

GESTIUNEA SPITALELOR

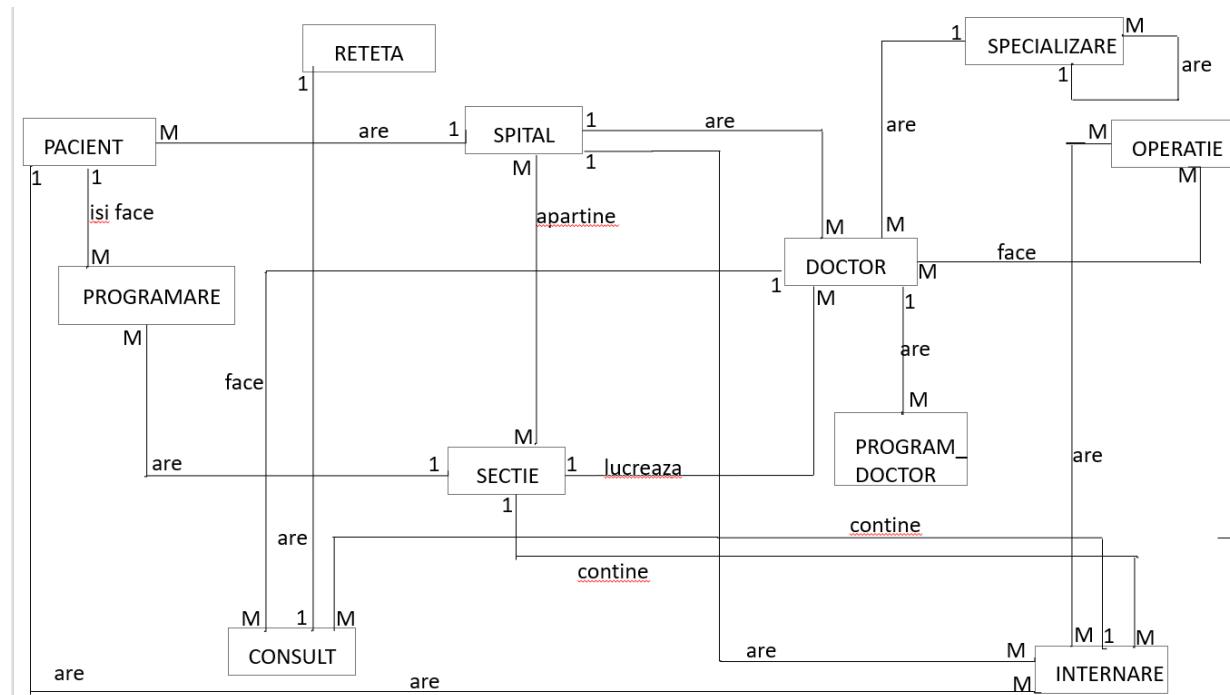
SANDU ANASTASIA GRUPA 242

EXERCITIU 1

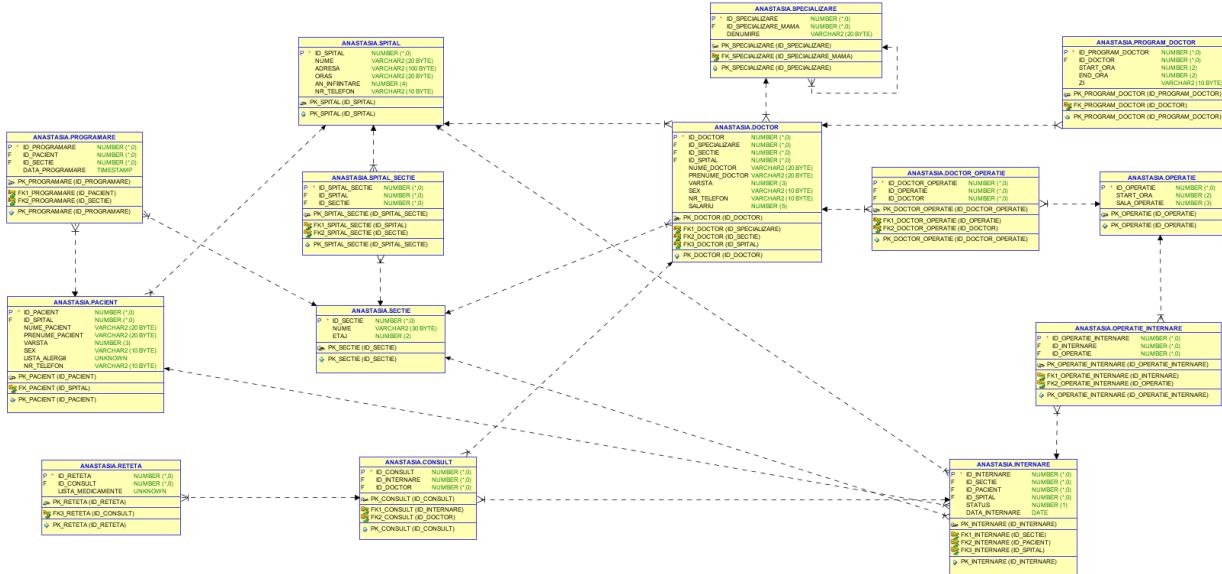
Baza de date conține informații despre spitale împreună cu pacienții, doctorii și secțiile acestor spitale cât și despre programările, consulturile și rețetele pacienților. De asemenea, reține date și despre internări și operații. Utilitatea acestei baze de date este de a tine evidența pacienților și doctorilor astfel încât să se poată gestiona programările și internările cât mai eficient în funcție de programul și nevoile tuturor.

Pacienții își pot face o programare și în caz de urgență se pot duce la spital pentru a se interna și dacă este cazul pentru operații. Operațiile se pot repeta deoarece dacă vorbim de exemplu de operația de apendicita atunci este aceeași operație care se realizează la mai mulți pacienți. Doctorii din spital au doar o specializare și datorită acestui fapt pot lucra la o singură secție. Aceștia pot avea programe diferite în funcție de ziua săptămânii, nu încep și nu termină zilnic la aceleași ore.

EXERCITIU 2



EXERCITIU 3



EXERCITIU 4

CREATE TABLE SPITAL(

```

ID_SPITAL INT NOT NULL,
NUME VARCHAR2(20),
ADRESA VARCHAR2(100),
CONSTRAINT PK_SPITAL PRIMARY KEY (ID_SPITAL)
);
```

ALTER TABLE SPITAL

ADD ORAS VARCHAR2(20);

ALTER TABLE SPITAL

ADD AN_INFINTARE NUMBER(4);

ALTER TABLE SPITAL

ADD NR_TELEFON VARCHAR2(10);

CREATE TABLE SECTIE(

```

ID_SECTIE INT NOT NULL,
NUME VARCHAR2(30),
CONSTRAINT PK_SECTIE PRIMARY KEY (ID_SECTIE)
```

);

ALTER TABLE SECTIE

ADD ETAJ NUMBER(2);

```
CREATE TABLE SPITAL_SECTIE(
    ID_SPITAL_SECTIE INT NOT NULL,
    ID_SPITAL INT,
    ID_SECTIE INT,
    CONSTRAINT PK_SPITAL_SECTIE PRIMARY KEY (ID_SPITAL_SECTIE),
    CONSTRAINT FK1_SPITAL_SECTIE FOREIGN KEY (ID_SPITAL) REFERENCES SPITAL(ID_SPITAL),
    CONSTRAINT FK2_SPITAL_SECTIE FOREIGN KEY (ID_SECTIE) REFERENCES SECTIE(ID_SECTIE)
);
```

```
CREATE TABLE SPECIALIZARE(
    ID_SPECIALIZARE INT NOT NULL,
    ID_SPECIALIZARE_MAMA INT,
    DENUMIRE VARCHAR2(20),
    CONSTRAINT PK_SPECIALIZARE PRIMARY KEY (ID_SPECIALIZARE),
    CONSTRAINT FK_SPECIALIZARE FOREIGN KEY (ID_SPECIALIZARE_MAMA) REFERENCES
    SPECIALIZARE(ID_SPECIALIZARE)
);
```

CREATE OR REPLACE TYPE TIP_ALERGII AS VARRAY(15) OF VARCHAR2(20);

/

```
CREATE TABLE PACIENT(
    ID_PACIENT INT NOT NULL,
    ID_SPITAL INT,
    NUME_PACIENT VARCHAR2(20),
    PRENUME_PACIENT VARCHAR2(20),
    VARSTA NUMBER(3),
```

```
SEX VARCHAR2(10),  
LISTA_ALERGII TIP_ALERGII,  
CONSTRAINT PK_PACIENT PRIMARY KEY (ID_PACIENT),  
CONSTRAINT FK_PACIENT FOREIGN KEY (ID_SPITAL) REFERENCES SPITAL(ID_SPITAL)  
);  
  
ALTER TABLE PACIENT  
ADD NR_TELEFON VARCHAR2(10);  
  
CREATE TABLE PROGRAMARE(  
    ID_PROGRAMARE INT NOT NULL,  
    ID_PACIENT INT,  
    ID_SECTIE INT,  
    DATA_PROGRAMARE TIMESTAMP,  
    CONSTRAINT PK_PROGRAMARE PRIMARY KEY (ID_PROGRAMARE),  
    CONSTRAINT FK1_PROGRAMARE FOREIGN KEY (ID_PACIENT) REFERENCES PACIENT(ID_PACIENT),  
    CONSTRAINT FK2_PROGRAMARE FOREIGN KEY (ID_SECTIE) REFERENCES SECTIE(ID_SECTIE)  
);  
  
CREATE TABLE DOCTOR(  
    ID_DOCTOR INT NOT NULL,  
    ID_SPECIALIZARE INT,  
    ID_SECTIE INT,  
    ID_SPITAL INT,  
    NUME_DOCTOR VARCHAR2(20),  
    PRENUME_DOCTOR VARCHAR2(20),  
    VARSTA NUMBER(3),  
    SEX VARCHAR2(10),  
    NR_TELEFON VARCHAR2(10),  
    SALARIU NUMBER(5),  
    CONSTRAINT PK_DOCTOR PRIMARY KEY (ID_DOCTOR),
```

```

CONSTRAINT FK1_DOCTOR FOREIGN KEY (ID_SPECIALIZARE) REFERENCES
SPECIALIZARE(ID_SPECIALIZARE),

CONSTRAINT FK2_DOCTOR FOREIGN KEY (ID_SECTIE) REFERENCES SECTIE(ID_SECTIE),

CONSTRAINT FK3_DOCTOR FOREIGN KEY (ID_SPITAL) REFERENCES SPITAL(ID_SPITAL)

);

CREATE TABLE PROGRAM_DOCTOR(
    ID_PROGRAM_DOCTOR INT NOT NULL,
    ID_DOCTOR INT,
    START_ORA NUMBER(2),
    END_ORA NUMBER(2),
    ZI VARCHAR2(10),
    CONSTRAINT PK_PROGRAM_DOCTOR PRIMARY KEY (ID_PROGRAM_DOCTOR),
    CONSTRAINT FK_PROGRAM_DOCTOR FOREIGN KEY (ID_DOCTOR) REFERENCES DOCTOR(ID_DOCTOR)
);

CREATE TABLE OPERATIE(
    ID_OPERATIE INT NOT NULL,
    START_ORA NUMBER(2),
    SALA_OPERATIE NUMBER(3),
    CONSTRAINT PK_OPERATIE PRIMARY KEY (ID_OPERATIE)
);

CREATE TABLE DOCTOR_OPERATIE(
    ID_DOCTOR_OPERATIE INT NOT NULL,
    ID_OPERATIE INT,
    ID_DOCTOR INT,
    CONSTRAINT PK_DOCTOR_OPERATIE PRIMARY KEY (ID_DOCTOR_OPERATIE),
    CONSTRAINT FK1_DOCTOR_OPERATIE FOREIGN KEY (ID_OPERATIE) REFERENCES
OPERATIE(ID_OPERATIE),
    CONSTRAINT FK2_DOCTOR_OPERATIE FOREIGN KEY (ID_DOCTOR) REFERENCES DOCTOR(ID_DOCTOR)
);

```

```
CREATE TABLE INTERNARE(
    ID_INTERNARE INT NOT NULL,
    ID_SECTIE INT,
    ID_PACIENT INT,
    ID_SPITAL,
    CONSTRAINT PK_INTERNARE PRIMARY KEY (ID_INTERNARE),
    CONSTRAINT FK1_INTERNARE FOREIGN KEY (ID_SECTIE) REFERENCES SECTIE(ID_SECTIE),
    CONSTRAINT FK2_INTERNARE FOREIGN KEY (ID_PACIENT) REFERENCES PACIENT(ID_PACIENT),
    CONSTRAINT FK3_INTERNARE FOREIGN KEY (ID_SPITAL) REFERENCES SPITAL(ID_SPITAL)
);
```

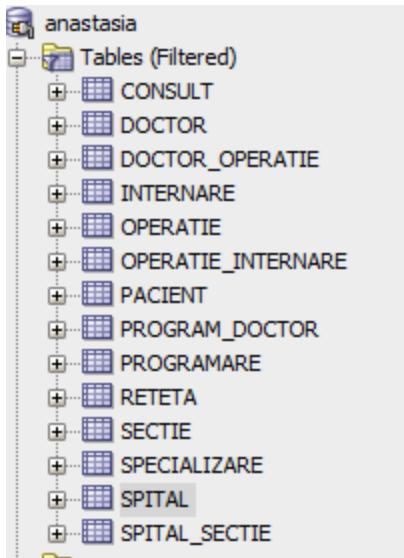
```
ALTER TABLE INTERNARE
ADD STATUS NUMBER(1);
```

```
ALTER TABLE INTERNARE
ADD DATA_INTERNARE DATE;
```

```
CREATE TABLE OPERATIE_INTERNARE(
    ID_OPERATIE_INTERNARE INT NOT NULL,
    ID_INTERNARE INT,
    ID_OPERATIE INT,
    CONSTRAINT PK_OPERATIE_INTERNARE PRIMARY KEY (ID_OPERATIE_INTERNARE),
    CONSTRAINT FK1_OPERATIE_INTERNARE FOREIGN KEY (ID_INTERNARE) REFERENCES
INTERNARE(ID_INTERNARE),
    CONSTRAINT FK2_OPERATIE_INTERNARE FOREIGN KEY (ID_OPERATIE) REFERENCES
OPERATIE(ID_OPERATIE)
);
```

```
CREATE TABLE CONSULT(
    ID_CONSULT INT NOT NULL,
    ID_INTERNARE INT,
```

```
ID_DOCTOR INT,
CONSTRAINT PK_CONSULT PRIMARY KEY (ID_CONSULT),
CONSTRAINT FK1_CONSULT FOREIGN KEY (ID_INTERNARE) REFERENCES INTERNARE(ID_INTERNARE),
CONSTRAINT FK2_CONSULT FOREIGN KEY (ID_DOCTOR) REFERENCES DOCTOR(ID_DOCTOR)
);
CREATE OR REPLACE TYPE TIP_MEDICAMENTE AS VARRAY(15) OF VARCHAR2(20);
/
CREATE TABLE RETETA(
ID_RETETA INT NOT NULL,
ID_DOCTOR INT,
ID_PACIENT INT,
ID_CONSULT INT,
LISTA_MEDICAMENTE TIP_MEDICAMENTE,
CONSTRAINT PK_RETETA PRIMARY KEY (ID_RETETA),
CONSTRAINT FK1_RETETA FOREIGN KEY (ID_DOCTOR) REFERENCES DOCTOR(ID_DOCTOR),
CONSTRAINT FK2_RETETA FOREIGN KEY (ID_PACIENT) REFERENCES PACIENT(ID_PACIENT),
CONSTRAINT FK3_RETETA FOREIGN KEY (ID_CONSULT) REFERENCES CONSULT(ID_CONSULT)
);
ALTER TABLE RETETA
DROP COLUMN ID_DOCTOR;
ALTER TABLE RETETA
DROP COLUMN ID_PACIENT;
```



EXERCITIU 5

INSERT INTO SPITAL

```
VALUES (1,'Spitalul Floreasca','Calea Floreasca 8','Bucuresti',1934,'0798786514');
```

INSERT INTO SPITAL

```
VALUES (2,'Spitalul Fundeni','Soseaua Fundeni 258','Bucuresti',1959,'0865379856');
```

INSERT INTO SPITAL

```
VALUES (3,'Spitalul Judetean','Strada Crișan 9','Slatina',1903,'0746816827');
```

INSERT INTO SPITAL

```
VALUES (4,'Spitalul Sf Ioan','Soseaua Vitan-Bârzești 13','Bucuresti',1979,'0789657812');
```

INSERT INTO SPITAL

```
VALUES (5,'Spitalul Mocrea','Soseaua Primaverii 19','Arad',1979,'0789657812');
```

INSERT INTO SPITAL

```
VALUES (6,'Spitalul Sf Ioan','Strada Vederii 28','Timisoara',1956,'0789984322');
```

INSERT INTO SECTIE

```
VALUES (1,'Cardiologie',1);
```

```
INSERT INTO SECTIE
VALUES (2,'Neurologie',2);
INSERT INTO SECTIE
VALUES (3,'Pediatrie',3);
INSERT INTO SECTIE
VALUES (4,'Oftalmologie',4);
INSERT INTO SECTIE
VALUES (5,'Pneumonie',5);
INSERT INTO SECTIE
VALUES (6,'Chirurgie',5);
INSERT INTO SECTIE
VALUES (7,'Oncologie',5);
```

```
INSERT INTO SPITAL_SECTIE
VALUES (1,1,2);
INSERT INTO SPITAL_SECTIE
VALUES (2,1,3);
INSERT INTO SPITAL_SECTIE
VALUES (3,1,4);
INSERT INTO SPITAL_SECTIE
VALUES (4,1,5);
INSERT INTO SPITAL_SECTIE
VALUES (5,2,3);
INSERT INTO SPITAL_SECTIE
VALUES (6,3,5);
INSERT INTO SPITAL_SECTIE
VALUES (7,3,4);
```

```
INSERT INTO SPITAL_SECTIE  
VALUES (8,4,1);  
  
INSERT INTO SPITAL_SECTIE  
VALUES (9,5,2);  
  
INSERT INTO SPITAL_SECTIE  
VALUES (10,5,3);
```

```
INSERT INTO SPECIALIZARE  
VALUES (1,NULL,'Cardiologie');  
  
INSERT INTO SPECIALIZARE  
VALUES (2,1,'Cardio-oncologie');  
  
INSERT INTO SPECIALIZARE  
VALUES (3,NULL,'Oncologie');  
  
INSERT INTO SPECIALIZARE  
VALUES (4,3,'Oncologie pediatrica');  
  
INSERT INTO SPECIALIZARE  
VALUES (5,NULL,'Chirurgie');  
  
INSERT INTO SPECIALIZARE  
VALUES (6,5,'Chirurgie vasculara');
```

```
DECLARE  
alergii TIP_ALERGII:=TIP_ALERGII();  
  
BEGIN  
alergii.extend;  
alergii(1):='Cardiovit';  
  
INSERT INTO PACIENT  
VALUES (1,2,'Sandu','Anastasia',20,'Feminin',alergii,'0746816827');
```

```
alergii(1):='Panangin';
INSERT INTO PACIENT
VALUES (2,3,'Danescu','Adela',20,'Feminin',alergii,'0760283936');

alergii(1):='Magnerot';
INSERT INTO PACIENT
VALUES (3,5,'Mihailescu','Teodor',20,'Masculin',alergii,'0787353720');

alergii(1):='Aspenter';
INSERT INTO PACIENT
VALUES (4,1,'Dirtu','Ecaterina',20,'Feminin',alergii,'0745382125');

alergii(1):='Cardiovit';
alergii.extend;
alergii(2):='Aspenter';
INSERT INTO PACIENT
VALUES (5,4,'Harnagea','Andrei',20,'Masculin',alergii,'0760856721');

INSERT INTO PACIENT
VALUES (6,5,'Oancea','Antonia',20,'Feminin',NULL,'0760324721');

INSERT INTO PACIENT
VALUES (7,3,'Cristea','Eduard',20,'Masculin',NULL,'0456889827');

INSERT INTO PACIENT
VALUES (8,4,'Petrescu','Alexandru',20,'Masculin',NULL,'0843215643');

INSERT INTO PACIENT
VALUES (9,2,'Pita','Bogdan',20,'Masculin',NULL,'0654329831');

END;

INSERT INTO PROGRAMARE
VALUES (1,5,1,TO_DATE('20/01/2023:12:00:00PM', 'DD-MM-YYYY HH:MI:SSPM'));
```

```
INSERT INTO PROGRAMARE  
VALUES (2,4,2,TO_DATE('08/02/2023:08:00:00PM', 'DD-MM-YYYY HH:MI:SSPM'));  
INSERT INTO PROGRAMARE  
VALUES (3,3,3,TO_DATE('27/02/2023:04:30:00PM', 'DD-MM-YYYY HH:MI:SSPM'));  
INSERT INTO PROGRAMARE  
VALUES (4,2,4,TO_DATE('07/03/2023:03:45:00PM', 'DD-MM-YYYY HH:MI:SSPM'));  
INSERT INTO PROGRAMARE  
VALUES (5,1,5,TO_DATE('30/04/2024:06:00:00PM', 'DD-MM-YYYY HH:MI:SSPM'));  
INSERT INTO PROGRAMARE  
VALUES (6,6,7,TO_DATE('01/09/2023:12:00:00PM', 'DD-MM-YYYY HH:MI:SSPM'));  
INSERT INTO PROGRAMARE  
VALUES (7,8,6,TO_DATE('17/08/2023:01:00:00PM', 'DD-MM-YYYY HH:MI:SSPM'));  
INSERT INTO PROGRAMARE  
VALUES (8,4,3,TO_DATE('28/06/2023:09:00:00PM', 'DD-MM-YYYY HH:MI:SSPM'));  
INSERT INTO PROGRAMARE  
VALUES (9,9,1,TO_DATE('11/02/2023:01:00:00PM', 'DD-MM-YYYY HH:MI:SSPM'));
```

```
INSERT INTO DOCTOR  
VALUES (1,1,1,1,'Gandore','Mihai',39,'Masculin','0746816827',1500);  
INSERT INTO DOCTOR  
VALUES (2,1,1,2,'Nistor','George',47,'Masculin','0767983467',1750);  
INSERT INTO DOCTOR  
VALUES (3,3,6,4,'Stanciu','Sergiu',31,'Masculin','0745783568',2400);  
INSERT INTO DOCTOR  
VALUES (4,4,6,1,'Hurloiu','Selena',39,'Feminin','0798672312',1350);  
INSERT INTO DOCTOR  
VALUES (5,6,7,5,'Alex','Andrei',55,'Masculin','0776654123',2000);  
INSERT INTO DOCTOR
```

```
VALUES (6,2,5,2,'Florea','George',28,'Masculin','0746754827',1700);
```

```
INSERT INTO OPERATIE
```

```
VALUES (1,10,77);
```

```
INSERT INTO OPERATIE
```

```
VALUES (2,12,103);
```

```
INSERT INTO OPERATIE
```

```
VALUES (3,08,271);
```

```
INSERT INTO OPERATIE
```

```
VALUES (4,15,41);
```

```
INSERT INTO OPERATIE
```

```
VALUES (5,14,210);
```

```
INSERT INTO DOCTOR_OPERATIE
```

```
VALUES (1,2,3);
```

```
INSERT INTO DOCTOR_OPERATIE
```

```
VALUES (2,3,4);
```

```
INSERT INTO DOCTOR_OPERATIE
```

```
VALUES (3,5,1);
```

```
INSERT INTO DOCTOR_OPERATIE
```

```
VALUES (4,1,5);
```

```
INSERT INTO DOCTOR_OPERATIE
```

```
VALUES (5,2,2);
```

```
INSERT INTO DOCTOR_OPERATIE
```

```
VALUES (6,5,2);
```

```
INSERT INTO DOCTOR_OPERATIE  
VALUES (7,1,1);  
  
INSERT INTO DOCTOR_OPERATIE  
VALUES (8,4,5);  
  
INSERT INTO DOCTOR_OPERATIE  
VALUES (9,5,4);  
  
INSERT INTO DOCTOR_OPERATIE  
VALUES (10,3,3);
```

```
INSERT INTO INTERNARE  
VALUES (1,2,5,3,1,TO_DATE('20/01/2023', 'DD-MM-YYYY'));  
  
INSERT INTO INTERNARE  
VALUES (2,5,1,4,1,TO_DATE('17/02/2023', 'DD-MM-YYYY'));  
  
INSERT INTO INTERNARE  
VALUES (3,4,2,5,0,NULL);  
  
INSERT INTO INTERNARE  
VALUES (4,1,3,2,1,TO_DATE('07/07/2023', 'DD-MM-YYYY'));  
  
INSERT INTO INTERNARE  
VALUES (5,3,4,1,0,NULL);
```

```
INSERT INTO OPERATIE_INTERNARE  
VALUES (1,2,5);  
  
INSERT INTO OPERATIE_INTERNARE  
VALUES (2,3,4);  
  
INSERT INTO OPERATIE_INTERNARE
```

```
VALUES (3,4,2);
INSERT INTO OPERATIE_INTERNARE
VALUES (4,1,3);
INSERT INTO OPERATIE_INTERNARE
VALUES (5,5,1);
INSERT INTO OPERATIE_INTERNARE
VALUES (6,4,2);
INSERT INTO OPERATIE_INTERNARE
VALUES (7,3,1);
INSERT INTO OPERATIE_INTERNARE
VALUES (8,2,5);
INSERT INTO OPERATIE_INTERNARE
VALUES (9,1,3);
INSERT INTO OPERATIE_INTERNARE
VALUES (10,3,4);
```

```
INSERT INTO CONSULT
VALUES (1,2,5);
INSERT INTO CONSULT
VALUES (2,3,4);
INSERT INTO CONSULT
VALUES (3,5,2);
INSERT INTO CONSULT
VALUES (4,1,1);
INSERT INTO CONSULT
VALUES (5,4,3);
```

DECLARE

medicamente TIP_MEDICAMENTE:=TIP_MEDICAMENTE();

BEGIN

medicamente.extend;

medicamente(1):='Cardiovit';

medicamente.extend;

medicamente(2):='Aspenter';

medicamente.extend;

medicamente(3):='Magnerot';

medicamente.extend;

medicamente(4):='Protecardin';

INSERT INTO RETETA

VALUES (1,3,medicamente);

medicamente(1):='Aspenter';

medicamente(2):='Panangin';

medicamente(3):='Thrombo';

medicamente(4):='Dipiridamol';

medicamente.extend;

medicamente(5):='Helcor';

INSERT INTO RETETA

VALUES (2,4,medicamente);

medicamente(1):='Rompirlin';

medicamente(2):='Santepirin';

medicamente(3):='Phelodia';

medicamente(4):='Aspacardin';

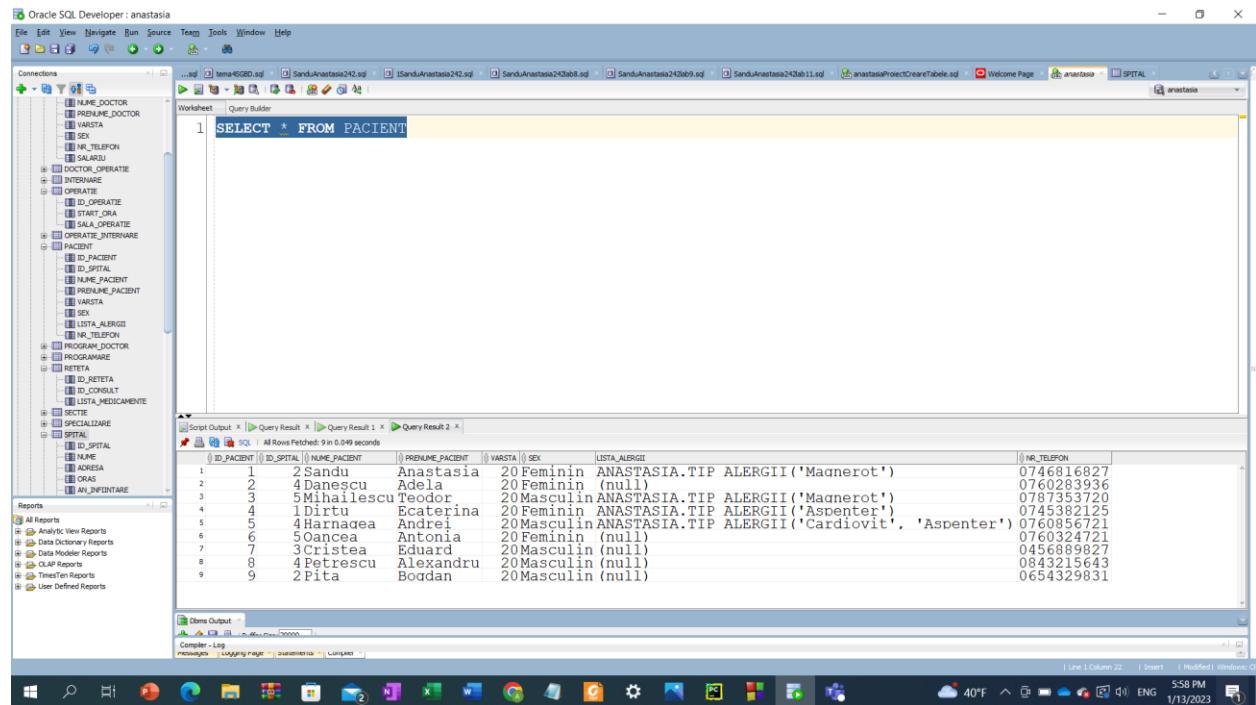
medicamente(5):='Aspimax';

medicamente.extend;

```
medicamente(6):='Magnerot';
INSERT INTO RETETA
VALUES (3,1,medicamente);
medicamente(1):='Meglucon';
medicamente(2):='Glucophage';
medicamente(3):='Meglucon';
medicamente(4):='Metformax';
medicamente(5):='Ibuprofen';
medicamente(6):='Belformin';
medicamente.extend;
medicamente.extend;
medicamente(7):='Siofor';
medicamente(8):='Aglurab';
INSERT INTO RETETA
VALUES (4,5,medicamente);
medicamente(1):='Glucovance';
medicamente(2):='Gluformin';
medicamente(3):='Metformin';
medicamente(4):='Neomid';
medicamente(5):='Brot';
medicamente(6):='Glibomet';
medicamente(7):='Normaglyc';
medicamente(8):='Langerin';
INSERT INTO RETETA
VALUES (5,2,medicamente);
END;
alter table reteta
drop column id_pacient
alter table reteta
```

```
drop column id_doctor
```

```
INSERT INTO PROGRAM_DOCTOR  
VALUES (1,5,10,20,'FRIDAY');  
  
INSERT INTO PROGRAM_DOCTOR  
VALUES (2,3,08,16,'THURSDAY');  
  
INSERT INTO PROGRAM_DOCTOR  
VALUES (3,2,12,22,'SATURDAY');  
  
INSERT INTO PROGRAM_DOCTOR  
VALUES (4,4,14,02,'MONDAY');  
  
INSERT INTO PROGRAM_DOCTOR  
VALUES (5,1,12,22,'WEDNESDAY');
```



The screenshot shows the Oracle SQL Developer interface with a query window open. The query is:

```
SELECT * FROM PACIENT
```

The results are displayed in a grid:

ID_PACIENT	ID_SPITAL	NUME_PACIENT	PRENUME_PACIENT	VARSTA	LISTA_ALERGII	NR_TELEFON
1	1	2Sandu	Anastasia	20	Feminin ANASTASIA.TIP ALERGII('Magnerot')	0746816827
2	2	4Danescu	Adela	20	Feminin (null)	0760283936
3	3	5Mihailescu	Teodor	20	Masculin ANASTASIA.TIP ALERGII('Magnerot')	0781353720
4	4	1Dumitrescu	Ecatezina	20	Feminin ANASTASIA.TIP ALERGII('Aspenter')	0745898225
5	5	4Hanganega	Andrei	20	Masculin ANASTASIA.TIP ALERGII('Cardiovit'. 'Aspenter')	0760326721
6	6	5Oancea	Aronia	20	Feminin (null)	0760324721
7	7	3Cristea	Eduard	20	Feminin (null)	0456899827
8	8	4Petrescu	Alexandru	20	Masculin (null)	0843215643
9	9	2Pita	Bogdan	20	Masculin (null)	0654329831

Oracle SQL Developer : anastasia

File Edit View Navigate Run Source Team Tools Window Help

Connections SPITAL

Worksheet Query Builder

```
1 SELECT * FROM SPITAL
```

Script Output | Query Result | Query Result 1 | Query Result 2

ID_SPITAL	NAME	ADRESA	ORAS	AN_INFINTARE	NR_TELEFON
1	Spitalul Floreasca	Calea Floreasca 8	Bucuresti	19340798786514	
2	Spitalul Fundeni	Soseaua Fundeni 258	Bucuresti	19030746816827	
3	Spitalul Judetean	Strada Crisan 9	Slatina	19030746816827	
4	Spitalul Sf Ioan	Soseaua Vitan-Bärzesti 13	Bucuresti	19790789657812	
5	Spitalul Mocrea	Soseaua Primaverii 19	Arad	19790789657812	
6	Spitalul Sf Ioan	Strada Vederii 28	Timisoara	19560789984322	

Doms Output | All Rows Fetched 6 in 0.049 seconds

Compiler Log | Logging Page | Database | Complex |

I Line | Column 21 | Insert | Modified | Windows | 5:59 PM | 1/13/2023

Oracle SQL Developer : anastasia

File Edit View Navigate Run Source Team Tools Window Help

Connections SPITAL

Worksheet Query Builder

```
1 SELECT * FROM PROGRAMARE
```

Script Output | Query Result | Query Result 1 | Query Result 2

ID_PROGRAMARE	ID_PACIENT	ID_SECTIE	DATA_PROGRAMARE
1	1	5	120-JAN-23 12.00.00.000000000 PM
2	2	4	208-FEB-23 08.00.00.000000000 PM
3	3	3	327-FEB-23 04.30.00.000000000 PM
4	4	2	407-MAR-23 03.45.00.000000000 PM
5	5	1	530-APR-24 06.00.00.000000000 PM
6	6	7	705-SEP-23 01.00.00.000000000 PM
7	7	8	607-AUG-23 01.00.00.000000000 PM
8	8	4	328-JUN-23 09.00.00.000000000 PM
9	9	9	111-FEB-23 01.00.00.000000000 PM

Doms Output | All Rows Fetched 9 in 0.049 seconds

Compiler Log | Logging Page | Database | Complex |

I Line | Column 25 | Insert | Modified | Windows | 5:59 PM | 1/13/2023

Oracle SQL Developer : anastasia

File Edit View Navigate Run Source Team Tools Window Help

Connections

Worksheet Query Builder

```
1 SELECT * FROM OPERATIE_INTERNARE
```

Script Output x | Query Result x | Query Result 1 x | Query Result 2 x

ID_OPERATIE_INTERNARE	ID_INTERNARE	ID_OPERATE
1	1	5
2	2	3
3	3	4
4	4	1
5	5	1
6	6	2
7	7	3
8	8	2
9	9	1
10	10	3

Doms Output x | All Rows Fetched: 10 in 0.078 seconds

Compiler Log | Logging Page | Database | Compose |

Reports

File Edit View Navigate Run Source Team Tools Window Help

Connections

Worksheet Query Builder

```
149
150
151
152
153
154
155
156
157
158
159 SELECT * FROM SPITAL_SECTIE
```

Script Output x | Query Result x | Explain Plan x | Query Result 2 x

ID_SPITAL_SECTIE	ID_SPITAL	ID_SECTIE
1	1	2
2	2	1
3	3	1
4	4	1
5	5	2
6	6	3
7	7	3
8	8	4
9	9	5
10	10	5

Doms Output x | All Rows Fetched: 10 in 0.011 seconds

Compiler Log | Logging Page | Database | Compose |

Oracle SQL Developer : anastasia

File Edit View Navigate Run Source Team Tools Window Help

Connections

Worksheet Query Builder

```
149
150
151
152
153
154
155
156
157
158
159 SELECT * FROM SPITAL_SECTIE
```

Script Output x | Query Result x | Explain Plan x | Query Result 2 x

ID_SPITAL_SECTIE	ID_SPITAL	ID_SECTIE
1	1	2
2	2	1
3	3	1
4	4	1
5	5	2
6	6	3
7	7	3
8	8	4
9	9	5
10	10	5

Doms Output x | All Rows Fetched: 10 in 0.011 seconds

Compiler Log | Logging Page | Database | Compose |

Reports

Oracle SQL Developer : anastasia

Connections Worksheet Query Builder

```

149
150
151
152
153
154
155
156
157
158
159 SELECT * FROM RETETA
160
161

```

Script Output | Query Result | Explain Plan | Query Result 2

ID_RETETA	ID_CONSULT	LISTA_MEDICAMENTE
1	1	3ANASTASIA.TIP MEDICAMENTE('Cardiovit', 'Aspenter', 'Magnerot', 'Proteccardin')
2	2	4ANASTASIA.TIP MEDICAMENTE('Aspenter', 'Panadol', 'Thrombo', 'Dipiridamol', 'Helcor')
3	3	1ANASTASIA.TIP MEDICAMENTE('Rompirlin', 'Santepirin', 'Phelodia', 'Aspacardin', 'Aspirimax', 'Magnerot')
4	4	5ANASTASIA.TIP MEDICAMENTE('Meclucon', 'Glucophage', 'Meclucon', 'Metformax', 'Ibuprofen', 'Belformin')
5	5	2ANASTASIA.TIP MEDICAMENTE('Glucovalence', 'Gluformin', 'Metformin', 'Neomid', 'Brot', 'Glibomet', 'Norma')

Doms Output | All Rows Parsed: 5 in 0.072 seconds

Compiler - Log | Logging Page | Database | Compiler

Line 159 Column 21 | Insert | Modified | Windows: 0

9:08 PM 1/13/2023

Oracle SQL Developer : anastasia

Connections Worksheet Query Builder

```

149
150
151
152
153
154
155
156
157
158
159 SELECT * FROM CONSULT
160
161

```

Script Output | Query Result | Explain Plan | Query Result 2

ID_CONSULT	ID_INTERNARE	ID_DOCTOR
1	1	5
2	2	3
3	3	4
4	4	1
5	5	2

Doms Output | All Rows Parsed: 5 in 0.009 seconds

Compiler - Log | Logging Page | Database | Compiler

Line 159 Column 22 | Insert | Modified | Windows: 0

9:08 PM 1/13/2023

Oracle SQL Developer : anastasia

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections

Worksheet Query Builder

```

149
150
151
152
153
154
155
156
157
158
159 SELECT * FROM OPERATIE
160
161

```

Script Output x | Query Result x | Explain Plan x | Query Result 2 x

ID_OPERATIE	START_ORA	SALA_OPERATIE
1	3	10
2	2	103
3	3	18
4	4	41
5	5	14
6	6	200

Doms Output: All Rows Fetched 6 in 0.03 seconds

Compiler Log messages Logging mode: Database

Line 159 Column 1 Insert Modified Windows: O 9:09 PM 1/13/2023

Oracle SQL Developer : anastasia

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections

Worksheet Query Builder

```

149
150
151
152
153
154
155
156
157
158
159 SELECT * FROM PROGRAM_DOCTOR
160
161

```

Script Output x | Query Result x | Explain Plan x | Query Result 2 x

ID_PROGRAM_DOCTOR	ID_DOCTOR	START_ORA	END_ORA	ZI
1	1	5	10	20 FRIDAY
2	2	3	8	16 THURSDAY
3	3	2	12	14 SATURDAY
4	4	4	14	2 MONDAY
5	5	1	12	22 WEDNESDAY

Doms Output: All Rows Fetched 5 in 0.049 seconds

Compiler Log messages Logging mode: Database

Line 159 Column 29 Insert Modified Windows: O 9:10 PM 1/13/2023

Oracle SQL Developer : anastasia

Connections Worksheet Query Builder

```

149
150
151
152
153
154
155
156
157
158
159 SELECT * FROM PROGRAMARE
160
161

```

Script Output | Query Result | Query Result 1 | Explain Plan | Query Result 2 |

ID_PROGRAMARE	ID_PACIENT	ID_SECTIE	DATA_PROGRAMARE
1	3	5	120-JAN-23 12:00:00.000000000 PM
2	4	4	208-FEB-23 08:00:00.000000000 PM
3	3	3	327-FEB-23 04:30:00.000000000 PM
4	4	2	407-MAR-23 03:45:00.000000000 PM
5	5	1	530-APR-24 06:00:00.000000000 PM
6	6	6	701-SEP-23 12:00:00.000000000 PM
7	7	8	617-AUG-23 01:00:00.000000000 PM
8	8	4	328-JUN-23 09:00:00.000000000 PM
9	9	9	111-FEB-23 01:00:00.000000000 PM

Doms Output | Compiler Log | Logging Page | Compose |

Line 159 Column 25 | Insert | Modified | Windows: C | 9:10 PM | 1/13/2023

Oracle SQL Developer : anastasia

Connections Worksheet Query Builder

```

149
150
151
152
153
154
155
156
157
158
159 SELECT * FROM INTERNARE
160
161

```

Script Output | Query Result | Query Result 1 | Explain Plan | Query Result 2 |

ID_INTERNARE	ID_SECTIE	ID_PACIENT	ID_SPITAL	STATUS	DATA_INTERNARE
1	1	5	3	120-JAN-23	
2	2	5	1	4	117-FEB-23
3	3	4	2	5	0 (null)
4	4	1	3	2	107-JUL-23
5	5	3	4	1	0 (null)

Doms Output | Compiler Log | Logging Page | Compose |

Line 159 Column 24 | Insert | Modified | Windows: C | 9:11 PM | 1/13/2023

Oracle SQL Developer : anastasia

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections

Worksheet Query Builder

```
149
150
151
152
153
154
155
156
157
158
159 SELECT * FROM SPECIALIZARE
160
161
```

Script Output x | Query Result x | Explain Plan x | Query Result 2 x

ID_SPECIALIZARE	ID_SPECIALIZARE_MAMA	DENUMIRE
1	(null)	Cardiologie
2	1	Cardio-oncologie
3	(null)	Oncologie
4	3	Oncologie pediatrica
5	5	(null) Chirurgie
6	6	Chirurgie vasculara

Doms Output: All Rows Fetched: 6 in 0.049 seconds

Compiler - Log messages Logging message Compiler

Line 159 Column 27 Insert Modified Windows 9:11 PM ENG 1/13/2023

Oracle SQL Developer : anastasia

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections

Worksheet Query Builder

```
149
150
151
152
153
154
155
156
157
158
159 SELECT * FROM SECTIE
160
161
```

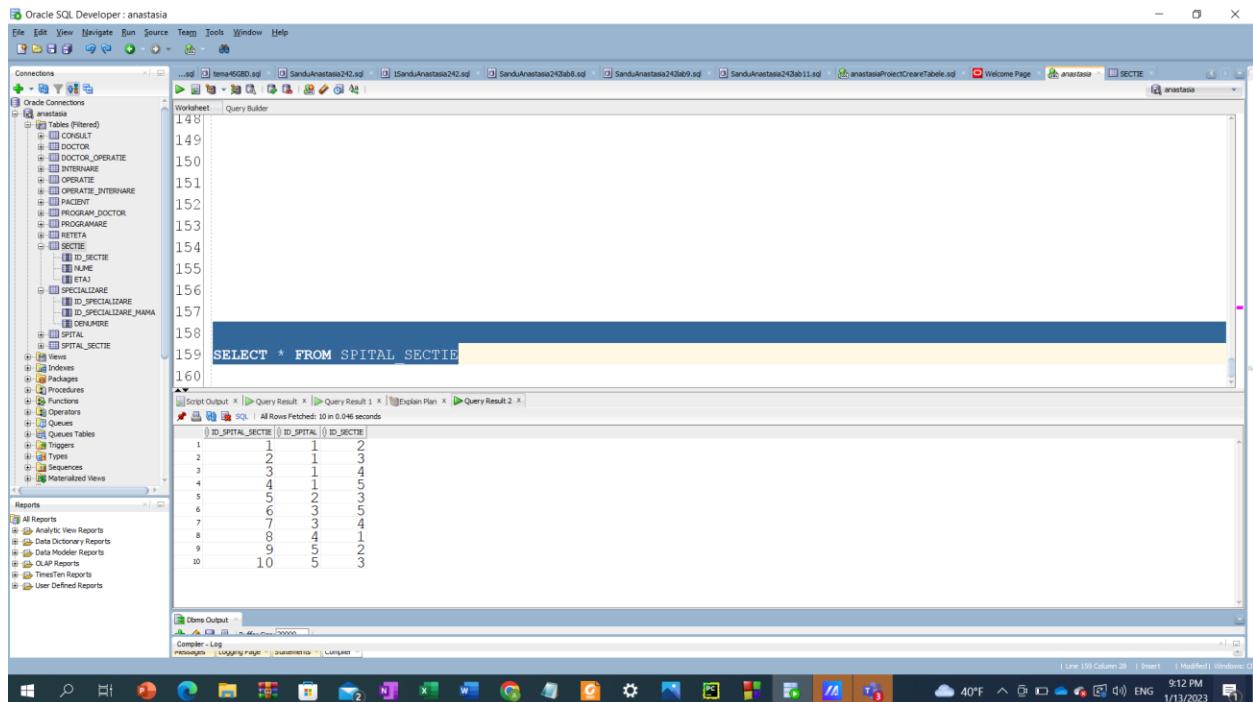
Script Output x | Query Result x | Explain Plan x | Query Result 2 x

ID_SECTIE	NAME	ETAJ
1	Cardiologie	1
2	Neurologie	2
3	Pediatrie	3
4	Oftalmologie	4
5	Pneumonie	5
6	Chirurgie	5
7	Oncologie	5

Doms Output: All Rows Fetched: 7 in 0.048 seconds

Compiler - Log messages Logging message Compiler

Line 159 Column 21 Insert Modified Windows 9:11 PM ENG 1/13/2023



EXERCITIU 6

--Pentru numele dat al unui spital sa se afiseze toti doctorii care lucreaza la spitalul respectiv

--impreuna cu programele lor.

CREATE OR REPLACE PROCEDURE P6(DENUMIRE SPITAL.NUME%TYPE)

IS

```

TYPE ORE_VECTOR IS VARRAY(100) OF NUMBER(2);

TYPE ZI_VECTOR IS VARRAY(100) OF VARCHAR2(10);

TYPE DOCTOR_RECORD IS RECORD
  (NUME DOCTOR.NUME_DOCTOR%TYPE,
  PRENUME DOCTOR.PRENUME_DOCTOR%TYPE,
  START_ORA ORE_VECTOR,
  END_ORA ORE_VECTOR,
  ZI ZI_VECTOR);
  
```

```

TYPE DOCTOR_VECTOR IS VARRAY(100) OF DOCTOR_RECORD;
T DOCTOR_VECTOR:= DOCTOR_VECTOR();
NUME_SPITAL SPITAL.NUME%TYPE;
NR NUMBER(2);

CURSOR c IS
  SELECT NUME_DOCTOR, PRENUME_DOCTOR
    FROM SPITAL S
   JOIN DOCTOR D ON (D.ID_SPITAL=S.ID_SPITAL)
   JOIN PROGRAM_DOCTOR P ON (D.ID_DOCTOR=P.ID_DOCTOR)
  WHERE NUME=DENUMIRE;

BEGIN
  SELECT NUME
    INTO NUME_SPITAL
    FROM SPITAL
   WHERE NUME=DENUMIRE;
  DBMS_OUTPUT.PUT_LINE('Spitalul ales este : ' || NUME_SPITAL);

  SELECT COUNT(NUME_DOCTOR)
    INTO NR
    FROM DOCTOR D
   JOIN SPITAL S ON (D.ID_SPITAL=S.ID_SPITAL)
  WHERE NUME=DENUMIRE;

  IF NR=0 THEN
    DBMS_OUTPUT.PUT_LINE('Nu lucreaza doctori la acest spital.');
  ELSIF NR=1 THEN
    DBMS_OUTPUT.PUT_LINE('Numarul de doctori care lucreaza in acest spital este : ' || NR);
    DBMS_OUTPUT.PUT_LINE('Doctorul si programul lui sunt : ');

```

```

ELSE
    DBMS_OUTPUT.PUT_LINE('Numarul de doctori care lucreaza in acest spital este : ' || NR);
    DBMS_OUTPUT.PUT_LINE('Doctorii si programul lor sunt : ');
    END IF;

FOR C_DOCTOR IN C LOOP
    T.EXTEND;
    T(T.LAST).NUME:=C_DOCTOR.NUME_DOCTOR;
    T(T.LAST).PRENUME:=C_DOCTOR.PRENUME_DOCTOR;

    SELECT START_ORA, END_ORA, ZI
    BULK COLLECT INTO
        T(T.LAST).START_ORA, T(T.LAST).END_ORA, T(T.LAST).ZI
    FROM SPITAL S
    JOIN DOCTOR D ON (D.ID_SPITAL=S.ID_SPITAL)
    JOIN PROGRAM_DOCTOR P ON (D.ID_DOCTOR=P.ID_DOCTOR)
    WHERE D.NUME_DOCTOR=C_DOCTOR.NUME_DOCTOR AND
    D.PRENUME_DOCTOR=C_DOCTOR.PRENUME_DOCTOR;
    END LOOP;

FOR I IN T.FIRST..T.LAST LOOP
    DBMS_OUTPUT.PUT_LINE(T(I).NUME || ' ' || T(I).PRENUME);
    FOR J IN T(I).START_ORA.FIRST..T(I).START_ORA.LAST LOOP
        DBMS_OUTPUT.PUT_LINE('In ziua de ' || T(I).ZI(J) || ' lucreaza de la ' || T(I).START_ORA(J) || ' la '
        || T(I).END_ORA(J));
    END LOOP;
    END LOOP;

EXCEPTION

```

```

WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20000,'Nu exista spitale cu numele dat');

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

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

END P6;
/
BEGIN
    P6('Spitalul Sf Ioan');
END;
/

```

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

- Connections:** A tree view of database objects under the schema "anastasia".
- Worksheet:** Displays the PL/SQL code for procedure P6. The code includes error handling for NO_DATA_FOUND, TOO_MANY_ROWS, and OTHERS exceptions, and a BEGIN block that calls P6 with the argument 'Spitalul Sf Ioan'.
- Status Bar:** Shows the message "PL/SQL procedure successfully completed."
- Script Output:** Shows the message "Procedure P6 compiled".
- Bottom Bar:** Includes standard Windows-style icons for file operations, search, and system status.

Oracle SQL Developer : anastasia

File Edit View Navigate Run Source Team Tools Window Help

Connections SPITAL anastasia

Worksheet Query Builder

```

76      RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
77  END P6;
78 /
79
80
81 BEGIN
82   P6('Spitalul Floreasca');
83 END;
84 /

```

Script Output x Query Result x Query Result 1 x Query Result 2 x

Procedure P6 compiled

Dbsn Output

anastasia x

```

Spitalul aies este : spitalui floreasca
Numarul de doctori care lucreaza in acest spital este : 2
Doctorii si programul lor sunt :
Gandore Mihai
In ziua de WEDNESDAY lucreaza de la 12 la 22
Hurloiu Selena
In ziua de MONDAY lucreaza de la 14 la 2

```

Compiler - Log messages Logging mode: Database... Compiler

Line 84 Column 2 Insert Modified Windows: C 6:12 PM 1/13/2023

Oracle SQL Developer : anastasia

File Edit View Navigate Run Source Team Tools Window Help

Connections SPITAL anastasia

Worksheet Query Builder

```

72      RAISE_APPLICATION_ERROR(-20000,'Nu exista spitale cu numele dat');
73      WHEN TOO_MANY_ROWS THEN
74        RAISE_APPLICATION_ERROR(-20001,'Există mai multe spitale cu numele dat');
75      WHEN OTHERS THEN
76        RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
77  END P6;
78 /
79
80
81 BEGIN
82   P6('Spitalul PROGETSGBD');
83 END;
84 /
85

```

Script Output x Query Result x Query Result 1 x Query Result 2 x

Error report -

```

ORA-20000: Nu exista spitale cu numele dat
ORA-06512: at "ANASTASIA.P6", line 68
ORA-06512: at line 2
20000. 00000 - "%s"
*Cause: The stored procedure 'raise_application_error'
was called which causes this error to be generated.

```

Dbsn Output

Compiler - Log messages Logging mode: Database... Compiler

Line 84 Column 2 Insert Modified Windows: C 6:13 PM 1/13/2023

Oracle SQL Developer : anastasia

Connections Worksheet Query Builder

```

72      RAISE_APPLICATION_ERROR(-20000,'Nu exista spitale cu numele dat');
73      WHEN TOO_MANY_ROWS THEN
74          RAISE_APPLICATION_ERROR(-20001,'Exista mai multe spitale cu numele dat');
75      WHEN OTHERS THEN
76          RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
77  END P6;
78 /
79
80
81 BEGIN
82     P6('Spitalul Sf Ioan');
83 END;
84 /
85

```

Script Output x | Query Result x | Query Result 1 x | Query Result 2 x

Reports Compiler Log Task completed in 0.102 seconds

Error report -
ORA-20001: Exista mai multe spitale cu numele dat
ORA-06512: at "ANASTASIA.P6", line 70
ORA-06512: at line 2

Doms Output

Compiler Log Logging Page Complex

File Edit View Navigate Run Source Team Tools Window Help

Oracle SQL Developer : anastasia

Connections Worksheet Query Builder

```

72      RAISE_APPLICATION_ERROR(-20000,'Nu exista spitale cu numele dat');
73      WHEN TOO_MANY_ROWS THEN
74          RAISE_APPLICATION_ERROR(-20001,'Exista mai multe spitale cu numele dat');
75      WHEN OTHERS THEN
76          RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
77  END P6;
78 /
79
80
81 BEGIN
82     P6('Spitalul Mocrea');
83 END;
84 /
85

```

Script Output x | Query Result x | Query Result 1 x | Query Result 2 x

Reports Compiler Log Task completed in 0.093 seconds

Error report -
ORA-20002: Alta eroare!
ORA-06512: at "ANASTASIA.P6", line 72
ORA-06512: at line 2

Doms Output

Compiler Log Logging Page Complex

File Edit View Navigate Run Source Team Tools Window Help

EXERCITIU 7

--Pentru numele unui spital dat, pentru toti pacientii programati la acest spital sa se afiseze toti doctorii care lucreaza in timpul programarii si la sectia lor

```

CREATE OR REPLACE PROCEDURE P7(DENUMIRE SPITAL.NUME%TYPE)
IS
    AUX_DATA_PROGRAMARE PROGRAMARE.DATA_PROGRAMARE%TYPE;
    AUX_COD_SECTIE PROGRAMARE.ID_SECTIE%TYPE;
    NUME DOCTOR.NUME_DOCTOR%TYPE;
    PRENUME DOCTOR.PRENUME_DOCTOR%TYPE;
    NUME_SPITAL SPITAL.NUME%TYPE;

CURSOR A IS
    SELECT DATA_PROGRAMARE, PROG.ID_SECTIE
    FROM SPITAL S
    INNER JOIN PACIENT P ON(S.ID_SPITAL=P.ID_SPITAL)
    INNER JOIN PROGRAMARE PROG ON(P.ID_PACIENT=PROG.ID_PACIENT)
    WHERE S.NUME=DENUMIRE;

CURSOR B(DATA_PROGRAMARE PROGRAMARE.DATA_PROGRAMARE%TYPE,COD_SECTIE
PROGRAMARE.ID_SECTIE%TYPE) IS
    SELECT NUME_DOCTOR, PRENUME_DOCTOR
    FROM DOCTOR D
    INNER JOIN PROGRAM_DOCTOR P ON(D.ID_DOCTOR=P.ID_DOCTOR)
    INNER JOIN SPITAL S ON(D.ID_SPITAL=S.ID_SPITAL)
    WHERE S.NUME = DENUMIRE AND D.ID_SECTIE=COD_SECTIE AND EXTRACT(HOUR FROM
DATA_PROGRAMARE) BETWEEN P.START_ORA AND P.END_ORA;

BEGIN
    SELECT NUME
    INTO NUME_SPITAL
    FROM SPITAL
    WHERE NUME=DENUMIRE;

```

```

DBMS_OUTPUT.PUT_LINE('Spitalul ales este ' || '' || NUME_SPITAL);

OPEN A;

LOOP

    FETCH A INTO AUX_DATA_PROGRAMARE,AUX_COD_SECTIE;

    EXIT WHEN A%NOTFOUND;

    IF B%ISOPEN THEN

        CLOSE B;

    END IF;

    OPEN B(AUX_DATA_PROGRAMARE,AUX_COD_SECTIE);

    LOOP

        FETCH B INTO NUME,PRENUME;

        DBMS_OUTPUT.PUT_LINE(NUME || '' || PRENUME);

        EXIT WHEN B%NOTFOUND;

    END LOOP;

    CLOSE B;

END LOOP;

CLOSE A;

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        RAISE_APPLICATION_ERROR(-20000,'Nu exista spitalul cu numele dat');

    WHEN TOO_MANY_ROWS THEN

        RAISE_APPLICATION_ERROR(-20001,'Exista mai multe spitale cu numele dat');

    WHEN OTHERS THEN

        RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');

END P7;
/

```

BEGIN

P7('Spitalul Fundeni');

END;

/

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the database schema for 'anastasia' with various objects like Tables, Procedures, Functions, Triggers, Types, and Directories. The central workspace contains the following PL/SQL code:

```
45      CLOSE B;
46  END LOOP;
47  CLOSE A;
48
49  EXCEPTION
50    WHEN NO_DATA_FOUND THEN
51      RAISE_APPLICATION_ERROR(-20000,'Nu exista spitalul cu numele dat');
52    WHEN TOO_MANY_ROWS THEN
53      RAISE_APPLICATION_ERROR(-20001,'Exista mai multe spitale cu numele dat');
54    WHEN OTHERS THEN
55      RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
56  END P7;
57 /
```

Below the code, a message indicates the procedure was successfully compiled:

PL/SQL procedure successfully completed.

In the bottom right corner of the interface, there is a status bar showing the date (1/13/2023), time (7:14 PM), and system information (Line 57 Column 2, Insert, Modified, Windows).

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the database schema with various objects like tables, packages, and procedures. The main workspace contains a PL/SQL script being run:

```
51      RAISE_APPLICATION_ERROR(-20000,'Nu exista spitalul cu numele dat');
52  WHEN TOO_MANY_ROWS THEN
53      RAISE_APPLICATION_ERROR(-20001,'Exista mai multe spitale cu numele dat');
54  WHEN OTHERS THEN
55      RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
56  END P7;
57 /
58
59 BEGIN
60     P7('Spitalul Fundeni');
61 END;
```

Below the script, the message "PL/SQL procedure successfully completed." is displayed. The bottom pane shows the output of the procedure execution, which lists names: Spitalul ales este Spitalul Fundeni, Doctorii care lucreaza in timpul programarilor si la sectia respectiva sunt : Alex Andrei, Alex Andrei, Nistor George.

```

50 WHEN NO_DATA_FOUND THEN
51     RAISE_APPLICATION_ERROR(-20000,'Nu exista spitalul cu numele dat');
52 WHEN TOO_MANY_ROWS THEN
53     RAISE_APPLICATION_ERROR(-20001,'Exista mai multe spitale cu numele dat');
54 WHEN OTHERS THEN
55     RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
56 END P7;
57 /
58
59 BEGIN
60     P7('Spitalul Sf Ioan');
61 END;
62 /
63

```

Error report -
ORA-20001: Exista mai multe spitale cu numele dat
ORA-06512: at "ANASTASIA.P7", line 53
ORA-06512: at line 2

EXERCITIU 8

--Pentru numele dat al unui pacient sa se verifice lista sa de alergii si reteta si sa se scoata medicamentele la care au alergie (daca este cazul). Daca s-au gasit medicamente la care pacientul --este alergic sa se micsoreze salariul doctorului care i-a prescris-o pana la cel mai mic salar existent --in spitalul si sectia acestuia.

CREATE OR REPLACE FUNCTION F8(NUME_PACIENT.NUME_PACIENT%TYPE)

RETURN TIP_MEDICAMENTE

IS

```

NOUA_LISTA_MEDICAMENTE TIP_MEDICAMENTE:= TIP_MEDICAMENTE();

AUX_LISTA_MEDICAMENTE TIP_MEDICAMENTE:=TIP_MEDICAMENTE();

AUX_LISTA_ALERGII TIP_ALERGII:=TIP_ALERGII();

AUX1_LISTA_ALERGII TIP_ALERGII:=TIP_ALERGII();

NR NUMBER(2);

AUX_NR NUMBER(2);

```

```
COD_SECTIE SECTIE.ID_SECTIE%TYPE;
COD_DOCTOR DOCTOR.ID_DOCTOR%TYPE;
COD_SPITAL SPITAL.ID_SPITAL%TYPE;
COD_RETETA RETETA.ID_RETETA%TYPE;

BEGIN
    AUX_LISTA_MEDICAMENTE.EXTEND;
    AUX_LISTA_MEDICAMENTE(1):='XX';

    SELECT NVL(LISTA_MEDICAMENTE,AUX_LISTA_MEDICAMENTE)
    INTO NOUA_LISTA_MEDICAMENTE
    FROM RETETA R
    INNER JOIN CONSULT C ON (C.ID_CONSULT = R.ID_CONSULT)
    INNER JOIN INTERNARE I ON(I.ID_INTERNARE = C.ID_INTERNARE)
    INNER JOIN PACIENT P ON(P.ID_PACIENT = I.ID_PACIENT)
    WHERE NUME_PACIENT=NUME;

    SELECT ID_RETETA
    INTO COD_RETETA
    FROM RETETA R
    INNER JOIN CONSULT C ON (C.ID_CONSULT = R.ID_CONSULT)
    INNER JOIN INTERNARE I ON(I.ID_INTERNARE = C.ID_INTERNARE)
    INNER JOIN PACIENT P ON(P.ID_PACIENT = I.ID_PACIENT)
    WHERE NUME_PACIENT=NUME;

    SELECT D.ID_DOCTOR
    INTO COD_DOCTOR
    FROM DOCTOR D
    INNER JOIN CONSULT C ON(C.ID_DOCTOR=D.ID_DOCTOR)
    INNER JOIN RETETA R ON(R.ID_CONSULT=C.ID_CONSULT)
```

```
WHERE ID_RETETA=COD_RETETA;
```

```
SELECT D.ID_SECTIE  
INTO COD_SECTIE  
FROM DOCTOR D  
INNER JOIN CONSULT C ON(C.ID_DOCTOR=D.ID_DOCTOR)  
INNER JOIN RETETA R ON(R.ID_CONSULT=C.ID_CONSULT)  
WHERE ID_RETETA=COD_RETETA;
```

```
AUX1_LISTA_ALERGII.EXTEND;  
AUX1_LISTA_ALERGII(1):='XX';
```

```
SELECT NVL(LISTA_ALERGII,AUX_LISTA_ALERGII), ID_SPITAL  
INTO AUX_LISTA_ALERGII, COD_SPITAL  
FROM PACIENT  
WHERE NUME_PACIENT=NUME;
```

```
NR:=0;  
NR:=NOUA_LISTA_MEDICAMENTE.COUNT;  
AUX_NR:=NR;  
  
FOR I IN 1..AUX_NR LOOP  
    FOR J IN AUX_LISTA_ALERGII.FIRST..AUX_LISTA_ALERGII.LAST LOOP  
        IF NOUA_LISTA_MEDICAMENTE(I)=AUX_LISTA_ALERGII(J) THEN  
            FOR K IN I..AUX_NR-1 LOOP  
                NOUA_LISTA_MEDICAMENTE(K):=NOUA_LISTA_MEDICAMENTE(K+1);  
            END LOOP;  
            AUX_NR:=AUX_NR-1;  
        END IF;
```

```

END LOOP;

END LOOP;

IF AUX_NR < NR THEN
    DBMS_OUTPUT.PUT_LINE('S-au gasit medicamente la care pacientul ' || NUME || ' are alergii.
Reteta corecta este : ');
    FOR I IN 1..AUX_NR LOOP
        DBMS_OUTPUT.PUT_LINE(NOUA_LISTA_MEDICAMENTE(I));
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('Ne pare rau pentru greseala! Doctorului i se va scadea salariul.');
END IF;

UPDATE DOCTOR A
SET A.SALARIU =
    (SELECT MIN(B.SALARIU)
     FROM DOCTOR B
     WHERE A.ID_SECTIE=B.ID_SECTIE AND A.ID_SPITAL=B.ID_SPITAL)
     WHERE ID_DOCTOR=COD_DOCTOR;

ELSE
    DBMS_OUTPUT.PUT_LINE('Nu s-au gasit medicamente la care ' || NUME || ' este alergic.');
END IF;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20000,'Nu exista pacienti cu numele dat');
    WHEN TOO_MANY_ROWS THEN
        RAISE_APPLICATION_ERROR(-20001,'Există mai mulți pacienți cu numele dat');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');

```

```
RETURN NOUA_LISTA_MEDICAMENTE;
```

```
END F8;
```

```
select * from doctor
```

```
DECLARE
```

```
AUX_MEDICAMENTE RETETA.LISTA_MEDICAMENTE%TYPE;
```

```
BEGIN
```

```
AUX_MEDICAMENTE:=F8('Sandu');
```

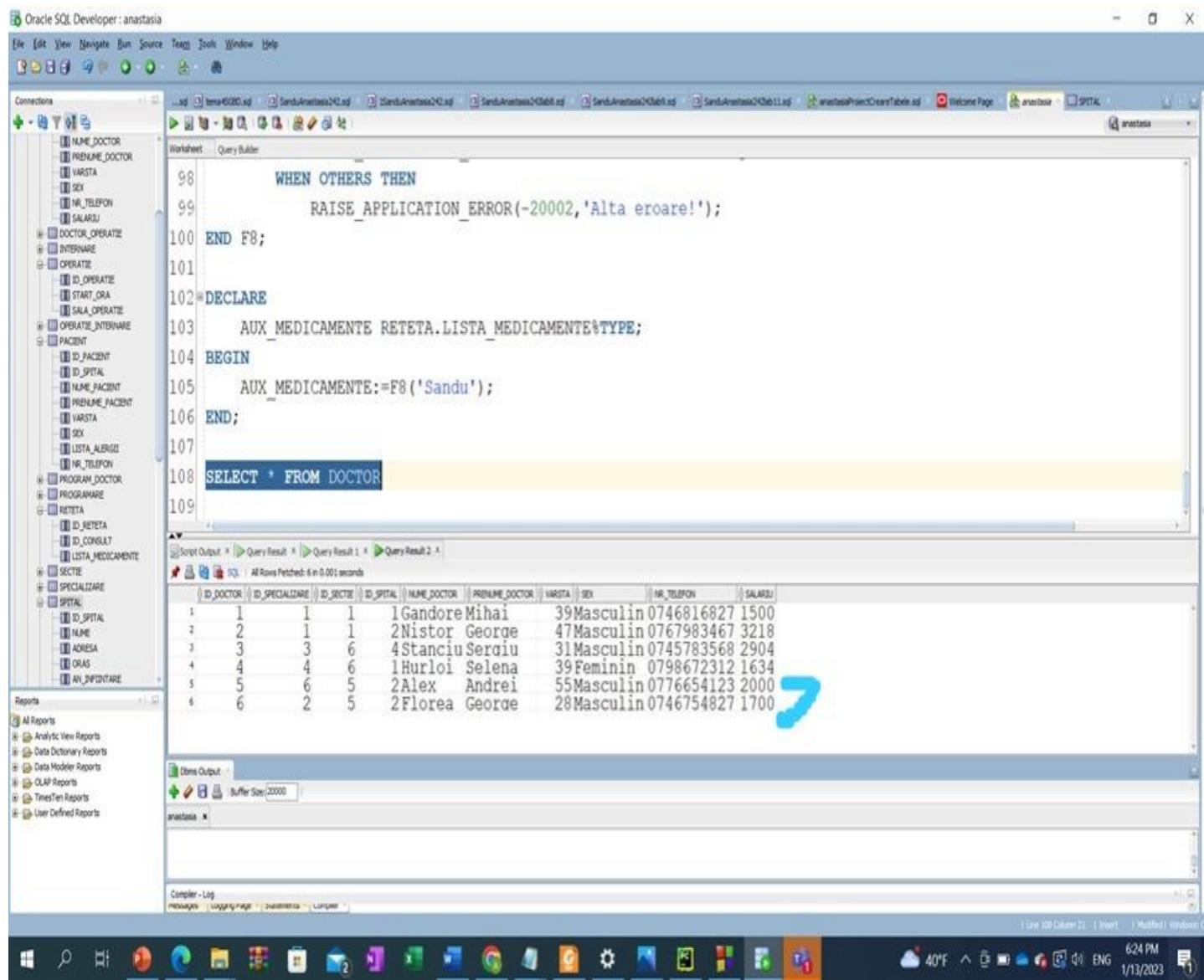
```
END;
```

The screenshot shows the Oracle SQL Developer interface. The main window displays a PL/SQL block:

```
91 RETURN NOUA_LISTA_MEDICAMENTE;
92
93 EXCEPTION
94   WHEN NO_DATA_FOUND THEN
95     RAISE_APPLICATION_ERROR(-20000,'Nu exista pacienti cu numele dat');
96   WHEN TOO_MANY_ROWS THEN
97     RAISE_APPLICATION_ERROR(-20001,'Există mai multi pacienti cu numele dat');
98   WHEN OTHERS THEN
99     RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
100 END F8;
101
```

The status bar at the bottom indicates "Function F8 compiled". Below the code editor is the "Script Output" pane, which shows the message "anastasia". The bottom right corner of the screen shows system information: "40°F", "624 PM", "ENG", and the date "1/13/2023".

--inainte



Oracle SQL Developer : anastasia

```

Connections Worksheet Query Builder
...sql item-65GDB.sql SanduAnastasia242.sql tSanduAnastasia242.sql SanduAnastasia242lab8.sql SanduAnastasia242lab9.sql SanduAnastasia242lab11.sql anastasiaProjectCreateTabele.sql Welcome Page anastasia SPITAL
+ [+] NAME_DOCTOR PRENUME_DOCTOR VARSTA NR_TELEFON SEX SALARIU
+ [+] DOCTOR_OPERATE
+ [+] INTERNARE
+ [+] OPERATE
+ [+] ID_OPERATE
+ [+] START_ORA
+ [+] SALA_OPERATE
+ [+] OPERATE_INTERNARE
+ [+] PACIENT
+ [+] ID_PACIENT
+ [+] ID_SPITAL
+ [+] NUME_PACIENT
+ [+] PRENUME_PACIENT
+ [+] VARSTA
+ [+] SEX
+ [+] LISTA_ALERGII
+ [+] NR_TELEFON
+ [+] PROGRAM_DOCTOR
+ [+] PROGRAMARE
+ [+] RETETA
+ [+] ID_RETETA
+ [+] ID_CONSULT
+ [+] LISTA_MEDICAMENTE
+ [+] SECTIE
+ [+] SPECIALIZARE
+ [+] SPITAL
+ [+] ID_SPITAL
+ [+] NUME
+ [+] ADRESA
+ [+] ORAS
+ [+] AN_INFINTARE
Reports Script Output Buffer Size:20000
anastasia
Script Output
Query Result x Query Result 1 x Query Result 2 x
Task completed in 0.302 seconds
PL/SQL procedure successfully completed.

S-au gasit medicamente la care pacientul Sandu are alergii. Reteta corecta este :
Rompirin
Santepirin
Phelodia
Aspacardin
Aspimax
Ne pare rau pentru qreseala! Doctorului i se va scadea salariul.

Compiler - Log Logging Page Diagnostics Compiler
Line 106 Column 5 Insert Modified Windows
627 PM 40°F ENG 1/13/2023

```

--dupa

Oracle SQL Developer : anastasia

```

Connections Worksheet Query Builder
...sql item-65GDB.sql SanduAnastasia242.sql tSanduAnastasia242.sql SanduAnastasia242lab8.sql SanduAnastasia242lab9.sql SanduAnastasia242lab11.sql anastasiaProjectCreateTabele.sql Welcome Page anastasia SPITAL
+ [+] NAME_DOCTOR PRENUME_DOCTOR VARSTA NR_TELEFON SEX SALARIU
+ [+] DOCTOR_OPERATE
+ [+] INTERNARE
+ [+] OPERATE
+ [+] ID_OPERATE
+ [+] START_ORA
+ [+] SALA_OPERATE
+ [+] OPERATE_INTERNARE
+ [+] PACIENT
+ [+] ID_PACIENT
+ [+] ID_SPITAL
+ [+] NUME_PACIENT
+ [+] PRENUME_PACIENT
+ [+] VARSTA
+ [+] SEX
+ [+] LISTA_ALERGII
+ [+] NR_TELEFON
+ [+] PROGRAM_DOCTOR
+ [+] PROGRAMARE
+ [+] RETETA
+ [+] ID_RETETA
+ [+] ID_CONSULT
+ [+] LISTA_MEDICAMENTE
+ [+] SECTIE
+ [+] SPECIALIZARE
+ [+] SPITAL
+ [+] ID_SPITAL
+ [+] NUME
+ [+] ADRESA
+ [+] ORAS
+ [+] AN_INFINTARE
Reports Script Output Buffer Size:20000
anastasia
Script Output
Query Result x Query Result 1 x Query Result 2 x
All Rows Fetched: 6 in 0.001 seconds
SELECT * FROM DOCTOR
1 1 1 1 Gandore Mihai 39 Masculin 0746816827 1500
2 2 1 1 Nistor George 47 Masculin 0767983467 3218
3 3 3 6 Stanciu Sergiu 31 Masculin 0745783568 2904
4 4 4 6 Hurloiu Selena 39 Feminin 0798672312 1634
5 5 6 5 Alex Andrei 55 Masculin 0776654123 1700
6 6 2 5 Florea George 28 Masculin 0746754827 1700
Doms Output
anastasia
Compiler - Log Logging Page Diagnostics Compiler
Line 108 Column 21 Insert Modified Windows
627 PM 40°F ENG 1/13/2023

```

Oracle SQL Developer : anastasia

```

101 select * from doctor
102
103
104
105 DECLARE
106   AUX_MEDICAMENTE RETETA.LISTA_MEDICAMENTE%TYPE;
107 BEGIN
108   AUX_MEDICAMENTE:=F8('Mihai');
109 END;
110
111

```

PL/SQL procedure successfully completed.

Dbsm Output:

Nu s-au gasit medicamente la care Mihai este alergic.

Oracle SQL Developer : anastasia

```

97 RAISE_APPLICATION_ERROR(-20001,'Există mai mulți pacienți cu numele dat');
98 WHEN OTHERS THEN
99   RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
100 END F8;
101
102 DECLARE
103   AUX_MEDICAMENTE RETETA.LISTA_MEDICAMENTE%TYPE;
104 BEGIN
105   AUX_MEDICAMENTE:=F8('PROIECTSGBD');
106 END;
107
108 SELECT * FROM DOCTOR
109

```

ORA-20000: Nu există pacienți cu numele dat
ORA-06512: at "ANASTASIA.F8", line 92
ORA-06512: at line 4
20000. 00000 - "%s"
*Cause: The stored procedure 'raise_application_error' was called which causes this error to be generated.

UPDATE PACIENT

SET NUME_PACIENT='Sandu'

WHERE ID_PACIENT=3;

--pentru a exemplifica eroare 'TOO MANY ROWS'

```

95      RAISE_APPLICATION_ERROR(-20000,'Nu exista pacient cu numele dat');
96      WHEN TOO_MANY_ROWS THEN
97          RAISE_APPLICATION_ERROR(-20001,'Există mai multi pacienti cu numele dat');
98      WHEN OTHERS THEN
99          RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
100  END F8;
101
102  DECLARE
103      AUX_MEDICAMENTE RETETA.LISTA_MEDICAMENTE%TYPE;
104  BEGIN
105      AUX_MEDICAMENTE:=F8('Sandu');
106  END;
107

```

Error report -

```

ORA-20001: Există mai multi pacienti cu numele dat
ORA-06512: at "ANASTASIA.F8", line 94
ORA-06512: at line 4

```

EXERCITIU 9

--Pentru un spital al carui nume este dat sa se verifice ce doctori sunt in operatie dupa programul lor si --sa li se mareasca salariul cu 10%

CREATE OR REPLACE TYPE TIP_DOCTORI AS VARRAY(100) OF VARCHAR2(30);

/

CREATE OR REPLACE PROCEDURE P9(DENUMIRE SPITAL.NUME%TYPE)

IS

```

DOCTORI TIP_DOCTORI:=TIP_DOCTORI();
COD_SPITAL SPITAL.ID_SPITAL%TYPE;
NUME_D DOCTOR.NUME_DOCTOR%TYPE;
PRENUME_D DOCTOR.PRENUME_DOCTOR%TYPE;

BEGIN
    SELECT ID_SPITAL
    INTO COD_SPITAL
    FROM SPITAL

```

```
WHERE NUME=DENUMIRE;
```

```
SELECT DISTINCT D.ID_DOCTOR  
BULK COLLECT INTO DOCTORI  
FROM DOCTOR D  
INNER JOIN DOCTOR_OPERATIE DO ON(D.ID_DOCTOR=DO.ID_DOCTOR)  
INNER JOIN OPERATIE O ON(O.ID_OPERATIE=DO.ID_OPERATIE)  
INNER JOIN OPERATIE_INTERNARE OI ON(OI.ID_OPERATIE=O.ID_OPERATIE)  
INNER JOIN INTERNARE I ON(I.ID_INTERNARE=OI.ID_INTERNARE)  
INNER JOIN SPITAL S ON(S.ID_SPITAL=I.ID_SPITAL)  
INNER JOIN PROGRAM_DOCTOR PD ON(PD.ID_DOCTOR=D.ID_DOCTOR)  
WHERE D.ID_SPITAL=COD_SPITAL AND PD.END_ORA<O.START_ORA;
```

```
UPDATE DOCTOR
```

```
SET SALARIU = SALARIU + SALARIU * 0.1  
WHERE ID_DOCTOR IN  
(SELECT DISTINCT D.ID_DOCTOR  
FROM DOCTOR D  
INNER JOIN DOCTOR_OPERATIE DO ON(D.ID_DOCTOR=DO.ID_DOCTOR)  
INNER JOIN OPERATIE O ON(O.ID_OPERATIE=DO.ID_OPERATIE)  
INNER JOIN OPERATIE_INTERNARE OI ON(OI.ID_OPERATIE=O.ID_OPERATIE)  
INNER JOIN INTERNARE I ON(I.ID_INTERNARE=OI.ID_INTERNARE)  
INNER JOIN SPITAL S ON(S.ID_SPITAL=I.ID_SPITAL)  
INNER JOIN PROGRAM_DOCTOR PD ON(PD.ID_DOCTOR=D.ID_DOCTOR)  
WHERE D.ID_SPITAL=COD_SPITAL AND PD.END_ORA<O.START_ORA);
```

```
IF DOCTORI.COUNT=0 THEN
```

```

DBMS_OUTPUT.PUT_LINE('Nu exista doctori care sa lucreze peste program la spitalul ' || 
DENUMIRE);

ELSIF DOCTORI.COUNT=1 THEN

DBMS_OUTPUT.PUT_LINE('Doctorul este caruia i s-a marit salariul este: ');

ELSE

DBMS_OUTPUT.PUT_LINE('Doctorii carora li s-a marit salariul sunt: ');

END IF;

FOR I IN DOCTORI.FIRST..DOCTORI.LAST LOOP

SELECT NUME_DOCTOR, PRENUME_DOCTOR

INTO NUME_D, PRENUME_D

FROM DOCTOR

WHERE ID_DOCTOR=DOCTORI(I);

DBMS_OUTPUT.PUT_LINE(NUME_D || ' ' || PRENUME_D);

END LOOP;

EXCEPTION

WHEN NO_DATA_FOUND THEN

RAISE_APPLICATION_ERROR(-20000,'Nu exista spitale cu numele dat');

WHEN TOO_MANY_ROWS THEN

RAISE_APPLICATION_ERROR(-20001,'Exista mai multe spitale cu numele dat');

WHEN OTHERS THEN

RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');

END P9;

BEGIN

P9('Spitalul Judetean');

END;

```

The screenshot shows the Oracle SQL Developer interface. The central area displays a PL/SQL procedure named P9. The code includes exception handling for NO_DATA_FOUND, TOO_MANY_ROWS, and OTHERS, and a call to P9 with the argument 'Spitalul Judetean'. The procedure is successfully compiled, as indicated by the message 'Procedure P9 compiled' in the script output window. The bottom status bar shows the date and time as 1/13/2023 6:34 PM.

```
57      END LOOP;
58
59  EXCEPTION
60    WHEN NO_DATA_FOUND THEN
61      RAISE_APPLICATION_ERROR(-20000,'Nu exista spitale cu numele dat');
62    WHEN TOO_MANY_ROWS THEN
63      RAISE_APPLICATION_ERROR(-20001,'Există mai multe spitale cu numele dat');
64    WHEN OTHERS THEN
65      RAISE APPLICATION_ERROR(-20002,'Alta eroare!');
66  END P9;
67
68 BEGIN
69   P9('Spitalul Judetean');

```

Type TIP_DOCTORI compiled

--inainte

Oracle SQL Developer : anastasia

File Edit View Navigate Run Source Team Tools Window Help

Connections

Worksheet Query Builder

```

61      RAISE_APPLICATION_ERROR(-20000,'NU exista spitale cu numele dat');
62      WHEN TOO_MANY_ROWS THEN
63          RAISE_APPLICATION_ERROR(-20001,'Există mai multe spitale cu numele dat');
64      WHEN OTHERS THEN
65          RAISE_APPLICATION_ERROR(-20002,'Alte eroare!');
66  END P9;
67
68 BEGIN
69     P9('Spitalul Fundeni');
70 END;
71
72 select * from doctor
73
74

```

Script Output x | Query Result x | Query Result 1 x | Query Result 2 x

All Rows Fetched: 6 in 0.002 seconds

ID_DOCTOR	ID_SPECIALIZARE	ID_SECTIE	ID_SPITAL	NUME_DOCTOR	PRENUME_DOCTOR	VARSTA	SEX	NR_TELEFON	SALARU
1	1	1	1	Gandore Mihai	39	Masculin	0746816827	1500	
2	2	1	1	Nistor George	47	Masculin	0767983467	3894	
3	3	3	6	Stanciu Sergiu	31	Masculin	0745783568	2904	
4	4	4	6	Hurloiu Selena	39	Feminin	0798672312	1634	
5	5	6	5	Alex Andrei	55	Masculin	0776654123	1700	
6	6	2	5	Florea George	28	Masculin	0746754827	1700	

Doms Output x

Compiler - Log

Messages Logging Page Statistics Compiler

Line 72 Column 21 | Insert | Modified | Windows G 6:40 PM 1/13/2023

Oracle SQL Developer : anastasia

Connections Worksheet Query Builder

```

61      RAISE_APPLICATION_ERROR(-20000,'Nu exista spitale cu numele dat');
62      WHEN TOO_MANY_ROWS THEN
63          RAISE_APPLICATION_ERROR(-20001,'Există mai multe spitale cu numele dat');
64      WHEN OTHERS THEN
65          RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
66  END P9;
67
68 BEGIN
69     P9('Spitalul Fundeni');
70 END;
71
72 select * from doctor

```

Script Output | Query Result | Query Result 1 | Query Result 2 | Task completed in 0.195 seconds

PL/SQL procedure successfully completed.

Dbsm Output | Buffer Size(20000) | anastasia

Doctorul este caruia i s-a marit salariul este:
Nistor George

Compiler - Log | Messages | Logging page | Databases | Compiled | Line 72 Column 5 | Insert | Modified | Windows 0

Oracle SQL Developer : anastasia

File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Query Builder

```

64      WHEN OTHERS THEN
65          RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
66  END P9;
67
68 BEGIN
69     P9('Spitalul Fundeni');
70 END;
71
72 select * from doctor
73
74

```

Script Output | Query Result | Query Result 1 | Query Result 2 | All Rows Fetched: 6 in 0.002 seconds

ID_DOCTOR	ID_SPECIALIZARE	ID_SECTIE	ID_SPITAL	ID_NUME_DOCTOR	ID_PRENUME_DOCTOR	ID_VARSTA	ID_SEX	ID_TELEFON	ID_SALARU
1	1	1	1	1	Gandore Mihai	39	Masculin	0746816827	1500
2	2	1	1	2	Nistor George	47	Masculin	0767983467	4283
3	3	3	6	4	Stanciu Sergiu	31	Masculin	0745783568	2904
4	4	4	6	5	Hurloiu Selena	39	Feminin	0798672312	1634
5	5	6	5	2	Alex Andrei	55	Masculin	0776654123	1700
6	6	2	5	2	Florea George	28	Masculin	0746754827	1700

Dbsm Output | Buffer Size(20000) | anastasia

Doctorul este caruia i s-a marit salariul este:
Nistor George

Compiler - Log | Messages | Logging page | Databases | Compiled | Line 72 Column 21 | Insert | Modified | Windows 0

Oracle SQL Developer : anastasia

Connections Worksheet Query Builder

```

58      WHEN NO_DATA_FOUND THEN
59          RAISE_APPLICATION_ERROR(-20000,'Nu exista spitale cu numele dat');
60      WHEN TOO_MANY_ROWS THEN
61          RAISE_APPLICATION_ERROR(-20001,'Există mai multe spitale cu numele dat');
62      WHEN OTHERS THEN
63          RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
64  END P9;
65
66 BEGIN
67     P9('Spitalul Mocrea');
68 END;
69

```

Script Output | Query Result | Query Result 1 | Explain Plan | Query Result 2 | Task completed in 0.109 seconds

PL/SQL procedure successfully completed.

Dms Output: Buffer Size:20000 anastasia

Nu există doctori care să lucreze peste program la spitalul Spitalul Mocrea

Compiler - Log | Logging Inactive | dmsmaster | Complete | Line 68 Column 5 | Insert | Modified | Windows: D

Oracle SQL Developer : anastasia

Connections Worksheet Query Builder

```

59      EXCEPTION
60      WHEN NO_DATA_FOUND THEN
61          RAISE_APPLICATION_ERROR(-20000,'Nu există spitale cu numele dat');
62      WHEN TOO_MANY_ROWS THEN
63          RAISE_APPLICATION_ERROR(-20001,'Există mai multe spitale cu numele dat');
64      WHEN OTHERS THEN
65          RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
66  END P9;
67
68 BEGIN
69     P9('Spitalul Judetean');
70 END;
71

```

Script Output | Query Result | Query Result 1 | Explain Plan | Query Result 2 | Task completed in 0.103 seconds

Error report -
ORA-20002: Alta eroare!
ORA-06512: at "ANASTASIA.P9", line 60
ORA-06512: at line 2

Dms Output: Buffer Size:20000 anastasia

Compiler - Log | Logging Inactive | dmsmaster | Complete | Line 70 Column 5 | Insert | Modified | Windows: D

Oracle SQL Developer : anastasia

```

EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20000,'Nu exista spitale cu numele dat');
  WHEN TOO_MANY_ROWS THEN
    RAISE_APPLICATION_ERROR(-20001,'Există mai multe spitale cu numele dat');
  WHEN OTHERS THEN
    RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
END P9;
/
BEGIN
  P9('Spitalul PROIECTSGBD');
END;
/

```

ORA-20000: Nu exista spitale cu numele dat
ORA-06512: at "ANASTASIA.P9", line 56
ORA-06512: at line 2
20000. 00000 - "%s"
*Cause: The stored procedure 'raise_application_error'
was called which causes this error to be generated.

Dbsm Output: Buffer Size[20000]

anastasia

Compiler - Log | Logging Inherit | anastasia | Complete

Oracle SQL Developer : anastasia

```

EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20000,'Nu exista spitale cu numele dat');
  WHEN TOO_MANY_ROWS THEN
    RAISE_APPLICATION_ERROR(-20001,'Există mai multe spitale cu numele dat');
  WHEN OTHERS THEN
    RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
END P9;
/
BEGIN
  P9('Spitalul Sf Ioan');
END;
/

```

Error report -
ORA-20001: Există mai multe spitale cu numele dat
ORA-06512: at "ANASTASIA.P9", line 58
ORA-06512: at line 2

Dbsm Output: Buffer Size[20000]

anastasia

Compiler - Log | Logging Inherit | anastasia | Complete

EXERCITIU 10

--Definiti un declasator care sa nu permite sa se poata face programari noi de sarbatori

CREATE OR REPLACE TRIGGER trig10

BEFORE INSERT OR UPDATE OR DELETE ON PROGRAMARE

BEGIN

```
IF TO_CHAR(sysdate, 'DD-MM') = '01-01' OR TO_CHAR(sysdate, 'DD-MM') = '01-12' OR  
TO_CHAR(sysdate, 'DD-MM') = '25-12'
```

THEN

```
RAISE_APPLICATION_ERROR(-20001,'Nu se pot face modificari de sarbatori!');
```

END IF;

END;

/

INSERT INTO PROGRAMARE

```
VALUES (9,5,2,TO_DATE('29/09/2023:08:00:00PM', 'DD-MM-YYYY HH:MI:SSPM'));
```

The screenshot shows the Oracle SQL Developer interface. In the central workspace, a query builder window displays the following PL/SQL code:

```
1 --sa nus e poate face programari noi de sarbatori
2 CREATE OR REPLACE TRIGGER trig10
3   BEFORE INSERT OR UPDATE OR DELETE ON PROGRAMARE
4 BEGIN
5   IF TO_CHAR(sysdate, 'DD-MM') = '01-01' OR TO_CHAR(sysdate, 'DD-MM') = '01-12' OR TO_CHAR(sysdate, 'DD-MM') =
6   THEN
7     RAISE_APPLICATION_ERROR(-20001,'Nu se pot face modificari de sarbatori!');
8   END IF;
9 END;
10 /
11 :
12 INSERT INTO PROGRAMARE
13 VALUES (9,5,2,TO_DATE('29/09/2023:08:00:00PM', 'DD-MM-YYYY HH:MI:SSPM'));
```

Below the code, the status bar indicates "PL/SQL procedure successfully completed." and "Trigger TRIG10 compiled". The bottom status bar shows system information like temperature (40°F), battery level (64%), and date/time (1/13/2023).

--am setat data la laptop sa fie Craciunul

The screenshot shows the Oracle SQL Developer interface. In the central workspace, a query builder window displays the following PL/SQL code:

```

1 CREATE OR REPLACE TRIGGER trig10
2   BEFORE INSERT OR UPDATE OR DELETE ON PROGRAMARE
3 BEGIN
4   IF TO_CHAR(sysdate, 'DD-MM') = '01-01' OR TO_CHAR(sysdate, 'DD-MM') = '01-12' OR TO_CHAR(sysdate, 'DD-MM') =
5     THEN
6     RAISE_APPLICATION_ERROR(-20001, 'Nu se pot face modificari de sarbatori!');
7   END IF;
8 END;
9 /
10
11 INSERT INTO PROGRAMARE
12 VALUES (9, 5, 2, TO_DATE('29/09/2023:08:00:00PM', 'DD-MM-YYYY HH:MI:SSPM'));
13

```

Below the code, the "Script Output" tab shows the error report:

Error report -
ORA-20001: Nu se pot face modificari de sarbatori!
ORA-06512: at "ANASTASIA.TRIG10", line 4
ORA-04088: error during execution of trigger 'ANASTASIA.TRIG10'

EXERCITIU 11

--Definiti un declansator care sa nu permita modificarea datei unei programari cu o noua data pentru care in acea sectie nu lucreaza un doctor

CREATE OR REPLACE TRIGGER trig11

BEFORE UPDATE OF DATA_PROGRAMARE ON PROGRAMARE

FOR EACH ROW

DECLARE

PRAGMA AUTONOMOUS_TRANSACTION;

NR NUMBER :=0;

BEGIN

SELECT COUNT(*)

INTO NR

FROM PROGRAMARE P

```

INNER JOIN SECTIE S ON(P.ID_SECTIE=S.ID_SECTIE)

INNER JOIN DOCTOR D ON(D.ID_SECTIE=S.ID_SECTIE)

INNER JOIN PROGRAM_DOCTOR PROG ON (PROG.ID_DOCTOR=D.ID_DOCTOR)

WHERE p.id_programare=:new.id_programare AND EXTRACT(HOUR FROM
:NEW.DATA_PROGRAMARE) BETWEEN PROG.START_ORA AND PROG.END_ORA AND
TO_CHAR(:NEW.DATA_PROGRAMARE, 'DAY')=PROG.ZI;

```

```

IF NR=0 THEN

    RAISE_APPLICATION_ERROR(-20002,'Nu se poate modifica data programarii deoarece nu lucreaza
niciun doctor la aceasta data!');

END IF;

END;

/

```

```

UPDATE PROGRAMARE

SET DATA_PROGRAMARE = TO_DATE('20/01/2023:12:00:00PM', 'DD-MM-YYYY HH:MI:SSPM')

WHERE ID_PROGRAMARE=1;

```

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

- Connections:** A tree view showing various database objects like NUME_DOCTOR, PRENUME_DOCTOR, VARSTA, NR_TELEFON, SALARIU, DOCTOR_OPERATIE, OPERATIE, ID_OPERATIE, START_ORA, SALA_OPERATIE, LISTA_MEDICAMENTE, PACIENT, ID_PACIENT, ID_SPITAL, NAME_PACIENT, PREGNANT_PACIENT, VARSTA, SEX, LISTA_ALERGI, TELEFON, PROGRAM_DOCTOR, PROGRAMARE, RETETA, ID_RETETA, COMENTARIU, LISTA_MEDICAMENTE, SECTIE, SPECIALIZARE, SPITAL, ID_SPITAL, NAME, ADRESA, ORAS, AN_INFIRMIARE.
- Worksheet:** The main area where the PL/SQL code is written. The code includes an IF block to check if NR=0, raising an application error if true. It also contains an UPDATE statement for the PROGRAMARE table.
- Script Output:** A panel at the bottom showing the result of the compilation: "Trigger TRIG11 compiled".
- Bottom Taskbar:** Shows the Windows taskbar with various icons and system status.

```

14     INNER JOIN DOCTOR D ON(D.ID_SECTIE=S.ID_SECTIE)
15     INNER JOIN PROGRAM_DOCTOR PROG ON (PROG.ID_DOCTOR=D.ID_DOCTOR)
16     WHERE p.id_programare=:new.id_programare AND EXTRACT(HOUR FROM :NEW.DATA_PROGRAMARE) BETWEEN PROG.START_ORA
17
18     IF NR=0 THEN
19         RAISE_APPLICATION_ERROR(-20002,'Nu se poate modifica data programarii deoarece nu lucreaza niciun doctor')
20     END IF;
21
22 /
23
24 UPDATE PROGRAMARE
25 SET DATA_PROGRAMARE = TO_DATE('20/01/2023:12:00:00PM', 'DD-MM-YYYY HH:MI:SSPM')
26 WHERE ID_PROGRAMARE=1;
27
28

```

ORA-20002: Nu se poate modifica data programarii deoarece nu lucreaza niciun doctor la aceasta data!
ORA-06512: at "ANASTASIA.TRIG11", line 14
ORA-04088: error during execution of trigger 'ANASTASIA.TRIG11'

EXERCITIU 12

--Definiti un declansator care se declanseaza la fiecare operatie LDD. Salvati operatiile facute si date despre acestea

CREATE TABLE ANASTASIA_MODIFICARI

(UTILIZATOR VARCHAR2(30),

NUME_BD VARCHAR2(50),

EVENIMENT VARCHAR2(20),

NUME_OBIECT VARCHAR2(30),

DATA_EVENIMENTULUI DATE);

SELECT * FROM ANASTASIA_MODIFICARI;

CREATE OR REPLACE TRIGGER TRIG12

AFTER CREATE OR ALTER OR DROP ON SCHEMA

BEGIN

INSERT INTO ANASTASIA_MODIFICARI

```
VALUES(SYS.LOGIN_USER, SYS.DATABASE_NAME, SYS.SYSEVENT, SYS.DICTIONARY_OBJ_NAME,
SYSDATE);

END;

/



CREATE TABLE ASISTENTA(
    ID_ASISTENTA INT NOT NULL,
    NUME_ASISTENTA VARCHAR2(20),
    PRENUME_ASISTENTA VARCHAR2(20),
    CONSTRAINT PK_ASISTENTA PRIMARY KEY (ID_ASISTENTA)
);

ALTER TABLE ASISTENTA
DROP COLUMN PRENUME_ASISTENTA;
ALTER TABLE ASISTENTA
ADD VARSTA NUMBER(2);
DROP TABLE ASISTENTA;
SELECT * FROM ANASTASIA_MODIFICARI;
```

Oracle SQL Developer : anastasia

File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Query Builder

```

4     EVENIMENT VARCHAR2 (20),
5     NUME_OBIECT VARCHAR2 (30),
6     DATA_EVENIMENTULUI DATE);
7
8 SELECT * FROM ANASTASIA_MODIFICARI;
9
10 CREATE OR REPLACE TRIGGER TRIG12
11     AFTER CREATE OR ALTER OR DROP ON SCHEMA
12 BEGIN
13     INSERT INTO ANASTASIA_MODIFICARI
14         VALUES (SYS.LOGIN_USER, SYS.DATABASE_NAME, SYS.SYSEVENT, SYS.DICTIONARY_OBJ_NAME, SYSDATE);
15 END;

```

Script Output X | Query Result X | Query Result 1 X | Query Result 2 X

Table ANASTASIA_MODIFICARI created.

Trigger TRIG12 compiled

Doms Output Buffer Size:20000

Compiler - Log Logging Page Statement Complex

Line 16 Column 2 Insert Modified Windows: O 657 PM 1/13/2023

Oracle SQL Developer : anastasia

File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Query Builder

```

1 CREATE TABLE ANASTASIA_MODIFICARI
2     (UTILIZATOR VARCHAR2(30),
3      NUME_BD VARCHAR2(50),
4      EVENIMENT VARCHAR2(20),
5      NUME_OBIECT VARCHAR2(30),
6      DATA_EVENIMENTULUI DATE);
7
8 SELECT * FROM ANASTASIA_MODIFICARI;
9
10 CREATE OR REPLACE TRIGGER TRIG12
11     AFTER CREATE OR ALTER OR DROP ON SCHEMA

```

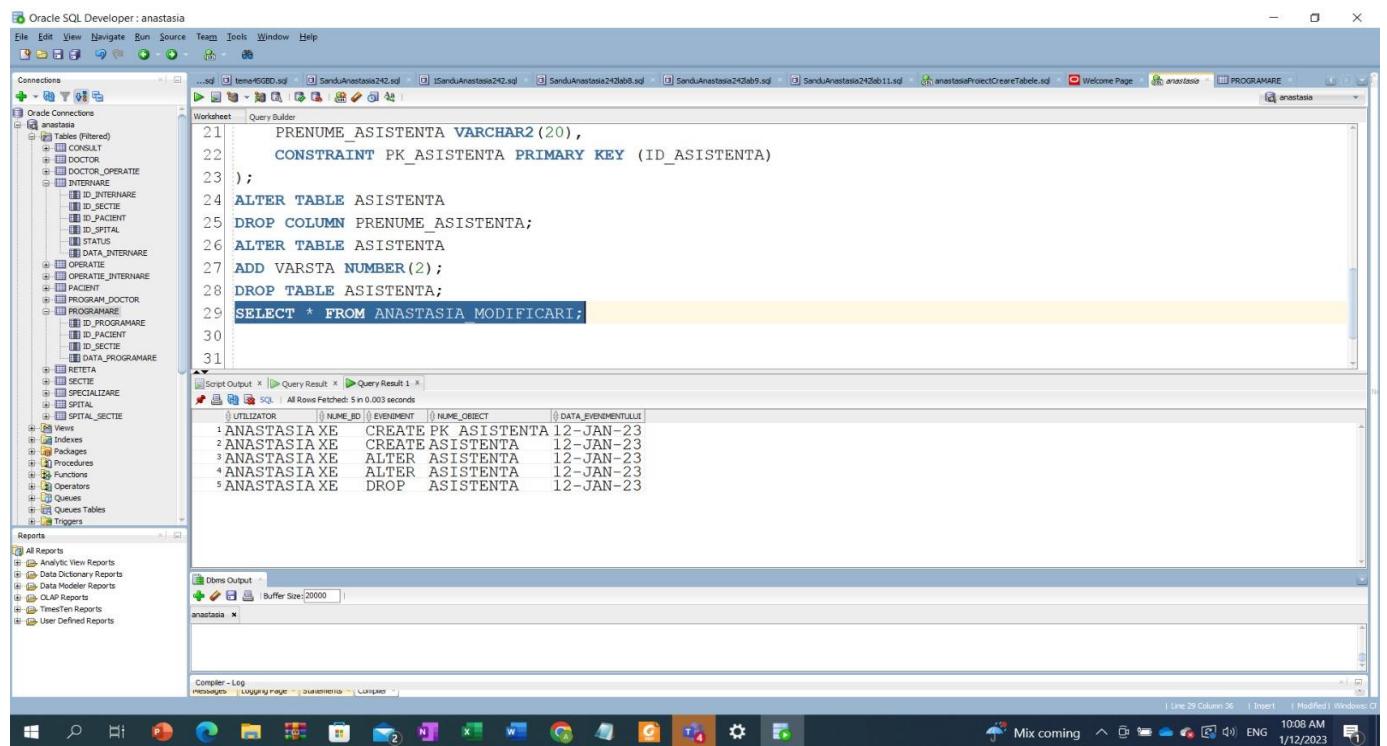
Script Output X | Query Result X | Query Result 1 X

SQL All Rows Fetched: 0 in 0.001 seconds

Doms Output Buffer Size:20000

Compiler - Log Logging Page Statement Complex

Line 8 Column 36 Insert Modified Windows: O Mix coming 10:04 AM 1/12/2023



EXERCITIU 13

--Sa se creeze un pachet care sa contine toate obiectele definite in proiect

CREATE OR REPLACE PACKAGE PACHET13 AS

```

PROCEDURE P6(DENUMIRE SPITAL.NUME%TYPE);
PROCEDURE P7(DENUMIRE SPITAL.NUME%TYPE);
FUNCTION F8(NUME PACIENT.NUME_PACIENT%TYPE)
RETURN TIP_MEDICAMENTE;
PROCEDURE P9(DENUMIRE SPITAL.NUME%TYPE);

```

END PACHET13;

/

CREATE OR REPLACE PACKAGE BODY PACHET13 IS

```

PROCEDURE P6(DENUMIRE SPITAL.NUME%TYPE)
IS
  TYPE ORE_VECTOR IS VARRAY(100) OF NUMBER(2);
  TYPE ZI_VECTOR IS VARRAY(100) OF VARCHAR2(10);
  TYPE DOCTOR_RECORD IS RECORD

```

```
(NUME DOCTOR.NUME_DOCTOR%TYPE,  
PRENUME DOCTOR.PRENUME_DOCTOR%TYPE,  
START_ORA ORE_VECTOR,  
END_ORA ORE_VECTOR,  
ZI ZI_VECTOR);  
  
TYPE DOCTOR_VECTOR IS VARRAY(100) OF DOCTOR_RECORD;  
T DOCTOR_VECTOR:= DOCTOR_VECTOR();  
  
NUME_SPITAL SPITAL.NUME%TYPE;  
NR NUMBER(2);
```

CURSOR C IS

```
SELECT NUME_DOCTOR, PRENUME_DOCTOR  
FROM SPITAL S  
JOIN DOCTOR D ON (D.ID_SPITAL=S.ID_SPITAL)  
JOIN PROGRAM_DOCTOR P ON (D.ID_DOCTOR=P.ID_DOCTOR)  
WHERE NUME=DENUMIRE;
```

BEGIN

```
SELECT NUME  
INTO NUME_SPITAL  
FROM SPITAL  
WHERE NUME=DENUMIRE;  
DBMS_OUTPUT.PUT_LINE('Spitalul ales este : ' || NUME_SPITAL);
```

```
SELECT COUNT(NUME_DOCTOR)  
INTO NR  
FROM DOCTOR D  
JOIN SPITAL S ON (D.ID_SPITAL=S.ID_SPITAL)  
WHERE NUME=DENUMIRE;
```

```
IF NR=0 THEN
    DBMS_OUTPUT.PUT_LINE('Nu lucreaza doctori la acest spital.');
ELSIF NR=1 THEN
    DBMS_OUTPUT.PUT_LINE('Numarul de doctori care lucreaza in acest spital este : ' || NR);
    DBMS_OUTPUT.PUT_LINE('Doctorul si programul lui sunt : ');
ELSE
    DBMS_OUTPUT.PUT_LINE('Numarul de doctori care lucreaza in acest spital este : ' || NR);
    DBMS_OUTPUT.PUT_LINE('Doctorii si programul lor sunt : ');
END IF;
```

```
FOR C_DOCTOR IN C LOOP
    T.EXTEND;
    T(T.LAST).NUME:=C_DOCTOR.NUME_DOCTOR;
    T(T.LAST).PRENUME:=C_DOCTOR.PRENUME_DOCTOR;
```

```
SELECT START_ORA, END_ORA, ZI
BULK COLLECT INTO
T(T.LAST).START_ORA, T(T.LAST).END_ORA, T(T.LAST).ZI
FROM SPITAL S
JOIN DOCTOR D ON (D.ID_SPITAL=S.ID_SPITAL)
JOIN PROGRAM_DOCTOR P ON (D.ID_DOCTOR=P.ID_DOCTOR)
WHERE D.NUME_DOCTOR=C_DOCTOR.NUME_DOCTOR AND
D.PRENUME_DOCTOR=C_DOCTOR.PRENUME_DOCTOR;
END LOOP;
```

```
FOR I IN T.FIRST..T.LAST LOOP
    DBMS_OUTPUT.PUT_LINE(T(I).NUME || ' ' || T(I).PRENUME);
    FOR J IN T(I).START_ORA.FIRST..T(I).START_ORA.LAST LOOP
```

```

        DBMS_OUTPUT.PUT_LINE('In ziua de ' || T(I).ZI(J) || ' lucreaza de la ' || T(I).START_ORA(J) || ' la '
|| T(I).END_ORA(J));

        END LOOP;

    END LOOP;

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        RAISE_APPLICATION_ERROR(-20000,'Nu exista spitale cu numele dat');

    WHEN TOO_MANY_ROWS THEN

        RAISE_APPLICATION_ERROR(-20001,'Există mai multe spitale cu numele dat');

    WHEN OTHERS THEN

        RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');

END P6;

PROCEDURE P7(DENUMIRE SPITAL.NUME%TYPE)

IS

    AUX_DATA_PROGRAMARE PROGRAMARE.DATA_PROGRAMARE%TYPE;

    AUX_COD_SECTIE PROGRAMARE.ID_SECTIE%TYPE;

    NUMELE DOCTOR.NUME_DOCTOR%TYPE;

    PRENUMELE DOCTOR.PRENUME_DOCTOR%TYPE;

    NUME_SPITAL SPITAL.NUME%TYPE;

CURSOR A IS

    SELECT DATA_PROGRAMARE, PROG.ID_SECTIE

    FROM SPITAL S

    INNER JOIN PACIENT P ON(S.ID_SPITAL=P.ID_SPITAL)

    INNER JOIN PROGRAMARE PROG ON(P.ID_PACIENT=PROG.ID_PACIENT)

    WHERE S.NUME=DENUMIRE;

```

```

CURSOR B(DATA_PROGRAMARE PROGRAMARE.DATA_PROGRAMARE%TYPE,COD_SECTIE
PROGRAMARE.ID_SECTIE%TYPE) IS
    SELECT NUME_DOCTOR, PRENUME_DOCTOR
    FROM DOCTOR D
    INNER JOIN PROGRAM_DOCTOR P ON(D.ID_DOCTOR=P.ID_DOCTOR)
    INNER JOIN SPITAL S ON(D.ID_SPITAL=S.ID_SPITAL)
    WHERE S.NUME = DENUMIRE AND D.ID_SECTIE=COD_SECTIE AND EXTRACT(HOUR FROM
DATA_PROGRAMARE) BETWEEN P.START_ORA AND P.END_ORA AND TO_CHAR(DATA_PROGRAMARE,
'DAY')=P.ZI;
BEGIN
    SELECT NUME
    INTO NUME_SPITAL
    FROM SPITAL
    WHERE NUME=DENUMIRE;

    DBMS_OUTPUT.PUT_LINE('Spitalul ales este ' || '' || NUME_SPITAL);

    OPEN A;
    LOOP
        FETCH A INTO AUX_DATA_PROGRAMARE,AUX_COD_SECTIE;
        EXIT WHEN A%NOTFOUND;
        IF B%ISOPEN THEN
            CLOSE B;
        END IF;
        OPEN B(AUX_DATA_PROGRAMARE,AUX_COD_SECTIE);
        LOOP
            FETCH B INTO NUMELE,PRENUMELE;
            DBMS_OUTPUT.PUT_LINE(NUMELE || '' || PRENUMELE);
            EXIT WHEN B%NOTFOUND;
        END LOOP;
    END LOOP;

```

```

CLOSE B;

END LOOP;

CLOSE A;

EXCEPTION

  WHEN NO_DATA_FOUND THEN

    RAISE_APPLICATION_ERROR(-20000,'Nu exista spitalul cu numele dat');

  WHEN TOO_MANY_ROWS THEN

    RAISE_APPLICATION_ERROR(-20001,'Exista mai multe spitale cu numele dat');

  WHEN OTHERS THEN

    RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');

END P7;

FUNCTION F8(NUME_PACIENT.NUME_PACIENT%TYPE)
RETURN TIP_MEDICAMENTE
IS
  NOUA_LISTA_MEDICAMENTE TIP_MEDICAMENTE:= TIP_MEDICAMENTE();
  AUX_LISTA_MEDICAMENTE TIP_MEDICAMENTE:=TIP_MEDICAMENTE();
  AUX_LISTA_ALERGII TIP_ALERGII:=TIP_ALERGII();
  AUX1_LISTA_ALERGII TIP_ALERGII:=TIP_ALERGII();
  NR NUMBER(2);
  AUX_NR NUMBER(2);
  COD_SECTIE SECTIE.ID_SECTIE%TYPE;
  COD_DOCTOR DOCTOR.ID_DOCTOR%TYPE;
  COD_SPITAL SPITAL.ID_SPITAL%TYPE;
  COD_RETETA RETETA.ID_RETETA%TYPE;
BEGIN
  AUX_LISTA_MEDICAMENTE.EXTEND;
  AUX_LISTA_MEDICAMENTE(1):='XX';

```

```
SELECT NVL(LISTA_MEDICAMENTE,AUX_LISTA_MEDICAMENTE)
INTO NOUA_LISTA_MEDICAMENTE
FROM RETETA R
INNER JOIN CONSULT C ON (C.ID_CONSULT = R.ID_CONSULT)
INNER JOIN INTERNARE I ON(I.ID_INTERNARE = C.ID_INTERNARE)
INNER JOIN PACIENT P ON(P.ID_PACIENT = I.ID_PACIENT)
WHERE NUME_PACIENT=NUME;
```

```
SELECT ID_RETETA
INTO COD_RETETA
FROM RETETA R
INNER JOIN CONSULT C ON (C.ID_CONSULT = R.ID_CONSULT)
INNER JOIN INTERNARE I ON(I.ID_INTERNARE = C.ID_INTERNARE)
INNER JOIN PACIENT P ON(P.ID_PACIENT = I.ID_PACIENT)
WHERE NUME_PACIENT=NUME;
```

```
SELECT D.ID_DOCTOR
INTO COD_DOCTOR
FROM DOCTOR D
INNER JOIN CONSULT C ON(C.ID_DOCTOR=D.ID_DOCTOR)
INNER JOIN RETETA R ON(R.ID_CONSULT=C.ID_CONSULT)
WHERE ID_RETETA=COD_RETETA;
```

```
SELECT D.ID_SECTIE
INTO COD_SECTIE
FROM DOCTOR D
INNER JOIN CONSULT C ON(C.ID_DOCTOR=D.ID_DOCTOR)
INNER JOIN RETETA R ON(R.ID_CONSULT=C.ID_CONSULT)
WHERE ID_RETETA=COD_RETETA;
```

```

AUX1_LISTA_ALERGII.EXTEND;
AUX1_LISTA_ALERGII(1):='XX';

SELECT NVL(LISTA_ALERGII,AUX_LISTA_ALERGII), ID_SPITAL
INTO AUX_LISTA_ALERGII, COD_SPITAL
FROM PACIENT
WHERE NUME_PACIENT=NUME;

NR:=0;
NR:=NOUA_LISTA_MEDICAMENTE.COUNT;
AUX_NR:=NR;

FOR I IN 1..AUX_NR LOOP
    FOR J IN AUX_LISTA_ALERGII.FIRST..AUX_LISTA_ALERGII.LAST LOOP
        IF NOUA_LISTA_MEDICAMENTE(I)=AUX_LISTA_ALERGII(J) THEN
            FOR K IN I..AUX_NR-1 LOOP
                NOUA_LISTA_MEDICAMENTE(K):=NOUA_LISTA_MEDICAMENTE(K+1);
            END LOOP;
            AUX_NR:=AUX_NR-1;
        END IF;
    END LOOP;
END LOOP;

IF AUX_NR < NR THEN
    DBMS_OUTPUT.PUT_LINE('S-au gasit medicamente la care pacientul ' || NUME || ' are alergii.
Reteta corecta este : ');
    FOR I IN 1..AUX_NR LOOP
        DBMS_OUTPUT.PUT_LINE(NOUA_LISTA_MEDICAMENTE(I));
    END LOOP;
END IF;

```

```

END LOOP;

DBMS_OUTPUT.PUT_LINE('Ne pare rau pentru greseala! Doctorului i se va scadea salariul.');

UPDATE DOCTOR A
SET A.SALARIU =
  (SELECT MIN(B.SALARIU)
   FROM DOCTOR B
   WHERE A.ID_SECTIE=B.ID_SECTIE AND A.ID_SPITAL=B.ID_SPITAL)
   WHERE ID_DOCTOR=COD_DOCTOR;

ELSE
  DBMS_OUTPUT.PUT_LINE('Nu s-au gasit medicamente la care ' || NUME || ' este alergic.');
END IF;

EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20000,'Nu exista pacienti cu numele dat');
  WHEN TOO_MANY_ROWS THEN
    RAISE_APPLICATION_ERROR(-20001,'Exista mai multi pacienti cu numele dat');
  WHEN OTHERS THEN
    RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');

RETURN NOUA_LISTA_MEDICAMENTE;

END F8;

PROCEDURE P9(DENUMIRE_SPITAL.NUME%TYPE)
IS
  DOCTORI TIP_DOCTORI:=TIP_DOCTORI();
  COD_SPITAL SPITAL.ID_SPITAL%TYPE;

```

```

NUME_D DOCTOR.NUME_DOCTOR%TYPE;
PRENUME_D DOCTOR.PRENUME_DOCTOR%TYPE;

BEGIN
    SELECT ID_SPITAL
    INTO COD_SPITAL
    FROM SPITAL
    WHERE NUME=DENUMIRE;

    SELECT DISTINCT D.ID_DOCTOR
    BULK COLLECT INTO DOCTORI
    FROM DOCTOR D
    INNER JOIN DOCTOR_OPERATIE DO ON(D.ID_DOCTOR=DO.ID_DOCTOR)
    INNER JOIN OPERATIE O ON(O.ID_OPERATIE=DO.ID_OPERATIE)
    INNER JOIN OPERATIE_INTERNARE OI ON(OI.ID_OPERATIE=O.ID_OPERATIE)
    INNER JOIN INTERNARE I ON(I.ID_INTERNARE=OI.ID_INTERNARE)
    INNER JOIN SPITAL S ON(S.ID_SPITAL=I.ID_SPITAL)
    INNER JOIN PROGRAM_DOCTOR PD ON(PD.ID_DOCTOR=D.ID_DOCTOR)
    WHERE D.ID_SPITAL=COD_SPITAL AND PD.END_ORA<O.START_ORA;

UPDATE DOCTOR
SET SALARIU = SALARIU + SALARIU * 0.1
WHERE ID_DOCTOR IN
    (SELECT DISTINCT D.ID_DOCTOR
    FROM DOCTOR D
    INNER JOIN DOCTOR_OPERATIE DO ON(D.ID_DOCTOR=DO.ID_DOCTOR)
    INNER JOIN OPERATIE O ON(O.ID_OPERATIE=DO.ID_OPERATIE)
    INNER JOIN OPERATIE_INTERNARE OI ON(OI.ID_OPERATIE=O.ID_OPERATIE)
    INNER JOIN INTERNARE I ON(I.ID_INTERNARE=OI.ID_INTERNARE)

```

```

    INNER JOIN SPITAL S ON(S.ID_SPITAL=I.ID_SPITAL)
    INNER JOIN PROGRAM_DOCTOR PD ON(PD.ID_DOCTOR=D.ID_DOCTOR)
    WHERE D.ID_SPITAL=COD_SPITAL AND PD.END_ORA<O.START_ORA);

IF DOCTORI.COUNT=0 THEN
    DBMS_OUTPUT.PUT_LINE('Nu exista doctori care sa lucreze peste program la spitalul ' || DENUMIRE);
ELSIF DOCTORI.COUNT=1 THEN
    DBMS_OUTPUT.PUT_LINE('Doctorul este caruia i s-a marit salariul este: ');
ELSE
    DBMS_OUTPUT.PUT_LINE('Doctorii carora li s-a marit salariul sunt: ');
END IF;

FOR I IN DOCTORI.FIRST..DOCTORI.LAST LOOP
    SELECT NUME_DOCTOR, PRENUME_DOCTOR
    INTO NUME_D, PRENUME_D
    FROM DOCTOR
    WHERE ID_DOCTOR=DOCTORI(I);
    DBMS_OUTPUT.PUT_LINE(NUME_D || ' ' || PRENUME_D);
END LOOP;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20000,'Nu exista spitale cu numele dat');
    WHEN TOO_MANY_ROWS THEN
        RAISE_APPLICATION_ERROR(-20001,'Există mai multe spitale cu numele dat');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');

END P9;

```

END PACHET13;

The screenshot shows the Oracle SQL Developer interface with the 'anastasia' database selected. The 'Worksheet' tab is active, displaying the PL/SQL code for package PACHET13. The code includes several procedures and functions, such as P6, P7, F8, and P9, which interact with tables like SPITAL, DOCTOR, and PACIENT. A trigger named 'trigger_triviz_dropped' is also present. The status bar at the bottom indicates the task was completed in 0.099 seconds.

```
CREATE OR REPLACE PACKAGE PACHET13 AS
  PROCEDURE P6(DENUMIRE SPITAL.NUME%TYPE);
  PROCEDURE P7(DENUMIRE SPITAL.NUME%TYPE);
  FUNCTION F8(NUME PACIENT.NUME_PACIENT%TYPE)
    RETURN TIP_MEDICAMENTE;
  PROCEDURE P9(DENUMIRE SPITAL.NUME%TYPE);
END PACHET13;

CREATE OR REPLACE PACKAGE BODY PACHET13 IS
  PROCEDURE P6(DENUMIRE SPITAL.NUME%TYPE)
  IS
    TYPE ORE_VECTOR IS VARRAY(100) OF NUMBER(2);
    TYPE ZI_VECTOR IS VARRAY(100) OF VARCHAR2(10);
    TYPE DOCTOR_RECORD IS RECORD
      (Nume DOCTOR Nume DOCTOR%TYPE);

```

Package PACHET13 compiled

The screenshot shows the Oracle SQL Developer interface with the 'anastasia' database selected. The 'Worksheet' tab is active, displaying the PL/SQL code for the package body PACHET13. The code includes an exception handling block that raises application errors for various conditions like no data found or too many rows. The status bar at the bottom indicates the task was completed in 0.106 seconds.

```
  INTO NUME_D, PRENUME_D
  FROM DOCTOR
  WHERE ID_DOCTOR=DOCTOR(I);
  DBMS_OUTPUT.PUT_LINE(NUME_D || ' ' || PRENUME_D);
END LOOP;

EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20000,'Nu exista spitale cu numele dat');
  WHEN TOO_MANY_ROWS THEN
    RAISE_APPLICATION_ERROR(-20001,'Există mai multe spitale cu numele dat');
  WHEN OTHERS THEN
    RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');

END P9;
END PACHET13;
```

Package Body PACHET13 compiled

```

293      WHEN OTHERS THEN
294          RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
295      END P9;
296  END PACHET13;
297
298  DECLARE
299      AUX_MEDICAMENTE RETETA.LISTA_MEDICAMENTE%TYPE;
300  BEGIN
301      AUX_MEDICAMENTE:=PACHET13.F8('Sandu');
302  END;

```

Package Body PACHET13 compiled

Dbsm Output: Buffer Size:20000

anastasia

S-au gasit medicamente la care pacientul Sandu are alergii. Reteta corecta este :

Rompirin
Santepirin
Phelodia
Aspacardin
Aspimax

EXERCITIU 14

--Sa se creeze un pachet care contine :

--prima procedura afiseaza orele la care au loc operatiile unui pacient dat si doctorii care le efectueaza

--a 2-a procedura afiseaza lista subspecializarilor dintr-o sectie data

--a 3-a functie extrage si afiseaza datele pacientilor dintr-un spital dat si returneaza numarul acestora

--a 4-a functie afiseaza numele pacientilor care au fost internati sau au avut programarea pana in data curenta si returneaza numarul acestora.

CREATE OR REPLACE PACKAGE PACHET14 AS

```

TYPE TIP_ORE IS VARRAY(100) OF NUMBER;

TYPE TIP_ID IS VARRAY(100) OF NUMBER;

TYPE TIP_DOCTORI IS VARRAY(100) OF VARCHAR2(30);

TYPE TIP_SUBSPECIALIZARI IS VARRAY(100) OF VARCHAR2(30);

TYPE TIP_PACIENT_NOU IS VARRAY(100) OF VARCHAR2(10);

PROCEDURE P10(NUME_PACIENT.NUME_PACIENT%TYPE);

PROCEDURE P11(DENUMIRE_SECTIE.NUME%TYPE);

```

```

TYPE PACIENT_RECORD IS RECORD
  (NUME PACIENT.NUME_PACIENT%TYPE,
  PRENUME PACIENT.PRENUME_PACIENT%TYPE,
  COD_SPITAL PACIENT.ID_SPITAL%TYPE,
  DATA_PROG PROGRAMARE.DATA_PROGRAMARE%TYPE,
  COD_SECTIE PROGRAMARE.ID_SECTIE%TYPE,
  COD_INTERNARE INTERNARE.ID_INTERNARE%TYPE
);

TYPE TIP_PACIENT IS VARRAY(100) OF PACIENT_RECORD;

FUNCTION F12(COD SPITAL.ID_SPITAL%TYPE)
  RETURN NUMBER;

FUNCTION F13(COD SPITAL.ID_SPITAL%TYPE)
  RETURN NUMBER;

END PACHET14;
/
CREATE OR REPLACE PACKAGE BODY PACHET14 IS
  PROCEDURE P10(NUME PACIENT.NUME_PACIENT%TYPE) IS
    LISTA_ID TIP_ID:=TIP_ID();
    LISTA_ORE TIP_ORE:= TIP_ORE();
    LISTA_DOCTORI TIP_DOCTORI:=TIP_DOCTORI();
    COD_PACIENT PACIENT.ID_PACIENT%TYPE;
    BEGIN
      SELECT ID_PACIENT
        INTO COD_PACIENT
        FROM PACIENT
       WHERE NUME_PACIENT=NUME;

      SELECT DISTINCT O.START_ORA, O.ID_OPERATIE
        BULK COLLECT INTO LISTA_ORE, LISTA_ID

```

```
FROM PACIENT P  
INNER JOIN INTERNARE I ON(I.ID_PACIENT=P.ID_PACIENT)  
INNER JOIN OPERATIE_INTERNARE OI ON(OI.ID_INTERNARE=I.ID_INTERNARE)  
INNER JOIN OPERATIE O ON(O.ID_OPERATIE=OI.ID_OPERATIE)  
WHERE P.ID_PACIENT=COD_PACIENT AND I.STATUS=1;
```

```
IF LISTA_ORE.COUNT = 0 THEN  
    DBMS_OUTPUT.PUT_LINE('Pacientul nu are nicio operatie');  
ELSE  
    DBMS_OUTPUT.PUT_LINE('Orele la care pacientul are operatii sunt : ');  
    FOR I IN LISTA_ORE.FIRST..LISTA_ORE.LAST LOOP  
        DBMS_OUTPUT.PUT_LINE(LISTA_ORE(I));  
        SELECT D.NUME_DOCTOR  
        BULK COLLECT INTO LISTA_DOCTORI  
        FROM OPERATIE O  
        INNER JOIN DOCTOR_OPERATIE DO ON (DO.ID_OPERATIE=O.ID_OPERATIE)  
        INNER JOIN DOCTOR D ON(D.ID_DOCTOR=DO.ID_DOCTOR)  
        WHERE O.ID_OPERATIE=LISTA_ID(I);  
  
        DBMS_OUTPUT.PUT_LINE('Doctorii care lucreaza la operatie sunt :');  
  
        FOR J IN LISTA_DOCTORI.FIRST..LISTA_DOCTORI.LAST LOOP  
            DBMS_OUTPUT.PUT_LINE(LISTA_DOCTORI(J));  
        END LOOP;  
    END LOOP;  
END IF;
```

```
EXCEPTION  
WHEN NO_DATA_FOUND THEN
```

```

        RAISE_APPLICATION_ERROR(-20000,'Nu exista pacienti cu numele dat');

WHEN TOO_MANY_ROWS THEN

        RAISE_APPLICATION_ERROR(-20001,'Exista mai multi pacienti cu numele dat');

WHEN OTHERS THEN

        RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');

END P10;

```

```

PROCEDURE P11(DENUMIRE SECTIE.NUME%TYPE) IS

    COD_SECTIE SECTIE.ID_SECTIE%TYPE;

    LISTA_SUBSPECIALIZARI TIP_SUBSPECIALIZARI:=TIP_SUBSPECIALIZARI();

BEGIN

    SELECT ID_SECTIE

    INTO COD_SECTIE

    FROM SECTIE

    WHERE NUME=DENUMIRE;

    SELECT A.DENUMIRE

    BULK COLLECT INTO LISTA_SUBSPECIALIZARI

    FROM SECTIE S

    INNER JOIN DOCTOR D ON(D.ID_SECTIE=S.ID_SECTIE)

    INNER JOIN SPECIALIZARE A ON(A.ID_SPECIALIZARE=D.ID_SPECIALIZARE)

    INNER JOIN SPECIALIZARE B ON ( A.ID_SPECIALIZARE_MAMA=B.ID_SPECIALIZARE)

    WHERE S.ID_SECTIE=COD_SECTIE;

    IF LISTA_SUBSPECIALIZARI.COUNT = 0 THEN

        DBMS_OUTPUT.PUT_LINE('Sectia nu are niciun doctor cu subspecializare');

    ELSE

        DBMS_OUTPUT.PUT_LINE('Subspecializarile doctorilor sunt : ');

        FOR I IN LISTA_SUBSPECIALIZARI.FIRST..LISTA_SUBSPECIALIZARI.LAST LOOP

```

```

        DBMS_OUTPUT.PUT_LINE(LISTA_SUBSPECIALIZARI(I));

    END LOOP;

END IF;

EXCEPTION

WHEN NO_DATA_FOUND THEN

    RAISE_APPLICATION_ERROR(-20000,'Nu exista sectii cu numele dat');

WHEN TOO_MANY_ROWS THEN

    RAISE_APPLICATION_ERROR(-20001,'Exista mai multi sectii cu numele dat');

WHEN OTHERS THEN

    RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');


```

END P11;

```

FUNCTION F12(COD SPITAL.ID_SPITAL%TYPE)

RETURN NUMBER IS

NR NUMBER;

DATE_PACIENT TIP_PACIENT:=TIP_PACIENT();

BEGIN

SELECT DISTINCT P.NUME_PACIENT, P.PRENUME_PACIENT, P.ID_SPITAL, PROG.DATA_PROGRAMARE,
PROG.ID_SECTIE, I.ID_INTERNARE

BULK COLLECT INTO DATE_PACIENT

FROM SPITAL S

INNER JOIN PACIENT P ON(S.ID_SPITAL=P.ID_SPITAL)

INNER JOIN PROGRAMARE PROG ON(PROG.ID_PACIENT=P.ID_PACIENT)

INNER JOIN INTERNARE I ON(I.ID_PACIENT = P.ID_PACIENT)

WHERE S.ID_SPITAL=COD;

NR:=DATE_PACIENT.COUNT;

```

```

IF NR=0 THEN
    DBMS_OUTPUT.PUT_LINE('Nu exista pacienti la acest spital.');
ELSE
    DBMS_OUTPUT.PUT_LINE('Pacientii si datele lor sunt : ');
    FOR I IN DATE_PACIENT.FIRST..DATE_PACIENT.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(DATE_PACIENT(I).NUME || ' ' || DATE_PACIENT(I).PRENUME || ' ' ||
DATE_PACIENT(I).COD_SPITAL || ' ' || DATE_PACIENT(I).DATA_PROG || ' ' ||
DATE_PACIENT(I).COD_SECTIE || ' ' || DATE_PACIENT(I).COD_INTERNARE);
    END LOOP;
END IF;

RETURN NR;
END F12;

FUNCTION F13(COD_SPITAL.ID_SPITAL%TYPE)
RETURN NUMBER IS
NR NUMBER;
LISTA_PACENTI_NOU TIP_PACIENT_NOU:=TIP_PACENT_NOU();
DATA_CURENTA DATE;
BEGIN
SELECT CURRENT_DATE
INTO DATA_CURENTA
FROM DUAL;

SELECT NUME_PACIENT
BULK COLLECT INTO LISTA_PACENTI_NOU
FROM SPITAL S
INNER JOIN PACIENT P ON(P.ID_SPITAL=S.ID_SPITAL)

```

```
INNER JOIN PROGRAMARE PROG ON(PROG.ID_PACIENT=P.ID_PACIENT)
INNER JOIN INTERNARE I ON(I.ID_SPITAL=S.ID_SPITAL)
WHERE S.ID_SPITAL=COD OR I.STATUS=1 AND PROG.DATA_PROGRAMARE < DATA_CURENTA OR
I.DATA_INTERNARE < DATA_CURENTA;
```

```
NR:=LISTA_PACIENTI_NOU.COUNT;
```

```
IF NR= 0 THEN
```

```
    DBMS_OUTPUT.PUT_LINE('Nu au fost pacienti pana acum.');
```

```
ELSE
```

```
    DBMS_OUTPUT.PUT_LINE('Pacientii internati sau consultati pana acum la spitalul ales sunt : ');
```

```
    FOR I IN LISTA_PACIENTI_NOU.FIRST..LISTA_PACIENTI_NOU.LAST LOOP
```

```
        DBMS_OUTPUT.PUT_LINE(LISTA_PACIENTI_NOU(I));
```

```
    END LOOP;
```

```
END IF;
```

```
RETURN NR;
```

```
END F13;END PACHET14;
```

Oracle SQL Developer : anastasia

Connections Worksheet Query Builder

```

CREATE OR REPLACE PACKAGE PACHET14 AS
    TYPE TIP_ORE IS VARRAY(100) OF NUMBER;
    TYPE TIP_ID IS VARRAY(100) OF NUMBER;
    TYPE TIP_DOCTORI IS VARRAY(100) OF VARCHAR2(30);
    TYPE TIP_SUBSPECIALIZARE IS VARRAY(100) OF VARCHAR2(30);
    TYPE TIP_FACIENT_NOU IS VARRAY(100) OF VARCHAR2(10);
    PROCEDURE P10(NUME_PACIENT,NUME_PACIENT%TYPE);
    PROCEDURE P11(DENUMIRE_SECTIE.NUME%TYPE);
    TYPE PACIENT_RECORD IS RECORD
        (NUME_PACIENT.NUME_PACIENT%TYPE,
         PRENUME_PACIENT.PRENUME_PACIENT%TYPE,
         COD_SPITAL_PACIENT.ID_SPITAL%TYPE,
         DATA_PROGR_PROGRAMARE.DATA_PROGRAMARE%TYPE,
         COD_SECTIE_PROGRAMARE.ID_SECTIE%TYPE,
         COD_INTERNARE_INTERNARE.ID_INTERNARE%TYPE
        );
    END;
END;

```

Script Output x | Query Result x | Explain Plan x | Query Result 2 x

Package Body PACHET14 compiled

Dbsm Output: Buffer Size[20000] anastasia x 14 Compiler - Log Logging message Database... Compiler... I Line 1 Column 1 I Insert I Modified I Windows... Task completed in 0.063 seconds

Reports

File Edit View Navigate Run Source Team Tools Window Help

Oracle SQL Developer : anastasia

Connections Worksheet Query Builder

```

    INNER JOIN PACIENT P ON(P.ID_SPITAL=S.ID_SPITAL)
    INNER JOIN PROGRAMARE PROG ON(PROG.ID_PACIENT=P.ID_PACIENT)
    INNER JOIN INTERNARE I ON(I.ID_SPITAL=S.ID_SPITAL)
    WHERE S.ID_SPITAL=2 AND I.STATUS=1 AND PROG.DATA_PROGRAMARE < DATA_CURENTA AND I.DATA_INTERNARE < DATA_CURENTA
    NR:=LISTA_PACIENTI_NOU.COUNT;

    IF NR= 0 THEN
        DBMS_OUTPUT.PUT_LINE('Nu au fost pacienti pana acum.');
    ELSE
        FOR I IN LISTA_PACIENTI_NOU.FIRST..LISTA_PACIENTI_NOU.LAST LOOP
            DBMS_OUTPUT.PUT_LINE(LISTA_PACIENTI_NOU(I));
        END LOOP;
    END IF;

```

Script Output x | Query Result x | Explain Plan x | Query Result 2 x

Package Body PACHET14 compiled

Dbsm Output: Buffer Size[20000] anastasia x 14 Compiler - Log Logging message Database... Compiler... I Line 171 Column 1 I Insert I Modified I Windows... Task completed in 0.067 seconds

Reports

File Edit View Navigate Run Source Team Tools Window Help

--prima procedura

Oracle SQL Developer : anastasia

File Edit View Navigate Run Source Team Tools Window Help

Connections

Worksheet Query Builder

```
170
171 BEGIN
172     PACHET14.P10('Sandu');
173 END;
```

Script Output X Query Result X Explain Plan X Query Result 2 X

PL/SQL procedure successfully completed.

Dbsm Output

anastasia

```
Orele la care pacientul are operatii sunt :
14
```

```
Doctorii care lucreaza la operatie sunt :
Gandore
Nistor
Hurloiu
```

Compiler - Log

messages Logging mode: automatic Compiler

Line 173 Column 5 Insert Modified Windows: C

This screenshot shows the Oracle SQL Developer interface. The 'Connections' tree on the left lists several tables and procedures under the 'anastasia' schema. In the 'Worksheet' tab, a PL/SQL block is run, resulting in the message 'PL/SQL procedure successfully completed.' Below the worksheet, the 'Dbsm Output' window displays the results of the query: 'Orele la care pacientul are operatii sunt : 14' and 'Doctorii care lucreaza la operatie sunt : Gandore, Nistor, Hurloiu'. The status bar at the bottom indicates the line number (173), column (5), and current mode (Modified).

Oracle SQL Developer : anastasia

File Edit View Navigate Run Source Team Tools Window Help

Connections

Worksheet Query Builder

```
167     END F13;
168 END PACHET14;
169
170
171 BEGIN
172     PACHET14.P10('PROJECTSGBD');
173 END;
```

Script Output X Query Result X Explain Plan X Query Result 2 X

Task completed in 0.062 seconds

Error report -
ORA-20000: Nu exista pacienti cu numele dat
ORA-06512: at "ANASTASIA.PACHET14", line 44
ORA-06512: at line 2
20000. 00000 - "%s"
*Cause: The stored procedure 'raise_application_error'
was called which causes this error to be generated.

Dbsm Output

anastasia

Compiler - Log

messages Logging mode: automatic Compiler

Line 173 Column 5 Insert Modified Windows: C

This screenshot shows the Oracle SQL Developer interface. Similar to the first one, it displays the 'anastasia' schema's tables and procedures. In the 'Worksheet' tab, a different PL/SQL block is run, which fails due to a missing patient record. The error message 'ORA-20000: Nu exista pacienti cu numele dat' is displayed in the 'Dbsm Output' window. The status bar at the bottom shows the line number (173), column (5), and current mode (Modified).

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, the 'Connections' tree is visible, showing various database objects like 'anastasia' and 'INTERNARE'. The main workspace contains a 'Worksheet' tab with the following PL/SQL code:

```

166
167     END F13;
168
169
170
171 BEGIN
172     PACHET14.P10('Dirtu');
173 END;
174
175

```

Below the code, a message reads: "PL/SQL procedure successfully completed." The bottom pane, 'Dbsn Output', shows the message: "Pacientul nu are nicio operatie". The system tray at the bottom right indicates the date as 1/13/2023 and the time as 10:32 PM.

--pentru a exemplifica exceptia 'TOO MANY ROWS'

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab contains the following PL/SQL code:

```

174
175
176 DECLARE
177     alergii TIP_ALERGII:=TIP_ALERGII();
178 BEGIN
179     alergii.extend;
180     alergii(1):='Asmetix';
181     INSERT INTO PACIENT
182     VALUES (10,2,'Sandu','Maria',34,'Feminin',alergii,'0754766823');
183 END;
184

```

Below the code, a message reads: "PL/SQL procedure successfully completed." The bottom pane, 'Dbsn Output', shows the message: "Nu au fost pacienti pana acum." The system tray at the bottom right indicates the date as 1/13/2023 and the time as 10:41 PM.

Oracle SQL Developer : anastasia

```

    ELSE
        FOR I IN LISTA_PACIENTI_NOU.FIRST..LISTA_PACIENTI_NOU.LAST LOOP
            DBMS_OUTPUT.PUT_LINE(LISTA_PACIENTI_NOU(I));
        END LOOP;
    END IF;

    RETURN NR;
END F13;
END PACHET14;

BEGIN
    PACHET14.P10('Sandu');
END;

```

Error report -

```

ORA-20001: Exista mai multi pacienti cu numele dat
ORA-06512: at "ANASTASIA.PACHET14", line 46
ORA-06512: at line 2

```

Doms Output:

Compiler Log Logging Page Documentation Help

Line 171 Column 5 Insert Modified Windows

--a doua functie

Oracle SQL Developer : anastasia

```

    END PACHET14;
168
169
170 BEGIN
171     PACHET14.P11('Pneumonie');
172 END;
173
174
175
176

```

PL/SQL procedure successfully completed.

Doms Output:

anastasia

Subspecializarile doctorilor sunt :

Cardio-oncologie

Chirurgie vasculara

Compiler Log Logging Page Documentation Help

Line 170 Column 1 Insert Modified Windows

Oracle SQL Developer : anastasia

```

167    END PACHET14;
168
169
170 BEGIN
171     PACHET14.P11('PROIECTSGBD');
172 END;
173
174
175
176
177
178
179

```

Error report -

```

ORA-20000: Nu exista sectii cu numele dat
ORA-06512: at "ANASTASIA.PACHET14", line 79
ORA-06512: at line 2
20000. 00000 - "%s"
*Cause:  The stored procedure 'raise_application_error'
          was called which causes this error to be generated.

```

Doms Output:

Compiler - Log Logging page anastasia Compiler

Line 172 Column 5 | Insert | Modified | Windows

Oracle SQL Developer : anastasia

```

164
165     RETURN NR;
166   END F13;
167 END PACHET14;
168
169
170 BEGIN
171     PACHET14.P11('Cardiologie');
172 END;
173
174

```

PL/SQL procedure successfully completed.

Doms Output:

anastasia

Compiler - Log Logging page anastasia Compiler

Line 172 Column 5 | Insert | Modified | Windows

--a 3 a functie

Oracle SQL Developer : anastasia

```

166      END F13;
167  END PACHET14;
168
169  DECLARE
170    NR NUMBER;
171  BEGIN
172    NR:=PACHET14.F12(2);
173  END;
174
175
176
PL/SQL procedure successfully completed.

Doms Output
anastasia x
Compiler - Log
messages Logging message Database message Compiler message
Buffer Size[20000] | Line 173 Column 5 | Insert | Modified | Windows | 10:37 PM | 499 | 39°F | ENG | 1/13/2023

```

Pacientii si datele lor sunt :
Sandu Anastasia 2 30-APR-24 06.00.00.000000 PM 5 2

Oracle SQL Developer : anastasia

```

166      END F13;
167  END PACHET14;
168
169  DECLARE
170    NR NUMBER;
171  BEGIN
172    NR:=PACHET14.F12(3);
173  END;
174
175
176
PL/SQL procedure successfully completed.

Doms Output
anastasia x
Compiler - Log
messages Logging message Database message Compiler message
Buffer Size[20000] | Line 173 Column 5 | Insert | Modified | Windows | 10:37 PM | 499 | 39°F | ENG | 1/13/2023

```

Nu exista pacienti la acest spital.

--ultima functie

Oracle SQL Developer : anastasia

File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Query Builder

```
166      END F13;
167  END PACHET14;
168 :
169  DECLARE
170    NR NUMBER;
171  BEGIN
172    NR:=PACHET14.F13(3);
173  END;
```

Script Output X | Query Result X | Explain Plan X | Query Result 2 X

PL/SQL procedure successfully completed.

Dbsn Output Buffer Size:20000 anastasia

Nu au fost pacienti pana acum.

Compiler - Log messages Logging mode: automatic Compiler

Line 173 Column 5 Insert Modified Windows: O

```
166      RETURN NR;
167  END F13;
168  END PACHET14;
169 :
170  DECLARE
171    NR NUMBER;
172  BEGIN
173    NR:=PACHET14.F13(5);
174  END;
```

PL/SQL procedure successfully completed.

Dbsn Output Buffer Size:20000 anastasia

Pacientii internati pana acum la spitalul ales sunt :

Mihaielescu
Oancea
Sandu

Compiler - Log messages Logging mode: automatic Compiler

Line 174 Column 5 Insert Modified Windows: O

Oracle SQL Developer : anastasia

File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Query Builder

```
166      RETURN NR;
167  END F13;
168  END PACHET14;
169 :
170  DECLARE
171    NR NUMBER;
172  BEGIN
173    NR:=PACHET14.F13(5);
174  END;
```

Script Output X | Query Result X | Explain Plan X | Query Result 2 X

PL/SQL procedure successfully completed.

Dbsn Output Buffer Size:20000 anastasia

Pacientii internati pana acum la spitalul ales sunt :

Mihaielescu
Oancea
Sandu

Compiler - Log messages Logging mode: automatic Compiler

Line 174 Column 5 Insert Modified Windows: O

```
166      RETURN NR;
167  END F13;
168  END PACHET14;
169 :
170  DECLARE
171    NR NUMBER;
172  BEGIN
173    NR:=PACHET14.F13(5);
174  END;
```

PL/SQL procedure successfully completed.

Dbsn Output Buffer Size:20000 anastasia

Pacientii internati pana acum la spitalul ales sunt :

Mihaielescu
Oancea
Sandu

Compiler - Log messages Logging mode: automatic Compiler

Line 174 Column 5 Insert Modified Windows: O