

SISTEME DE GESTIUNE A BAZELOR DE DATE

---

---

**SISTEM DE GESTIONARE A BAZEI DE DATE PENTRU O  
PLATFORMĂ ONLINE DE REZERVĂRI**

**AIRBNB**

---

---

Facultatea de Matematică și Informatică

Universitatea din București

Andra-Mihaela Andruță

Seria 23

Grupa 232

An Universitar 2024-2025

## CUPRINS

INTRODUCERE .....	3
1. BAZA DE DATE. UTILITATEA EI .....	4
Utilitatea bazei de date .....	4
2. DIAGRAMA ERD .....	5
3. DIAGRAMA CONCEPTUALĂ .....	6
4. DEFINIRE TABELE. CONSTRÂNGERI DE DATE .....	7
Adăugare constrângeri pe tabele .....	16
5. INSERĂRI ÎN TABELE .....	19
6. EXERCIȚIUL 6 – SUBPROGRAM STOCAT INDEPENDENT .....	32
7. EXERCIȚIUL 7 – SUBPROGRAM STOCAT INDEPENDENT + CURSOARE .....	41
8. EXERCIȚIUL 8 – SUBPROGRAM STOCAT INDEPENDENT – FUNCȚIE .....	50
9. EXERCIȚIUL 9 - SUBPROGRAM STOCAT INDEPENDENT – PROCEDURĂ .....	60
10. EXERCIȚIUL 10 – TRIGGER LMD – LA NIVEL DE COMANDĂ .....	73
11. EXERCIȚIUL 11- TRIGGER LMD – LA NIVEL DE LINIE .....	77
12. EXERCIȚIUL 12 – TRIGGER LDD .....	84
13. EXERCIȚIUL 13 – PACHET + DATE COMPLEXE .....	87

## **INTRODUCERE**

Tema proiectului meu este dezvoltarea unui sistem pentru gestionarea procesului de rezervare a locuințelor unor gazde de către utilizatorii platformei mele de tip Airbnb.

Proiectul meu reprezintă continuarea ideii pe care am dezvoltat-o în cadrul materiei Baze de Date din primul an deoarece este o idee care încă de atunci mi-a stârnit curiozitatea pentru că îmi doresc să eficientizez evidența și managementul rezervărilor, evidența locuințelor disponibile dintr-o anumită zonă, într-o anumită perioadă, precum și posibila generare de rapoarte a câștigurilor obținute de către gazde.

## **INFRASTRUCTURA UTILIZATĂ**

- **VERSIUNE SGBD :**

Oracle Database – versiunea 19c

Patch level : 19.3.0.0.

- **CONFIGURATIE HARDWARE**

Sistemul întregului proiect a fost implementat pe un laptop cu procesor Intel Core i5, 8th Gen, cu o frecvență de 1600 MHz, asigurând o performanță suficientă.

Memoria fizică totală este de 8 GB RAM care asigură rularea fluentă a instanței Oracle Database.

Memoria total alocată schemei mele este de 910 MB RAM care îi asigură rularea fluentă.

- **CONFIGURAȚIE SOFTWARE**

Sistemul de operare utilizat a fost Windows 11 Pro care a oferit un suport excelent aplicațiilor.

**NU** a fost utilizată o mașină virtuală, baza de date rulând direct pe laptopul fizic, utilizând resursele hardware disponibile.

## **1. BAZA DE DATE. UTILITATEA EI**

Modelul de date va gestiona informații legate de managementul anumitor locuințe care sunt disponibile pentru a fi rezervate într-o anumintă perioadă de timp.

Astfel,există mai multe tipuri de locuințe ,ce se găsesc în diverse locații,locații care la rândul lor se găsesc în țări și regiuni.Locuințele pot fi rezervate de către utilizatorii platformei.

Totodată, putem accesa si informații cu privire la gazdele locuintelor ce pot fi accesate pe platforma.

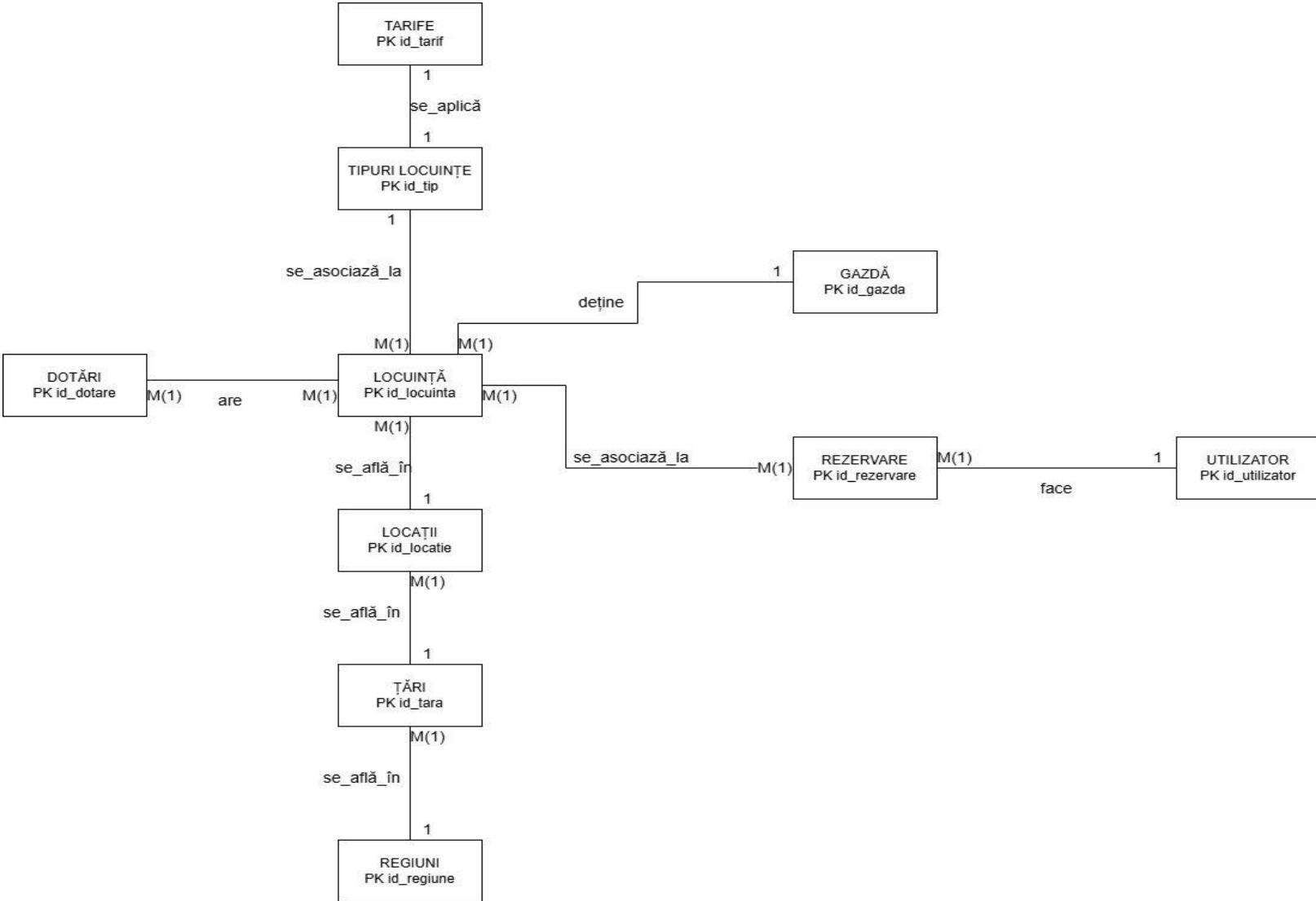
De asemenea,în cadrul bazei de date se va ține evidența gazdelor proprietăților respective ,a utilizatorilor platformei,cât si informații cu privire la dotările și tariful unei locuințe disponibile,dar și informații ale rezervărilor făcute.

### **Utilitatea bazei de date**

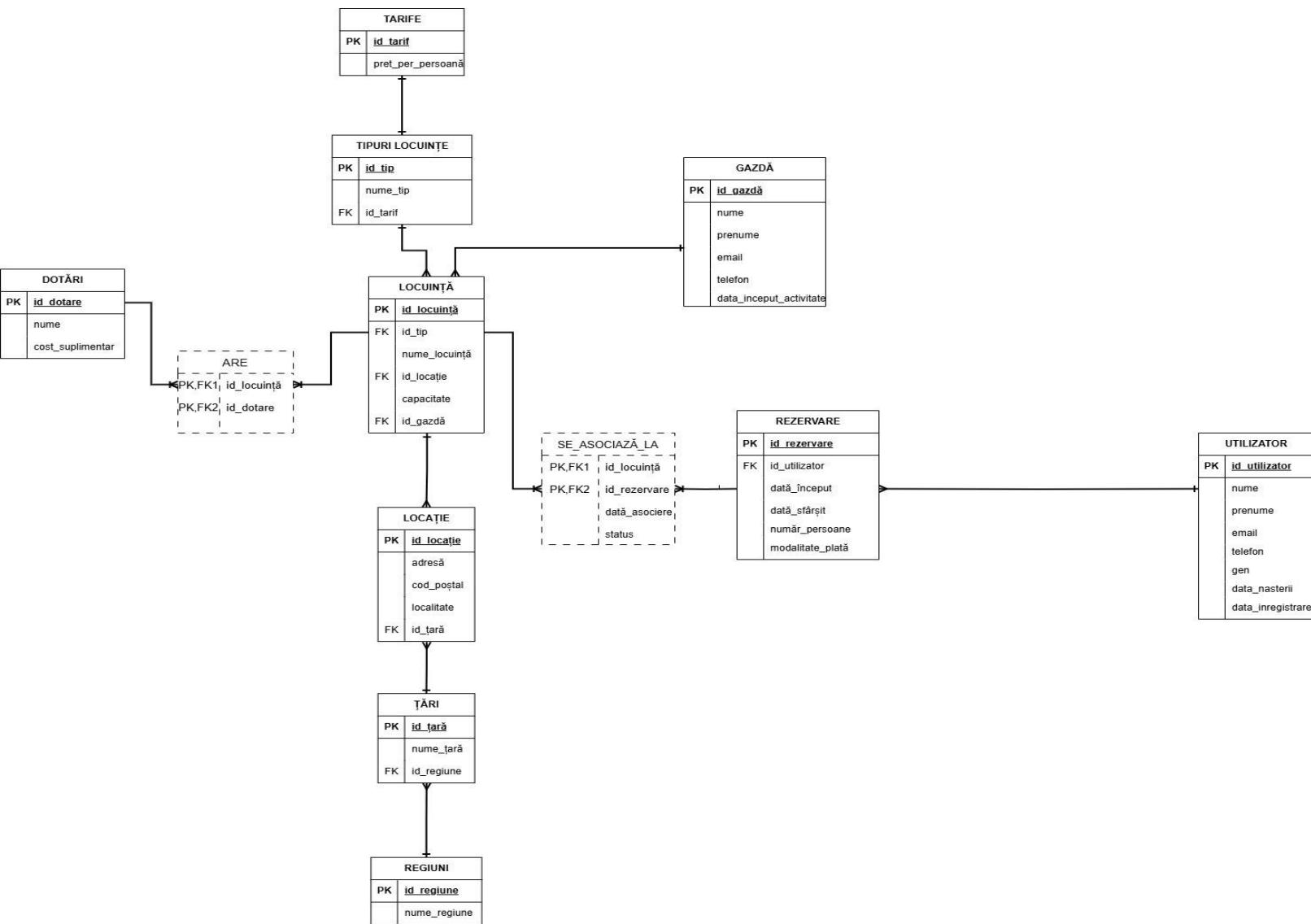
Baza de date este utilă deoarece consider că este destul de interesant, dificil și în același timp esențial gestionarea locuințelor disponibile destinate rezervării,cât și evidența tuturor datelor implicate.

Aceasta permite stocarea si gestionarea detaliilor utilizatorilor, cum ar fi numele, data nașterii și informațiile de contact.De asemenea prin intermediul său gazdele ar putea să își emită rapoarte cu privire la locuințele pe care le dețin, împreună cu rezervările făcute în cadrul acestora și câștigurile înregistrate în urma lor.Totodată,gazdele pot analiza datele în vederea ajustării tarifelor/dotărilor.

## 2. DIAGRAMA ERD



### 3. DIAGRAMA CONCEPTUALĂ



#### **4. DEFINIRE TABELE. CONSTRÂNGERI DE DATE.**

Începând cu această secțiune,o să atașez codul aferent fiecărei cerințe,însoțit de print-screen pentru a arăta corectitudinea codului și rezultatele execuției sale.

În ceea ce privește definirea tabelelor,am ales inițial să imi fac secvențe pentru a îmi ușura procesul de inserare în tabele,aferent exercițiului 5.

--secvențe

```
create sequence seq_utilizator  
increment by 1  
start with 100  
maxvalue 999  
nocycle;
```

```
create sequence seq_gazda
```

```
increment by 1  
start with 10  
maxvalue 99  
nocycle;
```

```
create sequence seq_rezervare
```

```
increment by 1  
start with 1000  
maxvalue 9999  
nocycle;
```

```
create sequence seq_locuinta
```

increment by 10

start with 200

maxvalue 9999

nocycle;

```
create sequence seq_tarife
```

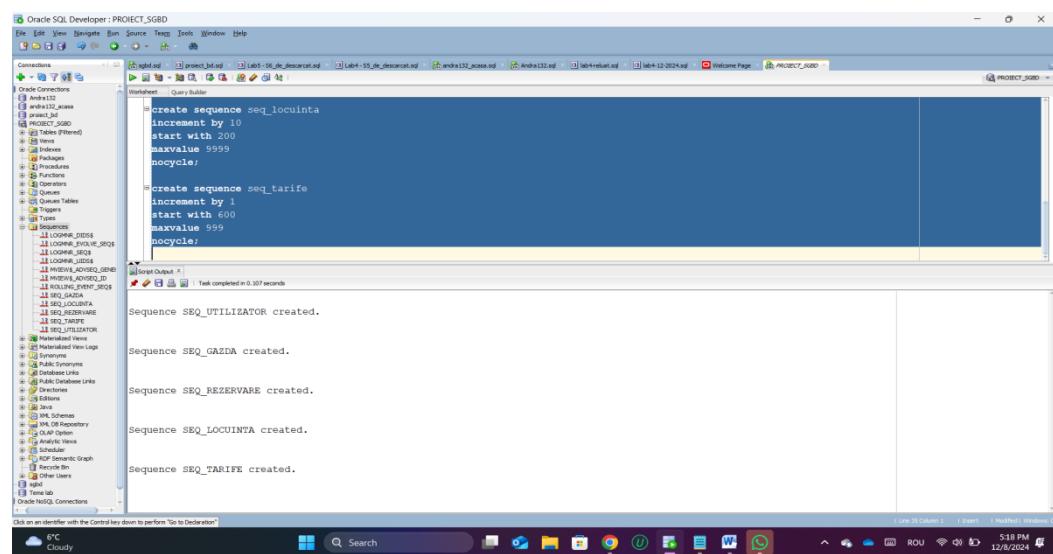
increment by 1

start with 600

maxvalue 999

nocycle;

În cadrul imaginii se poate observa rularea corectă și prezența secvențelor în baza de date, la secțiunea Sequences.



```

create table regiuni(
    id_regiune number constraint pk_regiune primary key,
    nume_regiune varchar2(30)
);

```

```

create table regiuni(
    id_regiune number constraint pk_regiune primary key,
    nume_regiune varchar2(30)
);

Table REGIUNI created.

Name          Null?    Type
-----        -----   -----
ID_REGIUNE    NOT NULL NUMBER
NUME_REGIUNE      VARCHAR2 (30)

```

```

create table tari(
    id_tara char(2) constraint pk_tara primary key,
    nume_tara varchar2(30),
    id_regiune number,
    constraint fk_tari foreign key (id_regiune) references regiuni(id_regiune)
);

```

```

create table tari(
    id_tara char(2) constraint pk_tara primary key,
    nume_tara varchar2(30),
    id_regiune number,
    constraint fk_tari foreign key (id_regiune) references regiuni(id_regiune)
);
desc tari;

```

```

Table TARI created.

Name          Null?    Type
-----        -----   -----
ID_TARA       NOT NULL CHAR (2)
NUME_TARA      VARCHAR2 (30)
ID_REGIUNE      NUMBER

```

```

create table locatie(
    id_locatie number constraint pk_locatie primary key,
    adresa varchar2(100),
    cod_postal varchar2(10),
    localitate varchar2(30),
    id_tara char(2),
    constraint fk_locatii foreign key (id_tara) references tari(id_tara));

```

```

create table locatie(
    id_locatie number constraint pk_locatie primary key,
    adresa varchar2(100),
    cod_postal varchar2(10),
    localitate varchar2(30),
    id_tara char(2),
    constraint fk_locatii foreign key (id_tara) references tari(id_tara)
);
desc locatie;

```

Script Output | Task completed in 0.627 seconds

Table LOCATIE created.

ERROR:  
ORA-04043: object locatii does not exist

Name	Null?	Type
ID_LOCATIE	NOT NULL	NUMBER
ADRESA		VARCHAR2(100)
COD_POSTAL		VARCHAR2(10)
LOCALITATE		VARCHAR2(30)
ID_TARA		CHAR(2)

```

create table tipuri_locuiente(
    id_tip number constraint pk_tip primary key,
    nume_tip varchar2(20),
    id_tarif number,
    constraint fk_tarif_tip foreign key (id_tarif) references tarife(id_tarif)
);
desc tipuri_locuiente;

```

```

create table tipuri_locuiente(
    id_tip number constraint pk_tip primary key,
    nume_tip varchar2(20)
);
desc tipuri_locuiente;

```

Script Output | Task completed in 0.731 seconds

Table TIPURI\_LOCUINTE created.

Name	Null?	Type
ID_TIP	NOT NULL	NUMBER
NUME_TIP		VARCHAR2(20)

```

create table dotari(
    id_dotare number constraint pk_dotare primary key,
    nume varchar2(20),
    cost_suplimentar number
);

```

The screenshot shows the SQL script being run in Oracle SQL Developer. The code is identical to the one above. Below the code, the output window shows:

```

Table DOTARI created.

Name          Null?    Type
-----        -----   -----
ID_DOTARE     NOT NULL NUMBER
NUME          VARCHAR2(20)
COST_SUPLIMENTAR NUMBER

```

```

create table gazda(
    id_gazda number constraint pk_gazda primary key,
    nume varchar2(30),
    prenume varchar2(30),
    email varchar2(40),
    telefon varchar(20),
    data_inceput_activitate date
);

```

```

create table gazda(
    id_gazda number constraint pk_gazda primary key,
    nume varchar2(30),
    prenume varchar2(30),
    email varchar2(40),
    telefon varchar(20),
    data_inceput_activitate date
);

desc gazda;

create table rezervare(
    ...
);

Script Output X | Task completed in 0.696 seconds
Table GAZDA created.

Name Null? Type
-----
ID_GAZDA NOT NULL NUMBER
NUME VARCHAR2(30)
PRENUME VARCHAR2(30)
EMAIL VARCHAR2(40)
TELEFON VARCHAR2(20)
DATA_INCEPUT_ACTIVITATE DATE

```

create table locuinta(

    id\_locuinta number constraint pk\_locuinta primary key,

    id\_tip number,

    nume\_locuinta varchar2(30),

    id\_locatie number,

    capacitate number,

    id\_gazda number,

    constraint fk\_locuinta foreign key (id\_tip) references tipuri\_locuinte(id\_tip),

    constraint fk\_locuinta2 foreign key (id\_locatie) references locatie(id\_locatie),

    constraint fk\_locuinta3 foreign key (id\_gazda) references gazda(id\_gazda)

);

```

create table locuinta(
    id_locuinta number constraint pk_locuinta primary key,
    id_tip number,
    nume_locuinta varchar2(30),
    id_locatie number,
    capacitate number,
    id_gazda number,
    constraint fk_locuinta foreign key (id_tip) references tipuri_locuinte(id_tip),
    constraint fk_locuinta2 foreign key (id_locatie) references locatie(id_locatie),
    constraint fk_locuinta3 foreign key (id_gazda) references gazda(id_gazda)
);
desc locuinta;

```

Script Output | Task completed in 0.61 seconds

Table LOCUINTA created.

Name	Null?	Type
ID_LOCUINTA	NOT NULL	NUMBER
ID_TIP		NUMBER
NUME_LOCUINTA		VARCHAR2(30)
ID_LOCATIE		NUMBER
CAPACITATE		NUMBER
ID_GAZDA		NUMBER

create table tarife(

```

id_tarif number constraint pk_tarif primary key,
pret_per_persoana number(5)

```

```

create table tarife(
    id_tarif number constraint pk_tarif primary key,
    pret_per_persoana number(5)
);
desc tarife;

```

```
create table utilizator(
```

Script Output | Task completed in 0.585 seconds

ID_LOCATIE	NUMBER
CAPACITATE	NUMBER
ID_GAZDA	NUMBER

Table TARIFE created.

Name	Null?	Type
ID_TARIF	NOT NULL	NUMBER
PRET_PER_PERSOANA		NUMBER(5)

);

create table utilizator(

```

id_utilizator number constraint pk_utilizator primary key,
nume varchar2(30),

```

```

prenume varchar2(30),

```

```

email varchar2(40),
data_inregistrare date,
telefon varchar(20),
gen varchar2(10),
data_nasterii date

```

The screenshot shows the SQL script being run in Oracle SQL Developer. The code creates a table named UTILIZATOR with columns: id\_utilizator (number, primary key), nume (varchar2(30)), prenume (varchar2(30)), email (varchar2(40)), data\_inregistrare (date), telefon (varchar(20)), gen (varchar2(10)), and data\_nasterii (date). After the creation command, a desc utility is used to describe the table. The output window shows the table was created successfully in 0.756 seconds.

```

create table utilizator(
    id_utilizator number constraint pk_utilizator primary key,
    nume varchar2(30),
    prenume varchar2(30),
    email varchar2(40),
    data_inregistrare date,
    telefon varchar(20),
    gen varchar2(10),
    data_nasterii date
);
desc utilizator;

```

Script Output X  
| Task completed in 0.756 seconds

Table UTILIZATOR created.

Name	Null?	Type
ID_UTILIZATOR	NOT NULL	NUMBER
NUME		VARCHAR2 (30)
PRENUME		VARCHAR2 (30)
EMAIL		VARCHAR2 (40)
DATA_INREGISTRARE		DATE
TELEFON		VARCHAR2 (20)
GEN		VARCHAR2 (10)
); DATA_NASTERII		DATE

```

create table rezervare(
    id_rezervare number constraint pk_rezervare primary key,
    id_utilizator number ,
    data_inceput date,
    data_sfarsit date,
    numar_persoane number,
    modalitate_plata varchar2(20),
    constraint fk_rezervare foreign key (id_utilizator) references utilizator(id_utilizator));

```

```

CREATE TABLE REZERVARE(
    ID_REZERVARE NUMBER CONSTRAINT PK_REZERVARE PRIMARY KEY,
    ID_UTILIZATOR NUMBER,
    ID_LOCUINTA NUMBER,
    ID_TARIF NUMBER,
    DATA_INCEPUT DATE,
    DATA_SFARSIT DATE,
    NUMAR_PERSOANE NUMBER,
    MODALITATE_PLATA VARCHAR2(20),
    CONSTRAINT FK_REZERVARE FOREIGN KEY (ID_UTILIZATOR) REFERENCES UTILIZATOR(ID_UTILIZATOR),
    CONSTRAINT FK_REZERVARE2 FOREIGN KEY (ID_LOCUINTA) REFERENCES LOCUINTA(ID_LOCUINTA),
    CONSTRAINT FK_REZERVARE3 FOREIGN KEY (ID_TARIF) REFERENCES TARIFE(ID_TARIF)
);

```

Table REZERVARE created.

Name	Null?	Type
ID_REZERVARE	NOT NULL	NUMBER
ID_UTILIZATOR		NUMBER
ID_LOCUINTA		NUMBER
ID_TARIF		NUMBER
DATA_INCEPUT		DATE
DATA_SFARSIT		DATE
NUMAR_PERSOANE		NUMBER
MODALITATE_PLATA		VARCHAR2(20)

```

create table se_asociaza_la(
    id_locuinta number references locuinta(id_locuinta),
    id_rezervare number references rezervare(id_rezervare),
    data_asociere date,
    status varchar2(10),
    constraint pk_se_asociaza_la primary key (id_locuinta,id_rezervare)
);

```

```

CREATE TABLE SE_ASOCIAZA_LA(
    ID_LOCUINTA NUMBER REFERENCES LOCUINTA(ID_LOCUINTA),
    ID_REZERVARE NUMBER REFERENCES REZERVARE(ID_REZERVARE),
    DATA_ASOCIERE DATE,
    CONSTRAINT PK_SE_ASOCIAZA_LA PRIMARY KEY (ID_LOCUINTA, ID_REZERVARE)
);

```

Table SE\_ASOCIAZA\_LA created.

Name	Null?	Type
ID_LOCUINTA	NOT NULL	NUMBER
ID_REZERVARE	NOT NULL	NUMBER
DATA_ASOCIERE		DATE

create table are(

```
id_dotare number references dotari(id_dotare),
id_locuinta number references locuinta(id_locuinta),
constraint pk_are primary key (id_locuinta,id_dotare)
```

The screenshot shows the SQL script being run in Oracle SQL Developer. The code creates the 'are' table with two foreign key constraints referencing 'dotari' and 'locuinta' tables, and a primary key constraint. It also creates the 'gazda' table with a primary key constraint. The 'Script Output' tab shows the execution results, including the table creation statements and the resulting columns and their types.

```
create table are(
    id_dotare number references dotari(id_dotare),
    id_locuinta number references locuinta(id_locuinta),
    constraint pk_are primary key (id_locuinta,id_dotare)
);
desc are;

create table gazda(
    id_gazda number constraint pk_gazda primary key,
    nume varchar2(30),
```

Script Output X | Task completed in 0.604 seconds

ID_RESERVARE	NOT NULL NUMBER
DATA_ASOCIERE	DATE

Table ARE created.

Name	Null?	Type
ID_DOTARE	NOT NULL NUMBER	
ID_LOCUINTA	NOT NULL NUMBER	

## Adăugare constrângeri pe tabele

ALTER TABLE gazda

```
ADD CONSTRAINT uq_email_g UNIQUE (email);
```

ALTER TABLE rezervare

```
ADD CONSTRAINT chk_date CHECK (data_inceput < data_sfarsit);
```

ALTER TABLE utilizator

```
ADD CONSTRAINT uq_email UNIQUE (email);
```

```
ALTER TABLE gazda
ADD CONSTRAINT uq_email_g UNIQUE (email);

ALTER TABLE rezervare
ADD CONSTRAINT chk_date CHECK (data_inceput < data_sfarsit);

ALTER TABLE utilizator
ADD CONSTRAINT uq_email UNIQUE (email);
```

alter table regiuni modify nume\_regiune not null;

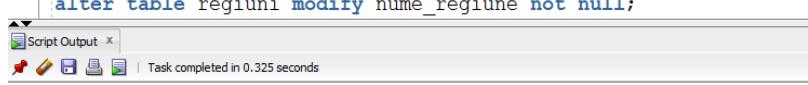


Table GAZDA altered.

Table REZERVARE altered.

Table UTILIZATOR altered.

alter table regiuni modify nume\_regiune not null;

alter table tari modify id\_regiune not null;

alter table locatie modify localitate not null;

alter table locatie modify id\_tara not null;

alter table locatie modify adresa not null;

alter table tipuri\_locuinte modify nume\_tip not null;

alter table locuinta modify id\_tip not null;

alter table locuinta modify nume\_locuinta not null;

alter table locuinta modify id\_locatie not null;

alter table locuinta modify capacitate not null;

alter table locuinta modify id\_gazda not null;

alter table dotari modify nume not null;

alter table gazda modify nume not null;

alter table gazda modify email not null;

alter table gazda modify data\_inceput\_activitate not null;

```

alter table rezervare modify id_utilizator not null;

alter table rezervare modify data_inceput not null;

alter table rezervare modify data_sfarsit not null;

alter table rezervare modify numar_persoane not null;

alter table rezervare modify modalitate_plata not null;

alter table tarife modify pret_per_persoana not null;

alter table utilizator modify nume not null;

alter table utilizator modify email not null;

alter table utilizator modify data_inregistrare not null;

alter table utilizator modify data_nasterii not null;

alter table se_asociaza_la modify data_asociere not null;

alter table se_asociaza_la modify status not null;

```

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

- Connections:** PROJECT\_SGBD is selected.
- Worksheet:** The Query Builder tab is active, displaying the ALTER TABLE script.
- Script Output:** Shows the execution results:
  - Table UTILIZATOR altered.
  - Table UTILIZATOR altered.
  - Table UTILIZATOR altered.
  - Table SE\_ASOCIAZA\_LA altered.
 The message "Script completed in 0.591 seconds" is also present.
- Status Bar:** Shows the date and time: 12/8/2024, 5:56 PM.

## 5. INSERĂRI ÎN TABELE

```
insert into regiuni values(1,'Europa de Est');  
insert into regiuni values(2,'Europa de Vest');  
insert into regiuni values(3,'Europa Centrala');  
insert into regiuni values(4,'Europa de Nord');  
insert into regiuni values(5,'Europa de Sud');  
insert into regiuni values(6,'America');
```

```
select * from regiuni;
```

The screenshot shows the MySQL Workbench interface. The top navigation bar includes tabs for 'sqldiagram', 'project\_id.sql', 'Lab5-16\_de\_descrevat.sql', 'Lab4 - SS\_de\_descrevat.sql', 'and\_e13\_acces.sql', 'Andra112.sql', 'bd4+reduz.sql', 'bd4+12-2024.sql', and 'PROJECT\_50801.sql'. The main area is titled 'Worksheet' and contains a 'Query Builder' tab. The query editor contains the following SQL code:

```
alter table utilizator modify data_inregistrare not null;  
alter table utilizator modify data_nasterii not null;  
alter table se_asociaza_la modify data_asociere not null;  
  
insert into regiuni values(1,'Europa de Est');  
insert into regiuni values(2,'Europa de Vest');  
insert into regiuni values(3,'Europa Centrala');  
insert into regiuni values(4,'Europa de Nord');  
insert into regiuni values(5,'Europa de Sud');  
insert into regiuni values(6,'America');  
  
select * from regiuni;  
  
insert into tari values('RO','Romania',1);
```

Below the query editor, the 'Results' tab is selected, showing the output of the last query:

ID_REGIUNE	NAME_REGIUNE
1	1.Europa de Est
2	2.Europa de Vest
3	3.Europa Centrala
4	4.Europa de Nord
5	5.Europa de Sud
6	6.America

```
insert into tari values('RO','Romania',1);  
insert into tari values('IT','Italia',5);  
insert into tari values('BE','Belgia',2);  
insert into tari values('CH','Elvetia',2);  
insert into tari values('DE','Germania',3);  
insert into tari values('ES','Spania',5);  
insert into tari values('FR','Franta',2);  
insert into tari values('NL','Olanda',2);
```

```

insert into tari values('UK','Regatul Unit',2);
insert into tari values('US','Statele Unite ale Americii',6);
insert into tari values('NO','Norvegia',4);
insert into tari values('GR','Grecia',5);

```

```
select * from tari;
```

The screenshot shows the SQL Server Management Studio interface. In the top window (Query Builder), there is a large block of SQL code for inserting data into the 'tari' table. In the bottom window (Script Output), the results of the query are displayed in a table format.

ID_TARA	NUME_TARA	ID_REGIUNE	
1	RO	Romania	1
2	IT	Italia	5
3	BE	Belgia	2
4	CH	Elvetia	2
5	DE	Germania	3
6	ES	Spania	5
7	FR	Franta	2
8	NL	Olanda	2
9	UK	Regatul Unit	2
10	US	Statele Unite ale Americii	6
11	NO	Norvegia	4
12	GR	Grecia	5

```

insert into locatie values(150,'Bulevardul Nicolae Balcescu 4','010011','Bucuresti','RO');--hotel
insert into locatie values(151,'Lazio 111','00100','Roma','IT');--ap
insert into locatie values(152,'Ixelles 2A','1004','Bruxelles','BE');--ap
insert into locatie values(153,'Ostermundigen 18C','03564','Berna','CH');--ap
insert into locatie values(154,'Lützowplatz 17, Mitte','89004','Berlin','DE');--hotel
insert into locatie values(155,'Calle de San Pedro Mártir 9','28001','Madrid','ES');--vila
insert into locatie values(156,'7 Arondismentul 19','75001','Paris','FR');--vila
insert into locatie values(157,'Keizersgracht 215','1011','Amsterdam','NL');--ap
insert into locatie values(158,'52-54 Upper Richmond Road, Wandsworth','SW1A 1AA','Londra','UK');--hotel
insert into locatie values(159,'East Village 4D','10001','New York','US');--ap

```

```
insert into locatie values(160,'Nesodden, Viken 15','0010','Oslo','NO');--cabana
```

```
insert into locatie values(161,'16 Voulis','10557','Atena','GR');--hotel
```

```
insert into locatie values(163,'Hoia Baciu 185','98054','Cluj','RO');--cabana
```

```
insert into locatie values(162,'Strada Tineretului 134','22224','Costinesti','RO');--pensiune
```

```
insert into locatie values(164,'Strada Reconstruc?iei 2','02468','Sibiu','RO');--pensiune
```

```
select * from locatie;
```

The screenshot shows the Oracle SQL Developer interface. At the top, there are several tabs: sql.sql, project.bdd.sql, Lab5-56\_de\_descarcat.sql, Lab4-55\_de\_descarcat.sql, andr132\_acces.sql, Andr132.sql, lab4-relat.sql, lab4-12-2024.sql, PROJECT\_SORDI.sql, and Welcome Page. Below the tabs is the 'Worksheet' section with a 'Query Builder' button. The query builder contains the following SQL code:

```
insert into locatie values(152,'Ixelles 2A','1004','Bruxelles','BE');--ap
insert into locatie values(153,'Ostermundigen 18C','03564','Berna','CH');--ap
insert into locatie values(154,'Lützowplatz 17, Mitte','89004','Berlin','DE');--hotel
insert into locatie values(155,'Calle de San Pedro Martir 9','28001','Madrid','ES');--ap
insert into locatie values(156,'7 Arondismentul 19','75001','Paris','FR');--vila
insert into locatie values(157,'Keizersgracht 215','1011','Amsterdam','NL');--ap
insert into locatie values(158,'52-54 Upper Richmond Road, Wandsworth','SW1A 1AA','Londra','UK');--hotel
insert into locatie values(159,'East Village 4D','10001','New York','US');--ap
insert into locatie values(160,'Nesodden, Viken 15','0010','Oslo','NO');--cabana
insert into locatie values(161,'16 Voulis','10557','Atena','GR');--hotel
insert into locatie values(163,'Hoia Baciu 185','98054','Cluj','RO');--cabana
insert into locatie values(162,'Strada Tineretului 134','22224','Costinesti','RO');--pensiune
insert into locatie values(164,'Strada Reconstruc?iei 2','02468','Sibiu','RO');--pensiune
select * from locatie;
```

Below the query builder is the 'Script Output' tab, which shows the results of the query. The results are displayed in a table with columns: ID\_LOCATE, ADRESA, COD\_POSTAL, LOCALITATE, and ID\_TARA. The data is as follows:

ID_LOCATE	ADRESA	COD_POSTAL	LOCALITATE	ID_TARA
1	150 Bulevardul Nicolae Balcescu 4	001001	Bucuresti	RO
2	151 Lazio 111	00100	Roma	IT
3	152 Ixelles 2A	1004	Bruxelles	BE
4	153 Ostermundigen 18C	03564	Berna	CH
5	154 Lützowplatz 17, Mitte	89004	Berlin	DE
6	155 Calle de San Pedro Martir 9	28001	Madrid	ES
7	156 7 Arondismentul 19	75001	Paris	FR
8	157 Keizersgracht 215	1011	Amsterdam	NL
9	158 52-54 Upper Richmond Road, Wandsworth	SW1A 1AA	Londra	UK
10	159 East Village 4D	10001	New York	US
11	160 Nesodden, Viken 15	0010	Oslo	NO
12	161 16 Voulis	10557	Atena	GR
13	163 Hoia Baciu 185	98054	Cluj	RO
14	162 Strada Tineretului 134	22224	Costinesti	RO
15	164 Strada Reconstruc?iei 2	02468	Sibiu	RO

```
insert into tipuri_locuinte values(10,'apartament single',600);
```

```
insert into tipuri_locuinte values(20,'apartament double',601);
```

```
insert into tipuri_locuinte values(30,'apartament triple',602);
```

```
insert into tipuri_locuinte values(40,'hotel 3 stele',603);
```

```
insert into tipuri_locuinte values(50,'hotel 4 stele',604);
```

```
insert into tipuri_locuinte values(60,'hotel 5 stele',605);
```

```
insert into tipuri_locuinte values(70,'cabana',606);
```

```
insert into tipuri_locuinte values(80,'pensiune',607);
```

```
insert into tipuri_locuinte values(90,'vila',608);
```

```
select * from tipuri_locuinte;
```

```
235:
236:
237: insert into tipuri_locuinte values(10,'apartament single',600);
238: insert into tipuri_locuinte values(20,'apartament double',601);
239: insert into tipuri_locuinte values(30,'apartament triple',602);
240: insert into tipuri_locuinte values(40,'hotel 3 stele',603);
241: insert into tipuri_locuinte values(50,'hotel 4 stele',604);
242: insert into tipuri_locuinte values(60,'hotel 5 stele',605);
243: insert into tipuri_locuinte values(70,'cabana',606);
244: insert into tipuri_locuinte values(80,'pensiune',607);
245: insert into tipuri_locuinte values(90,'vila',608);
246:
247: select * from tipuri_locuinte;
248:
249: insert into locuenta values(seq_locuenta.nextval,20,'Grand Hotel',150,1000,10);--hotel
```

The screenshot shows the Oracle SQL Developer interface. At the top, there is a code editor window containing the SQL script. Below it is a 'Script Output' tab showing the execution results, which include the inserted data into the 'tipuri\_locuinte' table and the creation of a new row in the 'locuenta' table. The bottom part of the screenshot shows the 'Query Result' tab displaying the contents of the 'tipuri\_locuinte' table.

ID_TIP	NUME_TIP	ID_TARIF
1	10 apartament single	600
2	20 apartament double	601
3	30 apartament triple	602
4	40 hotel 3 stele	603
5	50 hotel 4 stele	604
6	60 hotel 5 stele	605
7	70 cabana	606
8	80 pensiune	607
9	90 vila	608

```
insert into dotari values (200,'wifi',null);
```

```
insert into dotari values (201,'parcare',10);
```

```
insert into dotari values (202,'aer conditionat',null);
```

```
insert into dotari values (203,'gratar',null);
```

```
insert into dotari values (204,'pat copil',75);
```

```
insert into dotari values (205,'incalzire',null);
```

```
insert into dotari values (206,'piscina',40);
```

```
insert into dotari values (207,'semineu',null);
```

```
select * from dotari;
```

```

insert into locuinta values(seq_locuinta.nextval,40,'Buburuza',164,20,15);--pensiune

select * from locuinta;

insert into dotari values (200,'wifi',null);
insert into dotari values (201,'parcare',10);
insert into dotari values (202,'aer conditionat',null);
insert into dotari values (203,'gratar',null);
insert into dotari values (204,'pat copil',75);
insert into dotari values (205,'incalzire',null);
insert into dotari values (206,'piscina',40);
insert into dotari values (207,'semineu',null);
select * from dotari;

insert into are values(200,200);

```

Script Output | Explain Plan | Query Result | All Rows Fetched: 8 in 0.004 seconds

ID_DOTARI	NAME	COST_SUPRIMENTAR
1	200 wifi	(null)
2	201 parcare	10
3	202 aer conditionat	(null)
4	203 gratar	(null)
5	204 pat copil	75
6	205 incalzire	(null)
7	206 piscina	40
8	207 semineu	(null)

insert into gazda

values(seq\_gazda.nextval,'Brown','James','b\_js@yahoo.com','0779478514',to\_date('16/02/2016','dd/mm/yyyy'));

insert into gazda

values(seq\_gazda.nextval,'Ionescu','Mihaela','m\_I@gmail.com',null,to\_date('21/01/2007','dd/mm/yyyy'));

insert into gazda

values(seq\_gazda.nextval,'Morris','Steve','m\_s@yahoo.com',null,to\_date('21/06/1999','dd/mm/yy'));

insert into gazda

values(seq\_gazda.nextval,'Militaru','Bianca','b\_aru@yahoo.com','0778140009',to\_date('12/06/2009','dd/mm/yyyy'));

insert into gazda

values(seq\_gazda.nextval,'Rojo','Angela','A\_ELA@yahoo.com',null,to\_date('06/08/2003','dd/mm/yyyy'));

insert into gazda

values(seq\_gazda.nextval,'Vale','Estibaliz','e\_val@gmail.com',null,to\_date('03/10/2008','dd/mm/yyyy'));

insert into gazda

values(seq\_gazda.nextval,'King','John','kingj@gmail.com','478514',to\_date('16/07/2007','dd/mm/yyyy'));

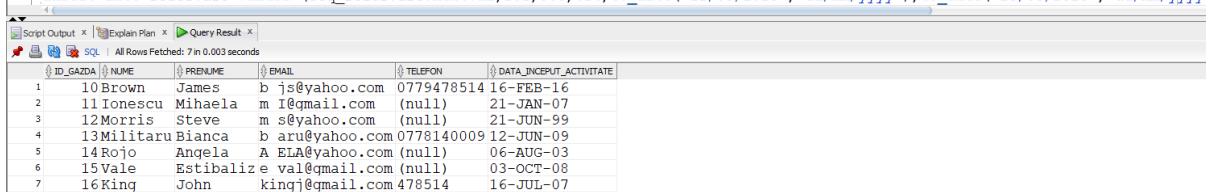
select \* from gazda;

```

insert into gazda values(seq_gazda.nextval,'Brown','James','b_js@yahoo.com','0779478514',to_date('16/02/2016','dd/mm/yyyy'));
insert into gazda values(seq_gazda.nextval,'Ionescu','Mihaela','m_I@gmail.com',null,to_date('21/01/2007','dd/mm/yyyy'));
insert into gazda values(seq_gazda.nextval,'Morris','Steve','m_s@yahoo.com',null,to_date('21/06/1999','dd/mm/yyyy'));
insert into gazda values(seq_gazda.nextval,'Militaru','Bianca','b_aru@yahoo.com','0778140009',to_date('12/06/2009','dd/mm/yyyy'));
insert into gazda values(seq_gazda.nextval,'Rojo','Angela','A_ELA@yahoo.com',null,to_date('06/08/2003','dd/mm/yyyy'));
insert into gazda values(seq_gazda.nextval,'Vale','Estibaliz','e_val@gmail.com',null,to_date('03/10/2008','dd/mm/yyyy'));
insert into gazda values(seq_gazda.nextval,'King','John','kingj@gmail.com','478514',to_date('16/07/2007','dd/mm/yyyy'));
select * from gazda;

insert into rezervare values (seq_rezervare.nextval,100,250,605,to_date('20/07/2024','dd/mm/yyyy'),to_date('26/07/2024','dd/mm/yyyy'));
insert into rezervare values (seq_rezervare.nextval,101,340,614,to_date('22/03/2023','dd/mm/yyyy'),to_date('25/03/2023','dd/mm/yyyy'))

```

Script Output | Explain Plan | Query Result |  


ID_GAZDA	NUME	PRENUME	EMAIL	TELEFON	DATA_INCEPUT_ACTIVITATE
1	10 Brown	James	b_js@yahoo.com	0779478514	16-FEB-16
2	11 Ionescu	Mihaela	m_I@gmail.com	(null)	21-JAN-07
3	12 Morris	Steve	m_s@yahoo.com	(null)	21-JUN-99
4	13 Militaru	Bianca	b_aru@yahoo.com	0778140009	12-JUN-09
5	14 Rojo	Angela	A_ELA@yahoo.com	(null)	06-AUG-03
6	15 Vale	Estibalize	e_val@gmail.com	(null)	03-OCT-08
7	16 King	John	kingj@gmail.com	478514	16-JUL-07

```

insert into locuinta values(seq_locuinta.nextval,60,'Grand Hotel',150,1000,10);--hotel 5

insert into locuinta values(seq_locuinta.nextval,30,'Lazio Bella ',151,5,11);--ap 3

insert into locuinta values(seq_locuinta.nextval,20,'Pink promise',152,4,12);--ap 2

insert into locuinta values(seq_locuinta.nextval,30,'Ap-yellow',153,6,13);--ap 3

insert into locuinta values(seq_locuinta.nextval,40,'Magic Column',154,250,10);--hotel 3

insert into locuinta values(seq_locuinta.nextval,90,'Vila Trevi',155,10,10);--vila

insert into locuinta values(seq_locuinta.nextval,90,'Vila Bleu',156,15,12);--vila

insert into locuinta values(seq_locuinta.nextval,10,'Majestic',157,2,14);--ap 1

insert into locuinta values(seq_locuinta.nextval,50,'Hotel Richmond',158,300,10);--hotel 4

insert into locuinta values(seq_locuinta.nextval,20,'EV stay',159,4,14);--ap 2

insert into locuinta values(seq_locuinta.nextval,70,'Nesodden house',160,8,12);--cabana

insert into locuinta values(seq_locuinta.nextval,50,'Hotel Areon',161,500,10);--hotel 4

insert into locuinta values(seq_locuinta.nextval,70,'Cabana Baciu',163,10,16);--cabana

insert into locuinta values(seq_locuinta.nextval,80,'Casa Alex',162,12,15);--pensiune

insert into locuinta values(seq_locuinta.nextval,80,'Buburuza',164,20,15);--pensiune

select * from locuinta;

```

```

247 select * from tipuri_locuinte;
248
249 insert into locuinta values(seq_locuinta.nextval,60,'Grand Hotel',150,1000,10);--hotel 5
250 insert into locuinta values(seq_locuinta.nextval,30,'Lazio Bella ',151,5,11);--ap 3
251 insert into locuinta values(seq_locuinta.nextval,20,'Pink promise',152,4,12);--ap 2
252 insert into locuinta values(seq_locuinta.nextval,30,'Ap-yellow',153,6,13);--ap 3
253 insert into locuinta values(seq_locuinta.nextval,40,'Magic column',154,250,10);--hotel 3
254 insert into locuinta values(seq_locuinta.nextval,90,'Vila Trevi',155,10,10);--vila
255 insert into locuinta values(seq_locuinta.nextval,90,'Vila Bleu',156,15,12);--vila
256 insert into locuinta values(seq_locuinta.nextval,10,'Majestic',157,2,14);--ap 1
257 insert into locuinta values(seq_locuinta.nextval,50,'Hotel Richmond',158,300,10);--hotel 4
258 insert into locuinta values(seq_locuinta.nextval,20,'EV stay',159,4,14);--ap 2
259 insert into locuinta values(seq_locuinta.nextval,70,'Nesodden house',160,8,10);--cabana
260 insert into locuinta values(seq_locuinta.nextval,50,'Hotel Areon',161,500,10);--hotel 4
261 insert into locuinta values(seq_locuinta.nextval,70,'Cabana Baciu',163,10,16);--cabana
262 insert into locuinta values(seq_locuinta.nextval,80,'Casa Alex',162,12,15);--pensiune
263 insert into locuinta values(seq_locuinta.nextval,80,'Buburuza',164,20,15);--pensiune
264

```

ID_LOCUNTA	ID_TIP	NAME_LOCUNTA	ID_LOCU	CAPACITATE	ID_GARANTIE
1	EV STAY	EV STAY	155	10	
2	VILA TREV	VILA TREV	156	10	
3	MAGIC COL	MAGIC COL	157	12	
4	VILA BLEU	VILA BLEU	158	15	
5	GRAND HOTEL	GRAND HOTEL	159	1000	10
6	LAZIO BELLA	LAZIO BELLA	160	5	11
7	AP-YELLOW	AP-YELLOW	161	300	10
8	PINK PROMISE	PINK PROMISE	162	12	15
9	MAJESTIC	MAJESTIC	163	10	16
10	WATER PARK	WATER PARK	164	20	15
11	NESODDEN HOUSE	NESODDEN HOUSE	165	8	12
12	50 HOTEL AREON	50 HOTEL AREON	166	500	10
13	CABANA BACIU	CABANA BACIU	167	10	16
14	CASA ALEX	CASA ALEX	168	12	15
15	BUBURUZA	BUBURUZA	169	20	15

insert into are values(200,200);

insert into are values(200,210);

insert into are values(200,220);

insert into are values(200,230);

insert into are values(200,240);

insert into are values(200,250);

insert into are values(200,260);

insert into are values(200,270);

insert into are values(200,280);

insert into are values(200,290);

insert into are values(200,310);

insert into are values(200,330);

insert into are values(200,340);

insert into are values(201,280);

```

insert into are values(201,200);
insert into are values(201,310);
insert into are values(202,200);
insert into are values(202,210);
insert into are values(202,220);
insert into are values(203,330);
insert into are values(203,340);
insert into are values(206,250);
insert into are values(206,260);
insert into are values(204,240);
insert into are values(205,300);
insert into are values(205,320);
insert into are values(207,270);

```

```
select * from are;
```

The screenshot shows the Oracle SQL Developer interface. The top navigation bar includes tabs for 'SQL', 'Worksheet', 'History', and 'Welcome Page'. Below the tabs, there are icons for 'Script', 'Output', 'SQL', 'Explain Plan', and 'Query Result'. The 'Worksheet' tab is active, displaying the following SQL code:

```

insert into are values(201,310);
insert into are values(202,200);
insert into are values(202,210);
insert into are values(202,220);
insert into are values(203,330);
insert into are values(203,340);
insert into are values(206,250);
insert into are values(206,260);
insert into are values(204,240);
insert into are values(205,300);
insert into are values(205,320);
insert into are values(207,270);
select * from are;

```

The 'Query Result' tab below shows the output of the query:

ID_DORMATE	ID_LOJOMANTA	
1	200	200
2	200	210
3	200	220
4	200	230
5	200	240
6	200	250
7	200	260
8	200	270
9	200	280
10	200	290
11	200	310
12	200	330
13	200	340
14	201	280
15	201	200
16	201	310

```
insert into tarife values(seq_tarife.nextval,100);
```

```

insert into tarife values(seq_tarife.nextval,140);
insert into tarife values(seq_tarife.nextval,250);
insert into tarife values(seq_tarife.nextval,200);
insert into tarife values(seq_tarife.nextval,270);
insert into tarife values(seq_tarife.nextval,389);
insert into tarife values(seq_tarife.nextval,99);
insert into tarife values(seq_tarife.nextval,70);
insert into tarife values(seq_tarife.nextval,85);
select * from tarife;

```

The screenshot shows the Oracle SQL Developer interface. On the left, a code editor displays a script with lines numbered 331 to 342. Lines 331 through 340 contain the `insert into tarife values` statements, and line 341 contains the `select * from tarife;` statement. On the right, the 'Query Result' tab is open, showing a table with two columns: ID\_TARIF and PRET\_PER\_PERSOANA. The table has 9 rows, with data matching the insert statements in the script.

ID_TARIF	PRET_PER_PERSOANA
1	600
2	601
3	602
4	603
5	604
6	605
7	606
8	607
9	608

---

```

insert into utilizator
values(seq_utilizator.nextval,'Andruta','Andra','andra.andruta60@yahoo.com',to_date('28/04/202
3','dd/mm/yyyy'),'0722313159','feminin',to_date('26/02/2004','dd/mm/yyyy'));

```

```

insert into utilizator
values(seq_utilizator.nextval,'Andruta','Irina','irina.andruta@gmail.com',to_date('02/12/2013','dd/
mm/yyyy'),'0722508756',null,to_date('21/01/1977','dd/mm/yyyy'));

```

```

insert into utilizator
values(seq_utilizator.nextval,'Andruta','Costi','costi.andruta@yahoo.com',to_date('11/11/2011','d
d/mm/yyyy'),null,'masculin',to_date('21/01/1978','dd/mm/yyyy'));

```

```

insert into utilizator
values(seq_utilizator.nextval,'Andruta','Maia','mandruta24@yahoo.com',to_date('10/05/2024','dd/mm/yyyy'),null,'feminin',to_date('24/06/2005','dd/mm/yyyy'));

insert into utilizator
values(seq_utilizator.nextval,'Rosu','Ioana','Rioana49@yahoo.com',to_date('17/04/2015','dd/mm/yyyy'),'0778145699',null,to_date('06/08/1947','dd/mm/yyyy'));

insert into utilizator values(seq_utilizator.nextval,'Vasile','Mona','m-vasile@gmail.com',to_date('17/04/2015','dd/mm/yyyy'),null,null,to_date('14/03/1998','dd/mm/yy'));

insert into utilizator
values(seq_utilizator.nextval,'Miinescu','Sorin','smorin@yahoo.com',to_date('10/10/2018','dd/mm/yyyy'),null,'masculin',to_date('08/04/2004','dd/mm/yyyy'));

insert into utilizator
values(seq_utilizator.nextval,'Popescu','Ion','ion_pope2016@yahoo.com',to_date('21/09/2016','dd/mm/yyyy'),null,'masculin',to_date('04/05/1996','dd/mm/yyyy'));

insert into utilizator
values(seq_utilizator.nextval,'Firica','Alexandru','alex.firica2005@gmail.com',to_date('21/02/2022','dd/mm/yyyy'),null,'masculin',to_date('09/11/2005','dd/mm/yyyy'));

```

select \* from utilizator;

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

- Script Output:** Contains the SQL code for inserting 10 rows into the 'utilizator' table.
- Explain Plan:** Shows the execution plan for the query.
- Query Result:** Displays the 10 rows inserted, with columns: ID\_UTILIZATOR, NUME, PRENUME, EMAIL, DATA\_INREGISTRARE, TELEFON, GEN, and DATA\_NASTERE.

ID_UTILIZATOR	NUME	PRENUME	EMAIL	DATA_INREGISTRARE	TELEFON	GEN	DATA_NASTERE	
1	100	Andruta	Andra	andra.andruta60@yahoo.com	28-APR-23	0722313159	feminin	26-FEB-04
2	101	Andruta	Irina	irina.andruta@gmail.com	02-DEC-13	0722508750	(null)	21-JAN-77
3	102	Andruta	Costi	costi.andruta@yahoo.com	11-NOV-11	(null)	masculin	21-JAN-78
4	103	Andruta	Maia	mandruta24@yahoo.com	10-MAY-24	(null)	feminin	24-JUN-05
5	104	Rosu	Ioana	Rioana49@yahoo.com	17-APR-15	0778145699	(null)	06-AUG-47
6	105	Vasile	Mona	m-vasile@gmail.com	17-APR-15	(null)	(null)	14-MAR-98
7	106	Miinescu	Sorin	smorin@yahoo.com	10-OCT-18	(null)	masculin	08-APR-04
8	107	Popescu	Ion	ion_pope2016@yahoo.com	21-SEP-16	(null)	masculin	04-MAY-96
9	108	Firica	Alexandru	alex.firica2005@gmail.com	21-FEB-22	(null)	masculin	09-NOV-05

```
insert into rezervare values
(seq_rezervare.nextval,100,to_date('20/07/2024','dd/mm/yyyy'),to_date('26/07/2024','dd/mm/yyyy'),2,'cash');

insert into rezervare values
(seq_rezervare.nextval,101,to_date('22/03/2023','dd/mm/yyyy'),to_date('25/03/2023','dd/mm/yyyy'),15,'card');

insert into rezervare values
(seq_rezervare.nextval,102,to_date('02/04/2025','dd/mm/yyyy'),to_date('03/04/2025','dd/mm/yyyy'),3,'transfer');

insert into rezervare values
(seq_rezervare.nextval,104,to_date('14/02/2019','dd/mm/yyyy'),to_date('16/02/2019','dd/mm/yyyy'),5,'card');

insert into rezervare values
(seq_rezervare.nextval,105,to_date('19/11/2019','dd/mm/yyyy'),to_date('29/11/2019','dd/mm/yyyy'),3,'cash');

insert into rezervare values
(seq_rezervare.nextval,106,to_date('08/10/2024','dd/mm/yyyy'),to_date('14/10/2024','dd/mm/yyyy'),8,'transfer');

insert into rezervare values
(seq_rezervare.nextval,107,to_date('20/12/2024','dd/mm/yyyy'),to_date('03/01/2025','dd/mm/yyyy'),4,'card');

insert into rezervare values
(seq_rezervare.nextval,103,to_date('20/01/2022','dd/mm/yyyy'),to_date('27/01/2022','dd/mm/yyyy'),2,'cash');

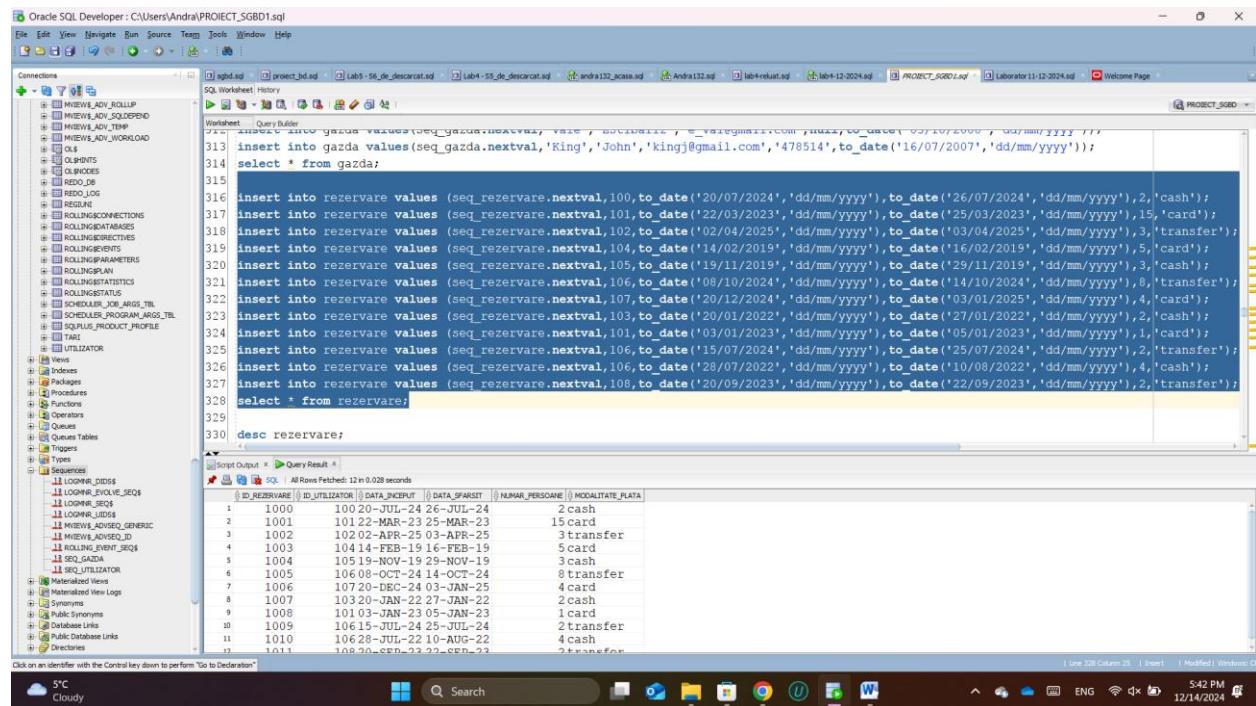
insert into rezervare values
(seq_rezervare.nextval,101,to_date('03/01/2023','dd/mm/yyyy'),to_date('05/01/2023','dd/mm/yyyy'),1,'card');

insert into rezervare values
(seq_rezervare.nextval,106,to_date('15/07/2024','dd/mm/yyyy'),to_date('25/07/2024','dd/mm/yyyy'),2,'transfer');

insert into rezervare values
(seq_rezervare.nextval,106,to_date('28/07/2022','dd/mm/yyyy'),to_date('10/08/2022','dd/mm/yyyy'),4,'cash');
```

insert into rezervare values  
 (seq\_rezervare.nextval,108,to\_date('20/09/2023','dd/mm/yyyy'),to\_date('22/09/2023','dd/mm/yyyy'),2,'transfer');

select \* from rezervare;



```

313 insert into gazda values (seq_gazda.nextval, 'Valie', 'Sorin', 's_valie@gmail.com', null, to_date('09/10/2000', 'dd/mm/yyyy'));
314 select * from gazda;
315
316 insert into rezervare values (seq_rezervare.nextval, 100, to_date('20/07/2024', 'dd/mm/yyyy'), to_date('26/07/2024', 'dd/mm/yyyy'), 2, 'cash');
317 insert into rezervare values (seq_rezervare.nextval, 101, to_date('22/03/2023', 'dd/mm/yyyy'), to_date('25/03/2023', 'dd/mm/yyyy'), 15, 'card');
318 insert into rezervare values (seq_rezervare.nextval, 102, to_date('02/04/2025', 'dd/mm/yyyy'), to_date('03/04/2025', 'dd/mm/yyyy'), 3, 'transfer');
319 insert into rezervare values (seq_rezervare.nextval, 104, to_date('14/02/2019', 'dd/mm/yyyy'), to_date('16/02/2019', 'dd/mm/yyyy'), 5, 'card');
320 insert into rezervare values (seq_rezervare.nextval, 105, to_date('19/11/2019', 'dd/mm/yyyy'), to_date('29/11/2019', 'dd/mm/yyyy'), 3, 'cash');
321 insert into rezervare values (seq_rezervare.nextval, 106, to_date('08/10/2024', 'dd/mm/yyyy'), to_date('14/10/2024', 'dd/mm/yyyy'), 8, 'transfer');
322 insert into rezervare values (seq_rezervare.nextval, 107, to_date('20/12/2024', 'dd/mm/yyyy'), to_date('03/01/2025', 'dd/mm/yyyy'), 4, 'card');
323 insert into rezervare values (seq_rezervare.nextval, 103, to_date('20/01/2022', 'dd/mm/yyyy'), to_date('27/01/2022', 'dd/mm/yyyy'), 2, 'cash');
324 insert into rezervare values (seq_rezervare.nextval, 101, to_date('03/01/2023', 'dd/mm/yyyy'), to_date('05/01/2023', 'dd/mm/yyyy'), 1, 'card');
325 insert into rezervare values (seq_rezervare.nextval, 106, to_date('15/07/2024', 'dd/mm/yyyy'), to_date('25/07/2024', 'dd/mm/yyyy'), 2, 'transfer');
326 insert into rezervare values (seq_rezervare.nextval, 106, to_date('28/07/2022', 'dd/mm/yyyy'), to_date('10/08/2022', 'dd/mm/yyyy'), 4, 'cash');
327 insert into rezervare values (seq_rezervare.nextval, 108, to_date('20/09/2023', 'dd/mm/yyyy'), to_date('22/09/2023', 'dd/mm/yyyy'), 2, 'transfer');
328 select * from rezervare;
329
330 desc rezervare;

```

ID_NUMERAR	ID_UTILIZATOR	DATA_ZAVERU	DATA_SFARSIT	NUMAR_PERSONE	_MODALITATE_PLATA
1	1000	100 20-JUL-24 26-JUL-24		2 cash	
2	1001	101 22-MAR-23 25-MAR-23		15 card	
3	1002	102 02-APR-25 03-APR-25		3 transfer	
4	1003	104 14-FEB-19 16-FEB-19		5 card	
5	1004	105 19-NOV-19 29-NOV-19		3 cash	
6	1005	106 08-OCT-24 14-OCT-24		8 transfer	
7	1006	107 20-DEC-24 03-JAN-25		4 card	
8	1007	103 20-JAN-22 27-JAN-22		2 cash	
9	1008	101 03-JAN-23 05-JAN-23		1 card	
10	1009	106 15-JUL-24 25-JUL-24		2 transfer	
11	1010	106 28-JUL-22 10-AUG-22		4 cash	
12	1011	108 20-GEN-23 22-GEN-23		2 transfer	

```

insert into se_asociaza_la values(250,1000,to_date('20/04/2024','dd/mm/yyyy'),'activa');

insert into se_asociaza_la values(340,1001,to_date('10/02/2023','dd/mm/yyyy'),'activa');

insert into se_asociaza_la values(200,1002,to_date('19/11/2024','dd/mm/yyyy'),'activa');

insert into se_asociaza_la values(210,1003,to_date('03/01/2019','dd/mm/yyyy'),'activa');

insert into se_asociaza_la values(220,1004,to_date('18/11/2019','dd/mm/yyyy'),'activa');

insert into se_asociaza_la values(300,1005,to_date('06/06/2024','dd/mm/yyyy'),'activa');

insert into se_asociaza_la values(200,1006,to_date('28/12/2023','dd/mm/yyyy'),'activa');

insert into se_asociaza_la values(240,1007,to_date('05/01/2022','dd/mm/yyyy'),'activa');

insert into se_asociaza_la values(230,1008,to_date('26/02/2022','dd/mm/yyyy'),'activa');

insert into se_asociaza_la values(270,1009,to_date('12/10/2023','dd/mm/yyyy'),'activa');

insert into se_asociaza_la values(290,1010,to_date('03/12/2021','dd/mm/yyyy'),'activa');

```

```
insert into se_asociaza_la values(250,1011,to_date('09/05/2023','dd/mm/yyyy'),'activa');
```

```
select * from se_asociaza_la;
```

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

```
156
157
158 insert into se_asociaza_la values(250,1000,to_date('20/04/2024','dd/mm/yyyy'),'activa');
159 insert into se_asociaza_la values(340,1001,to_date('10/02/2023','dd/mm/yyyy'),'activa');
160 insert into se_asociaza_la values(200,1002,to_date('19/11/2024','dd/mm/yyyy'),'activa');
161 insert into se_asociaza_la values(210,1003,to_date('03/01/2019','dd/mm/yyyy'),'activa');
162 insert into se_asociaza_la values(220,1004,to_date('18/11/2019','dd/mm/yyyy'),'activa');
163 insert into se_asociaza_la values(300,1005,to_date('06/06/2024','dd/mm/yyyy'),'activa');
164 insert into se_asociaza_la values(200,1006,to_date('28/12/2023','dd/mm/yyyy'),'activa');
165 insert into se_asociaza_la values(240,1007,to_date('05/01/2022','dd/mm/yyyy'),'activa');
166 insert into se_asociaza_la values(230,1008,to_date('26/02/2022','dd/mm/yyyy'),'activa');
167 insert into se_asociaza_la values(270,1009,to_date('12/10/2023','dd/mm/yyyy'),'activa');
168 insert into se_asociaza_la values(290,1010,to_date('03/12/2021','dd/mm/yyyy'),'activa');
169 insert into se_asociaza_la values(250,1011,to_date('09/05/2023','dd/mm/yyyy'),'activa');
170
171 select * from se_asociaza_la;
172
173 commit;
```

Below the code, there is a "Query Result" tab showing the output of the "select \* from se\_asociaza\_la;" command. The result is a table with four columns: ID\_LOCUENTA, ID\_RESERVAR, DATA\_ASOCIERE, and STATUS. The data is as follows:

	ID_LOCUENTA	ID_RESERVAR	DATA_ASOCIERE	STATUS
1	250	1000	20-APR-24	activa
2	340	1001	10-FEB-23	activa
3	200	1002	19-NOV-24	activa
4	210	1003	03-JAN-19	activa
5	220	1004	18-NOV-19	activa
6	300	1005	06-JUN-24	activa
7	200	1006	28-DEC-23	activa
8	240	1007	05-JAN-22	activa
9	230	1008	26-FEB-22	activa
10	270	1009	12-OCT-23	activa
11	290	1010	03-DEC-21	activa
12	250	1011	09-MAY-23	activa

```
commit;
```

## 6. EXERCITIUL 6 – SUBPROGRAM STOCAT INDEPENDENT

ENUNT:

Să se creeze un subprogram stocat, numit loc\_disponibile, care să primească ca parametri o regiune (p\_regiune) și o perioadă de timp (perioada\_start și perioada\_stop). Procedura trebuie să afișeze locuințele din regiunea specificată și să verifice disponibilitatea lor, după cum urmează:

- Dacă locuința este disponibilă în perioada transmisă se afișează numele locuinței și lista dotărilor acesteia.
- Dacă perioada se suprapune cu o rezervare activă, să va afișa un mesaj corespunzător.

In vederea rezolvării să se utilizeze cele 3 tipuri de colecții studiate.

--lista locuinte => tablou indexat

--istoricul de rezervari ale locuintei => tabele nested care retine perioada

--dotarile => varray

create or replace procedure loc\_disponibile(

    p\_regiune IN regiuni.nume\_regiune%type,

    perioada\_start IN date,

    perioada\_stop IN date) is

    type tablou\_index\_loc is table of locuinta.nume\_locuinta%type index by pls\_integer;

    lista\_locuinte tablou\_index\_loc;

    type tablou\_nested\_rezervari is table of varchar2(50);

    istoric\_rezervari\_start tablou\_nested\_rezervari;

    istoric\_rezervari\_sfarsit tablou\_nested\_rezervari;

    type dot is varray(26) of dotari.nume%type;

```

lista_dotari dot;

ok boolean;
nr number :=0;

begin
--locuintele din regiunea transmisa ca parametru
for rec in (select l.id_locuinta, l.nume_locuinta
             from locuinta l
             join locatie loc on l.id_locatie = loc.id_locatie
             join tari t on loc.id_tara = t.id_tara
             join regiuni reg on reg.id_regiune = t.id_regiune
             where reg.nume_regiune = p_regiune) loop
    lista_locuinte(rec.id_locuinta) := rec.nume_locuinta;
end loop;

dbms_output.put_line('Locuintele gasite in regiunea ''|| p_regiune ||': ');
dbms_output.put_line(");

nr := 0;
for i in lista_locuinte.first..lista_locuinte.last loop
    if lista_locuinte.exists(i) then
        nr := nr+1;
        dbms_output.put_line(nr ||'. '||lista_locuinte(i));
    end if;
end loop;

istoric_rezervari_start := tablou_nested_rezervari();

```

```

istoric_rezervari_sfarsit := tablou_nested_rezervari();
--pt fiecare locuinta gasita perioadele in care e rezervata
for aux in(
    select r.data_inceput,r.data_sfarsit
    from rezervare r
    join se_asociaza_la aux on aux.id_rezervare = r.id_rezervare
    where aux.id_locuinta = i
    and aux.status = 'activa' ) loop
    istoric_rezervari_start.extend;
    istoric_rezervari_sfarsit.extend;
    istoric_rezervari_start(istoric_rezervari_start.last) := aux.data_inceput;
    istoric_rezervari_sfarsit(istoric_rezervari_sfarsit.last) := aux.data_sfarsit;
end loop;

ok := true;

```

```

if istoric_rezervari_start.count = 0 then
    dbms_output.put_line('Locuinta '|| lista_locuinte(i)||' este disponibila in perioada
introdusa: '|| perioada_start||' / '|| perioada_stop);

```

```
lista_dotari := dot();
```

```

select d.nume
bulk collect into lista_dotari
from dotari d

```

```

join are a on d.id_dotare = a.id_dotare
where a.id_locuinta = i;

if lista_dotari.count <> 0 then
    dbms_output.put_line('Dotarile disponibile ale acestei locuinte sunt: ');
    for d_index in 1..lista_dotari.count loop
        dbms_output.put_line(' - '|| lista_dotari(d_index));
    end loop;
    DBMS_OUTPUT.PUT_LINE(");

dbms_output.put_line('=====');
DBMS_OUTPUT.PUT_LINE(");

else
    dbms_output.put_line('Nu s-au gasit dotari pentru aceasta locuinta.');
    DBMS_OUTPUT.PUT_LINE(");

dbms_output.put_line('=====');
DBMS_OUTPUT.PUT_LINE(");

end if;

else
    for j in 1 .. istoric_rezervari_start.count loop

```

```
    if istoric_rezervari_start(j) <= perioada_stop and istoric_rezervari_sfarsit(j) >=
perioada_start then
```

```
        ok := false;
```

```
        dbms_output.put_line('Perioada selectata se suprapune deja cu o rezervare.E
posibil sa nu mai avem locuri.');
```

```
        DBMS_OUTPUT.PUT_LINE(");
```

```
dbms_output.put_line('=====');
```

```
        DBMS_OUTPUT.PUT_LINE(");
```

```
    end if;
```

```
end loop;
```

```
if ok = true then
```

```
    dbms_output.put_line('Locuinta'|| lista_locuinte(i)|| este disponibila in perioada
introdusa:|| perioada_start||'-'|| perioada_stop);
```

```
    lista_dotari := dot();
```

```
for dotari_rec in(select d.nume
```

```
    from dotari d
```

```
    join are a on d.id_dotare = a.id_dotare
```

```
    where a.id_locuinta = i) loop
```

```
    lista_dotari.extend;
```

```
    lista_dotari(lista_dotari.last) := dotari_rec.nume;
```

```
end loop;
```

```
if lista_dotari.count <> 0 then
```

```
dbms_output.put_line('Dotarile disponibile ale acestei locuinte sunt: ');

for d_index in 1..lista_dotari.count loop

    dbms_output.put_line(' - '|| lista_dotari(d_index));

    end loop;

    DBMS_OUTPUT.PUT_LINE(");

dbms_output.put_line('=====');

DBMS_OUTPUT.PUT_LINE(");

else

    dbms_output.put_line('Nu s-au gasit dotari pentru aceasta locuinta.');

    DBMS_OUTPUT.PUT_LINE(");

dbms_output.put_line('=====');

DBMS_OUTPUT.PUT_LINE(");

end if;

end if;

end if;

end if;

end loop;

end loc_disponibile;
```

```

    dbms_output.put_line('Nu s-au gasit dotari pentru aceasta locuinta.');
    DBMS_OUTPUT.PUT_LINE('');
    dbms_output.put_line('=====');
    DBMS_OUTPUT.PUT_LINE('');
  end if;
end if;
end if;
end loop;

end loc_disponibile;

```

begin

```

loc_disponibile(
  p_regiune => 'Europa de Est',
  perioada_start => TO_DATE('09/11/2005', 'dd/mm/yyyy'),
  perioada_stop => TO_DATE('11/11/2005', 'dd/mm/yyyy')
);

```

end;

/

```

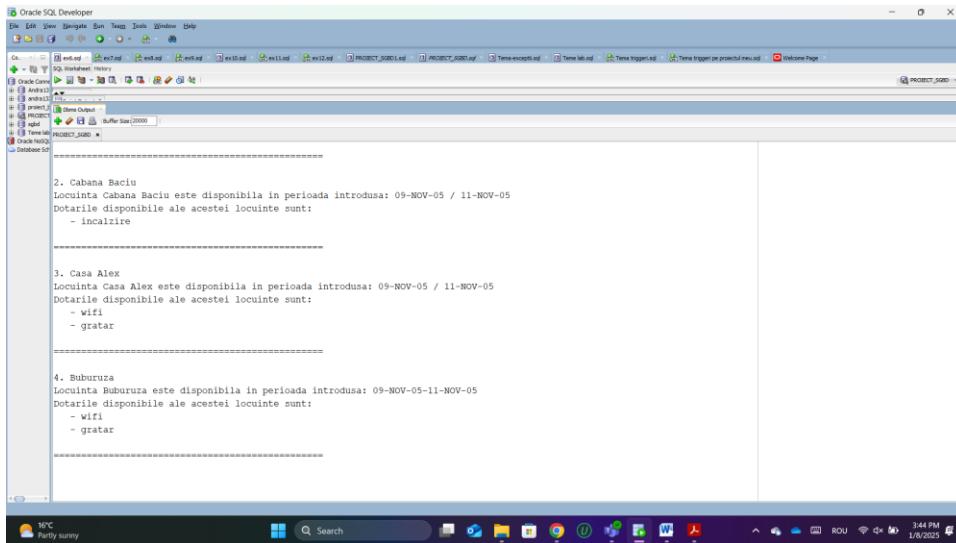
begin
  loc_disponibile(
    p_regiune => 'Europa de Est',
    perioada_start => TO_DATE('09/11/2005', 'dd/mm/yyyy'),
    perioada_stop => TO_DATE('11/11/2005', 'dd/mm/yyyy')
  );

```

PL/SQL procedure successfully completed.

Locuintele gasite in regiunea Europa de Est:

1. Grand Hotel  
Locuinta Grand Hotel este disponibila in perioada introdusa: 09-NOV-05-11-NOV-05  
Dotarile disponibile ale acestei locuinte sunt:
  - wifi
  - parcare
  - aer conditionat
2. Cabana Baciu  
Locuinta Cabana Baciu este disponibila in perioada introdusa: 09-NOV-05 / 11-NOV-05



begin

```

loc_disponibile(
    p_regiune => 'Europa de Est',
    perioada_start => TO_DATE('23/03/2023', 'dd/mm/yyyy'),
    perioada_stop => TO_DATE('24/03/2023', 'dd/mm/yyyy')
);

end;
/

```

Oracle SQL Developer : C:\Users\Andra\Desktop\FMI\ANUL 2\SEM 1\SGBD\project\ex6.sql

```

    perioda_stop => TO_DATE('11/11/2005', 'dd/mm/yyyy')
);

end;
/

begin
    loc_disponibile(
        p_regiune => 'Europa de Est',
        perioada_start => TO_DATE('23/03/2023', 'dd/mm/yyyy'),
        perioada_stop => TO_DATE('24/03/2023', 'dd/mm/yyyy')
    );
end;
/

```

PL/SQL procedure successfully completed.

Doms Output Buffer Size[20000]

PROJECT\_SGBD x

LOCUINTELE gasite in regiunea Europa de Est:

1. Grand Hotel  
Locuinta Grand Hotel este disponibila in perioada introdusa: 23-MAR-23-24-MAR-23  
Dotarile disponibile ale acestei locuinte sunt:
  - wifi
  - parcare
  - aer conditionat

Oracle SQL Developer : C:\Users\Andra\Desktop\FMI\ANUL 2\SEM 1\SGBD\project\ex6.sql

Doms Output Buffer Size[20000]

PROJECT\_SGBD x

LOCUINTELE gasite in regiunea Europa de Est:

1. Grand Hotel  
Locuinta Grand Hotel este disponibila in perioada introdusa: 23-MAR-23-24-MAR-23  
Dotarile disponibile ale acestei locuinte sunt:
  - wifi
  - parcare
  - aer conditionat
2. Cabana Baciu  
Locuinta Cabana Baciu este disponibila in perioada introdusa: 23-MAR-23 / 24-MAR-23  
Dotarile disponibile ale acestei locuinte sunt:
  - incalzire
3. Casa Alex  
Locuinta Casa Alex este disponibila in perioada introdusa: 23-MAR-23 / 24-MAR-23  
Dotarile disponibile ale acestei locuinte sunt:
  - wifi
  - gratar
4. Buburuza  
Perioada selectata se suprapune deja cu o rezervare.E posibil sa nu mai avem locuri.

## **7. EXERCIȚIUL 7 – SUBPROGRAM STOCAT INDEPENDENT + CURSOARE**

**ENUNT:**

Managerii aplicației doresc să implementeze o procedură stocată, numită `raport_venituri_pe_categorii`, care să genereze un raport al veniturilor obținute din rezervări, grupate pe categorii de locuințe, pentru un an introdus ca parametru (`p_an`).

Procedura trebuie să afișeze:

- Venitul total pentru fiecare categorie de locuințe, în anul specificat.
- Locuințele din fiecare categorie, împreună cu venitul lor individual în anul respectiv.

În vederea rezolvării, trebuie să existe minim 2 tipuri de cursoare studiate, iar unul din ele trebuie să fie parametrizat, dependent de celălalt cursor.

Totodata să se trateze și excepțiile care pot apărea în urma specificării anului pentru care se dorește realizarea raportului, iar în cazul în care acestea există, să fie inserate în tabelul de erori, care trebuie creat.

```
create sequence seq_erori
```

```
increment by 1
```

```
start with 1
```

```
nocycle;
```

```
create table erori (
```

```
    id_eroare number constraint pk_eroare primary key,
```

```
    mesaj varchar2(500),
```

```
    data_ora timestamp default current_timestamp,
```

```
    unde_apare varchar2(100)
```

```
)
```

```
create or replace procedure raport_venituri_pe_categorii (p_an in varchar2) is
```

```
-- cursor static pentru selectarea categoriilor distincte
```

```
cursor c_categorii is
```

```
select nume_tip
```

```
from tipuri_locuinte;
```

```
venituri sys_refcursor;
```

```
--cursor parametrizat pentru locatii care fac parte din categoria respectiva
```

```
cursor c_locuinta (p_categ varchar2,p_an number) is
```

```
select l.nume_locuinta, sum(r.numar_persoane * t.pret_per_persoana) as venit_locatie
```

```
from rezervare r
```

```
join se_asociaza_la s on s.id_rezervare = r.id_rezervare
```

```
join locuinta l on s.id_locuinta = l.id_locuinta
```

```
join tipuri_locuinte tip on tip.id_tip = l.id_tip
```

```
join tarife t on t.id_tarif = tip.id_tarif
```

```
where tip.nume_tip = p_categ
```

```
and extract(year from r.data_inceput) = p_an
```

```
and s.status = 'activa'
```

```
group by l.nume_locuinta;
```

```
categ_cur varchar2(50);
```

```
venit_total number;
```

```
an_valid number;
```

```
nume_loc varchar2(100);
```

```

venit_loc number;

null_error exception;

value_error7 exception;

begin

if p_an is null then

    raise null_error;

end if;

if not regexp_like(p_an, '^\\d+') then

    raise value_error7;

end if;

an_valid := to_number(p_an);

dbms_output.put_line('Raport venituri pentru anul ' || an_valid || ':');

dbms_output.put_line('=====');

-- parcurgere cursor static

for i in c_categorii loop

    categ_cur := i.nume_tip;

open venituri for

    'select sum(r.numar_persoane*t.pret_per_persoana)

    from rezervare r

```

```
join se_asociaza_la s on s.id_rezervare = r.id_rezervare
join locuinta l on s.id_locuinta = l.id_locuinta
join tipuri_locuinte tl on tl.id_tip = l.id_tip
join tarife t on t.id_tarif = tl.id_tarif
where tl.nume_tip = :cat and extract(year from r.data_inceput) = :an'
using categ_cur, an_valid;
```

```
fetch venituri into venit_total;
```

```
close venituri;
```

```
if venit_total is not null then
```

```
    dbms_output.put_line('Categorie: ' || categ_cur || ' - venit total: ' || venit_total || ' lei');
```

```
    dbms_output.put_line(");
```

```
else
```

```
    dbms_output.put_line('Categorie: ' || categ_cur || ' - venit total: 0 lei');
```

```
    dbms_output.put_line(");
```

```
end if;
```

```
for j in c_locuinta(categ_cur,an_valid) loop
```

```
    nume_loc := j.nume_locuinta;
```

```
    venit_loc := j.venit_locatie;
```

```
    dbms_output.put_line('Locuinta: ' || nume_loc || ' - venit total: ' || nvl(venit_loc, 0) || ' lei');
```

```
    dbms_output.put_line(");
```

```
end loop;
```

```

end loop;

dbms_output.put_line('=====');

exception

when null_error then

    insert into erori (id_eroare, mesaj, unde_apare)

        values (seq_erori.nextval, 'Parametrul p_an nu poate fi NULL!',

'raport_venituri_pe_categorii');

    dbms_output.put_line('Parametrul p_an nu poate fi NULL!');

when value_error7 then

    insert into erori (id_eroare, mesaj, unde_apare)

        values (seq_erori.nextval, 'Valoarea parametrului este gresita!!!',

'raport_venituri_pe_categorii');

    dbms_output.put_line('Valoarea parametrului este gresita');

when others then

    insert into erori (id_eroare, mesaj, unde_apare)

        values (seq_erori.nextval, 'A aparut o eroare neasteptata!', 'raport_venituri_pe_categorii');

    dbms_output.put_line('A aparut o eroare neasteptata!');

end report_venituri_pe_categorii;
/

```

```

    insert into erori (id_eroare, mesaj, unde_apare)
    values (seq_erori.nextval, 'Parametrul p_an nu poate fi NULL!', 'raport_venituri_pe_categorii');
    dbms_output.put_line('Parametrul p_an nu poate fi NULL!');
when value_error7 then
    insert into erori (id_eroare, mesaj, unde_apare)
    values (seq_erori.nextval, 'Valoarea parametrului este gresita!!!', 'raport_venituri_pe_categorii');
    dbms_output.put_line('Valoarea parametrului este gresita');
when others then
    insert into erori (id_eroare, mesaj, unde_apare)

```

Procedure RAPORT\_VENITURI\_PE\_CATEGORII compiled

DECLARE

an varchar2(20) := '&an\_tastatura';

BEGIN

raport\_venituri\_pe\_categorii(an);

END;

/

```

        values (seq_erori.nextval, 'A aparut o eroare neasteptata!', 'raport_venituri_pe_categorii');
        dbms_output.put_line('A aparut o eroare neasteptata!');

end report_venituri_pe_categorii;
/

DECLARE
an varchar2(20) := '&an_tastatura';
BEGIN
raport_venituri_pe_categorii(an);
END;
/

BEGIN
raport_venituri_pe_categorii(2024);

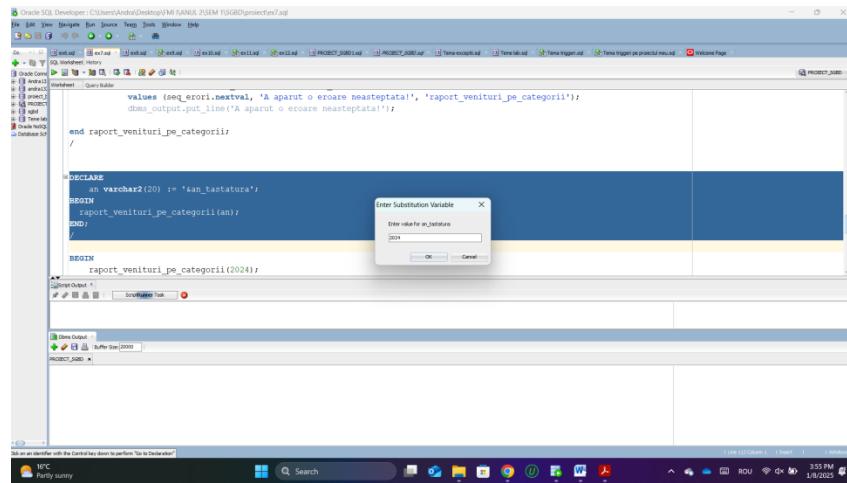
```

Enter Substitution Variable

Enter value for an\_tastatura:

OK Cancel

# TESTĂRI: PENTRU 2024



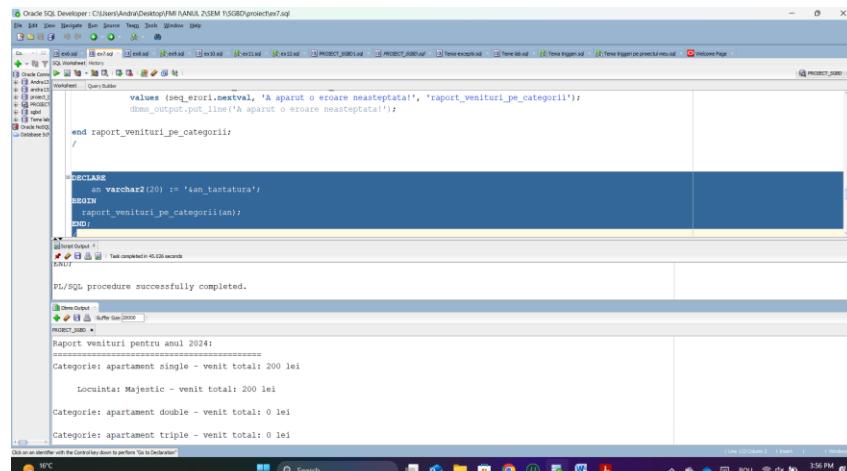
```
values (seq_eroori.nextval, 'A aparut o eroare neasteptata!', 'report_venituri_pe_categorii');
dbms_output.put_line('A aparut o eroare neasteptata!');

end report_venituri_pe_categorii;
/


DECLARE
    an varchar2(20) := 'an_tastatura';
BEGIN
    report_venituri_pe_categorii(an);
END;

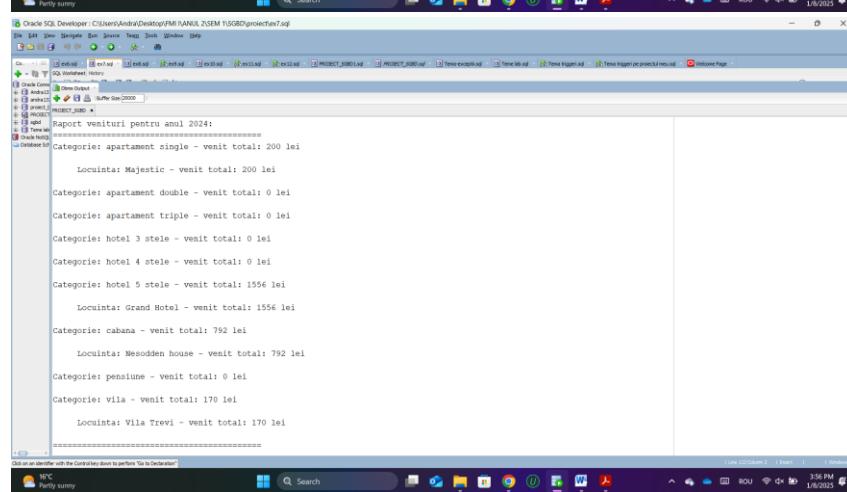
BEGIN
    report_venituri_pe_categorii(2024);

```



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

```
Raport venituri pentru anul 2024:
=====
Categorie: apartament single - venit total: 200 lei
    Locuinta: Majestic - venit total: 200 lei
    Categorie: apartament double - venit total: 0 lei
    Categorie: apartament triple - venit total: 0 lei
```



```
Categorie: apartament single - venit total: 200 lei
    Locuinta: Majestic - venit total: 200 lei
    Categorie: apartament double - venit total: 0 lei
    Categorie: apartament triple - venit total: 0 lei
    Categorie: hotel 3 stele - venit total: 0 lei
    Categorie: hotel 4 stele - venit total: 0 lei
    Categorie: hotel 5 stele - venit total: 1556 lei
        Locuinta: Grand Hotel - venit total: 1556 lei
    Categorie: cabana - venit total: 792 lei
        Locuinta: Mesodde house - venit total: 792 lei
    Categorie: pensiune - venit total: 0 lei
    Categorie: vila - venit total: 170 lei
        Locuinta: Vila Trevi - venit total: 170 lei
=====
```

## PENTRU VALOAREA NULL

The screenshot shows the Oracle SQL Developer interface. A PL/SQL procedure is being edited in the central workspace. A modal dialog titled "Enter Substitution Variable" is open, prompting for a value for the variable `an_tastatura`. The procedure code includes a `DBMS_OUTPUT.PUT_LINE` statement that references this variable.

```
values (seq_stori.nextval, 'A aparut o eroare neasteptata!', 'report_venituri_pe_categorii');
dbms_output.put_line('A aparut o eroare neasteptata!');

end report_venituri_pe_categorii;
/

DECLARE
    an varchar2(20) := 'ian_tastatura';
BEGIN
    report_venituri_pe_categorii(an);
END;

BEGIN
```

Below the code editor, the "Script Output" window shows the message "PL/SQL procedure successfully completed." and the message "Parametrul p\_an nu poate fi NULL!". A "Snipping Tool" window is also visible at the bottom right, indicating a screenshot was taken.

## PENTRU O VALOARE GRESITA – 2A

```

values (seq_eroari.nextval, 'aparut o eroare neasteptata!', 'report_venituri_pe_categorii');
    dms_output.put_line('A aparut o eroare neasteptata!');

end report_venituri_pe_categorii;
/


DECLARE
    an varchar2(20) := 'an_tastatura';
BEGIN
    report_venituri_pe_categorii(an);
END;
/

```

PL/SQL procedure successfully completed.

Parameterul p\_an nu poate fi NULL!

Valoarea parametrului este gresita

## TABELUL ERORI IN URMA ULTIMELOR ERORI GENERATE

```

select *
from erori;

```

ID_ERORAE	MESAJ	DATA_ORA	UNDE_APARE
1	Sunt mai multe rezervari cu aceeasi valoare maxima!!!	15-DEC-24 07.39.23.085000000 PM	rezervare maxima
2	2In tara introdusa nu s-au inregistrat rezervari!!!	15-DEC-24 07.39.27.986000000 PM	rezervare maxima
3	3NU s-a gasit tara in baza de date!	15-DEC-24 07.39.35.882000000 PM	rezervare maxima
4	4Valoarea parametrului este gresita!!!	15-DEC-24 07.39.42.250000000 PM	rezervare maxima
5	5Nu exista eroare: ORA-20100: Parametrul p_an nu poate fi NULL.	15-DEC-24 07.39.48.10.000000000 PM	report venituri pe categorii
6	6Nu aparut o eroare: ORA-20100: Parametrul p_an nu poate fi NULL.	15-DEC-24 07.51.34.000000000 PM	report venituri pe categorii
7	7Parametrul p_an nu poate fi NULL!	15-DEC-24 07.51.54.226000000 PM	report venituri pe categorii
8	8Parametrul p_an nu poate fi NULL!	15-DEC-24 07.54.19.097000000 PM	report venituri pe categorii
9	9Parametrul p_an nu poate fi NULL!	15-DEC-24 07.56.23.549000000 PM	report venituri pe categorii
10	10Aparut o eroare neasteptata!	15-DEC-24 07.56.26.659000000 PM	report venituri pe categorii
11	11Valoarea parametrului este gresita!!!	15-DEC-24 07.57.18.777000000 PM	rezervare maxima
12	12Valoarea parametrului este gresita!!!	15-DEC-24 07.57.57.228000000 PM	report venituri pe categorii
13	13Valoarea parametrului este gresita!!!	15-DEC-24 07.58.11.952000000 PM	report venituri pe categorii
14	21Parametrul p_an nu poate fi NULL!	08-JAN-25 03.57.16.679000000 PM	report venituri pe categorii
15	22Parametrul p_an nu poate fi NULL!	08-JAN-25 03.57.34.289000000 PM	report venituri pe categorii
16	23Valoarea parametrului este gresita!!!	08-JAN-25 03.58.26.542000000 PM	report venituri pe categorii

## **8. EXERCIȚIUL 8 – SUBPROGRAM STOCAT INDEPENDENT – FUNCȚIE**

### **ENUNT**

Să se implementeze o funcție stocată independentă, numită rezervare\_maxima, care primește ca parametru numele unei țări și returnează valoarea maximă a unei rezervări efectuate în țara respectivă, indiferent de locația din țară. Funcția trebuie să returneze, împreună cu valoarea rezervării, următoarele informații:

- Numele și prenumele utilizatorului care a efectuat rezervarea.
- ID-ul rezervării.
- Numele locuinței rezervate.

Să se trateze excepțiile posibile, inclusiv:

- Țara nu există în baza de date.
- Nu s-au găsit rezervări în țara respectivă.
- Cel puțin două rezervări au aceeași valoare maximă.
- Parametrul introdus este invalid (ex: un număr în loc de text).

La fel ca la exercițiul anterior, erorile apărute se vor stoca în tabelul de erori, deja creat.

```
create or replace type info_rezervare_record as object(
```

```
    val_max number,  
    nume_utilizator varchar2(30),  
    prenume_utilizator varchar2(30),  
    id_rezervare number,  
    nume_loc varchar2(30)  
);
```

```
create or replace function rezervare_maxima (tara_input tari.nume_tara%type) return  
info_rezervare_record is
```

```
    venit_max number := 0;  
    nume_u_func utilizator.nume%type;
```

```

prenume_u_func utilizator.prenume%type;
rezerv rezervare.id_rezervare%type;
loc_val locuinta.nume_locuinta%type;
numar_rezervari number := 0;

fara_rezervari exception;

begin
if tara_input is null or length(trim(tara_input)) = 0 or regexp_like(tara_input, '^\\d+$') then
    raise value_error;
end if;

declare
    tara_exista number := 0;
begin
    select count(*) into tara_exista
    from tari
    where lower(nume_tara) = lower(tara_input);

    if tara_exista = 0 then
        raise no_data_found;
    end if;

end;

select count(*) into numar_rezervari

```

```

from rezervare r
join se_asociaza_la s on s.id_rezervare = r.id_rezervare
join locuinta l on l.id_locuinta = s.id_locuinta
join locatie loc on l.id_locatie = loc.id_locatie
join tari ta on ta.id_tara = loc.id_tara
where s.status = 'activa'
and lower(ta.nume_tara) = lower(tara_input);

if numar_rezervari = 0 then
    raise fara_rezervari;
else
    select r.id_rezervare,l.nume_locuinta,max(t.pret_per_persoana*r.numar_persoane) into
rezerv, loc_val, venit_max
    from rezervare r
    join se_asociaza_la s on s.id_rezervare = r.id_rezervare
    join locuinta l on l.id_locuinta = s.id_locuinta
    join tipuri_locuinte tip on tip.id_tip = l.id_tip
    join tarife t on t.id_tarif = tip.id_tarif
    join locatie loc on l.id_locatie = loc.id_locatie
    join tari ta on ta.id_tara = loc.id_tara
    where s.status = 'activa'
    and lower(ta.nume_tara) = lower(tara_input)
    group by l.nume_locuinta,r.id_rezervare;
end if;
if venit_max <> 0 then

```

```
select u.nume,u.prenume into nume_u_func, prenume_u_func
from utilizator u
join rezervare r on u.id_utilizator = r.id_utilizator
where r.id_rezervare = rezerv;
end if;

return info_rezervare_record(venit_max,nume_u_func,prenume_u_func,rezerv,loc_val);
```

exception

when value\_error then

```
insert into erori (id_eroare, mesaj, unde_apare)
values (seq_erori.nextval, 'Valoarea parametrului este gresita!!!', 'rezervare_maxima');

dbms_output.put_line('Valoarea parametrului este gresita');

return info_rezervare_record(0, NULL, NULL, NULL, NULL);
```

when fara\_rezervari then

```
insert into erori (id_eroare, mesaj, unde_apare)
values (seq_erori.nextval, 'In tara introdusa nu s-au inregistrat rezervari!!!',
'rezervare_maxima');

dbms_output.put_line('In tara introdusa nu s-au inregistrat rezervari');

return info_rezervare_record(0, NULL, NULL, NULL, NULL);
```

when no\_data\_found then

```
insert into erori (id_eroare, mesaj, unde_apare)
values (seq_erori.nextval, 'Nu s-a gasit tara in baza de date!', 'rezervare_maxima');

dbms_output.put_line('Nu s-a gasit tara in baza de date!!!');

return info_rezervare_record(0, NULL, NULL, NULL, NULL);
```

```

when too_many_rows then

    insert into erori (id_eroare, mesaj, unde_apare)

        values (seq_erori.nextval, 'Sunt mai multe rezervari cu aceeasi valoare maxima!!!',
'rezervare_maxima');

    dbms_output.put_line('Sunt mai multe rezervari cu aceeasi valoare maxima!');

    return info_rezervare_record(0, NULL, NULL, NULL, NULL);

end rezervare_maxima;

/

```

The screenshot shows the Oracle SQL Developer interface with the 'Query Builder' tab selected. The code for the REZERVARE\_MAXIMA function is displayed in the central pane:

```

dbms_output.put_line('In tara introdusa nu s-au inregistrat rezervari');
return info_rezervare_record(0, NULL, NULL, NULL, NULL);
when no_data_found then
    insert into erori (id_eroare, mesaj, unde_apare)
        values (seq_erori.nextval, 'Nu s-a gasit tara in baza de date!', 'rezervare_maxima');
    dbms_output.put_line('Nu s-a gasit tara in baza de date!!!!');
    return info_rezervare_record(0, NULL, NULL, NULL, NULL);
when too_many_rows then
    insert into erori (id_eroare, mesaj, unde_apare)
        values (seq_erori.nextval, 'Sunt mai multe rezervari cu aceeasi valoare maxima!!!', 'rezervare_maxima');
    dbms_output.put_line('Sunt mai multe rezervari cu aceeasi valoare maxima!');

```

The bottom pane shows the 'Script Output' results:

```

Function REZERVARE_MAXIMA compiled

```

```

declare

    tara_test tari.nume_tara%type := '&p_tara';

    rezervare_test info_rezervare_record;

begin

    rezervare_test := rezervare_maxima(tara_test);

    dbms_output.put_line('=====');
    dbms_output.put_line('Tara: '|| tara_test);
    dbms_output.put_line('Valoare maxima: '|| rezervare_test.val_max);

```

```

dbms_output.put_line('Nume utilizator: '|| rezervare_test.nume_utilizator ||' '|
rezervare_test.prenume_utilizator);

dbms_output.put_line('Id-ul rezervarii: '|| rezervare_test.id_rezervare);

dbms_output.put_line('Locuinta: '|| rezervare_test.nume_loc);

end;

```

```

CREATE OR REPLACE PROCEDURE rezervare_maxima(tara_test IN OUT tara%TYPE)
IS
    rezervare_test rezervare%ROWTYPE;
BEGIN
    rezervare_test := rezervare_maxima(tara_test);

    dbms_output.put_line('-----');
    dbms_output.put_line('Tara: '|| tara_test);
    dbms_output.put_line('Valoare maxima: '|| rezervare_test.val_max);
    dbms_output.put_line('Nume utilizator: '|| rezervare_test.nume_utilizator ||' '|| rezervare_test.prenume_utilizator);
    dbms_output.put_line('Id-ul rezervarii: '|| rezervare_test.id_rezervare);
    dbms_output.put_line('Locuinta: '|| rezervare_test.nume_loc);

END;
/

```

```

insert into tari values('IDK','Danemarca',4);

select *
from erori;

```

TESTARI

## PENTRU ROMANIA

```

CREATE OR REPLACE PROCEDURE rezervare_maxima(tara_test IN OUT tara%TYPE)
IS
    rezervare_test rezervare%ROWTYPE;
BEGIN
    rezervare_test := rezervare_maxima(tara_test);

    dbms_output.put_line('-----');
    dbms_output.put_line('Tara: '|| tara_test);
    dbms_output.put_line('Valoare maxima: '|| rezervare_test.val_max);
    dbms_output.put_line('Nume utilizator: '|| rezervare_test.nume_utilizator ||' '|| rezervare_test.prenume_utilizator);
    dbms_output.put_line('Id-ul rezervarii: '|| rezervare_test.id_rezervare);
    dbms_output.put_line('Locuinta: '|| rezervare_test.nume_loc);

END;
/

```

```

insert into tari values('IDK','Romania',4);

select *
from erori;

```

PL/SQL procedure successfully completed.

```

declare
    tara_test tari.name_tariatype := 'ro_România';
    rezervare_test_info_rezervare_record;
begin
    rezervare_test := rezervare_maximatarare_test;

    dms_output.put_line('*****');
    dms_output.put_line('Tara: '|| tara_test);
    dms_output.put_line('Valoare maxima: '|| rezervare_test.val_max);
    dms_output.put_line('Nume utilizator: '|| rezervare_test.name_utilizator ||' '|| rezervare_test.prenume_utilizator);
    dms_output.put_line('Id-ul rezervarii: '|| rezervare_test.id_rezervare);
    dms_output.put_line('Locuita: '|| rezervare_test.num_loc);
end;
/

```

PL/SQL procedure successfully completed.

Disk Output:

```

SUNT MAI MULTE REZERVARI CU ACEEASI VALOARE MAXIMA!
-----
Tara: România
Valoare maxima: 0
Nume utilizator:
Id-ul rezervarii:
Locuita:

```

## PENTRU BELGIA

```

declare
    tara_test tari.name_tariatype := 'be_Belgia';
    rezervare_test_info_rezervare_record;
begin
    rezervare_test := rezervare_maximatarare_test;

    dms_output.put_line('*****');
    dms_output.put_line('Tara: '|| tara_test);
    dms_output.put_line('Valoare maxima: '|| rezervare_test.val_max);
    dms_output.put_line('Nume utilizator: '|| rezervare_test.name_utilizator ||' '|| rezervare_test.prenume_utilizator);
    dms_output.put_line('Id-ul rezervarii: '|| rezervare_test.id_rezervare);
    dms_output.put_line('Locuita: '|| rezervare_test.num_loc);
end;
/

```

insert into tari values('DK','Danemarca',4);

select \* from error;

PL/SQL procedure successfully completed.

Disk Output:

```

SUNT MAI MULTE REZERVARI CU ACEEASI VALOARE MAXIMA!
-----
Tara: România
Valoare maxima: 0
Nume utilizator:
Id-ul rezervarii:
Locuita:

```

```

declare
    tara_test tari.name_tariatype := 'dk_Danemarca';
    rezervare_test_info_rezervare_record;
begin
    rezervare_test := rezervare_maximatarare_test;

    dms_output.put_line('*****');
    dms_output.put_line('Tara: '|| tara_test);
    dms_output.put_line('Valoare maxima: '|| rezervare_test.val_max);
    dms_output.put_line('Nume utilizator: '|| rezervare_test.name_utilizator ||' '|| rezervare_test.prenume_utilizator);
    dms_output.put_line('Id-ul rezervarii: '|| rezervare_test.id_rezervare);
    dms_output.put_line('Locuita: '|| rezervare_test.num_loc);
end;
/

```

PL/SQL procedure successfully completed.

Disk Output:

```

Tara: Belgia
Valoare maxima: 420
Nume utilizator: Vasile Moma
Id-ul rezervarii: 1004
Locuita: Pink promise

```

## PENTRU DANEMARCA (pe care o adăugam la lista de ţări,pentru a evidenția comportamentului)

insert into tari values('DK','Danemarca',4);

```

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

Tara: Belgia
Valoare maxim: 400
Num_utilizator: Vasile Mona
Id-ul rezervarii: 1084
Locuinta: Fink promie

```

```

In tara introdusa nu s-a inregistrat rezervari
Tara: Danemarca
Valoare maxim: 0
Num_utilizator:
Id-ul rezervarii:
Locuinta:

```

PENTRU 1234

```

In tara introdusa nu s-a inregistrat rezervari
Tara: Danemarca
Valoare maxim: 0
Num_utilizator:
Id-ul rezervarii:
Locuinta:

```

```

PL/SQL procedure successfully completed.

SQL> select * from tara;
SQL> Valoarea parametrului este gresita
SQL> TARAID
SQL> Valoare maxima: 0
SQL> Nume utilizator:
SQL> Id-ul rezervarii:
SQL> Locuinta:

```

## PENTRU RUSIA

```

PL/SQL procedure successfully completed.

SQL> select * from tara;
SQL> Valoarea parametrului este gresita
SQL> TARA
SQL> Valoare maxima: 0
SQL> Nume utilizator:
SQL> Id-ul rezervarii:
SQL> Locuinta:

```

```

PL/SQL procedure successfully completed.

SQL> select * from tara;
SQL> Valoarea parametrului este gresita
SQL> TARA
SQL> Valoare maxima: 0
SQL> Nume utilizator:
SQL> Id-ul rezervarii:
SQL> Locuinta:

```

## TABELUL DE ERORI DUPA ULTIMELE ERORI APARUTE

Oracle SQL Developer : C:\Users\Andrei\Desktop\FMI\1ANUL\2SEM\1\\$GBD\project\ex8.sql

File Edit View Navigate Run Source Team Tools Window Help

Co... ex4.sql ex7.sql ex8.ad ex9.sql ex10.sql ex11.sql ex12.sql PROJECT\_SGBD\_1.sql PROJECT\_SGBD\_2.sql Tema-exceptii.sql Tema\_lab.sql Tema\_trigger.sql Tema trigger pe proiectul meu.sql Welcome Page

Project Oracle Conn... SQL Worksheet History

Andrei-PC\Andrei-PC

project\_1

project\_2

rgbd

Temă lab

Oracle Net8

Database Schemas

Query Builder

```
dbms_output.put_line('Tara: ''|| tara_test');
dbms_output.put_line('Valoare maxima: ''|| rezervare_test.val_max');
dbms_output.put_line('Nume utilizator: ''|| rezervare_test.nume_utilizator ||' ''|| rezervare_test.prenume_utilizator);
dbms_output.put_line('Id-ul rezervarii: ''|| rezervare_test.id_rezervare);
dbms_output.put_line('Locuinta: ''|| rezervare_test.nume_loc);

end;

insert into tari values('DK', 'Danemarca', 4);

select *
from erori;
```

Script Output x Query Result x

All Rows Fetched: 24 in 0.011 seconds

ID_BIROU	MESAJ	DATA_ORA	UNDE_APARE
8	Parametrul p an nu poate fi NULL!	15-DEC-24 07.54.15.097000000	PM rezervare venituri pe categorii
9	Parametrul p an nu poate fi NULL!	15-DEC-24 07.56.23.549000000	PM rezervare venituri pe categorii
10	10A aparut o eroare neasteptata!	15-DEC-24 07.56.26.659000000	PM rezervare venituri pe categorii
11	11Valoarea parametrului este gresita!!!	15-DEC-24 07.57.18.777000000	PM rezervare maxima
12	12Valoarea parametrului este gresita!!!	15-DEC-24 07.57.57.228000000	PM rezervare venituri pe categorii
13	13Valoarea parametrului este gresita!!!	15-DEC-24 07.58.11.952000000	PM rezervare venituri pe categorii
14	21Parametrul p an nu poate fi NULL!	08-JAN-25 03.57.16.679000000	PM rezervare venituri pe categorii
15	22Parametrul p an nu poate fi NULL!	08-JAN-25 03.57.34.289000000	PM rezervare venituri pe categorii
16	23Valoarea parametrului este gresita!!!	08-JAN-25 03.58.26.542000000	PM rezervare venituri pe categorii
17	24Sunt mai multe rezervari cu aceeasi valoare maxima!!!	08-JAN-25 04.08.29.688000000	PM rezervare maxima
18	25In tara introduse nu s-au inregistrat rezervari!!!	08-JAN-25 04.09.07.292000000	PM rezervare maxima
19	26Sunt mai multe rezervari cu aceeasi valoare maxima!!!	08-JAN-25 04.10.33.743000000	PM rezervare maxima
20	27Sunt mai multe rezervari cu aceeasi valoare maxima!!!	08-JAN-25 04.11.41.478000000	PM rezervare maxima
21	28In tara introduse nu s-au inregistrat rezervari!!!	08-JAN-25 04.12.28.540000000	PM rezervare maxima
22	29Nu s-a gasit tara in baza de date!	08-JAN-25 04.13.23.465000000	PM rezervare maxima
23	30Valoarea parametrului este gresita!!!	08-JAN-25 04.13.55.908000000	PM rezervare maxima
24	31Nu s-a gasit tara in baza de date!	08-JAN-25 04.14.44.414000000	PM rezervare maxima

Open Output

Click on an identifier with the Control key down to perform "Go To Declaration".

16°C  
Partly sunny

Search

File Edit View Navigate Run Source Team Tools Window Help

Line 135 Column 12 | Insert | 415 PM 1/8/2025

## 9. EXERCIȚIUL 9 - SUBPROGRAM STOCAT INDEPENDENT – PROCEDURĂ

ENUNT:

Să se implementeze o procedură stocată, numită analiza\_rezervari\_utilizator, care să analizeze comportamentul unui utilizator pe baza rezervărilor efectuate. Procedura primește ca parametri numele (p\_nume\_utilizator) și prenumele (p\_prenume\_utilizator) unui utilizator și returnează un raport de tip analiza\_record, care conține:

1. Numărul de modalități distincte de plată utilizate pentru rezervări active.
2. Lista tipurilor de locuințe rezervate.
3. Valoarea totală plătită pentru toate rezervările.
4. Cel mai frecvent tip de locuință rezervat.
5. Perioada totală de închiriere (în zile).
6. Categoria utilizatorului, stabilită astfel:
  - **Bronze:** < 2 rezervări.
  - **Silver:** între 2 și 5 rezervări.
  - **Gold:** între 5 și 10 rezervări.
  - **Premium:** > 10 rezervări.

Procedura trebuie să gestioneze următoarele cazuri de erori:

1. Utilizatorul nu există în baza de date.
2. Utilizatorul nu are rezervări active.
3. Există rezervări cu modalități de plată invalide (altele decât cash, transfer, card).
4. Apar erori legate de valori nule sau alte probleme neprevăzute.

Dacă apar erori , acestea vor fi stocate ca până acum în tabelul de erori.

```
create or replace type tipuri_varray as varray(200) of varchar2(20);
```

```
create or replace type analiza_record as object(
```

```
    numar_modalitati number,
```

```
    lista_tipuri tipuri_varray,
```

```
    valoare_totala number,
```

```
tip_frecvent varchar2(20),  
total_zile number,  
categorie_utilizator varchar2(50)  
);
```

```
create or replace procedure analiza_rezervari_utilizator(
```

```
    p_nume_utilizator in varchar2,  
    p_prenume_utilizator in varchar2,  
    p_analiza out analiza_record ) is
```

```
--=====exceptii proprii=====
```

```
ex_rezervari_active exception;
```

```
ex_valoare_zero exception;
```

```
ex_modalitate_invalida exception;
```

```
--=====
```

```
cod_u utilizator.id_utilizator%type;
```

```
numar_modalitati number := 0;
```

```

lista_tipuri tipuri_varray := tipuri_varray();

valoare_totala number := 0;

tip_frecvent varchar2(50);

total_zile number;

categorie_utilizator varchar2(50);

nr_rezerv number := 0;

mod_plata_nedefinit number:=0;

cursor c_tipuri is

    select distinct t.nume_tip

    from tipuri_locuinte t

    join locuinta l on l.id_tip = t.id_tip

    join se_asociaza_la s on l.id_locuinta = s.id_locuinta

    join rezervare r on s.id_rezervare = r.id_rezervare

    where r.id_utilizator = cod_u;

cursor curs_frecventa is

    select t.nume_tip, count(*) as numar

    from tipuri_locuinte t

    join locuinta l on l.id_tip = t.id_tip

    join se_asociaza_la s on s.id_locuinta = l.id_locuinta

    join rezervare r on r.id_rezervare = s.id_rezervare

    where r.id_utilizator = cod_u

```

```

group by t.nume_tip
order by count(*) desc;

c_frecv_rec curs_frecventa%rowtype;

begin
--iau id-ul util dupa numele dat ca parametru
select id_utilizator into cod_u
from utilizator
where lower(nume) = lower(p_nume_utilizator) and lower(prenume) =
lower(p_prenume_utilizator);

-- nr modalitatile lui de plata(le iau in calcul doar daca rezervarile au ramas active) si nr de
rezervari facute
select count(distinct r.modalitate_plata) into numar_modalitati
from rezervare r
join se_asociaza_la s on r.id_rezervare = s.id_rezervare
where r.id_utilizator = cod_u
and s.status = 'activa';

select count(*) into mod_plata_nedefinit
from rezervare r
join se_asociaza_la s on r.id_rezervare = s.id_rezervare
where r.id_utilizator = cod_u

```

```
and s.status = 'activa'  
and r.modalitate_plata not in ('cash', 'transfer', 'card');
```

```
if mod_plata_nedefinit > 0 then  
    raise ex_modalitate_invalida;  
end if;
```

```
select count(r.id_rezervare) into nr_rezerv  
from rezervare r  
where r.id_utilizator = cod_u;
```

```
--exceptie custom
```

```
if nr_rezerv = 0 then  
    raise ex_rezervari_active;  
end if;
```

```
--determinare categ utilizator in functie de nr de r
```

```
if nr_rezerv < 2 then  
    categorie_utilizator := 'Utilizator BRONZE';  
elsif nr_rezerv between 2 and 5 then  
    categorie_utilizator := 'Utilizator SILVER';  
elsif nr_rezerv between 5 and 10 then  
    categorie_utilizator := 'Utilizator GOLD';
```

```

else
    categorie_utilizator := 'Utilizator PREMIUM';
end if;

lista_tipuri := tipuri_varray();

-- lista de tipuri rezervate

for c_rec in c_tipuri loop
    lista_tipuri.extend;
    lista_tipuri(lista_tipuri.last) := c_rec.nume_tip;
end loop;

if lista_tipuri.count = 0 then
    dbms_output.put_line('Nu există tipuri de locuințe rezervate pentru utilizator.');
end if;

-- valoare totală ->>>>>> 5 dintre tabelele create

select nvl(sum(r.numar_persoane * t.pret_per_persoana), 0)
into valoare_totala
from rezervare r
join se_asociaza_la s on r.id_rezervare = s.id_rezervare
join locuinta l on s.id_locuinta = l.id_locuinta
join tipuri_locuinte tip on tip.id_tip = l.id_tip

```

```
join tarife t on t.id_tarif = tip.id_tarif  
where r.id_utilizator = cod_u;
```

```
if valoare_totala = 0 then  
    raise ex_valoare_zero;  
end if;
```

```
-- tipul cel mai frecvent  
open curs_frecventa;  
fetch curs_frecventa into c_frecv_rec;  
if curs_frecventa%found then  
    tip_frecvent := c_frecv_rec.nume_tip;  
else  
    tip_frecvent := null;  
end if;  
close curs_frecventa;
```

```
-- nr zile  
select sum(data_sfarsit - data_incep)   
into total_zile  
from rezervare  
where id_utilizator = cod_u;
```

```

p_analiza := analiza_record(
    numar_modalitati,
    lista_tipuri, -- Atribuie colecția inițializată
    valoare_totala,
    tip_frecvent,
    total_zile,
    categorie_utilizator
);

exception
    when ex_rezervari_active then
        insert into erori (id_eroare, mesaj, unde_apare)
            values (seq_erori.nextval, 'Utilizatorul există, dar nu are sau nu a avut rezervari active.', 'analiza_rezervari_utilizator');

        dbms_output.put_line('Utilizatorul există, dar nu are sau nu a avut rezervari active.');

    when ex_valoare_zero then
        insert into erori (id_eroare, mesaj, unde_apare)
            values (seq_erori.nextval, 'Valoarea totală este zero, verifică tarifele și rezervările.', 'analiza_rezervari_utilizator');

        dbms_output.put_line('Valoarea totală este zero, verifică tarifele și rezervările.');

    when ex_modalitate_invalida then
        insert into erori (id_eroare, mesaj, unde_apare)
            values (seq_erori.nextval, 'S-a înregistrat o modalitate invalidă de plată, verifică înregistrările.', 'analiza_rezervari_utilizator');

```

```
dbms_output.put_line('S-a inregistrat o modalitate invalida de plata,verificati  
inregistrarile.');
```

```
when no_data_found then
```

```
    insert into erori (id_eroare, mesaj, unde_apare)  
  
    values (seq_erori.nextval, 'Nu s-au gasit datele cautate!', 'analiza_rezervari_utilizator');  
  
    dbms_output.put_line('Nu s-au gasit datele cautate');
```

```
when too_many_rows then
```

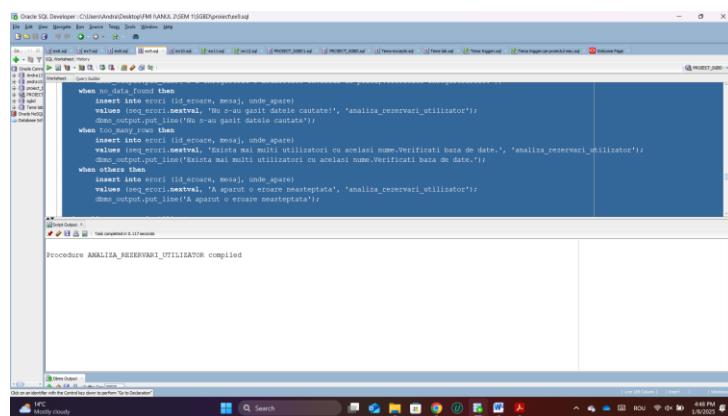
```
    insert into erori (id_eroare, mesaj, unde_apare)  
  
    values (seq_erori.nextval, 'Exista mai multi utilizatori cu acelasi nume.Verificati baza de  
date.', 'analiza_rezervari_utilizator');  
  
    dbms_output.put_line('Exista mai multi utilizatori cu acelasi nume.Verificati baza de date.');
```

```
when others then
```

```
    insert into erori (id_eroare, mesaj, unde_apare)  
  
    values (seq_erori.nextval, 'A aparut o eroare neasteptata', 'analiza_rezervari_utilizator');  
  
    dbms_output.put_line('A aparut o eroare neasteptata');
```

```
end analiza_rezervari_utilizator;
```

```
/
```



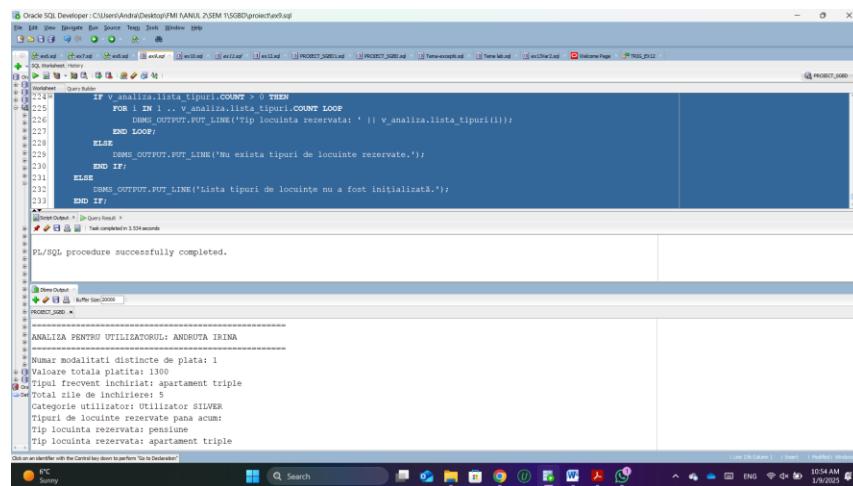
The screenshot shows the Oracle SQL Developer interface with the code for the ANALIZA\_REZERVARI\_UTILIZATOR procedure. The code includes logic for handling different error scenarios: no data found, too many rows, and others. It uses the DBMS\_OUTPUT package to print messages to the console. The code is well-formatted with proper indentation and line breaks.

```
when no_data_found then  
    insert into erori (id_eroare, mesaj, unde_apare)  
    values (seq_erori.nextval, 'Nu s-au gasit datele cautate!', 'analiza_rezervari_utilizator');  
    dbms_output.put_line('Nu s-au gasit datele cautate');  
when too_many_rows then  
    insert into erori (id_eroare, mesaj, unde_apare)  
    values (seq_erori.nextval, 'Exista mai multi utilizatori cu acelasi nume.Verificati baza de date.', 'analiza_rezervari_utilizator');  
    dbms_output.put_line('Exista mai multi utilizatori cu acelasi nume.Verificati baza de date.');//  
when others then  
    insert into erori (id_eroare, mesaj, unde_apare)  
    values (seq_erori.nextval, 'A aparut o eroare neasteptata', 'analiza_rezervari_utilizator');  
    dbms_output.put_line('A aparut o eroare neasteptata');
```

Procedure ANALIZA\_REZERVARI\_UTILIZATOR compiled

TESTARE:

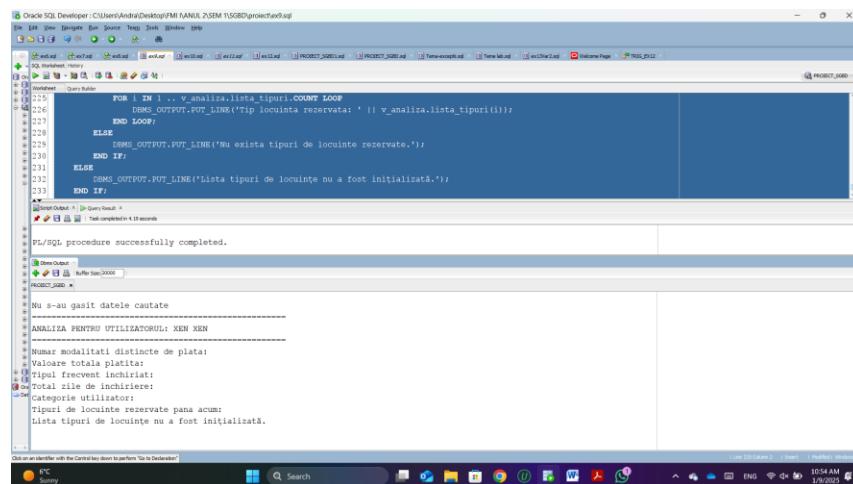
PENTRU ANDRUTA IRINA



```
PL/SQL procedure successfully completed.

-----ANALIZA PENTRU UTILIZATORUL: ANDRUTA IRINA-----
-----Numar modalitati distincte de plată: 1
-----Valoare totala platita: 1300
-----Tipul frequent inchiriator: apartament triple
-----Total zile de inchiriere: 5
-----Categorie utilizator: Utilizator SILVER
-----Tipuri de locuinte rezervate pana acum:
-----Tip locuinta rezervata: pensiune
-----Tip locuinta rezervata: apartament triple
```

PENTRU Xen Xen (NU EXISTA IN BD)



```
PL/SQL procedure successfully completed.

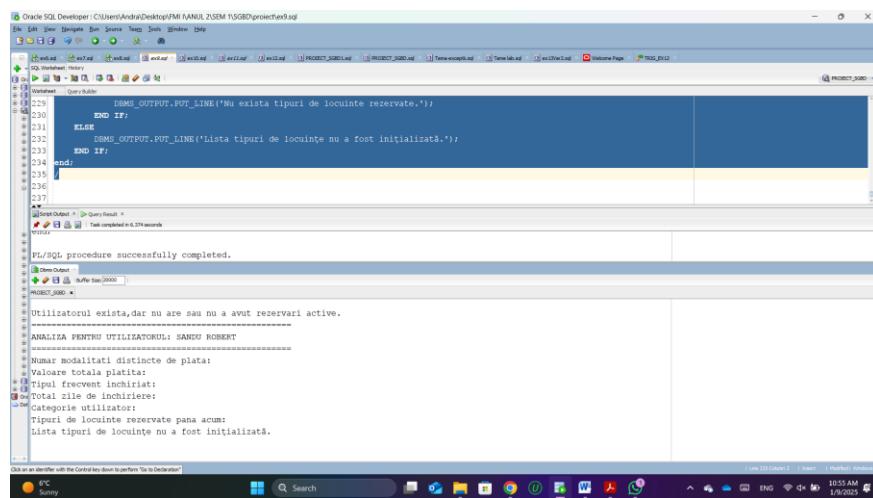
-----Nu s-au gasit datele cautate
-----ANALIZA PENTRU UTILIZATORUL: XEN XEN
-----Numar modalitati distincte de plată:
-----Valoare totala platita:
-----Tipul frequent inchiriator:
-----Total zile de inchiriere:
-----Categorie utilizator:
-----Tipuri de locuinte rezervate pana acum:
-----Lista tipuri de locuinte nu a fost initializată.
```

PENTRU Sandu Robert (este in baza de date, dar nu are rezervari)

Această inserare a fost facută în timpul efectuării proiectului pentru niște testări ale altor exerciții, iar la momentul verificării acestui exercițiu, Sandu Robert deja era prezent în baza de date. (voi pune mai jos inserarea acestuia care a fost făcută).

```
insert into utilizator  
(id_utilizator,nume,prenume,email,data_inregistrare,telefon,gen,data_nasterii)
```

```
values(seq_utilizator.nextval,'Sandu','Robert','sandu.test@gmail.com',to_date('05/10/2024','dd/mm/yyyy'),null,'masculin',to_date('03/03/2005','dd/mm/yyyy'));
```



```
DECLARE  
BEGIN  
    DBMS_OUTPUT.PUT_LINE('Nu există tipuri de locuințe rezervate.');  
    END IF;  
    ELSE  
        DBMS_OUTPUT.PUT_LINE('Lista tipuri de locuințe nu a fost inițializată.');//  
    END IF;  
END;  
  
PL/SQL procedure successfully completed.  
  
-----  
Utilizatorul există, dar nu are sau nu a avut rezervări active.  
-----  
ANALIZA PENTRU UTILIZATORUL: SANDU ROBERT  
-----  
Numar modalitati distincte de plată:  
Valoare totală platită:  
Tipul frecvent închiriat:  
Total zile de închirieri:  
Categorie utilizator:  
Tipuri de locuințe rezervate pana acum:  
Lista tipuri de locuințe nu a fost inițializată.
```

PENTRU Sandu Anastasia (Pe care o inserăm în baza de date, și care va avea modalitatea de plată pos (în tabelul rezervare), iar rezervarea nu e asociată încă cu o locuință -> se va genera eroarea de VALOARE ZERO)

Această inserare se face pentru a se analiza comportamentul excepțiilor.

```
insert into utilizator  
(id_utilizator,nume,prenume,email,data_inregistrare,telefon,gen,data_nasterii)
```

```
values(seq_utilizator.nextval,'Sandu','Anastasia','ana.test@gmail.com',to_date('05/10/2024','dd/mm/yyyy'),null,'feminin',to_date('03/03/2005','dd/mm/yyyy'));
```

```
insert into rezervare values  
(seq_rezervare.nextval,141,to_date('25/01/2025','dd/mm/yyyy'),to_date('25/03/2025','dd/mm/yyyy'),4,'pos');
```

Oracle SQL Developer (C:\Users\Andrei\Desktop\PMF\_FANUL\_25SM\15GB\projectedLog)

```
156
157
158    accept nume_c prompt 'Introduceti numele utilizatorului: '
159    accept prenume_c prompt 'Introduceti prenumele utilizatorului: '
160    declare
161        v_analiza analiza_records;
162        v_tariife tariife;
163        v_rezervari rezervari;
164        v_tipuri tipuri;
165        v_nr_rezervari number :=0;
166    begin
167        v_analiza := analizaRecords(v_tariife, v_rezervari, v_tipuri);
168        v_nr_rezervari := rezervari(v_rezervari);
169        if v_nr_rezervari > 0 then
170            dbms_output.put_line('Nr de rezervari: '||v_nr_rezervari);
171            dbms_output.put_line('Tipuri de locuinte disponibile: ');
172            dbms_output.put_line(v_tipuri);
173            dbms_output.put_line('Tariife disponibile: ');
174            dbms_output.put_line(v_tariife);
175            dbms_output.put_line('Rezervari efectuate: ');
176            dbms_output.put_line(v_rezervari);
177            dbms_output.put_line('Analiza efectuata: ');
178            dbms_output.put_line(v_analiza);
179        else
180            dbms_output.put_line('Nu exista rezervari pentru acest utilizator.');
181            dbms_output.put_line('Valoarea totala e zero, verificati tarifele si rezervarile.');
182        end if;
183    end;
```

PL/SQL procedure successfully completed.

SQL\*Plus: Release 19.0.0.0.0 - Production on Mar 10 2023 10:45:22

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

ANALIZA PENTRU UTILIZATORUL: SANU ANASTASIA

-----

Numar modalitati distincte de plată:

1 Valoare totala platita;

2 Tipuri frequent inchiriat;

3 Total nr de inchirieri;

4 Tipuri disponibile;

5 Tipuri de locuinte rezervate pana acum;

Lista tipuri de locuinte nu a fost initializata.

PENTRU Sandu Anastasia (ii asociem rezervarea cu o locuinta -> se va genera eroarea de modalitate invalida de plata)

Această inserare se face pentru a se analiza comportamentul exceptiilor.

```
insert into se_asociaza_la values(250,1043,to_date('20/01/2025','dd/mm/yyyy'),'activa');
```

PENTRU Dobrinoiu Mari -> TOO MANY ROWS

Aceste inserări se fac pentru a se analiza comportamentul exceptiilor.

insert into utilizator

(id utilizator,nume,prenume,email,data\_inregistrare,telefon,gen,data\_nasterii)

```
values(seq_utilizator.nextval,'Dobrinoiu','Mari','dmari.test@gmail.com',to_date('10/10/2023','dd/mm/yyyy'),null,'feminin',to_date('12/03/1990','dd/mm/yyyy'));
```

insert into utilizator

(id\_utilizator,nume,prenume,email,data\_inregistrare,telefon,gen,data\_nasterii)

```
values(seq_utilizator.nextval,'Dobrinoiu','Mari','dmari2.test@gmail.com',to_date('09/10/2023','dd/mm/yyyy'),null,'masculin',to_date('15/06/1988','dd/mm/yyyy'));
```

Analizăm tabelul de erori și observăm noile intrări care au apărut

Oracle SQL Developer : C:\Users\Andra\Desktop\FMI\ANUL 2\SEM 1\SGBD\project\ex9.sql

File Edit View Navigate Run Source Tools Window Help

SQL Worksheet: History

Query Builder

```
257  
258 delete from rezervare  
259 where modalitate_plata = 'pos';  
260  
261  
262 select * from erori;  
263  
264
```

Script Output x Query Result x

All rows Fetched: 22 in 0.007 seconds

ID_ERORARE	ERORARE	DATA_ERARE	USER_APARE
1	1VA aparut o eroare neasteptata!	15-DEC-24 07.56.26.625000000	PM raport venituri pe categorii
11	11Valoarea parametrului este gresita!!!	15-DEC-24 07.57.18.777000000	PM rezervare maxima
12	12Valoarea parametrului este gresita!!!	15-DEC-24 07.57.57.228000000	PM raport venituri pe categorii
13	13Valoarea parametrului este gresita!!!	15-DEC-24 07.58.11.592000000	PM raport venituri pe categorii
14	21Parametrul p an nu poate fi NULL!	08-JAN-25 03.57.16.679000000	PM raport venituri pe categorii
15	22Parametrul p an nu poate fi NULL!	08-JAN-25 03.57.34.289000000	PM raport venituri pe categorii
16	23Valoarea parametrului este gresita!!!	08-JAN-25 03.58.26.542000000	PM raport venituri pe categorii
17	24Suntem mai multe rezervari cu aceeasi valoare maxima!!!	08-JAN-25 04.08.29.688000000	PM rezervare maxima
18	25In tota introducere nu s-au inregistrat rezervari!!!	08-JAN-25 04.09.07.292000000	PM rezervare maxima
19	26Sunt mai multe rezervari cu aceeasi valoare maxima!!!	08-JAN-25 04.10.33.743000000	PM rezervare maxima
20	27Suntem mai multe rezervari cu aceeasi valoare maxima!!!	08-JAN-25 04.11.48.540000000	PM rezervare maxima
21	28In tota introducere nu s-au inregistrat rezervari!!!	08-JAN-25 04.12.04.540000000	PM rezervare maxima
22	29Nu s-a gasit tara in baza de date!	08-JAN-25 04.13.23.465000000	PM rezervare maxima
23	30Valoarea parametrului este gresita!!!	08-JAN-25 04.13.55.908000000	PM rezervare maxima
24	31Nu s-a gasit tara in baza de date!	08-JAN-25 04.14.44.414000000	PM rezervare maxima
25	42Utilizatorul exista,dar nu are sau nu a avut rezervari active.	09-JAN-25 10.51.51.853000000	AM analiza rezervari utilizator
43	43Nu s-a gasit datele cautate!	09-JAN-25 10.51.59.783000000	AM analiza rezervari utilizator
27	44Nu s-a gasit datele cautate!	09-JAN-25 10.54.27.160000000	AM analiza rezervari utilizator
28	45Utilizatorul exista,dar nu are sau nu a avut rezervari active.	09-JAN-25 10.55.27.800000000	AM analiza rezervari utilizator
29	46Utilizatorul exista,dar nu are sau nu a avut rezervari active.	09-JAN-25 10.58.00.894000000	AM analiza rezervari utilizator
30	47Valoarea totala e zero,verificati tarifele si rezervarile.	09-JAN-25 11.00.33.818000000	AM analiza rezervari utilizator
31	48-a inregistrat o modalitate invalida de plata,verificati inregistrarile.	09-JAN-25 11.05.16.269000000	AM analiza rezervari utilizator
32	49Există mai mulți utilizatori cu același nume.Verificati baza de date.	09-JAN-25 11.08.26.396000000	AM analiza rezervari utilizator

Output x Buffer Size:20000

Click on an identifier with the Control key down to perform "Go to Declaration".

6°C Mostly cloudy

Search

11:11 AM 1/9/2025

## **10. EXERCITIUL 10 – TRIGGER LMD – LA NIVEL DE COMANDĂ**

### **ENUNT**

Să se implementeze un mecanism de audit pentru tabela rezervare, care să monitorizeze toate modificările efectuate asupra acesteia (INSERT, UPDATE, DELETE). Acest mecanism va fi implementat folosind un trigger AFTER INSERT OR UPDATE OR DELETE și va salva informațiile relevante într-o tabelă separată, denumită rezervari\_audit.

Detaliile care trebuie înregistrate în tabela rezervari\_audit sunt:

- Tipul comenzi efectuate (INSERT, UPDATE, DELETE).
- Utilizatorul care a efectuat modificarea (SYS\_CONTEXT('USERENV', 'SESSION\_USER')).
- Data și ora comenzi (SYSTIMESTAMP).
- Numărul de rânduri (afectate).

```
create sequence seq_audit_r
```

```
increment by 1
```

```
start with 1
```

```
nocycle;
```

```
create table rezervari_audit(  
    id_audit number,  
    comanda varchar2(15),  
    utilizator varchar2(40),  
    data_comanda timestamp default current_timestamp,  
    randuri_afectate number  
);
```

```
drop table rezervari_audit;
```

```
drop sequence seq_audit_r;

create or replace trigger trig_ex10
    after insert or update or delete on rezervare
declare
    nr_randuri number := 0;
    t_comanda varchar2(15);
begin
    if INSERTING then
        t_comanda := 'INSERT';
        select count(*) into nr_randuri
        from rezervare
        where rowid in (select rowid from rezervare);
    elsif UPDATING then
        t_comanda := 'UPDATE';
        select count(*) into nr_randuri
        from rezervare
        where rowid in (select rowid from rezervare);
    elsif DELETING then
        t_comanda := 'DELETE';
        select count(*) into nr_randuri
        from rezervare;
    end if;
```

insert into rezervari\_audit

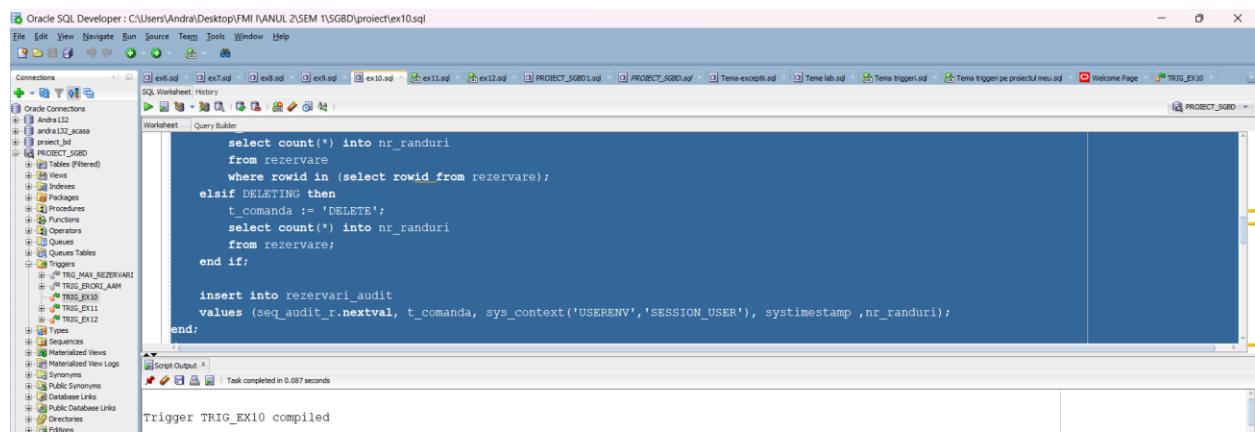
```

values (seq_audit_r.nextval, t_comanda, sys_context('USERENV','SESSION_USER'),
systimestamp ,nr_randuri);

```

end;

/



```

select count(*) into nr_randuri
from rezervare
where rowid in (select rowid from rezervare);
elsif DELETING then
  t_comanda := 'DELETE';
  select count(*) into nr_randuri
  from rezervare;
end if;

insert into rezervari_audit
values (seq_audit_r.nextval, t_comanda, sys_context('USERENV','SESSION_USER'), systimestamp ,nr_randuri);
end;
Trigger TRIG_EX10 compiled

```

## TESTARE

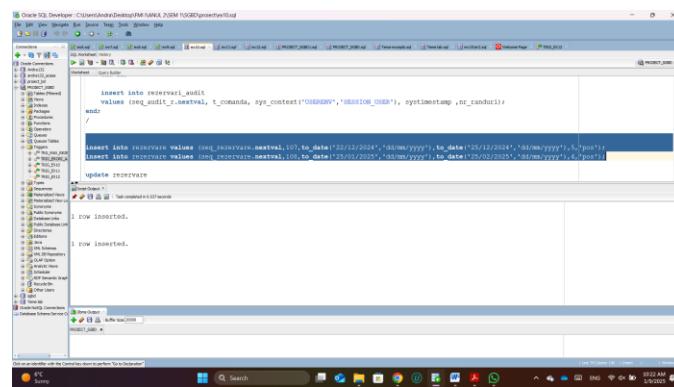
Vom avea două noi rezervări care au modalitatea de plată “pos”.

insert into rezervare values

(seq\_rezervare.nextval,107,to\_date('22/12/2024','dd/mm/yyyy'),to\_date('25/12/2024','dd/mm/yyyy'),5,'pos');

insert into rezervare values

(seq\_rezervare.nextval,108,to\_date('25/01/2025','dd/mm/yyyy'),to\_date('25/02/2025','dd/mm/yyyy'),4,'pos');



```

insert into rezervare values (seq_rezervare.nextval,107,to_date('22/12/2024','dd/mm/yyyy'),to_date('25/12/2024','dd/mm/yyyy'),5,'pos');

insert into rezervare values (seq_rezervare.nextval,108,to_date('25/01/2025','dd/mm/yyyy'),to_date('25/02/2025','dd/mm/yyyy'),4,'pos');

update rezervare
set t_comanda = 'POS'
where rowid = (select rowid from rezervare where seq_rezervare.nextval = 107);

1 row inserted.

1 row inserted.

1 row inserted.

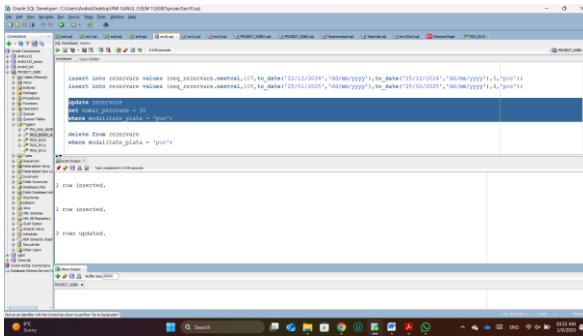
Trigger TRIG_EX10 compiled

```

update rezervare

```
set numar_persoane = 10
```

```
where modalitate_plata = 'pos';
```



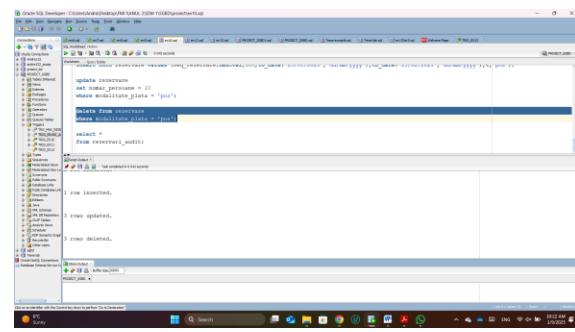
```
insert into rezervare values (no_rezervare.nextval,10,to_date('22/12/2024','dd/mm/yyyy'),to_date('25/12/2024','dd/mm/yyyy'),1,'pos')
insert into rezervare values (no_rezervare.nextval,10,to_date('25/12/2024','dd/mm/yyyy'),to_date('25/01/2025','dd/mm/yyyy'),1,'pos')
update rezervare
set numar_persoane = 10
where modalitate_plata = 'pos'

delete from rezervare
where modalitate_plata = 'pos'

1 row inserted.
1 row updated.
3 rows deleted.
```

```
delete from rezervare
```

```
where modalitate_plata = 'pos';
```



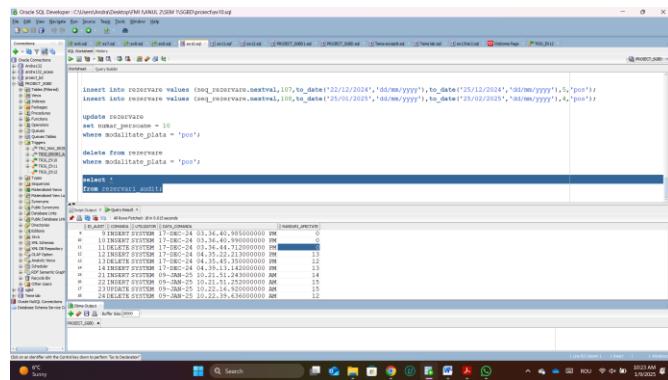
```
update rezervare
set numar_persoane = 10
where modalitate_plata = 'pos'

delete from rezervare
where modalitate_plata = 'pos'

select *
from rezervari_audit

1 row inserted.
3 rows updated.
3 rows deleted.
```

### Analizăm tabelul de audit



```
insert into rezervare values (no_rezervare.nextval,10,to_date('22/12/2024','dd/mm/yyyy'),to_date('25/12/2024','dd/mm/yyyy'),1,'pos')
insert into rezervare values (no_rezervare.nextval,10,to_date('25/12/2024','dd/mm/yyyy'),to_date('25/01/2025','dd/mm/yyyy'),1,'pos')

update rezervare
set numar_persoane = 10
where modalitate_plata = 'pos'

delete from rezervare
where modalitate_plata = 'pos'

select *
from rezervari_audit

1 row inserted.
1 row updated.
3 rows deleted.

24 rows selected.
```

Observăm că au apărut 2 insert-uri, care au făcut să crească numărul de rânduri. și totodată apar comenzi Update și Delete. La update numărul rămâne același în tabela rezervări, dar la delete observăm că scade cu 3 pentru că mai aveam încă o înregistrare existentă cu "pos".

## **11. EXERCITIUL 11- TRIGGER LMD – LA NIVEL DE LINIE**

### **ENUNT**

Să se implementeze un mecanism care să actualizeze automat discount-ul unui utilizator pe baza numărului total de zile rezervate. Acest mecanism trebuie implementat printr-un trigger BEFORE UPDATE OR INSERT pe tabela utilizator și trebuie să îndeplinească următoarele:

- Discount-ul unui utilizator se recalculează în funcție de numărul total de zile rezervate, luând în calcul doar rezervările active:
  - ➔ 5% pentru 10-25 zile rezervate.
  - ➔ 10% pentru 26-50 zile rezervate.
  - ➔ 20% pentru mai mult de 50 zile rezervate.
  - ➔ 0% pentru mai puțin de 10 zile rezervate.
- Triggerul trebuie să fie activat în următoarele cazuri:
  - ➔ Când se înregistrează o nouă rezervare asociată utilizatorului.
  - ➔ Când se adaugă un nou utilizator în baza de date.
  - ➔ Când se modifică datele unei rezervări (de exemplu, datele sau numărul de persoane).
- Dacă discount-ul calculat diferă de discount-ul existent sau dacă discount-ul existent este NULL, acesta trebuie actualizat.

Toate modificările asupra câmpului discount trebuie înregistrate într-o tabelă de audit denumită discount\_u\_audit, care va reține:

- id\_audit: ID unic pentru fiecare intrare (generat automat cu o secvență).
- id\_utilizator: ID-ul utilizatorului al cărui discount a fost modificat.
- old\_discount: Valoarea veche a discount-ului.
- new\_discount: Valoarea nouă a discount-ului.
- data\_modificare: Data și ora modificării.

În vederea rezolvării, tabela utilizator trebuie modificată pentru a include o coloană discount .

=====TABELA AUDIT=====

```
create sequence seq_audit_u
```

```
increment by 1
```

```
start with 1
```

```
nocycle;
```

```
create table discount_u_audit(
```

```
    id_audit number,
```

```
    id_utilizator number,
```

```
    old_discount number,
```

```
    new_discount number,
```

```
    data_modificare timestamp
```

```
);
```

=====MODIFICARE TABELA UTILIZATOR=====

```
alter table utilizator add (discount number);
```

```
select *
```

```
from utilizator;
```

```
update utilizator
```

```
set discount = 0;
```

=====TRIGGER=====

create or replace trigger trig\_ex11

before update or insert on utilizator

for each row

declare

durata\_rezervari number := 0;

discount\_actual number;

discount\_vechi number;

begin

select nvl(sum(r.data\_sfarsit-r.data\_inceput),0) into durata\_rezervari

from rezervare r

join se\_asociaza\_la s on r.id\_rezervare = s.id\_rezervare

where id\_utilizator = :new.id\_utilizator

and s.status = 'activa';

if durata\_rezervari between 10 and 25 then

    discount\_actual := 5; --5 la sută

elsif durata\_rezervari between 26 and 50 then

    discount\_actual := 10; --10 la sută

elsif durata\_rezervari > 50 then

    discount\_actual := 20;

else

```

discount_actual := 0;

end if;

discount_vechi := :new.discount;

if (discount_actual <> discount_vechi) or (discount_vechi is null) then

:new.discount := discount_actual;

insert into discount_u_audit values
(seq_audit_u.nextval,:new.id_utilizator,discount_vechi,discount_actual,systimestamp);

end if;

end;
/

```

The screenshot shows the Oracle SQL Developer interface. The central workspace displays the PL/SQL code. The status bar at the bottom right indicates that the trigger 'TRIG\_EX11' has been successfully compiled.

## TESTARE:

\*după alter tabelul utilizator arată aşa

```
insert into discount_u_audit values (seq_audit_u.nextval,:new.id_utilizator,discount_vechi,discount_actual,sysdate)
end if;
end

select *
from utilizator;
```

ID	Nume	Prenume	Email	Parola	Sex	Data Nasterii
1	Andrata Andris	andris.andrata4@yahoo.com	23-AFM-23	072231155	female	26-FEB-04 (null)
2	Andrata Irina	irina.andrata4@gmail.com	05-DBR-13	072230575	(null)	21-JUN-77 (null)
3	Andrata Mihai	mihai.andrata4@yahoo.com	10-MAY-14	072230576	male	05-JUL-00 (null)
4	Andrata Mihai	mihai.andrata4@yahoo.com	10-MAY-14	072230576	male	05-JUL-00 (null)
5	Vasile Mihai	m.vasile1998@gmail.com	17-APR-15	072230577	(null)	14-MAR-99 (null)
6	Popescu Ion	ion.popescu1984@yahoo.com	21-SEP-16	072230578	male	19-MAY-94 (null)
7	Firiciu Alex	firiciu2019@gmail.com	21-FEB-22	072230579	male	19-NOV-03 (null)
8	Tomaru Alessandra	alessandra.tomaru.test@gmail.com	20-NOV-24	072230580	female	10-NOV-05 (null)

\*facem update la discount

update utilizator

set discount = discount

where id\_utilizator in(select id\_utilizator

from utilizer);

```

delete from UTILIZATOR
where name = 'Thomar';

update UTILIZATOR
set discount = discount
where id_utilizator in(select id_utilizator
from UTILIZATOR);

Table UTILIZATOR altered.

Table UTILIZATOR altered.

10 rows updated.

SQL>

```

Session 2 Query Result:

```

SQL> select * from UTILIZATOR;
SQL> select * from reservation;

```

id_utilizator	name	email	password	date	discount	status
1	Irina	irina.andruita@yahoo.com	1234567890	2023-01-15	0.15	available
2	Irina	irina.andruita@gmail.com	02-DEC-13	gt22509754	0.15	available
3	Maria	mancutu24@yahoo.com	10-MAY-24	(null)	0.2	available
4	Maria	mancutu24@gmail.com	10-MAY-24	gt22509754	0.2	available
5	Maria	mancutu24@gmail.com	17-APR-15	(null)	0.2	available
6	Ion	ion.pod2010@yahoo.com	20-SEP-14	(null)	0.2	available
7	Ion	ion.pod2010@gmail.com	21-SEP-14	(null)	0.2	available
8	Alexandru	alexandru.alexandrut@gmail.com	20-DEC-24	(null)	0.2	available
9	Alexandru	alexandru.alexandrut@gmail.com	20-DEC-24	(null)	0.2	available

Acum tabelul discount\_u\_audit arată aşa

```

insert into reservation values (seq_reservation.nextval,100,to_date('25/01/2025','dd/mm/yyyy'),to_date('25/03/2025','dd/mm/yyyy'),4,'post');
update UTILIZATOR;

SQL>

```

Session 2 Query Result:

```

SQL> select * from reservation;

```

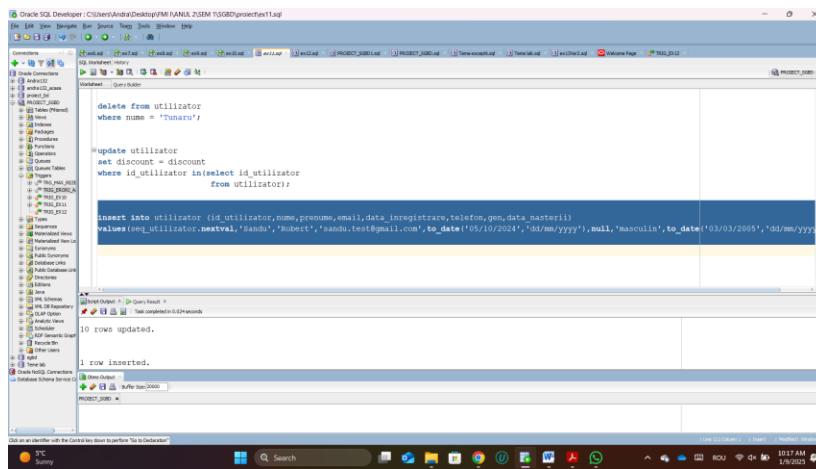
id_reservation	id_utilizator	date_start	date_end	type	status
1	7	106	0	0	available
2	21	105	0	509-DMN-25 10:00:09.760000000 AM	available
3	22	107	0	509-DMN-25 10:00:09.760000000 AM	available
4	24	105	0	509-DMN-25 10:00:42.023000000 AM	available
5	24	107	0	509-DMN-25 10:00:42.023000000 AM	available
6	25	108	0	509-DMN-25 10:00:47.670000000 AM	available
7	30	104	0	509-DMN-25 10:00:44.040000000 AM	available
8	32	100	0	009-DMN-25 10:11:34.586000000 AM	available
9	34	102	0	009-DMN-25 10:11:34.586000000 AM	available
10	36	104	0	009-DMN-25 10:11:34.586000000 AM	available
11	37	105	0	009-DMN-25 10:11:34.586000000 AM	available
12	38	106	0	009-DMN-25 10:11:34.587000000 AM	available
13	39	106	0	009-DMN-25 10:11:34.587000000 AM	available
14	40	107	0	009-DMN-25 10:11:34.587000000 AM	available
15	41	123	0	009-DMN-25 10:11:34.587000000 AM	available

\*Facem si un insert in utilizator

insert into utilizator

(id\_utilizator,nume,prenume,email,data\_inregistrare,telefon,gen,data\_nasterii)

values(seq\_utilizator.nextval,'Sandu','Robert','sandu.test@gmail.com',to\_date('05/10/2024','dd/mm/yyyy'),null,'masculin',to\_date('03/03/2005','dd/mm/yyyy'));



```
delete from utilizator
where nume = 'Tunaru';

update utilizator
set discount = discount -
1
where id_utilizator in (select id_utilizator
from utilizator);

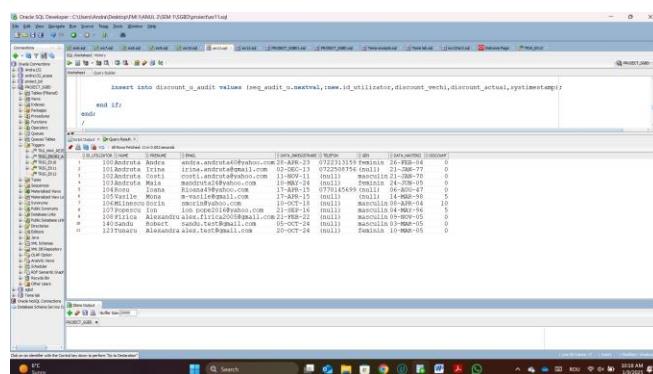
insert into utilizator (id_utilizator, nume, prenume, email, data_inregistrare, telefon, gen, data_nasterii)
values (seq_utilizator.nextval, 'Sandu', 'Robert', 'sandu.test@gmail.com', to_date('05/10/2024', 'dd/mm/yyyy'), null, 'masculin', to_date('03/03/2005', 'dd/mm/yyyy'));

10 rows updated.

1 row inserted.
```

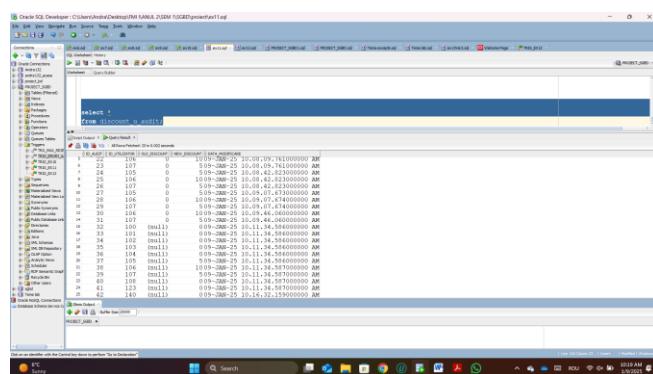
Si vedem ca neavând rezervări inserarea la discount nu e null, ci 0, aşa cum se specifică în trigger.

Şi totodată apare o linie nouă în tabelul de audit.



```
insert into discount_u_audit values (seq_discount_uAudit.nextval,new.id_utilizator,discoun_vechi,discoun_actual,sysdate);

end if;
end;
```



```
select *
from discount_u_audit;
```

## **12. EXERCITIUL 12 – TRIGGER LDD**

### **ENUNT**

Managerii platformei doresc să implementeze un mecanism de protecție și audit pentru baza de date, astfel încât ștergerea unor tabele considerate critice să fie prevenită. În cazul în care un utilizator încearcă să șteargă una dintre aceste tabele, încercarea trebuie:

- Prevenită, prin blocarea operației de ștergere.
- Monitorizată, prin înregistrarea detaliilor într-o tabelă de audit denumită audit\_drop.

Detaliile care trebuie înregistrate în tabela de audit sunt:

- ➔ Utilizatorul care a încercat să șteargă tabelul.
- ➔ Ora și data la care a avut loc încercarea.
- ➔ Numele tabelului protejat pe care s-a încercat ștergerea.

Tabelele protejate: Tabelele care trebuie protejate împotriva ștergerii sunt:

- ➔ DISCOUNT\_U\_AUDIT
- ➔ REZERVARI\_AUDIT
- ➔ ERORI

=====TABELA AUDIT=====

```
create sequence seq_audit_drop
increment by 1
start with 1
nocycle;
```

```
create table audit_drop(
    id_audit number,
    utilizator varchar2(60),
    data_incercare timestamp,
```

```
tabel varchar2(30)  
);
```

-----LISTA TABELE PROTEJATE IN VARRAY-----

```
create or replace type tabele_protejate as varray(10) of varchar2(30);  
/
```

```
create or replace trigger trig_ex12
```

```
before drop on schema
```

```
declare
```

```
    tab varchar2(30);
```

```
    tabele_protej tabele_protejate := tabele_protejate('DISCOUNT_U_AUDIT',  
'REZERVARI_AUDIT', 'ERORI');
```

```
begin
```

```
    tab := sys.dictionary_obj_name;
```

```
for i in 1..tabel_protej.count loop
```

```
    if upper(tab) = upper(tabel_protej(i)) then
```

```
        -- Înregistrează încercarea în tabelul de audit
```

```
        insert into audit_drop values (seq_audit_drop.nextval,
```

```
                                SYS_CONTEXT('USERENV','SESSION_USER'),
```

```
                                systimestamp,
```

```
                                tab);
```

```
        -- Afisează mesajul și ridică o eroare pentru a preveni ștergerea
```

```
        dbms_output.put_line('Stergerea tabelului ' || tab || ' este interzisa deoarece este un tabel  
protejat!');
```

```

        raise_application_error(-20001, 'Stergerea tabelului ' || tab || ' este interzisa!');

    end if;

end loop;

/

```

```

-----LISTA TABLEL PROTEJATE IN VARRAY-----
create or replace type tabelo_protecteaza as varray(10) of varchar2 (30);

/
create or replace trigger TRIG_EX12
before drop on schema
declare
    tab varchar2(30);
    tabelo_protecteaza := tabelo_protecteaza('DISCOUNT_U_AUDIT', 'RESERVARI_AUDIT', 'ENORI', 'TEST');
begin
    for i in 1..4 loop
        tab := sys.dictionary_obj.name;
        if tab in tabelo_protecteaza then
            raise_application_error(-20001, 'Stergerea tabelului ' || tab || ' este interzisa!');
        end if;
    end loop;
end;
Trigger TRIG_EX12 compiled

```

## TESTARE:

Încercăm să dăm drop la un tabel din cele din lista de protejate.

drop table discount\_u\_audit;

```

drop table discount_u_audit;
drop table discount_u_audit
Error starting at line 14 in command -
drop table discount_u_audit
ORA-04018: error during execution of trigger 'TRIG_EX12'
ORA-04018: error during execution of trigger 'TRIG_EX12' at level 1
ORA-20001: Stergerea tabelului DISCOUNT_U_AUDIT este interzisa!
ORA-04012 at line 14
ORA-04003: error during execution of trigger "trig_ex12"
ORA-04003: A routine error occurred during execution of a trigger.
Cause: A routine error occurred during execution of a trigger.
Action: Check the triggers which were involved in the operation.

```

Facem un select și observăm că nu s-a realizat drop-ul.

select \*

from discount\_u\_audit;

ID	NR_PIESE	DATA_CEDARE	DATA_MARFA
1	100	0	10.12.2012
2	100	0	10.12.2012
3	100	0	10.12.2012

stergeaza tabelulul DISCOUNT\_U\_AUDIT este interzisa deoarece este un tabel protejat!

## **13. EXERCITIUL 13 – PACHET + DATE COMPLEXE**

### **ENUNT**

Managerii doresc să se implementeze un sistem integrat de analiză și raportare pentru platforma de închiriere, care să faciliteze:

- Monitorizarea veniturilor generate de rezervări.
- Analizarea performanței gazdelor.
- Generarea de rapoarte financiare și statistice detaliate.

Sistemul trebuie să fie realizat sub forma unui pachet PL/SQL, capabil să gestioneze date complexe și să ofere rapoarte structurate, bazate pe datele din platformă. Pachetul va include mecanisme care să producă următoarele rapoarte:

- Raport anual al veniturilor:
  - ➔ Veniturile generate pentru fiecare lună dintr-un an specific.
  - ➔ Venitul total anual.

- Raport al rezervărilor dintr-un an specific:
  - ➔ Lista rezervărilor efectuate, incluzând:

ID-ul rezervării.

Utilizatorul care a efectuat rezervarea.

Locuința rezervată.

Perioada rezervării.

Costul total al rezervării.

- ➔ Totalul veniturilor generate din rezervări pentru anul respectiv.

- Clasamentul gazdelor de top:
  - ➔ Primele 3 gazde cu cele mai mari venituri pentru un an specific.
  - ➔ Pentru fiecare gazdă: Venitul total generat. Lista locuințelor rezervate și detalii despre acestea.

```
CREATE OR REPLACE PACKAGE pachetEx13 AS
```

```
-- Tipuri de date complexe
```

```
TYPE record_rezervari IS RECORD (
    id_rezervare NUMBER,
    id_utilizator NUMBER,
    id_locuinta NUMBER,
    data_inceput DATE,
    data_sfarsit DATE,
    cost_total NUMBER
);
```

```
TYPE lista_rezervari IS TABLE OF record_rezervari;
```

```
TYPE record_locuinta IS RECORD (
    id_locuinta NUMBER,
    nume_locuinta VARCHAR2(70),
    venit_locuinta NUMBER
);
```

```
TYPE lista_locuinte IS TABLE OF record_locuinta;
```

```
TYPE record_gazda IS RECORD (
    id_gazda NUMBER,
```

```
    venit_total NUMBER,  
    nume_gazda VARCHAR2(50),  
    prenume_gazda VARCHAR2(50),  
    locuinte_rezervate lista_locuinte  
);
```

```
TYPE lista_gazde IS TABLE OF record_gazda;
```

```
-- Functii
```

```
FUNCTION calculVenitLunar(luna VARCHAR2, year VARCHAR2) RETURN NUMBER;  
FUNCTION calculVenitAnual(year VARCHAR2) RETURN NUMBER;
```

```
-- Proceduri
```

```
PROCEDURE raportAnual(year VARCHAR2);  
PROCEDURE raportRezervariAn(year VARCHAR2);  
PROCEDURE top3Gazde(year VARCHAR2);
```

```
END pachetEx13;
```

```
/
```

```
CREATE OR REPLACE PACKAGE BODY pachetEx13 AS
```

```
rezervari lista_rezervari;
```

```

venituri_anuale NUMBER := 0;

FUNCTION calculVenitLunar(luna VARCHAR2, year VARCHAR2) RETURN NUMBER
IS
    total_luna NUMBER := 0;
BEGIN
    FOR rec IN (
        SELECT r.id_rezervare,
               SUM(r.numar_persoane * t.pret_per_persoana) + NVL(SUM(d.cost_suplimentar),
0) AS cost_total_rezervare
        FROM rezervare r
        JOIN se_asociaza_la s ON r.id_rezervare = s.id_rezervare
        JOIN locuinta l ON s.id_locuinta = l.id_locuinta
        JOIN tipuri_locuinte tip ON tip.id_tip = l.id_tip
        JOIN tarife t ON tip.id_tarif = t.id_tarif
        LEFT JOIN are a ON l.id_locuinta = a.id_locuinta
        LEFT JOIN dotari d ON a.id_dotare = d.id_dotare
        WHERE TRIM(TO_CHAR(data_inceput, 'Month')) = luna
              AND TO_CHAR(data_inceput, 'YYYY') = year
        GROUP BY r.id_rezervare
    ) LOOP
        total_luna := total_luna + rec.cost_total_rezervare;
    END LOOP;
    RETURN total_luna;
END calculVenitLunar;

```

```

FUNCTION calculVenitAnual(year VARCHAR2) RETURN NUMBER IS
    total_an NUMBER := 0;
    venit_lunar NUMBER;
    luna VARCHAR2(9); -- Numerele lunii

    BEGIN

        FOR i IN 1..12 LOOP
            luna := TRIM(TO_CHAR(TO_DATE(i, 'MM'), 'Month'));

            venit_lunar := calculVenitLunar(luna, year);

            total_an := total_an + venit_lunar;

        END LOOP;

        RETURN total_an;
    END calculVenitAnual;

```

```

-- Procedura raportAnual

PROCEDURE raportAnual(year VARCHAR2) IS
    total_an NUMBER := 0;
    venit_lunar NUMBER;
    luna VARCHAR2(9); -- Numerele lunii

    BEGIN

```

```
DBMS_OUTPUT.PUT_LINE('=====');
```

```
DBMS_OUTPUT.PUT_LINE('Raport anual pentru anul ' || year || ':');
```

```
DBMS_OUTPUT.PUT_LINE('=====');
```

```
FOR i IN 1..12 LOOP
```

```
    luna := TRIM(TO_CHAR(TO_DATE(i, 'MM'), 'Month'));
```

```
    venit_lunar := calculVenitLunar(luna, year);
```

```
    DBMS_OUTPUT.PUT_LINE('   '||i||'.Luna ' || luna || ': Venit ' || venit_lunar);
```

```
    total_an := total_an + venit_lunar;
```

```
END LOOP;
```

```
DBMS_OUTPUT.PUT_LINE(");
```

```
DBMS_OUTPUT.PUT_LINE('    Venit total anual: ' || total_an);
```

```
END reportAnual;
```

```
-- Procedura reportRezervariAn
```

```
PROCEDURE reportRezervariAn(year VARCHAR2) IS
```

```
    rezervari_an lista_rezervari := lista_rezervari();
```

```
    total_rezervari NUMBER := 0;
```

```
BEGIN
```

```

FOR rec IN (
    SELECT r.id_rezervare,
           r.id_utilizator,
           l.id_locuinta,
           r.data_inceput,
           r.data_sfarsit,
           (SUM(r.numar_persoane * t.pret_per_persoana) + NVL(SUM(d.cost_suplimentar),
0)) AS cost_total_rezervare
  FROM rezervare r
  JOIN se_asociaza_la s ON r.id_rezervare = s.id_rezervare
  JOIN locuinta l ON s.id_locuinta = l.id_locuinta
  JOIN tipuri_locuinte tip ON tip.id_tip = l.id_tip
  JOIN tarife t ON tip.id_tarif = t.id_tarif
  LEFT JOIN are a ON l.id_locuinta = a.id_locuinta
  LEFT JOIN dotari d ON a.id_dotare = d.id_dotare
 WHERE TO_CHAR(r.data_inceput, 'YYYY') = year
 GROUP BY r.id_rezervare, r.id_utilizator, l.id_locuinta, r.data_inceput, r.data_sfarsit
) LOOP

```

```

rezervari_an.EXTEND;
rezervari_an(rezervari_an.COUNT) := record_rezervari(
    rec.id_rezervare,
    rec.id_utilizator,
    rec.id_locuinta,
    rec.data_inceput,

```

```

        rec.data_sfarsit,
        rec.cost_total_rezervare
    );
    total_rezervari := total_rezervari + rec.cost_total_rezervare;
END LOOP;

DBMS_OUTPUT.PUT_LINE('=====
=====');
DBMS_OUTPUT.PUT_LINE('Raport rezervari pentru anul ' || year || ':');

DBMS_OUTPUT.PUT_LINE('=====
=====');
FOR i IN 1..rezervari_an.COUNT LOOP
    DBMS_OUTPUT.PUT_LINE('Rezervare ID: ' || rezervari_an(i).id_rezervare ||
        ', Utilizator ID: ' || rezervari_an(i).id_utilizator ||
        ', Locuinta ID: ' || rezervari_an(i).id_locuinta ||
        ', Perioada: ' || rezervari_an(i).data_inceput || ' - ' ||
        rezervari_an(i).data_sfarsit ||
        ', Cost total: ' || rezervari_an(i).cost_total);
END LOOP;

DBMS_OUTPUT.PUT_LINE(");
DBMS_OUTPUT.PUT_LINE('      Total venit din rezervari in acest an: ' || total_rezervari);
END reportRezervariAn;

```

```

PROCEDURE top3Gazde(year VARCHAR2) IS
    gazde lista_gazde := lista_gazde();
    venit_gazda NUMBER;
    nume_gazda VARCHAR2(100);
    prenume_gazda VARCHAR2(100);
    locuinte_rezervate lista_locuinte := lista_locuinte();

BEGIN
    FOR rec IN (
        SELECT l.id_gazda,
               SUM(r.numar_persoane * t.pret_per_persoana + NVL(d.cost_suplimentar, 0)) AS
               venit_total
        FROM locuinta l
        JOIN se_asociaza_la s ON l.id_locuinta = s.id_locuinta
        JOIN rezervare r ON s.id_rezervare = r.id_rezervare
        LEFT JOIN tipuri_locuinte tip ON tip.id_tip = l.id_tip
        LEFT JOIN tarife t ON tip.id_tarif = t.id_tarif
        LEFT JOIN are_a ON l.id_locuinta = a.id_locuinta
        LEFT JOIN dotari d ON a.id_dotare = d.id_dotare
        WHERE TO_CHAR(r.data_inceput, 'YYYY') = year
        GROUP BY l.id_gazda
        ORDER BY venit_total DESC
        FETCH FIRST 3 ROWS ONLY
    )
    loop
        nume_gazda := rec.nume_gazda;
        prenume_gazda := rec.prenume_gazda;
        locuinte_rezervate := rec.locuinte_rezervate;
        venit_gazda := rec.venit_total;
        exit;
    end loop;
end;

```

) LOOP

```
venit_gazda := rec.venit_total;
```

```
SELECT nume, prenume INTO nume_gazda, prenume_gazda  
FROM gazda  
WHERE id_gazda = rec.id_gazda;
```

FOR locuinta IN (

```
    SELECT l.id_locuinta, l.nume_locuinta  
    FROM locuinta l  
    JOIN se_asociaza_la s ON l.id_locuinta = s.id_locuinta  
    JOIN rezervare r ON s.id_rezervare = r.id_rezervare  
    WHERE l.id_gazda = rec.id_gazda  
    AND TO_CHAR(r.data_inceput, 'YYYY') = year
```

) LOOP

```
locuinte_rezervate.EXTEND;  
locuinte_rezervate(locuinte_rezervate.COUNT) := record_locuinta(  
    locuinta.id_locuinta,  
    locuinta.nume_locuinta,  
    0  
);
```

```

END LOOP;

gazde.EXTEND;

gazde(gazde.COUNT) := record_gazda(
    rec.id_gazda,
    venit_gazda,
    nume_gazda,
    prenume_gazda,
    locuinte_rezervate
);

locuinte_rezervate.DELETE;

END LOOP;

DBMS_OUTPUT.PUT_LINE('-----');

DBMS_OUTPUT.PUT_LINE('Top 3 gazde pentru anul ' || year || ':');

DBMS_OUTPUT.PUT_LINE('-----');

FOR i IN 1..LEAST(3, gazde.COUNT) LOOP

    DBMS_OUTPUT.PUT_LINE(");

    DBMS_OUTPUT.PUT_LINE(i||'.Gazda ID: ' || gazde(i).id_gazda ||
        ', Nume: ' || gazde(i).nume_gazda || ' ' || gazde(i).prenume_gazda ||
        ', Venit total: ' || gazde(i).venit_total);

    DBMS_OUTPUT.PUT_LINE('Locuinte rezervate:');

    FOR j IN 1..gazde(i).locuinte_rezervate.COUNT LOOP

```

```

        DBMS_OUTPUT.PUT_LINE(' Locuinta ID: ' ||
gazde(i).locuinte_rezervate(j).id_locuinta ||

        ', Nume locuinta: ' || gazde(i).locuinte_rezervate(j).nume_locuinta);

    END LOOP;

    END LOOP;

END top3Gazde;

END pachetEx13;

/

```

```

END LOOP;
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE('Top 3 gazde pentru anul ' || year || ':');
DBMS_OUTPUT.PUT_LINE('-----');

FOR i IN 1..LEAST(3, gazde.COUNT) LOOP
    DBMS_OUTPUT.PUT_LINE('');
    DBMS_OUTPUT.PUT_LINE('Gazda ID: ' || gazde(i).id_gazda ||
        ', Nume: ' || gazde(i).nume_gazda || ' ' || gazde(i).prenume_gazda ||
        ', Venit total: ' || gazde(i).venit_total);

    DBMS_OUTPUT.PUT_LINE('Locuinte rezervate:');
    FOR j IN 1..gazde(i).locuinte_rezervate.COUNT LOOP
        DBMS_OUTPUT.PUT_LINE(' Locuinta ID: ' || gazde(i).locuinte_rezervate(j).id_locuinta ||
            ', Nume locuinta: ' || gazde(i).locuinte_rezervate(j).nume_locuinta);
    END LOOP;
END LOOP;
END top3gazde;

END;

```

Package Output: Task completed in 0.04 seconds

Package Body PACHETEX13 compiled

## TESTARE

### 1 CALCUL VENIT LUNAR

\*pentru iulie 2024

```

BEGIN
    DBMS_OUTPUT.PUT_LINE('Venitul lunar pentru iulie 2024: ' ||
    pachetEx13.calculVenitLunar('July', '2024'));

END;
/

```

```

BEGIN
    DBMS_OUTPUT.PUT_LINE('Venitul lunar pentru iulie 2024: ' ||
    pachetEx13.calculVenitLunar('July', '2024'));
END;
/

```

Package PACHETEX13 compiled  
Package Body PACHETEX13 compiled  
PL/SQL procedure successfully completed.

Output  
PROJECT\_0001  
Venitul lunar pentru iulie 2024: 780

## 2 CALCUL VENIT ANUAL

\*pentru 2024

```

BEGIN
    DBMS_OUTPUT.PUT_LINE('Venitul anual pentru 2024: ' ||
    pachetEx13.calculVenitAnual('2024'));

END;
/

```

```

BEGIN
    DBMS_OUTPUT.PUT_LINE('Venitul anual pentru 2024: ' ||
    pachetEx13.calculVenitAnual('2024'));
END;
/

```

Package Body PACHETEX13 compiled  
PL/SQL procedure successfully completed.  
PL/SQL procedure successfully completed.

Output  
PROJECT\_0001  
Venitul anual pentru 2024: 6250

\*pentru 2020

BEGIN

```
DBMS_OUTPUT.PUT_LINE('Venitul anual pentru 2020: ' ||  
pachetEx13.calculVenitAnual('2020'));
```

END;

/

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

SQL Output
----- SELECT_000 ----- Venitul lunar pentru luna 2024: 780 Venitul anual pentru 2024: 4250 Venitul anual pentru 2020: 0

### 3 RAPORT ANUAL

\*pentru 2024

BEGIN

```
pachetEx13.raportAnual('2024');
```

END;

/

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

SQL Output
----- Raport anual pentru anul 2024: ----- 1.Luna January Venit 0 2.Luna February Venit 0 3.Luna March Venit 0 4.Luna April Venit 0 5.Luna May Venit 0 6.Luna June Venit 0 7.Luna July Venit 780 8.Luna August Venit 0 9.Luna September Venit 0 10.Luna October Venit 782 11.Luna November Venit 0 12.Luna Decemberi Venit 4470 Venit total anual: 4250

## 4 RAPORT REZERVARI PE AN

\*pentru 2024

BEGIN

```
pachetEx13.raportRezervariAn('2024');
```

END;

/

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there's a tree view of database objects. The main workspace contains the following PL/SQL code:

```
begin
    pachetEx13.raportRezervariAn('2024');
end;
```

Below the code, the message "PL/SQL procedure successfully completed." is displayed. The bottom pane shows the output of the procedure, which is a report titled "Report rezervari pentru anul 2024". The report lists several reservations with their details and total costs:

```
Report rezervari pentru anul 2024
-----
Rezervare ID: 1005, Utilizator ID: 106, Locuinta ID: 300, Perioada: 08-OCT-24 - 14-OCT-24, Cost total: 792
Rezervare ID: 1006, Utilizator ID: 109, Locuinta ID: 250, Perioada: 20-NOV-24 - 26-NOV-24, Cost total: 380
Rezervare ID: 1006, Utilizator ID: 107, Locuinta ID: 200, Perioada: 20-DEC-24 - 03-JAN-25, Cost total: 4678
Rezervare ID: 1009, Utilizator ID: 104, Locuinta ID: 270, Perioada: 11-NOV-24 - 19-NOV-24, Cost total: 400
```

Total venit din rezervari în acest an: 4250

## 5 TOP 3 GAZDE

\*pentru 2024

BEGIN

```
pachetEx13.top3Gazde('2024');
```

END;

/

The screenshot shows the Oracle SQL Developer interface. The main workspace contains the following PL/SQL code:

```
begin
    pachetEx13.top3Gazde('2024');
end;
```

Below the code, the message "PL/SQL procedure successfully completed." is displayed. The bottom pane shows the output of the procedure, which is a report titled "Top 3 gazde pentru anul 2024". The report lists three hosts with their names, room counts, and total stays:

```
Top 3 gazde pentru anul 2024
-----
1.Gazda ID: 10, Nume: Brown James, Venit total: 5058
Locuinte rezervate:
Locuinta ID: 250, Num. locuintai: Vila Trevi
Locuinta ID: 200, Num. locuintai: Grand Hotel

2.Gazda ID: 12, Nume: Morris Steve, Venit total: 792
Locuinte rezervate:
Locuinta ID: 300, Num. locuintai: Nesoddens house

3.Gazda ID: 14, Nume: Rojo Angela, Venit total: 400
Locuinte rezervate:
Locuinta ID: 270, Num. locuintai: Majestic
```