# **18-ma’ruza. PHP dasturlash tilida ma`lumotlar bazasi bilan ishlash.**

**Reja:**

1. MySQL MBBT ning asosiy operatorlari
2. MySQL so’rovlari bilan ishlovchi PHP buyruqlari bilan tanishish. mysql\_connect va mysql\_db\_select operatorlari bilan ishlash.
3. PHP sahifasi orqali ma’lumotlar bazasiga so’rovlarni tashkil etish. mysql\_query operatori imkoniyatlari.
4. MySQL tizimida ishlash uchun ma'lumotlar bazasini tanlash.

|  |  |
| --- | --- |
| **importance_of_education_1** | **O`quv modullari**  *MySQL, MBBT, PHP da MBBT bilan ishlash,mysql\_connect, mysql\_query* |

**MySQL MBBT ning asosiy operatorlari**

Ma'lumotlar bazasi bilan quyidagi amallarni bajarish mumkin:

1. Ma'lumotlar bazasini (MB) yaratish, o'chirish va o'zgartirish.
2. Foydalanuvchilarga vakolat berish, o'chirish, o'zgartirish, qo'shish.
3. Ma'lumotlar bazasi (jadval va yozuvlar)ga ma'lumotlarni kiritish, o'chirish va o'zgartirish;
4. MB ma'lumotlarini saralash.

Birinchi ikkita amalni bajarish huquqi faqatgina MBBT administratorlari va maxsus huquqi bor foydalanuvchilarga berilgan.

Ma'lumotlar bilan ishlashda ma'lumotlar bazasini yaratish, lozim bo'lganda bu ma'lumotlar saqlanadigan jadvallar strukturasini o'zgartirish va o'chirish kerak bo'ladi. Buning uchun SQL ning qu­yidagi operatorlari mavjud CREATE TABLE, ALTER TABLE va DROP TABLE.

Ma'lumotlar bazasini yaratish: CREATE TABLE operatori

CREATE TABLE operatori joriy ma'lumotlar bazasida berilgan nom bilan jadval yaratadi. Agar ma'lumotlar bazasi aktivlashmagan bo'lsa yoki berilgan jadval nomi mavjud bo'lsa, buyruqni bajarishda xatolik haqida ogohlantiradi.

Sintaksisi:

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] jadval\_nomi [(ustunlarni\_aniqlash,...)] |jadval\_opsiyasi] [select\_ifoda]

MySQL 3.22 va undan keyingi versiyalarida jadval nomini ma’lumotlar\_bazasi\_nomi.jadval\_nomi kabi ko'rsatadi. Yozuvning bu ko'rinishi ko'rsatilgan ma'lumotlar bazasi joriy yoki joriymasligiga bog'liq emas.

MySQL 3.23-versiyasida vaqtinchalik jadval yaratish uchun TEMPORARY kalit so'zi ishlatiladi. Vaqtinchalik jadval bog'lanish yakunlanganda avtomatik tarzda o'chiriladi, uning nomi esa beril­gan bog'lanish davomida haqiqiy hisoblanadi. Bu shuni anglatadiki, ikkita turli bog'lanishda mavjud jadval xuddi shunday nomli jadval bilan yoki bir xil nomli vaqtinchalik jadvallar bir-biri bilan konf-liktsiz ishlatilishi mumkin (yoki mavjud jadval vaqtinchalik jad­val o'chirilguncha yashirin bo'lib turadi). MySQL 4.0.2-versiyasida vaqtinchalik jadvallarni yaratish uchun CREATE TEMPORARY TABLES vakolatiga ega bo'lish zarur.

MySQL 3.23 va undan keyingi versiyalarida ko'rsatilgan jadval mavjud bo'lsa, xato bermasligi uchun IF NOT EXISTS kalit so'zi ishlatiladi. Bu holda jadval strukturalarining bir xilligi tekshirilmay-di.

Har bir jadval ma'lumotlar bazasi katalogida aniqlangan fayllar to'plami singari tasvirlanadi.

ustunlarni\_aniqlash parametri jadvalda qanday ustunlar bo'lishi kerakligini belgilaydi. Jadvalning har bir ustuni bo'sh bo'lishi (NULL), odatga ko'ra qiymatga ega bo'lishi, kalitli yoki avtoinkrement (AUTO\_INCREMENT) bo'lishi mumkin. Bundan tashqari, har bir ustun uchun unda saqlanadigan ma'lumotlar tipini ko'rsatish kerak. Agar NULL ham, NOT NULL ham ko'rsatilmasa, u holda ustun NULL ko'rsatilgandek, interpretatsiya qilinadi. Agar maydon avtoinkrement sifatida belgilansa, u holda jadvalga ma'lumot kiritilganda va bu maydonga bo'sh qiymat (NULL, yoki hech narsa yozilmaganda) yoki 0 kiritilganda uning qiymati har gal 1 ta qiymatga oshiriladi. Jadvalda avtoinkrement maydon faqatgina bitta bo'lishi mumkin, shuning uchun uni indekslash shart emas. Avtoinkrement ketma-ketligi 1 dan boshlanadi.

Qoidaga ko'ra ustunlar quyidagicha beriladi (ustunlarni\_aniqlash):

ustun\_\_nomi tipi [NOT NULL | NULL] [DEFAULT odatga ko'ra qiymati] [AUTO\_INCREMENT][PRIMARY KEY] [reference\_definition]

Ustun tipi (ustunlarni\_aniqlash parametrining ustunnomi tipi) quyidagilardan birortasini qabul qilishi mumkin:

• butun: INT[(length)] [UNSIGNED] [ZEROFILL]

* haqiqiy: REAL[(length,decimals)] [UNSIGNED] [ZERO-FILL]
* belgili: CHAR(length) [BINARY] va VARCHAR(length) [BI­NARY]
* katta obyektlar bilan ishlash uchun: BLOB
* matnli: TEXT
* sana va vaqt: DATE va TIME

Maydon (ustun) uzunligi kiritilayotgan qiymatda jami nechta raqam (belgi) qatnashishi kerakligini aniqlaydi. dec qatnashgan may-donlarda esa bu o'nlik kasrlarda nuqtadan so'ng, ya'ni kasr qismi nechta raqamdan iborat bo'lishi kerakligini bildiradi. Bu sonlar faqat­gina ustunning maksimal uzunligini hisoblash uchun ishlatiladi.

ustunlarni\_aniqlashda ustunlar soni va tipi bilan birga chegara-lash va tekshirish uchun kalitli va indeksli ustunlar ro'yxatini ham berish mumkin:

PRIMARY KEY (indekslangan\_ustun\_nomi, ...)

yoki

KEY [indeks\_nomi] (indekslangan\_ustun\_nomi,...) yoki

INDEX [indeks\_nomi] (indekslangan\_ustun\_nomi,...) yoki

U NIQUE [IN DEX] [indeks\_nomi] (indekslangan\_ustun\_ nomi,...) yoki

FULLTEXT [INDEX] [indeks\_\_nomi] (indekslangan\_ustun\_ nomi,...) yoki

[CONSTRAINT symbol]

FOREIGN KEY [indeks\_nomi] (indekslangan\_ustun\_nomi,...)

[reference\_definition]

yoki

CHECK (expr)

FOREIGN KEY, CHECK va REFERENCES lar ayni vaqtda MySQLda hech qanday vazifani bajarmaydi. Ular boshqa SQL­

serverlar bilan hamkorlik qilish uchun qo'shilgan. Shuning uchun ularga to'xtalmaymiz.

Ko'p hollarda ustunga kiritilgan qiymatlar bir-biridan farq qilishi mumkin. Agar ustun uchun UNIQUE cheklanish o'rnatilsa, bu us­tunga mavjud qiymatni kiritishga urinish rad etiladi. Bu cheklanish bo'sh emas (NOT NULL) deb e'lon qilingan ustunlarga qo'llanishi mumkin.

Masalan,

CREATE TABLE Student (id INT NOT NULL UNIQUE, surname VARCHAR(100), name VARCHAR(50), stipend INT, kurs INT, city VARCHAR(100), birthday DATA, univer INT );

Unikalligi talab qilinadigan ustunlardagi kalitlar (birlamchi ka-litlardan tashqari) kandidat kalitlar yoki kalitlar deyiladi.

Jadval cheklanishi UNIQUE ustunlar guruhiga o'rnatilishi mumkin. Bu bir necha ustunlar qiymatiari kombinatsiyasi unikalligini ta'minlaydi.

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. Birlamchi kalitlar NULL qiymatiga ega bo'lishi mumkin emas.

Masalan:

CREATE TABLE Student   
(id INT NOT NULL PRIMARY KEY, surname VARCHAR(100), name VARCHAR(50), stipend INT, kurs INT, city VARCHAR(100), birthday DATA, univjd INT);

Jadval yaratishda, uni ba'zi xususiyatlari (jadval\_opsiyasi)ni ham ko'rsatish kerak, masalan:

* jadval tipi: TYPE = {BDB | HEAP | ISAM | InnoDB | MERGE | MRGMYISAM | MY1SAM }
* avtoinkrement hisoblagichining boshlang'ich qiymati: AUTO INCREMENT = son
* jadval satrining o'rtacha uzunligi: AVGROWLENGTH = son
* jadval uchun izoh (60 belgidan iborat satr): COMMENT = «satr»
* satrning taxminiy eng katta va eng kichik qiymati: MAX\_ ROWS = son va MIN\_ROWS = son

NOT NULL jadval ustunidagi barcha yacheykalar qiymatga ega bo'lishi zarurligini bildiradi, agar NOT NULL ko'rsatilmasa, ustun qiymatlari bo'sh (NULL) bo'lishi mumkin.

AUTOINCREMENT — sonli ustunlar uchun ishlatiladigan MySQL ning maxsus imkoniyati. Agar jadvalga satrlarni qo'yishda ushbu maydonni bo'sh qoldirilsa, MySQL avtomatik tarzda identifikatorning unikal qiymatini qo'yadi. Bu qiymat maydondagi mavjud maksimal qiymatdan bittaga katta bo'ladi. Har bir jadvalda bunday maydon faqatgina bitta bo'ladi. AUTO\_INCREMENT ustunlari indekslangan bo'lishi zarur.

PRIMARY KEY kalit so'zi ustun nomidan so'ng aniqlanib, ush­bu ustun jadval uchun birlamchi kalit ekanligini anglatadi. Ushbu ustundagi ma'lumotlar unikal bo'lishi lozim. MySQL ushbu ustunni avtomatik tarzda indekslaydi.

UNSIGNED chegaralashi qo'yilgan ustun qiymati yoki butun son yoki nol bo'lishi mumkin.

Talabalar portalining o'quv jarayoni jadvallarini yaratamiz.

mysqi>CREATE TABLE Student

(stud\_id INT UNSIGNED NOT NULL PRIMARY KEY, surname VARCHAR(100), name VARCHAR(100), stipend INT, kurs INT, city VARCHAR(200), birthday DATE, univjd INT UNSIGNED NOT NULL );

mysql>CREATE TABLE exam\_marks (exam\_id INT UNSIGNED NOT NULL PRIMARY KEY, stud\_id INT UNSIGNED NOT NULL, subj\_id INT UNSIGNED NOT NULL, mark INT, exam\_date DATA );

mysql>CREATE TABLE subject

(subj\_id INT UNSIGNED NOT NULL PRIMARY KEY,  
subj\_name VARCHAR(IOO) NOT NULL,  
hour INT, semestr INT);  
mysql>CREATE TABLE lecturer

(lecturerJd INT UNSIGNED NOT NULL PRIMARY KEY, surname VARCHAR(100) NOT NULL, name VARCHAR(100) NOT NULL, city VARCHAR(200), univjd INT UNSIGNED NOT NULL );

mysql>CREATE TABLE subjject

(lecturerJd INT UNSIGNED NOT NULL, subjjd INT UNSIGNED NOT NULL); mysql>CREATE TABLE univercity

(univ\_id INT UNSIGNED NOT NULL PRIMARY KEY, univ\_name VARCHAR(100) NOT NULL, rating INT, city VARCHAR(200));

Jadval yaratishda CREATE buyrug'ining yana bir elementi SE­LECT ifodasi (select jfodasi) ham qatnashishi mumkin. [IGNORE | REPLACE] SELECT ... (SELECT ning ixtiyoriy to'g'ri ifodasi)

Agar CREATE buyrug'i yordamida jadval yaratishda select\_ifodasi ko'rsatilsa, u holda tanlashda olingan barcha maydonlar yara­tilgan jadvalga qo'shiladi.

MySql ning maxsus SHOW buyrug'i yordamida ma'lumotlar ba­zasi tarkibi (ma'lumotlar bazasi jadvallari va jadval maydonlari) ni ko'rish mumkin.

misol. Barcha ma'lumotlar bazalarini ko'rsatish:

mysql>SHOW databases;

misol. talaba ma'lumotlar bazasini joriy etish (aktivlashtirish):

mysql>use talaba;

misol. Joriy (aktiv) bazadagi barcha jadvallarini ko'rsatish:  
mysql>show tables;

MySQL ma'lumotlar bazasidagi jadvallar ro'yxatini chiqazadi:

Tables in talaba

student

exam\_marks

subject

lecturer

subjject

univercity

6 rows in set (0.07 sec)

MySQL so’rovlari bilan ishlovchi PHP buyruqlari bilan tanishish. mysql\_connect va mysql\_db\_select operatorlari bilan ishlash. PHP sahifasi orqali ma’lumotlar bazasiga so’rovlarni tashkil etish. mysql\_query operatori imkoniyatlari.

MySQL ma'lumotlar bazasiga WEB orqali PHP texnologiyasi yordamida so'rovlarni jo'natish algoritmi:

1. Foydalanuvchi tomonidan kiritilgan ma'lumotlarni filtrlash va tekshirish.
2. MySQL tizimi bilan bog'lanishni o'rnatish. Agar urinish muvaffaqiyatsiz yakunlansa, mos xabar chiqaziladi va jarayon yakunlanadi.
3. MySQL tizimida ishlash uchun ma'lumotlar bazasini tanlash. Agar tanlash urinishi muvaffaqiyatsiz yakunlansa, mos xabar chiqaziladi va jarayon yakunlanadi. So'rovlarni qayta ishlash uchun bir vaqtning o'zida bir nechta ma'lumotlar bazasini ochishga ruxsat beriladi.
4. Tanlangan bazada so'rovlarni qayta ishlash (ma'lumotlarni qo'shish uchun interfeys yaratish, ma'lumotlar bazasiga ma'lumotlarni yozish, ma'lumotlar bazasida saqlanayotgan ma'lumotlarni foydalanuvchiga taqdim etish).
5. So'rovlarni qayta ishlash tugatilgandan so'ng ma'lumotlar bazasi tizimi bilan bog'lanishni yopish yoki tugatish.

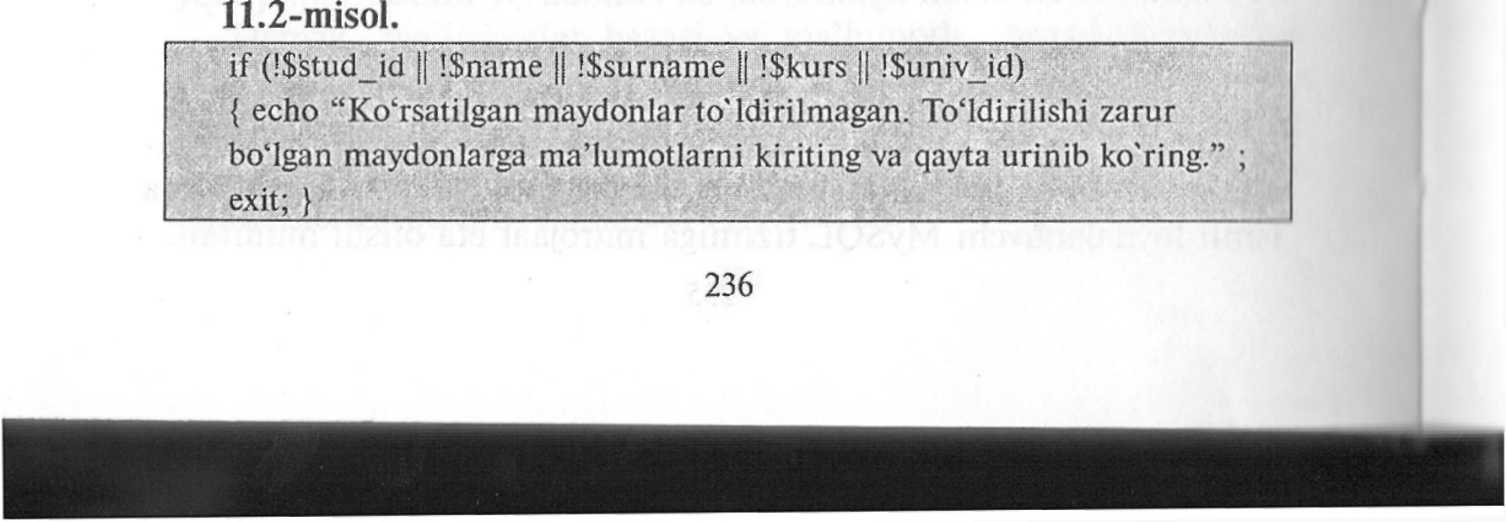
*1. Foydalanuvchi tomonidan kiritilgan ma'lumotlarni filtrlash va tekshirish.*

Avval foydalanuvchi tomonidan kiritilgan ma'lumot atrofidagi tasodifan yozilgan barcha ortiqcha bo'sh belgilarni (probellarni) o'chirish zarur. Buning uchun trim() funksiyasini qo'llanadi.

misol.

trim($city);

Keyingi bosqich — foydalanuvchi barcha maydonlarni to'ldirganligini tekshirish. trim() funksiyasi yordamida o'zgaruvchilar tarkibidagi qiymatlar tahrirlangandan so'ng o'zgaruvchilarni qiymatiari mavjud yoki mavjud emasligi tekshiriladi:



Foydalanuvchi ma'lumotlarini MySQL ma'lumotlar bazasiga yozishdan avval addslashes() va stripslashes() funksiyalari yordamida tahrirlab olish mumkin.

htmlspecialchars() funksiyasi ma'lumotlarda uchraydigan ((&) ampersand, (<) kichik, (>) katta, (") ikkitalik qo'shtirnoq) belgilarini HTML kodirovkasiga o'tkazish uchun ishlatiladi.

*2. MySQL tizimi bilan bog'lanishni o'rnatish.*

**mysql\_connect()** **funksiyasi**

mysql\_pconnect() funksiyasi MySQL tizimi bilan bog'lanishni o'rnatadi. Sintaksisi:

int mysql\_pconnect ([string xost [:port][:/soketga yo'l] [, string foydalanuvchi\_nomi] [,string parol])

yoki

resource mysql\_pconnect ([server satri [,username satri[,password satri [, int new\_link [, int client\_flags]]]]])

xost parametrida MySQL tizimining vakolatlar jadvalida ko'rsatilgan kompyuter xosti nomi beriladi. Bu nom MySQL ishlayotgan Web-serverga so'rovlarni yo'naltirish uchun ishlatiladi, bi­roq MySQL tizimiga masofadan turib ham bog'lanish mumkin. Xost nomi berilganda shart bo'lmagan parametrlari — port nomeri, soket(lokal xost uchun) yo'lini ham ko'rsatish mumkin. foydalanuvchi\_nomi va parol parametrlari MySQL vakolatlari jadvalida be­rilgan foydalanuvchi nomi va paroli bilan mos tushishi kerak. Agar xost parametri berilmagan bo'lsa, mysql\_pconnect() lokal xost bilan aloqani o'rnatishga harakat qiladi.

11.3-misol. MySQL bilan ochiq bog'lanish:

@ $db = mysql\_pconnect(“localhost”, “goyipov”, “12345”);

Berilgan misolda localhost — kompyuter xost nomi, goyipov – foydalanuvchi nomi, 12345 esa — parol. mysql\_pconnect() funksiyasi oldidan ishlatiladigan @ belgisi bog'lanish urinishi muvaffaqiyatsiz yakunlanganda chiqadigan xatolarni yashirish uchun ishlatiladi. Bu belgi die() funksiyasi chiqazadigan xatolar haqidagi xabarlarni al­mashtiradi.

Agar funksiyaga bir xil avvalgi parametrlar bilan ikki marta murojaat etilsa, yangi bog'lanish o'rnatilmaydi, lekin murojaat (ssil-ka)ni eski bog'lanishga qaytaradi. Bu holatga duch kelmaslik uchun, newlink parametri ishlatiladi. Bu parametr ixtiyoriy holatda yana bir bog'lanishni ochishga yordam beradi.

client\_flags parametri — bu quyidagi konstantalarni kombinatsiyasi: MYSQL\_CLIENT\_COMPRESS (siqish protokolini ishlatadi), MYSQL\_CLIENT\_IGNORE\_SPACE (funksiya nomidan so'ng probellarni o'qish uchun imkoniyat yaratadi), MYSQL\_CLIENT\_INTERACTIVE (bog'lanish yopilguncha- wait\_timeout bilan birga — interactive timeout sekund kutadi).

new\_link parametri PHP 4.2.0-versiyasidan, client\_flags para­metri esa PHP 4.3.0-versiyasidan boshlab qo'shilgan.

Tizim bilan bog'lanish agar mysql\_close() funksiyasi yordamida yopilmagan bo'lsa, skript bajarilishi tugatilganda yopiladi.

11.4-misol goyipov foydalanuvchisi uchun «12345» paroli bilan lo­kal serverdagi ma'lumotlar bazasiga bog'lanishni o'rnatamiz:

<?php

$conn = mysql\_pconnect(“localhost”, “goyipov”, “12345”)

or die (“Bog'lanish o'rnatilmadi: mysql\_error());

echo “Bog'lanish o'rnatildi”;

?>

Natija: Bog'lanish o'rnatildi

Agar dastur turli xostlarda bir nechta MySQL tizimlari bi­lan bog'lanishni o'rnatsa, kerakli MySQL tizimiga mysql\_pconnect() funksiyasi qaytargan bog'lanish identifikatordan foydalanib so'rovlarni yo'llash mumkin.

11.5-misol.

<?php

$linkl = @mysql\_pconnect("localhost", "goyipov", "12345") or die("localhost serveriga boglana olmadi!");

$link2 = @mysql\_pconnect(“www.birorhost.uz”, “foydalanuvchi”, “parol”) or die (“MySQL server ga bo’glana olmadi!”);

?>

$linkl va $link2 identifikatorlari ma'lumotlar bazasiga so'rovlar bilan oxirgi murojaatni uzatadi.

Ko'p foydalanuvchilar muhitida tizim resurslarini tejash uchun mysql\_pconnect() ni mysql\_connect() bilan birgalikda ishlatish tavsiya etiladi. mysql\_pconnect() funksiyasining parametr tiplanri va qaytaradigan qiymatiari mysql\_connect() funksiyasi bilan mos tushadi, bittagina farqi mysql\_connect() funksiyasi ma'lumotlar bazasiga doimiy bog'lanib turish uchun ishlatiladi.

*MySQL tizimida ishlash uchun ma'lumotlar bazasini tanlash.*

MySQL tizimiga muvaffaqiyatli bogianib boigandan so'ng, tizimda mavjud so'rovlar olib boriladigan ma'lumotlar bazasi tanlanadi. Bizning ma'lumotlar db\_gurux ma'lumotlar bazasida saqlanmoqda. MySQL ma'lumotlar bazasidan foydalanish uchun avval db\_gurux ma'lumotlar bazasi tanlanadi (aktivlashtiriladi).

PHP da buning uchun mysql\_select\_db funksiyasi ishlatiladi.

mysql\_select\_db() funksiyasi

Sintaksisi:

int mysql\_select\_db (string ma'lumotlar\_bazasi\_nomi [, int bog'lanish\_identifikatori])

ma'lumotlar\_bazasi\_nomi parametrida tanlanishi lozim bo'lgan ma'lumotlar bazasi ko'rsatiladi, bog'lanish identifikatori parametri­da esa bir nechta ochiq bog'lanishlardan ma'lumotlar\_bazasi\_nomi nomli bazaning bog'lanish identifikatori ko'rsatiladi.

Bu funksiya ma'lumotlar bazasini muvaffaqiyatli tanlaganda TRUE ni, aks holda FALSE ni qaytaradi.

11.6-misol. db\_gurux ma'lumotlar bazasini tanlaymiz:

|  |  |  |
| --- | --- | --- |
|  | <?php  @mysql\_connect(“localhost”, “goyipov”, “12345”)  or die (“Bog'lanish o'rnatilmadi”);  echo "Bog'lanish ocrnatildi<br>";  @mysqi\_select\_db(“db\_gurux”)  or die (“db\_gurux ma'lumotlar bazasi tanlanmadi!”);  ?> | |
| Ma'lumotlar bazasiga bog'lana olmaganda yoki ma'lumotlar ba­zasini tanlamaganda brauzerda quyidagicha xabarlar chiqaziladi. | | |
|  | Natija: Bog'lanish o'rnatildi  db\_gurux ma'lumotlar bazasi tanlanmadi! |  |

Agar dasturda faqatgina bitta ma'lumotlar bazasi tanlanayotgan bo'lsa, uning identifikatorini saqlash shart emas. Lekin bir nechta ma'lumotlar bazasini tanlashda ularning qaytaradigan identifikator-lari saqlanishi kerak. Chunki, keyinchalik kerakli bazaga so'rovlarni qayta ishlash uchun jo'natish imkoniyati mavjud bo'lsin. Agar identifikator ko'rsatilmagan bo'lsa, oxirgi tanlangan ma'lumotlar bazasi ishlatiladi.

mysql\_close() funksiyasi

mysql\_close() funksiyasi shart bo'lmagan bog'lanish\_identifikatori parametri bilan bog'lanishni yopadi. Agar parametr berilmagan boisa, mysql\_close() funksiyasi oxirgi ochiq bog'lanishni yopadi.

Sintaksisi:

int mysql\_close ([int bog'lanish\_identifikatori]) mysql\_pconnect() funksiyasi yordamida o'rnatilgan ochiq bog'lanishni yopish shart emas.

|  |  |
| --- | --- |
|  | ***MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!*** |

* 1. Ma’lumotlar bazasi nima?
  2. Ma’lumotlar bazasining web texnologiyasida ahamiyati qanday?
  3. MySQL qanday vazifa bajaradi?
  4. PHPning MySQL bilan integrsiayasi qanday?
  5. PHPning ma’lumotlar bazasi biln ishlash funksiyalarni ayting