bo'lib qolsa, sikl tanasi faqat bir marta bajariladi.

FOR...NEXT siklining yana bir ko'rinishi FOR Each...NEXT siklidir.

Bu sikl VBAda massivlarni tashkil qiluvchi obyektlarni qayta ishslashda foydalaniladi. Bu ko'rinishdagi siklda schotchik qatnashmaydi, sikl tanasi massiv yoki obyektlar oilasining har biri uchun ishlataladi.

Bu operator sintaksisi quyidagicha:

For Each <element> In <мажмуда>
<operatorlar bloki>
Next [<element>]

Bu yerda <element> – bu obyektlar oilasi elementlari murojaat qilish uchun ishlataladigan o'zgaruvchi:

<мажмуда> massiv yoki oila nomi.

Shu siklga doir misol keltiramiz. Quyida protsedura joriy ochiq ma'lumot bazasining barcha maydonlar ro'yxatini bosmaga chiqarish uchun mo'ljallangan protsedura keltirilgan:

```
Public Sub EnumerateAllFields ()  
    Dim MyBase As Database  
    Dim tdf As TableDef, fid As Field  
    Set MyBase = DBEngine.Workspaces(0). Databases(0)  
    For Each tdf In MyBase.TableDefs  
        Debug.Print "Jadval:" & tdf.Name  
        For Each fid In tdf.Fields  
            Debug.Print "Maydon:" & fid.Name  
        Next fid  
    NEXT TDF  
    Set MyBase = Nothing  
    End Sub  
    ub = Ubound (dArrey)  
    fFound = False  
    For I = Lbound (dArrey) to ub  
        If dArrey(i) = search Value Then  
            fFound = True  
        Exit For  
    NEXT
```

Bu misolda Dim operatori bilan quyidagi o'zgaruvchilar e'lon qilingan:

My Base – <ma'lumot baza obyekti>
tdt – ma'lumot baza jadvali
fid – jadval nomi.

Set operatori MY Base o'zgaruvchiga joriy ochiq ma'lumot bazasining nomi qiymatini tayinlaydi. So'ngra har bir aniqlangan jadval uchun jadval nomi, so'ngra xuddi shu kabi kiritilgan siklda maydon nomlari bosmaga chiqariladi.

5.6. Sikl va protseduralardan chiqish

Odatda protsedurani bajarilishi uning oxirgi operatoridan keyin tugaydi, siklning bajarilishi esa sikl ishini tugallash sharti bajarilgandan keyin yakunlanadi.

Lekin ko'p hollarda protsedura operatorlarining ba'zilarini bajarmay, uni tugatish yoki uning siklini muddatdan avval yakunlashga to'g'ri keladi.

Masalan: agar protsedurani bajarish jarayonida keyingi hisoblashlarni ma'nosiz (ahamiyatsiz) qiluvchi xatolik paydo bo'lsa, protseduradan darhol chiqish komandasini (buyrug'ini) bajarish kerak.

2-misol. Agar FOR...NEXT sikli massivda kerakli qiymatni qidirish uchun ishlatiladigan bo'lsa, qidirilayotgan element topilgandan so'ng, massivning qolgan elementlarini ko'rib chiqishni davom ettirishning ma'nosi yo'q. Boshqarish konstruksiyalaridan muddatdan avval chiqishni Exit operatorining bittasi bilan amalga oshirish mumkin.

DO...LOOP siklidan muddatdan avval chiqish uchun Exit DO operatori ishlatiladi, FOR siklidan chiqish uchun Exit FOR operatori ishlatiladi. Protsedura va funksiyalardan muddatdan avval chiqish uchun mos ravishda Exit Sub va Exit FUNCTION operatorlari ishlatiladi.

```
ub = Ubound (dArrey)
fFound = False
For I = Lbound (dArrey) to ub
    If dArrey(i) = search Value Then
```

```
fFound = True
Exit For
NEXT
```

Umuman olganda Exit operatorini ishlatish to'la o'rinli hisoblanmasa ham ulardan noiloj hollarda foydalanish tavsiya etiladi. Bu operatorni haddan tashqari ko'p ishlatish, yozilgan dastur matnini tushunishni va uni sozlashni qiyinlashtiradi.

```
i = Lbound (dArrey)
ub = Ubound (dArrey)
fFound = False
do
    If dArrey(i) = search Value Then fFound = True
    i = i + 1
Loop Until (i > ub) or fFound
```

5.7. Modular

Modul VBA da bitta dastur birligiga to'plangan, ilovalar uchun Visual Basic tilida yozilgan tavsiflar protseduralar to'plamidir.

Modullarning ikkita asosiy turi mavjud: klass (sinf) modullari va standart modullar.

Moduldagi har bir protsedura FUNCTION protsedura funk-siyasi yoki sub protsedura qism dasturi bo'lishi mumkin.

Klass modullari. Aniq formalar yoki hisobotlar bilan bog'langan dasturlar klass modullari hisoblanadi.

Klass modullari forma yoki hisobotlardagi hodisalarga javoban ishga tushadigan hodisalar qayta ishlovchi protseduralarni o'z ichiga oladi. Hodisalarni qayta ishlovchi protseduralar firma yoki tashkilot xarajatini boshqarish uchun va ularni jadvallarga qayta ishslash uchun, masalan, tugmani bosish protseduralari ishlataladi.

Forma yoki hisobotlarni qayta ishlovchi birinchi protsedurani yaratishda avtomatik ravishda u bilan bog'langan forma yoki hisobot modullari yaratiladi. Forma yoki hisobot modullarini qu-rish uchun konstruktor rejimida uskunalar paneli programma tugmasini bosish yetarlidir.

Formalar va hisobotlar modullarining protseduralarida standart modullarga qo'shilgan protseduralar chaqirishni saqlash mumkin.

Standart modular. Standart modular hech qanday modullar bilan bog'liq bo'lмаган umumiy modellarni, shuningdek, ixtiyoriy ma'lumot baza darchasidan ishga tushirish mumkin bo'lgan protseduralarni saqlaydi. Standart modullar global o'zgaruvchilarni e'lon qilish uchun ishlab chiqiladi. Agar modul protseduralarida ilovaning aniq obyektlariga (forma, hisobot, boshqarish elementi) murojaatlar ko'rsatilmasa, bunday modul ACCESS boshqa ilovalarda ishlatalishi mumkin.

Savol va topshiriqlar

1. VISUAL BASIC (VBA)da qanday ma'lumot toifalari ishlataladi?
2. VBAda dasturlashda qanday operatorlar ishlataladi?
3. VBA qism dasturlari va ularning ko'rinishlarini aytib bering.
4. Funksiya qism dasturlarining protseduradan farqini izohlab bering.
5. O'zgaruvchining ta'sir doirasi deyilganda nima tushuni-ladi?

VI BOB. SQL TILIDA SO'ROVLAR YARATISH

6.1. SQL standartlari va tarkibi

Birinchi relatsion ma'lumotlar bazasini boshqarish tizimlari (MBBT) IBM kompaniyasi tomonidan 1970-yil boshlarida yaratilgan. Xuddi o'sha vaqtarda shu sistemalarda ishlashga mo'ljallangan ma'lumotlar tili yaratilgan edi. Uni tajriba nusxasi SEQUEL (structured English query language – strukturalashgan inglizcha so'rovlар tili) keyinchalik bu qisqartma ixchamlashtirilib, SQL (structured query language) deb nomlandi. Aniq qilib aytganda bu MBBT qism tili hisoblanadi, chunki, MBBT boshqa til vositalariga ham ega. 1981-yilda IBM relatsion MBBT SQLIDS ishlab chiqdi. Xuddi shu vaqtida Relative Software Inc (hozirgi ORACLE Corporation) kompaniyasi relatsion MBBTni chiqargan edi. Bu mahsulotlar ma'lumotlar bazasini boshqarishga mo'ljallangan tizimlarni standarti bo'lib qoldi. Bu mahsulotlar tarkibiga SQL ham kiritilgan edi. SQL ma'lumotlari qism tili standartiga aylandi. Boshqa MBBT ishlab chiquvchilar o'zining SQL versiyasini ishlab chiqishdi. Ularda IBM mahsulotini asosiy imkoniyatlarigina bo'limgan. O'zining MBBT ba'zi afzalligini ko'rsatish uchun ishlab chiquvchilar SQLning ba'zi kengaytmasini (qo'shimchasini) kiritdilar. Shu bilan birga hamma tan oladigan SQL standarti ustida ish olib borildi. 1986-yili Amerikaning milliy standartlar instituti (American National Standards Institute ANSI) rasmiy SQL – 86 standartini chiqardi. U 1989-yili yangilandi va yangi SQL – 89 nomini oldi. 1992-yili bu standart SQL – 92 (ISO, IEC 9075:1992) deb nomlandi. Oxirgi SQL standartining versiyasi SQL – 2003 bo'lib hisoblanadi. SQL aniq MBBT amalga oshirilgani biroz standartdan farq qiladi, uni ishlab chiqaruvchi firmalarning o'zi e'lon qiladi. Masalan, ko'p MBBT (masalan, Access 2003, Postgre SQL – 7,3) SQL – 92 ni to'laligicha ta'minlamaydi. Ular standartga kirmagan elementlarni ham o'z ichiga oladi. Lekin MBBTni ishlab chiquvchilar ko'proq standartga mos kelishiga harakat qiladilar.

SQL yaratilayotganda, oddiy, tabiiy tilga yaqin relatsion ma'lumotlar bazasi so'rovlар tili sifatida loyihalangan.

SQL tili deklarativ (tavsiflovchi) til hisoblanadi, shu bilan u dasturlash tillaridan farq qiladi. Buning ma'nosi SQL tilida be-

riladigan ifodalar nima qilish kerakligini tavsiflaydi, ya’ni uni qanday bajarish kerakligini ko’rsatmaydi.

Hozirgi kunda SQL tilining ikkita ko’rinishi mavjud va ishlatalidi:

- interaktiv SQL;
- qurilgan (kiritilgan) SQL.

Interaktiv SQL interaktiv rejimda foydalanuvchilarni SQL so’rovlarini berish va natijalarni olishda ishlataladi.

Qurilgan yoki kiritilgan SQL odatda birorta dasturlash tilida (C, C++ va boshq.) yozilgan dastur ichiga kiritilgan SQL komandalaridan tashkil topadi. Bu esa tilni ishlataladigan dasturlarni relatsion ma’lumotlar bazasidagi ma’lumotlar bilan ishlashida moslashuvchanlikni va samaradorligini ta’minlaydi. Lekin bunda kiritilgan tilda qo’shimcha interfeys vositalar talab etiladi.

SQL tilining tarkibiy qismlari. Interaktiv va qurilgan SQL quyidagi tarkibiy qismlarga ega:

- Ma’lumotlarni aniqlash tili – DDL (Data Definition Language). Bu til ma’lumot bazasining har xil obyektlarini (jadval, indekslar, foydalanuvchi va boshqalarini) yaratish, o’zgartirish va olib tashlash imkonini beradi. DDL kiruvchi operatorlarga misol sifatida CREATE TABLE (jadval yaratish), CREATE SCHEMA (sxema yaratish) va boshqalar kiradi.

- Ma’lumotlar bilan manipulatsiya qilish tili – DML (Data Manipulation Language) ma’lumotlar bazasi bilan ishlashni ta’minlashga mo’ljallangan til hisoblanadi. Ularga jadvaldan ma’lumotlarni tanlash (SELECT), qo’shish (INSERT), o’zgartirish (UPDATE), olib tashlash (DELETE) kabi komandalar kiradi.

- Ma’lumotlarni boshqarish tili – DCL (Data Control Language) bu til ma’lumotlar bazasini har xil o’rinsiz zararlanishlar dan saqlashga mo’ljallangan.

So’rovlar yaratishning ba’zi qoidalari: SQL tilida ifodalarda kalit so’zlar kichik va katta harflarda yozilishi mumkin. Ifodalar bitta yoki bir nechta satrda yozilishi mumkin. Ifoda yakunida «;» belgi qo’yiladi. Agar SQL komandasini bir necha bloki yozilsa, ular bir-biridan «;» bilan ajratiladi. Barcha SQLning kalit so’zлari (komandalari, operatorlari va boshqalar) faqat xizmatchi so’z sifatida ishlatalishi mumkin, boshqa maqsadlarda, ya’ni jadval nomi, ustun nomi, o’zgaruvchi va boshqalar sifatida ishlatish

mumkin emas. SQL operatori xizmatchi so'zlar va foydalanuvchi qo'llaydigan so'zlardan tashkil topadi.

Xizmatchi so'zlar SQL tilining doimiy qismi bo'lib, ular aniq qiymatga ega. Ularni standartda ko'rsatilgandek yozish kerak va ularning 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 obyektivlarining har xil nomlaridan iborat bo'ladi (jadval, ustun, tasvirlar, indekslar va h.k.). Operatordan so'zlar o'rnatilgan sintaksis qoidalariga moslab joylashtiriladi. Til standartida bu ko'rsatilmagan bo'lsa ham, SQL tilining dialektlarida (ko'rinishida) matn tugallanganligini bildiruvchi belgi, ko'pgina hollarda nuqtali vergul «;» ishlataladi.

SQL operator komponentalari ko'pchiligi registrga bog'liq emas, ya'ni ixtiyoriy har qanday katta va kichik harflar ishlatalishi mumkin. Bularda bitta istisno bor. Bu istisno simvolli literallarga tegishli. Ularda literal ma'lumotlar, ularga mos bo'lgan ma'lumotlar bazasidagi qiymatlar qanday saqlansa shunday yozilishi kerak. Masalan: agar ma'lumotlar bazasidagi familiyaning qiymati «SWITH»_ko'rinishida bo'lsa, qidirish shartida «swith» simvol literal ko'rinishda berilsa, bunga tegishli yozuv hech qachon topilmaydi.

6.2. SQLda ma'lumot toifalari

SQL tilida har bir atributga uni qiymatiga mos ma'lumot toifalarini ko'rsatish mumkin. SQL – 2003 da quyidagi umumiy toifalar aniqlangan:

Simvolli satr toifasi;

CHARACTER [(uzunligi)] yoki CHAR [(uzunligi)].

Jadvalning matnli qiymatlari maydonlari CHAR toifasi bilan aniqlanadi. Bunday maydonlar fiksirlangan uzunlikka ega bo'lib, uzunligi «uzunlik» parametri bilan aniqlanadi. Bu parametrning qiymati 1 dan 255 gacha bo'ladi, ya'ni 255 tagacha simvolni joylashtirish mumkin.

Ma'lumot qidirilayotganda simvollar satrining uzunligi berilgan uzunlikdan kichik bo'lsa, maydon o'ng tomondan probellar bilan to'ldiriladi. SQL standartida o'zgaruvchan uzunlikdagi ma'lumotlar toifalari bor. Bu toifa VARCHAR (), CHARACTER VARYING yoki CHAR VARYING() toifadagi konstantalar apostrof ichiga olinib yoziladi.

SQL tilida standart sonli toifalarga quyidagilar kiradi:

- INTEGER -2^{31} dan $+2^{31}$ gacha diapazondagi butun sonlarni tasvirlashda ishlataladi;
- SMALLINT -2^{15} dan $+2^{15}$ gacha diapazondagi butun sonlarni tasvirlashda ishlataladi;
- DECIMAL (aniqlik, [masshtab]) – fiksirlangan nuqtali o‘nli son. Aniqlik sondagi qiymatli raqamlar sonini aniqlaydi. Masshtab nuqtadan o‘ng tomondagi raqamlarning maksimal sonini bildiradi;
- NUMERIC (aniqlik, [masshtab]) – DECIMAL kabi fiksirlangan nuqtali o‘nli sonni aniqlaydi;
- FLOAT [(aniqlik)] – suzuvchi nuqtali son, minimal aniqlik bilan ko‘rsatilgan;
- DOUBLE PRECISION – REAL toifadagi son kabi son. Lekin REALga nisbatan aniqligi ikki marta yuqori;
- ORACLE MBBTda qo‘sishimcha NUMBER ma’lumot toifasi kiritilgan. Uning yordamida barcha sonli ma’lumot toifalari, butun, fiksirlangan nuqtali va suzuvchi nuqtali sonlar tasvirlanadi. Uning formati quyidagicha:

NUMBER [(aniqlik,[masshtab])]

Agar aniqlik parametrining qiymati oshkor ravishda ko‘rsatilmasa, u 38 ga teng deb olinadi. Masshtab parametrining sukut bilan olingan qiymati 0 ga teng. Aniqlik parametrining qiymati 1 dan 38 gacha o‘zgaradi. Masshtab parametri qiymati 84 dan 128 gacha o‘zgaradi. Masshtabning manfiy qiymatlarini ishlatalish o‘nli nuqtani yuqori razryadlar tomon siljishiga olib keladi.

DECIMAL va NUMERIC toifalar NUMBER toifaga ekvivalent. Formati:

DECIMAL [(aniqlik,[masshtab])];
NUMERIC [(aniqlik,[masshtab])];
NUMBER [(aniqlik,[masshtab])].

Sana va vaqt toifasi. Sana va vaqt ma’lumotlarini tasvirlash toifalari nostandard hisoblanadi. ORACLE MBBTda sana va vaqt toifalarini tasvirlash uchun DATE toifasi kiritilgan. DATE toifasining kiritilishi sana va vaqt ma’lumotlari ustida arifmetik amallar bajarilishi imkonini beradi. DATE toifasidagi o‘zgaruvchiga butun sonni qo‘sish sanani shu songa teng kunlar soniga oshiradi. DATE toifasidagi konstantalar Operatsion tizimlarda kelishilgan formatlar asosida yoziladi. Masalan:

'03.05.2004' yoki '12/06 2004' yoki '03-nov-2004' yoki '03-apr-2004'.

Noaniq va o'tkazib yuborilgan ma'lumotlar. SQLda atribut qiymatlari noma'lum bo'lgan, o'tkazib yuborilgan yoki mavjud bo'limganlarini NULL bilan yoziladi. NULL qiymat oddiy tu-shunchada qiymat hisoblanmaydi. U faqat atributning haqiqiy qiymati tushirib qoldirilgan yoki noma'lumligini anglatadi. NULL ni ishlatalishda quyidagilarga e'tibor berish kerak:

Agregat funksiyalar ishlatilganida birorta atributni qiymatlari to'plami bo'yicha hisoblashlar bajarilganida aniqlikni ta'minlash maqsadida NULL qiymat hisobga olinmaydi.

- Shartli operatorlarda TRUE, FALSEdan tashqari UNKNOWN paydo bo'lsa, natija NULL qiymatda chiqadi.

- Bu qiymatni tekshirish uchun IS NULL yoki IS NOT NULLlardan foydalanamiz.

- Almashtirish funksiyalari argument NULL bo'lsa natija NULLga teng bo'ladi.

- Agregat funksiyalar yordamida, masalan, yig'indini hisoblash, amallarni bajarishda NULL hisobga olinmaydi.

Darslikda keltirilgan SQL so'rovlarni tuzish bilan bog'liq misollarda quyida keltirilgan jadvallardan tashkil topgan ma'lumotlar bazasidan foydalanildi.

SQL tilini konstuksiyalarida [] belgi (kvadrat qavslar) sintaksik konstruksiyani zarur bo'limgan element ko'rsatadi.

STUDENT

Student ID	SUR-NAME	NAME	STI-PEND	KURS	CITY	BIRTH-DAY	UNIV-ID
1	Ivanov	Ivan	150	1	Orel	3.12.92	10
3	Petrov	Pyotr	200	3	Kursk	1.12.90	10
6	Sidirov	Vadim	150	4	Moskva	7.06.89	28
10	Kuz-netsov	Baris	0	2	Brenek	08.12.91	10
12	Zayseva	Olga	25	2	Lipetsk	01.05.91	10
265	Pavlov	Andrey	0	3	Voronej	05.11.89	10
32	Kotov	Pavel	150	5	Belgorod	NULL	14
654	Lukin	Artyom	200	3	Voronej	01.12.91	10

276	Petrov	Anton	20	4	Null	05.08.91	22
55	Belkin	Vadim	50	5	Voronej	07.01.90	10

Student ID – talabani identifikatsiyalovchi sonli kod

SURNAME – talaba familiyasi

NAME – talaba ismi

STIPEND – talaba oladigan stipendiya

KURS – talaba o‘qiydigan kurs

CITY – talaba tug‘ilgan shahar

BIRTHDAY – talaba tug‘ilgan sana

UNIV ID – talaba o‘qiydigan universitetni identifikatsiyalovchi kod

LECTURER

LECTURER_ID	SURNAME	NAME	CITY	UNIV_ID
24	Kosmakov	Boris	Voronej	10
46	Nikonov	Ivan	Voronej	10
74	Lagutin	Pavel	Moskva	22
108	Otrukov	Nikolay	Moskva	22
276	Nikolayev	Viktor	Voronej	10
378	Sorokin	Andrey	Orel	10

LECTURER – o‘qituvchini identifikatsiyalovchi kod

SURNAME – talaba familiyasi

NAME – talaba ismi

CITY – talaba tug‘ilgan shahar

UNIV_ID – talaba o‘qiydigan universitetni identifikatsiyalovchi kod

SUBJECT

SUBJECT_ID	SUBJ-NAME	HOUR	SEMESTR
10	Informatika	56	1
22	Fizika	34	1
43	Matematika	56	2
56	Tarix	34	4
94	Ingлиз тили	56	3
73	J/tarbiya	34	5

SUBJECT_ID – o‘rgatiladigan fan identifikatori

SUBJ-NAME – o‘qitiladigan fan nomi

HOUR – fanni o‘qitish uchun ajratilgan soatlar

SEMESTR – fan o‘qitiladigan semestr

UNIVERSITY

UNIV ID	UNIV NAME	RATING	CITY
22	Mry	606	Moskva
10	Bry	296	Voronej
11	Hgy	345	Novosibirsk
32	Pry	416	Rostov
14	Bry	326	Belgorod
15	Try	368	Tomsk
18	Brma	327	Voronej

UNIV ID – talaba o‘qiydigan universitetni identifikasiyalovchi kod

UNIV NAME – universitet nomi

RATING – universitet reytingi

CITY – universitet joylashgan shahar

EXAM_MARKS

Exam_ID	Student ID	Subject id	Mark	Exam date
145	12	10	5	12.01.09
34	32	10	4	23.01.09
75	55	10	5	25.01.09
238	12	22	3	17.06.08
639	55	22	Null	22.06.08
43	6	22	4	18.01.09

Exam_ID – imtihon identifikatori

Student ID – talaba identifikatori

Subject id – o‘qitiladigan fan identifikatori

Mark – imtihon bahosi

Exam date – imtihon sanasi

SUBJ_LECT (o'qituvchilarning o'quv fanlari)

LECTURER_ID	SUBJECT_ID
24	10
46	22
74	43
108	56
276	94
328	73

LECTURER_ID – o'qituvchi identifikatori

SUBJECT_ID – o'qitiladigan fan identifikatori

6.3. Oddiy SELECT so'rovlar

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

SELECT operatori sodda holda quyidagi ko'rinishda bo'ladi:

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<OM SELECT operatorli ifoda>];

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

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

Masalan:

SELECT NAME, SURNAME FROM STUDENT;

Ixtiyoriy SQL so'rov operatori nuqta vergul «;» simvoli bilan tugaydi.

Keltirilgan so'rov STUDENT jadvalidan NAME va SURNAME maydonlarning barcha qiymatlarini ajratib olishni amalga oshiradi.

6.4. So‘rovlarda maxsus operatorlarni ishlatalish

All (barchasi) xizmatchi so‘zi distinct nisbatan teskari ta’sir qiladi, ya’ni natijaga takrorlanuvchi ma’lumotlarni qo’shib chiqaradi. SELECT operatorida shartlarni yozish uchun Where (qayerda) kalit so‘zi ishlataladi. Bu so‘z bilan beriladigan shart yordamida jadvaldan qaysi satrlar tanlab olinishi kerakligi ko’rsatiladi. So‘rov natijasidan iborat jadvalga Where da ko’rsatilgan shartni qanoatlantiruvchi (rost) satrlargina kiritiladi. Masalan: Petrov familiyali talabaning familiya va ismini chiqarish uchun quyidagi so‘rov yoziladi:

Select SURNAME, NAME From student

Where SURNAME ='Petrov'

Shartda ishlataladigan amallar. Shartda solishtirish va mantiqiy amallar ishlatalishi mumkin: Solishtirish amallari = (teng), <>(teng emas), >=(katta yoki teng), <=(kichik yoki teng), <(kichik), >(katta).

Mantiqiy amallar: AND, OR, NOT

Masalan: 3-kursda o‘qiydigan va stipendiya oladigan tala-balarning ismi va familiyasini chiqarish so‘rovini tuzing:

Select NAME, SURNAME From STUDENT

Where Kurs=3 AND STIPEND>0;

IN, BETWEEN, LIKE, IS NULL operatorlari. WHERE parametrida mantiqiy shartlarni berishda IN, BETWEEN, LIKE, IS NULL parametrлarni ishlatalish mumkin. IN operatori (ro‘yxatdagi ixtiyoriy elementga teng) va NOT IN (ro‘yxatdagi birorta ham elementga teng emas) tekshirilayotgan maydon qiymatini berilgan ro‘yxat bilan solishtirish uchun ishlataladi. Bu qiymatlar ro‘yxati IN operatorining o‘ng tomonida yoziladi. IN operatori ishlatib qurilgan shart rost deb hisoblanadi, agarda IN chap tomonida yozilgan maydon qiymati IN dan o‘ng tomonda qavs ichida yozilgan maydon qiymatlaridan biri bilan bir xil bo‘lsa.

NOT IN parametr qo’llab tuzilgan shart rost bo‘ladi, agarda NOT IN parametrda chap tomonda ko’rsatilgan maydon qiymatlaridan birortasi ham, NOT IN parametrining o‘ng tomonida qavs ichida keltirilgan ro‘yxatda berilgan birorta ham qiymatga mos kelmasa.

Masalan, EXAM_MARKS jadvalidan imtihonda faqat «4» va «5» baho olgan talabalar haqida ma’lumot chiqaring:

SELECT *

**FROM EXAM_MARKS
WHERE MARK IN (4,5);**

Masalan, imtihonda birorta ham «4» va «5» baho olmagan talabalarining ro'yxatini chiqarish uchun:

**SELECT *
FROM EXAM_MARKS
WHERE MARK NOT IN (4,5);**

BEETWEEN operatori. BEETWEEN operatori maydon qiymati berilgan intervalga joylashishi (kirishishi) shartini tekshirishda ishlataladi, ya'ni bunda atribut qiymatlari ro'yxati (IN parametridagi) o'rniiga, ularning o'zgarish chegarasi beriladi. Masalan, o'qitish uchun «30» va «40» soat oralig'ida vaqt ajratilgan, fanlar (predmetlar) haqidagi yozuvlarni chiqarish uchun so'rov ko'rinishi quyidagicha bo'ladi:

**SELECT *
FROM SUBJEKT
WHERE HOUR BETWEEN 30 AND 40;**

Chegaraviy qiymatlar (30 va 40 qiymatlari), solishtirilayotgan qiymatlar to'plamiga kiradi. BEETWEEN operatorini (parametri) sonli maydon va simvolli maydon toifalarida ham qo'llash mumkin.

LIKE operatori. LIKE operatorini faqat simvolli toifalar maydonlariga (CHAR yoki VARCHAR) qo'llash mumkin. Bu operator tekshirilayotgan satrli maydon qiymatiga, LIKE operatorida berilgan qism satrga (qidirish namunasi) kirishi (uchrashi) tekshiriladi. Berilgan qism satrini namunasi bo'yicha satrli qiymatlarni tanlash uchun qidirilayotgan satr namunasining shablonidan foydalanish mumkin. Bunda shablonda quyidagi simvoldan foydalanish mumkin.

Shablonda ishlataligan «_» belgi, shu belgi joylashgan yerda ixtiyoriy bitta simvol ishlatalishi mumkinligini anglatadi.

Shablonda ishlataligan «%» belgi, shu belgi joylashgan yerda ixtiyoriy uzunlikdagi ixtiyoriy simvollar ketma-ketligini ishlatalish mumkinligini anglatadi. Masalan, STUDENT jadvalidan familiyasi P harfi bilan boshlanadigan talabalar haqidagi axborotlarni chiqarish so'rovi quyidagicha bo'ladi:

**SELECT * FROM STUDENT
WHERE SURNAME LIKE 'P%';**

6.5. Chiqarilayotgan ma'lumotlarni o'zgartirish

SQL tilida ustun qiymatlari yoki ifodadagi konstantalari bilan ishlashga mo'ljallangan ma'lumotlarni o'zgartirish operatorlari va kiritilgan funksiyalar bilan ishslash amalga oshirilgan. Unda ustun nomi, konstanta va kiritilgan funksiyalardan tashkil topgan ifoda qiymatlaridan foydalanish imkoniyati bor. Bunda quyidagi amallar ishlatalidi: "-", "+", "*", "/"

Misol: select surname, name, stipend, kurs, (stipend*kurs)/2;
FROM student

WHERE kurs= 4 and stipend>0;

So'rov natijasi quyidagi ko'rinishda bo'ladi:

Surname	Name	Stipend	Kurs	
Sidirov	Vadim	150	4	300
Petrov	Anton	200	4	400

SQL tilida ma'lumotlarni almashtiruvchi va kiritilgan familiyalar ishlatalishi mumkin. Ular ustun qiymatlari bilan ishlashni yoki *const* sifati ifodalarda keladi. *Const* sifatida simvolli *const*, sonli *const* larni ishlatalish mumkin. Ular ustunlar ro'yxtigiga kiritiladi va xuddi virtual ustun kabi aniqlanadi. Agar so'rovda ustun o'rnida son kelsa, bu sonli *const* hisoblanadi. Simvolli *const* (' ') ichida yoziladi.

Misol: quyidagi so'rov ushbu jadvalni chaqiradi:

Select 'familiya', surname, 'ism', name, 100

From student WHERE kurs= 4 and stipend>0;

	Surname		Name	
familiya	Ivanov	ism	Ivan	100
familiya	Petrov	ism	Petr	100

Satrlarni ulash amali continct 2 ta va undan ko'p simvolli ustun qiymatlari bitta satrga joylashtirib boriladi.

Select surname ||'-'|| name, stipend

Where kurs = 4 and stipend>0;

	stipend
Sidirov_Vadim	150
Petrov_Anton	200

6.6. SQL tilining standart funksiyalari

Misol sifatida standart funksiyalari deganda quyidagilarni ko'ramiz:

Lower (<satr>) – berilgan satrni katta harflarga kichik harflarga almashtirib beradi;

Upper (<satr>) – kichik harflarni katta harflarga almashtirib beradi;

Init cap (<satr>) – satrdagi har bir so'zni 1- harfini bosh harf qilib beradi. Masalan, ularda quyidagi misolni ko'ramiz:

Surname	Name
Sidirov	Vadim
Petrov	Anton

Agregat funksiyalar yordamida, masalan, yig'indini hisoblash, amallarni bajarishda NULL hisobga olinmaydi.

Select lower (surname), Upper (name)

from student Where kurs=4 and

Stipend>0;

1. LPAD (<satr>, uzunlik, [<qism satr>]) – satrni chapdan to'ldirish.

2. RPAD (<satr>,<uzunlik>,[<qism satr>]); – satrni o'ngdan to'ldirish.

3. Sart berilgan uzunlikgacha qism satr bilan chapdan o'ngga to'ldiriladi.

Agar qism satr ko'rsatilmagan bo'lsa, satr sukut bilan, probel-lar bilan to'ldiriladi. Agar uzunlik satr uzunlikdan kichik bo'lsa berilgan satr ko'rsatilgan uzunlikkacha to'ldiriladi (balki, ketma-kebetlikni takrorlash bilan) qirqiladi:

LTRIM (<satr>,[<qism satr>]);

RTRIM (<satr>,[<qism satr>]).

Bu funksiyalarning vazifasi mos ravishda chapdag'i (o'ngdag'i) chegaraviy simvolni olib tashlashdan iborat. Olib tashlangan sim-vollar qism satrda ko'rsatiladi. Agar qism satr ishlamasra, probel-lari olib tashlanadi.

Substr (<satr>,<boshlanish>,[<soni>])

4. Quyida beriladigan funksiyalar satrdagi berilgan pozitsiyadan boshlab berilgan sondagicha simvollarni ajratib oladi. Agar soni ko'rsatilmagan bo'lsa, satrni boshidan oxirigacha ajratib ola-di.

Misol: **substr** (hurmatli do'stim: 10,6) ; => do'stim

5. Length (<satr>) vazifasi satrning uzunligini aniqlab borish-dan iborat.

Select 1pad (surname, 10, @)

Rpad (name, 10,\$) **from** student

Where kurs=3 and stipend>0;

So'rov natijasi quyidagi jadvaldan iborat bo'ladi:

@@@Petrov	Petr \$\$\$\$\$
@@@Pavlov	Andrey\$\$\$\$
@@@Lukin	Artyom \$\$\$

6. Select substr (name, 1,1) ||':|| Surname, city, length(city)

Where kurs in(2,3,4) **and** stipend>0;

So'rov natijasi quyidagi jadvaldan iborat bo'ladi:

	City	
A. Petrov	Kursk	5
S. Sidirov	Moskva	6
O. Zayseva	Lipetsk	7
A. Lukin	Voronej	7
A. Petrov	Null	Null

Misol:

Select surname, name, brithday;

To_char (birthday, DD,MM,YY)

From student;

So'rov natijasi quyidagi jadvaldan iborat bo'ladi:

Surname	Name	Birthday	
Ivanov	Ivan	3/11/1992	3.12.92

Ma'lumotlar bilan manipulatsiyalashli (agregat) funksiyalar.

Ma'lumotlar bilan manipulatsiyalashli funksiyalar jadvaldan yig'ilgan axborotlarni olish uchun xizmat qiladi. Bu funksiyalar jadvaldagi satrlar 6.9. Ma'lumotlar bilan manipulatsiyalashli funksiyalarda amal bajarib, 1 ta natija chiqaradi. 6.9. Ma'lumotlar bilan manipulatsiyalashli funksiyalar uchun quyidagi amallarni ishlatalamiz:

1. **Count** – jadvaldagi satrlar sonini aniqlab beradi.
2. **Sum** – ko'rsatilgan maydon qiymatlari yig'indisini hisoblaydi.
3. **AVG** – tanlab olingan maydon qiymatlarini o'rta arifmetigini hisoblaydi.
4. **MAX (MIN)** – tanlab olingan maydon qiymatlarning eng kattasini (kichigini) olib beradi.

Select so'rovida 6.9. Ma'lumotlar bilan manipulatsiyalashli funksiyalar maydon nomlari kabi ishlataladi. Maydon nomlari funksiyalar argumentlari sifatida keladi.

Misol:

1. Exam_marks jadvaldagi mark maydonning o'rtacha qiymatini hisoblash: select average (mark) from exam_marks;
2. Jadvaldagi satrlar (yoki yozuvlar) sonini hisoblash uchun quyidagi komandalarda foydalanamiz:

Select count (*) From exam_marks;

3. Select count (**distinct** aubj-ID); from subject;

Select komandasida **group by** parametr ham ishlalistish mumkin.

Bu parametr maydonni o'xshash parametrlari (aniqlanayotgan qiymat) bo'yicha 6.9. Ma'lumotlar bilan manipulatsiyalaydi va agregat funksiyalar ishlatilsa, ular shu 6.9. Ma'lumotlar bilan manipulatsiyalashga tegishli bo'ladi.

Misol: **select student_ID, Max(mark) from exam_marks Group by student_ID;**

6.9. Ma'lumotlar bilan manipulatsiyalashni bir necha maydon bo'yicha ham bajarish mumkin:

Select student_ID, subject_ID Max(mark) From exam_marks group by Student_ID, subject_ID;

6.9. Ma'lumotlar bilan manipulatsiyalashlar ichidan kerakli yozuvni ajratib olish uchun **having** ishlataladi:

Having faqat 6.9. Ma'lumotlar bilan manipulatsiyalashlar ichiga tegishli.

Select subj_name, max(hour) From subject Group by subj_name

Having max(hour)>=34;

Ba'zi hollarda jadvaldagi ma'lumotlarni tartiblash talab etiladi. Buning uchun **Order by parametric** ishlataladi. Bu parametr ko'rsatilgan maydonning barcha yozuvlarini o'sib borish tartibida saralaydi. **Order by desc** esa kamayish tartibida saralaydi. **Order by (ASC)** bo'lsa, o'sish tartibida saralaydi.

Misol:

Select * subject Order by subj_name;

Select* from subject Order by subj_name desc;

Tartiblash bir nechta maydon bo'yicha bo'lishi ham mumkin. Bunda avval tartiblash 1- maydon bo'yicha, keyin 2- maydon bo'yicha bajariladi.

Shuningdek, Order by parametri Group by parametri bilan ham ishlatalishi mumkin. Bunda Order by so'rov oxiri keladi va unda 6.9. Ma'lumotlar bilan manipulatsiyalashni ichidagi yozuvlarni tartiblaydi. Misollar:

Select * from subject Order by semester, subj_name;

Select subject, semester, subj_name Order by semester;

6.7. Bog'langan qism so'rovlar yaratish

SQL tili bitta so'rov ichiga 2-so'rovni joylashtirib ishlatalish imkonini beradi. Misol: Faraz qilaylik, talabaning «Petrov» familiyasi ma'lum bo'lsin, lekin uning STUDENT_ID maydoni qiymati noma'lum. Shu talabaning barcha baholari haqidagi axborotni (ma'lumotlarni) olish uchun quyidagi so'rovni tuzamiz:

Select * from exam_marks

Where student_ID = (select student_ID

From student where surname= 'Petrov') ;

Bog'langan qism so'rovli SQL so'rov quyidagicha ishlaydi:

1. Tashqi so'rovda nomi ko'rsatilgan jadvaldan satr tanlab olinadi.

2. Qism so'rov bajariladi va olingan qiymat shu satrni tahlil qilish uchun tashqi so'rovni where shartida ishlataladi.

3. Bu shartni baholash natijasi bo'yicha satrni chiqarilayotgan ma'lumotlar tarkibiga kiritish yoki kiritmaslik haqida qaror qabil qilinadi.

4. Sanab o‘tilgan qadamlar tashqi so‘rovni keyingi satri uchun qaytariladi.

Bu so‘rov faqat qism so‘rov bitta qiymat chiqarganda (ya’ni Petrov familiyasi jadvalda bitta bo‘lganda) to‘g‘ri ishlaydi. Ko‘p hollarda buning uchun ichki so‘rovdan yagona yechim olish uchun DISTINCT parametridan foydalananamiz. Qism so‘rovlarda IN operatori keng ishlataladi.

VORONEJ shahridan bo‘lgan talabalarning barcha baholari (EXAM_MARKS) haqidagi ma’lumotlarni quyidagi so‘rov yordamida olish mumkin.

```
SELECT * FROM EXEM_MARKS  
(WHERE STUDENT_ID IN ( SELECT STUDENT_ID  
FROM STUDENT
```

WHERE CITY='Voronej');

Bog‘langan qism so‘rovlarni yaratishga misollar:

20-yanvar 1999-yil imtihon o‘tadi, o‘qitilgan barcha predmetlar haqida axborotni chiqaring.

```
SELECT*FROM SUBJECT SU WHERE '20/01/1999' IN  
(SUBJECT EXAM_DATE FROM EXAM_MARKS_EX  
WHERE SU.SUBJ_ID =EX.SUBJ_ID);
```

Bu masalani ham jadvallarni birlashtirish amali yordamida bajarish mumkin.

```
SELECT DISTINCT SU.SUBJ_ID, SUBJ_NAME, HOUR,  
SEECTER  
FROM SUBJECT_FIRST, EXAM_MARKS_SECOND  
WHERE FIRST.SUBJ_ID = SECOND.SUBJ_ID  
AND SECOND.EXAM_DATE='20/01/1999';
```

EXITS operatorini ishlatalish. SQL so‘rov operatorida ishlataligan EXISTS parametri, mantiqiy ifoda kabi rost va yolg‘on qiymatlarni qabul qiladi. Bu operator argument sifatida qism so‘rovlarni ishlatab, bu operator qism so‘rovlarni bajarilishini rost deb hisoblaydi, agarda qism so‘rov chiquvchi ma’lumot bersa, ya’ni birorta topilgan qiymat chiqarsa, aks holda qism so‘rov natijasi yolg‘on hisoblanadi. EXISTS operatori UNKNOWN (noma’lum) qiymatini qabul qiladi.

Masalan:

EXAM_MARKS jadvalidan hech bo‘lmaganda bitta qoniqarsiz baho olgan talaba haqida ma’lumot chiqarish talab etilsa,

```
SELECT DISTINCT STUDENT_ID
```

```
FROM EXAM_MARKS A WHERE EXISTS  
(SELECT * FROM EXAM_MARKS B  
WHERE MARK<3 AND  
B.STUDENT_ID=A.STUDENT_ID);
```

6.8. Birlashtirish operatori

Birlashtirish operatori UNION ikkita yoki undan ortiq SQL so‘rovlarga chiqaradigan natijalarni yagona satrlar va ustunlar to‘plamiga birlashtiradi. Masalan, bitta jadvalda MOSKVAda yashaydigan talaba va o‘qituvchilarning identifikatorlarini bitta jadvalda olish uchun quyidagicha so‘rov ishlataladi:

```
SELECT 'STUDENT___', SURNAME, STUDENT_ID  
FROM STUDENT  
WHERE CITY=' MOSKVA'  
UNION  
SELECT 'O‘QITUVCHI ', SURNAME,  
LECTURER_ID FROM LECTURER  
WHERE CITY=' MOSKVA';
```

UNION takrorlanishlarni yo‘qotish. Oddiy so‘rovlardan farqli ravishda UNION avtomatik ravishda chiqarilayotgan ma’lumotlardan takrorlanuvchi satrlarni olib tashlaydi.

```
SELECT CITY FROM STUDENT  
UNION  
SELECT CITY FROM LECTURER;
```

Bir xil shahar nomi qaytarilmaydi, olib tashlanadi. Agar har bir so‘rovdagi barcha satrlarni boshqa birlashtiruvchi so‘rovarda bo‘lishidan qat’i nazar chiqarish talab etilsa UNION ALL konstruksiyasini ishlatalish kerak.

```
SELECT CITY FROM STUDENT  
UNION ALL  
SELECT CITY FROM LECTURER;  
Takrorlanishni bo‘ladi.
```

Jadvallarni INNER JOIN operatori yordamida ulash. Agar SELECT operatorida FROM so‘zidan keyin bitta jadval emas, ikkita jadval ishlatsa, unda so‘rov natijasi (WHERE so‘zi ko‘rsatilmasa) bitta jadvalning har bir satri ikkinchi jadvalning har bir satri bilan ulanadi. Bu ba’zan dekart ko‘paytma yoki jadvallarni birlashtirish deyiladi. WHERE ishlatsa jadval ixchamlanadi. SQL bir nechta jadvallardan ma’lumot chiqarishi mumkin. Masalan:

talabalar familiyasini (STUDENT jadvalidan) va har bir student uchun talabalar yashaydigan shahar, universitet nomini (UNIVERSITY jadvalidan) chiqarish kerak bo‘ladi, unda barcha talabalar va universitetlar ikkala jadvaldagi CITY maydon qiymatlari bir xil bo‘lgan yozuvlar kombinatsiyasini olish kerak. Buning uchun so‘rov

SELECT STUDENT.SURNAME, UNIVERSITY. UNIV_NAMES,

STUDENT.CITY FROM STUDENT, UNIVERSITY

WHERE STUDENT.CITY=UNIVERSITY.CITY;

Tenglikka asoslangan predikatlarni ishlatib birlashtirish ekvibirlashtirish deyiladi. Misolda ko‘rib chiqilgan jadvallarni birlashtirish ichki (INNER) birlashtirish deyiladi. Bunday birlashtirilishda bajarilayotgan so‘rovni ON so‘zida berilgan sharti rost bo‘lgan jadvallar satrlari olinadi. Yuqoridagi so‘rov, ya’ni talaba (STUDENT jadvali) va u yashaydigan shaharlar, universitet nomlari (UNIVERSITY jadvali) bilan chiqarish talab etilsa, JOIN kalit so‘zini ishlatib quyidagicha tashkil etiladi:

SELECT STUDENT.SURNAME,UNIVERSITY.UNIV_NAME, STUDENT.CITY

FROM STUDENT INNER JOIN UNIVERSITY. ON STUDENT.CITY=UNIVERSITY.CITY;

6.9. Ma’lumotlar bilan manipulatsiyalash

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‘sish), UPDATE (tiklash, yangilash), DELETE (olib tashlash, yo‘qotish) komandalaridir.

INSERT komandasini (buyrug‘i) jadvalga yangi satr qo‘sishni 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 ishlataladi:

Insert into Student (Student_ID, CITY, SURNAME, NAME)

Values (101, 'Moskva', 'Ivanov', 'Sasha');

Qavs ichidagi ro'yxatda nomi keltirilmagan ustunlarga avtomatik ravishda sukut bilan jadval tavsiflashda (**CREATE TABLE** komandasida) tayinlangan qiymat yoki NULL qiymat tayinlanadi.

INSERT komandasasi yordamida, bir jadvaldan qiymat tanlab olib uni boshqa jadvalga joylashtirish mumkin. Masalan: quyidagi so'rov

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

Bunda STUDENT1 jadvali **CREATE TABLE** komandasasi yordamida yaratilgan bo'lishi kerak va STUDENT jadvali strukturasiga o'xhash bo'lishi kerak.

Jadvaldagi satrlarni olib tashlash uchun **DELETE** komandasini ishlataladi. 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** komandasini bilan olib tashlash mumkin).

Jadval bir yo'la birorta shartni qanoatlantiradigan bir nechta satrni olib tashlash uchun WHERE parametridan foydalanish mumkin. Masalan:

DELETE FROM EXAM_MARKS

WHERE STUDENT_ID=103;

Satrlar guruhini olib tashlash ham mumkin:

DELETE FROM STUDENT1

WHERE CITY= 'Moskva';

UPDATE komandasini jadval satrlari yoki mavjud satrni ba'zi bir yoki barcha maydonlarining qiymatini tiklash, ya'ni o'zgartirish imkonini beradi. Masalan, UNIVERSITY1 jadvalidagi barcha universitetlarning reytingini 200 qiymatga o'zgartirish uchun quyidagi konstruksiyani ishlatish mumkin:

UPDATE UNIVERSITY1

SET RATING=200

WHERE CITY= 'Moskva';

Bu so'rov bajarilganda, faqat Moskvada joylashgan universitetlarning reytingi o'zgartiriladi.

UPDATE komandasini 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. Masalan: o'qitilayotgan fan nomi «MATEMATIKA» (uning uchun SUBJ_ID=43) «Oliy matematika» nomiga o'zgartirish talab etilsa va bunday indentifikatsion nomerni saqlab o'zgarishsiz qoldirish kerak bo'lsa, lekin shu bilan birga jadvaldagi mos satr maydonlariga o'qitiladigan fan haqida yangi ma'lumotlar kiritish uchun so'rov quyidagi ko'rinishda yoziladi:

UPDATE SUBJECT1

**SET SUBJ_NAME= 'Oliy matematika', HOUR=36,
SEMESTER=1**

WHERE SUBJ_ID=43;

UPDATE komandasining SET parametrida skalar ifodalarni ishlatish mumkin. Skalar ifodada maydonga qiymatning o'zgarish usulini ko'rsatadi:

UPDATE UNIVERSITY1

SET RATING=RATING*2;

Masalan: STUDENT1 jadvaldagi STIPEND maydon qiymatini Moskva shahri talabalari uchun 2 marta oshirish uchun quyidagi so'rov ishlatiladi:

UPDATE STUDENT1

SET STIPEND=STIPEND *2

WHERE CITY= 'Moskva';

SET predikat hisoblanmaydi, shuning uchun unda NULL qiymatni ko'rsatish mumkin:

UPDATE University1

```
SET RAYTING=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, INSERTli 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. Masalan: faraz qilaylik STUDENT1 jadvalining strukturasi (tuzilmasi), STUDENT jadval strukturasiga to'la mos bo'lsin. STUDENT jadvalidan Moskva shahri talabalari barchasi haqida yozuvlari bilan STUDENT1 jadvali to'ldirish imkonini beradigan so'rov ko'rinishi quyidagicha bo'ladi:

```
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');
```

6.10. Ma'lumotlar bazasi obyektlarini yaratish

Ma'lumotlar bazasi obyektlarini yaratish ma'lumotlarni aniqlash tili (DLL) operatorlari yordamida bajariladi. Ma'lumotlar bazasi jadvallari yaratish **CREATE TABLE** komandasini yordamida amalga oshiriladi. Bu komanda bo'sh jadval yaratadi, ya'ni jadvalda satrlar bo'lmaydi. Bu jadvalga qiymatlar **INSERT** komandasini yordamida kiritiladi. **CREATE TABLE** komandasini 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‘lmasganda 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);

Smallint;

Dec(detcimal);

Number ;

Float;

.....

Albatta ko‘rsatilishi zarur bo‘lgan ma’lumot toifasi – CHAR. Maydonga yozilgan real simvollar soni noldan (agar maydonda NULL qiymati bo‘lsa) **CREATE TABLE** berilgan maksimal qiymatgacha bo‘ladi. Masalan, 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),
BRITHDAY 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. Indekslar 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.

Masalan, agar EXAM_MARKS jadvalidan talabaning STUDENI_ID maydoni qiymati bo'yicha bahosini qidirish tez talab etilsa, unda shu maydon bo'yicha indeks bajariladi.

CREATE INDEX STUDENT_ID_1 ON EXAM_MARKS (STUDENT_ID);

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

DROP INDEX<INDEKS NOMI>;

Masalan, **DROP INDEX<STUDENT_ID_1>;**

Mavjud jadval strukturasini va parametrlarini modifikatsiyalash uchun **ALTER TABLE** komandasi ishlataladi. 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 tavslifini 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 kattalash-tirishi mumkin;

- NOT NULL o'rnatilishi uchun ustunda birorta ham NULL qiymat bo'imasligi kerak. NOT NULLni har doim bekor qilish mumkin;

- sukut bilan o'rnatilgan qiymatni har doim o'zgartirish mumkin.

Mavjud jadvalni olib tashlash uchun, oldin bu jadvaldan bar-cha ma'lumotlarni olib tashlash kerak, ya'ni uni bo'shatish kerak.

Sartlarga ega bo‘lgan jadvalni olib tashlab bo‘lmaydi. Bo‘sh jadvallarni olib tashlash quyidagi komanda bilan bajariladi:

```
DROP TABLE <jadval nomi>;
```

6.11. Cheklanishlar

Mumkin bo‘lgan ma’lumot qiymatlariga cheklanishlar bo‘lishi mumkin. Unda

CREATE TABLE komandasini quyidagicha bo‘ladi:

```
CREATE TABLE <jadval nomi>
```

```
(<ustun nomi><ma’lumot toifasi><ustunga cheklanishlar>,
<ustun nomi><ma’lumot toifasi><ustunga cheklanishlar>,
<jadvalga cheklanishlar>(<ustun nomi>[,<ustun nomi>]));
```

Masalan, **NONE** qiymatning STUDENT jadvalini aniqlashda jadvaldagi STUDENT_ID, SURNAME, NAME maydonlarida ishlatalishni taqiqlash uchun komanda quyidagicha bo‘ladi:

```
CREATE TABLE STUDENT
(STUDENT_ID INTEGER NOT NULL,
SURNAME CHAR(25) NOT NULL,
NAME CHAR(10) NOT NULL,
STIPEND INTEGER,
CITY CHAR(15),
BIRTHDAY DATE,
UNIV_ID INTEGER);
```

Ba’zi hollarda biror maydonga kiritilayotgan barcha qiymatlar bir-biridan farq qilishi kerak. Bunda shu maydon uchun **UNIQUE** (**YAGONA**) so‘z ishlataladi.

Masalan, STUDENT jadvalida STUDENT_ID qiymatlari farqli bo‘lishi uchun komanda quyidagicha bo‘ladi:

```
CREATE TABLE STUDENT
(STUDENT_ID INTEGER NOT NULL UNIQUE,
SURNAME CHAR(25) NOT NULL,
STIPEND INTEGER,
KURS INTEGER,
CITY CHAR(15),
BIRTHDAY DATE);
```

Jadvalda kalit maydonlarni ishlatalish komandasini quyidagicha bo‘ladi:

```
CREATE TABLE STUDENT
(STUDET_ID INTEGER PRIMER KEY,
```

**SURNAME CHAR(25) NOT NULL,
NAME CHAR(10) NOT NULL,
STIPEND INTEGER,
KURS INTEGER,
CITY CHAR(15),
BITHDAY DATE,
UNIV_ID INTEGER);**

Umuman olganda jadvalga kiritilayotgan ma'lumotlar toifasini aniqlash va uning o'lchamlarini ko'rsatish va ularni mos ravishda jadval atributlariga moslash ma'lum ma'noda cheklanishlarni aniqlaydi. Shu bilan birga, jadvallarni tavsiflashda ustunlarga yoki ustunlar guruhiga kiritiladigan qiymatlarga qo'shimcha cheklanishlar ham o'rnatilishi mumkin. Cheklanishlar jadval ustunlariiga va jadvallarga qo'yilishi mumkin. Jadvaldagি ustunlarga qo'yilgan asosiy cheklanishlar quyidagi jadvalda keltirilgan:

Ta'rif	Tavsif
NOT NULL	Jadval ustunida NULL qiymat bo'lmasligi, ya'ni uning qiymatlari aniqlangan bo'lishi kerak
UNIQUE	Ustunga kiritilayotgan qiymat shu ustunning boshqa barcha qiymatlaridan farq qilishi kerakligini, ya'ni yagona bo'lishi kerakligini belgilaydi
PRIMARY KEY	Ustun birlamchi kalit bo'lib keladi. Har bir jadvalda faqat bitta ustun birlamchi kalit bo'la oladi. U NULL qiymatga ega bo'lmasligi kerak va unga kiritilayotgan qiymat shu ustun qiymatlaridan farq qilishi kerakligini anglatadi
DEFAULT qiymat	Sukut bilan qiymat o'rnatiladi. Bunday cheklanishli ustunga yangi yozuv kiritilayotganda avtomatik ravishda ko'rsatilgan qiymatni oladi
CHECK (shart)	Ma'lumotlarni kiritishda shartni tekshirish imkonini beradi. Shart bajarilsa, qiymat saqlanib qoladi, aks holda saqlanmaydi

Jadvalga cheklanishlar jadvaldagи ustun ro'yxatlari keltirilib amalga oshiriladi.

Savol va topshiriqlar

1. DML operatorlari va ularning vazifalari.
2. DDL operatorlari va ularning vazifalari.

3. SQLda shartli so‘rovlardada qanday operatorlar (amallar) ishlataladi?
4. Sodda SELECT buyrug‘i parametrlarining vazifalarini ta’riflab bering.
5. SQL tilida universal dasturlash tillarida ishlataladigan qanday operatorlar uchramaydi?
6. SQL tilining 2 ta asosiy komponenti va uning bajaradigan funksiyalari.
7. SELECT operatori asosiy konstruksiyalari va ularga qo‘yiladigan cheklanishlar.
8. SELECTda guruhli (agregat funksiyalarni) ishlatish.
9. WHERE va HAVING konstruksiyalarining farqi.
10. Qism so‘rov bilan so‘rovlarni birlashtirishning farqi nimada?
11. Shartli so‘rovlar qanday tashkil qilinadi?
12. Guruhli funksiya ko‘rinishlari.
13. Guruhli funksiyaning vazifalari.

VII BOB. MA'LUMOTLARNI TAQSIMLANGAN QAYTA ISHLASH

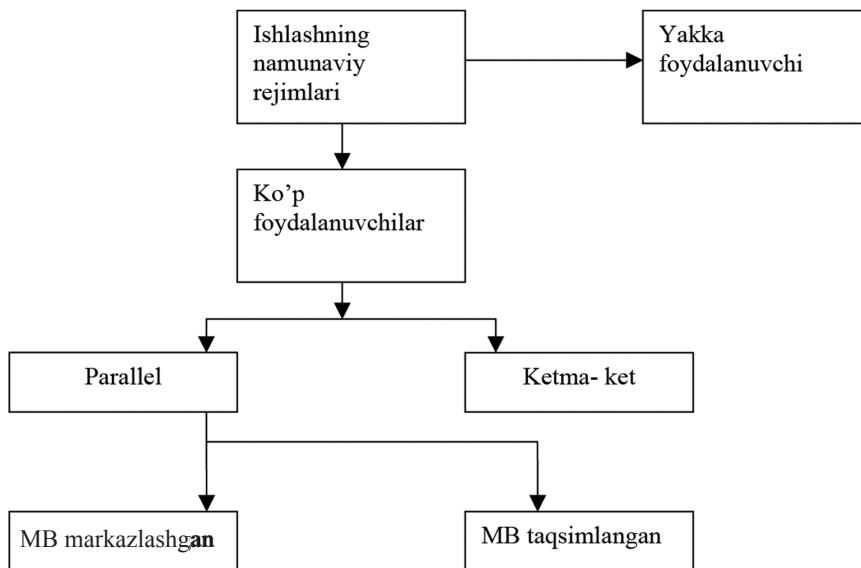
7.1. Umumiyl tushunchalar

Tarmoqqa ulanmagan shaxsiy kompyuterda MBBT joylashtirilganda, MB doim tanho huquqqa ega bo'lgan rejimda ishlataladi. Agar bu MB bilan bir nechta foydalanuvchi ishlasa (foydalansa) ham ularni faqat ketma-ket ishlata olishlari mumkin. Lekin local (mahalliy) ma'lumot bazasini qo'llash shuni ko'rsatdiki, ularga mo'ljallangan axborotlar ko'p foydalanuvchi xarakterga ega. Shuning uchun ma'lumot bazasi bilan foydalanuvchilarni bir vaqtida ishlash imkoniyatini ta'minlovchi MBBT ishlab chiqish zaruriyatini vujudga keltiradi.

Axborotlarga har xil foydalanuvchilarning bir vaqtida murojaat qilish imkoniyatini ta'minlovchi ma'lumotlar bazasini boshqarish tizimlariga, taqsimlangan ma'lumot bazalari boshqarish tizimlari deb ataladi.

Umuman olganda MB ishlatish rejimlari 7.1.1-rasmida ko'rsatilgan.

Taqsimlangan MBBTning ishlatilgan asosiy tushunchalarni ko'rib chiqamiz.



7.1.1- rasm. Ma'lumotlar bazasi bilan ishlash resurslari.

MB foydalanuvchisi ma'lumot bazasiga murojaat qiluvchi shaxs yoki dastur:

— so'rov foydalanuvchini MBga undan axborotlarni o'zgartirish, olish yoki kiritish maqsadida qo'llangan murojaat jarayoni;

— tranzaksiya MBdagi ma'lumotlarni takomillashtirish (olish, o'chirish, o'zgartirish va boshqalar) amallar ketma-ketligi. U MBni bir ziddiyatsiz holatlardan boshqa ziddiyatsiz holatga o'tkazadi.

MB mantiqiy tuzilmasini MBning fizik mustaqil bosqichda aniqlash hammadan yaqinroq, u MBning konseptual modeliga mos keladi.

MB topologiyasi yoki taqsimlangan ma'lumot bazasining strukturasi — tarmoqda MBni fizik tashkil qilishni taqsimlash sxemasi.

Lokal avtonomli (локальная автономность) — bu mahalliy MB axborotni va u bilan bog'langan ma'lumotlarni aniqlash mahalliy shaxsga tegishli ekanligini va u tomonidan boshqarishni anglatadi.

Uzoqlashgan so'rov — modemli bog'lanishi ishlatilib bajariladigan so'rov.

Uzoqlashgan tranzaksiyalarni amalga oshirish imkoniyati — SQL so'rovlar to'plamidan tashkil topgan bitta tranzaksiyani bitta uzoqlashgan tugunda qayta ishlash. Taqsimlangan tranzaksiyalarni ta'minlash bir nechta SQL so'rovlardan tashkil topgan bir necha tarmoq tugunlarida (uzoqlashgan yoki mahalliy) bajarilayotgan, har bir so'rov shu bitta tugunda qayta ishlanadi.

Taqsimlangan so'rov har xil tarmoq tugunlardagi MB ma'lumotlarini qayta ishlashda foydalanadigan so'rovdir. Taqsimlangan ma'lumotlarni qayta ishlash tizimlar birinchi avlodni MB bilan bog'langan. Ularda multiprogrammasi operatsion tizimlar asosida qurilib markaziy EHM tashqi xotira qurilmalarida saqlangan va murojaatni terminalni ko'p foydalanuvchi rejimida foydalangan.

Bunda foydalanuvchi terminallari o'z resurslariga (ya'ni ma'lumotlarni qayta ishlash va saqlash uchun ishlatiladigan) ega bo'limganlar.

7.2. Ma'lumotlarni taqsimlangan qayta ishlashda mijoz-server modeli

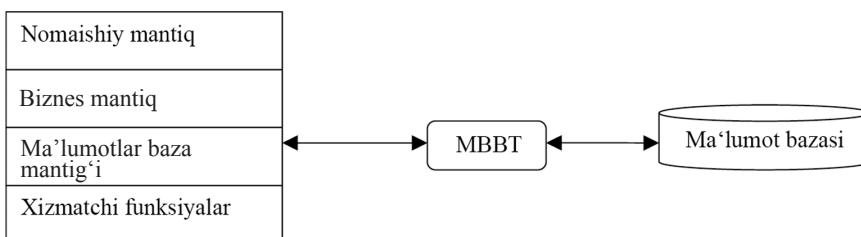
Taqsimlangan ma'lumotlar bazasi texnologiyalarida mijoz-server modeli 1990-yil ochiq tizimlarning paydo bo'lishi bilan bog'liq.

Kliyent server tushunchasi axborotlarni qayta ishlashning ikki jarayonidan tashkil topgan (kliyent va server) dasturiy ta'minot arxitekturasida ishlatalilgan (kliyent mijoz). Bu tizimda mijoz jarayoni serverdan ba'zi bir xizmatlarni bajarishni so'raydi, server esa ularning bajarilishini ta'minlaydi. Bunda bitta server ko'pincha mijozlarga xizmat ko'rsatadi. MB boshqarishning bu modeli texnik amalga oshirish, korxonalarini mahalliy hisoblash tarmoqlari bilan bog'liq bo'lganligi uchun axborotlarni bunday qayta ishlash jarayoni tashkil qilishning kliyent server arxitekturasi deb ataladi.

MBni boshqarish texnologiyasiga nisbatan tatbiq qilishdagi kliyent server texnologiyasining asosiy prinsipi inter ilovaning standart funksiyalarini, har xil tabiatga ega bo'lgan 5 guruhga ajratishdan iborat:

- ma'lumotlarni kiritish va akslantirish funksiyalari (presentations Logic);
- ilova masalalarini yechishning asosiy amallarini aniqlovchi amaliy funksiyalar (Business Logic);
- ilova ichida ma'lumotlarni qayta ishlash funksiyalari;
- axborot resurslarni boshqarish funksiyalari;
- birinchi tur guruh funksiyalari orasida bog'lanish vazifasini bajaruvchi xizmatchi funksiyalar.

Kliyent server arxitekturasida ma'lumotlar bazasi bilan ishlatalidigan ilovani namunaviy tizimli rasmida keltirilgan:



7.2.2-rasm. Ma'lumot bazasi bilan ishlaydigan namunaviy ilova tuzilmasi.

Nomaishiy mantiq ilovani ishlatayotganda foydalanuvchi o‘z ekranida ko‘radigan ilova qismi bilan aniqlanadi. Bunga ilovani ishslash jarayonida foydalanuvchi ko‘rayotgan yoki to‘ldiriladigan bir nechta interfeysli ekran formalari kiradi. Bu qismga shuningdek, foydalanuvchi ekraniga chiqarilayotgan qandaydir oraliq masalalarni yechish natijalari yoki ma’lumotnomma axborotlari kiradi:

- ekran tasvirlarini yaratish;
- ekran formalaridagi axborotlarni o‘qish va yozish;
- ekranni boshqarish.

Biznes-logika yoki xususiy ilova mantig‘i. Bu ilova kodining qismi bo‘lib, ilovaning aniq masalalarini yechishning xususiy algoritmlarini aniqlaydi. Odatda bu kod har xil dasturlash tillarini ishlatib yoziladi, jumladan C,C++, Visual Basic va boshqalar.

Ma’lumotlarni qayta ishslash mantig‘i bu ilova kodining qismi bo‘lib, ilova ichida ma’lumotlarni qayta ishslash bilan bevosita bog‘liq. Ma’lumotlarni bevosita MBBT boshqaradi. Ma’lumotlarga murojaatni ta’minlash uchun SQL tili ishlatiladi.

Ma’lumotlarni boshqarish protsessori – bu bevosita MBBT. Odatda MBBT funksiyalari ilova biznes mantig‘idan yashiringan bo‘lishi kerak, lekin ilova arxitekturasini ko‘rish uchun ularni ilovaning alohida qismi sifatida ajratish kerak.

Markazlashgan arxitekturada ilovani bu qismlari yagona muhitda joylashadi va bitta bajariluvchi dastur ichida guruhlanadi. Kliyent va server jarayonlari o‘rtasida masalalar har xil taqsimlashishi mumkin.

Taqsimlash xarakteriga qarab maksimallashning quyidagi modellarini ajratib ko‘rsatish mumkin:

- taqsimlangan prezентatsiya;
- uzoqlashgan prezентatsiya;
- taqsimlangan biznes mantiq;
- ma’lumotlarni taqsimlangan boshqarish;
- ma’lumotlarni uzoqlashgan boshqarish.

Bu shartli guruhlash ma’lumotlar bilan manipulatsiyalashda alohida masalalarni server va kliyent jarayonlari orasida qanday taqsimlanishi mumkunligini ko‘rsatadi.

7.3. Ma’lumot bazasining server modeli

Ikki sathli modellar. Ikki sathli modellar yuqorida sanab o‘tilgan 5 funksiya ikkita kliyent va server platformasida baja-

rilađigan jarayonlar o'rtasida taqsimlanishi natijasi deb qarash mumkin.

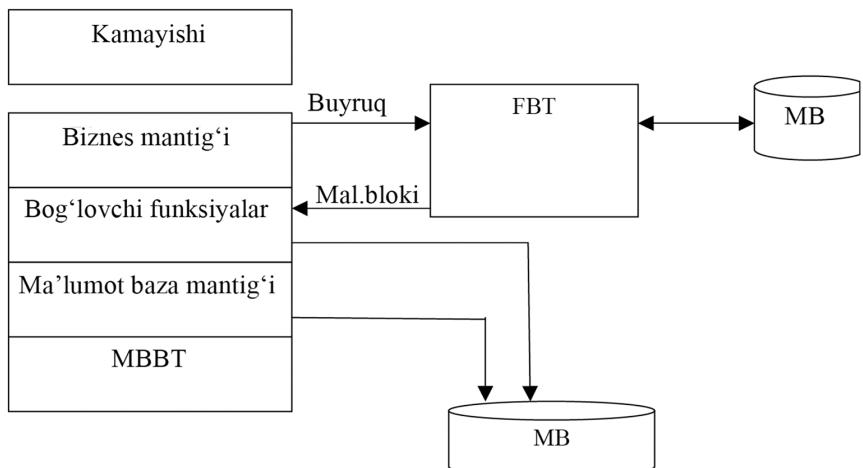
Har bir ikki sathli modelning asosiy xarakterli xususiyatlarini ko'rib chiqamiz:

- ma'lumotlarni uzoqdan boshqarish modeli;
- ma'lumotlarga uzoqdan murojaat modeli.

Ma'lumotlarni uzoqdan boshqarish modeli ba'zan fayl serverli model deb ataladi. Bu modelda namoyish mantig'i va biznes mantiq mijoz qismiga joylashadi.

Serverda ma'lumotlar fayli joylashtiriladi va faylga murojaat ta'minlanadi.

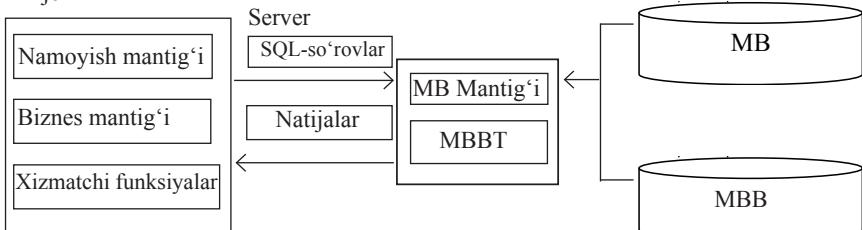
Axborot resurslarini boshqarish kliyent qismida joylashadi. Bu modelda funksiyalarning taqsimlanishi rasmda keltirilgan.

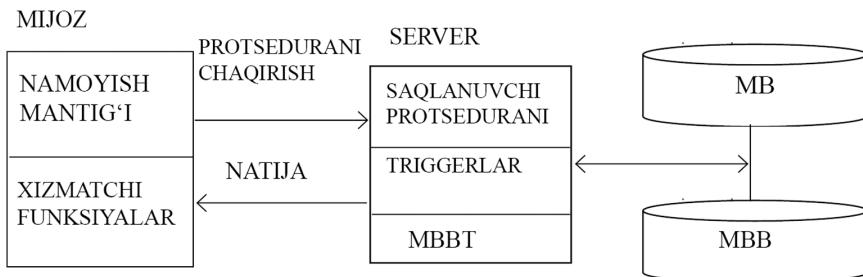


7.3.1-rasm. Ma'lumotlarga uzoqlashgan murojaat modeli.

Bu modelda MB serverga SQL tilida murojaat qiladi. Ma'lumotlarga uzoqlashgan murojaat modeli tuzilishi rasmda keltirilgan.

Mijoz





7.3.2-rasm. Ma'lumotlar bazasining faol server modeli.

Bu ikki model ham afzallik va kamchiliklarga ega. Shuning uchun hozirgi kunda ma'lumot baza server modeli keng tarqalgan 7.3.1-rasm.

Savol va topshiriqlar

1. Taqsimlangan ma'lumotlar bazalarini boshqarish tizimlari nima?
2. Taqsimlangan ma'lumotlar bazasi tuzilishi qanday?
3. Uzoqlashgan server so'rovni tushuntiring.
4. Kliyent server tushunchasi haqida ma'lumot bering.
5. Tarmoqning fayl server arxitekturasi nima?
6. Inter ilovaning standart funksiyalarini sanab bering.
7. Ikki bosqichli modelni tavsiylab bering.

VIII BOB. WEB TEXNOLOGIYALARI VA MA'LUMOTLAR BAZASI

8.1. PHP tili haqida

Hozirgi kunda internetni tez suratlar bilan rivojlanishi insonlar orasida yangi muloqot qilish usullarining taraqqiy qilinishiga olib keldi. WWW kommunikatsion muhitida ko'plab yangi saytlar ochilmoqda. Shu munosabatda yangi texnologiyalarga va ular bilan ishlaydigan mutaxassislarga talab ortib bormoqda. Shunday yangi texnologiyalardan biri – PHP hisoblanadi.

PHP tilining tarixi 1995-yildan boshlanadi. Mustaqil dasturchi Rasmus Lerdorf saytga uni onlayn ma'lumotlarini o'qish uchun tashrif buyuruvchilar sonini hisoblash uchun (Pere/CGI) ssenariyasini yozgan. Bu ssenariy 2 ta masalani yechgan. Tashrif buyuruvchilar haqida ma'lumotlar va web sahifaga tashrif buyurganlarning sonini chiqarish. Bu davrga kelib bu masalani yechish uchun maxsus vositalar bo'lмаган. Shu sababli ko'п dasturchilar uning ishiga qiziqib savol bilan murojaat qila boshladilar va mual-lif vositasini tekin tarqata boshladi, o'zi esa PHP vositalarini kengaytirish ustida ishlay boshladi. Shunday kengaytirishlardan biri HTML shaklida kiritilgan ma'lumotlarni simvolli o'zgaruvchilarga almashtirardi, bu esa ularni boshqa tizimlarga eksport qilish imkonini berardi. Mavjud PHP vositalarini kengaytirishlar PHP 2.0 versiyasining paydo bo'lishiga olib keldi. Uni takomillashtirishda butun dunyo dasturchilari ishtirot etdilar. 1997-yilda PHP qisqartmasi «Personal Home Page» ma'nosini («PHP Hypertext Processor» emas) anglatishi kelishib olindi.

PHPning muhim afzalliklaridan biri, u to'g'ridan to'g'ri HTML kodiga singdirilishi (o'rnatilishi) mumkin, shuning uchun dasturchiga HTML chiqarish uchun bir qancha buyruqdan iborat dastur tuzishga hojat yo'q. HTML va PHP kodlarini zaruriyatga qarab almashtirib turish mumkin. Masalan, PHPda quyidagi dastur fragmentini yozish mumkin.

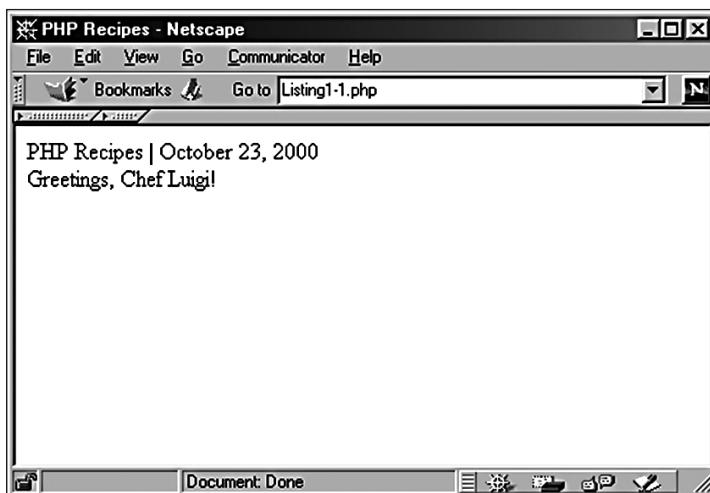
```
<html>
<title><? print "Hello world!"; ?></title>
</html>
```

"Hello world!" xabar Web sahifaning sarlavhasida chiqariladi. Shunisi qiziqliki, PHPni ekranlovchi ketma-ketligi (<?...?>) deb atlaluvchi ushbu konstruksiya ichiga yozilgan print komandasi tugalangan dasturdan iborat.

Quyidagi misolda PHPning HTML kodlari bilan integrallashuvi ko'rsatilgan:

Листинг 1.1. PHPning dinamik sahifasini yaratish.

```
<?
// bir nechta o'zgaruvchilarga qiymat tayinlash
$site_title = "PHP Recipes";
$bg_color = "white";
$user_name = "Chef Luigi";
?>
<html>
<head>
<title><? print $site_title; ?></title>
</head>
<body bgcolor="<? print $bg_color; ?>" >
<?
// tabrikni sana va foydalanuvchi nomi bilan chiqarish. print "PHP Recipes | ".date('F d, Y')." <br> Greetings, $user_name!
<br>";
?>
</body>
</html>
```



8.1.1-rasm. Brauzerda ssenariyning bajarilishi natijasi.

PHP dasturiga qo'yilgan masalani samarali va tez yechish vositalarini berishi kerak. PHP tilini loyihalashda bosh faktor

uning amaliyligidir. Uning amaliyligi uning 5 muhim xarakteristikasiga asoslangan:

- an’anaviyligi;
- oddiyligi;
- samaradorligi;
- xavfsizligi;
- moslashuvchanligi.

Bundan tashqari PHP tekin tarqatiladi.

PHP ko‘pgina konstruksiyalari Cu va Perl tillariga o‘xhash. PHP tilini kod ko‘p hollarda Cu va Pascal tilidagi dasturlardan farq qilmaydi.

PHPning asosiy konstruksiyalari. PHP kodini identifikatsiyalash «PHPga o‘tish» (escaping to PHP). PHPga o‘tishni rasmiy lashtirishning 4 varianti mavjud:

- standart teglar;
- qisqa teglar;
- script teglar;
- ASP uslubidagi teglar.

Standart teglar PHP dasturchilari tomonidan boshqa teglarga nisbatan ko‘proq ishlatiladi. Bu shakldagi yozish qulayligi va yaqqolligi ajralib turadi. <? php print "Welcome to the world of PHP!" :> Bunda yopuvchi teggacha ?> joylashgan barcha matn, PHP kodi deb talqin qilinadi.

Qisqa teglar PHPga o‘tishning ixcham yozuv ko‘rinishini ta’minlab beradi.

<? print "Welcome to the world of PHP!"; ?>

Sukut bilan qisqa teglar ishlatilmaydi. Ularni ishlatish uchun ularni faollashtirish kerak. Buni ikki usul bilan bajarish mumkin:

PHP kompilatsiya qilishda enable_short_bays kalitni ko‘rsatish short_open_tag parametrini PHPning fayliga kiritish
Script teglari.

Ba’zi bir matn tahririylari PHP kodini HTML kodi deb qabul qiladi. Bu web sahifa ustida ishlashni buzadi. Bu muammoni Script teglari ishlatib hal qilinadi.

```
<script language= "php">
<? Php print "Welcome to the of PHP!"; ?>
<Script>
```

ASP uslubidagi teglar qisqa teglarga o‘xhash, unda so‘rov belgisi o‘rniga foiz (%) belgisi ishlatiladi:

```
<% php print "welcome to the world of PHP"; %>
ASP stildagi tegni ixcham qiluvchi bitta xususiyati bor.
Kiritilgan PHP kodi uchun print komandasini kiritish shart
emas. Bunda ochuvchi tegdan keyin qo'yilgan tenglik belgi-
si (=) PHPga ko'rsatilgan o'zgaruvchi qiymatini chaqirishni
buyuradi.
```

<% = \$variable %>

Haqiqiy sonlarning standart ko'rinishi:

12.45

98. 6;

Eksponentali ko'rinishni:

3e8

5.9736e24

Satrli qiymatlar (string). Misollar:

thesaurus

49 ers

abc

&%\$#

PHPdan

PHP tilida izohlar ikki xil formatda beriladi:

Bitta satrli izohlar – odatda mahalliy kodlarga beriladigan tushuncha va eslatmalarda ishlataladi.

Ko'p satrli izohlar – bu izohlar psevdokodda algoritmlarni rasmiylashtirishda va to'la tushuntirishlar talab qilinganda ishlataladi. Bitta satrli izohlar yozishda ikkita simvoldan (belgidan) foydalanish mumkin. ("//") va ("#").

Misollar:

<?

```
// Atirgul rangini tanlang $rose_color = "red";
```

```
# Binafsha gul rangini tanlang $violet_color = "blue";
```

```
print "Roses are $rose_color, violets are $violet_color"
```

?>

Albatta ikkala uslubdagi izohlarni ko'p satrli izohlarda ham ishlatalish mumkin.

<?

```
// fayl: example. php
```

```
// muallif: U.Dj. Gilmor
```

```
// sana: 24-avgust 2011-y.
```

```

print "An example with comments";
?>
Ko'p satrli izohlar «C» dasturlash tili kabi rasmiylashtiriladi,
ya'ni uni boshlanishi va oxiri mos ravishda /* va */ belgi bilan
ajratiladi.
<?
/*
Ssenariy: multi_coramment_example.php
Vazifasi: ko'p satrli izohlarni ishlatishga misol
Muallif: U.Dj. Gilmor
Sana: 14-iyun 2011-y.
*/
print "A multiline comment can be found at the top of this
script!";
?>

```

8.2. O'zgaruvchilar va ma'lumot toifalari

O'zgaruvchilar va ma'lumot toifalari ixtiyoriy dasturlash tili-ning asosini tashkil qiladi va u bilan har xil toifadagi dasturchini ko'rsatish vositasi hisoblanadi. PHPda 6 toifa ishlatiladi:

- butun sonlar;
- haqiqiy sonlar;
- satrlar;
- massivlar;
- obyektlar;
- mantiqiy kattaliklar.

Butun sonlar kasr qismi bo'limgan va ketma-ket joylashgan bir yoki bir nechta raqamlardan tashkil topadi. Misollar:

5
591
52

Butun sonlar sakkizlik va o'n otilik sanoq sistemasida ham berilishi mumkin. Sakkizlik sanoq sistemasidagi sonlar 0 raqami bilan boshlanadi va undan keyin 0 dan 7 gacha sonlar keladi. Misollar:

0422
0534

O'n otilik sanoq sistemasida sonlar ox yoki OX bilan boshlanadi va 0 dan 9 gacha raqamlar va (A)dan to f (F)gacha harflar-dan iborat bo'ladi. Misollar:

0x3FF
0x22abc

Haqiqiy sonlar. Haqiqiy sonlar (suzuvchi nuqtali sonlar) butun sonlardan kasr qismining mavjudligi bilan farqlanadi. Ular yuqori aniqlikdagi qiymatlarni tasvirlash uchun ishlatiladi. Masalan, temperatura yoki pul kattaliklari. PHPda ikkita haqiqiy format standart va ilmiy (eksponensial) yozuv ishlatiladi. Standart yozuv tipik haqiqiy sonlarni tasvirlashga qulay. Masalan, pul kattaliklari. Misollar:

12.45
98.6

Ilmiy yozuvlar juda katta va juda kichik sonlarni tasvirlash uchun ishlatiladi. Masalan, planetalar orasidagi masofalar yoki atom o'lchamlari. Misollar:

3e8
5.9736e24

Satrli qiymatlar. Satr (String) deb bir butundek qaraladigan simvollar (belgilar) ketma-ketligiga aytildi. Bunda alohida simvollarga murojaat ta'minlanadi. Satrga misollar:

thesaurus
49ers
abc
&%/\$#

PHPda simvolli ma'lumot aniqlanmagan. Satrli toifa bitta yoki bir nechta simvollarni bir butunday qaraydi. Simvolli toifani uzunligi bitta simvoldan iborat satr deb hisoblash mumkin. Satrlar ikkita toifaga bo'linadi va ular mos ravishda bir juft qo'shtirnoq (" ") yoki apostrof (' ') bilan cheklanadi. Bu toifalar o'zaro bir-biridan farq qiladi. Qo'shtirnoq ichida yozilgan o'zgaruvchilar nomi mos qiymatlari bilan almashtiriladi. Apostrof ichidagi satrlar o'z qiymatlari bilan aniqlanadi. Quyidagi ikkita tavsiflash bir xil qiymat beradi. Misol:

```
$food = "meatloaf";  
$food = 'meatloaf';
```

Lekin quyidagi tavsiflar (e'lonlar) keskin farq qiladi.

```
$sentence = "My favorite food is $food";  
$sentence2 = 'My favorite food is $food';
```

O'zgaruvchi \$sentence quyidagi satr tayinlanadi

My favorite food is meatloaf.

E'tibor bering o'zgaruvchi \$food avtomatik ravishda talqin qilinadi. Ikkinci tomondan o'zgaruvchi \$sentence2 satr tayinlanadi.

My favorite food is \$food.

\$sentence o'zgaruvchidan, farqli \$sentence2 o'zgaruvchida \$food o'zgaruvchi talqin qilinmay qoldi. Bu apostrof va qo'shtirnoqlarning ishlatalishidagi farq bilan asoslanadi.

PHP tilida satrlarda ham xizmatchi simvollar (masalan, tabulatsiya simvoli yoki yangi satr belgisi) ishlataladi. Satrlardagi xizmatchi simvollar quyidagi jadvalda keltirilgan:

8.2.1-jadval.

Satrda xizmatchi belgilari

| Ketma-ketlik | Ma'nosi |
|--------------------|--|
| \n | Yangi satr |
| \r | Kursorni qaytarish |
| \t | Gorizontal tabulatsiya |
| \\" | Teskari yotiq chiziq |
| \\$ | Dollar belgisi |
| \" | Qo'shtirnoq |
| \[0-7]{1,3} | Sonning sakkizlik yozuvi (muntazam ifoda ko'rinishida) |
| \x[0-9A-Fa-f]{1,2} | Sonning o'n oltilik yozuvi (muntazam ifoda ko'rinishida) |

Qo'shtirnoqning ikkinchi muhim farqi shundaki, uning tarkibida ishlataligan barcha xizmatchi simvollar aniqlanadi. Apostrof ichida ishlataligan xizmatchi simvollardan faqat «\» va «\» aniqlangan. Satrning har bir simvoliga ketma-ket nomerlangan massiv elementlari kabi murojaat qilish mumkin. Misol:

```
$double_list = "item1\nitem2\nitem2";
```

```
$single_list = 'item1\nitem2\nitem2';
```

Massivlar. Bir o'lchamli massivlarda indekslar sifatida butun sonli indeks ishlataladi va u element o'rnnini aniqlaydi. Bir o'lchamli massiv elementlarining sintaksisi quyidagicha:

```
$имя[индекс];
```

Bir o'lchamli massiv quyidagicha yaratiladi:

```
$meat[0] = "chicken";
```

```
$meat[1] = "steak";
```

```
$meat[2] = "turkey";
Quyidagi komanda bajarilganda
print $meat[1];
Brauzerda quyidagi satr chiqadi:
steak
```

Massivlarni yaratishda array () funksiyasi ishlataladi. \$meat oldingi misoldan quyidagi komanda bilan yaratiladi:

```
$meat = array("chicken", "steak", "turkey");
```

Yuqorida keltirilgan print komandasini xuddi shunday natijaga olib keladi, ya'ni steak satrini chiqaradi.

PHPda massivning elementlarini nomerlash 0 dan boshlanadi. Massivning oxiriga yangi element kiritish uchun massiv o'zgaruvchisiga uning indeksini ko'rsatmasdan faqat qiymat tayinlansa bo'ladi. Demak, \$meat massivni boshqacha quyidagi usulda yaratish mumkin

```
$meat[] = "chicken";
$meat[] = "steak";
$meat[] = "turkey";
```

Ko'p o'lchamli indekslanuvchi massivlar. Ko'p o'lchamli indekslanuvchi massivlar indekslari bir nechta indeks bilan aniqlanadi. Nazariy jihatdan indekslar soni cheklanmagan. Amaliyotda esa massiv o'lchami 3 dan katta bo'lgan massivlar kam ishlataladi. Ko'p o'lchamli massiv sintaksisi quyidagicha:

```
$имя[индекс1][индекс2]..[индексN];
```

Ikki o'lchamli indeksli massiv elementiga quyidagicha murojaat qilish mumkin:

```
$position = $chess_board[5][4];
```

PHP tilida bundan tashqari assotsiativ massivlar (bir o'lchamli va ko'p o'lchamli) ham ishlataladi. Shuningdek, indekslari aralash (sonli indeksli va assotsiativ indeksli) massivlar bo'lishi mumkin.

Obyektlar. Sinf deb nomlanuvchi maxsus shablon asosida yaratiladigan o'zgaruvchi nusxasini obyekt deb qarash mumkin. Boshqa ma'lumot toifalaridan farqli PHP tilida obyektlar oshkor holda e'lon qilinishi kerak. Obyekt sinfning konkret nusxasi bo'lib, aniq xarakteristikali va funksional imkoniyatlari obyektlar yaratish uchun shablon sifatida ishlataladi. Demak, obyektlar e'lon qilinishdan avval u tegishli sind e'lon qilinishi kerak. Sinfni e'lon qilish va uning asosida obyekt yaratishga doir misol.

```

class appliance {
var power;
function set_power($on_off) {
$this->power = $on_off;
}
}
...

```

\$blender = new appliance;

Sinfni aniqlash birorta ma'lumotlar strukturasiga bog'langan atributlari va funksiyalari bilan beriladi. Bizning misolda bu struktura appliance (qurilma) nomi bilan atalgan. Bu struktura faqat bitta atributga power (quvvat)ga ega. Bu atributni o'zgartirish uchun set_power usuli yaratiladi. Shuni yodda tutish kerakki, sinfni aniqlash (e'lon) – faqat shablon bo'lib u bilan dasturda amal bajarish mumkin emas. Bu shablon asosida obyektlar yaratiladi. Obyektlar new kalit so'zi yordamida yaratiladi. Yuqorida keltirilgan misolda appliance sinfining \$blender obyekti yaratilgan. \$blender obyekti yaratilgandan so'ng uning quvvatini quyidagi usul bilan berish mumkin.

set_power: \$blender->set_power("on");

Mantiqiy kattaliklar (rost/yolg'on). Mantiqiy ma'lumot toifalarida faqat ikkita qiymat qabul qiladi:

Rost (True) va Yolg'on (False). Mantiqiy kattaliklar ikki usulda yaratiladi:

- shartlarni tekshirishda;
- o'zgaruvchining qiymati ko'rinishida.

Solishtirish bir qancha shakllarda mavjud. Ko'pgina hollarda ular («==» operatorini ishlatishda if shartli buyruqda uchraydi. Misol:

if (\$sum == 40) :

...

Tekshirish natijasi rost yoki yolg'on bo'ladi. \$sum yoki 40 ga teng yoki 40 ga teng emas. Agar \$sum 40 ga teng bo'lsa tekshirish to'g'ri natija beradi. Aksincha bo'lsa, natija «false»ga teng. Mantiqiy kattaliklar, shuningdek, o'zgaruvchiga rost yoki yolg'on qiymatlarni oshkor ravishda tayinlash bilan aniqlanishi mumkin.

\$flag = TRUE;

if (\$flag == TRUE) :

print "The flag is true!";

```

else :
print "The flag is false!";
endif;

```

Agar o'zgaruvchi \$flag rost bo'lsa, birinchi xabar chiqariladi, agar yolg'on bo'lsa ikkinchi xabar chiqariladi. Rost va yolg'on mantiqiy kattaliklarni bir va nol qiymatlar ko'rinishida ham berish mumkin. Bunda oldingi misol quyidagi ko'rinishda bo'ladi:

```

$flag = 1;
if ($flag == TRUE) ;
print "The flag is true!";
else :
print "The flag is false!";
endif;

Nihoyat, yana bir usul mavjud:
$flag = TRUE;
// Bu buyruq bajarilganda bilvosita
// "if ($flag == TRUE)" shart tekshiriladi
if ($flag) :
print "The flag is true!";
else :
print "The flag is false!";
endif ;

```

Identifikatorlar. Bu tushuncha o'zgaruvchilar, funksiyalar va foydalanuvchilar tomonidan aniqlanadigan boshqa obyektlarga ishlataladi. PHPda identifikatorlar bir nechta shartlarni qanoatlantirishi kerak. Identifikatorlar bir yoki bir necha simvollardan tashkil topib, harf yoki «_» simvol bilan boshlanadi. Identifikatorlar faqat harflar, raqamlar, «_» simvolidan va ASCII ni 127 dan 257 gacha kodli simvollardan tashkil topadi. Misollar:

| To'g'ri identifikatorlar | Noto'g'ri identifikatorlar |
|--------------------------|----------------------------|
| my_function | This&that |
| Size | !counter |
| _someword | 4ward |

Identifikatorlarda simvollarning registri hisobga olinadi. Demak, \$recipe nom \$Recipe farq qiladi. Identifikatorning uzunligi cheklanmaydi. Identifikator PHP tilidagi kalit (xizmatchi) so'zlar bilan bir xil bo'lmasligi kerak.

O‘zgaruvchilar. O‘zgaruvchilar – nomlangan xotira sohasi hisoblanadi. Bu xotira sohasida ma’lumot joylashgan bo‘lib u bilan dastur har xil amallar bajaradi. PHPda o‘zgaruvchi nomlari har doim \$ belgisi bilan boshlanadi. Quyidagi misolda to‘g‘ri yozilgan nomlar keltirilgan:

```
$color  
$operating_system  
$_some_variable  
$model
```

O‘zgaruvchi nomlari ham identifikatorlarga qo‘yilgan talablarga javob berishi kerak. Boshqacha aytganda, o‘zgaruvchi nomi harf yoki « _ » simvoli bilan boshlanib, harf, « _ », raqam yoki 127 dan 255 gacha bo‘lgan intervaldagi ASCII simvollaridan tashkil topadi. PHP tilida o‘zgaruvchilar maxsus e’lon qilishni talab qilmaydi. Buning o‘rniga o‘zgaruvchi dasturda birinchi marta ishlataliganda aniqlanadi. Bundan tashqari o‘zgaruvchi tipi (toifasi) unda saqlanadigan ma’lumot toifasi bilan aniqlanadi. Misollar:

```
$sentence = "This is a sentence."; // $sentence satr deb talqin qilinadi.
```

```
$price = 42.99; // $price haqiqiy son deb talqin qilinadi.
```

```
$weight = 185; // $weight butun son deb talqin qilinadi.
```

O‘zgaruvchi PHP ssenariyasi ixtiyoriy nuqtasida e’lon qilinishi mumkin.

O‘zgaruvchilarning ko‘rinish sohalari. O‘zgaruvchilarning ko‘rinish sohasi u e’lon qilingan dasturda murojaat qilish sohasi bilan aniqlanadi. PHP tilida o‘zgaruvchilar ko‘rinish sohasiga qarab 4 ta toifaga bo‘linadi:

- lokal (mahalliy) o‘zgaruvchilar;
- funksiya parametrlari;
- global o‘zgaruvchilar;
- statik o‘zgaruvchilar.

Lokal o‘zgaruvchilar. Funksiya ichida e’lon qilingan o‘zgaruvchilar lokal hisoblanadi. Boshqacha aytganda, shu funksiyadan murojaat qilish mumkin. Funksiyadan chiqilgandan so‘ng o‘zgaruvchi va uning qiymati yo‘qoladi. Lokal o‘zgaruvchilarning asosiy afzalligi – global o‘zgaruvchilarda uchraydigan tasodifiy yoki mo‘ljallangan takomillashtirish bilan bog‘liq bo‘lgan nojo‘ya ta’sirlarning bo‘lmaslidir. Misol:

```

$x = 4;
function assignx () {
$x = 0;
print "\$x inside function is $x. <br>";
}
assignx();
print "\$x outside of function is $x. <br>";

```

Bu fragment bajarilganda quyidagi natija chiqariladi:

\$x inside function is 0.

\$x outside of function is 4.

Misoldan ko‘rinadiki, \$x o‘zgaruvchining ikkita har xil qiymati chiqarilayapti. Gap shundaki, \$x o‘zgaruvchisi assignx funksiyasi ichida lokal tabiatga ega va uni qiymatining o‘zgarishi bu funksiyadan tashqaridagi qiymatida akslanmaydi. Xuddi shuningdek, \$x o‘zgaruvchini funksiyadan tashqarida takomillashtirish (o‘zgartirish) assignx() funksiya lokal o‘zgaruvchilarda akslanmaydi.

Funksiya parametrlari. PHP tilida funksiyani chaqirishda uzatiladigan har qanday parametrler funksiya sarlavhasida e’lon qilinishi kerak. Parametrlarga tashqari argumentdan qiymat berilishiga qaramasdan funksiyadan chiqilgandan keyin ularga murojaat qilib bo‘lmaydi. Parametrlar funksiya nomidan keyin qavs ichida yoziladi. Parametrlarni e’lon qilish o‘zgaruvchilarni e’lon qilishdan uncha farq qilmaydi. Misol:

// funksiya uzatilgan qiymatni 10 ga ko‘paytiradi va natijani qaytaradi.

```

function x10 ($value) {
$value = $value * 10;
return $value;
}

```

Berilgan funksiya ichida parametr murojaat qilinishiga va har xil amallar bajarilishiga qaramay, funksiyadan chiqqandan keyin parametr qiymati yo‘q qilinadi.

Global o‘zgaruvchilar. Bu o‘zgaruvchilarga dasturning ixtiyoriy nuqtasidan murojaat qilish mumkin. Global o‘zgaruvchi qiymatini o‘zgartirish uchun uni mos funksiyada global deb e’lon qilinishi kerak. Buning uchun o‘zgaruvchi nomining oldiga GLOBAL kalit so‘zi yozib qo‘yiladi. Misol:

```

$somevar = 15;
function addit() {

```

```

GLOBAL $somevar;
$somevar++;
print "Somevar is $somevar";
}
addit();

```

Bu misolda \$somevar ning qiymati 16 ga teng. Agar bu misolda GLOBAL \$somevar satri bo'lmasa \$somevar ga 1 qiymati tayinlanadi. Chunki bu o'zgaruvchi funksiyasi addit() ga nisbatan lokal hisoblanadi. Lokal o'zgaruvchi sukut bilan 0 boshlang'ich qiymatini oladi, so'ngra unga 1 qo'shiladi. Shunday qilib, 1 qiymati chiqariladi. Global o'zgaruvchilarni e'lon qilishning PHP \$GLOBALS() massivini ishlatib e'lon qilish usuli ham bor.

Statik o'zgaruvchilar. O'zgaruvchilarning yana bir toifasi statik deb ataladi. Parametr bilan e'lon qilingan va funksiyadan chiqilganda yo'qoladigan o'zgaruvchilardan farqli ravishda statik o'zgaruvchilar o'z qiymatlarini takror chaqirganda saqlab qoladi. O'zgaruvchini statik deb e'lon qilish uchun uning nomi oldiga STATIC kalit so'zi qo'yiladi. STATIC \$somevar;

Misol:

```

function keep_track() {
STATIC $count=0;
$count++;
print $count;
print "< br >";
}
keep_track();
keep_track();
keep_track();
keep_track();

```

Agar \$count o'zgaruvchisi statik deb e'lon qilinmaganda (ya'ni u lokal bo'lganda) natija quyidagi ko'rinishda bo'ladi:

```

1
1
1

```

Modomiki, \$count o'zgaruvchisi statik bo'lganligi uchun funksiyani har gal chaqirganda uning oldingi qiymati saqlanadi. Shuning uchun natija quyidagicha bo'ladi:

```

1
2
3

```

Toifalarni qayta yaratish (o'tkazish). Ba'zan o'zgaruvchilarni ularni yaratilganda ko'zda tutilmagan holda ishlatish talab etiladi. Masalan, satrli qiymat «15»ga butun 12 sonini qo'shish talab etilsin. PHPda o'zgaruvchi toifalarini oshkor almashtirishni bajarmasdan ham o'zgartirish mumkin. Bu hodisa toifalarni qayta o'zgartirish (o'tkazish) deyiladi.

Toifalarni almashtirish. O'zgaruvchini o'zi aniqlangan toifa dan boshqa toifaga oshkor holda o'tkazish toifalarni almashtirish deyiladi. Toifalarni o'zgartirish vaqtinchalik, bir marotaba yoki doimiy bo'lishi mumkin. O'zgaruvchini boshqa toifaga vaqtinchada o'tkazish uchun toifalarni almashtirish operatorlaridan foydalaniladi. Buning uchun kerakli toifa o'zgaruvchi oldiga qavs ichida qo'yiladi. Toifalarni o'zgartiruvchi operatorlar quyidagi jadvalda keltirilgan.

8.2.2-jadval.

O'zgaruvchilar toifasini almashtirish operatorlari

| Toifani almashtirish operatori | Yangi toifa |
|--------------------------------|-------------|
| (int) yoki (integer) | Butun son |
| (real), (double) yoki (float) | Haqiqiy son |
| (string) | Satr |
| (array) | Massiv |
| (object) | Obyekt |

Misollar:

```
$variable1= 13; // $variable1 ga 13 butun soni tayinlanadi
$variable2 = (double) $variable1; // $variable2 ga 13.0 soni
tayinlanadi
```

\$variable1 o'zgaruvchisi butun 13 soniga teng bo'lsa, uni almashtirish (double) haqiqiy toifaga o'zgartiradi (shuning uchun 13 soni 13.0 soniga aylanadi). Hosil bo'lgan qiymat \$variable2 o'zgaruvchiga tayinlanadi. Yuqorida ma'lumki, butun son haqiqiy songa qo'shilganda natija haqiqiy bo'ladi. Lekin natija toifasini oshkor almashtirish toifasi yordamida o'zgartirish mumkin.

Misol:

```
$variable1 = 4.0;
$variable2 = 5;
$variable3 = (int) $variable1 + $variable2; // $variable3 = 9
```

Haqiqiy toifa butun toifaga almashtirilganda u yaxlitlash bilan bajariladi. Misol:

```
$variable1 = 14.7;
```

```
$variable2 = (int) $variable1; // $variable2 = 14;
```

Satr yoki boshqa o'zgaruvchi toifasini massiv elementiga almashtirish mumkin. Bu holda o'zgartirilgan o'zgaruvchi massivning birinchi elementi bo'lib qoladi. Misol:

```
$variable1 = 1114;
```

```
$array1 = (array) $variable1;
```

```
print $array1[0]; // 1114 qiymat chiqariladi.
```

Nihoyat, ixtiyoriy ma'lumot toifasini obyektga almashtirish mumkin. Bunda o'zgaruvchi obyekt atributi bo'lib qoladi va unga scalar nomi beriladi. Misol:

```
$model = "Toyota";
```

```
$new_obj = (object) $model;
```

Berilgan satr qiymatiga murojaat quyidagi ko'rinishda bo'ladi:

```
print $new_obj->scalar;
```

Tayinlash operatori. PHP tilida o'zgaruvchiga qiymat tayinlash ikki xil bo'lishi mumkin:

O'zgaruvchiga qiymat berib tayinlash.

O'zgaruvchiga murojaat bo'yicha qiymat tayinlash.

Qiymat berib tayinlash eng keng tarqalgan tayinlash usuli. Bunday qiymat o'zgaruvchi bilan aniqlangan xotira sohasiga to'g'ridan to'g'ri yoziladi. Misollar:

```
$vehicle = "car";
```

```
$amount = 10.23;
```

Qiymat berib tayinlash funksiyalarda return komandasini bajarilganda ham amalgalashadi. Misol:

```
function simple () {
```

```
    return 5;
```

```
}
```

```
$return_value = simple();
```

Murojaat bo'yicha o'zgaruvchiga qiymat tayinlashda qiymat olayotgan o'zgaruvchi to'g'ridan to'g'ri qiymat joylashgan xotira sohasiga murojaat ko'rsatkich bilan bog'lanadi. Murojaat bo'yicha qiymat tayinlashda qiymati olinayotgan o'zgaruvchi manbayiga & (ampersent) simvoli qo'yiladi.

```
$dessert = "cake";
```

```

$dessert2 = $Sdessert;
$dessert2 = "cookies";
print "$dessert2 <br>"; // cookies satri chiqarildi
print Sdessert; // cookies satri qaytadan chiqarildi

```

Bu misoldan ko‘rinib turibdiki, \$dessert2 o‘zgaruvchini murojaat bo‘yicha \$dessert o‘zgaruvchi egallagan xotira bog‘lagandan keyin \$dessert2 ixtiyoriy o‘zgarishlari \$dessert avtomatik o‘zgarishiga olib keladi.

O‘zgaruvchi ichidagi o‘zgaruvchilar. Ko‘pgina vaziyatlarda o‘zgaruvchining mazmuni qiymatini boshqa o‘zgaruvchi nomi bilan ishlatish qulay. Misol:

```
$recipe = "spaghetti";
```

Odatda: spaghetti satrini o‘zgaruvchi nomi sifatida talqin qilish mumkin. Buning uchun tayinlash buyrug‘ida berilgan o‘zgaruvchi nomi oldiga ikkinchi \$ belgisi qo‘yiladi. Misol:

```
 $$recipe = "& meatballs";
```

Bu komanda "& meatballs" satrini "spaghetti" nomli o‘zgaruvchiga tayinlaydi. Demak, quyidagi ikkita komanda bir xil natija chiqaradi.

```
 print $recipe $spaghetti;
```

```
 print $recipe $($recipe);
```

ikkala holda ham "spaghetti & meatballs" satri chiqariladi.

Standart o‘zgaruvchilar. PHP tilida qator standart o‘zgaruvchilar ishlatiladi. Ular dasturchiga ichki konfiguratsiya haqida to‘la ma’lumot beradi. Ularning ba’zilarining qiymati PHP tomonidan, boshqalariniki operatsion tizimga bog‘liq ravishda va web serverga bog‘liq ravishda o‘zgaradi.

Konstantalar. Dasturni boshqarish jarayonida o‘zgarmaydigan nomlangan kattalik konstanta deyiladi. Konstantalar doimiy kattaliklar bilan, masalan, π soni bilan ishlash qulay (3,141592). PHPda konstantalar define() funksiyasi bilan aniqlanadi. Dasturda konstanta aniqlangandan so‘ng dasturda uni o‘zgartirib yoki qayta aniqlab bo‘lmaydi. Masalan, PHP ssenariysida π konstantani aniqlash quyidagi ko‘rinishda bo‘ladi:

```
define("PI", "3.141592");
```

Aniqlangan konstantani dasturda ishlatish mumkin.

```
print "The value of pi is". PI."<br>";
```

```
$pi2 = 2 * PI;
```

```
print "Pi doubled equals $pi2.";
```

Bu dastur fragmentini ishslash natijasi quyidagicha:

The value of pi is 3.141592.

Pi doubled equals 6.283184.

Bu yerda ikkita narsaga e'tibor berish kerak. Birinchidan, konstanta nomlarida \$ belgisi ko'rsatilmaydi. Ikkinchidan, konsstantani takomillashtirib (masalan, unga 2π) bo'lmaydi. Agar konstanta hisoblashlarda ishlatilsa natijani boshqa o'zgaruvchida saqlashga to'g'ri keladi.

8.3. Ifodalar, operatorlar va boshqaruvchi konstruksiyalar

Ixtiyoriy dasturlash tillarida ifodalar, operatorlar (amallar) va boshqarish konstruksiyalari muhim o'rinni egallaydi.

Ifodalar. Ifodalar dasturda bajariladigan amallarni tavsiflaydi. Har qanday ifoda hech bo'lmaganda bitta operanddan va bir qancha operatordan tashkil topadi.

Operand – bu dasturda qayta ishlanadigan birorta kattalik hisoblanadi. Misollar:

`$a++; // $a – operand;`

`$sum = $val1 + $val2; // $sum, $val1 va $val2 – operandlar.`

Operatorlar – ifodada operandlar ustida biror amal bajarishni ko'rsatuvchi simvolli belgilari. PHP tilida ikkita operandni birlashtiruvchi operator toifasi asosida, toifalarni avtomatik almashtrishni bajaradi. Jadvalda imtiyozlarni pasayib borishi tartibida joylashgan operatorlarning to'la ro'yxati keltirilgan.

8.3.1-jadval.

PHP operatorlari

| Operator | Assosiativlik | Maqsad |
|----------|---------------|--|
| () | - | Ustunlikni o'zgartirish |
| new | - | Obyekt nuxxalarini yaratish |
| ! ~ | Π | Mantiqiy inkor, razryadma razryad inkor |
| ++ -- | Π | inkrement, dekrement |
| @ | Π | Xatolarni maskirovka qilish |
| / * % | Л | Bo'lish, ko'paytirish, qoldiq |
| + - . | Л | Qo'shish, ayirish, konkatenatsiya |
| << >> | Л | Chapga surish, o'ngga surish (razryadma razryadli) |

| | | |
|---------------|---|--|
| < <= > >= | - | kichik, kichik yoki teng, katta, katta yoki teng |
| == != === <> | - | teng, teng emas, aynan teng (birdaylik), teng emas |
| & ^ | Л | AND, XOR va OR razryadma razryad amallari |
| && | Л | AND va OR mantiqiy amallari |
| ?: | П | Тернар operatori |
| = += *= /= .= | П | Tayinlash operatorlari |
| AND XOR OR | Л | AND, XOR va OR mantiqiy amallari |

Bu yerda, mos ravishda Π – o'ng, Λ – chap assosiativlikni bildiradi.

Operatorlar va operandlar konsepsiysi bilan tanishib bo'lgandan keyin, quyidagi ifodalarda keltirilgan misollar yetarli va tushunarli bo'ladi:

Misollar.

`$a = 5; // $a o'zgaruvchiga 5 butun sonni tayinlash;`

`$a = "5"; // $a o'zgaruvchiga "5" satrli kattalikni tayinlash;`

`$sum + 50 $some_int; // Yig'indi 50 + $some_int o'zgaruvchi $sum tayinlash;`

`Swine ="Zinfandel"; // "Zinfandel" satrini o'zgaruvchi $wine tayinlash;`

`$inventory++; // $inventory ning qiymatini 1 oshirish.`

Operator va operandlarni birlashtirib, hisoblashlarni bajarish uchun murakkab ifodalar hosil qilish mumkin. Misol:

`$total_cost = $cqst + (Scost * 0.06); // narxga 6-protsentli soliqni qo'shish.`

Operatorlarning ustunligi maktab matematika kursida o'tilgan tartibga bo'yusunadi. Misollar:

`$total_cost = $cost + $cost * 0.06;`

Bu buyruq quyidagi buyruqqa ekvivalent:

`$total_cost = $cost + ($cost * 0.06).`

Operatorlar assotsiativligi. U bir xil ustunli operatorlar ketma-ketligini aniqlaydi (8.3.1-jadvalga qarang).

`$value = 3*4*5*7*2;`

Bajarish ikki yo'nali shda amalga oshirilishi mumkin: chapdan o'ngga yoki o'ngdan chapga. Birinchi toifa amallarini assotsiativ-

ligida, ifodaga kiruvchi amallar chapdan o‘ngga qarab bajariladi. Masalan, komanda

$\$value = 3 * 4 * 5 * 7 * 2;$

Quyidagi komandaga ekvivalent

$\$value = (((3 * 4) * 5) * 7) * 2;$

Hisoblash natijasi 840 ga teng. Ko‘paytirish operatori (*) chap tomonli assotsiativlikka ega.

O‘ng tomonli assotsiativlik va bir xil ustunli operatorlar o‘ngdan chapga ishlov beriladi. Masalan:

fragment

$\$c = 5;$

$\$value = \$a - \$b - \$c;$

Ushbu fragmentga ekvivalent

$\$c = 5;$

$\$value = (\$a - (\$b - \$c));$

Bu ifodaga ishlov berganda \$value, \$a, \$b i \$c o‘zgaruvchilarga 5 qiymati tayinlanadi. Tayinlash operatori (=) o‘ng tomonli assotsiativlikka ega. Matematik operatorlar har xil matematik amallarni bajarishga mo‘ljallangan va ko‘pgina PHP dasturlarida ishlatiladi (8.3.2-jadval).

8.3.2-jadval.

Matematik operatorlar

| Misol | Nomi | Natija |
|--------------|--------------|-------------------------------|
| $\$a + \b | Qo‘sish | \$a va \$b Qo‘sish |
| $\$a - \b | Ayirish | \$a va \$b Ayirmasi |
| $\$a * \b | Ko‘paytirish | \$a va \$b Ko‘paytmasi |
| $\$a / \b | Bo‘lish | \$a ni \$b bo‘lish (bo‘linma) |
| $\$a \% \b | Qoldiq | \$a ni \$b bo‘lish qoldig‘i |

PHPda juda keng turkumdagи standart matematik funksiyalar kutubxonasi berilgan.

Tayinlash operatori. Bu operatorlar o‘zgaruvchining yangi qiymatini tayinlaydi. Oddiy holda tayinlash operatori kattalik qiymatini o‘zgartirish bilan cheklanadi, boshqa hollarda tayinlash oldidan birorta amal bajariladi. Bu operatorlarga misol jadvalda keltirilgan.

8.3.3-jadval.

Tayinlash operatori

| Misol | Nomi | Natija |
|------------------------|----------------------------|--|
| <code>\$a = 5;</code> | Tayinlash | <code>\$a o'zgaruvchi 5 ga teng</code> |
| <code>\$a += 5;</code> | Qo'shib tayinlash | <code>\$a o'zgaruvchi \$a va 5 ning yig'indisiga teng</code> |
| <code>\$a *= 5;</code> | Ko'paytirib tayinlash | <code>\$a o'zgaruvchi \$a va 5 ning ko'paytmasiga teng</code> |
| <code>\$a/=5;</code> | Bo'lib tayinlash | <code>\$a o'zgaruvchi \$a ni 5 ning bo'linmasiga teng</code> |
| <code>\$a .= 5;</code> | Konkatenatsiyali tayinlash | <code>\$a o'zgaruvchi \$a va 5 ning konkatenatsiyasiga teng</code> |

Satrli operatorlar. PHPda satrli operatorlar konkatenatsiyani qulay vositalar bilan ta'minlaydi. Konkatenatsiya deb ikkita va undan ortiq obyektlarni birlashishi (ulanishi) tushuniladi. Satrli operatorlarning ikki xili mavjud:

- Konkatenatsiya operatori (.)
- Tayinlashli konkatenatsiya operatori (q)

8.3.4-jadval.

Satr operatorlari

| Misol | Nomi | Natija |
|--------------------------------|----------------------------|---|
| <code>\$a = "abc"."def"</code> | Konkatenatsiyali | <code>\$a o'zgaruvchiga \$a va \$b konkatenatsiya natijasi tayinlanadi</code> |
| <code>\$a .= "ghijkl"</code> | Konkatenatsiyali tayinlash | <code>\$a o'zgaruvchini joriy qiymatiga "ghijkl" satrining konkatenatsiya natijasi tayinlanadi</code> |

Satrli operatorlarni ishlatishga doir misol:

```
// $a ga satrli qiymat "Spaghetti & Meatballs" tayinlanadi $a
= "Spaghetti" . "& Meatballs"
// $a ga satrli qiymat "Spaghetti & Meatballs are delicious"
tayinlanadi $a .= "are delicious";
```

Inkrement va dekrement operatorlari. Bu operatorlar o'zgaruvchining qiymatini 1 ga oshiradi (inkrement(++) va kamaytiradi (dekrement (--)), ya'ni 1 ga orttirish va kamaytirish uchun qulay ko'rinishni ta'minlaydi.

8.3.5-jadval.

Inkrement va dekrement operatorlari

| Misol | Nomi | Natija |
|---------------------------|-----------|-------------------------------|
| <code>++\$a, \$a++</code> | Inkrement | \$a o'zgaruvchi 1 ga ortadi |
| <code>--\$a, \$a--</code> | Dekrement | \$a o'zgaruvchi 1 ga kamayadi |

Bu operatorlar operanddan chap tomonda ham o'ng tomonda ham joylashishi mumkin. Quyidagi misolni ko'rib chiqamiz:

`$inventory = 15; // Sinventory ga butun son 15 tayinlansin`

`$old_inv = $inventory--; // AVVAL $old_inv Sinventory qiy-mati tayinlansin. // so'ngra Sinventory kamaytirilsin.`

`$orig_inventory = ++$inventory; // AVVAL Sinventory ort-tirilsin. So'ngra // orttirilgan qiymatni Sinventory`

`/ $orig_inventory o'zgaruvchiga tayinlansin.`

Mantiqiy operatorlar. Bu operatorlar matematik operatorlar bilan birga PHP ilovalarida muhim o'rinni egallaydi. Mantiqiy operatorlar dasturdagi komandalarni bajarish tartibini boshqaradi (8.3.6-jadval).

8.3.6-jadval.

Mantiqiy operatorlar

| Misol | Nomi | Natija |
|---------------------------------|-------------------------|---|
| <code>\$a && \$b</code> | Konyunksiya | Rost, agar ikkala operand rost bo'lsa |
| <code>\$aAND\$b</code> | Konyunksiya | Rost, agar ikkala operand rost bo'lsa |
| <code>\$a И \$b</code> | Dizyunksiya | Rost, agar hech bo'lmaganda operandlardan bittasi rost bo'lsa |
| <code>\$a OR \$b</code> | Dizyunksiya | Rost, agar hech bo'lmaganda operandlardan bittasi rost bo'lsa |
| <code>!\$a</code> | Inkor | Rost, agar \$a qiymati yolg'on bo'lsa |
| <code>NOT !\$a</code> | Inkor | Rost, agar \$a qiymati yolg'on bo'lsa |
| <code>\$a XOR \$b</code> | Dizyunksiya mus-tasnosi | Rost, agar operandlardan faqat bittasi rost bo'lsa |

Misollar:

`file_exists("filename.txt") OR print "File does not exist!";`
ikki variantdan bittasi bo'lishi mumkin:

fayl filename.txt mavjud;

"File does not exist!" xabar chiqariladi.

Tenglik operatorlari. Bu operatorlar ikkita kattalikni solishtirish va ularning ekvivalentligini tekshirish uchun mo‘ljallangan 8.3.7-jadval.

8.3.7-jadval.

Tenglik operatorlari

| Misol | Nomi | Natija |
|---------------|--------------------------------|---|
| $\$a == \b | Tenglikni tekshirish | Rost, agar \$a va \$b teng bo‘lsa |
| $\$a != \b | Tengsizlikni tekshirish | Rost, agar \$a va \$b teng bo‘lmasa |
| $\$a === \b | Aynan o‘xshashlikni tekshirish | Rost, agar \$a va \$b teng bo‘lsa va bir xil toifaga ega bo‘lsa |

Munosabat operatorlari. Bu operatorlar ham, mantiqiy operatorlar kabi dasturning ishlash mantig‘ini boshqaradi hamda ikkita va undan ortiq o‘zgaruvchilarni solishtirib qaror qabul qilishga mo‘ljallangan. Munosabat operatorlari 8.3.8-jadvalda keltirilgan.

8.3.8-jadval.

Solishtirish operatorlari

| Misol | Nomi | Natija |
|-----------------------|------------------|---|
| $\$a < \b | Kichik | Rost, agar \$a o‘zgaruvchi \$b dan kichik bo‘lsa |
| $\$a > \b | Katta | Rost, agar \$a o‘zgaruvchi \$b dan katta bo‘lsa |
| $\$a <= \b | Kichik yoki teng | Rost, agar \$a o‘zgaruvchi \$b dan kichik yoki teng bo‘lsa |
| $\$a >= \b | Katta yoki teng | Rost, agar \$a o‘zgaruvchi \$b dan katta yoki teng bo‘lsa |
| $(\$a - 12) ? 5 : -1$ | Terhap operatori | Agar o‘zgaruvchi \$a 12 teng bo‘lsa, 5 agar teng bo‘lmasa 1 qiymati qaytariladi |

Razryadma razryad bajarish operatorlari. Bu operatorlar butun sonlar ustida, sonlarni tashkil qiluvchi alohida bitlari satrida amal bajarishga mo‘ljallangan. Bu operatorlar 8.3.9 va 8.3.10-jadvallarda keltirilgan.

8.3.9-jadval.

O‘nlik son va ularning ikkilik ko‘rinishi

| O‘nlik son | Ikkilik ko‘rinishi |
|------------|-----------------------|
| 2 | 10 |
| 5 | 101 |
| 10 | 1010 |
| 12 | 1100 |
| 145 | 10010001 |
| 1 452 012 | 101100010011111101100 |

8.3.10-jadval.

Razryadma razryad bajarish operatorlari

| Misol | Nomi | Natija |
|--------------------|------------------------|--|
| $\$a\&\b | Konyunksiya | $\$a$ va $\$b$ ni bir xil razryadlarida joylashgan bitlari bilan, konyunksiya amali bajariladi |
| $\$a \b | Dizyunksiya | $\$a$ va $\$b$ ni bir xil razryadlarida joylashgan bitlari bilan, dizyunksiya amali bajariladi |
| $\$a^{\wedge} \b | Dizyunksiya mustasnosи | $\$a$ va $\$b$ ni bir xil razryadlarida joylashgan bitlari bilan, dizyunksiya mustasnosи amali, bajariladi |
| $\sim \$b$ | Inkor | $\$b$ o‘zgaruvchining barcha razryadlari teskariga (invertasiya) almashtiriladi |
| $\$a << \b | Chapga surish | $\$a$ o‘zgaruvchiga $\$b$ qiymati ikki bitiga chap tomonga surilib tayinlanadi |
| $\$a >> \b | O‘ngga surish | $\$a$ o‘zgaruvchiga $\$b$ qiymati ikki bitiga o‘ng tomonga surilib tayinlanadi |

Boshqarish konstruksiyalari. Boshqarish konstruksiyalari dasturchi ixtiyoriga murakkab dasturlar qurish uchun vositalar beradi. Bunday dasturlar shartlarni tekshirish va kiritilayotgan ma’lumotlar o‘zgarishida ta’sirlanishni tekshirish imkonini beradi. Qisqasi bu tuzilmalar dastur bajarilishini boshqaradi.

Shartlarni tekshirish. Boshqarish konstruksiyalari odatda shartlarni rost yoki yolg‘onligini tekshiradi va natijaga bog‘liq ravishda u yoki bu amalni bajaradi. Masalan, quyidagi ifodani ko‘ramiz: $\$a == \b . Bu ifoda agar $\$a$ teng bo‘lsa $\$b$ rost, aks holda yolg‘on hisoblanadi. Rost ifodaning natijasi 1 ga, yolg‘on

ifodaning natijasi 0 teng deb hisoblanadi. Quyidagi dastur fragmentini ko'rib chiqamiz:

```
$a = 5;  
$b = 5;  
print $a == $b;
```

Natijada 1 qiymati chiqariladi. Agar \$a yoki \$b ni 5 dan farqli qiymatga o'zgartirsak 0 qiymati chiqariladi.

IF buyrug'i. Bu buyruq tanlash buyrug'ining ko'rinishi bo'lib, u ifoda qiymatini hisoblab, uni rost yoki yolg'onligiga qarab dastur kodi blokini bajaradi. Uning 2 ta ko'rinishi (qisqa va to'liq) ishlataladi.

Qisqa ko'rinishining umumiy formati

```
if (ifoda) {
```

```
blok
```

```
}
```

Va to'liq ko'rinishining umumiy formati

```
if (ifoda) {
```

```
blok
```

```
}
```

```
else {
```

```
blok
```

```
}
```

Blokda bir yoki bir nechta buyruq joylashtirilishi mumkin.

Quyidagi misolda

shart tekshirilib 2 ta tasdiqdan biri chiqadi:

```
if ($cooking_weight < 200) {
```

```
print "This is enough pasta (< 200g) for 1-2 people";
```

```
}
```

```
else {
```

```
print "That's a lot of pasta. Having a party perhaps?";
```

```
}
```

Agar blok bitta buyruqdan iborat bo'lsa figurali qavslarning bo'lishi majburiy emas.

if (\$cooking_weight < 100) print «Are you sure this is enough?»;

ELSEIF buyrug'i. Bu buyruq IFni boshqarish tuzilmalariga qo'shimcha tekshirish satrini kiritadi va qaror qilish shartlari sonini oshiradi. Uning ko'rinishi (formati) quyidagicha:

```
if (ifoda) {
```

```

blok
}
elseif (ifoda) {
blok
}
Misol:
if ($cooking_weight < 200) {
print "This is enough pasta (< 200g) for 1-2 people";
}
elseif ($cooking_weight < 500) {
print "That's a lot of pasta. Having a party perhaps?"; }
}
else {
print "Whoa! Who are you cooking for, a football team?"; }
}

```

Shu bilan birga ta'kidlab o'tish kerak-ki, IF buyrug'ini ichida boshqa IF buyrug'ini joylashtirib bajarish mumkin. Bunda shartni tekshirishda maksimal nazorat qilish imkoniyati paydo bo'ladi. Ko'pgina masalalarni yechishda bir nechta shartlarni hisoblash va tekshirishga to'g'ri keladi. Bunday hollarda shartlarni arifmetik va mantiqiy operatorlar yordamida birlashtirib, murakkab shartlar tuzish mumkin.

Bloklarda alternativ cheklashlar. Boshqarish tuzilmalarida, blokning chegaralarini aniqlovchi ({ } – cheklovchidan tashqari) maxsus cheklovchilar ishlataladi. Dasturchiga qulaylik yaratish maqsadida PHP tili bloklarni cheklovchi alternativ format bilan ta'minlangan. Misol:

```

if (ifoda) :
blok
else :
blok
endif;
Demak, quyida keltirilgan 2 ta IF buyrug'i to'la ekvivalent:
if ($a== $b) {
print "Equivalent values!";
}
if ($a == $b) :
print "Equivalent values!";

```

endif;

Sikl buyruqlari. WHILE sikl (buyrug‘i). Bu tuzilma buyruqlar blokini ko‘p marta takroran (siklik) ravishda bajarishga mo‘l-jallangan. WHILE siklining umumiy formasi (formati) quyidagicha:

while (ifoda):

blok

endwhile;

WHILE siklining ishlashini ($n!$) faktorialni hisoblash das-turi misoldida ko‘rib chiqamiz. Bu yerda, $n=5$.

$\$n = 5;$

$\$nsoru = \$n;$

$\$factorial = 1; //$ Faktorialning boshlang‘ich qiymatini o‘rnatish

WHILE ($\$n > 0$):

$\$factorial - \$n * \$factorial;$

$\$n--;$ // $\$n$ ni 1 ga kamaytirish

end WHILE;

print "The factorial of $\$ncopy$ is $\$factorial.$ ";

Dastur quyidagi natijani chiqaradi:

The factorial of 5 is 120.

Bu misolda $\$n$ har iteratsiya oxirida kamayadi. Sikl sharti $\$n$ 0 ga teng bo‘lganda shart rost bo‘lmay qoladi ($0 > 0$ shart yolg‘on)) va sikl bajarilishi yakunlanadi.

DO ...WHILE sikli (buyrug‘i). Bu sikl WHILE sikli kabi ish-laydi. Faqat,

DO ...WHILE siklida shart har bir iteratsiyani boshida emas, oxirida tekshiradi. Bundan tashqari DO ...WHILE sikli har doim hech bo‘lmasganda bir marotaba bajariladi. WHILE sikli esa shart yolg‘on bo‘lganda biror marotaba ham bajarilmaydi. Bu operatorning formati quyidagicha:

do:

blok

WHILE (ifoda);

Misol :

Yuqorida faktorialni hisoblashni DO ...WHILE sikli bilan ba-jaramiz:

$\$n = 5;$

```

$ncopy = $n;
$factorial = 1; // Faktorialning boshlang'ich qiymatini
o'rnatish
do {
    $factorial = $n * $factorial;
    $n-- // Sn ni 1 kamaytirish
} WHILE ($n > 0);
print "The factorial of Sncopy is $factorial.";

```

FOR sikli. Bu sikl WHILE siklidan shu bilan farq qiladiki, shart blok ichida emas, boshqaruvchi konstruksiyaning o'zida o'zgartiriladi. FOR sikli ham shart rost bo'lib qolgunga qadar davom ettiriladi. Uni umumiy ko'rinishi quyidagicha:

for (shartni initsializatsiyalash (boshlang'ich qiymati): shart; orttirma)

```

{
blok
}
Endfor;
```

FOR siklining shartli qismi uchta elementdan tashkil topadi. Initsializatsiya (boshlang'ich qiymat berish) bir marta bajarilib sikl boshqaruvchi o'zgaruvchisining boshlang'ich qiymatini aniqlaydi. Shart har bir iteratsiyani boshida tekshiriladi va joriy siklini bajarilishi yoki bajarilmasligini aniqlab beradi. Orttirma sikl o'zgaruvchisini har bir iteratsiyada qanchaga o'zgarishini aniqlab beradi. Orttirma dasturchining maqsadiga qarab ko'payishi yoki kamayishi mumkin. Misol:

for (\$i = 10; \$1 <- 100: \$1 +=10) : // teskari chiziq o'zgarivchi
\$1 bo'lishi mumkin print "\ \$i = \$i
"; endfor; // interpolatsiyadan oldini oladi.

Dasturning bu fragmenti quyidagi natijani beradi:

```

$i = 10
$i = 20
$i = 30
$i = 40
$i = 50
$i = 60
$i = 70
$i = 80
$i = 90
```

$\$i = 100$

Bu misolda siklni boshqaruvchi o'zgaruvchi $\$i$ 10 ga teng boshlang'ich qiymatini oladi. Shart $\$i$ qiymati 100 dan katta bo'lguncha davom ettiriladi. Siklning har bir takrorlanishidan keyin $\$i$ qiymati 10 ga ortadi. $\$i$ qiymatini 10 ga oshirish uchun qo'shishli tayinlash operatori ishlataladi, chunki PHP tilida FOR siklida an'anaviy yozuv $\$i = \$i+10$ ni ta'minlamaydi. Keltirilgan misolni yechishni boshqacha bajarsa ham bo'ladi:

```
for ($i = 10; $i <=100; print "\$i - $i <br>". $i+=10);
```

Bunda ham bir xil natijaga ega bo'lamiz.

FOR sikli bir nechta maxsus xususiyatga ega. Masalan, bir nechta o'zgaruvchini bir vaqtida initsializatsiya qilish mumkin (ularni bir-biridan vergul bilan ajratib). Misol:

```
for ($x=0,$y=0: $x+$y<10; $x++) :  
    $y += 2;          // $u ni 2 orttirish  
    print "\$y = $y <Bk>"; // $u qiymatini chiqarish  
    $sum = $x + $y;  
    print "\$sum = $sum<BR>"; \\ $sum qiymatini chiqarish  
endfor;  
Natija  
$y = 2  
$sum = 2  
Sy = 4  
$sum = 5  
$y = 6  
$sum = 8  
$y = 8  
$sum = 11
```

Bu misol \$y joriy qiymati bilan \$x va \$y yig'indisini hisoblab chiqaradi.

Yig'indi ($\$x + \$y <10$) sikl chegarasidan chiqib ketgan bo'lsa ham, natijada $\$sum = 11$ qiymati chiqariladi. Bunga sabab siklga kirishda o'zgaruvchi $\$y$ teng 6 ga, $\$x$ teng 2 ga. O'zgaruvchining qiymatlari sikl shartiga mos keladi, shuning uchun $\$x$ va $\$y$ ga yangi qiymatlar tayinlanadi va 11 ga teng yig'indi natija sifatida chiqariladi. Keyingi tekshirishda shartli yig'indi 11 ga teng bo'lib, u 10 ga teng yakuniy qiymatdan oshib ketadi va sikl tugatiladi. FOR siklini boshqarish ifodasida ixtiyoriy komponentlar bo'lmasligi mumkin. Masalan, oldin initsializatsiya qilingan o'zgaruvchini

to‘g‘ridan to‘g‘ri aniq boshlang‘ich qiymat tayinlamasdan siklga uzatish mumkin, boshqa vaziyatlar ham bo‘lishi mumkin. Jumladan, sikl o‘zgaruvchisining orttirmasi siklda aniqlanadigan birorta shartga bog‘liq ravishda amalga oshirilishi mumkin. Bu holda boshqarish ifodasida orttirma ko‘rsatilmasligi kerak. Misol:

```
$x = 5;  
for (: : $x +=2):  
    print "$x ";  
    if ($x == 15):  
        break; // for siklidan chiqish  
    endif;  
endfor;
```

Natija quyidagi ko‘rinishda bo‘ladi:

```
5 7 9 11 13 15
```

FOR va WHILE sikllarining funksiyalari bir xil bo‘lsa ham, FOR sikli dasturni tushunarliroq qiladi. Buni FOR buyrug‘ini dasturchi ko‘rganda siklning bajarilish davrini va bajarilishi haqidagi zaruriy axborotni darrov ko‘radi. Ikkinci tomondan WHILE siklida boshqaruvchi o‘zgaruvchini qidirib tiklash uchun ortiqcha vaqt sarf qiladi. Katta dasturlarda bu ko‘p vaqt ni olishi mumkin.

FOREACH buyrug‘i. FOREACH tuzilmasi FOR tuzilmasining ko‘rinishi bo‘lib, massiv elementlarini tanlab chiqishni yengillashtirish maqsadida kiritilgan. FOREACHning ikki ko‘rinishi mavjud. Ular massivlarni har xil toifalari bilan ishlashga mo‘ljallangan. Uning formati quyidagicha:

```
foreach (massiv as $element) {  
    blok  
}  
foreach (massiv as $klyuch => $element) {  
    blok  
}
```

Masalan, quyidagi fragment bajarilganda:

```
$menu = arrau("pasta", "steak", "potatoes", "fish", "fries");  
foreach ($menu as $item) {  
    print "$item <BR>";  
}  
quyidagi natija chiqadi:  
pasta
```

steak
potatoes
fish
fries

Bu misolda ikkita holga e'tibor berish kerak. Birinchidan, FOREACH tuzilmasi massivni boshiga avtomatik qaytadi (boshqa siklik tuzilmalarda bunday emas). Ikkinchidan, hisoblagichni (schotchkik) oshkor ravishda oshirish yoki boshqa usulda keyingi massiv elementiga o'tishga zarurat yo'q. Bu FOREACH buyrug'inining har bir takrorlanishidan keyin avtomatik ravishda bajariladi. Bu siklning ikkinchi ko'rinishi assotsiativ massivlar bilan ishslashga mo'ljallangan.

SWITCH buyrug'i. Bu buyruqni ishlashi IF buyrug'inining ishslashini eslatadi. Ifodani hisoblash natijasi ro'yxatdagi qiymatlar bilan solishtiriladi. Bu ko'pincha bir nechta qiymatlarni tekshirishda qulay, chunki switch ni ishlatish dasturni ixcham va ravshan qiladi. Uning umumiy formati quyidagicha:

```
switch (ifoda) {  
    case (shart):  
        blok  
    case (shart):  
        blok  
    ...  
    default:  
        blok  
}
```

Tekshirilayotgan shart switch so'zidan keyin dumaloq qavs ichida ko'rsatiladi.

Uning hisoblangan natijasi ketma-ket case seksiyasi shartlari bilan solishtiriladi. Mos seksiya topilganda, shu seksiyaga tegishli buyruqlar bloki bajariladi. Agar moslik aniqlanmasa, majburiy bo'limgan default seksiya bloki bajariladi. Masalan, faraz qiling, dastur bir necha variantli yoyiluvchi ro'yxatni akslantirsin va ro'yxatning har bir satrni alohida case seksiyasi tomonidan bajariladigan buyruqqa mos kelsin. Buni switch buyrug'ini ishlatib amalga oshirish qulay. Misol:

```
$user_input = "recipes"; // Foydalanuvchi tanlagan buyruq  
switch ($user_input) :  
    case("search") :
```

```

print "Let's perform a search!";
break;
case("dictionary"):
print "What word would you like to look up?";
break;
case("recipes"):
print "Here is a list of recipes...";
break;
default:
print "Here is the menu...";
break;
endswitch;

```

Bu fragmentdan ko'rning turibdiki, switch dastur kodi aniq va yaqqol bo'lishini ta'minlaydi. Switch shartida ko'rsatilgan o'zgaruvchi (bizni misolda \$user_input) keyingi barcha case seksiyalardagi shartlar bilan solishtiriladi. Agar case seksiyada ko'rsatilgan qiymat solishtirayotgan o'zgaruvchi qiymatiga mos kelsa (bir xil bo'lsa), shu seksiya bloki bajariladi. Break komandasи qolgan case seksiyalarni tekshirishni to'xtatadi va switch tuzilmasini bajarishni tugatadi. Agar tekshirilayotgan shartlardan birortasi ham bajarilmasa, majburiy bo'limgan default seksiya bloki aktivlashadi. Agar default seksiyasi bo'lmasa va shartlardan birortasi ham bajarilmasa, switch buyrug'i ishini to'xtatadi va dasturning bajarilishi keyingi buyruqdan boshlab davom ettiriladi. Case seksiyasida break buyrug'i uchramasa, switch bajarilishi keyingi case blokidan davom ettiriladi, toki break buyrug'i uchramaguncha yoki switch tuzilmasi oxiriga yetmaguncha (chiqmaguncha). Quyidagi misol break buyrug'ini ishlatmaslik oqibatini namoyish qiladi.

Misol:

```

$value = 0.4;
switch($value):
case (0.4):
print "value is 0.4<br>";
case (0.6):
print "value is 0.6<br>";
break;
case (0.3):
print "value is 0.3<br>";

```

```

break;
default:
print "You didn't choose a value!";
break;
endswitch;
Natija quyidagi ko'rinishda bo'ladi:
value is 0.4
value is 0.6

```

Break buyrug‘ining bo‘lmasligi, print buyrug‘ini moslik to-pilgan seksiya blokinigina emas, balki boshqa case seksiyasida ham bajarilishiga olib keldi. So‘ngra switch tuzilmasi buyruqlari-ning bajarilishi ikkinchi print buyrug‘idan keyingi break koman-dasi bilan to‘xtatiladi.

BREAK buyrug‘i. Bu buyruq WHILE, FOR yoki SWITCH tuzilmalarida uchraganda, mos dasturni ishlashini bevosita to‘xtatadi. Uning sintaksisi quyidagicha:

BREAK n

n majburiy bo‘limgan parametri break buyrug‘i bilan tu-gaydigan boshqaruvchi tuzilmalar satrlarining sonini aniqlaydi. Masalan, break buyrug‘i ikkita WHILE buyrug‘ini ichida joy-lashgan bo‘lsa va break 2 bajarilsa (break keyin 2 soni turgan bo‘lsa) unda darrov ikkala sikldan ham chiqish bajariladi. Su-kut bilan n 1 teng. Bitta yuqori sathga chiqishni oshkor holda 1 ko‘rsatib yoki parametrsiz break buyrug‘ini bajarib amalga oshi-rish mumkin. Break if buyrug‘ida ishlatilmaydi. Foreach siklida break buyrug‘ini ishlatishga doir misol:

```

$arr =array(14, 12, 128, 34, 5);
$magic numBer = 128:
foreach ($arr as $val):
if ($val == $magic_number):
print "The magic number is in the array!";
break;
endif;
print "val is $val <br>";
endforeach;

```

Agar \$magic_number qiymati \$arr massivda bo‘lsa qidirish to‘xtatiladi. Natija quyidagi ko'rinishda bo'ladi:

```

val is 14
val is 12

```

The magic number is in the array!

CONTINUE buyrug‘i. Bu buyruq bajarilganda joriy iteratsiyada qolgan barcha buyruqlar o’tkazib yuboriladi va darhol yangi iteratsiya boshlanadi. Uning sintaksisi quyidagicha:

```
continue n;
```

Majburiy bo’lмаган n parametri continue ta’sir doirasini nechta tashqi sathlarigacha yetishini ko’rsatadi. Misol: 0 bilan berilgan chegara intervalida joylashgan barcha tub sonlarni hisoblash talab etilsin. Bizning ixtiyorimizda sonning tub ekanligi yoki emasligini aniqlaydigan is_prime() funksiyasi bo’lsin.

Misol:

```
$boundary =558;  
for ($i = 0; $i <= $boundary: $i++):  
if (! is_prime($i)):  
continue;  
endif;  
$prime_counter++;  
endfor;
```

Agar tekshirilayotgan son tub bo’lsa, if blok aylanib o’tiladi va o’zgaruvchi \$prime_counter qiymati birga oshadi. Aks holda continue buyrug‘i bajariladi va darhol keyingi iteratsiyaga o’tiladi.

Funksiyalar. Funksiyalar yordamida ko’p marotaba ishlatish komponentlarini yaratish mumkin. Zarurat bo’lganda ular yengil o’zgartiriladi. Funksiyalar Web ilovalarni yaratishda qulay vositalardan biri hisoblanadi. Funksional dasturlashda (funksiyalar yordamida) qisqa, ixcham va o’qishga qulay dasturlar yaratishga yordam beradi. Funksiyalar standart va foydalanuvchilar yaratgan funksiyalarga bo’linadi. Standart funksiyalar yangi ilovalar yaratishda ancha vaqt ni tejash imkoniyatini beradi.

Funksiya — yagona nomga ega bo’lib, aniq masalalarni yechishga mo’ljallangan dastur kodi hisoblanadi. Ular ilovani har xil nuqtalaridan nomi bilan chaqiriladi. Masalalar yechishga funksiyalar yordamida yondashishning asosiy afzalligi, funksiyalarni dastur kodi bir marta yozilishi va zarurat bo’lganda ko’p marta yengil takomillashtira olishdir. PHPda yangi funksiyani aniqlash murakkab emas. Funksiyaning PHP dasturini (ilovasini) ixtiyoriy nuqtasida joylashtirish mumkin, lekin dastur kodining tuzilishini tashkil qilish nuqtayi nazaridan, ssenariyada ishlatila-

digan barcha funksiyalarni ssenariya faylining boshiga joylashtirish maqsadga muvofiq hisoblanadi.

Dasturlash samaradorligini oshirish va kodini bir necha marta qayta ishlatish imkonini berishning yana bir usuli – funksiyalarni alohida faylga (biblioteka) ajratish hisoblanadi.

Funksiyani aniqlash odatda uchta qismdan tashkil topadi:

- funksiya nomi;

- qavslar ichida zarur bo‘lmagan kiruvchi parametrlar bir-biridan vergul bilan ajratilib sanab o‘tiladi;

- figurali qavs ichiga olingan funksiya tanasi.

PHP tilida funksianing berilish formati quyidagicha:

```
function funksiya_nomi ([&parametr1. $parametr2, ....  
$parametrn]) {  
    funksiya jismi  
}
```

Funksianing nomi identifikatorlar qo‘yadigan talablarga javob berishi kerak. Funksiya nomidan keyin albatta dumaloq qavslar ichida majburiy bo‘lmagan kirish parametrlarining (\$parametr1, \$parametr2, \$parametrn) ro‘yxati keltiriladi. Odatda PHP tilida kirish parametrlarining toifasini ko‘rsatish zarur emas. PHPda argumentlarni ularning qaysi toifalariga mosligi tekshirilmaydi. Lekin shunga e’tibor berish kerak-ki, kirish parametrlaridagi tasodifiy xatoliklari kutilmagan hollarga olib kelishi mumkin. (Parametrni qaysi toifaga tegishli ekanligini aniqlash uchun gettype() standart funksiyasidan foydalanish tavsija etiladi). Parametr qavslari yopilgandan keyin, figurali qavs ichida shu funksiyaga tegishli funksiya tanasi (kodi) yoziladi. Misol:

Web sahifada litsenzion axborotni chiqarish uchun funksiya yaratamiz.

```
function display_copyright() {  
    print "Copyright © 2001 PHP-Powered Recipes. All  
    Rights Reserved.";  
}
```

Agar sizning Web sahifangiz bir necha betdan iborat bo‘lsa, bu funksiyani har bir betni oxirida chiqarish yetarli va shu bilan bu matnni qayta yozishdan xolos bo‘lamiz. Parametrli funksiya yaratishga doir misol:

Display_copyright funksiyasini chaqirishda parametr uzatish ko‘rinishini qarab chiqamiz. Faraz qilaylik, har biriga alohida

nom tayinlangan bir nechta Web sahifani administratsiyalashga javob berishimiz kerak. Har bir saytda unga tegishli bo'lgan bir nechta o'zgaruvchilar bilan birga xususiy administrativ ssenariysi bor bo'lsin. O'zgaruvchilar tarkibiga sayt nomi \$site o'zgaruvchisi kiradi. Bu holda Display _copyright funksiyani quyidagicha yozish mumkin:

```
function display_copyright ($site_name) {  
    print "Copyright © 2001 $site_name. All Rights Reserved.;"  
}
```

Display_copyright funksiyadan tashqarida qiymat berilayotgan \$site_name

o'zgaruvchisi funksiyaga parametr sifatida uzatiladi. Uzatilgan qiymat funksiyani ixtiyoriy ishlatalishi va takomillashtirilishi mumkin. Lekin har qanday o'zgartirish shu funksiyaning ichida ta'sir qiladi. Funksiyani boshqa funksiya ichidan ham chaqirish mumkin.

Funksiya qiymatini qaytarish. Funksiya ishini yakunlagandan keyin ko'p hollarda birorta qiymat qaytarish foydali bo'ladi. Buning uchun chaqirilgan funksiya natijasi odatda birorta o'zgaruvchiga beriladi. Funksiyalar ixtiyoriy toifali qiymatlarni, jumladan masivlar va ro'yxatlarni qaytarishi mumkin. Misol:

Berilgan umumiy yig'indidan nalogni hisoblash va umumiy yig'indi nalog bilan qaytariladigan funksiya yaratish kerak. Funksiya nomi calculate_cost() deb nomlanadi. Masalani yechish algoritmi quyidagicha:

1. Funksiyani chaqirishdan avval o'zgaruvchilar \$price (tovar narxi) va \$tax (nalog stavkasi) qiymat berish.

2. Calculate_cost() funksiyani e'lon qilish. Funksiya chaqirilganda ikkita parametr (nalog stavkasi va tovar narxi) uzatiladi.

3. Nalogni hisobga olgan holda narxni hisoblash va uni return buyrug'i bilan qaytarish.

4. Calculate_cost() funksiyani chaqirish va \$total_cost o'zgaruvchiga funksiya qaytargan qiymatni tayinlash.

5. Mos xabarni chiqarish. Dastur listingi.

Listing 4.2. Nalogni hisoblash uchun funksiya yaratish.

```
$price = 24.99; $tax = .06;
```

```
function calculate_cost($tax, $price) {
```

```
    $sales_tax = $tax;
```

```
    return $price + ($price * $sales_tax);
}
// calculate_cost() funksiyani qiymat qaytarishiga e'tibor ber-  
ring:
```

```
    $total_cost = calculate_cost ($tax, $price);
// narxni ikkita o'nik raqamgacha yaxlitlang.
$total_cost = round($total_cost, 2);
print "Total cost: $".$total_cost;
// $total cost = 26.49
```

Shuningdek, funksiya ro'yxat yordamida bir nechta qiymatni bordaniga qaytarishi ham mumkin.

8.4. MySQL PHP tilida ma'lumotlar bazasi bilan ishlash vositasি

Ma'lumotlar bazasi va PHP. Internet taraqqiyotida katta hajmdagi axborotlarni saqlash va tanlash samarali vositalari muhim o'rin egallagan. Odatda axborotlarni saqlash uchun ma'lumotlar bazasi ishlatiladi. Taniqli saytlar YAHOO, AMAZON, EBAY ishlashida ma'lumotlar bazasi ishonchliligi muhim o'rin egallaydi. Web muhitida ma'lumotlar bazasi bilan ishlashni tashkil qilishda odatda birorta interfeys tili (masalan, PHP, Perl yoki Java) yordamida ma'lumotlar bazasi bilan bog'lanish yaratiladi, so'ngra dastur yordamida ma'lumotlar bazasiga so'rovlар bilan murojaat qilinadi. Hozirgi kunda PHPda mavjud barcha ma'lumot baza serverlari bilan ishlash imkoniyatlari bor (masalan, ADABAS D, INFORMIX, INTERBASE, MySQL, SyBASE, ODBC, ORACLE va boshqalar). PHP tilida ma'lumotlar bazasi bilan ishlashning ta'minlanishi, ma'lumotlar bazasi bilan bog'lanish, so'rovlarni qayta ishlash va bog'lanish, uzishni tashkil qiluvchi standart funksiyalar majmuasi bilan amalga oshiriladi. Jumladan MySQL serveri bilan ishlash PHP da ma'lumotlar bazasi bilan ishlashni tashkil qilish imkoniyatlari haqida ma'lumot beradi.

MySQL – SQL asosida ishlaydigan ma'lumot bazasini boshqarish tizimi hisoblanadi. (U DataKonsult AB Stokholm, Shvetsiya firmasi tomonidan yaratilgan va kuzatib boriladi). 1995-yildan boshlab u dunyoda eng keng tarqalgan ma'lumot bazasining boshqarish tili hisoblanadi. U yuqori ishslash tezligi, ishonchliligi va litsenziyalash siyosatining moslashuvchanligi bilan ajralib

turadi. MySQL yaxshi xarakteristikalari hamda ishlatishda oddiy va qulay bo'lgan o'zining keng standart interfeys funksiyalariga asosan PHPda ma'lumot bazasi bilan ishlashning asosiy vositasi bo'lib qoldi. Bundan tashqari MySQL serveri bilan ishlashning ta'minlanishi PHP paketi tarkibiga kiritilgan va u PHP bilan avtomatik o'rnatiladi. MySQL serveri o'rnatilgandan keyin uni sozlash kerak. Sozlash jarayoni yangi ma'lumotlar bazasini yaratish va MySQLning imtiyozlar jadvalini tahrirlashdan tashkil topadi. Imtiyozlar jadvali MySQLga murojaatni boshqaradi. Bu jadvalni to'g'ri sozlash ma'lumotlar bazasining xavfsizligini ta'minlashda muhim o'rin egallaydi.

8.5. MySQL bilan ishlash uchun PHPning standart funksiyalari

PHP ssenariyalarini MySQL serveri bilan o'zaro ishlashini ta'minlashni tashkil qiladigan PHP standart funksiyalarining ba'zilarini ko'rib chiqamiz. MySQL serveri bilan ishlashda umumiylar bajariladigan amallar ketma-ketligi quyidagicha:

1. MySQL serveri bilan bog'lanish o'rnatiladi. Agar bunday urinish muvaffaqiyatsiz tugallansa, unga mos xabar chiqarilsin va jarayon to'xtatilsin.

2. MySQL serverida ma'lumot bazasi tanlansin. Agar urinish muvaffaqiyatsiz tugallansa, unga mos xabar chiqarilsin va jarayon to'xtatilsin.

So'rvlarga ishlov berish uchun bir vaqtda bir nechta ma'lumot bazasini ochishga ruxsat beriladi.

3. Tanlab olingan ma'lumot bazasi (bazalari) so'rvlariga ishlov berish.

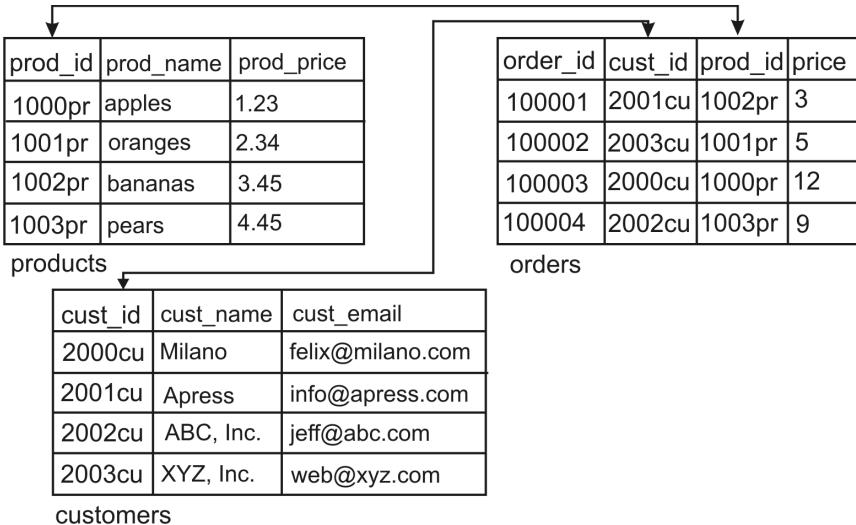
4. So'rvlarga ishlov berish yakunlangandan so'ng, ma'lumot baza serveri bilan bog'lanishni yopish.

Bu paragraf misollarini namoyish qilishda customer, orders va products jadvallari ishlatiladi.

Masalani yechish MySQL serveriga ulanish bilan boshlanadi.

mysql_connect() funksiysi. MySQL serveri bilan aloqa o'rnatishga xizmat qiladi. Unga muvaffaqiyatli ulangandan so'ng, ma'lumot bazasini tanlashga o'tish mumkin. mysql_connect() funksiyasining sintaksisi quyidagicha:

```
int mysql_connect ([string xost [:port] [:/yo'l//soketga] [, string foydalanuvchi nomi] [, string parol]])
```



8.5.1-rasm. Relatsion ma'lumotlar bazasiga doir misol.

Xost parametri orqali MySQL serverini imtiyozlar jadvalida ko'rsatilgan xost kompyuteri nomi uzatiladi. Shuningdek, uning o'zi MySQL ishlayotgan Web serverga so'rovlar jo'natish uchun ishlatiladi, chunki MySQL serveriga uzoqlashgan rejimda (holatda) ularish mumkin. Xost nomi bilan majburiy bo'lмаган parametrlar – port nomeri (raqami) shuningdek soketga yo'l (lokal xost uchun) ko'rsatilishi mumkin. Foydalanuvchi nomi va parol parametrlari mysql ni imtiyozlar jadvalida berilgan foydalanuvchi nomi va parolga mos bo'lishi kerak. E'tibor bering, barcha parametrlar majburiy bo'lmasligi mumkin, modomiki imtiyozlar jadvalini shunday sozlash mumkin-ki, u bog'lanishni tekshirmsandan bajarishni amalga oshirishi mumkin.

Agar xost parametri berilmagan bo'lsa, mysql_connect() funksiyasi lokal xost bilan aloqa o'rnatishga harakat qiladi. MySQL bilan bog'lanishni ochishga doir misol:

```
@mysql_connect(" local host", "web", "4tf9zzf") or die("Could not connect to MySQL server!");
```

Bu misolda localhost – kompyuter nomi, web – foydalanuvchi nomi, 4tfzzf – parol, funksiyani chaqirish oldidagi @ belgi bog'lanish muvaffaqiyatsiz tugaganda chiqadigan xatoliklar haqidagi barcha xabarlarni to'xtatadi va u die chaqirishida ko'rsatilgan

xabar bilan almashtiriladi. Bu misolda mysql_connect() chaqirishda qaytariladigan qiymat ishlatilmayotganiga e'tibor bering. Agar dasturda MySQL serveri bilan faqat bitta bog'lanish ishlatsa, bu me'yoriy hol. Agar dastur har xil xostdagи bir nechta MySQL serverlari bilan bog'lanish o'rnatса, unda mysql_connect() funksiyani chaqirishda qaytariladigan identifikator qiymatini saqlash kerak, chunki uning keyingi komandalarini kerakli MySQL serverlariga yuborish uchun ishlatiladi. Misol:

```
<?
$link1= @mysql_connect("www.somehost.com", "web", "ab-
cde") or die("Could not connect to MySQL server!");
$link2 = @mysql_connect("www.someotherhost.com", "usr",
"secret") or die ("Could not connect to MySQL server!");
?>
```

\$link1 va \$link2 identifikatorlar so'rovlar bilan ma'lumotlar bazasiga keyingi murojaatlarda uzatiladi.

mysql_select_db() funksiyasi. MySQL bilan muvaffaqiyatli bog'langandan so'ng, serverda joylashgan ma'lumot bazasini tanlab olish zarur. Buning uchun mysql_select_db() funksiyasi ishlatiladi. Uning sintaksi si quyidagicha:

```
int mysql_select_db (string ma'lumot_baza_nomi [, int
bog'lanish _identifikatori])
```

ma'lumotlar bazasi nomi parametri identifikatori mysql_select_db() funksiyasi bilan qaytariladigan, tanlab olinayotgan ma'lumotlar bazasi nomini aniqlaydi.

Bog'lanish identifikatori parametri MySQL serveri bilan bitta ochiq bog'lanish bo'lganda majburiy emas. Bir nechta ochiq bog'lanish bo'lganda, bu parametrini ko'rsatish zarur. mysql_select_db() funksiyasi bilan ma'lumotlar bazasini tanlab olishga doir misol:

```
<?
@mysql_connect("localhost", "web", "4tf9zzf")
or die("Could not connect to MySQL server!");
@mysql_select_db("company") or die ("Could not select com-
pany database!");
?>
```

Agar dasturda faqat bitta ma'lumotlar bazasi tanlab olinsa, uning identifikatorini saqlash majburiy emas. Lekin bir nechta ma'lumot bazasi tanlanayotganda, qaytariladigan identifikator

saqlanadi, chunki, so‘rovlarga ishlov berishda kerakli ma’lumotlar bazasiga murojaat qilish imkoniyatiga ega bo‘lishi kerak. Agar identifikator ko‘rsatilmasa oxirgi tanlab olingan ma’lumotlar bazi ishlataladi.

mysql_close() funksiyasi. MySQL serveri bilan ishni yakunlagandan so‘ng bog‘lanishni yopish uchun ishlataladi. mysql_close() funksiyasi majburiy bo‘lмаган parametr bilan aniqlanadigan bog‘lanishni yopadi. Agar parametr berilmasa, mysql_close() funksiyasi oxirgi ochiq bog‘lanishni yopadi. Bu funksianing sintaksisi quyidagicha:

```
int mysql_close ([int bog‘lanish _identifikatori])
```

Misol:

```
<?
```

```
@mysql_connect("localhost", "web", "4tf9zzzf")
```

```
or die ("Could not connect to MySQL server!");
```

```
@MySQL_select_db("company") or die ("Could not select  
company database!"); print "You're connected to a MySQL da-  
tabase!";
```

```
?>
```

Bu misolda bog‘lanish identifikatorini ko‘rsatish shart emas, chunki mysql_close() funksiyasi chaqirish vaqtida server bilan faqat bitta ochiq bog‘lanish mavjud.

mysql_query() funksiyasi. Bu funksiya ma’lumotlar bazasiga so‘rovlari bilan murojaat qilish uchun interfeys bilan ta’minlaydi. Bu funksianing sintaksisi quyidagicha:

```
int mysql_query (string so‘rov [, int bog‘lanish _identifikatori])
```

So‘rov parametrida SQL tilidagi so‘rov matni joylashadi. So‘rov majburiy bo‘lмаган bog‘lanish identifikatori bilan aniqlanadigan bog‘lanishga yoki parametr ko‘rsatilmaganda oxirgi ochiq bog‘lanishga uzatiladi. Bu funksiya so‘rovni qayta ishslash natijasini chiqarmaydi. SQL SELECT komanda muvaffaqiyatli bajarilganda natija identifikatori qaytariladi, u keyinchalik mysql_query() funksiyasiga so‘rov natijasini formatlash va akslantirish uchun uzatiladi. Agar so‘rovga ishlov berish muvaffaqiyatsiz tuga-sa, funksiya FALSE qiymatni qaytaradi.

mysql_affected_rows () funksiyasi. Bu funksiya SQL so‘rovlari (INSERT, UPDATE, REPLACE yoki DELETE komandalari bilan) qatnashayotgan yozuvlar sonini aniqlash uchun ishlataladi. Uning sintaksisi quyidagicha:

int mysql_affected_rows ([int bog'lanish _identifikatori])
Bu yerda bog'lanish identifikatori parametri majburiy emas.
Agar u ko'rsatilmasa, int mysql_affected_rows funksiya oxirgi
ochilgan bog'lanishni ishlatadi. Misol:

```
<?
// Serverga ulanish va ma'lumotlar bazasini tanlash
@mysql_connect("localhost", "web", "4tf9zzzf")
or die("Could not connect to MySQL server!");
@mysql_select_db("company") or die("Could not select company database!");
// So'rov yaratish
$query = "UPDATE products SET prod_name = \"canta-loupe\"
WHERE prod_id = '10001pr\";
// So'rovni bajarish $result = mysql_query($query);
// Tiklangan yozuvlar sonini aniqlash
print "Total row updated: ".mysql_affected_rows();
mysql_close();
?>
```

Bu dastur fragmenti bajarilganda quyidagi natija chiqariladi:
Total row updated: 1

int mysql_affected_rows funksiyasi SELECT komandasiga
asoslangan so'rovlari bilan ishlamaydi. SELECT komandasining
ishlashida (chaqirishda) qaytariladigan yozuvlar sonini aniqlash
uchun mysql_num_rows() funksiyasi ishlatiladi. Uning sintaksi
quyidagicha:

```
int mysql_num_rows(int natija)
mysql_num_rows( ), funksiyasini ishlatishga doir misol:
<?
// Serverga ulanish va ma'lumotlar bazasini tanlash
@mysql_connect("localhost", "web", "4tf9zzzf")
or die("Could not connect to MySQL server!");
@mysql_select_db("company") or die("Could not select company database!");
// nomi 'p' harfi bilan boshlanadigan barcha tovarlarni tanlash
$query = "SELECT prod_name FROM products WHERE
prod_name LIKE 'p%'";
// So'rovni bajarish
$result = mysql_query($query);
```

```
print "Total rows selected: ".mysql_num_rows($result);
mysql_close();
?>
```

Demak, jadvalda nomi p harfi bilan boshlanadigan bitta tovar bo‘lgani uchun faqat bitta yozuv qaytariladi.

Natija:

Total rows selected: 1

mysql_result() funksiyasi. Bu funksiya mysql_query() funksiyasi bilan birgalikda ishlataladi (SELECT buyrug‘i bilan so‘rov bajarilganda).

mysql_result funksiyasi sintaksisi quyidagicha:

int mysql_result (int natija_identifikator, int yozuv [. mixed pole])

natija identifikatori parametriga mysql_query() funksiyasi qaytaradigan qiymat uzatiladi. Yozuv parametri natija identifikatori bilan aniqlanadigan ma’lumotlar to‘plamini aniq yozuviga murojaat uchun ishlataladi. Nihoyat, majburiy bo‘lmagan parametr maydonida uzatilishi mumkin:

jadvaldagи maydon siljishi;

maydon nomi;

maydon_nomi_jadval_nomi formatidagi maydon nomi.

Misol: MySQL ma’lumotlar bazasida ma’lumotlarni tanlash va formatlash.

Dastur listingi:

<?

```
@mysql_connect("localhost", "web", "fftss")
or die("Could not connect to MySQL server!");
@mysql_select_db("company")
or die("Could not select products database!");
// products jadvalidan barcha yozuvlarni tanlash (ajratib
olish)
$query = "SELECT * FROM products"; $result = mysql_
query($query);
$x = 0;
print "<table>\n";
print "<tr>\n<th>Product ID</th><th>Product Name</
th><th>Product Price</th>\n</tr>\n";
WHILE ($x < mysql_numrows($result)) :
$id = mysql_result($result, $x, 'prod_id');
```

```

$name = mysql_result($result, $x, 'prod_name');
$price = mysql_result($result, $x, 'prod_price');
print "<tr>\n";
print "<td>$id</td>\n<td>$name</td>\n<td>$price</td>\n";
print "</tr>\n";
$x++;
endwhile;
print "</table>";
mysql_close();
?>

Bu misolni bajarish natijasida quyidagi natijaga ega bo'lamiz:
<table>
<tr>
<th>Product ID</th><th>Product Name</th><th>Product Price</th>
</tr>
<tr>
<td>1000pr</td>
<td>apples</td>
<td>1.23</td>
</tr>
<tr>
<td>1001pr</td>
<td>oranges</td>
<td>2.34</td>
</tr>
<tr>
<td>1002pr</td>
<td>bananas</td>
<td>3.45</td>
</tr>
<tr>
<td>1003pr</td>
<td>pears</td>
<td>4.45</td>
</tr>
</table>

```

`mysql_result` funksiyasi uncha katta bo‘lмаган ма’лумотлар то‘плами билан исхлasheda qulay, lekin ancha samarali исхlaysidigan boshqa funksiyalar ham mavjud, masalan, `mysql_fetch_row()` va `mysql_fetch_array()` funksiyalari.

Ko‘pgina hollarda alohida maydonlarni olish uchun `mysql_result()` funksiyasini ko‘p marta takroran qo‘llashdan ko‘ra, indekslanuvchi massiv (indeksi 0 dan boshlanadigan) elementlariga yozuvning barcha maydonlari qiymatini tayinlash qulay. Bu masala `mysql_fetch_row()` funksiyasi bilan yechiladi. Uning sintaksisi quyidagicha :

```
array mysql_fetch_row (int natija)
```

List() funksiyasini `mysql_fetch_row()` bilan birgalikda ishlatish bir qancha buyruqlarni tejash imkonini beradi.

Misol: Ma’лумотлarni `mysql_fetch_row()` funksiyasi bilan tanlash (ajratish).

```
// Ma’лумотлarni mysql_fetch_row() funksiyasi bilan ajratish
<?
@mysql_connect( "localhost", "web", "ffttss") or die("Could
not connect to MySQL server!");
@mysql_select_db("company") or die("Could not select pro-
ducts database!");
$query = "SELECT * FROM products";
$result = mysql_query($query);
print "<table>\n";
print "<tr>\n<th>Product ID</th><th>Product Name</
th><th>
```

```
Product Price</th>\n</tr>\n";
while ($row = mysql_fetch_array($result)) :
print "<tr>\n";
print "<td>".$row["prod_id"]."</td>\n<td>".$row["prod_
name"]."."
</td>\n<td>". $row["prod_price"]. "</td>\n";
print "</tr>\n";
endwhile;
print "</table>";
mysql_close();
?>
```

Keltirilgan ma’лумотлар ilovalar yozishga yetarli. Keltirilgan ma’лумотлар asosida oddiy qidiruv tizimining ilovasini yarat-

miz. Namoyish misoli HTML formasini ma'lumotlarni olishga qo'llashni namoyish qiladi, keyinchalik ular bazadan ma'lumotlar tanlab olish uchun ishlataladi.

Sodda qidiruv tizimi yaratish. Web tizimida ko'p hollarda qidirish tizimlari bilan ishlashga to'g'ri keladi. Sodda qidirish tizimida hech bo'limganda bitta kalit so'z ishlataladi. Bu so'z SQL so'rov tarkibiga kiritiladi, so'ngra u ma'lumotlar bazasidan axborot tanlashda ishlataladi. Qidirish natijasi qidirish tizimi yordamida u yoki bu mezon asosida formatlanadi (masalan, kategoriyalar yoki moslik darajasiga qarab). Quyida keltirilgan qidirish tizimi listin-gida mijozlar haqidagi axborot qidiriladi. Formada foydalanuvchi qidirish amalga oshiriladigan kalit so'z kiritadi va kategoriyasini tanlaydi (nomi, identifikatori yoki mijozning elektron pochta adresi). Agar foydalanuvchi kiritgan nom, identifikator yoki adresi (manzil) mavjud bo'lsa, qidirish tizimi bazadan qolgan atributlarni tanlab oladi. So'ngra xaridor identifikatori bo'yicha order jadvalidan buyurtmalar tanlab olinadi. Mijoz tomonidan rasmiylashtirilgan buyurtmalarning barchasi ular hajmining kamayib borish tartibida rasmiylashtiriladi (akslantiriladi). Agar kategoriyada (toifada) foydalanuvchi tomonidan berilgan kalit so'z uchramasa qidirish to'xtatiladi, dastur kerakli xabarni chiqaradi va yana forma akslantiriladi.

Sodda qidirish tizimi (searchengine.php).

```
Sodda qidirish tizimi (searchengine.php) listingi
<?
$form =
"<form action=\"Listing.php\" method=\"post\">
<input type=\"hidden\" name=\"seenform\" value=\"y\">
Keyword:<br>
<input type=\"text\" name=\"keyword\" size=\"20\" 
maxlength=\"20\" value= \" \"><br>
Search Focus:<br>
<select name=\"category\">
<option value=\"\">Choose a category:
<option value=\"cust_id\">Customer ID
<option value=\"cust_name\">Customer Name
<option value=\"cust_email\">Customer Email
</select><br>
<input type=\"submit\" value=\"search\">,
```

```

</form>
// Agar forma akslantirilmagan bo'lsa uni akslantiring:
if ($seenform != "y") :
print $form; else :
// MySQL serveriga ulanish va ma'lumotlar bazasini tanlash
@mysql_connect("localhost", "web", "fftss")
or die("Could not connect to MySQL server!");
@mysql_select_db("company")
or die("Could not select company database!");
// so'rov quring va bajaring:
$query = "SELECT cust_id, cust_name, cust_email
FROM customers WHERE $category = '$keyword'";
$result = mysql_query($query);
// Agar moslari topilmasa xabar chiqaring:
// va yana formani akslantiring:
if (mysql_num_rows($result) == 0) :
print "Sorry, but no matches were found. Please try your
search again:";
print $form;
// Mosliklar topildi. Natijani formatlang va chiqaring else :
// Formatlang va maydon qiymatlarini chiqaring:
list($id, $name, $email) = mysql_fetch_row($result);
print "<h3>Customer Information:</h3>";
print "<b>Name:</b> $name <br>";
print "<b>Identification #:</b> $id <br>";
print "<b>Email:</b> <a href=\"mailto:$email\">$email</a>
<br>";
print "<h3>Order History:</h3>";
// 'orders' jadvaliga so'rov quring va bajaring:
$query = "SELECT order_id, prod_id, quantity
FROM orders WHERE cust_id = '$id'
ORDER BY quantity DESC";
$result = mysql_query($query):
print "<table border = 1>";
print "<tr><th>Order ID</th><th>Product ID</th>
<th>Quantity</th></tr>";
// topilgan yozuvlarni formatlang va chiqaring.
while (list($order_id, $prod_id, $quantity) = mysql_fetch_
row($result));

```

```

print "<tr>";
print "<td>$order_id</td><td>$prod_id</td><td>$quantity</td>";
print "</tr>";
endwhile;
print "</table>";
endif;
endif;
?>

```

Agar Milano kalit so‘zini kiritsak va yoyiluvchi ro‘yxatdan Customer Name (Mijoz nomi) toifasini tanlasak, dastur qu-yidagicha axborot chiqaradi:

Customer information:

Name:Milano

Identification#:2000cu

Email:felix@milano.com

Order History:

| Order Id | Product Id | Quantity |
|----------|------------|----------|
| 100003 | 1000pr | 12 |
| 100005 | 1002pr | 11 |

Bu yerda sodda qidiruv tizimi ko‘rib chiqildi. Hozirgi kunda bir nechta kalit so‘zlar bo‘yicha qidirish, to‘liq bo‘limgan kalit so‘zlar bo‘yicha qidirish yoki o‘xshash kalit so‘zli yozuvlarni avtomatik qidirish kabi qo‘srimcha imkoniyatlari qidirish tizimlari mavjud.

Jadvallarda tartiblash. Ma’lumotlar bazasidan ma’lumotlar chiqarilayotganda, ularni har xil mezon bo‘yicha tartiblash imkoniyatlari bo‘lishini hisobga olish kerak.

Misol sifatida biz qidirish tizimi tomonidan chiqarilgan Order History nomli natija jadvaliga murojaat qilamiz. Faraz qilaylik, jadvalagi ma’lumotlar ro‘yxati yetarli daraja uzun bo‘lib, biz undagi ma’lumotlarni tovar identifikatori (yoki buyurtma identifikatori) bo‘yicha tartiblash talab etilsin. Tartiblashni shunday bajarish talab etiladiki, bunda listingdagi dastur kodiga o‘zgarish kiritmay, avtomatik ravishda ko‘rsatilgan ustunning ma’lumoti bo‘yicha tartiblash talab etiladi. Bunda dasturni ko‘rib chiqish jarayonida aniq toifa (kategoriya) ustuning nomi (nomi, joylashish sanasi, yuklanishlar soni yoki fayl o‘lchami) ustida sichqoncha bosilsa, shu ustun ma’lumotlari bo‘yicha avtomatik tartiblanadi.

Bunday imkoniyat quyidagi listingda amalga oshirilgan. Quyidagi listingda ma'lumotlar Order jadvalidan tanlab olishni bajarayapmiz. Sukut bilan ma'lumotlar jadvalda buyurtma hajmining kamayib borish tartibida (quanity maydoni) tartiblanadi. Lekin jadval maydon nomlarini ixtiyorida sichqoncha bosilsa, bu sahifa ko'rsatilgan ustun bo'yicha tartiblangan holda yuklanadi.

Listing. Jadvallarni tartiblash (tablesorter.php)

<?

```
// MySQL serveri ulanish va ma'lumotlar bazasini tanlash
@mysql_connect("localhost", "web", "fftss")
or die("Could not connect to MySQL server!");

@mysql_select_db("company")
or die("Could not select company database!");
// Agar $key o'zgaruvchi qiymati berilmagan bo'lsa, sukut bilan
// 'quantity' if (! isset($key)) qiymati ishlatalidi:
$key = "quantity"; endif;
// So'rov yaratish va bajarish.
// Tanlab olingan ma'lumotlar $key ustuni kamayib borishi
bo'yicha tartiblanadi.

$query = "SELECT order_id, cust_id, prod_id, quantity
FROM orders ORDER BY $key DESC"
$result = mysql_query($query);

// Jadval sarlavhasini yaratish:
print "<table border = 1>";
print "<tr>";
<th><a href=\"Listing.php?key=order_id\">Order ID</a></th>
<th><a href=\"Listing.php?key=cust_id\">Customer ID</a></th>
<th><a href=\"Listing6.php?key=prod_id\">Product ID</a></th>
<th><a href=\"Listing6.php?key=quantity\">Quantity</a>
</th></tr>";

// Jadvalning har bir satrini formatlang va chiqaring:
while (list($order_id, $cust_id, $prod_id, $quantity)
= mysql_fetch_row($result)) :
print "<tr>";
print "<td>$order_id</td><td>$cust_id</td><td>$prod_id</td><td>
```

```

$quantity</td>";
print "</tr>";
endwhile;
// jadvalni yakunlang:
print "</table>";

```

Bu listingning standart chiqadigan ma'lumotlari quyidagi ko'rinishda bo'ladi:

| <u>Order ID</u> | <u>Customer ID</u> | <u>Product ID</u> | <u>Quantity</u> |
|-----------------|--------------------|-------------------|-----------------|
| 100003 | 2000cu | 1000pr | 12 |
| 100005 | 2000cu | 1002pr | 11 |
| 100004 | 2000cu | 1000pr | 9 |
| 100002 | 2000cu | 1001pr | 5 |
| 100001 | 2000cu | 1002pr | 3 |

E'tibor bering: jadval sarlavhalari giperssilkalardan iborat. Modomiki, sukut bilan tartiblanish quantity maydoni bo'yicha amalga oshirilgani uchun yozuvlar oxirgi ustun qiymati kamayib borish bo'yicha tartiblangan. Agar Order_ID giperssilkada sichqoncha bosilsa, sahifa yozuvlar buyurtma identifikatorini kamayib borish tartibida yangidan yuklanadi. Jadval quyidagi ko'rinishda bo'ladi:

| <u>Order ID</u> | <u>Customer ID</u> | <u>Product ID</u> | <u>Quantity</u> |
|-----------------|--------------------|-------------------|-----------------|
| 100005 | 2000cu | 1002pr | 11 |
| 100004 | 2000cu | 1000pr | 9 |
| 100003 | 2000cu | 1000pr | 12 |
| 100002 | 2000cu | 1001pr | 5 |
| 100001 | 2000cu | 1002pr | 3 |

8.6. Ma'lumotlar bazasining ochiq arxitekturasi (ODBC – Open Data Base Connectivity)

Maxsus funksiyalar birorta aniq turdag'i ma'lumotlar bazasini boshqarish tizimi bilan yaxshi ishlaydi. Ba'zi hollarda bir nechta MBBT bilan (MySQL, Microsoft SQL SERVER, ACCESS va boshqalar) ishlashga to'g'ri keladi. Xuddi shunday muammolar MBBT bilan bog'liq bo'limgan ilovalar yaratishda paydo bo'ladi. Bunday ilovalar mavjud mijoz ma'lumotlar bazasining infrastrukturasi bilan ishlaydi. ODBC amaliy dastur interfeysidan iborat (API) bo'lib, umumiylab abstrakt funksiyalar to'plamini har xil turdag'i ma'lumotlar bazalari bilan ishlashish imkonini beradi. Bunday yondashishning ahamiyati katta, chunki har xil tur-

dagi ma'lumotlar bazalari bilan bir xil amallarni bajarish uchun dastur kodini qayta yozishdan xalos qiladi. ODBC yordamida ma'lumotlar bazasi serverlari bilan ishslash uchun ularga ODBC drayverlari mavjud bo'lishi kerak.

Microsoft tomonidan yaratilgan ODBC standarti, ochiq standart bo'lishiga qaramasdan u asosan WINDOWS, LINUX platformasida ishlaydigan MBBT uchun ishlatilmoqda. ODBC drayverlari maqsadi, platformasi va vazifasi bo'yicha farqlanadi.

PHPda ODBC ishlashini ta'minlash. PHPda ODBC funksiyalari, odatda ODBC umumiyligi funksiyalari deyiladi. Hozirgi kunda 40 ga yaqin ODBCning umumiyligi funksiyalari mavjud. Ummuman olganda, sodda hollarda ODBCga moslashgan ma'lumotlar bazasidan axborot olish uchun quyida keltirilgan bir nechta funksiyalar yetarlidir.

odbc_connect() – bu funksiya ODBCga moslashgan ma'lumot bazasi bilan bog'lanish o'rnatish uchun ishlatiladi. Uning sintaksi:

```
int odbc_connect (string ma'lumot_mambai, string foydalannuvchi_nomi, string parol [, int kursov_turi])
```

Bu yerda, ma'lumot_ manbai parametri, bog'lanish o'rnatilishiga harakat qilinayotgan ODBC moslashgan ma'lumot bazasini aniqlaydi. Foydalanuvchi_nomi va parol parametrlari yordamida ma'lumot manbaiga ularishi zarur bo'lgan foydalanuvchi nomi va paroli beriladi. Zarur bo'limgagan kursov_turi parametri ODBCni ba'zi bir drayverlari bilan ishlaganda vujudga keladigan muammolarni bartaraf qilishga mo'ljallangan. U quydagi 4 ta qiymatlardan birini qabul qilishi mumkin:

- SQL_CUR_USE_IF_NEEDED;
- SQL_CURSOR_USE_ODBC;
- SQL_CUR_USE_DRIVER;
- SQL_CUR_DEFAULT.

Dasturda Odbc_connect() funksiyasini ishlatishga doir misol ko'ramiz:

```
<?
obdc_connect("myAccessDB", "user", "secret")
or die( "Could not connect to ODBC database");
?>
```

odbc_close() – bu funksiya ODBC – moslashgan ma'lumot bazasi bilan ishslash yakunlanganda, bog'lanishni yopib, joriy

bog'lanishda ishlatilgan barcha resurslarni bo'shatish imkonini beradi. Uning sintaksisi quyidagicha:

```
void odbc_close ([int bog'lanish _identifikatori])
```

Bog'lanish _identifikatori parametri orqali bog'lanishning ochiq identifikatori uzatiladi. Misol:

```
<?
    obdc_connect("myAccessDB", "user", "secret")
    or die("Could not connect to ODBC database");
    print "Currently connected to ODBC database!"; odbc_close-
($connect);
?>
```

odbc_prepare(). Bevosita ishlov berishdan oldin so'rovni tayyorlash zarur. Bu masala odbc_prepare() funksiyasi bilan yechiladi. Uning sintaksisi quyidagicha:

```
int odbc_prepare (int bog'lanish _identifikatori, string
so'rov)
```

Bog'lanish identifikatori _ parametrida odbc_connect() funksiyasi qaytaradigan bog'lanish identifikatori uzatiladi. So'rov parametrida server tomonidan bajarilishi zarur bo'lgan so'rov matni uzatiladi. Agar so'rov bajarila olmasa funksiya FALSE qiymatini qaytaradi, aks holda natija identifikatori qaytariladi va u keyinchalik odbc_execute() funksiyani chaqirishda ishlatiladi.

odbc_execute() funksiyasi. odbc_prepare() funksiyasida tayyorlangan so'rov odbc_execute() bilan bajariladi. Uning sintaksisi quyidagicha:

```
int odbc_execute (int natija identifikatori [, array
parametrlar])
```

Natija identifikatori parametrida odbc_prepare() funksiyani chaqirishda qaytariladigan natija identifikatori uzatiladi. Majburiy bo'limgan parametr funksiyani chaqirishda qo'shimcha ma'lumotlar uzatilganda ishlatiladi. Misol:

```
<?
$connect = @obdc_connect("myAccessDB", "user", "secret")
or die("Could not connect to ODBC database");
$query = "UPDATE customers SET cust_id = \"Milano,
Inc.\""
WHERE cust_id \"2000cu\"";
$result = odbc_prepare($connect, $query) or die("Couldn't
prepare query!");
```

```
$result = odbc_execute($result) or die("Couldn't execute query!");
```

```
    odbc_close($connect);
```

```
?>
```

odbc_exec() funksiyasi. Bu funksiya odbc_prepare() va odbc_execute() funksiyalarini birlashtiradi. Uning sintaksisi quyidagicha:

```
int odbc_exec (int bog'lanish identifikatori, string so'rov)
```

Bu yerda bog'lanish identifikatori parametrida odbc_connect() funksiyasi qaytaradigan bog'lanish identifikatori uzatiladi. So'rov parametrida server tomonidan bajarilishi lozim bo'lgan so'rov matni uzatiladi. Agar so'rov bajarila olmasa, funksiya FALSE qiymatni qaytaradi, aks holda funksiyani keyingi chaqirilishlarida ishlatiladigan natija identifikatori qaytariladi. Misol:

```
<?
```

```
    odbc_connect("myAccessDB", "user", "secret")
```

```
    or die("Could not connect to ODBC database");
```

```
    $query = "SELECT * FROM customers";
```

```
    $result = odbc_exec($connect, $query) or die("Couldn't execute query!");
```

```
    odbc_close($connect) ;
```

```
?>
```

Bu misolda odbc_exec() funksiyasi matni \$query o'zgaruvchida joylashgan so'rovni bajarilishga harakat qiladi. So'rov muvaffaqiyatlari bajarilganda \$rezult o'zgaruvchiga natija identifikatori uzatiladi; so'rov bajarila olmasa FALSE qiymati tayinlanadi va die chaqirishda uzatiladigan xabar chiqariladi.

odbc_result_all() funksiyasi – bu funksiya odbc_exec() yoki odbc_execute() funksiyalarini chaqirishda olingan natija identifikatori uchun barcha yozuvlarni formatlaydi va akslantiradi. Uning sintaksisi quyidagicha:

```
int odbc_result_all (int natija identifikatori [, string jadval_formati]):
```

Natija identifikatori parametrida odbc_exec() yoki odbc_execute() funksiyalar muvaffaqiyatlari chaqirilganda qaytariladigan natija_identifikatori uzatiladi. Majburiy bo'limgan parametr jadval_formatida HTML jadvalining xarakteristikalari uzatiladi. Misol:

```
<?
```

```
    odbc_connect("myAccessDB", "user", "secret")
```

```
    or die("Could not connect to ODBC database");
```

```

$query = "SELECT * FROM customers";
$result = odbc_exec($connect, $query) or die("Couldn't execute query!");
odbc_result_all($result, "BGCOLOR=#c0c0c0' border='1'");
$);
odbc_close($connect);
?>

```

Bu misolning bajarilishi natijasida CUSTOMER jadvalidagi ma'lumotlar och kulrang fonda, qalinligi lga teng ramka (chorcho'p) ko'rinishidagi jadvalda chiqariladi.

| <u>cust_id</u> | <u>cust_name</u> | <u>cust_email</u> |
|-----------------------|-------------------------|--------------------------|
| 2000 cu | Milano | felix@milano.com |
| 2001 cu | Apress | info@apress.com |
| 2002 cu | ABC, Inc. | jeff@abc.com |
| 2003 cu | XYZ, Inc. | web@xyz.com |

8.6.1-rasm. Brauzerda ODBC ma'lumotlari.

odbc_free_result() funksiyasi. Dasturlashning yaxshi uslublariidan biri amallar bajarilib bo'lingandan keyin u bilan bog'langan barcha resurslarni bo'shatishdan iborat. ODBC so'rovlar bilan ishlaganda bu masala odbc_free_result() funksiyasi bajaradi. Uning formati quyidagicha:

```
int odbc_free_result (int natija identifikatori)
```

Funksiyaning parametrida, dasturda boshqa ishlatilmaydigan natija identifikatori uzatiladi. Umuman olganda ssenariya o'z ishini yakunlagandan keyin xotiraning barcha resurslari avtomatik ravishda bo'shatiladi. Shu sababli odbc_free_result l() funksiyasini katta hajmdagi xotirani sarf qiladigan katta so'rovlari (ssenariya doirasida bir qancha so'rovlari bo'lgan) yaratishda ishlatiladi. Misol:

```

<?
obdc_connect("myAccessDB", "user", "secret")
or die("Could not connect to ODBC database");
$query = "SELECT * FROM customers";
$result = odbc_exec($connect, $query) or die("Couldn't execute query!");
odbc_result_all($result, "BGCOLOR='#c0c0c0' border='1' ");

```

```
    odbc_free_result($result);
    odbc_close($connect);
?>
```

odbc_free_result() funksiyasi natija identifikatoridan foydalanishni yakunlagandan so'ng xotira tizimiga qaytariladi.

8.7. Microsoft ACCESS va PHP

Hozirgi kunda Microsoft ACCESS ma'lumotlar bazasini boshqarish tizimi eng ommabop tizim hisoblanadi. Buning asosiy sabablaridan biri, uning qulay grafik interfeysga ega ekanligidir. Microsoft ACCESS mustaqil ma'lumotlar bazasini boshqarish tizimi sifatida ishlashidan tashqari, uning grafik interfeysini boshqa ma'lumotlar bazasi – masalan, MySQL yoki Microsoft SQL SERVER bilan ishlashini tashkil qilishda foydalanish mumkin. ODBCni PHP tilida ishlashini namoyish qilish maqsadida PHP tilida Microsoft ACCESS ma'lumot bazasiga ulanish jarayonini ko'rib chiqamiz. Jarayonni qadamma-qadam tavsiflab beramiz.

1. Microsoft ACCESS ma'lumot bazasini yaratamiz. Misol sifatida xodimlar bilan bog'lanish (kontakte) jadvalini yaratamiz, uni bir nechta yozuvlar bilan to'ldiramiz.

2. Kompyuterda jadvalga nom berib saqlab qo'yamiz:

| Contacts : Table | | | | | |
|------------------|--------------|------------|------------|---------------|----------------------|
| | Contact ID | First Name | Last name | Cell Phone | Email |
| ▶ | 1 | William | Gilmore | (724)555-1234 | wj@wjgilmore.com |
| | 2 | Chef | Ale | (039)867-5309 | chef@phprecipes.com |
| | 3 | Pierre | BonneSoupe | (212)555-8743 | pierre@phprecipes.co |
| * | (AutoNumber) | | | | |

8.7.1-rasm. Jadval ko'rinishi.

3. Keyingi qadamda ACCESS ma'lumot bazasiga ODBC yordamida murojaat tashkil qilamiz. Buning uchun *Пуск > Настстройка > Панель управления* (*Start > Settings > Control Panel*) komandasini bajaramiz. Boshqarish panelida «Источник данных ODBC» (32) (*ODBC Data Source (32 bit)*) belgini topamiz. Bu belgi yordamida bizning sistemamizdagи ma'lumotlar manbai va har xil drayverlarni sozlashga mo'ljallangan ODBC Administratori ishga tushiriladi. Bu belgini sichqoncha bilan 2 marotaba chertib ishga tushiramiz. Administrator darchasi su-kut bilan Foydalanuvchi DSN (User DSN) vkladkasida ochiladi.

Bu vkladkada ma'lumotlar manbai sanab o'tilgan. Ular faqat shu kompyuterda ishlatishi mumkin va aniq foydalanuvchiga tegishli bo'ladi. Ushbu misolda xuddi shunday ma'lumot manbaidan foydalanamiz.

4. Darchani o'ng qismidagi «Добавить ...» (ADD....) tugmasini bosamiz. Bunda yangi manba bilan ishlashga mo'ljallangan drayverlar tanlash uchun yangi darcha ochiladi. Microsoft ACCESS DRIVER (.mdb) satrini tanlang va ГОТОВО (FINISH) tugmasini bosing.

5. Ekranda Microsoft ACCESS (ODBC Microsoft ACCESS setup) uchun ODBC drayverlarini o'rnatish uchun yangi darchasi paydo bo'ladi. Formada «Имя источника данных» (Data Source Name) ma'lumot manbai nomi matnli maydonni toping va unga Microsoft ACCESSda yaratilgan ma'lumot bazasi nomini kriting. Ma'lumot manba nomi maydonning bevosita ostida joylashgan matnli maydonga ham kiritilishi mumkin.

6. «Выбрать...» (select...) tugmasini bosing. WINDOWS provodnigi uslubidagi darcha paydo bo'ladi. Unda sizga murojaat qilish uchun ODBC vositalari yordamida amalga oshiriladigan ma'lumot bazasini topish tavsija etiladi.

7. Daraxt ko'rinishidagi kataloglar ro'yxatida o'zingizga tegishli ma'lumot bazasini toping va unda sichqonchani 2 marotaba bosing. Siz yana Microsoft ACCESS uchun ODBC drayverlarini o'rnatish darchasiga qaytasiz. Tanlab olingan ma'lumot bazasiga yo'l «Выбрать...» tugmasi ustida akslanadi. OK tugmasini bosing.

Aytib o'tilgan qadamlar bajarilgandan so'ng Microsoft ACCESS ma'lumot bazasi bilan ODBC vositalari yordamida ishlashning mumkin. Bu ma'lumot bazasi bilan ODBC orqali ishlash uchun ssenariyu yaratish yetarli.

Misol: quyidagi ssenariya asosida ODBCni umumiyligi funksiyalarini ishlatib, Microsoft ACCESS yaratilgan «Ulanishlar» jadvalidagi barcha axborotlarni chiqarish uchun ishlatish mumkin.

Listing. ODBC funksiyalarini MS Access bilan ishlash uchun qo'llash

```
<?php  
// Ma'lumotlar manbai ODBC ulanish 'ContactDB';connect q  
odbc_connect("ContactDB", "","","")  
or die("Couldn't connect to datasource.");  
// So'rov matnini yaratish
```

```

$query = "SELECT First_Name, Last_Name, Cell_Phone,
Email FROM Contacts";
// So'rovni tayyorlash
$result = odbc_prepare($connect,$query);
// So'rovni bajarish va natijalarni chiqarish
odbc_execute($result);
odbc_result_all($result, "BGCOLOR=#c0c0c0' border=1");
// Natijalarga ishlov berish yakunlandi, xotirani bo'shatish
odbc_free_result($result);
// Bog'lanishni yopish
odbc_close($connect);
?>

```

Natijada jadval quyidagi ko'rinishda bo'ladi:

| First_Name | Last_Name | Cell_Phone | Email |
|------------|------------|------------|----------------------|
| William | Gilmore | 7245551234 | wj@wjgilmore.com |
| Chef | Ale | 0398675309 | chef@phprecies.com |
| Pierre | BonneSoupe | 2125558743 | pierre@phprecies.com |

8.7.2-rasm. Web-brauzerda Contacts (ulanishlar) jadvalining mazmuni.

Savol va topshiriqlar

1. PHP tilining vazifasi.
2. PHP tilida o'zgaruvchilar va ma'lumot toifalari.
3. MySQL tilini tavsiflang.
4. MySQL tilining standart funksiyalari.
5. Bog'lanishning ochiq arxitekturasi.
6. PHPni boshqarish konstruksiyalari.

IX BOB. MA'LUMOTLAR BAZASINI ISHLATISH

9.1. Parallel qayta ishlashni boshqarish

Ko‘p foydalanimuvchi ma'lumot bazalari tashkilot va korxonalarining 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 muammojar tug‘dirmasligi talab etiladi. Bundan tashqari bir necha foydalanuvchilar so‘rovlari 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 operation 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 tuzulgan. 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 ishslash sharoitida ba’zan foydalanuvchilar ma'lumot bazasiga yakka holda foydalanish sharoitidagi natijalarni olish maqsadiga qaratilgan. Boshqa holdalarda foydalanuvchilarning kutilgan tarzda bir biriga ta’siri tushuniladi. Masalan, kompyuter tizimiga o‘z ma'lumotlarini kiritgan foydalanuvchi ma'lumot bazasidan bir qancha foydalanuvchi foydalanishiga qaramasdan bir xil natija olishi kerak. Shu bilan birga

foydaluvchi masalan, omborxona holati haqidagi axborotning oxirgi vaqtdagi hisobotini olishga harakat qilishi mumkin. Bunda shu paytda boshqa foydaluvchi 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, foydaluvchi parallel qayta ishlashni qattiq boshqarishi mumkin. Bunda u ma'lumot bazani boshqa foydaluvchilar 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 foydaluvchi 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. Masalan, yangi buyurtmani ro'yxatga olishda ma'lumot bazasi bilan bajarilgan ketma-ket ishlarni ko'rib chiqamiz. Bu quyidagi ketma-ketlikdan iborat:

1. Xaridorning to'lov puli oshishi bilan bog'liq bo'lgan yozuvni o'zgartirish kerak.

2. Sotuvchining komission puli oshishi bilan bog'liq bo'lgan yozuvni ham o'zgartirish kerak.

3. Ma'lumot bazasiga yangi buyurtma haqidagi yangi yozuvni qo'shish kerak. Masalan, bu ishlarni bajarishning oxirgi bosqichida fayl sohasida hajmi yetmaganligi munosabati bilan uchinchi qadam bajarilmadi. Bunda xaridor uchun bajarilmagan buyurtmaga haq to'lash bajariladi. Sotuvchi xaridorga yuborgan tovar uchun ustama haq oladi. Masaladan ko'rinish turibdiki, bu uch qadam bir butun bo'lib barchasi bajarilishi yoki birortasi ham bajarilmasligi kerak. Atomar tranzaksiyada ko'rsatilgan harakatlarda birortasi bajarilmasa ma'lumot bazasida o'zgarish amalga oshmaydi.

Tranzaksiyani bajarish chegaralarini ko'rsatish uchun tranzaksiyani boshlashga, tranzaksiyani saqlashga va tranzaksiyani qay-

tarishga komanda berish zarur. Har xil MBBTda bu komandalar formasi bo'yicha farqlanadi. Agar bir vaqtida 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. Ma'lumotlarni samarali himoya qilish uchun mos nazorat vositalari talab etiladi. MB muhim korporativ resurs hisoblanadi, shuning uchun uni himoya qilish katta ahamiyatga ega. MBda quyidagi xavflar mavjud:

- ma'lumotlarni o'g'rilarash va almashtirish;
- ma'lumotlarning maxfiyligini saqlash (sirni oshkor qilmashlik);
- ma'lumotlar bazasining bir butunligini saqlash;
- ma'lumotlar bazasiga murojaat qilishdan mahrum bo'lish.

MBda bo'ladigan xavflarni bartaraf qilish fizik nazoratdan boshlanib, administrativ tashkiliy ishlar bilan tugaydi. Hozirgi kunda kompyuter tizimlarini nazorat qilishni xilma-xil vositalari mavjud. Shu bilan birga MB bilan ishlaydigan MBBTni xavfsizligi muhim. Bu esa o'z navbatida MBBT ishlaydigan operatsion tizim xavfsizligi bilan bog'liq MBsini himoyalashda quyidagi usullardan foydalanish mumkin:

- foydalanuvchilarni mualliflashtirish;
- rezerv nuxsalar yaratish va tiklash;
- bir butunlikni ta'minlash;
- MBsiga himoyalash parollarini qo'yish.

9.2. Ishonchli loyihalash va administratsiyalash

MBBTni dasturiy ta'minot bazasida aniq AATlarni yaratish va ishonchli loyihalash texnologiyalari MBBT modeli va ma'lumotlar xavfsizligi texnologiyasi asosini qo'llab-quvvatlaydigan tizimlarining informatsion infrastrukturasida va kirishni cheklovchi qism tizimlarida logik xatolarni bartaraf qilishga yo'naltirilgan. Bu munosabatlarda asosiy va keng tarqalganlaridan biri strukturali-funksional yondashuv hisoblanadi. Foydalanuvchilar (subyektlar) va axborot tizimlari obyektlari (ma'lumotlar bazasi) ko'p sonli bo'lganda, kirishni cheklash sxemasi juda murakkab va chalkash bo'lishi mumkin. Bu administratsiyalashda qiyinchiliklar yaratadi va logik xatolar bo'lishiga imkon tug'diradi. Strukturali-funksio-

nal yondashish doirasida bunday xavfni bartaraf etish uchun ishchi guruh texnikasi qo'llaniladi.

Ishchi guruh ma'lumotlar bazasiga (o'xhash operatsiyalarni bajaruvchi) qandaydir umumiylar texnologik aloqador bo'lgan va umumiylar ma'lumotlarga nisbatan konfidensiallikning parametrlariga yaqin bo'lgan foydalanuvchilarni birlashtiradi. Sistema administratori ishchi guruhlarni belgilangan identifikatsiyaga va huquqlar yig'indisiga ega bo'lgan yoki ishchi guruhlar uchun maxsus ro'yxatlarni yaratishga ega bo'lgan foydalanuvchilar jamoasi ko'rinishida ishchi guruhlar yarata oladi. Har bir foydalanuvchi, albatta, qaysidir ishchi guruhga a'zo bo'lishi shart. Ishchi guruh uchun belgilangan huquqlar avtomatik ravishda hamma foydalanuvchilarga, qaysiki, kirishni cheklashni zonal-funksional prinsipining ba'zi elementlari bo'lgan guruh a'zolariga tatbiq qilinadi. Qo'shimcha ravishda har bir foydalanuvchini ro'yxatga olish yozuvida huquqlari aniqlashtirilgan va konkretlashtirilgan bo'lishi mumkin.

Ko'p hollarda bunday yondashish tizimda kirish subyektlarini sezilarli kamaytirish imkonini beradi, kirishni cheklash sxemasi ni anchagina oddiyashtiradi, «yorqin» va boshqaruvchan qiladi, va shu bilan birga, konkret foydalanuvchini konkret obyektga noto'g'ri kirish imkonini berish kabi logik xatolar bo'lishi ehtimalini pasaytiradi, konkret foydalanuvchini qator obyektlarga kirish huquqi oshib ketishini chegaralaydi, kirishga ortiqcha huquqlar berilishiga yo'l qo'ymaydi va hokazo. Ma'lumotlar bazasidagi jarayonlarda ishchi guruhlar texnologiyasida foydalanuvchi ham, ishchi guruh ham bir xil belgi bilan belgilanadi va shu asnoda MBBTning xavfsizlik yadrosi bu ikkala belgining haqiqiyligini tekshiradi.

Ishchi guruh texnologiyasi asosida kirish tizimini loyihalash «yuqoridan» (deduktiv), «pastdan» (induktiv) olib borilishi mumkin. Birinchi (deduktiv) usulda avvalo funksional struktura tahlili va foydalanuvchilar (subyektlar) tashkiliy ierarxiyasida ishchi guruhlar tashkil etiladi va kirishga guruhlar yo'naliishi hosil qilinadi. Keyin har bir foydalanuvchi uni tizimda ro'yxatga olishda uning funksiyalariga javob beruvchi bitta yoki bir nechta guruhlarga kiritiladi, va nihoyat, oxirida har bir foydalanuvchi uchun uning funksional talab xususiyatlari hamda ishonchli xarakteristikalarini tahlil qilinadi. Zarurat bo'lganda kirishga individual qo'shimcha yo'naliishlar beriladi. Bunday yondashish kirishda

xato topshiriqlar bo‘lish ehtimolini kamaytiradi va kirish tizimini ta’minlaydi, lekin o‘z navbatida subyektlarni obyektlarga kirishining individual huquqlari va guruhli «dublirovaniye» va subyektning har xil guruhlarda qatnashishi orqali bir xil obyektlarga kirish imkoniyatlari haddan tashqari ko‘pligi muammosini hosil qilishi mumkin.

Ikkinchi (induktiv) ishchi guruhlarni loyihalashtirish usulida birinchi navbatda subyektlarning (foydanuvchilarining) individual holda obyektlarga kirish topshirig‘ini amalga oshiriladi. Topshiriq foydanuvchilarini surishtirish, ularning funksional talablarini tahlil qilish va ishonchli funksional xarakteristikalar asosida amalga oshiriladi yoki obyektlar egalaridan subyektlarning individual so‘rovleri orqali ham tizim administratori tomonidan bajariladi. Keyin, albatta, tizim administratori tomonidan har xil subyektlarning kirishida umumiyligi va o‘xhash topshiriqlar tahlili amalga oshiriladi va shu asosda ishchi guruhlarga birlashildi. Ajralib turuvchi kirishning umumiyligi o‘rnatmalaridan kirishning guruhli ko‘rinishi sifatida foydalaniladi. Bunda ko‘p sonli subyektlar va obyektlar ishtirokidagi kirishning o‘xhashlik tahlili oddiy bo‘lmasal masala bo‘lib u tizim administratori tomonidan sezilarli darajada evrestik yechiladi.

Tizimni kuzatib borish va administratsiyalashtirish jarayonida ishonchlilikni va xavfsizlikni oshirishning qo‘sishimcha tashkiliy usuli umumiyligi administratsiya va xavfsizlik administratsiyalarini ajratish hisoblanadi. Umumiyligi administrator tizimni, axborot infrastrukturani – informatsion-logik sxema, obyektlarni (resurs va uskunalarini) konfidensialligini darajalashtirish, interfeys va dialogli elementlar, formalar, so‘rovnomalari kutubxonalarini, lug‘at-klassifikatsiyali baza, ma'lumotlarni rezervlashtirish va arxivlashtirishni yaratadi, qo‘llab-quvvatlaydi va boshqaradi.

Xavfsizlik administratori kirish-cheplash tizimini – foydanuvchilarining ishonchli xarakteristikalarini (ruxsatnomalari) aniq belgilangan ruxsatlarni, foydanuvchilarini kirish belgilarini tashkilashtirish va ro‘yxatga olishni tashkil qiladi va boshqaradi. Shunday qilib, aksiomatik prinsiplarni, modellarni va xavfsizlik siyosatini amalga oshirishni ta’minlovchi texnologik masalalar to‘plami, ish qurollari va vositalari nihoyatda keng. Bu o‘z o‘rnida obyektiv holda qandaydir yagona talablar shkalasi xavfsizligi ta’minlangan tizimlarni ishlab chiqarishni va baholashni talab qiladi.

9.3. MB administratorlari va foydalanuvchilari, ularning maqsadi, huquqi va majburiyatları

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 administratori 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 uskulnalariga 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 faydan iborat bo'lishidan va undan necha marotaba foydalanishdan qat'i 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 bat afsil ma'lumotiga ega bo'ladi, shu jumladan quyidagilarni o'z ichiga olgan holda:

- ma'lumotlar bazasi fayllariga kirgan nomalar;
- kirish ma'lumotlari va shu bilan bog'liq bo'lgan jadvallarning aniqlanishi;
- MBning xatolari va ishslash 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 zaxira ko'chirmalarini rejalashtirish.

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

Ma'lumotlar bazasi foydalanuvchilari uch asosiy kategoriya ajratiladi: yaratuvchilar, administratorlar va foydalanuvchilar. Yaratuvchilarning vazifasiga quyidagilar kiradi:

- ma'lumotlar bazasini loyihalashtirish va yaratish;
- ma'lumotlar bazasi strukturasini (tuzilishini) belgilangan talablarga muvofiq loyihalashtirish;
- eslash 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 administratorsiyalashtirish 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 strukurasini modifikasiyalash;
- 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;

- zaxira nusxalash va qayta tiklashni rejalahtirish;
- axborotni saqlash qurilmalaridagi arxiv ma'lumotlarini qo'l-lab-quvvatlash;
- zaxiraviy 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.

9.4. Axborot tizimlarini administratsiyalash

«Administratsiyalashtirish» atamasi umumiy ma'noda quyidagilarni yaratishdagi jarayonlar kompleksini aniqlab beradi: AATni funksiyalashtirishda ishonchlilik va samaradorlikni ta'minlash bilan bog'liq bo'lgan, AATni eksploatatsiya qilish va undan foydalanish; har xil kategoriyadagi foydalanuvchilarning ma'lumotlari xavfsizligini ta'minlash va jamoaviy ishini tashkil qilish.

Bu jarayonlar kompleksi yechilishi kerak bo'lgan masalalariga qarab, quydagi guruhlarga bo'linadi:

- tizimning strukturasi va predmetli sohasi jarayonlari, AATning tuzilishi, interfeysi va texnologik komponentlarini sozlashni ta'minlash va qo'llab-quvvatlash;
- ma'lumotlarning ishonchliliginini va saqlanishini ta'minlash;
- foydalanuvchilarni umumiy ma'lumotlar bilan jamoaviy ishlashini ta'minlash va tashkil etish.

Birinchi yo'naliш AAT ni loyihalashtirish va ekspluatatsiyaga kiritish bosqichlarida tizim administratorining qatnashishini belgilab beradi. Administrator bunda MBBT tanlash bo'yicha va uning yaratilayotgan ma'lumotlar bankining u yoki bu konseptual sxemasi komponenti xususiyatlarini realizatsiyasi ko'rinishida yaratuvchilar jamoasi eksperti bo'lib qatnashadi va yana tipovoy so'rovlar yaratish, ma'lumotlarning kirishi va chiqishi uchun ekranli formalari, hisobotlar qoliplari jarayonlarida qatnashadi. Loyihalashtirish bosqichida, va ayniqsa, tizimni ma'lumotlar bilan to'ldirish bosqichining keyingi ekspluatatsiyalash jarayonida tizimni ma'lumotlar bilan to'ldirishda tizim administratori ma'lumotlar bazasining loyihalashtirilgan ichki sxemasining adekvatliligi va samaradorligi tahlilini o'tkazadi va zarurat bo'lganda unga o'zgartirishlar kiritilishi mumkin. Bunday funksiyalarni ta'minlash uchun ay-

rim MBBTlar tez ta'mir qiluvchi tahlillagichlar ko'rinishidagi va ma'lumotlar bazalarining optimalliligi bor maxsus dasturiy asbob-uskunala ega bo'ladilar. Axborot tizimining predmetli sohasini ajralib turuvchi tomonlarini namoyon qiladigan funksiyalar guruhiga lig'aviy klassifikatsiyalashtirish bazasini tashkil etish va qo'llab-quvvatlash kiradi (lug'atlar, ma'lumotlar majmuasi, kalit so'zlar).

Yana bir muhim funksiya, informatsion tizimni ekspluatatsiyaaga kiritish bosqichida tizimni ma'lumotlar bilan boshlang'ich to'ldirish hisoblanadi. Buning uchun ko'p hollarda boshqa informatsion tizimlarning ma'lumotlaridan foydalanish mumkin bo'ladi, ular boshqa fizik formatlarda bo'lishi mumkin. Tashqi manbalardan ma'lumotlar importi, qoida bo'yicha, foydalaniladigan AAT MBBTlarning imkoniyatlariga qarab yuqori malakani talab etadi, AATdagi ma'lumotlar logik va fizik tashkil etuvchilari haqida to'liq taassurotlarga ega bo'lishi kerak va natijada bu tizim administratori tomonidan bajariladi. Relatsion MBBTlar asosidagi faktografik AATlarda shunga o'xshash masalalar jadvallari yaratish so'rovlari asosida yechiladi, bunda ma'lumotlar manbasi tashqi bazalarda yoki boshqa fayllarning formatlarida bo'ladi (elektron jadvallar, UFOD formatidagi matnli fayllar). Shunga ko'ra, AAT funksiyasining o'ziga xosligi teskari masalaning yechilishini ham talab etishi mumkin. Ma'lumotlarni tashqi tizimlarga va boshqa formatlarga eksporti masalasi ko'p hollarda yuqoridagilar kabi AAT administratorlari tomonidan yechiladi.

Ma'lumotlarning ishonchlilagini va saqlanish xususiyatini ta'minlash AAT administratorining asosiy majburiyatlaridan biri bo'lib, o'z navbatida quyidagi qator texnologik va profilaktik masalalar yechimini o'z ichiga oladi:

- rejalshtirish, konfiguratsiyalash va ma'lumotlar faylini o'z ichiga olgan tashqi xotiradan foydalanish qurilmalari tizimini qo'llab-quvvatlash;
- ma'lumotlarni arxivlashtirish va rezervlashtirish;
- to'xtalishlar va buzilishlardan keyin ma'lumotlarni qayta tiklash;
- ma'lumotlarning butunligini tekshirish va saqlash.

Ma'lumotlar bazalari fayllarining katta hajmliligi, oldin ham aytganimizdek, ularning diskli (tashqi) xotira uskunalarida tizimning ma'lumotlar bazalariga fayllar joylashuvি sxemalarini aniqlash AATni loyihalashtirishda alohida masaladir.

Texnologik xususiyatlar ta'sirida (real vaqt tizimlari) qator tavsiyalarga ko'ra ma'lumotlar bazalarining saqlanishliligi va ishga layoqatliligi o'ta kritik jarayon (voqelik, aspekt) hisoblanadi yoki ma'lumotlarning ma'noliligiga ko'ra ham xuddi shunday hisoblanadi. Bunday hollarda ma'lumotlarni issiq rezervlashtirish deb ataluvchi yondashuv qo'llaniladi. Bunda, diskli xotiraning ikkita alohida tizimida joylashtiriladigan, ikki bir xil ko'rinishdagi ko'zguli va parallel funksiyalanuvchi nusxalar bo'ladi.

Boshqa holatlarda ma'lumotlar saqlanishini ta'minlash uchun ma'lumotlarni arxivlashtirish va rezervlashtirish operatsiyasi qo'llaniladi. Ko'pgina hollarda arxivlashtirish, ularni uzoq muddatga kompakt holda saqlash maqsadida, qoidaga muvofiq, tashqi olib qo'yiluvchi tashuvchilar bilan amalga oshiriladi. Ma'lumotlarni arxivlashtirish funksiyalari ayrim hollarda MBBT ni o'zining ichki funksiyalari ro'yxatiga ham kirishi mumkin.

Qoidaga ko'ra, ma'lumotlarni rezervlashtirish ma'lumotlarni, bilgan holda, siqib qo'yishni nazarda tutmaydi, aksincha, texnologik yoki boshqa maqsadlarda fayllarni maxsus nusxalarini yaratish orqali amalga oshiriladi.

Texnologik maqsadlardan tashqari arxivlashtirish ham, rezervlashtirish ham yana bitta o'ta muhim operatsiya AAT administratori tomonidan amalga oshirilishi uchun profilaktik maqsadni ham bajaradi, bu to'xtalishlar va buzilishlardan keyin ma'lumotlarni qayta tiklash operatsiyasi. Ma'lumotlar bazalarining rezerv va arxiv nusxalarining borligi ma'lumotlarning asosiy fayli (fayllari) ishdan chiqqanda tizimning ishga layoqatliligin qayta tiklash imkonini beradi. Lekin, bu holda oxirgi arxivlashtirish va rezervlashtirishdan keyingi kiritilgan ma'lumotlar yoki ularning o'zgartirishlarining bir qismi yo'qotilishi mumkin. Bunday hollarda asosan mijoz-server tizimlari tomonidan realizatsiya qilinadigan umumiy ma'lumotlarni jamoaviy ishlatilishida bunday xavf bo'ladi. Shuning uchun «Mijoz-server» texnologiyasida ishlovchi sanoat MBBTlarida, qoida bo'yicha, alohida tashuvchida, asosiy ma'lumotlardan alohida o'rnatiladigan (joylashtiriladigan) ma'lumotlar bazasi joriy o'zgarishlarining maxsus journali yuritishi ko'p hollarda nazarda tutiladi. Belgilanganidek, bunday yondashish jurnallashtirish deb ataladi. O'zgarishlar jurnalida foydalanuvchi bilan ma'lumotlar bazasi o'rtasida bo'ladigan baracha manipulyatsiyalar uzluksiz aks ettirilib va protokollashtirilib boriladi. Natijada har qanday to'xtalishlarda arxiv nusxasi va

o‘zgarishlar jurnali yordamida tizim administratori to‘xtash momentigacha bo‘lgan ma’lumotlarni to‘laligicha qayta tiklay oladi.

Administrator funksiyalarini qayta tiklashni ta’minlovchi ro‘yxatga shuningdek, tashqi xotiraning diskli tashuvchilari profilaktikasi ham kiradi. Bu profilaktika tezkor tizimning maxsus dasturli-uskunali utilitlari bilan ta’minlangan – disklar holatini tekshirish, defragmentatsiyalash va hokazo. Yuklangan majburiyatlar alohida yo‘nalish tayyorgarligiga ega tizim dasturchilari kabi administratorlarga qo‘srimcha kasbiy tayyorgarlik talablarini qo‘yadi.

Ma’lumotlar butunligini tekshirish va qo‘llab-quvvatlash ham administratorlarning ajralmas funksiyasi hisoblanadi va konkret ma’lumotlar bazasida ma’lumotlar va aloqalarning butunligini cheklashni qo‘llab-quvvatlovchi MBBTni himoyalovchi mexanizmlarni sozlashni va funksiyalashni ta’minlaydi. Qoida bo‘yicha, ko‘pgina MBBTlarda ma’lumotlarni qo‘llab-quvvatlash va butlini nazorat qilishning o‘rnatilgan avtomat mexanizmlari bor. Shu bilan, qator hollarda, bog‘langan yozuvlarni kaskad yangilash va chiqarib tashlash misolidagi kabi, logik predmet maydoni bunday bog‘lanishlar butunligini ta’minlashni o‘rnatish imkonini bermaydi. Bunda ma’lumotlar bazalarini funksiyalashtirish samaradorligini pasaytiradigan, aloqalarning «uzilishlar» ko‘rinishidagi ko‘p muammolar, foydalanuvchilarni ma’lumotlar bazasi bilan ishlashi jarayonida paydo bo‘lishi mumkin. Bunday hollarda ma’lumotlar bazalari administratorining funksiyalaridan biri aloqalar butunligining nazorati va shunga o‘xhash muammolarni o‘z vaqtida bartaraf etishdan iborat. Ayrim MBBTlarda bunday masalalarni yechish uchun maxsus rejimlar va «taftish qilish» mexanizmlari hamda ma’lumotlar bazalari butunligini tiklash ko‘zda tutilgan. Boshqa MBBTlarda yuqoridaq masalalar so‘rovlar texnikasi yoki ishlov berish orqali yechiladi, shuningdek maxsus dasturiy utilitlarning AAT administratorlarining o‘zlari ham yechadi, qachonki ma’lumotlarni saqlashni vaqtinchalik reglamentlari (belgilangan vaqt ichida) o‘rnatilgan hollarda, ma’lumotlar taftishining o‘xhash masalalari ham administrator tomonidan yechiladi va «eskirgan» ma’lumotlar o‘z vaqtida yangilanishi kerak bo‘ladi yoki sistemadan chiqarib tashlanadi.

AAT administratori funksiyalarining katta kompleksi foydalanuvchilarni umumiyl ma’lumotlar bilan jamoaviy ishlashini ta’minlash va tashkillashtirish bilan bog‘liq tizimning bo‘lg‘usi

administratorining to‘la ishtirokidagi AATni loyihalashtirish bosqichida AATni funksiyalashtirish va undan foydalanishning tashkiliy sxemasi ishlab chiqariladi.

Sistemaning umumiyligi ma’lumotlarini ishlanishiga va undan foydalanishga kiritilishiga javobgar bo‘lgan ishchilarining funksional guruhlari texnologik predmetli soha va yechilishi kerak bo‘lgan masalalar doirasining xususiyatlaridan kelib chiqqan holda aniqlanadi. Shu asosda tizim foydalanuvchilarining ro‘yxatlari va sxemalari quriladi, ularning aniq funksiyalari, huquqlari aniqlanadi, muhim bo‘lgan texnologik va interfeys elementlari tuziladi (kiruvchi va chiquvchi ekranli formalar, so‘rovlar, hisobotlar shabloni va hokazolar). AAT administratori, ma’nosini bo‘yicha, tizimni ekspluatatsiya qiluvchilarining va abonent foydalanuvchilarining texnologik jarayonlarini tashkil etishda aynan tashkil etuvchi va rahbar hisoblanadi.

Sistema administratorining bu kompleks yo‘nalishi alohida, lekin boshqa funksiyalar bilan chambarchas bog‘langan bo‘lib, ma’lumotlarga ruxsat etilmagan kirish va ma’lumotlarni himoyalashni cheklash tizimini yaratish hamda qo‘llab-quvvatlashdan iborat.

Foydalanuvchilar va xizmat qiluvchi personalning tizimi hamda funksiyalar sxemasi masalalari va mas’ulligini asosida administrator tomonidan har xil foydalanuvchilarining va tashqi jarayonlarning kirishga kriteriyasi bo‘yicha ma’lumotlar bazasining obyektlarni kategoriyalashtirish sxemasi administrator tomonidan quriladi va qo‘llab-quvvatlanadi, shu sxema bilan joriy boshqaruv amalga oshiriladi va ma’lumotlarga kirish xavfsizligi hamda cheklashni olib tashlashni nazarda tutgan ma’lumotlarga ishlov berish jarayonining auditini amalga oshiriladi.

O‘z funksiyalarini va imkoniyatlari darajasida MBBTning o‘sishi, qoidaga ko‘ra, administratorni asosiy funksiyalar va masalalar bilan ta’minlovchi maxsus uskunalariga ega, oldin ham aytib o‘tilgan, tez ta’sirlilik va optimallik tahlili, arxivlashtirish utilitlari ma’lumotlarga kirishni va himoyalashni cheklashni olib tashlovchi kichik tizim hisoblanadi.

Shuni ham aytish kerakki, AAT administratorlari tashkil etilish bo‘yicha axborot xizmatlar ishlarining alohida shtatli kategoriyasiga kiradi va shu xizmatlar rahbarlariga bo‘ysunadi yoki ko‘p hollarda u korxona, tashkilot, alohida tashkilot bo‘limlarining axborot funksiyasini bajaradi.

ILOVALAR

ILOVA A

I. MBBT MS ACCESS 2010 ISHLASH

КО'NIKMALARINI EGALLASH

Microsoft Access da ma'lumotlar bazasi jadvallar, so'rovlar, formalar, hisobotlar, murojaat sahifalar, makroslar va modullar-dan tashkil topadi. Ma'lumotlar bazasini loyihalash ikkita asosiy fazadan tashkil topadi:

- mantiqiy modellashtirish;
- fizik modellashtirish.

Ma'lumot bazasini loyihalash jarayoni quyidagi bosqichlardan tashkil topadi:

1. Axborotlarni yig'ish;
2. Obyektlarni identifikatsiyalash;
3. Obyektlarni modellashtirish;
4. Har bir obyekt uchun obyektlar ma'lumot toifalarini identifikatsiyalash;
5. Munosabatlarni identifikatsiyalash;
6. Normallashtirish;
7. Fizik modelga almashtirish;
8. Ma'lumotlar bazasini yaratish.

1–6-bosqichlar mantiqiy modellashtirish fazasini aniqlaydi.
7–8-bosqichlar fizik modellashtirish fazasini tashkil qiladi.

Ma'lumotlar bazasini yaratish

Tajriba ishi 1: jadvallar

MS ACCESS 2010 quyidagi komanda bilan ishga tushiramiz.
меню Пуск > Программы > Microsoft Office > Microsoft Office Access 2010.

ACCESS ma'lumot bazasini yaratish menyusi chiqadi.

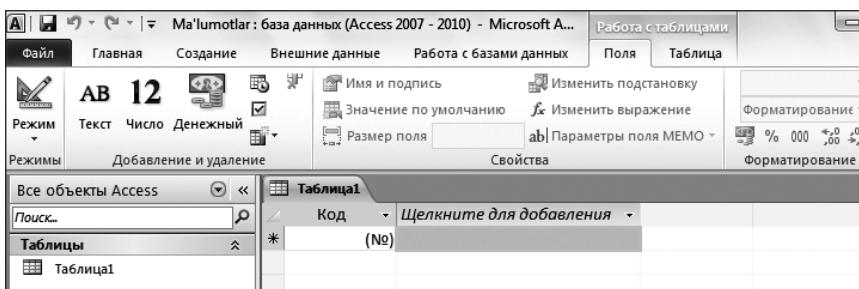


I.1.1-rasm. Access ning menu darchasi.

2. Darchani «**Создать**» bo‘limida «**Новая база данных**» (**Blank Database**) buyrug‘ini tanlang.

3. «**Новая база данных**» (**Blank Database**) bo‘limini «**Имя файла**» (**File Name**) maydonida fayl nomini kiriting va ma’lumot bazasi saqlanishi kerak bo‘lgan papkaga yo‘l ko‘rsating. Sukut bilan (standart holda) ma’lumot bazasi «*Му документы*» (My documents) papkasida saqlanadi. Yo‘lni o‘zgartirish uchun «**Имя файла**» (**File Name**) maydonidan chapda joylashgan knopkani bosish kerak.

4. «**Создать**» (**Create**) tugmasini bosing. Yangi ma’lumot baza yaratiladi va tablitsa rejimida yangi jadvalli ma’lumot baza darchasi ochiladi (I.1.2-rasm).



I.1.2-rasm. Ma’lumot baza darchasi.

Zarurat bo‘lsa yangi ma’lumot bazasi andozalar (shablonlar) yordamida yaratilishi mumkin. Bundan tashqari Microsoft Office Online veb tugunidan qo‘srimcha andozalar (shablonlar) yuklash imkoniyati mavjud.

Jadvallar yaratish. Ma’lumotlar bazasini yaratishda ma’lumotlar jadvallarda saqlanadi. Yangi ma’lumotlar bazasi yaratilganda yangi jadval avtomatik ravishda paydo bo‘ladi. Jadvallarni shuningdek, mavjud ma’lumotlar bazasiga yangi jadval qo’shish bilan, import qilish yoki boshqa ma’lumotlar manbai jadvallariga (masalan, MS Excel, tekstli fayl yoki boshqa ma’lumotlar bazasi jadvallari) murojaat qilish bilan yaratish mumkin.

Jadvallarni konstruktor rejimida yaratish

Vazifa: Konsrtuktor rejimida student va university jadvallari dan tashkil topgan TA’LIM ma’lumotlar bazasini yarating.

Yaratilayotgan ma’lumotlar bazasi jadvallarining mantiqiy loyihasini yaratib olamiz.

Jadvallarni mantiqiy loyihalash

Student jadvali

| Maydon nomi | Toifasi | O'lchami | Boshqa xossalari |
|-------------|------------|----------|------------------------|
| Stud_id | Счетчик | | kalit |
| Surname | Текстовый | 25 | Majburiy, indekslanadi |
| SName | Текстовый | 20 | Majburiy, indekslanadi |
| Kurs | Числовой | byte | Majburiy, indekslanadi |
| Sity | Текстовый | 15 | Majburiy, indekslanadi |
| Birthday | Дата/время | | Majburiy |
| Univ_id | Числовой | целое | Ikkilamchi kalit |

University jadvali

| Maydon nomi | Toifasi | O'lchami | Boshqa xossalari |
|-------------|-----------|----------|------------------------|
| Univ_id | Числовой | целое | kalit |
| Univ_name | Текстовый | 15 | Majburiy |
| Rating | Числовой | целое | Majburiy, indekslanadi |
| City | Текстовый | 20 | Majburiy, indekslanadi |

Student jadvali

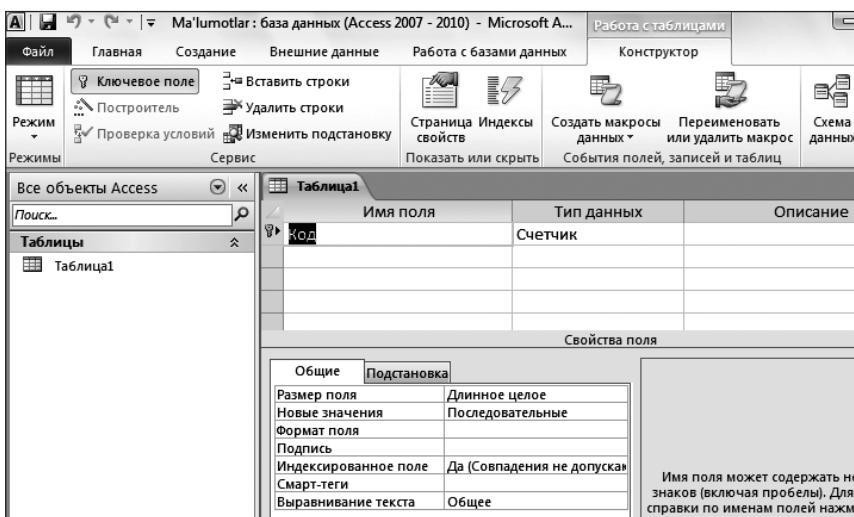
| STUD_ID | SUR-NAME | SNAME | KURS | CITY | BIRTHDAY | UNIV_ID |
|---------|----------|--------|------|----------|------------|---------|
| 1 | Imamov | Ilhom | 1 | Angren | 03.12.1992 | 10 |
| 2 | Po'latov | Po'lat | 3 | Qarshi | 01.12.1990 | 18 |
| 3 | Salimov | Sobir | 4 | Toshkent | 07.06.1989 | 22 |
| 4 | Kenjayev | Botir | 2 | Buxoro | 08.12.1991 | 18 |
| 5 | Zokirova | Oqila | 2 | Navoiy | 01.05.1991 | 15 |
| 6 | Pardayev | Asil | 3 | Andijon | 05.11.1989 | 11 |
| 7 | Komilov | Parda | 5 | Nukus | NULL | 14 |
| 8 | Latipov | Ahmad | 3 | Andijon | 01.12.1991 | 10 |
| 9 | Polatov | Asqar | 4 | NULL | 05.08.1991 | 22 |
| 10 | Bahromov | Vohid | 5 | Andijon | 07.01.1990 | 32 |

University jadvali

| UNIV_ID | UNIV_NAME | RATING | CITY |
|---------|-----------|--------|----------|
| 22 | TATU | 606 | Toshkent |
| 10 | TDTU | 296 | Andijon |
| 11 | NDU | 345 | Namangan |

| | | | |
|----|------|-----|-----------|
| 32 | SDU | 416 | Samarqand |
| 14 | NATU | 326 | Nukus |
| 15 | FDU | 368 | Fargona |
| 18 | AATU | 327 | Andijon |

1. MS ACCESS ni ishga tushiramiz.
2. Ma'lumot baza darchasida «**Создать» (Create)** vkladkasini tanlab, «**Таблицы» (TABLE) gurugida «**Конструктор таблиц**» (Table Design) bosiladi.**



1.1.3-rasm. «Конструктор таблиц» darchasi.

3. «**Имя поля**» (Field Name) ustuniga yaratilayotgan maydon nomini kiriting va ENTERni bosing.

4. «**Тип данных**» (Data Type) ustuniga ma'lumotni talab qilingan toifasini ro'xatdan tanlab oling.

5. «**Описание**» (Description) ustuniga jadvalni berilgan maydonini vasifasi va ishlatishini tushuntiruvch matnni kiriting (1.2.2-rasm).

6. «**Свойства поля**» (Field Properties) maydonni talab qilin-gan xossalariini o'rnatning (I.1.3-rasm):

«**Размер поля**» (Field Size) maydon hajmi.

«**Формат поля**»

«**Число деятичных знаков**» — o'nli raqamlar soni va bosh-qalar.

| Имя поля | Тип данных | Описание |
|----------|------------|-------------------|
| Stud_id | Счетчик | Student kodi |
| Surname | Текстовый | Student familiysi |
| Sname | Текстовый | Student ismi |

I.1.2-rasm. Ma'lumot toifasini o'rnatish.

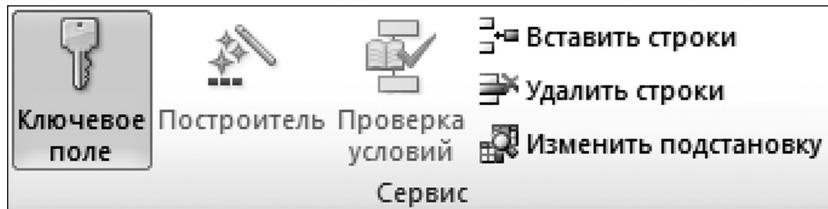
| Имя поля | Тип данных | Описание |
|----------|------------|----------------------------|
| STUD_ID | Счетчик | Student kodi |
| SURNAME | Текстовый | Student familiysi |
| SNAME | Текстовый | Student ismi |
| KURS | Текстовый | Student o'qiydigan bosqich |
| sadsd | Текстовый | |

I.1.3-rasm. Maydon xossalari o'rnatish.

7. Birlamchi kalitni o'rnatish.

7.1. «**Конструктор**» (**Design**) rejimida sichqoncha ko'rsatkichi bilan, kalit sifatida ishlatalishi mo'ljallangan maydon tavsifini belgilaymiz.

7.2. Uskunalar panelida «**Ключевое поле**» (**Primary key**) tugmasini cherting yoki maydon tavsifida sichqonchani o'ng tugmasini bosib, paydo bo'lgan kontekstli menyuda mos menyu punkti ni cherting (I.1.4-rasm.).



I.1.4-rasm. Servis guruhining yordamchi tugmalari.

| Имя поля | Тип данных | Страница свойств |
| --- | --- | --- |
| STUD_ID | Счетчик | Student kodi |
| SURNAME | Текстовый | Student familiyasi |
| SNAME | Текстовый | Student ismi |
| KURS | Текстовый | Student o\'qiyidigan |

I.1.5-rasm. Jadval strukturası.

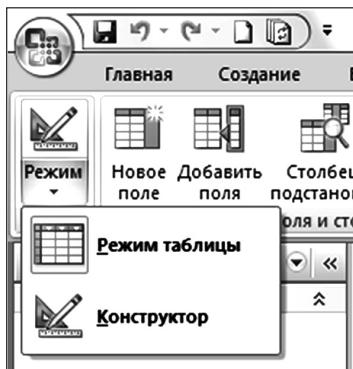
7.3. Murakkab kalitni aniqlashda, bir yo'la kalit tarkibiga kiruvchi barcha maydonlarni ajratamiz (masalan, ctrl tugmasini bosib turib ajratib) va ularga kalit tayyorlash 7.2-punktdagi kabi bajariladi).

8. Jadval tuzilishini saqlab qo'yish uchun quyidagi usullardan biridan foydalanamiz:

- «Панель быстрого доступа»да «Сохранить» (Save) tugmasini bosish bilan;
- «Документ» vkladkasida sichqonchani o'ng knopkasini bosamiz va silqib chiqqan kontekstli menyuda «Сохранить как» (Save as) buyrug'ini tanlaymiz;

Файл vkladkasini tanlab, so'ngra «Сохранить как» (Save as) buyrug'ini tanlaymiz. Yaratilgan jadval nomini kiritamiz va «Ok» tugmasi bosiladi, darcha yopiladi.

9. Jadvalga ma'lumotlarni kiritish. MS ACCESSda tuzilmasi (strukturasi) yaratilgan jadvalni to'dirish uchun **ВИД > РЕЖИМ ТАБЛИЦЫ** buyrug'ini bajarib, bo'sh jadval yaratamiz. Jadvalni ma'lumitlar bilan to'ldirish WORD matn tahririyatida jadvalga ma'lumotlar kiritish kabi amalga oshiriladi.



I.1.6-rasm. «Режим таблицы» da jadval yaratish.

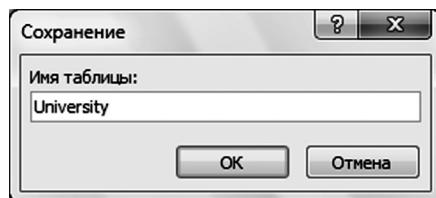
| Stud_id | Surname | Sname | Kurs | City | Birthday | Univ_id | Щелкните для изменения |
|---------|----------|-------|------|----------|------------|---------|------------------------|
| 1 | Imamov | Ilhom | 1 | Angren | 03.12.1992 | 10 | |
| 2 | Polatov | Polat | 3 | Qarshi | 01.12.1990 | 18 | |
| 3 | Salimov | Sobir | 4 | Toshkent | 07.06.1989 | 22 | |
| 4 | Kenjaev | Botir | 2 | Buxoro | 08.12.1991 | 18 | |
| 5 | Zokirova | Oqila | 2 | Navoi | 01.05.1991 | 15 | |
| 6 | Pardaev | Asil | 3 | Andijon | 05.11.2012 | 11 | |
| 7 | Komilov | Parda | 5 | Nukus | 06.12.1988 | 14 | |
| 8 | Latipov | Ahmad | 3 | Andijan | 01.12.1991 | 10 | |
| 9 | Polatov | Asqar | 4 | NULL | 05.08.1991 | 22 | |
| 10 | Bahromov | Vohid | 5 | Andijon | 07.01.1990 | 32 | |

I.1.7-rasm. Jadvalni ma'lumotlar bilan to'ldirish.

Ma'lumotlar bazasining ikkinchi Universitet jadvalini tuzilmasini quramiz va saqlaymiz (I.1.8. ... I.1.11-rasmlar).

| Имя поля | Тип данных | Описание |
|-----------|------------|------------------------------|
| Univ_id | Числовой | Universitet kodi kalit |
| Univ_name | Текстовый | Universitet nomi kodi |
| Rating | Числовый | Universitet reytingi |
| City | Текстовый | Universitet joylashgan shaha |

Rasm I.1.8. Universitet jadvali tuzilmasi.



I.1.9-rasm. Universitet jadvali tuzilmasini saqlash darchasi.

The screenshot shows the Microsoft Access 2010 interface with the 'University' table selected. The ribbon tabs at the top include 'Файл', 'Главная', 'Создание', 'Внешние данные', 'Работа с базами данных' (selected), 'Конструктор', and 'Справка'. The 'Режимы' dropdown on the left shows 'Режимы' and lists 'Таблицы', 'Student', and 'University'. The main area displays the 'University' table structure:

| | Имя поля | Тип данных | Описание |
|-----------|-----------|-------------------------------|----------|
| Univ_id | Числовый | Universitet kodi kalit | |
| Univ_name | Текстовый | Universitet nomi kodi | |
| Rating | Числовый | Universitet reytingi | |
| City | Текстовый | Universitet joylashgan shahar | |

A 'Свойства поля' (Field Properties) dialog is open for the 'Univ_id' field, showing the 'Общие' tab. The properties listed are:

| Общие | Подстановка |
|-------------------------|--------------------------------|
| Размер поля | Целое |
| Формат поля | |
| Число десятичных знаков | Авто |
| Маска ввода | |
| Подпись | |
| Значение по умолчанию | |
| Условие на значение | |
| Сообщение об ошибке | |
| Обязательное поле | Да |
| Индексированное поле | Да (Совпадения не допускаются) |
| Смарт-теги | |
| Выравнивание текста | Общее |

To the right of the properties, a note states: 'Тип данных определяет значения, которые можно хранить в этом поле. Для получения справки по типам данных нажмите клавишу F1.'

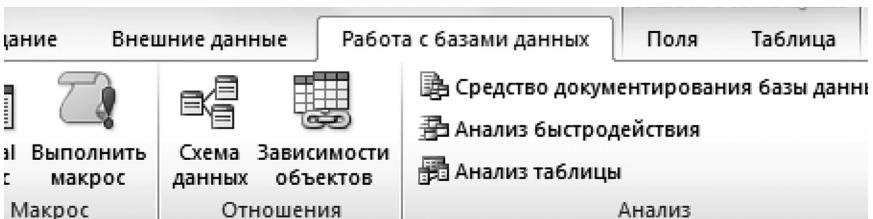
I.1.10-rasm. Saqlangan Universitet jadvali tuzilmasi

I.1.11-rasm. Saqlangan Universitet jadvali.

Jadvallar orasida aloqa o'rnatish. Odatda, ma'lumotlar bazasi bir nechta jadvaldan tashkil topadi. Bu jadvaldagi ma'lumotlarni birlashtirib o'rganish zarurati paydo bo'lib turgani uchun jadvallar orasida aloqa o'rnatiladi. Bu amalni bajarish uchun MS Access da vositalar mavjud. Odatda, ma'lumot baza jadvallari orasida- gi aloqalar ma'lumotlar sxemasi (**«Схема данных»**) yordamida o'rnatiladi.

Vazifa: Tajriba ishida yaratilgan ma'lumotlar bazasi jadvallari uchun ular orasida aloqalar o'rnatish yo'li bilan ma'lumotlar sxe- masini ishlab chiqing.

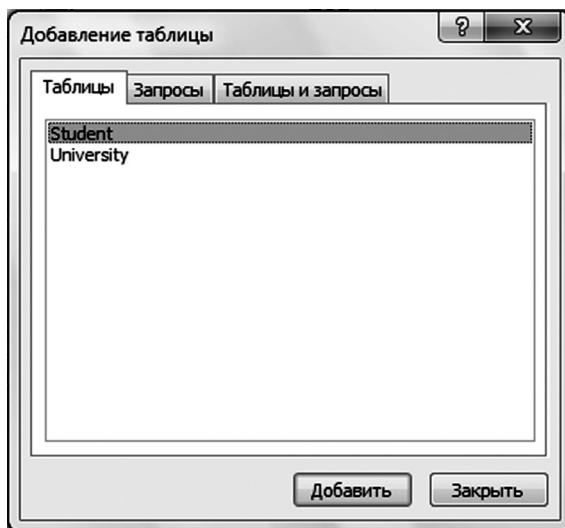
1. Jadvallar orasida aloqalarni o'rnatish uchun **«Работа с базами данных»** (**Database Tools**) vkladkasini **«Отношения»** (**Relationships**) guruhidagi **«Схема данных»** (**Relationships**) tugmasini bosamiz. Bunda **«Схема данных»** (**Relationships**) darchasi ochilib, unda barcha mavjud bog'lanishlar ko'rsatiladi. Agar aloqa o'rnatilmagan bo'lsa bu darcha Access jadvallarni qo'shish haqida so'rov chiqaradi (I.1.12-rasm). **«Схема данных»** (**Relationships**)ni **«Связи»** (**Relationships**) guruhidagi **«Таблица»** vkladkasi orqali ham ochish mumkin.



I.1.12-rasm. «Схема данных»ni tanlash variantlari.

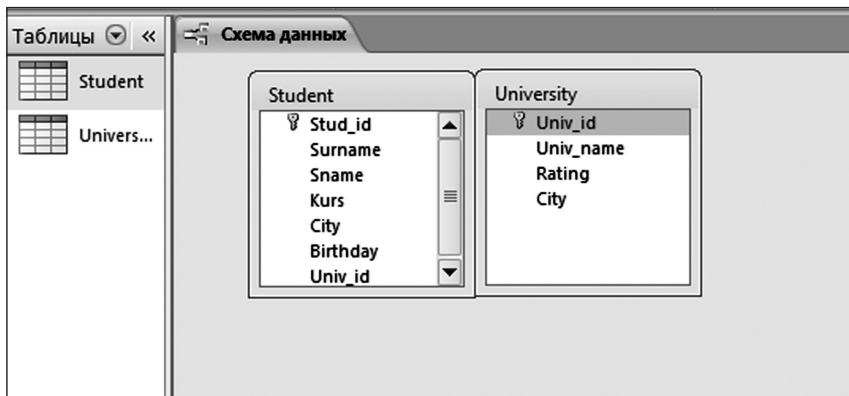
2. Jadvallar orasidagi aloqani **«Схема данных»** darchasi yordamida yoki **«Список полей»** sohasidan jadvalga maydonlarni ko'chirib o'tkazish yo'li bilan yaratish mumkin. Bunda umumiy maydon toifalari bir xil bo'lishi kerak (nomlari har xil bo'lsa ham).

3. Agar ma'lumot bazasida aloqalar o'rnatilmagan bo'lsa, avtomatik ravishda **«Добавление таблицы»** (**Show table**) muloqot darchasi ochiladi (I.1.13-rasm). Agar darcha ekranda akslansmasa **«Конструктор»** vkladkasining **«Связи»** (**Relationships**) guruhidagi **«Отобразить таблицы»** (**Show table**) tugmasini bosing.



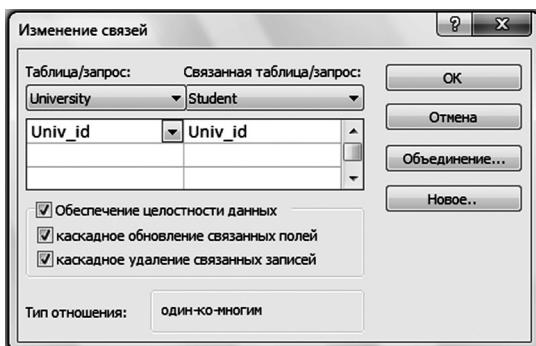
I.1.13-rasm. Bog'lanayotgan jadvallarni tanlab olish.

4 Bir yoki bir necha jadval yoki so'rovlarni tanlab oling, so'ngra, «Добавить» (Add) tugmasini bosing. «Схема данных» darchasida jadval va so'rovlarni qo'shishni yakunlagandan so'ng «Закрыть» (Close) tugmasini bosing.



I.1.14-rasm. «Схема данных» darchasi.

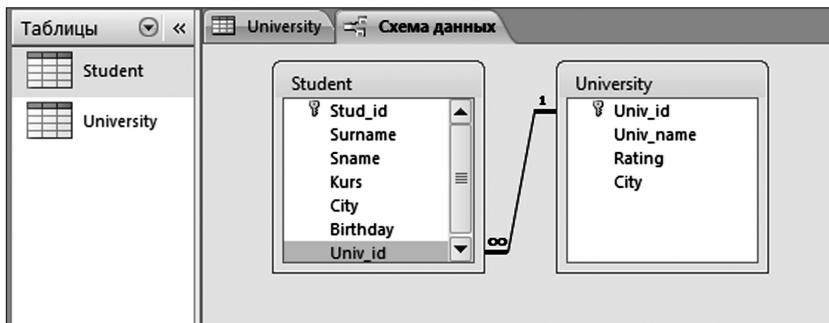
5. Bir jadvaldan (odatda, birlamchi kalit) boshqa jadvalning umumiyl maydoniga (odatda, tashqi kalit) ko'chirib o'tkazing. Bunda «Изменение связей» (Edit relationships) muloqot darchasi ochiladi.



I.1.15-rasm. «Изменение связей» muloqot darchasi.

6. Student jadvalini Univ_id maydonida sichqonchaning chap tugmasini bosgan holda University jadvalini Univ_id maydoniga joylashtiramiz. Bunda «*Изменение связей*» muloqot darchasi ochiladi. Bog‘lanayotgan maydon nomlari to‘gr‘i tanlanganligiga ishonch hosil qilib, darchani «*Обеспечение целостности данных*», «*Каскадное обновление связанных полей*», «*Каскадное удаление связанных полей*» maydonlariga bayroqchalar o‘rnatamiz va «*Создать*» tugmasini bosamiz.

7. Aloqalarni olib tashlash yoki o‘zgartirish uchun kursorni bog‘lanish chizig‘i ustiga joylashtiramiz va unda sichqonchaning tugmasini chertamiz. Tanlangan chiziq qalinlashadi. Tanlangan chiziq olingandan so‘ng, sichqonchani o‘ng knopkasini bosamiz va kontekstli menyudan kerakli menu punktini bajaramiz.



I.1.16-rasm. «Схема данных»da bog‘lanish o‘rnatilishi.

8. Ma’lumotlar sxemasi tuzilmasini saqlaymiz va uni yopamiz.

II. MICROSOFT ACCESS GA TASHQI MA'LUMOT BAZASIDAN JADVALLARNI IMPORT QILISH

Vazifa: O'qitish. assdb tashqi ma'lumot bazasidagi fanlar jadvalini import qiling.

1. «**Внешние данные**» (External date) vkladkasini tanlang, «**Импорт и связи**» guruhida Access tugmasini bosing.



II.2.1-rasm. «Импорт и связи» guruhi.

2. Ma'lumotlar manbaini ko'rsating (O'qitish.assbd ma'lumot bazasi saqlanayotgan papkaga yo'l ko'rsating). «**Переключатель Импорт таблиц, запросов, отчетов, макросов и модулей в текущую базу данных**» kategoriyasiga o'rnatilgan bo'lishi kerak. «Ok» tugmasini bosamiz.

3. Ochilgan darchada import qilish zarur bo'lgan (O'qitish. assbd) jadvalini ajratamiz va «Ok» tugmasini bosamiz.

4. Importlash qadamlarini saqlamagan holda jadval tuzilishi ni saqlaymiz.

MS Office Excel dan jadvallarni import qilish

Vazifa: Ma'lumotlar bazasiga fanlar jadvalini Fanlar.xls fayli dan import qilish.

1. «**Внешние данные**» (External date) vkladkasi «**Импорт и связи**» guruhini tanlaymiz. **Excel** tugmasini bosamiz.

2. Ma'lumotlar manbaini ko'rsatamiz (Fanlar.xls fayl joylashgan papkaga yo'l). «**Переключатель Импортировать данные источника в новую таблицу в текущей базы данных**» ga o'rnatamiz. «Ok» tugmasini bosamiz.

3. «**Листы**» (Show worksheets) obyektini tanlaymiz – «**Далее**» (Next) tugmasini bosamiz.

4. «**Первая строка содержит заголовки столбцов**» maydoniga bayroqcha o'rnatiladi. «**Далее**» (Next) tugmasini bosamiz.

5. Maydonlar tavsifini o'zgarishsiz qoldiring. «**Далее**» (Next) tugmasini bosamiz.

6. «**Переключатель**» guruhida «**Определить ключ**» punktini tanlang va silqib chiquvchi jadval nomlari ro'yxatidan fan kodi maydonini tanlang. «**Далее**» (Next) tugmasi bosiladi.

7. Kalit jadvaliga importni tasdiqlang. «**Готово**» (Finish) tugmasini bosing.

8. «**Конструктор**» rejimida jadval maydon toifalarini tahrirlab chiqing.

9. Jadval tuzilmasini saqlab qo'ying (importlash qadamlarini saqlamagan holda).

III. TAJRIBA ISHI

3. So'rovlар yaratish

So'rovlар ma'lumotlarini ko'rib chiqish, o'zgartirish va tahlil qilish hisoblashlarni bajarish, har xil jadvaldagi ma'lumotlarni birlashtirish yoki jadvaldagi ma'lumotlarni o'zgartirish yoki olib tashlash uchun ishlatish mumkin. So'rovlarni shuningdek, forma yoki hisobotlarga ma'lumotlarni kiritish uchun ham ishlatish mumkin. MS Access 2010 da quyidagi so'rovlarni yaratish mumkin:

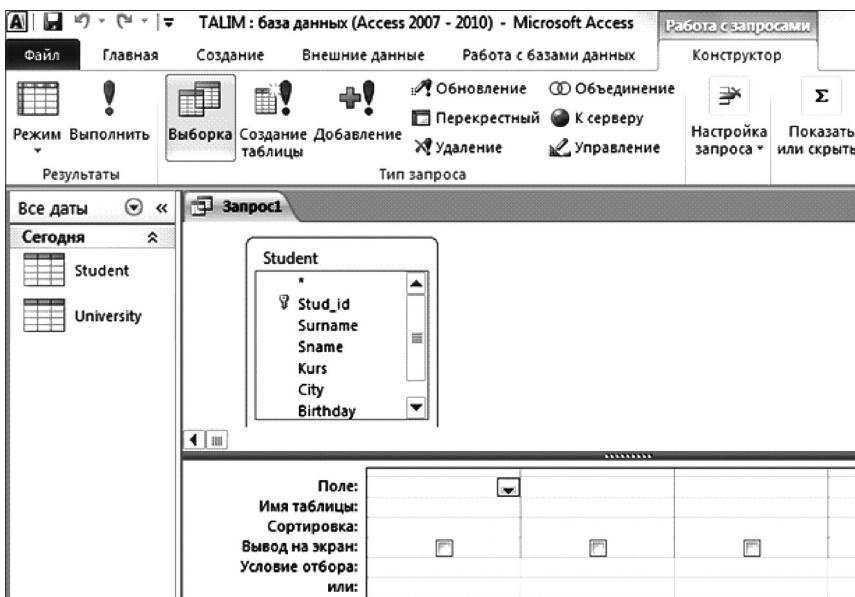
- tanlash so'rovlari;
- parametrli so'rovlar;
- kesishuv so'rovlari;
- o'zgartirish so'rovlari (yozuvlarni olib tashlash, tiklash, takomillashtirish so'rovlari) va boshqalar.

So'rovlarni MS Access 2010 da «**Мастер запросов**» (Query Wizard) yoki «**Конструктор запросов**» (Query Design) yordamida yaratish mumkin. So'rovlarni bitta yoki bir necha jadval asosida yaratish mumkin. Oxirgi holda jadvallar bog'langan bo'lishi kerak.

«**Конструктор запросов**» yordamida so'rovlар yaratish.

Vazifa: STUDENT jadvalidagi ma'lumotlarni familiyalar maydonidagi qiymatni tartiblagan holda chiqarish so'rovini tuzing.

1. TA'LIM ma'lumot baza jadvalini ochamiz.
2. «**Создание**» (Create) vkladkasini «**Запросы**» (Queries) guruhida «**Конструктор запросов**» tugmasini bosamiz. (III.3.1-rasm).



III.3.1-rasm. Spesifikator (So‘rov blankasi) darchasi.

3. «*Таблицы*» vkladkasining «*Добавление таблицы*» (*Show table*) muloqot darchasida Student jadvalining ustida sichqonchani ikki marta bosing.

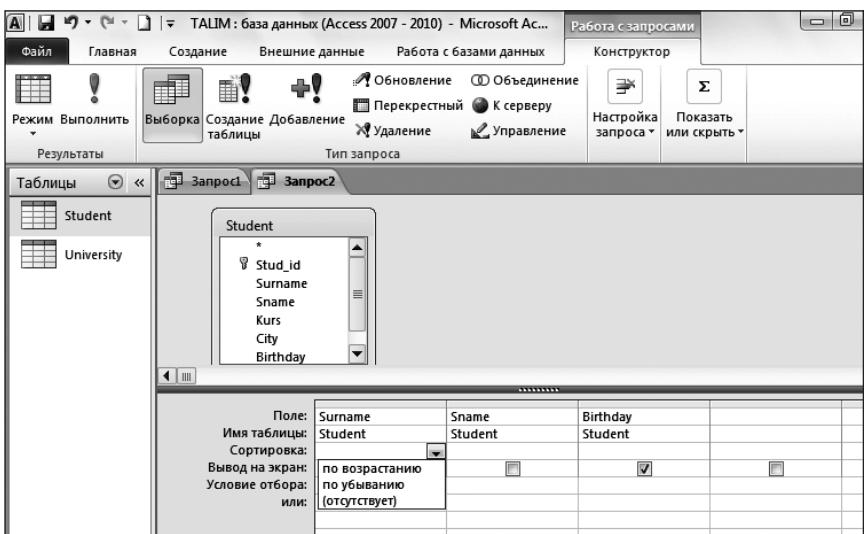
4. «*Добавление таблицы*» (*Show table*) muloqot darchasini yoping.

5. «*Конструктор запросов*» darchasini spetsifikatoriga (so‘rov blankasiga) Student jadvalidagi maydonlarni Surname, Sname, Birthday maydonlarida mos ravishda ketma-ket sichqonchaning chap tugmasini ikki marta cherting (maydonlarni spetsifikator jadvaliga ko‘chirib o’tkazish ham mumkin).

6. «*Сортировка*» (*Sort*) catrida familiya (Surname) maydoni bo‘yicha ma’lumotlarni alfavit tartibida joylashtirish uchun «*По возрастанию*» (*Ascending*)ni tanlab olinadi.

7. So‘rovni ishga tushirish uchun uskunalar panelidagi «*Выполнить*» (*Run*) tugmasini bosamiz.

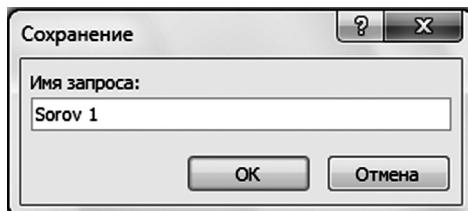
Yaratilgan so‘rovni So‘rov 1 nomi bilan saqlaymiz (so‘rov nomi vkladkasida sichqonchaning o‘ng knopkasini bosing va kontekstli menyuda «*Сохранить*» (*Save*) buyrug‘ini bajaring) (I.2.3-rasm).



III.3.2-rasm. Spetsifikatororda ma'lumotlarni joylashtirish.

| TALIM : база данных (Access 2007 - 2010) - Microsoft Access | | | | | |
|---|---------------------|----------|----------------|------------------------|-------|
| Файл | Главная | Создание | Внешние данные | Работа с базами данных | |
| Режимы | Вставить | Фильтр | Обновить все | Записи | Найти |
| Буфер обмена | Сортировка и фильтр | | | | |
| Запросы | Sorov1 | Surname | Vohid | 07.01.1990 | |
| | | Bahromov | Ilhom | 03.12.1992 | |
| | | Imamov | Botir | 08.12.1991 | |
| | | Komilov | Parda | 06.12.1988 | |
| | | Latipov | Ahmad | 01.12.1991 | |
| | | Pardaev | Asil | 05.11.2012 | |
| | | Polatov | Polat | 01.12.1990 | |
| | | Polatov | Asqar | 05.08.1991 | |
| | | Salimov | Sobir | 07.06.1989 | |
| | | Zokirova | Oqila | 01.05.1991 | |
| | | * | | | |

III.3.3-rasm. So'rov natijasi.



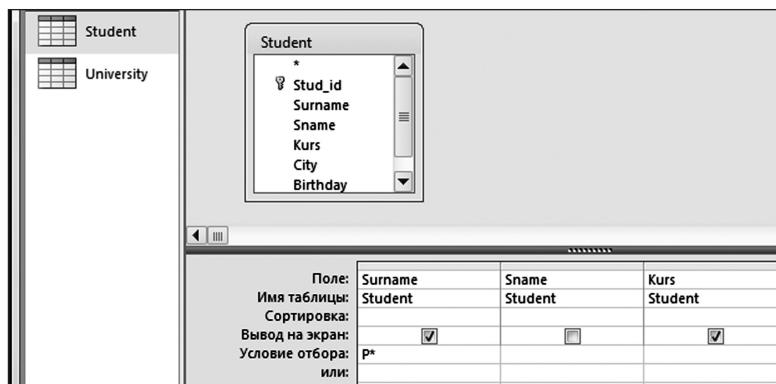
III.3.4-rasm. So‘rovni saqlash.

Andoza (shablon) ishlatib ma'lumotlar bazasidan ma'lumotlar chiqarish.

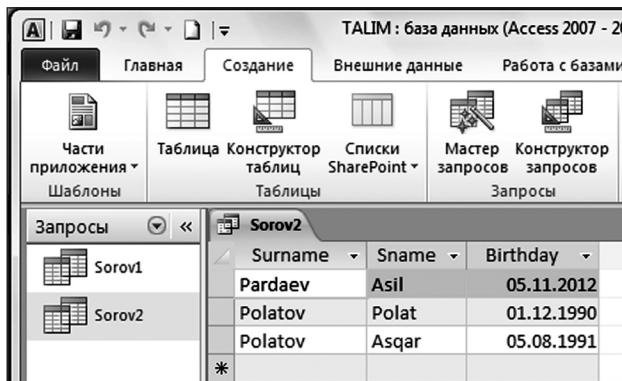
Vazifa: Student jadvalidan Surname, Sname, Kurs maydonlari o‘z ichiga olgan va familiyalari “P” harfi bilan boshlanadigan barcha talabalar ro‘yxatini chiqaring.

1. «**Конструктор**» rejimida Student jadvalini oching (III.3.4-rasm).
2. So‘rov uchun talab etilgan maydonlarni so‘rov blankasiga joylashtiring.
3. Surname ustunini «**Условие отбора**» (**Creteria**) satrida “P*” ifodani joylashtiring (III.3.5-rasm).
4. So‘rovni «**Выполнить**» (**Run**) komandasini bilan bajaring va natijani ko‘ring (III.3.6-rasm).
5. So‘rovni saqlang.

III.3.5-rasm. Jadvalni so‘rov uchun.



III.3.6-rasm. So‘rov blankasiga jadval maydonlarini o‘rnatish.



III.3.7-rasm. So‘rov natijasi.

Konstruktur rejimida bir necha (murakkab) shart bo‘yicha so‘rov yaratish.

Vazifa: University jadvalidan reyting bali 400 dan yuqori bo‘lgan va “T” harfi bilan boshlanadigan shaharda joylashgan universitetlar ro‘yxatini chiqaring. So‘rov quyidagi jadvallarni univ_name, rating, city maydonlarini o‘z ichiga olsin.

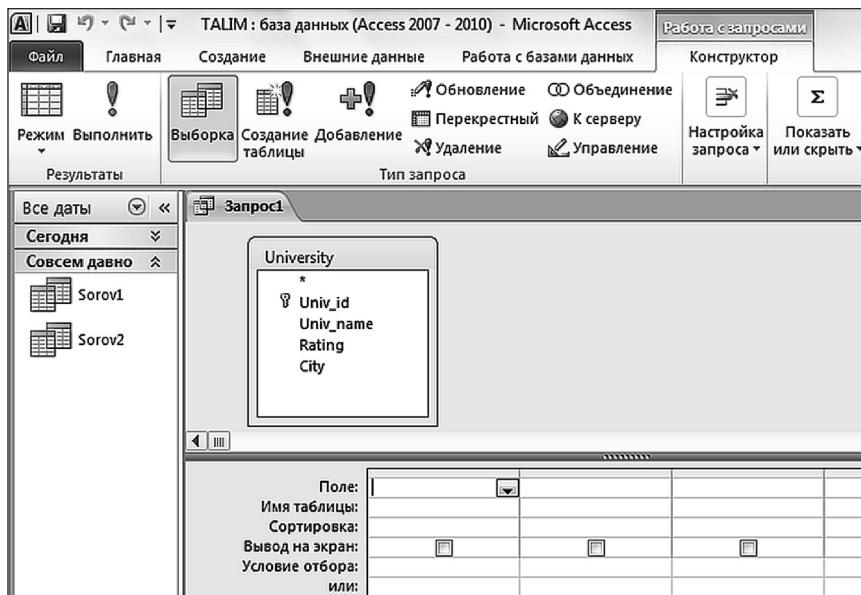
1. «**Конструктор запросов**» buyrug‘ini tanlang va paydo bo‘lgan «**Добавление таблиц**» darchasida universitet jadvalini belgilang va «**Добавить**» knopkasini bosing (III.3.7-rasm).

2. So‘rovda talab etilgan maydonlarni so‘rov blankasiga joylashtiring.

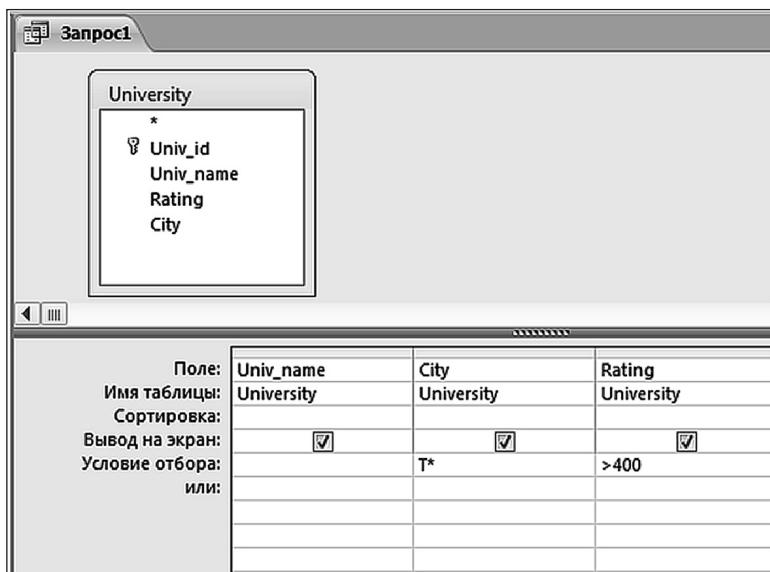
3. So‘rov blankasida city va rating ustunlarini «**Условие отбора**» (**Creteria**) satrida mos ravishda “T*” va “>400” ifodalarni kiriting (III.3.8-rasm).

4. «Выполнить» (Run)  komandası bilan so‘rovni bajaring va natijani ko‘ring (III.3.9-rasm).

5. So‘rovni saqlang va uni yoping.



III.3.9-rasm. «Конструктор запросов» jadvalini o‘rnatish.



III.3.10-rasm. So‘rov blankasiga maydonlarni joylashtirish.

| Univ_name | City | Rating |
|-----------|----------|--------|
| TATU | Toshkent | 520 |
| TDTU | Toshkent | 606 |
| * | | |

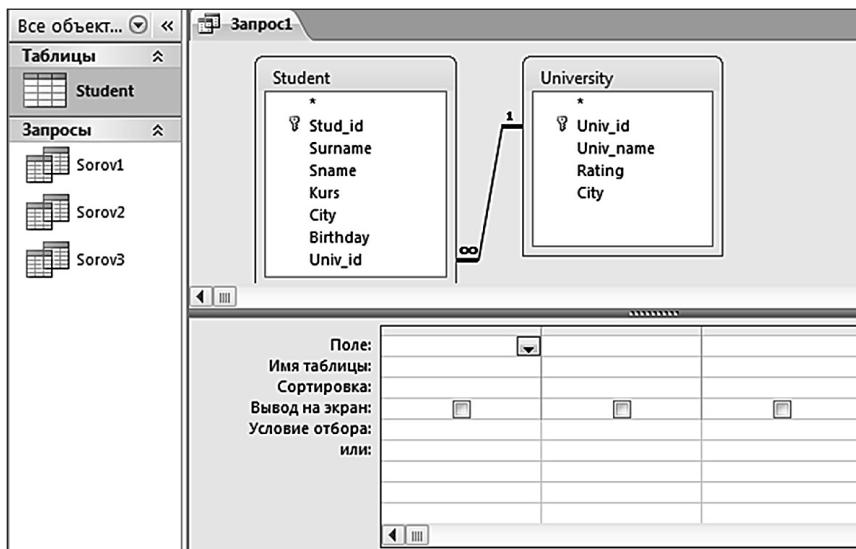
III.3.11-rasm. So‘rov natijasi.

Bir necha jadvallar asosida so‘rovlar yaratish

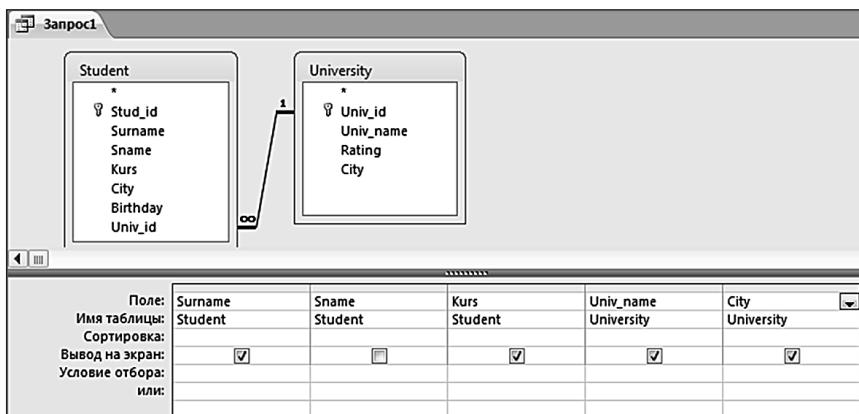
Vazifa: Student va University jadvali asosida talabalar tahlil olayotgan universitet nomlarini chiqarish so‘rovlarni yarating. Yaratilayotgan so‘rov quyidagi maydonlarni ko‘rsatilgan tartibda o‘z ichiga olsin: Surname, SName, Kurs, Univ_Name, University, City.

1. «*Конструктор запросов*»ni ishga tushiramiz va so‘rov yaratishga asos bo‘lgan jadvallarni tanlab olamiz (III.3.10-rasm).

2. Yaratilayotgan so‘rov maydonlarini ko‘rsatilgan tartibda so‘rov blankasiga ko‘chiramiz (maydon nomlarini sichqoncha bilan ikki marta chertish yo‘li bilan) (III.3.11-rasm).



III.3.12-rasm. Jadvallarni «*Конструктор запросов*» darchasiga o‘rnatish.



III.3.13-rasm. So‘rov maydonlarini so‘rov blankasiga o‘rnatish.

3. So‘rovni «*Выполнить*» (Run) komandasini bilan bajaramiz (III.3.12-rasm).

III.3.14-rasm. So‘rov natijasi.

Ko‘p jadvallar asosidagi shartli so‘rovlari.

Vazifa: Student va University jadvalidan foydalanib Toshkent shahridagi universitetlarda tahsil olayotgan talabalar haqidagi ma'lumotlarni chiqaring. Yaratilayotgan so‘rov quyidagi maydonlarni o‘z ichiga olsin: Surname, SName, Kurs, City, Univ_Name, University.city .

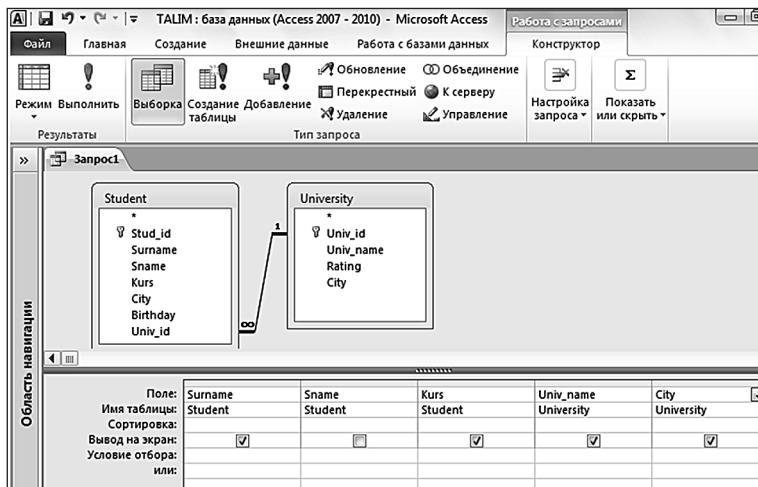
1. «*Конструктор запросов*» buyrug‘ini ishga tushiramiz va so‘rov yaratilayotgan jadvallarni tanlab olamiz.

2. Yaratilayotgan so‘rov maydonlarini so‘rov blankasiga jadvaldagи nomlarini mos ravishda sichqoncha bilan ikki martadan chertib o‘tkazamiz (III.3.13-rasm).

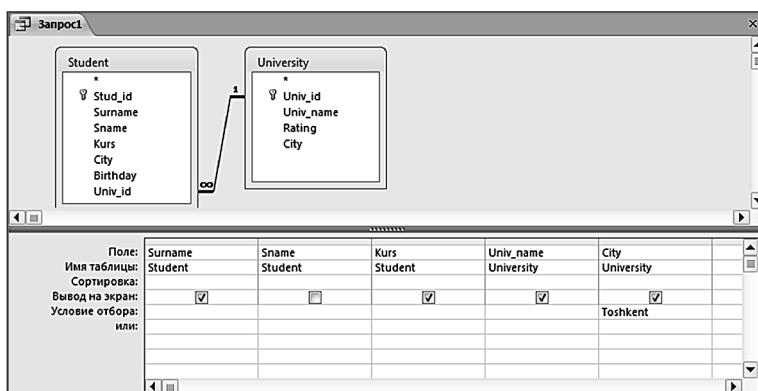
3. University.city ustunini «*Условие отбора*» (Criteria) satrida “Toshkent” so‘zini kiritamiz (III.3.14-rasm).

4. «*Выполнить*» (Run) komandasini yordamida so‘rovni bajaramiz. So‘rov natijasi (III.3.15-rasm) rasmida keltirilgan.

5. So‘rovni saqlaymiz (so‘rov vkladkasi ustida sichqonchani o‘ng tugmasi bosilib kontekstli menyudan «*Сохранить*» punkti tanlab olinadi. Paydo bo‘lgan darchada so‘rovga nom berib «OK» tugmasini bosamiz) va uni yopamiz (III.3.15-rasm).



III.3.15-rasm. So‘rov maydonlarini o‘rnatish.



III.3.16-rasm. Shartni o‘natish.

The screenshot shows the Oracle Database SQL Developer interface. On the left, there is a sidebar titled 'Запросы' (Queries) with five entries: 'Sorov1', 'Sorov2', 'Sorov3', 'Sorov4', and 'Sorov5'. To the right of the sidebar is a main window titled 'Sorov5' which displays a table with student data. The table has columns: Surname, Sname, Kurs, Student., Univ_name, and University.C. The data is as follows:

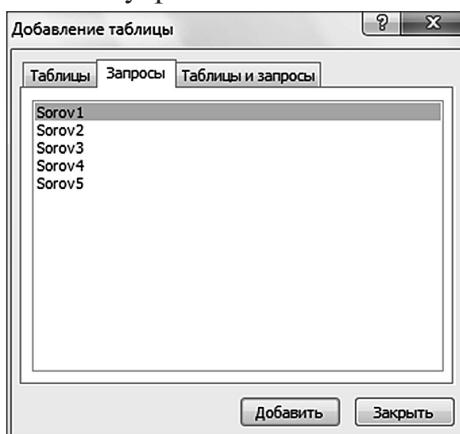
| Surname | Sname | Kurs | Student. | Univ_name | University.C |
|---------|-------|------|----------|-----------|--------------|
| Imamov | Ilhom | 1 | Angren | TATU | Toshkent |
| Latipov | Ahmad | 3 | Andijan | TATU | Toshkent |
| Salimov | Sobir | 4 | Toshkent | TDTU | Toshkent |
| Polatov | Asqar | 4 | NULL | TDTU | Toshkent |
| * | | | | | |
| | | | | | |
| | | | | | |

Rasm II.2.17. So‘rov natijasi.

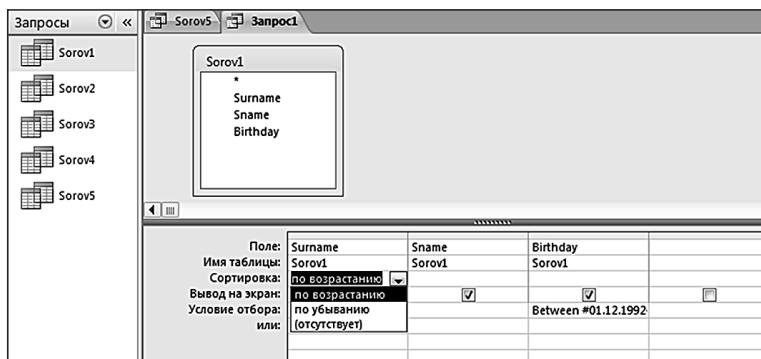
So‘rov asosida so‘rov yaratish

Vazifa: Sorov 1 (Student jadvalidan talabalar haqidagi axborotlar alfavit tartibida chiqarish so‘rovi) asosida tug‘ilgan sanasi 01.12.1991 dan 03.12.1992 gacha bo‘gan talabalarni ro‘xatini alfavit tartibida chiqaring.

1. «Конструктор запросов»ни ishga tushiramiz. «Добавление таблицы» darchasida «Запросы» vkladkasini ajratamiz. Unda Sorov 1 ni belgilaymiz va «Добавить» tugmasini bosamiz (III.3.16-rasm).
2. Yaratilayotgan so‘rov maydonlarini Surname, Sname, Birthday so‘rov blankasiga ko‘chiramiz (III.3.17-rasm).
3. Birthday maydoni ustunini «Условие отбора» satrida between #01.12.1992# and #03.12.1992# ifodani teramiz.
4. Surname ustunini «Сортировка» satrida ro‘yxatdan «по возрастанию» tanlab olinadi.
5. «Выполнить» buyrug‘ini bajaramiz.
6. So‘rov natijasi jadvalini zarurat bo‘lsa saqlab qo‘yamiz va uni tugmasi bilan yopamiz.



III.3.18-rasm. Sorov 1ni tanlash.



III.3.19-rasm. Shartlarni o‘rnatish.

| Surname | Sname | Kurs | City | Birthday |
|---------|-------|------|---------|------------|
| Imamov | Ilhom | 1 | Angren | 03.12.1992 |
| Kenjaev | Botir | 2 | Buxoro | 08.12.1991 |
| Latipov | Ahmad | 3 | Andijan | 01.12.1991 |
| * | | | | |

III.3.20-rasm. So‘rov natijasi.

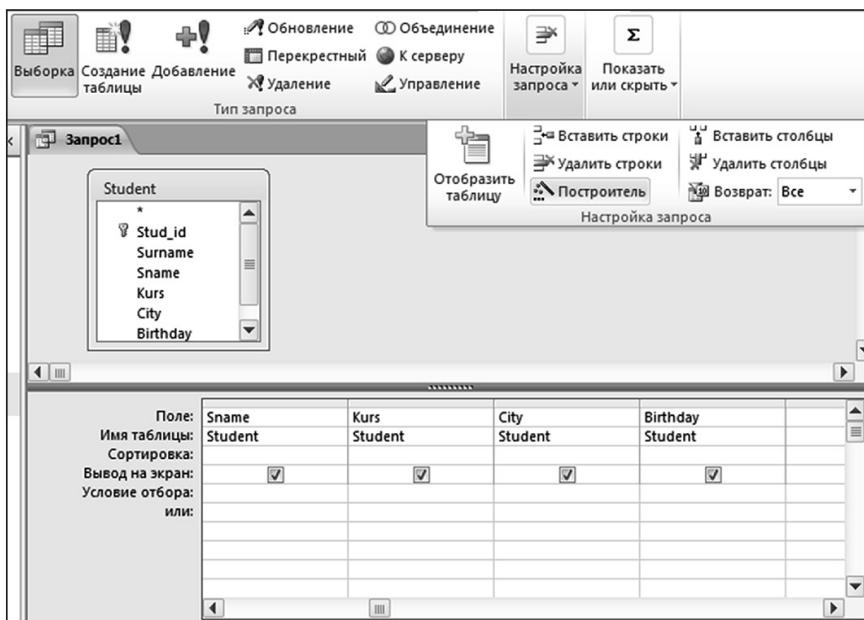
So‘rov shartlarini «Построитель выражений» darchasida o‘rnatish

Vazifa: Student jadvalidan tug‘ilgan sanasi 01.12.1992 dan 03.12.1992 gacha diapazonda (intervalda) bo‘lgan talabalar ro‘yxatini chiqaring. Ma’lumotlar alfavit tartibida chiqarilsin.

1. «Конструктор запросов» knopkasini bosamiz. «Добавление таблицы» darchasida «Таблицы» vkladkasini ajratamiz va Student jadvalini belgilaymiz. So‘ngra «Добавить» tugmasini bosamiz (III.3.18-rasm).

2. Kursorni Birthday ustunining «Условие отбора» maydoniga joylashtiring va «Построитель выражений» darchasini quyidagi usullarni biri bilan ishga tushiramiz:

- Ctrl+F2 tugmalarini bosib
- Uskunalar panelining «Настройка запросов» (Query Setup) guruhidagi «Построителя» tugmasini bosib
- Kursor turgan katakda sichqonchani o‘ng knopkasini bosib, kontekstli menyudan построит... punktini bosib.

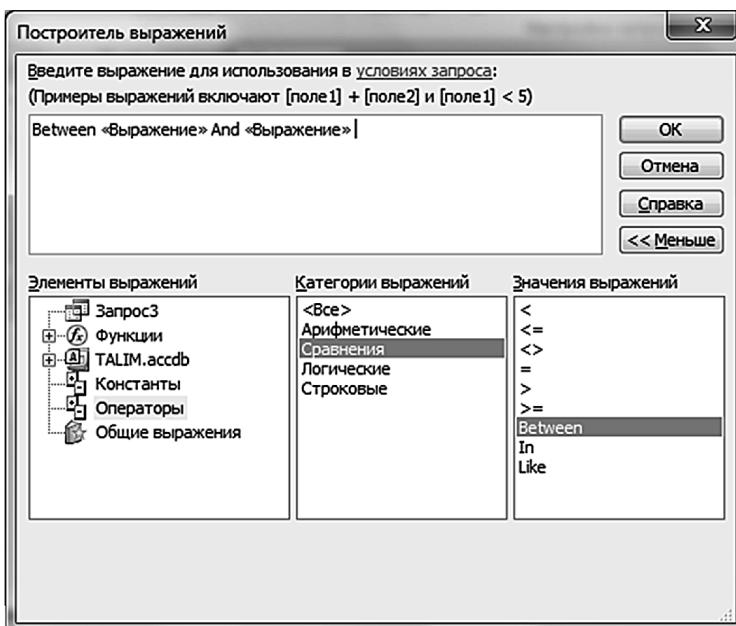


III.3.21-rasm. So‘rov maydonlarini o‘rnatish.

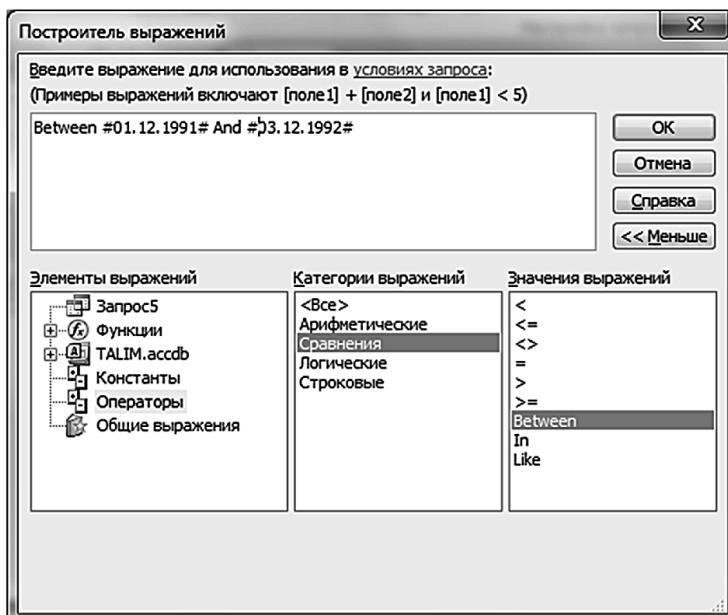
3. Ko‘rsatilgan usullardan biri bilan «*Построитель выражений*» darchasini ochamiz (III.3.19-rasm).
4. «*Построитель выражений*» darchasida «*Оператор*» (Operators)

«*Сравнения*» papkasini ochib between operatorini tanlang va uni ikki marta cherting (III.3.19-rasm).

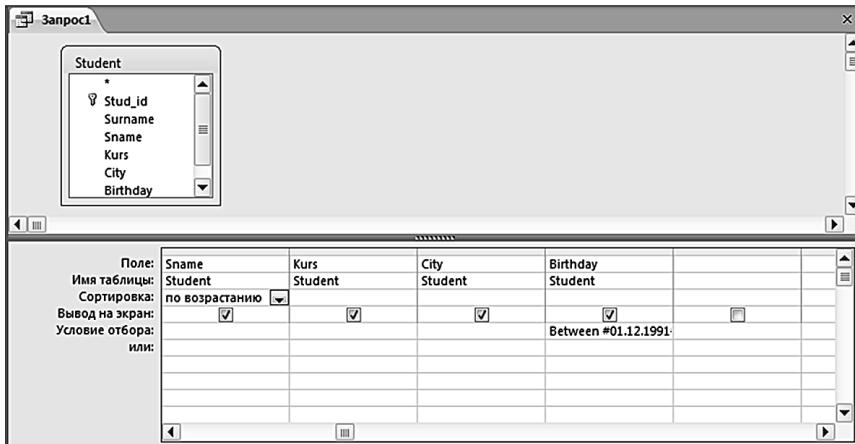
5. «*Построитель выражений*» darchasini ifodalar yozish maydonida between «*выражение*» and «*выражение*» ifodani tahrirlang. Buning uchun birinchi выражение so‘ziga kursorni joylab #01.12.1991# ifodani so‘ngra ikkinchi «*выражение*» so‘ziga kursorni joylab #03.12.1992# ifodani teramiz va «OK» tugmasini bosamiz.
6. Surname ustunini «*Сортировка*» satrida «*По возрастанию*» tartiblash rejimini ajratamiz (III.3.21-rasm).
7. So‘rovni «*Выполнить*» (Run) komandasini bajaramiz va natijani ko‘ramiz.
8. Yaratilgan so‘rovga nom qo‘yib saqlaymiz. So‘rov natijasi (III.3.22-rasm)da keltirilgan.



III.3.22-rasm. «Построитель выражений» darchasi.



III.3.23-rasm. «Построитель выражений» darchasida ifodaning ko‘rinishi.



III.3.24-rasm. Barcha shartlar o'rnatilgan darcha.

| So'rov6 | | | | |
|---------|-------|------|---------|------------|
| Surname | Sname | Kurs | City | Birthday |
| Imamov | Ilhom | 1 | Angren | 03.12.1992 |
| Kenjaev | Botir | 2 | Buxoro | 08.12.1991 |
| Latipov | Ahmad | 3 | Andijan | 01.12.1991 |
| * | | | | |

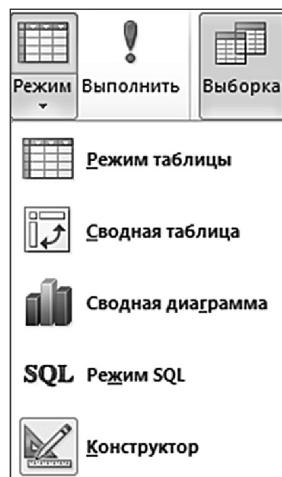
III.3.25-rasm. So'rov natijasi.

SQL rejimida so'rovlar yaratish

Ma'lumotlar bazasidan ma'lumotlarni chiqarish uchun SQL tili ishlataladi. Har bir bajarilayotgan so'rov aslida SQL ga asoslangan. MS Access da Jet SQL tili ishlataladi.

1. «Конструктор запросов» yordamida oddiy so'rov yaratamiz. Masalan Student jadvalidagi barcha ma'lumotlarni ko'rish so'rovini va uni bajaramiz.

2. Ochilgan darchada SQL rejimi ga o'tamiz (III.3.23-rasm), yaratilgan so'rovni SQL tilidagi ifodasi (III.3.24-rasm)da keltirilgan.



III.3.26-rasm.
SQL rejimini tanlash.

```
3anpoc1
SELECT Student.Surname, Student.Sname, Student.City, Student.Birthday
FROM Student
WHERE ((Student.Birthday) Between #12/1/1991# And #12/3/1992#))
ORDER BY Student.Surname;
```

III.3.27-rasm. Konstruktor rejimida yaratilgan so‘rovning SQL tilidagi ifodasi.

Masalan, Universitet haqida barcha ma’lumotlarni chiqarish uchun select operatori quyidagicha yoziladi: Select * from University; (III.3.24-rasm).

```
3anpoc1
SELECT * from University
```

III.3.28-rasm. Select komandasi.

| Univ_id | Univ_name | Rating | City |
|---------|-----------|--------|-----------|
| 10 | TATU | 520 | Toshkent |
| 11 | NDU | 345 | Namangan |
| 14 | NATU | 326 | Nukus |
| 15 | FDU | 368 | Farg’ona |
| 18 | AATU | 327 | Andijon |
| 22 | TDTU | 606 | Toshkent |
| 32 | SDU | 416 | Samarqand |
| * | | | |

III.3.29-rasm. So‘rov natijasi.

Masalan, Student jadvalidan faqat 3-kurs talabalar ro‘xatini chiqarish quyidagi komanda bilan amalgalashiriladi: (III.3.27–III.3.28-rasmlar).

Select Surname, Sname, Kurs from Student where Kurs=3;

```
3anpoc1
SELECT Surname, Sname,Kurs from Student where Kurs=3;
```

III.3.30-rasm. So‘rov ko‘rinishi.

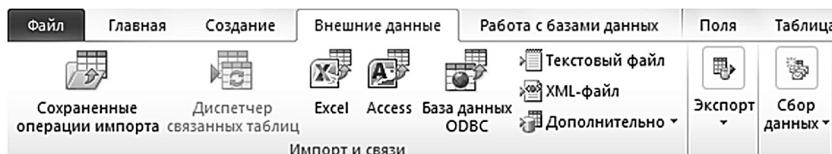
| Запрос1 | | |
|---------|-------|------|
| Surname | Sname | Kurs |
| Polatov | Polat | 3 |
| Pardaev | Asil | 3 |
| Latipov | Ahmad | 3 |
| * | | |
| | | |
| | | |

III.3.31-rasm. So'rov natijasi.

Microsoft Access ga tashqi ma'lumot bazasidan jadvallarni import qilish.

Vazifa: O'qitish. assdb tashqi ma'lumot bazasidagi fanlar jadvalini import qiling.

1. «**Внешние данные**» (External date) vkladkasini tanlang, «**Импорт и связи**» guruhida Access tugmasini bosing.



III.3.32-rasm. «*Импорт и связи*» guruhi.

2. Ma'lumotlar manbaini ko'rsating (O'qitish. assbd ma'lumot basasi saqlanayotgan papkaga yo'l ko'rsating). «**Переключатель Импорт таблиц, запросов, отчетов, макросов и модулей в текущую базу данных**» kategoriyasiga o'rnatilgan bo'lishi kerak. «Ok» tugmasini bosamiz.

3. Ochilgan darchada import qilish zarur bo'lgan (O'qitish. assbd) jadvalini ajratamiz va «Ok» tugmasini bosamiz.

4. Jadval tuzilishini saqlaymiz, importlash qadamlarini saqlamagan holda.

MS Office Excel dan jadvallarni import qilish

Vazifa: Ma'lumotlar bazasiga fanlar jadvalini Fanlar.xls fayli dan import qilish.

1. «**Внешние данные**» (External date) vkladkasi «**Импорт и связи**» guruhini tanlaymiz. Excel tugmasini bosamiz.

2. Ma'lumotlar manbaini ko'rsatamiz (Fanlar.xls fayl joylashgan papkaga yo'l). «**Переключатель Импортировать данные**

*источника в новую таблицу в текущей базы данных»*га о‘rnatamiz. «Ok» tugmasini bosamiz.

3. «Листы» (Show worksheets) obyektini tanlaymiz – «Далее» (Next) tugmasini bosamiz.

4. «Первая строка содержит заголовки столбцов» maydoniga bayroqcha o‘rnatiladi. «Далее» (Next) tugmasini bosamiz.

5. Maydonlar tavsifini o‘zgarishsiz qoldiring. «Далее» (Next) tugmasini bosamiz.

6. «Переключатель» guruhida «Определить ключ» punktini tanlang va silqib chiquvchi jadval nomlari ro‘yxatidan fan kodi maydonini tanlang. «Далее» (Next) tugmasini bosiladi.

7. Kalit jadvaliga importni tasdiqlang. «Готово» (Finish) tugmasini bosing.

8. «Конструктор» rejimida jadval maydon toifalarini tahrirlab chiqing.

9. Jadval tuzilmasini saqlab qo‘ying (importlash qadamlarini saqlamagan holda).

ILOVA B

MUSTAQIL BAJARISH UCHUN TOPSHIRIQLAR

Quyida keltirilgan predmet sohalar uchun relatsion ma’lumotlar bazasini yaratating. Yaratilgan ma’lumotlar bazasining Microsoft Access da fizik modelini quring.

1. Kutubxona predmet sohasi

Jadvallar

Kitoblar (Kitob kodi, nomi, muallif, narxi, janri)

O‘quvchi (O‘quvchi kodi, familiya, ism, sharifi, manzili, telefon)

Berilgan kitoblar (Kitob kodi, o‘quvchi kodi, berilgan sana, qaytish sanasi).

2. Mehmonxona predmet sohasi

Jadvallar

Mijoz (Mijoz kodi, familiya, ismi, pasport ma’lumotlari)

Xonalar (Xona kodi, xona, odamlar soni, narxi)

Joylashish (Joylashish kodi, mijoz kodi, xona kodi, joylashish sanasi, qaytish sanasi).

3. Tovarlar sotish predmet sohasi

Jadvallar

Tovarlar (Tovar kodi, nomi, ulgurji narxi, chakana narxi, tavisifi)

Xaridorlar (Xaridor kodi, telefon, manzil)

Bitim (Bitim kodi, bitim sanasi, tovar kodi, soni, xaridor kodi, ulgurji sotish belgisi).

4. Buyurtmalarni bajarish predmet sohasi

Jadvallar

Buyurtmachilar (Buyurtma kodi, nomi, manzili, telefon)

Tovarlar (Tovar kodi, narxi, yetkazib berish)

Buyurtma (Buyurtma kodi, buyurtmachi kodi, tovar kodi, tovar miqdori, sana).

5. Ehtiyyot qismlari sotish firma predmet sohasi

Jadvallar

Ta'minlovchilar (Ta'minlovchi kodi, nomi, manzil, telefon)

Detallar (Detal kodi, nomi, artikul, narxi)

Ta'minlash (Ta'minlovchi kodi, detal kodi, soni, sana).

6. Malaka oshirish kurslari predmet sohasi

Jadvallar

Guruuhlar (Guruuh nomeri, mutaxassislik, bo'lim, tinglovchilar soni)

O'qituvchi (O'qituvchi kodi, familiya, ism, sharifi, telefon, staj)

Yuklama (O'qituvchi kodi, guruuh nomeri, soatlar soni, fan, mashg'ulot turi, to'lov).

7. O'quv yuklamasini taqsimlash predmet sohasi

Jadvallar

O'qituvchi (O'qituvchi kodi, familiya, ism, sharifi, ilmiy darama, mansab, staj)

Fanlar (Fan kodi, nomi, soatlar soni)

Yuklama (O‘qituvchi kodi, fan kodi, guruh nomeri).

8. Yuk tashish predmet sohasi

Jadvallar

Marshrutlar (Marshrut kodi, nomi, uzoqligi, yo‘ldagi kunlar soni, to‘lov)

Haydovchilar (Haydovchi kodi, ism, staj)

Bajarilgan ish (Marshrut kodi, haydovchi kodi, jo‘nash sanasi, qaytish sanasi).

9. Turistik firma predmet sohasi

Jadvallar

Marshrutlar (Marshrut kodi, davlat, ob-havo, muddati, mehmonxona, narxi)

Yo‘llanma (Marshrut kodi, mijoz kodi, jo‘nash sanasi, soni, chegirma)

Mijoz (Mijoz kodi, familiya, ism, sharifi, manzili, telefon).

10. Telefonda so‘zlashuvlarni hisobga olish predmet sohasi

Jadvallar

Abonentlar (Abonent kodi, telefon raqami, INN, manzil)

Shaharlar (Shahar kodi, nomi, kunduzgi ta’rif, tungi ta’rif)

So‘zlashuvlar (So‘zlashuv kodi, abonent kodi, shahar kodi, sana, minutlar soni, sutka vaqt).

Topshiriqni bajarishga qo‘yiladigan talablar:

1. Berilgan predmet soha jadvallari bo‘yicha relatsion ma’lumotlar bazasini tuzing. Jadvallar orasida aloqa o‘rnating va ma’lumotlar bilan to‘ldiring.

2. Yaratilayotgan jadvallar, so‘rovlар va shartlar, topshiriqlar predmet sohaga to‘la mos kelishi kerak.

3. Har bir jadval 8–10 ta yozuvdan tashkil topishi kerak.

4. Jadvalga kiritilayotgan ma’lumotlar ma’noli bo‘lishi kerak.

5. Maydonning ma’lumot toifalari ularda saqlanadigan ma’lumotlarga mos kelishi kerak.

FOYDALANILGAN ADABIYOTLAR RO‘YXATI

1. Коноли Т., Брегк К. База данных, проектирование, реализация и сопровождения, теория и практика, Университет Пейсли, Шотландия, изд. — М.: СПБ. — Киев, 2003.
2. Диго С.М. База данных. Проектирование и создания. — М.: Изд.ЕАСИ, 2008.
3. Лори Улрих Филлер и др. ACCESS 2010.— М.: Диалектика, 2010.
4. Майкл Грокс и др. Microsoft Office ACCESS 2007. Библия пользователя. М.: Диалектика, 2008.
5. Григорьев Ю.А., Ревунков Г.И. Банки данных. — М.: Изд. МГТУ им. Баумана, 2002.
6. Метью Мак-Дональд. ACCESS 2007. Недостающее руководство. — СПБ.: БХВ— Петербург. 2007.
7. Роб Р. Система баз данных: проектирование, реализация и управления — издательство «БХВ — Санкт-Петербург», 2003.
8. Днепров А. Microsoft ACCESS 2007. — СПБ.: Питер, 2008.
9. Хомоненко М. Д. База данных. Под ред. проф. А.Д. Хомоненко. — 4-й изд., доп. и перераб. — СПБ.: Короно принт, 2004.
10. Советов В.Я. База данных. Теория и практика — М.: Высш. шк., 2005.
11. Астахова И.Ф., Толстобров А.П. SQL в примерах и задачах. — Учебное пособие. Новое знание, стр. 176, 2002.
12. Бен Форма. Освой самостостоятельно SQL. 10 минут на урок (3-ое издание) издательство «Vilyams», 2005.

MUNDARIJA

| | |
|---|-----------|
| I BOB. MA'LUMOTLAR BAZASINI LOYIHALASH ASOSLARI | 4 |
| 1.1. Ma'lumotlar bazasi asosidagi avtomatlashtirilgan axborot tizimlari | 4 |
| 1.2. Ma'lumotlar bazasining modellari | 5 |
| 1.3. Relatsion ma'lumotlar bazasi | 8 |
| 1.4. Relatsion ma'lumotlar bazasi jadvalini normallashtirish | 11 |
| 1.5. Jadvallar orasidagi bog'lanishlarni loyihalash | 16 |
| 1.6. Relatsion ma'lumotlar bazasining axborot modellari | 18 |
| II BOB. MA'LUMOTLAR BAZASINI BOSHQARISH TIZIMLARINI ISHLAB CHIQISH DASTURIY MAHSULOTLAR TAHLILI. | 21 |
| 2.1. Ma'lumotlar bazasini ishlab chiqish dasturiy vositalarinining rivojlanish tarixi | 21 |
| 2.2. SQL tili, MS SQL SERVER, ORACLE haqida umumiy ma'lumotlar | 22 |
| 2.3. Microsoft Access MBBT | 24 |
| 2.4. Microsoft ACCESS vizual muhitini bilan tanishish | 26 |
| III BOB. MICROSOFT ACCESS VOSITALARI ASOSIDA YORDAMCHI MA'LUMOTLAR BAZASINI ISHLAB CHIQISH TEKNOLOGIYALARI | 30 |
| 3.1. Ma'lumotlar bazasi jadvallarini ishlab chiqish texnologiyasi | 30 |
| 3.2. So'rovlar yaratish texnologiyasi | 41 |
| 3.3. So'rovlar yordamida hisoblashlarni avtomatlashtirish | 48 |
| IV BOB. MA'LUMOTLAR BILAN ISHLASHNI AVTOVOMATLASHTIRISH. | 50 |
| 4.1. Formalar yordamida ma'lumotlarni kiritish va tahlil qilish | 50 |
| 4.2. Ma'lumotlarni jadvalga kiritish uchun forma ishlab chiqish texnologiyasi | 50 |
| 4.3. So'rova ma'lumotlar kiritish uchun forma ishlab chiqish texnologiyasi | 56 |
| 4.4. Makroslar yordamida ma'lumotlar bazasi obyektlarini boshqarish | 60 |
| 4.5. Foydalanuvchi menyusini ishlab chiqish | 67 |
| V BOB. VISUAL BASIC FOR APPLICATIONS MUHITIDA BOSHQA-RUVCHI DASTURLAR ISHLAB CHIQISH. | 70 |
| 5.1 Visual Basic for Applications ning umumiy xarakteristikasi | 70 |
| 5.2. Protsedura va funksiyalar | 70 |
| 5.3. O'zgaruvchilar, konstantalar va ma'lumot toifalari | 73 |
| 5.4. O'zgaruvchilarning ta'sir sohasi va protseduralar | 79 |
| 5.5. VBA ni boshqarish konstruksiylari | 82 |

| | |
|--|------------|
| 5.6. Sikl va protseduralardan chiqish | 87 |
| 5.7. Modular | 88 |
| VI BOB. SQL TILIDA SO‘ROVLAR YARATISH | 90 |
| 6.1. SQL standartlari va tarkibi | 90 |
| 6.2. SQLda ma'lumot toifalari | 92 |
| 6.3. Oddiy SELECT so‘rovlari | 97 |
| 6.4. So‘rovarda maxsus operatorlarni ishlatalish | 98 |
| 6.5. Chiqarilayotgan ma'lumotlarni o‘zgartirish | 100 |
| 6.7. Bog‘langan qism so‘rovlар yaratish | 104 |
| 6.8. Birlashtirish operatori | 106 |
| 6.9. Ma'lumotlar bilan manipulatsiyalash | 107 |
| 6.10. Ma'lumotlar bazasi obyektlarini yaratish | 110 |
| 6.11. Cheklanishlar | 113 |
| VII BOB. MA’LUMOTLARNI TAQSIMLANGAN QAYTA ISHLASH | 116 |
| 7.1. Umumiy tushunchalar | 116 |
| 7.2. Ma'lumotlarni taqsimlangan qayta ishlashda mijoz-server modeli | 117 |
| 7.3. Ma'lumot bazasining server modeli | 119 |
| VIII BOB. WEB TEKNOLOGIYALARI VA MA’LUMOTLAR BAZASI | 122 |
| 8.1. PHP tili haqida | 122 |
| 8.2. O‘zgaruvchilar va ma'lumot toifalari | 126 |
| 8.3. Ifodalar, operatorlar va boshqaruvchi konstruksiyalar | 138 |
| 8.4. MySQL PHP tilida ma'lumotlar bazasi bilan ishslash vositasи | 157 |
| 8.5. MySQL bilan ishslash uchun PHPning standart funksiyalari | 158 |
| 8.6. Ma'lumotlar bazasining ochiq arxitekturasi (ODBC – Open Data Base Connectivity) | 170 |
| 8.7. Microsoft ACCESS va PHP | 175 |
| IX BOB. MA’LUMOTLAR BAZASINI ISHLATISH | 178 |
| 9.1. Parallel qayta ishlashni boshqarish | 178 |
| 9.2. Ishonchli loyihalash va administratsiyalash | 180 |
| 9.3. MB administratorlari va foydalanuvchilari, ularning maqsadi, huquqi va majburiyatları | 183 |
| 9.4. Axborot tizimlarini administasiyalash | 185 |
| ILOVA | 190 |
| FOYDALANILGAN ADABIYOTLAR RO‘YXATI | 221 |

**GAIPNAZAROV TAHIRITDIN SIDIKOVICH
IRGASHEVA DURDONA YAKUBDJANOVNA**

MA'LUMOTLAR OMBORI

Muharrir: M. Tursunova
Musahhihlar: M. Turdiyeva
Sahifalovchi: E. Muratov

O'zbekiston faylasuflari milliy jamiyati nashriyoti
100000, Toshkent, Matbuotchilar ko'chasi, 32.
Tel./faks: 239-88-61

Nashriyot litsenziysi: AI №216, 03.08.2012.
Bosishga ruxsat etildi 18.09.2017. Bichimi 60 x 90 1/16. Ofset qog'ozsi.
Shartli bosma tabog'i 14,5. Nashr hisob tabog'i 14,0. Adadi 1039 nusxa.
Buyurtma №476.

«NISO POLIGRAF» MChJ bosmaxonasida chop etildi.
Toshkent viloyati, O'rta Chirchiq tumani, «Oq-Ota» QFY,
Mash'al mahallasi, Markaziy ko'chasi, 1-uy.