

BAZE DE DATE - 1

Autor

id_autor	nume_autor
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

Carti

idcarte	autor	titlu	pret	cantitatea	id_autor
2-2222-222-10	Petru	GFGF Laborator Mysql-Php vbvbnb	950.00	23	1
2-2222-222-13	Vieru	MAMA	2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	4
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3	6
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3	2

Concepte de bază din teoria relațională a bazelor de date

- Înainte de a începe să realizăm o aplicație cu baze de date relaționale trebuie să cunoaștem ***principalele concepte din teoria relațională.***
- Acestea sunt absolut necesare, atât pentru analiza și proiectarea bazei de date relaționale, cât și pentru elaborarea programelor de aplicație într-un SGBD relațional ales.

Concepte de bază din teoria relațională a bazelor de date

Prezentăm în continuare aceste concepte:

1. DOMENIUL reprezintă un ansamblu de valori, caracterizat printr-un nume.

- Domeniul se poate defini explicit, prin enumerarea tuturor valorilor care aparțin acestuia, sau implicit, prin precizarea proprietăților pe care le au valorile domeniului respectiv.

Concepte de bază din teoria relațională a bazelor de date

2. RELAȚIA (tabela) reprezintă un subansamblu al produsului cartezian dintre mai multe domenii, caracterizat printr-un nume.

3. ATRIBUTUL este coloana unei relații caracterizate printr-un nume.

- Fiecare atribut își ia valorile dintr-un domeniu.
- ***Mai multe attribute pot lua valori din același domeniu.***

Concepte de bază din teoria relațională a bazelor de date

- **4. TUPLUL** este o linie dintr-o relație și nu are nume.
 - Valorile dintr-un tuplu aparțin produsului cartezian dintre domeniile relației.

Concepte de bază din teoria relațională a bazelor de date

5. SCHEMA RELAȚIEI este un ansamblu format din numele relației (**R**) urmat între paranteze rotunde de lista atributelor (**A_i**), pentru fiecare atribut precizându-se domeniul asociat (**D_i**):

$$R (A_1: D_1, A_2: D_2, \dots, A_n: D_n),$$

unde $n \geq m$ sunt numere întregi.

Concepte de bază din teoria relațională a bazelor de date

6. CHEIA este un atribut sau un ansamblu de attribute cu ajutorul căruia se poate identifica un tuplu dîintr-o relație.

Concepte de bază din teoria relațională a bazelor de date

- Numim **cheie primară** (*primary key PK*) a unei relații, un atribut (sau un grup de attribute) care identifică fără ambiguitate fiecare linie a relației.
- Numim **cheie straină** (*foreign key FK*) a unei relații un grup de attribute care pune în legătură linii din două tabele (relații).

Concepte de bază din teoria relațională a bazelor de date

Pentru exemplificare considerăm următoarea **baza de date FACULTATE** care conține **cinci tabele**:

1. Tabela **GRUPE** cu cheile: profil și **codgr**=codul grupei
2. Tabela **STUDENTI** cu cheile: codgr și **codst**=codul studentului
3. Tabela **DISCIPLINE** cu cheile **codob**=codul disciplinei și **denumire**
4. Tabela **PROFESORI** cu cheile **codprof**=codul profesorului, **numele** și **specializarea**
5. Tabela **INCADRARE** care are chei de la celelalte tabele: **codgr**, **codob** și **codprof**.

Concepte de bază din teoria relațională a bazelor de date

Cheile primare (identificatorii unici) sunt:

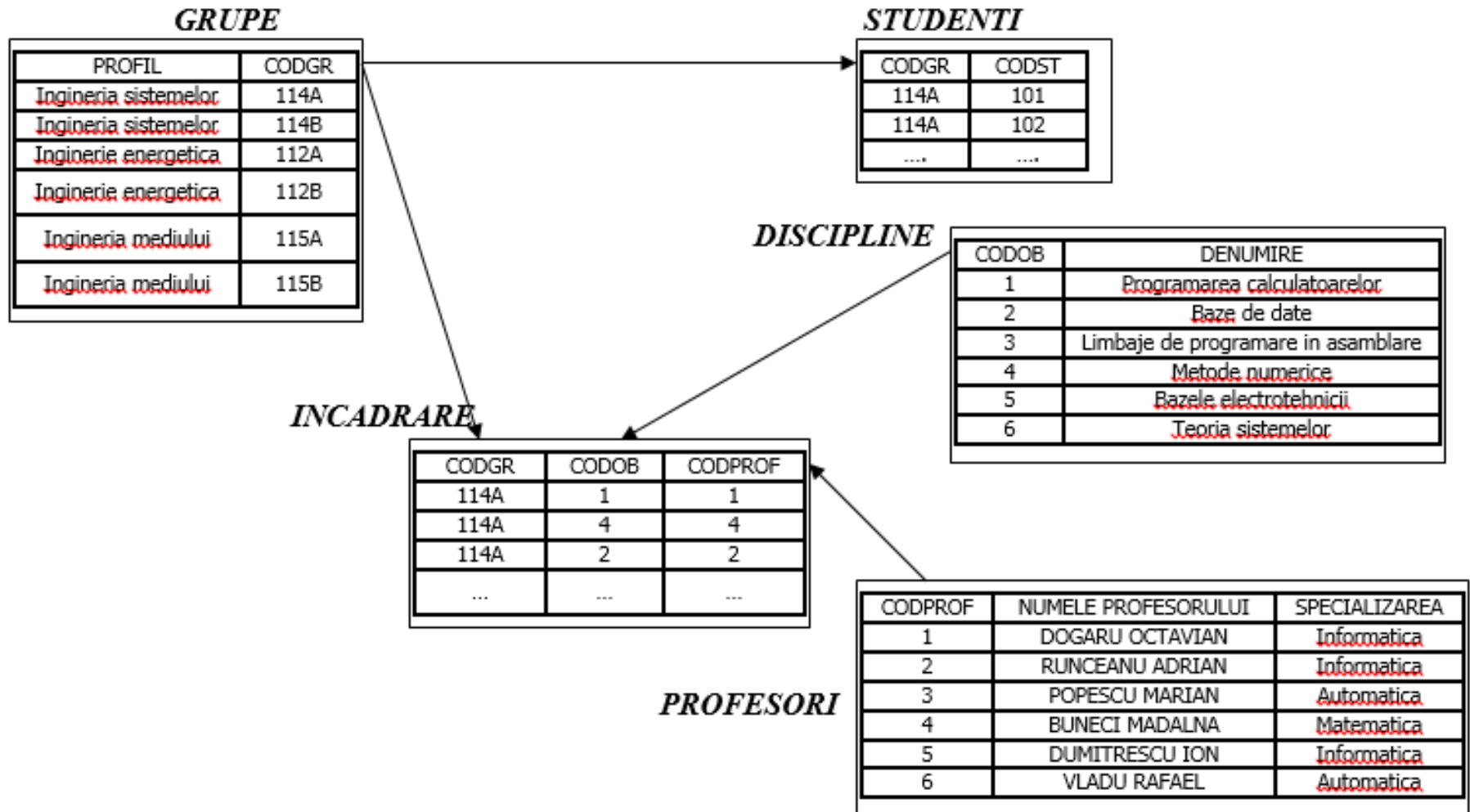
- în tabela GRUPE – **codgr**
- în tabela STUDENTI – **codst**
- în tabela DISCIPLINE – **codob**
- în tabela PROFESORI – **codprof**
- în tabela INCADRARE – **atributul compus** – **codgr+codob+codprof**

Concepte de bază din teoria relațională a bazelor de date

Cheile străine sunt:

- atributul **GRUPE.codgr** pentru tabela STUDENTI (refera tabela GRUPE)
- atributul **INCADRARE.codob** pentru tabela INCADRARE (refera tabela DISCIPLINE)
- atributul **INCADRARE.codgr** pentru tabela INCADRARE (refera tabela GRUPE)
- atributul **INCADRARE.codprof** pentru tabela INCADRARE (refera tabela PROFESORI).

Concepte de bază din teoria relațională a bazelor de date



Programarea bazelor de date

Limbajul SQL

Limbaajul SQL

Cereri SELECT pe o singură tabelă

1. **SELECT. Sintaxa. Efect. Rezultat**
2. **Lista SELECT**
3. **Clauza WHERE**

Limbajul SQL

Limbajul **SQL (Structured Query Language)** se bazează pe studiile lui E.F. Codd, prima implementare a acestui limbaj datând din anul 1970.

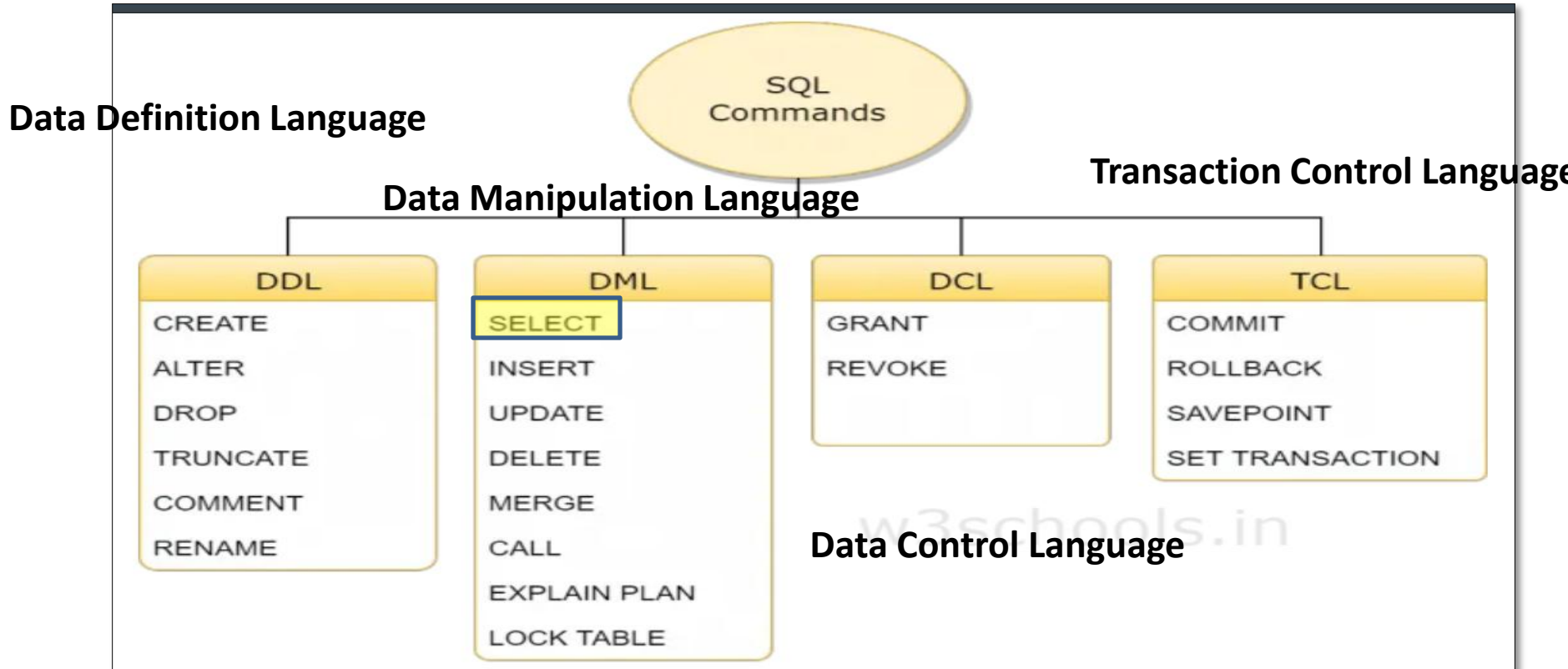
Este un limbaj complet standardizat și se poate utiliza pentru a accesa baze de date prezent în SGBD:

1. **Oracle**
2. **SQL Server**
3. **DB2**
4. **MySQL (open-source)**
5. **MsSQL**

Limbajul SQL

SQL utilizează o sintaxă foarte simplă și ușor de folosit.

Comenzile **SQL** sunt grupate în cinci categorii, astfel:





Limbajul SQL



1. **Limbajul de interogare** permite regăsirea liniilor memorate în tabelele bazelor de date.

Comanda utilizată este **SELECT**.

SELECT [**DISTINCT** | **ALL**] { * | [fieldExpression [**AS** newName]}]

FROM tableName [alias]

[**WHERE** condition]

[**GROUP BY** fieldName(s)]

[**HAVING** condition]

ORDER BY fieldName(s)



Limbajul SQL

2. Limbajul de manipulare a datelor (LMD) permite modificarea conținutului tabelelor.

Comenzile utilizate sunt:

- 1. INSERT** – pentru adăugarea de noi linii într-o tabelă
- 2. UPDATE** – pentru modificarea valorilor memorate într-o tabelă
- 3. DELETE** – pentru ștergerea liniilor dintr-o tabelă



Limbajul SQL

3. Limbajul de definire a datelor (LDD) permite definirea structurii tabelor ce compun bazele de date.

Comenzile utilizate sunt:

- 1. CREATE** – pentru crearea structurii unei baze de date sau a unei tabele
- 2. ALTER** – pentru modificarea structurii unei baze de date sau a unei tabele
- 3. DROP** – pentru ștergerea structurii bazei de date
- 4. RENAME** – schimbarea numelui unei tabele
- 5. TRUNCATE** – ștergerea conținutului unei tabele



Limbajul SQL

4. Limbajul de control al tranzacțiilor (LCT)

Comenzile utilizate sunt:

1. **COMMIT** – pentru ca modificările efectuate asupra bazei de date să devină permanente
2. **ROLLBACK** – permite renunțarea la ultimele modificări asupra bazei de date
3. **SAVEPOINT** – pentru definirea unui “punct de salvare” la care se poate reveni, renunțând la modificările făcute după acest punct asupra bazei de date



Limbajul SQL

5. Limbajul de control al datelor (LCD) permite definirea și modificarea drepturilor asupra bazelor de date.

Comenzile utilizate sunt:

- 1. GRANT** – pentru acordarea unor drepturi altor utilizatori asupra bazei de date
- 2. REVOKE** – pentru anularea unor anumite drepturi ale utilizatorilor

Limbajul SQL

Principalele elemente care compun o comanda SQL:

a) Nume

Toate obiectele dintr-o bază de date: tabele, coloane, vizualizări, indecși, etc, au un *nume*.

Numele poate fi orice șir de *maximum 30* de litere, cifre și caractere speciale (“_”, “#”, “\$”), *primul caracter fiind obligatoriu o literă*.

Limbajul SQL

b) Cuvinte rezervate

La fel ca în orice limbaj, și în **SQL** există o listă de cuvinte rezervate. Aceste cuvinte nu pot fi utilizate în alt scop decât cel definit inițial.

```
ACCESS      DISTINCT*  IS*
ADD         DROP
ALL*        LEVEL*
ALTER       ELSE*      LIKE*
AND*        EXCLUSIVE  LOCK
ANY*        EXISTS*   LONG
AS*
ASC*        FILE       MAXEXTENTS
AUDIT       FLOAT*     MINUS
BETWEEN*   FROM*     MODIFY*
BY*        NOAUDIT
```

c) Constante

O constantă sau literal este o valoare fixă care nu poate fi modificată.

Limbajul SQL

Există:

1. **Constante numerice**, de exemplu 4, 12.34, .9, etc.
 - Se observă că dacă un număr real are partea întreagă egală cu zero, atunci ea nu mai trebuie precizată.
2. **Constante alfanumerice (sau șir de caractere)**
 - Constantele șir de caractere sunt scrise între apostrofuri și sunt case-sensitive.
 - Exemple: 'abc', 'baza de date'.

Limbajul SQL

d) Variabile

Variabilele sunt date care pot avea în timp valori diferite.

O variabilă are întotdeauna un nume pentru a putea fi referită.

SQL acceptă două tipuri de variabile:

1. variabilele asociate numelor coloanelor asociate
2. variabile sistem

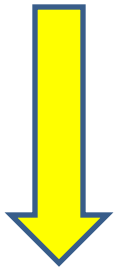
Limbajul SQL

e) Expresii

O expresie este formată din variabile, constante, operatori și funcții.

Operatorii care pot fi utilizați sunt împărțiți astfel:

1. Operatori aritmetici
2. Operatori alfanumerici
3. Operatori de comparatie
4. Operatori logici



Limbajul SQL

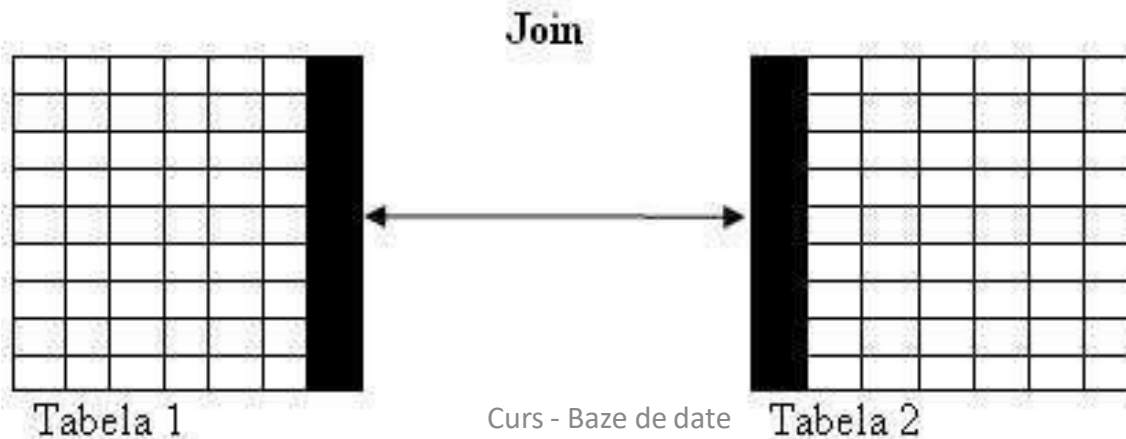
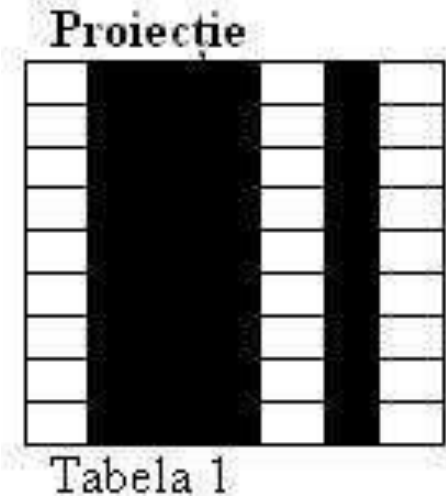
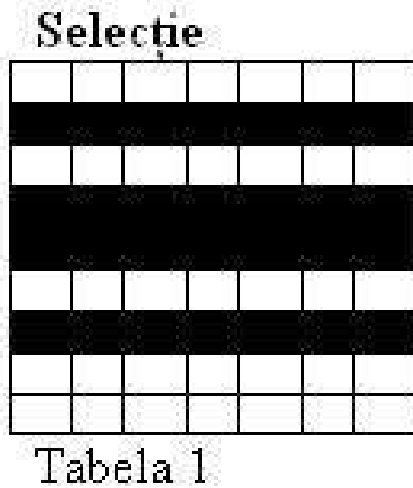
Cu ajutorul comenzii **SELECT** se pot realiza următoarele tipuri de operații:

- 1. Selecția** – constă în **filtrarea liniilor** care vor fi afișate
- 2. Proiecția** – constă în alegerea doar a **anumitor coloane** pentru a fi afișate
- 3. Join** – constă în preluarea datelor din două sau mai multe tabele “legate” conform unor reguli precizate



Limbajul SQL

Operațiile realizate cu ajutorul comenzii **SELECT**



Exemple

Considerăm următoarele tabele:

1. **STUD** având structura **MATR**, NUME, AN, GRUPA, DATAN, LOC, INDRUMATOR, PUNCTAJ, CODS
2. **SPECIALIZARE** având structura **CODS**, NUME, DOMENIU
3. **BURSA** având structura **TIP**, PMIN, PMAX, SUMA

Tabela STUD

MATR	NUME	AN	GRUPA	DATAN	LOC	INDRUMATOR	PUNCTAJ	CODS
1234	POPA MARCEL	1	114A	12-03-87	CHISINAU	1001	2345	1
1235	POPESCU ION	2	121B	02-04-89	ANENI	1001	1300	1
1236	AVRAM NICOLAE	1	115A	21-03-68	ANENI	1002	3000	2
1237	IONESCU MARIANA	2	116C	05-05-89	CHISINAU	1003	1234	3
1256	POPESCU GINA	3	114A	06-09-90	CAHUL	1002	3456	2

Tabelele SPECIALIZARE si BURSA

CODS	NUME	DOMENIU
1	AUTOMATICA	CALCULATOARE
2	ENERGETICA	INGINERIE ELECTRICA
3	MECANICA	INGINERIE MECANICA

Tip	Pmin	Pmax	Suma
Fara bursa	0	399	
Bursa sociala	400	899	100
Bursa de studiu	900	1799	150
Bursa de merit	1800	2499	200
Bursa de exceptie	2500	999	300

Tabelele SPECIALIZARE si BURSA

Autor

id_autor	nume_autor
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

Carti

idcarte	autor	titlu	pret	cantitatea	id_autor
2-2222-222-10	Petru	GFGF Laborator Mysql-Php vbvbvb	950.00	23	1
2-2222-222-13	Vieru	MAMA	2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	4
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3	6
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3	2

