--LABORATOR 7

------------------------------------------------------------------------------

!!!!!! Atentie la denumirea tabelelor si a constrangerilor!!!!!!

se foloseste -> “pnu” unde p reprezintă prima literă a prenumelui, iar nu reprezintă primele două litere ale numelui

------------------------------------------------------------------------------

Exerciţii [I]

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 tabelelor EMPLOYEES, respectiv DEPARTMENTS.

CREATE TABLE EMP\_pnu1 AS SELECT \* FROM employees;

CREATE TABLE DEPT\_pnu1 AS SELECT \* FROM departments;

select \* from employees;

select \* from departments;

2. Listaţi structura tabelelor sursă şi a celor create anterior. Ce se observă?

desc employees;

desc emp\_pnu1;

desc departments;

desc dept\_pnu1;

-- se copiaza structura tabelelor de baza

-- dar nu se copiaza si constrangerile de integritate (cheie primara, cheie externa)

3. Listaţi conţinutul tabelelor create anterior.

select \* from employees;

select \* from emp\_pnu1;

select \* from departments;

select \* from dept\_pnu1;

-- continutul tabelelor se copiaza in totalitate

4. Pentru introducerea constrângerilor de integritate, executaţi instrucţiunile LDD indicate în continuare.

--cheia primara din employees:

ALTER TABLE emp\_pnu1

ADD CONSTRAINT pk\_emp\_pnu1

PRIMARY KEY(employee\_id);

--cheia primara din departments:

ALTER TABLE dept\_pnu1

ADD CONSTRAINT pk\_dept\_pnu1

PRIMARY KEY(department\_id);

--legatura employees->departments prin department\_id

Cheia externa aici department\_id din employees refera cheia primara aici department\_id din departments:

ALTER TABLE emp\_pnu1

ADD CONSTRAINT fk\_emp\_dept\_pnu1

FOREIGN KEY(department\_id) REFERENCES dept\_pnu1(department\_id);

Ce constrângere nu am implementat?

--self join prin manager\_id->employee\_id

ALTER TABLE emp\_pnu1

ADD CONSTRAINT fk\_emp\_sef\_pnu1

FOREIGN KEY(manager\_id) REFERENCES emp\_pnu1(employee\_id);

--departments->employees prin manager\_id->employee\_id

ALTER TABLE dept\_pnu1

ADD CONSTRAINT fk\_dept\_sef\_pnu1 FOREIGN KEY(manager\_id) REFERENCES emp\_pnu1(employee\_id);

--- s-a produs un commit implicit

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

Analizaţi cazurile, precizând care este soluţia corectă şi explicând erorile celorlalte variante.

Pentru a anula efectul instrucţiunii(ilor) corecte, utilizaţi comanda ROLLBACK.

--a) not enough values

-- metoda implicita de inserare in care nu se specifica coloanele in clauza INTO, dupa numele tabelului

INSERT INTO DEPT\_pnu1

VALUES (300, 'Programare');!NUUU MERGEEEEE

desc dept\_pnu1;

--b)corecta

-- metoda explicita de inserare, in care se specifica coloanele in care dorim sa inseram

INSERT INTO DEPT\_pnu1 (department\_id, department\_name)

VALUES (300, 'Programare');

select \* from dept\_pnu1;

-- CREATE, ALTER, DROP -- realizeaza un commit implicit (o salvare a datelor)

rollback;

select \* from dept\_pnu1;

INSERT INTO DEPT\_pnu1 (department\_id, department\_name)

VALUES (300, 'Programare');

commit;

select \* from dept\_pnu1;

rollback; -- anuleaza modificarile pana la ultimul commit (implicit sau explicit)

select \* from dept\_pnu1;

--c) invalid number

INSERT INTO DEPT\_pnu1 (department\_name, department\_id)

VALUES (300, 'Programare');

--d)unique constraint (GRUPA41.PK\_DEPT\_PNU1) violated

INSERT INTO DEPT\_pnu1 (department\_id, department\_name, location\_id)

VALUES (300, 'Programare', null);

INSERT INTO DEPT\_pnu1 (department\_id, department\_name, location\_id)

VALUES (301, 'Programare', null);

commit;

--e) cannot insert NULL into ("GRUPA41"."DEPT\_PNU1"."DEPARTMENT\_ID")

INSERT INTO DEPT\_pnu1 (department\_name, location\_id)

VALUES ('Programare', null);

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).

Determinaţi ca efectele instrucţiunii să devină permanente.

insert into emp\_pnu1

values (257, NULL, 'nume257', 'email257', NULL, SYSDATE, 'IT\_PROG', NULL, NULL, NULL, 300);

desc emp\_pnu1;

select \* from emp\_pnu1

where employee\_id = 257;

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). Salvaţi înregistrarea.

INSERT INTO emp\_pnu1 (hire\_date, job\_id, employee\_id, last\_name, email, department\_id)

VALUES (sysdate, 'sa\_man', 278, 'nume\_278', 'email\_278', 300);

COMMIT ;

11. Să se creeze un fişier (script file) care să permită introducerea de înregistrări în tabelul EMP\_PNU

în mod interactiv. Se vor cere utilizatorului: codul, numele, prenumele si salariul angajatului.

Câmpul email se va completa automat prin concatenarea primei litere din prenume şi a primelor 7 litere din nume.

INSERT INTO emp\_pnu1 (employee\_id, first\_name, last\_name, email, hire\_date, job\_id, salary)

VALUES(&cod, '&&prenume', '&&nume', substr('&prenume',1,1) || substr('&nume',1,7), sysdate, 'it\_prog',&sal);

UNDEFINE prenume;

UNDEFINE nume;

define;

select \* from emp\_pnu1;

rollback;

9. Creaţi un nou tabel, numit EMP1\_PNU, care va avea aceeaşi structură ca şi EMPLOYEES, dar nici o înregistrare.

Copiaţi în tabelul EMP1\_PNU salariaţii (din tabelul EMPLOYEES) al căror comision depăşeşte 25% din salariu.

create table emp1\_pnu1 as select \* from employees;

select \* from emp1\_pnu1;

delete from emp1\_pnu1;

insert into emp1\_pnu1

select \*

from employees

where commission\_pct > 0.25;

rollback; -- ce se intampla daca executam aceasta comanda?

-- rollback anuleaza modificarile pana la ultimul commit - implicit sau explicit

-- anuleaza insert-ul si delete-ul

10. Inseraţi o nouă înregistrare în tabelul EMP\_PNU care să totalizeze salariile, să facă media comisioanelor, iar câmpurile de tip dată să conţină data curentă şi câmpurile de tip caracter să conţină textul 'TOTAL'. Numele şi prenumele angajatului să corespundă utilizatorului curent (USER). Pentru câmpul employee\_id se va introduce valoarea 0, iar pentru manager\_id şi department\_id se va da valoarea null.

INSERT INTO emp\_odi

SELECT 0,USER,USER, 'TOTAL', 'TOTAL',SYSDATE,

'TOTAL', SUM(salary), ROUND(AVG(commission\_pct)), null, null

FROM employees;

12. Creaţi 2 tabele emp2\_pnu şi emp3\_pnu cu aceeaşi structură ca tabelul EMPLOYEES,

dar fără înregistrări (acceptăm omiterea constrângerilor de integritate).

Prin intermediul unei singure comenzi, copiaţi din tabelul EMPLOYEES:

- 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.

Verificaţi rezultatele, apoi ştergeţi toate înregistrările din aceste tabele.

create table emp1\_pnu1 as select \* from employees;

create table emp2\_pnu1 as select \* from employees;

create table emp3\_pnu1 as select \* from employees;

select \* from emp1\_pnu1;

select \* from emp2\_pnu1;

select \* from emp3\_pnu1;

delete from emp1\_pnu1;

delete from emp2\_pnu1;

delete from emp3\_pnu1;

INSERT [ALL | FIRST]

WHEN condiţie THEN INTO…

[WHEN condiţie THEN INTO…

[ELSE INTO …]]

subcerere;

insert all

when salary < 5000 then into emp1\_pnu1

when salary >= 5000 and salary <= 10000 then into emp2\_pnu1

else into emp3\_pnu1

select \* from employees;