

--LABORATOR 4

1. Să se creeze tabelele EMP_pnu, DEPT_pnu (în care şirul de caractere "pnu", p reprezintă prima literă a prenumelui, iar nu reprezintă primele două litere ale numelui), prin copierea structurii şi conţinutului tabelor EMPLOYEES, respectiv DEPARTMENTS.

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

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

```
CREATE, ALTER, DROP -(comenzi LDD) - executa un commit automat(implicit)
```

```
COMMIT; -- salvarea tuturor modificarilor din tranzactia curenta
```

```
ROLLBACK; -- anularea tuturor modificarilor realizate in sesiunea/tranzactia curenta
```

```
-- de la conectarea pe server sau de la ultimul commit
```

```
SELECT * FROM emp;
```

```
SELECT * FROM dept;
```

```
ROLLBACK; -- in acest caz instructiunea rollback nu are efect deoarece CREATE executa un commit implicit
```

2. Listați structura tabelelor sursă și a celor create anterior. Ce se observă?

3. Listați conținutul tabelelor create anterior.

-- selectam continutul

SELECT * FROM emp;

SELECT * FROM dept; -- observam ca exista acelasi continut ca in tabelul de baza

-- (tabelul de unde am preluat datele -> employees)

-- listam structura tabelelor

desc emp;

desc dept; -- se pastreaza coloanele si tipurile de date

-- dar nu se copiaza si constrangerile de integritate (chei primare/ externe)

desc employees;

/*

In concluzie, utilizand instructiunea:

CREATE TABLE EMP AS SELECT * FROM employees;

se copiaza continutul (toate datele/ liniile din tabel)

alaturi de structura tabelelor (coloane, tipuri de date)

dar nu se copiaza si constrangerile de integritate

*/

4. Pentru introducerea constrângerilor de integritate,

executați instrucțiunile LDD indicate în continuare;

ALTER TABLE emp

ADD CONSTRAINT pk_emp PRIMARY KEY(employee_id);

```
ALTER TABLE dept
ADD CONSTRAINT pk_dept PRIMARY KEY(department_id);
```

```
ALTER TABLE emp
ADD CONSTRAINT fk_emp_dept
FOREIGN KEY(department_id) REFERENCES dept(department_id);
```

Ce constrângere nu am implementat?

```
ALTER TABLE dept
ADD CONSTRAINT fk_dept_emp
FOREIGN KEY(manager_id) references emp(employee_id); -- managerul de departament
```

```
ALTER TABLE emp
ADD CONSTRAINT fk_emp_emp
FOREIGN KEY(manager_id) references emp(employee_id); -- managerul unui angajat
```

5. Să se insereze departamentul 300, cu numele Programare în DEPT.

Analizați cazurile, precizând care este soluția corectă și explicând erorile celorlalte variante.

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

```
DESC dept;
```

--a) este un INSERT implicit - nu sunt specificate coloanele in clauza INSERT

```
INSERT INTO DEPT
```

```
VALUES (300, 'Programare'); -- EROARE! -> not enough values
```

--b) este un INSERT explicit -- trebuie specificate coloanele NOT NULL

-- coloanele care nu sunt specificate in INSERT

-- o sa fie completate automat cu valoarea NULL

```
INSERT INTO DEPT (department_id, department_name)
```

```
VALUES (300, 'Programare'); -- CORECTA
```

```
SELECT * FROM dept;
```

--c)

```
INSERT INTO DEPT (department_name, department_id)
```

```
VALUES (300, 'Programare'); -- EROARE! -> ordinea coloanelor din INSERT, respectiv din VALUES
```

--d)

```
INSERT INTO DEPT (department_id, department_name, location_id)
```

```
VALUES (300, 'Programare', null); --EROARE -> unique constraint (SYSTEM.PK_DEPT) violated
```

-- modificam valoarea cheii primare

```
INSERT INTO DEPT (department_id, department_name, location_id)
```

```
VALUES (301, 'Programare', null);
```

```
SELECT * FROM dept;
```

--e)

```
INSERT INTO DEPT (department_name, location_id)
```

```
VALUES ('Programare', null); -- EROARE! -> cannot insert NULL into  
("SYSTEM"."DEPT"."DEPARTMENT_ID")
```

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

-- Ce se întâmplă dacă executăm un rollback?

ROLLBACK;

SELECT * FROM dept;

-- VARIANTELE CORECTE

INSERT INTO DEPT (department_id, department_name)

VALUES (300, 'Programare');

COMMIT;

INSERT INTO DEPT (department_id, department_name, location_id)

VALUES (301, 'Programare', null);

ROLLBACK;

SELECT * FROM dept;

INSERT INTO DEPT (department_id, department_name, location_id)

VALUES (301, 'Programare', null);

COMMIT;

SELECT * FROM dept;

6. Să se insereze un angajat corespunzător departamentului introdus anterior în tabelul EMP_pnu, precizând valoarea NULL pentru coloanele a căror valoare nu este cunoscută la inserare (metoda implicită de inserare).

Determinați ca efectele instrucțiunii să devină permanente.

Atenție la constrângerile NOT NULL asupra coloanelor tabelului;

-- dorim sa inseram un angajat in depart 300

DESC emp;

select * from emp;

-- metoda implicită de inserare

INSERT INTO emp

VALUES (250, NULL, 'nume250', 'email250', NULL, SYSDATE, 'IT_PROG', NULL, NULL, NULL, 300);

-- EROARE!!!

INSERT INTO emp

VALUES (260, NULL, 'nume250', 'email250', NULL, SYSDATE, 'IT_PROG', NULL, NULL, NULL, 3000);

-- integrity constraint (SYSTEM.FK_EMP_DEPT) violated - parent key not found

COMMIT;

7. Să se mai introducă un angajat corespunzător departamentului 300,
precizând după numele tabelului lista coloanelor în care se introduc valori
(metoda explicita de inserare).

Se presupune că data angajării acestuia este cea curentă (SYSDATE). Salvați înregistrarea;

-- metoda explicita de inserare

INSERT INTO emp (hire_date, job_id, employee_id, last_name, email, department_id)

VALUES (sysdate, 'sa_man', 278, 'nume_278', 'email_278', 300);

COMMIT;

SELECT * FROM emp;

8. Creați un nou tabel, numit EMP1_PNU, care va avea aceeași structură ca și EMPLOYEES, dar nici o înregistrare. Copiați în tabelul EMP1_PNU salariații (din tabelul EMPLOYEES) al căror comision depășește 25% din salariu.

```
CREATE TABLE emp1 AS SELECT * FROM employees;
```

```
DELETE FROM emp1;
```

```
INSERT INTO emp1  
  SELECT *  
  FROM employees  
  WHERE commission_pct > 0.25;
```

```
SELECT * FROM EMP1;
```

-- Ce se intampla daca executam un rollback?

-- R: Se intoarce la CREATE deoarece aceasta instructiune este singura care executa un commit implicit;

-- testam ambele variante

```
ROLLBACK;
```

```
COMMIT;
```