

Termiz davlat universiteti

Fizika-matematika fakulteti

«Amaliy matematika va informatika» kafedarsi

Axborot resurslari va berilganlar bazalarini  
boshqarish  
(Uslubiy qo`llanma)

Termiz – 2015

Ushbu uslubiy qo'llanma Termiz davlat universiteti o'quv metodik kengashining 2015-yil sentabr oyidagi yig'ilishida ko'rib chiqilib, yig'ilish qaroriga asosan chop etishga ruxsat etildi va №\_\_ bayonnoma bilan ro'yxatga olindi.

Tuzuvchilar:

M.K.DJurayev – “Amaliy matematika va informatika” kafedrası  
k.o'qituvchisi

I.K. Choriyev - “Amaliy matematika va informatika” kafedrası  
o'qituvchisi

Taqrizchilar:

Iqtisod fanlari doktori:

Hatamov O.Q.

“VO`QT va MOI” informatika o'qituvchisi:

B. Xurramov

В учебном пособии рассмотрены вопросы информации ресурсы и управление базами данных, в частности, вопросы новые информационные технологии, графическое программирование, базы и банки данных, как составляются базисных программы.

Учебное пособие предназначено для студентов, обучающихся по направлению 513200 - Прикладные математики и информатика, и всем обучающимися бакалаври.

Ushbu o'quv qo'llanmada kompyuterlarning dasturiy ta'minoti, xususan tizimli dasturiy ta'minot, operasion tizimlar, faylli menedjerlar, dasturiy utilitlar, kompyuterlarning ishlash xavfsizligi kabi jarayonlar ko'rib chiqilgan.

O'quv qo'llanma 5480100 - "Amaliy matematika va informatika" ta'lim yo'nalishida ta'lim olayotgan talabalar, hamda barcha bakalavr tizimi talabalari uchun mo'ljallangan.

In scholastic allowance are considered questions of software computer and, in particular, questions of the operating system software: operating systems, file managers, program-utilities, bases to safety of the functioning computer. The Scholastic allowance is intended for student, training on direction 5480100 – an “praktical matematikal”

## Kirish

Mamlakatimiz ishlab chiqarish sohasini axboratlashtirish jamiyat rivojlanishining ob'ektiv jarayonini hamda zarur bo'lgan axborotlarni yig'ish, saqlash, uzatish, qayta ishlash va taqdim etishning tabiiy davomidir. Iqtisodiy, ishlab chiqarish, aloqa, ilmiy -tadqiqot, ta'lim, tibbiyot va biznes sohaslaridagi mehnat sifati, mehnat unumdorligi va samaradorlik daraja soni yuksaltirish ularda tadbiq qilinayotgan eng zamonaviy axborot- kommunikatsiya texnologiyalari bilan bog'liq.

Zamonaviy axborot- kommunikatsiya texnologiyalari to'plangan axborot mahsulotlarini kishilarga tezkor sur'atda yetkazib sermehnatlik daraja soni kamaytirgan holda mavjud muammolarni hal etish uchun keng imkoniyatlar yaratib bermoqda. Shuning uchun ham axborot-kommunikatsiyalar texnologiyalarini iqtisodiyotning barcha tarmoqlarida samarali qo'llash mamlakatni texnologik va iqtisodiy jihatdan rivojlantirishni ifodalovchi ko'rsatkich bo'lib xizmat qilmoqda.

O'zbekiston Respublikasi Prezidenti Islom Karimov mamlakatimizni ijtimoiy-iqtisodiy rivojlantirish hamda iqtisodiy dasturning eng muhim ustuvor yo'nalishlariga bag'ishlangan vazirlar mahkamasining majlisidagi ma'ruzasida quyidagilarni aytib o'tdilar:

“biz qisqa vaqt mobaynida nafaqat axborot xizmatlari ko'rsatishning ko'plab turlari bo'yicha mavjud kamchiliklarni bartaraf etishimiz, balki axborot-kommunikatsiya texnologiyalarini joriy etish borasida yuksak darajaga erishgan ilg'or mamlakatlar safiga qo'sholishimiz zarur”

XXI asr jahon talablariga mos ravishda javob beradigan kadrlar avvalom bir axborot-kommunikatsiya texnologiyalarini mukammal bolishlarini talab etmoqda. Insonning barcha sohalardagi faoliyati axborot bilan to'liq ta'minlanganligi va axborotlardan samarali foydalanishiga bog'liq bo'lmoqda. Katta hajmdagi axborotlar oqimidan kerakli bo'lganini tanlab olish, ularni qayta ishlash va foydalanish uchun har bir mutaxassis shaxsiy kompyuter va aloqa vositalaridan unumli foydalanishi kerakdir. Insonlarda axborot madaniyatini shakllantirish uchun «Informatika va axborot texnologiyalari» fani xizmat qiladi, uni o'rganish jaroyani maktab partasidan boshlanib, oliy o'quv yurtlarida davom etadi. Hozirgi kunda axborot jamiyatning strategik resursiga aylanib, davlatning iqtisodiy jihatdan rivojlanib borishini belgilab bermoqda.

O'zbekiston Respublikasi Prezidenti I. Karimov "Jahon moliyaviy-iqtisodiy inqirozi, O'zbekiston sharoitida uni bartaraf etishning yo'llari va choralari" asarida ta'kodlaganidek: Yurtimizda aloqa, axborotlashtirish, moliya, bank, transport xizmati ko'rsatish, maishiy texnika va avtomobillarni ta'mirlash sohaslari ancha yuqori sur'atlar bilan rivojlandi. So'ngi 2010-2014 yillarda o'rtacha 50 foiz o'sishni ta'minlayotgan axborot-kommunikatsiya texnologiyalari sohasida xizmat ko'rsatish jadal rivojlanayotganini alohida ta'kidlash lozim"<sup>1</sup>.

---

<sup>1</sup> I.A. Karimov. Jahon moliyaviy-iqtisodiy inqirozi, O'zbekiston sharoitida uni bartaraf etishning yo'llari va choralari. /I.A. Karimov. –T. O'zbekiston, 2009. -56 b.

Mamlakatimizda barqaror va samarali iqtisodiyotni shakllantirish borasida amalga oshirib kelinayotgan islohotlar bugungi kunda o'zining natijalarini namoyon etmoqda. Jumladan, qisqa vaqt ichida iqtisodiyotda chuqur tarkibiy o'zgarishlarni amalga oshirish, aholi daromadlarining o'sishini ta'minlash, samarali tashqi savdo hamda investitsiya jarayonlarini kuchaytirish, qishloq xo'jaligini isloh qilish, kichik biznes va xususiy tadbirkorlik sohasini barqaror rivojlantirish, bank-moliya tizimi faoliyatini mustahkamlashda ahamiyatli yutuqlar qo'lga kiritildi.

O'zbekistonning xalqaro iqtisodiy maydondagi nufuzi va mavqiy sezilarli darajada va muntazam o'sib bormoqda. Bunda mamlakatimiz rahbari Islom Karimov tomonidan ijtimoiy-iqtisodiy rivojlanish strategiyasining puxta ishlab chiqilganligi, iqtisodiy islohotlar maqsadi va vazifalari, amalga oshirish yo'llarining aniq va to'g'ri ko'rsatib berilganligi bo'sh maqsad yo'lidagi yutuq va marralarning salmoqli bo'lishiga imkon yaratdi.

Hozirgi davrda dunyo mamlakatlari ijtimoiy-iqtisodiy taraqqiyoti o'zining ma'no-mazmuni jihatidan oldingi bosqichlardan keskin farq qiladi. Bunda eng asosiy va muhim jihat – milliy iqtisodiyotlarning tobora integratsiyalashuvi va globallashtirish kuchayib borishidir. Ayni paytda bu jarayonlar xalqaro maydondagi raqobatning ham keskinlashuviga, har bir mamlakatning xalqaro mehnat taqsimotidagi o'z mavqeini mustahkamlash uchun kurashining kuchayishiga ham ta'sir ko'rsatadi.

Axborot resurslarini oqilona tashkil etish va foydalanishda ular mehnat, moddiy va energetik resurslar ekvivalenti sifatida namoyon bo'ladi. Ayni paytda axborot-bu boshqa barcha resurslardan oqilona va samarali foydalanish hamda ularni asrab-avaylashga ko'maklashuvchi yagona resurs turidir.

Axborotlashtirish jarayonini deganda – inson faoliyatining muhim yo'nalishlarida olingan bilimlardan samarali foydalanish uchun ko'rilgan kompleks chora-tadbirlar tushuniladi.

Zamonaviy va samarali yechimlar topish uchun ko'plab, struktura jihatidan murakkab axborot tizimi yaratilmoqda, natijada, axborotlashtirish jarayonida ishtirok etuvchilar soni kun sayin ortib bormoqda. Bu jamiyat va moddiy ishlab chiqarish tarmoqlarining ko'plab mablag'larini shu sohaga jalb qilishga olib kelmoqda. Bu o'z navbatida insonlarni axborot resurslaridan ratsional foydalanish yo'llarini qidirishga majbur qilmoqda. Zamonaviy sharoitda yangi axborot oqimi qanchalik tez ko'paysa shu bilan birga ularning eskirish muddatlari ham tezlashmoqda, bu o'z navbatida, axborotni tanlash, unga erishish qiyinchiliklarini keltirib chiqarmoqda.

Jamiyatni axborotlashtirish – yuridik va jismoniy shaxslarning axborotga bo'lgan ehtiyojlarini qondirish uchun axborot resurslari, axborot texnologiyalari hamda axborot tizimlaridan foydalangan holda sharoit yaratishning tashkiliy ijtimoiy iqtisodiy va ilmiy-texnikaviy jarayonidir.

## **1.BOB. AXBOROT RESURLARI VA ULARDAN FOYDALANISH SAMARALARI**

**Axborot resurs turlaridan biri sifatida.** Axborot tovarning har ikki xususiyatiga: iste'mol qiymatining mavjudligi bu (foydaliligi, qadr-qimmati) va qiymatiga (ya'ni ijtimoiy mehnatning ayrim sarfiga) ega bo'ladi.

Axborot va tovarlar sifatidagi buyum-narsalar ob'ektining umumiy va farqli xususiyatlari mavjud.

Bu ob'ektlarning odatdagi mahsulot va moddiy zahiralalar bilan quyidagi bog'likligi bor: ularga iste'mol so'rovi mavjud, ular mulkiy ob'ektlardir, ya'ni ularga ega bo'lish, foydalanish va egalik qilish mumkin, ular aniq ishlab chiqaruvchilar (ta'minotchilar)ga ega, ular qiymat va tegishli narxga ega, ular turli shart-sharoitlarda yetkazib berilishi mumkin.

Biroq axborot zahiralari va texnologiyalarining o'zlari bir satr jiddiy farqlari mavjudki, ular qatoriga quyidagilar kiradi:

- ✓ Cheklanmagan miqdorda sotish;
- ✓ Amaliy jihatdan yo'q bo'lmaslik;
- ✓ Aniq foydalanuvchilar shart-sharoitlariga individual moslash zaruriyati (umum tizimli paketlardan ham tashqari);
- ✓ Ob'ektlarni muallif hamrohligida yetkazib berish majburiyati;
- ✓ Nafaqat foydalanish, balki ikkilamchi tirajlashga ham turli cheklanmalar qo'yish imkoniyati;
- ✓ Mualliflik yoki ta'minotchilik xuquqlariga rioya qilishni identifikatsiyalash (bir xillashtirish) ning murakkabligi;
- ✓ Takrorlanuvchi ob'ektlarning ko'pligi. Turli o'xshash vazifalarni bajaruvchi mahsulotlardan farqli ravishda, axborotning bir aniqligini turlicha aks ettirishi mumkin. Masalan, savdo-sotiq uchun biror bir mahsulotning miqdori haqida yakuniy axborotning bir necha variantlari taklif etolishi mumkin;
- ✓ Bilvosita axborotning foydaliligi;
- ✓ Foydalanish natijasida qadrsizlanish. Chindan ham, ma'lumotlar bilan tanishib, ulardan ayrimlarining talabga javob bermasligiga ishonch hosil qilib, xarid haqidagi taklifni qondirish to'g'ri bo'ladi. Biroq takliflar bo'yicha ishonchli ma'lumotlardan foydalanmaslik uni majburiy unutish butunlay mumkin emas;
- ✓ Baholashni oldindan bolib bo'lmasligi. Agar mahsulot ba'zi cheklangan dinamikada baholansa (talabning bir miqdorda yo'q bo'lishli yuzaga kelishi mumkin emas), axborot esa (tekin texnologiya emas) bir lahzada butkul no'l darajaga tushib ketishi mumkin;
- ✓ Oddiy raqobat sharoitida taklif etilgan bir axborot ikkinchisining dolzarblik xususiyatini yo'qqa chiqarishi mumkin;
- ✓ Iste'molchilik xususiyatlarining qisman to'liq noaniqliligi;
- ✓ Faqat jismoniy eskirish va belgilangan noaniqlik vaqtida dolzarblikni yo'qotish mavjudligini anglatuvchi jismoniy yaroqlilik;

- ✓ Istemolchiga ma'lumotni qisqa vaqtda uzatish va shunday qisqa vaqtda tasdiqni qabul qolib qilish imkoniyati;
- ✓ Avtomatik tirajlanmaydigan texnologiyani doim ham aniq bir paytda yetkazmaslik;
- ✓ Yetkazib berishga doimiy ravishda tayyorlik;
- ✓ Ham sotuvchiga, ham xaridorga nisbatan ma'lumotlar va texnologiyalarni, shuningdek tovarni sotish sotmaslik vaqtini ham mahfiy saqlash imkoniyatining birligi.

Foydalanuvchilarning qiziqish darajasi quyidagi mezonlarga bog'liq:

a) sifatiga, ya'ni axborot qiymatini (foydaliligini) belgilovchi axborot ehtiyojlarini xaridlash darajasiga;

b) manfaatiga, ya'ni umuman iqtisodiy samaradorlikni oshirishga;

v) xarajatlarga, ya'ni axborot hajmi bilan belgilanadigan axborot qiymatiga.

Axborotning sifat xususiyatlari (foydaliligi)ga: to'laqonlilik, qabul qilishning bemalolligi, dolzarblik hozirjavoblik, aniqlilik va hokozalar kiradi.

1. To'laqonlilik. Axborot to'laqonliligi obyekt faoliyatining u bu tomonlarining miqdoriy va sifat parametrlarini aniq belgilash hamda Misol qarorlarni ishlab chiqarishda ifodalaniladi.

Axborotning noto'laqonliligi qarorlar qabul qilishda xatolarga olib kelishi mumkin.

2. Ishonchlilik qabul qilinadigan qarorlar samaradorligi saqlanadigan yetib kelgan va natijaviy axborotlarda muayyan darajada buzulishlarga yo'l qo'yadi.

3. Axborotni qabul qilishning bemalolligi vaqt birligida ma'lumotlarni qabul qilish tezligi bilan belgilanadi. Shu bois ham ma'lumotlar ko'proq jadval shaklida beriladi, u nafaqat axborot mazmunini ochib beradi, balki yengil qabul qilinadi ham.

4. Ma'lumotlarning dolzarbliligi muayyan vaqt mobaynida aniq vazifani amalga oshirish uchun yaroqliligini ifodalaydi. Shu bois ham dolzarblilik hozirjavoblilik va tezkorlik axborotga xos xususiyatlardir.

5. Kechikmaslik axborotning qulay belgilangan vaqtda kelib tushishini anglatadi. Bu talabni buzish axborotni qadrsizlantiradi.

6. Aniqlilik uning to'g'riligini, detallashtirish darajasini anglatadi. Axborotning aniqliligi uning barcha istemolchilar tomonidan bir xil qabul qilinishini ta'minlaydi.

7. Tezkorlik vaqt o'tgach axborot eskirishi va dolzarbliligini yo'qotishini aks ettiradi.

Axborotning o'z vaqtida qabul qilinmasligi qaror qabul qilishni kechiktiradi, oqibatda qabul qilinayotgan qarorlar o'zgaruvchan sharoitda talabga javob bermaydi. Axborot qanchalik tezkor bo'lsa, u shunchalik qimmatli bo'ladi.

Axborotning qadriligi aniqlik darajasi oshgani sayin xabar qilinayotgan va aniq natijalar o'rtasidagi farq kamayganda tez ko'tariladi. To'liqroq va ishonchli axborot to'g'ri qarorni qabul qilishni ta'minlaydi.

Axborotning qiymati ushlanib qilish vaqti oshishi bilan kamayadi, shu tufayli axborot eskiradi.

Axborotni qayta ishlashda ushlanib qilish kamayganda, birinchidan, qarorlar oldinroq qabul qilinishi mumkin, ikkinchidan, uning mazmuni yaxshilaniladi.

Axborot foydali bo`lishi uchun har bir daqiqada hal etiladigan muammo bilan bog`liq bo`lishi lozim. Faqat ishga tegishli axborotlar foydalanuvchilarga o`z vaqtida va mazmunli qaror qabul qilish imkonini beradi. Ular esa o`ziga zarur ma`lumotlarni izlashga ortiqcha vaqt sarf etmaydi. Agar mavjud axborot ishlab chiqilayotgan qaror bilan bog`liq bo`lmasa, u biror qiymatga ega bo`lmaydi. Foydalanuvchining axborot ishning qandaydir qismini bilishi ( bilmasligi) ham axborot qiymatining muhim omilidir.

«Texnologiya » atamasi grekcha *techne* so`zidan olingan bo`lib mohirlik, ustalik, biror ishni uddalay olishni anglatadi. Bu ma'lum bir jarayonga nisbatan qo`llanilgan. Jarayon deganda esa maqsadga erishishga yo`naltirilgan hatti harakatlar majmui tushunilgan. Ushbu jarayon kishi tomonidan tanlangan strategiya bilan belgilanadi va turli xildagi vositalar, usullar yordamida amalga oshiriladi.

Umumiy hollarda texnologiya deganda, mahsulotni ishlab chiqarish jarayonida amalga oshiriladigan xom-ashyo, material yarim tayyor mahsulot shakli, xususiyati, holatining o`zgarishi, uni qayta ishlash, tayyorlash usullarining majmui tushuniladi. Bu biror bir ishni yuqori darajada uddalash deganidir.

Axborot texnologiyalari to`g`risida gap ketganda, qayta ishlashning materiali sifatida ham, mahsulot sifatida ham axborot ishtirok etadi. Biroq bu obyekt, jarayon hodisa to`g`risidagi sifat jihatidan yangi ma`lumot bo`ladi. Texnologiya xodimning axborot bilan ishlash usuli va uslubi hamda texnik vositalar orqali namoyon bo`ladi.

Sanoat ishlab chiqarishida har qanday texnologiya mahsulotni yaratishning boshidan oxirigacha bo`lgan texnologik jarayonni qamrab oluvchi tarkibiy elementlari majmuining bayonini ifodalaydi. Tarkibiy elementlarining (texnologik operatsiyalar) tarkibi ikki asosiy omil bilan aniqlanadi: birinchidan, mazkur texnologik jarayon asosiga nisbatan sifatli usullar va printsiplar orqali, ikkinchidan, mahsulotni tayyorlashning oxirgi jarayonidagi texnologik operatsiyani bajarish uchun jalb etish mumkin bo`lgan asbob-uskuna vositalari orqali.

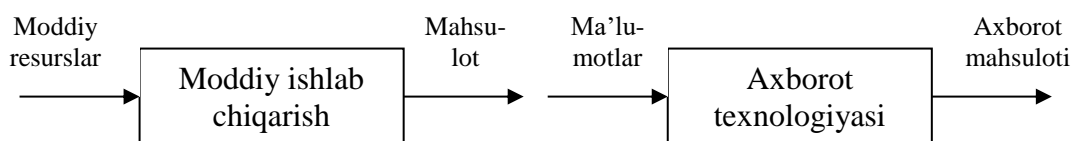
Uslublar ayrim mahsulotlarni olishning prinsipial imkoniyatini tavsiflab beradi. Ularning asosini inson tomonidan o`rganilgan (balki to`liq emas) tabiiy (fizik, kimyoviy, biologik) jarayonlar mazkur soha mutaxassislarining ilmiy izlanishlari natijasida to`plangan tajribani aks ettiruvchi ayrim qonuniyatlar tashkil etishi mumkin. Odatda muayyan bir texnologiya usullar va printsiplarni belgilovchi butun majmuaga tayanadi. Bu majmua elementlarining ahamiyati ham turlicha. Ulardan biri ishlab chiqarishning texnik jihatlarini, ikkinchisi ishning iqtisodiy tomonini, boshqa biri tashkiliy tuzilmani belgilaydi.

Uslublar va prinsiplarning turlicha roli ularning texnologiya tuzilmasiga nisbatan ta'siri har xil bo`lishini keltirib chiqaradi. Ba'zan ayrim uslub printsiplarning ishlab chiqarishga nisbatan ta'siri hisobga qilinmasligi mumkin.

Uslub va printsiplar ishlab chiqarishning oxirgi jarayonidagi mahsulotni olishni belgilab beradi. Ushbu mahsulotni olishga erishish uchun, ishni kim va qanday bajarish davommiyligi aniq belgilangan bo`ladi. Mahsulotni yaratish jarayonida turli xil ishni amalga oshirish uchun foydalanilishi mumkin bo`lgan asbob-uskuna vositalari texnologiya tarkibi uchun alohida ahamiyat kasb etadi. Asbob-uskuna vositalarining mavjudligi (yoki bo`lmasligi) tayyor mahsulot



ko`rinishida natijalar olish uchun zarur bo`lgan texnologik operatsiyalar ro`yxatini belgilaydi. Agar uni yaratish bo`yicha belgilangan barcha funktsiyalar amalga oshirilsa (asbob-uskuna vositalari yordamida ularni qo`llamasdan), amalda o`sha buyumni olish texnologiyasini ishlab chiqish mumkin. Aksincha, ayrim funktsiyalar bajarilmasa yoki mavjud asbob-uskunalar bilan uni bajarish o`ta murakkab bo`lsa, u holda tegishli operatsiyani bajara oladigan asbob-uskunalar yaratish vazifasi qo`yiladi bunday texnologiyani yaratish imkoniyati yo`qligi haqida qaror qabul qilinadi. Moddiy ishlab chiqarish texnologiyasi deganda tayyorlash, qayta ishlash vositalari va usullari orqali belgilanadigan xom-ashyo, material holati, xususiyati va shaklining o`zgarish jarayoni tushuniladi. Texnologiya moddiy mahsulot olish maqsadida materialning sifati boshlang`ich holatini o`zgartiradi (1.1-rasm).



1.1-rasm. Axborot texnologiyasi moddiy resurslarni qayta ishlash texnologiyasining analigi sifatida

Axborot shuningdek, resurs ham hisoblanadi. Uni qayta ishlash jarayonini xuddi moddiy resurslarni qayta ishlash jarayoni kabi texnologiya sifatida qabul qilish mumkin.

Axborot texnologiyasi - obyektning (axborot mahsulotining) holati, jarayon voqeaning yangi xususiyati to`g`risida axborot olish uchun ma`lumotlarni yig`ish, qayta ishlash va uzatish vositalari va usullari majmuidan foydalaniladigan jarayondir.

Moddiy ishlab chiqarish texnologiyasining maqsadi - inson tizimning ehtiyojini qondiruvchi mahsulot ishlab chiqarish sanaladi.

Axborot texnologiyasining maqsadi esa — axborot ishlab chiqarish bo`lib, uni tahlil etish va uning asosida biror bir harakatga qo`l urish uchun tegishli qaror qabul qilish hisoblanadi.

Ma'lumki, bitta va faqat o`sha moddiy resursga nisbatan har xil buyum mahsulot olish mumkin. Axborotni qayta ishlash texnologiyasiga nisbatan ham shunday bahoni bersa bo`ladi.

Moddiy va axborot texnologiyasining asosiy komponentlarini qiyoslash 1.1-jadvalda berilgan.

Axborotni yig`ish, uzatish, to`plash, qayta ishlash, saqlash, taqdim etish, foydalanish uslublari va usullari tizimi axborot texnologiyasi deb yuritiladi.

## Moddiy va axborot texnologiyasining asosiy komponentlarini qiyoslash jadvali

Texnologik komponentlar	
Moddiy mahsulot	Axborot mahsuloti
Xom ashyo va materiallar tayyorlash	Ma'lumotlar boshlang'ich axborotni yig'ish
Moddiy mahsulot ishlab chiqarish	Ma'lumotlarni qayta ishlash va yakuniy axborotga ega bo'lish
Istemolchilarga ishlab chiqarilgan mahsulotni sotish	Uning asosida qaror qabul qilish uchun yakuniy axborotni uzatish

**Zamonaviy axborot texnologiyasi** – shaxsiy kompyuterlardan keng foydalanishga, foydalanuvchilarning (dasturlash bo'yicha mutaxassis bo'lmaganlar) axborot jarayonida fail ishtirokiga, «do'stona» foydalanuvchi interfeysining yuqori darajada bo'lishiga, umumiy va muammo mazmunidagi amaliy dasturlar paketidan keng foydalanishga, EHM hisoblash tarmoqlari tufayli ma'lumotlarning uzoqdagi bazalariga kirib borish imkoniyatiga asoslangan texnologiyadir.

Zamonaviy axborot texnologiyalarini yaratishning uch asosiy tamoyillari quyidagilar:

1. Kompyuterli interaktiv muloqatli ish rejimi;
2. Boshqa dasturiy mahsulotlar bilan integratsiyalashish, o'zaro aloqa;
3. O'zgarish jarayonlarining ma'lumotlar va vazifaning qo'yilishi jihatidan moslashuvchanligi.

Zamonaviy axborot texnologiyalarini quyidagicha tasniflash mumkin: ta'minlovchi axborot texnologiyalari va funksional axborot texnologiyalari.

Ta'minlovchi zamonaviy axborot texnologiyalari turli vazifalarni hal etish uchun xilma-xil predmetli sohalarda asboblar majmui sifatida foydalanishi mumkin bo'lgan axborotni qayta ishlash texnologiyasidir. Ta'minlovchi axborot texnologiyalari vazifalarga yo'naltirilgan sinflarga nisbatan tasnif qilinishi mumkin.

Ta'minlovchi zamonaviy axborot texnologiyalari (TZAT) quyidagi komponentlardan tashkil topadi:

$$TZAT = IB + TS + P1$$

bu yerda,

IB- dasturlar qobig'i;

TS- hisoblash texnikasi (kompyuter);

P1- dasturlar qobig'ini to'ldirish va foydalanish bo'yicha qoidalar va cheklanishlar.

Ta'minlovchi zamonaviy axborot texnologiyalariga quyidagi dasturlar qobig'i kiradi:

- ✓ jadvalli protsessorlari;
- ✓ matn protsessorlari;
- ✓ gipermatn tizimlari;

- ✓ ekspert tizimlari;
- ✓ avtomatlashtirilgan ofis, qaror qabul qilishni qo'llab-quvvatlash tizimlari; statistik tizimlar;
- ✓ optimizatsion tizimlar.

Funksional zamonaviy axborot texnologiyalari (FZAT) ta'minlovchi zamonaviy axborot texnologiyalarining shunday modifikatsiyasini o'zida namoyon etadiki, unda predmetli texnologiyalardan birortasi amalga oshiriladi.

Ya'ni:

$$\text{FZAT} = \text{IB} + \text{TS} + \text{P2} + \text{P1} + \text{malumitlar}$$

Bu yerda, P2 - axborot texnologiyalarni realizatsiya qilishning qoidalari.

Funksional zamonaviy axborot texnologiyalarining asosi (bazasi) predmetli texnologiya hisoblanadi.

Chet el olimlari zamonaviy axborot texnologiyalari rivojlanishining quyidagi tendentsiyalarini keltiradi.

1. Axborot mahsulotlari ta'rifining o'zgartirishi bilan bog'liq, u ko'proq darajada hisoblash tahliliy ishining natijasi va kompyuterdan yakka tartibda foydalanuvchiga berilgan o'ziga xos xizmat o'rtasidagi munosabatga aylanmoqda.

2. Zamonaviy axborot texnologiyalari mantiqiy elementlarining parallel ravishda o'zaro ham bir-birini qo'llash qobiliyati, axborotning barcha turlari (matn, obzorlar, raqam, tovishlar) inson tomonidan sezgi organlari orqali bir vaqtda his qilishga yo'naltirishining birga qo'shilishi takidlanadi.

3. Axborot manbasidan to uning istemolchigacha bo'lgan yo'ldagi barcha oraliq bo'g'inlarini bartaraf qilinishi bashorat qilinadi. Masalan, muallim va o'quvchilar, sotuvchi va xaridorlar, asholachi va tinglovchilar, olimlarning o'zlari, mutaxassislarining korxonadagi bevosita muloqati videoanjumanlar tizimi, elektron do'kon, elektron pochta orqali amalga oshadi.

4. Yetakchi sifatida yo'ldoshli aloqa va umumjahon Internet global tarmog'idan foydalanish natijasida axborot texnologiyalarini globalashtirish tendentsiyasi davom etmoqda, u tufayli odamlar sayyoramizning istalgan nuqtasidan turib bir birlari va ma'lumotlarning umumiy bazasi bilan muloqat qilishlari mumkin.

5. Zamonaviy axborot texnologiyalari rivojlanishi jarayonining zamonaviy, oxirgi belgisi sifatida ko'rib chiqilmoqda, u moddiy ishlab chiqarish sohalari va axborot-kommunikatsiyalar biznesi o'rtasidagi farqlarning yo'qotilishi, firmalar va korporatsiyalar turlarini kattaroq diversifikatsiyalash, sanoatning turli xildagi tarmoqlari, moliyaviy sektor va xizmatlar sohasining o'zaro bir-biriga kirib borishidan iborat bo'ladi.

Zamonaviy axborot texnologiyalari rahbarlarga, mutaxassislarga, texnik xodimlarga axborotni qayta ishlash va qaror qabul qilishda, o'z vaqtida ishonchli va kerakli hajmda axborot olish, avtomatlashtirilgan ofislar tashkil etish, kompyuterlar va aloqa vositalarini qo'llagan holda tezkor majlislarni o'tkazish uchun mo'ljallangan zamonaviy axborot tizimlarini yaratish imkonini beradi.

Axborot texnologiyasi avtomatlashgan va an'anaviy (qog'oz) ko'rinishda amalga oshiriladi. Avtomatlashtirish hajmi va texnik vositalardan foydalanish turi aniq bir texnologiyaning mohiyatiga bog'liq.

**1.1. Avtomatlashtirilgan axborot texnologiyasi (AATex)** – boshqaruv vazifalarini hal etish uchun tizimli tashkil etilgan axborot jarayonlarini amalga oshirish usul va vositalari majmuidir. U hisoblash texnikasi va aloqa vositalaridan foydalanish asosida rivojlangan dasturiy ta'minotni qo'llash bazasida bajariladi.

Shunday qilib, avtomatlashtirilgan axborot texnologiyasi texnik vositalardan, ko'proq kompyuterlar, kommunikatsiya texnikalari, tashkiliy texnika vositalari, dasturiy ta'minot, tashkiliy – uslubiy materiallar va texnologik zanjirga birlashgan personaldan iborat bo'ladi. Ushbu harakat zanjiri axborotni yig'ish, uzatish, to'plash, saqlash, qayta ishlash, foydalanish va tarqatishni ta'minlaydi.

Demak, har qanday axborot texnologiyasining maqsadi - belgilangan tashuvchi vositada talab qilingan sifat darajasida kerakli axborotni olishdir. Ayni paytda axborotni qayta ishlash jarayonining tezkorligi va ishonchliligi, axborot resursidan foydalanish jarayonining ko'p mehnat talab qilishiga, ma'lumotlarni qayta ishlash qiymatiga nisbatan cheklashlar bor.

Avtomatlashtirilgan axborot texnologiyasini bir satir belgilarga, xususan, axborot tizimini avtomatlashtirishni amalga oshirish imkoniyati, avtomatlashtirilgan axborot texnologiyasining boshqaruv vazifalarini qamrash darajasi, texnologik operatsiyalar sinfi, foydalanuvchining interfeys turi, EHM tarmog'idan foydalanish variantlari va hokozo xususiyatlariga ko'ra tasniflash mumkin (1.2-rasm).

Avtomatlashtirilgan axborot texnologiyasini amalga oshirish usuliga ko'ra, an'anaviy va zamonaviy axborot tizimiga bo'linadi.

Boshqaruv vazifalarini qamrab olish darajasi bo'yicha avtomatlashtirilgan axborot texnologiyasining vazifalari ma'lumotlarni elektron usulda qayta ishlashni o'z ichiga oladi. Bunda EHM dan foydalangan holda ma'lumotlar qayta ishlash natijasida muayyan iqtisodiy masalalar hal etilib, boshqaruv faoliyatini avtomatlashtirish olib boriladi.

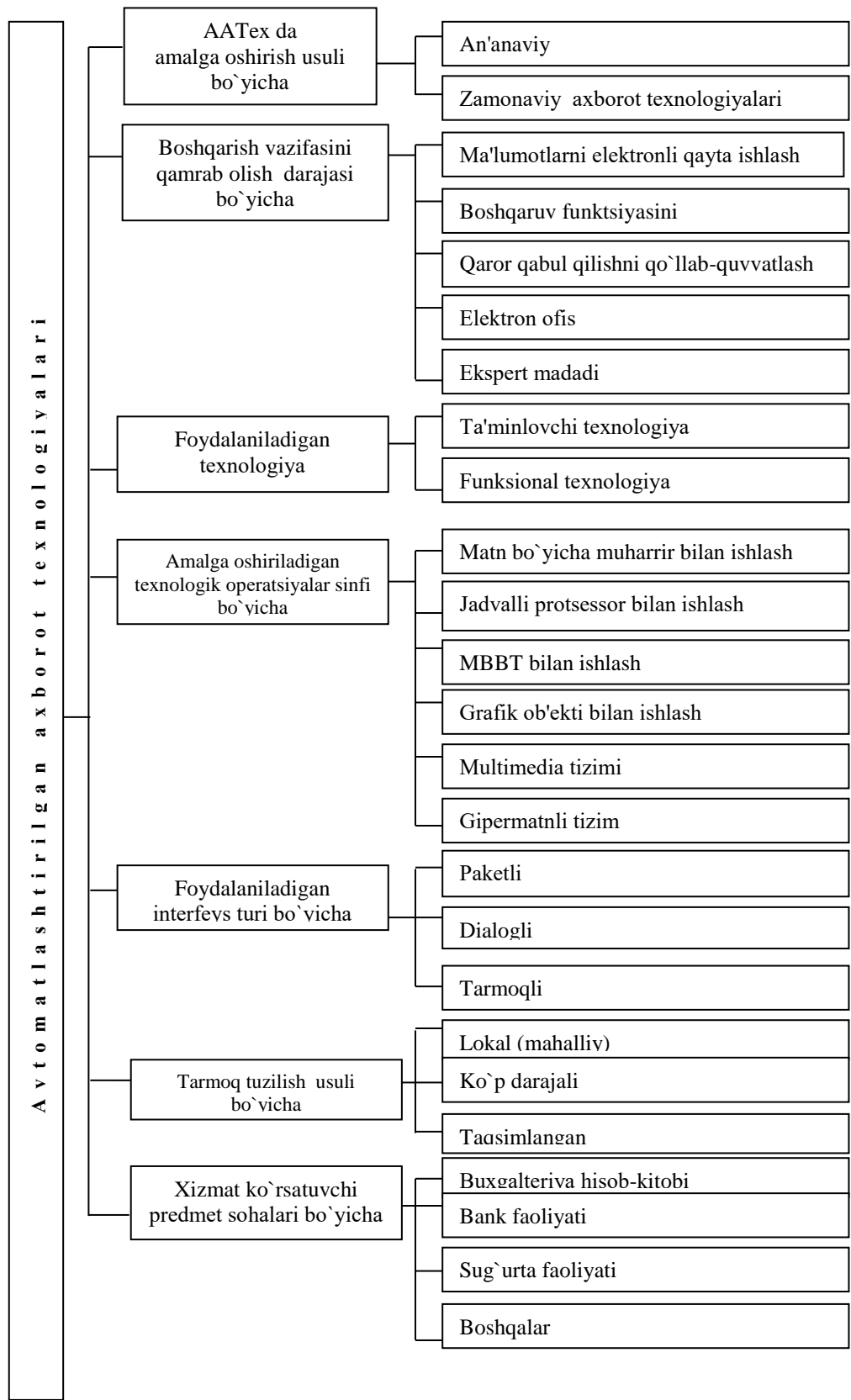
Boshqaruv faoliyatini avtomatlashtirishda boshqaruv qarorlarini tayyorlash uchun axborot – ma'lumotnoma rejimida ishlarni va doimiy hisobotni shakllantirish, xizmat vazifalarini kompleks ravishda hal etish uchun hisoblash vositalaridan (jumladan super EHM) foydalaniladi.

Ushbu guruhga qarorlarni qabul qilishni qo'llab-quvvatlash bo'yicha AATni ham o'qitish mumkin. U tahliliy ishlar va bashoratlarni shakllantirish, biznes-rejani tuzish, o'rganilayotgan jarayonlar, ishlab chiqarish-xo'jalik amaliyoti voqealari bo'yicha asoslangan baho va xulosalar chiqarish uchun iqtisodiy–matematik modellardan keng foydalanishni nazarda tutadi. Hozirda keng tadbir etilayotgan elektron ofis va qarorlar bo'yicha ekspert madadi deb nom olgan avtomatlashtirilgan axborot texnologiyasi ham mazkur guruhga mansub.

Axborotlashgan jamiyatning o'ziga xos jihatlari quyidagilarda namoyon bo'ladi:

- ✓ axborot iqtisodiyotining rivojlanishi;
- ✓ axborot tangligini bartaraf etish;
- ✓ axborot texnologiyasining globaligiga erishish;
- ✓ turli axborot resurslariga erkin kirib borilishi;
- ✓ axborot resurslarining ustunligini ta'minlash;
- ✓ yangi axborot texnikasi va texnologiyalarini keng qo'llash;
- ✓ boshqaruv faoliyatida axborotdan samarali foydalanish.

Axborotlashgan jamiyatda inson axborot bilan ishlash bo'yicha ma'lum darajadagi axborot madaniyatiga ega bo'lishi zarur. Buning uchun shaxsni axborotni tez qabul qilish va katta hajmini qayta ishlash, zamonaviy vositalar, usullar va texnologiyalardan foydalanishga tayyorlash lozim.



1.2-rasm. Avtomatlashtirilgan axborot texnologiyalari tasnifi

**Elektron ofis** muammo sohasidagi vazifalarni kompleks amalga oshirishni ta'minlaydigan ixtisoslashtirilgan dasturlar va axborot texnologiyalarini o'z ichiga oluvchi amaliy dasturlarning integratsiyalashgan paketi bo'lishini ko'zda tutadi.

Hozirda asbob-uskunalar va xodimlari turli binolarda joylashishi mumkin boʻlgan elektron ofislar keng joriy etilmoqda.

Hujjatlar, maʼlumotlar bazalari, aniq bir iqtisodiy obyekt yoki muassasa maʼlumotlari bilan uy sharoitida, mehmonxona yoki transport vositasida foydalanish virtual ofislarning avtomatlashtirilgan axborot texnologiyalari yuzaga kelgan.

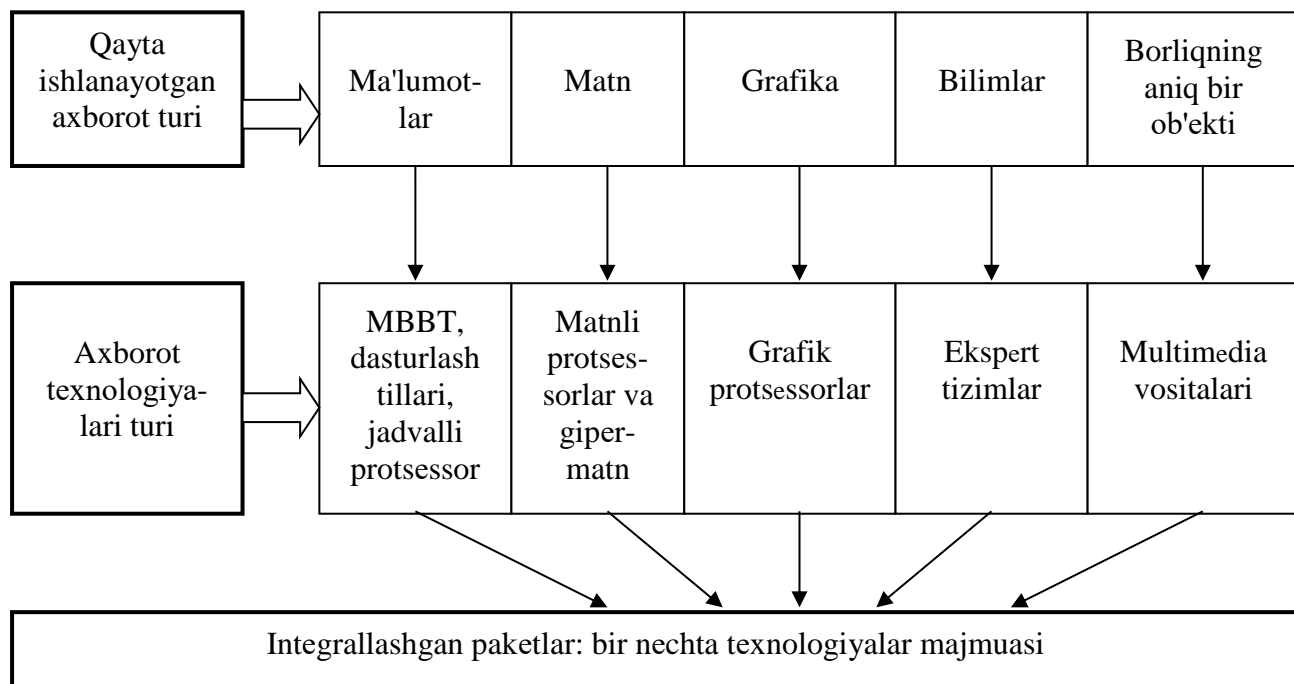
Bunday avtomatlashtirilgan axborot texnologiyalar hududiy yoki global tarmoqqa ulangan lokal (mahalliy) tarmoqda ishlashga asoslangan. Ana shunday abonent tizimi tufayli muassasa xodimlari qayerda boʻlishidan qatʼiy nazar umumiy tarmoqqa ulanish imkoniga ega boʻladi.

Foydalanilinish xususiyatlari boʻyicha texnologiyalar taʼminlovchi va funksional axborot texnologiyalari oʻzaro farqlanadi. Taʼminlovchi texnologiyalardan turli xil masalalarni hal etish uchun turli kredit sohasida vosita sifatida foydalansa boʻladi. Ular hal qilinadigan masalalar tartibiga koʻra ham tasniflanishi mumkin. Odatda mazkur texnologiyalar turli kompyuterlar va dasturiy muhitlarida bajariladi. Asosiy vazifa – ushbu texnologiyalarni yagona axborot tizimiga birlashtirishdir.

Funksional texnologiyalar – ayrim vazifalar, funksiyalarni avtomatlashtirishni taʼminlovchi texnologiyalar majmui sanaladi.

Amalga oshiriladigan texnologik operatsiyalar sinflari boʻyicha avtomatlashtirilgan axborot texnologiyasi mohiyatan, dasturiy jihatdan koʻrib chiqiladi va quyidagilarni oʻz ichiga oladi: matnni qayta ishlash, elektron jadvallar, avtomatlashtirilgan maʼlumotlar banki, grafik va ovozli axborotni qayta ishlash, multimedia va boshqa tizimlar.

Qayta ishlanadigan axborot va kompyuter axborot texnologiyalari 1.3-rasmda berilgan.



1.3.-rasm. Qayta ishlanadigan axborot va axborot texnologiyalari turlarining oʻzaro aloqasi

## **Axborotlarning grafikli ko`rinishi.**

**Kompyuter grafikasi** – bu EHM yordamida obyektlar modellari va ularning tasvirlarini yaratish, saqlash va qayta ishlash demakdir.

Kompyuter bilan matnli, grafik, audio va video axborot almashuvini dasturiy-texnik tashkil etish *multimedia–texnologiya* deb nom olgan. Bunday texnologiyani multimedia yordamida ega bo`lgan va kasb faoliyatida, o`quv, ta'lim, ilmiy ommobop, o`yin sohalarida foydalanish imkonini beruvchi maxsus dasturiy vositalar amalga oshiradi. Mazkur texnologiyani iqtisodiy faoliyatda qo`llash davomida kompyuterdan tasvirga ovoz berish, shuningdek, ular orqali inson nutqini tushunish, kompyuter orqali mutaxassisning ona tilida suhbat yuritish borasidagi imkoniyatlarga yo`l ochiladi.

**Foydalaniladigan interfeys** turi bo`yicha avtomatlashtirilgan axborot texnologiyasini foydalanuvchining axborot va hisoblash resurslariga kirib borish imkoniyati nuqtai nazardan ko`rib chiqish mumkin.

Shunday qilib, paketli avtomatlashtirilgan axborot texnologiyasi, axborotni qayta ishlash avtomat tarzda amalga oshirilayotgan paytda foydalanuvchining unga ta'sir ko`rsatish imkoniyatini yarata olmaydi. Bu shu bilan izohlanadiki, axborot qayta ishlashni tashkil etish operatsiyaning dasturiy jihatdan ketma-ketligi bajarolishiga bog`liq. Dialogli avtomatlashtirilgan axborot texnologiyasi paketli avtomatlashtirilgan axborot texnologiyasidan farqli ravishda, tizimda saqlanadigan axborot resurslaridan ma'lum bir vaqt davomida foydalanish bo`yicha cheksiz imkoniyatlar yaratib beradi. Ayni paytda tegishli masalalarni yechish va qaror qabul qilish uchun zarur bo`lgan barcha axborotga ega bo`ladi.

**Tarmoqli avtomatlashtirilgan axborot texnologiyasi** interfeysi aloqa vositalari rivojlangani bois aloqa kanallari orqali foydalanuvchiga hududiy taqsimlangan axborot va hisoblash resurslaridan foydalanish imkonini beradi. Bu esa uni ko`p funksiyali bo`lishiga va foydalanuvchilar o`rtasida keng tarqalishga sabab bo`lmoqda.

**Integrallashgan axborot texnologiyalari.** Hozirda turli xildagi axborot texnologiyalarini yagona kompyuter texnologiya kompleksiga birlashtirish tendentsiyasi kuzatilmoqda u integrallashgan axborot texnologiyalari deb yuritiladi.

Unda asosiy o`rinni egallovchi kommunikatsiya vositalari boshqaruv faoliyatini avtomatlashtirishda keng texnologik imkoniyatlarini ta'minlabgina qolmay, balki lokal, ko`p darajali, taqsimlangan, global hisoblash tarmoqlari, elektron pochta, integral xizmatning raqamli tarmoqlari kabi turli avtomatlashtirilgan axborot texnologiyasi tarmoq variantlarini yaratish asosi ham hisoblanadi. Bular bari obyektlarning ma'lumotlarini uzatish, qayta ishlash, to`plash va saqlash, himoya qilish qurilmalari orqali hosil qilinadigan obyektlar majmuasining o`zaro mantiqsiz, kutilmagan aloqalariga mo`ljallangan. Ushbu avtomatlashtirilgan axborot texnologiyasining tarmoq variantlari juda murakkab ma'lumotlarni qayta ishlay oladigan, ekspluatatsiya imkoniyatlari deyarli cheklanmagan, integrallashmagan kompyuter tizimlarini namoyon etadi.

Ma'lumotlarni qayta ishlashning integrallashgan kompyuter tizimlari murakkab axborot-texnologik va dasturiy kompleks sifatida loyihalashtiriladi. U



ma'lumotlarni taqdim etish va foydalanuvchilarning tizim komponentlari bilan o'zaro aloqalarining yagona usulini qo'llab-quvvatlaydi, mutaxassislarni ularning kasb ishida axborot va hisoblash ehtiyojini ta'minlaydi. Bunday tizimlar asosiy e'tiborni axborotni uzatish va qayta ishlash chog'ida ularning himoyasiga qaratadi. Axborotni himoya qilishning apparat-dasturiy usuli nisbatan keng tarqalgan. Xususan, axborotni uzatish va manzil bo'yicha yetkazib berish, abonentlarning umumiy foydalanish tarmog'ida (telefon, telegraf) ma'lumotlarni shifrlash va uni ochish jarayonida axborotning saqlanib qolishiga kafolat beradigan xususiyatlari bo'yicha tanlangan aloqa tizimlaridan foydalanish shular jumlasiga kiradi. Albatta, bunda foydalanuvchilar umumiy texnik vositalar, shifrlash algaritmlari va hokozolar borasida kelishib olishlari kerak.

Axborot almashinuvi va boshqaruvining tezkorligiga, xususan axborotni zudlik bilan qayta ishlashga nisbatan bo'lgan talabning kuchayishi nafaqat lokal, shuningdek bank, soliq, ta'minot, statistik boshqarishning ko'p darajali va taqsimlangan tizimlarini yaratishga olib keldi. Ularning axborot ta'minotini avtomatlashtirilgan ma'lumotlar banki amalga oshiradi. Mazkur ma'lumotlar bankida tegishli ko'p darajali iqtisodiy ob'ektarning tashkiliy-funksional strukturasi axborot massivlarini mashinaviy yuritishni hisobga olgan holda tuziladi.

Zamonaviy axborot-kommunikatsiyalar texnologiyalarida ushbu muammoni ma'lumotlarni qayta ishlashning taqsimlangan tizimi hal etadi. Bunda u ma'lumotlar bazalarining turli darajalari o'rtasida axborot almashinuvi uchun mo'ljallangan aloqa kanallaridan foydalanadi. Ma'lumotlar bazasini boshqarishning dasturiy vositalari murakkablashuvi hisobiga iqtisodiy hisob-kitob va boshqaruv qarorlarini ishlab chiqish chog'ida tezlik oshadi, axborotni muhofaza qilish va uning haqiqiyliги ta'minlanadi. Tashkiliy boshqaruvning ko'p darajali taqsimlangan kompyuter-axborot tizimlarida axborot bilan tezkor ishlash muammosini ham, boshqaruv qarorlarini ishlab chiqish va qabul qilish paytidagi iqtisodiy ahvolni tahlil qilish muammosini ham bir xilda muvaffaqiyatli hal etish mumkin.

Xorijiy mutaxassislar axborot texnologiyalari rivojlanishining beshta asosiy tendentsiyalarini ajratib ko'rsatadi:

**1.Axborot mahsulotlarining murakkablashuvi.** Axborot vositasi ko'rinishidagi axborot mahsuloti, ekspert ta'minoti xizmatining ma'lumotlar bazasi strategik ahamiyat kasb eta boradi. Turli shakldagi (nutq, ma'lumot, tasvir) axborot mahsulotlari eshitish, ko'rish va anglash uchun foydalanuvchining talabiga ko'ra ishlab chiqiladi hamda unga qulay vaqtda va shaklda mahsulotni yetkazib berish vositasi mavjud bo'ladi. Axborot mahsuloti borgan sari yakka foydalanuvchiga taqdim etiladigan o'ziga xos xizmat va hisobot-tahlil ishlari natijalari o'rtasidagi gibridga aylanib bormoqda.

**2.Birgalikda harakat qilish qobiliyati.** Axborot mahsulotining ahamiyati oshib borishi bilan mazkur mahsulotlarni kompyuter va inson yoki axborot tizimlari o'rtasida ideal tarzda almashuvini o'tqazish imkoniyati ilg'or texnologik muammo kasb etadi. Axborot mahsulotlarini qayta ishlash va uzatish muammosi ularning kelishi va tez harakatlanishi bo'yicha to'liq muvofiq bo'lishi lozim.

**3.Oraliq bo'g'inlarni tugatish.** Birgalikda harakatlanish qobiliyatining rivojlanishi axborot mahsulotlari almashish jarayonining taqimillashuviga, so'ngra,

axborot manbai yo`ldan istemolchiga qarab (ya'ni, bu sohadagi yetkazib beruvchi va istemolchilar) oraliq bo`g`inlar tugatiladi. Masalan, muallif va o`quvchi, sotuvchi va xaridor, qo`shiqchi va tinglovchi, o`qituvchi va o`quvchi yoki iqtisodiy obyektlarda mutaxassislar o`rtasida videokonferentsiya, elektron qisqa, elektron pochta tizimi orqali bevosita muloqat qilish imkoniyati tug`iladi.

**4.Globallashtirish.** Iqtisodiy obyekt yo`ldosh aloqa va Internet tarmog`idan foydalanib axborot texnologiyalari yordamida xohlagan joyda va xohlagan paytda ish olib borishi mumkin. Aynan Internet tufayli odamlar dunyoning har qanday nuqtasidan turib o`zaro muloqat qilish imkoniga ega. Bu holatda doimiy va yarim doimiy xarajatlar yanada keng geografik mintaqada taqsimlanish hisobiga ustuvorlikka ega bo`ladi.

**5.Konvergensiya.** Konvergensiya AATning zamonaviy rivojlanish jarayonining oxirgi bosqichi sifatida ko`rib chiqiladi. Bunda mahsulotlar va xizmatlar, axborot va dam olish, shuningdek, ovozli, raqamli hamda videosignallarni uzatish kabi ish rejimlari o`rtasidagi farq yo`qiladi. Moddiy ishlab chiqarish va axborot biznesi sohalari o`rtasidagi tafovut o`chib ketadi, firmalar va korporatsiyalarning faoliyat turlari diverfikatsiyasi, sanoat tarmoqlari, moliya sektori va xizmat sohalari o`zaro uyg`unlashib ketadi.

Shunday qilib, yangi axborot texnologiyalari – bu dunyo miqiyosida jamiyat taraqqiyotining sanoat asridan axborot asriga qarab o`tish asosidir. Mazkur tendentsiyaning biznesda qo`llanilishi quyidagi o`zgarishlarga olib keladi:

- har bir ish o`rnida resurslar yetarli bo`lganda axborotni qayta ishlash uchun taqsimlangan shaxsiy (persinal) hisoblashlarni amalga oshirish;
- xabarlarini jo`natish uchun ish o`rinlari birlashganda kommunikatsiyaning rivojlangan tizimini yaratish;
- iqtisodiy obyekt axborot oqimiga ulanganda, moslashuvchan global kommunikatsiyalarga ega bo`lishi;
- elektron savdo tizimini yaratish va rivojlantirish;
- iqtisodiy obyekt integratsiyasi – tashqi muhit tizimidagi oraliq bo`g`inlarni bartaraf etish.

Zamonaviy integrallashtirilgan amaliy dasturlar paketlari o`z ichiga quyidagilarni kiritadi: matn tahrirlagichi, elektron jadval, ma'lumotlar bazasini boshqarish tizimlari, kommunikatsion (olovchi) modul va boshqalar.

Integrallashtirilgan paketlarga qo`shimcha modul sifatida fayllarning eksport-import tizimi, kalkulyator, kalendar, dasturlash tizimlarini kiritish mumkin.

Uslubiy yo`naltirilgan amaliy dasturlar paketlarining farqi shundaki, ularning algoritmik asosi masala yechimining qandaydir ma'lum bir iqtisodiy - matematik usuliga asoslangandir. Bunday amaliy dasturlar paketlariga quyidagi usullarni kiritish mumkin:

- matematik dasturlash (chiziqli, dinamik, va bish.);
- tarmoqli rejalashtirish va boshqarish;
- ommaviy xizmat ko`rsatish;
- matematik statistika.

Uslubiy yo`naltirilgan paketlar qo`llanish sohasidan qat'iy nazar, masalaning umumiy holdagi yechimini beradi. Uning namunasi sifatida chiziqli dasturlash va tarmoqli rejalashtirish paketlarini ko`rsatib o`tish joizdir. Masalan, tarmoqli rejalashtirish amaliy dasturlar paketlari korxonada, bo`lim, seksiya, laboratoriya ish rejalarini shakllantirish imkoniyatini beradi. U jadval shaklida ifodalanib, unda har bir ish qachon boshlanib, qachon tugashi, ularni amalga oshirish uchun qancha va qanday zaxiralar kerakligi ko`rsatiladi.

Muammoga yo`naltirilgan amaliy dasturlar paketlari - bu eng keng tarqalgan amaliy dasturlar paketlardir. Umumiy holda muammoga yo`naltirilgan amaliy dasturlar paketlar deb xalq xo`jaligi, fan va texnikaning biror bir sohadagi mavjud muammoni yechishga qaratilgan amaliy dasturlar paketlarga aytiladi. Bunday amaliy dasturlar paketlarning juda ko`p turlari ichidan sanoatda va nosanoat sohasida boshqarish funksiyasini avtomatlashtirishga qaratilgan amaliy dasturlar paketlarni ko`rib chiqamiz.

Hozirgi kunning eng dolzarb masalasi – integrallashgan axborot tizimlarni yaratish masalasidir. Uning asosiy maqsadini qisqacha qilib «Mijoz-server» (clientgo`server) deb aytish mumkin. Uning asosiy ma`nosi mijoz-foydalanuvchi uchun u istagan har qanday xizmat servis soni bajo keltirishdir. Shu nuqtai nazardan qaraganda, zamonaviy G`arb bozorida iqtisodiy ishlab chiqarish faoliyatini avtomatlashtiradigan juda ko`p amaliy dasturlar paketlari mavjud. Ularni shartli ravishda 4 guruhga ajratish mumkin.

Katta yoki o`rtacha korxonaning hamma faoliyatini avtomatlashtirishga mo`ljallangan umumiy vazifalarga qaratilgan integrallashgan amaliy dasturlar paketlar kompleksi. Bularga yuqori narxli ko`p funktsional mahsulotlar hisoblanuvchi quyidagilar kiradi : Rg`3 (SAP), IRACLE, ADP «Galaktika» va boshqalar.

Ma'lum turdagi ishlab chiqarishni boshqaruvchi ilovalar kompleksi. Ularga quyidagilarni kiritish mumkin : EDWARDS, BAAN, PRIZM va hokozo.

Ixtisoslashgan dasturiy mahsulotlar: MMDS, MES ishlab chiqarishni moslashuvchan qiluvchi, uni bozor talablariga moslashuvini tezlashtiruvchi, sexlar ishini nazorat qiluvchi amaliy dasturlar paketlar.

Mahsulot ishlab chiqarishni ta'minlovchi hamma jarayonlar buyum detallarini loyihalashdan boshlab to foydalanuvga tayyor mahsulotni olguncha bo`lgan paytining zanjirini boshqaruvchi amaliy dasturlar paketlar: ERP tizimlari va boshqalar.

Ko`pchilik muammoga yo`naltirilgan amaliy dasturlar paketlar kompleksining narxi juda qimmat (ba'zan 1 mln. dollardan ham yuqori). Lekin shunga qaramay, ko`pgina G`arb firmalari o`z ish jarayonlarini avtomatlashtirish maqsadida muammoga yo`naltirilgan kompleks amaliy dasturlar paketlardan unumli foydalanish yo`lidan bormoqdalar.

Nosanoat sohasidagi muammoga yo`naltirilgan amaliy dasturlar paketlarga qo`yiladigan talablar ham sanoat sohasidagi amaliy dasturlar paketlarga qo`yiladigan talablarga ko`pchilik hollarda mos tushadi: ko`p satxli integrallashgan tizimlarni yaratish.

Muammoga yo`naltirilgan amaliy dasturlar paketlar ma'lum bir sohadagi muammoni amalga oshirishga mo`ljallangan. Masalan, buxgalteriya hisobi amaliy

dasturlar paketi EHM da quyidagi ishlarni amalga oshirishga imkon beradi: ish xaqi hisobi, turli vedimostlarni shakllantirish, har bir xizmatchining har oyga hisob kvitantsiyasini chop etish va hokozo.

Shaxsiy EHM ning ko`pgina amaliy dasturlar paketlari ham muammoga yo`naltirilgan paketlar turkumiga kiradi. Paketning asosiy maqsadi biror muammoni yechishda foydalanuvchini EHM bilan «do`stona aloqa» muhitini yaratishdan iborat bo`lib, u muloqat shaklida tashkil etilib, uning tezligi foydalanuvchini qoniqtiradigan bo`lishi kerak.

Amaliy dasturlar paketlari alohida modullardan, yetakchi dasturdan ko`pgina «monitor» deb ataluvchi yetakchi dastur amaliy dasturlar paketlarni modullar ishini tashkil etadi. Foydalanuvchining ko`rsatmalariga ko`ra boshqaruvchi dastur kerakli modullarni tanlaydi va ishchi dasturni shakllantiradi. Ishchi dastur to`g`ridan-to`g`ri foydalanuvchi vazifasini bajaradi, chiqish sonlarini kerakli shaklda tayyorlaydi va ularni foydalanuvchiga tahlil uchun chiqarib beradi. Kerakli amaliy dasturlar paketlarni chaqirish va u yordamida kerakli axborotni qayta ishlash operatsion tizim boshchiligida amalga oshiriladi.

Ayrim muammo sohaga tegishli amaliy dasturlar paketlariga AI Diasoft firmasi mahsulotlari : buxgalteriya hisoboti, moliya menejmenti, huquqiy tizimlar va h.k. kiradi.

Moliyaviy menejment amaliy dasturlar paketlari firmalar faoliyatini moliyaviy rejalashtirish va tahlil qilish zaruriyati irqasida paydo bo`ldi. Shu sinfga mansub bo`lgan amaliy dasturlar paketlar EDIP (Tsentrinvest Soft), Alt moliyaviy (Alt), moliyaviy tahlil (Infi-Soft) FICCAL (Tsentrinvest Soft) larni Misol qolib ko`rsatish mumkin.

Huquqiy ma`lumotnomalar tizimlari amaliy dasturlar paketlari uzluksiz ravishda to`lib boruvchi katta miqyosidagi qonunshunoslik axborotlari bilan unumli ishlash uchun xizmat qiladi. Bunday amaliy dasturlar paketlarga «Konsultant Plyus», «Garant», «Pravo» va boshqalar kiradi.

Keng qamrovli hisoblash tarmoqlarining asosiy vazifasi foydalanuvchining hudud bo`yicha yoyilgan umumtarmoq zahiralariga ma`lumotlar bazasiga, axborot uzatishga va boshqa ishlarga qulay, puxta erishish imkoniyatiga ega bo`lishdir.

Misol tariqasida keng qamrovli Internet tarmog`ida ishlatiladigan standart amaliy dasturlar paketlarni keltirish mumkin:

- to`g`rilash va kiritish vositalari - Nets Cape, Internet Explorer;
- elektron pochta (Mail), masalan Eudira.

Bank faoliyatidagi xalqaro tarmoqlarda ma`lumotlarni tayyorlash va uzatish uchun quyidagi standart amaliy dasturlar paketlar keng tarqalgan: Swift, Sprint, Reuters.

Hisoblash jarayonini ma'muriylashtirishni tashkil etishni ta'minlash uchun 50 % dan ortiq dunyo tizimlaridagi EHM larning lokal va global tarmoqlarida Bay Netwirqs (AQSh) firmasining amaliy dasturlar paketlaridan foydalaniladi, ular ma`lumotlari, kommutatirlar, konsentratrlar, marshrutizatorlar, xabarlar grafiklarini ma'muriylashtirishni boshqaradi.

## 1.1.Axborotlarning dasturiy va grafik ta`minotlari.

Shaxsiy kompyuterda ishlash jarayonida foydalanuvchi faqat amaliy dasturlarni boshqarishni bilishi yetarli emas. Kompyuterda ishlash jarayonida foydalanuvchidan axborotni tashqi ta'sirlardan saqlash, mavjud disklardan samarali foydalanish, axborotdan ruxsatsiz foydalanishni cheklash, magnit disk defektlarini sozlash, dasturiy viruslar bilan kurashish va boshqa amallarni bajara olishi talab etiladi. Shu sababli ham quyida ko`pchilik hollarda dastur-utilitalar deb nomlanuvchi servis dasturiy ta`minot tartibiga kiruvchi dasturlar, ularning afzalligi va ulardan foydalanish haqida bayon etishni lozim topdik.

**Servis dasturiy ta`minot** - foydalanuvchiga EHM bilan ishlashda qo`shimcha xizmatlar taqdim etuvchi va operatsion tizimlar imkoniyatlarini oshiruvchi dasturiy mahsulotlar to`plamidan iboratdir.

Funksional imkoniyatlariga ko`ra servis dasturiy vositalarini quyidagilarga ajratish mumkin:

- foydalanuvchi interfeysining foydalanish imkoniyatlarini kengaytiruvchi dasturlar;
- ma'lumotlarni buzilish va qoidasiz kirishlaridan himoya qiluvchi dasturlar;
- ma'lumotlarni qayta ishlovchi dasturlar (arxivlash dasturlari);
- disk va tezkor xotira qurilmasi o`rtasida ma'lumot almashinuvini tezlashtiruvchi dasturlar;
- disklar ishini nazorat, tahlil qiluvchi va ularga xizmat ko`rsatuvchi (disklarni formatlovchi, disklarni bo`laklarga ajratuvchi, mavjud defektlarni sozlovchi va h.k.).

Tashkil etish va amalga oshirish usuliga ko`ra servis vositalar qobiqli, utilitlar va mustaqil dastur shaklida taqdim etilishi mumkin.

**Qobiq dasturlar** - biror bir dastur va foydalanuvchi o`rtasidagi qatlam yoki boshqa dastur ustida ustqurma bo`lgan dastur. Qobiq dasturlar foydalanuvchiga sifat jihatidan yangi interfeys taqdim etadi. Amaliyotda quyidagi operatsion tizim qobiqlari keng tarqalgan: Norton Commander (NC) - Symantec firmasi mahsuloti va uning «klonlari» Vilkiv Commander (VC), Dis Navigatir (DN), Far (File and archive manager). Bulardan tashqari grafik interfeysli operatsion tizim qobiqlari mavjud: Windows 3.x.

**Utilitalar** - foydalanuvchiga disklar va faylli tizimlar bo`yicha qo`shimcha xizmat ko`rsatuvchi dasturlar. Utilitalar ko`pincha quyidagi vazifalarni bajaradi:

- disklarga xizmat ko`rsatish;
- fayl va kataloglarga xizmat ko`rsatish;
- kompyuter zaxiralari to`g`risida axborot taqdim etish;
- kompyuterlarni viruslardan himoya qilish.

Hozirgi vaqtda eng keng tarqalgan utilitalar:

- Norton Utilities - Symantec firmasi mahsuloti;
- Checoit PRI Deluxe 2.0 - Touch Stine firmasi ishlab chiqqan;
- PC Tools for Windows 2.0;

- Norton Backup, Fast Back Plus - disklarda axborotlarni rezerv nusxalarini tayyorlovchi dasturlar;
- virusga qarshi (antivirus) dasturlari - Norton Antivirus for Win95 – EHM dagi axborotni viruslardan (axborot mikroblari) zararlanishiga qarshi va zararlanish oqibatlarini tugatishga mo`ljallangan (Kasperskiyning Antiviral Toolkit pro (AVP));
- kommunikatsion (olovchi) utilitalar - EHM o`rtasidagi axborot ayirboshlashni tashkil etish uchun mo`ljallangan;
- kompyuterni tashxis (diagnostika) qiluvchi dasturlar - o`z nomadan ko`rinib turganidek, bu utilitalar EHM ning hamma qurilmalarining normal ish faoliyatlarini nazorat qilish, xotira miqdori, uning ishlatolish, disklarning turlari kabi ishlarni tekshirishni amalga ishiradi.

## **Dasturiy mahsulotlar tavsifi**

Foydalanish xususiyati va foydalanuvchilar kategoriyalariga ko`ra barcha dasturlarni ikki guruhga - utilitar dasturlar va dasturiy mahsulotlarga ajratish mumkin.

Utilitar dasturlar shu dasturlarni ishlab chiqaruvchilar ehtiyojini qondirish uchun mo`ljallangan. Ular ko`pincha servis rolini bajaradi yoki keng tarqalish uchun mo`ljallangan masalalarni hal etish dasturlari bo`ladi.

Dasturiy mahsulotlar foydalanuvchilar ehtiyojlarini qondirish, keng tarqatish va sotish uchun mo`ljallangan.

Hozirgi vaqtda dasturiy mahsulotlarni ochiq tarqatishning boshqa variantlari ham mavjud, ular global va mintaqaviy kommunikatsiyalardan foydalanish bilan yuzaga keladi:

1. Freeware - erkin tarqatiladigan foydalanuvchining o`zi qo`llab - quvvatlaydigan bepul dasturlar, ushbu dasturlarga zarur o`zgartirishlar kiritish mumkin.

2. Shareware - notijorat (shartli-to`lovsiz) dasturlar, ulardan odatda to`lovsiz foydalanish mumkin. Bunday mahsulotlardan doimiy foydalanilganda muayyan summa badal (vznos, plata) to`lanadi.

Dasturiy mahsulot foydalanishga tegishli ravishda tayyorlanishi zarur texnik hujjatlarga ega bo`lishi, shuningdek davlat ro`yxati kodi mavjud bo`lishi lozim.

Dasturiy mahsulot - sanoat mahsulotining istalgan turi kabi realizatsiyaga tayyorlangan ommaviy ehtiyojni muayyan muammo masalasini hal etish uchun o`zaro bog`langan dasturlar majmuasidir.

Dasturiy mahsulot dasturlashtirishning zamonaviy vositalari qo`llangan holda loyiha ishlarini bajarish sanoat texnologiyasi asosida ishlab chiqiladi. Uning

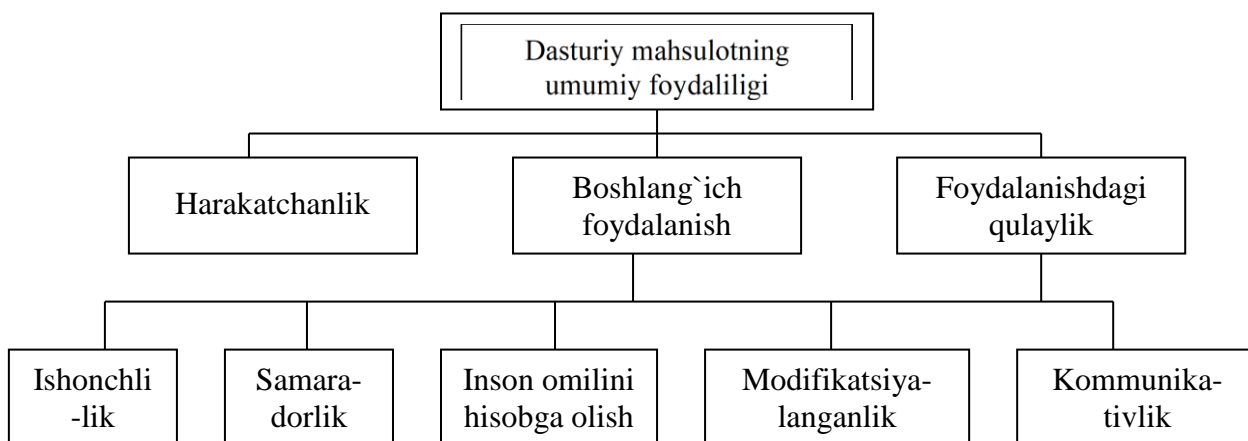
o`ziga xosligi axborot vositalaridan foydalanishni qayta ishlash xususiyatiga bog`liq holda algoritm va dasturlarni ishlab chiqish jarayonining noyobligidir.

Dasturiy mahsulotlarni tayyorlash (kuzatish)- dasturiy mahsulot ishga layoqatliligini qo`llab-quvvatlash, unga yangi versiyalar, o`zgartirishlar kiritish, topilgan xatolarni to`g`rilash va hokozolarni o`z ichiga oladi.

Dasturiy mahsulotlarning sifat ko`rsatkichlari xilma-xil, ular quyidagi jihatlarni aks ettiradi:

- dasturiy mahsulotdan qanchalik yaxshi (oddiy, ishonchli, samarali) foydalanish mumkinligi;
- dasturiy mahsulotdan qanchalik oson foydalanish mumkinligi;
- dasturiy mahsulotni qo`llashda sharoit o`zgarganda undan foydalanish mumkinligi yoki yo`qligi va boshqalar.

Dasturiy mahsulot xususiyatlari daraxt shaklida 1.2.1-rasmda keltirilgan.

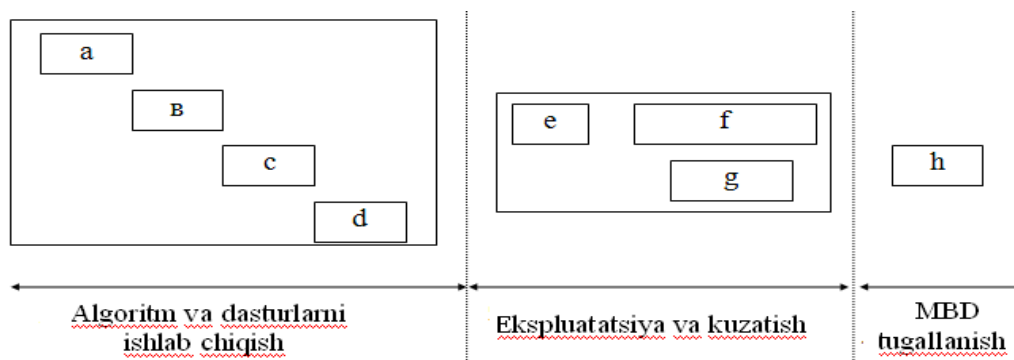


1.2.1-rasm. Dasturiy mahsulotlarning sifat xususiyatlari

Barcha dasturiy mahsulotlar o`zining mavjud bo`lish davriga (BBD) ega. Ular quyidagi bosqichlardan tashkil topadi:

- dasturiy mahsulotlarning marketingi, mahsulotlarga qo`yiladigan talablar spetsifikatsiyasini ishlab chiqish;
- dasturiy mahsulotlarning tuzilishini loyihalash;
- dasturlash, test o`tkazish, dasturni sozlash;
- dasturiy mahsulotni texnik va texnologik hujjatlar bilan ta'minlash;
- dasturiy mahsulotlar bozoriga chiqish, dasturiy mahsulotlarni tarqatish;
- dasturiy mahsulotlarni foydalanuvchi tomonidan ishga solish;
- dasturiy mahsulotlarni kuzatish;
- dasturiy mahsulotlarni savdodan olish, kuzatishni rad qilish.

Quyidagi 1.2.2-rasmda dasturiy mahsulotlarning mavjud bo`lish davri keltirilgan.



1.2.2-rasm. Dasturiy mahsulotlarning mavjud bo`lish davri

Murakkab iqtisodiy tizimlarda samarali boshqaruv ta'sirini ishlab chiqish uchun boshqaruvning tegishli algoritmlarini yaratish bilan bir satirda, turli-tuman axborotning katta hajmlarini qayta ishlab chiqish ham talab qilinadi. Iqtisodiyotda boshqaruv avtomatlashtirilgan axborot tizimlarini ishlab chiqish zaruriyati xuddi shundan kelib chiqqan.

Avtomatlashtirilgan axborot tizimlarining texnologik va funksional nuqtai nazardan ko`rib chiqishni bir necha tashkil qiluvchi elementlarga bo`lish mumkin. Texnologik nuqtai nazardan ko`rib chiqilganda avtomatlashtirilgan axborot tizimida boshqaruv apparati hamda texnik-iqtisodiy axborot, ularni texnologik ishlab chiqish usullari va vositalari o`zaro farqlanadi. Qolgan elementlar texnologik o`zaro big`langan, iqtisodiy-matematik usullar va boshqaruvning texnik vositalaridan yagona tizimli foydalanish sharoitida ma'lumotlarning avtomatlashtirilgan axborot texnologiyalarini tashkil qiladi.

Texnologik ta'minlanish iqtisodiy obyektlarni boshqaruvning avtomatlashtirilgan tizimlaridagi axborotli jarayonlarni EHM va boshqa texnik vositalar yordamida amalga oshiradi.

Texnologik ta'minlash metodik va tashkiliy-uslubiy materiallarni, EHM yordamida axborotni qayta ishlashning yagona texnologiyasini ta'minlovchi amallar va jarayonlarni qayta ishlash va joriy etishga mo`ljallangan tizimlar jamlanmasidan iborat. Ma'lum tartibda, aniq ketma-ketlikda bajariladigan amallar (protseduralar) majmuasi texnologik jarayonni tashkil etadi.

Texnologik jarayon usullari katta sondagi mumkin bo`lgan sharoitga bog`liq, ammo shunga qaramasdan, asosiy namunaviy amallarni ajratish mumkin.

Axborotni qayta ishlashning namunaviy texnologik jarayoni deyilganda, qo`yilgan masalaning ratsional yechilishini ta'minlaydigan, funksional tugallangan, qaytariladigan amallar to`plami tushuniladi.

Ratsional ishlab chiqilgan texnologik EHM lardan foydalanishda maksimal operativlik, samaradorlikni ta'minlashi lozim. Har bir berilgan holda texnologik jarayonni qurishning optimal variantini tanlash kerak. Texnologik ta'minotning namunaviy jarayonlari va amallari (protseduralari) quyidagilardan iborat:



- axborotni yig`ish va ro`yxatdan o`tkazish;
- axborotni tayyorlash;
- axborotni kiritish;
- qayta ishlash;
- axborotni uzatish;
- axborotni to`plash, jamlash va saqlash;
- bilimlarni taqdim etish.

Axborot texnologiyalari tayanch jarayonlari deb ma'lumotlarni qayta ishlash va yig`ish, ma'lumotlar bilan almashish va bilimlarni taqdim etish yoki formallashtirish mumkin bo`lgan va shuning asosida EHM va aloqa vositalari yordamida avtomatlashtirilgan jarayonlar tushuniladi. Axborot jarayonlarini avtomatlashni axborot texnologiyasining yaxlitligi kabi uchta darajasini keltirish mumkin: kontseptual, mantiqiy va fizik. Ammo axborotni ma'lumotlarga o`zgartirishdan oldin to`planishi, mos ravishda tayyorlanishi va shundan keyin EHMga kiritish mumkin.

Axborotni yig`ish va ro`yxatdan o`tkazish jarayoni deb bajarilayotgan operatsiyalar haqidagi boshlang`ich ma'lumotlarni birlamchi hujjatlarga tushirish va birlamchi hujjatlardan umumiy hujjatga ko`chirish protseduralariga aytiladi.

Axborotni yig`ish uni ro`yxatdan o`tkazish bilan kuzatiladi: hujjatda, mashinali tashuvchilarda eng keng tarqalgan axborot tashuvchi bo`lib qog`ozli hujjatlar hisoblanadi. Shu paytgacha birlamchi hujjatlarga axborotni yozish asosan qo`lda bajariladi, ammo axborotni qayd etish yetarlicha mehnat talab etuvchi protseduralar bo`lib qolmoqda.

Shuning uchun dastlabki hujjatlarni to`ldirishni imkon boricha avtomatlashtirishga harakat qilinadi. Bunda korxonaning quyi bo`linmalarida kompyuterdan foydalaniladi. Kompyuter tarmog`i mavjud bo`lganda ushbu hujjatlar kommunikatsion kanallar bo`yicha qayta ishlash uchun avtomatlashtirilgan axborot tizimining axborot bazasiga uzatiladi.

Axborotni to`plash protsedurasi obyektidan kelib tushgan xabar beruvchi axborot bergan inson tomonidan qabul qilinib va hujjat shaklidagi axborot tashuvchilariga tushirishdan iborat.

To`plangan axborot tayyorlanishi lozim, chunki kompyuterda joylashtirilgan muammoli soha modeli, kiritilishi lozim bo`lgan axborot tuzilishi va tashkil etilishiga o`z chegaralarini qo`yadi.

Tayyorlangan va kiritilishi lozim bo`lgan axborotni nazorati xatolar oldini olish, aniqlash va tuzishga qaratilgan. Har qanday xato kiritiladigan axborotni buzilishiga, demak qayta ishlash natijalari noaniqligiga hamda shuning hisobidan tizimni boshqarishda kamchiliklarga yo`l qo`yilishiga olib boradi. Axborot va ma'lumotlar to`liqligi va aniqligini nazorat qilish protseduralarini ko`zdan kechirish, mantiqiy va arifmetikdan iborat. Ko`zdan kechirish uslubidan ma'lumotlarni to`plash va tayyorlash bosqichlarida foydalaniladi va qo`lda bajariladi. Mantiqiy va arifmetik nazorat avtomatlashtirilgan uslub bo`lib, ma'lumotlarni o`zgartirish bosqichida qo`llaniladi.

Ko`zdan kechirish uslubida hujjatning to`liqligi, aktualligi, javobgar shaxslar imzosi va h.k. tekshirish maqsadida ko`rib chiqiladi.

Nazoratning mantiqiy uslubi asl ma'lumotlarni me'yoriy yoki qayta ishlangan ma'lumotlar bilan taqqoslaydi, funksional bog`liq ko`rsatkichlar bo`yicha mantiqiy qarama-qarshi bo`lmaslik bo`yicha nazariy tahlil olib boriladi.

Arifmetik uslub jadval shaklidagi hujjatlarni satir va ustunlar yig`indisi, formula bo`yicha, murakkab yoki tub sonlar belgilari, balans uslubi, qayta kiritish va shunga o`xshashlarga asoslanib nazorati tushuniladi. Tasodifiy yoki ataylab axborot buzolishining oldini olish uchun tashkiliy va maxsus tadbirlar o`tkaziladi.

Bu axborotni to`plash, tayyorlash va kiritishga mas'uliyatli xodimlar huquq va majburiyatlaridan iborat. Unga kiritishni avtomatlashtirilgan protokollari va axborot texnologiyasi muhitiga ruxsatli kirishni ta'minlash vositalari kiradi.

Hozirgi davrda dastlabki axborotni to`plash, tayyorlash va boshlang`ich nazoratga mas'ul xizmatchilari joylaridagi kompyuterlarda amalga oshirilmoqda, demak tayyorlash va kiritish fazalari birlashib ketadi.

Kompyuterga axborotni kiritish oldingi amalga oshirilgan protseduralarga ko`p jihatdan bog`liq. Hujjatlarda qayd qilingan axborot asosan qo`lda kiritiladi, bir turdagi axborotni ko`p marotaba kiritish tizimlarida skaner qurilmalaridan foydalanish keng tarqalgan. Aloqa kanallari orqali uzatilgan axborot kompyuter xotira qurilmasida qayd qilinadi va saqlanadi. Kiritish jarayoni bajarilganda korxonaning axborot bazasida ma'lumotlarning nazorati, dastlabki qayta ishlanishi va qayd qilish protseduralari zimmasiga yuklatiladi.

Iqtisodiy axborotni qayta ishlash natijaviy axborotga erishish maqsadida dastlabki ma'lumotlar ustida mantiqiy va arifmetik amallarni bajarishni ko`zda tutadi. Arifmetik amalga – algebraik qo`shish, bo`lish, ko`paytirish va boshqalar kirsa, mantiqiy amallarga – ma'lumotlarni solishtirish, birlashtirish, ma'lumotlar farqi va shunga o`xshash amallar kiradi. Axborotni qayta ishlash ishlab chiqilgan alqoritm asosida belgilovchi qabul qilingan qoidalar to`plami kiradi.

Axborotni qayta ishlash jarayonida joriy hisobotlar va belgilangan hisobotni berishda tasvirlangan natijaviy ko`rsatkichlar shakllantiriladi va monitor yoki chop qilish qurilmasida tasvirlanadi.

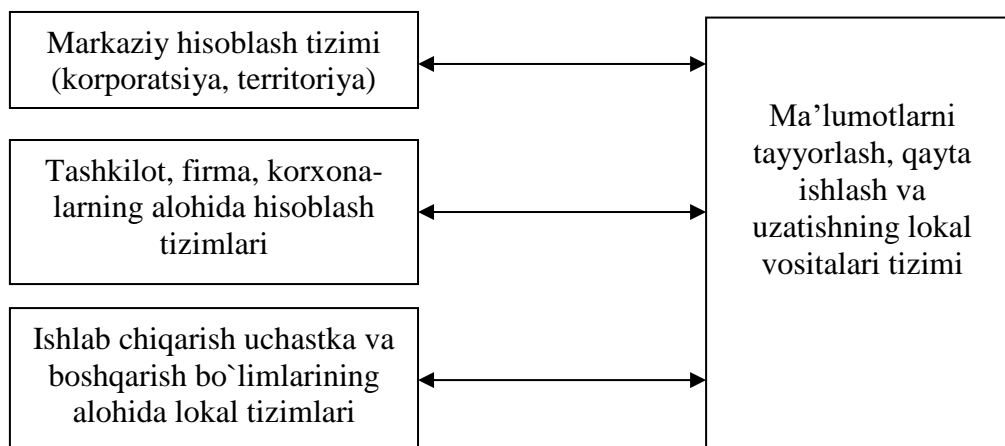
Axborotni qayta ishlash jarayoni bajariladigan amallarni nozorat qilish va ma'lumotlarni tuzatish bilan kuzatiladi. AAT (Avtomatlashtirilgan axborot tizimlari) da ma'lumotlarni tuzatish kiritish protsedurasi bajarilishida va uni tugatilganidan so`ng amalga oshiriladi. Ushbu jarayon bajarilishida o`tgan davr ma'lumotlarini ham tuzatishga e'tibor qaratalishi muhim ahamiyat kasb etadi.

Axborotni uzatish turli usullarda bajarilishi mumkin. U odatdagi aloqa vositalari – kurer, pochta orqali yuborish, transport vositalarida yetkazish orqali amalga oshirilishi mumkin. Hozirgi vaqtda axborotni uzatishda zamonaviy masoifaviy aloqa vositalari – elektron pochta, faks, fizik aloqa kanallari – telefon, optik tola, radiokanal, shu jumladan, sputnikli aloqalardan keng foydalanish imkoniyatlari ochilgan. Kommunikatsion tizimlar yordamida axborotni masofaga uzatish vositalari doimiy rivoj topib bormoqda. Axborotni uzatish uslubi xududiy taqsimlangan AAT firma obyektlarining uzoq masofada joylashgan filiallarini boshqarishda alohida o`rin tutadi.

Axborotni saqlash va jamlash kompyuterga kiritilgan ma'lumotlarni qayta ishlash uchun shu zahoti foydalanilmaydi. Ular axborot bazasida jamlanadi va saqlanadi. Axborot bazasida nafaqat birlamchi ma'lumotlar saqlanadi, balki turli masalalarni yechishda ko'p marotaba foydalaniladigan shartli-doimiy, doimiy, ma'lumotnoma va boshqa turdagi axborot ham saqlanadi. Boshlang'ich axborotni saqlash va jamlash qandaydir vaqt oralig'idagi olinadigan ma'lumotlar ehtiyoji bilan bog'langan. Kompyuter tizimining yana bir tomoni undan foydalanilgan vaqtdan boshlab kiritilgan axborotni saqlashga moyillik o'tgan davr natijaviy axborotini shakllantirish imkonini yaratadi.

Axborotni saqlash bilan ma'lumotlarni izlash prosedurasini ham bog'lash mumkin, ya'ni foydalanuvchi talabiga mos keluvchi axborotni tanlash prosedurasini.

Texnologik jarayonni qurishni avtomatlashtirilayotgan iqtisodiy obyektning tashkiliy tuzolishiga muvofiq boshqaruvning ierarxiya darajasi bo'yicha ularni tarqatishini hisobga olgan holda amalga oshirish maqsadga muvofiqdir. Ma'lumotlarni ishlab chiqish texnologik jarayonning uch bosqichli global tizimi quyidagi 1.2.3- rasmda keltirilgan.



1.2.3-rasm. Texnologik jarayonni ko'p bosqichli tashkil qilishning asosiy chizmasi

*Birinchi bosqich* - bitta yoki bir nechta kuchli shaxsiy kompyuter yoki meynfreymnlarni o'z ichiga oluvchi xududiy yoki idoralarning markaziy hisoblash tizimi. Uning asosiy vazifasi – umumiy iqtisodiy va moliyaviy nozorat, boshqaruv xodimlariga axborotli xizmat ko'rsatishdir.

*Ikkinchi bosqich* - korxonalar (birlashmalar), tashkilotlar va firmalarning hisoblash tizimlari, ular meynfreymlar, kuchli ShEHM larni o'z ichiga oladi, ma'lumotlarni ishlab chiqish va tartibiy birlik doirasida boshqaruvni ta'minlaydi.

*Uchinchi bosqich* - pastki darajadagi ishlab chiqarish uchastkalariga xizmat ko'rsatuvchi, ShEHM bazasi mahalliy taqsimlangan hisoblash tarmoqlari. Har bir uchastka ShEHM bilan jihozlangan, amalda bu belgilangan muammo soha doiralarida vazifaviy hisoblash tadbirlarini bajaruvchi avtomatlashtirilgan ish joyi bo'lishi mumkin.

**Ma'lumotlarni grafik usulida qayta ishlash texnologiyalari**-Zamonaviy axborot texnologiyalarining eng rivojlanib borayotgan sohalaridan biri - bu kompyuter grafikasi vositalaridir. Kompyuter grafikasi nafaqat iqtisodiyot masalalarini hal qilishda, balki xalq xo'jaligining boshqa sohalarida ham keng qo'llanilmoqda.

«Kompyuter grafikasi» tushunchasi adabiyotlarda juda keng talqin qilinadi. Xorijiy adabiyotlarda bu bilan bir satirda yana mashina grafikasi (inglizcha «machine graphics») va mikrokompyuter grafikasi (inglizcha «microcomputer graphics») atamaları ham qo'llaniladi.

Kompyuter grafikasini passiv (sust) va interaktiv guruhlarga bo'lish avvaldan ma'lum. Bundan bir necha o'n yil avval olimlarning, san'at ustalarining, multiplikatsion filmlar qahromonlarining suratlari alfavitli - raqamli chop etuvchi qurilmalardan chiqarish keng tarqalgan edi. Bunda rasm ma'lum nuqtalarga bir xil belgilarni tushirish orqali, rang tafovut esa ba'zi bir joylarda shu belgilarni bir necha marta tushirib (och rang uchun) mashina qog'ozi satxida hosil qilinadi. Tasvirni bunday usulda hosil qilish passiv kompyuter grafikasiga tegishlidir.

Interaktiv kompyuter grafikasi bu shundayki, bunda tasvirning holati, uning shakli, mazmuni, o'lchamlari va rangi displey ekranida interaktiv qurilmalar yordamida dinamik ravishda uzluksiz o'zgartirilib, boshqarib turiladi.

Zamonaviy shaxsiy kompyuterlarda hosil qilinadigan grafiklar amaliy jihatdan qaraganda hammasi interaktivdir.

Grafik tasvirni hosil qilishni zamonaviy usullari bilan yaxshiroq tanishish uchun kompyuter grafikasining usullarini ko'rib chiqamiz.

Vektorli funksional kompyuter grafikasi - bu usulda vektorli displeylardan foydalaniladi. Vektorli qurilmalarda xotirlovchi elektron nurli trubka qo'llanilib, nur ekran bo'yicha berilgan traektoriya bo'yicha bir marta yugurib o'tadi va shu holat ikkinchi buyruq kelmaguncha trubka xotira qurilmasida saqlanib qoladi.

Displey ekranidagi tasvir foydalanuvchi ko'ziga qo'zg'almaydigan bo'lishi uchun nur piksellar bo'yicha ko'p marta (sekundiga 16 marta) yugirib o'tadi.

Vektorli kompyuter grafikasida obyektни tasvirlash uchun matematik formulalar hamda kompyuter buyruqlaridan foydalaniladi. Bunda bazaviy element bo'lib chiziq hisoblanadi. Ushbu chiziqalar kombinatsiyalari har xil murakkab tasvirlarni yaratish imkonini yaratadi. Shuning uchun vektorli kompyuter grafikasini ko'pincha «obyektga yo'naltirilgan grafik», «chizish grafikasi» va ba'zi bir vaqtda «hisoblanadigan grafika» ham deyiladi.

Grafik tasvirlar yaratish va uni nomoyish etishda bugungi kunda ko'plab dasturlar ishlab chiqilgan bo'lib shulardan biri bu FLASH dasturi hisoblanadi bu dasturdan foydalanish vaqtida foydalanuvchilar o'zlariga kerakli bo'lgan ma'lumotlarni to'liq olishlari mumkin bo'ladi.

Flash texnologiyasi Shockwave Flash(SWF) vektor formatidagi grafiklardan foydalanishga asoslangan. SWF mualliflari tasvirlarni ifodalashni yuqori imkoniyatli taraflarini topa oldilar, grafika bilan ishlashni uskunaviy muhitini va natijalarni Web-sahifada aks ettirish mexanizmini ishlab chiqdilar. SWF ning qo'shimcha afzalligi undan bir qancha dasturiy-aparat platformalarida foydalanish

imkoniyatining mavjudligidir (xususan, *Macintosh* kompyuterlarida, *MacIS* operatsion tizimida va *Windows IS* bilan ishlovchi IBM kompyuterlarida). SWF ning yana bir asosiy afzallik tomoni: tasvirlar asosida anomatsiya yaratish bilan birgalikda unga interaktiv elementlarni va ovozlarni qo`shish imkoniyatidir.

Moslashuvchanlik va interaktiv multimedia dasturlarini yaratish mumkinligi Web dizaynerlar orasida SWF formatidagi dasturlarni ommaviylashishiga olib keldi. Shu sababli Macrimedia firmasoning SWF formati paydo bo`lishi bilan deyarli bir vaqtda ikki asosiy tarmoq brouzerlari (Internet Explorer va Netscape Cimmunicatir) uchun "Plug-In" komponenti yaratildi. Bu esa o`z navbatida SWF ni butun dunyo o`rgimchak to'rida yanada keng tarqalishiga olib keldi. Natijada bu ikki brouzer dasturchilari o`z mahsulotlari yadrosiga SWF ni qo`llab-quvvatlovchi qo`shimcha dastur kiritishga ahd qildilar.

Hozirgi vaqtda Macrimedia firmasi SWF formatini yaratuvchi dasturlarni yetarlicha qulay va sodda bo`lgan uskunalar va dasturiy vositalar bilan ta'minlashga erishdi. Macrimedia firmasi tomonidan ishlab chiqilgan ba'zi bir dasturiy vositalar: multimedia prezentasiyalarini yaratish uchun mo`ljallangan (Macrimedia Directir Shickwave Studio), grafik tasvirlarni tayyorlash uchun mo`ljallangan (Macrimedia Freehand va Macrimedia Firewirqs), interaktiv o`quv kurslarini yaratish uchun mo`ljallangan(Authirware va Macrimedia CiurseBuilder).

SWFni ko`pchilik qisqacha Flash deb yuritishga odatlangani sababli biz ham qulaylik uchun uni Flash deb yuritamiz. Demak Flash-texnologiya asosini quyidagilar tashkil qiladi:

- ✓ Vektorli grafika;
- ✓ Bir necha ko`rinishdagi anomatsiyalarni qo`llab-quvvatlash;
- ✓ Interaktiv elementlar interfeysini yaratish imkoniyati;
- ✓ Rastrli grafik formatidagi tasvirlarni import qilish imkoniyati;
- ✓ Sinhron tovishlarni qo`yish imkoniyati;
- ✓ Flash-filmlarni va Internetda ishlatiladigan ixtiyoriy grafik formatdagi tasvirlarni HTML formatiga eksport qilish;
- ✓ Platforma erkinligi;
- ✓ Web-brouzerda Flash-filmlarni avtonom rejimda ko`rish imkoniyati;
- ✓ Flash filmlarni yaratishda ko`pgina imkoniyatga ega bo`lgan mukammal

uskunaviy va dasturiy vositalarning mavjudligi va hokozo.

Hozirgi vaqtda liderlik holatida turgan aksariyat Web-sahifalar rastrli grafikadan foydalangan holda yaratilgan. Rastrli formatdagi grafik tasvirlar quyidagilardir: GIF(*Graphics Interchange Format* – ma'lumotlarni almashish uchun mo`ljallangan grafik format), JPEG(*Jiin Phitigraphic Experts GROUP* – Tasvir bo`yicha ekspertlar guruhini qo`shish) va PNG(*Pirtable Netwirq Graphics* – qo`shuvchan grafik format).

Rastrli grafikadan foydalanilganda, tasvirlar nuqtalar(piksellar) yig'indisi bilan ifodalanadi. Bu nuqtalarning bir qanchasi bir- biri bilan hech qanday bog`lanmagan, bu nuqtalarning har biri rang va kordinata talab qiladi. Oddiy holatlarda, qachon ikki xil rangli tasvirlardan(masalan, oq-qora rangli) foydalanilganda har bir piksel razryadi 2 bo`lishi yetarli bo`ladi: 0-qora, 1-oq.

256-rangli tasvirlarni ifodalash uchun har bir piksel kamida 8-razryadni talab qiladi ( $256-2^8$ ). Qiyinroq ifodalanuvchi fotografik rangli tasvirlarda har bir pikselning razryadi 24 yoki 32 gacha bo'lishi mumkin. Shu sababli rastrli tasvirlarda ranglar uyg'unligi va piksel razryadlarining soni ortishi fayl hajmini ham o'rtishga olib keladi. Rastrli tasvirlarni yana bir kamchiligi – tasvir (detallarning aniq va tiniq bo'lishi) sifati piksellar hajmiga va bu o'z navbatida monitoring ranglarga ruxsat berish chegarasiga bog'likligidir. Shu sababli tasvir ba'zi monitorlarda yaxshi aks etsa, ba'zi birlarida sifatsiz holatda chiqishi mumkin. Rastrli tasvirlar hajmini o'zgartirish ham- unchalik oson ish emas. Tasvir hajmini kattalashtirish uchun piksellar sonini ortirish kerak, kichraytirish uchun esa piksellar sonini kamaytirish kerak bo'ladi.

Vektorli grafika – umuman boshqacha texnologiyaga asoslangan.

Vektorli grafika – bu tasvirlarni rasmdagi joylashuvini matematik formulalar bilan aniqlanadigan egri chiziqlar yig'indisi sifatida ifodalash usulidir. Misol uchun, har qanday aylanani ifodalash uchun uch-to'rtta son yetarli, bular: radius, markaz koordinatalari va chiziq qalinligi. Shu sababli vektorli grafika rastrli grafikaga nisbatan bir satir afzalliklarga ega:

- matematik formulalar bilan ifodalanuvchi vektor grafikli tasvirlar piksellar bilan ifodalanuvchi rastrli grafik tasvirlarga nisbatan kompyuter xotirasidan o'zi joy egallaydi;
- tasvir (yoki uning alohida bir qismi) masshtabini uning sifatini yo'qotmagan holda o'zgartirish imkoniyatini chegaralanmaganligi;
- vektorli tasvirni hech qanday qiyinchiliksiz bir platformadan boshqasiga ko'chirish(moslashtirish) imkoniyati;

Lekin, vektorli tasvirlarda ham bazi bir kamchiliklar mavjud. Masalan, vektorli formatdagi fototasvirlarni yaratishni tasavvur qilish judayam qiyin. Bu muammoni hal qilish uchun Flash avtorlari shunday uskunalarini ishlab chiqishdiki, bu uskunalardan foydalanib, Web-sahifalarga nafaqat vektorli tasvirlarni, balki, rastrli tasvirlarni ham import qilib olish mumkin.

Flash da anomatsiya - xuddi multfilmlardagi kabi, obyektlar xususiyatini o'zgartirish orqali amalga oshirish natijasida yaratiladi. Masalan, obyekt tasvirda yo'qolishi yoki paydo bo'lishi mumkin, uning tasvirdagi joylashish holati, shakli, hajmi, rangi va hokozolari o'zgarishi mumkin.

Flash da obyektlarni harakatlantirish uning uchta turli xil mexanizmi orqali amalga oshiriladi:

- freymlarni ketma-ket qo'yish(“klassik”) anomatsiya. Bunda muallif yaratilayotgan “multik”ni har bir freymini o'zi yaratib chiqadi yoki boshqa dasturlardan import qiladi va ularni ketma-ket ravishda joylashtirib chiqishi natijasida anomatsiya hosil bo'ladi;
- avtomatik tarzda hosil qilingan anomatsiya (buni “tweened-anomatsiya” deb yuritiladi). Bunda muallif multiplikatsiyaning faqat birinchi va oxirgi freymlarini yaratadi, oraliqda joylashgan freymlarni Flash o'zi generatsiya qiladi(yaratadi). Tweened-anomatsiya ikki xil bo'ladi: obyektning joylashish holatiga asoslangan anomatsiya(motion animation) va obyekt

shaklini o'zgarishi (transformatsiya bo'lishi) ga asoslangan anomatsiya(shape animation);

- ssenariy asosida yuzaga keluvchi anomatsiya. Sssenariy obyekt bilan Flash ning dasturlash tili (ActionScript ) orqali muomala qilish natijasida yuzaga keladi. Flash dasturlash tilining tuzolishi Web-dasturlashda qo'llaniladigan boshqa dasturlash tillariga o'xshab ketadi (Masalan, JavaScript yoki VBScript).

Yuqorida keltirilgan mexanizmlarning har birini o'ziga yarasha yutuq va kamchiliklarga ega. Xususan, Tweened-anomatsiyaning e'tiborga molik tomonlari :

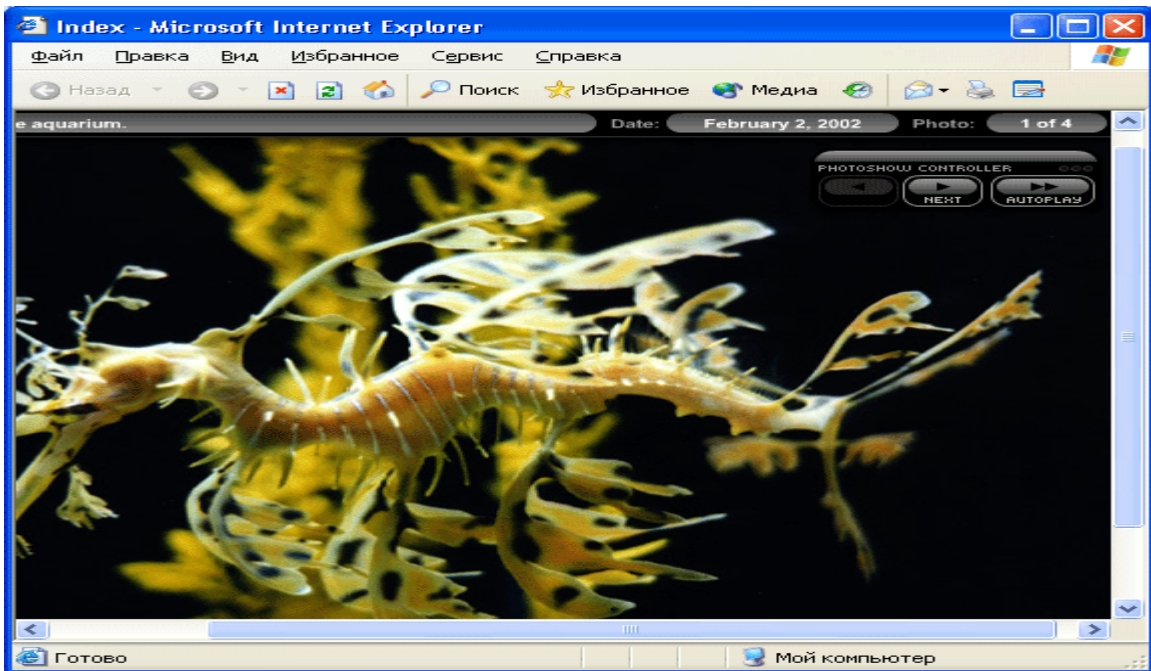
- muallif har bir kalit(bo'sh va oxirgi) freymni alohida-alohida yaratishi zarur;
- bunday anomatsiyalarni saqlashda Flash faqat birinchi va oxirgi freymlarni saqlaydi xolos, bu esa filmni hajmi kichkina bo'lishini ta'minlaydi.

Ammo, Tweened-anomatsiya sodda ko'rinishga ega bo'lgan, obyekt xususiyatlari tengdan o'zgaradigan syujetlarni yaratish uchungina qulay holos.

Ssenariy(ActionScript) yordamida ancha murakkab ko'rinishga ega bo'lgan syujetlarni yaratish mumkin, lekin... uchun ActionScript-tilini bilish talab etiladi. Flash yordamida anomatsiyali Web-sahifa yaratish imkoniyatlari bilan yaqindan tanishishni istaganlar uchun Disney kompaniyasining Web-sahifasiga([www.Disney.com](http://www.Disney.com)) murojaat etishlarini tavsiya qilamiz.

Flash - saytga bog'lanuvchilar klaviatura yoki sichqinchadan foydalanishlarini (masalan, filmning ixtiyoriy qismiga murojaat qilish, firmaga informatsiya qo'shish yoki biror obyekt qo'shish va h. ) nazarda tutgan holda interaktiv filmlar yaratish imkoniyatini beradi.

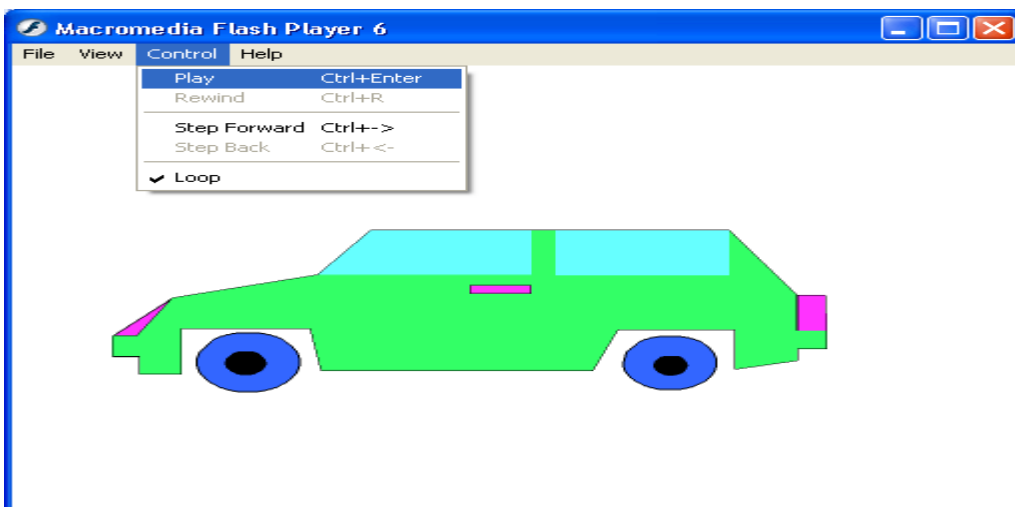
Interaktivlik – Flash-filmlarda kechadigan jarayonlar mos ravishda ActionScript da yozilgan buyruqlar ketma-ketligi (instruksiyalari) asosida amalga oshiriladi. Ya'ni, harakat bajariladi, qachonki u bilan bog'liq biror jarayon amalga oshrilsa. Bunday jarayonlar, xususan, filmning aniqlangan biror freymiga o'tish yoki foydalanuvchi tomonidan sahifadagi biror tugmani bosishi, klaviatura yordamida murojaati bo'lishi mumkin.



Rasm. 1.2.4. Flash da qilingan Web-sayt sahifalaridan birining ko`rinishi

Sahifaning yoki filmning biror bo`limida - tugmalar, film fragmentlari yoki ixtiyoriy biror freym – Interaktiv bo`lishi mumkin. Shu sababli ko`pgina standart harakatlarni ifodalshda bevosita ActionScript ga murojaat qilishga to`g`ri keladi. Standart harakatlar sifatida quyidagilarni nazarda tutilyapti: filmni yo`qlash va to`xtatish, ovozlarni o`chirish va yoqish, berilgan URL ga o`tish va h. Barcha standart harakatlarni o`ndan ortig`ini sanash mumkun .

Zarur paytlarda bir qancha shartlar asosida bajariluvchi murakkab ssenariylarni yaratish orqali kerakli natijaga erishiladi. Bunday ssenariy yaratish esa ActionScript da dastur tuzishda yuqori tajribaga ega bo`lishni talab qiladi. Quyidagi rasmda interaktiv filmga oid sodda Misoldan ko`rinish keltirilgan.



Rasm. 1.2.5. Interaktiv Flash-filmiga Misol



2.2. rasmda keltirilgan Flash-sahifada foydalanuvchi biror tugma yordamida uni boshqarsa yoki uning rangini va shaklini o`zgartira olsa, bu interaktiv filmga Misol bo`la oladi.

Flashda - filmlarda ovozdan foydalanishning bir qancha usullari mavjud. Siz o`z mahsulotingizga uzluksiz , film syujetiga va foydalanuvchi harakatiga bog`liq bo`lgan va bo`lmagan holda ishlovchi ovozli qism qo`shishingiz mumkin.

Muqobil variantlardan biri – anomatsiya va ovozni sinxrinlashtirishdir. Bundan tashqari Flash interaktivlikni oshirish uchun, tugmalarni ovozlar bilan bog`lash imkoniyatini beradi. Ovozlar bilan ishlashning yana bir varianti – bu ovozni ActionScript ssenariylari orqali boshqarishdir.

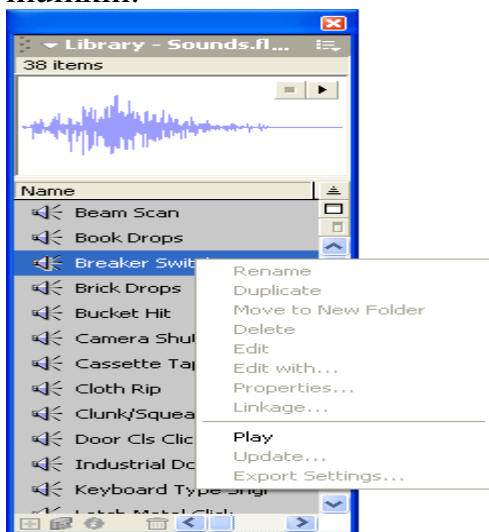
Flash da ovozlar bilan ishlashning ikki xil tipi mavjud: biror jarayon asosida boshqariladigan(event sound) va avlodli(uzluksiz) (stream sound).

Biror jarayon asosida boshqariladigan ovozlarda ma`lumot jarayonga mos ravishda to`liq yuklangan bo`lishi kerak, shundan keyingina ovoz yuklanadi va bu jarayon nihoyasiga yetmagunga qadar ovozli jarayon ham unga mos ravishda davom etadi. Avlodli tipda esa ovoz darhol, atigi birinchi bir nechta freymlar yuklangunga qadar boshlanib bo`ladi.

Ovozlar bilan ishlashda qulaylikni oshirish maqsadida Flash muallifga quyidagi imkoniyatlarni taqdim etadi:

- ovozlarni taqsimlab kutubxona yaratish, ya`ni bitta kutubxonadagi ovozdan bir necha filmlarda foydalanish imkoniyati;
- ovoz sifatini va ekspert qilinadigan film hajmini optimallashtirish maqsadida siqilgan formatdagi ovozli ma`lumotlarni parametrlarini boshqarish imkoniyati; bunda muallif filmdagi siqilgan ovoz parametrlarini alohida ovoz paramerini tanlagani kabi tanlay olishi mumkin.

Bundan tashqari, siz ovozga turli xildagi qo`shimcha effektlar qo`sha olishingiz mumkin.

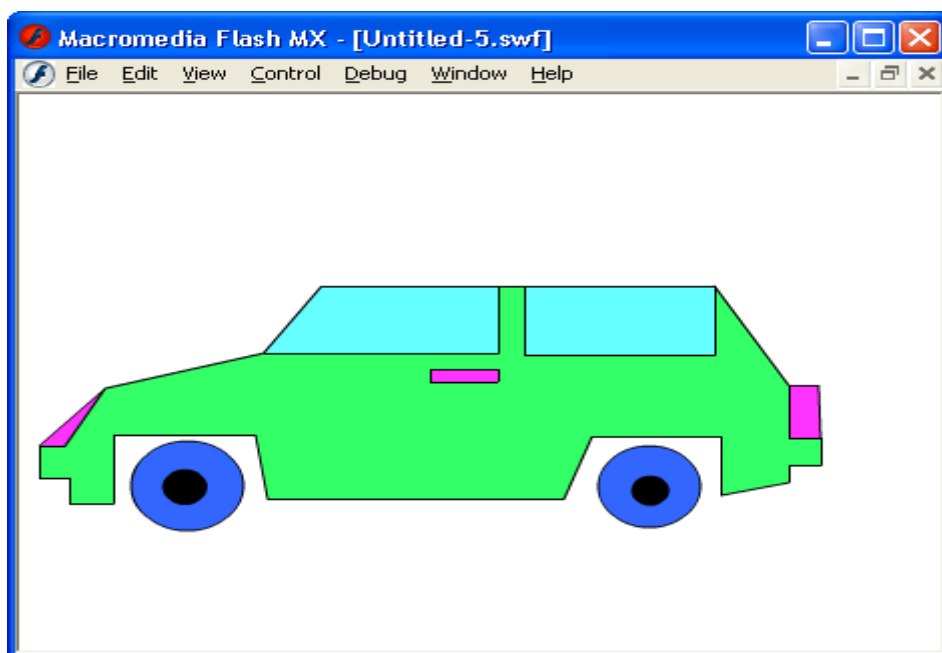


Rasm. 1.2.6. Ovoz parametrlari muharriri

## Flash filmlarni mustaqil ravishda ko`rish

Flash yordamida filmlarni yaratish va tahrirlashda muallif “ \*.FLA ” - kengaytmaga ega bo`lgan Flash formatidagi fayl ustida ishlaydi. Bu Flash ning ichki(xususiy) formati va u faqat shu formatdagi fayllarni tahrirlay oladi. Web-brouzerlar yordamida ko`rish uchun filmlar SWF formatda bo`lishi kerak. Biz yuqorida ta`kidlaganimizdek Web –brouzerlarning oxirgi versiyalari (Internet Explorer, Netscape va Ipera) aynan shu formatni maxsus ulangan Flash-playerlar yordamida qo`llab quvvatlaydi va ko`rsatadi. Shu Flash-playerlar - Flash tartibiga kiruvchi alohida dastur hisoblanadi va u yordamida SWF-fayllarni uning xususiy redaktori ochilmagan holda ham aktivlashtirish mumkin. Flash-player yordamida SWF-fayl yuklanganda muallif tomonidan qo`yilgan barcha funksional va interaktiv imkoniyatlar saqlanib qoladi.

Ikkinchi varianti Flash filmlarni “avtonom” holatda ko`rish uchun- “Universal priogratel”(Windows media player)dan foydalanish mumkin. Buning uchun Flash-film \*.AVI formatida ekspert qilinishi kerak. Lekin bunday ekspert qilish natijasida ekspert qilingan faylning hajmi bir necha marotaba kattarib ketadi.



Rasm. 1.2.7. Flash-player yordamida SWF-fayllarni ko`rish

\*.AVI formatda konvertasiya qilingan Flash filmlarning interaktivlik imkoniyati yo`qoladi, shu sababli bu variantda Flash filmlarni ko`rish unchalik qulay ahamiyat kasb etmaydi.

Flash filmlarni mustaqil ravishda ko`rishning yana bir usuli uni anomatsiyali \*.GIF formatda ekspert qilishdir. Lekin, bunda ham film o`z interaktivligini yo`qotadi.



Rasm. 1.2.8. \*AVI formatda ekspert qilingan Flash-filmni “Windows Media Player” yordamida ko`rish

### Flash – filmlarni grafik formatda konvertatsiya qilish to`g`risida

Flashdan “oddiy” grafik redaktor sifatida ham foydalanish mumkin. Bunda avval Flash yordamida tasvir yaratib olinadi, so`ngra uni boshqa bir grafik (vektorli yoki rastri) formatga ekspert qilinadi. Bu jarayon konvertatsiya qilish deyiladi.

Konvertatsiya qilish boshqa ba`zi grafik dasturlar tushuna oladigan grafik tasvirlarni Flash da ham yaratish mumkinligini beradi.

Quyidagi jadvalda Flash tushunadigan “qo`llab-quvvatlaydigan” ba`zi grafik formatlar keltirilgan:

Tabl 2.1.	Fayl tipi	Qengaytmasi	Ekspert	Import
	Adobe Illustratir (6.0 versiyagacha bo`lgan)	.eps, .ai	+	+
	Bitmap	.bmp	+	+
	Enhanced Windows Metafile	.emf	+	+
	FreeHand	.ft7, .ft7, .fig, J8, .fli9, .ft9	+	+
	GIF	.gif	+	+
	JPEG	.jpg	+	+
	PNG	.png	+	+
	Flash Player	.swf	+	+
	Windows Metafile	.wmf	+	+

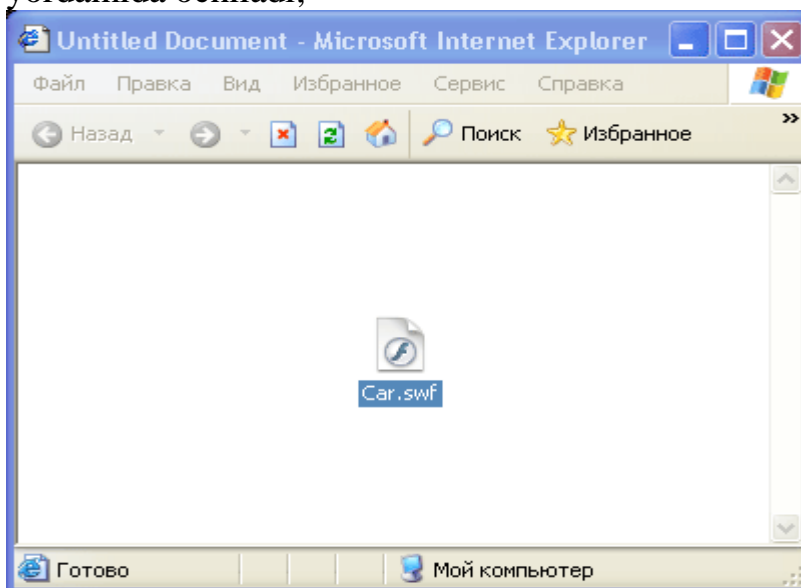
Qo`shimcha, agar kompyuterga QuiskTime 4 yoki uning oldingi versiyalari yoqlangan bo`lsa;			
PhitiShip	.psd	+	+
QuiskTime Image	.qtif	+	+
Solikin Graphics	.sai	+	+
TIFF	.tiff	+	+

### Flash va Web sahifa to`g`risida

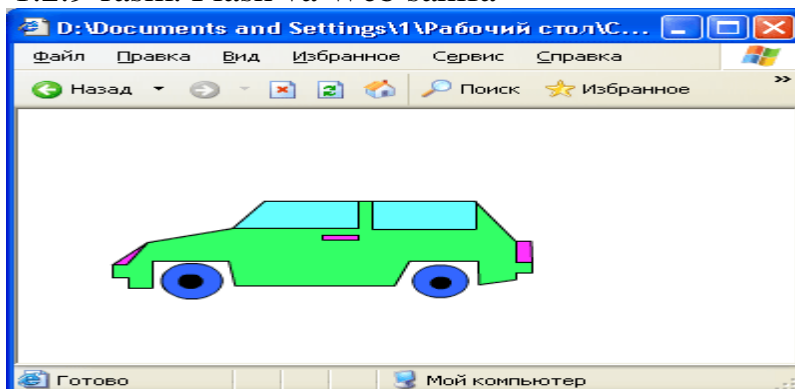
Turli holdagi Flash-filmlarni yaratish va ko`rish imkoniyatlari bo`lishiga qaramasdan, Flash-texnologiyadan bugungi kunda chiroyli va dinamik Web-sahifalarni yaratishda foydalanilyapti.

Flash-filmlarni brouzerlarda chiqarishni ta`minlashning 2 xil usuli mavjud:

- Flash-filmni SWF formatda saqlab, faylni sichqoncha yordamida u saqlangan oynadan brouzer oynasiga tashlanadi (1.2.9-rasm);
- Flash filmni HTML formatda eksport qilinadi, so`ngra uni brouzer yordamida ochiladi;



1.2.9-rasm. Flash va Web sahifa

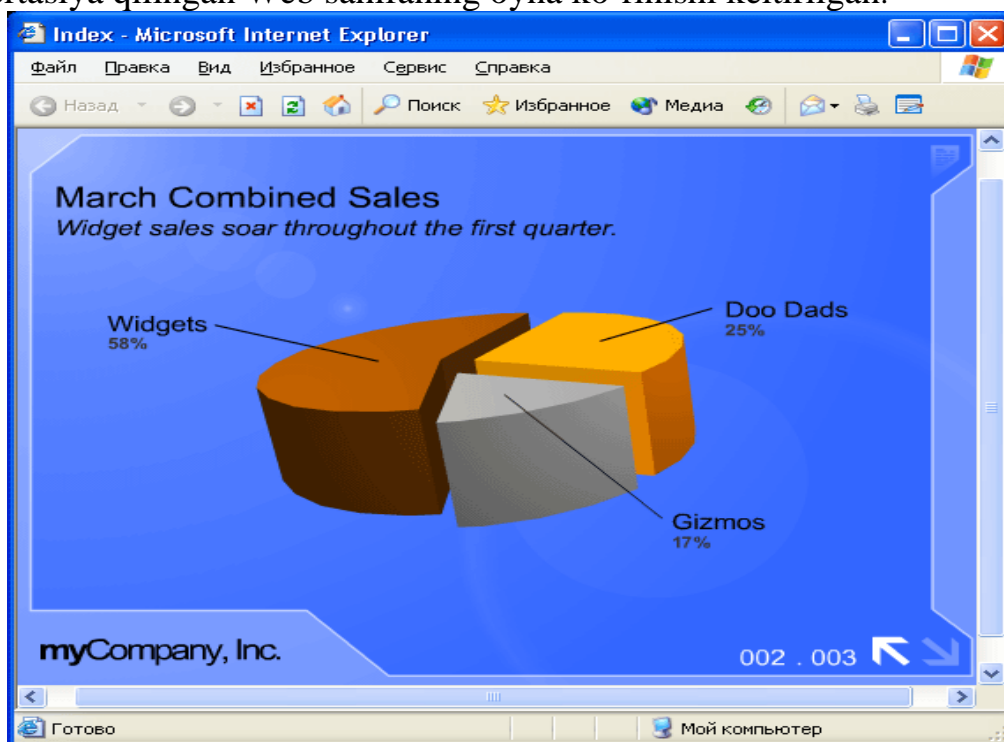


Rasm. 1.2.10. SWF formatdagi Flash-filmni brouzerga tashlab, uni ko`rish

Flash-filmlarni HTML ga o`qzash mexanizmi - ActiveX elementlaridan foydalanib yoki Plug-on komponentiga bo`lganish orqali amalga oshiriladi . Internet Explorer brouzeri uchun bu mexanizm <OBJECT> - tegi (HTML formatida xizmatchi siz) dan, Netscape Explorer brouzeri uchun <EMBED>- tegidan foydalaniladi. Har ikkala brouzer uchun ham umumiy bo`lishi uchun Flash dasturi filmni – HTML sahifaga konvertatsiya qilish vaqtida har ikkala tegni generatsiya qiladi (ishlatadi).

Flash-filmlardan Web-sahifa elementlari sifatida foydalanish variantlari ba`zilari quyida keltirilgan:

- sahifani estetik jihatdan chiroyli chiqishini ta`minlash maqsadida sodda ko`rinishga ega bo`lgan anomatsiyalardan foydalanish mumkin;
  - anomatsiyali tugmacha; bunday tugmachalardan gipersilka sifatida yoki qandaydir jarayonni faollashtirish maqsadida (masalan boshqa bir faylni yuklash, brouzerda yangi oyna ochish va h.) foydalanish mumkin.
  - Web-sahifaga bog`lanuvchilardan ma`lumot qabul qilish formasi sifatida;
  - Reklama bannerlari; 1.2.9 – rasmda reklamali bannerga Misol keltirilgan;
- Zarurat tug`ilganda Flash yordamida oddiy, sodda ko`rinishga ega bo`lgan HTML-sahifa yaratish mumkin. 1.2.10-rasmda Flashda tuzilib keyin HTML-kodga konvertatsiya qilingan Web sahifaning oyna ko`rinishi keltirilgan.



### **Flash-filmlar yaratishning umumiy sxemasi . Asosiy tushunchalar.**

Flash redaktor birinchi o`rinda Web-sahifa yaratishga mo`ljallangan bo`lishiga qaramasdan, u bilan ishlovchilar Web-sahifaning teg, atribut, sahifa kabi terminlariga kamdan kam duch kelishadi. Flashda filmlar yaratish texnologiyasida butunlay boshqa tushuncha va terminlar qo`llaniladi. Flashni yaxshi o`rganish uchun birinchi, unda qo`llaniladigan terminlar bilan tanishib chiqish maqsadga muvofiqdir.

Har qanday Flash-filmni yaratishda unga obyektli yondashishga to'g'ri keladi. Buni - filmning hamma elementi har xil tipli obyektlar kabi interpretatsiya bo'lishi va unda mavjud har bir obekt o'zining bir qancha amallarni bajaruvchi, bir qancha xususiyatlariga egaligi bilan tushuntirish mumkin. Masalan, "Текст" obyekti uchun simvollar o'lchami, rangi, shrift turi kabi xususiyatlari berilgan bo'lishi kerak. Yana "Текст" ustida tahrirlash, nusxa olish, qirqib olish, unga giperssilka qo'yish kabi amallarni bajarish mumkin. Xuddi shu kabi, grafik tasvirlarning ham, ovozlarning ham o'ziga hos xususiyatlari va uslublari mavjud.

Flash da ishlash paytida ba'zida "Obyekt" tushunchasi bilan birgalikda simvol (SyBBil) termini ham qo'llaniladi. Bu ikki termin orasidagi farq bilan tanishib chiqsak.

Simvolni - o'zining aniq bir xususiyatlariga ega bo'lgan obyektning bir shabloni sifatida tushunish mumkin. Simvol *maxsus kutubxona*(Library) da saqlanadi va undan nafaqat bitta, balki, bir nechta filmlarda foydalanish mumkin. Simvolning filmda joylashuvchi har bir nusxasi *simvol nusxasi* (Instance) deb yuritiladi. Simvol nusxasi o'zini asl nusxasida mavjud barcha xususiyatlarni o'zida saqlaydi va ular orasida bog'liqlik bo'ladi: agar simvolning asl nusxasidagi biror xususiyatga o'zgartirish kiritilsa qolgan barcha uning avlodlari(simvol nusxalari)ning ham shu xususiyatlari avtomatik tarzda o'zgaradi. Bunday yondashish film muallifining vaqti va kuchini tejashini, hamda film hajmini kichkina bo'lishini ta'minlaydi. Bundan tashqari, agar muallif filmning biror qismida simvol nusxasi xususiyatini o'zgartirishi kerak bo'lib qolsa, bu o'zgarishni simvolning asl nusxasiga hech qanday ta'sir ko'rsatmasdan amalga oshirish mumkin.

Flash-filmlarda dinamika qandaydir vaqt oralig'ida simvol nusxasini ba'zi xususiyatlarini(masalan, rangi, kordinatasini, o'lchamini va h.) va holatini o'zgarishini ta'minlaydi. Simvol nusxasining har bir holati filmning alohida *freym*(Frame)lari bilan bog'langan. Simvol holatini o'zgarishiga mos keluvchi freym – kalit freymi(Qeyframe) deb ataladi. Kalit freymi - foydalanuvchi o'zgartirishi mumkin bo'lgan ob'ekt husuyatlarini shu freymga to'g'ri kelishi yoki kelmasligini o'zi tekshiradi. Kalit freymlari uchun mo'ljallangan maxsus funksiya va buyruqlar bo'lib, ular haqida keyingi boblarda alohida to'xtalamiz. Filmida freymlarni almashtirish dinamikasi vaqt diagrammasi(Timeline) yordamida amalga oshiriladi,vaqt diagrammasi parametrlari sifatida – freymlarni almashinish chastotasini, obyektning harakatlantirishni boshlash va tugatish vaqti kabilarni ko'rsatish mumkin.

Filmida bir qancha obyektlardan foydalanish mumkin. Ularni har birining holati boshqasiga bog'liq bo'lmagan holda o'zgarishi yoki o'zgarmasligi mumkin. Masalan, agar biror obyektдан fan sifatida foydalanilsa uning holatini film oxirigacha o'zgarmaslikka to'g'ri keladi. Film elementlarini bir-biriga xalaqit bermasligini ta'minlash maqsadida ularni alohida-alohida qatlam(Layer)larga joylashtiriladi. Flash dasturchilari foydalanuvchi uchun qulay bo'lishini nazarda tutgan holda har bir qatlamni shaffof oyna sifatida bo'lishini ta'minlaganlar. Bu tasvirda obyektlarni bir-biri bilan qay holatda joylashtirishni, ular ustida alohida-alohida ish olib borishni osonlashtiradi. Bir necha playerlarda, lekin bitta freymda joylashgan obektlar bir butun tasvirni va bu tasvirlar jamlanmasi film sahna

### 35-betga keldim

ko`rinishini ifodalaydi. *Sahna*(Scene) – bu Flash bilan ishlashda foydalaniladigan terminlardan biri. Har bir sahna ko`rinishi qatlamlarda tasvirlangan elementlar birlashmasidan tashkil topadi. Oddiy ko`rinishga ega bo`lgan filmlarda bitta qatlamga ega bo`lgan bitta sahna yaratish va tasvirlash etarli bo`ladi. Murakkabroq filmlar uchun bir nechta sahnadan foydalanishga to`g`ri kelishi mumkin. Bir sahnadan boshqasiga o`tish uchun endi vaqt diagrammasidan foydalanib bo`lmaydi, buning uchun boshqacha mehanizmlardan foydalaniladi. Oddiy holda filmlarni bajarolishi(namoyon bo`lishi) ularning joylashish ketma-ketligi bo`yicha boradi. Nisbatan qiyinroq bo`lgan holatlarda ActionScript tilidan foydalanishga to`g`ri keladi.

Murakkab sahnali filmlarni yaratishda yana bir asosiy tushuncha – *klip*(Clip yoki Mivie Clip) dan foydalaniladi. Klip – bu simvolning maxsus tipidir. Uni o`zining vaqt diagrammasi yordamida yaratilgan va o`zining his parametrlariga ega bo`lgan(masalan, freymlarni almashinish chastatasi) ochiq ko`rinishli film deyish mumkin.

## 1.2. Davlat boshqaruvi tizimida axborot resurslari va dasturiy vositalaridan foydalanishning samaradorligi.

### 1. Server qurilmalari va ilovalari.

Server qurilmalariga server, server kompyuterlari, server administratiri, serverning dasturiy ta`minoti, server shkaflari, Web-server, fayl server, o`yin serverlari, super-kompyuterlar kiradi.

Server – tarmoq abonentlari bo`lib, u o`z resurslarini boshqa abonentlarga foydalanishga berib, lekin o`zi boshqa abonentlar resurslaridan foydalanmaydi, ya'ni faqat tarmoqqa ishlaydi. Tarmoqda server bir nechta bo`lishi mumkin, server uchun eng quvvatli kompyuter bo`lishi shart emas. Ajratilgan server-bu server faqat tarmoq masalalari uchun xizmat qiladi. Ajratilmagan server tarmoqqa xizmat ko`rsatishdan tashqari boshqa masalalarni ham hal qilishi mumkin.

Server administratori - Web serverni beto`xtov ishlashini ta'minlovchi, xatolarni to`g`rilovchi, server va ma'lumotlarni himoya qiluvchi mutaxassis.

Serverning dasturiy ta`minoti - axborot texnologiyalarida xisoblash tizimining dasturiy vositalari xisoblanib mijoz so`rovlariga xizmat ko`rsatish vazifasini bajaradi va unga ma'lum xizmat yoki resursdan foydalanish imkoniyatini beradi.

Server kompyuter deb shaxsiy kompyuterlar to`plamidan ajratilgan va inson qatnashmaydigan ma'lum bir xizmatni bajaradigan kompyuterga aytiladi. Server va ishchi stantsiyalar bir xil apparat ta`minotiga e'ga bo`lishlari mumkin lekin inson ishchi kuchining aralashuvi bilan farq qiladi.

Server shkaflari – server qurilmalari uchun shkaflar majmusi.

Web-server – tarmoqqa ulangan kompyuter yoki undagi dastur hisoblanib, umumiy resurslarni mijozga taqdim etish yoki ularni boshqarish vazifalarini bajaradi. Web-serverlar ma'lumotlar bazalari va multimediyali ma'lumotlarni bir biriga moslashtiradi; Web-serverda Web-saxifa va Web-saytlar saqlanadi.

Fayl server bu – server xotira qurilmalaridagi fayllarga murojlatni ta'minlashni bajaradi. Ma'lumotlarni almashinish uchun quyidagi protokollardan foydalaniladi: , , SFTP i HTTP.

O`yin serverlari – markaziy o`yin xolati uchun bir qancha foydalanuvchilarga bir vaqtning o`zida qatnashish imkonini beradi. Bir qancha o`yinlar maxsus yuqori tuzolishga ega serverlarni ta'lab qolishadi.

Sekundiga bir necha yuz milliondan bir necha o`n milliardgacha arifmetik amallarni bajara oladigan tezkor ko`pprotsessorli kompyuterlar – superkompyuterlar deb ataladi.

## **2.Aloqa va telekommunikatsiyaning zamonaviy tizimlari.**

Aloqa va telekommunikatsiyaning zamonaviy tizimlari IEEE tashkilotining standartlari asosida shakllantiriladi. IEEE 802.x standarti ettita sathli OSI modelining ikki quyi sathlari – fizik va kanal sathlariga tegishli bo`lgan standartlar to`plamidan iboratdir. 802 komitet tartibiga quyidagi bo`limlar kiradi:

- 802.1 Internetwirqing – tarmoqlarni birlashtirish, ya'ni bir-nechta tarmoqlarni birgalikda ishlashini ta'minlovchi standartlarni ishlab chiqish bo`limi;
- 802.2 Logical Link Contril (LLC) – ma'lumotlarni uzatishni mantiqiy boshqarish standartlarini ishlab chiqish bo`limi;
- 802.3 Ethernet, ma'lumotlarni uzatish muhitiga murajaat qolishning CSMA/CD usuli bo`yicha ishlaydigan LQT;
- 802.4 Tiqen Bus LAN - ma'lumotlarni uzatish muhitiga murajaat qolishning Tiqen Bus usuli bo`yicha ishlaydigan LQT;
- 802.5 Tiqen Ring LAN - ma'lumotlarni uzatish muhitiga murajaat qolishning Tiqen Ring usuli bo`yicha ishlaydigan LQT;
- 802.6 Metropolitan Area Network (MAN) - megapilislar tarmoqi yoki shaqar tarmog`i;
- 802.7 Briadband Technical Advosiry Group – keng polasali axborot o`tkazish tizimlari standartlari guruhi;
- 802.8 Fiber Optic Technical Advosiry Group – optik tolali tarmoqlar bo`yicha tehnik maslahat guruhi;
- 802.9 Integrated Voice and data Networks – tovush va ma'lumotlarni uzatuvchi integratsiyalangan tarmoqlar;
- 802.10 Network Security – tarmoq xavfsizligi;
- 802.11 Wireless Networks – o`tkazgichlarsiz tarmoqlar;
- 802.12 Demand Proirity Access LAN, 100VG-AnyLAN – ahamiyatga egalik darajasi asosida talab qilish usuli bo`yicha ishlaydigan LQT.

Bluetoohtexnologiyasi RS-232 kabelli bog`lanishning o`rnini bosuvchi sifatida sa'noatda keng ishlatiladigan ma'lumot almashinish texnologiyasi hisoblanadi. U ishlab chiqarolishining va ishlatolishining oddiyligi, aloqa kanalining zararlanishdan yuqori homayalanganligi, ma'lumot uzatishning yuqori tezligi jihatlarini hisobga olib ishlab chiqarish qurilmalarida ma'lumotlar al'mashinishda qo`llaniladi.

ZigBee texnologiyasi ko`p sonli batareya asosida ishlaydigan simsiz aloqa datchiklaridan ma'lumot yoqish uchun juda mos keladi. Kichkina ZigBee moduli



yuzlab datchiklardan iborat markaziy tarmoqqa birlashtirilgan halqa asosida ma'lumot almashinish vositasiga ega tarmoq yaratish imkonini beradi. ZigBee transoverlari Texas Instruments kompaniyasi tomonidan ishlab chiqarish avtomatik tizimlarini yaratish uchun ishlatiladi.

Ma'lumotlarni uzatish vositalari sifatida simsiz aloqa (radio, GSM/GPRS, WLAN) va simli aloqa (telefon tarmog'i, ISDN, xDSL, kompyuter tarmog'i) tarmoqlari (elektrik yoki optik tolali) mavjud. Bu tizimlarda ma'lumot o'tkazish uchun quyidagi protokollari ishlatiladi RS-232, RS-485, TCP/IP, Ethernet.

В качестве среды передачи данных используются как беспроводные (Ma'lumotlarni sifatli uzatish tizimlarida simsiz aloqalarni o'rnatishda ishlatiladi) yani (радио, GSM/GPRS, WLAN), и проводные (shuningdek simli aloqada ham) (телефонные (telefonli)), ISDN, xDSL, компьютерные (kompyuterli)) сети (электрические или оптические)(elektrikli yoki optik tolali tarmoqlarda). Для передачи данных в системах телеметрии могут использоваться протоколы РС-232, РС-485, TCP/IP, Ethernet. (Namiishli tizimlar tartibida ma'lumot uzatish protokollari )

Ethernet – bugungi kunda eng keng tarqalgan lokal tarmoq standarti hisoblanadi. Hozirgi vaqtda Ethernet protokollari asosida ishlaydigan tarmoqlar soni 5 milliondan ortiq va Ethernet tarmoq adapteri o'rnatilgan kompyuterlar soni 50 milliondan ortiq. Ethernet texnologiyasining keng tarqalgan standarti 10Base-T, 100Base-T bo'lib bu tarmoqda ma'lumotlarni uzatish muhiti sifatida ikkita ekranlanmagan o'ralma juftlik (Unshielded Twisted Pair, UTP) kabeli ishlatiladi. 3-inchi kategoriyali ko'p juftli (2 ta, 4 ta juftli) UTP kabeli telefon apparatlarini ulash uchun ham ishlatiladi.

FDDI texnologiyasini (Fiber Distributed Data Interface – ma'lumotlarni optik tolali kanallar orqali uzatuvchi taqsimlangan interfeys) ma'lumotlarni uzatish muhiti sifatida optik tolali kabel qo'llanilgan birinchi lokal kompyuter tarmog'i texnologiyasidir. Kadrlarni uzatish tezligi 100 BBit/sek. Tarmoq 100 km gacha uzunlikka ega bo'lgan ikkita optik tolali halqalardan iborat bo'lishi mumkin (hammasi bo'lib 200 km uzunlikdagi optik tolali kabel). halqalarga ulanishi mumkin bo'lgan stantsiyalarning maksimal soni – 500 taga teng. Stantsiyalar orasidagi maksimal masofa 2 km dan oshmasligi kerak. Bu texnologiyada 5 kategoriyali ekranlanmagan o'ralma juftlik kabeli (UTP) ham ishlatiladi, uning maksimal uzunligi 100 m gacha bo'lishi mumkin.

Katta hajmdagi ma'lumotlarni uzatish uchun masalan video ma'lumotlarni, telemetriya tizimlarida Wi-Fi va WiMax tizimlari ishlatiladi.

WLAN (simsiz aloqa tarmog'i) – ishlash radiusi 100 m gacha. Bu tarmoq asosida binolardagi guruhli resurslarga simsiz murojaat qilish imkonini beradi masalan universitet korpusida va boshqa tashkilotlarda. Odatda bu tarmoq korxonalarda simli tarmoqlarning davommi sifatida ishlatiladi. Kichkina korxonalarda WLAN to'liq simli lokal tarmoqning o'rniga ishlatiladi. WLAN ning asosiy standarti – 802.11.

WWAN (keng simsiz aloqa tarmog'i) – simsiz aloqa vositasi, Internet va korxonalar tarmoqlaridan mobil foydalanuvchilarga foydalanish imkonini ta'minlaydi. Etakchi

standartga ega emas, lekin GPRS texnologiyasi Yevropa va AQSh da keng qo`llaniladi.

Tarmoq texnologiyalarining rivojlanish bosqichlariga nazar tashlasak bugungi kunda simsiz aloqa tarmog`i Wi-Fi bir muncha afzalliklarga ega jumladan mobillilik, o`rnatish va ishlatilishining oddiyliigi va boshqalar. Wi-Fi 802.11 simsiz aloqa tarmog`i oylasi standarti asosida 1997 yil ishlab chiqarilgan. Wi-Fi texnologiyasi simsiz lokal kompyuter tarmog`ini tashkil qilish va internetdan yuqori darajadi foydalanish uchun ishlatiladi.

### 3. Keng palasali axborot o`tkazish tizimlari.

Bugungi kunda tarmoqqa bog`lanishning asosan uchta talabi muhim ahamiyat kasb etadi bular: yuqori tezlikda foydalanish, ishonchlilik va mobillilik. Bu uchta asosiy imkoniyatni o`zida mujassam qilgan simsiz aloqa texnologiyasi bu WiMAX (World wide Interoperability for Microwave Access) tizimi hisoblanadi. Bu tizim IEEE 802.16 standarti asosida ishlaydi. WiMAX so`zining ingliz tilidagi kengaytmasi "Worldwide Interoperability for Microwave Access" bo`lib - jahon keng palasali radioaloqa protokoli degan ma'noni bildiradi. WiMAX texnologiya-sining asosiy maqsadi keng spektrli qurilmalarga(ishchi stantsiyalar, ro`zg`or texnikalarida "aqilli uy", portativ qurilmalarda va mobil telefonlari) universal tartibda simsiz bog`lanish imkonini va ularning lokal tarmoqqa mantiqiy bog`lanishini ta'minlab beradi.

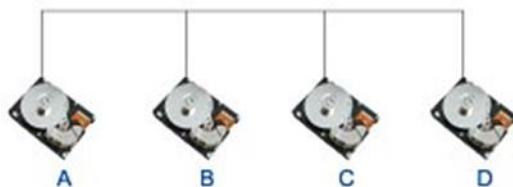
### 4. Katta hajmdagi ma'lumotlar massivini qayta ishlash va saqlash vositasi.

Raid - Redundant Array of Inexpensive Disks qimmat bo`lmagan disklarning ortiqcha massivlari. Raidning funkcionalligi quyidagicha: saqlovchi qurilmalardan massiv yaratilib u maxsus kontriller tomonidan boshqariladi va kompyuter uni yahlit katta hajmdagi disk sifatida ko`radi. Kiritish – chiqarish operatsiyalarni parallel bajarish natijasida tizimda yuqori unumdorlikka erishiladi va ma'lumotni ishonchliligini ta'minlash esa ma'lumotlarni ikkilamchi ko`paytirish yoki umumiy summasini hisoblash asosida bajariladi.

Raid massivlarni qo`llash faqatgina fizik jihatdan ishdan chiqqan qattiq diskda unumlidir. RAID massivlarning bir necha darajalari mavjud:

Raid 0,1,2,3,4,5,6,7. Bundan tashqari aralashgan darajalar ham mavjud:

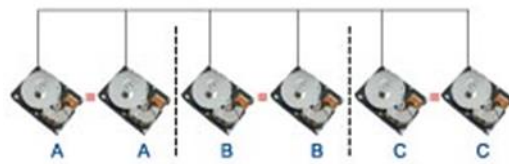
Raid10, 0+1,30,50 va h.k.z. Ularning funkcionalligi bilan tanishib chiqamiz.



**Raid 0** ishonchlilikka bardoshsiz disk massivi.

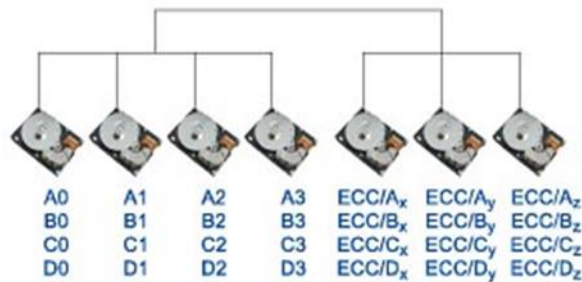
Ortiq ma`lumot saqlamaydigan disk massivi. Ma`lumotni bunday saqlash ishonchsiz chunki bir disk massivi ishdan chiqsa barcha ma`lumotlar yo`qoladi. Bu daraja ma`lumotlarni tez uzatish bosqichida ishlatish mumkin.

Raid 1. Oddiy xatoliklarga bardoshli massiv.



Akslantirilgan va duplekslangan disk massivi. Ikki saqlovchi bir xil ma`lumotni saqlab ular yagona mantiqiy disk hisoblanadi. Bir disk ishdan chiqsa ikkinchi disk uning vazifasini bajaradi. Bunday massivlarni amalga oshirishda kamida 2 ta qattiq disk kerak bo`ladi.

Raid 2. Hammaning kodini ishlatuvchi xatoliklarga bardoshli disk.



Raid 2 kam amaliyotda ham qo`llaniladi.

Raid 3. Xatolikka bardoshli massiv bo`lib ma`lumotni paralell ravishda kiritish va chiqarish va juftligini belgilash imkoniyatiga ega. Ixtiyoriy disk massivining ishdan chiqishi hech qanday ma`lumot yo`qilishiga olib kelmaydi. Ma`lumotlar bir xil hajmga bo`linib beriladi.



Raid 4 ma`lumotlar oqimi bloklarga bo`linadi. Bu turdagi massivda bir vaqtda bir necha massivdan ma`lumot o`qish imkoniyati mavjud. Bunday massiv kichik hajmdagi ma`lumotni uzatishda unumdorlikni oshirish imkoniyati mavjud.

### 1. Masofaviy terminallar.

Terminal hisoblash tizimining bir qismi bo`lib tizimga ma`lumotni kiritish va chiqarish vazifasini bajaradi. Masofaviy boshqarish rejimi mijozga server va ishchi stantsiyalarni tarmoq orqali boshqarish uchun xizmat qiladi. Bu rejim tarmoq

administratori tomonidan ajratilgan serverlarni ma'murlash va foydalanuvchilarga texnik xizmat ko'rsatishda qo'llaniladi.

Terminallar ikkita asosiy toifaga bo'linadi:

**Real fizik terminal**- tarmoq orqali keladigan ma'lumotni ko'rsatish bilan chegaralangan qurilma.

**Virtual terminal**- tarmoq dasturi bo'lib, fizik terminalni funksiyasini bajaradi.

Terminallar imkoniyati bo'yicha quydagi turlarga ega:

- ✓ Matnli terminallar.
- ✓ Grafik terminallar.
- ✓ «Intelektual» terminallar.

## 2. **Data-tsentr**lar – ma'lumotlarga ishlov berish markazlari.

Data-tsentr, ma'lumotlarni saqlash va ularga ishlov berish markazlari bo'lib, ular maxsus binolarda joylashgan bo'ladi. Data-tsentrda server va tarmoq qurilmalarining Xostinglari mavjud. Xosting – hisoblash quvvatlarini foydalanuvchilarga taqdim etish va ma'lumotlarni tarmoq serverida joylashtirish uchun mo'ljallangan xizmat turi. Xosting deb shuningdek foydalanuvchi qurilmalarini provayder maydonida joylashtirishni ta'minlaydi va yuqori tezlikdagi o'tkazish imkoniyatiga ega bo'lgan aloqa kanallariga ulashni ta'minlab beradi. Odatda Xosting sayt faylini serverda joylashtirishni ta'minlaydi. Masalan ma'lumotlar bazasi, elektron pochta Xostinglarini Misol qolib keltirish mumkin.

Data-tsentr ma'lumotlarga ishlov berish, saqlash va taqsimlash vazifasini bajaradi. Data-tsentr ma'lumotlarni biznes vazifalarini hal qilish uchun mo'ljallanadi. Hisoblash resurslari va ma'lumotlarni saqlash vositalari markazlashgani tufayli IT-infrastrukturaning narxi kamayadi.

Texnik vositalardan samaraliroq foydalaniladi. Data-tsentr odatda aloqa tugunlari yoki aloqa operatorlariga yaqinroq bo'lgan joylarda joylashadi. Kanallarning sifati va o'tkazuvchanlik qobiliyati amalga oshirilayotgan xizmatlar turiga ta'sir ko'rsatadi. Istalgan data-tsentrning sifatli ishlashining asosiy ko'rsatkichi bo'lib serverga murojaat etish vaqti ya'ni (aptaym) hisoblanadi.

Data-tsentrning tarixi katta hajmli kompyuter-xonalaridan boshlangan. U paytlarda kompyuter tizimlarini boshqarish uchun maxsus sharoitlar talab etilar edi. Kompyuterlar ko'p joy egallar edi. Axborot texnologiyalari rivojlanishi bilan, kompaniyalar ko'proq IT resurslarini nozorat qilishga e'tibor bermoqdalar va hozirgi paytda standart server stoykalari va kabel kanallaridan foydalaniladi. Data-tsentrning kuchli rivojlanishi 1995—2000 yillardan boshlandi. Kompaniyalar uchun Internet tarmog'iga muntazam ravishda va tezkor ulanish juda zarur bo'lgani sababli data-tsentr rivojlanishi kuchaytirildi.

Data-tsentrning tuzolishi quyidagicha:

- axborot infrastrukturasini. Axborot infrastrukturasini server qurilmalaridan iborat bo'lib, ma'lumotlarni qayta ishlash va saqlashni ta'minlaydi;

- telekommunikatsiya infrastrukturasi. Telekommunikatsiya infrastrukturasi data-tsentr tarkibiy qismlarini o`zaro bog`lashni va normal faoliyatini ta'minlab beradi.

-muhandislik infrastrukturasi. Muhandislik infrastrukturasi standart norma-larga mos keladigan xona harorati va namlik satxini, to`xtovsiz tok manbasidan foydalanish, yong`indan qo`riqlash tizimini nazorat qolishni ta'minlaydi. Yuqori malakali mutaxassislar doimiy ravishda barcha serverlarni monitoring qolishadi. Data-tsentrlerde qurilmalar odatda maxsus stoyka va shkaflarda joylashtiriladi.

### **3.Ma'lumotlarga parallel va taqsimlangan ravishda ishlov berishning apparat-dasturiy komplekslari**

Ma'lumotlarga taqsimlangan ravishda ishlov berish – o`ta murakkab masalalarni hisoblash tizimiga birlashtirilgan bir nechta kompyuterlar yordamida, yechish jarayoniga aytiladi. Taqsimlangan tizimlarda ketma-ket hisoblashlar bir nechta masalalarni bir vaqtni o`zida yechish usuli bilan amalga oshiriladi.

Parallel hisoblash tizimlari – bu shunday kompyuter va dasturiy tizimlarki, ular bir nechta hisoblash tugunlari yordamida bajariladi.

Masalan, ikki protsessorli kompyuterda ma'lumotlar massivini tez saralash uchun, massiv ikkiga bo`linadi, va har bittasi alohida protsessorida saralanadi. Massivning har birini saralash har xil vaqt talab etadi, shu sababli soxrinizatsiya zarur bo`ladi. Masalan ma'lumotlarni parallelligini ko`rib chiqamiz. Parallellikning asosiy g`oyasi shundan iboratki, bunda bitta operatsiya massivning barcha elementlari ustida bir vaqtning o`zida bajariladi. Massivning har xil frag-mentlari vektor protsessorlarida parallel mashinaning har xil protsessorlarida amalga oshiriladi. Ma'lumotlarni protsessorlarga taqsimlashni programma bajaradi. Dasturlash nuqtai nazaridan, bitta hisoblash masala bir nechta nisbatan alohida masalalarga bo`linadi va har bir protsessor o`zining masalasini bajaradi.

Taqsimlangan tizimlarda har bir tizim elementi uchun bajariladigan vazifalarning, resurslarning taqsimlanishi alohida ahamiyatga ega. Taqsimlangan tizimda yagona boshqaruvchi markaziy tugun mavjud bo`lmasligi sababli, bitta tugunning ishdan chiqishi, tizimdagi boshqa tugunlarga ta'sir ko`rsatmaydi. Bunga Internet tizimini Misol qolib keltirish mumkin. Masalan taqsimlangan ma'lumotlar bazasi.

Ma'lumotlarni taqsimlangan ravishda ishlov berishning quyidagi prinsiplari mavjud:

1. Lokal mustaqillik. Tugunlar alohida avtonom ravishda bo`lishi zarur. Tugundagi barcha operatsiyalar shu tugun tomonidan nazorat qilinadi.

2. Barcha tugunlar o`zaro teng.

3. Uzluksiz ishlash qobiliyati. Tizim ishonchliligi yuqori darajada bo`lishi.

4. Foydalanuvchilarning ma'lumotlari qayerda joylashganligi muhim ahamiyatga ega emasligi.

5. Fragmentatsiyaga bog`lanmasligi.

6. Replikatsiyaga bog`lanmasligi.

7. Taqsimlangan so`rovlarga ishlov berish. So`rovni bajarish uchun bir nechta tugunlarga murajaat etish talab etolishi mumkin.

8. Taqsimlangan tranzaksiyalarni boshqarish uchun ikkita usul mavjud: tiklash, boshlash, boshqarish, parallel ravishda qayta ishlash bilan boshqarish.

9. Operatsion tizimga bog`lanmasligi. BBBT lari turli operatsion tizimlar boshqaruvida ishlaydi.

10. Tarmoqqa bog`lanmaslik.

11. BBBT turiga bog`lanmaslik.

#### **4.Xosting tizimi va domenlarni boshqarish.**

Xosting hisoblash quvvatlaridan foydalangan holda ma`lumotni internet tarmog`ida ishlaydigan serverda joylashtirish xizmati hisoblanadi. Xosting deb yana mijozning qurilmalarini provayderning hududida joylashtirilgan va ularga yuqori o`tkazuvchanlikni ta`minlovchi aloqa kanallari bilan (odatda Internet) ta`minlashga aytiladi. Odatda Xosting saytlarga xizmat ko`rsatish paketiga kiritilgan bo`lib, sayt fayllarini serverda joylashtirish va bu fayllarga so`rovlar bilan qayta ishlash imkoniyatini beradi (vab server). Xostingni tanlashning asosiy tamoyillaridan biri bu saytning ishlashini ta`minlab beruvchi xizmatlariga bog`liq bo`lgan operatsion tizimdan iborat. Xostingning asosiy tavsifi uning u bu xizmat va imkoniyatlarni berishidir.

Ularga:

- ✓ CGI: Perl, PHP, Pythin, ASP, Ruby, JSP ta`minoti.
- ✓ .htaccess/.htpasswd (Apache uchun) ta`minoti.
- ✓ Ma`lumotlar bazasi ta`minoti.
- ✓ Xosting xizmatlarini quyidagi parametrlar bilan tanlash mumkin:
- ✓ Foydalanuvchi ma`lumotlariga disk joylarini ajratish.
- ✓ Oylik trafik miqdori.
- ✓ Bir qaydnoma nomiga bir necha saytlarning sonini joylashtirish.
- ✓ FTP foydalanuvchilarning soni.
- ✓ E-mail pochta soni va hajmi.
- ✓ Ma`lumotlar bazasi soni va unga ajratilgan hajm.
- ✓ Bir foydalanuvchiga berilgan jarayonlar soni.
- ✓ Operativ xotira hajmi, har bir foydalanuvchiga ajratilgan maksimal ish bajarish vaqti.
- ✓ Sifatga bog`liq cheklovlar:
- ✓ Serverning tezligiga ta`sir ko`rsatuvchi: markaziy protsessor bo`sh resursi, operativ xotirasi.

Ma`lumotlarni yuklashda kanallarning o`tkazish tezligi.

Birqancha pullik Xosting kompaniyalari bepul test Xosting xizmatini vaqtinchalik berishadi. Bundan maqsad Xosting kompaniyalari xizmatlarini tanlashda va uzoq vaqt davomida ishlatish mumkinligini ko`rish mumkin. Katta kompaniyalarda Xosting xizmatini yoqtirish, saytga domen nomini ro`yxatdan o`tkazish bilan amalga oshiriladi. Xosting kompaniyalari bilan shartnomani cho`zish domenni qaytadan ro`yxatdan o`tkazish bilan amalga oshiriladi. To`lov jihatdan Xosting pullik va bepul bo`ladi.

Xosting turlari

- ✓ Virtual Xosting provayderning qattiq diskini ijaraga olish.

- ✓ VDS – Virtual serverni arendaga olish.
- ✓ CillicaTION – fizik serverni arendaga olish.
- ✓ Domen
- ✓ Saytingizni internetdagi manzili.
- ✓ Adresni domen nomlarini registratori beradi.
- ✓ To`lov bir yilga amalga oshiriladi.
- ✓ Domen hususiyashtirilmaydi.

Domen - Internet tarmog`ining bir qismi hisoblanib, korxonalar tassarufiga berilgan maxsus nomlar bilan ajratilgan nom.

Misol. Tirg.uz

Domen nomi– Internet tarmog`ida ma'muriy avtonomiyaga ta'luqli hududni belgilovchi nom hisoblanadi.

DNS - Dimain Name System, domen nomlari tizimi. Har bir internetdagi sayt o`z domen nomiga ega. DNS tizimi server IP adresoni saytga big`lab beradi.

Sayt <http://cctld.uz/> UZ domenining ma'muriyati hisoblanadi.

“UZ” domeni zonasida domen nomlarini rasmiy ro`yxatga oluvchi tashkilotlar.

III «Tomas», III «Amaliy Aloqalar Biznesi», III «Sarkor Telecom», III «BILLUR COM», III «Arsenal-D», III «NEW LINE SILUTIONS».

Regional ro`yxatga olishda kompyuterlashtirish markazlari quyidagilar hisoblanadi: Farg`ona C.C. , Qoraqalpog`iston C.C. , Namangan C.C. , Navoi C.C., Samarqand C.C. Xorazm C.C. , Surxondaryo C.C. , Sirdaryo C.C., Qashqadaryo C.C.

qo`shimcha ma`lumot va hujjatlar quyidagi manzil bo`yicha ko`rish mumkin: <http://cctld.uz/info/>

## **5.Virtualizatsiya tizimlari.**

Virtualizatsiya uch toifaga bo`linadi:

- ✓ Taqdimot virtualizatsiyasi.
- ✓ Dasturiy virtualizatsiya.
- ✓ Server virtualizatsiya.

Taqdimot virtualizatsiyaga eng yorqin Misol Windows Server terminal xizmatidir. Terminal server mijozlarga o`zining hisoblash resurslarini taqdim qiladi, mijozdagi taqdimot serverda bajariladi va mijozga faqat uning ko`rinishi “taqdimot ko`rsatiladi. Bunday murajaat modeli birinchidan - mijoz tomonidagi apparat talablarini kamaytiradi, ikkinchidan- tarmoqning o`tkazish tezligini oshiradi, uchinchidan- axborot xavsizligini ta'minlaydi.

Dasturiy virtualizatsiya- alohida dasturlarni izilyatsiyalangan muhitda ishlashini ta'minlaydi( Sandbox). Bunday usul ko`pgina muammolarning yechimiga olib keladi. Birinchidan- alohida izilyatsiyalangan muhitda ishlovchi dastur operatsion tizimga va boshqa dasturlarni ishdan chiqishiga halaqit qilmaydi. Ikkinchidan virtuallashtirilgan dasturlarni markazlashtirilgan ravishda yagona manba tomonidan yangilash mumkin. Uchinchidan yagona fizik persional kompyuterda to`g`ri kelmaydigan bir necha dasturiy ta`minotlarni ishga tushirish mumkin.

Server virtualizatsiyasi- maxsus dasturiy ta`minot bo`lib, kompyuterning apparat imkoniyatlarini: protsessor, xotira, qattiq disk, va turli qurilmalarini imotatsiya qilib beradi. Bu texnologiyaning eng asosiy imkoniyati bir necha virtual kompyuterlarni yagona fizik kompyuterda ishlashi va shu bilan birga ulardagi operatsion tizimlar bir biriga halaqit qilmaydi.

Yuqoridagilarni inobatga olgan holda agar sizda yagona dasturiy ta`minotdan foydalanuvchi soni ko`p va tizim hudud bo`yicha taqsimlangan bo`lsa unda taqdimot virtualizatsiyasidan foydalanish maqsadga muvofiq.

Agar sizda bir necha dasturiy ta`minotlar yangi operatsion tizimda noto`g`ri ishlayotgan bo`lsa va bir biri bilan to`g`ri ishlamaydigan dasturlar bo`lsa dasturiy virtualizatsiyadan foydalanish maqsadga muvofiq.

Agar server xonasidagi serverlarni soni ko`payib ketgan va elektrenergiyani tejash kerak bo`lsa unda server virtualizatsiyasidan foydalanishingiz mumkin. Server virtualizatsiyasi eskirgan serverlarda ishlamaydi. Yangi texnologiyadagi server qurilmasini sotib olish kerak bo`ladi.

## **6. Axborot xavsizligini ta'minlashning zamonaviy usullari.**

Axborot xavsizligini ta'minlash deganda axborot xavsizligini ma'muriy, texnik va texnologik himoyalash tushuniladi. Ma`lumotlarga avtorizatsiyalanmagan murajaat, buzish, modifikatsiya qilish va ma`lumotni tarqatishni oldini olish kiradi. Axborot xavsizligi quyidagi maqsadlarni o`z ichiga qamrab olgan:

- ✓ Ma`lumotni konfidentsialligi. Ma`lumot tarqalishining oldini olish.
- ✓ Axborotni butunligini ta'minlash. Ma`lumotni uzatishda hech qanday o`zgartirish kiritmasligi.
- ✓ Axborotni belgilangan shaxslar uchun ochiqligi.
- ✓ Axborotni nozorat qilish.

Axborotni himoya qilishda axborot tizimiga qo`yilgan talablarga quyidagilar kiradi:

- ✓ Apparat himoyasi.
  - ✓ Dasturiy himoya.
  - ✓ Aloqa bilan ta'minlash(kommunikatsiya).
- Axborotni himoyalash mexanizmlari quyidagilar:

- ✓ Fizik darajadagi himoya.
  - ✓ Persinalni himoyalash.
  - ✓ Tashkiliy darajadagi himoya.
- Tashkiliy darajadagi himoyada:
- ✓ Himoyalash tizimini tashkil qolish.
  - ✓ Hodimlar bilan ishlashni tashkillashtirish.
  - ✓ Hujjatlar bilan ishlashni tashkillashtirish.
  - ✓ Texnik vositalar bilan ishlashni tashkillashtirish.
  - ✓ Tashqi va ichki hujumlarga qarshi ishni tashkillashtirish.
  - ✓ Hodimlar bilan tizimli ravishda ishlash va nozorat qilish kiradi.

Axborot xavsizligining himoyasini texnik vositalariga:

- ✓ Yong`in xavsizligi signalizatsiyasi.
- ✓ Raqamli video kuzatish tizimi.



✓ Nozorat va boshqarish tizimi.

Axborotni aloqa kanallari tomonidan chiqmasligini ta'minlashda quyidagi usullardan foydalaniladi:

- ✓ Ekranlashgan kabellardan foydalanish va ekranlashgan konstruksiyalarga joylashtirish.
- ✓ Aloqa liniyalariga yuqori chastotali filtrlarni joylashtirish.
- ✓ Ekranlashgan xonalarni qurish.
- ✓ Ekranlashgan qurilmalarni ishlatish.
- ✓ Shovqinga qarshi tizimlarni o'rnatish.
- ✓ Nozorat qilinadigan hududlarni yaratish.
- ✓ Axborotni himoyalashning apparat vositasi.

Himoyalash rekvezitlarni saqlash uchun maxsus registrlar:

- ✓ Parollar, identifikatsiyalovchi kodlar.
- ✓ Insonni individual xarakteristikasini o'lchovchi qurilmalardan foydalanish.
- ✓ Ma'lumotni shifrlovchi qurilmalar (kriptografik metodlar).
- ✓ Doimiy ma'nba bilan ta'minlovchi tizim. Doimiy ma'nba.
- ✓ Yuklamalarni rezervlash.
- ✓ Yuklanishlarning generatori.

Axborot xavsizligini ta'minlashning dasturiy vositalariga:

- ✓ Kutilmagan murajaatlarni oldini olish vositasi.
- ✓ Axborot oqimi ni modellashtiruvchi va tahlil qiluvchi tizim.
- ✓ Tarmoqni monitoring qiluvchi tizim.
- ✓ Protokollarning tahlili.
- ✓ Antivirus dasturlar.
- ✓ Tarmoqlararo ekranlar.

Kriptografik vositalar:

- ✓ Shifrlash, elektron raqamli imzo.
- ✓ Rezerv ko'chirish tizimi.

Autentifikatsiya tizimi:

- ✓ Parol.
- ✓ Murojbat kaliti.
- ✓ Sertifikat.
- ✓ Biometriya.

Tizim xavsizligini ta'minlashning uskunaviy vositasiga:

- ✓ Monitoring qiluvchi dasturiy ta'minot kiradi.
- ✓ Milliy axborot tizimini shakillantirishda axborot tizimlari majmuasini yaratish va integratsiyalash usullari.

Himoya tizimining kompleksligiga unda huquqiy, tashkiliy, muhandis – texnik va dasturiy – matematik elementlarning mavjudligi bilan erishiladi. Elementlar nisbati va ularning mazmuni tashkilotlarning axborotni himoyalash tizimining o'ziga xosligini va uning takrorlanmasligini hamda buzish qiyinligini ta'minlaydi. Aniq tizimni ko'p turli elementlardan iborat, deb tasavvur qilish mumkin. Tizim elementlarining mazmuni nafaqat uning o'ziga xosligini, balki axborotning qimmatligini va tizimning qiymatini hisobga olgan holda belgilangan himoya darajasini aniqlaydi.

## **2- BOB. Berilganlar bazasi. Shakl, so`rov va hisobot tayyorlash.**

### **Ma'lumotlardan jamoa bo`lib foydalanish**

Kompyuter yordamida qayta ishlanuvchi katta hajmdagi ma'lumotlarni zich saqlashni (takrorlamasdan) ta'minlash va ular ustida qidirish, tartiblash, qo`shish, o`zgartirish, qayta ishlash amallarini bajarish kabi masalalar mavjud. Shunday ekan, ma'lumotlarni eng yaxshi ko`rinishda ifodalashni tashkil etish hamda eng qulay boshqarolishini ta'minlash kerak.

Ma'lumotlar - keyingi qayta ishlash, saqlash va uzatish uchun yaroqli muayyan shaklda taqdim etilgan axborotlardir. Ma'lumotlar tuzilmasi – bir necha ma'lumotlar elementlarini bittaga birlashtirish usuli. Predmet sohasi – boshqaruvni tashkil etish va avtomatlashtirish maqsadida o`rganolishi lozim bo`lgan real dunyoning bo`lagidir. Predmet sohasi, agar unda mavjud obyektlar, ularning xususiyati va munosabatlari ma'lum bo`lsa, aniqlangan hisoblanadi. Ma'lumotlar modeli – ma'lumotlar va ular orasidagi aloqa ko`rinishidagi predmet sohasi haqidagi tushuncha. Ya'ni ma'lumotlar modeli – bu o`zari big`langan ma'lumotlar tuzilmalarining to`plami va bu tuzilmalar ustidagi amallarni anglatadi.

«Ma'lumotlar modeli» tushunchasi uchta tashkiliy qismdan iborat:

ma'lumotlarni tashkillashtirish (ma'lumotlar modeli obyektlarining turlari va miqdori, ma'lumotlar tuzilmasoni chegaralash);

ma'lumotlar ustida bajarish mumkin bo`lgan ko`plab amallar:

tanlash amallari (qidiruv), modifikatsiya amallari (qo`shish, o`chirish, ma'lumotlarni o`zgartirish);

ma'lumotlarning ishonchliligini va mantiqiy butunligini ta'minlash vositalari (ma'lumotlar qiymatlari va aloqani chegaralanishi). Ular orqali saqlanadigan axborotlarning qarama-qarshi bo`lmasligiga erishiladi.

Ma'lumotlar modelini tanlash axborot hajmi, yechiladigan masalaning murakkabligi va mavjud texnik hamda dasturiy ta`minotiga bog`liq.

Berilganlar bazasi(BB) – ma'lum predmet sohadagi ma'lumotlar to`plamidir. Ular ma'lum qoidalar (tasvirlash, saqlash va manipulyatsiya qilishning umumiy tamoyillari) bo`yicha tashkillashtirilgan hamda qayta ishlash dasturlariga bog`liq emas.

Ma'lumotlar bazasini boshqarish tizimi (BBBT) – ma'lumotlar bazasini yaratish va xizmat ko`rsatish hamda ulardagi ma'lumotlar ustida amallar bajarishni (ulardan foydalanish imkoniyati va qayta ishlash) ta'minlovchi dasturiy vositalar (dasturiy tizim yoki dastur paketi) to`plamidir. BBBT tarmoqli, oerarxik yoki relyatsion turdagi ma'lumotlar modellaridan birini qo`llaydi. Relyatsion model ma'lumotlarni jadval ko`rinishida taqdim etishga yo`naltirilgan, ya'ni ikki o`lchamli jadval ko`rinishida ma'lumotlarni tashkillashtiradi. To`plamlar nazariyasida jadval so`ziga munosabat (relation) atamasi mos keladi va model nomi shundan kelib chiqqan. Relyatsion ma'lumotlar bazasi – bu ma'lum predmet sohasida munosabatlar (to`g`ri to`rtburchakli jadvallar) to`plami sifatida mantiqiy tashkillashtirilgan ma'lumotlar bazasidir.

Predmet sohasining obykti - bu jadval, satr - obyektning bir nusxasi haqidagi yozuv. Ma'lum ma'lumotlar elementlarini jadvalning bir satriga joylashtirish ular orasidagi aloqa yoki munosabatni o`rnatishni bildiradi. Jadval ustunidagi (maydonidagi) qiymatlar obyektning tavsifini yoki xususiyatini (munosabat atributini) aniqlaydi. Jadval son jihatdan chegaralangan, ketma-ketligi aniqlangan ustunlardan iborat.

Satrlar soni ixtiyoriy bo`lib, tartiblanishida farq yo`q.

Jadvallar quyidagi xususiyatlarga ega:

ustunlarga (maydonlarga) noyob nom beriladi;

ustunlarning har bir elementi bir xil tabiatga ega, ya'ni ustunlar bir jinslidir;

jadvalda bir xil satr (yozuv) mavjud emas, ya'ni har qanday ikki satr hech bo`lmaganda bitta elementi (yozuv maydoni) bilan farqlanadi;

satr va ustunlarga ixtiyoriy ketma-ketlikda ishlov beraolishi mumkin.

Relyatsion ma'lumotlar bazasiga odatda bir qancha jadvallar kiradi. Jadvallar orasidagi aloqa kalitlar orqali amalga oshiriladi.

Kalit – qiymati tom ma'noda jadvaldagi yozuvni ifodalovchi atribut (maydon) yoki atributlar to`plami.

Relyatsion ma'lumotlar bazasida ma'lumotlarni saqlashning ustunlik jihatlari quyidagilarda ifodalanadi:

ma'lumotlarning har bir elementi faqat bitta jadvalda saqlanadi

(joyni tejash);

kiritiladigan o`zgartirishlar soddalashadi, xato qilish xavfi kamayadi (masalan, familiyani yozishda);

jadvallar orasidagi aloqalar o`zaro bog`langan axborotlarga ishlov berishni tezlashtiradi;

xato harflarga ega yozuvlar avtomat ravishda o`chiriladi.

Shaxsiy kompyuterning texnik imkoniyatlariga hozirgi vaqtda Relyatsion BBT juda ham mos keladi.

Ma'lumki, axborot tizimi (AT) barcha foydalanuvchilarni kerakli axborotlar bilan ta'minlovchi, berilgan predmet sohasi bo`yicha axborotlarni yig`ish, uzatish va qayta ishlash bo`yicha kommunikatsiya tizimini ifodalaydi. Zarur axborotlarni olish maqsadida ma'lumotlarni markazlashgan yig`ish va ulardan jamoaviy ko`p aspektli foydalanish uchun mo`ljallangan matematik, dasturiy, tili, tashkiliy va texnik vositalar tizimi axborot tizimi sifatida ifodalanadi.

quyidagilar ATning asosiy tashkil etuvchilari hisoblanadi:

ma'lumotlar bazasi. Unda zarur axborotlar va saqlanuvchi ma'lumotlar tuzilmasi ifodalanadi;

ma'lumotlar bazasini boshqarish tizimi. Unda ma'lumotlarni boshqarishning namunali amallari bajariladi;

amaliy dastur (foydalanuvchi ilovasi). Bu dastur foydalanuvchining axborot tizimi bilan muloqati uchun kerakli algoritmi ro`yobga oshiradi. Algoritmida BB bilan ishlash hamda ma'lumotlarga ishlov berishdagi barcha amallarni bajarish yoritilgan bo`ladi.

Predmet sohasini avtomatlashtirish turli masshtabda (ma'lumotlar, masalalar va boshqaruv vazifalari to'plamiga ishlov berishning alohida jarayonlaridan tortib to'plamga qadar) funksional to'plamga avtomatlashtirilgan axborot tizimini (AAT) yaratishga qadar amalga oshirish mumkin. AATni loyihalashtirish jarayonidan oldin predmet sohasining tahlili bajariladi. Ushbu bosqichda quyidagilar amalga oshiriladi:

barcha foydalanuvchilarning ehtiyojidan kelib chiqib, ma'lumotlar bazasida saqlanadigan ma'lumotlar va ularning manbaini aniqlash;

ma'lumotlarga ishlov berishda mavjud va kelajakda hal etilishi kerak masalalarni aniqlash;

tahlil natijalarini hujjatlashtirish.

Demak, axborot-dasturiy to'plamining AATni yaratish jarayonini ikki qismga bo'lish mumkin:

hal etilishi lozim bo'lgan masalalar tarkibini aniqlash; – integrallashgan ma'lumotlar bazasini yaratish.

Bu ikki qismning o'zari bog'langanligi ma'lumotlar bazasining tuza olishi berilgan predmet sohasidagi ma'lumotlarga ishlov berish masalasiga bog'liqligida namoyon bo'ladi.

Shunday qilib, berilganlar bazasi (BB) - bu predmet sohasining ayrim modeli bo'lib, unda predmet sohasining faqatgina axborot tizimi ishlashi uchun kerak bo'lgan faktlar aks etadi.

Foydalanuvchi ilovasini ishlab chiqish etadi BBni yaratish etapidan oldin yoki parallel olib boriladi. BB yaratilgan so'ng, berilgan masala ma'lumotlariga ishlov berish algoritmi BBSidan saralash orqali hosil qilingan chiqariluvchi hujjatlarni tayyorlash jarayonini anglatadi.

Berilgan masalani yechishda foydalanuvchi uchun mo'ljallangan so'rovlar, ekran shakllari, hisobotlar, makroslar, standart dasturlarni hosil qilish BBT vositalari yordamida amalga oshiriladi.

har qanday masalani yechish algoritmini yaratishda sodda hosil qilinuvchi so'rovlar yoki hisobotlar orqali ma'lumotlarga ishlov berishning umumlashtirilgan amallariga tayanish kerak.

Foydalanuvchining muloqat ilovasi (amaliy dastur) predmet sohasi ma'lumotlariga ishlov berishning barcha texnologiyalarini (jumladan BBni yuklash, yuritish va barcha masalalar to'plamini yechish) birlashtiradi.

Dastur «do'stona» interfeysga ega bo'lishi va berilgan masalalarni yechish uchun quyidagi buyruqlar to'plamini o'z ichiga olishi kerak:

yangi ma'lumotlar to'plamini (jadvallar, ro'yxatlar) yaratish;

qo'shish, o'chirish, o'zgartirish, ma'lumotlarni ko'rish; ma'lumotlarni saralash;

so'rov bo'yicha ma'lumotlarni tanlash;

chiquvchi hujjatlarni tuzish (ekranda ko'rish va chop etish imkoniyati bilan).

Muloqat oynalari, tugmalar, «tezkor klavishlar» va shu kabilardan foydalanib tuzilgan oeraxik menyu asosida foydalanuvchi interfeysi yarataolishi kerak.

ATdan foydalanishda BBT taqdim etuvchi quyidagi uskuna vositalaridan qo'llash kerak:

ma'lumotlarni kiritish uchun muloqot oynalarni va chiquvchi hujjatlarni hosil qiluvchi ekran shakllari va hisobotlar generatorlari (konstruktorlar, ustalar);

ma'lumotlarni filtrlash va qidirish uchun so'rov tillari;

axborotga ishlov berish algaritmlarini qo'llash uchun dasturlash tillari.

Ma'lumotlar bazasi yaratish va Axborot texnologiyalarini ishlab chiqishning barcha bosqichlari hujjatlashtirilgan bo'lishi kerak. Loyihalashtirish va qo'llash jarayonida tasvirlar, chizmalar, testlar, qo'l izlari kabi ishchi (oraliq) hujjatlar yaratiladi. Ba'zi ishchi hujjatlar keyinchalik hisobot (yakuniy) hujjatlar tarkibiga kiradi.

### **1.1.Berilganlar modeli**

Zamonaviy BBBT (Berilganlar bazasini boshqarish tizim) lari datalogik bosqichda xilma- xil BB bilan ishlashini ta'minlaydi. Hozirgi kunda eng ko'p o'rganilgan va keng ishlatiladigan soha bu ma'lumotlar modeli hisoblanadi. Ma'lumot modeli, bu BB ni ma'lumot elementlari to'plami orasidagi bog'lanish strukturalarini tasvirlovchi umumiy sxemadir. Ma'lumot modeli tushunchasini aniq ta'rifini ingliz olimi Kodd o'zining nazariyalarida berib o'tgan. U ma'lumot modelini 3 ta kerakli komponentasini keltirgan:

1. Mavjud bo'lgan ma'lumot strukturalarini aniqlash vositalari majmuasi;
2. Ma'lumotlarni bildirish va takomillashtirish uchun BB holatiga qo'llaniladigan amallar to'plami.
3. Oshkor holda BB holatini aniqlovchi va bir butunligini to'minlovchi vositalar to'plami.

Hozirgi kunda klassik hisoblashlarda 3 ta ma'lumot modeli ko'p ishlatiladi:

- ✓ Ierarxik ma'lumot modeli
- ✓ Tarmoqli ma'lumot modeli
- ✓ Relyatsion ma'lumot modeli

Har bir BB u yoki bu model asosida yoritiladi. Har bir BBBT esa u yoki bu ma'lumot modelini ta'minlaydi deyiladi. Masalan, ierarxik ma'lumot modeliga asoslangan sistema - ines sistemasi hisoblanadi. Tarmoqli modellarda esa – BANK IS, SETIR, Relyatsion modelga asoslangan sistemalar– Access, KARAT va boshqalarni o'z ichiga iladi.

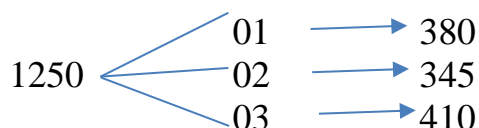
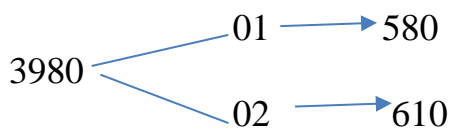
Ierarxik modellar.

Ierarxik modellarda yozuvlar daraxtsimon strukturali ko'rinishda bo'ladi. BBBT laridan ba'zi birlari faqat ierarxik tuzaolishga ega bo'lganlari bilan ishlatiladi. Ierarxik tuzaolishli ma'lumotlar sodda yaratiladi. Bu ko'pincha tadbirlarda qulay, lekin ma'lumotlarning ko'plari daraxtsimon strukturali bog'lanish tabiatiga ega emas.

Misol, faraz qilaylik 2 ta firma ishlab chiqargan mahsulotning barcha turlarining narxlari berilgan. Shu ma'lumotlarni narxlar ma'lumotnomasi qurilsin va kompyuter xotirasiga joylashtirilsin.

Faraz qilamiz, A va B 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. Firmalarning 3980 mahsulotini 2 ta sxema asosida tayyorlaymiz, ularni 01, 02 deb belgilaymiz shundan so'ng ular quyidagi narxni belgilaydilar 578, 612. 1250 mahsulotni 3ta sxema asosida tayyorlaymiz, ularni 01, 02,03 deb belgilaymiz va ular quyidagi narxni belgilaydi

380, 345 va 410. B firma 3 xil mahsulot ishlab chiqaradi ,ularning kodi mos ravishda 1250, 1640 va 1930 kodga ega bo`lsin. Ular ham o`zlarini ishlab chiqish sxemasi va narxiga ega bo`lsin.



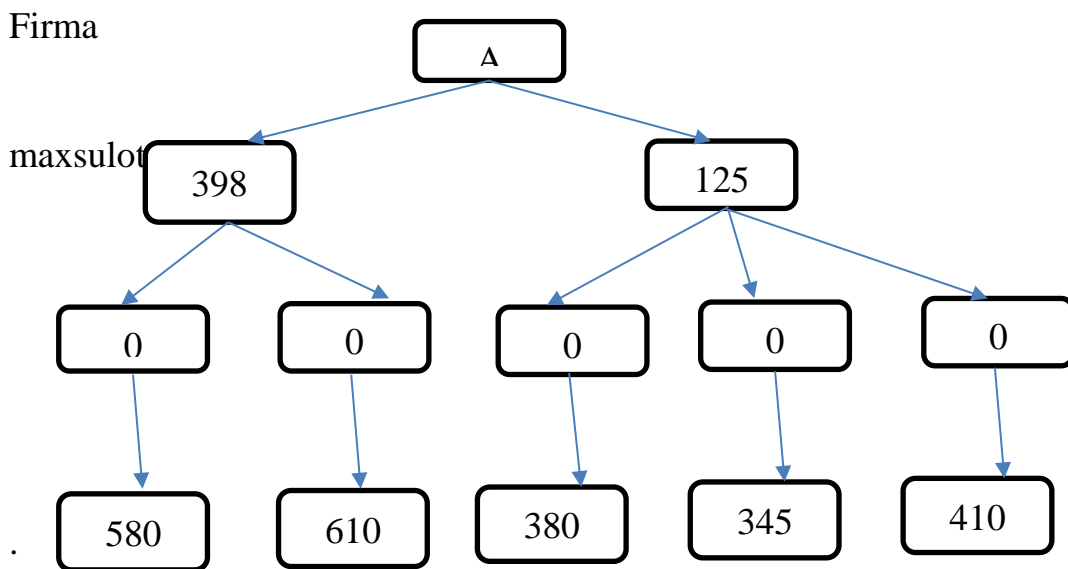
B firma 3 xil mahsulotni ishlab chiqaradi: 1250, 1640, 1930.

1250 —

1640 —

1930 —

Barcha mahsulotlar nomlari, ularni ishlab chiqish sxemasi va narxlarni o`z ichiga oluvchi ma`lumotnoma tuzish va mashina xotirasiga joylash talab etilsin. Keltirilgan ma`lumotlar strukturani ierarxik (daraxtsimon) ko`rinishda tasvirlashni amalga oshiramiz. Bunda quyidagiga ega bo`lamiz.



Bunday strukturani jadval ko`rinishida ham tasvirlash mumkin.

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

## Ma'lumotning tarmoqli modeli.

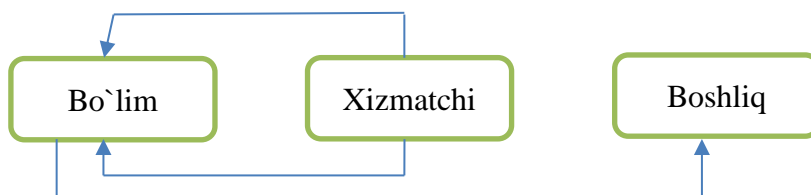
Agar berilgan element bir nechta o'zidan yuqori elementga suyansa tarmoqli ma'lumot elementiga ega bo'lamiz.

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.

Tarmoq strukturalarida element ixtiyoriy boshqa element bilan bog'lanishi mumkin. Ya'ni, tarmoqni bir necha kichkina obyektlardan tashkil topgan yirik obyekt deb qarash mumkin.

Ushbu chizmada bog'lanishlarni tarmoqli modelda tasvirlanishi keltirilgan.



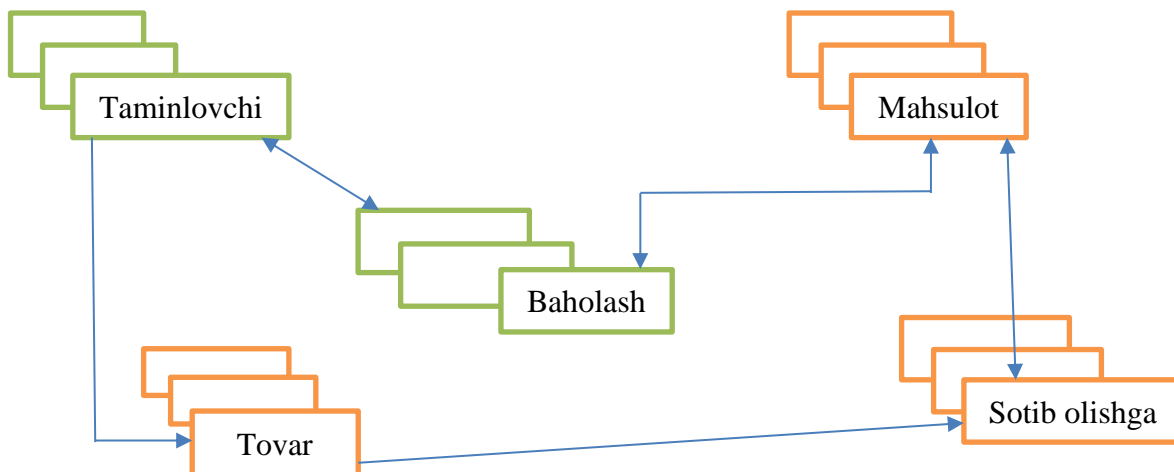
Shunday qilib, tarmoqli model ma'lumotli elementlari orasidagi xilma-xil bog'lanishlarni ixtiyoriy ko'rinishdagi grafik yordamida akslantiradi. Tarmoqli model yozuvlar to'plami va mos bog'lanishlar to'plamidan tashkil topadi. Bog'lanishlarni yaratish uchun alohida cheklanishlar quyilmaydi. Misol: oddiy tarmoqli BB sxemasi sifatida quyidagini keltirish mumkin.

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

1. BB 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. Bog'lanishlarni o'zgartirish

Misol:

Beshta toifali yozuvdan iborat bo'lgan tarmoqli modelga Misol.



Yuqoridagi sanab o`tilgan berilganlar bazalaridan tashqari hozirgi kunda quyidagi BB lar ham amaliyotga kirib kelmoqda.

1. Ko`p o`lchamli BBlari
2. Obyektga yo`naltirilgan BBlari

Shuningdek boshqa BB lariga asoslangan har hil tizimlar ham ishlab chiqilmoqda. Bular satrida quyidagilarni sanash mumkin.

1. Obyekt-Relyatsion modellar
2. Semantik modellar
3. Yo`naltirilgan modellar
4. Kintseptual va boshqa modellar

Ulardan ba'zilari BB bilimlari bazasi va dasturlash tillarini integratsiyalashga (birlashtirishga) xizmat qiladi.

## RELYATSION BERILGANLAR MODELI

Infologik modellashtirishning maqsadi – tuziladigan ma'lumotlar bazasida shakllanishi mumkin bo'lgan ma'lumotlarni tasvirlash va yig'ish usullarini odamlar uchun ayniqsa tabiiy ta'minlashdir. Shuning uchun ma'lumotlarning infologik modelini tabiiy tilga mos qilib qurishga harakat qilinadi.

Infologik modelni qurishning asosiy konstruktiv elementlari:

- ✓ mohiyat;
- ✓ mohiyatlar orasidagi bog`lanish;
- ✓ xossalar (atributlar).

**Mohiyat** – har xil obyekt (obyekt -bu biz ularni bir biridan farqlaymiz), uning ma'lumotlari ma'lumotlar bazasida saqlanadi. Mohiyat odam, jiy, samalyit, reyslar, gullar va boshqalar bo'lishi mumkin. Mohiyat turi va mohiyat nusxasi tushunchalari mavjud, ularni bir biridan farqlash kerak. Mohiyat turi tushunchasi bir jinsli narsalar, hodisalar, shaxslar uchun qaraladi. Mohiyat nusxasi tushunchasi aniq narsalar to`plami sifatida qaraladi. Masalan, mohiyat turi SHAHAR bo'lishi mumkin, uning nusxasi esa – TOSHKENT, MOSKVA bo'lishi mumkin.

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

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

Yana bir Misol: AVTOMOBIL mohiyati atributi bu – tur, marka, nomer belgisi, rangi va boshqalar. Bu erda tur bilan nusxa orasida farq mavjud. Rang atributi turi ko`p qiymatli yoki nusxada: qizil, ko`k, oq, va boshqalar.

Mohiyat va atribut turlari orasida absolyut farq yo`q. Mohiyat turi bilan aloqada faqat atribut bir. Boshqacha kintekstda atribut o'zi mustaqil mohiyat bo'lib qatnashishi ham mumkin. Masalan, avtomobil zavodi uchun rang – bu faqat ishlab chiqarish mahsuloti atributi, lak kraska fabrikasi uchun esa rang – mohiyat turidir.

**Kalit** –atributlarning minomal to`plami, uning qiymati yordamida kerakli mohiyat nusxasini topish mumkin. Masalan, samaliyotlar uchishi uchun "Reys



jadvali” mohiyati kalitlari “Reys nomeri” yoki “Jo`nash joyi”, “Uchish vaqti” va “Qo`nish joyi” to`plami bo`lishi mumkin.

**Aloqa** –ikki yoki undan ortiq mohiyatlarning biri bilan o`zaro bog`lanishidir. Agar ma`lumotlar bazasini faqat bir biri bilan bog`lanmagan ma`lumotlar bilan to`ldirishga yo`naltirilsa, u holda uning strukturasi juda oddiy bo`ladi. Lekin, MBni tuzishning asosiy talablaridan biri – bu boshqa mohiyat qiymati bo`yicha bitta mohiyatni izlab tipish imkoniyatini ta'minlashdir. Buning uchun ular orasida aloqani aniqlashni tashkil etish zarur. Real ma`lumotlar bazasida yuzlab yoki minglab mohiyatlar bir biri bilan millionlab bog`lanish o`rnatishi kamdan-kam. Bunday bog`lanishlar to`plami infiligik modelning murakkabligini aniqlaydi.

Ma`lumotlarni Relyatsion modeli asosida munosabat tushunchasi yotadi.

Munosabatni 2 o`lchamli jadvallar yordamida tavsiflash qulay. Jadval tushunarli ko`rimli va inson uchun oddiy bo`lishi kerak. Munosabatlar to`plami ma`lumotlarni saqlash uchun ishlatilishi mumkin. Shu bilan birga ular orasidagi bog`lanishlarni modellashtirish imkonini beradi.

Misol: 2 o`lchamli jadvalni ko`raylik.

Hizmatchi nomeri	F.I.SH	Unvoni	Tug`ilgan yili	Bo`lim	Mutahassis kodi	Mansab	Maosh
2518	Iriliv	t.f.n	1956	1	Amaliy	dots	26000
2567	Kim	t.f.d	1952	2	ASU	dokt	40000
2545	Yugay	p.f.n	1968	1	EVM	prof	60000
3267	Siativ	p.f.n.	1980	3	Amaliy	akad	80000

Yuqoridagi jadvalning 1-satrida atribut nomlari joylashgan.

2-satrida esa atributlarning qisqa nomlari berilgan.

3,4,5-satrlarda esa atribut qiymatlari berilgan

Yuqoridagi ko`rib chiqilgan ierarxik tarmoqli va boshqa ma`lumotlarni tasvirlash usullarini shunday 2 o`lchamli jadvalga keltirish mumkin. Bunday jadvallarning quyidagicha xususiyatlari mavjud.

1. Jadvalning xar bir ma`lumot elementi maydon hisoblanadi va takrorla-nuvchi guruxlarga bo`linmaydi.

2. Barcha ustunlar bir jinlidir.

3. Xar bir ustunga nom tayinlangan.

4. Jadvalda bir xil satr 2 marta uchramaydi.

5. Bunday jadvalda satr va ustunlar ixtiyoriy tartibda qaraladi va ixtiyoriy ketma-ketlikda ishlata olishi mumkin.

Bunday xususiyatli jadvallar munosabat deyiladi. Munosabat asosida qurilgan BB Relyatsion BB deyiladi.

Yuqoridagi jadvalni sxematik qisqartirilgan ko`rinishi xizmatchi (xiz.nom F.I.Sh. unvini, tug`ilgan yili, bo`lim, mut.kodi.mansab, maosh);

Bunday yozuv ma`lumotlar bazasi sxemasi deyiladi, shunday qilib, Relyatsion BB 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 xam  $n$ -darajali deyiladi.

Xar bir atribut qiymatlari to`plami domen deyiladi.

Munosabatda xar bir kortej o`zining kalit identofisatiriga (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 xech qanday atributni kalitdan olib tashlash mumkin emas.

Ekvivalent (soninom) tushunchalar

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

### **Munosabatlar ustida amallar.**

Munosabatlar ustida xar-xil amallar bajariladi. Relyatsion ma'lumotlar modelining xususiyatlaridan biri ma'lumotlarni qayta ishlashini Relyatsion algebra operatorlari (amallari) yordamida amalga oshirishdir. Relyatsion algebrada quyidagi 8 ta operator keng ishlatiladi.

Ulardan 4 tasi an'anaviy to`plamlar ustidagi amallarga kiradi.

An'anaviy (qabul qilingan) amallarni quyidagilar tashkil etadi.

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

Maxsus amallarga esa quyidagilari kiradi:

1. Tanlash (selektsiya)
2. Proektsiya
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 bo`lib:

1. Ulardagi xar bir daraja bir xil bo`lsa yoki ular bir xil atribut to`plamiga ega bo`lsa;
2. Sxema atributlarini shunday tartiblash mumkinki, bir xil o`rinda turib solishtirilayitgan atributlari bir xil domenda aniqlangan bo`lishi kerak.

## Misollar: 1. Birlashtirish amali

Familiyasi	Yoshi
Salimov	20
Ravshaniv	24
Safayev	35
Soatov	42

## Mijoz 1 va Mijoz 2 Birlashtirish operatiri yordamida jadvallar hosilqolish

Familiyasi	Yoshi
Salimov	20
Ravshanov	24
Safayev	35
Siatov	42
Xurramov	52
Ilimov	18
Shavkatov	30

### **Normalashtirish.**

Relyatsion BB munosabatlarida strukturali va semantik axborotlar sakllanishi mumkin. Strukturaviy axborotlarni biz munosabat sxemalari yordamida bilamiz. Semantik axborotlar esa munosabat sxemalarda ma'lum bo'lgan va hisobga olinadigan va atributlar o'rtasidagi funksional bog'lanishlar bilan ifodalanadi. BB sidagi munosabatlarda atributlarning tarkibi ikki 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 oshira olishi kerak.
3. Qo'yilgan BB munosabatlari normallasadi. Munosabatlarni normalashtirish BBda berilgan munosabatlarni dekompozitsiya (ajratish) jarayoni yordamida sodda va kichik munosabatlar hosil qilishdir.

Talaba kodi	Familiyasi	Telefon	Talaba
	Samaniv	4767777	2341717
	Bahodirov	1365556	2341717
	Murodov	1365656	2485888
	G'aniyev	2351717	2485888
	Qambarov	2381817	
	Zokorov	2351817	

Har bir munosabatda kortejlar identofosatir kalitiga ega bo'lishi kerak. Kalit quyidagi ikkita hossaga ega bo'lishi kerak:

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

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

1. To'plamlar ustida birlashtirish, kesishuv, ayirma, dekart ko'paytma va bo'lish amallari kiradi.
2. Maxsus Relyatsion amallar, ularga: proektsiya, 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) Obyektli munosabatlar;
- b) Bog'lanuvchi munosabatlar;

Obyektli munosabatlarda obyektlar haqidagi munosabatlar saqlanadi. Masalan, talaba munosabati. Bog'lanish munosabatlariga asosan, obyektli munosabatlarning kalitlari saqlanadi. Kalit atributlari oddiy va murakkab bo'lishi mumkin. Agar kalit ikkita va undan ortiq atributdan tashkil topgan bo'lsa, murakkab hisoblanadi.

### **Relyatsion algebra va uning amallari.**

Relyatsion BBBTda ma'lumotlar bilan ishlash uchun bir qancha tillar yaratilgan. Ba'zi hollarda bu tillarni ma'lumotlarning qism tillari deb yuritish mumkin. BB bilan ishlovchilar bu tillarni avtomatlashtirish jarayonlarida uch bosqichga bo'lgan holda ish olib borishadi:

- 1) Eng pastki bosqich – kortej deb ataladi. Bunda dasturchi yozuvlar yoki kortejlar 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 murajaat qiladi va mashina bu murajaatni qabul qiladi.

Relyatsion algebra amallarni operandlari sifatida doimiy yoki o'zgarmas va o'zgaruvchan munosabatlar ko'rinishida ishlatadi. Relyatsion algebrada 5ta amal ishlatiladi:

- 1) Birlashtirish,  $R$  va  $S$  munosabatlarni birlashtirish  $R \cup S$  ko'rinishida berilib, bu amalning natijasi  $R$  munosabatga tegishli bo'lgan yoki  $S$  munosabatga tegishli bo'lgan yoki ikkalasiga ham tegishli bo'lgan kortejlar to'plami deb tushiniladi. Bu amallarni bajarayotganda bir xil tartibda bo'lishi kerak.

Natijani tartibi ham operandlar tartibiga teng bo'ladi.

- 2) Ayirma  $R$  va  $S$  munosabatlarning 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 bo'lishi kerak.

3) Dekart ko'paytma. Bizda  $R$  va  $S$  munosabat berilgan bo'lsin.  $R$  munosabatning tartibi  $R$ - $R$  va  $S$  munosabatniki  $S$ - $Q$  ga teng bo'lsin.

Unda dekart ko'paytma  $R * S$  ko'rinishida yozilib, uning natija uzunligi  $R+Q$  ga teng bo'lgan kortejlar to'plamidan iborat bo'lib, bu kortejlarni birinchi  $R$  komponentasi  $R$  kortejga teng bo'ladi, qolgan  $q$  komponentasi  $S$  kortejga teng bo'ladi.

4) Priektsiya,  $R$  munosabatga bu amal tadbiriq etilganda,  $R$  munosabatdan ba'zi bir kimpinentalar olib tashlanadi. Qilganlari esa qaytadan tartiblanadi.

5) Seleksiya tanlash. Bu amal bajarilganda iperandlar sifatida munosabat atributlari ishtirok etadi va solishtirish arifmetik amallari:  $=, \neq, \leq, \geq, <, >$  va mantiqiy amallar:  $\vee$  (U), yoki ( $\vee$ ), nit amallari ishlatiladi.

Relyatsion BBBT da ma'lumotlar bilan ishlashda ishlatiladigan 2ta katta gurux tillari Relyatsion hisoblash deyiladi. Relyatsion hisoblash predikatlarini hisoblashga asoslangan bo'lib ifodalarni yozishga mo'ljallangan qoidalar to'plamidan iboratdir. Ular yordamida biz mavjud munosabatlardan yangi munosabatlar yaratishni ta'minlaymiz.

Bunday ifodalarni yozishda solishtirish amallari, mantiqiy amallar va mavjudlik kvanteri va umumiylik kvanteri ishlatiladi. Hozirgi paytda Relyatsion BBBTning taraqqiyotida yangi til QBE tili ishlatilmoqda. Bu tilda Relyatsion algebra va Relyatsion hisoblashlarda ko'zda tutilmagan bir qancha imkoniyatlar kirgan. Bu tilning xususiyati shundan iboratki, u terminallarda ishlashga muljallangan. So'rovlarni yaratish uchun maxsus ekran redaktiridan, munosabat redaktirlaridan foydalanamiz. QBE tilida foydalanuvchi o'zi olishini mo'ljallagan natijani so'rov ko'rinishida tasvirlaydi va BBBT uni kerakli amallar ketma – ketligiga aylantirib beradi.

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

1. 60- yillarning 2 – yarmida, bunda asosan ierarxik modellarga e'tibir 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, obyektga mo'ljallangan sistema.

### **Normallashtirish. Normal firmalar.**

BB uzluksiz o'zgarib turuvchi jarayon hisoblanadi. Bunda yangi ma'lumot elementlari qo'shiladi. Ular orasida yangi aloqalar yoki bog'lanishlar o'rnatiladi va ularni qayta ishlashni yangi usullari go'llaniladi. Bu jarayonda imkoni biricha foydalanuvchi yaratgan BB bilan ishlash uchun yaratilgan programma ilovasoni kam o'zgartirishga harakat qilinadi. 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 tizimlarini ishlab chiqaruvchilar ma'lumotlarni 3-normal firmada tasvirlab ishlatishni taklif etadilar.

Funksional bog'lanish tushunchasi. Relyatsion BBda ma'lumotlarni strukturasi tashgari ularni sxematik axborotiga ham e'tibor beriladi. BBning strukturasi haqidagi iaxborot munosabat sxemasi yordamida beriladi. Sxematik axborotlar esa atributlar orasidagi funksional bog'lanishlar orqali ifodalanadi. BB munosabatlarida atributlarning tarkibini quyidagi talablarga javob beradigan qolib guruxlash kerak:

- ✓ Atributlar orasidagi zaruriy bo'lmagan takrorlanishlar bo'lmashligi kerak.
- ✓ Atributlarni guruxlaganda ma'lumotlar takrorlanishi minimal darajada qolib 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:

A > V Shaxsiy nomer > Familiyasi  
Mansabi > Maish bog'lanishlari

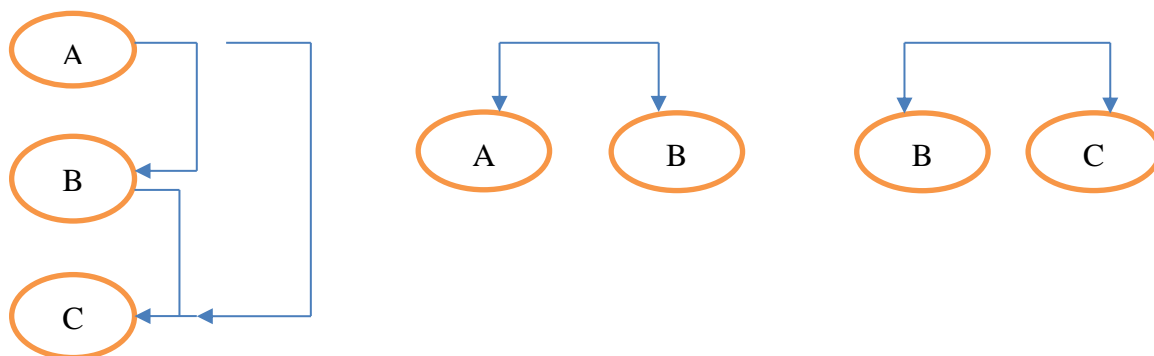
1 – normal firma.

Shaxsiy nomer	Predmet nomi	Siatlar soni	Familiya	Mansabi	Maishi	Kafedra	Tel.
207	Tizimli	100	Jo`rayev	k.o`q	50000	Amaliy matematika	1254789
208	BBBT	158	Jo`rayev	k.o`q	50000	Amaliy matematika	1254789
307	Sinli us	90	Sibiriv	Dits	70000	Amaliy matematika	3112564
308	Pigramm	200	Bo`riyev	Dits	70000	Amaliy matematika	5879632
309	Jar.tadqiq	80	Siativ	Prif	90000	Amaliy matematika	1248796

Agar munosabat 1-normal firmada 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 atributning qismiga bog'langan bo'lsa, u qisman bog'lanishli deyiladi. Bizning Misolda siatlar soni (kalit bo'lmagan atribut) predmetlar nomi atributiga qisman bog'langan. Agar kalit bo'lmagan atribut barcha murakkab kalitga bog'langan bo'lsa, va uning 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 > V bo'lsa, V > S bo'lsa, unda S A dan tranzitiv bog'langan bo'ladi. Bizni Misolda familiya, kafedra, telefon,

uchinchi normal forma (3nf) ko`rinishida qabul qilinadi. Ma'lumotlar munosabatlarida 2nf ga keltirilganda ham bir qancha niqulayliklar bo`ladi. Jumladan, ma'lumotlarda axborotning ortiqchaligi, amallarni bajarish qiyinligi va boshqalar. Bunday munosabatlarni 3nf ga keltiriladi va 3nf da tranzitiv bog`lanish yo`qotiladi. Agar, A, V, S R munosabatlarda 3 ta atribut yoki atributlar to`plami bo`lsin. Agar V atribut A atributga, S atribut esa V atributga bog`langan bo`lsa, ya'ni,  $A > V$  va  $V > S$  bunda teskari bog`lanishlar bo`lmasa, unda S atribut A atributga tranzitiv bog`langan deyiladi. Uni ko`pincha diagramma ko`rinishida quyidagicha belgilaymiz:

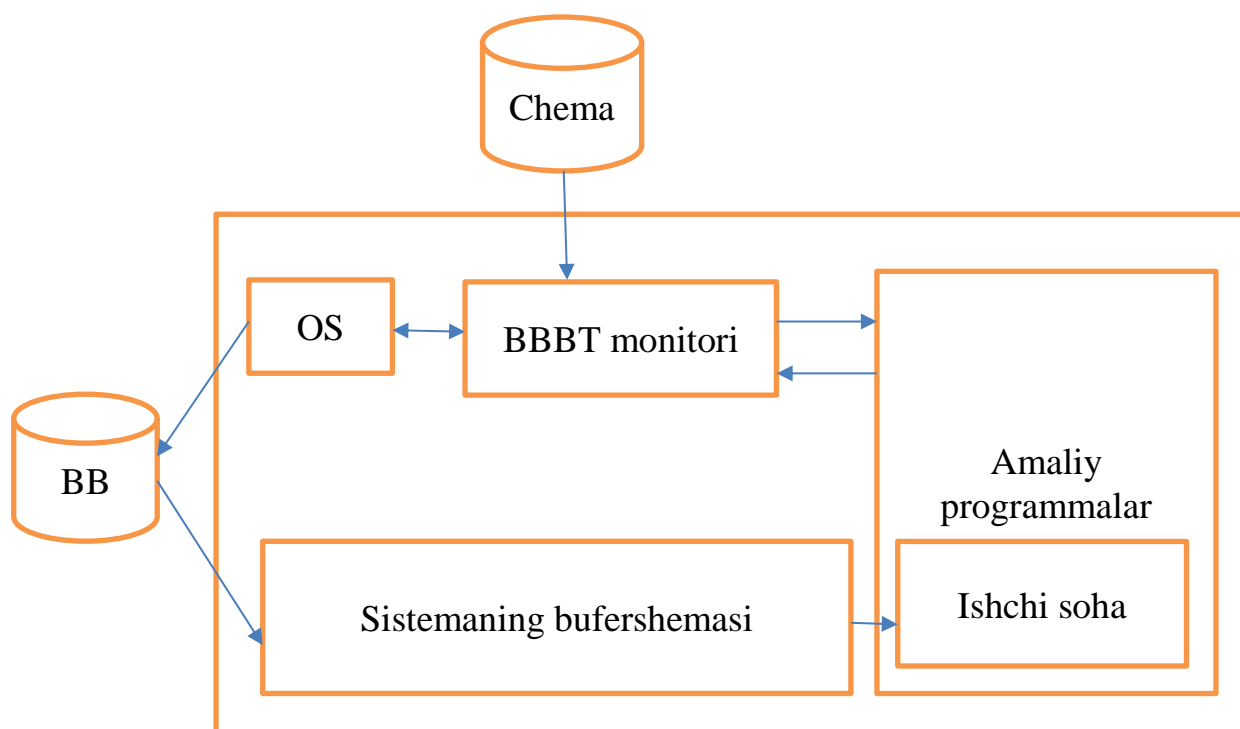


Masalan, xizmatchi (nomer, nomi, maish, liyiha\_nomeri, tugash sanasi).  
 hizmatchi (hiz-nomeri, hiz nomi, maish, liyixa nomeri).Liyiha (liyiha nomeri, tugash sanasi).

Shunday qilib, R munosabat 3nf da berilgan deyiladi, agarda, u 2nf da bo`lsa va R munosabatdagi birlamchi kalit bo`lmagan har bir atribut R munosabatni har bir mumkin bo`lgan kalit atributiga nitranzitiv bog`langan bo`lsa. Umuman ilganda normallashtirish jarayoni va munosabatni 3nf ga keltirish quyidagi bosqichlardan iborat bo`ladi:

1. Ma'lumotlarni ixtiyoriy strukturasiidan oddiy strukturali ikki o`lchamli jadvallarga o`tish va 1nf ni hosilqolish.
2. Kalit atributlari bilan barcha atributlar orasidagi mumkin bo`lgan to`liqmas funksional bog`lanishlarni yo`qotish va 2nf hosilqolish.
3. Mumkin bo`lmagan kalit atributlari va asosiy bo`lmagan (nikalit) atributlar orasidagi tranzitiv funksional bog`lanishlarni yo`qotish va 3nf ni hosilqolishni amalga ishiradi.

BBBT kimpinentalari va ularni IS va amaliy programmalar bilan o`zari bog`liqligi ma'lumotlarni fizik tasvirlashda muhim o`rin tutadi. BBBT murakkab til programma kompleksidan iborat bo`lib, BB ni ishlash imkoniyatini ta'minlaydi. BBBT tarkibiga sistemali pirammalar kompleksi kiradi. Bu kompleksni markaziy komponentasi manitor yoki boshqaruvchi programma hisoblanadi. manitor BBBT ni komponentalarini IS va amaliy programmalar bilan o`zari 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) ning bitta operatorini bajarishga tegishli bo'lgan amallar ketma-ketligi ko'rsatilgan.

Masalan, bu BBga ma'lumotlarni o'qish va so'rov operatori bo'lib hizmat qilsin. Unda nomerlangan strelkalar quyidagi ma'noga ega bo'ladi:

- 1) Amaliy programmalar BB ga (YaMD) operatori orqali murajaat qilsin. U manitor tomonidan tahlil qilinadi.
- 2) Talqin qilish jarayonida manitor oldindan translyatsiya qolib qo'yilgan sxemani ishlatadi.
- 3) Bu so'rovga tegishli ma'lumotlar aniqlanib bo'lingandan keyin, manitor IS ga tashqi xotiraga murajaat qilishni amalga oshirish talabi bilan murajaat qiladi.
- 4) IS BBga murajaatni bajaradi. Bu xuddi fayllarga murajaat qilish kabi oddiy bajariladi.
- 5) Talab qilingan ma'lumotlar tashqi xotiradan sistemani bufer sohasiga uzatiladi.
- 6) Ma'lumotlar amaliy programmalarini ishchi sohasiga jo'natiladi.
- 7) manitor amaliy programmaga so'rovni bajarish natijalar xabarini beradi.
- 8) Amaliy programma BB dan olingan ma'lumotlar ustida kerakli amallarni bajaradi.

**Adreslash usullari.** Bitta mashina ko'rsatmasi yordamida o'qish mumkin bo'lgan bitlar guruhi fizik yozuvlar deb ataladi. Fizik yozuvlar mashina xotirasining yacheykalarida saqlanadi va mashina adreslari yordamida identifikatsiyalanadi. Programmalar mantiqiy yozuvlarni kalitlar yordamida aniqlaydi. Programma uchun zarur bo'lgan ma'lumotni mantiqiy yozuv kalitlari yordamida fizik yozuvlarning adreslarini aniqlaydi.

Programma uchun zarur bo'lgan ma'lumotni mantiqiy yozuv kalitlari yordamida fizik yozuvlarning 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 quyidagi usullari mavjud:

1) Fayllarni ketma – ket saqlash usuli. Har bir yozuvning kaliti tekshiriladi. Bunday usul ko'p vaqtini talab etadi.

2) Blokli qidirish. Agar yozuvlar kalit bo'yicha tartiblangan bo'lsa, fayllarni skanerlashda har bir yozuvni o'qib chiqish talab etilmaydi. Bunday xollarda kerakli yozuvlarni tipish 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 qidirishni to'g'ridan – to'g'ri murajaat qurilmalarida ishlatib bo'lmaydi.

## **Ma'lumotlarning relyatsion strukturasi**

Ma'lumotlarni qayta ishlash uchun to'plamlar nazariyasi ishlatiladi (birlashma, kesishma, farqlash, dekart ko'paytma). Matematikadan ma'lumki har qanday ma'lumotlarni tasvirlash ikki o'lichivli maxsus turli bog'lanishli -relyatsion jadvallar to'plamiga keladi.

Relyatsion model ma'lumotlarining eng kichik birligi–bu model ma'lumotlar qiymati uchun alohida atamadir.

Bir xil turdagi atimar qiymatlar to'plamiga domen deyiladi. Masalan, Reys nomeri domeni- butun musbat sonlar to'plami.

**Domen** ma'nosi quyidagilardan iborat. Agar ikki atribut qiymatlari bir domendan olingan bo'lsa, unda bu ikkita ishlatiladigan atributlarni taqqoslash ma'nosi bir. Agar ikkita atribut qiymatlari har xil domenlardan ilingan bo'lsa, ular taqqoslash ma'ni bermaydi. Masalan, reys nomeri bilan chipta narxini solishtirish mumkinmi.

**Munosabat**  $D_1, D_2, \dots, D_n$  domenlarda (ularning bari har xil bo'lishligi shart emas) sarlavha va tanadan iborat bo'ladi.

**Sarlavha**  $A_1, A_2, \dots, A_n$  fiksirlanga atributlar to'plamidan iborat bo'lib, ular bilan bilan, y'ani  $A_i$  atributlar va ularni aniqlivchi  $D_1$  domenlar orasida o'zaro bir xil moslik mavjud.

**Tana** vaqt bo'yicha o'zgaruvchi kortejlar to'plamidan iborat bo'ladi. Bu yerda o'z navbatida kortrej juft atributlar qiymatlari  $(A_i:V_i)$ ,  $(i=1,2,\dots,n)$  iborat bo'ladi.

**Munosabat darajasi** – bu uning atributlar soni. munosabat darajasi bitta bo'lsa unar deyiladi. Ikkita daraja bo'lsa– binar deyiladi. Uch daraja bo'lsa ternar va hokoza.

**Kardinal son** yoki **munosabat quvvati** - bu uning kortejlar soni. Kardinal munosabatlar soni vaqt bo'yicha o'zgaradi.

Munosabat - bu to'plamlardir. To'plam esa ta'rif bo'yicha mos elementlarni o'ziga olmaydi. Aytaylik  $R$  munosabat  $A_1, A_2, \dots, A_n$  atributlarga ega. Aytiladiki,  $R$  munosabat atributlar to'plami  $K=(A_i, A_j, \dots, A_k)$  kalit bo'lishi mumkin qachon kim ikkita vaqtga bog'liq bo'lmagan shart bajarilsa:

**Unikallik:** Ixtiyoriy berilgan vaqt momentida ikkita har xil kortejlar  $A_i, A_j, \dots, A_k$  lar uchun bir xil qiymatga ega emas.

**Minomallik:**  $A_i, A_j, \dots, A_k$  atributlarda birortasi unikallikgi buzulmasa  $K$  dan o'chira olishi mumkin emas.

Ko'pgina relyatsion BBT foydalanuvchilar uchun quyidagi ekvivalent tushunchalarini ishlatish maqsadga muvofiqdir:

Munosabat – jadval (ayrim holda fayl);

Kortej – satir (ayrim holda yozuv);

Atribut – ustun, maydon.

Ko'p hollarda “Yozuv” ni “Yozuv nusxasi” deb, “Maydon” esa “Maydon nomi va turi” deb qabul qilinadi.

Relyatsion ma'lumotlar bazasi – bu ma'lumotlarni o'z ichiga oluvchi munosabatlar to'plami va ular  $BB$  da saqlanishi kerak. Foydalanuvchi bunday  $BB$ ni jadvallar to'plami deb qabul qilishi kerak, ya'ni.

1. Har bir jadval bir xil turdagi satirdan iborat bo'lib u unikal (yagona) nomga ega bo'lishi kerak;

2. Har bir satir fiksirlangan maydonlar va qiymatlariga ega;

3. Jadval satri bir biri bilan kamida biror bir qiymat bilan farq qiladi.;

4. Jadval ustuni bir qiymatli nomga yuboriladi va ma'lumotlarning bir jinsli qiymatlari joylashtiriladi (sana, familiya, butun son yoki pul belgisi yig'indi);

5. Ma'lumotlar bazasida joylashgan ma'lumotlar aniq qiymatlar ko'rinishida tasvirlanadi va ma'lumotlarni tasvirlashning bunday usuli yagona bo'lib hisoblanadi. Umuman olganda jadvallarni bir biri bilan bog'lashning qaysidir maxsus “aloqa” vositasi yoki ko'rsatmasi yo'q.

6. Jadval satirlari va ustunlari bilan ish yuritganda ma'lumotlarning qanday joylashishiga qaramasdan ular qayta ishlanishi mumkin. Bunga ko'proq jadvallarning nomi va ularning ustunlari soni imkon beradi.

Ma'lumotlarni saqlash uchun jadvallar sonini minomallashtirishga intilish  $BB$ ni yangilashda har xil muammolarga olib kelishi mumkin. Ma'lumotlarning relyatsion modeli uchun relyatsion algebra - munosabat amalida ishlashga qulay vosita mavjud.

Relyatsion algebraning barcha amallarini bajaradigan ma'lumotlar ustida monipulyatsiya (ish olib boradigan) qiladigan tillar yaratilgan. Bu tillar orasida *SQL (Structured Query Language – strukturalashtirilgan so'rov tili)* va *QBE (Quere-By-Example – namuna bo'yicha so'rov)* tillari eng ko'p tarqalgan. Bu ikki til ham yuqori darajali til bo'lib, ular yordamida foydalanuvchi kerakli ma'lumotlar bilan ish olib borishi mumkin.

## **Relyatsion ma'lumotlar bazasini loyihalash**

**Sohalar ma'lumotlar bazasi** – bu katta tashkilotlar axborot tizimlari hisoblanib, ular o'zida bir necha o'nlab BBni saqlaydi. Bu BBning har xil bo'limlarida joylashgan o'zaro bog'langan kompyuterlarda joylashgan bo'ladi.

**Amaliy ma'lumotlar bazasi** – bu bir yoki bir necha amaliy masalalarni yechish uchun zarur bo'lgan berilganlarni birlashtiruvchi ma'lumotlar bazasidir (masalan, moliya, talabalar, o'qituvchilar va boshqalar haqidagi berilganlar bo'lishi mumkin).

Sohalar ma'lumotlar bazasi har qanday joriy va kelgusi ilovalardan foydalanishni ta'minlab beradi. Uning ma'lumotlar elementlari amaliy ma'lumotlar elementlari to'plami bazasiga ham kiradi.

BBni loyihalashtirishning joriy va oldindan ko'rilgan ilovalariga asoslanib yuqori effektli axborot tizimlarini yaratishni yanada tezlashtirish mumkin. Shu sabab amaliy loyihalash BBni ishlab chiqaruvchilarni o'ziga jalb etmoqda. Bunday axborot tizimlarida ilovalar sonining oshib borishi amaliy BB sonini tez o'stirib yubormoqda.

Shunday qilib har bir qarab chiqilgan loyihalashga bo'lgan usul har xil yo'nalishdagi loyihalashtirish natijasiga ta'sirini ko'rsatadi.

Maqsadga erishish uchun loyihalash metodologiyasini tashkil etishda sohali va amaliy usullarni qo'llash effektivligiga bog'liqdir. Umuman olganda sohali usul boshlang'ich axborot strukturasi qurish uchun ishlatiladi, amaliy usul esa uni rivojlantirish maqsadida ma'lumotlarni qayta ishlash effektivligini oshirishda ishlatiladi.

Axborot tizimlarini loyihalashda bu tizimlarni to'liq analiz qilish va unga bo'lgan foydalanuvchilar talablarini aniqlashga olib keladi. Ma'lumotlarni yig'ish mohiyatini o'rganish bilan boshlanadi.

BBni loyihalashning asosiy maqsadi – bu saqlanadigan ma'lumotlarni kamaytirish, ishlatiladigan xotira hajmini tejash va ko'p qaytariladigan operatsiyalarni kamaytirishdir.

### **Ma'lumotlarni normallashtirish**

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

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

takrorlanuvchi ma'lumotlar guruhiga ega jadvallar relyatsion ma'lumotlar bazasiga kiritilmaydi.

Har qanday normallashtirilgan jadval avtomat ravishda birinchi normal formada qisqacha 1NF, deb hisoblanadi. Shunday qilib, umuman olganda, "normallashtirilgan" va "1NF da joylashgan" bitta manoni anglatadi. Lekin amaliyotda "normallashtirilgan" atamasi ko'prik ma'nosida ishlatiladi – "to'liq normallashtirilgan", ya'ni proyektda normallashtirishning hech qaysi prinsipi buzilmayapti degan manoda.

Keyingi pog'onadagi normallashtirishlarni ko'rib chiqamiz – ikkinchi normal forma (2NF), uchinchi normal forma (3NF) va hokozo. Umuman olganda, jadval 1NF da bo'lsa va undan tashqari yana bir qo'shimcha shartni qoniqtirsa, uning ma'nosi keyinroq ko'rib o'tiladi. Jadval 3NF da deyiladi, qachonki u 2NF da bo'lsa va yana bir qo'shimcha shartni qoniqtirsa va hokozo.

Har bir normal forma qandaydir ma'noda oldingisiga qaraganda ancha chegaralangan, lekin ma'qulroqdir. Bu shunga bog'likki, "N-chi normal forma" ega bo'lgan ba'zi yoqimsiz tomonlariga "(N+1)-chi normal forma" ega emas.

N-chi normal formaga qarab (N+1)-chi normal formaga qo'yilgan qo'shimcha shartning umumiy ma'nosi ana shunday yoqimsiz tomonlarini yo'qotishdan iborat.

Normallashtirish nazariyasi jadval maydonlari orasidagi u yoki bu bog'liqlikning birligiga asoslanadi. Bunday bog'liqliklarning ikki turi aniqlangan:

- funksional
- ko'p qiymatli

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

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

**Ko'p qiymatli bog'liqlik:** A maydon huddi shu jadvalning B maydonini ko'p qiymatli deb aniqlaydi, agar A maydonning har bir qiymatiga mos ravishda B maydonning aniqlangan qiymatlari bir bo'lsa.

Ko'p funksional bog'liqlikga Misol "O'qish jadvali"

Fan	Muallim	Darslik
Dasturlash	Abdullayev I.R.	Liberti Dj . Isviy samostiyatelni C++
Dasturlash	Raxminiv U.K.	Firsayt R. Pascal dlya vsex
Dasturlash	Kamolov N.N.	Liberti Dj. Isviy samostiyatelni C++
Dasturlash	Kamolov A.N.	Firsayt R. Pascal dlya vsex
Dasturlash	Raxmonov U.K.	Faysman K. Prifessionalniye programmirivaniye na Paskale

Jadvalda ko'p qiymatli bog'liqlik bir "Fan-Muallim": Fan (Dasturlash Misolida) bir nechta muallim tomonidan o'qilishi mumkin (Misolda Abdullayev, Raxmonov, Komalov). Yana boshqa bog'liqlik ham bir "Fan-Darslik":

informatikani o`rnatishda “Pascal dlya vsekh”, ”Izuchay samostayatelno C++” va “Professionalniye programmirivaniye na Pascale” darsliklari ishlatiladi. Bunda Muallim va Darslik funksional bog`lanmagan, bu esa ortiqchalikka olib keladi (yangi darslikni yozish uchun yana ikkita yangi satir yozish kerak bo`ladi). Bu jadvalni ikkiga ajratganda ish yaxshilanadi: (Fan-Muallim va Fan-Darslik)

### **Ma`lumotlar banki tushunchasi va uning tarkibiy qismlari.**

Axborot bilan ta`minlash tizimlarining rivojlantira olishi foydalanuvchilarning axborotga bo`lgan ehtiyojlarini qondirishda avtomatlashtirilgan ma`lumotlar bankiga o`rnatish imkoniyatini yaratdi.

Ma`lumotlar banki – zaruriy axborotni olish maqsadida ma`lumotlarni markazlashgan holda saqlash va jamoa bo`lib foydalanish uchun mo`ljallangan axborotli, tematik, dasturiy, tillar, tashkiliy va texnik vositalarining tizimi sifatida ifodalanishi mumkin.

Ma`lumotlar banki va uning ishlashi va qo`llana olishini ta`minlab beruvchi dasturiy to`plamlari bilan birgalikda avtomatlashtirilgan ma`lumotlar banki deb ataladi.

Avtomatlashtirilgan ma`lumotlar banki inson-mashina tizimidan iborat bo`lib, ichki foydalanuvchilar hamda kompyuter, kommunikatsiya texnikasi asosida axborotni zaruriy qayta ishlash va uzatishni amalga ishiruvchi texnologiyaning axborot jarayonlarini o`z ichiga iladi. Ma`lumotlar bankiga quyidagi asosiy talablar qo`yiladi:

- ✓ muammoli sohaning holatiga axborotning mos kelishi;
- ✓ ishlashning ishonchliligi;
- ✓ tezkorligi va unumdorligi;
- ✓ foydalanishning oddiy va qulayligi;
- ✓ foydalanishning ommaviyligi;
- ✓ axborotning himoyalanganligi;
- ✓ kengaytirish imkoniyatlarining mavjudligi.

Ma`lumotlar bankining tarkibi unga yuklatilgan vazifalar va yechaolishi kerak bo`lgan masalalarning xususiyatlaridan kelib chiqqan holda aniqlanadi.

Ma`lumotlar bankining asosiy vazifalari quyidagilardan iborat:

- ✓ axborotni saqlash va uni himoyalashni tashkil etish;
- ✓ saqlanilayotgan ma`lumotlarning davriy dolzarbligini ta`minlash;
- ✓ foydalanuvchilar va amaliy dasturlarning so`rovlari bo`yicha ma`lumotlarni izlash va tanlash;
- ✓ olingan ma`lumotlarni qayta ishlash va belgilangan shaklda natijalarni chiqarish.

Ma`lumotlar bankining tarkibiy elementlari satiriga quyidagilar kiradi:

- 1) bir yoki bir necha ma`lumotlar bazasi;
- 2) berilganlar bazalarini boshqarish tizimi(BBBT);
- 3) masalalarni echishning amaliy dasturlar to`plami;

Ma`lumotlar bazasi – ma`lumotlar bankining eng muhim tarkibiy qismi

hisoblanadi. Umumiy holda ma'lumotlar bazalariga keyingi tushunchani keltirish mumkin. Ma'lumotlar bazasi – zaruriy axborotni tezkor olish va modifikatsiyalash, axborotning minimal ortiqchaligi, amaliy dasturlarga bog'liq emasligi, izlash usulining umumiy boshqara olish imkoniyatlariga ega, katta miqdordagi ilovalar uchun foydalanish imkoniyatlari bilan tavsiflanuvchi o'zaro bog'liq ma'lumotlar yig'indisidir.

Ma'lumotlar bazasini qo'llash ushbu imkoniyatlarni beradi:

- ✓ so'rovlar majmuasini amalga oshirishni soddalashtirish;
- ✓ saqlanilayotgan ma'lumotlarning ortiqchaligini kamaytirish;
- ✓ axborot texnologiyalaridan foydalanish samaradorligini oshirish;
- ✓ ma'lumotlar dolzarbligini ta'minlash;
- ✓ amaliy dasturlarni o'zgartirmasdan ma'lumotlarni o'zgartirish;
- ✓ ma'lumotlar tuzilmasini qat'iyligini saqlash.

Ma'lumotlar bazasini samarali tashkil etish bu - ma'lumotlar bazasida mantiqan o'zaro bog'langan massivlarni tashkil etishdir.

Ma'lumotlar bazasini boshqarish uchun uni tashkil etish va yuritishda ixtisoslashgan samarali dasturiy vosita - ma'lumotlar bazasini boshqarish tizimidan foydalaniladi.

Ma'lumotlar bazasi qaytarilmaydigan ma'lumotlarning yaxlitlangan jamlamasi sanaladi. Uning asosida mazkur sohaning barcha masalalari hal etiladi. Ma'lumotlar bazasida ko'p qirrali kirish va aynan bir xil ma'lumotlardan turli foydalanuvchilar foydalanish imkoni mavjud.

Tashkil etiladigan ma'lumotlar bazasining tuzilmasi muammo sohasi ma'lumotlarining axborot mantiqiy modelini aks ettirishi lozim. Ma'lumotlar bazasidagi mantiqiy o'zaro bog'liqlik ma'lumotlar modeli turiga muvofiq tashkil etiladi.

Normativ-ma'lumotnomali va boshqa ma'lumotlar qoida bo'yicha alohida massivlarda joylashtiriladi. Bu massivlarni yuzaga keltirish va yuritish texnologiyasi u o'ziga xos xususiyatlarga ega. Mazkur massivlar ma'lumotlar bazasining boshlang'ich yuklanish bosqichida tashkil etiladi.

Operativ(tezkor) hisobot ma'lumotlari ma'lumotlar bazasiga vazifalarni echish reglamentiga muvofiq kiritiladi. Bu ma'lumotlar ma'lum davrda to'planadi. Ma'lum bir hisob-kitob qilingach (masalan, omborxonada qolgan tovarlarni hisob-kitob qilish), to'plangan tezkor hisobot ma'lumotlari yo'q qilinadi yoki arxivda saqlanadi.

Ma'lumotlar bazasi foydalanuvchining bitta shaxsiy kompyuterida monopol joylashgan bo'lishi mumkin. Bunday holatda u faqat ma'lum SHK tashqi xotirasiga joylashtiriladi va axborot bazasiga bir vaqtda bir necha foydalanuvchining kirishi ta'minlanadi.

Kompyuter tarmog'i mavjud bo'lgan holda ko'pchilik foydalaniladigan rejimda ishlash, «mashina-server»da joylashadigan markazlashgan ma'lumotlar bazasidan foydalanish imkoniyati tug'iladi. Bunday holatda har bir foydalanuvchi o'z shaxsiy kompyuteri orqali barcha foydalanuvchilar uchun umumiy bo'lgan markazlashgan axborot bazasiga kirishga ruxsat iladi.

Tarmoq texnologiyasida har bir foydalanuvchi o'z SHKda lokal ma'lumotlar

bazasini tashkil etishi ham mumkin. Bu ma'lumotlar bazasi faqat mazkur avtomatlashgan ish joyi uchun zarur bo'lgan axborotni saqlaydi. Kompyuter tarmog'ida ma'lumotlar bazasini tashkil etish va jamoa bo'lib foydalanish banklar, birja, investitsiya jamg'armalari va boshqa bozor iqtisodiy tuzilmalarining axborot tizimlari samaradorligini oshiradi.

Foydalaniladigan texnik va dasturiy vositalar konfiguratsiyasiga bog'liq holda turli ishlash texnologiyasi amalga ishiraolishi mumkin. Ma'lumotlarni tarmoqli ishlab chiqishning turli kontseptsiyalari mavjud, Misol tariqasida «fayl-server» va «mijozserver»ni keltirish mumkin.

«Fayl-server» kontseptsiyasi tarmoq operatsion tizimi yadrosi hamda markazlashgan holda saqlanuvchi fayllar joylashgan va «fayl-server» uchun ajratilgan kompyuterga mo'ljallangan. «Fayl-server»dagi ma'lumotlarning umumiy bazasiga jamoa holda kirish unga xos xususiyatdir. Foydalanuvchilardan biri tomonidan fayl yangilangan taqdirda boshqa foydalanuvchilarning kirishiga qarshi himoyalanaadi.

So'ralgan ma'lumotlar «fayl-server» dan ishchi stantsiyaga o'tkaziladi va ular ma'lumotlar bazasini boshqarish tizimi vositalari bilan qayta ishlanadi.

«Mijoz-server» kontseptsiyasi. Mazkur kontseptsiyaga asosan, ma'lumotlarni ishlab chiqish vazifasi mijoz - ishchi stantsiyasi va ma'lumotlar bazasining mashina serveri o'rtasida bo'lishi mo'ljallangan. Ma'lumotlarni qayta ishlashni mijoz so'raydi va u tarmoq bo'ylab ma'lumotlar bazasi serveriga uzatiladi. So'rov ma'lumotlari o'sha erda qidiriladi. Qayta ishlangan ma'lumotlar tarmoq orqali serverdan mijozga uzatiladi. «Mijoz-server» arxitekturasi uchun xos xususiyat - bu ma'lumotlar bazasidan so'rov uchun SQL dasturlash tilidan foydalanish. Bu tarmoq mijozlarining turli xildagi umumiy ma'lumotlar bilan ishlash imkonini beradi

Ma'lumotlar bazalarini boshqarish tizimi – bu ma'lumotlar bazasini yaratish, yuritish va foydalanish uchun mo'ljallangan dasturiy, tillar vositalarining to'plamidir.

Ma'lumotlar bazasiga murojzat qilish bilan bog'liq barcha jarayonlarni avtomatlashtirish uchun mo'ljallangan boshqarish dasturi – o'zagi BBBTning asosiy tarkibiy qismi hisoblanadi.

BBBTning tarkibiga ma'lumotlarni qayta ishlash dasturlarining to'plami kiradi. Bu to'plamni tarjimonlar (translyatorlar), talab va dasturlash tillari, muharrirlar, servis dasturlari va boshqalar tashkil qiladi.

BBBTning asosiy vositalarini quyidagilar tashkil etadi:

- ✓ ma'lumotlar bazalari tuzilmalariga topshiriq berish (tasvirlash) vositalari;
- ✓ ma'lumotlarni kiritish, ko'rish va muloqatlar rejimida ishlashga mo'ljallangan darcha shakllarini loyihalash vositalari;
- ✓ berilgan sharoitlarda ma'lumotlarni tanlash uchun talablar yaratish, shuningdek, ularni ishlash bo'yicha operatsiyalar bajarish vositalari;
- ✓ ishlov natijalarini foydalanuvchiga qulay ko'rinishda bosmaga chiqarish uchun ma'lumotlar bazasi ma'lumotlaridan hisobot yaratish vositalari;
- ✓ til vositalari - makroslar, qurilgan algaritmik til (Dbase, Visual Basic yoki boshqalar), talablar tili (QBE- Query Example, SQL) va h.k. Ular

ma' lumotlarni ishlashining nostandart algaritmlarini, shuningdek foydalanuvchi topshiriqlaridagi voqealarni ishlash protseduralarini bajarish uchun qo'llaniladi.

SHunday qilib, ma'lumotlar banki bir necha ma'lumotlar bazasi, boshqarish va amaliy dasturlardan tashkil topadi. Bu elementar AT ga yuklatilgan vazifalarni bajarishda asosiy rol o'ynaydi. SHu bilan birga, ma'lumotlar bankining samarali faoliyati uning ta'minlovchi elementlariga ham bog'liqdir. Bu ta'minot tarkibiga quyidagi elementlar kiradi.

Texnik ta'minot ma'lumotlar banki va foydalanuvchilarning ish faoliyatini avtomatlashtirish imkoniyatini yaratadigan texnik vositalardan tashkil topadi. Bunday vositalar jumlasiga EHM, tashqi qurilmalar, axborotni tashish, uzatish vositalari, aloqa tarmoqlari, abonentpunktlari va boshqalar kiradi.

**Matematik ta'minot** - funksional masalalarni echish va Ma'lumotlar bankini boshqarish usullari, matematik modellar va algaritmlar to'plamidan tashkil topadi.

**Dasturiy ta'minot**-Ma'lumotlar bankining faoliyatini amalga oshirish dasturlari va turli xil qo'shimcha vazifalarni bajarish uchun mo'ljallangan servis dasturlarning

to'plamidan iborat bo'ladi.

**Axborot ta'minoti**-ma'lumotlarni turkumlash va ixchamlashtirish, ifodalash va taqdim etish tizimlaridan tashkil topadi.

**Lingvistik ta'minot** -BBBTda foydalaniladigan tillar, lug'atlar majmuasi orqali tashkil qilinadi.

**Tashkiliy ta'minot** - Ma'lumotlar bankining kundalik faoliyatini ifodalovchi rasmiy hujjatlar, me'yoriy ko'rsatmalar to'plamidan iborat bo'ladi.

Yuqorida aytib o'tilgandek, ma'lumotlar banki inson-mashina tizimi hisoblanadi, ya'ni axborot tizimidir. Hozirgi kunda ma'lumotlar banklari inson faoliyatining turli sohalarida, masalan, iqtisodiyot tarmoqlarini boshqarishda, ilmiy tadqiqot ishlarini boshqarishda, ommaviy axborot xizmatlarini ko'rsatishda qo'llanilmoqda. Bunda ma'lumotlar bankini axborot tizimi sifatida faoliyat yuritishida quyidagi ikki shaklning biridan foydalaniladi:

Axborot tizimidan (AT) avtonom foydalanish. Bu shaklda AT boshqa tizim tarkibiga kirmaydi, balki mustaqil faoliyat ko'rsatadi. Bunga, masalan, samalyot va temir yo'l chiptalarini sotish tizimlari («Sirena», «Ekspress»), talab bo'yicha tegishli hujjatlarni tayyorlovchi axborot - qidirish tizimlari va boshqa turdagi axborotli xizmat ko'rsatish tizimlari Misol bo'laoladi.

AT dan yirik boshqarish tizimining tarkibiy qismi sifatida foydalanish.

Bunda hosil qilingan ma'lumotlar va chiquvchi axborotdan tizimning boshqa bo'limlarining faoliyatida ham foydalaniladi. Bunday AT ga, masalan, axborot o'qitish tizimlari, loyihalashtirishning avtomatlashtirilgan tizimlari, boshqarishning avtomatlashtirilgan axborot tizimlari Misol bo'ladi.

SHu kabi axborot tizimlarini bir qancha belgilarga ko'ra turkumlarga ajratish mumkin. Quyidagi rasmda AT ni turkumlashning asosiy belgilari ko'rsatilgan.

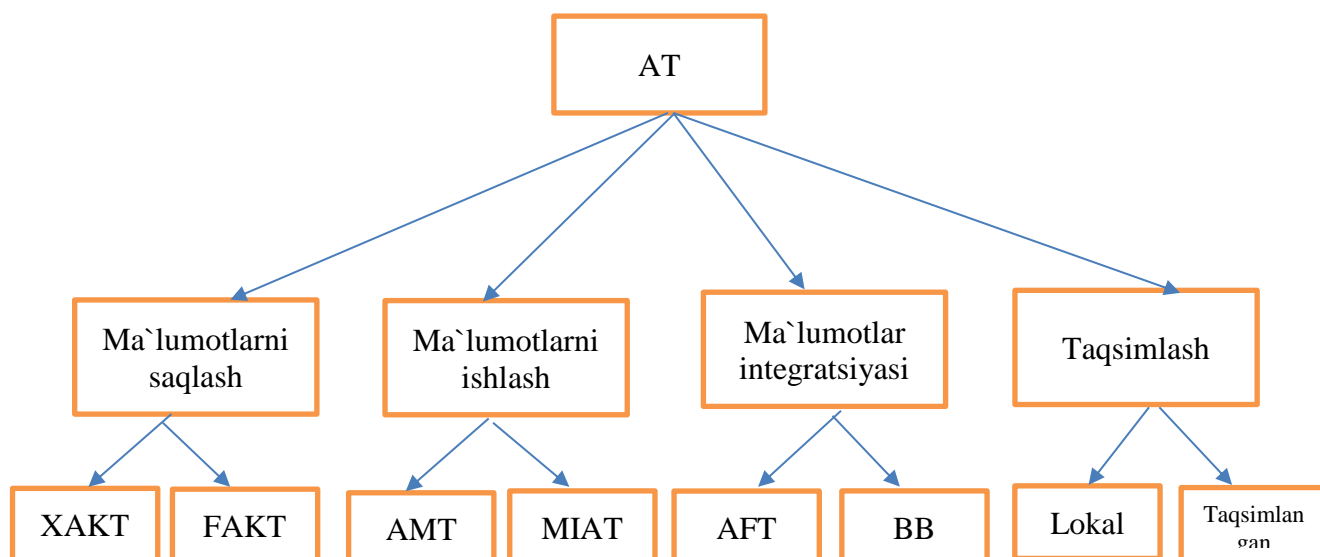
Hujjatli axborot qidirish tizimi (XAQT) hujjatlashtirilgan ma'lumotlarni saqlash va qayta ishlashni amalga ishiraadi. Kutubxona faoliyatining avtomatlashtirilgan tizimi XAQT ga Misol bo'ladi.



Faktografik axborot qidirish tizimi (FAQT) raqamli va mantli ma'lumotlarni saqlashda va qayta ishlashda qo'llaniladi. Tashkil qilinayotgan AATning asosiy qismi

FAQT turidagi tizimga Misol bo'ladi.

Ma'lumotlarni ishlash usuliga ko'ra AAT ikki qismga: axborot -ma'lumotnoma tizimi (AMT) va ma'lumotlarni ishlashning avtomatlashtirilgan tizimi (MIAT)ga bo'linadi.



AMT talab-javob tartibida ishlaydi. Bunday tizimda tegishli axborotlar talab bo'yicha qidiriladi va foydalanuvchiga qayta ishlamagan holda beriladi. Ikkinchi turdagi tizimda esa topilgan ma'lumotlar tegishli dasturlar yordamida ishlanadi va foydalanuvchiga beriladi.

Ma'lumotlarni integratsiyalashtirish darajasiga ko'ra AT avtonom va ma'lumotlar bazasidan tashkil topgan turlarga bo'linadi. Avtonom fayli tizimlarda (AFT) to'plangan ma'lumotlar o'zaro bog'lanmagan holatda bo'ladi. SHu sababli bunday turdagi tizimlar o'rniga ma'lumotlar bazasidan (MB) foydalanilmoqda.

Taqsimlash darajasiga ko'ra AT elementlari bitta EHMda (lokal) va hisoblash tarmog'ida (taqsimlangan) joylashgan turdagi tizimlarga bo'linadi.

### **Axborotlar bazasini tashkil etishning dasturiy vositalari.**

Umumiy holda axborot bazasini yuritishning dasturiy vositalarga servis dasturiy vositalar, umumiy maqsad uchun universal amaliy dastur vositalari va ixtisoslashgan amaliy dasturlar kiradi.

Berilganlar bazasini boshqarish tizimi (BBBT) – axborotlar bazasini yaratish va yuritish uchun alohida ahamiyatga ega. BBBT umumiy maqsadlarga mo'ljallangan universal amaliy dasturiy vositalarga mansub. BBBT- bu mashina tashuvchida mantiqiy o'zaro bog'langan ma'lumotlarni tashkil etish va yuritish

uchun mo`ljallangan nisbatan keng tarqalgan va samarali universal dasturiy vosita sanaladi.

BBBT yagona ma' lumotlar bazasida dubl qilinmaydigan ma' lumotlarni integratsiyalashni va ulardan ko`p maqsadlarda foydalanishni, bazadagi barcha ma' lumotlar butunligini va to`g`riligini ta' minlaydi va ma' lumotlarga ruxsatsiz kirishdan himoyalaydi.

BBBT berilganlar bazasidan so`rovlarni tashkil etishning moduli dasturiga ega bo`lmagan foydalanuvchiga mo`ljallangan qulay vositalarga ega.

BBBT asosida AB (axborot bazasini)ni ishlab chiqish, ma' lumotlar bazasini tashkil etish tuzilmasi bo'yicha masalalarni tayyorlashga mo`ljallangan. Bu masalalar bevosita axborot ta`minotining mashina tashqarisidagi sohasi bilan bog`langan. Ishlab chiqilgan ma' lumotlar bazasiga muvofiq uni BBBT vositalari bilan tashkil etish va uni ishga tushirish amalga oshiriladi.

Axborotlarni kiritish va nozorlashning mahsus dasturiy vositalaridan - yirik axborot bazasi uchun birlamchi massivlarni yaratish va ma' lumotlarni bazaga to`plashdan oldin qayta ishlash bosqichida foydalaniladi. Bazani yaratishdan oldingi qayta ishlash vositasi kompyuterga kiritiladigan axborotning xaqqoniyligini va katta massivdagi ma' lumotlarni yuklashga tayyorgarlikni avtomatlashtirilgan nozoratini ta' minlaydi.

Berilganlarni qayta ishlashning servis vositalari - axborot bazasiga xizmat ko`rsatish bo'yicha ko`makchi vazifalarni ta`minlashi lozim. Ular bazaning dasturiy vositalariga tegishli. Bular ma' lumotlar fayllari va mashina axborot tashuvchi vositalari bilan ishlash bo'yicha turli utilitlardir. Ularga quyidagilar mansub: nusxa olish, arxivlash, tiklash, antivirus vositalari, tarmoq utilitlari va boshqalar.

Foydalanuvchining amaliy dasturlari universal algaritmlash tillaridan birida yaratiladi. Bunday dasturlarda, odatda ularda ishlab chiqiladigan ma' lumotlardan mustaqil bo`lishi ta`minlanmagan. Ayrim joylarda bitta muammoli sohasining turli masalalariga oid axborot massivlarida ma' lumotlar takrorlanadi. Bu xil turli masalalar bo'yicha bir hil ma' lumotlarni bir necha marta kiritishga olib keladi va dastlabki ma' lumotlarga o`zgartirishlar kiritganda ancha muammolarni keltirib chiqaradi. SHuningdek, amaliy dasturlar BBBTda universal algaritmik tilda yarata olishi mumkin.

**Berilganlarning mustaqilligi** - BBBTga qo`yilgan asosiy talab, talablarning kuchli tili esa foydalanuvchining talablarini qondirishning muhim shartidir. Bu tillar assotsiativ manzillashtirish va ma' lumotlar to`plami bilan amallar bajarish vositalariga ega. Bu esa o`z navbatida EHMLardan samarali foydalanishga sharoit yaratadi.

BBBT ikki tartibda: interpretatir va kompilyator tartibda ishlashi mumkin.

Interpretatir tartibda dasturlarning buyruqlari bosqichma-bosqich, birin ketin bajariladi. Unda har bir buyruq nozorat qilinadi, so`ngra mashina tiliga aylantirib, bajariladi. Tegishli amallar bajarilgandan keyin, ular xotiradan o`chiriladi, tizim qayta ishlash bosqichiga o`tadi va keyingi buyruqni bajarishga kirishadi,

interpretatir tartibida «Exe» kengaytirmali fayl hosil qilinmaydi. Bunday faylni hosil qilish uchun kompilyator tartibida foydalaniladi. Kompilyator tartibida buyruqlar bevosita bajarilmaydi, balki ular «exe» faylga yoziladi. Exe faylni hosil qilish jarayoni ikki bosqichdan iborat bo`ladi: boshlang`ich dasturni nozorat qilish va uni ijobiy turga aylantirish; matn muharriri yordamida dasturni exe faylga aylantirish. Exe faylning bajarolishi uchun BBBT ning mavjud bo`lishi shart emas, Interpretatir tartibida ishlaydigan BBBTga dBase III Plus, FixBase va Karat kiradi, kompilyator tartibida Clipper, panel` tartibida esa Clarii ishlaydi. BBBT foydalanuvchi bilan ma`lumotlar bazasi o`rtasidagi aloqani ta`minlovchi dastur sifatida ishtirok etadi. Uning funktsiyalari menyu va dasturlar ko`rinishida namoyon bo`ladi.

Menyu tartibi BBBTning funktsiyalari ekranda tasvirlanadi. Foydalanuvchi kursorni harakatlantirish orqali tegishli funktsiyani aniqlashi va bajarishga chaqirishi lozim. Tizim aniqlangan funktsiyalarni bajarib bo`lgandan so`ng yana menyu holatiga qaytadi.

Dasturiy tartibda tegishli buyruqlar kiritiladi, dasturlar qayta ishlanadi va bajarishga chaqiriladi. Bu holda BBBT interpretator tartibida ishlaydi va foydalanuvchidan dasturlash tillarini bilish talab qilinadi.

BBBT da foydalaniladigan dasturlash tillariga umumiy talablar bilan bir satrda quyidagilar ham qo`yiladi:

- tilning to`liq bo`lishi;
- vazifalarni bajarish uchun tegishli vositalarning bo`lishi;
- aniqlangan ma`lumotlarni to`liq qayta ishlash va boshqalar.

Dasturlash tillari bir satr belgilarga ko`ra turkumlarga ajratiladi.

- ✓ o`zgaruvchanlik;
- ✓ jarayonlilik;
- ✓ foydalanilayotgan matematik aparat va boshqalar

BBBT dagi dasturlar tegishli buyruqlarning to`plamidan tashkil topadi.

Echilayotgan masalalarning qiyinligiga qarab, dasturlar oddiy yoki murakkab tuzilishiga ega bo`ladi. Oddiy tuzilishga ega bo`lgan dasturlarda buyruqlar ketma-ket joylashadi. Murakkab tuzilishli dasturlarda esa buyruqlar modullar holatida, ya`ni asosiy modul va quyi dasturlar to`plamidan iborat bo`ladi. Ma`lumotlar bazasini hosil

qilishda modullik tamoyilidan foydalanish qulay va samaralidir.

Ma`lumotlar bazasini boshqarish tizimi quyidagilarga imkon beruvchi dasturiy vositalarning to`plamidir.

- a) foydalanuvchilarni ma`lumotlarni aniqlash va amallar bajarish tili vositalari bilan ta`minlaydi. Bunday vositalarga ma`lumotlarni aniqlash tili (MAT) va ma`lumotlar bilan amallar bajarish(MABT) kiradi. Ma`lumotlar tili atamasi aytib o`tilganlarining har ikkalasini yoki ulardan birini angla tadi. Ma`lumotlar so`zi ma`lumotlar tilini SI++, Paskal va h.k. kabi tillar turidan farqlaydi. Lekin ma`lumotlar tili universal tilga, masalan S++, Paskalga kirita olishi mumkin. Bunday holatda dasturlashning universal tili va ma`lumotlar tili tegishli ravishda (kirituvchi) til va ma`lumotlarning til isti deb ataladi.

b) foydalanuvchining ma'lumotlar modelini qo'llab-quvvatlashni ta'minlaydi. Ma'lumotlar modeli - ba'zi ilovaga tegishli fizik ma'lumotlarning mantiqiy taqdim eta olishini aniqlash vositasidir.

v) aniqlash, yaratish va ma'lumotlar bilan mantiqiy amallar bajarish (ya'ni tanlash, yangilash, kiritish va yo'q qilish)ga imkon beruvchi MAT va MABT funksiyalarini amalga oshiruvchi dasturlar.

g) ma'lumotlarning himoyasi va yaxlitligini ta'minlaydi. Tizimdan foydalanish faqat shunga huquqi (himoyasi) bo'lgan foydalanuvchigagina ruxsat etiladi. Foydalanuvchilar ma'lumotlar ustida operatsiya bajarayotganlarida saqlanayotgan ma'lumotlarning muvofiqligi (yaxlitligi) ta'minlanadi. Gap shundaki, BBBT ko'plab foydalanuvchilar jam bo'lib ishlash rejimiga asosan tuziladi.

BBBTning tasniflanishini muhim belgilaridan biri bo'lgan ma'lumotlar modeli (tarmoqli, ierarxik yoki relyatsion) turlaridan biri tanlash imkonini beradi. BBBT ma'lumotlar bazalarining ko'p maqsadli tavsifini, ma'lumotlarni himoyalash va qayta tiklashni amalga oshiradi. Rivojlangan muloqat vositalari va yuqori darajali talablar tilining mavjudligi BBBTni oxirgi foydalanuvchi uchun oson vositaga aylantiradi.

To'liq variantda BBBT quyidagi komponentlardan tuza olishi mumkin:

- ✓ Klaviatura orqali ma'lumotlarni to'g'ridan-to'g'ri boshqarish imkonini beruvchi foydalanuvchining muhiti;
- ✓ Interpretator sifatida ish yurituvchi, ma'lumotlarga ishlov berish amaliy tizimini dasturlashning algoritmik tili. Interpretator dasturlarni tez tuzish va maromiga yetkazish imkonini beradi;
- ✓ Mustaqil EXE-fayl shaklidagi tayyor tijorat maxsulotiga tugallangan dastur ko'rinishini beruvchi kompilyator;
- ✓ Ko'p mehnat talab qiluvchi amallarni tez dasturlovchi utilit-dasturlar(hisobotlar, shakllar, jadvallar, darchalar, menyu va boshqa ilovalar generatorlari).

Hususan BBBT—bu foydalanuvchining aslahaviy qobig'i hisoblanadi.

BBBT tarkibida dasturlash tilining mavjudligi aniq masalalarni va aniq foydalanuvchiga mo'ljallangan ma'lumotlarga ishlov berishning murakkab tizimlarini yaratish imkonini beradi.

Berilganlar bazalarini loyihalashtirishda quyidagi tavsiflarni solishtirish va tahlil qilishga asoslangan BBBTni asoslab tanlab olish muhim vazifa hisoblanadi:

- dasturiy texnik bazasi (EHM turi va modeli, hisoblash vositalari konfiguratsiyasiga qo'yiladigan talablar, IT versiyasi);
- ma'lumotlar bazalarining turlari (amaliy, muammoli, lokal, integrellashgan, taqsimlangan);
- foydalanuvchilarning malakalari (BBBT bilan ishlash uchun maxsus tayyorgarlikga ega bo'lmagan foydalanuvchi, mutaxassisligi dasturchi bo'lmagan darajadagi ma'lumotlar bazasi bilan ishlashga tayyorgarligi bo'lgan muammoli sohasining mutaxassisi – foydalanuvchi, amaliy dasturchi, ma'lumotlar bazalarining administratori);
- ma'lumotlar bazalari bilan foydalanuvchilarning muloqat qilish vositalari

(dasturlash tillarini o'z ichiga oluvchi ma'lumotlar ustida ish olib borish va tasvirlash tili);

- ma'lumotlarni qayta ishlash rejimi (paketli, interaktiv, tarmoqli);
- ma'lumotlarni mantiqiy va fizik mustaqilligi;
- ma'lumotlar bazalari axborot strukturalarining asosiy hossalari (mantiqiy Tuza olishi - BBBT vositalari orqali amal qiladigan va tashkiliy tuzilmasini o'zgartirmasdan uni modifikatsiya qilish imkoniyati, ma'lumotlar turini kengaytirgan holda qarshiliksiz ishlov berishda);
- havfsizlik darajasini ta'minlash va ma'lumotlarning to'laqlonligi;
- xizmat ko'rsatishning standart vositalari mavjudligi (ma'lumotlar bazalarini kuzatishni doimiy dasturiy modullarini va ma'lumotlar lug'atini, ma'lumotlar bazalarini yengillashtiradigan (yukini tushiradigan), qayta tashkil etadigan va qayta strukturalashtiradigan, tiklaydigan jurnalni yuritish, kiritish va hisobotlar generatorlari va h.k.);
- ishlatish tavsiflari (loyihachilar haqida, sarmoya egalari haqida, moddiy texnik ta'minotga bo'lgan talablar, tarqatish shakli).

Tanlab olingan BBBT bir satr talablarni qondira olishi kerak. Bularga muammo sohasining barcha vazifalarini samarali bajara olishi; xotira resurslaridan samarali foydalanish uchun saqlanayotgan ma'lumotlar hajmini minimallashtirish;

samarali axborot qarorlarini qabul qilishga imkon yaratish; havfsizlikni ta'minlash jarayonini boshqarish; xodimlarga nisbatan ma'lumotlar bazalarini ishlatish bilan bog'liq qulaylikni yarata olishi, EHM ni ishlatish muammosini soddalashtirish.

Umumiy olganda zamonaviy BBBT lar quyidagi talablarga javob berishi lozim:

- ✓ ma'lumotlarning mustaqilligi;
- ✓ talablarning kuchli tili;
- ✓ javob(sado berish) ning qisqa vaqti;
- ✓ ma'lumotlar va kataliklarni qayta tashkil etishni qisqartirish yoki ulardan voz kechish.

### **Berilganlar bazasini tashkil etishda predmet sohalarning axborot obyektlaridan foydalanish**

Berilganlar bazasining predmet sohasi obyektlari korxonalar, bank sarmoyalari va h.k. hisoblanishi mumkin. Har bir obyekt uchun uning hossalari tavsiflovchi to'plamlar (maydonlar, rekvizitlar) ajratiladi. Xususan bank sarmoyadori – jismoniy shaxs uchun: familiyasi, ismi, sharifi, manzili, pasport ma'lumotlari, ish joyi, sarmoya turi, sarmoya hajmi va h.k. Korxonalar uchun uning nomi, hisob raqami, bank nomi va h.k.

Ma'lumotlar bazasida qanday axborot joylashishi kerak degan qarorni qabul qilishda predmet sohasini yoki qanday masalalarni echishga xizmat qilish ko'lamini aniqlab bo'lmaydi. Balki axborotni turli ko'rinishdagi axborotlar bilan ishlash intensivligi,

ularni dinamik tavsiflari, tuzatish chastatasi, ma'lumotlar orasidagi o'zaro bog'liqlik va o'zaro ta'sirlanishlarga bog'liqligini ko'rsatib beradi.

Amalda ko'pgina foydalanuvchilar ma'lumotlar modellaridan to'liq foydalanmasdan, ularning ba'zi qismlari bilan ish olib boradilar. Masalan: korxonalar hisobchilarini bank mijozlari jismoniy shaxslar qiziqitirmaydi. Shu sababli ko'p xillarda ma'lumotlarni qismlarga ajratish kerak (modelchalar, lokal modellar). Modelchalar deganda aniq bir foydalanuvchi yoki foydalanuvchilar guruhining umumiy qiziqishlari darajasiga aytiladi.

Predmet sohasining axborot obyekti - bu muayyan mohiyat, ya'ni MBda ular haqida axborot bo'lishi lozim bo'lgan real obyekt, hodisa, jarayon yoki voqeaning axborot tarzida aks ettira olishi tushuniladi. Axborot obyekti axborotning tarkibiy birligi sanaladi va me'yorlashtirish talablariga javob berishi lozim.

Axborot obyekti ma'lumotlarning chiziqli strukturasi ega, ya'ni unda guruhlarning ko'plab elementlar yo'q.

A obyektning strukturaviy aloqalari - bu axborot obyektlari juftliklari o'rtasidagi qo'sh (ikkili) aloqalar. Strukturaviy aloqalar turli axborot obyektlarining nusxalari orasidagi real munosabatlar hamda AI orasidagi funksional aloqalar bilan harakterlanadi.

Har bir axborot tizimi o'zining qo'llana olishiga ko'ra, axborot tizimining predmet sohasi deb ataluvchi, atrof-muhitning ma'lum qismi bilan boshlangan bo'ladi.

Predmet sohasining tahlili har qanday axborot tizimini ishlab chiqarishni zaruriy boshlang'ich bosqichi hisoblanadi. Aynan ushbu bosqichda bo'lajak tizim foydalanuvchilarining axborot talabini aniqlaydi va o'z navbatida ma'lumotlar bazasi mohiyatini aniqlab oladi.

Ushbu axborot tizimining predmet sohasi – birinchi navbatda uni foydalanuvchilari qiziqishini yirituvchi real obyektlarni ba'zi to'plamlar sifatida ko'rib chiqiladi. Predmet sohasining obyektlariga Misol tariqasida shaxsiy EXM, dasturiy mahsulotlar, ulardan foydalanuvchilarni kiritish mumkin. Bu obyektlarning har biri o'ziga xos xususiyatlar (atributlar) to'plamiga ega. Masalan: kompyuter ishlab chiqaruvchi forma nomi bilan, model identifikatsiyasi, mikroprosessor turi, tezkor va tashqi xotira hajmi, grafik karta turi va h.k. bilan ta'riflanadi.

Axborot obyekti - predmet sohasi - real obyekt, jarayon, hodisa yoki namoyon bo'lishni qaysidir mohiyatini tasvirlashga aytiladi. Axborot obyekti (mohiyati) obyekt (mohiyatni) sifat va miqdoriy tavsiflarini ko'rsatib beruvchi mantiqiy o'zari bog'langan atributlar (hossalar) to'plami ko'rinishida shakllanadi.

Predmet sohasining obyektlari orasida bog'lanishlar mavjud bo'lib, ular majburiy va fakultativ bog'lanishlardir. Agar yangitdan barpo etilgan obyekt predmet sohasining biror bir obyekti bilan bog'langan bo'lsa, u xolda ular orasida majburiy bog'lanish mavjud bo'ladi. Aks xolda bog'lanish fakultativ (majburiy emas) ko'rinishda bo'ladi.

Kadrlar axborot tizimining predmet sohasini "HODIMLAR" va "MANSAB" obyektlari orasida "MANSABGA TAYINLANISH" majburiy bog'lanishi mavjud. Korxonaga ishga qabul qilinayotgan har bir hodim biror-bir lavozimga tayinlanmay qolmaydi. Shuningdek biror-bir mansabga tayinlanmagan hodimning bo'lishi

mumkin emas. Shu bilan “MANSABGA TAYINLANISH” bogʻlanishi “HODIMLAR” va “MANSAB” obyekt turlari orasida fakultativ hisoblanadi. Predmet sohasining obyektlari majmui, obyektlar atributlari qiymatlari va ular orasidagi bogʻlanishlar vaqt mobaynida oʻzgarishi mumkin. Bu oʻzgarishlar yangi obyektlarni paydo boʻlishi yoki mavjud obyektlarning hisobdan chiqarib tashlanishi natijasida, ular orasidagi yangidan bogʻlanishlarni oʻrnata olishiga yoki mavjudlarini yoʻqolishiga olib keladi. Demak, maʼlum muddatda predmet sohasini qaysidir xolatini taqqoslash imkoniyati mavjud boʻladi.

Axborot obyekti tegishli rekvizit-belgilar hamda rekvizit-asoslar bilan berilgan satr miqdor va sifat harakteristikalari bilan aniqlanadi yoki belgilanadi. Yaʼni axborot obyekti mantiqiy oʻzaro bogʻliq rekvizitlarning maʼlum mohiyatini ifodalovchi majmuidan hosil qilinadi. Tovarlar, materiallar, boʻlinmalar, taʼminotchilar, buyurtmachilar, texnologik operatsiyalar, taʼminot va hokazolar harakteristikalarini aks ettiruvchi rekvizitlar majmui axborot obyektiga Misol boʻlishi mumkin.

AI strukturasi. AI rekvizitlarining tarkibi uning strukturasi belgilaydi. Har bir maʼlum strukturali AI obyekt sinf (tur)ini hosil qiladi. Unga GURUX, PREDMET, OʻQITUVCHI, KAFEDRA singari unikal ism yoki TIV, SKLAD, PIST kabi ramziy belgi bera olishi mumkin.

Rekvizit tarkibiga qoʻyiladigan talablar. Axborot obyekti rekvizitlarining majmui meʼyirlashtirish talablariga javob berishi lozim. Bu uning strukturasi «chiziqini» belgilaydi. Predmet sohasining tavsifi asosida rekvizitlarning funksional bogʻliqligi aniqlanishi va meʼyorlashtirish talablariga javob beradigan AI ajratib ilinishi mumkin.

AI nusxalari va ularni identifikatsiyalash. Maʼlum turdagi axborot obyektlari koʻplab realizatsiya - nusxalariga ega. AIning nusxasi rekvizitlarning aniq qiymati, yaʼni obyektning bir nusxasi haqidagi maʼlumotlar majmui bilan beriladi. AIning har bir nusxasi AI kalitining qiymati bilan birday belgilanishi (identifikatsiya qilinishi) lozim. AI kaliti bir yoki bir necha asosiy rekvizitlardan iborat. AI rekvizitlari kalitga funksional jihatdan bogʻliq boʻlgan asosiy va tavsifiy rekvizitlarga boʻlinadi.

Rekvizitlarning funksional bogʻliqligi. Rekvizitlarning funksional bogʻliqligi faqat kalitning bir qiymatiga tibe (tavsifiy), rekvizitning faqat bir qiymatiga mos boʻlgan holdagina roʻy beradi.

Rekvizitlarning funksional bogʻligining asosiy rekvizitidan tavsifiy (tibe) rekvizitga boruvchi strelkali chizig koʻrnishida tasvirlash mumkin. Asosiy rekvizit ajratib koʻrsatiladi (boʻrttiriladi). Bu aloqalarni bevosita PSning har bir hujjati rekvizitlarining tarkibi berilgan jadvalda aks ettirish qulaydir (1 - jadval).

## «Tovar» AO rekvizitlarining funksional bog`liqligi

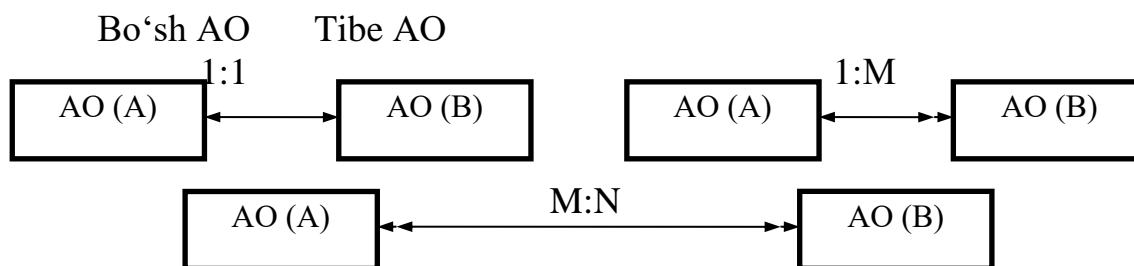
Hujjat	Rekvizit nomi	Rekvizit ismi	Funksional tibeliklar
Tovarlar ma`lumotnomasi	Tovar kodi	KODT	
	Nomi	NAIM	
	Bir dona bahosi	CENA	
	O`lchov birligi	EI	

qayd etib o`tamizki, rekvizitlarning funksional bog`liqligini aniqlashda arifmetik bog`liqlik (masalan, qiymatning miqdorga) hisobga olinmaydi.

Aning grafik tasviri. MAMMni grafik tasvirlashda Aning har bir turi to`g`ri burchak shaklida berilgan. Murakkab MAMMlar uchun axborot obyektining ismini, uning identifikatori (kaliti)ni belgilagan holda faqat AIni tasvirlash hamda 2-rasmda TOVAR va TA`MINOT axborot obyektlari uchun ko`rsatilganidek, ushbu tip obyektlarining ehtimolda tutiluvchi eng ko`p nusxalari sonini ko`rsatish bilan cheklanish maqsadga muvofiqdir.

TOVAR	10
TOV	
KODT	

TA`MINOT	110
POST	
KODT + NDOK	



Munosabatlarning grafik tasviri.

Ierarxik bog`liqlikni ko`rsatib berish uchun MAMM obyektlarini namunali shaklda tasvirlash maqsadga muvofiqdir. Namunali MAMMda axborot obyektlari yuqorida ko`rib chiqilgan me'yorlashtirish talablariga javob berishi lozim. Faqat 1:M va 1:1 tipdagi strukturaviy aloqalarga yo`l qo`yiladi. Bundan tashqari, Aillar MAMMda darajalari bo`yicha tartiblanishlari lozim.

MAMMning grafik tasvirida AIlarni darajalar bo`yicha tartiblash uchun, har bir aloqada bo'sh AI tibe AIga nisbatan yuqori darajada joylashtira olishi kerak. Bunday joylashtiruv har bir AIga beriluvchi daraja indeksiga muvofiq amalga oshiriladi.



AIning indeks darajasini oddiy modellarda aniqlash oson. Buning uchun MAMMning yuqori darajasidan shu AIgacha boʻlgan eng uzoq yoʻldagi aloqalar soni sanab chiqiladi. MAMMning yuqori darajasida «kiruvchi» aloqalarga ega boʻlmagan, yaʼni boshqa birin-bir AIga tibe boʻlmagan (boʻsh AIga ega boʻlmagan) Ailar mavjud boʻladi.

Namunali koʻrinishdagi ishlab chiqilgan MAMM yuqori darajada yagona ildiz obyektga ega boʻlgan toʻntarilgan «daraxt» koʻrinishida qatʼiy ierarxik boʻlishi mumkin. Shixobchali aloqalarga ega boʻlgan namunali MAMMda qatʼiy ierarxik MAMMdan farqli oʻlariq katta darajadagi bittadan ziyid AI bilan bogʻliq boʻlgan Ailar boʻlishi mumkin.

Ierarxik va shixobchali MAMM Relyatsion BBT bilan realizatsiya qilinishi mumkin.

qayd etib oʻtamizki, ierarxik MAMM ierarxik yoki shixobchali(tarmoqli) maʼlumotlar modeliga ega boʻlgan BBT vositalari orqali ham realizatsiya qilinishi mumkin. Shahobchali aloqalarga ega boʻlgan MAMM shahobchali maʼlumotlar modelini taʼminlab turuvchi BBT bilan realizatsiya qilinishi mumkin.

Mashina (tashuvchisi)(da maʼlumotlarni mantiqiy tashkil etishda foydalanayotgan dastur vositalari va mashinachi muhitida maʼlumotlarni qanday yuritishga bogʻliq. Maʼlumotlarni tashkil etishning mantiqiy usuli maʼlumotlar tuza olishining foydalanilayotgan turi hamda dastur vositalari orqali taʼminlanadigan model shakli bilan belgilanadi.

Maʼlumotlar modeli – oʻzari bogʻlangan maʼlumotlar tuza olishi va ular ustida bajariladigan operatsiyalar majmuidir. Model shakli va unda foydalaniladigan maʼlumotlar tuzolishining turi BBT yordamchi model yoki maʼlumotlarni ishlashning amaliy dasturi yaratiladigan dasturlash tuzumi tilida qoʻllanuvchi maʼlumotlarni tashkil etish va ishlash kontseptsiyasini aks ettiradi.

Qayd etish lozimki, aynan bir axborotni mashinaga joylashtirish uchun turli hil maʼlumotlar tuzish, ishlatish mumkin. Ulardan birini tanlash axborotlar bazasini yaratayotgan foydalanuvchi ixtiyorida boʻlib, koʻplab bilimlar, jumladan mavjud texnik va dasturiy taʼminlanganlikka bogʻliq, avtomatlashtirilayotgan vazifaning murakkabligi, axborot hajmi bilan belgilanadi.

## **Maʼlumotlar bazalarida obʼektga yoʻnaltirilgan murajaat va IDBC**

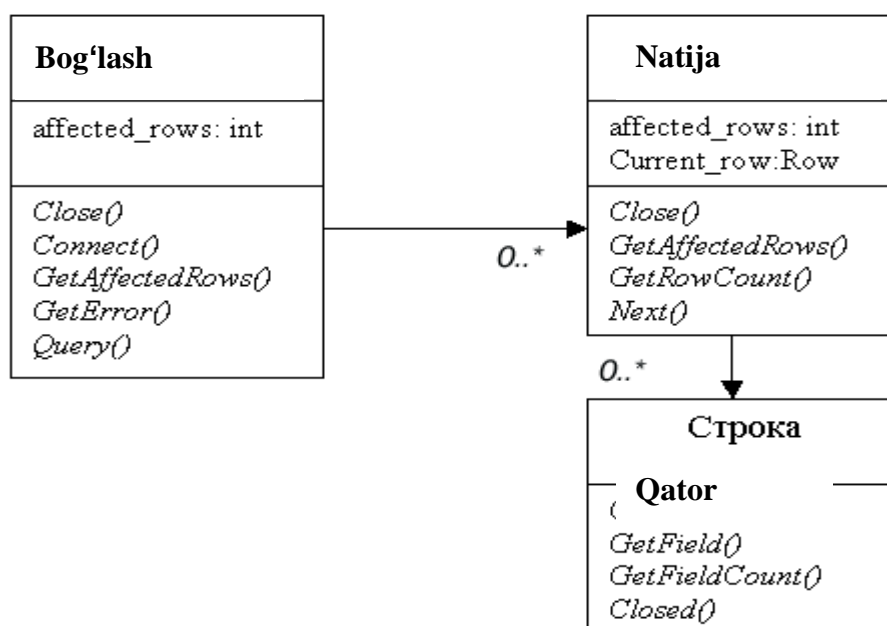
Agar siz MySQL uchun yaratilgan dasturni boshqa BBT ga koʻchirmoqchi boʻlsangiz oʻz kodingizni shu yadro API sidan foydalanadigan qolib qayta yozishingiz kerak.

Lekin dasturchilar boshqa maʼlumotlar bazasiga koʻchirish muammosidan asosan xalos boʻlganlar. Ularda yagona API, Ipen DataBase Cinnectivity API (IDBC), hamma SQL-maʼlumotlar bazalariga unifikatsiya qilingan interfeys mavjud.

IDBC hamma maʼlumotlar bazalariga yagona interfeys boʻlgani uchun, MySQL va boshqa BBT lar bilan ishlovchi dasturlar yaratish uchun uni oʻrganib

chiqish etarli. Agar siz kerakli tarzda IDBC dan foydalansangiz, siz yaratgan dasturlar ixtiyoriy BBBT bilan ishlay oladi.

Hamma APIlar kabi IDBC birgalikda ma'lum funksiyalar to'plamini ta'minlovchi sinflar va interfeyslar to'plamidir. IDBC xolida bu funksiyalar ma'lumotlar bazasiga murojaatni ta'minlaydi. IDBC API ni tashkil qiluvchi sinflar va interfeyslar ixtiyoriy turdagi ma'lumotlar murojrat qolishdagi umumiy tushunchalar abstraksiyasidir. Masalan, CInNECTION ma'lumotlar bazasi Bilan bog'lanishni tasvirlovchi interfeysdir. Shunga o'xshab ResultSet SQL SELECT komandasi qaytaruvchi natijaviy to'plamni tasvirlaydi. Tabiiyki ma'lumotlar bazasiga murojaat konkret detallari uning yaratuvchisiga bog'liq. IDBC bu detallar Bilan ishlamaydi. IDBC sinflarini ma'lumotlar bazasini dasturlashga ob'ektga-yo'naltirilgan usullar nuqtai nazaridan ko'rib chiqamiz.



### Ma'lumotlar bazasiga ob'ektga yo'naltirilgan murojaat bibliotekasi

BBBT Bilan ishlashni uchta asosiy tushuncha tasvirlaydi: ulanish, natijaviy to'plam va natijaviy to'plam satrlari. 2-5 rasm bu ob'ektlarni UML-diagrammada ko'rsatadi.

UML - bu Yangi Unifikatsiyalangan modellashtirish tili bo'lib , созданный Gradi Buch, Ayvar Yakobson va Djeyms Rambo (Grady Biich, Ivar Jacobsin, James Rumbaugh) tomonidan ob'ektga-yo'naltirilgan loyihalash va tahlilni hujjatlash Yangi standarti sifatida taklif qilingan.

Ixtiyoriy muxitda ma'lumotlar bazasiga murojaat ulanishdan boshlanadi. Bizning ob'ektga-yo'naltirilgan bibliotekamizni yaratish CInNECTION ob'ektini yaratishdan boshlanadi. Ob'ekt CInNECTION server Bilan bog'lanishni o'rnatish,

zarur ma'lumotlar bazasini tanlash, so'rovlarni uzatish va natija olishni bolishi kerak.

CinnectIOn sinfi usullari hamma BBBT lar uchun bir xildir. Lekin sinf ichida, kompilyatsiya qilinayotgan biblioteka uchun xos bo'lgan yopiq a'zolar berkitilgandir. Bog'lanish o'rnatishda ma'lumotlar Bilan bog'lanishni ta'minlaydigan sinf a'zolari farqli bo'lib qoladi.

Bu API yordamida yaratiladigan hamma amaliy dasturlarga ma'lumotlar bazasiga ulanish uchun CinnectIOn sinfi nusxasini uning konstruktorlaridan biri yordamida yaratish kerak bo'ladi. U kabi uzaolish uchun Amaliy dastur CinnectIOn nusxasini o'chirishi kerak. U to'g'ridan to'g'ri Clise() va Sonnect() usullariga murojaat qolib CinnectIOn nusxasini qaytadan ishlatishi mumkin.

CinnectIOn yana bir mantiqiy funksiyasi ma'lumotlar bazasi bilan aloqani uzish va dasturdan berkitilgan resurslarni ozod qilishdir. Bu funktsiyani Clise () usuli amalga oshiradi.

Bog'lanishni ochish va yopish odatda ma'lumotlar bazasiga komandalar yuboriladi. CinnectIOn sinfi argument sifatida SQL komanda oluvchi Query() usuli yordamida bajaradi. Agar komanda so'rov bo'lsa 2-5. rasmda ko'rsatilgan ob'ekt modelidan Result sinfi nusxasini qaytaradi. Agar komanda ma'lumotlarni Yangilayitgan bo'lsa, usul NULL qaytaradi va affected\_riws qiymatini o'zgartirilgan satrlar soniga teng qiladi.

Result sinfi natijaviy to'plam ma'lumotlariga hamda shu natijaviy to'plam bilan bog'liq meta ma'lumotlarga murojaatni ta'minlashi kerak. 2-5 rasmda ko'rsatilgan ob'ektli modelga asosan bizning Result sinfimiz natijaviy to'plam satrlarini sikl bo'yicha o'qish va undagi satrlar sonini aniqlashni ta'minlaydi.

Bizning Result sinfimiz natijaviy to'plam Bilan qatorma qator ishlaydi. Result sinfi nusxasini Query() usuli yordamida olgandan so'ng amaliy dastur to navbatdagi Next() usuli 0 qaytarmaguncha, ketma ket Next() va GetCurrentRiw() usullarini chaqirishi lozim.

Natijaviy to'plamning alohida satri bizning ob'ektli modelimizda Riw sinfi bilan tasvirlanadi.

Ma'lumotlarga murojaat massiv indeksi bo'yicha so'rov tomonidan berilgan tartibda amalga oshiriladi. Masalan, SELECT user\_id , password FROM users so'rov uchun indeks 0 foydalanuvchi nomini va indeks 1 – parolni ko'rsatadi. Bizning C++ API bu indeklashni foydalanuvchi uchun do'stona qiladi. GetField(1) yoki fields[0] birinchi maydonni qaytaradi.

## *PHP dasturlari*

PHP dasturlari ikki usulda bajara olishi mumkin: Web-server tomonidan stsensariy ilovasi va konsol dasturi sifatida.

Bizning maqsadimiz web ilovalarni dasturlash bo`lgani uchun asosan birinchi usulni ko`ramiz.

PHP odatda Internet bilan bog`liq dasturlar yaratish uchun ishlatiladi. Lekin PHP dan komanda satrlar interpretatori, asosan unix tizimlarda foydalanish mumkin. Oxirgisi CIRBA va CIM interfeyslar hamda PHP-GTK kengaytmasi yordamida mumkin. Bu xolda quyidagi masalalarni echish mumkin:

- Interaktiv komanda qatorlari yordamida ilovalar yaratish;
- Kross-platformali GUI ilovalarni PHP-GTK bibliotekasi yordamida yaratish;
- Windows va Linux uchun ba'zi masalalarni avtomatizatsiya qilish

Serverga brouzerning murojlat qilishida yordamida php-stsenariylari bajarolishini ko`rib chiqamiz. Avval borouzer .php kengaytmali sahifani so`raydi, so`ngra web-server dasturni PHP mashinadan o`tkazadi va natijani html-kod shaklida qaytaradi. Agar standart HTML sahifani olib, kengaytmasoni .php ga o`zgartirilsa va PHP mashinadan o`tkazilsa, foydalanuvchiga o`zgartirmasdan qaytaradi. Buf faylga PHP komandani qo`shish uchun, PHP komandalani maxsus teglar ichiga olish kerak. Bu teglarning 4 xil shakli mavjud bo`lib , Ixtiyoriy sidan foydalanish mumkin:

1. **XML qayta ishlash instruktsiyasi:**
2. `<?php`
3. ...
4. `?>`
5. **SGML qayta ishlash instruktsiyasi:**
6. `<?`
7. ...
8. `?>`
9. **HTML stsenariylari qayta ishlash instruktsiyasi:**
10. `<script language = "php">`
11. ...
12. `</script>`
13. **ASP uslubidagi instruktsiya:**
14. `<%`
15. ...

Biz XML yoki SGML uslubiga rioya qilamiz.

Xususan biror blok ichida PHPdan chiqish mumkin, faqat keyinchalik yana uning ichiga kirib kodni tugatish sharti bilan, quyidagi konstruktsiya mumkin:

```
<?
  if(5<3){
    echi("<p>Helli, world!<p>");
  }
?>
<p>Helli!</p>
// bu qator PHP kodi sifatida qaralmaydi
// va kod bloki bajarilayotgan bo`lsa chiqariladi
<?
  echi("<p>Helli, world!<p>");
}
?>
```

PHP da echi komandasi web – sahifalarda uchraydigan har qanday ma'lumotni( matn, HTML ajratuvchi simvoli, son) chiqarish uchun qo'llanadi. Uning ma'nosi misolda ko`rsatilgan.

PHP tilida izohlarni joylash uchun bir necha usullar mavjud. Eng soddasi ikkilik slash (//) dan foydalanish, shundan so`ng PHP satrlar oxirigacha yozilganni o`tkazib yuboradi. Bundan tashqari S (/\*...\*/) uslubidagi ko`p qatorli izohlardan foydalanish mumkin. Bir qatorli izohlar uchun (#) simvoldan foydalanish qulay.( UNIX script tillaridagi izoh).

```
<php
  echi("<p>Helli</p>"); // izoh
  echi("<p>Helli</p>"); # izoh
  /*
    bu ham izoh
  */
?>
```

Shuni esdan chiqarmaslik lozimki PHP uslubi izohlari faqat PHP chegaranishlari orasida ta'sir qiladi. Agar PHP bu izohlar simvollarini chegaranishlari tashqarisida uchratsa, ularni boshqa matnga o`xshab, html- sahifaga joylashtiradi.

Masalan:

```
<php
```

```
echi("<p>Helli</p>"); // normal izoh
?>
```

// bu izoh brouzerda ko`rinadi.

```
<!-- HTML izohi.
```

Bu izoh HTML kodda ko`rinadi, brouzerda emas -->

Izohlarni faqat operator oxiriga emas, quyidagicha joylash ham mumkin:

```
<?
```

```
  $a = "Helli, world";
```

```
  echistrstr($a, "H");
```

```
  // bu funktsiyani keyinchalik qarab chiqamiz
```

```
?>
```

RNR da o'zgaruvchilar dollar (\$) belgisidan boshlanadi. Bu simvoldan Ixtiyoriy sondagi harf, raqam va ostiga chiziq simvollar kelishi mumkin, lekin birinchi simvol albatta harf bo'lishi kerak. Shuni esda tutish kerakki, RNRda o'zgaruvchilarning nomlari kalit so'zlardan farqli registrga bog'liqdir.

RNR da o'zgaruvchilarni ta'riflaganda oshkora tipini ko'rsatish shart emas va dastur davomida bitta o'zgaruvchi har xil tiplarga ega bo'lishi mumkin.

O'zgaruvchi unga qiymat berilganda initsializatsiya qilinadi va dastur bajarilguncha mavjud bo'ladi. Ya'ni web-sahifa xolida to so'rov tugamaguncha.

Mijoz so'rovi web-server tomonidan tahlil qilinib, RNR mashinaga uzatilgandan so'ng, u so'rovga tegishli ma'lumotlarni o'z ichiga olgan va bajarish davomida murojaat qilish mumkin bo'lgan bir necha o'zgaruvchilarni yaratadi. Oldin RNR sizni tizimingiz atrof muhit o'zgaruvchilarini oladi vash u nomdagi va shu qiymatdagi RNR stsenariysi atrofidagi o'zgaruvchilarni yaratadi, toki servedagi stsenariylarga Mijoz tizimi xususiyatlari bilan ishlash mumkin bo'lsin. Bu o'zgaruvchilar **\$HTTP\_ENV\_VARS** assotsiativ massivga joylashtiriladi.

Tabiiyki **\$HTTP\_ENV\_VARS** massivi o'zgaruvchilari tizimga bog'liqdir (chunki ular aslida atrof muhit o'zgaruvchilaridir). Atrof muhit o'zgaruvchilari qiymatlarini sizni mashinangiz uchun env (Unix) yoki set (Windows) komandasi yordamida ko'rishingiz mumkin.

So'ngra RNR u GET-o'zgaruvchilarning guruhini yaratadi. Ular so'rov satrini tahlil qolishda yaratiladi. So'rov satri **\$QUERY\_STRING** o'zgaruvchida saqlanadi va so'ralgan URL dagi "?" simvoldan keyingi informatsiyadan iborat. RNR so'rov satrini & simvollar bo'yicha aloxida elementlarga ajratadi, va har bir elementda "=" belgisini qidiradi. Agar "=" belgisi topilgan bo'lsa, tenglik chap tomonidagi simvoldan iborat o'zgaruvchi yaratadi. Quyidagi formani ko'ramiz:

```
<form action = "http://lcalhist/PHP/test.php" method="get">
```

```
  HDD: <input type="text" name="HDD"/><br>
```

```
CDRIM: <input type="text" name="CDRIM"/><br>
<input type="submit"/>
```

Agar siz bu formada HDD qatorda "Maxtir", CDRIM qatorda "Nec" tersangiz, quyidagi so`rov shaklini xosil qiladi:

```
http://licalhist/PHP/test.php?HDD=Maxtir&CDRIM=Nec
```

Bizning misolimizda RNR quyidagi o'zgaruvchilarni yaratadi: **\$HDD** = "Maxtir" va **\$CDRIM** = "Nec".

Siz o'zingizni scriptingdagi (bizda – test.php) bu o'zgaruvchilar Bilan oddiy o'zgaruvchilar bilan ishlagandek ishlashingiz mumkin. Bizning misolimizda ular ekranga chiqariladi:

```
<?
  echi("<p>HDD is $HDD</p>");
  echi("<p>CDRIM is $CDRIM</p>");
?>
```

Agar sahifa so`rovi PIST usuli yordamida bajarilsa, PIST-o'zgaruvchilarning guruhi yaratilib, interpretatsiya qilinadi va **\$HTTP\_PIST\_VARS** massivga joylashtiriladi.

Konstantalar RNR da **define()** funksiyasi yordamida e'lon qilinadi:

```
define(CONSTANT, value)
```

Bu funksiya birinchi parametri – konstant nomi, ikkinchisi – uning qiymati. Konstantadan foydalanilganda nomi bo`yicha ilova qilinadi:

```
<?
  define(CONSTANT1,15);
  define(CONSTANT2,"\x20"); // kod probela
  define(CONSTANT3,"Helli");
  echi(CONSTANT1);
  echi(CONSTANT2);
  echi(CONSTANT3);
?>
```

Odatga ko`ra konstantalar nomlari yuqori registr harflari bilan yoziladi. Bu faqat odat bo`lsa ham unga rioya qolishni maslaxat beramiz, chunki yaxshi odatlarga rioya qilmaydigan dasturchilardan yimon dasturchilar chiqadi. Konstantalar aniqlanganligini **defined()** funksiyasi yordamida tekshirish mumkin:

```
<?
  define(CONSTANT,"Helli");
```

```

if(defined("CONSTANT"))
{
    echi("<p>CONSTANT is defined</p>");
}
?>

```

Yuqorida aytilganidek PHP tilida bitta o'zgaruvchini dastur bajara olish davomida satr yoki son sifatida ishlatish mumkin. Shu bilan birga PHP tilida o'zgaruvchilar bilan ishlanganda oshkor ko`rsata olishi mumkin bo`lgan asosiy ma'lumotlar tiplari to`plami mavjud:

- **integer;**
- **string;**
- **biilean;**
- **diuble;**
- **array;**
- **ibject;**

RNR o'zgaruvchiga tayinlagan tipni qaytaruvchi **gettype()** funktsiyasi mavjud:

```

<?
    $var = "5";
    $var1 = 5;
    echi(gettype($var));
    echi "<br>";
    echi(gettype($var1));
?>

```

Birinchi xolda RNR **string** qaytaradi, ikkinchi xolda **integer**.

Tipni oshkora o`rnatuvchi **settype()** funktsiyasi mavjuddir:

```

<?
    $var = "5";
    echi(gettype($var));
    settype($var,integer);
    echi "<br>";
    echi(gettype($var));
?>

```

Kodning bu fragmentini bajarish, avvalgisini bajarish Bilan bir xil natijaga olib keladi.



RNR tilida **settype()** funksiyasidan tashqari tipni o'zgartirish operatoridan foydalanish mumkin. Tipni o'zgartirish qavslarga olingan Yangi tipni ko'rsatish orqali bajariladi:

```
$var = (int)$var;
```

RNR quyidagi kodni bajarish natijasida, integer qaytaradi:

```
<?
```

```
$var = "5"; // string tip
```

```
$var = (int)$var; // int ga o'zgartiramiz
```

```
echi(gettype($var));
```

```
?>
```

### *Operatorlar*

#### *Arifmetik operatorlar*

+	Qo`shish
-	Ayirish
*	Ko`paytirish
/	Bo`lish
%	Modul bo`yicha qoldiqni hisoblash (masalan: 5 % 2=1)

#### *Razryadli operatorlar*

Bu operatorlar guruhi butun qiymatli operatorlarning bitli tasvirlari bilan ishlaydi. Asosan bu operatorlar mantiqiy bayroqlar to`plamini yaratish uchun ishlatiladi. Bizning misollarimizda bu operatorlarni kam ishlatamiz, lekin quyida shu operatorlardan foydalanishga misol keltiramiz.

&	Razryadli kon'yunktsiya (AND)
	Razryadli diz'yunktsiya (IR)
^	Razryadli ajratuvchi YOKI (XIR)
~	Razryadli inkor (NIT)
<<	Chap butun qiymatli operand bitli tasvirini o`ng butun qiymatli operand qiymatiga teng razryadlar soniga chapga siljitish

## **2.BOB. BERILGANLAR BAZALARINI TASHKIL ETISHDA DASTURLASHTIRISH TILLARIDAN FOYDALANISH**

### **2.1. SQL tili va uning tarkibi.**

SQL (Structured Query Language) tili strukturalashgan so`rov tili deyilib, u ma'lumotlar bazasi bilan a'loqa o`rnatish uchun xizmat qiladi. SQL tili 70 yillar oxirida IBM firmasi tomonidan System r nomli ma'lumotlar bazasini boshqarish tizimining tajribaviy loyahasini ishlab chiqish diorasida yaratildi. Keyinchalik IBM firmasi tomonidan System r ga yaqin bo'lgan yana ikkita tizim – SQL/DS va DB2 tizimlari ishlab chiqildi. Bu tilning xalqaro standarti 1986 yili ishlab chiqildi va u 1989 yilga kelib yanada kengaytirildi, lekin uning to'liq xalqari standarti 1992 yil qabul qilindi. 1995 yilga kelib SQL92 standarti yangi komponentalar bilan to'ldirildi. Birinchi bolib SQL dan foydalanish Oracle BBTda ishlatildi.

SQL quyidagi kimpinentalarni o'z ichiga iladi:

- administrativ ma'lumotlar vositasi;
- tranzaktsiyalar bilan boshqarish vositasi;
- DDL (Data DefiniTION Language) ma'lumotlarni aniqlash operatorlari;
- DML (Data ManipulaTION Language) ma'lumotlar ustida manipulyatsiya qilish (murakkab harakatlar bajarish) operatorlari;
- DQL (Data Query Language) ma'lumotlar bazasiga so`rovlar tili.

Adminstratov ma'lumotlar - ma'lumotlar bazasini yaratish, unga kirishga ruxsat berish va parollarni o'zgartirishda kerak bo`ladi.

Tranzaktsiya - bu ma'lumotlar bilan monopulyatsiya qolib ketma-ket operatsiyalar yordamida BBTga ta'sir etishdir. Tranzaktsiya yoki to'liq bajariladi va ma'lumotlar bazasini bir butun holatdan ikkinchi bir butun holatga o`tkazadi, yoki agar ma'lumotlar bazasiga ko`ra tranzaktsiyaning biror holati bajarilmaydigan bo'lsa yoki sistemada biror hatolik yuz bersa, ma'lumotlar bazasi boshlang'ich holatiga qaytadi.

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

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

### **Jadvallarni yaratish**

Ma'lumotlar bazasining jadvallari massiv ko`rinishida bo'lib , ular satr va ustunlardan iborat bo`ladi. Jadval tuzilgandan keyin uni to'ldirishga kirishish mumkin.

Jadvallar CREATE TABLE komandasi bilan yaratiladi. Bu komanda qatorlarsiz bo`sh jadval yaratadi. CREATE TABLE komandasi jadval nomini va jadval o`zini ma'lum tartibda ko`rsatilgan ustunlar nomlari ketma – ketligi ta'rifi

ko`rinishida aniqlaydi. U ma'lumotlar tiplari va ustunlar o`lchovini aniqlaydi. Har bir jadval juda bo`lmaganda bitta ustunga ega bo`lishi kerak.

**CREATE TABLE** komandasi sintaksisi:

```
CREATE TABLE <table-name >  
( <cilumn name> <data type>[(<size>)],  
<cilumn name> <data type>[(<size>)], ... );
```

Argument qiymati kattaligi ma'lumot turiga bog`liqdir. Agar siz maxsus ko`rsatmasangiz, tizim avtomatik qiymatni o`rnatadi.

Misol uchun sotuvchilar jadvalini yaratishni ko`rib chiqamiz:

```
CREATE TABLE Salepeople  
( SNum integer,  
SName char (10),  
City char (10),  
Cimm decimal );
```

*Jadvallarni o`chirish.*

Jadvalni o`chirish imkoniga ega bo`lish uchun, jadval egasi (Ya'ni yaratuvchisi) bo`lishingiz kerak. Faqat bo`sh jadvalni o`chirish mumkin. Qatorlarga ega bo`lgan, to`ldirilgan jadvalni o`chirish mumkin emas, Ya'ni jadval o`chirishdan oldin tozalangan bo`lishi kerak. Jadvalni o`chirish komandasi quyidagi ko`rinishga ega:

```
DRIP TABLE < table name >;
```

Masalan: **DRIP TABLE Salepeople;**

*Jadvalni yaratilgandan so`ng o`zgartirish.*

Jadvalni o`zgartirish uchun ALTER TABLE komandasidan foydalaniladi. Bu komanda jadvalga Yangi ustunlar qo`shish, ustunlarni o`chirish, ustunlar kattaligini o`zgartirish, hamda cheklanishlarni qo`shish va olib tashlash imkoniyatlariga ega. Bu komanda ANSI standarti qismi emas, shuning uchun har xil tizimlarda har xil imkoniyatlarga ega.

Jadvalga ustun qo`shish uchun komandaning tipik sintaksisi:

```
ALTER TABLE <table name> ADD <cilumn name>  
<data type> <size>;
```

**Masalan:**

**ALTER TABLE Salepeople ADD Phine CHAR(7);**

## **Jadvallar uchun cheklanishlar**

### ***Cheklanishlarni kiritish.***

Ko`p xollarda ustunga kiritilgan qiymatlar bir biridan farq qilishi kerak. Agar ustun uchun UNIQUE cheklanishi o`rnatilsa, bu ustunga mavjud qiymatni kiritishga urinish rad etiladi. Bu cheklanish bo`sh bo`lmaydigan (NIT NULL) deb e`lon qilingan maydonlarga qo`llanishi mumkin.

Masalan:

**CREATE TABLE Salepeople**

**( SNum integer NIT NULL UNIQUE,  
SName char (10),  
City char (10),  
Cimm decimal);**

Unikalligi talab qilinadigan maydonlar(birlamchi kalitlardan tashqari) nomzod kalitlar yoki unikal kalitlar deyiladi.

Jadval cheklanishi UNIQUE maydonlar guruhiga o`rnata olishi mumkin. Bu bir necha maydonlar qiymatlari kombinatsiyasi unikalligini ta`minlaydi. Bizning ma`lumotlar bazamizda har bir buyurtmachi bitta sotuvchiga birlashtirilgan. Ya`ni buyurtmachilar jadvalida buyurtmachi nomeri (cnum) va sotuvchi nomeri (snum) kombinatsiyasi unikal bo`lishi kerak. Bu cheklanishni UNIQUE (cnum, snum) yordamida, Custimers jadvalini yaratishda kiritish mumkin. Bu ustunlar uchun NIT NULL cheklanishini kiritish zarurdir.

### ***Birlamchi kalitlar cheklanishlari.***

SQL birlamchi kalitlarni to`g`ridan to`g`ri birlamchi kalit (PRIMARY KEY) cheklanishi orqali ta`riflaydi. PRIMARY KEY jadvalni yoki ustunlarni cheklashi mumkin. Bu cheklanish UNIQUE cheklanishi kabi ishlaydi, jadval uchun faqat bitta birlamchi kalit (Ixtiyoriy sondagi ustunlar uchun ) aniqlanishi mumkin bo`lgan xoldan tashqari. Birlamchi kalitlar NULL qiymatga ega bo`lishi mumkin emas.

Misol:

**CREATE TABLE Salepeople**

**( SNum integer NIT NULL PRIMARY KEY,  
SName char (10),  
City char (10),  
Cimm decimal);**

***Maydon qiymatlarini tekshirish (CHECK cheklanishi).***

CHECK cheklanishi jadvalga kiritilayotgan ma'lumot qabul qilinishidan oldin mos kelishi lozim bo'lgan shart kiritishga imkon beradi. CHECK cheklanishi CHECK kalit so'zi ko'rsatilgan maydondan foydalanuvchi predikat ifodalaridan iboratdir.

Misol: Salepeople jadvali Cimm ustuniga kiritilayotgan qiymat 1 dan kichik bo'lish sharti.

```
CREATE TABLE Salepeople  
( SNum integer NOT NULL PRIMARY KEY,  
SName char(10) NOT NULL UNIQUE,  
City char(10),  
Cimm decimal CHECK ( Cimm < 1 ));
```

### **Ko'zda tutilgan qiymatlarni o'rnatish.**

Biror bir maydon uchun qiymat ko'rsatmagan holda jadvalga satr qo'shsangiz, SQL bunday maydonga kiritish uchun ko'zda tutilgan qiymatga ega bo'lishi kerak, aks holda komanda rad etiladi. Eng umumiy ko'zda tutilgan qiymat NULL qiymatdir. CREATE TABLE komandasida ko'zda tutilgan qiymat DEFAULT operatori orqali, ustun cheklanishi sifatida ko'rsatiladi. Masalan:

```
CREATE TABLE Salepeople  
( SNum integer NOT NULL PRIMARY KEY,  
SName char(10) NOT NULL UNIQUE,  
City char(10) DEFAULT 'New York',  
Cimm decimal CHECK ( Cimm < 1 ));
```

### ***Ma'lumotlar yaxlitligini ta'minlash***

Jadval bir maydonidagi hamma qiymatlar boshqa jadval maydonida aks etsa, birinchi maydon ikkinchisiga ilova qiladi deyiladi. Bu ikki maydon orasidagi bog'liqlikni ko'rsatadi. Masalan, buyurtmachilar jadvalida har bir buyurtmachi, sotuvchilar jadvalida o'ziga biriktirilgan sotuvchiga ilova qiluvchi SNum maydoniga ega. Bir maydon ikkinchisiga ilova qilsa tashqi kalit, u ilova qilayotgan maydon ajdod kalit deyiladi. Buyurtmachilar jadvalidagi SNum maydoni tashqi kalit, sotuvchilar jadvalidagi SNum - ajdod kalitdir.

Tashqi kalit bitta maydondan iborat bo'lishi shart emas. Birlamchi kalit kabi, tashqi kalit bitta modul sifatida qayta ishlanuvchi bir necha maydonlarga ega bo'lishi mumkin. Maydon tashqi kalit bo'lsa ilova qitlayitgan jadval bilan ma'lum usulda bog'liqdir. Tashqi kalit har bir qiymati (satri), ajdod kalitning bitta va faqat bitta qiymatiga( satri) ilova qilishi kerak. Bu xolda tizim ilovali yaxlit xolatda deyiladi

Shu bilan birga ajdod kalit qiymati tashqi kalit bir necha qiymatlariga ilova qilishi mumkin. ***Cheklanish FOREIGN KEY.***

SQL ilovali yaxlitlikni FOREIGN KEY yordamida ta'minlaydi. Tashqi kalit vazifasi ajdod kalitda ko'rsatilmagan qiymatlarni tashqi kalit maydonlariga kiritmaslikdir. FOREIGN KEY cheklanish sintaksisi:

**FOREIGN KEY <column list> REFERENCES  
<pktable> [<column list>]**

Birinchi ro'yxat buyruq tomonidan o'zgartiriluvchi ustunlar ro'yxatidir. Pktable - bu ajdod kalitli jadval. Ikkinchi ustunlar ro'yxati bu ajdod kalitni tashkil qiluvchi ustunlardir.

Misol uchun Sotuvchilar jadvaliga ilova qiluvchi tashqi kalit sifatida e'lon qilingan SNum maydoniga ega bo'lgan Buyurtmachilar jadvalini yaratamiz:

**CREATE TABLE Custimers  
( CNum integer NOT NULL PRIMARY KEY,  
CName char(10),  
City char(10),  
SNum integer,  
FOREIGN KEY (SNum) REFERENCES Salepeople (SNum) );**

### **Maydonlar qiymatlarini kiritish, o'chirish va o'zgartirish**

Hamma satrlar SQLda INSERT komandasi yordamida kiritiladi. INSERT quyidagi formatlardan biriga ega bo'lishi mumkin:

**INSERT INTO <table name | view name> [(column [,column] ...)]  
VALUES ( <value> [,<value>] ... );**

yoki

**INSERT INTO <table name | view name> [(column [,column] ...)]**

Ostki so'rov;

Masalan, sotuvchilar jadvaliga satr kiritish uchun quyidagi shartdan foydalanishingiz mumkin:

**INSERT INTO Salepeople  
VALUES (11, 'Peel', 'Toshkent', .12);**

Siz nom kiritish uchun ustunlar ko'rsatishingiz mumkin. Bu nomlarni ixtiyoriy tartibda kiritishga imkon beradi. Masalan:

```
INSERT INTO Salepeople (Sname, Cimm, SNum)
VALUES ('Peel', .12, 11);
```

E'tibor bering City ustuni tashlab yuborilgan, chunki unga ko'zda tutilgan qiymat kiritiladi.

### *Satrlarni o`chirish.*

Satrlarni jadvaldan DELETE komandasi bilan o`chirish mumkin. U aloxida qiymatlarni emas faqat satrlarni o`chiradi. DELETE quyidagi formatga ega:

```
DELETE FROM <table name | view name>
[WHERE search-cindition];
```

Masalan, Sotuvchilar jadvalidagi hamma satrlarni o`chirish uchun, quyidagi shartni kiritish mumkin:

```
DELETE FROM Salepeople;
```

Ma'lum satrlarni o`chirish uchun predikatdan foydalaniladi. Masalan, jadvaldan Axelrid sotuvchini o`chirish uchun:

```
DELETE FROM Salepeople
WHERE SNum = 13;
```

### **Maydon qiymatlarini o'zgartirish.**

Bu o'zgartirish UPDATE komandasi yordamida bajariladi. Bu komandada UPDATE ifodasidan so'ng jadval nomi va SET ifodasidan so'ng ma'lum ustun uchun o'zgartirish ko'rsatiladi. UPDATE ikki formatga ega. Ulardan birinchisi:

```
UPDATE <table name | view name>
SET cilumn = expression [, cilumn = expression] ...
[WHERE search-cindition]
```

bu erda expression - bu ustun | ifoda | konstanta | o'zgaruvchi.

Masalan, hamma buyurtmachilar baxosoni 200 ga o'zgartirish uchun quyidagini kiritishingiz mumkin:

```
UPDATE Custimers
SET Rating = 200;
```

Ma'lum satrlarni o'zgartirish uchun DELETE dagi kabi predikatdan foydalanish kerak. Masalan Peel (SNum=11) sotuvchining hamma buyurtmachilari uchun bir xil o'zgartirishni quyidagicha kiritish mumkin:

```
UPDATE Custimers  
SET Rating = 200  
WHERE SNum = 11;
```

### **SELECT so`rov operatori**

SELECT operatori BB jadvallaridan natijaviy to`plam olish uchun mo`ljallangan ifodadir. Biz SELECT operatori yordamida so`rov beramiz, u bo`lsa ma'lumotlarning natijaviy to`plamini qaytaradi. Bu ma'lumotlar jadval shaklida qaytariladi. Bu jadval keyingi SELECT operatori tomonidan qayta ishlanishi mumkin va xokazo.

Operator SQL92 standartiga ko`ra quyidagi ko`rinishga ega:

```
SELECT -- ALL ----- sxema , ustun ----  
    -- DISTINCT -- ---- * -----  
FROM -- sxema , Jadval .. -----  
WHERE -- izlash sharti -----  
GROUP BY -- sxema , ustun -----  
HAVING -- izlash sharti -----  
ORDER BY – tartiblash spetsifikatori -----
```

OFFICES jadvalidagi hamma yozuvlarni qaytaruvchi sodda so`rov ko`ramiz.

```
SELECT * FROM OFFICES
```

### ***SELECT yordamida ma'lumotlarni tanlash***

SELECT operatori albatta "qaytariluvchi ustunlar ro`yxati " ni o`zichiga olishi kerak.

Misol: Hamma xizmatchilarning nomlari, ofislari va ishga olish sanalari ro`yxatini xosil qilish.

```
SELECT NAME, REP_OFFICE, HIRE_DATE FROM SALESREPS
```

Ba`zida ma'lumotlarni tanlashda qaytariluvchi qiymatlar xosil bo`ladi.

Bu xol yuz bermasligi uchun DISTINCT operatoridan foydalanish lozim. Masalan, quyidagicha:

```
SELECT DISTINCT MGR FROM OFFICES
```

### ***SELECT operatori WHERE sharti***

Endi WHERE ifodasidan foydalanib ba`zi so`rovlarni ko`rib chiqamiz: Sotuvlar haqiqiy hajmi rejadan oshgan ofislarni ko`rsating.



**SELECT CITY, SALES, TARGET FROM OFFICES WHERE SALES > TARGET**

Identifikatori 105 ga teng bo`lgan xizmatchi nomi haqiqiy va rejadagi sotuvlar hajmini ko`rsating:

**SELECT SALES, NAME, QUITA FROM SALESREPS WHERE EMPL\_NUM = 105**

Agar izlash sharti TRUE, bo`lsa qator natijaviy to`plamga qo`shiladi, agar izlash sharti FALSE bo`lsa, qator natijaviy to`plamga qo`shilmaydi, agar NULL bo`lsa ham natijaviy to`plamdan chiqariladi! O`z ma`nosiga ko`ra WHERE, keraksiz yozuvlarni chiqarib, kerakligini qoldiruvchi filtr sifatida ishlatiladi!

### **Mantiqiy operatorlar**

#### ***BETWEEN va ON Operatorlari***

BETWEEN ifodasi bu qiymatlar diapazoniga tegishlilikni tekshirishdir. Ifoda sintaksisi quyidagicha:

NOT ifodasi shartni teskarisiga o`giradi, ya`ni tegishli emas ma`noni bildiradi.

Misol: Narxi har xil diapazonga mos keluvchi buyurtmalarni topish.

**SELECT ORDER\_NUM, AMOUNT  
FROM ORDERS  
WHERE AMOUNT BETWEEN 20.000 AND 29.999**

NOT ifodasi yordamida berilgan diapazonga tegishlilikni tekshirish mumkin, masalan: Sotuvlar haqiqiy hajmlari rejaning 80 dan 120 prosentigacha bo`lgan diapazonga tushmaydigan xizmatchilar ro`yxatini chiqarish.

**SELECT NAME, SALES, QUITA  
FROM SALESREPS  
WHERE SALES NOT BETWEEN (0.8 \* QUITA) AND (1.2 \* QUITA)**

Ifoda ON to`plamga tegishlilikni tekshiradi.

To`rtta aniq xizmatchilar tomonidan olingan hamma buyurtmalarni aniqlash.

**SELECT ORDER\_NUM, REP, AMOUNT  
FROM ORDERS  
WHERE REP IN (107, 109, 101, 103)**

NOT ON yordamida diapazonga "tegishli emaslikni " tekshirish mumkin.

### *Operator LIKE*

Quyidagicha '%' shablonli LIKE operatorini qo'llaymiz:

```
SELECT COMPANY, CREDIT_LIMIT  
FROM CUSTIMERS  
WHERE COMPANY LIKE '%n'
```

Bu xolda LIKE '%n' operatori 'n' harfiga tugaydigan hamma yozuvlarni ko'rsatadi, agar '%' shablona birinchi kelsa:

```
SELECT COMPANY, CREDIT_LIMIT  
FROM CUSTIMERS  
WHERE COMPANY LIKE '%gan'
```

Ba'zida '%' shablona o'rniga '\*' belgisi qo'llanadi, masalan MS SQL uchun, c:\>dir \*.exe!

Agar faqat bitta simvol ixtiyoriy bo'lsa '\_' shablona qo'llanadi. Masalan:

```
SELECT COMPANY, CREDIT_LIMIT  
FROM CUSTIMERS  
WHERE COMPANY LIKE 'Ap_lsin'
```

### *Operator IS NULL*

Qo'llaymiz: Ofisga biriktirilmagan xizmatchini toping.

```
SELECT NAME FROM SALESREPS  
WHERE REP_OFFICE IS NULL
```

NOT shartini qo'llash mumkin: Ofisga biriktirilgan hamma xizmatchilarni toping.

```
SELECT NAME FROM SALESREPS  
WHERE REP_OFFICE IS NOT NULL
```

### *Yozuvlarni tartiblash, ORDER BY jumlasini*

Oldin ko'rilgan so'rovlarda natijalar ixtiyoriy tartibda olingan edi. Agar o'quvchilar ro'yxatini alfavit tartibida yoki tovarlar narxini kamayish tartibida chiqarish zarur bo'lsachi? Buning uchun SELECT operatori tarkibida ORDER BY ifodasi ko'zda tutilgan.

Avval quyidagi Misolni ko'ramiz: Har bir ofis uchun sotuvlar haqiqiy hajmlarini regionlar nomlari, har bir regionda esa shaharlar nomlari bo'yicha alfavit tartibida ko'rsatish.

**SELECT CITY, REGION, SALES  
FROM OFFICES  
ORDER BY REGION, CITY**

Ma'lumotlar bilan ishlash uchun mo'ljallangan BBT ichki tili ikki qismdan tashkil topgan:

Ma'lumotlarni aniqlash tili (DATA DEFINITION LANGUAGE) va ma'lumotlar bilan manipulyatsiya 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 ularning tarkibida barcha hisoblarni bajarish uchun zarur bo'lganda kerak bo'ladigan til konstruktsiyalari bo'lmaydi (shartli o'tish amallari, yoki sikl operatori). Ma'lumotlar bazasida ishlaydigan ixtiyoriy til foydalanuvchiga quyidagi imkoniyatlarni berishi kerak:

- Ma'lumotlar bazasi va jadvallarni tuzaolishi va tafsifini yarata olish;
- Ma'lumotlar bilan manipulyatsiya qilishning 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 tuzaolishi 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 strukturasi va sintaksi-sini (yozilishini) bir qancha BBTishlatishining imkonini beradi. Bu talablarning barchasiga SQL javob beradi. SQL-(STRUCTURE QUERY LANGUAGE) (strukturalashgan so'rovlar tili)SQL ma'lumotlarni almashtirish tili bo'lib hisoblanadi, ya'ni u jadvallar bilan ishlashga mo'ljallangan bo'lib, kiruvchi ma'lumotlar talab etilgan chiquvchi ko'rinishga ega xilga aylantirib beradi.

SQL tili ikkita asosiy komponentga ega:

- DDL tili (DATA DEFINITION LANGUAGE ma'lumotlarni aniqlash tili) bu ma'lumotlar baza strukturasi aniqlash va ma'lumotlarga murajaatni boshqarish uchun mo'ljallangan.

DML tili (DATA MANIPULATION LANGUAGE) ma'lumotlarni manipulyatsiyalash tili. Ma'lumotlarni ajratish va tiklash uchun mo'ljallangan bu til protsedura tili, shuning uchun unga qanday axborot olish kerakligi ko'rsatiladi, lekin axborotni qanday olish emas. Boshqacha aytganda SQL tili ma'lumot-larga murajaat usulini ko'rsatishni talab etmaydi.

SQL tili operatorlarni erkin formatda yozilishini ta'minlaydi. Buning ma'nosi, operatorlar elementlarining yozilishi ekrandan fiksirlangan joylarga bog'liq emas. Komanda strukturasi bir qancha kalit xizmatchi so'zlar bilan beriladi, masalan:CREATE TABLE (sozdat tablitse- jadval yaratish)INSERT (vstavka-qo'yish) SELECT (vibrat-ajratib olish)SQL operatori xizmatchi so'zlari va foydalanuvchi qo'llaydigan o'zgaruvchilaridan 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 qoidalariga asosida beriladi. Ular o`z navbatida ma'lumot baza obyektivlarini xar xil nomlaridan iborat bo`ladi (jadval, ustun, tasvirlar, indekslar va x.k.). Operatorlar so`zlar o`rnatilgan sintaksis qoidalariga moslab joylashtiriladi. Til standartida bu ko`rsatilmagan bo`lsa xam, SQL tilining dialektlarida matn tugallanganini bildiruvchi belgi, ko`pgina xollarda nuqtali vergul(;) ishlatiladi. SQL operatori komponentalarini ko`pchilligi registrga bog`liq emas, ya'ni ixtiyoriy xar qanday katta va kichik xarflar ishlatishi mumkin. Bularda bitta istuni bor. Bu istun simvulli literallarga tegishli bo`ladi. Ularda litera ma'lumotlar, ularga mos bo`lgan ma'lumotlar bazasidagi qiymatlar qanday saqlansa shunday yozilishi kerak. Masalan: agar ma'lumotlar bazasida familiyaning qiymati «SWITH» ko`rinishida bo`lsa, qidirish shartida «SWITH» simvol neteral ko`rinishida berilsa, bunga tegishli yozuv xech qachon topilmaydi. SQL tili erkin formatga ega bo`lgani uchun, SQLning aloxida operatorlari va ularning ketma-ketligini, aloxida ajratib yozish va tekislab yozishni ishlatish mumkin, bunda quyidagi qoidalarga bo`ysinish talab etiladi: -operatoridagi xar bir konstruksiya yangi satrdan boshlanishi kerak;

-xar bir konstruksiya boshlanishida tashlab ketiladigan bo`sh pozitsiyalardan tashqari, boshqa operator konstruksiyalari xam bo`lishi kerak;

-agar konstruksiya bir necha qismdan iborat bo`lsa, ularning xar biri qism yangi satrlaridan bo`sh o`rinlarning oldingi konstruksiyaga nisbatan siljitib yoziladi. Amaliyotda ma'lumotlar bazasi tuzaolishni (asosan uning jadvallarini) aniqlash uchun DDL operatorlari ishlatiladi, bu jadvallarni ma'lumotlar bilan to`ldirish uchun va ulardan axborotlarni so`rovlar yordamida ajratib olish uchun - DML operatorlari ishlatiladi.

Ma'lumotlar bilan manipulyatsiyalashda SQL tilini ishlatish vaqtida DML operatorlarini qo`llashni amalga oshirish mumkin.

SQL tilining ikkita shakli mavjud, shu ikki shakl dastur moboynda ishlatiladi.

-interaktiv SQL

- qurilgan (kiritilgan) SQL

Interaktiv SQLda foydalanuvchi SQL- so`rovlarni va natija interaktiv rejimida olinadi. Ko`rilgan SQL, SQL komandalaridan tashkil topib, u boshqa birorta tilga (S++, Delphi) da yozilgan dastur ichiga joylashtiriladi. Bu shunday tillarda ishlatilganda dasturlarni samarador, quvvatli qiladi. Ularga Relyatsion ma'lumotlar bazasi bilan ishlash imkonini beradi. SQL ma'lumot toifalari. Simvollar satr ma'lumotlar toifasi SQL standartida matnlarni faqat bitta tavsifi keltiriladi. Uning sintaksisi CHARACTER[(uzunligi)] yoki CHAR[(qiymati)] jadvalning matnli qiymati CHAR toifasidagi fiksirlangan uzunlikda bo`lishi mumkin. Bu parametr qiymati 1- 255 bo`lishi mumkin, ya'ni u 255 simvolgacha bo`lishi mumkin. SQL tilini ba'zi birlaridagina o`zgaruvchan uzunlikdagi satr toifalari bor. Bu toifalar quyidagicha tavsiflanadi:

VARCHAR(), CHARVALUE yoki CHAR VARYING()

Ixtiyoriy uzunlikdagi matnli tasvirni tasvirlaydi. CHARACTER va VARCHAR toifasidagi konstantalar apostrof ichiga yoziladi.

Quyidagi yozuvlarni barchasi ekvivalent VARCHAR [(UZUNLIGI)], CHARVARYING [(UZUNLIGI)] CHARACTER VARYING[(UZUNLIGI)]

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

Sonli ma`lumot toifalari SQL standartida quyidagi son toifasida ishlatiladi. INTEGER-butun sonlar uchun -2 -31...2

SMILLINT-butun sonlar 2-15...2

DECIMAL (aniqlik[masshtab]) -Fiksirlangan nuqtali o`nli son aniqlik sondagi qiymatli raqamlar masshtabi o`nli nuqtadan undagi raqamlarning maksimal sonini ko`rsatadi NUMERIC(aniqlik[masshtab])- Fiksirlangan nuqtali o`nli son, aniqlik sondagi qiymatli raqamlar masshtabi o`nli nuqtadan undagi raqamlarning maksimal sonini ko`rsatadi FLOAT[(aniqlik)]suzuvchi nuqtali son monomal aniqlik bilan beriladi. Shunday qilib xulosa qilish mumkin:

1. Simvolli satrlar toifasi. Character [uzunligi] [char] uzun) da ko`rsatiladi.

Bundan tashqari o`zgaruvchan uzunlikdagi simvolli satrlar toifasini xam ishlatamiz. Bunda o`zgaruvchi toifalar ixtiyoriy uzunlikda bo`ladi.

Bunda uzunliklar zarur bo`lmagan parametrlar hisoblanadi. Agar ular ishlatilmasa, unda 1 ta simvolga joy ajratiladi.

Simvolli satrlarni belgilashni yana bir usuli bor.

Varchar [(uzunlik)] yoki charvaryona [(uzunlik)]

2. Ma`lumotlarning sonli tiplari

1. Integer

2. Smollint

3. Decimal (aniqlik, masshtab). (DEC) fiksirlangan (\_\_\_) li sonlarni tasvirlash uchun ishlatiladi.

Aniqlik- sondagi qiymatli raqamlar. Masshtab- nuqtadan keyingi o`ng tomonda turgan raqamlarni maksimal soni.

4. Numeric (aniqlik, (masshtab)..

5. FLOAT (aniqlik).Suzuvchi nuqtali son va undagi monomal aniqlikni bildiradi.

6. REAL FLOAT operatorlari tup sonlar bilan qo`shimcha ish olib boradi.Sana va vaqt toifasidagi ma`lumotlar standarti qo`shimcha qilinmagan.

Bularning yozolishini texnik xujjatlarda ko`rish mumkin.

SQL da atribut qiymatlari noma'lum bo`lgan o`tkazib yuborilgan yoki mavjud bo`lmaganlarini NULL bilan yoziladi. NULL qiymati oddiy tushunchada qiymat hisoblanmaydi. U faqat atributning xaqiqiy qiymati tushib qoldirilgan yoki noma'lumligini aniglatadi. NULL ni ishlatishda quyidagilarga e'tibor berish kerak:

- shartli operatorlarda TRUE, FALSE dan tashqari UNKIWN paydo bo`lsa natija NULL qiymatda chiqadi.

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

- almashtirish funktsiyalari xam argument sifatida NULL bo`lsa natija NULL ga teng bo`ladi.

Misol uchun quyidagi dastur asosida ma`lumotlar bilan ishlash texnologiyasini ko`rib o`tamiz.

SQL buyruqlari yordamida kontekst menyuni tarqatish usullari dasturi

```
usesDataModule;
```

```
{ $ R*.dfm }
```

```
{ Uzoqdagi tanlab olingan yozuv }
```

```
pricedureTFirml.DeleteRecClick(Sender:Tlbject);
```

```
begin
```

```
if PipupMenul.PopupComponent=DB Gridlthen begin
```

```
with DataModels.StudentsTbld obegin
```

```
if State=dsBriwsethen
```

```
if MessageDlg(' Siz yozilgan malumotni o`chirishni hohlaysizmi?',
```

```
mtCinformation.[mbYes.mbNi] ,0)=mrYe sthen
```

```
Delete;
```

```
end;
```

```
end;
```

```
if PipupMenul.PopupComponent=DBGrid2 then begin
```

```
with DataModels.GradeTbl di begin
```

```
if State=dsBriwse then
```

```
if MessageDlg(' Siz yozilgan malumitni o`chirishni hohlaysizmi?',
```

```
mtCinformation,[mbYes.mbNi].0)=mrYes then
```

```
Delete;
```

```
end;
```

```
end;
```

```
end;
```

```
{ Saqlanishda o`zgarish }
```

```
Pricedure TForml.PostRec Click(Sender:TI bject):
```

```
begin
```

```
if Pipup Menul. Popup Componen t=DBGridl the nbegin
```

```
with DataModels.Students Tbl di begin
```

```
if Statei n[dsInsert, dsEdit] the n
```

```
Pist;
```

```
end:
```

```
end ;
```

```
if PipupMenul. Pipup Component=DB Grid2 then begin
```

```
with DataModels.Grade Tbl di begin
```

```

    if State in[dsInsert.ds Edit] then
        Pist;
    end;
end;
end;
{Yangi ma`lumot qo`shish}
Pricedure TFirml.NewRecClick(Sender:TIBject);
begin
    if PipupMenul.PipupComponent=DB Grid1 then begin
        with DataModels.Students Tbl di begin
            if State=ds Briwse then
                Insert;
            end;
        end;
    end;
    if PipupMenul.PipupComponent=DBGrid2 then begin
        withDataModels.GradeTbldi begin
            if State=dsBriwse then
                Insert;
            end;
        end;
    end;
end;
end;

```

## **SQL tarkiblangan so`rovlar tili yordamida turli Jadvallar uchun cheklanishlar o`rnatish.**

Cheklanishlarni kiritish.

Jadval yaratayotganingizda (yoki uni o`zgartirayotganingizda), siz maydonlarga kiritilayotgan qiymatlarga cheklanishlar o`rnatishingiz mumkin. Bu xolda SQL cheklanishlarga to`g`ri kelmaydigan hamma qiymatlarni rad etadi. Cheklanishlarning ikki asosiy turi mavjud: - ustun va jadval cheklanishlari. Ularning farqi shundaki ustun cheklanishi faqat ayrim ustunlarga qo`llanadi, jadval cheklanishi bo`lsa bir yoki bir necha ustunlar guruhiga qo`llanadi. Ustun cheklanishi ustun nomi oxiriga ma`lumotlar tipidan so`ng va verguldan oldin qo`yiladi. Jadval cheklanishi jadval nomi oxiriga so`nggi dumaloq verguldan oldin qo`yiladi. Cheklanishlar hisobga olingan CREATE TABLE komandasi sintaksisi qiyidagicha:

```

CREATE TABLE < table name >
( <cilumn name> <data type> <cilumn cinstraint>,
<cilumn name> <data type> <cilumn cinstraint> ...
<table cinstraint> ( <cilumn name>
[, <cilumn name> ]).. );

```

Maydonga bo`sh (NULL) qiymatlar kiritaolishining oldini olish uchun CREATE TABLE komandasida NOT NULL cheklanishi ishlatiladi. Bu cheklanish faqat har xil ustunlar uchun o`rnatiladi.

Masalan, shu narsa aniqki, birlamchi kalitlar hech qachon bo`sh bo`lmasliklari kerak, shuning uchun Salepeople jadvalini quyidagicha yaratish mumkin:

```
CREATE TABLE Salepeople  
( Snum integer NOT NULL,  
  Sname char (10),  
  city char (10),  
  cimm decimal);
```

Ko`p xollarda ustunga kiritilgan qiymatlar bir biridan farq qilishi kerak. Agar ustun uchun UNIQUE cheklanishi o`rnatilsa, bu ustungasha mavjud qiymatni kiritish va o`rinish rad etiladi. Bu cheklanish bo`sh bo`lmaydigan (NOT NULL) deb e`lon qilingan maydonlarga qo`llanishi mumkin.

Masalan:

```
CREATE TABLE Salepeople  
( SNum integer NIT NULL UNIQUE,  
  SName char (10),  
  City char (10),  
  Cimm decimal);
```

Unikalligi talab qilinadigan maydonlar(birlamchi kalitlardan tashqari) nomzod kalitlar yoki unikal kalitlar deyiladi.

Jadval cheklanishi UNIQUE maydonlar guruhiga o`rnatilishi mumkin. Bu bir necha maydonlar qiymatlari kombinatsiyasi unikalligini ta`minlaydi. Bizning ma`lumotlar bazamizda har bir buyurtmachi bitta sotuvchiga birlashtirilgan. Ya`ni buyurtmachilar jadvalida buyurtmachi nomeri (cnum) va sotuvchi nomeri (snum) kombinatsiyasi unikal bo`lishi kerak. Bu cheklanishni UNIQUE (cnum, snum) yordamida, Custimers jadvalini yaratishda kiritish mumkin. Bu ustunlar uchun NOT NULL cheklanishini kiritish zarurdir.

SQL birlamchi kalitlarni to`g`ridan to`g`ri birlamchi kalit (PRIMARY KEY) cheklanishi orqali ta`riflaydi. PRIMARY KEY jadvalini yoki ustunlarni cheklashi mumkin. Bu cheklanish UNIQUE cheklanishi kabi ishlaydi, jadval uchun faqat bitta birlamchi kalit (ixtiyoriy sondagi ustunlar uchun ) aniqlanishi mumkin bo`lgan xoldan tashqari. Birlamchi kalitlar NULL qiymatga ega bo`lishi mumkin emas.

Misol:

```
CREATE TABLE Salepeople  
( SNum integer NIT NULL Primary key,  
  SName char (10),  
  City char (10),  
  Cimm decimal);
```

PRIMARY KEY cheklanishi qiymatlar unikal kombinatsiyasini tashkil qiluvchi bir necha maydonlar uchun qo`llanishi mumkin. Masalan PRIMARY KEY cheklanishini juftliklar uchun qo`llash mumkin:



```
CREATE TABLE Namefoeld
  (firstname char (10) NIT NULL,
lastname char (10) NIT NULL
city char (10),
PRIMARY KEY (firstname, lastname ));
```

### **Maydon qiymatlarini tekshirish (CHECK cheklanishi).**

CHECK cheklanishi jadvalga kiritilayotgan ma'lumot qabul qilinishidan oldin mos kelishi lozim bo'lgan shart kiritishga imkon beradi. CHECK cheklanishi CHECK kalit so'zi ko'rsatilgan maydondan foydalanuvchi predikat ifodalaridan iboratdir.

Misol: Salepeople jadvali Comm ustuniga kiritilayotgan qiymat 1 dan kichik bo'lish sharti.

```
CREATE TABLE Salepeople
  (SNum integer NIT NULL PRIMARY KEY,
SName char(10) NIT NULL UNIQUE,
City char(10),
Cimm decimal CHECK ( Cimm < 1 ));
```

CHECK cheklanishidan maydonga ma'lum qiymatlarini kiritishdan ximoya qilib, xatolar oldini olish uchun foydalanish mumkin. Masalan maxsulotni sotish shaxobchalariga ega bo'lgan shaharlar faqat Toshkent, Ashxabad, Moskva va Astona bo'lsin.

```
CREATE TABLE Salepeople
  (SNum integer NIT NULL PRIMARY KEY,
SName char(10) NIT NULL UNIQUE,
City char(10) CHECK (City IN (' Toshkent ', Ashxabad ', Moskva
', Astona ')),
Cimm decimal CHECK ( Cimm < 1 ));
```

CHECK jadval cheklanishi sifatida kelishi mumkin. Bu shartga bir necha maydon kiritishga imkon beradi.

Masalan:

```
CREATE TABLE Salepeople
  (SNum integer NOT NULL PRIMARY KEY,
SName char(10) NOT NULL UNIQUE,
City char(10),
Cimm decimal,
CHECK (Simm < .15 IR City = ' Ashxabad')));
```

Biror bir maydon uchun qiymat ko'rsatmagan xolda jadvalga satr qo'shsangiz, SQL bunday maydonga kiritish uchun ko'zda tutilgan qiymatga ega bo'lishi kerak, aks xolda komanda rad etiladi. Eng umumiy ko'zda tutilgan qiymat NULL

qiymatdir. CREATE TABLE komandasida ko'zda tutilgan qiymat DEFAULT operatori orqali, ustun cheklanishi sifatida ko'rsatiladi. Masalan:

```
CREATE TABLE Salepeople  
( SNum integer NOT NULL PRIMARY KEY,  
SName char(10) NOT NULL UNIQUE,  
City char(10) DEFAULT Moskva ',  
Cimm decimal CHECK ( Cimm < 1 ));
```

Jadval bir maydonidagi hamma qiymatlar boshqa jadval maydonida aks etsa, birinchi maydon ikkinchisiga ilova qiladi deyiladi. Bu ikki maydon orasidagi bog'liqlikni ko'rsatadi. Masalan, buyurtmachilar jadvalida har bir buyurtmachi, sotuvchilar jadvalida o'ziga birlashtirilgan sotuvchiga ilova qiluvchi SNum maydoniga ega. Bir maydon ikkinchisiga ilova qilsa tashqi kalit, u ilova qilayotgan maydon ajdod kalit deyiladi. Buyurtmachilar jadvalidagi SNum maydoni tashqi kalit, sotuvchilar jadvalidagi SNum - ajdod kalitdir.

Tashqi kalit bitta maydondan iborat bo'lishi shart emas. Birlashtiruvchi kalit kabi, tashqi kalit bitta modul sifatida qayta ishlanuvchi bir necha maydonlarga ega bo'lishi mumkin. Maydon tashqi kalit bo'lsa ilova qilayotgan jadval bilan ma'lum usulda bog'liqdir. Tashqi kalitning har bir qiymati (satri), ajdod kalitning bitta va faqat bitta qiymatiga (satriga) ilova qilishi kerak. Bu xolda tizim ilovalari yaxlit xolatda deyiladi. Shu bilan birga ajdod kalit qiymati tashqi kalit bir necha qiymatlariga ilova qilishi mumkin.

SQL ilovalari yaxlitlikni FOREIGN KEY yordamida ta'minlaydi. Tashqi kalit vazifasi ajdod kalitda ko'rsatilmagan qiymatlarni tashqi kalit maydonlariga kiritmaslikdir. FOREIGN KEY cheklanishi sintaksisi:

```
FOREIGN KEY <column list> REFERENCES  
<pktable> [<column list>]
```

Birinchi ro'yxat komanda tomonidan o'zgartiriluvchi ustunlar ro'yxatidir. Pktable - bu ajdod kalitli jadval. Ikkinchi ustunlar ro'yxati bu ajdod kalitni tashkil qiluvchi ustunlardir.

Misol uchun Sotuvchilar jadvaliga ilova qiluvchi tashqi kalit sifatida e'lon qilingan SNum maydoniga ega bo'lgan buyurtmachilar jadvalini yaratamiz:

```
CREATE TABLE Customers  
( CNum integer NOT NULL PRIMARY KEY,  
CName char(10),  
City char(10),  
SNum integer,  
FOREIGN KEY (SNum) REFERENCES Salepeople (SNum) );
```

Tashqi kalitni ustunlar cheklanishi sifatida berish mumkin. Buning uchun FOREIGN KEY - ko'rinishi - ko'rsatkichli cheklanish (REFERENCES) qo'llanadi:

```
CREATE TABLE Customers
```

( **CNum integer NOT NULL PRIMARY KEY,**  
**CName char(10),**  
**City char(10),**  
**SNum integer REFERENCES Salepeople (snum) );**

FIREOGN KEY cheklanishidan jadval yoki ustun cheklanishi sifatida foydalanganda ajdod kalit ustunlarini ko`rsatmaslik mumkin, agar ular PRIMARY KEY cheklanishiga ega bo`lsa.

Kalitlarga cheklanish.

Ilovali yaxlitlikni ta'minlash tashqi kalit yoki ajdod kalit maydonlari qiymatlariga cheklanishlar o`rnatishni talab qiladi. Ajdod kalit tarkiblangan bo`lib , tashqi kalit har bir qiymati bitta satrga mos kelishi ta'minlangan bo`lishi kerak. Bu kalit unikal bo`lib , bo`sh (NULL) qiymatlarga ega bo`lmasligi kerak. Shuning uchun ajdod kalit maydonlari PRIMARY KEY cheklanishiga ega bo`lishi yoki NOT NULL cheklanishi bilan birga UNIQUE deb e`lon qilinishi kerak.

Tashqi kalit ajdod kalitda mavjud qiymatlarga yoki bo`sh (NULL) qiymatga ega bo`lishi mumkin. Boshqa qiymat kiritishga urinish rad etiladi. Tashqi kalitga NOT NULL deb e`lon qilish mumkin, lekin bu maqsadga muvofiq emas. Masalan, siz qaysi sotuvchi mos kelishini bilmasdan oldin buyurtmachini kiritmoqchisiz. Bu xolda NULL qiymatdan foydalanib, keyinchalik uni aniq qiymatga almashtirish mumkin.

Tashqi kalit maydonlariga INSERT yoki UPDATE yordamida kiritilayotgan qiymatlar ajdod kalitlariga oldin kiritilgan bo`lishi kerak. Tashqi kalit ixtiyariy satrini DELETE yordamida o`chirish mumkin. ANSI ta`rifi bo`yicha: tashqi kalit yordamida ilova qilinayotgan ajdod kalit qiymatini o`chirib yoki o`zgartirib bo`lmaydi. Bu shuni bildiradiki buyurtmalar jadvalida buyurtmalarga ega buyurtmachini buyurtmachilar jadvalidan o`chirib bo`lmaydi. ANSI tarkibiga kirmagan ajdod kalit maydonlarini o`zgartirish yoki o`chirish qoidalari mavjud:

- 1) Cheklangan (RESTRICT) o`zgartishlar. Siz (ANSI usulida) ajdod kalitlarda cheklangan deb ko`rsatishingiz yoki man qilishingiz mumkin.
- 2) Kaskadlanuvchi (CASCADE) o`zgartishlar. Agarda ajdod kalitga o`zgartish kiritsangiz, tashqi kalitda xuddi shunday o`zgartishlar avtomatik yuz beradi.
- 3) Bo`sh (NULL) o`zgartishlar. Siz ajdod kalitda o`zgartirish kiritganingizda tashqi kalit maydonlari avtomatik NULL qiymat oladi (tashqi kalitda NULL qiymat ruxsat etilgan bo`lsa).

Yuqorida ko`rsatilgan effektlar UPDATE va DELETE komandalari bajarilganda ajdod kalit o`zgarishini ko`rsatadi va quyidagicha aniqlanadi:

**CREATE TABLE <table-name >**

**( <cilumn name> <data type>[(<size>)],**

**<cilumn name> <data type>[(<size>)],**

**...**

**FIREGIGN KEY (<cilumn name>,...) REFERENCES <table name>[(<cilumn name>, ...)]**

**IN UPDATE [CASCADE|RESTRICT|SET NULL]  
IN DELETE [CASCADE|RESTRICT|SET NULL],  
... );**

Misol. Siz sotuvchi nomerini o'zgartirmoqchisiz, lekin uning hamma buyurtmachilarini saqlab qolmoqchisiz. Lekin bu sotuvchi firmadan bo'shab ketsa siz uning buyurtmachilarini boshqa sotuvchiga maxkamlashingiz kerak. Buni bajarish uchun kaskad effektli UPDATE va cheklanishli DELETE berishingiz kerak.

**CREATE TABLE Custimers  
(CNum integer NIT NULL PRIMARY KEY,  
CName char(10) NIT NULL,  
City char(10),  
Rating integer,  
SNum integer REFERENCES Salepeople  
IN UPDATE CASCADE  
IN DELETE RESTRICT);**

Agar endi sotuvchilar jadvalidan Peel ni o'chirmoqchi bo'lsangiz, to buyurtmachilar jadvalida Hoffman va Clemens ning SNum maydonini boshqa tayinlangan sotuvchiga o'zgartirishingiz kerak. Boshqa tomondan Peel SNum maydonini 1009 ga o'zgartirsangiz Hoffman va Clemens ham avtomatik o'zgaradi.

Komanda yoki komandalar guruhi bajarilgandan so'ng o'zgartishlar ma'lumotlar bazasida saqlanib qolishi yoki rad etilishini xal qilishingiz lozim. Bu maqsadda komandalar tranzaktsiya deb ataluvchi guruhlariga birlashtiriladi.

Har doim SQL seans boshlaganingizda tranzaktsiya ham boshlanadi. Hamma komandalar tranzaktsiya qismi hisoblanadi, toki ularni COMMIT yoki ROLLBACK komandasi kiritib tugatmaguningizcha. COMMIT o'zgarishlarni doimiy qiladi, ROLLBACK bo'lsa rad qiladi. Yangi tranzaktsiya COMMIT yoki ROLLBACK komandasidan so'ng boshlanadi.

Ko'pgina realizatsiyalarda siz AUTOCOMMIT parametrlarini o'rnatishingiz mumkin. Bu hamma qadamlarni avtomatik eslab qoladi. Xatoga olib keluvchi qadamlar teskarisiga bajariladi. Buni quyidagicha bajarish mumkin:

SET AUTOCOMMIT ON; Oldingi xolatga quyidagicha qaytish mumkin SET AUTOCOMMIT OFF;

Ba'zi komandalar ya'ni ALTER, CREATE, DROP, GRANT, REVOKE, kabi COMMIT ni avtomatik bajaradi.

### **Maydonlarni kiritish, o'chirish va o'zgartirish. Qiymatlarni kiritish.**

Hamma satrlar SQLda INSERT komandasi yordamida kiritiladi. INSERT quyidagi formatlardan biriga ega bo'lishi mumkin:

**INSERT INTO <table name | view name> [(column [,column] ...)]**

**VALUES (<value> [,<value>] ... );**

yoki

**INSERT INTO <table name | view name> [(cilumn [,cilumn] ...)]**

Ostki so`rov;

Masalan, sotuvchilar jadvaliga satr kiritish uchun quyidagi shartdan foydalanishingiz mumkin:

**INSERT INTO Salepeople**

**VALUES (11, 'Peel', 'Astona', .12);**

Siz nom kiritish uchun ustunlar ko`rsatishingiz mumkin. Bu nomlarni ixtiyoriy tartibda kiritishga imkon beradi. Masalan:

**INSERT INTO Salepeople (Sname, Cimm, SNum)**

**VALUES ('Peel', .12, 11);**

E`tibor bering City ustuni tashlab yuborilgan, chunki unga ko`zda tutilgan qiymat kiritiladi.

Siz INSERT komandasidan bir jadvaldan qiymat tanlab, so`rov bilan ishlatish uchun, ikkinchisiga joylashishda foydalanishingiz mumkin. Buning uchun siz VALUES ifoda soni (oldingi Misoldagi) mos so`rovga almashtiringiz kerak:

**INSERT INTO Astinastaff**

**SELECT \* FROM Salespeople**

**WHERE City = 'Astona';**

Satrlarni jadvaldan DELETE komandasi bilan o`chirish mumkin. U aloxida qiymatlarni emas faqat satrlarni o`chiradi. DELETE quyidagi formatga ega:

**DELETE FROM <table name | view name>**

**[WHERE search-cindITION];**

Masalan, Sotuvchilar jadvalidagi hamma satrlarni o`chirish uchun, quyidagi shartni kiritish mumkin:

**DELETE FROM Salepeople;**

Ma`lum satrlarni o`chirish uchun predikatdan foydalaniladi. Masalan, jadvaldan Axelrod sotuvchini o`chirish uchun:

**DELETE FROM Salepeople**

**WHERE SNum = 13;**

Bu o`zgartirish UPDATE komandasi yordamida bajariladi. Bu komandada UPDATE ifodasidan so`ng jadval nomi va SET ifodasidan so`ng ma`lum ustun uchun o`zgartirish ko`rsatiladi. UPDATE ikki formatga ega. Ulardan birinchisi:

**UPDATE <table name | view name>**

**SET cilumn = expression [, cilumn = expression] ...**

**[WHERE search-cindition]**

bu erda expressiin - bu ustun | ifoda | konstanta | o`zgaruvchi.

Ikkinchi variant:

**UPDATE <table name>**  
**SET cilumn = expression, ...**  
**[ FROM table-list ]**  
**[ WHERE search-cindition ]**

Masalan, hamma buyurtmachilar baxosoni 200 ga o'zgartirish uchun quyidagini kiritishingiz mumkin:

**UPDATE Custimers**  
**SET Rating = 200;**

Ma'lum satrlarni o'zgartirish uchun DELETE dagi kabi predikatdan foydalanish kerak. Masalan Peel (SNum=11 ) sotuvchining hamma buyurtmachilari uchun bir xil o'zgartirishni quyidagicha kiritish mumkin:

**UPDATE Custimers**  
**SET Rating = 200**  
**WHERE SNum = 11;**

SET vergul bilan ajratilgan ixtiyoriy sondagi ustunlarga qiymat tayinlashi mumkin. Ixtiyoriy jadval satrlari uchun qiymat tayinlanishi mumkin, lekin bir vaqtning o'zida faqat bitta satrga qiymat tayinlanadi. Masalan:

**UPDATE Salepeople**  
**SET SName = 'Gibsin', City = 'Boston', Cimm = .10**  
**WHERE SNum = 14;**

Siz UPDATE komandasining SET jumlasida skalyar ifodalardan o'zgartirilayitgan maydon ifodasiga qo'shgan xolda foydalanishingiz mumkin. Masalan:

**UPDATE Salepeople**  
**SET Cimm = Cimm \* 2;**

SELECT operatori BB jadvallaridan natijaviy to'plam olish uchun mo'ljallangan ifodadir. Biz SELECT operatori yordamida so'rov beramiz, u bo'lsa ma'lumotlar natijaviy to'plamini qaytaradi. Bu ma'lumotlar jadval shaklida qaytariladi. Bu jadval keyingi SELECT operatori tomonidan qayta ishlanishi mumkin va xokazo. Operator SQL92 standartiga ko'ra quyidagi ko'rinishga ega:

**SELECT -- ALL ----- sxema , ustun ----**  
**-- DISTINCT -- ---- \* -----**  
**FROM -- sxema , Jadval .. -----**  
**WHERE -- izlash sharti -----**  
**GROUP BY -- sxema , ustun -----**  
**HAVING -- izlash sharti -----**  
**ORDER BY – tartiblash spetsifikatori -----**

Birinchi qoida, SELECT ifodasi o'zichiga albatta FROM ifodasini olishi kerak. Qolgan ifodalar kerak bo'lsa ishlatiladi.

SELECT ifodasidan so'ng so'rovda qaytariluvchi ustunlar ro'yxati yoziladi.

FROM ifodasidan so'ng so'rovni bajarish uchun jadvallar nomi yoziladi.

WHERE ifodasidan so'ng agar ma'lum satrlarni qaytarish lozim bo'lsa, izlash sharti yoziladi.

GROUP BY ifodasi guruhlariga ajratilgan natijaviy so'rov yaratishga imkon beradi.

HAVING ifodasidan guruhlarini qaytarish sharti yoziladi va GROUP BY bilan birga ishlatiladi.

ORDER BY ifodasi ma'lumotlar natijaviy to'plamini tartiblash yo'nalishini aniqlaydi.

OFFICES jadvalidagi hamma yozuvlarni qaytaruvchi sodda so'rov ko'ramiz.

**SELECT \* FROM OFFICES**

***SELECT* yordamida ma'lumotlarni tanlash**

SELECT operatori albatta "qaytariluvchi ustunlar ro'yxati " ni o'zichiga olishi kerak, ya'ni:

**SELECT FILED1, FIELD2, FIELD3 ... FROM ...**

FILED1, FIELD2, FIELD3 qaytariluvchi ustunlar ro'yxati bo'lib, ma'lumotlar ketma ketligi shu tartibda qaytariladi!

Ya'ni "qaytariluvchi ustunlar ro'yxati" hisoblanuvchi ustunlar va konstantalarni o'zichiga olishi mumkin.

**SELECT FILED1, (FIELD2 - FIELD3) "CINST" ... FROM ...**

FROM jumlasini "jadval spetsifikatorlari", ya'ni so'rovni tashkil qiluvchi jadvallar nomini o'zichiga oladi. Bu jadvallar so'rov asosini tashkil qiluvchi jadvallar deyiladi.

Misol: Hamma xizmatchilarning nomlari, ofislari va ishga olish sanalari ro'yxatini xosil qilish.

**SELECT NAME, REP\_OFFICE, HIRE\_DATE FROM SALESREPS**

SELECT operatori qaytaruvchi ustunlar ixtiyoriysi hisoblanuvchi, ya'ni natijada mustaqil ustun sifatida tasvirlanuvchi matematik ifoda bo'lishi mumkin.

Misol: Har bir ofis uchun shaharlar, regionlar va sotuvlar rejasi qanchaga ortishi yoki kami bilan bajarilganligi ro`yxatini ko`rib o`tamiz.

**SELECT CITY, REGION, (SALES-TARGET) FROM OFFICES**

Har bir xizmatchi uchun rejadagi sotuvlar hajmini haqiqiy sotuvlar hajmining 3% foiziga oshirish!

**SELECT NAME, QUITA, (QUITA +((SALES/100)\*3)) FROM SALESREPS**

Ba`zida ustunlardan biri izlash shartiga bog`liq bo`lmagan qiymat qaytarishi kerak bo`ladi!

Masalan: Har bir shahar uchun sotuvlar hajmlari ro`yxatini chiqaring.

**SELECT CITY, 'Has sales if', SALES FROM OFFICES**

'Has sales if' bu konstantalar ustunidir.

Ba`zida ma`lumotlarni tanlashda qaytariluvchi qiymatlar xosil bo`ladi.

Bu xol yuz bermasligi uchun DISTINCT operatoridan foydalanish lozim. Masalan, quyidagicha:

**SELECT DISTINCT MGR FROM OFFICES**

**SELECT operatori WHERE sharti**

Endi WHERE ifodasidan foydalanib ba`zi so`rovlarni ko`rib chiqamiz: Sotuvlar haqiqiy hajmi rejadan oshgan ofislarni ko`rsating.

**SELECT CITY, SALES, TARGET FROM OFFICES WHERE SALES > TARGET**

Identifikatori 105 ga teng bo`lgan xizmatchi nomi haqiqiy va rejadagi sotuvlar hajmini ko`rsating:

**SELECT SALES, NAME, QUITA FROM SALESREPS WHERE EMPL\_NUM = 105**

Agar izlash sharti TRUE, bo`lsa qator natijaviy to`plamga qo`shiladi, agar izlash sharti FALSE bo`lsa, qator natijaviy to`plamga qo`shilmaydi, agar NULL bo`lsa ham natijaviy to`plamdan chiqariladi! O`zma`nosiga ko`ra WHERE, keraksiz yozuvlarni chiqarib, kerakligini qoldiruvchi filtr sifatida ishlatiladi!

Asosiy izlash shartlari "predikatlar", beshta. Ularni ko`rib chiqamiz:

1. Solishtirish, ya`ni bir shart natijasi ikkinchisi bilan solishtiriladi. Birinchi so`rov kabi.

2. Qiymatlar diapazoniga tegishlilikni tekshirish. Masalan berilgan qiymat diapazonga kiradimi yo`qmi.

3. To`plam elementiligini tekshirish. Masalan, ifoda qiymati to`plamdagi biror qiymat bilan ustma ust tushadimi.



4. Shablonga moslikni tekshirish. Ustundagi satrli qiymat shablonga mos keladimi.

5. NULL qiymatga tenglikka tekshirish.

Solishtirish amallari maydon va konstantalarni solishtirish amallarini o'zichiga olishi mumkin: 2015 yilgacha ishga olingan hamma xizmatchilar nomlarini toping.

```
SELECT NAME FROM SALESREPS
```

```
WHERE HIRE_DATE < TI_DATE('01.06.2015','DD/MM/YYYY')
```

```
TI_DATE('01.06.2015','DD/MM/YYYY') - PL/SQL Oracle sana bilan  
ishlash standart funksiyasi.
```

Yoki arifmetik ifodalarni o'zichiga olishi mumkin: Haqiqiy sotuvlar hajmi rejaning 80 foizidan kam bo'lgan ofislar ro'yxatini chiqaring.

```
SELECT CITY, SALES, TARGET FROM OFFICES
```

```
WHERE SALES < (0.8 * TARGET)
```

Ko'p xollarda izlash birlamchi kaliti bo'yicha konstantalar bilan solishtirish so'rovlaridan foydalaniladi, masalan shahar telefon tarmog'i abonentlari, axir ikkita bir xil nomerlar mavjud emas!

## **Mantiqiy operatorlar**

### ***BETWEEN va IN Operatorlari***

BETWEEN ifodasi bu qiymatlar diapazoniga tegishlilikni tekshirishdir. Ifoda sintaksisi quyidagicha:

**--- tekshirilayotgan ifoda ----- BETWEEN ----- quyi ifoda AND yuqori ifoda - NOT -**

NOT ifodasi shartni teskarisiga o'giradi, ya'ni tegishli emas ma'noni bildiradi.

Misol: Narxi har xil diapazonga mos keluvchi buyurtmalarni topish.

```
SELECT ORDER_NUM, AMOUNT
```

```
FROM ORDERS
```

```
WHERE AMOUNT BETWEEN 20.000 AND 29.999
```

NOT ifodasi yordamida berilgan diapazonga tegishlilikni tekshirish mumkin, masalan: Sotuvlar haqiqiy hajmlari rejaning 80 dan 120 prosentigacha bo'lgan diapazonga tushmaydigan xizmatchilar ro'yxatini chiqarish.

```
SELECT NAME, SALES, QUITA
```

```
FROM SALESREPS
```

```
WHERE SALES NOT BETWEEN (0.8 * QUITA) AND (1.2 * QUITA)
```

Ifoda IN to`plamga tegishlilikni tekshiradi. Komanda sintaksisi quyidagicha:

```
--- tekshirilayotgan ifoda ----- IN ----- (-- cinst -----)
      - NOT -           -- , -----
```

2015 yil iyun oyining har xil kunlarida qilingan hamma buyurtmalarni aniqlash.

```
SELECT ORDER_NUM, ORDER_DATE, AMOUNT
FROM ORDERS
WHERE ORDER_DATE IN (TI_DATE('14.06.2015','DD/MM/YYYY'),
TI_DATE('08.06.2015','DD/MM/YYYY'),
TI_DATE('29.06.2015','DD/MM/YYYY'),
TI_DATE('04.06.2015','DD/MM/YYYY'))
```

Sanalar bilan shu tarzda ishlanadi.

To`rtta aniq xizmatchilar tomonidan olingan hamma buyurtmalarni aniqlash.

```
SELECT ORDER_NUM, REP, AMOUNT
FROM ORDERS
WHERE REP IN (107, 109, 101, 103)
```

NOT IN yordamida diapazonga "tegishli emaslikni " tekshirish mumkin.

### *Operator LIKE*

LIKE ifodasi sintaksisi SQL92 standarti bo`yicha quyidagi ko`rinishga ega:

```
--- ustun nomi----- LIKE (shablon) -----
      NOT           ESCAPE (o`tkazish nomi)
```

Sodda so`rov bajaramiz: "Apelsin" kompaniyasi uchun kredit limitini ko`rsatish:

```
SELECT CIMPANY, CREDIT_LIMIT
FROM CUSTIMERS
WHERE CIMPANY = 'Apelsin'
```

Quyidagicha '%' shablonli LIKE operatorini qo`llaymiz:

```
SELECT CIMPANY, CREDIT_LIMIT
FROM CUSTIMERS
WHERE CIMPANY LIKE '%n'
```

Bu xolda LIKE '%n' operatori 'n' harfiga tugaydigan hamma yozuvlarni ko`rsatadi, agar '%' shablona birinchi kelsa:

```
SELECT CIMPANY, CREDIT_LIMIT  
FROM CUSTIMERS  
WHERE CIMPANY LIKE '%gan'
```

Ba`zida '%' shablona o`rniga '\*' belgisi qo`llanadi, masalan MS SQL uchun, c:\>dir \*.exe!

Agar faqat bitta simvol ixtiyoriy bo`lsa '\_' shablona qo`llanadi. Masalan:

```
SELECT CIMPANY, CREDIT_LIMIT  
FROM CUSTIMERS  
WHERE CIMPANY LIKE 'Ap_lsin'
```

### *Operator IS NULL*

SELECT operatori uchun NULL qiymati bilan ishlash qoidalarini ko`ramiz.

Hali ofisga biriktirilmagan xizmatchini topish:

```
SELECT NAME FROM SALESREPS  
WHERE REP_OFFICE = NULL
```

SQL quyidagi satrni uchratganda:

```
REP_OFFICE = NULL
```

Quyidagi shartni tekshiradi:

```
NULL = NULL
```

Bunday tekshirish yana NULL qaytaradi! Qiymat tekshiruvchi operator uchun agar natija TRUE bo`lmasa, satr natijaviy to`plamga kirmaydi! Lekin bunday satrlar aslida mavjuddir! Bu holda NULL qiymatiga tekshirish to`g`ri operatorini qo`llash lozim:

```
----- ustunning nomi IS ----- NULL -----  
NOT
```

Qo`llaymiz: Ofisga biriktirilmagan xizmatchini toping.

```
SELECT NAME FROM SALESREPS  
WHERE REP_OFFICE IS NULL
```

NOT shartini qo'llash mumkin: Ofisga birlashtirilgan hamma xizmatchilarni toping.

```
SELECT NAME FROM SALESREPS  
WHERE REP_OFFICE IS NOT NULL
```

WHERE shartida qo'shma operatorlar

Izlashning "qo'shma" shartlarini ko'rib chiqamiz. WHERE operatorida OR, AND, NOT operatorlari bilan bog'langan bir necha izlash shartlarini qo'llash mumkin. Bu operatorlar sintaksisi quyidagicha:

NOT, OR, AND operatorlarning sintaksisi.

```
(----- WHERE ----- SHART -----)  
    (--- NOT ---)  
    (----- AND -----)  
    (----- OR -----)
```

Bu operatorlar yordamida yaratilgan bir necha so'rovlarni ko'rib chiqamiz.

Masalan: Sotuvlari haqiqiy hajmi rejadagidan yoki \$300.00 dan kam bo'lgan xizmatchilarni aniqlash

```
SELECT NAME, QUITA, SALES  
FROM SALESREPS  
WHERE SALES < QUITA IR SALES < 300.0
```

Sotuvlari haqiqiy hajmi rejadagidan va \$300.00 dan kam bo'lgan xizmatchilarni aniqlash

```
SELECT NAME, QUITA, SALES  
FROM SALESREPS  
WHERE SALES < QUITA AND SALES < 300.0
```

Sotuvchilari haqiqiy hajmi rejadagidan kam, lekin \$150.00 dan ko'p bo'lgan xizmatchilarni aniqlash

```
SELECT NAME, QUITA, SALES  
FROM SALESREPS  
WHERE (SALES < QUITA) AND (NOT SALES > 150.000)
```

AND ifodasi algebrasi.

Qiymat	Natija
-----	-----

**TRUE AND TRUE -> TRUE**  
**FALSE AND TRUE -> FALSE**  
**TRUE AND FALSE -> FALSE**  
**FALSE AND FALSE ->FALSE**  
**NULL AND TRUE -> NULL**  
**TRUE AND NULL -> NULL**  
**FALSE AND NULL ->FALSE**  
**NULL AND FALSE ->FALSE**  
**NULL AND NULL -> NULL**

OR ifodasi algebrasi.

Qiymat	Natija
-----	-----
<b>TRUE IR TRUE</b>	<b>-&gt; TRUE</b>
<b>FALSE IR TRUE</b>	<b>-&gt; TRUE</b>
<b>TRUE IR FALSE</b>	<b>-&gt; TRUE</b>
<b>FALSE IR FALSE</b>	<b>-&gt; FALSE</b>
<b>NULL IR TRUE</b>	<b>-&gt; TRUE</b>
<b>TRUE IR NULL</b>	<b>-&gt; TRUE</b>
<b>FALSE IR NULL</b>	<b>-&gt; NULL</b>
<b>NULL IR FALSE</b>	<b>-&gt; NULL</b>
<b>NULL IR NULL</b>	<b>-&gt; NULL</b>

NOT ifodasi algebrasi.

Qiymat	Natija
-----	-----
<b>NOT TRUE</b>	<b>-&gt; FALSE</b>
<b>NOT FALSE</b>	<b>-&gt;TRUE</b>
<b>NOT NULL</b>	<b>-&gt; NULL</b>

Qo`shimcha izlash operatorlarining har biri o`zustivorligiga ega. Eng yuqori ustivorlik NOT ga tegishli, undan so`ng AND va oxirida OR!

SQL92 standartida IS operatori yordamida ifoda rost, yolg'on yoki aniqlanmaganligini tekshirish mumkin. Uning sintaksisi quyidagicha:

***IS operatori sintaksisi***

**----- Solishtirish ----- IS (----- TRUE -----)**  
**(----- FALSE -----)**

**--- Mantiqiy ifoda --- (----- UNKNIWN -----)**

Masalan quyidagicha yozish mumkin:  $((SALES - QUITA) > 100.000)$  IS UNKNIWN . Bunday shart SALES yoki QUITA ustunlari NULL qiymatga ega satrlarni izlashga imkon beradi.

Oldin ko`rilgan so`rovlarda natijalar ixtiyoriy tartibda olingan edi. Agar o`quvchilar ro`yxatini alfavit tartibida yoki tovarlar narxini kamayish tartibida chiqarish zarur bo`lsachi? Buning uchun SELECT operatori tarkibida ORDER BY ifodasi ko`zda tutilgan. Uning sintaksisi:

**----- ORDER BY – ustun nomi ----- .**  
**-- ustun tartib raqami --- ----- ASC -----**  
**----- DESC -----**  
**----- , -----**

Avval quyidagi Misolni ko`ramiz: Har bir offis uchun sotuvlar haqiqiy hajmlarini regionlar nomlari, har bir regionda esa shaharlar nomlari bo`yicha alfavit tartibida ko`rsatish.

**SELECT CITY, REGION, SALES**  
**FROM OFFICES**  
**ORDER BY REGION, CITY**

ORDER BY ifodasidan keyin kelgan ustun ASOSIY kalitdir, undan keyingi ustunlar ikkinchi darajali kalitlardir. Yozuvlarni o`shish hamda kamayishi bo`yicha tartiblash mumkin.

Masalan: Sotuvlari haqiqiy hajmlari kamayish tartibida offislar ro`yxatini chiqarish.

**SELECT CITY, REGION, SALES**  
**FROM OFFICES**  
**ORDER BY SALES DESC**

Sotuvlar hajmlarini DESC predikatini qo`llab kamayish tartibida chiqaramiz. O`shish tartibida chiqarish uchun ASC predikati qo`llanadi. Bu predikat ko`zda tutilgan bo`lib , uni ko`rsatish sharrt emas. Agar ustun hisoblanuvchi bo`lib , nomga ega bo`lmasa uning tartib nomerini ko`rsatish mumkin!

Masalan, quyidagicha: Sotuvlar haqiqiy va rejadagi hajmlari ayirmasi kamayish tartibida ofislar ro`yxatini chiqaring.

**SELECT CITY, REGION, (SALES - TARGET)**  
**FROM OFFICES**

## **ORDER BY 3 DESC**

Shu kabi ORDER BY ifodasida ustunlar nomlari va nomerlari, hamda DESC, ASC predikatlarini qo`llab murakkab tartiblash shartlarini xosil qilish mumkin.

Masalan: Regionlar nomlari, har bir regionda sotuvlar haqiqiy va rejadagi hajmlari ayirmasi kamayish tartibida offislar ro`yxatini chiqaring.

```
SELECT CITY, REGION, (SALES - TARGET)
FROM OFFICES
ORDER BY REGION ASC, 3 DESC
```

### **Bir necha jadvallar bilan ishlash**

Jamlashtirish relyatsion ma'lumotlar bazasi operatsiyalaridan biri bo`lib , jadvallar orasidagi aloqani belgilaydi va ulardan ma'lumotni bitta komanda yordamida ajratishga imkon beradi. Har xil jadvallarda bir xil nomli ustunlar bo`lishi mumkin bo`lgani uchun, kerakli ustun uchun jadval nomi prefiksi ishlatiladi.

Jamlashda jadvallar FROM ifodasidan so`ng ro`yxat sifatida tasvirlanadi. So`rov predikati ixtiyoriy jadval ixtiyoriy ustuniga tegishli bo`lishi mumkin. Jamlash eng soddasi bu dekart ko`paytmasi, uni quyidagicha bajarish mumkin:

```
SELECT Custimers.*, Salepeople.*
FROM Salepeople, Custimers; *
```

Lekin bu erda hosil bo`lgan jadval keraksiz ma'lumotlarga ega. Keraksiz satrlarni olib tashlash uchun WHERE jumlasidan foydalaniladi.

Masalan: berilgan shahardagi sotuvchilar va buyurtmachilar ixtiyoriy kombinatsiyasini ko`rish uchun quyidagini kiritish lozim:

```
SELECT Custimers.CName, Salepeople.SName, Salepeople.City
FROM Salepeople, Custimers
WHERE Salepeople.City = Custimers.City;
```

Jamlashda SQL bir necha jadval satrlari kombinatsiyasini predikatlar bo`yicha solishtirishdir. Asosan ma'lumotlar ilovali yaxlitlik asosida tekshirilib, ajratib olinadi.

Misol: har bir sotuvchiga mos keluvchi buyurtmachilar ro`yxati:

```
SELECT Custimers.CName, Salepeople.SName
FROM Custimers, Salepeople
WHERE Salepeople.SNum = Custimers.SNum;
```

Tenglikka asoslangan predikatlardan foydalanuvchi jamlanmalar, tenglik bo`yicha jamlanma deb atalib, jamlanmalarning eng umuuiy ko`rinishidir. Shu bilan birga ixtiyoriy relyatsion operatoridan foydalanish mumkin.

Jamlashlar bir jadval satriga ikkinchi jadval satrlarini mos qo`yishga imkon beradi. Jamlashlar asosiy turi bu ichki jamlashdir. Jadvallarni ichki jamlash ikki jadval usutunlarini tenglashtirishga asoslangandir:

**SELECT *book, title, authir, name***

**FROM *authir, book***

**WHERE *book, authir = authir, id***

MySQLda jamlashning kuchliroq tipi ya'ni chap tashqi jamlash( yoki tashqi jamlash) dan foydalanishga imkon beradi.

Jamlash bu turitning ichki jamlashdan farqi shundaki natijaga o`ng jadvalda mos ustunga ega bo`lmagan chap jadval ustunlari qo`shiladi. Agar avtorlar va kitoblar misoliga e'tibor bersangiz natijaga ma'lumotlar bazasida kitoblarga ega bo`lmagan kitoblar kirmagan edi.

Ko`p xollarda o`ng jadvalda mos bo`lmagan chap jadvaldagi satrlarni chiqarish kerak bo`ladi. Buni tashqi jamlash yordamida amalga oshirish mumkin:

**SELECT *book.title, authir.name***

**FROM *authir***

**LEFT JOIN *book* IN *book.authir = authir.id***

E'tibor bering tashqi jamlanmada WHERE o`rniga IN kalit so'zi ishlatiladi.

MySQL tabiiy tashqi jamlashdan (*natural iuter joon*) foydalanishga imkon beradi. Tabiiy tashqi jamlash ikki jadval ikki ustuni bir xil nom va bir xil tipga ega bo`lgan hamda shu ustundagi qiymatlar teng bo`lgan satrlarni birlashtirishga imkon beradi:

**SELECT *my\_prid.name***

**FROM *my\_prid***

**NATURAL LEFT JOIN *their\_prid***

***Jadvallarni o'zi bilan jamlash.***

Jadvallarni o'zi bilan jamlash uchun har bir satrning o'zi yoki boshqa satrlar bilan kombinatsiyasoni xosil qilishingiz mumkin. So`ngra har bir satr predikat yordamida baxolanadi. Bu turdagi jamlash boshqa turdagi jamlashdan farq qilmaydi, farqi ikki jadval bir xildir. Jadvallarni jamlashda qaytariluvchi ustun nomlari oldiga jadval nomi qo'yiladi. Bu usutunlarga so`rovlarda murojaat qilish uchun har xil nomlarga ega bo`lishi kerak. Buning uchun vaqtinchalik nomlar ya'ni psevdonomlar qo`llanadi. Ular so`rov FROM jumlasida jadval nomidan so`ng bo`shliq qo'yib yoziladi.

Misol: bir xil reytingga ega hamma buyurtmachilar juftlarini topish.



```
SELECT a.CName, b.CName, a.Rating  
FROM Custimers a, Custimers b  
WHERE a.Rating = b.Rating;
```

Bu xolda SQL a va b jadvallarni jamlagandek ish tutadi. Yuqorida keltirilgan misolda ortiqcha satrlar mavjud, har bir kombinatsiya uchun ikkita qiymat. Birinchi psevdonomdagi A qiymat ikkinchi psevdonomdagi B qiymat bilan kombinatsiyasi olinadi, so`ngra ikkinchi psevdonomdagi A qiymat birinchi psevdonomdagi B qiymat bilan kombinatsiyasi olinadi.

Har gal satr o`zi bilan solishtiriladi. Buning oldini olish sodda usuli ikki qiymatga cheklanish kiritish, toki birinchi qiymat ikkinchisidan kichik bo`lsin yoki alfavit bo`yicha oldin kelsin. Bu predikatni asimmetrik qiladi, natijada xudi shu qiymatlar teskari tartibda olinmaydi.

Misol:

```
SELECT a.CName, b.CName, a.Rating  
FROM Custimers a, Custimers b  
WHERE a.Rating = b.Rating  
AND a.CName < b.CName;
```

Bu misolda agar birinchi kombinatsiya ikkinchi shartni qanoatlantirsa u chiqariladi, lekin teskari kombinatsiya bu shartni qanoatlantirmaydi va aksincha. Siz SELECT ifodasida yoki so`rovning FROM jumlasida keltirilgan har bir psevdonom yoki jadvalni ishlatishingiz shart emas. Siz har xil jadvallar, hamda bitta jadval har psevdonomlaridan iborat jamlanma yaratishingiz mumkin.

### **Sodda joylashtirilgan ostki so`rovlar.**

SQL yordamida so`rovlarni bir birining ichiga joylashtirishingiz mumkin. Odatda ichki so`rov qiymat xosil qiladi va bu qiymat tashqi predikat tomonidan tekshirilib, to`g`ri yoki noto`g`riligi tekshiriladi.

Misol: bizga sotuvchi nomi ma`lum: Motika, lekin biz SNum maydoni qiymatini bilmaymiz va buyurtmachilar jadvalidan hamma buyurtmalarni ajratib olmoqchimiz. Buni quyidagicha amalga oshirish mumkin:

```
SELECT * FROM ORDERS  
WHERE SNum =  
( SELECT SNum FROM Salepeople  
WHERE SName = 'Mitika' );
```

Avval ichki so`rov bajariladi, so`ngra uning natijasi tashqi so`rovni xosil qilish uchun ishlatiladi (SNum ostki so`rov natijasi bilan solishtiriladi).

Ostki so`rov bitta ustun tanlashi lozim, bu ustun qiymatlari tipi predikatda solishtiriladigan qiymat tipi bilan bir xil bo`lishi kerak. Siz ba`zi xollarda ostki so`rov bitta qiymat xosil qilishi uchun DISTINCT operatoridan foydalanishingiz mumkin.

Misol: Hoffman (CNum=21) ga xizmat ko`rsatuvchi sotuvchilar hamma buyurtmalarini topish lozim bo`lsin.

**SELECT \* FROM ORDERs**

**WHERE SNum = ( SELECT DISTINCT SNum FROM ORDERs  
WHERE CNum = 21 );**

Bu xolda ostki so`rov faqat bitta 11 qiymat chiqaradi, lekin umumiy xolda bir necha qiymatlar bo`lishi mumkin va ular ichidan DISTINCT faqat bittasini tanlaydi.

Ixtiyoriy sondagi satrlar uchun avtomatik ravishda bitta qiymat xosil qiluvchi funktsiya turi - agregat funktsiya bo`lib , undan ostki so`rovda foydalanish mumkin.

Masalan, siz summasi 4 oktyabrdagi bajarilishi lozim bo`lgan buyurtmalar summasi o`rta qiymatidan yuqori bo`lgan hamma buyurtmalarni ko`rmoqchisiz:

**SELECT \* FROM ORDERs**

**WHERE AMT >**

**( SELECT AVG (AMT) FROM ORDERs**

**WHERE Odate = '2014/10/04' );**

Shuni nazarda tutish kerakki guruhlangan agregat funktsiyalar GROUP BY ifodasi terminlarida aniqlangan agregat funktsiyalar bo`lsa ko`p qiymatlar xosil qilishi mumkin.

Agar ostki so`rov IN operatoridan foydalanilsa, Ixtiyoriy sondagi satrlar xosil qilish mumkin.

Misol: Toshkentdagi sotuvchilar uchun hamma buyurtmalarni ko`rsatish.

**SELECT \* FROM ORDERs**

**WHERE SNum IN**

**( SELECT SNum FROM Salepeople**

**WHERE City = 'Toshkent' );**

Bu natijani jamlanma orqali xosil qilish mumkin. Lekin odatda ostki so`rovli so`rovlar tezroq bajariladi. Siz ostki so`rov SELECT jumlasida ustunga asoslangan ifodadan foydalanishingiz mumkin. Bu relyatsion operatorlar yordamida yoki IN yordamida amalga oshirilishi mumkin. Siz ostki so`rovlarni HAVING ichida ishlatishingiz mumkin. Bu ostki so`rovlar agar ko`p qiymatlar qaytarmasa xususiy agregat funktsiyalaridan yoki GROUP BY yoki HAVING operatorlaridan foydalanishi mumkin.

Misol:

**SELECT Rating, COUNT (DISTINCT CNum) FROM Custimers**

**GROUP BY Rating**

**HAVING Rating >**

**( SELECT AVG (Rating) FROM Custimers**

**WHERE City = 'San Jise' );**

Bu komanda San Jise dagi baxolari o`rtachadan yuqori bo`lgan buyurtmachilarni aniqlaydi.

Korrellangan (mutanosib) joylashtirilgan ostki so`rovlar.

SQL tilida ostki so`rovlardan foydalanilganda tashqi so`rov FROM qismidagi ichki so`rovga mutanosib so`rov yordamida murojaat qilishingiz mumkin. Bu xolda ostki so`rov asosiy so`rov har bir satri uchun bir martadan bajariladi.

Misol: 3- oktyabrda buyurtma bergan hamma buyurtmachilarni toping.

```
SELECT * FROM Custimers a  
WHERE '2014/10/03' IN  
( SELECT Odate FROM ORDERs b  
WHERE a.CNum = b.CNum );
```

Bu misolda tashqi so`rovning Cnum maydoni o'zgargani uchun ichki so`rov tashqi so`rovning har bir satri uchun bajarilishi kerak. Ichki so`rov bajarilishini talab qiladigan tashqi so`rov satri joriy satr -nomzod deyiladi. Mutanosib ostki so`rov bilan bajariladigan baxolash protsedurasi quyidagicha:

1. Tashqi so`rovda nomlangan jadvaldan satrni tanlash. Bu kelajak satr -nomzodi.
2. Tashqi so`rov FROM jumlasida nomlangan psevdonomda bu satr – nomzod qiymatlarini saqlab qo`yish.
3. Ostki so`rovni bajarish. Tashqi so`rov uchun berilgan psevdonom topilgan hamma joyda joriy satr-nomzod qiymatidan foydalanish. Tashqi so`rov satr-nomzodlari qiymatlaridan foydalanish, tashqi ilova deyiladi.
4. Tashqi so`rov predikatini 3 qadamda bajariluvchi ostki so`rov natijalari asosida baxolash. U chiqarish uchun satr-nomzod tanlanishini belgilaydi.
5. Jadval keyingi satr-nomzodlari uchun protsedurani qaytarish va shu tarzda toki hamma jadval satrlari tekshirilib bo`lmaguncha.

Yuqoridagi misolda SQL quyidagi protsedurani amalga oshiradi:

1. U buyurtmachilar jadvalidan Hoffman satrini tanlaydi.
2. Bu satrni joriy satr-nomzod sifatida a – psevdonom bilan saqlaydi.
3. So`ngra ostki so`rovni bajaradi. Ostki so`rov CNum maydonning qiymati CNum qiymatiga teng satrlarni topish uchun buyurtmachilar jadvali hamma satrlarini ko`rib chiqadi. Xozir a.CNum qiymati 21 ga ya`ni Hoffman satrining CNum maydoni qiymatiga teng. Shundan so`ng shu satrlarning Odate maydonlari qiymatlari to`plamini xosil qiladi.
4. Shundan so`ng asosiy so`rov predikatida 3 oktyabrdagi qiymat shu to`plamga tegishliligini tekshiradi. Agar bu rost bo`lsa Hoffman satrini chiqarish uchun tanlaydi.
5. Shundan so`ng u butun protsedurani Goovanni satrini satr –nomzod sifatida foydalanib qaytaradi va saqlab qo`yadi, toki buyurtmachilar hamma satri tekshirilib bo`lmaguncha.

Ba`zida xatolarni topish uchun maxsus yaratilgan so`rovlardan foydalanish kerak bo`ladi.

Misol: Quyidagi so`rov buyurtmachilar jadvalini ko`rib chiqib SNum va CNum mos kelishini tekshiradi va mos bo`lmagan satrlarni chiqaradi.

```
SELECT * FROM ORDERs main  
WHERE NOT SNum =  
( SELECT SNum FROM Custimers  
WHERE CNum = main.CNum );
```

Asosiy so`rov asoslangan jadvalga asoslanuvchi mutanosib so`rovdan foydalanishingiz mumkin.

Misol: sotib olishlar buyurtmachilari uchun o`rta qiymatdan yuqori bo`lgan hamma buyurtmalarni topish.

```
SELECT * FROM ORDERs a  
WHERE AMT >  
( SELECT AVG (AMT) FROM ORDERs b  
WHERE b.CNum = a.CNum );
```

HAVING operatoridan ostki so`rovlarda foydalanilganidek mutanosib ostki so`rovlarda ham foydalanish mumkin.

HAVING ifodasida mutanosib ostki so`rovdan foydalanganda HAVING o`zida ishlatilishi mumkin bo`lgan pozitsiyalarga tashqi ilovalarni cheklab qo`yishingiz kerak. Chunki HAVING ifodasida faqat agregat SELECT ifodasida ko`rsatilgan funksiyalardan yoki GROUP BY ifodasida ko`rsatilgan maydonlardan foydalanish mumkin. Ulardan siz tashqi ilova sifatida foydalanishingiz mumkin. Buning sababi shuki HAVING tashqi so`rovdagi satrlar uchun emas guruhlar uchun baxolanadi. Shuning uchun ostki so`rov bir marta satr uchun emas guruh uchun bajariladi.

Misol: Buyurtmalar jadvalidagi sotib olishlar summalarini sanalar bo`yicha guruhlab summasini hisoblash kerak bo`lsin. Shu bilan birga summa maksimal summadan kamida 2000.00 ga ko`p bo`lmagan sanalarni chiqarib tashlash kerak bo`lsin:

```
SELECT Odate, SUM (AMT) FROM ORDERs a  
GROUP BY Odate  
HAVING SUM (AMT) >  
( SELECT 2000.00 + MAX (AMT) FROM ORDERs b  
WHERE a.Odate = b.Odate );
```

Ostki so`rov asosiy so`rovning ko`rilayotgan agregat guruhi sanasiga teng sanaga ega hamma satrlar uchun MAX qiymat hisoblaydi. Bu WHERE ifodasidan foydalanib bajarilishi lozim. Ostki so`rovning o`zi GROUP BY yoki HAVING operatorlarini ishlatmasligi kerak.

### ***EXISTS operatoridan foydalanish.***

EXISTS - bu "TRUE" yoki "FALSE" qaytaruvchi operatoridir. Bu shuni bildiradiki, u predikatda avtonom yoki mantiqiy operatorlar AND, OR, va NOT yordamida tuzilgan mantiqiy ifodalar bilan kombinatsiya qilingan xolda ishlatilishi mumkin. U ostki so`rovni "TRUE" deb baxolaydi agar u Ixtiyoriy natija xosil qilsa va "FALSE" deb baxolaydi hech qanday natija xosil qilmasa.

Misol: Agar buyurtmachilardan juda bo'lmasa bittasi San Jise shaxrida yashasa, buyurtmachilar jadvalidagi ma'lumotlarni chiqaring.

```
SELECT CNum, CName, City FROM Custimers  
WHERE EXISTS  
( SELECT * FROM Custimers  
WHERE City = 'San Jise' );
```

EXISTS ni faqat sodda ostki so'rov bilan emas mutanosib so'rov bilan ishlatish mumkin. Bu holda EXISTS ichki ostki so'rovni tashqining har bir satri uchun tekshiradi.

***ALL, ANY, SIME operatorlaridan foydalanish.***

ANY, ALL, va SIME ostki so'rovlarni argument sifatida qabul qiluvchi EXISTS operatorni eslatadi, lekin relyatsion operatorlar bilan birga ishlatilishi bilan farq qiladi. Bu tomondan ular ostki so'rovlarga qo'llaniluvchi IN operatorini eslatadi, lekin undan farqli faqat ostki so'rovlar bilan ishlashadi. SOME va ANY operatorlari o'zaro almashinuvchan.

Misol: bir shaharda joylashgan sotuvchilar bilan buyurtmachilarni topish uchun ANY operatoridan foydalanish.

```
SELECT * FROM Salepeople  
WHERE City = ANY ( SELECT City FROM Custimers );
```

Operator ANY ostki so'rov chiqargan hamma qiymatlarni oladi, (bu Misol uchun – Buyurtmachilar jadvalidagi hamma City qiymatlari), va rost deb baxolaydi agar ularning ixtiyoriysi (ANY) tashqi so'rov satridagi shahar qiymatiga teng bo'lsa. ANY operatori o'rniga ON yoki EXISTS ishlatish mumkin, lekin ANY “=” operatoridan boshqa relyatsion operatorlarni ishlatishi mumkin. Misol: hamma sotuvchilarni alfavit bo'yicha kelgan buyurtmachilari bilan birga topish.

```
SELECT * FROM Salepeople  
WHERE SName < ANY ( SELECT CName FROM Custimers );
```

ANY to'la bir qiymatli emas. Misol: Rimdagi buyurtmachilarga ko'ra yuqori reytinga ega buyurtmachilarni topish.

```
SELECT * FROM Custimers  
WHERE Rating > ANY ( SELECT Rating FROM Custimers WHERE  
City = 'Rome' );
```

Ingliz tilida ixtiyoriysidan katta " ( bu yerda City = Rime ) " baxolash quyidagicha talqin qilinadi, bu baxolash qiymati har bir City = Rime xoldagi baxolash qiymatidan katta bo'lishi kerak. SQL tilida ANY operatoridan foydalanilganda bunday emas. ANY to'g'ri deb baxolanadi agar ostki so'rov shartga mos keluvchi Ixtiyoriy qiymat topsa. Yuqorida ko'rilgan Misol 300 va 200 baxoli hamma buyurtmachilarni topadi, chunki 300>200 Rimdagi Givanni uchun va 200>100 Rimdagi Pereira uchun.

Soddaroq aytganda < ANY ifodasi eng katta tanlangan qiymatdan kichik qiymatni, > ANY - eng kichik tanlangan qiymatdan katta qiymatni bildiradi.

ALL yordamida, predikat rost hisoblanadi, ostki so`rov tanlagan har bir qiymat tashqi so`rov predikatidagi shartga mos kelsa.

Misol: Rimdagi har bir buyurtmachidan baxolari yuqori bo`lgan buyurtmachilarni chiqaring.

```
SELECT * FROM Custimers
```

```
WHERE Rating > ALL (SELECT Rating FROM Custimers WHERE City = 'Rime' );
```

Bu operator Rimdagi hamma buyurtmachilar baxolari qiymatlarini tekshiradi. Shundan so`ng Rimdagi hamma buyurtmachilardan baxosi yuqori bo`lgan buyurtmachilarni topadi. Rimda eng yuqori baxo - Giivanni (200). Demak 200 dan yuqori qiymatlar olinadi.

ANY operatori uchun bo`lgani kabi ALL operatori uchun ham IN va EXISTS yordamida alternativ konstruktsiyalar yaratish mumkin.

ALL asosan tengsizliklar bilan ishlatiladi, chunki qiymat "hammasi uchun teng" ostki so`rov natijasi bo`lishi mumkin agar hamma natijalar bir xil bo`lsa. SQL da <> ALL ifoda aslida ostki so`rov natijasining " hech qaysisiga teng emas " ma`noni bildiradi. Boshqacha qolib aytganda proedikat rost agar berilgan qiymat ostki so`rov natijalari orasida topilmagan bo`lsa. Agar oldingi Misolda tenglik tengsizlikka almashtirilsa, reytingi 300 ga teng bo`lgan hamma buyurtmachilar chiqariladi, chunki ostki so`rov 100 va 200 ga teng reytinglarni topgan.

ALL va ANY – orasidagi asosiy farq, ostki so`rov hech qanday natija qaytarmagan holatda ko`rinadi. Bu xolda ALL - avtomatik ("TRUE") ga teng, ANY bo`lsa avtomatik ("FALSE") ga teng.

Misol: Buyurtmachilar jadvali hammasoni chiqarish

```
SELECT * FROM Custimers
```

```
WHERE Rating > ALL ( SELECT Rating FROM Custimers WHERE City = 'Biston' );
```

Ko`rsatilgan operatorlar bilan ishlashda NULL qiymatlar ma`lum muammolarni keltirib chiqaradi. SQL predikatda solishtirayitgan qiymatlardan biri bo`sh (NULL) qiymat bo`lsa, natija noaniqdir. Noaniq predikat, noto`g`ri predikatga o`xshash, shuning uchun satr tashlab yuboriladi.

*UNION ifodasidan foydalanish.*

UNION ifodasi bir yoki bir necha SQL so`rovlar natijasoni birlashtirishga imkon beradi.

Misol: Londonda joylashgan hamma sotuvchilar va buyurtmachilarni bitta jadvalda chiqaring.

```
SELECT SNum, SName FROM Salepeople
```

```
WHERE City = 'London'
```

```
UNION
```

```
SELECT CNum, CName FROM Custimers
```

```
WHERE City = 'London';
```

Ikki yoki undan ortiq jadvallar jamlanganda ularning chiqish ustunlari jamlash uchun o'zaro muvofiq bo'lishi kerak. Bu shuni bildiradiki, har bir so'rov bir xil sondagi ustunlarni ko'rsatib, bu ustunlar mos tartibda kelishi va har biriga mos tiplarga ega bo'lishi kerak. Sonli maydonlar bir xil tipga va kattalikka ega bo'lishi kerak. Simvollar maydonlar bir xil sondagi simvollarga ega bo'lishi kerak. Moslik ta'minlovchi yana bir shart bo'sh (NULL) qiymatlar jamlanma ixtiyoriy ustunida man etilgan bo'lishi kerak. Bu qiymatlar boshqa jamlovchi so'rovlarda ham man etilgan bo'lishi kerak. Bundan tashqari siz ostki so'rovlarda UNION operatoridan, hamda jamlovchi so'rov SELECT operatorida agregat funktsiyalardan foydalanishingiz mumkin emas. Siz individual so'rovlardagi kabi natijani tartiblash uchun ORDER BY operatoridan foydalanishingiz mumkin. Jamlanma ustunlari chiqarish ustunlari bo'lgani uchun ular nomlarga ega bo'lmaydi, shuning uchun nomeriga qarab aniqlanishi lozim. Demak ORDER BY operatorida ustun nomeri ko'rsatolishi lozim. Foydali jamlanmalardan biri ikki so'rovni jamlashda ikkinchi so'rov birinchi so'rov chiqarib tashlagan satrlarni tanlashidir. Bu tashqi jamlanma deyiladi.

Misol: O'z shaharlarida buyurtmachilarga ega yoki ega emasligini ko'rsatgan xolda hamma sotuvchilarni chiqarish.

```
SELECT Salepeople.SNum, SName, CName, Cimm FROM Salepeople,  
Custimers WHERE Salepeople.City = Custimers.City
```

**UNION**

```
SELECT SNum, SName, 'NO MATCH', Cimm FROM Salepeople  
WHERE NOT City = ANY ( SELECT City FROM Custimers ) ORDER BY 2  
DESC;
```

Har gal bir necha so'rovlarni jamlaganda yumaloq qavslar yordamida baxolash mezonini ko'rsatishingiz mumkin. Ya'ni

```
query X UNION query Y UNION query Z;
```

o'rniga, yoki

```
( query X UNION query Y )UNION query Z;
```

yoki

```
query X UNION ( query Y UNION query Z );
```

ko'rsatishingiz mumkin. Chunki UNION bitta dublikatlarni yo'qotib boshqasoni qoldirishi mumkin. Quyidagi ikki ifoda

```
( query X UNION ALL query Y )UNION query Z;
```

```
query X UNION ALL( query Y UNION query Z );
```

bir xil natija qaytarishi shart emas, agar ikkilangan satrlar unda o'chirilgan bo'lsa.

**So'rovlarda guruhlash va funktsiyalar**

***Agregat funktsiyalar qo'llanishi***

Agregat (yoki STATIK) funktsiyalar, sonli yoki hisoblanuvchi ustunlar bilan ishlaydi. Agregat funktsiya argumenti butun ustun bo'lib, bitta qiymat qaytaradi.

Bu funksiyalarni ko`rib chiqamiz:

- SUM() – Ustundagi hamma qiymatlar summasini hisoblaydi.
- AVG() – Ustundagi hamma qiymatlar o`rtasi qiymatini hisoblaydi.
- MIN() – Ustundagi hamma qiymatlar eng kichigini aniqlaydi.
- MAX() – Ustundagi hamma qiymatlar eng kattasini aniqlaydi.
- COUNT() – Ustundagi qiymatlar sonini hisoblaydi.
- COUNT(\*) – So`rov natijalari jadvalidagi satrlar sonini hisoblaydi.

Komandalar sintaksisi ko`rinishi:

```
-- SUM ----- ( --- ifoda ----- ) -----  
-- -- DISTINCT – ustun nomi ---  
-- AVG ----- ( --- ifoda ----- ) -----  
-- -- DISTINCT -- ustun nomi ---  
-- MIN ----- ( ifoda ) -----  
-- MAX ----- ( ifoda ) -----  
-- COUNT ----- ( ----- ustun nomi) -----  
-- -- DISTINCT --  
-- COUNT(*) -----
```

Agregatlash argumenti bo`lib ustun nomidan tashqari ixtiyoriy matematik ifoda xizmat qilishi mumkin. Misol uchun quyidagi so`rovda: Sizni kompaniyangizda reja bajara olishi o`rtacha prosentni qancha?

```
SELECT AVG(100 * (SALES/QUITA))  
FROM SALESREPS
```

Yana bir shakl: Sizni kompaniyangizda reja bajara olishi o`rtacha prosentni qancha?

```
SELECT AVG(100 * (SALES/QUITA)) PRICENT  
FROM SALESREPS
```

Bu xolda ustun nomi ma'noliroq, lekin bu asosiysi emas. Ustunlar summasoni hisoblab ko`ramiz. SUM() funksiyasini qo`llaymiz, ustun sonli bo`lishi kerak! Masalan, quyidagicha: Kompaniya xizmatchilari sotuvlar hajmi rejadagi va haqiqiy o`rta qiymati qanchaga teng?

```
SELECT SUM(QUITA), SUM(SALES)  
FROM SALESREPS
```

AVG() agregatlash funksiyasiga yana bir necha sodda Misollarni ko`ramiz. Masalan: "ACI" ishlab chiqaruvchi mollari o`rtacha narxini hisoblang.

```
SELECT AVG(PRICE)  
FROM PRIDUCTS  
WHERE MFR_ID = 'ACO'
```



Ekstremumlarni topish funktsiyalari yani MIN(), MAX() funktsiyalarini ko`ramiz. Bu funktsiyalar sonli ustunlar, sanalar va satrli o`zgaruvchilar bilan ishlaydi. Eng sodda qo`llanishi sonlar bilan ishlash.

Masalan quyidagi so`rov beramiz: Eng ko`p va kam sotuvlar rejadagi hajmi?

```
SELECT MIN(QUITA), MAX(QUITA)  
FROM SALESREPS
```

Bu sonlarni o`zichiga olgan ustunlardir. Yana bir so`rov beramiz: Bazadagi buyurtmalarning ichida eng oldin berilgan so`rov sanasi?

```
SELECT MIN(ORDER_DATE)  
FROM ORDERS
```

Satrlar bilan ishlaganda har xil SQL serverlardagi kodirovkalar har xil natija berishi mumkin. Yozuvlar sonini sanash uchun COUNT() qo`llanadi. Bu funktsiya son qiymat qaytaradi

Masalan: Kompaniyamiz mijozlari soni nechta?

```
SELECT COUNT(CUST_NUM)  
FROM CUSTIMERS
```

Yana bir so`rov: Qancha xizmatchi rejani ortig`i bilan bajardi?

```
SELECT COUNT(NAME)  
FROM SALESREPS  
WHERE SALES > QUITA
```

COUNT(\*) funktsiyasi qiymatlar sonini emas, satrlar sonini hisoblaydi. Quyidagicha yozish mumkin:

```
SELECT COUNT(*)  
FROM ORDERS  
WHERE AMOUNT > 250
```

*NULL qiymat va agregat funktsiyalar*

Ustun qiymati NULL bo`lsa AVG(), MIN(), MAX(), SUM(), COUNT() funktsiyalari qanday qiymat qaytaradi? ANSI/ISI qoidalariga ko`ra "agregat funktsiyalar NULL qiymatni e`tiborga olmaydi"! Quyidagi so`rov ko`ramiz:

```
SELECT COUNT(*), COUNT(SALES), COUNT(QUITA)  
FROM SALESREPS
```

Jadval bitta lekin so`rovdagi qiymatlar har xil. Chunki QUITA maydoni-NULL qiymatni o`zichiga oladi. COUNT funktsiyasi COUNT(maydon)

ko`rinishda bo`lsa NULL qiymatni e`tiborga olmaydi, COUNT(\*) bo`lsa satrlar umumiy sonini xisoblaydi. MIN(), MAX() funktsiyalari ham NULL qiymatni e`tiborga olmaydi, lekin AVG(), SUM() - NULL qiymat mavjud bo`lsa chalkashtiradi. Masalan, quyidagi so`rov:

```
SELECT SUM(SALES), SUM(QUITA), (SUM(SALES) - SUM(QUITA)),  
(SUM(SALES - QUITA))  
FROM SALESREPS
```

(SUM(SALES)-SUM(QUITA)) va (SUM(SALES-QUITA)) ifodalari agar QUITA, maydoni NULL qiymatga ega bo`lsa har xil qiymat qaytaradi. Ya`ni ifoda SUM(ustun qiymati - NULL) Yana NULL qaytaradi!

Shunday qilib:

1. Agar ustundagi qiymatlardan biri NULL ga teng bo`lsa, funktsiya natijasini hisoblashda ular tashlab yuboriladi!

2. Agar ustundagi hamma qiymatlar NULL ga teng bo`lsa, AVG(), SUM(), MIN(), MAX() funktsiyalari NULL qaytaradi! Funktsiya COUNT() nol qaytaradi!

3. Agar ustunda qiymatlar bo`lmasa (Ya`ni ustun bo`sh), AVG(), SUM(), MIN(), MAX() funktsiyalari NULL qaytaradi! Funktsiya COUNT()nol qaytaradi!

4. Funktsiya COUNT(\*) satrlar sonini hisoblaydi va ustunda NULL qiymat bor yo`qligiga bog`lik emas! Agar ustunda satrlar bo`lmasa, bu funktsiya nol qaytaradi!

DISTINCT funktsiyasoni agregat funktsiyalar bilan birga ishlatish mumkin. Masalan quyidagi so`rovlarda:

1. Kompaniyamizda qancha har xil raportlar nomlari mavjud?

```
SELECT COUNT(DISTINCT TITLE)  
FROM SALESREPS
```

DISTINCT va agregatlar ishlashda quyidagi qoidalar mavjud. Agar siz DISTINCT va agregat funktsiyani ishlatsangiz uning argumenti faqat ustun nomi bo`lishi mumkin, ifoda argument bo`lolmaydi. MIN(), MAX() funktsiyalarida DISTINCT ishlatish ma`nosi yo`q! COUNT() funktsiyasida DISTINCT ishlatiladi, lekin kam xollarda. COUNT(\*) funktsiyasiga umuman DISTINCT qo`llab bo`lmaydi, chunki u satrlar sonini hisoblaydi! Bitta so`rovda DISTINCT faqat bir marta qo`llanishi mumkin! Agarda u agregat funktsiya argumenti sifatida qo`llanilsa, boshqa argument bilan qo`llash mumkin emas!

*Agregatlar va ma`lumotlarni guruhlash*

Agregat funktsiyalar jadval uchun natijaviy satr xosil qiladi. Masalan: Buyurtma o`rtacha narxi qancha?

```
SELECT AVG(AMOUNT)  
FROM ORDERS
```

Masalan, oralik natijani topish lozim bo`lsin. Bu holda guruhlanishli so`rov yordam beradi. Ya'ni SELECT operatorining GROUP BY ifodasi. Avval GROUP BY ifodasi qatnashgan quyidagi so`rovni ko`ramiz: Har bir xizmatchi uchun buyurtma o`rtacha narxi qancha?

```
SELECT REP, AVG(AMOUNT)
FROM ORDERS
GROUP BY REP
```

REP maydoni bu xolda guruhlash maydonidir, Ya'ni REP maydonning hamma qiymatlari guruhlarga ajratiladi va har bir guruh uchun AVG(AMOUNT) ifodasi hisoblanadi! Ya'ni quyidagilar bajariladi:

1. So`rovlar har bir xizmatchiga bittadan guruhga ajratiladi. Har bir guruhda REP maydoni bir xil qiymatga ega.

2. Har bir guruh uchun guruhga kiruvchi hamma satrlar bo`yicha AMOUNT ustuni o`rta qiymati hisoblanadi va bita natijaviy satr xosil qilinadi. Bu qator guruh uchun REP ustuni qiymati vash u guruh uchun so`rov o`rta qiymatini o`zichiga oladi.

Shunday qilib, GROUP BY ifodasi qo`llanilgan so`rov, "GURUHLANISHLI SO`ROV " deb ataladi! Shu ifodadan keyin kelgan ustun "guruhlash ustuni " deyiladi. Yana bir necha guruhlanishli so`rovlarni ko`rib chiqamiz.

Har bir offis uchun sotuvlarning rejalashtirilgan hajmi diapazoni qancha?

```
SELECT REP_OFFICE, MIN(QUITA), MAX(QUITA)
FROM SALESREPS
GROUP BY REP_OFFICE
```

Yana bir so`rov: Har bir offisda qancha xizmatchi ishlaydi?

```
SELECT REP_OFFICE, COUNT(*)
FROM SALESREPS
GROUP BY REP_OFFICE
```

Yana bir guruhlanishli qiziqarli so`rov: Har bir xizmatchi nechta mijozga xizmat ko`rsatadi?

```
SELECT COUNT(DISTINCT CUST_NUM), 'CUSTIMERS FOR
SALESREPS', CUST_REP
FROM CUSTIMERS
GROUP BY CUST_REP
```

Bu erda 'CUSTIMERS FIR SALESREPS' psevdodaydonning ishlatolishiga e'tibor bering. So`rov natijalarini bir nechta ustun bo`yicha guruhlash mumkin. Masalan, quyidagicha:

Har bir xizmatchi uchun har bir klient bo`yicha buyurtmalar umumiy sonini hisoblash.

```
SELECT REP, CUST, SUM(AMOUNT)
```

**FROM ORDERS**  
**GROUP BY REP, CUST**

Lekin ikki ustun bo`yicha guruhlashda natijalar ikki darajasiga ega guruhlar va ostki guruhlar yaratish mumkin emas. Lekin tartiblashni qo`llash mumkin. Shu bilan birga GROUP BY ishlatilganda so`rov natijalari avtomatik tartiblanadi. Quyidagi so`rovni ko`ramiz:

Har bir xizmatchi uchun har bir klient bo`yicha buyurtmalar umumiy sonini hisoblash; so`rov natijalarini klientlar va xizmatchilar bo`yicha tartiblash.

**SELECT REP, CUST, SUM(AMOUNT)**  
**FROM ORDERS**  
**GROUP BY REP, CUST**  
**ORDER BY REP, CUST**

Shunday qolib GROUP BY ifodasi SELECT ni guruhlarni qayta ishlashga majbur qiladi.

MS SQL serverida CIMPUTE ifodasi mavjud bo`lib relyatsion so`rovlar asoslariga zid keladi. Lekin uning yordamida saqlanuvchi protseduralardan foydalanmasdan shunga o`xshash natijalarni olish mumkin. Guruhlanishli so`rovlar uchun chegaralar mavjud. Satrlarni hisoblanuvchi ifoda asosida guruhlash mumkin emas. Qaytarila yotgan qiymatlar elementlariga ham chegaralar mavjud. Qaytariluvchi ustun bo`lishi mumkin:

1. Konstantalar.
2. Guruhga kirgan hamma satrlar uchun bitta qiymat qaytaruvchi agregat funktsiya.
3. Guruh hamma satrlarida bir xil qiymatga ega guruhlash ustuni.
4. Ko`rsatilgan elementlarni o`zi ichiga oluvchi ifoda.

Odatda guruhlanishli so`rovlar qaytaruvchi ustunlarga guruhlash ustuni va agregat funktsiya kiradi. Agar agregat ko`rsatilmasa GROUP BY dan foydalanmasdan DISTINCT ifodasidan foydalanish yetarli. Agar so`rovga guruhlash ustuni qo`shilmasa, u yoki bu satr qaysi guruhga tegishligini aniqlash mumkin emas. Shu kabi SQL92 guruhlanishli so`rovlarni taxlil qilishda birlamchi va ikkilamchi kalitlar haqidagi ma'lumot ishlatilmaydi.

Har bir xizmatchi uchun buyurtmalar umumiy sonini hisoblash.

**SELECT EMPL\_NUM, NAME, SUM(AMOUNT)**  
**FROM ORDERS, SALESREPS**  
**WHERE REP = EMPL\_NUM**  
**GROUP BY EMPL\_NUM, NAME**

Yana soddaroq shakl:

Har bir xizmatchi uchun buyurtmalar umumiy sonini hisoblash.

**SELECT NAME, SUM(AMOUNT)**  
**FROM ORDERS, SALESREPS**

**WHERE REP = EMPL\_NUM**

**GROUP BY NAME**

Agar guruhlash maydonlaridan birida NULL qiymat mavjud bo`lsa qaysi guruhga tegishli bo`ladi? WHERE ifodasida NULL va NULL tenglikka solishtirish natijasi yana NULL beradi. Shuning uchun ANSI/ISI standartida GROUP BY ifodasida NULL qiymatlar teng deb qabul qilingan.

***Guruhlash va HAVING yordamida ajratish***

Shart bo`yicha satrlarni ajratish uchun WHERE ifodasidan foydalangan edik. Shart bo`yicha guruhlarni ajratish uchun HAVING operatori mavjuddir. Uning sintaksisi WHERE operatori bilan bir xil va ulardan birgalikda foydalanish mumkin. Quyidagi so`rovni ko`ramiz:

Buyurtmalar umumiy narxi \$300 dan ortiq xizmatchilar uchun buyurtma o`rtacha narxi qanchaga teng?

**SELECT REP, AVG(AMOUNT)**

**FROM ORDERS**

**GROUP BY REP**

**HAVING SUM(AMOUNT) > 300**

Ko`rinib turibdiki HAVING SUM(AMOUNT) > 300 ifodasi satrlarni guruhlash sharti sifatida kelmoqda.

Agar SUM(AMOUNT) > 300 sharti yolg'on bo`lsa, bu guruh natijaviy to`plamdan chiqariladi. Agar rost bo`lsa guruh natijaviy to`plamga kiradi!

Yana bir Misol ko`raylik: Ikki va undan ortiq xizmatchiga ega har bir ofisning hamma xizmatchilari uchun rejadagi va haqiqiy sotuvlar umumiy hajmini hisoblash.

**SELECT CITY, SUM(QUITA), SUM(SALESREPS.SALES)**

**FROM OFFICES, SALESREPS**

**WHERE OFFICE = REP\_OFFICE**

**GROUP BY CITY**

**HAVING COUNT(\*) >= 2**

Bu Misolda WHERE va HAVING ifodalari o`z funktsiyalarini bajaradilar. Shunga e`tibor berish kerakki HAVING ifodasida agregat funktsiyalardan foydalaniladi, So`rov bajarolishini ko`ramiz:

1. OFFICES va SALESREPS jadvallari xizmatchi yashaydigan shaharni topish uchun qo`shiladilar.
2. Qo`shilgan jadval satrlarlari ofislar bo`yicha guruhlanadilar.
3. Ikkidan kam satrga ega guruhlar tashlab yuboriladi. Ular HAVING ifodasi talabiga javob bermaydilar.
4. Har bir guruh uchun haqiqiy va rejadagi sotuvlar hajmlari hisoblanadi.

Murakkabroq Misolni ko`ramiz:

Har bir tovar nomi uchun narxi, ombordagi soni va buyurtma berilganlar umumiy sonini ko`rsating, agar uning uchun buyurtma berilganlar umumiy soni ombordagi umumiy soni 75 foizidan ko`p bo`lsa.

```
SELECT DESCRIPTION, PRICE, QTY_IN_HAND, SUM(QTY)  
FROM PRIDUCTS, ORDERS  
WHERE MFR = MFR_ID  
GROUP BY MFR_ID, PRIDUCT_ID, DESCRIPTION, PRICE,  
QTY_IN_HAND  
HAVING SUM(QTY) > (0.75 * QTY_IN_HAND)  
ORDER BY QTY_IN_HAND DESC
```

HAVING uchun qo`shimcha chegaralar mavjuddir. Bu ifoda juda bo`lmasa bitta agregat funktsiyani o`zichiga olishi kerak. Chunki WHERE aloxida satrlarga HAVING satrlar guruhlariga qo`llanadi. NULL qiymat uchun WHERE ifodasiga o`xshab quyidagi qoida o`rinli. Agar izlash sharti NULL qiymatga ega bo`lsa satrlar guruhi tashlab yuboriladi. HAVING ifodasoni GROUP BY siz qo`llash mumkin. Bu xolda natija hamma satrlardan iborat guruh deb qaraladi, lekin amalda bu kam qo`llanadi.

## **CGI DAN FOYDALANIB DASTURLASH**

CGI haqida tushuncha. CGI — bu serverdagi dasturlar web-server orqali klentlarga uzatishni belgilovchi qoidalar to`plamidir. CGI Spetsifikatsiya NTML va NTTR larga yangi xarakteristikalar, yangi formalar kiritish olishiga olib keldi. Agar CGI dasturlarga klentlar ma`lumotlar yubirilsa, forma klentga dasturlar uchun ma`lumot yuborishga imkon beradi. Endi foydalanuvchi Joriy vaqtni ko`rib qilmasdan o`zgartirishi ham mumkin! CGI formalari Web dunyosida chinakam interaktivlik uchun eshik ochdi. Keng tarqalgan CGI amaliy dasturlarni o`z ichiga oladi. Dinamik HTML. Butun bir saytlar bitta CGI-p dastur tomonidan generatsiya qilishi mumkin. Foydalanuvchi bergan so`zlar asosida hujjatlarni tipishga imkon beruvchi qidiruv mexanizmlari. Mehmonlar kitoblari va foydalanuvchilar o`z ma`lumotlarini qildirishi mumkin bo`lgan e`lonlar taxtalari. Byurtmalar blankalari. Anketalar. Serverda joylashtirilgan ma`lumotlar bazasidan kerakli ma`lumotni ajratib olish. CGI to`rtta usul yordamida CGI-dastur va web-server orasida hamda shunga ko`ra Web klient orasida ma`lumotlar almashadi: Atrof muhit o`zgaruvchilari. Komanda satri. Kiritish standart qurilmasi.

### ✓ Chiqarish standart qurilmasi.

Bu to`rtta usul yordamida server klent uzatgan hamma ma`lumotlarni, CGI-dasturga uzatadi. Keyin CGI-dastur o`zishini bajarib, chiqish ma`lumotlarini serverga, u bo`lsa klentga yuboradi.

Server CGI-dasturni bajarganda, avvalambor unga atrofmuhit o`zgaruvchilari shaklida ishlash uchun ba`zi ma`lumotlarni yuboradi. Spetsifikatsiyada rasmiy ravishda o`n yettita o`zgaruvchilar ta`riflangan, lekin norasmiy ravishda quyidagi HTTP (nec) zams/n mexanizmi yordamida programma ko`proq ishlatiladi.

CGI-dastur bu o`zgaruvchilarga, komanda satirini ishga tushirishga komanda protsessorining ixtiyoriy o`zgaruvchisiga murajaat qilgani kabi murajaat qiladi. Komanda protsessori ssenariyasida, masalan, atrof-muhit o`zgaruvchisi FIOga \*FIO sifatida murajaat qilishi mumkin. Bu o`zgaruvchilar CGI-dasturi tomonidan foydalanishi va hatto o`zgartira olishi mumkin. Lekin bu o`zgartirishlar dasturni ishga tushirgan web-serverga ta`sir qilmaydi.

CGI argumentlari CGI-dasturga komanda satri satrlar shaklida uzatadi. Bu usul amaliyotda kam qo`llanadi, shuning uchun biz unga batafsil to`xtalib o`tirmaymiz. Bu usul mazmuni shundan iboratki, o`zgaruvchi birma-bir doiraga tushib qolsa yani buni quyidagi ko`rinishda yozib olamiz QUERY STRING “=” simbolni o`z ichiga olmasa, CGI-dastur QUERY STRING dan qilingan komanda satri parametrlari bilan bajariladi. Masalan, `http: // www. myserver. com/cgi-bin/finger? root satr finger root ni www. myserver. com da ishga tushiradi.`

### **Chiqarish standart qurilmasi.**

CGI-dastur tomonidan chiqarish standart qurilmasiga yuboriladigan ma`lumot web-server tomonidan o`qiladi va klentga yuboriladi.

Agar senariy nomi *nph* dan boshlansa bu ma`lumotlar to`g`ri klientga web-server aralashmasdan yuboriladi. Bu holda CGI-dastur klientga tushunarli bo`lgan to`g`ri HTTP sarlavha hosil qilishi kerak. Aks holda HTTP-sarlavha yaratishni web serverga qo`yib bering.

Agar siz *nph*-ssenariydan foydalanmasangiz, serverga sizni natijangiz haqida ma`lumot beruvchi yana bitta direktiva berish kerak. Odatda bu NTTR-sarlavha Content-Ture, lekin Location sarlavha bo`lishi ham mumkin. Sarlavhadan so`ng bo`sh satr, ya`ni satrni o`tkazish yoki CR/LF kombinatsiyasi kelishi kerak.

Sarlavha Content-Type serverga, CGI-p dasturingiz qanday turdagi ma`lumot hosil qilgani to`g`risida xabar beradi. Agar bu HTML sahifa bo`lsa, satr quyidagicha bo`lishi kerak `ContentType: tey/html.` Sarlavha Location serverga klentni yuborish kerak bo`lgan boshqa URL — yoki shu serverdagi boshqa yo`lni

ko`rsatadi Sarlavha quyidagi ko`rinishga ega bo`lishi kerak:

`http://www. myserver. cim/aNOther/place/.`

HTTP sarlavhalari va bo`sh satrdan so`ng dasturingiz hosil

qilgan ma'lumotlarni yuborish mumkin — HTML sahifa, tasvir, matn yoki yana boshqa ma'lumot. Apache serveri bilan birga o'rnatilishi mumkin bo'lgan CGI-dasturlar ichida nph-test-cgiva test-cgi mavjud bo'lib, ular nph va nph bo'lmagan usullardagi sarlavhalar orasidagi farqni yaxshi ko'rsatadi.

Holatni eslab qolish.

Holatni eslab qilish faqat jinoyatchilar bilan kurashish uchun emas, foydalanuvchilaringizga yaxshi xizmat qilish uchun ham zarurdir. Muammoning kelib chiqish sababi shundaki, HTTP bu «xotirasiz» protokoldir. Bu shuni bildiradiki, klient serverga ma'lumotlar yuboradi, server ma'lumotlarni klientlarga qaytaradi va har biri o'z yo'lida davom etadi. Server klient haqida keyingi amallarda kerak bo'ladigan ma'lumotlarni saqlab qolmaydi. Klient hamma odatda bajarilgan amallar haqidagi ma'lumotni keyinchalik foydalanish uchun saqlab qolmaydi. Bu World Wide Web dan foydalanishga sezilarli chegara qo'yadi.

Bunday protokolda CGI ssenariylarini yaratish, suhbatni eslab qololmaslikka o'xshagandir. Har gal kim bilandir gaplashganda, oldin qancha suhbatlashganingizga qaramasdan, qaytadan tanishishga va suhbat mazmunini izlashga to'g'ri keladi. Netscape Navigator payti bo'lishi bilan klient qismda cookies deb atalgan shoshilinch yaratilgan usul qo'llana boshladi.

Bu usul klient va server orasida u yoki bu tomonga uzatish mumkin bo'lgan, Content-Type va Location sarlavhalariga o'xshash Yangi HTTP-sarlavha yaratishdan iborat. Klient brouzeri cookie sarlavha olgandan so'ng, cookie da ma'lumotlar va cookie ta'sir o'tkazadigan domen nomini saqlab qo'yishi

kerak. Shundan so'ng shu domenga tegishli URLga tashrif buyurilganda cookie sarlavha CGI-dasturlarda ishlata olishi uchun serverga qaytarolishi kerak.

Bu usul, ya'ni cookie asosan foydalanuvchi identifikatorini saqlash uchun ishlatiladi. Foydalanuvchi haqidagi ma'lumot server mashinasida saqlanishi mumkin. Bu foydalanuvchining unikal ID si cookie sifatida brouzerga yuborilishi mumkin, shundan so'ng foydalanuvchi har gal saytga tashrif buyurganda, brouzer avtomatik ravishda serverga shu ID ni yuboradi. Server

ID ni CGI dasturga uzatadi, u bo'lsa bu faylni ochadi va foydalanuvchi haqidagi hamma ma'lumotlardan foydalanish imkoniga ega bo'ladi. Bu hammasi foydalanuvchi bilmagan holda yuz beradi.

Bu usul foydaligiga qaramasdan, ko'pgina katta saytlar uni Holatni eslab qolishning yagona usuli sifatida ishlatmaydilar.

Buning bir necha sabablari bir. Avvalam bor hamma brouzerlar ham cookie ni qo'llamaydi. Yaqin paytgacha ko'zi ojiz foydalanuvchilar uchun asosiy brouzer — Linux — cookie ni qo'llamas edi.



«Rasmiy» u hali ham qoʻllanmaydi, lekin keng tarqalgan baʼzi «yon shoxlari» qoʻllanadi. Ikkinchidan cookie foydalanuvchini maʼlum mashinaga bogʻlab qoʻyadi. Web ning eng katta ustunligi shundaki, unga dunyoning ixtiyoriy nuqtasidan murojaat qilish mumkin. Sizni web-sahifangiz qayerda yaratilgani va saqlanayotganidan qatʼiy nazar uni ixtiyoriy Internetga ulangan mashinada koʻrsatish mumkin. Agar siz cookie ni qoʻllovchi saytga begona mashinadan murojaat qilmoqchi boʻlsangiz, sizni cookie yordamida qoʻllangan shaxsiy maʼlumotlaringiz yoʻqoladi.

Koʻpgina saytlar hali ham cookie dan foydalanuvchi sahifalarini persinallashtirish uchun foydalanadi, lekin uni anʼanaviy «registratsiya nomi/paril» uslubidagi interfeys bilan toʻldiradi. Agar saytga cookie ni qoʻllamaydigan brouzer orqali murojlat qilinsa, sahifa birinchi koʻrinishda oʻrnatiladigan registratsiya nomi va parol kiritish firmasiga ega boʻladi. Foydalanuvchilarni choʻchitmaslik uchun bu forma sodda va kichik boʻladi. Foydalanuvchi formaga registratsiya nomi va parol kiritgandan soʻng, CGI nom xuddi cookie dan yuborilgandek foydalanuvchi haqidagi maʼlumotlarni saqlovchi faylni topadi. Bu usuldan foydalanib foydalanuvchi dunyoning ixtiyoriy nuqtasida persinallashtirgan web-saytda registratsiyadan oʻtishi mumkin.

CGI va maʼlumotlar bazalari.

Internet davri boshlangandan beri maʼlumotlar bazalari World Wide Web bilan uzviy bogʻlangandir. Amaliy jihatdan koʻpchilik Web ni ulkan multimediya maʼlumotlari bazasi deb qaraydi.

Qidiruv mashinalari maʼlumotlar bazalari ustunligini har kuni koʻrsatadi. Qidiruv mashinasi siz soʻraganda butun Internet boʻyicha kalit soʻzlarni qidirib oʻtirmaydi. Buning oʻrniga sayt yaratuvchilari boshqa dasturlar yordamida ulkan koʻrsatkich yaratadilar. Bu koʻrsatkich qidiruv mexanizmi yozuvlarni ajratib

oladigan maʼlumotlar bazasi boʻlib xizmat qiladi. Maʼlumotlar bazalari maʼlumotni ixtiyoriy murojaatda tez tanlashga imkon beradigan shaklda saqlaydi.

Oʻzgaruvchanligi tufayli maʼlumotlar bazalari Web imkonini yanada kengaytiradi: universal interfeysga aylantiradi. Masalan, tizimli administratirlashni administratirni kerakli tizimda registratsiya qilish talabi oʻrniga masofadan web-interfeys orqali bajarishga imkon beradi. Maʼlumotlar bazasini Web ga ulash Internet interaktivligi Yangi darajasi asosida yotadi.

Maʼlumotlar bazalarining Web ga ulanish natijasi: dunyodagi axborotning koʻp qismi maʼlumotlar bazalarida saqlanmoqda.

Web paydo boʻlmasdan mavjud boʻlgan maʼlumotlar bazalari merosga olingan (legacy) maʼlumotlar bazalari deyiladi (Oxirgi paytda yaratilgan va Web ga ulanmagan maʼlumotlar bazalariga qarama-qarshi). Koʻpgina korporatsiyalar (va hatto ayrim shaxslar) bu meros qilgan maʼlumotlar bazalariga Web orqali murojaatni taʼminlashadi. Agar sizni meros qilgan maʼlumotlar bazangiz MySQL yoki mSQL boʻlmasa, bu mavzu qaralmaydi.

Yuqorida aytilganidek, ma'lumotlar bazalari va Web orasidagi bog'lanishlar imkoniyatlari cheksizdir. Hozirgi davrda Webdan murojaat qilish imkoniga ega bo'lgan minglab unikal va foydali ma'lumotlar bazalari mavjud. Bu amaliy dasturlardan tashqari har xil turdagi ma'lumotlar bazalari mavjuddir. Ba'zilar

MySQL va mSQL ma'lumotlar bazalari bilan interfeys sifatida CGI-dasturlardan foydalanadilar. Bular biz uchun eng qiziqarlidir. Boshqalari kommertsial Amaliy dasturlardan keng tarqalgan kichik Microsoft Access i Claris FileMaker Pri kabi ma'lumotlar bazalari bilan bog'lanish uchun foydalanadilar. Yana boshqalari eng sodda bo'lgan matnli fayllar bilan ishlaydilar.

### **Berilganlar bazasini yaratish texnologiyalari.**

SQL tilining berilish namunalarida berilganlar bazasini yaratish operatorlari berilmagan, har bir ko'rinish o'z usullariga ega. Aytaylik ACCESS BBT da har bir berilganlar bazasi alohida .mdb kengaytmali fayldan tashkil topadi. MS SQL SERVERda ham .mdf kengaytmali fayl bo'ladi, undan tashqari, bu BBT **CREATE DATABASE** operatori yordamida berilganlar bazasini yaratish mumkin, masalan:

```
CREATE DATABASE Talaba S ON
    (NAME=talaba_dat
    FILENAME='C:\ST\ StbaseSQL.mdf'
    SIZE=10
    MAXSIZE=90
    FILEGROWTH=5)
LOG ON
    (NAME=TALABA_log
    FILENAME='C:\ST\ StbaseSQL.idf'
    SIZE=5
    MAXSIZE=25
    FILEGROWTH=5)
```

Berilganlar bazasining konseptual modelini tuzish uchun chizmalar ishlatiladi. Chizmalar **CREATE SCHEMA** operatori orqali beriladi, uning sintaksisi quyidagicha:

<chizma yaratish operatori>::= **CREATE SCHEMA** [<chizma nomi>]

va bunda quyidagi operator baza egasi hisoblanadi.

**AUTORIZATION**<baza egasi>

Yuqoridagi holat uchun misol ko`rib o`tamiz.

Misol:

**CREATE SCHEMA** talaba **AUTORIZATION** sa

Bu operatoridan keyin chizmaga kirgan barcha obektlarni yaratish operatorlari keltiriladi.

Chizmani olib tashlash uchun quyidagi operator ishlatiladi:

<chizmani olib tanlash>::= **DROP SCHEMA** [<chizma nomi>][**CASCADE** | **RESTRICT**]

Agar **RESTRICT** ko`rsatilsa, faqat chizma bo`sh bo`lsa, olib tashlanadi.

**CASCADE** ko`rsatilsa, shu chizma bilan bog`langan barcha jadvallar va boshqa obektlar olib tashlanadi, keyin esa chizma olib tashlanadi. Shuni aytib o`tish kerakki, ko`p BBB tizimlarida bunday operatorlar mavjud emas va bu ishni alohida aloqa orqali bajariladi.

### **Jadval yaratish.**

Jadval yaratish uchun **CREATE TABLE** operatori ishlatiladi, uning formati quyidagicha:

<jadval yaratish operatori>::= **CREATE TABLE**<jadval nomi>

( {<ustun nomi>{<domen nomi>|<berilgan turi>}

[**NOT NULL** | **NULL**][**UNIQUE**][**PRIMARY KEY**]

[<kelishuv bo`yicha qiymat>][<chek chegarasi>}] [,...] [<chegara tavsifi>)

<chegara tavsifi>::={ [ **CONSTRAINT**<chegaralanish nomi> ]  
 [<birlamchi kalit>][<muqobil kalit>[,...]]  
 [<tashqi kalit>] [ **CHECK**(<jadval bo'yicha shart>)] } [,...]

Har bir ustun uchun uning turi yoki domeni ko'rsatiladi. Kalit so'zlar ustunning xususiyatini berish uchun quyidagicha ishlatiladi:

- ✓ **NULL**-ustun qiymati bo'sh bo'lishi mumkin;
- ✓ **NOT NULL**- ustun qiymati majburiydir;
- ✓ **PRIMARY KEY**-ustun birlamchi kalit;
- ✓ **UNIQUE**- ustun qiymati noyob;
- ✓ **DEFAULT**-ustun uchun kelishuv bo'yicha qiymat berish;
- ✓ **CHECK**- ustun uchun chegara berish.

Qolgan parametrlar jadvalga tegishli, bo'lib ularga **CONSTRAINT** kalit so'zi bilan nom berish mumkin. Bu nom jadvalni o'zgartirishda ishlatish mumkin.

Misol:

talaba (id\_talaba, zach\_nom, familiya, ismi, sharifi, id\_millati, jinsi, pas\_ser, pas\_nom, tug\_yil, manzil, og\_irligi, tug\_joy, id\_tum, id\_res, id\_fak) munosabatining jadvali yaratilsin:

Oldin domenlarni yaratamiz

```
CREATE DOMAIN Jinsi CHAR (3)
CHECK VALUE IN ('erkak', 'ayol')
CREATE DOMAIN Dtug_yil INT
CHECK VALUE BETWEEN 2001 and 2015
CREATE TABLE talaba (id_talaba INTEGER PRIMARY KEY,
zach_nom CHAR(8) NOT NULL UNIQUE,
familiya VARCHAR (40) NOT NULL,
ismi VARCHAR (40) NOT NULL,
sharifi VARCHAR (40),
id_millati INTEGER,
Jinsi Jinsi,
pas_ser CHAR (2) NOT NULL,
pas_nom CHAR (8) NOT NULL
tug_yil Tug_yil,
manzil VARCHAR (255),
og_irligi DECIMAL (6,2),
tug_joy VARCHAR (50) DEFAULT 'Toshkent',
id_tum INTEGER,
id_ray INTEGER,
```

```

id_res INTEGER,
id_fak INTEGER,
stipendiya INTEGER,
UNIQUE (pas_ser, pas_nom),
CONSTRAINT Millat
FOREIGN KEY (id_millati)
REFERENCES millatlar
ON DELETE SET NULL,
CONSTRAINT oblast
FOREIGN KEY (id_obl)
REFERENCES viloyat
ON DELETE SET NULL,
CONSTRAINT Rayon
FOREIGN KEY (id_ray)
REFERENCES tumanlar
ON DELETE SET NULL,
CONSTRAINT Fakultet
FOREIGN KEY (id_fak)
REFERENCES fakultet
ON DELETE CASCADE
ON UPDATE CASCADE)

```

### **Jadval tavsifini o`rganish**

Jadval yaratilgandan keyin uni tuzilishini o`zgartirish uchun ALTER TABLE operatori qo`llaniladi, bu operator yordamida quyidagi ilovalarni amalga oshirish mumkin:

- jadvalga yangi ustun kiritish;
- jadvaldan ustunni o`chirish;
- jadval uchun yangi chegaralanishlar kiritish;
- jadval chegaralanishlarini o`chirish;
- ustun uchun kelishuv qiymatlarini o`rnatish;
- kelishuv qiymatlarini o`cherish.

Bu operatorning formati quyidagicha:

```

{ADD[COLUMN]<jadval nomi >
{<domen nomi> | <berilganlar turi>}
[NOT NULL | NULL ][UNIQUE]
[<kelishuv bo`yicha qiymat>][<check
chegarasi>]][,...]
| DROP [COLUMN]{<ustun nomi>
[RESTRICT | CASCADE]} [...]
| {ADD<chegara tavsifi>} [...]
| DROP CONSTRAINT {<chegara nomi>
[RESTRICT | CASCADE]} [...]}

```

```
| ALTER [COLUMN] {<ustun nomi>  
{SET DEFAULT <kelishuv qiymati>  
| DROP DEFOULT}}[,...]
```

Misol:

talaba jadvalidan zach\_nom ustunni o`cherish kerak, yangi ustun qo`shilsin kir\_yil nomli.

```
ALTER TABLE talaba  
  ADD COLUMN kir_yil INT NOT NULL DEFAULT  
  2011 CHECK kir_yil>2010;  
ALTER TABLE talaba  
  DROP COLUMN zach_nom RESTRICT;
```

### Jadvalni olib tashlash

Jadvalni olib tashlash uchun CREquyidagi operator ishlatiladi:

```
<jadval olib tashlash operatori>::=DROP TABLE <jadval nomi>  
[RESTRICT | CASCADE]
```

Jadvalni olib tashlashda, barcha satrlar ham yo`qotiladi. Agar **RESTRICT** kalit so`zi ko`rsatilgan bo`lsa, boshqa obyektlarda jadvalga murojaat bo`lsa (masalan tasavvurlarda), u holda jadval olib tashlanmaydi, **CASCADE** kalit so`zi berilganda esa, barcha havola qilgan obyektlar ham o`chiriladi va h.k. Shu sababli, **CASCADE** kalit so`zni ehtiyotkorlik bilan ishlatmoq kerak.

Misol:

```
DROP TABLE talaba RESTRICT
```

### Indeks yaratish

Indeksler jadval satrlarini tezkor qidirish uchun ishlatiladi. Agar jadvalda indekslar mavjud bo`lsa har satr qo`shilganda jadval indeks bo`yicha qayta tartiblanadi shu sababli yangi satr kiritish vaqti cho`ziladi, lekin jadvalda qiymat qidirish ancha tezlashadi. SQL tilining nomunasida bunday operator yo`q, lekin barcha dialektlarda bunday operator bor. Indeks jadvalini yaratish uchun CREAT INDEX operatorini qo`llash mumkin.

```
<indeks yaratish operatori>::= CREATE [UNIQUE] INDEX  
  <indeks nomi> {ON <jadval nomi>  
  ({<ustun nomi> [ASC] | DESC}} [,...])
```

Misol. Birlamchi kalit bo`yicha indeks yaratilsin:

```
CREATE UNIQUE INDEX indtal ON talaba  
  (id_talaba)
```

Indeksni olib tashlash uchun **DROP INDEX** operatori ishlatiladi:

<indeksni olib tashlash operatori>::= **DROP INDEX** <indeks nomi>

Misol:

**DROP INDEX** indtal

### **Tasavvurlar**

Tasavvur- bu bitta yoki bir nechta fizik jadvallar ustidan bajarilgan amallar natijasida hosil bo`lgan natijaviy dinamik jadvaldir. Tasavvur bu so`rov natijasida yaratilgan virtual jadval hisoblanadi. Tasavvurni haqiqiy jadvaldan farqi shundaki, bunday jadval fizik mavjud emas. Uning ustunlari va satrlari haqiqiy jadvaldan olingan va tasavvur ustidan bajarilgan barcha ishlovlar haqiqiy jadvallarni o`zgartiradi. BBBT tasavvurlarni tavsifini berilganlar bazasida saqlaydi. Bunday tasavvurga murojaat qilinganda BBBT ikki usul bilan ishlashi mumkin.

Birinchi usul bo`yicha tasavvurga bo`lgan har bir murojaatda ko`rsatilgan so`rov bajariladi va natijada vaqtincha jadval hosil bo`ladi, tasavvur ishlatilgandan so`ng jadval tizim tomonidan olib tashlanadi.

Ikkinchi usulda esa, tasavvurlar uchun alohida vaqtincha jadval tuziladi va tasavvurga kirgan jadvallarning o`zgarishining jarayonida birdaniga vaqtincha jadval ham o`zgaradi.

### **Tasavvur yaratish**

Tasavvurni yaratish uchun **CREATE VIEW** operatori ishlatiladi uning sintaksisi quyidagicha:

<tasavvurni yaratish operatori>::=**CREATE VIEW** < tasavvur nomi>[(*ustun yangi nomi* >[,...])] **AS**  
<tanlash operatori>[**WITH** [**CASCADED** |**LOCAL**]  
**CHECK OPTION**]

Tasavvur **SELECT** operatori orqali tavsiflanadi. Natijaviy jadvalning ustunlariga yangi nomlar berish mumkin, bu holda ro`yhatdagi ustunlar nomining soni natijaviy jadvaldagi ustunlarning soniga teng bo`lishi kerak. Agarda natijaviy jadvalda hisoblanuvchi ustunlar yoki bir hil nomli ustunlar mavjud bo`lsa, ustunlarning yangi nomlariko`rsatilishi shart. Tanlash operatori orqali berilgan so`rov qismi aniqlovchi so`rov, deb nomlanadi. **WITH CHECK OPTION** konstruksiyasi ko`rsatilsa, o`zgartirilgan satr aniqlovchi so`rovning WHERE bo`limidagi ko`rsatilgan shartni qonoatlantirmasa, manba jadvalga mos satr o`zgartirilmaydi.

Misol:

Gorizontal tasavvurni yaratish.

Faqat 1 raqamli fakultet talabalarini ko`rish imkoniyatini beruvchi tasavvurni yaratish quyidagicha bo`ladi:

```
CREATE VIEW faktalaba
AS SELECT *
FROM talaba
WHERE kod_fak=1;
```

Bu operatorni bajarish natijasida faktalaba nomli tasavvur yaratiladi. Bu tasavvur manba jadvalni barcha ustunlaridan iborat bo`lib, faqat bir raqamli fakultetga tegishli satrlarni ko`rsatadi. Bunday so`rov gorizantal tasavvur deb nomlanadi, chunki gorizantal chiziq bilan cheklanadi. Bu tasavvurni ko`rish uchun quyidagi operatorni ishlatish mumkin:

```
SELECT * FROM fak_talaba
```

Tasavvur orqali berilganlar bazasini himoya qilish mumkin, chunki tashkilotning har bir bo`limi faqat o`ziga tegishli ma`lumotlarni ko`rish imkoniyatiga ega bo`ladi.

Misol:

Vertikal tasavvurni yaratish.

Faqat, 1 raqamli fakultetning talabalarini ko`rish imkoniyati bo`lsin.

Stipendiya to`g`risadgi ma`lumot, faqat hisob bo`limiga tegishli bo`lgani uchun tasavvurdada ko`rsatilmasin. Asosan talabaning tartib raqami, reyting raqami, familiyasi, ismi, otasining ismi, tug`ilgan yili va jinsi ko`rsatilsin.

```
CREATE VIEW faktalaba1
AS SELECT id_talaba, reyt_nom
familiya, ismi, sharifi, tug_yil,
jinni, id_fak
From talaba
WHERE id_fak=1;
```

Bu tasavvur foydalanuvchiga faqat ma`lum ustunlardagi ma`lumotlarni ko`rishga imkoniyat beradi, boshqa ustunlar himoyalangan bo`ladi va imtiyozga ega bo`lgan foydalanuvchilarning ko`rish imkoniyati bo`ladi. Bu tasavvurni boshqacha ham berish mumkin:

```
CREATE VIEW faktalaba1
AS SELECT id_talaba, reyt_nom
familiya, ismi, sharifi, tug_yil,
jinni,
FROM faktalaba
```

Manba jadvallarning bir nechta ustunlari tushirilgan tasavvurlar vertical tasavvurlar, deb nomlanadi. Agar jadval bilan har hil foydalanuvchilar Ishlasa, vertical tasavvurlarni qo`llash maqul hisoblanadi, chunki bu holda faqat kerakli ustunlar chiqar edi holos.



## TRANZAKSIYANI QO`LLASH

SQL namunasida tranzaksiya modelini tavsiflashda ikkita mahsus operatorlar qo`llaniladi – **COMMIT** va **ROLLBACK**. Bir necha ketma-ket SQL operatorlarni tiklash nuqtai nazaridan butun bo`linmas amal *tranzaksiya* deyiladi. Tranzaksiya berilganlar bazasini butunligini saqlash uchun qo`llaniladi. SQL tilining namunasida yozilishi bo`yicha, har bir operator bajarilishini boshlaganda, tranzaksiya avtomatik ravishda boshlanadi. Tranzaksiya tugamaguncha, ya`ni tranzaksiya oshkor ravishda to`xtatilmaguncha, boshqa parallel ishlagan tranzaksiyalar, kiritilgan o`zgarishlarni hisobga olmaydi. Tranzaksiyaning bajarilishi

- **COMMIT** operatorini kiritish orqali. Bu holda tranzaksiya muvaffaqiyatli tugaydi va bazaga kiritilgan o`zgarishlar doimiy saqlanib qoladi. Bundan keyin bajarilgan har qanday operator yangi tranzaksiyani boshlaydi.
- **ROLLBACK** operatorini kiritish orqali. Bu holda tranzaksiya jarayonida qilingan barcha o`zgarishlar qayta joyiga olib kelinadi (man etiladi) va baza tranzaksiya boshlanishdan oldingi holatga qaytadi. Tranzaksiya muvaffaqiyatsiz bajarilgan deb hisoblanadi. Bundan keyin bajarilgan har qanday operator yangi tranzaksiyani boshlaydi.
- Agarda **COMMIT** operatori oshkor ravishda kiritilmasa ham SQL operatori programma orqali bajarilsa va programma bajarishni muvaffaqiyatli bajarildi deb hisoblanadi.
- Agar SQL operatori programma orqali bajarilsa, programma bajarilishi avariyali tugatilsa, oxirgi tranzaksiyada qilingan o`zgarishlar qayta joyiga olib kelinadi va baza tranzaksiya boshlanishdan oldingi holatga qaytadi.

SQL tilida bir-biriga kirgan tranzaksiyalarni qo`llash man etilgan. **SET TRANSACTION** operatorlar orqali foydalanuvchilar tranzaksiya jarayonini boshqarishi mumkin.

Bu operatorning formati quyidagicha bo`ladi:

```
<tranzaksiya o`rnatish> ::= SET TRANSACTION [ READ ONLY |  
READ WRITE ] [ISOLATION LEVEL {READ UNCOMMITTED |  
READ COMMITTED | REPEATABLE READ | SERIALIZABLE }]
```

**READ ONLY** kvalifikatori, tranzaksiyada berilganlarni faqat o`qish mumkinligini bildiradi. **READ WRITE** – tranzaksiyada ham o`qish, ham ham yozish mumkinligini ko`rsatadi. Kelishuv bo`yicha **READ WRITE** hisoblanadi (bu holda **READ UNCOMMITTED** ko`rsatmasligi kerak).

**ISOLATION LEVEL** kvalifikatori izolatsiya darajasini ko`rsatadi. Bu kvalifikator tranzaksiyalar parallel bajarish jarayonida tartibni ko`rsatadi. Bajarish tartibi natija (seriyasida) quyidagi izolyatsiya darajalari ishlatilishi mumkin:

- **READ UNCOMMITTED** – tugatilmagan o`qish darajasi;
- **READ COMMITTED** – tugatilgan o`qish darajasi;

- **REPEATABLE READ** – qayta o`qish darajasi;
- **SERIALIZABLE** – tartib bo`yicha o`qish darajasi.

Agar ba`zi tranzaksiyalar quyi darajaga ega bo`lib tartibsiz ishlasa, quyidagi xato holatlar bo`lishi mumkin:

- yangi fiksirlanmagan berilganni o`qish (<<iflos>> o`qish);
- bir maydondan ikki marta har xil qiymat o`qish (qaytarilmas o`qish);
- bitta shart bo`yicha ikki marta har hil satrlar sonini olish (fantom).

Bu holatlar uchun har bir izolyatsiya darajasi p`ziga mos ruxsatlar beradi:

Izolyatsiya darajasi	<<Iflos>> o`qish	Qaytarilmas o`qish	Fantomlar
<b>READ UNCOMMITTED</b>	Ha	Ha	Ha
<b>READ COMMITTED</b>	Yo`q	Ha	Ha
<b>REPEATABLE READ</b>	Yo`q	Yo`q	Ha
<b>SERIALIZABLE</b>	Yo`q	Yo`q	Yo`q

Yangilanish natijalari yo`qolmasligi kerak, shu sababli eng quyi darajali izolyatsiyadagi tranzaksiyalar bilan ehtiyot bo`lib ishlash kerak. Agarda quyi darajali izolyatsiyadagi tranzaksiyalar bilan ehtiyot bo`lib ishlash kerak. Agarda quyi darajali izolyatsiyada **READ COMMITTED** ko`rsatilsa, kelishuv bo`yicha **READ ONLY** qabul qilinadi. To`liq havfsizlikni faqat **SERIALIZABLE** yuqori darajali izolyatsiya beradi (oshkormas qiymat).

### Berilganlarni butunligini saqlashdagi chegaralarni tekshirish

Ba`zi bir holatlarda, berilganlarni butunligini tekshirish vazifasi har bir operatoridan keyin darhol bajarilmasdan, aksincha har bir tranzaksiya bajarilgandan keyin tekshiriladi. Bunday imkoniyatlar jadalni tafsiflashda **NOT DEFERRABLE** (darhol) yoki **DEFERRABLE** (kechiktirilgan) chegaralarni tekshirish rejimlarini berish orqali amalga oshiriladi, uning sintaksisi quyidagicha:

<tekshirish rejimi>::={**DEFERRABLE [ INITIALLY DEFERRED | INITIALLY IMMEDIATE]**} | {**NOT DEFERRABLE**}

Kelishuv bo`yicha **NOT DEFERRABLE** rejimi o`rnatiladi. Agar **DEFERRABLE** ko`rsatilsa, ikkita rejimni ishlatish mumkin, darhol tekshirish yoki kechiktirilgan tekshirish, bu rejimlarni tranzaksiya bajarish jarayonida o`zgartirish mumkin. Buni **SET CONSTRAINTS** operatori orqali bajarish mumkin. Operatorning sintaksisi quyidagicha:

<chegaralash rejimi>::= **SET CONSTRAINTS {ALL | <constraint nomi> [,...]} { DEFERRED | IMMEDIATE }**

Bunda **DEFERRED** har bir chegaraga yoki ko`rsatilgan chegaraga kechiktirilgan tekshirish rejimini o`rnatadi, ya`ni tranzaksiya bajarishni tugatish oldidan chegaralar tekshiriladi va shart qanoatlanirilmasa, tranzaksiya qaytariladi. **IMMEDIATE** darhol tekshirish rejimini o`rnatadi, ya`ni har bir amaldan so`ng chegaralar tekshiriladi va shart bajarilmasa, tranzaksiya qaytariladi.

Boshlang`ich holatini **INITIALLY DEFERRED** yoki **INITIALLY IMMEDIATE** bilan beriladi. Kelishuv bo`yicha **INITIALLY IMMEDIATE** rejimi o`rnatiladi.

### **Berilganlar bazasiga kirish uchun ruxsat berishni boshqarish**

Har bir BBB tizimida foydalanuvchilarga berilganlarga kirish ruxsatini beradigan imkoniyatlar bo`lishi kerak. SQL tilida bunday imkoniyatlarni GRANT va REVOKE operatorlari beradi. Bu operatorlar orqali berilganlar bazasidagi jadvallar himoyalanaadi. Himoyalash usuli foydalanuvchilarning identifikatori va ularga berilgan egalik huquqi hamda imtiyozlar orqali bajariladi.

### **Foydalanuvchilarning identifikatori va egalik huquqi**

Berilgan bazasidan foydalanuvchini belgilaydigan SQL tilining oddiy identifikatori, *foydalanuvchilar identifikatori* deb nomlanadi. Har bir foydalanuvchiga berilganlar bazasini administrator tomonidan noyob identifikator beriladi va bu identifikator beriladi va bu identifikator himoyalash uchun parol ham beriladi. BBBT tomonidan bajarilgan har bir SQL operatori biror bir foydalanuvchi tomonidan bajariladi. Foydalanuvchining identifikatori orqali foydalanuvchiga qaysi ob`ektlarga havola qilish va ular ustidan qanday vazifalarni bajarish huquqi beriladi.

Har bir obyektning o`z egasi bor. Chizmani berish konstruksiyani **AUTHORIZATION** kalit so`zida shu chizmaga kirgan barcha obyektlarni egasi foydalanuvchining identifikatori orqali beriladi.

Boshida faqat obyekt egasi obyekt ustidan har qanday amallar bajarish huquqiga ega.

## **BERILGANLAR BAZALARINING SERVERLARI**

Mijozlarga xizmat ko`rsatishda kompyuter va programma vositalaridan iborat, lokal va global kompyuter tarmoqlaridagi serverlar – ishchi stansiya va/yoki boshqa serverlar keng qo`llaniladi.

Serverlarga misollar:

- hamma ishchi stansiyalardagi umumiy fayllar bazasini qo`llab quvvatlovchi gafayllar server;
- global tarmoq – Internetdagi axborotlarni yetkazishni ta`minlovchi, internet server;
- electron pochta bilan ishlashni ta`minlovchi pochta server;

- local tarmoqdan soʻrovlarni qabul qilib, natijani qaytarishni taminlovchi berilganlar bazalarining server.

Berilganlar bazalarining server deganda, «mijoz-server» arxitekturasiga asoslangan, server va mijoz qismlarini turli nisbatlarda oʻz ichiga olgan, BBBT tushiniladi. Eng keng tarqalgan serverlar – Microsoft SQL Server, Oracle, IBM DB2 Universal DataBase, Informix va boshqalar.

Ushbu serverlardagi bitta berilganlar bazasining hajmi million terabayt ga yetishi mumkin.

### **Taqsimlangan ishlovning tayanch arxitekturalari**

Berilganlarga ishlov berish texnologiyasiga koʻra *BB markazlashtirilgan va taqsimlangan boʻlimlardan* iborat.

*Markazlashgan* BB bitta hisoblash tizimining xotirasida saqlanadi. Bu hisoblash tizimi meynfreym boʻlishi mumkin- u holda, unga kirish termenallar orqali yoki shaxsiy kompyuterlar local tarmogʻidagi fayl server amalga oshiradi.

*Taqsimlangan* BB bir nechta, hattoki kesishuvchi yoki bir-birini qaytaruvchi qismlardan iborat boʻlishi va hisoblash tarmogʻining turli tugunlarida saqlanishi mumkin. Bunday BB bilan ishlash *taqsimlangan berilganlar bazasini boshqarish tizimi* orqali olib boriladi.

Berilganlarga kirish usuli boʻyicha BB *lokal kirish va tarmoq orqali kirishga* boʻlinadi. Hozirgi barcha zamonaviy berilganlar bazasiga koʻp foydalanuvchili tarmoqdan kirish usulini qoʻllash mumkin.

*Markazlashgan berilganlar bazasiga tarmoqdan kirish quyidagi toʻrtta turdagi arxitekturaga* ega

- «fayl-server»;
- «mijoz – server» BB
- «nozik» mijoz ilovalar server- BB server;
- Berilganlar bazasining faol server.

### **«Fayl-server» arxitekturasi**

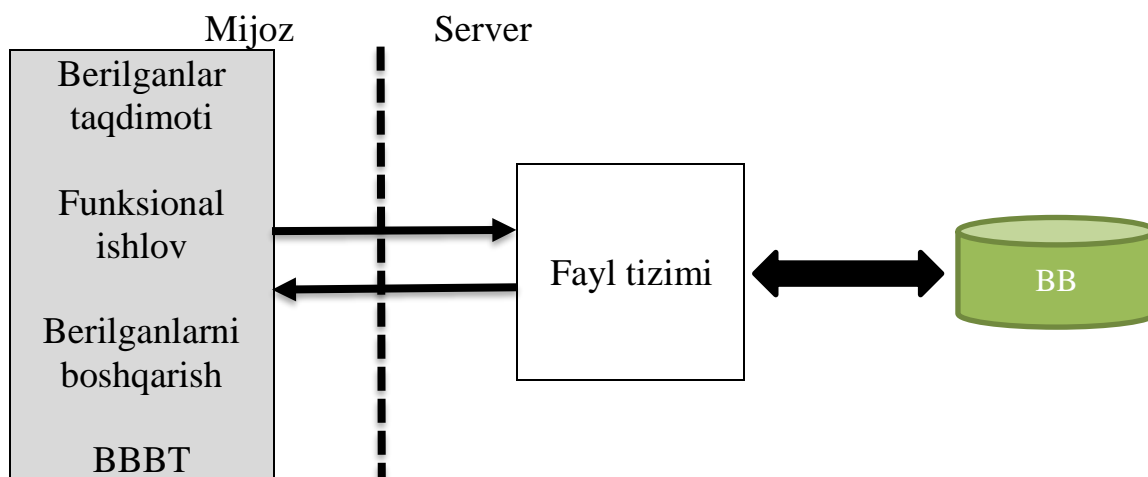
«Fayl-server» arxitekturasida, berilganlar bazasiga kirish uchun kompyuterlarning biri markaziy mashina (fayllar serveri) sifatida olinadi. Unda umumiy foydal aniladigan markazlashgan BB bir yoki bir nechta fayllar sifatida saqlanadi. Tarmoqdagi qolgan barcha kompyuterlar ishchi stansiya funksiyalarini bajaradi. BB fayllar foydalanuvchilar soʻroviga qarab ishchi stansiyalariga uzatiladi va ularda berilganlar qayta ishlanadi. Agarda berilganlar bazalarining *bitta fayliga koʻp murojaatlar* boʻlsa, axborot tizimining *samaradorligi pasayadi*. Bunday holatda, foydalanuvchilar oʻzining ishchi stansiyasida local berilganlar bazasini hosil qilishi ham mumkin.

“Fayl-server” arxitekturasida berilganlarga maqsadli ishlov va boshqarish mijoz kompyuterida maxsus fayllardan iborat, BB esa server-kompyuter saqlanadi. Bunda server tarkibi BBB tizimining vositalari bilan emas, balki masofali taqsimlangan fayllarga kirishni taminlovchi operatsion tizimni tarkibiy qismlari orqali ifodalanadi. Server berilganlarni saqlash, soʻrovlarga xizmat koʻrsatish va olingan natijalarni foydalanuvchiga maxsus ishlov uchun joʻnatadi. SQL tilidagi berilganlar bazasiga

so`rov, BBT orqali kiritish-chiqarish buyruqlarga aylantiriladi va server-kompyuterning operatsion tizimida ularga ishlov beriladi.

Bunday ikki tarkibga ajratish quyidagi imkoniyatlarni yaratadi:

- berilganlar bazasi bilan bir vaqtda bir nechta amaliy programmalarining ishlashi;
- boshqarish funksiyasini markazlashtirish, ya'ni berilganlar himayasi va butunligi, zahiralardan birgalikda foydalanish;
- taqsimlangan berilganlar bazalarida so`rovlarning parallel ishlovini ta'minlash;
- tarmoq va ishchi stansiyalar zahiralarni bo`shatish va/yoki yengillashtirish;
- berilganlar bazalarining mahsus serverlaridan foydalanish hisobiga berilganlarni boshqarish unumdorligini oshirish;



“Fayl-server” arxitekturasi kamchiliklari quyidagilardan iborat:

- tarmoqdagi va mijoz kompyuteridagi yuqori yuklamalar yuzaga keladi, chunki berilganlarni almashish fayl tizimidagi axborot birligi darajasida bajariladi;
- berilganlar himoyasi quyi darajada bo`ladi, chunki berilganlar bazasining fayllariga kirish server – kompyuterning operatsion tizimining umumiy vositalari orqali boshqariladi;
- mijoz tomonidan jamlangan funksional ishlovning tijorat qoidalari bir-biriga zid bo`lishi mumkin.

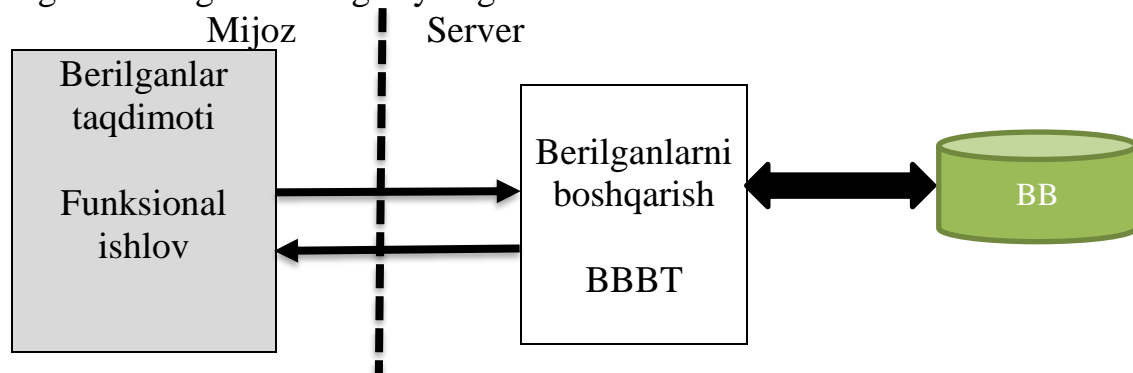
Berilganlar bazasining chizmalari uchun axborot o`lchovining eng katta yig`ma vaqt sarf bo`lishi “fayl-server” arxitekturasi nozik xususiyatlaridan bittasi hisoblanadi.

### «Mijoz server» arxitekturasi

«Mijoz server» arxitekturasi server OT orqali boshqariladigan ajratilgan serverda, maxsus programmalar ta'minoti esa BB serveriga o`rnatiladi. Bu arxitekturani

ajratilgan serverli arxitektura, deb ham atashadi. U hamma relatsion BBBT tomonidan qo`llab quvvatlanadi.

BBBT ikki qismga bo`linadi: mijoz va server. BB severning asosiy vazifasi SQL tilidan foydalanish. So`rovlar mijoz tomonidan BB serveriga beriladi, so`ngra serverda qidiruv va berilganlarni tanlab olish jarayoni amalga oshiriladi. Shartlarga javob beruvchi yozuvlar, serverda jamlanadi va so`rov ishlov qilingandan keyin foydalanuvchiga uzatiladi. Shu tarzda, tarmoqdagi ma'lumotlarni uzatish miqdori bir necha marta kamayadi. Mijoz ilovalar stolbop BBB tizimlarida amalga oshirilishi mumkin. SQL-server bilan mijoz ilovalarning o`zaro harakati ODBC-drayver orqali amalga oshiriladi. ODBC-drayver turli jinsdagi berilganlar bazalariga kirish algoritmining standartiga aylangan.



«Mijoz-server» arxitekturasining afzalliklari quyidagilardan iborat:

- server va mijoz kompyuteridagi yuklama kamayadi;
- kirish-chiqishni optimallashtirish va buferlash evaziga ishlov unumdorligi oshadi va tarmoq grafigi kamayadi;
- BBBt vositalari orqali berilganlarning himoyasi amalga oshirish natijasida, foydalanuvchi uchun ruhsat etilmagan harakatlarni to`xtatadi;
- server tranzaksiyalarni boshqarishni amalga oshiradi va bir vaqtda berilganlarni o`zgartirishga qilingan urinishlar to`xtatiladi;

«Mijoz-server» arxitekturasining kamchiliklari esa quyidagilardan iborat:

- funksional ishlovning tijorat mantig`i va berilganlar tasavvuri bir qancha foydalanuvchilarning ilovalari uchun bir xilda bo`lishi mumkin, bu esa zahiralarga bo`lgan talabni oshiradi, masalan programmalar kodi va so`rovlarni takrorlash;
- mijoz tarafda jamlangan funksional ishlovning tijorat qoidalari bir-biriga zid bo`lishi mumkin.

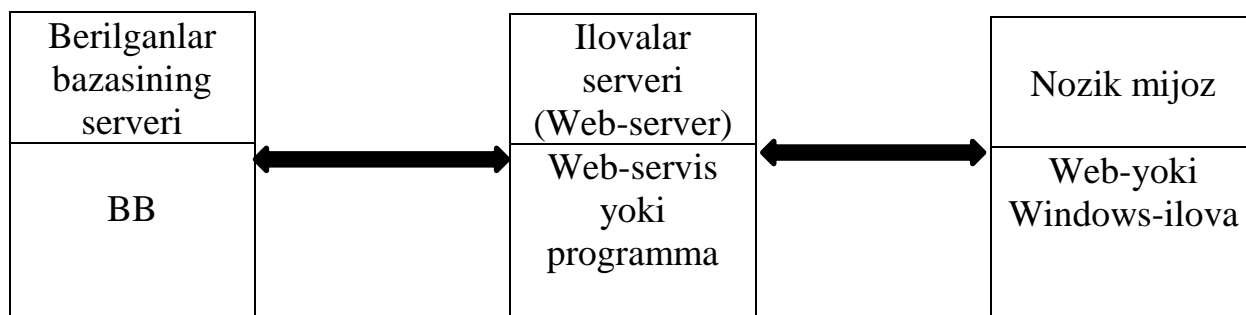
### «Uch bosqichli» arxitektura

«Uch bosqichli» arxitektura Internet va Intranet tarmoqlarida ishlaydi. Mijoz qism foydalanuvchilari bilan o`zaro harakatlanuvchi qism, Web-brauzerda HTML sahifani yoki Web-servisidagi Windows-ilovalarida iborat. «Nozik» mijozlar eng sodda ilovalardan iborat bo`lib, faqatgina berilganlarni uzatish, lokal xeshlash, foydalanuvchi aloqa vositalarini tavsiflash, lokal xeshlash, foydalanuvchi aloqa

vositalarini tavsiflash, tahrirlash va oddiygina ishlov vazifalarini ta'minlash bilan shug'ullanadi.

*Mijoz ilovalari BB serveriga to'g'ridan-to'g'ri murajaat qilmaydi, balki maxsus programma ta'minoti bo'lmish oraliq qatlam orqali ishlaydi. Bu bitta bo'g'in yoki murakkab tuzilish bo'lishi mumkin. Maxsus programma ta'minoti bo'lmish oraliq qatlamni ilovalar serveri deyiladi, u mijozlarning so'rovlarini qabul qiladi, ularga programmalashtirilgan mantiqiy-tijorat qoidalariga asosan ishlov beradi, zarurat tug'ilganda BB serveriga mos shaklga o'zgartirib uzatadi. Ilovalar serveri sifatida Web-servis keladi. Misol uchun, Oracle Forms Server.*

*BB serveri qabul qilgan so'rovlarni bajaradi va narijalarni ilovalar serveriga jo'natadi, u esa ishlov natijalarini mijozga qaytaradi.*



Sunday qilib, BB «Uch bosqichli» arxitekturaning tarkibi quyidagilardan tashkil topgan:

- berilganlarni uzatish, tavsiflash, tahrirlash va sodda ishlovni ta'minlovchi «nozik» mijoz ilovalaridan;
- bitta kompyuterda yoki tarqatilgan holatda lokal tarmoqda, bir yoki bir nechta bo'g'inli oraliq qatlam hisoblanadigan – *programmalar ta'minotida*;
- BB qo'llab-quvvatlovchi va so'rovlarga ishlov beruvchi – *BB serverlaridan*.

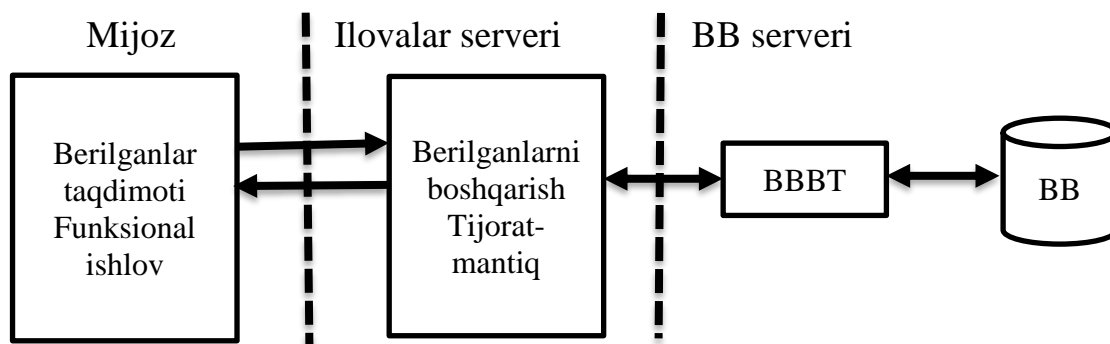
Ayrim programmalash muhitlarida mijoz programma ta'minoti va oraliq programma ta'minotini yaratish vositalari va tarkibli jihozlari mavjud. Ilovalar serveri BB serveri bilan aloqa qilganda, berilganlarga kirishdagi texnologiyalardan qo'yilgan masalaga nisbatan birini tanlab foydalanadi.

Masofali mijoz ilovalarini yaratishda, mahsus tarkiblar termasidan iborat DataSnap texnologiyasidan foydalaniladi. Bu texnologiya standart transport vositalaridan foydalanib, masofadagi mijoz ilovalarini ilovalar serveriga bog'lanishni ta'minlaydi. DataSnap tarkiblari AppServer aloqalaridan foydalanib, ilovalar serveridagi funksiyalarga kirishni ham ta'minlaydi.

Mijoz kompyuterlari ko'pincha o'z bo'lgani uchun, murakkab mantiqiy-tijorat masalalarini amalga oshirishda ilovalar serveri hamda foydalanuvchilarning bir jinsli so'rovlarini bajarishni optimallashtirish usullaridan foydalanish, tizimni tezligini oshirishda muhim ahamiyatga ega. Masalan, BB serveri juda ham yuklangan bo'lsa, ilovalar serveri mustaqil ravishda foydalanuvchi so'rovlariga ishlov berishni BB serverini qo'shimcha yuklamasdan o'zi bajarishi mumkin.

Zahiralarga bo'lgan talablar darajasini keyinchalik kamaytirish ilovalar serverining kiritilishi hisobiga erishilgan. Ilovalar serveriga berilganlarni

boshqarish programmasi tarkiblarining kattagina qismi va tijorat-mantiqning ham katta qismi o`tkaziladi. Bunda BB serverlari faqatgina BBBT funksiyasini ta'minlaydi. Ilovalar serverida mijoz ishchi stansiyalarining amaliy programmalar bilan aloqasi API (Application Programming Interface) aloqalari orqali o`rnatiladi. Mijoz qismining vazifasi ilovalar serveridagi funksiyalarni chaqirishdan iborat bo`ladi. Amaliyprogrammalar BB serveriga SQL so`rovlar orqali murojaat qiladi.



«Uch bosqichli» arxitekturaning afzalliklari quyidagilardan iborat:

- berilganlarga ishlovdagi umumiy funksiyalarni mijoz ilovalar to`plamida ko`p marotaba ishlatilishi;
- tijorat-mantiqni markazlashgan holda yuritish, ya'ni bazalarga o`zgarishlar kiritganda mijoz ilovalarda ularni nusxasini olmaslik;
- mijoz kompyuterlarda programma ta'minotining berilganlarga kirishi boshqarish qismini o`rnatishni kerak emasligi;
- berilganlar bazasiga kirishni optimallashtirish;
- avtonom holatda, berilganlar o`zgartirilayotgan vaqtda, qoldirilgan berilganlar bazasini yangilash imkoniyati berilganlar bazasiga, keyingi ulanishda amalga oshiriladi;
- bir-birini o`zaro almashtiruvchi, bir nechta ilovalar serverdagi programma ta'minotini nusxalash natijasida tezlikni va ishonchlikni ortishi;
- foydalanuvchi vakolatlarini tekshirishi BB serveridan ilovalar serveriga ko`chirish;
- ilovalar serveri va BB serverining parallel ishlashi natijasida ilovalar serveri quvvatining kamayishi;

Ilovalar serverining mavjudligi tizimning havfsizligini oshiradi, ya'ni unda foydalanuvchilarning mualliflik huquqini tekshirish va havfsizlikka doir boshqa funksiyalarni ham amalga oshirish mumkin. Bundan tashqari, berilganlarni uzatishning himoyalangan kanallaridan oson foydalanish mumkin.

**Berilganlar bazasining «faol server»** arxitekturasida, tijorat mantig`ining qarama-qarshiliklari server tomonidan nazorat qilinadi. Tijorat mantig`ining funksiyalari mijoz va server tomonlari orasida taqsimlanadi. Umumiy funksiyalar saqlanuvchi proseduralar sifatida tayyorlanadi. Berilganlar bazasidagi hodisalarni kuzatish uchun triggerlar mexanizmi kiritiladi. Hodisa yuz bersa SQL operatori bajariladi yoki triggerga bog`liq saqlangan prosedura chaqiriladi.



Saqlanuvchi proseduralar va triggerlar BB bilan ishlovchi ixtiyoriy mijoz ilovalar tomonidan foydalangan bo'lishi mumkin. Bu programma kodlarini nusxalashni kamaytiradi va har bir so'rovni kompilatsiya qilish zaruratidan xalos etadi. Arxitekturaning kamchiligi server yuklamasining ortishi bilan bog'liq bo'ladi. Bunday arxitekturaning ba'zan «nozik» mijozli model, deb atashadi. Avvalgi arxitekturalarning hammasini «semiz» mijozli model, deb atashgan, chunki unda mijoz tomonidan ko'pgina funksiyalar amalga oshirilgan.

Trigger – bu BBBT yadro tomonidan bajariladigan programma bo'lib, ular BB jadvalarini tiklashdan oldin va keyin bajariladi. Ya'ni triggerlar BB jadvalini tiklashda oldin va keyin, BBBT yadrosi tomonidan bajariladigan programmalaridir. Ular berilganlar bazasini butunligi avtomatik ravishda taminlab beradi.

BB serveri modelini ORACLE, Sybase, Informix, Ingress, Progress kabi BBB tizimlari ta'minlaydi.

Berilganlar bazasining faol server arxitekturasining afzalliklari:

- bu BBBT asosida yaratilgan tizimlar yuqori ishlab chiqarishga ega, chunki shinalar bo'yicha faqat SQL so'rovlari va bajarish natijalari uzatiladi;
- BBBT taqsimlangan qayta ishlashni ta'minlaydi;
- bu BBBT doirasida juda ko'p server programmalar bo'lib, ular ilovalar yaratishni yengillashtiradi.

*Kamchiligi:*

- ular oldingi sinf BBB tizimiga nisbatan qimmat, o'zlashtirish qiyin;
- ularni samarali ishlashi uchun yuqori tezlikli server va tarmoqlar talab etiladi.

### **So'rovlarni optimallashtirish va qayta ishlash**

Taqsimlangan berilganlar bazasini berilganlarni joylashtirish va ularni qidirish muhim masalalardan biridir. Berilganlarni qidirish berilganlarni tuzilishiga moslab amalga oshiriladi. 1-holda lokal berilganlar bazasida berilganlarni tuzilishida bir-biridan farq qiladi. 2-holda BB umumiy tuzilishiga ega bo'ladi, lokal berilganlar bazasining tuzilishida faqat aniq qiymatlar bilan farq qiladi. Taqsimlangan berilganlar bazasida ma'lumotlarni qidirishni quyidagi usullari mavjud:

- ✓ foydalanuvchi eng yaqin tizim bilan o'zaro bog'langan, agar eng yaqin BBB tizimida kerakli ma'lumot bo'lmasa, unda qidirish berilganlar bazasida bajariladi.
- ✓ Qidirilayotgan berilganlar tuzilishi axborot bo'yicha tugunlarda amalga oshiriladi. Bunday berilganlar tuzilishibarcha lokal tizimlarda saqlanishi lozim.

Bunda bitta tizim ajratib boshqaruvchi deb nomlanadi va unda barcha berilganlarning tuzilishi saqlanadi. Foydalanuvchi 1-qidirishda kerakli berilganni topolmasa, u berilganlarning tuzilishlariga murojaat qiladi.

