

Proiect Baze de Date

(Baza de date pentru scoala)

- Student: Maxim Claudiu Stefan
- Grupa: 1309B
- Coordonator: Mironeanu Catalin
- Git: https://github.com/maximclaudiu/Baze_Date

Aplicatia are ca scop stocarea si organizarea datelor unei scoli/liceu. Aceasta pastreaza date atat despre angajati, cat si despre elevii respectivei institutii.

Prin parcurgerea tabelelor, putem afla date despre angajati, cum ar fi orasul din care provin, titlul locului de munca pe care il ocupa, varsta si multe altele. Despre elevi se pot afla varsta, clasa din care fac parte, orasul de provenienta, dar si notele pe care acestia le primesc la diverse discipline.

Proiectul prezinta o fundatie al unui concept ce poate fi extins pentru stocarea si manipularea mai multor informatii referitoare la institutia dorita (cartile din biblioteca, echipamentele tehnice, mai multe informatii despre angajati si elevi etc.).

Aplicatia poate fi accesata oricand urmarind link-ul :[Proiect BD](#)

Note: aplicatia este hostata remote pe un server, aceasta se opreste din rulare daca nu este folosita, drept urmare la prima deschidere, incarcarea dureaza pana intr-un minut. Din aceleasi motive, indexarile autoincrementate in tabel cresc din 10 in 10.

Tehnologii folosite:

1. Pentru interfata:

a) Html:

- i. Crearea navigatiei si schimbarea paginilor.
- ii. Implementarea formularelor de adaugare in tabele.
- iii. Afisarea datelor pe ecran sub forma de tabele.
- iv. Prezentarea optiunilor de manipulare a datelor intr-un mod cat mai simplist si usor de inteles.

b) CSS:

- i. Stilizarea paginilor web create.
- ii. Confera interfetei un aspect placut si usor de navigat.

c) JavaScript:

- i. Usureaza munca vizitatorului de a accesa/adauga/sterge informatii in tabelele dorite.

2. Pentru baza de date:

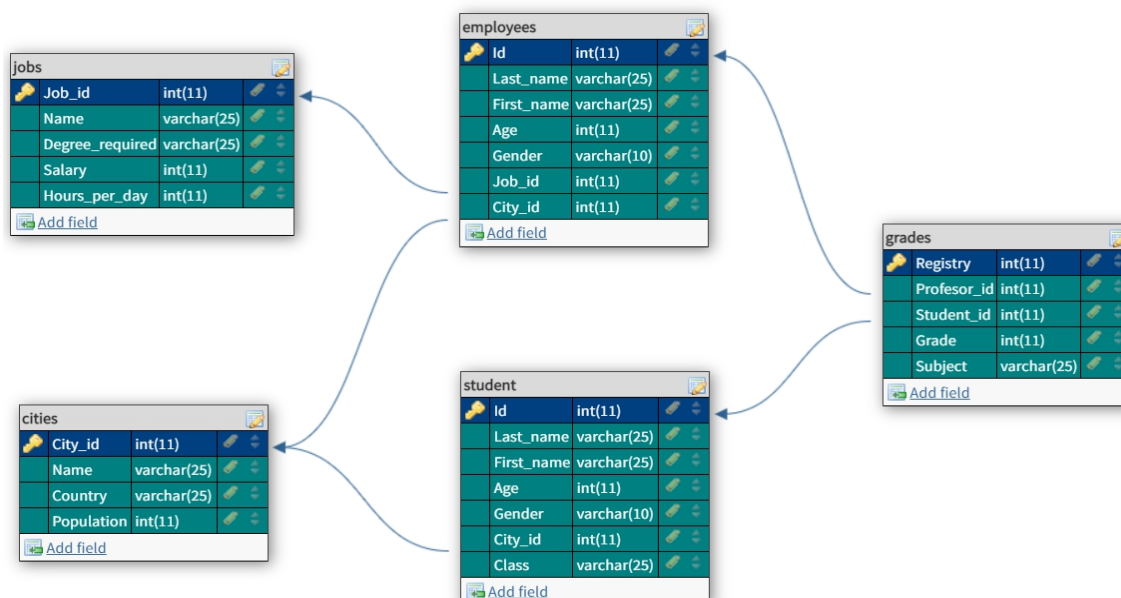
a) MySql:

- i. Crearea bazei de date si a tabelelor acesteia.
- ii. Executare de cod SQL pentru operatiunile de vizualizare, adaugare si stergere.

b) PHP:

- i. Crearea unei conexiuni la baza de date folosita.
- ii. Legatura intre butoanele din interfata si comenzile SQL rulate in fundal.

Structura si inter-relationarea tabelelor:



1. Prima forma normala:
 - a) Nu contine mai multe valori pe un camp(ex: "Nume: Popescu, Vlad" -> "Nume: Popescu; Prenume: Vlad").
 - b) Datele de pe o coloana sunt de acelasi tip (ex: int, data, varchar).
 - c) Coloanele au nume unice.
2. A doua forma normala:
 - a) Se afla in prima forma normala.
 - b) Elementele tabelor sunt dependente doar de cheia primara.

Constrangeri:

- Coloanele Job_id, City_id, Id (employees), Id (student) si Registry sunt primary key-uri in tabelele lor.
- City_id este foreign key, atat in tabela employees, cat si in tabela student (one-to-many).
- Job_id este foreign key in tabela employees (one-to-many).
- Id (employees) si Id (student) sunt foreign key-uri in tabela grades (Profesor_id, Student_id) (many-to-many).

Conectarea la baza de date:

Conectarea a fost realizata prin intermediul unei functii php.

Ex: \$mysqli= new mysqli (\$host, \$username, \$user_pass, \$database);

Conexiunea este stocata in variabila \$mysqli, prin intermediul careia se vor executa comenzile SQL la alegerea utilizatorului.

ID	LAST NAME	FIRST NAME	AGE	GENDER	CITY	JOB
1	Mihalache	Andrei	45	Male	Constanta	Bibliotecar
11	Mihalache	Miruna	41	Female	Constanta	Administrator
21	Cazacu	Razvan	29	Male	Iasi	Profesor
31	Andronescu	Catalin	51	Male	Vaslui	Gardian
41	Moisei	Eliza	39	Female	Iasi	Director
51	Caruntu	Teodor	43	Male	Suceava	Profesor
61	Dragomir	Maria	37	Female	Cluj	Ingrijitor
71	Tanase	Andrei	26	Female	Bucuresti	Invatator
81	Gradinaru	Mihai	29	Male	Bucuresti	Invatator
91	Manole	Serban	35	Male	Iasi	Profesor
101	Manole	Alexandra	33	Female	Iasi	Profesor

Fig 1. Vizualizarea datelor din tabela.

First name:

Last name:

Age:

Gender:

From City:

Job:

1
11
21
31
41
51
61

Fig 2. Adaugarea unei inregistrari in tabela si demonstrarea selectiei unei chei straine.

Exemple cod vizualizare:

```

1. $sql = "SELECT * FROM ". basename($_SERVER["REQUEST_URI"], ".php");
2. $result= $mysqli->query($sql);
3. if ($result->num_rows>0){
4.     while ($row = $result->fetch_assoc()) {
5.         echo " <tr><td>" . $row["Id"]. "</td>
           <td>" . $row["Last_name"]. "</td>
           <td>" . $row["First_name"]. "</td>
           <td>" . $row["Age"]. "</td>
           <td>" . $row["Gender"]. "</td>
           <td>" . $row2["Name"]. "</td>
           <td>" . $row3["Name"]. "</td> </tr>";
6.     }
7. }

```

- a) 1. crearea comenzii sql pentru selectarea datelor din tabela. Paginile web au aceeași denumire cu tabelele din baza, `basename($_SERVER["REQUEST_URI"], ".php")` returnează acest nume fără '.php' de la final, pentru a obține o comandă sql validă
- b) 2. rularea scriptului prin intermediul conexiunii php.
- c) 3. dacă numărul de rezultate obținute este mai mare de 0 (zero).
- d) 4. variabila `$row` păstrează, sub forma unui array, rezultatele de pe o linie din tabela. (ex: `echo $row["Id"];`)
- e) 5. returnăm un string ce va fi interpretat de html ca fiind o nouă linie în tabela de afișare, unde `"$row["X"]"` este valoarea aflată în tabela pe coloana "X", la linia curentă (din 3 și 4: cât timp mai există date în result, mai bine spus, nu au fost afișate toate liniile din tabela, `$row` primește alt array de valori specific unei anumite linii din tabela).

Exemplu cod stergere:

- 1. `$sql = "DELETE FROM ". $_POST['page']." WHERE Id = '". $_POST['delete'] .'"";`
- 2. `$result= $mysqli->query($sql);`
 - a) 1. Crearea comenzii de stergere a unei înregistrări din tabela, specifică id-ului primit prin `"$_POST"`.
 - b) 2. executarea comenzii sql definită precedent prin intermediul conexiunii php.