

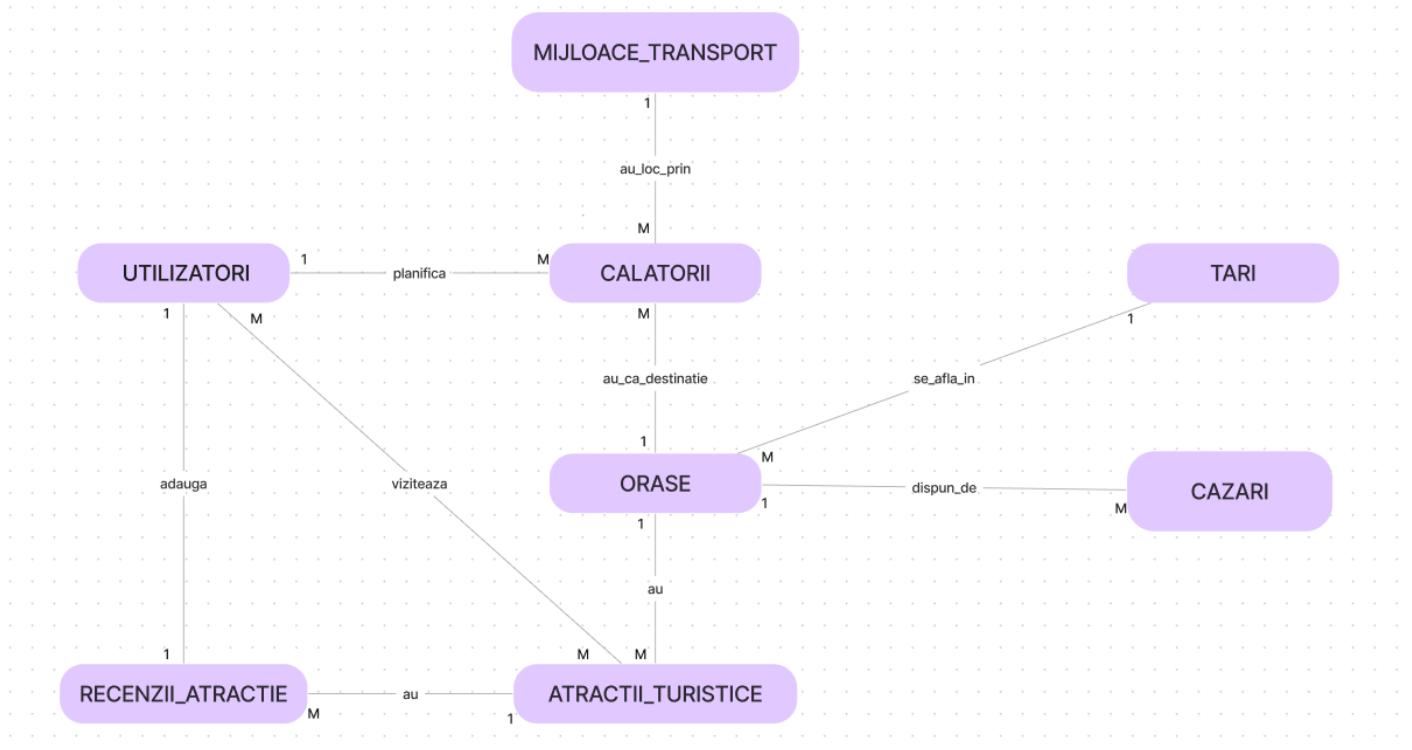
Gestionarea bazei de date a unei aplicatii mobile de Travel

1. Descriere si utilitate

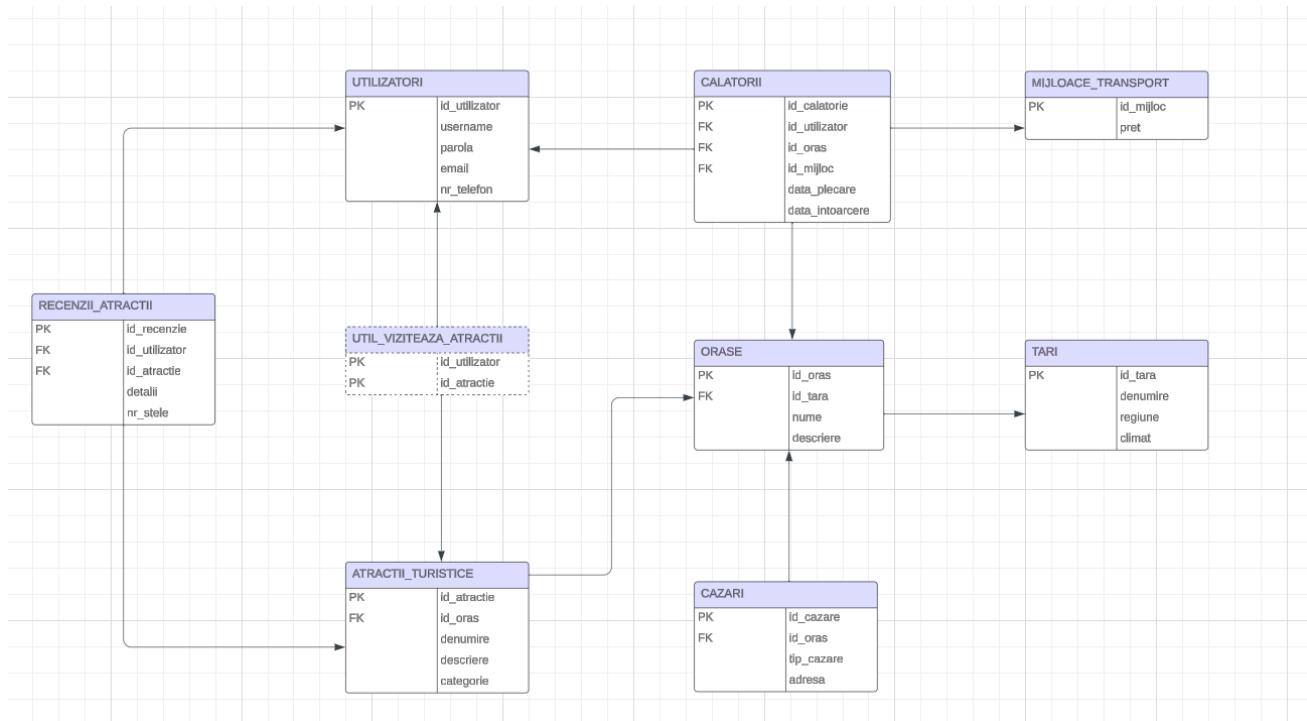
Scopul proiectului este gestionarea bazei de date a unei aplicatii pe mobil care inregistreaza calatorii utilizatorului, stocand date precum mijlocul de transport folosit, destinatia, data plecarii si cea a intoarcerii. In acelasi timp, utilizatorul se poate inspira din atractiile turistice care se afla in principalele orase ale lumii. Oferim informatii despre climatul tarii careia apartine destinatia, descrieri la fiecare oras si atractie, posibilitati de cazare in unele orase, pe categorii.

Aplicatia ofera posibilitatea de a citi recenzii ale diferitelor atractii, iar utilizatorii pot scrie si ei recenzii locatiilor vizitate de acestia alaturi de un rating de maxim 5 stele.

2. Diagrama Entitate-Relatie



3. Diagrama conceptuala



4. Definirea tabelelor

UTILIZATORI

```

create table utilizatori (
    id_utilizator int not null Primary Key,
    username varchar2(20),
    parola varchar2(20),
    email varchar2(50),
    nr_telefon varchar2(15)
);
    
```

--4. Definirea tabelelor

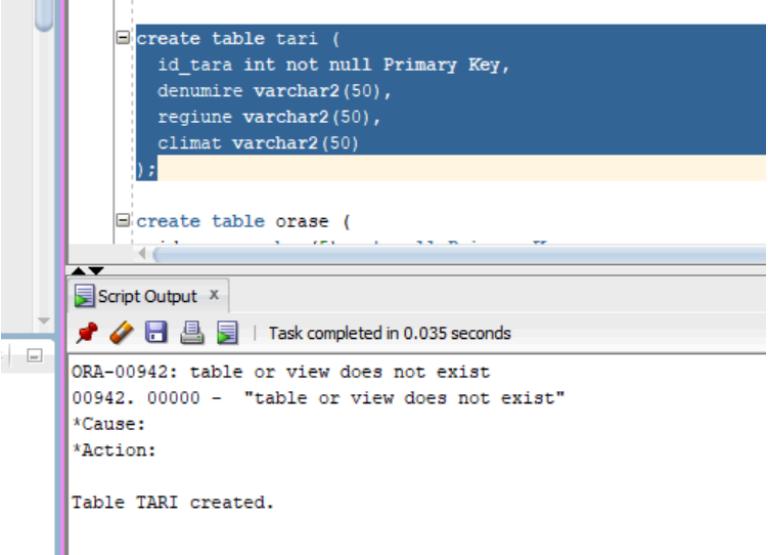
```

create table utilizatori (
    id_utilizator int not null Primary Key,
    username varchar2(20),
    parola varchar2(20),
    email varchar2(50),
    nr_telefon varchar2(15)
);

Table UTILIZATORI created.
    
```

TARI

```
create table tari (
    id_tara int not null Primary Key,
    denumire varchar2(50),
    regiune varchar2(50),
    climat varchar2(50)
);
```



The screenshot shows the Oracle SQL Developer interface. In the main editor area, the SQL code for creating the TARI table is displayed. Below the editor, the 'Script Output' tab is selected, showing the execution results. The output indicates that the table was created successfully.

```
create table tari (
    id_tara int not null Primary Key,
    denumire varchar2(50),
    regiune varchar2(50),
    climat varchar2(50)
);

create table orase (
    id_oras number(5) not null Primary Key,
    id_tara int,
    nume varchar2(50),
    descriere varchar2(255),
    foreign key (id_tara) references tari(id_tara)
);
```

Script Output | Task completed in 0.035 seconds

ORA-00942: table or view does not exist
00942. 00000 - "table or view does not exist"
*Cause:
*Action:

Table TARI created.

ORASE

```
create table orase (
    id_oras number(5) not null Primary Key,
    id_tara int,
    nume varchar2(50),
    descriere varchar2(255),
    foreign key (id_tara) references tari(id_tara)
);
```

```
create table orase (
    id_oras number(5) not null Primary Key,
    id_tara int,
    nume varchar2(50),
    descriere varchar2(255),
    foreign key (id_tara) references tari(id_tara)
);

```

Script Output X | Task completed in 0.035 seconds

Table ORASE created.

CALATORII

```
create table calatorii (
    id_calatorie int not null Primary Key,
    id_utilizator int,
    id_oras number(5),
    id_mijloc number(6),
    data_plecare date,
    data_intoarcere date,
    foreign key (id_utilizator) references utilizatori(id_utilizator),
    foreign key (id_oras) references orase(id_oras),
    foreign key (id_mijloc) references mijloace_transport(id_mijloc)
);
```

```
create table calatorii (
    id_calatorie int not null Primary Key,
    id_utilizator int,
    id_oras number(5),
    id_mijloc number(6),
    data_plecare date,
    data_intoarcere date,
    foreign key (id_utilizator) references utilizatori(id_utilizator),
    foreign key (id_oras) references orase(id_oras),
    foreign key (id_mijloc) references mijloace_transport(id_mijloc)
);

```

Script Output X | Task completed in 0.035 seconds

Table CALATORII created.

MIJLOACE_TRANSPORT

```
create table mijloace_transport (
    id_mijloc number(6) not null Primary Key,
    denumire varchar2(50),
    tip varchar2(50),
    firma varchar2(50),
    pret number(5,2)
);
```

```
alter table mijloace_transport
add categorie varchar2(50);
```

The screenshot shows the Oracle SQL Developer interface. In the top window, there is a code editor containing the following SQL script:

```
create table mijloace_transport (
    id_mijloc number(6) not null Primary Key,
    denumire varchar2(50),
    tip varchar2(50),
    firma varchar2(50),
    pret number(5,2)
);

alter table mijloace_transport
add categorie varchar2(50);
```

In the bottom window, titled "Script Output", the results of the execution are displayed:

```
Table CALATORII dropped.

Table MIJLOACE_TRANSPORT created.
```

The status bar at the bottom of the output window indicates "Task completed in 0.035 seconds".

ATRACTII

```
create table atractii (
    id_atractie int not null Primary Key,
    id_oras number(5),
    denumire varchar(50),
    descriere varchar(255),
    categorie varchar(50),
    foreign key (id_oras) references orase(id_oras)
);
```

```
create table atractii (
    id_atractie int not null Primary Key,
    id_oras number(5),
    denumire varchar(50),
    descriere varchar(255),
    categorie varchar(50),
    foreign key (id_oras) references orase(id_oras)
);

create table recenziile_atractie (
    id_recenzie int not null Primary Key,
    id_user int,
```

Script Output X | Task completed in 0.035 seconds

Table ATRACTII created.

RECENZII_ATRACTIE

```
create table recenziile_atractie (
    id_recenzie int not null Primary Key,
    id_user int,
    id_atractie int,
    detalii varchar2(255),
    nr_stele number(1) not null check (nr_stele >= 1 and nr_stele <= 5),
    foreign key (id_user) references utilizatori(id_utilizator),
    foreign key (id_atractie) references atractii(id_atractie)
);
```

```
create table recenziile_atractie (
    id_recenzie int not null Primary Key,
    id_user int,
    id_atractie int,
    detalii varchar2(255),
    nr_stele number(1) not null check (nr_stele >= 1 and nr_stele <= 5),
    foreign key (id_user) references utilizatori(id_utilizator),
    foreign key (id_atractie) references atractii(id_atractie)
);

create table cazari (
    id_cazare int not null Primary Key,
```

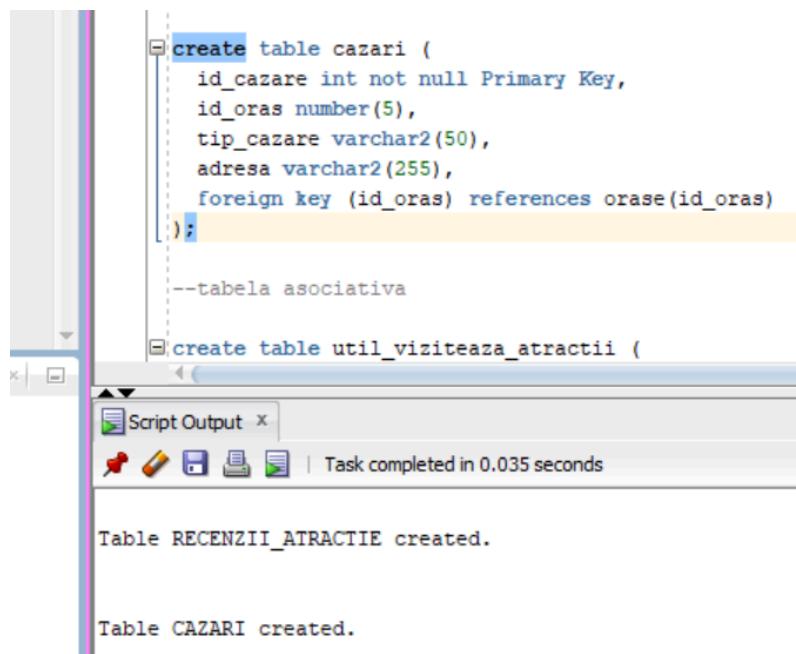
Script Output X | Task completed in 0.035 seconds

Table ATRACTII created.

Table RECENZII_ATRACTIE created.

CAZARI

```
create table cazari (
    id_cazare int not null Primary Key,
    id_oras number(5),
    tip_cazare varchar2(50),
    adresa varchar2(255),
    foreign key (id_oras) references orase(id_oras)
);
```



The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor with the following SQL script:

```
create table cazari (
    id_cazare int not null Primary Key,
    id_oras number(5),
    tip_cazare varchar2(50),
    adresa varchar2(255),
    foreign key (id_oras) references orase(id_oras)
);

--tabela asociativa

create table util_viziteaza_atractii (
```

In the bottom-right pane, there is a "Script Output" window with the following log:

```
Table RECENZII_ATRACTIE created.

Table CAZARI created.
```

UTIL_VIZITEAZA_ATRACTII

```
create table util_viziteaza_atractii (
    id_vizita int not null Primary Key,
    id_utilizator int,
    id_atractie int,
    data_vizita date,
    foreign key (id_utilizator) references utilizatori(id_utilizator),
    foreign key (id_atractie) references atractii(id_atractie)
);
```

```
--tabela asociativa

create table util_viziteaza_atractii (
    id_vizita int not null Primary Key,
    id_utilizator int,
    id_atractie int,
    data_vizita date,
    foreign key (id_utilizator) references utilizatori(id_utilizator),
    foreign key (id_atractie) references atractii(id_atractie)
);

Table UTIL_VIZITEAZA_ATRACTII created.

1 row inserted.
```

5. Adaugarea de inregistrari in tabele

UTILIZATORI

```
insert into utilizatori (id_utilizator, username, parola, email, nr_telefon)
values (1, 'anapopescu123', 'parolaAna', 'ana.popescu@example.com', '0712345678');
```

```
insert into utilizatori (id_utilizator, username, parola, email, nr_telefon)
values (2, 'mihaiionescu456', 'parolaMihai', 'mihai.ionescu@example.com', '0723456789');
```

```
insert into utilizatori (id_utilizator, username, parola, email, nr_telefon)
values (3, 'elenaradu789', 'parolaElena', 'elena.radu@example.com', '0734567890');
```

```
insert into utilizatori (id_utilizator, username, parola, email, nr_telefon)
values (4, 'alexandradumitru101', 'parolaAlex', 'alexandra.dumitru@example.com',
'0745678901');
```

```
insert into utilizatori (id_utilizator, username, parola, email, nr_telefon)
values (5, 'andreigheorghe202', 'parolaAndrei', 'andrei.gheorghe@example.com',
'0756789012');
```

```
insert into utilizatori (id_utilizator, username, parola, email, nr_telefon)
values (6, 'cristinastoica303', 'parolaCristina', 'cristina.stoica@example.com', '0767890123');
```

```
insert into utilizatori (id_utilizator, username, parola, email, nr_telefon)
values (7, 'ionconstantin404', 'parolalon', 'ion.constantin@example.com', '0778901234');
```

```
insert into utilizatori (id_utilizator, username, parola, email, nr_telefon)
```

```
values (8, 'ralucaflorescu505', 'parolaRaluca', 'raluca.florescu@example.com',  
'0789012345');
```

```
insert into utilizatori (id_utilizator, username, parola, email, nr_telefon)  
values (9, 'gabrielmoldovan606', 'parolaGabriel', 'gabriel.moldovan@example.com',  
'0790123456');
```

```
insert into utilizatori (id_utilizator, username, parola, email, nr_telefon)  
values (10, 'dianamarin707', 'parolaDiana', 'diana.marin@example.com', '0701234507');
```

```
insert into utilizatori  
values (11, 'elenaradu789', 'parolaparola', 'elena1989@yahoo.com', '0725472817');
```

The screenshot shows the MySQL Workbench interface with two tabs: 'Script Output' and 'Query Result'. The 'Query Result' tab is active and displays the following SQL code and its execution results:

```
values (8, 'ralucaflorescu505', 'parolaRaluca', 'raluca.florescu@example.com', '0789012345');

insert into utilizatori (id_utilizator, username, parola, email, nr_telefon)
values (9, 'gabrielmoldovan606', 'parolaGabriel', 'gabriel.moldovan@example.com', '0790123456');

insert into utilizatori (id_utilizator, username, parola, email, nr_telefon)
values (10, 'dianamarin707', 'parolaDiana', 'diana.marin@example.com', '0701234507');

insert into utilizatori
values (11, 'elenaradu789', 'parolaparola', 'elena1989@yahoo.com', '0725472817');

select * from utilizatori;
```

Below the code, the results are shown in a table:

ID_UTILIZATOR	USERNAME	PAROLA	EMAIL	NR_TELEFON
1	1 anapopescu123	parolaAna	ana.popescu@example.com	0712345678
2	2 mihaiionescu456	parolaMihai	mihai.ionescu@example.com	0723456789
3	3 elenaradu789	parolaElena	elena.radu@example.com	0734567890
4	4 alexandradumitru101	parolaAlex	alexandra.dumitru@example.com	0745678901
5	5 andreigheorghe202	parolaAndrei	andrei.gheorghe@example.com	0756789012
6	6 cristinastoica303	parolaCristina	cristina.stoica@example.com	0767890123
7	7 ionconstantin404	parolaIon	ion.constantin@example.com	0778901234
8	8 ralucaflorescu505	parolaRaluca	raluca.florescu@example.com	0789012345
9	9 gabrielmoldovan606	parolaGabriel	gabriel.moldovan@example.com	0790123456
10	10 dianamarin707	parolaDiana	diana.marin@example.com	0701234507
11	11 elenaradu789	parolaparola	elena1989@yahoo.com	0725472817

TARI

```
insert into tari (id_tara, denumire, regiune, climat)  
values (11, 'Romania', 'Europa', 'temperat');
```

```
insert into tari (id_tara, denumire, regiune, climat)  
values (21, 'Franta', 'Europa', 'temperat');
```

```
insert into tari (id_tara, denumire, regiune, climat)  
values (31, 'Olanda', 'Europa', 'maritim temperat');
```

```
insert into tari (id_tara, denumire, regiune, climat)
values (41, 'Iordania', 'Asia', 'subtropical');
```

```
insert into tari (id_tara, denumire, regiune, climat)
values (51, 'Brazilia', 'America de Sud', 'tropical');
```

```
insert into tari (id_tara, denumire, regiune, climat)
values (61, 'Marea Britanie', 'Europa', 'oceanic');
```

```
insert into tari (id_tara, denumire, regiune, climat)
values (71, 'Australia', 'Oceania', 'temperat');
```

```
insert into tari (id_tara, denumire, regiune, climat)
values (81, 'Italia', 'Europa', 'mediteranean');
```

```
insert into tari (id_tara, denumire, regiune, climat)
values (91, 'India', 'Asia', 'tropical');
```

```
insert into tari (id_tara, denumire, regiune, climat)
values (101, 'Finlanda', 'Europa', 'subpolar');
```

```
insert into tari (id_tara, denumire, regiune, climat)
values (81, 'Italia', 'Europa', 'mediteranean');

insert into tari (id_tara, denumire, regiune, climat)
values (91, 'India', 'Asia', 'tropical');

insert into tari (id_tara, denumire, regiune, climat)
values (101, 'Finlanda', 'Europa', 'subpolar');

select * from tari;
```

Script Output x Query Result x

SQL | All Rows Fetched: 10 in 0.004 seconds

ID_TARA	DENUMIRE	REGIUNE	CLIMAT
1	11 Romania	Europa	temperat
2	21 Franta	Europa	temperat
3	31 Olanda	Europa	maritim temperat
4	41 Iordania	Asia	subtropical
5	51 Brazilia	America de Sud	tropical
6	61 Marea Britanie	Europa	oceanic
7	71 Australia	Oceania	temperat
8	81 Italia	Europa	mediteranean
9	91 India	Asia	tropical
10	101 Finlanda	Europa	subpolar

ORASE

```
insert into orase (id_oras, id_tara, nume, descriere)
values (100, 11, 'Bucuresti', 'Capitala Romaniei');
```

```
insert into orase (id_oras, id_tara, nume, descriere)
values (200, 21, 'Paris', 'Capitala Frantei');
```

```
insert into orase (id_oras, id_tara, nume, descriere)
values (300, 31, 'Amsterdam', 'Capitala Olandei');
```

```
insert into orase (id_oras, id_tara, nume, descriere)
values (400, 41, 'Amman', 'Capitala Iordaniei');
```

```
insert into orase (id_oras, id_tara, nume, descriere)
values (500, 51, 'Brasilia', 'Capitala Braziliei');
```

```
insert into orase (id_oras, id_tara, nume, descriere)
values (600, 61, 'Londra', 'Capitala Marii Britanii');
```

```
insert into orase (id_oras, id_tara, nume, descriere)
values (800, 81, 'Roma', 'Capitala Italiei');
```

```
insert into orase (id_oras, id_tara, nume, descriere)
values (900, 91, 'Mumbai', 'Cel mai populat oras din India');
```

```
insert into orase (id_oras, id_tara, nume, descriere)
values (1000, 101, 'Helsinki', 'Capitala Finlandei');
```

```
insert into orase (id_oras, id_tara, nume, descriere)
values (401, 41, 'Petrica', null);
```

```
insert into orase (id_oras, id_tara, nume, descriere)
values (601, 61, 'Oxford', null);
```

```
insert into orase (id_oras, id_tara, nume, descriere)
values (602, 61, 'Birmingham', 'Cea mai mare celebrare a Sf Patrick din Anglia');
```

```
insert into orase (id_oras, id_tara, nume, descriere)
values (101, 11, 'Ploiesti', null);
```

```
insert into orase (id_oras, id_tara, nume, descriere)
values (102, 11, 'Brasov', null);
```

```
insert into orase (id_oras, id_tara, nume, descriere)
values (103, 11, 'Constanta', null);
```

```

insert into orase (id_oras, id_tara, nume, descriere)
values (101, 11, 'Ploiesti', null);

insert into orase (id_oras, id_tara, nume, descriere)
values (102, 11, 'Brasov', null);

insert into orase (id_oras, id_tara, nume, descriere)
values (103, 11, 'Constanta', null);

select * from orase;

insert into atractii (id_atractie, id_oras, denumire, descriere, categorie)

```

Script Output | Query Result | All Rows Fetched: 15 in 0.003 seconds

ID_ORAS	ID_TARA	NUME	DESCRIERE
1	100	11 Bucuresti	Capitala Romaniei
2	200	21 Paris	Capitala Frantei
3	300	31 Amsterdam	Capitala Olandei
4	400	41 Amman	Capitala Iordaniei
5	500	51 Brasilia	Capitala Braziliei
6	600	61 Londra	Capitala Marii Britanii
7	800	81 Roma	Capitala Italiei
8	900	91 Mumbai	Cel mai populat oras din India
9	1000	101 Helsinki	Capitala Finlandei
10	401	41 Petra	(null)
11	601	61 Oxford	(null)

ATRACTII

```
insert into atractii (id_atractie, id_oras, denumire, descriere, categorie)
values (1001, 100, 'Palatul Parlamentului', 'Cel mai mare palat administrativ pentru utilizare civila', 'Arhitectura');
```

```
insert into atractii (id_atractie, id_oras, denumire, descriere, categorie)
values (1002, 200, 'Turnul Eiffel', 'Cunoscutul turn de fier din inima Parisului', 'Monument');
```

```
insert into atractii (id_atractie, id_oras, denumire, descriere, categorie)
values (1003, 300, 'Muzeul Van Gogh', 'Gazda pentru cele mai multe lucrari ale artistului Van Gogh', 'Muzeu');
```

```
insert into atractii (id_atractie, id_oras, denumire, descriere, categorie)
values (1004, 400, 'Petra', 'Oras antic Petra din Iordania', 'Sit istoric');
```

```
insert into atractii (id_atractie, id_oras, denumire, descriere, categorie)
values (1005, 100, 'Muzeul Satului', null, 'Muzeu');
```

```
insert into atractii (id_atractie, id_oras, denumire, descriere, categorie)
values (1006, 600, 'Muzeul Britanic', 'Una dintre cele mai mari colectii de arta si artefacte din lume', 'Muzeu');
```

```
insert into atractii (id_atractie, id_oras, denumire, descriere, categorie)
values (1007, 800, 'Coloseum', 'Amfiteatru roman antic din centrul Romei', 'Arheologie');
```

```
insert into atractii (id_atractie, id_oras, denumire, descriere, categorie)
values (1008, 600, 'Palatul Buckingham', 'Resedinta principala din Londra a monarhilor britanici', 'Reper cultural');
```

```
insert into atractii (id_atractie, id_oras, denumire, descriere, categorie)
values (1009, 400, 'Cetatea Ammanului', 'Este considerata printre cele mai vechi locuri locuite in continuu din lume', 'Sit istoric');
```

```
insert into atractii (id_atractie, id_oras, denumire, descriere, categorie)
values (1010, 401, 'Al-Kazneh', 'Faimosul monument al Petra sculptat in stana', 'Monument');
```

```
insert into atractii (id_atractie, id_oras, denumire, descriere, categorie)
values (1011, 601, 'Universitatea Oxford', 'Una dintre cele mai prestigioase universitati din lume', 'Educatie');
```

```
insert into atractii (id_atractie, id_oras, denumire, descriere, categorie)
values (1012, 602, 'Birmingham St. Patricks Day Parade', 'Parada anuala in cinstea Sf. Patrick', 'Eveniment cultural');
```

```
insert into atractii (id_atractie, id_oras, denumire, descriere, categorie)
values (1013, 101, 'Muzeul Ceasului', 'Expozitie dedicata evolutiei timpului', 'Muzeu');
```

```
insert into atractii (id_atractie, id_oras, denumire, descriere, categorie)
values (1014, 102, 'Castelul Bran', 'Cunoscut si sub numele de Castelul lui Dracula', 'Cetate');
```

```
insert into atractii (id_atractie, id_oras, denumire, descriere, categorie)
values (1015, 103, 'Plaja Mamaia', 'Statiune de la malul marii in Constanta', 'Recreere');
```

The screenshot shows a SQL editor window with the following content:

```
insert into atractii (id_atractie, id_oras, denumire, descriere, categorie)
values (1012, 602, 'Birmingham St. Patricks Day Parade', 'Parada anuala in cinstea Sf. Patrick', 'Eveniment cultural')

insert into atractii (id_atractie, id_oras, denumire, descriere, categorie)
values (1013, 101, 'Muzeul Ceasului', 'Expozitie dedicata evolutiei timpului', 'Muzeu');

insert into atractii (id_atractie, id_oras, denumire, descriere, categorie)
values (1014, 102, 'Castelul Bran', 'Cunoscut si sub numele de Castelul lui Dracula', 'Cetate');

insert into atractii (id_atractie, id_oras, denumire, descriere, categorie)
values (1015, 103, 'Plaja Mamaia', 'Statiune de la malul marii in Constanta', 'Recreere');

select * from atractii;
```

Below the code, the "Script Output" tab shows the message: "All Rows Fetched: 15 in 0.002 seconds". The "Query Result" tab displays a table with the following data:

ID_ATRACTIE	ID_ORAS	DENUMIRE	DESCRIERE
1	1001	100 Palatul Parlamentului	Cel mai mare palat administrativ pentru utilizare civila
2	1002	200 Turnul Eiffel	Cunoscutul turn de fier din inima Parisului
3	1003	300 Muzeul Van Gogh	Gazda pentru cele mai multe lucrari ale artistului Van Gogh
4	1004	400 Petra	Oras antic Petra din Iordania
5	1005	100 Muzeul Satului	(null)
6	1006	600 Muzeul Britanic	Una dintre cele mai mari colectii de arta si artefacte din lume
7	1007	800 Coloseum	Amfiteatru roman antic din centrul Romei
8	1008	600 Palatul Buckingham	Resedinta principala din Londra a monarhilor britanici
9	1009	400 Cetatea Ammanului	Este considerata printre cele mai vechi locuri locuite in continut
10	1010	401 Al-Kazneh	Faimosul monument al Petra sculptat in stana

MIJLOACE_TRANSPORT

```
insert into mijloace_transport (id_mijloc, denumire, firma, pret, categorie)
values (100001, 'Tren CFR Express', 'CFR', 50.00, 'tren');
```

```
insert into mijloace_transport (id_mijloc, denumire, firma, pret, categorie)
values (100002, 'IR 1656', 'Regio', 30.00, 'tren');
```

```
insert into mijloace_transport (id_mijloc, denumire, firma, pret, categorie)
values (100003, 'AirJet A320', 'AirJet', 200.00, 'avion');
```

```
insert into mijloace_transport (id_mijloc, denumire, firma, pret, categorie)
values (100004, 'FlySky 737', 'FlySky', 180.00, 'avion');
```

```
insert into mijloace_transport (id_mijloc, denumire, firma, pret, categorie)
values (100005, 'Express Train 2000', 'CFR', 60.00, 'tren');
```

```
insert into mijloace_transport (id_mijloc, denumire, firma, pret, categorie)
values (100006, 'Regional Express', 'Regio', 35.00, 'tren');
```

```
insert into mijloace_transport (id_mijloc, denumire, firma, pret, categorie)
values (100007, 'Swift Transporter', 'CFR', 55.00, 'tren');
```

```
insert into mijloace_transport (id_mijloc, denumire, firma, pret, categorie)
values (100008, 'MetroExpress', 'Regio', 40.00, 'tren');
```

```
insert into mijloace_transport (id_mijloc, denumire, firma, pret, categorie)
values (100009, 'High-Speed Voyager', 'CFR', 75.00, 'tren');
```

```
insert into mijloace_transport (id_mijloc, denumire, firma, pret, categorie)
values (100010, 'SkyBlitz 747', 'SkyBlitz', 250.00, 'avion');
```

```
insert into mijloace_transport (id_mijloc, denumire, firma, pret, categorie)
values (100011, 'JetStream 3000', 'JetStream', 220.00, 'avion');
```

```
insert into mijloace_transport (id_mijloc, denumire, firma, pret, categorie)
values (100012, 'FlySky 999', 'FlySky', 300.00, 'avion');
```

```
insert into mijloace_transport (id_mijloc, denumire, firma, pret, categorie)
values (100013, 'FlySky 738', 'FlySky', 180.00, 'avion');
```

```
insert into mijloace_transport (id_mijloc, denumire, firma, pret, categorie)
values (100014, 'UltraJet X10', 'UltraJet', 280.00, 'avion');
```

```

values (100011, 'JetStream 3000', 'JetStream', 220.00, 'avion');

insert into mijloace_transport (id_mijloc, denumire, firma, pret, categorie)
values (100012, 'FlySky 999', 'FlySky', 300.00, 'avion');

insert into mijloace_transport (id_mijloc, denumire, firma, pret, categorie)
values (100013, 'FlySky 738', 'FlySky', 180.00, 'avion');

insert into mijloace_transport (id_mijloc, denumire, firma, pret, categorie)
values (100014, 'UltraJet X10', 'UltraJet', 280.00, 'avion');

select * from mijloace_transport;

```

The screenshot shows the Oracle SQL Developer interface. At the top, there are tabs for 'Script Output' and 'Query Result'. Below the tabs, it says 'SQL | All Rows Fetched: 14 in 0.002 seconds'. The main area displays a table with 14 rows of data.

	ID_MIJLOC	DENUMIRE	FIRMA	PRET	CATEGORIE
1	100001	Tren CFR Express	CFR	50	tren
2	100002	IR 1656	Regio	30	tren
3	100003	AirJet A320	AirJet	200	avion
4	100004	FlySky 737	FlySky	180	avion
5	100005	Express Train 2000	CFR	60	tren
6	100006	Regional Express	Regio	35	tren
7	100007	Swift Transporter	CFR	55	tren
8	100008	MetroExpress	Regio	40	tren
9	100009	High-Speed Voyager	CFR	75	tren
10	100010	SkyBlitz 747	SkyBlitz	250	avion
11	100011	JetStream 3000	JetStream	220	avion

CALATORII

```

insert into calatorii (id_calatorie, id_utilizator, id_oras, id_mijloc, data_plecare,
data_intoarcere)
values (61, 1, 101, 100001, to_date('2024-01-15', 'YYYY-MM-DD'), to_date('2024-01-20',
'YYYY-MM-DD'));

insert into calatorii (id_calatorie, id_utilizator, id_oras, id_mijloc, data_plecare,
data_intoarcere)
values (62, 2, 200, 100005, to_date('2023-02-10', 'YYYY-MM-DD'), to_date('2023-02-15',
'YYYY-MM-DD'));

insert into calatorii (id_calatorie, id_utilizator, id_oras, id_mijloc, data_plecare,
data_intoarcere)
values (63, 3, 300, 100009, to_date('2024-03-05', 'YYYY-MM-DD'), to_date('2024-03-10',
'YYYY-MM-DD'));

insert into calatorii (id_calatorie, id_utilizator, id_oras, id_mijloc, data_plecare,
data_intoarcere)
values (64, 4, 401, 100012, to_date('2024-04-20', 'YYYY-MM-DD'), to_date('2024-04-25',
'YYYY-MM-DD'));

```

```

insert into calatorii (id_calatorie, id_utilizator, id_oras, id_mijloc, data_plecare,
data_intoarcere)
values (65, 5, 500, 100014, to_date('2023-05-15', 'YYYY-MM-DD'), to_date('2023-05-20',
'YYYY-MM-DD'));

insert into calatorii (id_calatorie, id_utilizator, id_oras, id_mijloc, data_plecare,
data_intoarcere)
values (66, 1, 101, 100002, to_date('2024-06-10', 'YYYY-MM-DD'), to_date('2024-06-15',
'YYYY-MM-DD'));

insert into calatorii (id_calatorie, id_utilizator, id_oras, id_mijloc, data_plecare,
data_intoarcere)
values (67, 2, 200, 100010, to_date('2024-07-05', 'YYYY-MM-DD'), to_date('2024-07-10',
'YYYY-MM-DD'));

insert into calatorii (id_calatorie, id_utilizator, id_oras, id_mijloc, data_plecare,
data_intoarcere)
values (68, 1, 601, 100010, to_date('2022-01-15', 'YYYY-MM-DD'), to_date('2022-02-19',
'YYYY-MM-DD'));

```

The screenshot shows the Oracle SQL Developer interface. On the left, there is a code editor window containing the four `INSERT` statements for the `calatorii` table. On the right, there is a "Query Result" tab showing the output of the `SELECT * FROM calatorii;` query. The result is a table with 8 rows, each representing a trip record with columns: ID_CALATORIE, ID_UTILIZATOR, ID_ORAS, ID_MIJLOC, DATA_PLECARE, and DATA_INTOARCERE.

ID_CALATORIE	ID_UTILIZATOR	ID_ORAS	ID_MIJLOC	DATA_PLECARE	DATA_INTOARCERE
1	61	1	101	100001 15-JAN-24	20-JAN-24
2	64	4	401	100012 20-APR-24	25-APR-24
3	66	1	101	100002 10-JUN-24	15-JUN-24
4	67	2	200	100010 05-JUL-24	10-JUL-24
5	62	2	200	100005 10-FEB-23	15-FEB-23
6	63	3	300	100009 05-MAR-24	10-MAR-24
7	65	5	500	100014 15-MAY-23	20-MAY-23
8	68	1	601	100010 15-JAN-22	19-FEB-22

CAZARI

```

insert into cazari (id_cazare, id_oras, tip_cazare, adresa)
values (999, 101, 'Hotel', 'Strada Lautari, Nr. 38');

insert into cazari (id_cazare, id_oras, tip_cazare, adresa)
values (998, 200, 'Airbnb', 'Rue de Rivoli, Nr. 2');

insert into cazari (id_cazare, id_oras, tip_cazare, adresa)

```

```

values (997, 300, 'Hotel', 'Spadinalaan 97');

insert into cazari (id_cazare, id_oras, tip_cazare, adresa)
values (996, 800, 'Hotel', 'Via le Marco Polo 102');

insert into cazari (id_cazare, id_oras, tip_cazare, adresa)
values (995, 600, 'Vila', 'Bishop St. 70');

insert into cazari (id_cazare, id_oras, tip_cazare, adresa)
values (994, 100, 'Airbnb', 'Bd. Dacia, Nr. 6');

insert into cazari (id_cazare, id_oras, tip_cazare, adresa)
values (992, 600, 'Airbnb', 'Melbury Rd. 8');

select * from cazari;

insert into recenziile_atractie (id_recenzie, id_user, id_atractie)

```

Script Output x Query Result x

SQL | All Rows Fetched: 7 in 0.003 seconds

ID_CAZARE	ID_ORAS	TIP_CAZARE	ADRESA
1	999	101 Hotel	Strada Lautari, Nr. 38
2	998	200 Airbnb	Rue de Rivoli, Nr. 2
3	997	300 Hotel	Spadinalaan 97
4	996	800 Hotel	Via le Marco Polo 102
5	995	600 Vila	Bishop St. 70
6	994	100 Airbnb	Bd. Dacia, Nr. 6
7	992	600 Airbnb	Melbury Rd. 8

RECENZII_ATRACTIE

```

insert into recenziile_atractie (id_recenzie, id_user, id_atractie, detalii, nr_stele)
values (51, 1, 1001, 'Prea mare, istoric neplacut, boring', 1);

insert into recenziile_atractie (id_recenzie, id_user, id_atractie, detalii, nr_stele)
values (52, 2, 1002, 'A fost o experienta placuta, dar nu cum ma asteptam din cauza aglomeratiei.', 4);

```

```
insert into recenziile_atractie (id_recenzie, id_user, id_atractie, detalii, nr_stele)
values (53, 3, 1003, 'Super distractiv! Am invatat asa multe, recomand tuturor!', 5);
```

```
insert into recenziile_atractie (id_recenzie, id_user, id_atractie, detalii, nr_stele)
values (54, 4, 1004, 'Locatie de vis! Apartine patrimoniului UNESCO, as reveni.', 5);
```

```
insert into recenziile_atractie (id_recenzie, id_user, id_atractie, detalii, nr_stele)
values (55, 5, 1005, 'Nu a fost rau, dar nici extraordinar.', 3);
```

```
insert into recenziile_atractie (id_recenzie, id_user, id_atractie, detalii, nr_stele)
values (56, 1, 1015, 'Un loc minunat pentru relaxare, dar foarte scump', 3);
```

```
insert into recenziile_atractie (id_recenzie, id_user, id_atractie, detalii, nr_stele)
values (57, 2, 1012, 'M-am distrat de minune', 5);
```

```
insert into recenziile_atractie (id_recenzie, id_user, id_atractie, detalii, nr_stele)
values (58, 9, 1014, 'Experienta unica! Dar nu pentru copii', 5);
```

```
insert into recenziile_atractie (id_recenzie, id_user, id_atractie, detalii, nr_stele)
values (59, 7, 1002, 'Nu recomand.', 1);
```

```
insert into recenziile_atractie (id_recenzie, id_user, id_atractie, detalii, nr_stele)
values (56, 1, 1015, 'Un loc minunat pentru relaxare, dar foarte scump', 3);

insert into recenziile_atractie (id_recenzie, id_user, id_atractie, detalii, nr_stele)
values (57, 2, 1012, 'M-am distrat de minune', 5);

insert into recenziile_atractie (id_recenzie, id_user, id_atractie, detalii, nr_stele)
values (58, 9, 1014, 'Experienta unica! Dar nu pentru copii', 5);

insert into recenziile_atractie (id_recenzie, id_user, id_atractie, detalii, nr_stele)
values (59, 7, 1002, 'Nu recomand.', 1);

select * from recenziile_atractie;
```

The screenshot shows a SQL query window with the following details:

- Script Output:** Shows the executed SQL code.
- Query Result:** Shows a table with 9 rows of data, corresponding to the inserted records.
- Table Headers:** ID_RECENZIE, ID_USER, ID_ATRACTIE, DETALII.
- Data Rows:**

ID_RECENZIE	ID_USER	ID_ATRACTIE	DETALII
1	51	1	1001 Prea mare, istoric neplacut, boring
2	52	2	1002 A fost o experienta placuta, dar nu cum ma asteptam din cauza aglomeratiei.
3	53	3	1003 Super distractiv! Am invatat asa multe, recomand tuturor!
4	54	4	1004 Locatie de vis! Apartine patrimoniului UNESCO, as reveni.
5	55	5	1005 Nu a fost rau, dar nici extraordinar.
6	56	1	1015 Un loc minunat pentru relaxare, dar foarte scump
7	57	2	1012 M-am distrat de minune
8	58	9	1014 Experienta unica! Dar nu pentru copii
9	59	7	1002 Nu recomand.
- Timing:** All Rows Fetched: 9 in 0.001 seconds

UTIL_VIZITEAZA_ATRACTII

```
insert into util_viziteaza_atractii (id_vizita, id_utilizator, id_atractie, data_vizita)
values (1, 1, 1001, to_date('2023-01-15', 'YYYY-MM-DD'));
```

```
insert into util_viziteaza_atractii (id_vizita, id_utilizator, id_atractie, data_vizita)
values (2, 2, 1002, to_date('2023-02-10', 'YYYY-MM-DD'));
```

```
insert into util_viziteaza_atractii (id_vizita, id_utilizator, id_atractie, data_vizita)
values (3, 3, 1003, to_date('2023-03-05', 'YYYY-MM-DD'));
```

```
insert into util_viziteaza_atractii (id_vizita, id_utilizator, id_atractie, data_vizita)
values (4, 4, 1004, to_date('2023-04-20', 'YYYY-MM-DD'));
```

```
insert into util_viziteaza_atractii (id_vizita, id_utilizator, id_atractie, data_vizita)
values (5, 5, 1005, to_date('2023-05-15', 'YYYY-MM-DD'));
```

```
insert into util_viziteaza_atractii (id_vizita, id_utilizator, id_atractie, data_vizita)
values (6, 10, 1006, to_date('2023-06-10', 'YYYY-MM-DD'));
```

```
insert into util_viziteaza_atractii (id_vizita, id_utilizator, id_atractie, data_vizita)
values (7, 9, 1007, to_date('2023-07-05', 'YYYY-MM-DD'));
```

```
insert into util_viziteaza_atractii (id_vizita, id_utilizator, id_atractie, data_vizita)
values (8, 7, 1001, to_date('2023-08-20', 'YYYY-MM-DD'));
```

```
insert into util_viziteaza_atractii (id_vizita, id_utilizator, id_atractie, data_vizita)
values (9, 6, 1002, to_date('2023-09-15', 'YYYY-MM-DD'));
```

```
insert into util_viziteaza_atractii (id_vizita, id_utilizator, id_atractie, data_vizita)
values (10, 5, 1003, to_date('2024-01-01', 'YYYY-MM-DD'));
```

```
values (7, 9, 1007, to_date('2023-07-05', 'YYYY-MM-DD'));

insert into util_viziteaza_atractii (id_vizita, id_utilizator, id_atractie, data_vizita)
values (8, 7, 1001, to_date('2023-08-20', 'YYYY-MM-DD'));

insert into util_viziteaza_atractii (id_vizita, id_utilizator, id_atractie, data_vizita)
values (9, 6, 1002, to_date('2023-09-15', 'YYYY-MM-DD'));

insert into util_viziteaza_atractii (id_vizita, id_utilizator, id_atractie, data_vizita)
values (10, 5, 1003, to_date('2024-01-01', 'YYYY-MM-DD'));

select * from util_viziteaza_atractii;
```

The screenshot shows the Oracle SQL Developer interface. The top part is a code editor with the SQL script displayed. The bottom part is a results window titled "Query Result" showing the output of the "select * from util_viziteaza_atractii;" command. The results are presented in a table with four columns: ID_VIZITA, ID_UTILIZATOR, ID_ATRACTIE, and DATA_VIZITA. The data is as follows:

ID_VIZITA	ID_UTILIZATOR	ID_ATRACTIE	DATA_VIZITA
1	1	1	1001 15-JAN-23
2	2	2	1002 10-FEB-23
3	3	3	1003 05-MAR-23
4	4	4	1004 20-APR-23
5	5	5	1005 15-MAY-23
6	6	10	1006 10-JUN-23
7	7	9	1007 05-JUL-23
8	8	7	1001 20-AUG-23
9	9	6	1002 15-SEP-23
10	10	5	1003 01-JAN-24

6. Subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții studiate

Realizati un subprogram care pentru o tara data afiseaza: descrierile oraselor ce aparțin tarii, d-urile oraselor din acea tara, si id-urile cazarilor care se pot gasi in orasele din acea tara.

```
CREATE OR REPLACE PROCEDURE ex6(v_nume_tara IN VARCHAR2) AS

TYPE index_by_table_type IS TABLE OF VARCHAR2(50) INDEX BY VARCHAR2(50);
v_index_by index_by_table_type;

TYPE nested_table_type IS TABLE OF NUMBER(5)INDEX BY PLS_INTEGER;
v_nested nested_table_type;

TYPE varray_type IS VARRAY(10) OF NUMBER(5);
v_varray varray_type:= varray_type();

v_id_tara NUMBER(5);

BEGIN
    SELECT id_tara INTO v_id_tara FROM tari WHERE denumire = v_nume_tara;

    IF v_id_tara IS NOT NULL THEN

        FOR o IN (SELECT UNIQUE id_oras FROM orase WHERE id_tara = v_id_tara) LOOP
            v_nested(v_nested.COUNT + 1) := o.id_oras;
        END LOOP;

        FOR caz IN (SELECT id_cazare, id_oras FROM cazari) LOOP
            FOR i IN 1..v_nested.COUNT LOOP
                IF caz.id_oras = v_nested(i) THEN
                    v_varray.EXTEND;
                    v_varray(v_varray.LAST) := caz.id_cazare;
                END IF;
            END LOOP;
        END LOOP;

        FOR inregistrare IN (SELECT t.denumire, t.climat, o.descriere
                             FROM orase o
                             JOIN tari t ON o.id_tara = t.id_tara
                             WHERE t.id_tara = v_id_tara)
        LOOP
            v_index_by(inregistrare.denumire) := inregistrare.descriere;
        END LOOP;
    END IF;
END;
```

```

DBMS_OUTPUT.PUT_LINE('Descriere: ' || v_index_by(v_nume_tara));

END LOOP;

FOR i IN v_nested.FIRST..v_nested.LAST
LOOP
    DBMS_OUTPUT.PUT_LINE('ID Oras: ' ||v_nested(i));
END LOOP;

FOR i IN v_varray.FIRST..v_varray.LAST
LOOP
    DBMS_OUTPUT.PUT_LINE('ID Cazare: ' || v_varray(i));
END LOOP;

ELSE
    DBMS_OUTPUT.PUT_LINE('Nu există țară cu numele ' || v_nume_tara);
END IF;
END ex6;
/

```

The screenshot shows the Oracle SQL Developer interface with the 'Worksheet' tab selected. The code for procedure `ex6` is displayed in the worksheet window.

```

--subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții studiate.
-- Realizati un subprogram care pentru o tara data afiseaza: descrierile oraselor ce apartin tarii,
--idurile oraselor din acea tara, si idurile cazarilor care se pot gasi in orasele din acea tara

CREATE OR REPLACE PROCEDURE ex6(v_nume_tara IN VARCHAR2) AS

    TYPE index_by_table_type IS TABLE OF VARCHAR2(50) INDEX BY VARCHAR2(50);
    v_index_by index_by_table_type;

    TYPE nested_table_type IS TABLE OF NUMBER(5) INDEX BY PLS_INTEGER;
    v_nested nested_table_type;

    TYPE varray_type IS VARRAY(10) OF NUMBER(5);
    v_varray varray_type:= varray_type();

    v_id_tara NUMBER(5);

BEGIN
    SELECT id_tara INTO v_id_tara FROM tari WHERE denumire = v_nume_tara;
    IF v_id_tara IS NOT NULL THEN

```

```
IF v_id_tara IS NOT NULL THEN

    FOR o IN (SELECT UNIQUE id_oras FROM orase WHERE id_tara = v_id_tara) LOOP
        v_nested(v_nested.COUNT + 1) := o.id_oras;
        END LOOP;

        FOR caz IN (SELECT id_cazare, id_oras FROM cazari) LOOP
            FOR i IN 1..v_nested.COUNT LOOP
                IF caz.id_oras = v_nested(i) THEN
                    v_varray.EXTEND;
                    v_varray(v_varray.LAST) := caz.id_cazare;
                END IF;
            END LOOP;
        END LOOP;

        FOR inregistrare IN (SELECT t.denumire, t.climat, o.descriere
                             FROM orase o
                             JOIN tari t ON o.id_tara = t.id_tara
                             WHERE t.id_tara = v_id_tara)
            LOOP
```

Worksheet Query Builder

```
LOOP

    v_index_by(inregistrare.denumire) := inregistrare.descriere;
    DBMS_OUTPUT.PUT_LINE('Descriere: ' || v_index_by(v_numar_tara));

    END LOOP;

    FOR i IN v_nested.FIRST..v_nested.LAST
    LOOP
        DBMS_OUTPUT.PUT_LINE('ID Oras: ' || v_nested(i));
    END LOOP;

    FOR i IN v_varray.FIRST..v_varray.LAST
    LOOP
        DBMS_OUTPUT.PUT_LINE('ID Cazare: ' || v_varray(i));
    END LOOP;
    ELSE
        DBMS_OUTPUT.PUT_LINE('Nu există țară cu numele ' || v_numar_tara);
    END IF;
END ex6;
/
```

The screenshot shows the Oracle SQL Developer interface. The Worksheet window contains a PL/SQL block:

```

FOR i IN v_varray.FIRST..v_varray.LAST
LOOP
    DBMS_OUTPUT.PUT_LINE('ID Cazare: ' || v_varray(i));
END LOOP;
ELSE
    DBMS_OUTPUT.PUT_LINE('Nu există țară cu numele ' || v_num_tara);
END IF;
END ex6;
/

BEGIN
    ex6('Marea Britanie');
END;
/

```

The Script Output window shows the results of the execution:

```

Descriere: Capitala Marii Britanii
Descriere:
Descriere: Cea mai mare celebrare a Sf Patrick din Anglia
ID Oras: 600
ID Oras: 601
ID Oras: 602
ID Cazare: 995
ID Cazare: 992

```

7. Subprogram stocat individual care foloseste 2 cursoare, dintre care unul parametrizat dependent de celalalt

Dat fiind id-ul unui utilizator, sa se retine intr-un cursor detaliile de calatorie ale acestuia, apoi sa se afiseze climatele tarilor in care se afla orasele in care acest user a calatorit.

```
CREATE OR REPLACE PROCEDURE ex7(e_id_utilizator IN NUMBER) IS
```

```
    CURSOR cursor_calatorii IS
```

```
        SELECT c.id_calatorie, o.nume AS nume_oras, t.id_tara, t.climat
        FROM calatorii c
        JOIN orase o ON c.id_oras = o.id_oras
        JOIN tari t ON o.id_tara = t.id_tara
        WHERE c.id_utilizator = e_id_utilizator;
```

--cursor parametrizat dependent de cursor_calatorie:

```
CURSOR cursor_climat (e_id_tara IN NUMBER) IS
```

```
    SELECT climat
```

```
    FROM tari
```

```
WHERE id_tara = e_id_tara;

v_cal_id calatorii.id_calatorie%TYPE;
v_nume_oras orase.nume%TYPE;
v_id_tara tari.id_tara%TYPE;
v_climat tari.climat%TYPE;

BEGIN

    OPEN cursor_calatorii;

    LOOP
        FETCH cursor_calatorii INTO v_cal_id, v_nume_oras, v_id_tara, v_climat;
        EXIT WHEN cursor_calatorii%NOTFOUND;

        OPEN cursor_climat(v_id_tara);

        LOOP
            FETCH cursor_climat INTO v_climat;
            EXIT WHEN cursor_climat%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE(v_climat);
        END LOOP;

        CLOSE cursor_climat;

    END LOOP;

    CLOSE cursor_calatorii;
END ex7;
/
```

```
CREATE OR REPLACE PROCEDURE ex7(e_id_utilizator IN NUMBER) IS
  CURSOR cursor_calatorii IS
    SELECT c.id_calatorie, o.numere AS nume_oras, t.id_tara, t.climat
    FROM calatorii c
    JOIN orase o ON c.id_oras = o.id_oras
    JOIN tari t ON o.id_tara = t.id_tara
    WHERE c.id_utilizator = e_id_utilizator;

    --cursor parametrizat dependent de cursor_calatorie:
    CURSOR cursor_climat (e_id_tara IN NUMBER) IS
      SELECT climat
      FROM tari
      WHERE id_tara = e_id_tara;

    v_cal_id calatorii.id_calatorie%TYPE;
    v_numere_oras orase.numere%TYPE;
    v_id_tara tari.id_tara%TYPE;
    v_climat tari.climat%TYPE;

BEGIN
  OPEN cursor_calatorii;
```

```
  OPEN cursor_calatorii;

  LOOP
    FETCH cursor_calatorii INTO v_cal_id, v_numere_oras, v_id_tara, v_climat;
    EXIT WHEN cursor_calatorii%NOTFOUND;

    OPEN cursor_climat(v_id_tara);

    LOOP
      FETCH cursor_climat INTO v_climat;
      EXIT WHEN cursor_climat%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE(v_climat);
    END LOOP;

    CLOSE cursor_climat;

  END LOOP;

  CLOSE cursor_calatorii;
END ex7;
/
```

```

        cursor cursor_1;
END ex7;
/

BEGIN
    ex7(1);
END;
/

```

Script Output | Query Result | Task completed in 0.05 seconds

```

temperat
temperat
oceanic

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

```

8. Subprogram de tip functie - cu 3 tabele si minim 2 exceptii proprii

Afiseaza suma stelelor din recenzii doar daca atractia a fost vizitata de utilizatorul care a scris recenzia.

```

CREATE OR REPLACE FUNCTION ex8(v_username utilizatori.username%type)
RETURN NUMBER IS rezultat NUMBER(2);
contor NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO contor
    FROM utilizatori
    WHERE username = v_username;

    IF contor = 0 THEN
        RAISE_APPLICATION_ERROR(-20000, 'Nu exista utilizatori cu numele dat');
    ELSIF contor>1 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi utilizatori cu numele dat');
    END IF;

    SELECT COUNT(r.nr_stele)
    INTO rezultat
    FROM utilizatori u

```

```

JOIN recenzii_atractie r ON r.id_user = u.id_utilizator
JOIN util_viziteaza_atractii a ON a.id_atractie = r.id_atractie
AND u.username = v_username;

RETURN rezultat;
END ex8;
/
BEGIN
DBMS_OUTPUT.PUT_LINE(ex8('anapopescu123'));
END;
/

```

```

CREATE OR REPLACE FUNCTION ex8(v_username utilizatori.username%type)
    RETURN NUMBER IS rezultat NUMBER(2);
    contor NUMBER;
    BEGIN
        SELECT COUNT(*)
        INTO contor
        FROM utilizatori
        WHERE username = v_username;

        IF contor = 0 THEN
            RAISE_APPLICATION_ERROR(-20000, 'Nu exista utilizatori cu numele dat');
        ELSIF contor>1 THEN
            RAISE_APPLICATION_ERROR(-20001, 'Există mai multi utilizatori cu numele dat');
        END IF;

        SELECT COUNT(r.nr_stele)
        INTO rezultat
        SELECT COUNT(r.nr_stele)
        INTO rezultat
        FROM utilizatori u
        JOIN recenzii_atractie r ON r.id_user = u.id_utilizator
        JOIN util_viziteaza_atractii a ON a.id_atractie = r.id_atractie
        AND u.username = v_username;

        RETURN rezultat;
    END ex8;
    /

```

```

BEGIN
    DBMS_OUTPUT.PUT_LINE(ex8('anapopescu123'));
END;
/

```

DECLARE

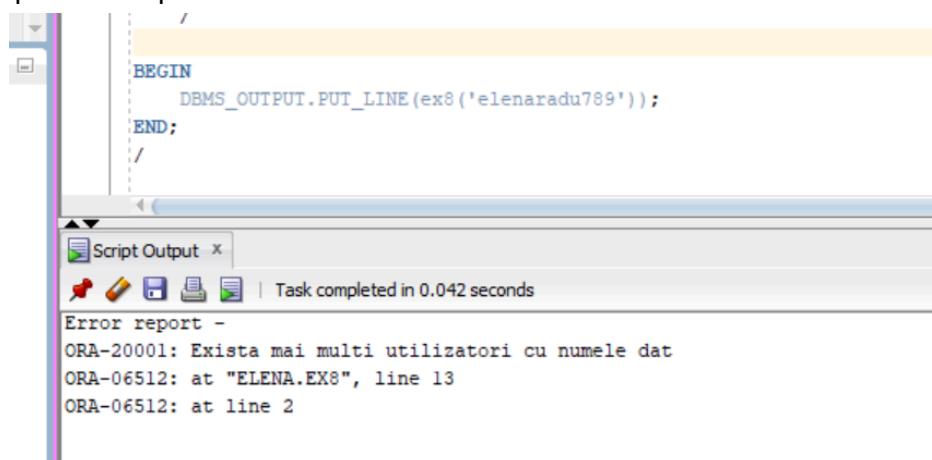
Script Output X
 Task completed in 0.033 seconds

2

PL/SQL procedure successfully completed.

Exceptii:

-prima exceptie

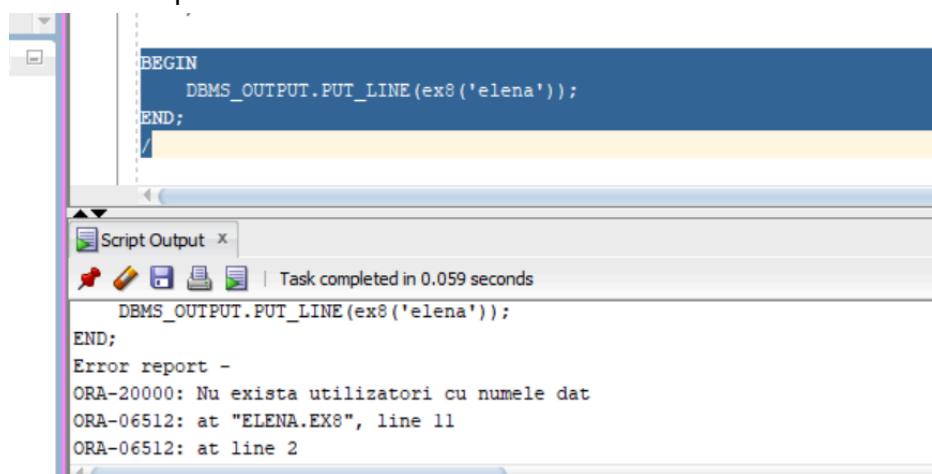


```
BEGIN
    DBMS_OUTPUT.PUT_LINE(ex8('elenaradu789'));
END;
/
```

Script Output | Task completed in 0.042 seconds

Error report -
ORA-20001: Există mai mulți utilizatori cu numele dat
ORA-06512: at "ELENA.EX8", line 13
ORA-06512: at line 2

-a doua exceptie



```
BEGIN
    DBMS_OUTPUT.PUT_LINE(ex8('elena'));
END;
/
```

Script Output | Task completed in 0.059 seconds

```
DBMS_OUTPUT.PUT_LINE(ex8('elena'));
```

Error report -
ORA-20000: Nu există utilizatori cu numele dat
ORA-06512: at "ELENA.EX8", line 11
ORA-06512: at line 2

9. Subprogram de tip procedura

- 5 tabele
- Exceptiile NO_DATAT_FOUND si TOO_MANY_ROWS

--ex9

--Fiind dat username-ul unui utilizator, sa se afiseze daca a calatorit undeva
--numele orasului, climatul tarii in care se afla orasul respectiv si categoria
--mijlocului de transport cu care s-a deplasat. Tratati cazurile in care
--există mai multi utilizatori cu același nume sau utilizatorul nu are
--nicio calatorie înregistrată.

```
CREATE OR REPLACE PROCEDURE ex9(e_username IN utilizatori.username%TYPE) IS
    v_id_utilizator utilizatori.id_utilizator%TYPE;
    v_id_oras orase.id_oras%TYPE;
    v_denumire_tara tari.denumire%TYPE;
    v_climat_tara tari.climat%TYPE;
    v_nume_oras orase.nume%TYPE;
```

```

v_tip_transport mijloace_transport.denumire%TYPE;

CURSOR calatorii_cursor IS
    SELECT u.id_utilizator, c.id_oras, o.nume, t.denumire, t.climat
    FROM utilizatori u
    JOIN calatorii c ON u.id_utilizator = c.id_utilizator
    JOIN orase o ON c.id_oras = o.id_oras
    JOIN tari t ON o.id_tara = t.id_tara
    WHERE u.username = e_username;

BEGIN
    BEGIN

        SELECT id_utilizator INTO v_id_utilizator
        FROM utilizatori
        WHERE username = e_username;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RAISE_APPLICATION_ERROR(-20000, 'Nu exista utilizatori cu numele dat');

        WHEN TOO_MANY_ROWS THEN
            RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi utilizatori cu numele dat');

    END;
    --daca exista
    IF v_id_utilizator IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE('Utilizatorul ' || e_username || ' a calatorit in:');

        FOR calatorie IN calatorii_cursor LOOP
            v_id_oras := calatorie.id_oras;
            v_nume_oras := calatorie.nume;
            v_denumire_tara := calatorie.denumire;
            v_climat_tara := calatorie.climat;

            -- tipul mijlocului de transport
            BEGIN
                SELECT denumire INTO v_tip_transport
                FROM mijloace_transport
                WHERE id_mijloc IN (
                    SELECT id_mijloc
                    FROM calatorii
                    WHERE id_oras = v_id_oras AND id_utilizator = v_id_utilizator
                );
            END;
        END;

        DBMS_OUTPUT.PUT_LINE('Oras: ' || v_nume_oras);
        DBMS_OUTPUT.PUT_LINE('Tara: ' || v_denumire_tara);
        DBMS_OUTPUT.PUT_LINE('Climat: ' || v_climat_tara);
    END;

```

```

        DBMS_OUTPUT.PUT_LINE('Mijloc Transport: ' || v_tip_transport);
    END LOOP;
END IF;
END ex9;
/
BEGIN
    ex9('andreigheorghe202');
END;
/

```

The screenshot shows the Oracle SQL Developer interface with the code editor open. The code is a PL/SQL procedure named ex9. A cursor named calatorii_cursor is declared, which performs a multi-table join across four tables: utilizatori, calatorii, orase, and tari. The WHERE clause filters for a specific user. Below the cursor declaration, there is a BEGIN block containing logic to handle user existence and a conditional block to check if the user exists.

```

CREATE OR REPLACE PROCEDURE ex9(e_username IN utilizatori.username%TYPE) IS
    v_id_utilizator utilizatori.id_utilizator%TYPE;
    v_id_oras orase.id_oras%TYPE;
    v_denumire_tara tari.denumire%TYPE;
    v_climat_tara tari.climat%TYPE;
    v_nume_oras orase.nume%TYPE;
    v_tip_transport mijloace_transport.denumire%TYPE;

    CURSOR calatorii_cursor IS
        SELECT u.id_utilizator, c.id_oras, o.nume, t.denumire, t.climat
        FROM utilizatori u
        JOIN calatorii c ON u.id_utilizator = c.id_utilizator
        JOIN orase o ON c.id_oras = o.id_oras
        JOIN tari t ON o.id_tara = t.id_tara
        WHERE u.username = e_username;

    BEGIN
        BEGIN
            SELECT id_utilizator INTO v_id_utilizator
            FROM utilizatori
            WHERE username = e_username;
        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                RAISE_APPLICATION_ERROR(-20000, 'Nu exista utilizatori cu numele dat');

            WHEN TOO_MANY_ROWS THEN
                RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi utilizatori cu numele dat');

        END;
        --daca exista
        IF v_id_utilizator IS NOT NULL THEN
    
```

```

        RAISE;
    END;
    --daca exista
    IF v_id_utilizator IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE('Utilizatorul ' || e_username || ' a calatorit in:');

        FOR calatorie IN calatorii_cursor LOOP
            v_id_oras := calatorie.id_oras;
            v_nume_oras := calatorie.nume;
            v_denumire_tara := calatorie.denumire;
            v_climat_tara := calatorie.climat;

            -- tipul mijlocului de transport
            BEGIN
                SELECT denumire INTO v_tip_transport
                FROM mijloace transport

```

```

                    SELECT denumire INTO v_tip_transport
                    FROM mijloace_transport
                    WHERE id_mijloc IN (
                        SELECT id_mijloc
                        FROM calatorii
                        WHERE id_oras = v_id_oras AND id_utilizator = v_id_utilizator
                    );
                END;

                DBMS_OUTPUT.PUT_LINE('Oras: ' || v_nume_oras);
                DBMS_OUTPUT.PUT_LINE('Tara: ' || v_denumire_tara);
                DBMS_OUTPUT.PUT_LINE('Climat: ' || v_climat_tara);
                DBMS_OUTPUT.PUT_LINE('Mijloc Transport: ' || v_tip_transport);
            END LOOP;
        END IF;
    END ex9;

```

```

        DBMS_OUTPUT.PUT_LINE('Oras: ' || v_nume_oras);
        DBMS_OUTPUT.PUT_LINE('Tara: ' || v_denumire_tara);
        DBMS_OUTPUT.PUT_LINE('Climat: ' || v_climat_tara);
        DBMS_OUTPUT.PUT_LINE('Mijloc Transport: ' || v_tip_transport);
    END LOOP;
    END IF;
END ex9;
/

```

```

BEGIN
    ex9('andreibogheorghe202');
END;
/

```

Script Output x

Task completed in 0.038 seconds

```

Procedure EX9 compiled

Utilizatorul andreibogheorghe202 a calatorit in:
Oras: Brasilia
Tara: Brazilia
Climat: tropical
Mijloc Transport: UltraJet X10

```

Prima exceptie:

The screenshot shows a PL/SQL script in the editor and its execution output in the Script Output window.

```
END ex9;
/
BEGIN
    ex9('elena');
END;
/
```

Script Output window content:

```
Task completed in 0.045 seconds
END;
Error report -
ORA-20000: Nu exista utilizatori cu numele dat
ORA-06512: at "ELENA.EX9", line 25
ORA-06512: at line 2
20000. 00000 - "%s"
*Cause: The stored procedure 'raise_application_error'
      was called which causes this error to be generated.
```

A doua exceptie:

The screenshot shows a PL/SQL script in the editor and its execution output in the Script Output window.

```
/
BEGIN
    ex9('elenaradu789');
END;
/
```

Script Output window content:

```
Task completed in 0.033 seconds
END;
Error report -
ORA-20001: Există mai multi utilizatori cu numele dat
ORA-06512: at "ELENA.EX9", line 28
ORA-06512: at line 2
```

10. Trigger LMD la nivel de comanda

create or replace trigger ex10

```

before delete on orase
begin
    RAISE_APPLICATION_ERROR(-20001, 'Este interzisa stergerea datelor din tabelul
ORASE');
end;
/
--ex10
-- un trigger de tip LMD la nivel de comandă

create or replace trigger ex10
    before delete on orase
begin
    RAISE_APPLICATION_ERROR(-20001, 'Este interzisa stergerea datelor din tabelul ORASE');
end;
/
delete from orase
WHERE id_oras = '101';

```

Trigger EX10 compiled

Error starting at line : 425 in command -
 delete from orase
 WHERE id_oras = '101'
 Error at Command Line : 425 Column : 13
 Error report -
 SQL Error: ORA-20001: Este interzisa stergerea datelor din tabelul ORASE
 ORA-06512: at "ELENA.EX10", line 2
 ORA-04088: error during execution of trigger 'ELENA.EX10'

11. Trigger LMD la nivel de linie

Trigger-ul interzice cresterea pretului unui mijloc de transport cu mai mult de 50%

```

create or replace trigger ex11
    before update on mijloace_transport
        for each row
begin
    if (:new.pret - :old.pret > :old.pret*(1/2)) then
        raise_application_error(-20001, 'Pretul nu poate fi crescut cu mai mult de 50%');
    end if;
end;
/

```

```

create or replace trigger ex11
  before update on mijloace_transport
  for each row
begin
  if (:new.pret - :old.pret > :old.pret*(1/2)) then
    raise_application_error(-20001, 'Pretul nu poate fi crescut cu mai mult de 50%');
  end if;
end;
/

update mijloace_transport
set pret = pret + 90
where id_mijloc = 100009;

```

Script Output X Query Result X

Trigger EX11 compiled

Error starting at line : 459 in command -
update mijloace_transport
set pret = pret + 90
WHERE id_mijloc = 100009
Error at Command Line : 459 Column : 8
Error report -
SQL Error: ORA-20001: Pretul nu poate fi crescut cu mai mult de 50%
ORA-06512: at "ELENA.EX11", line 3
ORA-04088: error during execution of trigger 'ELENA.EX11'

12. Trigger de tip LDD

-- un trigger care se declanseaza de fiecare data la executia unei operatii LDD
-- si salveaza modificarile in tabela modificari

```

create table modificari(
  utilizator varchar2(30),
  obiect_modificat varchar2(50),
  activitate varchar2(50));

create or replace trigger ex12
  after create or drop or alter on schema

BEGIN
  insert into modificari
  values(SYS.LOGIN_USER, SYS.DICTIONARY_OBJ_NAME,
SYS.DICTIONARY_OBJ_TYPE);

END;
/

create table cazari_e as select * from cazari;

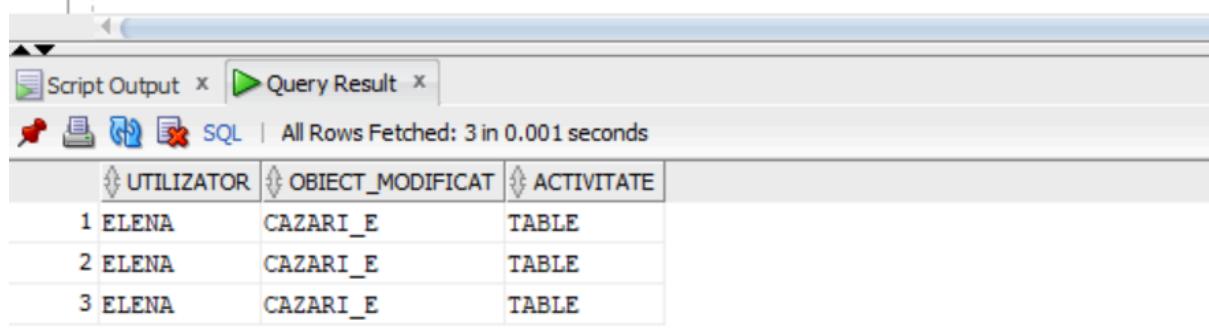
```

```
alter table cazari_e  
drop column adresa;
```

```
drop table cazari_e;
```

```
select * from modificari;
```

```
create table modificari(  
    utilizator varchar2(30),  
    obiect_modificat varchar2(50),  
    activitate varchar2(50));  
  
create or replace trigger ex12  
    after create or drop or alter on schema  
  
BEGIN  
    insert into modificari  
    values(SYS.LOGIN_USER, SYS.DICTIONARY_OBJ_NAME, SYS.DICTIONARY_OBJ_TYPE);  
  
END;  
/  
  
create table cazari_e as select * from cazari;  
  
alter table cazari_e  
drop column adresa;  
  
drop table cazari_e;
```



The screenshot shows the Oracle SQL Developer interface. At the top, there are two tabs: "Script Output" and "Query Result". The "Script Output" tab is active, showing the execution of the provided PL/SQL script. The "Query Result" tab shows the output of the "select * from modificari;" query, which returns three rows of data:

	UTILIZATOR	OBJECT_MODIFICAT	ACTIVITATE
1	ELENA	CAZARI_E	TABLE
2	ELENA	CAZARI_E	TABLE
3	ELENA	CAZARI_E	TABLE