

**O'ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI VA
KOMMUNIKATSIYALARINI RIVOJLANTIRISH VAZIRLIGI**

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

TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI

**“MA'LUMOTLAR BAZASINI
BOSHQARISH TIZIMI”**

FANI BO'YICHA

MA'RUZA MATNLARI

Tuzuvchi:

J.T.Usmonov

TOSHKENT - 2016

Mazkur ma'ruza matnlari Oliy va o'rta maxsus ta'lim vazirligining 2016 yil 6 aprelidagi 137-sonli buyrug'i bilan tasdiqlangan o'quv reja va dastur asosida tayyorlandi.

Tuzuvchi: TATU assistenti J.T.Usmonov

MA'RUZA MASHG'ULOTLAR MAZMUNI

№	Ma`ruzalar	Ajratilgan soat
1.	Fanning maqsadi va vazifalari. Asosiy tushuncha va ta'riflar. Ma'lumotlar bazasiga qo'yiladigan talablar.	2
2.	Ma'lumot bazasi tizimining arxitekturasi. Uch bosqichli arxitektura. Tashqi bosqich. Konseptual bosqich. Ichki bosqich.	2
3.	Ma'lumot baza modeli. Moxiyat-aloha modeli.	2
4.	Relyatsion ma'lumot modeli. Ma'lumotlar bazasida munosabatlar.	2
5.	Relyatsion algebra va relyatsion xisobot elementlari	2
6.	Ma'lumotlar bazasini rejalashtirish, loyihalashlash va administratorlash.	2
7.	Ma'lumotlar bazasini normallashtirish. Normal formalar 1NF, 2NF, 3NF va Kodd.	2
8.	SQLtili. SQL operatorlarini yozish.	2
9.	Ma'lumotlar manipulyatsiya qilish. Oddiy so'rovlar yaratish.	2
10.	SQLtili. Ma'lumotlar tavsiflash.	2
11.	Tranzaksiyalarni boshqarish. So'rovlar yaratish va qayta ishlash.	2
12.	Ma'lumotlar bazasini administratorlash va xavfsizligini ta'minlash	2
13.	Ochiq ma'lumotlar bazasi aloqasi (ODBC) interfeysi	2
14.	Ma'lumotlar bazasiga murojaatni tashkil etishda ODBC va S++ dan foydalanish	2
15.	S++ va SQL orqali yangi ma'lumotlar qo'shish, o'zgartirish va o'chirish.	2
16.	ADO texnologiyasidan foydalanish. Ma'lumotlar bazasiga murojaatni tashkil etishda ADO va S++ dan foydalanish	2
17.	ADO va C++ orqali maydon qiymatlarini kiritish, yozuvlarni o'zgartirish, qo'shish va o'zgartirish.	2
18.	XML va ma'lumotlar bazasi	2
	Jami:	36

1–mavzu. Fanning maqsadi va vazifalari. Asosiy tushuncha va ta’riflar. Ma’lumotlar bazasiga qo‘yiladigan talablar

Reja

1. Axborot tizimlari tushunchasi.
2. Ma’lumotlar bazasi.
3. Asosiy tushuncha va ta’riflar. Ma’lumotlar bazasiga qo‘yiladigan asosiy talablar.

Tayanch iboralar: *Fayl, yozuv, ma’lumot bazasi, ma’lumot banki, avtomatlashgan axborot tizimi, ob’ekt, predmet soha.*

Ko‘pgina masalalarni yechish asosida axborotlar yoki axborotlarni qayta ishlash yotadi. Axborotlarni qayta ishlashni yengillashtirish maksadida axborot tizimlari yaratiladi. Avtomatlashgan axborot tizimlari (AAT) (Axborot tizimlar AT) deb, shunday tizimlarga aytamizki, ularning tarkibida texnik vositalar, jumladan shaxsiy kompyuterlar ishtrok etadi. ATlarni keng ma’noda axborotni qayta ishlaydigan ixtiyoriy tizimni tushunish mumkin. Tadbiq etish sohasiga qarab, ATlar ishlab chiqarish sohasida, talim sohasida, sog‘liqni saqlash sohasida, harbiy sohada va boshqa sohalarda ishlatiladigan tizimlarga ajratish mumkin.

ATlarni maqsadli ishlatilishiga qarab bir qancha kategoriyalarga ajratishimiz mumkin. Jumladan boshqariluvchi, axborot qidiruv, axborot ma’lumotnoma va boshqa tizimlarga ajratiladi.

ATlarning tor ma’noda ba’zi bir amaliy masalalarni yechishga ishlatiladigan apparat dasturiy vositalari majmuasini ham tushunamiz. Masalan, tashkilotlarda ishlatiladigan kadrlarni hisobga olish va kuzatish, omborxona yoki skladlardagi material va tovarlarni hisobga olish va nazorat qilish, buxgalter masalarini yechish va boshqalar.

ATlarni asosida ma’lumotlar bazasi yotadi. MB deganda, ma’lumotlarni shunday o‘zaro bog‘langan to‘plamini tushunamizki, u mashina xotirasida saqlanib, maxsus ma’lumotlarni bazasini boshqarish tizimi dasturiy vositasi yordamida to‘ldirilishi, o‘zgartirilishi, takomillashtirilishi mumkin.

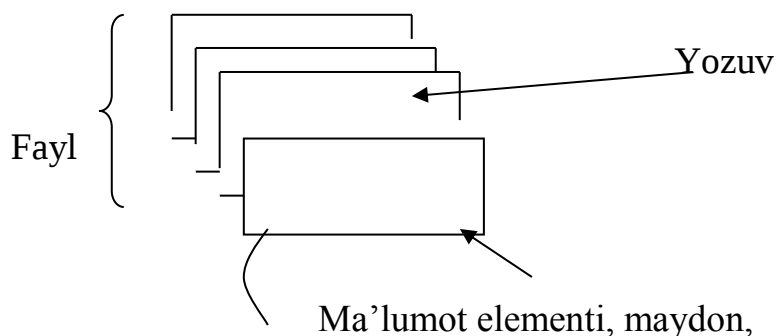
Aniq ma’lumotlarni (masalani) hal qilishda inson real dunyoni u yoki bu sohasi bilan cheklanadi. Bunday hollarda faqat ba’zibir ob’ektlarni o‘rganishgina qiziqish o‘yg‘otadi. Bunday ob’ektlarni majmuasini **predmet soxa** deyiladi.

Ob’ekt – bu ixtiyoriy predmet, xodisa, tushuncha yoki jarayondir.

Ma’lumot – bu uni ma’nosiga e’tibor bermay qaraladigan ixtiyoriy simvollar to‘plamidir. O‘zaro bog‘langan ma’lumotlar ma’lumotlar tizimi deyiladi.

Barcha ob’ektlar atributlari bilan xarakterlanadi. Masalan, ob’ekt sifatida fakultet, kutubxona, kompyuter va boshqalarni qarash mumkin. Jumladan, kompyuter ob’ektini atributi sifatida hisoblash tezligini, operativ xotira xajmi, o‘lchamlari va boshqalarni ko‘rish mumkin. Atributlarda saqlanadigan xabarlar ma’lumotlarni qiymatlari deyiladi. Masalan, operativ xotira xajmi 128 MB, EHM hisoblash tezligi sekundiga 5 mln.ta amal. Atributning qiymatlari mavjudki, ular yordamida ob’ektlarni identifikatsiyalash mumkin. Bog‘langan atributlarni qiymatlarni birlashtirsak, ma’lumot yozuvlarni hosil qilamiz. Tartiblangan

yozuvlarnig majmuasi ma'lumot fayli deyiladi.



Ma'lumotlarni nomlangan eng kichik birligi ma'lumot elementidir. U ko'pincha maydon deb aytiladi va bayt va bitlardan tashkil topadi. Ma'lumotlar agregati ma'lumot elementini nomlangan to'plamidir.

Ma'lumotlar bazasini ishlatish afzalliklari. Ma'lumot bazasini administratori. **MB administratori** deyilganda birorta shaxs yoki bir necha shaxslardan iborat bo'lgan va MBni loyihalash, uzatish va samarador ishlashini ta'minlovchidir.

Ma'lumotlar bazasi tushunchasi bilan ma'lumotlar banki tushunchasi ham mavjud (ishlatiladi). Ma'lumotlar banki (MBn) tushunchasi ikki xil talqin etiladi.

1. Hozirgi kunda ma'lumotlar markazlashmagan holda (ishchi o'rinlarda) ShK yordamida qayta ishlanadi. Ilgari ular alohida xonalarda joylashgan EHM larda (hisoblash markazlarida (HM)) markazlashgan holda qayta ishlangan. XM lariga axborotlar tashqi qurilmalar orqali kelib to'plangan. Ma'lumotlar bazasi markazlashgani hisobiga ularni ma'lumotlar banki deb atashgan. Bunda ma'lumotlarga murojat etish ishchi stansiyalardan markazlashgan xoldatashkil etilgan va shuning uchun ma'lumotlar banki bilan ma'lumotlar bazasi tushunchalari o'rtasida farq qilingan.

Hozirgi kunda ko'p hollarda ma'lumotlar bazasi markazlashmagan holda tashkil kilinmokda. Shuning uchun ma'lumotlar banki va ma'lumotlar bazasi sinonim suzlar sifatida xam ishlatiladi.

2. Boshkacha talkinda, ma'lumotlar banki deyilganda ma'lumotlar bazasi uni boshqarish tizimi(MBBT) tushuniladi.

Ma'lumot bazasi bilan ishlaydigan dasturni ilova (prilojenie) deb ataladi. Bitta ma'lumot bazasi bilan juda ko'p ishlashi mumkin.

MB ni ishlatish afzalliklari:

- Ixchamligi;
- Axborotlarni qayta ishlash tezligini oshishi;
- Kam mexnat sarfi;
- Har doim yangi axborot olish imkoniyati;
- Ma'lumotlarni ortiqchaligini kamayishi.

Nazorat savollari

1. Quyidagi terminlarni (tushunchalarni) tushuntiring:
 - a) ma'lumotlar;

- b) Ma'lumotlar bazasi;
 - v) Ma'lumotlar bazasini boshkarish tizimi (MBBT);
 - g) ma'lumotlardan mustakillik;
 - d) Xavfsizlik;
 - e) Butunligi;
 - j) tasvirlanish.
2. Predmet soha deganda nimani tushunasiz.
 3. Ma'lumotlar bazasiga qo'yiladigan talablarni tavsiflab bering.
 4. Ob'ekt atributi nima.

2–mavzu. Ma'lumot bazasi tizimining arxitekturasini. Uch bosqichli arxitektura. Tashqi bosqich. Konseptual bosqich. Ichki bosqich.

Reja

1. Ma'lumotlar bazasini sinflarga ajratish.
2. Ma'lumotlar bazasini uch bosqichli arxitekturasini.
3. Ma'lumotlarni fizik va mantiqiy tavsifi.
4. Ma'lumotlar bazasini boshqarish tizimini tashkil etuvchilari.

Tayanch iboralar: *Ma'lumotlar logik tasviri, arxitektura, administrator, konseptual, MBBT.*

Ob'ektlarni sinflarga ajratish deyilganda, barcha ob'ektlar to'plamini birorta norasmiy belgi bo'yicha qism to'plamlarga ajratishni tushunamiz. MB ni ko'pligini hisobga olib, uni sinflarga ajratish belgilari xilma – xil. Hozirgi kunda MB ni quyidagi sinflari ko'p ishlatiladi:

1. MB ma'lumotlarni tasvirlash shakliga qarab: video, audio, multimedia guruxlariga ajratish mumkin.
2. Video MB ma'lumotlarini ko'rinishiga qarab o'z navbatida matnli va grafik tasvirli bo'ladi.
3. Matnli MB ma'lumotlarni strukturalashganiga qarab strukturalashgan, qisman strukturalashgan va strukturalashmagan MB ga bo'linadi.
4. Strukturalashgan MB o'z navbatida ma'lumotlarni modeliga qarab: ierarxik, tarmoqli, relyatsion, ob'ektki relyatsion, ob'ektga yo'naltirilgan MB ga bo'linadi. Bundan tashqari strukturalashgan MB lari strategik va dinamik shuningdek, markazlashgan va taqsimlangan MB ga bo'linadi. MB ni foydalanuvchilar soniga qarab: bitta va ko'p foydalanuvchili MB ga bo'lamiz va ular ma'lumotlarni saqlanishiga qarab operatsion va analitik bo'ladi.

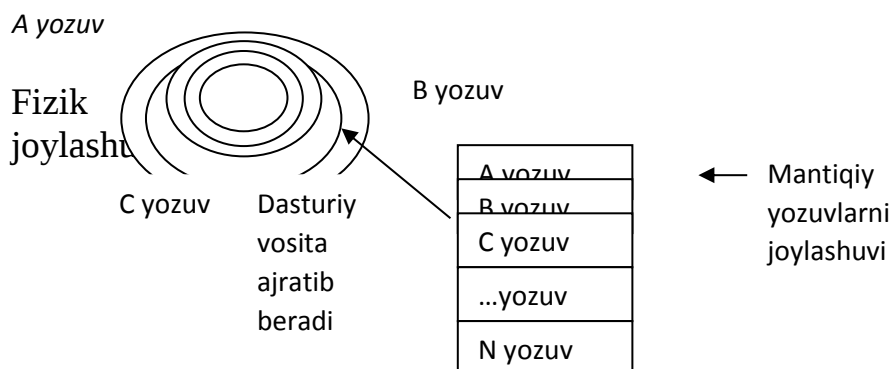
Sanab o'tilgan guruxlardan tashqari iqtisodiy nuqtai nazardan pulli va pulsiz MB ga bo'linadi. Shuningdek, murojaat qilish darajasiga qarab: ommabop va murojaati cheklangan MB ga bo'linadi.

MB ni logik va fizik tasvirlash. Ma'lumotlarni tavsiflash va ular orasidagi munosabat aloqalar o'rnatish 2 xil bo'ladi:

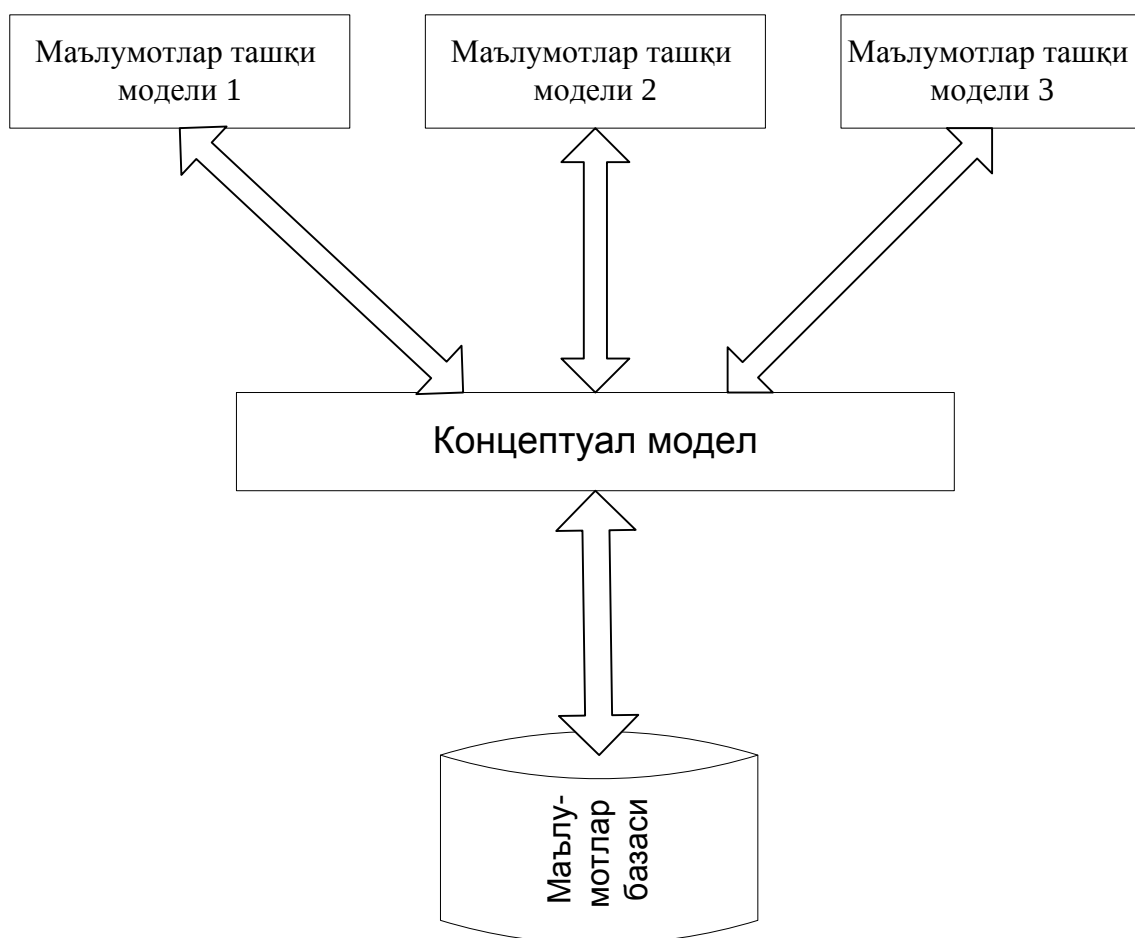
1. Logik yoki mantiqiy;

2. Fizik;

Fizik tasvirlashda ma'lumotlar mashinani tashqi xotirasida saqlashi bilan farqlanadi. Mantiqiy tasvirlashda esa amaliy dasturchi yoki foydalanuvchi tomonidan ma'lumotlarni tasvirlash ko'rinishi tushuniladi.



Ma'lumotlar bazasini boshqarish tizimining uch bosqichli arxitekturasini. Ma'lumotlar bazasini boshqarish tizimini qanday bo'lishini (qurilishi) o'rganishdagi ilmiy izlanishlar, ularni amalga oshirishni xilma – xil usullarini taklif qildi. Bulardan eng hayotiysi, Amerika standartlashtirish komiteti ANSI (American National Standards Institute) tomonidan taqdim etilgan MBni uch bosqichli tashkil qilish bo'lib chiqdi.



Ma'lumotlar bazasini boshqarish tizimini uch bosqichli modeli

1. Tashqi modellar – eng yuqori bosqich, bunda har bir model o'zini ma'lumotlar tasvir qabul qiladi. Har bir ilova, o'ziga kerakli zarur bo'lgan

ma'lumotlarni ko'radi va qayta ishlaydi. Masalan, ishchilarni malakasi bo'yicha taqsimlash tizimi, xizmatchi malakasi haqidagi ma'lumotlarni ishlatadi, uni oklad, manzili, telefoni haqidagi axborotlar qiziqitirmaydi va aksincha, oxirgi ma'lumotlar xodimlar bo'limi qism sistemasida ishlatiladi.

2. Konseptual bosqich – markaziy boshqarish zvenosi bo'lib, bunda MB eng umumiy holda tasvirlanib, u shu MB bilan ishlaydigan barcha ilovalar ishlatiladigan ma'lumotlarni qamrab oladi. Umuman konseptual bosqich MB yaratilgan predmet sohani umumlashgan modelini akslantiradi. Bu model ob'ektlarning muhim xossalarini akslantiradi.

3. Fizik bosqich – fayllarda joylashgan ma'lumotlarni tashqi axborot saqlovchilarida joylashishini belgilaydi. Bu arxitektura ma'lumotlar bilan ishlaganda mantiqiy va fizik mustaqillikni ta'minlab beradi.

Mantiqiy mustaqiliylik bitta ilovani o'zgartirishni, shu baza bilan ishlaydigan boshqa ilovani o'zgartirmasdan amalga oshirishni bildiradi.

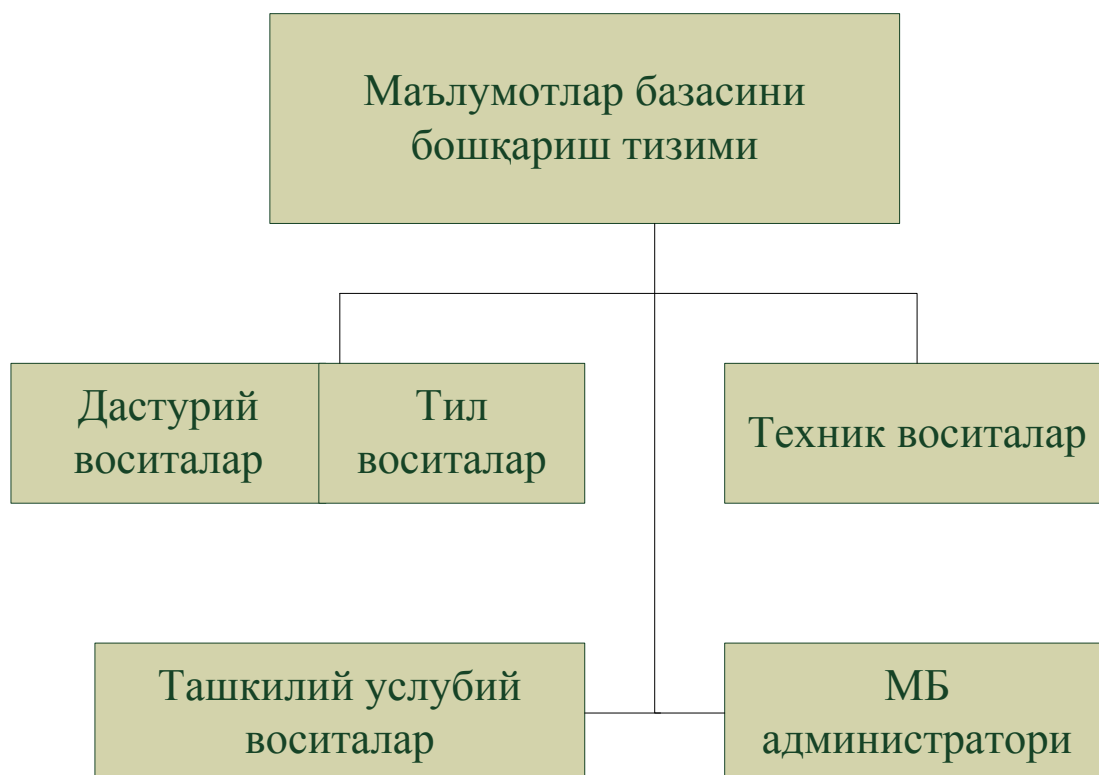
Fizik mustaqiliylik, saqlanuvchi ma'lumotlarni bir qattiq diskdan boshqasiga ko'chirganda uni ishlash qobiliyatini saqlab qolgan holda o'tkazishni bildiradi.

MBBTni tarkibi. MBBT shunday dastur qobig'iki, uning yordamida jadvallarni strukturasi, jadvallar orasidagi bog'lanish, jadvallarni ma'lumotlar bilan to'ldirgandan keyin, uning yordamida MB yaratiladigan programma vositasidir. Shu munosaabt bilan MBBT bir qancha tarkibiy qismlardan iborat.

Dasturiy vositalariga translyatorlar va MB ga ma'lumotlarni kiritadigan, qayta ishlaydigan, saqlaydigan, takomillashtiridigan, testdan o'tkazadigan, ma'lumotlarni kiritish chiqarishni ta'minlaydigan boshqarish tizimlari kiradi. Asosiy dasturlash tili sifatida Object Pascal, C++ kabi tillarni ishlatiladi.

MBBT ni paydo bo'lish tarixida 3 ta til qo'shib ishlatilgan:

1. Ma'lumotlarni tavsiflash tili – MTT (YaOD). Uni yordamida MB jadvallarini strukturalari quriladi



МББТ tarkibi

2. Ma'lumotlar bilan ishlaydigan til – MIT (YaMD). Bu til MB sini ma'lumotlar bilan to'ldirish va uni tiklash amallarni (olib tashlash, takomillashtirish va b.) bajarishda ishlatiladi.

3. So'rovlar tili – YaZ. Bu til yordamida qidirish mezonlari asosida kerakli axborotlarni topish va ularni chiqarish uchun xizmat qiladi.

Hozirgi kunda barcha aytilgan tillarni vazifasini SQL tili bajaradi.

Texnik vositalar sifatida, asosan, shaxsiy kompyuterlar va super kompyuterlarni ishlatamiz. Uslubiy – metodik vositalar – bu ko'rsatmalar, metodik va me'yoriy materiallarni majmuasi bo'lib, ular yordamida MB va MBBT dan foydalanish yo'llari ko'rsatiladigan vositalaridir. MBBT dan ikki gurux shaxslari foydalanadi:

1. Chekli yoki oddiy foydalanuvchilar;
2. MB administratori;

MB administratorini xizmat doirasiga quyidagi ishlar kiraddi:

- a) Predmet sohani tahlili va foydalanuvchilar va axborotni o'rnini aniqlash;
- b) Ma'lumotlarni tuzilishini loyihalash va ularni takomillashtirish;
- c) Qo'yilgan topshiriqlar va ma'lumotlarni bir butunligini ta'minlash;
- d) MB ni yuklash va yuritish;
- e) Ma'lumotlarni himoya qilish;
- f) MB ni tiklashni ta'minlab berish;
- g) MB ga murojaatlarni yiqish va statistik qayta ishlab berish;
- h) MB ga ko'p foydalanuvchilar rejimida ishlaganda, ma'lumotlarni o'chib

ketishidan ximoya qilish;

- i) Texnik vositalar nosoz bo'lib ishdan chiqqanda, ma'lumotlarni saqlash va qayta tiklash ishlarini bajarish;

Nazorat savollari

1. Ma'lumotlarni qaysi belgilari bo'yicha sinflarga ajratamiz.
2. Ma'lumotlarni logik (mantiqiy) va fizik tasvirlash.
3. Ma'lumot bazasini uch bosqichli arxitekturasini nima.
4. Ma'lumotlar bazasi administratorini asosiy vazifalarini aytib bering.
5. Ma'lumotlar bazasida tasvirlar qanday saqlanadi. Misollar keltiring.
6. Ma'lumotlar bazasini uch bosqichli arxitekturasini chizmasini tasvirlang.
7. So'rovlarni qayta ishlashda MBBTning bajaradigan ishlar ketma-ketligini tavsiflab bering.

3-mavzu. Ma'lumot baza modeli. Moxiyat-aloqa modeli.

REJA:

1. Ma'lumot modeli tushunchasi.
2. Ierarxik (shajara) ma'lumot modeli.
3. Tarmoqli ma'lumot modeli.
4. Predmet sohani tavsiflash usullari.
5. Mohiyat-aloqa diagrammasi.

Tayanch iboralar: infologik model, datalogik model moslik, Ulman Chen usuli, mohiyat, ma'lumot modeli, ierarxik, tarmoqli model, relyatsion model.

Zamonaviy MBBTlari datalogik bosqichda xilma-xil MB bilan ishlashni ta'minlaydi. Hozirgi kunda eng ko'p o'rganilgan va keng ishlatiladigan ma'lumot modellari kiradi. Ma'lumot modeli, bu MB ni ma'lumot elementlari to'plami orasidagi bog'lanish strukturalarini tasvirlovchi umumiy sxemadir. Ma'lumot modeli tushunchasini aniq ta'rifini Kodd bergan. U ma'lumot modelini 3 ta kerakli komponentasini keltirgan:

1. mavjud bo'lgan ma'lumot strukturalarini aniqlash vositalari majmuasi;
2. ma'lumotlarni qidirish va takomillashtirish uchun MB xolatiga qo'llaniladigan amallar to'plami;
3. oshkor holda MB holatini aniqlovchi va bir butunligini ta'minlovchi vositalar to'plami;

Xozirgi kunda klassik xisoblashlarda 3 ta ma'lumot modeli ko'p ishlatiladi:

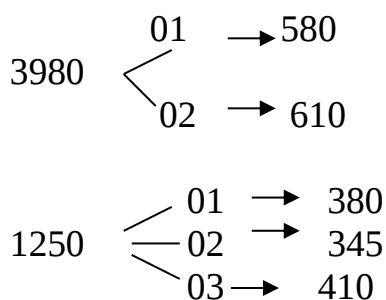
- Ierarxik ma'lumot modeli;
- Tarmoqli ma'lumot modeli;
- Relyatsion MM.

Har bir MB u yoki bu model asosida yoritiladi. Har bir MBBT esa u yoki bu ma'lumot modelini ta'minlaydi deyiladi. Masalan, ierarxik ma'lumot modeliga asoslangan tizim – **ines** tizimidir. Tarmoqli modellarda esa – **BANK OS**, **SETOR**,

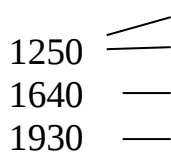
relyatsion modelga asoslangan tizimlar – **Access, KARAT** va boshqalar.

Ierarxik modellar. Ierarxik modellarda yozuvlar daraxtsimon strukturali ko‘rinishda bo‘ladi. MBBT lardan ba’zi birlari faqat ierarxik tuzilishga ega bo‘lganlari bilan ishlatiladi. Ierarxik tuzilishli ma’lumotlar sodda yaratiladi. Bu ko‘pincha tadbiqlarda qulay, lekin ma’lumotlarni ko‘plari daraxtsimon strukturali bog‘lanish tabiatiga ega emas.

Misol, faraz qilaylik 2 ta firma ishlab chiqargan mahsulotning barcha turlarini narxlari berilgan. SHu ma’lumotlarni narxlar ma’lumotnomasi qurilsin va kompyuter xotirasiga joylashtirilsin. Faraz qilamiz, A va V firmalar mos ravishda 2 xil ko‘rinishdagi mahsulot chiqaryapti. Har bir mahsulot ko‘rinishi har xil texnologiya asosida bajariladi. Bunda uning narxi ham shunga qarab bo‘ladi. 3980 mahsulotni 2ta sxema asosida tayyorlaymiz ,ularni 01, 02 deb belgilaymiz va ular kuyidagi narxni belgilaydi 578,612. 1250 maxsulotni 3ta sxema asosida tayyorlaymiz ,ularni 01, 02,03 deb belgilaymiz va ular kuyidagi narxni belgilaydi 380,345, va 410. V firma 3 xil maxsulot ishlab chikaradi ,ularni kodi mos ravishda 1250, 1640 va 1930 kodga ega bo‘lsin. Ular xam o‘zlarini ishlab chikish sxemasi va narxiga ega bo‘lsin

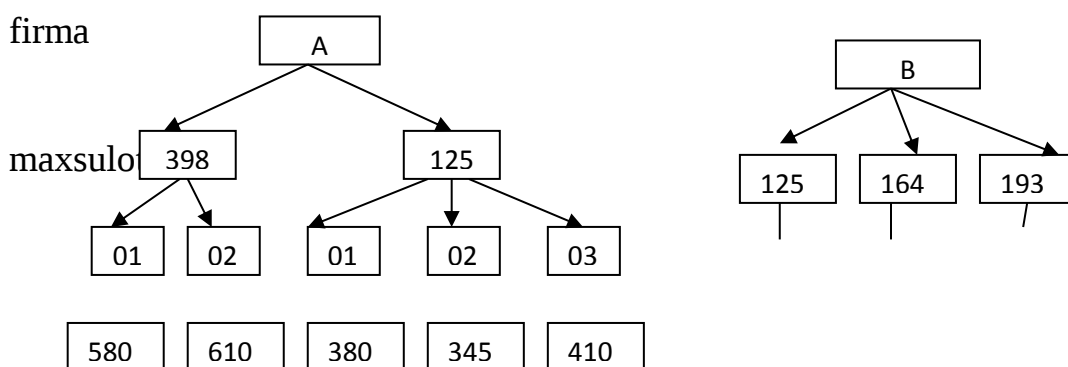


V firma 3 xil mahsulotni ishlab chiqaradi: 1250, 1640, 1930.



Barcha maxsulotlar nomlari, ularni ishlab chiqish sxemasi va narxlarni uz ichiga oluvchi ma’lumotnoma tuzish va mashina xotirasiga joylash talab etilsin.

Keltirilgan ma’lumotlar strukturani ierarxik (daraxtsimon) ko‘rinishda tasvirlash kulay.





Bunday strukturani jadval ko`rinishida ham tasvirlash mumkin.

Format	Mahsulot	Texnologiya	Narx
A	3980	01	586
A	3980	02	610
A	1250	01	380
A	1250	02	345
A	1250	03	410

Ma'lumotni tarmoqli modeli.

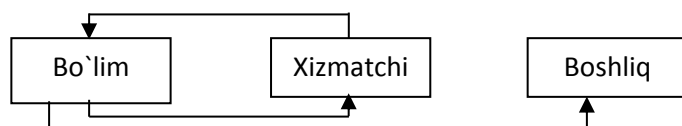
Agar berilgan element bir nechta o'zidan yuqori elementga suyansa tarmoqli ma'lumot elementiga ega bo'lamiz.

Tarmoqli ma'lumotlar modeli. Agar munosabatdagi joriy element bir necha berilgan elementga ega bo'lsa, bunday bog'lanishlarni ierarxik strukturalar bilan tavsiflab bo'lmaydi.

Bunday tuzilmalar tarmoqli graflar bilan tavsiflanadi. Tarmoqli strukturalarida element ixtiyoriy boshqa element bilan bog'lanishi mumkin. Ya'ni, tarmoqli bir necha kichkina ob'ektlardan topgan yirik ob'ekt deb qarash mumkin.

Ushbu chizmada bog'lanishlarni tarmoqli modelda tasvirlanishi keltirilgan. Shunday qilib, tarmoqli model ma'lumotlar elementlari orasidagi xilma-xil bog'lanishlarni ixtiyoriy ko'rinishdagi grafik yordamida akslantiradi. Tarmoqli model yozuvlar to'plami va mos bog'lanishlar to'plamidan tashkil topadi. Bog'lanishlarni yaratish uchun alohida cheklanishlar qo'yilmaydi. Misol: oddiy tarmoqli MB sxemasi sifatida quyidagini keltirish mumkin.

Bo'limda ishlaydi

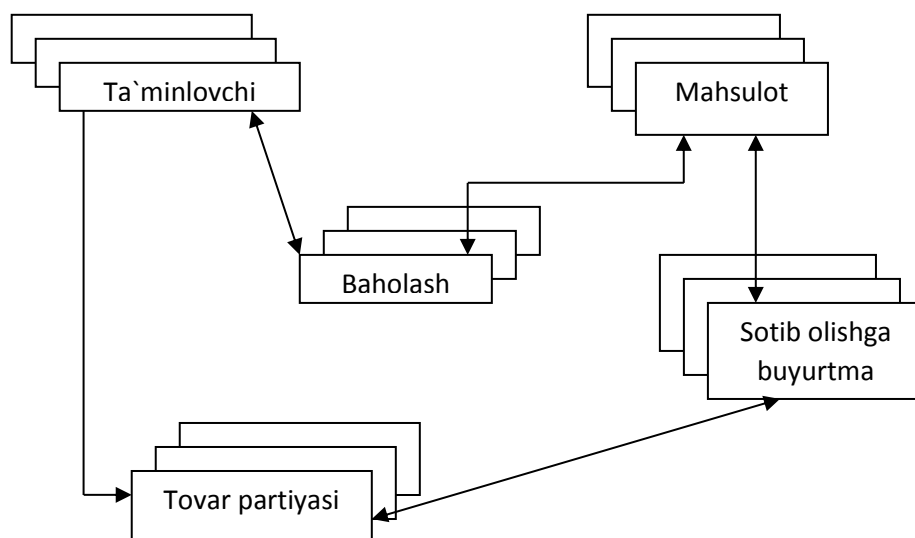


Boshliqqa ega

Tarmoqli MB turida ma'lumotlar bilan quyidagi ishlarni bajarish mumkin.

1. MB yozuvlarini qidirish
2. Yangi yozuvni yaratish
3. Joriy yozuvni olib tashlash
4. Joriy yozuvni tiklash
5. Yozuvni bog'lanishga qo'shish
6. Yozuvni bog'lanishdan olib tashlash
7. Boglanishlarni o'zgartirish

Misol:



Yuqoridagi sanab o`tilgan MM dan tashkari hozirgi kunda quyidagi MM lar ham amaliyotga kirib kelmokda.

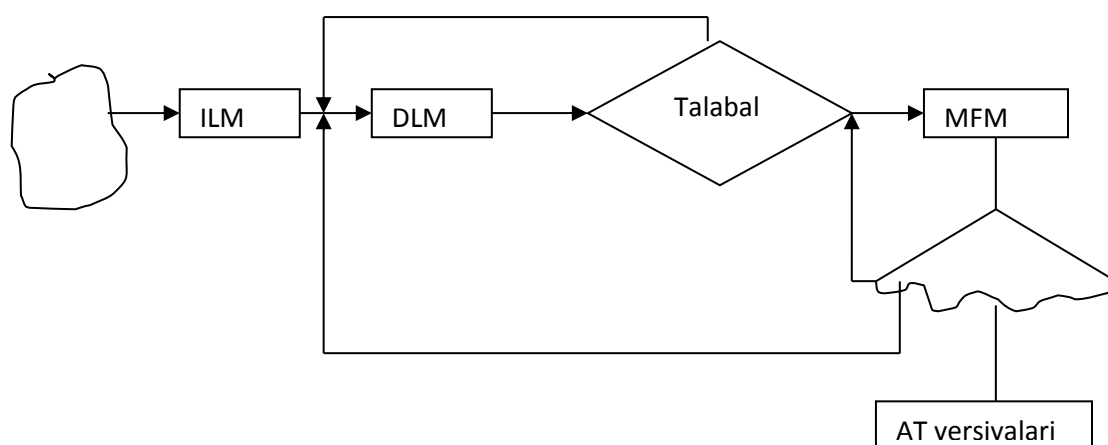
1. Ko`p o`lchamli MM lar
2. Ob`ektga yo`naltirilgan MM lar

Shuningdek boshqa MM larga asoslangan har xil tizimlar ham ishlab chiqilmoqda. Bular qatorida quyidagilarni sanash mumkin.

1. Ob`ekt- relyatsion
2. Semantik
3. Yo`naltirilgan
4. Konseptual va boshqalar

Ulardan ba`zilar MB bilimlari bazasi va dasturlash tillarini integratsiyalashga (birlashtirishga) xizmat qiladi.

Hozirgi kunda axborot tizimlarini loyihalash xilma-xil usullari mavjud. Umuman olganda, axborot tizimlarini dastuuriy ta'minotini yaratish iterativ xarakterga ega. Axborot tizimlarini loyihalashni asosiy bosqichlari va ular orasidagi bog`lanish quyidagi chizmada keltirilgan:

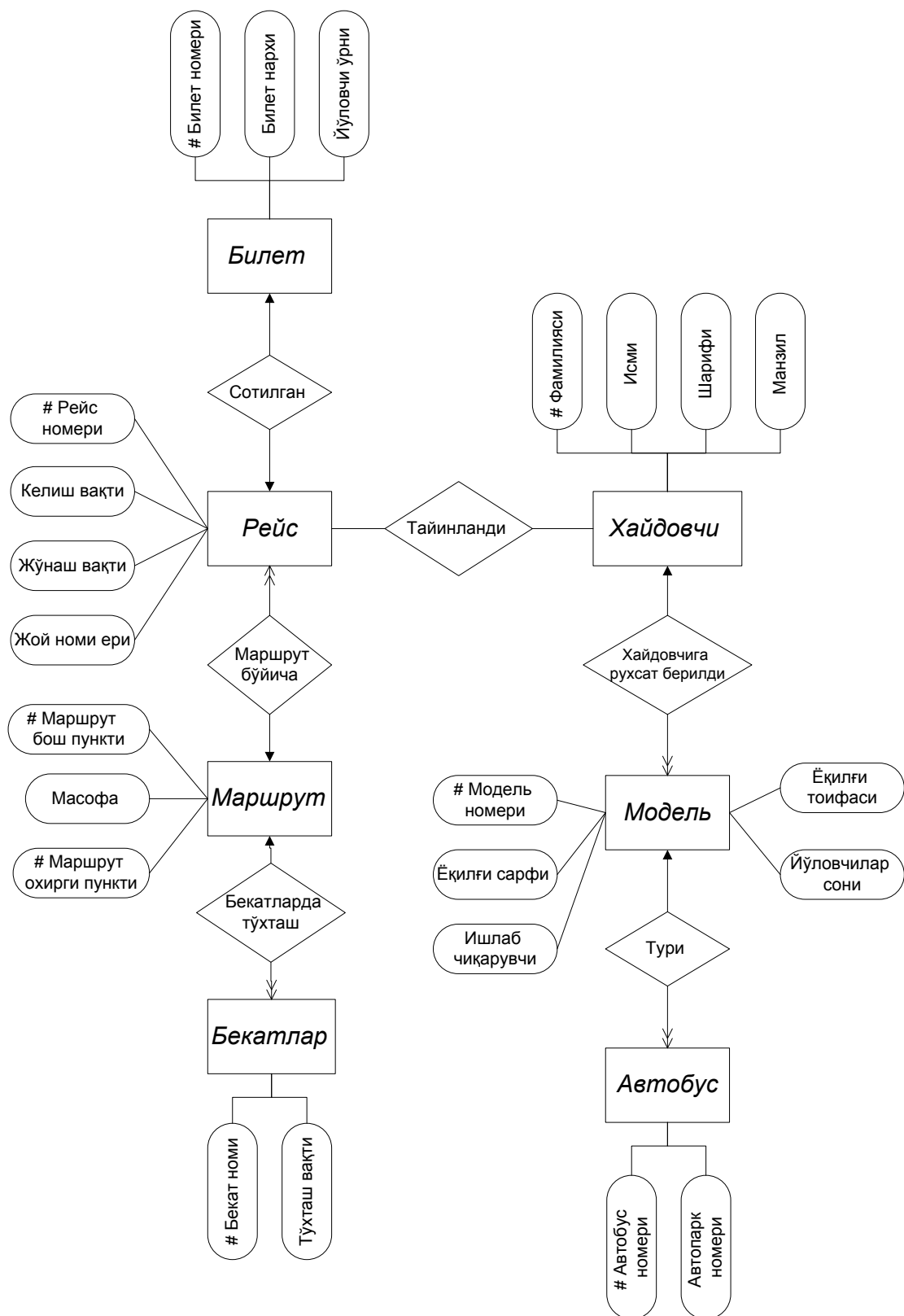


PS – predmet soha;
ILM – infologik model;
DLM – datalogik model;
MFM – ma'lumotlarni fizik modeli;
AT – axborot tizimi;

AT loyihalashni 1 – bosqichida predmet sohasida mantiqiy informatsion tuzilmasini quramiz. U PS ni va foydalanuvchini talablarini o'zida mujassamlashtiradi. Bunda biz aniq MBBT ga bog'lanmagan ravishda bu ishlarni bajaramiz, ya'ni PS ni informatsion-logik tavsifi bajariladi. Bu bosqich infologik model qurish bosqichi deb ataladi. MBBT vositasi yordamida ma'lumotlarni mantiqiy bog'lanishlarini tashkil qilish ma'lumotlar bazasini DLM ini bildiradi. bu model yordamida ma'lumotlar elementlari orasidagi mantiqiy bog'lanishlarni aks ettiradi. DLM ni ma'lumotlarni saqlash muhiti bilan bog'laydigan bosqich ma'lumotlarni fizik modeli deyiladi.

Hozirgi kunda PS ni tavsiflash uchun ko'p usullar mavjud. SHulardan biri ob'ekt-aloqa usulidir. Bu usulni ba'zan Ulman – CHen usuli ham deyiladi. PS ni mohiyat aloqa usulida tavsiflaganda quyidagi bosqichlarda i sholib boriladi:

1. PS ni ob'ektlari aniqlanadi.
2. Ob'ekt sohalari (atributlari) belgilanadi va uning kalit parametri aniqlanadi. Kalit parametri ob'ektni identifikatsiyalaydi.
3. Ob'ektlar o'rtasida aloqa o'rnatiladi va ular sinflarga ajratiladi.
4. Maxsus belgilar kiritilib, ob'ekt aloqa diagrammasi o'rnatiladi. Bu diagramma PS ning infologik modeli grafik tasviri hisoblanadi.



Ulman – CHen diagrammasi

Nazorat savollari

1. Ma'lumot baza modeli nima?
2. Ierarxik (shajara) modeli ma'lumot va uning asosiy xarakteristikallari
3. Tarmoqli model ma'lumot va uning asosiy xarakteristikallari
4. PS moxiyat aloqa usulida tavsiflaganda qanday ishlar bajariladi

5. Mosliklarni (munosabatlar) qanday turlari bor . Ularni tavsiflang.
6. Moxiyat aloqa diagrammasi qanday quriladi.
7. Axborot tizimlarini loyixalashga infologik yondoshishni asosiy koidalarini tushuntiring.
8. Mohiyat-aloka modelini asosiy konstruksiyalarini tushuntirib bering.

4–mavzu. Relyatsion ma’lumot modeli. Ma’lumotlar bazasida munosabatlar.

REJA

1. Ma’lumotlar bazasida munosabatlar.
2. Ma’lumotlarni tasvirlashda jadvallardan foydalanish.
3. Relyatsion ma’lumotlar bazasini asosiy tushunchalari.
4. Relyatsion algebra va uning amallari.

Tayanch iboralar: kortej, daraja, domen, relyatsion hisoblash.

Ma’lumotlarni relyatsion modeli asosida munosabat tushunchasi yotadi. Munosabatni 2 o’lchamli jadvallar yordamida tavsiflash qulay. Jadval tushunarli ko’rimli va inson uchun oddiy. Munosabatlar to’plami ma’lumotlarni saqlash uchun ishlatilishi mumkin. Shu bilan birga ular orasidagi bog’lanishlarni modellashtirish imkonini beradi.

Misol: 2 ulchamli jadvalni kuramiz.

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

Yuqoridagi jadvalda

1-satr-atribut nomlari

2-satr atributlarni qisqa nomlari

3,4,5-satr atribut qiymatlari;

Yukoridagi ko’rib chiqilgan ierarxik tarmoqli va boshqa ma’lumotlarni tasvirlash usullarini shunday 2 o’lchamli jadvalga keltirish mumkin. Bunday jadvallarni quyidagi xususiyatlari bo’ladi.

1. jadvalni xar bir ma’lumot elementi maydon xisoblanadi va takrorlanuvchi guruxlar bo’lmaydi;
2. barcha ustunlar bir jinslidir;
3. xar bir ustunga nom tayinlangan;

4. jadvalda bir xil satr 2 marta uchramaydi;
5. bunday jadvalda satr va ustunlar ixtiyoriy tartibda karaladi va ixtiyoriy ketma-ketlikda ishlatilishi mumkin.

Bunday xususiyatlarga jadvalar munosabat deyiladi. Munosabat asosida qurilgan MB relyatsion MB deyiladi.

Yuqoridagi jadvalni sxematik qisqartirilgan ko`rinishi (sxemasi)

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

Bunday yozuv ma'lumot baza sxemasi deyiladi

Shunday qilib, relyatsion MB ma'lumot elementlar to`plami asosida quriladi.

Munosabat yoki jadvalni kortejlar to`plami deb qarash mumkin.

Agar jadvalda n ta ustun bo`lsa, u n tartibli kortejdan iborat deyiladi va munosabat ham n -darajali deyiladi.

Har bir atribut qiymatlari to`plami domen deyiladi.

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

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

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

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

Ob'ektlarni tahlil qilib bo`lingandan so`ng, shu ob'ektga qo'yiladigan boshlang'ich so`rovni ishlab chiqamiz. Avtovakzalni faoliyati iqtisodiy va texnik ko`rsatgichlar bilan bog`liq bo`lganligi uchun, yo`lovchilarga informatsion

ma'lumot berganligi uchun yaratayotgan informatsion sistemamiz quyidagi so'rovlarga javob berishi kerak:

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

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

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

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

Ekvivalent (sinonim) tushunchalar

Fayl	Jadval	Munosabat	moxiyat
Yozuv	Satr	Kartej	Moxiyat nusxasi
Maydon	Ustun	Atribut	Atribut

Relyatsion MB munosabatlarida strukturali va semantik axborotlar saqlanishi mumkin. Strukturaviy axborotlarni biz munosabat sxemalar yordamida bilamiz.

Semantik axborotlar esa munosabat sxemalarda ma'lum bo'lgan va hisobga olinadigan va atributlar o'rtasidagi funksional bog'lanishlar bilan ifodalanadi. MBdagi munosabatlarda atributlarni tarkibi 2 quyidagi talabga javob berishi kerak.

1. Atributlar o'rtasida noxush funksional bog'lanishlar bo'lmasligi kerak.
2. Atributlar guruxlanishi ma'lumotlar takrorlanishidan eng kam holatining taxlillash kerak va ular qayta ishlash va tiklashni qiyinchiliksiz amalga oshirilishi kerak.
3. Qo'yilgan MB munosabatlari normallasadi. Munosabatlar normalashtirish MB da berilgan munosabatlarni dekompozitsiya (ajratish) jarayoni yordamida sodda va kichik munosabatlar hosil qilishdir.

Talaba kodi	Famliyasi	Telefon	Talaba
	Ashurov	4767777	2341717

	Soliev	1365556	2341717
	Soliev	1365656	2485888
	Amirov	2351717	2485888
	Amirov	2381817	
	Amirov	2351817	

Har bir munosabatda kortejlar identifikator kalitiga ega bo'lishi kerak. Kalit quyidagi ikkita xossaga ega bo'lishi kerak:

1. Kartej kalit qiymati bilan bir qiymatli ifodalanishi kerak;
2. Kalitda ortiqchalik bo'lmashligi kerak. Bu degani hech qanday atributni kalitdan olib tashlash mumkin emas.

Relyatsion MB da axborotlarni ortiqchaligini normallashtirish yo'li bilan kamaytiriladi. Jadvallar ustida har xil amallar bajarish mumkin. Bu amallarni tartiblab ishlab chiqqan odam Kodd. Amallarga quyidagilar kiradi:

- ❖ To'plamlar ustida birlashtirish, kesishuv, ayirma, dekart ko'paytma va bo'lish amallari kiradi.
- ❖ Maxsus relyatsion amallar, ularga: proeksiya, birlashtirish, ajratish (tanlab olish) amallari kiradi.

Munosabatlar ustida amalni bajarish uchun ishlatiladigan tillarni ikki sinfga ajratishimiz mumkin:

- a) Relyatsion algebra tillari;
- b) Relyatsion hisoblash tillari.

Munosabatlar o'z mazmuniga qarab ikki sinfga ajratiladi:

- a) Ob'ekli munosabatlar;
- b) Bog'lanuvchi munosabatlar;

Ob'ekli munosabatlarda ob'ektlar haqidagi munosabatlar saqlanadi. Masalan, talaba munosabati. Bog'lanish munosabatlarida asosan, ob'ekli munosabatlarning kalitlari saqlanadi. Kalit atributlari oddiy va murakkab bo'lishi mumkin. Agar kalit ikkita va undan ortiq atributdan tashkil topgan bo'lsa, murakkab hisoblanadi.

Familiya	Kurs	Mutaxassislik
Sobirov	2	Matematika
Aliev	4	Fizika
Xabirov	3	Ximiya

Talaba	Fan
Sobirov	Algebra
Aliev	Tarix
Aliev	Algebra
Xabirov	Programmairi vaniya

Nomi	Semestr
Algebra	4
Tarix	2
Programm.	1

Nazorat savollari

1. Relyatsion ma'lumotlar bazasini asosiy tushunchalari.

2. Munosobat xossalari
3. Munosobatlar sxemasi.
4. Relyatsion algebra amallari.

5 – mavzu. Relyatsion algebra va relyatsion xisobot elementlari

REJA

5. Ma'lumotlar bazasida munosobatlar.
6. Munosabatlar ustida amallar.
7. Relyatsion ma'lumotlar bazasini asosiy tushunchalari.
8. Relyatsion algebra va uning amallari.

Tayanch so'zlar: dekart, kesishuv, birlashtirish, domen, relyatsion.

Munosabatlar ustida amallar. Munosabatlar ustida xar-xil amallar bajariladi. Relyatsion ma'lumotlar modelini xususiyatlaridan biri ma'lumotlarni qayta ishlashni relyatsion algebra operatorlari (amallari) yordamida amalga oshirishdir. Relyatsion algebrada quyidagi 8 ta operator keng ishlatiladi. Ulardan 4 tasi an'anaviy tuplamlar ustidagi amallar kiradi.

An'anaviy (qabul qilingan) amallarga quyidagilar kiradi.

1. Birlashtirish
2. Kesishuv
3. Ayirma
4. Dekart ko'paytma

Maxsus amallarga esa quyidagilari kiradi:

1. Tanlash (seleksiya)
2. Proeksiya
3. Qo'shish
4. Bo'lish

Munosabatlar ustida bajariladigan birlashtirish, kesishuv, ayiruv amallari operatorlarning tili yoki turi bo'yicha mosligini talab etadi. 2 ta munosabat tipi bo'yicha mos keladi, agarda ularda ekvivalent munosabat sxemasi bulib:

1. ulardagi xar bir darajasi bir xil bo'lsa yoki ular bir xil atribut to'plamiga ega bulsa;
2. sxema atributlarini shunday tartiblash mumkinki, bir xil o'rinda turib solishtirilayotgan atributlari bir xil domenda aniqlangan bo'lishi kerak.

Misollar: 1. Birlashtirish amali

Familiyasi	Yosh
Karimov	20
Odilov	23
Isaev	35
Aliev	49

Familiyasi	Yoshi
Eragshiev	19
Ilxomov	30
Karimov	20
Azizov	51

Mijoz 1 V Mijoz 2

Familiyasi	YOshi
Karimov	20
Odilov	23
Isaev	35
Aliev	49
Ergashev	19
Ilxomov	30
Azizov	51

Birlashtirish operatori

2.) Mijoz 1 $\wedge$ Mijoz 2

Familiyasi	YOshi
Karimov	20

Kesishuv operatori

3.) Mijoz/mijoz 2

Familiyasi	YOshi
Odilov	23
Isaev	35
Aliev	49

Ayirma operatori

2.) Dekart ko'paytmada munosabat operatorlari xar-xil sxemada bo'lishi mumkin.

Familiyasi
Alimov
Ashurov
Oripov

Fan	Sana
Matem	09.01.2009
Tarix	14.01.2009

Matematik munosabatlar darajasi operant munosabat darajalarining yig'indisiga teng. Kuvvati esa operant kuvvatlarini ko'paytmasiga teng. Kuyidagni jadvalda ularning dekart kupaytmasi keltirilgan.

Kaydnoma

Familiya	Fan	Sana
Alimov	Matem	02.01.09
Alimov	Tarix	14.01.09
Ashurov	Matem	09.01.09
Ashurov	Tarix	14.01.09
Oripov	Matem	09.01.09
Oripov	Tarix	14.01.09

Celeksiya (tanlash) amali 1 ta munosabat ustida bajariladi. Natija munosabatda biror shart buyicha tanlab olingan kartejlar katnashadi.

Kushish amali 2 ta operant ustida bajariladi. Xar bir munosabata kaysi atribut buyicha kushish bajarilayotgan bulsa, u ajratiladi.

Natija munosabat 1 va 2-munosabatni barcha atributlarini uz ichiga oladi.

Misol:

Gurux

Mutaxassislik	Talaba kodi
Matematika	1
Fizika	4
Ximiya	5

Talaba

Talab kodi	Familiya	Kurs
1	Dibrov	1
2	Sattorov	1
3	Pulatov	2
4		1
5	Ashurov	3

Sardor

Mutaxassislik	Talaba kodi	Familiya	Kurs
Matem	1	Diyorov	1
Fizika	4		1
Ximiya	5	Ashurov	3

Normalashtirish

Har bir munosabatda kortejlar identifikator kalitiga ega bo'lishi kerak. Kalit quyidagi ikkita xossaga ega bo'lishi kerak:

3. Kartej kalit qiymati bilan bir qiymatli ifodalanishi kerak;
4. Kalitda ortiqchalik bo'lmasligi kerak. Bu degani hech qanday atributni kalitdan olib tashlash mumkin emas.

Relyatsion MB da informatsiyalarni ortiqchaligini normallashtirish yo'li bilan kamaytiriladi. Jadvallar ustida har xil amallar bajarish mumkin. Bu amallarni tartiblab ishlab chiqqan odam Kodd. Amallarga quyidagilar kiradi:

- ❖ To'plamlar ustida birlashtirish, kesishuv, ayirma, dekart ko'paytma va bo'lish amallari kiradi.
- ❖ Maxsus relyatsion amallar, ularga: proeksiya, birlashtirish, ajratish (tanlab olish) amallari kiradi.

Munosabatlar ustida amalni bajarish uchun ishlatiladigan tillarni ikki sinfga ajratishimiz mumkin:

- c) Relyatsion algebra tillari;
- d) Relyatsion hisoblash tillari.

Munosabatlar o'z mazmuniga qarab ikki sinfga ajratiladi:

- c) Ob'ekli munosabatlar;
- d) Bog'lanuvchi munosabatlar;

Ob'ekli munosabatlarda ob'ektlar haqidagi munosabatlar saqlanadi. Masalan, talaba munosabati. Bog'lanish munosabatlarida asosan, ob'ekli munosabatlarning kalitlari saqlanadi. Kalit atributlari oddiy va murakkab bo'lishi mumkin. Agar kalit ikkita va undan ortiq atributdan tashkil topgan bo'lsa, murakkab hisoblanadi.

Familiya	Kurs	Mutaxassislik
Sobirov	2	Matematika
Aliev	4	Fizika

Xabirov	3	Ximiya
---------	---	--------

Talaba	Fan
Sobirov	Algebra
Aliev	Tarix
Aliev	Algebra
Xabirov	Programmairi vaniya

Nomi	Semestr
Algebra	4
Tarix	2
Programm.	1

Relyatsion algebra va uning amallari.

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

- 1) Eng pastki bosqich – kortej deb ataladi. Bunda dasturchi yozuvlar yoki kartijlar bilan ishlaydi.
- 2) Relyatsion algebra deyiladi. Bunda foydalanuvchi munosabatlar ustida yuqori bosqichli amallar to'plamini kiritadi.
- 3) Eng yuqori bosqich – hisoblash bosqichi. Bunda foydalanuvchi bevosita kompyuterga maxsus tillarda murojaat qiladi va mashina bu murojaatni qabul qiladi.

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

- 1) Birlashtirish , R va S munosabatlarni birlashtirish $R \cup S$ ko'rinishida berilib, bu amalni natijasi R munosabatga tegishli bo'lgan yoki S munosabatga tegishli bo'lgan yoki ikkalasiga ham tegishli bo'lgan kartejlar to'plamidir. Bu amallarni bajarayotganda bir xil tartibda bo'lishi kerak. Natijani tartibi ham operandlar tartibiga teng bo'ladi.
- 2) Ayirma R va S munosabatlarni ayirmasi $R - S$ ko'rinishida yoziladi va undagi kortejlar to'plami R munosabatga tegishli, lekin S munosabatga tegishli bo'lmagan kortejlardir. Bu amalni bajarganda ham operandlarni tartibi bir xil bulishi kerak.
- 3) Dekart ko'paytma. Bizda R va S munosabat berilgan bo'lsin. R munosabatni tartibi $R - R$ va S munosabatniki $S - S$ ga teng bo'lsin. Unda dekart ko'paytma $R * S$ ko'rinishida yozilib, uning natijasi uzunligi $R + S$ ga teng bo'lgan kortejlar to'plamidan iborat bo'lib, bu kortejlarni birinchi R komponentasi R kortejga teng bo'ladi, qolgan S komponentasi S kortejga teng bo'ladi.
- 4) Proeksiya, R munosabatga bu amal tadbiiq etilganda, R munosabatdan

ba'zi bir komponentalar olib tashlanadi. Qolganlari esa qaytadan tartiblanadi.

- 5) Seleksiya tanlash. Bu amal bajarilganda operandlar sifatida munosabat atributlari ishtirok etadi va solishtirish arifmetik amallari: $=$, $\neq$, $\leq$, $\geq$, $<$, $>$ va mantiqiy amallar: va (U), yoki (V), not amallari ishlatiladi.

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

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

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

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

Nazorat savollari

1. Relyatsion ma'lumotlar bazasini asosiy tushunchalari.
2. Munosobat xossalari.
3. Munosobatlar sxemasi.
4. Relyatsion algebra amallari.
5. Relyatsion hisoblash amallari.

7–mavzu. Ma'lumotlar bazasini normallashtirish. Normal formalar 1NF, 2NF, 3NF va Kodd

REJA:

1. MBni normallashtirish.
2. Funktsional bog'lanishlar va ularni turlari.
3. 1NF.
4. 2NF.
5. 3NF.

Tayanch iboralar: funktsional bog'lanish, normal forma, anomaliyalar.

MB uzluksiz o'zgarib turadi. Unda yangi ma'lumot elementlari qo'shiladi. Ular orasida yangi aloqalar yoki bog'lanishlar o'rnatiladi va ularni qayta ishlashni yangi usullari qo'llaniladi. Bu jarayonda imkoni boricha foydalanuvchi yaratgan MB bilan ishlash uchun yaratilgan dastur ilovasini kam o'zgartirishga harakat qiladi. Bu muammoni hal qilish uchun ma'lumot elementlarini asosli ravishda guruxlarga birlashtirish va ular uchun kalitlarni aniqlash yo'li bilan hal qilinishi mumkin. Hozirgi kunda axborot tizimlari ishlab chiqaruvchilar ma'lumotlarni 3 – normal formada tasvirlab ishlatishni taklif etadilar.

Funksional bog'lanish tushunchasi. Relyasion MB da ma'lumotlarni strukturasidan tashqari ularni sxematik informatsiyasiga ham etibor beriladi. MB ni strukturalari haqidagi informatsiya munosabat sxemasi yordamida beriladi. Sxematik informatsiyalar esa atributlar orasidagi funksional bog'lanishlar orqali ifodalanadi. MB munosabatlarida atributlarni tarkibini quyidagi talablarga javob beradigan qilib guruxlash kerak:

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

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

Misol. A va V atributlar berilgan bo'lsin. Agar ixtiyoriy vaqtda A atributni bittadan ortiq bo'lmagan qiymatimos kelsa, unda V atributda funksional bog'langan deyiladi va quyidagicha belgilanadi:

$$\left. \begin{array}{l} A \rightarrow V \text{ Shaxsiy nomer} \rightarrow \text{Familiya} \\ \text{Masabi} \rightarrow \text{Maosh} \end{array} \right\} \text{ bog'lanishlar}$$

1 – normal forma.

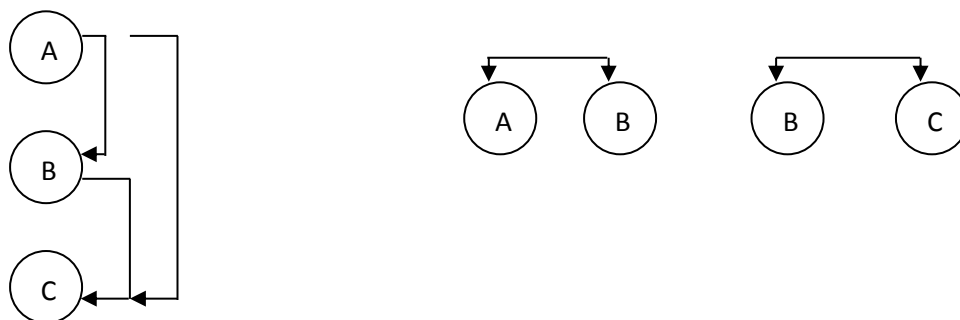
SHaxsiy nomer	Predmet nomi	Soatlar soni	Familiya	Mansabi	Maoshi	Kafedra	Tel.
201	EHM	36	Ergashev	Dots.	70000	EVM	4-89
201	SHK	72	Ergashev	Dots.	70000	EVM	4-89
202	MBBT	48	Komilov	Dots.	70000	EVM	4-89
301	MBBT	48	Babaev	Prof.	100000	ASU	5-19
401	Fizika	52	G'aniev	Ass.	50000	FE	4-12
401	Optika	20	G'aniev	Ass.	50000	FE	4-12

Agar munosabat 1-normal formada bo'lsa – 1nf, unda barcha kalit bo'lmagan atributlar kalit atributga funksional bog'langan. Lekin, bog'lanish darajasi har xil. Agar kalit bo'lmagan atribut kalit atributni qismiga bog'langan bo'lsa, u qisman bog'lanishli deyiladi. Bizning misolda soatlar soni (kalit bo'lmagan atribut) predmetlar nomi atributiga qisman bog'langan. Agar kalit bo'lmagan atribut barcha murakkab kalitga bog'langan bo'lsa, va uni qismiga bog'langan bo'lmasa, unda bu atributni murakkab kalitga to'la funksional bog'lanish deyiladi. Agar, A,V,S atributlar berilgan bo'lsa va unda $A \rightarrow V$ bo'lsa, $V \rightarrow S$ bo'lsa, unda S A dan tranzitiv bog'langan bo'ladi. Bizni misolda familiya, kafedra, telefon.

Uchinchi normal forma (3nf). Ma'lumotlar munosabatlarda 2nf ga

keltirilganda ham bir qancha noqulayliklar bo'лади. Jumladan, ma'lumotlarda informatsiyani ortiqchaligi, amallarni bajarish qiyinligi va boshqalar. Bunday munosabatlarni 3nf ga keltiriladi. 3nf da tranzitiv bog'lanish yo'qotiladi.

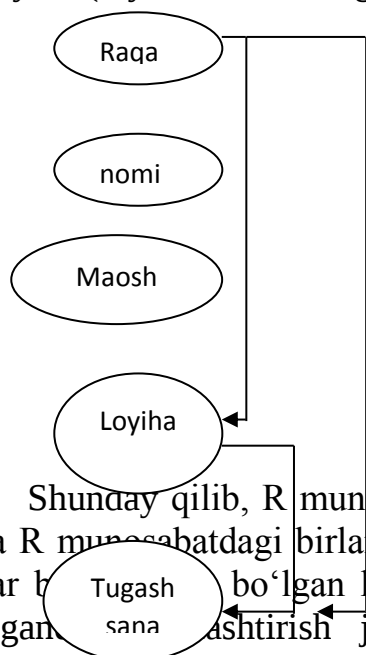
Agar, A,V,S R munosabatini 3 ta atributi yoki atributlar to'plami bo'lsin. Agar V atribut A atributga, S atribut esa V atributga bog'langan bo'lsa, ya'ni , $A \rightarrow V$ va $V \rightarrow S$ Bunda teskari bog'lanishlar bo'lmasa, unda S atribut A atributga tranzitiv bog'langan deyiladi. Uni ko'pincha diagramma ko'rinishida quyidagicha belgilaymiz:



3nf bu sxemadan o'tishi uni 2 ta munosabatga ajratish bilan bajariladi, ya'ni, Masalan, xizmatchi (raqam, nomi, maosh, loyiha_nomeri, tugash sanasi).

Xizmatchi (hiz-nomeri, hiz nomi, maosh, loyixa nomeri)

Loyiha (loyiha nomeri, tugash sanasi).

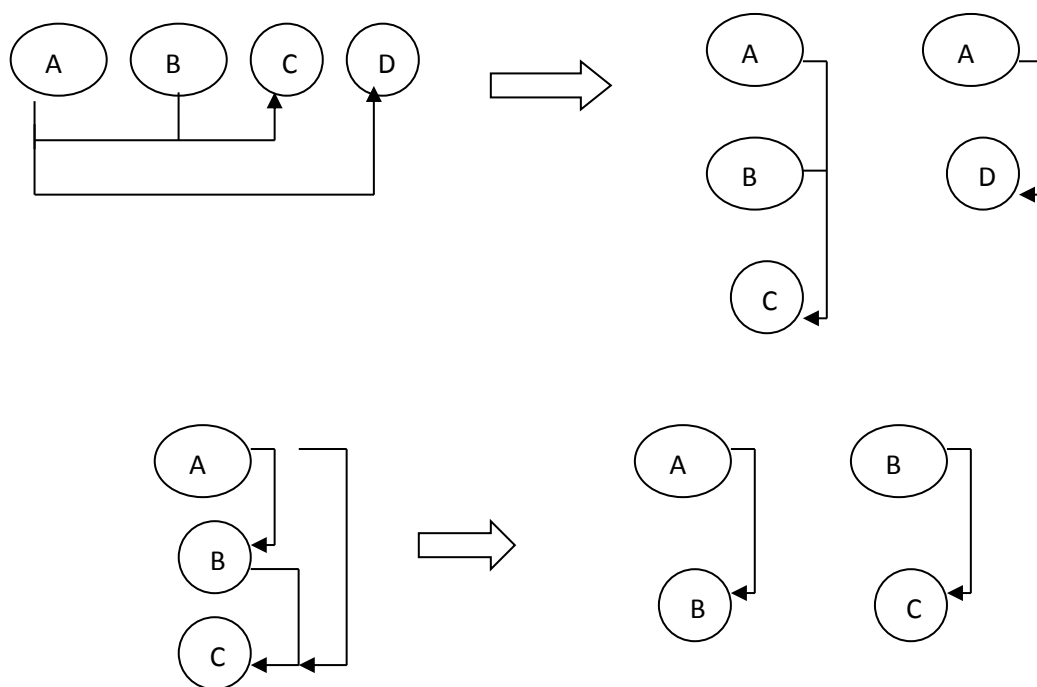


Xizmatchi (xiz raqam, xiz nomi, maosh, loyixa raqami).

Loyiha (loyiha, tugash sanasi).

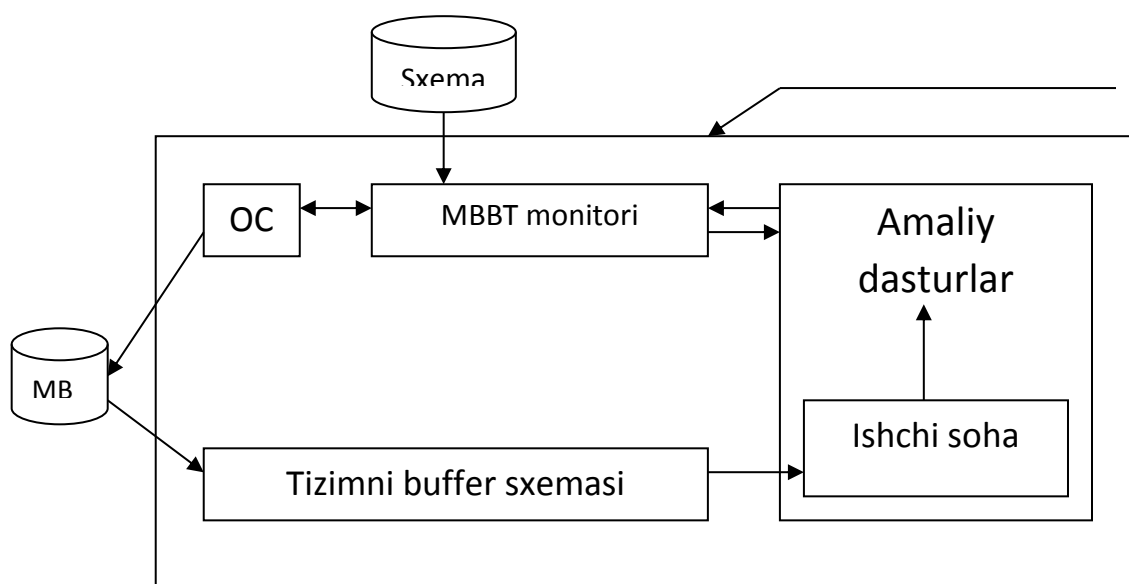
Shunday qilib, R munosabat 3nf da berilgan deyiladi, anarda, u 2nf da bo'lsa va R munosabatdagi birlamchi kalit bo'lmagan har bir atribut R munosabatni har har b bo'lgan kalit atributiga notranzitiv bog'langan bo'lsa. Umuman olgan ashtirish jarayoni va munosabatni 3nf ga keltirish quyidagi bosqichlardan iborat bo'лади:

1. Ma'lumotlarni ixtiyoriy strukturasidan oddiy strukturali 2 o'lchamli jadvallarga o'tish va 1nf ni hosil qilish.
2. Kalit atributlari bilan barcha atributlar orasidagi mumkin bo'lgan to'liqmas funksional bog'lanishlarni yo'qotish va 2nf hosil qilish.
3. Mumkin bo'lmagan kalit atributlari va asosiy bo'lmagan (nokalit) atributlar orasidagi tranzitiv funksional bog'lanishlarni yo'qotish va 3nf ni hosil qilish



Ma'lumotlar bazasi va MBBT ni fizik tashkil etish.

MBBT komponentalari va ularni OS va amaliy programmalar bilan o'zaro bog'liqligi ma'lumotlarni fizik tasvirlashda muhim o'rin to'rtadi. MBBT murakkab til programm kompleksidan iborat bo'lib, MB ni ishlash imkoniyatini ta'minlaydi. MBBT tarkibiga sistemali prorammlar kompleksi kiradi. Bu kompleksni markaziy komponentasi manitor yoki boshqaruvchi programma hisoblanadi. Manitor MBBT ni komponentalarini OS va amaliy programmalar bilan o'zaro ta'sirini tashkil qiladi. Bu komponentalarning fizik tashkil etuvchilari quyidagi chizmada berilgan:



Bu chizmada nomerlangan strelkalar bilan amaliy programma tarkibidagi ma'lumotlar bilan ishlash tili (YAMD) ni bitta operatorini bajarishiga tegishli bo'lgan amallar ketma – ketligi ko'rsatilgan.

Masalan, bu MB dan ma'lumotlarni o'qish so'rovini operatori bo'lib hizmat qilsin. Unda nomerlangan strelkalar quyidagi ma'noga ega:

- 1) Amaliy programmalar MB ga (YAMD) operatori orqali murojaat qilsin. Uni manitor tomonidan tahlil qilinadi.
- 2) Talqin qilish jarayonida manitor oldindan translyasiya qilib qo'yilgan sxemani ishlatadi.
- 3) Bu so'rovga tegishli ma'lumotlar aniqlanib bo'lingandan keyin, manitor OS ga tashqi xotiraga murojaat qilishni amalga oshirish talabi bilan murojaat qiladi.
- 4) OS MB ga murojaatni bajaradi. Bu xuddi fayllarga murojaat qilish kabi oddiy bajariladi.
- 5) Talab qilingan ma'lumotlar tashqi xotiradan sistemani bufer sohasiga o'zatiladi.
- 6) Ma'lumotlar amaliy programmalarini ishchi sohasiga jo'natiladi.
- 7) Manitor amaliy programmaga so'rovni bajarish natijalarini xabarini beradi.
- 8) Amaliy programma MB dan olingan ma'lumotlar ustida kerakli amallarni bajaradi.

Adreslash usullari. Bitta mashina ko'rsatmasi yordamida o'qish mumkin bo'lgan bitlar guruxi fizik yozuvlar deb ataladi. Fizik yozuvlar mashina xotirasining yacheykalarida saqlanadi va mashina adreslari yordamida identifikatsiyalanadi. Programmlar mantiqiy yozuvlarni kalitlar yordamida aniqlaydi. Programma uchun zarur bo'lgan ma'lumotni mantiqiy yozuv kalitlari yordamida fizik yozuvlarni adreslarani aniqlaydi. Programma uchun zarur bo'lgan ma'lumotni mantiqiy yozuv kalitlari yordamida fizik yozuvlarni adreslarini aniqlaymiz. Kalit qiymatlari juda ko'p bo'lganligi uchun mashina adreslar bilan munosiblikni aniqlash uchun xilma – xil adreslash usulidan foydalanamiz. Kalit sifatida har bir yozuvda joylashgan piksellangan uzunlikdagi maydonlardan foydalanamiz. Ba'zi hollarda kalit sifatida bir nechta maydon olinadi va bunda ulangan kalitlar hosil qilinadi. Fayllardagi yozuvlarni bir qiymatli aniqlash uchun albatta yagona kalit mavjud bo'lishi kerak va bunday kalitlar birlamchi kalitlar deb ataladi.

Yozuvlarni adreslashning quysidagi usullari mavjud:

- 1) Fayllarni ketma – ket saqlash usuli. Har bir yozuvni kaliti tekshiriladi. Bunday usul ko'p vaqtni talab etadi.
- 2) Blokli qidirish. Agar yozuvlar kalit bo'yicha tartiblangan bo'lsa, fayllarni skanerlashda har bir yozuvni o'qib chiqish talab etilmaydi. Bunday xollada kerakli yozuvdarni topish uchun blokli qidirish usulidan foydalanamiz. bunda yozuvlar bloklarga guruxlanadi va har bir blok bir

martadan tekshiriladi, kerakli yozuv qidirib topilguncha.

- 3) Binar qidirish. Bunda soha o'rtasidagi yozuv topiladi va uning kaliti qidirish tartibi bilan solishtiriladi. So'ngra qidirish sohasi ikkiga ajratiladi va har bir yarmi alohida qidiriladi. Binar qidirish to'g'ridan – to'g'ri murojaat qurilmalarida ishlatib bo'lmaydi.

Nazorat savollari

1. Munosabotlarni normallashtirish.
2. Munosabotlar atributlarini qo'yiladigan talablar.
3. Funksional bog'lanish turlari.
4. 1NF va undagi anomaliyalar nimada?
5. 2NF va undagi anomaliyalar nimada?
6. 3NF va undagi anomaliyalar nimada?

8–MAVZU. SQLtili. SQL operatorlarini yozish.

REJA:

1. SQL tilining vazifalari
2. Interaktiv va qurilgan SQL
3. SQL ma'lumot toifalari
4. SQL tilining SELECT (tanlash) operatori va uning parametrlari

Tayanch iboralar: *SQL, SELECT, DISTINCT, FROM, WHERE, ORDER BY, GROUP BY, HAVING, UNION, SELECT,*

Ma'lumotlar bilan ishlash uchun mo'ljallangan MBBTichki tili ikki qismdan tashkil topgan: Ma'lumotlar aniqlash tili (DATA DEFINITION LANGUAGE va ma'lumotlar bilan manipulyasiya qilish tili (DATA MANIPULATION LANGUAGE)

DDL_tili ma'lumotlar baza sxemasini aniqlash uchun ishlatiladi.

DML_tili bazada saqlanayotgan ma'lumotlarni o'qish va tiklash uchun ishlatiladi.

Bu tillar ma'lumotlarni qism tillari yuqori darajali dasturlash tillari deyiladi, chunki ularni tarkibida barcha xisoblarni bajarish uchun zarur bo'lganda bo'ladigan til kostruksiyalari bo'lmaydi (shartli o'tish amallari, yokissikl operatori) Ma'lumot bazasi ishlaydigan ixtiyoriy til foydalanuvchiga quyidagi imkoniyatlarni berishi kerak:

- ma'lumot bazasi va jadvallarni tuzilishi va tafsifini yarata olish;
- ma'lumotlar bilan manipulyasiya qilishni asosiy amallarini jumladan jadvalga ma'lumotlarni kiritish,(qo'shish),ularni takomillashtirish, jadvaldan ma'lumotlarni olib tashlash;
- oddiy va murakkab so'rovlarni bajarish ;

SQL tilining vazifasi. Ma'lumotlar bazasi bilan ishlash tillari ko'rsatilgan masalalarni foydalanuvchi kim kuch sarf qilib echishi imkoniyatini berishi kerak.

Tilning komandalarini tuzilishi va sintaksisi etarli darajada sodda va o'rganishga oson bo'lishi kerak. Bundan tashqari u universal bo'lishi kerak. Ya'ni qandaydir standart talablariga javob berishi kerak. Bu esa uni komanda strukturasini va sintaksisini (yozilishini) bir qancha MBBTishlatishini imkonini beradi.

Bu talablarni barchasiga SQL javob beradi.

SQL-(STRUCTURE QUERY LANGUAGE)

(STRUKTURALASHGAN SO'ROVLAR TILI)

SQL ma'lumotlarni almashtirish tili bo'lib xisoblanadi, ya'ni u jadvallar bilan ishlash mo'ljallangan bo'lib, kiruvchi ma'lumotlar talab etilgan chiquvchi ko'rinishga ega xolga aylantirib beradi.

SQL tili ikkita asosiy komponentga ega:

-DDL tili (DATA DEFINITION LANGUAGE **ma'lumotlar aniqlash tili**) bu ma'lumotlar ba'zi strukturasini aniqlash va ma'lumotlarga murojaatni boshqarish uchun mo'ljallangan.

DML tili (DATA MANIPULATION LANGUAGE) ma'lumotlarni manipulyasiyalash tili. Ma'lumotlarni ajratish va tiklash uchun mo'ljallangan.

-bu til noprotsedura tili, shuning uchun unga qanday axborot olishi kerakligi ko'rsatiladi, lekin axborotni qanday olish emas. Boshqacha aytganda SQL tili ma'lumotlarga murojaat usulini ko'rsatishni talab etmaydi.

SQL tili operatorlarni erkin formatda yozishini ta'minlaydi. Buning ma'nosi, operatorlar elementlarini yozilishi ekrandan fiksirlangan joylarga bog'liq emas. Komanda strukturasini bir qancha kalit xizmatchi so'zlar bilan beriladi, masalan:

CREATE TABLE (sozdat tablitsi- jadval yaratish)

INSERT (vstavka-qo'yish)

SELECT (vybrat-ajratib olish)

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

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

SQL operator komponentlarini ko'pchiligi registrga bog'liq emas, ya'ni ixtiyoriy xar qanday katta va kichik xarflr ishlatishi mumkin.

Bularda bitta istisno bor. Bu istisno simvolli literallarga tegishli. Ulardalitera ma'lumotlar, ularga mos bo'lgan ma'lumotldar bazasidagi qiymatlar qanday saqlansa shunday yozilishi kerak. Masalan:agar ma'lumotlar bazasida familiyaning qiymati «SWITH»ko'rinishida bo'lsa, qidirish shartida «SWITH» simvol literal ko'rinishida berilsa, bunga tegishli yozuv hech kachon topilmaydi.

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

- operatordagi xar bir konstruksiya yangi satrdan boshlanishi kerak;
- xar bir konstruksiya boshlanishida tashlab ketladigan bo'sh pozitsiyalar, boshqa operator konstruksiyalari xam bo'lishi kerak;
- agar konstruksiya bir necha qismdan iborat bo'lsa, ularning xar biri qism yangi satrlardan bo'sh o'rinlarni oldingi konstruksiyaga nisbatan siljitib yoziladi.

Amaliyotda ma'lumot bazasi tuzilishini (asosan uni jadvallarini) aniqlash uchun **DDL** operatorlari ishlatadi, bu jadvallarni ma'lumotlar bilan to'ldirish uchun va ulardan axborotlarni so'rovlar yordamida ajratib olish uchun - **DML** operatorlari ishlatiladi.

Ma'lumotlar bilan manipulyasiyalash SQL tilini **DML** peratorlarini ko'llaymiz. **Interaktiv va qurilgan SQL.** SQL tilini ikkita shaklida mavjud va ishlatiladi.

- interaktiv SQL
- qurilgan (kiritilgan) SQL

Interaktiv SQLda foydalanuvchi SQL- so'rovlar va natija interaktiv rejimda olinadi.

Qurilgan SQL, SQLkomandalaridan tashkil topib, u boshqa birorta tilga (S++S, Delphi) yozilgan dastur ichiga joylashtiriladi. Bu shunday tillarni ishlatadigan dasturlarni samarador, quvvatli qiladi. Ularga relyasion ma'lumotlar bazasi bilan ishlash imkonini beradi.

SQL ma'lumot toifalari. Simvollar satr ma'lumot toifasi SQL standartida matnlarni faqat bitta tavsifi keltiriladi. Uning sintaksisi

**CHARACTER[(UZUNLIGI)] YOKI
CHAR[(UZUNLIGI)]**

Jadvalni matnli qiymatli **CHAR** toifasidagi fikrlangshan uzunlikda bo'lishi mumkin. Bu parametr qiymati 1 - 255 bo'lishi mumkin, ya'ni u 255 simvolgacha bo'lishi mumkin. SQL tilini ba'zi birlardagina o'zgaruvchan uzunlikdagi satr toifalari bor. Bu toifalar quyidagicha tavsiflanadi:

VARCHAR(),CHARVARYING YOKI CHARVARYING()

Ixtiyoriy uzunlikdagi matnli tasvirni tasvirlaydi.

CHARACTER va **VARCHAR** toifasidagi konstantalar apostrof ichiga yoziladi.

Quyidagiyozuvlarni barchasi ekvivalent

VARCHAR [(UZUNLIGI)],

CHARVARYING [(UZUNLIGI)]

CHARACTER VARYING[(UZUNLIGI)]

Agar uzunlik oshkor ko'rsatilmasa, u birga teng deb qabul qilinadi, ya'ni barcha xollarda bita simvoldan iborat bo'ladi.

Sonli ma'lumot toifalari SQL standartida quyidagi son toifasida ishlatiladi.

INTEGER-butun sonlar uchun $-2^{31} \dots 2^{31}$

SMOLLINT-butun sonlar $2^{-15} \dots 2^{15}$

DECIMAL (aniqlik[masshtab]) -Fiksirlangan nuqtali unli son aniqlik sondagi qiymatli raqamlar masshtab unli nuqtadan undagi raqamlarning maksimal soni kursatadi

NUMERIC(aniqlik[masshtab])- Fiksirlangan nuqtali unli son. aniqlik sondagi qiymatli raqamlar masshtab unli nuqtadan undagi raqamlarning maksimal soni kursatadi

FLOAT[(aniklik)]suzuvchi nuktali son minimal aniklik bilan

Beriladi. Shunday qilib xulosa qilish mumkin:

Simolli satrlar toifasi. Character [uzunligi] [char] uzun) da kursatiladi.

Bundan tashkari uzgaruvchan uzunlikdagi simvolli satrlar toifasini xam ishlatamiz. Bunda uzgaruvchi toifalar ixtiyoriy uzunlikda buladi. Bunda uzunliklar zarur bulmagan parametrlar xisoblashadi. Agar ular ishlatilmasa, unda 1 ta simvolga joy ajratiladi.

Simvolli satrlarni belgilashni yana bir usuli bor.

Varchar [(uzunlik)] yoki charvaryina [(uzunlik)]

1. Ma'lumotlarni sonli tiplari

1. Integer

2. Smollint

3. Decimal (aniklik, masshtab). (DEC) fiksirlangan (___) li sonlarni tasvirlash uchun ishlatiladi.

Aniklik- sondagi kiymatli rakamlar.

Masshtab- nuqtadan keyingi ung tomonda turgan rakamlarni maksimal soni

4. Numeric (aniklik, (masshtab)..

5. FLOAT (aniklik). Suzunlikuvchi nuktali son va undagi minimal aniklikni bildiradi.

6. REAL FLOAT kabi tip sonlari bilan kushimcha kilingan.

Sana va vakt toifasidagi ma'lumotlar standarti kushimcha kilinmagan. Bular yozilishini texnik xujjatlarda kurish kerak.

NOANIQ VA O'TKAZIB YUBORILGAN MA'LUMOTLAR

SQL da atribut kiymatlari noma'lum bulgan utkazibi yuborilgan yoki mavjud bulmaganlarini NULL bilan yoziladi. NULL kiymat oddiy tushunchada kiymat kiymat xisoblanmaydi. U fakat atributni xakikiy kiymati tushib koldirilgan yoki noma'lumligini anglatadi. NULL ni ishlatishda kuyidagilarga e'tibor berish kerak:

Agregat f-yalar ishlatilganda birorta atributni kiymatlar tuplami buyicha xisoblashlar bajarilganda aniklikni ta'minlash maksadida NULL kiymat xisobga olinmaydi.

- shartli operatorlarda TRUE, FALSE dan tashkari UNKOWN paydo bulsa natija NULL kiymatda chikadi.

- bu kiymatni tekshirish uchun IS NULL yoki IS NOT NULL lardan foydalanamiz.

- almashtirish fuknsiyalari xam argument sifatida NULL bulsa natija NULL ga teng buladi.

Misollar:

O'quv misoli: Echilaetgan misollar kuyidagi jadvalari asosida bajarilgan;

MB jadvallari:

STUDENT (TALABA)

Stud_ID	Surname	Name	Stipend	Kurs	City	Birthday	Univ_ID
1	Ivanov	Ivan	150	1	Orel	3.12.92	10
3	Petrov	Petr	200	3	Kursk	1.12.90	10
6	Sidirov	Vadim	150	4	Moskva	7.06.89	28
10	Kuzunliknetsov	Boris	0	2	Brenek	08.12.91	10
12	Zayseva	Olga	25	2	Lipetsk	1.05.91	10
265	Pavlov	Andrey	0	3	Voronej	5.11.89	10
32	Kotov	Pavel	150	5	Belgorod	NULL	14
654	Muxin	Artem	200	3	Voronej	1.12.91	10
276	Petrov	Anton	20	4	NULL	5.08.91	22
55	Belkin	Vadim	250	5	Voronej	7.01.90	10

LECTURER (O'qituvchi)

Lecturer_ID	Surname	Name	City	Univ_ID
24	Komsmakov	Boris	Voronej	10
46	Nikonov	Ivan	Voronej	10
74	Lagutin	Pavel	Moskva	22
108	Otrukov	Nikolay	Moskva	22
276	Nikolaev	Viktor	Voronej	10
378	Sorokin	Andrey	Orel	10

SUBJECT (O'qitiladigan fanlar)

Subj_ID	Subj_name	Hour	Semester
10	Informatika	56	1
22	Fizika	34	1
43	Matematika	56	2
56	Tarix	34	4
94	Ingliz tili	56	3
73	j/tarbiya	34	5

University(Universitet)

Univ_ID	Univ_name	Rating	City
22	MGU	606	Moskva
10	BGU	296	Voronej
11	HGU	345	Novosibirsk
32	PGU	416	Rostov
14	BGU	326	Belgorodok
15	TGU	368	Tomsk
18	BGMA	327	Voronej

EXAM_MARKS

Exam_ID	Student_ID	Subj_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

SUBY_LECT (ukituvchilarni ukuv fanlari)

Lecturer_ID	Subj-Id
24	10
46	22
74	43
108	56
276	94
328	73

SELECT operatori

SELECT (tanlash) SQL tilining eng muxim va kup ishlatiladigan operatori xisoblanadi. U ma'lumotlar bazasi jadvalidan axborotlarni tanlab olish uchun muljallangan.

SELECT operatori sodda xolda kuyidagi kurinishda yoziladi.

SELECT [DISTINCT] <atributlar ruyxati>

FROM <jadvallar ruyxati>

[WHERE <tanlash sharti>]

[ORDER BY < atributlar ruyxati >]

[GROUP BY < atributlar ruyxati >]

[HAVING <shart>]

[UNION <ON SELECT operatorli ifoda>] ;

Bu erda kvadrat kavslarda yozilgan elementlar xar doim xam yozilishi shart emas

SELECT xizmatchi suzunliki ma'lumot bazasidan axborotni tanlab olish operatori yozilganini anglatadi. **SELECT** suzunlikidan keyin bir biridan vergul bilan ajratilib suralaetgan maydon nomlari (atributlar ruyxati) yoziladi. **SELECT** surov operatorini zarur xizmatchi suzunliki **FROM** (undan, dan) xisoblanadi. **FROM** suzunlikidan keyin axborot olinayotgan jadval nomlari bir biridan vergul bilan ajratilib yoziladi.

Masalan:

SELECT Name, Surname

FROM STUDENT ;

Ixtiyoriy SQL surov operatori nukta vergul (;) simvoli bilan tugaydi.

Keltirilgan surov STUDENT jadvalidan Name va Surname maydonlarni barcha kiymatlarini ajratib olishni amalga oshiradi. Natijada kuyidagi jadval xosil buladi. O'quv misoli:

Name	Surname
Ivan	Ivanov
Petr	Petrov
Vadim	Sidirov

Boris	Kuzunliknetsov
Olga	Zayseva
Andrey	Pavlov
Pavel	Kotov
Artem	Muxin
Anton	Petrov
Vadim	Belkin

Nazorat savollari

1. SQL tilining 2 ta asosiy komponentasi va ularning bajaradigan funktsiyalari.
2. **SELECT** operatori asosiy konstruksiyalari va ularga qo'yiladigan cheklanishlar.
3. **SELECT** da guruhli (agregat funksiyalarni) ishlatish.
4. **GROUP BY PARAMETRI ISHLATISH**
5. **WHERE** va **HAVING** konstruksiyalarni farqi.
6. Qism so'rov bilan so'rovlarni birlashtirishni farqi.

9-mavzu. Ma'lumotlar manipulyatsiya qilish. Oddiy so'rovlar yaratish.

Reja:

1. Murakkab surovlar yaratish (tashkil qilish)
2. Guruxli funksiyalarni so'rovlarda ishlatish.
3. Tasavurlar.

Tayanch iboralar: Guruxli funksiyalar, so'rovlar, manipulyatsiya, view, tasavur, tranzatsiya.

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

Select Surname, Name, from STUDENT Where Surname='Petrov';

Surname	Name
Petrov	Petr
Petrov	Anton

Where shartida solishtirish amallari jumladan, =, <, <=, >=, ≠ shuningdek, mantiqiy amallar «and» “ot”, “not” amallari ishlashi mumkin. Ular yordamida murakkab shartlar tuzunlikiladi.

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

Select name, Surname From STUDENT Where kurs=3 and stipend>0;

Mantiqiy shartlarni berishda where parametri tarkibida IN, BETWEEN, Like, is null amallari xam ishlatiladi.

In, not parametli ishlatilganda tekshirilayotgan maydon kiymati berilgan ruyxat bilan solishtiriladi. Bu ruyxat in operatori ung tomonidani () ichida yoziladi.

1. Exam baxolari jadvaldan «4» va «5» baxoli talabalar ruyxatini chiqaring.

Select *

From exam-marks

Where mark in (4,5);

Birorta ham 4,5 olmagan talaba haqidagi ma'lumotlar olish uchun not it yoziladi.

Between amali maydon kiymatini berilgan intervalga kirganligini tekshirish uchun ishlatiladi.

2. 30 va 40 soat doirasida ukitiladigan fanlarni chiqarish so'rovini tuzunliking.

Select*; from subject

Where hour between 30 and 40;

Between amali maydonla sonli va simvolli bluganda xam ishlatiladi.

Like amali f-t simvolli toifadagi maydonlar uchun ishlatiladi. Bu amal maydonni satri matnlarni like dan sung kursatilgan kism satr bilan solishtiradi.

3. studentlar jadvalida familiyasi «M» xarfi bilan boshlanadigan talaba haqida ma'lumot chikaring.

Select* from student

Where surname like P%;

%- foiz belgisi shu pozitsiyada ixtiyoriy simvollar ketma-ketligi kelishini anglatadi. Bundan tashkari «__» shi ular ma'nosi, shu pozitsiyada ixtiyoriy 1 ta simvol kelishini anglatadi.

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

SQL tilida ma'lumotlarni almashtiruchi va kiritilgan familiyalar ishlatilishi mumkin. Ular ustun qiymatlari bilan ishlashi uchun yoki const sifati ifodalarda keladi. const sifatida simvolli const, sonli constlarni ishlatish mumkin. Ular ustunlar ro'yxatiga kiritiladi va xudi 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 chiqaradi.

Select 'familiya', surname, 'imya', 'name', 100

From STUDENT

	Surname		Name	
Familiya	Ivanov	Imya	Ivan	100
Familiya	Petrov	Imya	Petr	100

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

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

From student

Where kurs ning 4 and stipend >0

Natija

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

Satrlarni ulash amali yordami const 2 ta va undan ko`p simvulli ustun qiymatlari bitta satrga joylashtirib boriladi.

Select surname //'-'// name, stipend

Weher kurs ning and stipend >0;

	Stipend
Sidirov_ Vadim	150
Petrov_ Anton	200

SQL da almashtirish funksiyalari bilan ishlash

1. Lower (<satr>)- berilgan satrni kichik xarflarga almashtirib beradi.
2. Upper- (<satr>) – kichik xarflarni kata xarflarga almashtirib beradi.
3. Init cap- (<satr>)- satrdagi xar bir suzunlikni 1-xarfini bosh xarf kilib beradi. Maslan, ularga kuyidagi misolni kuramiz.

Select lower (surname)

Upper (name) from student

Where kurs=4 and stipend 20;

Surname	Name
Sidirov	Vadim
Petrov	Anton

4. **LPAD** (<satr>, uzunlikunlik, [< kism satr>]) **RPAD** (<satr>, <uzunlikunlik>, [<kism satr>]);

Berilgan uzunlikunlikdagi kism satrni chapdan, unгдаy joylashtiriladi.

Agar kism satr kursatilmagan bulsa, satr sukut bilan, probellar bilan tuldiriladi. Agar uzunlikunlik satr uzunlikunlikdan kerak bulsa berilgan satr kursatilgan uzunlikunlikgacha kirkiladi.

5. **LTRIM** (<satr>), [<kism satr>]);

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

Bu funksiyalarni vazifasi mos ravishda chapdagi (ungday) chegaraviy simvol olib tashlashdan iborat. Olib tashlangan simollar kism satrda kursatiladi. Agar kism satr ishlamasa, probellari olib tashlanadi.

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

6. Kuyidagi bu funksiyalari satrdan berilgan pozitsiyadan boshlab berilgan sondagi simvollari ajratib olinadi. Agar soni kursatilmagan bulsa satrni boshidan oxirigacha ajratib olinadi.

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

7. **Length** (<satr>) vazifasi satrni uzunlikunligini aniklab borishdan iborat
Select l pad (Surname, 10, D)
LPad (Name, 10,8), from STUDENT
Where kurs=3 and stipend>0

αααα Petrov	Petr \$\$\$\$\$\$
αααα Pavlov	Andrey \$\$\$\$
αααααα Lukin	Artem \$\$\$\$\$

8. **Select substr** (name, 1,1) //' '// Surname;
City length (City) from STUDENT
Where krus in (2,3,4) and stipend>0;

	City	
A.Petrov	Kursk	5
S.Sidorov	Moskva	6
....		

Misol:

Select Surname, Name, Brithday;
Tochar (birthday, DD MM, YY)
From STUDENT

Surname	Name	Birthday	
Ivanov	Ivan	3/11/992	3.12.92

Guruxli (agregat) funksiyalar.

Guruxli funksiyalar jadvaldan yig`ilgan axborotlarni olish uchun xizmat qiladi. Bu funksiyalari jadvaldagi satrlar guruxi bilan amal bajarib, 1 ta natija chiqaradi. Guruxli funksiyalar uchun quyidagi amallarni ishlatamiz.

1. Count- jadvaldagi satrlar sonini aniklab beradi.
2. Sum- kursatilgan maydo kiymatlarini yigindisini xisoblaydi.
3. AVG-tanlab olingan maydon kiymatlarini urta arifmetigini xisoblaydi.
4. MAX (min)-tanlab olingan maydon kiymatlarini eng kattasini (kichigini) topib beradi.

Select surovida guruxli funksiyalar maydon nomlari kabi ishlatiladi. Maydon nomlari funksiyalar argumentlari sifatida keladi.

Misol:

1. Exam_marks jadvaldagi select averad (mark)
2. Jadvaldagi satrlar (yoki yozuvlar) sonini xisoblash uchun quyidagi komandadan foydalanamiz.

Select count (*)

From EXAMS_MARKS

3. Select count (**distinct** subj_ID); from SUBJECT ;

Select komandasida group by parametr xam ishlashi. Bu paramet bir maydon uxshash parametrlari (aniklanayotgan kiymati) buyicha guruxlaydi va agregat funksiyalar ishlatilsi, ular shu guruxga buladi.

Misol: **Select** student_ID

Max (mark) from exam_marks

Group by student_ID

Guruxlashni bir nechta maydon buyicha xam bajarish mumkin.

Select student_ID, subject_ID

Max (mark)

From exam-marks group by

Student_ID, subject_ID

Guruxlar ichidan kerakli yozuvlarni ajratib olish uchun having ishlatiladi.

Having=where, f-t having guruxlar ichiga tegishli

Select Subj_name, max (hour)

From SUBJECT;

Group by Subj_name

Having max (Hour)>= 34;

Ba'zi xollarda natija jadvalidagi ma'lumotlarni tartiblab talab etiladi. Buning uchun Order by parametri ishlashi. Bu parametri kursatilgan maydon barcha yozuvlarni usib borishi tartibida tartiblab beradi. Order by desc yozilsa kamayishi tartibida yoziladi. Order by (ASC) bulsa usish tartibida yoziladi.

Misol:

1. **Select*** from Subject

Order by Subj_name

2. **Select***from Subject

Order by Subj-name desc

Tartiblash bir nechta maydon barcha bajarilishi xam mumkin. Bunda avval tartiblash 1-maydon buyicha keyin 2-maydon buyicha bajariladi.

SHuningdek order by parametri group by parametri bilan birga ishlatilishi mumkin. Bunda oreder by surovda oxiri keladi va unda guruxni ichidagi yozuvlar tartiblaydi.

Misol:

1. **Select * from** SUBJECT

Order by Semester, Subj-name;

2. **Select** subj-name

Semester, subj-name

Order by semester

SQL tili 1 ta surov ichiga 2-surovni joylashtirib ishlatish imkonini beradi. Misol: birorta talabani familiyasi buyicha uning ID sini topish talab etilsa, va bu talabani barcha baxolari xakidagi ma'lumotni kurmokchi bulsak kuyidagi surovni yozish mumkin.

Select*; from exam_marks

Where student_ID (select student_I)

From student where surname= 'Petrov'

Jadvallar bilan ishlaganda, ba'zan ustun va jadval nomlarini kayta aniklashga yoki kayta nomlashga tugri keladi. Bunday masalalar kupincha birorta ifodalarni xisoblaganda, virtual ustunga joylashganda unga nom kuyish yoki ba'zan natija jadvali ustunini nomlashda kerak buladi.

Faculty

Name	Stipend	Select name AS
Ivanov	150	Name_ talaba,
Petrov	200	2* stipend AS yangi St

Faculty

Name	Stipend	Select name
		Name_ talaba,
		2 * stipend yangi stip

Xuddi shuningdek biz ustun nomlarini xam uzunligartirishimiz mumkin.

EXITS operatori

SQL da ishlatiladigan EXITS operatori mantikiy ifoda kabi rost va yolg'on kiymatlar k-k. bu operator argument sifatida kism surovlarni ishlatadi. Agar kism surov birorta kiymati rost, aks xolda yolg'on bulishi. Misol. Imtixonlar jadvalidan xech bulmaganda talaba xakidagi ma'lumotni olgan talaba xakidagshi ma'lumotni chikarish uchun kuyidagi surov yoziladi.

Select distinct student_ID

From ecam_marks A where

Exists (select * from EXAM-MARKS)

B where mark<3 and B

Student_ID= A student_ID

Birlashtirish operatori- UNION , operator 2 yoki unidan ortik SQL surovlar chikaradigan natijalarni yagona satr va ustunlar tuplamiga birlashtiradi.

Misol: 1 ta jadvalda Moskva shaxridan bulgan talaba va ukituvchilarni familiyalari va ID kodlarini joylashtirish uchun kuyidagi SQL surovni yozamiz.

Select 'TALABA____', Surname,

Student_ID, from STUDENT

Where city= 'Moskva'

union

Select "o'qituvchi____", Surname,

Lecturer_ID

From LECTURER where city= 'Moskva';

Takrorlanuvchi satrlar paydo bulishi uchun «UNION ALL» deb yoziladi.

Jadvallarni INNER JOIN operator bilan birlashtirish.

Agar select operatorida from suzunlik keyin 1 ta jadval emas 2 ta jadval ishlatilsa, bunda surovni natija tashkil kiluvchi jadval 1 jadvali xar bir satri bilan 2-jadvalni xar bir satriga ulashdan (kombinatsiyadan) yaratiladi. Bu amal jadvallarni birlashtirish deyiladi.

Misol: St. jadvaldagi talabalar fan va xar bir talaba uchun shu talabalar yashaydigan universitet nomini (UN jadvaldan) chikarish kerak bulsa unda buni kuyidagicha bajarimiz.

```
Select STUDENT.Surname, UNIVERSITY.Univ_name, STUDENT.City  
From STUDENT INNER JOIN UNIVERSITY on STUDENT.City =  
UNIVERSITY.City;
```

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

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

Tasavurlar xam vaktinchalik jadvallar bo'lib, ular kup foydalanuvchilvr murojat qilishi mumkin va u ma'lumot bazasidan majburan olib tashlanguncha mavjud bo'ladi.

Tasavurlar MB oddiy jadvallariga o'xshash bo'lib, ma'lumotlar saqlovchi fizik ob'ekt xisoblanmaydi. Tasavurlarda ma'lumotlar jadvallardan tanlab olinadi

Tasavurlar foydalanuvchilardan jadvallarni ba'zi ustunlarini yashirish uchun yoki ko'pincha foydalanuvchiga kerak bo'lgan bir nechta jadvaldan bitta yaratish kerak bo'ladi. Misol sifatida 3ta jadvaldan tashkil topgan oddiy ma'lumot bazasini qarab chiqamiz.

Tovarlar (ID -tovar , nomi, narxi, tavsifi)

Mijozlar(ID - mijoz, ismi, manzili, telefon)

Sotish(ID- tovar,soni, mijoz)

Tashkil qilish nuqtai nazaridan bu ma'lumot bazasi yomon loyixalanmagan. Lekin ba'zi masalarni echishda foydalanuvchini(menedjerni) mijoz va tovar identifikatorlari qiziqitirmaydi. Aniqrog'ini aytganada, unga bitta jadval kerak bo'ladi. Masalan bu jadval SOTISH_TAXLILI (tovar, soni,bag'osi, narxi, mijoz). Bu jadvalni berilgan uchta jadvaldan quyidagi so'rov yordamida hosil qilish mumkin.

```
SELECT Tovarlar.Nomi AS Tovar, Sotish.Soni*Tovarlar.Bahosi AS Narxi,  
Mijoz.Ismi || ' .Manzil: ' || Mijoz.Manzil || ' . tel. ' || Mijoz.Telefon AS Mijoz  
FROM Sotish, Tovarlar, Mijozlar WHERE Sotish. ID – mijoz= Mijozlar.ID –  
mijoz AND Sotish.ID- tovar = Tovarlar. ID –tovar;
```

Ko'rib chiqilgan so'rov uchta jadvalni birlashtirishidan iborat bo'lib, ularga narx va mijoz ustunlari qo'shilgandir.Buning natijasida quyidagi jadval xosil bo'ladi.

Agar bu jadval **SELECT** operatorini natijasi emas , tasavur bo'lganda edi , unga oddiy ma'lumot bazasini oddiy jadvali kabi murojat qilinardi. Ko'p xollarda esa MB uchta jadvaldan iborat ekanligini xisobga olmay , bitta tasavur bilan ishlanardi.

Tasavurlar yaratish uchun **CREATE VIEW** komandasi ishlatiladi
Uni formati quyidagicha:

```
CREATE VIEW «tasavur nomi» AS « select so'rovi » ;
```

Tasavurlarga xam ma'lumot baza jadvallari kabi nom beriladi. Bu nom birorta xam jadval nomi bilan bir xil bo'lmasligi kerak. AS so'zidan keyin ma'lumotlar

tanlashga uchun so'rov iborasi yoziladi.

Товары : таблица				
ID_товара	Наименование	Цена	Описание	
1	Хлеб	25,50	Батон	
2	Молоко	15,00	Луговое, пакет 1л	
3	Пиво	24,00	Балтика 7, банка 0.5л	
4	Мясо	170,00	Говядина, 1 кг	
Запись: 1 из 4				
Клиенты : таблица				
ID_клиента	Имя	Телефон	Адрес	
1	Иванов	111-1111	Санкт-Петербург	
2	Петров	111-2222	Москва	
3	Сидоров	222-3333	Санкт-Петербург	
4	Федоров	444-4444	Миргород	
Запись: 1 из 4				
Продажи : таблица				
ID_товара	Количество	ID_клиента		
1	20	2		
2	150	2		
2	50	1		
1	200	3		
3	3	1		
2	40	1		
4	2	4		
Запись: 1 из 7				

Ma'lumot bazasini berilgan jadvallari

Обзор продаж : запрос на выборку					
Товар	Количество	Цена	Стоимость	Клиент	
Пиво	3	24,00	72	Иванов. Адрес: Санкт-Петербург. тел.111-1111	
Молоко	50	15,00	750	Иванов. Адрес: Санкт-Петербург. тел.111-1111	
Молоко	40	15,00	600	Иванов. Адрес: Санкт-Петербург. тел.111-1111	
Молоко	150	15,00	2250	Петров. Адрес: Москва. тел.111-2222	
Хлеб	20	25,50	510	Петров. Адрес: Москва. тел.111-2222	
Хлеб	200	25,50	5100	Сидоров. Адрес: Санкт-Петербург. тел.222-3333	
Мясо	2	170,00	340	Федоров. Адрес: Миргород. тел.444-4444	
Запись: 1 из 7					

Uchta so'rovni birlashtirish natijasi

Masalan:

YUqorida berilgan jadvallar uchun tasavur yaratish uchun quyidagi komanda yoziladi.

```
CREATE VIEW sotish_taxlili AS SELECT Tovarlar.Nomi AS Tovar,  
Sotish.Soni*Tovarlar.Bahosi AS Narxi, Mijoz.Ismi || ' .Manzil: ' || Mijoz.Manzil  
|| ' . tel. ' || Mijoz.Telefon AS Mijoz FROM Sotish, Tovarlar, Mijozlar WHERE  
Sotish. ID – mijoz= Mijozlar.ID – mijoz AND Sotish.ID- tovar = Tovarlar. ID –  
tovar;
```

Natijada sotish taxlili nomli virtual jadval yaratiladi. Unga surovlar yordamida murojat qilish mumkin:

Masalan:

```
Select * from sotish-taxlili where tovar = 'moloko ' ;
```

Tasavurlar MB jadvallar olib tashlangan kabi olib tashlanadi: Masalan

```
DROP VIEW «tasavur nomi»;
```

```
DROP VIEW Sotish_taxlili;
```

Nazorat savollari

1. Shartli so'rovlar qanday tashkil qilinadi?
2. Guruhli funksiya vazifalari?
3. Guruhli funksiya kurinshlari?
4. Tranzaksiya nima?

10–mavzu. SQL tili. Ma'lumotlar tavsiflash.

REJA:

1. SQL tilida ma'lumotlarni butunligini ta'minlash.
2. Ma'lumot jadvallarini yaratish.
3. Qism so'rovlar.

Tayanch iboralar: CREATE TABLE, Char(character, Int(integer), Smallint, Dec(decimal), Number, Float,

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

ko'rinishi quyidagicha:

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

CREATE TABLE xususiyati quyidagicha:

SQL ishlatiladigan ma'lumot toifalari ANSI standartiga berilgan..

Char(character)

Int(integer);

Smallint,

Dec(decimal),

Number,

Float,

.....

Albatta ko'rsatilishi zarur bo'lgan ma'lumot toifasi -CHAR . Maydonga
ezilgan real simvollar soni noldan (agar maydonda NULL qiymati bo'lsa)

CREATE TABLE da 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),

BIRTHDAY DATE,

UNIV_ID INTEGER);

Jadvaldagi ma'lumotlarni maydonlar bo'yicha qidirish- tanlash amaliy etarli
darajada tezlatish uchun ma'lumotlarni berilgan maydon bo'yicha indeksatsiya
qilish ishlatiladi. Indeksni bitta ekanligi bir nechta maydon bo'yicha bajarish mumkin.
Indeks komandasini ko'rinishi:

CREATE INDEX < indeks 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 talabani STUDENT_ID maydoni
qiymati bo'yicha baxosini 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 uni nomini albatta bilish kerak) quyidagi
komanda ishlatiladi.

DROP INDEX < INDEX NOMI>;

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

Mavjud jadval strukturasi va parametrlar uchun **ALTER TABLE** komandasi
ishlatiladi. Masalan jadvalga ustunlar qo'shish **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 eziladi. Jadvalga bir nechta ustun xam ko'shsa bo'ladi. Ular bir biridan vergul bilan ajratiladi.

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

Ustun xarakteristikalarini modifikatsiyalashda quyidagi cheklanishlarni xisobga olish kerak :

- Ma'lumot toifasini o'zgartirishni, faqat ustun bo'sh bo'lsa bajarish mumkin
- To'ldirilmagan ustun uchun o'lcham/ aniqlik uzunlikgartirish mumkin.
- To'ldirilgan ustun uchun o'lcham/ aniqlik faqat kattalashtirish mumkin.
- NOTNULL urnatilishi uchun ustunda birorta xam NULL qiymat bulmasligi kerak.
- Sukut bilan o'rnatilgan qiymatni xar doim uzunlikgartirish mumkin

Jadvallarni olib tashlash quyidagi komanda bilan bajariladi

DROP TABLE <jadval nomi>;

Mumkin bo'lgan ma'lumot qiymatlar cheklanishlar bo'lishi mumkin. Unda

CREATE TABLE komandasi 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 , **NULL** qiymatni **STUDENT** jadvalini aniqlashda jadvalidagi **STUDENT_ID** ,**SURNAME** ,**NAME** , maydonlarida ishlatishni 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** ,

KURS **INTEGER** ,

CITY **CHAR**(15) ,

BIRTHDAY **DATE** ,

UNIV_ID **INTEGER**);

Ba'zi xollarda birir maydonga kiritilaetgan barcha qiymatlar bir biridan farq qilishi kerak. Bunda shu maydon uchun **UNIQUE (yagona)** so'z ishlatiladi.

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** ,

NAME **CHAR**(10) **NOT NULL** ,

STIPEND **INTEGER** ,

KURS **INTEGER** ,

CITY CHAR(15),

BIRTHDAY DATE;

Jadvalda kalit maydonlarni ishlatish komandasi quyidagicha bo'ladi

CREATE TABLE STUDENT

(STUDENT_ID INTEGER PRIMARY KEY ,

SURNAME CHAR (25) NOT NULL,

NAME CHAR(10) NOT NULL ,

STIPEND INTEGER,

KURS INTEGER,

CITY CHAR(15),

BIRTHDAY DATE,

UNIV_ID INTEGER);

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

INSERT komandasi (buyruqi) jadvalga yangi satr qo'shishni amalga oshiradi. Sodda holda uning ko'rinishi quyidagicha:

Insert into <jadval nomi> values (<qiymat>,<qiymat>);

Bunday yozuvda VALUES kalit so'zidan keyin qavs ichida ko'rsatilgan qiymatlar jadvaldagi yangi qo'shilgan satrning maydonlariga kiritiladi. Kiritish jadvalini CREATE TABLE operatori bilan yaratilish paytidagi ustunlarni ko'rsatish tartibida amalga oshiriladi. Masalan, STUDENT jadvalida yangi satrni qo'shish quyidagicha amalga oshirish mumkin.

Insert into Student

Values (101, 'Ivanov', 'Aleksandr', 200,3, 'Moskva', '6/10/1979' , 15);

Agar birorta maydonga NULL qiymati qo'shish zarur bo'lsa u oddiy qiymat kabi kiritiladi.

Insert into Student

Values (101, 'Ivanov', Null, 200,3, 'Moskva', '6/10/1979' , 15);

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

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

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

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

INSERT komandasi 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** komandasi yordamida yaratilgan bo'lishi kerak va STUDENT jadvali strukturaga o'xshash bo'lishi kerak. Jadvaldagi satrlarni olib tashlash uchun **DELETE** komandasi ishlatiladi.

Quyidagi ifoda EXAM_MARKS1 jadvalidan barcha satrlarni olib tashlaydi.
DELETE FROM EXAM_MARKS1;

Buning natijasida jadval bo'sh bo'lib qoladi(bundan so'ng uni **DROP TABLE** komandasi bilan 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 komandasi jadval satrlari yoki mavjud satrni ba'zi bir yoki barcha maydonlarini qiymatini tiklash, ya'ni o'zgartirish imkonini beradi. Masalan: UNIVERSITET1 jadvalidagi, barcha universitetlarini reytingini 200 qiymatga o'zgartirish uchun quyidagi konstruksiyani ishlatish mumkin:

UPDATE UNIVERSITY1

SET RATING=200;

Jadvaldagi maydon qiymatlarini o'zgartirish kerak bo'lgan aniq satrlarni ko'rsatish uchun, **UPDATE** komandasi **WHERE** parametrida predikat ishlatish mumkin.

UPDATE UNIVERSITY1

SET RATING=200

WHERE CITY= 'Moskva';

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

UPDATE komandasi faqat 1ta ustun emas balki ustunlar to'plamini o'zgartirish imkonini beradi. Qiymatlari, modifikatsiya (takomillashtirilgan) qilinishi zarur bo'lgan aniq ustunlarni ko'rsatish uchun, **SET** parametri ishlatildi. Masalan: o'qitilayotgan fan nomi "MATEMATIKA" (uning uchun SUBJ_ID=43) "Oliy matematika" nomiga o'zgartirish talab etilsa va bunday indetifikatsion nomeri saqlab o'zgarish qoldirish kerak bo'lib, lekin shu bilan birga jadvaldagi mos satr maydonlariga o'qitiladigan fan haqida yangi ma'lumotlar kiritish uchun so'rov quyidagi ko'rinishda bo'ladi.

UPDATE SUBJECT1

SET SUBJ_NAME= 'Высшая математика', HOUR=36, SEMESTER=1

WHERE SUBJ_ID=43;

UPDATE komandasini **SET** parametrida skalyar ifodalarni ishlatish mumkin. Skalyar ifodada maydon sifatida o'zgartirilayotgan va boshqa maydonlar kiritilib, u maydon qiymatini o'zgarish usulini ko'rsatadi

UPDATE UNIVERSITY1
SET RATING=RATING*2;

Masalan: STUDENT1 jadvaldagi STIPEND maydon qiymatini Moskva shahri talablari 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 STUDENT1
SET STIPEND=NULL
WHERE CITY= 'Moskva';

INSERTda qism so'rovlarini ishlatish

INSERT operatorini qism so'rovi bilan ishlatish 1 ta jadvalga birdaniga bir nechta satr yuklash imkonini beradi. VALUES ishlatuvchi INSERT operatori 1ta satr qo'shsa INSERTli qism so'rov jadvalga, 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 jadvalini strukturasi(tuzunlikilmasi), STUDENT jadval strukturasi to'la mos bo'lsin. STUDENT jadvalidan Moskva shahri talabalar barchasi haqida yozuvlari bilan STUDENT1 jadvalni 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 parametrda 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');

Tashqi so'rovlar jadvaliga asoslangan qism so'rovlarini ishlatish

Nazorat savollari

1. SQL tilining jadval yaratish komandalari.
2. Indeks komandasini ko‘rinishi va uning vazifasi.
3. SQL tilining jadval to‘ldirish va takomillashtirish komandalarini tavsiflang.

11- mavzu: Tranzaksiyalarni boshqarish. So‘rovlar yaratish va qayta ishlash.

REJA:

1. Ma’lumotlarni taqsimlangan qayta ishlash.
2. Zamonaviy taqsimlangan ma’lumotlar bazasini arxitekturasini.

Tayanch iboralar: *server, mijoz, tranzaksiya, commit, rollback, fayl server.*

Ma’lumotlar bilan ishlaganda ma’lumotlarni yo‘qotish yoki uzulishh extimoli bor. Bu extimollik MB qancha katta bo‘lsa va unga beriladigan so‘rovlar murakkab bo‘lganda ortadi. MBBT da ma’lumotlarni maxsus ximoyalash vositalari bor. Lekin ba’zi xollarda SQL yordamida tashkil qilinadigan ximoya usullaridan foydalanish mumkin. Bunda bir necha SQL operatorlari tranzaksiya deb nomlanuvchi bitta bloka birlashtiriladi.

Tranzaksiyada yoki barcha opreatorlar bajariladi, yoki birortasi xam bajarilmaydi.Oxirgi xolda MB tranzatsiya bajarishgacha bo‘lgan boshlang‘ich xolatga qaytariladi.Boshqacha aytganda tranzaksiyada birorta operator bajarilmasa, unda bu tranzatsiyada barcha bajarilgan operatorni barcha ishlari bekor qilinadi. Buni otkat (orqaga qaytish) deyiladi. Tranzaksiyada SQL barcha operatorlarini ishlatish mumkin.Bundan tashqari

COMMIT - Bajarishni yakunlash;

ROLLBACK - Orqaga qaytish

komandalari xam ishlatiladi.

Agar axborot tizimi bir nechta bog‘langan komptterlarda amalga oshirilsa, u taqsimlangan deyiladi. MB taqsimlangan bo‘lganda ular faqat fizik jihatdan ajratilag bo‘ladi, logik jihatdan esa ular integrallashgan bo‘ladi, ya’ni barcha MB ixtiyoriy tugun komptterlardan murojaat qilish imkoniga ega. Taqsimlangan MB ishlarni bir qancha tashkil etuvchilar o‘rtasida taqsimlanishi bilan birga ularning alog‘ida komponentalarini ishdan chiqishiga, sezgirlik kamayishiga olib keladi. Ma’lumotlarni bir butunligini saqlash ta’minlanadi. Bu taqsimlangan MB ni afzalligidir. SHu bilan birga taqsimlangan MB bir qancha kamchiliklarga ham ega.

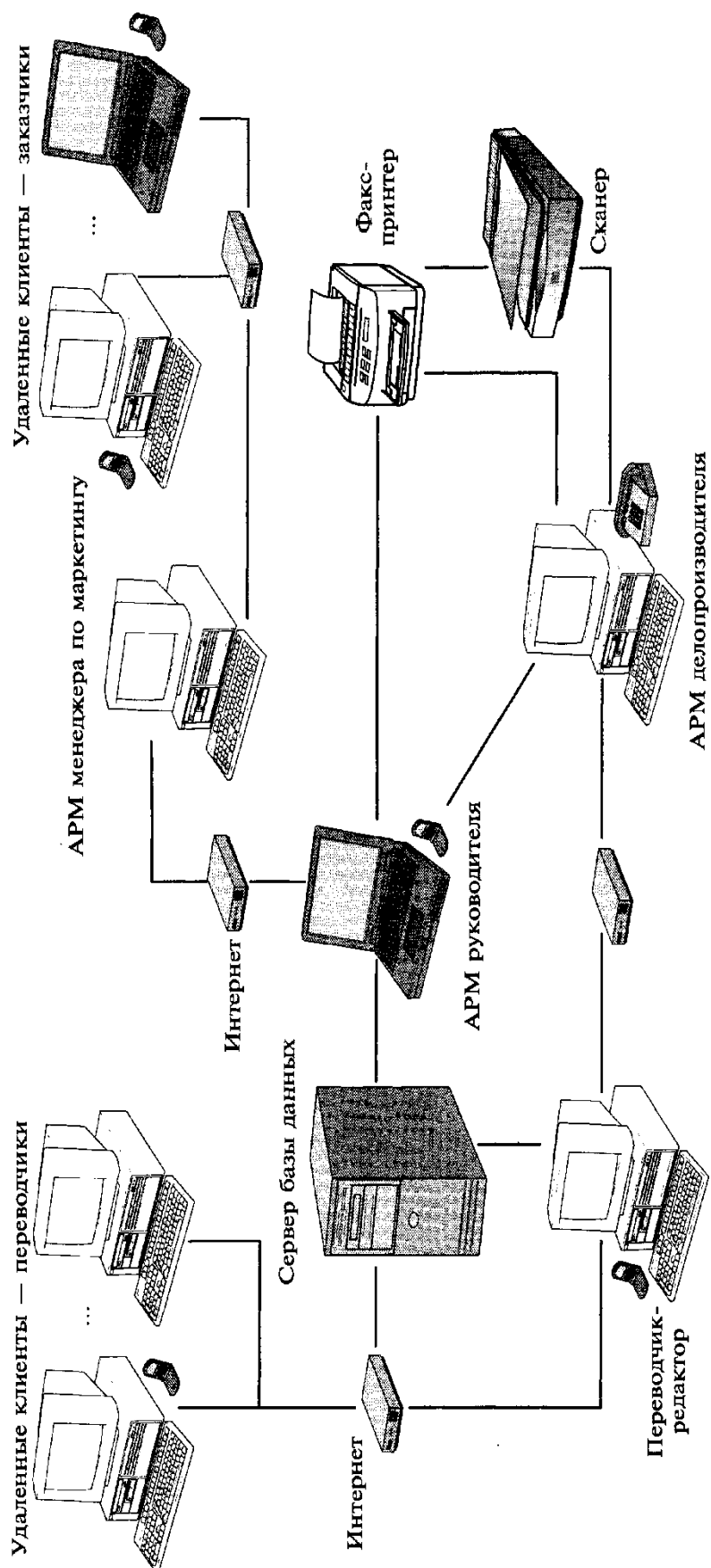
Jumladan ularni loyihalash va MB ni ko‘zatish murakkab, MB ni takomillashtirish va sinxron qayta ishlash muammolari qiyinlashadi, MB ni mahfiyligini saqlash murakkablashadi. MB ni komponentalarini bir jinsliligiga qarab ular bir jinsli va har xil jinsli sistemalarga bo‘linadi. Resurslarni taqsimlanishiga qarab esa MB taqsimlangan sistemaga va MBBT taqsimlangan sistemasiga ajratamiz. Buni quyidagi chizmada ko‘rsatishimiz mumkin:

Taqsimlangan MB da informatsiyani joylashtirish va ularni qidirish muhim masalardan biridir. Ma’lumotlarni qidirish ma’lumotlarni strukturasiga moslab,

yoki qiymatiga moslab amalga oshiriladi. 1 – holda lokal MB da ma'lumotlarni strukturasi bir – biridan farq qiladi. 2 – holda MB umumiy strukturaga ega bo'ladi, lokal MB da faqat aniq qiymatlar bilan farq qiladi. Taqsimlangan MB da ma'lumotlarni qidirishni quyidagi variantlari mavjud:

- 1) Foydalanuvchi eng yaqin sistema bilan o'zaro bog'langan, agar eng yaqin MBBT da kerakli informatsiya bo'lmasa, unda qidirish MB da bajariladi.
- 2) Qidirilayotgan ma'lumotlar strukturali informatsiya bo'yicha tugunlarda amalga oshiriladi. Bunday struktura informatsiyasi barcha lokal sistemalarda saqlanishi lozim.

Bunda bitta sistema ajratib boshqaruvchi deb hisoblanadi va unda barcha informatsiya strukturalari saqlanadi. Foydalanuvchi 1- qidirishda kerakli axborotni topolmasa, u struktura informatsiyasiga murojaat qiladi.



Taqsimlangan ma'lumotlarni qayta ishlashni asosiy mohiyati shundan iboratki,

foydalanuvchi axborotlar bilan ta'minlovchi va ist'emol qiluvchi bir necha tarmoq xizmatlari va amaliy jarayonlar bilan ishlash imkoniyatiga ega bo'ladi.

Ma'lumotlar qayta ishlashni taqsimlangan tizimlari (MQITT – SROD(Системы распределенной обработки данных)) asosini ma'lumotlar bazasini boshqarishni taqsimlangan tizimi (MBBT – RSUBD распределенная система управления базой данных) tashkil qiladi.

Hozirgi kunda MQITT keng rivojlanib boryapti. Bunga birinchi o'rinda biznes – arxitekturani axborot tizimlaridagi ideologiyasiga keng ta'siridan deb tushuntirish mumkin.

MQITT tarqqiyoti jarayonida ma'lumotlarga murojaat modellari ham rivojlanib bordi. Hozirgi kunda uchta asosiy modeli ishlatilmoqda.

1. Fayl – server modeli
2. Ma'lumot baza serveri modeli
3. Ilova serveri modeli

Fayl server modelida ilova ishchi stansiyalarida bajariladi. Ilova MBBTni yadrosi va foydalanuvchi bilan, talab qilingan hisoblash mantiqini ta'minlovchi, muloqatni tashkil qilish modeliga ega bo'lib, ko'pincha fayl serverli modelida MBBTni yadrosi ilovani boshqa komponentalari bilan bog'liq bo'lgan funksiyalar to'plamidan iborat. Fayl serverida faqat MB (indeksli ma'lumot fayli va boshqalar) va ba'zi bir texnologik fayllar (Overlayli fayllar, tartiblangan fayllar va boshqalar). MBBT murojaat qilish operatori, amaliy dastur ((AD – PP) prikladnaya programma) kiritilgan (kodlangan) bo'ladi. U MBBT yadrosi tomonidan ishchi stansiyalarda qayta ishlanadi. MBni boshqarish tizimi operatorlar bajarish uchun, MB fayliga murojaat tashkil qiladi.

Tarmoq bo'yicha, ma'lumotlar, indekslar, oraliq va natijaviy ma'lumotlar, texnologik fayllarni bloklarini o'qish/yozish uchun so'rovlar jo'natiladi.

Fayl – server asosida hozirgi kunda FoxPro, Clipper, Paradox kabi MBBT ishlaydi. Bu sinf MBBT ancha arzon, ularni o'rnatish engil, o'zlashtirish oson. Lekin ular bir qancha kamchiliklarga ega.

Bu MBBT asosida ishlab chiqarilgan tizimlar ishlab chiqarish past, chunki barcha oraliq ma'lumotlar past tezlikka ega bo'lgan tarmoq shinalari bo'yicha uzunlikatiladi, amaliy dastur va MBBT yadrosi kam quvvatli ishchi stansiyalarda bajariladi.

Bunday MBBT taqsimlangan qayta ishlashni ta'minlamaydi.

Ma'lumot baza serveri modeli. Ma'lumot baza serveri modelida, ilova ham asosan ishchi stansiyalarda bajariladi. Ilovaga foydalanuvchi va biznes – qoida bilan muloqot tashkil qilish uchun model kiritilgan. MBBT yadrosi barcha ishchi stansiyalar uchun umumiy bo'lib, u serverda ishlaydi. MBBT murojaat qilish operatori (SQL – operatorlar) AD – amaliy dasturga kodlanib kiritilgan bo'lsa ham, ishchi stansiyada bajarilmaydi. U qayta ishlash uchun serverga jo'natiladi. MBBT yadrosi indekslar va boshqa qo'shimcha (oraliq) ma'lumotlar murojaat qilib, so'rovni translyasiya qiladi va uni bajaradi.

Ishchi stansiyalarga faqat operatorlarga qayta ishlash natijalari jo'natiladi.

Zamonaviy MBBT serverda saqlanuvchi protseduralar va triggerlar ham ishga tushirilishi mumkin. Saqlovchi protsedura va triggerlar MBBT yadrosi bilan

birgalikda MBBT serverini tashkil qiladi. Saqlanuvchi protseduralarga ishchi stansiyalardagi ilovalardan ham murojaat qilsa bo‘ladi. Bu amaliy dastur kodi hajmini kamaytirishga imkon beradi va ishchi stansiyalardan SQL – operatorlar oqimini kamaytirish imkonini beradi, bu esa kerakli SQL – operatorlar guruhini saqlovchi protseduralarda kodlash mumkin.

Trigger – bu MBBT yadro tomonidan barariladigan dasturlar bo‘lib ular MB jadvallarini tiklashdan (UPDATE, INSERT, DELETE) oldin va keyin bajariladi. YA’ni triggerlar ma’lumot baza jadvallarini tiklashda (UPDATE, INSERT, DELETE) oldin va keyin, MBBT yadrosi tomonida bajariladigan dasturlardir. Ular MBni butunligini avtomatik ravishda ta’minlab beradi.

MB serveri modeli quyidagi MBBTlar ta’minlaydi. ORACLE, Sybase, Informix, Ingress, Progress va boshqalar. ORACLE, Sybase, Informix MBBT bozorini 80% egallagan.

MBBT bu sinfini afzalliklari:

Bu MBBT asosida yaratilgan sistemalar yuqori ishlab chiqarishga ega, chunki shinalar bo‘yicha faqat SQL so‘rovlari va bajarish natijalari uzunlikatiladi.

- MBBT taqsimlangan qayta ishlashni ta’minlaydi
- Bu MBBT doirasida juda ko‘p server programmalar bo‘lib, ular ilovalar yaratishni va taqsimlangan tizimlar yaratishni engillashtiradi.

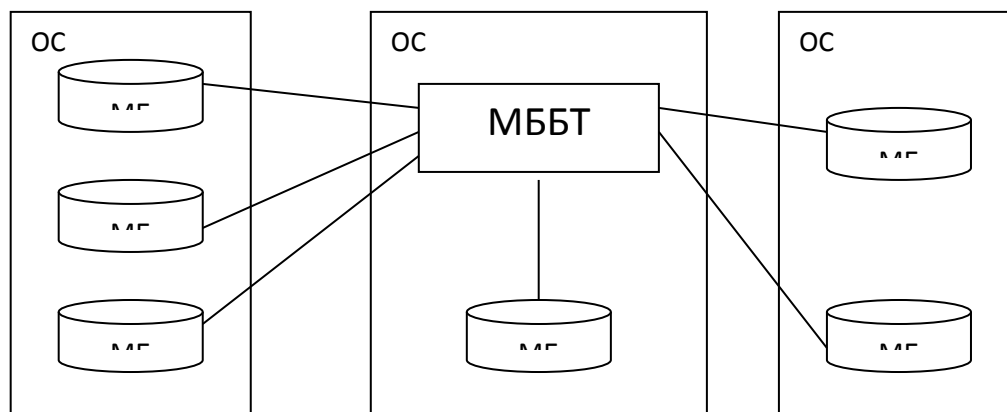
Kamchiligi:

- Ular oldingi sinf MBBTga nisbatan qimmat, o‘zlashtirish qiyin
- Ularni samarali ishlashi uchun yuqori tezlikli (shuning uchun qimmat) server va tarmoqlar talab etiladi.

Ilova serveri modeli.

Ilova sereverini saqlovchi protseduralar yordamida ham tashkil qilish mumkin, lekin ularni amalga oshirish uchun yuqori bosqichli tillar ishlatiladi (masalan, ORACLE – PL/SQL tili). Shuning uchun ilovalar ko‘p resurslar talab qiluvchi (resursoemkimi) shu bilan birga bu tillarning imkoniyatlari cheklangan; ular yordamida ma’lumotlar “nozik” qayta ishlashni (masalan, bitlar darajasida) tashkil qilib bo‘lmaydi.

Saqlanuvchi protseduralar taqsimlangan ifodalarni ta’minlamaydi, ya’ni Ular kerakli dasturlarni boshqa serverda avtomatik ishga tushurib berishni ta’minlamaydi. Bu kamchilikni yo‘qotish uchun, maxsus vositalar ishlab chiqilgan. Ular ko‘pincha tranzaksiya menedjrlari, tranzaksiya monitorlari OLTP (ONLINE TRANSACTION PROCESSING) deb ataladi.



3) va u barcha kerakli informatsiyani qidirishni davom ettiradi.

Informatsiya strukturasi asosan, tarmoqni lug‘at ma’lumotnomasida saqlaydi. Bunday lug‘at tarmoq tugunlaridagi ma’lumotlarni joylanishi haqidagi axborotni, tugunlardagi MB ni umumiy logik strukturasi va foydalanuvchilar va ularga bo‘lgan huquqlar haqidagi ma’lumotlarni saqlaydi.

Nazorat savollari.

1. Fayl – serverli ma’lumot bazasini afzallik va kamchiliklarni aytib bering.
2. Ilova serveri modeli qanday muamolarni yechish uchun ishlab chiqilgan?
3. Ochiq sistemalarga (tizimlarga) tegishli xususiyatlarni sanab o‘ting.
4. Ma’lumotlarni taqsimlangan qayta ishlashtizimlarini rivojlanish istiqbollari qanday?

VI. MUSTAQIL TA'LIM MAVZULARI

1. Объектга йўналтирилган маълумотлар базаси.
2. Кодд реляцион алгебраси. Реляцион ҳисоблаш.
3. Объектли маълумотлар базаси. Лойихалаш, қайта ишлаш ва тадқиқ этиш.
4. NoSql маълумотлар базаси ва унинг асосий принциплари.
5. Маълумотлар базасини шакллантиришда технологиялар перспективалари.
6. Амалиётда маълумотлар базасини қуришда маълумотларнинг реляцион модели.
7. ORACLE маълумотлар базаси.
8. Маълумот базаси тизимининг архитектураси.
9. Реляцион маълумот модели. Маълумотлар базасида муносабатлар.
10. Маълумотлар базасини режалаштириш, лойихалашлаш ва администраторлаш.
11. SQL тили. SQL операторларини ёзиш.
12. Транзакцияларни бошқариш.
13. Маълумотлар базасини администраторлаш ва хавфсизлигини таъминлаш.
14. Очiq маълумотлар базаси алоқаси (ODBC) интерфейси.
15. ADO технологиясидан фойдаланиш.
16. ADO ва C++ орқали майдон қийматларини киритиш, ёзувларни ўзгартириш, қўшиш ва ўзгартириш.
17. XML ва маълумотлар базаси
18. C++ ва SQL орқали янги маълумотлар қўшиш, ўзгартириш ва ўчириш.

VII. GLOSSARIY

Termin	O'zbek tilidagi sharhi	Ingliz tilidagi sharhi
SQL	Strukturalangan so'rovlar tili	Structured Query Language

Access	Microsoft Office RMBBT dasturi	The Microsoft Office RDBMS application
Append	Jadval oxirigacha ma'lumotlarni qo'shish	Adding data to the end of table
Accending order	Eng past va eng yuqori uchun sanada asoslangan matn sohasida alifbo tartibi	In order from lowest to highest. Also called alphabetical order, when a sort is based on a text field, and chronological, when a sort is based on a date field
Autonumber field	Yozishga qaraganda katta maydonga qo'shimcha ravishda avtomatik saqlash	A field that automatically stores a numeric value ,that is one greater than that in the last record added
Database	Tegishli ma'lumotlarni yig'ish	An organized collection of related data
Database schema	Ma'lumotlar bazasida jadvallar yacheykasiga ma'lumotlarning bayoni va ma'lumotlarni uzatishni tashkil etish	A description of the data, and the organization of the data, into tables in a relational database
Datasheet	Ma'lumotlar uchun satrlar ustunlar sohalarda va yozuvlar bilan tashkil etish	The data for a table organized with fields in columns and records in rows
Datasheet view	Satrlar ustunlar sohalarda va yozuvlar bilan, bir ma'lumot sahifasida bir stol asosiy tuzilishini ko'rish uchun ishlatiladi	Used to display the basic structure of a table in a datasheet, with fields in columns and records in rows
Date/time field	Maydon sana yoki vaqtni saqlaydi	A field that stores a date or time
Decending order	Oliy maqsadidan eng past uchun	In order from highest to lowest
Design view	Jadvallar uchun maydon ta'riflar ko'rsatadi	The table view that shows the field definitions for a table
Entry	Jadval uchun ma'lumotlar	The data for a field
Field	Jadvaldagi maydonlarni belgilaydi	A column in a table. Used to store data
OLAP	Хақиқий вақтда маълумотларга аналитик	On-Line Analytical Processing

	ишлов бериш	
OLTP	Хақиқий вақтда транзакцияларга ишлов бериш	On-Line Transaction Processing

Form	So`rovlar yordamida ma`lumotlarni ko`rishda ishlatiladi	A database object used for entering records into a table, and for viewing existing records
Long integer	Uzun butun toifa	A field size that indicates a whole number
Lookup field	Maydondagi ma`lumotlarni saqlaydi	A field that stores data; retrieved from a field in another table
Name	Jadval nomi bo`lib, soz orqali ifodalanadi	Word or words, used to describe the data stored in a field
Primary key	Birlamchi kalit hisoblanadi	A field in a table that is designated to contain unique data.
RDBMS	Relatsion ma`lumotlar bazasini boshqarish tizimi	(Relational Database Management System) A software application that contains tools to manage data, answer queries, create user-friendly forms for data entry, and generate printed reports.
Record	Jadvaldagi maydonlarni uchun ma`lumotlar majmui	A set of data for fields in a table
Table	Ma`lumotlar bazasi obyekti. Satr va ustunlar ichiga tashkil etilgan tegishli ma`lumotlarni saqlaydi	A database object that stores related data organized into rows and columns.
Text field	Jadvallarda belgilar (harflar, belgilar, so'zlar, harflar va raqamlar kombinatsiyasini) hisob talab qilmaydigan va sonlar saqlaydi	A field that stores characters (letters, symbols, words, a combination of letters and numbers) and numbers that do not require calculations.
Updating	Yozuvni o`zgartirish	Modifying a record
Yes/No field	ha / yo'q, to'g'ri / noto'g'ri, yoki / off vakillik qilish.	A field that is either selected or not selected to represent yes/no, true/false, or on/off.
ERP	Корхона ресурсларини режалаштириш	Enterprise Resource Planning
CRM	Мижозлар билан ўзаро	Customer Relations

	муносабатларни бошқариш	Management
LAN	Локал ҳисоблаш тармоғи	Local Area Network
MAN	Маҳаллий ҳисоблаш тармоғи	Metropolitan Area Network
WAN	Худудий ҳисоблаш тармоғи	Wide Area Network
ISO	Ҳалқаро стандартлаштириш ташкilotи	International Organization for Standardization
WWW	Умумжаҳон ўргамчак тўри	World Wide Web
ASCII	Ахборот алмашишнинг Америка стандарти	American Standard Code for Information Interchange

VIII. ADABIYOTLAR RO'YXATI

I. Me'yoriy- huquqiy xujjatlar.

1. O'zbekiston Respublikasi Prezidentining «Oliy ta'lim muassasalarining rahbar va pedagog kadrlarini qayta tayyorlash va malakasini oshirish tizimini yanada takomillashtirish chora-tadbirlari to'g'risida» 2015 yil 12 iyundagi PF-4732-son Farmoni.
2. O'zbekiston Respublikasi Prezidentining 2010 yil 2 noyabrdagi «Oliy malakali ilmiy va ilmiy-pedagogik kadrlar tayyorlash tizimini yanada takomillashtirish chora-tadbirlari to'g'risida»gi PQ-1426-sonli Qarori.
3. Kadrlar tayyorlash milliy dasturi. O'zbekiston Respublikasi Oliy Majlisining Axborotnomasi, 1997 yil. 11-12-son, 295-modda.
4. O'zbekiston Respublikasi Prezidentining 2012 yil 24 iyuldagi «Oliy malakali ilmiy va ilmiy-pedagog kadrlar tayyorlash va attestatsiyadan o'tkazish tizimini yanada takomillashtirish to'g'risida»gi PF-4456-son Farmoni.

II. Maxsus adabiyotlar.

1. Григорьев Ю.А., Ревунков Г.И.. Банки данных. М.: Изд. МГТУ им. Баумана, 2002.
2. Роб П. Системы баз данных: проектирование, реализация и управление (5-е издание) издательство "БХВ - Санкт-Петербург" · 1200 стр, 2003 г. ·.
3. Григорьев Ю.А., Плутенко А.Д.. Жизненный цикл проектов распределенных баз данных. Благовещенск АмГУ, 1999.
4. Дунаев С.С. Доступ к базам данных и техника работы в сети. Практические приемы современного программирования. М.: Диалог – МИФИ, 1999.
5. Дж.Ульман, Дж Уидом. Введение системы баз данных. Пер.с англ. М.: «Лори», 2000.
6. Диго С.М. Базы данных Проектирование и использование. издательство "Финансы и статистика" · 592 стр, 2005 г.
7. Конноли Т., Брегг К. Базы данных, проектирование, реализация и сопровождения, теория и практика, Университет Пейсли, Шотландия, изд. М.- СПб.- Киев, 2003.
8. Четвериков, В. Н. Базы и банки данных [Текст] : учебник для вузов по спец. "Автоматизир. системы управления" / Г. И. Ревунков, Э. Н. Самохвалов. - М. : Высш. шк., 1987. - 248 с. : ил. - Библиогр.: с.246 (14 назв.). Предм. указ.: с. 247.
9. А. Д. Хомоненко, В. М. Цыганков, М. Г. Мальцев Базы данных [Текст] : учебник для вузов / - 4-е изд., доп. и перераб. - СПб : Корона принт, 2004.

- 736 с. - 1 экз.

10. Б. Я. Советов, В. В. Цехановский, В. Д. Чертовский. Базы данных. Теория и практика [Текст] : учебник для студ. вузов / - М. : Высш. шк., 2005. - 463 с. : ил. - Список лит. с. 459-460. - 2 экз.

11. Четвериков В.Н. и другие «Базы и банки данных» Москва, ВШ, 1987г.

12. Кливерт Ч. Энциклопедия ползователя Делпхи2 ДиаСофт, Киев, 1996г.

13. А.Я.Архангелский Делпхи 7, Москва, изд. «Бином», 2003 г.

14. Т. Коннолли, К. Брегг. Базы данных. Университет Пейсли, Шотландия, изд М СПБ Киев, 2003г.

III. Internet resurslar.

1. [www. Ziyonet.uz](http://www.Ziyonet.uz)
2. www.library.tuit.uz
3. www.intuit.ru
4. <http://www.w3schools.com>
5. <http://www.sql-tutorial.ru>
6. <http://pitbooks.ru>
7. <http://www.torrentino.ru>