

MySQL

Violeta Chis



Ce este MySQL?

- un SGBD
- este un server care oferă acces multi-user la o serie de baze de date
- litera “ M ” in “*AMP” (Apache, MySQL , PHP)
 - LAMP - Linux
 - SAMP – Solaris
 - WAMP – Windows
 - XAMP

Ce este MySQL?

- Serverul de baze de date MySQL este foarte rapid, fiabil și ușor de utilizat.
- Deși în curs de dezvoltare constantă, MySQL Server oferă astăzi un set bogat și util de funcții.
- Conectivitatea, viteza și securitatea face serverul MySQL foarte potrivit pentru accesarea bazelor de date de pe Internet.

MySQL

este un RDBMS rapid, ușor de utilizat, personalizabil, program foarte puternic în sine

acceptă baze de date mari, de până la 50 de milioane de rânduri sau mai multe

funcționează pe mai multe SO și cu mai multe limbaje (PHP, PERL, C, C ++, JAVA etc.)

funcționează foarte repede și funcționează bine chiar și cu seturi mari de date

este un open-source

MySQL

- MySQL Server folosește tehnologia client/server
- Software-ul de baze de date MySQL este un sistem client/server care constă dintr-un server SQL multi-thread care acceptă back-end-uri diferite, mai multe programe și biblioteci client diferite, instrumente administrative și o gamă largă de interfețe de programare a aplicațiilor (API).

MySQL versus Oracle

Feature	MySQL	Oracle
Standard Heap Tables, B-Tree Indexes, Index-Organized Tables	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Partitioned Tables/Indexes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ACID Transaction Support	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Row-Level Locking, MVCC (readers don't block writers)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Server-Enforced Referential Integrity	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Advanced Indexing (Clustered, Full-Text)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Robust datatype support (BLOB's, varchar, datetime, numerics, etc.)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Special Web-style result-set data cache (Oracle 11g only)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Replication (statement and row-based)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Stored Procedures, Triggers, Functions, Cursors, Updateable Views	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Highly-Available Clustered Database	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Cost-based Optimizer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Backup with Point-in-Time Recovery	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Terabyte Database Size Capable	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Open Source	<input checked="" type="checkbox"/>	

Definirea și proiectarea bazei de date

Dialectul MySQL este, în mod similar cu celelalte dialecte SQL, împărțit în 2 seturi de instrucțiuni

- **definirea datelor**
- **manipularea datelor.**

Definirea datelor se face, prin intermediul limbajului de definire a datelor **DDL (Data Definition Language)**.

Acest sub-set include instrucțiuni pentru:

- crearea schemei unei baze de date;
- definirea structurilor logice și fizice;
- adăugarea și ștergerea tabelelor la schema bazelor de date;
- adăugarea și ștergerea câmpurilor într-un tabel preexistent;
- optimizarea bazelor de date (indexare);
- introducerea unor restricții de integritate;
- introducerea unor restricții la utilizarea bazelor de date;
- ștergerea bazelor de date.

MySQL

Pentru a administra bazele de date MySQL se poate folosi

- modul linie de comandă
- Aplicația gratuită numita **PHPMyAdmin** scrisă în PHP
- [MySQL GUI Tools](#) este o suită compusă din 3 aplicații cu interfață grafică pentru administrarea MySQL (MySQL Administrator, MySQL Query Browser, MySQL Migration Toolkit).



```
%mysql
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 49 to server version: 3.23.51

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

phpMyAdmin - pachet de scripturi php pt gestion. b.d.folosind o interfață web.



localhost >> localhost | phpMyAdmin 2.6.2-p11 - Design and Fun in .Ro Style

File Edit View Favorites Tools Help

Back Search Favorites

Address http://localhost/phpmyadmin/ Go Links

Bine ai venit la phpMyAdmin 2.6.2-p11

MySQL 4.1.9-max ruland pe localhost ca root@localhost

MySQL

- Creaza baza de date noua Gestionare
- Afiseaza informatiile runtime MySQL
- Afiseaza variabilele de sistem MySQL
- Afiseaza procesele
- Set de caractere si gestionarea acestora
- Mo:oare de stocare
- Reincarcare MySQL
- Drepturi de acces
- Baze de date
- Export

phpMyAdmin

- Language: Romanian (ro-utf-8)
- Setul de caractere MySQL: UTF-8 Unicode (utf8)
- MySQL connection collation: utf8_general_ci
- Sablon / Stil: Original
- [Documentatie phpMyAdmin](#)
- [Arata informatia PHP](#)
- [Pagina oficiala phpMyAdmin](#)
- [\[ChangeLog\]](#) [\[CVS\]](#) [\[Lists\]](#)

Baza de date: (Baze de date) ...

Selectati baza de date

MySQL Query Browser - Connection: root@localhost:3306 / magazin

File Edit View Query Script Tools Window Help

Transaction Explain Compare

Resultset 1

SQL Query Area

```
1 describe produse;
```

Field	Type	Null
idProdus	int(10) unsigned	NO
numeProdus	varchar(100)	NO
descriere	text	NO
pret	decimal(10,2)	NO
tipProdus	enum('digital','marfa','serviciu')	NO
idCategorie	int(11)	NO

Schemata Bookmarks History

- information_schema
- magazin
 - categorii
 - produse
 - idProdus
 - numeProdus
 - descriere
 - pret
 - tipProdus
 - idCategorie
- mysql

Syntax Functions Params Trx

- Data Definition Statements
- Data Manipulation Statements
- MySQL Utility Statements
- MySQL Transactional and Locking ...
- Database Administration Statements
- Replication Statements
- SQL Syntax for Prepared Statements

6 rows fetched in 0,0410s (0,003: Edit Apply Changes Discard Changes First Last Search

1: 1

Tipuri de tabele MySQL

- MySQL suportă două categorii de tabele:
 1. **tabele securizate (InnoDB și BDB)**
 2. **nesecurizate (HEAP, ISAM, MERGE și MyISAM).**
- La crearea unei tabele trebuie specificat tipul acesteia.
- Dacă nu se specifică tipul tabelei, va fi creată o tabelă de tipul InnoDB.
- MySQL va crea întotdeauna un fișier cu extensia „.frm” pentru a păstra structura tabelei. Indecșii și datele vor fi stocate în alte fișiere, în funcție de tipul tabelei.

Elemente de bază ale SQL

În funcție de efectul lor instrucțiunile SQL pot fi împărțite în 3 categorii

- operațiuni de **manipulare a datelor DML (Data Manipulation Language)**. -- referitoare la introducerea, actualizarea, stergerea și căutarea datelor:
 - **INSERT**,
 - **UPDATE**,
 - **DELETE**,
 - **SELECT**
- operațiuni de **schimbare a structurii datelor** - încadrate în **DDL (Data Definition Language)**. Instrucțiunile de bază din această categorie sunt:
 - **CREATE**,
 - **ALTER**,
 - **RENAME**,
 - **TRUNCATE**,
 - **DROP**

și se referă la schimbarea structurii unui tabel (modificarea coloanelor sau numărului de coloane, adăugarea de indecși) sau la schimbarea bazei de date
- **administrarea privilegiilor** asupra bazei de date. Această categorie din limbaj este denumită **DCL (Data Control Language)**, iar instrucțiunile de bază sunt:
 - **GRANT** (pentru acordarea de drepturi)
 - **REVOKE** (pentru stergerea drepturilor)

- Intr-o sesiune de lucru (sau când execut instrucțiuni sql din PHP) vom selecta baza de date la care se lucrează, altfel numele tabelor vor fi precedate de numele bazei de date.
- Fiecare instrucțiune sql se termina cu punct și virgulă (;).
- Comentariile în MySQL pot fi:
 - comentarii de o linie - introduse de caracterul # sau –
 - comentarii de mai multe linii, între /* și *
- Valorile ce au ca tip de date un șir de caractere sau dată calendaristică, trebuie incluse între ghilimele simple ' ' sau ghilimele duble " ".
- Valoarea NULL indică lipsa unei valori în câmpul respectiv. Ea este diferită fata de 0 (zero) pentru numere, sau șirul gol ("") pentru șiruri de caractere. Se poate specifica pentru anumite coloane faptul că nu acceptă această valoare prin atributul NOT NULL.

Tipuri de date

MySQL folosește multe tipuri de date diferite împărțite în trei categorii

1. Numeric
2. Dată și oră
3. Tipuri de șiruri.

1. Tipuri NUMERICE

- Toate tipurile numerice permit definirea lor cu semn sau fără semn, folosind atributele **signed** și **unsigned**. În mod implicit sunt definite cu semn.
- O altă opțiune pentru tipurile numerice este **zerofill**. Aceasta determină complectarea cu zero-uri în partea stângă a unei valori numerice, în cazul în care aceasta nu acoperă dimensiunea rezervată pentru tipul numeric stabilit.

Tipuri NUMERICE

INT - număr întreg de dimensiuni **normale** care poate fi cu semn (intervalul admis este de la -2147483648 la 2147483647) sau fără semn (intervalul admis este de la 0 la 4294967295). Puteți specifica o lățime de până la 11 cifre.

TINYINT - număr întreg **foarte** mic care poate fi cu semn (intervalul admis este de la -128 la 127) sau fără semn (intervalul admis este de la 0 la 255). Puteți specifica o lățime de până la 4 cifre.

SMALLINT - număr întreg **mic** care poate fi cu semn (intervalul admis este de la -32768 la 32767 sau fără semn (intervalul admis este de la 0 la 65535). Puteți specifica o lățime de până la 5 cifre.

MEDIUMINT - număr întreg de dimensiuni **medii** care poate fi cu semn (intervalul admis este de la -8388608 la 8388607) sau fără semn (intervalul admis este de la 0 la 16777215). Puteți specifica o lățime de până la 9 cifre.

BIGINT - număr întreg **mare** care poate fi cu semn (intervalul admis este de la -9223372036854775808 până la 9223372036854775807) sau fără semn (intervalul admis este de la 0 la 18446744073709551615). Puteți specifica o lățime de până la 20 de cifre.

Tipuri NUMERICE

FLOAT (M, D) - număr cu virgulă flotantă, nu poate fi folosit fără semn. Puteți defini numărul total de cifre (M) și numărul de zecimale (D), val implicite (10,2). Pentru zecimale putem avea valori ≤ 24

DOUBLE (M, D) – permite valori cu zecimale, cu precizie dublă. Nu poate fi folosit fără semn. Val implicită (16,4).

DECIMAL (M, D) - nr real cu virg mobilă despachetat, fiecare zecimală corespunde unui octet

Tipuri calendaristice

DATA - O dată în format AAAA-MM-DD, între 1000-01-01 și 9999-12-31. De exemplu, 30 decembrie 1973 va fi păstrat ca 1973-12-30.

DATETIME - O combinație de date și ora în format AAAA-MM-DH HH: MM: SS, între 1000-01-01 00:00:00 și 9999-12-31 23:59:59. De exemplu, ora 3:30 după-amiaza din 30 decembrie 1973 va fi păstrată în perioada 1973-12-30 15:30:00.

TIMESTAMP - similar cu DATETIME, însă are un interval de valori mai mic (poate ține date între 1970-01-01 00:00:01 și 2038-01-09 03:14:07). Diferența mai importantă poate fi că are capacitatea de actualizare automată la operații de INSERT și UPDATE.

TIME - Stocază ora într-un format HH: MM: SS.

YEAR (M) - Stocază un an într-un format de 2 cifre sau 4 cifre. Dacă lungimea este specificată ca 2 (de exemplu YEAR (2)), ANUL poate fi între 1970 și 2069 (70 - 69). Dacă lungimea este specificată ca 4, atunci ANUL poate fi între 1901 și 2155. Lungimea implicită este 4.

Tipuri șir de caractere

CHAR (M) - Un șir de lungime fixă între 1 și 255 de caractere în lungime. Definirea unei lungimi nu este necesară, dar implicit este 1.

VARCHAR (M) - Un șir de lungime variabilă între 1 și 255 de caractere. Trebuie să definiți o lungime atunci când creați un câmp VARCHAR.

BLOB sau **TEXT** - Un câmp cu o lungime maximă de 65535 de caractere. BLOB-urile sunt „Obiecte mari binare” și sunt utilizate pentru a stoca cantități mari de date binare, cum ar fi imagini sau alte tipuri de fișiere. Câmpurile definite ca TEXT conțin, de asemenea, cantități mari de date. Diferența dintre cele două este că sortările și comparațiile de date stocate sunt sensibile cu majuscule între literele BLOB și nu sunt sensibile la litere mari și minuscule în câmpurile TEXT. Nu specificați o lungime cu BLOB sau TEXT.

TINYBLOB sau **TINYTEXT** - o coloană BLOB sau TEXT cu o lungime maximă de 255 de caractere. Nu specificați o lungime cu TINYBLOB sau TINYTEXT.

Tipuri șir de caractere

MEDIUMBLOB sau **MEDIUMTEXT** - o coloană BLOB sau TEXT cu lungimea maximă de 16777215 caractere. Nu trebuie specificată lungimea.

LOB sau **LONGTEXT** - o coloană BLOB sau TEXT cu o lungime maximă de 4294967295 caractere. Nu trebuie specificată lungimea.

ENUM - o enumerare. Când definiți un ENUM, creați o listă de elemente din care trebuie selectată valoarea (sau poate fi NULL). De exemplu, dacă ai dori ca câmpul tău să conțină „A” sau „B” sau „C”, ai defini ENUM-ul tău ca ENUM („A”, „B”, „C”) și numai acele valori (sau NULL) ar putea vreodată să populeze acel câmp.

Atribute ale unei coloane

AUTO_INCREMENT poate fi adăugat unei coloane de tip numeric

NULL - indică lipsa informației în câmpul respectiv.

NOT NULL - dacă nu vreau să accept valori nule pentru coloana respectivă. In acest caz, dacă tot nu precizez o valoare pentru câmpul respectiv la operațiunea de INSERT, se introduce valoarea implicită, specificată cu atributul **DEFAULT**

DEFAULT- în lipsa unei valori la INSERT pentru coloana respectivă, se memorează automat valoarea ce urmează după atributul **DEFAULT**

UNSIGNED (fără semn) ce arată faptul că se stochează doar valori pozitive (și pentru întregi se dublează valoarea maximă stocată).

ZEROFILL completează cu o un număr astfel încât să ajungă la x cifre: dacă atribuim coloanei de exemplu **INT(x)**.

CHARACTER_SET Pentru coloane de tip șir de caractere (**CHAR**, **VARCHAR**, **TEXT**, etc) se poate specifica un set de caractere (ex: latin1 sau utf8)

COLLATE specifica ceea ce se numeste "collation", termen ce desemnează modul în care sunt comparate și ordonate caracterele dintr-un set de caractere.

Crearea bazei de date

Sintaxa

create database [if not exists] name

Clauza **if not exists** inhibă afișarea unui mesaj de eroare în cazul în care în sistem există o altă bază de date cu același nume.

```
mysql> create database ex1;  
Query OK, 1 row affected (0.00 sec)  
  
mysql>
```

Comanda **use** stabilește baza de date pentru care se vor executa interogările ulterioare.

Ștergerea unei baze de date

Sintaxa

```
drop database [if exists] nume_baza_date;
```


Comanda Create Table

- Nume tabel
- Nr câmpuri
- Definiții pentru fiecare câmp

Sintaxa

```
CREATE TABLE table_name (column_name  
column_type);
```

Comanda Create Table

```
create table tutorials_tbl(  
    cursId INT NOT NULL AUTO_INCREMENT,  
    denumire VARCHAR(100) NOT NULL,  
    autor VARCHAR(40) NOT NULL,  
    data DATE,  
    PRIMARY KEY (cursId)  
);
```

```
CREATE TABLE Personal
(
Nume varchar(30),
prenume varchar(30),
virsta int
);
```

```
mysql> show columns from personal;
```

Field	Type	Null	Key	Default	Extra
nume	varchar(30)	YES		NULL	
prenume	varchar(30)	YES		NULL	
virsta	int(11)	YES		NULL	

3 rows in set (0.01 sec)

```
mysql>
```

Comanda Insert

Sintaxa

```
INSERT INTO table_name ( field1, field2,...fieldN )  
VALUES  
( value1, value2,...valueN );
```

- Pentru a insera tipuri de date șir, este necesar să păstrați toate valorile în ghilimele duble sau simple. De exemplu „valoare”.

Comanda Insert

Ex inserarea datelor:

```
root@host# mysql -u root -p password;
Enter password:*****
mysql> use TUTORIALS;
Database changed

mysql> INSERT INTO tutorials_tbl
->(tutorial_title, tutorial_author, submission_date)
->VALUES
->("Learn PHP", "John Poul", NOW());
Query OK, 1 row affected (0.01 sec)
```

Comanda Insert

```
INSERT INTO dvd SET (nume_film = "Lost",  
    pret="25.00", descriere="este un film foarte  
    bun");
```

```
INSERT INTO dvd (nume_film, pret, descriere)  
    VALUES ("Lost", "25.00", "este un film foarte  
    bun");
```

Comanda SELECT

- este utilizată pentru a selecta datele din baza de date MySQL
- *Sintaxa*

```
SELECT field1, field2,...fieldN  
FROM table_name1, table_name2...  
[WHERE Clause]  
[OFFSET M ][LIMIT N]
```

Comanda SELECT

- Puteți utiliza una sau mai multe tabele separate prin virgulă pentru a include diferite condiții folosind o clauză WHERE, dar clauza WHERE este o parte opțională a comenzii SELECT.
- Puteți obține unul sau mai multe câmpuri într-o singură comandă SELECT.
- Puteți specifica stea (*) în locul câmpurilor. În acest caz, SELECT va returna toate câmpurile.
- Puteți specifica orice condiție folosind clauza WHERE.
- Puteți specifica o compensare folosind OFFSET de unde SELECT va începe returnarea înregistrărilor. În mod implicit, compensarea începe de la zero.
- Puteți limita numărul de returnări folosind atributul LIMIT.

Comanda SELECT

```
mysql> select * from Personal;
+-----+-----+-----+
| Nume   | prenume | virsta |
+-----+-----+-----+
| Creanga | Ion     | 42     |
+-----+-----+-----+
| Eminescu | Mihai  | 28     |
+-----+-----+-----+
1 row in set (0.00 sec)
```