

SQL, MYSQL, PHPMYADMIN

SQL – Structured Query Language

Termenul de **baza de date** apare in 1967 si reprezinta o colectie de informatii, relatiile logice dintre aceste informatii si tehnicile de prelucrare corespunzatoare (sortare, regasire, apreciere, stergere, adaugare, inserare, modificare).

Vom crea o baza de date:

```
CREATE DATABASE curs ;
```

Selectam baza de date:

```
USE curs;
```

Vom crea o tabela simpla **studenti**, care va avea trei campuri: **nume**, **prenume**, **varsta**.

Campurile **nume** si **prenume** sunt de tip sir de caractere, iar **varsta** este de tip numar.

```
CREATE TABLE studenti (  
    nume VARCHAR( 64 ) ,  
    prenume VARCHAR( 64 ) ,  
    varsta INT ) ;
```

Vom introduce 4 studenti in tabela studenti baza de date:

```
INSERT INTO studenti (  
    nume,  
    prenume,  
    varsta  
)  
VALUES  
    ('Popescu', 'Ion', 25),  
    ('Costea', 'Dorel', 26),  
    ('Popa', 'Marin', 25)  
);
```

Vom folosi comanda **SELECT** pentru a extrage toti studentii din tabela **studenti**. Caracterul * semnifica faptul ca dorim sa extragem toate campurile din tabela.

```
SELECT * FROM studenti
```

| | | |
|---------|-------|----|
| Popescu | Ion | 25 |
| Costea | Dorel | 26 |
| Popa | Marin | 25 |

Comanda **SELECT** accepta o serie de parametri, cu care se poate controla in detaliu:

1. ce campuri din tabela sunt citite (de exemplu, doar nume si prenume, fara varsta)
2. ordinea in care sunt preluate campurile (de exemplu, in ordine alfabetica)
3. care randuri sunt citite din tabela (de exemplu, doar studentii cu varsta<26 ani)
4. cate campuri sunt citite (de exemplu, doar primele 100 de inregistrari)
5. cum sunt afisate campurile citite (campurile de data se pot afisa in diverse formate)

Vom afisa doar campurile **nume** si **prenume**:

```
SELECT nume, prenume FROM studenti
```

| | |
|---------|-------|
| Popescu | Ion |
| Costea | Dorel |
| Popa | Marin |

Vom afisa inregistrările in ordine alfabetica dupa nume si prenume

```
SELECT nume, prenume FROM studenti ORDER BY nume, prenume
```

| | |
|---------|-------|
| Costea | Dorel |
| Popa | Marin |
| Popescu | Ion |

Vom afisa din tabela doar studentii cu varsta mai mica de 26 de ani, folosind clauza **WHERE**

```
SELECT * FROM studenti WHERE varsta <26
```

| | | |
|---------|-------|----|
| Popescu | Ion | 25 |
| Popa | Marin | 25 |

Vom afisa primii 3 studenti, ordonati dupa varsta

```
SELECT * FROM studenti ORDER BY varsta LIMIT 3
```

| | | |
|---------|-------|----|
| Popescu | Ion | 25 |
| Popa | Marin | 25 |
| Costea | Dorel | 26 |

Vom afisa toti studentii cu litere mari folosind una din numeroasele functii MySQL, **UPPER()**.

Aceasta converteste campurile text date ca parametrii in litere mari:

```
SELECT UPPER( nume ) , UPPER( prenume ) , varsta FROM student
```

| | | |
|---------|-------|----|
| POPESCU | ION | 25 |
| COSTEA | DOREL | 26 |
| POPA | MARIN | 25 |

Pentru actualizarea datelor, vom folosi comanda **UPDATE**. Este important ca o comanda **UPDATE** sa fie insotita de o conditie **WHERE**, in caz contrar fiind afectate toate inregistrarile din tabela.

```
UPDATE studenti SET nume = 'Dima', prenume = 'Ionel', varsta = 27,  
WHERE nume = 'Costea' AND prenume = 'Dorel'
```

Comanda are urmatorul rezultat:

| | | |
|------|-------|----|
| Dima | Ionel | 27 |
| Popa | Marin | 25 |
| Doru | Ionel | 27 |

Pentru stergerea datelor folosim comanda **DELETE**. Apelata fara parametri conditionali, adica o clauza **WHERE**, va sterge TOATE inregistrarile din tabela:

```
DELETE * FROM studenti
```

Pentru a sterge de exemplu, doar intrarea corespunzatoare studentului "Ion Popescu", scriem:

```
DELETE from studenti where nume='popescu' AND prenume='ion'
```

Dupa operatia de stergere, tabela va contine urmatoarele inregistrari:

| | | |
|--------|-------|----|
| Costea | Dorel | 26 |
| Popa | Marin | 25 |

MySQL

MySQL este un sistem de gestiune a bazelor de date, foarte rapid si robust. MySQL este un server multi-user (suporta mai multi utilizatori) si multi-thread (mai multe fire de executie). Limbajul de interogare folosit este SQL, care este limbajul standard de interogare a bazelor de date. Este disponibil in mod public din 1996, iar in clipa de fata MySQL este disponibil sub o licenta Open Source, dar daca este nevoie sunt disponibile si licente comerciale.

MySQL se poate instala odata cu un pachet complet pentru web development (ex. XAMPP) sau poate fi instalat separat. Ca popularitate in domeniul bazelor de date pentru web, MySQL se situeaza pe primul loc in lume(peste 11 milioane de instalari).

Functii PHP specifice comunicarii cu MySQL

mysql_connect - realizeaza o conexiune cu baza de date

mysql_close – inchide o conexiune

mysql_create_db – creeaza o baza de date

mysql_select_db – selecteaza o baza de date

mysql_drop_db – sterge o baza de date

mysql_query – executa o comanda SQL

mysql_fetch_object – extrage un row dintr-un rezultat sub forma de obiect

mysql_fetch_array – extrage un row dintr-un rezultat sub forma de obiect

mysql_insert_id – returneaza ID-ul AUTO_INCREMENT al ultimei valori inserate

PhpMyAdmin

PhpMyAdmin este un utilitar open source scris in PHP, si care este folosit pentru administrarea bazelor de date MySQL din cadrul unui browser. PhpMyAdmin poate fi folosit pentru crearea si stergerea bazelor de date, tabele, poate adauga/edita/sterge campuri, poate executa interactiv comenzi SQL, poate gestiona utilizatori si permisi, etc. PhpMyAdmin poate fi accesat local la adresa :

<http://localhost/phpmyadmin/>

OBS – PhpMyAdmin ruleaza implicit cu urmatoarele setari pentru conectarea la serverul mysql: user=root, parola vida, care sunt setarile implicite in instalarea unui pachet **XAMPP** sau **EasyPHP**. Daca vedeti mesajul *access denied*, poate fi necesara editarea fisierului phpmyadmin/config.inc.php si modificarea datelor implicite de conectare.