**Laborator 6**

1. Informatii despre orasele si departamentele care se gasesc in acele locatii. Se vor afisa informatii chiar si despre locatiile in care nu se gasesc departamente.

Plus se pune la departamente deoarece unele locatii **nu** **au departamente,** deci la departamente completam cu null.

select city, state\_province, country\_id, department\_id, department\_name, manager\_id

from locations l, departments d

where l.location\_id = d.location\_id (+);

La urmatorul exemplu se vor afisa 45 de rezultate datorita accesului la conexiune.

select city, state\_province, country\_id, department\_id, department\_name, manager\_id

from locations l full outer join departments d on (l.location\_id = d.location\_id);

1. Sa se obtina departamentele in care nu lucreaza nimeni (nu e introdus nici un salariat).

select department\_name, department\_id

from departments

minus

select d.department\_name, d.department\_id

from departments d, employees e

where e.department\_id=d.department\_id;

Alta varianta!

select department\_id

from departments

minus

select unique department\_id

from employees;

Varianta cu subcerere!

select d.department\_id

from departments d

where d.department\_id not in (select unique e.department\_id from employees e where e.department\_id is not null);

1. Codurile departamentelor al caror nume contine sirul -re- si in care lucreaza angajati cu codul job-ului -HR\_REP-.

select d.department\_id

from departments d

where lower(department\_name) like '%re%'

intersect

select d.department\_id

from departments d, employees e

where d.department\_id=e.department\_id and upper(job\_id) like 'HR\_REP';

1. Copie tabele.

create table EMPLOYEES\_EIS as select \* from employees;

select \* from employees\_eis;

create table DEPARTMENTS\_EIS as select \* from departments;

select \* from departments;

1. Stergere tabel.

drop table DEPARTMENTS\_EIS;

1. Descriere tabel.

describe employees\_eis;

describe departments\_eis;

1. Setare cheie primara pe o tabela.

!!! Dupa ce am copiat tabelele la punctul 4, cheile primare nu mai au constrangere not null, deci nu mai sunt setate ca fiind chei primare. Trebuie sa setam noi asta.

alter table employees\_eis

add constraint pk\_employees\_eis primary key (employee\_id);

alter table departments\_eis

add constraint pk\_departments\_eis primary key (department\_id);

1. Setare cheie externa pe o tabela.

!!! Dupa ce am copiat tabelele la punctul 4, cheile externe nu mai sunt setate. Trebuie sa setam noi asta.

alter table employees\_eis

add constraint fk\_employees\_departments\_eis foreign key (department\_id)

references departments\_eis(department\_id);

1. Verificare constrangeri.

select \*

from user\_constraints

where lower(table\_name) = 'employees\_eis';

1. Inserare valori in tabele.

insert into departments\_eis

values (300, 'Programare', null, null);

Trebuie sa avem neaparat atatea valori in paranteze cate sunt in descrierea tabelului! Daca nu vrem sa facem asta, specificam unde introducem, iar restul valorilor sunt puse automat null. Datele trebuie sa corespunda, sa fie puse in ordinea corecta (de exemplu nu se poate converti department\_name la department\_id)!

insert into departments\_eis (department\_id, department\_name)

values (300, 'Programare')

Verificare inserare:

select \* from departments\_eis where department\_id = 300;

Prin **rollback** se anuleaza ultimele operatii facute pe tabele pana la un commit (inserare, stergere, etc.)

Prin **commit** se seteaza pentru totdeauna operatia in tabel si apare si pentru ceilalti ulilizatori.

**CREATE TABLE** si **DROP TABLE** fac commit-ul automat.

Comenzi laborator:

create table EMPLOYEES\_EIS as select \* from employees;

select \* from employees\_eis;

create table DEPARTMENTS\_EIS as select \* from departments;

select \* from departments;

drop table DEPARTMENTS\_EIS;

create table DEPARTMENTS\_EIS as select \* from departments;

select \* from departments;

describe employees\_eis;

describe departments\_eis;

alter table employees\_eis

add constraint pk\_employees\_eis primary key (employee\_id);

alter table departments\_eis

add constraint pk\_departments\_eis primary key (department\_id);

alter table employees\_eis

add constraint fk\_employees\_departments\_eis foreign key (department\_id)

references departments\_eis(department\_id);

select \*

from user\_constraints

where lower(table\_name) = 'employees\_eis';

select \*

from user\_constraints

where lower(table\_name) = 'departments\_eis';

insert into departments\_eis

values (300, 'Programare', null, null);

select \* from departments\_eis where department\_id = 300;

insert into departments\_eis (department\_id, department\_name)

values (300, 'Programare');

select \* from departments\_eis where department\_id = 300;

commit;