INFORMATICĂ · CLASA A XII-A · BAZE DE DATE — SQL · LECȚIA 4

SQL (1): crearea și modificarea tabelelor (CREATE, ALTER, DROP)

Durată: 60 min Nivel: mate-info (clasa a XII-a) Înainte: lecțiile XII.3

CE VEI ȘTI DUPĂ LECȚIE

- Creezi tabele cu tipuri de date potrivite și constrângeri (PRIMARY KEY, NOT NULL, UNIQUE, CHECK, FOREIGN KEY).
- Modifici structura unui tabel cu ALTER TABLE.
- Ștergi tabele cu DROP TABLE și înțelegi ordinea impusă de cheile străine.

1. Limbajul SQL

SQL (Structured Query Language) este limbajul standard pentru baze de date relaționale. Comenzile se împart în:

- **DDL** (definirea datelor): `CREATE`, `ALTER`, `DROP` — structura;
- **DML** (manipularea datelor): `INSERT`, `UPDATE`, `DELETE`, `SELECT` — conținutul.

Cuvintele cheie nu țin cont de litere mari / mici; prin convenție le scriem cu majuscule. Fiecare comandă se termină cu `;`.

2. Tipuri de date uzuale

Tip	Ce memorează	Exemplu
`INTEGER` / `INT`	întregi	42
`DECIMAL(p, s)` / `NUMERIC`	numere cu zecimale exacte	`DECIMAL(4,2)` → 9.75
`VARCHAR(n)`	text de cel mult n caractere	`'Popescu'`
`CHAR(n)`	text de exact n caractere	CNP: `CHAR(13)`
`DATE`	dată calendaristică	`'2026-10-17'`

(În Oracle se folosesc des `NUMBER` și `VARCHAR2`; ideea e aceeași.)

3. CREATE TABLE

```sql
CREATE TABLE clase (
    id_clasa    INTEGER PRIMARY KEY,
    nume_clasa  VARCHAR(10) NOT NULL UNIQUE,
    profil      VARCHAR(30),
    diriginte   VARCHAR(50)
);

CREATE TABLE elevi (
    id_elev       INTEGER PRIMARY KEY,
    nume          VARCHAR(40) NOT NULL,
    prenume       VARCHAR(40) NOT NULL,
    data_nasterii DATE,
    id_clasa      INTEGER,
    FOREIGN KEY (id_clasa) REFERENCES clase(id_clasa)
);

CREATE TABLE discipline (
    id_disc   INTEGER PRIMARY KEY,
    denumire  VARCHAR(40) NOT NULL UNIQUE
);

CREATE TABLE note (
    id_nota   INTEGER PRIMARY KEY,
    id_elev   INTEGER NOT NULL REFERENCES elevi(id_elev),
    id_disc   INTEGER NOT NULL REFERENCES discipline(id_disc),
    nota      INTEGER CHECK (nota BETWEEN 1 AND 10),
    data_nota DATE
);
```

ATENȚIE — GREȘELI FRECVENTE

Tabelele se creează **în ordinea** dependențelor: întâi CLASE, apoi ELEVI (care face referire la CLASE), apoi NOTE. La ștergere, ordinea e inversă.

4. ALTER TABLE

```sql
ALTER TABLE elevi ADD email VARCHAR(80);          -- adăugare coloană
ALTER TABLE elevi DROP COLUMN email;              -- ștergere coloană (nu în toate SGBD-urile vechi)
ALTER TABLE clase RENAME TO clase_liceu;          -- redenumire tabel
```

5. DROP TABLE

```sql
DROP TABLE note;
DROP TABLE elevi;
DROP TABLE discipline;
DROP TABLE clase;
```

CAPCANĂ

`DROP TABLE` șterge definitiv tabelul **cu tot conținutul**. Nu există „Undo”.

6. Exerciții

1. Scrie comenzile CREATE pentru baza de date a unei biblioteci (CARTI, CITITORI, IMPRUMUTURI) cu toate cheile și constrângerile.
2. Adaugă în tabelul ELEVI o coloană `telefon` de tip text cu 10 caractere.
3. Ce se întâmplă dacă încerci să ștergi tabelul CLASE înaintea tabelului ELEVI? De ce?

7. Rezumat

- `CREATE TABLE nume ( câmp TIP constrângeri, … );`
- Constrângeri: `PRIMARY KEY`, `NOT NULL`, `UNIQUE`, `CHECK`, `FOREIGN KEY … REFERENCES`.
- `ALTER TABLE` modifică structura; `DROP TABLE` șterge definitiv.
