--LABORATOR 8

3. Introduceti coloana comision in tabelul ANGAJATI\_pnu. Coloana va avea tipul de date NUMBER(4,2).

ALTER TABLE nume\_tabel

DROP COLUMN/ADD/MODIFY;

ALTER TABLE angajati\_pnu1

ADD(comision number(4,2));

select \* from angajati\_pnu1;

desc angajati\_pnu1;

4. Este posibilă modificarea tipului coloanei salariu în NUMBER(6,2) – 6 cifre si 2 zecimale?

ALTER TABLE angajati\_pnu1

MODIFY(salariu number(6,2)); --column to be modified must be empty to decrease precision or scale

-dar se poate mari chiar daca nu e goala

select \* from angajati\_pnu1;

desc angajati\_pnu1;

5. Setaţi o valoare DEFAULT pentru coloana salariu.

ALTER TABLE angajati\_pnu1

MODIFY(salariu number(8,2) default 100);

select \* from angajati\_pnu1;

desc angajati\_pnu1;

6. Modificaţi tipul coloanei comision în NUMBER(2, 2) şi al coloanei salariu la NUMBER(10,2),

în cadrul aceleiaşi instrucţiuni.

ALTER TABLE angajati\_pnu1

MODIFY(comision number(2,2),

salariu number(10,2)

);

select \* from angajati\_pnu1;

desc angajati\_pnu1;

7. Actualizati valoarea coloanei comision, setând-o la valoarea

0.1 pentru salariaţii al căror job începe cu litera A. (UPDATE)

UPDATE angajati\_pnu1

SET comision = 0.1

WHERE initcap(job) like 'A%';

commit;

select \* from angajati\_pnu1;

desc angajati\_pnu1;

8. Modificaţi tipul de date al coloanei email în VARCHAR2.

ALTER TABLE angajati\_pnu1

MODIFY(email varchar2(15));

9. Adăugaţi coloana nr\_telefon în tabelul ANGAJATI\_pnu, setându-i o valoare implicită.

select \* from angajati\_pnu1;

desc angajati\_pnu1;

ALTER TABLE angajati\_pnu1

ADD(nr\_telefon varchar2(10) default '0754343234');//sir de caractere ca sa inceapa cu 0

10. Vizualizaţi înregistrările existente. Suprimaţi coloana nr\_telefon.

ALTER TABLE angajati\_pnu1

DROP COLUMN nr\_telefon;

11. Creaţi şi tabelul DEPARTAMENTE\_pnu, corespunzător schemei relaţionale:

DEPARTAMENTE\_pnu (cod\_dep# number(2), nume varchar2(15), cod\_director number(4))

specificând doar constrângerea NOT NULL pentru nume (nu precizaţi deocamdată constrângerea de cheie primară);

CREATE TABLE departamente\_pnu

(cod\_dep number(2),

nume varchar2(15) constraint nume\_dep\_pnu not null,

cod\_director number(4)

);

DESC departamente\_pnu;

12. Introduceţi următoarele înregistrări în tabelul DEPARTAMENTE\_pnu:

INSERT INTO departamente\_pnu

VALUES(10, 'Administrativ', 100);

INSERT INTO departamente\_pnu

VALUES(20, 'Proiectare', 101);

INSERT INTO departamente\_pnu

VALUES(30, 'Programare', NULL);

select \* from departamente\_pnu;

commit;

13. Se va preciza apoi cheia primara cod\_dep, fără suprimarea şi recrearea tabelului (comanda ALTER).

ALTER TABLE departamente\_pnu

ADD CONSTRAINT pk\_dep\_pnu PRIMARY KEY(cod\_dep);

select \* from departamente\_pnu;

desc departamente\_pnu;

desc angajati\_pnu1;

14. Să se precizeze constrângerea de cheie externă pentru coloana cod\_dep din ANGAJATI\_pnu:

a) fără suprimarea tabelului (ALTER TABLE);

ALTER TABLE angajati\_pnu1

ADD CONSTRAINT fk\_ang\_dep\_pnu FOREIGN KEY(cod\_dep) REFERENCES departamente\_pnu(cod\_dep);

b) prin suprimarea şi recrearea tabelului, cu precizarea noii constrângeri la nivel de coloană ({DROP, CREATE} TABLE).

De asemenea, se vor mai preciza constrângerile (la nivel de coloană, dacă este posibil):

- PRIMARY KEY pentru cod\_ang; ===

- FOREIGN KEY pentru cod\_sef; ===

- UNIQUE pentru combinaţia nume + prenume; ===

- UNIQUE pentru email; ===

- NOT NULL pentru nume; ===

- verificarea cod\_dep >0;

- verificarea ca salariul sa fie mai mare decat comisionul\*100.

ANGAJATI\_pnu(cod\_ang number(4), nume varchar2(20), prenume varchar2(20), email char(15), data\_ang date,

job varchar2(10), cod\_sef number(4), salariu number(8, 2), cod\_dep number(2))

DROP TABLE angajati\_pnu1;

CREATE TABLE angajati\_pnu1

(cod\_ang number(4) constraint pk\_ang\_pnu primary key,

nume varchar2(20) constraint nume\_ang\_pnu not null,

prenume varchar2(20),

email char(15) unique,

data\_ang date default sysdate,

job varchar2(10),

cod\_sef number(4) constraint fk\_sef\_pnu references angajati\_pnu1(cod\_ang),

salariu number(8,2),

cod\_dep number(2) constraint fk\_ang\_dep\_pnu references departamente\_pnu(cod\_dep),

comision number(2,2),

check(cod\_dep > 0),

constraint num\_pren\_pnu unique(nume, prenume),

constraint verif\_sal check(salariu > comision \* 100)

);

16. Reintroduceţi date în tabel, utilizând (şi modificând, dacă este necesar) comenzile salvate anterior.

INSERT INTO angajati\_pnu1

VALUES(100, 'nume1', 'prenume1', 'email1', sysdate, 'Director ',null, 20000,10, 0.1);

INSERT INTO angajati\_pnu1

VALUES(101,'nume2' , 'prenume2', 'email2',to\_date('02-02-2004','dd-mm- yyyy'),'Inginer', 100, 10000 ,10, 0.2);

INSERT INTO angajati\_pnu1

VALUES(102,'nume3' , 'prenume3', 'email3',to\_date('05-06-2000','dd-mm-yyyy'),'Analist', 101, 5000 ,20, 0.1);

INSERT INTO angajati\_pnu1

VALUES(103,'nume4','prenume4', 'email4', sysdate, 'Inginer ',100,9000,20, 0.1);

INSERT INTO angajati\_pnu1

VALUES(104,'nume5', 'prenume5', 'email5', sysdate, 'Analist', 101, 3000 ,30, 0.1);

commit;

select \* from angajati\_pnu1;

21. Adăugaţi un nou departament, cu numele Analiza, codul 60 şi directorul null în DEPARTAMENTE\_pnu. COMMIT.

INSERT INTO departamente\_pnu

VALUES (60, 'Analiza', null);

SELECT \* FROM departamente\_pnu;

select \* from angajati\_pnu1;

COMMIT;

22. (Incercaţi să) ştergeţi departamentul 20 din tabelul DEPARTAMENTE\_pnu. Comentaţi.

DELETE FROM departamente\_pnu

WHERE cod\_dep = 20; --integrity constraint (GRUPA41.FK\_ANG\_DEP\_PNU) violated - child record found

23. Ştergeţi departamentul 60 din DEPARTAMENTE\_pnu. ROLLBACK.

DELETE FROM departamente\_pnu

WHERE cod\_dep = 60;

SELECT \* FROM departamente\_pnu;

rollback;

24. Se doreşte ştergerea automată a angajaţilor dintr-un departament, odată cu suprimarea departamentului.

Pentru aceasta, este necesară introducerea clauzei ON DELETE CASCADE în definirea constrângerii de cheie externă.

Suprimaţi constrângerea de cheie externă asupra tabelului ANGAJATI\_pnu şi reintroduceţi această constrângere,

specificând clauza ON DELETE CASCADE.

ALTER TABLE angajati\_pnu1

DROP CONSTRAINT FK\_ANG\_DEP\_PNU;

ALTER TABLE angajati\_pnu1

ADD CONSTRAINT FK\_ANG\_DEP\_PNU FOREIGN KEY(cod\_dep)

REFERENCES departamente\_pnu(cod\_dep) ON DELETE CASCADE;

25. Ştergeţi departamentul 20 din DEPARTAMENTE\_pnu. Ce se întâmplă? Rollback.

DELETE FROM departamente\_pnu

WHERE cod\_dep = 20;

SELECT \* from angajati\_pnu1;

select \* from departamente\_pnu;

rollback;

27. Actualizaţi tabelul DEPARTAMENTE\_PNU, astfel încât angajatul având codul 102 să devină directorul departamentului 30. Ştergeţi angajatul având codul 102 din tabelul ANGAJATI\_pnu. Analizaţi efectele comenzii. Rollback.

ALTER TABLE departamente\_pnu

ADD CONSTRAINT cod\_director\_fk FOREIGN KEY(cod\_director)

REFERENCES angajati\_pnu (cod\_ang) ON DELETE SET NULL;

UPDATE departamente\_pnu

SET cod\_director = 102

WHERE cod\_dep = 30;

SELECT \* FROM departamente\_pnu;

SELECT \* FROM angajati\_pnu;

DELETE from angajati\_pnu where cod\_ang = 102;

◼ Putem sterge angajatul 102 pentru ca nu are subalterni si in momentul stergerii, conform constrangerii ON DELETE SET NULL, cod\_director devine NULL;

ROLLBACK;

Este posibilă suprimarea angajatului având codul 101? Comentaţi.

◼ Nu putem sterge angajatul 101 deoarece are subalterni;

28. Adăugaţi o constrângere de tip check asupra coloanei salariu, astfel încât acesta să nu poată depăşi 30000.

ALTER TABLE angajati\_pnu

ADD CONSTRAINT v\_sal\_pnu CHECK (salariu <= 30000);

UPDATE angajati\_pnu

SET salariu = 35000

where cod\_ang = 100; -- nu putem adauga un salariu mai mare de 30 000, conform noii constrangeri adaugate.

29. Dezactivaţi constrângerea creată anterior.

ALTER TABLE angajati\_pnu DISABLE CONSTRAINT v\_sal\_pnu;