

# **Administrarea unui Spital**

**Pasare Roxana-Francisca, grupa 243**

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

Baza de date pe care am decis sa o creez contine informatii despre gestiunea unui spital, despre departamentele si saloanele acestuia, dar si despre doctorii care lucreaza in cadrul sau. Acestia urmaresc evolutia pacientilor , dar si starea lor rezultata din evaluarea realizata intr-un salon.

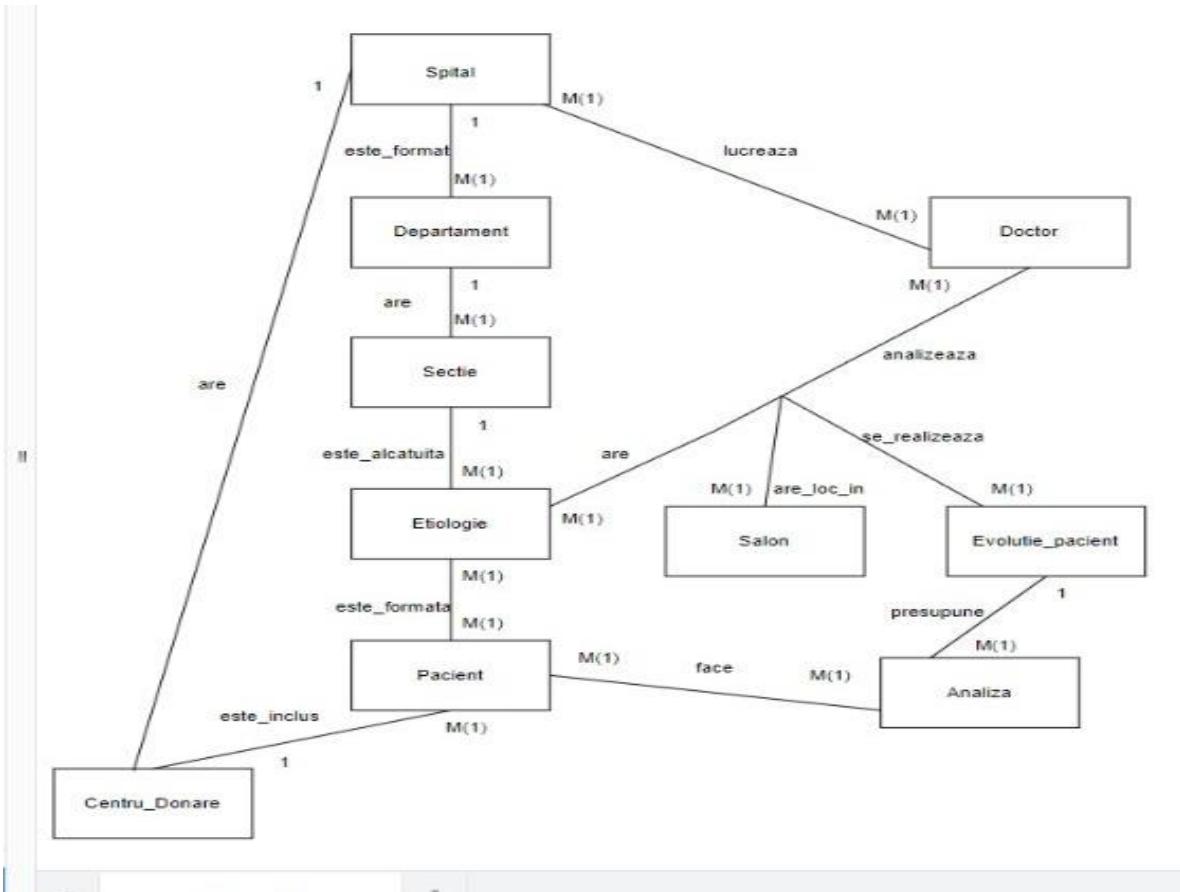
Fiecare pacient este repartizat la un department, intr-o sectie si intr-un salon, participand la anumite studii realizate de catre doctori. Doctorii au posibilitatea de a lucra la mai multe spitale, iar pacientii se pot inscrie la centrele de donare in cadrul unui spital.

Evolutia pacientilor este in functie de etiologie, fiind urmarita in incinta diferitor saloane in cadrul spitalului.

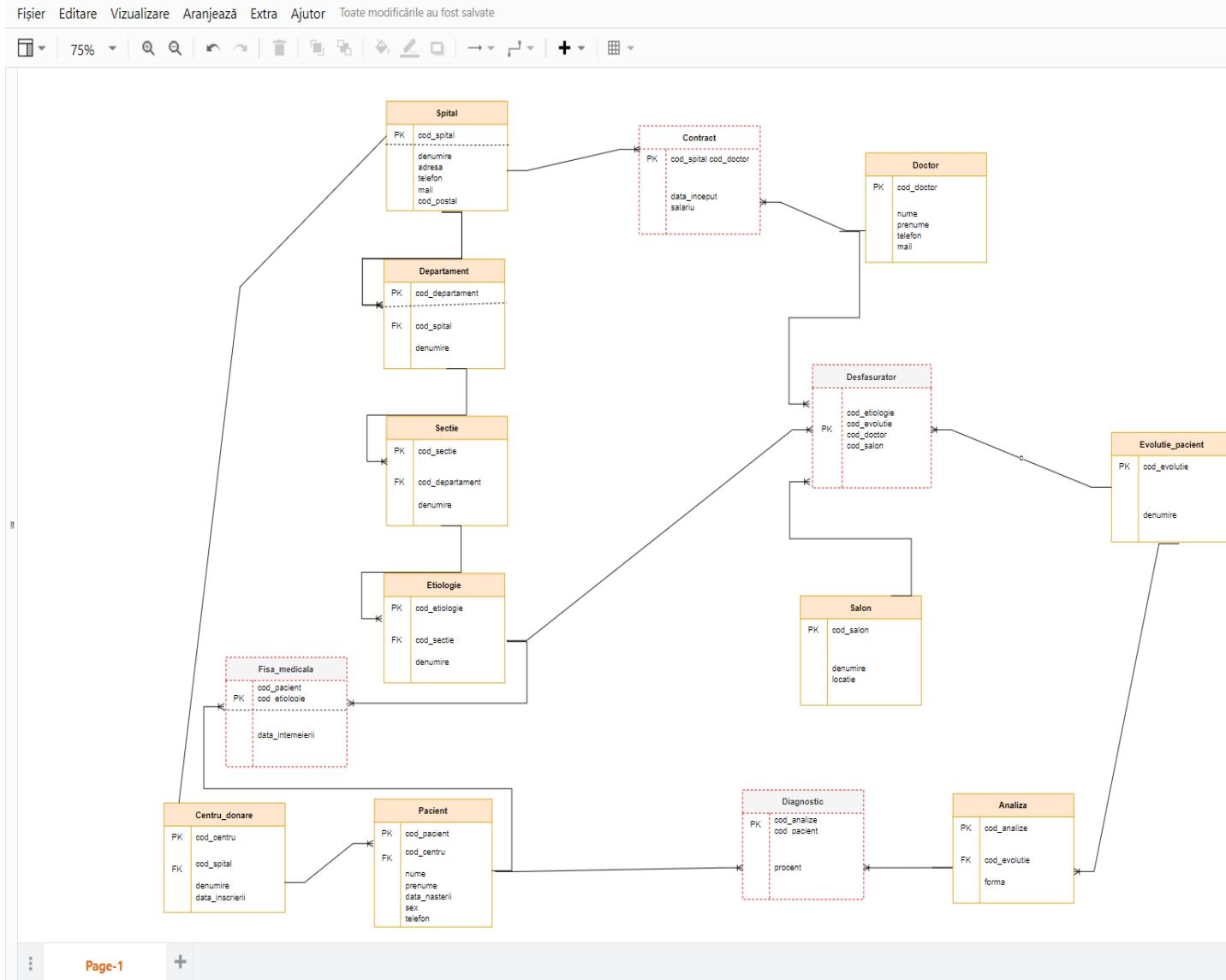
Un doctor are contracte cu spitalele in care lucreaza, acesta putand sa aiba una sau mai multe specializari in cadrul unui spital.

O analiza se poate realiza in mai multe moduri: prin sange, prin urina si prin saliva. Rezultatul acesteia difera in functie de afectiunile pacientului. Analizele unui pacient sunt considerate perfecte daca peste 95% din rezultate au valori normale.

## **2. Realizați diagrama entitate-relație (ERD).**



**3. Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare.**



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

**5. Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).**

## ----Tabel SPITAL----

**CREATE TABLE SPITAL(COD\_SPITAL NUMBER(6) CONSTRAINT PKEY\_SPITAL PRIMARY KEY,**

```
DENUMIRE VARCHAR(100) CONSTRAINT DENUMIRE_SPITAL NOT NULL,  
ADRESA VARCHAR(100) CONSTRAINT ADRESA_SPITAL NOT NULL,  
TELEFON VARCHAR(20) CONSTRAINT TELEFON_SPITAL NOT NULL,  
MAIL VARCHAR(50) UNIQUE,  
COD_POSTAL VARCHAR(6)  
);
```

--inserari spital

**INSERT INTO SPITAL**

```
VALUES(1,'Spitalul Universitar de Urgente Bucuresti', 'St. Splaiul independentei nr.169', '021-318-0523',  
'universitarurgente@gamil.com', '210728');
```

**INSERT INTO SPITAL**

```
VALUES(2,'Spitalul Clinic CF2 Bucuresti', 'Bd. Marasesti nr. 63', '037-229-8711', 'clinicurgentecf@gamil.com', '284928');
```

**INSERT INTO SPITAL**

```
VALUES(3,'Spitalul CF Witting', 'St. Calea Plevei nr. 142-144', '021-317-0068', 'secretariat@spitalcfwiting.com', '284263');
```

**INSERT INTO SPITAL**

```
VALUES(4,'Spitalul Universitar de Urgenta Elias', 'Bd. Marasti nr. 17', '021-312-0465', 'spital@elias.com', '837239');
```

**INSERT INTO SPITAL**

```
VALUES(5,'Spitalul Clinic Coltea', 'Bd. Ion C. Bratianu nr. 17', '031-333-2873', 'urgente@coltea.com', '906232');
```

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, tabs include 'Project.sql', 'Project', 'CONTRACT', 'Dbms Output', and 'Project\_Continuare'. The main area is a 'Worksheet' tab with the following SQL code:

```

CREATE TABLE SPITAL
VALUES(3,'Spitalul CF Witting', 'St. Calea Plevei nr. 142-144', '021-317-0068', 'secretariat@spitalcfwiting.com', '284263');

CREATE TABLE SPITAL
VALUES(4,'Spitalul Universitar de Urgenta Elias', 'Bd. Marasti nr. 17', '021-312-0465', 'spital@elias.com', '837239');

CREATE TABLE SPITAL
VALUES(5,'Spitalul Clinic Coltea', 'Bd. Ion C. Bratianu nr. 17', '031-333-2873', 'urgente@coltea.com', '906232');

SELECT * FROM SPITAL;

--tabel centru_donare
CREATE TABLE CENTRU_DONARE(COD_CENTRU NUMBER(5) CONSTRAINT PKEY_CENTRU PRIMARY KEY,
                           DENUMIRE VARCHAR(100) CONSTRAINT DENUMIRE_CENTRU NOT NULL,
                           DATA_INSCRIERII DATE CONSTRAINT DATA_INSCRIERII_FIXA NOT NULL,
                           COD_SPITAL NUMBER(5), CONSTRAINT FK_CENTRU FOREIGN KEY(COD_SPITAL) REFERENCES SPITAL(COD_SPITAL));

--inserari centru_donare
INSERT INTO CENTRU_DONARE

```

Below the code, the 'Script Output' tab shows the results of the insertions:

COD_SPITAL	DENUMIRE	ADRESA	TELEFON	MAIL	COD_POSTAL
1	1 Spitalul Universitar de Urgente Bucuresti	St. Splaiul Independentei nr.169	021-318-0523	universitarurgente@gmail.com	210728
2	2 Spitalul Clinic CF2 Bucuresti	Bd. Marasesti nr. 63	037-229-8711	clinicurgenteef@gmail.com	284263
3	3 Spitalul CF Witting	St. Calea Plevei nr. 142-144	021-317-0068	secretariat@spitalcfwiting.com	284263
4	4 Spitalul Universitar de Urgenta Elias	Bd. Marasti nr. 17	021-312-0465	spital@elias.com	837239
5	5 Spitalul Clinic Coltea	Bd. Ion C. Bratianu nr. 17	031-333-2873	urgente@coltea.com	906232

The status bar at the bottom indicates 'All Rows Fetched: 5 in 0.005 seconds'.

### --Tabel CENTRU\_DONARE---

```

CREATE TABLE CENTRU_DONARE(COD_CENTRU NUMBER(5) CONSTRAINT PKEY_CENTRU PRIMARY KEY,
                           DENUMIRE VARCHAR(100) CONSTRAINT DENUMIRE_CENTRU NOT NULL,
                           DATA_INSCRIERII DATE CONSTRAINT DATA_INSCRIERII_FIXA NOT NULL,
                           COD_SPITAL NUMBER(5), CONSTRAINT FK_CENTRU FOREIGN KEY(COD_SPITAL) REFERENCES
                           SPITAL(COD_SPITAL));

```

--inserari centru\_donare

```

INSERT INTO CENTRU_DONARE
VALUES(10,'CDSUUB',TO_DATE('07-02-2010','DD-MM-YYYY'),1);

```

```

INSERT INTO CENTRU_DONARE
VALUES(11,'CDSCCB',TO_DATE('17-04-2007','DD-MM-YYYY'),2);

```

```

INSERT INTO CENTRU_DONARE
VALUES(12,'CDSCW',TO_DATE('27-05-2011','DD-MM-YYYY'),3);

```

```

INSERT INTO CENTRU_DONARE
VALUES(13,'CDSUUE',TO_DATE('15-02-2010','DD-MM-YYYY'),4);

```

```

INSERT INTO CENTRU_DONARE
VALUES(14,'CDSCC',TO_DATE('01-12-2004','DD-MM-YYYY'),5);

```

Oracle SQL Developer : Project\_CentruDonare

File Edit View Navigate Run Source Team Tools Window Help

Project.sql Project CONTRACT Dbms Output Project\_CentruDonare

CREATE TABLE 13 of 16 Worksheet Query Builder

```

VALUES(11,'CDSCCB',TO_DATE('17-04-2007','DD-MM-YYYY'),2);

INSERT INTO CENTRU_DONARE
VALUES(12,'CDSCW',TO_DATE('27-05-2011','DD-MM-YYYY'),3);

INSERT INTO CENTRU_DONARE
VALUES(13,'CDSUUE',TO_DATE('15-02-2010','DD-MM-YYYY'),4);

INSERT INTO CENTRU_DONARE
VALUES(14,'CDSCC',TO_DATE('01-12-2004','DD-MM-YYYY'),5);

SELECT * FROM CENTRU_DONARE;

---departament---
CREATE TABLE DEPARTAMENT(COD_DEPARTAMENT NUMBER(5) CONSTRAINT PKEY_DEPARTAMENT PRIMARY KEY,
DENUMIRE VARCHAR(100) CONSTRAINT DENUMIRE_DEPARTAMENT NOT NULL,
COD_SPITAL NUMBER(5), CONSTRAINT FK_DEPARTAMENT FOREIGN KEY(COD_SPITAL) REFERENCES SPITAL(COD_SPITAL));

```

Script Output | Query Result | Query Result 1 | Query Result 2 | All Rows Fetched: 5 in 0.004 seconds

COD_CENTRU	DENUMIRE	DATA_INSCRIERII	COD_SPITAL
1	10 CDSUUE	07-FEB-10	1
2	11 CDSCCB	17-APR-07	2
3	12 CDSCW	27-MAY-11	3
4	13 CDSUUE	15-FEB-10	4
5	14 CDSCC	01-DEC-04	5

Messages - Log

17°C Cloudy

### ---Tabel DEPARTAMENT---

```

CREATE TABLE DEPARTAMENT(COD_DEPARTAMENT NUMBER(5) CONSTRAINT PKEY_DEPARTAMENT PRIMARY KEY,
DENUMIRE VARCHAR(100) CONSTRAINT DENUMIRE_DEPARTAMENT NOT NULL,
COD_SPITAL NUMBER(5), CONSTRAINT FK_DEPARTAMENT FOREIGN KEY(COD_SPITAL) REFERENCES SPITAL(COD_SPITAL));

```

```
INSERT INTO DEPARTAMENT
```

```
VALUES(20,'URGENTE',1);
```

```
INSERT INTO DEPARTAMENT
```

```
VALUES(21,'ATI',1);
```

```
INSERT INTO DEPARTAMENT
```

```
VALUES(22,'CHIRURGIE GENERALA',1);
```

```
INSERT INTO DEPARTAMENT
```

```
VALUES(23,'MEDICINA INTERNA',2);
```

```
INSERT INTO DEPARTAMENT
```

```
VALUES(24,'MEDICINA LEGALA',2);
```

```
INSERT INTO DEPARTAMENT
```

```
VALUES(25,'MEDICINA PRIMARA',2);
```

source Team Tools Window Help

Welcome Page Project

Worksheet Query Builder

```
--departament--
CREATE TABLE DEPARTAMENT(COD_DEPARTAMENT NUMBER(5) CONSTRAINT PKEY_DEPARTAMENT PRIMARY KEY,
    DENUMIRE VARCHAR(100) CONSTRAINT DENUMIRE_DEPARTAMENT NOT NULL,
    COD_SPITAL NUMBER(5), CONSTRAINT FK_DEPARTAMENT FOREIGN KEY(COD_SPITAL) REFERENCES SPITAL(COD_SPITAL));

INSERT INTO DEPARTAMENT
VALUES(20,'URGENTE',1);

INSERT INTO DEPARTAMENT
VALUES(21,'ATI',1);

INSERT INTO DEPARTAMENT
VALUES(22,'CHIRURGIE GENERALA',1);

INSERT INTO DEPARTAMENT
VALUES(23,'MEDICINA INTERNA',2);

INSERT INTO DEPARTAMENT
VALUES(24,'MEDICINA LEGALA',2);

INSERT INTO DEPARTAMENT
VALUES(25,'MEDICINA PRIMARA',2);

SELECT * FROM DEPARTAMENT;
```

Query Result | All Rows Fetched: 6 in 0.004 seconds

	COD_DEPARTAMENT	DENUMIRE	COD_SPITAL
1	20 URGENTE	1	
2	21 ATI	1	
3	22 CHIRURGIE GENERALA	1	
4	23 MEDICINA INTERNA	2	
5	24 MEDICINA LEGALA	2	
6	25 MEDICINA PRIMARA	2	

### ---Tabel SECTIE--

```
CREATE TABLE SECTIE(COD_SECTIE NUMBER(5) CONSTRAINT PKEY_SECTIE PRIMARY KEY,
    DENUMIRE VARCHAR(100) CONSTRAINT DENUMIRE_SECTIE NOT NULL,
    COD_DEPARTAMENT NUMBER(5), CONSTRAINT FK_SECTIE FOREIGN KEY(COD_DEPARTAMENT) REFERENCES
    DEPARTAMENT(COD_DEPARTAMENT));
```

### --INSERARE SECTIE

```
INSERT INTO SECTIE
VALUES(30,'GASTROLOGIE',20);
```

```
INSERT INTO SECTIE
```

```
VALUES(31,'UROLOGIE',20);
```

```
INSERT INTO SECTIE
```

```
VALUES(32,'CARDIOLOGIE',21);
```

```
INSERT INTO SECTIE
```

```
VALUES(33,'GINECOLOGIE',21);
```

```
INSERT INTO SECTIE
```

```
VALUES(34,'ORTOPEDIE',22);
```

```
INSERT INTO SECTIE
```

```
VALUES(35,'PEDIATRIE',22);
```

```
SELECT * FROM SECTIE;
```

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

- Toolbar:** File, Edit, View, Navigate, Run, Source, Team, Tools, Window, Help.
- Project:** Project\_Continuare
- Script Editor:** Contains the following SQL code:

```
CREATE TABLE ETIOLOGIE(COD_ETIOLOGIE NUMBER(5) CONSTRAINT PKEY_ETIOLOGIE PRIMARY KEY,
DENUMIRE VARCHAR(100) CONSTRAINT DENUMIRE_ETIOLOGIE NOT NULL,
COD_SECTIE NUMBER(5), CONSTRAINT FK_ETIOLOGIE FOREIGN KEY(COD_SECTIE) REFERENCES SECTIE(COD_SECTIE));
```
- Worksheet:** Shows the results of the SELECT query:

```
SELECT * FROM SECTIE;
```

Output:

COD_SECTIE	DENUMIRE	COD_DEPARTAMENT
1	30 GASTROLOGIE	20
2	31 UROLOGIE	20
3	32 CARDIOLOGIE	21
4	33 GINECOLOGIE	21
5	34 ORTOPEDIE	22
6	35 PEDIATRIE	22
- Script Output:** All Rows Fetched: 6 in 0.003 seconds
- System Status Bar:** 17°C Cloudy, Search, Taskbar icons.

-----TABEL PACIENT----

```
CREATE TABLE PACIENT(COD_PACIENT NUMBER(5) CONSTRAINT PKEY_PACIENT PRIMARY KEY,
NUME VARCHAR(100) CONSTRAINT NUME_PACIENT NOT NULL,
PRENUME VARCHAR(100) CONSTRAINT PRENUME_PACIENT NOT NULL,
DATA_NASTERII DATE CONSTRAINT DATA_NASTERII_FIXA NOT NULL,
SEX VARCHAR(10) CONSTRAINT SEX_PACIENT NOT NULL,
TELEFON VARCHAR(20) CONSTRAINT TELEFON_PACIENT NOT NULL,
MAIL VARCHAR(50) UNIQUE,
COD_CENTRU NUMBER(5), CONSTRAINT FK_PACIENT FOREIGN KEY(COD_CENTRU) REFERENCES
CENTRU_DONARE(COD_CENTRU);
```

INSERT INTO PACIENT

```
VALUES(51,'POPESCU', 'IRINA', TO_DATE('24-04-1993','DD-MM-YYYY'),'F','0745-263-972','popescuirina@yahoo.com',10);
```

INSERT INTO PACIENT

```
VALUES(61,'POPESCU', 'DAN', TO_DATE('12-09-1990','DD-MM-YYYY'),'M','0790-263-972','popescudan@yahoo.com',10);
```

INSERT INTO PACIENT

```
VALUES(52,'IONESCU', 'MIHAI', TO_DATE('25-11-1989','DD-MM-YYYY'),'M','0765-433-823','ionescumihai25@gmail.com',10);
```

INSERT INTO PACIENT

```
VALUES(53,'DUMITRESCU', 'DAN', TO_DATE('11-08-1997','DD-MM-YYYY'),'M','0734-435-893','dandumitrescu8@yahoo.com',10);
```

INSERT INTO PACIENT

```
VALUES(54,'CONSTANTINESCU', 'LUMINITA', TO_DATE('24-12-2000','DD-MM-YYYY'),'F','0775-678-543','cntlumy@yahoo.com',null);
```

INSERT INTO PACIENT

```
VALUES(55,'IROD', 'ANDREEA', TO_DATE('21-09-1987','DD-MM-YYYY'),'F','0744-098-322','andreeairod@gmail.com',null);
```

INSERT INTO PACIENT

```
VALUES(56,'MARCU', 'ELY', TO_DATE('21-09-1984','DD-MM-YYYY'),'F','0789-536-645','marcuely@gmail.com',11);
```

INSERT INTO PACIENT

```
VALUES(57,'PATRASCU', 'CLAUDIU', TO_DATE('25-02-1967','DD-MM-YYYY'),'M','0795-543-657','clapatrascu@gmail.com',null);
```

**INSERT INTO PACIENT**

```
VALUES(58,'RUS', 'ALIN',TO_DATE('08-03-1945','DD-MM-YYYY'),'M','0754-523-535','alinrus8@yahoo.com',11);
```

**INSERT INTO PACIENT**

```
VALUES(59,'EFTENIE', 'ROBERTA', TO_DATE('23-11-1987','DD-MM-YYYY'),'F','0744-235-345','robyeftenoiu@yahoo.com',null);
```

**INSERT INTO PACIENT**

```
VALUES(60,'IFTIM', 'ANDREEA', TO_DATE('21-04-1998','DD-MM-YYYY'),'F','0735-454-322','iftimandreea21@gmail.com',10);
```

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. Below the menu is a toolbar with various icons. The main workspace has tabs for Project.sql, Project, CONTRACT, Dbms Output, and Project\_Continue. A CREATE TABLE dialog is open above a Worksheet tab. The Worksheet tab contains the following SQL code:

```
CREATE TABLE PACIENT
(
    COD_PACIENT NUMBER(5) CONSTRAINT PKEY_PACIENT PRIMARY KEY,
    NUME VARCHAR(50),
    PRENUME VARCHAR(50),
    DATA_NASTERII DATE,
    SEX CHAR(1),
    TELEFON NUMBER(10),
    MAIL VARCHAR(100),
    COD_CENTRU NUMBER(5)
);

INSERT INTO PACIENT
VALUES (57, 'POPESCU', 'CLAUDIU', TO_DATE('25-02-1967','DD-MM-YYYY'), 'M', 0745-263-972, popescuirina@yahoo.com, 10);

INSERT INTO PACIENT
VALUES(58, 'RUS', 'ALIN',TO_DATE('08-03-1945','DD-MM-YYYY'),'M','0754-523-535','alinrus8@yahoo.com',11);

INSERT INTO PACIENT
VALUES(59, 'EFTENIE', 'ROBERTA', TO_DATE('23-11-1987','DD-MM-YYYY'),'F','0744-235-345','robyeftenoiu@yahoo.com',null);

INSERT INTO PACIENT
VALUES(60, 'IFTIM', 'ANDREEA', TO_DATE('21-04-1998','DD-MM-YYYY'),'F','0735-454-322','iftimandreea21@gmail.com',10);
```

Below the code, the Script Output tab shows the message "All Rows Fetched: 12 in 0.004 seconds". The main pane displays a grid of 12 rows of data from the PACIENT table:

COD_PACIENT	NUME	PRENUME	DATA_NASTERII	SEX	TELEFON	MAIL	COD_CENTRU
1	51 POPESCU	IRINA	24-APR-93	F	0745-263-972	popescuirina@yahoo.com	10
2	61 POPESCU	DAN	12-SEP-90	M	0790-263-972	popescudan@yahoo.com	10
3	52 IONESCU	MIHAI	25-NOV-89	M	0765-433-823	ionescumihai25@gmail.com	10
4	53 DUMITRESCU	DAN	11-AUG-97	M	0734-435-893	dandumitrescu8@yahoo.com	10
5	54 CONSTANTINESCU	LUMINITA	24-DEC-00	F	0775-678-543	cntlumy@yahoo.com	(null)
6	55 IROD	ANDREEA	21-SEP-87	F	0744-098-322	andreeairod@gmail.com	(null)
7	56 MARCU	ELY	21-SEP-84	F	0789-536-645	marcuely@gmail.com	11
8	57 PATRASCU	CLAUDIU	25-FEB-67	M	0795-543-657	clauapatrascu@gmail.com	(null)
9	58 RUS	ALIN	08-MAR-45	M	0754-523-535	alinrus8@yahoo.com	11
10	59 EFTEHIE	ROBERTA	23-NOV-87	F	0744-235-345	robyeftenoiu@yahoo.com	(null)
11	60 IFTIM	ANDREEA	21-APR-98	F	0735-454-322	iftimandreea21@gmail.com	10
12	64 POP	EMA	14-SEP-95	F	0724-343-354	emapop@yahoo.com	10

### ---Tabel ETIOLOGIE---

```
CREATE TABLE ETIOLOGIE(COD_ETIOLOGIE NUMBER(5) CONSTRAINT PKEY_ETIOLOGIE PRIMARY KEY,
DENUMIRE VARCHAR(100) CONSTRAINT DENUMIRE_ETIOLOGIE NOT NULL,
COD_SECTIE NUMBER(5), CONSTRAINT FK_ETIOLOGIE FOREIGN KEY(COD_SECTIE) REFERENCES
SECTIE(COD_SECTIE));
```

### ---INSERARE ETIOLOGIE--

```
INSERT INTO ETIOLOGIE
```

```
VALUES(40,'TOXICA',30);
```

```
INSERT INTO ETIOLOGIE
```

```
VALUES(41,'CONGENITALA',30);
```

```
INSERT INTO ETIOLOGIE
```

```
VALUES(42,'ISCHEMICA',31);
```

```
INSERT INTO ETIOLOGIE
```

```
VALUES(43,'INFECTIONASA',31);
```

```
INSERT INTO ETIOLOGIE
```

```
VALUES(44,'RENALA',31);
```

```
INSERT INTO ETIOLOGIE
```

```
VALUES(45,'ETIOLOGIE GENETICA',32);
```

```
INSERT INTO ETIOLOGIE
```

```
VALUES(46,'IDIOPATICA',32);
```

### ---Tabel DOCTOR---

```
CREATE TABLE DOCTOR(COD_DOCTOR NUMBER(5) CONSTRAINT PKEY_DOCTOR PRIMARY KEY,
NUME VARCHAR(100) CONSTRAINT NUME_DOCTOR NOT NULL,
PRENUME VARCHAR(100) CONSTRAINT PRENUME_DOCTOR NOT NULL,
TELEFON VARCHAR(20) CONSTRAINT TELEFON_DOCTOR NOT NULL,
MAIL VARCHAR(50) UNIQUE);
```

#### INSERT INTO DOCTOR

```
VALUES(100,'MARTIN', 'ILEANA','0767-263-342','martinileana@yahoo.com');
```

#### INSERT INTO DOCTOR

```
VALUES(101,'DIACONESCU', 'MIHAI','0732-453-535','diaconescumihai@gmail.com');
```

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes Source, Team, Tools, Window, and Help. Below the menu is a toolbar with various icons. The main area is divided into two panes: a Worksheet pane on the left and a Query Builder pane on the right.

The Worksheet pane contains the following SQL code:

```
--ETIOLOGIE--  
CREATE TABLE ETIOLOGIE(COD_ETIOLOGIE NUMBER(5) CONSTRAINT PKEY_ETIOLOGIE PRIMARY KEY,  
DENUMIRE VARCHAR(100) CONSTRAINT DENUMIRE_ETIOLOGIE NOT NULL,  
COD_SECTIE NUMBER(5), CONSTRAINT FK_ETIOLOGIE FOREIGN KEY(COD_SECTIE) REFERENCES SECTIE(COD_SECTIE));  
  
--INSERARE ETIOLOGIE--  
INSERT INTO ETIOLOGIE  
VALUES(40, 'TOXICA', 30);  
  
INSERT INTO ETIOLOGIE  
VALUES(41, 'CONGENITALA', 30);  
  
INSERT INTO ETIOLOGIE  
VALUES(42, 'ISCHEMICA', 31);  
  
INSERT INTO ETIOLOGIE  
VALUES(43, 'INFECTIOASA', 31);  
  
INSERT INTO ETIOLOGIE  
VALUES(44, 'RENALA', 31);  
  
INSERT INTO ETIOLOGIE  
VALUES(45, 'ETIOLOGIE GENETICA', 32);
```

The Query Result pane at the bottom shows the results of the last query:

	COD_ETIOLOGIE	DENUMIRE	COD_SECTIE
1	40	TOXICA	30
2	41	CONGENITALA	30
3	42	ISCHEMICA	31
4	43	INFECTIOASA	31
5	44	RENALA	31
6	45	ETIOLOGIE GENETICA	32
7	46	IDIOPATICA	32

**INSERT INTO DOCTOR**  
VALUES(102,'GHEORGHE', 'DAN','0738-435-345','dangh1980@yahoo.com');

**INSERT INTO DOCTOR**  
VALUES(103,'IROD', 'ANA','0723-345-435','annairod@yahoo.com');

**INSERT INTO DOCTOR**  
VALUES(104,'TELEMAN', 'DENIS','0790-453-425','denisstl@gmail.com');

**INSERT INTO DOCTOR**  
VALUES(105,'ISTRATE', 'MARIA','0744-567-423','mariaistrate@gmail.com');

**INSERT INTO DOCTOR**  
VALUES(106,'NITA', 'MINODORA','0749-987-476','nitaminodora@gmail.com');

**INSERT INTO DOCTOR**  
VALUES(107,'ROGOCIU', 'TUDOR','0790-889-283','tudorrogociu@gmail.com');

**INSERT INTO DOCTOR**  
VALUES(108,'MARTIN', 'IRIS','0783-737-943','martiniris@gmail.com');

**INSERT INTO DOCTOR**  
VALUES(109,'DIACONESCU', 'MIHAELA','0732-539-865','diaconescumihaela@gmail.com');

**INSERT INTO DOCTOR**  
VALUES(110,'GHEORGHE', 'ISIS','0787-892-238','irisgh@gmail.com');

**INSERT INTO DOCTOR**  
VALUES(111,'IROD', 'ANORIS','0729-982-927','anorisirod@gmail.com');

**INSERT INTO DOCTOR**  
VALUES(112,'TELEMAN', 'DAMIAN','0722-232-343','damianteleman@gmail.com');

**INSERT INTO DOCTOR**

```
VALUES(113,'ISTRATE', 'MARIAN','0722-483-293','istratemarian@gmail.com');
```

The screenshot shows the Oracle SQL Developer interface. In the left sidebar, there's a tree view of 'Project\_Continueare' containing various tables like ANALIZE, CENTRU\_DONAT, CONTRACT, DEPARTAMENT, DESFASURATOR, DIAGNOSTIC, DOCTOR, ETIOLOGIE, EVOLUITE\_PACI, and FISA\_MEDICALA. The 'doctor' table is selected. The 'Worksheet' tab contains the following SQL code:

```

VALUES(110,'GHEORGHE', 'ISIS','0787-892-238','irisgh@gmail.com');

INSERT INTO DOCTOR
VALUES(111,'IROD', 'ANORIS','0729-982-927','anorisirod@gmail.com');

INSERT INTO DOCTOR
VALUES(112,'TELEMAN', 'DAMIAN','0722-232-343','damianteleman@gmail.com');

INSERT INTO DOCTOR
VALUES(113,'ISTRATE', 'MARIAN','0722-483-293','istratemarian@gmail.com');

SELECT * FROM DOCTOR;

```

The 'Results' tab displays the output of the last SELECT statement, showing 13 rows of data:

	COD_DOCTOR	NUME	PRENUME	TELEFON	MAIL
1	100	MARTIN	ILEANA	0767-263-342	martinileana@yahoo.com
2	101	DIACONESCU MIHAI		0732-453-535	diaconescumihai@gmail.com
3	102	GHEORGHE	DAN	0738-435-345	danh1980@yahoo.com
4	103	IROD	ANA	0723-345-435	annairod@yahoo.com
5	104	TELEMAN	DENIS	0790-453-425	denisst1@gmail.com
6	105	ISTRATE	MARIA	0744-567-423	mariistrate@gmail.com
7	106	NITA	MINODORA	0749-987-476	nitaminodora@gmail.com
8	107	ROGOCIU	TUDOR	0790-889-283	tudorrogociu@gmail.com
9	108	MARTIN	IRIS	0783-737-943	martiniris@gmail.com
10	109	DIACONESCU MIHAELA		0732-539-865	diaconescumihela@gmail.com
11	110	GHEORGHE	ISIS	0787-892-238	irisgh@gmail.com
12	111	IROD	ANORIS	0729-982-927	anorisirod@gmail.com
13	112	TELEMAN	DAMIAN	0722-232-343	damianteleman@gmail.com

### ---Tabel FISA\_MEDICALA --

```
CREATE TABLE FISA_MEDICALA(COD_PACIENT NUMBER(5) CONSTRAINT PK_REPARTIZARE_PACIENT REFERENCES
PACIENT(COD_PACIENT),
```

```
DATA_INTEMEIERII DATE CONSTRAINT DATA_INTEMEIRII_FIXA NOT NULL,
```

```
COD_ETIOLOGIE NUMBER(5) CONSTRAINT PK_REPARTIZARE_ETIOLOGIE REFERENCES
ETIOLOGIE(COD_ETIOLOGIE),
```

```
CONSTRAINT PK_REPARTIZARE PRIMARY KEY(COD_PACIENT, COD_ETIOLOGIE));
```

```
INSERT INTO FISA_MEDICALA
```

```
VALUES(51,TO_DATE('07-02-2007','DD-MM-YYYY'),40);
```

```
INSERT INTO FISA_MEDICALA
```

```
VALUES(52,TO_DATE('10-10-2009','DD-MM-YYYY'),41);
```

```
INSERT INTO FISA_MEDICALA
```

```
VALUES(53,TO_DATE('01-06-2015','DD-MM-YYYY'),40);
```

```
INSERT INTO FISA_MEDICALA  
VALUES(54,TO_DATE('11-11-2011','DD-MM-YYYY'),43);
```

```
INSERT INTO FISA_MEDICALA  
VALUES(55,TO_DATE('12-01-2013','DD-MM-YYYY'),45);
```

```
INSERT INTO FISA_MEDICALA  
VALUES(56,TO_DATE('17-11-2010','DD-MM-YYYY'),45);
```

```
INSERT INTO FISA_MEDICALA  
VALUES(57,TO_DATE('19-04-2006','DD-MM-YYYY'),42);
```

```
INSERT INTO FISA_MEDICALA  
VALUES(58,TO_DATE('22-08-2002','DD-MM-YYYY'),43);
```

```
INSERT INTO FISA_MEDICALA  
VALUES(59,TO_DATE('16-09-2009','DD-MM-YYYY'),44);
```

```
INSERT INTO FISA_MEDICALA  
VALUES(60,TO_DATE('07-12-2013','DD-MM-YYYY'),43);
```

```

--FISA MEDICALA--
CREATE TABLE FISA_MEDICALA(COD_PACIENT NUMBER(5) CONSTRAINT PK_REPARTIZARE_PACIENT REFERENCES PACIENT(COD_PACIENT),
                           DATA_INTEMERII DATE CONSTRAINT DATA_INTEMERII_FIXA NOT NULL,
                           COD_ETIOLOGIE NUMBER(5) CONSTRAINT PK_REPARTIZARE_ETIOLOGIE REFERENCES ETIOLOGIE(COD_ETIOLOGIE),
                           CONSTRAINT PK_REPARTIZARE PRIMARY KEY(COD_PACIENT, COD_ETIOLOGIE));

INSERT INTO FISA_MEDICALA
VALUES($1,TO_DATE('07-02-2007','DD-MM-YYYY'),40);

INSERT INTO FISA_MEDICALA
VALUES($2,TO_DATE('10-10-2009','DD-MM-YYYY'),41);

INSERT INTO FISA_MEDICALA
VALUES($3,TO_DATE('01-06-2015','DD-MM-YYYY'),40);

INSERT INTO FISA_MEDICALA
VALUES($4,TO_DATE('11-11-2011','DD-MM-YYYY'),43);

INSERT INTO FISA_MEDICALA
VALUES($5,TO_DATE('12-01-2013','DD-MM-YYYY'),45);

INSERT INTO FISA_MEDICALA
VALUES($6,TO_DATE('17-11-2010','DD-MM-YYYY'),45);

```

Query Result | All Rows Fetched: 9 in 0.006 seconds

	COD_PACIENT	DATA_INTEMERII	COD_ETIOLOGIE
1	51	07-FEB-07	40
2	52	10-OCT-09	41
3	53	01-JUN-15	40
4	54	11-NOV-11	43
5	55	12-JAN-13	45
6	57	19-APR-06	42
7	58	22-AUG-02	43

### ---Tabel CONTRACT---

```

CREATE TABLE CONTRACT(COD_SPITAL NUMBER(5) CONSTRAINT PK_CONTRACT_FACULTATE REFERENCES
                      SPITAL(COD_SPITAL),

                      DATA_START DATE CONSTRAINT DATA_START_FIXA NOT NULL,
                      COD_DOCTOR NUMBER(5) CONSTRAINT PK_CONTRACT_DOCTOR REFERENCES DOCTOR(COD_DOCTOR),
                      SALARIU NUMBER CONSTRAINT SALARIU_CONTRACT NOT NULL,
                      CONSTRAINT PK_CONTRACT PRIMARY KEY(COD_SPITAL, COD_DOCTOR));

```

INSERT INTO CONTRACT

```
VALUES (1,TO_DATE('12-12-1990','DD-MM-YYYY'),100,5000);
```

INSERT INTO CONTRACT

```
VALUES (2,TO_DATE('22-02-1989','DD-MM-YYYY'),101,6453);
```

INSERT INTO CONTRACT

```
VALUES (3,TO_DATE('29-01-1978','DD-MM-YYYY'),100,7387);
```

```
INSERT INTO CONTRACT  
VALUES (1,TO_DATE('12-12-1990','DD-MM-YYYY'),102,5000);
```

```
INSERT INTO CONTRACT  
VALUES (2,TO_DATE('22-02-1989','DD-MM-YYYY'),105,6453);
```

```
INSERT INTO CONTRACT  
VALUES (4,TO_DATE('29-01-1978','DD-MM-YYYY'),104,7387);
```

```
INSERT INTO CONTRACT  
VALUES (5,TO_DATE('12-12-1990','DD-MM-YYYY'),102,5000);
```

```
INSERT INTO CONTRACT  
VALUES (2,TO_DATE('22-02-1989','DD-MM-YYYY'),103,6453);
```

```
INSERT INTO CONTRACT  
VALUES (4,TO_DATE('29-01-1978','DD-MM-YYYY'),100,7387);
```

```
INSERT INTO CONTRACT  
VALUES (1,TO_DATE('12-12-1990','DD-MM-YYYY'),104,5000);
```

```
INSERT INTO CONTRACT  
VALUES (1,TO_DATE('12-12-1990','DD-MM-YYYY'),104,5000);
```

```
INSERT INTO CONTRACT  
VALUES (1,TO_DATE('10-12-1998','DD-MM-YYYY'),113,5000);
```

```
INSERT INTO CONTRACT  
VALUES (1,TO_DATE('19-09-1999','DD-MM-YYYY'),111,5000);
```

```
INSERT INTO CONTRACT  
VALUES (4,TO_DATE('18-01-2003','DD-MM-YYYY'),108,5700);
```

```
INSERT INTO CONTRACT  
VALUES (3,TO_DATE('13-03-1989','DD-MM-YYYY'),107,5300);
```

```
INSERT INTO CONTRACT
```

```
VALUES (1,TO_DATE('06-07-2001','DD-MM-YYYY'),109,5050);
```

```
INSERT INTO CONTRACT
```

```
VALUES (1,TO_DATE('12-02-2001','DD-MM-YYYY'),110,4000);
```

```
INSERT INTO CONTRACT
```

```
VALUES (1,TO_DATE('14-12-1990','DD-MM-YYYY'),106,7000);
```

```
INSERT INTO CONTRACT
```

```
VALUES (1,TO_DATE('18-01-1998','DD-MM-YYYY'),107,7000);
```

```
INSERT INTO CONTRACT
```

```
VALUES (2,TO_DATE('12-12-1990','DD-MM-YYYY'),111,6700);
```

```
INSERT INTO CONTRACT
```

```
VALUES (4,TO_DATE('23-11-1999','DD-MM-YYYY'),108,5000);
```

```
INSERT INTO CONTRACT
```

```
VALUES (5,TO_DATE('24-05-1991','DD-MM-YYYY'),111,5900);
```

```
INSERT INTO CONTRACT
```

```
VALUES (5,TO_DATE('27-10-1995','DD-MM-YYYY'),104,5000);
```

Oracle SQL Developer : Project\_Continuare

File Edit View Navigate Run Source Team Tools Window Help

Project.sql Project CONTRACT Dbms Output Project\_Continuare

CREATE TABLE

Worksheet Query Builder

```

VALUES (5,TO_DATE('24-05-1991','DD-MM-YYYY'),111,5900);

INSERT INTO CONTRACT
VALUES (5,TO_DATE('27-10-1995','DD-MM-YYYY'),104,5000);

SELECT * FROM CONTRACT;

-----SALON-----
CREATE TABLE SALON(COD_SALON NUMBER(5) CONSTRAINT PKEY_SALON PRIMARY KEY,
DENUMIRE VARCHAR(100) CONSTRAINT DENUMIRE_SALON NOT NULL,

```

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

All Rows Fetched: 21 in 0.004 seconds

COD_SPITAL	DATA_START	COD_DOCTOR	SALARIU
6	4 29-JAN-78	104	7387
7	5 12-DEC-90	102	5000
8	2 22-FEB-89	103	6453
9	4 29-JAN-78	100	7387
10	1 12-DEC-90	104	5000
11	1 10-DEC-98	113	5000
12	1 19-SEP-99	111	5000
13	4 18-JAN-03	108	5700
14	3 13-MAR-89	107	5300
15	1 06-JUL-01	109	5050
16	1 12-FEB-01	110	4000
17	1 14-DEC-90	106	7000
18	1 18-JAN-98	107	7000
19	2 12-DEC-90	111	6700
20	5 24-MAY-91	111	5900
21	5 27-OCT-95	104	5000

Messages - Log

12°C Cloudy Search

### --Tabel EVOLUTIE\_PACIENT--

```
CREATE TABLE EVOLUTIE_PACIENT(COD_EVOLUTIE NUMBER(5) CONSTRAINT PKEY_EVOLUTIE PRIMARY KEY,
DENUMIRE VARCHAR(100) CONSTRAINT DENUMIRE_STARE NOT NULL);
```

INSERT INTO EVOLUTIE\_PACIENT

```
VALUES(300,'BUNA');
```

```
INSERT INTO EVOLUTIE_PACIENT  
VALUES(301,'LA FEL');
```

```
INSERT INTO EVOLUTIE_PACIENT  
VALUES(302,'REA');
```

```
INSERT INTO EVOLUTIE_PACIENT  
VALUES(303,'DEFECTUOASA');
```

```
INSERT INTO EVOLUTIE_PACIENT  
VALUES(304,'PERFECTA');
```

### ---Tabel DIAGNOSTIC---

```
CREATE TABLE DIAGNOSTIC(PROCENT NUMBER(37,2) CONSTRAINT PROCENT_ANALIZA NOT NULL,  
COD_PACIENT NUMBER(5) CONSTRAINT PK_PACIENT REFERENCES PACIENT(COD_PACIENT),
```

The screenshot shows the Oracle SQL Developer interface. The top part displays the creation of the `EVOLUTIE_PACIENT` table and five corresponding INSERT statements. The bottom part shows the execution of a `SELECT * FROM EVOLUTIE_PACIENT;` query, which returns the following result set:

COD_EVOLUTIE	DENUMIRE
1	300 BUNA
2	301 LA FEL
3	302 REA
4	303 DEFECTUOASA
5	304 PERFECTA

```
COD_ANALIZE NUMBER(5) CONSTRAINT PK_ANALIZE REFERENCES ANALIZE(COD_ANALIZE),
CONSTRAINT PK_DIAGNOSTIC PRIMARY KEY(COD_PACIENT, COD_ANALIZE);
```

```
INSERT INTO DIAGNOSTIC
VALUES(90.1,51,400);
```

```
INSERT INTO DIAGNOSTIC
VALUES(80.2,52,401);
```

```
INSERT INTO DIAGNOSTIC
VALUES(60.9,52,402);
```

```
INSERT INTO DIAGNOSTIC
VALUES(45.9,54,403);
```

```
INSERT INTO DIAGNOSTIC
VALUES(70.78,55,404);
```

```
INSERT INTO DIAGNOSTIC
VALUES(82.76,56,405);
```

```
INSERT INTO DIAGNOSTIC
VALUES(93.60,404);
```

```
INSERT INTO DIAGNOSTIC
VALUES(54.98,52,403);
```

```
INSERT INTO DIAGNOSTIC
VALUES(88.55,401);
```

```
INSERT INTO DIAGNOSTIC
VALUES(39.5,60,405);
```

```
INSERT INTO DIAGNOSTIC
VALUES(32.55,53,402);
```

```
INSERT INTO DIAGNOSTIC
```

```
VALUES(59.9,59,402);
```

### ----Tabel DIAGNOSTIC---

The screenshot shows a SQL query builder interface with a toolbar at the top and a main workspace divided into two panes. The top pane is titled 'Worksheet' and contains the following SQL code:

```
--DIAGNOSTIC--  
CREATE TABLE DIAGNOSTIC(PROCENT NUMBER(37,2) CONSTRAINT PROCENT_ANALIZA NOT NULL,  
COD_PACIENT NUMBER(5) CONSTRAINT PK_PACIENT REFERENCES PACIENT(COD_PACIENT),  
COD_ANALIZE NUMBER(5) CONSTRAINT PK_ANALIZE REFERENCES ANALIZE(COD_ANALIZE),  
CONSTRAINT PK_DIAGNOSTIC PRIMARY KEY(COD_PACIENT, COD_ANALIZE));  
  
INSERT INTO DIAGNOSTIC  
VALUES(50.1,51,400);  
  
INSERT INTO DIAGNOSTIC  
VALUES(80.2,52,401);  
  
INSERT INTO DIAGNOSTIC  
VALUES(60.9,52,402);  
  
INSERT INTO DIAGNOSTIC  
VALUES(45.9,54,403);  
  
INSERT INTO DIAGNOSTIC  
VALUES(70.78,55,404);  
  
INSERT INTO DIAGNOSTIC
```

The bottom pane is titled 'Query Result' and displays a table with the following data:

	PROCENT	COD_PACIENT	COD_ANALIZE
1	90.1	51	400
2	80.2	52	401
3	60.9	52	402
4	45.9	54	403
5	70.78	55	404
6	82.76	56	405
7	93	60	404

### ---Tabel ANALIZE---

```
CREATE TABLE ANALIZE(COD_ANALIZE NUMBER(5) CONSTRAINT PKEY_ANALIZE PRIMARY KEY,
```

```
FORMA VARCHAR(100) CONSTRAINT FORMA_ANALIZE NOT NULL,
```

```
COD_EVOLUTIE NUMBER(5), CONSTRAINT FK_ANALIZE FOREIGN KEY(COD_EVOLUTIE) REFERENCES  
EVOLUTIE_PACIENT(COD_EVOLUTIE));
```

```
INSERT INTO ANALIZE
```

```
VALUES(400,'SANGE',300);
```

```
INSERT INTO ANALIZE
```

```
VALUES(401,'URINA',301);
```

```
INSERT INTO ANALIZE
```

```
VALUES(402,'SALIVA',300);
```

```
INSERT INTO ANALIZE
```

```
VALUES(403,'SANGE',302);
```

```
INSERT INTO ANALIZE
```

```
VALUES(404,'URINA',304);
```

```
INSERT INTO ANALIZE
```

```
VALUES(405,'SALIVA',300);
```

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Team, Tools, Window, and Help. The toolbar has icons for New, Open, Save, Print, Run, Stop, Refresh, Undo, Redo, Copy, Paste, Find, Replace, and Insert. The Project tab is selected. The left pane shows a tree view of the project with 'Welcome Page' and 'Project'. The right pane has two tabs: 'Query Builder' and 'Query Result'. The 'Query Builder' tab contains the following code:

```
SELECT * FROM CONTRACT;
-----SALON-----
CREATE TABLE SALON(COD_SALON NUMBER(5) CONSTRAINT PKKEY_SALON PRIMARY KEY,
DENUMIRE VARCHAR(100) CONSTRAINT DENUMIRE_SALON NOT NULL,
LOCATIE VARCHAR(100) CONSTRAINT LOCATIE_SALON NOT NULL);

INSERT INTO SALON
VALUES (200,'212','ETAJ 2');

INSERT INTO SALON
VALUES (201,'172','ETAJ 1');

INSERT INTO SALON
VALUES (202,'021','PATER');

INSERT INTO SALON
VALUES (203,'567','ETAJ 5');

INSERT INTO SALON
VALUES (204,'722','ETAJ 7');

INSERT INTO SALON
VALUES (205,'281-B','ETAJ 2, ARIPA B');
```

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

COD_SALON	DENUMIRE	LOCATIE
1	200 212	ETAJ 2
2	201 172	ETAJ 1
3	202 021	PATER
4	203 567	ETAJ 5
5	204 722	ETAJ 7
6	205 281-B	ETAJ 2, ARIPA B
7	206 562-A	ETAJ 5, ARIPA A

The second tab in the 'Query Result' section contains the following DML statements:

```
INSERT INTO ANALIZE
VALUES(402,'SALIVA',300);

INSERT INTO ANALIZE
VALUES(403,'SANGE',302);

INSERT INTO ANALIZE
VALUES(404,'URINA',304);

INSERT INTO ANALIZE
VALUES(405,'SALIVA',300);

SELECT * FROM ANALIZE;
```

The third tab in the 'Query Result' section shows the results of the last query:

COD_ANALIZE	FORMA	COD_EVOLUTIE
1	400 SANGE	300
2	401 URINA	301
3	402 SALIVA	300
4	403 SANGE	302
5	405 SALIVA	300
6	404 URINA	304

At the bottom of the interface, status bars indicate 'Line 364 Column 23' and 'Modified Windows: 0'.

### **---Tabel SALON---**

```
CREATE TABLE SALON(COD_SALON NUMBER(5) CONSTRAINT PKEY_SALON PRIMARY KEY,  
DENUMIRE VARCHAR(100) CONSTRAINT DENUMIRE_SALON NOT NULL,  
LOCATIE VARCHAR(100) CONSTRAINT LOCATIE_SALON NOT NULL);
```

```
INSERT INTO SALON
```

```
VALUES (200,'212','ETAJ 2');
```

```
INSERT INTO SALON
```

```
VALUES (201,'172','ETAJ 1');
```

```
INSERT INTO SALON
```

```
VALUES (202,'021','PATER');
```

```
INSERT INTO SALON
```

```
VALUES (203,'567','ETAJ 5');
```

```
INSERT INTO SALON
```

```
VALUES (204,'722','ETAJ 7');
```

```
INSERT INTO SALON
```

```
VALUES (205,'281-B','ETAJ 2, ARIPA B');
```

```
INSERT INTO SALON
```

```
VALUES (206,'562-A','ETAJ 5, ARIPA A');
```

### **---Tabelul DESFASURATOR---**

```
CREATE TABLE DESFASURATOR(COD_ETIOLOGIE NUMBER(5) CONSTRAINT PK_ETI_DESFASURATOR REFERENCES  
ETIOLOGIE(COD_ETIOLOGIE),  
COD_EVOLUTIE NUMBER(5) CONSTRAINT PK_EV_DESFASURATOR REFERENCES  
EVOLUTIE_PACIENT(COD_EVOLUTIE),  
COD_DOCTOR NUMBER(5) CONSTRAINT PK_DOC_DESFASURATOR REFERENCES DOCTOR(COD_DOCTOR),  
COD_SALON NUMBER(5) CONSTRAINT PK_SALON_DESFASURATOR REFERENCES SALON(COD_SALON),  
CONSTRAINT PK_DESFASURATOR PRIMARY KEY(COD_ETIOLOGIE, COD_EVOLUTIE, COD_DOCTOR, COD_SALON));
```

```
INSERT INTO DESFASURATOR
```

```
VALUES(40,300,101,203);
```

```
INSERT INTO DESFASURATOR
```

```
VALUES (41,302,103,200);
```

```
INSERT INTO DESFASURATOR
```

```
VALUES (42,305,105,205);
```

```
INSERT INTO DESFASURATOR
```

```
VALUES (43,304,104,204);
```

```
INSERT INTO DESFASURATOR
```

```
VALUES (44,302,104,203);
```

```
INSERT INTO DESFASURATOR
```

```
VALUES (45,304,103,204);
```

```
INSERT INTO DESFASURATOR
```

```
VALUES (46,303,104,203);
```

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes 'File', 'Team', 'Tools', 'Window', and 'Help'. Below the menu is a toolbar with various icons. The main area has tabs for 'Worksheet' and 'Query Builder', with 'Worksheet' currently selected. The worksheet contains the following SQL code:

```

-----DESFASURATOR-----
CREATE TABLE DESFASURATOR(COD_ETIOLOGIE NUMBER(5) CONSTRAINT PK_ETI_DESFASURATOR REFERENCES ETIOLOGIE(COD_ETIOLOGIE),
                           COD_EVOLUTIE NUMBER(5) CONSTRAINT PK_EV_DESFASURATOR REFERENCES EVOLUTIE_PACIENT(COD_EVOLUTIE),
                           COD_DOCTOR NUMBER(5) CONSTRAINT PK_DOC_DESFASURATOR REFERENCES DOCTOR(COD_DOCTOR),
                           COD_SALON NUMBER(5) CONSTRAINT PK_SALON_DESFASURATOR REFERENCES SALON(COD_SALON),
                           CONSTRAINT PK_DESFASURATOR PRIMARY KEY(COD_ETIOLOGIE, COD_EVOLUTIE, COD_DOCTOR, COD_SALON));

INSERT INTO DESFASURATOR
VALUES (40,300,101,201);

INSERT INTO DESFASURATOR
VALUES (41,302,103,200);

INSERT INTO DESFASURATOR
VALUES (42,305,105,205);

INSERT INTO DESFASURATOR
VALUES (43,304,104,204);

INSERT INTO DESFASURATOR
VALUES (44,302,104,203);

INSERT INTO DESFASURATOR

```

The 'Query Result' tab is open below the worksheet, showing the results of the last six INSERT statements. The results are as follows:

	COD_ETIOLOGIE	COD_EVOLUTIE	COD_DOCTOR	COD_SALON
1	40	300	101	201
2	41	302	103	200
3	43	304	104	204
4	44	302	104	203
5	45	304	103	204
6	46	303	104	203

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

--pentru un spital specificat, pentru fiecare salon al acestuia,

--afisam pacientii din cadul etiologilor care sunt

--gazduite de catre salonul curent sau(daca nu) "NU EXISTA!"

```
CREATE OR REPLACE PROCEDURE Exercitiul6(numespital SPITAL.denumire%TYPE)
```

```
AS
```

```
    TOTAL NUMBER(6);
```

```
    TYPE TABLOU_INDEXAT1 IS TABLE OF SALON%ROWTYPE INDEX BY PLS_INTEGER;
```

```
    SALOANE TABLOU_INDEXAT1;
```

```
    TYPE TABLOU_INDEXAT2 IS TABLE OF VARCHAR(200) INDEX BY PLS_INTEGER;
```

```
    NUME_PACIENT TABLOU_INDEXAT2;
```

```
    TYPE TABLOU_IMBRICAT IS TABLE OF ETIOLOGIE%ROWTYPE;
```

```
    ETIOLOGII TABLOU_IMBRICAT := TABLOU_IMBRICAT();
```

```
BEGIN
```

```
    SELECT COUNT(*)
```

```
        INTO TOTAL
```

```
        FROM SPITAL SP, DEPARTAMENT D, SECTIE S, ETIOLOGIE E
```

```
        WHERE SP.COD_SPITAL = D.COD_SPITAL AND D.COD_DEPARTAMENT = S.COD_DEPARTAMENT AND S.COD_SECTIE  
        = E.COD_SECTIE AND UPPER(SP.denumire) LIKE UPPER(numespital);
```

```
    ETIOLOGII.extend(TOTAL + 1);
```

```
    SELECT *
```

```
        BULK COLLECT INTO SALOANE
```

```
        FROM SALON;
```

```
    SELECT E.COD_ETIOLOGIE, E.DENUMIRE, E.COD_SECTIE
```

```
        BULK COLLECT INTO ETIOLOGII
```

```
        FROM SPITAL SP, DEPARTAMENT D, SECTIE S, ETIOLOGIE E
```

```

        WHERE SP.COD_SPITAL = D.COD_SPITAL AND D.COD_DEPARTAMENT = S.COD_DEPARTAMENT AND
        S.COD_SECTIE = E.COD_SECTIE AND UPPER(SP.denumire) LIKE UPPER(numespital);

FOR i IN SALOANE.first..SALOANE.last LOOP
    DBMS_OUTPUT.PUT_LINE('SALONUL NUMARUL: ' || SALOANE(i).denumire);
    DBMS_OUTPUT.PUT_LINE('-----');
    FOR j IN ETIOLOGII.first..ETIOLOGII.last LOOP
        DBMS_OUTPUT.PUT_LINE('ETIOLOGIA ' || ETIOLOGII(j).denumire);
        DBMS_OUTPUT.PUT_LINE('-----');
        SELECT P.NUME || ' ' || P.PRENUME
        BULK COLLECT INTO NUME_PACIENT
        FROM PACIENT P, FISA_MEDICALA FM, DESFASURATOR DESF
        WHERE P.COD_PACIENT = FM.COD_PACIENT AND FM.COD_ETIOLOGIE = ETIOLOGII(j).COD_ETIOLOGIE AND
        DESF.COD_ETIOLOGIE = ETIOLOGII(j).COD_ETIOLOGIE and SALOANE(i).COD_SALON = DESF.COD_SALON;

        IF NUME_PACIENT.count > 0 THEN
            TOTAL := 0;
            FOR k in NUME_PACIENT.first..NUME_PACIENT.last LOOP
                TOTAL := TOTAL + 1;
                DBMS_OUTPUT.PUT_LINE(TOTAL || '.' || NUME_PACIENT(k));
            END LOOP;
        END IF;

        IF NUME_PACIENT.count = 0
            THEN DBMS_OUTPUT.PUT_LINE('NU EXISTA!');
        END IF;
        DBMS_OUTPUT.NEW_LINE;
        DBMS_OUTPUT.NEW_LINE;
        DBMS_OUTPUT.NEW_LINE;
    END LOOP;
END LOOP;
END;
/
BEGIN
Exercitiul6('Spitalul Universitar de Urgente Bucuresti');
END;

```

```

--6
--pentru un spital specificat, pentru fiecare salon al acestuia,
--afisam pacientii din cadoul etiologilor care sunt
--gasite de catre salonul curent sau(daca nu "NU EXISTA")
CREATE OR REPLACE PROCEDURE Exercitiul6(numeSpital SPITAL.denumire%TYPE)
AS
    TYPE TABLOU_INDEXAT1 IS TABLE OF SALON%ROWTYPE INDEX BY PLS_INTEGER;
    SALOANE TABLOU_INDEXAT1;

    TYPE TABLOU_IMBRICAT IS TABLE OF ETIOLOGIE%ROWTYPE;
    ETIOLOGII TABLOU_IMBRICAT := TABLOU_IMBRICAT();

    TYPE TABLOU_INDEXAT2 IS TABLE OF VARCHAR(200) INDEX BY PLS_INTEGER;
    NUME_PACIENT TABLOU_INDEXAT2;

    TOTAL NUMBER(6);

BEGIN
    SELECT *
      BULK COLLECT INTO SALOANE
        FROM SALON;

    SELECT COUNT(*)
  
```

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

**--afisați pacientii care au analizele de sange bune în procentaj de peste 45%**

```
CREATE OR REPLACE PROCEDURE Exercitiul7(tip_analize analize.forma%TYPE, procent_analize diagnostic.procent%TYPE)
```

AS

CURSOR analizee IS

```

SELECT cod_analize, cod_evolutie
  FROM analize
 WHERE UPPER(forma) LIKE UPPER(tip_analize);

```

CURSOR evolutii (cod\_evolutie.cod\_evolutie%TYPE) IS

```

SELECT denumire
  FROM evolutie_pacient
 WHERE cod_evolutie = cod;

```

```
CURSOR pacienti (cod analize.cod_analize%TYPE ) IS
  SELECT p.nume || '' || p.prenume || ' are procentul ' || d.procent as result
  FROM diagnostic d, pacient p
  WHERE d.cod_pacient = p.cod_pacient AND d.cod_analize = cod AND d.procent >= procent_analize;
```

```
denumire_stare evolutie_pacient.denumire%TYPE;
cod evolutie_pacient.cod_evolutie%TYPE;
```

```
BEGIN
```

```
FOR analizee IN analize LOOP
```

```
  OPEN evolutii(analize.cod_evolutie);
  FETCH evolutii INTO denumire_stare;
  DBMS_OUTPUT.PUT_LINE('Starea pacientului: ' || denumire_stare);
  DBMS_OUTPUT.PUT_LINE('-----');
  CLOSE evolutii;
```

```
FOR pacient IN pacienti(analize.cod_analize) LOOP
```

```
  DBMS_OUTPUT.PUT_LINE(pacient.result);
END LOOP;
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.NEW_LINE;
DBMS_OUTPUT.NEW_LINE;
END LOOP;
```

```
END;
```

```
/
```

```
BEGIN
```

```
Exercitiul7('SANGE', 40.0);
```

```
END;
```

```
/
```

```

-- Oracle SQL Developer Screenshot Description

The screenshot shows the Oracle SQL Developer interface. 
- **File Bar:** File, Edit, View, Navigate, Run, Source, Team, Tools, Window, Help.
- **Toolbars:** Standard toolbar with icons for New, Open, Save, Print, etc.
- **Panels:** 
  - **Dbsm Output:** Shows output from a previous run, including patient status like 'Starea pacientului: BUNA' and 'POPESCU IRINA are procentul 90.1'.
  - **Worksheet:** Displays the PL/SQL code for the 'analize' procedure. The code uses DBMS_OUTPUT.PUT_LINE to print patient names and their percentages. It includes loops to handle multiple patients and a final call to Exercitiul7.
  - **Script Output:** Shows the message 'PL/SQL procedure successfully completed.' and a note about task completion time.
- **Status Bar:** Shows the line number (625), column number (5), and other system information like battery level and signal strength.

```

**8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele definite. Definiți minim 2 excepții. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.**

**Rezolvare:**

--pentru un doctor al carui nume de familie apare de cel putin doua ori,  
--determinati la cate spitale este angajat

**CREATE OR REPLACE FUNCTION Exercitiul8(COD DOCTOR.COD\_DOCTOR%TYPE) RETURN NUMBER**

**IS**

```

TYPE TABLOU_INDEXAT IS TABLE OF CONTRACT%ROWTYPE INDEX BY PLS_INTEGER;
CONTRACTE TABLOU_INDEXAT;

```

```
NUME_DOCTOR DOCTOR.NUME%TYPE;
```

```
NUMAR_SPITALE NUMBER;
```

```

NO_DATA_FOUND_DOCTOR EXCEPTION;
NO_DATA_FOUND_SPITAL EXCEPTION;
COD_INVALID EXCEPTION;

BEGIN
  IF COD < 0 THEN
    RAISE COD_INVALID;
  END IF;

  SELECT *
  BULK COLLECT INTO CONTRACTE
  FROM CONTRACT
  WHERE COD_DOCTOR = COD;

  IF SQL%NOTFOUND THEN
    RAISE NO_DATA_FOUND_DOCTOR;
  END IF;

  SELECT NUME
  INTO NUME_DOCTOR
  FROM DOCTOR
  WHERE COD_DOCTOR = COD;

  SELECT COUNT(S.COD_SPITAL)
  INTO NUMAR_SPITALE
  FROM SPITAL S
  JOIN CONTRACT C ON (C.COD_SPITAL = S.COD_SPITAL)
  WHERE C.COD_DOCTOR = COD
  AND (SELECT COUNT(NUME)FROM DOCTOR WHERE UPPER(NUME) LIKE
  UPPER(NUME_DOCTOR)) >= 2;

  IF NUMAR_SPITALE = 0 THEN
    RAISE NO_DATA_FOUND_SPITAL;

```

```

ELSE RETURN NUMAR_SPITALE;
END IF;

EXCEPTION
    WHEN COD_INVALID THEN
        DBMS_OUTPUT.PUT_LINE('CODUL DOCTORULUI TREBUIE SA FIE >=1');
        RETURN -1;
    WHEN NO_DATA_FOUND_DOCTOR THEN
        DBMS_OUTPUT.PUT_LINE('IN TABELUL CONTRACT NU EXISTA NICIUN DOCTOR CU
CODUL: ' || cod);
        RETURN -1;
    WHEN NO_DATA_FOUND_SPITAL THEN
        DBMS_OUTPUT.PUT_LINE('NU EXISTA SPITALE IN TABEL CARE SA INDEPLINEASCA
ACESTE CONDITII!');
        RETURN -1;
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('MESAJ EROARE: ' || SQLERRM);
        DBMS_OUTPUT.PUT_LINE('COD EROARE: ' || SQLCODE);
        RETURN -1;
END;
/

DECLARE
    COD DOCTOR.COD_DOCTOR%TYPE := &COD;
    TOTAL NUMBER;

BEGIN
    TOTAL := Exercitiul8(COD);
    IF TOTAL >=0 THEN
        DBMS_OUTPUT.PUT_LINE('PENTRU PROFESORUL CU CODUL: ' || COD ||' AVEM UN TOTAL
DE: '|| TOTAL );
    END IF;
END;
/

```

**Cod spitale:**

-100

-112

-107

--1

```
PENTRU PROFESORUL CU CODUL: 100 AVEM UN TOTAL DE: 3
IN TABELUL CONTRACT NU EXISTA NICIUN DOCTOR CU CODUL: 112
NU EXISTA SPITALE IN TABEL CARE SA INDEPLINEASCA ACESTE CONDITII!
CODUL DOCTORULUI TREBUIE SA FIE >=1
```

**9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să utilizeze într-o singură comandă SQL 5 dintre tabelele definite. Tratați toate exceptiile care pot apărea, incluzând exceptiile NO\_DATA\_FOUND și TOO\_MANY\_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.**

**Rezolvare:**

--PENTRU UN PACIENT CU UN PRENUME DAT, AFISEZ EVOLUTIILE

--ACESTUIA TINAND CONT DOAR DE ACELEA CARE S-AU STABILIT

--IN SALOANE CARE AU CIFRA '5' IN COMPONENTA

```

CREATE OR REPLACE PROCEDURE Exercitiul9(PRENUME_PACIENT
PACIENT.PRENUME%TYPE)

AS

    TYPE TABLOU_INDEX IS TABLE OF EVOLUTIE_PACIENT.DENUMIRE%TYPE INDEX BY
PLS_INTEGER;

    STAREA_EVOLUTIEI TABLOU_INDEX;

    TYPE TABLOU_INDEX_PACIENTI IS TABLE OF PACIENT%ROWTYPE INDEX BY
PLS_INTEGER;

    PACIENTI TABLOU_INDEX_PACIENTI;

    NO_DATA_FOUND_PRENUME EXCEPTION;
    NO_DATA_FOUND_EVOLUTIE EXCEPTION;
    TO_MANY_ROWS_NEW EXCEPTION;

BEGIN

    SELECT *
    BULK COLLECT INTO PACIENTI
    FROM PACIENT
    WHERE UPPER(PRENUME) = UPPER(PRENUME_PACIENT);

    IF SQL%NOTFOUND THEN
        RAISE NO_DATA_FOUND_PRENUME;
    END IF;

    IF PACIENTI.COUNT >= 2 THEN
        RAISE TO_MANY_ROWS_NEW;
    END IF;

    SELECT EV.DENUMIRE
    BULK COLLECT INTO STAREA_EVOLUTIEI
    FROM EVOLUTIE_PACIENT EV
    JOIN DESFASATOR D ON (D.COD_EVOLUTIE = EV.COD_EVOLUTIE)
    JOIN SALON S ON (D.COD_SALON = S.COD_SALON)

```

```

JOIN ETIOLOGIE ET ON (D.COD_ETIOLOGIE = ET.COD_ETIOLOGIE)
JOIN FISA_MEDICALA FM ON (FM.COD_ETIOLOGIE = ET.COD_ETIOLOGIE)
JOIN PACIENT P ON (P.COD_PACIENT = FM.COD_PACIENT)
WHERE UPPER(P.PRENUME) = UPPER(PRENUME_PACIENT)
AND S.DENUMIRE LIKE '%5%';

IF STAREA_EVOLUTIEI.COUNT = 0 THEN
    RAISE NO_DATA_FOUND_EVOLUTIE;
END IF;

DBMS_OUTPUT.PUT_LINE('EVOLUTIA PACIENTULUI CU NUMELE ' || PRENUME_PACIENT || '
ESTE:');
FOR i IN STAREA_EVOLUTIEI.FIRST..STAREA_EVOLUTIEI.LAST LOOP
    DBMS_OUTPUT.PUT_LINE(STAREA_EVOLUTIEI(i));
END LOOP;

EXCEPTION
WHEN TO_MANY_ROWS_NEW THEN
    DBMS_OUTPUT.PUT_LINE('EXISTA MAI MULTI PACIENTI CU ACEL PRENUME');

WHEN NO_DATA_FOUND_PRENUME THEN
    DBMS_OUTPUT.PUT_LINE('NU EXISTA STUDENTI CU ACEST PRENUME!');

WHEN NO_DATA_FOUND_EVOLUTIE THEN
    DBMS_OUTPUT.PUT_LINE('NU EXISTA SPITALE IN TABEL CARE SA INDEPLINEASCA
ACESTE CONDITII!');

WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('MESAJ EROARE: ' || SQLERRM);
    DBMS_OUTPUT.PUT_LINE('COD EROARE: ' || SQLCODE);

END;

```

/

**DECLARE**

```
prenume PACIENT.PRENUME%TYPE := '&prenume';
```

**BEGIN**

```
    Exercitiu9(prenume);
```

**END;**

/

--Nume introduse:

--Irina

--Dan

--Francisca

--Mihai

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Team, Tools, Window, and Help. The toolbar contains various icons for file operations. The left sidebar features a 'Connections' tree view with several database objects like DBMS\_STATS\_OBJ\_LIST\_TAB, DDL\_REQUESTS, etc., and a 'Reports' section with options like All Reports, Analytic View Reports, Data Dictionary Reports, etc. The main workspace displays the output of a PL/SQL script. The output window title is 'Project.sq' and it shows the following text:

```
EVOLUTIA PACIENTULUI CU NUMELE: IRINA  
BUNA  
  
EVOLUTIA PACIENTULUI CU NUMELE IRINA ESTE:  
BUNA  
  
EXISTA MAI MULTI PACIENTI CU ACEL PRENUME  
  
NU EXISTA STUDENTI CU ACEST PRENUME!  
  
NU EXISTA SPITALE IN TABEL CARE SA INDEPLINEASCA ACESTE CONDITII!
```

At the bottom, there's a 'Messages - Log' panel with tabs for Messages, Logging Page, Statements, and Compiler. The system tray at the bottom shows the weather as 13°C and mostly sunny, along with other system icons.

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

**Rezolvare:**

--trigger-ul se va declasa atunci cand vreau sa modific  
--(in cazul meu sa introduc o noua valoare) tabela pacient deoarece  
--nu pot modifica tabela pacient in afara programului de lucru sau zile libere

```
CREATE OR REPLACE TRIGGER Exercitiul10
BEFORE INSERT OR UPDATE OR DELETE ON PACIENT
DECLARE
BEGIN
IF(
    TO_CHAR(sysdate, 'HH24') NOT BETWEEN 8 AND 20
    OR
    TO_CHAR(sysdate, 'DD-MM') = '25-12'
    OR
    TO_CHAR(sysdate, 'DD-MM') = '01-01'
    THEN RAISE_APPLICATION_ERROR(-20001, 'Operatiile asupra tabelului sunt permise doar in timpul programului de lucru si
in zilele lucratoare!');
END IF;
END;
/
INSERT INTO PACIENT
VALUES(64,'POP', 'EMA', TO_DATE('14-09-1995','DD-MM-YYYY'),'F','0724-343-354','emapop@yahoo.com',10);
```

Oracle SQL Developer : Project\_Continuare

File Edit View Navigate Run Source Team Tools Window Help

Connections Project\_Continuare Project.sgl Project PACIENT Dbms Output Project\_Continuare

Tables (Filtered)

- ANALIZE
- CENTRU\_DONARE
- CONTRACT
- DEPARTAMENT
- DESFSRATOR
- DIAGNOSTIC
- DOCTOR
- ETIOLOGIE
- EVOLUTIE\_PACIENT
- FISA\_MEDICALA
- PACIENT
  - COD\_PACIENT
  - NUME
  - PRENUME
  - DATA\_NASTERII
  - SEX

Reports All Reports Analytic View Reports Data Dictionary Reports Data Modeler Reports OLAP Reports TimesTen Reports User Defined Reports

Worksheet Query Builder

```
--10
--trigger-ul se va declasa atunci cand vrea sa modifica tabela pacient deoarece
-- nu pot modifica tabela pacient in afara programului de lucru sau zile libere
CREATE OR REPLACE TRIGGER Exercitiul10
BEFORE INSERT OR UPDATE OR DELETE ON PACIENT
DECLARE
BEGIN
IF(
    TO_CHAR(sysdate, 'HH24') BETWEEN 8 AND 20
    OR
    TO_CHAR(sysdate, 'DD-MM') = '25-12'
    OR
    TO_CHAR(sysdate, 'DD-MM') = '01-01')
    THEN RAISE_APPLICATION_ERROR(-20001, 'Operatiile asupra tabelului sunt permise doar in timpul programului de lucru si in zilele lucratoare!');
END IF;
END;
/

```

Script Output X Query Result X Task completed in 0.12 seconds

Trigger EXERCITIUL10 compiled

```
Error starting at line : 528 in command -
INSERT INTO PACIENT
VALUES(64,'POP', 'EMA', TO_DATE('14-09-1995','DD-MM-YYYY'),'F','0724-343-354','emapop@yahoo.com',10)
Error report -
ORA-20001: Operatiile asupra tabelului sunt permise doar in timpul programului de lucru si in zilele lucratoare!
ORA-06512: at "C##FRANCISCAPASARE.EXERCITIUL10", line 9
ORA-04088: error during execution of trigger 'C##FRANCISCAPASARE.EXERCITIUL10'
```

Messages - Log

Saved: Table C##FRANCISCAPASARE.PACIENT @Project\_Continuare

OneDrive Screenshot saved The screenshot was added to your OneDrive.

15°C Mostly cloudy

Q Search

4:20 1/3/2024

--daca stergem NOT-ul din conditie

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the 'Connections' and 'Tables (Filtered)' sections, which include tables like ANALIZE, CENTRU\_DONARE, CONTRACT, DEPARTAMENT, DESFASURATOR, DIAGNOSTIC, DOCTOR, ETIOLOGIE, EVOLUTIE\_PACIENT, FISA\_MEDICALA, and PACIENT. The main workspace contains a SQL script for creating a trigger named 'Exercitiul10'. The script checks if the date of birth is between 1970 and 2019. If it's not, it raises an application error. It also inserts a new row into the PACIENT table with values: ID 64, first name 'POF', last name 'EMA', birth date '14-09-1995', gender 'F', phone '0724-343-354', email 'emapop@yahoo.com', and age 10. The 'Script Output' tab shows the execution results, including an ORA-04080 error message and a successful compilation of the trigger.

```

--10
--trigger-ul se va declasa atunci cand vreau sa modific tabela pacient deoarece
-- nu pot modifica tabela pacient in afara programului de lucru sau zile libere
CREATE OR REPLACE TRIGGER Exercitiul10
BEFORE INSERT OR UPDATE OR DELETE ON PACIENT
DECLARE
BEGIN
IF(
    TO_CHAR(sysdate, 'HH24') NOT BETWEEN 8 AND 20
    OR
    TO_CHAR(sysdate, 'DD-MM') = '25-12'
    OR
    TO_CHAR(sysdate, 'DD-MM') = '01-01')
        THEN RAISE_APPLICATION_ERROR(-20001, 'Operatiile asupra tabelului sunt permise doar in timpul programului de lucru si in zilele lucratoare!');
END IF;
/
INSERT INTO PACIENT
VALUES(64,'POF','EMA',TO_DATE('14-09-1995','DD-MM-YYYY'),'F','0724-343-354','emapop@yahoo.com',10);
SELECT * FROM PACIENT

```

## 11. Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.

**Rezolvare:**

--trigger-ul se va declasa atunci cand vreau sa modific tabela contract deoarece

--nu pot sa am un doctor angajat mai devreme 1970 sau

--care are un salariu mai mare de 10000 lei

```

CREATE OR REPLACE TRIGGER Exercitiul11
BEFORE INSERT OR UPDATE OR DELETE ON CONTRACT
FOR EACH ROW
DECLARE
BEGIN
IF TO_CHAR(:NEW.DATA_START, 'YYYY') < '1970'
    THEN RAISE_APPLICATION_ERROR(-20001, 'Un doctor nu poate sa fie angajat mai devreme de anul 1970!');
END IF;
IF :NEW.SALARIU >10000
    THEN RAISE_APPLICATION_ERROR(-20001, 'Un doctor nu poate avea salariu mai mare de 10000 lei!');

```

```

END IF;

END;

/

INSERT INTO CONTRACT
VALUES (4,TO_DATE('02-01-1969','DD-MM-YYYY'),107,8976);

```

```

INSERT INTO CONTRACT
VALUES (1,TO_DATE('29-12-2000','DD-MM-YYYY'),105,10001);

```

```

INSERT INTO CONTRACT
VALUES (3,TO_DATE('27-08-2000','DD-MM-YYYY'),106,6000);

```

The screenshot shows the Oracle SQL Developer interface with a script editor window containing PL/SQL code. The code defines three INSERT statements into the CONTRACT table and includes a comment block at the end. Below the code, the 'Script Output' tab displays the execution results, including an error message about a trigger named 'EXERCITIUL11'. The 'Messages - Log' tab at the bottom shows the error details.

```

END IF;
END;
/
INSERT INTO CONTRACT
VALUES (4,TO_DATE('02-01-1969','DD-MM-YYYY'),107,8976);

INSERT INTO CONTRACT
VALUES (1,TO_DATE('29-12-2000','DD-MM-YYYY'),105,10001);

INSERT INTO CONTRACT
VALUES (3,TO_DATE('27-08-2000','DD-MM-YYYY'),106,6000);

--12
--trigger-ul se va declasa atunci la fiecare operatie ldd
--modificarile sunt salvate in tabela new_updates
CREATE TABLE new_updates(

```

```

ORA-0408: at "C#FRANCISCAPASARE.EXERCITIUL11", line 8
ORA-04088: error during execution of trigger 'C#FRANCISCAPASARE.EXERCITIUL11'

Error starting at line : 46 in command -
INSERT INTO CONTRACT
VALUES (1,TO_DATE('29-12-2000','DD-MM-YYYY'),105,10001)
Error report -
ORA-20001: Un doctor nu poate avea salariu mai mare de 10000 lei!
ORA-06512: at "C#FRANCISCAPASARE.EXERCITIUL11", line 7
ORA-04088: error during execution of trigger 'C#FRANCISCAPASARE.EXERCITIUL11'

1 row inserted.

```

## 12. Definiți un trigger de tip LDD. Declanșați trigger-ul.

--trigger-ul se va declasa atunci la fiecare operatie ldd

--modificarile sunt salvate in tabela new\_updates

`CREATE TABLE new_updates(`

```
    nume_bd VARCHAR2(50),
    user_logat VARCHAR2(50),
    eveniment VARCHAR2(50),
    tip_object_referit VARCHAR2(50),
    nume_object_referit VARCHAR2(50),
    data_realizarii DATE
);
```

```
CREATE OR REPLACE TRIGGER Exercitiul12
AFTER CREATE OR ALTER OR DROP ON SCHEMA
BEGIN
INSERT INTO new_updates
VALUES(SYS.DATABASE_NAME, SYS.LOGIN_USER, SYS.SYSEVENT,
SYS.DICTIONARY_OBJ_TYPE, SYS.DICTIONARY_OBJ_NAME, SYSDATE);
END;
/
```

```
CREATE TABLE HAINA(COD_HAINA NUMBER(37,2) CONSTRAINT PKEY_HAINA PRIMARY KEY,
DENUMIRE VARCHAR(50));
SELECT * FROM HAINA;
```

```
ALTER TABLE HAINA
ADD (ANOTIMP VARCHAR2(50));
```

```
ALTER TABLE HAINA
DROP COLUMN DENUMIRE;
```

```
DROP TABLE HAINA;
```

Oracle SQL Developer : Project\_Continare

File Edit View Navigate Run Source Team Tools Window Help

Project.sql Project CONTRACT Dbms Output Project\_Continare

CREATE TABLE

Worksheet Query Builder

```
LOGON
INSERT INTO new_updates
VALUES(SYS.DBNAME, SYS.LOGIN_USER, SYS.SYSEVENT, SYS.DICTIONARY_OBJ_TYPE, SYS.DICTIONARY_OBJ_NAME, SYSDATE);
END;
/
CREATE TABLE HAINA(COD_HAINA NUMBER(37,2) CONSTRAINT PKEY_HAINA PRIMARY KEY,
DENUMIRE VARCHAR(50));
SELECT * FROM HAINA;
ALTER TABLE HAINA
ADD (ANOTIMP VARCHAR2(50));
ALTER TABLE HAINA
DROP COLUMN DENUMIRE;
DROP TABLE HAINA;
SELECT * FROM new_updates;
```

Script Output | Query Result | Query Result 1

All Rows Fetched: 5 in 0.001 seconds

NUME_BD	USER_LOGAT	EVENIMENT	TIP_OBJECT_REFERIT	NUME_OBJECT_REFERIT	DATA_REALIZARI
1 XE	##FRANCISCAPASARE	CREATE	INDEX	PKEY_HAINA	03-JAN-23
2 XE	##FRANCISCAPASARE	CREATE	TABLE	HAINA	03-JAN-23
3 XE	##FRANCISCAPASARE	ALTER	TABLE	HAINA	03-JAN-23
4 XE	##FRANCISCAPASARE	ALTER	TABLE	HAINA	03-JAN-23
5 XE	##FRANCISCAPASARE	DROP	TABLE	HAINA	03-JAN-23

Messages - Log

Cloudy 17°C

Search

WS VS Code

Line 585 Column 22 | Inse