



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale
2007-2013



Platformă de e-learning și curriculum e-content pentru învățământul superior tehnic

Proiect nr. 154/323 cod SMIS – 4428 cofinanțat de prin Fondul European de Dezvoltare Regională “Investiții pentru viitorul dumneavoastră”.

Programul Operațional Sectorial Creșterea Competitivității Economice - POS CCE



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale
2007-2013



Platformă de e-learning și curriculum e-content pentru învățământul superior tehnic

Baze de date

18. Definirea și manipularea datelor în baze de date

Introducere

În funcție de rolul pe care îl au, comenzile SQL(*Structured Query Language*) pot fi împărțite astfel :

- comenzi de definire a datelor (DDL - *Data Definition Language*);
- comenzi de manipulare a datelor (DML - *Data Manipulation Language*);
- comenzi de control pentru tranzacții;
- comenzi de control pentru sesiuni;
- comenzi de control pentru sistem.

Comenzi DDL

- Comenzile DDL permit efectuarea următoarelor operațiuni:
 - Crearea, modificarea și ștergerea de obiecte într-o schemă;
 - Crearea de useri, roluri și privilegii;
 - Crearea și alocarea de spații logice pentru useri și tablespace-uri;
 - Setarea de parametri pentru sistemul de gestiune;
 - Setarea de parametri pentru dicționarul bazei de date;
 - Auditarea bazei de date;
 - Setarea și Consultarea dicționarului bazei de date.
- Comenzile DDL de bază sunt:
 - CREATE- pentru a crea noi obiecte și structuri în baza de date;
 - ALTER – pentru modificarea de obiecte și structuri existente;
 - DROP – pentru ștergerea din dicționar de obiecte și structuri.

Comenzi CREATE - Oracle

- CREATE database- creează o bază de date;
- CREATE function- creează o funcție stocată în memorie;
- CREATE index -creează un index pentru un tabel sau un grup;
- CREATE package -creează specificația unui pachet stocat în memorie;
- CREATE procedure-creează o procedură stocată în memorie;
- CREATE role-creează un rol(un set de privilegii care pot fi acordate utilizatorilor);
- CREATE synonym-creează un sinonim pentru un obiect din schemă;
- CREATE table-creează un tabel;
- CREATE tablespace – creează un nou tablespace în baza de date;
- CREATE TRIGGER- creează un trigger pe baza de date;
- CREATE user-creează un nou utilizator al bazei de date;

Crearea unei baze de date

- SQL>CREATE DATABASE new_data
CONTROL FILE ora_file
LOGFILE GROUP1 (' disk02.log1.log ') SIZE 50K
MAXLOGFILES 5
MAXLOGHISTORY 100
DATAFILE ' disk02:data1.dat ' SIZE 5M
MAXDATAFILES 10
MAXINSTANCES 2 ;

Crearea unui tabel

- SQL>CREATE TABLE angajati

```
( id_ang    number(4)    PRIMARY KEY,
  nume      varchar2(30),
  functie   varchar2(20) REFERENCES functii(den_functie),
  id_sef    number(4)    REFERENCES angajati(id_ang),
  data_ang  date,
  salariu   number(7,2),
  comision  number(7,2),
  id_dep    number(2) ,
  CONSTRAINT FK_ANG FOREIGN KEY (id_dep)
  REFERENCES departamente(id_dep) );
```

Crearea unui tablespace

- SQL> CREATE TABLESPACE data
DATAFILE ' disk02:data1.dat ' SIZE 2M
DEAFULT STORAGE (INITIAL 10K NEXT 50K
MINEXTENTS 1 MAXEXTENTS 10
PCTINCREASE 10)
ON LINE ;

Crearea unui user

- SQL> CREATE USER student
IDENTIFIED BY stud
DEFAULT TABLESPACE data
TEMPORARY TABLESPACE temp
QUOTA 10M ON data
QUOTA 1M ON temp
QUOTA 5M ON system
PROFILE student ;

Comenzi ALTER - Oracle

- ALTER database- modifică o bază de date;
- ALTER function- recompilează o funcție stocată în memorie;
- ALTER index -modifică un index pentru un tabel sau un grup;
- ALTER package -recompilează un pachet stocat în memorie;
- ALTER procedure-recompilează o procedură stocată în memorie;
- ALTER role-modifică un rol
- ALTER synonym-modifică un sinonim pentru un obiect din schemă;
- ALTER table-modifică un tabel;
- ALTER tablespace – modifică parametrii unui tablespace în baza de date;
- ALTER trigger- recompilează un trigger pe baza de date;
- ALTER user- modifică parametrii unui utilizator al bazei de date;

Exemple:

- Adăugarea unei noi coloane într-un tabel:
 - SQL>ALTER TABLE departamente ADD (interfon number(3) NOT NULL);
- Ștergerea unei coloane dintr-un tabel:
 - SQL>ALTER TABLE departamente DROP COLUMN interfon;
- Modificarea dimensiunii unei coloane :
 - SQL> ALTER TABLE departamente MODIFY (telefon varchar2(12));
- Adăugarea unei noi constrângeri:
 - SQL>ALTER TABLE departamente ADD CONSTRAINT UK_DEN UNIQUE (den_dep);
- Ștergerea unei constrângeri:
 - SQL>ALTER TABLE departamente DROP CONSTRAINT UK_DEN;
- Dezactivarea unei constrângeri:
 - SQL>ALTER TABLE departamente DISABLE CONSTRAINT UK_DEN;
- Marcarea unei coloane ca neutilizabilă:
 - SQL>ALTER TABLE departamente SET UNUSED (interfon);

Comanda DROP

- Comanda DROP șterge obiecte din dicționarul bazei de date. Sintaxa generală este:

- **DROP *cuvant_cheie* *nume_obiect/comanda***

unde ***cuvant_cheie*** poate fi:

- funcție/procedură
- index
- pachet
- rol
- sinonim
- tabel
- tablespace
- trigger
- utilizator
- vedere
- privilegiu

Exemple:

- Ştergerea unui index:
 - SQL>DROP INDEX ind_depart ;
- Ştergerea unui tabel:
 - SQL>DROP TABLE departamente ;
- Ştergerea unei funcţii/proceduri stocate:
 - SQL>DROP FUNCTION/PROCEDURE calcul_salariu ;
- Ştergerea unui pachet:
 - SQL>DROP PACKAGE pac_contab;
- Ştergerea unui user:
 - SQL>DROP USER student ;
- Ştergerea unei constrângeri:
 - SQL>DROP CONSTRAINT PK_depart ;

Comenzile DML (*Data Manipulation Language*)

- Comenzile DML sunt:
- **INSERT** - permite inserarea de linii într-un tabel. ;
- **UPDATE** - permite modificarea uneia sau mai multor linii dintr-un tabel;
- **DELETE** - permite ștergerea uneia sau mai multor linii dintr-un tabel;
- **TRUNCATE** - permite ștergerea tuturor liniilor dintr-un tabel, fără posibilitatea de rollback.

Comanda INSERT

- Această comandă este folosită pentru inserarea de linii într-un tabel și are următoarea sintaxă:
- **INSERT INTO [schema.] table_name [view_name] [@dblink]
[column 1, column 2,]
VALUES (expr1, expr2.)
subquery**

unde :

- **schema** – este schema unde este creat tabelul (specifică userul și baza de date)
- **table_name** – este numele tabelului
- **view_name** – este numele unui view creat pe un tabel
- **column** – este numele coloanei
- **expr** – reprezintă valoarea aferentă coloanei
- **subquery** – este o subcerere care returnează linii cu date

Exemple:

- **SQL> INSERT INTO intrari_gestiune
VALUES(999,'12-DEC-2006',33,2,1,124.50);**
- **SQL>INSERT INTO iesiri_gestiune
(NR_DOC_OUT,DATA_DOC_OUT,COD_PRODUS,
CANT_OUT,COD_UM) VALUES(777,'15-DEC-2006',333,10,2);**
- **SQL>INSERT INTO stocuri_gestiune
SELECT i.nr_doc_in, i.data_doc_in, i.cod_produs,i.cant_in- SUM(e.cant_out), i.cod_um,
i.pret_unitar*(i.cant_in - SUM(e.cant_out)), sysdate
FROM intrari_gestiune i, iesiri_gestiune e
WHERE
i.nr_doc_in=e.nr_doc_in(+) AND
i.data_doc_in=e.data_doc_in (+) AND
i.cod_produs=e.cod_produs (+) AND
i.cod_um=e.cod_um(+)
GROUP BY i.nr_doc_in, i.data_doc_in, i.cod_produs, i.cod_um, i.cant_in, i.pret_unitar;**

Comanda UPDATE

- Este folosita pentru actualizarea datelor in baza de date si are urmatoarea sintaxa:
- **UPDATE [schema.] table_name[view_name][@dblink]**
SET column=expr
column=subquery
(column1,column2,...) IN [NOT IN] subquery
WHERE condition

unde :

- **schema** – este schema unde este creat tabelul (specifica userul si baza de date)
- **table_name** – este numele tabelului
- **column** – este numele coloanei
- **condition** – conditia care trebuie indeplinita pentru modificarea liniilor
- **subquery** – este o subcerere care returneaza linii cu date

Exemple:

- **SQL>UPDATE intrari_gestiune SET pret_unitar=pret_unitar*1.1;**
- **SQL>UPDATE intrari_gestiune SET pret_unitar=pret_unitar*1.1
WHERE cod_produș IN (11,12);**
- **SQL>UPDATE stocuri_gestiune s
SET
valoare=stoc*(SELECT pret_unitar
FROM intrari_gestiune i
WHERE i.cod_produș IN (11,12) AND
i.cod_um = s.cod_um AND
i.nr_doc_in = s.nr_doc_in AND
i.cod_produș = s.cod_produș AND
i.data_doc_in = s.data_doc_in),
data_stoc=sysdate
WHERE cod_produș IN (11,12);**

Comanda DELETE

- Este folosită pentru ștergerea liniilor dintr-un tabel sau view și are următoarea sintaxă:
- **DELETE FROM** [schema.]table_name[view_name][@dblink]
WHERE condition
(column1,column2,..) IN [NOT IN] subquery

unde :

- **schema** – este schema unde este creat tabelul (specifică userul și baza de date)
- **table_name** – este numele tabelului
- **condition** – condiția care trebuie îndeplinită pentru liniile șterse
- **column** – este numele coloanei
- **subquery** – este o subcerere care returnează linii cu date

Exemple:

- SQL>DELETE FROM stocuri_gestiune;
- SQL>DELETE FROM stocuri_gestiune
WHERE data_stoc LIKE '%NOV%';
- SQL>DELETE FROM stocuri_gestiune
WHERE
(nr_doc_in,data_doc_in) NOT IN
(SELECT DISTINCT nr_doc_in,data_doc_in
FROM intrari_gestiune);

Comanda TRUNCATE

- Este folosită pentru ștergerea tuturor liniilor dintr-un tabel, fără posibilitatea de rollback și are următoarea sintaxă:

```
TRUNCATE TABLE [schema.]table_name  
DROP/REUSE STORAGE
```

unde :

- **schema** – este schema unde este creat tabelul (specifică userul și baza de date)
- **table_name** – este numele tabelului
- **DROP STORAGE** – eliberează spațiul rezultat din ștergerea liniilor
- **REUSE STORAGE** – păstrează spațiul alocat, rezultat din ștergerea liniilor

Exemplu:

- SQL>TRUNCATE TABLE vanzari_zilnice;