INFORMATICĂ · CLASA A XII-A · BAZE DE DATE — SQL · LECȚIA 6

SQL (3): funcții de agregare, GROUP BY, HAVING, JOIN și subinterogări

Durată: 60 min Nivel: mate-info (clasa a XII-a) Înainte: lecțiile XII.5

CE VEI ȘTI DUPĂ LECȚIE

- Folosești COUNT, SUM, AVG, MIN, MAX.
- Grupezi rezultate cu GROUP BY și filtrezi grupurile cu HAVING.
- Combini date din mai multe tabele cu INNER JOIN și LEFT JOIN.
- Scrii subinterogări în WHERE.

1. Funcții de agregare

```sql
SELECT COUNT(*) FROM elevi;                                  -- câți elevi
SELECT AVG(nota) FROM note WHERE id_disc = 1;                -- media la disciplina 1
SELECT MIN(nota), MAX(nota) FROM note;
SELECT COUNT(DISTINCT id_clasa) FROM elevi;                  -- în câte clase sunt elevi
```

2. GROUP BY și HAVING

„Media fiecărui elev” — calculăm o valoare **pe grup**:

```sql
SELECT id_elev, AVG(nota) AS medie
FROM note
GROUP BY id_elev
HAVING AVG(nota) >= 9            -- filtru pe GRUPURI (după agregare)
ORDER BY medie DESC;
```

LA BAC

- `WHERE` filtrează **rândurile** înainte de grupare; `HAVING` filtrează **grupurile** după agregare. - În `SELECT`, alături de funcțiile de agregare, pot apărea doar coloanele din `GROUP BY`. - Ordinea clauzelor: `SELECT … FROM … WHERE … GROUP BY … HAVING … ORDER BY …`.

3. JOIN — legarea tabelelor

Numele elevilor împreună cu numele clasei:

```sql
SELECT e.nume, e.prenume, c.nume_clasa
FROM elevi e
INNER JOIN clase c ON e.id_clasa = c.id_clasa;
```

- `e` și `c` sunt **aliasuri** pentru tabele (scriem mai puțin);
- `INNER JOIN` păstrează doar rândurile care au pereche în ambele tabele;

- `LEFT JOIN` păstrează **toate** rândurile din tabelul din stânga (cele fără pereche primesc NULL).

```sql
-- toate clasele și numărul de elevi din fiecare (inclusiv clasele fără elevi)
SELECT c.nume_clasa, COUNT(e.id_elev) AS nr_elevi
FROM clase c
LEFT JOIN elevi e ON e.id_clasa = c.id_clasa
GROUP BY c.id_clasa, c.nume_clasa;
```

Trei tabele

```sql
-- media fiecărui elev la fiecare disciplină
SELECT e.nume, d.denumire, AVG(n.nota) AS medie
FROM note n
JOIN elevi e ON n.id_elev = e.id_elev
JOIN discipline d ON n.id_disc = d.id_disc
GROUP BY e.id_elev, e.nume, d.id_disc, d.denumire;
```

4. Subinterogări

O interogare în interiorul alteia:

```sql
-- elevii cu cel puțin o notă sub 5
SELECT nume, prenume FROM elevi
WHERE id_elev IN (SELECT id_elev FROM note WHERE nota < 5);

-- notele peste media generală
SELECT * FROM note
WHERE nota > (SELECT AVG(nota) FROM note);
```

5. Exerciții pe baza de date ȘCOALA

Încarcă întâi scriptul `scoala.sql` (din același folder) în DB Browser for SQLite.

1. Câte note de 10 sunt în total? Dar la fiecare disciplină?
2. Afișează clasele care au cel puțin 3 elevi.
3. Afișează elevii (nume, clasă) care nu au nicio notă (indiciu: LEFT JOIN + `IS NULL` sau `NOT IN`).
4. Afișează disciplina cu cea mai mare medie.
5. Afișează primii 3 elevi după media generală.

6. Rezumat

- `COUNT`, `SUM`, `AVG`, `MIN`, `MAX`.
- `GROUP BY` = câte un rezultat pe grup; `HAVING` = filtru pe grupuri.
- `JOIN … ON` leagă tabelele prin cheile străine; `LEFT JOIN` păstrează și rândurile fără pereche.
- Subinterogări în `WHERE … IN (SELECT …)` sau comparații cu un singur rezultat.