1. TEORIE

Limbajul de manipulare a datelor (LMD)
Limbajul de control al datelor (LCD)

➔ Comenzile SQL care alcătuiesc **LMD** sunt:
- ❖ ***SELECT*** - regăsirea datelor
- ❖ ***INSERT*** - adăugarea de noi înregistrări
- ❖ ***UPDATE*** - modificarea valorilor coloanelor din înregistrările existente
- ❖ ***DELETE*** - suprimarea de înregistrări
- ❖ ***MERGE*** - adăugarea sau modificarea condiționată de înregistrări

tranzacţie = operaţie asupra unei baze de date care implică una sau mai multe modificări în unul sau mai multe tabele

➔ Comenzile SQL care alcătuiesc **LCD** sunt:
- ❖ ***ROLLBACK*** - pentru a renunța la modificările aflate în aşteptare
  - în urma execuției acesteia:
    - ➢ se încheie tranzacția
    - ➢ se anulează modificările asupra datelor
    - ➢ se restaurează starea lor precedentă
    - ➢ se eliberează blocările asupra liniilor
- ❖ ***COMMIT*** - determină încheierea tranzacției curente şi permanentizarea modificărilor care au intervenit pe parcursul acesteia
  - instrucțiunea suprimă toate punctele intermediare definite în tranzacție şi eliberează blocările tranzacției.

  OBS: O comandă LDD (*CREATE*, *ALTER*, *DROP*) determină un COMMIT implicit.
- ❖ ***SAVEPOINT*** - marchează un punct intermediar în procesarea tranzacției
  - în acest mod este posibilă împărțirea tranzacției în subtranzacții

1. Comanda INSERT

a. Inserări mono-tabel

```
INSERT INTO nume_tabel (coloana1, coloana2, coloana3,...)
VALUES (valoare1, valoare2, valoare3,...);
```

Dacă sunt introduse valori pentru toate coloanele din tabel, instrucțiunea poate fi scrisă astfel:

```
INSERT INTO nume_tabel
VALUES (valoare1, valoare2, valoare3,...);
```

În acest caz valorile trebuie introduse în aceeași ordine în care se regăsesc coloanele în tabel.

Inserarea într-un tabel se poate face şi prin intermediul unei subcereri, astfel:

```
INSERT INTO nume_tabel (coloana1, coloana2, coloana3,...)
SELECT …

INSERT INTO nume_tabel
SELECT …
```

Dacă în tabel se introduc linii prin intermediul unei subcereri, coloanele din lista *SELECT* trebuie să corespundă, ca număr şi tip, celor precizate în clauza *INTO*. În absența unei liste de coloane în clauza *INTO*, subcererea trebuie să furnizeze valori pentru fiecare atribut al tabelului destinație, respectând ordinea acestora.

b. Inserări multi-tabel

i. Inserări necondiţionate

```
INSERT ALL
      INTO nume_tabel_1 [...]
      INTO nume_tabel_2 [...]
      INTO nume_tabel_n [...]
SELECT ... ;
```

ii. Inserări condiţionate

```
INSERT [ALL | FIRST]
      WHEN condiţie1 THEN INTO nume_tabel_1
      WHEN condiţie2 THEN INTO nume_tabel_2
      ELSE INTO nume_tabel_n
SELECT ...;
```

- *ALL* determină evaluarea tuturor condițiilor din clauzele *WHEN*. Pentru cele a căror valoare este *TRUE*, se inserează înregistrarea specificată în opțiunea *INTO* corespunzătoare.
- *FIRST* determină inserarea corespunzătoare primei clauze *WHEN* a cărei condiție este evaluată *TRUE*. Toate celelalte clauze *WHEN* sunt ignorate.

2. Comanda UPDATE

```
UPDATE nume_tabel
SET coloana1 = valoare1, coloana2 = valoare2, ...
[WHERE conditie];
```

sau

```
UPDATE nume_tabel
SET (coloana1, coloana2, ...) = (SELECT ...)
[WHERE conditie];
```

Observații:

- de obicei pentru identificarea unei linii se foloseşte o condiție ce implică cheia primară;
- dacă nu apare clauza *WHERE* atunci sunt afectate toate liniile tabelului specificat;

3. Comanda DELETE

```
DELETE FROM nume_tabel
[WHERE conditie];
```

- dacă nu se specifică nicio condiţie, vor fi şterse toate liniile din tabel

4. Comanda MERGE

➔ permite inserarea sau actualizarea condiționată a datelor dintr-un tabel al bazei de date

```
MERGE INTO nume_tabel_1
USING nume_tabel_2 / subcerere / vizualizare
ON condiţie
WHEN MATCHED THEN
      UPDATE SET
            coloana_1 = valoare_1
            coloana_n = valoare_n
WHEN NOT MATCHED THEN
      INSERT  (coloana_1, … , coloana_n)
      VALUES (valoare_1, … , valoare_n);
```

Instrucțiunea efectuează:

- *UPDATE,* dacă înregistrarea există deja în tabel
- *INSERT,* dacă înregistrarea este nouă.

---

2. EXERCIŢII

[Comanda INSERT]

1. Să se creeze tabelele *EMP_pnu*, *DEPT_pnu* (în şirul de caractere “pnu”, *p* reprezintă prima literă a prenumelui, iar *nu* reprezintă primele două litere ale numelui dumneavoastră), prin copierea structurii şi conținutului tabelelor *EMPLOYEES*, respectiv *DEPARTMENTS*. Listați structura tabelelor sursă şi a celor create anterior. Ce se observă?

```
CREATE TABLE emp_pnu AS
SELECT * FROM employees;

CREATE TABLE dept_pnu AS
SELECT * FROM departments;
```

2. Pentru introducerea constrângerilor de integritate, executați instrucțiunile LDD indicate în continuare. Prezentarea detaliată a LDD se va face în cadrul laboratorului 9.

```
ALTER TABLE emp_pnu
ADD CONSTRAINT pk_emp_pnu PRIMARY KEY(employee_id);

ALTER TABLE dept_pnu
ADD CONSTRAINT pk_dept_pnu PRIMARY KEY(department_id);

ALTER TABLE emp_pnu
ADD CONSTRAINT fk_emp_dept_pnu
   FOREIGN KEY(department_id) REFERENCES dept_pnu(department_id);
```

Obs: Ce constrângere nu am implementat?

3. Să se insereze departamentul 300, cu numele *Programare* în *DEPT_pnu*.
   Analizați cazurile, precizând care este soluția corectă şi explicând erorile celorlalte variante. Pentru a anula efectul instrucțiunii(ilor) corecte, utilizați comanda *ROLLBACK*.
   a. *INSERT INTO DEPT_pnu*
      VALUES (300, 'Programare');

   b. *INSERT INTO DEPT_pnu (department_id, department_name)*
      VALUES (300, 'Programare');

   c. *INSERT INTO DEPT_pnu (department_name, department_id)*
      VALUES (300, 'Programare');

   d. *INSERT INTO DEPT_pnu (department_id, department_name, location_id)*
      VALUES (300, 'Programare', null);

   e. *INSERT INTO DEPT_pnu (department_name, location_id)*
      VALUES ('Programare', null);

   Executați varianta care a fost corectă de două ori. Ce se obține şi de ce?

4. Să se insereze un angajat corespunzător departamentului introdus anterior în tabelul *EMP_pnu*, precizând valoarea *NULL* pentru coloanele a căror valoare nu este cunoscută la inserare (metoda implicită de inserare). Determinați ca efectele instrucțiunii să devină permanente.
5. Să se mai introducă un angajat corespunzător departamentului 300, precizând după numele tabelului lista coloanelor în care se introduc valori (metoda explicita de inserare). Salvați înregistrarea.
6. Încercați, dacă este posibilă, introducerea unui angajat, precizând pentru valoarea *employee_id* o subcerere care returnează codul maxim + 1.
7. Creați un nou tabel, numit *EMP1_PNU*, care va avea aceeaşi structură ca şi *EMPLOYEES*, dar nici o înregistrare. Copiați în tabelul *EMP1_PNU* salariații (din tabelul *EMPLOYEES*) al căror comision depăşeşte 25% din salariu.
8. Inserați o nouă înregistrare în tabelul *EMP_PNU* care să totalizeze salariile, să facă media comisioanelor, iar câmpurile de tip dată să conțină data curentă şi câmpurile de tip caracter să conțină textul 'TOTAL'. Numele şi prenumele angajatului să corespundă utilizatorului curent (*USER*). Pentru câmpul *employee_id* se va introduce valoarea 0, iar pentru *manager_id* şi *department_id* se va da valoarea null.
9. Creați 2 tabele *emp2_pnu* şi *emp3_pnu* cu aceeaşi structură ca tabelul *EMPLOYEES*, dar fără înregistrări (acceptăm omiterea constrângerilor de integritate). Prin intermediul unei singure comenzi, copiați din tabelul *EMPLOYEES*:
   - în tabelul *EMP1_PNU* salariații care au salariul mai mic decât 5000;
   - în tabelul *EMP2_PNU* salariații care au salariul cuprins între 5000 şi 10000;
   - în tabelul *EMP3_PNU* salariații care au salariul mai mare decât 10000.

   Verificați rezultatele, apoi ştergeți toate înregistrările din aceste tabele.
10. Să se creeze tabelul *EMP0_PNU* cu aceeaşi structură ca tabelul *EMPLOYEES* (fără constrângeri), dar fără nici o înregistrare. Copiați din tabelul *EMPLOYEES*:
    - în tabelul *EMP0_PNU* salariații care lucrează în departamentul 80;
    - în tabelul *EMP1_PNU* salariații care au salariul mai mic decât 5000;

- în tabelul *EMP2_PNU* salariații care au salariul cuprins între 5000 şi 10000;
- în tabelul *EMP3_PNU* salariații care au salariul mai mare decât 10000.

Dacă un salariat se încadrează în tabelul *EMP0_PNU* atunci acesta nu va mai fi inserat şi în alt tabel (tabelul corespunzător salariului său).

[Comanda UPDATE]

11. Măriți salariul tuturor angajaților din tabelul *EMP_PNU* cu 5%. Vizualizati, iar apoi anulați modificările.
12. Schimbați jobul tuturor salariaților din departamentul 80 care au comision în 'SA_REP'. Anulați modificările.
13. Să se promoveze Douglas Grant la manager în departamentul 20, având o creştere de salariu cu 1000$. Se poate realiza modificarea prin intermediul unei singure comenzi?
14. Schimbați salariul şi comisionul celui mai prost plătit salariat din firmă, astfel încât să fie egale cu salariul si comisionul şefului său.
15. Pentru fiecare departament să se mărească salariul celor care au fost angajați primii astfel încât să devină media salariilor din companie.
16. Creați un script prin intermediul caruia sa fie posibilă actualizarea în mod interactiv de înregistrări ale tabelului *dept_pnu*. Se va cere codul departamentului care urmează a fi actualizat, se va afişa linia respectivă, iar apoi se vor cere valori pentru celelalte câmpuri.

[Comanda DELETE]

17. Ştergeți angajații care nu au comision. Ce înregistrări se pot şterge? Creați un nou tabel, copie a tabelului employees, fără a pune și constrangerile. Repetați ștergerea pe acesta.
18. Suprimați departamentele care un au nici un angajat. Anulați modificările.

[LMD, LCD]

19. Creați un nou tabel *DEPT_PNU2*, fără a copia și constrangerile.
20. Să se mai introducă o linie în tabelul *DEPT_PNU2*.
21. Să se marcheze un punct intermediar în procesarea tranzacției.
22. Să se şteargă tot conținutul tabelului. Listați conținutul tabelului.
23. Să se renunțe la cea mai recentă operație de ştergere, fără a renunța la operația precedentă de introducere.

[Comanda MERGE]

24. Creați un nou tabel, numit *EMP2_PNU*, care va avea acelaşi conținut ca şi *EMPLOYEES*. Să se şteargă din tabelul *EMP2_PNU* toți angajații care câştigă comision. Să se introducă sau să se actualizeze datele din tabelul *EMP2_PNU* pe baza tabelului *employees*.