2. Data Access

Adrian Adiaconitei

LINK Academy

Obiective

- ✓ Limbajul SQL
- ✓ Storage Engine
- ✓ Index / constrangeri
- ✓ DQL- SELECT

Recapitulare-Bazele de date

- ✓ Eficienţă:
  - -Scriem mai puţin cod.
- ✓ Siguranţă:
  - -Se păstrează integritatea datelor.
- ✓ Accesibilitatea:
  - -Asigură muncă simultană pentru mai mulţi utilizatori.

Recapitulare- Bazele de date

- ✓ Spaţiu pentru stocarea datelor.
- ✓ Ne preocupăm mai puţin de salvarea datelor.
- ✓ Pot lucra simultan mai mulţi utilizatori.
- ✓ Datele sunt stocate astfel încât să nu fie corupte.

Recapitulare

Softuri cunoscute de SGBD:

(SGBD - Sistem de Gestiune a Bazelor de Date DBMS - DataBase Management System)

- ✓ **Oracle**
- ✓ **MySql**
- ✓ **PostgreSql**
- ✓ **Microsoft Sql**
- ✓ **DB2 (IBM)**

Recapitulare

Structura informatiei dintr-un SGBD relational:

- ✓ **Baza de date**
- ✓ **Tabele** (relatii)
  - coloane(atribute)
  - randuri (inregistrari/entitati)

Recapitulare

MySQL – este un software open-source de tip SGBD relational, ce opereaza intr-o arhitectura **client - server**

- ✓ **Serverul MySQL –** gertioneaza basele de date stocate pe hard-disk sau in memorie. Serverul este reprezentat de **mysqld**
- ✓ **Client MySQL:**
  - Pentru manipularea structurii bazelor de date
    - **mysql** pentru linia de comanda
    - **mysqldump**: realizeaza backup
    - **mysqlimport:** importa date dintr-un fisier in baza de date
  - Pentru administrarea serverului
    - **mysqladmin**: monitorizeaza si administreaza serverul MySQL

Recapitulare

Aplicatii folosite pentru intreactiunea cu MySQL:

- ✓ **MySQL Workbench**
- ✓ PhpMyAdmin
- ✓ Linia de comanda : mysql

Limbajul SQL(Structured Query Language)

✓ **SQL -** Este un limbaj neprocedural si declarativ, orientat pe multimi. Limbaj de programare folosit pentru a lucra cu bazele de date relationale. Este creat de IBM, în anul 1970. Standarizat în anul 1986.

Conventii de notare :

- **majuscule pentru cuvintele rezervate**
- **litere mici pentru cuvinte definite de utilizator**
- **numele bazelor de date / tabele nu pot contine: . / **
- bara verticala | indica posibilitatea alegerii dintre mai multe variante
- acoladele { } indica un element necesar
- parantezele drepte [ ] indica un element optional

https://www.w3schools.com/sql/default.asp

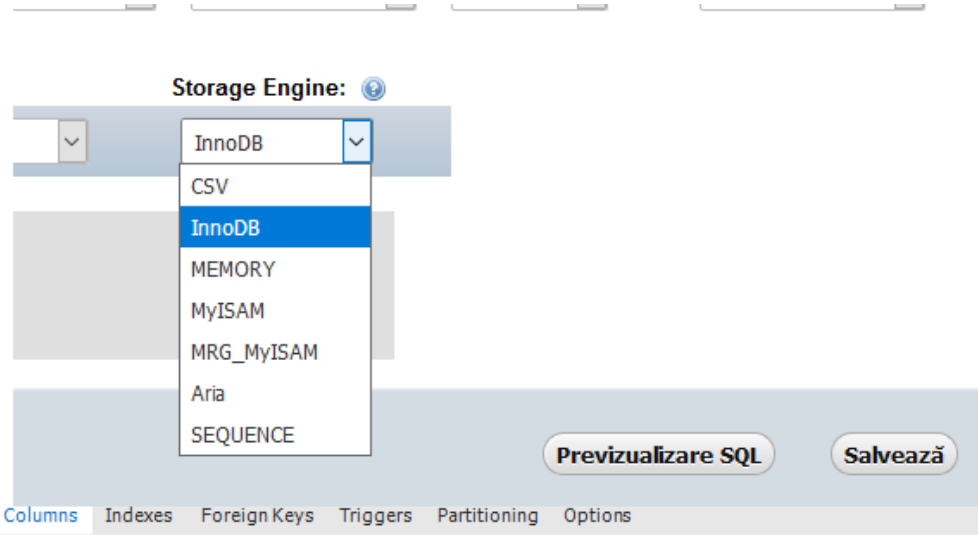
Limbajul SQL (MySQL)

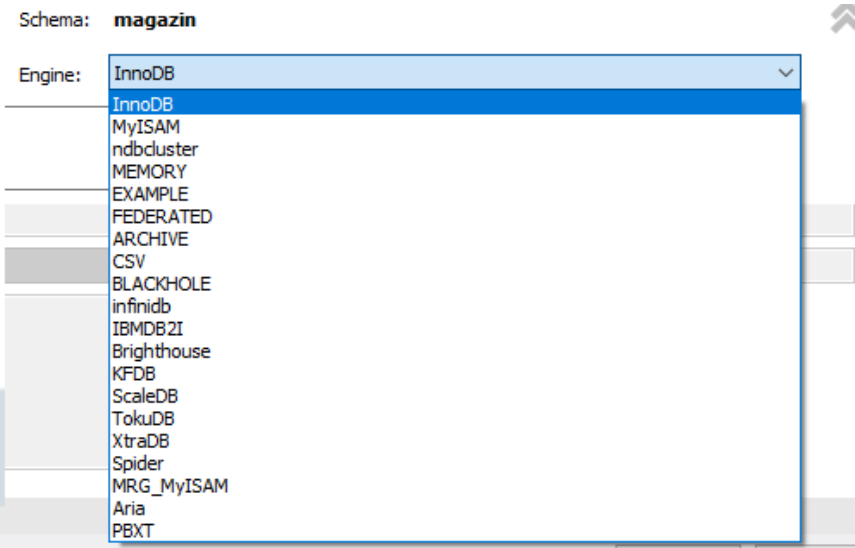
- ✓ **DDL (Data Definition Language)**: creare, modificare, stergere, vizualizare bade de date si tabele; CREATE, ALTER, DROP
- ✓ **DML (Data Manipulation Language)**: manipularea datelor din tabele : INSERT, UPDATE, DELETE
- ✓ **DQL – (Data Query Language):** permit obţinerea datelor din baza de date;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

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

DDL (Data Definition Language):

Storage Engine (Metoda de stocare) : Alegerea metodei de stocare potrivita are implicatii majore asupra performantei aplicatiei **(performantei lucrului cu datele)**





DDL - Storage Engine (Metoda de stocare) :

- ✓ MyISAM:
  - ✓ metoda de stocare implicita in MySql
  - ✓ performanta ridicata (resurse ocupate si viteza)
  - ✓ posibilitatea cautarii in intregul text (index FULLTEXT)
  - ✓ blocare acces la nivel de table
  - ✓ nu accepta tranzactii ; nu accepta FOREIGN KEY
  - ✓ probleme relative la integritatea datelor
- ✓ InnoDB
  - ✓ devine metoda de stocare implicita in MySql daca la instalare se alege model transactional
  - ✓ performanta medie (resurse ocupate si viteza)
  - ✓ blocare acces la nivel de linie ; incepand cu MySql 5.6.4 este introdus index FULLTEXT
  - ✓ accepta tranzactii , accepta FOREIGN KEY
  - ✓ probleme mai putine la integritatea datelor prin **constrangeri** intre tabele
- ✓ Memory
  - ✓ metoda de stocare recomandata pentru tabele temporare
  - ✓ performanta maxima (viteza – datele sunt stocate in RAM)
  - ✓ la oprirea server-ului datele se pierd, tabelul este pastrat dar va fi fara nici o linie
  - ✓ nu accepta tipuri de date mari (BLOB, TEXT) – maxim 255 octeti
  - ✓ nu accepta index FULLTEXT ; nu accepta tranzactii ; nu accepta FOREIGN KEY ;probleme relative la integritatea datelor

DDL (Data Definition Language):

Aplicatia1. Folosind MySQL Workbench/PhpMyadmin sa se creeze:

- un fisier .sql unde se vor salva comenzile
- o baza de date : **magazin**
- o tabela: **produse** (idprodus, numep, cantitate, idfirma, firma, adresafirma, modelp, stocp, pret, categoriep, stocp, descrierep, data_addp)
- sa se genereze Diagrama EER(extended entity–relationship)
- adaugati cateva inregistrari in tabela produse

DDL - ANOMALII:

- ✓ **Redundanta**: Redundanta reprezinta stocarea in mod nejustificata a unei aceleiasi informatii de mai multe ori in baza de date. Observam ca pentru fiecare produs este stocat numele si adresa firmei, desi ele sunt unic determinate de idfirma.
- ✓ **Anomalia de stergere**: La stergerea din relatie a unui produs al unui furnizor/firma se pierd automat si datele despre acesta.
- ✓ **Anomalia de actualizare**: In cazul actualizarii unei informatii redundante, se poate intampla ca operatia sa modifice unele aparitii ale acesteia iar altele sa ramana cu vechea valoare.
- ✓ **Anomalia de inserare**: Nu putem insera date despre o firma(numele si adresa sa) decat daca exista in stoc un produs furnizat de acesta.

Solutie: Modelul Diagrama EER(extended entity–relationship) , Constrangeri, Normalizare

DDL - CONSTRANGERI:

- ✓ Constrangerile de integritate reprezinta reguli pe care valorile continute intr-o tabela trebuie sa le respecte. Sunt verificate automat de SGBD atunci cand au loc operatii de modificare a continutului tabelelor (adaugare, stergere,update).

- ✓ Exista trei tipuri de constrangeri
  1. **de cheie** (cheie primara, cheie secundara): Cheia primara trebuie sa fie unica
  2. **de referinta**: O cheie externa trebuie sa fie ori **null** in intregime, ori sa corespunda unei valori a cheii primare asociate.
  3. **de entitate**: Atributele cheii primare trebuie sa fie diferite de valoarea null.

DDL - CONSTRANGERI:

TIPURI DE CONSTRANGERI

- ✓ NOT NULL: valorile nu pot fi nule
- ✓ PRIMARY KEY: defineste cheia primara a unei tabele (unica nu poate fi NULL)
- ✓ UNIQUE: defineste o alta cheie a tabelei (unica poate fi NULL, poate fi aplicata la una sau mai multe coloane)
- ✓ FOREIGN KEY: defineste o cheie straina (externa) in relatii dintre doua tabele (Ex. Stergerea in cascada din 2 tabele)
- ✓ CHECK: introduce o conditie (expresie logica)
  Ex. ALTER TABLE student ADD CHECK (grupa >=1 AND grupa <=4);

DDL - CONSTRANGERI:

Aplicatia 2: Creati o baza de date facultate, apoi o tabela cu numele student si campurile (idstudent,nume, prenume, grupa, email,data_add, status)

1. Constrangeri: idstudent : cheie primara, email unic, nume+prenume unic, grupa obligatorie si cu valori 1,2,3,4, status poate fi doar : admis, respins, neevaluat, cu valoarea default neevaluat.
2. Verificati structura tabelei
3. Inserati in modul visual date. Vizualizati datele
4. Golirea tabela (stergeti doar datele)
5. Stergeti tabela si baza de date

Recapitulare

1. **SGBD-ul oferă următoarele servicii de bază:**
   a. Arhivarea și dezarhivarea datelor
   b. Gestiunea accesului concurențial la date
   c. Limbaj de programare visual pentru accesul la date
   d. Scanarea datelor
   e. Optimizarea datelor
2. **Utilizați instrucțiunea _____ pentru a adăuga o coloană unui table(existent).**
   a. EDIT
   b. ALTER
   c. MODIFY

Recapitulare

1. **SGBD-ul oferă următoarele servicii de bază:**
   a. Arhivarea și dezarhivarea datelor
   b. Gestiunea accesului concurențial la date
   c. Limbaj de programare visual pentru accesul la date
   d. Scanarea datelor
   e. Optimizarea datelor

2. **Utilizați instrucțiunea _____ pentru a adăuga o coloană unui table(existent).**
   a. EDIT
   b. ALTER
   c. MODIFY

Recapitulare

3. **Restricția CHECK:**
    a. Validează valoarea unei coloane
    b. Reface datele în caz de eroare
    c. Asigură ordonarea datelor
    d. Împiedică redundanța datelor
    e. Asigură backup-ul bazei de date

4. **Restricția NOT NULL:**
    a. Permite indexarea tabelelor
    b. Împiedică folosirea valorilor nule
    c. Asigură integritatea datelor
    d. Împiedică redundanța datelor
    e. Asigură coerența datelor

Recapitulare

3. **Restricția CHECK:**
   a. Validează valoarea unei coloane
   b. Reface datele în caz de eroare
   c. Asigură ordonarea datelor
   d. Împiedică redundanța datelor
   e. Asigură backup-ul bazei de date

4. **Restricția NOT NULL:**
   a. Permite indexarea tabelelor
   b. Împiedică folosirea valorilor nule
   c. Asigură integritatea datelor
   d. Împiedică redundanța datelor
   e. Asigură coerența datelor

Recapitulare

5. Un tabel poate avea cel mult:

a. O singură cheie de index
b. O singură cheie externă
c. O singură cheie secundară
d. O singură cheie primară
e. O singură cheie unică

6. Coloanele care participă la o restricție de unicitate:

a. Pot avea valori nule
b. Nu pot avea valori nule
c. Trebuie să fie de tip numeric
d. Trebuie să fie de tip autonumber
e. Trebuie să fie de tip CHAR

Recapitulare

5. Un tabel poate avea cel mult:

a. O singură cheie de index
b. O singură cheie externă
c. O singură cheie secundară
d. O singură cheie primară
e. O singură cheie unică

6. Coloanele care participă la o restricție de unicitate:

a. Pot avea valori nule
b. Nu pot avea valori nule
c. Trebuie să fie de tip numeric
d. Trebuie să fie de tip autonumber
e. Trebuie să fie de tip CHAR

Recapitulare

7. O valoare nulă:

a. Este un spațiu liber

b. Este un șir vid

c. Tip de date speciala care nu e egală cu nimic altceva

d. Trebuie să fie de tip autonumber

e. Este valoarea zero

8.Ce este prescurtarea SQL?

a. Strong Question Language

b. Structured Query Language

c. Structured Question Language

Recapitulare

7. O valoare nulă:

a. Este un spațiu liber

b. Este un șir vid

c. Tip de date speciala care nu e egală cu nimic altceva

d. Trebuie să fie de tip autonumber

e. Este valoarea zero

8.Ce este prescurtarea SQL?

a. Strong Question Language

b. Structured Query Language

c. Structured Question Language

Recapitulare

9. Comanda DROP:

a. Face parte din limbajul de definire a datelor și șterge un obiect existent în baza de date

b. Face parte din limbajul de definire a datelor și șterge o înregistrare existentă în baza de date

c. Face parte din limbajul de manipulare a datelor și șterge un obiect existent în baza de date

10. Instrucțiunea ALTER TABLE permite:

a. Modificarea conținutului unui rând

b. Adăugarea unui rând

c. Adăugarea unei restricții

d. Modificarea interogării datelor din tabelă

e. Crearea unei tabele părinte

Recapitulare

9. Comanda DROP:

a. Face parte din limbajul de definire a datelor și șterge un obiect existent în baza de date

b. Face parte din limbajul de definire a datelor și șterge o înregistrare existentă în baza de date

c. Face parte din limbajul de manipulare a datelor și șterge un obiect existent în baza de date

10. Instrucțiunea ALTER TABLE permite:

a. Modificarea conținutului unui rând

b. Adăugarea unui rând

c. Adăugarea unei restricții

d. Modificarea interogării datelor din tabelă

e. Crearea unei tabele părinte

Recapitulare

11. Care este diferenta dintre tipurile de date FLOAT si DOUBLE?
12. Care este diferenta dintre tipurile de date CHAR si VARCHAR?
13. Care este diferenta dintre o cheie primara si o cheie de tip unic?
14. Care este diferenta dintre DROP si TRUNCATE?
15. Care este diferenta dintre CREATE DATABASE si CREATE SCHEMA

Recapitulare

11. Care este diferenta dintre tipurile de date FLOAT si DOUBLE?

Diferenta este ca DOUBLE ofera o precizie mai mare decat FLOAT.

FLOAT are 32 biți (4 octeți) cu precizie de 8 locuri. DOUBLE are 64 biți (8 octeți) cu precizie de 16 locuri.

Ambele au o opțiune suplimentară numită UNSIGNED si se definesc la fel in virgula mobila

- FLOAT(size,d)
- DOUBLE(size,d)

Avantajul la virgula mobila este ca intr-un numar mic de octeti se pot reprezenta numere intr-un interval foarte mare.

In schimb are dezavantajul ca rezultatele calculelor nu sunt intotdeaua exacte (asa cum iar iesi unui contabil cu creionul si hartia :)).

Deci, in aplicatiile in care este ceruta o anumita precizie, este bine ca atat in bazele de date cat si in program sa se evite tipurile in virgula mobila (float, double).

Decimal: numere in virgula-fixa

Recapitulare

12. Care este diferenta dintre tipurile de date CHAR si VARCHAR?

- **CHAR de la 0 la 255**
- **VARCHAR de la 0 la 65,535**
- **Modul de stocare a informatiei**

Value	CHAR(4)	Storage Required	VARCHAR(4)	Storage Required
''	' '	4 bytes	''	1 byte
'ab'	'ab '	4 bytes	'ab'	3 bytes
'abcd'	'abcd'	4 bytes	'abcd'	5 bytes
'abcdefgh'	'abcd'	4 bytes	'abcd'	5 bytes

Recapitulare

13. Care este diferenta dintre o cheie primara si o cheie de tip unic?

- **Cheia primară este o coloană care definește în mod unic un rând într-un tabel al unei baze de date relaționale. O tabelă poate avea cel mult o cheie primară. Cheia primară impune constrângerea implicită NOT NULL. Deci, o coloană care este definită ca cheia primară nu poate avea valori NULL în ea.**
- **Cheia unică este o singură coloană sau un set de coloane care pot identifica un rând într-un tabel. Deci, o cheie unică este constrânsă astfel încât nici două valori ale ei să nu fie egale. O proprietate importantă este faptul că cheile unice nu impun constrângerea NOT NULL. Deoarece NULL reprezintă lipsa unei valori, dacă două rânduri au NULL într-o coloană, atunci nu înseamnă că valorile sunt egale. Un tabel poate avea mai multe chei unice.**

Recapitulare

14. Care este diferenta dintre DROP si TRUNCATE?

DROP folosit atat pentru baza de date cat si pentru tabele, sterge structura si datele. DROP INDEX

TRUNCATE folosit doar pentru tabele , sterge/goleste datele, structura tabelului ne fiind afectata

15. Care este diferenta dintre CREATE DATABASE si CREATE SCHEMA

-nu este diferenta

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 utlizate;
- ✓ **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.

SELECT

- ✓ Este cea mai importanta interogare SQL.
- ✓ Intelegerea setarilor si utilizarea inteligenta a indecsilor stau la baza eficientei unei aplicatii
- ✓ E absolut necesara realizarea interogarii in asa fel incat datele returnate sa fie exact cele dorite (prelucrarea sa se realizeze pe server-ul MySql)
- ✓ Interogarea unei baze de date reprezinta un proces de selectie care restrange informatiile extrase din baza de date la acele linii care indeplinesc criteriile dorite.
- ✓ Instructiunea **SELECT** este principalul instrument pentru regasirea datelor dintr-o baza de date relationala, premtand selectarea a una sau mai multe campuri, dupa diferite criterii.

Aplicatia 3

Cu ajutorul fisierului universitate.sql, actualizati baza de date universitate.

1. Cu ajutorul interogarii SELECT, afisati data curenta si versiunea de MySql
2. Selectati toate liniile si toate coloanele din tabelul studenti.
3. Selectati doar numele cursurilor disponibile
4. Selectati doar numele si prenumele studentilor inscrisi la facultate
5. Afisati numarul de student din anul 1, 2, 3
6. Afisati valoarea totala a burselor din anul I;
7. Studentii care au bursa sa li se schimbe statusul in bursieri

Aplicatia 4

1.Redenumiti, la afisare, numele coloanei "nume" din tabelul profesori in "Nume Profesor

- folosind cuvantul cheie AS
- fara cuvantul cheie AS

2.Afisati numarul de studenti din fiecare an, in umatoarea forma:

An Nr_studenti

Aplicatia 5

1. listarea studentilor nascuti incepand cu anul 1996;
2. listarea studnetilor nascuti in anii 1993 si 1994. (AND si BETWEEN)
3. listati toti profesorii cu gradul I si II
4. listati cursurile din anul 2 semesteul 2
5. listati toti studentii care serbeaza ziua de onomastica de SF. ION (LIKE)
6. Listati toti studentii care s-au nascut in aceeasi zi. Ex. ‘09-21’
7. Listati toti studentii care nu au bursa (IS NULL) apoi cei care au bursa (IS NOT NULL)
8. Listati studentii care au bursa anuala mai mare de 4000

Aplicatia 6

1. Listati alfabetic dupa nume studentii
2. Listati alfabetic dupa nume si prenume studentii
3. Listati alfabetic dupa nume si prenume si descrescator dupa an studentii
4. Listati profesorii in ordine aleatoare. Repetati interogarea
5. Listati alfabetic primi 5 studenti cei mai tineri
6. Listati alfabetic urmatorii 5 studenti cei mai tineri

Aplicatia 7

1. Listati numarul studentilor din fiecare an (o singura interogare GROUP BY) :

anul nr_student

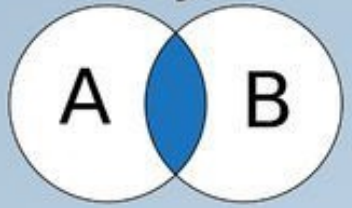
2. Listati numarul studentilor din fiecare an (o singura interogare GROUP BY) cu conditia sa fie mai multi de 9 studnti in fieacre an.

anul nr_studenti

SELECT - Interogari din tabele multiple

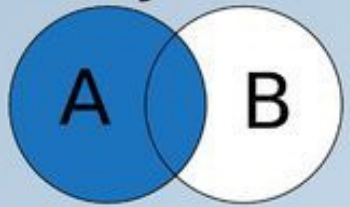
JOIN

INNER JOIN



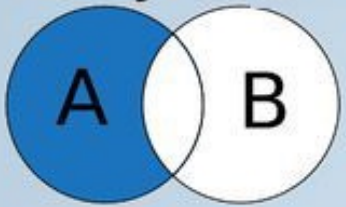

```
SELECT *
FROM A
INNER JOIN B ON A.key = B.key
```

LEFT JOIN




```
SELECT *
FROM A
LEFT JOIN B ON A.key = B.key
```

LEFT JOIN




```
SELECT *
FROM A
LEFT JOIN B ON A.key = B.key
WHERE B.key IS NULL
```

LINKAcademy

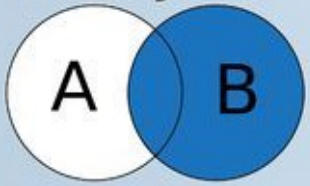
SELECT - JOIN

https://sqlzoo.net/

MySQL JOINS

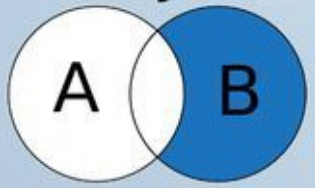
http://sql.sh/

RIGHT JOIN



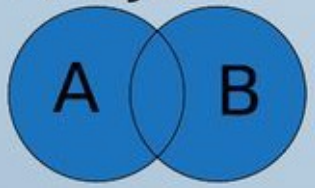

```
SELECT *
FROM A
RIGHT JOIN B ON A.key = B.key
```

RIGHT JOIN



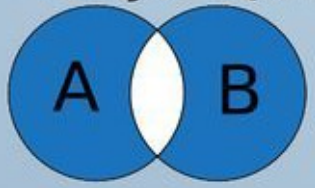

```
SELECT *
FROM A
RIGHT JOIN B ON A.key = B.key
WHERE B.key IS NULL
```

FULL JOIN




```
SELECT *
FROM A
FULL JOIN B ON A.key = B.key
```

FULL JOIN (sans intersection)




```
SELECT *
FROM A
FULL JOIN B ON A.key = B.key
WHERE A.key IS NULL
OR B.key IS NULL
```

SELECT - Interogari din tabele multiple

Aplicatia 8

Executati urmatoarele interogari:

1. SELECT c.id_curs, c.titlu_curs, n.valoare FROM `cursuri` c NATURAL JOIN note n;
2. SELECT * FROM student CROSS JOIN note;
3. SELECT * FROM student JOIN note;
4. SELECT * FROM student INNER JOIN note;
5. SELECT * FROM student , note;
6. SELECT * FROM student CROSS JOIN note WHERE prenume='Andrei';
7. SELECT * FROM student CROSS JOIN note CROSS JOIN cursuri;
8. SELECT * FROM cursuri INNRT JOIN note USING (id_curs)
9. SELECT * FROM cursuri c INNER JOIN note n on c.id_curs= n.id_curs

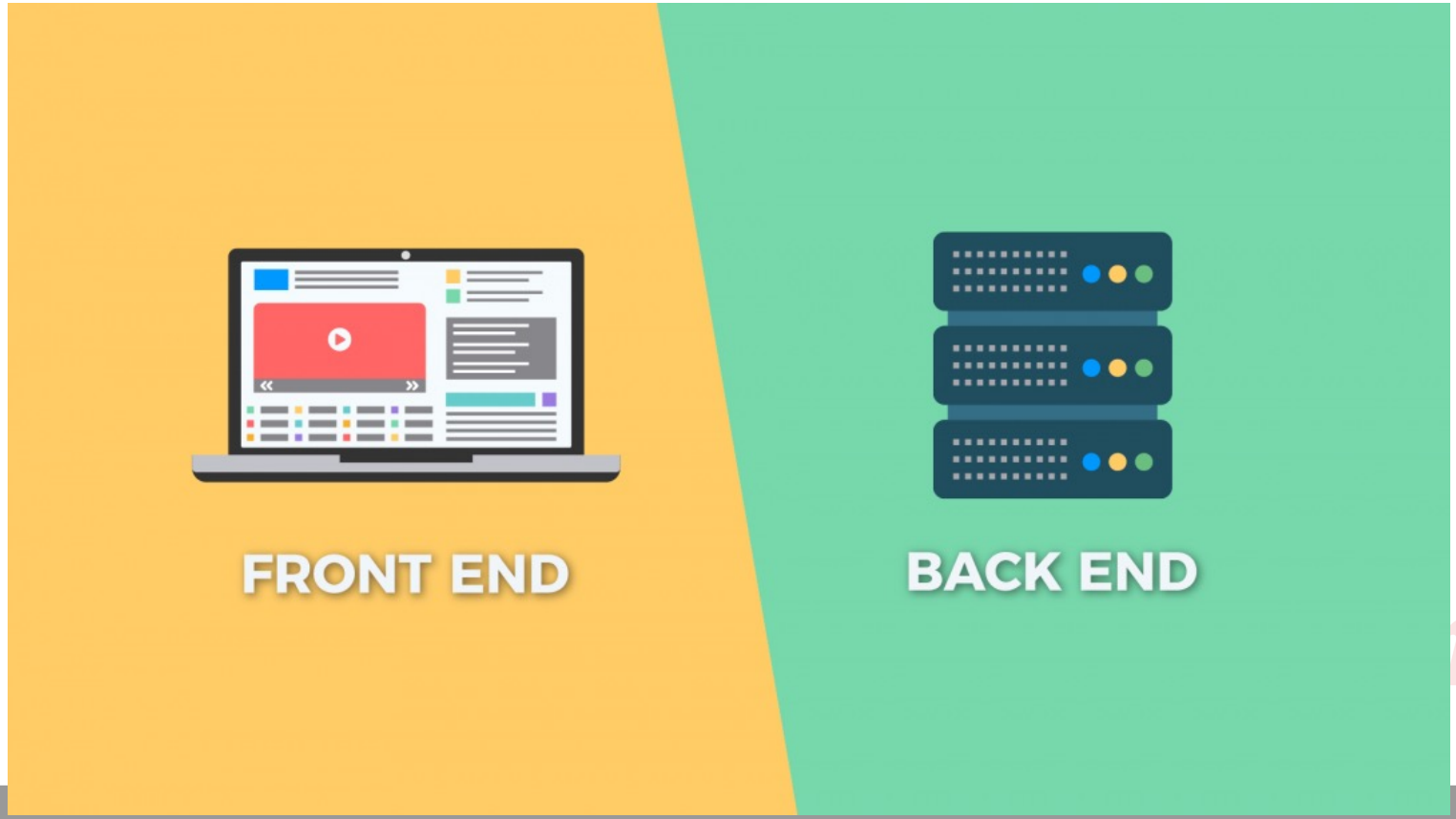
SELECT - Interogari din tabele multiple

Aplicatia 9

1. Listati ce note a pus un anumit professor la un anumit curs
2. Cate cursuri sunt fara profesor?
3. Cati studenti nu s-au inscris la nici un curs?
4. Cati studenti au minim 2 note/ sunt inscrisi la minim 2 cursuri
5. Cati strudenti si profesori isi serbeaza ziua de Sf. ION?
6. Pentru un anumit student doresc sa vad ce note are la cursurile sale
7. Calculati media notelor primite pentru un anumit student

App – CRUD

App – CRUD



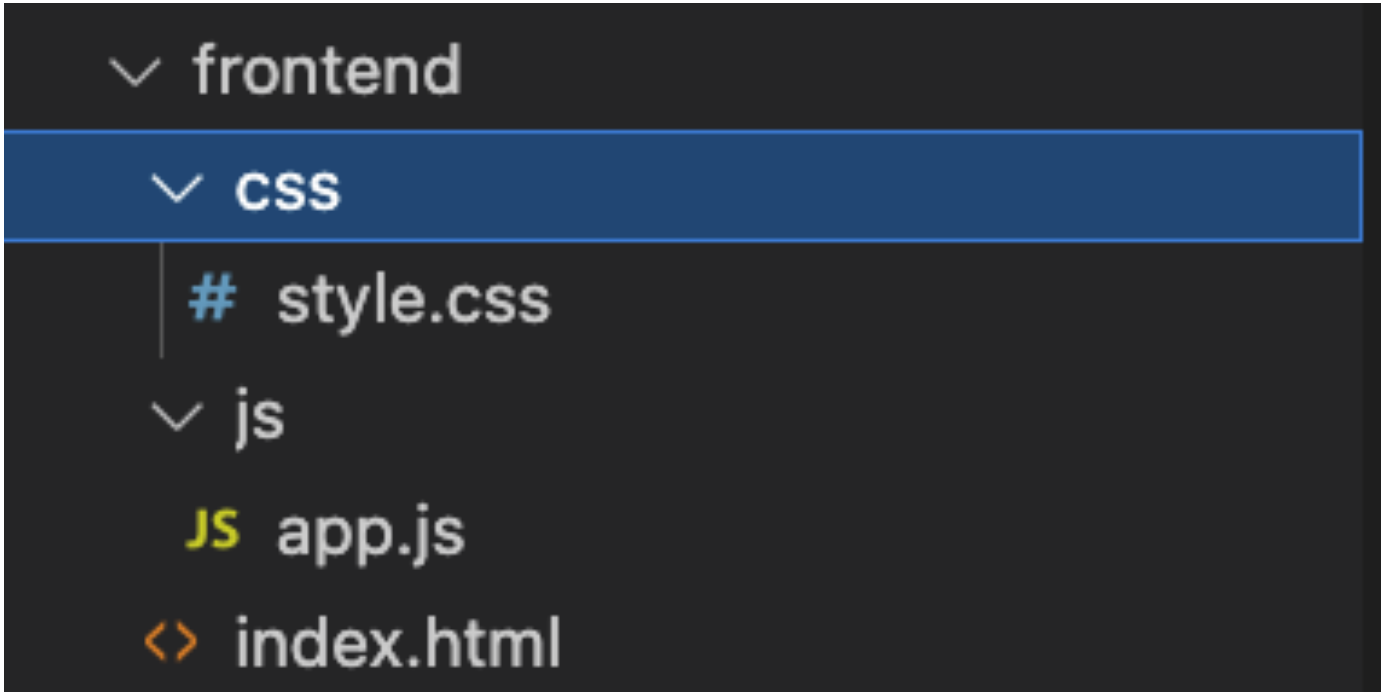
LINKAcademy

App – CRUD

Front-End

App – CRUD

Front-End



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": {

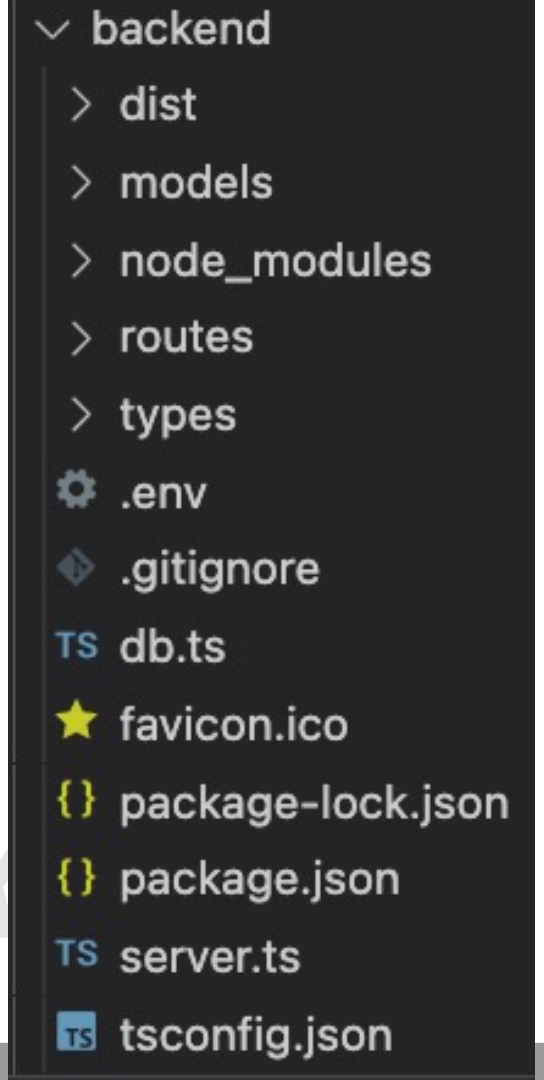
"outDir": ”./dist"

}

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

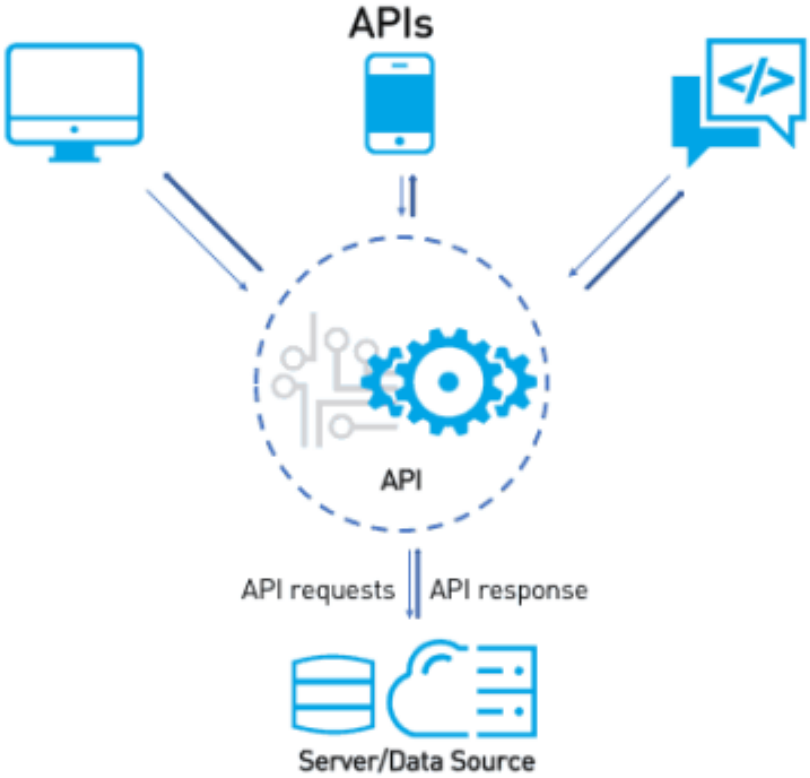
Front-end + Back-End+MySql

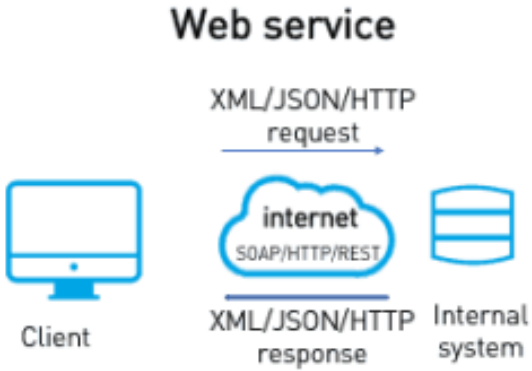
1. Adaugam cereri Ajax in app.js

2. Afisam datele in index.html

App – CRUD

Front-end + Back-End+MySql





AJAX

Browserele protejează utilizatorul de diferite atacuri şi, în caz că nu există un CORS header din partea serverului, browserul va crede că este vorba de o fraudă şi nu va executa AJAX.

AJAX – Suport din partea serverului

```
// CORS
app.use((req, res, next) => {
    res.append('Access-Control-Allow-Origin', ['*']);
    res.append('Access-Control-Allow-Methods',
    'GET,PUT,POST,DELETE');
    res.append('Access-Control-Allow-Headers', 'Content-
    Type');
     next();
});
```

LINKAcademy

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