



PROIECT LA DISCIPLINA SGBD ORACLE

PROFESOR COORDONATOR
IULIANA BOTHA

STUDENT
LUNGU VANESA-DENISA

Bucuresti, 2024

CLIENTI

<u>ID_CLIENT</u>	NUME	PRENUME	CNP	ADRESA	NUMAR_DE_TELEFON	EMAIL
NUMBER(6)	VARCHAR2(20)	VARCHAR2(20)	NUMBER(13)	VARCHAR2(50)	NUMBER(10)	VARCHAR2(20)

CONTURI

<u>ID_CONT</u>	<u>ID_CLIENT</u>	TIP_CONT	SOLD	DATA_DESCHIDERII
NUMBER(6)	NUMBER(6)	VARCHAR2(15)	NUMBER(10)	DATE

IMPRUMUTURI

<u>ID_IMPRUMUT</u>	<u>ID_CLIENT</u>	SUMA_IMPRUMUTATA	DATA_ACORDARII	DATA_SCADENTA	RATA DOBANZII	SUMA_RAMBURSATA	STATUS
NUMBER(6)	NUMBER(6)	NUMBER(10)	DATE	DATE	NUMBER(4)	NUMBER(10)	VARCHAR2(15)

TRANZACTII

<u>ID_TRANZACTIE</u>	<u>ID_CONT</u>	TIP_TRANZACTIE	SUMA	DATA_TRANZACTIEI	DESCRIERE
NUMBER(6)	NUMBER(6)	VARCHAR2(10)	NUMBER(10)	DATE	VARCHAR2(30)

ORASE

<u>ID_ORAS</u>	NUME_ORAS
NUMBER(6)	VARCHAR2(15)

DEPARTAMENTE

<u>ID_DEPARTAMENT</u>	<u>ID_ORAS</u>	NUME_DEPARTAMENT
NUMBER(6)	NUMBER(6)	VARCHAR2(50)

ANGAJATI

<u>ID_ANGAJAT</u>	<u>ID_DEPARTAMENT</u>	<u>ID_MANAGER</u>	NUME	PRENUME	SALARIU	DATA_ANGAJARII
NUMBER(6)	NUMBER(6)	NUMBER(6)	VARCHAR2(20)	VARCHAR2(20)	NUMBER(10)	DATE

SUCURSALE



ID_SUCURSALE	ID_ORAS	NUME_SUCURSALA	ADRESA
NUMBER(6)	NUMBER(6)	VARCHAR2(15)	VARCHAR2(40)

Tema proiectului este reprezentata prin schema unei banci, schitata anterior. Am facut aceasta alegere, deoarece, avand in vedere ca tema trebuia sa fie una economica, banca mi s-a parut unul dintre cele mai ample si concrete exemple.

Banca este o instituție financiară care are ca obiect principal de activitate atragerea de depozite și acordarea de credite. Băncile sunt organizate sub forma unei societăți comerciale pe acțiuni și își desfășoară activitatea sub supravegherea băncii centrale.

Centrala este centrul operațional al instituției de credit. Centrala este responsabilă pentru elaborarea politicilor și a strategiilor bancare, coordonarea activităților tuturor sucursalelor, filialelor și agențiilor și pentru asigurarea respectării normelor, regulamentelor și legilor bancare în întreaga organizație. Centrala se ocupă de luarea deciziilor strategice și de administrarea riscurilor la nivelul întregii instituții.

Sucursalele sunt entități bancare care funcționează ca extensii ale centralei în diferite zone geografice. Deși sunt gestionate de centrală, sucursalele operează cu un nivel oarecare de autonomie. Ele oferă majoritatea serviciilor bancare, cum ar fi depuneri, retrageri, împrumuturi și servicii de consultanță financiară, și sunt în contact direct cu clienții instituției de credit.

Filialele sunt entități separate, deși sunt deținute integral sau în proporție majoritară de bancă. Filialele pot oferi servicii similare cu cele ale sucursalelor, dar operează sub propriul lor brand și adesea în sectoare specifice de piață sau în zone geografice unde centrala sau sucursalele nu sunt prezente. Ele au propria lor conducere și pot lua decizii la nivel local, deși direcția strategică și supervizarea generală vin de la centrală.

Agenteiile sunt unități mai mici ale băncii, de obicei situate în zonele mai puțin populate pentru a furniza servicii bancare comunităților locale. Agențiile oferă servicii limitate comparativ cu sucursalele sau filialele, cum ar fi procesarea tranzacțiilor zilnice sau servicii de bază pentru clienți. Sunt gestionate de sucursale sau direct de către centrală, în funcție de modelul organizațional al băncii.

CREAREA SI POPULAREA TABELELOR:

--CREAREA TABELEI CLIENTI

```
CREATE TABLE Clienti(  
id_client number(6),  Nume  
varchar2(20),  
Prenume varchar2(20),  
CNP number(13),  
Adresa varchar2(50),  
Numar_de_telefon number(10),  
Email varchar2(20)  
);  
  
ALTER TABLE Clienti  
ADD CONSTRAINT id_client_pk primary key(id_client); describe  
Clienti;
```

The screenshot displays the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' tab is active, showing a list of commands: 1. --CREAREA TABELEI CLIENTI, 2. CREATE TABLE Clienti..., 3. id_client number(6),, 4. Nume varchar2(20),, 5. Prenume varchar2(20),, 6. CNP number(13),, 7. Adresa varchar2(50),, 8. Numar_de_telefon number(10),, 9. Email varchar2(20),, 10.);, 11. Describe Clienti;, 12. , 13. > ALTER TABLE Clienti..., 14. --CREAREA TABELEI CONTURI.

The 'Describe' tab is selected, showing the table structure for 'CLIENTI'. The table has 7 columns: ID_CLIENT, NUME, PRENUME, CNP, ADRESA, NUMAR_DE_TELEFON, and EMAIL. The 'ID_CLIENT' column is the primary key.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CLIENTI	ID_CLIENT	NUMBER	-	6	0	1	-	-	-
	NUME	VARCHAR2	20	-	-	-	✓	-	-
	PRENUME	VARCHAR2	20	-	-	-	✓	-	-
	CNP	NUMBER	-	13	0	-	✓	-	-
	ADRESA	VARCHAR2	50	-	-	-	✓	-	-
	NUMAR_DE_TELEFON	NUMBER	-	10	0	-	✓	-	-
	EMAIL	VARCHAR2	20	-	-	-	✓	-	-

--CREAREA TABELEI CONTURI

```
CREATE TABLE Conturi(  id_cont  
number(6) primary key,  
id_client number(6),  
  
Tip_cont varchar2(15),  
Sold number(10),  
Data_deschiderii date,  
CONSTRAINT id_client_fk foreign key (id_client) REFERENCES Clienti (id_client)
```

);

The screenshot shows the APEX SQL Workshop interface. The top bar includes the APEX logo, navigation tabs (App Builder, SQL Workshop, Team Development, Gallery), a search bar, and a user profile. The main area is divided into a command editor and a results pane. The command editor shows the following SQL commands:

```
11  
12 > ALTER TABLE Clienti...  
16 --CREAREA TABELEI CONTURI  
17 CREATE TABLE Conturi(  
18     id_cont number(6) primary key,  
19     id_client number(6),  
20     Tip_cont varchar2(15),  
21     Sold number(10),  
22     Data_desciderii date,  
23     CONSTRAINT id_client_fk foreign key (id_client) REFERENCES Clienti (id_client)  
24 );  
25 describe Conturi;  
26 --CREAREA TABELEI IMPRUMUTURI  
27 > CREATE TABLE Imprumuturi(
```

The results pane shows the description of the 'CONTURI' table:

Object Type	TABLE	Object	CONTURI						
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CONTURI	ID_CONT	NUMBER	-	6	0	1	-	-	-
	ID_CLIENT	NUMBER	-	6	0	-	✓	-	-
	TIP_CONT	VARCHAR2	15	-	-	-	✓	-	-
	SOLD	NUMBER	-	10	0	-	✓	-	-
	DATA_DESCIDERII	DATE	7	-	-	-	✓	-	-

--CREAREA TABELEI IMPRUMUTURI

CREATE TABLE Imprumuturi(

id_imprumut number(6), id_client
number(6),

Suma_imprumutata number(10),

Data_acordarii date,

Data_scadenta date,

Rata_dobanzii number(4),

Suma_rambursata number(10),

Status varchar2(15),

CONSTRAINT id_imprumut_pk primary key (id_imprumut),

CONSTRAINT fk_id_client foreign key (id_client) REFERENCES Clienti (id_client)

);

ALTER TABLE Imprumuturi

DROP COLUMN Suma_rambursata;

describe Imprumuturi;

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language: SQL Rows: 10 Clear Command Find Tables Save Run

```

30 Suma_imprumutata number(10),
31 Data_acordarii date,
32 Data_scadenta date,
33 Rata_dobanzii number(4),
34 Suma_rambursata number(10),
35 Status varchar2(15),
36 CONSTRAINT id_imprumut_pk primary key (id_imprumut),
37 CONSTRAINT fk_id_client foreign key (id_client) REFERENCES Clienti (id_client)
38 );
39
40 ALTER TABLE Imprumuturi
41 DROP COLUMN Suma_rambursata;
42 describe Imprumuturi;
43

```

Results Explain Describe Saved SQL History

Object Type: TABLE Object: IMPRUMUTURI

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
IMPRUMUTURI	ID_IMPRUMUT	NUMBER	-	6	0	1	-	-	-
	ID_CLIENT	NUMBER	-	6	0	-	-	-	-
	SUMA_IMPRUMUTATA	NUMBER	-	10	0	-	-	-	-
	DATA_ACORDARII	DATE	7	-	-	-	-	-	-
	DATA_SCADENTA	DATE	7	-	-	-	-	-	-
	RATA DOBANZII	NUMBER	-	4	0	-	-	-	-
	STATUS	VARCHAR2	15	-	-	-	-	-	-

Activate Windows
Go to Settings to activate Windows.

Copyright © 1994, 2023, Oracle and/or its affiliates. Oracle APEX 23.21

--CREAREA TABELEI TRANZACTII

```

CREATE TABLE Tranzactii(
id_tranzactie number(6) primary key,
id_cont number(6),
Tip_tranzactie varchar2(10),
Suma number(10),
Data_tranzactiei date,
Descriere varchar2(30)
);

```

```

ALTER TABLE TRANZACTII
DROP COLUMN TIP_TRANZACTIE;

```

```

ALTER TABLE TRANZACTII
ADD CONSTRAINT id_cont_fk foreign key (id_cont) REFERENCES Conturi (id_cont);
describe Tranzactii;

```

APEX SQL Workshop interface showing SQL commands and results.

```

--CREATE TABLE TRANZACTIE
46 CREATE TABLE TRANZACTIE(
47     id_tranzactie number(6) primary key,
48     id_cont number(6),
49     Tip_tranzactie varchar2(10),
50     Suma number(10),
51     Data_tranzactiei date,
52     Descriere varchar2(100)
53 );
54 ALTER TABLE TRANZACTIE
55 DROP COLUMN TIP_TRANZACTIE;
56
57 ALTER TABLE TRANZACTIE
58 ADD CONSTRAINT id_cont_fk foreign key (id_cont) REFERENCES Conturi (id_cont);
59 describe TRANZACTIE;
60 --CREAREA TABELEI ORASE

```

Results tab showing table structure:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TRANZACTIE	ID_TRANZACTIE	NUMBER	-	6	0	1	-	-	-
	ID_CONT	NUMBER	-	6	0	-	✓	-	-
	SUMA	NUMBER	-	10	0	-	✓	-	-
	DATA_TRANZACTIEI	DATE	7	-	-	-	✓	-	-
	DESCRIERE	VARCHAR2	100	-	-	-	✓	-	-

--CREAREA TABELEI ORASE

```

CREATE TABLE Orase(
    id_oras
    number(6) primary key,
    nume_oras varchar2(15)
);
describe Orase;

```

APEX SQL Workshop interface showing SQL commands and results.

```

--CREAREA TABELEI TRANZACTII
44 > CREATE TABLE TRANZACTII(
45     id_tranzactie number(6) primary key,
46     id_cont number(6),
47     Tip_tranzactie varchar2(10),
48     Suma number(10),
49     Data_tranzactiei date,
50     Descriere varchar2(100)
51 );
52
53
54 > ALTER TABLE TRANZACTII
55 DROP COLUMN TIP_TRANZACTIE;
56
57 > ALTER TABLE TRANZACTII
58 ADD CONSTRAINT id_cont_fk foreign key (id_cont) REFERENCES Conturi (id_cont);
59 describe TRANZACTII;
60
61 --CREAREA TABELEI ORASE
62 CREATE TABLE Orase(
63     id_oras number(6) primary key,
64     nume_oras varchar2(15)
65 );
66 describe Orase;
67
68 --CREAREA TABELEI DEPARTAMENTE
69 CREATE TABLE Departamente(
70     id_departament number(6) primary key,
71     nume_departament varchar2(50)
72 );
73 describe Departamente;

```

Results tab showing table structure:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORASE	ID_ORAS	NUMBER	-	6	0	1	-	-	-
	NUME_ORAS	VARCHAR2	15	-	-	-	✓	-	-

--CREAREA TABELEI DEPARTAMENTE

```

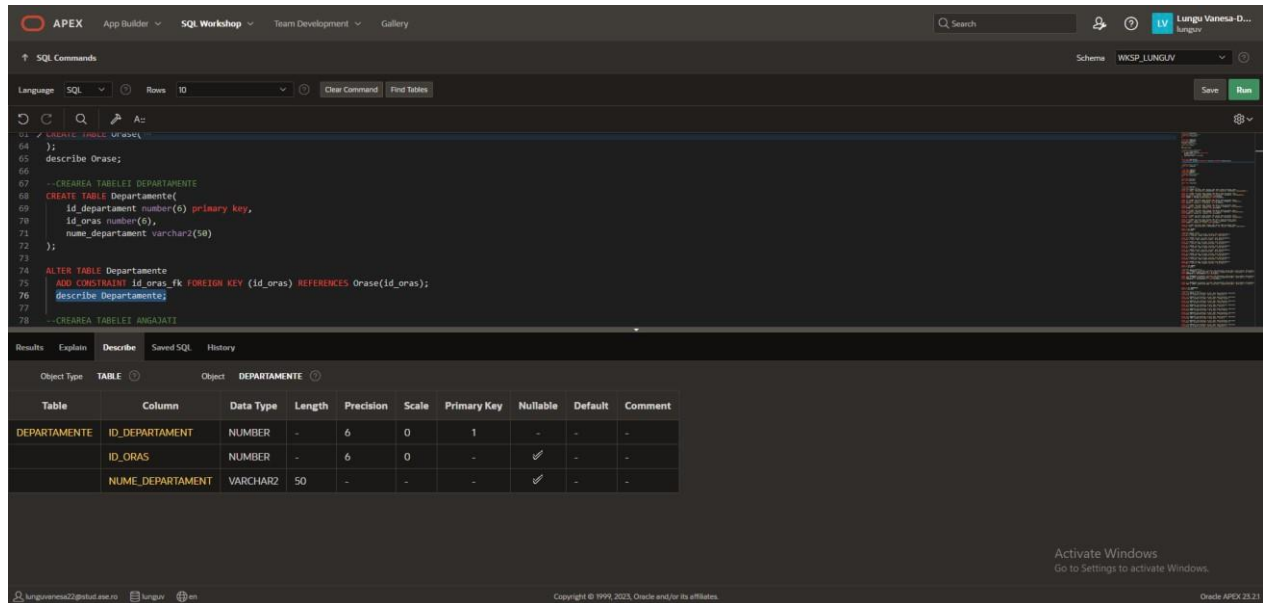
CREATE TABLE Departamente(
    id_departament number(6) primary key,
    id_oras number(6),
    nume_departament
    varchar2(50)
);

```

ALTER TABLE Departamente

ADD CONSTRAINT id_oras_fk FOREIGN KEY (id_oras) REFERENCES Orase(id_oras);

describe Departamente;



--CREAREA TABELEI ANGAJATI

CREATE TABLE Angajati(id_angajat

number(6) primary key,

id_departament number(6),

id_manager number(6), Nume

varchar2(20),

Prenume varchar2(20),

Salariu number(8),

Data_angajarii date

);

ALTER TABLE ANGAJATI

ADD CONSTRAINT id_departament_fk foreign key (id_departament) REFERENCES Departamente (id_departament);

ALTER TABLE ANGAJATI

ADD Varsta number(2);

ALTER TABLE ANGAJATI

ADD CONSTRAINT restrictie_varsta CHECK (Varsta BETWEEN 18 AND 62);

describe Angajati;

The screenshot shows the Oracle APEX SQL Workshop interface. The top bar includes the APEX logo, navigation tabs (App Builder, SQL Workshop, Team Development, Gallery), a search bar, and a user profile. The main area is divided into a SQL editor and a results pane. The SQL editor contains the following commands:

```
-- Create table ANGAJATI
CREATE TABLE ANGAJATI (
  ID_ANGAJATI NUMBER(6) PRIMARY KEY,
  ID_DEPARTAMENT NUMBER(6) NOT NULL,
  ID_MANAGER NUMBER(6) NOT NULL,
  NUME VARCHAR2(20) NOT NULL,
  PRENUME VARCHAR2(20) NOT NULL,
  SALARIU NUMBER(8) NOT NULL,
  DATA_ANGAJARII DATE NOT NULL,
  VARSTA NUMBER(2) NOT NULL
);

-- Add foreign key constraint
ALTER TABLE ANGAJATI
ADD CONSTRAINT fk_id_departament FOREIGN KEY (ID_DEPARTAMENT) REFERENCES DEPARTAMENTE (ID_DEPARTAMENT);

-- Add foreign key constraint
ALTER TABLE ANGAJATI
ADD CONSTRAINT fk_id_manager FOREIGN KEY (ID_MANAGER) REFERENCES ANGAJATI (ID_ANGAJATI);

-- Add check constraint
ALTER TABLE ANGAJATI
ADD CONSTRAINT restrictie_varsta CHECK (Varsta BETWEEN 18 AND 62);

-- Describe the table
DESCRIBE ANGAJATI;
```

The results pane shows the table structure for ANGAJATI:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ANGAJATI	ID_ANGAJATI	NUMBER	-	6	0	1	-	-	-
	ID_DEPARTAMENT	NUMBER	-	6	0	-	✓	-	-
	ID_MANAGER	NUMBER	-	6	0	-	✓	-	-
	NUME	VARCHAR2	20	-	-	-	✓	-	-
	PRENUME	VARCHAR2	20	-	-	-	✓	-	-
	SALARIU	NUMBER	-	8	0	-	✓	-	-
	DATA_ANGAJARII	DATE	7	-	-	-	✓	-	-
	VARSTA	NUMBER	-	2	0	-	✓	-	-

--CREAREA TABELEI SUCURSALE

CREATE TABLE Sucursale(

id_sucursala number(6) primary key,

id_oras number(6),

Nume_sucursala varchar2(15),

Adresa varchar2(40),

CONSTRAINT fk_id_oras FOREIGN KEY (id_oras) REFERENCES Orase (id_oras)

);

ALTER TABLE Sucursale

ADD CONSTRAINT fk_id_oras FOREIGN KEY (id_oras) REFERENCES Orase(id_oras);

DROP TABLE Sucursale;

CREATE TABLE Sucursale(

id_sucursala number(6) primary key,

id_oras number(6),

Nume_sucursala varchar2(15),

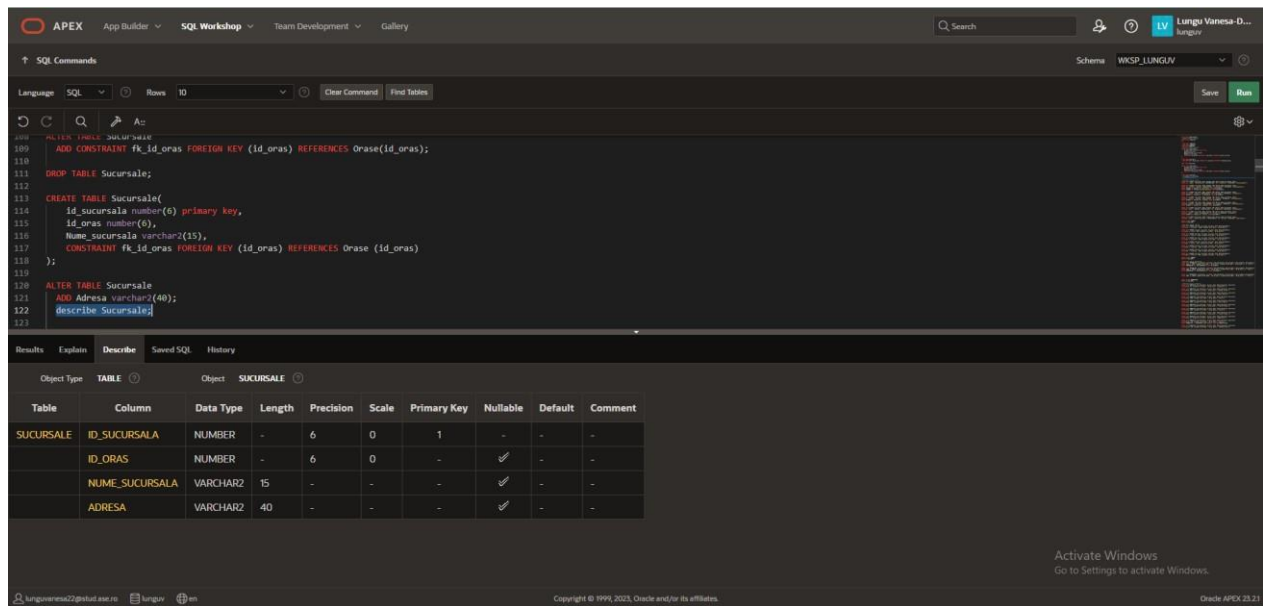
CONSTRAINT fk_id_oras FOREIGN KEY (id_oras) REFERENCES Orase (id_oras)

);

ALTER TABLE Sucursale

ADD Adresa varchar2(40);

describe Sucursale;



The screenshot shows the APEX SQL Workshop interface. The SQL Commands tab is active, displaying a script that creates the Sucursale table and then alters it to add an Adresa column. The Results tab is also visible, showing the table structure for SUCURSALE.

```
208 ALTER TABLE Sucursale
209 ADD CONSTRAINT fk_id_oras FOREIGN KEY (id_oras) REFERENCES Orase(id_oras);
210
211 DROP TABLE Sucursale;
212
213 CREATE TABLE Sucursale(
214   id_sucursala number(6) primary key,
215   id_oras number(6),
216   Nume_sucursala varchar2(15),
217   CONSTRAINT fk_id_oras FOREIGN KEY (id_oras) REFERENCES Orase (id_oras)
218 );
219
220 ALTER TABLE Sucursale
221 ADD Adresa varchar2(40);
222 describe Sucursale;
```

Object Type	TABLE	Object	SUCURSALE						
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SUCURSALE	ID_SUCURSALA	NUMBER	-	6	0	1	-	-	-
	ID_ORAS	NUMBER	-	6	0	-	✓	-	-
	NUME_SUCURSALA	VARCHAR2	15	-	-	-	✓	-	-
	ADRESA	VARCHAR2	40	-	-	-	✓	-	-

--POPULAREA TABELEI CLIENTI

INSERT INTO CLIENTI (id_client, Nume, Prenume, CNP, Adresa, Numar_de_telefon, Email)

VALUES (1, 'Lungu', 'Vanesa-Denisa', 6040105170041, 'Str. Ghiocelilor', 0752884130, 'denisalungu434');

INSERT INTO CLIENTI (id_client, Nume, Prenume, CNP, Adresa, Numar_de_telefon, Email)

VALUES (2, 'Istrate', 'Andreea', 8530162842134, 'Str Batranetii', 0752884131, 'istrateandreea');

UPDATE CLIENTI SET Adresa='Str. Batranetii' WHERE ID_CLIENT=2;

UPDATE CLIENTI SET Prenume='Andreea-Alexandra' WHERE ID_CLIENT=2;

INSERT INTO CLIENTI (id_client, Nume, Prenume, CNP, Adresa, Numar_de_telefon, Email)

VALUES (3, 'Firan', 'Marian', 5926148294145, 'Str 18 Decembrie', 0752884132, 'firanmarian');

UPDATE CLIENTI SET Adresa='Str. 18 Decembrie' WHERE ID_CLIENT=3;

INSERT INTO CLIENTI (id_client, Nume, Prenume, CNP, Adresa, Numar_de_telefon, Email)

VALUES(4, 'Fluturu', 'Andreea', 6925174284718, 'Str Frumoasa', 0752884133, 'fluturuandreea');

UPDATE CLIENTI SET Adresa='Str. Frumoasa' WHERE ID_CLIENT=4;

INSERT INTO CLIENTI (id_client, Nume, Prenume, CNP, Adresa, Numar_de_telefon, Email)

```
VALUES(5, 'Fluturu', 'Patrisia', 7291752956247, 'Str Frumoasa', 0728519274, 'fluturupatrisia'); UPDATE
CLIENTI SET Adresa='Str. Frumoasa' WHERE ID_CLIENT=5;
```

```
INSERT INTO CLIENTI (id_client, Nume, Prenume, CNP, Adresa, Numar_de_telefon, Email)
VALUES(6, 'Fuica', 'Andreea', 5195274915284, 'Str. Margeanului', 0751927539, 'fuicaandreea');
```

```
INSERT INTO CLIENTI (id_client, Nume, Prenume, CNP, Adresa, Numar_de_telefon, Email)
VALUES(7, 'Fuica', 'Diana', 8192548164294, 'Str Frunzei', 0715294629, 'fuicadiana'); UPDATE
CLIENTI SET Adresa='Str. Frunzei' WHERE ID_CLIENT=7;
```

```
INSERT INTO CLIENTI (id_client, Nume, Prenume, CNP, Adresa, Numar_de_telefon, Email)
VALUES(8, 'Tudor', 'Razvan-Constantin', 5927149265172, 'Str. Salciei', 0724678123, 'tudorrazvan');
```

```
SELECT * FROM CLIENTI
ORDER BY ID_CLIENT;
```

The screenshot shows the APEX SQL Workshop interface. The SQL Commands pane contains the following code:

```
150 UPDATE CLIENTI SET Adresa='Str. Frunzei' WHERE ID_CLIENT=7;
151
152 INSERT INTO CLIENTI (id_client, Nume, Prenume, CNP, Adresa, Numar_de_telefon, Email)
153 VALUES(8, 'Tudor', 'Razvan-Constantin', 5927149265172, 'Str. Salciei', 0724678123, 'tudorrazvan');
154
155 SELECT * FROM CLIENTI
156 ORDER BY ID_CLIENT;
157
158 --POPULAREA TABELEI CONTURI
159 INSERT INTO CONTURI (id_cont, id_client, Tip_cont, Sold, Data_deschiderii)
160 VALUES(10, 1, 'Economii', 12000, to_date('12.01.2022', 'dd.mm.yyyy'));
161
162 INSERT INTO CONTURI (id_cont, id_client, Tip_cont, Sold, Data_deschiderii)
163 VALUES(20, 2, 'Bursa', 900, to_date('17.11.2023', 'dd.mm.yyyy'));
164
165 INSERT INTO CONTURI (id_cont, id_client, Tip_cont, Sold, Data_deschiderii)
```

The Results pane shows the output of the SQL commands. The first command updates the address of client 7. The second command inserts a new client. The third command inserts a new account. The fourth command inserts a new account. The fifth command inserts a new account. The Results pane shows the output of the SQL commands. The first command updates the address of client 7. The second command inserts a new client. The third command inserts a new account. The fourth command inserts a new account. The fifth command inserts a new account.

ID_CLIENT	NUME	PRENUME	CNP	ADRESA	NUMAR_DE_TELEFON	EMAIL
1	Lungu	Vanessa-Denisa	6040105170041	Str. Ghiocelilor	752884130	denisalungu434
2	Istrate	Andreea-Alexandra	8530162842134	Str. Batranetii	752884131	istrateandreea
3	Firan	Marian	5926148294045	Str. 18 Decembrie	752884132	firanmarian
4	Fluturu	Andreea	69257428478	Str. Frumoasa	752884133	fluturuandreea
5	Fluturu	Patrisia	7291752956247	Str. Frumoasa	728519274	fluturupatrisia
6	Fuica	Andreea	5195274915284	Str. Margeanului	751927539	fuicaandreea
7	Fuica	Diana	8192548164294	Str. Frunzei	715294629	fuicadiana
8	Tudor	Razvan-Constantin	5927149265172	Str. Salciei	724678123	tudorrazvan

--POPULAREA TABELEI CONTURI

```
INSERT INTO CONTURI (id_cont, id_client, Tip_cont, Sold, Data_deschiderii)
VALUES(10, 1, 'Economii', 12000, to_date('12.01.2022', 'dd.mm.yyyy'));
```

```
INSERT INTO CONTURI (id_cont, id_client, Tip_cont, Sold, Data_deschiderii) VALUES(20,
2, 'Bursa', 900, to_date('17.11.2023', 'dd.mm.yyyy'));
```

```
INSERT INTO CONTURI (id_cont, id_client, Tip_cont, Sold, Data_deschiderii) VALUES(30,
3, 'Salariu', 8200, to_date('19.05.2018', 'dd.mm.yyyy'));
```

```
INSERT INTO CONTURI (id_cont, id_client, Tip_cont, Sold, Data_deschiderii) VALUES(40,
4, 'Economii', 4800, to_date('13.07.2020', 'dd.mm.yyyy'));
```

```
INSERT INTO CONTURI (id_cont, id_client, Tip_cont, Sold, Data_deschiderii) VALUES(50,
5, 'Bursa', 730, to_date('20.01.2018', 'dd.mm.yyyy'));
```

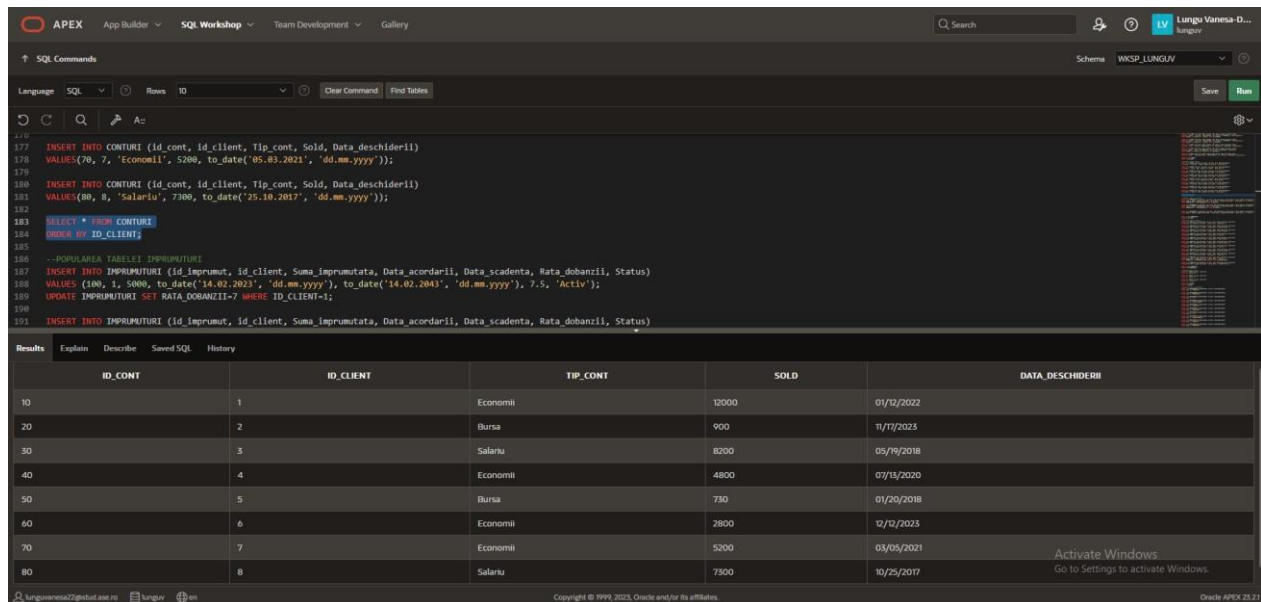
```
INSERT INTO CONTURI (id_cont, id_client, Tip_cont, Sold, Data_deschiderii)
VALUES(60, 6, 'Economii', 2800, to_date('12.12.2023', 'dd.mm.yyyy'));
```

```
INSERT INTO CONTURI (id_cont, id_client, Tip_cont, Sold, Data_deschiderii) VALUES(70,
7, 'Economii', 5200, to_date('05.03.2021', 'dd.mm.yyyy'));
```

```
INSERT INTO CONTURI (id_cont, id_client, Tip_cont, Sold, Data_deschiderii) VALUES(80,
8, 'Salariu', 7300, to_date('25.10.2017', 'dd.mm.yyyy'));
```

```
SELECT * FROM CONTURI
```

```
ORDER BY ID_CLIENT;
```



The screenshot shows the APEX SQL Workshop interface. The SQL Commands pane contains the following code:

```

177 INSERT INTO CONTURI (id_cont, id_client, Tip_cont, Sold, Data_deschiderii)
178 VALUES(70, 7, 'Economii', 5200, to_date('05-03-2021', 'dd.mm.yyyy'));
179
180 INSERT INTO CONTURI (id_cont, id_client, Tip_cont, Sold, Data_deschiderii)
181 VALUES(80, 8, 'Salariu', 7300, to_date('25.10.2017', 'dd.mm.yyyy'));
182
183 SELECT * FROM CONTURI
184 ORDER BY ID_CLIENT;
185
--POPULAREA TABELEI IMPRUMUTURI
187 INSERT INTO IMPRUMUTURI (id_imprumut, id_client, Suma_imprumutata, Data_acordarii, Data_scadenta, Rata_dobanzii, Status)
188 VALUES (100, 1, 5000, to_date('14-02-2021', 'dd.mm.yyyy'), to_date('14-02-2043', 'dd.mm.yyyy'), 7.5, 'Activ');
189 UPDATE IMPRUMUTURI SET RATA_DOBANZII=7 WHERE ID_CLIENT=1;
190
191 INSERT INTO IMPRUMUTURI (id_imprumut, id_client, Suma_imprumutata, Data_acordarii, Data_scadenta, Rata_dobanzii, Status)

```

The Results pane shows the output of the SELECT statement:

ID_CONT	ID_CLIENT	TIP_CONT	SOLD	DATA_DESCHIDERII
10	1	Economii	10000	01/12/2022
20	2	Bursa	900	11/11/2023
30	3	Salariu	8200	05/19/2018
40	4	Economii	4800	07/13/2020
50	5	Bursa	730	01/20/2018
60	6	Economii	2800	12/12/2023
70	7	Economii	5200	03/05/2021
80	8	Salariu	7300	10/25/2017

--POPULAREA TABELEI IMPRUMUTURI

```
INSERT INTO IMPRUMUTURI (id_imprumut, id_client, Suma_imprumutata, Data_acordarii,
Data_scadenta, Rata_dobanzii, Status)
```

```
VALUES (100, 1, 5000, to_date('14.02.2023', 'dd.mm.yyyy'), to_date('14.02.2043', 'dd.mm.yyyy'), 7.5, 'Activ');
```

```
UPDATE IMPRUMUTURI SET RATA_DOBANZII=7 WHERE ID_CLIENT=1;
```

```
INSERT INTO IMPRUMUTURI (id_imprumut, id_client, Suma_imprumutata, Data_acordarii, Data_scadenta, Rata_dobanzii, Status)
```

```
VALUES (300, 3, 30000, to_date('09.11.2021', 'dd.mm.yyyy'), to_date('09.11.2051', 'dd.mm.yyyy'), 8, 'Activ');
```

```
UPDATE IMPRUMUTURI SET RATA_DOBANZII=NULL WHERE ID_CLIENT=3;
```

```
INSERT INTO IMPRUMUTURI (id_imprumut, id_client, Suma_imprumutata, Data_acordarii, Data_scadenta, Rata_dobanzii, Status)
```

```
VALUES (800, 8, 24000, to_date('20.05.2020', 'dd.mm.yyyy'), to_date('20.05.2048', 'dd.mm.yyyy'), 7, 'Activ');
```

```
SELECT * FROM IMPRUMUTURI
```

```
ORDER BY ID_CLIENT;
```

The screenshot shows the APEX SQL Workshop interface. The SQL Commands pane contains the following code:

```
191 INSERT INTO IMPRUMUTURI (id_imprumut, id_client, Suma_imprumutata, Data_acordarii, Data_scadenta, Rata_dobanzii, Status)
192 VALUES (200, 2, 30000, to_date('09.11.2021', 'dd.mm.yyyy'), to_date('09.11.2051', 'dd.mm.yyyy'), 8, 'Activ');
193 UPDATE IMPRUMUTURI SET RATA_DOBANZII=NULL WHERE ID_CLIENT=3;
194
195
196 INSERT INTO IMPRUMUTURI (id_imprumut, id_client, Suma_imprumutata, Data_acordarii, Data_scadenta, Rata_dobanzii, Status)
197 VALUES (800, 8, 24000, to_date('20.05.2020', 'dd.mm.yyyy'), to_date('20.05.2048', 'dd.mm.yyyy'), 7, 'Activ');
198
199
200 SELECT * FROM IMPRUMUTURI
201 ORDER BY ID_CLIENT;
202
203 --POPULAREA TABELEI TRANZACTII
204 INSERT INTO TRANZACTII (id_tranzactie, id_cont, Suma, Data_tranzactiei, Descriere)
205 VALUES(1,10, 20, to_date('12.12.2023', 'dd.mm.yyyy'), 'Cumparaturi');
```

The Results pane shows the output of the SELECT statement:

ID_IMPRUMUT	ID_CLIENT	SUMA_IMPRUMUTATA	DATA_ACORDARII	DATA_SCADENTA	RATA_DOBANZII	STATUS
100	1	5000	02/14/2023	02/14/2043	7.5	Activ
300	3	30000	11/09/2021	11/09/2051	-	Activ
800	8	24000	05/20/2020	05/20/2048	7	Activ

3 rows returned in 0.03 seconds

--POPULAREA TABELEI TRANZACTII

```
INSERT INTO TRANZACTII (id_tranzactie, id_cont, Suma, Data_tranzactiei, Descriere) VALUES(1,10, 20, to_date('12.12.2023', 'dd.mm.yyyy'), 'Cumparaturi');
```

```
INSERT INTO TRANZACTII (id_tranzactie, id_cont, Suma, Data_tranzactiei, Descriere)
```

```
VALUES(2,10, 500, to_date('11.12.2023', 'dd.mm.yyyy'), 'Comanda online');
```

```
INSERT INTO TRANZACTII (id_tranzactie, id_cont, Suma, Data_tranzactiei, Descriere)
VALUES(3,10, 150, to_date('12.12.2023', 'dd.mm.yyyy'), 'Plata factura');
```

```
INSERT INTO TRANZACTII (id_tranzactie, id_cont, Suma, Data_tranzactiei, Descriere)
VALUES(4,30, 1500, to_date('20.11.2023', 'dd.mm.yyyy'), 'Chirie');
```

```
INSERT INTO TRANZACTII (id_tranzactie, id_cont, Suma, Data_tranzactiei, Descriere)
VALUES(5,30, 2000, to_date('01.12.2023', 'dd.mm.yyyy'), 'Rata credit');
```

```
INSERT INTO TRANZACTII (id_tranzactie, id_cont, Suma, Data_tranzactiei, Descriere)
VALUES(6,50, 300, to_date('08.12.2023', 'dd.mm.yyyy'), 'Transfer bancar');
```

```
INSERT INTO TRANZACTII (id_tranzactie, id_cont, Suma, Data_tranzactiei, Descriere)
VALUES(7,80, 1000, to_date('04.12.2023', 'dd.mm.yyyy'), 'Chirie');
```

```
INSERT INTO TRANZACTII (id_tranzactie, id_cont, Suma, Data_tranzactiei, Descriere)
VALUES(8,80, 500, to_date('15.05.2020', 'dd.mm.yyyy'), 'Transfer bancar');
```

```
INSERT INTO TRANZACTII (id_tranzactie, id_cont, Suma, Data_tranzactiei, Descriere)
VALUES(9,60, 86, to_date('12.12.2023', 'dd.mm.yyyy'), 'Cumparaturi');
```

```
INSERT INTO TRANZACTII (id_tranzactie, id_cont, Suma, Data_tranzactiei, Descriere)
VALUES(10,20, 300, to_date('10.10.2023', 'dd.mm.yyyy'), 'Facturi');
```

```
UPDATE TRANZACTII SET DESCRIERE='Plata factura' WHERE ID_TRANZACTIE=10;
```

```
INSERT INTO TRANZACTII (id_tranzactie, id_cont, Suma, Data_tranzactiei, Descriere) VALUES(11,70, 330,
to_date('05.06.2023', 'dd.mm.yyyy'), 'Comanda online');
```

```
SELECT * FROM TRANZACTII
ORDER BY ID_TRANZACTIE;
```

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language: SQL Rows: 10

```

231 INSERT INTO TRANZACTII (id_tranzactie, id_cont, suma, data_tranzactiei, descriere)
232 VALUES(10,20, 300, to_date('10.10.2023', 'dd.mm.yyyy'), 'Facturi');
233 UPDATE TRANZACTII SET DESCRIERE='Plata factura' WHERE ID_TRANZACTIE=10;
234
235 INSERT INTO TRANZACTII (id_tranzactie, id_cont, suma, data_tranzactiei, descriere)
236 VALUES(11,70, 110, to_date('05.06.2023', 'dd.mm.yyyy'), 'Comanda online');
237
238 SELECT * FROM TRANZACTII
239 ORDER BY ID_TRANZACTIE;
240
241 --POPULAREA TABELEI ORASE
242 INSERT INTO ORASE (id_oras, nume_oras)
243 VALUES (1, 'Bucuresti');
244
245 INSERT INTO ORASE (id_oras, nume_oras)

```

Results Explain Describe Saved SQL History

ID_TRANZACTIE	ID_CONT	SUMA	DATA_TRANZACTIEI	DESCRIERE
1	10	20	12/12/2023	Comparaturi
2	10	500	12/11/2023	Comanda online
3	10	150	12/12/2023	Plata factura
4	30	1500	11/20/2023	Chirie
5	30	2000	12/01/2023	Rata credit
6	50	300	12/08/2023	Transfer bancar
7	80	3000	12/04/2023	Chirie
8	80	500	05/15/2020	Transfer bancar

Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.1

--POPULAREA TABELEI ORASE

INSERT INTO ORASE (id_oras, nume_oras)

VALUES (1, 'Bucuresti');

INSERT INTO ORASE (id_oras, nume_oras)

VALUES (2, 'Galati');

INSERT INTO ORASE (id_oras, nume_oras)

VALUES (3, 'Craiova');

SELECT * FROM ORASE

ORDER BY ID_ORAS;

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language: SQL Rows: 10

```

246 VALUES (2, 'Galati');
247
248 INSERT INTO ORASE (id_oras, nume_oras)
249 VALUES (3, 'Craiova');
250
251 SELECT * FROM ORASE
252 ORDER BY ID_ORAS;
253
254 --POPULAREA TABELEI DEPARTAMENTE
255 INSERT INTO DEPARTAMENTE(id_departament, id_oras, nume_departament)
256 VALUES (110, 1, 'FINANTE');
257
258 INSERT INTO DEPARTAMENTE(id_departament, id_oras, nume_departament)
259 VALUES (120, 1, 'MARKETING');
260
261 INSERT INTO DEPARTAMENTE(id_departament, id_oras, nume_departament)

```

Results Explain Describe Saved SQL History

ID_ORAS	NUME_ORAS
1	Bucuresti
2	Galati
3	Craiova

3 rows returned in 0.02 seconds Download

Activate Windows
Go to Settings to activate Windows.

Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.1

--POPULAREA TABELEI DEPARTAMENTE

```
INSERT INTO DEPARTAMENTE(id_departament, id_oras, nume_departament) VALUES  
(110, 1, 'FINANTE');
```

```
INSERT INTO DEPARTAMENTE(id_departament, id_oras, nume_departament) VALUES  
(120, 1, 'MARKETING');
```

```
INSERT INTO DEPARTAMENTE(id_departament, id_oras, nume_departament)  
VALUES (130, 1, 'IT');
```

```
INSERT INTO DEPARTAMENTE(id_departament, id_oras, nume_departament) VALUES  
(140, 1, 'RESURSE UMANE');
```

```
INSERT INTO DEPARTAMENTE(id_departament, id_oras, nume_departament) VALUES  
(210, 2, 'FINANTE');
```

```
INSERT INTO DEPARTAMENTE(id_departament, id_oras, nume_departament)  
VALUES (220, 2, 'MARKETING');
```

```
INSERT INTO DEPARTAMENTE(id_departament, id_oras, nume_departament)  
VALUES (230, 2, 'IT');
```

```
INSERT INTO DEPARTAMENTE(id_departament, id_oras, nume_departament)  
VALUES (240, 2, 'RESURSE UMANE');
```

```
INSERT INTO DEPARTAMENTE(id_departament, id_oras, nume_departament)  
VALUES (310, 3, 'FINANTE');
```

```
INSERT INTO DEPARTAMENTE(id_departament, id_oras, nume_departament)  
VALUES (320, 3, 'MARKETING');
```

```
INSERT INTO DEPARTAMENTE(id_departament, id_oras, nume_departament)
```


VALUES (330, 3, 'IT');

INSERT INTO DEPARTAMENTE(id_departament, id_oras, nume_departament)

VALUES (340, 3, 'RESURSE UMANE');

SELECT * FROM DEPARTAMENTE

ORDER BY ID_DEPARTAMENT;

The screenshot shows the APEX SQL Workshop interface. The SQL Commands pane contains the following code:

```
-- POPULAREA TABELEI ANGAJATI
282 INSERT INTO DEPARTAMENTE(id_departament, id_oras, nume_departament)
283 VALUES (320, 3, 'MARKETING');
284
285 INSERT INTO DEPARTAMENTE(id_departament, id_oras, nume_departament)
286 VALUES (330, 3, 'IT');
287
288 INSERT INTO DEPARTAMENTE(id_departament, id_oras, nume_departament)
289 VALUES (340, 3, 'RESURSE UMANE');
290
291 SELECT * FROM DEPARTAMENTE
292 ORDER BY ID_DEPARTAMENT;
293
-- POPULAREA TABELEI ANGAJATI
294 INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Salariu, Data_angajarii)
295 VALUES (1, 110, 100, 'Lungu', 'Dragos', 8000, to_date('11.02.2020', 'dd.mm.yyyy'));
```

The Results pane shows the output of the SELECT statement:

ID_DEPARTAMENT	ID_ORAS	NUME_DEPARTAMENT
110	1	FINANTE
120	1	MARKETING
130	1	IT
140	1	RESURSE UMANE
210	2	FINANTE
220	2	MARKETING
230	2	IT
240	2	RESURSE UMANE

--POPULAREA TABELEI ANGAJATI

INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Salariu, Data_angajarii)

VALUES (1, 110, 100, 'Lungu', 'Dragos', 8000, to_date('11.02.2020', 'dd.mm.yyyy')); UPDATE
ANGAJATI SET VARSTA=24 WHERE ID_ANGAJAT=1;

INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Salariu, Data_angajarii)

VALUES (100, 110, NULL, 'Popescu', 'Ana', 7000, to_date('12.02.2018', 'dd.mm.yyyy')); UPDATE
ANGAJATI SET VARSTA=40 WHERE ID_ANGAJAT=100;

INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)

VALUES(2, 120, 100, 'Ionescu', 'Mihai', 35, 9000, to_date('12.02.2018', 'dd.mm.yyyy'));

INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)

VALUES(3, 130, 1, 'Popa', 'Ioana',40, 7500, to_date('13.02.2010', 'dd.mm.yyyy'));

INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)

VALUES(4, 140, 1, 'Radu', 'Elena',41, 8200, to_date('14.02.2015', 'dd.mm.yyyy'));

INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)

VALUES(5, 210, 3, 'Dumitru', 'Andrei',40, 8500, to_date('15.02.2011', 'dd.mm.yyyy'));

INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)

VALUES(6, 220, 100, 'Stanescu', 'Maria',35, 7800, to_date('16.02.2011', 'dd.mm.yyyy'));

INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)

VALUES(7, 230, 2, 'Moldovan', 'Cristina',32, 9200, to_date('17.02.2013', 'dd.mm.yyyy'));

INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)

VALUES(8, 240, 100, 'Dinu', 'Alexandru',34, 8700, to_date('18.02.2014', 'dd.mm.yyyy'));

INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)

VALUES(9, 310, 5, 'Gheorghiu', 'Laura',33, 8000, to_date('19.02.2015', 'dd.mm.yyyy'));

INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)

VALUES(10, 320, 100, 'Antonescu', 'Robert',25, 9300, to_date('20.02.2020', 'dd.mm.yyyy'));

INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)

VALUES(11, 330, 5, 'Mihai', 'Andreea',23, 7600, to_date('21.02.2022', 'dd.mm.yyyy'));

INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)

VALUES(12, 340, 2, 'Georgescu', 'Alex',21, 8800, to_date('22.02.2023', 'dd.mm.yyyy'));

```
INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)
```

```
VALUES(13, 110, 4, 'Florescu', 'Diana',28, 9200, to_date('23.02.2018', 'dd.mm.yyyy'));
```

```
INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)
```

```
VALUES(14, 120, 100, 'Dobre', 'Gabriel',29, 7900, to_date('24.02.2017', 'dd.mm.yyyy'));
```

```
INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)
```

```
VALUES(15, 130, 12, 'Stanciu', 'Elena',38, 8700, to_date('25.02.2012', 'dd.mm.yyyy'));
```

```
INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)
```

```
VALUES(16, 140, 100, 'Ilie', 'Ana-Maria',45, 9400, to_date('26.02.2010', 'dd.mm.yyyy'));
```

```
INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)
```

```
VALUES(17, 210, 1, 'Balan', 'Ionut',42, 8100, to_date('27.02.2015', 'dd.mm.yyyy'));
```

```
INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)
```

```
VALUES(18, 220, 2, 'Cristea', 'Teodora',29, 8700, to_date('28.02.2018', 'dd.mm.yyyy'));
```

```
INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)
```

```
VALUES(19, 230, 5, 'Serban', 'Raluca',36, 8000, to_date('29.02.2016', 'dd.mm.yyyy'));
```

```
INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)
```

```
VALUES(20, 240, 12, 'Anton', 'Adrian',30, 9100, to_date('01.03.2016', 'dd.mm.yyyy'));
```

```
INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)
```

```
VALUES(21, 310, 100, 'Tudor', 'Madalina',49, 7800, to_date('02.03.2013', 'dd.mm.yyyy'));
```

```
INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)
```

```
VALUES(22, 320, 15, 'Costin', 'Sorin',50, 9300, to_date('03.03.2012', 'dd.mm.yyyy'));
```

```
INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)
```

```
VALUES(23, 330, 12, 'Gavrilă', 'Catalina',55, 8600, to_date('04.03.2020', 'dd.mm.yyyy'));
```

```
INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)
```

```
VALUES(24, 340, 1, 'Mocanu', 'Alina',20, 8900, to_date('05.03.2023', 'dd.mm.yyyy'));
```

```
SELECT * FROM ANGAJATI
```

```
ORDER BY ID_ANGAJAT;
```

The screenshot displays the APEX SQL Workshop interface. The top section shows the SQL Commands editor with the following code:

```
368 INSERT INTO ANGAJATI (id_angajat, id_departament, id_manager, Nume, Prenume, Varsta, Salariu, Data_angajarii)
369 VALUES(24, 340, 1, 'Mocanu', 'Alina',20, 8900, to_date('05.03.2023', 'dd.mm.yyyy'));
370
371
372 SELECT * FROM ANGAJATI
373 ORDER BY ID_ANGAJAT;
374
375 --POPULAREA TABELEI SUCURSALE
376 INSERT INTO SUCURSALE (id_sucursala, id_oras, Nume_sucursala)
377 VALUES(11,1,'BT CENTRU');
378 UPDATE SUCURSALE SET NUME_SUCURSALA='BT UNIRII' WHERE ID_SUCURSALA=11;
379 UPDATE SUCURSALE SET ADRESA='BLD UNIRII 25' WHERE ID_SUCURSALA=11;
380
381 INSERT INTO SUCURSALE (id_sucursala, id_oras, Nume_sucursala)
382 VALUES(12,1,'BT UNIVERSITATE');
383 UPDATE SUCURSALE SET ADRESA='BLD REGINA ELISABETA 16' WHERE ID_SUCURSALA=12;
```

The bottom section shows the Results tab with a table of data:

ID_ANGAJAT	ID_DEPARTAMENT	ID_MANAGER	NUME	PRENUME	SALARIU	DATA_ANGAJARI	VARSTA
1	110	100	Lungu	Dragos	8000	02/11/2020	24
2	120	100	Ionescu	Mihai	9000	02/12/2018	35
3	150	1	Popa	Ioana	7500	02/13/2010	40
4	140	1	Radu	Elena	8200	02/14/2015	41
5	210	3	Dumitru	Andrei	8500	02/15/2011	40
6	220	100	Stancu	Maria	7800	02/16/2011	35
7	230	2	Moldovan	Cristina	9200	02/17/2013	
8	240	100	Dimu	Alexandru	8700	02/18/2014	34

--POPULAREA TABELEI SUCURSALE

```
INSERT INTO SUCURSALE (id_sucursala, id_oras, Nume_sucursala)
```

```
VALUES(11,1,'BT CENTRU');
```

```
UPDATE SUCURSALE SET NUME_SUCURSALA='BT UNIRII' WHERE ID_SUCURSALA=11; UPDATE
SUCURSALE SET ADRESA='BLD UNIRII 25' WHERE ID_SUCURSALA=11;
```

```
INSERT INTO SUCURSALE (id_sucursala, id_oras, Nume_sucursala)
```

```
VALUES(12,1,'BT UNIVERSITATE');
```

```
UPDATE SUCURSALE SET ADRESA='BLD REGINA ELISABETA 16' WHERE ID_SUCURSALA=12;
```

```
INSERT INTO SUCURSALE (id_sucursala, id_oras, Nume_sucursala)
VALUES(13,1,'BT RAHOVA');
UPDATE SUCURSALE SET ADRESA='SOS ALEXANDRIEI 16' WHERE ID_SUCURSALA=13;
```

```
INSERT INTO SUCURSALE (id_sucursala, id_oras, Nume_sucursala)
VALUES(14,2,'BT BRAILEI');
UPDATE SUCURSALE SET ADRESA='STRADA BRAILEI 152' WHERE ID_SUCURSALA=14;
```

```
INSERT INTO SUCURSALE (id_sucursala, id_oras, Nume_sucursala)
VALUES(15,2,'BT MICRO 18');
UPDATE SUCURSALE SET ADRESA='STRADA BRAILEI 192' WHERE ID_SUCURSALA=15;
```

```
INSERT INTO SUCURSALE (id_sucursala, id_oras, Nume_sucursala)
VALUES(16,2,'BT MAZEPA');
UPDATE SUCURSALE SET ADRESA='STRADA BRAILEI 132' WHERE ID_SUCURSALA=16;
```

```
INSERT INTO SUCURSALE (id_sucursala, id_oras, Nume_sucursala)
VALUES(17,2,'BT MICRO 14');
UPDATE SUCURSALE SET ADRESA='STRADA BRAILEI 200' WHERE ID_SUCURSALA=17;
```

```
INSERT INTO SUCURSALE (id_sucursala, id_oras, Nume_sucursala)
VALUES(18,3,'BT ELECTROPTR');
UPDATE SUCURSALE SET ADRESA='STR. IMPARATUL TRAIAN 219' WHERE ID_SUCURSALA=18;
```

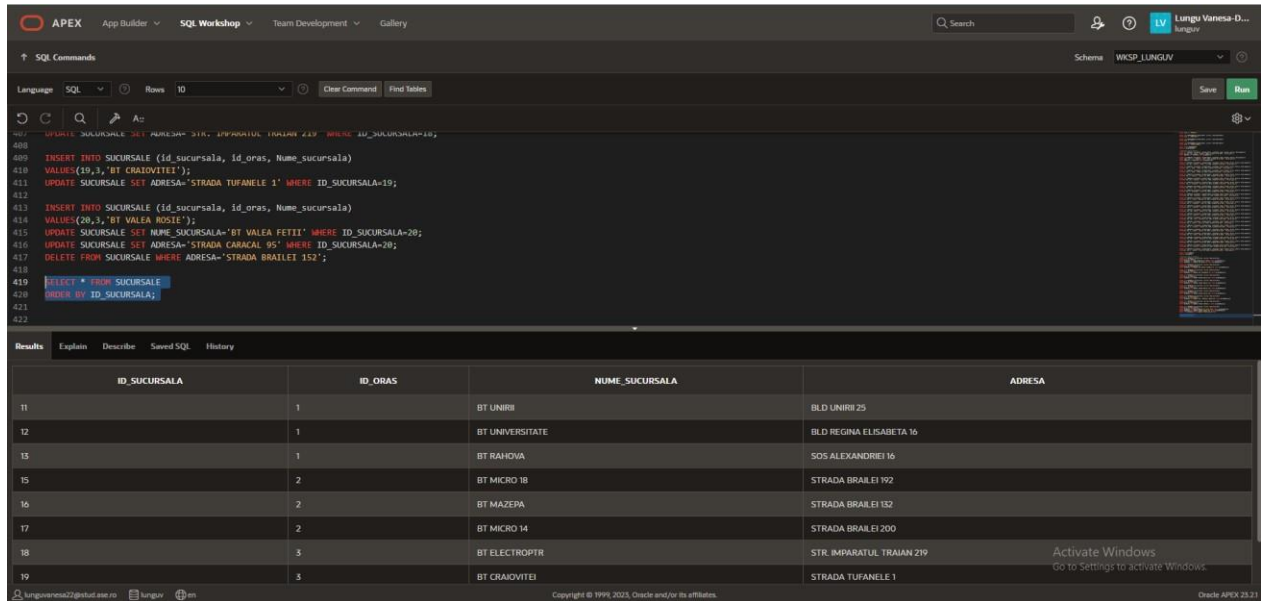
```
INSERT INTO SUCURSALE (id_sucursala, id_oras, Nume_sucursala)
VALUES(19,3,'BT CRAIOVITEI');
UPDATE SUCURSALE SET ADRESA='STRADA TUFANELE 1' WHERE ID_SUCURSALA=19;
```

```
INSERT INTO SUCURSALE (id_sucursala, id_oras, Nume_sucursala)
VALUES(20,3,'BT VALEA ROSIE');
UPDATE SUCURSALE SET NUME_SUCURSALA='BT VALEA FETII' WHERE ID_SUCURSALA=20;
UPDATE SUCURSALE SET ADRESA='STRADA CARACAL 95' WHERE ID_SUCURSALA=20;
```

DELETE FROM SUCURSALE WHERE ADRESA='STRADA BRAILEI 152';

SELECT * FROM SUCURSALE

ORDER BY ID_SUCURSALA;



INTEROGARI PENTRU TABELELE CREATE:

Pentru a face posibile interogari, am facut urmatoarele modificari asupra bazei de date:

UPDATE ANGAJATI SET id_angajat=25 WHERE ID_ANGAJAT=100;

UPDATE ANGAJATI SET data_angajarii=to_date('13.02.2018', 'dd.mm.yyyy') where id_angajat=2;

Delete from Imprumuturi where id_client=8;

INSERT INTO IMPRUMUTURI (id_imprumut, id_client, Suma_imprumutata, Data_acordarii, Data_scadenta, Rata_dobanzii, Status)

VALUES (500, 5, 24000, to_date('20.05.2020', 'dd.mm.yyyy'), to_date('20.05.2048', 'dd.mm.yyyy'), 8, 'Activ');

-----STRUCTURI DE CONTROL-----

Ex 1:

Verificati daca fiecare angajat este eligibil pentru promovare în funcție de vechimea sa în companie și de salariu (cel puțin 5 ani de vechime si un salariu mai mare de 5000 lei).

DECLARE

v_nume angajati.nume %type; v_prenume angajati.prenume %type; v_salariu angajati.salariu %type;

v_data_angajarii angajati.data_angajarii %type; v_vechime number(3); v_min number (3); v_max number(3); i number(3);

BEGIN

SELECT MIN(ID_ANGAJAT), MAX(ID_ANGAJAT) INTO v_min, v_max FROM angajati; i:=v_min;

WHILE i<=v_max LOOP

```

SELECT nume, prenume, salariu, data_angajarii, (sysdate-data_angajarii)/365 INTO v_nume,
v_prenume, v_salariu, v_data_angajarii, v_vechime FROM angajati where id_angajat=i;
IF v_vechime>=5 AND v_salariu>5000 THEN dbms_output.put_line('Angajatul '||v_nume||'
'||v_prenume||' are o vechime de '||v_vechime||' ani, si un salariu de '||v_salariu||' lei, deci este
eligibil pentru o promovare.');
```

```

ELSE dbms_output.put_line('Angajatul '||v_nume||' '||v_prenume||' are o vechime de
'||v_vechime||' ani, si un salariu de '||v_salariu||' lei, deci nu este eligibil pentru o promovare.');
```

```

END IF; i:=i+1;

END LOOP;

END;
```

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Workshop' tab is active, showing a SQL command window with a PL/SQL script. The script is a procedure that iterates through employees, checking their tenure and salary to determine if they are eligible for a promotion. The script includes variable declarations, a loop to fetch employee data, and conditional logic to output promotion eligibility. The 'Run' button is visible on the right. Below the script, the 'Results' pane shows the output of the script, listing the names of employees who are eligible for promotion.

```

327
328 --Verificati daca fiecare angajat este eligibil pentru promovare in functie de vechimea sa in companie si de salariu (cel putin 5 ani de vechime si un salariu mai mare de 5000 lei).
329
330 DECLARE
331 v_nume angajati.nume %type;
332 v_prenume angajati.prenume %type;
333 v_salariu angajati.salariu %type;
334 v_data_angajarii angajati.data_angajarii %type;
335 v_vechime number(3);
336 v_min number(3);
337 v_max number(3);
338 i number(3);
339 BEGIN
340 SELECT MIN(ID_ANGAJAT), MAX(ID_ANGAJAT) INTO v_min, v_max FROM angajati;
341 i:=v_min;
342 WHILE i<=v_max LOOP
343 SELECT nume, prenume, salariu, data_angajarii, (sysdate-data_angajarii)/365 INTO v_nume, v_prenume, v_salariu, v_data_angajarii, v_vechime FROM angajati where id_angajat=i;
344 IF v_vechime>=5 AND v_salariu>5000 THEN dbms_output.put_line('Angajatul '||v_nume||' '||v_prenume||' are o vechime de '||v_vechime||' ani, si un salariu de '||v_salariu||' lei, deci este eligibil pentru o promovare.');
```

```

345 ELSE dbms_output.put_line('Angajatul '||v_nume||' '||v_prenume||' are o vechime de '||v_vechime||' ani, si un salariu de '||v_salariu||' lei, deci nu este eligibil pentru o promovare.');
```

```

346 END IF;
347 i:=i+1;
348 END LOOP;
349 END;
```

The 'Results' pane shows the following output:

```

Angajatul Antonescu Robert are o vechime de 5 ani, si un salariu de 9300 lei, deci este eligibil pentru o promovare.
Angajatul Mihai Andreasa are o vechime de 3 ani, si un salariu de 7600 lei, deci nu este eligibil pentru o promovare.
Angajatul Georgescu Alex are o vechime de 2 ani, si un salariu de 8800 lei, deci nu este eligibil pentru o promovare.
Angajatul Florescu Diana are o vechime de 7 ani, si un salariu de 9200 lei, deci este eligibil pentru o promovare.
Angajatul Dobrescu Gabriel are o vechime de 8 ani, si un salariu de 7900 lei, deci este eligibil pentru o promovare.
Angajatul Stanciu Elena are o vechime de 13 ani, si un salariu de 8700 lei, deci este eligibil pentru o promovare.
Angajatul Ilie Ana-Maria are o vechime de 15 ani, si un salariu de 9400 lei, deci este eligibil pentru o promovare.
```

Ex 2:

Aplicati un discount la suma unui împrumut pentru toți clienții care au dobândă peste 7% și termenul de scadență peste 10 ani (5% reducere).

```

DECLARE
```

```

v_nume clienti.nume %type; v_prenume clienti.prenume %type; v_suma
imprumuturi.suma_imprumutata %type; v_reducere number(5); v_suma_modificata number(8);
v_termen number(3);
```

```

v_rata imprumuturi.rata_dobanzii %type; v_min number (3); v_max number(3); i number(3);
```

```

BEGIN
```

```

SELECT MIN(ID_client), MAX(ID_client) INTO v_min, v_max FROM Imprumuturi; i:=v_min;
```

```

WHILE i<=v_max LOOP
```

```

SELECT nume, prenume INTO v_nume, v_prenume FROM clienti where id_client=i;
```

```

SELECT suma_imprumutata, (data_scadenta-sysdate)/365, rata_dobanzii INTO v_suma, v_termen,
v_rata FROM imprumuturi where id_client=i;
```

```

IF v_rata>=5 AND v_termen>10 THEN v_reducere:=0.05*v_suma; v_suma_modificata:=v_suma-
0.05*v_suma; ELSE v_reducere:=0; v_suma_modificata:=v_suma; END IF; i:=i+2;
```

```

dbms_output.put_line('Clientul '||v_nume||' '||v_prenume||' are un imprumut in valoare de
'||v_suma||' lei, pe un termen de '||v_termen||' ani, caruia i se va aplica o reducere de
```

```
'||v_reducere||' lei, deci valoarea finala a imprumutului dupa aplicarea reducerii va fi de
'||v_suma_modificata||' lei.');
```

END LOOP;

END;

The screenshot shows the APEX SQL Workshop interface. The top bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. A search bar and user profile 'LINGU VANESA' are on the right. The main area displays a PL/SQL block with line numbers 353 to 378. The block declares variables for client details, loan terms, interest rates, and a loop counter. It uses a WHILE loop to iterate through a range of clients, calculating the final loan amount after applying a discount based on the interest rate and term. The results pane at the bottom shows the output for three clients: Lingua Vanesa-Denisa, Clientul Firan Marian, and Clientul Flutura Patrisia.

```

353 DECLARE
354   v_nume clienti.nume %type;
355   v_prenume clienti.prenume %type;
356   v_suma imprumuturi.suma_imprumutata %type;
357   v_reducere number(5);
358   v_suma_modificata number(8);
359   v_termen number(3);
360   v_rata imprumuturi.rata_dobanzii %type;
361   v_min number(3);
362   v_max number(3);
363   i number(3);
364 BEGIN
365   SELECT MIN(ID_client), MAX(ID_client) INTO v_min, v_max FROM Imprumuturi;
366   i:=v_min;
367   WHILE i<=v_max LOOP
368     SELECT nume, prenume INTO v_nume, v_prenume FROM clienti where id_client=i;
369     SELECT suma_imprumutata, (data_scadenta-sysdate)/365, rata_dobanzii INTO v_suma, v_termen, v_rata FROM imprumuturi where id_client=i;
370     IF v_rata>=5 AND v_termen>10 THEN v_reducere:=0.05*v_suma;
371     v_suma_modificata:=v_suma-0.05*v_suma;
372     ELSE v_reducere:=0;
373     v_suma_modificata:=v_suma;
374     END IF;
375     i:=i+2;
376     dbms_output.put_line('Clientul '||v_nume||' '||v_prenume||' are un imprumut in valoare de '||v_suma||' lei, pe un termen de '||v_termen||' ani, caruia i se va aplica o reducere de '||v_reducere||' lei, deci valoarea finala
377     '||v_suma_modificata||' lei.');
```

Results

Clientul Lingua Vanesa-Denisa are un imprumut in valoare de 5000 lei, pe un termen de 18 ani, caruia i se va aplica o reducere de 250 lei, deci valoarea finala a imprumutului dupa aplicarea reducerii va fi de 4750 lei.
Clientul Firan Marian are un imprumut in valoare de 30000 lei, pe un termen de 27 ani, caruia i se va aplica o reducere de 0 lei, deci valoarea finala a imprumutului dupa aplicarea reducerii va fi de 30000 lei.
Clientul Flutura Patrisia are un imprumut in valoare de 24000 lei, pe un termen de 24 ani, caruia i se va aplica o reducere de 1200 lei, deci valoarea finala a imprumutului dupa aplicarea reducerii va fi de 22800 lei.

Statement processed.

Acelasi exercitiu, insa folosind structura CASE in loc de IF:

DECLARE

```
v_nume clienti.nume %type; v_prenume clienti.prenume %type; v_suma
imprumuturi.suma_imprumutata %type; v_reducere number(5); v_suma_modificata number(8);
v_termen number(3);
```

```
v_rata imprumuturi.rata_dobanzii %type; v_min number(3); v_max number(3); i number(3);
```

BEGIN

```
SELECT MIN(ID_client), MAX(ID_client) INTO v_min, v_max FROM Imprumuturi; i:=v_min;
```

WHILE i<=v_max LOOP

```
SELECT nume, prenume INTO v_nume, v_prenume FROM clienti where id_client=i;
```

```
SELECT suma_imprumutata, (data_scadenta-sysdate)/365, rata_dobanzii INTO v_suma, v_termen,
v_rata
```

```
FROM imprumuturi where id_client=i;
```

```
v_reducere:=CASE WHEN v_rata>=5 AND v_termen>10 THEN 0.05*v_suma
```

```
ELSE 0 END; v_suma_modificata:=CASE WHEN v_rata>=5 AND v_termen>10 THEN v_suma-v_reducere
ELSE v_suma END;
```

```
i:=i+2;
```

```
dbms_output.put_line('Clientul '||v_nume||' '||v_prenume||' are un imprumut in valoare de
'||v_suma||' lei, pe un termen de '||v_termen||' ani, caruia i se va aplica o reducere de
'||v_reducere||' lei, deci valoarea finala a imprumutului dupa aplicarea reducerii va fi de
'||v_suma_modificata||' lei.');
```

END LOOP;

END;

The screenshot shows the APEX SQL Workshop interface. The top bar includes the APEX logo, navigation tabs (App Builder, SQL Workshop, Team Development, Gallery), a search bar, and a user profile for LUNGU VANESA. The main area displays a PL/SQL script with line numbers 380 to 406. The script calculates the final loan amount for clients with a loan term greater than 2 years, applying a 5% discount. The results pane at the bottom shows the output of the script, listing three clients and their final loan amounts.

```

380 DECLARE
381   v_nume clienti.nume %type;
382   v_prenume clienti.prenume %type;
383   v_suma imprumuturi.suma_imprumutata %type;
384   v_reducere number(5);
385   v_suma_modificata number(8);
386   v_termen number(3);
387   v_rata imprumuturi.rata_dobanzii %type;
388   v_min number(3);
389   v_max number(3);
390   i number(3);
391 BEGIN
392   SELECT MIN(id_client), MAX(id_client) INTO v_min, v_max FROM imprumuturi;
393   i:=v_min;
394   WHILE i<=v_max LOOP
395     SELECT nume, prenume INTO v_nume, v_prenume FROM clienti WHERE id_client=i;
396     SELECT suma_imprumutata, (data_scadenta-sysdate)/365, rata_dobanzii INTO v_suma, v_termen, v_rata FROM imprumuturi WHERE id_client=i;
397     v_reducere:=CASE WHEN v_rata>5 AND v_termen>10 THEN 0.05*v_suma
398     ELSE 0
399     END;
400     v_suma_modificata:=CASE WHEN v_rata>5 AND v_termen>10 THEN v_suma-v_reducere
401     ELSE v_suma
402     END;
403     i:=i+1;
404     dbms_output.put_line('Clientul '||v_nume||' '||v_prenume||' are un imprumut in valoare de '||v_suma||' lei, pe un termen de '||v_termen||' ani, caruia i se va aplica o reducere de '||v_reducere||' lei, deci valoarea finala
405     END LOOP;
406 END;

```

Results

Clientul Lungu Vanesia-Denisa are un imprumut in valoare de 5000 lei, pe un termen de 18 ani, caruia i se va aplica o reducere de 250 lei, deci valoarea finala a imprumutului dupa aplicarea reducerii va fi de 4750 lei.
 Clientul Firan Marian are un imprumut in valoare de 30000 lei, pe un termen de 22 ani, caruia i se va aplica o reducere de 0 lei, deci valoarea finala a imprumutului dupa aplicarea reducerii va fi de 30000 lei.
 Clientul Fluturu Patrisia are un imprumut in valoare de 24000 lei, pe un termen de 24 ani, caruia i se va aplica o reducere de 1200 lei, deci valoarea finala a imprumutului dupa aplicarea reducerii va fi de 22800 lei.

Statement processed.

Ex 3:

Calculati și afisati suma soldurilor pentru conturile de economii deschise de mai mult de doi ani pentru fiecare client.

```

DECLARE
v_nume clienti.nume %type; v_prenume clienti.prenume %type; v_sold conturi.sold %type; v_suma
number(10):=0; v_cont conturi.tip_cont %type; v_data number(3);
BEGIN
FOR i IN 1..8 LOOP
SELECT nume, prenume INTO v_nume, v_prenume FROM clienti WHERE id_client=i;
SELECT tip_cont, sold, (sysdate-data_deschiderii)/365 INTO v_cont, v_sold, v_data FROM conturi
WHERE id_client=i;
IF v_data>2 AND v_cont LIKE 'Economii' THEN v_suma := v_suma + v_sold; END IF;
dbms_output.put_line('Clientul '||v_nume||' '||v_prenume||' prezinta un cont de '||v_cont||' cu un
sold de '||v_sold||' lei. Contul a fost deschis in urma cu '||v_data||' ani.');
```

```

END LOOP;
dbms_output.put_line('Suma soldurilor conturilor de economii deschise de mai mult de doi ani este
'||v_suma||' lei.');
```

```

END;
```

The screenshot shows the APEX SQL Workshop interface. At the top, there's a navigation bar with 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. A search bar is on the right. Below the navigation bar, the 'SQL Commands' tab is active, showing a PL/SQL script. The script calculates and displays the sum of loan amounts for clients who have opened more than two savings accounts. The script includes a loop to iterate through clients and a conditional statement to update the loan status based on the interest rate. The 'Results' tab at the bottom shows the output of the script, which lists the client names, their loan amounts, and the status of their savings accounts. The status is 'Inactiv' for interest rates below 5%, 'Activ' for interest rates between 5% and 8%, and 'Pericol' for interest rates above 8%.

```

408 --Calculati și afisati suma soldurilor pentru conturile de economii deschise de mai mult de doi ani pentru fiecare client.
409 DECLARE
410     v_nume clienti.nume %TYPE;
411     v_prenume clienti.prenume %TYPE;
412     v_sold conturi.sold %TYPE;
413     v_suma number(10)=0;
414     v_cont conturi.tip_cont %TYPE;
415     v_data number(1);
416 BEGIN
417     FOR i IN 1..8 LOOP
418         SELECT nume, prenume INTO v_nume, v_prenume FROM clienti WHERE id_client=i;
419         SELECT tip_cont, sold, (sysdate-data_deschiderii)/365 INTO v_cont, v_sold, v_data FROM conturi WHERE id_client=i;
420         IF v_data >= 2 AND v_cont = 'Economie' THEN v_suma := v_suma + v_sold;
421     END IF;
422     dbms_output.put_line('Clientul '||v_nume||' '||v_prenume||' prezinta un cont de '||v_cont||' cu un sold de '||v_sold||' lei. Contul a fost deschis in urma cu '||v_data||' ani. ');
423     END LOOP;
424     dbms_output.put_line('Suma soldurilor conturilor de economii deschise de mai mult de doi ani este '||v_suma||' lei. ');
425 END;
426

```

Results

Clientul Lungu Vanesa-Denisa prezinta un cont de Economii cu un sold de 12000 lei. Contul a fost deschis in urma cu 3 ani.
 Clientul Istrate Andreia-Alina prezinta un cont de Bursa cu un sold de 900 lei. Contul a fost deschis in urma cu 1 ani.
 Clientul Firan Marian prezinta un cont de Salariu cu un sold de 8200 lei. Contul a fost deschis in urma cu 6 ani.
 Clientul Fluturu Andreea prezinta un cont de Economii cu un sold de 4800 lei. Contul a fost deschis in urma cu 4 ani.
 Clientul Fluturu Patricia prezinta un cont de Bursa cu un sold de 730 lei. Contul a fost deschis in urma cu 7 ani.
 Clientul Fuica Andreea prezinta un cont de Economii cu un sold de 2800 lei. Contul a fost deschis in urma cu 1 ani.
 Clientul Faica Diana prezinta un cont de Economii cu un sold de 5200 lei. Contul a fost deschis in urma cu 4 ani.
 Clientul Tudor Razvan-Constantin prezinta un cont de Salariu cu un sold de 7300 lei. Contul a fost deschis in urma cu 7 ani.
 Suma soldurilor conturilor de economii deschise de mai mult de doi ani este 22000 lei.

Statement processed.

Ex 4:

Actualizează statusul unui împrumut în funcție de rata dobanzii. Dacă rata dobânzii este mai mică de 5%, să fie setat Inactiv, dacă este între 5% și 8% să fie setat Activ, iar pentru mai mare de 8% să fie setat Pericol.

```

DECLARE
v_id_client Imprumuturi.id_client%TYPE :=3; (&c voiam sa dau valoarea de la tastatura, dar nu merge in apex, genereaza eroare, asa ca am pus direct valoarea ca sa functioneze)
v_rata_dobanzii Imprumuturi.Rata_dobanzii%TYPE;
v_status_nou VARCHAR2(15);

BEGIN
SELECT Rata_dobanzii INTO v_rata_dobanzii
FROM Imprumuturi
WHERE id_client = v_id_client;

v_status_nou := CASE WHEN v_rata_dobanzii < 5 THEN 'Inactiv'
WHEN v_rata_dobanzii BETWEEN 5 AND 8 THEN 'Activ'
ELSE 'Pericol'
END;

UPDATE Imprumuturi SET Status = v_status_nou WHERE id_client = v_id_client;
dbms_output.put_line('Statusul imprumutului a fost actualizat: ' || v_status_nou);

END;

```

```
444 select * from imprumuturi;
```

ID_IMPRUMUT	ID_CLIENT	SUMA_IMPRUMUTATA	DATA_ACORDARII	DATA_SCADENTA	RATA DOBANZII	STATUS
100	1	5000	02/14/2023	02/14/2043	7	Activ
300	3	30000	11/09/2021	11/09/2051	-	Activ
500	5	24000	05/20/2020	05/20/2048	8	Activ

3 rows returned in 0.01 seconds [Download](#)

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language: SQL Rows: 10 Clear Command Find Tables Save Run

```

426 --Actualizează statusul unui împrumut în funcție de rata dobanzii. Dacă rata dobanzii este mai mică de 5%, să fie setat Inactiv, dacă este între 5% și 8% să fie setat Activ, iar pentru mai mare de 8% să fie setat Pericol.
427
428 DECLARE
429   v_id_client imprumuturi.id_client%TYPE := 1; -- 1: luăm ca valoare de la tastatură, dar nu merge în apex, generarea erorilor, așa că am pus direct valoarea ca să funcționeze
430   v_rata_dobanzii imprumuturi.rata_dobanzii%TYPE;
431   v_status_nou VARCHAR2(15);
432 BEGIN
433   SELECT rata_dobanzii INTO v_rata_dobanzii
434   FROM imprumuturi
435   WHERE id_client = v_id_client;
436   v_status_nou := CASE WHEN v_rata_dobanzii < 5 THEN 'Inactiv'
437   WHEN v_rata_dobanzii BETWEEN 5 AND 8 THEN 'Activ'
438   ELSE 'Pericol';
439   EMP;
440   UPDATE imprumuturi SET Status = v_status_nou WHERE id_client = v_id_client;
441   dms_output.put_line('Statusul împrumutului a fost actualizat: ' || v_status_nou);
442 END;
443
444 select * from imprumuturi;
```

Results Explain Describe Saved SQL History

Statusul împrumutului a fost actualizat: Pericol

1 row(s) updated.

0.02 seconds

Copyright © 1996, 2024, Oracle and/or its affiliates. Oracle APEX 24.1

```
444 select * from imprumuturi;
```

ID_IMPRUMUT	ID_CLIENT	SUMA_IMPRUMUTATA	DATA_ACORDARII	DATA_SCADENTA	RATA DOBANZII	STATUS
100	1	5000	02/14/2023	02/14/2043	7	Activ
300	3	30000	11/09/2021	11/09/2051	-	Pericol
500	5	24000	05/20/2020	05/20/2048	8	Activ

3 rows returned in 0.00 seconds [Download](#)

Comanda a trecut direct pe ramura else a structurii, deoarece comisionul pentru clientul cu id_client=3 era NULL.

Ex 1: Creează un cursor care să returneze numele, prenumele clientului, tipul de cont si soldul acestuia, cu conditia ca acesta sa depaseasca suma de 5000 lei.

DECLARE

CURSOR c IS SELECT cl.num, cl.prenume, co.tip_cont, co.sold FROM clienti cl JOIN conturi co USING (id_client) WHERE co.sold>5000;

v_num clienti.num%TYPE;

v_prenume clienti.prenume%TYPE;

v_tip_cont conturi.tip_cont%TYPE;

v_sold conturi.sold%TYPE;

BEGIN

OPEN c;

LOOP

FETCH c INTO v_num, v_prenume, v_tip_cont, v_sold;

EXIT WHEN c%NOTFOUND;

DBMS_OUTPUT.PUT_LINE('Clientul ' || v_num || ' ' || v_prenume || ' are un cont pentru ' || v_tip_cont || ' cu un sold de ' || v_sold || ' lei.');

END LOOP;

CLOSE c;

END;

```

644 select * from imprumuturi;
645
646 -- Ex 1: Creează un cursor care să returneze numele, prenumele clientului, tipul de cont si soldul acestuia, cu conditia ca acesta sa depaseasca suma de 5000 lei.
647
648 CURSOR c IS SELECT cl.num, cl.prenume, co.tip_cont, co.sold FROM clienti cl JOIN conturi co USING (id_client) WHERE co.sold>5000;
649 v_num clienti.num%TYPE;
650 v_prenume clienti.prenume%TYPE;
651 v_tip_cont conturi.tip_cont%TYPE;
652 v_sold conturi.sold%TYPE;
653 BEGIN
654 OPEN c;
655 LOOP
656 FETCH c INTO v_num, v_prenume, v_tip_cont, v_sold;
657 EXIT WHEN c%NOTFOUND;
658 DBMS_OUTPUT.PUT_LINE('Clientul ' || v_num || ' ' || v_prenume || ' are un cont pentru ' || v_tip_cont || ' cu un sold de ' || v_sold || ' lei.');

```

Results

Clientul	Prenume	Tipul de cont	Soldul
Clientul Lungu Vanesa-Denisa		are un cont pentru Economii	cu un sold de 12000lei.
Clientul Fuica Diana		are un cont pentru Economii	cu un sold de 5200lei.
Clientul Firan Marian		are un cont pentru Salariu	cu un sold de 8200lei.
Clientul Tudor Razvan-Constantin		are un cont pentru Salariu	cu un sold de 7300lei.

Statement processed.

0.03 seconds

Ex 2: Creaza un cursor pentru a afisa numele, prenumele, id-ul clientului si al tranzactiei efectuate impreuna cu suma tranzactiei, doar pentru tranzactiile efectuate dupa data de 1.12.2023.

DECLARE

```
CURSOR c IS SELECT cl.numc, cl.prenume, id_client, id_cont, tr.id_tranzactie, suma FROM clienti cl JOIN
conturi co USING (id_client) JOIN tranzactii tr USING (id_cont) WHERE tr.data_tranzactiei>TO_DATE('01-
12-2023', 'DD-MM-YYYY');
```

```
BEGIN
```

```
FOR v IN c LOOP
```

```
DBMS_OUTPUT.PUT_LINE('Clientul ' || v.numc || ' ' || v.prenume || ' cu id-ul ' || v.id_client || ' a
efectuat o tranzactie cu id-ul ' || v.id_tranzactie || ' pentru suma de ' || v.suma || ' lei.');
```

```
END LOOP;
```

```
END;
```

The screenshot shows the APEX SQL Workshop interface. The 'SQL Commands' pane contains the following code:

```
457 EXIT WHEN :NOTFOUND;
458 DBMS_OUTPUT.PUT_LINE('Clientul ' || v.numc || ' ' || v.prenume || ' are un cont pentru ' || v.tip_cont || ' cu un sold de ' || v.sold || ' lei.');
```

The 'Results' pane displays the output of the query, showing client details and transaction information:

```
Clientul Lungu Vanesa-Denisa cu id-ul 1 a efectuat o tranzactie cu id-ul 1 pentru suma de 20 lei.
Clientul Lungu Vanesa-Denisa cu id-ul 1 a efectuat o tranzactie cu id-ul 2 pentru suma de 500 lei.
Clientul Lungu Vanesa-Denisa cu id-ul 1 a efectuat o tranzactie cu id-ul 3 pentru suma de 150 lei.
Clientul Fulca Andreea cu id-ul 6 a efectuat o tranzactie cu id-ul 9 pentru suma de 86 lei.
Clientul Fluturu Patrisia cu id-ul 5 a efectuat o tranzactie cu id-ul 6 pentru suma de 300 lei.
Clientul Tudor Razvan-Constantin cu id-ul 8 a efectuat o tranzactie cu id-ul 7 pentru suma de 1000 lei.
```

Ex 3: Creaaza un cursor pentru a selecta și afișa numele, prenumele, id-ul clientului, contul si id-ul sau, soldurile conturilor si data ultimei tranzactii, pentru clientii care au conturi inactive (nu au avut tranzacții în ultimele 15 luni).

```
DECLARE
```

```
CURSOR c IS SELECT DISTINCT cl.numc, cl.prenume, id_client, id_cont,co.tip_cont, co.sold FROM clienti
cl JOIN conturi co USING (id_client) JOIN tranzactii USING (id_cont) WHERE data_tranzactiei >
ADD_MONTHS(SYSDATE, -15);
```

```
CURSOR d(a number) is select data_tranzactiei from tranzactii WHERE id_cont IN (SELECT id_cont from
conturi where id_client=a) AND data_tranzactiei > ADD_MONTHS(SYSDATE, -15) ORDER BY
data_tranzactiei DESC FETCH FIRST 1 ROW ONLY;
```

```
BEGIN
```

```
FOR v IN c LOOP
```

```
dbms_output.put_line('Clientul ' || v.numc || ' ' || v.prenume || ' cu id-ul ' || v.id_client || ' are un cont
utilizat pentru ' || v.tip_cont || ' inactiv, cu id-ul ' || v.id_cont || ' si un sold de ' || v.sold || ' lei.');
```

```
FOR w IN d (v.id_client) LOOP
```

```
dbms_output.put_line(' Data ultimei tranzactii efectuate este ' || w.data_tranzactiei);
```

```
END LOOP;
```

```
END LOOP;
```

```
END;
```

The screenshot shows the APEX SQL Workshop interface. The top bar includes the APEX logo, navigation tabs (App Builder, SQL Workshop, Team Development, Gallery), a search bar, and a user profile for LUNGU VANESA. The main area displays a PL/SQL script with line numbers 467 to 485. The script includes a loop for outputting client information and a cursor for selecting active clients. The 'Results' tab at the bottom shows the output of the script, listing client details and transaction dates.

```
467 FOR v IN c LOOP
468   DBMS_OUTPUT.PUT_LINE('Clientul ' || v.num || ' ' || v.prenume || ' cu id-ul ' || v.id_client || ' a efectuat o tranzactie cu id-ul ' || v.id_tranzactie || ' pentru suma de ' || v.suma || ' lei. ');
469 END LOOP;
470 END;
471
472 --Ex 3: Creaza un cursor pentru a selecta si afisa numele, prenumele, id-ul clientului, contul si id-ul sau, soldurile conturilor si data ultimei tranzactii, pentru clientii care au conturi inactivate (nu au avut tranzactii)
473 DECLARE
474   CURSOR c IS SELECT cl.nume, cl.prenume, id_client, id_cont, co.tip_cont, co.sold FROM clienti cl JOIN conturi co USING (id_client) JOIN tranzactii tr USING (id_client, id_cont) WHERE data_tranzactii > ADD_MONTHS(SYSDATE, -15);
475   CURSOR d(a number) IS select data_tranzactiei from tranzactii WHERE id_cont IN (SELECT id_cont from conturi where id_client=a);
476 BEGIN
477   FOR v IN c LOOP
478     dbms_output.put_line('Clientul ' || v.num || ' ' || v.prenume || ' cu id-ul ' || v.id_client || ' are un cont utilizat pentru ' || v.tip_cont || ' inactiv, cu id-ul ' || v.id_cont || ' si un sold de ' || v.sold || ' lei. ');
479   FOR w IN d (v.id_client) LOOP
480     dbms_output.put_line('      Data ultimei tranzactii efectuate este ' || w.data_tranzactiei);
481   END LOOP;
482 END LOOP;
483 END;
484
485 --Ex 4: Afisati tranzactiile celor mai activi 3 clienti (cei care au realizat cele mai multe tranzactii.)
```

Results Explain Describe Saved SQL History

Clientul Tudor Razvan-Constantin cu id-ul 8 are un cont utilizat pentru Salariu inactiv, cu id-ul 80 si un sold de 7300 lei.
Data ultimei tranzactii efectuate este 12/04/2023
Clientul Lungu Vanesa-Denisa cu id-ul 1 are un cont utilizat pentru Economii inactiv, cu id-ul 10 si un sold de 12000 lei.
Data ultimei tranzactii efectuate este 12/12/2023
Clientul Fluturu Patrisia cu id-ul 5 are un cont utilizat pentru Bursa inactiv, cu id-ul 50 si un sold de 730 lei.
Data ultimei tranzactii efectuate este 12/08/2023
Clientul Istrute Andreia-Alexandra cu id-ul 2 are un cont utilizat pentru Bursa inactiv, cu id-ul 20 si un sold de 900 lei.
Data ultimei tranzactii efectuate este 10/10/2023
Clientul Fulca Andreea cu id-ul 6 are un cont utilizat pentru Economii inactiv, cu id-ul 60 si un sold de 2800 lei.
Data ultimei tranzactii efectuate este 12/12/2023
Clientul Firan Marian cu id-ul 3 are un cont utilizat pentru Salariu inactiv, cu id-ul 30 si un sold de 8200 lei.
Data ultimei tranzactii efectuate este 12/01/2023

Ex 4: Afisati tranzactiile celor mai activi 3 clienti (cei care au realizat cele mai multe tranzactii).

DECLARE

CURSOR c IS SELECT cl.nume, cl.prenume, id_client, id_cont, count(id_tranzactie) AS numar_tranzactii
FROM clienti cl JOIN conturi co USING (id_client) JOIN tranzactii tr USING (id_client, id_cont) GROUP BY cl.nume,
cl.prenume, id_client, id_cont ORDER BY numar_tranzactii DESC FETCH FIRST 3 ROWS ONLY;

CURSOR d(a number) IS select data_tranzactiei from tranzactii WHERE id_cont IN (SELECT id_cont from
conturi where id_client=a);

BEGIN

FOR v IN c LOOP

dbms_output.put_line('Clientul ' || v.num || ' ' || v.prenume || ' cu id-ul ' || v.id_client || ' este unul
dintre cei mai activi clienti, efectuand ' || v.numar_tranzactii || ' tranzactii.');

FOR w IN d (v.id_client) LOOP

dbms_output.put_line(' Tranzactiile au fost efectuate in data de: ' || w.data_tranzactiei);

END LOOP;

END LOOP;

END;

The screenshot shows the APEX SQL Workshop interface with the same PL/SQL script as the previous image. The 'Results' tab at the bottom shows the output of the script, listing client details and transaction dates for the most active clients.

```
486 DECLARE
487   CURSOR c IS SELECT cl.nume, cl.prenume, id_client, id_cont, count(id_tranzactie) AS numar_tranzactii FROM clienti cl JOIN conturi co USING (id_client) JOIN tranzactii tr USING (id_client, id_cont) GROUP BY cl.nume, cl.prenume, id_client
488   CURSOR d(a number) IS select data_tranzactiei from tranzactii WHERE id_cont IN (SELECT id_cont from conturi where id_client=a);
489 BEGIN
490   FOR v IN c LOOP
491     dbms_output.put_line('Clientul ' || v.num || ' ' || v.prenume || ' cu id-ul ' || v.id_client || ' este unul dintre cei mai activi clienti, efectuand ' || v.numar_tranzactii || ' tranzactii. ');
492   FOR w IN d (v.id_client) LOOP
493     dbms_output.put_line('      Tranzactiile au fost efectuate in data de: ' || w.data_tranzactiei);
494   END LOOP;
495 END LOOP;
496 END;
```

Results Explain Describe Saved SQL History

Clientul Lungu Vanesa-Denisa cu id-ul 1 este unul dintre cei mai activi clienti, efectuand 3 tranzactii.
Tranzactiile au fost efectuate in data de: 12/12/2023
Tranzactiile au fost efectuate in data de: 12/11/2023
Tranzactiile au fost efectuate in data de: 12/12/2023
Clientul Firan Marian cu id-ul 3 este unul dintre cei mai activi clienti, efectuand 2 tranzactii.
Tranzactiile au fost efectuate in data de: 11/20/2023
Tranzactiile au fost efectuate in data de: 12/01/2023
Clientul Tudor Razvan-Constantin cu id-ul 8 este unul dintre cei mai activi clienti, efectuand 2 tranzactii.
Tranzactiile au fost efectuate in data de: 05/15/2020
Tranzactiile au fost efectuate in data de: 12/04/2023

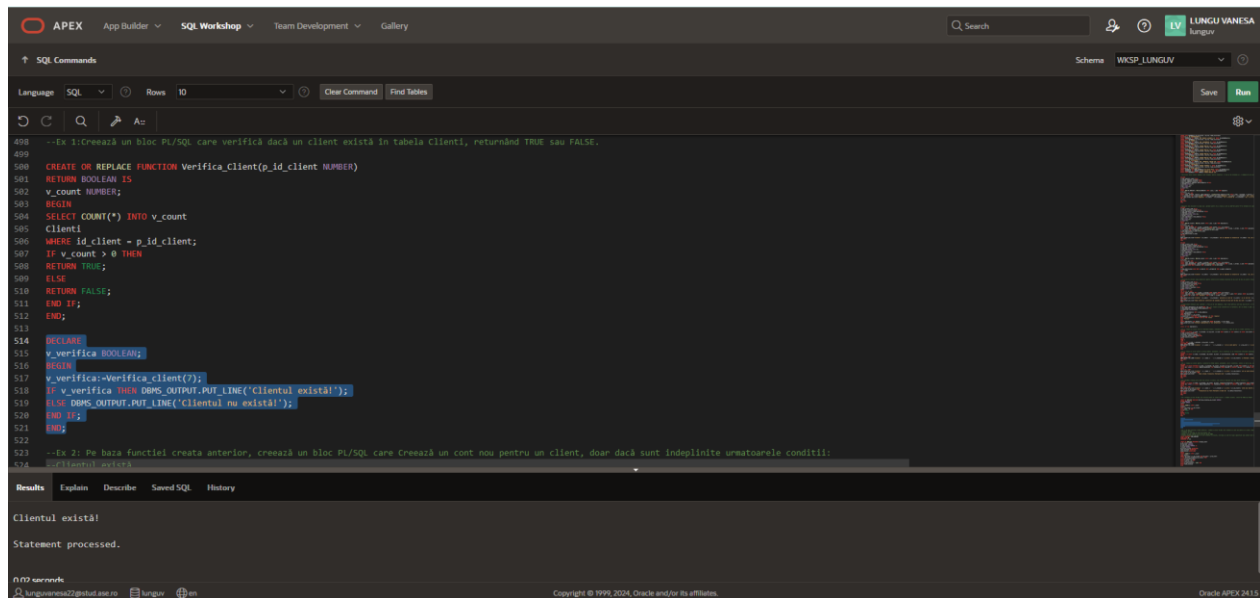
Statement processed.

-----EXCEPTII SI SUBPROGRAME-----

Ex 1: Creează un bloc PL/SQL care verifică dacă un client există în tabela Clienti, returnând TRUE sau FALSE.

```
CREATE OR REPLACE FUNCTION Verifica_Client(p_id_client NUMBER)
RETURN BOOLEAN IS
v_count NUMBER;
BEGIN
SELECT COUNT(*) INTO v_count
Clienti
WHERE id_client = p_id_client;
IF v_count > 0 THEN
RETURN TRUE;
ELSE
RETURN FALSE;
END IF;
END;
```

```
DECLARE
v_verifica BOOLEAN;
BEGIN
v_verifica:=Verifica_client(7);
IF v_verifica THEN DBMS_OUTPUT.PUT_LINE('Clientul există!');
ELSE DBMS_OUTPUT.PUT_LINE('Clientul nu există!');
END IF;
END;
```



Ex 2: Pe baza funcției create anterior, creează un bloc PL/SQL care Creează un cont nou pentru un client, doar dacă sunt îndeplinite următoarele condiții:

Clientul există.

Clientul nu are deja un cont de același tip.

Soldul inițial trebuie să fie de cel puțin 1000.

Tratați excepțiile în cazul în care clientul nu există, are deja un cont de tipul specificat sau soldul este mai mic de 1000 lei.

```
CREATE SEQUENCE SEQ_CONTURI
```

```
START WITH 90
```

```
INCREMENT BY 10;
```

```
CREATE OR REPLACE PROCEDURE Creeaza_Cont(
```

```
  p_id_client NUMBER,
```

```
  p_tip_cont VARCHAR2,
```

```
  p_sold_initial NUMBER) IS
```

```
  v_count NUMBER;
```

```
  Nu_exista_client EXCEPTION;
```

```
  Cont_existent EXCEPTION;
```

```
  Sold_invalid EXCEPTION;
```

```
BEGIN
```

```
  SELECT COUNT(*) INTO v_count
```

```
  FROM Conturi
```

```
  WHERE id_client = p_id_client AND Tip_cont = p_tip_cont;
```

```
  IF NOT Verifica_Client(p_id_client) THEN
```

```
    RAISE Nu_exista_client;
```

```
  ELSIF v_count > 0 THEN
```



```

RAISE Cont_existent;
ELSIF p_sold_initial < 1000 THEN
RAISE Sold_invalid;
ELSE
INSERT INTO Conturi(id_cont, id_client, Tip_cont, Sold, Data_deschiderii)
VALUES (SEQ_CONTURI.NEXTVAL, p_id_client, p_tip_cont, p_sold_initial, SYSDATE);
DBMS_OUTPUT.PUT_LINE('Cont creat cu succes!');
END IF;
EXCEPTION
WHEN Nu_exista_client THEN
DBMS_OUTPUT.PUT_LINE('Clientul nu există!');
WHEN Cont_existent THEN
DBMS_OUTPUT.PUT_LINE('Clientul are deja un cont de acest tip!');
WHEN Sold_invalid THEN
DBMS_OUTPUT.PUT_LINE('Soldul inițial este prea mic!');
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Eroare neașteptată: ' || SQLERRM);
END;

BEGIN
Creeaza_Cont(20, 'Salariu', 100);
END;

```

The screenshot displays the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' tab is active, showing a script with line numbers 545 to 569. The script defines a procedure to create a client account, handling exceptions for non-existent clients, existing accounts, invalid initial sales, and other errors. The script is executed, and the 'Results' tab at the bottom shows the output: 'Clientul nu există!' and 'Statement processed.'.

```

545 WHEN Nu_exista_client THEN
546   DBMS_OUTPUT.PUT_LINE('Clientul nu există!');
547 ELSE
548   RAISE Cont_existent;
549   RAISE Sold_invalid;
550 ELSE
551   INSERT INTO Conturi(id_cont, id_client, Tip_cont, Sold, Data_deschiderii)
552   VALUES (SEQ_CONTURI.NEXTVAL, p_id_client, p_tip_cont, p_sold_initial, SYSDATE);
553   DBMS_OUTPUT.PUT_LINE('Cont creat cu succes!');
554 END IF;
555 EXCEPTION
556 WHEN Nu_exista_client THEN
557   DBMS_OUTPUT.PUT_LINE('Clientul nu există!');
558 WHEN Cont_existent THEN
559   DBMS_OUTPUT.PUT_LINE('Clientul are deja un cont de acest tip!');
560 WHEN Sold_invalid THEN
561   DBMS_OUTPUT.PUT_LINE('Soldul inițial este prea mic!');
562 WHEN OTHERS THEN
563   DBMS_OUTPUT.PUT_LINE('Eroare neașteptată: ' || SQLERRM);
564 END;
565
566 BEGIN
567   Creeaza_Cont(20, 'Salariu', 100);
568 END;
569

```

Results

```

Clientul nu există!
Statement processed.

```

Copyright © 1996, 2024, Oracle and/or its affiliates. Oracle APEX 24.1.5

The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The main area displays a SQL script for creating a table and inserting data. The script includes comments in Romanian and uses PL/SQL blocks. The 'Run' button is highlighted in green. The bottom bar shows 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'.

```

--> CREATE nu_exista_client;
--> ELSEIF v_cont > 0 THEN
--> RAISE Cont_existent;
--> ELSEIF p_sold_initial < 1000 THEN
--> RAISE Sold_invalid;
--> ELSE
--> INSERT INTO Conturi(id_client, Tip_cont, Sold, Data_deschiderii)
--> VALUES (SEQ_CONTURI.NEXTVAL, p_id_client, p_tip_cont, p_sold_initial, SYSDATE);
--> DBMS_OUTPUT.PUT_LINE('Cont creat cu succes!');
--> END IF;
--> EXCEPTION
--> WHEN Nu_exista_client THEN
--> DBMS_OUTPUT.PUT_LINE('Clientul nu exista!');
--> WHEN Cont_existent THEN
--> DBMS_OUTPUT.PUT_LINE('Clientul are deja un cont de acest tip!');
--> WHEN Sold_invalid THEN
--> DBMS_OUTPUT.PUT_LINE('Soldul initial este prea mic!');
--> WHEN OTHERS THEN
--> DBMS_OUTPUT.PUT_LINE('Intrare neasteptata: ' || SQLERRM);
--> END;
-->
--> BEGIN
--> Creeaza_cont(?, 'Economii', 1000);
--> END;

```

The bottom bar shows 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, showing the output of the SQL statement: 'Clientul are deja un cont de acest tip!' and 'Statement processed.'

565	BEGIN
566	Creeaza_Con(7, 'Salaris', 5000);
567	END;
568	
569	
570	SELECT * FROM Conturi WHERE id_client=7;
571	

Results	Explain	Describe	Saved SQL	History
ID_CONT	ID_CLIENT	TIP_CONT	SOLD	DATA_DESCRIBERII
70	7	Economii	5200	03/05/2021
90	7	Salaris	5000	12/19/2024

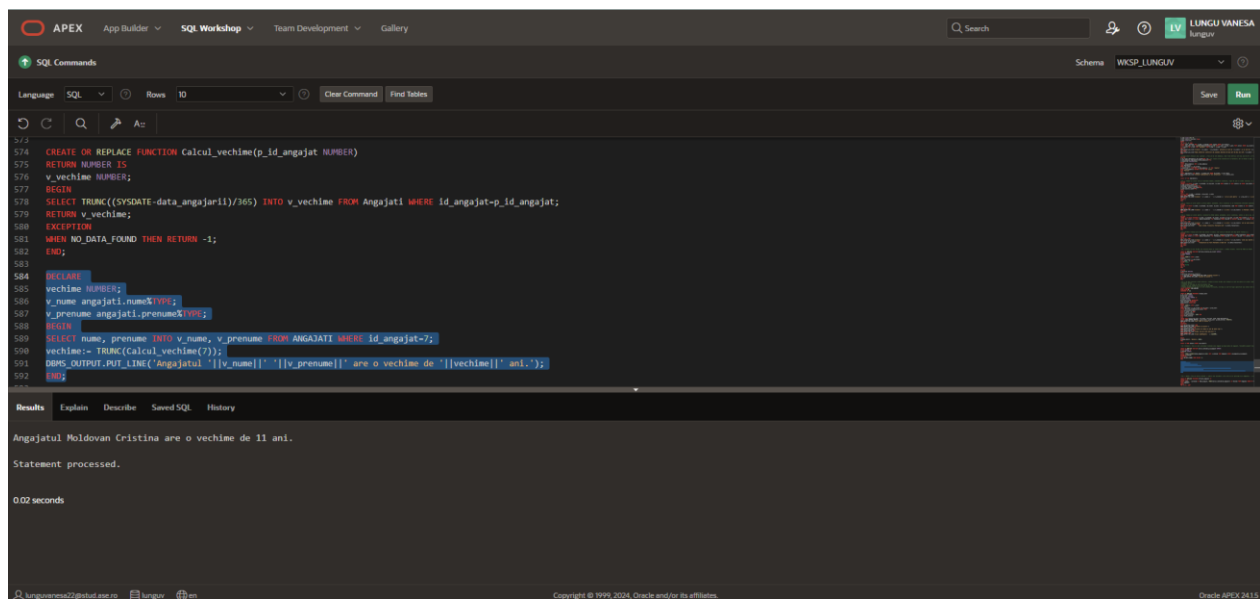
Copyright © 1999, 2024, Oracle and/or its affiliates. Oracle APEX 24.1.5

Ex 3: Creează o funcție care calculează vechimea unui angajat pe baza datei de angajare. Folosește această funcție într-o procedură care listează toți angajații cu vârsta calculată.

```

CREATE OR REPLACE FUNCTION Calcul_vehime(p_id_angajat NUMBER)
RETURN NUMBER IS
v_vehime NUMBER;
BEGIN
SELECT TRUNC((SYSDATE-data_angajarii)/365) INTO v_vehime FROM Angajati WHERE
id_angajat=p_id_angajat;
RETURN v_vehime;
EXCEPTION
WHEN NO_DATA_FOUND THEN RETURN -1;
END;

DECLARE
vehime NUMBER;
v_nume angajati.nume%TYPE;
v_prename angajati.prenume%TYPE;
BEGIN
SELECT nume, prename INTO v_nume, v_prename FROM ANGAJATI WHERE id_angajat=7;
vehime:= TRUNC(Calcul_vehime(7));
DBMS_OUTPUT.PUT_LINE('Angajatul '||v_nume||' '||v_prename||' are o vechime de '||vehime||'
ani.');
```



Ex 4: Apelați funcția Calcul_vechime în cadrul unei proceduri, prin care se vor parcurge toți angajatii, în scopul afișării vechimii fiecăruia.

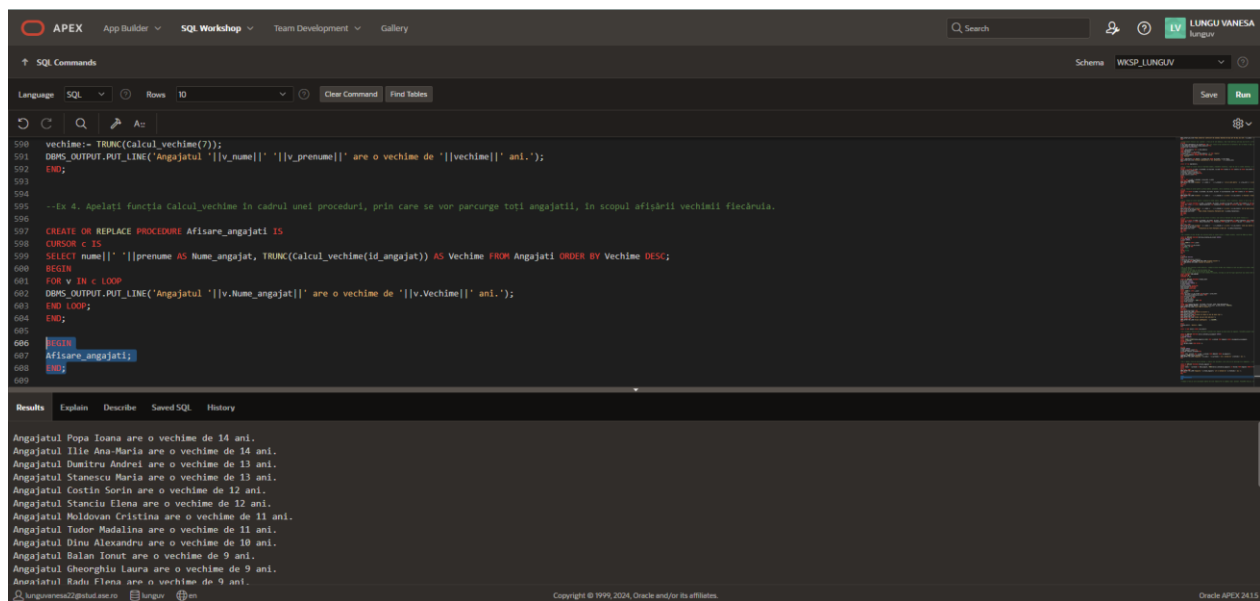
```
CREATE OR REPLACE PROCEDURE Afisare_angajati IS
CURSOR c IS
SELECT nume || ' ' || prenume AS Nume_angajat, TRUNC(Calcul_vechime(id_angajat)) AS Vechime FROM
Angajati ORDER BY Vechime DESC;
BEGIN
FOR v IN c LOOP
DBMS_OUTPUT.PUT_LINE('Angajatul ' || v.Nume_angajat || ' are o vechime de ' || v.Vechime || ' ani.');
```



```
END LOOP;
END;
```



```
BEGIN
Afisare_angajati;
END;
```



Ex 5: Creează o funcție care calculează numărul de zile rămase până la scadența unui împrumut.

CREATE OR REPLACE FUNCTION Zile_Pana_La_Scadenta(p_data_scadenta DATE)

RETURN NUMBER IS

v_zile NUMBER;

SCADENTA_DEPASITA EXCEPTION;

BEGIN

v_zile := p_data_scadenta - SYSDATE;

IF v_zile < 0 THEN

RAISE SCADENTA_DEPASITA;

END IF;

RETURN v_zile;

EXCEPTION

WHEN SCADENTA_DEPASITA THEN

DBMS_OUTPUT.PUT_LINE('Data scadenței a fost depășită!');

RETURN -1;

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('Eroare neașteptată: ' || SQLERRM);

RETURN -1;

END;

SELECT id_imprumut, suma_imprumutata, TRUNC(Zile_Pana_La_Scadenta(data_scadenta)) AS
Zile_pana_la_data_scadenta FROM Imprumuturi;

Schema **WKSP_LUNGV**

Save Run

Explain Describe Saved SQL History

Download