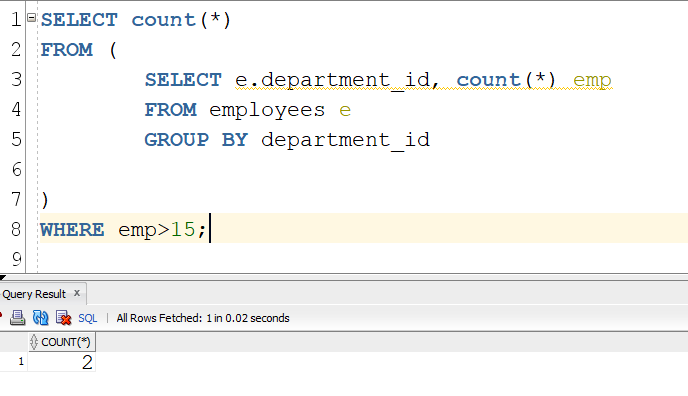
# Tema Laborator saptamana 3

Cocheci Cristiana, 252

25. Să se obţină numărul departamentelor care au cel puţin 15 angajaţi.



SELECT count(\*)

FROM (

SELECT e.department\_id, count(\*) emp

FROM employees e

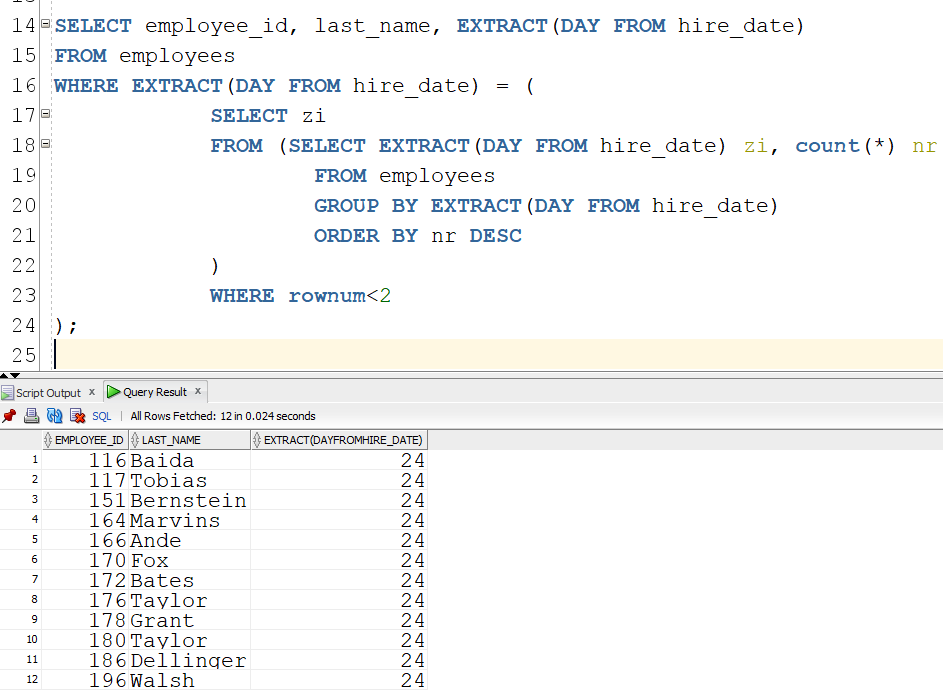
GROUP BY department\_id

)

WHERE emp>15;

26. Sa se afiseze salariatii care au fost angajati în aceeaşi zi a lunii în care cei mai multi dintre

salariati au fost angajati.



SELECT employee\_id, last\_name, EXTRACT(DAY FROM hire\_date)

FROM employees

WHERE EXTRACT(DAY FROM hire\_date) = (

SELECT zi

FROM (SELECT EXTRACT(DAY FROM hire\_date) zi, count(\*) nr

FROM employees

GROUP BY EXTRACT(DAY FROM hire\_date)

ORDER BY nr DESC

)

WHERE rownum<2 );

27. Sa se afiseze numele si salariul celor mai prost platiti angajati din fiecare departament.

SELECT last\_name, salary, nvl(to\_char(department\_id),'fara\_departamnet')

FROM employees

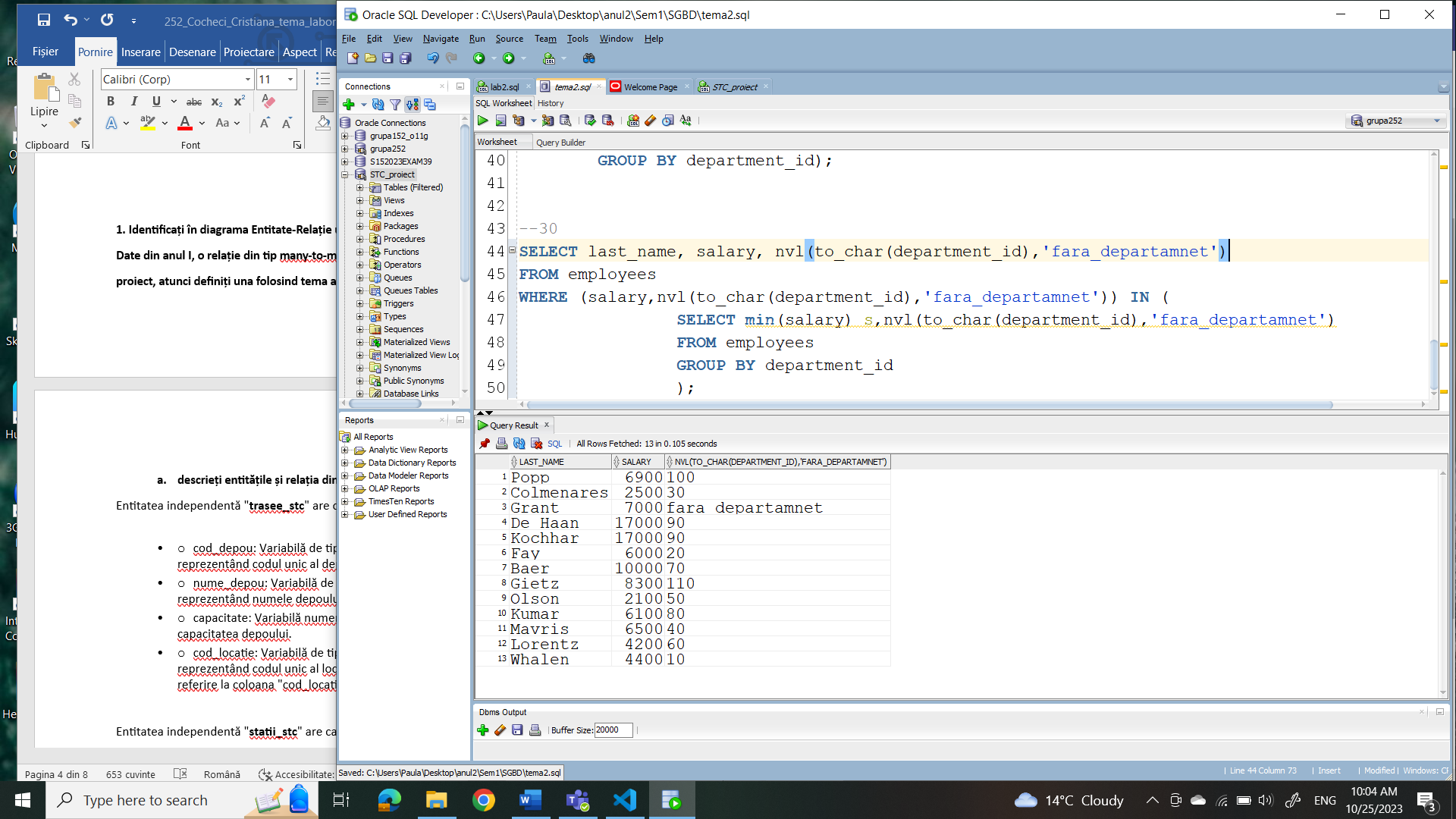
WHERE (salary,nvl(to\_char(department\_id),'fara\_departamnet')) IN (

SELECT min(salary) s,nvl(to\_char(department\_id),'fara\_departamnet')

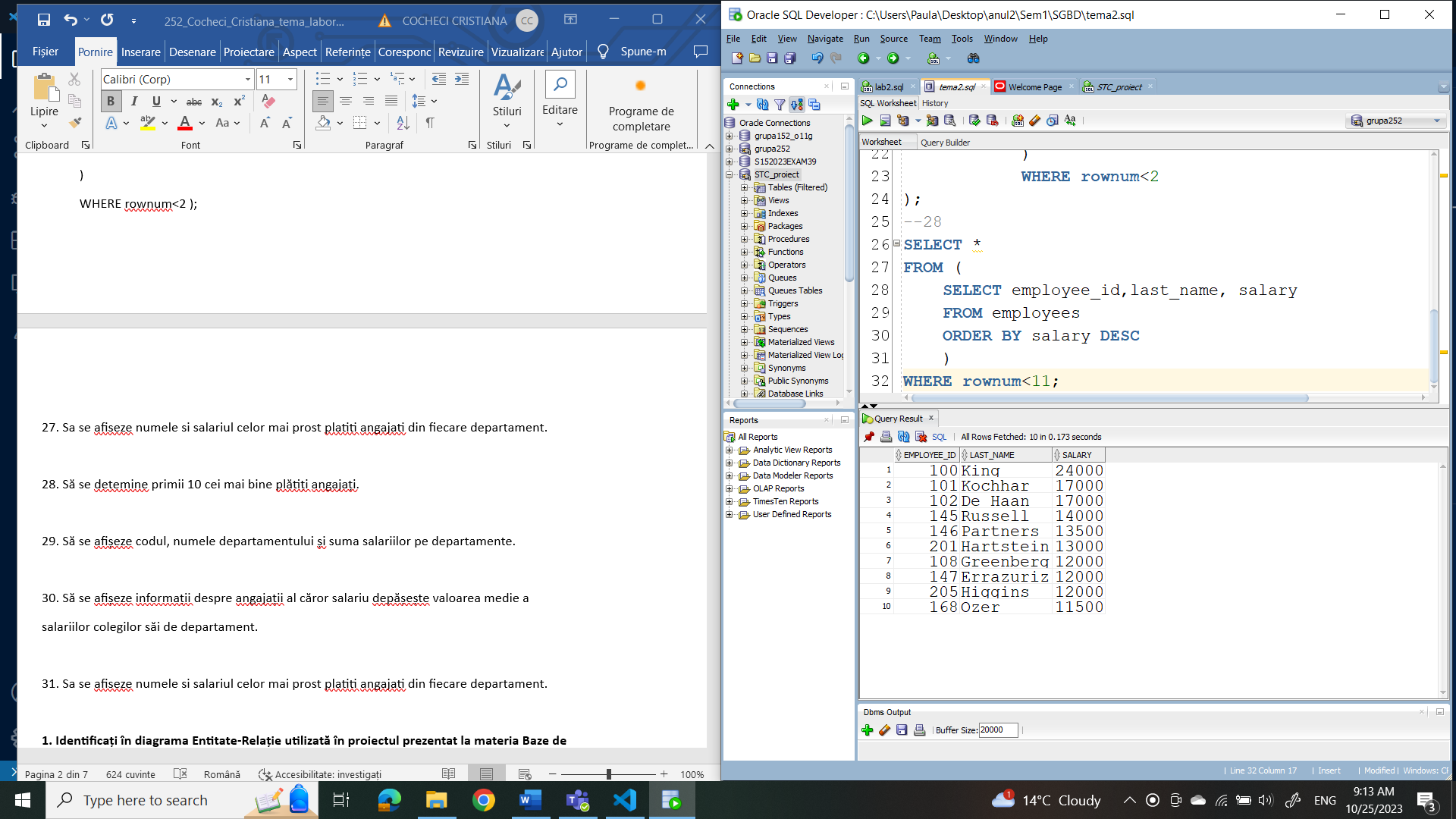
FROM employees

GROUP BY department\_id

);



28. Să se detemine primii 10 cei mai bine plătiţi angajaţi.



--28

SELECT \*

FROM (

SELECT employee\_id,last\_name, salary

FROM employees

ORDER BY salary DESC

)

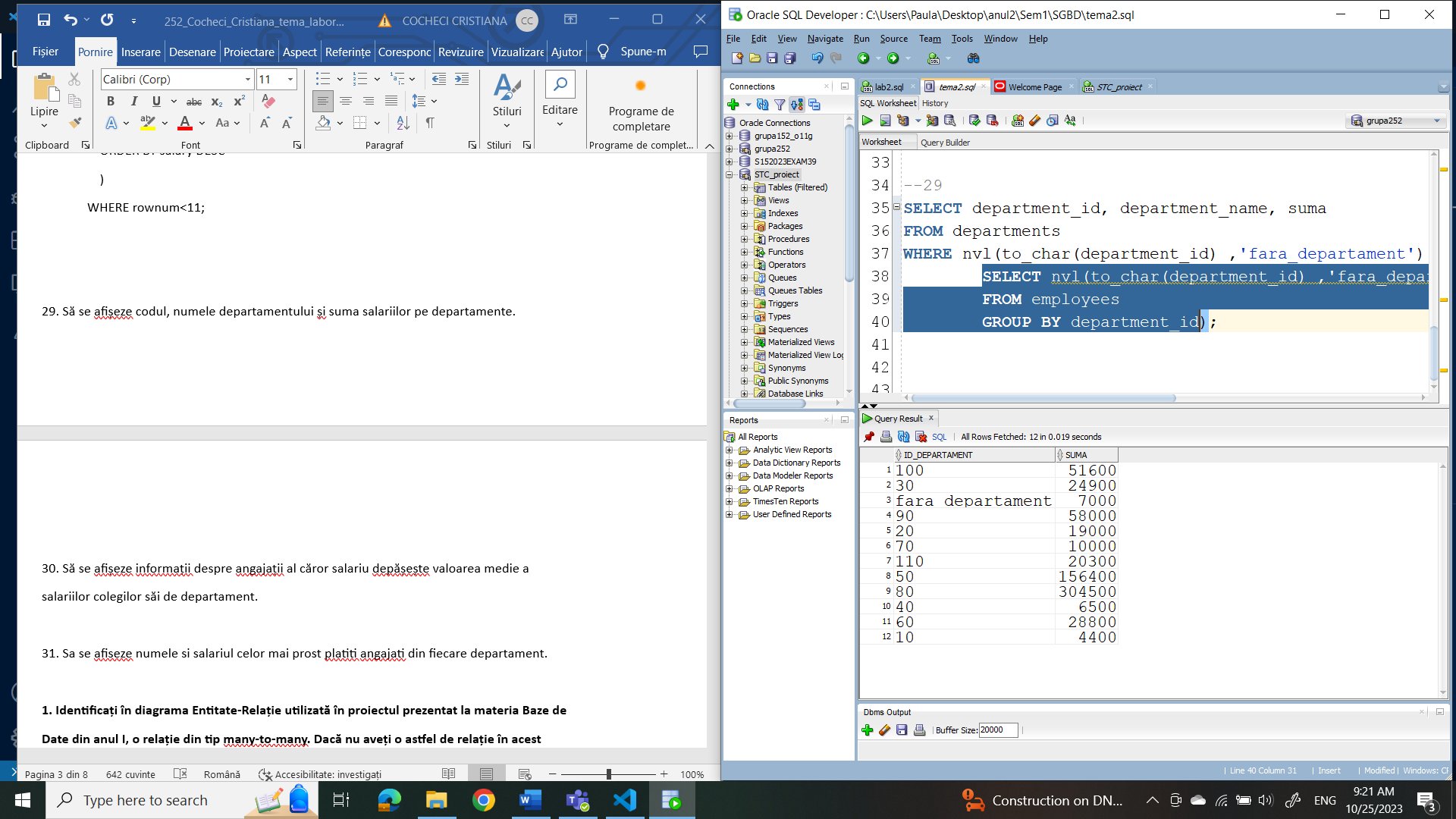
WHERE rownum<11;

29. Să se afişeze codul, numele departamentului şi suma salariilor pe departamente.

SELECT nvl(to\_char(department\_id) ,'fara\_departament') id\_departament, nvl(sum(salary),0) suma

FROM employees

GROUP BY department\_id



30. Să se afişeze informaţii despre angajaţii al căror salariu depăşeşte valoarea medie a

salariilor colegilor săi de departament.

SELECT employee\_id,salary, nvl(to\_char(department\_id),'fara\_departamnet') dep

FROM employees e

WHERE salary>

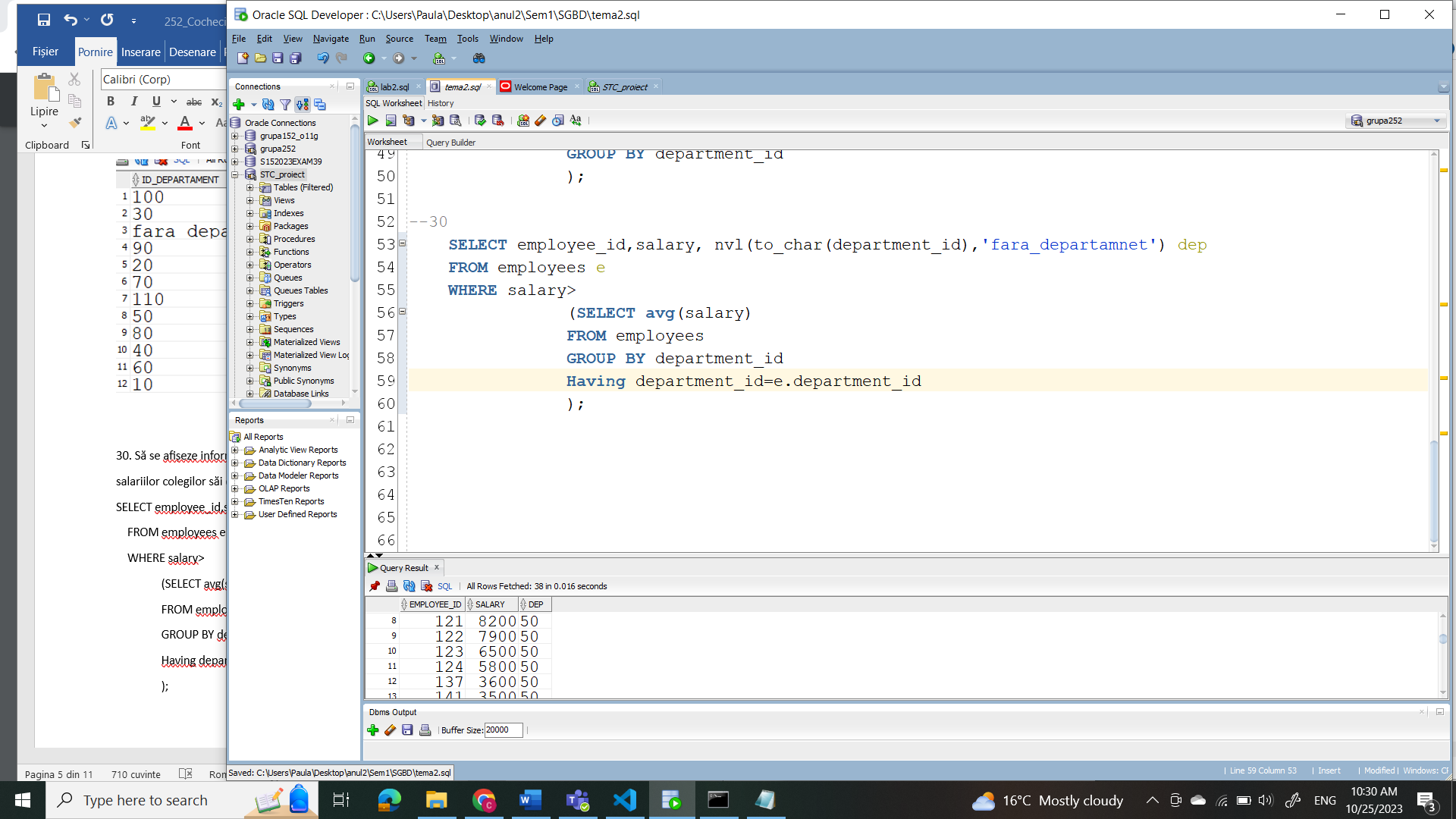
(SELECT avg(salary)

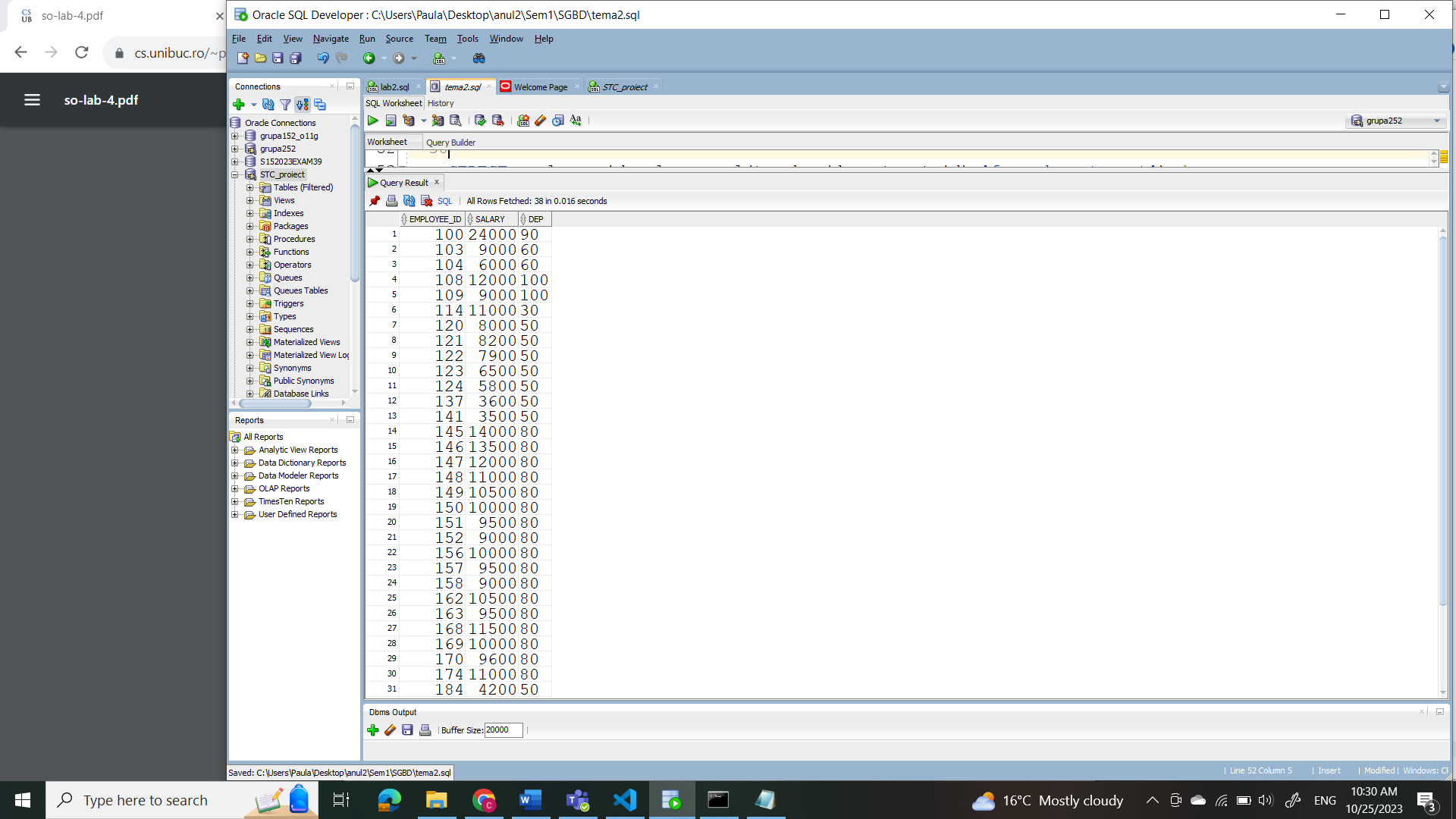
FROM employees

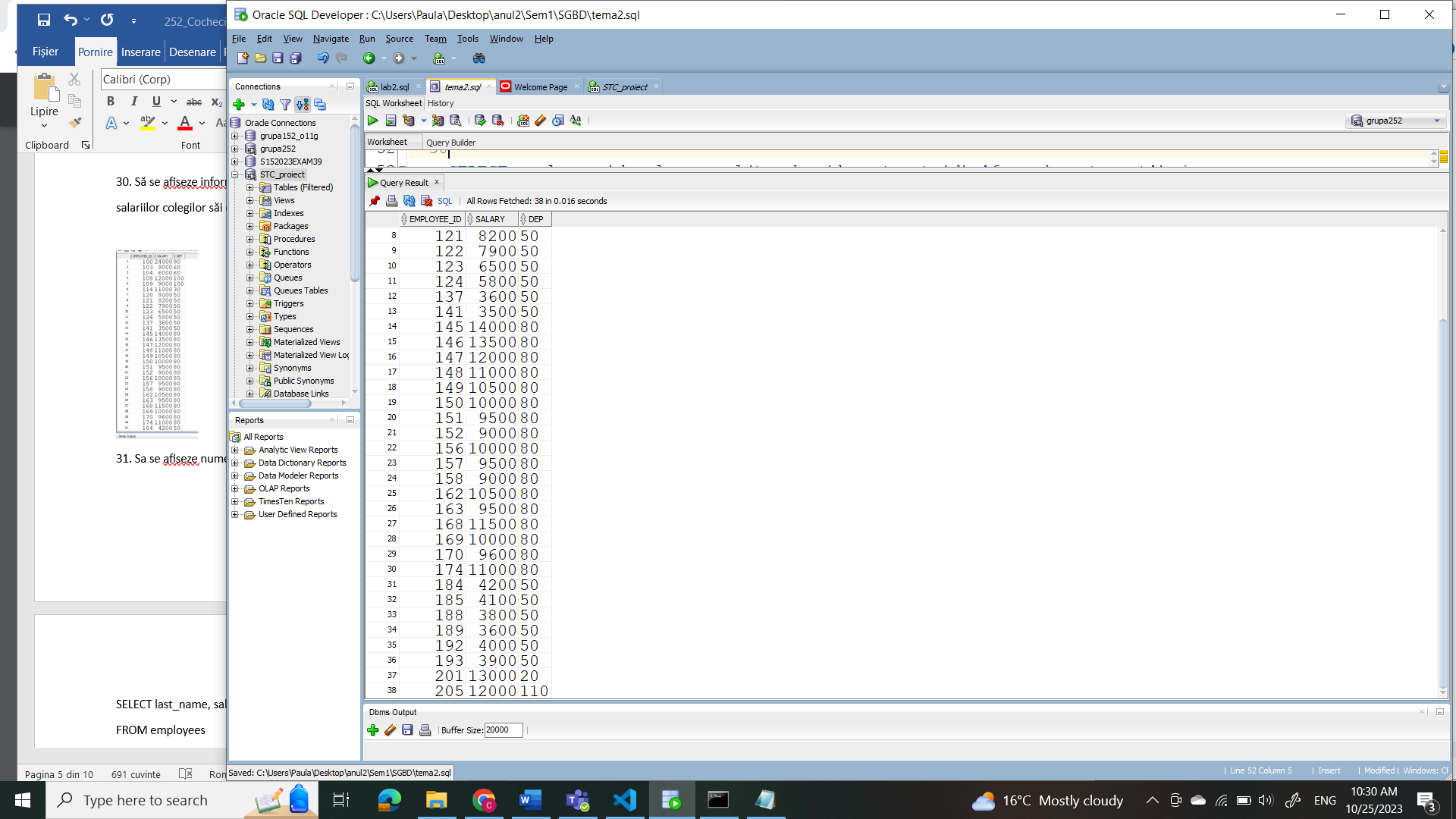
GROUP BY department\_id

Having department\_id=e.department\_id

);







31. Sa se afiseze numele si salariul celor mai prost platiti angajati din fiecare departament.

SELECT last\_name, salary, nvl(to\_char(department\_id),'fara\_departamnet')

FROM employees

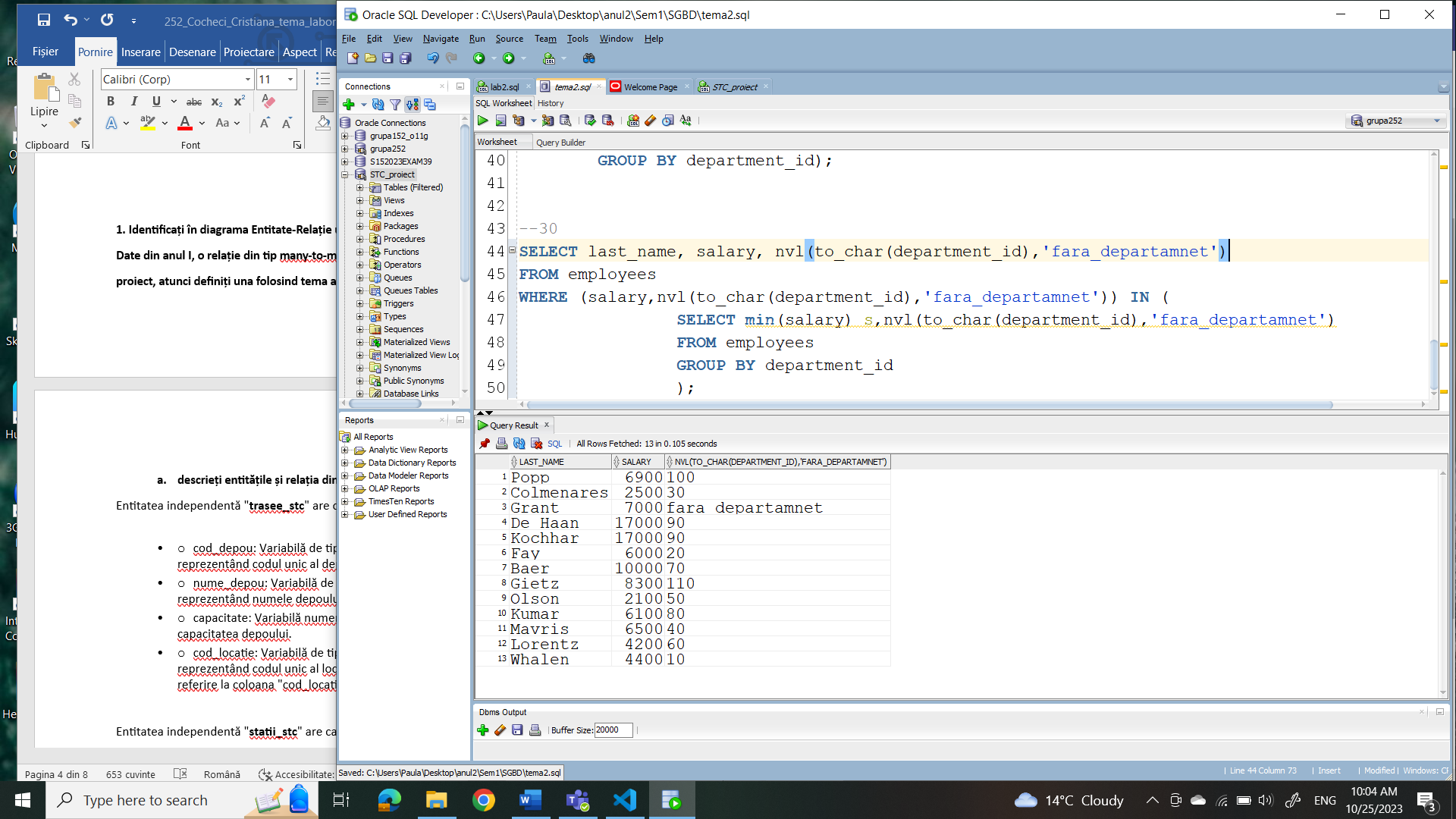
WHERE (salary,nvl(to\_char(department\_id),'fara\_departamnet')) IN (

SELECT min(salary) s,nvl(to\_char(department\_id),'fara\_departamnet')

FROM employees

GROUP BY department\_id

);



**1. Identificați în diagrama Entitate-Relație utilizată în proiectul prezentat la materia Baze de**

**Date din anul I, o relație din tip many-to-many. Dacă nu aveți o astfel de relație în acest**

**proiect, atunci definiți una folosind tema aleasă în acel proiect:**

1. **descrieți entitățile și relația dintre acestea;**

Entitatea independentă "**trasee\_stc**" are ca atribute:

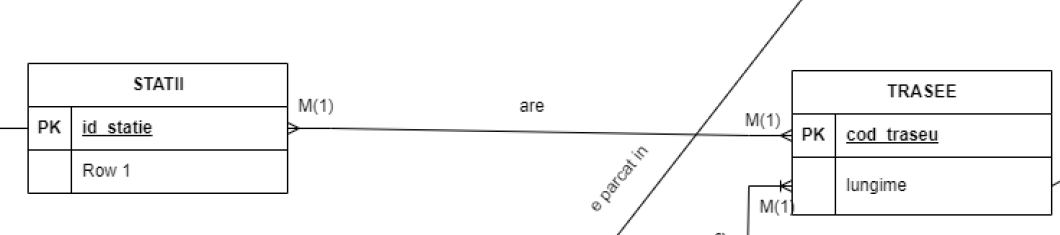
* o cod\_depou: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al depoului.
* o nume\_depou: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând numele depoului.
* o capacitate: Variabilă numerică cu o lungime maximă de 5 cifre, reprezentând capacitatea depoului.
* o cod\_locatie: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al locației. Această coloană este o cheie străină care face referire la coloana "cod\_locatie" din tabela "locatii\_stc".

Entitatea independentă "**statii\_stc**" are ca atribute:

* o cod\_statie: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al stației.
* o nume\_statie: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând numele stației.
* o cod\_locatie: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al locației. Această coloană este o cheie străină care face referire la coloana "cod\_locatie" din tabela "locatii\_stc".

**TRASEU\_are\_STATII** = relatie dintre entitatile TRASEE si STATII reprezentand statiile pe care le are un traseu. Relatia are cardinalitate minima 1:1 si cardinalitate maxima n:n (un traseu poate avea mai multe statii si o statie poate apartine de mai multe trasee).

1. **realizați diagrama Entitate-Relație doar pentru această parte din proiect;**

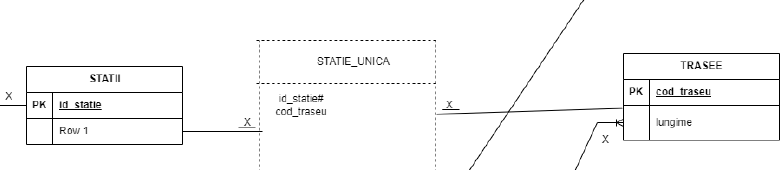


**c. descrieți modul de transformare al acestei relații în diagrama conceptuală, precizând**

**toate cheile primare, cheile externe și alte atribute esențiale;**

Se adauga un tabel asociativ „Statie unica” cu cheie primara compusa din cheile primare ale celorlalte doua trasee. Astfel relatia M:M se transforma in doua relatii de tip 1:M.

1. **realizați diagrama conceptuală doar pentru această parte din proiect;**



**e. pe baza diagramei conceptuale de la punctul d, definiți în SQL tabelele și toate**

**constrângerile necesare;**

CREATE TABLE trasee\_stc(

cod\_traseu VARCHAR2(25) primary key,

capat1 VARCHAR(25),

capat2 VARCHAR(25),

lungime NUMBER(7)

);

CREATE TABLE statii\_stc(

cod\_statie VARCHAR2(25) primary key,

nume\_statie VARCHAR2(25)

);

CREATE TABLE statie\_unica\_stc(

cod\_statie VARCHAR2(25) references statii\_stc(cod\_statie),

cod\_traseu VARCHAR2(25) references trasee\_stc(cod\_traseu),

primary key (cod\_statie, cod\_traseu)

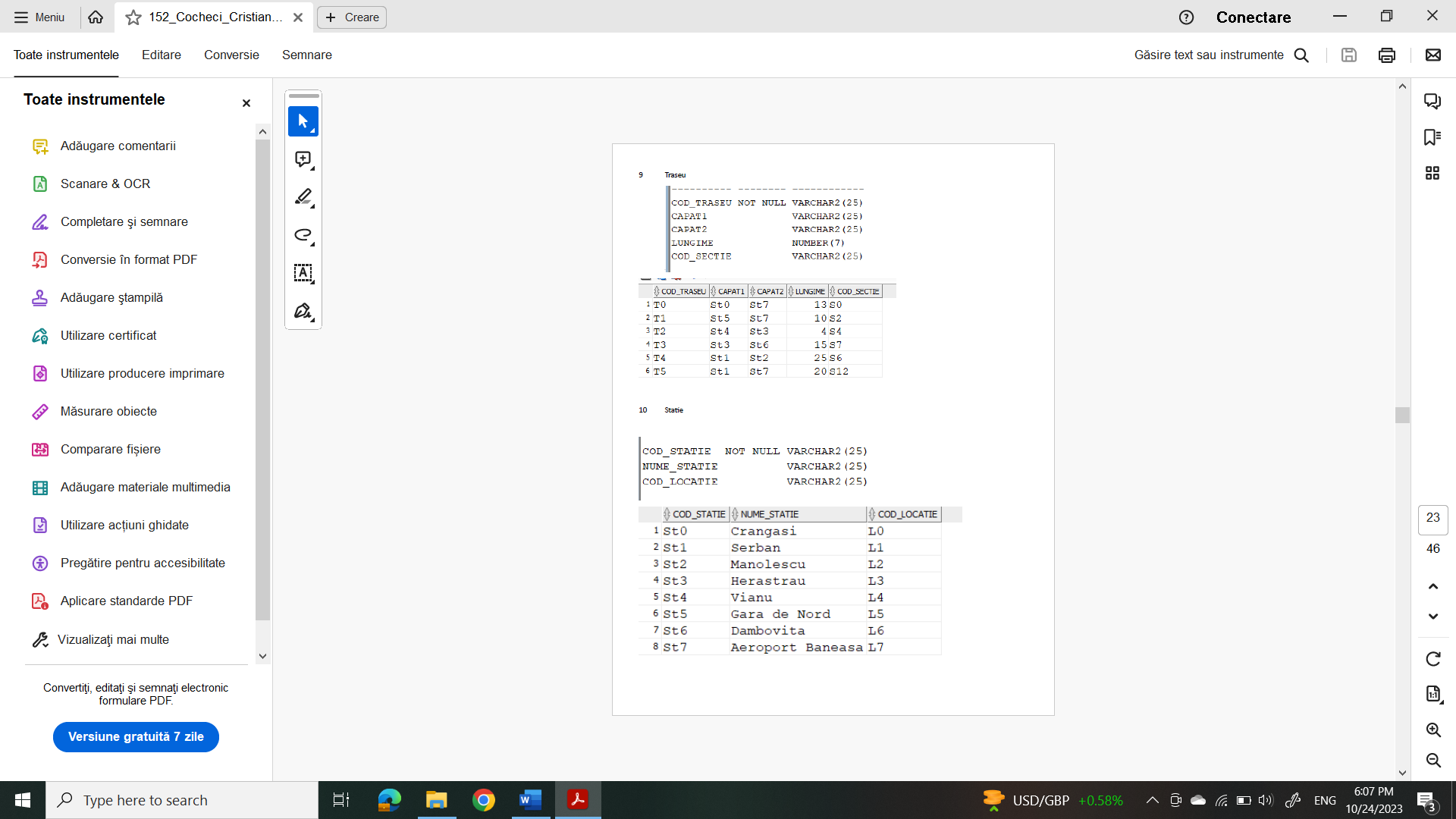
);

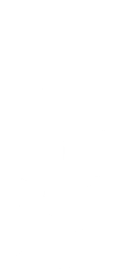
ALTER TABLE trasee\_stc

ADD CONSTRAINT fkey\_capat1 FOREIGN KEY(capat1) REFERENCES statii\_stc(cod\_statie);

ALTER TABLE trasee\_stc

ADD CONSTRAINT fkey\_capat2 FOREIGN KEY(capat2) REFERENCES statii\_stc(cod\_statie);



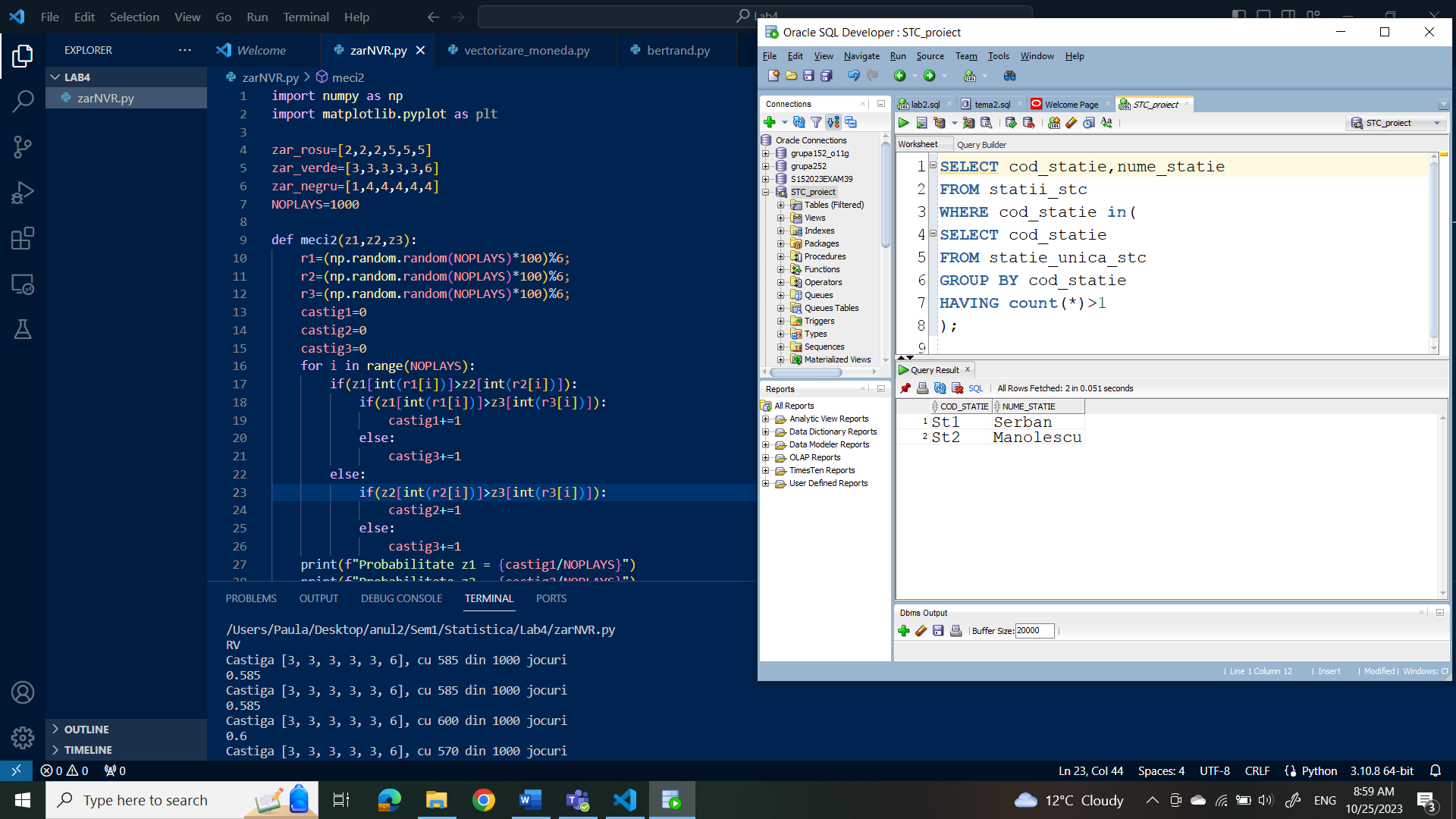


**f. adaptați una dintre cerințele exercițiilor 4-12 pentru diagrama obținută la punctul d**

**(formulați cerința înlimbaj natural, inserați 5-10 înregistrări în fiecare tabelă utilizată,**

**apoi rezolvați cererea propusă în SQL).**

**--Ce statii apartin la mai mult de un traseu?**



SELECT cod\_statie,nume\_statie

FROM statii\_stc

WHERE cod\_statie in(

SELECT cod\_statie

FROM statie\_unica\_stc

GROUP BY cod\_statie

HAVING count(\*)>1

);