

# 1. Data Access

Adrian Adiaconitei

***LINKAcademy***

# Agenda

- ✓ Baze de date: MySql
- ✓ XML
- ✓ Json
- ✓ Arhitectura Client-Server
- ✓ WebSocket
- ✓ Salvarea datelor pe parte de client: Cookie-uri  
Local storage

# Objective

- ✓ Introducere în MySQL
  - Introducere în bazele de date
  - Instalarea și pornirea serverului MySQL
  - Proiectarea bazei de date
  - MySQL Workbench/PhpMyAdmin
- ✓ Crearea unei baza de date
  - Tipurile de date
  - Cheile primare și străine

# Datele structurate/nestructurate

- ✓ Noțiunea de **data** se referă la tot ceea ce se găsește în format utilizabil pe calculator.

Datele pot fi:

- ✓ **Nestructurate**

- Imagini, muzică
- Conținuturi din foldere

- ✓ **Structurate**

- Tabele

# Ce este o BAZA DE DATE?

- ✓ **BAZA DE DATE** este un sistem proiectat pentru a oferi un mecanism organizat, capabil să stocheze, să actualizeze și să regăsească informația. Exemplu: o bibliotecă, cartea cu numerele de telefon /agenda telefonică
- ✓ O **baza de date** nu este altceva decât o colecție de date (informații) ce descriu mai multe subiecte (obiecte) similare.

Bazele de date, de pe calculator cel mai des, se clasifică în funcție de abordarea organizatorică. Abordarea organizatorică se mai numește și **model al bazei de date**. Se poate spune că modelul bazei de date determină structura logică a acesteia, precum și modul în care datele sunt stocate, organizate și manipulate.

# Bazele de date

**Definiție:** O baza de date (BD) este un ansamblu structurat de date înregistrat pe suporturi accesibile calculatorului pentru a satisface simultan cerințele mai multor utilizatori într-un mod selectiv și în timp util.

**Definiție:** Un sistem de gestiune a bazelor de date (SGBD/DBMS) este ansamblul de programe care permit utilizatorului să interacționeze cu o bază de date. (Interfața dintre datele fizice și utilizator)

# Modelele bazei de date

Există diferite modele ale bazei de date:

- ✓ Baza de date Flat-File: datele sunt salvate într-un singur fișier/ tabel, fiecare înregistrare ocupând câte o linie ( Ex. Fișiere CSV)
- ✓ Baze de date ierarhice: informația este structurată sub forma de arbore (Ex. Fișiere XML)
- ✓ **Baza de date relațională:** datele sunt salvate în mai multe tabele

.....

# Bazele de date

- ✓ Baze de date relaționale (SQL=Structured Query Language )
- ✓ Baze de date noSQL



# Bazele de date noSql

- Bazele de date NoSQL sunt baze de date care nu au o structură tabelară a datelor, ci folosesc cel mai des colecții de perechi de chei și de valori
- Aceste baze de date deseori se numesc (se găsesc în forma) key value store, indexator, document store, baza de date graph...)
- La aceste baze de date, accentul este pus pe viteza de obținere și de actualizare a datelor, în timp ce autenticitatea datelor și reducerea redundanței este în al doilea plan

# Bazele de date noSql

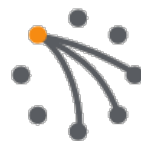
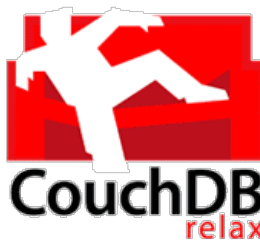
<https://firebase.google.com/>

<https://www.mongodb.com/>

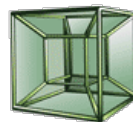
APACHE  
**HBASE**



**Cassandra**



**riak**

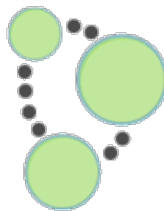


**mongoDB**



**Firebase**

**HYPERTABLE INC**



**Neo4j**



**redis**

<https://www.g2.com/categories/nosql-databases>

# Bazele de date relationale

Informațiile sunt organizate în tabele, fiecare linie descriind câte un subiect/**entitate/tuplu**.

Liniile nu trebuie să fie identice!

O linie poartă numele de înregistrare. Descrierea fiecărei însușiri a subiectului constituie un **câmp**. Informația dintr-un câmp nu trebuie să poată fi descompusă! **Pentru fiecare câmp se alege un nume, un tip de date și o gamă de reprezentare.**

# Bazele de date relationale

Oracle

ORACLE®

Microsoft Sql Server



MySql



PostgreSQL

PostgreSQL



LINK Academy

# Bazele de date

- ✓ <https://bambooagile.eu/insights/the-best-database-for-node-js/>
- ✓ [Top DB](#)



# Bazele de date relaționale

- ✓ **MySQL** este dezvoltată de o companie suedeză numită MySQL AB, fondată de **David Axmark și Michael Widenius** în 1995.
- ✓ [www.mysql.com](http://www.mysql.com) a fost lansat în 1998 împreună cu versiunea 3.21. Prima versiune pentru Windows a fost lansată în 1998 și putea rula pe Windows 95 și Windows NT
- ✓ Sun Microsystems cumpără MySQLAB în 2008, dar păstrează proiectul neschimbat
- ✓ În urma interesului Oracle pentru achiziționarea Sun, și prin extensie MySQL, Michael Widenius creează un fork numit MariaDB în 2009
- ✓ Oracle achiziționează Sun în ianuarie 2010 și preia astfel MySQL
- ✓ Dezvoltarea MySQL și MariaDB continuă în paralel([MySQL 8.0](#) și [MariaDB 11.1](#))

# NoSQL + SQL = MySQL



NoSQL + SQL = MySQL

cademy

# Instalarea serverului MySQL

- ✓ <https://dev.mysql.com/downloads/mysql/>
- ✓ <https://dev.mysql.com/downloads/installer/>

## MySQL Installer for Windows

**All MySQL Products. For All Windows Platforms.  
In One Package.**

Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.



my



# Instalarea serverului MySQL

XAMPP Control Panel v3.2.2 [ Compiled: Nov 12th 2015 ]

**XAMPP Control Panel v3.2.2**

Modules Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache			Start Admin Config Logs
<input type="checkbox"/>	MySQL	11500	3306	Stop Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

Config Netstat Shell Explorer Services Help Quit

1:51:29 AM [main] XAMPP Installation Directory: "c:\xampp\  
1:51:29 AM [main] Checking for prerequisites  
1:51:31 AM [main] All prerequisites found  
1:51:31 AM [main] Initializing Modules  
1:51:31 AM [main] Starting Check-Timer  
1:51:31 AM [main] Control Panel Ready  
1:52:01 AM [mysql] Attempting to start MySQL app...  
1:52:01 AM [mysql] Status change detected: running

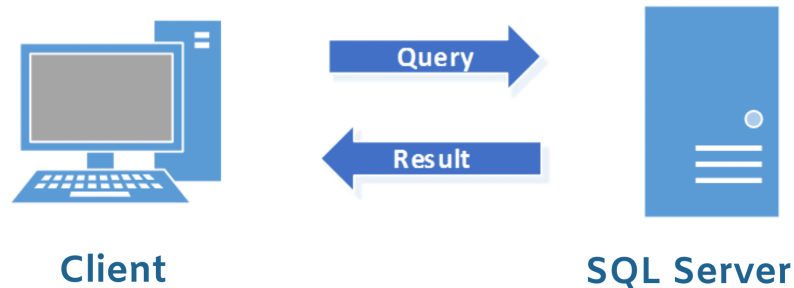
# Introducere în MySQL

- Mediul MySQL este alcătuit din două părți:
  - serverul care gestionează datele,
  - clientul care “cere” de la server execuția anumitor operații
- SQL = Structured Query Language
- **Baza de date:** colecție de informații stocate și organizate
- `SELECT title FROM tabela WHERE nume= 'ion';`

# Introducere în MySQL

Interacțiunea cu MySQL:

- ✓ Linia de comanda  
( `mysql -u username -p`  
`SHOW databases;`)
- ✓ Mysql Workbench
- ✓ Interfața web (phpMyAdmin)
- ✓ Limbaj de programare (nodejs)



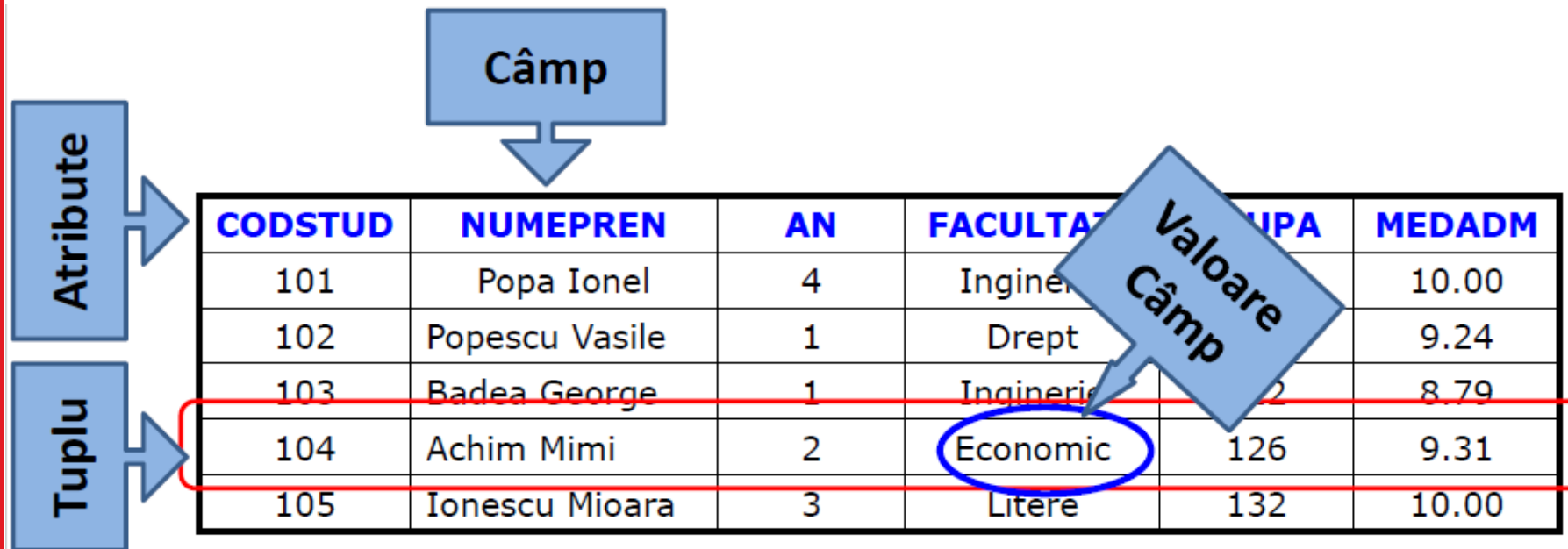
# Introducere în MySQL

Tip	Numar de octeți	Exemplu
<b>CHAR(n)</b>	<b>Exact n (&lt;=255)</b>	<b>CHAR(5)</b>
<b>VARCHAR(n)</b>	<b>Până la n (&lt;=65535)</b>	<b>VARCHAR(100) Greetings – 9 octeți</b>
<b>BINARY(n)</b>		
<b>BYTE(n)</b>	<b>Exact n (&lt;=255)</b>	
<b>VARBINARY(n)</b>	<b>Până la n (&lt;=65535)</b>	
<b>TINYTEXT(n)</b>	<b>Până la n (&lt;=255)</b>	Șiruri de caractere
<b>TEXT(n)</b>	<b>Până la n (&lt;=65535)</b>	
<b>MEDIUMTEXT(n)</b>	<b>Până la n (&lt;=16777215)</b>	
<b>LONGTEXT(n)</b>	<b>Până la n (&lt;=4294967295)</b>	

# Introducere în MySQL

TINYBLOB(n)	Pana la n ( $\leq 255$ )	Date binare
BLOB(n)	Pana la n ( $\leq 65535$ )	
MEDIUMBLOB(n)	Pana la n ( $\leq 16777215$ )	
LONGBLOB(n)	Pana la n ( $\leq 4294967295$ )	
TINYINT	1 octet [-128,127] [0,255]	
SMALLINT	2 octeți	
MEDIUMINT	3 octeți	
INT, INTEGER	4 octeți	
BIGINT	8 octeți	
FLOAT	4 octeți	
DOUBLE, REAL	8 octeți	
DATETIME	'0000-00-00 00:00:00'	
DATE	'0000-00-00'	
TIMESTAMP	'0000-00-00 00:00:00'	
TIME	'00:00:00'	
YEAR	0000	

# Modelul de date relațional



# Cheia primară/secundară

- ✓ **Cheia primară** este principalul factor de restricționare într-un tabel. Aceasta este proprietatea coloanei, care nu asigura repetarea valorilor într-o coloană. Cheia primară este o caracteristica găsită în aproape fiecare tabel, chiar dacă există și posibilitatea evitării introducerii acesteia. Cheia primară este unică
- ✓ Cheia straină este simbolul bazei de date relaționale. Constragerea cheii primare (Foreign Key Constraint) mentine relația fizică între două tabele.

# Limbajul SQL ( MySQL)

- ✓ **DDL (Data Definition Language)**: creare, modificare, stergere, vizualizare baze de date și tabele; **CREATE, ALTER, DROP**
- ✓ **DML (Data Manipulation Language)**: manipularea datelor din tabele : **INSERT, UPDATE, DELETE**
- ✓ **DQL – (Data Query Language)**: permit obținerea datelor din baza de date; **SELECT**
- ✓ **DCL (Data Control Language)**: permit administratorilor să controleze accesul la datele din baza de date și folosirea diferitelor privilegii ale sistemului SGBD; **GRANT, REVOKE**

<https://downloads.mysql.com/docs/refman-5.7-en.a4.pdf>



# DDL

```
CREATE DATABASE gregs_list;
```

Comanda SQL care  
crează baza de date

Numele bazei  
de date

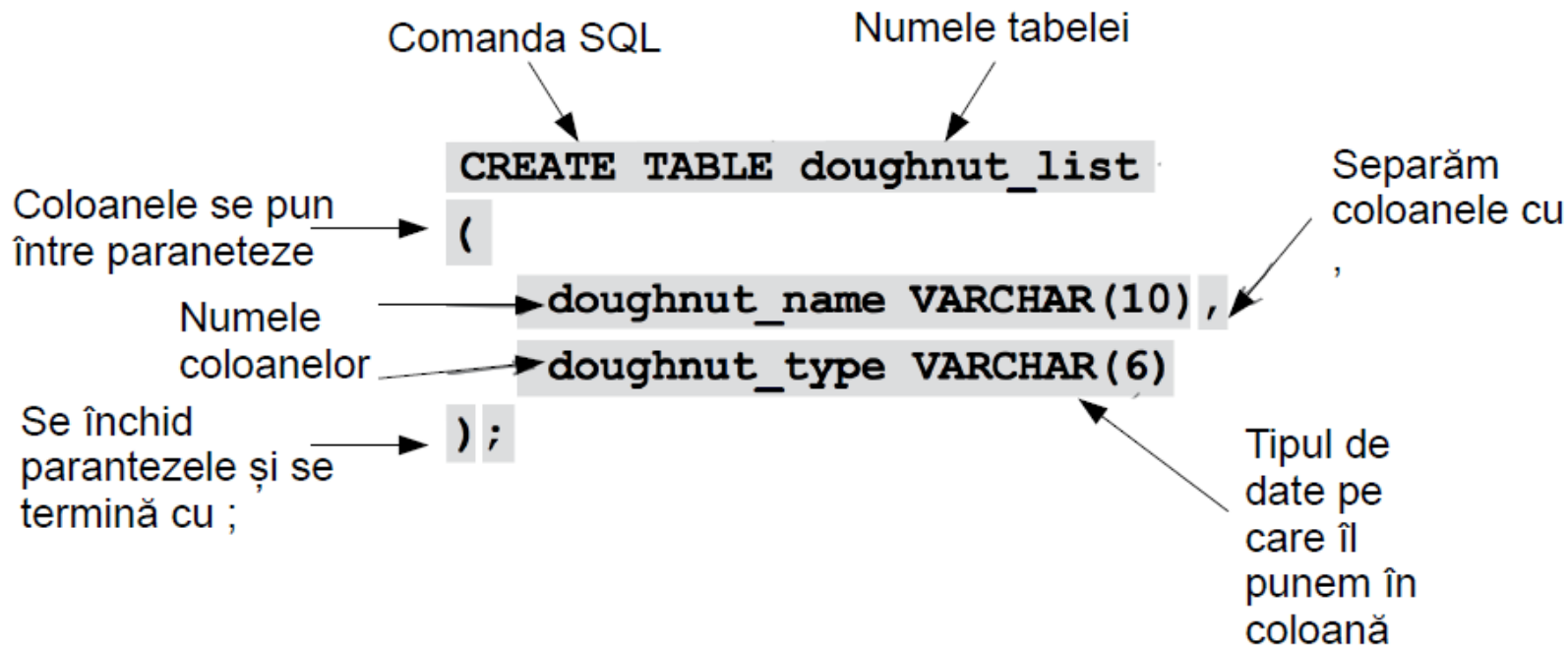
Fiecare comandă  
se termină cu ;

```
USE gregs_list;
```

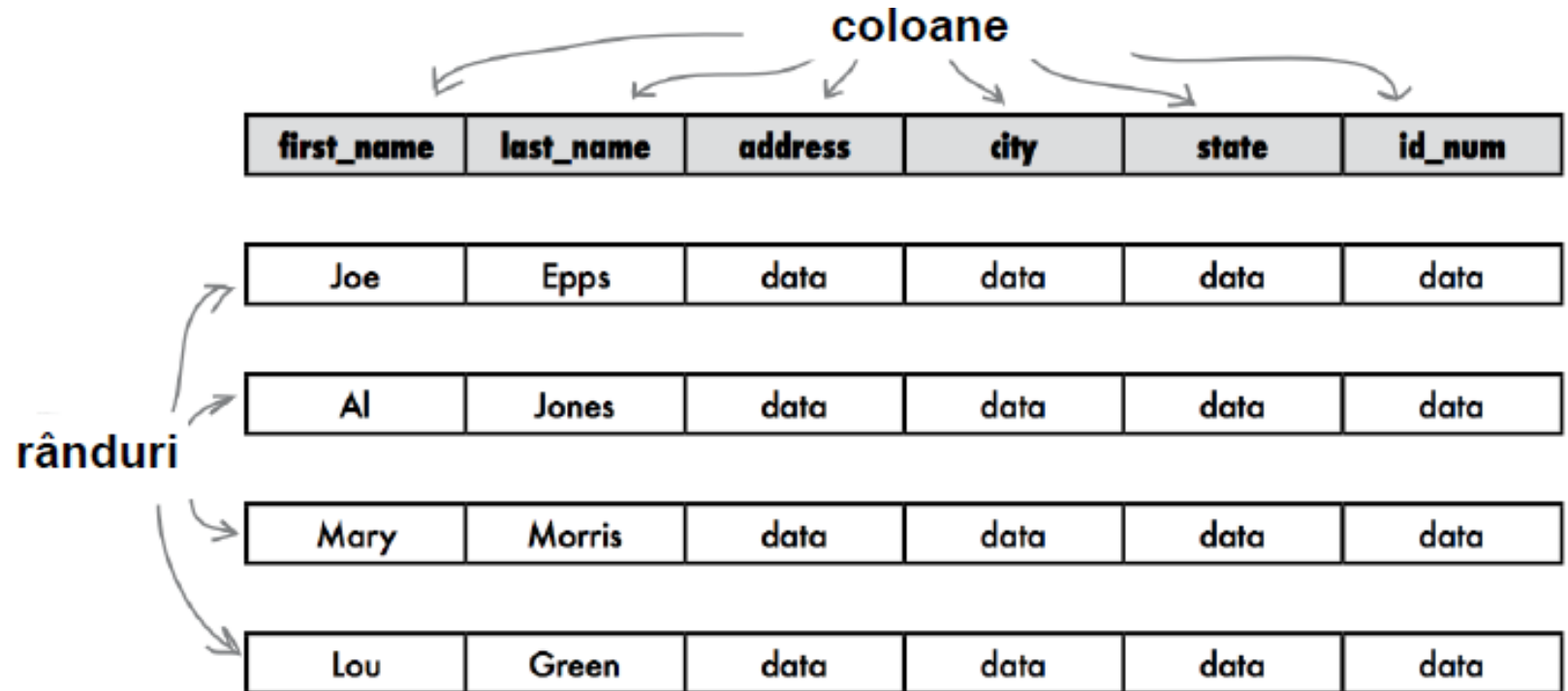
După ce creăm baza de date  
e nevoie să o folosim, folosind  
această comandă.

# DDL

## Crearea tabelelor



# DDL



# DDL (Data Definition Language):

**Aplicația1.** Folosind MySQL Workbench sau PhpMyadmin să se creeze:

- un fișier .sql unde se vor salva comenzile
- o baza de date : **mydb2023**
- o tabela: **jsusers** (id, nume, prenume, email, telefon, data nastere, data adaugare)
- adaugați câteva înregistrări în tabela **jsusers**

# Provocare

Sa se creeze o baza de date relationala care contine urmatoarele informatii despre:

Examen- numar examen, denumire examen, sala examen, ora examenului.

Profesor – numar matricol, nume, prenume; un profesor poate participa examene ca si asistent sau titular.

Student - numar matricol, cnp, nume, prenume, data nasterii. Pentru fiecare student se vor inregistra notele obtinute la fiecare examen.

Se cere:

- 1.Sa fie efectuata analiza atributelor folosite, indicandu-se tipul de date al fiecarui atribut, formuland consideratiuni asupra domeniilor de valori ale atributelor, etc.
- 2.Sa se reprezinte in modelul relational diagrama obtinuta la punctul 1, descriind modul de obtinere a relatiilor si schemele acestora. De asemenea, vor fi precizate cheile relatiilor.

# DML - INSERT

numele tabelii                      numele coloanelor

↓    ↓

```
INSERT INTO your_table (column_name1, column_name2,...)
VALUES ('value1', 'value2',...);
```

↑    ↑

valorile încep  
după cuvântul  
VALUES                      valorile

# DML – UPDATE

- ✓ **UPDATE** [LOW\_PRIORITY] [IGNORE]  
*table\_reference* **SET** *assignment\_list* [WHERE  
*where\_condition*] [ORDER BY ...] [LIMIT  
*row\_count*]
- ✓ **UPDATE t1 SET col1 = col1 + 1, col2 = col1;**

# DML – DELETE

- ✓ **DELETE** [LOW\_PRIORITY] [QUICK] [IGNORE]  
**FROM** *tbl\_name* [PARTITION (*partition\_name* [, *partition\_name*] ...)] [**WHERE** *where\_condition*]  
[ORDER BY ...] [LIMIT *row\_count*]
- ✓ **DELETE FROM** student **WHERE** nume=  
'popescu' ORDER BY data\_nastere LIMIT 1;



# Ștergerea datelor:

- ✓ **DROP DATABASE *databaseName*;**
- ✓ **DROP TABLE [IF EXISTS] *tableName*;**
- ✓ **TRUNCATE TABLE *tableName***
- ✓ **DELETE FROM *tableName* WHERE condiții**

# DQL – (Data Query Language):

**SELECT** [ALL | DISTINCT | DISTINCTROW ]  
[HIGH\_PRIORITY] [STRAIGHT\_JOIN] select\_expr, ...  
[**FROM** table\_references [**WHERE**  
where\_condition] [**GROUP BY** {col\_name | expr |  
position} [ASC | DESC], ... [WITH ROLLUP]] [**HAVING**  
where\_condition] [**ORDER BY** {col\_name | expr |  
position} [ASC | DESC], ...] [**LIMIT** {[offset,  
row\_count | row\_count OFFSET offset]} ]

# SELECT

Scopul instructiunii **SELECT** este de a regasi si afisa datele din unul sau mai multe tabele ale bazei de date.

Secventa de prelucrare dintr-o instructiune SELECT poate contine urmatoarele optiuni:

- ✓ **FROM** - specifica tabelul sau tabelele care vor fi utilizate;
- ✓ **WHERE** - filtreaza randurile supuse unei anumite conditii;
- ✓ **GROUP BY** - formeaza grupuri de randuri cu aceeasi valoare a coloanei;
- ✓ **HAVING** - filtreaza grupurile supuse unei anumite conditii;
- ✓ **ORDER BY** - specifica ordinea iesirii.
- ✓ **LIMIT** - limiteaza rezultatele returnate.

**Ordinea clauzelor din instructiunea SELECT nu poate fi schimbata.** Singurele clauze obligatorii sunt primele doua: SELECT si FROM; restul sunt optionale. Operatiunea SELECT este inchisa: rezultatul unei interogari dintr-un tabel este un alt tabel.

# Instrumente MySQL:

- ✓ [Mysql Workbench](#)
- ✓ [phpMyAdmin](#)
- ✓ HeidiSQL, SequelPro, DataGrip
- ✓ [<http://downloads.mysql.com/docs/sakila-db.zip>](#)
- ✓ [<https://insights.stackoverflow.com/survey/2019#technology--databases>](#)
- ✓ [<https://db-engines.com/en/ranking>](#)

# App – CRUD



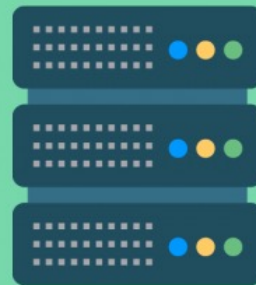
**{REST:API}**



# App – CRUD



**FRONT END**



**BACK END**

# App – CRUD

Front-End

HTML



CSS



JS



BS



# App – CRUD

## Front-End

✓ frontend

✓ CSS

# style.css

✓ js

JS app.js

<> index.html



# App – CRUD

## Back-End

1. `cd backend/`
2. `npm init --yes`    sau `npm init -y => package.json`
3. `npm install express dotenv cors`
4. `npm i -D concurrently nodemon`
5. `npm i -D typescript @types/express @types/node @type/cors`
- `//npm install --dev typescript @types/express @types/node`
6. `npx tsc --init => tsconfig.json`

# App – CRUD

Back-End

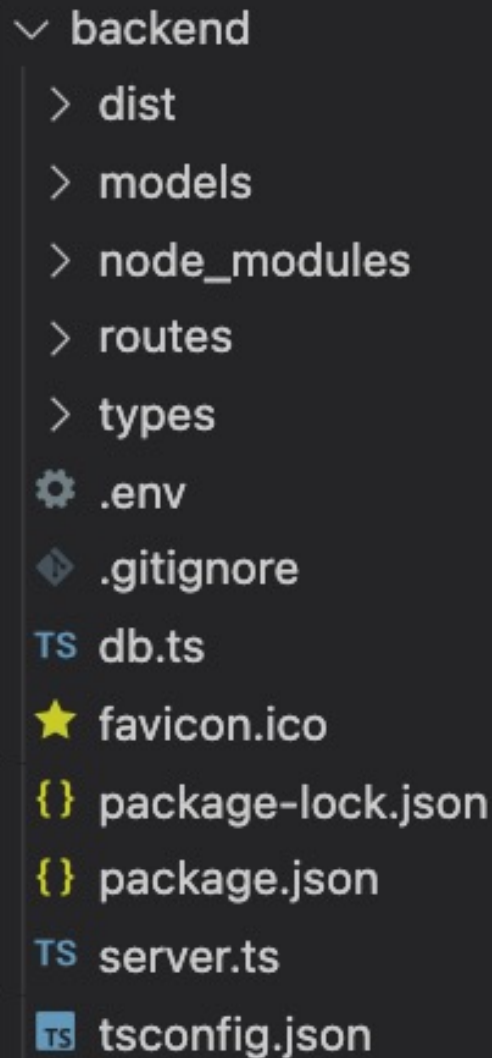
7. In fisierul tsconfig.json

```
"compilerOptions": {  
  "outDir": "./dist"  
}
```

8. In fisierul .env

**PORT=3002**

9. Facem fisierul: server.ts



# App – CRUD

## Back-End

### 10. Modificam fisieul package.json

```
"scripts": {  
    "build": "npx tsc",  
    "start": "node dist/server.js",  
    "dev": "concurrently \"npx tsc --watch\" \"nodemon -q  
dist/server.js\""  
}
```

### 11. npm run dev

Server is running at <https://localhost:3002>

# App – CRUD

DB: MySQL

1.

Metoda 1: <https://www.db4free.net>

Metoda 2: [Instalare MySql](#)

Metoda 3: [Instalare Xampp](#)

# App – CRUD

DB: MySQL

2. Configurare baza de date

3. Folosim baza de date de la Aplicatia1 sau  
Rulare fisier :curs7/mycrud/data/**db.sql**

4. Conectare la baza de date

# App – CRUD

## Back-End+MySQL

1. `npm install body-parser mysql2`
2. `npm install --save-dev @types/body-parser @types/mysql @types/dotenv`
3. In fisierul `.env`

**DB\_HOST="localhost"**

**DB\_USER="username"**

**DB\_PWD="password"**

**DB\_NAME="dbname"**

# App – CRUD

## Back-End+MySql

4. curs7/mycrud/backend/**db.ts**
5. curs7/mycrud/backend/**types**/User.ts
6. curs7/mycrud/backend/**models**/user.ts
7. curs7/mycrud/backend/**routes**/userRouter.ts
8. Modificam fisierul server.ts
9. npm run dev
10. <http://localhost:3002/users>

# App – CRUD

Front-end + Back-End+MySQL

1. Adaugam cereri Ajax in app.js
2. Afisam datele in index.html



# App – CRUD

## Front-end + Back-End+MySQL

Index	GET	/users	returns list of all items
Show	GET	/users/:id	returns item id
Create	Post	/users	creates a new item
Update	Put/Patch	/users/:id	Updated item ID
Destroy	Delete	/users/:id	Deletes item ID