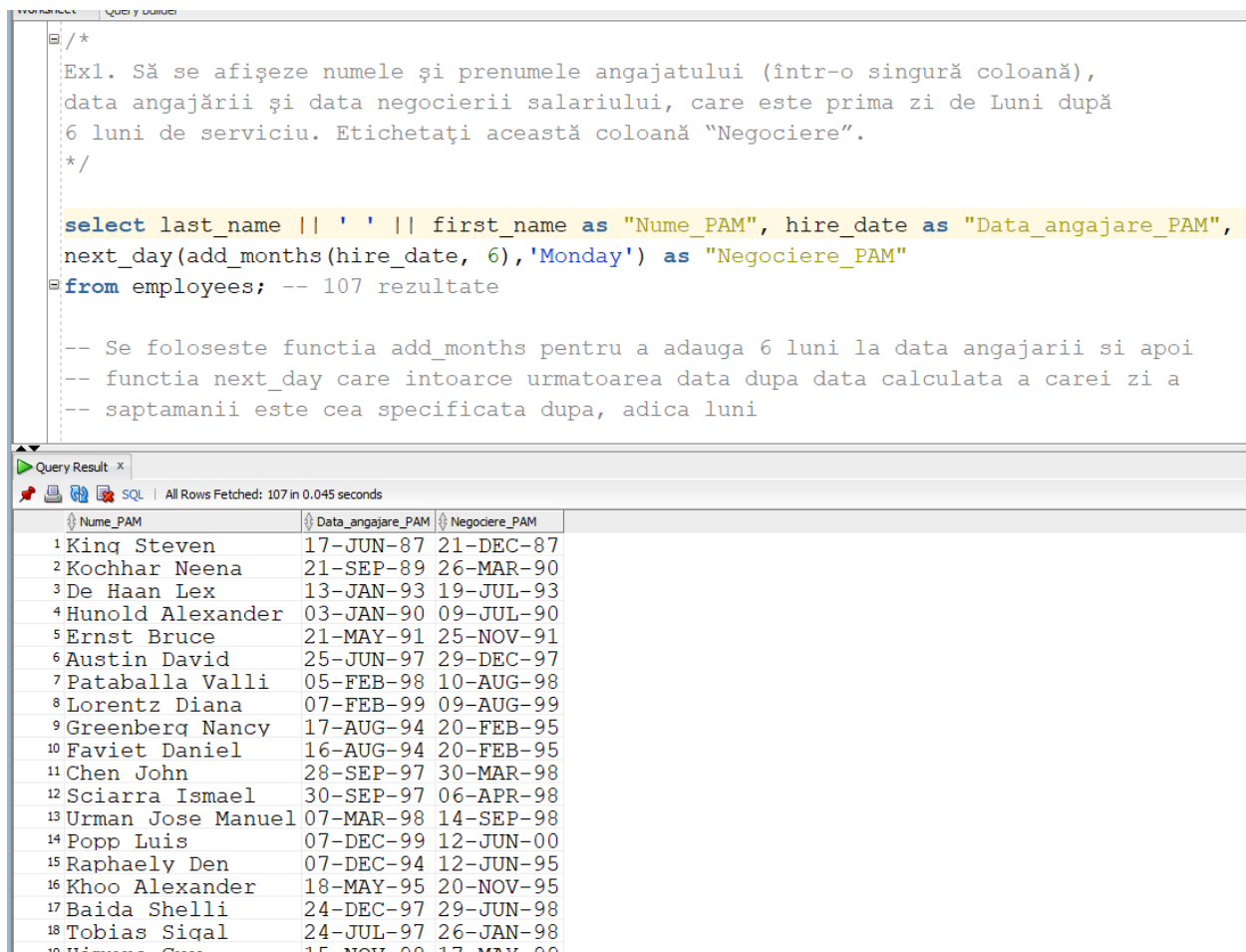


Exercitiul 1

```
select last_name || ' ' || first_name as "Nume_PAM", hire_date as "Data_angajare_PAM",  
next_day(add_months(hire_date, 6), 'Monday') as "Negociere_PAM"  
from employees; -- 107 rezultate
```

-- Se foloseste functia add_months pentru a adauga 6 luni la data angajarii si apoi
-- functia next_day care intoarce urmatoarea data dupa data calculata a carei zi a
-- saptamanii este cea specificata dupa, adica luni



The screenshot shows a SQL IDE with a query editor and a results pane. The query editor contains a SQL query with comments in Romanian. The results pane shows the output of the query, which is a table with three columns: Nume_PAM, Data_angajare_PAM, and Negociere_PAM. The results are sorted by the first column.

```
/*  
Ex1. Să se afișeze numele și prenumele angajatului (într-o singură coloană),  
data angajării și data negocierii salariului, care este prima zi de Luni după  
6 luni de serviciu. Etichetați această coloană "Negociere".  
*/  
  
select last_name || ' ' || first_name as "Nume_PAM", hire_date as "Data_angajare_PAM",  
next_day(add_months(hire_date, 6), 'Monday') as "Negociere_PAM"  
from employees; -- 107 rezultate  
  
-- Se foloseste functia add_months pentru a adauga 6 luni la data angajarii si apoi  
-- functia next_day care intoarce urmatoarea data dupa data calculata a carei zi a  
-- saptamanii este cea specificata dupa, adica luni
```

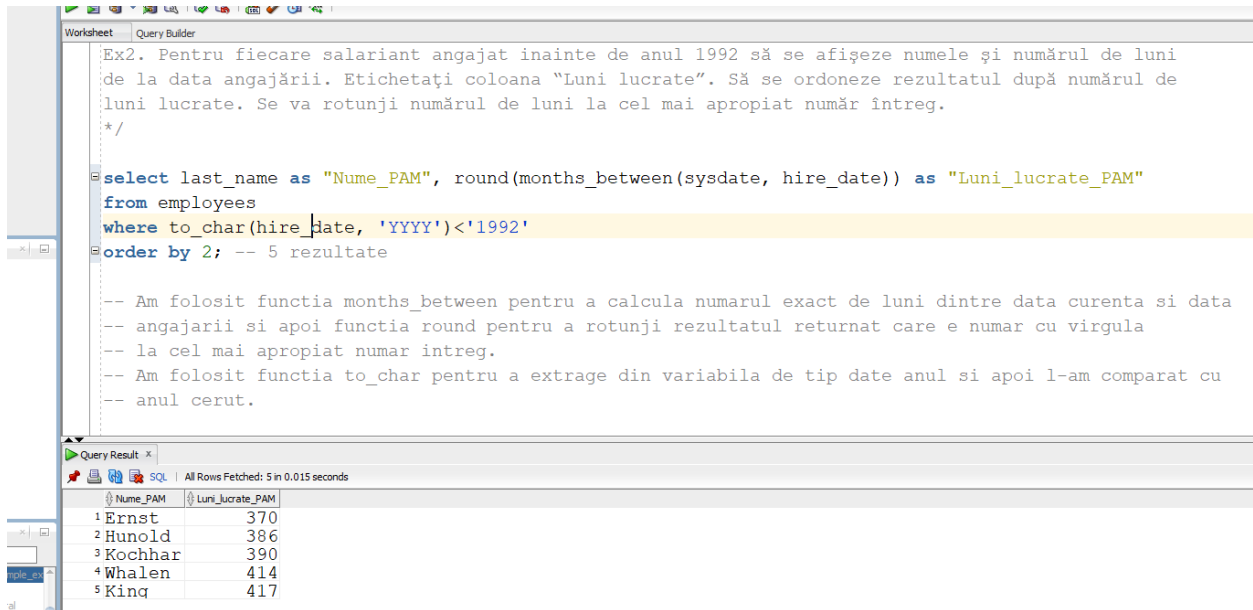
Nume_PAM	Data_angajare_PAM	Negociere_PAM
King Steven	17-JUN-87	21-DEC-87
Kochhar Neena	21-SEP-89	26-MAR-90
De Haan Lex	13-JAN-93	19-JUL-93
Hunold Alexander	03-JAN-90	09-JUL-90
Ernst Bruce	21-MAY-91	25-NOV-91
Austin David	25-JUN-97	29-DEC-97
Pataballa Valli	05-FEB-98	10-AUG-98
Lorentz Diana	07-FEB-99	09-AUG-99
Greenberg Nancy	17-AUG-94	20-FEB-95
Faviet Daniel	16-AUG-94	20-FEB-95
Chen John	28-SEP-97	30-MAR-98
Sciarra Ismael	30-SEP-97	06-APR-98
Urman Jose Manuel	07-MAR-98	14-SEP-98
Popp Luis	07-DEC-99	12-JUN-00
Raphaely Den	07-DEC-94	12-JUN-95
Khoo Alexander	18-MAY-95	20-NOV-95
Baida Shelli	24-DEC-97	29-JUN-98
Tobias Sigal	24-JUL-97	26-JAN-98
Higgins Guy	15-NOV-98	17-MAY-99

Exercitiul 2

```
select last_name as "Nume_PAM", round(months_between(sysdate, hire_date)) as "Luni_lucrate_PAM"
from employees
where to_char(hire_date, 'YYYY') < '1992'
order by 2; -- 5 rezultate
```

-- Am folosit functia months_between pentru a calcula numarul exact de luni dintre data curenta si data angajarii si apoi functia round pentru a rotunji rezultatul returnat care e numar cu virgula la cel mai apropiat numar intreg.

-- Am folosit functia to_char pentru a extrage din variabila de tip date anul si apoi l-am comparat cu anul cerut.



The screenshot shows the SQL Developer interface. The top pane displays a query with a comment in Romanian explaining the task: to list employees hired before 1992, showing their last name and the number of months between their hire date and the current date, rounded to the nearest integer. The query is as follows:

```
select last_name as "Nume_PAM", round(months_between(sysdate, hire_date)) as "Luni_lucrate_PAM"
from employees
where to_char(hire_date, 'YYYY') < '1992'
order by 2; -- 5 rezultate
```

Below the query, there are three lines of comments explaining the use of the `months_between`, `round`, and `to_char` functions.

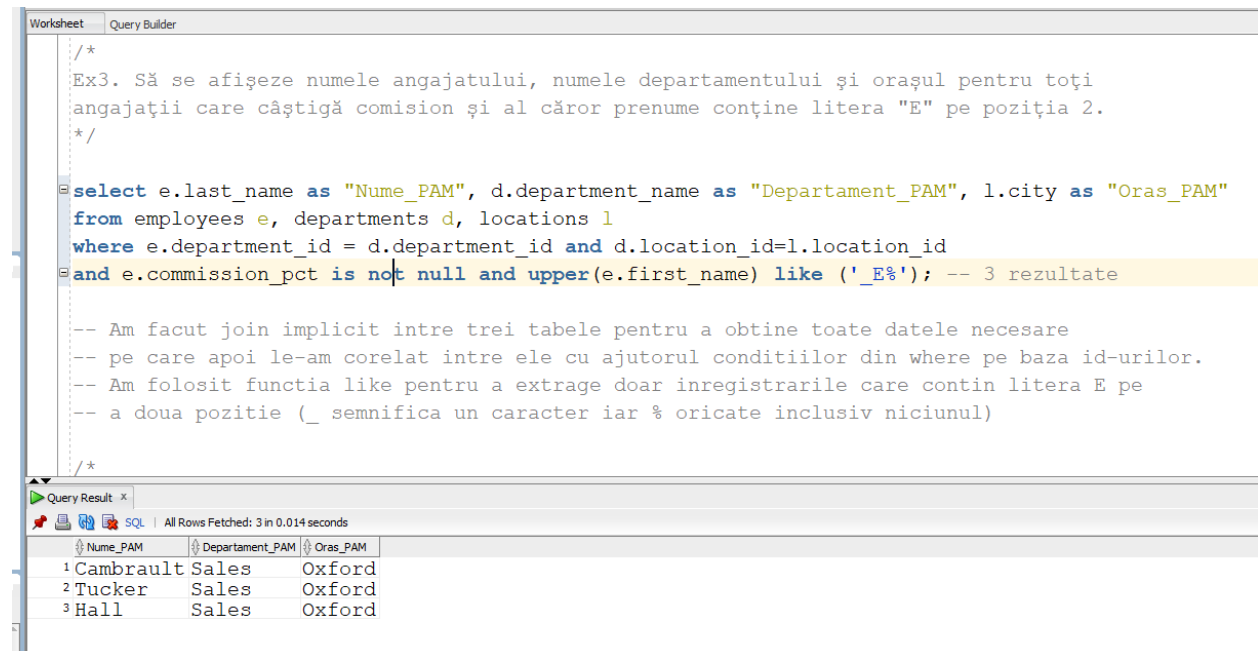
The bottom pane shows the query results in a table with two columns: `Nume_PAM` and `Luni_lucrate_PAM`. The results are ordered by the second column.

	Nume_PAM	Luni_lucrate_PAM
1	Ernst	370
2	Hunold	386
3	Kochhar	390
4	Whalen	414
5	King	417

Exercitiul 3

```
select e.last_name as "Nume_PAM", d.department_name as "Departament_PAM", l.city as "Oras_PAM"
from employees e, departments d, locations l
where e.department_id = d.department_id and d.location_id=l.location_id
and e.commission_pct is not null and upper(e.first_name) like ('_E%'); -- 3 rezultate
```

-- Am facut join implicit intre trei tabele pentru a obtine toate datele necesare
-- pe care apoi le-am corelat intre ele cu ajutorul conditiilor din where pe baza id-urilor.
-- Am folosit functia like pentru a extrage doar inregistrarile care contin litera E pe
-- a doua pozitie (_ semnifica un caracter iar % oricate inclusiv niciunul)
-- Am verificat existenta unui comision prin compararea campului cu null.



```
/*
Ex3. Să se afișeze numele angajatului, numele departamentului și orașul pentru toți
angajații care câștigă comision și al căror prenume conține litera "E" pe poziția 2.
*/

select e.last_name as "Nume_PAM", d.department_name as "Departament_PAM", l.city as "Oras_PAM"
from employees e, departments d, locations l
where e.department_id = d.department_id and d.location_id=l.location_id
and e.commission_pct is not null and upper(e.first_name) like ('_E%'); -- 3 rezultate

-- Am facut join implicit intre trei tabele pentru a obtine toate datele necesare
-- pe care apoi le-am corelat intre ele cu ajutorul conditiilor din where pe baza id-urilor.
-- Am folosit functia like pentru a extrage doar inregistrarile care contin litera E pe
-- a doua pozitie (_ semnifica un caracter iar % oricate inclusiv niciunul)

/*
```

Query Result: x

All Rows Fetched: 3 in 0.014 seconds

	Nume_PAM	Departament_PAM	Oras_PAM
1	Cambrault	Sales	Oxford
2	Tucker	Sales	Oxford
3	Hall	Sales	Oxford

Exercitiul 4

```
select e.last_name as "Nume_PAM", d.department_name as "Departament_PAM", j.job_title as  
"Nume_job_PAM"  
from employees e, departments d, jobs j  
where e.department_id=d.department_id and e.job_id=j.job_id and upper(e.last_name) like ('%A%')  
and lower(j.job_title) like ('a%t'); -- 6 rezultate
```

-- Am facut join implicit intre trei tabele pentru a obtine toate datele necesare
-- pe care apoi le-am corelat intre ele cu ajutorul conditiilor din where pe baza id-urilor
-- si apoi am extras numai cele care indeplineau conditiile din cerinta.
-- Am folosit functia like pentru a extrage doar inregistrarile careau litera A inclusa în nume
-- si titlul job-ului incepe cu litera "a" și se termină cu litera "t" (% semnifica oricate caractere inclusiv
-- niciunul)

```
/*  
Ex4. Să se afișeze numele salariatului, numele departamentului si titlul job-ului pentru toți salariații  
care au litera A inclusă în nume si titlul job-ului incepe cu litera "a" și se termină cu litera "t".  
*/  
  
select e.last_name as "Nume_PAM", d.department_name as "Departament_PAM", j.job_title as "Nume_job_PAM"  
from employees e, departments d, jobs j  
where e.department_id=d.department_id and e.job_id=j.job_id and upper(e.last_name) like ('%A%') and  
lower(j.job_title) like ('a%t'); -- 6 rezultate  
  
-- Am facut join implicit intre trei tabele pentru a obtine toate datele necesare  
-- pe care apoi le-am corelat intre ele cu ajutorul conditiilor din where pe baza id-urilor  
-- si apoi am extras numai cele care indeplineau conditiile din cerinta.  
-- Am folosit functia like pentru a extrage doar inregistrarile careau litera A inclusa în nume  
-- si titlul job-ului incepe cu litera "a" și se termină cu litera "t" (% semnifica oricate caractere inclusiv niciunul)
```

Query Result X

All Rows Fetched: 6 in 0.015 seconds

Nume_PAM	Departament_PAM	Nume_job_PAM
Whalen	Administration	Administration Assistant
De Haan	Executive	Administration Vice President
Kochhar	Executive	Administration Vice President
Sciarra	Finance	Accountant
Faviet	Finance	Accountant
Urman	Finance	Accountant

Exercitiul 5

- Varianta 1 - Decode

```
select e.last_name as "Nume_PAM", e.hire_date as "Data_angajarii_PAM", j.job_title as  
"Nume_job_PAM", e.salary as "Salariu_PAM", e.salary + decode(to_char(e.hire_date, 'YYYY'), '1989',  
e.salary*0.2, '1990', e.salary*0.15, '1991', e.salary * 0.1, 0) as "Salariu_marire_PAM"  
from employees e, jobs j  
where e.job_id = j.job_id; -- 107 rezultate
```

-- Am folosit functia decode pentru a returna diverse valori in functie de valoarea pe care
-- o are anul angajarii. Primul argument este comparat cu al doilea si in caz de egalitate se
-- returneaza a treilea argument altfel se compara cu primul cu al patrulea si in caz
-- de egalitate se returneaza al cincilea altfel se continua comparatia si daca nu e egal cu nici una din
-- variantele date va returna ultimul parametru adica 0.
-- Am facut join implicit intre tabela employees si jobs punand conditia ca job id din
-- cele doua tabele sa fie egale.

```
/*  
Ex5. Să se afișeze numele, data angajării, titlul job-ului, salariul și o coloană reprezentând salariul  
după ce se aplică o mărire, astfel: pentru salariații angajați în 1989 creșterea este de 20%, pentru cei  
angajați în 1990 creșterea este de 15%, iar salariul celor angajați în anul 1991 crește cu 10%.  
Pentru salariații angajați în alți ani valoarea nu se modifică. (2 rezolvări: DECODE și CASE).  
*/  
  
select e.last_name as "Nume_PAM", e.hire_date as "Data_angajarii_PAM", j.job_title as "Nume_job_PAM",  
e.salary as "Salariu_PAM", e.salary + decode(to_char(e.hire_date, 'YYYY'), '1989', e.salary*0.2, '1990', e.salary*0.15, '1991',  
e.salary * 0.1, 0) as "Salariu_marire_PAM"  
from employees e, jobs j  
where e.job_id = j.job_id; -- 107 rezultate  
  
-- Am folosit functia decode pentru a returna diverse valori in functie de valoarea pe care  
-- o are anul angajarii. Primul argument este comparat cu al doilea si in caz de egalitate se  
-- returneaza a treilea argument altfel se compara cu primul cu al patrulea si in caz  
-- de egalitate se returneaza al cincilea altfel se continua comparatia si daca nu e egal cu nici una din variantele date  
-- va returna ultimul parametru adica 0.  
-- Am facut join implicit intre tabela employees si jobs punand conditia ca job id din  
-- cele doua tabele sa fie egale.
```

Query Result x

All Rows Fetched: 107 in 0.046 seconds

Nume_PAM	Data_angajarii_PAM	Nume_job_PAM	Salariu_PAM	Salariu_marire_PAM
1 Gietz	07-JUN-94	Public Accountant	8300	8300
2 Higgins	07-JUN-94	Accounting Manager	12000	12000
3 Whalen	17-SEP-84	Administration Assistant	4400	4400
4 King	17-JUN-87	President	24000	24000
5 De Haan	13-JAN-93	Administration Vice President	17000	17000
6 Kochhar	21-SEP-89	Administration Vice President	17000	20400
7 Chen	28-SEP-97	Accountant	8200	8200
8 Faviet	16-AUG-94	Accountant	9000	9000
9 Popp	07-DEC-99	Accountant	6900	6900

- Varianta 2 - Case

```
select e.last_name as "Nume_PAM", e.hire_date as "Data_angajarii_PAM", j.job_title as
"Nume_job_PAM",
e.salary as "Salariu_PAM", e.salary +
case to_char(e.hire_date, 'YYYY')
when '1989' then e.salary*0.2
when '1990' then e.salary*0.15
when '1991' then e.salary*0.1
else 0
end as "Salariu_marire_PAM"
from employees e, jobs j
where e.job_id = j.job_id; -- 107 rezultate
```

-- Am folosit clauza case pentru a returna diverse valori in functie de valoarea pe care
 -- o are anul angajarii. Argumentul din case adica anul angajarii este comparat
 -- cu valorile din when si in caz de egalitate se returneaza valoarea de dupa then
 -- si in caz ca nu se indeplineste aceasta conditie pentru niciun when se returneaza valoarea de
 -- pe else adica 0.
 -- Am facut join implicit intre tabela employees si jobs punand conditia ca job id din
 -- cele doua tabele sa fie egale.

```
select e.last_name as "Nume_PAM", e.hire_date as "Data_angajarii_PAM", j.job_title as "Nume_job_PAM",
e.salary as "Salariu_PAM", e.salary +
case to_char(e.hire_date, 'YYYY')
when '1989' then e.salary*0.2
when '1990' then e.salary*0.15
when '1991' then e.salary*0.1
else 0
end as "Salariu_marire_PAM"
from employees e, jobs j
where e.job_id = j.job_id; -- 107 rezultate
```

-- Am folosit clauza case pentru a returna diverse valori in functie de valoarea pe care
 -- o are anul angajarii. Argumentul din case adica anul angajarii este comparat
 -- cu valorile din when si in caz de egalitate se returneaza valoarea de dupa then
 -- si in caz ca nu se indeplineste aceasta conditie pentru niciun when se returneaza valoarea de pe
 -- else adica 0.
 -- Am facut join implicit intre tabela employees si jobs punand conditia ca job id din
 -- cele doua tabele sa fie egale.

	Nume_PAM	Data_angajarii_PAM	Nume_job_PAM	Salariu_PAM	Salariu_marire_PAM
1	Gietz	07-JUN-94	Public Accountant	8300	8300
2	Higgins	07-JUN-94	Accounting Manager	12000	12000
3	Whalen	17-SEP-87	Administration Assistant	4400	4400
4	King	17-JUN-87	President	24000	24000
5	De Haan	13-JAN-93	Administration Vice President	17000	17000
6	Kochhar	21-SEP-89	Administration Vice President	17000	20400
7	Chen	28-SEP-97	Accountant	8200	8200
8	Faviet	16-AUG-94	Accountant	9000	9000
9	Popp	07-DEC-99	Accountant	6900	6900
10	Sciarra	30-SEP-97	Accountant	7700	7700
11	Urman	07-MAR-98	Accountant	7800	7800