

# Algebra Relatională și Limbajul MySQL

## 1 Introducere

Algebra relațională este fundamentul teoretic al bazelor de date relaționale și definește operații formale pentru manipularea tabelor (relațiilor). Limbajul MySQL este o implementare practică a acestor operații, permițând manipularea și interogarea bazelor de date relaționale.

## 2 Elementele Algebrei Relaționale și Echivalentul MySQL

### 2.1 Selecție ( $\sigma$ )

Operația de selecție filtrează rândurile unei tabele pe baza unei condiții.

**Algebra relațională:**

$$\sigma_{condi\breve{t}ie}(Tabel)$$

**MySQL:**

```
SELECT *  
FROM Tabel  
WHERE condi ie ;
```

**Exemplu:** Obțineți toți studenții cu vârsta peste 21 de ani.

$$\sigma_{Varsta>21}(Student)$$

```
SELECT *  
FROM Student  
WHERE Varsta > 21;
```

### 2.2 Proiecție ( $\pi$ )

Operația de proiecție selectează coloane specifice dintr-o tabelă.

**Algebra relațională:**

$$\pi_{coloane}(Tabel)$$

**MySQL:**

```
SELECT coloane  
FROM Tabel;
```

**Exemplu:** Listați numele și vârsta tuturor studenților.

$$\pi_{Nume,Varsta}(Student)$$

```
SELECT Nume, Varsta  
FROM Student;
```

## 2.3 Reuniune ( $\cup$ )

Operația de reuniune combină rândurile din două tabele, eliminând duplicatele.

**Algebra relațională:**

$$Tabel1 \cup Tabel2$$

**MySQL:**

```
SELECT *  
FROM Tabel1  
UNION  
SELECT *  
FROM Tabel2;
```

## 2.4 Diferență ( $-$ )

Operația de diferență returnează rândurile din primul tabel care nu sunt în cel de-al doilea.

**Algebra relațională:**

$$Tabel1 - Tabel2$$

**MySQL:**

```
SELECT *  
FROM Tabel1  
WHERE NOT EXISTS (  
    SELECT *  
    FROM Tabel2  
    WHERE Tabel1.CheiePrimara = Tabel2.CheiePrimara  
);
```

**Exemplu:** Găsiți studenții care nu sunt înscriși la niciun curs.

$$Student - \pi_{StudentID}(Inscriere)$$

```
SELECT *  
FROM Student  
WHERE StudentID NOT IN (SELECT StudentID FROM Insciriere);
```

## 2.5 Produs Cartezien ( $\times$ )

Produsul cartezian combină fiecare rând din primul tabel cu fiecare rând din cel de-al doilea.

**Algebra relațională:**

$$Tabel1 \times Tabel2$$

**MySQL:**

```
SELECT *  
FROM Tabel1 , Tabel2;
```

## 2.6 Îmbinare ( $\bowtie$ )

Îmbinarea combină două tabele pe baza unei condiții.

**Algebra relațională:**

$$Tabel1 \bowtie_{condiție} Tabel2$$

**MySQL:**

```
SELECT *
FROM Tabel1
JOIN Tabel2
ON condi ie;
```

**Exemplu:** Listați studenții împreună cu cursurile la care sunt înscriși.

$$Student \bowtie_{Student.StudentID=Inscriere.StudentID} Inscriere$$

```
SELECT Student.Nume, Inscriere.CursID
FROM Student
JOIN Inscriere
ON Student.StudentID = Inscriere.StudentID;
```

## 2.7 Împărțire (/)

Împărțirea returnează toate rândurile din primul tabel care sunt asociate cu toate rândurile din cel de-al doilea.

**Algebra relațională:**

$$Tabel1/Tabel2$$

**MySQL:**

```
SELECT DISTINCT A.Coloana1
FROM Tabel1 A
WHERE NOT EXISTS (
    SELECT *
    FROM Tabel2 B
    WHERE NOT EXISTS (
        SELECT *
        FROM Tabel1 C
        WHERE C.Coloana1 = A.Coloana1
        AND C.Coloana2 = B.Coloana2
    )
);
```

**Exemplu:** Găsiți studenții care urmează toate cursurile oferite de universitate.

## 3 Probleme Propuse

1. Traduceți în MySQL următoarea expresie:

$$\sigma_{Nota > 7}(Inscriere)$$

2. Traduceți în algebra relațională următoarea interogare MySQL:

```
SELECT DISTINCT StudentID
FROM Inscriere
WHERE Nota >= 5;
```

3. Traduceți în MySQL următoarea expresie:

$$\pi_{Nume}(Student \bowtie_{Student.StudentID=Inscriere.StudentID} Inscriere)$$

4. Traduceți în algebra relațională următoarea interogare MySQL:

```

SELECT Nume
FROM Student
WHERE Varsta > 20;

```

5. Traduceți în MySQL următoarea expresie:

$$\pi_{Nume}(Student) - \pi_{Nume}(Student \bowtie_{Student.StudentID=Inscriere.StudentID} Inscriere)$$

6. Traduceți în algebra relațională următoarea interogare MySQL:

```

SELECT CursID , AVG(Nota)
FROM Insciere
GROUP BY CursID;

```

7. Traduceți în MySQL următoarea expresie:

$$\pi_{StudentID}(\sigma_{Nota=10}(Inscriere))$$

8. Traduceți în algebra relațională:

```

SELECT StudentID
FROM Insciere
WHERE Nota IS NULL;

```

9. Traduceți în MySQL următoarea expresie:

$$\sigma_{Nume='Matematica'}(Curs)$$

10. Traduceți în algebra relațională:

```

SELECT DISTINCT StudentID
FROM Insciere
WHERE Nota >= 5;

```