LABORATOR 7 SQL

Limbajul de definire a datelor (CREATE, ALTER, DROP)

O bază de date *Oracle* poate conţine mai multe structuri de date. În general, instrucţiunile *LDD* sunt utilizate pentru definirea structurii corespunzătoare obiectelor unei scheme: tabele, vizualizări, vizualizări materializate, indecşi, sinonime, *cluster*-e, proceduri şi funcţii stocate, declanşatori, pachete, legături între baze de date etc.

Instrucţiunile *LDD* permit:

- crearea, modificarea şi suprimarea obiectelor unei scheme şi a altor obiecte ale bazei de date, inclusiv baza însăşi şi utilizatorii acesteia (CREATE, ALTER, DROP);
- modificarea numelor obiectelor unei scheme (RENAME);
- ştergerea datelor din obiectele unei scheme, fără suprimarea structurii obiectelor respective (TRUNCATE).

Implicit, o instrucţiune *LDD* permanentizează (COMMIT) efectul tuturor instrucţiunilor precedente şi marchează începutul unei noi tranzacţii.

Instrucţiunile *LDD* au efect imediat asupra bazei de date şi înregistrează informaţia în dicţionarul datelor.

Reguli de numire a obiectelor bazei de date

- Identificatorii obiectelor trebuie să înceapă cu o literă şi să aibă maxim 30 de caractere, cu excepţia numelui bazei de date care este limitat la 8 caractere şi celui al legăturii unei baze de date, a cărui lungime poate atinge 128 de caractere.
- Numele poate conţine caracterele A-Z, a-z, 0-9, _, $ şi #.
- Două obiecte ale aceluiaşi utilizator al *server*-ului *Oracle* nu pot avea acelaşi nume.
- Identificatorii nu pot fi cuvinte rezervate ale *server*-ului *Oracle*.
- Identificatorii obiectelor nu sunt *case-sensitive*.

Crearea tabelelor

```
CREATE TABLE [schema.]nume_tabel (
    nume_coloana tip_de_date [DEFAULT expr], ...);

CREATE TABLE nume_tabel [(col1, col2...)]
   AS subcerere;
```

1. **a.** Creaţi tabelul *salariat_**** având următoarea structură:

Nume	Caracteristici	Tip
cod_ang	NOT NULL	NUMBER(4)
nume		VARCHAR2(25)

prenume		VARCHAR2(25)
functia		VARCHAR2(20)
sef		NUMBER(4)
data_angajarii	valoare implicită data curentă	DATE
varsta		NUMBER(2)
email		CHAR(20)
salariu	NOT NULL, valoare implicită 0	NUMBER(9,2)

```
CREATE TABLE salariat_*** (
  cod_ang         NUMBER(4) NOT NULL,
  nume            VARCHAR2(25),
  prenume         VARCHAR2(25),
  functia         VARCHAR2(20),
  sef             NUMBER(4),
  data_angajarii  DATE DEFAULT SYSDATE,
  varsta          NUMBER(2),
  email           CHAR(20),
  salariu         NUMBER(9,2) DEFAULT 0 NOT NULL);
```

b. Afişaţi structura tabelului creat anterior.

2. Se dau următoarele informaţii:

cod _ang	nume	prenume	functia	sef	data_angajarii	varsta	email	salariu
1	...	...	director	null	...	30	...	5500
2	...	...	economist	1	...	25	...	0
3	...	...	functionar	1	...	45	...	3000
4	...	...	economist	1	...	35	...	1000

a. Inseraţi în tabelul *salariat_*** prima înregistrare din tabelul de mai sus fără să precizaţi lista de coloane în comanda *INSERT*.

b. Inseraţi a doua înregistrare, folosind o listă de coloane din care excludeţi coloanele *data_angajarii* şi *salariu*. Observaţi apoi rezultatul.

c. Inserați celelalte 2 înregistrări.

3. Creaţi tabelul *economist_*** care să conţină economiştii din tabelul *salariat_***, având următoarele coloane: codul, numele, salariul anual şi data angajării. Verificaţi cum a fost creat tabelul şi ce date conţine.

Modificarea tabelelor

Modificarea structurii unui tabel se realizează cu ajutorul comenzii ***ALTER TABLE*** şi poate consta în:

- **adăugarea unei coloane noi**
  - nu se poate specifica poziţia unei coloane noi în structura tabelului;
  - o coloană nouă devine automat ultima în cadrul structurii tabelului.

```
ALTER TABLE nume_tabel
ADD (coloana tip_de_date [NOT NULL] [DEFAULT] expr][, ...]);
```

- **redenumirea unei coloane**

```
ALTER TABLE nume_tabel
RENAME COLUMN nume_vechi TO nume_nou;
```

- **modificarea unei coloane**
  - schimbarea tipului de date, a dimensiunii sau a valorii implicite a acesteia; schimbarea valorii implicite afectează numai inserările care succed modificării
  - dimensiunea unei coloane numerice sau de tip şir de caractere poate fi mărită, dar nu poate fi micşorată decât dacă acea coloană conţine numai valori *null* sau dacă tabelul nu conţine nici o linie
  - tipul de date al unei coloane poate fi modificat doar dacă valorile coloanei respective sunt *null*

```
ALTER TABLE nume_tabel
MODIFY (coloana tip_de_date [NOT NULL | NULL]
                            [DEFAULT expr][, ...]);
```

- **eliminarea unei coloane**

```
ALTER TABLE nume_tabel
DROP COLUMN coloana;

ALTER TABLE nume_tabel
DROP (coloana1, coloana2, …);
```

4. Adăugaţi o nouă coloană tabelului *salariat_**** care să conţină data naşterii.

5. Modificaţi dimensiunea coloanei *nume* la 30 şi pe cea a salariului la 12 cu 3 zecimale.

6. Modificaţi tipul coloanei *email* la *VARCHAR2*.

7. Modificaţi valoarea implicită a coloanei *data_angajarii* la data sistemului+ o zi.

8. Modificaţi numele coloanei *varsta* din tabelul *salariat_**** cu *varsta_ang.*

9. Eliminaţi coloana *varsta_ang* din tabelul *salariat_****.

Eliminarea tabelelor

- Ştergerea fizică a unui tabel, inclusiv a înregistrărilor acestuia, se realizează prin comanda

```
DROP TABLE nume_tabel;
```

- Pentru ştergerea conţinutului unui tabel şi păstrarea structurii acestuia se poate utiliza comanda:

```
TRUNCATE TABLE nume_tabel;
```

Observaţie: Fiind operaţie LDD, comanda *TRUNCATE* are efect definitiv.

10. Ştergeţi conţinutul tabelului *economist_**** folosind comanda TRUNCATE. Verificaţi că tabelul nu conţine date. După utilizarea comenzii ROLLBACK, tabelul va conţine datele iniţiale?

Redenumirea tabelelor

Comanda ***RENAME*** permite redenumirea unui tabel, vizualizare, secvenţă sau sinonim privat.

```
RENAME nume1_obiect TO nume2_obiect;
```

- În urma redenumirii sunt transferate automat constrângerile de integritate, indecşii şi privilegiile asupra vechilor obiecte.
- Sunt invalidate toate obiectele ce depind de obiectul redenumit, cum ar fi vizualizări, sinonime, proceduri sau funcţii stocate.

11. Redenumiţi tabelul *economist_**** cu *eco_****.

Constrângeri

Constrângerile pot fi create cu tabelul sau adăugate ulterior cu o comandă ALTER TABLE.
Tipuri de constrângeri:

- **constrângerea de validare**
  - implicaţie: coloana sau o expresie de coloane trebuie să verifice condiţia specificată.
  - sintaxa: `CHECK(conditie)`
- **constrângerea *not null***
  - implicaţie: coloana nu poate conţine valori *null*
  - sintaxa: `NOT NULL`
  - echivalenţă: `CHECK(coloană IS NOT NULL)`
- **constrângerea de unicitate**
  - implicaţie: coloana sau o combinaţie de coloane nu poate conţine valori duplicate
  - sintaxa: `UNIQUE(col1, col2, …)`
- **constrângerea de cheie primară**
  - implicaţie: coloana sau o combinaţie de coloane nu poate conţine valori duplicate sau valori *null*

- scop: se identifică în mod unic orice înregistrare din tabel
- echivalenţă: NOT NULL + UNIQUE;
- sintaxa: `PRIMARY KEY(col1, col2, … )`

- **constrângerea de cheie externă**
  - scop: stabileşte o relaţie de tip *copil – părinte* (*many-to-one*) între o coloană a tabelului şi o altă coloană (declarată cheie primară) dintr-un tabel specificat
  - implicaţii:
    - dacă această constrângere este declarată fără opţiunile ON DELETE CASCADE sau ON DELETE SET NULL atunci din tabelul *părinte* nu pot fi şterse înregistrări dacă acestea au înregistrări corespondente în tabelul *copil*
    - în tabelul *copil* coloana declarată cheie externă poate să aibă valoarea *null* sau o valoare care are corespondent în tabelul *părinte*
  - sintaxa:
```
[FOREIGN KEY nume_col]
  REFERENCES nume_tabel(nume_coloana)
  [ON DELETE {CASCADE| SET NULL}]
```
    - *FOREIGN KEY* se utilizează într-o constrângere la nivel de tabel pentru a defini coloana din tabelul *copil*;
    - *REFERENCES* identifică tabelul *părinte* şi coloana corespunzătoare din acest tabel;
    - *ON DELETE CASCADE* determină ca odată cu ştergerea unei înregistrări din tabelul *părinte* să fie şterse şi înregistrările dependente din tabelul *copil*;
    - *ON DELETE SET NULL* determină modificarea automată a valorilor cheii externe din tabelul *copil* la valoarea *null*, atunci când se şterge valoarea *părinte*.

Adăugarea constrângerilor la crearea tabelului (CREATE TABLE)

```
CREATE TABLE [schema.]nume_tabel (
      nume_coloana1 tip_de_date [DEFAULT expr]
           [constrângere la nivel de coloană],
      nume_coloana2 tip_de_date [DEFAULT expr]
           [constrângere la nivel de coloană, ...]
      [constrângere1 la nivel de tabel,
       constrângere2 la nivel de tabel, ...])
```

12. Ştergeţi şi apoi creaţi din nou tabelul *salariat_*** cu următoarea structură:

NUME	TIP	CONSTRÂNGERE
cod_ang	NUMBER(4)	Cheie primară
nume	VARCHAR2(25)	Nu permite valori *null*
prenume	VARCHAR2(25)	

data_nasterii	DATE	data_nasterii<data_angajarii
functia	VARCHAR2(9)	Nu permite valori *null*
sef	NUMBER(4)	Cheie externă care referă coloana *cod_ang* din acelaşi tabel
data_angajarii	DATE	
email	VARCHAR2(20)	Nu permite duplicate
salariu	NUMBER(12,3)	Nu permite valori negative
cod_dept	NUMBER(4)	
		Combinaţia dintre *nume*, *prenume* şi *data naşterii* să fie unică.

Observaţie: Constrângerile care referă mai mult de o coloană se poate declara doar la nivel de tabel.

```
CREATE TABLE salariat_*** (
  cod_ang  NUMBER(4) PRIMARY KEY,
  nume     VARCHAR2(25) NOT NULL,
  prenume  VARCHAR2(25),
  data_nasterii DATE,
  functia  VARCHAR2(9) NOT NULL,
  sef      NUMBER(4) REFERENCES salariat_*** (cod_ang),
  data_angajarii DATE DEFAULT SYSDATE,
  email    VARCHAR2(20) UNIQUE,
  salariu  NUMBER(9,2) CONSTRAINT ck1_*** CHECK(salariu > 0),
  cod_dep  NUMBER(4),
  CONSTRAINT ck2_*** CHECK(data_angajarii > data_nasterii),
  CONSTRAINT u_*** UNIQUE(nume, prenume, data_nasterii));
```

13. Ştergeţi tabelul *salariat_****, iar apoi recreaţi-l implementând toate constrângerile la nivel de tabel.

Observaţie: Constrângerea de tip NOT NULL se poate declara doar la nivel de coloană.

```
CREATE TABLE salariat_*** (
cod_ang NUMBER(4),
nume    VARCHAR2(25) NOT NULL,
prenume VARCHAR2(25),
data_nasterii DATE,
functia VARCHAR2(9) NOT NULL,
sef     NUMBER(4),
data_angajarii DATE DEFAULT SYSDATE,
email   VARCHAR2(20),
salariu NUMBER(9,2),
```

```
cod_dep NUMBER(4),
CONSTRAINT pk_***  PRIMARY KEY(cod_ang),
CONSTRAINT fk1_*** FOREIGN KEY(sef) REFERENCES salariat_***(cod_ang),
CONSTRAINT u1_***  UNIQUE(email),
CONSTRAINT ck1_*** CHECK(data_angajarii > data_nasterii),
CONSTRAINT ck2_*** CHECK(salariu > 0),
CONSTRAINT u2_***  UNIQUE(nume,prenume,data_nasterii));
```

14. **a**. Creaţi tabelul *departament_**** cu următoarea structură.

NUME	TIP	CONSTRÂNGERI
cod_dep	NUMBER(4)	Cheie primară
nume	VARCHAR2(20)	
oras	VARCHAR2(25)	Nu permite valori *null*.

b. Adăugaţi constrângerea NOT NULL pe coloana *nume*.

```
ALTER TABLE departament_***
MODIFY nume NOT NULL;
```

c. Eliminaţi constrângerea NOT NULL definită pe coloana *oras*.

Adăugarea constrângerilor ulterior creării tabelului.
Eliminarea, activarea sau dezactivarea constrângerilor

- **Adăugare constrângere**

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

- **Eliminare constrângere**

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

- **Activare/dezactivare constrângere**

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

ALTER TABLE nume_tabel
ENABLE|DISABLE nume_constr;
```

Observaţie: Comanda *ALTER TABLE* în variatele date mai sus nu se aplică pentru constrângerile de tip *NOT NULL*.

15. Inseraţi o nouă înregistrare în *salariat_**** de forma următoare:

cod_ang	nume	prenume	data_nasterii	functia	sef	data_angajarii	email	salariu	cod_dep
2	N2	P2	11-JUN-1960	director	1	sysdate	E2	20000	10

Ce observaţi? Introduceţi înregistrarea, dar specificând valoarea *null* pentru coloana *sef*.

16. Încercaţi să adăugaţi o constrângere de cheie externă pe coloana *cod_dep* din tabelul *salariat_****. Ce observaţi?

```
ALTER TABLE salariat_***
ADD CONSTRAINT fk2_*** FOREIGN KEY(cod_dep) REFERENCES
    departament_***(cod_dep);
```

17. Inseraţi o nouă înregistrare în *departament_****. Apoi, adăugaţi constrângerea de cheie externă definită anterior.

cod_dep	nume	oras
10	Economic	Bucuresti

18. Inseraţi noi înregistrări în tabelul *salariat_****, respectiv în tabelul *departament_****. Care trebuie să fie ordinea de inserare?

cod_ang	nume	prenume	data_nasterii	functia	sef	data_angajarii	email	salariu	cod_dep
3	N3	P3	11-JUN-1967	jurist	2	sysdate	E3	5000	20

cod_dep	nume	oras
20	Juridic	Constanta

19. Ştergeţi departamentul 20 din tabelul *departament_****. Ce observaţi?

20. Ştergeţi constrângerea *fk2_****. Recreaţi această constrângere adăugând opţiunea *ON DELETE CASCADE*.

```
ALTER TABLE salariat_***
DROP CONSTRAINT fk2_***;

ALTER TABLE salariat_***
ADD CONSTRAINT fk2_*** FOREIGN KEY(cod_dep) REFERENCES
    departament_***(cod_dep) ON DELETE CASCADE;
```

21. Ştergeţi departamentul 20 din tabelul *departament_****. Ce observaţi în tabelul *salariat_****? Anulaţi modificările.

22. Ştergeţi constrângerea *fk2_****. Recreaţi această constrângere adăugând opţiunea *ON DELETE SET NULL*.

```
ALTER TABLE salariat_***
DROP CONSTRAINT fk2_***;

ALTER TABLE salariat_***
ADD CONSTRAINT fk2_*** FOREIGN KEY (cod_dep) REFERENCES
    departament_***(cod_dep) ON DELETE SET NULL;
```

23. Ştergeţi departamentul 10 din tabelul *departament_****. Ce observaţi în tabelul *salariat_****? Anulaţi modificările.

Consultarea dicţionarului datelor

Tipuri de vizualizări ale dicţionarului datelor:

- *USER_** - obiecte definite de utilizatorul curent;
- *ALL_** - obiecte la care are acces utilizatorul curent;
- *DBA_** - obiecte la care are acces administratorul bazei de date.

Informaţii despre tabelele create se găsesc în vizualizările:

- *USER_TABLES* – informaţii complete despre tabelele utilizatorului curent;
- *COLS* – informaţii despre coloanele tabelelor;
- *TAB* – informaţii de bază despre tabelele şi vizualizările utilizatorului curent.

Informaţii despre constrângerile definite de utilizatorul curent se găsesc în vizualizarea:

- *USER_CONSTRAINTS*

Informaţii despre coloanele asociate unei constrângeri:

- *USER_CONS_COLUMNS*

24. a. Afişaţi structura vizualizării *USER_TABLES.*
b. Afişaţi numele tabelelor create.

25. a. Afişaţi structura vizualizării *COLS.*
b. Afişaţi numele şi tipul de date al coloanelor tabelului *departament_**** .

26. a. Afişaţi structura vizualizării *USER_CONSTRAINTS.*
b. Afişaţi informaţii despre constrângerile definite asupra tabelelor *departament_**** şi *salariat_****.

27. a. Afişaţi structura vizualizării *USER_CONS_COLUMNS.*
b. Afişaţi numele tabelului şi coloana pe care este definită constrângerea fk2_***.
c. Modificaţi cererea anterioară astfel încât să afişaţi şi informaţii suplimentare despre constrângerea fk2_***.

TEMĂ

28. **a.** Creaţi copii pentru tabelele *work* şi *projects*, denumite *work_*** şi *projects_***.

b. Adăugaţi constrângerile de cheie externă pentru tabelul *work_*** (*projects_*** şi *emp_***).

c. Considerând că un angajat poate să lucreze în cadrul unui proiect doar o singură perioadă de timp, adăugaţi constrângerea de cheie primară tabelului *work_***.

d. Fără a specifica numele constrângerii, eliminaţi constrângerea de cheie primară adăugată tabelului *work_***.

e. Considerând că un angajat poate să lucreze în cadrul unui proiect în mai multe perioade de timp, adăugaţi constrângerea de cheie primară tabelului *work_***.

f. Inserați o nouă înregistrare în tabela *projects_***, respectând constrângerile impuse.

g. Inserați o nouă înregistrare în tabela *work_***, respectând constrângerile impuse.