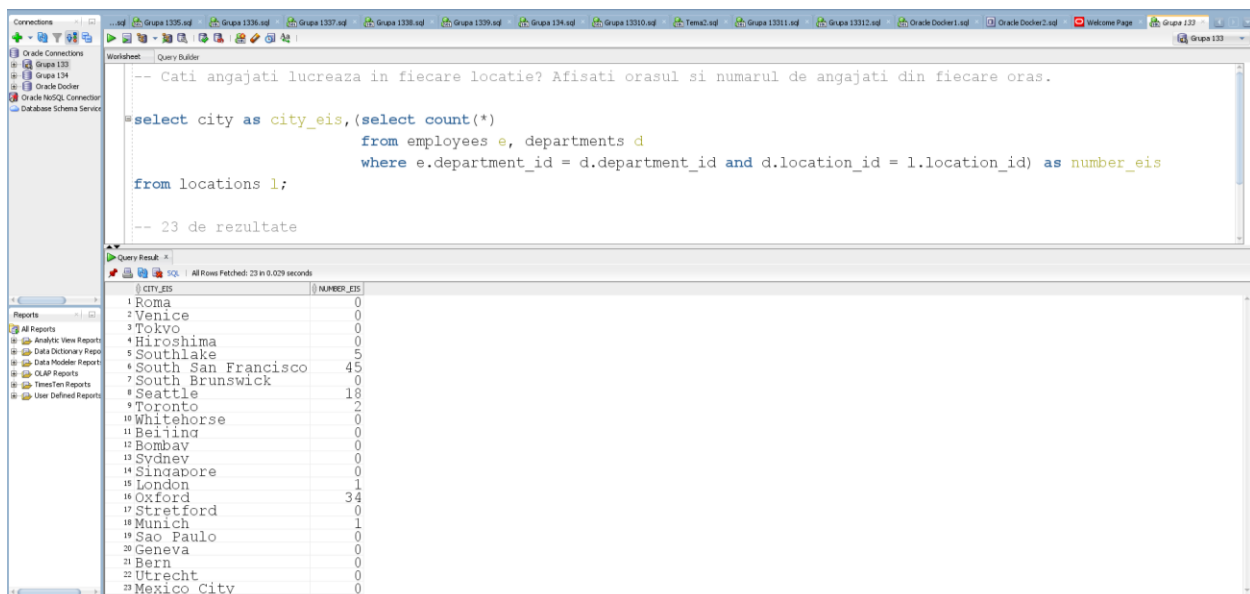


TEMA 3 (LABORATOR BD)

EXERCITIUL 1

```
select city as city_eis,(select count(*)
                        from employees e, departments d
                        where e.department_id = d.department_id and d.location_id = l.location_id) as
number_eis
from locations l;
```

Am folosit o subcerere în clauza select. Pentru fiecare locație din tabelul de locații se selectează prin intermediul departamentelor doar angajații care lucrează în locația respectivă. Liniile generate se numără cu ajutorul funcției count. Fiind corelată pe rând cu fiecare linie din tabelul de locații, cu ajutorul acestei subcereri obținem inclusiv locațiile în care nu lucrează nici un angajat.



The screenshot shows an Oracle SQL Developer interface. The top pane displays a query: `-- Cati angajati lucreaza in fiecare locatie? Afisati orasul si numarul de angajati din fiecare oras.`
`select city as city_eis,(select count(*)`
 `from employees e, departments d`
 `where e.department_id = d.department_id and d.location_id = l.location_id) as number_eis`
`from locations l;`
`-- 23 de rezultate`
The bottom pane shows the query results in a table with two columns: CITY_EIS and NUMBER_EIS. The results are as follows:

CITY_EIS	NUMBER_EIS
1 Roma	0
2 Venice	0
3 Tokyo	0
4 Hiroshima	0
5 Southlake	5
6 South San Francisco	45
7 South Brunswick	0
8 Seattle	18
9 Toronto	2
10 Whitehorse	0
11 Beijing	0
12 Bombay	0
13 Sydney	0
14 Singapore	0
15 London	1
16 Oxford	34
17 Stretford	0
18 Munich	1
19 Sao Paulo	0
20 Geneva	0
21 Bern	0
22 Utrecht	0
23 Mexico City	0

EXERCITIUL 2

```
select l.location_id as location_id_eis, city as city_eis, (select nvl(to_char(max(salary)),
                                                                'Nu exista angajati in aceasta locatie,
                                                                deci nu exista salarii')
from employees e, departments d
where e.department_id = d.department_id and
d.location_id = l.location_id) as max_salary_eis
from locations l;
```

Folosind același raționament ca la exercițiul 1, am folosit funcția max pentru a afla salariul maxim al angajaților care au fost selectați cu ajutorul subcererii. Am folosit NVL deoarece subcererea întorcea

valoarea NULL in momentul in care nu existau angajați într-o locație. Ca ambele valori din funcția NVL să corespundă la tipul de date, am transformat salariul maxim în șir de caractere cu ajutorul TO_CHAR.

-- Pentru fiecare locatie, care este salariul maxim al angajatilor care lucreaza in acele locatii.

```

select l.location_id as location_id_eis, city as city_eis, (select nvl(to_char(max(salary)), 'Nu exista angajati in acest
from employees e, departments d
where e.department_id = d.department_id and d.location_id = l

from locations l;

```

-- 23 de rezultate

LOCATION_ID_EIS	CITY_EIS	MAX_SALARY_EIS
1000	Roma	Nu exista angajati in aceasta locatie. deci nu exista salarii
1100	Venice	Nu exista angajati in aceasta locatie. deci nu exista salarii
1200	Tokyo	Nu exista angajati in aceasta locatie. deci nu exista salarii
1300	Hiroshima	Nu exista angajati in aceasta locatie. deci nu exista salarii
1400	Southlake	9000
1500	South San Francisco	8200
1600	South Brunswick	Nu exista angajati in aceasta locatie. deci nu exista salarii
1700	Seattle	24000
1800	Toronto	13000
1900	Whitehorse	Nu exista angajati in aceasta locatie. deci nu exista salarii
2000	Beijing	Nu exista angajati in aceasta locatie. deci nu exista salarii
2100	Bombay	Nu exista angajati in aceasta locatie. deci nu exista salarii
2200	Sydney	Nu exista angajati in aceasta locatie. deci nu exista salarii
2300	Singapore	Nu exista angajati in aceasta locatie. deci nu exista salarii
2400	London	6500
2500	Oxford	14000
2600	Stretford	Nu exista angajati in aceasta locatie. deci nu exista salarii
2700	Munich	10000
2800	Sao Paulo	Nu exista angajati in aceasta locatie. deci nu exista salarii
2900	Geneva	Nu exista angajati in aceasta locatie. deci nu exista salarii
3000	Bern	Nu exista angajati in aceasta locatie. deci nu exista salarii
3100	Utrecht	Nu exista angajati in aceasta locatie. deci nu exista salarii
3200	Mexico City	Nu exista angajati in aceasta locatie. deci nu exista salarii

EXERCITIUL 3

```

select l.location_id, city, nvl(last_name, 'Nu exista angajat') as name_eis, nvl(to_char(salary), 'Nu exista
salariu') as salary_eis
from locations l, employees ee, departments dd
where ee.department_id(+) = dd.department_id and l.location_id = dd.location_id (+)
and nvl(to_char(salary), 'Nu exista salariu') = (select nvl(to_char(max(salary)), 'Nu exista salariu')
from employees e, departments d
where e.department_id = d.department_id
and d.location_id = l.location_id);

```

Pentru a afișa inclusiv locațiile în care nu se găsesc angajați am folosit join-ul cu (+). Pentru fiecare înregistrare din tabelul de angajați, dacă angajatul lucrează în locația respectivă și salariul său este maxim, îl afișez. În cazul în care nu avem angajați în acea locație, afișez mesaje corespunzătoare cu ajutorul funcției NVL. Salariul maxim a fost calculat ca la exercițiul anterior.

Prima variantă afișează toate locațiile, inclusiv pe cele fără angajați. Următoarea variantă afișează doar locațiile în care se găsesc angajați. Ca la exercițiile anterioare, aceasta folosește subcerere în clauza select pentru a afișa salariul maxim și verifică

```

select l.location_id as location_id_eis, city as city_eis, last_name as last_name_eis,
(select nvl(to_char(max(salary)), 'Nu exista salarii')
from employees e, departments d
where e.department_id = d.department_id and d.location_id = l.location_id) as max_salary_eis
from locations l, employees ee, departments dd
where ee.department_id = dd.department_id and l.location_id = dd.location_id

```

and nvl(to_char(salary), 'Nu exista salarii') = (select nvl(to_char(max(salary)), 'Nu exista salarii'))
 from employees e, departments d
 where e.department_id = d.department_id
 and d.location_id = l.location_id);

Prima varianta

Afișarea tuturor locațiilor din tabel:

Query Builder

```
-- Pentru fiecare locatie, care este numele angajatilor care au salariul maxim si lucreaza in acele locatii.
select l.location_id, city, nvl(last_name, 'Nu exista angajat') as name_eis, nvl(to_char(salary), 'Nu exista salariu') as
from locations l, employees ee, departments dd
where ee.department_id(+) = dd.department_id and l.location_id = dd.location_id (+)
and nvl(to_char(salary), 'Nu exista salariu') = (select nvl(to_char(max(salary)), 'Nu exista salariu')
from employees e, departments d
where e.department_id = d.department_id and d.location_id = l.location_id);

-- 23 de rezultate
```

Script Output: All Rows Fetched: 23 in 0.029 seconds

LOCATION_ID	CITY	NAME_EIS	SALARY_EIS
1700	Seattle	King	24000
1400	Southlake	Hunold	9000
1500	South San Francisco	Fripo	8200
2500	Oxford	Russell	14000
1800	Toronto	Hartstein	13000
2400	London	Mavris	6500
2700	Munich	Baer	10000
3200	Mexico City	Nu exista angajat	Nu exista salariu
1000	Roma	Nu exista angajat	Nu exista salariu
2600	Stretford	Nu exista angajat	Nu exista salariu
1300	Hiroshima	Nu exista angajat	Nu exista salariu
2300	Singapore	Nu exista angajat	Nu exista salariu
1900	Whitehorse	Nu exista angajat	Nu exista salariu
1100	Venice	Nu exista angajat	Nu exista salariu
2800	Sao Paulo	Nu exista angajat	Nu exista salariu
2900	Geneva	Nu exista angajat	Nu exista salariu
2100	Bombay	Nu exista angajat	Nu exista salariu
1600	South Brunswick	Nu exista angajat	Nu exista salariu
3100	Utrecht	Nu exista angajat	Nu exista salariu
2200	Sydney	Nu exista angajat	Nu exista salariu

A doua varianta

Afișarea locațiilor în care se găsesc angajați:

Query Builder

```
-- Pentru fiecare locatie, care este numele angajatilor care au salariul maxim si lucreaza in acele locatii.
select l.location_id as location_id_eis, city as city_eis, last_name as last_name_eis,
(select nvl(to_char(max(salary)), 'Nu exista salarii'))
from employees e, departments d
where e.department_id = d.department_id and d.location_id = l.location_id) as max_salary_eis
from locations l, employees ee, departments dd
where ee.department_id = dd.department_id and l.location_id = dd.location_id
and nvl(to_char(salary), 'Nu exista salarii') = (select nvl(to_char(max(salary)), 'Nu exista salarii')
from employees e, departments d
where e.department_id = d.department_id and d.location_id = l.location_id);

-- 7 rezultate
```

Script Output: All Rows Fetched: 7 in 0.024 seconds

LOCATION_ID_EIS	CITY_EIS	LAST_NAME_EIS	MAX_SALARY_EIS
1700	Seattle	King	24000
1400	Southlake	Hunold	9000
1500	South San Francisco	Fripo	8200
2500	Oxford	Russell	14000
1800	Toronto	Hartstein	13000
2400	London	Mavris	6500
2700	Munich	Baer	10000

EXERCITIUL 4

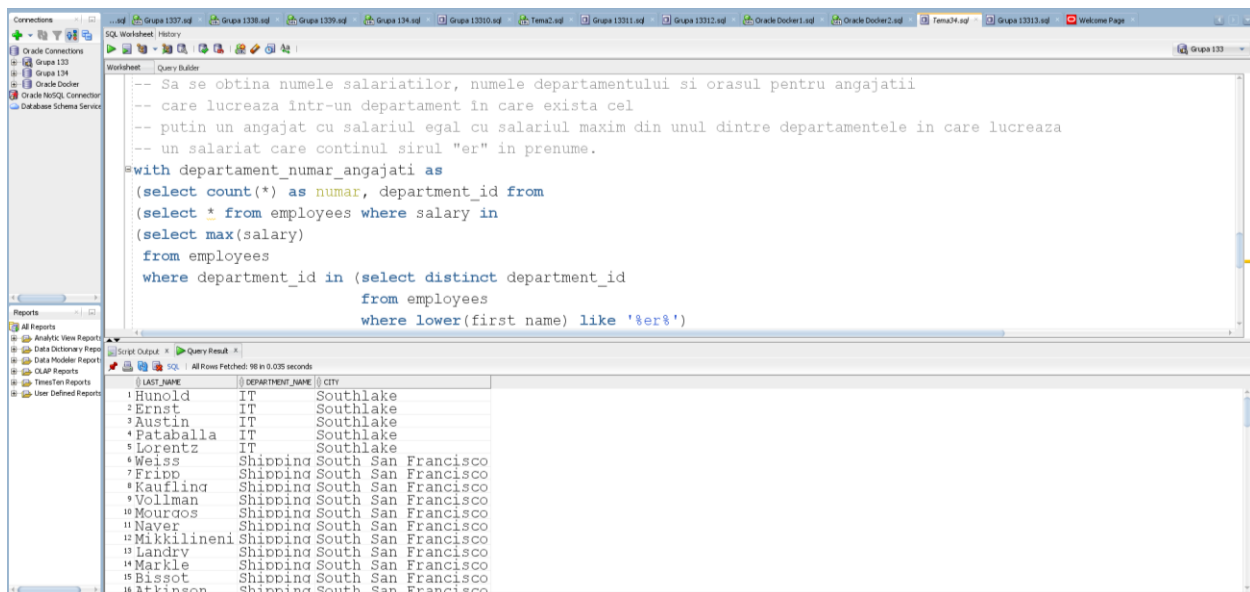
```
with departament_numar_angajati as
(select count(*) as numar, department_id
from (select *
      from employees
      where salary in (select max(salary)
                      from employees
                      where department_id in (select distinct department_id
                                             from employees
                                             where lower(first_name) like '%er%')
      group by department_id))
group by department_id)
select last_name as last_name_eis, department_name as department_name_eis, city as city_eis
from employees e, departments d, locations l, departament_numar_angajati aux
where e.department_id = d.department_id and d.location_id = l.location_id
and e.department_id = aux.department_id and aux.numar >= 1;
```

Cu ajutorul lui WITH am creat un tabel auxiliar in care am reținut numărul de angajați din departamentul corespunzător.

Crearea tabelului DEPARTAMENT_NUMAR_ANGAJATI:

Inițial am selectat angajații cu "er" in nume, apoi maximul salarial pe fiecare departament in care se afla aceștia cu ajutorul GROUP BY. Ulterior am selectat angajații care au salariul egal cu unul din salariile obținute mai devreme. În cele din urmă i-am numărat și am afișat departamentul în care lucrează.

Tabelul auxiliar fiind creat, am selectat datele cerute făcând JOIN între cele patru tabele. Ultimele două condiții asigură faptul că se selectează din tabelul auxiliar departamentul în care lucrează angajatul, obținându-se numărul de angajați corespunzător acestuia. Numărul trebuie să fie mai mare sau egal cu unu.



The screenshot shows the SQL Developer interface. The top pane displays a query in the Query Builder. The query uses a WITH clause to create a subquery named 'departament_numar_angajati' which counts employees by department where the salary is equal to the maximum salary of employees whose first name contains 'er'. The main query then joins this subquery with the employees, departments, and locations tables to find employees whose department has at least one such high-salary employee.

The bottom pane shows the query results in a table with columns LAST_NAME, DEPARTMENT_NAME, and CITY. The results list 16 employees, all of whom are in the 'Shipping' department and located in 'South San Francisco'.

LAST_NAME	DEPARTMENT_NAME	CITY
Hunold	IT	Southlake
Ernst	IT	Southlake
Austin	IT	Southlake
Pataballa	IT	Southlake
Lorentz	IT	Southlake
Weiss	Shipping	South San Francisco
Frippe	Shipping	South San Francisco
Kaufling	Shipping	South San Francisco
Vollman	Shipping	South San Francisco
Mourgos	Shipping	South San Francisco
Nayer	Shipping	South San Francisco
Mikkilineni	Shipping	South San Francisco
Landry	Shipping	South San Francisco
Markle	Shipping	South San Francisco
Bissot	Shipping	South San Francisco
Atkinson	Shipping	South San Francisco

The screenshot shows the SQL Developer interface with a query in the main window. The query is as follows:

```

from employees
where department_id in (select distinct department_id
                        from employees
                        where lower(first_name) like '%er%')
group by department_id)
group by department_id)
select last_name as last_name_eis, department_name as departmet_name_eis, city as city_eis
from employees e, departments d, locations l, departament_numar_angajati aux
where e.department_id = d.department_id and d.location_id = l.location_id
and e.department_id = aux.department_id and aux.numar >= 1;
-- 98 de rezultate

```

The Results pane at the bottom shows the following data:

LAST_NAME_EIS	DEPARTMENT_NAME_EIS	CITY_EIS
Atkinson	Shipping	South San Francisco
Marlow	Shipping	South San Francisco
Olson	Shipping	South San Francisco
Mallin	Shipping	South San Francisco
Rogers	Shipping	South San Francisco
Gee	Shipping	South San Francisco
Philtanker	Shipping	South San Francisco
Ladwig	Shipping	South San Francisco
Stiles	Shipping	South San Francisco
Seo	Shipping	South San Francisco
Patel	Shipping	South San Francisco
Rais	Shipping	South San Francisco
Davies	Shipping	South San Francisco
Matos	Shipping	South San Francisco
Vargas	Shipping	South San Francisco
Taylor	Shipping	South San Francisco

EXERCITIUL 4

- a. select distinct department_id
from employees
where lower(first_name) like '%er%' and department_id is not null;

The screenshot shows the SQL Developer interface with a query in the main window. The query is as follows:

```

-- In ce departamente lucreaza salariatii care continul sirul "er" in prenume?
select distinct department_id
from employees
where lower(first_name) like '%er%' and department_id is not null;
-- 6 rezultate

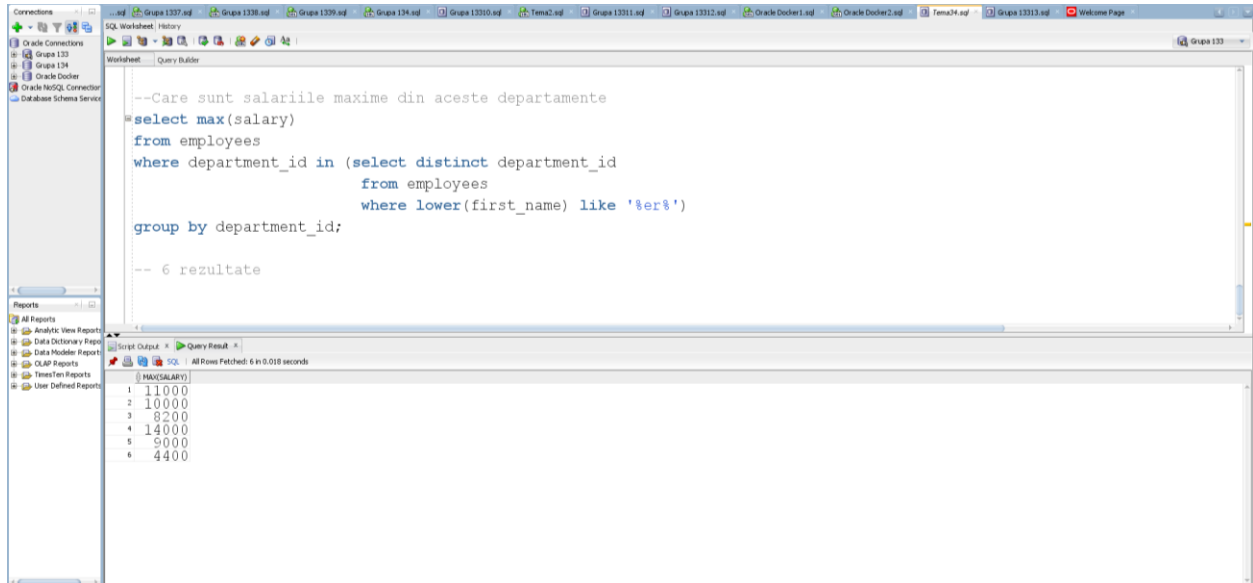
```

The Results pane at the bottom shows the following data:

DEPARTMENT_ID
30
70
50
80
60
10

- b. select max(salary)
from employees
where department_id in (select distinct department_id
from employees
where lower(first_name) like '%er%')

group by department_id;



The screenshot shows the SQL Developer interface with a query window. The query is as follows:

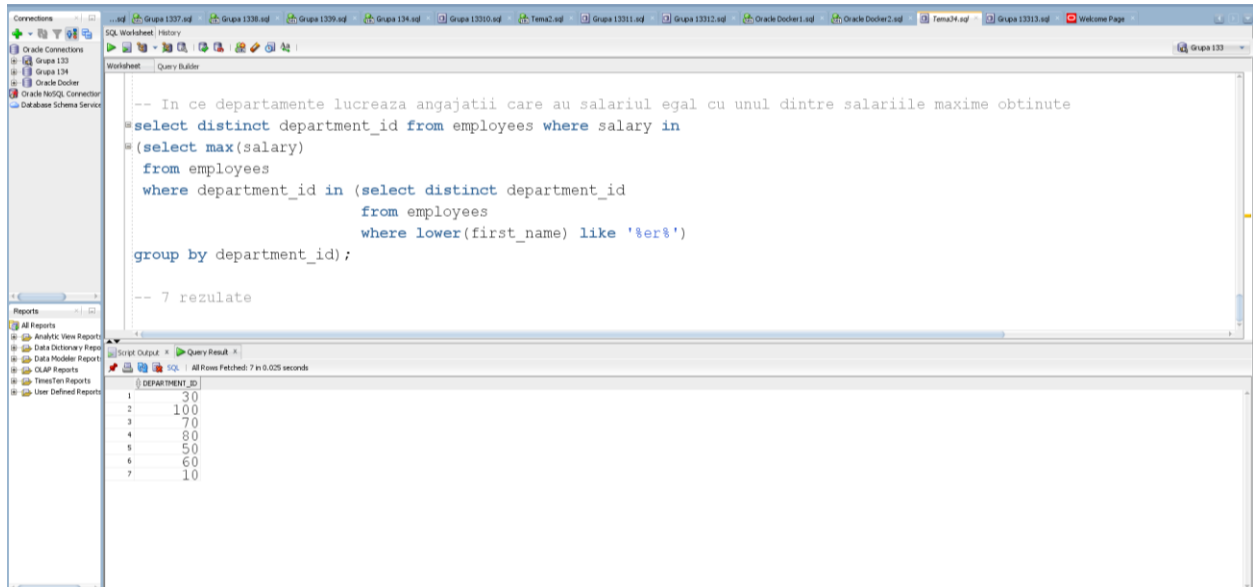
```
--Care sunt salariile maxime din aceste departamente
select max(salary)
from employees
where department_id in (select distinct department_id
                        from employees
                        where lower(first_name) like '%er%')
group by department_id;

-- 6 rezultate
```

The results pane shows the following data:

MAXSALARY
1 11000
2 10000
3 8200
4 14000
5 9000
6 4400

- c. select distinct department_id from employees where salary in
(select max(salary)
from employees
where department_id in (select distinct department_id
from employees
where lower(first_name) like '%er%')
group by department_id);



The screenshot shows the SQL Developer interface with a query window. The query is as follows:

```
-- In ce departamente lucreaza angajatii care au salariul egal cu unul dintre salariile maxime obtinute
select distinct department_id from employees where salary in
(select max(salary)
from employees
where department_id in (select distinct department_id
                        from employees
                        where lower(first_name) like '%er%')
group by department_id);

-- 7 rezultate
```

The results pane shows the following data:

DEPARTMENT_ID
1 30
2 100
3 70
4 80
5 50
6 60
7 10