

# Gestiunea aplicației MyMoto

Stoinea Maria-Miruna  
GRUPA 242 An 2

# Cuprins

1. Prezentați pe scurt baza de date (utilitatea ei). ....	2
2. Realizați diagrama entitate-relație (ERD).....	3
3. Pornind de la diagrama entitate-relație realizați <b>diagrama conceptuală</b> a modelului propus, integrând toate atributele necesare: entitățile, relațiile și atributele trebuie definite în limba română.....	4
4. Implementați în Oracle <b>diagrama conceptuală</b> realizată: definiți toate tabelele, definind toate constrângerile de integritate necesare (chei primare, cheile externe etc).....	6
6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind <b>un subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții</b> studiate. Apelați subprogramul. .	65
7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind <b>un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare</b> studiate, unul dintre acestea fiind cursor parametrizat, dependent de celălalt cursor. Apelați subprogramul.....	75
8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind <b>un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele definite</b> . Definiți minim 2 exceptii proprii. Apelați subprogramul astfel încât să evidențiați toate cazurile definite și tratate. .....	80
9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind <b>un subprogram stocat independent de tip procedură care să utilizeze într-o singură comandă SQL 5 dintre tabelele definite</b> . Tratați toate exceptiile care pot apărea, incluzând exceptiile NO_DATA_FOUND și TOO_MANY_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate. .....	86
10. Definiți un <b>trigger de tip LMD la nivel de comandă</b> . Declanșați trigger-ul. ....	96
11. Definiți un <b>trigger de tip LMD la nivel de linie</b> . Declanșați trigger-ul. ....	101
12. Definiți un trigger de tip LDD. Declanșați trigger-ul. ....	105
13. Definiți <b>un pachet care să conțină toate obiectele definite</b> în cadrul proiectului. ....	110
14. Definiți <b>un pachet care să includă tipuri de date complexe și obiecte necesare unui flux de acțiuni integrate</b> , specifice bazei de date definite (minim 2 tipuri de date, minim 2 funcții, minim 2 proceduri). .....	124

**1. Prezentați pe scurt baza de date (utilitatea ei).**

---

## **Temă**

Tema acestui proiect este dat de gestiunea unei baze de date realizată pentru o aplicație mobilă denumită "MyMoto", care are scopul de a monitoriza motocicletele personale ale utilizatorilor și de a reține detalii esențiale despre acestea.

Această bază de date poate fi utilizată pentru a ajuta la planificarea reviziilor și a altor activități de întreținere, pentru a urmări istoricul de service și întreținere al fiecărei motociclete și pentru a gestiona detaliile despre motor, frâne și alte componente specifice.

## **Utilitate**

Utilitatea acestei baze de date este de a îmbunătăți gestionarea motocicletelor personale ale utilizatorilor, oferindu-le acces rapid și ușor la informații detaliate despre motocicletele lor și istoricul acestora, și ajutându-i să-și gestioneze mai eficient întreținerea și costurile.

Aceste informații pot fi utilizate pentru a planifica revizii și alte activități de întreținere, pentru a gestiona costurile de întreținere a motocicletelor și pentru a urmări istoricul de service. În plus, această bază de date poate ajuta la crearea unor rapoarte detaliate despre întreținerea motocicletelor, ceea ce poate fi util în cazul în care un utilizator dorește să vândă sau să cumpere o motocicletă.

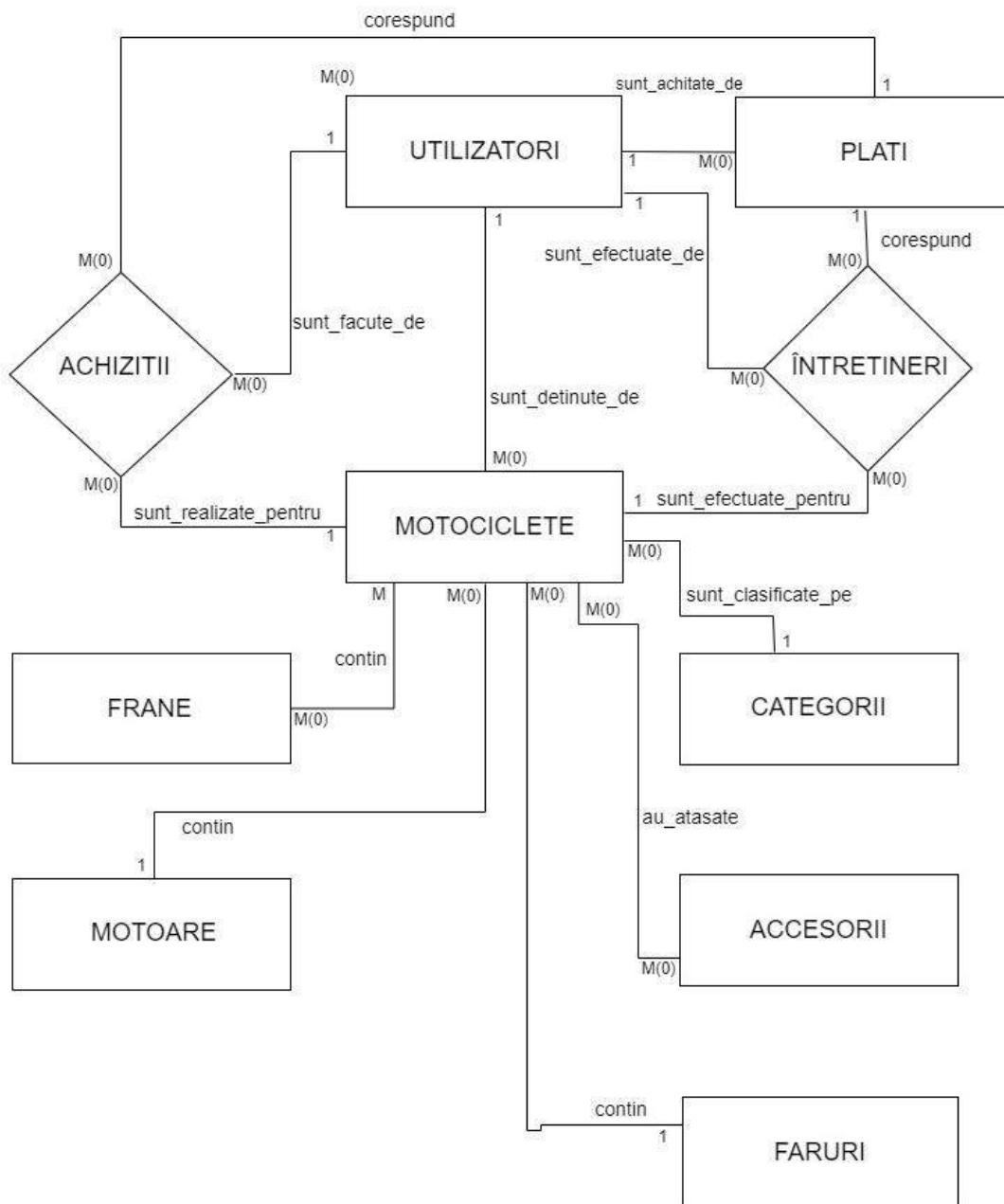
Funcționalitățile acestei baze de date includ stocarea și gestionarea informațiilor despre utilizatori și motociclete, inclusiv informații detaliate despre motor, rezervor și alte componente specifice. De asemenea, aceasta permite adăugarea și actualizarea informațiilor despre întreținerea motocicletelor, precum și generarea de rapoarte despre istoricul de service și întreținere al acestora.

Prin urmare, această bază de date este foarte utilă pentru utilizatorii care dețin motociclete personale, oferindu-le acces rapid la informații detaliate despre motocicletele lor și istoricul acestora. Acest lucru poate ajuta la planificarea reviziilor și a altor activități de întreținere, menținând motocicletele în condiții optime și reducând costurile generale de întreținere.

**2. Realizați diagrama entitate-relație (ERD): entitățile, relațiile și atributele**

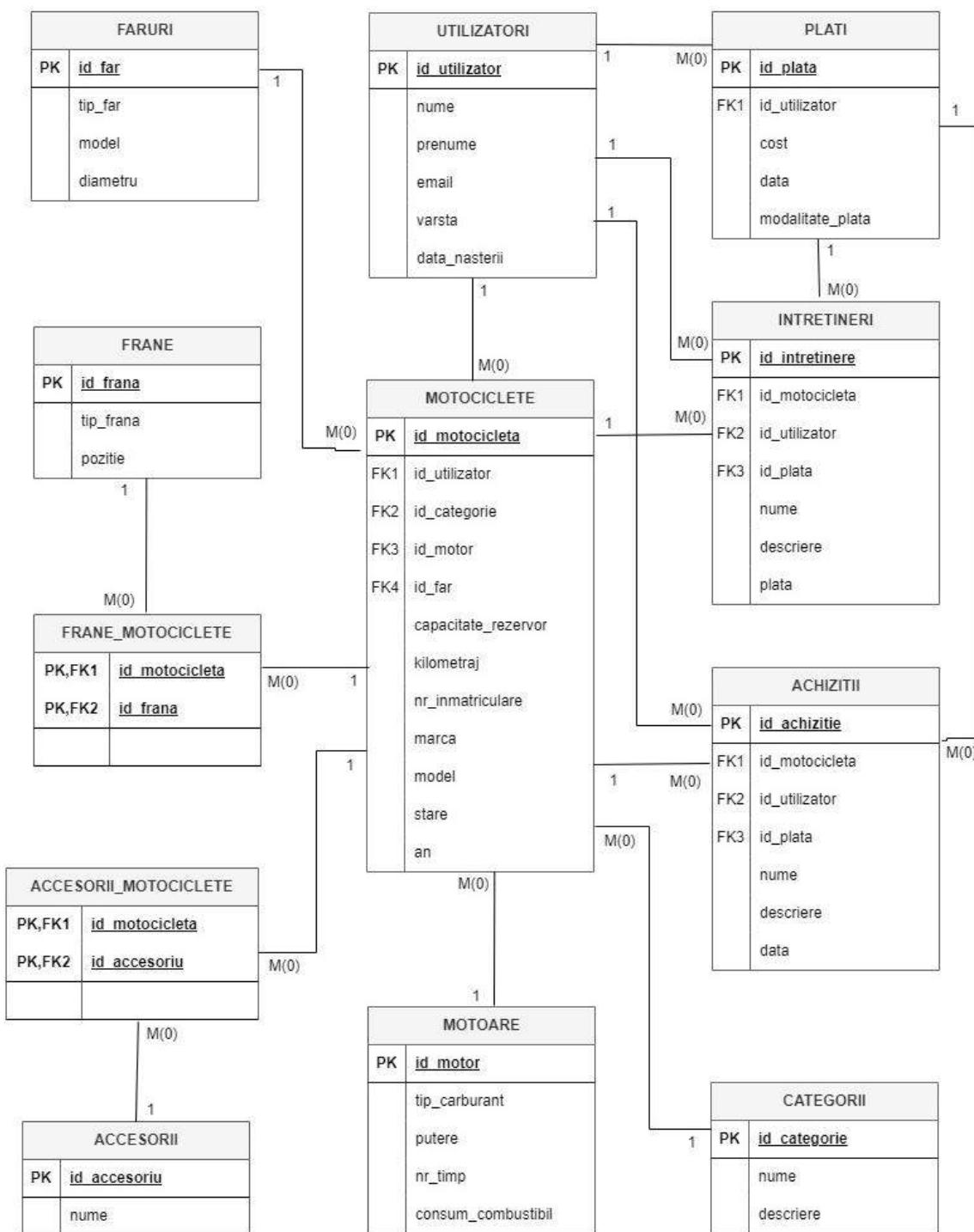
*trebuie definite în limba română (vezi curs SGBD / model de diagrama ERD;*

*nu se va accepta alt format).*

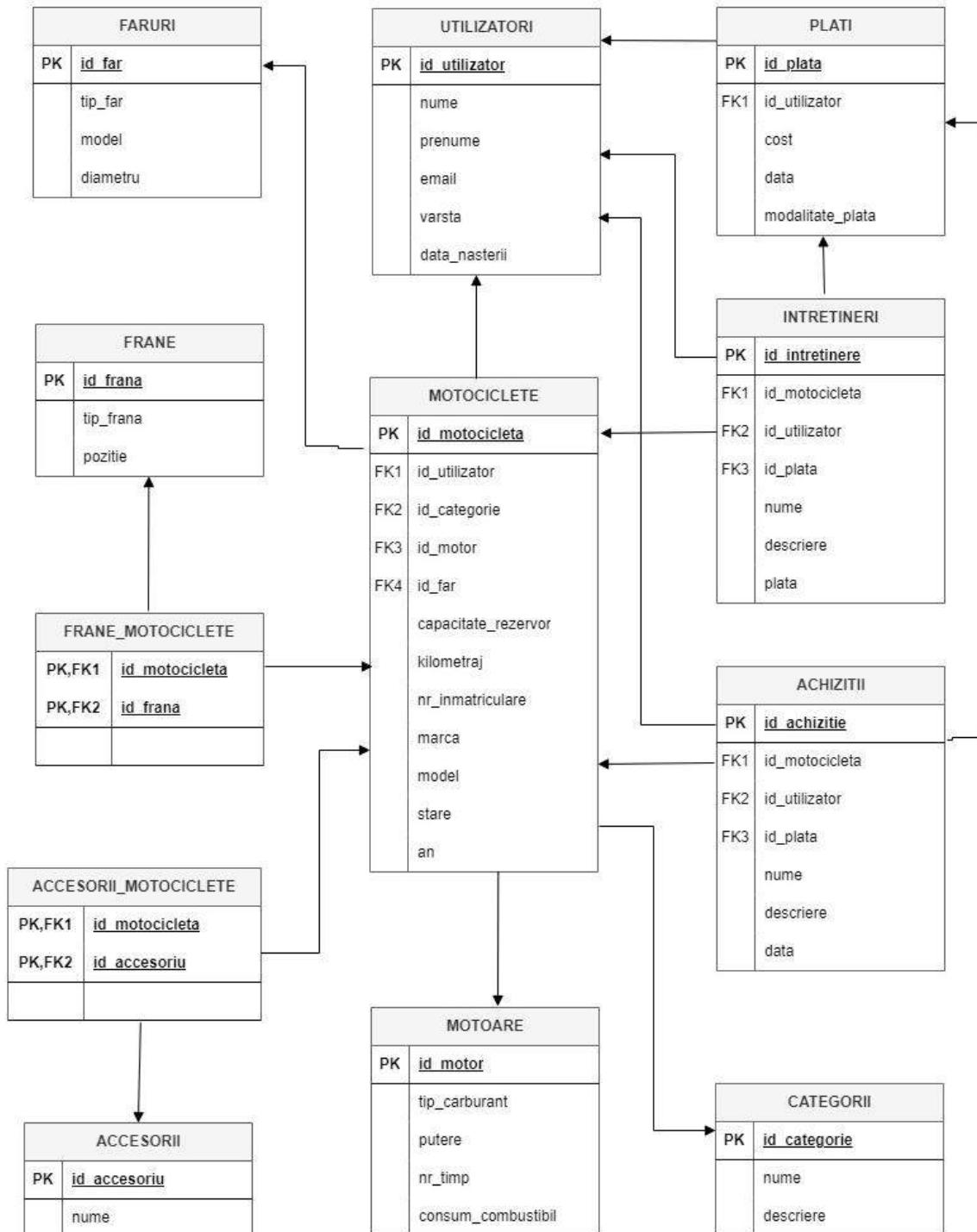


**3. Pornind de la diagrama entitate-relație realizati diagrama conceptuală a**

**modelului propus, integrând toate atributele necesare: entitățile, relațiile și atributele trebuie definite în limba română.**



➤ Diagrama conceptuală, creată după modelul dat la curs



(Am ales să pun ambele variante, deoarece în cadrul laboratorului ne-a fost indicat să adăugăm și cardinalitățile pentru ca diagrama conceptuală să fie completă, deși în cerința de proiect s-a sugerat să aplicăm formatul de la curs)

---

**4. Implementați în Oracle diagrama conceptuală realizată: definiți toate  
tabelele, definind toate constrângările de integritate necesare (chei primare,  
cheile externe etc).**

---

➤ Cod SQL

```
--Crearea unei secvențe ce va fi utilizata în inserarea înregistrărilor în tabele (punctul 11).
```

```
CREATE SEQUENCE FARURI_SEQ START WITH 1;  
CREATE SEQUENCE UTILIZATORI_SEQ START WITH 1;  
CREATE SEQUENCE MOTOCICLETE_SEQ START WITH 1;  
CREATE SEQUENCE MOTOARE_SEQ START WITH 1;  
CREATE SEQUENCE PLATI_SEQ START WITH 1;  
CREATE SEQUENCE INTRETNERI_SEQ START WITH 1;  
CREATE SEQUENCE ACHIZITII_SEQ START WITH 1;  
CREATE SEQUENCE FRANE_SEQ START WITH 1;  
CREATE SEQUENCE ACCESORII_SEQ START WITH 1;  
CREATE SEQUENCE CATEGORII_SEQ START WITH 1;
```

```
CREATE TABLE FARURI(
```

```
    id_far INT DEFAULT FARURI_SEQ.NEXTVAL PRIMARY KEY,  
    tip_far VARCHAR(30),  
    model VARCHAR(30),  
    diametru INT  
);
```

```
CREATE TABLE UTILIZATORI (
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
id_utilizator INT DEFAULT UTILIZATORI_SEQ.NEXTVAL PRIMARY KEY,  
nume VARCHAR(255),  
prenume VARCHAR(255),  
email VARCHAR(255),  
varsta INT,  
data_nasterii DATE  
);
```

CREATE TABLE FRANE (

```
id_fрана INT DEFAULT FRANE_SEQ.NEXTVAL PRIMARY KEY,  
tip_fрана VARCHAR(255),  
pozitie VARCHAR(255)  
);
```

CREATE TABLE ACCESORII (

```
id_accesoriu INT DEFAULT ACCESORII_SEQ.NEXTVAL PRIMARY KEY,  
nume VARCHAR(255)  
);
```

CREATE TABLE CATEGORII (

```
id_categorie INT DEFAULT CATEGORII_SEQ.NEXTVAL PRIMARY KEY,  
nume VARCHAR(255),  
descriere VARCHAR(255)  
);
```

CREATE TABLE MOTOARE (

```
id_motor INT DEFAULT MOTOARE_SEQ.NEXTVAL PRIMARY KEY,  
putere INT,  
consum_combustibil DECIMAL(10, 2),  
nr_timpuri INT,
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

carburant VARCHAR(255),

CONSTRAINT nr\_timp\_check CHECK (nr\_timp IN (2,4))

);

CREATE TABLE PLATI (

id\_plata INT DEFAULT PLATI\_SEQ.NEXTVAL PRIMARY KEY,

id\_utilizator INT,

cost DECIMAL(10, 2),

data DATE,

modalitate\_plata VARCHAR(100),

CONSTRAINT modalitate\_plata\_check CHECK (modalitate\_plata IN ('cash', 'card')),

FOREIGN KEY (id\_utilizator) REFERENCES UTILIZATORI(id\_utilizator)

);

CREATE TABLE MOTOCICLETE (

id\_motocicleta INT DEFAULT MOTOCICLETE\_SEQ.NEXTVAL PRIMARY KEY,

id\_utilizator INT,

id\_categorie INT,

id\_motor INT,

id\_far INT,

capacitate\_rezervor INT,

kilometraj INT,

nr\_inmatriculare VARCHAR(255),

marca VARCHAR(255),

model VARCHAR(255),

stare VARCHAR(255),

an INT,

FOREIGN KEY (id\_utilizator) REFERENCES UTILIZATORI(id\_utilizator),

FOREIGN KEY (id\_categorie) REFERENCES CATEGORII(id\_categorie),

FOREIGN KEY (id\_motor) REFERENCES MOTOARE(id\_motor),

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
FOREIGN KEY (id_far) REFERENCES FARURI(id_far)
);

```

```
CREATE TABLE INTRETNERI (
    id_intretinere INT DEFAULT INTRETNERI_SEQ.NEXTVAL PRIMARY KEY,
    id_motocicleta INT,
    id_utilizator INT,
    id_plata INT,
    nume VARCHAR(255),
    descriere VARCHAR(255),
    data DATE,
    FOREIGN KEY (id_motocicleta) REFERENCES MOTOCICLETE(id_motocicleta),
    FOREIGN KEY (id_utilizator) REFERENCES UTILIZATORI(id_utilizator),
    FOREIGN KEY (id_plata) REFERENCES PLATI(id_plata)
);

```

```
CREATE TABLE ACHIZITII (
    id_achizitie INT DEFAULT ACHIZITII_SEQ.NEXTVAL PRIMARY KEY,
    id_plata INT,
    id_utilizator INT,
    id_motocicleta INT,
    nume VARCHAR(255),
    descriere VARCHAR(255),
    data DATE,
    FOREIGN KEY (id_plata) REFERENCES PLATI(id_plata),
    FOREIGN KEY (id_utilizator) REFERENCES UTILIZATORI(id_utilizator),
    FOREIGN KEY (id_motocicleta) REFERENCES MOTOCICLETE(id_motocicleta)
);

```

```
CREATE TABLE ACCESORII_MOTOCICLETE (

```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

id\_motocicleta INT,

id\_accesoriu INT,

PRIMARY KEY (id\_motocicleta, id\_accesoriu),

FOREIGN KEY (id\_motocicleta) REFERENCES MOTOCICLETE(id\_motocicleta),

FOREIGN KEY (id\_accesoriu) REFERENCES ACCESORII(id\_accesoriu)

);

CREATE TABLE FRANE\_MOTOCICLETE (

id\_motocicleta INT,

id\_fрана INT,

PRIMARY KEY (id\_motocicleta, id\_fрана),

FOREIGN KEY (id\_motocicleta) REFERENCES MOTOCICLETE(id\_motocicleta),

FOREIGN KEY (id\_fрана) REFERENCES FRANE(id\_fрана)

);

## ➤ Print-Screen

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays a project named 'create\_project' with various database objects like tables, sequences, and packages. The central workspace contains a query builder window with the following SQL code:

```
--Create unei securante ce va fi utilizata in inserarea inregistrarilor in tabele (punctul 1).
CREATE SEQUENCE FABRICI_SEQ START WITH 1;
CREATE SEQUENCE UTILIZATORI_SEQ START WITH 1;
CREATE SEQUENCE MOTOCICLETE_SEQ START WITH 1;
CREATE SEQUENCE MOTOCARE_SEQ START WITH 1;
CREATE SEQUENCE PLATI_SEQ START WITH 1;
CREATE SEQUENCE INTREIHEBURI_SEQ START WITH 1;
CREATE SEQUENCE ACHIZITII_SEQ START WITH 1;
CREATE SEQUENCE FRANE_SEQ START WITH 1;
CREATE SEQUENCE ACCESORII_SEQ START WITH 1;
```

The 'Script Output' pane at the bottom shows the results of the sequence creation:

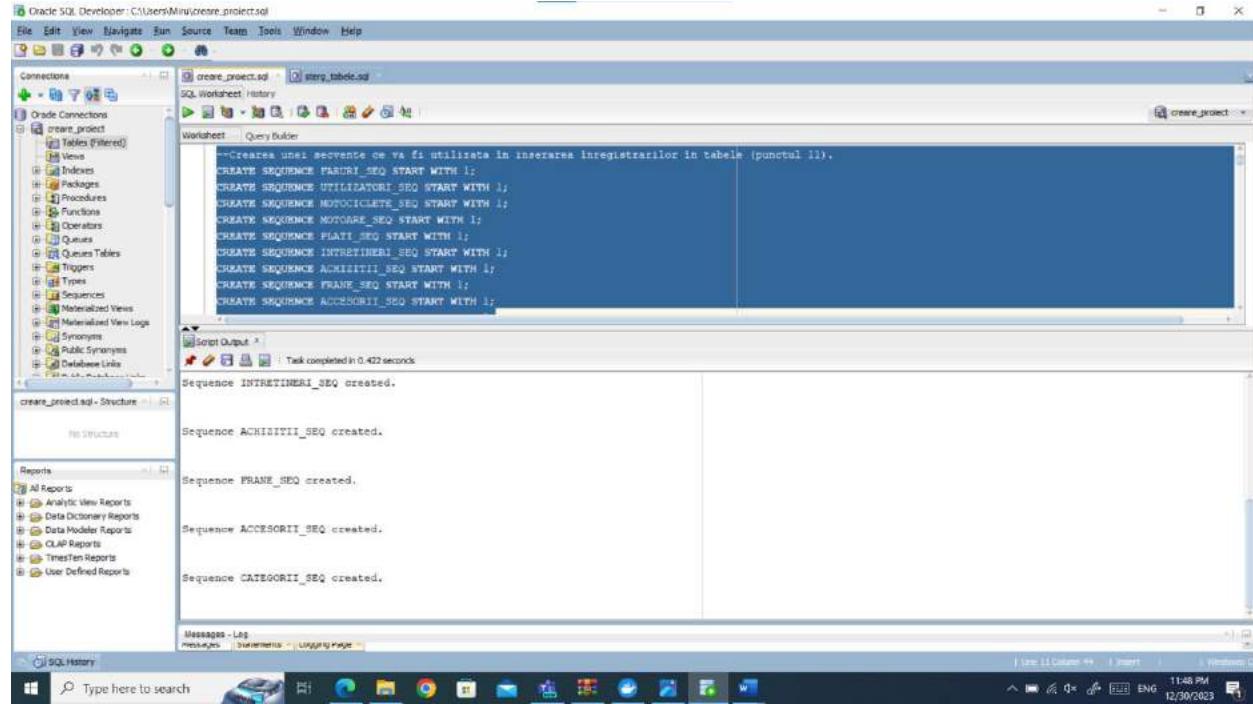
```
Sequence FABRICI_SEQ created.
Sequence UTILIZATORI_SEQ created.
Sequence MOTOCICLETE_SEQ created.
Sequence MOTOCARE_SEQ created.
Sequence PLATI_SEQ created.
```

The status bar at the bottom right indicates the current time as 11:47 PM and the date as 12/30/2023.

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

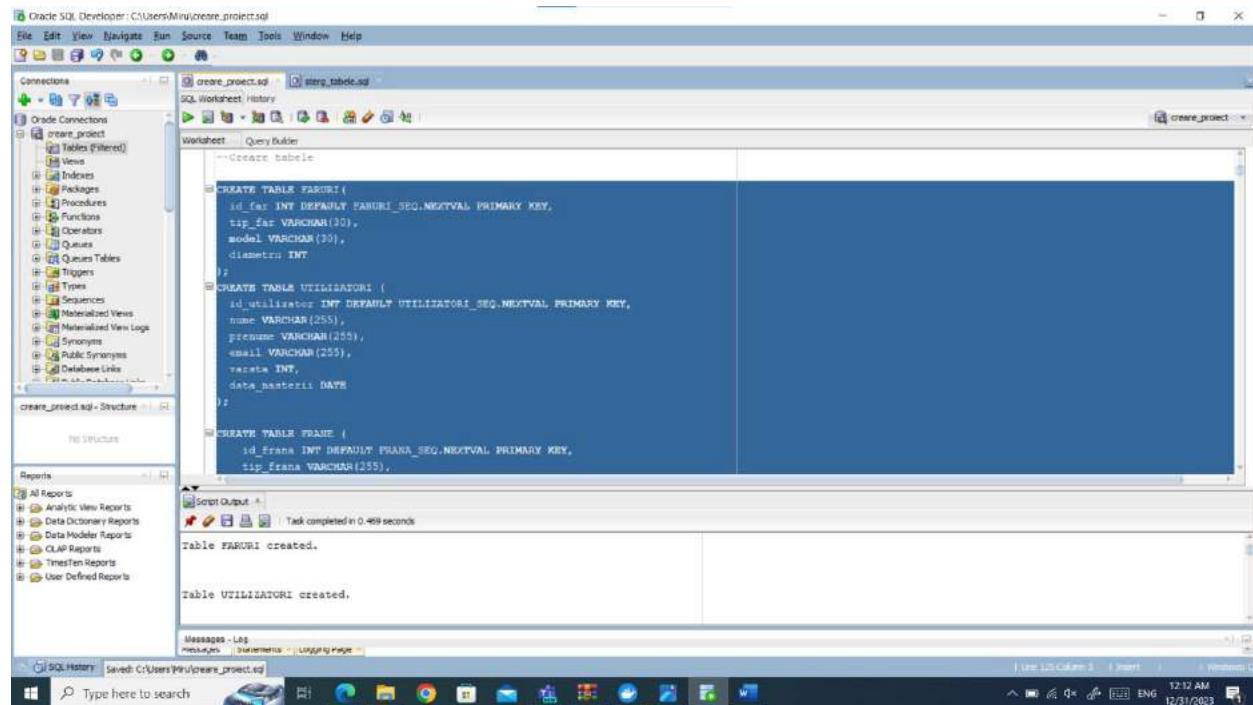
Grupa 242



The screenshot shows the Oracle SQL Developer interface with the project 'create\_project' open. In the 'Worksheet' tab, there are two scripts: 'create\_project.sql' and 'sterg\_tablele.sql'. The 'create\_project.sql' script contains SQL code to create sequences for various tables. The 'Script Output' pane shows the results of running the script, indicating that six sequences were created: INTRETINERI\_SEQ, ACHIZITII\_SEQ, PRANE\_SEQ, ACCESORII\_SEQ, CATEGORII\_SEQ, and LOGO\_SEQ. The status message 'Task completed in 0.423 seconds' is also visible.

```
--Creates unei securite ce va fi utilizata in inserarea inregistrarilor in tablele (punctul 1).
CREATE SEQUENCE PARURI_SEQ START WITH 1;
CREATE SEQUENCE UTILIZATORI_SEQ START WITH 1;
CREATE SEQUENCE MOTOCICLETE_SEQ START WITH 1;
CREATE SEQUENCE MOTORRE_SEQ START WITH 1;
CREATE SEQUENCE PLATI_SEQ START WITH 1;
CREATE SEQUENCE INTRETINERI_SEQ START WITH 1;
CREATE SEQUENCE ACHIZITII_SEQ START WITH 1;
CREATE SEQUENCE PRANE_SEQ START WITH 1;
CREATE SEQUENCE ACCESORII_SEQ START WITH 1;
```

Sequence INTRETINERI\_SEQ created.  
Sequence ACHIZITII\_SEQ created.  
Sequence PRANE\_SEQ created.  
Sequence ACCESORII\_SEQ created.  
Sequence CATEGORII\_SEQ created.



The screenshot shows the Oracle SQL Developer interface with the project 'create\_project' open. In the 'Worksheet' tab, there are two scripts: 'create\_project.sql' and 'sterg\_tablele.sql'. The 'create\_project.sql' script contains SQL code to create three tables: FARURI, UTILIZATORI, and PRANE. The 'Script Output' pane shows the results of running the script, indicating that three tables were created: FARURI, UTILIZATORI, and PRANE. The status message 'Task completed in 0.469 seconds' is also visible.

```
--Create table
CREATE TABLE FARURI (
    id_far INT DEFAULT FARURI_SEQ.NEXTVAL PRIMARY KEY,
    tip_far VARCHAR(30),
    model VARCHAR(30),
    diametru INT
);

CREATE TABLE UTILIZATORI (
    id_utilizator INT DEFAULT UTILIZATORI_SEQ.NEXTVAL PRIMARY KEY,
    nume VARCHAR(255),
    prenume VARCHAR(255),
    email VARCHAR(255),
    varsta INT,
    data_nasterii DATE
);

CREATE TABLE PRANE (
    id_prana INT DEFAULT PRANE_SEQ.NEXTVAL PRIMARY KEY,
    tip_prana VARCHAR(255),
    tip_fara INT
);
```

Table FARURI created.  
Table UTILIZATORI created.

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

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

- Connections:** Oracle Connections, currently connected to "create\_project".
- SQL Worksheet:** History tab is selected.
- Query Builder:** Contains the following SQL code:

```
CREATE TABLE FRAME (
    id_fraza INT DEFAULT FRAMA_SEQ.NEXTVAL PRIMARY KEY,
    tip_fraza VARCHAR(255),
    pozitie VARCHAR(255)
);

CREATE TABLE ACCESORII (
    id_accesorii INT DEFAULT ACCESORII_SEQ.NEXTVAL PRIMARY KEY,
    nume VARCHAR(255)
);

CREATE TABLE CATEGORII (
    id_categorie INT DEFAULT CATEGORII_SEQ.NEXTVAL PRIMARY KEY,
    nume VARCHAR(255),
    descriere VARCHAR(255)
);
```

**Script Output:** Task completed in 0.469 seconds.

Table FRAME created.  
Table ACCESORII created.  
Table CATEGORII created.

**Messages - Log:** No messages.

**SQL History:** Saved: C:\Users\Miru\create\_project.sql

Type here to search

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

- Connections:** Oracle Connections, currently connected to "create\_project".
- SQL Worksheet:** History tab is selected.
- Query Builder:** Contains the following SQL code:

```
CREATE TABLE MOTOCARE (
    id_motor INT DEFAULT MOTOCARE_SEQ.NEXTVAL PRIMARY KEY,
    putere INT,
    consum_combustibil DECIMAL(10, 2),
    nr_timpi INT,
    carburant VARCHAR(255),
    CONSTRAINT nr_timp_check CHECK (nr_timpi IN (2, 4))
);

CREATE TABLE PLATI (
    id_plata INT DEFAULT PLATI_SEQ.NEXTVAL PRIMARY KEY,
    id_utilizator INT,
    cost DECIMAL(10, 2),
    data DATE,
    modalitate_plata VARCHAR(100),
    CONSTRAINT modalitate_plata_check CHECK (modalitate_plata IN ('cash', 'card')),
    FOREIGN KEY (id_utilizator) REFERENCES UTILIZATORI(id_utilizator)
);
```

**Script Output:** Task completed in 0.469 seconds.

Table MOTOCARE created.  
Table PLATI created.

**Messages - Log:** No messages.

**SQL History:** Saved: C:\Users\Miru\create\_project.sql

Type here to search

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, under 'Connections' and 'create\_project', there is a 'Tables (Filtered)' section. A table named 'MOTOCICLET' is listed. The 'Script Output' pane at the bottom shows the SQL code for creating the table:

```
CREATE TABLE MOTOCICLET (
    id_motocicleta INT DEFAULT MOTOCICLETSEQ.NEXTVAL PRIMARY KEY,
    id_utilizator INT,
    id_categoria INT,
    id_motor INT,
    id_fax INT,
    capacitate_reservor INT,
    kilometraj INT,
    nr_inmatriculare VARCHAR(255),
    marca VARCHAR(255),
    model VARCHAR(255),
    starea VARCHAR(255),
    an INT,
    FOREIGN KEY (id_utilizator) REFERENCES UTILIZATORI(id_utilizator),
    FOREIGN KEY (id_categoria) REFERENCES CATEGORII(id_categoria),
    FOREIGN KEY (id_motor) REFERENCES MOTOGRE(id_motor),
    FOREIGN KEY (id_fax) REFERENCES FAXURI(id_fax)
);
```

The output window shows the results of the execution:

```
Table PLATI created.  
Table MOTOCICLET created.  
Table INTRETNITERI created.
```

At the bottom right, the status bar indicates: Line 80 Column 1 Insert 12:15 AM 12/31/2023.

This screenshot shows the Oracle SQL Developer interface again. The 'create\_project' connection is selected. The 'Tables (Filtered)' section shows two new tables: 'INTRETNITERI' and 'ACHIZITII'. The 'Script Output' pane shows the SQL code for their creation:

```
CREATE TABLE INTRETNITERI (
    id_intretinere INT DEFAULT INTRETNITERI_SEQ.NEXTVAL PRIMARY KEY,
    id_motocicleta INT,
    id_utilizator INT,
    id_plata INT,
    nume VARCHAR(255),
    descriere VARCHAR(255),
    data DATE,
    FOREIGN KEY (id_motocicleta) REFERENCES MOTOCICLET(id_motocicleta),
    FOREIGN KEY (id_utilizator) REFERENCES UTILIZATORI(id_utilizator),
    FOREIGN KEY (id_plata) REFERENCES PLATI(id_plata)
);

CREATE TABLE ACHIZITII (
    id_achiziție INT DEFAULT ACHIZITII_SEQ.NEXTVAL PRIMARY KEY,
    id_plata INT,
    id_utilizator INT,
    id_motocicleta INT,
    nume VARCHAR(255),
    descriere VARCHAR(255),
    data DATE
);
```

The output window shows the results of the execution:

```
Table INTRETNITERI created.  
Table ACHIZITII created.
```

At the bottom right, the status bar indicates: Line 80 Column 1 Insert 12:15 AM 12/31/2023.

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface with the 'create\_project.sql' script open in the workspace. The code defines the 'ACHIZITII' table with columns: id\_achizitie (INT, primary key), id\_plata (INT), id\_utilizator (INT), id\_motocicleta (INT), nume (VARCHAR(255)), descriere (VARCHAR(255)), and data (DATE). Foreign keys link 'id\_plata' to 'PLATI', 'id\_utilizator' to 'UTILIZATORI', and 'id\_motocicleta' to 'MOTOCICLETE'. The 'Script Output' pane shows the command 'Table INTRETIENERI created.' and the message 'Task completed in 0.469 seconds'.

```
descriere VARCHAR(255),
data DATE,
FOREIGN KEY (id_motocicleta) REFERENCES MOTOCICLETE(id_motocicleta),
FOREIGN KEY (id_utilizator) REFERENCES UTILIZATORI(id_utilizator),
FOREIGN KEY (id_plata) REFERENCES PLATI(id_plata)
);

CREATE TABLE ACHIZITII (
    id_achizitie INT DEFAULT ACHIZITII_SEQ.NEXTVAL PRIMARY KEY,
    id_plata INT,
    id_utilizator INT,
    id_motocicleta INT,
    nume VARCHAR(255),
    descriere VARCHAR(255),
    data DATE,
    FOREIGN KEY (id_plata) REFERENCES PLATI(id_plata),
    FOREIGN KEY (id_utilizator) REFERENCES UTILIZATORI(id_utilizator),
    FOREIGN KEY (id_motocicleta) REFERENCES MOTOCICLETE(id_motocicleta)
);
```

The screenshot shows the Oracle SQL Developer interface with the 'create\_project.sql' script open in the workspace. The code defines two tables: 'ACCESORII\_MOTOCICLETE' with columns id\_motocicleta (INT, primary key), id\_accesorii (INT), and id\_accesoriu (INT), and 'FRANE\_MOTOCICLETE' with columns id\_motocicleta (INT, primary key), id\_frana (INT), and id\_frama (INT). Both tables have foreign keys linking 'id\_motocicleta' to 'MOTOCICLETE'. The 'Script Output' pane shows the commands 'Table ACHIZITII created.', 'Table ACCESORII\_MOTOCICLETE created.', and 'Table FRANE\_MOTOCICLETE created.' followed by the message 'Task completed in 0.469 seconds'.

```
CREATE TABLE ACCESORII_MOTOCICLETE (
    id_motocicleta INT,
    id_accesorii INT,
    id_accesoriu INT,
    PRIMARY KEY (id_motocicleta, id_accesorii),
    FOREIGN KEY (id_motocicleta) REFERENCES MOTOCICLETE(id_motocicleta),
    FOREIGN KEY (id_accesorii) REFERENCES ACCESORII(id_accesorii)
);

CREATE TABLE FRANE_MOTOCICLETE (
    id_motocicleta INT,
    id_frana INT,
    PRIMARY KEY (id_motocicleta, id_frama),
    FOREIGN KEY (id_motocicleta) REFERENCES MOTOCICLETE(id_motocicleta),
    FOREIGN KEY (id_frama) REFERENCES FRANE(id_frama)
);
```

**5. Adăugați informații coerente în tabelele create (minim 5 înregistrări**

**pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela  
asociativă).**

---

1. Pentru tabelul Faruri

➤ Cod SQL:

```
INSERT INTO FARURI (tip_far, model, diametru)
VALUES ('Halogen', 'H4', 35);
```

```
INSERT INTO FARURI (tip_far, model, diametru)
VALUES('Halogen', 'H7', 40);
```

```
INSERT INTO FARURI (tip_far, model, diametru)
VALUES('LED', 'SMD', 25);
```

```
INSERT INTO FARURI (tip_far, model, diametru)
VALUES ('Xenon', 'D2S', 35);
```

```
INSERT INTO FARURI (tip_far, model, diametru)
VALUES ('Halogen', 'H1', 30);
```

```
INSERT INTO FARURI (tip_far, model, diametru)
VALUES ('LED', 'COB', 35);
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
INSERT INTO FARURI (tip_far, model, diametru)
```

```
VALUES ('Xenon', 'D1S', 25);
```

```
INSERT INTO FARURI (tip_far, model, diametru)
```

```
VALUES ('Halogen', 'H3', 30);
```

```
INSERT INTO FARURI (tip_far, model, diametru)
```

```
VALUES ('LED', 'RGB', 40);
```

```
INSERT INTO FARURI (tip_far, model, diametru)
```

```
VALUES ('Xenon', 'D3S', 25);
```

```
SELECT *
```

```
FROM FARURI
```

```
ORDER BY ID_FAR;
```

➤ Print-Screen:

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the project structure for 'create\_project'. The 'Tables' section is expanded, showing 'FARIURI' as the selected table. The 'Script Output' pane at the bottom shows the execution results of an SQL script named 'sterg\_tablede.sql'. The script contains several INSERT statements into the 'FARIURI' table, each adding a new row with values for tip\_far, model, and diametru. The output indicates that 1 row was inserted successfully.

```
--1.Pentru tabelul FARIURI

INSERT INTO FARIURI (tip_far, model, diametru)
VALUES ('Halogen', 'H4', 35);

INSERT INTO FARIURI (tip_far, model, diametru)
VALUES('Halogen', 'H7', 40);

INSERT INTO FARIURI (tip_far, model, diametru)
VALUES('LED', 'SMD', 25);

INSERT INTO FARIURI (tip_far, model, diametru)
VALUES ('Xenon', 'D2S', 35);

INSERT INTO FARIURI (tip_far, model, diametru)
VALUES ('Halogen', 'H1', 30);

INSERT INTO FARIURI (tip_far, model, diametru)
VALUES ('LED', 'COB', 35);

INSERT INTO FARIURI (tip_far, model, diametru)
VALUES ('Xenon', 'D1S', 25);
```

Script Output: Task completed in 0.271 seconds  
1 row inserted.

This screenshot is identical to the one above, showing the Oracle SQL Developer interface with the 'create\_project' project open. The 'Tables' section of the sidebar shows 'FARIURI' selected. The 'Script Output' pane at the bottom shows the execution results of the 'sterg\_tablede.sql' script. The output shows four separate '1 row inserted.' messages, indicating that four rows were deleted from the 'FARIURI' table.

```
INSERT INTO FARIURI (tip_far, model, diametru)
VALUES ('Xenon', 'D1S', 25);

INSERT INTO FARIURI (tip_far, model, diametru)
VALUES ('Halogen', 'H3', 30);

INSERT INTO FARIURI (tip_far, model, diametru)
VALUES ('LED', 'RGB', 40);

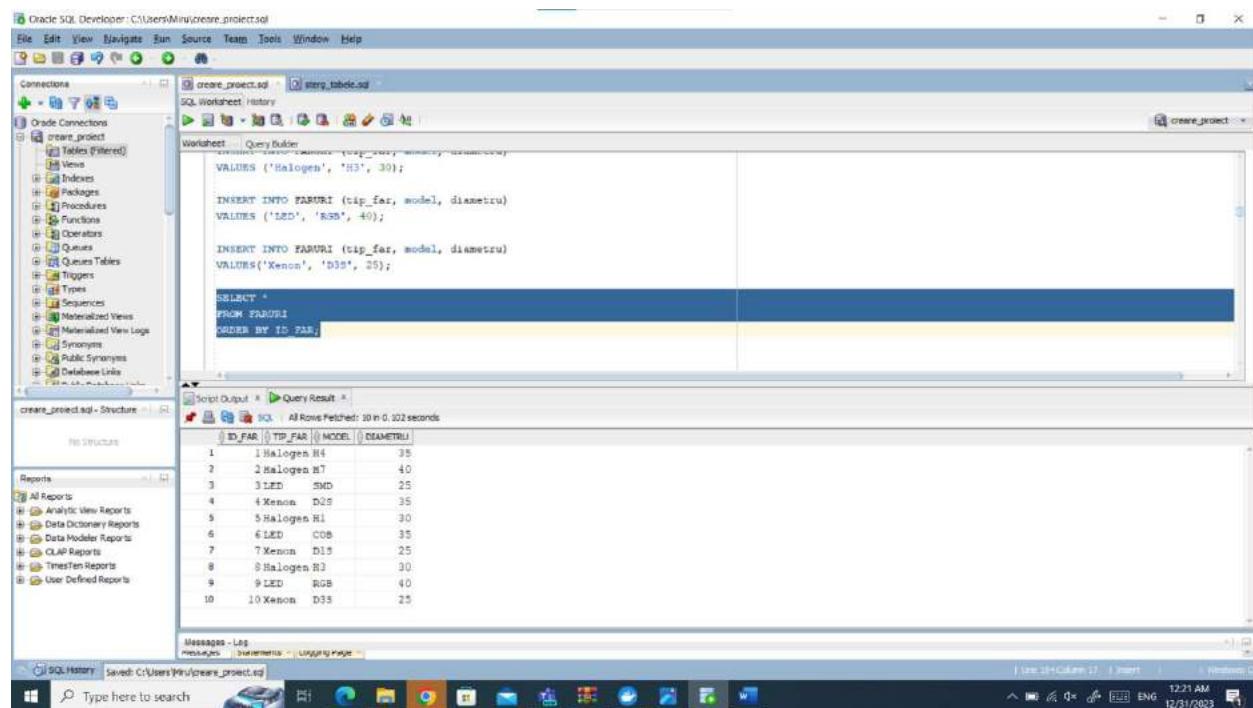
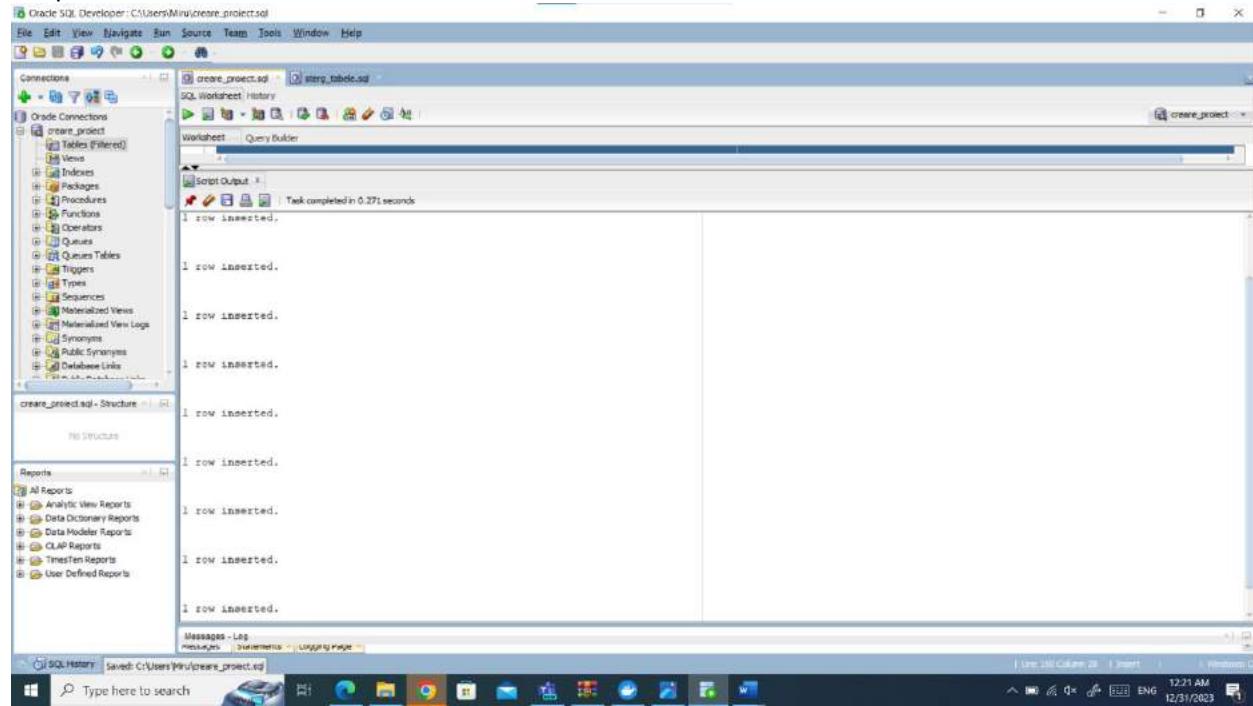
INSERT INTO FARIURI (tip_far, model, diametru)
VALUES ('Xenon', 'D3S', 25);
```

Script Output: Task completed in 0.271 seconds  
1 row inserted.  
1 row inserted.  
1 row inserted.  
1 row inserted.

## Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242



## 2. Pentru tabelul UTILIZATORI

- Cod SQL

```
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES ('Popescu', 'Ion',  
'popescu.ion@gmail.com', 35, '12-FEB-1990');
```

```
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES('Ionescu', 'Maria',  
'ionescu.maria@yahoo.com', 27, '20-MAY-1996');
```

```
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES('Georgescu',  
'Alexandru', 'georgescu.alex@yahoo.com', 41, '01-NOV-1980');
```

```
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES('Radulescu',  
'Andreea', 'radulescu.andreea@gmail.com', 23, '08-SEP-1998');
```

```
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES('Popa', 'Mihai',  
'popa.mihai@gmail.com', 31, '17-APR-1990');
```

```
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES('Cristea', 'Ana',  
'cristea.ana@yahoo.com', 29, '25-JUL-1992');
```

```
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES('Dumitrescu',  
'Alexandra', 'dumitrescu.alexandra@gmail.com', 38, '30-DEC-1983');
```

```
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES('Constantinescu',  
'Cristian', 'constantinescu.cristian@yahoo.com', 45, '06-MAR-1976');
```

```
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES('Popovici',  
'Diana', 'popovici.diana@gmail.com', 26, '14-AUG-1995');
```

```
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES('Muresan',  
'Radu', 'muresan.radu@gmail.com', 33, '10-JAN-1988');
```

```
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES('Muresan',  
'Radu', 'muresan.radu@gmail.com', 33, '10-JAN-1988');
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

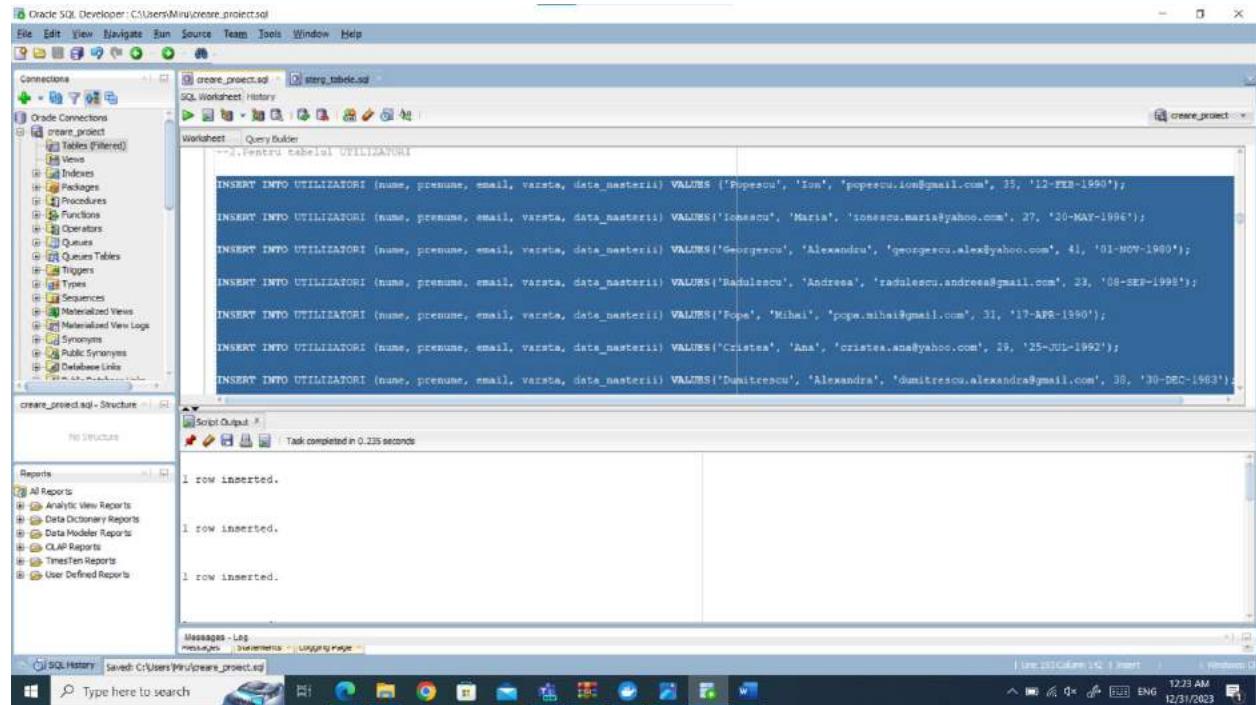
Grupa 242

INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data\_nasterii) VALUES('Muresan', 'Radu', 'muresan.radu@gmail.com', 33, '10-JAN-1988');

INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data\_nasterii) VALUES('Muresan', 'Radu', 'muresan.radu@gmail.com', 33, '10-JAN-1988');

```
SELECT *
FROM UTILIZATORI
ORDER BY ID_UTILIZATOR;
```

## ➤ Print-Screen



The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the project structure for 'create\_project' with various database objects like tables, views, and packages. The central workspace contains a query builder window with the following SQL code:

```
--> Inserturi tabelul UTILIZATORI
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES ('Popescu', 'Ion', 'popescu.ion@gmail.com', 35, '12-FEB-1990');
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES ('Ionescu', 'Maria', 'ionescu.maria@yahoo.com', 27, '20-MAY-1996');
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES ('Georgescu', 'Alexandru', 'georgescu.alex@yahoo.com', 41, '01-NOV-1980');
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES ('Radulescu', 'Andreea', 'radulescu.andreea@gmail.com', 23, '08-SEZ-1998');
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES ('Pop', 'Mihai', 'pop.mihai@gmail.com', 31, '17-APR-1990');
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES ('Cristea', 'Ana', 'cristea.ana@yahoo.com', 29, '25-JUL-1992');
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES ('Dumitrescu', 'Alexandra', 'dumitrescu.alexandra@gmail.com', 38, '30-DEC-1993');
```

The 'Script Output' pane below the query builder shows the results of the insertions:

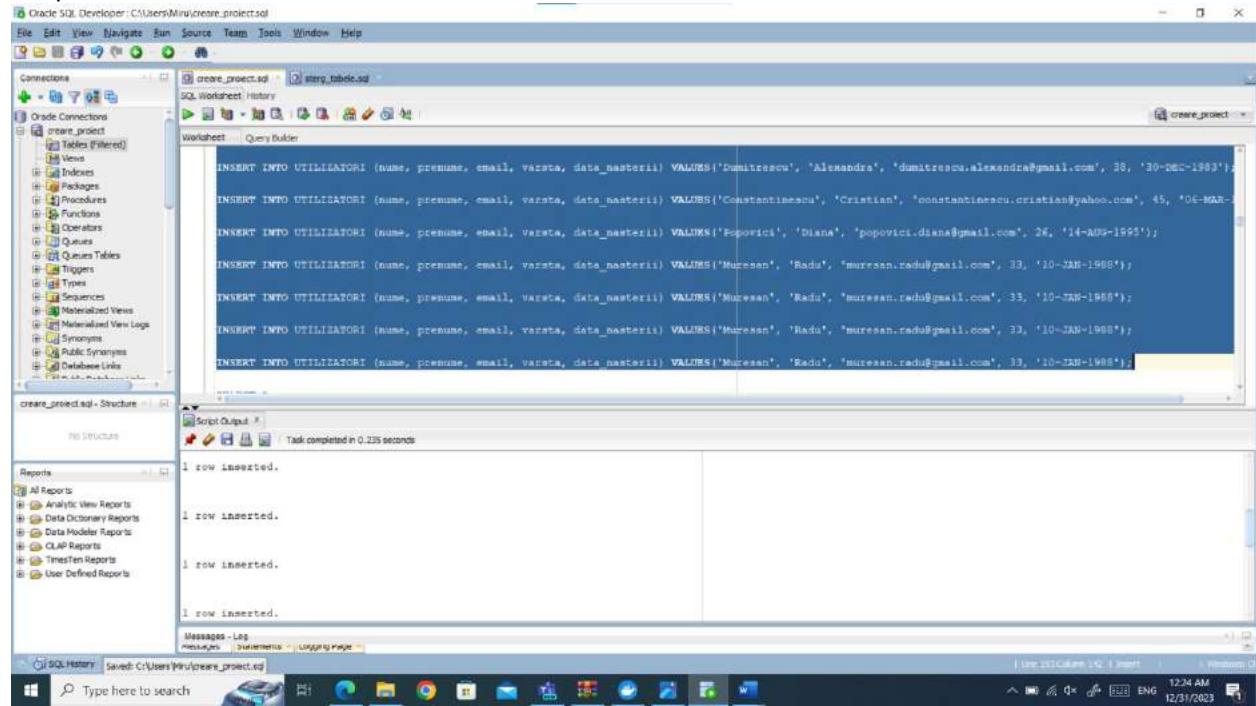
```
I row inserted.
I row inserted.
I row inserted.
```

The bottom status bar indicates the task completed in 0.135 seconds.

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242



The screenshot shows the Oracle SQL Developer interface. In the central workspace, several SQL statements are being run in a query builder. The statements are as follows:

```
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES('Dumitrescu', 'Alexandra', 'dumitrescu.alexandra@gmail.com', 38, '30-DEC-1983');

INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES('Constantinescu', 'Cristian', 'constantinescu.cristian@yahoo.com', 45, '06-MAR-1978');

INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES('Popovici', 'Diana', 'popovici.diana@gmail.com', 26, '14-AUG-1995');

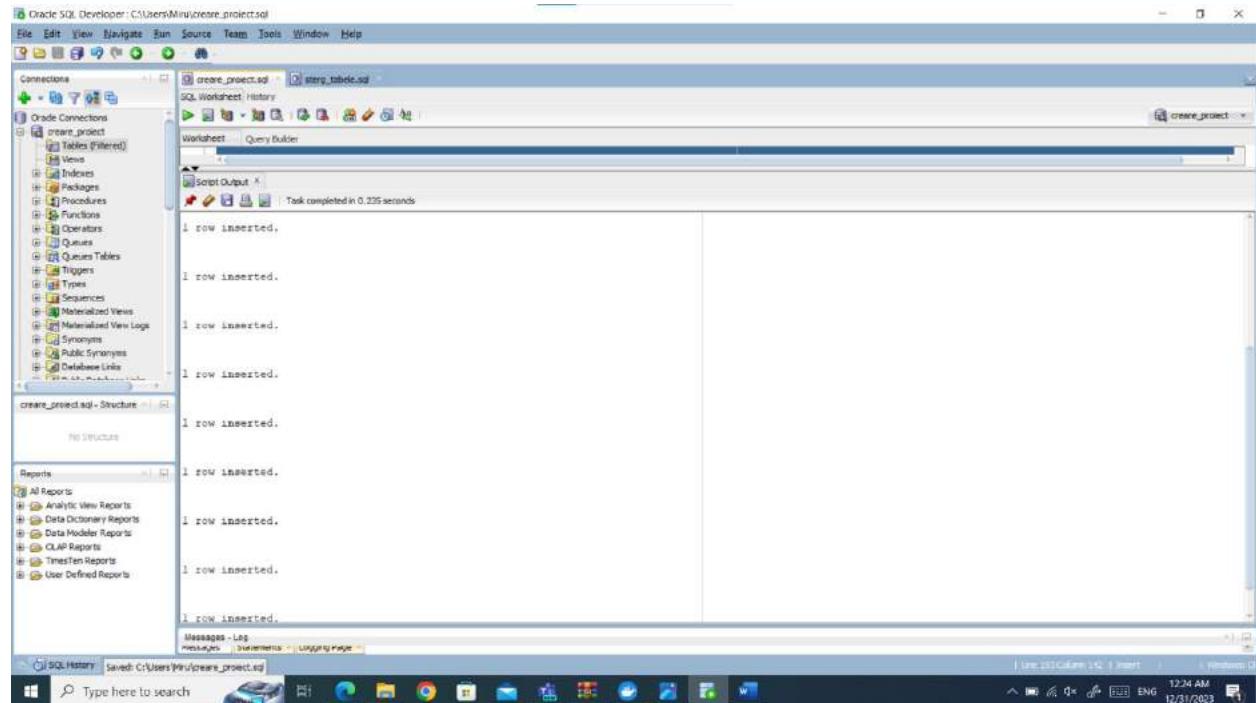
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES('Mureșan', 'Radu', 'muresan.radu@gmail.com', 33, '10-JAN-1988');

INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES('Mureșan', 'Radu', 'muresan.radu@gmail.com', 33, '10-JAN-1988');

INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES('Mureșan', 'Radu', 'muresan.radu@gmail.com', 33, '10-JAN-1988');

INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES('Mureșan', 'Radu', 'muresan.radu@gmail.com', 33, '10-JAN-1988');
```

Below the workspace, the script output shows the results of each insert statement: "1 row inserted." repeated seven times.



This screenshot shows the Oracle SQL Developer interface again, with the same set of SQL insert statements being run in a query builder. The results in the script output are identical to the first screenshot, showing "1 row inserted." for each of the seven statements.

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there's a tree view of database objects under 'create\_project'. The 'Tables' node is expanded, showing 'Tables (Filtered)'. In the center, the 'Worksheet' tab contains a SQL query:

```
INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES('Muresan', 'Radu', 'muresan.radu@gmail.com', 33, '10-JAN-1988');

INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES('Muresan', 'Radu', 'muresan.radu@gmail.com', 33, '10-JAN-1988');

INSERT INTO UTILIZATORI (nume, prenume, email, varsta, data_nasterii) VALUES('Muresan', 'Radu', 'muresan.radu@gmail.com', 33, '10-JAN-1988');

SELECT *
FROM UTILIZATORI
ORDER BY ID_UTILIZATOR;
```

Below the query, the 'Query Result' tab displays the results of the executed query:

ID_UTILIZATOR	NUME	PREGNAME	EMAIL	VARSTA	DATA_NASTERII
1	I Popescu	Ion	popescu.ion@gmail.com	35	12-FEB-90
2	Ionescu	Maria	ionescu.maria@yahoo.com	27	20-NOV-96
3	Gorogescu	Alexandru	georgescu.alex@yahoo.com	41	01-NOV-80
4	Radulescu	Andreea	radulescu.andreea@gmail.com	23	09-SEP-98
5	P Popa	Mihai	popa.mihai@gmail.com	31	17-AEP-90
6	Cristea	Ana	cristea.anu@yahoo.com	29	25-JUL-92
7	Dumitrescu	Alexandra	dumitrescu.alexandra@gmail.com	36	30-DEC-93
8	C Constantinescu	Cristian	constantinescu.cristian@yahoo.com	45	06-MAR-76
9	P Popovici	Diana	popovici.diana@gmail.com	26	14-AUG-95
10	Muresan	Radu	muresan.radu@gmail.com	33	10-JAN-88
11	Muresan	Radu	muresan.radu@gmail.com	33	10-JAN-88
12	Muresan	Radu	muresan.radu@gmail.com	33	10-JAN-88
13	Muresan	Radu	muresan.radu@gmail.com	33	10-JAN-88

### 3. Pentru tabelul FRANE

#### ➤ Cod SQL

INSERT INTO FRANE (tip\_frama, pozitie) VALUES ('Ferodo Disc', 'Fata-Dreapta');

INSERT INTO FRANE (tip\_frama, pozitie) VALUES('Ferodo Disc', 'Spate-Stanga');

INSERT INTO FRANE (tip\_frama, pozitie) VALUES('Ferodo Disc','Spate-Dreapta');

INSERT INTO FRANE (tip\_frama, pozitie) VALUES('Brembo Disc', 'Fata-Stanga');

INSERT INTO FRANE (tip\_frama, pozitie) VALUES('Brembo Disc', 'Fata-Dreapta');

INSERT INTO FRANE (tip\_frama, pozitie) VALUES('Brembo Disc', 'Spate-Stanga');

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
INSERT INTO FRANE (tip_frama, pozitie) VALUES('Brembo Disc','Spate-Dreapta');
```

```
INSERT INTO FRANE (tip_frama, pozitie) VALUES('TRW Placute', 'Fata-Stanga');
```

```
INSERT INTO FRANE (tip_frama, pozitie) VALUES('TRW Placute','Spate-Stanga');
```

```
INSERT INTO FRANE (tip_frama, pozitie) VALUES('TRW Placute','Fata-Dreapta');
```

```
SELECT *
```

```
FROM FRANE
```

```
ORDER BY ID_FRANA;
```

## ➤ Print-Screen

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The title bar indicates the connection is to 'create\_project' and the current file is 'sterg\_table.sql'. The left sidebar contains a tree view of database objects under 'Connections' for the 'create\_project' connection, including Tables, Views, Indexes, Packages, Procedures, Functions, Operators, Queues, Queues Tables, Triggers, Types, Sequences, Materialized Views, Materialized View Logs, Synonyms, Public Synonyms, and Database Links. Below this is a 'Structure' tab. The main workspace is titled 'Worksheet - Query Builder' and contains the following SQL code:

```
INSERT INTO FRANE (tip_frama, pozitie) VALUES ('Pezudo Disc', 'Fata-Dreapta');

INSERT INTO FRANE (tip_frama, pozitie) VALUES('Pezudo Disc', 'Spate-Stanga');

INSERT INTO FRANE (tip_frama, pozitie) VALUES('Pezudo Disc','Spate-Dreapta');

INSERT INTO FRANE (tip_frama, pozitie) VALUES('Brembo Disc', 'Fata-Stanga');

INSERT INTO FRANE (tip_frama, pozitie) VALUES('Brembo Disc', 'Fata-Dreapta');

INSERT INTO FRANE (tip_frama, pozitie) VALUES('Brembo Disc', 'Spate-Stanga');

INSERT INTO FRANE (tip_frama, pozitie) VALUES('Brembo Disc','Spate-Dreapta');

INSERT INTO FRANE (tip_frama, pozitie) VALUES('TRW Placute', 'Fata-Stanga');
```

Below the worksheet is a 'Script Output' window showing the results of the executed queries:

```
1 row inserted.

1 row inserted.

1 row inserted.

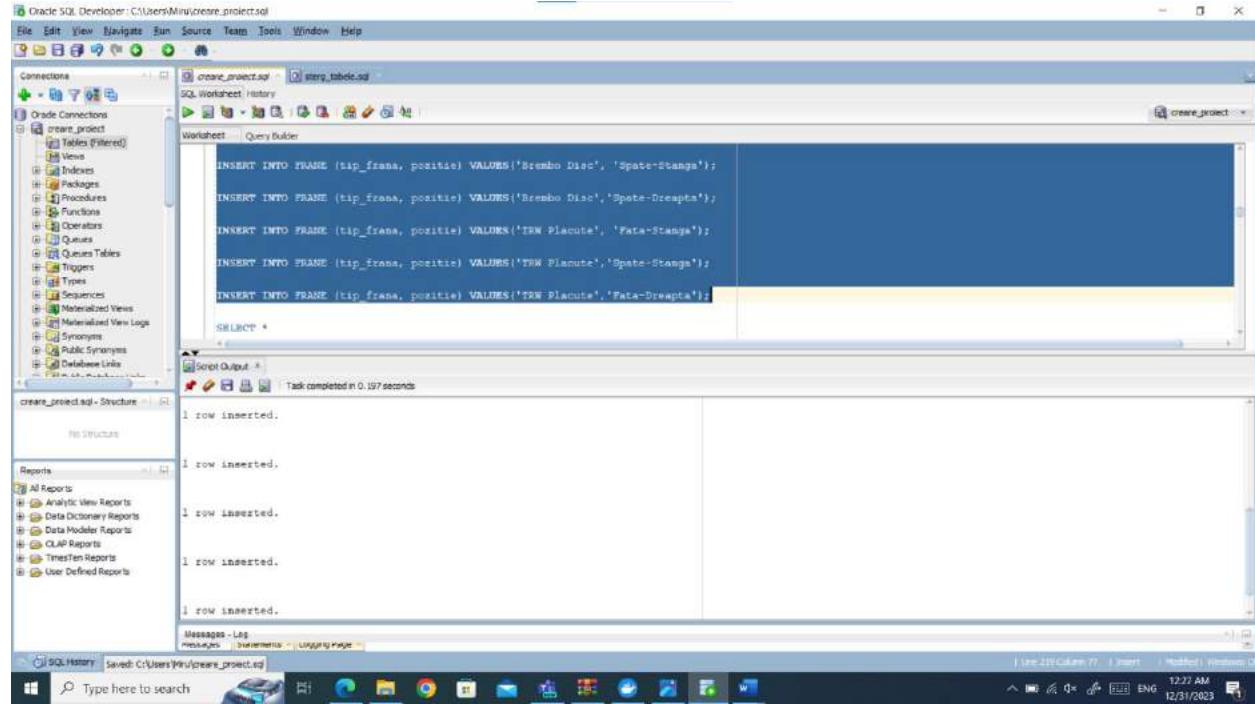
1 row inserted.
```

The bottom status bar shows the path 'Saved C:\Users\Miru\pearse\_project.ed', the system tray with icons for battery, volume, and network, and the system clock showing '12:26 AM 12/31/2023'.

## Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

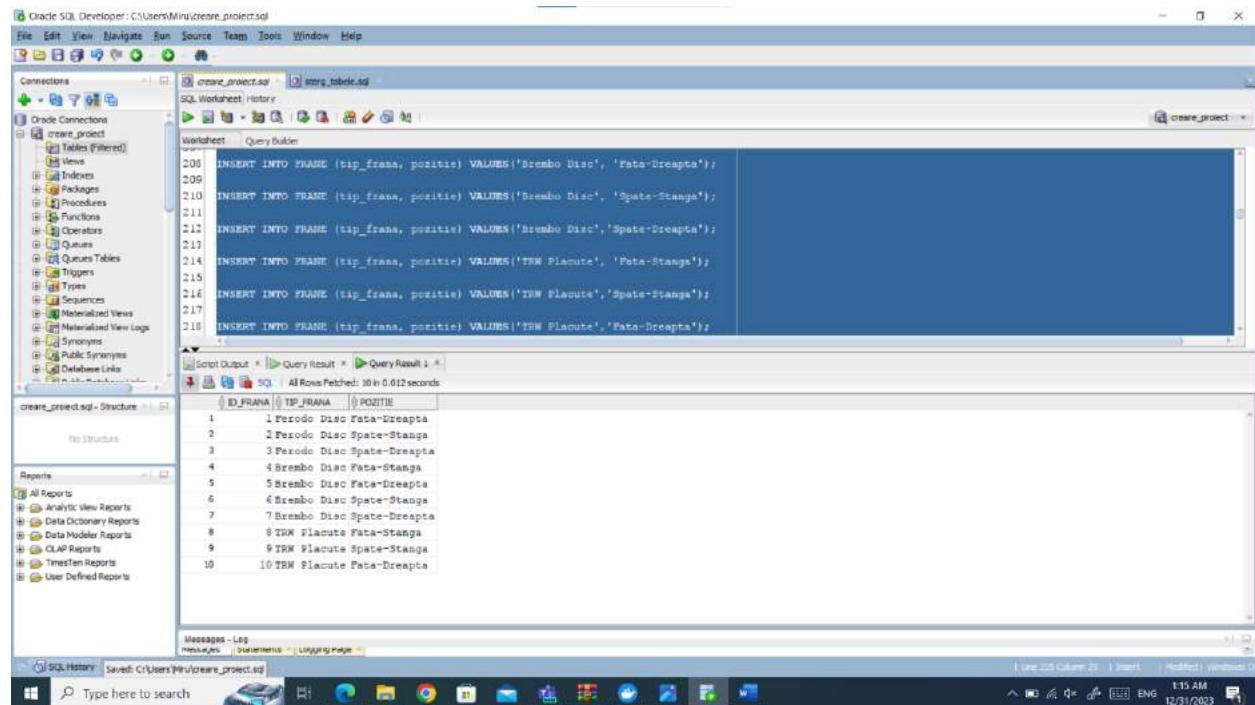
Grupa 242



The screenshot shows the Oracle SQL Developer interface. In the central workspace, there are five identical SQL statements being run:

```
INSERT INTO FRANE (tip_frama, pozitie) VALUES('Brembo Disc', 'Spate-Stanga');
INSERT INTO FRANE (tip_frama, pozitie) VALUES('Brembo Disc', 'Spate-Dreapta');
INSERT INTO FRANE (tip_frama, pozitie) VALUES('TRW Placute', 'Fata-Stanga');
INSERT INTO FRANE (tip_frama, pozitie) VALUES('TRW Placute', 'Fata-Dreapta');
SELECT *
```

Below each statement, the output shows "1 row inserted." The status bar at the bottom indicates "Task completed in 0.197 seconds".



The screenshot shows the Oracle SQL Developer interface. In the central workspace, there are ten SQL statements being run:

```
208 INSERT INTO FRANE (tip_frama, pozitie) VALUES('Brembo Disc', 'Fata-Dreapta');
209
210 INSERT INTO FRANE (tip_frama, pozitie) VALUES('Brembo Disc', 'Spate-Stanga');
211
212 INSERT INTO FRANE (tip_frama, pozitie) VALUES('Brembo Disc', 'Spate-Dreapta');
213
214 INSERT INTO FRANE (tip_frama, pozitie) VALUES('TRW Placute', 'Fata-Stanga');
215
216 INSERT INTO FRANE (tip_frama, pozitie) VALUES('TRW Placute', 'Fata-Stanga');
217
218 INSERT INTO FRANE (tip_frama, pozitie) VALUES('TRW Placute', 'Fata-Dreapta');
```

Below the statements, a "Query Result" tab displays the data from the FRANE table:

ID_FRANA	TIPI_FRANA	POZITIE
1	1 Ferodo Disc	Fata-Dreapta
2	2 Ferodo Disc	Spate-Stanga
3	3 Ferodo Disc	Spate-Dreapta
4	4 Brembo Disc	Fata-Stanga
5	5 Brembo Disc	Spate-Dreapta
6	6 Brembo Disc	Spate-Stanga
7	7 Brembo Disc	Spate-Dreapta
8	8 TRW Placute	Fata-Stanga
9	9 TRW Placute	Spate-Stanga
10	10 TRW Placute	Fata-Dreapta

The status bar at the bottom indicates "All Rows Fetched: 10 in 0.012 seconds".

#### 4. Pentru tabelul ACCESORII

➤ Cod SQL

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
INSERT INTO ACCESORII (nume) VALUES ('Casca');
```

```
INSERT INTO ACCESORII (nume) VALUES('Manusi');
```

```
INSERT INTO ACCESORII (nume) VALUES('Jacheta moto');
```

```
INSERT INTO ACCESORII (nume) VALUES('Pantaloni moto');
```

```
INSERT INTO ACCESORII (nume) VALUES('Bocanci moto');
```

```
INSERT INTO ACCESORII (nume) VALUES('Ochelari moto');
```

```
INSERT INTO ACCESORII (nume) VALUES('Costum de ploaie');
```

```
INSERT INTO ACCESORII (nume) VALUES('Antifurt');
```

```
INSERT INTO ACCESORII (nume) VALUES('Pilot automat');
```

```
INSERT INTO ACCESORII (nume) VALUES('Topcase');
```

```
SELECT *
```

```
FROM ACCESORII
```

```
ORDER BY ID_ACCESORIU;
```

➤ Print-Screen

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the project structure for 'create\_project' with a single table named 'ACCESSORII'. The central workspace contains a 'Query Builder' window with the following SQL script:

```
INSERT INTO ACCESSORII (nume) VALUES ('Casca');
INSERT INTO ACCESSORII (nume) VALUES ('Nanus');
INSERT INTO ACCESSORII (nume) VALUES ('Jacheta moto');
INSERT INTO ACCESSORII (nume) VALUES ('Pantaloni moto');
INSERT INTO ACCESSORII (nume) VALUES ('Bocanci moto');
INSERT INTO ACCESSORII (nume) VALUES ('Ochelari moto');
INSERT INTO ACCESSORII (nume) VALUES ('Costum de ploaie');
```

The 'Script Output' tab shows the results of the execution:

```
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
```

The status bar at the bottom indicates the task completed in 0.891 seconds.

This screenshot shows the same Oracle SQL Developer environment after running the previous SQL script. The 'Query Builder' window now contains the following modified script:

```
INSERT INTO ACCESSORII (nume) VALUES ('Bocanci moto');
INSERT INTO ACCESSORII (nume) VALUES ('Ochelari moto');
INSERT INTO ACCESSORII (nume) VALUES ('Costum de ploaie');
INSERT INTO ACCESSORII (nume) VALUES ('Antifurt');
INSERT INTO ACCESSORII (nume) VALUES ('Pilota automata');
INSERT INTO ACCESSORII (nume) VALUES ('Topcane');
```

The 'Script Output' tab shows the results of the execution:

```
1 row inserted.
1 row inserted.
1 row inserted.
```

The status bar at the bottom indicates the task completed in 0.891 seconds.

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying an SQL script named 'sterg\_tablele.sql'. The script contains several INSERT statements into a table named 'ACCESORII' and a final SELECT statement. The 'Script Output' tab shows the results of the execution: 1 row inserted for each of the five items listed in the script, followed by a 'Query Result' section showing the same data. The status bar at the bottom indicates the task completed in 0.891 seconds.

```
INSERT INTO ACCESORII (nume) VALUES('Ochelari moto');

INSERT INTO ACCESORII (nume) VALUES('Costum de ploaie');

INSERT INTO ACCESORII (nume) VALUES('Antifurt');

INSERT INTO ACCESORII (nume) VALUES('Pilot automat');

INSERT INTO ACCESORII (nume) VALUES('Topcase');

SELECT *
FROM ACCESORII
ORDER BY ID_ACCESORIU;
```

1 row inserted.  
1 row inserted.  
1 row inserted.  
1 row inserted.  
>>Query Run In:Query Result

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying an SQL script named 'sterg\_tablele.sql'. The script contains several INSERT statements into a table named 'ACCESORII' and a final SELECT statement. The 'Script Output' tab shows the results of the execution: 1 row inserted for each of the five items listed in the script, followed by a 'Query Result' section showing the same data. The status bar at the bottom indicates all rows fetched in 0.015 seconds.

```
INSERT INTO ACCESORII (nume) VALUES('Ochelari moto');

INSERT INTO ACCESORII (nume) VALUES('Costum de ploaie');

INSERT INTO ACCESORII (nume) VALUES('Antifurt');

INSERT INTO ACCESORII (nume) VALUES('Pilot automat');

INSERT INTO ACCESORII (nume) VALUES('Topcase');

SELECT *
FROM ACCESORII
ORDER BY ID_ACCESORIU;
```

ID_ACCESORIU	NUME
1	1Casca
2	2Manusi
3	3Jacheta moto
4	4Pantaloni moto
5	5Bocanci moto
6	6Ochelari moto
7	7Costum de ploaie
8	8Antifurt
9	9Pilot automat
10	10Topcase

## 5. Pentru tabelul CATEGORII

➤ Cod SQL

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

INSERT INTO CATEGORII (nume, descriere) VALUES ('Cruiser', 'Motocicleta cu un stil clasic si confortabila pentru plimbari lungi');

INSERT INTO CATEGORII (nume, descriere) VALUES('Sport', 'Motocicleta cu un stil sportiv, agresiv si performanta mare');

INSERT INTO CATEGORII (nume, descriere) VALUES('Adventure', 'Motocicleta pentru aventura, cu suspensii lungi si putere mare');

INSERT INTO CATEGORII (nume, descriere) VALUES('Touring', 'Motocicleta pentru turism, cu un stil confortabil si spatiios');

INSERT INTO CATEGORII (nume, descriere) VALUES('Naked', 'Motocicleta fara carenaj, cu un stil minimal si agresiv');

INSERT INTO CATEGORII (nume, descriere) VALUES('Scooter', 'Motocicleta cu o platforma de picioare si un spatiu de depozitare mare');

INSERT INTO CATEGORII (nume, descriere) VALUES('Cafe Racer', 'Motocicleta cu un stil vintage si modificari pentru performanta');

INSERT INTO CATEGORII (nume, descriere) VALUES('Chopper', 'Motocicleta cu un stil personalizat si lunga deosebit de lunga');

INSERT INTO CATEGORII (nume, descriere) VALUES('Bobber', 'Motocicleta cu un stil minimalist si construita cu piese vechi');

INSERT INTO CATEGORII (nume, descriere) VALUES('Street', 'Motocicleta cu un stil urban si agresiv');

SELECT \*

FROM CATEGORII

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

ORDER BY ID\_CATEGORIE;

➤ Print-Screen

The screenshot shows the Oracle SQL Developer interface with a project named "create\_project". In the "SQL Worksheet" tab, six INSERT statements are executed into a table named "CATEGORII". The statements insert the following categories:

```
INSERT INTO CATEGORII (nume, descriere) VALUES ('Cruiser', 'Motociclete cu un stil clasic și confortabil pentru plimbări lungi');

INSERT INTO CATEGORII (nume, descriere) VALUES ('Sport', 'Motociclete cu un stil sportiv, agresiv și performanță mare');

INSERT INTO CATEGORII (nume, descriere) VALUES ('Adventure', 'Motocicleta pentru aventură, cu suspensii lungi și putere mare');

INSERT INTO CATEGORII (nume, descriere) VALUES ('Touring', 'Motocicletă pentru turism, cu un stil confortabil și spațios');

INSERT INTO CATEGORII (nume, descriere) VALUES ('Naked', 'Motocicleta fără oxigen, cu un stil minimalist și agresiv');

INSERT INTO CATEGORII (nume, descriere) VALUES ('Scooter', 'Motocicleta cu o platformă de picioare și un spațiu de depozitare mare');
```

The "Script Output" and "Query Result" panes show the results of each insert statement, indicating "1 row inserted" for each. The status bar at the bottom right shows the task completed in 0.607 seconds.

The screenshot shows the Oracle SQL Developer interface with the same project "create\_project". In the "SQL Worksheet" tab, five INSERT statements are executed into the "CATEGORII" table, followed by a SELECT statement:

```
INSERT INTO CATEGORII (nume, descriere) VALUES ('Scooter', 'Motocicleta cu o platformă de picioare și un spațiu de depozitare mare');

INSERT INTO CATEGORII (nume, descriere) VALUES ('Cafe Racer', 'Motocicleta cu un stil vintage și modificari pentru performanță');

INSERT INTO CATEGORII (nume, descriere) VALUES ('Chopper', 'Motocicleta cu un stil personalizat și lungă deosebit de lungă');

INSERT INTO CATEGORII (nume, descriere) VALUES ('Bobber', 'Motocicleta cu un stil minimalist și construită cu piese vechi');

INSERT INTO CATEGORII (nume, descriere) VALUES ('Street', 'Motocicleta cu un stil urban și agresiv');

SELECT * FROM CATEGORII;
```

The "Script Output" and "Query Result" panes show the results of each insert statement, indicating "1 row inserted" for each. The status bar at the bottom right shows the task completed in 0.607 seconds.

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface with a project named 'create\_project'. In the 'Worksheet' tab, a query is run to insert three rows into the 'CATEGORII' table:

```
CREATE OR REPLACE TABLE CATEGORII (nume, descriere) ENGINE(ORACLE) NOLOGGING LOGFILE 1 TO 2 FOR LOGGING AS MAXLOGFILE 2 FOR PERFORMANCE;
```

```
INSERT INTO CATEGORII (nume, descriere) VALUES('Chopper', 'Motocicleta cu un stil personalizat si lunga deosebit de lunga');
```

```
INSERT INTO CATEGORII (nume, descriere) VALUES('Bobber', 'Motocicleta cu un stil minimalist si construita cu piese vechi');
```

```
INSERT INTO CATEGORII (nume, descriere) VALUES('Street', 'Motocicleta cu un stil urban si agresiv');
```

```
SELECT * FROM CATEGORII ORDER BY ID_CATEGORII;
```

The 'Script Output' tab shows the results of the insertions:

```
1 row inserted.
```

The 'Query Result' tab shows the final state of the table:

ID_CATEGORII	NUME	DESCRIBE
1	Cruiser	Motocicleta cu un stil clasic si confortabil pentru plimbari lungi
2	Sport	Motocicleta cu un stil sportiv, agresiv si performanta mare
3	Adventure	Motocicleta pentru aventura, cu suspansii lungi si putere mare
4	Touring	Motocicleta pentru turism, cu un stil confortabil si spatiu mare
5	Naked	Motocicleta fara caramaj, cu un stil minimalist si agresiv
6	Scooter	Motocicleta cu o platforma de picioare si un spatiu de depozitare mare
7	Cafe Racer	Motocicleta cu un stil vintage si modificari pentru performanta
8	Chopper	Motocicleta cu un stil personalizat si lunga deosebit de lunga
9	Bobber	Motocicleta cu un stil minimalist si construita cu piese vechi
10	Street	Motocicleta cu un stil urban si agresiv

The screenshot shows the Oracle SQL Developer interface with a project named 'create\_project'. In the 'Worksheet' tab, a query is run to insert ten rows into the 'CATEGORII' table:

```
CREATE OR REPLACE TABLE CATEGORII (nume, descriere) ENGINE(ORACLE) NOLOGGING LOGFILE 1 TO 2 FOR LOGGING AS MAXLOGFILE 2 FOR PERFORMANCE;
```

```
INSERT INTO CATEGORII (nume, descriere) VALUES('Chopper', 'Motocicleta cu un stil personalizat si lunga deosebit de lunga');
```

```
INSERT INTO CATEGORII (nume, descriere) VALUES('Bobber', 'Motocicleta cu un stil minimalist si construita cu piese vechi');
```

```
INSERT INTO CATEGORII (nume, descriere) VALUES('Street', 'Motocicleta cu un stil urban si agresiv');
```

```
SELECT * FROM CATEGORII ORDER BY ID_CATEGORII;
```

The 'Script Output' tab shows the results of the insertions:

```
10 rows inserted.
```

The 'Query Result' tab shows the final state of the table:

ID_CATEGORII	NUME	DESCRIBE
1	Cruiser	Motocicleta cu un stil clasic si confortabil pentru plimbari lungi
2	Sport	Motocicleta cu un stil sportiv, agresiv si performanta mare
3	Adventure	Motocicleta pentru aventura, cu suspansii lungi si putere mare
4	Touring	Motocicleta pentru turism, cu un stil confortabil si spatiu mare
5	Naked	Motocicleta fara caramaj, cu un stil minimalist si agresiv
6	Scooter	Motocicleta cu o platforma de picioare si un spatiu de depozitare mare
7	Cafe Racer	Motocicleta cu un stil vintage si modificari pentru performanta
8	Chopper	Motocicleta cu un stil personalizat si lunga deosebit de lunga
9	Bobber	Motocicleta cu un stil minimalist si construita cu piese vechi
10	Street	Motocicleta cu un stil urban si agresiv

## 6. Pentru tabelul MOTOARE

➤ Cod SQL

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
INSERT INTO MOTOARE (putere, consum_combustibil, nr_timp, carburant) VALUES (150, 8.5, 4,'benzina');
```

```
INSERT INTO MOTOARE (putere, consum_combustibil, nr_timp, carburant) VALUES(120, 7.2, 4,'motorina');
```

```
INSERT INTO MOTOARE (putere, consum_combustibil, nr_timp, carburant) VALUES(90, 6.8, 2,'etanol');
```

```
INSERT INTO MOTOARE (putere, consum_combustibil, nr_timp, carburant) VALUES(110, 7.9, 4,'benzina');
```

```
INSERT INTO MOTOARE (putere, consum_combustibil, nr_timp, carburant) VALUES(100, 7.1, 4,'motorina');
```

```
INSERT INTO MOTOARE (putere, consum_combustibil, nr_timp, carburant) VALUES(80, 5.9, 2,'benzina');
```

```
INSERT INTO MOTOARE (putere, consum_combustibil, nr_timp, carburant) VALUES(120, 7.6, 4,'motorina');
```

```
INSERT INTO MOTOARE (putere, consum_combustibil, nr_timp, carburant) VALUES(140, 8.9, 4,'benzina');
```

```
INSERT INTO MOTOARE (putere, consum_combustibil, nr_timp, carburant) VALUES(110, 7.3, 2,'etanol');
```

```
INSERT INTO MOTOARE (putere, consum_combustibil, nr_timp, carburant) VALUES(95, 6.5, 2, 'benzina');
```

```
SELECT *
```

```
FROM MOTOARE
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

ORDER BY ID\_MOTOR;

➤ Print-Screen

The image displays two side-by-side screenshots of the Oracle SQL Developer interface. Both screenshots show the same workspace configuration with multiple tabs open. The tabs include 'create\_project.sql', 'sterg\_table.sql', 'SQL Worksheet: History', and 'create\_project'. The 'Worksheet' tab is active in both instances. The script being run is as follows:

```
INSERT INTO MOTOARE (putere, consum_combustibil, nr_timpuri, carburant) VALUES(150, 8.5, 4,'benzina');

INSERT INTO MOTOARE (putere, consum_combustibil, nr_timpuri, carburant) VALUES(120, 7.2, 4,'motorina');

INSERT INTO MOTOARE (putere, consum_combustibil, nr_timpuri, carburant) VALUES(90, 6.5, 2,'etanol');

INSERT INTO MOTOARE (putere, consum_combustibil, nr_timpuri, carburant) VALUES(110, 7.5, 4,'benzina');

INSERT INTO MOTOARE (putere, consum_combustibil, nr_timpuri, carburant) VALUES(100, 7.1, 4,'motorina');

INSERT INTO MOTOARE (putere, consum_combustibil, nr_timpuri, carburant) VALUES(80, 5.8, 2,'benzina');

INSERT INTO MOTOARE (putere, consum_combustibil, nr_timpuri, carburant) VALUES(120, 7.6, 4,'motorina');

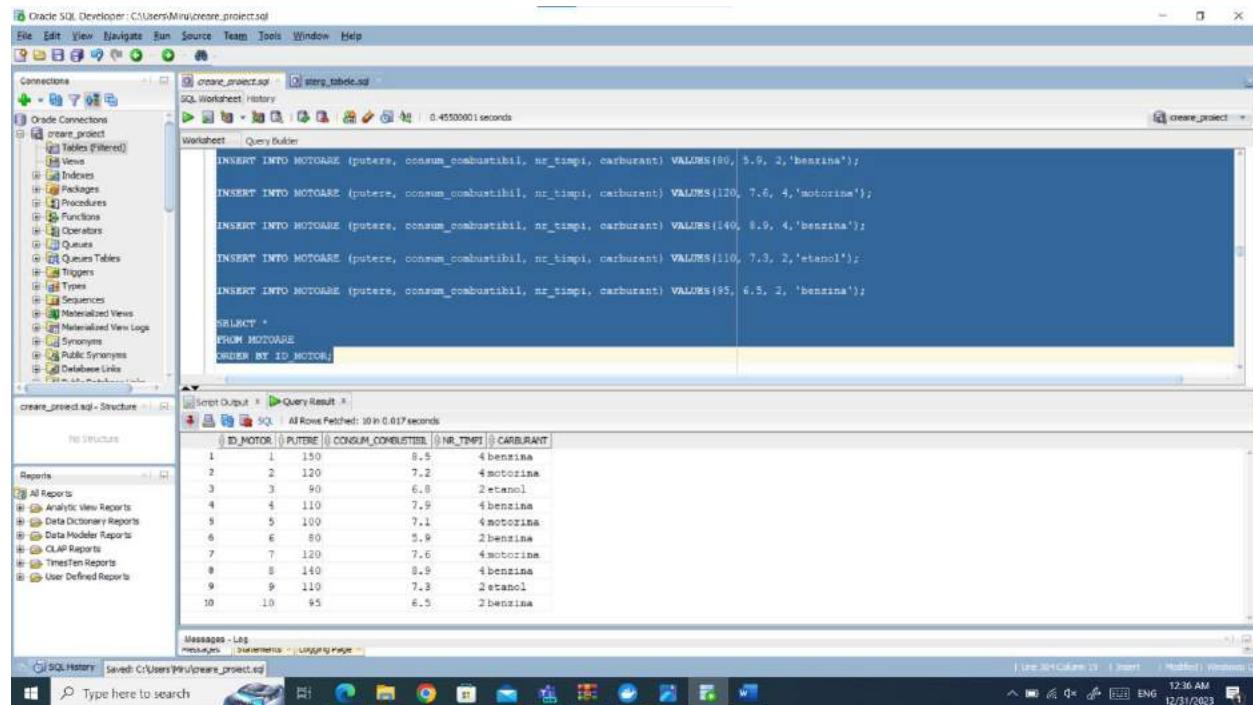
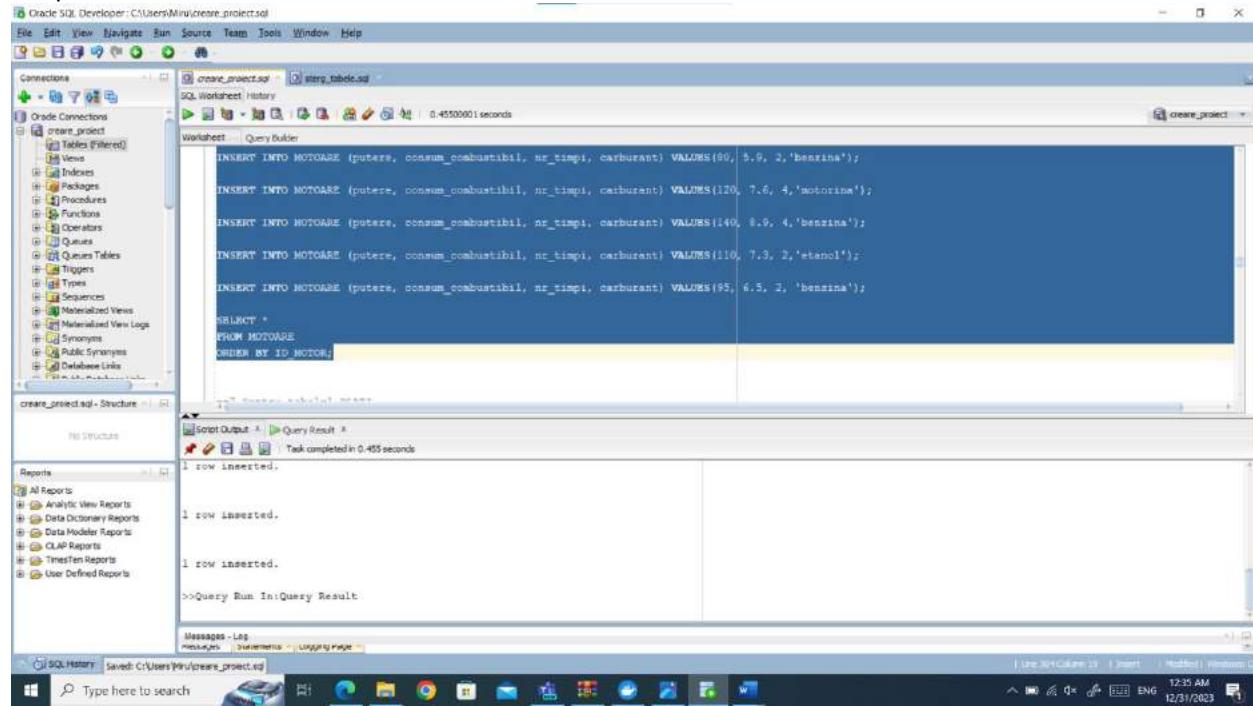
DELETE FROM MOTOARE; INSERT INTO MOTOARE (putere, consum_combustibil, nr_timpuri, carburant) VALUES(100, 8.0, 4,'benzina');
```

In the first screenshot, the 'Script Output' tab shows the results of the last four insert statements, each returning '1 row inserted.' The second screenshot shows the results of all eight insert statements, with the last four also returning '1 row inserted.'. Both screenshots also show the 'Query Result' tab with the message 'Task completed in 0.455 seconds'. The bottom status bar indicates the date and time as '12/31/2003 12:15 AM'.

## Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242



## 7. Pentru tabelul PLATI

➤ Cod SQL

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(10, 56.25, '03-JAN-2022',  
'cash');
```

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(2, 98.50, '05-JAN-2022',  
'card');
```

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(13, 75.00, '07-FEB-2022',  
'cash');
```

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(10, 32.80, '09-MAR-2022',  
'card');
```

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(12, 62.10, '11-APR-2022',  
'cash');
```

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(5, 128.45, '13-APR-2022',  
'card');
```

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(2, 93.20, '15-JUN-2022',  
'cash');
```

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(10, 44.70, '17-JUL-2022',  
'card');
```

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(13, 81.90, '19-JUL-2022',  
'cash');
```

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(8, 56.00, '21-NOV-2022',  
'card');
```

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(8, 150.00, '20-JAN-2022',  
'card');
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(8, 6.00, '13-NOV-2022',  
'card');
```

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(8, 231.00, '14-NOV-2022',  
'card');
```

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(8, 80.00, '15-FEB-2022',  
'card');
```

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(8, 59.00, '12-JUL-2022',  
'card');
```

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(8, 56.00, '21-JAN-2022',  
'card');
```

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(7, 50.00, '28-FEB-2022',  
'card');
```

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(9, 52.00, '21-JAN-2022',  
'card');
```

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(9, 56.00, '23-APR-2022',  
'card');
```

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(8, 6.00, '22-MAY-2022',  
'card');
```

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(8, 46.00, '20-MAR-2022',  
'card');
```

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(8, 5.00, '21-APR-2022',  
'card');
```

## Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
INSERT INTO PLATI (id_plata,id_utilizator, cost, data, modalitate_plata) VALUES(23,8, 56.00, '01-NOV-2022', 'card');
```

```
INSERT INTO PLATI (id_plata,id_utilizator, cost, data, modalitate_plata) VALUES(24,8, 56.00, '01-NOV-2022', 'cash');
```

```
INSERT INTO PLATI (id_plata,id_utilizator, cost, data, modalitate_plata) VALUES(26,8, 56.00, '01-NOV-2022', 'cash');
```

```
INSERT INTO PLATI (id_plata,id_utilizator, cost, data, modalitate_plata) VALUES(27,8, 56.00, '01-NOV-2022', 'cash');
```

```
SELECT *  
FROM PLATI  
ORDER BY ID_PLATA;
```

### ➤ Print-Screen

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab contains a SQL script for inserting data into the PLATI table:

```
--> Fentru tabelul PLATI  
-----  
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(10, 56.25, '03-JAN-2022', 'cash');  
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(2, 98.00, '05-JAN-2022', 'card');  
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(13, 75.00, '07-FEB-2022', 'cash');  
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(10, 32.00, '09-MAR-2022', 'card');  
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(12, 62.10, '11-APR-2022', 'cash');
```

The 'Script Output' tab shows the results of the insertions:

```
1 row inserted.  
1 row inserted.  
1 row inserted.  
1 row inserted.  
1 row inserted.
```

The bottom status bar indicates the task completed in 0.961 seconds.

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the project structure for 'create\_project'. The 'Tables' section is expanded, showing 'Tables (Filtered)'. The 'create\_project.sql' script is open in the central workspace, containing a series of INSERT statements into a table named PLATI. The 'Script Output' pane at the bottom shows the results of the execution.

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(5, 128.45, '13-AVR-2022', 'card');

INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(3, 99.20, '15-JUN-2022', 'cash');

INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(10, 44.70, '17-JUL-2022', 'card');

INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(13, 81.90, '19-JUL-2022', 'cash');

INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(8, 56.00, '21-NOV-2022', 'card');

INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(9, 150.00, '20-JAN-2022', 'card');

INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(6, 6.00, '13-NOV-2022', 'card');

INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(6, 231.00, '14-NOV-2022', 'card');

INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(8, 80.00, '15-FEB-2022', 'card');

INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(8, 59.00, '12-JUL-2022', 'card');

INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(8, 56.00, '21-JAN-2022', 'card');
```

1 row inserted.

This screenshot shows the same Oracle SQL Developer session after a SELECT statement has been added to the query. The 'create\_project.sql' script now includes both INSERT and SELECT statements. The 'Script Output' pane shows the results of the execution.

```
INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(9, 52.00, '21-JAN-2022', 'card');

INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(9, 56.00, '23-APR-2022', 'card');

INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(6, 6.00, '22-MAY-2022', 'card');

INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(6, 46.00, '20-MAR-2022', 'card');

INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(8, 5.00, '21-APR-2022', 'card');

INSERT INTO PLATI (id_plate,id_utilizator, cost, data, modalitate_plata) VALUES(23,5, 56.00, '01-NOV-2022', 'card');

INSERT INTO PLATI (id_plate,id_utilizator, cost, data, modalitate_plata) VALUES(24,8, 56.00, '01-NOV-2022', 'cash');

INSERT INTO PLATI (id_plate,id_utilizator, cost, data, modalitate_plata) VALUES(24,8, 56.00, '01-NOV-2022', 'cash');

INSERT INTO PLATI (id_plate,id_utilizator, cost, data, modalitate_plata) VALUES(27,6, 56.00, '01-NOV-2022', 'cash');

INSERT INTO PLATI (id_utilizator, cost, data, modalitate_plata) VALUES(6, 56.00, '01-NOV-2022', 'cash');

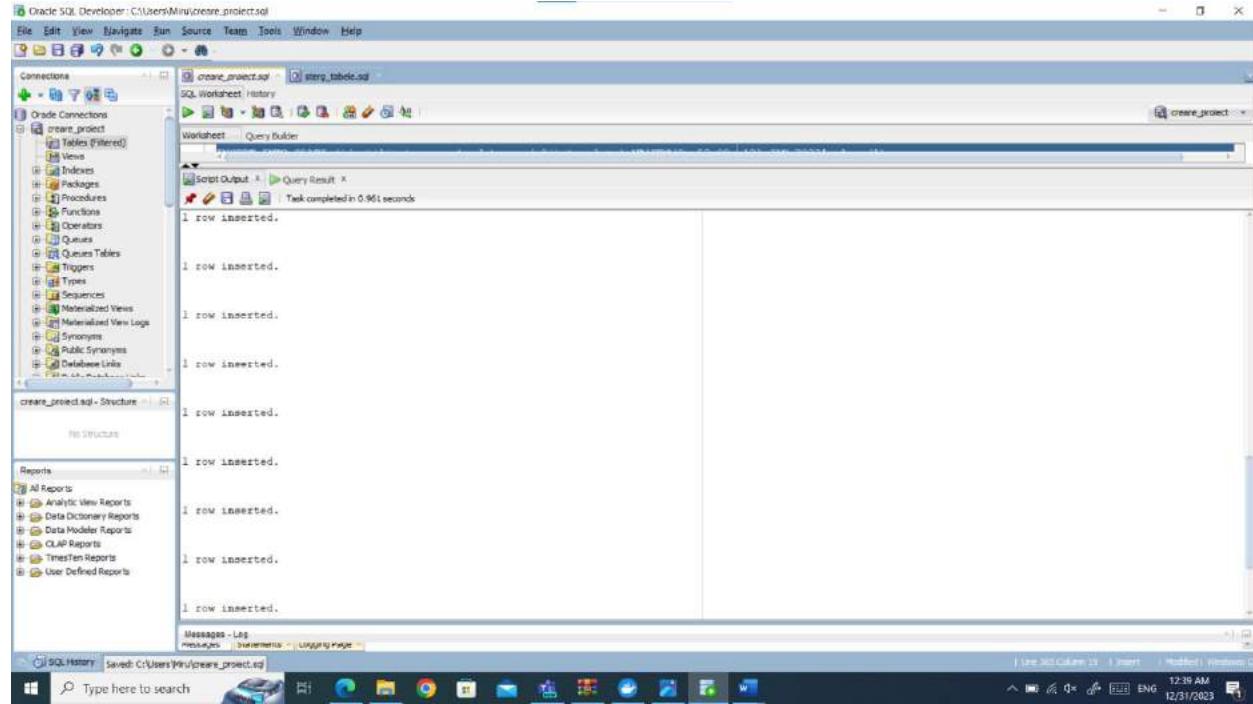
SELECT *
FROM PLATI
ORDER BY ID_PLATA;
```

1 row inserted.

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242



ID_PLATA	ID_UTILIZATOR	COST	DATA	MODALITATE_PLATA
1	1	10	56.25-03-JUN-22	cash
2	2	98.505-1-JUN-22	card	
3	3	13	75.07-FEB-22	cash
4	4	10	32.009-MAR-22	card
5	5	12	62.111-APR-22	cash
6	6	5 128.45 13-APR-22	card	
7	7	2	93.215-1-JUN-22	cash
8	8	10	44.717-JUL-22	card
9	9	13	81.919-JUL-22	cash
10	10	6	56.21-MAY-22	card
11	11	8	150.20-JUN-22	card
12	12	8	613-NOV-22	card
13	13	8	231.14-NOV-22	card
14	14	8	60.15-PEB-22	card
15	15	8	59.12-JUL-22	card
16	16	8	56.21-JAN-22	card
17	17	7	50.20-FEB-22	card
18	18	9	52.21-JAN-22	card
19	19	9	56.23-APR-22	card
20	20	8	6 22-MAY-22	card
21	21	8	46.20-MAR-22	card
22	22	8	5 21-APR-22	card
23	23	8	56.01-NOV-22	card
24	24	8	56.01-NOV-22	cash

## 8. Pentru tabelul MOTOCICLETE

➤ Cod SQL

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (11, 1, 1, 1, 5000, 'B-01-ABC', 'Honda', 'CBR600RR', 'foarte buna', 2018,10);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (11, 1, 2, 2, 7000, 'B-02-DEF', 'Yamaha', 'R6', 'buna', 2016,15);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (2, 2, 3, 3, 10000, 'B-03-GHI', 'Kawasaki', 'Ninja ZX-6R', 'foarte buna', 2020,21);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (2, 2, 4, 4, 8000, 'B-04-JKL', 'Suzuki', 'GSX-R750', 'buna', 2014,17);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (13, 3, 5, 5, 12000, 'B-05-MNO', 'Ducati', 'Panigale V4', 'excelenta', 2019,19);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (13, 3, 6, 6, 9000, 'B-06-PQR', 'Triumph', 'Street Triple RS', 'buna', 2015,18);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (9, 4, 7, 7, 6000, 'B-07-STU', 'BMW', 'S1000RR', 'foarte buna', 2017,19);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (9, 4, 8, 8, 15000, 'B-08-VWX', 'KTM', 'RC 390', 'buna', 2016,14);
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (5, 5, 9, 9, 11000, 'B-09-YZ1', 'Harley-Davidson', 'Sportster 1200', 'buna', 2013, 13);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (5, 5, 10, 10, 20000, 'B-10-234', 'Indian', 'Chieftain Dark Horse', 'foarte buna', 2018, 12);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (5, 5, 10, 10, 20000, 'B-10-234', 'Indian', 'Chieftain Dark Horse', 'foarte buna', 2018, 11);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (5, 5, 10, 10, 20000, 'B-10-234', 'Indian', 'Chieftain Dark Horse', 'foarte buna', 2018, 18);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (5, 5, 10, 10, 20000, 'B-10-234', 'Indian', 'Chieftain Dark Horse', 'foarte buna', 2018, 19);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (5, 5, 10, 10, 20000, 'B-10-234', 'Indian', 'Chieftain Dark Horse', 'foarte buna', 2018, 21);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (5, 5, 10, 10, 20000, 'B-10-234', 'Indian', 'Chieftain Dark Horse', 'foarte buna', 2018, 21);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (5, 5, 10, 10, 20000, 'B-10-234', 'Indian', 'Chieftain Dark Horse', 'foarte buna', 2018, 20);
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (2, 5, 3, 3, 10000, 'B-03-GHI', 'Kawasaki', 'Ninja ZX-6R', 'foarte buna', 2020, 21);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (2, 5, 4, 4, 8000, 'B-04-JKL', 'Suzuki', 'GSX-R750', 'buna', 2014, 17);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (13, 5, 5, 5, 12000, 'B-05-MNO', 'Ducati', 'Panigale V4', 'excelenta', 2019, 19);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (13, 5, 6, 6, 9000, 'B-06-PQR', 'Triumph', 'Street Triple RS', 'buna', 2015, 18);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (9, 5, 7, 7, 6000, 'B-07-STU', 'BMW', 'S1000RR', 'foarte buna', 2017, 19);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (9, 5, 8, 8, 15000, 'B-08-VWX', 'KTM', 'RC 390', 'buna', 2016, 14);
```

```
SELECT *
```

```
FROM MOTOCICLETE
```

```
ORDER BY ID_MOTOCICLETA;
```

➤ Print-Screen

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. The title bar says "Oracle SQL Developer: C:\Users\Miru\create\_project.oe". The left sidebar shows a tree view of the project structure under "create\_project", including "Tables (Filtered)", "Views", "Indexes", "Packages", "Procedures", "Functions", "Operators", "Quotas", "Queues Tables", "Triggers", "Types", "Sequences", "Materialized Views", "Materialized View Logs", "Synonyms", "Public Synonyms", and "Database Links". The main workspace is titled "create\_project.sql" and contains the following SQL code:

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometri, nr_immatriculare, marca, model, stare, an, capacitate_reservor)
VALUES (1, 1, 1, 1, 5000, 'B-01-ABC', 'Honda', 'CBR600RR', 'foarte buna', 2016,10);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometri, nr_immatriculare, marca, model, stare, an, capacitate_reservor)
VALUES (1, 1, 2, 2, 7000, 'B-02-DEF', 'Yamaha', 'R6', 'buna', 2016,15);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometri, nr_immatriculare, marca, model, stare, an, capacitate_reservor)
VALUES (2, 2, 3, 3, 10000, 'B-03-GHI', 'Kawasaki', 'Ninja ZX-6R', 'foarte buna', 2020,21);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometri, nr_immatriculare, marca, model, stare, an, capacitate_reservor)
VALUES (2, 2, 4, 4, 8000, 'B-04-JKL', 'Suzuki', 'GSX-R750', 'buna', 2014,17);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometri, nr_immatriculare, marca, model, stare, an, capacitate_reservor)
VALUES (10, 3, 45, 5, 12000, 'B-05-MNO', 'Ducati', 'Panigale V4', 'exceleanta', 2019,19);
```

The "Script Output" tab shows the results of the execution: "1 row inserted.", "1 row inserted.", "1 row inserted.", "1 row inserted.", and "1 row inserted.". The status bar at the bottom right indicates "12:41 AM 12/31/2023".

This screenshot shows the continuation of the project creation process. The title bar remains "Oracle SQL Developer: C:\Users\Miru\create\_project.oe". The left sidebar is identical to the previous screenshot. The main workspace now contains a longer sequence of SQL INSERT statements:

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometri, nr_immatriculare, marca, model, stare, an, capacitate_reservor)
VALUES (13, 3, 6, 6, 9000, 'B-06-PQR', 'Triumph', 'Street Triple RS', 'buna', 2015,18);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometri, nr_immatriculare, marca, model, stare, an, capacitate_reservor)
VALUES (9, 4, 7, 7, 6000, 'B-07-STU', 'BMW', 'S1000RR', 'foarte buna', 2017,19);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometri, nr_immatriculare, marca, model, stare, an, capacitate_reservor)
VALUES (9, 4, 8, 8, 15000, 'B-08-VWX', 'KTM', 'RC 390', 'buna', 2016,14);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometri, nr_immatriculare, marca, model, stare, an, capacitate_reservor)
VALUES (5, 5, 9, 9, 11000, 'B-09-YZI', 'Harley-Davidson', 'Sportster 1200', 'buna', 2013,13);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometri, nr_immatriculare, marca, model, stare, an, capacitate_reservor)
VALUES (5, 5, 10, 10, 20000, 'B-10-ZXA', 'Indian', 'Chieftain Dark Horse', 'foarte buna', 2018,12);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometri, nr_immatriculare, marca, model, stare, an, capacitate_reservor)
VALUES (5, 5, 10, 10, 20000, 'B-10-234', 'Indian', 'Chieftain Dark Horse', 'foarte buna', 2018,11);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometri, nr_immatriculare, marca, model, stare, an, capacitate_reservor)
VALUES (5, 5, 10, 10, 20000, 'B-10-234', 'Indian', 'Chieftain Dark Horse', 'foarte buna', 2018,18);
```

The "Script Output" tab shows the results of the execution: "1 row inserted.", and "1 row inserted.". The status bar at the bottom right indicates "12:41 AM 12/31/2023".

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. The title bar says "Oracle SQL Developer : C:\Users\Miru\create\_project.sql". The left sidebar shows a tree view of the project structure under "create\_project", including "Tables (Filtered)", "Views", "Indexes", "Packages", "Procedures", "Functions", "Operators", "Quotas", "Queues", "Triggers", "Types", "Sequences", "Materialized Views", "Materialized View Logs", "Synonyms", "Public Synonyms", and "Database Links". The main workspace is titled "sterg\_table.sql" and contains the following SQL code:

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_fax, kilometraj, nr_immatriculare, marca, model, stare, an_capacitate_reservor)
VALUES (5, 5, 10, 10, 30000, 'B-10-234', 'Indian', 'Chieftain Dark Horse', 'foarte buna', 2010,19);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_fax, kilometraj, nr_immatriculare, marca, model, stare, an_capacitate_reservor)
VALUES (5, 5, 10, 10, 30000, 'B-10-234', 'Indian', 'Chieftain Dark Horse', 'foarte buna', 2018,21);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_fax, kilometraj, nr_immatriculare, marca, model, stare, an_capacitate_reservor)
VALUES (5, 5, 10, 10, 30000, 'B-10-234', 'Indian', 'Chieftain Dark Horse', 'foarte buna', 2018,21);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_fax, kilometraj, nr_immatriculare, marca, model, stare, an_capacitate_reservor)
VALUES (5, 5, 10, 10, 30000, 'B-10-234', 'Indian', 'Chieftain Dark Horse', 'foarte buna', 2018,20);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_fax, kilometraj, nr_immatriculare, marca, model, stare, an_capacitate_reservor)
VALUES (2, 5, 3, 3, 10000, 'B-03-GHZ', 'Kawasaki', 'Ninja ZX-6R', 'foarte buna', 2020,21);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_fax, kilometraj, nr_immatriculare, marca, model, stare, an_capacitate_reservor)
VALUES (2, 5, 4, 4, 8000, 'B-04-JKL', 'Suzuki', 'GSX-R750', 'buna', 2014,17);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_fax, kilometraj, nr_immatriculare, marca, model, stare, an_capacitate_reservor)
VALUES (13, 5, 5, 5, 12000, 'B-05-MNO', 'Ducati', 'Panigale V4', 'exceleenta', 2019,19);

1 row inserted.
```

The "Script Output" tab shows the message "Task completed in 0.788 seconds". The status bar at the bottom right shows "12:42 AM 12/31/2023".

This screenshot shows the continuation of the project "create\_project" in Oracle SQL Developer. The title bar is the same: "Oracle SQL Developer : C:\Users\Miru\create\_project.sql". The left sidebar shows the same project structure. The main workspace is titled "sterg\_table.sql" and contains more SQL code:

```
VALUES (2, 5, 3, 3, 10000, 'B-03-GHZ', 'Kawasaki', 'Ninja ZX-6R', 'foarte buna', 2020,21);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_fax, kilometraj, nr_immatriculare, marca, model, stare, an_capacitate_reservor)
VALUES (2, 5, 4, 4, 8000, 'B-04-JKL', 'Suzuki', 'GSX-R750', 'buna', 2014,17);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_fax, kilometraj, nr_immatriculare, marca, model, stare, an_capacitate_reservor)
VALUES (13, 5, 5, 5, 12000, 'B-05-MNO', 'Ducati', 'Panigale V4', 'exceleenta', 2019,19);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_fax, kilometraj, nr_immatriculare, marca, model, stare, an_capacitate_reservor)
VALUES (13, 5, 6, 6, 9000, 'B-06-PQR', 'Triumph', 'Street Triple RS', 'buna', 2015,18);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_fax, kilometraj, nr_immatriculare, marca, model, stare, an_capacitate_reservor)
VALUES (9, 5, 7, 7, 6000, 'B-07-STW', 'BMW', 'S1000RR', 'foarte buna', 2017,19);

INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_fax, kilometraj, nr_immatriculare, marca, model, stare, an_capacitate_reservor)
VALUES (9, 5, 8, 8, 15000, 'B-08-VWX', 'Yamaha', 'RC 360', 'buna', 2016,14);

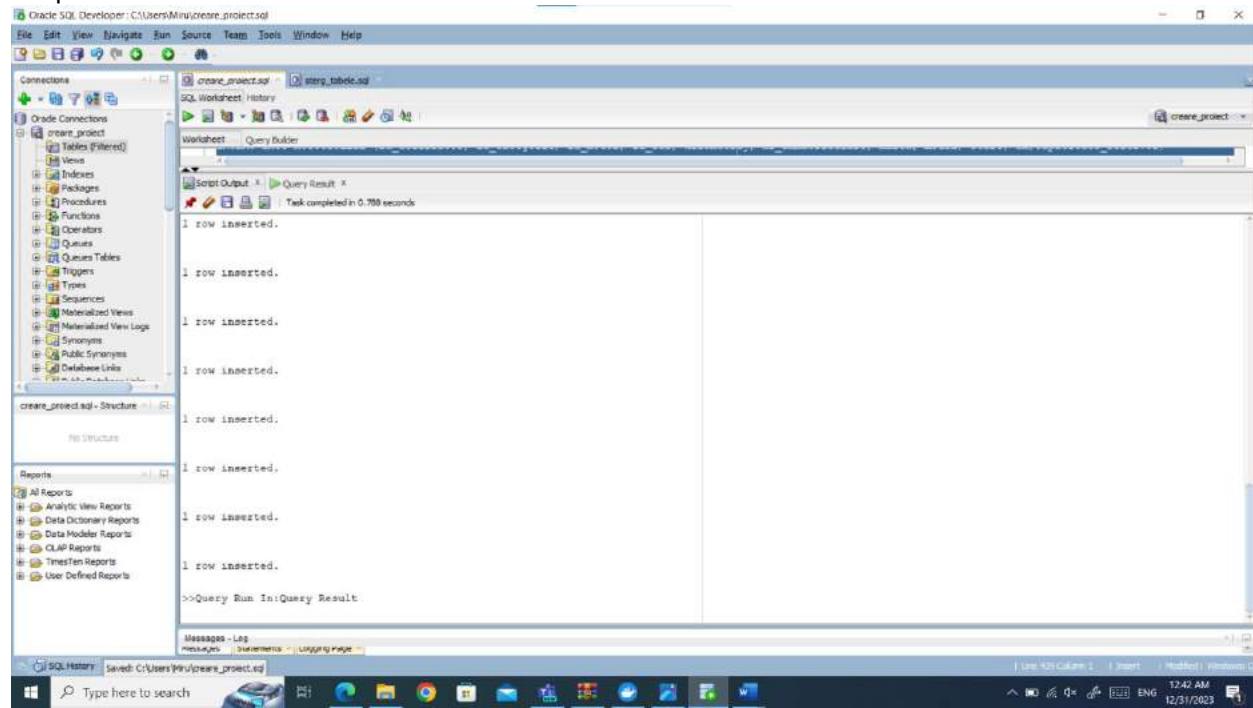
SELECT *
FROM MOTOCICLETE
ORDER BY ID MOTOCICLETA;
```

The "Script Output" tab shows the message "Task completed in 0.788 seconds". The status bar at the bottom right shows "12:42 AM 12/31/2023".

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242



ID_MOTOCICLETA	ID_UTILIZATOR	ID_CATEGORIE	ID_MOTOR	ID_PAR	CAPACITATE_BENZINOVOR	KILOMETRAG	NR_DMATRICULARE	MARCA	MODEL	STARE	MN	
1	1	11	1	1	10	5000	B-01-ABC	Honda	CBR600RR	foarte buna	2018	
2	2	11	1	2	2	15	7000	B-02-DEF	Yamaha	R6	buna	2016
3	3	2	2	3	3	21	16000	B-03-GHI	Kawasaki	Ninja ZX-6R	foarte buna	2020
4	4	2	2	4	4	17	8000	B-04-JKL	Suzuki	GSX-R750	buna	2014
5	5	13	3	6	6	18	9000	B-05-PQR	Triumph	Street Triple RS	buna	2015
6	6	9	4	7	7	19	6000	B-07-STU	BMW	s1000RR	foarte buna	2017
7	7	9	4	8	8	14	15000	B-08-VWX	KTM	RC 390	buna	2016
8	8	5	5	9	9	13	11000	B-09-YZ1	Harley-Davidson	Sportster 1200	buna	2013
9	9	5	5	10	10	12	20000	B-10-Z34	Indian	Chieftain Dark Horse	foarte buna	2018
10	10	5	5	10	10	11	20000	B-10-Z34	Indian	Chieftain Dark Horse	foarte buna	2018
11	11	5	5	10	10	18	20000	B-10-Z34	Indian	Chieftain Dark Horse	foarte buna	2018
12	12	5	5	10	10	19	20000	B-10-Z34	Indian	Chieftain Dark Horse	foarte buna	2018
13	13	5	5	10	10	21	20000	B-10-Z34	Indian	Chieftain Dark Horse	foarte buna	2018
14	14	5	5	10	10	21	20000	B-10-Z34	Indian	Chieftain Dark Horse	foarte buna	2018
15	15	5	5	10	10	20	20000	B-10-Z34	Indian	Chieftain Dark Horse	foarte buna	2018
16	16	2	5	3	3	21	16000	B-03-GHI	Kawasaki	Ninja ZX-6R	foarte buna	2020
17	17	2	5	4	4	17	8000	B-04-JKL	Suzuki	GSX-R750	buna	2014
18	18	13	5	5	5	19	12000	B-05-MNO	Ducati	Panigale V4	excellenta	2019
19	19	13	5	6	6	18	9000	B-06-PQR	Triumph	Street Triple RS	buna	2015
20	20	9	5	7	7	19	6000	B-07-STU	BMW	s1000RR	foarte buna	2017
21	21	9	5	8	8	14	15000	B-08-VWX	KTM	RC 390	buna	2016

## 9. Pentru tabelul INTRETNITERI

➤ Cod SQL

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
```

```
VALUES (11, 2, 3, 'Schimb ulei', 'Schimb de ulei la motor', '15-MAR-2022');
```

```
INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
```

```
VALUES(12, 13, 4, 'Inlocuire baterie', 'Inlocuirea bateriei vechi cu una noua', '27-DEC-2021');
```

```
INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
```

```
VALUES(13, 13, 5, 'Schimb cauciucuri', 'Schimb de anvelope', '02-APR-2022');
```

```
INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
```

```
VALUES(4, 5, 6, 'Curatare carburator', 'Curatare carburator si sistem de injectie', '18-OCT-2021');
```

```
INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
```

```
VALUES(15, 6, 7, 'Inlocuire placute frana', 'Inlocuire placute de frana', '01-FEB-2022');
```

```
INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
```

```
VALUES(6, 7, 8, 'Inlocuire lant', 'Inlocuirea lantului de transmisie', '20-NOV-2021');
```

```
INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
```

```
VALUES(9, 8, 9, 'Reglare suspensie', 'Reglarea suspensiei fata/spate', '05-APR-2022');
```

```
INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
```

```
VALUES(3, 9, 10, 'Inlocuire bujie', 'Inlocuirea bujiei cu una noua', '13-MAY-2022');
```

```
INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
```

```
VALUES(9, 10, 1, 'Curatare filtru aer', 'Curatare filtru de aer', '21-SEP-2021');
```

```
INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
```

```
VALUES(10, 11, 2, 'Verificare sistem electric', 'Verificare si testare sistem electric', '10-DEC-2021');
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
VALUES(11, 13, 4, 'Inlocuire senzor temperatura', 'Inlocuirea senzorului de temperatura', '22-JUN-2022');
```

```
INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
VALUES(12, 5, 5, 'Schimb lichid frana', 'Schimb de lichid de frana', '28-JUN-2022');
```

```
INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
VALUES(12, 5, 5, 'Schimb lichid frana', 'Schimb de lichid de frana', '28-JUN-2022');
```

```
INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
VALUES(12, 5, 5, 'Schimb lichid frana', 'Schimb de lichid de frana', '28-JUN-2022');
```

```
SELECT *
FROM INTRETINERI
ORDER BY ID_INTRETINERE;
```

➤ Print-Screen

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the project structure for 'create\_project' with various database objects like tables, procedures, and triggers. The main workspace contains two tabs: 'create\_project.sql' and 'sterg\_tablele.sql'. The 'create\_project.sql' tab is active and shows a script with several INSERT statements into a table named 'INTRETINERI'. The script includes comments indicating the purpose of each insert (e.g., changing oil, replacing battery, changing oil filter). The 'Script Output' tab shows the results of the execution: five rows inserted successfully. The bottom status bar indicates the task completed in 0.677 seconds at 10:00 AM on 12/31/2023.

```
--9. Pentru tabela INTRETINERI
INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plate, nume, descriere, data)
VALUES (1, 2, 3, 'Schimb ulei', 'Schimb de ulei la motor', '15-MAR-2022');

INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plate, nume, descriere, data)
VALUES (12, 13, 4, 'Inlocuire baterie', 'Inlocuirea bateriei vechi cu una noua', '17-OCT-2021');

INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plate, nume, descriere, data)
VALUES (13, 13, 5, 'Schimb cauciucuri', 'Schimb de anvelope', '02-APR-2022');

INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plate, nume, descriere, data)
VALUES (4, 5, 6, 'Curatare carburator', 'Curatare carburator si sistem de inietie', '18-OCT-2021');
```

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the database structure for the 'create\_project' schema, including tables, views, indexes, packages, procedures, functions, operators, queues, triggers, types, sequences, materialized views, and public synonyms. The central workspace contains a 'Query Builder' window with the following SQL code:

```
454
455 INSERT INTO INTRETNITERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
456 VALUES(15, 6, 7, 'Inlocuire placute frane', 'Inlocuire placute de frane', '01-FEB-2022');
457
458 INSERT INTO INTRETNITERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
459 VALUES(6, 7, 8, 'Inlocuire lant', 'Inlocuirea lantului de transmisie', '20-NOV-2021');
460
461 INSERT INTO INTRETNITERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
462 VALUES(9, 8, 9, 'Reglare suspensie', 'Reglarea suspensiiei fata/spate', '05-APR-2022');
463
464 INSERT INTO INTRETNITERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
465 VALUES(3, 9, 10, 'Inlocuire bujie', 'Inlocuirea bujiei cu una noua', '13-MAY-2022');
466
```

The 'Script Output' tab shows the results of the execution: "1 row inserted." repeated five times.

This screenshot shows the same Oracle SQL Developer session after inserting more entries into the 'INTRETNITERI' table. The workspace now contains the following SQL code:

```
467 INSERT INTO INTRETNITERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
468 VALUES(9, 10, 1, 'Curatare filtru aer', 'Curatare filtru de aer', '21-SEP-2021');
469
470 INSERT INTO INTRETNITERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
471 VALUES(10, 11, 2, 'Verificare sistem electric', 'Verificare si testare sistem electric', '19-OCT-2021');
472
473 INSERT INTO INTRETNITERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
474 VALUES(11, 12, 4, 'Inlocuire senzor temperatura', 'Inlocuirea senzorului de temperatura', '22-JUN-2022');
475
476 INSERT INTO INTRETNITERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
477 VALUES(12, 5, 5, 'Schimb lichid frana', 'Schimb de lichid de frana', '28-JUN-2022');
478
479 INSERT INTO INTRETNITERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
```

The 'Script Output' tab shows the results of the execution: "1 row inserted." repeated six times.

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, under 'Connections', there is a connection named 'create\_project'. The 'Tables (Filtered)' section shows a single table named 'sterg\_tablele'. In the central 'Worksheet' pane, a script named 'create\_project.sql' is being run. The script contains several INSERT statements into a table named 'INTRETINERI'. The first few lines of the script are:

```
476 INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
477 VALUES(12, 5, 5, 'Schimb lichid frana', 'Schimb de lichid de frana', '28-JUN-2022');
478
479 INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
480 VALUES(12, 5, 5, 'Schimb lichid frana', 'Schimb de lichid de frana', '28-JUN-2022');
481
482 INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
483 VALUES(12, 5, 5, 'Schimb lichid frana', 'Schimb de lichid de frana', '28-JUN-2022');
484
485
486
487 FROM INTRETINERI
488 ORDER BY ID INTRETINERI;
```

Below the script, the 'Script Output' and 'Query Result' panes show the results of the insertions. The 'Script Output' pane says '1 row inserted.' for each of the four rows. The 'Query Result' pane shows the same four rows of data from the 'INTRETINERI' table.

This screenshot shows the same Oracle SQL Developer session after the data has been inserted. The 'Worksheet' pane now displays the full script, including the SELECT statement at the end:

```
400 VALUES(12, 5, 5, 'Schimb lichid frana', 'Schimb de lichid de frana', '28-JUN-2022');
401
402 INSERT INTO INTRETINERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
403 VALUES(12, 5, 5, 'Schimb lichid frana', 'Schimb de lichid de frana', '28-JUN-2022');
404
405
406
407 FROM INTRETINERI
408 ORDER BY ID INTRETINERI;
```

The 'Script Output' pane shows 'All Rows Fetched: 14 in 0.018 seconds'. The 'Query Result' pane displays 14 rows of data from the 'INTRETINERI' table, listing various maintenance items like oil changes, battery replacements, and filter cleanings, along with their dates.

## 10. Pentru tabelul ACHIZTII

➤ Cod SQL

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
INSERT INTO ACHIZITII (id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES  
(2, 4, 5, 'Motocicleta Yamaha R6', 'Motocicleta sportiva cu 600cc, culoare rosie', '15-MAR-2022');
```

```
INSERT INTO ACHIZITII (id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES  
(1, 3, 4, 'Motocicleta Harley Davidson Fat Boy', 'Chopper clasic, culoare neagra mata', '02-APR-2021');
```

```
INSERT INTO ACHIZITII (id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES  
(10, 2, 2, 'Casca HJC RPHA 11', 'Casca integrala pentru motociclete, culoare alb?', '08-MAY-2022');
```

```
INSERT INTO ACHIZITII (id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES  
(9, 4, 4, 'Anvelopa Michelin Road 5', 'Anvelopa sportiva pentru motociclete, dimensiune 180/55 R17',  
'20-JAN-2023');
```

```
INSERT INTO ACHIZITII (id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES  
(8, 5, 5, 'Geanta laterală Givi Xstream', 'Geanta de depozitare pentru motociclete, capacitate 25L', '14-FEB-2022');
```

```
INSERT INTO ACHIZITII (id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES  
(7, 6, 3, 'Ulei motocicleta Motul 7100', 'Ulei de motor sintetic, pentru motociclete cu performante  
ridicate', '02-MAR-2022');
```

```
INSERT INTO ACHIZITII (id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES  
(6, 10, 7, 'Manusi moto Alpinestars GP Plus', 'Manusi de piele pentru motociclete, cu protectii la degete si  
încheieturi', '28-APR-2022');
```

```
INSERT INTO ACHIZITII (id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES  
(5, 8, 8, 'Placute frâna Brembo', 'Placute de frâna de înalta performanta pentru motociclete', '04-MAY-  
2022');
```

```
INSERT INTO ACHIZITII (id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

(4, 9, 9, 'Lant moto DID VX', 'Lant de transmisie pentru motociclete, cu tehnologie X-ring si tratament special antirugina', '31-JAN-2022');

```
INSERT INTO ACHIZITII (id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES  
(3, 7, 5, 'Sa motocicleta La Rosa', 'Sa de pasager pentru motociclete, din piele maro si cusaturi decorative',  
'19-FEB-2022');
```

```
INSERT INTO ACHIZITII (id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES  
(2, 1, 1, 'Proteza motor Yamaha R1', 'Proteza de motor pentru motocicleta Yamaha R1', '17-MAR-2021');
```

```
INSERT INTO ACHIZITII (id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES  
(1, 2, 2, 'Perna de gel pentru sa', 'Perna de gel pentru sa de motocicleta, pentru un confort sporit în timpul  
calatoriilor lungi', '09-MAY-2022');
```

```
INSERT INTO ACHIZITII (id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES  
(1, 2, 2, 'Perna de gel pentru sa', 'Perna de gel pentru sa de motocicleta, pentru un confort sporit în timpul  
calatoriilor lungi', '09-MAY-2022');
```

```
INSERT INTO ACHIZITII (id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES  
(1, 2, 2, 'Perna de gel pentru sa', 'Perna de gel pentru sa de motocicleta, pentru un confort sporit în timpul  
calatoriilor lungi', '09-MAY-2022');
```

```
SELECT *
```

```
FROM ACHIZITII
```

```
ORDER BY ID_ACHIZITIE;
```

➤ Print-Screen

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the 'Connections' tree, which includes a project named 'create\_project'. Under 'Tables (filtered)', there is a single table named 'sterg\_table'. The main workspace is titled 'create\_project.sql' and contains the following SQL code:

```
491 INSERT INTO ACHIZITII(id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES
492 (2, 4, 5, 'Motocicleta Yamaha R6', 'Motocicleta sportiva cu 600cc, culoare rosie', '15-MAR-2022');
493
494 INSERT INTO ACHIZITII(id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES
495 (1, 3, 4, 'Motocicleta Harley Davidson Fat Boy', 'Chopper classic, culoare neagra mată', '02-APR-2021');
496
497 INSERT INTO ACHIZITII(id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES
498 (10, 2, 2, 'Casca HJC RPHA 11', 'Casca integrală pentru motociclete, culoare albă', '08-MAR-2022');
499
500 INSERT INTO ACHIZITII(id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES
501 (9, 4, 4, 'Anvelopă Michelin Road 5', 'Anvelopă sportivă pentru motociclete, dimensiune 180/55 R17', '20-JAN-2023');
502
503 INSERT INTO ACHIZITII(id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES
504 (8, 3, 3, 'Geanta laterală Givi Kattrax', 'Geanta de depozitare pentru motociclete, capacitate 25L', '14-FEB-2022');
```

The 'Script Output' tab shows the message '1 row inserted.' indicating the success of the last query.

The screenshot shows the Oracle SQL Developer interface. The title bar indicates the project is 'Saved: C:\Users\Minu\create\_project.ed'. The main workspace displays a SQL Worksheet titled 'create\_project.sql' containing several SQL INSERT statements for creating records in a table named 'ACHIZITII'. The statements insert data for various oil and frame products. The bottom pane shows the results of the execution, with each statement returning '1 row inserted.' The left sidebar shows the project structure with a connection to 'create\_project' and a table named 'Tables (filtered)'. The bottom navigation bar includes tabs for 'Messages - Log', 'Messages', 'Instrumentation', and 'Logging Page'.

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows two instances of Oracle SQL Developer running on a Windows desktop. Both windows have the title bar "Oracle SQL Developer : C:\Users\Miru\create\_project.sql".

The left window displays the following SQL script:

```
523 INSERT INTO ACCESORII (id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES
524 (1, 1, 1, 'Proteza motor Yamaha R1', 'Proteza de motor pentru motociclete Yamaha R1', '17-MAR-2021');
525
526 INSERT INTO ACCESORII (id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES
527 (1, 2, 2, 'Perna de gel pentru sa', 'Perna de gel pentru sa de motocicleta, pentru un confort sporit in timpul calatoriilor lungi', '09-MAY-2022');
528
529 INSERT INTO ACCESORII (id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES
530 (1, 2, 2, 'Perna de gel pentru sa', 'Perna de gel pentru sa de motocicleta, pentru un confort sporit in timpul calatoriilor lungi', '09-MAY-2022');
531
532 INSERT INTO ACCESORII (id_plata, id_utilizator, id_motocicleta, nume, descriere, data) VALUES
533 (1, 2, 2, 'Perna de gel pentru sa', 'Perna de gel pentru sa de motocicleta, pentru un confort sporit in timpul calatoriilor lungi', '09-MAY-2022');
534
535 SELECT *
536 FROM ACCESORII
537 ORDER BY ID_ACCESORII;
```

The right window shows the execution results:

```
1 row inserted.
1 row inserted.
1 row inserted.
```

The bottom status bar indicates "Task completed in 0.64 seconds".

The second window on the desktop also displays the same SQL script and results, with the status bar showing "All Rows Fetched: 14 in 0.02 seconds".

The tables pane on the left of both windows lists the following tables:

- Tables (Filtered)
- Views
- Indexes
- Packages
- Procedures
- Functions
- Operators
- Quotas
- Queues Tables
- Triggers
- Types
- Sequences
- Materialized Views
- Materialized View Logs
- Synonyms
- Public Synonyms
- Database Links

The Reports pane lists various report types, and the bottom status bar shows the date and time as 12/31/2023 10:5 AM.

## 11. Pentru tabelul ACCESORII\_MOTOCICLETE

➤ Cod SQL

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES (1, 2);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(1, 4);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(1, 6);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(2, 1);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(2, 3);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(2, 5);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(3, 1);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(3, 4);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(3, 7);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(4, 3);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(4, 5);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(4, 8);
```

```
SELECT *
```

```
FROM ACCESORII_MOTOCICLETE
```

```
ORDER BY ID_MOTOCICLETA;
```

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

## ➤ Print-Screen

The image displays two side-by-side screenshots of the Oracle SQL Developer interface. Both screenshots show the same workspace configuration with a central 'Worksheet' tab containing an 'Query Builder' panel.

**Worksheet Panel Content:**

```
542 INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesorii) VALUES (1, 2);
543
544 INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesorii) VALUES(1, 4);
545
546 INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesorii) VALUES(1, 6);
547
548 INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesorii) VALUES(2, 1);
549
550 INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesorii) VALUES(2, 3);
551
552 INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesorii) VALUES(2, 5);
```

**Script Output Panel:**

1 row inserted.  
1 row inserted.

**Messages Panel:**

Task completed in 0.619 seconds

**Bottom Status Bar:**

Line 500 Column 25 Insert Method: Inserted 107 AM 12/31/2023 ENG

## Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows three separate sessions running simultaneously in Oracle SQL Developer:

- Session 1 (Top):** A single INSERT query into the `ACCESORII_MOTOCICLETE` table, which inserts three rows with IDs 4, 5, and 6 respectively.
- Session 2 (Middle):** A single SELECT query from the `ACCESORII_MOTOCICLETE` table, ordered by `ID_MOTOCICLETA`. The result set contains 12 rows, showing pairs of `ID_MOTOCICLETA` and `ID_ACCESORIU`.
- Session 3 (Bottom):** A single SELECT query from the `FRANE_MOTOCICLETE` table, ordered by `ID_MOTOCICLETA`. The result set contains 12 rows, showing pairs of `ID_MOTOCICLETA` and `ID_FRANA`.

The bottom session's results are displayed in a table:

ID_MOTOCICLETA	ID_FRANA
1	1
2	1
3	1
4	2
5	2
6	2
7	3
8	3
9	3
10	4
11	4
12	4

## 12. Pentru tabelul FRANE\_MOTOCICLETE

➤ Cod SQL

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES (1, 1);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(1, 3);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(1, 5);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(2, 2);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(2, 4);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(2, 6);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(3, 3);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(3, 5);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(3, 7);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(4, 4);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(4, 6);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(4, 8);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(5, 5);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(5, 7);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(5, 9);
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(6, 6);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(6, 8);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(6, 10);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(7, 7);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(7, 9);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(7, 1);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(8, 8);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(8, 10);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(8, 2);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(9, 9);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(9, 1);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(10,1);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(10,2);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES (11, 1);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(11, 3);
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(11, 5);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(12, 2);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(12, 4);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(12, 6);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(13, 3);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(13, 5);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(13, 7);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(14, 4);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(14, 6);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(14, 8);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(15, 5);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(15, 7);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(15, 9);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(10, 6);
```

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frama) VALUES(11, 8);
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
INSERT INTO FRANE_MOTOCICLETE (id_motocicleta, id_frana) VALUES(12, 10);
```

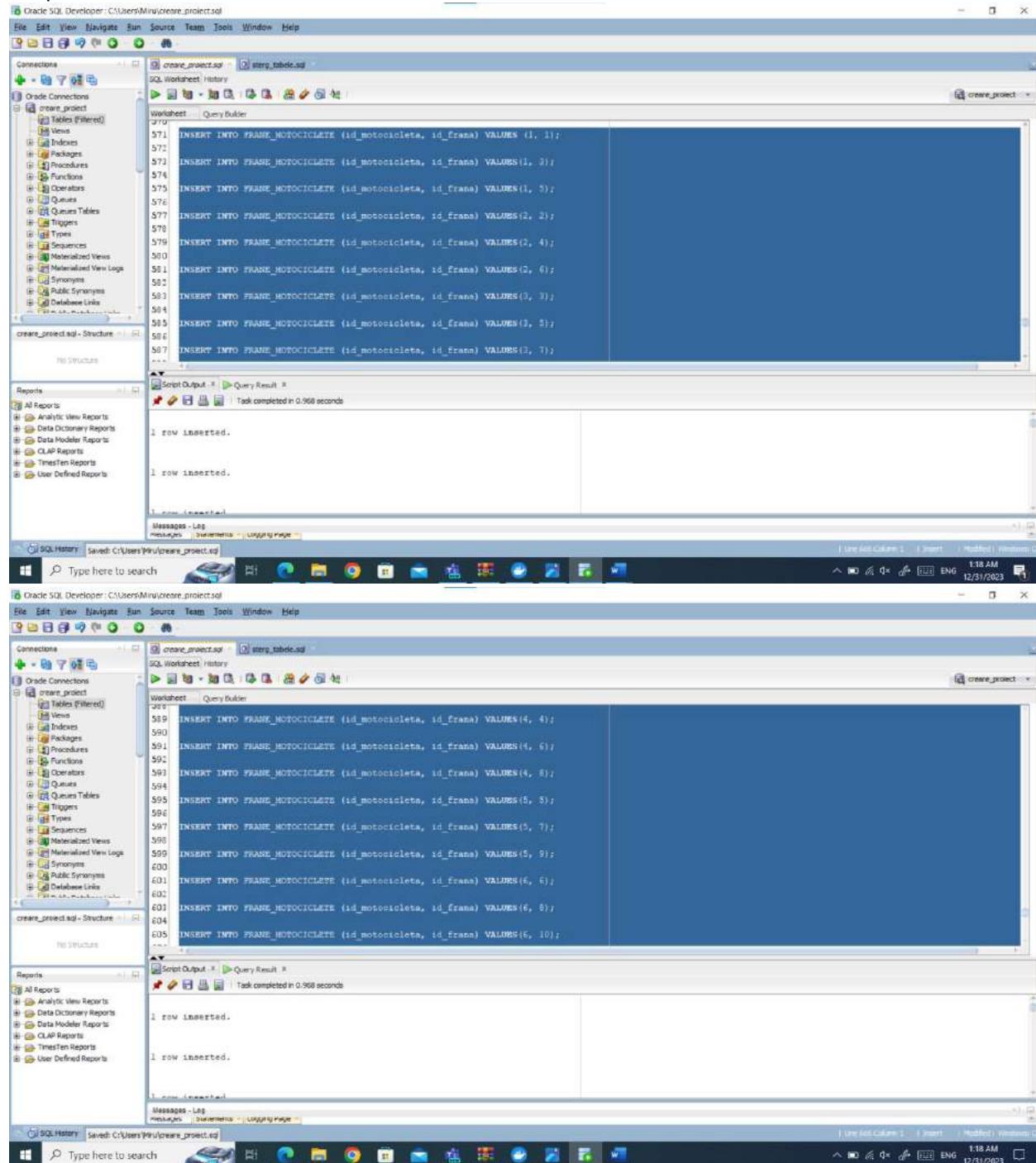
```
SELECT *
FROM FRANE_MOTOCICLETE
ORDER BY ID_MOTOCICLETA;
```

➤ Print-Screen

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242



The image shows two side-by-side screenshots of the Oracle SQL Developer application. Both windows have the title bar "Oracle SQL Developer : C:\Users\Miru\create\_project.sql". The left window displays a script named "create\_project.sql" with the following content:

```
571 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frama) VALUES (1, 1);
572
573 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frama) VALUES(1, 3);
574
575 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frama) VALUES(1, 5);
576
577 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frama) VALUES(2, 2);
578
579 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frama) VALUES(2, 4);
580
581 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frama) VALUES(2, 6);
582
583 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frama) VALUES(3, 3);
584
585 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frama) VALUES(3, 5);
586
587 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frama) VALUES(3, 7);
```

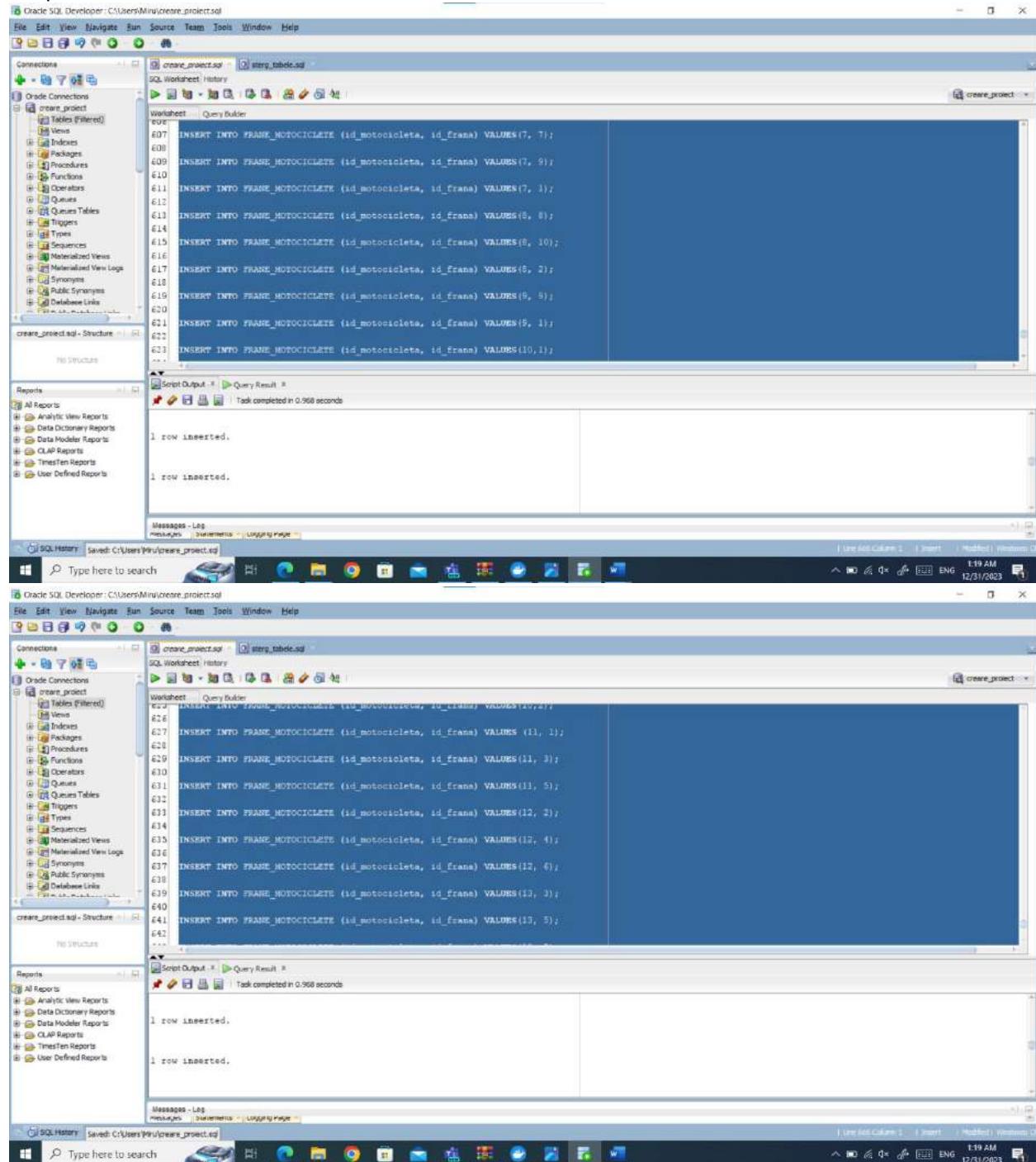
The "Query Result" tab shows the output: "1 row inserted." and "1 row inserted.".

The right window shows the same script and output. Both windows have a status bar at the bottom indicating "1 row inserted." and "1 row inserted.".

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242



The image shows two side-by-side instances of Oracle SQL Developer. Both windows have the title bar "Oracle SQL Developer : C:\Users\Miru\create\_project.sql". The left window's status bar shows "Saved: C:\Users\Miru\create\_project.sql" and the right window's status bar shows "Saved: C:\Users\Miru\create\_project.ed". Both windows display the same SQL script in the Worksheet pane:

```
607 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frama) VALUES(7, 7);
608
609 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frama) VALUES(7, 9);
610
611 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frama) VALUES(7, 1);
612
613 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frama) VALUES(8, 8);
614
615 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frama) VALUES(8, 10);
616
617 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frama) VALUES(8, 2);
618
619 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frama) VALUES(8, 9);
620
621 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frama) VALUES(9, 11);
622
623 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frama) VALUES(10,11);
...
642
```

The Script Output pane shows the results of the execution:

```
1 row inserted.
1 row inserted.
```

The bottom status bars of both windows indicate the current time as 11:19 AM and the date as 12/31/2023.

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays a tree view of database objects under the connection 'create\_project'. The central workspace contains a 'Query Builder' window with the following SQL script:

```
645 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frame) VALUES(14, 4);
646 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frame) VALUES(14, 6);
648 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frame) VALUES(14, 8);
650
651 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frame) VALUES(15, 5);
652
653 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frame) VALUES(15, 7);
654
655 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frame) VALUES(15, 9);
656
657 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frame) VALUES(10, 6);
658
659 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frame) VALUES(11, 8);
660
661 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frame) VALUES(12, 10);
662
663 SELECT *
664 FROM FRAME_MOTOCICLETE
665 ORDER BY ID_MOTOCICLETA;
```

The 'Script Output' tab shows the results of the execution:

```
1 row inserted.
```

The status bar at the bottom right indicates the task completed in 0.968 seconds.

This screenshot shows a second instance of the Oracle SQL Developer interface with the same setup. The 'Query Builder' window contains the same SQL script:

```
645 INSERT INTO FRAME_MOTOCICLETE (id_motocicleta, id_frame) VALUES(14, 4);
```

The 'Script Output' tab shows the results of the execution:

```
1 row inserted.
```

The status bar at the bottom right indicates the task completed in 0.968 seconds.

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

Oracle SQL Developer : C:\Users\Miru\create\_project.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections create\_project Tables (Filtered) Views Indexes Packages Procedures Functions Operators Queues Triggers Types Sequences Materialized Views Materialized View Logs Synonyms Public Synonyms Database Links

create\_project.sql - Structure PLS STRUCTURE Reports All Reports Analytic View Reports Data Dictionary Reports Data Modeler Reports CLAP Reports TimesTen Reports User Defined Reports

create\_project.sql : sterg\_table.sql

SQL Worksheet History

Worksheet Query Builder

Script Output Query Result All Rows Fetched: 46 in 0.018 seconds

ID_MOTOCICleta	ID_Franka
1	1
2	1
3	1
4	2
5	2
6	2
7	3
8	3
9	3
10	4
11	4
12	4
13	5
14	5
15	5
16	6
17	6
18	6
19	7
20	7
21	7
22	8
23	8
24	8

Messages - Log

Type here to search

12:00 AM 12/31/2023

Oracle SQL Developer : C:\Users\Miru\create\_project.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections create\_project Tables (Filtered) Views Indexes Packages Procedures Functions Operators Queues Triggers Types Sequences Materialized Views Materialized View Logs Synonyms Public Synonyms Database Links

create\_project.sql - Structure PLS STRUCTURE Reports All Reports Analytic View Reports Data Dictionary Reports Data Modeler Reports CLAP Reports TimesTen Reports User Defined Reports

create\_project.sql : sterg\_table.sql

SQL Worksheet History

Worksheet Query Builder

Script Output Query Result All Rows Fetched: 46 in 0.018 seconds

ID_MOTOCICleta	ID_Franka
23	8
24	8
25	9
26	9
27	10
28	10
29	10
30	11
31	11
32	11
33	11
34	12
35	12
36	12
37	12
38	13
39	13
40	13
41	14
42	14
43	14
44	15
45	15
46	15

Messages - Log

Type here to search

12:00 AM 12/31/2023

**6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții studiate. Apelați subprogramul.**

---

➤ *Cerință în limbaj natural*

Pentru toate motocicletele care au kilometrajul mai mare decât o valoare dată (colecție care să rețină id-urile acestor motociclete) și sunt deținute de utilizatorii care au un număr maxim de motociclete (colecție care să rețină id-ul acestor utilizatori), să se calculeze numărul de accesorii necesare (atașate) în medie la o astfel de motocicletă (colecție care să rețină pentru fiecare motocicletă numărul de accesorii ale ei). Pentru o veridicitate suplimentară a informațiilor obținute, pe parcursul problemei se va afișa numărul maxim de motociclete deținute de un utilizator, lista utilizatorilor care dețin aceste motociclete și lista motocicletelor acestora. La final, să se afișeze lista motocicletelor cu restricția precizată (care să aibă kilometrajul mai mare decât o valoare dată) alături de numărul de accesorii atașate și o valoare care să reprezinte numărul mediu de accesorii pe care le deține o astfel de motocicletă.

Pentru a evidenția mai bine rezultatele obținute la această cerință, voi adăuga în tabelul MOTOCICLETE încă 5 înregistrări, iar în tabelul ACCESORII\_MOTOCICLETE voi adăuga încă 25 de înregistrări :

➤ *Cod SQL Înregistrări Adăugate*

```
--Înregistrari noi pentru MOTOCICLETE
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatriculare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (9, 1, 3, 5, 12000, 'B-123-XYZ', 'Honda', 'CBR600RR', 'Excelenta', 2019, 18);
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (9, 2, 6, 8, 8000, 'C-456-UVW', 'Yamaha', 'YZF-R1', 'Buna', 2020, 17);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (9, 3, 9, 2, 6000, 'D-789-ABC', 'Kawasaki', 'Ninja 650', 'Foarte Buna', 2021, 15);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (9, 4, 1, 7, 10000, 'E-012-XYZ', 'Suzuki', 'GSX-R1000', 'Buna', 2018, 19);
```

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj,  
nr_inmatricularare, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (3, 4, 2, 7, 10000, 'E-012-XYZ', 'Suzuki', 'GSX-R1000', 'Buna', 2018, 19);
```

--Inregistrari noi pentru ACCESORII\_MOTOCICLETE

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(8, 1);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(8, 2);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(9, 3);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(10, 4);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(10, 5);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(11, 6);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(12, 7);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(12, 8);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(12, 9);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(12, 10);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(14, 1);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(15, 2);
```

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(15, 3);
```

## Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(41, 4);
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(43, 5);
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(43, 6);
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(43, 7);
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(44, 8);
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(44, 9);
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(6, 1);
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(6, 2);
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(6, 3);
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(6, 4);
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(7, 5);
INSERT INTO ACCESORII_MOTOCICLETE (id_motocicleta, id_accesoriu) VALUES(21, 5);
```

### ➤ Print-Screen Înregistrări Adăugate

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

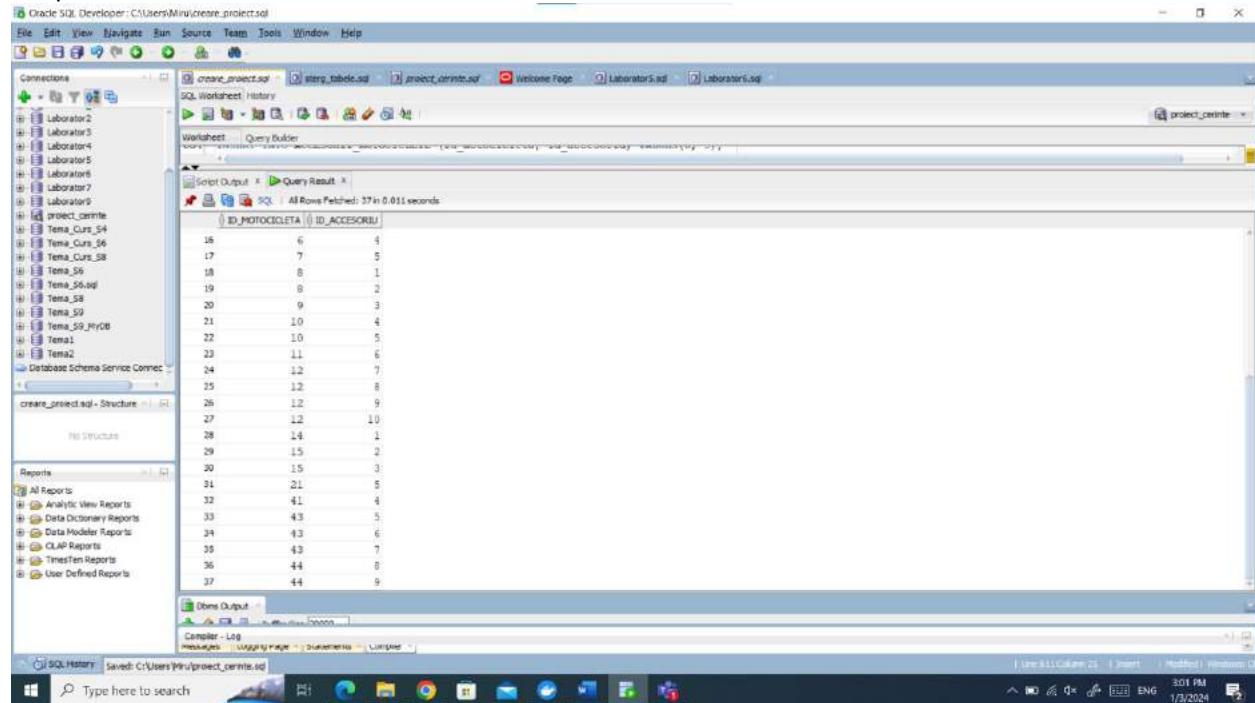
- Connections:** Shows multiple connections including "Laborator2", "Laborator3", "Laborator4", "Laborator5", "Laborator6", "Laborator7", "Laborator9", "project\_center", "Tema\_Cura\_54", "Tema\_Cura\_56", "Tema\_Cura\_58", "Tema\_56", "Tema\_58", "Tema\_58", "Tema\_59", "Tema\_59", "Tema\_SR\_MyOB", "Tema1", "Tema2".
- Script Output:** Displays the SQL code used to insert data into the "MOTOCICLETE" table. The code includes 13 INSERT statements with various values for columns like ID\_UTILIZATOR, ID\_CATEGORIE, ID\_MOTOR, ID\_FAR, KILOMETRAT, NR\_INMATRICULARE, MARCA, MODEL, STARE, and AN.
- Query Result:** Shows the resulting table with 13 rows of motorcycle data. The columns are: ID\_MOTOCICLETA, ID\_UTILIZATOR, ID\_CATEGORIE, ID\_MOTOR, ID\_FAR, CAPACITATE\_RESERVAR, KILOMETRAT, NR\_INMATRICULARE, MARCA, MODEL, STARE, and AN.
- Reports:** Shows a list of reports including All Reports, Analytic View Reports, Data Dictionary Reports, Data Model Reports, CLAP Reports, Timestamp Reports, and User Defined Reports.
- SQL History:** Shows the saved query: C:\Users\Miru\project\_center.sql.

ID_MOTOCICLETA	ID_UTILIZATOR	ID_CATEGORIE	ID_MOTOR	ID_FAR	CAPACITATE_RESERVAR	KILOMETRAT	NR_INMATRICULARE	MARCA	MODEL	STARE	AN
21	21	9	5	8	8	14	15000-B-08-VMX	KTM	RC 390	Buna	2016
22	41	9	1	3	5	18	12000-B-123-XYZ	Honda	CBR600RR	Exceleenta	2019
23	42	9	2	6	8	17	6000-C-456-UVW	Yamaha	YZF-R1	Buna	2020
24	43	9	3	9	2	15	6000-D-789-ABC	Kawasaki	Ninja 650	Poarte Buna	2021
25	44	9	4	1	7	19	10000-E-012-XYZ	Suzuki	GSX-R1000	Buna	2018
26	61	3	4	2	7	19	10000-E-012-XYZ	Suzuki	GSX-R1000	Buna	2018

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242



## ➤ Cod PL/SQL

CREATE OR REPLACE PROCEDURE cerinta6

```
(v_kilometraj motociclete.kilometraj%TYPE DEFAULT 7000)
```

IS

```
TYPE tablou_indexat IS TABLE OF NUMBER INDEX BY PLS_INTEGER;  
t_indexat tablou_indexat;
```

```
TYPE tablou_imbricat IS TABLE OF NUMBER;  
t_imbricat tablou_imbricat := tablou_imbricat();
```

```
TYPE vector IS VARRAY(100) OF NUMBER;  
v vector:= vector();
```

```
nr_max INT;  
v_nume UTILIZATORI.NUME%TYPE;
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
nr_accesorii INT;
nr_mediul_accesorii INT;
nr_total_accesorii INT;

BEGIN

-- aflam numarul maxim de motociclete detinute de catre un utilizator
SELECT MAX(COUNT(M.ID_MOTOCICLETA))
INTO nr_max
FROM MOTOCICLETE M
GROUP BY M.ID_UTILIZATOR;

DBMS_OUTPUT.PUT_LINE('Numarul maxim de motociclete detinute de un utilizator: '||nr_max);
DBMS_OUTPUT.PUT_LINE('.....');

DBMS_OUTPUT.PUT_LINE('Utilizatorii care detin numarul maxim de motociclete:');
DBMS_OUTPUT.PUT_LINE('.....');

-- parcurgem toti utilizatorii care detin nr_max motociclete
-- si stocam in t_indexat id-ul utilizatorilor cu numar maxim de motociclete detinute
FOR rec_id_utilizator IN ( SELECT M.ID_UTILIZATOR
                            FROM MOTOCICLETE M
                            GROUP BY M.ID_UTILIZATOR
                            HAVING COUNT(M.ID_MOTOCICLETA) = nr_max) LOOP
    t_indexat(rec_id_utilizator.ID_UTILIZATOR) := 1;
    SELECT
        (U.NUME||' '||U.PRENUME) INTO v_nume
    FROM UTILIZATORI U
    WHERE U.ID_UTILIZATOR=rec_id_utilizator.ID_UTILIZATOR;
    DBMS_OUTPUT.PUT_LINE(v_nume);
END LOOP;
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
DBMS_OUTPUT.PUT_LINE('Motocicletelor acestor utilizatori:');
DBMS_OUTPUT.PUT_LINE('.....');
DBMS_OUTPUT.PUT_LINE('.....');

--parcugem utilizatorii detin un numar maxim de motociclete
FOR i IN t_indexat.FIRST .. t_indexat.LAST
LOOP
  IF t_indexat.EXISTS(i) THEN --conditie de existenta
    DBMS_OUTPUT.PUT_LINE('Id-urile motocicletelor utilizatorului cu id-ul'||i);
    DBMS_OUTPUT.PUT_LINE('.....');
    --parcugem motocicletele detinute de utilizatorul cu id=i
    FOR motocicleta_rec IN ( SELECT M.ID_MOTOCICLETA, M.KILOMETRAJ
      FROM MOTOCICLETE M
      WHERE M.ID_UTILIZATOR = i) LOOP
      --daca motocicleta are kilometrajul mai mare decat valoarea data ca parametru
      --o adaugam in t_imbricat
      IF motocicleta_rec.KILOMETRAJ>v_kilometraj THEN
        t_imbricat.EXTEND;
        t_imbricat(t_imbricat.count) := motocicleta_rec.ID_MOTOCICLETA;
      END IF;
      DBMS_OUTPUT.PUT_LINE(motocicleta_rec.ID_MOTOCICLETA);
    END LOOP;
  END IF;
END LOOP;

DBMS_OUTPUT.PUT_LINE('Numarul de accesoriile pentru fiecare motocicleta care are kilometrajul
mai mare decat valoarea data');
DBMS_OUTPUT.PUT_LINE('.....');
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

--parcurgem motocicletele detinute de utilizatorii care au un numar maxim de motociclete

FOR i IN 1..(t\_imbricat.COUNT-1) LOOP

--retinem in nr\_accesorii numarul de accesorii atasate la o astfel de motocicleta

SELECT COUNT(AM.ID\_ACSESORIU) INTO nr\_accesorii

FROM ACCESORII\_MOTOCICLETE AM

JOIN MOTOCICLETE M ON M.ID\_MOTOCICLETA = AM.ID\_MOTOCICLETA

WHERE AM.ID\_MOTOCICLETA = t\_imbricat(i);

IF t\_imbricat.EXISTS(i) THEN --conditie de existenta

v.EXTEND;

v(i):=nr\_accesorii;

DBMS\_OUTPUT.PUT\_LINE('Motocicleta cu id'||t\_imbricat(i)||' are'||v(i)||' accesorii');

END IF;

END LOOP;

--parcurgem vectorul si aflam numarul total de accesorii atasate la motocicletele

--care sunt detinute de catre utilizatorii care au un numar maxim de motociclete

nr\_total\_accesorii := 0;

FOR i IN 1..v.COUNT LOOP

nr\_total\_accesorii := nr\_total\_accesorii + v(i);

END LOOP;

--facem media accesoriilor detinute de o motocicleta cu conditiile specificate

nr\_accesorii:=v.COUNT;

IF nr\_accesorii> 0 THEN --tratam cazul de impartire la 0

nr\_mediul\_accesorii := nr\_total\_accesorii/nr\_accesorii;

ELSE

nr\_mediul\_accesorii := 0;

END IF;

DBMS\_OUTPUT.PUT\_LINE('Numarul mediu de accesorii este'||nr\_mediul\_accesorii);

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

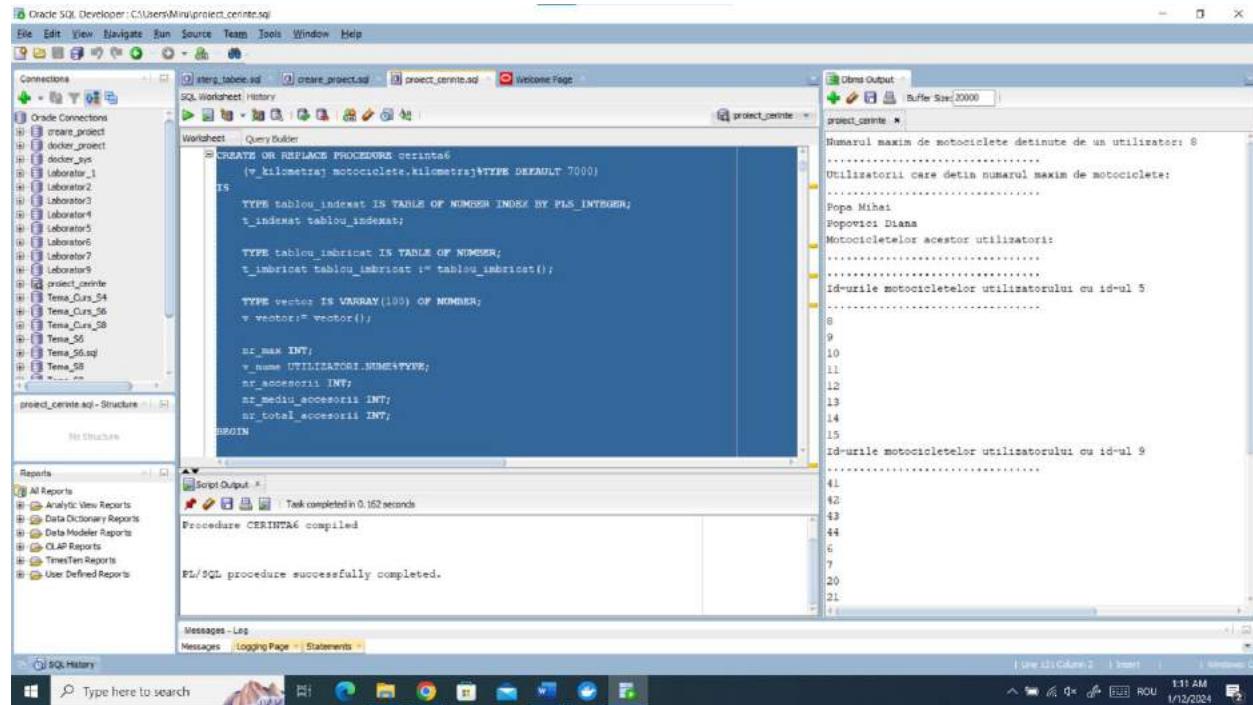
## EXCEPTION

```
WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20000, 'Nu exista motociclete cu specificatiile cerute');

WHEN OTHERS THEN
    RAISE_APPLICATION_ERROR(-20002, 'Alta eroare!');

END cerinta6;
/
BEGIN
cerinta6(12000);
END;
/
```

### ➤ Print-Screen



# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
--afisam numarul maxim de motociclete detinute de catre un utilizator
SELECT MAX(COUNT(M.ID_MOTOCICLETA))
INTO nr_max
FROM MOTOCICLETE M
GROUP BY M.ID_UTILIZATOR;

DBMS_OUTPUT.PUT_LINE('Numarul maxim de motociclete detinute de un utilizator: '||nr_max);
DBMS_OUTPUT.PUT_LINE('.....');

DBMS_OUTPUT.PUT_LINE('Utilizatorii care detin numarul maxim de motociclete: ');
DBMS_OUTPUT.PUT_LINE('.....');

--parcurgem toti utilizatorii care detin nr_max motociclete
--si stocam in t_indexat id-ul utilizatorilor cu numar maxim de motociclete detinute
FOR rec_id_utilizator IN ( SELECT M.ID_UTILIZATOR
                           FROM MOTOCICLETE M
                           GROUP BY M.ID_UTILIZATOR
                           HAVING COUNT(M.ID_MOTOCICLETA) = nr_max) LOOP
    t_indexat(rec_id_utilizator.ID_UTILIZATOR) := i;
    SELECT
        (U.NUME||' ('||U.PRENUME)||' ') INTO v_name
    FROM UTILIZATORI U
    WHERE U.ID_UTILIZATOR=rec_id_utilizator.ID_UTILIZATOR;
    DBMS_OUTPUT.PUT_LINE(v_name);
END LOOP;

Numarul de accesorii pentru fiecare motocicleta care are kilometri:
.....
Motocicleta cu id 9 are 1 accesoriu
Motocicleta cu id 10 are 2 accesorii
Motocicleta cu id 11 are 1 accesoriu
Motocicleta cu id 12 are 4 accesorii
Motocicleta cu id 13 are 0 accesorii
Motocicleta cu id 14 are 1 accesoriu
Motocicleta cu id 15 are 2 accesorii
Motocicleta cu id 7 are 1 accesoriu
Numarul mediu de accesorii este 2
```

```
END LOOP;

DBMS_OUTPUT.PUT_LINE('Motocicletele acestor utilizatori:');
DBMS_OUTPUT.PUT_LINE('.....');
DBMS_OUTPUT.PUT_LINE('.....');

--parcurgem utilizatorii care detin un numar maxim de motociclete
FOR i IN t_indexat.FIRST .. t_indexat.LAST
LOOP
    IF t_indexat(i).EXISTS() THEN --conditie de existenta
        DBMS_OUTPUT.PUT_LINE('Id-urile motocicletelor utilizatorului cu id-ul '||i);
        DBMS_OUTPUT.PUT_LINE('.....');
        --parcurgem motocicletele detinute de utilizatorul cu id'i
        FOR motocicleta_rec IN ( SELECT M.ID_MOTOCICLETA, M.KILOMETRAG
                                   FROM MOTOCICLETE M
                                   WHERE M.ID_UTILIZATOR = i) LOOP
            --daca motociclete are kilometrajul mai mare decat valoarea data ca parametru
            --o adaugam in t_imbricat
            IF motocicleta_rec.KILOMETRAG > kilometraj THEN
                t_imbricat(t_imbricat.count) := motocicleta_rec.ID_MOTOCICLETA;
            END IF;
        DBMS_OUTPUT.PUT_LINE(motocicleta_rec.ID_MOTOCICLETA);
    END LOOP;
END LOOP;

Numarul de accesorii pentru fiecare motocicleta care are kilometri:
.....
Motocicleta cu id 9 are 1 accesoriu
Motocicleta cu id 10 are 2 accesorii
Motocicleta cu id 11 are 1 accesoriu
Motocicleta cu id 12 are 4 accesorii
Motocicleta cu id 13 are 0 accesorii
Motocicleta cu id 14 are 1 accesoriu
Motocicleta cu id 15 are 2 accesorii
Motocicleta cu id 7 are 1 accesoriu
Numarul mediu de accesorii este 2
```

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface with the 'project\_cerinte' procedure open in the workspace. The code implements a solution for calculating the average number of accessories per motorcycle for a specific user. It uses a cursor to iterate through the accessories attached to each motorcycle and calculates the total count and average for that user.

```
END IF;
DBMS_OUTPUT.PUT_LINE(motocicleta_rec.ID_MOTOCICLETA);
END LOOP;
END IF;
END LOOP;

DBMS_OUTPUT.PUT_LINE('Numarul de accesoriis pentru fiecare motocicleta care are kilometrajul mai mare decat valoarea data');
DBMS_OUTPUT.PUT_LINE('.....');

--parcurgem motoricilele detinute de utilizatorii care au un numar maxim de motociclete
FOR i IN 1..t_imbricat.COUNT LOOP
--scriem in nr_accesoriis numarul de accesoriis atașate la o astfel de motocicleta
SELECT COUNT(M.ID_ACSESORIU) INTO nr_accesoriis
FROM ACCESORII_MOTOCICLETE AM
JOIN MOTOCICLETE M ON M.ID_MOTOCICLETA = AM.ID_MOTOCICLETA
WHERE AM.ID_MOTOCICLETA = t_imbricat(i);
IF t_imbricat.EXISTS(i) THEN --conditie de existenta
    v_EXTENO:=nr_accesoriis;
    DBMS_OUTPUT.PUT_LINE('Motocicleta cu id '||i||' are '||v_EXTENO||' accesoriis');
END IF;
END LOOP;

--parcurgem vectorul si aflam numarul total de accesoriis atașate la motocicletele
```

Script Output: Task completed in 0.162 seconds  
Procedure CERINTA6 compiled  
Messages - Log  
Messages - Logging Page  
Statements

The screenshot shows the Oracle SQL Developer interface with the 'project\_cerinte' procedure open in the workspace. This version of the code is more concise, using a cursor to calculate the total number of accessories and then dividing by the count of motorcycles to find the average. It also includes error handling for cases where no motorcycle exists for a given user ID.

```
--parcurgem vectorul si aflam numarul total de accesoriis atașate la motocicletele
--care sunt detinute de acest utilizator care are un numar maxim de motociclete
nr_total_accesoriis := 0;
FOR i IN 1..v_extenso.COUNT LOOP
    nr_total_accesoriis := nr_total_accesoriis + v(i);
END LOOP;

--facem media accesoriilor detinute de o motocicleta cu conditiile specificate
nr_accesoriis:=v_extenso.COUNT;
IF nr_accesoriis> 0 THEN --tratam cazul de impartire la 0
    nr_mediu_accesoriis:= nr_total_accesoriis/nr_accesoriis;
ELSE
    nr_mediu_accesoriis:= 0;
END IF;

DBMS_OUTPUT.PUT_LINE('Numarul mediu de accesoriis este '|nr_mediu_accesoriis|);

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20000, 'Nu exista motociclete cu specificatiile cerute');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20002,'Alte eroare!');
END cerinta6;
```

Script Output: Task completed in 0.162 seconds  
Procedure CERINTA6 compiled  
Messages - Log  
Messages - Logging Page  
Statements

## Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
project_cerinte.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections project_cerinte.ad
SQL Worksheet History project_cerinte
Worksheet Query Builder
procedure CERINTA6(kilometraj NUMBER)
begin
    declare
        nr_accesorii := 0;
        nr_total_accesorii := 0;
        nr_mediu_accesorii := 0;
    begin
        if nr_accesorii > 0 then
            nr_mediu_accesorii := nr_total_accesorii / nr_accesorii;
        else
            nr_mediu_accesorii := 0;
        end if;

        dbms_output.put_line('Numarul mediu de accesoriile este: ' || nr_mediu_accesorii);
    exception
        when no_data_found then
            raise_application_error(-20000, 'Nu exista date');
        when others then
            raise_application_error(-20002, 'Alte erori');
    end;
end;
/
begin
    cerinta6(12000);
end;
```

Script Output

Procedure CERINTA6 compiled.

PL/SQL procedure successfully completed.

Messages - Log Messages - Logging Page Statements

Output:

```
10
11
12
13
14
15
Id-urile motocicletelor utilizatorului cu id-ul 9
-----
16
17
18
19
20
21
Numarul de accesoriile pentru fiecare motocicleta care are kilometrajul mai mare decat valoarea data
-----
Motocicleta cu id 9 are 1 accesoriu
Motocicleta cu id 10 are 2 accesoriu
Motocicleta cu id 11 are 1 accesoriu
Motocicleta cu id 12 are 4 accesoriu
Motocicleta cu id 13 are 0 accesoriu
Motocicleta cu id 14 are 1 accesoriu
Motocicleta cu id 15 are 2 accesoriu
Motocicleta cu id 7 are 1 accesoriu
Numarul mediu de accesoriile este 2
```

---

**7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat, dependent de celălalt cursor. Apelați subprogramul.**

---

➤ *Cerința în limbaj natural*

Pentru fiecare categorie care cuprinde un număr de motociclete mai mare decât o valoare specificată, să se afișeze numele acesteia, numărul de motociclete care fac parte din ea și lista motocicletelor care fac parte din această categorie.

➤ *Cod PL/SQL*

```
CREATE OR REPLACE PROCEDURE cerinta7
(v_nrmin_moto INT DEFAULT 0)
IS
--cursor parametrizat care afiseaza informatii despre motocicletele care fac parte dintr-o categorie data
--(numar de inmatriculara, numele proprietarului, marca si modelul)

CURSOR c_moto (v_id_categorie NUMBER) IS
    SELECT M.NR_INMATRICULARA numar_inmatriculara, (U.NUME||' '||U.PRENUME)
    nume_proprietar, (M.MARCA||' '||M.MODEL) informatii
    FROM MOTOCICLETE M
    JOIN UTILIZATORI U ON U.ID_UTILIZATOR=M.ID_UTILIZATOR
    WHERE M.ID_CATEGORIE = v_id_categorie;
--tratarea cazului in care nu exista categorii care sa contin mai mult de v_nrmin_moto motociclete
verifica INT:=0;
BEGIN
--parcurgem categoriile care contin un numar de motociclete mai mare decat valoarea v_nrmin_moto
data
FOR rec_categ IN (
    SELECT
        C.ID_CATEGORIE ID, C.NUME NUME, COUNT(M.ID_MOTOCICLETA)
    NR_MOTOCICLETE
    FROM MOTOCICLETE M
    RIGHT JOIN CATEGORII C ON M.ID_CATEGORIE=C.ID_CATEGORIE
    GROUP BY C.ID_CATEGORIE, C.NUME
    HAVING COUNT(M.ID_MOTOCICLETA)>v_nrmin_moto) LOOP
DBMS_OUTPUT.PUT_LINE('.....');
--afisare adaptata la numarul de motociclete
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
IF rec_categ.nr_motociclete=0 THEN
    DBMS_OUTPUT.PUT_LINE('Categoria'||rec_categ.nume|| nu contine nicio motocicleta');
ELSIF rec_categ.nr_motociclete=1 THEN
    DBMS_OUTPUT.PUT_LINE('Categoria'||rec_categ.nume|| contine o singura motocicleta');
ELSE
    DBMS_OUTPUT.PUT_LINE('Categoria'||rec_categ.nume|| contine'||rec_categ.nr_motociclete|| motociclete');
END IF;
DBMS_OUTPUT.PUT_LINE('.....');

FOR rec_moto IN c_moto(rec_categ.id) LOOP
    DBMS_OUTPUT.PUT_LINE('Numar de inmatriculare: '|rec_moto.numar_inmatriculare||' Proprietar: '|rec_moto.nume_proprietar|| Informatii: '|rec_moto.informatii);
    END LOOP;
    verifica:=1;
    END LOOP;

IF verifica=0 THEN
    DBMS_OUTPUT.PUT_LINE('Nu exista categorii care sa contina mai mult de'||v_nrmin_moto|| motociclete');
END IF;
END cerinta7;
/
BEGIN
cerinta7(2);
END;
/
```

➤ Print-Screen

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

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

- Connections:** Oracle Connection, project\_centele.
- SQL Worksheet:** History tab, showing the creation of the procedure `cerinta7`.
- Code:**

```
CREATE OR REPLACE PROCEDURE cerinta7
    (v_nrmax_moto INT DEFAULT 0)
IS
    --cursor parametrizat care afiseaza informatii despre motocicletele care fac parte dintr-o categorie data
    --(numar de inmatriculate, numele proprietarului, marca si modelul)

    CURSOR c_moto (v_id_categorie NUMBER) IS
        SELECT M.ID_INMATRICULARE numar_inmatriculare, (U.NUME)||' ('||U.PRENUME) nume_proprietar, (M.NUMAR)||'-'||M
        FROM MOTOCICLETE M
        JOIN UTILIZATORI U ON U.ID_UTILIZATOR=M.ID_UTILIZATOR
        WHERE M.ID_CATEGORIE = v_id_categorie;
    --tratarea cazului in care nu exista categorii care sa contin mai mult de v_nrmax_moto motociclete
    --verifica INT>0;
BEGIN
    --parcurgem categoriile care contin un numar de motociclete mai mare decat valoarea v_nrmax_moto data
    FOR rec_cat IN (
        SELECT
            C.ID_CATEGORIE ID, C.NUME NUME, COUNT(M.ID_MOTOCICLETA) NR_MOTOCICLETE
        FROM MOTOCICLETE M
        RIGHT JOIN CATEGORII C ON M.ID_CATEGORIE=C.ID_CATEGORIE
    )
    LOOP
        IF rec_cat.NR_MOTOCICLETE > v_nrmax_moto THEN
            DBMS_OUTPUT.PUT_LINE('Categorie '||rec_cat.NUME||' contine '||rec_cat.NR_MOTOCICLETE||' motociclete');
        END IF;
    END LOOP;
END;
```
- Output:** Shows the execution results:
  - Categorie Cruiser contine 3 motociclete
  - Numar de inmatriculare: B-123-KVZ Proprietar
  - Numar de inmatriculare: B-02-DEF Proprietar
  - Numar de inmatriculare: B-01-ABC Proprietar
  - Categorie Touring contine 5 motociclete
  - Numar de inmatriculare: E-012-KVZ Proprietar
  - Numar de inmatriculare: E-012-KVZ Proprietar
  - Numar de inmatriculare: B-08-VKX Proprietar
  - Numar de inmatriculare: B-07-STU Proprietar
  - Numar de inmatriculare: E-012-KVZ Proprietar
  - Categorie Sport contine 3 motociclete
  - Numar de inmatriculare: B-03-GHI Proprietar
  - Numar de inmatriculare: B-04-JKL Proprietar
  - Numar de inmatriculare: C-456-UWV Proprietar
  - Categorie Naked contine 14 motociclete
  - Numar de inmatriculare: B-03-GHI Proprietar
  - Numar de inmatriculare: B-04-JKL Proprietar
  - Numar de inmatriculare: B-10-234 Proprietar
- Messages - Log:** PL/SQL procedure successfully completed.
- Messages - Logging Page:** Statements

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

- Connections:** Oracle Connection, project\_cerinte.ad
- SQL Worksheet:** History tab, project\_cerinte.ad
- Script Output:** Task completed in 0.115 seconds
- Procedure:** project\_cerinte.ad
- Output:** The procedure outputs the following information:
  - Categoria Cruiser contine 3 motociclete
  - Numar de immatriculare: B-123-KVZ Proprietar
  - Numar de immatriculare: B-02-DEP Proprietar
  - Numar de immatriculare: B-01-ABC Proprietar
  - Categoria Touring contine 5 motociclete
  - Numar de immatriculare: E-012-XV1 Proprietar
  - Numar de immatriculare: E-012-XV1 Proprietar
  - Numar de immatriculare: B-08-VWV Proprietar
  - Numar de immatriculare: E-07-STV Proprietar
  - Numar de immatriculare: E-012-XV1 Proprietar
  - Categoria Sport contine 3 motociclete
  - Numar de immatriculare: B-03-GHI Proprietar
  - Numar de immatriculare: B-04-JKL Proprietar
  - Numar de immatriculare: C-456-UWV Proprietar
  - Categoria Naked contine 14 motociclete
  - Numar de immatriculare: B-03-GHI Proprietar
  - Numar de immatriculare: B-04-JKL Proprietar
  - Numar de immatriculare: B-10-234 Proprietar
  - Numar de immatriculare: B-10-234 Proprietar
  - Numar de immatriculare: B-10-234 Proprietar

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, the path is set to 'File > New > SQL Script'. The 'Connections' pane on the left lists several projects and connections, with 'project\_center' selected. The 'SQL Worksheet' tab contains the following PL/SQL code:

```
ELSE
    DBMS_OUTPUT.PUT_LINE('Categoria "'||rec_categ);
END IF;
DBMS_OUTPUT.PUT_LINE('.....');

FOR rec_moto IN c_moto(rec_categ.id) LOOP
    DBMS_OUTPUT.PUT_LINE('Numar de inmatriculare:');
    ENO LOOP;
    verifica:=1;
    ENO LOOP;

    IF verifica=0 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista categorii care sa');
    END IF;
    ENO cerintas7;

BEGIN
    cerinta7(2);
END;
```

The 'Dbsn Output' pane on the right displays the results of the procedure execution, which lists motorcycle categories and their details:

- Categoria Cruiser contine 3 motociclete
  - Numar de inmatriculare: B-123-XYZ Proprietar: Popovici Diana Informatii: Honda CB600RR
  - Numar de inmatriculare: B-02-DEF Proprietar: Muresan Radu Informatii: Yamaha RX
  - Numar de inmatriculare: B-01-ABC Proprietar: Muresan Radu Informatii: Honda CB600RR
- Categoria Touring contine 5 motociclete
  - Numar de inmatriculare: E-012-XYZ Proprietar: Ionescu Maria Informatii: Suzuki GSX-R1000
  - Numar de inmatriculare: E-012-XYZ Proprietar: Georgescu Alexandru Informatii: Suzuki GSX-R1000
  - Numar de inmatriculare: B-06-VWX Proprietar: Popovici Diana Informatii: BMW S1000RR
  - Numar de inmatriculare: B-07-STV Proprietar: Popovici Diana Informatii: BMW S1000RR
  - Numar de inmatriculare: E-012-XYZ Proprietar: Popovici Diana Informatii: Suzuki GSX-R1000
- Categoria Sport contine 3 motociclete
  - Numar de inmatriculare: B-03-GHI Proprietar: Ionescu Maria Informatii: Kawasaki Ninja ZX-6R
  - Numar de inmatriculare: B-04-JKL Proprietar: Ionescu Maria Informatii: Suzuki GSX-R750
  - Numar de inmatriculare: C-456-UVM Proprietar: Popovici Diana Informatii: Yamaha YZF-R1
- Categoria Naked contine 14 motociclete
  - Numar de inmatriculare: B-03-GHI Proprietar: Ionescu Maria Informatii: Kawasaki Ninja ZX-6R
  - Numar de inmatriculare: B-04-JKL Proprietar: Ionescu Maria Informatii: Suzuki GSX-R750
  - Numar de inmatriculare: B-10-234 Proprietar: Popa Mihai Informatii: Indian Chieftain Dark Horse
  - Numar de inmatriculare: B-10-234 Proprietar: Popa Mihai Informatii: Indian Chieftain Dark Horse
  - Numar de inmatriculare: B-10-234 Proprietar: Popa Mihai Informatii: Indian Chieftain Dark Horse
  - Numar de inmatriculare: B-10-234 Proprietar: Popa Mihai Informatii: Indian Chieftain Dark Horse
  - Numar de inmatriculare: B-03-GHI Proprietar: Ionescu Maria Informatii: Kawasaki Ninja ZX-6R
  - Numar de inmatriculare: B-04-JKL Proprietar: Ionescu Maria Informatii: Suzuki GSX-R750
  - Numar de inmatriculare: C-456-UVM Proprietar: Popovici Diana Informatii: Yamaha YZF-R1
  - Numar de inmatriculare: B-03-GHI Proprietar: Ionescu Maria Informatii: Kawasaki Ninja ZX-6R
  - Numar de inmatriculare: B-04-JKL Proprietar: Ionescu Maria Informatii: Suzuki GSX-R750
  - Numar de inmatriculare: B-10-234 Proprietar: Popa Mihai Informatii: Indian Chieftain Dark Horse
  - Numar de inmatriculare: B-10-234 Proprietar: Popa Mihai Informatii: Indian Chieftain Dark Horse
  - Numar de inmatriculare: B-10-234 Proprietar: Popa Mihai Informatii: Indian Chieftain Dark Horse

This screenshot shows the Oracle SQL Developer interface again, with the 'File > New > SQL Script' path selected. The 'Connections' pane shows 'project\_center' as the active connection. The 'SQL Worksheet' tab contains the same PL/SQL code as the previous screenshot:

```
ELSE
    DBMS_OUTPUT.PUT_LINE('Categoria "'||rec_categ);
END IF;
DBMS_OUTPUT.PUT_LINE('.....');

FOR rec_moto IN c_moto(rec_categ.id) LOOP
    DBMS_OUTPUT.PUT_LINE('Numar de inmatriculare:');
    ENO LOOP;
    verifica:=1;
    ENO LOOP;

    IF verifica=0 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista categorii care sa');
    END IF;
    ENO cerintas7;

BEGIN
    cerinta7(2);
END;
```

The 'Dbsn Output' pane shows the same results as the first screenshot, listing motorcycle categories and their details.

**8. Formulați în limbaj natural o problemă pe care să o rezolvăți folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele definite. Definiți minim 2 excepții proprii.**

**Apelați subprogramul astfel încât să evidențiați toate cazurile definite și tratațe.**

---

➤ *Cerința în limbaj natural*

Pentru utilizatorul a cărui nume de familie este dat, să se afișeze numărul de motociclete pe care le deține care merg pe benzină.

➤ *Cod PL/SQL*

```
CREATE OR REPLACE FUNCTION cerinta8
```

```
    (v_nume utilizatori.nume%TYPE DEFAULT 'Georgescu') RETURN NUMBER
IS
    nr_motociclete NUMBER;
    --excepții proprii
    NO_DATA_FOUND_UTILIZATOR EXCEPTION;
    NO_DATA_FOUND_MOTOCICLETA EXCEPTION;
    contor NUMBER;
BEGIN
    SELECT COUNT(*) INTO contor
    FROM UTILIZATORI
    WHERE UPPER(NUME)=UPPER(v_nume);
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

--exceptie pentru niciun utilizator existent cu numele dat

IF contor=0 THEN

RAISE NO\_DATA\_FOUND\_UTILIZATOR;

--exceptie pentru mai multi utilizatori care indeplinesc conditia ceruta

ELSIF contor>1 THEN

RAISE TOO\_MANY\_ROWS;

END IF;

SELECT COUNT(\*) INTO contor

FROM MOTOCICLETE M

JOIN UTILIZATORI U ON U.ID\_UTILIZATOR=M.ID\_UTILIZATOR

WHERE UPPER(U.NUME)=UPPER(v\_nume);

--exceptie pentru utilizatorii care detin nicio motocicleta in baza de date

IF contor=0 THEN

RAISE NO\_DATA\_FOUND\_MOTOCICLETA;

END IF;

SELECT

COUNT(M.ID\_MOTOCICLETA) INTO nr\_motociclete

--luam si utilizatorii care nu au motociclete care sa consume benzina

FROM UTILIZATORI U

LEFT JOIN MOTOCICLETE M ON U.ID\_UTILIZATOR = M.ID\_UTILIZATOR

--selectam motocicletele care merg pe benzina

WHERE M.ID\_MOTOCICLETA IS NULL OR M.ID\_MOTOCICLETA IN (

SELECT M1.ID\_MOTOCICLETA

FROM MOTOCICLETE M1

JOIN MOTOARE MT1 ON MT1.ID\_MOTOR=M1.ID\_MOTOR

WHERE UPPER(MT1.CARBURANT) = 'BENZINA')

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
    GROUP BY U.ID_UTILIZATOR, U.NUME
    HAVING UPPER(U.NUME)=UPPER(v_num);
    RETURN nr_motociclete;
EXCEPTION
    WHEN NO_DATA_FOUND_UTILIZATOR THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista niciun utilizator in baza de date cu numele dat');
        RETURN -1;
    WHEN NO_DATA_FOUND_MOTOCICLETA THEN
        DBMS_OUTPUT.PUT_LINE('Utilizatorul specificat nu are motociclete salvate in baza de date');
        RETURN -1;
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Utilizatorul specificat nu detine motociclete care sa indeplineasca
conditia');
        RETURN -1;
    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE('Există mai mulți utilizatori cu numele dat');
        RETURN -1;
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Cod: ' || SQLCODE);
        DBMS_OUTPUT.PUT_LINE('Mesaj: ' || SQLERRM);
        RETURN -1;
END cerinta8;
/
BEGIN
    --cazul în care utilizatorul dat detine motociclete care consumă benzina
    DBMS_OUTPUT.PUT_LINE('Pentru Popa: '||cerinta8('Popa'));
    DBMS_OUTPUT.PUT_LINE('.....');
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

--cazul in care nu exista un utilizator cu numele dat in baza de date

```
DBMS_OUTPUT.PUT_LINE('Pentru Vornicu:'||cerinta8('Vornicu'));
```

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

--cazul in care utilizatorul dat nu detine motociclete (in general)

```
DBMS_OUTPUT.PUT_LINE('Pentru Popescu:'||cerinta8('Popescu'));
```

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

--cazul in care utilizatorul dat nu detine motociclete care sa consume benzina

```
DBMS_OUTPUT.PUT_LINE('Pentru Georgescu:'||cerinta8('Georgescu'));
```

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

--cazul in care exista mai multi utilizatori cu numele dat

```
DBMS_OUTPUT.PUT_LINE('Pentru Muresan:'||cerinta8('Muresan'));
```

END;

/

➤ *Print-Screen*

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. In the central workspace, a PL/SQL block is being written:

```
101  CREATE OR REPLACE FUNCTION cerinta$  
102    (v_name utilizator) RETURN NUMBER  
103  IS  
104      nr_motociclete NUMBER;  
105      --exceptie proprietar  
106      NO_DATA_FOUND_UTILIZATOR EXCEPTION;  
107      NO_DATA_FOUND_MOTOCICLETA EXCEPTION;  
108      contor NUMBER;  
109      BEGIN  
110          SELECT COUNT(*) INTO contor  
111          FROM UTILIZATORI  
112          WHERE UPPER(UNAME)=UPPER(v_name);  
113          --exceptie pentru niciun utilizator existent cu numele dat  
114          IF contor=0 THEN  
115              RAISE NO_DATA_FOUND_UTILIZATOR;  
116          --exceptie pentru mai multi utilizatori care indeplinesc conditia ceruta  
117          ELSEIF contor>1 THEN  
118              RAISE TOO_MANY_ROWS;  
119          END IF;  
120          SELECT COUNT(*) INTO contor  
121          FROM MOTOCICLETE M  
122          JOIN UTILIZATORI U ON U.ID_UTILIZATOR=M.ID_UTILIZATOR  
123          WHERE UPPER(U.UNAME)=UPPER(v_name);  
124          --exceptie pentru utilizatorii care detin nicio motocicleta in baza de date  
125          IF contor=0 THEN  
126              RAISE NO_DATA_FOUND_MOTOCICLETA;  
127          END IF;  
128          SELECT  
129              COUNT(M.ID_MOTOCICLETA) INTO nr_motociclete  
130              --luam si utilizatorii care nu au motociclete care sa consume benzina  
131              FROM UTILIZATORI U  
132              LEFT JOIN MOTOCICLETE M ON U.ID_UTILIZATOR = M.ID_UTILIZATOR  
133              --selecatam motocicletele care merg pe benzina  
134              WHERE M.ID_MOTOCICLETA IS NULL OR M.ID_MOTOCICLETA IN  
135                  MOTOCICLETE_BENZINA;
```

The right pane shows the Dbsn Output window with the results of the function execution:

```
Functie Popas: ?  
.....  
Nu exista niciun utilizator in baza de date cu numele dat  
Functie Vornicu: -1  
.....  
Utilizatorul specificat nu are motociclete salvate in baza de date  
Functie Popescu: -1  
.....  
Utilizatorul specificat nu detine motociclete care sa indeplineasca conditia  
Functie Georgescu: -1  
.....  
Exista mai multi utilizatori cu numele dat  
Functie Muresan: -1
```

At the bottom, the status bar indicates the date and time: 1/10/2024 12:13 AM.

The screenshot shows the Oracle SQL Developer interface. In the central workspace, a PL/SQL block is being written:

```
221  BEGINIF contor>1 THEN  
222      RAISE TOO_MANY_ROWS;  
223  END IF;  
224  
225  SELECT COUNT(*) INTO contor  
226  FROM MOTOCICLETE M  
227  JOIN UTILIZATORI U ON U.ID_UTILIZATOR=M.ID_UTILIZATOR  
228  WHERE UPPER(U.UNAME)=UPPER(v_name);  
229  
230  --exceptie pentru utilizatorii care detin nicio motocicleta in baza de date  
231  IF contor=0 THEN  
232      RAISE NO_DATA_FOUND_MOTOCICLETA;  
233  END IF;  
234  
235  SELECT  
236      COUNT(M.ID_MOTOCICLETA) INTO nr_motociclete  
237      --luam si utilizatorii care nu au motociclete care sa consume benzina  
238      FROM UTILIZATORI U  
239      LEFT JOIN MOTOCICLETE M ON U.ID_UTILIZATOR = M.ID_UTILIZATOR  
240      --selecatam motocicletele care merg pe benzina  
241      WHERE M.ID_MOTOCICLETA IS NULL OR M.ID_MOTOCICLETA IN  
242          MOTOCICLETE_BENZINA;
```

The right pane shows the Dbsn Output window with the results of the function execution:

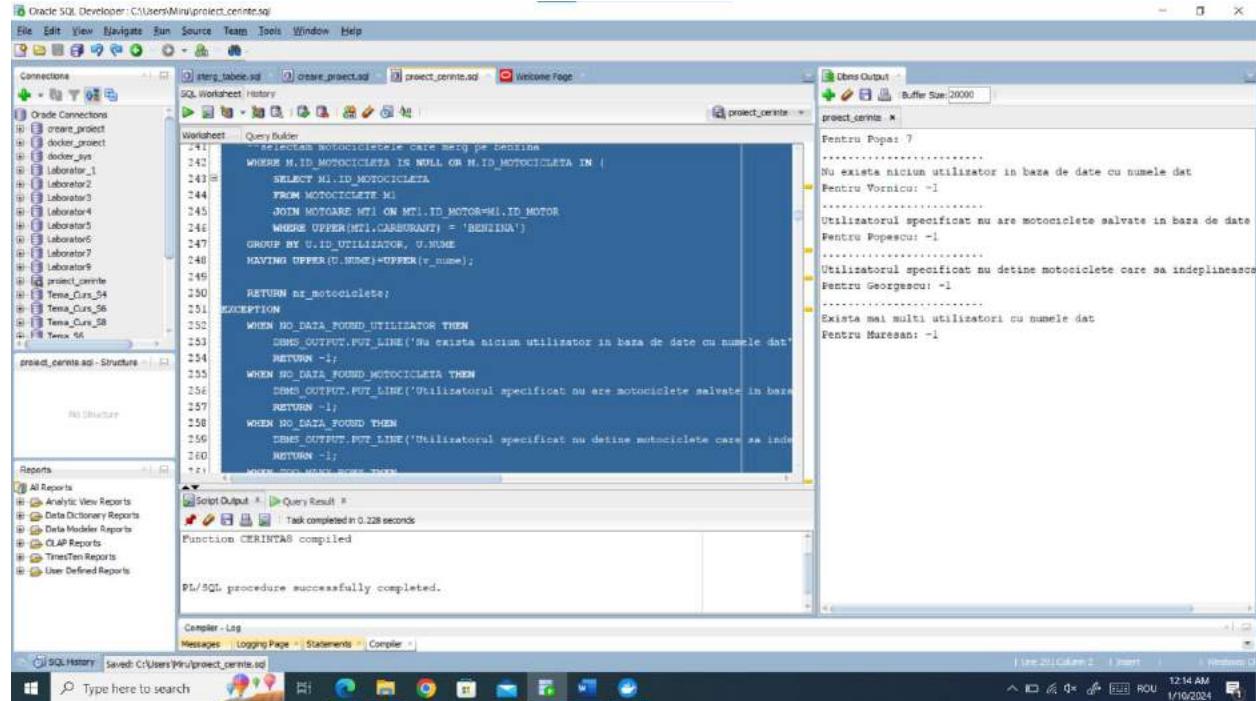
```
Functie Popas: ?  
.....  
Nu exista niciun utilizator in baza de date cu numele dat  
Functie Vornicu: -1  
.....  
Utilizatorul specificat nu are motociclete salvate in baza de date  
Functie Popescu: -1  
.....  
Utilizatorul specificat nu detine motociclete care sa indeplineasca  
Functie Georgescu: -1  
.....  
Exista mai multi utilizatori cu numele dat  
Functie Muresan: -1
```

At the bottom, the status bar indicates the date and time: 1/10/2024 12:13 AM.

# Sisteme de gestiune a bazelor de date

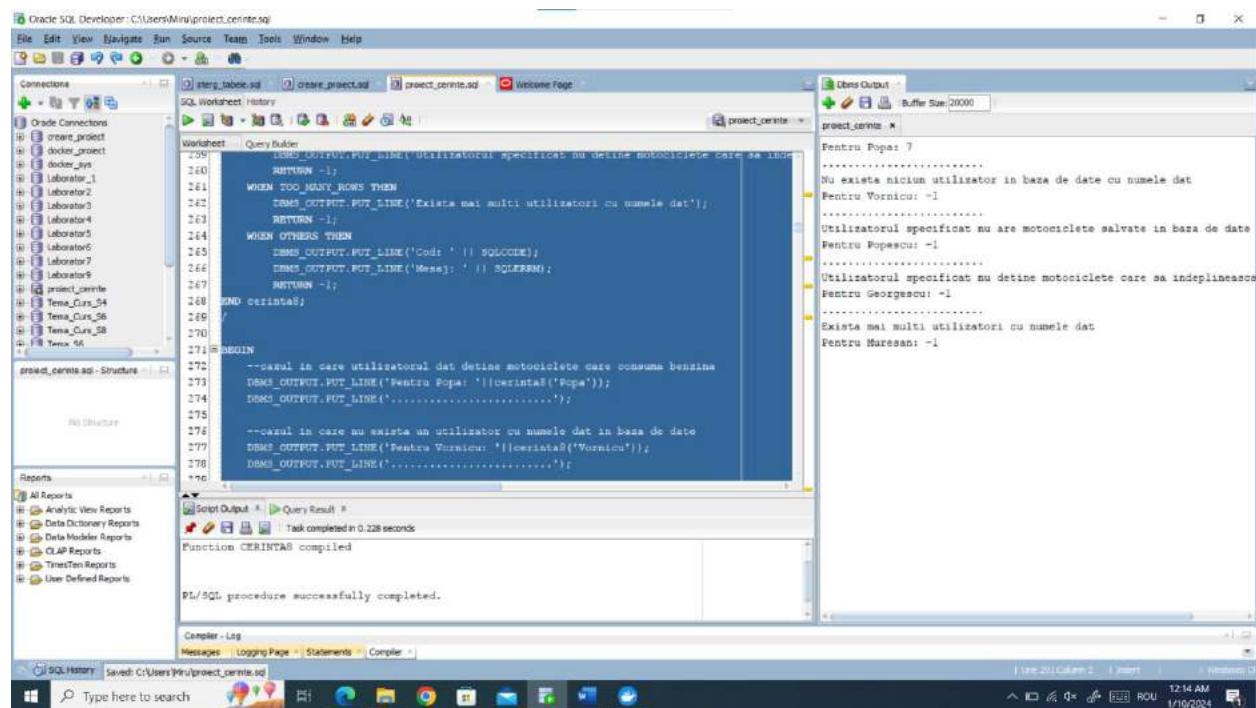
Stoine Maria-Miruna

Grupa 242



```
141    --Verificam motocicletele care merg pe benzina
142    WHERE M.ID_MOTOCICLETA IS NULL OR M.ID_MOTOCICLETA IN (
143        SELECT M1.ID_MOTOCICLETA
144        FROM MOTOCICLETA M1
145        JOIN MOTOCARE MT1 ON MT1.ID_MOTOR=M1.ID_MOTOR
146        WHERE UPPER(MT1.CARBURANT) = 'BENZINA')
147    GROUP BY U.ID_UTILIZATOR, U.NUME
148    HAVING UPPER(U.NUME)=UPPER(v_nume);
149
150    RETURN nr_motociclete;
151
152    EXCEPTION
153        WHEN NO_DATA_FOUND_UTILIZATOR THEN
154            DBMS_OUTPUT.PUT_LINE('Nu exista niciun utilizator in baza de date cu numele dat');
155            RETURN -1;
156        WHEN NO_DATA_FOUND_MOTOCICLETA THEN
157            DBMS_OUTPUT.PUT_LINE('Utilizatorul specificat nu are motociclete salvate in baza de date');
158            RETURN -1;
159        WHEN NO_DATA_FOUND THEN
160            DBMS_OUTPUT.PUT_LINE('Utilizatorul specificat nu detine motociclete care sa indeplineasca criteriile cerintei');
161            RETURN -1;
162        WHEN TOO_MANY_ROWS THEN
163            DBMS_OUTPUT.PUT_LINE('Există mai multi utilizatori cu numele dat');
```

PL/SQL procedure successfully completed.



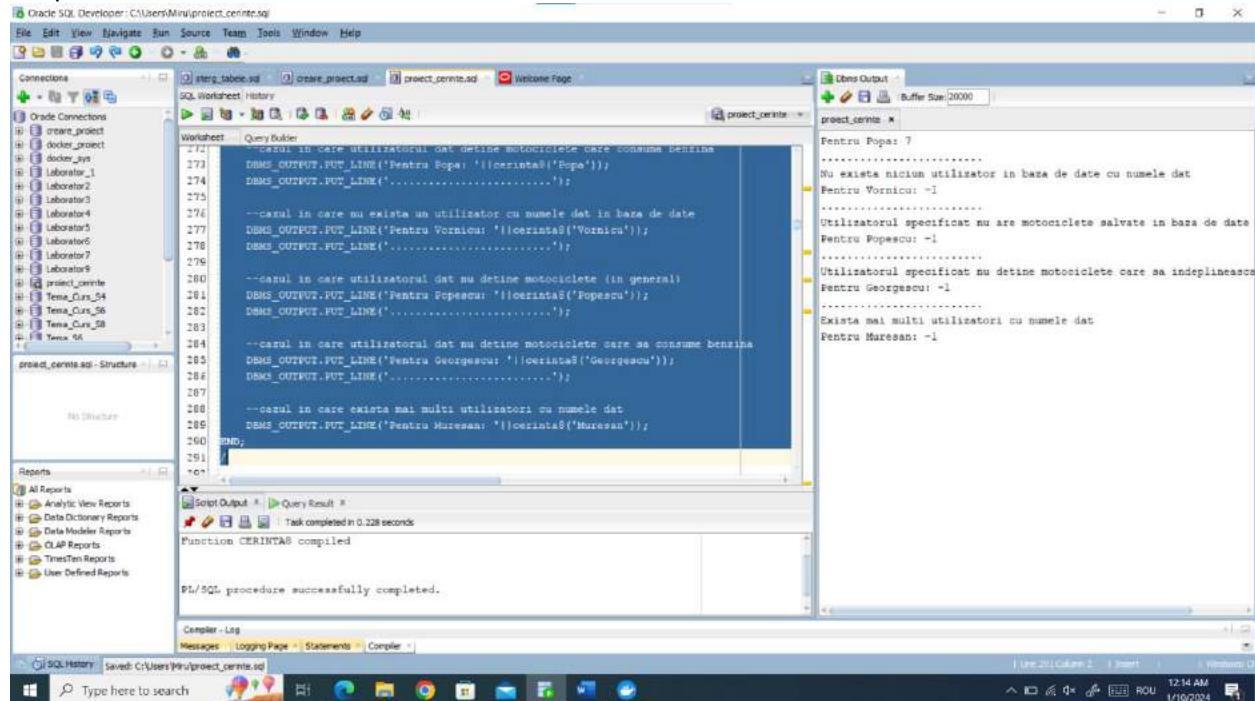
```
159        DBMS_OUTPUT.PUT_LINE('Utilizatorul specificat nu detine motociclete care sa indeplineasca criteriile cerintei');
160        RETURN -1;
161    WHEN TOO_MANY_ROWS THEN
162        DBMS_OUTPUT.PUT_LINE('Există mai multi utilizatori cu numele dat');
163        RETURN -1;
164    WHEN OTHERS THEN
165        DBMS_OUTPUT.PUT_LINE('Cod: ' || SQLCODE);
166        DBMS_OUTPUT.PUT_LINE('Mesaj: ' || SQRERRM);
167        RETURN -1;
168    END cerinta$;
169
170
171    BEGIN
172        --casul in care utilizatorul dat detine motociclete care consuma benzina
173        DBMS_OUTPUT.PUT_LINE('Pentru Popai: '||cerinta$('Popa'));
174        DBMS_OUTPUT.PUT_LINE('');
175
176        --casul in care nu exista un utilizator cu numele dat in baza de date
177        DBMS_OUTPUT.PUT_LINE('Pentru Vornicu: '||cerinta$('Vornicu'));
178        DBMS_OUTPUT.PUT_LINE('');
```

PL/SQL procedure successfully completed.

## Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242



**9. Formulați în limbaj natural o problemă pe care să o rezolvati folosind un**

*subprogram stocat independent de tip procedură care să utilizeze într-o*

*singură comandă SQL 5 dintre tabelele definite. Tratați toate excepțiile care*

*pot apărea, incluzând excepțiile NO\_DATA\_FOUND și TOO\_MANY\_ROWS.*

**Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.**

- *Cerință în limbaj natural*

În cadrul unui studiu se cere să se afle care este, în medie, suma pe care o alocă șoferii tineri adulți (utilizatorii cu vârstă cuprinsă între 25 și 34 de ani) în scopul întreținerii motocicletelor personale care fac parte dintr-o categorie care conține un string specificat. Astfel, se va afișa numele categoriei pentru care se face studiul și o scurtă descriere a acesteia, iar apoi suma medie cerută și numărul de persoane (utilizatori) care au participat la studiu.

Pentru această cerință, voi mai adăuga o categorie „Sport Cruiser” pentru a evidenția toate cazurile de excepții, inclusiv cea denumită TOO\_MANY\_ROWS.

➤ *Cod SQL Adăugare Înregistrare*

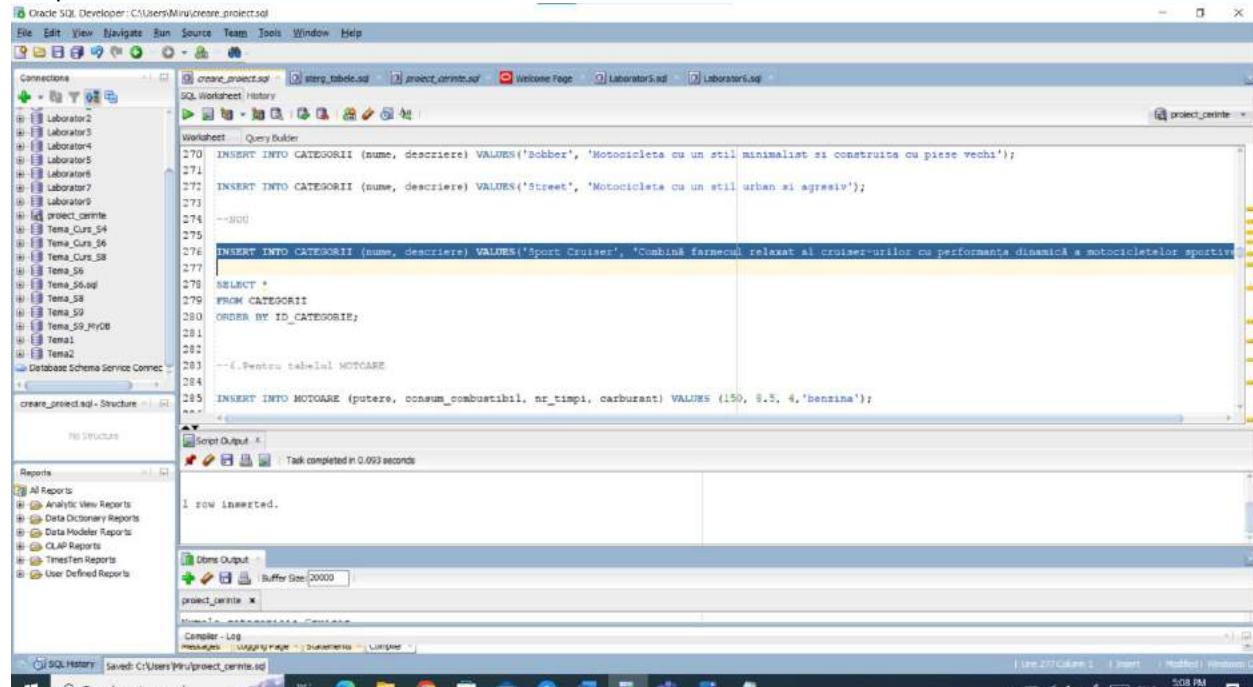
```
INSERT INTO CATEGORII (nume, descriere) VALUES('Sport Cruiser', 'Combină farmecul relaxat al cruiser-urilor cu performanța dinamică a motocicletelor sportive');
```

➤ Print-Screen Adaugare Inregistrare

## Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

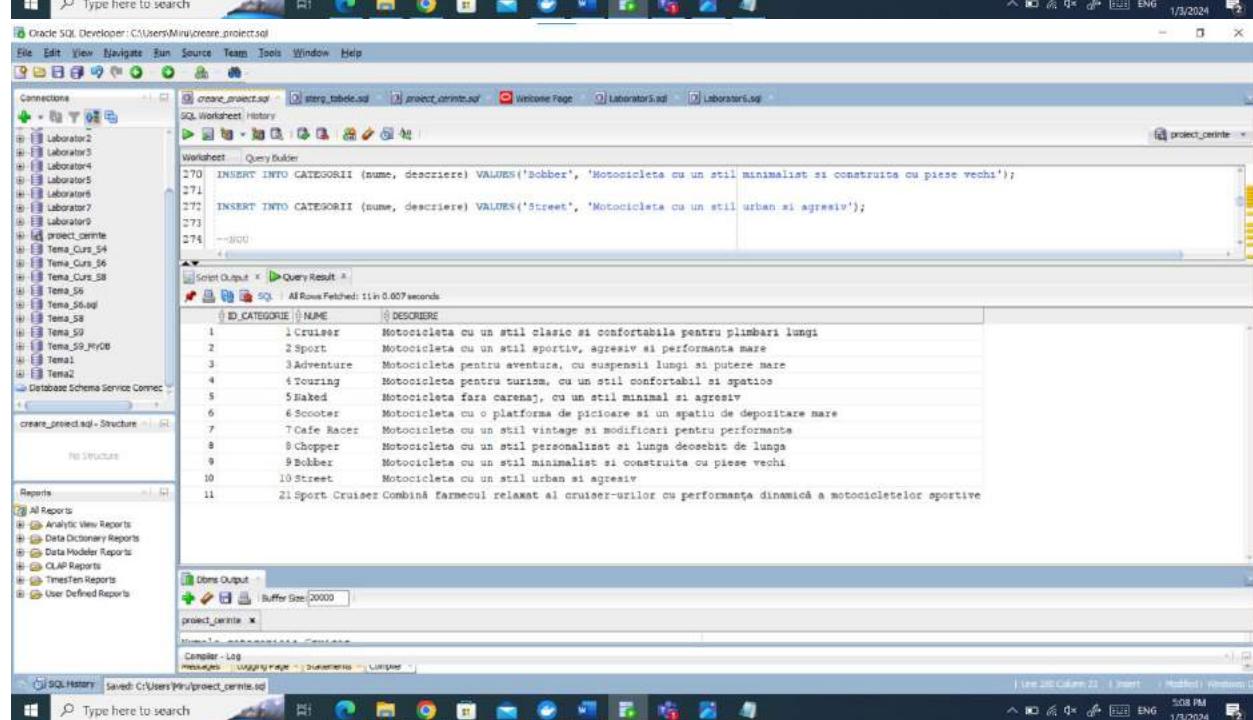
Grupa 242



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

- Connections:** A tree view showing various database connections.
- Worksheet:** A query builder window containing the following PL/SQL code:

```
270 INSERT INTO CATEGORII (nume, descriere) VALUES('Bobber', 'Motocicleta cu un stil minimalist si construita cu piese vechi');
271
272 INSERT INTO CATEGORII (nume, descriere) VALUES('Street', 'Motocicleta cu un stil urban si agresiv');
273
274 --SQL
275
276 INSERT INTO CATEGORII (nume, descriere) VALUES('Sport Cruiser', 'Combina faramul relaxat al cruiser-urilor cu performanta dinamica a motocicletelor sportive');
277
278 SELECT *
279 FROM CATEGORII;
280 ORDER BY ID_CATEGORIE;
281
282 --i. Pentru tabelul MOTOCARE
283
284 INSERT INTO MOTOCARE (putere, consum_combustibil, nr_timpuri, carburant) VALUES (150, 8.5, 4,'benzina');
```
- Script Output:** Shows the message "1 row inserted."
- Query Result:** Shows the results of the SELECT query from the CATEGORII table, listing 11 rows with columns ID\_CATEGORIE, NUME, and DESCRIERE. The data includes categories like Cruiser, Sport, Adventure, Touring, Naked, Scooter, Cafe Racer, Chopper, Bobber, Street, and Sport Cruiser.
- Reports:** A sidebar showing report options.
- Bottom Bar:** Shows the status bar with "Saved: C:\Users\Minu\project\_permitte.sql", the date "1/3/2024", and the time "5:08 PM".



## Rezolvare cerinta

➤ Cod PL/SQL

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

CREATE OR REPLACE PROCEDURE cerinta9

```
(v_categorie categorii.nume%TYPE DEFAULT 'Naked')
```

IS

```
NO_DATA_FOUND_INTRETINERE EXCEPTION;
```

```
NO_DATA_FOUND_MOTOCICLETA EXCEPTION;
```

```
contor NUMBER;
```

```
nr_utilizatori NUMBER;
```

```
nr_intretineri NUMBER;
```

```
cost_total NUMBER;
```

```
suma_medie NUMBER;
```

```
categ_info categorii.descriere%TYPE;
```

BEGIN

```
DBMS_OUTPUT.PUT_LINE('Numele categoriei: '||v_categorie);
```

SELECT

```
COUNT(*) INTO contor
```

```
FROM CATEGORII
```

```
WHERE INSTR(UPPER(NUME), UPPER(v_categorie))>0;
```

--exceptie pentru cazul in care nu exista categoria data

IF contor=0 THEN

```
RAISE NO_DATA_FOUND;
```

ELSIF contor>1 THEN

```
RAISE TOO_MANY_ROWS;
```

END IF;

SELECT

```
DESCRIBE INTO categ_info
```

```
FROM CATEGORII
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
WHERE INSTR(UPPER(NUME), UPPER(v_categorie))>0;
```

```
DBMS_OUTPUT.PUT_LINE('Descriere: '||categ_info);
```

```
SELECT
```

```
    COUNT(*) INTO contor
  FROM MOTOCICLETE M
  JOIN CATEGORII C ON C.ID_CATEGORIE=M.ID_CATEGORIE
 WHERE UPPER(NUME)=UPPER(v_categorie);
```

--exceptie pentru cazul in care inca nu exista motociclete inregistrate in baza de date care sa faca parte din acea categorie

```
IF contor=0 THEN
```

```
    RAISE NO_DATA_FOUND_MOTOCICLETA;
```

```
END IF;
```

--selectam toate intretinerile detinute de utilizatorii cu varsta cuprinsa in intervalul dat alocate

--motocicletelor din categoria specificata ca parametru si retinem in cost\_total - suma totala a acestora, in

--nr\_intretineri - numarul acestora, in nr\_utilizatori - numarul de utilizatori cu specificatiile date

--care detin motociclete din categoria care contin stringul dat ca parametru

```
SELECT
```

```
    SUM(P.COST), COUNT(DISTINCT I.ID_INTRETINERE), COUNT(DISTINCT
I.ID_UTILIZATOR)
```

```
    INTO cost_total, nr_intretineri, nr_utilizatori
```

```
  FROM INTRETINERI I
```

```
  JOIN UTILIZATORI U ON U.ID_UTILIZATOR=I.ID_UTILIZATOR
```

```
  JOIN PLATI P ON P.ID_PLATA=I.ID_PLATA
```

```
  JOIN MOTOCICLETE M ON M.ID_MOTOCICLETA=I.ID_MOTOCICLETA
```

```
  JOIN CATEGORII C ON C.ID_CATEGORIE=M.ID_CATEGORIE
```

```
 WHERE U.VARSTA<=34 AND U.VARSTA>=25 AND INSTR(UPPER(C.NUME),
UPPER(v_categorie)) > 0;
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

--exceptie pentru cazul in care nu exista intretineri efectuate motocicletelor care sa respecte conditiile cerute

IF nr\_intretineri=0 THEN

    RAISE NO\_DATA\_FOUND\_INTRETINERE;

END IF;

--calculam media

suma\_medie:=cost\_total/nr\_intretineri;

DBMS\_OUTPUT.PUT\_LINE('Suma medie alocata intretinerilor pentru aceasta categorie este  
'||suma\_medie);

DBMS\_OUTPUT.PUT\_LINE('Numarul de participanti la studiu: '|nr\_utilizatori);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

    DBMS\_OUTPUT.PUT\_LINE('Nu exista nicio categorie in baza de date cu numele specificat');

WHEN NO\_DATA\_FOUND\_MOTOCICLETA THEN

    DBMS\_OUTPUT.PUT\_LINE('Inca nu exista motociclete in baza de date din aceasta categorie');

WHEN NO\_DATA\_FOUND\_INTRETINERE THEN

    DBMS\_OUTPUT.PUT\_LINE('Nu exista intretineri efectuate pentru motocicletele din categoria specificata');

WHEN TOO\_MANY\_ROWS THEN

    DBMS\_OUTPUT.PUT\_LINE('Există mai multe categorii care să contină acest nume');

WHEN OTHERS THEN

    DBMS\_OUTPUT.PUT\_LINE('Cod: ' || SQLCODE);

    DBMS\_OUTPUT.PUT\_LINE('Mesaj: ' || SQLERRM);

END cerinta9;

/

BEGIN

--Cazul ideal, in care exista intretineri alocate motocicletelor din categoria data

cerinta9('Naked');

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

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

--Cazul in care nu exista categoria cu numele specificat

```
cerinta9('Supersportive');
```

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

--Cazul in care nu exista motociclete in baza de date care sa faca parte din categoria specificata

```
cerinta9('Bobber');
```

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

--Cazul in care nu exista intretineri efectuate pentru motocicletele din categoria specificata

```
cerinta9('Adventure');
```

--Cazul in care exista mai multe categorii in baza de date care sa contina stringul specificat

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

```
cerinta9('Cruiser');
```

END;

/

➤ *Print-Screen*

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there are several tabs: 'sterig\_tablee.ad', 'create\_project.ad', 'project\_cerinte.ad', and 'Welcome Page'. The 'project\_cerinte.ad' tab is active. The 'SQL Worksheet' tab contains the following PL/SQL code:

```
104  CREATE OR REPLACE PROCEDURE cerinta9
105    (v_categorie categorii.nume%TYPE DEFAULT 'Naked')
106  AS
107    NO_DATA_FOUND INTEGRITATE EXCEPTION;
108    NO_DATA_FOUND_MOTOCICLETA EXCEPTION;
109    control NUMBER;
110    nr_utilizatori NUMBER;
111    nr_intretineri NUMBER;
112    cost_total NUMBER;
113    suma_medi NUMBER;
114    categ_info categorii.descriere%TYPE;
115
116  BEGIN
117
118    DBMS_OUTPUT.PUT_LINE('Numele categoriei: '||v_categorie);
119
120    SELECT
121      COUNT(*) INTO costor
122    FROM CATEGORII
123    WHERE INSTR(UPPER(NUME), UPPER(v_categorie))>0;
124
125    --acoptez pentru ca nu exista categoria data
126    IF costor=0 THEN
127      RAISE NO_DATA_FOUND;
128    ELSEIF costor>1 THEN
129      RAISE TOO_MANY_ROWS;
130    END IF;
131
132    SELECT
133      DESCRIBE INTO categ_info
134    FROM CATEGORII
135    WHERE INSTR(UPPER(NUME), UPPER(v_categorie))>0;
136    DBMS_OUTPUT.PUT_LINE('Descriere: '||categ_info);
137
138
139    SELECT
140      COUNT(*) INTO costor
141    FROM MOTOCICLETE M
142    JOIN CATEGORII C ON C.ID_CATEGORIE=M.ID_CATEGORIE
143    WHERE INSTR(UPPER(NUME), UPPER(v_categorie))>0;
```

The 'Output' tab on the right displays the results of the procedure execution:

```
Numele categoriei: Naked
Descrierea Motocicleta fara carenaj, cu un stil minimalist si agresiv
Suma media alocate intretinerilor pentru aceasta categorie este 61.33181818181818
Numarul de participanti la studiu: 5
.....
Numele categoriei: Supersportive
Nu existe nicio categorie in baza de date cu numele specificat
.....
Numele categoriei: Bobber
Descrierea Motocicleta cu un stil minimalist si construita cu piese vechi
Inca nu exista motociclete in baza de date din aceasta categorie
.....
Numele categoriei: Adventure
Descrierea Motocicleta pentru aventura, cu suspensii lungi si putere mare
Nu existsa intretineri efectuate pentru motocicletele din categoria specificata
.....
Numele categoriei: Cruiser
Exista mai multe categorii care sa contin acest nume
```

The status bar at the bottom indicates: 'PL/SQL procedure successfully completed.' and the current date and time: '1/10/2024 12:18 AM'.

This screenshot is nearly identical to the one above, showing the same Oracle SQL Developer interface and the creation of the 'cerinta9' procedure. The code and output are identical, indicating a successful compilation and execution of the procedure.

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

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

- Connections:** Shows various database connections including 'create\_project', 'laborator\_1', 'laborator\_2', 'laborator\_3', 'laborator\_4', 'laborator\_5', 'laborator\_6', 'laborator\_7', 'laborator\_8', 'laborator\_9', 'project\_center', 'Tema\_Curs\_54', 'Tema\_Curs\_56', 'Tema\_Curs\_58', and 'Tema\_64'.
- SQL Worksheet:** Displays the PL/SQL code for procedure CERINTA9. The code performs several joins and calculations to find categories with specific characteristics.
- Script Output:** Shows the successful compilation of the procedure.
- Messages:** Displays log messages related to the procedure's execution.
- Compiler - Log:** Provides detailed logs of the compilation process.
- SQL History:** Shows the saved file path: C:\Users\Minu\project\_center.adb.

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

- Connections:** Shows various database connections including 'create\_project', 'laborator\_1', 'laborator\_2', 'laborator\_3', 'laborator\_4', 'laborator\_5', 'laborator\_6', 'laborator\_7', 'laborator\_8', 'laborator\_9', 'project\_center', 'Tema\_Curs\_54', 'Tema\_Curs\_56', 'Tema\_Curs\_58', and 'Tema\_64'.
- SQL Worksheet:** Displays the PL/SQL code for procedure CERINTA9. This version includes DBMS\_OUTPUT.PUT\_LINE statements to print results to the console.
- Script Output:** Shows the successful compilation of the procedure.
- Messages:** Displays log messages related to the procedure's execution.
- Compiler - Log:** Provides detailed logs of the compilation process.
- SQL History:** Shows the saved file path: C:\Users\Minu\project\_center.adb.

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
179:      WHEN TOO_MANY_ROWS THEN
180:        DBMS_OUTPUT.PUT_LINE('Există mai multe categorii care să conțin acest nume');
181:      WHEN OTHERS THEN
182:        DBMS_OUTPUT.PUT_LINE('Cod: ' || SQLCODE);
183:        DBMS_OUTPUT.PUT_LINE('Mesaj: ' || SQLERRM);
184:    END cerinta9;
185:
186:
187:  BEGIN
188:    --Casul ideal, în care există întrețineri alocate motocicletelor din categoria data
189:    --cerinta9('Naked');
190:    DBMS_OUTPUT.PUT_LINE('.....');
191:
192:    --Casul în care nu există categorii cu numele specificat
193:    --cerinta9('Supersportive');
194:    DBMS_OUTPUT.PUT_LINE('.....');
195:
196:    --Casul în care nu există motociclete în baza de date care să facă parte din categoria specificată
197:    --cerinta9('Bobber');
198:    DBMS_OUTPUT.PUT_LINE('.....');
199:
200:    --Casul în care nu există întrețineri efectuate pentru motocicletele din categoria specificată
201:    --cerinta9('Adventure');
202:    DBMS_OUTPUT.PUT_LINE('.....');
203:
204:    --Casul în care există mai multe categorii în baza de date care să conțin strîngul specificat
205:    DBMS_OUTPUT.PUT_LINE('.....');
206:    cerinta9('cruiser');
207:
208:  END;
```

PL/SQL procedure successfully completed.

```
180:      cerinta9('Naked');
181:      DBMS_OUTPUT.PUT_LINE('.....');
182:
183:      --Casul în care nu există categoria cu numele specificat
184:      --cerinta9('Supersportive');
185:      DBMS_OUTPUT.PUT_LINE('.....');
186:
187:      --Casul în care nu există motociclete în baza de date care să facă parte din categoria specificată
188:      --cerinta9('Bobber');
189:      DBMS_OUTPUT.PUT_LINE('.....');
190:
191:      --Casul în care nu există întrețineri efectuate pentru motocicletele din categoria specificată
192:      --cerinta9('Adventure');
193:
194:      --Casul în care există mai multe categorii în baza de date care să conțin strîngul specificat
195:      DBMS_OUTPUT.PUT_LINE('.....');
196:      cerinta9('cruiser');
```

PL/SQL procedure successfully completed.

**10. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.**

Voi defini un trigger la nivel de comandă, denumit „cerinta10”, cu următoarele funcționalități:

- Operațiile de actualizare (UPDATING) ale tabelului INTRETINERI se pot realiza doar în intervalul orar 8:00 și 20:00, în timpul zilelor lucrătoare (de luni până vineri). În cazul neîndeplinirii unei condiții dintre acestea, triggerul va ridica o eroare, blocând astfel actualizarea și afișând un mesaj specific.
  - Operațiile de stergere (DELETING) și de inserare (INSERTING) ale tabelului INTRETINERI se pot realiza doar în intervalul orar 8:00 și 20:00, în timpul zilelor lucrătoare (de luni până vineri) și în zilele din lună care nu coincid cu prima sau ultima zi. La fel ca în cazul de mai sus, în cazul neîndeplinirii unei condiții dintre acestea, triggerul va ridica o eroare, blocând astfel stergerea/inserarea și afișând un mesaj specific.

```
CREATE OR REPLACE TRIGGER cerinta10
BEFORE INSERT OR UPDATE OR DELETE ON INTRETNERI
DECLARE
    ora NUMBER;
    zi_saptamana VARCHAR(20);
    zi_luna NUMBER;
BEGIN
    --selectam ora (numarul din doua cifre), ziua saptamanii, ziua lunii
    SELECT
        TO_NUMBER(TO_CHAR(SYSDATE, 'HH24')), TO_CHAR(SYSDATE, 'DY'),
        TO_NUMBER(TO_CHAR(SYSDATE, 'DD'))
    INTO ora, zi_saptamana, zi_luna
    FROM DUAL;

    IF (ora < 8 OR ora > 20) OR zi_saptamana IN ('SUN', 'SAT') OR zi_luna IN (1, EXTRACT(DAY
    FROM LAST_DAY(SYSDATE))) THEN
        IF INSERTING THEN
            RAISE_APPLICATION_ERROR(-20001, 'Inserarea in tabelul INTRETNERI se poate realiza de
            luni pana vineri, intre orele 8:00-20:00 si nu in prima sau ultima zi a lunii!');
        ELSIF DELETING THEN
            RAISE_APPLICATION_ERROR(-20002, 'Stergerea in tabelul INTRETNERI se poate realiza de
            luni pana vineri, intre orele 8:00-20:00 si nu in prima sau ultima zi a lunii!');
        END IF;
       ELSIF (ora < 8 OR ora > 20) OR zi_saptamana IN ('SUN', 'SAT') THEN
        IF UPDATING THEN
            RAISE_APPLICATION_ERROR(-20003, 'Actualizările in tabelul INTRETNERI se poate realiza
            de luni pana vineri, intre orele 8:00-20:00!');
        END IF;
    END IF;
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

END;

/

--varianta care declanseaza triggerul

SELECT

```
TO_CHAR(SYSDATE, 'HH24') ora, TO_CHAR(SYSDATE, 'DY') zi_saptamana,  
TO_NUMBER(TO_CHAR(SYSDATE, 'DD')) zi_luna
```

FROM dual;

INSERT INTO INTRETINERI (id\_motocicleta, id\_utilizator, id\_plata, nume, descriere, data)

```
VALUES (18, 13, 10, 'Schimb ulei', 'Schimb de ulei la motor', '22-DEC-2023');
```

--varianta care nu il declanseaza

SELECT

```
TO_CHAR(SYSDATE, 'HH24') ora, TO_CHAR(SYSDATE, 'DY') zi_saptamana,  
TO_NUMBER(TO_CHAR(SYSDATE, 'DD')) zi_luna
```

FROM dual;

INSERT INTO INTRETINERI (id\_motocicleta, id\_utilizator, id\_plata, nume, descriere, data)

```
VALUES (18, 13, 10, 'Schimb ulei', 'Schimb de ulei la motor', '22-DEC-2023');
```

➤ *Print-Screen*

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there are connections and a project structure for 'project\_center'. The 'SQL Worksheet' tab is active, displaying the following PL/SQL code:

```
--10. Definiti un trigger de tip IMD la nivel de comandă. Declanșează trigger-ul.
CREATE OR REPLACE TRIGGER cerintal0
BEFORE INSERT OR UPDATE OR DELETE ON INTRETIMERI
DECLARE
    ora NUMBER;
    zi_septamana VARCHAR(20);
    zi_luna NUMBER;
BEGIN
    --selectam ora (numarul din două cifre), ziua săptămânii, ziua lunii
    SELECT
        TO_NUMBER(TO_CHAR(SYSDATE, 'HH24')), TO_CHAR(SYSDATE, 'DY'), TO_NUMBER(TO_CHAR(SYSDATE, 'DD'))
    INTO ora, zi_septamana, zi_luna
    FROM DUAL;

    IF (ora < 8 OR ora > 20) OR zi_septamana IN ('SUN', 'SAT') OR zi_luna IN (1, EXTRACT(DAY FROM LAST_DAY(SYSDATE))) THEN
        IF INSERTING THEN
            RAISE_APPLICATION_ERROR(-20001,'Inserarea în tabelul INTRETIMERI se poate realiza de luni până vineri, între orele
        ELSEIF DELETING THEN
            RAISE_APPLICATION_ERROR(-20002,'Ștergerea din tabelul INTRETIMERI se poate realiza de luni până vineri, între orele
        END IF;
    ELSEIF (ora < 8 OR ora > 20) OR zi_septamana IN ('SUN', 'SAT') THEN
        IF UPDATING THEN
            RAISE_APPLICATION_ERROR(-20003,'Actualizările în tabelul INTRETIMERI se pot realiza de luni până vineri, între orele 8:00-20:00!');
        END IF;
    END IF;
END /
```

The 'Script Output' panel shows the message: "Trigger CERINTA10 compiled". The status bar at the bottom right indicates the date and time: "1/11/2024 1:30 AM".

This screenshot is identical to the one above, showing the Oracle SQL Developer interface with the same code for the trigger CERINTA10. The code defines a trigger that restricts insertions, deletions, and updates to specific times of day and days of the week. The 'Script Output' panel shows the trigger has been successfully compiled. The status bar at the bottom right shows the date and time: "1/11/2024 1:30 AM".

- Varianta care declanșează triggerul, testat la ora 23:31 UTC

## Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. In the central workspace, there is a query editor window titled "Query Builder". The code in the editor is as follows:

```
RAISE_APPLICATION_ERROR(-20003, 'Accesabilitatea in tactiul INTREAINERI se poate realiza de luni pana vineri, intre orele 8:00-20:00');
END IF;
END IF;
/
--varianta care declanseaza triggerul
SELECT
    TO_CHAR(SYSDATE, 'MM2') ora, TO_CHAR(SYSDATE, 'DY') zi_septamana, TO_NUMBER(TO_CHAR(SYSDATE, 'DO')) zi_luna
FROM dual;

INSERT INTO INTREAINERI (id_motociclista, id_utilizator, id_plata, nume, descriere, data)
VALUES (18, 13, 10, 'Schimb ulei', 'Schimb de ulei la motor', '22-DEC-2023');
```

Below the editor, the "Script Output" tab is selected, showing the following error message:

```
Error starting at line : 447 in command -
INSERT INTO INTREAINERI (id_motociclista, id_utilizator, id_plata, nume, descriere, data)
VALUES (18, 13, 10, 'Schimb ulei', 'Schimb de ulei la motor', '22-DEC-2023')
Error report -
ORA-20001: Inserarea in tabelul INTREAINERI se poate realiza de luni pana vineri, intre orele 8:00-20:00 si nu in prima sau ultima zi a lunii!
ORA-06512: at "PROJECT.CERINTA10", line 14
ORA-04080: error during execution of trigger 'PROJECT.CERINTA10'
```

The status bar at the bottom right indicates the date and time as 1/11/2024 13:11 AM.

This screenshot shows the same Oracle SQL Developer environment as the previous one, but the "Script Output" tab now displays a successful execution message:

```
SQL> All Rows Fetched: 1 in 0.005 seconds
```

Below the output, the "Query Result" tab is selected, showing the results of the query:

ORA	ZI_SEPTAMANA	ZI_LUNA
123	WED	10

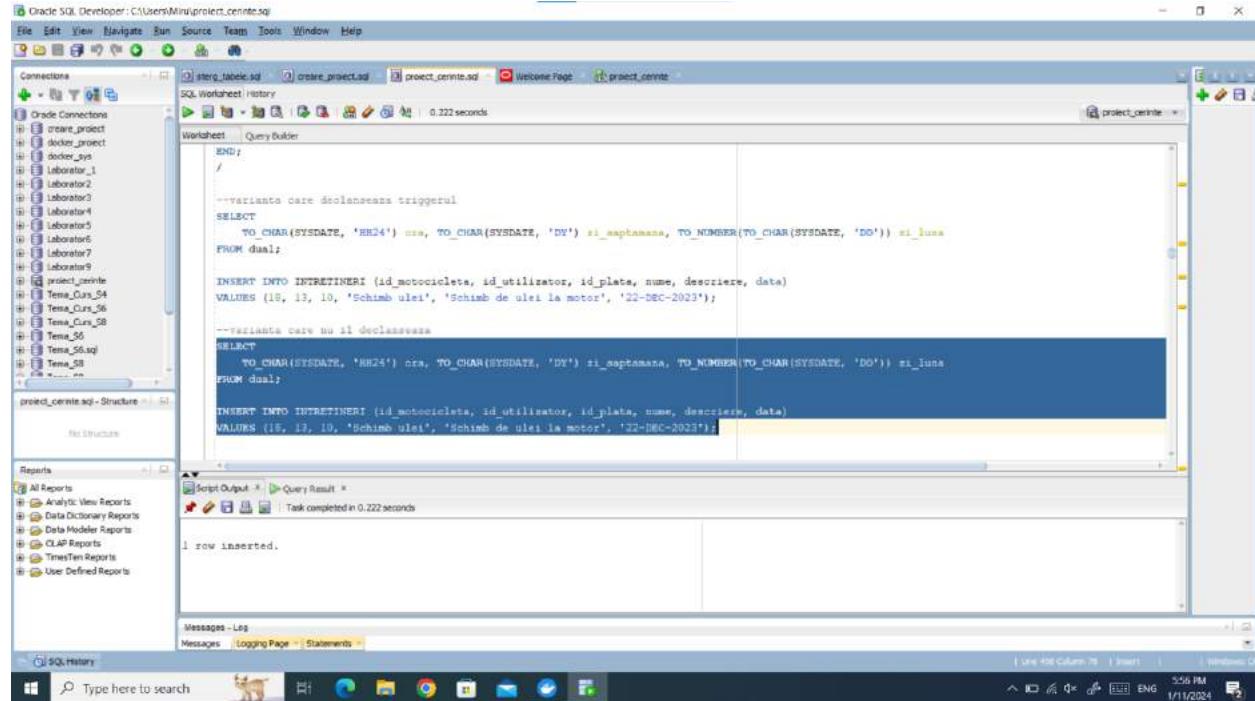
The status bar at the bottom right indicates the date and time as 1/11/2024 13:11 AM.

- Varianta care nu declanșează triggerul, testat la ora 15:57 UTC

## Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242



The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar lists several projects and tables, including 'project\_center' and various 'Tema\_Curs' tables. The 'Worksheet' tab contains a query builder with the following PL/SQL code:

```
END;
/
--varianta care declansaza triggerul
SELECT
    TO_CHAR(SYSDATE, 'HH24') ora, TO_CHAR(SYSDATE, 'DY') zi_saptamana, TO_NUMBER(TO_CHAR(SYSDATE, 'DO')) zi_luna
FROM dual;

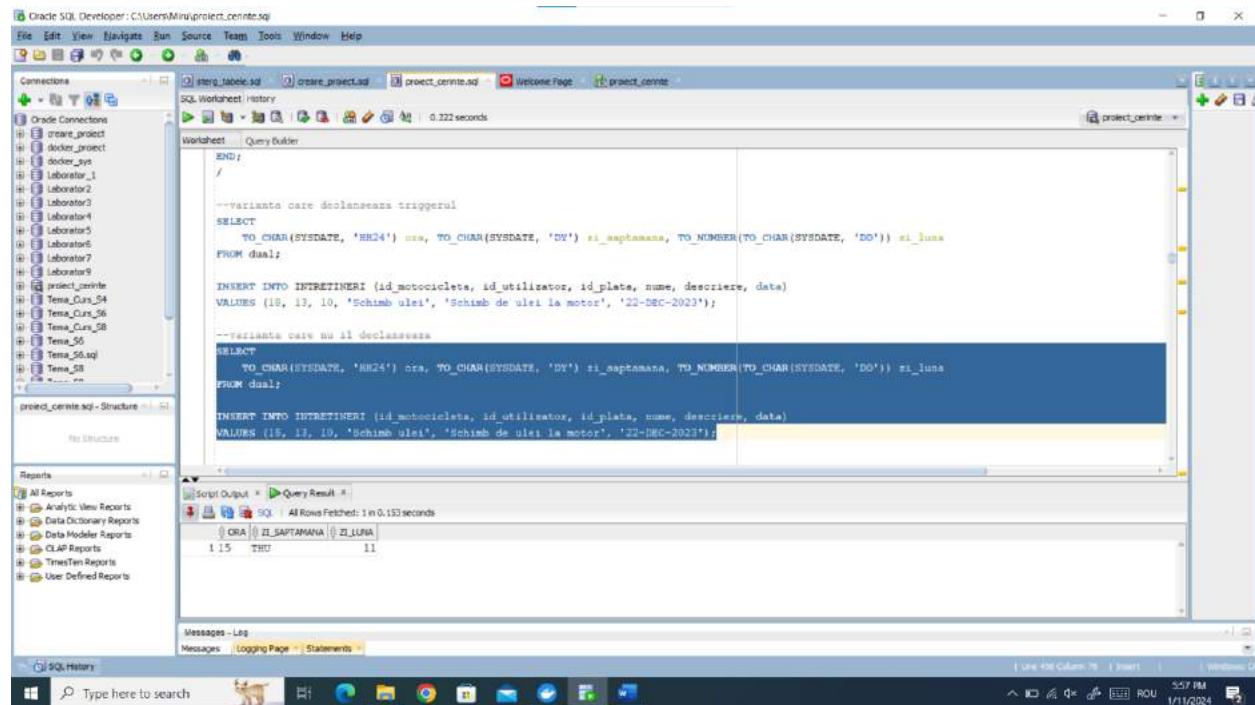
INSERT INTO INTREAINERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
VALUES (10, 13, 10, 'Schimb ulei', 'Schimb de ulei la motor', '22-DEC-2023');

--varianta care nu il declansaza
SELECT
    TO_CHAR(SYSDATE, 'HH24') ora, TO_CHAR(SYSDATE, 'DY') zi_saptamana, TO_NUMBER(TO_CHAR(SYSDATE, 'DO')) zi_luna
FROM dual;

INSERT INTO INTREAINERI (id_motocicleta, id_utilizator, id_plata, nume, descriere, data)
VALUES (10, 13, 10, 'Schimb ulei', 'Schimb de ulei la motor', '22-DEC-2023');
```

The 'Script Output' tab shows the result of the first insert statement:

```
1 row inserted.
```



This screenshot is identical to the one above, showing the same Oracle SQL Developer interface and the execution of the same PL/SQL code. The 'Script Output' tab now displays the results of both insert statements:

ORA	ZI_SAPTA MANA	ZI_LUNA
11:15	THU	11

---

**11. Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.**

---

Voi defini un trigger la nivel de linie, denumit „cerinta11”, cu următoarele funcționalități:

- Operațiile de inserare (INSERT) și modificare (UPDATE) aplicate asupra tabelului MOTOCICLETE se pot realiza pentru coloana „stare” cu valoarea „Excelenta” numai dacă motocicleta nu este mai veche de 5 ani și are un kilometraj sub 5000. În cazul neîndeplinirii acestor condiții, triggerul va ridica o eroare, blocând astfel actualizarea / inserarea și afișează un mesaj specific.
- Operațiile de inserare (INSERT) și modificare (UPDATE) aplicate asupra tabelului MOTOCICLETE se pot realiza pentru coloana „stare” cu valoarea „Foarte buna” numai dacă motocicleta nu este mai veche de 8 ani și are un kilometraj sub 10000. În cazul neîndeplinirii acestor condiții, triggerul va ridica o eroare, blocând astfel actualizarea / inserarea și afișează un mesaj specific.
- Operațiile de inserare (INSERT) și modificare (UPDATE) aplicate asupra tabelului MOTOCICLETE se pot realiza pentru coloana „stare” cu valoarea „Buna” numai dacă motocicleta nu este mai veche de 11 ani și are un kilometraj sub 30000. În cazul neîndeplinirii acestor condiții, triggerul va ridica o eroare, blocând astfel actualizarea / inserarea și afișează un mesaj specific.

➤ *Cod PL/SQL*

```
CREATE OR REPLACE TRIGGER cerinta11
BEFORE INSERT OR UPDATE ON MOTOCICLETE
FOR EACH ROW
DECLARE
    varsta_motocicleta INT;
BEGIN
    varsta_motocicleta := EXTRACT(YEAR FROM SYSDATE)-:NEW.AN;
    IF varsta_motocicleta>5 AND :NEW.KILOMETRAJ>10000 AND UPPER(:NEW.STARE) =
'EXCELENȚA' THEN
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
    RAISE_APPLICATION_ERROR(-20001, 'Starea "Excelenta" nu este permisa pentru acest grad de uzura varsta-kilometraj.');
```

```
ELSIF varsta_motocicleta>8 AND :NEW.KILOMETRAJ>100000 AND UPPER(:NEW.STARE) = 'FOARTE BUNA' THEN
```

```
    RAISE_APPLICATION_ERROR(-20001, 'Starea "Foarte buna" nu este permisa pentru acest grad de uzura varsta-kilometraj.');
```

```
ELSIF varsta_motocicleta>11 AND :NEW.KILOMETRAJ>300000 AND UPPER(:NEW.STARE) = 'BUNA' THEN
```

```
    RAISE_APPLICATION_ERROR(-20001, 'Starea "Buna" nu este permisa pentru acest grad de uzura varsta-kilometraj.');
```

```
END IF;
```

```
END;
```

```
/
```

--varianta care declanseaza triggerul

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj, nr_inmatriculara, marca, model, stare, an, capacitate_rezervor)
```

```
VALUES (2, 4, 2, 6, 12000, 'E-012-XYZ', 'Suzuki', 'GSX-R1000', 'Excelenta', 2013, 19);
```

--varianta care nu declanseaza triggerul, deoarece respecta restrictiile cerute

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_far, kilometraj, nr_inmatriculara, marca, model, stare, an, capacitate_rezervor)
```

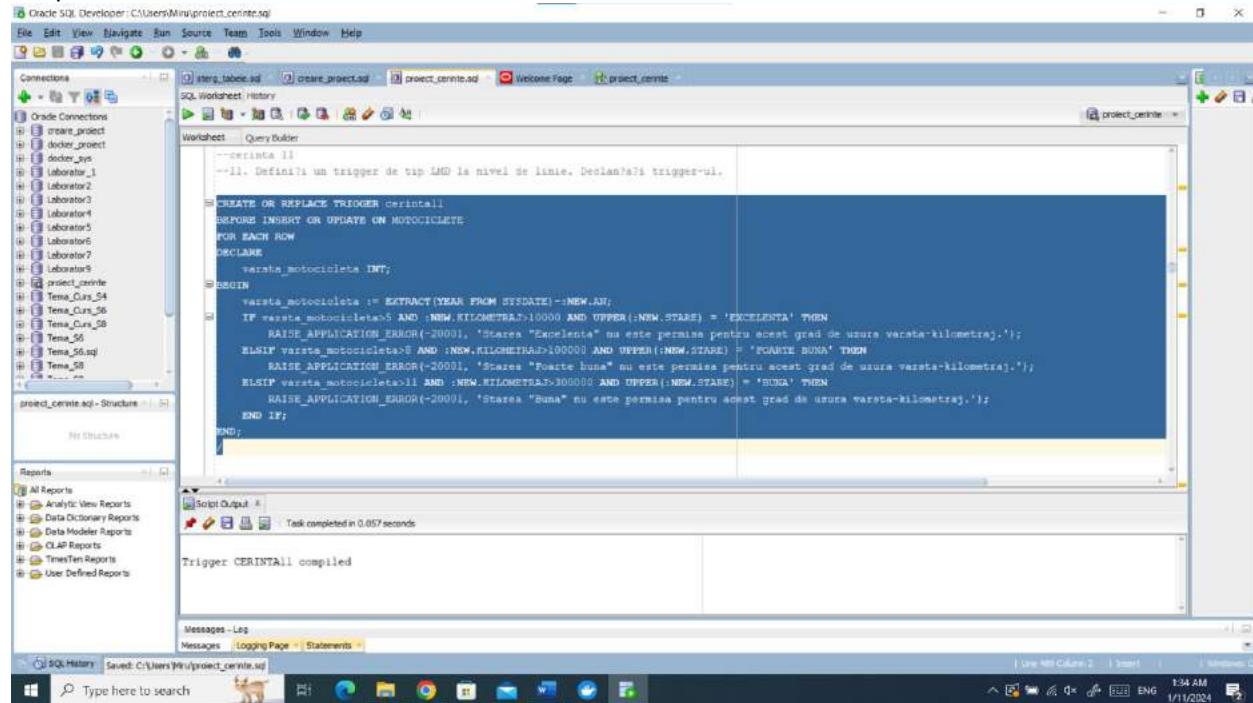
```
VALUES (2, 4, 2, 6, 9000, 'E-012-XYZ', 'Suzuki', 'GSX-R1000', 'Excelenta', 2013, 19);
```

➤ *Print-Screen*

## Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242



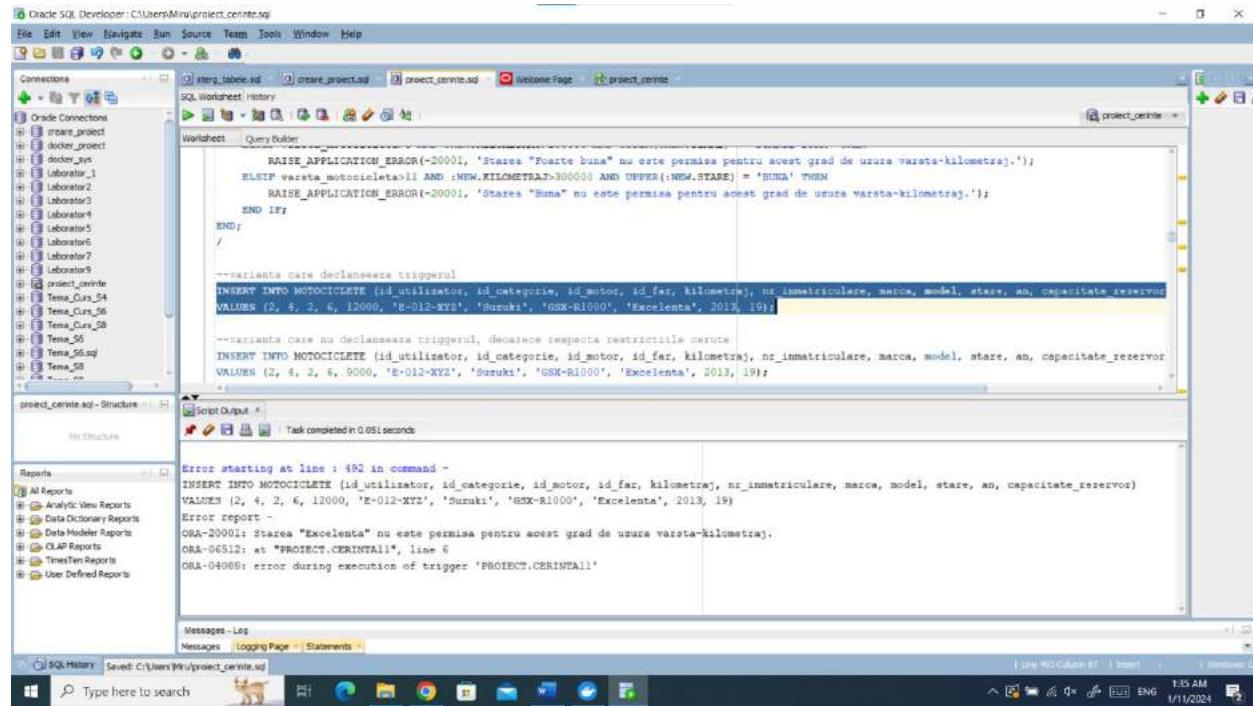
The screenshot shows the Oracle SQL Developer interface. In the central workspace, there is a 'Query Builder' window displaying the following PL/SQL code:

```
--cerintă 11
--11. Definiți un trigger de tip I&D la nivel de linie. Descrieți trigger-ul.

CREATE OR REPLACE TRIGGER cerintall
BEFORE INSERT OR UPDATE ON MOTOCICLETE
FOR EACH ROW
DECLARE
    varsta_motocicleta INT;
BEGIN
    varsta_motocicleta := EXTRACT(YEAR FROM SYSDATE)-:NEW.an;
    IF varsta_motocicleta<5 AND (:NEW.KILOMETRaj)>10000 AND UPPER(:NEW.stare) = 'EXCELENȚA' THEN
        RAISE_APPLICATION_ERROR(-20001, 'Starea "Exclenta" nu este permisa pentru acest grad de usura varsta-kilometraj.');
    ELSIF varsta_motocicleta>8 AND (:NEW.KILOMETRaj)>10000 AND UPPER(:NEW.stare) = 'FOARTE BUNA' THEN
        RAISE_APPLICATION_ERROR(-20001, 'Starea "Foarte buna" nu este permisa pentru acest grad de usura varsta-kilometraj.');
    ELSIF varsta_motocicleta>11 AND (:NEW.KILOMETRaj)>30000 AND UPPER(:NEW.stare) = 'BUNA' THEN
        RAISE_APPLICATION_ERROR(-20001, 'Starea "Buna" nu este permisa pentru acest grad de usura varsta-kilometraj.');
    END IF;
END;
```

The 'Script Output' panel below the code shows the message: 'Task completed in 0.057 seconds'. The 'Messages - Log' panel indicates that the trigger 'CERINTALL' was compiled successfully.

### ➤ Varianta care declanșează triggerul



The screenshot shows the Oracle SQL Developer interface. In the central workspace, there is a 'Query Builder' window displaying the following PL/SQL code:

```
--varianta care declanșează triggerul
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_fax, kilometraj, nr_inmatriculare, marca, model, stare, an, capacitate_rezervor)
VALUES (2, 4, 2, 6, 12000, 'E-012-XYZ', 'Suzuki', 'GSX-R1000', 'Excelenta', 2013, 19);

--varianta care nu declanșează triggerul, deoarece respectă restricțiile cerute
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_fax, kilometraj, nr_inmatriculare, marca, model, stare, an, capacitate_rezervor)
VALUES (2, 4, 2, 6, 9000, 'E-013-XXX', 'Suzuki', 'GSX-R1000', 'Excelenta', 2013, 19);
```

The 'Script Output' panel shows the error message: 'Error starting at line : 492 in command -'. The detailed error message is:

```
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_fax, kilometraj, nr_inmatriculare, marca, model, stare, an, capacitate_rezervor)
VALUES (2, 4, 2, 6, 12000, 'E-012-XYZ', 'Suzuki', 'GSX-R1000', 'Excelenta', 2013, 19)
Error report -
ORA-20001: Starea "Excelenta" nu este permisa pentru acest grad de usura varsta-kilometraj.
ORA-06512: at "PROJECT.CERINTALL", line 6
ORA-04008: error during execution of trigger 'PROJECT.CERINTALL'
```

The 'Messages - Log' panel shows the error details.

### ➤ Varianta care nu declanșează triggerul, deoarece respectă restricțiile impuse

## Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
END IF;
/
--varianță care declanșează triggerul
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_făr, kilometraj, nr_inmatriculare, marca, model, stare, an, capacitate rezervor)
VALUES (2, 4, 2, 6, 12000, 'E-012-XYZ', 'Suzuki', 'GSX-R1000', 'Excellent', 2013, 19);

--varianță care nu declanșează triggerul, deoarece respectă restricțiile cerute
INSERT INTO MOTOCICLETE (id_utilizator, id_categorie, id_motor, id_făr, kilometraj, nr_inmatriculare, marca, model, stare, an, capacitate rezervor)
VALUES (2, 4, 2, 6, 9000, 'E-012-XYZ', 'Suzuki', 'GSX-R1000', 'Excellent', 2013, 19);

--cerință 12
```

1 row inserted.

---

### 12. Definiți un trigger de tip LDD. Declanșați trigger-ul.

---

Voi defini un trigger de tip LDD, denumit „cerinta12”, cu următoarele funcționalități:

- La fiecare operație de tip CREATE, ALTER si DROP efectuată asupra bazei de date, trigger-ul va insera în tabelul monitor\_SMM informații cu privire la numele bazei de date, utilizatorul care a efectuat operația, tipul de eveniment (CREATE, ALTER, DELETE), tipul obiectului asupra căruia s-a efectuat operația (ex: TABLE, INDEX, VIEW), numele obiectului asupra căreia s-a efectuat operația și data la care a avut loc evenimentul.

➤ Cod PL/SQL

| CREATE TABLE monitor\_SMM(

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
nume_baza_de_date VARCHAR2(50),  
user_logat VARCHAR2(30),  
eveniment VARCHAR2(30),  
tip_object VARCHAR2(30),  
nume_object VARCHAR2(30),  
data DATE  
);
```

CREATE OR REPLACE TRIGGER cerinta12

AFTER CREATE OR DROP OR ALTER ON SCHEMA

DECLARE

```
v_nume_baza_de_date VARCHAR2(50);  
v_user_logat VARCHAR2(30);  
v_eveniment VARCHAR2(30);  
v_tip_object VARCHAR2(30);  
v_nume_object VARCHAR2(30);  
v_data DATE;
```

BEGIN

```
v_nume_baza_de_date := SYS.DATABASE_NAME;  
v_user_logat := SYS.LOGIN_USER;  
v_eveniment := SYS.SYSEVENT;  
v_tip_object := SYS.DICTIONARY_OBJ_TYPE;  
v_nume_object := SYS.DICTIONARY_OBJ_NAME;  
v_data := SYSDATE;
```

INSERT INTO monitor\_SMM

```
VALUES (v_nume_baza_de_date, v_user_logat, v_eveniment, v_tip_object, v_nume_object, v_data);
```

END;

/

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

CREATE TABLE GPS (

    ID\_GPS NUMBER PRIMARY KEY,

    Model VARCHAR2(50),

    Latitudine NUMBER,

    Longitudine NUMBER

);

SELECT \* FROM GPS;

ALTER TABLE GPS ADD (Status VARCHAR2(20));

ALTER TABLE GPS ADD (Precizie NUMBER);

ALTER TABLE GPS DROP COLUMN Status;

DROP TABLE GPS;

SELECT \*

FROM monitor\_SMM;

➤ *Print-Screen*

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : C:\Users\Minu\project\_cerinte.sql". The main window displays a SQL Worksheet titled "Query builder" containing the following PL/SQL code:

```
387 CREATE OR REPLACE TRIGGER cerintail2
388 AFTER CREATE OR DROP OR ALTER ON SCHEMA
389 DECLARE
390   v_name_baza_de_date VARCHAR2(50);
391   v_user_logat VARCHAR2(30);
392   v_eveniment VARCHAR2(30);
393   v_tip_object VARCHAR2(30);
394   v_name_object VARCHAR2(30);
395   v_data DATE;
396 BEGIN
397   v_name_baza_de_date := SYS.DATABASE_NAME;
398   v_user_logat := SYS.LOGIN_USER;
399   v_eveniment := SYS.EVENTNAME;
400   v_tip_object := SYS.DICTIONARY_OBJ_TYPE;
401   v_name_object := SYS.DICTIONARY_OBJ_NAME;
402   v_data := SYSDATE;
403
404   INSERT INTO monitor_SME
405   VALUES (v_name_baza_de_date, v_user_logat, v_eveniment, v_tip_object, v_name_object, v_data);
406 END;
```

The "Script Output" tab at the bottom shows the message: "Trigger CERINTAIL2 compiled". The status bar at the bottom right indicates "1 line, 472 columns, 1 insert, 1 modified, 0 removed, 0 rows affected" and the date/time "1/3/2024 10:41 PM".

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, the path is set to 'File > Open > Oracle SQL Developer: C:\Users\Minu\project\_center.sql'. The main window displays a 'Worksheet' tab with the following SQL code:

```
408 CREATE TABLE GPS (
409   ID_GPS NUMBER PRIMARY KEY,
410   Model VARCHAR2(50),
411   Latitudine NUMBER,
412   Longitudine NUMBER
413 );
414
415
416 SELECT * FROM GPS;
417
418 ALTER TABLE GPS ADD (Status VARCHAR2(20));
419
420 ALTER TABLE GPS ADD (Precisie NUMBER);
421
422 ALTER TABLE GPS DROP COLUMN Status;
```

Below the worksheet, the 'Script Output' tab shows the results of the executed queries:

```
Table GPS created.
>>Query Run In:Query Result
Table GPS altered.

Table GPS altered.
```

This screenshot shows the continuation of the Oracle SQL Developer session. The 'Worksheet' tab now contains additional SQL code:

```
412   Latitudine NUMBER,
413   Longitudine NUMBER
414 );
415
416 SELECT * FROM GPS;
417
418 ALTER TABLE GPS ADD (Status VARCHAR2(20));
419
420 ALTER TABLE GPS ADD (Precisie NUMBER);
421
422 ALTER TABLE GPS DROP COLUMN Status;
423
424 DROP TABLE GPS;
425
426
427 SELECT *
428 FROM monitor_SBM;
```

The 'Script Output' tab shows the results of the dropped table command:

```
Table GPS altered.

Table GPS altered.

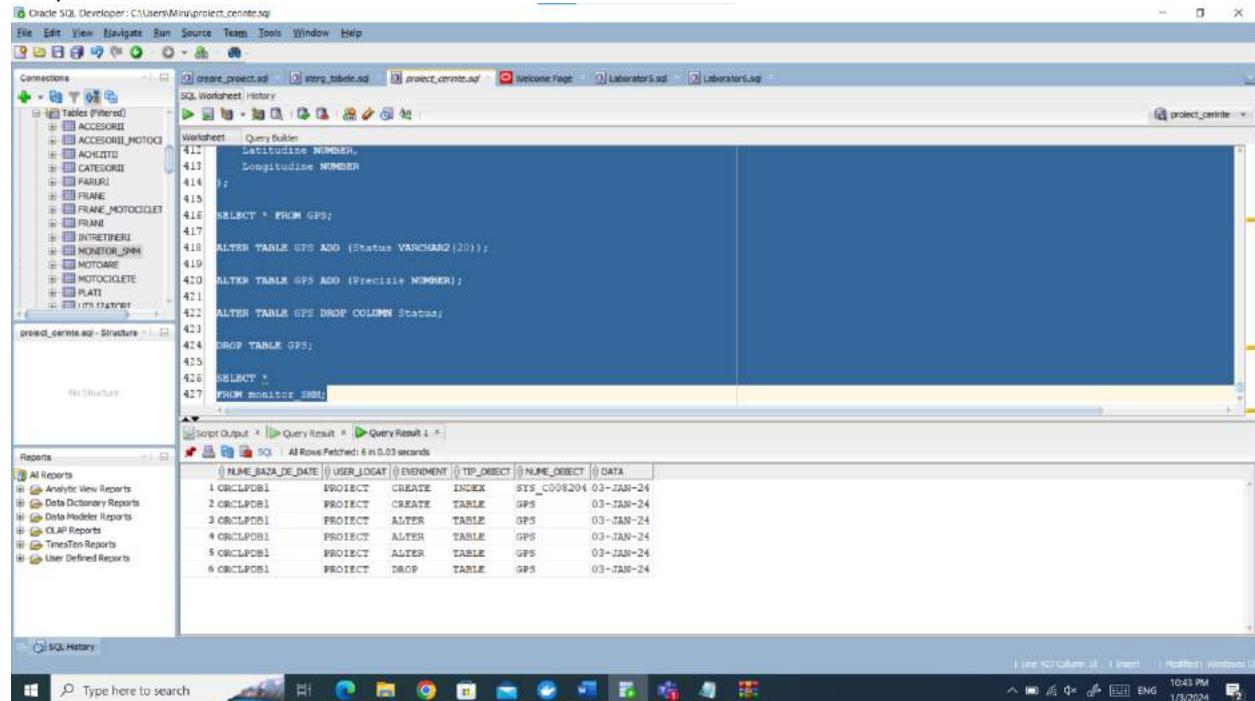
Table GPS altered.

Table GPS dropped.
```

## Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242



**13. Definiți un pachet care să conțină toate obiectele definite în cadrul**

**proiectului.**

➤ *Cod PL/SQL*

CREATE OR REPLACE PACKAGE cerinta13 IS

PROCEDURE cerinta6 (v\_kilometraj motociclete.kilometraj%TYPE DEFAULT 7000);

PROCEDURE cerinta7 (v\_nrmin\_moto INT DEFAULT 0);

FUNCTION cerinta8 (v\_numa utilizatori.nume%TYPE DEFAULT 'Georgescu') RETURN NUMBER;

PROCEDURE cerinta9 (v\_categorie categorii.nume%TYPE DEFAULT 'Georgescu');

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

END cerinta13;

/

CREATE OR REPLACE PACKAGE BODY cerinta13 IS

--cerinta 6

PROCEDURE cerinta6 (v\_kilometraj motociclete.kilometraj%TYPE DEFAULT 7000)

AS

TYPE tablou\_indexat IS TABLE OF NUMBER INDEX BY PLS\_INTEGER;

t\_indexat tablou\_indexat;

TYPE tablou\_imbricat IS TABLE OF NUMBER;

t\_imbricat tablou\_imbricat := tablou\_imbricat();

TYPE vector IS VARRAY(100) OF NUMBER;

v vector:= vector();

nr\_max INT;

v\_nume UTILIZATORI.NUME%TYPE;

nr\_accesorii INT;

nr\_mediul\_accesorii INT;

nr\_total\_accesorii INT;

BEGIN

SELECT MAX(COUNT(M.ID\_MOTOCICLETA))

INTO nr\_max

FROM MOTOCICLETE M

GROUP BY M.ID\_UTILIZATOR;

DBMS\_OUTPUT.PUT\_LINE('Numarul maxim de motociclete detinute de un utilizator: '|nr\_max);

DBMS\_OUTPUT.PUT\_LINE('.....');

```
DBMS_OUTPUT.PUT_LINE('Utilizatorii care detin numarul maxim de motociclete:');
DBMS_OUTPUT.PUT_LINE('.....');

FOR rec_id_utilizator IN ( SELECT M.ID_UTILIZATOR
                           FROM MOTOCICLETE M
                           GROUP BY M.ID_UTILIZATOR
                           HAVING COUNT(M.ID_MOTOCICLETA) = nr_max) LOOP
    t_indexat(rec_id_utilizator.ID_UTILIZATOR) := 1;
    SELECT
        (U.NUME||' '||U.PRENUME) INTO v_nume
    FROM UTILIZATORI U
    WHERE U.ID_UTILIZATOR=rec_id_utilizator.ID_UTILIZATOR;
    DBMS_OUTPUT.PUT_LINE(v_nume);
END LOOP;

DBMS_OUTPUT.PUT_LINE('Motocicletelor acestor utilizatori:');
DBMS_OUTPUT.PUT_LINE('.....');
DBMS_OUTPUT.PUT_LINE('.....');

FOR i IN t_indexat.FIRST .. t_indexat.LAST
LOOP
    IF t_indexat.EXISTS(i) THEN
        DBMS_OUTPUT.PUT_LINE('Id-urile motocicletelor utilizatorului cu id-ul'||i);
        DBMS_OUTPUT.PUT_LINE('.....');
        FOR inner_rec IN ( SELECT M.ID_MOTOCICLETA, M.KILOMETRAJ
                           FROM MOTOCICLETE M
                           WHERE M.ID_UTILIZATOR = i) LOOP
            IF inner_rec.KILOMETRAJ>v_kilometraj THEN
                t_imbricat.EXTEND;
                t_imbricat(t_imbricat.count) := inner_rec.ID_MOTOCICLETA;
            END IF;
        END LOOP;
    END IF;
END LOOP;
```

```
        END IF;

        DBMS_OUTPUT.PUT_LINE(inner_rec.ID_MOTOCICLETA);

    END LOOP;

    END IF;

END LOOP;

DBMS_OUTPUT.PUT_LINE('Numarul de accesorii pentru fiecare motocicleta care are kilometrajul
mai mare decat valoarea data');

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

FOR i IN 1..(t_imbricat.COUNT-1) LOOP

    SELECT COUNT(AM.ID_ACCESORIU) INTO nr_accesorii
    FROM ACCESORII_MOTOCICLETE AM
    JOIN MOTOCICLETE M ON M.ID_MOTOCICLETA = AM.ID_MOTOCICLETA
    WHERE AM.ID_MOTOCICLETA = t_imbricat(i);

    IF t_imbricat.EXISTS(i) THEN

        v.EXTEND;
        v(i):=nr_accesorii;

        DBMS_OUTPUT.PUT_LINE('Motocicleta cu id'||t_imbricat(i)||' are'||v(i)||' accesorii');

    END IF;

END LOOP;

nr_total_accesorii := 0;
FOR i IN 1..v.COUNT LOOP
    nr_total_accesorii := nr_total_accesorii + v(i);
END LOOP;

nr_accesorii:=v.COUNT;
IF nr_accesorii> 0 THEN
    nr_mediul_accesorii := nr_total_accesorii/nr_accesorii;
ELSE
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
    nr_mediul_accesorii := 0;
```

```
END IF;
```

```
DBMS_OUTPUT.PUT_LINE('Numarul mediul de accesorii este'||nr_mediul_accesorii);
```

```
EXCEPTION
```

```
    WHEN NO_DATA_FOUND THEN
```

```
        RAISE_APPLICATION_ERROR(-20000, 'Nu exista motociclete cu specificatiile cerute');
```

```
    WHEN OTHERS THEN
```

```
        RAISE_APPLICATION_ERROR(-20002, 'Alta eroare!');
```

```
END cerinta6;
```

```
--cerinta7
```

```
PROCEDURE cerinta7 (v_nrmin_moto INT DEFAULT 0)
```

```
IS
```

```
    CURSOR c_moto (v_id_categorie NUMBER) IS
```

```
        SELECT M.NR_INMATRICULARE numar_inmatriculare, (U.NUME||' '||U.PRENUME)  
        nume_proprietar, (M.MARCA||' '||M.MODEL) informatii
```

```
        FROM MOTOCICLETE M
```

```
        JOIN UTILIZATORI U ON U.ID_UTILIZATOR=M.ID_UTILIZATOR
```

```
        WHERE M.ID_CATEGORIE = v_id_categorie;
```

```
verifica INT:=0;
```

```
BEGIN
```

```
    FOR rec_categ IN (
```

```
        SELECT
```

```
            C.ID_CATEGORIE ID, C.NUME NUME, COUNT(M.ID_MOTOCICLETA)  
        NR_MOTOCICLETE
```

```
        FROM MOTOCICLETE M
```

```
        RIGHT JOIN CATEGORII C ON M.ID_CATEGORIE=C.ID_CATEGORIE
```

```
        GROUP BY C.ID_CATEGORIE, C.NUME
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
HAVING COUNT(M.ID_MOTOCICLETA)>v_nrmin_moto) LOOP
```

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

```
IF rec_categ.nr_motociclete=0 THEN
```

```
    DBMS_OUTPUT.PUT_LINE('Categoria'||rec_categ.nume|| nu contine nicio motocicleta');
```

```
ELSIF rec_categ.nr_motociclete=1 THEN
```

```
    DBMS_OUTPUT.PUT_LINE('Categoria'||rec_categ.nume|| contine o singura motocicleta');
```

```
ELSE
```

```
    DBMS_OUTPUT.PUT_LINE('Categoria'||rec_categ.nume|| contine  
'||rec_categ.nr_motociclete|| motociclete');
```

```
END IF;
```

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

```
FOR rec_moto IN c_moto(rec_categ.id) LOOP
```

```
    DBMS_OUTPUT.PUT_LINE('Numar de inmatricularare: '|rec_moto.numar_inmatricularare||  
Proprietar: '|rec_moto.nume_proprietar|| Informatii: '|rec_moto.informatii|);
```

```
END LOOP;
```

```
verifica:=1;
```

```
END LOOP;
```

```
IF verifica=0 THEN
```

```
    DBMS_OUTPUT.PUT_LINE('Nu exista categorii care sa contina mai mult de'||v_nrmin_moto||  
motociclete');
```

```
END IF;
```

```
END cerinta7;
```

--cerinta8

```
FUNCTION cerinta8 (v_nume utilizatori.nume%TYPE DEFAULT 'Georgescu') RETURN NUMBER
```

```
IS
```

```
    nr_motociclete NUMBER;
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
    NO_DATA_FOUND_UTILIZATOR EXCEPTION;  
  
    NO_DATA_FOUND_MOTOCICLETA EXCEPTION;  
  
    contor NUMBER;  
  
BEGIN  
  
    SELECT COUNT(*) INTO contor  
    FROM UTILIZATORI  
    WHERE UPPER(NUME)=UPPER(v_nume);
```

--exceptie pentru niciun utilizator existent cu numele dat

```
IF contor=0 THEN  
  
    RAISE NO_DATA_FOUND_UTILIZATOR;  
  
--exceptie pentru mai multi utilizatori care indeplinesc conditia ceruta  
  
ELSIF contor>1 THEN  
  
    RAISE TOO_MANY_ROWS;  
  
END IF;
```

```
SELECT COUNT(*) INTO contor  
FROM MOTOCICLETE M  
JOIN UTILIZATORI U ON U.ID_UTILIZATOR=M.ID_UTILIZATOR  
WHERE UPPER(U.NUME)=UPPER(v_nume);
```

--exceptie pentru utilizatorii care detin nicio motocicleta in baza de date

```
IF contor=0 THEN  
  
    RAISE NO_DATA_FOUND_MOTOCICLETA;  
  
END IF;
```

```
SELECT  
    COUNT(M.ID_MOTOCICLETA) INTO nr_motociclete  
FROM UTILIZATORI U  
LEFT JOIN MOTOCICLETE M ON U.ID_UTILIZATOR = M.ID_UTILIZATOR
```

```
WHERE M.ID_MOTOCICLETA IS NULL OR M.ID_MOTOCICLETA IN (
    SELECT M1.ID_MOTOCICLETA
    FROM MOTOCICLETE M1
    JOIN MOTOARE MT1 ON MT1.ID_MOTOR=M1.ID_MOTOR
    WHERE UPPER(MT1.CARBURANT) = 'BENZINA')
    GROUP BY U.ID_UTILIZATOR, U.NUME
    HAVING UPPER(U.NUME)=UPPER(v_num);
)

RETURN nr_motociclete;

EXCEPTION
    WHEN NO_DATA_FOUND_UTILIZATOR THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista niciun utilizator in baza de date cu numele dat');
        RETURN -1;
    WHEN NO_DATA_FOUND_MOTOCICLETA THEN
        DBMS_OUTPUT.PUT_LINE('Utilizatorul specificat nu are motociclete salvate in baza de date');
        RETURN -1;
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Utilizatorul specificat nu detine motociclete care sa indeplineasca conditia');
        RETURN -1;
    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE('Există mai mulți utilizatori cu numele dat');
        RETURN -1;
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Cod: ' || SQLCODE);
        DBMS_OUTPUT.PUT_LINE('Mesaj: ' || SQLERRM);
        RETURN -1;
END cerinta8;
```

--cerinta9

PROCEDURE cerinta9 (v\_categorie categorii.nume%TYPE DEFAULT 'Georgescu')

IS

NO\_DATA\_FOUND\_CATEGORIE EXCEPTION;  
NO\_DATA\_FOUND\_INTRETINERE EXCEPTION;  
NO\_DATA\_FOUND\_MOTOCICLETA EXCEPTION;  
TOO\_MANY\_ROWS\_CATEGORIE EXCEPTION;  
contor NUMBER;  
nr\_utilizatori NUMBER;  
nr\_intretineri NUMBER;  
cost\_total NUMBER;  
suma\_medie NUMBER;  
categ\_info categorii.descriere%TYPE;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Numele categoriei: '|v\_categorie);

SELECT

COUNT(\*) INTO contor  
FROM CATEGORII  
WHERE INSTR(UPPER(NUME), UPPER(v\_categorie))>0;

--exceptie pentru cazul in care nu exista categoria data

IF contor=0 THEN

RAISE NO\_DATA\_FOUND\_CATEGORIE;

ELSIF contor>1 THEN

RAISE TOO\_MANY\_ROWS\_CATEGORIE;

END IF;

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

SELECT

```
    DESCRIERE INTO categ_info
    FROM CATEGORII
    WHERE INSTR(UPPER(NUME), UPPER(v_categorie))>0;
    DBMS_OUTPUT.PUT_LINE('Descriere: '|categ_info);
```

SELECT

```
    COUNT(*) INTO contor
    FROM MOTOCICLETE M
    JOIN CATEGORII C ON C.ID_CATEGORIE=M.ID_CATEGORIE
    WHERE UPPER(NUME)=UPPER(v_categorie);
```

--exceptie pentru cazul in care inca nu exista motociclete inregistrate in baza de date care sa faca parte din acea categorie

IF contor=0 THEN

```
    RAISE NO_DATA_FOUND_MOTOCICLETA;
    END IF;
```

SELECT

```
    SUM(P.COST), COUNT(DISTINCT I.ID_INTRETNERE), COUNT(DISTINCT
I.ID_UTILIZATOR)
    INTO cost_total, nr_intretineri, nr_utilizatori
    FROM INTRETNERI I
    JOIN UTILIZATORI U ON U.ID_UTILIZATOR=I.ID_UTILIZATOR
    JOIN PLATI P ON P.ID_PLATA=I.ID_PLATA
    JOIN MOTOCICLETE M ON M.ID_MOTOCICLETA=I.ID_MOTOCICLETA
    JOIN CATEGORII C ON C.ID_CATEGORIE=M.ID_CATEGORIE
    WHERE U.VARSTA<=34 AND U.VARSTA>=25 AND INSTR(UPPER(C.NUME),
UPPER(v_categorie)) > 0;
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

--exceptie pentru cazul in care nu exista intretineri efectuate motocicletelor care sa respecte conditiile cerute

IF nr\_intretineri=0 THEN

RAISE NO\_DATA\_FOUND\_INTRETINERE;

END IF;

suma\_medie:=cost\_total/nr\_intretineri;

DBMS\_OUTPUT.PUT\_LINE('Suma medie alocata intretinerilor pentru aceasta categorie este'||suma\_medie);

DBMS\_OUTPUT.PUT\_LINE('Numarul de participanti la studiu: '||nr\_utilizatori);

EXCEPTION

WHEN NO\_DATA\_FOUND\_CATEGORIE THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista nicio categorie in baza de date cu numele specificat');

WHEN NO\_DATA\_FOUND\_MOTOCICLETA THEN

DBMS\_OUTPUT.PUT\_LINE('Inca nu exista motociclete in baza de date din aceasta categorie');

WHEN NO\_DATA\_FOUND\_INTRETINERE THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista intretineri efectuate pentru motocicletele din categoria specificata');

WHEN TOO\_MANY\_ROWS\_CATEGORIE THEN

DBMS\_OUTPUT.PUT\_LINE('Există mai multe categorii care să conțină acest nume');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Cod: ' || SQLCODE);

DBMS\_OUTPUT.PUT\_LINE('Mesaj: ' || SQLERRM);

END cerinta9;

END cerinta13;

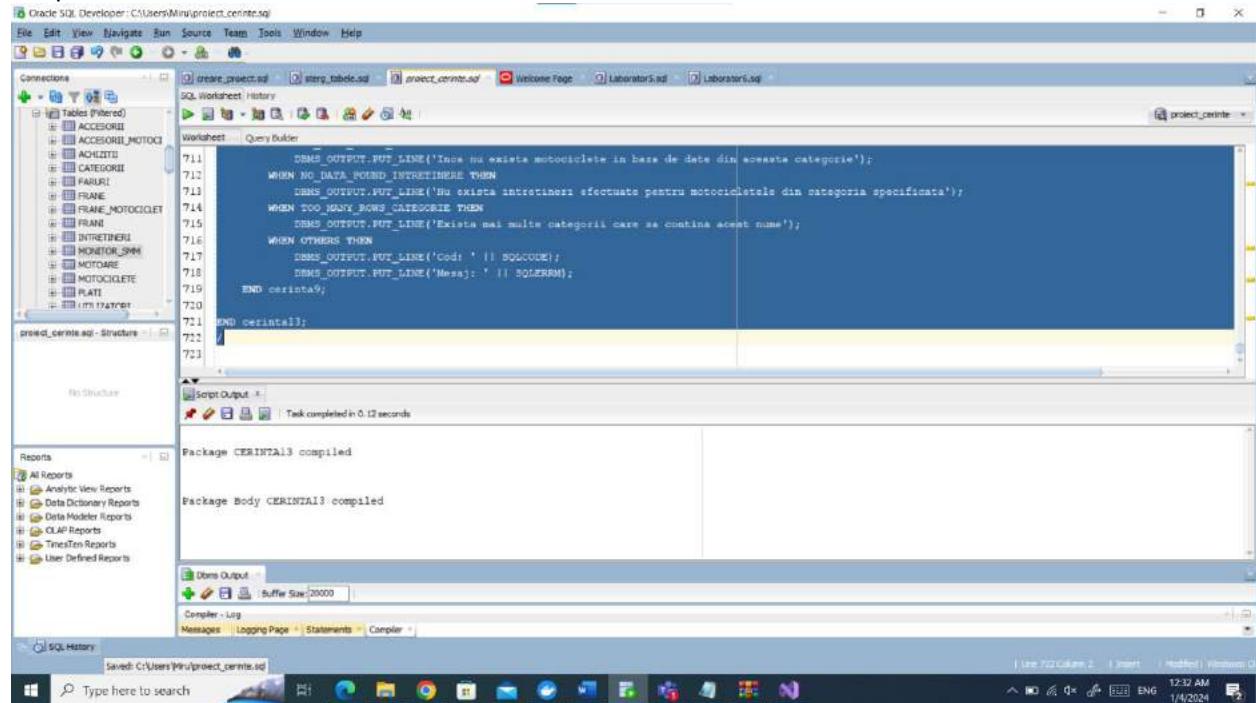
/

➤ Print-Screen

## Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242



### ➤ Testare pachet

```
EXECUTE cerinta13.cerinta6(12000);
```

```
EXECUTE cerinta13.cerinta7(2);
```

```
BEGIN
```

```
    DBMS_OUTPUT.PUT_LINE(cerinta13.cerinta8('Popa'));
```

```
END;
```

```
/
```

```
EXECUTE cerinta13.cerinta9('Naked');
```

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows two instances of Oracle SQL Developer running side-by-side. Both instances have the same database connection and workspace setup.

**Top Window (Left):**

- Connections:** Shows tables like ACCESSORII, ACCESSORII\_MOTOC, ACHIZITII, CATEGORII, FARURI, FRANE, FRANE\_MOTOCYCLET, FRANI, INTRETERII, MONITOR\_SHM, MOTOCYCLET, PLATE, and UNIMATRICE.
- SQL Worksheet - Query Builder:** Contains the following PL/SQL code:

```
722 /
723
724 EXECUTE cerista3.cerista6(12000);
725 EXECUTE cerista3.cerista7(2);
726 BEGIN
727   DBMS_OUTPUT.PUT_LINE(cerista3.cerista8('Popa'));
728 END;
729
730 EXECUTE cerista3.cerista9('Naked');
731
```
- Script Output:** Displays successful execution messages:

```
PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.
```
- Reports:** Shows All Reports, Analytic View Reports, Data Dictionary Reports, Data Model Reports, CLAP Reports, Timesten Reports, and User Defined Reports.
- SQL History:** Shows the saved file path: C:\Users\Minu\project\_cerinte.sql.

**Bottom Window (Right):**

- Connections:** Same as the top window.
- SQL Worksheet - Query Builder:** Contains the following PL/SQL code:

```
project_permitie.x
Numarul maxim de motociclete detinute de un utilizator: 8
Utilizatorii care detin numarul maxim de motociclete:
Foca Mihai
Popovici Diana
Motocicletelor acestor utilizatori:
Id-urile motocicletelor utilizatorului cu id-ul 5
```
- Script Output:** Displays the results of the query:

```
8
9
10
11
12
13
14
15
Id-urile motocicletelor utilizatorului cu id-ul 9
41
42
43
44
6
7
```
- Reports:** Same as the top window.
- SQL History:** Shows the saved file path: C:\Users\Minu\project\_permitie.sql.

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

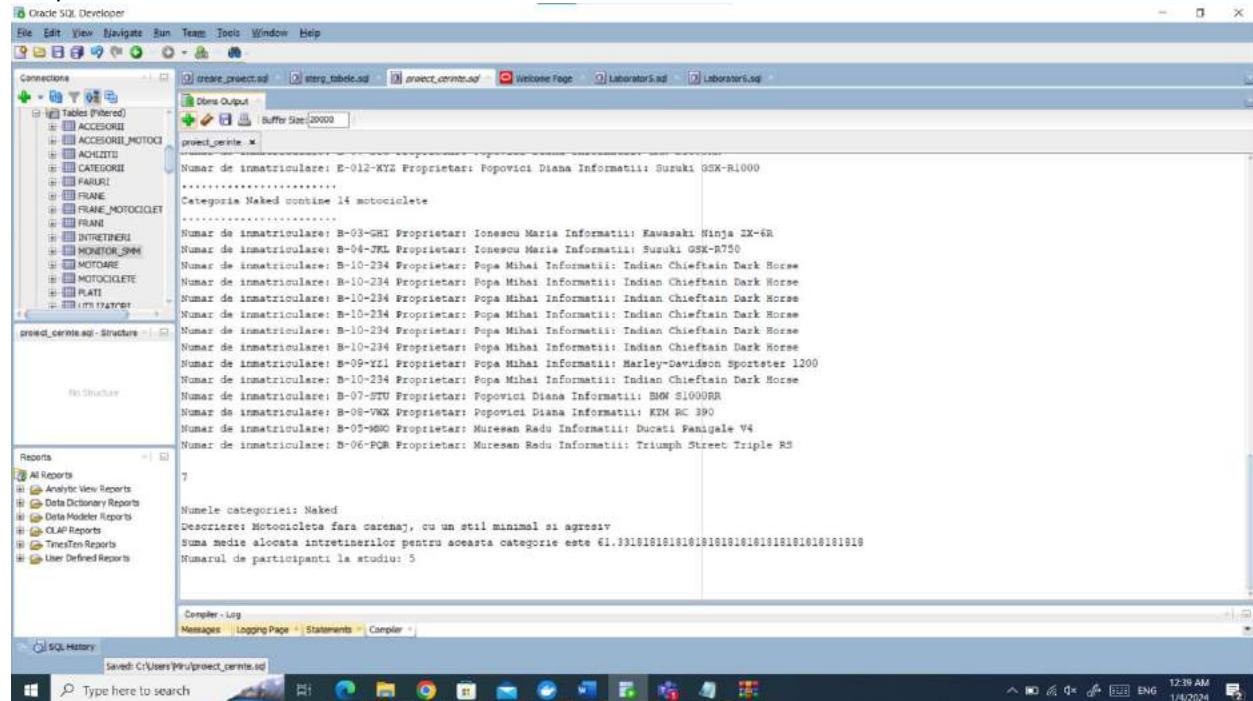
```
Oracle SQL Developer
File Edit View Navigate Run Team Tools Window Help
Connections
Tables (Inherited)
    ACCESSORII
    ACCESSORII_MOTOC
    ACHIZITII
    CATEGORII
    FARURI
    FRAME
    FRAME_MOTOCICLET
    FRAME_MOTOC
    INTRETELERI
    MONITOR_SHM
    MOTOCARE
    MOTOCICLETE
    PLATE
    IMAGINI
project_perinte.x
Open Output Buffer Size:20000
7
20
21
Numarul de accesoriile pentru fiecare motocicleta care are kilometrajul mai mare decat valoarea data
-----
Motocicleta cu id 9 are 1 accesoriu
Motocicleta cu id 10 are 2 accesoriile
Motocicleta cu id 11 are 1 accesoriu
Motocicleta cu id 12 are 4 accesoriile
Motocicleta cu id 13 are 0 accesoriile
Motocicleta cu id 14 are 1 accesoriu
Motocicleta cu id 15 are 2 accesoriile
Motocicleta cu id 7 are 1 accesoriu
Numarul mediu de accesoriile este 2
-----
Categorie Cruiser contine 3 motociclete
-----
Numar de inmatriculari: B-123-KYZ Proprietar: Popovici Diana Informatii: Honda CBR600RR
Numar de inmatriculari: B-02-DEF Proprietar: Muresan Radu Informatii: Yamaha R6
Numar de inmatriculari: B-01-AMC Proprietar: Muresan Radu Informatii: Honda CBR600RR
-----
Categorie Sport contine 3 motociclete
-----
Numar de inmatriculari: B-03-GHI Proprietar: Ionescu Maria Informatii: Kawasaki Ninja ZX-6R
Numar de inmatriculari: B-04-JKL Proprietar: Ionescu Maria Informatii: Suzuki GSX-R750
Numar de inmatriculari: C-156-QWW Proprietar: Popovici Diana Informatii: Yamaha YZF-R1
Compiler - Log
Messages Logging Page Statements Compiler
Saved: C:\Users\Miru\project_perinte.sql
Type here to search
12:38 AM 1/4/2024
```

```
Oracle SQL Developer
File Edit View Navigate Run Team Tools Window Help
Connections
Tables (Inherited)
    ACCESSORII
    ACCESSORII_MOTOC
    ACHIZITII
    CATEGORII
    FARURI
    FRAME
    FRAME_MOTOCICLET
    FRAME_MOTOC
    INTRETELERI
    MONITOR_SHM
    MOTOCARE
    MOTOCICLETE
    PLATE
    IMAGINI
project_perinte.x
Open Output Buffer Size:20000
7
Categorie Touring contine 4 motociclete
-----
Numar de inmatriculari: B-012-KYZ Proprietar: Georgescu Alexandru Informatii: Suzuki GSX-R1000
Numar de inmatriculari: B-08-VWX Proprietar: Popovici Diana Informatii: KTM RC 390
Numar de inmatriculari: B-07-STU Proprietar: Popovici Diana Informatii: BMW S1000RR
Numar de inmatriculari: E-012-KYZ Proprietar: Popovici Diana Informatii: Suzuki GSX-R1000
-----
Categorie Naked contine 14 motociclete
-----
Numar de inmatriculari: B-03-GHI Proprietar: Ionescu Maria Informatii: Kawasaki Ninja ZX-6R
Numar de inmatriculari: B-04-JKL Proprietar: Ionescu Maria Informatii: Suzuki GSX-R750
Numar de inmatriculari: B-10-234 Proprietar: Popa Mihai Informatii: Indian Chieftain Dark Horse
Numar de inmatriculari: B-10-234 Proprietar: Popa Mihai Informatii: Indian Chieftain Dark Horse
Numar de inmatriculari: B-10-234 Proprietar: Popa Mihai Informatii: Indian Chieftain Dark Horse
Numar de inmatriculari: B-10-234 Proprietar: Popa Mihai Informatii: Indian Chieftain Dark Horse
Numar de inmatriculari: B-10-234 Proprietar: Popa Mihai Informatii: Indian Chieftain Dark Horse
Numar de inmatriculari: B-09-YZL Proprietar: Popa Mihai Informatii: Harley-Davidson Sportster 1200
Numar de inmatriculari: B-10-234 Proprietar: Popa Mihai Informatii: Indian Chieftain Dark Horse
Numar de inmatriculari: B-07-STU Proprietar: Popovici Diana Informatii: BMW S1000RR
Numar de inmatriculari: B-08-VWX Proprietar: Popovici Diana Informatii: KTM RC 390
Numar de inmatriculari: B-05-MNO Proprietar: Muresan Radu Informatii: Ducati Panigale V4
Numar de inmatriculari: B-06-PQR Proprietar: Muresan Radu Informatii: Triumph Street Triple RS
7
Compiler - Log
Messages Logging Page Statements Compiler
Saved: C:\Users\Miru\project_perinte.sql
Type here to search
12:38 AM 1/4/2024
```

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242



---

#### 14. Definiți un pachet care să includă tipuri de date complexe și obiecte

necesare unui flux de acțiuni integrate, specifice bazei de date definite

(minim 2 tipuri de date, minim 2 funcții, minim 2 proceduri).

---

Acest pachet, denumit "cerinta14," conține următoarele elemente:

- Tipuri de date:
  - MotocicleteUtilizatorType: Un tabel de tip NUMBER folosit pentru a stoca ID-urile motocicletelor unui utilizator.

- NumeAccesoriiType: Un tabel indexat cu date de tip VARCHAR2(50) folosit pentru a stoca numele accesoriilor.

- Funcții:

- GetByIdMotocicleta: Returnează ID-ul unei motociclete pe baza numărului de înmatriculare.
- GetByIdUtilizator: Returnează ID-ul utilizatorului care este proprietarul unei motociclete al carei id\_motocicleta este dat.
- Get\_UtilizatorInfo: Returnează informațiile despre un utilizator pe baza ID-ului acestuia.
- GetMotocicleteUtilizator: Returnează o listă de ID-uri ale motocicletelor asociate unui utilizator.

- Proceduri:

- afisare\_InfoMotocicleta: Afisează informații despre o motocicletă pe baza ID-ului acesteia.
- plati\_motocicleta: Afisează informații legate de plățile (suma medie/număr elemente/ suma totală pentru întrețineri și accesori) efectuate pentru o motocicletă pe baza ID-ului acesteia.
- Adauga\_AccesoriuMotocicleta: Adaugă un accesoriu la o motocicletă pe baza ID-ului motocicletei și numelui accesoriului.
- Afiseaza\_InfoMotocicleteUtilizator: Afisează informații despre toate motocicletele unui utilizator pe baza ID-ului acestuia.
- actiuni\_motocicleta: Realizează diverse acțiuni legate de o motocicletă pe baza numărului de înmatriculare și a unui cod asociat fiecărei funcționalități. Aceasta procedură include utilizarea tuturor subprogramelor și tipurilor de date definite în pachet, atât prin apelarea lor în mod direct, cât și prin apelarea unui subprogram care le apelează la rândul lui.

➤ *Cod PL/SQL*

```
CREATE OR REPLACE TYPE InfoUtilizator AS OBJECT (
```

```
    id_utilizator INT,  
    nume VARCHAR2(50),  
    prenume VARCHAR2(50),
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

email VARCHAR2(100),

varsta INT,

data\_nasterii DATE

);

/

CREATE OR REPLACE PACKAGE cerinta14 IS

TYPE MotocicleteUtilizatorType IS TABLE OF NUMBER;

TYPE NumeAccesoriiType IS TABLE OF VARCHAR2(50) INDEX BY PLS\_INTEGER;

--returneaza id-ul motocicletei al carui numar de inmatricularare este specificat

FUNCTION GetIdMotocicleta (v\_nr\_inmatricularare motociclete.nr\_inmatricularare%TYPE) RETURN NUMBER;

--returneaza id-ul utilizatorului care detine motocicleta data

FUNCTION GetIdUtilizator (v\_id\_motocicleta motociclete.id\_motocicleta%TYPE) RETURN NUMBER;

--returneaza informatii despre un utilizator al carui id este specificat

FUNCTION Get\_UtilizatorInfo(v\_id\_utilizator utilizatori.id\_utilizator%TYPE) RETURN InfoUtilizator;

--afiseaza informatii despre motocicleta al carui id este specificat

PROCEDURE afisare\_InfoMotocicleta(v\_id\_motocicleta motociclete.id\_motocicleta%TYPE);

--afiseaza detalii despre platile efectuate asupra unei motociclete al carui id este specificat

PROCEDURE plati\_motocicleta(v\_id\_motocicleta motociclete.id\_motocicleta%TYPE);

--atsam un accesoriu la o motocicleta al carui id este dat

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
PROCEDURE Adauga_AccesoriuMotocicleta(v_id_motocicleta IN NUMBER, nume_accesoriu IN
VARCHAR2);
```

--returneaza un tablou imbricat care contine idurile motocicletelor detinute de utilizatorul cu id-ul dat

```
FUNCTION GetMotocicleteUtilizator (v_id_utilizator utilizatori.id_utilizator%TYPE) RETURN
MotocicleteUtilizatorType;
```

--afiseaza informatii despre motocicletele unui utilizator al carui id este dat

```
PROCEDURE Afiseaza_InfoMotocicleteUtilizator (v_id_utilizator utilizatori.id_utilizator%TYPE);
```

--returneaza un tablou indexat care retine denumirile accesoriilor motocicletei al carui id este dat

```
FUNCTION GetAccesoriuMotocicleta(v_id_motocicleta motociclete.id_motocicleta%TYPE)
RETURN NumeAccesoriuType;
```

--o procedura care afiseaza informatii in functie de cerintele utilizatorului

```
PROCEDURE actiuni_motocicleta (v_nr_inmatriculare motociclete.nr_inmatriculare%TYPE, tasta
NUMBER DEFAULT 0);
```

END cerinta14;

/

```
CREATE OR REPLACE PACKAGE BODY cerinta14 IS
```

```
FUNCTION GetIdMotocicleta
```

```
(v_nr_inmatriculare motociclete.nr_inmatriculare%TYPE) RETURN NUMBER
```

AS

```
v_id_motocicleta motociclete.id_motocicleta%TYPE;
```

BEGIN

SELECT

```
ID_MOTOCICLETA INTO v_id_motocicleta
```

```
FROM MOTOCICLETE
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
    WHERE UPPER(NR_INMATRICULARE)=UPPER(v_nr_inmatriculare);

    RETURN v_id_motocicleta;

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        RAISE_APPLICATION_ERROR(-20000,'Nu exista motociclete cu acest numar de inmatriculare');

    WHEN TOO_MANY_ROWS THEN

        RAISE_APPLICATION_ERROR(-20001,'Există mai multe motociclete cu acest numar de inmatriculare');

    WHEN OTHERS THEN

        RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');

END GetIdMotocicleta;
```

FUNCTION GetIdUtilizator

```
(v_id_motocicleta motociclete.id_motocicleta%TYPE) RETURN NUMBER
```

IS

```
    v_id_utilizator utilizatori.id_utilizator%TYPE;

BEGIN

    SELECT

        ID_UTILIZATOR INTO v_id_utilizator

    FROM MOTOCICLETE

    WHERE ID_MOTOCICLETA=v_id_motocicleta;

    RETURN v_id_utilizator;

END GetIdUtilizator;
```

FUNCTION Get\_UtilizatorInfo(v\_id\_utilizator UTILIZATORI.ID\_UTILIZATOR%TYPE) RETURN InfoUtilizator

AS

```
    v_nume VARCHAR2(50);

    v_prenume VARCHAR2(50);
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
v_email VARCHAR2(100);

v_varsta INT;

v_data_nasterii DATE;

v_info_utilizator InfoUtilizator;

BEGIN

    SELECT nume, prenume, email, varsta, data_nasterii

    INTO v_nume, v_prenume, v_email, v_varsta, v_data_nasterii

    FROM UTILIZATORI

    WHERE id_utilizator = v_id_utilizator;

    v_info_utilizator := InfoUtilizator(

        id_utilizator => v_id_utilizator,

        nume => v_nume,

        prenume => v_prenume,

        email => v_email,

        varsta => v_varsta,

        data_nasterii => v_data_nasterii

    );

    RETURN v_info_utilizator;

END Get_UtilizatorInfo;
```

PROCEDURE afisare\_InfoMotocicleta(v\_id\_motocicleta motociclete.id\_motocicleta%TYPE)

IS

```
v_nr_inmatriculare VARCHAR(50);

v_marca VARCHAR(50);

v_model VARCHAR(50);

v_an INT;

v_capacitate_rezervor INT;

v_kilometraj INT;
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

BEGIN

SELECT

```
nr_inmatriculare, marca, model, an, capacitate_rezervor, kilometraj  
INTO v_nr_inmatriculare, v_marca, v_model, v_an, v_capacitate_rezervor, v_kilometraj  
FROM MOTOCICLETE  
WHERE ID_MOTOCICLETA=v_id_motocicleta;
```

```
DBMS_OUTPUT.PUT_LINE('Numar de inmatriculare: '||v_nr_inmatriculare);
```

```
DBMS_OUTPUT.PUT_LINE('Marca: '||v_marca);
```

```
DBMS_OUTPUT.PUT_LINE('Model: '|| v_model);
```

```
DBMS_OUTPUT.PUT_LINE('An de fabricatie: '||v_an);
```

```
DBMS_OUTPUT.PUT_LINE('Capacitatea rezervorului: '||v_capacitate_rezervor);
```

```
DBMS_OUTPUT.PUT_LINE('Kilometraj: '||v_kilometraj);
```

END afisare\_InfoMotocicleta;

PROCEDURE plati\_motocicleta(v\_id\_motocicleta motociclete.id\_motocicleta%TYPE)

IS

```
v_total_intretineri NUMBER:=0;
```

```
v_total_achizitii NUMBER:=0;
```

```
v_medie_intretineri NUMBER:=0;
```

```
v_medie_achizitii NUMBER:=0;
```

```
cnt_intretineri NUMBER:=0;
```

```
cnt_achizitii NUMBER:=0;
```

BEGIN

FOR rec\_intretinere IN (

SELECT

```
P.COST suma
```

```
FROM INTRETINERI I
```

```
JOIN PLATI P ON I.ID_PLATA=P.ID_PLATA
```

```
WHERE I.ID_MOTOCICLETA=v_id_motocicleta) LOOP
```

```
cnt_intretineri:= cnt_intretineri+1;  
v_total_intretineri:= v_total_intretineri+rec_intretinere.suma;  
END LOOP;  
IF cnt_intretineri<>0  
    THEN v_medie_intretineri:=v_total_intretineri/cnt_intretineri;  
END IF;  
  
FOR rec_achizitie IN (  
    SELECT  
        P.COST suma  
    FROM ACHIZITII I  
    JOIN PLATI P ON I.ID_PLATA=P.ID_PLATA  
    WHERE I.ID_MOTOCICLETA=v_id_motocicleta) LOOP  
  
    cnt_achizitii:= cnt_achizitii+1;  
    v_total_achizitii:= v_total_achizitii+rec_achizitie.suma;  
END LOOP;  
IF cnt_achizitii<>0  
    THEN v_medie_achizitii:=v_total_achizitii/cnt_achizitii;  
END IF;  
  
DBMS_OUTPUT.PUT_LINE('Pentru aceasta motocicleta, informatiile platilor efectuate sunt:');  
IF cnt_achizitii=0  
    THEN DBMS_OUTPUT.PUT_LINE('Nu s-au efectuat achizitii pentru aceasta motocicleta');  
ELSE  
    DBMS_OUTPUT.PUT_LINE('- S-au efectuat'||cnt_achizitii||'achizitii cu un costul total de  
'||v_total_achizitii||'lei');  
    DBMS_OUTPUT.PUT_LINE('Valoarea medie a acestora per achizitie este de'||v_medie_achizitii);  
END IF;
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

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

IF cnt_intretineri=0

    THEN DBMS_OUTPUT.PUT_LINE('Nu s-au efectuat intretineri pentru aceasta motocicleta');

ELSE

    DBMS_OUTPUT.PUT_LINE('- S-au efectuat'||cnt_intretineri||'achizitii cu un costul total de
'||v_total_intretineri||'lei');

    DBMS_OUTPUT.PUT_LINE('Valoarea medie a acestora per intretinere este de
'||v_medie_intretineri);

END IF;

END plati_motocicleta;
```

PROCEDURE Adauga\_AccesoriuMotocicleta(

```
    v_id_motocicleta IN NUMBER,
    nume_accesoriu IN VARCHAR2)
```

IS

```
    v_id_accesoriu accesorii.id_accesoriu%TYPE;
    NO_DATA_FOUND_ACESORIU EXCEPTION;
    NO_DATA_FOUND_MOTOCICLETA EXCEPTION;
    TOO_MANY_ROWS_ACESORIU EXCEPTION;
    ACCESORIU_DEJA_ATASAT EXCEPTION;
    contor INT;
```

BEGIN

```
    SELECT
        COUNT(*) INTO contor
    FROM MOTOCICLETE
    WHERE ID_MOTOCICLETA=v_id_motocicleta;
```

```
    IF contor=0 THEN
        RAISE NO_DATA_FOUND_MOTOCICLETA;
    END IF;
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
SELECT  
    COUNT(*) INTO contor  
    FROM ACCESORII  
    WHERE UPPER(NUME)=UPPER(nume_accesoriu);
```

```
IF contor=0 THEN  
    RAISE NO_DATA_FOUND_ACCESORIU;  
ELSIF contor>1 THEN  
    RAISE TOO_MANY_ROWS_ACCESORIU;  
END IF;
```

```
SELECT  
    ID_ACCESORIU INTO v_id_accesoriu  
    FROM ACCESORII  
    WHERE UPPER(NUME)=UPPER(nume_accesoriu);
```

```
SELECT  
    COUNT(*) INTO contor  
    FROM ACCESORII_MOTOCICLETE  
    WHERE ID_ACCESORIU=v_id_accesoriu AND ID_MOTOCICLETA=v_id_motocicleta;
```

```
IF contor=1 THEN  
    RAISE ACCESORIU_DEJA_ATASAT;  
END IF;
```

```
INSERT INTO ACCESORII_MOTOCICLETE (ID_MOTOCICLETA, ID_ACCESORIU)  
VALUES (v_id_motocicleta, v_id_accesoriu);
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

COMMIT;

EXCEPTION

WHEN NO\_DATA\_FOUND\_MOTOCICLETA THEN

DBMS\_OUTPUT.PUT\_LINE('Motocicleta cu id-ul specificat nu exista in baza de date');

WHEN NO\_DATA\_FOUND\_ACCESORIU THEN

DBMS\_OUTPUT.PUT\_LINE('Accesorul cu numele specificat nu exista in baza de date! Mai intai adaugati-l.');

WHEN TOO\_MANY\_ROWS\_ACCESORIU THEN

DBMS\_OUTPUT.PUT\_LINE('Exista mai multe accesorii cu numele specificat in baza de date');

WHEN ACCESORIU\_DEJA\_ATASAT THEN

DBMS\_OUTPUT.PUT\_LINE('Accesorul cu numele specificat este deja atasat la aceasta motocicleta');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Cod: ' || SQLCODE);

DBMS\_OUTPUT.PUT\_LINE('Mesaj: ' || SQLERRM);

END Adauga\_AccesoriuMotocicleta;

FUNCTION GetMotocicleteUtilizator

(v\_id\_utilizator UTILIZATORI.ID\_UTILIZATOR%TYPE) RETURN MotocicleteUtilizatorType

IS

v\_id\_motocicleta MOTOCICLETE.ID\_MOTOCICLETA%TYPE;

v\_motociclete\_utilizator MotocicleteUtilizatorType;

NO\_DATA\_FOUND\_UTILIZATOR EXCEPTION;

NO\_DATA\_FOUND\_MOTOCICLETA EXCEPTION;

contor INT;

BEGIN

SELECT COUNT(\*)

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

INTO contor

FROM UTILIZATORI

WHERE ID\_UTILIZATOR = v\_id\_utilizator;

IF contor = 0 THEN

RAISE NO\_DATA\_FOUND\_UTILIZATOR;

END IF;

SELECT COUNT(\*)

INTO contor

FROM MOTOCICLETE

WHERE ID\_UTILIZATOR = v\_id\_utilizator;

IF contor = 0 THEN

RAISE NO\_DATA\_FOUND\_MOTOCICLETA;

END IF;

v\_motociclete\_utilizator := MotocicleteUtilizatorType();

FOR motocicleta\_rec IN (SELECT ID\_MOTOCICLETA

FROM MOTOCICLETE

WHERE ID\_UTILIZATOR = v\_id\_utilizator) LOOP

v\_id\_motocicleta := motocicleta\_rec.ID\_MOTOCICLETA;

v\_motociclete\_utilizator.EXTEND;

v\_motociclete\_utilizator(v\_motociclete\_utilizator.LAST) := v\_id\_motocicleta;

END LOOP;

RETURN v\_motociclete\_utilizator;

EXCEPTION

WHEN NO\_DATA\_FOUND\_UTILIZATOR THEN

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
    DBMS_OUTPUT.PUT_LINE('Utilizatorul cu id-ul specificat nu exist?!');

    RETURN MotocicleteUtilizatorType();

WHEN NO_DATA_FOUND_MOTOCICLETA THEN

    DBMS_OUTPUT.PUT_LINE('Utilizatorul cu id-ul specificat nu de?ine motociclete!');

    RETURN MotocicleteUtilizatorType();

WHEN OTHERS THEN

    DBMS_OUTPUT.PUT_LINE('Cod: ' || SQLCODE);

    DBMS_OUTPUT.PUT_LINE('Mesaj: ' || SQLERRM);

    RETURN MotocicleteUtilizatorType();

END GetMotocicleteUtilizator;
```

PROCEDURE Afiseaza\_InfoMotocicleteUtilizator

(v\_id\_utilizator UTILIZATORI.ID\_UTILIZATOR%TYPE)

AS

```
    info_utilizator InfoUtilizator;
    v_motociclete_utilizator MotocicleteUtilizatorType;
```

BEGIN

v\_motociclete\_utilizator := GetMotocicleteUtilizator(v\_id\_utilizator);

info\_utilizator:=Get\_UtilizatorInfo(v\_id\_utilizator);

IF v\_motociclete\_utilizator.COUNT=0 THEN

DBMS\_OUTPUT.PUT\_LINE('Utilizatorul'||info\_utilizator.nume||' nu detine nicio motocicleta');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Utilizatorul'||info\_utilizator.nume||' '||info\_utilizator.prenume||' detine'||v\_motociclete\_utilizator.COUNT||' motociclete:');

DBMS\_OUTPUT.PUT\_LINE('.....');

FOR i IN 1..v\_motociclete\_utilizator.COUNT LOOP

afisare\_InfoMotocicleta(v\_motociclete\_utilizator(i));

DBMS\_OUTPUT.PUT\_LINE('.....');

END LOOP;

END IF;

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

END Afiseaza\_InfoMotocicleteUtilizator;

FUNCTION GetAccesoriiMotocicleta

(v\_id\_motocicleta MOTOCICLETE.ID\_MOTOCICLETA%TYPE) RETURN NumeAccesoriiType

IS

v\_accesorii NumeAccesoriiType:=NumeAccesoriiType();

contor INT;

NO\_DATA\_FOUND\_MOTOCICLETA EXCEPTION;

BEGIN

SELECT

COUNT(\*) INTO contor

FROM MOTOCICLETE

WHERE ID\_MOTOCICLETA=v\_id\_motocicleta;

IF CONTOR=0 THEN

RAISE NO\_DATA\_FOUND\_MOTOCICLETA;

END IF;

FOR accesoriu\_rec IN (SELECT A.NUME

FROM ACCESORII\_MOTOCICLETE AM

JOIN ACCESORII A ON AM.ID\_ACCESORIU = A.ID\_ACCESORIU

WHERE AM.ID\_MOTOCICLETA = v\_id\_motocicleta) LOOP

v\_accesorii(v\_accesorii.COUNT + 1) := accesoriu\_rec.nume;

END LOOP;

RETURN v\_accesorii;

EXCEPTION

WHEN NO\_DATA\_FOUND\_MOTOCICLETA THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista motocicleta in baza de date cu id-ul specificat!');

RETURN NumeAccesoriiType();

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Cod: ' || SQLCODE);
    DBMS_OUTPUT.PUT_LINE('Mesaj: ' || SQLERRM);
    RETURN NumeAccesoriiType();
END GetAccesoriiMotocicleta;

PROCEDURE actiuni_motocicleta (v_nr_inmatriculare motociclete.nr_inmatriculare%TYPE, tasta
NUMBER DEFAULT 0)
IS
    v_id_motocicleta motociclete.id_motocicleta%TYPE;
    info_utilizator InfoUtilizator;
    v_accesorii NumeAccesoriiType;
BEGIN
    v_id_motocicleta:=GetIdMotocicleta(v_nr_inmatriculare);
    DBMS_OUTPUT.PUT_LINE('Ati introdus tasta'||tasta);
    IF tasta=0 THEN
        DBMS_OUTPUT.PUT_LINE('Actiune anulata cu succes!');
    ELSIF tasta=1 THEN
        --afisarea detaliilor proprietarului motocicletei
        DBMS_OUTPUT.PUT_LINE('Actiune realizata cu succes!');
        DBMS_OUTPUT.PUT_LINE('Informatiile proprietarului motocicletei cu numarul de inmatriculare
specificat:');
        DBMS_OUTPUT.PUT_LINE('.....');
        info_utilizator:=Get_UtilizatorInfo(GetIdUtilizator(v_id_motocicleta));
        DBMS_OUTPUT.PUT_LINE('Nume: '||info_utilizator.nume);
        DBMS_OUTPUT.PUT_LINE('Prenume: '||info_utilizator.prenume);
        DBMS_OUTPUT.PUT_LINE('Email: '||info_utilizator.email);
        DBMS_OUTPUT.PUT_LINE('Varsta: '||info_utilizator.varsta);
        DBMS_OUTPUT.PUT_LINE('Data nasterii: '||info_utilizator.data_nasterii);
    ELSIF tasta=2 THEN
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
--afisarea detaliilor platilor efectuate pentru o anumita motocicleta  
DBMS_OUTPUT.PUT_LINE('Actiune realizata cu succes!');  
  
DBMS_OUTPUT.PUT_LINE('Afisarea detaliilor platilor efectuate pentru motocicleta cu numarul  
de inmatriculara specificat');  
  
DBMS_OUTPUT.PUT_LINE('.....');  
plati_motocicleta(v_id_motocicleta);  
  
ELSIF tasta=3 THEN  
  
--adaugarea unui anumit accesoriu pentru o motocicleta  
DBMS_OUTPUT.PUT_LINE('Actiune realizata cu succes!');  
  
DBMS_OUTPUT.PUT_LINE('Atasarea unui accesoriu la motocicleta cu numarul de inmatriculara  
specificat');  
  
DBMS_OUTPUT.PUT_LINE('.....');  
Adauga_AccesoriuMotocicleta(v_id_motocicleta, 'Casca');  
  
ELSIF tasta=4 THEN  
  
--afisarea motocicletelor proprietarului  
DBMS_OUTPUT.PUT_LINE('Actiune realizata cu succes!');  
  
DBMS_OUTPUT.PUT_LINE('Afisarea motocicletelor proprietarului care detine motocicleta cu  
numarul de inmatriculara specificat');  
  
DBMS_OUTPUT.PUT_LINE('.....');  
Afiseaza_InfoMotocicleteUtilizator(GetIdUtilizator(v_id_motocicleta));  
  
ELSIF tasta=5 THEN  
  
--afisarea denumirilor accesoriile atasate la aceasta motocicleta  
DBMS_OUTPUT.PUT_LINE('Actiune realizata cu succes!');  
  
DBMS_OUTPUT.PUT_LINE('Afisarea numelor accesoriilor atasate la motocicleta cu numarul de  
inmatriculara specificat');  
  
DBMS_OUTPUT.PUT_LINE('.....');  
v_accesorii:=GetAccesoriuMotocicleta(v_id_motocicleta);  
  
IF v_accesorii.COUNT = 0 THEN  
  
DBMS_OUTPUT.PUT_LINE('Nu exista accesoriu atasate la aceasta motocicleta');  
  
ELSE  
  
FOR i IN 1..v_accesorii.COUNT LOOP
```

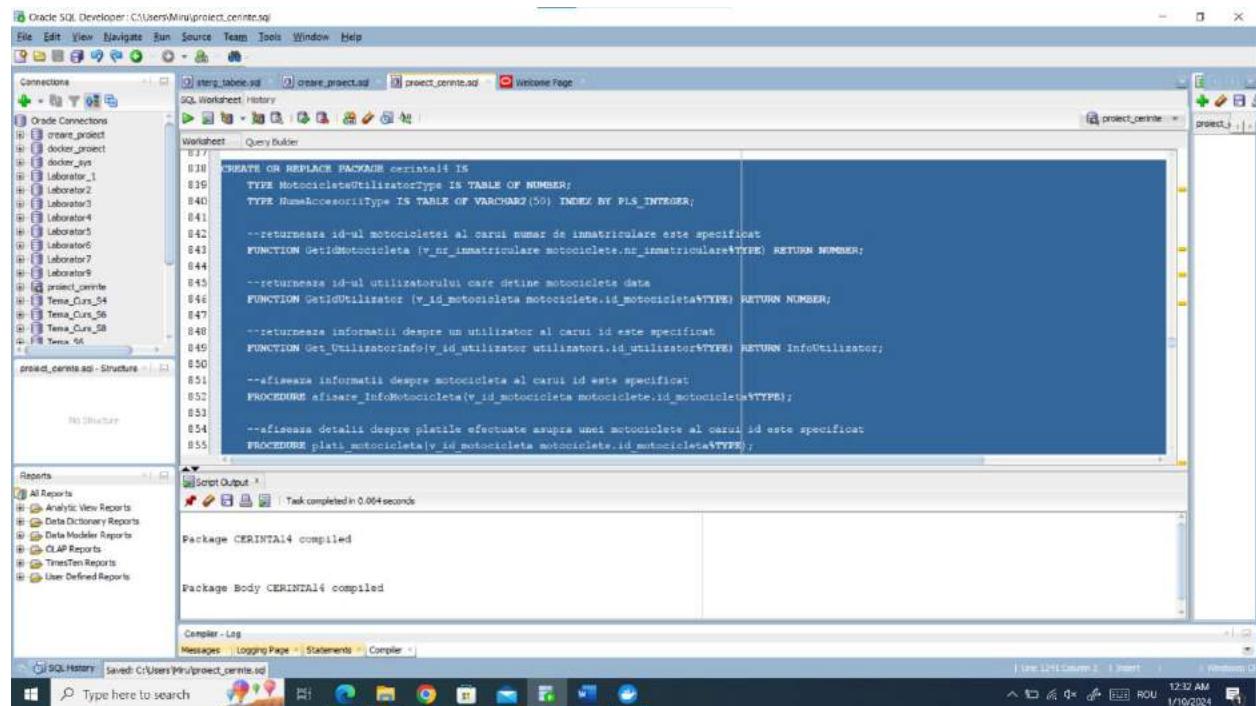
# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
        DBMS_OUTPUT.PUT_LINE(v_accesorii(i));  
    END LOOP;  
  
    END IF;  
  
    ELSE  
  
        DBMS_OUTPUT.PUT_LINE('Nu exista nicio actiune pentru tasta introdusa!');  
  
    END IF;  
  
END actiuni_motocicleta;  
  
END cerinta14;  
/  
/
```

## ➤ Print-Screen



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

- Connections:** Shows various database connections including 'create\_project'.
- SQL Worksheet:** Displays the PL/SQL code for package CERINTA14. The code includes declarations for cursor types (MotocicleteUtilizatorType, NumarImmatriculareType), functions (GetIdMotocicleta, GetIdUtilizator, GetUtilizatorInfo), and procedures (afiseaza\_infoMotocicleta, afiseaza\_detalii\_plastile).
- Script Output:** Shows the compilation results:
  - Package CERINTA14 compiled
  - Package Body CERINTA14 compiled
- Compiler - Log:** Shows the log message: "Task completed in 0.004 seconds".
- Bottom Status Bar:** Shows the path "Saved: C:\Users\Minu\project\_cerinta.ad", the date "1/10/2024", and the time "12:32 AM".

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

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

- Connections:** Shows various database connections including 'create\_project', 'docker\_project', 'docker\_pys', 'Laborator\_1', 'Laborator\_2', 'Laborator\_3', 'Laborator\_4', 'Laborator\_5', 'Laborator\_6', 'Laborator\_7', 'Laborator\_8', 'Laborator\_9', 'project\_center', 'Tema\_Curs\_54', 'Tema\_Curs\_56', 'Tema\_Curs\_58', and 'Tema\_64'.
- SQL Worksheet:** Displays the creation of a package named CERINTA14. The code includes several procedures and functions such as `GetMotocicletaUtilizator`, `Afiseaza_InfoMotocicleteUtilizator`, `GetAccesoriiMotocicleta`, and `actiuni_motocicleta`.
- Script Output:** Shows the message "Task completed in 0.064 seconds".
- Compiler - Log:** Shows the compilation status: "Package CERINTA14 compiled" and "Package Body CERINTA14 compiled".
- Bottom Status Bar:** Shows the path "C:\Users\Minu\project\_center.sql", the search bar "Type here to search", and the system status "12:33 AM ROU 1/16/2024".

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

- Connections:** Shows various database connections including 'create\_project', 'docker\_project', 'docker\_pys', 'Laborator\_1', 'Laborator\_2', 'Laborator\_3', 'Laborator\_4', 'Laborator\_5', 'Laborator\_6', 'Laborator\_7', 'Laborator\_8', 'Laborator\_9', 'project\_center', 'Tema\_Curs\_54', 'Tema\_Curs\_56', 'Tema\_Curs\_58', and 'Tema\_64'.
- SQL Worksheet:** Displays the body of the package CERINTA14, specifically the `GetIdMotocicleta` function which retrieves the ID of a motorcycle given its registration number.
- Script Output:** Shows the message "Task completed in 0.064 seconds".
- Compiler - Log:** Shows the compilation status: "Package CERINTA14 compiled" and "Package Body CERINTA14 compiled".
- Bottom Status Bar:** Shows the path "C:\Users\Minu\project\_center.sql", the search bar "Type here to search", and the system status "12:33 AM ROU 1/16/2024".

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. The workspace contains several tabs: 'sterig\_tablee.sql', 'create\_project.ad', 'project\_permit.ad', and 'Welcome Page'. The central area is a 'Worksheet' tab titled 'Query Builder' with the following PL/SQL code:

```
891    WHEN OTHERS THEN
892      RAISE_APPLICATION_ERROR(-20002,'Alte eroare!');
893  END GetIdMotocicleta;
894
895  FUNCTION GetIdUtilizator
896    (v_id_motocicleta NUMBER) RETURN NUMBER
897  IS
898    v_id_utilizator utilizzatori.id_utilizator%TYPE;
899
900    BEGIN
901      SELECT
902        id_utilizator INTO v_id_utilizator
903      FROM MOTOCICLETE
904      WHERE id_motocicleta=v_id_motocicleta;
905      RETURN v_id_utilizator;
906    END GetIdUtilizator;
907
908  FUNCTION Get_UtilizatorInfo(v_id_utilizator UTILIZATOR.ID_UTILIZATOR%TYPE) RETURN InfoUtilizator
909  AS
```

The 'Script Output' panel shows the message: 'Task completed in 0.064 seconds'. Below it, the compiler log indicates that the 'Package CERINTA14 compiled' and 'Package Body CERINTA14 compiled'. The status bar at the bottom right shows the date and time: '1/16/2024 12:33 AM'.

This screenshot shows the Oracle SQL Developer interface again. The workspace tabs are the same: 'sterig\_tablee.sql', 'create\_project.ad', 'project\_permit.ad', and 'Welcome Page'. The central 'Worksheet' tab contains the following PL/SQL code:

```
909  FUNCTION Get_UtilizatorInfo(v_id_utilizator UTILIZATOR.ID_UTILIZATOR%TYPE) RETURN InfoUtilizator
910  AS
911    v_name VARCHAR2(50);
912    v_prenume VARCHAR2(50);
913    v_email VARCHAR2(100);
914    v_varsta INT;
915    v_data_nasterii DATE;
916    v_info_utilizator InfoUtilizator;
917
918  BEGIN
919    SELECT name, prenume, email, varsta, data_nasterii
920    INTO v_name, v_prenume, v_email, v_varsta, v_data_nasterii
921    FROM UTILIZATORI
922    WHERE id_utilizator = v_id_utilizator;
923
924    v_info_utilizator := InfoUtilizator(
925      id_utilizator => v_id_utilizator,
926      nume => v_name,
927      prenume => v_prenume,
928      email => v_email,
```

The 'Script Output' panel shows the message: 'Task completed in 0.064 seconds'. The compiler log indicates that the 'Package CERINTA14 compiled' and 'Package Body CERINTA14 compiled'. The status bar at the bottom right shows the date and time: '1/16/2024 12:33 AM'.

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface with the 'afisare\_InfoMotocicleta' procedure in the central workspace. The code defines a package body CERINTA14 with a procedure that takes an ID as input and prints details about the motorcycle. The output is directed to DBMS\_OUTPUT.PUT\_LINE statements.

```
928     data_variate => v_variate;
929     data_msterii => v_data_msterii
930   );
931
932   RETURN v_info_utilizator;
933 END Get_UtilizatorInfo;
934
935 PROCEDURE afisare_InfoMotocicleta(v_id_motocicleta motociclete.id_motocicleta%TYPE)
936 IS
937   v_nr_inmatriculare VARCHAR(50);
938   v_marca VARCHAR(50);
939   v_model VARCHAR(50);
940   v_an INT;
941   v_capacitate_reservor INT;
942   v_kilometraj INT;
943
944 BEGIN
945   SELECT
946     nr_inmatriculare, marca, model, an, capacitate_reservor, kilometraj
947   INTO v_nr_inmatriculare, v_marca, v_model, v_an, v_capacitate_reservor, v_kilometraj
948   FROM MOTOCICLETA
949   WHERE ID_MOTOCICLETA=v_id_motocicleta;
950
951   DBMS_OUTPUT.PUT_LINE('Numar de inmatriculare: '||v_nr_inmatriculare);
952   DBMS_OUTPUT.PUT_LINE('Marca: '||v_marca);
953   DBMS_OUTPUT.PUT_LINE('Model: '||v_model);
954   DBMS_OUTPUT.PUT_LINE('An de fabricatie: '||v_an);
955   DBMS_OUTPUT.PUT_LINE('Capacitate rezervorului: '||v_capacitate_reservor);
956   DBMS_OUTPUT.PUT_LINE('Kilometraj: '||v_kilometraj);
957
958 END afisare_InfoMotocicleta;
959
960 PROCEDURE plati_motocicleta(v_id_motocicleta motociclete.id_motocicleta%TYPE)
961 IS
962   v_total_intretineri NUMBER:=0;
963   v_total_achizitii NUMBER:=0;
964   v_media_intretineri NUMBER:=0;
965   v_media_achizitii NUMBER:=0;
966   cont_intretineri NUMBER:=0;
967   cont_achizitii NUMBER:=0;
968
969 BEGIN
970   FOR rec_intretinere IN (
971     SELECT
972       F.COST suma
973     FROM INTRETINERE I
974     JOIN MOTOCICLETA M ON I.ID_MOTOCICLETA = M.ID_MOTOCICLETA
975     WHERE M.ID_MOTOCICLETA = v_id_motocicleta
976   ) LOOP
977     cont_intretineri := cont_intretineri + 1;
978     cont_achizitii := cont_achizitii + rec_intretinere.suma;
979   END LOOP;
980   v_total_intretineri := cont_intretineri;
981   v_total_achizitii := cont_achizitii;
982   v_media_intretineri := v_total_intretineri / cont_intretineri;
983   v_media_achizitii := v_total_achizitii / cont_achizitii;
984
985   DBMS_OUTPUT.PUT_LINE('Total intretineri: '||v_total_intretineri);
986   DBMS_OUTPUT.PUT_LINE('Total achizitii: '||v_total_achizitii);
987   DBMS_OUTPUT.PUT_LINE('Media intretineri: '||v_media_intretineri);
988   DBMS_OUTPUT.PUT_LINE('Media achizitii: '||v_media_achizitii);
989
990   DBMS_OUTPUT.PUT_LINE('Plata totala este: '||cont_achizitii);
991
992 END plati_motocicleta;
```

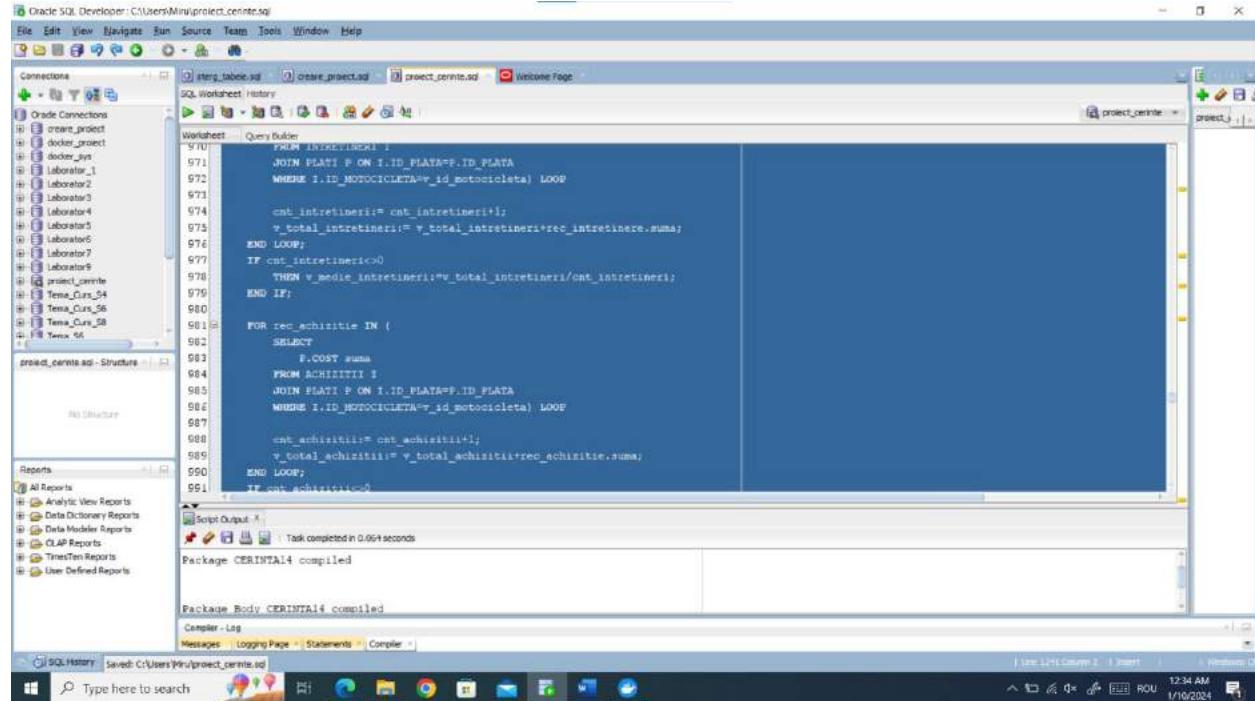
The screenshot shows the Oracle SQL Developer interface with the 'plati\_motocicleta' procedure in the central workspace. The code defines a package body CERINTA14 with a procedure that calculates totals and averages for maintenance and purchases of a motorcycle based on its ID. The output is directed to DBMS\_OUTPUT.PUT\_LINE statements.

```
949     data_variate => v_variate;
950     data_msterii => v_data_msterii
951   );
952
953   RETURN v_info_utilizator;
954 END Get_UtilizatorInfo;
955
956 PROCEDURE afisare_InfoMotocicleta(v_id_motocicleta motociclete.id_motocicleta%TYPE)
957 IS
958   v_nr_inmatriculare VARCHAR(50);
959   v_marca VARCHAR(50);
960   v_model VARCHAR(50);
961   v_an INT;
962   v_capacitate_reservor INT;
963   v_kilometraj INT;
964
965 BEGIN
966   SELECT
967     nr_inmatriculare, marca, model, an, capacitate_reservor, kilometraj
968   INTO v_nr_inmatriculare, v_marca, v_model, v_an, v_capacitate_reservor, v_kilometraj
969   FROM MOTOCICLETA
970   WHERE ID_MOTOCICLETA=v_id_motocicleta;
971
972   DBMS_OUTPUT.PUT_LINE('Numar de inmatriculare: '||v_nr_inmatriculare);
973   DBMS_OUTPUT.PUT_LINE('Marca: '||v_marca);
974   DBMS_OUTPUT.PUT_LINE('Model: '||v_model);
975   DBMS_OUTPUT.PUT_LINE('An de fabricatie: '||v_an);
976   DBMS_OUTPUT.PUT_LINE('Capacitate rezervorului: '||v_capacitate_reservor);
977   DBMS_OUTPUT.PUT_LINE('Kilometraj: '||v_kilometraj);
978
979 END afisare_InfoMotocicleta;
980
981 PROCEDURE plati_motocicleta(v_id_motocicleta motociclete.id_motocicleta%TYPE)
982 IS
983   v_total_intretineri NUMBER:=0;
984   v_total_achizitii NUMBER:=0;
985   v_media_intretineri NUMBER:=0;
986   v_media_achizitii NUMBER:=0;
987   cont_intretineri NUMBER:=0;
988   cont_achizitii NUMBER:=0;
989
990 BEGIN
991   FOR rec_intretinere IN (
992     SELECT
993       F.COST suma
994     FROM INTRETINERE I
995     JOIN MOTOCICLETA M ON I.ID_MOTOCICLETA = M.ID_MOTOCICLETA
996     WHERE M.ID_MOTOCICLETA = v_id_motocicleta
997   ) LOOP
998     cont_intretineri := cont_intretineri + 1;
999     cont_achizitii := cont_achizitii + rec_intretinere.suma;
1000   END LOOP;
1001   v_total_intretineri := cont_intretineri;
1002   v_total_achizitii := cont_achizitii;
1003   v_media_intretineri := v_total_intretineri / cont_intretineri;
1004   v_media_achizitii := v_total_achizitii / cont_achizitii;
1005
1006   DBMS_OUTPUT.PUT_LINE('Total intretineri: '||v_total_intretineri);
1007   DBMS_OUTPUT.PUT_LINE('Total achizitii: '||v_total_achizitii);
1008   DBMS_OUTPUT.PUT_LINE('Media intretineri: '||v_media_intretineri);
1009   DBMS_OUTPUT.PUT_LINE('Media achizitii: '||v_media_achizitii);
1010
1011   DBMS_OUTPUT.PUT_LINE('Plata totala este: '||cont_achizitii);
1012
1013 END plati_motocicleta;
```

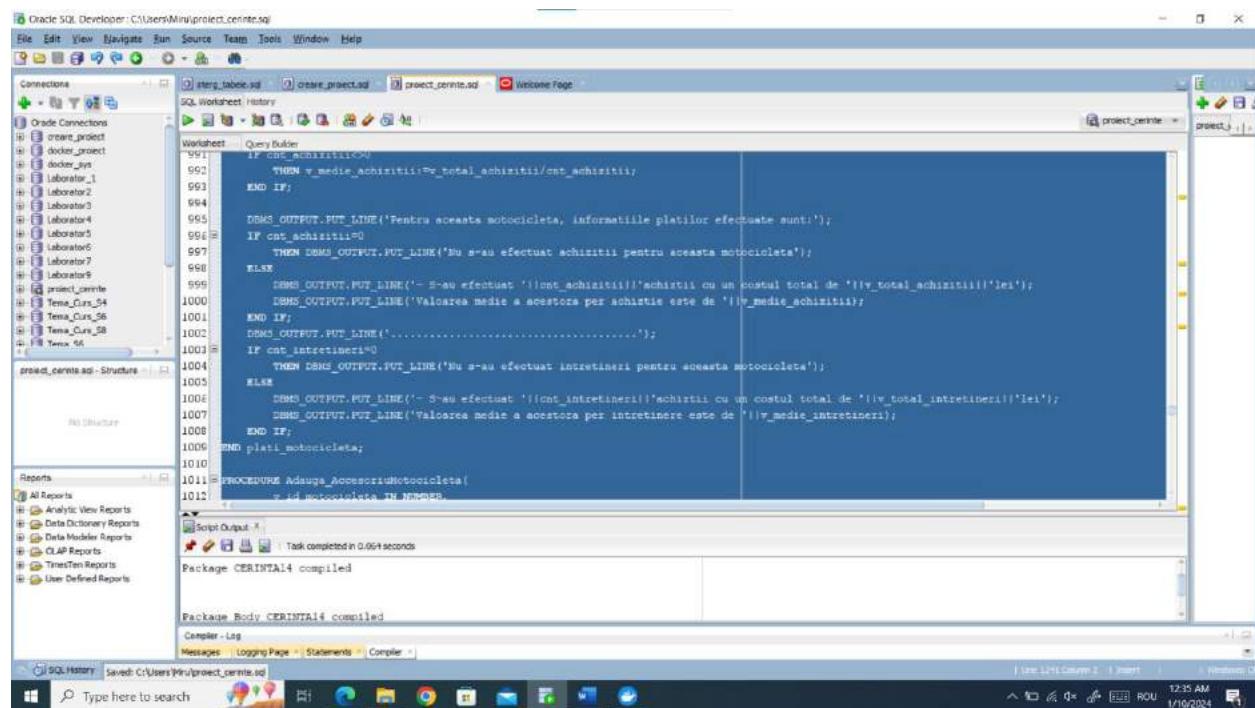
# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242



```
910    FORX INTRETNIREI
911    LOOP
912        JOIN PLATI P ON I.ID_PLATA=P.ID_PLATA
913        WHERE I.ID_MOTOCICLETA=v_id_motocicleta) LOOP
914
915        cnt_intretineri:=cnt_intretineri+1;
916        v_total_intretineri:=v_total_intretineri+rec_intretinere.suma;
917    END LOOP;
918    IF cnt_intretineri>0
919    THEN v_medie_intretineri:=v_total_intretineri/cnt_intretineri;
920    END IF;
921
922    FOR rec_achizitie IN (
923        SELECT
924            E.COST suma
925            FROM ACHIZITII E
926            JOIN PLATI P ON I.ID_PLATA=E.ID_PLATA
927            WHERE I.ID_MOTOCICLETA=v_id_motocicleta) LOOP
928
929        cnt_achizitii:=cnt_achizitii+1;
930        v_total_achizitii:=v_total_achizitii+rec_achizitie.suma;
931    END LOOP;
932    IF cnt_achizitii>0
933    THEN
934        v_medie_achizitii:=v_total_achizitii/cnt_achizitii;
935    END IF;
936
937    FORX ACHIZITIE
938    LOOP
939        E.COST suma
940        FROM ACHIZITII E
941        JOIN PLATI P ON I.ID_PLATA=E.ID_PLATA
942        WHERE I.ID_MOTOCICLETA=v_id_motocicleta) LOOP
943
944        cnt_achizitii:=cnt_achizitii+1;
945        v_total_achizitii:=v_total_achizitii+rec_achizitie.suma;
946    END LOOP;
947    IF cnt_achizitii>0
948    THEN
949        v_medie_achizitii:=v_total_achizitii/cnt_achizitii;
950    END IF;
951
952    FORX ACHIZITIE
953    LOOP
954        E.COST suma
955        FROM ACHIZITII E
956        JOIN PLATI P ON I.ID_PLATA=E.ID_PLATA
957        WHERE I.ID_MOTOCICLETA=v_id_motocicleta) LOOP
958
959        cnt_achizitii:=cnt_achizitii+1;
960        v_total_achizitii:=v_total_achizitii+rec_achizitie.suma;
961    END LOOP;
962    IF cnt_achizitii>0
963    THEN
964        v_medie_achizitii:=v_total_achizitii/cnt_achizitii;
965    END IF;
966
967    DBMS_OUTPUT.PUT_LINE('Pentru aceasta motocicleta, informatiile platilor efectuate sunt:');
968    IF cnt_achizitii=0
969    THEN DBMS_OUTPUT.PUT_LINE('Nu s-au efectuat achizitii pentru aceasta motocicleta');
970    ELSE
971        DBMS_OUTPUT.PUT_LINE(' - S-au efectuat '||cnt_achizitii||' achizitii cu un costul total de '||v_total_achizitii||'lei');
972        DBMS_OUTPUT.PUT_LINE('Valoarea media a acestora per achizitie este de '||v_medie_achizitii);
973    END IF;
974
975    DBMS_OUTPUT.PUT_LINE('.');
976
977    IF cnt_intretineri=0
978    THEN DBMS_OUTPUT.PUT_LINE('Nu s-au efectuat intretineri pentru aceasta motocicleta');
979    ELSE
980        DBMS_OUTPUT.PUT_LINE(' - S-au efectuat '||cnt_intretineri||' intretineri cu un costul total de '||v_total_intretineri||'lei');
981        DBMS_OUTPUT.PUT_LINE('Valoarea media a acestora per intretinere este de '||v_medie_intretineri);
982    END IF;
983
984    END plati_motocicleta;
985
986    PROCEDURE Adauga_AccesoriiMotocicleta
987        v_id_motocicleta IN NUMBER;
```



```
991    IF cnt_achizitii<0
992    THEN v_medie_achizitii:=v_total_achizitii/cnt_achizitii;
993    END IF;
994
995    DBMS_OUTPUT.PUT_LINE('Pentru aceasta motocicleta, informatiile platilor efectuate sunt:');
996    IF cnt_achizitii=0
997    THEN DBMS_OUTPUT.PUT_LINE('Nu s-au efectuat achizitii pentru aceasta motocicleta');
998    ELSE
999        DBMS_OUTPUT.PUT_LINE(' - S-au efectuat '||cnt_achizitii||' achizitii cu un costul total de '||v_total_achizitii||'lei');
1000        DBMS_OUTPUT.PUT_LINE('Valoarea media a acestora per achizitie este de '||v_medie_achizitii);
1001    END IF;
1002
1003    DBMS_OUTPUT.PUT_LINE('.');
1004
1005    IF cnt_intretineri=0
1006    THEN DBMS_OUTPUT.PUT_LINE('Nu s-au efectuat intretineri pentru aceasta motocicleta');
1007    ELSE
1008        DBMS_OUTPUT.PUT_LINE(' - S-au efectuat '||cnt_intretineri||' intretineri cu un costul total de '||v_total_intretineri||'lei');
1009        DBMS_OUTPUT.PUT_LINE('Valoarea media a acestora per intretinere este de '||v_medie_intretineri);
1010    END IF;
1011
1012    END plati_motocicleta;
1013
1014    PROCEDURE Adauga_AccesoriiMotocicleta
1015        v_id_motocicleta IN NUMBER;
```

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. The title bar says "Oracle SQL Developer : C:\Users\Minu\project\_center.sql". The main area displays the following PL/SQL code:

```
1011 PROCEDURE schimba_accesorii(motocicleta);
1012   v_id_accesoriu IN NUMBER;
1013   nume_accesoriu IN VARCHAR2;
1014
1015   TS
1016   v_id_accesoriu accesoriu.id_accesoriu%TYPE;
1017   NO_DATA_FOUND ACESORIU EXCEPTION;
1018   NO_DATA_FOUND MOTOCICLETA EXCEPTION;
1019   TOO_MANY_ROWS ACESORIU EXCEPTION;
1020   ACESORIU_DEJA_ATASAT EXCEPTION;
1021   control INT;
1022
1023   BEGIN
1024     SELECT COUNT(*) INTO control
1025     FROM MOTOCICLETA
1026     WHERE ID_MOTOCICLETA=v_id_motocicleta;
1027
1028     IF control=0 THEN
1029       RAISE NO_DATA_FOUND_MOTOCICLETA;
1030     END IF;
1031
1032     SELECT
1033       COUNT(*) INTO control
1034     FROM ACESORII
1035     WHERE UPPER(NUME)=UPPER(nume_accesoriu);
1036
1037     IF control=0 THEN
1038       RAISE NO_DATA_FOUND_ACESORIU;
1039     ELSIF control>1 THEN
1040       RAISE TOO_MANY_ROWS_ACESORIU;
1041     END IF;
1042
1043     SELECT
1044       ID_ACESORIU INTO v_id_accesoriu
1045     FROM ACESORII
1046     WHERE UPPER(NUME)=UPPER(nume_accesoriu);
1047
1048
1049     SELECT
1050       COUNT(*) INTO control
1051     FROM ACESORII_MOTOCICLETE
1052     WHERE ID_ACESORIU=v_id_accesoriu AND ID_MOTOCICLETA=v_id_motocicleta;
```

The code implements a procedure to change accessories for a motorcycle. It first checks if the motorcycle exists. Then, it checks if the accessory name is unique. If not, it raises an exception. Finally, it selects the ID of the existing accessory or inserts a new one if none exists.

The screenshot shows the Oracle SQL Developer interface. The title bar says "Oracle SQL Developer : C:\Users\Minu\project\_center.sql". The main area displays the following output from the previous code execution:

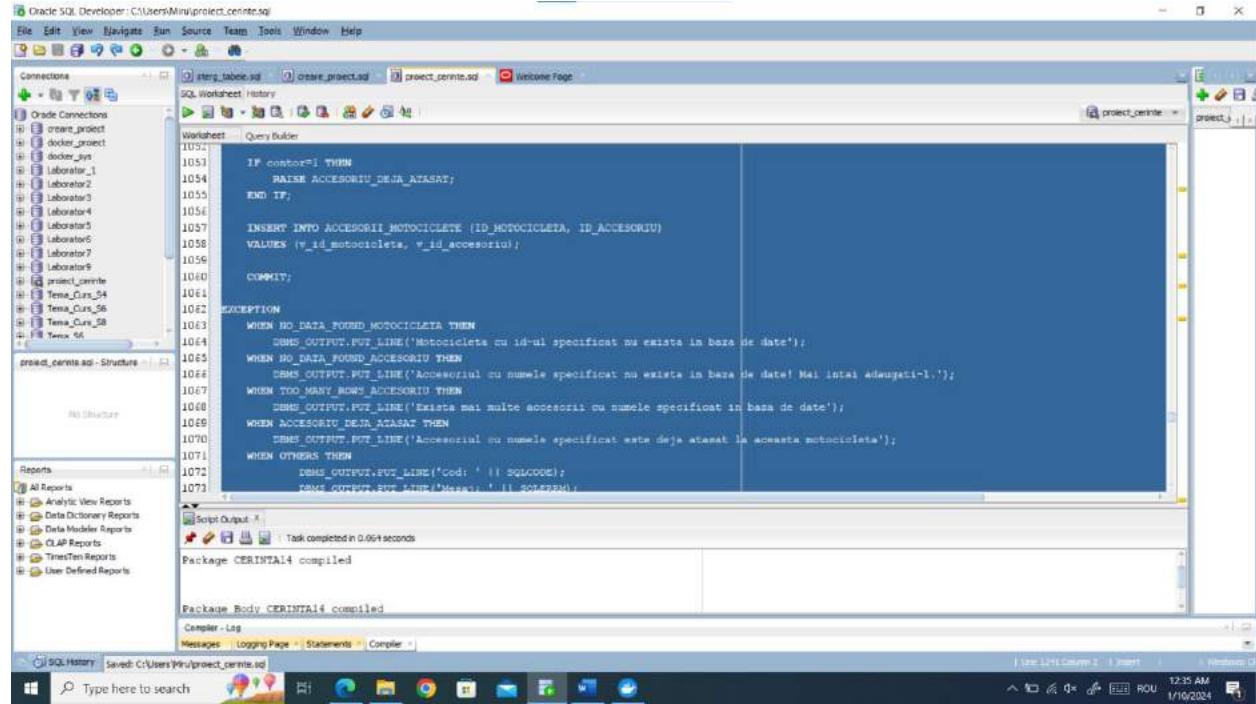
```
Task completed in 0.064 seconds
Package CERINTA14 compiled
```

The package has been successfully compiled. The status message indicates the task completed in 0.064 seconds.

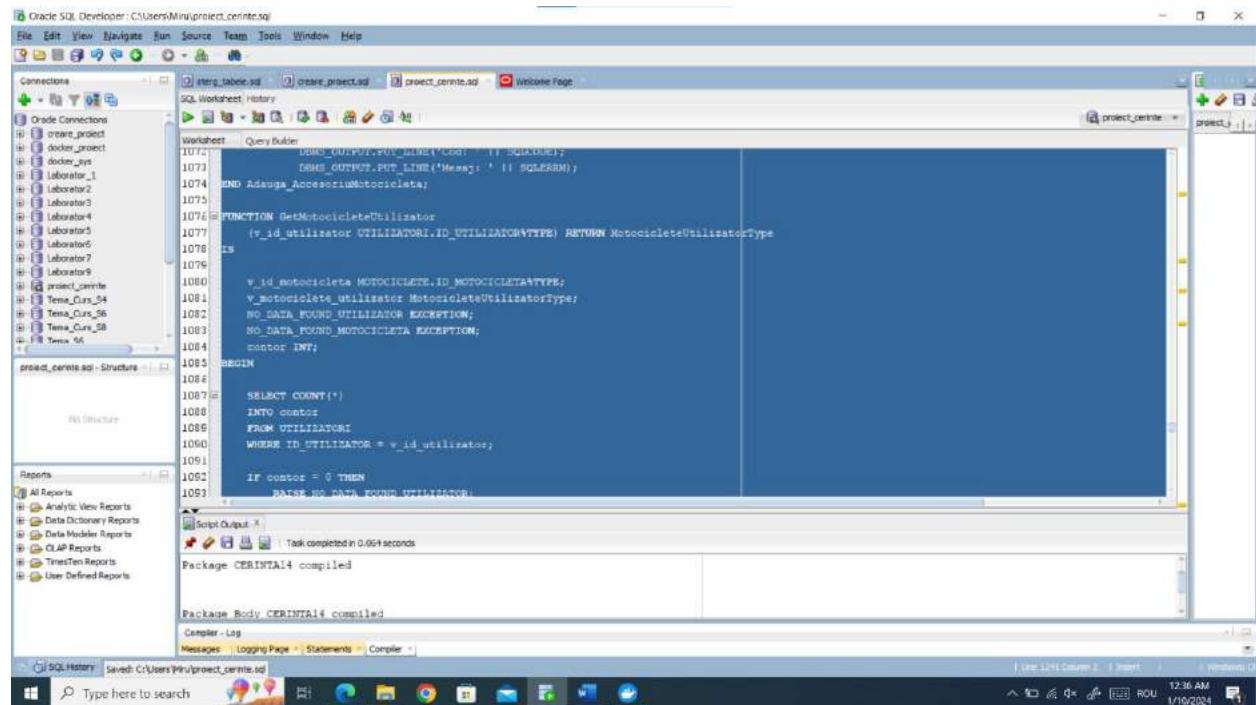
# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242



```
1051 IF contor>1 THEN
1052    RAISE ACCESSORIU_DEJA_ATASAT;
1053 END IF;
1054
1055 INSERT INTO ACCESSORII_MOTOCICLETA (ID_MOTOCICLETA, ID_ACCESSORIU)
1056 VALUES (v_id_motocicleta, v_id_accesoriu);
1057
1058 COMMIT;
1059
1060 EXCEPTION
1061 WHEN NO_DATA_FOUND_MOTOCICLETA THEN
1062    DBMS_OUTPUT.PUT_LINE('Motocicleta cu id-ul specificat nu exista in baza de date');
1063 WHEN NO_DATA_FOUND_ACCESSORIU THEN
1064    DBMS_OUTPUT.PUT_LINE('Accesoriu cu numele specificat nu exista in baza de date! Mai intai adaugati-l.');
1065 WHEN TOO_MANY_ROWS_ACCESSORIU THEN
1066    DBMS_OUTPUT.PUT_LINE('Există mai multe accesorii cu numele specificat in baza de date!');
1067 WHEN ACCESSORIU_DEJA_ATASAT THEN
1068    DBMS_OUTPUT.PUT_LINE('Accesoriu cu numele specificat este deja atasat la aceasta motocicleta');
1069 WHEN OTHERS THEN
1070    DBMS_OUTPUT.PUT_LINE('Cod: ' || SQLCODE);
1071    RAISE_APPLICATION_ERROR(-10000, 'Mesaj: ' || SQLERRM);
1072
1073
```



```
1071    DBMS_OUTPUT.PUT_LINE('Cod: ' || SQLCODE);
1072    DBMS_OUTPUT.PUT_LINE('Mesaj: ' || SQLERRM);
1073 END Adauga_AccesoriumMotocicleta;
1074
1075 FUNCTION GetMotocicleteUtilizator
1076     (v_id_utilizator UTILIZATOR.ID_UTILIZATOR%TYPE) RETURN MotocicleteUtilizatorType
1077 IS
1078
1079    v_id_motocicleta MOTOCICLETA.ID_MOTOCICLETA%TYPE;
1080    v_Motociclete_Utilizator MotocicleteUtilizatorType;
1081    NO_DATA_FOUND_UTILIZATOR EXCEPTION;
1082    NO_DATA_FOUND_MOTOCICLETA EXCEPTION;
1083    contor INT;
1084
1085 BEGIN
1086
1087    SELECT COUNT(*)
1088    INTO contor
1089    FROM UTILIZATORI
1090    WHERE ID_UTILIZATOR = v_id_utilizator;
1091
1092    IF contor = 0 THEN
1093        RAISE NO_DATA_FOUND_UTILIZATOR;
1094
1095
```

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

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

- Project:** project\_center
- Connection:** Oracle Connections
- File:** C:\Users\Minu\project\_center.sql
- Worksheet:** Query Builder
- Code:** A PL/SQL block for a package named CERINTA14. The code includes a cursor for a table named MOTOCICLETA and a procedure named GetMotocicleteUtilizator.

```
1094  SELECT COUNT(*)
1095    INTO contor
1096   FROM MOTOCICLETA
1097  WHERE ID_UTILIZATOR = v_id_utilizator;
1098
1099  IF contor = 0 THEN
1100    RAISE NO_DATA_FOUND_MOTOCICLETA;
1101  END IF;
1102
1103  v_motociclete_utilizator := MotocicleteUtilizatorType();
1104
1105  FOR motocicleta_rec IN (SELECT ID_MOTOCICLETA
1106                            FROM MOTOCICLETA
1107                           WHERE ID_UTILIZATOR = v_id_utilizator) LOOP
1108    v_id_motocicleta := motocicleta_rec.ID_MOTOCICLETA;
1109    v_motociclete_utilizator.EXTEND;
1110    v_motociclete_utilizator(v_motociclete_utilizator.LAST) := v_id_motocicleta;
1111  END LOOP;
1112
1113  RETURN v_motociclete_utilizator;
1114
1115  EXCEPTION
1116    WHEN NO_DATA_FOUND_UTILIZATOR THEN
1117      DBMS_OUTPUT.PUT_LINE('Utilizatorul cu id-ul specificat nu exist!');
1118      RETURN MotocicleteUtilizatorType();
1119    WHEN NO_DATA_FOUND_MOTOCICLETA THEN
1120      DBMS_OUTPUT.PUT_LINE('Utilizatorul cu id-ul specificat nu define motocicleta');
1121      RETURN MotocicleteUtilizatorType();
1122    WHEN OTHERS THEN
1123      DBMS_OUTPUT.PUT_LINE('Cod: ' || SQLCODE);
1124      DBMS_OUTPUT.PUT_LINE('Mesaj: ' || SQLERRM);
1125      RETURN MotocicleteUtilizatorType();
1126  END GetMotocicleteUtilizator;
1127
1128  PROCEDURE Afiseaza_InfoMotocicleteUtilizator
1129    (v_id_utilizator UTILIZATOR.ID_UTILIZATOR%TYPE)
1130  AS
1131    info_utilizator InfoUtilizator;
1132    v_motociclete_utilizator MotocicleteUtilizatorType;
1133
1134  BEGIN
1135    v_motociclete_utilizator := GetMotocicleteUtilizator(v_id_utilizator);
```

- Script Output:** Task completed in 0.064 seconds
- Compiler Log:** Package CERINTA14 compiled
- Messages:** Logging Page | Statements | Compiler

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

- Project:** project\_center
- Connection:** Oracle Connections
- File:** C:\Users\Minu\project\_center.sql
- Worksheet:** Query Builder
- Code:** A PL/SQL block for a package named CERINTA14. The code includes a cursor for a table named MOTOCICLETA and a procedure named GetMotocicleteUtilizator.

```
1114
1115  RETURN v_motociclete_utilizator;
1116  EXCEPTION
1117    WHEN NO_DATA_FOUND_UTILIZATOR THEN
1118      DBMS_OUTPUT.PUT_LINE('Utilizatorul cu id-ul specificat nu exist!');
1119      RETURN MotocicleteUtilizatorType();
1120    WHEN NO_DATA_FOUND_MOTOCICLETA THEN
1121      DBMS_OUTPUT.PUT_LINE('Utilizatorul cu id-ul specificat nu define motocicleta');
1122      RETURN MotocicleteUtilizatorType();
1123    WHEN OTHERS THEN
1124      DBMS_OUTPUT.PUT_LINE('Cod: ' || SQLCODE);
1125      DBMS_OUTPUT.PUT_LINE('Mesaj: ' || SQLERRM);
1126      RETURN MotocicleteUtilizatorType();
1127  END GetMotocicleteUtilizator;
1128
1129  PROCEDURE Afiseaza_InfoMotocicleteUtilizator
1130    (v_id_utilizator UTILIZATOR.ID_UTILIZATOR%TYPE)
1131  AS
1132    info_utilizator InfoUtilizator;
1133    v_motociclete_utilizator MotocicleteUtilizatorType;
1134
1135  BEGIN
1136    v_motociclete_utilizator := GetMotocicleteUtilizator(v_id_utilizator);
```

- Script Output:** Task completed in 0.064 seconds
- Compiler Log:** Package CERINTA14 compiled
- Messages:** Logging Page | Statements | Compiler

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface with the code for the `GetMotocicleteUtilizator` function. The code uses a cursor `v_motociclete_utilizator` to retrieve information about motorcycle users. It handles cases where no users are found or when there are multiple users. For each user, it retrieves their name and all associated motorcycles. The code is annotated with comments explaining its logic.

```
1115:  v_motociclete_utilizator := GetMotocicleteUtilizator(v_id_utilizator);
1116:  info_utilizator:=Get_UtilizatorInfo(v_id_utilizator);
1117:  IF v_motociclete_utilizator.COUNT>0 THEN
1118:    DBMS_OUTPUT.PUT_LINE('Utilizatorul ''||info_utilizator.name||'' nu detine nicio motocicleta');
1119:  ELSE
1120:    DBMS_OUTPUT.PUT_LINE('Utilizatorul ''||info_utilizator.name||'' detine ''||v_motociclete_utilizator.COUNT||'';
1121:    DBMS_OUTPUT.PUT_LINE('.....');
1122:    FOR i IN 1..v_motociclete_utilizator.COUNT LOOP
1123:      afisare_infoMotociclete(v_motociclete_utilizator(i));
1124:      DBMS_OUTPUT.PUT_LINE('.....');
1125:    END LOOP;
1126:  END IF;
1127:  END Afisaza_InfoMotocicleteUtilizator;
1128:
1129: FUNCTION GetAccesoriiMotocicleta
1130:   (v_id_motocicleta MOTOCICLETA.ID_MOTOCICLETA%TYPE) RETURN NameAccesoriiType
1131: IS
1132:   v_accesorii NameAccesoriiType:=NameAccesoriiType();
1133:   counter INT;
1134:
1135: BEGIN
1136:   BEGIN
1137:     SELECT
1138:       COUNT(*) INTO counter
1139:     FROM MOTOCICLETA
1140:     WHERE ID_MOTOCICLETA=v_id_motocicleta;
1141:
1142:     IF counter=0 THEN
1143:       RAISE NO_DATA_FOUND_MOTOCICLETA;
1144:     END IF;
1145:
1146:     FOR accesoriu_rec IN (SELECT A.DENUMIRE
1147:                            FROM ACCESORII_MOTOCICLETA AM
1148:                            JOIN ACCESORII A ON AM.ID_ACCESORIU = A.ID_ACCESORIU
1149:                            WHERE AM.ID_MOTOCICLETA = v_id_motocicleta) LOOP
1150:       v_accesorii(v_accesorii.COUNT + 1) := accesoriu_rec.name;
1151:     END LOOP;
1152:
1153:     RETURN v_accesorii;
1154:   EXCEPTION
1155:     WHEN NO_DATA_FOUND_MOTOCICLETA THEN
1156:       DBMS_OUTPUT.PUT_LINE('Nu exista motocicleta in baza de date cu id-ul specificat!');
1157:       RETURN NameAccesoriiType();
1158:     WHEN OTHERS THEN
1159:       DBMS_OUTPUT.PUT_LINE('Eroare la recuperarea accesoriilor pentru motocicleta cu id-ul specificat!');
1160:   END;
1161:
1162: END;
1163:
1164: END CERINTA14;
1165:
1166: /
```

The screenshot shows the Oracle SQL Developer interface with the code for the `GetAccesoriiMotocicleta` function. The function takes a motorcycle ID as input and returns a list of accessories. It first checks if the motorcycle exists. If it does, it retrieves all accessories associated with that motorcycle. If the motorcycle does not exist, it raises an error. The code is annotated with comments explaining its logic.

```
1132:  v_accesorii NameAccesoriiType:=NameAccesoriiType();
1133:  counter INT;
1134:
1135: BEGIN
1136:   BEGIN
1137:     SELECT
1138:       COUNT(*) INTO counter
1139:     FROM MOTOCICLETA
1140:     WHERE ID_MOTOCICLETA=v_id_motocicleta;
1141:
1142:     IF counter=0 THEN
1143:       RAISE NO_DATA_FOUND_MOTOCICLETA;
1144:     END IF;
1145:
1146:     FOR accesoriu_rec IN (SELECT A.DENUMIRE
1147:                            FROM ACCESORII_MOTOCICLETA AM
1148:                            JOIN ACCESORII A ON AM.ID_ACCESORIU = A.ID_ACCESORIU
1149:                            WHERE AM.ID_MOTOCICLETA = v_id_motocicleta) LOOP
1150:       v_accesorii(v_accesorii.COUNT + 1) := accesoriu_rec.name;
1151:     END LOOP;
1152:
1153:     RETURN v_accesorii;
1154:   EXCEPTION
1155:     WHEN NO_DATA_FOUND_MOTOCICLETA THEN
1156:       DBMS_OUTPUT.PUT_LINE('Nu exista motocicleta in baza de date cu id-ul specificat!');
1157:       RETURN NameAccesoriiType();
1158:     WHEN OTHERS THEN
1159:       DBMS_OUTPUT.PUT_LINE('Eroare la recuperarea accesoriilor pentru motocicleta cu id-ul specificat!');
1160:   END;
1161:
1162: END;
1163:
1164: END CERINTA14;
1165:
1166: /
```

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
1177    WHEN OTHERS THEN
1178        DBMS_OUTPUT.PUT_LINE('Cod: ' || SQLCODE);
1179        DBMS_OUTPUT.PUT_LINE('Mesaj: ' || SQUERERR);
1180        RETURN NameAccessoriType();
1181    END GetAccesoriiMotociclete;
1182
1183    PROCEDURE actiuni_motocicleta (v_nr_inmatriculara motocicleta.nr_inmatriculara%TYPE, taxa NUMBER DEFAULT 0)
1184    IS
1185        v_id_motocicleta motocicleta.id_motocicleta%TYPE;
1186        info_utilizator InfoUtilizator;
1187        v_accesorii NameAccessoriType;
1188
1189    BEGIN
1190        v_id_motocicleta:=GetIdMotocicleta(v_nr_inmatriculara);
1191        DBMS_OUTPUT.PUT_LINE('Am introducut taxa'||taxa);
1192        IF taxa<0 THEN
1193            DBMS_OUTPUT.PUT_LINE('Actiune anulata cu succes!');
1194        ELSIF taxa>1 THEN
1195            --afisarea detaliilor proprietarului motocicletei
1196            DBMS_OUTPUT.PUT_LINE('Actiune realizata cu succes!');
1197            DBMS_OUTPUT.PUT_LINE('Informatie proprietarul motocicletei cu numarul de inmatriculara specificat:');
1198            DBMS_OUTPUT.PUT_LINE('.....');
1199            DBMS_OUTPUT.PUT_LINE('.....');
1200            DBMS_OUTPUT.PUT_LINE('.....');
1201            DBMS_OUTPUT.PUT_LINE('.....');
1202            DBMS_OUTPUT.PUT_LINE('.....');
1203            DBMS_OUTPUT.PUT_LINE('Data nasterii: '||info_utilizator.data_nasterii);
1204        ELSIF taxa=1 THEN
1205            --afisarea detaliilor platiilor efectuate pentru o anumita motocicleta
1206            DBMS_OUTPUT.PUT_LINE('Actiune realizata cu succes!');
1207            DBMS_OUTPUT.PUT_LINE('Afisarea detaliilor platilor efectuate pentru motociclete cu numarul de inmatriculara specificat');
1208            DBMS_OUTPUT.PUT_LINE('.....');
1209            plati_motocicleta(v_id_motocicleta);
1210        ELSIF taxa=3 THEN
1211            --adaugarea unui nou accesoriu pentru o motocicleta
1212            DBMS_OUTPUT.PUT_LINE('Actiune realizata cu succes!');
1213            DBMS_OUTPUT.PUT_LINE('Adaugarea unui accesoriu la motocicleta cu numarul de inmatriculara specificat');
1214            DBMS_OUTPUT.PUT_LINE('.....');
1215            Adauga_AccesoriuMotocicleta(v_id_motocicleta, 'Camca');
1216        ELSIF taxa=4 THEN
1217            --afisarea motocicletelor proprietarului
1218            DBMS_OUTPUT.PUT_LINE('Actiune realizata cu succes!');
```

```
1197    WHEN OTHERS THEN
1198        DBMS_OUTPUT.PUT_LINE('.....');
1199        info_utilizator:=Get_UtilizatorInfo(GetIdUtilizator(v_id_motocicleta));
1200        DBMS_OUTPUT.PUT_LINE('Nume: '||info_utilizator.name);
1201        DBMS_OUTPUT.PUT_LINE('Prenume: '||info_utilizator.prenume);
1202        DBMS_OUTPUT.PUT_LINE('Email: '||info_utilizator.email);
1203        DBMS_OUTPUT.PUT_LINE('Varsta: '||info_utilizator.varsta);
1204        DBMS_OUTPUT.PUT_LINE('Data nasterii: '||info_utilizator.data_nasterii);
1205    ELSIF taxa=2 THEN
1206        --afisarea detaliilor platilor efectuate pentru o anumita motocicleta
1207        DBMS_OUTPUT.PUT_LINE('Actiune realizata cu succes!');
1208        DBMS_OUTPUT.PUT_LINE('Afisarea detaliilor platilor efectuate pentru motociclete cu numarul de inmatriculara specificat');
1209        DBMS_OUTPUT.PUT_LINE('.....');
1210        plati_motocicleta(v_id_motocicleta);
1211    ELSIF taxa=3 THEN
1212        --adaugarea unui nou accesoriu pentru o motocicleta
1213        DBMS_OUTPUT.PUT_LINE('Actiune realizata cu succes!');
1214        DBMS_OUTPUT.PUT_LINE('Adaugarea unui accesoriu la motocicleta cu numarul de inmatriculara specificat');
1215        DBMS_OUTPUT.PUT_LINE('.....');
1216        Adauga_AccesoriuMotocicleta(v_id_motocicleta, 'Camca');
1217    ELSIF taxa=4 THEN
1218        --afisarea motocicletelor proprietarului
1219        DBMS_OUTPUT.PUT_LINE('Actiune realizata cu succes!');
```

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

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

- File Path:** C:\Users\Minu\project\_center.sql
- Connections:** Shows various database connections including docker\_project, docker\_pys, Laborator\_1, Laborator2, Laborator3, Laborator4, Laborator5, Laborator6, Laborator7, Laborator8, project\_center, Teme\_Curs\_54, Teme\_Curs\_56, Teme\_Curs\_58, and Teme\_64.
- SQL Worksheet:** Displays the PL/SQL code for package CERINTA14. The code handles actions for a motorcycle with ID 123-XYZ. It includes logic to check if accessories exist and to execute actions like 'acumul\_moto' or 'cerintal4.acumul\_moto' based on user input.
- Script Output:** Shows a message indicating the task completed in 0.064 seconds.
- Compiler Log:** Shows the package was compiled successfully.
- Messages:** Shows the saved file path: C:\Users\Minu\project\_center.sql.

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

- File Path:** C:\Users\Minu\project\_center.sql
- Connections:** Shows various database connections including docker\_project, docker\_pys, Laborator\_1, Laborator2, Laborator3, Laborator4, Laborator5, Laborator6, Laborator7, Laborator8, project\_center, Teme\_Curs\_54, Teme\_Curs\_56, Teme\_Curs\_58, and Teme\_64.
- SQL Worksheet:** Displays the same PL/SQL code as the previous screenshot, but with additional lines for executing actions on the motorcycle.
- Script Output:** Shows a message indicating the task completed in 0.064 seconds.
- Compiler Log:** Shows the package was compiled successfully.
- Messages:** Shows the saved file path: C:\Users\Minu\project\_center.sql.

➤ Testare pachet

Deoarece ultima procedură, denumită sugestiv *actiuni\_motocicleta*, le apelează pe toate celelalte prin intermediul celui de-al doilea parametru, care transmite un cod specific pentru fiecare acțiune dorită pentru a se efectua asupra unei motociclete cu un număr de înmatriculare specificat, voi apela toate cazurile acestei proceduri, pe rând. De asemenea, după voi apela într-un bloc separat și fiecare procedură/funcție pe rând, pentru a se evidenția mai bine funcționalitatea fiecăreia.

➤ Cod PL/SQL

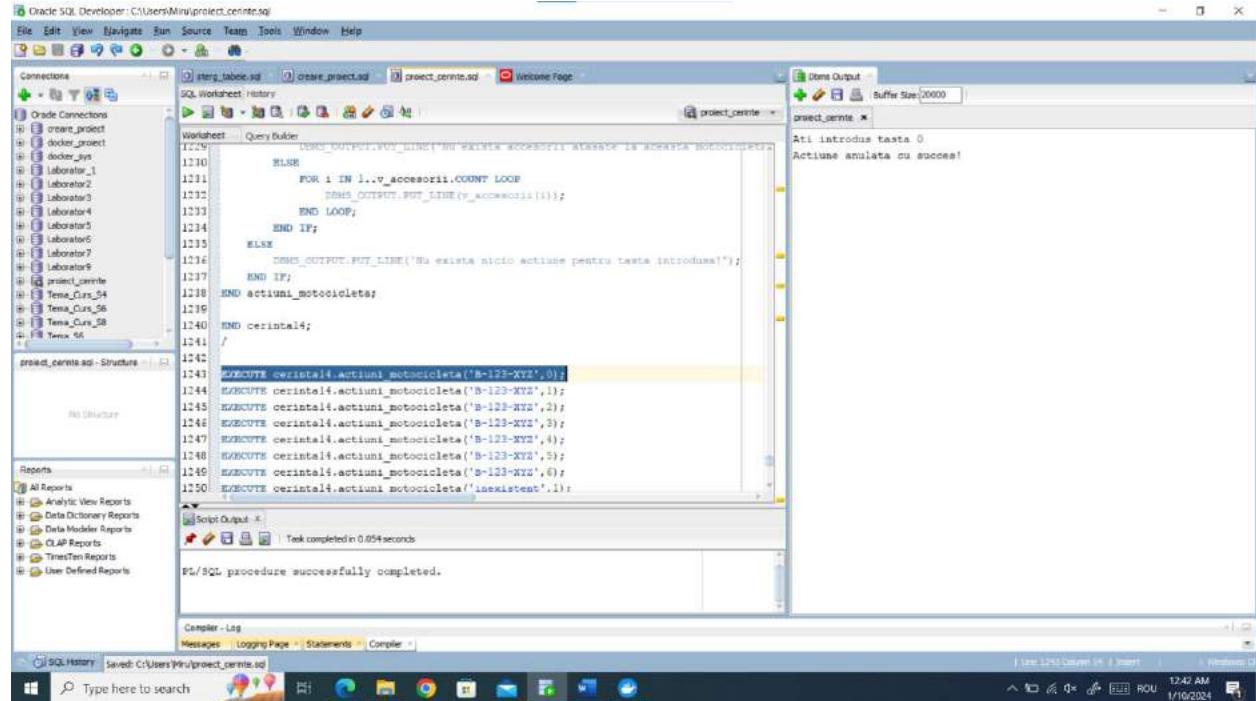
```
EXECUTE cerinta14.actiuni_motocicleta('B-123-XYZ',0);
EXECUTE cerinta14.actiuni_motocicleta('B-123-XYZ',1);
EXECUTE cerinta14.actiuni_motocicleta('B-123-XYZ',2);
EXECUTE cerinta14.actiuni_motocicleta('B-123-XYZ',3);
EXECUTE cerinta14.actiuni_motocicleta('B-123-XYZ',4);
EXECUTE cerinta14.actiuni_motocicleta('B-123-XYZ',5);
EXECUTE cerinta14.actiuni_motocicleta('B-123-XYZ',6);
EXECUTE cerinta14.actiuni_motocicleta('inexistent',1);
EXECUTE cerinta14.actiuni_motocicleta('B-10-234',1);
```

➤ Pentru codul 0

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242



```
1229      DBMS_OUTPUT.PUT_LINE('Nu exista accesorii stocate la anexa motocicleta');
1230
1231      ELSE
1232          FOR i IN 1..v_accesorii.COUNT LOOP
1233              DBMS_OUTPUT.PUT_LINE(v_accesorii(i));
1234          END LOOP;
1235      END IF;
1236
1237      ELSE
1238          DBMS_OUTPUT.PUT_LINE('Nu exista nicio actiune pentru testa introducuta');
1239      END IF;
1240
1241      END actiuni_motocicleta;
1242
1243      EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',0);
1244      EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',1);
1245      EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',2);
1246      EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',3);
1247      EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',4);
1248      EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',5);
1249      EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',6);
1250      EXECUTE cerintal4.actiuni_motocicleta('inexistent',1);
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2388
2389
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2398
2399
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2498
2499
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2598
2599
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2698
2699
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2798
2799
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2898
2899
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2998
2999
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3098
3099
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3198
3199
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3239
3240
3241
3242
3243
3244
3245
3246
3247
32
```

## Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

➤ Pentru codul 2

```
1229    DBMS_OUTPUT.PUT_LINE('Nu exista accesoriul acesta');
1230    ELSE
1231        FOR i IN 1..v_accesorii.COUNT LOOP
1232            DBMS_OUTPUT.PUT_LINE(v_accesorii(i));
1233        END LOOP;
1234    END IF;
1235
1236    ELSE
1237        DBMS_OUTPUT.PUT_LINE('Nu exista nicio actiune pentru acesta');
1238    END IF;
1239    END actiuni_motocicleta;
1240
1241    END cerintal4;
1242
1243    EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',0);
1244    EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',1);
1245    EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',2);
1246    EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',3);
1247    EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',4);
1248    EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',5);
1249    EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',6);
1250    EXECUTE cerintal4.actiuni_motocicleta('inexistent',1);
1251
1252
1253    PL/SQL procedure successfully completed.
```

➤ Pentru codul 3

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

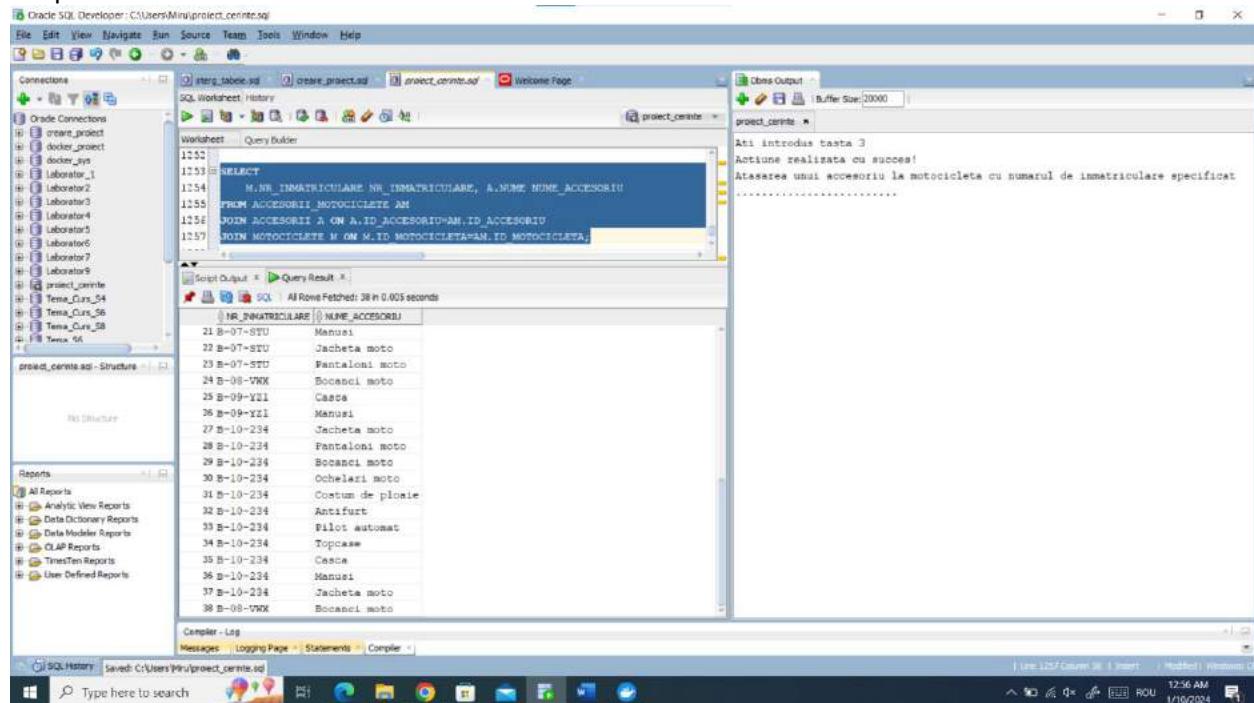
The screenshot shows the Oracle SQL Developer interface. In the central workspace, a PL/SQL procedure named `project_centerite` is being run. The code performs several database operations, including executing stored procedures `cerintal4.actiuni_motocicleta` with various parameters (e.g., 'B-123-XYZ', 1, 3, 5, 6) and selecting data from tables like `MOTOCICLETE` and `ACCESORII`. The output window displays a success message: "Ati introdus tasta 3" and "Actiune realizata cu succes! Atasarea unui accesoriu la motocicleta cu numarul de immatriculare specificat". The status bar at the bottom right indicates the date and time as 1/16/2024 12:55 AM.

The screenshot shows the Oracle SQL Developer interface again. This time, a query is being run. The code uses joins between tables `ACCESORII`, `MOTOCICLETE`, and `ACCESORIU_MOTOCICLETA` to retrieve data. The output window shows the results of the query, which lists various accessories (e.g., Cască, Bucaneți moto, Ochelari moto, Costum de ploaie, Antifurt, Pilot automat, Manusi, Pantalon moto, Ochelari moto, Cască, Jacheta moto, Bucaneți moto, Cască, Pantalon moto, Costum de ploaie, Jacheta moto, Bucaneți moto, Antifurt) along with their respective motorcycle numbers (e.g., B-123-XYZ, D-759-ABC, E-012-XYZ). The status bar at the bottom right indicates the date and time as 1/16/2024 12:56 AM.

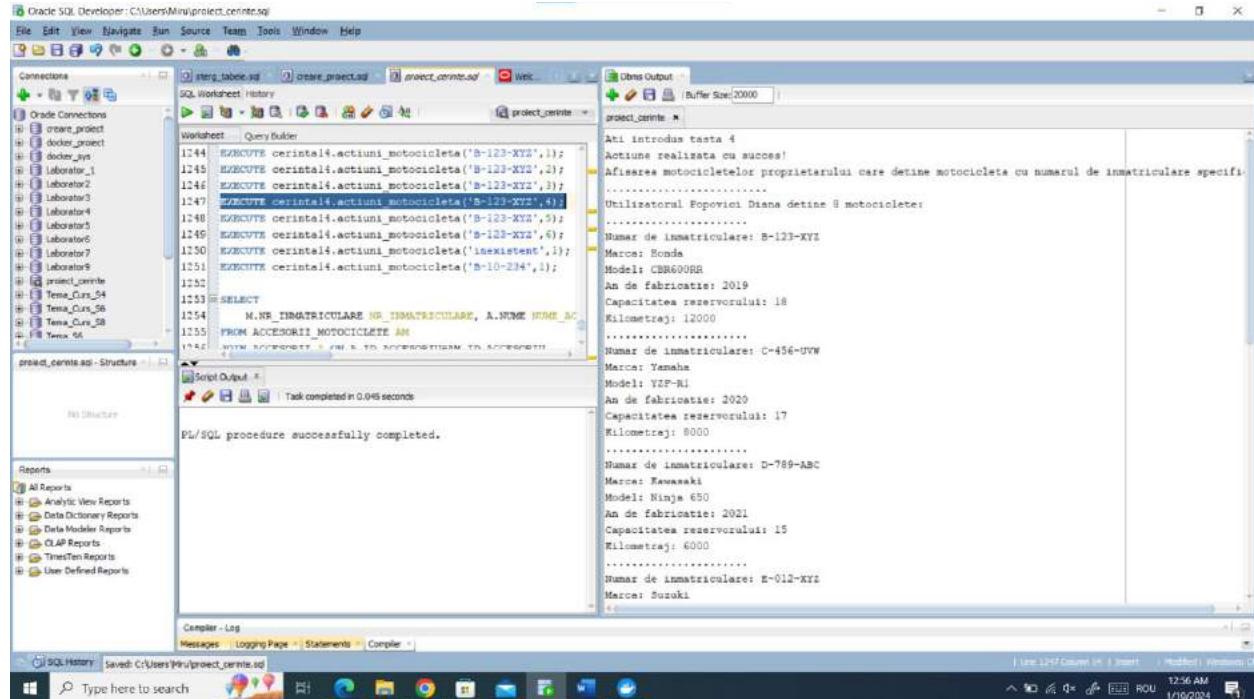
Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242



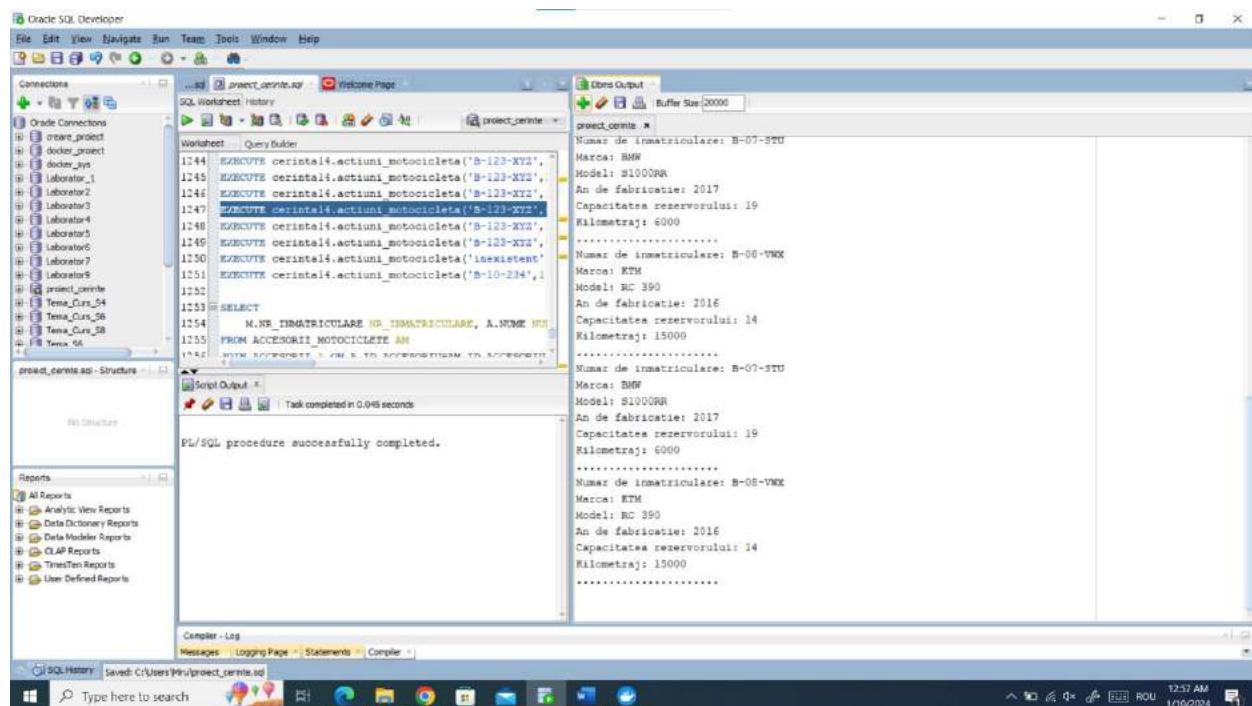
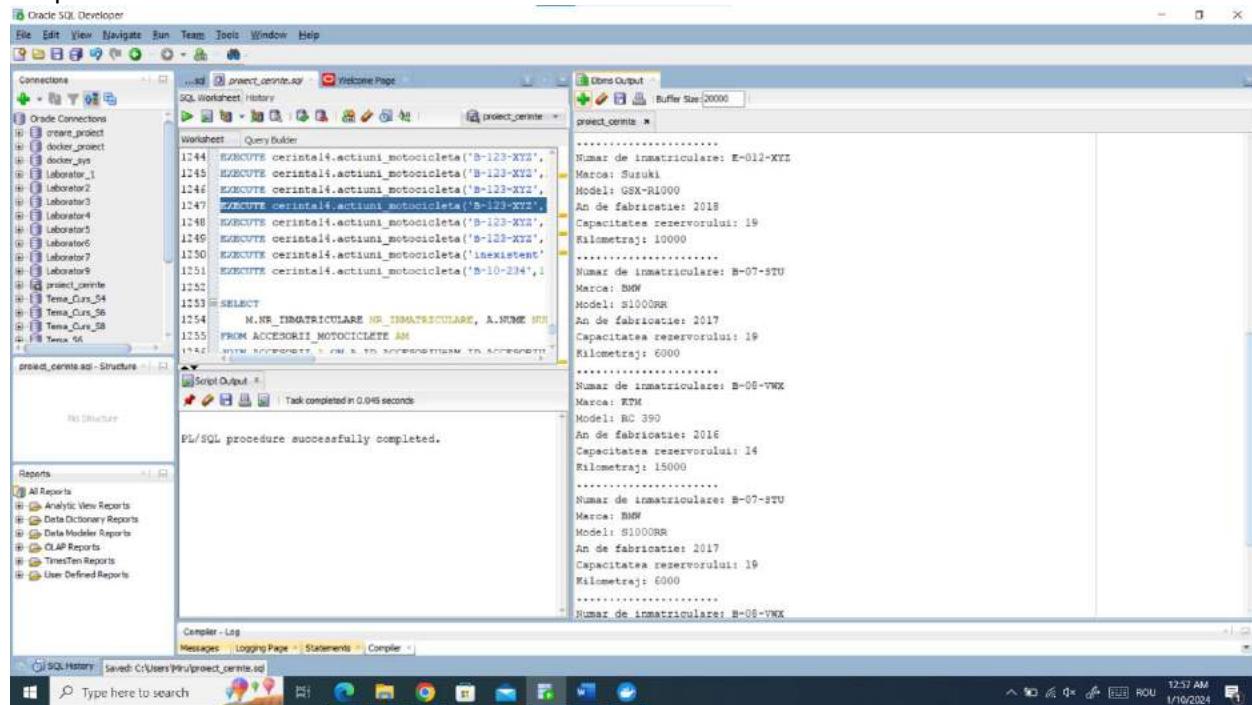
➤ Pentru codul 4



## Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242



➤ Pentru codul 5

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

The screenshot shows the Oracle SQL Developer interface. In the central workspace, a PL/SQL procedure named 'project\_center.ad' is being run. The code includes several EXECUTE statements for a procedure named 'cerintal4.actiuni\_motocicleta' with various parameters like 'B-123-XYZ'. It also includes a SELECT statement to retrieve data from tables 'ACCESORII\_MOTOCICLETE' and 'AM'. The 'Output' tab shows the results of the procedure execution, indicating that 5 tasks were introduced and successfully completed. The status message 'PL/SQL procedure successfully completed.' is displayed at the bottom of the output window.

```
1244 EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',1);
1245 EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',2);
1246 EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',3);
1247 EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',4);
1248 EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',5);
1249 EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',6);
1250 EXECUTE cerintal4.actiuni_motocicleta('inexistent',1);
1251 EXECUTE cerintal4.actiuni_motocicleta('B-10-234',1);

1252
1253 SELECT
1254     M.NR_INMATRICULARE NR_INMATRICULARE, A.NUME NUME_ACCESSORIU
1255     FROM ACCESORII_MOTOCICLETE AM
1256     WHERE ACCESSION_ID = 1 AND ACCESSION_ID = ACCESSORY_ID;

PL/SQL procedure successfully completed.
```

## ➤ Pentru codul 6

The screenshot shows the Oracle SQL Developer interface. Similar to the previous one, it displays a PL/SQL procedure 'project\_center.ad' running. The code is identical to the first screenshot. However, the 'Output' tab shows an error message: 'Nu există nicio acțiune pentru tasta introdusă!' (No action exists for the introduced key). The status message 'PL/SQL procedure successfully completed.' is also present.

```
1244 EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',1);
1245 EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',2);
1246 EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',3);
1247 EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',4);
1248 EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',5);
1249 EXECUTE cerintal4.actiuni_motocicleta('B-123-XYZ',6);
1250 EXECUTE cerintal4.actiuni_motocicleta('inexistent',1);
1251 EXECUTE cerintal4.actiuni_motocicleta('B-10-234',1);

1252
1253 SELECT
1254     M.NR_INMATRICULARE NR_INMATRICULARE, A.NUME NUME_ACCESSORIU
1255     FROM ACCESORII_MOTOCICLETE AM
1256     WHERE ACCESSION_ID = 1 AND ACCESSION_ID = ACCESSORY_ID;

PL/SQL procedure successfully completed.
```

## Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

- Pentru codul 1 cu un număr de inmatriculare neexistent

The screenshot shows the Oracle SQL Developer interface. In the central workspace, there is a SQL Worksheet containing the following code:

```
1144 EXECUTE cerinal4.actiuni_motocicleta('B-123-XYZ',1);
1145 EXECUTE cerinal4.actiuni_motocicleta('B-123-XYZ',2);
1146 EXECUTE cerinal4.actiuni_motocicleta('B-123-XYZ',3);
1147 EXECUTE cerinal4.actiuni_motocicleta('B-123-XYZ',4);
1148 EXECUTE cerinal4.actiuni_motocicleta('B-123-XYZ',5);
1149 EXECUTE cerinal4.actiuni_motocicleta('B-123-XYZ',6);
1150 EXECUTE cerinal4.actiuni_motocicleta('inexistent',1);
1151 EXECUTE cerinal4.actiuni_motocicleta('B-10-234',1);
1152
1153 SELECT
```

In the Script Output pane, the following error message is displayed:

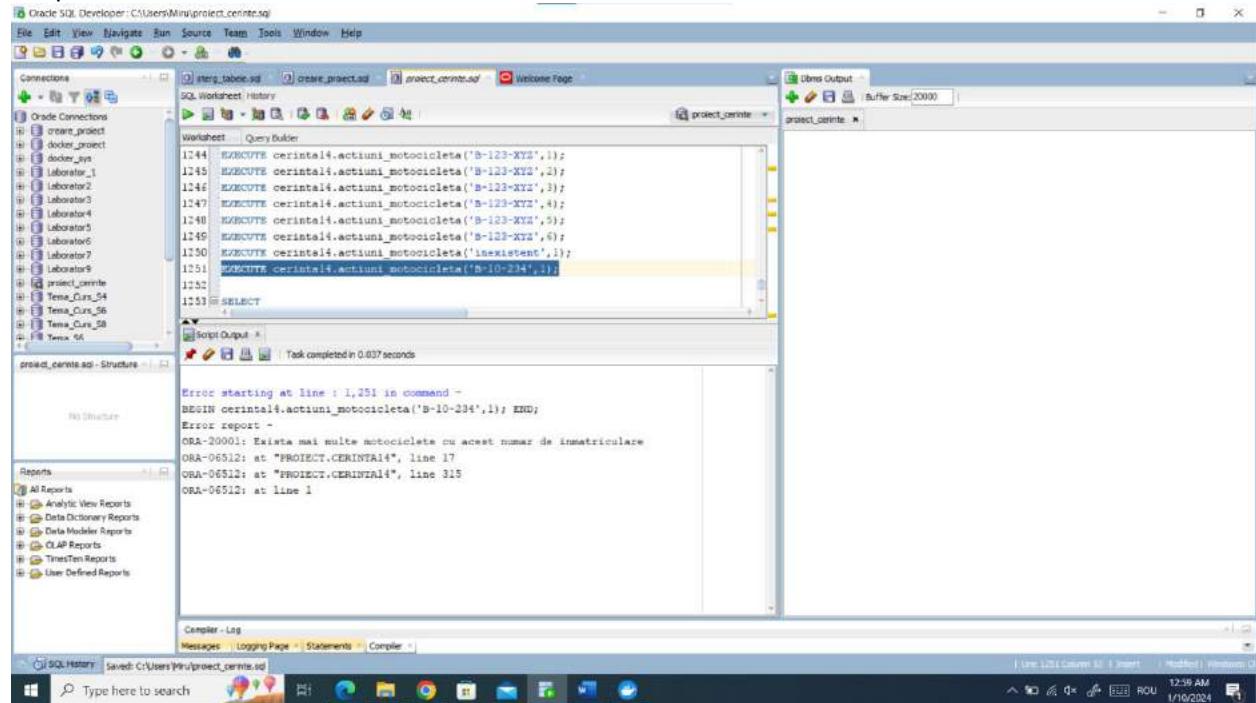
```
Error starting at Line : 1,150 in command -
BEGIN cerinal4.actiuni_motocicleta('inexistent',1); END;
Error report -
ORA-20000: Nu exista motociclete cu acest numar de inmatriculare
ORA-06512: at "PROJECT.CERINAL4", line 15
ORA-06512: at "PROJECT.CERINAL4", line 315
ORA-06512: at line 1
20000. 00000 - "%s"
*Cause:  The stored procedure 'raise_application_error'
        was called which causes this error to be generated.
*Action:  Correct the problem as described in the error message or contact
          the application administrator or DBA for more information.
```

- Pentru codul 1 cu un număr de inmatriculare neexistent care apare la mai multe motociclete

## Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242



- Pentru blocul care conține toate funcțiile, procedurile și colecțiile stocate în pachet, apelate sau declarate

SELECT

```
M.NR_INMATRICULARE NR_INMATRICULARE, A.NUME NUME_ACCESORIU  
FROM ACCESORII_MOTOCICLETE AM  
JOIN ACCESORII A ON A.ID_ACCESORIU=AM.ID_ACCESORIU  
JOIN MOTOCICLETE M ON M.ID_MOTOCICLETA=AM.ID_MOTOCICLETA;
```

DECLARE

```
info InfoUtilizator;  
motociclete_utilizator cerinta14.MotocicleteUtilizatorType;  
nume_accesorii cerinta14.NumeAccesoriiType;  
BEGIN
```

```
DBMS_OUTPUT.PUT_LINE('ID-ul motocicletei cu numarul de inmatriculare B-123-XYZ:  
'||cerinta14.GetIdMotocicleta('B-123-XYZ'));
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

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

DBMS_OUTPUT.PUT_LINE('ID-ul utilizatorului care detine motocicleta cu id-ul 3:
'||cerinta14.GetIdUtilizator(1));

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

DBMS_OUTPUT.PUT_LINE('ID-ul utilizatorului care detine motocicleta cu id-ul 3:
'||cerinta14.GetIdUtilizator(1));

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

info:=cerinta14.Get_UtilizatorInfo(1);

DBMS_OUTPUT.PUT_LINE('Utilizatorul care are id-ul 1 se numeste'||info.nume||' '||info.prenume||' si
are emailul'||info.email);

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

DBMS_OUTPUT.PUT_LINE('Informatii despre motocicleta cu id-ul 1');
cerinta14.afisare_InfoMotocicleta(1);
DBMS_OUTPUT.PUT_LINE('.....');

DBMS_OUTPUT.PUT_LINE('Informatii despre platile alocate motocicletei cu id-ul 1');
cerinta14.plati_motocicleta(1);
DBMS_OUTPUT.PUT_LINE('.....');

DBMS_OUTPUT.PUT_LINE('Atasam accesoriul Casca cu id-ul 5');
cerinta14.Adauga_AccesoriuMotocicleta(5, 'Casca');
DBMS_OUTPUT.PUT_LINE('.....');

DBMS_OUTPUT.PUT_LINE('Stocam in motociclete_utilizator motocicletele utilizatorului cu id-ul
1');
motociclete_utilizator:=cerinta14.GetMotocicleteUtilizator(1);
IF motociclete_utilizator.COUNT = 0 THEN
```

Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

```
    DBMS_OUTPUT.PUT_LINE('Nu exista motociclete detinute de utilizatorul cu id=1');
```

ELSE

```
    FOR i IN 1..motociclete_utilizator.COUNT LOOP
```

```
        DBMS_OUTPUT.PUT_LINE(motociclete_utilizator(i));
```

```
    END LOOP;
```

```
END IF;
```

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

```
DBMS_OUTPUT.PUT_LINE('Afisam informatii despre motocicletele detinute de utilizatorul cu id-ul 1');
```

```
cerinta14.Afiseaza_InfoMotocicleteUtilizator(1);
```

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

```
DBMS_OUTPUT.PUT_LINE('Stocam in nume_accesorii accesoriile motocicletei cu id-ul 1');
```

```
nume_accesorii:=cerinta14.GetAccesoriiMotocicleta(1);
```

```
IF nume_accesorii.COUNT = 0 THEN
```

```
    DBMS_OUTPUT.PUT_LINE('Nu exista accesorii atasate de motocicleta cu id=1');
```

ELSE

```
    FOR i IN 1..nume_accesorii.COUNT LOOP
```

```
        DBMS_OUTPUT.PUT_LINE(nume_accesorii(i));
```

```
    END LOOP;
```

```
END IF;
```

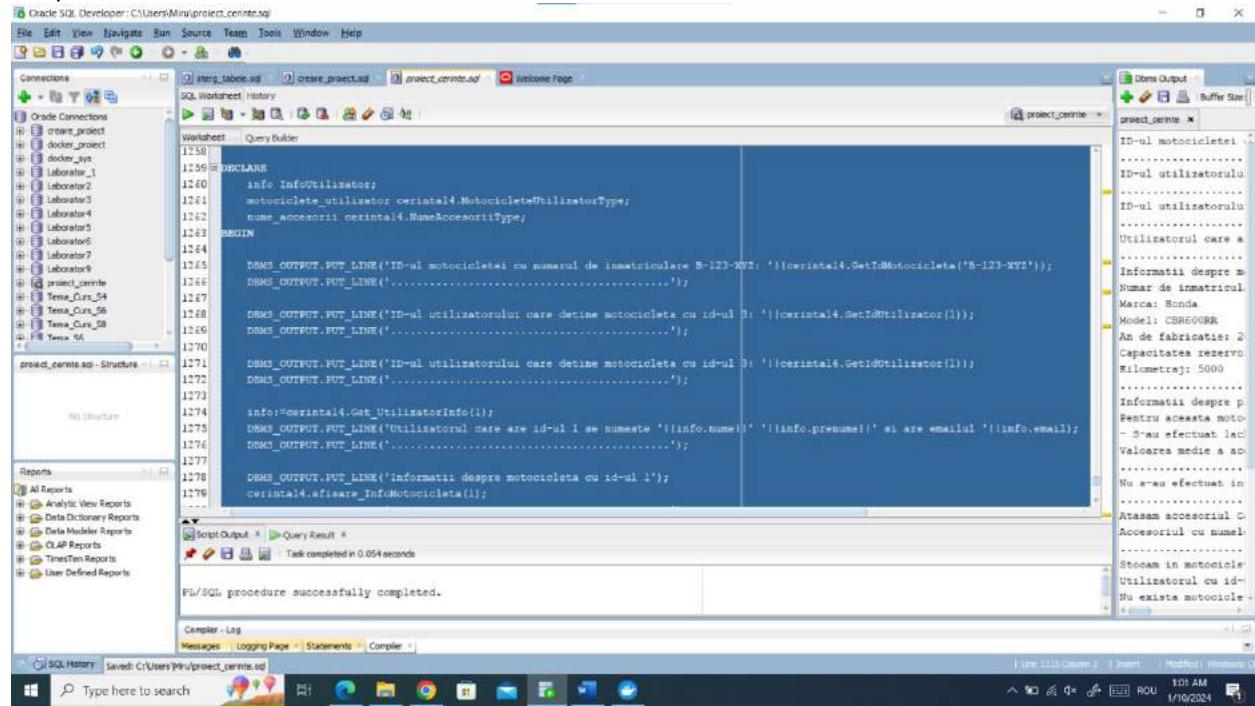
```
END;
```

```
/
```

# Sisteme de gestiune a bazelor de date

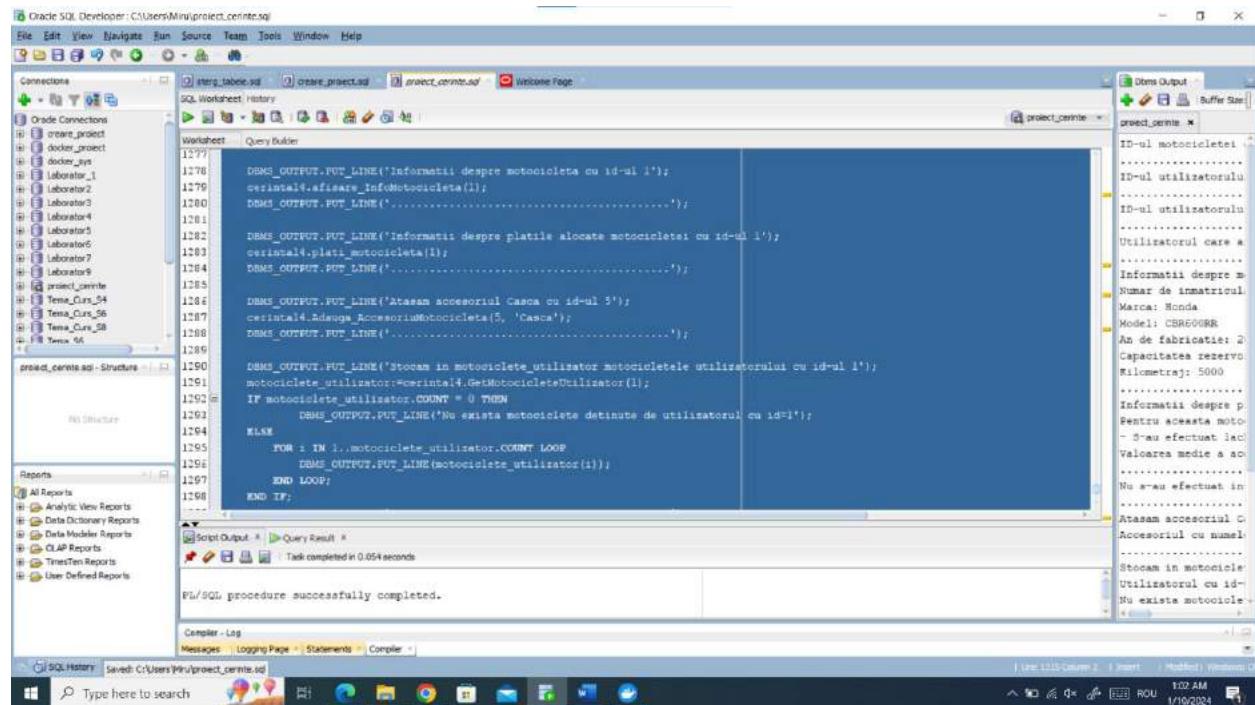
Stoine Maria-Miruna

Grupa 242



```
1250  DECLARE
1251    info_infoUtilizator;
1252    motociclete_utilizator cerintal4.MotocicleteUtilizatorType;
1253    name_accesorii cerintal4.NumeAccesoriiType;
1254
1255  BEGIN
1256
1257    DBMS_OUTPUT.PUT_LINE('ID-ul motocicletei cu numarul de inmatriculare S-123-XYZ: '||cerintal4.GetIdMotocicleta('S-123-XYZ'));
1258    DBMS_OUTPUT.PUT_LINE('.....');
1259
1260    DBMS_OUTPUT.PUT_LINE('ID-ul utilizatorului care detine motocicleta cu id-ul 3: '||cerintal4.GetIdUtilizator(1));
1261    DBMS_OUTPUT.PUT_LINE('.....');
1262
1263    DBMS_OUTPUT.PUT_LINE('ID-ul utilizatorului care detine motocicleta cu id-ul 3: '||cerintal4.GetIdUtilizator(1));
1264    DBMS_OUTPUT.PUT_LINE('.....');
1265
1266    info:cerintal4.Get_UtilizatorInfo();
1267    DBMS_OUTPUT.PUT_LINE('Utilizatorul care are id-ul 1 se numeste "'||info.nume||' '||info.prenume||' si are emailul "'||info.email||'");
1268    DBMS_OUTPUT.PUT_LINE('.....');
1269
1270    DBMS_OUTPUT.PUT_LINE('Informatii despre motocicleta cu id-ul 1:');
1271    cerintal4.afisare_InfoMotocicleta(1);
1272
1273
1274
1275
1276
1277
1278    DBMS_OUTPUT.PUT_LINE('Informatii despre motocicleta cu id-ul 1:');
1279    cerintal4.afisare_InfoMotocicleta();
1280    DBMS_OUTPUT.PUT_LINE('.....');
1281
1282    DBMS_OUTPUT.PUT_LINE('Informatii despre platile alocate motocicletei cu id-ul 1:');
1283    cerintal4.plati_motocicleta();
1284    DBMS_OUTPUT.PUT_LINE('.....');
1285
1286    DBMS_OUTPUT.PUT_LINE('Atasam accesoriul Casca cu id-ul 5');
1287    cerintal4.Adauga_AccesoriuMotocicleta5('Casca');
1288    DBMS_OUTPUT.PUT_LINE('.....');
1289
1290    DBMS_OUTPUT.PUT_LINE('Stocam in motociclete_utilizator motocicletele utilizatorului cu id-ul 1');
1291    motociclete_utilizator:=cerintal4.GetMotocicleteUtilizator(1);
1292    IF motociclete_utilizator.COUNT = 0 THEN
1293      DBMS_OUTPUT.PUT_LINE('Nu exista motociclete detinute de utilizatorul cu id=1');
1294    ELSE
1295      FOR i IN 1..motociclete_utilizator.COUNT LOOP
1296        DBMS_OUTPUT.PUT_LINE(motociclete_utilizator(i));
1297      END LOOP;
1298    END IF;
```

PL/SQL procedure successfully completed.



```
1277
1278    DBMS_OUTPUT.PUT_LINE('Informatii despre motocicleta cu id-ul 1');
1279    cerintal4.afisare_InfoMotocicleta();
1280    DBMS_OUTPUT.PUT_LINE('.....');
1281
1282    DBMS_OUTPUT.PUT_LINE('Informatii despre platile alocate motocicletei cu id-ul 1');
1283    cerintal4.plati_motocicleta();
1284    DBMS_OUTPUT.PUT_LINE('.....');
1285
1286    DBMS_OUTPUT.PUT_LINE('Atasam accesoriul Casca cu id-ul 5');
1287    cerintal4.Adauga_AccesoriuMotocicleta5('Casca');
1288    DBMS_OUTPUT.PUT_LINE('.....');
1289
1290    DBMS_OUTPUT.PUT_LINE('Stocam in motociclete_utilizator motocicletele utilizatorului cu id-ul 1');
1291    motociclete_utilizator:=cerintal4.GetMotocicleteUtilizator(1);
1292    IF motociclete_utilizator.COUNT = 0 THEN
1293      DBMS_OUTPUT.PUT_LINE('Nu exista motociclete detinute de utilizatorul cu id=1');
1294    ELSE
1295      FOR i IN 1..motociclete_utilizator.COUNT LOOP
1296        DBMS_OUTPUT.PUT_LINE(motociclete_utilizator(i));
1297      END LOOP;
1298    END IF;
```

PL/SQL procedure successfully completed.

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

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

- Connections:** Shows various database connections including docker\_project, docker\_pys, Laborator\_1, Laborator2, Laborator3, Laborator4, Laborator5, Laborator6, Laborator7, Laborator8, project\_centerite, Teme\_Curs\_54, Teme\_Curs\_56, Teme\_Curs\_58, and Teme\_64.
- SQL Worksheet:** History tab is selected. The code in the Query Builder pane is:

```
1254   ELSE
1255     FOR i IN 1..motociclete_utilizator.COUNT LOOP
1256       DBMS_OUTPUT.PUT_LINE(motociclete_utilizator(i));
1257     END LOOP;
1258   END IF;
1259   DBMS_OUTPUT.PUT_LINE('.');
1300
1301   DBMS_OUTPUT.PUT_LINE('Afiseaza informatii despre motocicletele detinute de utilizatorul cu id-ul 1');
1302   dbms_output.put_line(dbms_output.get_line);
1303   DBMS_OUTPUT.PUT_LINE('.');
1304
1305   DBMS_OUTPUT.PUT_LINE('Stocam in nume_accesorii accesoriile motocicletei cu id-ul 1');
1306   nume_accesorii:=dbms_output.get_line;
1307   IF nume_accesorii.COUNT = 0 THEN
1308     DBMS_OUTPUT.PUT_LINE('Nu exista accesoriu stocate de motocicleta cu id=1');
1309   ELSE
1310     FOR i IN 1..nume_accesorii.COUNT LOOP
1311       DBMS_OUTPUT.PUT_LINE(nume_accesorii(i));
1312     END LOOP;
1313   END IF;
1314
1315 END;
```

- Script Output:** Task completed in 0.054 seconds.
- DBMS Output:** Shows the output of the procedure execution, including information about the motorcycle with ID 1 and its accessories.
- Compiler - Log:** No errors or warnings.
- Messages:** No messages.
- Log Page:** No logs.
- Statements:** No statements.
- Compiler:** No compiler errors.

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

- Connections:** Shows various database connections including docker\_project, docker\_pys, Laborator\_1, Laborator2, Laborator3, Laborator4, Laborator5, Laborator6, Laborator7, Laborator8, project\_centerite, Teme\_Curs\_54, Teme\_Curs\_56, Teme\_Curs\_58, and Teme\_64.
- SQL Worksheet:** History tab is selected. The code in the Query Builder pane is identical to the one in the previous screenshot.

```
1254   ELSE
1255     FOR i IN 1..motociclete_utilizator.COUNT LOOP
1256       DBMS_OUTPUT.PUT_LINE(motociclete_utilizator(i));
1257     END LOOP;
1258   END IF;
1259   DBMS_OUTPUT.PUT_LINE('.');
1300
1301   DBMS_OUTPUT.PUT_LINE('Afiseaza informatii despre motocicletele detinute de utilizatorul cu id-ul 1');
1302   dbms_output.put_line(dbms_output.get_line);
1303   DBMS_OUTPUT.PUT_LINE('.');
1304
1305   DBMS_OUTPUT.PUT_LINE('Stocam in nume_accesorii accesoriile motocicletei cu id-ul 1');
1306   nume_accesorii:=dbms_output.get_line;
1307   IF nume_accesorii.COUNT = 0 THEN
1308     DBMS_OUTPUT.PUT_LINE('Nu exista accesoriu stocate de motocicleta cu id=1');
1309   ELSE
1310     FOR i IN 1..nume_accesorii.COUNT LOOP
1311       DBMS_OUTPUT.PUT_LINE(nume_accesorii(i));
1312     END LOOP;
1313   END IF;
1314
1315 END;
```

- Script Output:** Task completed in 0.054 seconds.
- DBMS Output:** Shows the output of the procedure execution, including information about the motorcycle with ID 1 and its accessories.
- Compiler - Log:** No errors or warnings.
- Messages:** No messages.
- Log Page:** No logs.
- Statements:** No statements.
- Compiler:** No compiler errors.

# Sisteme de gestiune a bazelor de date

Stoine Maria-Miruna

Grupa 242

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

- Connections:** Shows several connections including "sterg\_tablee.ad", "create\_project.ad", "project\_cerinte.ad", and "Writeline Page".
- SQL Worksheet:** Displays a PL/SQL block named "proiect\_cerinte". The code uses DBMS\_OUTPUT.PUT\_LINE to print information about motorcycles and accessories.
- Output:** The "DBMS Output" pane shows the results of the procedure execution, including:
  - A motorcycle record: Marca: Honda, Model: CBR600RR, An de fabricatie: 2010, Capacitate rezervorului: 10, Kilometraj: 5000.
  - Information about motorcycle parts:
    - Plates assigned to motorcycle ID 1: Cost total 98.5 lei.
    - Average price per part: 98.5.
    - No maintenance history for this motorcycle.
  - Accessories attached to motorcycle ID 1: Casca (id 5).
  - Accessory ID 5 already attached to motorcycle ID 1.
  - Information about motorcycle parts:
    - Parts assigned to user ID 1: No parts assigned.
    - Parts assigned to user ID 1: No parts assigned.
    - Parts assigned to user ID 1: No parts assigned.
    - Parts assigned to user ID 1: No parts assigned.
    - Parts assigned to user ID 1: No parts assigned.
  - Information about motorcycle parts:
    - Parts assigned to user ID 1: No parts assigned.
    - Parts assigned to user ID 1: No parts assigned.
    - Parts assigned to user ID 1: No parts assigned.
    - Parts assigned to user ID 1: No parts assigned.
    - Parts assigned to user ID 1: No parts assigned.
- Script Output:** Shows the message "PL/SQL procedure successfully completed."
- Compiler - Log:** Shows the log messages: "Saved: C:\Users\Minu\project\_cerinte.ad", "Messages", "Logging Page", "Statements", and "Compiler".