1. TEORIE

Limbajul de definire a datelor (LDD)

➔ pentru definirea structurii corespunzătoare obiectelor unei scheme: tabele, vizualizări, proceduri şi funcții stocate, declanşatori etc.

➔ Comenzile SQL care alcătuiesc **LDD** sunt:

- ❖ ***CREATE, ALTER, DROP*** - crearea, modificarea şi suprimarea obiectelor
- ❖ ***RENAME*** - modificarea numelor obiectelor unei scheme
- ❖ ***TRUNCATE*** - ştergerea datelor din obiectele unei scheme, fără suprimarea structurii obiectelor respective

➔ Implicit, o instrucțiune LDD permanentizează (COMMIT) efectul tuturor instrucțiunilor precedente şi marchează începutul unei noi tranzacții.

➔ Instrucțiunile LDD au efect imediat asupra bazei de date şi înregistrează informația în dicționarul datelor.

1. Crearea tabelelor

```
CREATE TABLE nume_tabel (
      nume_coloana_1 tip_date [DEFAULT valoare]
                  [constrangeri_nivel_coloana]
      nume_coloana_n tip_date [DEFAULT valoare]
                  [constrangeri_nivel_coloana]
[constrangeri_nivel_tabel]

);
```

sau

```
CREATE TABLE nume_tabel [(nume_coloana_1,..., nume_coloana_n)]
AS SELECT ...;
```

Constrângerile definite asupra unui tabel pot fi de următoarele tipuri:

1. ***NOT NULL*** - coloana nu poate conține valoarea null
   - se poate defini numai la nivel de coloană

```
CREATE TABLE nume_tabel (
      nume_coloana_1 tip_date
            [CONSTRAINT nume_constr] NOT NULL
      ...
);
```

2. ***UNIQUE*** - pentru coloane sau combinații de coloane care trebuie să aibă valori unice în cadrul tabelului

La nivel de coloană:

```
CREATE TABLE nume_tabel (
      nume_coloana_1 tip_date
            [CONSTRAINT nume_constr] UNIQUE
      …
);
```

La nivel de tabel:

```
CREATE TABLE nume_tabel (
      nume_coloana_1 tip_date,
      nume_coloana_2 tip_date,
      nume_coloana_3 tip_date,
      …
      [CONSTRAINT nume_constr]
            UNIQUE (nume_coloana_1, ...)
);
```

- în acest caz combinația de coloane trebuie să fie unică

3. ***PRIMARY KEY*** - identifică în mod unic orice înregistrare din tabel. Implică NOT NULL + UNIQUE;

La nivel de coloană:

```
CREATE TABLE nume_tabel (
      nume_coloana_1 tip_date [CONSTRAINT
      nume_constr] PRIMARY KEY
      …
);
```

La nivel de tabel:

```
CREATE TABLE nume_tabel (
      nume_coloana_1 tip_date,
      nume_coloana_2 tip_date,
      nume_coloana_3 tip_date,
      …
      [CONSTRAINT nume_constr]
PRIMARY KEY (nume_coloana_1, nume_coloana_2, ...)
);
```

4. ***FOREIGN KEY*** - stabileşte o relație de cheie externă între o coloană a tabelului şi o coloană dintr-un tabel specificat

```
CREATE TABLE nume_tabel (
      nume_coloana_1 tip_date,
      nume_coloana_2 tip_date,
      nume_coloana_3 tip_date,
      ...
[CONSTRAINT nume_constr]
      FOREIGN KEY (nume_coloana)
            REFERENCES nume_tabel (nume_coloana)
                  [ON DELETE CASCADE | SET NULL];
```

➔ *FOREIGN KEY* - defineşte coloana din tabelul „copil“;

➔ *REFERENCES* - identifică tabelul „părinte“ şi coloana corespunzătoare din acest tabel;

➔ *ON DELETE CASCADE* - determină ca, odată cu ştergerea unei linii din tabelul „părinte“, să fie şterse şi liniile dependente din tabelul „copil“;

➔ *ON DELETE SET NULL* determină modificarea automată a valorilor cheii externe la valoarea *null*, atunci când se şterge valoarea „părinte“.

5. ***CHECK*** - pune o condiţie pe anumite coloane
   - se poate implementa la nivel de coloană doar dacă nu referă o altă coloană a tabelului

La nivel de coloană:

```
CREATE TABLE nume_tabel (
      nume_coloana_1 tip_date [CONSTRAINT
      nume_constr] CHECK (nume_coloana_1 conditie)
      …
);
```

La nivel de tabel:

```
CREATE TABLE nume_tabel (
      nume_coloana_1 tip_date,
      nume_coloana_2 tip_date,
      nume_coloana_3 tip_date,
      …
      [CONSTRAINT nume_constr] CHECK (conditie)
);
```

Observaţii:

➔ Constrângerile pot fi create o dată cu tabelul sau adăugate ulterior cu o comandă *ALTER TABLE*.

➔ În cazul în care cheia primară (sau o cheie unică) este compusă, ea nu poate fi definită la nivel de coloane, ci doar la nivel de tabel.

Principalele ***tipuri de date*** pentru coloanele tabelelor sunt următoarele:

- **VARCHAR2 (n** [CHAR | BYTE]**)** - defineşte un şir de caractere de dimensiune variabilă, având lungimea maximă de *n* caractere / octeţi
- **CHAR (n** [CHAR | BYTE]**)** - reprezintă un şir de caractere de lungime fixă având *n* caractere / octeţi
- **NUMBER(p, s)** - reprezintă un număr având *p* cifre, dintre care *s* cifre formează partea zecimală
- **DATE** - reprezintă date calendaristice valide

2. Modificarea structurii tabelelor

➔ se face prin intermediul comenzii ALTER TABLE

Posibile modificări:

a. La nivel de coloană

i. Adăugarea unei noi coloane

➔ coloana nouă devine automat ultima în cadrul structurii tabelului

```
ALTER TABLE nume_tabel
ADD (nume_coloana tip_de_date ...);
```

ii. Modificarea unei coloane

➔ schimbarea tipului de date, a dimensiunii sau a valorii implicite a acesteia (schimbarea valorii implicite afectează numai inserările care succed modificării)

```
ALTER TABLE nume_tabel
MODIFY (coloana tip_de_date ...);
```

iii. Eliminarea unei coloane

```
ALTER TABLE nume_tabel
DROP COLUMN coloana;
```

Observaţii:

➔ dimensiunea unei coloane numerice sau de tip caracter poate fi mărită, dar nu poate fi micşorată decât dacă acea coloană conține numai valori *null* sau dacă tabelul nu conține nici o linie.

➔ tipul de date al unei coloane poate fi modificat doar dacă valorile coloanei respective sunt *null*.

➔ o coloană *CHAR* poate fi convertită la tipul de date *VARCHAR2* sau invers, numai dacă valorile coloanei sunt *null* sau dacă nu se modifică dimensiunea coloanei.

b. La nivel de constrângere

i. Adăugarea unei constrângeri

```
ALTER TABLE nume_tabel
ADD [CONSTRAINT nume_constr] tip_constr (coloana);
```

ii. Eliminarea unei constrângeri

```
ALTER TABLE nume_tabel
DROP PRIMARY KEY | UNIQUE(col1, col2, ...) |
      CONSTRAINT nume_constr;
```

iii. Activarea / Dezactivarea unei constrângeri

```
ALTER TABLE nume_tabel
MODIFY CONSTRAINT nume_constr ENABLE | DISABLE;
```

sau

```
ALTER TABLE nume_tabel
ENABLE | DISABLE CONSTRAINT nume_constr;
```

3. Suprimarea tabelelor

➔ ştergerea fizică a unui tabel, inclusiv a înregistrărilor acestuia, se realizează prin comanda:

```
DROP TABLE nume_tabel;
```

➔ pentru ştergerea conținutului unui tabel şi păstrarea structurii acestuia se poate utiliza comanda:

```
TRUNCATE TABLE nume_tabel;
```

4. Redenumirea tabelelor

➔ comanda ***RENAME*** permite redenumirea unui tabel, vizualizare sau secvență:

```
RENAME nume1_obiect TO nume2_obiect;
```

Observaţii:

➔ În urma redenumirii sunt transferate automat constrângerile de integritate, indecşii şi privilegiile asupra vechilor obiecte.

5. Consultarea dicţionarului datelor

Informații despre tabelele create se găsesc în vizualizările:

- **USER_TABLES** - informații complete despre tabelele utilizatorului.
- **TAB** - informații de bază despre tabelele existente în schema utilizatorului.

Informații despre constângeri găsim în **USER_CONSTRAINTS**, iar despre coloanele implicate în constrângeri în **USER_CONS_COLUMNS**.

2. EXERCIŢII

1. Să se creeze tabelul ANGAJATI_pnu (pnu se alcatuieşte din prima literă din prenume şi primele două din numele studentului) corespunzător schemei relaționale:

ANGAJATI_pnu(cod_ang number(4),
nume varchar2(20),
prenume varchar2(20),
email char(15),
data_ang date,
job varchar2(10),
cod_sef number(4),
salariu number(8, 2),
cod_dep number(2)
);

în următoarele moduri:

a) fără precizarea vreunei chei sau constrângeri;
b) cu precizarea cheilor primare la nivel de coloană si a constrangerilor NOT NULL pentru coloanele nume şi salariu;
c) cu precizarea cheii primare la nivel de tabel si a constrângerilor NOT NULL pentru coloanele nume şi salariu.

Se presupune că valoarea implicită a coloanei data_ang este SYSDATE.

Obs: Nu pot exista două tabele cu acelaşi nume în cadrul unei scheme, deci recrearea unui tabel va fi precedată de suprimarea sa.

2. Adăugați următoarele înregistrări în tabelul ANGAJATI_pnu:

Cod_ang	Nume	Prenume	Email	Data_ang	Job	Cod_sef	Salariu	Cod_dep
100	Nume1	Prenume1	Null	Null	Director	null	20000	10
101	Nume2	Prenume2	Nume2	02-02-2004	Inginer	100	10000	10
102	Nume3	Prenume3	Nume3	05-06-2000	Analist	101	5000	20
103	Nume4	Prenume4	Null	Null	Inginer	100	9000	20
104	Nume5	Prenume5	Nume5	Null	Analist	101	3000	30

Prima si a patra înregistrare vor fi introduse specificând coloanele pentru care introduceți date efectiv, iar celelalte vor fi inserate fără precizarea coloanelor în comanda INSERT.

Salvați comenzile de inserare.

3. Creați tabelul ANGAJATI10_pnu, prin copierea angajaților din departamentul 10 din tabelul ANGAJATI_pnu.
4. Introduceti coloana comision în tabelul ANGAJATI_pnu. Coloana va avea tipul de date NUMBER(4,2).
5. Este posibilă modificarea tipului coloanei salariu în NUMBER(6,2)?
6. Setați o valoare DEFAULT pentru coloana salariu.
7. Modificați tipul coloanei comision în NUMBER(2, 2) şi al coloanei salariu la NUMBER(10, 2), în cadrul aceleiaşi instrucțiuni.
8. Actualizati valoarea coloanei comision, setând-o la valoarea 0.1 pentru salariații al căror job începe cu litera A.

9. Modificați tipul de date al coloanei email în VARCHAR2.
10. Adăugați coloana nr_telefon în tabelul ANGAJATI_pnu, setându-i o valoare implicită.
11. Vizualizați înregistrările existente. Suprimați coloana nr_telefon. Ce efect ar avea o comandă ROLLBACK în acest moment?
12. Redenumiți tabelul ANGAJATI_pnu în ANGAJATI3_pnu.
13. Consultați vizualizarea TAB din dicționarul datelor. Redenumiți angajati3_pnu în angajati_pnu.
14. Suprimați conținutul tabelului angajati10_pnu, fără a suprima structura acestuia.
15. Creați şi tabelul DEPARTAMENTE_pnu, corespunzător schemei relaționale:

```
DEPARTAMENTE_pnu( cod_dep number(2),
                  nume varchar2(15),
                  cod_director number(4)
);
```

specificând doar constrângerea NOT NULL pentru nume (nu precizați deocamdată constrângerea de cheie primară).

16. Introduceți următoarele înregistrări în tabelul DEPARTAMENTE_pnu:

Cod_dep	Nume	Cod_director
10	Administrativ	100
20	Proiectare	101
30	Programare	Null

17. Precizați cheia primară cod_dep, fără suprimarea şi recreerea tabelului.
18. Să se precizeze constrângerea de cheie externă pentru coloana cod_dep din ANGAJATI_pnu:
    a. fără suprimarea tabelului;
    b. prin suprimarea şi recrearea tabelului, cu precizarea noii constrângeri la nivel de coloană. De asemenea, se vor mai preciza constrângerile (la nivel de coloană, dacă este posibil):
        - PRIMARY KEY pentru cod_ang;
        - FOREIGN KEY pentru cod_sef;
        - UNIQUE pentru combinația nume + prenume;
        - UNIQUE pentru email;
        - NOT NULL pentru nume;
        - verificarea cod_dep >0;
        - verificarea ca salariul sa fie mai mare decat comisionul * 100.
19. Reintroduceți date în tabel, utilizând (şi modificând, dacă este necesar) comenzile salvate anterior.
20. Ce se întâmplă dacă se încearcă suprimarea tabelului departamente_pnu?
21. Analizați structura vizualizărilor USER_TABLES, TAB, USER_CONSTRAINTS.
22. a) Listați informațiile relevante (cel puțin nume, tip şi tabel) despre constrângerile asupra tabelelor angajati_pnu şi departamente_pnu.

Obs: Tipul constrângerilor este marcat prin:

- P - pentru cheie primară
- R – pentru constrângerea de integritate referențială (cheie externă);
- U – pentru constrângerea de unicitate (UNIQUE);
- C – pentru constrângerile de tip CHECK.

b) Aflați care sunt coloanele la care se referă constrângerile asupra tabelelor *angajati_pnu* şi *departamente_pnu*.

23. Încercaţi să introduceți constrângerea NOT NULL asupra coloanei email. Dacă nu este posibil, modificaţi tabelul astfel încât să se poata aplica această constrângere.
24. Încercați să adăugați o nouă înregistrare în tabelul ANGAJATI_pnu, care să corespundă codului de departament 50. Se poate?
25. Adăugați un nou departament, cu numele Analiza, codul 60 şi directorul null în DEPARTAMENTE_pnu. COMMIT.
26. Încercați să ştergeți departamentul 20 din tabelul DEPARTAMENTE_pnu. Comentați.
27. Ştergeți departamentul 60 din DEPARTAMENTE_pnu. ROLLBACK.
28. Încercați să introduceți un nou angajat, specificând valoarea 114 pentru cod_sef. Ce se obține? Adăugați un nou angajat, având codul 114. Incercați din nou introducerea înregistrării.
29. Se doreşte ştergerea automată a angajaților dintr-un departament, odată cu suprimarea departamentului. Pentru aceasta, este necesară introducerea clauzei ON DELETE CASCADE în definirea constrângerii de cheie externă. Suprimați constrângerea de cheie externă asupra tabelului ANGAJATI_pnu şi reintroduceți această constrângere, specificând clauza ON DELETE CASCADE.
30. Ştergeți departamentul 20 din DEPARTAMENTE_pnu. Ce se întâmplă? Rollback.
31. Introduceți constrângerea de cheie externă asupra coloanei *cod_director* a tabelului DEPARTAMENTE_pnu. Se doreşte ca ştergerea unui angajat care este director de departament să atragă după sine setarea automată a valorii coloanei cod_director la *null*.
32. Actualizați tabelul DEPARTAMENTE_PNU, astfel încât angajatul având codul 102 să devină directorul departamentului 30. Ştergeți angajatul având codul 102 din tabelul ANGAJATI_pnu. Analizați efectele comenzii. Rollback.

    Este posibilă suprimarea angajatului având codul 101? Comentați.
33. Adăugați o constrângere de tip *check* asupra coloanei salariu, astfel încât acesta să nu poată depăşi 30000.
34. Incercați actualizarea salariului angajatului 100 la valoarea 35000.
35. Dezactivați constrângerea creată anterior şi reîncercați actualizarea. Ce se întâmplă dacă încercăm reactivarea constrângerii?