

Universitatea Constantin Brâncuși din Târgu Jiu

Facultatea: Inginerie

Program de conversie profesională a cadrelor didactice din învățământul preuniversitar: Informatică, Tehnologia Informației și a Comunicațiilor

Laborator 8

Limbajul de manipulare a datelor (LMD)

Limbajul de control al datelor (LCD)

A. Comenzile SQL care alcătuiesc LMD permit:

1. regăsirea datelor (*SELECT*)
2. adăugarea de noi înregistrări (*INSERT*)
3. modificarea valorilor coloanelor din înregistrările existente (*UPDATE*)
4. adăugarea sau modificarea condiționată de înregistrări (*MERGE*)
5. stergerea de înregistrări (*DELETE*)

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.

B. Comenzile SQL care alcătuiesc LCD sunt:

a. **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.

b. **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.

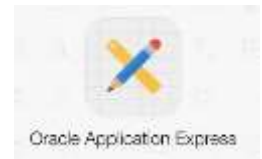
c. **SAVEPOINT** - Instrucțiunea *SAVEPOINT* marchează un punct intermediar în procesarea tranzacției. În acest mod este posibilă împărțirea tranzacției în subtranzacții. Această instrucțiune nu face parte din standardul *ANSI* al limbajului *SQL*.

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 și tip, celor precizate în clauza *INTO*. În absența unei liste de coloane în clauza *INTO*, subcererea trebuie să furnizeze valori pentru fiecare atribut al obiectului destinație, respectând ordinea în care acestea au fost definite.

Observații (tipuri de date):

- Pentru claritate, este recomandată utilizarea unei liste de coloane în clauza *INSERT*.
- În clauza *VALUES*, valorile de tip caracter și 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 funcții.

Adăugarea unei linii care va conține 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 șirurilor de caractere sau al datelor calendaristice se poate preciza șirul vid (").

Observații (erori):

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

- nu a fost specificată o valoare pentru o coloană *NOT NULL*;
- există valori duplicate care încalcă o constrângere de unicitate;
- a fost încălcată constrângerea de cheie externă sau o constrângere de tip *CHECK*;
- există o incompatibilitate în privința 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 operații independente *INSERT INTO...SELECT...*, unde n reprezintă numărul tabelor destinație. Aceasta presupunea n procesări ale aceleiași surse de date și, prin urmare, creșterea de n ori a timpului necesar procesului.

Sintaxa comenzii *INSERT* în acest caz poate fi:

- Pentru inserări necondiționate:

```
INSERT ALL INTO... [INTO ...]  
subcerere;
```

- Pentru inserări condiționate:

```
INSERT [ALL | FIRST]  
WHEN condiție THEN INTO...  
[WHEN condiție THEN INTO...  
[ELSE INTO ...]]  
subcerere;
```

Baze de date - SQL (2022)



- *ALL* determină evaluarea tuturor condițiilor din clauzele *WHEN*. Pentru cele a căror valoare este *TRUE*, se inserează înregistrarea specificată în opțiunea *INTO* corespunzătoare.
- *FIRST* determină inserarea corespunzătoare primei clauze *WHEN* a cărei condiție este evaluată *TRUE*. Toate celelalte clauze *WHEN* sunt ignorate.

Exercitii rezolvate:

1. Să se creeze tabelele *Angajati_pnu*, *Departamente_pnu* (în șirul de caractere “pnu”, *p* reprezintă prima literă a prenumelui, iar *nu* reprezintă primele două litere ale numelui dumneavoastră), prin copierea structurii și conținutului tabelor *Angajati*, respectiv *Departamente*.

Solutie:

```
CREATE TABLE Angajati_pnu AS SELECT * FROM EMP;
```

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below the navigation bar, there is a 'SQL Commands' section with a 'Language' dropdown set to 'SQL' and 'Rows' set to '10'. The main text area contains the SQL command: `1 CREATE TABLE Angajati_pnu AS SELECT * FROM EMP;`. Below the text area, there is a 'Results' section with tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, showing the message 'Table created.' and the execution time '0.11 seconds'.

```
CREATE TABLE Departamente_pnu AS SELECT * FROM DEPT;
```

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below the navigation bar, there is a 'SQL Commands' section with a 'Language' dropdown set to 'SQL' and 'Rows' set to '10'. The main text area contains the SQL command: `1 CREATE TABLE Departamente_pnu AS SELECT * FROM DEPT;`. Below the text area, there is a 'Results' section with tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, showing the message 'Table created.' and the execution time '0.09 seconds'.



2. Să se insereze departamentul 99, cu numele *Programare* în *Departamente_pnu*. Analizați cazurile, precizând care este soluția corectă și explicând erorile celorlalte variante.

Pentru a anula efectul instrucțiunii(ilor) corecte, utilizați comanda *ROLLBACK*.

Soluție:

```
INSERT INTO Departamente_pnu VALUES (99, 'Programare');
```

```
1 INSERT INTO Departamente_pnu VALUES (99, 'Programare');
```

Results Explain Describe Saved SQL History

```
ORA-00947: not enough values
ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_210200", line 673
ORA-06512: at "SYS.DBMS_SQL", line 1658
ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_210200", line 659
ORA-06512: at "APEX_210200.WWV_FLOW_DYNAMIC_EXEC", line 1829
```

```
INSERT INTO Departamente_pnu (deptno, dname) VALUES (99, 'Programare');
```

```
1 INSERT INTO Departamente_pnu (deptno, dname) VALUES (99, 'Programare');
```

Results Explain Describe Saved SQL History

```
1 row(s) inserted.
```

```
INSERT INTO Departamente_pnu (dname, deptno) VALUES (99, 'Programare');
```

```
1 INSERT INTO Departamente_pnu (dname, deptno) VALUES (99, 'Programare');
```

Results Explain Describe Saved SQL History

```
ORA-01722: invalid number
ORA-06512: at "SYS.DBMS_SQL", line 1721
1. INSERT INTO Departamente_pnu (dname, deptno) VALUES (99, 'Programare');
```

```
INSERT INTO Departamente_pnu (deptno, dname, loc) VALUES (99, 'Programare', null);
```

```
1 INSERT INTO Departamente_pnu (deptno, dname, loc) VALUES (99, 'Programare', null);
```

Results Explain Describe Saved SQL History

```
1 row(s) inserted.
```

```
INSERT INTO Departamente_pnu (denumire, id_locatie) VALUES ('Programare', null);
```



```
1 INSERT INTO Departamente_pnu (dname, loc) VALUES ('Programare', null);
2
3
```

Results Explain Describe Saved SQL History

```
ORA-01400: cannot insert NULL into ("UCRM"."DEPARTAMENTE_PNU"."DEPTNO")
ORA-06511: at "SYS.DBMS_SQL", line 1721

1. INSERT INTO Departamente_pnu (dname, loc) VALUES ('Programare', null);
```

Executați varianta care a fost corectă de două ori. Ce se obține și de ce?

3. Este posibilă introducerea de înregistrări prin intermediul subcererilor (specificate în locul tabelului). Ce reprezintă, de fapt, aceste subcereri?

Să se analizeze următoarele comenzi *INSERT*:

```
INSERT INTO Angajati_pnu (empno, ename, hiredate, job, sal, comm, deptno)
VALUES (252, 'Nume252', SYSDATE, 'SA_REP', 5000, NULL, 20);
```

```
SELECT empno, ename, hiredate, job, sal, comm, deptno
FROM Angajati_pnu
WHERE empno = 252;
```

ROLLBACK;

SQL Commands

Language SQL Rows 20 Clear Command Find Tables

```
1 INSERT INTO Angajati_pnu (empno, ename, hiredate, job, sal, comm, deptno)
2 VALUES (252, 'Nume252', SYSDATE, 'SA_REP', 5000, NULL, 20);
3
4 SELECT empno, ename, hiredate, job, sal, comm, deptno
5 FROM Angajati_pnu
6 WHERE empno = 252;
7
8 ROLLBACK;
```

```
INSERT INTO
(SELECT id_angajat, nume, email, data_ang, id_functie, salariu,
comision
FROM Angajati_pnu)
VALUES (252, 'Nume252', 'nume252@emp.com', SYSDATE, 'SA_REP', 5000, NULL);
```

Baze de date - SQL (2022)



```
SELECT id_angajat, nume, email, data_ang, id_funcctie, salariu, comision
FROM Angajati_pnu
WHERE id_angajat = 252;
```

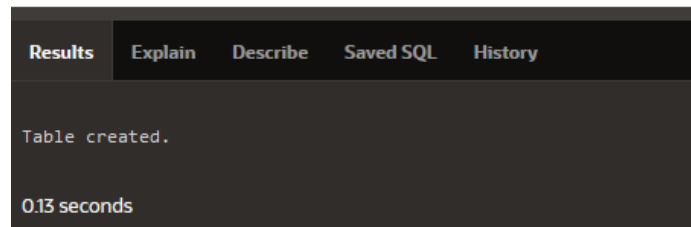
ROLLBACK;

```
1  INSERT INTO
2  | (SELECT empno, ename, hiredate, job, sal, comm, deptno
3  |   FROM Angajati_pnu)
4  VALUES (252, 'Nume252', SYSDATE, 'SA_REP', 5000, NULL, 20);
5
6  SELECT empno, ename, hiredate, job, sal, comm, deptno
7  FROM Angajati_pnu
8  WHERE empno = 252;
9
10 ROLLBACK;
```

4. Creați un nou tabel, numit *ANGAJATI_PNU1*, care va avea aceeași structură ca și *ANGAJATI*, dar nici o înregistrare. Copiați în tabelul *ANGAJATI_PNU1* salariații (din tabelul *ANGAJATI*) al căror comision depășește 25% din salariu.

CREATE TABLE Angajati_pnu1 AS SELECT * FROM EMP;

```
1  CREATE TABLE Angajati_pnu1 AS SELECT * FROM EMP;
2
3
```



```
DELETE FROM Angajati_pnu1;
```



```
1 DELETE FROM Angajati_pnu1;
2
3
```

Results	Explain	Describe	Saved SQL	History
15 row(s) deleted.				
0.01 seconds				

INSERT INTO Angajati_pnu1
SELECT *
FROM Angajati
WHERE comm > 0.25;

```
1 INSERT INTO Angajati_pnu1
2 SELECT *
3 FROM Angajati
4 WHERE comm > 0.25;
5
6
7
```

Results	Explain	Describe	Saved SQL	History
ORA-00942: table or view does not exist ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_210200", line 673 ORA-06512: at "SYS.DBMS_SYS_SQL", line 1658 ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_210200", line 659 ORA-06512: at "APEX_210200.WWV_FLOW_DYNAMIC_EXEC", line 1829 4. WHERE comm > 0.25;				

SELECT empno, ename, sal, comm
FROM Angajati_pnu;



```
1 SELECT empno, ename, sal, comm
2 FROM Angajati_pnu;
3
```

EMPNO	ENAME	SAL	COMM
7839	KING	5000	-
7698	BLAKE	2850	-
7782	CLARK	2450	-
7566	JONES	2975	-
7788	SCOTT	3000	-
7902	FORD	3000	-
7369	SMITH	800	-
7499	ALLEN	1600	300
7521	WARD	1250	500
7654	MARTIN	1250	1400

ROLLBACK;

SQL Commands

Language: SQL Rows: 20

1 ROLLBACK;

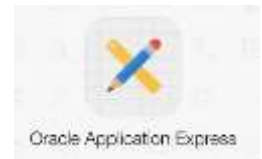
2

3

4

Results Explain Describe Saved SQL History

Rollback statement not applicable. All statements are automatically committed.



Ce va contine tabelul ANGAJATI_PNU1 in urma acestei succesiuni de comenzi?

A screenshot of the SQL Developer interface. The top bar shows 'SQL Commands'. Below it, the 'Language' is set to 'SQL' and 'Rows' is set to '20'. There are buttons for 'Clear Command' and 'Find Tables'. The main area shows a SQL query:

```
1 SELECT empno, ename, sal, comm
2 FROM Angajati_pnu1;
3
4
5
```

The bottom bar shows tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active and displays 'no data found'.

5. Inserați o nouă înregistrare în tabelul *ANGAJATI_PNU* care să totalizeze salariile, să facă media comisioanelor, iar câmpurile de tip dată să conțină data curentă și câmpurile de tip caracter să conțină textul 'TOTAL'. Numele și prenumele angajatului să corespundă utilizatorului curent (*USER*). Pentru câmpul *id_angajat* se va introduce valoarea 0, iar pentru *id_manager* și *id_dept* se va da valoarea null.

```
INSERT INTO Angajati_pnu
  SELECT 0, USER, 'TOTAL', 0, SYSDATE, SUM(sal), ROUND(AVG(comm), 0), null
  FROM EMP;
```

```
select * from Angajati_pnu;
```



```

1 select * from Angajati_pnu;
2

```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	11/17/1981	5000	-	10
7698	BLAKE	MANAGER	7839	05/01/1981	2850	-	30
7782	CLARK	MANAGER	7839	06/09/1981	2450	-	10
7566	JONES	MANAGER	7839	04/02/1981	2975	-	20
7788	SCOTT	ANALYST	7566	12/09/1982	3000	-	20
7902	FORD	ANALYST	7566	12/03/1981	3000	-	20
7369	SMITH	CLERK	7902	12/17/1980	800	-	20
7499	ALLEN	SALESMAN	7698	02/20/1981	1600	300	30
7521	WARD	SALESMAN	7698	02/22/1981	1250	500	30
7654	MARTIN	SALESMAN	7698	09/28/1981	1250	1400	30
7844	TURNER	SALESMAN	7698	09/08/1981	1500	0	30
7876	ADAMS	CLERK	7788	01/12/1983	1100	-	20
7900	JAMES	CLERK	7698	12/03/1981	950	-	30
7934	MILLER	CLERK	7782	01/23/1982	1300	-	10
9000	IONFSCU	SALESMAN	7566	02/17/2022	1000	-	20
0	APEX_PUBLIC_USER	TOTAL	0	02/23/2022	30025	550	-

6. Măriți salariul tuturor angajaților din tabelul *ANGAJATI_PNU* cu 5%. Vizualizati, iar apoi anulați modificările.

```

UPDATE Angajati_pnu
SET sal = sal * 1.05;

```

```

1 UPDATE Angajati_pnu
2 SET sal = sal * 1.05;
3
4
5

```

Results	Explain	Describe	Saved SQL	History
16 row(s) updated.				
0.00 seconds				

```

SELECT * FROM Angajati_pnu;

```



```

1 select * from Angajati_pnu;
2
    
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	11/17/1981	5250	-	10
7698	BLAKE	MANAGER	7839	05/01/1981	2992.5	-	30
7782	CLARK	MANAGER	7839	06/09/1981	2572.5	-	10
7566	JONES	MANAGER	7839	04/02/1981	3123.75	-	20
7788	SCOTT	ANALYST	7566	12/09/1982	3150	-	20
7902	FORD	ANALYST	7566	12/03/1981	3150	-	20
7369	SMITH	CLERK	7902	12/17/1980	840	-	20
7499	ALLEN	SALESMAN	7698	02/20/1981	1680	300	30
7521	WARD	SALESMAN	7698	02/22/1981	1312.5	500	30
7654	MARTIN	SALESMAN	7698	09/28/1981	1312.5	1400	30
7844	TURNER	SALESMAN	7698	09/08/1981	1575	0	30
7876	ADAMS	CLERK	7788	01/12/1983	1155	-	20
7900	JAMES	CLERK	7698	12/03/1981	997.5	-	30
7934	MILLER	CLERK	7782	01/23/1982	1365	-	10
9999	IONESCU	SALESMAN	7566	02/17/2022	1050	-	20
0	APEX_PUBLIC_USER	TOTAL	0	02/23/2022	31526.25	550	-

16 rows returned in 0.00 seconds [Download](#)

ROLLBACK;

SQL Commands

Language SQL Rows 20

↶ ↷ 🔍 🔧 A::

```

1 ROLLBACK;
2
3
4
    
```

Results Explain Describe Saved SQL History

Rollback statement not applicable. All statements are automatically committed.



Probleme propuse spre rezolvare:

1. Schimbați jobul tuturor salariaților din departamentul 80 care au comision în 'SA_REP'. Anulați modificările.

Să se promoveze Ceausescu V. la manager în departamentul 20, având o creștere de salariu cu 1000\$.

Schimbați salariul și comisionul celui mai prost plătit salariat din firmă, astfel încât să fie egale cu salariul și comisionul directorului.

2. Ștergeți toate înregistrările din tabelul DEPT_PNU. Anulați modificările.

```
DELETE FROM dept_pnu;  
SELECT * FROM dept_pnu;  
ROLLBACK;
```

3. Ștergeți angajații care nu au comision. Anulați modificările.

Suprimați departamentele care nu au nici un angajat. Anulați modificările.

Eliminați angajații care nu aparțin unui departament valid. Anulați modificările.

4. Să se ștergă un angajat din tabelul ANGAJATI_PNU, al cărui cod este introdus de utilizator (se va utiliza o variabilă de substituție în comanda DELETE). Modificările să devină permanente.

Să se mai introducă o linie în tabel, rulând încă o dată fișierul creat la exercițiul 10.

Să se marcheze un punct intermediar în procesarea tranzacției.

```
SAVEPOINT p
```

Să se ștergă tot conținutul tabelului. Listați conținutul tabelului.

Să se renunțe la cea mai recentă operație de ștergere, fără a renunța la operația precedentă de introducere.

```
ROLLBACK TO p
```

Listați conținutul tabelului. Determinați ca modificările să devină permanente.