

Baze de date - Anul 1

Laborator 3

Limbaajul de control al datelor (LCD)

Tranzacția este o unitate logică de lucru, constituită dintr-o secvență de comenzi care trebuie să se execute atomic (ca un întreg) pentru a menține consistența bazei de date.

Server-ul *Oracle* asigură consistența datelor pe baza tranzacțiilor, inclusiv în eventualitatea unei anomalii a unui proces sau a sistemului. Tranzacțiile oferă mai multă flexibilitate și control în modificarea datelor.

Comenzile SQL care alcătuiesc **LCD** sunt:

- **ROLLBACK** — pentru a renunța la modificările aflate în așteptare se utilizează instrucțiunea *ROLLBACK*. În urma execuției acesteia, se încheie tranzacția, se anulează modificările asupra datelor, se restaurează starea lor precedentă și se eliberează blocările asupra liniilor.
- **COMMIT** - determină încheierea tranzacției curente și permanentizarea modificărilor care au intervenit pe parcursul acesteia. Instrucțiunea suprimă toate punctele intermediare definite în tranzacție și eliberează blocările tranzacției.

Obs: O comandă LDD (*CREATE*, *ALTER*, *DROP*) determină un **COMMIT** implicit.

- **SAVEPOINT** - Instrucțiunea *SAVEPOINT* marchează un punct intermediar în procesarea tranzacției. În acest mod este posibilă împărțirea tranzacției în subtranzacții. Această instrucțiune nu face parte din standardul *ANSI* al limbajului SQL.

Limbaajul de definire a datelor (LDD)

- 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.
- Aceste instrucțiuni 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 maximum 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*.

1. Crearea tabelelor

➤ Formele simplificate ale comenzii de creare a tabelelor sunt:

```
CREATE TABLE nume_tabel (
    coloana_1 tip_date [DEFAULT valoare]
                                [constrangere_nivel_coloana [constrangere_nivel_coloana]...],
    .....
    coloana_n tip_date [DEFAULT valoare]
                                [constrangere_nivel_coloana [constrangere_nivel_coloana]...],
    [constrangeri_nivel_tabel]
);
```

sau

```
CREATE TABLE nume_tabel [(coloana_1,..., coloana_n)]
    AS subcerere;
```

Obs: Nu pot exista două tabele cu același nume în cadrul unei scheme; recrearea unui tabel va fi precedată de suprimarea sa prin comanda:

```
DROP TABLE ANGAJATI_pnu;
```

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

Tip de date	Descriere
VARCHAR2(n) [BYTE CHAR]	Definește un șir de caractere de dimensiune variabilă, având lungimea maximă de <i>n</i> octeți sau caractere. Valoarea maximă a lui <i>n</i> corespunde la 4000 octeți, iar cea minimă este de un octet sau un caracter.
CHAR(n) [BYTE CHAR]	Reprezintă un șir de caractere de lungime fixă având <i>n</i> octeți sau caractere. Valoarea maximă a lui <i>n</i> corespunde la 2000 octeți. Valoarea implicită și minimă este de un octet.
NUMBER(p, s)	Reprezintă un număr având <i>p</i> cifre, dintre care <i>s</i> cifre formează partea zecimală
LONG	Conține șiruri de caractere având lungime variabilă, care nu pot ocupa mai mult de 2GB.
DATE	Reprezintă date calendaristice valide, între 1 ianuarie 4712 i.Hr. și 31 decembrie 9999 d.Hr.

➤ **Constrângerile** definite asupra unui tabel pot fi de următoarele tipuri:

- **NOT NULL** - coloana nu poate conține valoarea *Null*; (**NOT NULL**)
- **UNIQUE** – pentru 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. Implică **NOT NULL** + **UNIQUE**; (**PRIMARY KEY (col1, col2, ...)**)
- **FOREIGN KEY** - stabilește o relație de cheie externă între o coloană a tabelului și o 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 (condiție)*).

Obs:

- Constrângerile pot fi create o dată cu tabelul sau adăugate ulterior cu o comandă *ALTER TABLE*.
- Constrângerile de tip *CHECK* se pot implementa la nivel de coloană doar dacă nu referă o altă coloană a tabelului.
- 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.
- Constrângerea de tip *NOT NULL* se poate declara doar la nivel de coloană

2. Modificarea (structurii) tabelelor

➤ **Modificarea structurii unui tabel** se face cu ajutorul comenzii *ALTER TABLE*. Forma comenzii depinde de tipul modificării aduse:

- 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 din structura tabelului:

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

Obs:

- 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.

➤ Comanda *ALTER* permite **adăugarea unei constrângeri într-un tabel existent, eliminarea, activarea sau dezactivarea constrângerilor**.

- Pentru adăugare de constrângeri, comanda are forma:

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

- Pentru eliminare de constrângeri:

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

- Pentru activare/dezactivare constrângere:

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

3. Suprimarea 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*;

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

4. Redenumirea tabelelor

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

RENAME *nume1_obiect TO* *nume2_obiect*;

Obs:

- 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.

5. Consultarea dicționarului 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_COLUMN*