

## Tema: Interogări din mai multe tabele. Interogări cu grupări. Clauzele GROUP BY și HAVING

- se referă la tehnici fundamentale în manipularea datelor din baze de date relaționale folosind limbajul SQL.

### Interogări din mai multe tabele

În baze de date relaționale, datele sunt distribuite în mai multe tabele legate între ele prin chei primare și chei externe. Pentru a obține informații relevante din astfel de structuri, trebuie să utilizăm **JOIN-uri**.

#### Tipuri de JOIN-uri

- **INNER JOIN** – returnează doar înregistrările care au corespondență în ambele tabele.
- **LEFT JOIN** – returnează toate înregistrările din tabelul din stânga și doar cele potrivite din tabelul din dreapta.
- **RIGHT JOIN** – returnează toate înregistrările din tabelul din dreapta și doar cele potrivite din tabelul din stânga.
- **FULL JOIN** – returnează toate înregistrările din ambele tabele.

#### Exemplu de interogare cu JOIN

Să presupunem că avem două tabele:

- Elevi (id\_elev, nume, clasa)
- Note (id\_nota, id\_elev, materie, nota)

*Dacă dorim să aflăm numele elevilor și notele lor la diferite materii, putem folosi:*

**SELECT** Elevi.nume, Note.materie, Note.nota

**FROM** Elevi

**INNER JOIN** Note **ON** Elevi.id\_elev = Note.id\_elev;

## 2. Clauza GROUP BY

Aceasta este utilizată pentru a grupa rezultatele în funcție de o coloană și este folosită frecvent împreună cu funcții de agregare (COUNT, SUM, AVG, MAX, MIN).

#### Exemplu de utilizare

*Dacă dorim să aflăm media notelor la fiecare materie, putem folosi:*

**SELECT** materie, **AVG**(nota) **AS** medie\_nota

**FROM** Note

**GROUP BY** materie;

Această interogare grupează notele pe materii și calculează media fiecărei grupe.

### 3. Clauza HAVING

Aceasta este utilizată pentru a filtra rezultatele după ce au fost aplicate funcții de agregare (spre deosebire de `WHERE`, care filtrează înainte de grupare).

#### Exemplu de utilizare

*Dacă dorim să selectăm doar materiile unde media notelor este mai mare de 7:*

```
SELECT materie, AVG(nota) AS medie_nota
```

```
FROM Note
```

```
GROUP BY materie
```

```
HAVING AVG(nota) > 7;
```

### 4. Exemplu complex: JOIN + GROUP BY + HAVING

*Dacă dorim să aflăm media notelor pentru fiecare elev și să selectăm doar elevii cu media peste 8, putem scrie:*

```
SELECT Elevi.nume, AVG(Note.nota) AS medie_nota
```

```
FROM Elevi
```

```
INNER JOIN Note ON Elevi.id_elev = Note.id_elev
```

```
GROUP BY Elevi.nume
```

```
HAVING AVG(Note.nota) > 8;
```

#### Alte exemple:

*Scrie o interogare care să returneze numărul de elevi din fiecare clasă.*

```
SELECT clasa, COUNT(id_elev) AS numar_elevi
```

```
FROM Elevi
```

```
GROUP BY clasa;
```

*Afișează numele elevilor care au media notelor mai mare de 8.*

```
SELECT Elevi.numa, AVG(Note.nota) AS medie_nota  
FROM Elevi  
INNER JOIN Note ON Elevi.id_elev = Note.id_elev  
GROUP BY Elevi.numa  
HAVING AVG(Note.nota) > 8;
```

*Listează materiile la care au fost înregistrate cel puțin 5 note.*

```
SELECT materie, COUNT(nota) AS numar_note  
FROM Note  
GROUP BY materie  
HAVING COUNT(nota) >= 5;
```

*Afișează numele elevului care a primit cele mai multe note.*

```
SELECT Elevi.numa, COUNT(Note.nota) AS numar_note  
FROM Elevi  
INNER JOIN Note ON Elevi.id_elev = Note.id_elev  
GROUP BY Elevi.numa  
ORDER BY numar_note DESC  
LIMIT 1;
```

*Identifică clasele unde media notelor este mai mică de 7.*

```
SELECT Elevi.clasa, AVG(Note.nota) AS medie_clasa  
FROM Elevi  
INNER JOIN Note ON Elevi.id_elev = Note.id_elev  
GROUP BY Elevi.clasa  
HAVING AVG(Note.nota) < 7;
```