--LABORATOR 6 - continuare

9. Să se afişeze lista angajaţilor care au lucrat NUMAI pe proiecte

conduse de managerul de proiect având codul 102.

select \* from projects; -- managerul 102 conduce proiectele p1 si p3

select \* from works\_on; -- angajatii care lucreaza NUMAI pe proiecte conduse de 102

-- pot lucra la toate proiectele lui 102 sau doar la o parte din ele

-- dar acestia nu pot lucra la alte proiecte care nu apartin managerului 102

=> 136, 140, 150, 162, 176

select employee\_id

from works\_on

where project\_id IN (select project\_id

from projects

where project\_manager = 102

) -- obtin angajatii care lucreaza la toate proiectele conduse de 102

-- sau doar la o parte din proiectele conduse de 102

-- dar printre acesti angajati pot fi si angajati care lucreaza la alte proiecte

-- care nu il au manager pe 102 (ex cazul prezentat mai jos cu ang500)

/\*

ex general - nu exista in baza de date

ang300 - p1 si p2 -> 300 p1 /

ang400 - p1 si p3 -> 400 p1 / 400 p3

ang500 - p1, p3, p2 -> 500 p1 / 500 p3

\*/

MINUS -- din care elimin angajatii care lucreaza la alte proiecte care nu sunt conduse de 102

select employee\_id

from works\_on

where project\_id not in (select project\_id

from projects

where project\_manager = 102

);

10. a) Să se obţină numele angajaţilor care au lucrat CEL PUTIN pe aceleaşi proiecte

ca şi angajatul având codul 200.

select \* from works\_on; -- ang 200 lucreaza la proiectele p2 si p3

CEL PUTIN -> angajatii trebuie sa lucreze MACAR - cel putin - pe aceleasi proiecte la care lucreaza ang 200

-> ei pot lucra si la alte proiecte dar obligatoriu trebuie sa lucreze la toate proiectele lui 200

ex: ang400 - p2 - gresit pt ca ang400 nu lucreaza cel putin pe aceleasi proiecte cu 200

ang500 - p2 p3 - corect

ang600 - p1 p2 p3 - corect pt ca lucreaza cel putin la proiectele lui 200, chiar daca are si alte proiecte in plus

=> in baza noastra de date avem angajatii: 101, 145, 148

select employee\_id, last\_name

from employees join works\_on using (employee\_id)

where project\_id IN (select project\_id

from works\_on

where employee\_id = 200

)

and employee\_id != 200

group by employee\_id, last\_name

having count(project\_id) = (select count(project\_id)

from works\_on

where employee\_id = 200

);

b) Să se obţină numele angajaţilor care au lucrat CEL MULT pe aceleaşi proiecte

ca şi angajatul având codul 200.

select \* from works\_on; -- ang 200 lucreaza la p2 si p3

CEL MULT - angajatii trebuie sa lucreza la toate proiectele lui 200

- sau MACAR la un proiect la care lucreaza 200

- dar angajatii nu pot lucra la alte proiecte la care ang 200 nu lucreaza

=> angajatii 101, 145, 148, 150 (doar p3), 162 (doar p3), 176 (doar p3)

ex general - ang500 lucreaza p1, p2, p3 -> nu afiseaza acest ang pt ca lucreaza si alte proiecte

-> fata de cele la care lucreaza ang 200

-- metoda 1

select employee\_id, last\_name

from employees join works\_on using (employee\_id)

where project\_id IN (select project\_id

from works\_on

where employee\_id = 200

) -- angajatii care lucreaza la toate proiectele lui 200 sau doar la o parte din ele

and employee\_id != 200

MINUS -- eliminam angajatii care lucreaza la alte proiecte fata de proiectele ang 200

select employee\_id, last\_name

from employees join works\_on using (employee\_id)

where project\_id not in (select project\_id

from works\_on

where employee\_id = 200

);

-- metoda 2

select employee\_id, last\_name

from employees join works\_on using (employee\_id)

where project\_id IN (select project\_id

from works\_on

where employee\_id = 200

) -- angajatii care lucreaza la toate proiectele lui 200 sau doar la o parte din ele

and employee\_id != 200

MINUS -- eliminam angajatii care lucreaza la alte proiecte fata de proiectele ang 200

select employee\_id, last\_name

from employees join works\_on using (employee\_id)

where project\_id in (select project\_id

from works\_on -- din lista tuturor proiectelor la care lucreaza angajati

MINUS -- ELIMINAM proiectele la care lucreaza ang 200

select project\_id

from works\_on

where employee\_id = 200 --> obtin in aceasta subcerere proiectele la care nu lucreaza 200

);

--metoda 3

select employee\_id, last\_name

from employees join works\_on using (employee\_id)

where project\_id IN (select project\_id

from works\_on

where employee\_id = 200

)

and employee\_id != 200

group by employee\_id, last\_name

having count(project\_id) <= (select count(project\_id)

from works\_on

where employee\_id = 200

) -- <= inseamna ca angajatii lucreaza la toate proiectele lui 200

-- sau doar la o parte din ele

MINUS

select employee\_id, last\_name

from employees join works\_on using (employee\_id)

where project\_id in (select project\_id

from works\_on -- din lista tuturor proiectelor la care lucreaza angajati

MINUS -- ELIMINAM proiectele la care lucreaza ang 200

select project\_id

from works\_on

where employee\_id = 200 --> obtin in aceasta subcerere proiectele la care nu lucreaza 200

);

11. Să se obţină angajaţii care au lucrat pe ACELEASI proiecte ca şi angajatul având codul 200.

-- ang 200 lucreaza la p2 si p3

-- aceleasi proiecte -> angajatii sa lucreze EXACT pe proiectele la care lucreaza ang 200

-- ang nu pot lucra si la alte proiecte la care ang 200 nu lucreaza

-> 101, 145, 148

select \* from works\_on;

select employee\_id, last\_name

from employees join works\_on using (employee\_id)

where project\_id IN (select project\_id

from works\_on

where employee\_id = 200

)

and employee\_id != 200

group by employee\_id, last\_name

having count(project\_id) = (select count(project\_id)

from works\_on

where employee\_id = 200

)

MINUS

select employee\_id, last\_name

from employees join works\_on using (employee\_id)

where project\_id in (select project\_id

from works\_on -- din lista tuturor proiectelor la care lucreaza angajati

MINUS -- ELIMINAM proiectele la care lucreaza ang 200

select project\_id

from works\_on

where employee\_id = 200 --> obtin in aceasta subcerere proiectele la care nu lucreaza 200

);

-- se poate inlocui ang 200 cu ang 176 pentru verificare

--variabile de substitutie

8. Să se determine departamentele având media salariilor mai mare decât un număr dat.

select department\_id, round(avg(salary))

from employees

group by department\_id

having avg(salary) >&p;

select department\_id, round(avg(salary))

from employees

group by department\_id

having avg(salary) > &&p;

define;

undefine p;

15. Sa se afiseze o coloana aleasa de utilizator, dintr-un tabel ales de utilizator,

ordonand dupa aceeasi coloana care se afiseaza. De asemenea, este obligatorie precizarea unei conditii WHERE.

SELECT &&p\_coloana -- && determina ca valoarea lui p\_coloana sa nu mai

--fie ceruta si pentru clauza ORDER BY, urmand sa

--fie utilizata valoarea introdusa aici pentru toate

--aparitiile ulterioare ale lui &p\_coloana

FROM &p\_tabel

WHERE &p\_where

ORDER BY &p\_coloana;

define;

I.

SELECT employee\_id, last\_name, salary, department\_id

FROM employees

WHERE employee\_id = &p\_cod;

II.

DEFINE p\_cod;

SELECT employee\_id, last\_name, salary, department\_id

FROM employees

WHERE employee\_id = &p\_cod;

UNDEFINE p\_cod;

III.

DEFINE p\_cod=200;

SELECT employee\_id, last\_name, salary, department\_id

FROM employees

WHERE employee\_id = &p\_cod;

UNDEFINE p\_cod;

define;

IV.

ACCEPT p\_cod PROMPT "cod= ";

SELECT employee\_id, last\_name, salary, department\_id

FROM employees

WHERE employee\_id = &p\_cod;

16. Să se realizeze un script (fişier SQL\*Plus) prin care să se afişeze numele,

job-ul şi data angajării salariaţilor care au început lucrul între 2 date calendaristice introduse de utilizator.

Să se concateneze numele şi job-ul, separate prin spaţiu şi virgulă, şi să se eticheteze coloana "Angajati".

Se vor folosi comanda ACCEPT şi formatul pentru data calendaristica MM/DD/YY.

ACCEPT data\_inceput PROMPT 'Introduceti data de inceput'

ACCEPT data\_sfarsit PROMPT 'Introduceti data de sfarsit'

SELECT last\_name, job\_id, hire\_date

FROM employees

WHERE hire\_date BETWEEN TO\_DATE('&data\_inceput', 'mm/dd/yy')

AND TO\_DATE('&data\_sfarsit', 'mm/dd/yy');

define;

TEMA 7:

Ex1: Cati subalterni are fiecare angajat?

Se vor afisa codul, numele, prenumele si numarul de subalterni.

--unesc tabelul cu managerii si numarul lor de angajati

--cu tabelul cu employee\_id care nu sunt manageri si au 0 angajati

with tabel1 as (select manager\_id ,count(employee\_id) numar

from employees

group by manager\_id

union

select employee\_id , 0

from employees

where employee\_id not in (select manager\_id

from employees

group by manager\_id

)

)

select employee\_id, last\_name, first\_name, numar

from employees ee, tabel1 tt

where ee.employee\_id=tt.manager\_id;

Ex3: Sa se listeze pentru fiecare angajat orasul in care a lucrat cele mai multe zile

with tabel as (

select employee\_id,department\_id, max(to\_date(end\_date, 'dd-mm-yy')-to\_date(start\_date, 'dd-mm-yy'))

from job\_history

group by employee\_id, department\_id--114, 102,101,200,122,201,176

union

select employee\_id, department\_id, 0

from employees

where employee\_id not in (select employee\_id

from job\_history

group by employee\_id

)

)

select e.employee\_id, e.last\_name, l.city

from employees e join tabel t on (e.employee\_id=t.employee\_id)

join departments d on (t.department\_id=d.department\_id)

join locations l on (l.location\_id=d.location\_id);

Ex2: Care sunt departamentele (cod si nume) care contin cel putin doua job-uri distincte?

select department\_id, department\_name

from departments

where department\_id in (

select e.department\_id

from employees e join jobs j on (e.job\_id=j.job\_id)

where e.department\_id is not null

group by e.department\_id, j.job\_id

having count(j.job\_id)>=2

);