

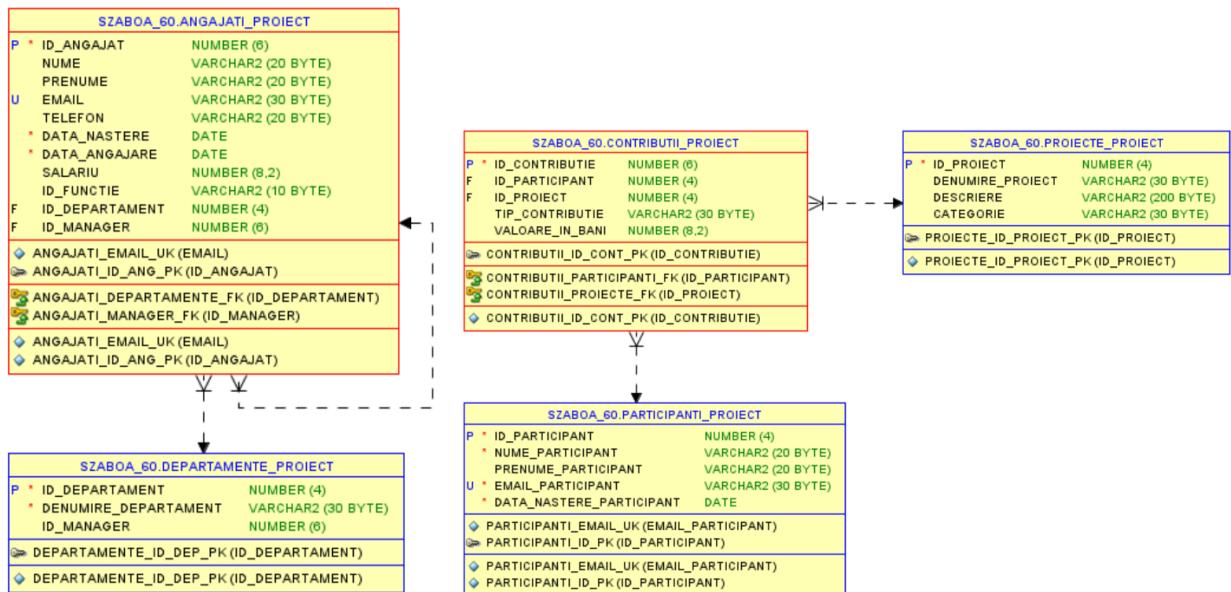
## Proiect SGBD

### Descrierea problemei cu caracter economic

In contextul afacerilor moderne, organizatiile isi desfasoara activitatea intr-un mediu dinamic si competitiv, ceea ce le determina sa ia decizii strategice informate, bazate pe date. In acest sens, bazele de date devin un instrument esential pentru colectarea, stocarea si analiza informatiilor de care au nevoie managerii si angajatii. Cu toate acestea, exista o problema comună cu care se confrunta majoritatea companiilor: gestionarea proiectelor. Astfel, este necesara o solutie integrata pentru a urmari, monitoriza si gestiona proiectele in mod eficient si eficace.

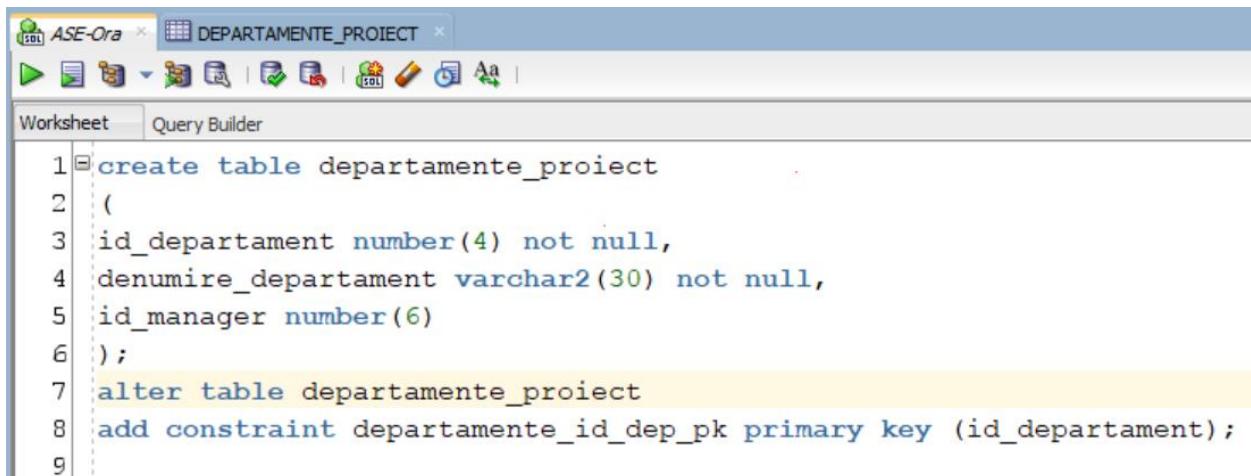
Pentru a raspunde acestei probleme, am creat o baza de date PL/SQL care include tabelele DEPARTAMENTE, ANGAJATI, PARTICIPANTI, CONTRIBUTII si PROIECTE. Scopul acestui proiect este de a oferi o solutie robusta si scalabila pentru gestionarea proiectelor, care sa ajute organizatiile sa isi optimizeze procesele interne si sa isi maximizeze performanta.

### Schema conceptuala a tabelelor



### Crearea tabelelor cu restrictiile de integritate aferente

```
create table departamente_proiect
(
    id_departament number(4) not null,
    denumire_departament varchar2(30) not null,
    id_manager number(6)
);
alter table departamente_proiect
add constraint departamente_id_dep_pk primary key (id_departament);
```

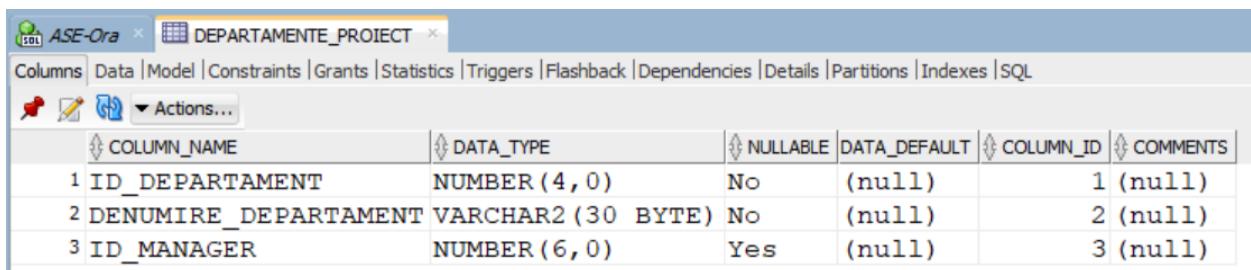


The screenshot shows the Oracle SQL Developer interface with the tab 'DEPARTAMENTE\_PROIECT' selected. The code area contains the SQL statements for creating the table and adding a primary key constraint.

```

1 create table departamente_proiect
2 (
3     id_departament number(4) not null,
4     denumire_departament varchar2(30) not null,
5     id_manager number(6)
6 );
7 alter table departamente_proiect
8 add constraint departamente_id_dep_pk primary key (id_departament);
9

```



The screenshot shows the 'Columns' tab for the 'DEPARTAMENTE\_PROIECT' table in Oracle SQL Developer. The table has three columns:

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_DEPARTAMENT	NUMBER (4, 0)	No	(null)	1	(null)
2 DENUMIRE_DEPARTAMENT	VARCHAR2 (30 BYTE)	No	(null)	2	(null)
3 ID_MANAGER	NUMBER (6, 0)	Yes	(null)	3	(null)

```
create table angajati_proiect
(
    id_angajat number(6) not null,
    nume varchar2(20),
    prenume varchar2(20),
    email varchar2(30),
    telefon varchar2(20),
    data_nastere date not null,
    data_angajare date not null,
    salariu number(8,2),
    id_functie varchar2(10),
    id_departament number(4),
    id_manager number(6)
);
alter table angajati_proiect
add constraint angajati_id_ang_pk primary key (id_angajat);
alter table angajati_proiect
add constraint angajati_email_uk unique (email);
alter table angajati_proiect
add constraint angajati_departamente_fk foreign key (id_departament)
references departamente_proiect (id_departament);
alter table angajati_proiect
add constraint angajati_manager_fk foreign key (id_manager)
references angajati_proiect (id_angajat);
alter table angajati_proiect
add constraint angajati_salariu_min
check (salariu > 0);
```

ASE-Ora ANGAJATI\_PROIECT

Worksheet Query Builder

```

1 create table angajati_proiect
2 (
3     id_angajat number(6) not null,
4     nume varchar2(20),
5     prenume varchar2(20),
6     email varchar2(30),
7     telefon varchar2(20),
8     data_nastere date not null,
9     data_angajare date not null,
10    salariu number(8,2),
11    id_functie varchar2(10),
12    id_departament number(4),
13    id_manager number(6)
14 );
15 alter table angajati_proiect
16 add constraint angajati_id_ang_pk primary key (id_angajat);
17 alter table angajati_proiect
18 add constraint angajati_email_uk unique (email);
19 alter table angajati_proiect
20 add constraint angajati_departamente_fk foreign key (id_departament)
21 references departamente_proiect (id_departament);
22 alter table angajati_proiect
23 add constraint angajati_manager_fk foreign key (id_manager)
24 references angajati_proiect (id_angajat);
25 alter table angajati_proiect
26 add constraint angajati_salariu_min
27 check (salariu > 0);
28

```

ASE-Ora ANGAJATI\_PROIECT

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

Actions... ▾

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_ANGAJAT	NUMBER (6,0)	No	(null)	1	(null)
2 NUME	VARCHAR2 (20 BYTE)	Yes	(null)	2	(null)
3 PRENUME	VARCHAR2 (20 BYTE)	Yes	(null)	3	(null)
4 EMAIL	VARCHAR2 (30 BYTE)	Yes	(null)	4	(null)
5 TELEFON	VARCHAR2 (20 BYTE)	Yes	(null)	5	(null)
6 DATA_NASTERE	DATE	No	(null)	6	(null)
7 DATA_ANGAJARE	DATE	No	(null)	7	(null)
8 SALARIU	NUMBER (8,2)	Yes	(null)	8	(null)
9 ID_FUNCTIE	VARCHAR2 (10 BYTE)	Yes	(null)	9	(null)
10 ID_DEPARTAMENT	NUMBER (4,0)	Yes	(null)	10	(null)
11 ID_MANAGER	NUMBER (6,0)	Yes	(null)	11	(null)

```

create table proiecte_proiect
(
    id_proiect number(4),
    denumire_proiect varchar2(30),
    descriere varchar2(200),
    categorie varchar2(30)
);
alter table proiecte_proiect
add constraint proiecte_id_proiect_pk primary key (id_proiect);

```

The screenshot shows the ASE-Ora SQL Worksheet interface. The title bar says "ASE-Ora" and "PROIECTE\_PROIECT". The toolbar includes icons for running, saving, and zooming. The status bar shows "0.109 seconds". The main area has tabs for "Worksheet" and "Query Builder", with "Worksheet" selected. The code is displayed in a numbered format:

```

1 create table proiecte_proiect
2 (
3     id_proiect number(4),
4     denumire_proiect varchar2(30),
5     descriere varchar2(200),
6     categorie varchar2(30)
7 );
8 alter table proiecte_proiect
9 add constraint proiecte_id_proiect_pk primary key (id_proiect);
10

```

The screenshot shows the ASE-Ora interface with the "PROIECTE\_PROIECT" table selected. The top menu bar includes "Columns", "Data", "Model", "Constraints", "Grants", "Statistics", "Triggers", "Flashback", "Dependencies", "Details", "Partitions", "Indexes", and "SQL". Below the menu is an "Actions..." dropdown with icons for insert, update, delete, and refresh. The table structure is shown in a grid:

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_PROIECT	NUMBER(4, 0)	No	(null)	1	(null)
2 DENUMIRE_PROIECT	VARCHAR2(30 BYTE)	Yes	(null)	2	(null)
3 DESCRIERE	VARCHAR2(200 BYTE)	Yes	(null)	3	(null)
4 CATEGORIE	VARCHAR2(30 BYTE)	Yes	(null)	4	(null)

```

create table participanti_proiect
(
    id_participant number(4) not null,
    nume_participant varchar2(20) not null,
    prenume_participant varchar2(20),
    email_participant varchar2(30) not null,
    data_nastere_participant date not null
);
alter table participanti_proiect
add constraint participanti_id_pk primary key (id_participant);
alter table participanti_proiect
add constraint participanti_email_uk unique (email_participant);

```

The screenshot shows the Oracle SQL Developer interface with the following details:

- Toolbar:** Includes icons for running the script, saving, and zooming.
- Top Bar:** Shows the connection name "ASE-Ora" and the table name "PARTICIPANTI\_PROJECT". It also displays the time taken for the operation: "0.15000001 seconds".
- Worksheet Tab:** Active tab, showing the SQL code for creating the table and adding constraints.
- Code Area:**

```

1 create table participanti_proiect
2 (
3     id_participant number(4) not null,
4     nume_participant varchar2(20) not null,
5     prenume_participant varchar2(20),
6     email_participant varchar2(30) not null,
7     data_nastere_participant date not null
8 );
9 alter table participanti_proiect
10 add constraint participanti_id_pk primary key (id_participant);
11 alter table participanti_proiect|
12 add constraint participanti_email_uk unique (email_participant);
13

```

The screenshot shows the Oracle SQL Developer interface with the following details:

- Toolbar:** Includes icons for running the script, saving, and zooming.
- Top Bar:** Shows the connection name "ASE-Ora" and the table name "PARTICIPANTI\_PROJECT".
- Table Structure View:** Shows the columns of the "PARTICIPANTI\_PROJECT" table with their data types, nullability, default values, column IDs, and comments.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_PARTICIPANT	NUMBER (4,0)	No	(null)	1	(null)
2 NUME_PARTICIPANT	VARCHAR2 (20 BYTE)	No	(null)	2	(null)
3 PRENUME_PARTICIPANT	VARCHAR2 (20 BYTE)	Yes	(null)	3	(null)
4 EMAIL_PARTICIPANT	VARCHAR2 (30 BYTE)	No	(null)	4	(null)
5 DATA_NASTERE_PARTICIPANT	DATE	No	(null)	5	(null)

```

create table contributii_proiect
(
    id_contributie number(6),
    id_participant number(4),
    id_proiect number(4),
    tip_contributie varchar2(30),
    valoare_in_bani number(8,2)
);

alter table contributii_proiect
add constraint contributii_id_cont_pk primary key (id_contributie);

alter table contributii_proiect
add constraint contributii_participanti_fk foreign key (id_participant)
references participanti_proiect (id_participant);

alter table contributii_proiect
add constraint contributii_proiecte_fk foreign key (id_proiect)
references proiecte_proiect (id_proiect);

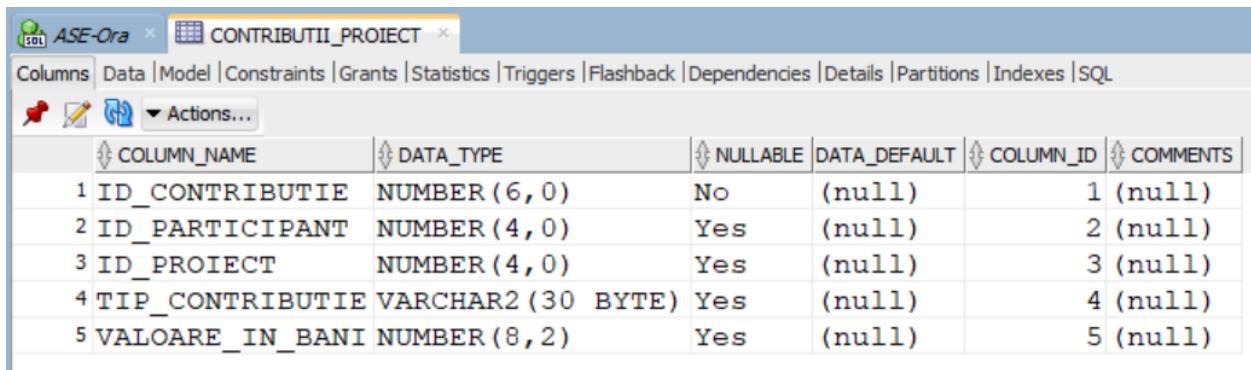
```

The screenshot shows the Oracle SQL Developer interface with a worksheet tab open. The code in the worksheet is identical to the one above, defining the 'contributii\_proiect' table and its constraints. The code is numbered from 1 to 17. The 'contributii\_id\_cont\_pk' constraint is highlighted in yellow.

```

1 create table contributii_proiect
2 (
3     id_contributie number(6),
4     id_participant number(4),
5     id_proiect number(4),
6     tip_contributie varchar2(30),
7     valoare_in_bani number(8,2)
8 );
9 alter table contributii_proiect
10 add constraint contributii_id_cont_pk primary key (id_contributie);
11 alter table contributii_proiect
12 add constraint contributii_participanti_fk foreign key (id_participant)
13 references participanti_proiect (id_participant);
14 alter table contributii_proiect
15 add constraint contributii_proiecte_fk foreign key (id_proiect)
16 references proiecte_proiect (id_proiect);
17

```



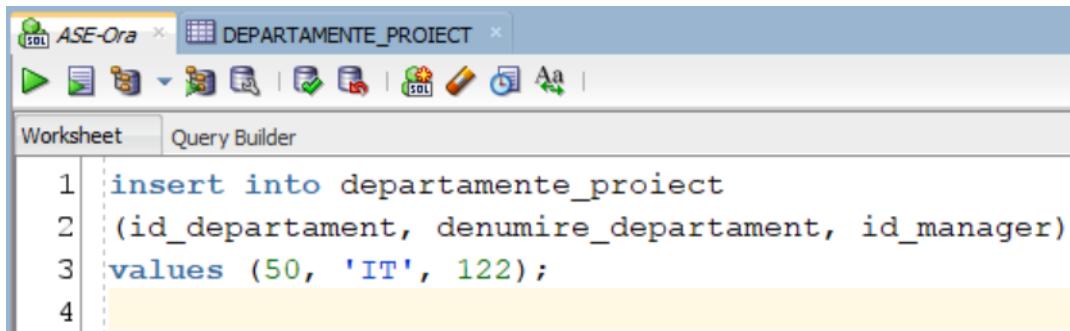
The screenshot shows the Oracle SQL Developer interface with the title bar "ASE-Ora" and "CONTRIBUTII\_PROJECT". The "Columns" tab is selected. The table structure is as follows:

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_CONTRIBUTIE	NUMBER (6, 0)	No	(null)	1	(null)
2 ID_PARTICIPANT	NUMBER (4, 0)	Yes	(null)	2	(null)
3 ID_PROIECT	NUMBER (4, 0)	Yes	(null)	3	(null)
4 TIP_CONTRIBUTIE	VARCHAR2 (30 BYTE)	Yes	(null)	4	(null)
5 VALOARE_IN_BANI	NUMBER (8, 2)	Yes	(null)	5	(null)

### Interactiunea cu serverul prin intermediul comenzilor SQL (LDD si LMD)

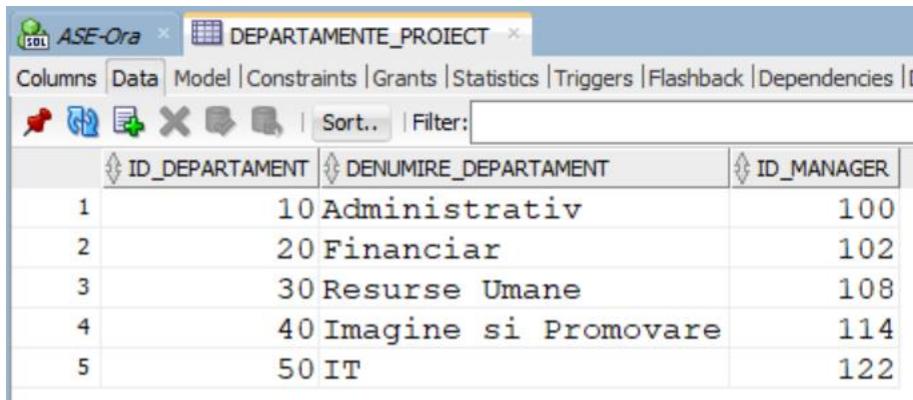
--Exemple pentru inserarea datelor in tabelele create

```
insert into departamente_proiect
(id_departament, denumire_departament, id_manager)
values (50, 'IT', 122);
```



The screenshot shows the Oracle SQL Developer interface with the title bar "ASE-Ora" and "DEPARTAMENTE\_PROJECT". The "Worksheet" tab is selected. The SQL code is:

```
1 insert into departamente_proiect
2 (id_departament, denumire_departament, id_manager)
3 values (50, 'IT', 122);
4
```



The screenshot shows the Oracle SQL Developer interface with the title bar "ASE-Ora" and "DEPARTAMENTE\_PROJECT". The "Columns" tab is selected. The table data is:

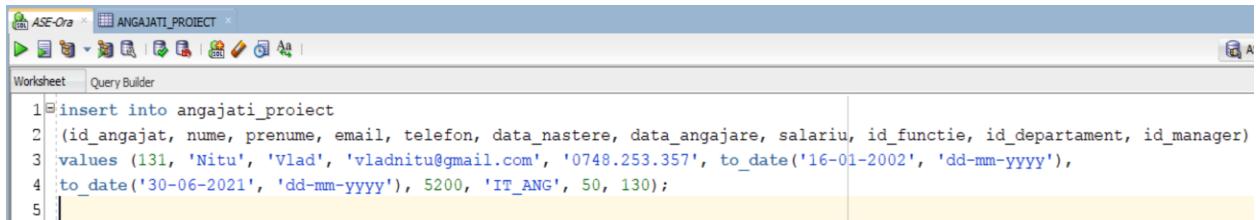
ID_DEPARTAMENT	DENUMIRE_DEPARTAMENT	ID_MANAGER
1	10 Administrativ	100
2	20 Financiar	102
3	30 Resurse Umane	108
4	40 Imagine si Promovare	114
5	50 IT	122

insert into angajati\_proiect

(id\_angajat, nume, prenume, email, telefon, data\_nastere, data\_angajare, salariu, id\_functie,  
id\_departament, id\_manager)

values (131, 'Nitu', 'Vlad', 'vladnitu@gmail.com', '0748.253.357', to\_date('16-01-2002', 'dd-mm-yyyy'),

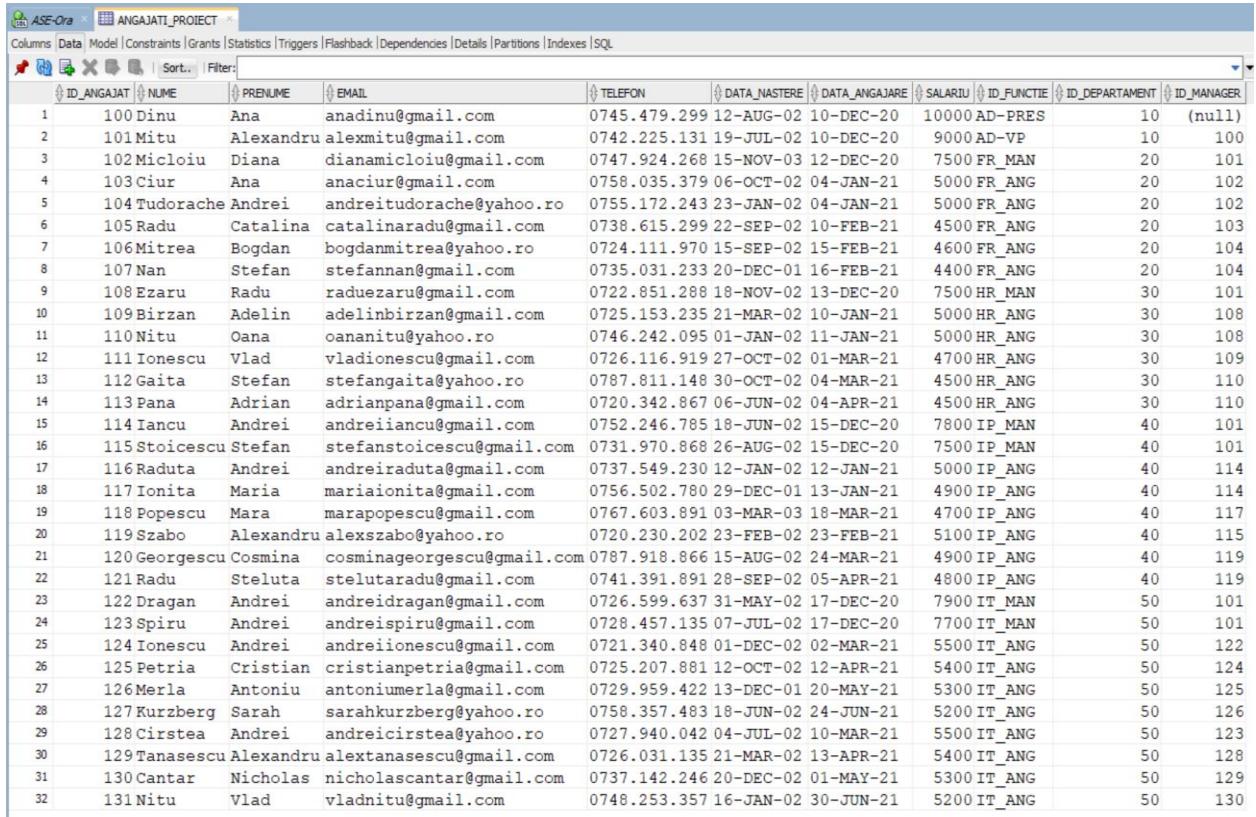
to\_date('30-06-2021', 'dd-mm-yyyy'), 5200, 'IT\_ANG', 50, 130);



```

1| insert into angajati_proiect
2| (id_angajat, nume, prenume, email, telefon, data_nastere, data_angajare, salariu, id_functie,
3| values (131, 'Nitu', 'Vlad', 'vladnitu@gmail.com', '0748.253.357', to_date('16-01-2002', 'dd-mm-yyyy'),
4| to_date('30-06-2021', 'dd-mm-yyyy'), 5200, 'IT_ANG', 50, 130);
5|

```



ID_ANGAJAT	NUME	PRENUME	EMAIL	TELEFON	DATA_NASTERE	DATA_ANGAJARE	SALARIU	ID_FUNCTIE	ID_DEPARTAMENT	ID_MANAGER	
1	100	Dinu	Ana	anadinu@gmail.com	0745.479.299	12-AUG-02	10-DEC-20	10000	AD-PRES	10	(null)
2	101	Mitu	Alexandru	alexmitu@gmail.com	0742.225.131	19-JUL-02	10-DEC-20	9000	AD-VP	10	100
3	102	Micloiu	Diana	dianamicloiu@gmail.com	0747.924.268	15-NOV-03	12-DEC-20	7500	FR_MAN	20	101
4	103	Ciur	Ana	anaciur@gmail.com	0758.035.379	06-OCT-02	04-JAN-21	5000	FR_ANG	20	102
5	104	Tudorache	Andrei	andreitudorache@yahoo.ro	0755.172.243	23-JAN-02	04-JAN-21	5000	FR_ANG	20	102
6	105	Radu	Catalina	catalinaradu@gmail.com	0738.615.299	22-SEP-02	10-FEB-21	4500	FR_ANG	20	103
7	106	Mitrea	Bogdan	bogdanmitrea@yahoo.ro	0724.111.970	15-SEP-02	15-FEB-21	4600	FR_ANG	20	104
8	107	Nan	Stefan	stefannan@gmail.com	0735.031.233	20-DEC-01	16-FEB-21	4400	FR_ANG	20	104
9	108	Ezaru	Radu	raduezaru@gmail.com	0722.851.288	18-NOV-02	13-DEC-20	7500	HR_MAN	30	101
10	109	Birzan	Adelin	adelinbirzan@gmail.com	0725.153.235	21-MAR-02	10-JAN-21	5000	HR_ANG	30	108
11	110	Nitu	Oana	oanานitu@yahoo.ro	0746.242.095	01-JAN-02	11-JAN-21	5000	HR_ANG	30	108
12	111	Ionescu	Vlad	vladionescu@gmail.com	0726.116.919	27-OCT-02	01-MAR-21	4700	HR_ANG	30	109
13	112	Gaita	Stefan	stefangaita@yahoo.ro	0787.811.148	30-OCT-02	04-MAR-21	4500	HR_ANG	30	110
14	113	Pana	Adrian	adriapanana@gmail.com	0720.342.867	06-JUN-02	04-APR-21	4500	HR_ANG	30	110
15	114	Iancu	Andrei	andreiancu@gmail.com	0752.246.785	18-JUN-02	15-DEC-20	7800	IP_MAN	40	101
16	115	Stoicescu	Stefan	stefanstoicescu@gmail.com	0731.970.868	26-AUG-02	15-DEC-20	7500	IP_MAN	40	101
17	116	Raduta	Andrei	andreiraduta@gmail.com	0737.549.230	12-JAN-02	12-JAN-21	5000	IP_ANG	40	114
18	117	Ionita	Maria	mariaionita@gmail.com	0756.502.780	29-DEC-01	13-JAN-21	4900	IP_ANG	40	114
19	118	Popescu	Mara	marapopescu@gmail.com	0767.603.891	03-MAR-03	18-MAR-21	4700	IP_ANG	40	117
20	119	Szabo	Alexandru	alexszabo@yahoo.ro	0720.230.202	23-FEB-02	23-FEB-21	5100	IP_ANG	40	115
21	120	Georgescu	Cosmina	cosminageorgescu@gmail.com	0787.918.866	15-AUG-02	24-MAR-21	4900	IP_ANG	40	119
22	121	Radu	Steluta	stelutaradu@gmail.com	0741.391.891	28-SEP-02	05-APR-21	4800	IP_ANG	40	119
23	122	Dragan	Andrei	andreidragan@gmail.com	0726.599.637	31-MAY-02	17-DEC-20	7900	IT_MAN	50	101
24	123	Spiru	Andrei	andreispiru@gmail.com	0728.457.135	07-JUL-02	17-DEC-20	7700	IT_MAN	50	101
25	124	Ionescu	Andrei	andreiiionescu@gmail.com	0721.340.848	01-DEC-02	02-MAR-21	5500	IT_ANG	50	122
26	125	Petria	Cristian	cristianpetria@gmail.com	0725.207.881	12-OCT-02	12-APR-21	5400	IT_ANG	50	124
27	126	Merla	Antoniu	antoniumerla@gmail.com	0729.959.422	13-DEC-01	20-MAY-21	5300	IT_ANG	50	125
28	127	Kurzberg	Sarah	sarahkurzberg@yahoo.ro	0758.357.483	18-JUN-02	24-JUN-21	5200	IT_ANG	50	126
29	128	Cirstea	Andrei	andreicirstea@yahoo.ro	0727.940.042	04-JUL-02	10-MAR-21	5500	IT_ANG	50	123
30	129	Tanasescu	Alexandru	alextanasescu@gmail.com	0726.031.135	21-MAR-02	13-APR-21	5400	IT_ANG	50	128
31	130	Cantar	Nicholas	nicholascantar@gmail.com	0737.142.246	20-DEC-02	01-MAY-21	5300	IT_ANG	50	129
32	131	Nitu	Vlad	vladnitu@gmail.com	0748.253.357	16-JAN-02	30-JUN-21	5200	IT_ANG	50	130

```
insert into proiecte_proiect
(id_proiect, denumire_proiect, descriere, categorie)
values (2, 'Next In Tech', 'Un proiect in care promovam tehnologia si inovatia.',
'tehnologie');
```

```
insert into proiecte_proiect
(id_proiect, denumire_proiect, descriere, categorie)
values (2, 'Next In Tech', 'Un proiect in care promovam tehnologia si inovatia.', 'tehnologie');
```

ID_PROJECT	DENUMIRE_PROJECT	DESCRIERE	CATEGORIE
1	Dorinrete Indeplinite	Un proiect in care ii ajutam cu bani, mancare sau lucruri pe cei care au nevoie.	caritabil
2	2 Next In Tech	Un proiect in care promovam tehnologia si inovatia.	tehnologie
3	3 Entertainment Masters	Un proiect in care vrem sa aducem un strop de distractie in viata tuturor.	entertainment

```
insert into participanti_proiect
(id_participant, nume_participant, prenume_participant, email_participant,
data_nastere_participant)
values (7, 'Ilie', 'David', 'davidilie@gmail.com', to_date('15-12-2004', 'dd-mm-yyyy'));
```

```
insert into participanti_proiect
(id_participant, nume_participant, prenume_participant, email_participant, data_nastere_participant)
values (7, 'Ilie', 'David', 'davidilie@gmail.com', to_date('15-12-2004', 'dd-mm-yyyy'));
```

ID_PARTICIPANT	NUME_PARTICIPANT	PRENUME_PARTICIPANT	EMAIL_PARTICIPANT	DATA_NASTERE_PARTICIPANT
1	Szabo	Tiberiu	tiberiuszabo@gmail.com	15-JUL-06
2	Popescu	Teodora	teopopescu@gmail.com	23-JUL-07
3	Lasue	Nathalie	nathalielasue@gmail.com	12-AUG-06
4	Popescu	Andrei	andreipopescu@gmail.com	24-FEB-01
5	Voinea	Andrei	andreivoinea@gmail.com	20-DEC-01
6	Capatina	Bogdan	bogdancapatina@gmail.com	12-APR-05
7	Ilie	David	davidilie@gmail.com	15-DEC-04

```
insert into contributii_proiect
(id_contributie, id_participant, id_proiect, tip_contributie, valoare_in_bani)
values (6, 7, 3, 'donatie in lei', 800);
```

The screenshot shows two windows in Oracle SQL Developer. The top window is a 'Worksheet' showing the SQL code:

```
1 insert into contributii_proiect
2 (id_contributie, id_participant, id_proiect, tip_contributie, valoare_in_bani)
3 values (6, 7, 3, 'donatie in lei', 800);
```

The bottom window is a 'CONTRIBUTII\_PROJECT' table viewer showing the data:

ID_CONTRIBUTIE	ID_PARTICIPANT	ID_PROIECT	TIPOContributie	VALOARE_IN_BANI
1	1	1	1 jucarii	(null)
2	2	2	1 dulciuri	(null)
3	3	4	2 donatie in lei	1000
4	4	5	2 donatie in lei	1400
5	5	6	3 mancare si bauturi	(null)
6	6	7	3 donatie in lei	800

--Exemplu de update (actualizarea tabelelor)

--Modificarea tipului de contributie cu denumirea "donatie in lei" in "donatie"

```
update contributii_proiect
set tip_contributie = 'donatie'
where tip_contributie = 'donatie in lei';
```

The screenshot shows a 'Worksheet' window with the following SQL code:

```
1 update contributii_proiect
2 set tip_contributie = 'donatie'
3 where tip_contributie = 'donatie in lei';
```

ID_CONTRIBUTIE	ID_PARTICIPANT	ID_PROIECT	TIP_CONTRIBUTIE	VALOARE_IN_BANI
1	1	1	1 jucarii	(null)
2	2	2	1 dulciuri	(null)
3	3	4	2 donatie	1000
4	4	5	2 donatie	1400
5	5	6	3 mancare si bauturi	(null)
6	6	7	3 donatie	800

--Modificarea salariului cu 100 pentru angajatii cu data angajarii mai devreme de 01-02-2021

```
update angajati_proiect
set salariu = salariu + 100
where data_angajare < to_date('01-02-2021', 'dd-mm-yyyy');
```

```
1 update angajati_proiect
2 set salariu = salariu + 100
3 where data_angajare < to_date('01-02-2021', 'dd-mm-yyyy');
```

ID_PROJECT	DENUMIRE_PROJECT	DESCRIERE	CATEGORIE
1	1 Dorinte Indeplinite	Un proiect in care ii ajutam cu bani, mancare sau lucruri pe cei care au nevoie.	caritabil
2	2 Next In Tech	Un proiect in care promovam tehnologia si inovatia.	tehnologie
3	3 Entertainment Masters	Un proiect in care vrem sa aducem un strop de distractie in viata tuturor.	entertainment
4	4 Amenajare Scena	Amenajam scena pentru un posibil eveniment din viitor.	intern

--Exemplu de delete (stergerea din baza de date)

--Se va sterge proiectul cu denumirea "Amenajare scena"

```
delete from proiecte_proiect
where denumire_proiect = 'Amenajare Scena';
```

ID_PROJECT	DENUMIRE_PROJECT	DESCRIERE	CATEGORIE
1	Dorinte Indeplinite	Un proiect in care ii ajutam cu bani, mancare sau lucruri pe cei care au nevoie.	caritabil
2	Next In Tech	Un proiect in care promovam tehnologia si inovatia.	tehnologie
3	Entertainment Masters	Un proiect in care vrem sa aducem un strop de distractie in viata tuturor.	entertainment

--Exemplu de interogare a datelor din table

--Se vor afisa angajati cu functie de manager

```
select * from angajati_proiect
```

```
where id_functie like '%MAN';
```

ID_ANGAJAT	NUME	PRENUME	EMAIL	TELEFON	DATA_NASTERE	DATA_ANGAJARE	SALARIU	ID_FUNCTIE	ID_DEPARTAMENT	ID_MANAGER
1	102 Micloiu	Diana	dianamicloiu@gmail.com	0747.924.268	15-NOV-03	12-DEC-20	7600	FR_MAN	20	101
2	108 Ezaru	Radu	raduezaru@gmail.com	0722.851.288	18-NOV-02	13-DEC-20	7600	HR_MAN	30	101
3	114 Iancu	Andrei	andreiiancu@gmail.com	0752.246.785	18-JUN-02	15-DEC-20	7900	IP_MAN	40	101
4	115 Stoicescu	Stefan	stefanstoicescu@gmail.com	0731.970.868	26-AUG-02	15-DEC-20	7600	IP_MAN	40	101
5	122 Dragan	Andrei	andreidragan@gmail.com	0726.599.637	31-MAY-02	17-DEC-20	8000	IT_MAN	50	101
6	123 Spiru	Andrei	andreispiru@gmail.com	0728.457.135	07-JUL-02	17-DEC-20	7800	IT_MAN	50	101

--Se vor afisa numele si functia angajatilor cu prenumele Andrei

```
select id_angajat, nume, prenume, id_functie
```

```
from angajati_proiect
```

```
where prenume = 'Andrei';
```

ID_ANGAJAT	NUME	PRENUME	ID_FUNCTIE
1	104 Tudorache	Andrei	FR_ANG
2	114 Iancu	Andrei	IP_MAN
3	116 Raduta	Andrei	IP_ANG
4	122 Dragan	Andrei	IT_MAN
5	123 Spiru	Andrei	IT_MAN
6	124 Ionescu	Andrei	IT_ANG
7	128 Cirstea	Andrei	IT_ANG

--Se vor afisa departamentele angajatilor al caror nume incepe cu litera M

```
select a.id_angajat, a.nume, a.prenume, a.id_departament, d.denumire_departament
from angajati_proiect a, departamente_proiect d
where a.id_departament = d.id_departament
and a.nume like 'M%';
```

The screenshot shows the Oracle SQL Developer interface. The top window is titled 'DEPARTAMENTE\_PROJECT' and contains the SQL query:

```
1 select a.id_angajat, a.nume, a.prenume, a.id_departament, d.denumire_departament
2 from angajati_proiect a, departamente_proiect d
3 where a.id_departament = d.id_departament
4 and a.nume like 'M%';
```

The bottom window is titled 'Query Result' and displays the results of the query:

ID_ANGAJAT	NUME	PRENUME	ID_DEPARTAMENT	DENUMIRE_DEPARTAMENT
1	101Mitu	Alexandru	10	Administrativ
2	102Micloiu	Diana	20	Financiar
3	106Mitrea	Bogdan	20	Financiar
4	126Merla	Antoniu	50	IT

The screenshot shows the Oracle SQL Developer interface. The top window is titled 'PARTICIPANTI\_POSIBILI' and displays the table structure:

ID_PARTICIPANT	NUME_PARTICIPANT	PRENUME_PARTICIPANT	EMAIL_PARTICIPANT	DATA_NASTERE_PARTICIPANT
1	3 Lasue	Nathalie	nathanielasue@yahoo.ro	12-AUG-06
2	100 Dita	Mara	maradita@gmail.com	01-DEC-02
3	101 Dita	Stefan	stefandita@gmail.com	20-JUN-07

--Exemplu de merge

```
merge into participanti_posibili po
```

```
using participanti_proiect pr
```

```
on (po.id_participant = pr.id_participant)
```

```
when not matched then
```

```
    insert (po.id_participant, po.nume_participant, po.prenume_participant, po.email_participant,
    po.data_nastere_participant)
```

```
    values (pr.id_participant, pr.nume_participant, pr.prenume_participant,
    pr.email_participant, pr.data_nastere_participant)
```

```
when matched then
```

```
    update set po.email_participant = pr.email_participant;
```

ID_PARTICIPANT	NUME_PARTICIPANT	PRENUME_PARTICIPANT	EMAIL_PARTICIPANT	DATA_NASTERE_PARTICIPANT
1	3 Lasue	Nathalie	nathalielasue@gmail.com	12-AUG-06
2	100 Dita	Mara	maradita@gmail.com	01-DEC-02
3	101 Dita	Stefan	stefandita@gmail.com	20-JUN-07
4	1 Szabo	Tiberiu	tiberiuszabo@gmail.com	15-JUL-06
5	2 Popescu	Teodora	teopopescu@gmail.com	23-JUL-07
6	4 Popescu	Andrei	andreipopescu@gmail.com	24-FEB-01
7	5 Voinea	Andrei	andreivoinea@gmail.com	20-DEC-01
8	6 Capatina	Bogdan	bogdancapatina@gmail.com	12-APR-05
9	7 Ilie	David	davidilie@gmail.com	15-DEC-04

--Exemplu cu ANY

--Se vor afisa angajatii care nu fac parte din departamentul IT si au salariul mai mic decat unul dintre angajat din departamentul IT

```
select id_angajat, nume, prenume, id_functie, salariu
from angajati_proiect
where salariu < any (select salariu from angajati_proiect where id_functie like '%IT%')
and id_functie <> 'IT_MAN' and id_functie <> 'IT_ANG'
order by salariu desc;
```

ID_ANGAJAT	NUME	PRENUME	ID_FUNCTIE	SALARIU
1	114 Iancu	Andrei	IP_MAN	7900
2	108 Ezaru	Radu	HR_MAN	7600
3	102 Micloiu	Diana	FR_MAN	7600
4	115 Stoicescu	Stefan	IP_MAN	7600
5	109 Birzan	Adelin	HR_ANG	5100
6	104 Tudorache	Andrei	FR_ANG	5100
7	103 Ciur	Ana	FR_ANG	5100
8	110 Nitu	Oana	HR_ANG	5100
9	116 Raduta	Andrei	IP_ANG	5100
10	119 Szabo	Alexandru	IP_ANG	5100
11	117 Ionita	Maria	IP_ANG	5000
12	120 Georgescu	Cosmina	IP_ANG	4900
13	121 Radu	Steluta	IP_ANG	4800
14	118 Popescu	Mara	IP_ANG	4700
15	111 Ionescu	Vlad	HR_ANG	4700
16	106 Mitrea	Bogdan	FR_ANG	4600
17	105 Radu	Catalina	FR_ANG	4500
18	112 Gaita	Stefan	HR_ANG	4500
19	113 Pana	Adrian	HR_ANG	4500
20	107 Nan	Stefan	FR_ANG	4400

--Exemplu cu ALL

--Se vor afisa in ordine descrescatoare a salariilor angajatii cu salariul mai mic decat toti angajatii din departamentul Imagine si Promovare

```
select id_angajat, nume, prenume, id_functie, salariu
from angajati_proiect
where salariu < all (select salariu from angajati_proiect where id_functie like '%IP%')
order by salariu desc;
```

The screenshot shows the ASE-Ora SQL Workbench interface. The top window is titled 'ANGAJATI\_PROIECT'. Below it, the 'Worksheet' tab is active, displaying the following SQL code:

```
1 select id_angajat, nume, prenume, id_functie, salariu
2 from angajati_proiect
3 where salariu < all (select salariu from angajati_proiect where id_functie like '%IP%')
4 order by salariu desc;
```

Below the worksheet, the 'Query Result' tab is open, showing the following table with 5 rows of data:

ID_ANGAJAT	NUME	PRENUME	ID_FUNCTIE	SALARIU	
1	106	Mitrea	Bogdan	FR_ANG	4600
2	112	Gaita	Stefan	HR_ANG	4500
3	105	Radu	Catalina	FR_ANG	4500
4	113	Pana	Adrian	HR_ANG	4500
5	107	Nan	Stefan	FR_ANG	4400

--Exemplu jonctiune de egalitate

--Se vor afisa date despre angajatii al caror nume incepe cu litera S, pe baza id\_departament

```
select a.id_angajat, a.nume, a.prenume, a.id_departament, d.denumire_departament
from angajati_proiect a, departamente_proiect d
where a.id_departament = d.id_departament
and a.nume like 'S%';
```

The screenshot shows a SQL query being run in ASE-Ora. The query selects information from the ANGAJATI\_PROJECT and DEPARTAMENTE\_PROJECT tables based on department ID and name starting with 'S%'. The results are displayed in a grid:

ID_ANGAJAT	NUME	PRENUME	ID_DEPARTAMENT	DENUMIRE_DEPARTAMENT
1	115 Stoicescu	Stefan	40	Imagine si Promovare
2	119 Szabo	Alexandru	40	Imagine si Promovare
3	123 Spiru	Andrei	50	IT

--Exemplu de jonctiune externă

--Se vor afisa date despre participant si contributiile acestora, indiferent daca valorile sunt nule

```
select p.id_participant, p.nume_participant, p.prenume_participant, c.tip_contributie,
c.valoare_in_bani
from participanti_proiect p, contributii_proiect c
where p.id_participant = c.id_participant(+);
```

The screenshot shows a query running in ASE-Ora. It selects data from the PARTICIPANTI\_PROJECT and CONTRIBUTII\_PROJECT tables using an outer join condition. The results are shown in a table:

ID_PARTICIPANT	NUME_PARTICIPANT	PRENUME_PARTICIPANT	TIP_CONTRIBUTIE	VALOARE_IN_BANI
1	1 Szabo	Tiberiu	jucarii	(null)
2	2 Popescu	Teodora	dulciuri	(null)
3	3 Lasue	Nathalie	(null)	(null)
4	4 Popescu	Andrei	donatie	1000
5	5 Voinea	Andrei	donatie	1400
6	6 Capatina	Bogdan	mancare si bauturi	(null)
7	7 Ilie	David	donatie	800
8	8 Sturza	Mihai	(null)	(null)
9	9 Simion	Stefan	(null)	(null)
10	10 Stefan	Roberto	(null)	(null)

--Exemplu de interogari subordinate--Se vor afisa date despre angajatii care au acelasi id\_departament ca angajatul Adrian

```
select * from angajati_proiect
where id_departament = (select id_departament from angajati_proiect
                          where upper(prenume) = 'ADRIAN');
```

The screenshot shows the Oracle SQL Developer interface. The top part displays the SQL query:

```
1 | select * from angajati_proiect
2 | where id_departament = (select id_departament from angajati_proiect
3 |                           where upper(prenume) = 'ADRIAN');
```

The bottom part shows the results of the query, titled "Query Result". The results are as follows:

ID_ANGAJAT	NUME	PRENUME	EMAIL	TELEFON	DATA_NASTERE	DATA_ANGAJARE	SALARIU	ID_FUNCTIE	ID_DEPARTAMENT	ID_MANAGER
1	108 Ezaru	Radu	raduezaru@gmail.com	0722.851.288	18-NOV-02	13-DEC-20	7600	HR_MAN	30	101
2	109 Birzan	Adelin	adelinbirzan@gmail.com	0725.153.235	21-MAR-02	10-JAN-21	5100	HR_ANG	30	108
3	110 Nitu	Oana	oananutu@yahoo.ro	0746.242.095	01-JAN-02	11-JAN-21	5100	HR_ANG	30	108
4	111 Ionescu	Vlad	vladionescu@gmail.com	0726.116.919	27-OCT-02	01-MAR-21	4700	HR_ANG	30	109
5	112 Gaita	Stefan	stefangaita@yahoo.ro	0787.811.148	30-OCT-02	04-MAR-21	4500	HR_ANG	30	110
6	113 Pana	Adrian	adrianpana@gmail.com	0720.342.867	06-JUN-02	04-APR-21	4500	HR_ANG	30	110

--Se vor afisa date despre angajatii care au salariul mai mare decat angajatul Petria

```
select * from angajati_proiect
where salariu > (select salariu from angajati_proiect
                      where lower(nume) = 'petria');
```

The screenshot shows the Oracle SQL Developer interface. The top part displays the SQL query:

```
1 | select * from angajati_proiect
2 | where salariu > (select salariu from angajati_proiect
3 |                       where lower(nume) = 'petria');
```

The bottom part shows the results of the query, titled "Query Result". The results are as follows:

ID_ANGAJAT	NUME	PRENUME	EMAIL	TELEFON	DATA_NASTERE	DATA_ANGAJARE	SALARIU	ID_FUNCTIE	ID_DEPARTAMENT	ID_MANAGER
1	100 Dinu	Ana	anadinu@gmail.com	0745.479.299	12-AUG-02	10-DEC-20	10100	AD-PRES	10	(null)
2	101 Mitu	Alexandru	alexandru.mitu@gmail.com	0742.225.131	19-JUL-02	10-DEC-20	9100	AD-VP	10	100
3	102 Micloiu	Diana	dianamicloiu@gmail.com	0747.924.268	15-NOV-03	12-DEC-20	7600	FR_MAN	20	101
4	108 Ezaru	Radu	raduezaru@gmail.com	0722.851.288	18-NOV-02	13-DEC-20	7600	HR_MAN	30	101
5	114 Iancu	Andrei	andreiiancu@gmail.com	0752.246.785	18-JUN-02	15-DEC-20	7900	IP_MAN	40	101
6	115 Stoicescu	Stefan	stefanstoicescu@gmail.com	0731.970.868	26-AUG-02	15-DEC-20	7600	IP_MAN	40	101
7	122 Dragan	Andrei	andreidragan@gmail.com	0726.599.637	31-MAY-02	17-DEC-20	8000	IT_MAN	50	101
8	123 Spiru	Andrei	andreaspiru@gmail.com	0728.457.135	07-JUL-02	17-DEC-20	7800	IT_MAN	50	101
9	124 Ionescu	Andrei	andreionescu@gmail.com	0721.340.848	01-DEC-02	02-MAR-21	5500	IT_ANG	50	122
10	128 Cirstea	Andrei	andreicirstea@yahoo.ro	0727.940.042	04-JUL-02	10-MAR-21	5500	IT_ANG	50	123

--Exemplu de functii SQL--Functia de concatenare

```
select 'Angajatul ' || nume || '' || prenume || ' are salariul ' || salariu || ' RON.'
from angajati_proiect;
```

The screenshot shows the Oracle SQL Developer interface. The query window contains the following code:

```
1 select 'Angajatul ' || nume || '' || prenume || ' are salariul ' || salariu || ' RON.'
2 from angajati_proiect;
```

The results window displays the output of the query, which concatenates the first name and last name of each employee followed by their salary and currency.

ANGAJATUL  NUME  ''  PRENUME  'ARESALARIUL'  SALARIU  'RON.'
1 Angajatul Dinu Ana are salariul 10100 RON.
2 Angajatul Mitu Alexandru are salariul 9100 RON.
3 Angajatul Micloiu Diana are salariul 7600 RON.
4 Angajatul Ciur Ana are salariul 5100 RON.
5 Angajatul Tudorache Andrei are salariul 5100 RON.
6 Angajatul Radu Catalina are salariul 4500 RON.
7 Angajatul Mitrea Bogdan are salariul 4600 RON.
8 Angajatul Nan Stefan are salariul 4400 RON.
9 Angajatul Ezaru Radu are salariul 7600 RON.
10 Angajatul Birzan Adelin are salariul 5100 RON.
11 Angajatul Nitu Oana are salariul 5100 RON.
12 Angajatul Toma are salariul 4700 RON.

--Se va afisa numele prescurtat folosind functia SUBSTR

```
select distinct substr(prenume,1,4) "Prescurtare"
from angajati_proiect
where prenume = 'Alexandru';
```

The screenshot shows the Oracle SQL Developer interface. The query window contains the following code:

```
1 select distinct substr(prenume,1,4) "Prescurtare"
2 from angajati_proiect
3 where prenume = 'Alexandru';
```

The results window displays the output of the query, which extracts the first four characters of the first name 'Alexandru'.

Prescurtare
1 Alex

--Se va calcula cate luni au trecut de la ultima angajare

```
select months_between (sysdate , max(data_angajare))
" Luni trecute"
from angajati_proiect;
```

The screenshot shows the Oracle SQL Developer interface. The top bar has tabs for 'ASE-Ora' and 'ANGAJATI\_PROJECT'. The main area is a 'Worksheet' tab containing the following SQL code:

```
1 select months_between (sysdate , max(data_angajare))
2 " Luni trecute"
3 from angajati_proiect;
4
```

Below the code, the 'Query Result' tab is open, showing the output:

```
Luni trecute
1 18.25256085722819593787335722819593787336
```

Information at the bottom of the result tab indicates 'All Rows Fetched: 1 in 0.012 seconds'.

--Functia ADD\_MONTHS

```
select add_months (max(data_angajare),6)
from angajati_proiect;
```

The screenshot shows the Oracle SQL Developer interface. The top bar has tabs for 'ASE-Ora' and 'ANGAJATI\_PROJECT'. The main area is a 'Worksheet' tab containing the following SQL code:

```
1 select add_months (max(data_angajare),6)
2 from angajati_proiect;
3
```

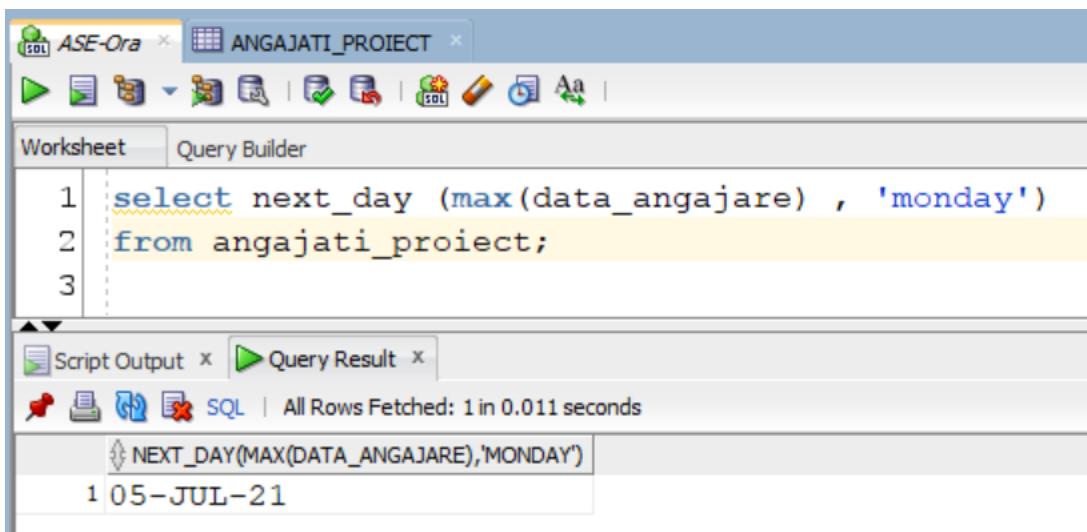
Below the code, the 'Query Result' tab is open, showing the output:

```
ADD_MONTHS(MAX(DATA_ANGAJARE),6)
1 31-DEC-21
```

Information at the bottom of the result tab indicates 'All Rows Fetched: 1 in 0.013 seconds'.

--Se va afisa data primei zi de luni dupa ultima angajare

```
select next_day (max(data_angajare) , 'monday')
from angajati_proiect;
```



The screenshot shows the ASE-Ora SQL Workbench interface. The title bar says "ASE-Ora" and "ANGAJATI\_PROJECT". The toolbar has various icons for database management. The main area is divided into "Worksheet" and "Query Builder" tabs, with "Worksheet" selected. The query in the worksheet is:

```

1 select next_day (max(data_angajare) , 'monday')
2 from angajati_proiect;
3

```

Below the worksheet is a "Script Output" tab showing the results of the query. The output window title is "Query Result" and it displays:

```

SQL | All Rows Fetched: 1 in 0.011 seconds
NEXT_DAY(MAX(DATA_ANGAJARE),'MONDAY')
1 05-JUL-21

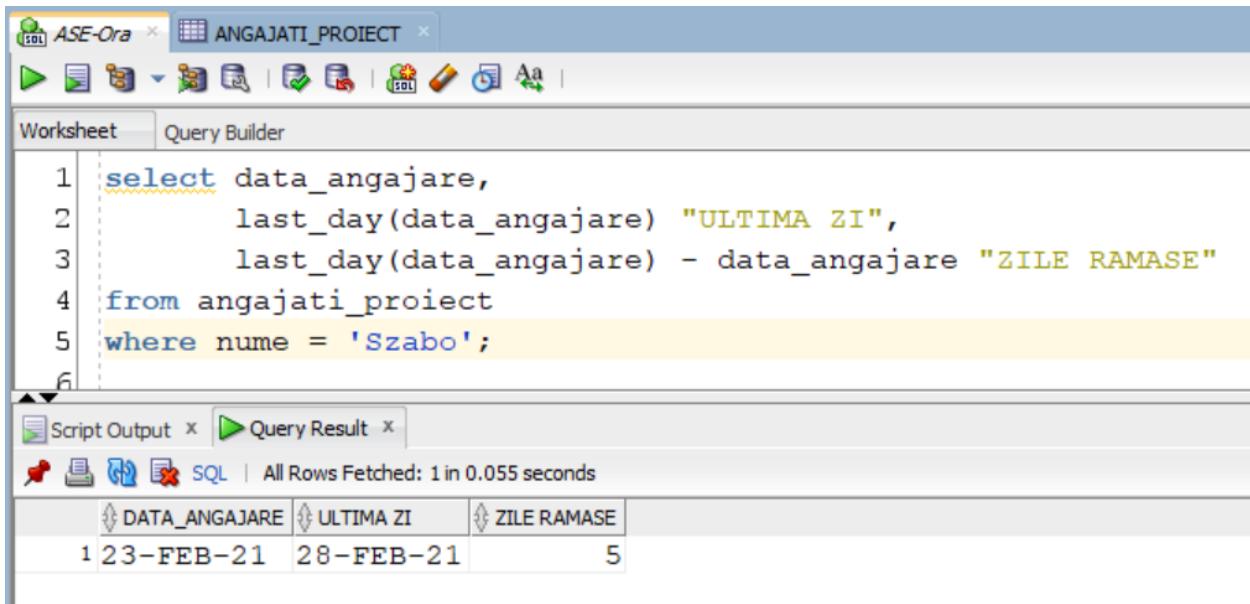
```

--Se va afisa cate zile din luna curenta au ramas dupa data angajarii unui angajat

```

select data_angajare,
       last_day(data_angajare) "ULTIMA ZI",
       last_day(data_angajare) - data_angajare "ZILE RAMASE"
  from angajati_proiect
 where nume = 'Szabo';

```



The screenshot shows the ASE-Ora SQL Workbench interface. The title bar says "ASE-Ora" and "ANGAJATI\_PROJECT". The toolbar has various icons for database management. The main area is divided into "Worksheet" and "Query Builder" tabs, with "Worksheet" selected. The query in the worksheet is:

```

1 select data_angajare,
2        last_day(data_angajare) "ULTIMA ZI",
3        last_day(data_angajare) - data_angajare "ZILE RAMASE"
4   from angajati_proiect
5  where nume = 'Szabo';
6

```

Below the worksheet is a "Script Output" tab showing the results of the query. The output window title is "Query Result" and it displays:

```

SQL | All Rows Fetched: 1 in 0.055 seconds
DATA_ANGAJARE    ULTIMA ZI      ZILE RAMASE
1 23-FEB-21     28-FEB-21      5

```

--Se va calcula salariul dupa o crestere salariala

```
select salariu + TO_NUMBER('1000.00', '9999D99')
"CRESTERE SALARIALA"
from angajati_proiect
WHERE nume = 'Szabo';
```

The screenshot shows the Oracle SQL Developer interface. The top bar has tabs for 'ASE-Ora' and 'ANGAJATI\_PROJECT'. The main area is a 'Worksheet' tab with the following SQL code:

```
1 select salariu + TO_NUMBER('1000.00', '9999D99')
2 "CRESTERE SALARIALA"
3 from angajati_proiect
4 WHERE nume = 'Szabo';
```

Below the code, the 'Script Output' tab shows the results:

	CRESTERE SALARIALA
1	6100

A message at the bottom of the output tab says 'All Rows Fetched: 1 in 0.148 seconds'.

--Se vor afisa date despre angajati, cat si managerul direct; daca nu exista, se va afisa un mesaj

```
select id_angajat, nume, prenume, id_functie,
nvl(to_char(id_manager), 'Presedintele nu are manager') id_manager
from angajati_proiect
where id_departament = 10;
```

The screenshot shows the Oracle SQL Developer interface. The top bar has tabs for 'ASE-Ora' and 'ANGAJATI\_PROJECT'. The main area is a 'Worksheet' tab with the following SQL code:

```
1 select id_angajat, nume, prenume, id_functie,
2 nvl(to_char(id_manager), 'Presedintele nu are manager') id_manager
3 from angajati_proiect
4 where id_departament = 10;
5
```

Below the code, the 'Script Output' tab shows the results:

ID_ANGAJAT	NUME	PRENUME	ID_FUNCTIE	ID_MANAGER
1	100	Dinu Ana	AD-PRES	Presedintele nu are manager
2	101	Mitu Alexandru	AD-VP	100

A message at the bottom of the output tab says 'All Rows Fetched: 2 in 0.021 seconds'.

--Se va calcula numarul de angajati din departamentul IT folosind functia COUNT

```
select count (distinct id_angajat)
```

```
"ANGAJATI IT"
```

```
from angajati_proiect
```

```
where id_functie like '%IT%';
```

```
1 select count (distinct id_angajat)
2 "ANGAJATI IT"
3 from angajati_proiect
4 where id_functie like '%IT%';
```

Script Output | Query Result | All Rows Fetched: 1 in 0.014 seconds

	ANGAJATI IT
1	10

--Se va afisa suma salariilor tuturor angajatilor din departamentul HR folosind SUM

```
select sum (salariu)
```

```
"SALARIILE HR"
```

```
from angajati_proiect
```

```
where id_functie like '%HR%';
```

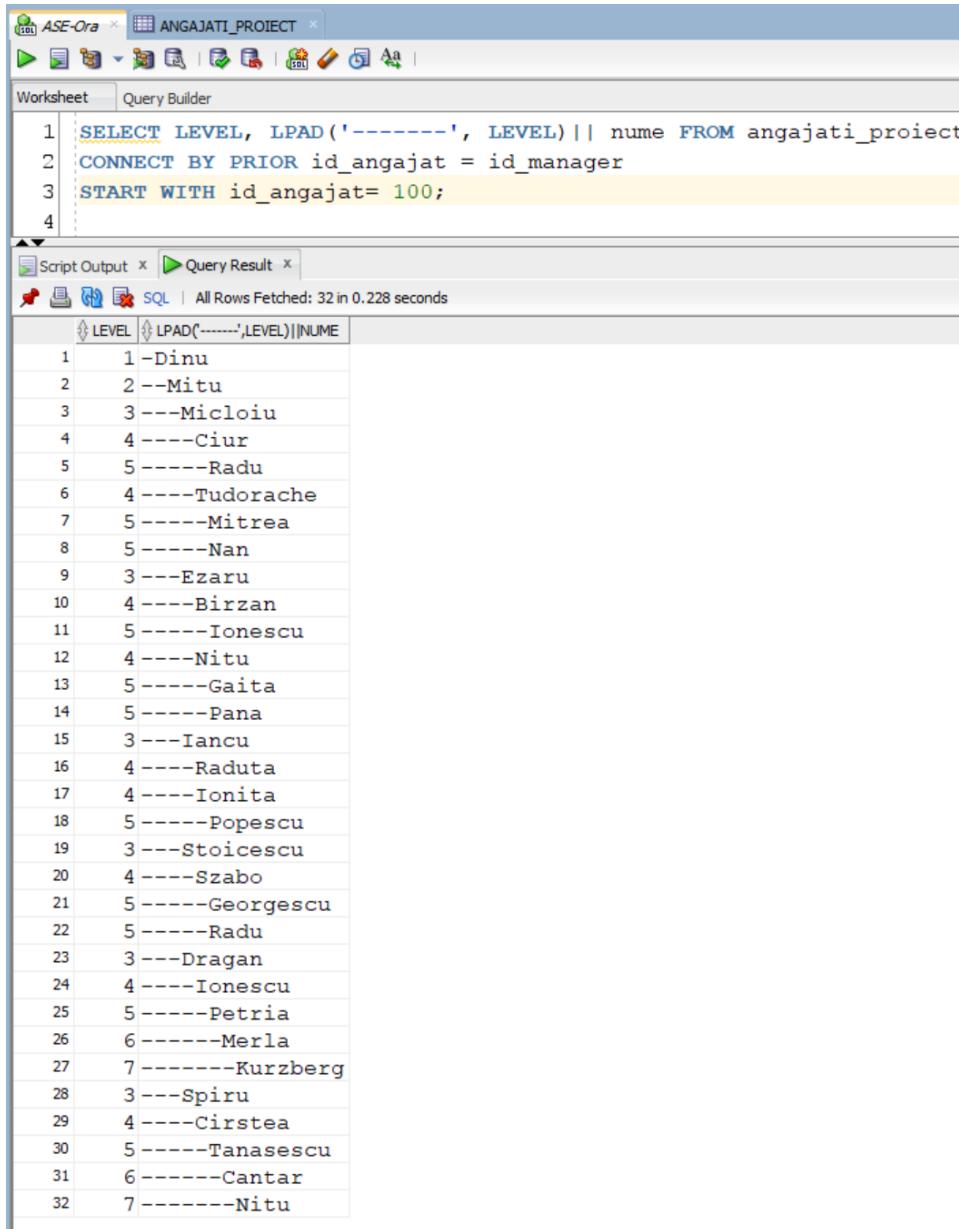
```
1 select sum (salariu)
2 "SALARIILE HR"
3 from angajati_proiect
4 where id_functie like '%HR%';
```

Script Output | Query Result | All Rows Fetched: 1 in 0.018 seconds

	SALARIILE HR
1	31500

--Parcurgerea structurilor ierarhice--Parcurgere TOP-DOWN

```
SELECT LEVEL, LPAD('-----', LEVEL)|| nume FROM angajati_proiect
CONNECT BY PRIOR id_angajat = id_manager
START WITH id_angajat= 100;
```



The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there are two tabs: 'ASE-Ora' and 'ANGAJATI\_PROJECT'. Below them is a toolbar with various icons. The main area has two tabs: 'Worksheet' and 'Query Builder', with 'Worksheet' currently selected. The code in the worksheet is:

```
1 | SELECT LEVEL, LPAD('-----', LEVEL)|| nume FROM angajati_proiect
2 | CONNECT BY PRIOR id_angajat = id_manager
3 | START WITH id_angajat= 100;
4 |
```

Below the code, the 'Query Result' tab is open, showing the output of the query. The results are presented in a table with two columns: 'LEVEL' and 'LPAD('-----',LEVEL)||NUME'. The data is as follows:

LEVEL	LPAD('-----',LEVEL)  NUME
1	1 -Dinu
2	2 --Mitu
3	3 ---Micloiu
4	4 ----Ciur
5	5 -----Radu
6	4 ----Tudorache
7	5 -----Mitrea
8	5 -----Nan
9	3 ---Ezaru
10	4 ----Birzan
11	5 -----Ionescu
12	4 ----Nitu
13	5 -----Gaita
14	5 -----Pana
15	3 ---Iancu
16	4 ----Raduta
17	4 ----Ionita
18	5 -----Popescu
19	3 ---Stoicescu
20	4 ----Szabo
21	5 -----Georgescu
22	5 -----Radu
23	3 ---Dragan
24	4 ----Ionescu
25	5 -----Petria
26	6 -----Merla
27	7 -----Kurzberg
28	3 ---Spiru
29	4 ----Cirstea
30	5 -----Tanasescu
31	6 -----Cantar
32	7 -----Nitu

At the bottom of the results window, it says 'All Rows Fetched: 32 in 0.228 seconds'.

--Parcurgere BOTTOM-UP

```
SELECT id_angajat, nume, id_manager, LEVEL FROM angajati_proiect
CONNECT BY id_angajat = PRIOR id_manager
START WITH nume= 'Kurzberg'
ORDER BY LEVEL;
```

ID_ANGAJAT	NUME	ID_MANAGER	LEVEL
1	127 Kurzberg	126	1
2	126 Merla	125	2
3	125 Petria	124	3
4	124 Ionescu	122	4
5	122 Dragan	101	5
6	101 Mitu	100	6
7	100 Dinu	(null)	7

--Functia DECODE

--Se vor afisa numele, functia si salariul cu un bonus de 250 daca angajatul este din departamentul IT sau 150 daca angajatul este din departamentul Imagine si Promovare

```
select nume, prenume, id_functie,
(case
when id_functie like 'IT%' then 250
when id_functie like 'IP%' then 150
else 0
end) + salariu "SALARIU CU BONUS"
from angajati_proiect;
```

ASE-Ora > ANGAJATI\_PROJECT >

Worksheet    Query Builder

```

1 | select nume, prenume, id_functie,
2 |   (case
3 |     when id_functie like 'IT%' then 250
4 |     when id_functie like 'IP%' then 150
5 |     else 0
6 |   end) + salariu "SALARIU CU BONUS"
7 |   from angajati_proiect;
8 |

```

Script Output x | Query Result x

All Rows Fetched: 32 in 0.017 seconds

	NUME	PRENUME	ID_FUNCTIE	SALARIU CU BONUS
5	Tudorache	Andrei	FR_ANG	5100
6	Radu	Catalina	FR_ANG	4500
7	Mitrea	Bogdan	FR_ANG	4600
8	Nan	Stefan	FR_ANG	4400
9	Ezaru	Radu	HR_MAN	7600
10	Birzan	Adelin	HR_ANG	5100
11	Nitu	Oana	HR_ANG	5100
12	Ionescu	Vlad	HR_ANG	4700
13	Gaita	Stefan	HR_ANG	4500
14	Pana	Adrian	HR_ANG	4500
15	Iancu	Andrei	IP_MAN	8050
16	Stoicescu	Stefan	IP_MAN	7750
17	Raduta	Andrei	IP_ANG	5250
18	Ionita	Maria	IP_ANG	5150
19	Popescu	Mara	IP_ANG	4850
20	Szabo	Alexandru	IP_ANG	5250
21	Georgescu	Cosmina	IP_ANG	5050
22	Radu	Steluta	IP_ANG	4950
23	Dragan	Andrei	IT_MAN	8250
24	Spiru	Andrei	IT_MAN	8050
25	Ionescu	Andrei	IT_ANG	5750
26	Petria	Cristian	IT_ANG	5650
27	Merla	Antoniu	IT_ANG	5550
28	Kurzberg	Sarah	IT_ANG	5450
29	Cirstea	Andrei	IT_ANG	5750
30	Tanasescu	Alexandru	IT_ANG	5650
31	Cantar	Nicholas	IT_ANG	5550
32	Nitu	Vlad	IT_ANG	5450

--Functia CASE (aceeasi cerinta ca la DECODE)

```
select nume, prenume, id_functie,
decode(id_functie, 'IT_ANG', 250, 'IT_MAN', 250,
'IP_ANG', 150, 'IP_MAN', 150, 0)
+ salariu "SALARIU CU BONUS"
from angajati_proiect;
```

--Functiile UNION, INTERSECT, MINUS--Se vor afisa angajatii din departamentul IT care nu sunt manageri

```
select * from angajati_proiect where id_functie like 'IT%'
minus
select * from angajati_proiect where id_functie like '%MAN';
```

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, the title bar says 'ASE-Ora ANGAJATI\_PROJECT'. Below it, there are several icons. The 'Worksheet' tab is selected. In the main area, there is a code editor window containing the following SQL query:

```
1 select * from angajati_proiect where id_functie like 'IT%'
2 minus
3 select * from angajati_proiect where id_functie like '%MAN';
```

Below the code editor is a toolbar with icons for script output, query result, and other functions. The status bar at the bottom indicates 'All Rows Fetched: 8 in 0.017 seconds'. The 'Query Result' tab is active, displaying the following table:

ID_ANGAJAT	NUME	PRENUME	EMAIL	TELEFON	DATA_NASTERE	DATA_ANGAJARE	SALARIU	ID_FUNCTIE	ID_DEPARTAMENT	ID_MANAGER
1	124	Ionescu	Andrei	andreiionescu@gmail.com	0721.340.848	01-DEC-02 02-MAR-21	5500	IT_ANG	50	122
2	125	Petria	Cristian	cristianpetria@gmail.com	0725.207.881	12-OCT-02 12-APR-21	5400	IT_ANG	50	124
3	126	Merla	Antoniu	antoniumerla@gmail.com	0729.959.422	13-DEC-01 20-MAY-21	5300	IT_ANG	50	125
4	127	Kurzberg	Sarah	sarahkurzberg@yahoo.ro	0758.357.483	18-JUN-02 24-JUN-21	5200	IT_ANG	50	126
5	128	Cirstea	Andrei	andreicirstea@yahoo.ro	0727.940.042	04-JUL-02 10-MAR-21	5500	IT_ANG	50	123
6	129	Tanasescu	Alexandru	alexantanasescu@gmail.com	0726.031.135	21-MAR-02 13-APR-21	5400	IT_ANG	50	128
7	130	Cantar	Nicholas	nicholascantar@gmail.com	0737.142.246	20-DEC-02 01-MAY-21	5300	IT_ANG	50	129
8	131	Nitu	Vlad	vladnitu@gmail.com	0748.253.357	16-JAN-02 30-JUN-21	5200	IT_ANG	50	130

--Exemplu de tabela virtuala

```

CREATE OR REPLACE VIEW v_angajati_proiect_10
AS SELECT * FROM angajati_proiect
WHERE id_departament = 10
WITH READ ONLY;
SELECT * FROM v_angajati_proiect_10;
UPDATE v_angajati_proiect_10
SET salariu = salariu + 1000;

```

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, the connection is set to ASE-Ora and the schema is ANGAJATI\_PROJECT. The main area is a Worksheet tab where the following SQL code is entered:

```

1 CREATE OR REPLACE VIEW v_angajati_proiect_10
2 AS SELECT * FROM angajati_proiect
3 WHERE id_departament = 10
4 WITH READ ONLY;
5
6 SELECT * FROM v_angajati_proiect_10;

```

Below the code, there are two tabs: Script Output and Query Result. The Query Result tab shows the results of the SELECT query:

ID_ANGAJAT	NUME	PRENUME	EMAIL	TELEFON	DATA_NASTERE	DATA_ANGAJARE	SALARIU	ID_FUNCTIE	ID_DEPARTAMENT	ID_MANAGER	
1	100	Dinu	Ana	anadinu@gmail.com	0745.479.299	12-AUG-02	10-DEC-20	10100	AD-PRES	10	(null)
2	101	Mitu	Alexandru	alexmitu@gmail.com	0742.225.131	19-JUL-02	10-DEC-20	9100	AD-VP	10	100

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, the connection is set to ASE-Ora and the schema is ANGAJATI\_PROJECT. The main area is a Worksheet tab where the following SQL code is entered:

```

8 UPDATE v_angajati_proiect_10
9 SET salariu = salariu + 1000;

```

Below the code, there are two tabs: Script Output and Query Result. The Script Output tab displays the error message:

```

Error starting at line : 8 in command -
UPDATE v_angajati_proiect_10
SET salariu = salariu + 1000
Error at Command Line : 9 Column : 5
Error report -
SQL Error: ORA-42399: cannot perform a DML operation on a read-only view
42399.0000 - "cannot perform a DML operation on a read-only view"

Error starting at line : 8 in command -
UPDATE v_angajati_proiect_10
SET salariu = salariu + 1000
Error at Command Line : 9 Column : 5
Error report -
SQL Error: ORA-42399: cannot perform a DML operation on a read-only view
42399.0000 - "cannot perform a DML operation on a read-only view"

```

--Exemplu de index

```
select * from participanti_proiect
where prenume_participant = 'Tiberiu';
create index idx_pren on participanti_proiect(prenume_participant);
drop index idx_pren;
```

The screenshot shows the Oracle SQL Developer interface. In the top tab bar, it says "ASE-Ora > PARTICIPANTI\_PROJECT". The main area is a "Worksheet" containing the following SQL code:

```
1 | select * from participanti_proiect
2 | where prenume_participant = 'Tiberiu';
```

Below the code, there are tabs for "Script Output", "Autotrace", "Query Result", and "Explain Plan". The "Explain Plan" tab is selected, showing the following execution plan:

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	3
TABLE ACCESS	PARTICIPANTI_PROJECT	FULL	1	3
Filter Predicates	PRENUME_PARTICIPANT='Tiberiu'	OPTIONS=FULL		

The screenshot shows the Oracle SQL Developer interface. In the top tab bar, it says "ASE-Ora > PARTICIPANTI\_PROJECT". The main area is a "Worksheet" containing the following SQL code:

```
1 | select * from participanti_proiect
2 | where prenume_participant = 'Tiberiu';
3 |
4 | create index idx_pren on participanti_proiect(prenume_participant);
```

Below the code, there are tabs for "Script Output", "Autotrace", "Query Result", and "Explain Plan". The "Explain Plan" tab is selected, showing the following execution plan:

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
CREATE INDEX STATEMENT	IDX_PREN	NON UNIQUE		409
INDEX BUILD	IDX_PREN	CREATE INDEX		409
SORT	IDX_PREN	FAST FULL SCAN		
INDEX				

--Exemplu secenta

```
CREATE SEQUENCE seq_contrib
START WITH 250 INCREMENT BY 10
MAXVALUE 1500 NOCYCLE;
```

```
INSERT INTO contributii_proiect
VALUES (7, 4, 2, 'donatie', seq_contrib.nextval);
```

```
INSERT INTO contributii_proiect
VALUES (8, 5, 2, 'donatie', seq_contrib.nextval);
```

The screenshot shows the ASE-Ora SQL interface with the following details:

- Worksheet Tab:** Contains the following SQL code:
 

```

1 CREATE SEQUENCE seq_contrib
2 START WITH 250 INCREMENT BY 10
3 MAXVALUE 1500 NOCYCLE;
4
5 INSERT INTO contributii_proiect
6 VALUES (7, 4, 2, 'donatie', seq_contrib.nextval);
7
8 INSERT INTO contributii_proiect
9 VALUES (8, 5, 2, 'donatie', seq_contrib.nextval);
10
11 SELECT * FROM contributii_proiect;
```
- Script Output Tab:** Shows the results of the execution:
 

```
SQL | All Rows Fetched: 8 in 0.006 seconds
```

ID_CONTRIBUTIE	ID_PARTICIPANT	ID_PROIECT	TIP_CONTRIBUTIE	VALOARE_IN_BANI
1	1	1	1 jucarii	(null)
2	2	2	1 dulciuri	(null)
3	3	4	2 donatie	1000
4	4	5	2 donatie	1400
5	5	6	3 mancare si bauturi	(null)
6	6	7	3 donatie	800
7	7	4	2 donatie	250
8	8	5	2 donatie	260

**PL/SQL****Structuri de control. Gestionarea cursorilor.**

--Se incearca modificarea datelor din tabela CONTRIBUTII pe baza id-ului

--Folosind atributul cursorului implicit SQL%NOTFOUND in conditia structurii IF

--Mesajul afisat indica daca modificarea s-a realizat sau nu

```
BEGIN
```

```
    UPDATE contributii_proiect
```

```
    SET tip_contributie = 'jucarii'
```

```
    WHERE id_contributie = 20;
```

```
IF SQL%NOTFOUND THEN
```

```
    DBMS_OUTPUT.PUT_LINE('Nu exista contributie cu acest numar in tabela.');
```

```
ELSE
```

```
    DBMS_OUTPUT.PUT_LINE('Modificarea contributiei s-a realizat cu succes.');
```

```
END IF;
```

```
ROLLBACK;
```

```
END;
```

```
/
```

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, 'ASE-Ora' is selected. The main area has tabs for 'CONTRIBUTII\_PROJECT' and 'Query Builder'. The 'Worksheet' tab is active, displaying the following PL/SQL code:

```

1 SET SERVEROUTPUT ON
2
3 --Se incearca modificarea datelor din tabela CONTRIBUTII pe baza id-ului
4 --Folosind atributul cursorului implicit SQL%NOTFOUND in conditia structurii IF
5 --Mesajul afisat indica daca modificarea s-a realizat sau nu
6
7 BEGIN
8     UPDATE contributii_proiect
9     SET tip_contributie = 'jucarii'
10    WHERE id_contributie = 20;
11
12 IF SQL%NOTFOUND THEN
13     DBMS_OUTPUT.PUT_LINE('Nu exista contributie cu acest numar in tabela.');
14 ELSE
15     DBMS_OUTPUT.PUT_LINE('Modificarea contributiei s-a realizat cu succes.');
16 END IF;
17 ROLLBACK;
18
19 /

```

In the bottom 'Script Output' tab, the results of the execution are shown:

```

Script Output X
Task completed in 0.156 seconds
Nu exista contributie cu acest numar in tabela.

PL/SQL procedure successfully completed.

```

```

1 SET SERVEROUTPUT ON
2
3 --Se incearca modificarea datelor din tabela CONTRIBUTII pe baza id-ului
4 --Folosind atributul cursorului implicit SQL%NOTFOUND in conditia structurii IF
5 --Mesajul afisat indica daca modificararea s-a realizat sau nu
6
7 BEGIN
8   UPDATE contributii_proiect
9     SET tip_contributie = 'jucarii'
10    WHERE id_contributie = 2;
11
12 IF SQL%NOTFOUND THEN
13   DBMS_OUTPUT.PUT_LINE('Nu exista contributie cu acest numar in tabela.');
14 ELSE
15   DBMS_OUTPUT.PUT_LINE('Modificarea contributiei s-a realizat cu succes.');
16 END IF;
17 ROLLBACK;
18 END;
19 /

```

Script Output X | Task completed in 0.052 seconds

Modificarea contributiei s-a realizat cu succes.

PL/SQL procedure successfully completed.

--Se va efectua o marire salariala de 100 celor care au salariul sub 5000

--Se va afisa cati dintre cei care au salariul peste 5000 se incadreaza aici dupa marire

DECLARE

```

v_nrang NUMBER := 0;
nr_salarii_mari NUMBER := 0;
v_aux angajati_proiect.salariu%TYPE;
```

BEGIN

```

SELECT salariu INTO v_aux
FROM angajati_proiect
WHERE salariu < 5000 AND (salariu + 100) >= 5000;
```

```
v_nrang := SQL%ROWCOUNT;
```

```
UPDATE angajati_proiect
SET salariu = salariu + 100
WHERE salariu < 5000;

DBMS_OUTPUT.PUT_LINE('A fost acordata o marire salariala de 100 celor care aveau sub
5000.');
```

```
FOR a IN (SELECT * FROM angajati_proiect
           WHERE salariu >= 5000)
LOOP
    DBMS_OUTPUT.PUT_LINE('Nume: ' || a.nume || '; Salariu: ' || a.salariu);
    nr_salarii_mari := nr_salarii_mari + 1;
END LOOP;
```

```
CASE
    WHEN nr_salarii_mari = 0 THEN
        DBMS_OUTPUT.PUT_LINE('Nu a fost gasit niciun angajat cu salariul cel putin 5000.');
    WHEN nr_salarii_mari = 1 THEN
        DBMS_OUTPUT.PUT_LINE('A fost gasit un singur angajat cu salariul cel putin 5000.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Au fost gasiti ' || nr_salarii_mari || ' angajati cu salariul cel
putin 5000, dintre care ' || v_nrang || ' dupa marire.');
    END CASE;
```

```
ROLLBACK;
```

```
END;
```

```
/
```

```

23
24 DECLARE
25   v_nrang NUMBER := 0;
26   nr_salarii_mari NUMBER := 0;
27   v_aux angajati_proiect.salariu%TYPE;
28
29   CURSOR c_ang IS SELECT nume, salariu, id_functie
30     FROM angajati_proiect
31   WHERE id_departament = 40;
32
33   ang_num angajati_proiect.nume%TYPE;
34   ang_sal angajati_proiect.salariu%TYPE;
35   ang_fct angajati_proiect.id_functie%TYPE;
36
37   BEGIN
38     DBMS_OUTPUT.PUT_LINE('Lista salariilor angajatilor din departamentul Imagine si');
39     DBMS_OUTPUT.PUT_LINE('Promovare:');
40
41     FOR i IN 1..v_nrang
42     LOOP
43       IF v_aux >= 5000 THEN
44         DBMS_OUTPUT.PUT_LINE(v_nrang || '. ' || v_aux);
45         v_nrang := v_nrang + 1;
46       END IF;
47     END LOOP;
48
49     DBMS_OUTPUT.PUT_LINE(nr_salarii_mari || ' au fost gasiti cu salariul cel putin 5000, dintre care ' ||
50                           v_nrang || ' au marirea de minim 5000.');
51   END;
52
53   v_nrang := 0;
54   nr_salarii_mari := 0;
55   v_aux := null;
56
57   PL/SQL procedure successfully completed.

```

--Afisare informatii despre angajatii din departament utilizand un cursor explicit

DECLARE

```

CURSOR c_ang IS SELECT nume, salariu, id_functie
  FROM angajati_proiect
 WHERE id_departament = 40;

```

```

ang_num angajati_proiect.nume%TYPE;
ang_sal angajati_proiect.salariu%TYPE;
ang_fct angajati_proiect.id_functie%TYPE;

```

BEGIN

```

DBMS_OUTPUT.PUT_LINE('Lista salariilor angajatilor din departamentul Imagine si');
DBMS_OUTPUT.PUT_LINE('Promovare:');

```

```

OPEN c_ang;
LOOP
  FETCH c_ang INTO ang_num, ang_sal, ang_fct;
  EXIT WHEN c_ang%NOTFOUND;
  DBMS_OUTPUT.PUT_LINE(ang_num || ' cu functia de ' || ang_fct || ' are salariul ' || ang_sal);
END LOOP;
CLOSE c_ang;

END;
/

```

The screenshot shows the Oracle SQL Developer interface with two main panes: a Worksheet pane and a Script Output pane.

**Worksheet pane:**

```

64 DECLARE
65   CURSOR c_ang IS SELECT nume, salariu, id_functie_
66     FROM angajati_proiect
67     WHERE id_departament = 40;
68
69   ang_num angajati_proiect.nume%TYPE;
70   ang_sal angajati_proiect.salariu%TYPE;
71   ang_fct angajati_proiect.id_functie%TYPE;
72
73 BEGIN
74   DBMS_OUTPUT.PUT_LINE('Lista salariilor angajatilor din departamentul Imagine si Promovare:');
75   OPEN c_ang;
76   LOOP
77     FETCH c_ang INTO ang_num, ang_sal, ang_fct;
78     EXIT WHEN c_ang%NOTFOUND;
79     DBMS_OUTPUT.PUT_LINE(ang_num || ' cu functia de ' || ang_fct || ' are salariul ' || ang_sal);
80   END LOOP;
81   CLOSE c_ang;
82
83 END;
84 /

```

**Script Output pane:**

```

Lista salariilor angajatilor din departamentul Imagine si Promovare:
Iancu cu functia de IP_MAN are salariul 7900
Stoicescu cu functia de IP_MAN are salariul 7600
Raduta cu functia de IP_ANG are salariul 5100
Ionita cu functia de IP_ANG are salariul 5000
Popescu cu functia de IP_ANG are salariul 4700
Szabo cu functia de IP_ANG are salariul 5100
Georgescu cu functia de IP_ANG are salariul 4900
Radu cu functia de IP_ANG are salariul 4800

PL/SQL procedure successfully completed.

```

--Se vor afisa primele 2 proiecte cu cele mai multe contributii

DECLARE

```
CURSOR c IS SELECT pp.id_proiect, pp.denumire_proiect denumire,
COUNT(cp.id_contributie) nr_contributii

        FROM proiecte_proiect pp, contributii_proiect cp

        WHERE pp.id_proiect = cp.id_proiect

        GROUP BY pp.id_proiect, pp.denumire_proiect

        ORDER BY COUNT(cp.id_contributie) DESC

        FETCH FIRST 2 ROWS ONLY;
```

```
crow c%ROWTYPE;
```

BEGIN

```
    OPEN c;
```

```
LOOP
```

```
    FETCH c INTO crow;
```

```
    EXIT WHEN c%NOTFOUND;
```

```
    DBMS_OUTPUT.PUT_LINE('Proiectul ' || crow.denumire || ' are ' || crow.nr_contributii || '
contributii.');
```

```
END LOOP;
```

```
CLOSE c;
```

```
END;
```

```
/
```

```

88 10 DECLARE
89 11   CURSOR c IS SELECT pp.id_project, pp.denumire_project denumire, COUNT(cp.id_contributie) nr_contributii_
90 12     FROM proiecte_proiect pp, contributii_proiect cp
91 13     WHERE pp.id_proiect = cp.id_proiect
92 14     GROUP BY pp.id_project, pp.denumire_project
93 15     ORDER BY COUNT(cp.id_contributie) DESC
94 16     FETCH FIRST 2 ROWS ONLY;
95
96 17   crow c%ROWTYPE;
97
98 18 BEGIN
99 19   OPEN c;
100 20   LOOP
101 21     FETCH c INTO crow;
102 22     EXIT WHEN c%NOTFOUND;
103 23     DBMS_OUTPUT.PUT_LINE('Proiectul ' || crow.denumire || ' are ' || crow.nr_contributii || ' contributii.');
104 24   END LOOP;
105 25   CLOSE c;
106
107 26 END;
108 27 /

```

Script Output X | Task completed in 0.038 seconds

Proiectul Next In Tech are 4 contributii.  
Proiectul Entertainment Masters are 2 contributii.

PL/SQL procedure successfully completed.

--Se vor afisa informatii despre donatii si se va calcula suma totala de bani stransi

DECLARE

v\_total NUMBER := 0;

CURSOR c\_contrib IS SELECT id\_contributie, valoare\_in\_bani  
FROM contributii\_proiect;

BEGIN

FOR i IN c\_contrib

LOOP

IF i.valoare\_in\_bani IS NOT NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Donatia cu id-ul ' || i.id\_contributie || ' este in valoare de ' ||  
i.valoare\_in\_bani);

v\_total := v\_total + i.valoare\_in\_bani;

ELSE

DBMS\_OUTPUT.PUT\_LINE('Donatia cu id-ul ' || i.id\_contributie || ' nu este in bani.');

END IF;

END LOOP;

```
DBMS_OUTPUT.PUT_LINE('Suma de bani adunata din donatii este de ' || v_total);
```

```
END;
```

```
/
```

```

112 DECLARE
113     v_total NUMBER := 0;
114
115     CURSOR c_contrib IS SELECT id_contributie, valoare_in_bani_
116                             FROM contributii_proiect;
117
118 BEGIN
119     FOR i IN c_contrib
120     LOOP
121         IF i.valoare_in_bani IS NOT NULL THEN
122             DBMS_OUTPUT.PUT_LINE('Donatia cu id-ul ' || i.id_contributie || ' este in valoare de ' || i.valoare_in_bani);
123             v_total := v_total + i.valoare_in_bani;
124         ELSE
125             DBMS_OUTPUT.PUT_LINE('Donatia cu id-ul ' || i.id_contributie || ' nu este in bani.');
126         END IF;
127     END LOOP;
128
129     DBMS_OUTPUT.PUT_LINE('Suma de bani adunata din donatii este de ' || v_total);
130
131 END;
132

```

Script Output X | Task completed in 0.049 seconds

```

Donatia cu id-ul 1 nu este in bani.
Donatia cu id-ul 2 nu este in bani.
Donatia cu id-ul 3 este in valoare de 1000
Donatia cu id-ul 4 este in valoare de 1400
Donatia cu id-ul 5 nu este in bani.
Donatia cu id-ul 6 este in valoare de 800
Donatia cu id-ul 7 este in valoare de 250
Donatia cu id-ul 8 este in valoare de 260
Suma de bani adunata din donatii este de 3710

PL/SQL procedure successfully completed.

```

### Tratarea si gestionarea exceptiilor

--Se va afisa numele complet si departamentul angajatului al carui prenume este introdus de la tastatura

--Se vor trata cazurile in care sunt prea multi sau nu exista angajati prin exceptii

```
DECLARE
```

```

v_nume angajati_proiect.nume%TYPE;
v_dep departamente_proiect.denumire_departament%TYPE;
v_prenume angajati_proiect.prenume%TYPE := '&prenume';
prea_multi_angajati EXCEPTION;

```

```
PRAGMA EXCEPTION_INIT(prea_multi_angajati,-01422);

BEGIN
    SELECT a.nume, d.denumire_departament
    INTO v_nume, v_dep
    FROM angajati_proiect a, departamente_proiect d
    WHERE a.id_departament = d.id_departament
    AND a.prenume = v_prenume;
    DBMS_OUTPUT.PUT_LINE(v_prenume || ' ' || v_nume || ' face parte din departamentul ' ||
v_dep);

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista angajat.');
    WHEN prea_multi_angajati THEN
        DBMS_OUTPUT.PUT_LINE('Sunt mai multi angajati cu acest nume.');
        FOR v IN (SELECT a.nume as nume, d.denumire_departament as dep
                   FROM angajati_proiect a, departamente_proiect d
                   WHERE a.id_departament = d.id_departament
                   AND a.prenume = v_prenume)
        LOOP
            DBMS_OUTPUT.PUT_LINE(v_prenume || ' ' || v.nume || ' este din ' || v.dep);
        END LOOP;
    END;
/

```

comenzi\_pl-sql.sql ANGAJATI\_PROJECT

SQL Worksheet | History | 6.92799997 seconds | ASE-Ora

Worksheet | Query Builder

```

147   FROM angajati_proiect a, departamente_proiect d
148   WHERE a.id_departament = d.id_departament
149   AND a.prenume = v_prenume;
150   DBMS_OUTPUT.PUT_LINE(v_prenume || ' ' || v_nume || ' face parte din departamental ' || v_dep);
151
152 EXCEPTION
153   WHEN NO_DATA_FOUND THEN
154     DBMS_OUTPUT.PUT_LINE('Nu exista angajat.');
155   WHEN prea_multi_angajati THEN
156     DBMS_OUTPUT.PUT_LINE('Sunt mai multi angajati cu acest nume.');
157     FOR v IN (SELECT a.nume as nume, d.denumire_departament as dep_
158       FROM angajati_proiect a, departamente_proiect d
159       WHERE a.id_departament = d.id_departament
160       AND a.prenume = v_prenume)
161     LOOP
162       DBMS_OUTPUT.PUT_LINE(v_prenume || ' ' || v.nume || ' este din ' || v.dep);
163     END LOOP;
164 END;
165 /
166
167

```

Script Output X | Task completed in 6.928 seconds

Sunt mai multi angajati cu acest nume.  
Andrei Tudorache este din Financiar  
Andrei Raduta este din Imagine si Promovare  
Andrei Iancu este din Imagine si Promovare  
Andrei Ionescu este din IT  
Andrei Spiru este din IT  
Andrei Cirstea este din IT  
Andrei Dragan este din IT

PL/SQL procedure successfully completed.

comenzi\_pl-sql.sql ANGAJATI\_PROJECT

SQL Worksheet | History | 4.67 seconds | ASE-Ora

Worksheet | Query Builder

```

147   FROM angajati_proiect a, departamente_proiect d
148   WHERE a.id_departament = d.id_departament
149   AND a.prenume = v_prenume;
150   DBMS_OUTPUT.PUT_LINE(v_prenume || ' ' || v_nume || ' face parte din departamental ' || v_dep);
151
152 EXCEPTION
153   WHEN NO_DATA_FOUND THEN
154     DBMS_OUTPUT.PUT_LINE('Nu exista angajat.');
155   WHEN prea_multi_angajati THEN
156     DBMS_OUTPUT.PUT_LINE('Sunt mai multi angajati cu acest nume.');
157     FOR v IN (SELECT a.nume as nume, d.denumire_departament as dep_
158       FROM angajati_proiect a, departamente_proiect d
159       WHERE a.id_departament = d.id_departament
160       AND a.prenume = v_prenume)
161     LOOP
162       DBMS_OUTPUT.PUT_LINE(v_prenume || ' ' || v.nume || ' este din ' || v.dep);
163     END LOOP;
164 END;
165 /
166
167

```

Script Output X | Task completed in 4.67 seconds

FROM angajati\_proiect a, departamente\_proiect d  
WHERE a.id\_departament = d.id\_departament  
AND a.prenume = v\_prenume)

LOOP  
DBMS\_OUTPUT.PUT\_LINE(v\_prenume || ' ' || v.nume || ' este din ' || v.dep);  
END LOOP;

END;  
Nu exista angajat.

PL/SQL procedure successfully completed.

--Se vor majora salariile care au valoarea sub un anumit prag

--In caz contrar, se va gestiona printr-o exceptie

DECLARE

nu\_exista\_angajati EXCEPTION;

BEGIN

UPDATE angajati\_proiect

SET salariu = salariu + 150

WHERE salariu < &prag\_salariu\_minim;

IF SQL%FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('S-au modificat ' || SQL%ROWCOUNT || ' salarii');

ELSE

RAISE nu\_exista\_angajati;

END IF;

ROLLBACK;

EXCEPTION

WHEN nu\_exista\_angajati THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista angajati cu salariul sub prag.');

END;

/

```

170 DECLARE
171   nu_exista_angajati EXCEPTION;
172
173 BEGIN
174   UPDATE angajati_proiect
175   SET salariu = salariu + 150
176   WHERE salariu < &prag_salariu_minim;
177
178 IF SQL%FOUND THEN
179   DBMS_OUTPUT.PUT_LINE('S-au modificat ' || SQL%ROWCOUNT || ' salarii');
180 ELSE
181   RAISE nu_exista_angajati;
182 END IF;
183 ROLLBACK;
184
185 EXCEPTION
186   WHEN nu_exista_angajati THEN
187     DBMS_OUTPUT.PUT_LINE('Nu exista angajati cu salariul sub prag.');
188
189 END;
190

```

Script Output: Task completed in 4.695 seconds

```

ROLLBACK;

EXCEPTION
  WHEN nu_exista_angajati THEN
    DBMS_OUTPUT.PUT_LINE('Nu exista angajati cu salariul sub prag.');

END;
S-au modificat 9 salarii

PL/SQL procedure successfully completed.

```

--Se vor afisa ultimii 5 angajati in functie de data angajarii

--Vom primi eroare, deoarece utilizand FOR se incearca deschiderea cursorului de 2 ori

DECLARE

```

CURSOR c IS SELECT nume, prenume, data_angajare
  FROM angajati_proiect
  ORDER BY data_angajare DESC
  FETCH FIRST 5 ROWS ONLY;

```

BEGIN

OPEN c;

FOR r IN c LOOP

```
DBMS_OUTPUT.PUT_LINE(r.prenume || ' ' || r.nume || ' e angajat din ' || r.data_angajare);
```

END LOOP;

## EXCEPTION

```

    WHEN CURSOR_ALREADY_OPEN THEN
        DBMS_OUTPUT.PUT_LINE('Cursorul este deja deschis!');

END;
/

```

```

193 --Vom primi eroare, deoarece utilizand FOR se incerca deschiderea cursorului de 2 ori
194
195 DECLARE
196     CURSOR c IS SELECT nume, prenume, data_angajare...
197         FROM angajati_proiect
198         ORDER BY data_angajare DESC
199         FETCH FIRST 5 ROWS ONLY;
200
201 BEGIN
202     OPEN c;
203     FOR r IN c LOOP
204         DBMS_OUTPUT.PUT_LINE(r.prenume || ' ' || r.nume || ' e angajat din ' || r.data_angajare);
205     END LOOP;
206
207 EXCEPTION
208     WHEN CURSOR_ALREADY_OPEN THEN
209         DBMS_OUTPUT.PUT_LINE('Cursorul este deja deschis!');
210
211 END;
212 /

```

Script Output X | Task completed in 0.044 seconds

Cursorul este deja deschis!

PL/SQL procedure successfully completed.

--Varianta corecta este fara deschiderea cursorului prin comanda OPEN c

```

DECLARE
    CURSOR c IS SELECT nume, prenume, data_angajare
        FROM angajati_proiect
        ORDER BY data_angajare DESC
        FETCH FIRST 5 ROWS ONLY;

```

```

BEGIN
  FOR r IN c LOOP
    DBMS_OUTPUT.PUT_LINE(r.prenume || ' ' || r.nume || ' e angajat din ' || r.data_angajare);
  END LOOP;

EXCEPTION
  WHEN CURSOR_ALREADY_OPEN THEN
    DBMS_OUTPUT.PUT_LINE('Cursorul este deja deschis!');

END;
/

```

The screenshot shows the Oracle SQL Developer interface with a query worksheet titled 'comenzi\_pl-sql.sql'. The code in the worksheet is as follows:

```

213
214 --Varianta corecta este fara deschiderea cursorului prin comanda OPEN c
215
216 DECLARE
217   CURSOR c IS SELECT nume, prenume, data_angajare_
218   FROM angajati_proiect
219   ORDER BY data_angajare DESC
220   FETCH FIRST 5 ROWS ONLY;
221
222 BEGIN
223   FOR r IN c LOOP
224     DBMS_OUTPUT.PUT_LINE(r.prenume || ' ' || r.nume || ' e angajat din ' || r.data_angajare);
225   END LOOP;
226
227 EXCEPTION
228   WHEN CURSOR_ALREADY_OPEN THEN
229     DBMS_OUTPUT.PUT_LINE('Cursorul este deja deschis!');
230
231 END;
232 /

```

The output window below the worksheet shows the results of the execution:

```

PL/SQL procedure successfully completed.

Vlad Nitu e angajat din 30-JUN-21
Sarah Kurzberg e angajat din 24-JUN-21
Antoniu Merla e angajat din 20-MAY-21
Nicholas Cantar e angajat din 01-MAY-21
Alexandru Tanasescu e angajat din 13-APR-21

PL/SQL procedure successfully completed.

```

--Se va compara suma valorilor contributiilor cu o suma prag  
--Daca fondurile nu sunt suficiente, se va gestiona printr-o exceptie

DECLARE

```
v_contributii_cash NUMBER;
CURSOR c IS SELECT id_contributie, valoare_in_bani
    FROM contributii_proiect;
fonduri_insuficiente EXCEPTION;
v_id contributii_proiect.id_contributie%TYPE;
v_valoare contributii_proiect.valoare_in_bani%TYPE;
suma_prag NUMBER := &suma_prag;
```

BEGIN

```
v_contributii_cash := 0;
OPEN c;
LOOP
    FETCH c INTO v_id, v_valoare;
    EXIT WHEN c%NOTFOUND;
    v_contributii_cash := v_contributii_cash + NVL(v_valoare, 0);
END LOOP;
```

```
CLOSE c;
IF v_contributii_cash < suma_prag THEN
```

```
    RAISE fonduri_insuficiente;
ELSE
    DBMS_OUTPUT.PUT_LINE('Fonduri suficiente');
END IF;
```

EXCEPTION

```
WHEN fonduri_insuficiente THEN
    DBMS_OUTPUT.PUT_LINE('Valoarea contributiilor este o suma prea mica');
END;
/
```

```

245 BEGIN
246     v_contributii_cash := 0;
247     OPEN c;
248     LOOP
249         FETCH c INTO v_id, v_valoare;
250         EXIT WHEN c%NOTFOUND;
251         v_contributii_cash := v_contributii_cash + NVL(v_valoare, 0);
252     END LOOP;
253     CLOSE c;
254     IF v_contributii_cash < suma_prag THEN
255         RAISE fonduri_insuficiente;
256     ELSE
257         DBMS_OUTPUT.PUT_LINE('Fonduri suficiente');
258     END IF;
259
260     EXCEPTION
261     WHEN fonduri_insuficiente THEN
262         DBMS_OUTPUT.PUT_LINE('Valoarea contributiilor este o suma prea mica');
263     END;
264
265 END;
266

```

Script Output X | Task completed in 15.759 seconds

```

EXCEPTION
  WHEN fonduri_insuficiente THEN
    DBMS_OUTPUT.PUT_LINE('Valoarea contributiilor este o suma prea mica');

END;
Valoarea contributiilor este o suma prea mica

PL/SQL procedure successfully completed.

```

## Pachete. Functii. Proceduri.

CREATE OR REPLACE PACKAGE pachet\_proceduri\_proiect AS

--Afisare detalii despre angajatii dintr-un departament

PROCEDURE Afisare\_angajati\_departament(departament\_id IN ANGAJATI\_PROIECT.id\_departament%TYPE);

--Valoarea totala a contributiilor unui participant

PROCEDURE Valoare\_contributii (p\_id NUMBER);

--Afisarea tuturor angajatilor in ordinea cronologica a angajarilor

PROCEDURE Ordine\_angajari;

END;

/

CREATE OR REPLACE PACKAGE BODY pachet\_proceduri\_proiect AS

PROCEDURE Afisare\_angajati\_departament(departament\_id IN  
ANGAJATI\_PROIECT.id\_departament%TYPE)

AS

v\_count NUMBER;

CURSOR c\_angajati IS

SELECT nume, email, telefon, data\_nastere, salariu, id\_functie  
FROM ANGAJATI\_PROIECT  
WHERE id\_departament = departament\_id;

BEGIN

SELECT COUNT(\*)  
INTO v\_count  
FROM DEPARTAMENTE\_PROIECT  
WHERE id\_departament = departament\_id;

IF v\_count = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista departamentul'||departament\_id);  
RETURN;

END IF;

DBMS\_OUTPUT.PUT\_LINE('Detalii angajati departament'||departament\_id);

FOR angajat IN c\_angajati LOOP

DBMS\_OUTPUT.PUT\_LINE('Nume: '|angajat.nume);  
DBMS\_OUTPUT.PUT\_LINE('Email: '|angajat.email);  
DBMS\_OUTPUT.PUT\_LINE('Telefon: '|angajat.telefon);  
DBMS\_OUTPUT.PUT\_LINE('Data nasterii: '|angajat.data\_nastere);

```

DBMS_OUTPUT.PUT_LINE('Salariu: '||angajat.salariu);
DBMS_OUTPUT.PUT_LINE('Functie: '||angajat.id_functie);
DBMS_OUTPUT.PUT_LINE('-----');
END LOOP;

```

EXCEPTION

```

WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Nu exista angajati in departamentul '||departament_id);
END;

```

```

PROCEDURE Valoare_contributii (p_id NUMBER)
AS
    nr_contrib NUMBER;
    nu_sunt_contributii EXCEPTION;
    valoare_contributii NUMBER := 0;

```

```

BEGIN
    SELECT COUNT(*)
    INTO nr_contrib
    FROM contributii_proiect
    WHERE id_participant = p_id;

```

```

    IF nr_contrib = 0 THEN
        RAISE nu_sunt_contributii;
    END IF;

```

```

FOR c IN (SELECT id_contributie, valoare_in_bani FROM contributii_proiect WHERE
id_participant = p_id) LOOP

```

```
    valoare_contributii := valoare_contributii + NVL(c.valoare_in_bani, 0);
END LOOP;

DBMS_OUTPUT.PUT_LINE('Valoarea contributiilor pentru participantul ' || p_id || ' este ' ||
valoare_contributii);
```

EXCEPTION

```
WHEN nu_sunt_contributii THEN
    DBMS_OUTPUT.PUT_LINE('Nu exista participant cu acest id!');
```

END;

PROCEDURE Ordine\_angajari

AS

```
CURSOR c IS SELECT a.nume, a.prenume, a.data_angajare, d.denumire_departament
    FROM angajati_proiect a, departamente_proiect d
    WHERE a.id_departament = d.id_departament
    ORDER BY data_angajare, a.id_angajat;
r c%ROWTYPE;
```

BEGIN

```
    DBMS_OUTPUT.PUT_LINE('Ordinea angajarilor:');
```

```
    OPEN c;
```

LOOP

```
    FETCH c INTO r;
```

```
    EXIT WHEN c%NOTFOUND;
```

```
    DBMS_OUTPUT.PUT_LINE(r.nume || ' ' || r.prenume || ' *** ' || r.denumire_departament
    || ' *** ' || r.data_angajare);
```

```

END LOOP;

CLOSE c;

END;

```

```
END;
```

```
/
```

### --Apel proceduri din pachetul de proceduri

```

EXECUTE pachet_proceduri_proiect.Afisare_angajati_departament(20);

/

```

The screenshot shows the Oracle SQL Developer interface. The top window displays the PL/SQL code:

```

293    DBMS_OUTPUT.PUT_LINE('Email: '||angajat.email);
294    DBMS_OUTPUT.PUT_LINE('Telefon: '||angajat.telefon);
295    DBMS_OUTPUT.PUT_LINE('Data nasterii: '||angajat.data_nastere);
296    DBMS_OUTPUT.PUT_LINE('Salariu: '||angajat.salariu);
297    DBMS_OUTPUT.PUT_LINE('Functie: '||angajat.id_functie);
298    DBMS_OUTPUT.PUT_LINE('-----');
299
300 EXCEPTION
301
302   WHEN NO_DATA_FOUND THEN
303     DBMS_OUTPUT.PUT_LINE('Nu exista angajati in departamentul '||departament_id);
304 END;
305 /
306
307 --apel procedura--
308
309 EXECUTE Afisare_angajati_departament(20);
310 /

```

The bottom window shows the 'Script Output' tab with the results of the execution:

```

Detalii angajati departament 20
Nume: Micloiu
Email: dianamicloiu@gmail.com
Telefon: 0747.924.268
Data nasterii: 15-NOV-03
Salariu: 7600
Functie: FR_MAN
-----
Nume: Ciur
Email: anaciur@gmail.com
Telefon: 0758.035.379
Data nasterii: 06-OCT-02
Salariu: 5100
Functie: FR_ANG
-----
```

EXECUTE pachet\_proceduri\_proiect.Valoare\_contributii(4);

/

```

329: FOR c IN (SELECT id_contributie, valoare_in_bani FROM contributii_proiect WHERE id_participant = p_id) LOOP
330:   valoare_contributii := valoare_contributii + NVL(c.valoare_in_bani, 0);
331: END LOOP;
332:
333: DBMS_OUTPUT.PUT_LINE('Valoarea contributiilor pentru participantul ' || p_id || ' este ' || valoare_contributii);
334:
335: EXCEPTION
336:   WHEN nu_sunt_contributii THEN
337:     DBMS_OUTPUT.PUT_LINE('Nu exista participant cu acest id!');
338:
339: END;
340: /
341:
342: --apel--
343:
344: EXECUTE Valoare_contributii(4);
345: /
346:

```

Script Output:

```

Procedure VALORE_CONTRIBUTII compiled

Valoarea contributiilor pentru participantul 4 este 1250

PL/SQL procedure successfully completed.

Valoarea contributiilor pentru participantul 1 este 0

PL/SQL procedure successfully completed.

Nu exista participant cu acest id!

```

EXECUTE pachet\_proceduri\_proiect.Ordine\_angajari();

/

```

352:   FROM angajati_proiect a, departamente_proiect d
353:   WHERE a.id_departament = d.id_departament
354:   ORDER BY data_angajare, a.id_angajat;
355:
356:   r c%ROWTYPE;
357:
358: BEGIN
359:   DBMS_OUTPUT.PUT_LINE('Ordinea angajarilor:');
360:   OPEN c;
361:   LOOP
362:     FETCH c INTO r;
363:     EXIT WHEN cNOTFOUND;
364:     DBMS_OUTPUT.PUT_LINE(r.nume || ' ' || r.prenume || ' ' || r.denumire_departament || ' ' || r.data_angajare);
365:   END LOOP;
366:   CLOSE c;
367:
368: END;
369: /
370:
371: --apel--
372: EXECUTE Ordine_angajari();
373:

```

Script Output:

```

Ordinea angajarilor:
Dinu Ana *** Administrativ *** 10-DEC-20
Mitu Alexandru *** Administrativ *** 10-DEC-20
Micloiu Diana *** Financiar *** 12-DEC-20
Ezaru Radu *** Resurse Umane *** 13-DEC-20
Iancu Andrei *** Imagine si Promovare *** 15-DEC-20
Stoicescu Stefan *** Imagine si Promovare *** 15-DEC-20
Dragan Andrei *** IT *** 17-DEC-20
Spiru Andrei *** IT *** 17-DEC-20
Ciur Ana *** Financiar *** 04-JAN-21
Tudorache Andrei *** Financiar *** 04-JAN-21
Birzan Adelina *** Resurse Umane *** 10-JAN-21

```

CREATE OR REPLACE PACKAGE pachet\_functii\_proiect AS

--Se afiseaza contributiile unui participant sub forma de lista, oferind ca parametru id-ul participantului

FUNCTION Afisare\_contributii (p\_id IN contributii\_proiect.id\_participant%TYPE) RETURN VARCHAR2;

--Verificare daca angajatul cu id-ul introdus are sau nu 2 ani vechime

FUNCTION Verificare\_vechime (p\_id angajati\_proiect.id\_angajat%TYPE) RETURN BOOLEAN;

--Afisare detalii despre proiectul cu id-ul introdus

FUNCTION Detalii\_proiect (p\_id proiecte\_proiect.id\_proiect%TYPE) RETURN VARCHAR2;

END;

/

CREATE OR REPLACE PACKAGE BODY pachet\_functii\_proiect AS

FUNCTION Afisare\_contributii (p\_id IN contributii\_proiect.id\_participant%TYPE) RETURN VARCHAR2

IS

nr\_contrib NUMBER;

nu\_sunt\_contributii EXCEPTION;

v\_denum VARCHAR2(100) := ";

BEGIN

SELECT COUNT(\*)

INTO nr\_contrib

FROM contributii\_proiect

WHERE id\_participant = p\_id;

```

IF nr_contrib = 0 THEN
    RAISE nu_sunt_contributii;
END IF;

FOR c IN (SELECT tip_contributie FROM contributii_proiect WHERE id_participant =
p_id) LOOP
    v_denum := v_denum || c.tip_contributie || ';';
END LOOP;
RETURN v_denum;

EXCEPTION
    WHEN nu_sunt_contributii THEN RETURN 'Nu exista participant!';
END;

FUNCTION Verificare_vechime (p_idang angajati_proiect.id_angajat%TYPE) RETURN
BOOLEAN
IS
    v_vechime NUMBER;
BEGIN
    SELECT TRUNC((SYSDATE-data_angajare)/365) INTO v_vechime
    FROM angajati_proiect
    WHERE id_angajat = p_idang;
    IF v_vechime >= 2 THEN
        RETURN TRUE;
    ELSE
        RETURN FALSE;
    END IF;

END;

```

```

FUNCTION Detalii_proiect (p_id proiecte_proiect.id_proiect%TYPE) RETURN
VARCHAR2
IS
nr_proiecte NUMBER;
nu_sunt_proiecte EXCEPTION;
v_det VARCHAR2(200) := "";

BEGIN
SELECT COUNT(*)
INTO nr_proiecte
FROM proiecte_proiect
WHERE id_proiect = p_id;

IF nr_proiecte = 0 THEN
RAISE nu_sunt_proiecte;
END IF;

FOR c IN (SELECT denumire_proiect, descriere FROM proiecte_proiect WHERE
id_proiect = p_id) LOOP
v_det := v_det || c.denumire_proiect || ' - ' || c.descriere;
END LOOP;

RETURN v_det;

EXCEPTION
WHEN nu_sunt_proiecte THEN RETURN 'Nu exista proiect!';

END;
/

```

--Apel functii din pachetul de functii

```

DECLARE
    v_denum VARCHAR2(100);

BEGIN
    v_denum := pachet_functii_proiect.Afisare_contributii(6);
    IF v_denum = 'Nu există participant!' THEN
        DBMS_OUTPUT.PUT_LINE('Nu există participant!');
    ELSE
        DBMS_OUTPUT.PUT_LINE(v_denum);
    END IF;
END;
/

```

```

comenzi_plsql.sql  CONTRIBUTII_PROJECT
SQL Worksheet | History
Worksheet | Query Builder
ASE-Ora
398
399 EXCEPTION
400   WHEN nu_sunt_contributii THEN RETURN 'Nu există participant!';
401
402 END;
403 /
404
405 --apel--
406
407 DECLARE
408   v_denum VARCHAR2(100);
409
410 BEGIN
411   v_denum := Afisare_contributii(4);
412   IF v_denum = 'Nu există participant!' THEN
413     DBMS_OUTPUT.PUT_LINE('Nu există participant!');
414   ELSE
415     DBMS_OUTPUT.PUT_LINE(v_denum);
416   END IF;
417 END;
418 /
419
Script Output X
donatie; donatie;

PL/SQL procedure successfully completed.

```

```

DECLARE
    v_id angajati_proiect.id_angajat%TYPE:=&v;

BEGIN
    DBMS_OUTPUT.PUT_LINE('Verificare angajat'||v_id);
    IF (pachet_functii_proiect.Verificare_vechime(v_id)) THEN
        DBMS_OUTPUT.PUT_LINE('Angajatul are cel putin 2 ani vechime');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Angajatul nu are 2 ani vechime');
    END IF;

END;
/

```

The screenshot shows the Oracle SQL Developer interface with a worksheet tab open. The code in the worksheet is:

```

434     ELSE
435         RETURN FALSE;
436     END IF;
437
438 END;
439 /
440
441 --apel--
442
443 DECLARE
444     v_id angajati_proiect.id_angajat%TYPE:=&v;
445
446 BEGIN
447     DBMS_OUTPUT.PUT_LINE('Verificare angajat'||v_id);
448     IF (Verificare_vechime(v_id)) THEN
449         DBMS_OUTPUT.PUT_LINE('Angajatul are cel putin 2 ani vechime');
450     ELSE
451         DBMS_OUTPUT.PUT_LINE('Angajatul nu are 2 ani vechime');
452     END IF;
453
454 END;
455

```

The output window below shows the results of the execution:

```

Script Output X
Task completed in 4.855 seconds
ELSE
    DBMS_OUTPUT.PUT_LINE('Angajatul nu are 2 ani vechime');
END IF;

END;
Verificare angajat 101
Angajatul are cel putin 2 ani vechime

PL/SQL procedure successfully completed.

```

```

DECLARE
    v_det VARCHAR2(200);
    v_id proiecte_proiect.id_proiect%TYPE:=&idproiect;

BEGIN
    v_det := pachet_functii_proiect.Detalii_proiect(v_id);
    IF v_det = 'Nu există proiect!' THEN
        DBMS_OUTPUT.PUT_LINE('Nu există proiect!');
    ELSE
        DBMS_OUTPUT.PUT_LINE(v_det);
    END IF;
END;
/

```

```

475
476 FOR c IN (SELECT denumire_proiect, descriere FROM proiecte_proiect WHERE id_proiect = p_id) LOOP
477     v_det := v_det || c.denumire_proiect || ' - ' || c.descriere;
478 END LOOP;
479
480 RETURN v_det;
481
482 EXCEPTION
483     WHEN nu_sunt_proiecte THEN RETURN 'Nu există proiect!';
484
485 END;
486 /
487 --apel--
488
489 DECLARE
    v_det VARCHAR2(200);
    v_id proiecte_proiect.id_proiect%TYPE:=&idproiect;
490
491 BEGIN
    v_det := Detalii_proiect(v_id);
    IF v_det = 'Nu există proiect!' THEN
        DBMS_OUTPUT.PUT_LINE('Nu există proiect!');
    ELSE
        DBMS_OUTPUT.PUT_LINE(v_det);
    END IF;
492 END;
493 /

```

Script Output | Task completed in 2.28 seconds  
Dorinte Indeplinite - Un proiect în care îi ajutam cu bani, mancare sau lucruri pe cei care au nevoie.

PL/SQL procedure successfully completed.

## **Declansatori**

--Prin modificarea id-ului de participant se va face modificarea pentru fiecare contributie a acestuia

```
CREATE OR REPLACE TRIGGER trg_update_participanti
```

```
BEFORE UPDATE OF id_participant ON participanti_proiect
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    UPDATE contributii_proiect
```

```
        SET id_participant = :NEW.id_participant
```

```
        WHERE id_participant = :OLD.id_participant;
```

```
END;
```

```
/
```

```
UPDATE participanti_proiect
```

```
SET id_participant = 100
```

```
WHERE id_participant = 5;
```

ID_CONTRIBUTIE	ID_PARTICIPANT	ID_PROIECT	TIP_CONTRIBUTIE	VALOARE_IN_BANI
1	1	1	1 jucarii	(null)
2	2	2	1 dulciuri	(null)
3	3	4	2 donatie	1000
4	4	100	2 donatie	1400
5	5	6	3 mancare si bauturi	(null)
6	6	7	3 donatie	800
7	7	4	2 donatie	250
8	8	100	2 donatie	260

--Nu va fi permisa diminuarea salariului sub valoarea minima de 3000

```
CREATE OR REPLACE TRIGGER trg_stop_taiere_sal
```

```
BEFORE UPDATE OF salariu ON angajati_proiect
FOR EACH ROW
BEGIN
  IF :NEW.salariu < 3000 THEN
    RAISE_APPLICATION_ERROR(-20010, 'Angajatul nu poate avea salariul sub minimul
    pe economie!');
  END IF;
END;
```

/

```
UPDATE angajati_proiect
```

```
SET salariu = 2000
```

```
WHERE id_angajat = 120;
```

The screenshot shows the Oracle SQL Developer interface. The code editor at the top contains the trigger definition and the update statement. The bottom pane shows the 'Script Output' tab with the results of the execution.

```

541 --Nu va fi permisa diminuarea salariului sub valoarea minima de 3000
542
543 CREATE OR REPLACE TRIGGER trg_stop_taiere_sal
544
545   BEFORE UPDATE OF salariu ON angajati_proiect
546   FOR EACH ROW
547   BEGIN
548     IF :NEW.salariu < 3000 THEN
549       RAISE_APPLICATION_ERROR(-20010, 'Angajatul nu poate avea salariul sub minimul
      pe economie!');
550     END IF;
551   END;
552   /
553
554 UPDATE angajati_proiect
555 SET salariu = 2000
556 WHERE id_angajat = 120;
  
```

Script Output X | Task completed in 0.199 seconds

Trigger TRG\_STOP\_TAIERE\_SAL compiled

Error starting at line : 554 in command -  
 UPDATE angajati\_proiect  
 SET salariu = 2000  
 WHERE id\_angajat = 120  
 Error report -  
 ORA-20010: Angajatul nu poate avea salariul sub minimul pe economie!  
 ORA-06512: at "SZABOA\_60.TRG\_STOP\_TAIERE\_SAL", line 3  
 ORA-04088: error during execution of trigger 'SZABOA\_60.TRG\_STOP\_TAIERE\_SAL'

--Nu vom putea adauga un nou proiect pe lista daca nu face parte din una din cele trei categorii

```
CREATE OR REPLACE TRIGGER trg_categorie_invalida
```

```
BEFORE INSERT ON proiecte_proiect
```

```
FOR EACH ROW
```

```
DECLARE
```

```
    v_categorie proiecte_proiect.categorie%TYPE;
```

```
BEGIN
```

```
    v_categorie := :NEW.categorie;
```

```
    IF v_categorie NOT IN ('caritate', 'tehnologie', 'entertainment') THEN
```

```
        RAISE_APPLICATION_ERROR(-20011, 'Proiectul trebuie să se încadreze în una dintre categoriile: caritate, tehnologie, entertainment.');
```

```
    END IF;
```

```
END;
```

```
/
```

```
INSERT INTO proiecte_proiect
```

```
VALUES (4, 'Music4All', 'Un proiect in care gazduim un festival de muzica.', 'festival');
```

The screenshot shows the Oracle SQL Developer interface. The code editor contains the SQL script for creating a trigger and inserting a row into the 'proiecte\_proiect' table. The trigger 'trg\_categorie\_invalida' checks if the inserted category is one of three specified ('caritate', 'tehnologie', 'entertainment'). If not, it raises an application error (-20011). The insert statement attempts to add a project with category 'festival', which fails due to the trigger constraint.

```

558 --Nu vom putea adauga un nou proiect pe lista daca nu face parte din una din cele trei categorii
559
560 CREATE OR REPLACE TRIGGER trg_categorie_invalida
561
562     BEFORE INSERT ON proiecte_proiect
563     FOR EACH ROW
564     DECLARE
565         v_categorie proiecte_proiect.categorie%TYPE;
566     BEGIN
567         v_categorie := :NEW.categorie;
568         IF v_categorie NOT IN ('caritate', 'tehnologie', 'entertainment') THEN
569             RAISE_APPLICATION_ERROR(-20011, 'Proiectul trebuie să se încadreze în una dintre categoriile: caritate, tehnologie, entertainment.');
570         END IF;
571     END;
572 /
573
574
575 INSERT INTO proiecte_proiect
576 VALUES (4, 'Music4All', 'Un proiect in care gazduim un festival de muzica.', 'festival')
577

```

Script Output:

```

Task completed in 0.279 seconds
Trigger TRG_CATEGORIE_INVALIDA compiled

Error starting at line : 575 in command -
INSERT INTO proiecte_proiect
VALUES (4, 'Music4All', 'Un proiect in care gazduim un festival de muzica.', 'festival')
Error report -
ORA-20011: Proiectul trebuie sa se incadreze in una dintre categoriile: caritate, tehnologie, entertainment.
ORA-06512: at "SZABOA_60.TRG_CATEGORIE_INVALIDA", line 6
ORA-04088: error during execution of trigger 'SZABOA_60.TRG_CATEGORIE_INVALIDA'

```