Facultatea de Matematică și Informatică

GESTIUNEA UNEI AGENȚII DE TURISM

Dobre Adriana Lia

Grupa 241

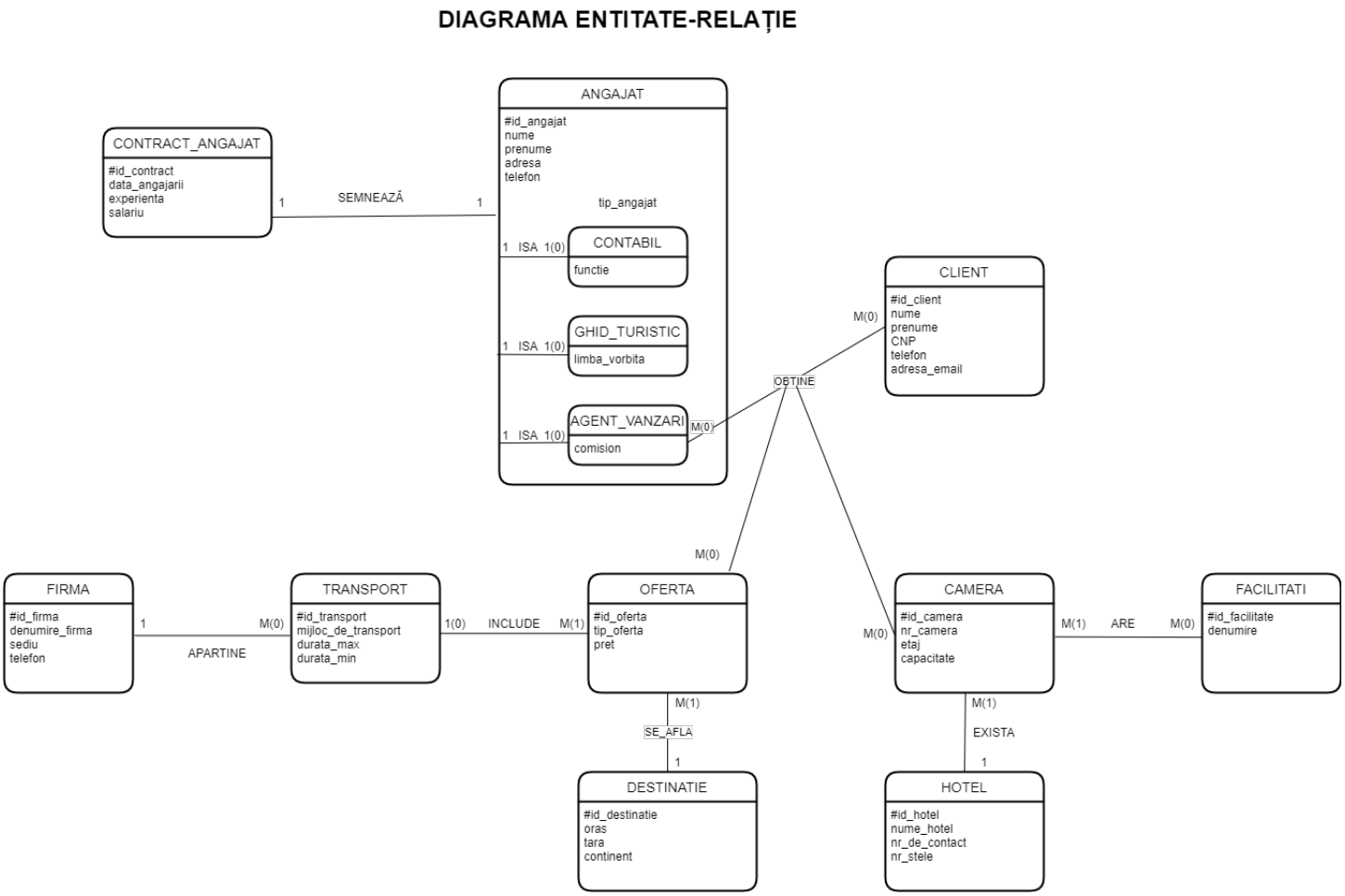
1. Utilitatea bazei de date

Modelul de date gestionează informații legate de organizarea și funcționarea activităților din cadrul unei agenții de turism. Buna funcționare a acestora este asigurată de angajații agenției care pot fi de mai multe tipuri : contabil, ghid turistic sau agent de vânzare. Fiecare angajat are un contract de muncă.

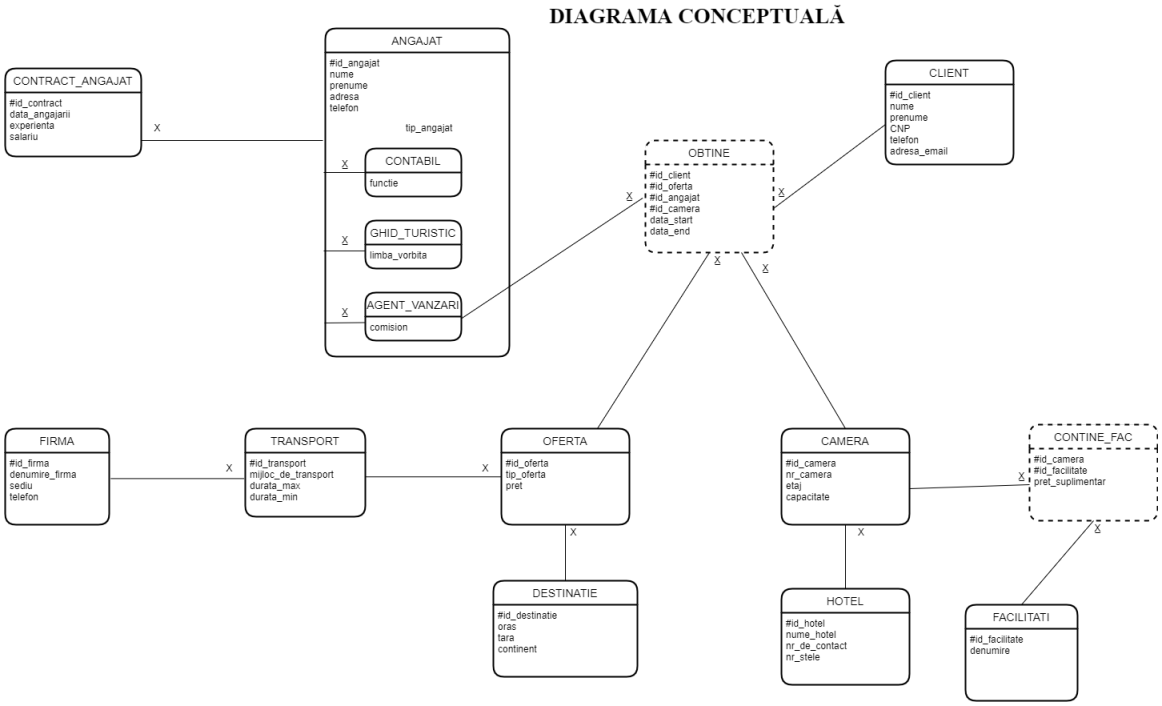
Clienții care aleg să își organizeze vacanța cu ajutorul agenției de turism apelează la agenții de vânzări ai companiei. La rândul lui, agentul analizează cerințele clientului și le oferă cea mai bună ofertă de vacanță.

În momentul în care clienții acceptă oferta, aceștia rezervă o cameră care poate avea mai multe facilități. Oferta prezentată de agent are o destinație specifică și poate include transport care este asigurat de firme specializate în acest domeniu.

2. Diagrama ERD



3. Diagrama Conceptuală



4. Implementare tabele

```
create table ANGAJAT
 (id_angajat number(5) constraint pk_ang primary key,
  nume varchar2(20) not null,
  prenume varchar2(20) not null,
  adresa varchar2(20) not null,
  telefon varchar2(20) not null,
  tip_angajat varchar2(20) not null,
  functie varchar2(40),
  limba_vorbita varchar2(20),
  comision number(10));
alter table ANGAJAT
add constraint tel_unic unique(telefon);
alter table ANGAJAT
add constraint nume_prenume_unic unique(nume, prenume);

create table CONTRACT_ANGAJAT
 (id_contract number(5) constraint pk_contract primary key,
  id_angajat number(5) not null constraint fkey_ang references ANGAJAT(id_angajat),
  data_angajarii date default sysdate ,
  experienta number(3),
  salariu number(10) not null,
  constraint ang unique(id_angajat));
```

```
create table CLIENT
 (id_client number(5) constraint pk_client primary key,
  nume varchar2(20) constraint nume_client not null,
  prenume varchar2(20) constraint prenume_client not null,
  CNP varchar2(20) not null,
  telefon varchar2(20) not null,
  adresa_email varchar(40) not null,
  constraint cnp_unic unique(CNP),
  constraint nr_tel unique(telefon),
  constraint email unique(adresa_email),
  constraint nume_pren_unic unique(nume, prenume));

create table HOTEL
 (id_hotel number(5) constraint pk_hotel primary key,
  nume_hotel varchar2(20) not null,
  nr_de_contact varchar2(20) not null,
  nr_stele number(10),
  constraint nr_contact unique(nr_de_contact));

create table CAMERA
 (id_camera number(5) constraint pk_camera primary key,
  id_hotel number(5) not null constraint fkey_hotel references HOTEL(id_hotel),
  nr_camera number(5) not null,
  etaj number(5),
  capacitate varchar2(20));

create table FACILITATI
 (id_facilitate number(5) constraint pk_fac primary key,
  denumire varchar2(40) not null);

create table CONTINE_FAC
 (id_camera number(5) constraint fk_cam references CAMERA(id_camera),
 id_facilitate number(5) constraint fk_fac references FACILITATI(id_facilitate),
 pret_suplimentar number(5),
 constraint pk_compus2 primary key(id_camera, id_facilitate));

create table DESTINATIE
 (id_destinatie number(5) constraint pk_destinatie primary key,
 oras varchar2(20) not null,
 tara varchar(20) not null,
 continent varchar2(20) not null);

create table FIRMA
 (id_firma number(5) constraint pk_firma primary key,
 denumire_firma varchar2(20) not null,
 sediu varchar2(20),
 telefon varchar(20) not null,
```

```
 constraint t_unic unique (telefon));

create table TRANSPORT
 (id_transport number(5) constraint pk_transport primary key,
 id_firma number(5) not null constraint fk_firma references FIRMA(id_firma),
 mijloc_de_transport varchar2(20) not null,
 durata_max number(10),
 durata_min number(10),
 constraint max_min check(durata_max>=durata_min));

create table OFERTA
 (id_oferta number(5) constraint pk_oferta primary key,
  tip_oferta varchar2(30) not null,
  pret number(10) not null,
  id_destinatie number(5)not null constraint fk_dest references DESTINATIE(id_destinatie),
  id_transport number(5) constraint fk_tr references TRANSPORT(id_transport));

create table OBTINE
 (id_client number(5) constraint fk_ct references CLIENT(id_client),
 id_angajat number(5) constraint fk_a references ANGAJAT(id_angajat),
 id_oferta number(5) constraint fk_of references OFERTA(id_oferta),
 id_camera number(5) constraint fk_cam1 references CAMERA(id_camera),
 data_start date default sysdate,
 data_end date default sysdate,
 constraint pk_compus3 primary key(id_client, id_angajat, id_oferta, id_camera),
 constraint start_end check(data_end>=data_start));
```

5. Inserare

```
CREATE SEQUENCE SEQ_ANG
INCREMENT by 10
START WITH 10
MAXVALUE 10000
NOCYCLE;

insert into ANGAJAT
values (10 , 'Popescu', 'Ion', 'Strada Bucuriei', '0745678192', 'Contabil',
               'inregistreaza tranzactii', NULL, NULL);

insert into ANGAJAT
values (SEQ_ANG.NEXTVAL, 'Costache', 'Ioana', 'Strada Aviatiei', '0731278142',
               'Ghid turistic', NULL , 'engleza' , NULL);

insert into ANGAJAT
values (SEQ_ANG.NEXTVAL, 'Bucur', 'Alexandru', 'Strada Soarelui', '0749078122',
               'Agent de vanzare', NULL, NULL, 5);
```

```
insert into ANGAJAT
values (SEQ_ANG.NEXTVAL, 'Marinescu', 'Catalina', 'Strada Victoriei', '0721908192',
                'Contabil', 'verifica tranzactii', NULL, NULL);

insert into ANGAJAT
values (SEQ_ANG.NEXTVAL, 'Tudorache', 'Marian', 'Strada Romana', '0731182812',
                'Ghid turistic', NULL , 'franceza' , NULL);

insert into ANGAJAT
values (SEQ_ANG.NEXTVAL, 'Sandu', 'Maria', 'Strada Ciresului', '0749197132',
                'Agent de vanzare', NULL, NULL, 4);

insert into ANGAJAT
values (SEQ_ANG.NEXTVAL, 'Radu', 'Alexandru', 'Strada Libertatii', '0749017132',
                'Agent de vanzare', NULL, NULL, 2);

insert into ANGAJAT
values (SEQ_ANG.NEXTVAL, 'Enache', 'Andrei', 'Strada Independentei', '0739022132',
                'Agent de vanzare',NULL, NULL, 1);

insert into ANGAJAT
values (SEQ_ANG.NEXTVAL, 'Enache', 'Alexandra', 'Strada Ion Creanga', '0746617138',
                'Agent de vanzare', NULL, NULL, 3);

select *
from angajat;
```

```
select *
from angajat;
```

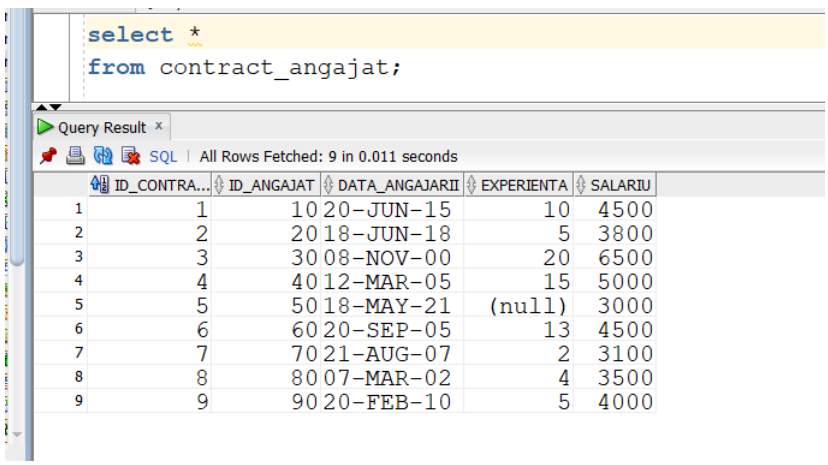
Query Result

All Rows Fetched: 9 in 0.011 seconds

	ID_ANGAJ...	NUME	PRENUME	ADRESA	TELEFON	TIP_ANGAJAT	FUNCTIE	LIMBA_VORBITA	COMISION
1	10	Popescu	Ion	Strada Bucuriei	0745678192	Contabil	inregistreaza tranzactii	(null)	(null)
2	20	Costache	Ioana	Strada Aviatiei	0731278142	Ghid turistic	(null)	engleza	(null)
3	30	Bucur	Alexandru	Strada Soarelui	0749078122	Agent de vanzare	(null)	(null)	5
4	40	Marinescu	Catalina	Strada Victoriei	0721908192	Contabil	verifica tranzactii	(null)	(null)
5	50	Tudorache	Marian	Strada Romana	0731182812	Ghid turistic	(null)	franceza	(null)
6	60	Sandu	Maria	Strada Ciresului	0749197132	Agent de vanzare	(null)	(null)	4
7	70	Radu	Alexandru	Strada Libertatii	0749017132	Agent de vanzare	(null)	(null)	2
8	80	Enache	Andrei	Strada Independentei	0739022132	Agent de vanzare	(null)	(null)	1
9	90	Enache	Alexandra	Strada Ion Creanga	0746617138	Agent de vanzare	(null)	(null)	3

```
insert into CONTRACT_ANGAJAT
values (1, 10, to_date('20-06-2015','dd-mm-yyyy'), 10, 4500);

insert into CONTRACT_ANGAJAT
values (2, 20, to_date('18-06-2018','dd-mm-yyyy'), 5, 3800);

insert into CONTRACT_ANGAJAT
values (3, 30, to_date('08-11-2000','dd-mm-yyyy'), 20, 6500);

insert into CONTRACT_ANGAJAT
values (4, 40, to_date('12-03-2005','dd-mm-yyyy'), 15, 5000);
```

```
insert into CONTRACT_ANGAJAT
values (5, 50, to_date('18-05-2021','dd-mm-yyyy'), NULL, 3000);

insert into CONTRACT_ANGAJAT
values (6, 60, to_date('20-09-2005','dd-mm-yyyy'), 13, 4500);

insert into CONTRACT_ANGAJAT
values (7, 70, to_date('21-08-2007','dd-mm-yyyy'), 2, 3100);

insert into CONTRACT_ANGAJAT
values (8, 80, to_date('07-03-2002','dd-mm-yyyy'), 4, 3500);

insert into CONTRACT_ANGAJAT
values (9, 90, to_date('20-02-2010','dd-mm-yyyy'), 5, 4000);

select *
from contract_angajat;
```



```
select *
from contract_angajat;
```

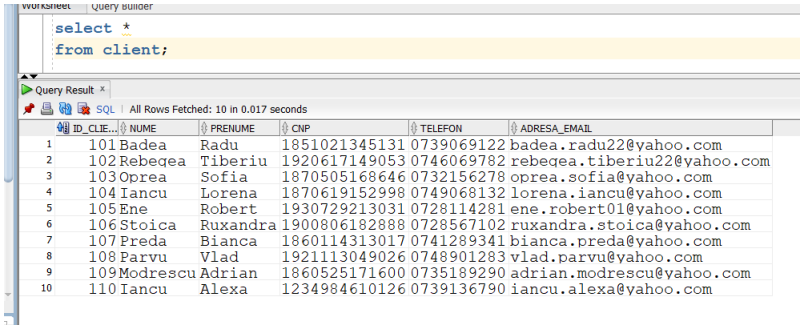
Query Result

SQL | All Rows Fetched: 9 in 0.011 seconds

	ID_CONTRA...	ID_ANGAJAT	DATA_ANGAJARII	EXPERIENTA	SALARIU
1	1	10	20-JUN-15	10	4500
2	2	20	18-JUN-18	5	3800
3	3	30	08-NOV-00	20	6500
4	4	40	12-MAR-05	15	5000
5	5	50	18-MAY-21	(null)	3000
6	6	60	20-SEP-05	13	4500
7	7	70	21-AUG-07	2	3100
8	8	80	07-MAR-02	4	3500
9	9	90	20-FEB-10	5	4000

```
CREATE SEQUENCE SEQ_CL
INCREMENT by 1
START WITH 100
MAXVALUE 10000
NOCYCLE;

insert into CLIENT
values (SEQ_CL.NEXTVAL, 'Badea', 'Radu', '1851021345131', '0739069122',
            'badea.radu22@yahoo.com');

insert into CLIENT
values (SEQ_CL.NEXTVAL, 'Rebegea', 'Tiberiu', '1920617149053', '0746069782',
             'rebegea.tiberiu22@yahoo.com');
```

```
insert into CLIENT
values (SEQ_CL.NEXTVAL, 'Oprea', 'Sofia', '1870505168646', '0732156278',
            'oprea.sofia@yahoo.com');

insert into CLIENT
values (SEQ_CL.NEXTVAL, 'Iancu', 'Lorena', '1870619152998', '0749068132',
            'lorena.iancu@yahoo.com');

insert into CLIENT
values (SEQ_CL.NEXTVAL, 'Ene', 'Robert', '1930729213031', '0728114281',
            'ene.robert01@yahoo.com');

insert into CLIENT
values (SEQ_CL.NEXTVAL, 'Stoica', 'Ruxandra', '1900806182888', '0728567102',
            'ruxandra.stoica@yahoo.com');

insert into CLIENT
values (SEQ_CL.NEXTVAL, 'Preda', 'Bianca', '1860114313017', '0741289341',
            'bianca.preda@yahoo.com');

insert into CLIENT
values (SEQ_CL.NEXTVAL, 'Parvu', 'Vlad', '1921113049026', '0748901283',
            'vlad.parvu@yahoo.com');

insert into CLIENT
values (SEQ_CL.NEXTVAL, 'Modrescu', 'Adrian', '1860525171600', '0735189290',
            'adrian.modrescu@yahoo.com');

insert into CLIENT
values (SEQ_CL.NEXTVAL, 'Iancu', 'Alexa', '1234984610126', '0739136790',
             'iancu.alexa@yahoo.com');

insert into CLIENT
values (SEQ_CL.NEXTVAL, 'Teches', 'Laurentiu', '1646134529013', '0721137690',
             'teches.laur@yahoo.com');

select *
from client;
```



Worksheet | Query Builder

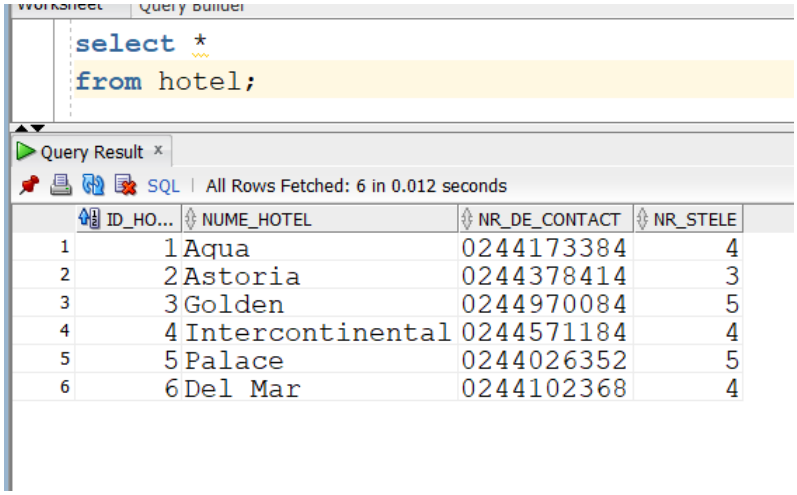
```
select *
from client;
```

Query Result

SQL | All Rows Fetched: 10 in 0.017 seconds

	ID_CLIE...	NUME	PRENUME	CNP	TELEFON	ADRESA_EMAIL
1	101	Badea	Radu	1851021345131	0739069122	badea.radu22@yahoo.com
2	102	Rebegea	Tiberiu	1920617149053	0746069782	rebegea.tiberiu22@yahoo.com
3	103	Oprea	Sofia	1870505168646	0732156278	oprea.sofia@yahoo.com
4	104	Iancu	Lorena	1870619152998	0749068132	lorena.iancu@yahoo.com
5	105	Ene	Robert	1930729213031	0728114281	ene.robert01@yahoo.com
6	106	Stoica	Ruxandra	1900806182888	0728567102	ruxandra.stoica@yahoo.com
7	107	Preda	Bianca	1860114313017	0741289341	bianca.preda@yahoo.com
8	108	Parvu	Vlad	1921113049026	0748901283	vlad.parvu@yahoo.com
9	109	Modrescu	Adrian	1860525171600	0735189290	adrian.modrescu@yahoo.com
10	110	Iancu	Alexa	1234984610126	0739136790	iancu.alexa@yahoo.com

```
insert into HOTEL
values (1, 'Aqua', '0244173384', 4);

insert into HOTEL
values (2, 'Astoria', '0244378414', 3);

insert into HOTEL
values (3, 'Golden', '0244970084', 5);

insert into HOTEL
values (4, 'Intercontinental', '0244571184', 4);

insert into HOTEL
values (5, 'Palace', '0244026352', 5);

insert into HOTEL
values (6, 'Del Mar', '0244102368', 4);

select *
from hotel;
```



	ID_HO...	NUME_HOTEL	NR_DE_CONTACT	NR_STELE
1	1	Aqua	0244173384	4
2	2	Astoria	0244378414	3
3	3	Golden	0244970084	5
4	4	Intercontinental	0244571184	4
5	5	Palace	0244026352	5
6	6	Del Mar	0244102368	4

```
CREATE SEQUENCE SEQ_CAM
INCREMENT by 1
START WITH 200
MAXVALUE 10000
NOCYCLE;

insert into CAMERA
values (SEQ_CAM.NEXTVAL,1 , 301, 1, 'dubla');

insert into CAMERA
values (SEQ_CAM.NEXTVAL, 1, 302, 1, 'standard');
```

```
insert into CAMERA
values (SEQ_CAM.NEXTVAL, 1, 303, 1, 'dubla');

insert into CAMERA
values (SEQ_CAM.NEXTVAL, 2, 301, 2, 'tripla');

insert into CAMERA
values (SEQ_CAM.NEXTVAL, 3, 310, 4, 'dubla');

insert into CAMERA
values (SEQ_CAM.NEXTVAL, 4, 340, 5, 'tripla');

insert into CAMERA
values (SEQ_CAM.NEXTVAL, 5, 313, 3, 'tripla');

insert into CAMERA
values (SEQ_CAM.NEXTVAL, 6, 321, 8, 'matrimoniala');

insert into CAMERA
values (SEQ_CAM.NEXTVAL, 3, 317, 7, 'dubla');

insert into CAMERA
values (SEQ_CAM.NEXTVAL, 5, 302, 2, 'dubla');

select *
from camera;
```

```
select *
from camera;
```

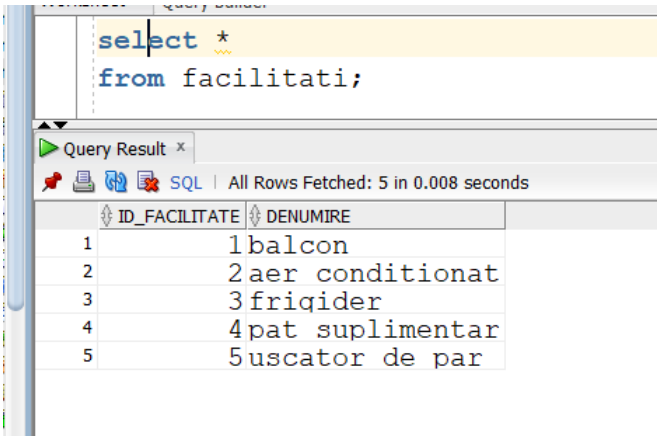
Query Result

SQL | All Rows Fetched: 10 in 0.024 seconds

	ID_CAME...	ID_HOTEL	NR_CAMERA	ETAJ	CAPACITATE
1	201	1	301	1	dubla
2	202	1	302	1	standard
3	203	1	303	1	dubla
4	204	2	301	2	tripla
5	205	3	310	4	dubla
6	206	4	340	5	tripla
7	207	5	313	3	tripla
8	208	6	321	8	matrimoniala
9	209	3	317	7	dubla
10	210	5	302	2	dubla

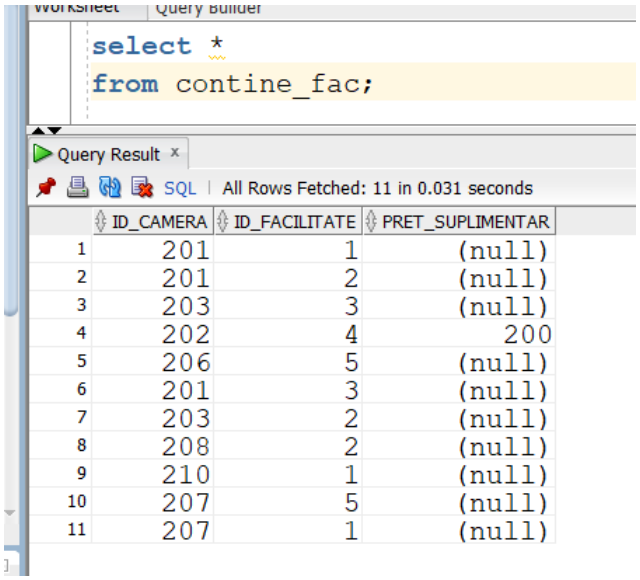
```
insert into FACILITATI
values (1, 'balcon');

insert into FACILITATI
values (2, 'aer conditionat');
```

```
insert into FACILITATI
values (3, 'frigider');

insert into FACILITATI
values (4, 'pat suplimentar');

insert into FACILITATI
values (5, 'uscator de par');

select *
from facilitati;
```




```
insert into CONTINE_FAC
values (201, 1, null);

insert into CONTINE_FAC
values (201, 2, null);

insert into CONTINE_FAC
values (203, 3, null);

insert into CONTINE_FAC
values (202, 4, 200);

insert into CONTINE_FAC
values (206, 5, null);

insert into CONTINE_FAC
values (201, 3, null);

insert into CONTINE_FAC
values (203, 2, null);

insert into CONTINE_FAC
values (208, 2, null);
```

```
insert into CONTINE_FAC
values (210, 1, null);

insert into CONTINE_FAC
values (207, 5, null);

insert into CONTINE_FAC
values (207, 1, null);

select *
from contine_fac;
```



	ID_CAMERA	ID_FACILITATE	PRET_SUPLIMENTAR
1	201	1	(null)
2	201	2	(null)
3	203	3	(null)
4	202	4	200
5	206	5	(null)
6	201	3	(null)
7	203	2	(null)
8	208	2	(null)
9	210	1	(null)
10	207	5	(null)
11	207	1	(null)

```
insert into DESTINATIE
values (1, 'Constanta', 'Romania','Europa');

insert into DESTINATIE
values (2, 'Paris', 'Franta','Europa');

insert into DESTINATIE
values (3, 'Barcelona', 'Spania','Europa');

insert into DESTINATIE
values (4, 'Venetia', 'Italia','Europa');

insert into DESTINATIE
values (5, 'Brasov', 'Romania','Europa');

insert into DESTINATIE
values (6, 'Cluj Napoca','Romania', 'Europa');

select *
from destinatie;
```

```
select *
from destinatie;
```

Query Result

SQL | All Rows Fetched: 6 in 0.009 seconds

	ID_DESTINA...	ORAS	TARA	CONTINENT
1	1	Constanta	Romania	Europa
2	2	Paris	Franta	Europa
3	3	Barcelona	Spania	Europa
4	4	Venetia	Italia	Europa
5	5	Brasov	Romania	Europa
6	6	Cluj Napoca	Romania	Europa

```
insert into FIRMA
values (1, 'Matrix Transport', 'Bucuresti','0742102828');

insert into FIRMA
values (2, 'Mateo Tour', 'Bucuresti','0743792828');

insert into FIRMA
values (3, 'Air Bucharest', 'Bucuresti','0742100815');

insert into FIRMA
values (4, 'AVX Auto', 'Bucuresti','0722101228');

insert into FIRMA
values (5, 'Blue Air', 'Bucuresti','0736725278');

insert into FIRMA
values (6, 'TAROM', 'Bucuresti','0729912267');

select *
from firma;
```

Worksheet | Query Builder

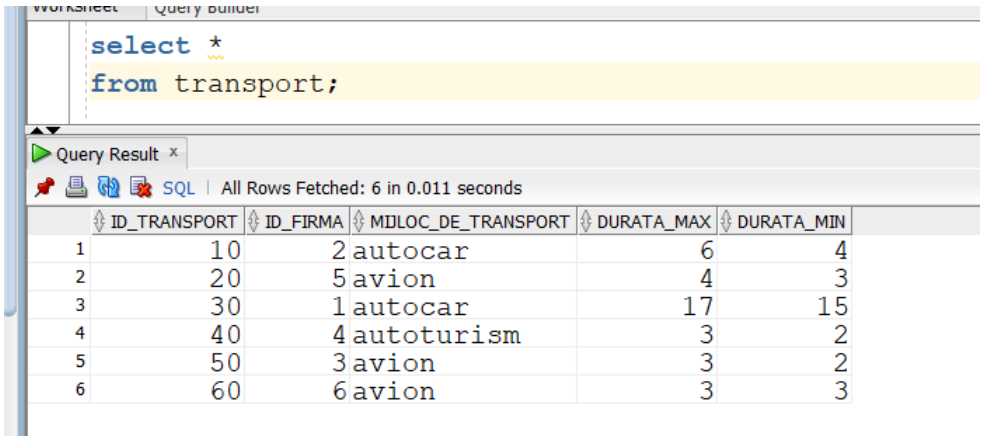
```
select *
from firma;
```

Query Result

SQL | All Rows Fetched: 6 in 0.054 seconds

	ID_FIRMA	DENUMIRE_FIRMA	SEDIU	TELEFON
1	1	Matrix Transport	Bucuresti	0742102828
2	2	Mateo Tour	Bucuresti	0743792828
3	3	Air Bucharest	Bucuresti	0742100815
4	4	AVX Auto	Bucuresti	0722101228
5	5	Blue Air	Bucuresti	0736725278
6	6	TAROM	Bucuresti	0729912267

```
CREATE SEQUENCE SEQ_TR
INCREMENT by 10
START WITH 10
MAXVALUE 10000
NOCYCLE;

insert into TRANSPORT
values (SEQ_TR.NEXTVAL, 2 , 'autocar' , 6, 4);

insert into TRANSPORT
values (SEQ_TR.NEXTVAL, 5 , 'avion' , 4, 3);

insert into TRANSPORT
values (SEQ_TR.NEXTVAL, 1 , 'autocar' , 17, 15);

insert into TRANSPORT
values (SEQ_TR.NEXTVAL, 4 , 'autoturism' , 3, 2);

insert into TRANSPORT
values (SEQ_TR.NEXTVAL, 3 , 'avion' , 3, 2);

insert into TRANSPORT
values (SEQ_TR.NEXTVAL, 6 , 'avion' , 3, 3);

select *
from transport;
```



	ID_TRANSPORT	ID_FIRMA	MIJLOC_DE_TRANSPORT	DURATA_MAX	DURATA_MIN
1	10	2	autocar	6	4
2	20	5	avion	4	3
3	30	1	autocar	17	15
4	40	4	autoturism	3	2
5	50	3	avion	3	2
6	60	6	avion	3	3

```
insert into OFERTA
values (100, 'sejur', 5500,1,10);

insert into OFERTA
values (200, 'sejur', 3800,1,null);

insert into OFERTA
values (300, 'sejur', 7000,2,20);
```

```
insert into OFERTA
values (400, 'sejur', 9500,4,50);

insert into OFERTA
values (500, 'sejur', 3200,1,null);

insert into OFERTA
values (600, 'sejur', 5000,1,null);

insert into OFERTA
values (700, 'sejur', 9500,3,60);

insert into OFERTA
values (800, 'sejur', 5500,1,60);

insert into OFERTA
values (900, 'city break', 3000,1,null);

insert into OFERTA
values (1000, 'sejur', 5000,6,30);

insert into OFERTA
values (1100, 'sejur', 8600,5,40);

insert into OFERTA
values (1200, 'city break', 7000,3,50);

select *
from oferta;
```

Worksheet | Query Builder

```
select *
from oferta;
```

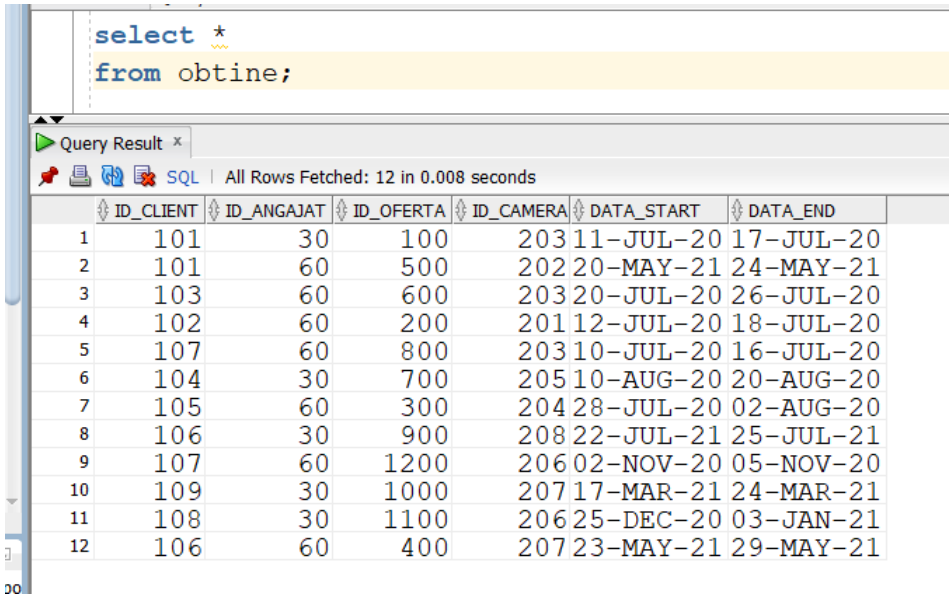
Query Result

SQL | All Rows Fetched: 12 in 0.011 seconds

	ID_OFERTA	TIP_OFERTA	PRET	ID_DESTI...	ID_TRANSPORT
1	100	sejur	5500	1	10
2	200	sejur	3800	1	(null)
3	300	sejur	7000	2	20
4	400	sejur	9500	4	50
5	500	sejur	3200	1	(null)
6	600	sejur	5000	1	(null)
7	700	sejur	9500	3	60
8	800	sejur	5500	1	60
9	900	city break	3000	1	(null)
10	1000	sejur	5000	6	30
11	1100	sejur	8600	5	40
12	1200	city break	7000	3	50

```
insert into OBTINE
values (101, 30, 100,203, to_date('11-07-2020','dd-mm-yyyy'),
          to_date('17-07-2020','dd-mm-yyyy'));

insert into OBTINE
values (101, 60, 500, 202,to_date('20-05-2021','dd-mm-yyyy'),
           to_date('24-05-2021','dd-mm-yyyy'));

insert into OBTINE
values (103, 60, 600, 203, to_date('20-07-2020','dd-mm-yyyy'),
           to_date('26-07-2020','dd-mm-yyyy'));

insert into OBTINE
values (102, 60, 200, 201, to_date('12-07-2020','dd-mm-yyyy'),
           to_date('18-07-2020','dd-mm-yyyy'));

insert into OBTINE
values (107, 60, 800, 203, to_date('10-07-2020','dd-mm-yyyy'),
           to_date('16-07-2020','dd-mm-yyyy'));

insert into OBTINE
values (104, 30, 700, 205, to_date('10-08-2020','dd-mm-yyyy'),
           to_date('20-08-2020','dd-mm-yyyy'));

insert into OBTINE
values (105, 60, 300, 204, to_date('28-07-2020','dd-mm-yyyy'),
           to_date('02-08-2020','dd-mm-yyyy'));

insert into OBTINE
values (106, 30, 900, 208, to_date('22-07-2021','dd-mm-yyyy'),
           to_date('25-07-2021','dd-mm-yyyy'));

insert into OBTINE
values (107, 60, 1200, 206, to_date('02-11-2020','dd-mm-yyyy'),
           to_date('05-11-2020','dd-mm-yyyy'));

insert into OBTINE
values (109, 30, 1000, 207, to_date('17-03-2021','dd-mm-yyyy'),
           to_date('24-03-2021','dd-mm-yyyy'));

insert into OBTINE
values (108, 30, 1100, 206, to_date('25-12-2020','dd-mm-yyyy'),
           to_date('03-01-2021','dd-mm-yyyy'));

insert into OBTINE
values (106, 60, 400, 207, to_date('23-05-2021','dd-mm-yyyy'),
           to_date('29-05-2021','dd-mm-yyyy'));
```

```
select *
from obtine;
```



```
select *
from obtine;
```

Query Result

SQL | All Rows Fetched: 12 in 0.008 seconds

	ID_CLIENT	ID_ANGAJAT	ID_OFERTA	ID_CAMERA	DATA_START	DATA_END
1	101	30	100	203	11-JUL-20	17-JUL-20
2	101	60	500	202	20-MAY-21	24-MAY-21
3	103	60	600	203	20-JUL-20	26-JUL-20
4	102	60	200	201	12-JUL-20	18-JUL-20
5	107	60	800	203	10-JUL-20	16-JUL-20
6	104	30	700	205	10-AUG-20	20-AUG-20
7	105	60	300	204	28-JUL-20	02-AUG-20
8	106	30	900	208	22-JUL-21	25-JUL-21
9	107	60	1200	206	02-NOV-20	05-NOV-20
10	109	30	1000	207	17-MAR-21	24-MAR-21
11	108	30	1100	206	25-DEC-20	03-JAN-21
12	106	60	400	207	23-MAY-21	29-MAY-21

6. Subprogram stocat - 2 tipuri de colecție

Pentru un client introdus de la tastatură, afișați numărul de oferte achiziționate, detalii despre acestea și facilitățile pe care le conține.

```
CREATE OR REPLACE PROCEDURE p_ex6
    (v_nume client.nume%TYPE)
   IS
      nr_oferte NUMBER;
      nr_fac NUMBER;
      cod_client client.id_client%TYPE;
      TYPE tablou_imbricat IS TABLE OF obtine%ROWTYPE;
      info_of tablou_imbricat;
      TYPE tablou_indexat IS TABLE OF facilitati%ROWTYPE
                     INDEX BY BINARY_INTEGER;
      fac_camera tablou_indexat;
      exc_oferte EXCEPTION;
      exc_fac EXCEPTION;
    BEGIN
      --pentru a intra pe exceptia de no_data_found trebuie sa verificam
      --daca numele exista in tabel, daca nu faceam acest select se facea
      --confuzie intre clientii care nu exista si clientii care nu au
      --achizitionat oferte(nu intra niciodata pe no_data_found,
      --ci intra pe exc_oferte)
      SELECT id_client
      INTO cod_client
      FROM client
      WHERE UPPER(nume) = UPPER(v_nume);
      --retinem detaliile despre oferte
      SELECT o.id_client,o.id_angajat,o.id_oferta,o.id_camera, data_start,
   data_end
      BULK COLLECT INTO info_of
```

```
   FROM client c, obtine o
   WHERE UPPER(c.nume) = UPPER(v_nume)
         and o.id_client = c.id_client;
   nr_oferte := info_of.COUNT;
   IF nr_oferte = 0 THEN --daca clientul nu a achizitionat nicio oferta
        RAISE exc_oferte;
   ELSIF nr_oferte = 1 THEN --daca clientul a achizionat doar o oferta
       DBMS_OUTPUT.PUT_LINE('Clientul ' || v_nume || ' a achizitionat o
oferta.');
       DBMS_OUTPUT.PUT_LINE('Oferta achizitionata este:');
   ELSE
       DBMS_OUTPUT.PUT_LINE('Clientul ' || v_nume || ' a achizitionat ' ||
nr_oferte || ' oferte.');
       DBMS_OUTPUT.PUT_LINE('Ofertele achizitionate sunt:');
   END IF;
   --parcurgem fiecare oferta
   FOR i IN info_of.FIRST..info_of.LAST LOOP
      --afisam detaliile despre oferta
      DBMS_OUTPUT.PUT_LINE(i || '. Oferta ' || info_of(i).id_oferta || ' ->
incepe pe data de ' || info_of(i).data_start);
      DBMS_OUTPUT.PUT_LINE('              -> se termina pe data de ' ||
info_of(i).data_end);
      DBMS_OUTPUT.PUT_LINE('              -> este rezervata pe camera ' ||
info_of(i).id_camera);
      --retinem facilitatile camerei
      SELECT f.id_facilitate,denumire
      BULK COLLECT INTO fac_camera
      FROM obtine o, contine_fac cf, facilitati f
      WHERE id_oferta = info_of(i).id_oferta
            and o.id_camera = cf.id_camera
            and cf.id_facilitate = f.id_facilitate;
      nr_fac := fac_camera.COUNT;
      IF nr_fac = 0 THEN --in cazul in care camera nu are facilitati
         RAISE exc_fac;
      ELSIF nr_fac = 1 THEN --in cazul in care camera are doar o facilitate
         DBMS_OUTPUT.PUT_LINE('Camera ' || info_of(i).id_camera || ' are o
singura facilitate:');
      ELSE
         DBMS_OUTPUT.PUT_LINE('Camera ' || info_of(i).id_camera || ' are '
|| nr_fac || ' facilitati:');
      END IF;
      --afisam facilitatile camerei
      FOR j in fac_camera.FIRST..fac_camera.last LOOP
          DBMS_OUTPUT.PUT_LINE('-> ' || fac_camera(j).denumire);
      END LOOP;
      DBMS_OUTPUT.NEW_LINE;
  END LOOP;
  EXCEPTION --tratam exceptii
      WHEN NO_DATA_FOUND THEN
            RAISE_APPLICATION_ERROR(-20000, 'Nu exista clientul.');
      WHEN TOO_MANY_ROWS THEN
            RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi clienti cu
acelasi nume.');
      WHEN exc_oferte THEN
            RAISE_APPLICATION_ERROR(-20002, 'Clientul nu a achizitionat
nicio oferta.');
```

```
        WHEN exc_fac THEN
              RAISE_APPLICATION_ERROR(-20003, 'Camera nu are facilitati.');
END p_ex6;
/
--citim de la tastatura clientul si apelam procedura pentru clientul introdus
DECLARE
    v_nume client.nume%TYPE := Initcap('&p_nume');
BEGIN
    p_ex6(v_nume);
END;
/
--am apelat pentru clientul Iancu
BEGIN
   p_ex6('Iancu');
END;
/
```

Am apelat procedura p_ex6 pentru clientul Iancu.

```
BEGIN
   p_ex6('Iancu');
END;
/

--ex 7
```

Script Output | Query Result

Task completed in 0.066 seconds

```
Error starting at line : 102 in command -
BEGIN
   p_ex6('Iancu');
END;
Error report -
ORA-20001: Exista mai multi clienti cu acelasi nume.
ORA-06512: at "PROIECT1.P_EX6", line 67
ORA-06512: at line 2
```

Am apelat procedura p_ex6 pentru clientul Badea.

```
        FOR j in fac_camera.FIRST..fac_camera.last LOOP
            DBMS_OUTPUT.PUT_LINE('-> ' || fac_camera(j).denumire);
        END LOOP;
        DBMS_OUTPUT.NEW_LINE;
    END LOOP;
    EXCEPTION --tratam exceptii
        WHEN NO_DATA_FOUND THEN
              RAISE_APPLICATION_ERROR(-20000, 'Nu exista clientul.');
        WHEN TOO_MANY_ROWS THEN
              RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi clienti cu acelasi nume.');
        WHEN exc_oferte THEN
              RAISE_APPLICATION_ERROR(-20002, 'Clientul nu a achizitionat nicio oferta.');
        WHEN exc_fac THEN
              RAISE_APPLICATION_ERROR(-20003, 'Camera nu are facilitati.');
END p_ex6;
/

DECLARE
    v_nume client.nume%TYPE := Initcap('&p_nume');
BEGIN
    p_ex6(v_nume);
```

Script Output | Query Result

Task completed in 2.404 seconds

```
PL/SQL procedure successfully completed.
```

```
Clientul Badea a achizitionat 2 oferte.
Ofertele achizitionate sunt:
1. Oferta 100 -> incepe pe data de 11-JUL-20
              -> se termina pe data de 17-JUL-20
              -> este rezervata pe camera 203
Camera 203 are 2 facilitati:
-> aer conditionat
-> frigider

2. Oferta 500 -> incepe pe data de 20-MAY-21
              -> se termina pe data de 24-MAY-21
              -> este rezervata pe camera 202
Camera 202 are o singura facilitate:
-> pat suplimentar
```

Am apelat procedura p_ex6 pentru clientul Adriana.

```
DECLARE
    v_nume client.nume%TYPE := Initcap('&p_nume');
BEGIN
    p_ex6(v_nume);
END;
/
```

Script Output | Query Result

Task completed in 3.372 seconds

```
DECLARE
    v_nume client.nume%TYPE := Initcap('Adriana');
BEGIN
    p_ex6(v_nume);
END;
Error report -
ORA-20000: Nu exista clientul.
ORA-06512: at "PROIECT1.P_EX6", line 65
ORA-06512: at line 4
20000. 00000 -  "%s"
*Cause:    The stored procedure 'raise_application_error'
           was called which causes this error to be generated.
*Action:   Correct the problem as described in the error message or contact
           the application administrator or DBA for more information.
```

Am apelat procedura p_ex6 pentru clientul Ene.

Worksheet | Query Builder

```
        WHEN exc_oferte THEN
            RAISE_APPLICATION_ERROR(-20002, 'Clientul nu a achizitionat nicio oferta.');
        WHEN exc_fac THEN
            RAISE_APPLICATION_ERROR(-20003, 'Camera nu are facilitati.');
END p_ex6;
/

DECLARE
    v_nume client.nume%TYPE := Initcap('&p_nume');
BEGIN
    p_ex6(v_nume);
END;
/
```

Script Output | Query Result

Task completed in 2.537 seconds

```
Error starting at line : 95 in command -
DECLARE
    v_nume client.nume%TYPE := Initcap('Ene');
BEGIN
    p_ex6(v_nume);
END;
Error report -
ORA-20003: Camera nu are facilitati.
ORA-06512: at "PROIECT1.P_EX6", line 71
ORA-06512: at line 4
```

```
Clientul Ene a achizitionat o oferta.
Oferta achizitionata este:
1. Oferta 300 -> incepe pe data de 28-JUL-20
              -> se termina pe data de 02-AUG-20
              -> este rezervata pe camera 204
```

Am apelat procedura p_ex6 pentru clientul Teches.

```
DECLARE
    v_nume client.nume%TYPE := Initcap('&p_nume');
BEGIN
    p_ex6(v_nume);
END;
/
```

```
Script Output   Query Result
Task completed in 3.343 seconds
Error starting at line : 95 in command -
DECLARE
    v_nume client.nume%TYPE := Initcap('Teches');
BEGIN
    p_ex6(v_nume);
END;
Error report -
ORA-20002: Clientul nu a achizitionat nicio oferta.
ORA-06512: at "PROIECT1.P_EX6", line 69
ORA-06512: at line 4
```

7. Subprogram - cursoare

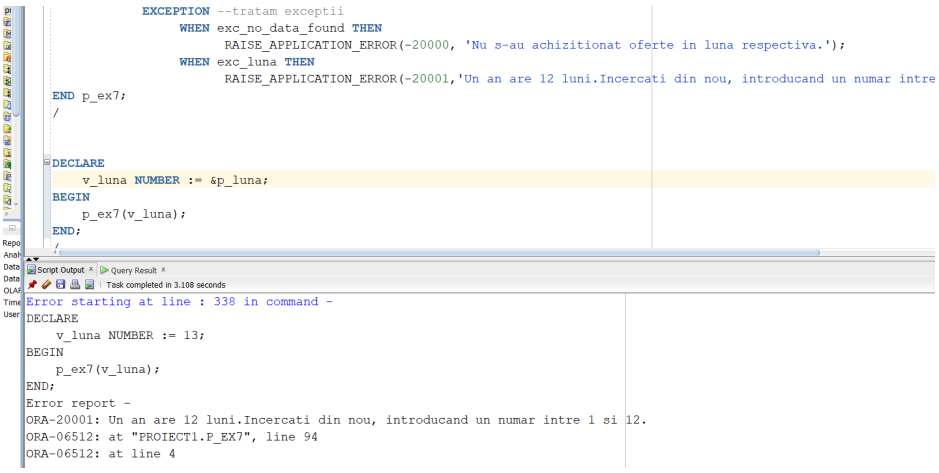
Pentru o lună introdusă de la tastatură, obtineți pentru fiecare destinație top 3 cele mai scumpe oferte. Dacă există mai multe oferte care au același preț, atunci acestea vor ocupa aceeași poziție în top 3.

```
CREATE OR REPLACE PROCEDURE p_ex7
        (luna NUMBER)
      IS
        --am luat un cursor pentru destinatii
        --daca nu faceam cu UNION nu lua destinatiile care nu au avut
        --nicio oferta in luna introdusa de la tastatura
        CURSOR cursor_destinatie IS
            SELECT d.id_destinatie, MAX(oras),COUNT(o.id_oferta)
            FROM destinatie d, oferta o, obtine ob
            WHERE d.id_destinatie = o.id_destinatie
                   AND TO_CHAR(data_start, 'MM') = luna
                   AND o.id_oferta = ob.id_oferta
            GROUP BY d.id_destinatie
            UNION
            SELECT id_destinatie,oras,0
            FROM destinatie
            WHERE id_destinatie NOT IN (SELECT d.id_destinatie
                                        FROM destinatie d, oferta o, obtine ob
                                        WHERE d.id_destinatie = o.id_destinatie
                                        AND TO_CHAR(data_start, 'MM') = luna
                                              AND o.id_oferta = ob.id_oferta);
        --am luat un cursor pentru oferte
        CURSOR cursor_oferta IS
            SELECT o.id_oferta, pret, id_destinatie
            FROM obtine o, oferta oft
            WHERE TO_CHAR(data_start, 'MM') = luna
                  AND o.id_oferta = oft.id_oferta
```

```
        ORDER BY pret DESC;
    cod_dest destinatie.id_destinatie%TYPE;
    nume_dest destinatie.oras%TYPE;
    nr_oferte NUMBER(5);
    nr_oferte_total NUMBER(5);
    cod_oferta oferta.id_oferta%TYPE;
    pret_of oferta.pret%TYPE;
    cod_destinatie oferta.id_destinatie%TYPE;
    top number(1) :=0; --top
    aux number; --retinem pretul anterior
    prima_trecere number :=1;
    exc_luna EXCEPTION;
    exc_no_data_found EXCEPTION;
    BEGIN
       --verificam daca luna introdusa poate exista
       IF luna <= 0 OR luna > 12 THEN
           RAISE exc_luna;
       END IF;
       --calculam nr total de oferte achizitionate in luna respectiva
       SELECT count(data_start)
       INTO nr_oferte_total
       FROM obtine
       WHERE TO_CHAR(data_start, 'MM') = luna;
       --verificam daca in tabel exista vreo oferta achizitionata in luna
introdusa
       IF nr_oferte_total = 0 THEN
           RAISE exc_no_data_found;
       ELSIF nr_oferte_total = 1 THEN --daca exista doar o oferta
achizitionata
           DBMS_OUTPUT.PUT_LINE('In luna a-' || luna || '-a s-a
achizitionat doar o oferta.');
       ELSE
           DBMS_OUTPUT.PUT_LINE('In luna a-' || luna || '-a s-au
achizitionat in total ' || nr_oferte_total || ' oferte.');
       END IF;
       OPEN cursor_destinatie; --deschidem cursorul destinatie
       DBMS_OUTPUT.PUT_LINE('In luna a-' || luna || '-a se regasesc
urmatoarele topuri:' );
       LOOP
        top := 0; --topul porneste de la 0 la fiecare destinatie
        prima_trecere := 1; --prima trecere la fiecare destinatie
        FETCH cursor_destinatie
        INTO cod_dest,nume_dest, nr_oferte;--incarcam linia in variabilele
mentionate
        EXIT WHEN cursor_destinatie%NOTFOUND; --iesim cand nu mai avem date
        DBMS_OUTPUT.PUT_LINE('Destinatia: ' || nume_dest);
        IF nr_oferte=0 THEN --daca nr de oferte este 0 pt destinatia
respectiva
             DBMS_OUTPUT.PUT_LINE('Nimeni nu a plecat in aceasta destinatie
 in luna a-' || luna || '-a.');
        ELSE
            OPEN cursor_oferta; --deschidem cursorul cursor_oferta
            LOOP
                FETCH cursor_oferta
                INTO cod_oferta, pret_of,cod_destinatie; --incarcam linia
in variabilele mentionate
```

```
                    EXIT WHEN cursor_oferta%NOTFOUND OR top = 3; --iesim cand
   nu avem date sau cand ajungem la top 3
                    IF cod_dest = cod_destinatie THEN
                       IF prima_trecere = 1 THEN --daca suntem la prima trecere
                          aux := pret_of;  --retinem pretul curent
                       END IF;
                       IF aux <> pret_of OR prima_trecere = 1 THEN
                             --daca suntem la prima trecere sau daca pretul
                             --anterior si pretul curent sunt diferite
                              DBMS_OUTPUT.PUT_LINE('Top ' || (top+1));
                              top := top + 1; --crestem in top
                              aux := pret_of; --retinem pretul curent
                       END IF;
                       prima_trecere := prima_trecere +1;
                       DBMS_OUTPUT.PUT_LINE('-> oferta: ' || cod_oferta || ' la
    pretul: ' || pret_of);
                    END IF;
                 END LOOP;
                 CLOSE cursor_oferta; --inchidem cursorul cursor_oferta
            END IF;
            DBMS_OUTPUT.NEW_LINE;
            END LOOP;
            CLOSE cursor_destinatie; --inchidem cursorul cursor_destinatie;
            EXCEPTION --tratam exceptii
                 WHEN exc_no_data_found THEN
                       RAISE_APPLICATION_ERROR(-20000, 'Nu s-au achizitionat
   oferte in luna respectiva.');
                 WHEN exc_luna THEN
                       RAISE_APPLICATION_ERROR(-20001,'Un an are 12 luni.
   Incercati din nou, introducand un numar intre 1 si 12.');
END p_ex7;
/

DECLARE
    v_luna NUMBER := &p_luna;
BEGIN
    p_ex7(v_luna);
END;
/
```

Am apelat procedura p_ex7 pentru luna 13.



Am apelat procedura p_ex7 pentru luna 7.

```
SQL Worksheet  History
Worksheet  Query Builder
            CLOSE cursor_destinatie; --inchidem cursorul cursor_destinatie;
            EXCEPTION --tratam exceptii
                WHEN exc_no_data_found THEN
                    RAISE_APPLICATION_ERROR(-20000, 'Nu s-au achizitionat oferte in
                WHEN exc_luna THEN
                    RAISE_APPLICATION_ERROR(-20001,'Un an are 12 luni.Incercati din
END p_ex7;
/

DECLARE
    v_luna NUMBER := &p_luna;
BEGIN
    p_ex7(v_luna);
END;
/
```

```
Script Output  Query Result
Task completed in 1.885 seconds

new:DECLARE
    v_luna NUMBER := 7;
BEGIN
    p_ex7(v_luna);
END;

PL/SQL procedure successfully completed.
```

```
Buffer Size: 20000
proiect1
In luna a-7-a s-au achizitionat in total 6 oferte.
In luna a-7-a se regasesc urmatoarele topuri:
Destinatia: Constanta
Top 1
-> oferta: 100 la pretul: 5500
-> oferta: 800 la pretul: 5500
Top 2
-> oferta: 600 la pretul: 5000
Top 3
-> oferta: 200 la pretul: 3800

Destinatia: Paris
Top 1
-> oferta: 300 la pretul: 7000

Destinatia: Barcelona
Nimeni nu a plecat in aceasta destinatie in luna a-7-a.

Destinatia: Venetia
Nimeni nu a plecat in aceasta destinatie in luna a-7-a.

Destinatia: Brasov
Nimeni nu a plecat in aceasta destinatie in luna a-7-a.

Destinatia: Cluj Napoca
Nimeni nu a plecat in aceasta destinatie in luna a-7-a.
```

Am apelat procedura p_ex7 pentru luna 4.

```
Worksheet  Query Builder
            CLOSE cursor_destinatie; --inchidem cursorul cursor_destinatie;
            EXCEPTION --tratam exceptii
                WHEN exc_no_data_found THEN
                    RAISE_APPLICATION_ERROR(-20000, 'Nu s-au achizitionat oferte in luna respectiva.');
                WHEN exc_luna THEN
                    RAISE_APPLICATION_ERROR(-20001,'Un an are 12 luni.Incercati din nou, introducand un numar intre 1 si 12.');
END p_ex7;
/
```

```
Script Output  Query Result
Task completed in 4.819 seconds
Error starting at line : 338 in command -
DECLARE
    v_luna NUMBER := 4;
BEGIN
    p_ex7(v_luna);
END;
Error report -
ORA-20000: Nu s-au achizitionat oferte in luna respectiva.
ORA-06512: at "PROIECT1.P_EX7", line 92
ORA-06512: at line 4
20000. 00000 -  "%s"
*Cause:    The stored procedure 'raise_application_error'
           was called which causes this error to be generated.
*Action:   Correct the problem as described in the error message or contact
           the application administrator or DBA for more information.
```

Activate Windo

8. Subprogram stocat de tip funcție

Definiți o funcție stocată care determină numărul de clienți care au colaborat cu agentul de vânzare al cărui nume este introdus de la tastatură. Afișați și numele clienților.

```
CREATE OR REPLACE FUNCTION f_ex8
   (nume_agent angajat.nume%TYPE)
   RETURN NUMBER IS
     nr_clienti NUMBER;
     TYPE tablou_imbricat IS TABLE OF client%ROWTYPE;
     info_clienti tablou_imbricat;
     exc_clienti EXCEPTION;
     BEGIN
         SELECT COUNT(id_client) --calculam nr de clienti
```

```
        INTO nr_clienti
        FROM angajat ang, obtine ob
        WHERE ang.id_angajat = ob.id_angajat(+)
              AND UPPER(nume) = UPPER(nume_agent)
              AND UPPER(tip_angajat) = UPPER('agent de vanzare')
        GROUP BY ang.id_angajat; --grupam dupa id-ul angajatului
        IF nr_clienti = 0 THEN --daca agentul nu are clienti
             RAISE exc_clienti;
        END IF;
        DBMS_OUTPUT.PUT_LINE('Agentul de vanzare ' || nume_agent || ' a
   colaborat cu:');
        --retinem informatiile despre clientii care au apelat
        --la un  agent intr-un tablou imbricat
        SELECT c.id_client,c.nume,c.prenume, c. CNP, c.telefon, c.adresa_email
        BULK COLLECT INTO info_clienti
        FROM angajat ang, obtine ob, client c --join pe 3 tabele diferite
        WHERE ang.id_angajat = ob.id_angajat AND c.id_client = ob.id_client
             AND UPPER(ang.nume) = UPPER(nume_agent)
             AND UPPER(tip_angajat) = UPPER('agent de vanzare');
             --conditiile de join si si conditia pentru a prelua
             --doar informatiile referitoare la angajatul dorit
        --afisam clientii
        FOR i in info_clienti.FIRST..info_clienti.last LOOP
            DBMS_OUTPUT.PUT_LINE(i || '. ' || info_clienti(i).nume || ' ' ||
   info_clienti(i).prenume);
        END LOOP;
        RETURN nr_clienti; -- returnam rezultatul
        EXCEPTION
           WHEN NO_DATA_FOUND THEN
              RAISE_APPLICATION_ERROR(-20000, 'Nu exista agentul de vanzare');
           WHEN TOO_MANY_ROWS THEN
              RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi agenti cu
   acelasi nume');
           WHEN exc_clienti THEN
              RAISE_APPLICATION_ERROR(-20002, 'Agentul nu a avut clienti.');
END f_ex8;
/

DECLARE
  v_nume angajat.nume%TYPE := Initcap('&p_nume');
  nr_clienti NUMBER := f_ex8(v_nume);
BEGIN
  IF nr_clienti = 1 THEN
    DBMS_OUTPUT.PUT_LINE('Agentul ' || v_nume || ' a coloborat cu un client.');
  ELSE
    DBMS_OUTPUT.PUT_LINE('Agentul ' || v_nume || ' a coloborat cu ' ||
   nr_clienti || ' clienti.');
  END IF;
END;
/
DECLARE
    nr_clienti NUMBER;
BEGIN
    nr_clienti := f_ex8('Adriana');
END;
/
```

Am apelat functia f_ex8 pentru agentul de vanzare Adriana.

```
DECLARE
    nr_clienti NUMBER;
BEGIN
    nr_clienti := f_ex8('Adriana');
END;
/
```

Script Output | Query Result | Task completed in 0.048 seconds

```
DECLARE
    nr_clienti NUMBER;
BEGIN
    nr_clienti := f_ex8('Adriana');
END;
Error report -
ORA-20000: Nu exista agentul de vanzare
ORA-06512: at "PROIECT1.F_EX8", line 34
ORA-06512: at line 4
20000. 00000 -  "%s"
*Cause:    The stored procedure 'raise_application_error'
           was called which causes this error to be generated.
*Action:   Correct the problem as described in the error message or contact
           the application administrator or DBA for more information.
```

Am apelat functia f_ex8 pentru agentul de vanzare Bucur.

```
        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                RAISE_APPLICATION_ERROR(-20000, 'Nu exista agentul de vanzare');
            WHEN TOO_MANY_ROWS THEN
                RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi agenti cu acelasi nume');
            WHEN exc_clienti THEN
                RAISE_APPLICATION_ERROR(-20002, 'Agentul nu a avut clienti.');
END f_ex8;
/

DECLARE
    v_nume angajat.nume%TYPE := Initcap('&p_nume');
    nr_clienti NUMBER := f_ex8(v_nume);
BEGIN
    IF nr_clienti = 1 THEN
        DBMS_OUTPUT.PUT_LINE('Agentul ' || v_nume || ' a coloborat cu un client.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Agentul ' || v_nume || ' a coloborat cu '|| nr_clienti || ' client
    END IF;
END;
/
```

Script Output | Query Result | Task completed in 10.252 seconds

```
PL/SQL procedure successfully completed.
```

proiect1

```
Agentul de vanzare Bucur a colaborat cu:
1. Badea Radu
2. Iancu Lorena
3. Stoica Ruxandra
4. Parvu Vlad
5. Modrescu Adrian
Agentul Bucur a coloborat cu 5 clienti.
```

Am apelat functia f_ex8 pentru agentul de vanzare Radu.

```
DECLARE
    v_nume angajat.nume%TYPE := Initcap('&p_nume');
    nr_clienti NUMBER := f_ex8(v_nume);
BEGIN
    IF nr_clienti = 1 THEN
        DBMS_OUTPUT.PUT_LINE('Agentul ' || v_nume || ' a coloborat cu un client.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Agentul ' || v_nume || ' a coloborat cu '|| nr_clienti || ' clienti.');
```

Script Output | Query Result | Task completed in 2.553 seconds

```
DECLARE
    v_nume angajat.nume%TYPE := Initcap('Radu');
    nr_clienti NUMBER := f_ex8(v_nume);
BEGIN
    IF nr_clienti = 1 THEN
        DBMS_OUTPUT.PUT_LINE('Agentul ' || v_nume || ' a coloborat cu un client.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Agentul ' || v_nume || ' a coloborat cu '|| nr_clienti || ' clienti.');
    END IF;
END;
Error report -
ORA-20002: Agentul nu a avut clienti.
ORA-06512: at "PROIECT1.F_EX8", line 38
ORA-06512: at line 3
```

Am apelat functia f_ex8 pentru agentul de vanzare Enache.

```
DECLARE
    v_nume angajat.nume%TYPE := Initcap('&p_nume');
    nr_clienti NUMBER := f_ex8(v_nume);
BEGIN
    IF nr_clienti = 1 THEN
       DBMS_OUTPUT.PUT_LINE('Agentul ' || v_nume || ' a coloborat cu un client.');
    ELSE
       DBMS_OUTPUT.PUT_LINE('Agentul ' || v_nume || ' a coloborat cu '|| nr_clienti || ' clienti.');
```

Script Output | Query Result

Task completed in 2.883 seconds

```
DECLARE
    v_nume angajat.nume%TYPE := Initcap('Enache');
    nr_clienti NUMBER := f_ex8(v_nume);
BEGIN
    IF nr_clienti = 1 THEN
       DBMS_OUTPUT.PUT_LINE('Agentul ' || v_nume || ' a coloborat cu un client.');
    ELSE
       DBMS_OUTPUT.PUT_LINE('Agentul ' || v_nume || ' a coloborat cu '|| nr_clienti || ' clienti.');
    END IF;
END;
Error report -
ORA-20001: Exista mai multi agenti cu acelasi nume
ORA-06512: at "PROIECT1.F_EX8", line 36
ORA-06512: at line 3
```

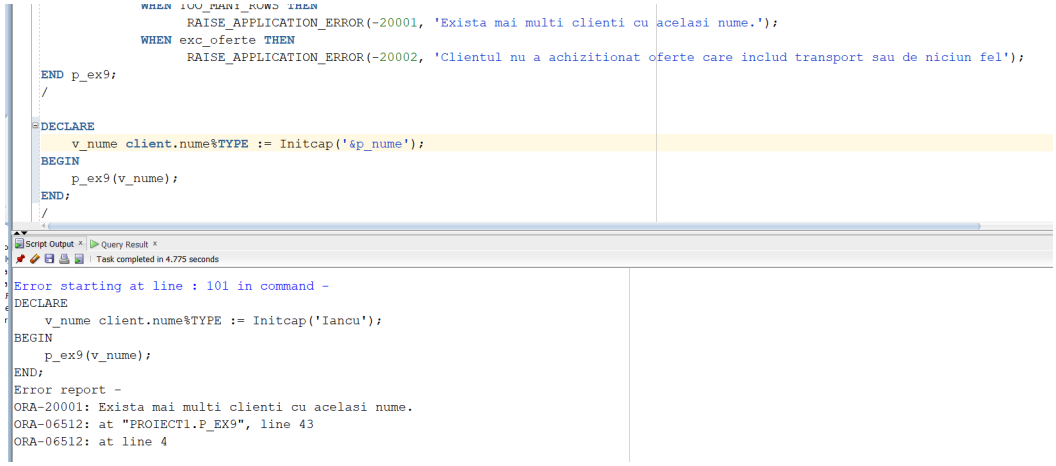
9. Subprogram stocat de tip procedură

Afișați pentru un client introdus de la tastatură firmele și tipurile de transport ale ofertelor achiziționate.

```
CREATE OR REPLACE PROCEDURE p_ex9
    (nume_client client.nume%TYPE)
    IS
      --am luat un cursor pentru oferte
      CURSOR cursor_oferta IS
            SELECT o.id_oferta,tip_oferta,mijloc_de_transport,denumire_firma
            FROM client c , oferta o ,obtine ob, transport t,firma f
            WHERE c.id_client = ob.id_client AND o.id_oferta = ob.id_oferta
                     AND t.id_transport = o.id_transport
                     AND f.id_firma = t.id_firma
                     AND UPPER(nume) = UPPER(nume_client);
      nr_oferte NUMBER(2);
      prenume_client client.prenume%TYPE;
      exc_oferte EXCEPTION;
      BEGIN
          --am retinut in variabile prenumele clientului si nr de oferte
   achizitionate
          SELECT prenume, NVL((SELECT COUNT(o.id_oferta)
                       FROM client c , oferta o ,obtine ob, transport t,firma f
                       WHERE c.id_client = ob.id_client
                             AND o.id_oferta = ob.id_oferta
                             AND t.id_transport = o.id_transport
                             AND f.id_firma = t.id_firma
                             AND UPPER(c.nume) = UPPER(nume_client)
                       GROUP BY nume), 0)
          INTO prenume_client, nr_oferte
          FROM client
          WHERE UPPER(nume) = UPPER(nume_client);
          IF nr_oferte = 0 THEN --daca clientul nu a achizitionat nicio oferta
```

```
            RAISE exc_oferte;
        ELSIF nr_oferte = 1 THEN --daca clientul a achizitionat doar o oferta
              DBMS_OUTPUT.PUT_LINE('Clientul ' || nume_client || ' ' ||
prenume_client || ' a achizitionat o singura oferta care include transport:'
);
        ELSE
             DBMS_OUTPUT.PUT_LINE('Clientul ' || nume_client || ' ' ||
prenume_client || ' a achizitionat ' || nr_oferte || ' oferte care includ
transport:');
        END IF;
        --afisam ofertele
        FOR i IN cursor_oferta LOOP
             DBMS_OUTPUT.PUT_LINE('-> Oferta ' || i.id_oferta || ' -> este de
 tip ' || i.tip_oferta);
             DBMS_OUTPUT.PUT_LINE('              -> s-a ajuns la destinatie
cu ' || i.mijloc_de_transport);
             DBMS_OUTPUT.PUT_LINE('              -> firma care s-a ocupat de
transport este: ' || i.denumire_firma);
             DBMS_OUTPUT.PUT_LINE('
--------------------------------------------------------------------------------
');
        END LOOP;
        EXCEPTION --tratam exceptii
           WHEN NO_DATA_FOUND THEN
                 RAISE_APPLICATION_ERROR(-20000, 'Nu exista clientul.');
           WHEN TOO_MANY_ROWS THEN
                 RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi clienti cu
 acelasi nume.');
           WHEN exc_oferte THEN
                 RAISE_APPLICATION_ERROR(-20002, 'Clientul nu a achizitionat
oferte care includ transport sau de niciun fel');
END p_ex9;
/

DECLARE
    v_nume client.nume%TYPE := Initcap('&p_nume');
BEGIN
    p_ex9(v_nume);
END;
/
```

Am apelat procedura p_ex9 pentru clientul Iancu.



Am apelat procedura p_ex9 pentru clientul Laura.

```
DECLARE
    v_nume client.nume%TYPE := Initcap('&p_nume');
BEGIN
    p_ex9(v_nume);
END;
/
```

Script Output | Task completed in 2.85 seconds

```
Error starting at line : 101 in command -
DECLARE
    v_nume client.nume%TYPE := Initcap('Laura');
BEGIN
    p_ex9(v_nume);
END;
Error report -
ORA-20000: Nu exista clientul.
ORA-06512: at "PROIECT1.P_EX9", line 41
ORA-06512: at line 4
20000. 00000 -  "%s"
*Cause:    The stored procedure 'raise_application_error'
           was called which causes this error to be generated.
*Action:   Correct the problem as described in the error message or contact
           the application administrator or DBA for more information.
```

Am apelat procedura p_ex9 pentru clientul Preda.

```
        FOR i IN cursor_oferta LOOP
            DBMS_OUTPUT.PUT_LINE('-> Oferta ' || i.id_oferta || ' -> e
            DBMS_OUTPUT.PUT_LINE('              -> s-a ajuns la destin
            DBMS_OUTPUT.PUT_LINE('              -> firma care s-a ocup
            DBMS_OUTPUT.PUT_LINE('-----------------------------------
        END LOOP;
        EXCEPTION --tratam exceptii
           WHEN NO_DATA_FOUND THEN
                RAISE_APPLICATION_ERROR(-20000, 'Nu exista clientul.')
           WHEN TOO_MANY_ROWS THEN
                RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi clie
           WHEN exc_oferte THEN
                RAISE_APPLICATION_ERROR(-20002, 'Clientul nu a achizit
END p_ex9;
/
```

Script Output | Query Result | Task completed in 3.502 seconds

```
new:DECLARE
    v_nume client.nume%TYPE := Initcap('Preda');
BEGIN
    p_ex9(v_nume);
END;

PL/SQL procedure successfully completed.
```

```
Clientul Preda Bianca a achizitionat 2 oferte care includ transport:
-> Oferta 800 -> este de tip sejur
              -> s-a ajuns la destinatie cu avion
              -> firma care s-a ocupat de transport este: TAROM
-----------------------------------------------------------------
-> Oferta 1200 -> este de tip city break
               -> s-a ajuns la destinatie cu avion
               -> firma care s-a ocupat de transport este: Air Bucharest
-----------------------------------------------------------------
```

Am apelat procedura p_ex9 pentru clientul Rebegea.

```
Worksheet  Query Builder
            WHEN TOO_MANY_ROWS THEN
                 RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi clienti cu acelasi nume.');
            WHEN exc_oferte THEN
                 RAISE_APPLICATION_ERROR(-20002, 'Clientul nu a achizitionat oferte care includ transport sau de niciun fel');
END p_ex9;
/

DECLARE
    v_nume client.nume%TYPE := Initcap('&p_nume');
BEGIN
    p_ex9(v_nume);
END;
/
```

```
Script Output  Query Result
Task completed in 4.107 seconds
Error starting at line : 101 in command -
DECLARE
    v_nume client.nume%TYPE := Initcap('Rebegea');
BEGIN
    p_ex9(v_nume);
END;
Error report -
ORA-20002: Clientul nu a achizitionat oferte care includ transport sau de niciun fel
ORA-06512: at "PROIECT1.P_EX9", line 45
ORA-06512: at line 4
```

10. Trigger de tip LMD la nivel de comandă

Să se scrie un trigger TRIG_EX10, care să permită lucrul asupra tabelului contract_angajat doar în zilele lucrătoare și între orele 9-20.

```
CREATE OR REPLACE TRIGGER trig_ex10
BEFORE INSERT OR DELETE OR UPDATE on contract_angajat
BEGIN
  --conditia pentru permisiuni
  IF (TO_CHAR(SYSDATE,'D') <= 1 AND TO_CHAR(SYSDATE,'D') >6)
     OR (TO_CHAR(SYSDATE,'HH24') NOT BETWEEN 9 AND 20) THEN
     --am luat pe cazuri(in caz de inserare,stergere sau actualizare)
    IF INSERTING THEN
       RAISE_APPLICATION_ERROR(-20001,'Inserarea in tabel se poate face doar in
    cursul zilelor lucratoare si in intervalul 9-20');
    ELSIF DELETING THEN
       RAISE_APPLICATION_ERROR(-20002,'Stergerea din tabel se poate face doar
   in cursul zilelor lucratoare si in intervalul 9-20');
    ELSE
       RAISE_APPLICATION_ERROR(-20003,'Actualizarea tabelului se poate face
   doar in cursul zilelor lucratoare si in intervalul 9-20');
    END IF;
  END IF;
END;/
drop trigger trig_ex10;

update contract_angajat
set experienta = 3
where id_contract = 7;

select *
from contract_angajat
where id_contract = 7;
```

Am creat triggerul trig_ex10 și am apelat în afara orelor permise.

```
CREATE OR REPLACE TRIGGER trig_ex10
BEFORE INSERT OR DELETE OR UPDATE on contract_angajat
BEGIN
  --conditia pentru permisiuni
  IF (TO_CHAR(SYSDATE,'D') <= 1 AND TO_CHAR(SYSDATE,'D') >6)
     OR (TO_CHAR(SYSDATE,'HH24') NOT BETWEEN 9 AND 20) THEN
     --am luat pe cazuri erorile(in caz de inserare,stergere sau actualizare)
    IF INSERTING THEN
       RAISE_APPLICATION_ERROR(-20001,'Inserarea in tabel se poate face doar in cursul z
             lucratoare si in intervalul 9-20');
    ELSIF DELETING THEN
       RAISE_APPLICATION_ERROR(-20002,'Stergerea din tabel se poate face doar in cursul z
```

Script Output | Task completed in 0.1 seconds

```
Error starting at line : 32 in command -
update contract_angajat
set experienta = 3
where id_contract = 7
Error report -
ORA-20003: Actualizarea tabelului se poate face doar in cursul zilelor
             lucratoare si in intervalul 9-20
ORA-06512: at "PROIECT1.TRIG_EX10", line 13
ORA-04088: error during execution of trigger 'PROIECT1.TRIG_EX10'
```

Am apelat pentru contractul cu id-ul 7.
Așa erau datele din tabel înainte de crearea triggerului.

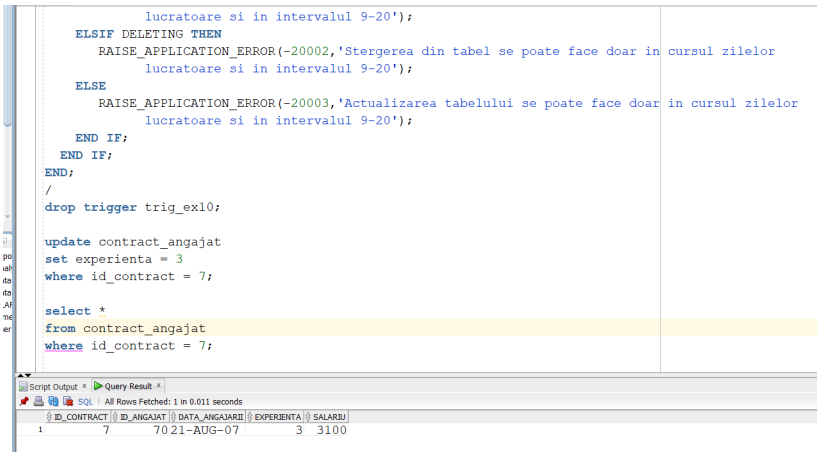
```
             lucratoare si in intervalul 9-20');
    ELSIF DELETING THEN
       RAISE_APPLICATION_ERROR(-20002,'Stergerea din tabel se poate face doar in cursul zilelor
             lucratoare si in intervalul 9-20');
    ELSE
       RAISE_APPLICATION_ERROR(-20003,'Actualizarea tabelului se poate face doar in cursul zilelor
             lucratoare si in intervalul 9-20');
    END IF;
  END IF;
END;
/
drop trigger trig_ex10;

update contract_angajat
set experienta = 3
where id_contract = 7;

select *
from contract_angajat
where id_contract = 7;
```

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.008 seconds

	ID_CONTRACT	ID_ANGAJAT	DATA_ANGAJARII	EXPERIENTA	SALARIU
1	7	70	21-AUG-07	2	3100

Datele au devenit după apelare :



11. Trigger de tip LMD la nivel de linie

Să se scrie un trigger TRIG_EX11, care să nu permită mărirea comisionului angajaților care nu au cel puțin 10 ani de experiență.

```
CREATE OR REPLACE TRIGGER trig_ex11
BEFORE UPDATE OF comision ON angajat
FOR EACH ROW
DECLARE
   experienta_angajat contract_angajat.experienta%TYPE;
BEGIN
   SELECT experienta
   INTO experienta_angajat
   FROM contract_angajat
   WHERE id_angajat IN (:NEW.id_angajat, :OLD.id_angajat);
   IF experienta_angajat < 10 AND :NEW.comision > NVL(:OLD.comision,0)  THEN
      RAISE_APPLICATION_ERROR(-20001, 'Nu puteti mari comisionul.Angajatul are
   mai putin de 10 ani de experienta.');
   END IF;
END;
/

drop trigger trig_ex11;

update angajat
set comision = 2
where id_angajat = 20;

update angajat
set comision = 6
where id_angajat = 30;

select ang.id_angajat, comision,experienta
from angajat ang, contract_angajat c
where ang.id_angajat = c.id_angajat;
```

Am creat triggerul trig_ex11. Am apelat pentru un set de date care nu respectă cerințele.

```
    FROM contract_angajat
    WHERE id_angajat IN (:NEW.id_angajat, :OLD.id_angajat);
    IF experienta_angajat < 10 AND :NEW.comision > NVL(:OLD.comision,0)  THEN
       RAISE_APPLICATION_ERROR(-20001, 'Nu puteti mari comisionul.Angajatul are mai putin de 10 ani de experienta.');
    END IF;
END;
/

drop trigger trig_ex11;

update angajat
set comision = 2
where id_angajat = 20;
```

Script Output | Query Result | Task completed in 0.063 seconds

```
Error starting at line : 62 in command -
update angajat
set comision = 2
where id_angajat = 20
Error report -
ORA-20001: Nu puteti mari comisionul.Angajatul are mai putin de 10 ani de experienta.
ORA-06512: at "PROIECT1.TRIG_EX11", line 9
ORA-04088: error during execution of trigger 'PROIECT1.TRIG_EX11'
```

Am apelat pentru angajatul cu id-ul 20.
Așa erau datele din tabel înainte de crearea triggerului.

```
select ang.id_angajat, comision,experienta
from angajat ang, contract_angajat c
where ang.id_angajat = c.id_angajat;

--ex12
```

Script Output | Query Result | All Rows Fetched: 9 in 0.01 seconds

	ID_ANGAJAT	COMISION	EXPERIENTA
1	70	2	2
2	80	1	4
3	90	3	5
4	10	(null)	10
5	20	(null)	5
6	30	5	20
7	40	(null)	15
8	50	(null)	(null)
9	60	4	13

Datele au devenit după apelare(se actualizează,deoarece respectă condițiile) :

```
update angajat
set comision = 6
where id_angajat = 30;


rollback;
select ang.id_angajat, comision,experienta
from angajat ang, contract_angajat c
where ang.id_angajat = c.id_angajat;
```

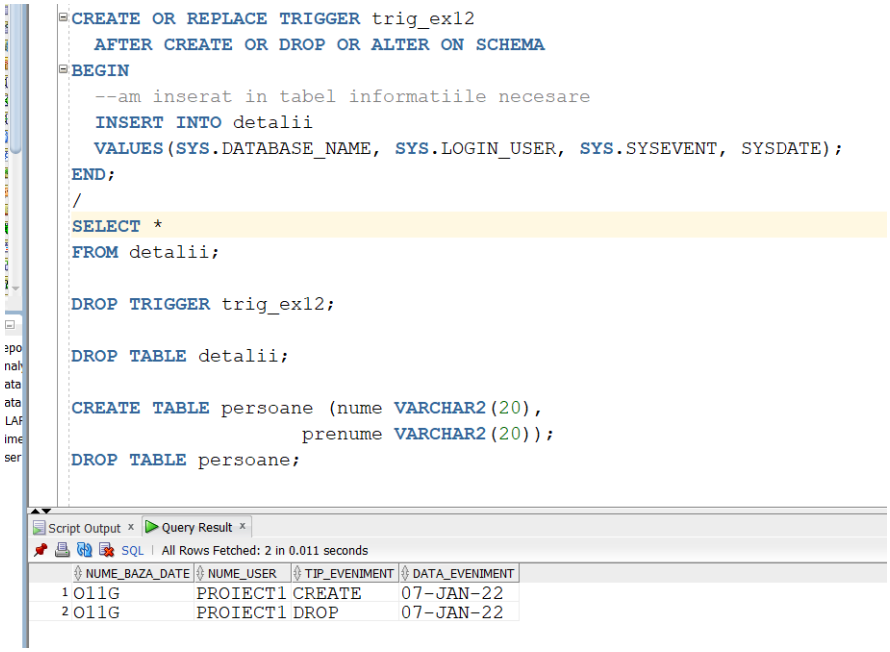
Script Output | Query Result | All Rows Fetched: 9 in 0.009 seconds

	ID_ANGAJAT	COMISION	EXPERIENTA
1	70	2	2
2	80	1	4
3	90	3	5
4	10	(null)	10
5	20	(null)	5
6	30	6	20
7	40	(null)	15
8	50	(null)	(null)
9	60	4	13

12. Trigger de tip LDD.

Definiți un trigger TRIG_EX12 care să introducă date în tabelul DETALII după ce utilizatorul a folosit o comandă LDD.

```
--am creat tabelul detalii
CREATE TABLE detalii
   (nume_baza_date VARCHAR2(30),
    nume_user VARCHAR2(30),
    tip_eveniment VARCHAR2(30),
    data_eveniment DATE);

CREATE OR REPLACE TRIGGER trig_ex12
  AFTER CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
  --am inserat in tabel informatiile necesare
  INSERT INTO detalii
  VALUES(SYS.DATABASE_NAME, SYS.LOGIN_USER, SYS.SYSEVENT, SYSDATE);
END;
/
SELECT *
FROM detalii;

DROP TRIGGER trig_ex12;

DROP TABLE detalii;

CREATE TABLE persoane (nume VARCHAR2(20),
                       prenume VARCHAR2(20));
DROP TABLE persoane;
```



13. Pachet - obiecte definite până acum

```
CREATE OR REPLACE PACKAGE pachet_ex13 AS
    PROCEDURE p_ex6 (v_nume client.nume%TYPE);
    PROCEDURE p_ex7 (luna NUMBER);
    FUNCTION f_ex8 (nume_agent angajat.nume%TYPE) RETURN NUMBER;
    PROCEDURE p_ex9 (nume_client client.nume%TYPE);
END pachet_ex13;
/

CREATE OR REPLACE PACKAGE BODY pachet_ex13 AS
    PROCEDURE p_ex6 (v_nume client.nume%TYPE) IS
          nr_oferte NUMBER;
          nr_fac NUMBER;
          cod_client client.id_client%TYPE;
          TYPE tablou_imbricat IS TABLE OF obtine%ROWTYPE;
          info_of tablou_imbricat;
          TYPE tablou_indexat IS TABLE OF facilitati%ROWTYPE
                         INDEX BY BINARY_INTEGER;
          fac_camera tablou_indexat;
          exc_oferte EXCEPTION;
          exc_fac EXCEPTION;
        BEGIN
          --pentru a intra pe exceptia de no_data_found trebuie sa verificam
          --daca numele exista in tabel, daca nu faceam acest select se facea
          --confuzie intre clientii care nu exista si clientii care nu au
          --achizitionat oferte(nu intra niciodata pe no_data_found,
          --ci intra pe exc_oferte)
          SELECT id_client
          INTO cod_client
          FROM client
          WHERE UPPER(nume) = UPPER(v_nume);
          --retinem detaliile despre oferte
          SELECT o.id_client,o.id_angajat,o.id_oferta,o.id_camera,data_start,
   data_end
          BULK COLLECT INTO info_of
          FROM client c, obtine o
          WHERE UPPER(c.nume) = UPPER(v_nume) and o.id_client = c.id_client;
          nr_oferte := info_of.COUNT;
          IF nr_oferte = 0 THEN --daca clientul nu a achizitionat nicio oferta
               RAISE exc_oferte;
          ELSIF nr_oferte = 1 THEN --daca clientul a achizionat doar o oferta
              DBMS_OUTPUT.PUT_LINE('Clientul ' || v_nume || ' a achizitionat o
   oferta.');
              DBMS_OUTPUT.PUT_LINE('Oferta achizitionata este:');
          ELSE
              DBMS_OUTPUT.PUT_LINE('Clientul ' || v_nume || ' a achizitionat '
   || nr_oferte || ' oferte.');
              DBMS_OUTPUT.PUT_LINE('Ofertele achizitionate sunt:');
          END IF;
          --parcurgem fiecare oferta
          FOR i IN info_of.FIRST..info_of.LAST LOOP
             --afisam detaliile despre oferta
             DBMS_OUTPUT.PUT_LINE(i || '. Oferta ' || info_of(i).id_oferta || '
    -> incepe pe data de ' || info_of(i).data_start);
```

```
          DBMS_OUTPUT.PUT_LINE('              -> se termina pe data de ' ||
info_of(i).data_end);
          DBMS_OUTPUT.PUT_LINE('              -> este rezervata pe camera '
|| info_of(i).id_camera);
          --retinem facilitatile camerei
          SELECT f.id_facilitate,denumire
          BULK COLLECT INTO fac_camera
          FROM obtine o, contine_fac cf, facilitati f
          WHERE id_oferta = info_of(i).id_oferta
                and o.id_camera = cf.id_camera
                and cf.id_facilitate = f.id_facilitate;
          nr_fac := fac_camera.COUNT;
          IF nr_fac = 0 THEN --in cazul in care camera nu are facilitati
             RAISE exc_fac;
          ELSIF nr_fac = 1 THEN --in cazul in care camera are doar o
facilitate
             DBMS_OUTPUT.PUT_LINE('Camera ' || info_of(i).id_camera || ' are
 o singura facilitate:');
          ELSE
             DBMS_OUTPUT.PUT_LINE('Camera ' || info_of(i).id_camera || ' are
 ' || nr_fac || ' facilitati:');
          END IF;
          --afisam facilitatile camerei
          FOR j in fac_camera.FIRST..fac_camera.last LOOP
              DBMS_OUTPUT.PUT_LINE('-> ' || fac_camera(j).denumire);
          END LOOP;
          DBMS_OUTPUT.NEW_LINE;
      END LOOP;
      EXCEPTION --tratam exceptii
          WHEN NO_DATA_FOUND THEN
                RAISE_APPLICATION_ERROR(-20000, 'Nu exista clientul.');
          WHEN TOO_MANY_ROWS THEN
                RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi clienti cu
 acelasi nume.');
          WHEN exc_oferte THEN
                RAISE_APPLICATION_ERROR(-20002, 'Clientul nu a achizitionat
nicio oferta.');
          WHEN exc_fac THEN
                RAISE_APPLICATION_ERROR(-20003, 'Camera nu are facilitati.')
;
 END p_ex6;

 PROCEDURE p_ex7 (luna NUMBER) IS
         --am luat un cursor pentru destinatii
         --daca nu faceam cu UNION nu lua destinatiile care nu au avut nicio
         --oferta in luna introdusa de la tastatura
         CURSOR cursor_destinatie IS
             SELECT d.id_destinatie, MAX(oras),COUNT(o.id_oferta)
             FROM destinatie d, oferta o, obtine ob
             WHERE d.id_destinatie = o.id_destinatie
                   AND TO_CHAR(data_start, 'MM') = luna
                   AND o.id_oferta = ob.id_oferta
             GROUP BY d.id_destinatie
             UNION
             SELECT id_destinatie,oras,0
             FROM destinatie
```

```
            WHERE id_destinatie NOT IN (SELECT d.id_destinatie
                                        FROM destinatie d,oferta o,obtine ob
                                        WHERE d.id_destinatie = o.id_destinatie
                                        AND TO_CHAR(data_start, 'MM') = luna
                                        AND o.id_oferta = ob.id_oferta);
        --am luat un cursor pentru oferte
        CURSOR cursor_oferta IS
            SELECT o.id_oferta, pret, id_destinatie
            FROM obtine o, oferta oft
            WHERE TO_CHAR(data_start, 'MM') = luna
                  AND o.id_oferta = oft.id_oferta
            ORDER BY pret DESC;
        cod_dest destinatie.id_destinatie%TYPE;
        nume_dest destinatie.oras%TYPE;
        nr_oferte NUMBER(5);
        nr_oferte_total NUMBER(5);
        cod_oferta oferta.id_oferta%TYPE;
        pret_of oferta.pret%TYPE;
        cod_destinatie oferta.id_destinatie%TYPE;
        top number(1) :=0; --top
        aux number; --retinem pretul anterior
        prima_trecere number :=1;
        exc_luna EXCEPTION;
        exc_no_data_found EXCEPTION;
        BEGIN
           --verificam daca luna introdusa poate exista
           IF luna <= 0 OR luna > 12 THEN
               RAISE exc_luna;
           END IF;
           --calculam nr total de oferte achizitionate in luna respectiva
           SELECT count(data_start)
           INTO nr_oferte_total
           FROM obtine
           WHERE TO_CHAR(data_start, 'MM') = luna;
           --verificam daca in tabel exista vreo oferta
           --achizitionata in luna introdusa
           IF nr_oferte_total = 0 THEN
               RAISE exc_no_data_found;
           ELSIF nr_oferte_total = 1 THEN --daca exista doar o oferta
achizitionata
               DBMS_OUTPUT.PUT_LINE('In luna a-' || luna || '-a s-a
achizitionat doar o oferta.');
           ELSE
               DBMS_OUTPUT.PUT_LINE('In luna a-' || luna || '-a s-au
achizitionat in total ' || nr_oferte_total || ' oferte.');
           END IF;
           OPEN cursor_destinatie; --deschidem cursorul destinatie
           DBMS_OUTPUT.PUT_LINE('In luna a-' || luna || '-a se regasesc
urmatoarele topuri:' );
           LOOP
            top := 0; --topul porneste de la 0 la fiecare destinatie
            prima_trecere := 1; --prima trecere la fiecare destinatie
            FETCH cursor_destinatie
            INTO cod_dest,nume_dest, nr_oferte;--incarcam linia in
variabilele mentionate
            EXIT WHEN cursor_destinatie%NOTFOUND; --iesim cand nu mai avem
```

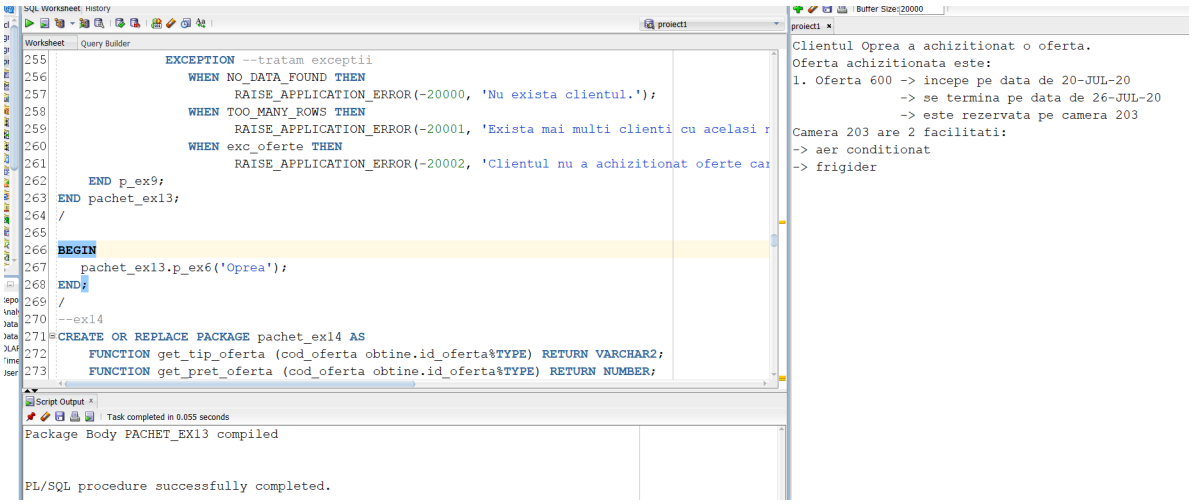
```
date
            DBMS_OUTPUT.PUT_LINE('Destinatia: ' || nume_dest);
            IF nr_oferte=0 THEN --daca nr de oferte este 0 pt destinatia
respectiva
                 DBMS_OUTPUT.PUT_LINE('Nimeni nu a plecat in aceasta
destinatie in luna a-' || luna || '-a.');
            ELSE
                OPEN cursor_oferta; --deschidem cursorul cursor_oferta
                LOOP
                    FETCH cursor_oferta
                    INTO cod_oferta, pret_of,cod_destinatie; --incarcam
linia in variabilele mentionate
                    EXIT WHEN cursor_oferta%NOTFOUND OR top = 3; --iesim
cand nu avem date sau cand ajungem la top 3
                    IF cod_dest = cod_destinatie THEN
                       IF prima_trecere = 1 THEN --daca suntem la prima
trecere
                          aux := pret_of;  --retinem pretul curent
                       END IF;
                       IF aux <> pret_of OR prima_trecere = 1 THEN
                       --daca suntem la prima trecere sau daca pretul
                       --anterior si pretul curent sunt diferite
                              DBMS_OUTPUT.PUT_LINE('Top ' || (top+1));
                              top := top + 1; --crestem in top
                              aux := pret_of; --retinem pretul curent
                       END IF;
                       prima_trecere := prima_trecere +1;
                       DBMS_OUTPUT.PUT_LINE('-> oferta: ' || cod_oferta ||
' la pretul: ' || pret_of);
                    END IF;
                END LOOP;
                CLOSE cursor_oferta; --inchidem cursorul cursor_oferta
            END IF;
            DBMS_OUTPUT.NEW_LINE;
            END LOOP;
            CLOSE cursor_destinatie; --inchidem cursorul cursor_destinatie;
            EXCEPTION --tratam exceptii
                 WHEN exc_no_data_found THEN
                       RAISE_APPLICATION_ERROR(-20000, 'Nu s-au
achizitionat oferte in luna respectiva.');
                 WHEN exc_luna THEN
                       RAISE_APPLICATION_ERROR(-20001,'Un an are 12 luni.
Incercati din nou, introducand un numar intre 1 si 12.');
 END p_ex7;

 FUNCTION f_ex8 (nume_agent angajat.nume%TYPE) RETURN NUMBER IS
      nr_clienti NUMBER;
      TYPE tablou_imbricat IS TABLE OF client%ROWTYPE;
      info_clienti tablou_imbricat;
      exc_clienti EXCEPTION;
      BEGIN
          SELECT COUNT(id_client) --calculam nr de clienti
          INTO nr_clienti
          FROM angajat ang, obtine ob
          WHERE ang.id_angajat = ob.id_angajat(+)
                AND UPPER(nume) = UPPER(nume_agent)
```

```
                AND UPPER(tip_angajat) = UPPER('agent de vanzare')
          GROUP BY ang.id_angajat; --grupam dupa id-ul angajatului
          IF nr_clienti = 0 THEN --daca agentul nu are clienti
               RAISE exc_clienti;
          END IF;
          DBMS_OUTPUT.PUT_LINE('Agentul de vanzare ' || nume_agent || ' a
colaborat cu:');
          --retinem informatiile despre clientii care
          --au apelat la un agent intr-un tablou imbricat
          SELECT c.id_client, c.nume, c.prenume, c. CNP, c.telefon, c.
adresa_email
          BULK COLLECT INTO info_clienti
          FROM angajat ang, obtine ob, client c --join pe 3 tabele diferite
          WHERE ang.id_angajat = ob.id_angajat
               AND c.id_client = ob.id_client
               AND UPPER(ang.nume) = UPPER(nume_agent)
               AND UPPER(tip_angajat) = UPPER('agent de vanzare');
               --conditiile de join si si conditia pentru a prelua
               --doar informatiile referitoare la angajatul dorit
          --afisam clientii
          FOR i in info_clienti.FIRST..info_clienti.last LOOP
              DBMS_OUTPUT.PUT_LINE(i || '. ' || info_clienti(i).nume || ' '
|| info_clienti(i).prenume);
          END LOOP;
          RETURN nr_clienti; -- returnam rezultatul
          EXCEPTION
             WHEN NO_DATA_FOUND THEN
                RAISE_APPLICATION_ERROR(-20000, 'Nu exista agentul de
vanzare');
             WHEN TOO_MANY_ROWS THEN
                RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi agenti cu
acelasi nume');
             WHEN exc_clienti THEN
                RAISE_APPLICATION_ERROR(-20002, 'Agentul nu a avut clienti.'
);
 END f_ex8;


 PROCEDURE p_ex9 (nume_client client.nume%TYPE) IS
       --am luat un cursor pentru oferte
       CURSOR cursor_oferta IS
         SELECT o.id_oferta, tip_oferta,mijloc_de_transport, denumire_firma
         FROM client c , oferta o ,obtine ob, transport t,firma f
         WHERE c.id_client = ob.id_client AND o.id_oferta = ob.id_oferta
                 AND t.id_transport = o.id_transport
                 AND f.id_firma = t.id_firma
                 AND UPPER(nume) = UPPER(nume_client);
       nr_oferte NUMBER(2);
       prenume_client client.prenume%TYPE;
       exc_oferte EXCEPTION;
       BEGIN
           --am retinut in variabile prenumele clientului
           --si nr de oferte achizitionate
           SELECT prenume, NVL((SELECT COUNT(o.id_oferta)
                 FROM client c , oferta o ,obtine ob, transport t,firma f
                       WHERE c.id_client = ob.id_client
```

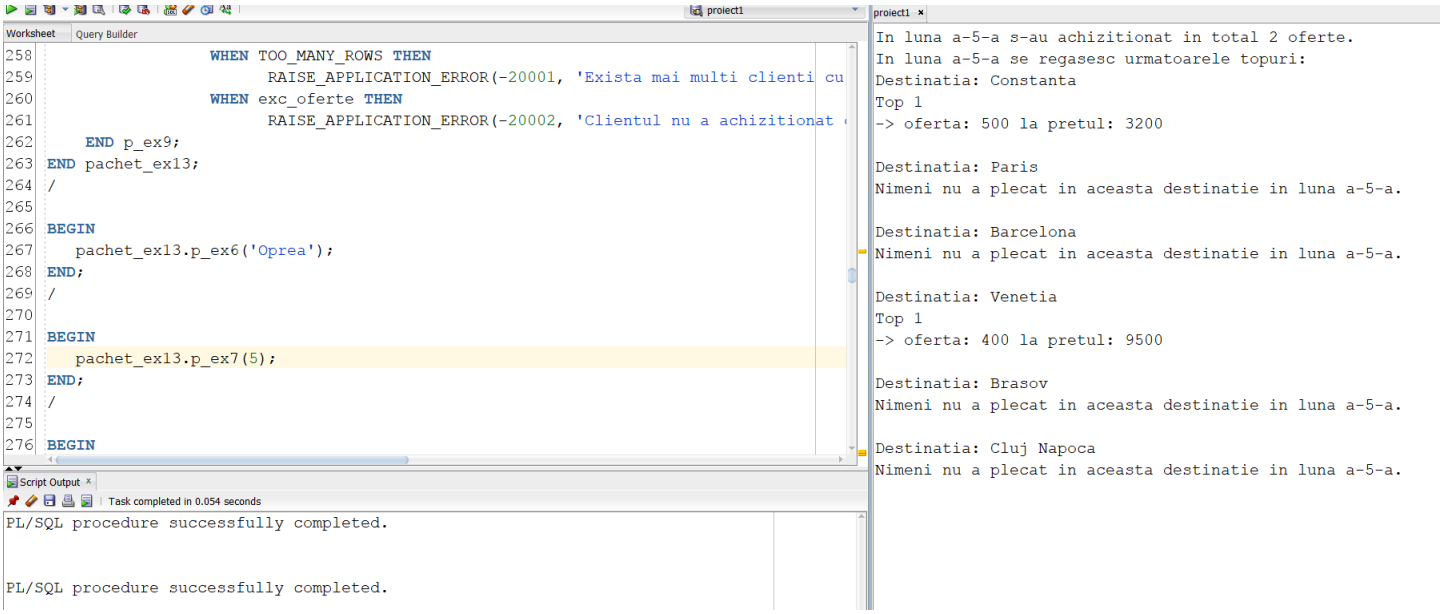
```
                            AND o.id_oferta = ob.id_oferta
                            AND t.id_transport = o.id_transport
                            AND f.id_firma = t.id_firma
                            AND UPPER(c.nume) = UPPER(nume_client)
                        GROUP BY nume), 0)
            INTO prenume_client, nr_oferte
            FROM client
            WHERE UPPER(nume) = UPPER(nume_client);
            IF nr_oferte = 0 THEN --daca clientul nu a achizitionat nicio
   oferta
                RAISE exc_oferte;
            ELSIF nr_oferte = 1 THEN --daca clientul a achizitionat doar o
   oferta
                 DBMS_OUTPUT.PUT_LINE('Clientul ' || nume_client || ' ' ||
   prenume_client || ' a achizitionat o singura oferta care include transport:'
   );
            ELSE
                 DBMS_OUTPUT.PUT_LINE('Clientul ' || nume_client || ' ' ||
   prenume_client || ' a achizitionat ' || nr_oferte || ' oferte care includ
   transport:');
            END IF;
            --afisam ofertele
            FOR i IN cursor_oferta LOOP
                 DBMS_OUTPUT.PUT_LINE('-> Oferta ' || i.id_oferta || ' ->
   este de tip ' || i.tip_oferta);
                 DBMS_OUTPUT.PUT_LINE('              -> s-a ajuns la
   destinatie cu ' || i.mijloc_de_transport);
                 DBMS_OUTPUT.PUT_LINE('              -> firma care s-a ocupat
    de transport este: ' || i.denumire_firma);
                 DBMS_OUTPUT.PUT_LINE('
   ------------------------------------------------------------------------------
   ');
            END LOOP;
            EXCEPTION --tratam exceptii
               WHEN NO_DATA_FOUND THEN
                     RAISE_APPLICATION_ERROR(-20000, 'Nu exista clientul.');
               WHEN TOO_MANY_ROWS THEN
                     RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi
   clienti cu acelasi nume.');
               WHEN exc_oferte THEN
                     RAISE_APPLICATION_ERROR(-20002, 'Clientul nu a
   achizitionat oferte care includ transport sau de niciun fel');
    END p_ex9;
END pachet_ex13;
/

BEGIN
   pachet_ex13.p_ex6('Oprea');
END;
/

BEGIN
   pachet_ex13.p_ex7(5);
END;
/
```

```
DECLARE
    nr_clienti NUMBER := pachet_ex13.f_ex8('Bucur');
BEGIN
    IF nr_clienti = 1 THEN
       DBMS_OUTPUT.PUT_LINE('Agentul a coloborat cu un client.');
    ELSE
       DBMS_OUTPUT.PUT_LINE('Agentul a coloborat cu '|| nr_clienti || ' clienti
   .');
    END IF;
END;
/

BEGIN
   pachet_ex13.p_ex9('Modrescu');
END;
/
```

Am apelat din pachet procedura p_ex6 pentru clientul Oprea.



Am apelat din pachet procedura p_ex7 pentru luna 5.



Am apelat din pachet functia f_ex8 pentru agentul de vânzare Bucur.

```
END;
/

BEGIN
   pachet_ex13.p_ex7(5);
END;
/

DECLARE
    nr_clienti NUMBER := pachet_ex13.f_ex8('Bucur');
BEGIN
    IF nr_clienti = 1 THEN
       DBMS_OUTPUT.PUT_LINE('Agentul a coloborat cu un client.');
    ELSE
       DBMS_OUTPUT.PUT_LINE('Agentul a coloborat cu '|| nr_clienti || ' clienti.');
    END IF;
END;
/

BEGIN
   pachet_ex13.p_ex6('Oprea');
END;
/
```

```
Agentul de vanzare Bucur a colaborat cu:
1. Badea Radu
2. Iancu Lorena
3. Stoica Ruxandra
4. Parvu Vlad
5. Modrescu Adrian
Agentul a coloborat cu 5 clienti.
```

```
PL/SQL procedure successfully completed.
```

Am apelat din pachet procedura p_ex6 pentru clientul Modrescu.

```
BEGIN
   pachet_ex13.p_ex7(5);
END;
/

DECLARE
    nr_clienti NUMBER := pachet_ex13.f_ex8('Bucur');
BEGIN
    IF nr_clienti = 1 THEN
       DBMS_OUTPUT.PUT_LINE('Agentul a coloborat cu un client.')
    ELSE
       DBMS_OUTPUT.PUT_LINE('Agentul a coloborat cu '|| nr_clien
    END IF;
END;
/

BEGIN
   pachet_ex13.p_ex9('Modrescu');
END;
/
--ex14
```

```
PL/SQL procedure successfully completed.
```

```
Clientul Modrescu Adrian a achizitionat o singura oferta care include transport:
-> Oferta 1000 -> este de tip sejur
              -> s-a ajuns la destinatie cu autocar
              -> firma care s-a ocupat de transport este: Matrix Transport
-----------------------------------------------------------------------------
```

14. Pachet - flux de acțiuni integrate

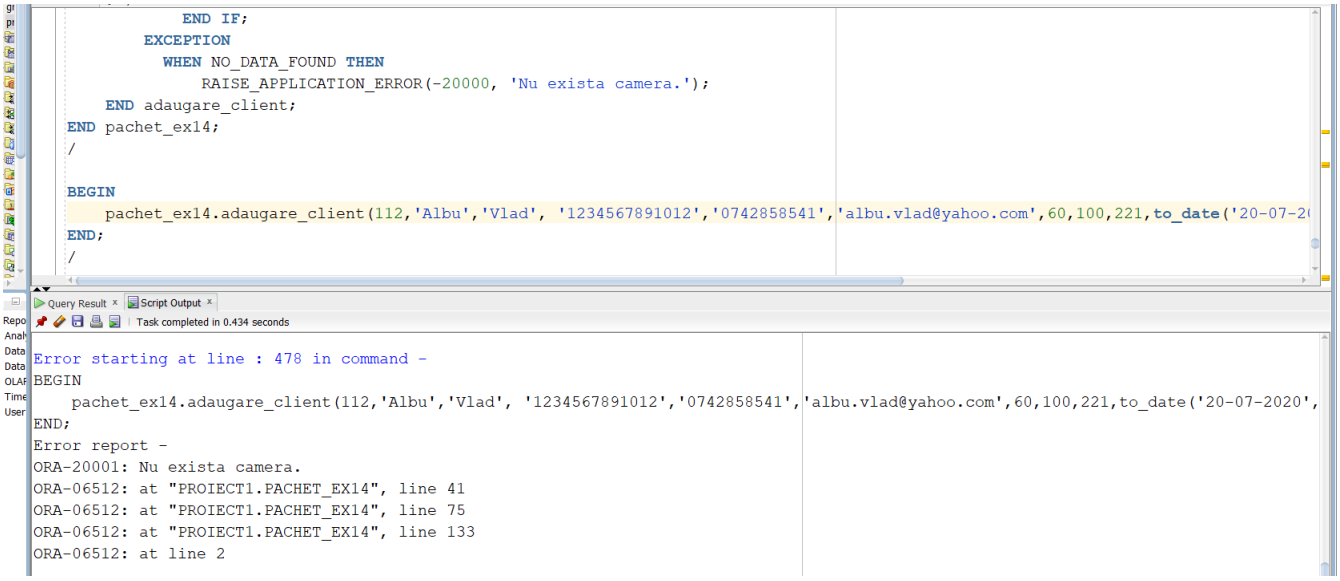
Definiți un pachet care să permită adăugarea unui client nou care face o rezervare nouă și se vor afișa detaliile rezervării(tip,hotel,cameră,agent,facilități).Pachetul va conține o procedură (are ca parametrii detaliile despre client și rezervare) care va permite adăugarea unui client și adăugarea unei noi rezervări,rezervare care se face prin intermediul altei proceduri(are ca parametrii detalii despre rezervare). Folosiți funcții pentru a prelua detaliile despre rezervare(tipul,hotelul,agentul,prețul).

```
CREATE OR REPLACE PACKAGE pachet_ex14 AS
    FUNCTION get_tip_oferta (cod_oferta obtine.id_oferta%TYPE) RETURN VARCHAR2;
    FUNCTION get_pret_oferta (cod_oferta obtine.id_oferta%TYPE) RETURN NUMBER;
    FUNCTION get_nume_hotel (cod_camera obtine.id_camera%TYPE) RETURN VARCHAR2;
    FUNCTION get_nume_agent (cod_agent obtine.id_angajat%TYPE) RETURN VARCHAR2;
    PROCEDURE adaugare_rezervare
         (cod_client obtine.id_client%TYPE,
          cod_angajat obtine.id_angajat%TYPE,
          cod_oferta obtine.id_oferta%TYPE,
          cod_camera obtine.id_camera%TYPE,
          data_venirii obtine.data_start%TYPE,
          data_plecarii obtine.data_end%TYPE);
    PROCEDURE adaugare_client
         (cod_client client.id_client%TYPE,
          nume_client client.nume%TYPE,
          prenume_client client.prenume%TYPE,
          CNP_client client.CNP%TYPE,
          nr_telefon client.telefon%TYPE,
          email client.adresa_email%TYPE,
          cod_angajat obtine.id_angajat%TYPE,
          cod_oferta obtine.id_oferta%TYPE,
          cod_camera obtine.id_camera%TYPE,
          data_venirii obtine.data_start%TYPE,
          data_plecarii obtine.data_end%TYPE);
END pachet_ex14;
/

CREATE OR REPLACE PACKAGE BODY pachet_ex14 AS
    FUNCTION get_tip_oferta (cod_oferta obtine.id_oferta%TYPE)
    RETURN VARCHAR2 IS
       tip_oft oferta.tip_oferta%TYPE;
       BEGIN
         --retinem tipul ofertei
         SELECT tip_oferta
         INTO tip_oft
         FROM oferta
         WHERE id_oferta = cod_oferta;
         RETURN tip_oft;
       EXCEPTION
          WHEN NO_DATA_FOUND THEN
              RAISE_APPLICATION_ERROR(-20000, 'Nu exista oferta.');
    END get_tip_oferta;

    FUNCTION get_pret_oferta (cod_oferta obtine.id_oferta%TYPE)
    RETURN NUMBER IS
       pret_oft oferta.pret%TYPE;
       BEGIN
          --retinem pretul ofertei
          SELECT pret
          INTO pret_oft
          FROM oferta
          WHERE id_oferta = cod_oferta;
          RETURN pret_oft;
       EXCEPTION
          WHEN NO_DATA_FOUND THEN
              RAISE_APPLICATION_ERROR(-20000, 'Nu exista oferta.');
```

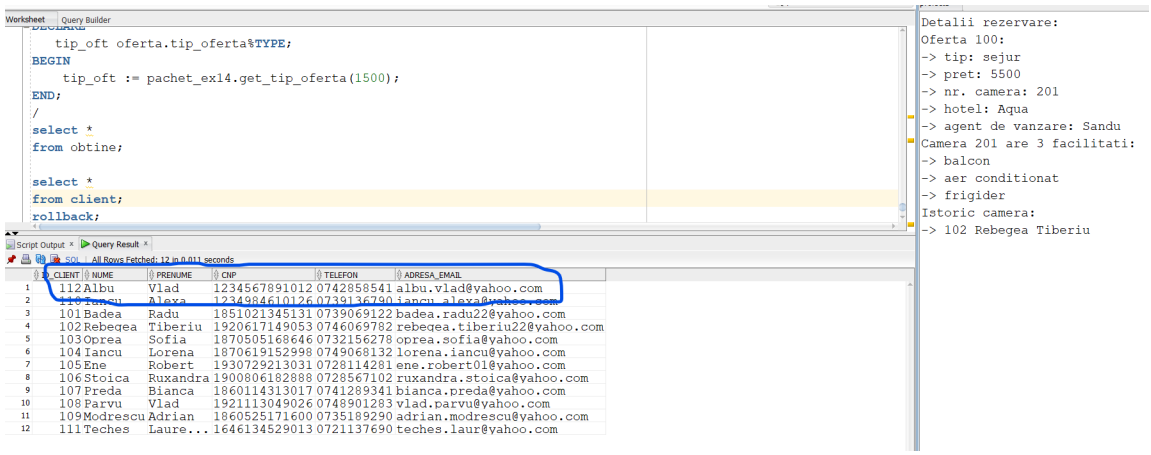
```
END get_pret_oferta;

FUNCTION get_nume_hotel (cod_camera obtine.id_camera%TYPE)
RETURN VARCHAR2 IS
    hotel_nume hotel.nume_hotel%TYPE;
    BEGIN
        --retinem hotelul unde s-a facut rezervarea
        SELECT nume_hotel
        INTO hotel_nume
        FROM hotel h, camera c
        WHERE id_camera = cod_camera AND c.id_hotel = h.id_hotel;
        RETURN hotel_nume;
    EXCEPTION
      WHEN NO_DATA_FOUND THEN
          RAISE_APPLICATION_ERROR(-20001, 'Nu exista camera.');
END get_nume_hotel;

FUNCTION get_nume_agent (cod_agent obtine.id_angajat%TYPE)
RETURN VARCHAR2 IS
      nume_agent angajat.nume%TYPE;
      BEGIN
          --retinem numele agentului
          SELECT nume
          INTO nume_agent
          FROM angajat
          WHERE id_angajat = cod_agent;
          RETURN nume_agent;
      EXCEPTION
        WHEN NO_DATA_FOUND THEN
          RAISE_APPLICATION_ERROR(-20003, 'Nu exista agentul de vanzare.');
END get_nume_agent;

PROCEDURE adaugare_rezervare
     (cod_client obtine.id_client%TYPE,
      cod_angajat obtine.id_angajat%TYPE,
      cod_oferta obtine.id_oferta%TYPE,
      cod_camera obtine.id_camera%TYPE,
      data_venirii obtine.data_start%TYPE,
      data_plecarii obtine.data_end%TYPE) IS
      tip_oft oferta.tip_oferta%TYPE;
      pret_oft oferta.pret%TYPE;
      nume_agent angajat.nume%TYPE;
      hotel_nume hotel.nume_hotel%TYPE;
      nr_fac NUMBER(5);
      TYPE tablou_indexat IS TABLE OF facilitati.denumire%TYPE INDEX BY
BINARY_INTEGER;
      fac_camera tablou_indexat;
      BEGIN
        tip_oft := get_tip_oferta(cod_oferta);
        pret_oft := get_pret_oferta(cod_oferta);
        hotel_nume := get_nume_hotel(cod_camera);
        nume_agent := get_nume_agent(cod_angajat);
        --inseram in tabel rezervarea
        INSERT INTO obtine
        VALUES(cod_client, cod_angajat, cod_oferta,cod_camera, data_venirii
, data_plecarii);
```

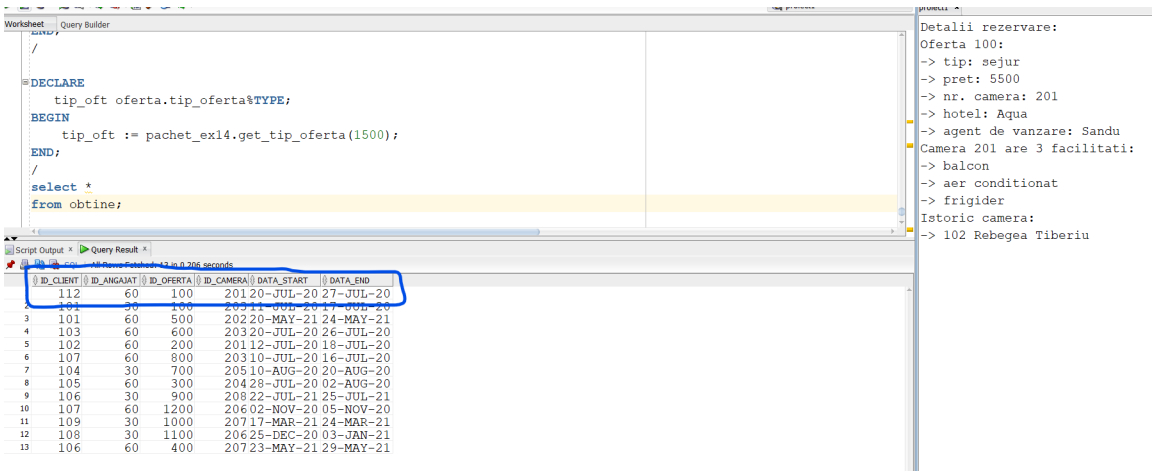
```
        --ne ocupam de afisarea detaliilor despre rezervare
        DBMS_OUTPUT.PUT_LINE('Detalii rezervare:');
        DBMS_OUTPUT.PUT_LINE('Oferta ' || cod_oferta || ':');
        DBMS_OUTPUT.PUT_LINE('-> tip: ' || tip_oft);
        DBMS_OUTPUT.PUT_LINE('-> pret: ' || pret_oft);
        DBMS_OUTPUT.PUT_LINE('-> nr. camera: ' || cod_camera);
        DBMS_OUTPUT.PUT_LINE('-> hotel: ' || hotel_nume);
        DBMS_OUTPUT.PUT_LINE('-> agent de vanzare: ' || nume_agent);
        --retinem facilitatile camereo
        SELECT denumire
        BULK COLLECT INTO fac_camera
        FROM facilitati f, contine_fac cf
        WHERE f.id_facilitate = cf.id_facilitate
              AND id_camera = cod_camera;
        nr_fac := fac_camera.COUNT;
        IF nr_fac = 0 THEN --in cazul in care camera nu are facilitati
            DBMS_OUTPUT.PUT_LINE('Camera ' || cod_camera || ' nu are
facilitati.');
        ELSIF nr_fac = 1 THEN
            DBMS_OUTPUT.PUT_LINE('Camera ' || cod_camera || ' are o singura
 facilitate:');
        ELSE
            DBMS_OUTPUT.PUT_LINE('Camera ' || cod_camera || ' are ' ||
nr_fac || ' facilitati:');
        END IF;
        --afisam facilitatile camerei
        FOR i in fac_camera.FIRST..fac_camera.last LOOP
             DBMS_OUTPUT.PUT_LINE('-> ' || fac_camera(i));
        END LOOP;
 END adaugare_rezervare;
 PROCEDURE adaugare_client
      (cod_client client.id_client%TYPE,
       nume_client client.nume%TYPE,
       prenume_client client.prenume%TYPE,
       CNP_client client.CNP%TYPE,
       nr_telefon client.telefon%TYPE,
       email client.adresa_email%TYPE,
       cod_angajat obtine.id_angajat%TYPE,
       cod_oferta obtine.id_oferta%TYPE,
       cod_camera obtine.id_camera%TYPE,
       data_venirii obtine.data_start%TYPE,
       data_plecarii obtine.data_end%TYPE) IS
       TYPE tablou_imbricat IS TABLE OF client%ROWTYPE;
       istoric_camera tablou_imbricat;
       nr_clienti NUMBER(5); -- nr de clienti rezervati pana in prezent in
camera respectiva
       BEGIN
        --retinem intai istoricul camerei rezervate
        --pentru a evita sa apara si numele clientului actual
        SELECT c.id_client,nume, prenume,CNP,telefon,adresa_email
        BULK COLLECT INTO istoric_camera
        FROM obtine ob, client c
        WHERE c.id_client = ob.id_client
              AND id_camera = cod_camera
              AND data_start < data_venirii;
        --inseram datele clientului in tabel
```

```
            INSERT INTO client
            VALUES(cod_client,nume_client,prenume_client,CNP_client,nr_telefon,
    email);
            --facem rezervarea clientului
            adaugare_rezervare(cod_client,cod_angajat,cod_oferta,cod_camera,
    data_venirii,data_plecarii);
            --ne ocupam sa afisam istoricul camerei rezervate
            nr_clienti := istoric_camera.COUNT;
            IF nr_clienti = 0 THEN
               DBMS_OUTPUT.PUT_LINE('Camera nu a mai fost rezervata in trecut')
    ;
            ELSE
               DBMS_OUTPUT.PUT_LINE('Istoric camera:');
               FOR i in istoric_camera.FIRST..istoric_camera.last LOOP
                 DBMS_OUTPUT.PUT_LINE('-> ' || istoric_camera(i).id_client || '
     '
                          || istoric_camera(i).nume || ' ' || istoric_camera(i).
    prenume);
               END LOOP;
            END IF;
        EXCEPTION
          WHEN NO_DATA_FOUND THEN
             RAISE_APPLICATION_ERROR(-20000, 'Nu exista camera.');
     END adaugare_client;
END pachet_ex14;
/

BEGIN
     pachet_ex14.adaugare_client(112,'Albu','Vlad', '1234567891012','0742858541'
    ,'albu.vlad@yahoo.com',60,100,221,to_date('20-07-2020','dd-mm-yyyy'),to_date
    ('27-07-2020','dd-mm-yyyy'));
END;
/
```

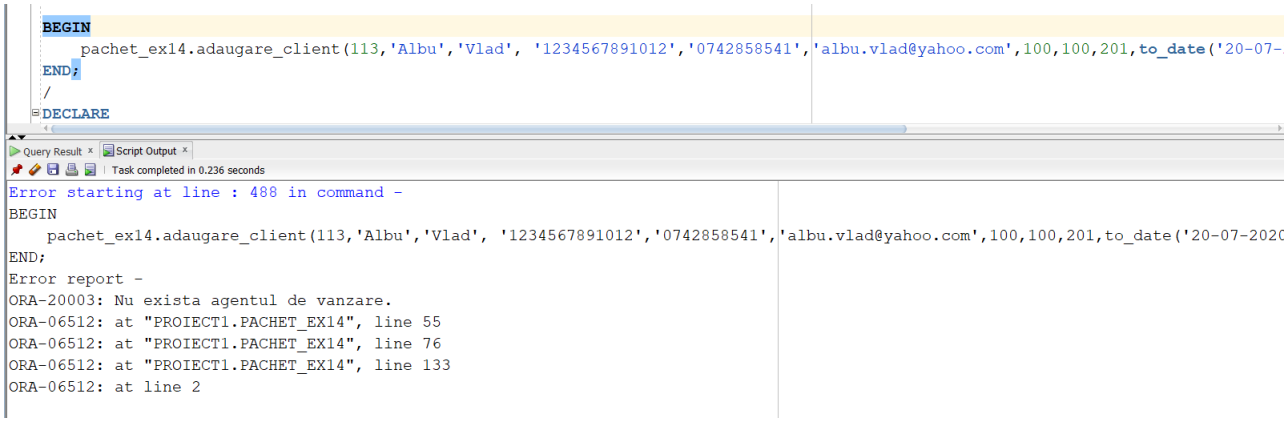


```
BEGIN
    pachet_ex14.adaugare_client(112,'Albu','Vlad', '1234567891012','0742858541'
   ,'albu.vlad@yahoo.com',60,100,201,to_date('20-07-2020','dd-mm-yyyy'),to_date
   ('27-07-2020','dd-mm-yyyy'));
END;
/
```

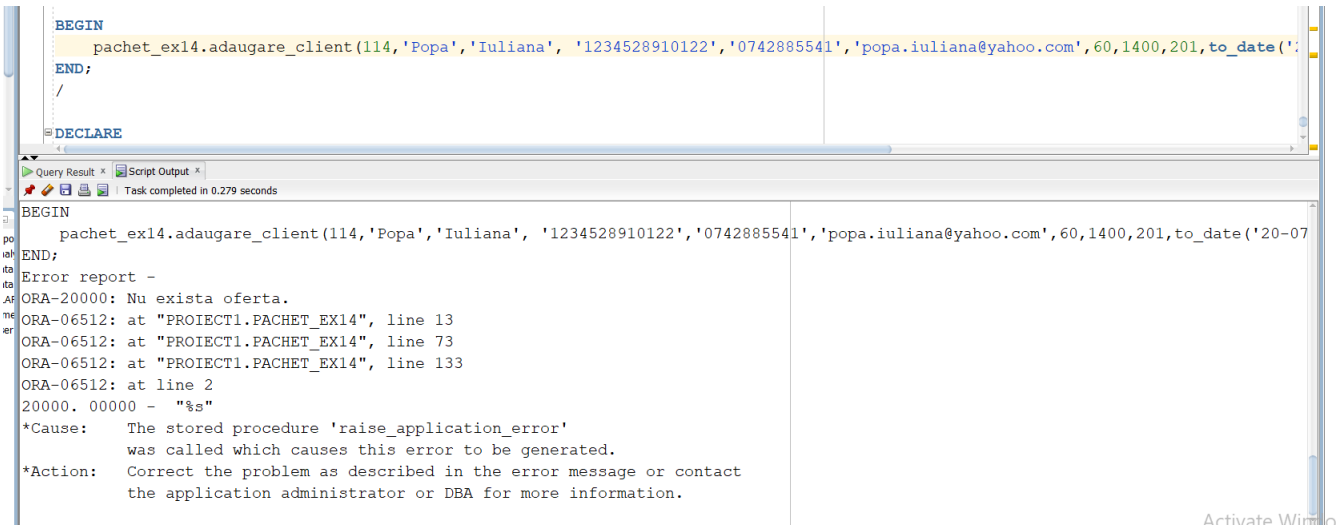




```
BEGIN
    pachet_ex14.adaugare_client(113,'Albu','Vlad', '1234567891012','0742858541'
   ,'albu.vlad@yahoo.com',100,100,201,to_date('20-07-2020','dd-mm-yyyy'),
   to_date('27-07-2020','dd-mm-yyyy'));
END;
/
```




```
BEGIN
    pachet_ex14.adaugare_client(114,'Popa','Iuliana', '1234528910122','
   0742885541','popa.iuliana@yahoo.com',60,1400,201,to_date('20-07-2020','dd-mm
   -yyyy'),to_date('27-07-2020','dd-mm-yyyy'));
END;
/
```




```
DECLARE
   tip_oft oferta.tip_oferta%TYPE;
BEGIN
    tip_oft := pachet_ex14.get_tip_oferta(1500);
END;
/
```

```
DECLARE
   tip_oft oferta.tip_oferta%TYPE;
BEGIN
    tip_oft := pachet_ex14.get_tip_oferta(1500);
END;
/
select *
from obtine;
```

Query Result | Script Output

Task completed in 0.048 seconds

```
BEGIN
   tip_oft := pachet_ex14.get_tip_oferta(1500);
END;
Error report -
ORA-20000: Nu exista oferta.
ORA-06512: at "PROIECT1.PACHET_EX14", line 13
ORA-06512: at line 4
20000. 00000 -  "%s"
*Cause:    The stored procedure 'raise_application_error'
           was called which causes this error to be generated.
*Action:   Correct the problem as described in the error message or contact
           the application administrator or DBA for more information.
```