

# **Baze de date**

Facultatea de Electronica si Telecomunicatii

**Sl. Dr. Ing. Marian Constantin Bucos**

## Cuprins:

1. Tehnologii Web .....	4
1.1. Aplicatii Web dinamice .....	4
1.2. Arhitectura .....	4
1.3. Servere Web.....	5
1.4. Baze de date pentru Web .....	6
1.5. Limbaje de scripting pentru Web.....	7
2. Serverul de baza de date MySQL .....	10
2.1. Caracteristici generale.....	10
2.2. Clienti MySQL.....	10
2.3. Conectarea la serverul MySQL folosind clientul linie de comanda .....	11
2.4. Interogari de stare .....	12
2.5. Rularea unui fisier extern de comenzi SQL.....	12
2.6. Sistemul de privilegii al MySQL .....	13
2.7. Crearea utilizatorilor si stabilirea privilegiilor.....	14
3. Baze de date. Design si implementare .....	15
3.1. Notiuni introductive.....	15
3.2. Proiectarea unei baze de date .....	16
3.3. Crearea bazei de date in MySQL .....	17
3.4. Tabele.....	17
3.5. Tipuri de date, operatori si functii.....	17
3.5.1. Tipuri de date .....	17
3.5.2. Constante, identificatori, comentarii.....	18
3.5.3. Operatori .....	18
3.5.4. Functii – utilizare functii de biblioteca .....	18
4. Gestionarea datelor folosind interogari SQL .....	23
4.1. Adaugarea inregistrarilor in tabele.....	23
4.2. Modificarea inregistrarilor din tabele .....	24

4.3. Stergerea inregistrarilor din tabele.....	24
4.4. Interogarea datelor .....	24
5. Limbajul de scripting PHP.....	29
5.1. Notiuni introductive.....	29
5.2. Inserarea codului PHP intr-o pagina HTML.....	29
5.3. Generarea dinamica a continutului .....	29
5.4. Tipuri de date .....	30
5.5. Identificatori, constante, variabile.....	30
5.6. Extragerea datelor din formularele HTML.....	31
5.7. Operatori, expresii.....	32
5.8. Instructiuni .....	32
5.8.1. Instructiuni conditionale .....	32
5.8.2. Instructiuni repetitive .....	33
5.9. Tablouri.....	35
5.10. Functii .....	36
6. Accesarea serverului MySQL utilizand PHP.....	38
6.1. Conectarea la serverul de baza de date .....	38
6.2. Rularea inerogarilor .....	39
6.3. Obtinerea si afisarea datelor .....	39
6.4. Adaugarea datelor din pagina Web in baza de date.....	42
6.5. Stergerea datelor prin intermediul unei pagini Web .....	43
6.6. Modificare datelor in interfata Web.....	44

# 1. Tehnologii Web

## 1.1. Aplicatii Web dinamice

Aplicatiile Web dinamice reprezinta in general site-uri Web complexe, interactive, care utilizeaza facilitatile oferite de limbajele de scripting (PHP, ASP, JSP, ColdFusion, Perl) si bazele de date (MySQL, MsSQL, PostgreSQL, Access).

Cele mai intalnite aplicatii Web dinamice sunt - portalurile, formularele, magazinele virtuale, forumurile.

Dezvoltarea de aplicatii Web necesita existenta urmatoarelor tehnologii:

- Server Web – soft care permite stocarea si livrarea aplicatiilor Web in exterior (Internet);
- Sistem de Gestiune Baze de Date (SGBD) – care va stoca informatia pentru aplicatiile Web;
- Limbaje de scripting – folosite in scrierea programelor (scripturilor) care ofera continutul dinamic pentru aplicatiile Web.

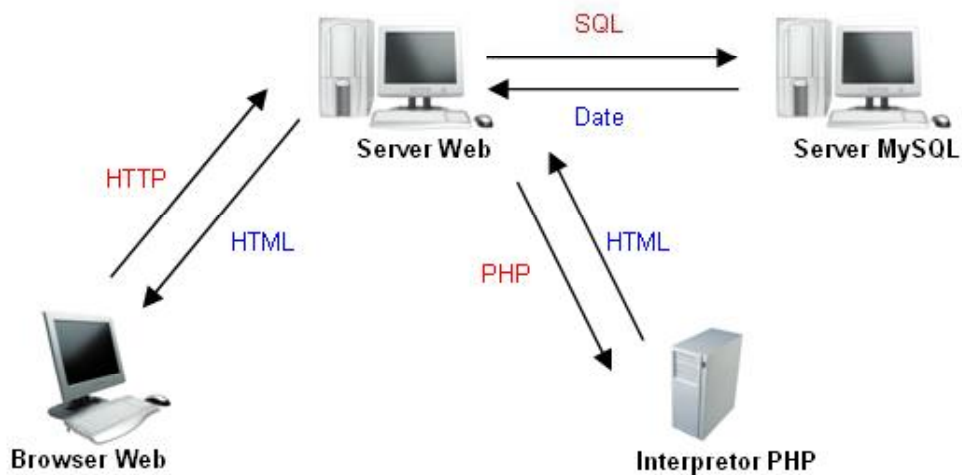
## 1.2. Arhitectura

Aplicatiile dinamice presupun existenta unei arhitecturi formata din:

- browser Web;
- Internet;
- server Web;
- server de baze de date;
- interpretor PHP.

In momentul in care utilizatorul acceseaza o pagina Web dinamica, prin intermediul unui browser Web, se transmite o cerere HTTP catre serverul Web. Daca serverul Web identifica tipul paginii ca fiind .php, atunci pagina este trimisa interpretorului PHP. Dupa interpretare, rezulta o

pagina care contine cod HTML (si eventual JavaScript), care va fi returnata serverului Web. Sripturile PHP pot contine si cereri SQL, care vor fi adresate serverului de baze de date (mySQL).



Software-ul necesar pentru dezvoltarea de aplicatii Web dinamice va cuprinde: un server Web (Apache, IIS) un interpretor PHP, un editor de text (Notepad), un browser Web (IE, Firefox, Opera) si un server de baze de date (mySQL).

### 1.3. Servere Web

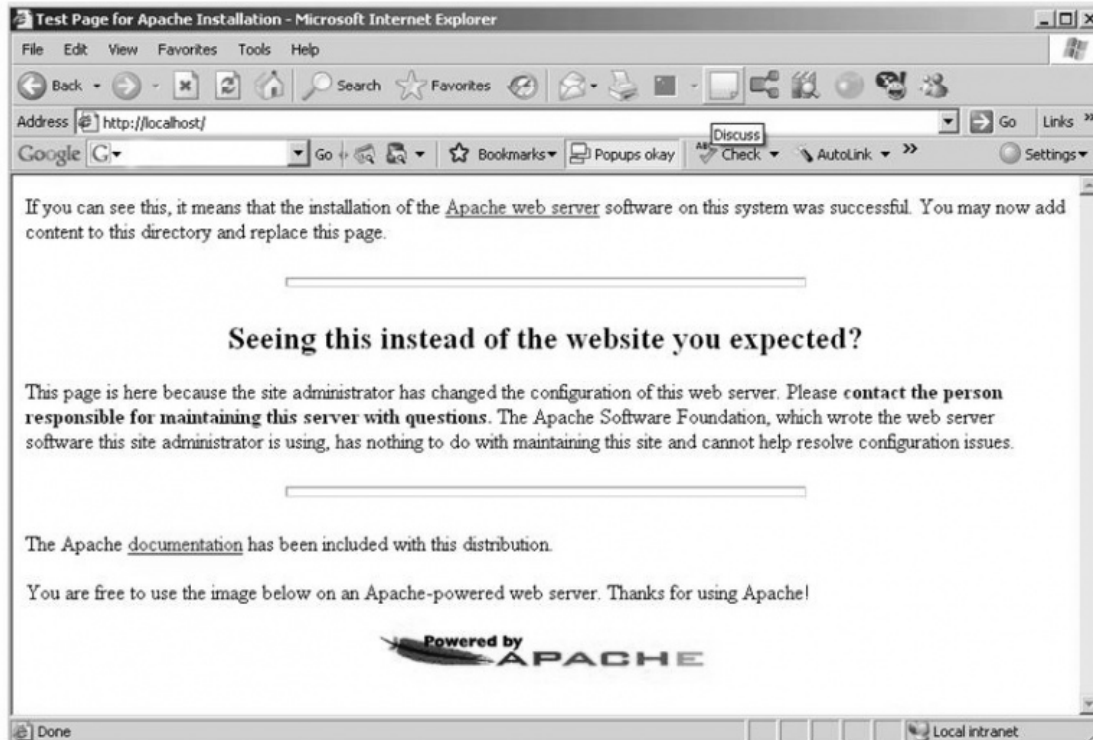
Server Web reprezita computerul si pachetul software inclus care ofera o categorie specifica de servicii unui client ce ruleaza pe un alt calculator. Serverul Web administreaza si ofera aplicatiile Web accesibile pe orice calculator conectat la Internet.

Cele mai utilizate servere Web in momentul de fata sunt:

- Apache Web Server – server Web Open Source dezvoltat de The Apache Software Foundation;
- Internet Information Services – server promovat de catre compania Microsoft.

Pentru instalarea serverului Apache avem nevoie mai intai de kit-ul de instalare, care poate fi descarcat de la adresa <http://www.apache.org> .

Se executa programul de instalare. Dupa instalare se creaza un set de fisiere de configurarea in folder-ul conf (prezinta interes fisierul httpd.conf) si folder-ul htdocs – locatie in care vor fi depozitate aplicatiile web.



Testarea serverului Apache se realizeaza in browser Web prin introducerea numelui calculatorului, sau a termenului localhost daca aplicatia este rulata local.

Pentru laboratorul de BDMM se vor utiliza urmatoarele versiuni de soft:

- Apache Web Server 2.2.8
- PHP 5.2.5
- MySQL 5.1.23

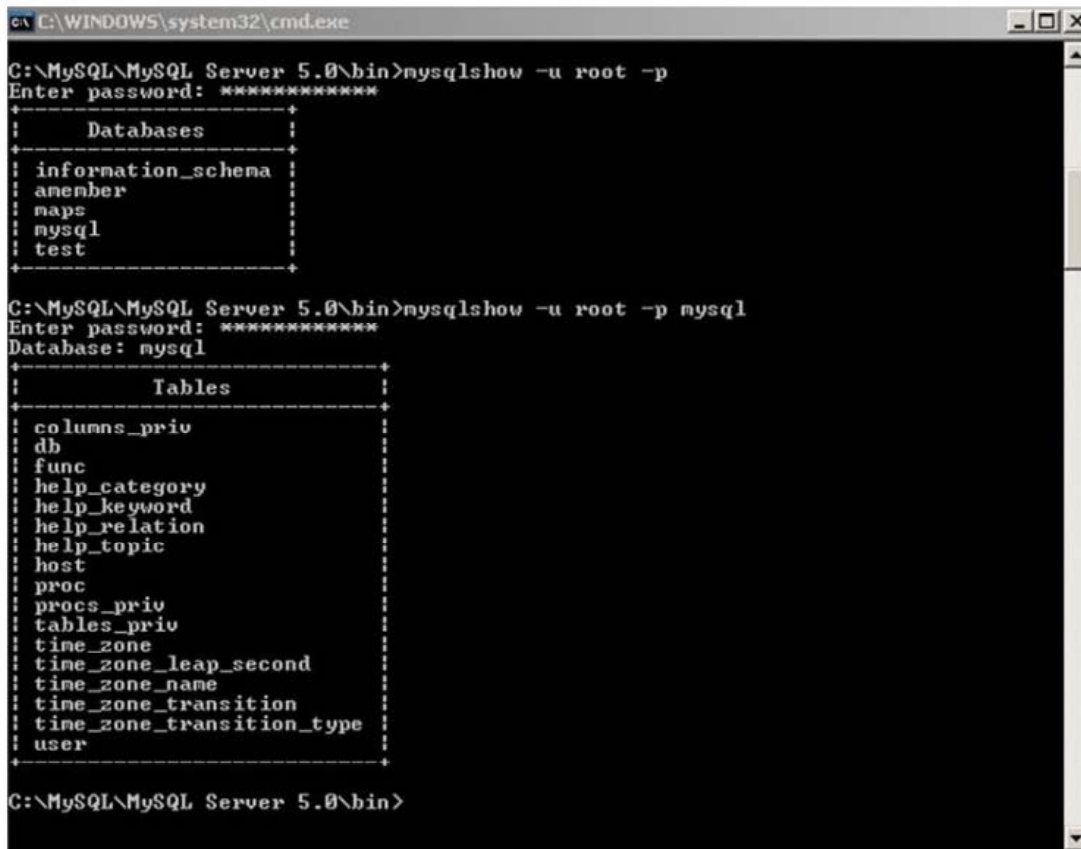
## 1.4. Baze de date pentru Web

Serverul de baze de date ( sau SGBD ) este responsabila pentru stocarea datelor, gasirea lor eficienta bazata pe interogari definite de client si pentru manipularea diferitelor nivele de securitate pentru aceste date.

MySQL este cel mai popular sistem de gestiune de baze de date in regim Open Source, destinat utilizarii cu PHP. Serverul de baze de date mySQL este un sistem de gestiune de baze de date relational construit pe modelul client-server ce foloseste limbajul SQL.

Testarea serverului mySQL se poate realiza prin urmatoarele comenzi:

- mysqlshow
- mysqlshow mysql



```
C:\WINDOWS\system32\cmd.exe
C:\MySQL\MySQL Server 5.0\bin>mysqlshow -u root -p
Enter password: *****
+-----+
| Databases |
+-----+
| information_schema |
| anenber |
| maps |
| mysql |
| test |
+-----+

C:\MySQL\MySQL Server 5.0\bin>mysqlshow -u root -p mysql
Enter password: *****
Database: mysql
+-----+
| Tables |
+-----+
| columns_priv |
| db |
| func |
| help_category |
| help_keyword |
| help_relation |
| help_topic |
| host |
| proc |
| procs_priv |
| tables_priv |
| time_zone |
| time_zone_leap_second |
| time_zone_name |
| time_zone_transition |
| time_zone_transition_type |
| user |
+-----+

C:\MySQL\MySQL Server 5.0\bin>
```

La instalarea serverului mySQL sunt create automat bazele de date mysql, care stocheaza informatii despre utilizatori si drepturi de acces, si test.

## 1.5. Limbaje de scripting pentru Web

Limbajele de scripting sunt utilizate pe Web pentru a oferi continut dinamic sau pentru a comunica cu servere de baze de date. Exista doua astfel de categorii de limbaje de scripting:

- server-side scripting – pentru dezvoltarea de scripturi ce ruleaza pe un server: PHP, ASP,

JSP, Perl, ColdFusion;

- client-side scripting – pentru crearea de scripturi ce ruleza intr-un browser Web: JavaScript, VBScript;

Pentru utilizarea PHP ca si modul al unui server Web este necesara instalarea sau copierea arhivei pentru PHP, si configurarea modulului si a serverului Web dupa cum urmeaza.

Configurarea serverului Web Apache consta in editarea fisierului httpd.conf, si presupune urmatoorii pasi pentru a recunoaste PHP ca si modul. Mai intai se adauga urmatoarele doua randuri in fisierul mai sus mentionat:

- LoadModule php5\_module "c:/php/php5apache2\_2.dll"
- AddType application/x-httpd-php .php
- DirectoryIndex index.php index.html index.html.var

In ceea ce priveste modulul PHP, acesta poate fi configurat prin editarea fisierului php.ini. Se poate stabili calea catre root-ul serverului Apache daca se editeaza linia:

```
doc_root = "c:\Apache Group\Apache2\htdocs"
```

Pentru versiunea 5 de PHP suportul pentru baze de date mySQL nu mai este activat din start. Daca PHP este utilizat cu baze de date mySQL atunci trebuie copiat fisierul libmysql.dll din c:\php in c:\windows\system si configurat fiserul php.ini:

```
extension_dir = "c:\php\ext"
```

,dupa care este sters caracterul ; din linia extension=php\_mysql.dll.

De fiecare data cand fisierul php.ini este editat, este necesara repornirea serverului Apache.

Pentru a verifica daca modulul PHP a fost instalat si configurat cu succes se poate apela la functia phpinfo().

PHP – Hypertext Preprocessor;

ASP – Active Server Pages;

JSP – Java Server Pages;



SQL – Structured Query Language;  
HTTP – HyperText Transport Protocol;  
HTML – Hypertext Markup Language;  
DBMS – Data Base Management System;  
IIS – Internet Information Services;

## **2. Serverul de baza de date MySQL**

### **2.1. Caracteristici generale**

Serverul de baze de date MySQL este un sistem de gestiune a bazelor de date relational construit pe modelul client-server ce foloseste limbajul SQL, este shareware, foarte rapid si usor de utilizat.

Cateva din facilitatile oferite de MySQL merita sa fie mentionate:

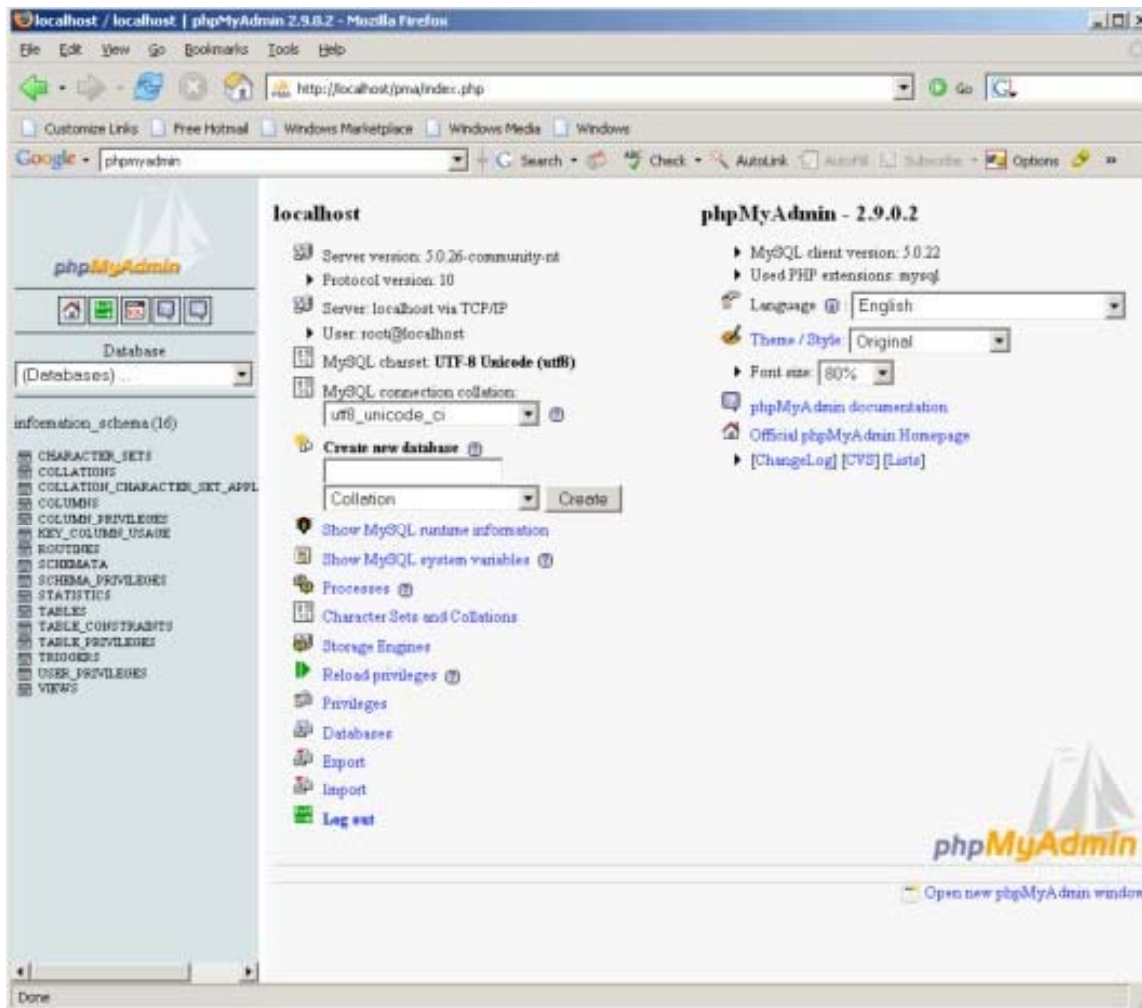
- este construit pentru a functiona ca multi-thread, multi-procesor;
- functioneaza pe o multitudine de sisteme de operare;
- suporta baze de date de mari dimensiuni;
- sistem de parole si privilegii foarte flexibil si sigur;
- accepta ca API C, C++, Java, Perl, PHP, Python, etc.

### **2.2. Clienti MySQL**

Conectarea la serverul de baze de date mySQL se realizeaza prin intermediul unor aplicatii denumite clienti mySQL. Clientii sunt aplicatii care acceseaza un serviciu la distanta prin intermediul unui calculator.

Serverul mySQL este livrat cu un astfel de client (mysql.exe), care poarta denumirea client tip linie de comanda. Acest client permite conectarea atat local cat si la un server la distanta.

Un alt client foarte popular este phpMyAdmin, care ofera posibilitatea administrarii bazelor de date intr-o interfata Web dezvoltata in PHP. Aplicatia poate fi descarcata de la adresa <http://www.phpmyadmin.net>.



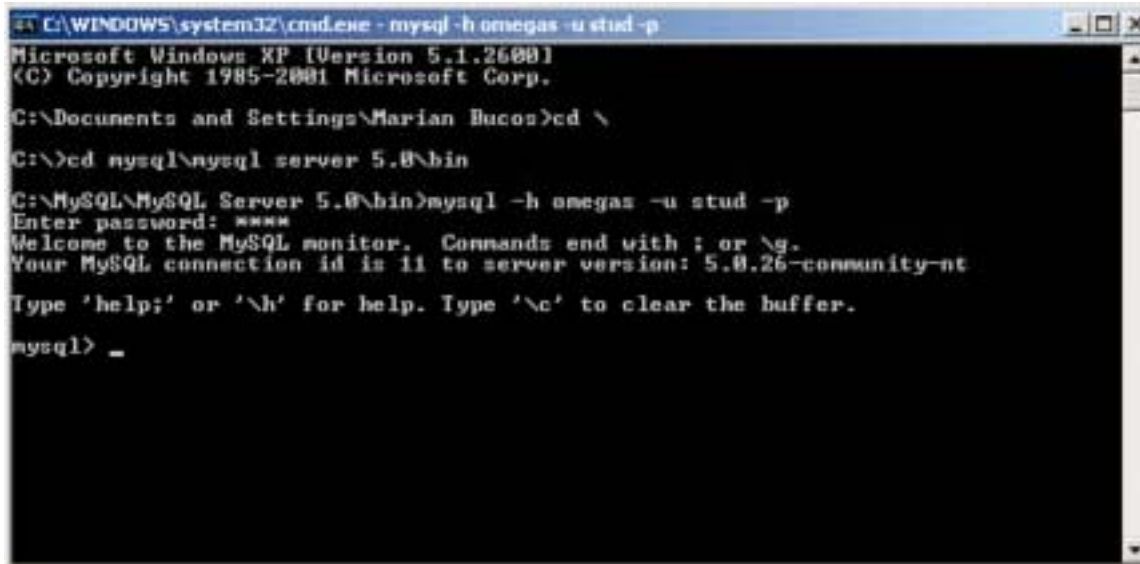
### 2.3. Conectarea la serverul MySQL folosind clientul linie de comanda

Clientul tip linie de comanda va fi apelat din subdirectorul bin al directorului de instalare, folosind urmatoarea comanda:

```
mysql -h nume_server -u nume_utilizator -p
```

,unde:

- \* h nume\_server este numele masinii pe care se gaseste instalat serverul;
- \* u nume\_utilizator este numele utilizatorului care acceseaza serverul;
- \* p activeaza citirea unei parole pentru contul de utilizator.



```
C:\WINDOWS\system32\cmd.exe - mysql -h omegas -u stud -p
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Marian Bucos>cd \
C:\>cd mysql\mysql server 5.0\bin
C:\MySQL\MySQL Server 5.0\bin>mysql -h omegas -u stud -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11 to server version: 5.0.26-community-nt
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> _
```

## 2.4. Interogari de stare

Dupa conectare pot fi rulate interogari care furnizeaza date despre starea serverului: versiunea programului, data curenta de pe server.

! Verificati versiune serverului de baza de date omegas, si data la un moment dat. `mysql> select version(), current_date;`

Obs. Interogari se pot despartii pe mai multe randuri si se incheie cu ; . Pot contine orice combinatie de caractere mari/mici, limbajul de interogare nefiind case sensitive.

Pentru a obtine informatii despre server (bazele de date de pe server, tabelele dintr-o baza de date, coloanele dintr-un tabel al unei baze de date) se pot rula interogari show.

! Din clientul linie de comanda se vor verifica urmatoarele interogari show:

```
mysql> show databases;
```

```
mysql> show tables from mysql;
```

```
mysql> show columns from mysql.user;
```

```
mysql> show columns from user from mysql;
```

## 2.5. Rularea unui fisier extern de comenzi SQL

Clientul mysql.exe se poate utiliza si pentru a rula un fisier text de comenzi SQL. In acest caz poate fi utilizata comanda:

```
mysql -h nume_server -u nume_utilizator -p < nume_fisier.sql
```

Deasemenea, fisierele de comenzi SQL pot fi rulate din linia de comanda mysql utilizand sintaxa:

```
mysql> source nume_fisier.sql
```

shareware – software distribuit gratuit (sau pentru o taxa simbolica), pentru care se aplica anumite reguli;

multi-thread – descrie un program care este proiectat pentru a avea parti ale codului executate concurent;

API – Application Programming Interface

## 2.6. Sistemul de privilegii al MySQL

Functia principala a sistemului de privilegii al MySQL este aceea de a autentifica si autoriza utilizatorii conectati la server.

Autorizarea se refera la permisiunea de a rula interogari precum SELECT, INSERT, UPDATE sau DELETE. O clasa aparte de privilegii se refera la drepturile de administrare si de interactiune cu sistemul de operare.

La instalarea sistemului este creat implicit un utilizator numit root care are toate drepturile activate. Acest utilizator trebuie folosit, din motive de securitate, doar pentru administrare.

Pentru fiecare utilizator care va interactiona cu serverul trebuie creat un utilizator.

<b>Privilegiu</b>	<b>Semnificatie</b>
select	permite selectarea(vizualizarea) datelor
insert	permite adaugarea de noi inregistrari
update	permite modificarea datelor
delete	permite stergerea inregistrarilor
index	permite crearea/stergerea indecsilor
alter	permite redenumirea sau modificarea structurii tablei

create	permite crearea unei baze de date/tabele
drop	permite stergerea unei baze de date/tabele
grant	permite delegarea privilegiilor catre alt utilizator
shutdown	permite oprirea serverului din programul curent
process	permite vizualizarea/oprirea proceselor in executie
file	permite schimbul de date intre tabele si fisiere

## 2.7. Crearea utilizatorilor si stabilirea privilegiilor

Comanda GRANT permite crearea utilizatorilor si/sau stabilirea, respectiv modificarea privilegiilor acestora pe patru nivele de privilegii:

- global: se aplica tuturor bazelor de date existente pe server;
- database: se aplica tuturor tabelelor dintr-o baza de date;
- table: se aplica tuturor coloanelor dintr-o tabela;
- column: se aplica doar coloanelor specificate explicit.

Forma generala a comenzii GRANT este:

```
GRANT privilegii ON componenta
TO nume_utilizator [IDENTIFIED BY 'parola'] [WITH GRANT OPTION]
```

Parola unui utilizator se poate modifica utilizand comanda:

```
SET PASSWORD FOR nume_utilizator = PASSWORD('noua_parola')
```

Inversul comenzii GRANT este comanda REVOKE. Ea este utilizata pentru a retrage drepturile de la un utilizator. Sintaxa sa este asemanatoare cu cea a comenzii GRANT:

```
REVOKE privilegii ON componenta FROM nume_utilizator
```

### 3. Baze de date. Design si implementare

#### 3.1. Notiuni introductive

O baza de date reprezinta o colectie de date. Ca si exemple de baze de date putem avea: fisierele din directoarele unei partitii, agenda telefonica, registrul. Bazele de date necesita existenta unui sistem de administrare, care se prezinta sub forma unui software ce poarta denumirea de Sistem de Gestiune a Bazelor de Date (SGBD) (DBMS). Cele mai utilizate sisteme de gestiune a bazelor de date sunt: Oracle, SQLServer, Access, mSQL, PostgreSQL si MySQL.

Exista mai multe tipuri de sisteme de gestiune (ierarhic, orientat-obiect, XML), dar cel mai popular este sistemul relational (RDBMS).

Bazele de date relationale sunt construite din relatii, care mai poarta denumirea de tabele. Datele privitoare la clientii unei firme alcatuiesc un exemplu de tabel (clienti):

Clienti

IdClient	Nume	Adresa	Oras
1	Marian Bucos	Ardealului nr.23	Timisoara
2	Avram Mircea	Labirint bl.4 ap.23	Orsova
3	Dima Corina	1 Decembrie 1918	Bacau

Tabelul are un nume, un numar de coloane, fiecare corespunzand unui anumit tip de date, si inregistrari ce corespund unui client. Coloanele mai poarta denumirea de campuri sau attribute.

Bazele de date sunt alcatuite din unul sau mai multe tabele. Pentru exemplul nostru consideram o alta tabela, comenzi, cu urmatoarele campuri: IdComanda, IdClient, Valoare, Data.

## Comenzi

IdComanda	IdClient	Valoare	Data
1	2	134.67	2005-07-12
2	1	23.9	2006-01-20
3	3	150	2006-03-13
4	1	1234	2006-09-24

### 3.2. Proiectarea unei baze de date

Proiectarea unei baze de date este un proces foarte important care cuprinde urmatorii pasi:

- analiza problemei;
- identificarea datelor;
- normalizarea.

Normalizarea unei baze de date consta in principal in descompunerea modelului bazei de date in mai multe relatii astfel incat sa se reduca la maxim redundanta datelor si implicit sa elimine anomaliiile de actualizare.

Forme normale:

**1NF** – domeniul atributelor sa cuprinda valori atomice; se interzic campurile compuse sau relatii in relatii;

**2NF** – 1NF; orice atribut neprim (care nu face parte din cheia primara) sa fie complet dependent functional de cheia primara a relatiei;

**3NF** – 2NF; nu exista nici un atribut neprim care sa fie dependent tranzitiv de cheia primara a relatiei;

In interiorul tabelor prezinta importanta campurile sau grupurile de campuri care identifica unic inregistrarile sau care ajuta la legarea datelor din mai multe tabele. Cheia primara reprezinta un atribut sau un grup de attribute care identifica unic o inregistrare. Se numeste cheie externa un atribut sau un grup de attribute care constituie o cheie primara intr-o alta relatie.

Identificati in tabelele clienti si comenzi attributele care sunt chei primare, respectiv externe.



### 3.3. Crearea bazei de date in MySQL

Pentru a crea o baza de date se utilizeaza comanda:

```
create database [if not exists] nume_baza_date;
```

Clauza *if not exists* inhiba afisarea unui mesaj de eroare in cazul in care in sistem exista o alta baza de date cu acelasi nume.

Stergerea unei baze de date se realizeaza cu sintaxa:

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

Daca sintaxa comenzii *drop* include clauza *if exists* nu sunt afisate mesajele de eroare care pot aparea daca se incearca stergerea unei baze de date care nu exista.

Comanda *use* stabileste baza de date pentru care se vor executa interogari ulterioare. La un moment dat poate fi activa o singura baza de date.

### 3.4. Tabele

Pentru a crea o tabela se va utiliza comanda:

```
create table [if not exists] nume_tabel [definitie]
```

, unde definitie:

```
nume_coloana tip [not null | null] [default valoare]  
[auto_increment] [primary key]
```

Stergerea unei tabele se face folosind urmatoarea sintaxa:

```
drop table [if exists] nume_tabel;
```

### 3.5. Tipuri de date, operatori si functii

#### 3.5.1. Tipuri de date

Tipuri de date: numerice, logice, data calendaristica, timp, siruri de caractere, date binare mari (blob).

Optiuni: unsigned, zerofill, binary.

1. Tipuri de date numerice: tinyint, smallint, mediumint,int, bigint, float, double, real, decimal.

2. Tipuri de date data calendaristica si timp: date, datetime, timestamp, time, year.
3. Tipuri de date sir de caractere: char, bit, bool, varchar.
4. Tipuri de date binare mari: tinyblob, tinytext, blob, text, mediumblob, mediumtext, longblob,mediumtext.

### 3.5.2. Constante, identificatori, comentarii

Constante tip sir de caractere: 'constanta sir' , "alta constanta sir".

Constante intregi, reale: 1345, 543.36 .

Constanta NULL – nici o valoare; NULL<>0 ; NULL<>' '.

Numele pentru baza de date, tabele, coloane trebuie sa indeplineasca conditiile normale pentru identificatori in limbaje de programare: sa fie o combinatie de litere, cifre si semne grafica care incep cu o litera.

Comantarii pe o singura linie: # comentariu1

-- comentariu 2 --

Comentariu pe mai multe linii: /\* comentariu 3.1

comentariu 3.2 \*/

### 3.5.3. Operatori

- grupare: (, )

- aritmetici: +, -, \*, /

- logici: NOT !, OR ||, AND &&

- de comparare: =, <>, !=, <, >, <=, >=, IS NULL, expr BETWEEN min AND max, expr IN (value....), IF (expr1,expr2,expr3)

### 3.5.4. Functii – utilizare functii de biblioteca

#### 1. Functii pe siruri de caractere

- ASCII (str) – intoarce codul ASCII al caracterului de pe pozitia 1 din sir;

- CONV (N, from\_base, to\_base) – converteste numarul N considerat in baza from\_base in valoarea sa in baza to\_base.

- CHAR (N,...) – convertește sirul de numere primite într-un sir de caractere ale caror coduri ASCII sunt egale cu cele din sirul initial.
- CONCAT(str1, str2, ...) – concatenează sirurile primite ca argument, rezultând sirul rezultat.
- LENGTH(str) – întoarce lungimea (numarului de caractere) a sirului primit ca si argument.
- LOCATE(substr, str)
  - POSITION (substr IN str) – caută prima apariție a sirului substr în sirul str; returnează poziția.
- LEFT(str, len) – întoarce un sir format din primele len caractere din sirul dat.
- RIGHT(str, len) – întoarce un sir format din ultimele len caractere din sirul dat.
- SUBSTRING(str,pos,len)
  - MID(str,pos,len) – întorc subsirul de pe poziția pos, de lungime len caractere în sirul str.
- LTRIM(str), RTRIM(str), TRIM(str) – întoarce sirul str fără spațiile de la început, sfârșit, sau din ambele părți.
- SPACE(N) – întoarce un sir format din N spații.
- LOWER(str), UPPER(str) – întorc sirul str cu toate caracterele convertite la litere mici, mari.
- sir (NOT) LIKE tipar – compară sirul sir cu tiparul dat

## 2. Funcții matematice

- ABS (x) – valoarea absolută (fără semn) a lui x.
- SIGN (x) – testează semnul lui x, întoarce -1 negativ, 1 pozitiv, 0 în caz contrar.
- MOD(n,m), % - modulo (restul împărțirii) lui n la m.
- FLOOR(x) – cea mai mare valoare întreagă mai mică decât x (rotunjire în jos).
- CEILING(x) – cea mai mică valoare întreagă mai mare decât x (rotunjire în sus).
- ROUND(x) – rotunjire la cel mai apropiat întreg.
- EXP(x) – exponențiala (e la puterea x).
- LOG(x) – logaritm natural din x.
- POW(x,y) – x la puterea y.

- SQRT(x) – radicat de ordinul 2 din x.
- PI() – numarul PI.
- COS(x), SIN(x), ... – cosinus, sinus, ...
- RAND(), RAND(n) – genereaza un numar pseudo-aleator in intervalul [0, 1]; prin n se poate preciza o valoare de initializare calcul.
- LEAST(x,y,...) – intoarce cea mai mica valoare din lista valorilor specificate.
- GREATEST(x,y, ...) – intoarce cea mai mare valoare din lista valorilor specificate.

### **3. Functii de tip data calendaristica si timp**

- DAYOFWEEK(date) – indexul zilei din saptamana al datei specificate (1-d,...).
- DAYOFYEAR(date) – numarul zilei din an pentru data specificata.
- DAYNAME(date) – numele in engleza al zilei din data specificata.
- MONTHNAME(date) – numele in engleza al lunii din data specificata.
- YEAR(date) – extrage anul din data specificata.
- HOUR(time) – extrage ora din timpul specificat.
- MINUTE(time) – extrage minutul din timpul specificat.
- SECOND(time) – extrage secunda din timpul specificat.
- CURDATE() – intoarce data curenta in format 'YYYY-MM-DD'.
- CURTIME() – intoarce ora curenta in format 'HH:MM:SS'.
- NOW(), SYSDATE() – intoarce data si ora curenta in format 'YYYY-MM-DD HH:MM:SS'.
- SEC\_TO\_TIME(seconds) – intoarce timpul in format 'HH:MM:SS' reprezentat de numarul de secunde specificate.
- TIME\_TO\_SEC(time) – converteste timpul specificat in numar de secunde scurse de la ora 00:00:00.

### **4. Functii speciale**

- DATABASE() – numele bazei de date active.
- USER() – numele utilizatorului conectat la server.
- VERSION() – returneaza versiunea serverului MySQL instalat.

**DBMS** – Database Management System. Mecanism pentru administrarea si manipularea datelor din bazele de date.

**RDBMS** – Relational Database Management System

**Ex:** Proiectati o baza de date pentru gestiunea unor operatii de depunere si extragere pentru conturi bancare. Ea va cuprinde date despre deponenti, conturi, operatii si tipuri operatii.

```
create database banca;
use banca;
create table conturi(
    nr_cont char(6) not null default "",
    nr_buletin char(8) not null default "",
    tip_cont char(3) default null,
    termen_depozit int(3) default '3',
    sold int(11) default null,
    primary key (nr_cont));
create table deponent(
    nr_buletin char(8) not null default "",
    cnp char(13) not null default "",
    nume char(24) not null default "",
    prenume char(24) not null default "",
    datan date default null,
    oras char(24) default 'timisoara',
    adresa char(32) default null,
    telefon char(9) default null,
    primary key (nr_buletin));
create table operatii(
    nr_cont char(6) not null default "",
    suma int(11) default null,
    data timestamp(14) not null,
    termen_depozit int(3) default '3',
```

```
sold int(11) default null,  
primary key (nr_cont, data));
```

## 4. Gestionarea datelor folosind interogari SQL

### 4.1. Adaugarea inregistrarilor in tabele

Comanda INSERT permite inserarea (adaugarea) de noi inregistrari intr-o tabela.

Comanda INSERT din mysql are sintaxa de baza:

```
INSERT [LOW_PRIORITY| DELAYED] [IGNORE]
      [INTO] nume_tabel [(nume_coloana, ...)]
      VALUES (expresie, ...), (...), ...
```

LOW\_PRIORITY – se foloseste pentru a intarzia scrierea efectiva a datelor in tabela pana cand alti utilizatori nu mai citesc date din tabela. Efectul este blocarea executiei pana cand se reuseste scrierea efectiva.

DELAYED – are actiune opusa parametrului precedent. Inregistrarea care trebuie adaugata este pusa intr-o coada de asteptare pe server si controlul revine la client, ca si cum scrierea ar fi fost facuta efectiv.

IGNORE – este util daca se insereaza mai multe inregistrari simultan. Prin folosirea parametrului IGNORE inregistrarile gresite sunt ignorate, dar restul sunt adaugate fara raportarea unei erori.

INTO – este optional, se foloseste pentru compatibilitatea cu alte sisteme SQL.

VALUES – specifica seturile de date (inregistrarile) care se vor adauga in tabela.

Tema 1

Adaugati o inregistrare in una din tabelele bazei de date BANCA. Vizualizati mai intai structura tabelor acestei baze de date.

Folositi instructiunea INSERT pentru a adauga simultan mai multe inregistrari in una din tabelele bazei de date BANCA.

## 4.2. Modificarea inregistrarilor din tabele

Comanda UPDATE permite actualizarea (modificarea) valorilor dintr-o tabela.

```
UPDATE [LOW_PRIORITY] [IGNORE] nume_tabel  
SET nume_col1 = expresie1, nume_col2 = expresie2, ....  
[WHERE conditie_actualizare]
```

WHERE conditie\_actualizare – indica acele inregistrari care sunt actualizate.

Tema 2

Modificati valorile pentru doua coloane ale unei tabele din baza de date BANCA.

Modificarea se va realiza pentru o singura inregistrare folosind o conditie de actualizare.

## 4.3. Stergerea inregistrarilor din tabele

Pentru stergerea datelor din tabele se foloseste comanda DELETE.

Datele o data sterse nu mai pot fi recuperate.

Sintaxa comenzii DELETE este:

```
DELETE [LOW_PRIORITY] FROM nume_tabel  
[WHERE conditie_stergere]
```

Tema 3

Stergeti o inregistrare din una din tabelele bazei de date BANCA folosind conditia de stergere.

## 4.4. Interogarea datelor

### Selectia simpla

Comanda SELECT este cea mai utilizata comanda SQL. Ea permite atat regasirea si vizualizarea datelor din tabelele bazei de date cat si calcularea unor expresii care nu au legatura cu datele din tabele.

Sintaxa comenzii SELECT este:



```
SELECT [DISTINCT|ALL] expresie _selectie  
[FROM referinte_tabele] [WHERE conditie_selectie]  
[ORDER BY nume_coloana ASC|DESC]
```

DISTINCT – permite eliminarea randurilor duplicate din selectie.

ALL – rezultatul selectiei va cuprinde si randurile duplicate.(implicit)

FROM referinte\_tabele – specifica tabela sau tabelele din care se selecteaza datele. In cazul in care referinta cuprinde mai multe tabele operatia este una de JOIN.

WHERE conditie\_selectie – specifica conditia de includere a datelor in selectie. In cazul operatiunii de JOIN aici se vor adauga si constrangerile date de relatiile dintre tabele.

ORDER BY – permite ordonarea rezultatului dupa anumite coloane sau expresii

#### Tema 4

Selectati toate inregistrarile dintr-o tabela a bazei de date BANCA.

Selectati datele din 3 coloane care indeplinesc o anumita conditie de selectie.

Afisati datele din 2 coloane ale tablei ordonate dupa una din coloane ascendent.

#### **Folosirea alias-urilor**

Pentru a simplifica scrierea instructiunilor se pot folosi denumiri prescurtate, numite alias-uri (porecle). Ele pot fi folosite pentru a desemna tabele cat si coloane.

Ex.

```
SELECT alias_t1.sold AS alias_c1 FROM conturi alias_t1 ORDER BY alias_c1;
```

#### **#alias**

```
select d.nume, c.nr_cont, c.tip_cont, c.sold  
from deponent d, conturi c where d.nr_buletin=c.nr_buletin;
```

#### **#alias**

```
select deponent.nume, conturi.tip_cont, sum(sold) as sold_tip_cont  
from deponent left join conturi using(nr_buletin) group by nume, tip_cont;
```

## **Extragerea informatiei de sumarizare**

Informatiile de sumarizare reprezinta informatii globale despre datele din tabele. Informatiile de sumarizare se extrag pe grupuri de inregistrari.

Gruparea se realizeaza folosind urmatoorii parametrii in sintaxa comenzii SELECT:

GROUP BY {coloana | expresie} – specifica dupa ce valori se vaface gruparea. In majoritatea cazurilor expresia de grupare este reprezentata de o singura coloana. Toate inregistrarile cu aceeasi valoare pentru expresia de grupare vor fi considerate ca facand parte din acelasi grup.

Functii de sumarizare: count(),avg(), min(), max(), sum()

## **JOIN**

Limbajul SQL foloseste comanda SELECT pentru implementarea operatorului JOIN. Acest operator permite colectarea datelor din tabele aflate in legaturi relationale. Folosirea comenzii SELECT pentru a face JOIN presupune specificarea in clauza FROM a tabelor de unde se preiau datele si, in lista de proiectie, a campurilor care vor face parte din rezultat.

```
SELECT lista_campuri FROM lista_tabele
```

```
WHERE conditii_join AND conditii_selectie ORDER BY ...
```

Exista 2 categorii de JOIN:

INNER JOIN – se includ in rezultat doar campurile care au corespondent la ambele capete ale relatiei

OUTER JOIN – va completa automat campurile care lipsesc din tabela corespondenta cu valoarea NULL (LEFT OUTER JOIN ; RIGHT OUTER JOIN)

Sintaxa comenzii SELECT se modifica in felul urmatoar:

```
SELECT lista_campuri
```

```
FROM tabela1 [INNER JOIN | LEFT OUTER JOIN | RIGHT OUTER JOIN] tabela2
```

```
ON conditie_join
```

```
WHERE conditie_selectie
```

## ORDER BY ...

Ex.

**#inner join**

#selecteaza toate conturile pe care le are un deponent

**#1**

```
select deponent.nume, conturi.nr_cont, conturi.tip_cont from deponent, conturi
where deponent.nr_buletin=conturi.nr_buletin;
```

**#2**

```
select deponent.nume, conturi.nr_cont, conturi.tip_cont
from deponent inner join conturi on deponent.nr_buletin=conturi.nr_buletin;
```

**#3**

#campul folosit cu using() trebuie sa fie acelasi in ambele tabele

```
select deponent.nume, conturi.nr_cont, conturi.tip_cont
from deponent inner join conturi using(nr_buletin);
```

**#left join**

#daca vrem info despre toti deponentii indiferent daca au sau nu

```
select deponent.nume, conturi.nr_cont, conturi.tip_cont
from deponent left join conturi on deponent.nr_buletin=conturi.nr_buletin;
```

Ex.

**#sum** – calculeaza suma valorilor expresiei pentru toate inregistrarile din fiecare grup

```
select deponent.nume, conturi.tip_cont, sum(sold)
from deponent left join conturi using(nr_buletin) group by nume, tip_cont;
```

**#max** – calculeaza valoarea maxima a expresiei pentru toate inregistrarile

```
select deponent.nume, conturi.tip_cont, max(sold) as max_sold_tip_cont
from deponent left join conturi using(nr_buletin) group by tip_cont
order by max_sold_tip_cont desc;
```

**#min** - calculeaza valoarea minima a expresiei pentru toate inregistrarile

```
select deponent.nume, conturi.tip_cont, min(sold) as min_sold_tip_cont
```

```
from deponent left join conturi using(nr_buletin) group by tip_cont  
order by min_sold_tip_cont desc;
```

**#avg** – calculeaza media aritmetica a val. expresiei pentru inregistrarile din fiecare grup

```
select deponent.nume, conturi.tip_cont, avg(sold) as avg_sold_tip_cont  
from deponent left join conturi using(nr_buletin) group by nume, tip_cont  
order by avg_sold_tip_cont desc;
```

**#count** – numara cate inregistrari sunt selectate in fiecare grup

```
select deponent.nume, conturi.tip_cont, count(tip_cont) as count  
from deponent left join conturi using(nr_buletin) group by nume, tip_cont  
order by count desc;
```

## 5. Limbajul de scripting PHP

### 5.1. Notiuni introductive

Limbajul PHP este un limbaj de tip script creat special pentru Web. Este un limbaj interpretat de server (server-side scripting language). Permite generarea flexibila a unor pagini dinamice care ofera informatii in timp real.

Sa se afiseze prin intermediul unui script mesajul Hello World!.

```
<?php
    echo 'Hello World!';
?>
```

### 5.2. Inserarea codului PHP intr-o pagina HTML

Scripturile PHP sunt destinate generarii dinamice de continut in paginile Web. Un astfel de script trebuie inclus intr-o pagina HTML. Se pot utiliza urmatoarele tipuri de tag-uri HTML:

-stilul scurt (nu este valabil pentru php5):

```
<? echo "text generat dinamic"; ?>
```

-stilul XML:

```
<?php echo "text generat dinamic"; ?>
```

-stilul SCRIPT:

```
<SCRIPT LANGUAGE='php'> echo "text generat dinamic"; </SCRIPT>
```

-stilul ASP:

```
<% echo "text generat dinamic"; %>
```

### 5.3. Generarea dinamica a continutului

Codul PHP nu va fi vizibil in sursa trimisa de serverul Web catre navigatorul clientului. Acest text va fi interpretat de catre modulul PHP din server si doar textul generat de acest script va fi trimis mai departe de catre server.

La scrierea codului, instructiunile PHP din script vor fi terminate printr-un separator ; .

Comentariile pot fi adaugate in trei feluri:

- multilinie : `/* comentariu */`
- linie: `// comentariu`
- linie: `# comentariu`

Ex: generarea dinamica de cod – afisarea datei si orei

```
<?php
    echo "Data curenta pe server: " ;
    echo date("H:i, jS F"); // se obtine data si ora curenta cu ajutorul functiei date()
    echo "<br>";
?>
```

## 5.4. Tipuri de date

Limbajul PHP suporta urmatoarele tipuri de date: Integer, Double, String, Array, Object.

In PHP variabilele nu trebuie declarate explicit (ele sunt considerate ca fiind declarate implicit la prima lor utilizare de cod), iar tipul de date asigurat este determinat de valoarea retinuta in acel moment.

```
$var = 1;
$var = "sirul 1";
```

Tipul datelor extrase din variabile poate fi schimbat fortat (casting sau conversie de tip) in expresiile unde acestea intervin. Acest lucru se face prin precizarea noului tip intre paranteze rotunde in fata variabilei respective:

```
$var = 1;
$var2 = (double)$var1 + 3.52;
```

## 5.5. Identificatori, constante, variabile

Identificatorii sunt cuvinte care denota entitati intr-un program. Exemple de identificatori sunt numele de variabile, de functii sau de clase.

Reguli:           - pot fi de orice lungime si pot fi formati din litere, cifre, liniuta de subliniere sau semnul \$;

- nu pot incepe cu o cifra;
- sunt de tip case sensitive.

PHP permite folosirea a doua tipuri de constante: constante tip valoare, constante cu nume.

```
define(PI, 3.14);
$unUnghi = PI;
$altUnghi = 2*PI;
```

Variabilele sunt locatii de memorie desemnate printr-un identificator care incepe cu \$. Ele permit memorarea valorilor de diverse tipuri. O variabila se declara implicit prima data cand este folosita.

## 5.6. Extragerea datelor din formularele HTML

Principalul mecanism de colectarea a datelor de la utilizatori prin intermediul unui site Web il reprezinta formularele HTML.

formular.php

```
<html>
<head>
<title>form</form>
</head>
<body>
<form name="formular" method="post" action="rezultat.php">
<p>Date personale
<p>Nume <input type="type" name="nume" size="36" maxlength="36"><br>
  Prenume <input type="type" name="prenume" size="36" maxlength="36"><br>
  Mijlocul de transport preferat
  <select name="transport">
    <option>masina</option>
    <option>trenul</option>
    <option>avionul</option>
    <option>vaporul</option>
  </select>
<p><input type="submit" name="trimite" value="Submit">
</form>
</body>
```

```

</html>
    rezultat.php
<html>
<head>
<title>rezultat</title>
</head>
<body>
<?php
    $nume = $_POST["nume"];
    $prenume = $_POST["nume"];
    $transport = $_POST["transport"];
    echo "Nume persoana: ". $nume."<br>";
    echo "Prenume persoana: ". $prenume."<br>";
    echo "Mijloc de transport preferat: ". $transport."<br>";
?>
</body>
</html>

```

## 5.7. Operatori, expresii

Operatorii sunt simboluri care pot fi utilizate pentru a efectua operatii asupra unor variabile si constante de diverse tipuri.

- Tipuri:
1. aritmetici: +, -, \*, /, %, ++, --
  2. pe siruri de caractere: .
  3. de asignare: =, +=, -=, \*=, /=, %=, .=
  4. de comparare: ==, !=, <>, <, >, <=, >=
  5. logici: !, &&, and, ||, or, xor
  6. aditionali: conditional, suprimare eroare

Operatorul conditional are sintaxa: `conditie ? valoare_adevarata : valoare_falsa`

```
echo ($varsta<18) ? "adolescent" : "matur";
```

Operatorul de suprimare a erorilor @ are sintaxa: `@(expresie);`

```
echo ($b/0);
```

```
echo @($b/0);
```



O atentie deosebita trebuie acordata sirurilor de caractere intre ghilimele " ". La evaluarea expresiei in care acestea apar, toate numele de variabile care apar in interiorul lor sunt inlocuite cu valorile corespunzatoare.

```
$nume = "Vasile";  
$sir1 = "Domnul $nume are masina rosie!";  
$sir2 = 'Domnul $nume are masina rosie!';  
echo $sir1; //afiseaza Domnul Vasile are masina rosie!  
echo $sir2; //afiseaza Domnul $nume are masina rosie!
```

## 5.8. Instructiuni

### 5.8.1. Instructiuni conditionale

- a) – instructiunea **IF** permitea luarea de decizii privind executia unei alte instructiuni sau a unui bloc de instructiuni;

```
if (expr1) instructiune_1  
elseif (expr2) instructiune_2  
<? $num=7;  
    if ($num>5) { echo " num este mai mare decat 5"; }  
    echo " textul acesta se afiseaza tot timpul"; ?>  
  
<? $varsta=32;  
    if ($varsta>20 && $varsta<40) { echo " varsta este intre 20 si 40"; }    ?>  
  
<? $varsta=32;  
    if (!(($varsta>20)) { echo " varsta nu este mai mare de 20"; }    ?>
```

- b) – instructiunea **SWITCH** este similara cu o serie de instructiuni if care testeaza mai multe valori posibile pentru o anumita expresie;

```
switch (expr) {  
    case val_1: instructiune_1; break;  
    case val_2: instructiune_2; break;  
    ...  
    case val_n: instructiune_n; break;  
    default: instructiune_default; break;  
}
```

```

<? $num=5;
  switch ($num) {
    case 5: echo "num este 5"; break;
    case 10: echo "num este 10"; break;
    default: echo "num nu este 5 sau 10"; break;
  } ?>

```

### 5.8.2. Instructiuni repetitive

a) – instructiunea **WHILE** se foloseste pentru a implementa o bucla repetitiva cu test initial;

```

while (expr)
  instructiune

```

```

<? $i=0;
  while ($i<6) {
    echo $i. "<br>";
    $i++;
  } ?>

```

b) – instructiunea **DO-WHILE** este asemanatoare cu instructiunea while dar se foloseste pentru a implementa o bucla repetitiva cu test final;

```

do
  instructiune

```

```

while (expr)
<? $i = 0;
  do {
    echo $i;
    $i++;
  } while ($i < 6); ?>

```

c) – instructiunea **FOR** este utilizata pentru a implementa bucle repetitive cu numar fix de pasi;

```

for (expr1; expr2; expr3)
  instructiune
<? for ($i=0; $i<6; $i++) {
  echo $i.<br>; } ?>

```

#### d) **BREAK & EXIT**

```
<? for ($i=0; $i<20; $i++) {  
    echo $i. "<br>";  
    if ($i==14) break; }  
echo "bucla oprita la ".$i; ?>
```

```
<? echo "acest text este afisat ";  
    exit;  
    echo "acest text nu va fi afisat"; ?>
```

## 5.9. Tablouri

Declararea unui tablou se face la fel ca declararea oricarei variabile.

– tablou indexat numeric

```
$tablou = array("element_1", "element_2", "element_3", "element_4", "element_5");
```

```
<? for($i=0; $i<5; $i++) { //parcurgere cu for  
    echo "elementul no ".$i. "este ".$tablou[$i]. "<br>";  
} ?>
```

count(\$tablou)

sort(\$tablou) – sortare ascendenta

rsort(\$tablou) – sortare descendenta

– tablou asociativ

```
$tablou = array("en"=>"Hi there", "fr"=>"Salut", "es"=>"Digame");
```

```
echo $tablou["fr"];
```

```
<? $tablou = array("en"=>"Hi there", "fr"=>"Salut", "es"=>"Digame");  
    foreach($tablou as $value) {  
        echo " Value..": $value."<br>";  
    } ?>
```

```
<? $i=0;  
    $tablou = array("en"=>"Hi there", "fr"=>"Salut", "es"=>"Digame");  
    foreach($tablou as $key => $value) {  
        echo "Key..": $key." Value..": $value."<br>";  
    } ?>
```

```
    $i++;  
} ?>
```

– tablou multidimensional

```
<? $tablou = array( array ("1", "2", "3", "4"),  
    array ("a", "b", "c", "d"));  
echo $tablou[0][1]; ?>
```

```
<? $a[0][0] = "a"; $a[0][1] = "b"; $a[1][0] = "y"; $a[1][1] = "z";  
    foreach ($a as $v1) {  
        foreach ($v1 as $v2) {  
            echo "$v2\n";  
        }  
    }  
} ?>
```

## 5.10. Functii

Sunt utilizate pentru a grupa instructiunile care executa un anumit algoritm. Permit re folosirea codului ori de cate ori este nevoie de acel algoritm intr-un program.

– declarare functie:

```
function nume_functie() {  
    //secventa de instructiuni  
}
```

### Functii de prelucrare tablouri

- 1 void `asort` (array tablou)
- 2 void `arsort` (array tablou)
- 3 int `count` (array tablou)
- 4 array `array_keys`(array tablou);

### Functii matematice

- 5 tip\_n `abs`(tip\_n numar)
- 6 float `ceil`(float numar)
- 7 float `floor`(float numar)
- 8 float `round`(float numar)

- 9 float cos(float unghi)
- 10 float sin(float unghi)
- 11 float tan(float unghi)
- 12 float exp(float numar)
- 13 tip\_n pow(tip\_n baza, tip\_n exponent)
- 14 float log(float numar)
- 15 float sqrt(float numar)

### **Funcții de prelucrare șiruri de caractere**

- 16 string chr(int ascii)
- 17 int ord(string sir)
- 18 array explode(string separator, string sir)
- 19 string implode(string lipici, array tablouSiruri)
- 20 string ltrim(string sir)
- 21 string rtrim(string sir)
- 22 string trim(string sir)
- 23 void printf(string format, tip\_oarecare valoare1....)
- 24 int strpos(string sirOriginal, string sirCautat)
- 25 string substr(string sir, int start, [int lungime])
- 26 int strlen(string sir)
- 27 string strtolower(string sir)
- 28 string strtoupper(string sir)

## 6. Accesarea serverului MySQL utilizand PHP

### 6.1. Conectarea la serverul de baza de date

Pentru a putea interoga o baza de date , o pagina PHP trebuie sa stabileasca in prealabil o conexiune cu serverul mySQL. Conexiunea poate fi privita ca un canal de comunicatie prin care programul transmite cereri SQL iar serverul returneaza raspunsurile corespunzatoare.

Functia care realizeaza conectarea persistenta la serverul de baza de date este *mysql\_pconnect()*.

In general trebuiesc introduse host-ul server-ului mySQL, numele cu care utilizatorul se conecteaza si parola folosita de acesta. Toate acestea sunt optionale, iar daca nu sunt specificate functia foloseste setarile default – localhost pentru host, numele utilizatorului cu care procesul PHP ruleaza si o parola vida.

Functia returneaza o legatura catre baza de date sau eroare in caz de nereusita.

O alta functie care realizeaza conectarea – conexiune nepersistenta – la serverul de baza de date este *mysql\_connect()*.

O conexiune va fi inchisa cand se incheie executia unui script – terminare pagina – sau cand se apeleaza functia *mysql\_close()*. O conexiune persistenta ramane deschisa si dupa ce executia scriptului s-a terminat si nu poate fi inchisa folosind functia *mysql\_close()*.

```
<?php
$link = mysql_connect("localhost","stud19","stud");

if ($link) echo 'conexiunea s-a realizat cu succes';
else echo 'nu se poate realiza o conexiune la serverul de baza de date';
?>
```

Cand se realizeaza conectarea la serverul de baza de date trebuie stabilita si baza de date cu care se va lucra. Acest lucru se face din PHP utilizand functia *mysql\_select\_db()*.

```
<?php
$link = mysql_connect("localhost","stud19","stud")
    or die('nu se poate realiza o conexiune la serverul de baza de date');

$bd = mysql_select_db("banca");
```

```
if ($bd) echo 'baza de date a fost selectata';
else echo 'baza de date nu poate fi selectata';
?>
```

## 6.2. Rularea interogarilor

Pentru rularea unei interogari pe server se utilizeaza functia *mysql\_query()*.

Functia presupune existenta unei conexiuni deschise spre server, selectia prealabila a bazei de date si existenta unor drepturi suficiente pentru rularea interogarii. Functia returneaza false daca interogarea nu a putut fi executata pe server.

```
<?php
$link = mysql_connect("localhost","stud19","stud")
        or die('nu se poate realiza o conexiune la serverul de baza de date');
$bd = mysql_select_db("banca") or die('baza de date nu poate fi selectata');

$result = mysql_query("select * from deponent");

if ($result) echo 'interogarea a fost rulata cu succes';
else echo 'interogarea nu poate fi rulata';
?>
```

**Obs: ! die()** – afiseaza un mesaj si termina executia scriptului;

## 6.3. Obtinerea si afisarea datelor

Pentru a afla numarul inregistrarilor afectate de o interogare actiune (insert , delete , update) se poate folosi functia *mysql\_affected\_rows()*.

```
<?php
$link = mysql_connect("localhost","stud19","stud")
        or die('nu se poate realiza o conexiune la serverul de baza de date');
$bd = mysql_select_db("banca") or die('baza de date nu poate fi selectata');

$result = mysql_query("insert into deponent(nr_buletin,cnp,nume,prenume)
                        values('342674','73463726','pop','sebastian')")
```

```
or die('interogarea nu poate fi rulata');
```

```
echo 'inregistrari afectate: '.mysql_affected_rows();
```

```
?>
```

In urma rularii unei interogari select se poate folosi functia *mysql\_num\_rows()* pentru a afla numarul inregistrarilor din rezultat.

```
<?php
```

```
$link = mysql_connect("localhost","stud19","stud")
```

```
or die('nu se poate realiza o conexiune la serverul de baza de date');
```

```
$bd = mysql_select_db("banca") or die('baza de date nu poate fi selectata');
```

```
$result = mysql_query("select * from deponent")
```

```
or die('interogarea nu poate fi rulata');
```

```
echo 'inregistrari afectate: '.mysql_num_rows($result);
```

```
?>
```

Exista trei metode diferite de a interpreta si prelucra rezultatul unei interogari select:

a) rezultatul este preluat prin functia *mysql\_fetch\_array* si interpretat ca un tablou asociativ; acest tablou are ca si chei numele coloanelor selectate iar ca si valori, valorile corespondente din rezultat; fiecare apel succesiv al functiei va returna urmatoarea inregistrare din rezultat.

```
<?php
```

```
$link = mysql_connect("localhost","stud19","stud")
```

```
or die('nu se poate realiza o conexiune la serverul de baza de date');
```

```
$bd = mysql_select_db("banca") or die('baza de date nu poate fi selectata');
```

```
$result = mysql_query("select * from deponent")
```

```
or die('interogarea nu poate fi rulata');
```

```
while($row = mysql_fetch_array($result)) {
```

```
    echo $row["nume"].' '.$row["prenume"].'<br>';
```

```
}
```

```
?>
```

b) rezultatul este preluat prin functia *mysql\_fetch\_rows* si interpretat ca un tabel indexat numeric; acest tablou are ca si indexi numerele 0,1,2 etc. iar ca si valori, valorile



corespondente coloanelor de pe pozitiile respective din rezultat; fiecare apel succesiv al functiei va returna urmatoarea inregistrare din rezultat.

```
<?php
$link = mysql_connect("localhost","stud19","stud")
    or die('nu se poate realiza o conexiune la serverul de baza de date');
$dbd = mysql_select_db("banca") or die('baza de date nu poate fi selectata');

$result = mysql_query("select * from deponent")
    or die('interogarea nu poate fi rulata');

while($row = mysql_fetch_row($result)) {
    echo $row[2].' '.$row[3].<br>;
}
?>
```

c) rezultatul este preluat prin functia `mysql_fetch_object` si interpretat ca un obiect.

```
<?php
$link = mysql_connect("localhost","stud19","stud")
    or die('nu se poate realiza o conexiune la serverul de baza de date');
$dbd = mysql_select_db("banca") or die('baza de date nu poate fi selectata');

$result = mysql_query("select * from deponent")
    or die('interogarea nu poate fi rulata');

while($row = mysql_fetch_object($result)) {
    echo $row->nume.' '.$row->prenume.<br>;
}
?>
```

Fiecare apel succesiv al functiei `mysql_fetch_object` va returna urmatoarea interogare din rezultat.

```
<?php
$link = mysql_connect("localhost","stud19","stud")
    or die('nu se poate realiza o conexiune la serverul de baza de date');
$dbd = mysql_select_db("banca") or die('baza de date nu poate fi selectata');

$result = mysql_query("select * from deponent")
    or die('interogarea nu poate fi rulata');
```

```

for($i=0; $i<mysql_num_rows($result); $i++) {
    $row = mysql_fetch_object($result);
    echo $row->nume.' '.$row->prenume.<br>;
}
?>

```

## 6.4. Adaugarea datelor din pagina Web in baza de date

Pentru a adauga date dintr-o pagina web se realizeaza o interogare INSERT pe server utilizand functia mysql\_query().

Aceasta functie presupune existenta unei conexiuni deschise spre server, selectia prealabila a bazei de date si existenta unor drepturi suficiente pentru rularea interogarii.

Urmatorul script implementeaza un mecanism de introducere (adaugare) de noi inregistrari in tabela deponent din baza de date banca.

```

<?php
$link = mysql_connect("localhost","stud19","stud")
    or die('nu se poate realiza o conexiune la serverul de baza de date');
$dbd = mysql_select_db("banca") or die('baza de date nu poate fi selectata');

if ($_GET["action"] == 'insert' && $_POST["nr_buletin"] <> "")
    mysql_query("insert into deponent(nr_buletin,cnp,nume,prenume,oras,adresa)
        values('".$_POST["nr_buletin"]."',".$_POST["cnp"]."',".$_POST["nume"]."',
            '".$_POST["prenume"]."',".$_POST["oras"]."',".$_POST["adresa"]."')")
    or die('interogarea nu poate fi rulata');

$result = mysql_query("select * from deponent")
    or die('interogarea 2 nu poate fi rulata');
echo '[Nr. buletin] - [CNP] - [Nume] - [Prenume] - [Oras] <br />';
while($row = mysql_fetch_object($result)) {
    echo $row->nr_buletin.' '.$row->cnp.' '.$row->nume.' '.$row->prenume.' '.$row->oras.<br />;
}
?>

```

```

<form action="<?php echo $PHP_SELF;?>?action=insert" method="post">

```

```

Nr. buletin: <input type="text" name="nr_buletin"><br />
CNP: <input type="text" name="cnp"><br />
Nume: <input type="text" name="nume"><br />
Prenume: <input name="prenume" type="text"><br />
Oras: <select name="oras">
    <option value="Bucuresti">Bucuresti</option>
    <option value="Timisoara">Timisoara</option>
    <option value="Caransebes">Caransebes</option>
</select><br />
Adresa; <textarea name="adresa" cols="15"></textarea><br />
<input type="submit" name="Submit" value="Submit">
</form>

```

## 6.5. Stergerea datelor prin intermediul unei pagini Web

Pentru a sterge date de pe server se ruleaza o interogare DELETE utilizand functia `mysql_query()`.

Scriptul din exemplul de mai jos realizeaza stergerea individuala a inregistrarilor.

Inregistrarile din deponent sunt afisate in urma rularii unei interogari SELECT. Se observa ca alegerea inregistrarii ce urmeaza a fi stearsa se face prin intermediul unui link in care este transmisa variabila `$delete`, ce pastreaza `nr_buletin` al persoanei. Interogarea DELETE se ruleaza doar daca este transmisa o variabila de tipul `$delete`.

```

<?php
$link = mysql_connect("localhost","stud19","stud")
    or die('nu se poate realiza o conexiune la serverul de baza de date');
$dbd = mysql_select_db("banca") or die('baza de date nu poate fi selectata');

if ($_GET["delete"])
    mysql_query("delete from deponent where nr_buletin='".$_GET["delete"].'")
    or die('interogarea nu poate fi rulata');

$result = mysql_query("select * from deponent") or die('interogarea nu poate fi rulata');

echo '[Nr. buletin] - [CNP] - [Nume] - [Prenume] - [Oras] <br />';

```

```

while($row = mysql_fetch_object($result)) {
    echo '<a href = "'.$PHP_SELF.'?delete='.$row->nr_buletin.'">'.$row->nr_buletin.'</a> '.
        $row->cnp.' '.$row->nume.' '.$row->prenume.' '.$row->oras.'<br />';
}
?>

```

## 6.6. Modificare datelor in interfata Web

Modificarea datelor presupune rulara unei interogari UPDATE. Si in acest caz se utilizeaza tot functia mysql\_query().

```

<?php
$link = mysql_connect("localhost","stud19","stud")
    or die('nu se poate realiza o conexiune la serverul de baza de date');
$dbd = mysql_select_db("banca") or die('baza de date nu poate fi selectata');

if ($_GET["action"] == 'update')
    mysql_query("update deponent set cnp='".$_POST["cnp"]."',nume='".$_POST["nume"]."',
        prenume='".$_POST["prenume"]."',oras='".$_POST["oras"]."',adresa='".$_POST["nume"]."'
        where nr_buletin='".$_GET["nr_buletin"]."' ") or die('interogarea nu poate fi rulata');

$result = mysql_query("select * from deponent") or die('interogarea nu poate fi rulata');

echo '[Nr. buletin] - [CNP] - [Nume] - [Prenume] - [Oras] <br />';
while($row = mysql_fetch_object($result)) {
    echo '<a href = "'.$PHP_SELF.'?update='.$row->nr_buletin.'">'.$row->nr_buletin.'</a> '.
        $row->cnp.' '.$row->nume.' '.$row->prenume.' '.$row->oras.'<br />';
}
if ($_GET["update"]) {
    $result = mysql_query("select * from deponent where nr_buletin='".$_GET["update"]."' ") or
    die('interogarea nu poate fi rulata');
    $row = mysql_fetch_object($result);
}
?>

<form action="<?php echo $PHP_SELF;?>?action=update&nr_buletin=<?php echo $row->nr_buletin;?>"
method="post">

```

```
Nr. buletin: <?php echo $row->nr_buletin;?><br />
CNP: <input type="text" name="cnp" value="<?php echo $row->cnp;?>"><br />
Nume: <input type="text" name="nume" value="<?php echo $row->nume;?>"><br />
Prenume: <input name="prenume" type="text" value="<?php echo $row->prenume;?>"><br />
Oras: <select name="oras">
    <option value="Bucuresti" <?php if ($row->oras == 'Bucuresti') echo
    'selected';?>>Bucuresti</option>
    <option value="Timisoara" <?php if ($row->oras == 'Timisoara') echo
    'selected';?>>Timisoara</option>
    <option value="Caransebes" <?php if ($row->oras == 'Caransebes') echo
    'selected';?>>Caransebes</option>
</select><br />
Adresa; <textarea name="adresa" cols="15"><?php echo $row->adresa;?></textarea><br />
<input type="submit" name="Submit" value="Submit">
</form>

<?php } ?>
```