

**O‘ZBEKISTON RESPUBLIKASI**  
**OLIY VA O‘RTA MAXSUS TA‘LIM VAZIRLIGI**  
**FARG‘ONA DAVLAT UNIVERSITETI**  
**FIZIKA-MATEMATIKA FAKULTETI**

Informatika o‘qitish metodikasi yo‘nalishi 13.404-guruh bitiruvchisi

**Djuraboyeva(Sobirova) Gulnoraxon Shuxratjon qizining**

**“SQL tilini o‘qitishda usulbiy xususiyatlar taxlili”**

mavzusidagi

**BITIRUV MALAKAVIY ISHI**

**Ilmiy raxbar: I.Tojimamatov**

Farg‘ona – 2017

Bitiruv malakaviy ish kafedraning 2017 yil \_\_\_\_\_dagi \_\_\_\_\_ -  
yig'ilishida muxokama qilingan va himoyaga tavsiya etilgan.

Kafedra mudiri

fizika - matematika fanlari nomzodi \_\_\_\_\_ I.Xaydarov

Taqrizchilar:

1. \_\_\_\_\_ FARPI, texnika fanlari  
(*imzo*) nomzodi dotsent. A.A. Xolmurzayev

2. \_\_\_\_\_ "Axborot texnologiyalari" kafedrası  
(*imzo*) o'qituvchisi I.Tojimamatov



## MUNDARIJA

KIRISH.....	5
I BOB. SQL TILINI O‘QITISHDA USULBIY XUSUSIYATLARNI PEDAGOGIK VA INNOVATSIYA JIHATLARI .....	9
1.1. Axborot kommunikatsion texnologiyalari fanlarini o‘qitishda qo‘llaniladigan uslublar .....	9
1.2. SQL tilini o‘qitishda pedagogik-innovatsion usullar. ....	19
II BOB. SQL TILINI FANINING TUZILISHI VA BO‘LIMLARI MAZMUNI.	28
2.1 SQL tili va ma’lumotlar tiplari.....	28
2.2 SQL tili jadvallar bilan ishlash.....	29
2.3 Maydonlarni qiymatlarini kiritish, o‘chirish va o‘zgartirish.....	39
2.4 SELECT so‘rov operatori .....	42
2.5. Mantiqiy operatorlar.....	46
2.6. So‘rovlarda guruhlash va funksiyalar .....	52
2.7. Foydalanuvchilar va ular imtiyozlari .....	62
2.8 Tarmoqda ma’lumotlar bazalari arxitekturasi.....	69
III BOB. MICROSOFT ACCESS DASTURIDA ODDIY SQL TILI SO‘ROVLARNI YARATISH USLUBLARI.....	77
3.1. Microsoft Access dasturida oddiy SQL tili so‘rovlarni yaratish uslublari .....	77
3.2. Microsoft Access dasturida oddiy SQL tili so‘rovlarni yaratish. ....	82
XULOSA.....	86
FOYDALANILGAN ADABIYOTLAR RO‘YXATI.....	88

## KIRISH.

Mustaqillik yillarining dastlabki kunlaridan boshlab mamlakatimizda axborotlashgan jamiyat qurish borasida keng islohatlar olib borilayotgan davrda yoshlarning axborot madaniyatini shakllantirish ta'lim tizimining muhim vazifalaridan biridir. O'quvchi va talabalarning axborot madaniyatini shakllantirish o'rta maxsus, kasb –hunar ta'lim muassasalarida va keyingi bosqichda Oliy ta'lim muassasalarida “Informatika” va “Informatika” fanlarini o'qitish davrida amalga oshiriladi.

Shuning uchun, maktablarda, o'rta maxsus, kasb-hunar ta'lim muassasalarida “Axborot kommunikatsion texnologiyalari ”fanlarini zamonaviy pedagogik va axborot texnologiyalari asosida o'qitish muhim ahamiyat kasb etadi. Ushbu fan, birinchidan umumta'lim fanlardan biri bo'lib hisoblansa, ikkinchidan fan doirasida olingan bilim, malaka va ko'nikmalarni o'quvchilar bo'lg'usi kasbiy faoliyatlarida qo'llay olishlari kerak. Shuning uchun, maktab, o'rta maxsus, kasb-hunar ta'lim tizimidagi va oliy ta'lim tizimidagi har qanday umumta'lim fanning xususiyatlaridan kelib chiqqan holda, “Informatika” va “Dasturlash”, MBBT va dasturlash tillari shuningdek SQL tili kabi fanlaridan ham o'qitishning samaradorligini oshirish uchun kelgusida biz kasbiy faoliyatimizda o'quv maqsadlarimizni aniq va ravshan qo'yishimiz, o'qitishdan kuzatiladigan natijani belgilashimiz va o'quv jarayonida innovatsion ta'lim texnologiyalarini joriy etishimiz lozim.

Hozirgi davrda olimlar innovatsion ta'lim texnologiyalari tarkibini samarali usul va vositalar asosida tizimlashtirib, ularning tub mohiyatini ochib berishga erishmoqdalar. Ayniqsa, axborot-kommunikatsiya texnologiyalarning jadal rivojlanishi axborot-ta'lim muhitini shakllantirishga va innovatsion darslarni tashkil etishga keng imkoniyatlarni ochib bermoqda.

Bunday innovatsion ta'limning istiqbolli yo'nalishlaridan biri –bu ta'lim tizimining SQL tilini o'qitishda usulbiy xususiyatlar taxlilini qilib uni o'qitish

ishlab chiqish hamda ta'lim tizimiga joriy joriy etishdir. Ushbu mulohazalardan kelib chiqqan holda bitiruv malakaviy ishimizning mavzusini **“SQL tilini o‘qitishda usulbiy xususiyatlar taxlili”** deb belgiladik.

### **Bitiruv malakaviy ishi mavzusining dolzarbligi:**

Mavzuning dolzarbligi so‘nggi yillarda informatika va dasturlash fanlarini, jumladan SQL tili fanini o‘qitishda mustaqil ta’limga kuchayganligi, ishlab chiqarish va turli sohalarda ma’lumotlarni tez qayta ishlashga talabninig ortganligi sababli fani o‘qitishning yangi innovatsion uslublarini ishlab chiqarish, taxlil qilish va tadbiq etish masalalaridir.

### **Bitiruv malakaviy ishi ob’ekti va predmetining belgilanishi:**

**SQL tilini** o‘qitish darslari orqali, talim jarayonini taxlil qilish ishning ob’ekti hisoblanadi.

SQL tilini o‘qitishda usulbiy xususiyatlar taxlili asosida innovatsion uslublarni ishlab chiqish va ulardan foydalanib o‘qitilishida talabalar(o‘quvchilar) ta’lim - tarbiyasining takomillashib borishi va kompyuterli ta’lim jarayonida amal qiladigan bog‘lanishlarga oid prinsip, vosita, usullar tizimi bitiruv malakaviy ishning predmeti hisoblanadi.

### **Bitiruv malakaviy ishi maqsadi va vazifalari.**

Bitiruv malakaviy ishimizning maqsadi SQL tilini o‘qitishda usulbiy xususiyatlar taxlili qilib, innovatsion uslublarni ishlab chiqish va ulardan foydalanish natijasida ta’lim samaradorligiga erishishdan iborat.

Bitiruv malakaviy ishimizni `bajarish jarayonidagi asosiy **vazifalar** belgilab olindi:

1. Shu muammoga oid ilmiy adabiyotlar va ilmiy tadqiqot ishlarini tahlil etish, ilmiy-nazariy, amaliy asoslarini o‘rganish.

2. SQL tilini o'qitishda innovatsion uslublarni ishlab chiqish va ulardan foydalanishga doir ilg'or pedagogik tajribalarni o'rganish hamda umumlashtirish.
3. SQL tilini o'qitishda innovatsion uslublarni ishlab chiqish va ulardan foydalanishda ta'lim-tarbiya samaradorligiga erishishning shakl va omillarini ishlab chiqish hamda ularning samaradorligini tajriba-sinov orqali aniqlash.
4. SQL tilini o'qitishda usulbiy xususiyatlar tahlilini amalga oshirish.
5. MS Access MBBT si yordamida oddiy SQL so'rovlarni yaratish.
6. Xulosalar yaratish va amaliy tavsiyalar ishlab chiqish.

#### **Bitiruv malakaviy ishini asosiy masallari va farazalari.**

SQL tilini o'qitishda usulbiy xususiyatlarni taxlil qilishda innovatsion uslublarni ishlab chiqish va ulardan o'rinli foydalanilsa, ta'lim-tarbiya samaradorligiga erishish mumkin, agarda:

- mazkur bosqichda ta'limni tashkil etishga nisbatan yangicha yondashuv qaror topsa;
- SQL tilini o'qitishda innovatsion uslublarni ishlab chiqish va ulardan o'rinli foydalanilsa, unga qo'yiladigan talablarga erishilsa;
- SQL tilini o'qitishda fikrlarini amaliyotga tatbiq etilishiga erishilsa.

#### **Bitiruv malakaviy ishida qo'llanilgan uslublarning qisqacha tavsifi.**

SQL tilini o'qitishda usulbiy xususiyatlar tahlili muammosiga bag'ishlangan pedagogik va metodik adabiyotlardagi ma'lumotlarni nazariy jihatdan tahlil qilish; nazorat va baholash; tajriba-sinov; pedagogik monitoring; qiyosiy taqqoslash; tadqiqot natijalarini umumlashtirish, kuzatish, suhbat, test metodlari, olingan natijalarni qayta ishlash uchun matematik statistika metodlari qo'llanildi.

**Bitiruv malakaviy ishi natijalarining ilmiy yangiligi va amaliy ahamiyati.**

- SQL tilini o‘qitishda qo‘llaniladigan ta’lim metodlari ilmiy nazariy jihatdan tahlil qilindi;

- SQL tilini o‘qitishda innovatsion uslublarni ishlab chiqish va ulardan foydalanish jarayonidagi shart-sharoitlar, omillar aniqlandi hamda mazkur jarayonni tashkil etish, boshqarish va samaradorlikka erishish yo‘llari ilmiy-metodik jihatdan asoslandi;

- SQL tilini o‘qitishda innovatsion uslublarni ishlab chiqish va ulardan foydalanishning samarali vositalari, omillari belgilab olindi;

- SQL tili fanini o‘qitishda pedagogik va axborot texnologiyalaridan foydalanishda o‘qitishning innovatsion uslublarni ishlab chiqish va ulardan foydalanish va qo‘llash jarayonining ilmiy-metodik asoslari yaratildi va ilmiy-uslubiy tavsiyalar berildi.

- kompyuterli ta’limda o‘quv materialini o‘zlashtirish jarayonini tashkil qilishga o‘quvchi faoliyati nuqtai nazaridan yondashildi.

#### **Bitiruv malakaviy ishining o‘rganganlik darajasi:**

Bitiruv malakaviy ishni tayyorlash davomida Birinchi Prezidentimiz I.A.Karimov asarlaridan shuningdek R.M.Fedorov, A.Mixaylov, Sh.Nazirovlarning shu mavzuga oid kitoblardan hamda internet saytlaridan foydalanildi.

#### **Bitiruv malakaviy ishi tarkibining qisqacha tavsifi.**

Bitiruv malakaviy ishi kirish, uchta bob va xulosadan tashkil topgan. 45 ta nomdagi foydalanilgan adabiyotlar ro‘yxati bo‘lib, adabiyot va o‘quv qo‘llanmalar, ilmiy tadqiqot ishlari, internet ilova va resurslaridan hamda bitiruv malakaviy ishining asosiy xajmi 83 bet matn, 10 ta rasm va 1-ta jadvaldan tashkil topgan.



## **I BOB. SQL TILINI O‘QITISHDA USULBIY XUSUSIYATLARNI PEDAGOGIK VA INNOVATSIYA JIHATLARI.**

### **1.1. Axborot kommunikatsion texnologiyalari fanlarini o‘qitishda qo‘llaniladigan uslublar.**

Ta’lim tizimidagi har qanday fanning xususiyatlaridan kelib chiqqan holda, “Axborot kommunikatsion texnologiyalari” fanidan ham o‘qitishning samaradorligini oshirish uchun o‘quv maqsadlarimizni aniq va ravshan qo‘yishimiz, o‘qitishdan kuzatiladigan natijani belgilashimiz kerak. Aks xolda o‘qitiladigan fanlarga ehtiyoj noaniq bo‘ladi va uning natijasi talab darajasida bo‘lmashligi mumkin.

Chunki, insoniyat taraqqiyotining barcha bosqichlarida muhim muammo ta’lim maqsadi bo‘lib, shaxsni komil insonga aylantirish masalasi uning asosiy g‘oyasini tashkil etgan. Hozirgi kunda ta’lim maqsadlariga erishish uchun olimlar turli interfaol usullarni mashg‘ulotlarning xususiyatlaridan kelib chiqqan holda joriy etishni tavsiya etmoqdalar.

Mening bitiruv malakaviy ishim “Axborot kommunikatsion texnologiyalari” fanlaridan bo‘lgan SQL tilini o‘qitishda usulbiy xususiyatlar taxlil qilishga bag‘ishlanganligi bois, quyida men hozirgi kunda mazkur fan doirasida samarali hisoblangan interfaol usullarni tahlil etishni lozim topdim.

Dastavval “Axborot kommunikatsion texnologiyalari” fanidan ma’ruza interfaol usulini qo‘llash uslubiyotini ko‘rib chiqamiz. Hozirgi kunda ta’lim muassasalarida keng tarqalgan interfaol usullardan biri ma’ruza usulidir.

Ma’ruza – bu o‘qituvchi tomonidan bilimni bayon etishdir. Ma’ruzada keng hajmli bilim monologik yo‘l bilan bayon qilinadi. Ma’ruza metodida ko‘zlangan maqsad – o‘quvchilarda bilimni mustaqil egallash qobiliyatini hosil qilishdir.

Fan bo‘yicha o‘qitiladigan ma’ruza yangi bilimni o‘qitishda uning qismlarini umumlashtirishda, murakkab qonun-qoidalarni yakunlashda, axborot va u bilan bog‘liq bo‘lgan jarayonlarni tahlil etishda hamda muammoli masalalarni

o‘rganishda, o‘quv fanlari o‘rtasidagi o‘zaro aloqani o‘rganishda qo‘llashimiz kerak.

“Axborot kommunikatsion texnologiyalari” fanidan ma‘ruzaning mazmunli, samarali bo‘lishi uchun quyidagilarga e‘tibor berish lozim:

- aniq o‘ylab tuzilgan rejani o‘quvchilarga tanishtirish;
- rejaning har bir qismi bo‘yicha berilgan ma‘lumotni yakunlash;
- bayon o‘quvchilarga tushunarli va ko‘rgazmali, yorqin bo‘lishi;
- bayon o‘quvchilarda muhim o‘rinlarni bilib olishga qiziqish uyg‘otadigan bo‘lishi kerak.

Ma‘ruza mavzu bo‘yicha qilinadigan suhbat, seminar bo‘lib, bu o‘qituvchi rahbarligida o‘tkaziladigan mashg‘ulot o‘quvchilarda mustaqil fikrlash, mustaqil faoliyat ko‘rsatish, darslik bilan mustaqil ishlash, ijodiy fikrlash qobiliyatining rivoj topishiga yordam beradigan darajada bo‘lishi lozim.

Menyu interfaol usuli. Ushbu usul ham pedagogik texnologiyalarda keng qo‘llaniladi. Yakka tartibda yoki kichik guruhlar bilan ish olib borishni xohlagan ijodkor o‘qituvchiga “menyu” uslubini tavsiya etamiz. Bunda kichik guruh o‘quvchilariga aniq topshiriq beriladi. Misol, “Algoritmlarning turlari” tasnifi xilma-xildir. O‘qituvchi har bir guruh uchun alohida topshiriq tayyorlaydi:

- Algoritmning turlari.
- Algoritmlarning farqlari.
- Algoritmlarning xossalari.
- Algoritmlarning xussiyatlari.

Har bir guruh topshiriq oladi va 5 daqiqa davomida muhokama qilishadi, so‘ngra har bir guruhlardan bittadan vakil o‘qituvchi stoliga kelishadi va tayyor “menyu”ni olib qaytishadi. Bu uzun qirqilgan qog‘ozlar – “turlar” bo‘lib, ularda xilma-xil axborot turlari bayon qilingan bo‘ladi. Guruh vakili turlarni tanlab oladi va o‘z guruhiga olib keladi, keyin boshqa o‘quvchi shu ishni bajaradi va zaruriy materiallar yig‘ilguncha bu holat davom etadi. 10 daqiqa davomida topshiriq

muhokama qilinadi va rasmiylashtiriladi. Bu jadval qo'lda tayyorlanishi ham mumkin yoki javoblar og'zaki ham bo'lishi mumkin. Boshqa guruh vakillari savollar berishadi, o'qituvchi esa guruh ishlarini, sardor faoliyatini baholab boradi.

Aqliy hujum metodi. Ushbu usul hozirgi kunda juda keng tarqalgan zamonaviy o'qitish usullaridan biridir. Unda qatnashchilar birlashgan holda qiyin muammoni yyechishga harakat qiladilar va muammoni yyechish uchun shaxsiy g'oyalarini ilgari suradilar.

“Aqliy hujum”ning quyidagi qoidalari mavjud:

- Fikr va g'oyalar hech qanday cheklanmagan holda iloji boricha qattiqroq aytilishi lozim;
- Bildirilgan fikr va g'oyalar takliflar berish to'xtatilmaguncha muhokama qilinmaydi, baholanmaydi;
- Bildirilgan har qanday g'oya va fikrlar hisobga olinadi;
- Qancha va ko'p g'oya va fikrlar bildirilsa shuncha yaxshi;
- Bildirilgan g'oya va fikrlarni to'ldirish va yanada kengaytirish mumkin bo'ladi;
- Barcha aytilgan takliflar yozib boriladi va takliflarni bildirish uchun vaqt aniq belgilanadi.

“Aqliy hujum” usulini “Axborot kommunikatsion texnologiyalari” fani doirasida qo'llash yaxshi samara beradi.

O'z o'ringni top interfaol usuli. Ushbu usul muammoni o'rta tashlab, uni yyechish orqali pedagogik maqsadlarga erishish ta'minlanadi. Agar auditoriyada fan bo'yicha tortishuvlar vujudga kelsa, masalan qurilma yoki dasturning ishlash prinsipi bo'yicha, unda bu uslub yordamida muammoning yechimini topish talab etiladi. Unga ko'ra:

- Auditoriyaning qarama-qarshi burchaklariga ikkita plakat osiladi. Ularning biriga “roziman”, boshqasiga “rozi emasman” degan so'zlar yozilishi kerak. Plakatlarda o'sha qurilma yoki dasturning ishlash prinsipiga oid savollar

bo'yicha bildirilgan qarama-qarshi fikrlar yoziladi.

- Muammo bo'yicha fikrlar bildirilgach, o'quvchilar ichida munozara davomida o'z nuqtai-nazarini o'zgartirganlar va boshqa guruhga o'tuvchilar bo'lishi mumkin. Bunday hollarda ular o'z o'rnini o'zgartirish sabablarini asoslashi kerak bo'ladi.
- Opponentlardan ishtirokchilar ichidan muammo yuzasidan eng ishonchli fikr aytgan o'quvchilarni aniqlash so'raladi.

Klaster – “axborotlarni yozish” interfaol usuli. Bizning kuzatuvlarimiz natijasida “Axborot kommunikatsion texnologiyalari” fani doirasida yaxshi samara bergan usullardan biri bo'lib hisoblanadi. Ushbu interfaol usul ham hozirgi kunda keng tarqalgandir. “Klaster” – inglizcha so'z bo'lib, g'uncha, bog'lam ma'nosini anglatadi. Axborotlar klasterlarga ajratish interfaol pedagogik strategiya bo'lib, u ko'p variantli fikrlashni, rag'batlantirish uchun qo'llash mumkin. Asosan, u yangi fikrlarni uyg'otish va muayyan mavzu bo'yicha ancha fikr yuritishga chorlaydi. Klaster tuzish ketma – ketligi quyidagi bosqichlardan iborat:

- sinf yozuv taxtasi o'rtasiga katta qog'oz varag'iga asosiy so'z yoki gap yoziladi;
- sizni fikringizcha bu mavzuga tegishli bo'lgan so'zlar yoki gaplarni yozing o'tkazing;
- tushuncha va g'oyalar to'g'risidagi o'zaro bog'lanishni o'rgating.
- Eslagan variantlaringizning hammasini yozing.

Interfaol videousul. Ma'ruza mashg'ulotlarini tashkil etishda juda samarali hisoblangan usullardan biridir. Videousul – qurilmalarning ishlash prinsiplarini ko'rgazmali o'zlashtirishga asoslangan bo'lib, unda kineskop, kodoskop, proektor, kinoapparat, o'quv televidenie, videomagnitafon, axborotni monitorga aks ettiruvchi kompyuterlardan foydalanishimiz talab etiladi.

O'quv jarayonida videousuldan foydalanish, bizning quyidagi ta'lim-tarbiyaviy vazifalarni unumli yyechishingizni ta'minlaydi:

- yangi bilimlarni bayon etish, ya'ni juda sekin kechadigan jarayonlar bilan tanishish, bevosita kuzatish mumkin bo'lmagan, shuningdek, tez sodir bo'ladigan jarayonlar, bevosita kuzatishlar hodisalarning mohiyatini ochib bera olmagan holda qo'llaniladi;
- murakkab sxemalar, qurilmalar va kompyuterning ishlash tamoyillarini dinamikada tushuntirish;
- turli xil ish faoliyatini bajarish algoritmini tushuntirish;
- videohujjatlarni taqdim etish;
- mashq qilish ishlarini bajarish, sxemalarni modellashtirish, kerakli rejimlarni berish;
- o'quv-mashq va tadqiqot ishlarini olib borish uchun ma'lumotlar bazasini yaratish.

Ushbu interfaol usul yordamida "Axborot kommunikatsion texnologiyalari" fanidan ko'rgazmali ma'ruza matnlarini tashkil etishimiz yaxshi samara beradi.

Kubiklar interfaol usuli. Ushbu interfaol usul ham juda samarali va keng tarqalgandir. Kubiklar o'zgartirilayotgan kelajakka asoslanib mavzuni qo'rib chiqishda yordam beradigan o'qitish uslubi hisoblanadi.

Bu usul qurilayotgan sxemalarni turli tomondan, qadamba – qadam, osondan qiyinga tomon yo'nalishda tasavvur etish imkoniyatini beradi. Kubikning har bir tomoni muayyan topshiriqni ifodalaydi:

- Bu nima? Ko'rayotgan narsani o'lchamlari, shaklini tasavvur eting, eslang va yozma ravishda ta'riflang?
- Taqqoslang: u nimaga o'xshaydi, nimadan farq qiladi?.
- Assotsiatsiya. Ta'surotingizni izohlang? U sizni nima to'g'risida o'ylashga majbur qiladi? Xayolingizga nima keladi?
- Tahlil qiling. Bu sxema qanday yasalgan?, qanday elementlardan tuzilgan?, nimalardan tashkil topgan?
- Qo'llang. Bu sxema nimaga yaraydi? Uni qayerda qo'llash mumkin?

- “ha” va “yo‘q” larni asoslang. Bunda ishonchli dalillar va asoslovchi fikrlarni ayting.

Keys stadi interfaol usuli. Umumta’lim fanlarni o‘qitishda juda qulay hamda ishonchli usullardan biri bo‘lib hisoblanadi. Chunki, keys o‘quv jarayonida yuzaga kelgan aniq muammoli vaziyatning tavsilotidir. Keys – metod ishlab chiqarish masalalarini mashg‘ulotlarda tahlil qilish va hal qilish metodi bo‘lib, u zamonaviy ta’lim usullaridan biri hisoblanadi

Keys stadi- o‘qitish texnologiyasining mohiyati shundan iboratki, unda ishtirokchilarga haqiqiy hayotiy vaziyat bo‘yicha fikr yuritish taklif qilinib, bu vaziyat bayonida nafaqat amaliy masala ifodalanib qolmasdan, undagi muammoni yechish jarayonida o‘zlashtirilishi zarur bo‘lgan o‘quv materiali ham ifodalanadi. Vaziyatning bunday usulidagi tahlili, o‘quvchining bo‘lajak kasbiy faoliyati tajribasini oldindan egallashga ham kuchli ta’sir ko‘rsatadi, o‘qishga nisbatan qiziqish va motivlarning vujudga kelishiga asos bo‘lib xizmat qiladi. Keys usuli pedagogik o‘yin tavsifidagi ta’lim turi bo‘lib, o‘zida o‘yinni ijro etish bilan birga, intellektual yuksalish va nazorat malakasini ham mujassamlantiradi. Vaziyatning qulay tanlanganligi – uni olinayotgan buyum bilan belgilanadi. Vaziyatlar haqqoniy va shartli, me’yordagi yoki shartli, me’yordagi yoki favqulotli, nazoratli yoki nazoratsiz, kritik tavsiyaga ega bo‘lishi mumkin.

Keys texnologiyasini qo‘llash jarayonida o‘quvchilarni vaziyatni tahlil qilishga o‘rgatish maqsadida, vaziyatni quyidagi tartibda bayon qilish mumkin:

- Muammoli (muammolar shunday tarkib tuziladiki, u muammoli vaziyatlar majmuasini ishlab chiqishni talab qilib, ularning turlari, echilish usullariga muvofiq holda tanlanadi);
- Tizimli (vaziyatlar tarkibining tavsiflari va vazifalarini aniqlash);
- Sabab – oqibatli (vaziyatni kelib chiqargan sabablarni aniqlash);
- Tavsiyanomali (vaziyatni hal qilishda ishtirok etuvchilar amal qilishi zarur bo‘lgan tavsiyanomalar tayyorlash);

- Dasturiy – maqsadli (joriy vaziyatlar uchun tadbirlar dasturlarini ishlab chiqish);
- Tashhisli (vaziyatdagi faoliyat mazmunini tashxislash, uni modellashtirish va o‘ta maqbullashtirish).

O‘quv mashg‘ulotlarida keyslarni hal qilish algoritmi quyidagicha:

1. Topshiriqni berish (topshiriqni bajarish muddatini belgilash. Keysning yechimini baholash tizimi bilan tanishtirish, darsning texnologik modelini aniqlash).
2. O‘qituvchining kirish so‘zi. Asosiy muammoli savollarning qo‘yilishi.
3. O‘quvchilarni 4– 6 kishidan iborat mikroguruhlariga ajratish.
4. O‘quvchilarning mikroguruhdagi faoliyatini tashkil qilish (mikroguruhlarni nomlash, yetakchilarni va ekspert guruhini aniqlash).
5. Mikroguruhlardagi javoblar bilan tanishishni tashkil qilish.
6. Mikroguruhlararo munozara (polilog)ni tashkil qilish.
7. O‘qituvchining umumlashtiruvchi so‘zi, uning vaziyat yechimi to‘g‘risidagi fikri.
8. O‘quvchilarni ekspertlar tomonidan baholanishi.
9. O‘quvchilarning mashg‘ulot haqidagi fikrlari.

Ushbu usulga ko‘ra mashg‘ulot oxirida o‘qituvchining umumlashtiruvchi so‘zi hamda mashg‘ulot bo‘yicha xulosalar chiqariladi.

Debat interfaol usuli. Unga ko‘ra o‘rtaga tashlangan muammo asosida, muammoning yechimini topishda o‘quvchi o‘zgaralarni o‘z yondoshuvining to‘g‘riligiga ishonirishdir. O‘z fikrini aniq va mantiqiy bayon etish buning uchun ishonarli dalillar va xulosalar topish ko‘nikmalarini shakllantirishda debatlar o‘tkaziladi. Bu metoddan foydalanib dars o‘tishda mahalliy mutaxassislardan taklif etiladi.

Debatda qatnashish uchun iqtidorli o‘quvchilar tanlanadi va ular ikki guruhiga ajratiladi: rezolyusiyani qo‘llash quvvatlaydigan va ularga qarshi chiquvchi ishtirokchilar; ular debatlarini o‘tkazish qoidasini chuqur o‘zlashtirgan

bo‘lishlari kerak. Guruh sardori reglamentga rioya qilishni kuzatuvchi – yordamchini tayinlaydi.

Debatlarni o‘tkazish tartibi:

- O‘qituvchi va debat ishtirokchilari auditoriyadan joy olishadi. Qulay bo‘lishi uchun rezolyusiyani qo‘llab – quvvatlaydiganlar o‘qituvchining o‘ng tomoniga, qarshilar esa chap tomoniga joylashishgani ma’qul.
- O‘qituvchi qisqacha muhokama mavzusini va ikkala rezolyusiya variantlarini bayon qiladi, so‘zga chiquvchilar uchun vaqtni belgilaydi
- Dastlab rezolyusiyani qo‘llaydiganlarga so‘z beriladi va undan konstruktiv argumentlarni bayon qilish so‘raladi.
- Guruh o‘quvchilarining biriga so‘z beriladi, u fikrini argumentlar yordamida asoslaydi.
- Keyingi o‘quvchiga so‘z beriladi, u o‘z fikrlarini rezolyusiya bo‘yicha bayon qiladi. Debat ishtirokchilari navbatma-navbat chiqish qilgunlariga qadar davom etadi.
- Masala talashish davomida yangi argumentlarni keltirish mumkin emas.
- Ishtirokchilar shunday argumentlarni ko‘rsatish kerakki: ularning qarashlari muammo yechimiga zid ham bo‘lishi mumkin; barchani o‘ylashga majbur etsin; hayratda qolarli darajada ifodalanishi lozim.

Debat dars so‘ngida qarshi tomon qabul qilgan rezolyusiyaning baholashi kerak. O‘rtaga tashlangan qarashlarni hayotga qanchalik foydali yoki zararli ekanligi muhokama qilinadi.

“Muammoli vaziyat” interfaol usuli. “Muammoli vaziyat” usuli – o‘quvchilarda muammoli vaziyatlarning sabab va oqibatlarini tahlil qilish hamda ularning yechimini topish bo‘yicha ko‘nikmalarini shakllantirishga qaratilgan usuldir.

“Muammoli vaziyat” usuli uchun tanlangan muammoning murakkabligi o‘quvchilarning bilim darajalariga mos kelishi kerak. Ular qo‘yilgan muammoning



yechimini topishga qodir bo‘lishlari kerak, aks holda yechimni topa olmagach, o‘quvchilarning qiziqishlari so‘nishiga, o‘zlariga bo‘lgan ishonchlarining yo‘qolishiga olib keladi. “Muammoli vaziyat” usuli qo‘llanilganda o‘quvchilar mustaqil fikr yuritishni, muammoning sabab va oqibatlarini tahlil qilishni, uning yechimini topishni o‘rganadilar.

“Muammoli vaziyat” usulining tuzilmasi quyidagicha:

- Muammoli vaziyat tavsifini keltirish;
- Guruhlarga bo‘lish;
- Guruhlarning muammoli vaziyatning kelib chiqish sabablarini aniqlash;
- Guruhlarning muammoli vaziyatning oqibatlari to‘g‘risida fikr yuritishi;
- Guruhlarning muammoli vaziyatning yechimini ishlab chiqish;
- To‘g‘ri yechimlarni tanlash.

Muammoli vaziyat” usulining bosqichlari quyidagilardan iborat:

1. O‘qituvchi mavzu bo‘yicha muammoli vaziyatni tanlaydi, maqsad va vazifalarni aniqlaydi. O‘qituvchi o‘quvchilarga muammoni bayon qiladi.

2. O‘qituvchi o‘quvchilarni topshiriqning maqsad, vazifalari va shartlari bilan tanishtiradi.

3. O‘qituvchi o‘quvchilarni kichik guruhlariga ajratadi.

4. Kichik guruhlar berilgan muammoli vaziyatni o‘rganadilar. Muammoning kelib chiqish sabablarini aniqlaydilar va har bir guruh taqdimot qiladi. Barcha taqdimotdan so‘ng bir xil fikrlar jamlanadi.

5. Bu bosqichda berilgan vaqt mobaynida muammoning oqibatlari to‘g‘risida fikr-mulohazalarini taqdimot qiladilar. Taqdimotdan so‘ng bir xil fikrlar jamlanadi.

6. Muammoning yechishning turli imkoniyatlarini muhokama qiladilar, ularni tahlil qiladilar. Muammoli vaziyatni yechish yo‘llarini ishlab chiqadilar.

7. Kichik guruhlar muammoli vaziyatning yechimi bo‘yicha taqdimot qiladilar va o‘z variantlarini taklif etadilar.

8. Barcha taqdimotdan so‘ng bir xil yechimlar jamlanadi. Guruh o‘qituvchi bilan birgalikda muammoli vaziyatni yechish yo‘llarining eng maqbul variantlarini tanlab oladi.

“Muammoli vaziyat” usulining afzalliklari:

- O‘quvchilarda mustaqil fikrlash qobiliyatlarini shakllantiradi;
- O‘quvchilar muamoning sabab, oqibat va yechimlarini topishni o‘rganadilar;
- O‘quvchilarning bilim va qobiliyatlarini baholash uchun yaxshi imkoniyat yaratiladi;
- O‘quvchilar fikr va natijalarni tahlil qilishni o‘rganadilar.

“Muammoli vaziyat” usulining kamchiliklari:

- O‘quvchilarda yuqori motivatsiya talab etiladi;
- Qo‘yilgan muammo o‘quvchilarning bilim darajasiga mos kelishi kerak.

Ushbu interfaol usul hozirgi kunda texnik fanlardan o‘quv mashg‘ulotlarning sifatini oshirishda joriy etilgan.

Albatta, zamonaviy pedagogik va axborot texnologiyalarining jadal rivojlanishi, shuningdek bilimlar hamda ma’lumotlarning keskin oshib borishi ta’lim tizimida an’anaviy bo‘lmagan o‘qitishning yangi interfaol usullarini joriy etishni taqozo etmoqda. Har bir o‘qitishning interfaol usuli o‘zining yutuq va kamchiliklariga ega ekanligini bilishimiz va “Axborot kommunikatsion texnologiyalari” fanini o‘qitishda ularni joriy etish orqali o‘quvchilarni fan doirasida bilim, malaka va ko‘nikmalarini shakllantirish pedagogik maqsadlarga erishinning muhim omillaridan biri bo‘lib hisoblanadi.

## **1.2. SQL tilini o‘qitishda pedagogik-innovatsion usullar.**

Ta’lim-tarbiya tizimiga sezilarli samaralar beruvchi yangilik kiritish: yangi konsepsiyalar Davlat Ta’lim Standarti, alternativ o‘quv reja va dasturlari, o‘qitishning axborotli texnologiyalarini joriy etishning davlat dasturlari, jahon “Internet” tarmog‘iga chiqish va masofaviy ta’limni joriy etish bo‘yicha davlat rejasi va dasturi, shuningdek hozirgi zamon talablari asosidagi yangi o‘quv muassasalari va shu kabi yo‘nalishlarni yanada jonlashtirish ta’lim ta’lim tizimi oldida turgan muhim vazifalardan biri. SQL tilini o‘qitishda pedagogik-innovatsion usullarni qo‘llash ayni vazifani xal etishda olib borilayotgan ishlardan biri sanaladi. Quyida pedagogning fanni o‘qitishda joriy etilishi kerak bo‘lgan innovatsion usullarni taxlilini keltirib o‘taman:

### **1. Fani o‘qitishda mazmun-mohiyat eng muhim hisoblanib, u quyidagi mezonlar bilan bog‘liqdir:**

- o‘quv materiallaridagi bayon qilinadigan ma’lumotlarni axborot ko‘rinishiga keltirish va ularda uzviylik hamda izchillikni saqlab qolish;
- o‘quv materiallarini o‘zaro bog‘liq bo‘lgan modul bloklari shaklida ifodalashga erishish;
- o‘quv materiallarini o‘rganishga “Kompleks yondashuv” nomli tadqiqot usulidan foydalanishga erishish va undan muammoli hamda dasturiy ta’lim metodlaridan foydalanish;
- ta’lim-tarbiya jarayonida axborot muhitini yaratish borasida o‘rganilayotgan sohaning axborotli ta’minotini yaratishga yetishish;
- didaktik o‘yin(shu jumladan kompyuterli o‘yinlar va trenejerlar) orqali turli voqea va hodisalardagi jarayonlar kechishini ifodalovchi masalalar (muammolar)ni yechish;

- bu yoʻnalishdagi ijodiy izlanishlarni olib borishda "Ta'lim-fan-amaliyot" hamda nazariya bilan amaliyot uygʻunligiga eʼtibor berish;
- innovatsion usullar qoʻllangan holatdagi yutuq va kamchiliklarni baholab borishga erishish va kerakli joylarda tegishli tuzatishlar kirita olinishi lozim va shu kabilar.

## **2. Pedagogik innovatsiyaning muhim belgilari:**

- ta'lim-tarbiya jarayonini oldindan mavjud manbalarini hisobga olgan holda va fan-texnikaning eng soʻngi mavzuga(fanga) mos yutuqlarini eʼtiborga olib, ta'lim jarayonini loyihalashtirish va uning natijalarini prognoz(bashorat) qila olish;

- ta'lim-tarbiya jarayoniga "tizimiy yondashuv" tadqiqot usulini qoʻllashda, qaralayotgan tizimining elementlari va qism tizimlari orasidagi bogʻlanishlarni eʼtiborga olish va bular asosida talabalarning oʻqish-bilish faoliyatini tasvirlaydigan ta'lim-tarbiya jarayoni rejasini tuzish;

- ta'lim-tarbiyada maqsad aniq boʻlishi, uning diagnostikasi(tashhisi) va pronosi(bashorati) hamda talabaning oʻzlashtirishi sifatini xolis baholash mezonini boʻlishi kerak;

- ta'lim tizimi tuzilishi, tarkibi, mazmuni bir butun yaxlit boʻlib, ularning oʻzaro bogʻliqligi, bir-birini bosqichma-bosqich toʻldirilib borilishi taʼminlangan boʻlishi kerak;

- ta'lim va tarbiyada olib boriladigan faoliyatning optimal variantlarini taʼminlay olish lozim;

- ta'lim-tarbiya jarayonining texnik va informatsion taʼminoti yetarli boʻlishi lozim;

- yaratilgan pedagogik innovatsiya(PI)ni talabalarning ilmiy-uslubiy salohiyatini hisobga olgan holda oʻqitish;

- ta'lim sifatiga eʼtibor bergan holda shakllantiruvchi va yakuniy nazoratlar olish;

- rejalashtirilgan joriy, oraliq va yakuniy nazoratlarni o'tkazish va uning chuqur tahliliga erishish orqali talabalar o'zlashtirish monitoringi ko'rsatkichini doimo e'lon qilib borish;

- ta'lim-tarbiya samaradorligini aniqlash mezonini bo'yicha yakuniy nazorat va amaliyotga joriy etish bo'yicha uslubiy tavsiyalar berish.

### **3. Pedagogik innovatsiya imkoniyatlari:**

- "Fan – ta'lim - amaliyot" uzviyligini ta'minlaydi;

- PI ning maqsadi, mazmun-mohiyati, muhim belgilari, ularning uslubiy va axborotli ta'minotlari hamda imkoniyatlari bo'yicha batafsil ma'lumot berishga asoslar yetarli bo'ladi;

- PI dan foydalanishda o'rganiladigan manba(mavzu, fan, tarbiya yo'nalishlari va hakoza)ga qarab uni takomillashtirish yoki keraksiz qismlarini e'tiborga olmaslik mumkin;

- PI dan foydalanishda o'zgina o'zgartirishlar kiritish orqali o'xshash manbalarni o'rganish hamda foydalanish mumkin va shu kabilar.

### **4. Pedagogik innovatsiyaning vazifalari:**

- talabani "Kadrlar tayyorlash Milliy Dasturi" talablari asosidagi zamonaviy raqobatbardosh mutaxassis darajasiga yetkazishda zamonaviy ko'makchi bo'lish;

- o'qituvchi malakasini oshirish, talaba kasbiy fazilatlarini takomillashtirish yo'llarini ishlab chiqishdagi ijobiy ishlarini rivojlantirishga erishish;

- reyting, test usullaridan ta'lim-tarbiya jarayonida keng foydalanish;

- ilg'or pedagogik texnologiyalarni tahlil qilish, saralash va joriy etishga erishish hamda pedagogik samaralarini aniqlash;

- ta'lim dasturlarining invariantlarini, modifikatsiyalarini yaratish, ayniqsa elektron darsliklar hamda o'quv qo'llanmalari yaratishga erishish va ularni amaliyotga joriy etish;

- ta'lim-tarbiya jarayonining yangi axborotli va pedagogik texnologiyalarini qo'llash va unda sifat ko'rsatkichlari hamda samaradorlikni baholab borish;

- o'z mustaqil fikrini bayon qilishga sharoit yaratish va ularda tanqidiy fikrni bayon qilish imkoniyatlarini yaratish;

- "Ustoz- shogird" tizimidagi erkinliklarni rivojlantirish va shu kabilar.

### **5. Pedagogik innovatsiyaning rivojlanish jarayoni bosqichlari:**

- ta'lim-tarbiyani yanada rivojlantirish va takomillashtirishning zarurligini tushunishning paydo bo'lishi va o'zida paydo bo'lgan g'oyalar asosida ta'lim-tarbiyadagi muammo yechimini rivojlantirish va loyihalashtirish hamda tajriba-sinov ishlari modellarini yaratishga kirishish;

- shakllana boshlagan g'oyalar va ular asosidagi loyihalash-rivojlantirish ishlarini qo'yilgan maqsadlar bo'yicha aniq modellashtirish, yangicha tafakkur va tajriba rejalari qabul qilinib, qo'llab-quvvatlanadi, natijada aniq yangi g'oya paydo bo'lib, uning rivojlanishi uchun muhit va sinov-tajriba maydonchalaridagi ishlar natijalari bo'yicha axborot muhiti yaratiladi va joriy etish ko'lami aniqlanadi;

- qo'llanishi kerak bo'lgan pedagogik innovatsiyaning me'yoriy hujjatlari tayyorlanadi va DTS talablariga tegishli tuzatishlar kiritiladi, keng joriy etishga tavsiyalar beriladi va shu kabilar.

### **6. Pedagogik innovatsiya quyidagi tamoyillarga asoslanadi:**

- ta'lim – tarbiyaning yagona tizimdan iborat ekanligi, undagi uzluksiz ta'lim tizimi va tarkibining yaxlitligi, birligi hamda bir-biri bilan uzviy bog'langanligi;

- ta'lim-tarbiya jarayonini olib borishda optimal variantlarni topishga erishish;
- zamonaviylik, yani ta'lim-tarbiya uslubiyatini zamonaviy fan-texnika yutuqlari bilan boyitib borish;
- ilmiylik, yani ta'lim-tarbiya tizimini muntazam ravishda rivojlantirib borish maqsadida uzluksiz ilmiy izlanishlar va tadqiqotlar olib borish;
- moddiy-texnika bazani zamon talabi asosida yaratish, pedagogik jarayonda axborotlashtirish muhitiga e'tibor berish;
- axborot texnologiyalari va texnikalaridan foydalanish samaradorligini oshiruvchi elektron darslik va o'quv qo'llanmalarini ishlab chiqish, shuningdek masofaviy ta'lim va "Internet" dan foydalanish imkoniyatlarini kengaytirish;
- talabalar bilimni baholashni avtomatlashtirish hamda bilim monitoringini muntazam ravishda nazorat qilib borish va shu kabilar.

#### **7. Pedagogik innovatsiyani rivojlantirish mexanizmlari:**

- ta'lim muassasasida ijodiy muhit yaratish, bu boradagi tadbirkorlik va tashabbuskorlikni qo'llab-quvvatlash;
- ta'lim innovatsiyalari bo'yicha olib borilayotgan izlanishlarga shart-sharoitlar yaratish va ularning ommaviy tadbiriga ko'maklashish;
- ilmiy – izlanuvchanlikni rag'batlantirish, tanlovlar o'tkazish hamda ularni iqtisodiy jihatdan qo'llab-quvvatlash;
- istiqbolli ishlanmalarni, samarali ta'lim uslublari, shakllari, yo'llarini uzluksiz ta'lim tizimiga kiritishga ko'maklashish, yangiliklarni muvofiqlashtirish va to'plash va shu kabilar.

#### **8. Pedagogik innovatsiyalarni hayotga tadbir etishning tashkiliy asoslari:**

- **pedagogik innovatsiyalarni yig'ishda** innovatsiyalar bo'yicha aniq ma'lumotlar bazasini tashkil qilish va ta'lim-tarbiya sohasining me'yoriy hujjatlari bo'yicha ma'lumotlar omborini yaratiladi;

- **pedagogik innovatsiyani tanlashda** quyidagilarga e'tibor beriladi: pedagogik innovatsiyaning pedagogik samaradorligini muntazam ravishda o'rganib, tahlil qilib borish; pedagogik innovatsiya bo'yicha tajriba-sinov ishlarini tahlil qilib borish, umumlashtirish. Ommalashtirishlar bo'yicha tavsiyalar tayyorlash;

- **pedagogik innovatsiyani joriy qilishda** uni joriy etish jarayonini muntazam ravishda kuzatib borish, pedagogik innovatsiyaning rivojlanish mexanizmlariga alohida e'tibor berish va shu kabilar.

#### **9. Pedagogik innovatsiyadan foydalanishga uslubiy tavsiyalar:**

- ta'lim-tarbiya sohasidagi davlat me'yoriy hujjatlariga e'tibor berish;
- pedagogik innovatsiya sohasidagi ma'lumotlarni to'plash, saqlash, qayta ishlash va uzatishga tayyorlash orqali axborot holiga keltirish;
- pedagogik innovatsiyaning joriy etilishi bo'yicha axborot tizimini yaratish;
- pedagogik innovatsiya bo'yicha kompyuter tarmog'ida saydlar ochish va undan foydalanish ko'lamini kengaytirish va h.k.

#### **10. Pedagogik innovatsiyalarni joriy etishda bo'lajak o'qituvchilarda shakllanadigan innovatsion xislatlar:**

- **talabning ijodkorligi** xislati, bunga talaba o'zining intellektual qobiliyati hamda muntazam ravishda o'z ustida ishlashi orqali erishadi;
- **talabning kompyuter savodxonligi va axborot madaniyati yuksakligi** xislati;
- **talabning yangilikni joriy etishdagi faoliyatni oxirigacha yetkaza olish** xislati;



- **talabaning innovatsion faoliyatda ilg'or bo'lish xislati;**
- **talaba shaxsining g'ayrat-shijoatlik xislati;**
- **yetakchi kasbiy pedagogik mahoratga ega bo'la olishlik mahorati.**

Bu qayd etilgan pedagogik innovatsiyani ishlab chiqish joriy etishning zaruriy ketma-ketligini amalga oshirish orqali ta'lim tizimida sezilarli sifat o'zgarishlariga va yuqori samaradorlikka erishiladi. Albatta bu zaruriy ketma-ketlikni amalga oshirish borasida keng miqyosda katta ilmiy tadqiqot ishlari olib borilishi mumkin.

Ta'lim jarayoniga pedagogik innovatsiyalarni ishlab chiqish va qo'llash yaqin o'ttiz yillik tarixga ega. Bu davrda rivojlangan xorijiy davlatlarda yangi pedagogik texnologiyalar, yangi axborot texnologiyalari ishlab chiqildi. Bu borada respublikamizda ham sezilarli ishlar olib borilmoqda. Shuningdek, Guliston davlat universitetida ham yuqorida qayd etilgan pedagogik innovatsiyalarni ishlab chiqish joriy etishning zaruriy ketma-ketligi bo'yicha diqqatga sazovor ishlar qilinmoqda. Jumladan, universitetda "Innovatsion uslublarni yaratish va joriy etish markazi" tashkil etilgan. Uning tarkibida quyidagi bo'limlar mavjud:

- innovatsion uslublar ishlab chiqish, "Kadrlar tayyorlash milliy dasturi" asosidagi ilg'or pedagogik texnologiyalarni va ijodkor o'qituvchilar ish tajribalarini o'rganish, ularni to'plash va ommalashtirish bo'limi;
- iqtidorli talabalar bilan ishlash bo'limi;
- pedagogik innovatsiyalar asosidagi o'quv-uslubiy qo'llanmalarni tayyorlash va nashr qildirish bo'limi va shu kabilar.

Demak, pedagogik innovatsiyalar keng qamrovli, ko'p qirrali, murakkab va ijodiy tashkiliy-pedagogik faoliyat bo'lib, uning yordamida ta'lim-tarbiya jarayonining istiqbolli va samarali uslubiyatini yaratish mumkin hamda "Kadrlar tayyorlash milliy dasturi" da qayd etilgan raqobatbardosh zamonaviy mutaxassislarni tayyorlashni kafolatlaydi.

Quyidagi jadvalda ta'lim-tarbiya jarayoni yo'nalishida an'anaviy va innovatsion yondashuvlarning mohiyati keltirilgan.

1-jadval

<b>Tarbiya jarayonining yo'nalishi</b>	<b>An'anaviy yondashuv</b>	<b>Innovatsion yondashuv</b>
1. Tarbiyachi-o'qituvchining vazifasi	O'rgatish, ko'rsatish, tayyorlash, mashqni ko'rib chiqish.	Umumiy yo'nalish berish va yakuniy xulosaning to'g'riligini ko'rib chiqish.
2. Maqsadning qo'yilishi.	Tarbiyaviy tadbirlar to'g'risida umumiy ma'lumot berish.  Shaxsning xulq-atvori va madaniyatiga ijobiy ta'sir ko'rsatish orqali orzu istaklarni ro'yobga chiqarish.	Shaxsni o'z-o'zini ochish, o'zini namoyon etish, o'z-o'ziga o'lgan munosabat orqali o'zining individualligini ko'rsata bilish, ezgu fikr asosida o'z-o'zini rivojlantirishga erishish.
3. Vazifaning belgilanishi.	Maqsadga oid ilmiy amaliy manbalar bilan tanishtirish va tegishli xulosalar chiqarish hamda topshiriqlar berish.	Mavzu yuzasidan ilmiy-amaliy manbalar mazmunini aniqlash, tahlil etish va tegishli tavsiyalar ishlab chiqish.
4. Metodlar.	Suhbat, savol-javob, kuzatish, mashq va boshqalar.	Zanjir, shaxsiy fikr, aqliy hujum, debat va boshqalar.
5. Vositalar.	Chizma, jadval, darslik va	Elektron darslik va

	qo'llanmalar.	qo'llanmalar, multimedia materiallari.
6. Motivatsiya.	Tashqi topshiriqlarni puxta bajarish va baholash. Yangi topshiriqlar olishga hozirlik ko'rish.	Refleksiv: o'z-o'zini tahlil etish, o'zidagi kamchiliklarni isloh etish, o'zining faoliyatiga o'zi baho berish.
7. Munosabat.	Sub'ekt-ob'ekt	Sub'ekt- sub'ekt
8. Ustuvorlik.	Faol harakatga intilish.	Ijtimoiy faollikka erishish.
9. Mashg'ulotning turi.	Guruh(sinf jamoasi) savol-javob, hikoya, rivoyat aytish, mashq bajarish va boshqalar.	Kichik guruhlar va yakka tartibda evristik suhbat, rolli va harakatli o'yinlar, krossvord yechish, ijodiy mashqlar, loyiha tayyorlash va boshqalar.
10. Natija.	Reja-dastur doirasida faoliyat yuritish.	Ilmiy-ijodiy va innovatsion yondashuv asosida faoliyat olib borish.

## II-BOB. SQL TILINI FANINING TUZILISHI VA BO'LIMLARI MAZMUNI.

### 2.1 SQL tili va ma'lumotlar tiplari

#### *Ma'lumotlar tiplari.*

SQL tilida quyidagi asosiy ma'lumotlar tiplari ishlatilib, ularning formatlari har xil MBBT lar uchun farq qilishi mumkin:

- INTEGER - butun son (odatda 10 tagacha qiymatli raqam va ishora).
- SMALLINT - "qisqa butun" (odatda 5 tagacha qiymatli raqam va ishora).
- DECIMAL(p, q) - o'nli son, p raqam va ishoradan iborat ( $0 < p < 16$ ). O'nli nuqtadan so'ng raqamlar soni q orqali beriladi ( $q < p$ , agar  $q = 0$  bo'lsa, tashlab yuborilishi mumkin).
- FLOAT - haqiqiy son 15 ta qiymatli raqam va butun darajadan iborat. Daraja MBBT tipi bilan aniqlanadi (masalan, 75 yoki 307).
- CHAR(n) - uzunligi o'zgarmas, n ga teng bo'lgan simvolli qator ( $0 < n < 256$ ).
- VARCHAR(n) - uzunligi o'zgaruvchi, n simvoldan oshmagan simvolli qator ( $n > 0$  va har xil MBBT larda har xil lekin 4096 dan kam emas).
- DATE - maxsus komanda orqali aniqlanuvchi formatdagi sana (Subase da ko'zda tutilgan bo'yicha yy/mm/dd); sana maydonlari bizning eramizdan oldin bir necha mingyilliklardan boshlanuvchi va bizning eramiz beshinchi- o'ninchi mingyilligi bilan cheklangan haqiqiy sanalarni o'z ichiga olishi mumkin.
- TIME - maxsus komanda orqali aniqlanuvchi formatdagi vaqt (ko'zda tutilgan bo'yicha hh.mm.ss).
- DATETIME - sana va vaqt kombinatsiyasi. (Sybase da TIMESTAMP ).

**MONEY** -maxsus komanda orqali aniqlanuvchi formatdagi pul. Format o‘z ichiga pul birligi simvoli (\$, rub, ...) va uning joylashuvi (suffiks yoki prefiks), kasr qism aniqligi va pul qiymatini ko‘rsatish shartlarini oladi.

## 2.2 SQL tili jadvallar bilan ishlash

### *Jadvallarni yaratish.*

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 >**

**( <column name> <data type>[(<size>)],**

**<column name> <data type>[(<size>)], ... );**

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

Bundan buyon quyida keltirilgan 3 ta jadvaldan iborat ma’lumotlar bazasini ko‘ramiz.

Jadval Sotuvchilar (Salepeople):

Snu	Sname	City	Comm
m			
11	Peel	London	0.12
12	Serres	San Jose	0.13

14 Motika London 0.11

SNum – har bir sotuvchi unikal nomeri,

SName – sotuvchi nomi,

City – sotuvchi adresi ( shahar ),

Comm – sotuvchilarning oʻnli shakldagi komission foydasi.

Jadval Buyurtmachilar (Customers):

Cnum	Cname	City	Rating	SNum
21	Hoffman	London	100	11
22	Giovanni	Rome	200	13
23	Liu	SanJose	200	12

CNum – har bir buyurtmachi unikal nomeri,

CName – buyurtmachi nomi,

City – buyurtmachi adresi ( shahar ),

Rating – buyurtmachining boshqalardan ustunlik darajasini koʻrsatuvchi kod ( reyting ),

SNum – shu buyurtmachiga tayinlangan sotuvchi nomeri.

Jadval Buyurtma (Orders):

Onum	AMT	Odate	CNum	Snum
38	4723.00	1990/10/0	26	11

310 1309.95 1990/10/0 24 12

6

ONum – har bir sotib olish unikal nomeri,

AMT – sotib olish summasi qiymati,

ODate – sotib olish sanasi,

CNum – sotib oluvchi buyurtmachi nomeri,

SNum – sotuvchining nomeri.

Misol uchun sotuvchilar jadvalini yaratishni ko‘rib chiqamiz:

**CREATE TABLE Salepeople**

( **SNum integer,**

**SName char (10),**

**City char (10),**

**Comm 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:

**DROP TABLE < table name >;**

Masalan: **DROP 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 <column name>  
<data type> <size>;
```

**Masalan:**

```
ALTER TABLE Salepeople ADD Phone CHAR(7);
```

### **Jadvallar uchun cheklanishlar**

#### ***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. Cheklanishlar ikki asosiy turi mavjud: - ustun va jadval cheklanishlari. Ularning farqi shundaki ustun cheklanishi stolbtsa 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:

```
CREATE TABLE < table name >  
( <column name> <data type> <column constraint>,  
<column name> <data type> <column constraint> ...  
<table constraint> ( <column name>
```



[, <column name> ]... );

Maydonga bo'sh (NULL) qiymatlar kiritilishi 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),**

**comm decimal);**

Ko'p xollarda ustunga kiritilgan qiymatlar bir biridan farq qilishi kerak. Agar ustun uchun UNIQUE cheklanishi o'rnatilsa, bu ustungsha mavjud qiymatni kiritishga urinish rad etilapdi. Bu cheklanish bo'sh bo'lmaydigan (NOT NULL) deb e'lon qilingan maydonlarga qo'llanishi mumkin.

Masalan:

**CREATE TABLE Salepeople**

**( SNum integer NOT NULL UNIQUE,**

**SName char (10),**

**City char (10),**

**Comm decimal);**

Unikalligi talab qilinadigan maydonlar(birlamchi kalitlardan tashqari) kandidat 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 biriktirilgan. Ya'ni Buyurtmachilar jadvalida buyurtmachi nomeri (cnum) va sotuvchi nomeri (snum) kombinatsiyasi unikal bo'lishi kerak. Bu cheklanishni UNIQUE (cnum, snum) yordamida, Customers jadvalini yaratishda kiritish mumkin. Bu ustunlar uchun NOT 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, faqat 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 NOT NULL PRIMARY KEY,**

**SName char (10),**

**City char (10),**

**Comm 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 Namefield**

**( firstname char (10) NOT NULL,**

**lastname char (10) NOT 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 ifodadan iboratdir.

Misol: Salepeople jadvali Comm 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),**

**Comm decimal CHECK ( Comm < 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 London, Barselona, San Xose va Nyu York bo'lsin.

**CREATE TABLE Salepeople**

**( SNum integer NOT NULL PRIMARY KEY,**

**SName char(10) NOT NULL UNIQUE,**

**City char(10) CHECK (City IN ('London', 'New York', 'San Jose', 'Barselona'))),**

**Comm decimal CHECK ( Comm < 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),**

**Comm decimal,**

**CHECK (Somm < .15 OR City = 'Barcelona')));**

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

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 'New York',  
  Comm    decimal CHECK ( Comm < 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 bita maydondan iborat bo'lishi shart emas. Birlamchi kalit kabi, tashqi kalit bita modul sifatida qayta ishlanuvchi bir necha maydonlarga ega bo'lishi mumkin. Maydon tashqi kalit bo'lsa ilova qilayotgan jadval bilan ma'lumi usulda bog'liqdir. Tashqi kalit har bir qiymati (satri), ajdod kalitning bitta va faqat bitta qiymatiga (satriga) 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 kalitmaydonlariga 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) );**

FOREIGN KEY cheklanishidan jadval o'ki ustun cheklanishi sifatida foydalanganda ajdod kalit ustunlarini ko'rsatmaslik musmkin, 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 majud 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 konkret qiymatga almashtirish mumkin.

## **2.3 Maydonlarni qiymatlarini kiritish, o'chirish va o'zgartirish**

### ***Qiymatlarni kiritish.***

Hamma satrlar SQLda INSERT komandasi yordamida kiritiladi. INSERT quyidagi formatlar 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] ...)]
```

```
ostkiso‘rov;
```

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

```
INSERT INTO Salepeople
```

```
VALUES (11, 'Peel', 'London', .12);
```

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

```
INSERT INTO Salepeople (Sname, Comm, 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 ifodasini (oldingi misoldagi) mos so‘rovga almashtiringiz kerak:

```
INSERT INTO Londonstaff
```

```
SELECT * FROM Salespeople
```

```
WHERE City = 'London';
```

*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-condition];**

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;**

*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 column = expression [, column = expression] ...**

**[WHERE search-condition]**

bu erda expression - bu ustun | ifoda | konstanta | o‘zgaruvchi.

Ikkinchi variant:

**UPDATE <table name>**

**SET column = expression, ...**

**[ FROM table-list ]**

**[ WHERE search-condition ]**

Masalan, hamma buyurtmachilar baxosini 200 ga o'zgartirish uchun quyidagini kiritishingiz mumkin:

**UPDATE Customers**

**SET Rating = 200;**

Ma'lum satrlarni o'zgartirish uchun DELETE dagi kabi predikatdan foydalanish kerak. Masalan Peel ( imeyushogo SNum=11 ) sotuvchining hamma buyurtmachilari uchun bir xil o'zgartirishni quyidagicha kiritish mumkin vьpolnit:

**UPDATE Customers**

**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 = 'Gibson', City = 'Boston', Comm = .10**

**WHERE SNum = 14;**

Siz UPDATE komandasining SET jumlasida skalyar ifodalardan o'zgartirilayotgan maydon ifodasiga qo'shgan xolda foydalanishingiz mumkin. Masalan:

**UPDATE Salepeople**

**SET Comm = Comm \* 2;**

## **2.4 SELECT so'rov operatori**

SELECT operatori MB 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'z ichiga 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 guruhlarga ajratilgan natijaviy so'rov yaratishga imkon beradi.

HAVING ifodasidan guruhlarni 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 *(vse!) FROM OFFICES
```

*SELECT yordamida ma'lumotlarni tanlash*

SELECT Operatori albatta "qaytariluvchi ustunlar ro'yxati " ni o'z ichiga 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'z ichiga olishi mumkin.

**SELECT FILED1, (FIELD2 - FIELD3) "CONST" ... FROM ...**

FROM jumlasini "jadval spetsifikatorlari", Ya'ni so'rovni tashkil qiluvchi jadvallar nomini o'z ichiga 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**

operatori **SELECT** 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 ortig'i yoki kami bilan bajarilganligi ro'yxati.

**SELECT CITY, REGION, (SALES-TARGET) FROM OFFICES**

Har bir xizmatchi uchun rejadagi sotuvlar xajmini haqiqiy sotuvlar xajmining 3% foiziga oshirish!

**SELECT NAME, QUOTA, (QUOTA +((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 xajmlari ro'yxatini chiqaring.

**SELECT CITY, 'Has sales of', SALES FROM OFFICES**

'Has sales of' 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 xajmi rejadan oshgan ofislarni ko'rsating.

**SELECT CITY, SALES, TARGET FROM OFFICES WHERE SALES  
> TARGET**

**Zdes WHERE SALES > TARGET, znachit, esli SALES bolshe  
TARGET!**

Identifikatori 105 ga teng bo'lgan xizmatchi nomi haqiqiy va rejadagi  
sotuvlar xajmini ko'rsating:

**SELECT SALES, NAME, QUOTA FROM SALESREPS WHERE  
EMPL\_NUM = 105**

**Zdes WHERE EMPL\_NUM = 105, oznachaet, EMPL\_NUM ravno  
105!**

Agar izlash sharti ROST(TRUE), bo'lsa qator natijaviy to'plamga  
qo'shiladi, agar izlash sharti LOJNO(FALSE), 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, kerakligimni  
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. Shablona moslikni tekshirish. Ustundagi satrli qiymat shablona  
mos keladimi.
5. NULL qiymatga tenglikka tekshirish.

Solishtirish amallari maydon va konstantalarni solishtirish amallarini o‘z ichiga olishi mumkin: 1988 yilgacha ishga olingan hamma xizmatchilar nomlarini toping.

```
SELECT NAME FROM SALESREPS
```

```
WHERE HIRE_DATE <TO_DATE('01.06.1988','DD/MM/YYYY')
```

TO\_DATE('01.06.1988','DD/MM/YYYY') - PL/SQL Oracle sana bilan ishlash standart funktsiyasi.

Yoki arifmetik ifodalarni o‘z ichiga olishi mumkin: Haqiqiy sotuvlar xajmi 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!

## **2.5. 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 xajmlari rejaning 80 dan 120 protsentigacha bo'lgan diapazonga tushmaydigan xizmatchilar ro'yxatini chiqarish.

```
SELECT NAME, SALES, QUOTA  
FROM SALESREPS  
WHERE SALES NOT BETWEEN (0.8 * QUOTA) AND (1.2 *  
QUOTA)
```

Ifoda IN to'plamga tegishlilikni tekshiradi. Komanda sintaksisi quyidagicha:

```
--- tekshirilayotgan ifoda ----- IN ----- (-- const -----)  
- NOT - -- , -----
```

1990 yil iyun oyining har xil kunlarida qilingan hamma buyurtmalarni aniqlash.

```
SELECT ORDER_NUM, ORDER_DATE, AMOUNT  
FROM ORDERS  
WHERE ORDER_DATE IN (TO_DATE('14.06.1990','DD/MM/YYYY'),  
TO_DATE('08.06.1990','DD/MM/YYYY'),  
TO_DATE('29.06.1990','DD/MM/YYYY'),  
TO_DATE('04.06.1990','DD/MM/YYYY'))
```

Sanalar bilan shu tarzda ishlanadi.

To'rtta konkret 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:

```
--- IMYA STOLBTSA ----- LIKE (shablon) -----
```

-

## NOT

## ESCAPE (o'tkazish nomi)

Sodda so'rov bajaramiz: "Apelsin" kompaniyasi uchun kredit limitini ko'rsatish:

```
SELECT COMPANY, CREDIT_LIMIT  
FROM CUSTOMERS  
WHERE COMPANY = 'Apelin'
```

Quyidagicha '%' shablonli LIKE operatorini qo'llaymiz:

```
SELECT COMPANY, CREDIT_LIMIT  
FROM CUSTOMERS  
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 CUSTOMERS  
WHERE COMPANY LIKE '%gan'
```

Ba'zida '%' shablona o'rniga '\*' belgisi qo'llanadi znak, masalan v MS SQL uchun, c:\>dir \*.exe!

Agar faqat bitta simvol ixtiyoriy bo'lsa '\_' shablona qo'llanadi. Masalan:

```
SELECT COMPANY, CREDIT_LIMIT  
FROM CUSTOMERS  
WHERE COMPANY LIKE 'Ap_lsin'
```

### *Operator IS NULL*

SELECT operatori uchun NULL qiymati bilan ishlash qoidalarini ko'ramiz.

Konkret misol 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 biriktirilgan 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 xajmi rejadagidan yoki \$300.00 dan kam bo'lgan xizmatchilarni aniqlash

**SELECT NAME, QUOTA, SALES**

**FROM SALESREPS**

**WHERE SALES < QUOTA OR SALES < 300.0**

Sotuvlari haqiqiy xajmi rejadagidan va \$300.00 dan kam bo'lgan xizmatchilarni aniqlash

**SELECT NAME, QUOTA, SALES**

**FROM SALESREPS**

**WHERE SALES < QUOTA AND SALES < 300.0**

Sotuvlari haqiqiy xajmi rejadagidan kam, lekin \$150.00 dan ko'p bo'lgan xizmatchilarni aniqlash

**SELECT NAME, QUOTA, SALES**

**FROM SALESREPS**

**WHERE (SALES < QUOTA) 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
--------	--------

-----  
**TRUE OR TRUE -> TRUE**

**FALSE OR TRUE -> TRUE**  
**TRUE OR FALSE -> TRUE**  
**FALSE OR FALSE -> FALSE**  
**NULL OR TRUE -> TRUE**  
**TRUE OR NULL -> TRUE**  
**FALSE OR NULL -> NULL**  
**NULL OR FALSE -> NULL**  
**NULL OR NULL -> NULL**

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'shma izlash operatorlarining har biri o'z ustivorligiga ega. Eng yuqori ustivorlik NOT ga tegishli, undan so'ng AND va oxirida OR!

## 2.6. So‘rovlarda guruhlash va funksiyalar

### *Agregat funksiyalar qo‘llanishi*

Agregat (yoki STATIK) funksiyalar, 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 bajarilishi o‘rtacha protsenti qancha?

```
SELECT AVG(100 * (SALES/QUOTA))
```

```
FROM SALESREPS
```

Yana bir shakl: Sizni kompaniyangizda reja bajarilishi o‘rtacha protsenti qancha?

```
SELECT AVG(100 * (SALES/QUOTA)) PROCENT
```

```
FROM SALESREPS
```

Bu xolda ustun nomi ma’noliroq, lekin bu asosiysi emas. Ustunlar summasini hisoblab ko‘ramiz. SUM() funksiyasini qo‘llaymiz, ustun sonli bo‘lishi kerak! Masalan, quyidagicha: Kompaniya xizmatchilari sotuvlar xajmi rejadagi va haqiqiy o‘rta qiymati qanchaga teng?

```
SELECT SUM(QUOTA), 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 PRODUCTS
```

```
WHERE MFR_ID = 'ACI'
```

Ekstremumlarni topish funksiyalari yani MIN(), MAX() funksiyalarini ko‘ramiz. Bu funksiyalar sonli ustunlar, sanalar va satrli o‘zgaruvchilar bilan ishlaydi. Eng sodd qo‘llanishi sonlar bilan ishlash.

Masalan quyidagi so‘rov beramiz: Eng ko‘p va kam sotuvlar rejadagi xajmi?

```
SELECT MIN(QUOTA), MAX(QUOTA)  
FROM SALESREPS
```

Bu sonlarni o'z ichiga 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 CUSTOMERS
```

Yana bir so'rov: Qancha xizmatchi rejani ortig'i bilan bajardi?

```
SELECT COUNT(NAME)  
FROM SALESREPS  
WHERE SALES > QUOTA
```

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/ISO qoidalariga ko'ra "agregat funktsiyalar NULL qiymatni e'tiborga olmaydi"! Quyidagi so'rov ko'ramiz:

```
SELECT COUNT(*), COUNT(SALES), COUNT(QUOTA)  
FROM SALESREPS
```

Jadval bitta lekin so'rovdagi qiymatlar har xil. Chunki QUOTA maydoni-NULL qiymatni o'z ichiga oladi. COUNT funksiyasi COUNT(maydon) ko'rinishda bo'lsa NULL qiymatni e'tiborga olmaydi, COUNT(\*) bo'lsa satrlar umumiy sonini hisoblaydi. MIN(), MAX() funksiyalari ham NULL qiymatni e'tiborga olmaydi, lekin AVG(), SUM() - NULL qiymat mavjud bo'lsa chalkashtiradi. Masalan, quyidagi so'rov:

```
SELECT SUM(SALES), SUM(QUOTA), (SUM(SALES) -  
SUM(QUOTA)), (SUM(SALES - QUOTA))  
FROM SALESREPS
```

(SUM(SALES)-SUM(QUOTA)) va (SUM(SALES-QUOTA)) ifodalari agar QUOTA, 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() funksiyalari NULL qaytaradi! Funktsiya COUNT() nol qaytaradi!
3. Agar ustunda qiymatlar bo'lmasa (ya'ni ustun bo'sh), AVG(), SUM(), MIN(), MAX() funksiyalari NULL qaytaradi! Funktsiya COUNT()nol qaytaradi!
4. Funktsiya COUNT(\*) satrlar sonini hisoblaydi va ustunda NULL qiymat bor yo'qligiga bog'liq emas! Agar ustunda satrlar bo'lmasa, bu funktsiya nol qaytaradi!

DISTINCT funksiyasini agregat funksiyalar 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, oraliq 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 xizmatchaga 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'z ichiga 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 ofis uchun sotuvlarning rejalashtirilgan xajmi diapazoni qancha?

```
SELECT REP_OFFICE, MIN(QUOTA), MAX(QUOTA)
```

```
FROM SALESREPS
```

```
GROUP BY REP_OFFICE
```

Yana bir so'rov: Har bir ofisda 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), 'CUSTOMERS FOR  
SALESREPS', CUST_REP
```

**FROM CUSTOMERS**

**GROUP BY CUST\_REP**

Bu erda 'CUSTOMERS FOR SALESREPS' psevdó maydonning ishlatilishiga 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 qilib GROUP BY ifodasi SELECT ni guruhlarni qayta ishlashga majbur qiladi.

MS SQL serverida COMPUTE 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 o'tgan 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'z 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/ISO 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 foydalanigsh 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 xajmini hisoblash.

```
SELECT CITY, SUM(QUOTA), 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. Yana shunga e'tibor berish kerakki HAVING ifodasida agregat funktsiyalardan foydalaniladi. So'rov bajarilishini 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 xajmlari 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_ON_HAND, SUM(QTY)  
FROM PRODUCTS, ORDERS  
WHERE MFR = MFR_ID
```

**GROUP BY MFR\_ID, PRODUCT\_ID, DESCRIPTION, PRICE,  
QTY\_ON\_HAND**

**HAVING SUM(QTY) > (0.75 \* QTY\_ON\_HAND)**

**ORDER BY QTY\_ON\_HAND DESC**

HAVING uchung qo'shimcha chegaralar mavjuddir. Bu ifoda juda bo'lmasa bita agregat funktsiyani o'z ichiga olishi kerak. Chunki WHERE alohida 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 ifodasini GROUP BY siz qo'llash mumkin. Bu xolda natija hamma satrlardan iborat guruh deb qaraladi, lekin amalda bu kam qo'llanadi.

## **2.7. Foydalanuvchilar va ular imtiyozlari**

### ***Foydalanuvchilar.***

SQL muhitida har bir foydalanuvchi maxsus identifikatsiton nom, murojlat identifikatoriga (ID) ega. Ma'lumotlar bazasiga yuborilgan komanda ma'lum foydalanuvchi bilan yoki boshqacha aytganda maxsus murojaat identifikatori bilan bog'lanadi. SQL ma'lumotlar bazasida ID ruxsat – bu foydalanuvchi nomi va SQL komanda bilan bog'langan murojaat identifikatoriga ilova qiluvchi maxsus kalit so'z USER dan foydalanishi mumkin.

Registratsiya bu kompyuter tizimiga kirish huquqini olish uchun foydalanuvchi bajarishi kerak bo'lgan protseduradir. Bu protsedura foydalanuvchi bilan qaysi murojaat ID si bog'lanishini aniqlaydi. Odatda har bir ma'lumotlar bazasidan foydalanuvchi o'zining ID siga ega bo'lishi kerak va registratsiya jarayonida haqiqiy foydalanuvchiga aylanadi. Lekin ko'p masalalarga ega

foydalanuvchilar bir necha murojaat ID lari bilan registratsiyadan o'tishlari 1yoki bir necha foydalanuvchi bitta murojaat ID sidan foydalanishlari mumkin.

### ***Imtiyozlar.***

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

Foydalanuvchiga tayinlash mumkin bo'lgan imtiyozlar:

- **SELECT** Bu imtiyozga ega foydalanuvchi jadvallarda so'rovlar bajarishi mumkin.
- **INSERT** Bu imtiyozga ega foydalanuvchi jadvalda INSERT komandasini bajarishi mumkin.
- **UPDATE** Bu imtiyozga ega foydalanuvchi jadvalda UPDATE komandasini bajarishi mumkin. Bu imtiyozni jadvalning ayrim ustunlari uchun cheklab qo'yishingiz mumkin.
- **DELETE** Bu imtiyozga ega foydalanuvchi jadvalda DELETE komandasini bajarishi mumkin.

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

Bundan tashqari siz ob'ekt nostandart imtiyozlarini uchratasiz, masalan INDEX (INDEKS) –jadvalda indeks yaratish huquqini beruvchi, SYNONYM (SINONIM)- ob'ekt uchun sinonim yaratish huquqini beruvchi va ALTER (IZMENIT)- jadvalda ALTER TABLE komandasini bajarish xuquqini beruvchi. SQL mexanizm foydalanuvchilarga bu imtiyozlarni GRANT komandasi yordamida beradi.

### ***GRANT Komandasi***

GRANT komandasining 4 formati mavjud bo'lib, ulardan biri konkret ob'ekt ustidan, konkret foydalanuvchilarga konkret imtyozlar berish bo'lib, quyidagi ko'rinishga ega:

```
GRANT privilege ON [creator.]tablename TO userid, ... [WITH GRANT OPTION]
```

Bu erda

-privilege – tayinlanayotgan imtiyozlar ro'yxati,

-tablename - jadval nomi,

-userid – imtyozlar olgan foydalanuvchilar ro'yxati.

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

Ma'lum foydalanuvchilarga imtiyozlarni SQL Central da ikki usul bilan tayinlash mumkin. Birinchidan Users & Groups papkasini tanlash va ma'lum foydalanuvchi xossalarini ro'yxatdan chaqirish (sichqoncha o'ng klavishchasini bosish va menyu Properties punktini tanlash). So'ngra Permissions qo'shimcha



sahifasida kerakli jadvalni tanlab imtiyozni o'rnatish. Ikkinchidan Tables yoki Views papkasida ma'lum jadval yoki tasavvur xossalar oynasini chaqirish, so'ngra Permissions qo'shimcha sahifasiga o'tish va GRANT tugmasi yordamida kerakli foydalanuvchini tanlab, imtiyozni o'rnatish.

### ***Ma'lum ustunlarga imtiyozlarni cheklanish***

Bu cheklanish UPDATE va REFERENCES imtiyozlarida ishlatilishi mumkin. Bu xolda imtiyoz ko'rsatilgandan so'ng qavs ichida shu imtiyoz qo'llaniluvchi ustunlar ko'rsatiladi (agar ustunlar ko'rsatilmagan bo'lsa, imtiyoz butun jadvalga ta'sir o'tkazadi).

Masalan:

GRANT UPDATE (City, Comm) ON Salespeople TO Diane; - bu Diane ga Salepeople jadvalining City va Comm ustunlari qiymatlarini o'zgartirish huquqini beradi yoki GRANT REFERENCES (CName, CNum) ON Customers TO Stephen; - bu komanda Stephen ga CNum va CName ustunlarini o'zining jadvallaridagi ixtiyoriy tashqi kalitlarga nisbatan ajdod kalit sifatida ishlatish huquqini beradi. Stephen (CName, CNum) yoki (CNum, CName) ustunlarni, jadvalarining ikki ustuni bilan tashqi kalit yordamida mos kelgan ikki -ustunli ajdod kalit sifatida aniqlashi mumkin. Yoki u maydonga individual murojaat qilish uchun ajratilgan tashqi kalitlar yaratishi mumkin.

### ***ALL va PUBLIC argumentlaridan foydalanish.***

ALL jadvalda hamma imtiyozlarni berish uchun ishlatiladi.

Masalan:

## **GRANT ALL ON Customers TO Stephen;**

Agar siz imtiyozlarni publikatsiya (PUBLIC) uchun uzatsangiz, hamma foydalanuvchilar avtomatik ravishda ularni qabul qiladi. Odatda bu ma'lum asos jadvallarda yoki tasavvurlarda (VIEW) imtiyozi uchun qo'llanadi. Ixtiyoriy foydalanuvchiga Buyurtmalar jadvalini ko'rish imkonini berish uchun, siz quyidagini kiritishingiz mumkin:

## **GRANT SELECT ON Orders TO PUBLIC;**

### ***WITH GRANT OPTIONS yordamida imtiyozlar berish.***

Ba'zida jadval yaratuvchisiga boshqa foydalanuvchilar uning jadvalida imtiyozlarni uzatish imkoniga ega bo'lishlari kerak. Odatda bu bir yoki bir necha xodimlar bir necha yoki hamma asos jadvallarni yaratib, ularni shu jadvallar bilan ishlaydigan xodimlarga topshiradigan tizimlarda zarurdir.

SQL da buning uchun WITH GRANT OPTION ifodasidan foydalaniladi.

Masalan:

Agar Diane buyurtmachilar jadvalida Adrian boshqa foydalanuvchilarga SELECT imtiyozini berish huquqiga ega bo'lishini istasa, unga WITH GRANT OPTION ifodasidan foydalanib SELECT imtiyozini beradi:

## **GRANT SELECT ON Customers TO Adrian WITH GRANT OPTION;**

Adrian uchinchi shaxslarga SELECT imtiyozini berish huquqiga ega bo'lgandan so'ng quyidagi komandani berishi mumkin

## **GRANT SELECT ON Diane.Customers TO Stephen;**

yoki

**GRANT SELECT ON Diane.Customers TO Stephen WITH GRANT OPTION;**

Huquq olgan foydalanuvchilar jadvalga murojaat qilganda jadval egasining murojaat ID sini o‘rnatishlari lozim, chunki jadval yaratuvchiga tegishlidir.

### ***Imtiyozlarni rad etish***

Imtiyozlarni REVOKE komandasi yordamida rad etish mumkin, uning sintaksisi GRANT ga o‘xshash, lekin teskari ta’sirga ega.

Masalan Adrian va Stephen uchun sotib oluvchilar jadvalida INSERT va DELETE imtiyozlarini rad etish uchun quyidagi komandadan foydalanish lozim: REVOKE INSERT, DELETE ON Customers FROM Adrian, Stephen;

Imtiyozlarni rad etishda quyidagi qoidalarga rioya qilinadi: imtiyozlar ularni bergan foydalanuvchi tomonidan rad etiladi va rad etish kasakadlanadi, ya’ni undan shu imtiyozlarni olgan hamma foydalanuvchilarga tarqaladi.

### ***Imtiyozlar boshqa turlari (tizim imtiyozlari).***

Ma’lumotlar maxsus ob’ektlari terminlarida aniqlanmaydigan imtiyozlar tizim imtiyozlari yoki ma’lumotlar bazalari qoidalari deb ataladi. Umumiy yondoshishda uchta asosiy tizim imtiyozlari mavjud:

- CONNECT (Ulash),
- RESOURCE (Resurs),
- DBA (Ma’lumotlar Bazasi Administratori).

Soddaroq qilib aytish mumkinki CONNECT agar ob’ekt imtiyozlari uzatilgan bo‘lsa registratsiya qilinish, tasavvurlar va sinonimlar yaratish huquqidan iborat. RESOURCE asos jadvallari yaratish huquqidan iborat. DBA bu ma’lumotlar

bazasida foydalanuvchiga eng yuqori imkoniyatlar beruvchi superfoydalanuvchi imtiyozidir. Ma'lumotlar bazasi administratori funksiyasiga ega bir yoki bir necha foydalanuvchi shu imtiyozga ega bo'lishi mumkin.

Faqat DBA identifikatorli foydalanuvchi CONNECT, RESOURCE va DBA imtiyozlarini berishi mumkin.

Foydalanuvchiga resurs yoki administrator imtiyozini berish uchun quyidagi komandalarni bajarish etarli GRANT RESOURCE TO userid; yoki mos ravishda GRANT DBA TO userid.

SQL Central da RESOURCE va DBA v imtiyozlarini berish uchun Users & Groups papkasida foydalanuvchi xossalarini chaqirib, Authorities qo'shimcha sahifasida mos imtiyoz yoniga belgi qo'yish lozim.

### ***Foydalanuvchilarni yaratish va o'chirish.***

Foydalanuvchini yaratish unga CONNECT imtiyozini uzatish yo'li bilan bajariladi. Kommanda sintaksisi quyidagicha:

```
GRANT CONNECT TO <userid> IDENTIFIED BY <password>;
```

Bu userid nomli foydalanuvchi yaratilishiga olib kelitb, unga registratsiya qilinish huquqini beradi vaunga parol password tayinlaydi.

Foydalanuvchini o'chirish REVOKE komandasi yordamida CONNECT imtiyozini rad etish orqali amalga oshiriladi. Faqat bazada o'z jadvallariga ega bo'lmagan foydalanuvchini o'chirish mumkin, chunki bu jadvallar egasiz qoladi. Shuning uchun bunday foydalanuvchini o'chirishdan oldin uning hamma jadvallarini o'chirish kerak.

## **2.8 Tarmoqda ma'lumotlar bazalari arxitekturasi**

Ma'lumotlar bazasining bor imkoniyatlari u bilan ishlashga mo'ljallangan instrumentlar orqali ochiladi. Biz quyida bunday instrumentlarni hozirgi davrda keng tarqalgan dasturlash tillari yordamida yaratishni ko'ramiz. Biznes uchun web Amaliy dasturlari yaratish misolida biz API Lar, hamda MySQL va mSQL imkoniyatlaridan maksimal foydalanish uchun zarur bo'lgan instrumentlarni ko'rib chiqamiz. Biz avval ma'lumotlar bazasi bilan ishlovchi amaliy dasturlar arxitekturasi va SGI- dasturlashga to'xtalamiz. Asosan MySQL va mSQL uchun har xil tillarda dasturlashni ko'ramiz.

### ***Ma'lumotlar bazasi bilan ishlash amaliy dasturlari arxitekturasi***

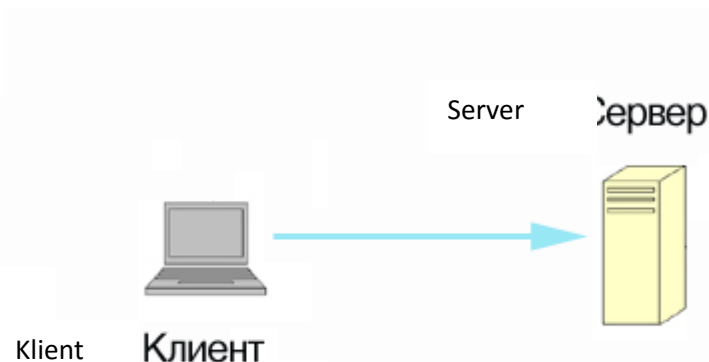
Ma'lumotlar bazasi bilan ishlash uchun har xil tillarda amaliy dasturlar yaratish xususiyatlarni ko'rishdan oldin, bu amaliy dasturlarni loyihalashni ko'rib chiqish kerak. Biz kontseptual tushuncha ya'ni ma'lumotlar bazasini dasturlash asosida yotadigan klient/ server arxitekturasini ko'rib chiqamiz. Bu masalalar MySQL va mSQL da dasturlash uchun muxim, lekin faqat ularga xos emas. Aksincha ular ma'lumotlar bazalarini dasturlash ixtiyoriy muxiti uchun muximdir. Agar arxitektura printsiplarini hisobga olmasa, sizni amaliy dasturlaringiz talablarigizga javob berolmaydi va o'zgaruvchi muxitga moslasha olmaydi. Biz quyida murakkab mavzularga ya'ni oddiy ikki bo'g'inli arxitektura, ob'ektlar va relyatsion ma'lumotlar orasidagi munosabat, hamda yangi uch bo'g'inli klient/server arxitekturasiga to'xtalib o'tamiz.

### ***Klient/server arxitekturasi***

Sodda xolda klient/server arxitektura amaliy dasturdagi qayta ishlashni ikki yoki undan ko'p mantiqiy qismlarga ajratishga asoslangan. Ma'lumotlar bazasi qandaydir amaliy dastur tomonidan foydalanish uchun yaratilgandir. Soddalashtirib aytish mumkinki ma'lumotlar bazasi klient/server arxitekturasining bir qismini tashkil qiladi. Ma'lumotlar bazasi «server», undan foydalanuvchi har qanday amaliy dastur «klient». Odatda klient va server har xil mashinalarda

joylashgan; ko'p xollarda klient amaliy dasturi ma'lumotlar bazasiga do'stona interfeysdir. Quyidagi rasm 2-1 grafik shaklda klient/server sodda tizimi tasviri berilgan.

Balkim siz Internetda bunday strukturani uchratgan bo'lishingiz mumkin. Moxiyat bo'yicha biz Internet uchun klient/server amaliy dasturlarini yaratish masalasiga murojaat qilamiz. Masalan WWW ulkan klient/server tipidagi amaliy dastur bo'lib, unda Web-brouzer klient Web-server bo'lsa serverdir. Bu stsenariyda server relyatsion ma'lumotlar bazasi serveri emas, maxsus fayl-serverdir. Serverning muxim xususiyati shundaki, u klientga ma'lumotlarni ma'lum shaklda uzatadi.



**1-rasm.** *Kient/server arxitekturasi*

Ma'lumotlar bazasi bilan ishlaydigan amaliy dasturlar yaratilganda avvalambor klientni ma'lumotlar bazasi bilan bog'lash imkoniyatiga ega bo'lish kerak. Ma'lumotlar bazalari yaratuvchilar dasturchilardan konkret tilga mo'ljallangan, API yordamida bog'lanish asosiy mexanizmlarni berkitadilar. Ma'lumotlar bazasi bilan ishlovchi dastur yaratganingizda sizni so'rovlaringizni tarmoq orqali ma'lumotlar bazasi serveriga uzatiluvchi TCP/IP paketlariga translyatsiya qiladi.

Ma'lumotlar bazasiga murojaat API larining tashqi ko'rinishi har xil va dasturlash tillariga, ko'p xollarda ma'lumotlar bazasining o'ziga bog'liq. MySQL uchun API lar mSQL bilan o'xshash qilib yaratilgani uchun, biz ko'radigan API lar orasidagi farq minimaldir.

### ***Ma'lumotlarni qayta ishlash***

Ma'lumotlar bazasi ishlashga mo'ljallangan dastur tranzaktsiyalarni boshqarish va natijaviy to'plamlarni qayta ishlash uchun mo'ljallangan instrumentdir. Masalan, sizni amaliy dasturingiz adres kitobi bo'lsa, natijaviy to'plamlarni qayta ishlash, jadvaldan hamma satrlarni ajratish va foydalanuvchiga ko'rsatishdan iborat bo'ladi. Tranzaktsiyalarni boshqarish address va person jadvallarida o'zgarishlar birgalikda amalga oshirishni ta'minlashdan iborat bo'ladi.

*MySQL da tranzaktsiyalar qo'llanmaydi. Ma'lumotlar bazasida har qanday o'zgarish avtomatik ravishda siz so'raganingizda yuz beradi. Bu cheklanish ma'lumotlar bazasiga ikki bog'liq murojaat oraliq'ida rad etish natijasiga ma'lumotlar yaxlitligi buzilmasligi uchun maxsus usullar qo'llashga majbur qiladi.*

Amaliy dastur ishidagi ikki muxim moment bu ulanish va uzilishdir. Tushunarliki so'rovni bajarishdan oldin ma'lumotlar bazasiga ulanish lozim. Lekin ko'pincha masalaning ikkinchi tomoni haqida unutiladi, ya'ni «o'zidan keyin tozalash». Har doim egallangan ma'lumotlar bazasi resurslarini ozod qilish lozim, agar ular kerak bo'lmasa. Ko'p yashovchi amaliy dasturlarda masalan Internet demoni kabi, beparvo yozilgan tizim ko'p vaqt ma'lumotlar bazasi resurslarini egallashi mumkin va oxir borib tizimni blokirovka qilishi mumkin.

«O'zidan keyin tozalash» xatolarni to'g'ri qayta ishlashdan iborat. Yaxshi dasturlash tillari g'ayri odiy xolatlar (tarmoqning rad qilishi, qo'shilganda qaytariluvchi kalitlar, SQL sintaksisi xatosi va xokazo) qayta ishlovilarini o'tkazib yuborishni qiyinlashtiradi. Qaysi tilni tanlagandan qat'iy nazar, API bu chaqirig'ida qanday g'ayri oddiy xolatlar yuzaga kelishi mumkinligini va har bir g'ayri oddiy xolatda nima qilish kerakligini bilishingiz kerak. MySQL uchun S-bibliotekalari ma'lumotlar bazalarini satrlar to'plami sifatida tasvirlashga asoslangan. Bu shuni bildiradiki S bibliotekalari ma'lumotlar bilan ma'lumotlar bazasida qanday tasvirlangan bo'lsa shu shaklda qaya ishlashga imkon beradi.

Ob'ektga-yo'naltirilgan muxitdan relyatsion ma'lumotlar bazasiga murojaat aloxida paradoksni keltirib chiqaradi: relyatsion Dune ma'lumotlar ustida ish olib

borsa, ob'ektlar dunyosi faoliyat sxemalari asosida ma'lumotlarni inkapsulyatsiya qilish bilan shug'ullanadi. Ob'ektga-yo'naltirilgan amaliy dasturlar ma'lumotlar bazasi ob'ektlarni saqlash vositasi bo'lib xizmat qiladi. Ob'ektga-yo'naltirilgan dasturlash so'rov ma'lumotlarni satrlar to'plami deb emas, ob'ektlar to'plami deb qaraydi.

### ***Ob'ekli/relyatsion modellashtirish***

Ob'ektga yo'naltirilgan amaliy dastur yaratuvchisi oldida relyatsion ma'lumotlar bazasidan foydalanishda tug'iladigan masala, relyatsion ma'lumotlarni qanday qilib ob'ektlarga akslantirishdir. Birinchi yo'l ob'ekt atributlarini jadval maydonlariga akslantirish. Lekin bu usul ancha noqulaydir.

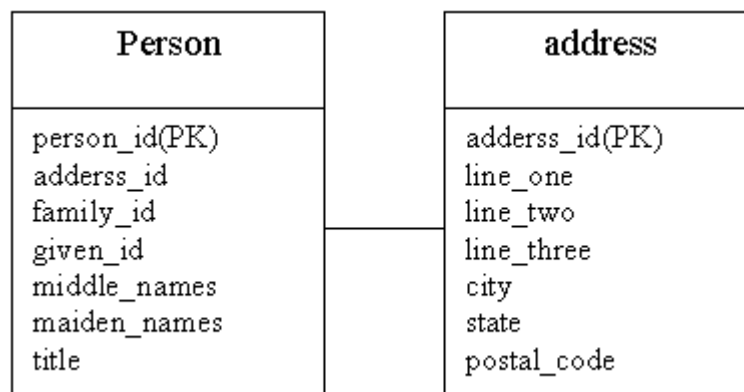
- Ob'ektlar o'z atributlarida faqat berilganlarni saqlamaydilar. Ularda kolleksiyalar va boshqa ob'ektlar bilan bog'lanishlar saqlanadi.
- Ko'pgina relyatsion ma'lumotlar bazalarida vorislikni modellashtirish imkoniyati yo'qdir.

### **Ob'ekli-relyatsion modellashtirish uchun amaliy qoidalar**

- Har bir saqlanuvchi sinf uchun ma'lumotlar bazasida o'zining jadvali mavjud.
- Ob'ektlarning sodda tipli (butun, simvol, satr va xokazo) maydonlariga ma'lumotlar bazasidagi mos jadvalning ustunlariga mos qo'yilgan.
- Ma'lumotlar bazasi jadvalining har bir satri saqlanuvchi sinf nusxasiga mos keladi.
- Har bir ob'ektlar orasidagi «ko'pga-ko'p» turdagi bog'lanish jadval – bog'lovchini talab qiladi, xuddi shuningdek ma'lumotlar bazasi «ko'pga-ko'p» turdagi ob'ektlari uchun shu talab qilinadi.
- Vorislik sinf va ostki sinfga mos keluvchi jadvallar orasidagi «birga-bir» munosabat yordamida modellashtiriladi.

Oldin ko'rilgan adreslar kitobini eslang. Masalan address va person jadvallariga ega bo'lsin, 2-2 rasmdagi kabi.

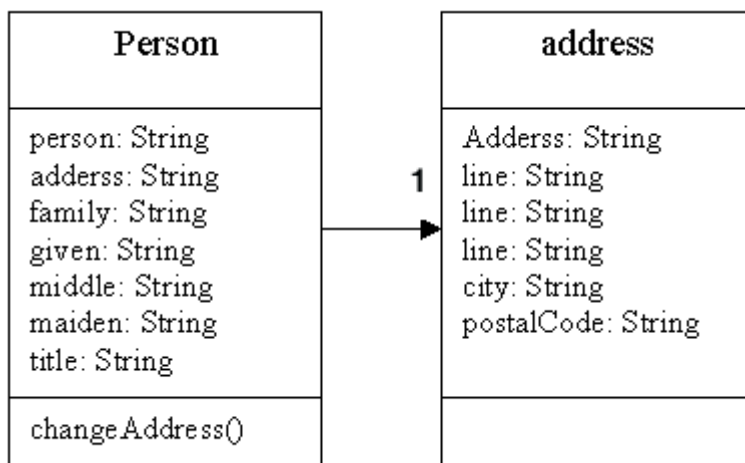




**2-rasm.** Adres kitobi amaliy dasturi ma'lumotlar modeli

*Dasturchilarga uchraydigan ravshan bo'lmagan muammo mavjuddir. Ob'ektga-yo'naltirilgan dasturlashni relyatsion ma'lumotlarga qo'llash - bu ma'lumotlarni olib, darhol ob'ekt nusxasini yaratishdir. Amaliy dastur ma'lumotlar bilan faqat ob'ektlar orqali ishlashi kerak. Ko'pgina traditsion dasturlash usullari, shu qatorda S, PowerBuilder va VisualBasic da dasturlash, yaratuvchi bazadan ma'lumotlarni olib, keyin qayta ishlashga asoslangan. Asosiy farq shundaki ma'lumotlar bazalarini ob'ektga-yo'naltirilgan dasturlashda siz ma'lumotlar bilan emas, ob'ektlar bilan ishlaysiz.*

2 -rasmda ko'rsatilgan ob'ektli model, 1-rasmda ko'rsatilgan ma'lumotlar modeliga mos keladi. Ma'lumotlar bazasi har bir satri dasturiy ob'ektga aylantiriladi. Shunday qilib amaliy dastur natijaviy to'plamni qabul qiladi va har qaytariluvchi satr uchun Address yoki Person yangi nusxasini yaratadi. Eng qiyini yuqorida ko'rsatilgan muammoni xal qilish: qanday qilib amaliy dasturda odam va uning adresi orasida bog'liqlikni o'rnatish. Ob'ekt Person, albatta shu odamga tegishli ob'ekt Address ga ilovaga ega, lekin ob'ekt Address ni relyatsion baza person jadvalida saqlash mumkin emas. Ma'lumotlar modeli ob'ektlar orasidagi bog'lanishni tashqi kalitlar yordamida saqlaydi. Buning uchun person jadvaliga address\_id kiritiladi.



**3-rasm** Adres kitobi sodda amaliy dasturi ob'ektli modeli

Ob'ektli modelning ozgina murakkablashi bizning ob'ektlarimiz va ma'lumotlar modellari orasida moslikni o'rnatishda katta muammolarga olib keladi.

Masalan Person sinfi Entity avlodi va Company sinfi ham Entity avlodi bo'lsin. Entity ni Person dan yoki Companynidan qanday ajratish mumkin? Yuqorida keltirilgan qoida faqat yo'llanmadir. Bazi xollarda asos sinf abstrakt bo'lib ma'lumotlar bazasida unga bog'liq ma'lumotlarga ega emas. Bu xolda bu sinf uchun ma'lumotlar bazasida ob'ekt mavjud bo'lmaydi.

### ***Uch bo'g'inli arxitektura***

Shu paytgacha biz WWW va biznes amaliy dasturlari bilan ishlash eng sodda arxitekturasi klient/server arxitekturasi muxokama qildik. Lekin bu arxitekturani Amaliy dasturlar rivojlanishi bilan takomillashtirish ancha murakkabdir. Bu arxitekturada ob'ektga-yo'naltirilgan dasturlash imkoniyatlaridan foydalanish ham qiyin. Birinchi muammo «nozik klientlar» haqidagi baxslarda o'z aksini topdi. Nozik klientlarga bo'lgan talab, klientga uzatilayotgan ma'lumotlar o'sib borish tendetsiyasidan kelib chiqdi. Bu muammo PowerBuilder va VisualBasic larda ko'rindi. Ular bazadan ma'lumotlarni GUI ga oladi va bu ma'lumotlar ustidagi hamma amallarni GUI da bajaradi.

Foydalanuvchi interfeysini baza yadrosiga bog‘lab qo‘yish foydalanuvchilar soni va ma’lumotlar xajmi oshishi bilan o‘zgartirish va masshtablash qiyin bo‘lgan dasturlar yaratilishiga olib keladi. Agar sizda foydalanuvchi interfeysi yaratish tajribasi bo‘lsa, foydalanuvchi xohishiga qarab interfeysni qayta ishlab chiqish muammosiga duch kelgansiz. Bunday qayta ishlashni kamaytirish yo‘li GUI uchun faqat bita vazifa – foydalanuvchi interfeysi vazifasini qoldirish kerak. Foydalanuvchi bunday interfeysi chindan ham nozik klientdir.

Masshtablanishga ta’sir o‘tkazish boshqa tomondan ham ko‘rinadi. Agar foydalanuvchilar soni va ma’lumotlar hajmi oshgani munosabati bilan amaliy dasturni qayta ishlab chiqish kerak bo‘lsa, modifikatsiya ma’lumotlar bazasiga o‘zgartirish kiritish yo‘li bilan amalga oshirilishi mumkin. Masalan ma’lumotlar bazasini bir necha serverlarga taqsimlash yo‘li bilan. Interfeysni ma’lumotlar bazasiga bog‘lab qo‘yish masshtablash muammosini xal qilish uchun GUI ni o‘zgartirishga majbur qiladi. Aslida esa bu server bilan bog‘liq muammolardir.

Nozik klientlar – bugunda yagona yo‘nalish emas. Boshqa yo‘nalish- koddan qayta foydalanish. har xil amaliy dasturlar uchun kod biznes logika deb atalgan qayta ishlashga yo‘naltiriladi. Agar biznes logika foydalanuvchi interfeysida joylashgan bo‘lsa, koddan qayta foydalanishni ta’minlash qiyin bo‘ladi. Bu muammoni xal qilish yo‘li amaliy dasturni ikki qismga emas uch qismga ajratishdir. Bunday arxitektura uch bo‘g‘inli deyiladi.

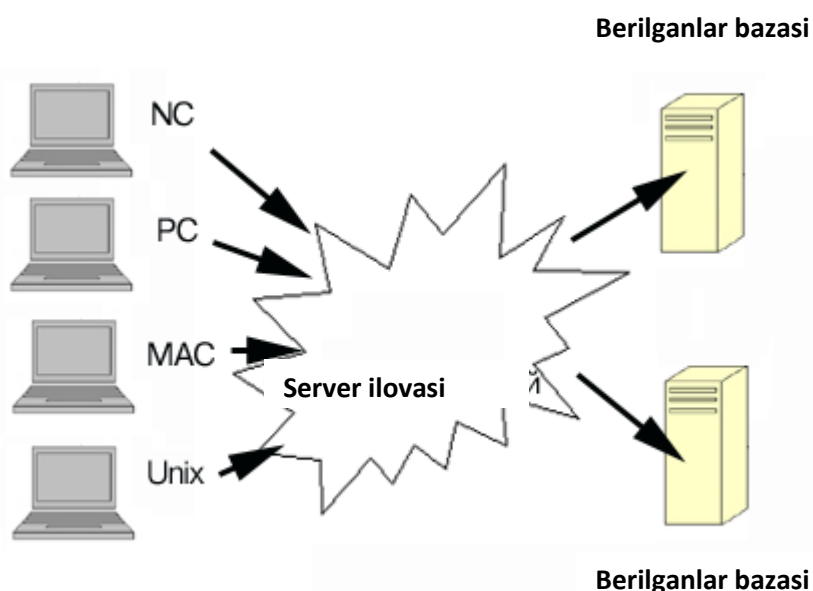
*Klientdagi foydalanuvchi interfeys haqida gapirganimizda, mantiqiy farqni nazarda tutamiz. Nozik klient bir turi «O‘ta nozik klient», bo‘lib ko‘pchilik Web-sahifa deb qabul qiladi. Web-sahifa dinamik tarzda Web-serverda yaratilishi mumkin. Bu holda klient ishining ko‘p qismi serverda HTML-sahifalarni dinamik generatsiya qilish shaklida bajariladi.*

2-1 rasmda ko‘rsatilgan ikki bo‘g‘inli arxitekturani 2-4 rasmda ko‘rsatilgan uch bo‘g‘inli arxitektura bilan solishtiring. Biz foydalanuvchi interfeysi va ma’lumotlar bazasi orasida qo‘shimcha qatlam joylashtirdik. Bu yangi qatlam

amaliy dasturlar serveri o'zida biror soha uchun umumiy bo'lgan amaliy dastur ish mantig'i – biznes mantiqni oladi. Klient o'rta yarus ob'ektlarini ko'rish vositasi, ma'lumotlar bazasi bo'lsa shu ob'ektlar omboriga aylanadi.

Sizni eng katta yutug'ingiz - foydalanuvchi interfeysi bilan ma'lumotlar bazasini ajratilishidir. Endi sizga ma'lumotlar bazasi haqidagi ma'lumotni GUI ga kiritishga to'g'ri kelmaydi. Balkim, ma'lumotlar bazasi bilan qanday ishlash haqidagi hamma ma'lumotlar, o'rta yarusda joylashishi mumkin.

Amaliy dasturlar serverining ikki asosiy vazifasi – ma'lumotlar bazasiga ulanishlarni izolyatsiya qilish va biznes mantiq uchun markazlashgan omborni ta'minlash. Foydalanuvchi interfeysi faqat ma'lumotlarni kiritish va akslantirish bilan shug'ullanadi, ma'lumotlar bazasi yadrosi bo'lsa faqat ma'lumotlar bazasi muammolari bilan shug'ullanadi. Ma'lumotlarni qayta ishlashni markazlashtirish Amaliy dasturlar serverining bitta dasturini har xil foydalanuvchi interfeyslari ishlatishi mumkin va har gal yangi amaliy dastur yaratilganda ma'lumotlarni qayta ishlash qoidalarini yozish kerak bo'lmay qoladi.



*4-rasm. Uch bo'g'inli arxitektura*

### **III-BOB. MICROSOFT ACCESS DASTURIDA ODDIY SQL TILI SO‘ROVLARNI YARATISH USLUBLARI**

#### **3.1. Microsoft Access dasturida oddiy SQL tili so‘rovlarni yaratish uslublari**

Xozir kunda barcha sohalarda eng ko‘p qo‘llaniladigan ma’lumotlar bazasini boshqarish tizimlaridan biri SQL dir. Shuning uchun ham u haqida qisqacha to‘htalib o‘tamiz.

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 loyihasini ishlab chiqish doirasida yaratildi. Keyinchalik IBM firmasi tomonidan System R ga yaqin bo‘lgan yana ikkita tizim - SQL/DS va DB2 tizimlari ishlab chiqildi. Bu tilning xalqoro standarti 1986 yili ishlab chiqildi va u 1989 yilga kelib yanada kengaytirildi, lekin uning to‘liq xalqaro standarti 1992 yil qabul qilindi. 1995 yilga kelib SQL92 standarti yangi komponentalar bilan to‘ldirildi. Birinchi bolib SQL dan foydalanish Oracle MBBTda ishlatildi. SQLquyidagi komponentalarni o‘z ichiga oladi:

- administrativ ma’lumotlar vositasi;
- tranzaksiyalar bilan boshqarish vositasi;
- DLL (Data Definition Language) ma’lumotlarni aniqlash operatorlari;
- DML (Data Manipulation Language) ma’lumotlar ustida manipulyatsiya qilish(murakkab harakatlar bajarish) operatorlari;
- DQL (Data Query Language) ma’lumotlar bazasiga so‘rovlar tili.

Administrativ ma’lumotlar - ma’lumotlar bazasini yaratish, unga kirishga ruxsat berish va parollarni o‘zgartirishda kerak bo‘ladi. Tranzaksiya - bu ma’lumotlar bilan monipulyatsiya qilib ketma-ket operatsiyalar yordamida MBBTga ta’sir etishdir. Tranzaksiya yoki to‘liq bajariladi va ma’lumotlar bazasini

bir butun holatdan ikkinchi bir butun holatga o'tkazadi yoki agar ma'lum sababga ko'ra tranzaksiyaning biror holati bajarilmaydigan bo'lsa yoki sistemada biror xatolik yuz bersa, ma'lumotlar bazasi boshlang'ich holatiga qaytadi.

Axborot texnologiyalarning rivojlanishi va axborot oqimlarining tobora ortib borishi, ma'lumotlarning tez o'zgarishi kabi holatlar insoniyatni bu ma'lumotlarni o'z vaqtida qayta ishlash choralari qidirib topishga undaydi. Ma'lumotlarni saqlash, uzatish va qayta ishlash uchun ma'lumotlar bazasi (MB) ni yaratish, so'ngra undan keng foydalanish bugungi kunda dolzarb bo'lib qolmoqda.

Ma'lumotlar bazasi – bu o'zaro bog'langan va tartiblangan ma'lumotlar majmuasi bo'lib, u ko'rilayotgan ob'ektlarning xususiyatini, holatini va ob'ektlar o'rtasidagi munosabatni ma'lum sohada tavsiflaydi.

Darhaqiqat, hozirgi kunda inson hayotida MBda kerakli axborotlarni saqlash va undan oqilona foydalanish juda muhim rol o'ynaydi. Sababi: jamiyat taraqqiyotining qaysi jabhasiga nazar solmaylik o'zimizga kerakli ma'lumotlarni olish uchun, albatta, MBga murojaat qilishga majbur bo'lamiz. Demak, MBni tashkil qilish axborot almashuv texnologiyasining eng dolzarb hal qilinadigan muammolaridan biriga aylanib borayotgani davr taqozasi.

Ma'lumki, MB tushunchasi fanga kirib kelgunga qadar, ma'lumotlardan turli ko'rinishda foydalanish juda qiyin edi. Dastur tuzuvchilar ma'lumotlarini shunday tashkil qilar edilarki, u faqat qaralayotgan masala uchungina o'rinli bo'lardi. Har bir yangi masalani hal qilishda ma'lumotlar qaytadan tashkil qilinir va bu hol yaratilgan dasturlardan foydalanishni qiyinlashtirar edi.

Shuni qayd qilish lozimki, MBni yaratishda ikkita muhim shartni hisobga olmoq zarur:

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

Ikkinchidan, MBdagi kerakli ma'lumotni bilish yoki izlash uchun biror dastur tuzishga hojat qolmasin.

Shuning uchun ham MBni tashkil etishda ma'lum qonun va qoidalarga amal qilish lozim. Bundan buyon axborot so'zini ma'lumot so'zidan farqlaymiz, ya'ni axborot so'zini umumiy tushuncha sifatida qabul qilib, ma'lumot deganda aniq bir belgilangan narsa yoki hodisa sifatlarini nazarda tutamiz.

Bugungi kunda ma'lumotlarni eng ishonchli saqlaydigan vositalardan biri esa hozirgi zamon kompyuterlaridir. Kompyuterlarda saqlanadigan MB - bu maxsus formatga ega bo'lgan muayyan tuzilmali fayl demakdir. Kompyuter xotirasida har bir fayl, yozuv deb ataladigan bir xil turdagi qismlardan iborat bo'ladi. Yozuv-o'zaro bog'langan ma'lumotlarning bir qismidir. Fayldagi yozuvlar soni, qaralayotgan ma'lumotning o'lchoviga bog'liq. Har bir yozuv esa maydon deb ataladigan bo'laklardan tashkil topadi. Maydon ma'lumotlarning, imkoni boricha, qisqa to'plamidan iborat bo'lishi lozim. Har bir maydon, o'zi ifodalaydigan ma'lumotlariga ko'ra, biror nomga ega bo'ladi Shuni ta'kidlash lozimki, hozirgi vaqtda deyarli barcha MBBTlar asosan relyasion modellar asosida tashkil qilinmoqda. Shuni nazarda tutgan Microsoft Office korporatsiyasi ham eng ommalashgan dastur vositalarga ega, bu dastur vositalari ixtiyoriy sohada yuqori darajadagi professional hujjatlar tayyorlash imkonini beradi.

Shulardan biri MBlar bilan ishlashga mo'ljallangan Microsoft Access dasturi bo'lib, bu dastur Visual Basic for Application dasturlash muhitida makroslar yaratish va boshqa bir qancha imkoniyatlarga egaki, bu foydalanuvchiga har tomonlama mukammal bo'lgan hujjatlar tayyorlashga yordam beradi.

Microsoft Office ning o'zbek tilidagi varianti yo'qligi, uning faqat ingliz va rus tilida yaratilgan lahjalaridangina foydalanish imkoniyatiga ega ekanligimizdan mazkur fikrlar Microsoft Access ning ruscha lahjasiga tayanib yozilgan. Microsoft Access dasturi ham relyasion modellar asosiga qurilgan bo'lib, unda tashkil qilinadigan.

MBlar jadval ko‘rinishida aks etadi. Bunday jadvaldagi ustunlar maydon deb, satrlar esa yozuv deb ataladi.

Maydon – ma’lumotlarni tashkil etishning oddiy birligi bo‘lib, ma’lumotning alohida, bo‘linmas birligiga egaligi rekvizitga mos keladi.

Yozuv– mantiqiy bog‘langan rekvizitlarga mos keluvchi maydonlar yig‘indisidir. Yozuvning tuzilishi o‘z tarkibiga mos har bir oddiy ma’lumotga ega maydonlar tarkibi va ketma-ketligi bilan belgilanadi.

Demak, maydon MBning asosiy tuzilmali elementi bo‘lib quyidagi parametrlar bilan ifodalanadi:

- uzunligi (belgi va simvollarda ifodalanib baytlarda o‘lchanadi),
- nomi (maydonning o‘ziga xos alohida xususiyati),
- podpis- imzo (ustun sarlavhasi haqida ma’lumot).

Maydonlar xususiyatiga va tarkibiga qarab quyidagi turlarga bo‘linadi:

1. Matnli maydon.
2. Sonli maydon.
3. Vaqt va sanani ifodalovchi maydon.
4. Mantiqiy maydon (1 0; Ha yoki yo‘q; rost yoki yolg‘on kabi mantiqiy birliklar bilan ifodalanadi).
5. Pul birliklarida ifodalangan maydon (raqamlar pul birliklari bilan birgalikda ifodalanadi).
6. OLE maydoni (shakl, tasvir, rasm, musiqiy kliplar va videoyozuvlar shaklida ifodalanadi).



7. MEMO maydoni- matn uzunligi 256 simvoldan uzun bo'lgan maydonda faqat matnning qayerdaligini ifodalovchi ko'rsatkich turadi. Bu holda har bir maydonda 535 simvol saqlanishi mumkin.

8. Schetchik (sanovchi) maydoni - maydonda turgan ifoda avtomatik ravishda sanalib o'zgaradi.

So'rovlar ko'rish, taxlil qilish va berilganlarni o'zgartirish orqali berilgan mezonlarni qondirishga mo'ljallangan. Access da so'rovlar parametrlari so'rov konstruktori oynasida beriladigan QBE – so'rovlar (Query By Example – namuna bo'yicha so'rov) va so'rovlar tashkil qilishda SQL tilining buyruqlari va funktsiyalari qo'llaniladigan SQL-so'rovlar (Structured Query Language – so'rovlarning strukturali tili)ga bo'linadi. Access QBE - so'rovlarni osongina SQL-so'rovlarga va teskarisiga o'tkazadi.

QBE - so'rovlarning eng ko'p tarqalgan turlaridan biri tanlanma so'rovidir.

**Вид** menyusining **Объекты базы данных** buyrug'ini aktivlashtiring, ochilgan qism menyudan **Запросы** buyrug'ini bajaring (yoki ma'lumotlar bazasining **Запросы** ob'ektidagi qo'yilmani sichqon yordamida ikki marta bosing). **Создат** tugmasini bosing. Monitor ekranida **Новый запрос** muloqat oynasi ochiladi va bu oynada dastur so'rovlar tanlash usullaridan birini tanlashni taklif etadi:

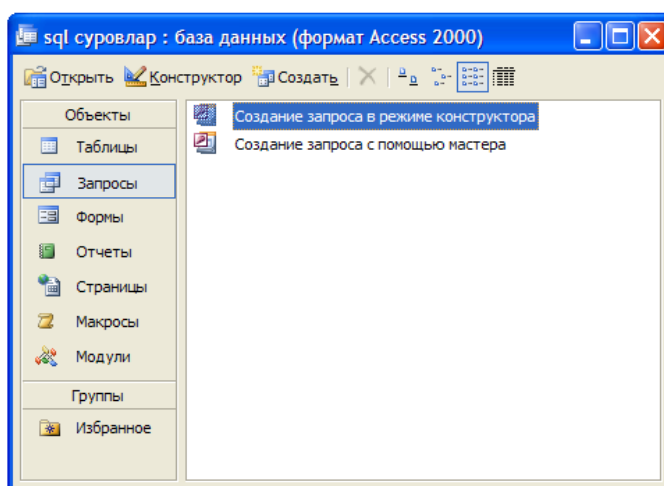
- ✓ Konstruktor– so'rovlarni usta yordamisiz tuzish;
- ✓ Oddiy so'rovlar – tanlangan maydonlar asosida oddiy so'rov tuzish;
- ✓ Qamrovchi so'rov – ma'lumotlar elektron jadvallardagi kabi kompakt formatga ega bo'lgan so'rov tuzish;
- ✓ Takrorlanuvchi yozuvlar - jadvaldagi takrorlanuvchi yozuvlarni yoki oddiy so'rovni tanlovchi so'rov tuzish;

- ✓ Bo‘ysinmaydigan yozuvlar - jadvaldagi boshqa jadvallar yozuvlari bilan aloqada bo‘lmagan yozuvlarni tanlovchi so‘rov tuzish.

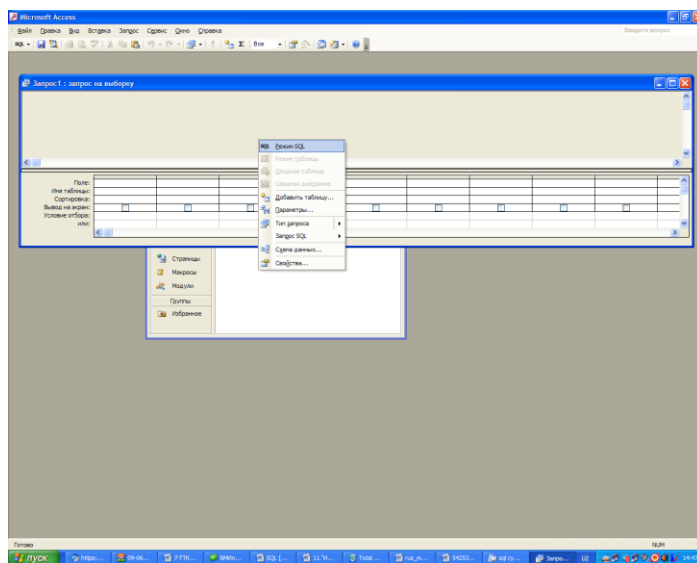
### 3.2. Microsoft Access dasturida oddiy SQL tili so‘rovlarni yaratish.

Microsoft Access dasturida oddiy SQL tili so‘rovlarni yaratish uchun qulay sharoitlar yaratilgan bo‘lib, quyida misollar tarzida asosida chiqishimiz mumkin:

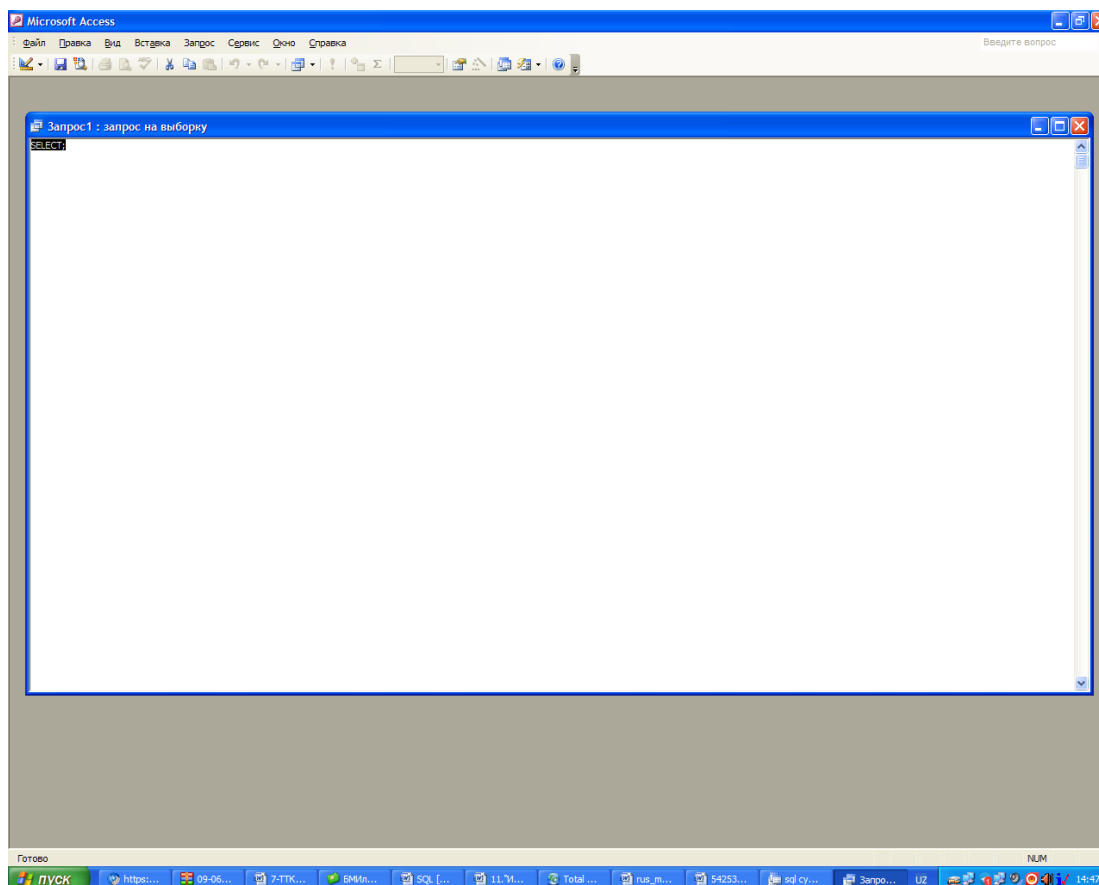
SQL tili rejimiga o‘tish uchun Microsoft Access dasturining Запросы (So‘rovlar) oynasidan konstruktor rejimi orqali SQL tili rejimiga o‘tish karak.



5-rasm. Microsoft Access dasturining Запросы (So‘rovlar) oynasi



6-rasm. Microsoft Access dasturining SQL rejimiga o‘tish



7-rasm. . *Microsoft Access dasturining SQL rejimi*

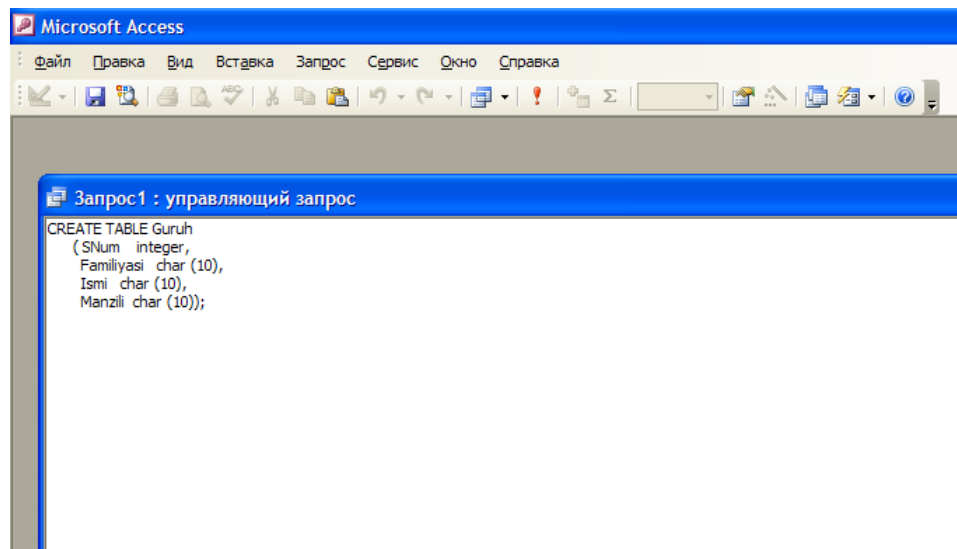
Ushbu oynada SQL so‘rovlari yozilishi mumkin va ishga tushirilib teshirilishi mumkin. Quyida ddiy misollar bilan SQL so‘rovlari yozishni ko‘rib chiqamiz.

Jadvallar yaratish. 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. Guruh nomli jadval yaratish so‘rovi quyidagicha yaratiladi:

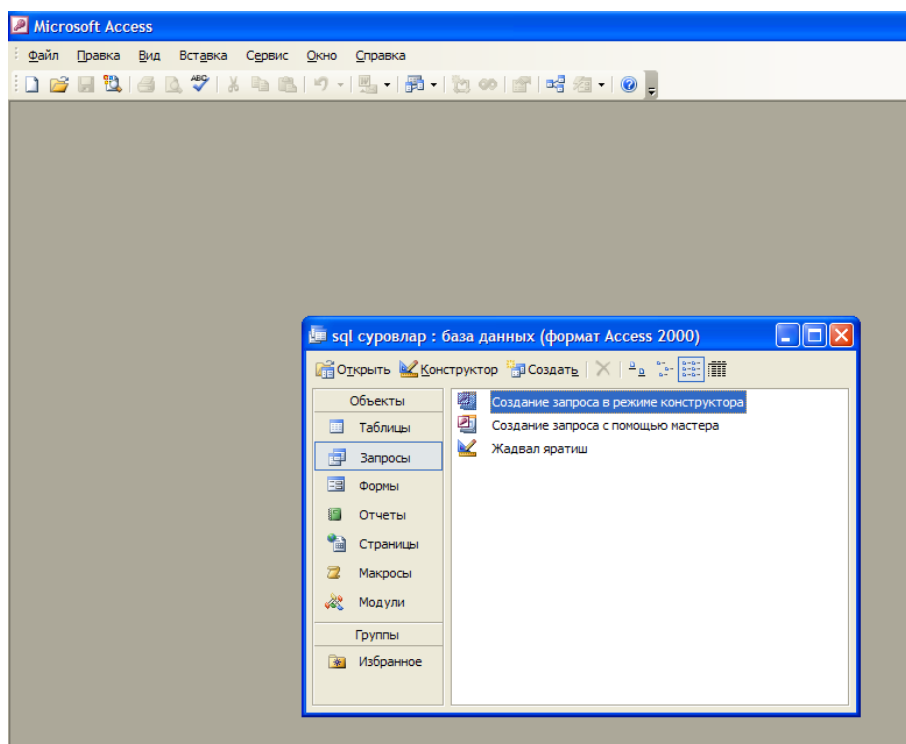
### **CREATE TABLE Guruh**

**( SNum integer,  
Familiyasi char (10),  
Ismi char (10),  
Manzili char (10),**

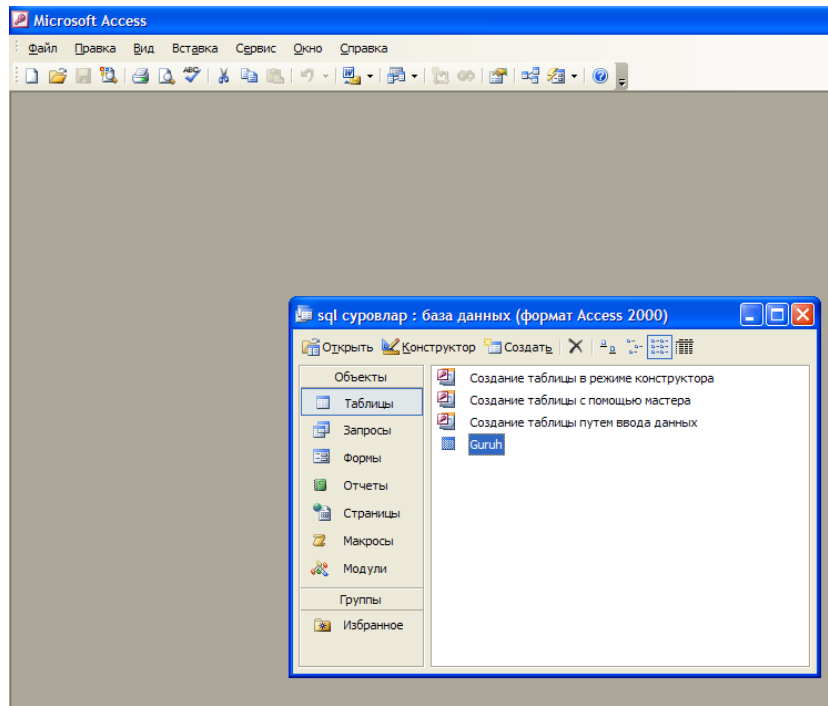
**Comm decimal);**



8-rasm. *SQL da jadval yaratish so`rovi.*



9-rasm. *SQL da yaratilgan jadval yaratish so`rovini ko`rinishi.*



10-rasm. So`rovni natijasi.

Umuman olganda shu jadval yaratish misolidagi kabi juda ham ko`plab oddiy SQL so`rovlarni Microsoft Access dasturida yaratish mumkin.

## XULOSA

Axborot tizimlari va texnologiyalari kundan - kunga kishilik faoliyatining turli sohalarida keng qo'llanimoqda. Ularni yaratish, ishga tushirish va hayotga keng tadbiiq etishdan maqsad – jamiyat va inson butun hayot faoliyatini axborotlashtirish borasidagi muammolarini hal etishdir. Hozirgi davrga kelib, axborot xuddi an'anaviy resurslar kabi izlab topish, tarqatish mumkin bo'lgan resursga aylandi. Ushbu resursning foydalanadigan umumiy hajmi kelgusida davlatlarning strategik imkoniyatini, shuningdek mudofaa qobiliyatini ham belgilab beradi deyishga jiddiy asos bor. Axborot resurslarini oqilona tashkil etish va foydalanish jarayonida ular mehnat, moddiy va energetik resurslar ekvivalenti sifatida namoyon bo'ladi.

Shu sopani rivojlantirish albatta soha fanlarini o'qitishning sifatga ham bog'liqdir. Ushbu bitiruv malakaviy ishda **SQL tilini o'qitishda usulbiy xususiyatlar taxlili misolida o'qitishning** innovatsion uslublarni ishlab chiqish va ulardan foydalanishning pedagogik va nazariy asoslarini metodik jihatdan tahlil qilib quyidagi xulosalar olindi:

- axborot kommunikatsion texnologiyalari fanlarini o'qitishda qo'llaniladigan uslublar o'rganildi va taxlil qilindi;
- SQL tilini o'qitishda pedagogik-innovatsion usullar o'rganildi;
- SQL tilini fanining tuzilishi va bo'limlari mazmuni o'rganildi va taxlil qilindi;
- Microsoft Access dasturida oddiy SQL tili so'rovlarni yaratish uslublari amaliy jihatdan o'rganildi;
- Microsoft Access dasturida oddiy SQL tili so'rovlarni yaratilib amaliy sinab ko'rildi.

Shuningdek, bitiruv malakaviy ishini vazifalaridan kelib chiqib quyidagi ishlar bajarildi:

- mavzuga oid ilmiy adabiyotlar va ilmiy tadqiqot ishlarini tahlil etilib, ilmiy-nazariy, amaliy asoslarini o'rganildi.

- SQL tilini o'qitishda innovatsion uslublarni ishlab chiqildi va ulardan foydalanishga doir ilg'or pedagogik tajribalarni o'rganilib umumlashtirildi.

- SQL tilini o'qitishda innovatsion uslublarni ishlab chiqish va ulardan foydalanishda ta'lim-tarbiya samaradorligiga erishishning shakl va omillarini ishlab chiqish hamda ularning samaradorligini tajriba-sinov orqali aniqlash ishlari olib borildi.

- SQL tilini o'qitishda usulbiy xususiyatlar tahlilini amalga oshirildi.

- MS Access MBBT si yordamida oddiy SQL so'rovlarni yaratildi.

Bir so'z bilan aytganda bitiruv malakaviy ishi o'z oldiga qo'ygan maqsadga erishdi deb hisoblayman. Ishni bajarish davomida olgan nazariy va amaliy bilimlarimni kelgusida o'z kasbimni rivojlantirishda qo'llayman. Yoshlarni bilimli qilishda va kasbga qiziqishlarini shakillantirishda ushbu ish natijalarini qo'llab, vatanimizni rivojlantirishda o'z hissamni qo'shaman degan umiddaman.

## FOYDALANILGAN ADABIYOTLAR RO'YXATI

1. *Karimov I.A.* Mamlakatimizda demokratik islohotlarni yanada chuqurlashtirish va fuqarolik jamiyatini rivojlantirish Konsepsiyasi. – T., 2010.
2. *Karimov I.A.* O'zbekiston: milliy istiqlol, iqtisod, siyosat, mafkura. T.1. – T., 1996.
3. *Karimov I.A.* Bizdan ozod va obod Vatan qolsin. T. 2. – T., 1996.
4. *Karimov I.A.* Vatan sajdagoh kabi muqaddasdir. T. 3. – T., 1996.
5. *Karimov I.A.* Bunyodkorlik yo'lidan. T.4. – T., 1996.
6. O'zbekiston Respublikasi Prezidenti I.A. Karimovning 2007 yil 3 apreldagi PQ-614–son «O'zbekiston Respublikasida axborotning kriptografik himoyasini tashkil etish chora-tadbirlari to'g'risidagi» qarori.
7. *Xaritonova I.A., Mixeeva V.D.* 3D Max 2000: разработка приложений. – СПб.: БХВ – Питербург, 2001. – 832s.: il.
8. *A.YA. Arxangelskiy.* Программирование в Delphi 7. — М.: ООО «Бином-Пресс» 2003 г. — 1152 s.: il.
9. *R.M. Fedorov.* SQL Server 2000. — М.: ООО « Бином-Пресс », 2005 г. — 600 s.: il.
10. *A.Mixaylov* Разработка баз данных MySQL. — М.: ООО «Бином-Пресс», 2002 г. - 300
11. Базы данных 1. Осн. реляц баз данных. Ташкенский Университет Информационных Технологий 2006г.
12. Базы данных. Основы SQL, Ташкенский Университет Информационных Технологий 2006г.
13. Базы данных .SQL – практические проекты, Ташкенский Университет Информационных Технологий 2006г.
14. Базы данных в среде ОС WINDOWS, Ташкенский Университет Информационных Технологий 2006г.
15. Ma'lumotlar bazasi, SH. Nazirov, 2007g.



16. Ma'lumotlar bazasini dasturlash chuqurlashtirilgan kursi, SH. Nazirov, 2007g.
17. Базы данных в 3D Max , Fan va texnologiya, 2005g.
18. Механизмы защиты в сетях ЭВМ / S. Maftik.- М.: Mir, 1993.-216 s.
19. Материал из Википедии — свободной энциклопедии.  
<http://ru.wikipedia.org/wiki/Криптография>. (Internet resurslari)
20. Материал из Википедии — свободной энциклопедии.  
[http://ru.wikipedia.org/wiki/Генератор\\_псевдослучайных\\_чисел](http://ru.wikipedia.org/wiki/Генератор_псевдослучайных_чисел). (Internet resurslari)
21. Wikipedia: Random number generator attack. 2007. Retrieved Nov.27, 2007. (Internet resurslari)
22. Haahr, M. 2007. Random.org. Retrieved November 27, 2007. Random.org. (Internet resurslari)
23. California Soil Resource Lab. 2007. Retrieved November 27, 2007. (Internet resurslari)
24. HotBits. 2007. Retrieved November 27, 2007. (Internet resurslari)
25. Wikipedia: Blum Blum Shub. 2007. Retrieved November 27, 2007. (Internet resurslari)
26. Bishop, M. 2004. Introduction to Computer Security. Prentice Hall PTR. (Internet resurslari)
27. Методы генерации псевдослучайных чисел: Дзехо сепес /Fushimi Masanori //VSP.-1980.-№G-32668, Vol.21,№9.-P.968-974. (Ilmiy maqola va tadqiqot)
28. Псевдослучайные последовательности [http://www.hardline.ru/selfteachers/Info/Security/Protection\\_to\\_information\\_/6/Index5.htm](http://www.hardline.ru/selfteachers/Info/Security/Protection_to_information_/6/Index5.htm). (Internet resurslari)
29. Wikipedia: Random Number Generation. 2007. Retrieved November 27, 2007. (Internet resurslari)
30. O'zbekiston Respublikasining «Davlat sirlarini saqlash to'g'risida»gi 1993 yil 7 may 848-XII-son qonuni // O'zbekiston Respublikasi Oliy Kengashining Axborotnomasi. – 1993. – №5. – 232-m.

31. O‘zbekiston Respublikasining «Elektron hisoblash mashinalari uchun yaratilgan dasturlar va ma’lumotlar bazalarining huquqiy himoyasi to‘g‘risida»gi 1994 yil 6 may 1060-XII-son qonuni //
32. Menezes A., P. van Oorshot, Vanstone S. Handbook of Applied Cryptography. CRC Press. - 1997. - P. 780.
33. Nechvatal J. and others. Report on the Development of the Advanced Encryption Standart (AES). 2000. // <http://csrc.nist.gov/encryption/aes/round2/r2report.pdf>
34. Park S. K., Miller K. W. Random Number Generators: Good Ones Are Hard to Find. // Communications of the ACM. v. 31, n. 10, Oct 1988.-pp. 1192-1201.
35. Peterson I. Monte Carlo Physics: A Cautionary Lesson. // Science News. v. 142, n. 25.- 19 Dec 1992.-p. 422.
36. Rogaway Phillip. Software-Optimized Encryption Algorithm// J. CRYPTOLOGY. 1998. - N 11. - p. 273-287.
37. Schneier B. Applied Cryptography. Wiley, 1996.
38. <http://msdn2.microsoft.com/>
39. <http://www.sql.ru/>
40. <http://www.rushelp.com>
41. [www.tuit.uz](http://www.tuit.uz)
42. [www.pedagog.uz](http://www.pedagog.uz)
43. [www.infoman.uz](http://www.infoman.uz)