
Limbajul de manipulare a datelor (LMD)

- Comenzile SQL care alcătuiesc **LMD** permit:
  - regăsirea datelor (*SELECT*);
  - adăugarea de noi înregistrări (*INSERT*);
  - modificarea valorilor coloanelor din înregistrările existente (*UPDATE*);
  - adăugarea sau modificarea condiţionată de înregistrări (*MERGE*);
  - suprimarea de înregistrări (*DELETE*).

I. Comanda *INSERT*

1. Inserări mono-tabel

Comanda INSERT are următoarea sintaxă simplificată:

INSERT INTO *obiect* [***AS*** *alias*] [(*nume_coloană* [, *nume_coloană* ...])]
{***VALUES*** ({*expr* | ***DEFAULT***} [, {*expr* | ***DEFAULT***} ...])
| *subcerere*}

Subcererea specificată în comanda *INSERT* returnează linii care vor fi adăugate în tabel.

Dacă în tabel se introduc linii prin intermediul unei subcereri, coloanele din lista *SELECT* trebuie să corespundă, ca număr şi tip, celor precizate în clauza *INTO*. În absenţa unei liste de coloane în clauza *INTO*, subcererea trebuie să furnizeze valori pentru fiecare atribut al obiectului destinaţie, respectând ordinea în care acestea au fost definite.

Observaţii (tipuri de date):

- Pentru claritate, este recomandată utilizarea unei liste de coloane în clauza *INSERT*.
- În clauza *VALUES*, valorile de tip caracter şi dată calendaristică trebuie incluse între apostrofuri. Nu se recomandă includerea între apostrofuri a valorilor numerice, întrucât aceasta ar determina conversii implicite la tipul *NUMBER*.
- Pentru introducerea de valori speciale în tabel, pot fi utilizate funcţii.

Adăugarea unei linii care va conţine valori *null* se poate realiza în mod:

- implicit, prin omiterea numelui coloanei din lista de coloane;
- explicit, prin specificarea în lista de valori a cuvântului cheie *null*

În cazul şirurilor de caractere sau al datelor calendaristice se poate preciza şirul vid ('').

Observaţii (erori):

Server-ul *Oracle* aplică automat toate tipurile de date, domeniile de valori şi constrângerile de integritate. La introducerea sau actualizarea de înregistrări, pot apărea erori în următoarele situaţii:

- nu a fost specificată o valoare pentru o coloană *NOT NULL*;
- există valori duplicat care încalcă o constrângere de unicitate;
- a fost încălcată constrângerea de cheie externă sau o constrângere de tip *CHECK*;
- există o incompatibilitate în privinţa tipurilor de date;
- s-a încercat inserarea unei valori având o dimensiune mai mare decât a coloanei corespunzătoare.

2. Inserari multi-tabel

O inserare multi-tabel presupune introducerea de linii calculate pe baza rezultatelor unei subcereri, într-unul sau mai multe tabele. Acest tip de inserare, introdus de *Oracle9i*, este util în mediul *data warehouse*.

Pentru o astfel de inserare, în versiunile anterioare lui *Oracle9i* erau necesare *n* operaţii independente *INSERT INTO…SELECT…*, unde *n* reprezintă numărul tabelelor destinaţie. Aceasta presupunea *n* procesări ale aceleiaşi surse de date şi, prin urmare, creşterea de *n* ori a timpului necesar procesului.

Sintaxa comenzii *INSERT* în acest caz poate fi:

- Pentru inserări necondiţionate:

INSERT ALL INTO… [***INTO***…]
subcerere;

- Pentru inserări condiţionate:

INSERT [***ALL*** | ***FIRST***]
WHEN *condiţie* ***THEN INTO***…
[***WHEN*** *condiţie* ***THEN INTO***…
[***ELSE INTO*** …]]
subcerere;

- *ALL* determină evaluarea tuturor condiţiilor din clauzele *WHEN*. Pentru cele a căror valoare este *TRUE*, se inserează înregistrarea specificată în opţiunea *INTO* corespunzătoare.

- *FIRST* determină inserarea corespunzătoare primei clauze *WHEN* a cărei condiţie este evaluată *TRUE*. Toate celelalte clauze *WHEN* sunt ignorate.

II. Comanda UPDATE

Sintaxa simplificată a comenzii ***UPDATE*** este:

UPDATE nume_tabel [alias] SET
col1 = expr1[, col2=expr2]
[WHERE conditie];

sau

UPDATE nume_tabel [alias] SET
(col1,col2,...) = (subcerere)
[WHERE conditie];

Observaţii:

- de obicei pentru identificarea unei linii se foloseşte o condiţie ce implică cheia primară;
- dacă nu apare clauza *WHERE* atunci sunt afectate toate liniile tabelului specificat;
- cazurile în care instrucţiunea *UPDATE* nu poate fi executată sunt similare celor în care eşuează instrucţiunea *INSERT*. Acestea au fost menţionate anterior.

III. Comanda DELETE

Sintaxa simplificată a comenzii ***DELETE*** este:

DELETE FROM nume_tabel
[WHERE conditie];

Daca nu se specifica nici o conditie, vor fi şterse toate liniile din tabel.

Exerciţii [I]

1. Să se insereze departamentul 300, cu numele *Programare* în *DEPT_pnu (copie a tabelului Departments)*.
   Analizaţi cazurile, precizând care este soluţia corectă şi explicând erorile celorlalte variante. Pentru a anula efectul instrucţiunii(ilor) corecte, utilizaţi comanda *ROLLBACK*.
   a) *INSERT INTO DEPT_pnu*
      VALUES (300, 'Programare');
   b) *INSERT INTO DEPT_pnu (department_id, department_name)*
      VALUES (300, 'Programare');
   c) *INSERT INTO DEPT_pnu (department_name, department_id)*
      VALUES (300, 'Programare');
   d) *INSERT INTO DEPT_pnu (department_id, department_name, location_id)*
      VALUES (300, 'Programare', null);
   e) *INSERT INTO DEPT_pnu (department_name, location_id)*
      VALUES ('Programare', null);
2. Încercaţi dacă este posibilă introducerea unui angajat, precizând pentru valoarea *employee_id* o subcerere care returnează (codul maxim +1).
3. Creaţi un nou tabel, numit *EMP1_PNU*, care va avea aceeaşi structură ca şi *EMPLOYEES*, dar nicio înregistrare. Copiaţi în tabelul *EMP1_PNU* salariaţii (din tabelul *EMPLOYEES*) al căror salariu este cel puțin 10000.
4. Să se creeze tabelele necesare cu aceeaşi structură ca a tabelului *EMPLOYEES* (fără constrângeri şi fără înregistrări). Copiaţi din tabelul *EMPLOYEES*:
   - în tabelul *EMP0_PNU* salariaţii care lucrează în departamentul 80;
   - în tabelul *EMP1_PNU* salariaţii care au salariul mai mic decât 5000;
   - în tabelul *EMP2_PNU* salariaţii care au salariul cuprins între 5000 şi 10000;
   - în tabelul *EMP3_PNU* salariaţii care au salariul mai mare decât 10000.

   Dacă un salariat se încadrează în tabelul emp0_pnu atunci acesta nu va mai fi inserat şi în alt tabel (tabelul corespunzător salariului său).

Exerciţii [II]

5. Măriţi salariul tuturor angajaţilor din tabelul *EMP_PNU* cu 5%. Vizualizati, iar apoi anulaţi modificările.
6. Să se promoveze Douglas Grant la manager în departamentul 20, având o creştere de salariu cu 1000$. Se poate realiza modificarea prin intermediul unei singure comenzi?
7. Să se modifice jobul şi departamentul angajatului având codul 114, astfel încât să fie la fel cu cele ale angajatului având codul 205.

Exercitii [III]

8. Ştergeţi toate înregistrările din tabelul *DEPT_PNU*. Ce înregistrări se pot şterge? Anulaţi modificările.
9. Ştergeţi angajaţii care nu au comision. Anulaţi modificările.

Definirea secvenţelor

Secvenţa este un obiect al bazei de date ce permite generarea de întregi unici pentru a fi folosiţi ca valori pentru cheia primară sau coloane numerice unice. Secvenţele sunt independente de tabele, aşa că aceeaşi secvenţă poate fi folosită pentru mai multe tabele.

Crearea secvenţelor se realizează prin comanda *CREATE SEQUENCE*, a cărei sintaxă este:

CREATE SEQUENCE *nume_secv*
[INCREMENT BY *n] [**START WITH** n] [{**MAXVALUE** n | **NOMAXVALUE**}] [{**MINVALUE** n | **NOMINVALUE**}] [{**CYCLE** | **NOCYCLE**}]*
*[{**CACHE** n | **NOCACHE**}]*

La definirea unei secvenţe se pot specifica:

- numele secvenţei
- diferenţa dintre 2 numere generate succesiv, implicit fiind 1 (*INCREMENT BY*);
- numărul initial, implicit fiind 1 (*START WITH*);
- valoarea maximă, implicit fiind 1027 pentru o secvenţă ascendentă şi –1 pentru una descendentă;
- valoarea minimă, implicit fiind 1 pentru o secvenţă ascendentă şi -1027 pentru o secvenţă descendentă;
- dacă secvenţa ciclează după ce atinge limita; (*CYCLE*)
- câte numere să încarce în *cache server*, implicit fiind încărcate 20 de numere (*CACHE*).

Informaţii despre secvenţe găsim în dicţionarul datelor. Pentru secvenţele utilizatorului curent, interogăm *USER_SEQUENCES*. Alte vizualizări utile sunt *ALL_SEQUENCES* şi *DBA_SEQUENCES*.

Pseudocoloanele *NEXTVAL* şi *CURRVAL* permit lucrul efectiv cu secvenţele.

- *Nume_secv.NEXTVAL* - returnează următoarea valoare a secvenţei, o valoare unică la fiecare referire. Trebuie aplicată cel puţin o dată înainte de a folosi *CURRVAL*;
- *Nume_secv.CURRVAL* – obţine valoarea curentă a secvenţei.

Obs: Pseudocoloanele se pot utiliza în:

- lista *SELECT* a comenzilor ce nu fac parte din subcereri;
- lista *SELECT* a unei cereri ce apare într un *INSERT*;
- clauza *VALUES* a comenzii *INSERT*;
- clauza *SET* a comenzii *UPDATE*.

Obs: Pseudocoloanele nu se pot utiliza:

- în lista *SELECT* a unei vizualizări;
- într-o comanda *SELECT* ce conţine *DISTINCT*, *GROUP BY*, *HAVING* sau *ORDER BY*;
- într-o subcerere în comenzile *SELECT*, *UPDATE*, *DELETE*
- în clauza *DEFAULT* a comenzilor *CREATE TABLE* sau *ALTER TABLE*.

Ştergerea secvenţelor se face cu ajutorul comenzii *DROP SEQUENCE*.

DROP SEQUENCE *nume_secventa;*

Exerciţii [II]

1. Creaţi o secvenţă pentru generarea codurilor de departamente, *SEQ_DEPT_PNU*. Secvenţa va începe de la 400, va creşte cu 10 de fiecare dată şi va avea valoarea maximă 10000, nu va cicla şi nu va încărca nici un număr înainte de cerere.
2. Creaţi o secvenţă pentru generarea codurilor de angajaţi, *SEQ_EMP_PNU*. Să se modifice toate liniile din *EMP_PNU* (dacă nu mai există, îl recreeaţi), regenerând codul angajaţilor astfel încât să utilizeze secvenţa *SEQ_EMP_PNU* şi să avem continuitate în codurile angajaţilor.
3. Ștergeți secvența *SEQ_DEPT_PNU*.