EXERCITIILE 11-16

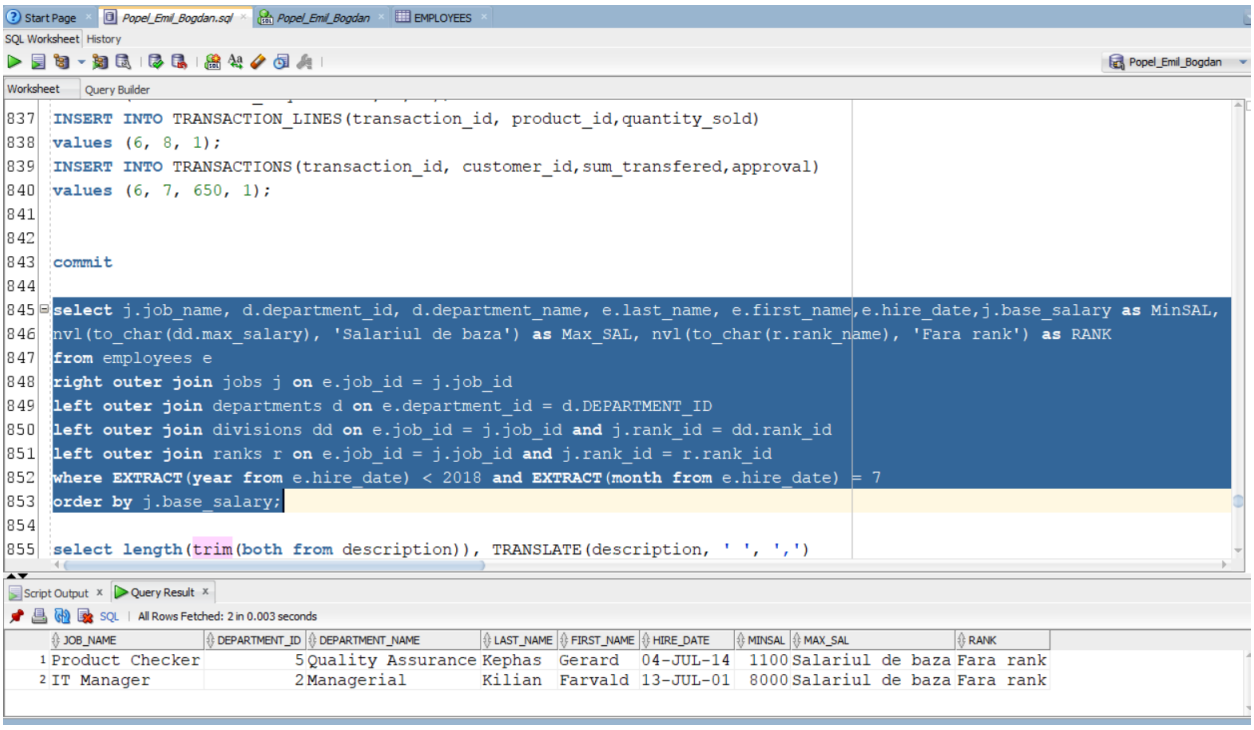
--Numele Jobului, id-ul si numele departamentului, salariul de baza si datele angajatilor care au fost angajati inainte de 2018, dar in luna iulie. Se vor afisa de asemenea joburile care nu au angajati si cele care nu fac parte dintr-un departament.

--left outer join -> toate elementele din employees, inclusiv null

--right outer join [tabel]-> toate elementele din [tabel], inclusiv valoarea null.

--nvl(val1, val2) -> verifica daca val1 = null. Daca nu este, returneaza val1. Daca val1 = null, va returna val2.

```
select j.job_name, d.department_id, d.department_name, e.last_name,
e.first_name,e.hire_date,j.base_salary as MinSAL,

nvl(to_char(dd.max_salary), 'Salariul de baza') as Max_SAL, nvl(to_char(r.rank_name), 'Fara rank') as
RANK

from employees e

right outer join jobs j on e.job_id = j.job_id

left outer join departments d on e.department_id = d.DEPARTMENT_ID

left outer join divisions dd on e.job_id = j.job_id and j.rank_id = dd.rank_id

left outer join ranks r on e.job_id = j.job_id and j.rank_id = r.rank_id

where EXTRACT(year from e.hire_date) < 2018 and EXTRACT(month from e.hire_date) = 7

order by j.base_salary;
```



```
INSERT INTO TRANSACTION_LINES(transaction_id, product_id,quantity_sold)
values (6, 8, 1);
INSERT INTO TRANSACTIONS(transaction_id, customer_id,sum_transfered,approval)
values (6, 7, 650, 1);


commit

select j.job_name, d.department_id, d.department_name, e.last_name, e.first_name,e.hire_date,j.base_salary as MinSAL,
nvl(to_char(dd.max_salary), 'Salariul de baza') as Max_SAL, nvl(to_char(r.rank_name), 'Fara rank') as RANK
from employees e
right outer join jobs j on e.job_id = j.job_id
left outer join departments d on e.department_id = d.DEPARTMENT_ID
left outer join divisions dd on e.job_id = j.job_id and j.rank_id = dd.rank_id
left outer join ranks r on e.job_id = j.job_id and j.rank_id = r.rank_id
where EXTRACT(year from e.hire_date) < 2018 and EXTRACT(month from e.hire_date) = 7
order by j.base_salary;

select length(trim(both from description)), TRANSLATE(description, ' ', ',')
```

Script Output | Query Result — All Rows Fetched: 2 in 0.003 seconds

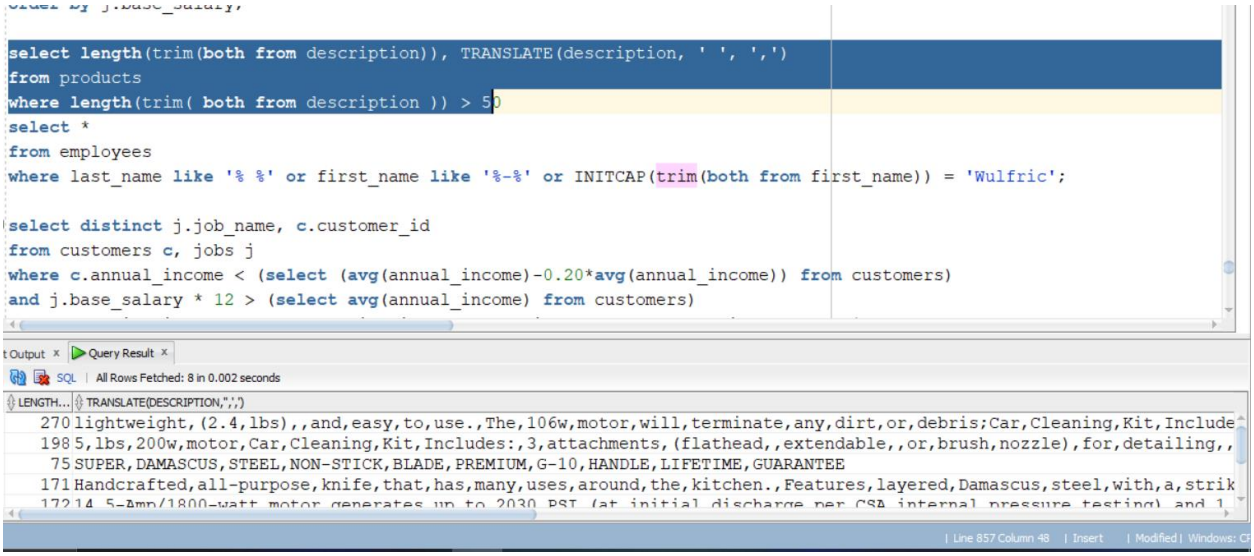
	JOB_NAME	DEPARTMENT_ID	DEPARTMENT_NAME	LAST_NAME	FIRST_NAME	HIRE_DATE	MINSAL	MAX_SAL	RANK
1	Product Checker	5	Quality Assurance	Kephas	Gerard	04-JUL-14	1100	Salariul de baza	Fara rank
2	IT Manager	2	Managerial	Kilian	Farvald	13-JUL-01	8000	Salariul de baza	Fara rank

--urmatoarea cereri scot in evidenta functii sql pe siruri de caractere. Prima modifica descrierile produselor mai lungi de 50 de caractere. Spatiile se vor transforma in virgule. Se vor afisa lungimile si descrierile dupa modificare.

select length(trim(both from description)), TRANSLATE(description, ' ', ',')

from products

where length(trim(both from description)) > 50

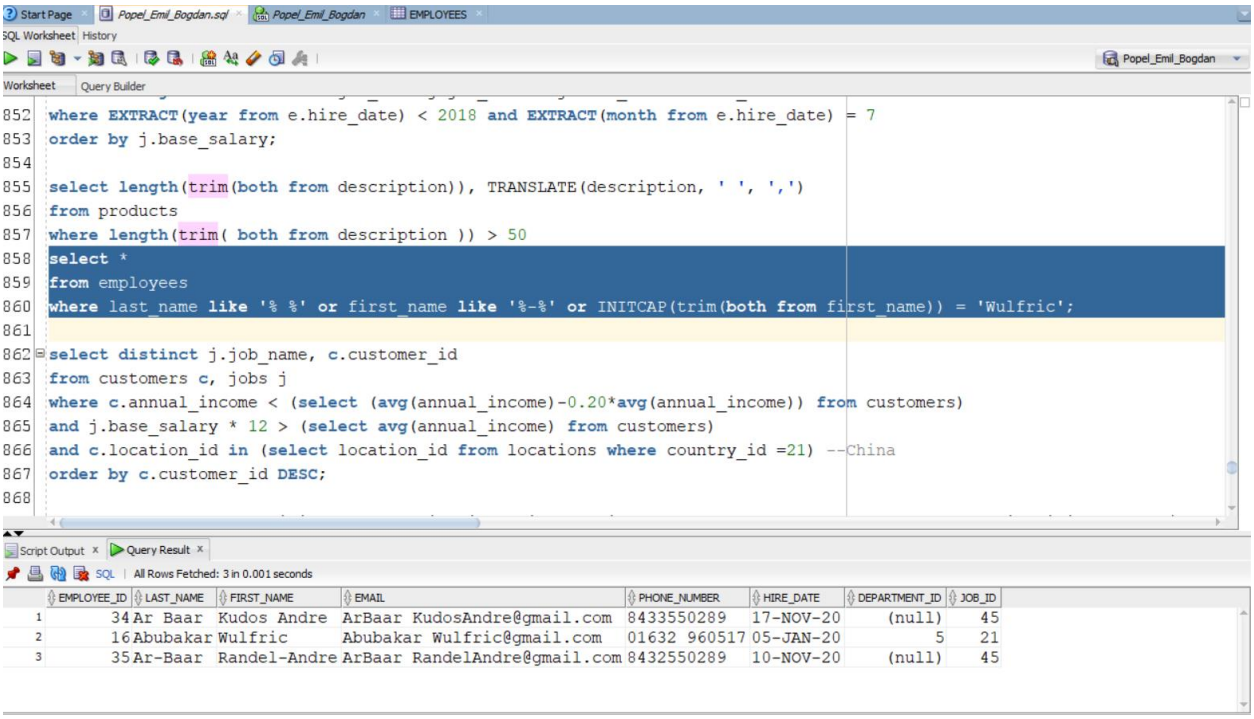


```
select length(trim(both from description)), TRANSLATE(description, ' ', ',')
from products
where length(trim( both from description )) > 50
select *
from employees
where last_name like '% %' or first_name like '%-%' or INITCAP(trim(both from first_name)) = 'Wulfric';

select distinct j.job_name, c.customer_id
from customers c, jobs j
where c.annual_income < (select (avg(annual_income)-0.20*avg(annual_income)) from customers)
and j.base_salary * 12 > (select avg(annual_income) from customers)
```

Query Result — All Rows Fetched: 8 in 0.002 seconds

	LENGTH...	TRANSLATE(DESCRIPTION,",",")
	270	lightweight,(2.4,lbs),,and,easy,to,use.,The,106w,motor,will,terminate,any,dirt,or,debris;Car,Cleaning,Kit,Include
	198	5,lbs,200w,motor,Car,Cleaning,Kit,Includes:,3,attachments,(flathead,,extendable,,or,brush,nozzle),for,detailing,,
	75	SUPER,DAMASCUS,STEEL,NON-STICK,BLADE,PREMIUM,G-10,HANDLE,LIFETIME,GUARANTEE
	171	Handcrafted,all-purpose,knife,that,has,many,uses,around,the,kitchen.,Features,layered,Damascus,steel,with,a,strik
	172	14.5-Amp/1800-watt motor generates up to 2030 PSI (at initial discharge per CSA internal pressure testing) and 1

```
select *
from employees
where last_name like '% %' or first_name like '%-%' or INITCAP(trim(both from first_name)) = 'Wulfric';
```



Cererea urmatoare afiseaza joburile care annual au un salariu mai mare decat un client considerat “Low-Income / Low-Medium Income” din China, dar si id-urile clientiilor respectivi.

```
select distinct j.job_name
from customers c, jobs j
where c.annual_income < (select (avg(annual_income)-0.20*avg(annual_income)) from customers)
and j.base_salary * 12 > (select avg(annual_income) -0.20*avg(annual_income) from customers)
and c.location_id in (select location_id from locations where country_id =21)
```

```
861
862 select distinct j.job_name, c.customer_id
863 from customers c, jobs j
864 where c.annual_income < (select (avg(annual_income)-0.20*avg(annual_income)) from customers)
865 and j.base_salary * 12 > (select avg(annual_income) from customers)
866 and c.location_id in (select location_id from locations where country_id =21) --China
867 order by c.customer_id DESC;
868
869 select distinct salaryLimit.val as "Clienti de nivel 1 si 2", AverageYearlySalary.vall as "Salariu Minim Anual c/dh",
870 SalMaxAnual.valll as "medie salariu maxim anual c/dh"
871 from  (select avg(annual_income)-0.20*avg(annual_income) val from customers) salaryLimit,
```

Script Output | Query Result | All Rows Fetched: 14 in 0.003 seconds

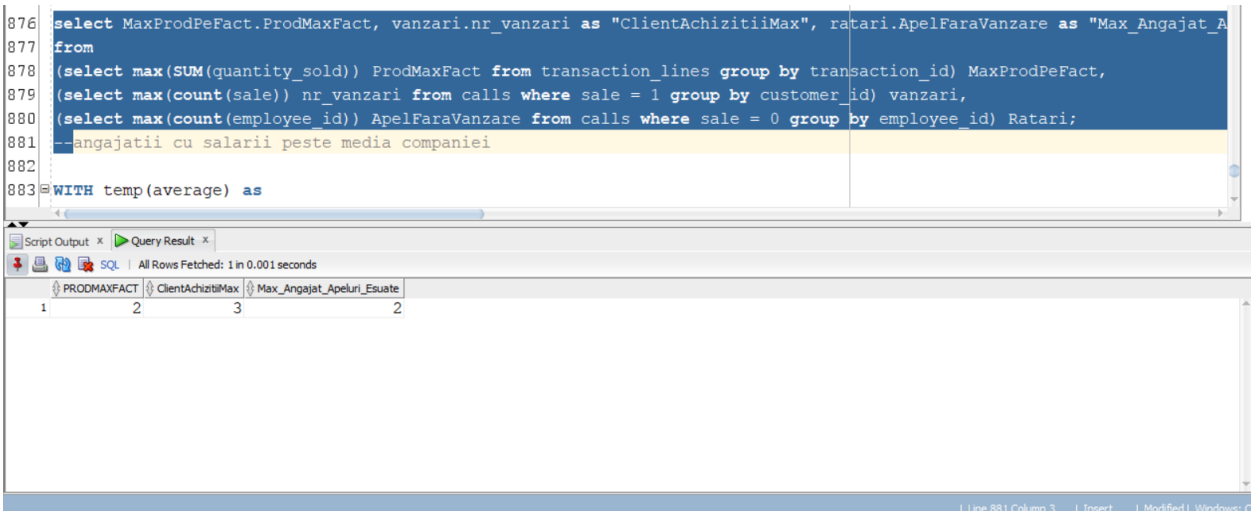
	JOB_NAME	CUSTOMER_ID
1	Bureaucracy Manager	8
2	Callers Manager	8
3	Data-Providers Manager	8
4	IT Manager	8
5	Managerial Manager	8
6	Quality Assurance Manager	8
7	Training Manager	8
8	Bureaucracy Manager	7
9	Callers Manager	7
10	Data-Providers Manager	7

Line 867 Column 30 | Insert | Modified | Windows: CF

--Urmatoarea cerere genereaza date statistice despre firma precum:

- Numarul de produse maxim din cadrul unei facturi
- Numarul de tranzactii maxim realizate de un client
- Numarul maxim de apeluri fara a face o vanzare din partea unui angajat.

```
select MaxProdPeFact.ProdMaxFact, vanzari.nr_vanzari as "ClientAchizitiiMax", ratari.ApelFaraVanzare as "Max_Angajat_Apeluri_Esuate"

from

(select max(SUM(quantity_sold)) ProdMaxFact from transaction_lines group by transaction_id) MaxProdPeFact,

(select max(count(sale)) nr_vanzari from calls where sale = 1 group by customer_id) vanzari,

(select max(count(employee_id)) ApelFaraVanzare from calls where sale = 0 group by employee_id) Ratari;
```



--angajatii cu salarii peste media companiei:

```
WITH temp(average) as
   (SELECT avg(base_salary)
   from jobs)
      SELECT job_name, base_salary
      FROM jobs, temp
      WHERE jobs.base_salary > temp.average;
```

```
WITH temp(average) as
    (SELECT avg(base_salary)
    from jobs)
        SELECT job_name, base_salary
        FROM jobs, temp
        WHERE jobs.base_salary > temp.average;

select e.last_name as Nume, j.job_name as Job, d.department_name, j.base
DECODE(e.department_id, 3, j.base_salary*1.15, 2,
j.base_salary*1.27, 5, j.base_salary*1.10, j.base_salary)as Salariu_Mar
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 7 in 0.002 seconds

	JOB_NAME	BASE_SALARY
1	Managerial Manager	10000
2	IT Manager	8000
3	Training Manager	8000
4	Quality Assurance Manager	8000
5	Callers Manager	8000
6	Data-Providers Manager	8000
7	Bureaucracy Manager	8000

-- cresc salariile de baza ale angajatilor din departamentele 2, 3 si 5

--nu sunt afisati angajatii fara departament

select e.last_name as Nume, j.job_name as Job, d.department_name, j.base_salary as BAZA,

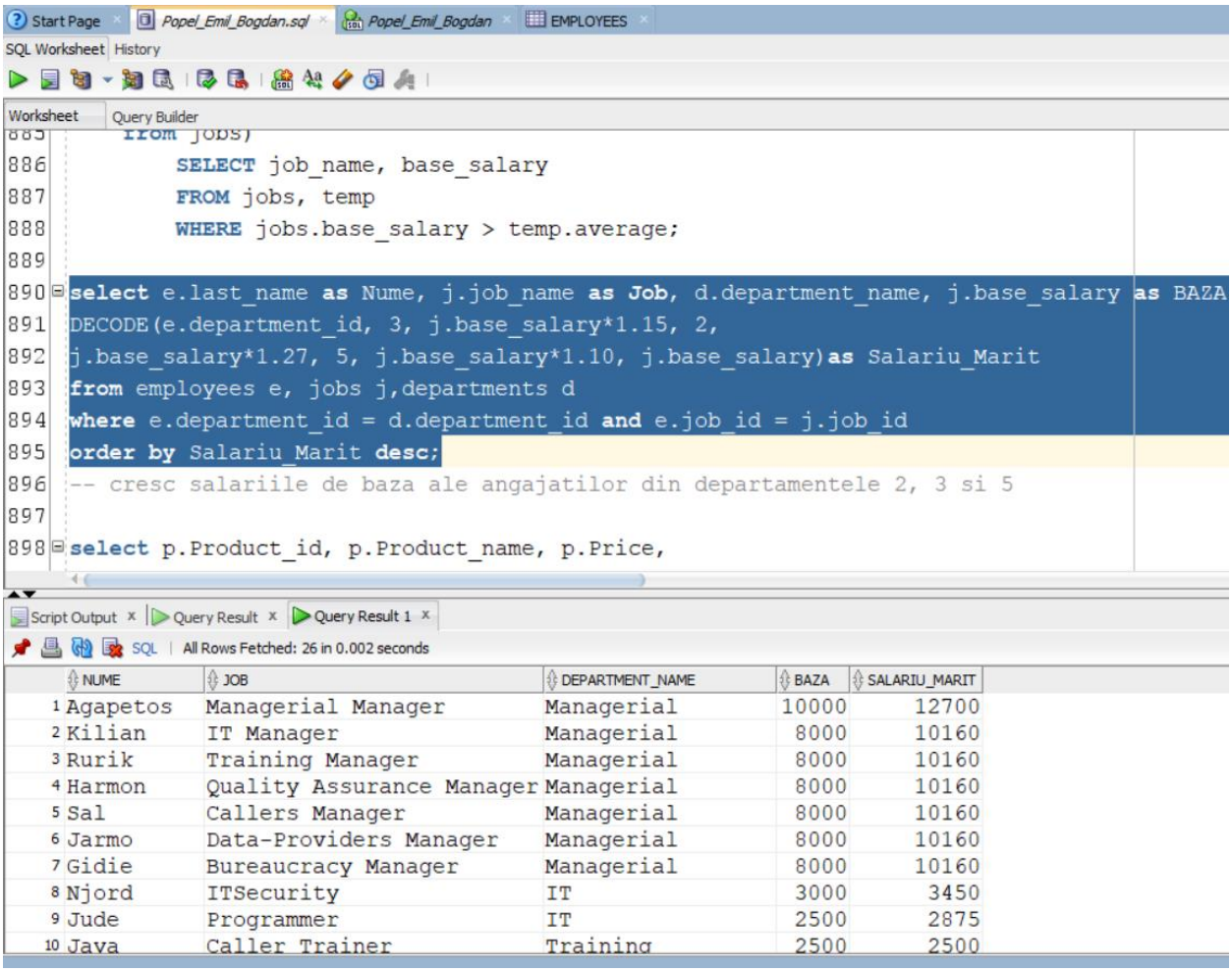
DECODE(e.department_id, 3, j.base_salary*1.15, 2,

j.base_salary*1.27, 5, j.base_salary*1.10, j.base_salary)as Salariu_Marit

from employees e, jobs j,departments d

where e.department_id = d.department_id and e.job_id = j.job_id

order by Salariu_Marit desc;



-- cresc salariile de baza ale angajatilor din departamentele 2, 3 si 5

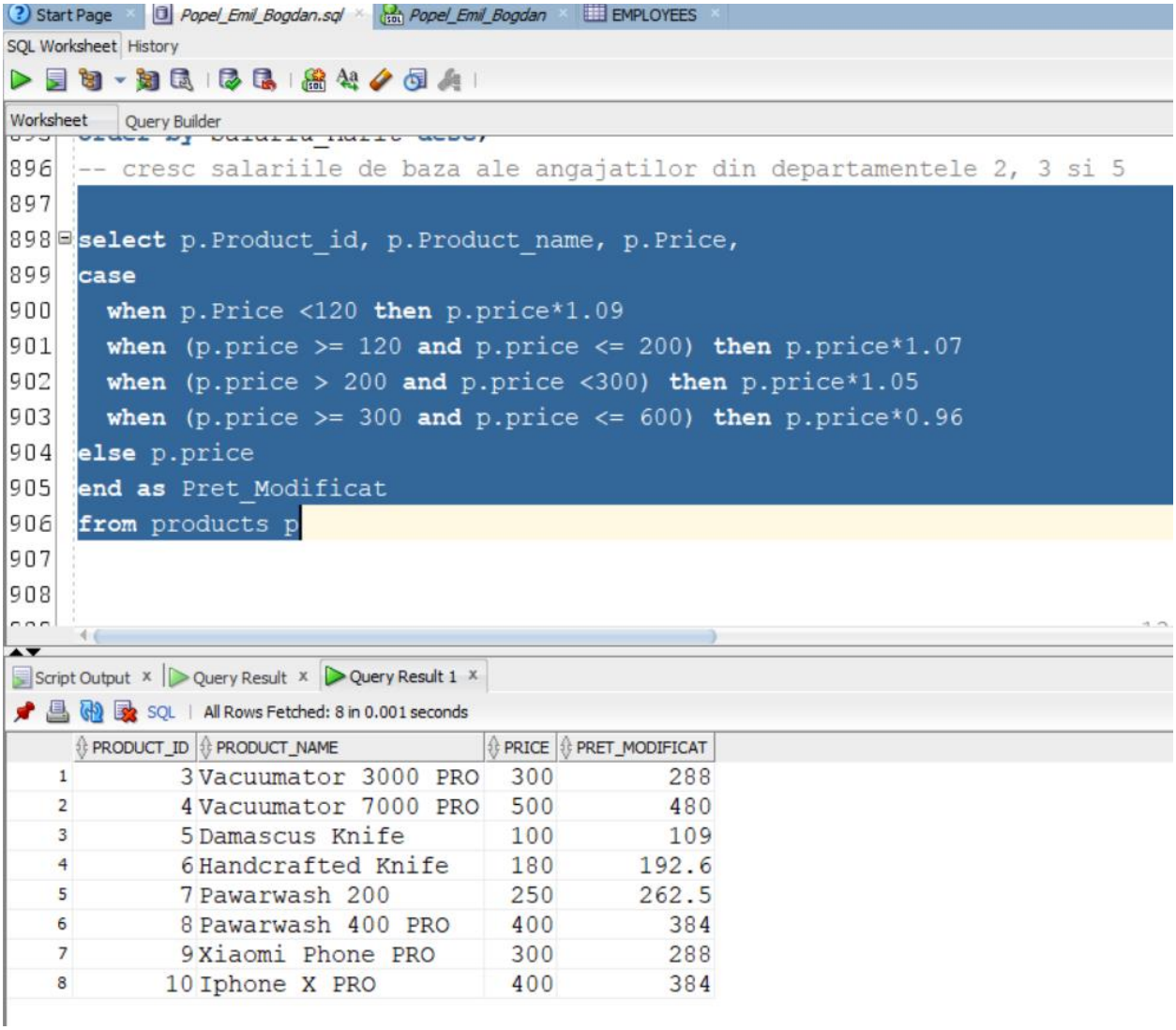
--modific preturile produselor in functie de preturile anterioare.

Case ia la rand cazurile de when, iar cand conditia este indeplinita, se returneaza valoarea din then. Daca un caz favorabil nu este intalnit, se preia valoarea din else. La final, coloana se va numi “Pret_modificat”.

```
select p.Product_id, p.Product_name, p.Price,
case
  when p.Price <120 then p.price*1.09
  when (p.price >= 120 and p.price <= 200) then p.price*1.07
  when (p.price > 200 and p.price <300) then p.price*1.05
  when (p.price >= 300 and p.price <= 600) then p.price*0.96
else p.price
```

end as Pret_Modificat

from products p



----------13

--Modific numele produselor cu pretul mai mare de 300. Li se adauga cuvantul “PRO” in nume.

update products

set product_name = CONCAT(product_name, ' PRO')

where products.price in (select price from products where price >= 300);

```
update products
set product_name = CONCAT(product_name, ' PRO')
where products.price in (select price from products where price >= 300);

update employees
set email = replace(email, '@CallMeBack', '@gmail.com')
where employees.job_id in (select job_id from jobs where employees.job_id = jobs.job_id
and lower(trim(both from jobs.job_name)) != 'caller');

update employees
set phone_number = replace(phone_number, '+', '')
where employees.phone_number in (select phone_number from employees where trim(both from employees.phone_number) like
```

Script Output | Query Result | Query Result 1

Task completed in 0.001 seconds

```
5 rows updated.
```

--Toti angajatii care nu sunt “Calleri”, vor aveam adresa de email cu sufixul @gmail.com, in loc de “@CallMeBack.com”.

update employees

set email = replace(email, '@CallMeBack', '@gmail.com')

where employees.job_id in (select job_id from jobs where employees.job_id = jobs.job_id

and lower(trim(both from jobs.job_name)) != 'caller');

```
update employees
set email = replace(email, '@CallMeBack', '@gmail.com')
where employees.job_id in (select job_id from jobs where employees.job_id = jobs.job_id
and lower(trim(both from jobs.job_name)) != 'caller');

update employees
set phone_number = replace(phone_number, '+', '')
where employees.phone_number in (select phone_number from employees where trim(both from emp
```

Script Output | Query Result | Query Result 1

Task completed in 0.001 seconds

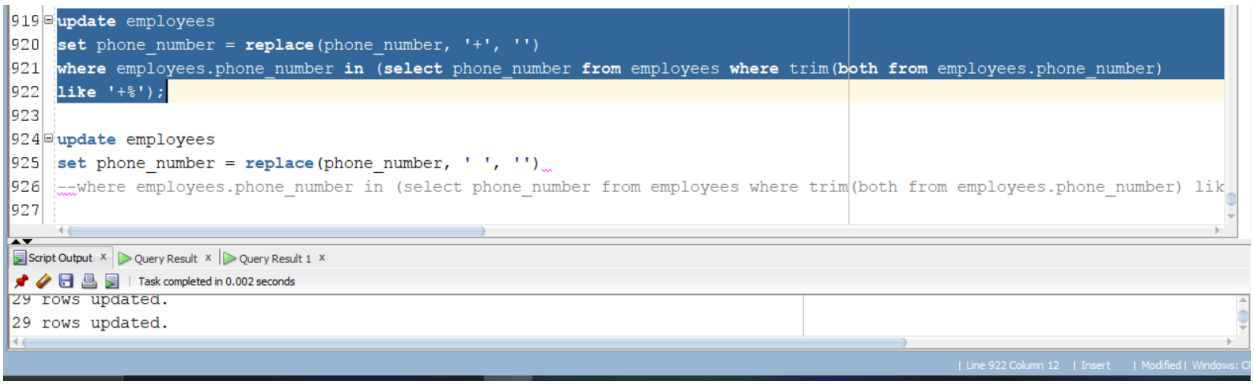
```
29 rows updated.
```

--toate numerele de telefon care incep cu ‘+’ vor fi modificate, pentru a nu mai avea + in fata, ci direct prefixul.

update employees

set phone_number = replace(phone_number, '+', '')

where employees.phone_number in (select phone_number from employees where trim(both from employees.phone_number) like '+%')



--toate locatiile, clientii, furnizorii si departamentele bazei noastre de date. Daca una din acestea nu se afla in locatia respectiva, valoarea ramane null. Se afiseaza de asemena tara din care locatia face parte.

select l.location_id as LOCATII, c.location_id as Clienti, s.location_id as Furnizori, d.location_id as Departament,

cc.country_name as TARA

from locations l

FULL outer join customers c on l.location_id = c.location_id

full outer join suppliers s on l.location_id = s.location_id

full outer join departments d on l.location_id = d.location_id

full outer join countries cc on l.country_id = cc.country_id

----------16

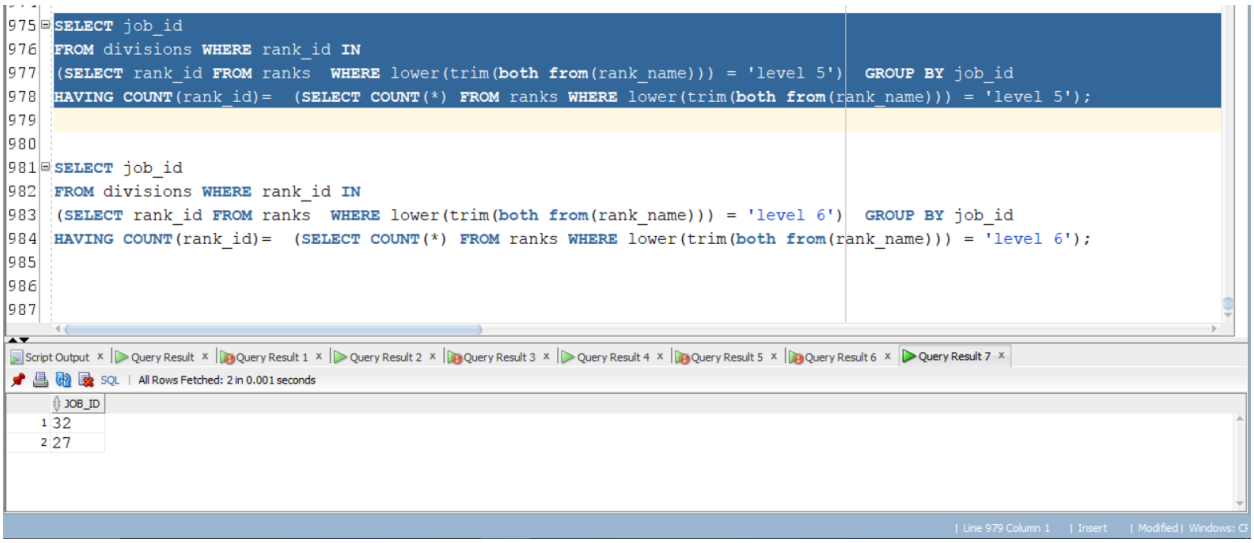
Cererile urmatoare furnizeaza codurile joburilor care au rank (level 5 sau level 6), conform modelului din laborator. Modelul de date nu imi permite sa fac ceva mai complex, nu am datele sau structura la dispozitie.

SELECT job_id

FROM divisions WHERE rank_id IN

(SELECT rank_id FROM ranks WHERE lower(trim(both from(rank_name))) = 'level 5') GROUP BY job_id

HAVING COUNT(rank_id)= (SELECT COUNT(*) FROM ranks WHERE lower(trim(both from(rank_name))) = 'level 5');



SELECT job_id

FROM divisions WHERE rank_id IN

(SELECT rank_id FROM ranks WHERE lower(trim(both from(rank_name))) = 'level 6') GROUP BY job_id

HAVING COUNT(rank_id)= (SELECT COUNT(*) FROM ranks WHERE lower(trim(both from(rank_name))) = 'level 6');