Baze de Date
Solutii – Laborator 3 – Laborator 4 – Saptamana 6

--LABORATOR 3 - SAPTAMANA 6

```
SELECT employee_id
FROM employees

UNION ALL

SELECT department_id
FROM departments;

-- EXEMPLE OUTER JOIN
-- afisam toti angajatii indiferent daca au sau nu departament

SELECT e.last_name, e.salary, job_title, e.manager_id, e.department_id
FROM employees e LEFT JOIN jobs j ON (e.job_id = j.job_id)
        LEFT JOIN departments d ON (e.department_id = d.department_id)
        LEFT JOIN locations l ON (d.location_id = l.location_id)
        LEFT JOIN countries c ON (l.country_id = c.country_id);
```

CEZARA BENEGUI

6. Se cer codurile departamentelor al căror nume conţine şirul “re” sau în care lucrează angajaţi având codul job-ului “SA_REP”.

```
-- UNION
select department_id
from departments
where upper(department_name) like '%RE%'

UNION

select department_id
from employees
where upper(job_id) = 'SA_REP';

-- JOIN

select distinct d.department_id
from employees e join departments d on (e.department_id = d.department_id)
where upper(department_name) like '%RE%'
      or upper(job_id) = 'SA_REP';

--De ce varianta aceasta nu returneaza la fel ca cea anterioara?
--R: In acest caz exista departamente fara angajati
```

CEZARA BENEGUI

--Cum le afisam si pe acestea?

--R: utilizam un outer join

```
select distinct d.department_id
from employees e right join departments d on (e.department_id =
d.department_id)
where upper(department_name) like '%RE%'
    or upper(job_id) = 'SA_REP';
```

--Cum afisam si angajatii care nu au departament?

```
select distinct d.department_id
from employees e full join departments d on (e.department_id =
d.department_id)
where upper(department_name) like '%RE%'
    or upper(job_id) = 'SA_REP';
```

8. Să se obțină codurile departamentelor în care nu lucreaza nimeni

(nu este introdus nici un salariat în tabelul employees). Se cer doua solutii.

```
-- MINUS
select department_id
from departments -- din lista tuturor departamentelor din bd (unice)
        -- department_id este cheie primara in departments
```

CEZARA BENEGUI

MINUS --eliminam

select department_id

from employees; -- departamentele in care lucreaza angajati

-- department_id este cheie externa in employees

--SAU UTILIZAND JOIN

-- departamentele care au angajati

select e.department_id

from employees e join departments d on (e.department_id = d.department_id);

-- departamentele care nu au angajati

select *

from employees e right join departments d on (e.department_id = d.department_id)

where e.department_id is null;

--where employee_id is null

CEZARA BENEGUI

-- LABORATOR 4 - SAPTAMANA 6

LMD - select, insert, update, delete, merge

LDD - CREATE, ALTER, DROP, TRUNCATE - executa un commit implicit

LCD - ROLLBACK, COMMIT, SAVEPOINT

```
CREATE TABLE TEST
(ID NUMBER(10),
NUME VARCHAR2(20)
);

INSERT INTO TEST VALUES (1, 'ABC');
INSERT INTO TEST VALUES (1, 'ABC');

SELECT * FROM TEST;
```

ROLLBACK; - ANULEAZA EFECTUL UNOR INSTRUCTIUNI

- anuleaza pana la ultimul commit

COMMIT; - salveaza efectul unor instructiuni

- salveaza pana la ultimul commit

CEZARA BENEGUI

- sau de la ultima conectare pe server

desc test;

insert

insert

insert

rollback - anuleaza cele 3 inserari

update

commit - salveaza ultimul update

delete

delete

insert

create - executa commit automat -> salveaza insertul si cele doua stergeri

rollback -> nu are efect deoarece create a executat commit implicit

CEZARA BENEGUI

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}

OBS!!!

-- coloanele din lista SELECT trebuie să corespundă, ca număr şi tip, celor precizate

-- în clauza INTO

1. Să se creeze tabelele EMP_pnu, DEPT_pnu prin copierea structurii şi conţinutului

tabelelor EMPLOYEES, respectiv DEPARTMENTS.

-- în care şirul de caractere “pnu” ->

-- p reprezintă prima literă a prenumelui ->

-- iar nu reprezintă primele două litere ale numelui)

CREATE TABLE EMP_pnu AS SELECT * FROM employees;

CREATE TABLE DEPT_pnu AS SELECT * FROM departments;

CEZARA BENEGUI

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

-- listam structura

desc emp_pnu;

desc employees;

3. Listaţi conţinutul tabelelor create anterior.

--lista continutul

select * from emp_pnu;

select * from employees;

CEZARA BENEGUI

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

```
ALTER TABLE emp_pnu
ADD CONSTRAINT pk_emp_pnu PRIMARY KEY(employee_id);

ALTER TABLE dept_pnu
ADD CONSTRAINT pk_dept_pnu PRIMARY KEY(department_id);

ALTER TABLE emp_pnu
ADD CONSTRAINT fk_emp_dept_pnu FOREIGN KEY(department_id)
REFERENCES dept_pnu(department_id);
```

Obs: Ce constrângere nu am implementat?

```
ALTER TABLE emp_pnu
ADD CONSTRAINT fk_emp_emp_pnu FOREIGN KEY(manager_id)
REFERENCES emp_pnu(employee_id); -- managerul unui angajat

ALTER TABLE dept_pnu
ADD CONSTRAINT fk_dept_emp_pnu FOREIGN KEY(manager_id)
REFERENCES emp_pnu(employee_id); -- managerul de departament
```

CEZARA BENEGUI