

LABORATOR 7 SQL

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

O bază de date *Oracle* poate conține mai multe structuri de date. În general, instrucțiunile *LDD* sunt utilizate pentru definirea structurii corespunzătoare obiectelor unei scheme: tabele, vizualizări, vizualizări materializate, indecși, sinonime, *cluster*-e, proceduri și funcții stocate, declanșatori, pachete, legături între baze de date etc.

Instrucțiunile *LDD* permit:

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

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

Instrucțiunile *LDD* au efect imediat asupra bazei de date și înregistrează informația în dicționarul datelor.

Reguli de numire a obiectelor bazei de date

- Identificatorii obiectelor trebuie să înceapă cu o literă și să aibă maxim 30 de caractere, cu excepția numelui bazei de date care este limitat la 8 caractere și celui al legăturii unei baze de date, a cărui lungime poate atinge 128 de caractere.
- Numele poate conține caracterele A-Z, a-z, 0-9, _, \$ și #.
- Două obiecte ale aceluiași utilizator al *server*-ului *Oracle* nu pot avea același 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. Creați tabelul *timp_** care va conține următoarele coloane:**

- *data1*, care va stoca data calendaristică și timpul (ore, minute, secunde);
- *data2*, care va stoca data calendaristică și timpul (ore, minute, secunde, fracțiuni de secunde);

- *data3*, care va stoca data calendaristică, timpul (ore, minute, secunde, fracțiuni de secunde) și diferența de fus orar față de ora universală.
- b.** Inserați o linie în acest tabel cu valorile curente ale datei calendaristice, apoi afișați informațiile inserate.

```
CREATE TABLE timp_***
(data1 DATE,
 data2 TIMESTAMP(5),
 data3 TIMESTAMP(5) WITH TIME ZONE);

INSERT INTO timp_***
VALUES(sysdate, systimestamp, systimestamp);
```

- 2. a.** Creați 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.** Afișați structura tabelului creat anterior.

3. Se dau următoarele informații:

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

- Inserați în tabelul *salariat_**** prima înregistrare din tabelul de mai sus fără să precizați lista de coloane în comanda *INSERT*.
 - Inserați a doua înregistrare, folosind o listă de coloane din care excludeți coloanele *data_angajarii* și *salariu*. Observați apoi rezultatul.
 - Creați un *script* care să permită inserarea de valori de la tastatură cu ajutorul căruia inserați următoarele două înregistrări.
4. Creați tabelul *economist_**** care să conțină economiștii din tabelul *salariat_****, având următoarele coloane: codul, numele, salariul anual și data angajării. Verificați cum a fost creat tabelul și ce date conține.

Modificarea tabelelor

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

- **adăugarea unei coloane noi**

- nu se poate specifica poziția 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 șir de caractere poate fi mărită, dar nu poate fi micșorată 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*

```
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, ...);
```

5. Adăugați o nouă coloană tabelului *salariat_**** care să conțină data nașterii.
6. Modificați dimensiunea coloanei *nume* la 30 și pe cea a salariului la 12 cu 3 zecimale.
7. Modificați tipul coloanei *email* la *VARCHAR2*.
8. Modificați valoarea implicită a coloanei *data_angajarii* la data sistemului+ o zi.
9. Modificați numele coloanei *varsta* din tabelul *salariat_**** cu *varsta_ang*.
10. Eliminați coloana *varsta_ang* din tabelul *salariat_****.

Eliminarea tabelelor

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

```
DROP TABLE nume_tabel;
```

- Pentru ștergerea conținutului unui tabel și păstrarea structurii acestuia se poate utiliza comanda:

```
TRUNCATE TABLE nume_tabel;
```

Observație: Fiind operație LDD, comanda *TRUNCATE* are efect definitiv.

11. Ștergeți conținutul tabelului *timp_**** folosind comanda *TRUNCATE*. Verificați că tabelul nu conține date. După utilizarea comenzii *ROLLBACK*, tabelul va conține datele inițiale?

Redenumirea tabelelor

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

```
RENAME nume1_obiect TO nume2_obiect;
```

- În urma redenumirii sunt transferate automat constrângerile de integritate, indecșii și privilegiile asupra vechilor obiecte.
 - Sunt invalidate toate obiectele ce depind de obiectul redenumit, cum ar fi vizualizări, sinonime, proceduri sau funcții stocate.
12. Redenumiți 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**
 - implicație: coloana sau o expresie de coloane trebuie să verifice condiția specificată.
 - sintaxa: CHECK (conditie)
- **constrângerea not null**
 - implicație: coloana nu poate conține valori *null*
 - sintaxa: NOT NULL
 - echivalentă: CHECK (coloană IS NOT NULL)
- **constrângerea de unicitate**
 - implicație: coloana sau o combinație de coloane nu poate conține valori duplicate
 - sintaxa: UNIQUE (col1, col2, ...)
- **constrângerea de cheie primară**
 - implicație: coloana sau o combinație de coloane nu poate conține valori duplicate sau valori *null*
 - scop: se identifică în mod unic orice înregistrare din tabel
 - echivalentă: NOT NULL + UNIQUE;
 - sintaxa: PRIMARY KEY (col1, col2, ...)
- **constrângerea de cheie externă**
 - scop: stabilește o relație de tip *copil – părinte (many-to-one)* între o coloană a tabelului și o altă coloană (declarată cheie primară) dintr-un tabel specificat
 - implicații:
 - dacă această constrângere este declarată fără opțiunile ON DELETE CASCADE sau ON DELETE SET NULL atunci din tabelul *părinte* nu pot fi șterse 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* și coloana corespunzătoare din acest tabel;
 - *ON DELETE CASCADE* determină ca odată cu ștergerea unei înregistrări din tabelul *părinte* să fie șterse și 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 șterge 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, ...])
```

13. Ștergeți și apoi creați din nou tabelul *salariat_**** cu următoarea structură:

NUME	TIP	CONSTRÂNGERE
cod_ang	NUMBER(4)	Cheie primară
nume	VARCHAR2(25)	Nu permite valori <i>null</i>
prenume	VARCHAR2(25)	
data_nasterii	DATE	data_nasterii < data_angajarii
functia	VARCHAR2(9)	Nu permite valori <i>null</i>
sef	NUMBER(4)	Cheie externă care referă coloana <i>cod_ang</i> din același tabel
data_angajarii	DATE	
email	VARCHAR2(20)	Nu permite duplicate
salariu	NUMBER(12,3)	Nu permite valori negative
cod_dept	NUMBER(4)	
		Combinatia dintre <i>nume</i> , <i>prenume</i> și <i>data nașterii</i> să fie unică.

Observație: 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(numa, prenuma, data_nasterii));
```

14. Ștergeți tabelul *salariat_****, iar apoi recreați-l implementând toate constrângerile la nivel de tabel.

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

```
CREATE TABLE salariat_*** (
cod_ang  NUMBER(4),
numa     VARCHAR2(25) NOT NULL,
prenuma  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(numa,prenuma,data_nasterii));
```

15. a. Creați tabelul *departament_**** cu următoarea structură.

NUME	TIP	CONSTRÂNGERI
cod_dep	NUMBER(4)	Cheie primară
numa	VARCHAR2(20)	
oras	VARCHAR2(25)	Nu permite valori <i>null</i> .

- b. Adăugați constrângerea NOT NULL pe coloana *numa*.

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

- c. Eliminați 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;
```

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

16. Inserați 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 observați? Introduceți înregistrarea, dar specificând valoarea *null* pentru coloana *sef*.

17. Încercați să adăugați o constrângere de cheie externă pe coloana *cod_dep* din tabelul *salariat_****. Ce observați?

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

18. Inserați o nouă înregistrare în *departament_****. Apoi, adăugați constrângerea de cheie externă definită anterior.

cod_dep	nume	oras
10	Economic	Bucuresti

19. Inserați 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

20. Ștergeți departamentul 20 din tabelul *departament_****. Ce observați?
21. Ștergeți constrângerea *fk2_****. Recreați această constrângere adăugând opțiunea *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;
```
22. Ștergeți departamentul 20 din tabelul *departament_****. Ce observați în tabelul *salariat_****? Anulați modificările.
23. Ștergeți constrângerea *fk2_****. Recreați această constrângere adăugând opțiunea *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;
```
24. Ștergeți departamentul 10 din tabelul *departament_****. Ce observați în tabelul *salariat_****? Anulați modificările.
25. a. Creați copii pentru tabelele *work* și *projects*, denumite *work_**** și *projects_****.
b. Adăugați constrângerile de cheie externă pentru tabelul *work_**** (*projects_**** și *emp_****).
c. Considerând că un angajat poate să lucreze în cadrul unui proiect doar o singură perioadă de timp, adăugați constrângerea de cheie primară tabelului *work_****.
d. Fără a specifica numele constrângerii, eliminați 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, adăugați constrângerea de cheie primară tabelului *work_****.

Consultarea dicționarului datelor

Tipuri de vizualizări ale dicționarului 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.

Informații despre tabelele create se găsesc în vizualizările:

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

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

- *USER_CONSTRAINTS*

Informații despre coloanele asociate unei constrângeri:

- *USER_CONS_COLUMNS*

28. a. Afișați structura vizualizării *USER_TABLES*.
b. Afișați numele tabelelor create.
29. a. Afișați structura vizualizării *COLS*.
b. Afișați numele și tipul de date al coloanelor tabelului *dept_****.
30. a. Afișați structura vizualizării *USER_CONSTRAINTS*.
b. Afișați informații despre constrângerile definite asupra tabelelor *departments* și *work_***.
31. a. Afișați structura vizualizării *USER_CONS_COLUMNS*.
b. Afișați numele tabelului și coloana pe care este definită constrângerea *fk2_****.
c. Modificați cererea anterioară astfel încât să afișați și informații suplimentare despre constrângerea *fk2_****.