--LABORATOR 9

1. Să se creeze o vizualizare VIZ\_EMP30\_PNU, care conţine codul, numele, email-ul si salariul

angajaţilor din departamentul 30. Să se analizeze structura şi conţinutul vizualizării.

Ce se observă referitor la constrângeri? Ce se obţine de fapt la interogarea conţinutului vizualizării?

Inseraţi o linie prin intermediul acestei vizualizări; comentaţi.

CREATE OR REPLACE VIEW VIZ\_EMP30\_PNU AS

(SELECT employee\_id, last\_name, email, salary

FROM emp\_pnu

WHERE department\_id = 30

);

DESC VIZ\_EMP30\_PNU;

SELECT \* FROM VIZ\_EMP30\_PNU;

INSERT INTO VIZ\_EMP30\_PNU

VALUES(559,'last\_name','eemail',10000); --cannot insert NULL into ("GRUPA44"."EMP\_PNU"."HIRE\_DATE")

DROP VIEW VIZ\_EMP30\_PNU;

desc emp\_pnu;

2. Modificaţi VIZ\_EMP30\_PNU astfel încât să fie posibilă inserarea/modificarea conţinutului tabelului de bază prin intermediul ei.

Inseraţi şi actualizaţi o linie (cu valoarea 601 pentru codul angajatului)

prin intermediul acestei vizualizări.

Obs: Trebuie introduse neapărat în vizualizare coloanele care au constrângerea NOT NULL în tabelul de bază

(altfel, chiar dacă tipul vizualizării permite operaţii LMD,

acestea nu vor fi posibile din cauza nerespectării constrângerilor NOT NULL).

CREATE OR REPLACE VIEW VIZ\_EMP30\_PNU AS

(SELECT employee\_id, last\_name, email, salary, hire\_date, job\_id, department\_id

FROM emp\_pnu

WHERE department\_id = 30

);

DESC VIZ\_EMP30\_PNU;

SELECT \* FROM VIZ\_EMP30\_PNU;

SELECT \* FROM EMP\_PNU;

INSERT INTO VIZ\_EMP30\_PNU

VALUES(601, 'last\_name', 'eemail', 10000, SYSDATE, 'IT\_PROG', 30);

SELECT \* FROM VIZ\_EMP30\_PNU;

SELECT \* FROM EMP\_PNU;

Unde a fost introdusă linia?

🡪 la final

Mai apare ea la interogarea vizualizării?

🡪 da

🡪 dar in tabelul de baza nu apare

Ce efect are următoarea operaţie de actualizare?

UPDATE viz\_emp30\_pnu

SET hire\_date = hire\_date-15

WHERE employee\_id = 601;

Ştergeţi angajatul având codul 601 prin intermediul vizualizării. Analizaţi efectul asupra tabelului de bază.

DELETE FROM viz\_emp30\_pnu

WHERE employee\_id = 601;

COMMIT;

3. Să se creeze o vizualizare, VIZ\_EMPSAL50\_PNU, care contine coloanele cod\_angajat, nume, email,

functie, data\_angajare si sal\_anual corespunzătoare angajaţilor din departamentul 50.

Analizaţi structura şi conţinutul vizualizării.

CREATE OR REPLACE VIEW VIZ\_EMPSAL50\_PNU AS

SELECT employee\_id, last\_name, email, job\_id, hire\_date, salary\*12 sal\_anual

FROM emp\_pnu

WHERE department\_id = 50;

DESC VIZ\_EMPSAL50\_PNU;

SELECT \* FROM VIZ\_EMPSAL50\_PNU;

4. a) Inseraţi o linie prin intermediul vizualizării precedente. Comentaţi.

INSERT INTO VIZ\_EMPSAL50\_PNU(employee\_id, last\_name, email, job\_id, hire\_date)

VALUES(568, 'last\_name', 'email000', 'IT\_PROG', sysdate);

select \* from emp\_pnu;

b) Care sunt coloanele actualizabile ale acestei vizualizări? Verificaţi răspunsul în

dicţionarul datelor (USER\_UPDATABLE\_COLUMNS).

select \*

from USER\_UPDATABLE\_COLUMNS

where table\_name = 'VIZ\_EMPSAL50\_PNU';

5. a) Să se creeze vizualizarea VIZ\_EMP\_DEP30\_PNU, astfel încât aceasta să includă coloanele

vizualizării VIZ\_EMP30\_PNU, precum şi numele şi codul departamentului.

Să se introducă aliasuri pentru coloanele vizualizării.

! Asiguraţi-vă că există constrângerea de cheie externă între tabelele de bază ale acestei vizualizări.

CREATE OR REPLACE VIEW VIZ\_EMP\_DEP30\_PNU AS

SELECT v.\*,d.department\_name

FROM VIZ\_EMP30\_PNU v JOIN departments d ON(d.department\_id = v.department\_id);

b) Inseraţi o linie prin intermediul acestei vizualizări.

INSERT INTO VIZ\_EMP\_DEP30\_PNU(employee\_id,last\_name,email,salary,job\_id,hire\_date,department\_id)

VALUES (358, 'lname', 'email', 15000, 'IT\_PROG', sysdate, 30);

SELECT \* FROM VIZ\_EMP\_DEP30\_PNU;

SELECT \* FROM VIZ\_EMP30\_PNU;

select \* from departments;

select \*

from USER\_UPDATABLE\_COLUMNS

where table\_name = 'VIZ\_EMP\_DEP30\_PNU';

d) Ce efect are o operaţie de ştergere prin intermediul vizualizării viz\_emp\_dep30\_pnu? Comentaţi

DELETE FROM VIZ\_EMP\_DEP30\_PNU WHERE employee\_id = 358;

6. Să se creeze vizualizarea VIZ\_DEPT\_SUM\_PNU, care conţine codul departamentului şi pentru fiecare departament

salariul minim, maxim si media salariilor. Ce fel de vizualizare se obţine (complexa sau simpla)?

Se poate actualiza vreo coloană prin intermediul acestei vizualizări?

CREATE OR REPLACE VIEW VIZ\_DEPT\_SUM\_PNU AS

(SELECT department\_id, MIN(salary) min\_sal, MAX(salary) max\_sal, round(AVG(salary)) med\_sal

FROM employees RIGHT JOIN departments USING (department\_id)

GROUP BY department\_id

);

SELECT \* FROM VIZ\_DEPT\_SUM\_PNU;

select \*

from USER\_UPDATABLE\_COLUMNS

where table\_name = 'VIZ\_DEPT\_SUM\_PNU';

🡪 vizualizare complexa nu se poate sterge, updata, insera nimic nicaieri

7. Modificaţi vizualizarea VIZ\_EMP30\_PNU astfel încât să nu permită modificarea sau inserarea de linii

ce nu sunt accesibile ei. Vizualizarea va selecta şi coloana department\_id.

Daţi un nume constrângerii şi regăsiţi-o în vizualizarea USER\_CONSTRAINTS din dicţionarul datelor.

Încercaţi să modificaţi şi să inseraţi linii ce nu îndeplinesc condiţia department\_id = 30.

CREATE OR REPLACE VIEW VIZ\_EMP30\_PNU AS

(SELECT employee\_id, last\_name, email, salary, hire\_date, job\_id, department\_id

FROM emp\_pnu

WHERE department\_id = 30

)

WITH READ ONLY CONSTRAINT verific;

INSERT INTO VIZ\_EMP30\_PNU

VALUES(600, 'last\_name', 'eemail', 10000, SYSDATE, 'IT\_PROG', 50);

--cannot perform a DML operation on a read-only view

9. Să se selecteze numele, salariul, codul departamentului şi salariul maxim din departamentul din care face parte,

pentru fiecare angajat. Este necesară o vizualizare inline?

SELECT last\_name, salary, department\_id, (SELECT MAX(salary), department\_id

FROM employees

WHERE department\_id = E.department\_id) max\_salary

FROM employees E;

--SAU:

SELECT last\_name, salary, E.department\_id, max\_sal

FROM employees E join (SELECT MAX(salary) max\_sal, department\_id

FROM employees

GROUP BY department\_id) max\_salary

ON (E.department\_id = max\_salary.department\_id);

10. Să se creeze o vizualizare VIZ\_SAL\_PNU, ce conţine numele angajaţilor, numele departamentelor,

salariile şi locaţiile (oraşele) pentru toţi angajaţii. Etichetaţi sugestiv coloanele.

Consideraţi ca tabele de bază tabelele originale din schema HR. Care sunt coloanele actualizabile?

CREATE OR REPLACE VIEW VIZ\_SAL\_PNU AS

(SELECT last\_name, department\_name, salary, city

FROM employees JOIN departments USING(department\_id)

JOIN LOCATIONS USING(location\_id)

);

SELECT \* FROM VIZ\_SAL\_PNU;

select \*

from USER\_UPDATABLE\_COLUMNS

where table\_name = 'VIZ\_SAL\_PNU';

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

CREATE SEQUENCE SEQ\_test

INCREMENT BY 10

START WITH 400

MAXVALUE 10000

NOCYCLE

NOCACHE;

SELECT \* FROM dept\_pnu;

INSERT INTO dept\_pnu

VALUES (SEQ\_test.nextval, 'DeptNou', null, null);

DELETE FROM dept\_pnu

WHERE DEPARTMENT\_ID = SEQ\_test.currval; -- nu merge in delete

DELETE FROM dept\_pnu

WHERE DEPARTMENT\_ID = 410;

DROP SEQUENCE SEQ\_test;