

Aplicație pentru evidența editurilor dintr-o librărie

1.Descrierea cerinței:

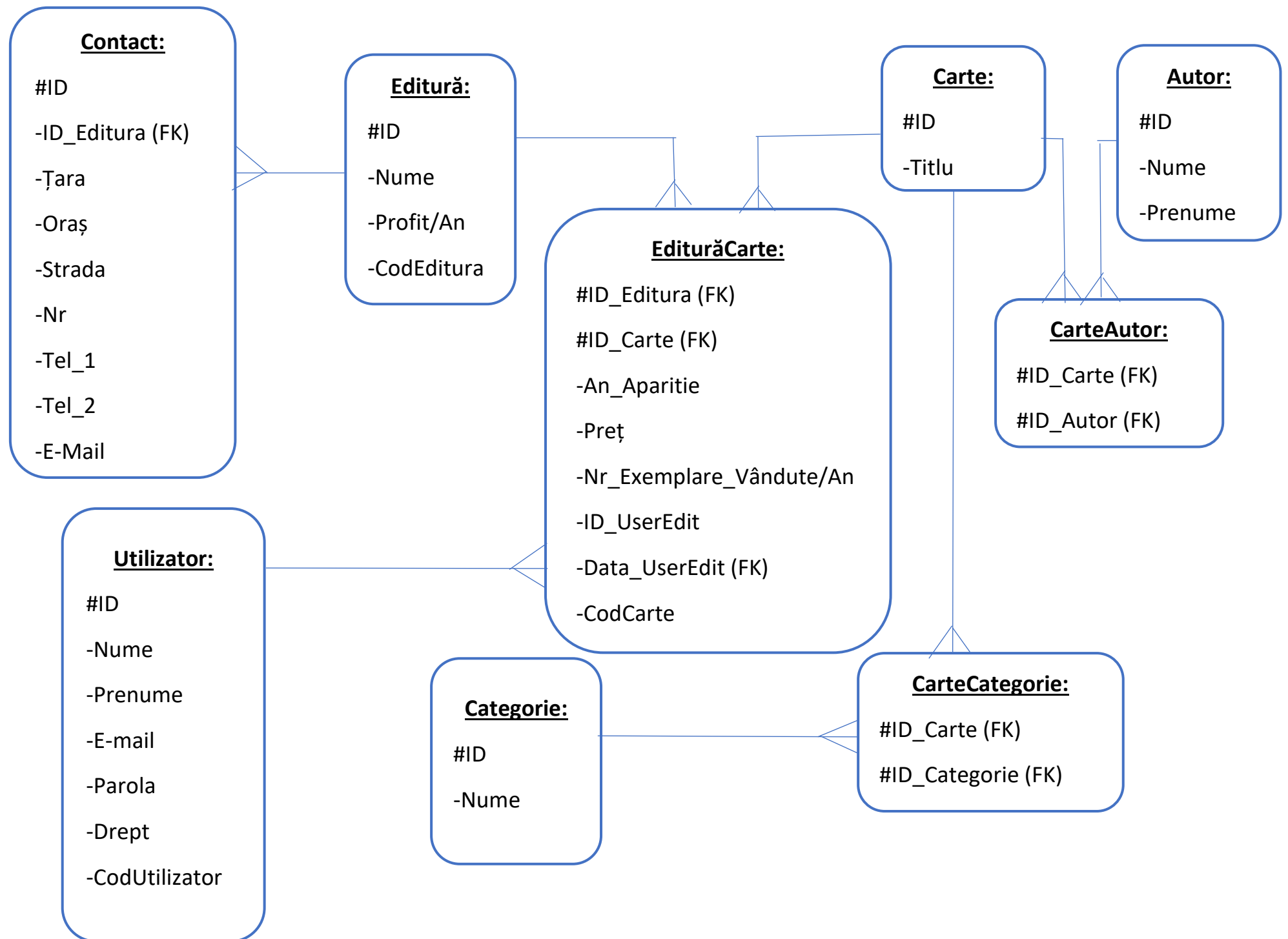
O bibliotecă dorește să țină evidența editurilor pe care le are disponibile. Fiecare editură are un set de cărți pe care le pune la ofertă în librăria respectivă. De asemenea, pentru o editură se cunoaște profitul pe care îl obține în fiecare an în urma vânzărilor sale, dar și un set de contacte (adresă, e-mail, telefon). O carte are un titlu, unul sau mai mulți autori și poate aparține uneia sau mai multor categorii (literatură românească, SF, publicistică...). Trebuie ținut cont de faptul că un titlu se poate regăsi la mai multe edituri (de exemplu, un volum de poezii de Eminescu se poate regăsi la foarte multe edituri), pentru fiecare versiune cunoscându-se un an al apariției, prețul și numărul de exemplare vândute într-un an.

Cu privire la utilizatori, fiecare are nume, prenume, un e-mail, o parolă, un cod unic și un tip de drepturi (admin/user).

În cadrul aplicației se consideră că utilizatorul trebuie să poată căuta cărți după diverse criterii (editura, dar și titlu, numele autorului, anul apariției...) și să primească drept rezultat un tabel pe care să-l poată sorta după coloanele sale. Mai mult, pentru fiecare element din tabelul afișat vor putea fi disponibile opțiunile de editare/ștergere (numai utilizatorul cu drept de admin va avea dreptul să execute operațiile). Utilizatorul cu drept de admin va putea mai mult să adauge alte cărți. Similar, se pot căuta și edita înregistrările cu privire la edituri sau chiar la autori. De asemenea, va exista și un tab pentru evidența utilizatorilor, însă acolo va avea acces doar utilizatorul admin, deoarece doar acesta ar trebui să aibă dreptul de a vedea toți utilizatorii aplicației (și să-i și gestioneze).

2.Etapa de proiectare:

Mai jos am realizat schema bazei de date, în care am ilustrat tabelele, câmpurile și relațiile între acestea:



3.Constrângerile de integritate:

Pentru a reduce numărul de interogări la ștergere, am utilizat mai multe constrângeri de integritate pentru cheile străine din tabelele mele:

- *ID_UserEdit* din **EditurăCarte** este cheie străină care face referință la tabelul **Utilizator**. În momentul în care șterg un utilizator din baza de date, în **EditurăCarte** va apărea NULL în câmpul *ID_UserEdit*, deoarece am setat o constrângere de tip set null;
- *ID_Editura* din **Contact** este cheie străină care face referință la tabelul **Editură**. În momentul în care șterg o editură, trebuie să dispară și setul său de contacte din baza de date, deci am folosit o constrângere de tip cascade;
- *ID_Editura* din **EditurăCarte** este cheie străină care face referință la tabelul **Editură**. În momentul în care șterg o editură, trebuie să dispară și cărțile publicate de aceasta din baza de date, deci am folosit o constrângere de tip cascade;
- *ID_Carte* din **EditurăCarte** este cheie străină care face referință la tabelul **Carte**. În momentul în care șterg un titlu, trebuie să dispară și versiunile sale de la diverse edituri din baza de date, deci am folosit o constrângere de tip cascade;
- *ID_Carte* din **CarteCategorie** este cheie străină care face referință la tabelul **Carte**. În momentul în care șterg un titlu, trebuie să dispară și legăturile cu categoriile din baza de date, deci am folosit o constrângere de tip cascade;
- *ID_Autor* din **CarteAutor** este o cheie străină care face referință la tabelul **Autor**. În momentul în care se șterge un autor, trebuie să dispară și cărțile acestuia, deci implicit și înregistrările corespunzătoare din tabelul de legătură, motiv pentru care am folosit constrângere de tip cascade;
- *ID_Carte* din **CarteAutor** este o cheie străină care face referință la tabelul **Carte**. În momentul în care se șterge un titlu, trebuie să dispară înregistrările corespunzătoare din tabelul de legătură, motiv pentru care am folosit constrângere de tip cascade;

4.Functionarea aplicatiei:

Aplicația începe cu o pagină de LogIn, unde utilizatorul își scrie credențialele. Dacă sunt corecte, acesta intră în aplicația propriu-zisă. În toată aplicația este prezentă bara de meniu cu toate taburile în care are acces: Stoc de cărți, Adăugare de carte, Gestiune de autori, Gestiune de utilizatori, Gestiune de edituri.

Stoc de cărți: aici poate căuta o carte după unul sau mai multe criterii (până la 5 criterii) diferite: Titlu, Preț, Anul apariției, Editură, Cod ISBN, Autor, Categorie, Numărul de exemplare vândute în ultimul an. Dacă face o căutare cu search-bar-ul gol, va primi toate înregistrările din baza de date. Tabelul returnat conține titlul cărții, editura, autorul (autorii), categoria (categoriile), prețul, numărul de exemplare vândute în ultimul an, anul publicării. Pentru ușurința citirii, tabelul vine în plus și cu funcția de paginare (paginarea se asigură din interogarea bazei de date). Deasupra capului de tabel se poate selecta câte înregistrări să fie pe pagină. Sub tabel, lângă săgețile pentru parcurgerea paginilor este un dropdown pentru saltul la o anumită pagină (numărul total de înregistrări este calculat în prealabil pentru a determina numărul de pagini). Fiecare înregistrare este însoțită și de un buton pentru detalii. Acesta deschide o fereastră pop-up care conține pe larg toate detaliile unei cărți: atât ce este enumerat în tabel (desfășurat în mod clar), cât și codul ISBN, numele utilizatorului care a introdus cartea respectivă în baza de date și data acestei proceduri. Mai mult, în același pop-up se găsesc și butoanele pentru editare/ștergere (la acestea are acces doar utilizatorul admin).

Adaugă o carte: Aici se găsește o formă pentru adăugarea unei noi cărți în baza de date.

Gestionează autorii: Metoda de căutare și afișare este similară ca în Stoc de cărți, doar că nu există decât un singur criteriu de căutare (Numele), iar înregistrările nu mai vin cu buton pentru detalii, ci au direct cele 2 butoane pentru ștergere și editare. În plus, pentru adăugare, butonul se află pe aceeași pagină. Adăugarea, editarea și ștergerea se pot face doar de către utilizatorul admin.

Gestionează edituri: Căutarea unei anumite edituri se poate face după nume, profit anual sau cod. Rezultatul conține numele, profitul și numărul de titluri care se regăsesc în baza de date. La fel ca în Stoc de cărți, înregistrările vin cu un buton

pentru detalii. Detaliile cuprind numele, codul, profitul, numărul de cărți din baza de date, dar și punctele de contact, împreună cu butoanele de ștergere și editare. În aceeași pagină se găsește și butonul de adăugare a unei edituri.

Gestionează utilizatorii: În această pagină are acces doar utilizatorul admin. Aici se pot vedea toți utilizatorii care au acces la aplicație, e-mail-urile acestora și tipul de drept pe care îl au. Căutarea se poate face după nume sau după drept. Fiecare înregistrare vine cu butoanele de ștergere și editare. Pe aceeași pagină se găsește și butonul pentru adăugarea unui nou utilizator.

5.Interogările folosite:

INSERT:

- 1) INSERT INTO carti (Titlu) VALUES(?); ([Stoc_Add.php](#))
- 2) INSERT INTO carte_autor VALUES((SELECT MAX(ID_Carte) FROM carti), ?);([Stoc_Add.php](#))
- 3) INSERT INTO carte_categorie VALUES((SELECT MAX(ID_Carte) FROM carti), ?); ([Stoc_Add.php](#))
- 4) INSERT INTO editura_carte
VALUES(?, (SELECT MAX(ID_Carte) FROM carti), \$_SESSION["ID_User"], 'date('Y-m-d')', ?, ?, ?, ?);
([Stoc_Add.php](#))
- 5) INSERT INTO editura_carte
VALUES(?, (SELECT ID_Carte FROM carti WHERE Titlu=?), \$_SESSION["ID_User"], 'date('Y-m-d')', ?,
?, ?, ?);([Stoc_Add.php](#))

- 6) INSERT INTO autori(Nume, Prenume) VALUES(?, ?); ([Autori_Add.php](#))
- 7) INSERT INTO edituri(Nume, Profit_An, Cod_Editura) VALUES(?, ?, ?); ([Edituri_Add.php](#))
- 8) INSERT INTO contacte(ID_Editura, Tara, Oras, Strada, Nr, Tel_1, Tel_2, E_mail)
VALUES((SELECT MAX(ID_Editura) FROM edituri), ?, ?, ?, ?, ?, ?, ?); ([Edituri_Add.php](#))
- 9) INSERT INTO utilizatori(Nume, Prenume, E_mail, Drept, Parola, CodUtilizator)
VALUES(?, ?, ?, ?, ?, uuid()); ([Utilizatori_Add.php](#))

UPDATE:

- 1) UPDATE editura_carte SET Cod_Carte=? WHERE Cod_Carte=?; ([Stoc_Edit.php](#))
- 2) UPDATE carti SET Titlu=? WHERE ID_Carte=(SELECT ID_Carte FROM editura_carte WHERE Cod_Carte=?);
([Stoc_Edit.php](#))
- 3) UPDATE carte_autor SET ID_Autor=?
WHERE ID_Carte=(SELECT ID_Carte FROM editura_carte WHERE Cod_Carte=?); ([Stoc_Edit.php](#))
- 4) UPDATE carte_categorii SET ID_Categorie=?
WHERE ID_Carte=(SELECT ID_Carte FROM editura_carte WHERE Cod_Carte=?); ([Stoc_Edit.php](#))

- 5) UPDATE editura_carte SET ID_Editura=?, ID_UserEdit=\$_SESSION["ID_User"] , Data_UserEdit='date('Y-m-d')', An_Aparitie=?, Pret=?, ExemplareVandute_An=? WHERE Cod_Carte=?; (Stoc_Edit.php)
- 6) UPDATE autori SET Nume=?, Prenume=? WHERE ID_Autor=?;(Autori_Edit.php)
- 7) UPDATE edituri SET Cod_Editura=? WHERE Cod_Editura=?; (Edituri_Edit.php)
- 8) UPDATE edituri SET Nume=?, Profit_An=? WHERE Cod_Editura=?;(Edituri_Edit.php)
- 9) UPDATE contacte SET Tara=?, Oras=?, Strada=?, Nr=?, Tel_1=?, Tel_2=?, E_mail=?
WHERE ID_Editura=(SELECT ID_Editura FROM edituri WHERE Cod_Editura=?); (Edituri_Edit.php)
- 10) UPDATE utilizatori SET Nume=?, Prenume=?, E_mail=?, Drept=? WHERE ID_Utilizator=?; (Utilizatori_Edit.php)

DELETE:

- 1) DELETE FROM carti WHERE ID_Carte=(SELECT ID_Carte FROM editura_carte WHERE Cod_Carte=?);
(Stoc_Sterge.php)
- 2) DELETE FROM carti WHERE ID_Carte IN (SELECT ID_Carte FROM carte_autor WHERE ID_Autor=?);
(Autori_Sterge.php)
- 3) DELETE FROM autori WHERE ID_Autor=?; (Autori_Sterge.php)

- 4) DELETE FROM carti
WHERE ID_Carte NOT IN (SELECT ID_Carte FROM editura_carte WHERE ID_Editura !=?);
([Edituri_Sterge.php](#))
- 5) DELETE FROM edituri WHERE ID_Editura=?; ([Edituri_Sterge.php](#))
- 6) DELETE FROM utilizatori WHERE ID_Utilizator=?; ([Utilizatori_Sterge.php](#))

SELECT-uri simple:

- 1) SELECT `E_mail`, Parola, Drept, CodUtilizator, ID_Utilizator, CONCAT(Prenume, ' ', Nume) AS Nume FROM utilizatori WHERE `E_mail`=?; ([LogIn.php](#))
- 2) SELECT ID_Editura, Nume FROM edituri; ([Scripturi.php](#))
- 3) SELECT * FROM categorii; ([Scripturi.php](#))
- 4) SELECT ID_Autor, CONCAT(Prenume, ' ', Nume) AS Nume FROM autori; ([Scripturi.php](#))

- 5) SELECT Titlu, edituri.Nume AS Nume_Editura, An_Aparitie, GROUP_CONCAT(DISTINCT categorii.Nume) AS Nume_Categorie, Pret, ExemplareVandute_An, GROUP_CONCAT(DISTINCT CONCAT(autori.Prenume, ' ', autori.Nume)) AS Nume_Autor, Cod_Carte, CONCAT(utilizatori.Prenume, ' ', utilizatori.Nume) AS Nume_User, Data_UserEdit
FROM editura_carte
INNER JOIN carti ON editura_carte.ID_Carte=carti.ID_Carte
INNER JOIN edituri ON editura_carte.ID_Editura=edituri.ID_Editura
INNER JOIN carte_categorii ON editura_carte.ID_Carte=carte_categorii.ID_Carte
INNER JOIN categorii ON carte_categorii.ID_Categorie=categorii.ID_Categorie
INNER JOIN carte_autor ON carte_autor.ID_Carte=editura_carte.ID_Carte
INNER JOIN autori ON autori.ID_Autor=carte_autor.ID_Autor
LEFT JOIN utilizatori ON editura_carte.ID_UserEdit=utilizatori.ID_Utilizator
WHERE Cod_Carte=? GROUP BY Titlu"; ([Stoc_Detalii.php](#))
- 6) SELECT ID_Carte FROM editura_carte WHERE Cod_Carte=? AND Cod_Carte!=?; ([Stoc_Edit.php](#))
- 7) SELECT Titlu, ID_Editura, An_Aparitie, ExemplareVandute_An, Pret, Cod_Carte
FROM editura_carte INNER JOIN carti ON editura_carte.ID_Carte=carti.ID_Carte
WHERE Cod_Carte=?; ([Stoc_Edit.php](#))
- 8) SELECT Titlu, edituri.Nume AS Nume_Editura, An_Aparitie, GROUP_CONCAT(DISTINCT categorii.Nume) AS Nume_Categorie, Pret, ExemplareVandute_An, GROUP_CONCAT(DISTINCT CONCAT(autori.Prenume, ' ', autori.Nume)) AS Nume_Autor, Cod_Carte \$query_0 \$sort LIMIT \$e_p OFFSET \$offset; ([Stoc_Result.php](#))

- **\$query_0**, cuprinde clauzele FROM, INNER JOIN, dar și WHERE, unde condițiile sunt create dinamic pe baza datelor venite de la căutare (câte criterii și de ce tip)

= FROM editura_carte

INNER JOIN carti ON editura_carte.ID_Carte=carti.ID_Carte

INNER JOIN edituri ON editura_carte.ID_Editura=edituri.ID_Editura

INNER JOIN carte_categorii ON editura_carte.ID_Carte=carte_categorii.ID_Carte

INNER JOIN categorii ON carte_categorii.ID_Categorie=categorii.ID_Categorie

INNER JOIN carte_autor ON carte_autor.ID_Carte=editura_carte.ID_Carte

INNER JOIN autori ON autori.ID_Autor=carte_autor.ID_Autor

WHERE

- **\$sort**, cuprinde o clauză ORDER BY, construită dinamic pe baza criteriilor de sortare.

9) SELECT COUNT(Cod_Carte) FROM editura_carte WHERE Cod_Carte=?; ([Stoc_Sterge.php](#))

10) SELECT ID_Autor, Prenume, Nume FROM autori WHERE ID_Autor=?; ([Autori_Edit.php](#))

11) SELECT COUNT(ID_Carte) AS NrTitluri, CONCAT(autori.Prenume, ' ', autori.Nume) AS Nume_Autor,
autori.ID_Autor FROM carte_autor RIGHT JOIN autori ON carte_autor.ID_Autor=autori.ID_Autor
GROUP BY Nume_Autor HAVING Nume_Autor LIKE ? LIMIT \$e_p OFFSET \$offset;
([Autori_Result.php](#))

12) SELECT COUNT(ID_Autor) FROM autori WHERE ID_Autor=?; ([Autori_Sterge.php](#))

13) SELECT ID_Editura FROM edituri WHERE Cod_Editura=?; ([Edituri_Add.php](#))

- 14) SELECT Nume, Profit_An, edituri.ID_Editura, COUNT(Cod_Carte) AS NrTitluri, Cod_Editura
FROM edituri LEFT JOIN editura_carte ON edituri.ID_Editura=editura_carte.ID_Editura
WHERE edituri.ID_Editura=? GROUP BY edituri.ID_Editura; ([Edituri_Detalii.php](#))
- 15) SELECT Tara, Oras, Strada, Nr, Tel_1, Tel_2, E_mail FROM contacte WHERE ID_Editura=?;([Edituri_Detalii.php](#))
- 16) SELECT ID_Editura FROM edituri WHERE Cod_Editura=? AND Cod_Editura!=?; ([Edituri_Edit.php](#))
- 17) SELECT Nume, Profit_An, Cod_Editura FROM edituri
WHERE ((isset(\$nr)) ? "ID_Editura=?" : "Cod_Editura=?"); ([Edituri_Edit.php](#))
- 18) SELECT Nume, Profit_An, edituri.ID_Editura, COUNT(Cod_Carte) AS NrTitluri
FROM edituri LEFT JOIN editura_carte ON edituri.ID_Editura=editura_carte.ID_Editura \$query_0
GROUP BY edituri.ID_Editura \$sort LIMIT \$e_p OFFSET \$offset; ([Edituri_Result.php](#))
- \$query_0, o clauză WHERE construită dinamic,
 - \$sort, o clauză ORDER BY construită dinamic,
- 19) SELECT COUNT(ID_Editura) AS Nr_R FROM edituri \$query_0; ([Edituri_Result.php](#))
- \$query_0, același ca la 18),
- 20) SELECT COUNT(ID_Editura) FROM edituri WHERE ID_Editura=?; ([Edituri_Sterge.php](#))
- 21) SELECT ID_Utilizator, Prenume, Nume, Drept, E_mail FROM utilizatori WHERE ID_Utilizator=?;
([Utilizatori_Edit.php](#))

22) SELECT COUNT(ID_Utilizator) AS Nr_R **\$query_0**; ([Utilizatori_Result.php](#))

- **\$query_0** = „FROM utilizatori WHERE” ... ,

23) SELECT CONCAT(Prenume, ' ', Nume) AS Nume_User, E_mail, Drept, ID_Utilizator
\$query_0 \$sort LIMIT **\$e_p** OFFSET **\$offset**; ([Utilizatori_Result.php](#))

- **\$query_0**, același ca la 22),
- **\$sort**, clauză ORDER BY construită dinamic,

24) SELECT COUNT(ID_Utilizator) FROM utilizatori WHERE ID_Utilizator=?; ([Utilizatori_Sterge.php](#))

SELECT-uri complexe:

1) SELECT SUM(Nr_Rez) AS Nr_R FROM (
SELECT COUNT(autori.ID_Autor) AS Nr_Rez, CONCAT(Prenume, ' ', Nume) AS Nume_Autor
FROM carte_autor RIGHT JOIN autori ON carte_autor.ID_Autor=autori.ID_Autor
GROUP BY Nume_Autor HAVING Nume_Autor LIKE ?) AS autori_total; ([Autori_Result.php](#))

2) SELECT SUM(Nr_Rez) AS Nr_R
FROM (SELECT COUNT(DISTINCT Cod_Carte) AS Nr_Rez **\$query_0**) AS stoc; ([Stoc_Result.php](#))

- **\$query_0**, este cel de la 10) (de la SELECT-uri simple),

3) SELECT Tara, Oras, Strada, Nr, Tel_1, Tel_2, E_mail FROM contacte
WHERE ID_Editura=" . ((isset(**\$nr**)) ? "?" : "(SELECT ID_Editura FROM edituri WHERE Cod_Editura=?));
([Edituri_Edit.php](#))

- 4) SELECT carte_autor.ID_Autor FROM carte_autor WHERE ID_Carte=(SELECT ID_Carte FROM editura_carte WHERE Cod_Carte=?); ([Stoc_Edit.php](#))
- 5) SELECT carte_categorie.ID_Categorie FROM carte_categorie WHERE ID_Carte=(SELECT ID_Carte FROM editura_carte WHERE Cod_Carte=?); ([Stoc_Edit.php](#))