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

ISLOM KARIMOV NOMIDAGI TOSHKENT DAVLAT TEXNIKA UNIVERSITETI

MA’LUMOTLAR BAZASINI BOSHQARISH VA DASTURLASH TEXNOLOGIYALARI

fanidan laboratoriya ishlarini bajarish uchun

USLUBIY KO‘RSATMALAR

I - qism

TOSHKENT – 2020

Tuzuvchilar: Sodiqova Sh.Sh., Bekturdiyev S.Sh.

«Ma’lumotlar bazasini boshqarish va dasturlash texnologiyalari» fanidan laboratoriya ishlarini bajarish uchun uslubiy ko‘rsatma. – Toshkent, ToshDTU: 2020. 74 b.

Uslubiy ko‘rsatmada «Ma’lumotlar bazasini boshqarish va dasturlash texnologiyalari» fani bo‘yicha laboratoriya ishlarini bajarishga doir ko‘rsatmalar hamda ma’lumotlar bazasini yaratish va ma’lumotlar bazasini boshqarish tizimi bilan ishlashga oid misollar keltirilgan.

Ushbu uslubiy ko‘rsatma oliy o‘quv yurtlarining 5330200 - «Informatika va axborot texnologiyalari» (sanoat ishlab chiqarishda) va 5312700 - «Intellektual muhandislik tizimlari» ta’lim yo‘nalishlari talabalari, hamda unga turdosh mutaxassislik talabalari uchun mo‘ljallangan.

Islom Karimov nomidagi ToshDTU ilmiy-uslubiy kengashi qaroriga muvofiq chop etildi.

Taqrizchilar:

Xudoyberganov M.U. – O‘zMU, «Hisoblash matematikasi va axborot tizimlari» kafedrasi dotsenti, f.m.f.n.

Sagatov M.V. – ToshDTU «Axborot tizimlari» kafedrasi mudiri, texnika fanlari doktori, professor.

KIRISH

O‘zbekiston Respublikasi ta’lim tizimi islohotlarining keng ko‘lamda olib borilishi, ta’lim jarayonidagi qo‘llaniladigan har bir texnologiya, sanoatga joriy etishga mo‘ljallangan topshiriqlar talabalarning ijodiy fikrlashga va kasbiy rivojlanishiga qaratilgan bo‘lishi lozim. Uslubiy ko‘rsatma Kadrlar tayyorlash milliy dasturi, “Ta‘lim to‘g‘risida”gi O‘zbekiston Respublikasi qonunining qoidalariga muvofiq holda bo‘lib, milliy tajribaning tahlili va ta’lim tizimidagi jahon miqyosidagi yutuqlar asosida tayyorlangan. Bugungi kunda talabalarga har bir fandan nazariy bilimlarni amaliyotga tatbiq etishni mukammal o‘rgata oladigan o‘quv qo‘llanmalarning mavjudligi muhim masalalardan biridir.

Ma’lumotlar bazasini boshqarish va dasturlash texnologiyalari fani sanoatning barcha sohalarini raqamli texnologiya asosida yangilashga o‘tish davrida dolzarb ahamiyat kasb etib kelmoqda.

Har bir sanoat korhonalarida turli hildagi ma’lumotlarga ishlob berishga to‘g‘ri keladi. Ma’lumotlarni saqlash, ularga tog‘ri ishlov berish va ulardan foydalanish uchun ma’lumotlar bazasini boshqarish va dasturlash texnologiyalari fanini chuqur o‘zlashtirish maqsadga muvofiqdir.

Mazkur uslubiy ko‘rsatmada ma’lumotlar bazasi jadvallarini yaratishning dastlabki qadamlaridan boshlab to so‘rovlar yaratib, ular asosida kerakli axborotga ega bo‘lish jarayonlari batafsil yoritib berilgan. Har bir amaliyot ishini bajarish uchun namuna sifatida misollar keltirilan va yechish usullari rasmlar orqali ko‘rsatib berilgan.

1-laboratoriya ishi

Ma‘lumotlar bazasini boshqarish tizimlarni o‘rnatish va sozlash.

Ishdan maqsad: ma‘lumotlar bazasini boshqarish dasturiy vositalari bilan ishlar, sozlash va tahrirlash ko‘nikmalariga ega bo‘lish.

Masalani qo‘yilishi: ma‘lumotlar bazasini boshqarish vositalarnini o‘rnatish va sozlashni o‘rganish.

Uslubiy ko`rsatmalar:. Zamonaviy MB texnologiyasida MB.ni yaratish, unga xizmat ko‘rsatish va foydalanuvchilarning MB bilan ishlashiga imkon yaratish maxsus dasturiy uskunalar yordamida amalga oshiriladi. Bunday dasturiy uskunalar majmuasi ma‘lumotlar bazasini boshqarish tizimlari (MBBT) deb ataladi.

MBBT — MB.ni yaratish, uni dolzarb holatda ushlab turish, kerakli axborotni topishni tashkil etish va boshqa xizmat ko`rsatish uchun zarur bo‘ladigan dasturiy va til vositalari majmuasidir.

MBBT misoli sifatida quyidagilarni keltirish mumkin:

- DBASE dasturi;
- Microsoft Access;
- Microsoft Fox Pro for DOS;
- Microsoft Fox Pro for WINDOWS;
- Microsoft MySQL;
- Paradox for DOS;
- Paradox for WINDOWS.

MB bilan ishlashga kirishishdan oldin ma‘lumotlarni tasvirlash modelini tanlab olish kerak. U quyidagi talablarga javob berishi lozim:

- ✓ axborotni ko`rgazmali tasvirlash;
- ✓ axborotni kiritishda soddalik;
- ✓ axborot bazasiga kiritilgan ma‘lumotdan foydalanish imkoniyatining mavjudligi;
- ✓ MB.ning ochiqligini ta`minlash (yangi ma`lumotlar va maydonlar qo`shish, ularni olib tashlash imkoniyatlari va hokazo).

MB bitta yoki bir nechta modellarga asoslangan bo‘lishi mumkin. Ular qanday modelga o‘zining xossalari (parametrlari) bilan tavsiflanuvchi obyekt sifatida qarash mumkin. Shunday obyekt ustida biror amal (ish) bajarsa bo‘ladi. MB modellarining uchta asosiy turlari mavjud: ***relyatsion, iyerarxik*** va ***semantik*** tarmoq.

Relyatsion (lotin tilidagi relatio — munosabat so`zidan olingan) modelda ma`lumotlarni saqlash uni tashkil etuvchi qismlari orasidagi munosabatlarga asoslangan. Eng sodda holda u ikki o`lchovli massiv yoki jadvaldan iborat bo`ladi. Murakkab axborot modellari ana shunday jadvallarning o`zaro bog`langan to`plamidan iborat.

MB.ning ***iyerarxik*** modeli pastki pog`onadagi yuqori pog`onadagiga bo`ysinish tartibida joylashgan elementlar to`plamidan iborat bo`ladi va ag`darilgan daraxt(graf)ni tashkil etadi. Ushbu model satx, tugun, bog`lanish kabi parametrlar bilan tavsiflanadi. Uning ishlash tamoyili shundayki, quyi satxdagi bir nechta tugunlar bog`lanish yordamida yuqoriroq satxdagi bitta tugun bilan bog`langan bo`ladi. ***Tugun*** — bu ierarxiyaning berilgan satxida joylashgan elementning axborot modelidir.

MB.ning ***semantik*** tarmoq modeli iyerarxik modelga o`xshashdir. U ham tugun, sath, bog`lanish kabi asosiy parametrlarga ega. Lekin semantik tarmoq modelida turli sathdagi elementlar orasida «erkin», ya`ni «har biri hamma bilan» ma`noli bog`lanish qabul qilingan.

Ko`pchilik MB.lar jadval tuzilmasiga ega. Unda ma`lumotlar manzili satr va ustunlar kesishmasi bilan aniqlanadi. MB.da ustunlar — ***maydonlar***, satrlar esa ***yozuvlar*** deb ataladi. ***Maydonlar*** MB.ning tuzilmasini, ***yozuvlar*** esa, unda joylashgan ma`lumotlarni tashkil etadi.

Maydonlar — MB. tuzilmasining asosiy elementlaridir. Ular ma`lum xususiyatlarga ega bo`ladilar. Ular qanday maydonning asosiy xususiyati uning uzunligidir. Maydon uzunligi undagi belgilar soni bilan ifodalanadi.

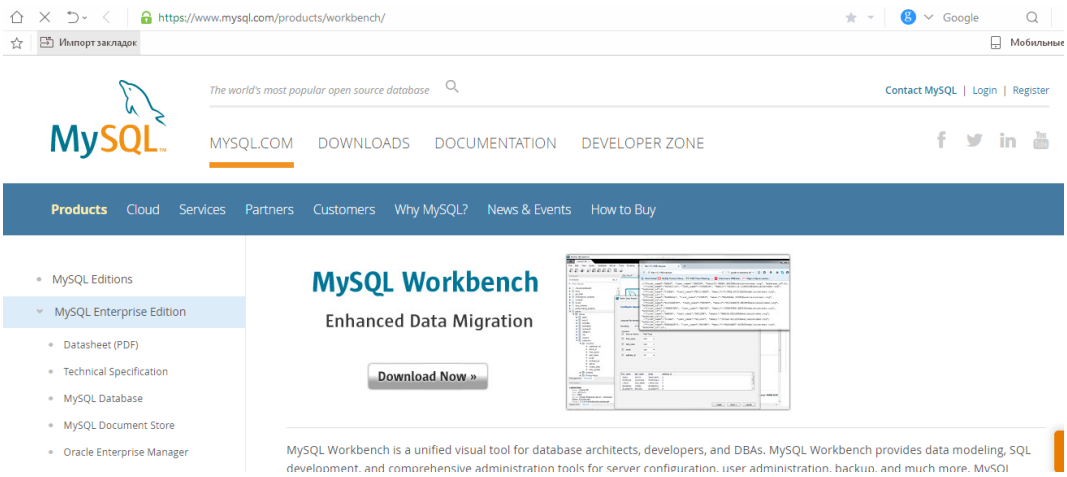
Maydonning yana bir xususiyati, uning nomidir. Maydonda uning nomidan tashkari yana imzo xususiyati ham mavjud. ***Imzo*** — ustunning sarlavhasida aks

ettiriladigan axborotdir. Uni maydon nomi bilan aralashtirib yubormaslik lozim. Agar imzo berilmagan bo`lsa sarlavhada maydon nomi yozib qo`yiladi. Turli tipdagi maydonlar turli maqsadlarda ishlatiladi va turli xossalarga ega bo`ladi.

Ma‘lumotlar bazasi haqida bilimlarga ega bo‘ldik endigi navbatda dasturiy vositalarni o‘rnatish va ular bilan ishlash ko‘nikmalarini hosil qilamiz.

Yaratrayotgan bazamizning tuzilishi haqida , jadval maydonlari xususiyatlari haqida to‘xtalsak. Dasturiy vosita sifatida MySQL WorkBrench va Open Server larni tanlaymiz. MySQL WorkBrench dasturi yordamida biz bazamizning umumiy strukturasini hosil qilishimiz mumkin. Ma‘lumotlarni qayta ishlash uchun esa Open Server ni yordamchi dasturidan foydalanamiz.

Dastlab MySQL Workbrench dasturiy ta‘minotini o‘rnatib olishimiz kerak bo‘ladi. Buni uchun https://www.mysql.com/products/workbench/ havolaga murojat qilib dasturiy ta‘minotni yuklab olamiz.(1.1-rasmda keltirligan oyna ochiladi)



1.1-rasm. MySQL WorkBrench dasturi havolasi

MySQL Workbench – bu vizual ma‘lumotlar bazasini yaratish uchun vosita bo‘lib, integratsiyalashgan dizaynda MySQL ma‘lumotlar bazasi tizimini yagona muhitda modellashtirish, yaratish tizimi.

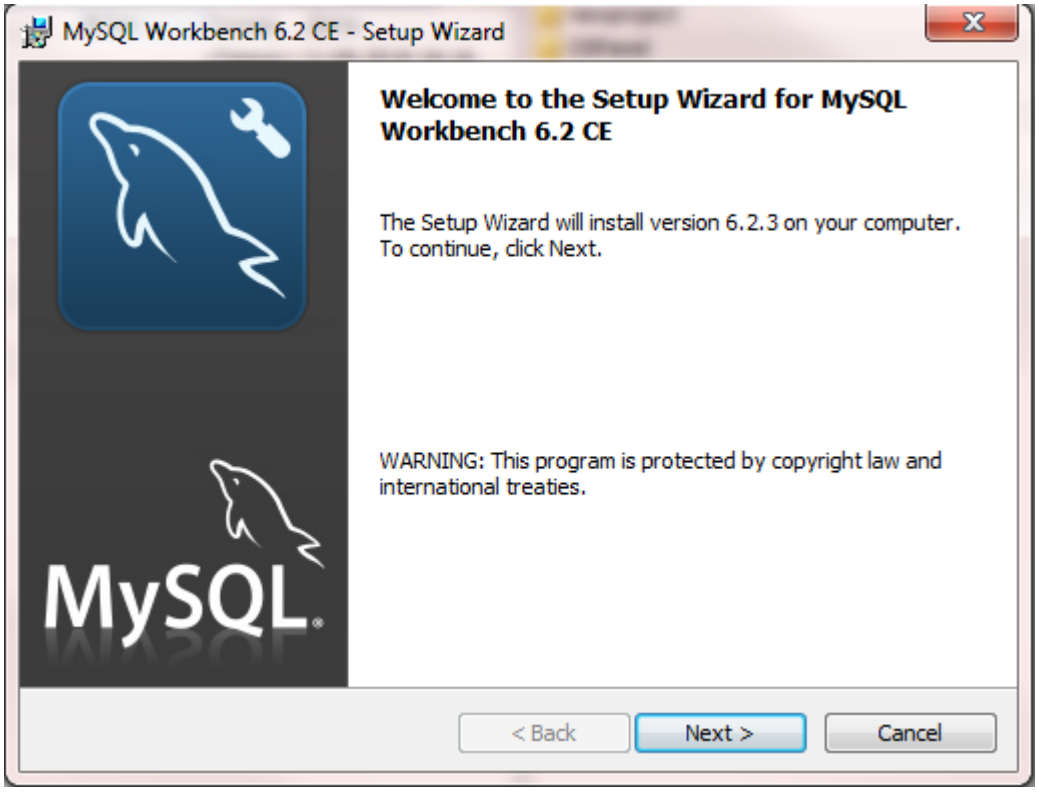
MySQL Workbench ikki veriyani taklif etiladi:

1. **Community Edition** - bepul GNU GPL ostida tarqatiladigan versiya;
2. **Standard Edition** - pullik vaersiya. Bu varsiya ishlab chiqaruvchi va MB administratorlarini ish samaradorligini oshiradanidan qo‘shimcha funksiyalarni o‘z ichiga olgan.

Dastur xususiyatlari:

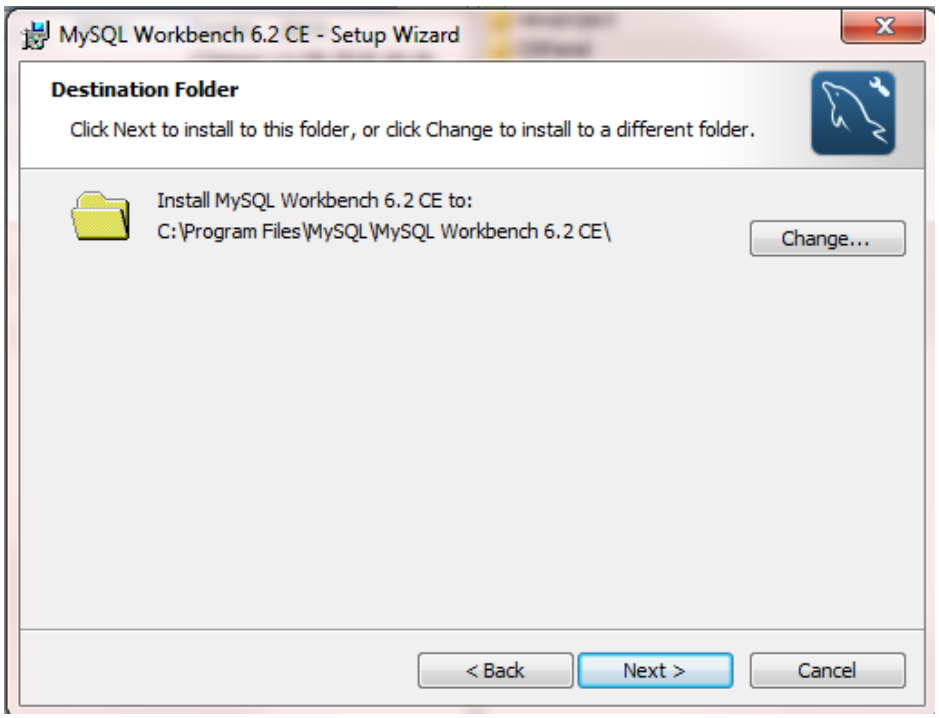
1. Ma‘lumotlar bazasi modelini grafik shaklida tasvirlash.
2. Jadvallar o‘rtasidagi munosabatlarni yaratish bilan bog‘liq bo‘lgan ko‘plab munosabatlarni o‘rnatish aniq va funksional mexanizmiga ega.
3. SQL so‘rov muharriri ularni darhol serverga yuborishga va jadval shaklida javob olishga imkon beradi.
4. Vizual rejimda jadvaldagi ma‘lumotlarni tahrirlash imkoniyati.
5. Reverse Engineering - mavjud ma‘lumotlar bazasi serveridan jadvallar tuzilishini tiklash.

Dastlab dasturni o‘rnatish uchun rasmiy saytdan yuklab olish kerak. Dasturni yuklab olish uchun saytda ro‘yxatdan o‘tish zarur. Dasturning quyidagi versiyasini o‘rnatamiz: «**mysql-workbench-6-2-3-64-bit-multi-win**».



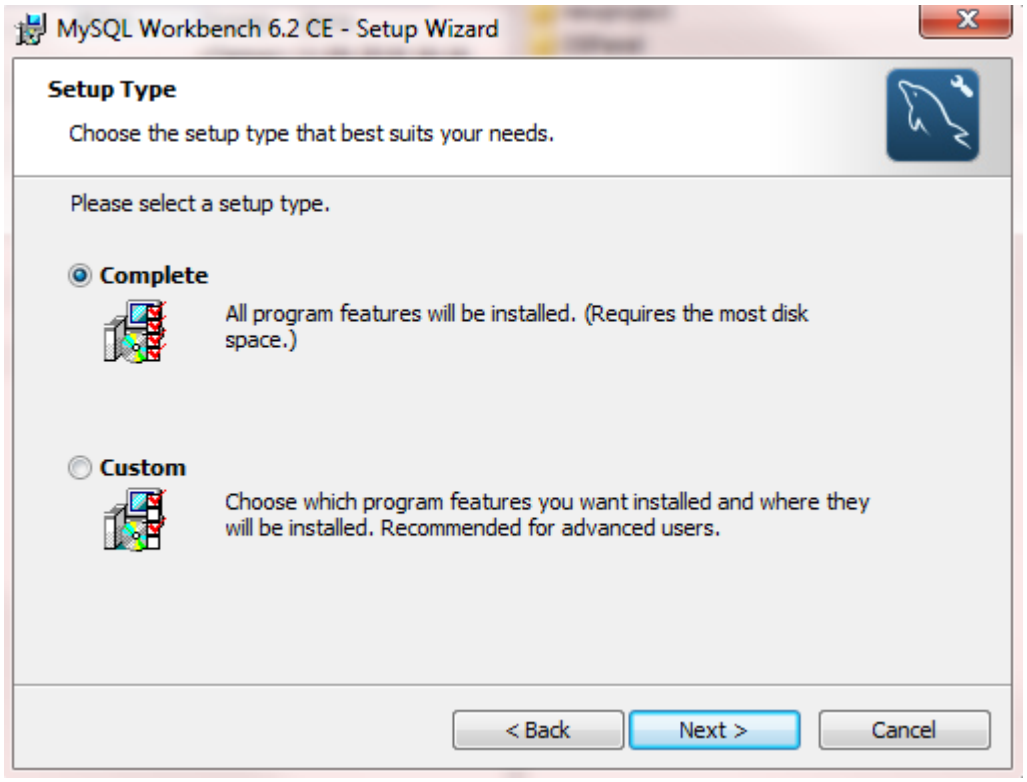
1.2-rasm. MySQL WorkBench 6.2 veriyasi

Keying qadamda dastur o‘rnatilishi kerak bo‘ladigan joy tanlanadi va davom etish tugmasi bosiladi.



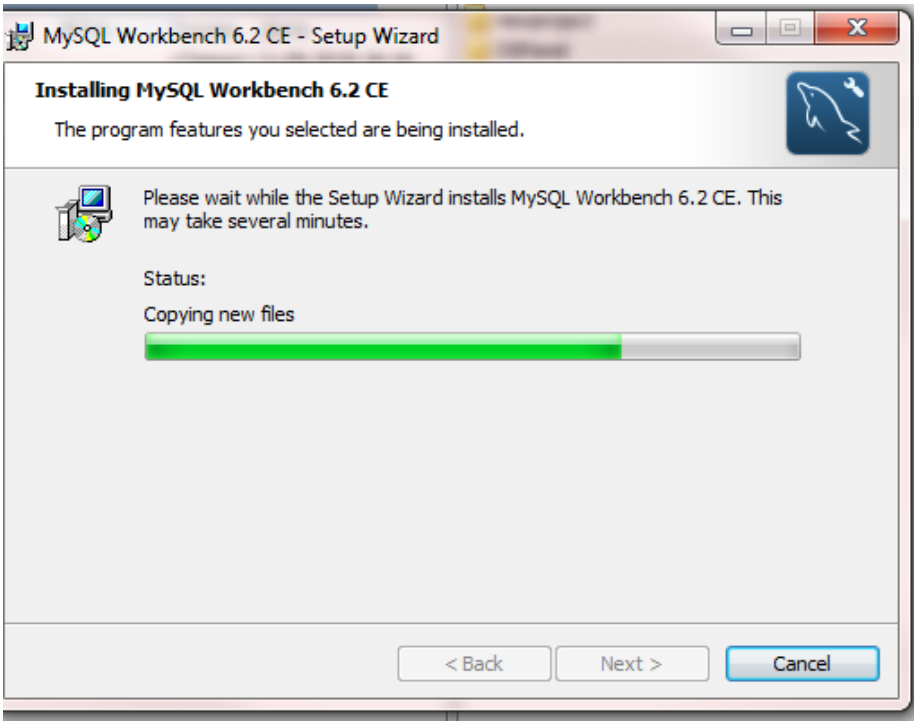
1.3-rasm. MySQL WorkBrench o‘rnatiladigan joyni tanlash

Kerakli turi tanlanadi dasturning. Bu joyda avtomat tanlanganini o‘zgartirmaslik tavsiya qilinadi.



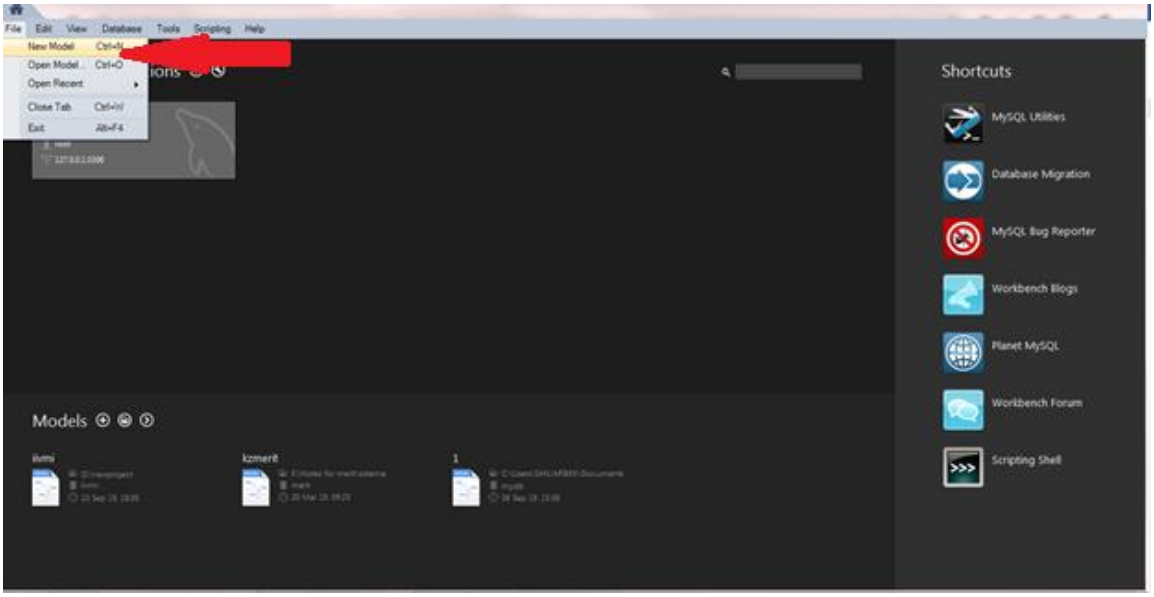
1.4-rasm. MySQL WorkBrench zarur turini tanlash

Tanlangan turdagi MySQL WorkBrench ko‘rsatilgan joyimizga o‘rnatilishi boshlanadi.



1.5-rasm. MySQL WorkBrench o‘rnatilishi

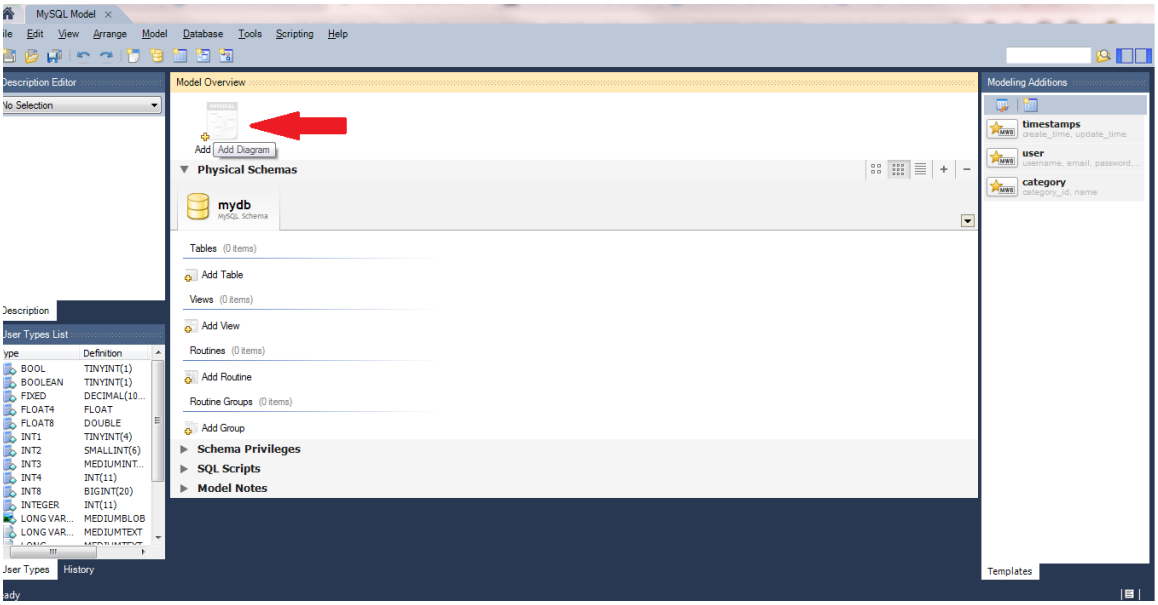
Shundan keyin dasturiy ta‘minot o‘rnatilishi yakuniga yetadi va quyidagi oyna ochiladi.



1.6-rasm. MySQL WorkBrench birinchi oynasi

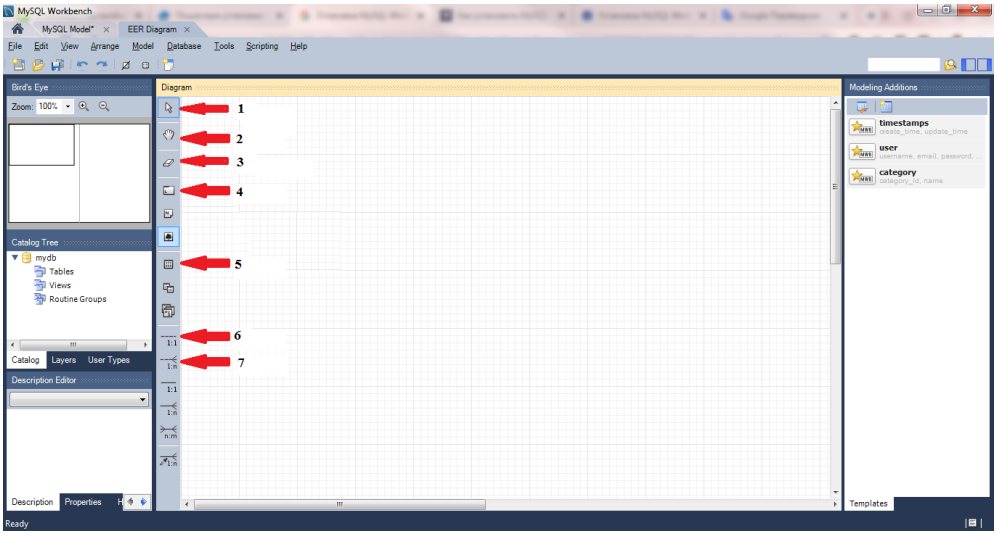
Shundan so‘ng tanlangan predmet soha bo‘yicha bazamizning homaki nusxasi asosidagi asl shaklini yaratishimiz mumkin bo‘ladi.

Buni uchun tepadagi menyular satridan ***File*** bo‘limidan ***New Model*** buyrug‘i tanlandi yoki klaviaturadan ***Ctrl N*** tugmalari bosiladi.



1.7-rasm. MySQL WorkBrench.da asosiy oynasi

Undan keying navbatda ***EER Diagram*** tugmasi tanlanadi. Bunda yangi bosh oyna va chap tomonda uskunalar paneli ochiladi.



1.8-rasm. EER Diagram oynasi

1 - yaratilgan jadvallarni belgilash uskunasi.

2 - yaratilgan jadvallarni siljitish uchun ishlatiladigan uskuna.

3 - ortiqcha kiritilganlarni o‘chirish uskunasi

4 - yangi qatlam qo‘shish uskunasi.

5 - yangi jadval qo‘shish uskunasi.

6,7 - jadvallarni bir biriga bog‘lash uskunalari.

Yuqoridagi uskunalardan foydalangan holda jadvallar hosil qilishimiz va ularni bir-biri bilan bog‘lashligimiz mumkin bo‘ladi.(keyingi darslarda bular bilan to‘laroq tanishamiz)

Keyingi masalamiz ikkinchi dasturimizni o‘rnatish va undan foydalanish haqida bo‘ladi. Biz foydalanadigan dastur bu ***Open Server.***

Open Server – bu server platforma nusxasining jamlagan dasturiy ta‘minotdir. Web dasturchilar uchun juda qulay bo‘lgan dasturiy ta‘minotlar majmuasiga ega hisoblanadi.

Dasturiy ta‘minot to‘plami qulaylik, ko‘p funksional interfeysga egalik, tarkibiy qismlarni boshqarish va sozlashning barcha imkoniyatlarning jamlaganligi bilan xarakterlanadi. Ushbu platforma web-loyihalarni ishlab chiqish, tuzatish va sinovdan o‘tkazish, shuningdek mahalliy tarmoqlarda web-xizmatlarni taqdim etish uchun keng qo‘llaniladi.

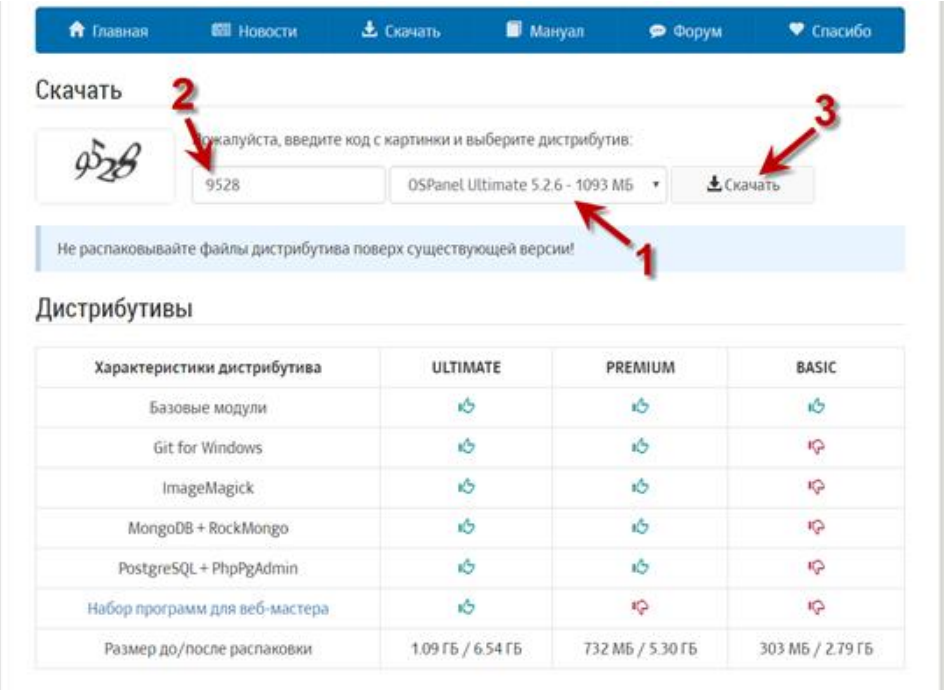
Bir so‘z bilan aytganda, Open Server to‘g‘ridan-to‘g‘ri shaxisy kompyuterda saytlar yaratishga imkon beradi. Saytdagi yangi modulni sinab ko‘rish, dizaynni o‘zgartirish va hokazolarni tekshirishga yordam beradi.

Open Server haqida to‘liq ma‘lumotni rasmiy web-sayt (https://ospanel.io/) da o‘qishingiz mumkin.

Xo‘sh, keling, Open Server.ni o‘rnatishni boshlaylik

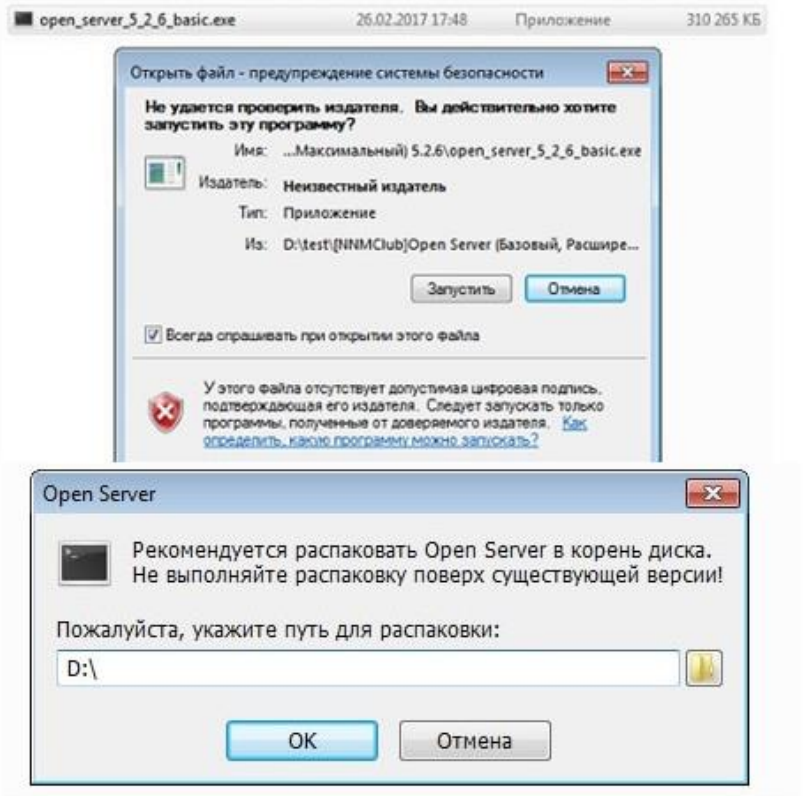
Avvalo, biz dasturni o‘zini yuklab olishimiz kerak. Buning uchun rasmiy web-saytni «Yuklab olish» (Скачать) bo‘limiga o‘tiladi, dasturning kerakli versiyasini tanlanadi (1).

Ta‘kidlash joizki, dasturiy mahsulotni uchta versiyasi mavjud: ***Ultimate, Premium, Basic,*** ularning o‘zaro qanday farq qilishi rasmiy saytdagi taqqoslash jadvalida aniq ko‘rinadi. Bu joyda ***ULTIMATE*** yuklab olingan. Davom etish uchun rasmda keltirilgan kodni kiritish (2) va yuklab olish tugmasini (3) boshish kerak bo‘ladi.



1.9-rasm. Open Server ni yuklab olish

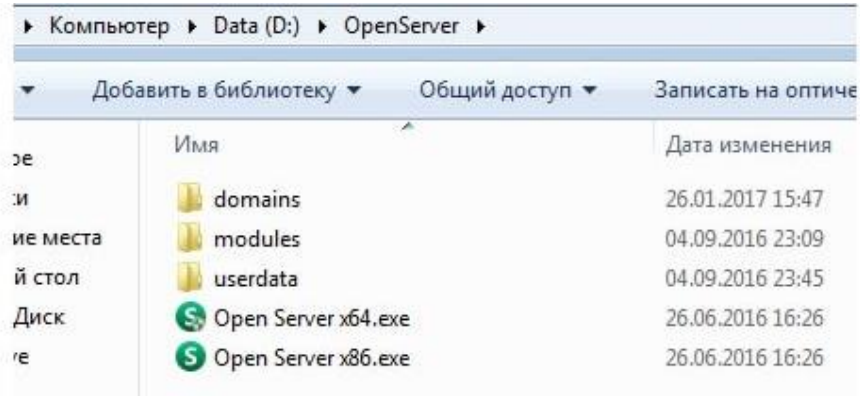
Dasturiy ta‘minot arxiv sifatida taqdim etiladi. Ishga tushirilganda Open Server paketi arxivdan chiqariladigan diskni ko‘rsatish kerak bo‘ladi



1.20-rasm. Open Serverni o‘rnatishni boshlash

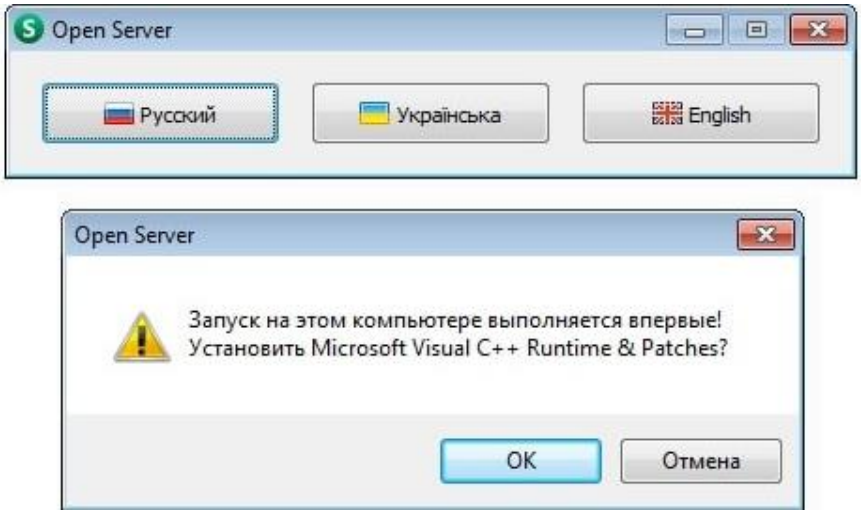
O‘rnatish joyini tanlanadi. Shuni yodda tutingki dasturiy ta‘minotning OpenServer deb nomlagan papkasini o‘zingiz yaratmang uni dasturnig o‘zi yaratadi.

O‘rnatishdan so‘ng Open Server papkasida bir nechta papkalar va yashil piktogrammalarga ega 2 ta fayl paydo bo‘ladi. Operatsion tizimingizga mos keladigan faylni ishga tushiring (32 bit uchun x86, 64 bit uchun x64). Agar siz bit nimaligini bilmasangiz, ikkala faylni ham sinab ko‘ring. Hech qanday dasturda ham, operatsion tizimda ham buzilishlar bo‘lmaydi.



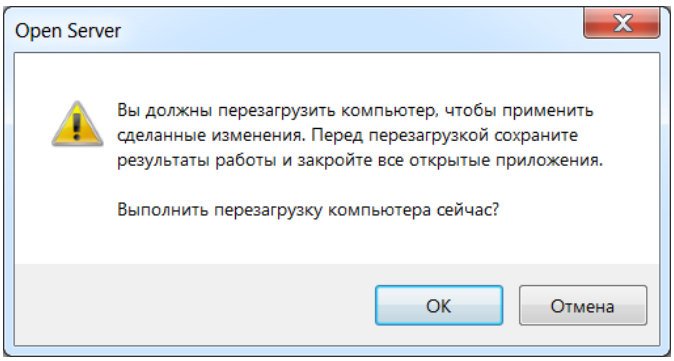
1.21-rasm. Open Server o‘rnatilgan papka

Birinchi ishga tushirishda tilni tanlash so‘raladi. Bundan tashqari, Open Server.ni birinchi ishga tushirishda Microsoft Visual C ++ qo‘shimchalarni o‘rnatish taklif qilinadi va shunda «Ok» ni bosish kerak.



1.22-rasm. Open Server.ni birinchi ishga tushirishda

Va bizda qoladigan oxirgi narsa, barcha o‘zgarishlar kuchga kirishi uchun kompyuterni o‘chirib yooqish kerak bo‘ladi. Buni uchun ekranda chiqqan so‘roqqa «Ok» ni bosish kifoya.

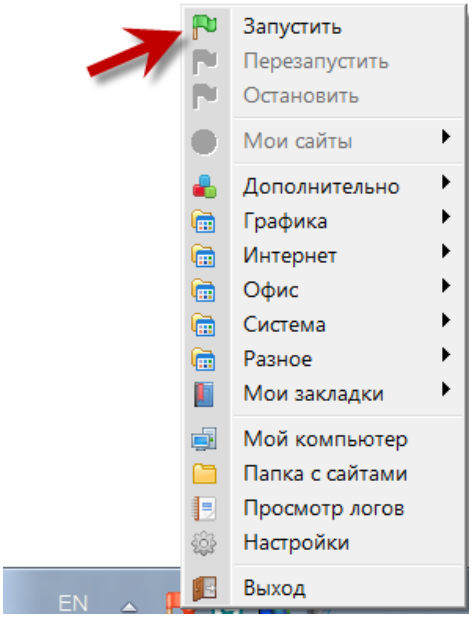


1.23-rasm. Open Server ni o‘rnatishni yakunlash

Kompyuter qayta ishga tushish bilan lokal server bilan ishlashni boshlash mumkin bo‘ladi. Open Server o‘rnatilgan papkadan ishga tushiriladi, shunda qizil bayroq ko‘rinishidagi belgi ekranning o‘ng pastki burchagida paydo bo‘lganligini ko‘rish mumkin. Bu degani, dastur faollashtirilgan, ammo serverning o‘zi hali ishga tushirilmagan.

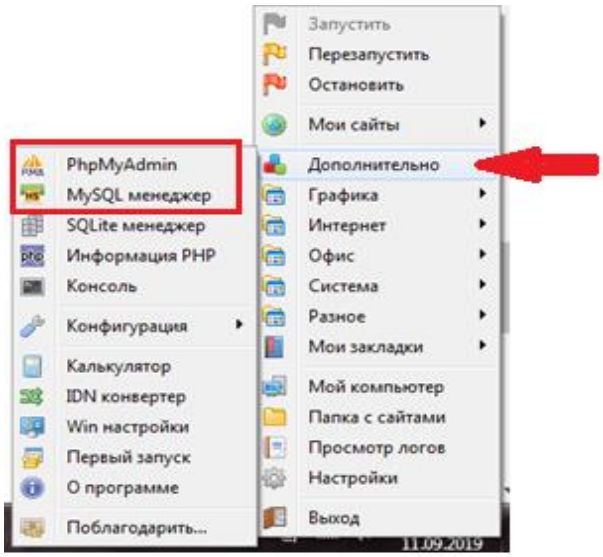
1.24-rasm.Open Server faollashtirilgan

Serverni ishga tushirish uchun bir sichqoncha tugmachasi bilan belgi bosiladi va ochilgan kontekst menyusidagi ***«Ishga tushirish» (Запустить)*** bandini tanlanadi. Lokal server ishga tushishi bilan bayroq yashil rangga aylanadi.



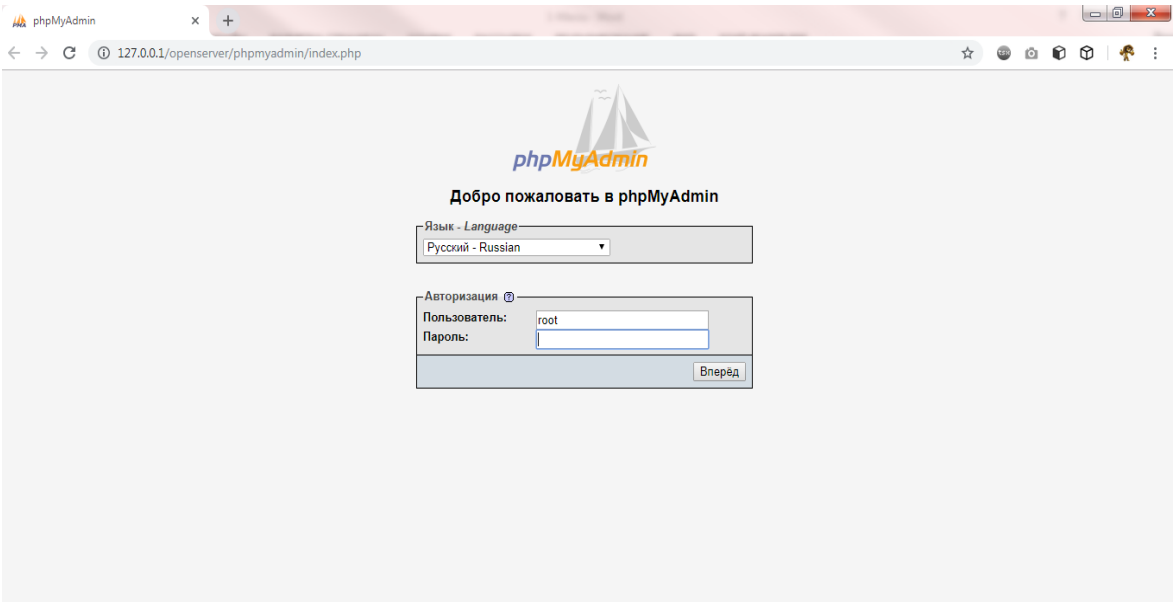
1.25-rasm. Open Serverni ishga tushirish

Lokal server ishga tushgandan keyin baza bilan ishlash uchun ***«Qo‘shimcha yuksiyalar» (Дополнительно)*** bo‘lmidan ***PhpMyAdmin*** yoki ***MySQL менеджер*** bandlari tanlanadi.



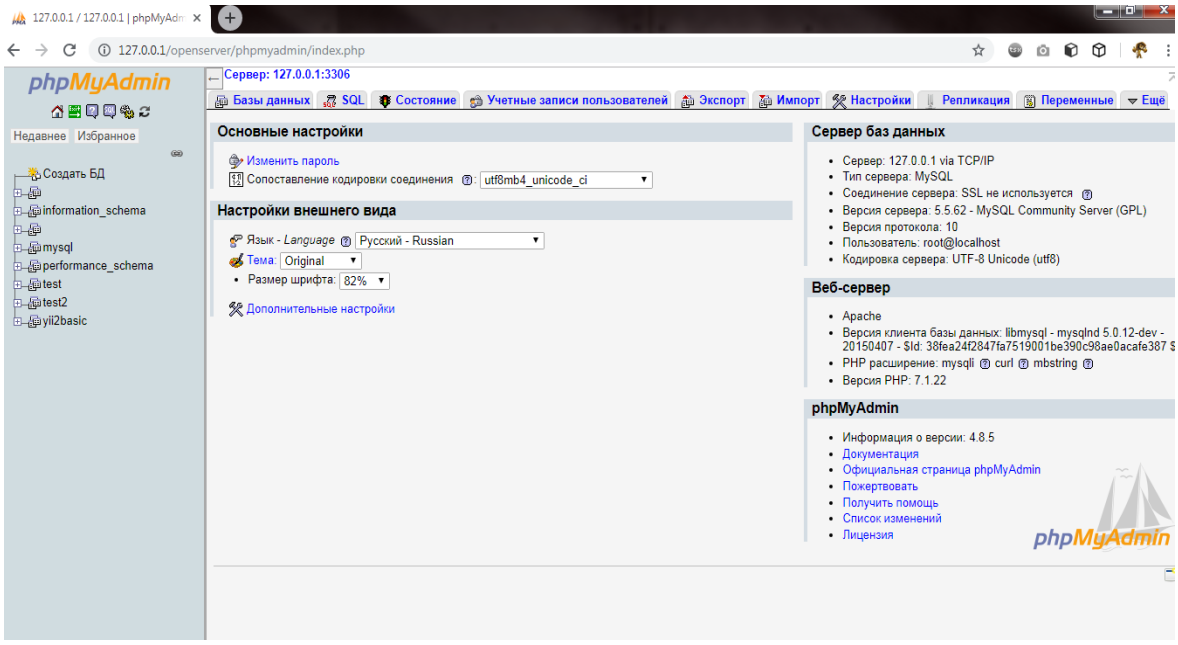
1.26-rasm. Open Serverdagi lokal bazani ishga tushirish

PhpMyAdmin bandi tanlangandan keyin web brauser ishga tushib unda 1.27-ramdagi havola hosil bo‘ladi. Bazaga kirish uchun foydalanuvchi va uning parolini so‘raydi. Foydalanuvchi nomiga ***root*** yoziladi va parol joyi bo‘sh qoladi (hech narsa yozilmaydi) ***«Davom etish» (Вперёд)*** tumasi bosilandi.



1.27-rasm. Lokal bazaga kirish oynasi

Oxirgi ochilgan oynada kompyuterdagi barcha lokal bazalar ro‘yxati namayon bo‘ladi. Bu joyda yangi baza yaratish, bazaga jadvallar qo‘shish, jadvallarni to‘ldirish mumkin bo‘ladi.



1.28-rasm. Komyuterdagi lokal bazalar oynasi

Nazorat savollari.

1. Qanday ma‘lumotlar bazasini boshqarish tizimlarini bilasiz ?
2. Ma‘lumotlarni tasvirlash modellari qanday talablari bor ?
3. MB modellarining asosiy turlari ?
4. MySQL WorkBrench haqida nimalar bilasiz ?
5. Open Server dasturining xususiyatlari ?

Topshiriqlar

Har bir amaliy ish uchun topshiriqlar quyidagi jadvalda keltirilgan bo‘lib, talaba keltirilgan obyektlardan birini tanlashi va shu obyekt uchun barcha amaliy vazifalarni bajarishi kerak.

1.	Dorixona
2.	Sug’urta kompaniyasi
3.	Internet do’kon
4.	Qurilish mollari do’koni

5.	Uy oldi sotdisi
6.	Kompyuter magazin
7.	Salon
8.	Avia kassa
9.	Avtosalon
10.	Maishiy texnika ustaxonasi
11.	Fermer xo’jaligi
12.	Kutubxona
13.	Kiyim kechak do’koni
14.	Oziq ovqat do’koni
15.	Omborxona
16.	Maktab
17.	Hayvonlar do’koni
18.	O’yinchoq magazini
19.	Universitet
20.	Yotoqxona
21.	Oshxona

2- laboratoriya ishi

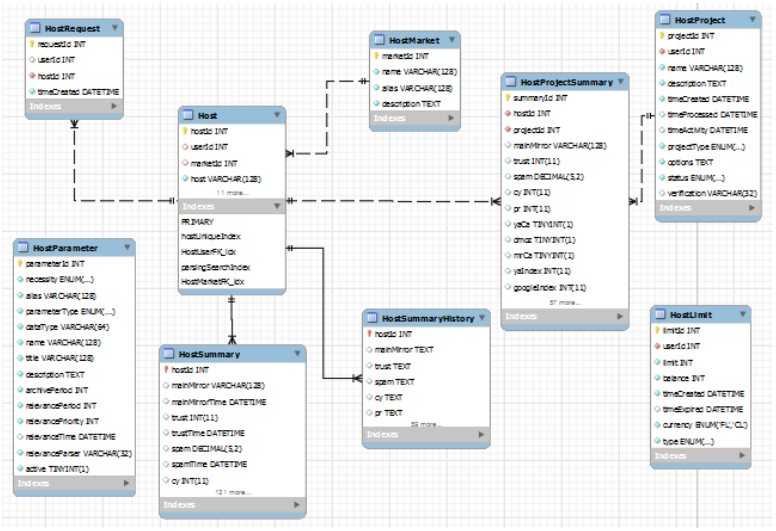
SQL tili asosida ishlaydigan tizimlar muhokamasi. Predmet soha strukturasini yaratish. (MySQL WorkBrench asosida)

Ishdan maqsad: ma‘lumotlar bazasini boshqarish dasturiy vositalari bilan ishlar, sozlash va tahrirlash ko‘nikmalariga ega bo‘lish.

Masalani qo‘yilishi: ma‘lumotlar bazasini boshqarish vositalarnini o‘rnatish va sozlashni o‘rganish.

Uslubiy ko`rsatmalar: Loyihalarning o‘sishi bilan dasturiy qismning murakkabligi oshadi, bir tomonidan ishlov beriladigan ma‘lumotlar miqdori, shuningdek ma‘lumotlar sxemasining murakkabligi oshadi. Kichik loyihalardan katta loyihalarga, ***cms***-dan ramkalarga o‘tish, ko‘pchilik MySQL-ga sodiq qoladi. Ammo ko‘p sonli jadval va havolalarga ega bo‘lgan murakkab ma‘lumotlar bazasini yaratish uchun ***PhpMyAdmin***-ning imkoniyatlari deyarli yo‘q. Shunday qilib, MySQL bilan ishlash uchun ajoyib bepul stol usti dasturi bo‘lgan MySQL Workbenchdan foydalanish tavsiya etiladi.

MySQL Workbench - *bu ma‘lumotlar bazasini loyihalash, modellashtirish, yaratish va ishlashni MySQL ma‘lumotlar bazasi tizimi uchun yagona muammosiz muhitga birlashtiradigan vizual ma‘lumotlar bazasini loyihalash vositasi*.

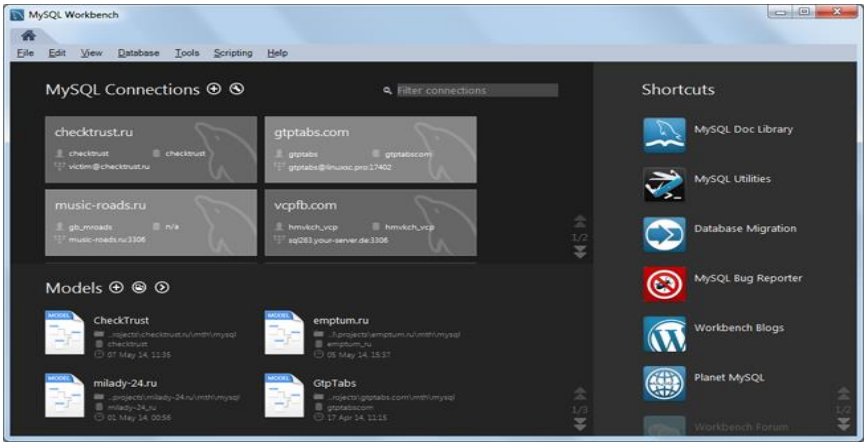


2.1-rasm. MySQL Workbench da yaratilgan model sxemasi

Shuni qayt etish kerakki loyihaning sxemalarini tez va qulay tarzada obyektlari loyihalash va ular o‘rtasidagi munosabatlar, sxemadagi o‘zgarishlar tez va oson amalga oshish, masofadagi server bilan osonlikcha sinxronlashtirish, EER

diagrammalarining grafik muharriri, ma‘lumotlar modelining katta sxemasini ko‘rish kabilarni bu dastur orqali bajarish juda qulay hisoblanadi.

Dasturning boshlang‘ich ekrani uning faoliyatining asosiy yo‘nalishlarini - ma‘lumotlar bazasi modellarini loyihalash va ularni boshqarish:

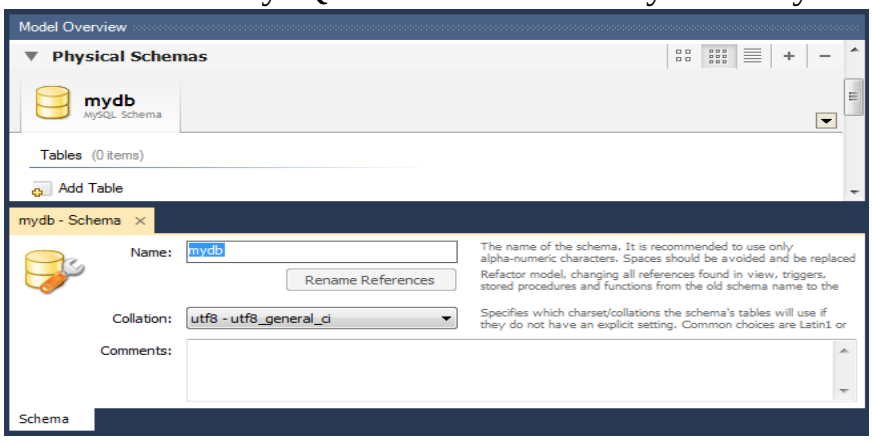


2.2-rasm. MySQL Workbench ishchi oynasi

Ekranning yuqori qismida loyihalarni MySQL serverlariga ulanishlarning ro‘yxati va ekranning pastki qismida so‘nggi ochiq ma‘lumotlar modellari ro‘yxati joylashdi. Ish odatda ma‘lumotlar sxemasini yaratish yoki mavjud tuzilmani MySQL Workbenchga yuklash bilan boshlanadi.

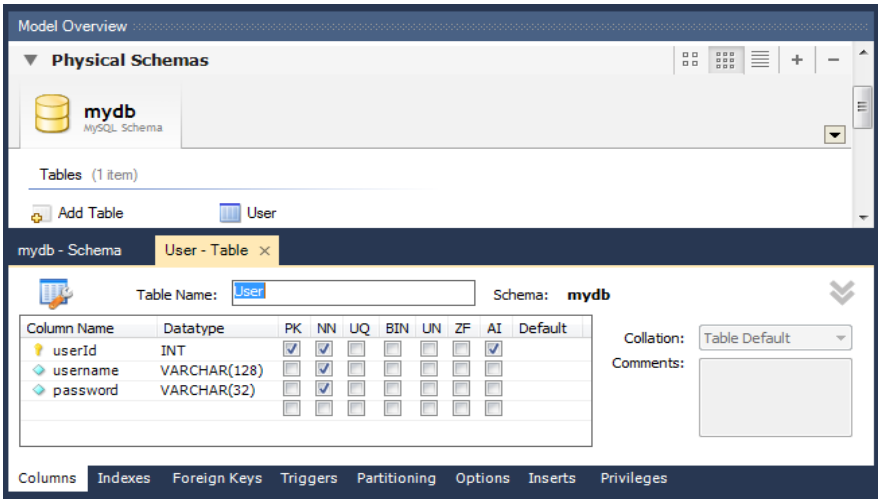
Modelni qo‘shish uchun «*Models*» sarlavhasi yonidagi ortiqcha belgisini bosish mumkin yoki «*File* → *Yangi model*» (*Ctrl* + *N*) ni tanlanadi:

2.3-rasm. MySQL Workbench model yaratish oynasi



Ushbu ekranda ma‘lumotlar bazasi nomini kiritiladi, standart kodlashni tanlanadi va agar kerak bo‘lsa, sharh maydonini to‘ldiriladi. Jadval yaratishni boshlash mumkin.

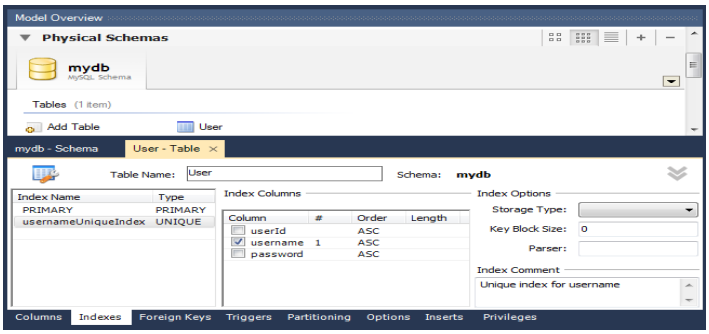
Loyihalar bazalari ro‘yxati va ma‘lumotlar bazasidagi jadvallar ro‘yxati «*Physical Schemas*» yorlig‘ida joylashgan. Jadval yaratish uchun «*+Add Tabl*» ustiga ikki marta bosiladi:



2.4-rasm. MySQL Workbench jadval yaratish oynasi

Maydonlar ro‘yxati va ularning xususiyatlarini tahrirlash uchun qulay interfeys ochiladi. Bu yerda maydon nomini, ma‘lumotlar turini, shuningdek maydonlar uchun turli xil atributlarni belgilash mumkin: maydonga birlamchi kalitni (PK-primary key), Null emasligi (NN-Not Null), ikkilik (BIN-binary), noyob (UQ-unique) va boshqalar maydoni uchun avto kattalashtirish (AI-auto incrementation), standart qiymat (Default).

Jadvallarni boshqarish interfeysida «Indekslar» yorlig‘ida jadval indekslarini qo‘shish, o‘chirish va tahrirlash mumkin:

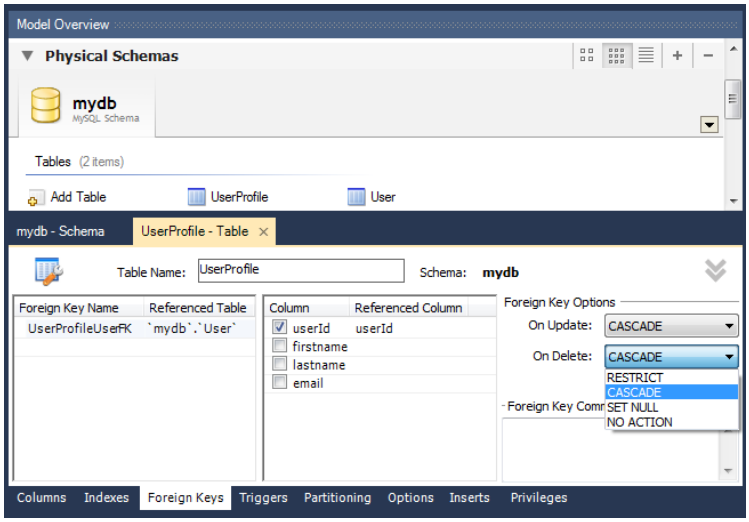


2.5-rasm. MySQL Workbench indeks o‘rnatish oynasi

Indeks nomini kiritiladi, uning turini tanlang, so‘ngra ushbu indeksda ishtirok etadigan maydonlar ro‘yxatini kerakli tartibda saralash tanlanadi. Tashqi kalitlarni va jadvallarni bog‘lanishlarini o‘rnatish faqatgina InnoDB jadvallari orqali amalga

oshiriladi. Jadvallar o‘rtashiga bog‘anishlarni ta‘minlash uchun har bir jadvalda ***«Foreign Keys»*** mavjud bo‘ladi.

Bog‘lanishni qo‘shish uchun ikkilamchi jadvalining ***«Foreign Keys»*** yorlig‘i ochiladi, tashqi kalit nomini kiriting va birlamchi jadval tanlanadi. Keyinchalik ***Column*** bo‘lmidan ikkilamchi jadvaldadi kalit maydoni tanlanadi, ***Referenced Column*** bo‘limi esa birlachi jadvalda tanlanadi. (maydonlar toifalari mos bo‘lishi shart) tashki kalitlar yaratilgandan keyin ikkilmchi jadvalda tegishli ko‘rsatkichlar avtomat ravishda hosil bo‘ladi.



2.6-rasm. MySQL Workbench ***«Foreign Keys»*** o‘rnatish oynasi

«Foreign Key Options» bo‘limida tegishli maydonni o‘zgartirganda (ON UPDATE) va brilamchi yozuvini o‘chirishda (ON DELETE) tashqi kalitning ishlashini sozlaymiz:

RESTRICT – o‘zgarishlar vaqtidagi xatolikni aniqlash/birlamchi yozuvni o‘chirish

CASCADE – tashqi kalit yozuvi o‘zgarganda birlamchi yozuvni o‘zgartirish, birlamchi yozuv o‘zgarganda ikkilamchi yozuvni o‘chirish.

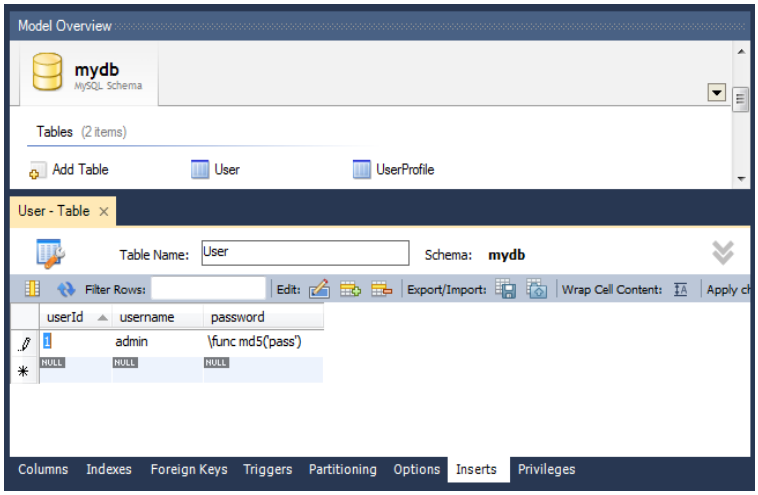
SET NULL – tashqi kalit o‘zgartirilganda NULL qiymat belgilash/ ikkilamchini o‘chirish.

NO ACTION – hech narsa qilmaslik, lekin bu RESTRICT ning anologi.

Yuqoridagi misolda birlamchi User jadvali bilan bog‘lanish uchun UserProfile ikkilamchi jadvaliga tashqi kalitni qo‘shildi. UserId maydonini tahrirlash va User

jadvalidagi elementlarni o‘chirishda, UserProfile jadvalidagi tegishli yozuvlar bilan avtomatik ravishda o‘zgarishlar yuz beradi.

Loyihani yaratishda ko‘pincha ma‘lumotlar bazasiga boshlang‘ich ma‘lumotlarini qo‘shish kerak bo‘ladi. Bular asosiy toifalar, admin foydalanuvchilar va boshqalar bo‘lishi mumkin. MySQL Workbench jadvalini boshqarish tizimida ***«Intersts»*** oynasida mavjud:

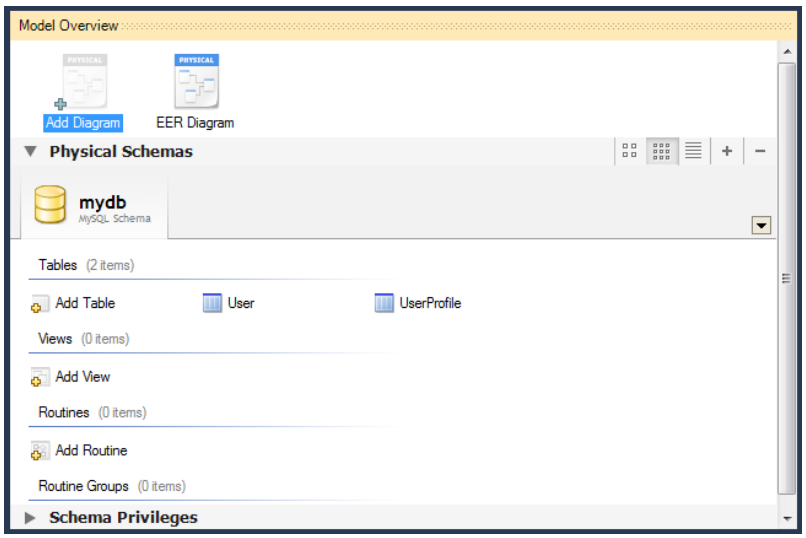


2.7-rasm. MySQL Workbench ***«Intersts»*** oynasi

Misollardan ko‘rinib turibdiki, agar ma‘lumotlar bazasiga yozishdan oldin MySQL ba‘zi funksiyalarini ma‘lumotlarga qo‘llash kerak bo‘ladi, bu quyidagi sintaksis yordamida amalga oshiriladi bu sintaksis *\func functionName(‘data‘)* masalan *\func md5(‘password‘).*

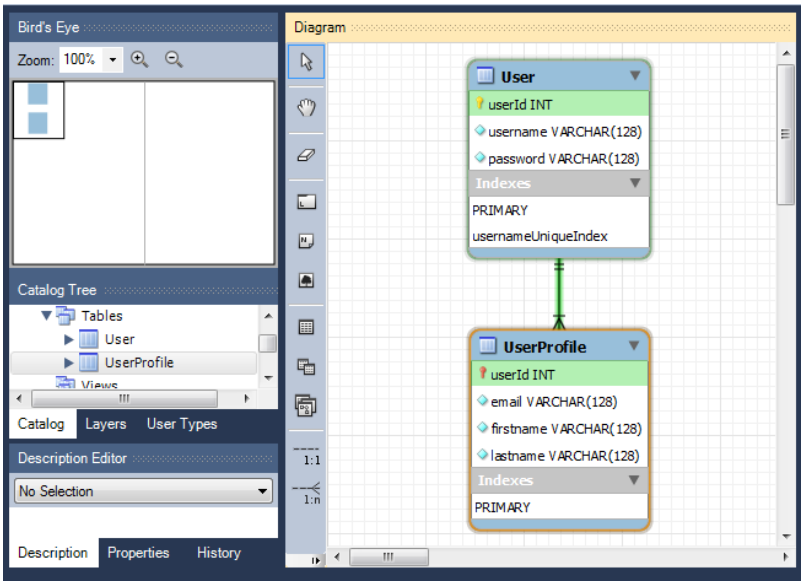
Ma‘lumotni kiritgandan so‘ng, siz *«Apply Changes»* ("O‘zgarishlarni qo‘llash") tugmachasini bosish orqali lokal ma‘lumotlar bazasida saqlash kerak.

EER sxemasini yaratish (диаграммы "сущность-связь) Ma‘lumotlar sxemasi, obyektlar va ularning o‘zaro aloqalarini grafik shaklida ko‘rsatish uchun MySQL Workbench-da EER diagramma muharriri mavjud. Ma‘lumotlar bazasini boshqarish ekranining yuqori qismida diagramma yaratish uchun *«+Add Diagram»* belgisini ikki marta bosiladi:



2.8-rasm. MySQL Workbench ***«Add Diagram»*** oynasi

Uning interfeysida jadvallar yaratish va tahrirlashingiz, ular orasidagi turli xil aloqalarni qo‘shish mumkin. Diagrammada oldindan mavjud bo‘lgan jadvalni qo‘shish uchun uni «Catalog Tree» panelidan tortib o‘tkazish kifoya.

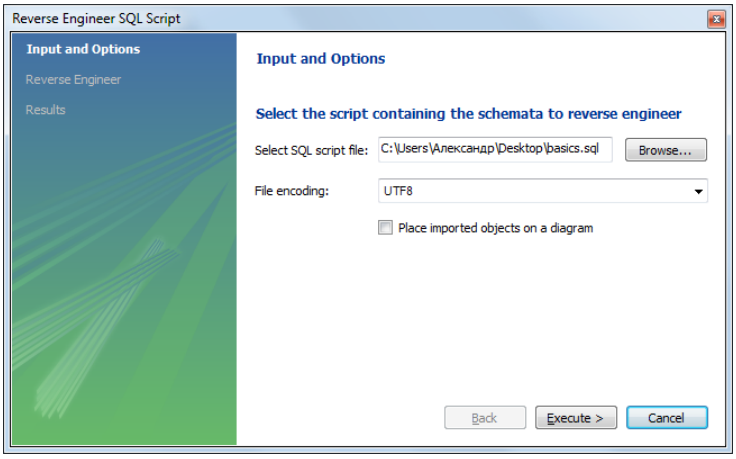


2.9-rasm. MySQL Workbench ***«Add table»*** oynasi

Ma‘lumotlar sxemasini grafik faylga eksport qilish uchun «*File → Export*» - tanlanadi, so‘ng variantlardan birini tanlanadi (PNG, SVG, PDF, PostScript File).

Agar tayyor ma‘lumotlar sxemasi bo‘lsa, uni keyinchalik ishlash uchun osonlikcha MySQL Workbench-ga kiritish mumkin.

Modelni SQL faylidan import qilish uchun *«File → Import → Reverse Engineer MySQL Create Script...»* - tanlanadi, so‘ngra kerakli SQL fayli tanlanadi va *«Execute>»*ni bosiladi.



2.10-rasm. MySQL Workbench sinxronizatsiya qilish oynasi

MySQL Workbench shuningdek ma‘lumot modelini to‘g‘ridan-to‘g‘ri masofaviy server bilan sinxronizatsiya qilishni ta‘minlaydi. Buning uchun MySQL-ga masofadan kirishni yaratish kerak, uni keying darsda ko‘rib chiqishni davom ettiramiz.

Nazorat savollari.

1. MySQL WorkBrench avfzaliklari haqida nimalarni bilasiz ?
2. Ma‘lumotlarni modellari tasvirlash qanday amalga oshiriladi ?
3. MySQL WorkBrench modelini bazaga sinxronisatsiya qilish tartibi ?

3- laboratoriya ishi

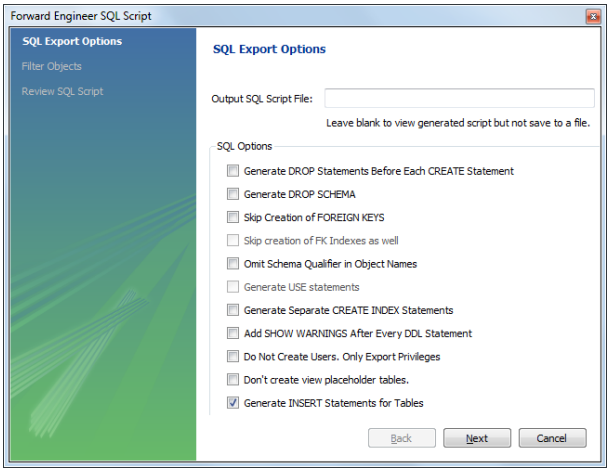
Loyihalanayotgan ma‘lumotlar bazasi vositalaridan foydalangan holda misollar yechish. (MySQL Workbench asoslari: masofaviy serverni ulash va sinxronlashtirish)

Ishdan maqsad: ma‘lumotlar bazasini boshqarish dasturiy vositasi yoramida masofaviy serverga ulanishni sozlash va serverdagi ma‘lumotlarni qayta ishlash, tahrirlash ko‘nikmalariga ega bo‘lish.

Masalani qo‘yilishi: ma‘lumotlar bazasini boshqarish vositalari yordamida masofaviy serverdagi ma‘lumotlarni qayta ishlash.

Uslubiy ko`rsatmalar: Oldingi darsda MySQL Workbench dasturini o‘rganish, ma‘lumotlar modelini ishlab chiqish va tashkilot bilan munosabatlarning EER diagrammasini yaratishning asosiy printsiplari misolini ko‘rsatdik. Endigi navbat MySQL Workbench-ni amalda qo‘llashga keldi, shuning uchun bu darsda masofaviy serverga ulanishni yaratish, mwb modelini serverga yuklash, ish davomida ma‘lumotlar sxemasi yangilanishlarini sinxronlashtirish, shuningdek MySQL Workbench-dan foydalanib MySQL-serverni boshqarishlarni keltiramiz.

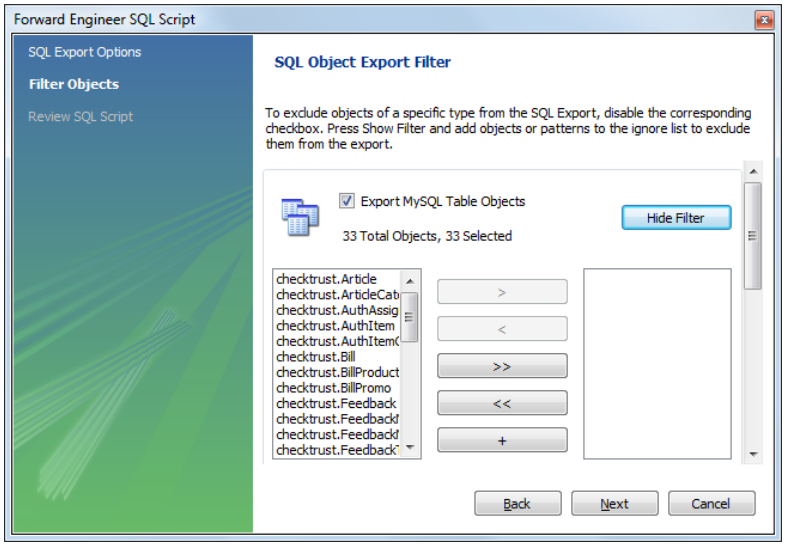
MySQL Workbench modelini SQL dump-ga eksport qilish. MySQL Workbench-dan serverga ma‘lumotlar sxemasini olishning eng tezkor usuli bu ***mwb*** modelining SQL dampini yaratishdir. Buni amalga oshirish uchun dasturda masofadan ulanishni yaratish shart emas, ammo bu usuldan strukturani va asosiy ma‘lumotlarni serverga bir martalik yuklashda foydalandagan afzal. Shunday qilib, modelni ochish uchun ***«File → Export → Forward Engineer SQL CREATE Script...» (Ctrl + Shift + G)*** ni tanlang:



3.1-rasm. MySQL Workbench modelini eksport qilish

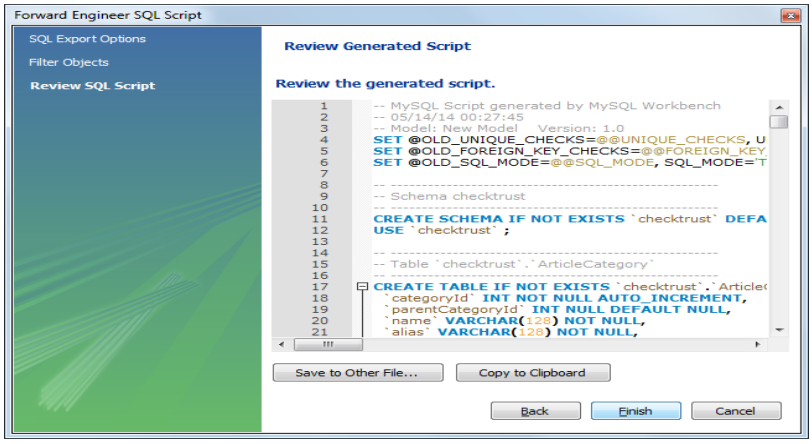
Agar siz faylga dampini yozishni xohlasangiz, *«Output SQL Script File»* maydonida faylga yo‘lni belgilang. Eksport sozlamalari oynasini ochamiz.

«Generate INSERT Statements for Tables» oynasidagi *damp* asosiy ma‘lumotlar, «Intsert» oynasida namunaviy jadvallarni tahrirlash interfeysi joylashgan. *«Next»* tugmachasini bosgandan so‘ng, eksport qilish mumkin bo‘lgan narsalar ro‘yxati ko‘rinadi. Jadvallarni eksport qilish uchun *«Export MySQL Table Objects»* -ni tanlang va ularni tanlab eksport qilish uchun *«Show Filter»* ni bosiladi va kerak bo‘lgan jadvallarni tanladani:



3.2-rasm. Jadval ob‘yektlarini eksport qilish

«Next» tugmachasini bosish orqali oynada tayyor SQL skriptni ko‘rinadi, uni *«Copy to Clipboard»* tugasni bosib nusxa qilib olib biron-bir faylga yozib olish mumkin.



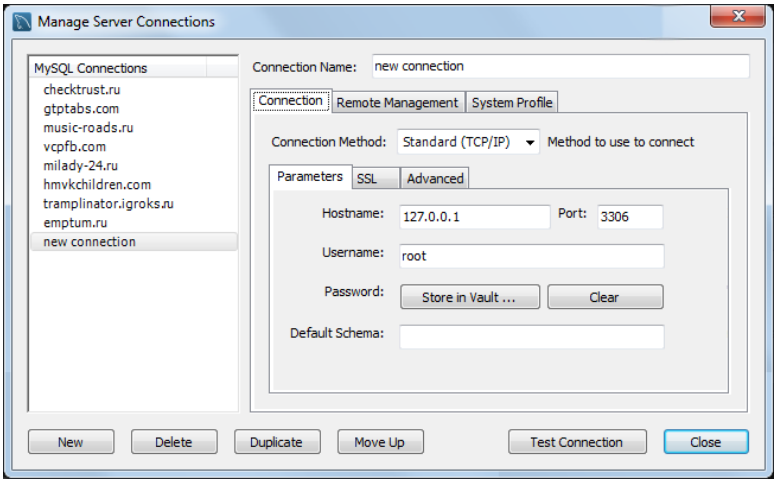
3.3-rasm. SQL skriptni ko‘rinishi

Ma‘lumotlar bazasi tuzilishi va ma‘lumotlar serverga yetib borishi uchun juda ko‘p usullar mavjud va ulardan eng keng tarqalgani ***PHPMyAdmin*** orqali importdir.

MySQL serveriga masofadan ulanishni yaratishni boshlaymiz. Masofaviy serverga ulanish uchun asosiy menyular satridan ***«Database → Manage Connections...»*** boʻlmiga oʻtiladi va yangi ochilgan oynadan ***«New»*** tugmasi bosiladi. MySQL Workbench serverga ulanishning uchta usulini taklif etadi: foydalanuvchi orqali ***toʻgʻridan-toʻgʻri*** ulanish, bu MySQL-ga masofadan kirish huquqini beradi (odatda bunday foydalanuvchilarning kirishlari IP boʻyicha cheklangan boʻladi), ***socket /pipe*** ulanish (*socket* fayli Unix uchun yoki *pipe* Windows uchun), shuningdek, ***ssh-tunnel*** orqali ulanish (ssh ulanishda MySQL foydalanuvchisi tegishli huquqlari belgilanadi).

Masofaviy serverga ulanish variantlarini koʻrib chiqamiz: Ulanish uchun dialog oynasida *«Standard: TCP/IP»* ulanish turini tanlaymiz:

3.4-rasm. MySQL Workbench serverga ulanish



•***«Host»*** maydoniga MySQL server manzilini yoki sayt manzilini kiriting (agar MySQL server veb-serverning oʻzida joylashgan boʻlsa).

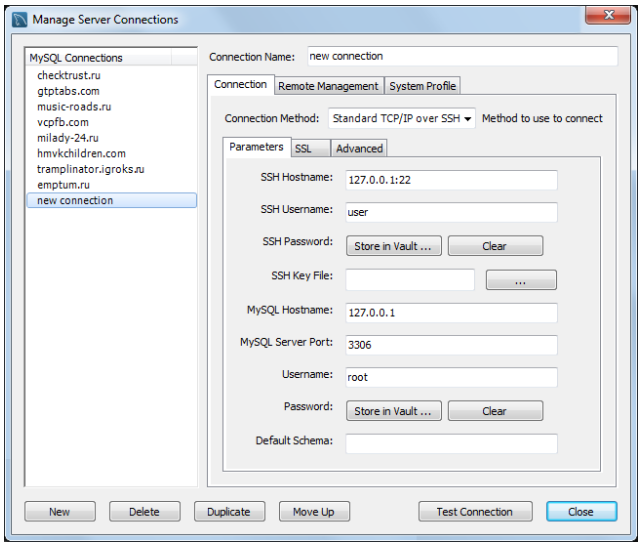
•***«Port»*** koʻpincha 3306 ga teng boʻladi.

•MySQL foydalanuvchi nomini (***«Username»***), parolni (***«Password»***) va maʻlumotlar bazasining nomini (***«Default Schema »***) kiritiladi.

•Ulanishni yaratgandan soʻng, *«Test Connection»* (Ulanishni tekshirish) tugadi bosiladi va ekranda *«Connection parameters are correct.»* (Ulanishning parametrlari toʻgʻri) xabarini chiqishi kutiladi.

Agar biror narsa noto‘g‘ri bo‘lsa, foydalanuvch masofadan kirish huquqiga egaligi yoki yo‘qligini, shuningdek, ushbu foydalanuvchi uchun ruxsat berilgan IP manzilini kompyuterda mavjudligini ro‘yxatda tekshirish kerak.

SSH tuneli orqali (TCP/IP ustida SSH orqali). Bunda ulanish uchun dialogda ***«Standart: TSH / IP orqali SSH»*** ulanish turini tanlang:



3.5-rasm. SSH tuneli orqali bog‘lanish

✓ Bu yerda web-serverning manzilini kiritishimiz kerak (***«SSH Hostname»)*** (agar kerak bo‘lsa, port ikki nuqta orqali ko‘rsatilgan, masalan, ***«linuxsc.pro:18752»)***

✓ SSH foydalanuvchi nomi va parolini kiritish (***«SSH Username»*** va ***«SSH Password»)***

✓ MySQL serverining manzili veb-serverga nisbatan kiritilishi kerak, ya‘ni agar MySQL va veb-server bitta kompyuterda bo‘lsa, ***«MySQL Hostname»*** maydonida ***«127.0.0.1»*** ni qoldiriladi.

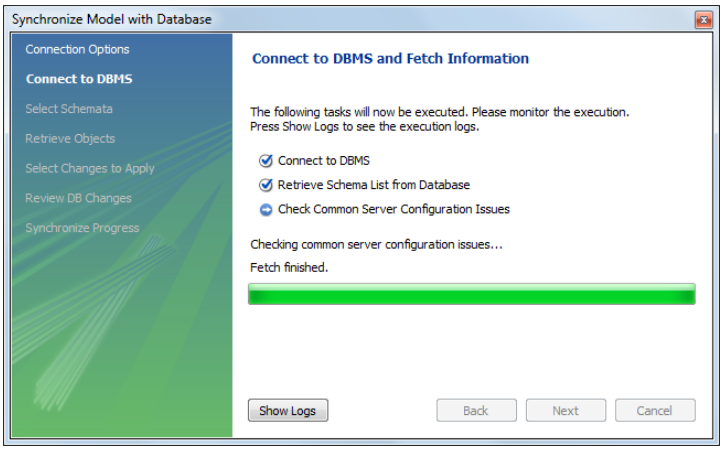
✓ ***«MySQL Server Porti»*** standart bo‘yicha 3306

✓ ***«Username»*** va ***«Password»*** maydonlarida MySQL foydalanuvchisi foydalanuvchi nomi va paroliga kiritiladi.

✓ ***«Test Connection»*** tugmachasi yordamida ulanishni tekshiriladi

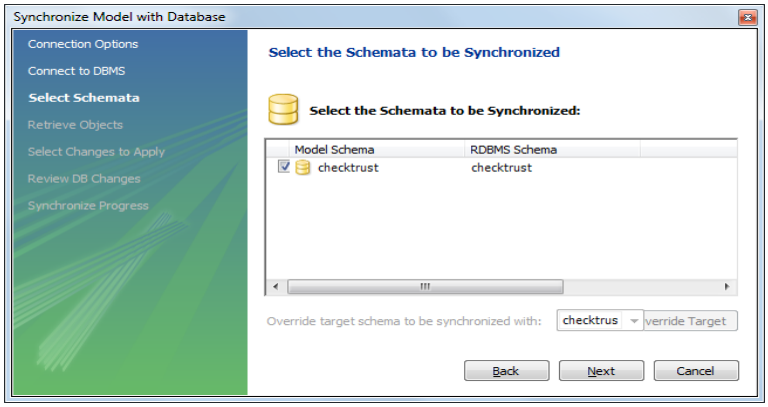
Mavjud ulanishlar ro‘yxatini dialog oynasida ***«Database → Manage Connections...»*** ni tanlash orqali ko‘rish mumkin.

Ma‘lumotlar tuzilishini sinxronlashtirish. MySQL Workbenchda ma‘lumotlar bazasi va mahalliy modelni sinxronlashtirish uchun maxsus vosita mavjud. Kerakli modelni ochib, «Database → Synchronize Model...» (Ctrl + Shift + G) tanlanadi, shundan so‘ng biz saqlangan masofaviy ulanishlardan birini tanlab, uning parametrlarini o‘zgartirishimiz mumkin. Ma‘lumotlar bazasiga ulanish uchun «Next» bosiladi:



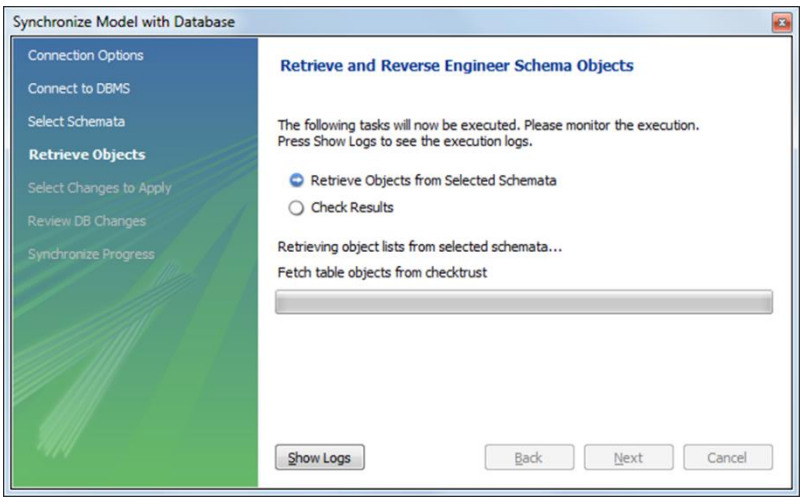
3.6-rasm. MySQL Workbenchda bog‘lanishni tekshirish

Serverga ulangandan va «Keyingi» tugmachasini bosgandan so‘ng biz sinxronizatsiya uchun mavjud bo‘lgan modellar ro‘yxatini (chap ustunda) va ma‘lumotlar bazasini (o‘ng ustunda) ko‘ramiz:



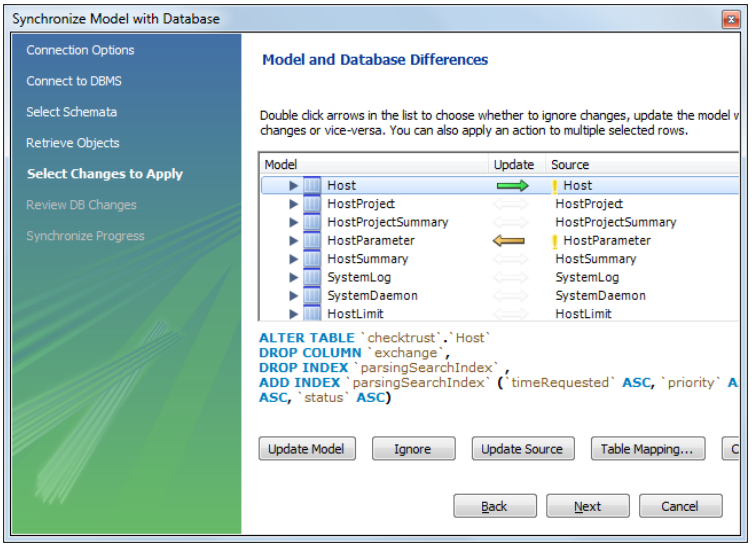
3.7-rasm. MySQL Workbenchda bazani tanlash

Kerakli ma‘lumotlar bazasi va sxemasini tanlab, *«Next»* ("Keyingi") ni bosing, masofaviy ma‘lumotlar bazasining tuzilishini va bizning modelimizni taqqoslash jarayonini boshlang:



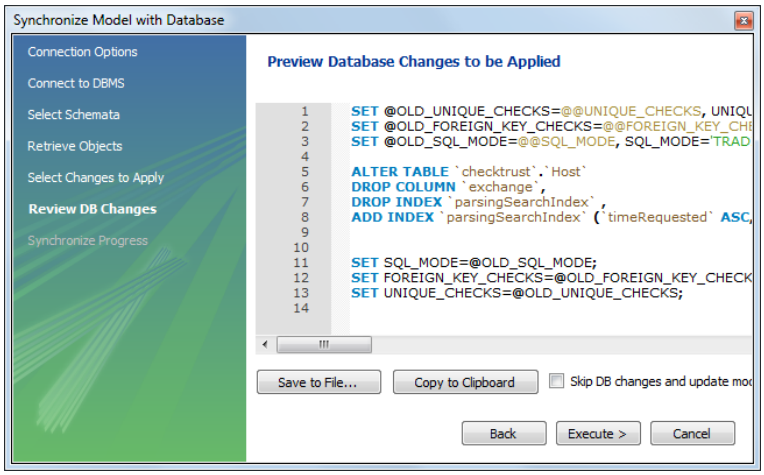
3.8-rasm. MySQL Workbench baza bilan bog‘lanishni tekshirish

Jarayonni tugatgandan so‘ng, biz ma‘lumotlar sxemasi va masofaviy ma‘lumotlar bazasi o‘rtasidagi farqlar ro‘yxatini ko‘rishimiz mumkin:



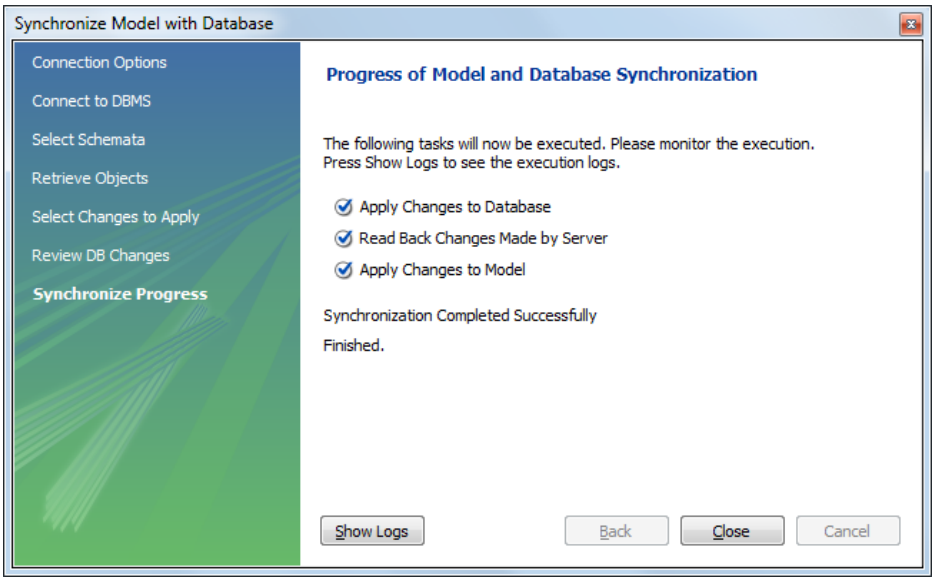
3.9-rasm. MySQL Workbench bazada yaratiladigan jadvallar

Bu yerda biz jadvalni birlashtirishni sozlashimiz mumkin: bizning o‘zgarishlarimizni serverga kiritish ("Update Source"), konfiguratsiyani serverdan mahalliy modelga yuklab olish ("Update Model") yoki farqlarni e‘tiborsiz qoldirish ("Ignore"). Bundan tashqari, butun ma‘lumotlar bazasi uchun va har bir jadval uchun alohida konfiguratsiya varianti sifatida mavjud. Jadvallardan birini va qo‘shilish usulini tanlashda biz sinxronizatsiya jarayonida bajariladigan SQL so‘rovlarini ko‘rishimiz mumkin, buni uchun ***"Next"*** tugmachasini bosib so‘rovlarning to‘liq to‘plamini ko‘rish mumkin:



3.10-rasm. SQL so‘rovlarini ko‘rinishi

SQL so‘rovlarini ko‘rib chiqqaningizdan so‘ng, ***"Execute>"*** tugmachasini bosiladi, sinxronizatsiyalashni bajarish boshlanadi. Agar hamma narsa yaxshi bo‘lsa, bizga quyidagi hisobot ko‘rinadi:



3.11-rasm. Sinxronizatsiya tugashi oynasi

Xatolar bo‘lsa, yuqoridagi bilan bir xil dialog oynasi hosil bo‘ladi.

Yuqorida tavsiflangan sinxronizatsiya faqat masofaviy ma‘lumotlar bazasining ma‘lumotlar tuzilishi va mahalliy modelning kombinatsiyasini ta‘minlaydi, ammo modelga kiritilgan boshlang‘ich ma‘lumotlarga ta‘sir qilmaydi ("Inersts"). Agar siz ularni yuklamoqchi bo‘lsangiz, " Database → Forward Engineer..." (Ctrl + G) ni tanlang, so‘ngra avval saqlangan ulanishlardan birini tanlang (yoki yangisini yarating) va *" Next "* ni bosing. Bundan tashqari, agar ma‘lumot sxemasini serverga oddiy holda yuklashda foydalanish mumkin.

4- laboratoriya ishi

SQL tilida cheklovlar turlari va o‘rnatish vositalaridan foydalanib masalalar yechish

Ishdan maqsad: ma‘lumotlar bazasini boshqarish dasturiy vositasi yoramida ma‘lumotlarni qayta ishlash, tahrirlash ko‘nikmalariga ega bo‘lish.

Masalani qo‘yilishi: ma‘lumotlar bazasini boshqarish vositalari yordamida ma‘lumotlarni qayta ishlash.

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

CREATE TABLE (создать таблици- jadval yaratish)

INSERT (вставка-qo‘yish)

SELECT (выбрать-ajratib olish)

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

Jadval yaratayotganingizda (yoki uni o‘zgartirayotganingizda), siz maydonlarga kiritilayotgan qiymatlarga cheklanishlar o‘rnatishingiz mumkin. Bu holda SQL cheklanishlarga to‘g‘ri kelmaydigan hamma qiymatlarni rad etadi. Cheklanishlar ikki asosiy turi mavjud: — ustun va jadval cheklanishlari. Ularning farqi shundaki ustun cheklanishi faqat ayrim ustunlarga qo‘llanadi, jadval cheklanishi bo‘lsa bir yoki bir necha ustunlar guruhiga qo‘llanadi. Ustun cheklanishi ustun nomi oxiriga ma‘lumotlar tipidan so‘ng va verguldan oldin qo‘yiladi. Jadval cheklanishi jadval nomi oxiriga so‘nggi dumaloq verguldan oldin qo‘yiladi. Cheklanishlar hisobga olingan

CREATE TABLE komandasi sintaksisi:

CREATE TABLE < table name >

(Ccolumn name> <data type> <column constraint,

Ccolumn name> <data type> <column constraint ...

<table constraint (<column name> [, Ccolumn name>])...)

Maydonga bo‘sh (NULL) qiymatlar kiritilishining 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 **employee** jadvalini quyidagicha yaratish mumkin:

CREATE TABLE Employee(
id integer NOT NULL,
name varchar (45),
birthday date,
seriya varchar(45),
address varchar(255),
phone varchar(15),
staji int,
status tinyint(1),
is_delete tinyint(1)
)

Ko‘p hollarda ustunga kiritilgan qiymatlar bir-biridan farq qilishi kerak. Agar ustun uchun **UNIQUE** cheklanishi o‘rnatilsa, bu ustunga mavjud qiymatni kiritishga urinish rad etiladi.

Bu cheklanish bo‘sh bo‘lmaydigan (NOT NULL) deb e‘lon qilingan maydonlarga qo‘llanishi mumkin.

Masalan:

CREATE TABLE Employee(
id integer NOT NULL UNIQUE,
name varchar (45),
birthday date,
seriya varchar(45),
address varchar(255),
phone varchar(15),
staji int,

status tinyint(1),

is_delete tinyint(1)

)

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 mijoz bitta hodimga biriktirilgan. Ya‘ni Mijozlar jadvalida mijoz nomeri (mid) va hodim nomeri (hid) kombinatsiyasi unikal bo‘lishi kerak. Bu cheklanishni ***UNIQUE*** (mid, hid) yordamida, Haridorlar jadvalini yaratishda kiritish mumkin. Bu ustunlar uchun ***NOT NULL*** cheklanishini kiritish zarurdir.

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

Misol:

CREATE TABLE Employee(

id integer NOT NULL PRIMARY KEY,

name varchar (45),

birthday date,

seriya varchar(45),

address varchar(255),

phone varchar(15),

staji int,

status tinyint(1),

is_delete tinyint(1)

)

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 Employee(
id integer NOT NULL,
name varchar (45),
birthday date,
seriya varchar(45),
address varchar(255),
phone varchar(15),
staji int,
status tinyint(1),
is_delete tinyint(1),
PRIMARY KEY (id, seriya)
)
```

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

Misol: ***Employee*** jadvali ***status*** ustuniga kiritilayotgan qiymat 1 dan kata yoki teng bo‘lish sharti.

```
CREATE TABLE Employee(
id integer NOT NULL PRIMARY KEY,
name varchar (45),
birthday date,
seriya varchar(45),
address varchar(255),
phone varchar(15),
staji int,
status tinyint(1) CHECK(status>=1),
is_delete tinyint(1)
)
```

CHECK cheklanishidan maydonga ma‘lum qiymatlarini kiritishdan himoya qilib, xatolar oldini olish uchun foydalanish mumkin. Masalan hodimlar adresi faqat Toshkent, Buxoro, Samarqand va Guliston shaharlar bo‘lsin.

```
CREATE TABLE Employee(
id integer NOT NULL PRIMARY KEY,
name varchar (45),
birthday date,
seriya varchar(45),
address varchar(255) CHECK (address
in("Toshkent","Buxoro","Samarqand","Guliston"),
phone varchar(15),
staji int,
status tinyint(1) CHECK(status>=1),
is_delete tinyint(1)
)
```

CHECK jadval cheklanishi sifatida kelishi mumkin. Bu shartga bir necha maydon kiritishga imkon beradi.

Masalan:

```
CREATE TABLE Employee(
id integer NOT NULL PRIMARY KEY,
name varchar (45),
birthday date,
seriya varchar(45),
address varchar(255),
phone varchar(15),
staji int,
status tinyint(1),
is_delete tinyint(1),
 CHECK (staji >=1 or address
in("Toshkent","Buxoro","Samarqand","Guliston")
```

)

Biror bir maydon uchun qiymat koʻrsatmagan holda jadvalga satr qoʻshsangiz, SQL bunday maydonga kiritish uchun koʻzda tutilgan qiymatga ega boʻlishi kerak, aks holda komanda rad etiladi. Eng umumiy koʻzda tutilgan qiymat ***NULL*** qiymatdir. ***CREATE TABLE*** komandasida koʻzda tutilgan qiymat ***DEFAULT*** operatori orqali, ustun cheklanishi sifatida koʻrsatiladi.

Masalan:

CREATE TABLE Employee(
id integer NOT NULL PRIMARY KEY,
name varchar (45),
birthday date,
seriya varchar(45),
address varchar(255),
phone varchar(15),
staji int,
status tinyint(1)) DEFAULT "NULL",
is_delete tinyint(1),
CHECK (staji >=1 or address in("Toshkent","Buxoro","Samarqand","Guliston")
)

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

Tashqi kalit bitta maydondan iborat boʻlishi shart emas. Birlamchi kalit kabi, tashqi kalit bitta modul sifatida qayta ishlanuvchi bir necha maydonlarga ega boʻlishi mumkin. Maydon tashqi kalit boʻlsa ilova qilayotgan jadval bilan maʻlum 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 holatda deyiladi.

Shu bilan birga ajdod kalit qiymati tashqi kalit bir necha qiymatlariga ilova qilishi mumkin.

Cheklanish **FOREIGN KEY**.

SQL ilovali yaxlitlikni **FOREIGN KEY** yordamida ta‘minlaydi. Tashqi kalit vazifasi ajdod kalitda ko‘rsatilmagan qiymatlarni tashqi kalit maydonlariga kiritmaslikdir. **FOREIGN KEY** 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 Hodimlar jadvaliga ilova qiluvchi tashqi kalit sifatida e‘lon qilingan **MNum** maydoniga ega bo‘lgan Mijozlarlar jadvalini yaratamiz:

```
CREATE TABLE Clients(
id integer NOT NULL PRIMARY KEY,
name varchar (45),
address varchar(255),
phone varchar(15),
MNum integer,
status tinyint(1) ) DEFAULT "NULL",
is_delete tinyint(1),
FOREIGN KEY (MNum) REFERENCES Employee (MNum)
)
```

Tashqi kalitni ustunlar cheklanishi sifatida berish mumkin. Buning uchun ***FOREIGN KEY*** ko‘rinishi — ko‘rsatkichli cheklanish (***REFERENCES***) qo‘llanadi:

FOREIGN KEY cheklanishidan jadval yoki ustun cheklanishi sifatida foydalanganda, agar ular ***PRIMARY KEY*** cheklanishiga ega bo‘lsa, ajdod kalit

ustunlarini koʻrsatmaslik mumkin.

Ilovali yaxlitlikni taʻminlash tashqi kalit yoki ajdod kalit maydonlari qiymatlariga cheklanishlar oʻrnatishni talab qiladi. Ajdod kalit tarkiblangan boʻlib, tashqi kalit har bir qiymati bitta satrga mos kelishi taʻminlangan boʻlishi kerak. Bu kalit unikal boʻlib, boʻsh (***NULL***) qiymatlarga ega boʻlmasligi kerak. Shuning uchun ajdod kalit maydonlari ***PRIMARY KEY*** cheklanishiga ega boʻlishi yoki ***NOT NULL*** cheklanishi bilan birga ***UNIQUE*** deb eʻlon qilinishi kerak.

Tashqi kalit ajdod kalitda mavjud qiymatlarga yoki boʻsh (***NULL***) qiymatga ega boʻlishi mumkin. Boshqa qiymat kiritishga urinish rad etiladi. Tashqi kalitga ***NOT NULL*** deb eʻlon qilish mumkin, lekin bu maqsadga muvofiq emas. Masalan, siz qaysi sotuvchi mos kelishini bilmasdan oldin buyurtmachini kiritmoqchisiz. Bu holda ***NULL*** qiymatdan foydalanib, keyinchalik uni konkret qiymatga almashtirish mumkin.

Tashqi kalit maydonlariga ***INSERT*** yoki ***UPDATE*** yordamida kiritilayotgan qiymatlar ajdod kalitlariga oldin kiritilgan boʻlishi kerak. Tashqi kalit ixtyoriy satrini ***DELETE*** yordamida oʻchirish mumkin. ***ANSI*** taʻrifi boʻyicha: tashqi kalit yordamida ilova qilinayotgan ajdod kalit qiymatini oʻchirib yoki oʻzgartirib boʻlmaydi. Bu shuni bildiradiki, buyurtmalar jadvalida buyurtmalarga ega buyurtmachini, buyurtmachilar jad-validan oʻchirib boʻlmaydi. ***ANSI*** tarkibiga kirmagan ajdod kalit maydonlarini oʻzgartirish yoki oʻchirish qoidalari rnavjud:

1)Cheklangan (***RESTRICT***) oʻzgartishlar. Siz (***ANSI*** usulida) ajdod kalitlarda cheklangan deb koʻrsatishingiz yoki man qilishingiz mumkin.

2)Kaskadlanuvchi (***CASCADE***) oʻzgartishlar. Agarda ajdod kalitda oʻgartish kiritsangiz, tashqi kalitda xuddi shunday oʻzgartishlar avtomatik yuz beradi.

3) Boʻsh (***NULL***) oʻzgartishlar. Siz ajdod kalitda oʻzgartirish kiritganingizda tashqi kalit maydonlari avtomatik ***NULL*** qiymat oladi (tashqi kalitda ***NULL*** qiymat ruxsat etilgan boʻlsa).

Yuqorida koʻrsatilgan efiektlar ***UPDATE*** va ***DELETE*** komandalari bajarilganda ajdod kalit oʻzgarishini koʻrsatadi va quyidagicha aniqlanadi:

CREATE TABLE <table-name >

(Ccolumn name> Cdata type>[(Csize>)],

Ccolumn name> Cdata type>[(Csize>)],

FOREGIGN KEY (Ccolumn name>,„) REFERENCES Ctable name>[(Ccolumn name>, ...)]

ON UPDATE [CASCADE|RESTRICT|SET NULL],

ON DELETE [CASCADE|RESTRICT|SET NULL], ...)

Misol. Siz sotuvchi nomerini o‘zgartirmoqchisiz, lekin uning hamma buyurtmachilarini saqlab qolmoqchisiz. Lekin bu sotuvchi firmadan bo‘shab ketsa siz uning buyurtmachilarini boshqa sotuvchiga mahkamlashingiz kerak. Buni bajarish uchun kaskad effektli ***UPDATE*** va cheklanishli ***DELETE*** berishingiz kerak.

CREATE TABLE Clients

(id integer NOT NULL PRIMARY KEY,

name varchar (45),

address varchar(255),

phone varchar(15),

status tinyint(1)) DEFAULT "NULL",

is_delete tinyint(1),

MNum integer, REFERENCES Employee

ON UPDATE CASCADE

ON DELETE RESTRICT

Agar endi sotuvchilar jadvalidan ***Peel*** ni o‘chirmoqchi bo‘lsangiz, buyurtmachilar jadvalida ***Hoffman*** va ***Clemens*** ning ***SNum*** maydonini boshqa tayinlangan sotuvchiga o‘zgartirishingiz kerak. Boshqa tomondan ***Peel SNum*** maydonini 1009 ga o‘zgartirsangiz ***Hoffman*** va ***Clemens*** ham avtomatik o‘zgaradi.

5- laboratoriya ishi

SQL tilining skalyar ifodalarining maxsus jihatlarining muhokamasi.

Ishdan maqsad: ma‘lumotlar bazasini boshqarish dasturiy vositasi yoramida masofaviy serverga ulanishni sozlash va serverdagi ma‘lumotlarni qayta ishlash, tahrirlash ko‘nikmalariga ega bo‘lish.

Masalani qo‘yilishi: ma‘lumotlar bazasini boshqarish vositalari yordamida masofaviy serverdagi ma‘lumotlarni qayta ishlash.

Uslubiy ko`rsatmalar: Hamma satrlar SQLda INSERT komandasi yordamida kiritiladi. INSERT quyidagi formatlardan biriga ega bo‘lishi mumkin:

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

VALUES (<value> [,<value>] ...);

yoki

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

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

INSERT INTO `employee` (`id`, `name`, `birthday`, `passport_series`, `pin_number`, `stir_number`, `adress`, `phone_number`,`staji`, `status`, `is_delete`)

VALUES (NULL, ‘Yaxshiyev Bekzod Obidjonovich‘, ‘1987-10-01‘, ‘AB0992022‘, ‘897456321‘, ‘5214785‘, ‘Toshkent shahri Yashanaobod tumani Temur Malik ko\‘chasi 32 - uy 45 - xonadon‘, ‘782547896‘, ‘3‘, ‘1‘, ‘1‘);

Ma‘lumotlar kiritish paytida aynan jadvalda berilgan ketmaketlikda bo‘lishi shart emas va bu nomlarni ixtiyoriy tartibda kiritishga imkon beradi. Masalan:

INSERT INTO `employee` (`id`, `birthday`,`name`, `passport_series`, `adress`, `phone_number`,`staji`,`pin_number`, `stir_number`, `status`, `is_delete`)

VALUES (NULL, ‘1987-10-01‘, ‘Yaxshiyev Bekzod Obidjonovich‘, ‘AB0992022‘, ‘NuLL, ‘782547896‘, ‘3‘, ‘897456321‘, ‘5214785‘, ‘NULL‘, ‘1‘);

E‘tibor bering, **adress** va **status** 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 Toshkentstaff

*SELECT * FROM employee*

WHERE adress = "Toshkent"

Satrlarni jadvaldan **DELETE** komandasi bilan o‘chirish mumkin. U alohida qiymatlarni emas, faqat satrlarni o‘chiradi. **DELETE** quyidagi formatga ega:

DELETE FROM <table name | view name>

[WHERE search-condition]

Masalan, Hodimlar jadvalidagi hamma satrlarni o‘chirish uchun quyidagi shartni kiritish mumkin:

DELETE FROM employee

Ma‘lum satrlarni o‘chirish uchun predikatdan foydalaniladi. Masalan, jadvaldan alohida hodimni o‘chirish uchun:

DELETE FROM employee

WHERE id = 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 yerda expression — bu ustun | ifoda | konstanta | o‘zga- ruvchi.

Ikkinchi variant:

UPDATE <table name>

SET column = expression, ...

[FROM table-list]

[WHERE search-condition]

Masalan, hamma buyurtmachilar bahosini 200 ga o‘zgartirish uchun quyidagini kiritishingiz mumkin:

UPDATE country

SET name = "Iralndiya"

Ma‘lum satrlarni o‘zgartirish uchun **DELETE** dagi kabi predikatdan foydalanish kerak. Masalan ma‘lum bir hodim (id=l5) ning hamma mijozlari uchun bir xil o‘zgartirishni quyidagicha kiritish mumkin:

UPDATE hodim

SET summa = 20000

WHERE id = 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 region

SET name = "Andijon", district = "Marhamat", code = .60 WHERE id = 4

Siz **UPDATE** komandasining **SET** jumlasida skalyar ifodalardan o‘zgartirilayotgan maydon ifodasiga qo‘shgan holda foydalanishingiz mumkin. Masalan:

UPDATE emloyee

*SET staji = staji * 2*

6- laboratoriya ishi

Triggerlarni yozilishi bo‘yicha misollar yechish.

Ishdan maqsad: ma‘lumotlar bazasida dasturlarda triggerlarni ishlab chiqish va ulardan foydalanish bo‘yicha amaliy ko‘nikmalarga ega bo‘lish..

Masalani qo‘yilishi: ma‘lumotlar bazasini boshqarish vositalari yordamida masofaviy serverdagi ma‘lumotlarni qayta ishlash.

Uslubiy ko`rsatmalar:

Trigger - bu saqlanadigan protseduralarning maxsus turi bo‘lib, server tomonidan avtomatik ravishda jadvallar yoki berilganlarni o‘zgarishiga javoban

ishga tushiriladi. Har bir trigger maʻlum bir jadval yoki berilganga bogʻlangan boʻladi.

Triggerlar maʻlumotlar yaxlitligini taʻminlash uchun ishlatiladi va murakkab biznes mantigʻini amalga oshirish, shuningdek, maʻlumotlarni oʻzgartirish haqida nazoratni yaratish.

Trigger tomonidan amalga oshirilgan barcha maʻlumotlar oʻzgarishi bitta operatsiya sifatida koʻrib chiqiladi. Xatolar aniqlanganda yoki maʻlumotlar yaxlitligi buzilgan taqdirda ushbu operatsiya orqaga qaytariladi. Shunday qilib, oʻzgartirish taqiqlanadi va Trigger tomonidan kiritilgan barcha oʻzgarishlar ham bekor qilinadi.

Trigger turlari

Triggerlar maʻlumotni aniqlash tili (**DataDefinitionLanguage** - DDL), maʻlumot manipulyatsiyasi tili (**DataManipulationLanguage** - DML) va kirish triggerlariga boʻlinadi. Oxirgi turdagi trigger SQL Serverda roʻyxatdan oʻtishda ishga tushirildi.

DML triggerlari mantiqiy (kontseptual) DELETED va INSETED jadvallaridan foydalanadilar. Foydalanuvchining bajargan karakatlarini triggerlar jadvallarga qayt qilib boradi. DELETED va INSETED jadvallari qaysidir foydalanuvchi tomonidan oʻzgartirilgan eski yoki yangi qator qiymatlarini oʻz ichiga oladi.

Jadvaldagi maʻlumotlar oʻzgarishi turiga koʻra toʻrtta triggerlar mavjud:

- ✓ INSERT TRIGGER - maʻlumotlarni kiritish paytida INSERT buyrugʻi yordamida tekshirishlar ishga tushiriladi;
- ✓ UPDATE TRIGGER - UPDATE triggerlar buyrugʻi yordamida maʻlumotlarni oʻzgartirishga urinishda ishga tushiriladi;
- ✓ DELETE TRIGGER - DELETE buyrugʻi yordamida maʻlumotlarni oʻchirishga harakat qilganingizda ushbu turdagi triggerlar ishga tushiriladi
- ✓ Bir vaqtning oʻzida sodir boʻlishni va tasodifan mos kelishini hisobga olgan holda yaratilgan triggerlar

Bundan tashqari qoʻllanishiga koʻra triggerlar kalssifikatsiyasi FOR (AFTER) va INSTEAD OF triggerlarga boʻlinadi.

Jadvaldagi maʻlumotlarni oʻzgartiruvchi buyruq bajarilgandan soʻng FOR (AFTER) triggeri bajariladi. Agar biron bir sababga koʻra buyruq muvaffaqiyatli bajarilmasa, trigger ham bajarilmaydi. Foydalanuvchi soʻrovi natijasida olingan maʻlumotlar oʻzgarishi va triggerning bajarilishi bitta tranzaktsiyaning tanasida amalga oshiriladi. Agar trigger ishlaqmasa foydalanuvchi oʻzgarishlari ham orqaga qaytadi.

SQL Server 2008 R2 standartida barcha triggerlar FOR (AFTER) turidagi triggerlardir.

Berilish uchun FOR (AFTER) turdagi triggerlarni aniqlab boʻlmaydi. Har bir jadval va har bir operatsiya uchun bir nechta AFTER triggerlarini belgilashingiz mumkin (INSERT, UPDATE, DELETE).

INSTEAD OF trigger ularni oʻchirishga olib kelgan harakatlarni chetlab oʻtish uchun amalga oshiriladi, bu harakatlar oʻrniga. Masalan, INSTEAD OF triggeri bor jadvalni yangilash ushbu tekshirish ishga tushiradi. Natijada, yangilash bayonnomasi oʻrniga tetik kodi bajariladi.

Jadvallar va berilganlar uchun INSTEAD OF triggerlarni aniqlash mumkin. Har bir operatsiya uchun siz faqat bitta INSTEAD triggerini aniqlay olasiz (INSERT, UPDATE, DELETE).

INSTEAD OF triggeri qayta tiklanish uchun WITH CHECK OPTION parametrlaridan foydalanishga ruxsat beryaydi. SQL Serverda xatolik sodir boʻladi agarda INSTEAD OF triggeri qayta tiklanuvchi WITH CHECK OPTION parametric bilan qoʻshilsa. Foydalanuvchi ALTER VIEW yordamida INSEAD OF ni qoʻshishdan oldin ushbu parametrni olib tashlashi kerak.

Vaqtinchalik yoki tizim jadvallarida triggerlarni oʻrnatolmaysiz, bunday jadvallarga T-SQL triggerning oʻzi koʻrsatmalarida murojaat etilishi mumkin.

Jadvalda INSTEAD OF DELETE va INSTEAD OF UPDATE triggerlarini aniqlay olmaysiz, maʻlumotlarning yaxlitlik cheklovlari mos ravishda ON DELETE yoki ON UPDATE ga oʻrnatiladi.

Ikkala sinf triggerlarini jadvalga bogʻlab qoʻyish mumkin: INSTEAD OF va FOR (AFTER). Agar ikkala sinfning cheklashlari va triggerlari jadvalda aniqlangan

bo‘lsa, ularning birinchisi - INSTEAD OF trigger cheklovlar qayta ishlanadi va oxirgi trigger - FOR (AFTER) bajariladi. Agar cheklashlar buzilgan bo‘lsa, INSTEAD OF triggerning harakatlari qaytariladi. Agar cheklovlar buzilsa yoki jadvalni o‘zgartirishga imkon bermaydigan boshqa hodisalar ro‘y bersa, FOR (AFTER) triggeri bajarilmaydi.

Saqlangan protseduralarga asoslangan holda triggerlarning joylashishga qarab 32-pag‘onadagi triggerlarning rekursiv operatsiya ham ishlashi mumkin. **Sp_settriggerorder** tizimida saqlangan protseduradan foydalanib, qaysi trigger birinchi navbatda bajarilishini va qaysi oxirini belgilash mumkin.

Trigger yaratish operato‘rining sintaksisi.

CREATETRIGGER<trigger_nomi>

ON[<*sxemanomi*>]{ <*jadvallar nomi*> | <*berilganlarnomi*> }

[WITH ENCRYPTION | EXECUTE AS {<CALLER | SELF | <USER>>}]

{ FOR | AFTER | INSTEAD OF }

{ [DELETE] [,] [INSERT] [,] [UPDATE] }

[WITH APPEND]

[NOT FOR REPLICATION]

AS

{*sql_statement*| EXTERNAL NAME <*method specifier*> }

Sxema nomi – DML triggerga tegishli bo‘lgan sxema nomi. DML triggerlari ular uchun yaratilgan jadval sxemasi maydoni yoki berilganlar bilan chegaralangan. <*Sxema nomi*> argumentini DDL yoki kiritish triggerlari uchun aniqlab bo‘lmaydi.

Trigger nomi – Trigger identifikatori identifikatorlar uchun qoidalarga amal qilishi kerak, *trigger_name* # yoki ## harflari bilan boshlanishi mumkin emas.

Jadvallar/Berilganlar – jadvallar yoki berilganlar DML trigger yordamida bajariladi.

WITH ENCRYPTION – CREATETRIGGER ko‘rsatmalari matnini shifrlaydi va unga kirish imkoni bo‘lmaydi (jumladan administratorga ham). SQL Server replikatsiyasi bir qismida WITH ENCRYPTION argumentidan foydalanishda

triggerlarni e‘lon qilishga ruxsat bermaydi. CLR triggerlari uchun WITH ENCRYPTION parametrni aniqlab bo‘lmaydi.

EXECUTE AS – bu trigger bajarilishi uchun xavfsizlik kontekstini belgilaydi. Bu trigger tomonidan foydalanuvchini boshqarish, SQL Serverda har qanday ma‘lumotlar bazasi obyektlarida ruxsatlarni tekshirish uchun foydalaniladi.

FOR | AFTER – AFTER tipi shuni ko‘rsatadiki DML triggeri SQL qaydnomalarini operatsiyalari muvaffaqiyatli tugallangandan keying ishga tushadi. Trigger bajarilishi uchun berilgan barcha kaskadli harakatlar va cheklanishlash muvaffaqiyatli yakunlanishi kerak.

INSTEAD OF – shuni bildiradiki DML triggerda SQL qaydnomalari o‘rnida trigger ishlatiladi va shu tarzda trigger bajarilgan buyruqlarni aniqlaydi. DDL triggerlari yoki kiritish triggerlari uchun INSTEAD OF argumentini ko‘rsatish mumkin emas.

{[DELETE] [,] [INSERT] [,] [UPDATE]} – ma‘lumotlarni o‘zgartirish uchun ko‘rsatmalarni belgilaydi, natijada DML ishga tushiradi agar u jadval yoki ko‘rinishga tegishli bo‘lsa. Kamida bitta qaydnoma ko‘rsatilishi kerak. Triggerlar har qanday kombinatsiya va tartibga ruxsat etadi.

INSTEAD OF triggerning DELETE parametri kaskadli ON DELETE harakati bilan bog‘langan jadvallarda ruxsat etilmaydi. Huddi shunday, UPDATE parametrini ON UPDATE kaskadli harakati bilan bog‘laydigan jadvallarda qo‘llab bo‘lmaydi.

WITH APPEND – Oldindan bor bo‘lgan toifadagi triggerni qo‘shmoqchi ekanligingizni bildiradi. WITH APPEND argumentidan INSTEAD OF va AFTER triggerlari aniq ko‘rsatilgan vaqtda foydalanib bo‘lmaydi. WITH APPEND argumentida FOR parametridan faqat INSEAD OF ni ishlatmasdan yoki bazi sabablarag ko‘ra teskari aloqani qo‘llab – quvatlash uchun ALTER foydalanish mumkin. Bunda EXTERNAL NAME parametrlari ishlatilgan bo‘lsa WITH APPEND argumenti ishlatilmaydi.

NOT FOR REPLICATION – ushbu parametr bilan triggerni yaratishda jadvallarni o‘zgartirayotganda uni ishga tushirish taqiqlanadi mexanizmlari

repleksiyasi bajariladi. Koʻpincha replikatsiya quyi tizimini qoʻllab-quvvatlash uchun tizim triggerlarini yaratishda ishlatiladi.

sql_statement – trigger ishga tushganda bajariladigan buyruqlar toʻplami (trigger tanasi).

Triggerlarga misollar.

Shunday qilib triggerlarni oʻrganishni boshlaymiz, ikkita jadval yaratib olamiz. Birinchi jadvalda biz umumiy foydalanuvchilar haqidagi maʻlumotlarni saqlaymiz va ikkinchida esa tizimga kirgan vaqtni va id ni saqlaymiz:

```
CREATE TABLE users(
 id INTEGER PRIMARY KEY,
 name TEXT NOT NULL,
 age INTEGER NOT NULL,
 address TEXT NOT NULL,
 mydate TEXT NOT NULL
);

CREATE TABLE user_log (
id_u INTEGER NOT NULL,
u_date TEXT NOT NULL
 );
```

SQL soʻrov bajarilishidan oldin ishga tushadigan triggerni yozaylik:

```
CREATE TRIGGER my_u_log BEFORE INSERT
 ON users
 BEGIN
 INSERT INTO user_log(id_u, u_date) VALUES (NEW.id,
datetime('now'));
 END;
```

Ushbu jadval foydalanuvchilar jadvaliga yangi qator qoʻshilishdan oldin ishga tushadi. Buni koʻrib chiqaylik:

INSERT INTO users(name, age, address, mydate)

VALUES (Sanjarbek, 28, ‘Bagat‘, ***datetime***(‘now‘));

SELECT **** FROM*** users;

id	name	age	***address***	mydate
1	Sanjarbek	28	Bagat	2019-10-08 11:35:01

SELECT **** FROM*** user_log;

id_u	u_date
-1	2019-10-08 11:35:01

Ko‘rishimiz mumkinki ***user_log*** jadvaliga ma‘lumotlar avtomatik ravishda qo‘shilganini ko‘rishimiz mumkin. Shuni ko‘rishimiz mumkinki ***user_log*** jadvalidagi ***u_date*** ustuniga ***users*** jadvalidagi ***mydate*** ustunidagi ma‘lumotlar yozildi. Lekin ***id_u*** ustinida -1 turganini ko‘rishimiz mumkin, bunga sabab SQLite3 aniqlay olmaydi ***users*** jadvalidagi ***id*** ustunina qanday qiymat turganini.

SQL **AFTER** trigger: so‘rovlar bajarilgandan keyin ishga tushiriladigan trigger.

Oldin yozgan so‘ravimizni o‘zgartiramiz jadval avvalgidek qoladi, ammo triggerning bir qismini o‘zgartiramiz BEFORE ni o‘rniga AFTER va so‘rovni bajargandan keyin nima hodisa yuz beradi.

```
1  CREATE TRIGGER my_u_log AFTER INSERT
2
3  ON users
4
5  BEGIN
6
7  INSERT INTO user_log(id_u, u_date) VALUES (NEW.id, datetime('now'));
8
9  END;
```

Triggerni yaratib oldik endi users jadvaliga ma‘lumotlar kiritamiz va user_log jadvaliga ma‘lumotlar avtomatik tarzda to‘ldiriladi.

INSERT INTO users(name, age, address, mydate)
VALUES (Sanjarbek, 28, ‘Bagat‘, ***datetime***(‘now‘));

INSERT INTO users(name, age, address, mydate)
VALUES (Safarmurod, 25, Tashkent, ***datetime***(‘now‘));

SELECT **** FROM*** users;

id	name	age	***address***	mydate
1	Sanjarbek	28	Bagat	2019-10-08 11:35:01
1	Safarmurod	25	Tashkent	2019-10-08 11:35:01

SELECT **** FROM*** user_log;

id_u	u_date
1	2019-10-08 11:35:01
2	2019-10-08 11:35:01

Endi ko‘rishimiz mumkinki ikkinchi jadvalga identifikator to‘g‘ri yozilgan. NEW modifikatoriga e‘tibor beraylik.

NEW modifakotori ma‘lumotlar bazasini boshqarish tizimlarida yangi qiymatlarni kiritishni belgilaydigan trigger tanasida ishlatiladigan kalit so‘zdir. Umid qilamanki BEFORE va AFTER faqrlarini tushundingiz.

Nazorat savollari

1. Benuqsonlikni cheklash nimani anglatadi?
2. Butunlik uchun cheklash turlarini sanab bering.
3. Triggerlar bilan qanday yaxlitlikni cheklash mumkin?
4. DDL triggerlar ma‘lumotlar bazasida qanday o‘zgarishlar faollashadi?
5. Saqlangan protsedurada trigger tomonidan bajariladigan amallarni kodlash mumkinmi?

6. Triggerlar va saqlanadigan protseduralar o‘rtasidagi farqlar qanday?

7. CREATE TRIGGER iborasining sintaksisiga izoh bering.

8. INSERTED va DELETED jadvallarining maqsadi nima?

9. Qaysi tizim protsedurasi trigger kodini olishga imkon beradi?

10. Ma‘lumotlar bazasi obyekti ishga tushirilishini qayerdan bilaman?

11. Muayyan jadval uchun triggerlar ro‘yxatini qanday olish mumkin?

7- laboratoriya ishi

Jadvallarni qo‘shish. Murakkab so‘rovlar vositalaridan foydalangan holda masalalar yechish.

Ishdan maqsad: ma‘lumotlar bazasini boshqarish dasturiy vositasi yoramida masofaviy serverga ulanishni sozlash va serverdagi ma‘lumotlarni qayta ishlash, tahrirlash ko‘nikmalariga ega bo‘lish.

Masalani qo‘yilishi: ma‘lumotlar bazasini boshqarish vositalari yordamida masofaviy serverdagi ma‘lumotlarni qayta ishlash.

Uslubiy ko`rsatmalar: Jadvallarni jamlashtirish. Jamlashtirish relyatsion ma‘lumotlar bazasi operatsiyalaridan biri bo‘lib, jadvallar orasidagi aloqani belgilaydi va ulardan ma‘lumotni bitta komanda yordamida ajratishga imkon beradi. Har xil jadvallarda bir xil nomli ustunlar bo‘lishi mumkin boigani uchun, kerakli ustun uchun jadval nomi prefiksi ishlatiladi. Jamlashda jadvallar FROM ifodasidan so‘ng ro‘yxat sifatida tasvirlanadi. So‘rov predikati ixtiyoriy jadval ixtiyoriy ustuniga tegishli bo‘lishi mumkin. Jamlashning eng soddasi bu dekart ko‘paytmasi bo‘lib, uni quyidagicha bajarish mumkin:

SELECT Customers.*, Salepeople.* FROM Salepeople, Customers*

Lekin bu yerda hosil bo‘lgan jadval keraksiz ma‘lumotlarga ega. Keraksiz satrlarni olib tashlash uchun **WHERE** jumlasidan foydalaniladi.

Masalan: berilgan shahardagi sotuvchilar va buyurtmachilar ixtiyoriy kombinatsiyasini ko‘rish uchun quyidagini kiritish lozim:

SELECT Customers.CName, Salepeople.SName, Salepeople.City FROM Salepeople, Customers WHERE Salepeople.City = Customers.City

Jamlashda **SQL** bir necha jadval satrlari kombinatsiyasini predikatlar bo‘yicha solishtiradi. Asosan ma‘lumotlar ilovali yaxlitlik asosida tekshirilib, ajratib olinadi.

Misol: har bir sotuvchiga mos keluvchi buyurtmachilar ro‘yxati:

SELECT Customers.CName, Salepeople.SName FROM Customers, Salepeople WHERE Salepeople. SNum = Customers.SNum

Tenglikka asoslangan predikatlardan foydalanuvchi jamlanmalar, tenglik bo‘yicha jamlanma deb atalib, jamlanmalarning eng umumiy ko‘rinishidir. Shu bilan birga ixtiyoriy relyatsion operatordan foydalanish mumkin.

Sodda joylashtirilgan ostki so‘rovlar.

SQL yordamida so‘rovlami bir-birining ichiga joylashtirishingiz mumkin. Odatda ichki so‘rov qiymat hosil qiladi va bu qiymat tashqi predikat tomonidan tekshirilib, to‘g‘ri yoki noto‘g‘riligi tekshiriladi.

Misol: bizga sotuvchi nomi maium: Motika, lekin biz SNum maydoni qiymatini bilmaymiz va Buyurtmachilar jadvalidan hamma buyurtmalarni ajratib olmoqchimiz. Buni quyidagicha amalga oshirish mumkin:

SELECT * FROM Orders WHERE SNum = (SELECT SNum FROM Salepeople WHERE SName = “Motika”)

Awal ichki so‘rov bajariladi, so‘ngra uning natijasi tashqi so‘rovni hosil qilish uchun ishlatiladi (***SNum*** ostki so‘rov nati jasi bilan solishtiriladi).

Ostki so‘rov bitta ustun tanlashi lozim, bu ustun qiymatlari tipi predikatda solishtiriladigan qiymat tipi bilan bir xil boiishi kerak. Siz ba‘zi hollarda ostki so‘rov bitta qiymat hosil qilishi uchun ***DISTINCT*** operatoridan foydalanishingiz mumkin.

Misol: Hoffman ***(CNum=21)*** ga xizmat ko‘rsatuvchi sotuvchilar hamma buyurtmalarini topish lozim boisin:

SELECT * FROM Orders WHERE SNum = (SELECT DISTINCT SNum FROM Orders WHERE CNum = 21)

Bu holda ostki so‘rov faqat bitta qiymat chiqaradi, lekin umumiy holda bir necha qiymatlar bo‘lishi mumkin va ular ichidan ***DISTINCT*** faqat bittasini tanlaydi. Ixtiyoriy sondagi satrlar uchun avtomatik ravishda bitta qiymat hosil qiluvchi funksiya turi — agregat funksiya boiib, undan ostki so‘rovda foydalanish mumkin.

Masalan, siz summasi 4 oktabrda bajarilishi lozim boigan buyurtmalar summasi o‘rta qiymatidan yuqori boigan hamma buyurtmalarni ko‘rmoqchisiz:

SELECT * FROM Orders WHERE AMT > (SELECT AVG (AMT) FROM Orders WHERE ODate = “1990/10/04”)

Shuni nazarda tutish kerakki, guruhlangan agregat funktsiyalar ***GROUP BY*** ifodasi terminlarida aniqlangan agregat fimksiyalar boisa, ko‘p qiymatlar hosil qilishi mumkin.

Agar ostki so‘rov ***IN*** operatoridan foydalanilsa, ixtiyoriy sondagi satrlar hosil qilish mumkin.

Misol: Londondagi sotuvchilar uchun hamma buyurtmalami ko‘rsatish:

SELECT * FROM Orders WHERE SNum IN (SELECT SNum FROM Salepeople WHERE City = “London”)

Bu natijani jamlanma orqali hosil qilish mumkin. Lekin odatda ostki soiovli so‘rovlar tezroq bajariladi. Siz ostki so‘rov

SELECT jumlasida ustunga asoslangan ifodadan foydalanishingiz mumkin. Bu relyatsion operatorlar yordamida yoki ***IN*** yordamida amalga oshirilishi mumkin. Siz ostki so‘rovlami ***HAVING*** ichida ishlatishingiz mumkin. Bu ostki so‘rovlar agar ko‘p qiymatlar qaytarmasa xususiy agregat funksiyalaridan yoki GROUP BY yoki ***HAVING*** operatorlaridan foydalanishi mumkin.

Misol:

SELECT Rating, COUNT (DISTINCT CNum) FROM Customers GROUP BY Rating HAVING Rating > (SELECT AVG (Rating) FROM Customers WHERE City = “San Hose”)

Bu komanda San Hose dagi baholari o‘rtachadan yuqori boigan buyurtmachilarni aniqlaydi.

UNION ifodasidan foydalanish.

UNION ifodasi bir yoki bir necha ***SQL*** so‘rovlar natijasini birlashtirishga imkon beradi.

Misol: Londonda joylashgan hamma sotuvchilar va buyurt- machilarni bitta jadvalda chiqaring:

SELECT SNum, SName FROM Salepeople WHERE City = “London” UNION SELECT CNum, CName FROM Customers WHERE City = “London”

Nazorat savollari

1. SQL nima va uning yangi standarti qachon qabul qilingan ?
2. Qaysi buyruq yordamida jadvaldan ma‘lumotlar qidiriladi ?
3. Qaysi buyruqlar yordamida jadvaldagi zarur ma‘lumotlar olinadi?

8- laboratoriya ishi

SQL tilida foydalanuvchi tomonidan aniqlanadigan ma‘lumot turlarining muxim jixatlarining muhokamasi. Turlarga ajratilgan jadvallarni aniqlash

Ishdan maqsad: ma‘lumotlar bazasini boshqarish dasturiy vositasi yoramida masofaviy serverga ulanishni sozlash va serverdagi ma‘lumotlarni qayta ishlash, tahrirlash ko‘nikmalariga ega bo‘lish.

Masalani qo‘yilishi: ma‘lumotlar bazasini boshqarish vositalari yordamida masofaviy serverdagi ma‘lumotlarni qayta ishlash.

Uslubiy ko`rsatmalar: 2012 yilning 1-aprelidan Microsoft SQL Server 2012 ma‘lumotlar bazasini yakuniy «versiya»si chiqarildi. Yangi «versiya»da ko‘pgina yangiliklar mavjud. Ushbu maqolada shu ma‘lumotlar bazasi bilan ishlovchi dasturchilar uchun yangi kiritilgan imkoniyatlar haqida to‘xtalib o‘tiladi.

1. Bazadan olinadigan qatorlarni sonini kamaytirish uchun ishlatiladigan **OFFSET** va **FETCH** deb nomlangan yangi buyruqlar. Endilikda SQL Server da qulay «betlash» vujudga keldi va natijani ikkinchi o‘nta yozuvni (2chi betini) olish uchun so‘rov quyidagi ko‘rinishda bo‘ladi:

SELECT DepartmentID, Name, GroupName

FROM HumanResources.Department

ORDER BY DepartmentID

OFFSET 10 ROWS

FETCH NEXT 10 ROWS ONLY;

SQL Server 2005/2008 da esa xuddi shu so‘rov quyidagicha yozilar edi:

WITH CTE AS(SELECT DepartmentID, Name, GroupName, ROW_NUMBER()

OVER (ORDER BY DepartmentID) AS RowNum

FROM HumanResources.Department)

SELECT DepartmentID, Name, GroupName

FROM CTE

WHERE RowNum BETWEEN 10 AND 19

ORDER BY RowNum

2. **OVER** operatorida oynali funksialarn bilan ishlash yanada takomillashdi. **ROWS** va **RANGE** ni ishlatib bo‘lim ichidagi yozuvlarni avvalgi va oxirgi nuqtalarini belgilab, yana ham qisqartirish mumkin. Masalan, o‘sib boruvchi natijani oladigan so‘rov quyidagicha bo‘ladi:

SELECT DepartmentID,

SaleDate,

SUM(SalesYTD) OVER (PARTITION BY DepartmentID ORDER BY SaleDate ROWS UNBOUNDED PRECEDING) AS Total

FROM dbo.Sales

3. Yangi tahliliy funksiyalar qo‘shildi.

- FIRST_VALUE
- LAST_VALUE
- CUME_DIST
- PERCENTILE_DISC
- PERCENT_RANK
- PERCENTILE_CONT
- LEAD
- LAG

Masalan, biron bir loyixa uchun 4 xil narxni hisoblab chiqishim kerak (open, high, low, close). Bunday so‘rovni juda oson yozish mumkin:

SELECT MIN(Ask) OVER (PARTITION BY Pair, Candle) AS Low,

MAX(Ask) OVER (PARTITION BY Pair, Candle) AS High,

FIRST_VALUE(Ask) OVER (PARTITION BY Pair, Candle) AS Open,
LAST_VALUE(Ask) OVER (PARTITION BY Pair, Candle) AS Close
FROM dbo.Quotes
WHERE Pair='EURUSD' and Candle = 100

4. 14 ta yangi funksiyalar paydo bo'ldi

Keltirish funksiyalari:

PARSE

TRY_CONVERT

TRY_PARSE

Sana va vaqt funksiyalari:

DATEFROMPARTS

DATETIME2FROMPARTS

DATETIMEFROMPARTS

DATETIMEOFFSETFROMPARTS

EOMONTH

SMALLDATETIMEFROMPARTS

TIMEFROMPARTS

Mantiqiy funksiyalar:

CHOOSE

IIF

Qator bilan ishlovchi funksiyalar:

CONCAT

FORMAT

Misol uchun, oy kunini aniqlash uchun avval so'rov mana bunday yozilar edi:

SET @LastDayOfMonth = dateadd(month,1,dateadd(day,1-day(@d),@d))-1

SQL Server 2012 da esa:

SET @LastDayOfMonth = EOMONTH (@d)

Yoki masalan mana bu so'rovni o'rniga:

CASE WHEN (@a > @b) THEN ‘TRUE‘ ELSE ‘FALSE‘ END

quyidagicha yozish mumkin:

IIF (@a > @b, ‘TRUE‘, ‘FALSE‘)

5. FileTable jadvallari

Endi SQL Server fayllar va hujjatlarni maxsus jadvallarda saqlaydigan boʻldi. Bunday jadvallar bilan Windows dasturlari boshqa fayllar bilan qanday ishlasa xuddi shunday ishlay oladi. Bunada OT ga hech qanday oʻzgartirish kiritlmaydi.

Batafsil: http://msdn.microsoft.com/ru-ru/library/ff929144.aspx

Agregat (yoki STATIK) funksiyalar, sonli yoki hisoblanuvchi ustunlar bilan ishlaydi. Agregat funksiya argumenti butun ustun boʻlib, bitta qiymat qaytaradi. Bu funksiyalami koʻrib chiqamiz:

SUM() — Ustundagi hamma qiymatlar summasini hisoblaydi.

AVG() — Ustundagi hamma qiymatlar oʻrtacha qiymating hisoblaydi.

MIN() — ustundagi hamma qiymatlar eng kichigini aniqlaydi.

MAX() — ustundagi hamma qiymatlar eng kattasini aniqlaydi.

COUNT() — ustundagi qiymatlar sonini hisoblaydi.

COUNT(*) — soʻrov natijalari jadvalidagi satrlar sonini hisoblaydi.

Agregatlash argumenti boʻlib ustun nomidan tashqari ixtiyoriy matematik ifoda xizmat qilishi mumkin. Misol uchun quyidagi soʻrovda: Sizni kompaniyangizda reja bajarilishi oʻrtacha foizi qancha?

SELECT AVG(100 * (SALES/QUOTA)) FROM SALESREPS

Yana bir shakl: Sizni kompaniyangizda reja bajarilishi oʻrtacha foizi qancha?

SELECT AVG(100 * (SALES/QUOTA)) PROCENT ROM SALESREPS

Bu holda ustun nomi maʻnoliroq, lekin bu asosiysi emas. Ustunlar summasini hisoblab koʻramiz. SUM() funksiyasini qoilaymiz, ustun sonli boiishi kerak. Masalan, quyidagicha: Kompaniya xizmatchilari sotuvlar hajmi rejadagi va haqiqiy oʻrta qiymati qanchaga teng?

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

AVG() agregatlash funksiyasiga, yana bir necha sodda misollarni koʻramiz. Masalan: “ACI” ishlab chiqaruvchi mollari oʻrtacha narxini hisoblang.

SELECT AVG(PMCE) FROM PRODUCTS WHERE MFR_ID = “ACI”

Ekstremumlarni topish funksiyalari, yani MIN(), MAX() funksiyalarini koʻramiz. Bu funksiyalar sonli ustunlar, sanalar va satrli oʻzgaruvchilar bilan ishlaydi. Eng sodda qoʻlanishli sonlar bilan ishlash:

Masalan, quyidagicha soʻrov beramiz: Eng koʻp va kam sotuvlar rejadagi hajmi?

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

Bu sonlami oʻz ichiga olgan ustunlardir. Yana bir soʻrov beramiz: Bazadagi buyurtmalarning ichida eng oldin berilgan soiov 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() qoilanadi. Bu funksiya son qiymat qaytaradi.

Masalan: Kompaniyamiz mijozlari soni nechta?

SELECT COUNT(CUST_NUM) FROM CUSTOMERS

COUNT(*) funksiyasi qiymatlar sonini emas, satrlar sonini hisoblaydi. Quyidagicha yozish mumkin:

SELECT COUNT(*) FROM ORDERS WHERE AMOUNT > 250

Agregatlar va та ʻlumotlarni guruhlash.

Agregat funksiyalar jadval uchun natijaviy satr hosil qiladi.

Masalan: Buyurtma oʻrtacha narxi qancha?

SELECT AVG(AMOUNT) FROM ORDERS

Masalan, oraliq natijani topish lozim boʻlsin. Bu holda guruhlanishli soʻrov yordam beradi. Yaʻni SELECT operatorining GROUP BY ifodasi. Awal GROUP BY ifodasi qatnashgan quyidagi soʻrovni koʻramiz: Har bir xizmatchi uchun buyurtma oʻrtacha narxi qancha?

SELECT REP, AVG(AMOUNT) FROM ORDERS GROUP BY REP

REP maydoni bu holda guruhlash maydonidir, yaʻni REP maydonning hamma qiymatlari guruhlarga ajratiladi va har bir guruh uchun AVG(AMOUNT) ifodasi hisoblanadi.

Har bir ofis uchun sotuvlarning rejalashtirilgan hajmi 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

Guruhlash va ***HAVING*** yordamida ajratish. Shart bo‘yicha satrlami ajratish uchun ***WHERE*** ifodasidan foydalangan edik. Shart bo‘yicha guruhlami ajratish uchun ***HAVING*** operatori mavjuddir. Uning sintaksisi ***WHERE*** operatori bilan bir xil va ulardan birgalikda foydalanish mumkin.

Quyidagi so‘rovni ko‘ramiz: Buyurtmalar umumiy narxi 300$ dan ortiq xizmatchilar uchun buyurtma o‘rtacha narxi qanchaga teng ?

SELECT REP, AVG(AMOLINT) 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.

9- laboratoriya ishi

Shartlarni tekshirishga asoslangan operator va funksiyalar. Guruh funnksiyalar bilan ishlash. So‘rovlarni shakllantirish bo‘yicha misollar yechish.

Ishdan maqsad: ma‘lumotlar bazasini boshqarish dasturiy vositasi yoramida masofaviy serverga ulanishni sozlash va serverdagi ma‘lumotlarni qayta ishlash, tahrirlash ko‘nikmalariga ega bo‘lish.

Masalani qo‘yilishi: ma‘lumotlar bazasini boshqarish vositalari yordamida masofaviy serverdagi ma‘lumotlarni qayta ishlash.

Uslubiy ko`rsatmalar: Ma‘lumotlar bazasining o‘rganishda «view»lar, saqlangan protseduralar va funksiyalarni chetlab o‘tib bo‘lmaydi. «View» - bu

bazada saqlab qo‘yiladigan virtual jadvallardan tashkil topgan hisoblanadi. «View»larning jadvallardan farqi shundaki ularda ma‘lumotlar saqlanmaydi. Balki «view»lardagi so‘rovlar orqali ma‘lumotlar jadvallardan yoki boshqa «view»lardan olinadi. «View»larni doimiy bo‘lmagan o‘zgaruvchan virtual jadval deb qarashimiz mumkin. Misol uchun bizda juda ko‘p takrorlanadigan so‘rovimiz bo‘lishi mumkin va bu so‘rovni natijasi birnechta jadvallardan olinib chiqariladi, shuni «view» qilib saqlab qo‘yilsa bo‘ladi. «View» larga so‘rovga mos nom berib bazada saqlab qo‘yishimiz mumkin bo‘ladi. Murakkab so‘rovi berilgan bo‘lsin va uni natijasi quyidagicha. (9.1-rasm)

```
SELECT [FirstName] AS ‘ISMI‘,[LastName] AS ‘FAMILIYASI‘, [Title] AS ‘LAVOZIMI‘, [CompanyName] AS ‘TASHKILOT NOMI‘ ,[ContactName] AS ‘MA`SUL HODIM F.I.O‘, [ContactTitle] AS ‘MA`SUL XODIM LAVOZIMI‘
    FROM Employees AS emp
    LEFT JOIN Orders AS ord ON ord.EmployeeID = emp.EmployeeID
    LEFT JOIN Customers AS cust ON cust.CustomerID = ord.CustomerID
      ORDER BY [FirstName]
```

Results | Messages

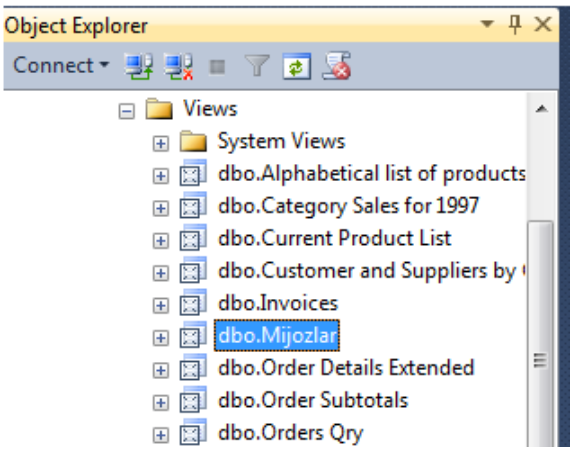
	ISMI	FAMILIYASI	LAVOZIMI	TASHKILOT NOMI	MA`SUL HODIM F.I.O	MA`SUL HODIM LAVOZIMI
1	Andrew	Fuller	Vice President, Sales	Blondesddsl pere et fils	Frederique Citeaux	Marketing Manager
2	Andrew	Fuller	Vice President, Sales	Morgenstern Gesundkost	Alexander Feuer	Marketing Assistant
3	Andrew	Fuller	Vice President, Sales	Berglunds snabbkop	Christina Berglund	Order Administrator
4	Andrew	Fuller	Vice President, Sales	Vins et alcools Chevalier	Paul Henriot	Accounting Manager
5	Andrew	Fuller	Vice President, Sales	Magazzini Alimentari Riuniti	Giovanni Rovelli	Marketing Manager
6	Andrew	Fuller	Vice President, Sales	Lonesome Pine Restaurant	Fran Wilson	Sales Manager
7	Andrew	Fuller	Vice President, Sales	Die Wandernde Kuh	Rita Muller	Sales Representative

9.1-rasm. Murakkab so‘rov natijasi

Yuqoridagi murakkab so‘rov berilgan va undan takror foydalanishimiz uchun «view» qilib saqlab qo‘yamiz. Buni uchun quyidagicha kod yoziladi.

```
CREATE VIEW Mijozlar
AS
SELECT [FirstName] AS ‘ISMI‘, [LastName] AS ‘FAMILIYASI‘, [Title] AS ‘LAVOZIMI‘, [CompanyName] AS ‘TASHKILOT NOMI‘, [ContactName] AS ‘MA`SUL HODIM F.I.O‘, [ContactTitle] AS ‘MA`SUL HODIM LAVOZIMI‘
    FROM Employees AS emp
```

```
LEFT JOIN Orders AS ord ON ord.EmployeeID = emp.EmployeeID
LEFT JOIN Customers AS cust ON cust.CustomerID = ord.CustomerID
```



9.2-rasm. Murakkab so‘rov natijasi

«View» lar bazada ko‘rinib turganidek Views bo‘lmida saqlanadi va ularni virtual jadvallar deb atashimiz mumkin bo‘ladi. «View» lar bilan ham jadvallar bilan ishlagandek ishlash mumkin. Masalan hozir yaratilgan «Mijozlar» viewdan ma‘lumot olmoqchi bo‘lsak quyidagicha komanda yoziladi.

```
SELECT [ISMI], [FAMILIYASI], [LAVOZIMI] , [TASHKILOT NOMI],
    [MA`SUL HODIM F.I.O], [MA`SUL HODIM LAVOZIMI]
FROM [Northwind].[dbo].[Mijozlar]
```

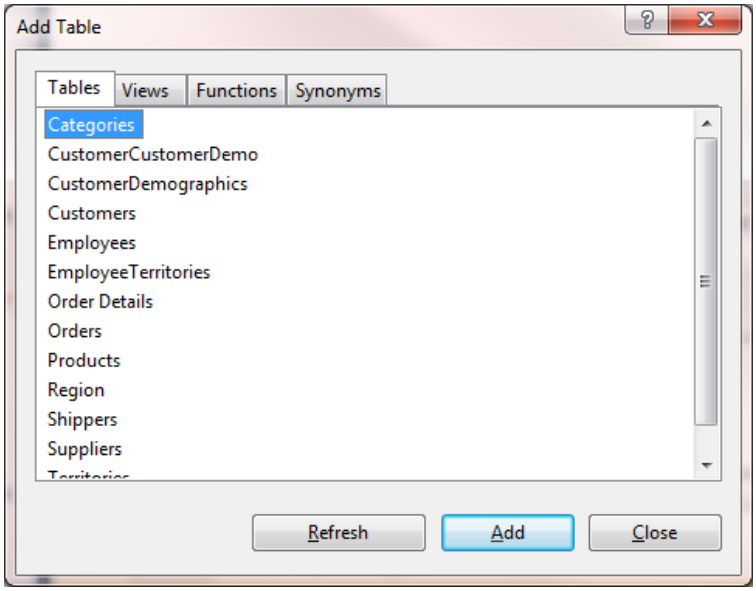
Bundan ko‘rinib turibdiki keltirilgan bazadan jadvalni nomi emas oldin hosil qilingan «Mijozlar» «view»iga murojat qilinyapti.

Results | Messages

	ISMI	FAMILIYASI	LAVOZIMI	TASHKILOT NOMI	MA'SUL HODIM F.I.O	MA'SUL HODIM LAVOZIMI
1	Nancy	Davolio	Sales Representative	Ernst Handel	Roland Mendel	Sales Manager
2	Nancy	Davolio	Sales Representative	Wartian Herkku	Pirkko Koskitalo	Accounting Manager
3	Nancy	Davolio	Sales Representative	Magazzini Alimentari Riuniti	Giovanni Rovelli	Marketing Manager
4	Nancy	Davolio	Sales Representative	QUICK-Stop	Horst Kloss	Accounting Manager
5	Nancy	Davolio	Sales Representative	Tradicao Hipermercados	Anabela Domingues	Sales Representative
6	Nancy	Davolio	Sales Representative	Tortuga Restaurante	Miguel Angel Paolino	Owner
7	Nancy	Davolio	Sales Representative	Tortuga Restaurante	Miguel Angel Paolino	Owner
8	Nancy	Davolio	Sales Representative	Romero y tomillo	Alejandra Camino	Accounting Manager
9	Nancy	Davolio	Sales Representative	Du monde entier	Janine Labrune	Owner
10	Nancy	Davolio	Sales Representative	Rattlesnake Canyon Gro...	Paula Wilson	Assistant Sales Representa...

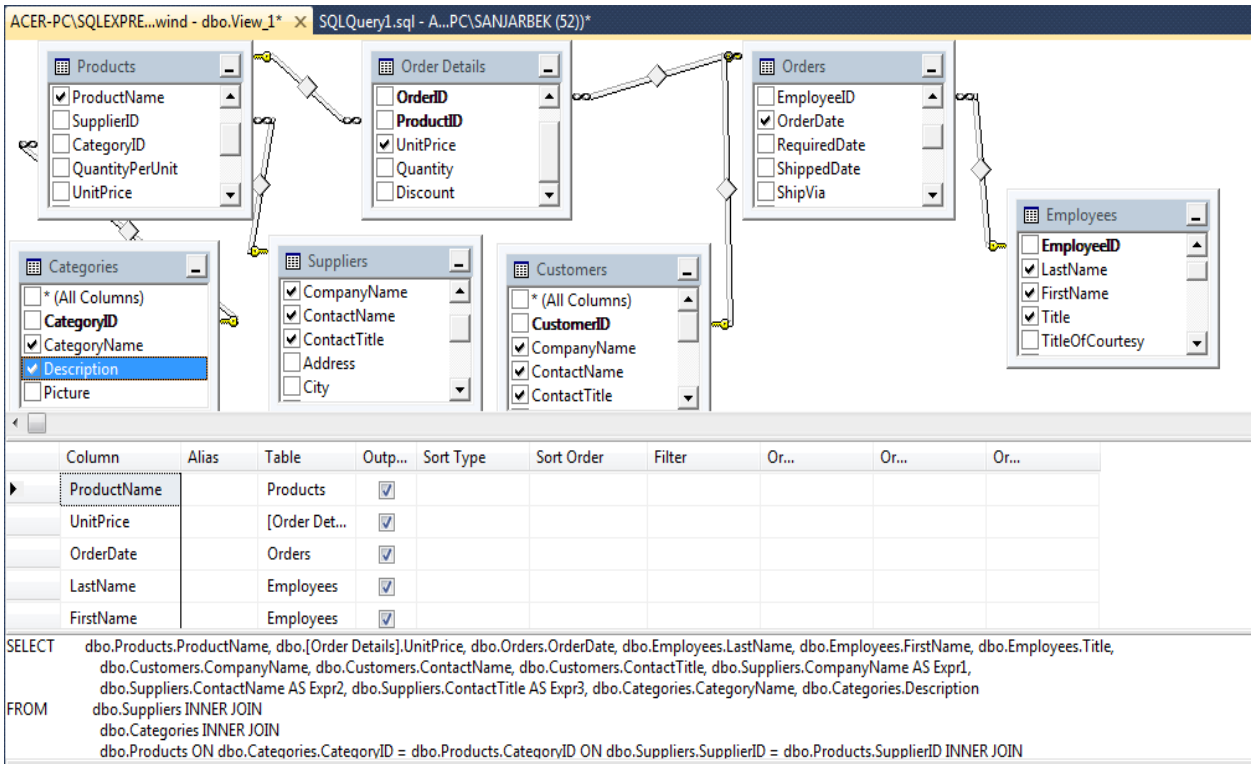
9.3-rasm. Mijozlar view so‘rovi natijasi

Shuningdek «view»larni grafik rejimda ham yaratish mumkin bo‘ladi. Buni uchun Views bo‘lmiga borib sichqonchaning o‘bg tugamsi boshiladi va quyidagi oyna ochiladi.



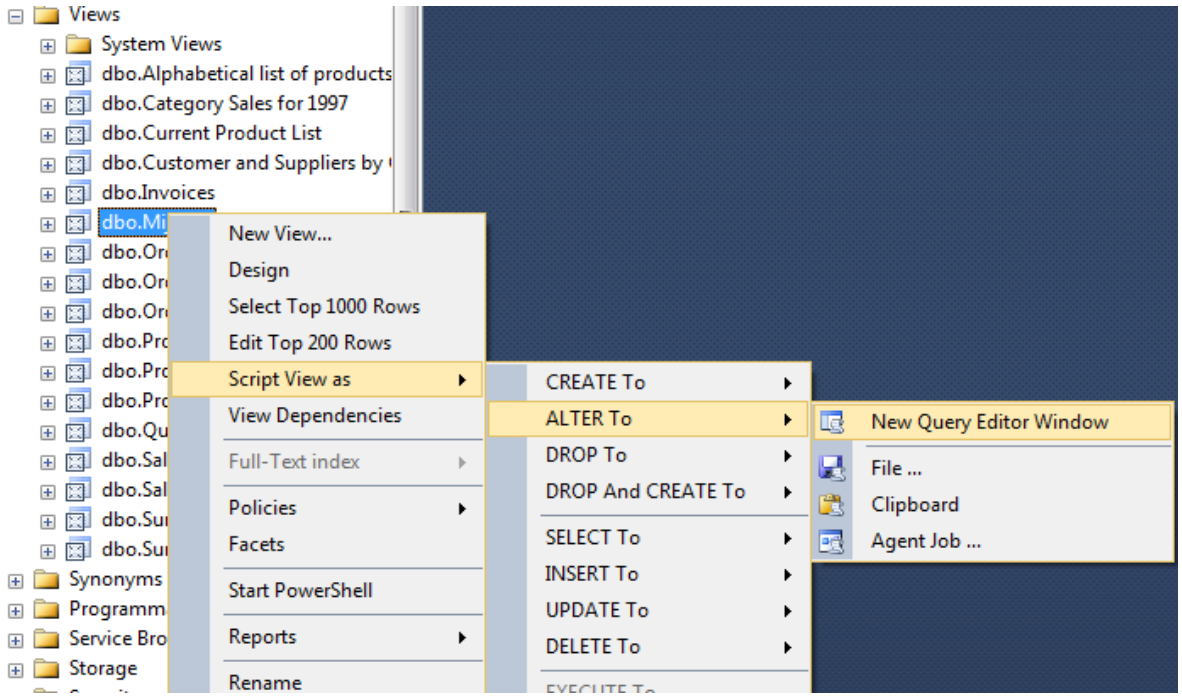
9.4-rasm. Mijozlar view so‘rovi natijasi

Kearkli jadvallar tanlab olinadi va shu jadvallardan ustunlarni tanlash kerak bo‘ladi. Jadvalar o‘zaro ***JOIN*** operatorlari bilan bog‘lanadi. Ctrl + S tugmasini bosib saqlab qo‘yish mumkin.



9.5-rasm. Viewni grafik shaklda yaratish oynasi

Oldindan hosil qilingan viewga qo‘shimchalar kiritish uchun ALTER buyrug‘udan foydalanamiz. Buni uchun viewni ustiga borib sichqonchaning o‘ng tugmasi bosiladi Script View as – ALTER To – New Query Editor Window ketma-ketligi tanlanadi.



9.6-rasm. Viewni o‘zgartishish oynasi

Mijoz viewda quyidagi kodlar hosil bo‘ladi va undan SELECT operatoridan keyin kerakli maydonlarni yozish yoki o‘chirish mumkin bo‘ladi.

```
USE [Northwind]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER VIEW [dbo].[Mijozlar]
AS
SELECT [FirstName] AS ‘ISMI‘,[LastName] AS ‘FAMILIYASI‘, [Title] AS ‘LAVOZIMI‘ ,[CompanyName] AS ‘TASHKILOT NOMI‘, [ContactName] AS ‘MA`SUL HODIM F.I.O‘ , [ContactTitle] AS ‘MA`SUL HODIM LAVOZIMI‘
FROM Employees AS emp
```

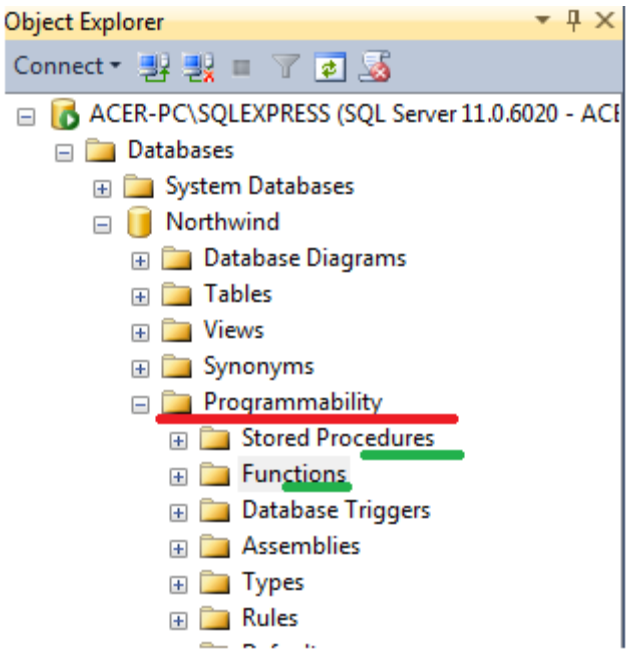
```
LEFT JOIN Orders AS ord ON ord.EmployeeID = emp.EmployeeID
LEFT JOIN Customers AS cust ON cust.CustomerID = ord.CustomerID
GO
```

Viewlarni o‘chirish uchun DROP buyrug‘idan foydalaniladi.

```
DROP VIEW Mijozlar
```

Viewlar haqida ma‘lumotlarga ega bo‘lindi endi foydalanish davomida yanada ko‘proq ma‘lumotlarga ega bo‘lish mumkin bo‘ladi.

Saqlangan protseduralar va funksiyalar haqida ham to‘xtalib o‘tsak. Saqlangan protsedura ma‘lumotlar bazasining maxsus obyekti bo‘lib, unda SQL tilida yozilgan kodlar saqlanadi. Saqlangan protsedura bu boshqa dasturlash tillaridagi oddiy protseduralarga o‘xshab ketadi. Ularda ham kiruvchi va chiquvchi parametrlari, o‘zgaruvchilari bo‘lishi mumkin. Protseduralarda baza bilan bog‘liq bo‘lgan DDL va DML operatsiyalr, sikllar bajarilishi mumkin. Yuqorida keltirilgan fikrlar funksiyalarga ham ta‘luqli bo‘lib, funksiyaning protseduradan farqli tamoni shundaki funksiyani SQL so‘rovlari ichida oddiy operator kabi ishlatish mumkin. Protsedurani chaqirish uchun esa «Execute» buyrug‘i ishlatiladi. SQL Serverda juda ko‘plab sistema shunksiyalari mavjud, bularga oldingi darsda o‘rganilgan GETDATE(), CAST() larni misol qilsih mumkin. Shuningdek funksiyalari o‘zimiz ham yaratishimiz mumkin. Hozirgi kunda barcha ma‘lumotlar bazalarida saqlangan protseduralardan foydalaniladi. Shunday qilib SQL tilida kodlarni katta bo‘lib ketishini oldini olishi uchun ko‘p ishlatiladigan so‘rovlarni saqlab qo‘yishda protsedura yoki funksiyalardan foydalaniladi. Bundan tashqari bitta saqlangan pritsedurani ichidagi so‘rovdan boshqa saqlangan protsedurani ham chaqirishimiz mumkin bo‘ladi. SQL serverda funksiyalar va protseduralar Programmablity bo‘lmida saqlanadi. Functions bo‘lmida foydalanuvchi funksiyalari va Stored Procedures bo‘lmida protseduralar saqlanadi. (5-rasm)



9.7-rasm. Funksiyalar va protseduralar oynasi

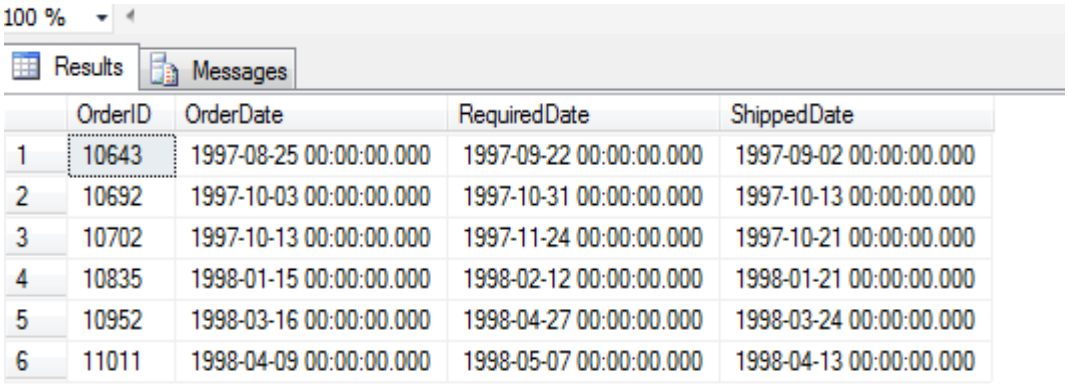
Protsedura yaratishni ko‘rib chiqamiz. Buni uchun yangi so‘rov oynasi ochiladi va yaratishning kodlari yoziladi.

```
USE [Northwind]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Buyurtmalar] @CustomerIDnchar(5)
AS
SELECT OrderID,
  OrderDate,
  RequiredDate,
  ShippedDate
FROM Orders
WHERE CustomerID = @CustomerID
ORDER BY OrderID
```

Yuqoridagi kod yordamida Buyurtmalar buyurtmalar protsedurasi yartilgan va unga kiruvchi parameter sifatida CustomerID tanlangan. Bu protsedurani tahlil qiladigan bo‘lsak Orders yani burutmalar jadvalidan CustomerID parametri protseduraga jo‘natilgan CustomerID ga teng bo‘lgan mijozlar uchun OrderID, OrderDate, RequiredDate, ShippedDate sutunlarini chiqarib beradi.

Protsedurani ishga tushirish uchun SQL so‘rovlar oynasiga yuqidagicha kod yoziladi.

EXECUTE Buyurtmalar @CustomerID = ‘ALFKI‘ yoki EXECUTE Buyurtmalar ‘ALFKI‘

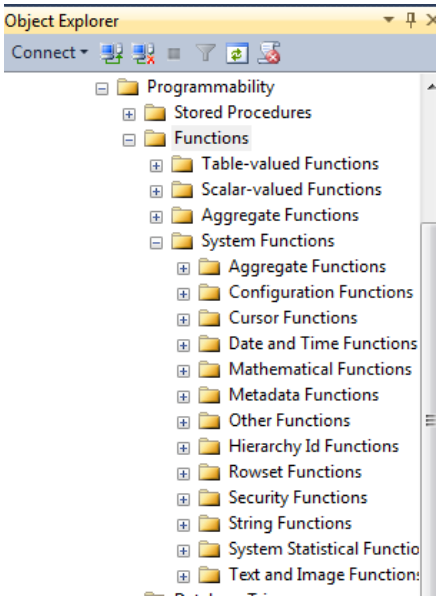

100 %

Results | Messages

	OrderID	OrderDate	RequiredDate	ShippedDate
1	10643	1997-08-25 00:00:00.000	1997-09-22 00:00:00.000	1997-09-02 00:00:00.000
2	10692	1997-10-03 00:00:00.000	1997-10-31 00:00:00.000	1997-10-13 00:00:00.000
3	10702	1997-10-13 00:00:00.000	1997-11-24 00:00:00.000	1997-10-21 00:00:00.000
4	10835	1998-01-15 00:00:00.000	1998-02-12 00:00:00.000	1998-01-21 00:00:00.000
5	10952	1998-03-16 00:00:00.000	1998-04-27 00:00:00.000	1998-03-24 00:00:00.000
6	11011	1998-04-09 00:00:00.000	1998-05-07 00:00:00.000	1998-04-13 00:00:00.000

9.8-rasm. Buyurtmalr protsedurasining natijasi

Funksiyalarga keladigan bo‘lsak foydalanuvchi tomonidan yaratilgan funksiyalar yuqoridan keltirganimizdek Functions bo‘lmida saqlanadi. Rasmdan ko‘rinib turibdiki bu bo‘limda foydalanuvchi tomonidan yozilgan funksiyalar yo‘q faqatgina System Functions bo‘lmida SQL serverning standart funksiyalri mavjud.



9.9-rasm. Buyurtmalar protsedurasining natijasi

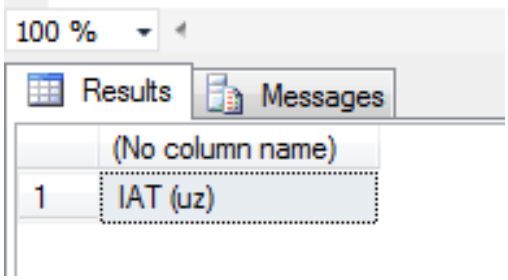
Talabaning guruhi raqami kiritilsa u qaysi yo‘nalishda o‘qishligi haqida ma‘lumot chiqaruvchi funksiya yarataylik. Funksiyaning nomini [Yo`nalishi] va unga kriuvchi parametrni esa @Guruh deb belgilaylik. Bu funksiya quyidagicha ko‘rinishga ega bo‘ladi.

```
CREATE FUNCTION [Yo`nalishi]
(@Guruh varchar(20))
RETURNS varchar(20)
AS
BEGIN
declare @guruh1 varchar(20)
select @guruh1 =case @Guruh
when ‘119-18‘ then ‘IMT (uz)‘
when ‘120-18‘ then ‘IMT (rus)‘
when ‘125-18‘ then ‘IAT (uz)‘
when ‘126-18‘ then ‘IAT (uz)‘
when ‘127-18‘ then ‘IAT (rus)‘
when ‘85-17‘ then ‘IMT (uz)‘
else ‘Buni aniqlashda muammo bor (Noqulaylik uchun uzr so`raymiz)‘
end
```

```
return @guruh1
END
```

Funksiyani ishga tushirish uchun so‘rovlar oynasiga quyigicha kod yoziladi.

```
SELECT dbo.[Yo`nalishi]('126-18')
```



9.10-rasm. Buyurtmalr protsedurasining natijasi.

Funksiyalarni o‘chirish uchun esa DROP FUNCTION buyrug‘idan foydanalaniladi.

Agar foydalanuvchi tomonidan belgilangan funksiya (UDF) SCHEMABINDING bandini ishlatmasdan yaratilgan bo‘lsa, asosiy obyektlardagi o‘zgarishlar funksiyaning aniqlanishiga ta‘sir qilishi va funksiya chaqirilganda notog‘ri natijalarni olishi mumkin.

BEGIN ... END blokidagi ko‘rsatmalar nojo‘ya ta‘sir ko‘rsatmaydi. Funksiyalarning nojo‘ya ta‘siri - bu resursning holatidagi doimiy o‘zgarishlar, ularning maydoni funksiyalar chegarasidan tashqarida joylashganligi, masalan, ma‘lumotlar bazasi jadvalini o‘zgartirish. Funksiya ichidagi ko‘rsatmalar faqat ushbu funksiyaga mos keladigan ob‘ektlarni o‘zgartirishi mumkin. Ma‘lumotlar bazasi jadvallaridagi o‘zgarishlar, berilgan funksiyada lokal bo‘lmagan kursorlar bilan ishlash, elektron pochtani yuborish, katalogni o‘zgartirishga urinish, foydalanuvchiga qaytarilgan natijalar to‘plami - bu funksiya ichida bajarib bo‘lmaydigan harakatlar misollari.

Funksiyalar doirasida ruxsat berilgan ko‘rsatmalar turlari quyidagilarni o‘z ichiga oladi.

- Berilgan funksiya uchun o‘zgaruvchilar va kursorlarni aniqlash uchun ishlatiladigan DECLARE ko‘rsatmalari.

- Berilgan funksiya uchun lokal obyektlarga qiymatlarni berish, masalan, SET bayonotidan foydalanib, skalalar va jadvalga mahalliy o‘zgaruvchilarga qiymatlar berish.
- Oqimni boshqarish bo‘yicha TRY ... CATCH ko‘rsatmalari
- Funksiyaga tegishli bo‘lgan o‘zgaruvchiga qiymatlarni belgilovchi va tanlov ro‘yxatlari bo‘lgan SELECT ko‘rsatmalari.
- Jadval parametrlarini funksiyaga mos ravishda o‘zgartiradigan UPDATE, INSERT va DELETE ko‘rsatmalari.
- Kengaytirilgan saqlangan protsedurani chaqiradigan EXECUTE.

SQL da yuqori natijalarga, optimal so‘rovlarga erishish uchun funksiyalar orasida emas, balki tayanch jadvallar o‘rtasida birlashtirishni belgilash tavsiya etiladi.

ADABIYOTLAR

1. Knuth, D. E. The Art of Computer Programming Vol. I: Fundamental Algorithms, Addison – Wesley, Reading, Mass. (Русский перевод: Кнут Д. Искусство программирования. Том 1: Основные алгоритмы. – М., Издательский дом «Вильямс», 2000.)
2. Джон Бентли. Жемчужины программирования.- СПб: Питер, 2002.-272 с.
3. Дейт К., Хью Дарвен. Основы будущих систем баз данных. - М: Янус-К, 2004.
4. Нагао М., Катаяма Т., Уэмура С. Структуры и базы данных. - М.: Мир, 2001. - 197 с.
5. Бойко В.В., Савинков В.М. Проектирование баз данных информационных систем. - М.: Финансы и статистика, 2009. - 351 с.
6. Кириллов В.В. Структурированный язык запросов (SQL). - СПб. ИТМО, 2000. - 80 с.
7. Дейт К. Введение в системы баз данных. 8-е изд.,- М.: СПб.: Вильяме.- 2005.
8. Хомоненко А.Д. Базы данных. Учебник для высших учебных заведений / Под ред. np(xj). А. Д. Хомопепко. - 6-е изд., доп. - СПб.: КОРОНА-Век, 2009. - 736 с.
9. Гектор Гарсиа-Молина, Джеффри Ульман, Дженифер Уидом. Системы баз данных. Полный курс. - Москва, Санкт-Петербург, Киев, Вильямс, 2003.
10. Система управления базами данных: P:Base System V/ С. В. Горин, С. В. Кравченко, Г. Г. Кузина, А. В. Силантьева. -Москва: Изд-во МГУ, 2003.
11. Архангельский А.Я. Работа с локальными базами данных в Delphi 5: производственно-практическое издание. – М.: Наука, 2000.
12. Грабер М. Введение в SQL. - М.: Лори, 2001. - 379 с.
13. Ким М. Курс лекции по «Транзакционная обработка». - Т., 2001.
14. Когаловский М.Р. Энциклопедия технологий баз данных. - М.: Финансы и статистика, 2002.

15. Боуман Д, Эмерсон С., Дарновски М. Практическое руководство по SQL. - Киев: Диалектика, 1997.

16. Джексон Г. Проектирование реляционных баз данных для использования с микроЭВМ. - М.: Мир, 2004. - 252 с.

17. Диго С.М. Проектирование и использование баз данных. - М.: Финансы и статистика, 1998. - 208 с.

MUNDARIJA

QAYDLAR UCHUN

Tuzuvchilar: Sodiqova Sh.Sh., Bekturdiyev S.Sh.

MA’LUMOTLAR BAZASINI BOSHQARISH VA DASTURLASH TEXNOLOGIYALARI

fanidan laboratoriya mashg‘ulotlarni bajarish uchun

USLUBIY KO‘RSATMALAR

I qism

Muharrir: Sidikova K.A.