

LABORATOR 8

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, clustere, proceduri și funcții stocate, declanșatori, pachete stocate 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.

Definirea unui obiect presupune : crearea (CREATE), modificarea (ALTER) și suprimarea sa (DROP).

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;
```

Exercițiul 1. Creați tabelul *salariat_**** având următoarea structură:

| Nume | Caracteristici | Tip |
|----------------|--------------------------------|--------------|
| Cod_angajat | 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 |
| email | | CHAR(10) |
| salariu | Valoare implicită 0 | NUMBER(9,2) |

Exercițiul 2. Afișați structura tabelului creat anterior.

Se dau următoarele valori:

| COD | NUME | PRENUME | FUNCTIA | SEF | DATA_ANG | VARSTA | EMAIL | SALARIU |
|-----|-------|---------|------------|------|----------|--------|-------|---------|
| 1 | | | Director | null | | 30 | null | 5500 |
| 2 | | | functionar | 1 | | 25 | | 0 |
| 3 | | | economist | 1 | | 45 | | 3000 |
| 4 | | | functionar | 1 | | 35 | | 1000 |

Exercițiul 3. Inserați în tabelul salariat_*** prima înregistrare din tabelul de mai sus fără să precizați lista de coloane în comanda INSERT.

Exercițiul 4. Inserați a doua înregistrare folosind o listă de coloane din care excludeți data_angajarii și salariul care au valori implicite. Observați apoi rezultatul.

Exemplul 5. Creați tabelul functionar_*** care să conțină funcționarii 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 face cu ajutorul comenzii **ALTER TABLE**.

Modificarea poate consta în:

- adăugarea unei noi coloane (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 [DEFAULT expr][, ...]);
```

- 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 [DEFAULT expr][, ...]);
```

- eliminarea unei coloane:

```
ALTER TABLE nume_tabel
```

```
DROP COLUMN coloana;
```

sau

```
ALTER TABLE nume_tabel
```

```
DROP (coloana);
```

Observații:

- dimensiunea unei coloane numerice sau de tip caracter 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*.
- 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.

Exercițiul 6. Adăugați o nouă coloană tabelului salariat_*** care să conțină data nașterii.

Exercițiul 7. Modificați dimensiunea coloanei *nume* la 30 și pe cea a salariului la 12 cu 3 zecimale.

Exercițiul 8. Modificați tipul coloanei *email* la *VARCHAR2*.

Exercițiul 9. Modificați valoarea implicită a coloanei *data_angajarii* la data sistemului+ o zi.

Exercițiul 10. Eliminați coloana *varsta* 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.

Redenumirea tabelelor

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

```
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 sau proceduri și funcții stocate.

Exemplul 11. Redenumiți tabelul *functionar_**** cu *funct_****.

```
RENAME functionar_*** TO funct_***;
```

Exercițiul 12. Recreați tabelul *functionar_**** utilizând tabelul *funct_****. (CREATE TABLE AS SELECT)

Exercițiul 13. Eliminați tabelul *funct_****.

Constrângeri

Tipuri de constrângeri:

- **NOT NULL** - coloane ce nu pot conține valoarea *Null*; (NOT NULL)
- **UNIQUE** - coloane sau combinații de coloane care trebuie să aibă valori unice în cadrul tabelului; (UNIQUE (col1, col2, ...))
- **PRIMARY KEY** - identifică în mod unic orice înregistrare din tabel. Echivalent cu NOT NULL + UNIQUE; (PRIMARY KEY (col1, col2, ...))
- **FOREIGN KEY** - stabilește o relație de cheie externă - cheie primară între o coloană a tabelului și o altă coloană dintr-un tabel specificat.
[FOREIGN KEY nume_col]
REFERENCES nume_tabel(nume_coloana)

- [ON DELETE {CASCADE| SET NULL}]
- *FOREIGN KEY* este utilizat 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 linii din tabelul „părinte“, să fie șterse și liniile dependente din tabelul „copil“;
- *ON DELETE SET NULL* determină modificarea automată a valorilor cheii externe la valoarea *null*, atunci când se șterge valoarea „părinte“.
- **CHECK** - o condiție care să fie adevărată la nivel de coloană sau linie (CHECK (conditie)).

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

Adăugarea constrângerilor la crearea tabelului (CREATE TABLE)

```
CREATE TABLE [schema.]nume_tabel (
    nume_coloana          tip_de_date          [DEFAULT          expr]
    [constrangere_de_coloana], ...
    ..[constrangere la nivel de tabel])
```

Exemplul 14. Ștergeți și apoi creați din nou tabelul salariat_*** cu următoarea structură.

| NUME | TIP | CONSTRÂNGERE |
|----------------|--------------|---|
| cod_ang | NUMBER(4) | Cheie primară |
| nume | VARCHAR2(25) | NOT NULL |
| prenume | VARCHAR2(25) | |
| data_nasterii | DATE | data_nasterii<data_angajarii |
| functia | VARCHAR2(9) | NOT NULL |
| sef | NUMBER(4) | Referă ca și cheie externă cod_ang din același tabel |
| data_angajarii | DATE | |
| Email | VARCHAR2(20) | unic |
| salariu | NUMBER(12,3) | > 0 |
| cod_dept | NUMBER(4) | |
| | | Combinatia NUME + PRENUME să fie unică |

Observație:

Constrângerile de tip CHECK se pot implementa la nivel de coloană doar dacă nu referă o altă coloană a tabelului.

```

DROP TABLE salariat_***;
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) CHECK (salariu > 0),
    cod_dep NUMBER(4),
    CONSTRAINT const_c_*** CHECK (data_angajarii > data_nasterii),
    CONSTRAINT const_u_*** UNIQUE (nume,prenume,data_nasterii));

```

Exercițiul 15. 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.

Exercițiul 16. Creați tabelul departament_*** care să aibă următoarea structură.

| NUME | TIP | CONSTRÂNGERI |
|---------|--------------|---------------|
| COD_DEP | NUMBER(4) | Cheie primară |
| NUME | VARCHAR2(20) | Not null |
| ORAS | VARCHAR2(25) | |

Adăugarea constrângerilor ulterior creării tabelului, eliminarea, activarea sau dezactivarea constrângerilor (ALTER TABLE)

- adaugă constrângeri (fără NOT NULL)


```
ALTER TABLE nume_tabel
ADD [CONSTRAINT nume_constr] tip_constr (coloana);
```
- elimină constrângeri (fără NOT NULL)


```
ALTER TABLE nume_tabel
DROP [CONSTRAINT nume_constr] tip_constr (coloana);
```
- adaugă constrângerea NOT NULL


```
ALTER TABLE nume_tabel
MODIFY coloana NOT NULL;
```
- elimină constrângerea NOT NULL


```
ALTER TABLE nume_tabel
MODIFY coloana NULL;
```
- activare/dezactivare constrângere


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

 sau


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

Exercițiul 17. Inserați o nouă înregistrare în salariat_*** de forma:

| cod | nume | prenume | data_n | functia | sef | data_ang | email | salariu | cod_dep |
|-----|------|---------|-------------|-----------|-----|----------|-------|---------|---------|
| 2 | N2 | P2 | 11-JUN-1960 | economist | 1 | Sysdate | E2 | 2000 | 10 |

Ce observați? Introduceți înregistrarea dar specificând valoarea NULL pentru coloana *sef*.

Exemplul 18. Încercați să adăugați o constrângere de cheie externă pe *cod_dep* din *salariat_****. Ce observați?

```
ALTER TABLE salariat_***
ADD CONSTRAINT cce2_*** FOREIGN KEY (cod_dep) REFERENCES
departament_*** (cod_dep);
```

Exercițiul 19. Inserați o nouă înregistrare în *departament_****. Apoi adăugați constrângerea de cheie externă definită anterior.

| cod_dep | nume | loc |
|---------|----------|-----------|
| 10 | Economic | Bucuresti |

Exercițiul 20. Inserați noi înregistrări în *salariat_****, respectiv în *departament_****. Care trebuie să fie ordinea de inserare?

| cod | nume | prenume | data_n | functia | sef | data_ang | email | salariu | cod_dep |
|-----|------|---------|-------------|---------|-----|----------|-------|---------|---------|
| 3 | N3 | P3 | 11-JUN-1967 | jurist | 2 | Sysdate | E3 | 2500 | 20 |

| cod_dep | nume | loc |
|---------|---------|-----------|
| 20 | Juritic | Constanta |

Exercițiul 21. Ștergeți departamentul 20 din tabelul *departament_****. Ce observați?

Exemplul 22. Ștergeți constrângerea *cce2_****. Recreați această constrângere adăugând opțiunea **ON DELETE CASCADE**.

```
ALTER TABLE salariat_***
DROP CONSTRAINT cce2_***;
ALTER TABLE salariat_***
ADD CONSTRAINT cce2_*** FOREIGN KEY (cod_dep) REFERENCES
departament_*** (cod_dep) ON DELETE CASCADE;
```

Exercițiul 23. Ștergeți departamentul 20 din tabelul *departament_****. Ce observați în tabelul *salariat_****? Anulați modificările.

Exercițiul 24. Ștergeți constrângerea *cce2_****. Recreați această constrângere adăugând opțiunea **ON DELETE SET NULL**.

Exemplul 25. Încercați să ștergeți departamentul 10 din tabelul *departament_****. Ce observați?

Consultarea dicționarului datelor

Informații despre tabelele create se găsesc în vizualizările :

- **USER_TABLES** – informații complete despre tabelele utilizatorului curent.
- **ALL_TABLES** – informații complete despre tabelele tuturor utilizatorilor.
- **COLS** – informații despre coloane.
- **TAB** – informații de bază despre tabelele existente în schema utilizatorului curent.

Informații despre constrângeri găsim în :

- **USER_CONSTRAINTS** – informații despre constrângerile definite de utilizatorul curent
- **ALL_CONSTRAINTS** – informații despre constrângerile definite de toți utilizatorii.