

SGBD - Proiect

Baza de date a unui spital

GAVRILĂ ANDREEA

Grupa 244

1. Prezentați pe scurt baza de date (utilitatea ei). (pg 3)
2. Realizați diagrama entitate-relație (ERD)..... (pg 4)
3. Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare. (pg 5)
4. Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, implementând toate constrângările de integritate necesare (chei primare, cheile externe etc)..... (pg 6)
5. Adăugați informații coerente în tabelele create (minim 3-5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă)..... (pg 11)
6. Definiți un subprogram stocat care să utilizeze un tip de colecție studiat. Apelați subprogramul..... (pg 38)
7. Definiți un subprogram stocat care să utilizeze un tip de cursor studiat. Apelați subprogramul..... (pg 43)
8. Definiți un subprogram stocat de tip funcție care să utilizeze 3 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate..... (pg 47)
9. Definiți un subprogram stocat de tip procedură care să utilizeze 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate..... (pg 52)
10. Definiți un *trigger* de tip LMD la nivel de comandă. Declanșați *trigger*-ul. (pg 58)
11. Definiți un *trigger* de tip LMD la nivel de linie. Declanșați *trigger*-ul..... (pg 61)
12. Definiți un *trigger* de tip LDD. Declanșați *trigger*-ul..... (pg 63)
13. Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului. (pg 65)
14. Definiți un pachet care să includă tipuri de date complexe și obiecte necesare pentru acțiuni integrate.... (pg 84)

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

- Baza de date a fost concepută pentru a facilita buna funcționare a unui spital și de a gestiona toate informațiile corespunzătoare referitoare la angajații instituției, pacienți, cum sunt îngrijitori aceștia, dar și ce tratamente se vor administra de către doctori.

- În baza de date se gasesc informații cu privire la:

- angajații instituției:
 - id-ul de identificare, date personale, id-ul secției în care lucrează și unde este cazul, codul șefului;
 - sunt clasificați în doctori (cu diferite specializari), asistente (cu statutul conform calificării) și infirmiere (cu roluri specifice);

- secțiile spitalului:

- id-ul lor, numele, categoria din care fac parte (mixt/adulți/copii) și zona (roșie, verde, tampon);

- saloanele spitalului:

- id-ul acestora, numele și id-ul secției în care se află;

- pacienții internați:

- id-ul de identificare, date personale, regimul alimentar, id-ul salonului în care se află, data internării și data externării;

- Baza de date îmi permite să vizualizez doar salonul în care a stat pacientul la ultima internare.

- Dacă un pacient se va interna a doua oară, se vor actualiza datele referitoare la id-ul salonului în care se află, data internării și data externării.

- tratamentele administrate:

- id-ul lor, descrierea, utilizarea și data expirării;

- îngrijirea pacienților:

- id-ul pacientului, id-ul asistentei și data îngrijirii;

- Într-o zi, o asistentă raportează că îngrijește pacientul X. În aceeași zi, nu poate raporta această acțiune de două ori. Dar o asistentă poate îngriji mai mulți pacienți diferiți într-o singură zi. Iar un pacient poate fi îngrijit de mai multe asistente în aceeași zi.

- Dacă un pacient se va interna a doua oară, în continuare pot afla ce asistente l-au îngrijit în trecut, căutându-l după id-ul său.

- tratarea pacienților:

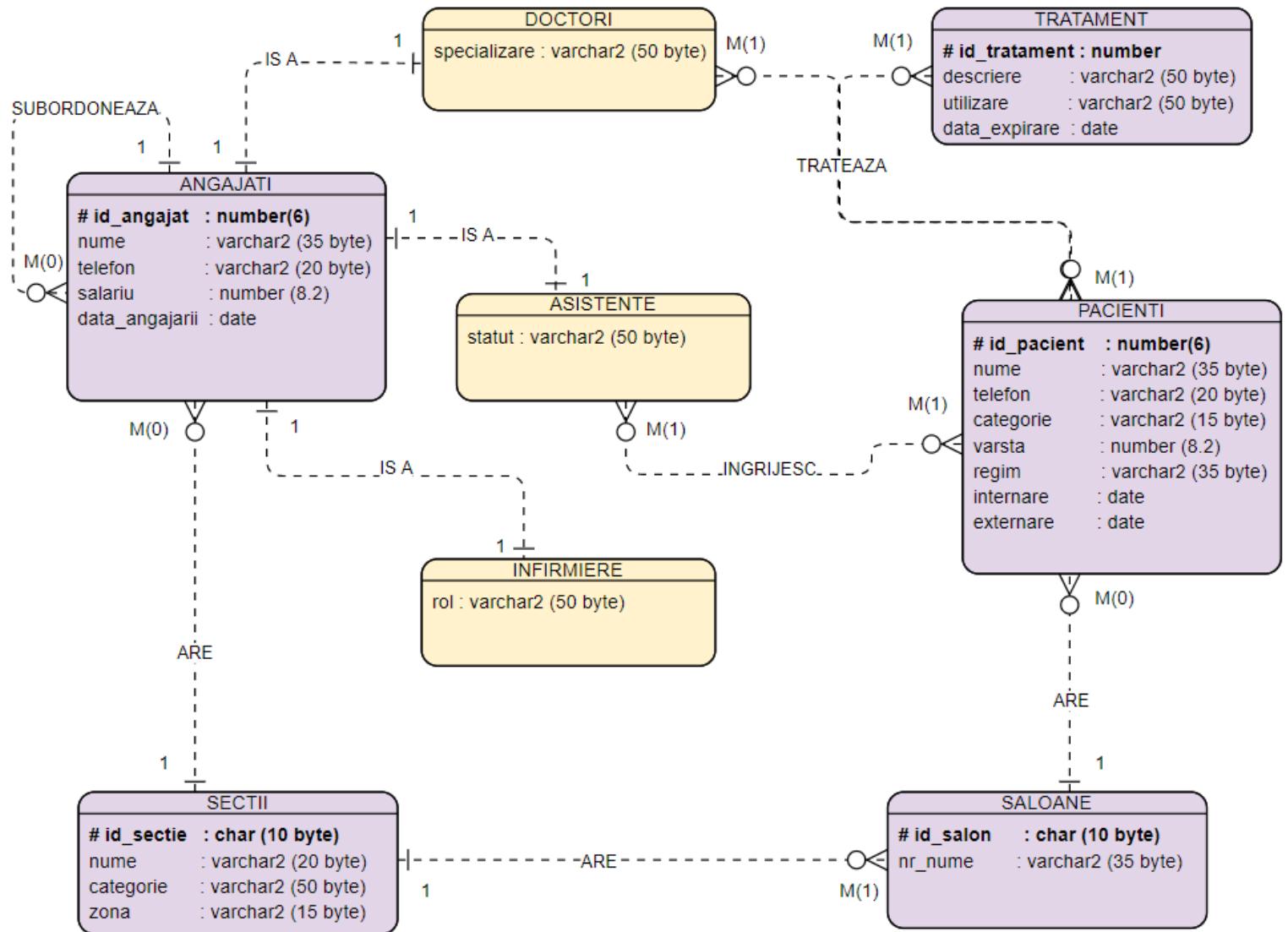
- id-ul doctorului, id-ul pacientului, id-ul tratamentului și data administrației tratamentului;

- Într-o zi, un doctor raportează că tratează pacientul X, cu medicamentul V. În aceeași zi, nu poate raporta această acțiune de două ori. Un doctor poate trata același pacient în data respectivă doar cu alt medicament. Dar un doctor poate trata mai mulți pacienți diferiți într-o singură zi, cu medicamente diferite. Iar dacă este cazul, un pacient poate fi tratat de mai mulți doctori în aceeași zi.

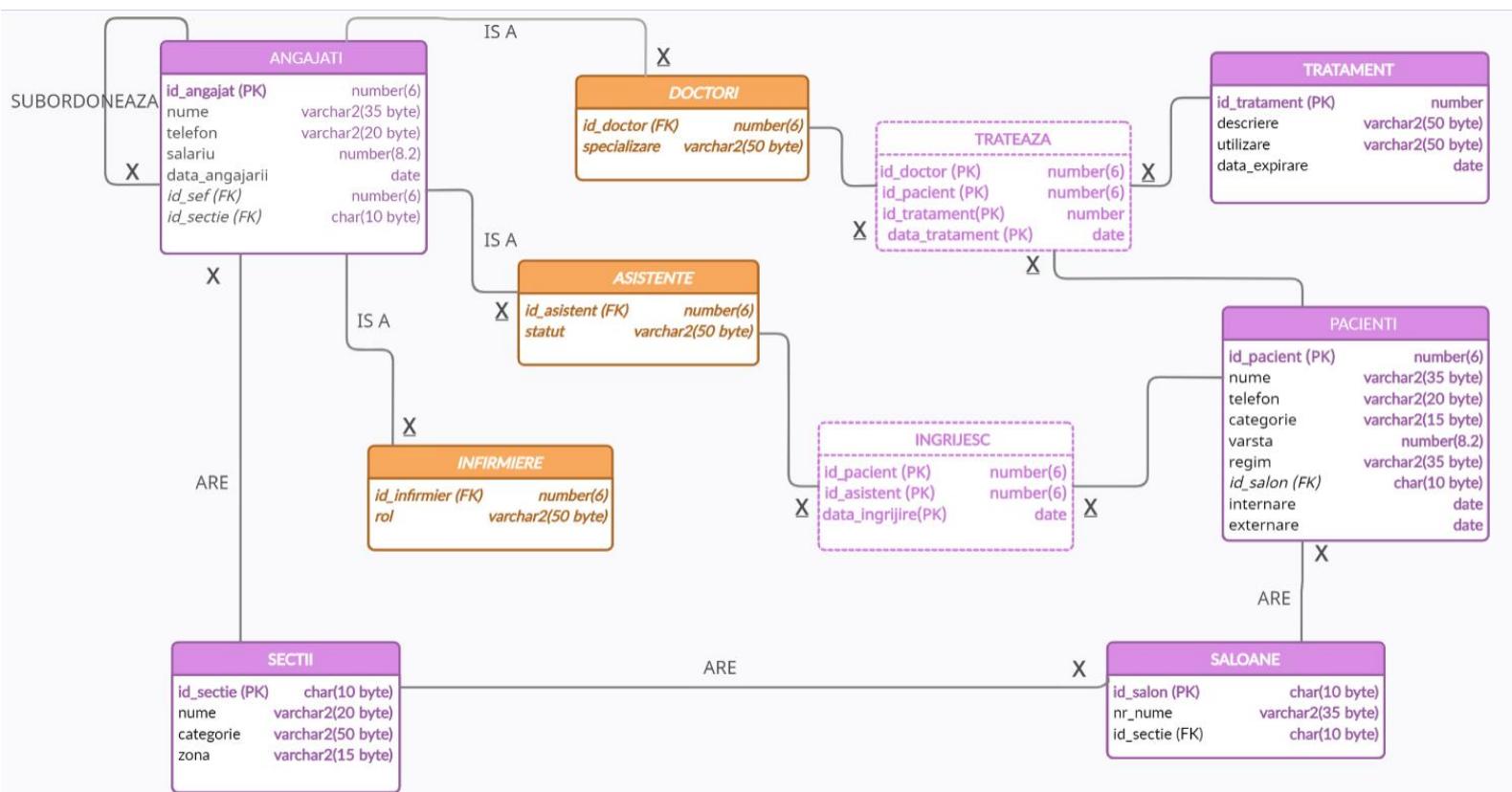
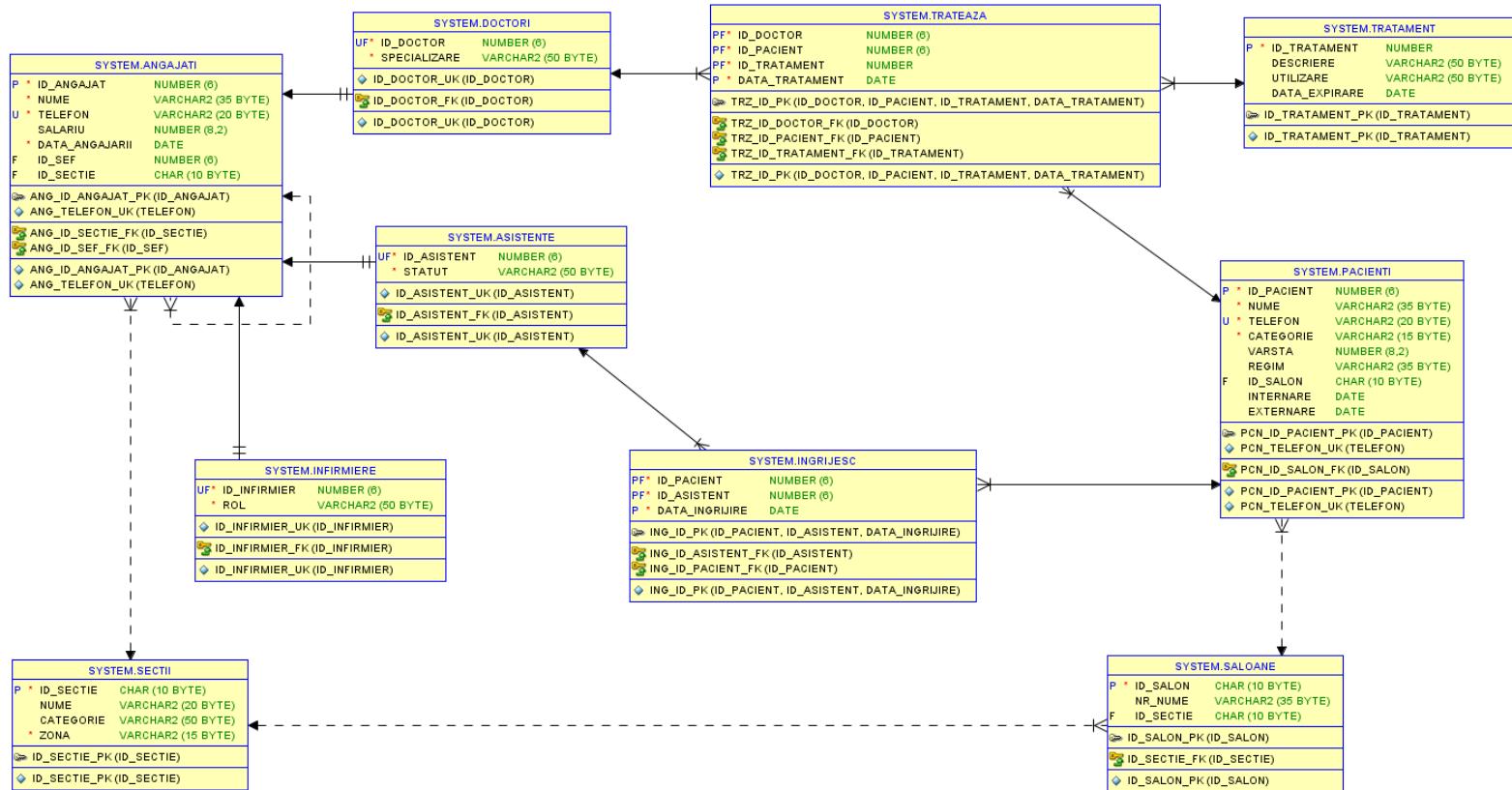
- Dacă un pacient se va interna a doua oară, în continuare pot afla ce doctori l-au tratat anterior și ce tratamente i s-au administrat în trecut, căutându-l după id-ul său.

- Codul sursă este se află în fișierul *Gavrila_Andreea_SQL.sql*.

2. Realizați diagrama entitate-relație (ERD).



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



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

-----SCRIPT-----

```
SET FEEDBACK 1
SET NUMWIDTH 10
SET LINESIZE 80
SET TRIMSPPOOL ON
SET TAB OFF
SET PAGESIZE 100
SET ECHO OFF

-- *****
-- Creez tabela TRATAMENT care contine info despre tratamentele utilizate.

-- ***** Creare tabel TRATAMENT .....

CREATE TABLE TRATAMENT
(
    id_tratament      NUMBER      CONSTRAINT id_tratament_nn      NOT NULL
    , descriere        VARCHAR2(50)
    , utilizare        VARCHAR2(50)
    , data_expirare   date
);
CREATE UNIQUE INDEX id_tratament_pk
ON tratament (id_tratament);

ALTER TABLE tratament
ADD ( CONSTRAINT id_tratament_pk
      PRIMARY KEY (id_tratament)
);
select * from tratament;

-- *****
-- Creez tabela SECTII care contine info despre sectiile spitalului.

-- ***** Creare tabel SECTII .....

CREATE TABLE SECTII
(
    id_sectie        CHAR(10)    CONSTRAINT id_sectie_nn      NOT NULL
    , nume            VARCHAR2(20)
    , categorie       VARCHAR2(50)
    , zona            VARCHAR2(15)  CONSTRAINT zona_nn      NOT NULL
                                CONSTRAINT zona_ck      CHECK (zona IN ('ROSIE','VERDE','TAMPON'))
);
CREATE UNIQUE INDEX id_sectie_pk
ON sectii (id_sectie);

ALTER TABLE sectii
ADD ( CONSTRAINT id_sectie_pk
      PRIMARY KEY (id_sectie)
);
select * from sectii;
```

```

-- *****
-- Creez tabela SALOANE care contine info despre soloanele spitalului.
--
-- ***** Creare tabel SALOANE ....
```

CREATE TABLE SALOANE

- (
- id_salon** **CHAR(10)** **CONSTRAINT id_salon_nn** **NOT NULL**
- ,**nr_nume** **VARCHAR2(35)**
- ,**id_sectie** **CHAR(10)**
-)

CREATE UNIQUE INDEX id_salon_pk

ON saloane (id_salon);

ALTER TABLE saloane

- ADD (CONSTRAINT id_salon_pk**
- PRIMARY KEY (id_salon)**
- ,**CONSTRAINT id_sectie_fk**
- FOREIGN KEY (id_sectie)**
- REFERENCES sectii(id_sectie)**
-)

select * from saloane;

```

-- *****
-- Creez tabela PACIENTI care contine info despre pacientii spitalului.
--
-- ***** Creare tabel PACIENTI ....
```

CREATE TABLE PACIENTI

- (
- id_pacient** **NUMBER(6)**
- ,**nume** **VARCHAR2(35)** **CONSTRAINT pcn_nume_nn** **NOT NULL**
- ,**telefon** **VARCHAR2(20)** **CONSTRAINT pcn_telefon_nn** **NOT NULL**
- ,**categorie** **VARCHAR2(15)** **CONSTRAINT pcn_categorie_nn** **NOT NULL**
- ,**varsta** **NUMBER(8,2)** **CONSTRAINT pcn_categorie_ck** **CHECK (categorie IN ('ADULT','COPIL','SUGAR'))**
- ,**regim** **VARCHAR2(35)**
- ,**id_salon** **CHAR(10)**
- ,**internare** **DATE**
- ,**externare** **DATE**
- ,**CONSTRAINT pcn_telefon_uk** **UNIQUE (telefon)**
-)

CREATE UNIQUE INDEX pcn_id_pacient_pk

ON pacienti (id_pacient);

ALTER TABLE pacienti

- ADD (CONSTRAINT pcn_id_pacient_pk**
- PRIMARY KEY (id_pacient)**
- ,**CONSTRAINT pcn_id_salon_fk**
- FOREIGN KEY (id_salon)**
- REFERENCES saloane (id_salon)**
-)

select * from pacienti;

```

-----  

-- **** Creez tabela ANGAJATI care contine info despre angajatii spitalului.  

--  

-- ***** Creare tabel ANGAJATI ....  

CREATE TABLE ANGAJATI  

(  

    id_angajat      NUMBER(6)      CONSTRAINT ang_nume_nn      NOT NULL  

    , nume          VARCHAR2(35)    CONSTRAINT ang_telefon_nn    NOT NULL  

    , telefon        VARCHAR2(20)    CONSTRAINT ang_salariu_min  CHECK (salariu > 0)  

    , salariu        NUMBER(8,2)     CONSTRAINT ang_data_angajarii_nn NOT NULL  

    , data_angajarii DATE          CONSTRAINT ang_telefon_uk    UNIQUE (telefon)  

    , id_sef         NUMBER(6)  

    , id_sectie      CHAR(10)  

    , CONSTRAINT ang_telefon_uk  UNIQUE (telefon)  

);  

CREATE UNIQUE INDEX ang_id_angajat_pk  

ON angajati (id_angajat);  

ALTER TABLE angajati  

ADD ( CONSTRAINT ang_id_angajat_pk  

      PRIMARY KEY (id_angajat)  

    , CONSTRAINT ang_id_sectie_fk  

      FOREIGN KEY (id_sectie)  

      REFERENCES sectii (id_sectie)  

    , CONSTRAINT ang_id_sef_fk  

      FOREIGN KEY (id_sef)  

      REFERENCES angajati  

);  

select * from angajati;  

-----  

CREATE TABLE DOCTORI  

(  

    id_doctor      NUMBER(6)      CONSTRAINT id_doctor_nn      NOT NULL  

    , specializare  VARCHAR2(50)    CONSTRAINT spec_nn        NOT NULL  

    , CONSTRAINT id_doctor_uk    UNIQUE (id_doctor)  

);  

ALTER TABLE doctori  

ADD ( CONSTRAINT id_doctor_fk  

      FOREIGN KEY (id_doctor)  

      REFERENCES angajati (id_angajat)  

);  

select * from doctori;  

-----  

CREATE TABLE ASISTENTE  

(  

    id_asistent    NUMBER(6)      CONSTRAINT id_asistent_nn    NOT NULL  

    , statut        VARCHAR2(50)    CONSTRAINT statut_nn      NOT NULL  

    , CONSTRAINT id_asistent_uk  UNIQUE (id_asistent)  

);

```

```

ALTER TABLE asistente
ADD ( CONSTRAINT id_asistent_fk
      FOREIGN KEY (id_asistent)
      REFERENCES angajati (id_angajat)
);

```

```
select * from asistente;
```

```

CREATE TABLE INFIRMIERE
(
    id_infirmier NUMBER(6) CONSTRAINT id_infirmier_nn NOT NULL
    ,rol VARCHAR2(50) CONSTRAINT rol_nn NOT NULL
    ,CONSTRAINT id_infirmier_uk UNIQUE (id_infirmier)
);

```

```

ALTER TABLE infirmiere
ADD ( CONSTRAINT id_infirmier_fk
      FOREIGN KEY (id_infirmier)
      REFERENCES angajati (id_angajat)
);

```

```
select * from infirmiere;
```

```
-- Creez tabela asociativa INGRIJESC care contine info despre
-- cum sunt ingrijiti pacientii de asistente, la o anumita data
-- *****
-- ***** Creare tabel asociativ INGRIJESC ....
```

```

CREATE TABLE INGRIJESC
(
    id_pacient NUMBER(6) CONSTRAINT ing_id_pacient_fk REFERENCES pacienti (id_pacient)
    , id_asistent NUMBER(6) CONSTRAINT ing_id_asistent_fk REFERENCES asistente (id_asistent)
    , data_ingrijire DATE CONSTRAINT ing_data_ingrijire_nn NOT NULL
    , CONSTRAINT ing_id_pk PRIMARY KEY (id_pacient, id_asistent, data_ingrijire)
);

```

```
select * from ingriesc;
```

```
-- Creez tabela asociativa TRATEAZA care contine info despre
-- tratarea pacientilor de doctori, ce tratamente au la o anumita data
-- *****
-- ***** Creare tabel asociativ TRATEAZA ....
```

```

CREATE TABLE TRATEAZA
(
    id_doctor NUMBER(6) CONSTRAINT trz_id_doctor_fk REFERENCES doctori (id_doctor)
    , id_pacient NUMBER(6) CONSTRAINT trz_id_pacient_fk REFERENCES pacienti (id_pacient)
    , id_tratament NUMBER CONSTRAINT trz_id_tratament_fk REFERENCES tratament (id_tratament)
    , data_tratament DATE CONSTRAINT trz_data_tratament_nn NOT NULL
);

```

```
, CONSTRAINT trz_id_pk PRIMARY KEY (id_doctor, id_pacient, id_tratament, data_tratament)  
);
```

```
select * from trateaza;  
commit;
```

The screenshot shows a grid of 10 panes, each representing a different table or query result in Oracle SQL Developer. The panes are arranged in two columns and five rows.

- Row 1:** Contains one pane showing the structure of the `PACIENTI` table with columns: ID_PACIENT, NUME, TELEFON, CATEGORIE, VARSTA, REGIM, ID_SALON, INTERNARE, EXTERNARE.
- Row 2:** Contains three panes. The first shows the structure of the `TRATEAZA` table with columns: ID_TRATAMENT, DESCRIERE, UTILIZARE, DATA_EXPIRARE. The second shows the creation of the `PACIENTI` table. The third shows the completion of the task.
- Row 3:** Contains three panes. The first shows the structure of the `SECTIE` table with columns: ID_SECTIE, NUME, CATEGORIE, ZONA. The second shows the alteration of the `PACIENTI` table. The third shows the creation of the `INGRIJESC` table.
- Row 4:** Contains three panes. The first shows the structure of the `SALON` table with columns: ID_SALON, NR_NUME, ID_SECTIE. The second shows the creation of the `TRATEAZA` table. The third shows the commit operation.
- Row 5:** Contains four panes. The first shows the structure of the `ANGAJAT` table with columns: ID_ANGAJAT, NUME, TELEFON, SALARIU, DATA_ANGAJARII, ID_SEF, ID_SECTIE. The second shows the structure of the `DOCTOR` table with columns: ID_DOCTOR, SPECIALIZARE. The third shows the structure of the `ASISTENT` table with columns: ID_ASISTENT, STATUT. The fourth shows the structure of the `INFIRMIER` table with columns: ID_INFIRMIER, ROL.
- Row 6:** Contains one pane showing the structure of the `INGRIJIRE` table with columns: ID_PACIENT, ID_ASISTENT, DATA_INGRIJIRE.
- Row 7:** Contains one pane showing the structure of the `TRATAMENT` table with columns: ID_DOCTOR, ID_PACIENT, ID_TRATAMENT, DATA_TRATAMENT.

5. Adăugați informații coerente în tabelele create (minim 3-5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).

-----SCRIPT-----

```
select * from tratament;
select * from sectii;
select * from saloane;
select * from pacienti;
select * from angajati;
select * from doctori;
select * from asistente;
select * from infirmiere;
select * from ingrijesc;
select * from trateaza;
```

INSERT ALL

```
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (1, 'Biosun', 'Protectie', TO_DATE('05/03/2022', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (2, 'Hepiflor', 'Protectie', TO_DATE('18/05/2021', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (3, 'Linex', 'Protectie', TO_DATE('12/06/2022', 'DD/MM/YYYY'))

INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (11, 'Ampiplus', 'Antibiotic', TO_DATE('12/02/2022', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (12, 'Azitromicina', 'Antibiotic', TO_DATE('17/06/2023', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (13, 'Cefotaz', 'Antibiotic', TO_DATE('16/08/2022', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (14, 'Cefort', 'Antibiotic', TO_DATE('14/10/2021', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (15, 'Gentamicina', 'Antibiotic', TO_DATE('13/12/2022', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (16, 'Gentamicina', 'Antibiotic', TO_DATE('04/01/2023', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (17, 'Klacid', 'Antibiotic', TO_DATE('31/05/2022', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (18, 'Zinat', 'Antibiotic', TO_DATE('10/03/2021', 'DD/MM/YYYY'))

INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (21, 'Adrenalina', 'Antisoc', TO_DATE('10/03/2021', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (22, 'Dexametazona', 'Antisoc', TO_DATE('17/05/2022', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (23, 'Hidrocortizon', 'Antisoc', TO_DATE('13/07/2021', 'DD/MM/YYYY'))

INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (31, 'Aspatofort', 'Vitamine', TO_DATE('10/07/2021', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (32, 'B1', 'Vitamine', TO_DATE('11/03/2022', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (33, 'B6', 'Vitamine', TO_DATE('16/05/2021', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (34, 'C', 'Vitamine', TO_DATE('11/02/2023', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (35, 'Calciu gluconic', 'Vitamine', TO_DATE('28/04/2022', 'DD/MM/YYYY'))
```

```
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (36, 'Fitomenadion', 'Vitamine', TO_DATE('19/06/2022', 'DD/MM/YYYY'))
```

```
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (41, 'Captopril', 'Cardiac', TO_DATE('10/03/2021', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (42, 'Dipiridamol', 'Cardiac', TO_DATE('18/02/2022', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (43, 'Furosemid', 'Cardiac', TO_DATE('14/01/2023', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (44, 'Tertensiv', 'Cardiac', TO_DATE('11/05/2021', 'DD/MM/YYYY'))
```

```
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (51, 'Pentoxiretard', 'Neurologic', TO_DATE('10/03/2021', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (52, 'Milgama', 'Neurologic', TO_DATE('05/08/2022', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (53, 'Piracetam', 'Neurologic', TO_DATE('17/09/2021', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (54, 'Fenobarbital', 'Neurologic', TO_DATE('14/10/2023', 'DD/MM/YYYY'))
```

```
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (61, 'Dicarbocalm', 'Stomac', TO_DATE('10/10/2021', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (62, 'Omeran', 'Stomac', TO_DATE('09/11/2022', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (63, 'NoSpa', 'Stomac', TO_DATE('02/03/2022', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (64, 'Spaverin', 'Stomac', TO_DATE('05/07/2023', 'DD/MM/YYYY'))
```

```
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (71, 'Ambroxol', 'Tuse', TO_DATE('10/04/2021', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (72, 'Codeina', 'Tuse', TO_DATE('14/03/2021', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (73, 'Mucosolvan', 'Tuse', TO_DATE('15/07/2022', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (74, 'Paxeladine', 'Tuse', TO_DATE('16/10/2022', 'DD/MM/YYYY'))
INTO tratament (id_tratament, descriere, utilizare, data_expirare)
VALUES (75, 'Tusocalm', 'Tuse', TO_DATE('13/06/2023', 'DD/MM/YYYY'))
```

```
SELECT * FROM dual;
select * from tratament;
```

ID_TRATAMENT	DESCRIERE	UTILIZARE	DATA_EXPIRARE
4	11 Ampiplus	Antibiotic	12-FEB-22
5	12 Azitromicina	Antibiotic	17-JUN-23
6	13 Cefotaz	Antibiotic	16-AUG-22
7	14 Cefort	Antibiotic	14-OCT-21
8	15 Gentamicina	Antibiotic	13-DEC-22
9	16 Gentamicina	Antibiotic	04-JAN-23
10	17 Klacid	Antibiotic	31-MAY-22
11	18 Zinat	Antibiotic	10-MAR-21
12	21 Adrenalina	Antisoc	10-MAR-21
13	22 Dexametazona	Antisoc	17-MAY-22

----- SECTII -----

INSERT ALL

```
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (1, 'UPU', null , 'ROSIE')
```

```
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (2, 'ATI', null , 'ROSIE')
```

```
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (3, 'LABORATOR', null , 'VERDE')
```

```
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (41, 'RADIOLOGIE', null , 'ROSIE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (42, 'RADIOLOGIE', null , 'VERDE')
```

```
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (51, 'CHIRURGIE', 'adulti' , 'ROSIE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (52, 'CHIRURGIE', 'adulti' , 'VERDE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (53, 'CHIRURGIE', 'adulti' , 'TAMPON')
```

```
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (61, 'ORTOPEDIE', 'adulti' , 'ROSIE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (62, 'ORTOPEDIE', 'adulti' , 'VERDE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (63, 'ORTOPEDIE', 'adulti' , 'TAMPON')
```

```
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (71, 'NEUROLOGIE', 'adulti' , 'ROSIE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (72, 'NEUROLOGIE', 'adulti' , 'VERDE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (73, 'NEUROLOGIE', 'adulti' , 'TAMPON')
```

```
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (81, 'INTERNE', 'adulti' , 'ROSIE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (82, 'INTERNE', 'adulti' , 'VERDE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (83, 'INTERNE', 'adulti' , 'TAMPON')
```

```
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (91, 'PNEUMOLOGIE', 'adulti' , 'ROSIE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (92, 'PNEUMOLOGIE', 'adulti' , 'VERDE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (93, 'PNEUMOLOGIE', 'adulti' , 'TAMPON')
```

```
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (101, 'ONCOLOGIE', 'adulti' , 'ROSIE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (102, 'ONCOLOGIE', 'adulti' , 'VERDE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (103, 'ONCOLOGIE', 'adulti' , 'TAMPON')
```

```

INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (111, 'OFTALMOLOGIE', 'adulti', 'ROSIE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (112, 'OFTALMOLOGIE', 'adulti', 'VERDE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (113, 'OFTALMOLOGIE', 'adulti', 'TAMPON')

```

```

INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (121, 'ORL', 'adulti', 'ROSIE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (122, 'ORL', 'adulti', 'VERDE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (123, 'ORL', 'adulti', 'TAMPON')

```

```

INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (131, 'PEDIATRIE', 'copii', 'ROSIE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (132, 'PEDIATRIE', 'copii', 'VERDE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (133, 'PEDIATRIE', 'copii', 'TAMPON')

```

```

INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (141, 'GINECOLOGIE', 'adulti', 'ROSIE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (142, 'GINECOLOGIE', 'adulti', 'VERDE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (143, 'GINECOLOGIE', 'adulti', 'TAMPON')

```

```

INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (151, 'NEONATOLOGIE', 'copii', 'ROSIE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (152, 'NEONATOLOGIE', 'copii', 'VERDE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (153, 'NEONATOLOGIE', 'copii', 'TAMPON')

```

```

INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (161, 'INFECTIOASE', 'adulti', 'ROSIE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (162, 'INFECTIOASE', 'adulti', 'VERDE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (163, 'INFECTIOASE', 'adulti', 'TAMPON')

```

```

INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (171, 'PSIHIATRIE', 'adulti', 'ROSIE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (172, 'PSIHIATRIE', 'adulti', 'VERDE')
INTO SECTII (id_sectie, nume, categorie, zona)
VALUES (173, 'PSIHIATRIE', 'adulti', 'TAMPON')

```

```

SELECT * FROM dual;
select * from sectii;

```

ID_SECTIE	NUME	CATEGORIE	ZONA
1 1	UPU	(null)	ROSIE
2 2	ATI	(null)	ROSIE
3 3	LABORATOR	(null)	VERDE
4 41	RADIOLOGIE	(null)	ROSIE
5 42	RADIOLOGIE	(null)	VERDE
6 51	CHIRURGIE	adulti	ROSIE
7 52	CHIRURGIE	adulti	VERDE
8 53	CHIRURGIE	adulti	TAMPON
9 61	ORTOPEDIE	adulti	ROSIE
10 62	ORTOPEDIE	adulti	VERDE
11 63	ORTOPEDIE	adulti	TAMPON

```

203 | INTO SECTII (id_sectie, nume, categorie, zona)
204 | VALUES (162, 'INFECTIOASE', 'adulti', 'VERDE')
205 | INTO SECTII (id_sectie, nume, categorie, zona)
206 | VALUES (163, 'INFECTIOASE', 'adulti', 'TAMPON')
207 |
208 | INTO SECTII (id_sectie, nume, categorie, zona)
209 | VALUES (171, 'PSIHIATRIE', 'adulti', 'ROSIE')
210 | INTO SECTII (id_sectie, nume, categorie, zona)
211 | VALUES (172, 'PSIHIATRIE', 'adulti', 'VERDE')
212 | INTO SECTII (id_sectie, nume, categorie, zona)
213 | VALUES (173, 'PSIHIATRIE', 'adulti', 'TAMPON')
214 |
215 | SELECT * FROM dual;
216 | select * from sectii;
217 |

Script Output X | Query Result X
| Task completed in 0.051 seconds
37 rows inserted.

44 rows inserted.

```

----- SALOANE -----

INSERT ALL

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('1_A', 'A', '1')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('1_B', 'B', '1')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('2_A', 'A', '2')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('2_B', 'B', '2')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('41_A', 'A', '41')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('41_B', 'B', '41')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('42_A', 'A', '42')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('42_B', 'B', '42')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('51_A', 'A', '51')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('51_B', 'B', '51')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('52_A', 'A', '52')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('52_B', 'B', '52')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('53_A', 'A', '53')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('53_B', 'B', '53')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('61_A', 'A', '61')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('61_B', 'B', '61')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('62_A', 'A', '62')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('62_B', 'B', '62')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('63_A', 'A', '63')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('63_B', 'B', '63')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('71_A', 'A', '71')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('71_B', 'B', '71')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('72_A', 'A', '72')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('72_B', 'B', '72')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('73_A', 'A', '73')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('73_B', 'B', '73')
```

```
-----
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('81_A', 'A', '81')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('81_B', 'B', '81')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('82_A', 'A', '82')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('82_B', 'B', '82')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('83_A', 'A', '83')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('83_B', 'B', '83')
```

```
-----
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('91_A', 'A', '91')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('91_B', 'B', '91')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('92_A', 'A', '92')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('92_B', 'B', '92')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('93_A', 'A', '93')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('93_B', 'B', '93')
```

```
-----
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('101_A', 'A', '101')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('101_B', 'B', '101')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('102_A', 'A', '102')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('102_B', 'B', '102')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
```

```
-----
```

```
VALUES ('103_A', 'A', '103')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('103_B', 'B', '103')
```

```
-----
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('111_A', 'A', '111')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('111_B', 'B', '111')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('112_A', 'A', '112')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('112_B', 'B', '112')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('113_A', 'A', '113')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('113_B', 'B', '113')
```

```
-----
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('121_A', 'A', '121')
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
```

```
VALUES ('121_B', 'B', '121')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('122_A', 'A', '122')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('122_B', 'B', '122')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('123_A', 'A', '123')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('123_B', 'B', '123')
-----
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('131_A', 'A', '131')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('131_B', 'B', '131')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('132_A', 'A', '132')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('132_B', 'B', '132')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('133_A', 'A', '133')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('133_B', 'B', '133')
-----
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('141_A', 'A', '141')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('141_B', 'B', '141')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('142_A', 'A', '142')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('142_B', 'B', '142')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('143_A', 'A', '143')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('143_B', 'B', '143')
-----
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('151_A', 'A', '151')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('151_B', 'B', '151')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('152_A', 'A', '152')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('152_B', 'B', '152')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('153_A', 'A', '153')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('153_B', 'B', '153')
-----
```

```
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('161_A', 'A', '161')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('161_B', 'B', '161')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('162_A', 'A', '162')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('162_B', 'B', '162')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('163_A', 'A', '163')
```

```

INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('163_B', 'B', '163')
-----
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('171_A', 'A', '171')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('171_B', 'B', '171')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('172_A', 'A', '172')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('172_B', 'B', '172')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('173_A', 'A', '173')
INTO SALOANE (id_salon, nr_nume, id_sectie)
VALUES ('173_B', 'B', '173')
-----
SELECT * FROM dual;
select * from saloane;
commit;

```

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Script Output' tab displays the SQL code used to insert data into the SALOANE table. On the right, the 'Query Result' tab shows the confirmation message '78 rows inserted.'.

```

382 INTO SALOANE (id_salon, nr_nume, id_sectie)
383 VALUES ('172_A', 'A', '172')
384 INTO SALOANE (id_salon, nr_nume, id_sectie)
385 VALUES ('172_B', 'B', '172')
386 INTO SALOANE (id_salon, nr_nume, id_sectie)
387 VALUES ('173_A', 'A', '173')
388 INTO SALOANE (id_salon, nr_nume, id_sectie)
389 VALUES ('173_B', 'B', '173')
390 -----
391 SELECT * FROM dual;
392 select * from saloane;

```

78 rows inserted.

----- ANGAJATI -----

----- DOCTORI -----

INSERT ALL

```

INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (1, 'Seceleanu', 0745693502, 7501, TO_DATE('05/03/1990', 'DD/MM/YYYY'),null, 1 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (2, 'Geru', 0745573912, 7379, TO_DATE('04/07/1989', 'DD/MM/YYYY'),null, 1 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (3, 'Tarcuta', 0756493124, 7289, TO_DATE('01/04/1993', 'DD/MM/YYYY'),null, 1 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (4, 'Marin', 0745328367, 7060, TO_DATE('07/09/1994', 'DD/MM/YYYY'),null, 1 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (5, 'Stan', 0745397761, 7565, TO_DATE('01/07/1996', 'DD/MM/YYYY'),null, 2 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (6, 'Zarzu', 0745368371, 7456, TO_DATE('02/08/1993', 'DD/MM/YYYY'),null, 2 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (7, 'Lungu', 0765327670, 7493, TO_DATE('03/09/1995', 'DD/MM/YYYY'),null, 2 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (8, 'Andronic', 0746359747, 5098, TO_DATE('11/04/2005', 'DD/MM/YYYY'),null, 3 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (9, 'Luca', 0710054131 ,6067 , TO_DATE('03/02/2005', 'DD/MM/YYYY'),null,41 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (10, 'Ciuchi', 0711507299 ,5002 , TO_DATE('19/10/2007', 'DD/MM/YYYY'),null,42 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (11, 'Manolache', 0701829928 ,6032 , TO_DATE('14/10/2012', 'DD/MM/YYYY'),null,51 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (12, 'Vartic', 0738873162 ,6901 , TO_DATE('18/03/1993', 'DD/MM/YYYY'),null,52 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (13, 'Bejenariu', 0706206823 ,5482 , TO_DATE('20/06/1999', 'DD/MM/YYYY'),null,61 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)

```

```

VALUES (14, 'Bratu',0788813011 ,5173 , TO_DATE('05/05/2018', 'DD/MM/YYYY'),null,62 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (15, 'Paraschivescu',0739061814 ,6934 , TO_DATE('07/06/2005', 'DD/MM/YYYY'),null, 71 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (16, 'Enache',0723356223 ,6599 , TO_DATE('26/12/2003', 'DD/MM/YYYY'),null,72 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (17, 'Androne',0754037725 ,5060 , TO_DATE('26/02/1990', 'DD/MM/YYYY'),null,81 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (18, 'Halmaciu',0731525122 ,5256 , TO_DATE('25/04/2003', 'DD/MM/YYYY'),null,82
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (19, 'Dinu',0772468443 ,5716 , TO_DATE('27/12/1996', 'DD/MM/YYYY'),null,91 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (20, 'Beceru',0768788298 ,5895 , TO_DATE('02/12/2005', 'DD/MM/YYYY'),null,92 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (21, 'Musteata',0707635503 ,5283 , TO_DATE('24/11/2011', 'DD/MM/YYYY'),null,101 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (22, 'Asmarandei',0720548233 ,5845 , TO_DATE('24/01/2014', 'DD/MM/YYYY'),null,102
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (23, 'Ganea',0788648938 ,6364 , TO_DATE('26/10/1990', 'DD/MM/YYYY'),null,111 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (24, 'Chirtes',0785821651 ,6010 , TO_DATE('24/07/1992', 'DD/MM/YYYY'),null,112 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (25, 'Dantes', 0744635018 ,5963 , TO_DATE('20/10/2001', 'DD/MM/YYYY'),null,121 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (26, 'Solomon', 0731277957 ,5546 , TO_DATE('01/09/2004', 'DD/MM/YYYY'),null,122 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (27, 'Hirjanu', 0733481166 ,6326 , TO_DATE('29/11/2009', 'DD/MM/YYYY'),null,131 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (28, 'Dogaru', 0717585671 ,6285 , TO_DATE('16/06/2014', 'DD/MM/YYYY'),null,132 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (29, 'Radoi',0702814103 ,5580 , TO_DATE('23/01/1996', 'DD/MM/YYYY'),null,141 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (30, 'Iorga',0742522211 ,5108 , TO_DATE('29/11/1995', 'DD/MM/YYYY'),null,142 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (31, 'Roman',0716370663 ,5413 , TO_DATE('16/03/2006', 'DD/MM/YYYY'),null,151 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (32, 'Andrei',0764319059 ,5896 , TO_DATE('08/09/2008', 'DD/MM/YYYY'),null,152 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (33, 'Pancescu',0744961280 ,5991 , TO_DATE('06/12/2014', 'DD/MM/YYYY'),null,161 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (34, 'Matescu',0758744671 ,6431 , TO_DATE('05/03/2012', 'DD/MM/YYYY'),null,162 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (35, 'Aldea',0702365674 ,5502 , TO_DATE('26/05/2017', 'DD/MM/YYYY'),null,171 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (36, 'Anghel',0748702976 ,5846 , TO_DATE('10/09/2003', 'DD/MM/YYYY'),null,172 )
SELECT * FROM dual;
select * from angajati;
commit;

```

INSERT ALL

```

INTO DOCTORI (id_doctor, specializare)
VALUES (1, 'Medicina_Urgenta')
INTO DOCTORI (id_doctor, specializare)
VALUES (2, 'Medicina_Urgenta')
INTO DOCTORI (id_doctor, specializare)
VALUES (3, 'Medicina_Urgenta')
INTO DOCTORI (id_doctor, specializare)
VALUES (4, 'Medicina_Urgenta')

```

```
INTO DOCTORI (id_doctor, specializare)
VALUES (5, 'Anestezie_Terapie_Intensiva')
INTO DOCTORI (id_doctor, specializare)
VALUES (6, 'Anestezie_Terapie_Intensiva')
INTO DOCTORI (id_doctor, specializare)
VALUES (7, 'Anestezie_Terapie_Intensiva')
INTO DOCTORI (id_doctor, specializare)
VALUES (8, 'Laborator_Epidemiolog')
INTO DOCTORI (id_doctor, specializare)
VALUES (9, 'Radioterapie')
INTO DOCTORI (id_doctor, specializare)
VALUES (10, 'Radioterapie')
INTO DOCTORI (id_doctor, specializare)
VALUES (11, 'Chirurgie_generala')
INTO DOCTORI (id_doctor, specializare)
VALUES (12, 'Chirurgie_cardiovasculara')
INTO DOCTORI (id_doctor, specializare)
VALUES (13, 'Ortopedie_traumatologie')
INTO DOCTORI (id_doctor, specializare)
VALUES (14, 'Ortopedie_pediatrica')
INTO DOCTORI (id_doctor, specializare)
VALUES (15, 'Neurologie')
INTO DOCTORI (id_doctor, specializare)
VALUES (16, 'Neurologie_pediatrica')
INTO DOCTORI (id_doctor, specializare)
VALUES (17, 'Medicina_interna')
INTO DOCTORI (id_doctor, specializare)
VALUES (18, 'Medicina_interna')
INTO DOCTORI (id_doctor, specializare)
VALUES (19, 'Pneumologie')
INTO DOCTORI (id_doctor, specializare)
VALUES (20, 'Pneumologie_pediatrica')
INTO DOCTORI (id_doctor, specializare)
VALUES (21, 'Oncologie')
INTO DOCTORI (id_doctor, specializare)
VALUES (22, 'Oncologie_pediatrica')
INTO DOCTORI (id_doctor, specializare)
VALUES (23, 'Oftalmologie')
INTO DOCTORI (id_doctor, specializare)
VALUES (24, 'Oftalmologie')
INTO DOCTORI (id_doctor, specializare)
VALUES (25, 'ORL')
INTO DOCTORI (id_doctor, specializare)
VALUES (26, 'ORL')
INTO DOCTORI (id_doctor, specializare)
VALUES (27, 'Pediatrie')
INTO DOCTORI (id_doctor, specializare)
VALUES (28, 'Pediatrie')
INTO DOCTORI (id_doctor, specializare)
VALUES (29, 'Ginecologie')
INTO DOCTORI (id_doctor, specializare)
VALUES (30, 'Ginecologie')
INTO DOCTORI (id_doctor, specializare)
VALUES (31, 'Neonatologie')
INTO DOCTORI (id_doctor, specializare)
VALUES (32, 'Neonatologie')
INTO DOCTORI (id_doctor, specializare)
VALUES (33, 'Boli_infectioase')
INTO DOCTORI (id_doctor, specializare)
```

```

VALUES (34, 'Boli infectioase')
INTO DOCTORI (id_doctor, specializare)
VALUES (35, 'Psihiatrie')
INTO DOCTORI (id_doctor, specializare)
VALUES (36, 'Psihiatrie')

-----  

SELECT * FROM dual;
select * from doctori;
commit;

-----  

----- ASISTENTE -----  

-----  

INSERT ALL
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (37, 'Iacob', 0760058261, 4775 , TO_DATE('14/06/1992', 'DD/MM/YYYY'),1, 1 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (38, 'Mihai',0762169693 , 4873 , TO_DATE('12/08/1990', 'DD/MM/YYYY'),2,1 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (39, 'Jitea', 0721540147, 4988 , TO_DATE('05/05/1990', 'DD/MM/YYYY'),3,1 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (40, 'Copacel', 0711525821, 4897 , TO_DATE('23/10/1994', 'DD/MM/YYYY'),4,1 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (41, 'Ratusanu',0766210085 , 3641 , TO_DATE('03/09/2000', 'DD/MM/YYYY'),5,2 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (42, 'Hanganu',0788844054 , 3577 , TO_DATE('12/06/1992', 'DD/MM/YYYY'),6,2 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (43, 'Manas',0738474764 , 3867 , TO_DATE('20/04/1995', 'DD/MM/YYYY'),7,2 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (44, 'Nechita',0781258389 , 4232 , TO_DATE('30/07/2013', 'DD/MM/YYYY'),8,3 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (45, 'Ursu',0732181446 , 4419 , TO_DATE('01/03/2002', 'DD/MM/YYYY'),9,41 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (46, 'Demsa',0776407261 , 4306 , TO_DATE('29/05/2014', 'DD/MM/YYYY'),10,42 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (47, 'Constandis',0786089856 , 4068 , TO_DATE('26/08/2012', 'DD/MM/YYYY'),11,51)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (48, 'Blanaru',0703906534 , 4668 , TO_DATE('26/05/1990', 'DD/MM/YYYY'),12,52 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (49, 'Moisa',0779068733 , 4897 , TO_DATE('06/07/1997', 'DD/MM/YYYY'),13,61 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (50, 'Ciubotaru',0727739719 , 3542 , TO_DATE('23/10/1998', 'DD/MM/YYYY'),14,62
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (51, 'Tiron',0702619437 , 4606 , TO_DATE('26/12/1999', 'DD/MM/YYYY'),15,71 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (52, 'Grigore',0730109126 , 4418 , TO_DATE('22/07/1991', 'DD/MM/YYYY'),16,72 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (53, 'Cibotaru',0789745296 , 4791 , TO_DATE('24/07/2005', 'DD/MM/YYYY'),17, 81)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (54, 'Olteanu',0705315497 , 3547 , TO_DATE('05/01/2000', 'DD/MM/YYYY'),18, 82)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (55, 'Enea',0775622051 , 4870 , TO_DATE('19/05/2004', 'DD/MM/YYYY'),19,91 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (56, 'Grajdeanu',0758536668 , 4023 , TO_DATE('12/07/1995', 'DD/MM/YYYY'),20, 92)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (57, 'Manaila',0718711753 , 3860 , TO_DATE('08/02/1990', 'DD/MM/YYYY'),21,101 )

```

```

INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (58, 'Triscaru',0705161819 , 3765 , TO_DATE('22/01/2006', 'DD/MM/YYYY'),22,102)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (59, 'Moise',0765752290 , 3774 , TO_DATE('04/03/1996', 'DD/MM/YYYY'),23,111 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (60, 'Ghiuta',0739943441 , 3503 , TO_DATE('24/02/1990', 'DD/MM/YYYY'),24,112 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (61, 'Vintila',0700746667 , 3764 , TO_DATE('28/05/2014', 'DD/MM/YYYY'),25,121 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (62, 'Andries',0755971411 , 3863 , TO_DATE('02/07/1999', 'DD/MM/YYYY'),26,122)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (63, 'Gavrila',0766218692 , 4925 , TO_DATE('12/01/1995', 'DD/MM/YYYY'),27,131)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (64, 'Batinru',0713434091 , 4816 , TO_DATE('18/01/1996', 'DD/MM/YYYY'),28,132 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (65, 'Chiriac',0742120779 , 3536 , TO_DATE('07/12/1997', 'DD/MM/YYYY'),29,141)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (66, 'Ciurcan',0731243929 , 3852 , TO_DATE('19/08/1996', 'DD/MM/YYYY'),30,142)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (67, 'Potirniche',0779326052 , 4712 , TO_DATE('21/06/1994', 'DD/MM/YYYY'),31,151)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (68, 'Botezatu',0777658031 , 4069 , TO_DATE('27/04/2012', 'DD/MM/YYYY'),32,152 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (69, 'Sisu',0734675339 , 3502 , TO_DATE('23/04/1997', 'DD/MM/YYYY'),33,161 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (70, 'Rosca',0740374171 , 3863 , TO_DATE('03/08/2001', 'DD/MM/YYYY'),34,162 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (71, 'Anutei',0736472612 , 3764 , TO_DATE('21/07/2012', 'DD/MM/YYYY'),35,171 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (72, 'Asandei',0776812252 , 3536 , TO_DATE('01/09/2003', 'DD/MM/YYYY'),36,172 )
SELECT * FROM dual;
select * from angajati;
commit;

```

INSERT ALL

```

INTO ASISTENTE (id_asistent, statut) --UPU
VALUES (37, 'Generalist')
INTO ASISTENTE (id_asistent, statut)
VALUES (38, 'Generalist')
INTO ASISTENTE (id_asistent, statut)
VALUES (39, 'Generalist')
INTO ASISTENTE (id_asistent, statut)
VALUES (40, 'Generalist')
INTO ASISTENTE (id_asistent, statut) --ATI
VALUES (41, 'Generalist_ATI')
INTO ASISTENTE (id_asistent, statut)
VALUES (42, 'Generalist_ATI')
INTO ASISTENTE (id_asistent, statut)
VALUES (43, 'Generalist_ATI')
INTO ASISTENTE (id_asistent, statut) --Lab
VALUES (44, 'Laborator')
INTO ASISTENTE (id_asistent, statut) --Radiologie
VALUES (45, 'Radiologie')
INTO ASISTENTE (id_asistent, statut)
VALUES (46, 'Radiologie')
INTO ASISTENTE (id_asistent, statut) --Chirurgie
VALUES (47, 'Generalist_Bloc_Operator')
INTO ASISTENTE (id_asistent, statut)

```

```
VALUES (48, 'Generalist_Bloc_Operator')
INTO ASISTENTE (id_asistent, statut) --Ortopedie
VALUES (49, 'Generalist')
INTO ASISTENTE (id_asistent, statut)
VALUES (50, 'Generalist_Bloc_Operator')
INTO ASISTENTE (id_asistent, statut) --Neurologie
VALUES (51, 'Generalist')
INTO ASISTENTE (id_asistent, statut)
VALUES (52, 'Generalist')
INTO ASISTENTE (id_asistent, statut) --Interne
VALUES (53, 'Generalist')
INTO ASISTENTE (id_asistent, statut)
VALUES (54, 'Generalist')
INTO ASISTENTE (id_asistent, statut) --Pneumologie
VALUES (55, 'Generalist')
INTO ASISTENTE (id_asistent, statut)
VALUES (56, 'Generalist')
INTO ASISTENTE (id_asistent, statut) --Oncologie
VALUES (57, 'Generalist_Oncologie')
INTO ASISTENTE (id_asistent, statut)
VALUES (58, 'Generalist_Oncologie')
INTO ASISTENTE (id_asistent, statut) --Oftalmo
VALUES (59, 'Generalist')
INTO ASISTENTE (id_asistent, statut)
VALUES (60, 'Generalist_Bloc_Operator')
INTO ASISTENTE (id_asistent, statut) --ORL
VALUES (61, 'Generalist')
INTO ASISTENTE (id_asistent, statut)
VALUES (62, 'Generalist_Bloc_Operator')
INTO ASISTENTE (id_asistent, statut) --Pediatrie
VALUES (63, 'Pediatru')
INTO ASISTENTE (id_asistent, statut)
VALUES (64, 'Generalist_Pediatru')
INTO ASISTENTE (id_asistent, statut) --Ginecologie
VALUES (65, 'Generalist')
INTO ASISTENTE (id_asistent, statut)
VALUES (66, 'Generalist')
INTO ASISTENTE (id_asistent, statut) --Neonatologie
VALUES (67, 'Generalist')
INTO ASISTENTE (id_asistent, statut)
VALUES (68, 'Generalist_Pediatru')
INTO ASISTENTE (id_asistent, statut) --Infectioase
VALUES (69, 'Generalist')
INTO ASISTENTE (id_asistent, statut)
VALUES (70, 'Generalist')
INTO ASISTENTE (id_asistent, statut) --Psihiatrie
VALUES (71, 'Generalist_Psihiatrie')
INTO ASISTENTE (id_asistent, statut)
VALUES (72, 'Generalist_Psihiatrie')
```

```
SELECT * FROM dual;
select * from asistente;
commit;
```

```
----- INFIRMIERE -----
```

```
INSERT ALL
```

```
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
```

```

VALUES (73, 'Adavanei', 0755644620, 2922 , TO_DATE('11/10/1995', 'DD/MM/YYYY'),37, 1)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (74, 'Duta',0749023868 , 2720 , TO_DATE('17/02/2014', 'DD/MM/YYYY'),38,1 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (75, 'Alupei', 0720321267, 2877 , TO_DATE('30/08/1996', 'DD/MM/YYYY'),39,1 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (76, 'Rogoză', 0798171043, 2919 , TO_DATE('26/12/2009', 'DD/MM/YYYY'),40,1 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (77, 'Maran',0711041977 , 2614 , TO_DATE('23/04/2005', 'DD/MM/YYYY'),41,2 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (78, 'Palaghita',0782857780 , 2515 , TO_DATE('30/09/2008', 'DD/MM/YYYY'),42,2 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (79, 'Girileanu',0702158623 , 2831 , TO_DATE('23/02/2012', 'DD/MM/YYYY'),43,2 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (80, 'Nistor',0738817697 , 2755 , TO_DATE('30/01/2007', 'DD/MM/YYYY'),44,3 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (81, 'Munteanu',0792616673 , 2308 , TO_DATE('28/10/2000', 'DD/MM/YYYY'),45,41 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (82, 'Bichescu',0701985980 , 2212 , TO_DATE('24/07/1994', 'DD/MM/YYYY'),46,42 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (83, 'Lepsă',0791847730 , 2075 , TO_DATE('18/07/2004', 'DD/MM/YYYY'),47,51 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (84, 'Ciorteanu',0759091177 , 2227 , TO_DATE('09/02/1991', 'DD/MM/YYYY'),48,52)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (85, 'Bucioaca',0761062974 , 2498 , TO_DATE('01/05/1999', 'DD/MM/YYYY'),49,61 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (86, 'Andone',0796064717 , 2388 , TO_DATE('13/05/1992', 'DD/MM/YYYY'),50,62 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (87, 'Marasescu',0747462138 , 2076 , TO_DATE('25/11/1994', 'DD/MM/YYYY'),51,71)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (88, 'Leahu',0711466302 , 2364 , TO_DATE('23/10/2004', 'DD/MM/YYYY'),52,72 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (89, 'Croitoru',0744489606 , 2620 , TO_DATE('14/02/1991', 'DD/MM/YYYY'),53,81)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (90, 'Stingă',0782137496 , 2772 , TO_DATE('15/11/1994', 'DD/MM/YYYY'),54,82)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (91, 'Comaneci',0716614905 , 2645 , TO_DATE('12/04/1998', 'DD/MM/YYYY'),55,91 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (92, 'Ciucă',0710579469 , 2685 , TO_DATE('01/01/2002', 'DD/MM/YYYY'),56,92)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (93, 'Doaga',0718380584 , 2040 , TO_DATE('06/10/2008', 'DD/MM/YYYY'),57,101 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (94, 'Gitui',0714857276 , 2690 , TO_DATE('03/04/2002', 'DD/MM/YYYY'),58,102 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (95, 'Ciopragă',0782892671 , 2449 , TO_DATE('16/04/2006', 'DD/MM/YYYY'),59,111)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (96, 'Craciun',0707192461 , 2283 , TO_DATE('25/03/2002', 'DD/MM/YYYY'),60,112 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (97, 'Ciocan',0782860726 , 2070 , TO_DATE('20/05/1997', 'DD/MM/YYYY'),61,121 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (98, 'Sandu',0757695389 , 2129 , TO_DATE('31/10/2003', 'DD/MM/YYYY'),62,122 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (99, 'Vilea',0730523804 , 2068 , TO_DATE('30/11/2002', 'DD/MM/YYYY'),63,131 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (100, 'Tapalaga',0731350190 , 2164 , TO_DATE('21/06/1999', 'DD/MM/YYYY'),64,132)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (101, 'Postolache',0742820716 , 2148 , TO_DATE('23/11/1995', 'DD/MM/YYYY'),65,141
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (102, 'Cozma',0790019181 , 2108 , TO_DATE('29/12/2012', 'DD/MM/YYYY'),66,142 )

```

```
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (103, 'Puscasu', 0779993825 , 2235 , TO_DATE('30/10/1997', 'DD/MM/YYYY'),67,151)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (104, 'Stochici', 0768111714 , 2074 , TO_DATE('11/11/2007', 'DD/MM/YYYY'),68,152)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (105, 'Ionita', 0757792978 , 2044 , TO_DATE('28/02/2002', 'DD/MM/YYYY'),69,161 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (106, 'Hanganu', 0787543807 , 3037 , TO_DATE('27/12/1995', 'DD/MM/YYYY'),70,162)
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (107, 'Racu', 0701416660 , 2453 , TO_DATE('28/02/2012', 'DD/MM/YYYY'),71,171 )
INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (108, 'Irimia', 0785471831 , 2434 , TO_DATE('13/05/1996', 'DD/MM/YYYY'),72,172)
SELECT * FROM dual;
select * from angajati;
commit;
```

```
INSERT ALL
INTO INFIRMIERE (id_infirmier, rol) --UPU
VALUES (73 , 'Infirmier')
INTO INFIRMIERE (id_infirmier, rol)
VALUES (74 , 'Ingrijitor')
INTO INFIRMIERE (id_infirmier, rol)
VALUES (75 , 'Brancardier')
INTO INFIRMIERE (id_infirmier, rol)
VALUES (76 , 'Brancardier')
INTO INFIRMIERE (id_infirmier, rol) --ATI
VALUES (77 , 'Infirmier')
INTO INFIRMIERE (id_infirmier, rol)
VALUES (78 , 'Ingrijitor')
INTO INFIRMIERE (id_infirmier, rol)
VALUES (79 , 'Ingrijitor')
INTO INFIRMIERE (id_infirmier, rol) --Laborator
VALUES (80 , 'Infirmier')
INTO INFIRMIERE (id_infirmier, rol) --Radiologie
VALUES (81 , 'Infirmier')
INTO INFIRMIERE (id_infirmier, rol)
VALUES (82 , 'Infirmier')
INTO INFIRMIERE (id_infirmier, rol) --Chirurgie
VALUES (83 , 'Infirmier')
INTO INFIRMIERE (id_infirmier, rol)
VALUES (84 , 'Brancardier')
INTO INFIRMIERE (id_infirmier, rol) --Ortopedie
VALUES (85 , 'Infirmier')
INTO INFIRMIERE (id_infirmier, rol)
VALUES (86 , 'Brancardier')
```

```

INTO INFIRMIERE (id_infirmier, rol) --Neurologie
VALUES (87 , 'Infirmier')
INTO INFIRMIERE (id_infirmier, rol)
VALUES (88 , 'Ingrijitor')
INTO INFIRMIERE (id_infirmier, rol) --Interne
VALUES (89 , 'Infirmier')
INTO INFIRMIERE (id_infirmier, rol)
VALUES (90 , 'Ingrijitor')
INTO INFIRMIERE (id_infirmier, rol) --Pneumo
VALUES (91 , 'Infirmier')
INTO INFIRMIERE (id_infirmier, rol)
VALUES (92 , 'Ingrijitor')
INTO INFIRMIERE (id_infirmier, rol) --Oncologie
VALUES (93 , 'Infirmier')
INTO INFIRMIERE (id_infirmier, rol)
VALUES (94 , 'Ingrijitor')
INTO INFIRMIERE (id_infirmier, rol) --Oftalmo
VALUES (95 , 'Infirmier')
INTO INFIRMIERE (id_infirmier, rol)
VALUES (96 , 'Ingrijitor')
INTO INFIRMIERE (id_infirmier, rol) --ORL
VALUES (97 , 'Infirmier')
INTO INFIRMIERE (id_infirmier, rol)
VALUES (98 , 'Ingrijitor')
INTO INFIRMIERE (id_infirmier, rol) --Pediatrie
VALUES (99 , 'Infirmier')
INTO INFIRMIERE (id_infirmier, rol)
VALUES (100 , 'Ingrijitor')
INTO INFIRMIERE (id_infirmier, rol) --Ginecologie
VALUES (101 , 'Infirmier')
INTO INFIRMIERE (id_infirmier, rol)
VALUES (102 , 'Ingrijitor')
INTO INFIRMIERE (id_infirmier, rol) --Neonatologie
VALUES (103 , 'Infirmier')
INTO INFIRMIERE (id_infirmier, rol)
VALUES (104 , 'Ingrijitor')
INTO INFIRMIERE (id_infirmier, rol) --Infectioase
VALUES (105 , 'Infirmier')
INTO INFIRMIERE (id_infirmier, rol)
VALUES (106 , 'Ingrijitor')
INTO INFIRMIERE (id_infirmier, rol) --Psihiatrie
VALUES (107 , 'Infirmier')
INTO INFIRMIERE (id_infirmier, rol)
VALUES (108 , 'Brancardier')

```

```

SELECT * FROM dual;
select * from infirmiere;
commit;

```

```

478 INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
479 VALUES (33, 'Pancescu',0744961280 ,5991 , TO_DATE('06/12/2014', 'DD/MM/YYYY'),null,161 )
480 INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
481 VALUES (34, 'Matescu',0758744671 ,6431 , TO_DATE('05/03/2012', 'DD/MM/YYYY'),null,162 )
482
483 INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
484 VALUES (35, 'Aldea',0702365674 ,5502 , TO_DATE('26/05/2017', 'DD/MM/YYYY'),null,171 )
485 INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
486 VALUES (36, 'Anghel',0748702976 ,5846 , TO_DATE('10/09/2003', 'DD/MM/YYYY'),null,172 )
487
488 SELECT * FROM dual;

```

36 rows inserted.

ID_ANGLER	NUME	TELEFON	SALARIU	DATA_ANGAJARE	ID_SEF	ID_SECTIE
37	37 Iacob	760058261	4775	14-JUN-92	1 1	
38	38 Mihai	762169693	4873	12-AUG-90	2 1	
39	39 Jitea	721540147	4988	05-MAY-90	3 1	
40	40 Copacel	711525821	4897	23-OCT-94	4 1	
41	41 Ratusanu	766210085	3641	03-SEP-00	5 2	
42	42 Hanganu	788844054	3577	12-JUN-92	6 2	
43	43 Manas	738474764	3867	20-APR-95	7 2	
44	44 Nechita	781258389	4232	30-JUL-13	8 3	
45	45 Ursu	732181446	4419	01-MAR-02	9 41	
46	46 Demsa	776407261	4306	29-MAY-14	10 42	

ID_DOCTOR	SPECIALIZARE
11	11 Chirurgie_generala
12	12 Chirurgie_cardiovasculara
13	13 Ortopedie_traumatologie
14	14 Ortopedie_pediatrica
15	15 Neurologie
16	16 Neurologie_pediatrica
17	17 Medicina_interna
18	18 Medicina_interna
19	19 Pneumologie
20	20 Pneumologie_pediatrica
21	21 Oncologie
22	22 Oncologie_pediatrica

ID_ASISTENT	STATUT
4	40 Generalist
5	41 Generalist_ATI
6	42 Generalist_ATI
7	43 Generalist_ATI
8	44 Laborator
9	45 Radiologie
10	46 Radiologie
11	47 Generalist_Bloc_Operator
12	48 Generalist_Bloc_Operator

ID_INFIRMIER	ROL
13	85 Infirmier
14	86 Brancardier
15	87 Infirmier
16	88 Ingrijitor
17	89 Infirmier
18	90 Ingrijitor
19	91 Infirmier
20	92 Ingrijitor
21	93 Infirmier

----- PACIENTI -----

INSERT ALL

```
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (1, 'Grecu', 0719657127, 'ADULT', 60, null, '1_A' ,TO_DATE('02/12/2020', 'DD/MM/YYYY'), TO_DATE('02/12/2020',
'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (2, 'Stroe', 0784582055, 'ADULT', 73 ,null, '1_A' ,TO_DATE('02/12/2020', 'DD/MM/YYYY'), TO_DATE('02/12/2020',
'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (3, 'Bunea', 0758008790, 'ADULT', 61 ,null, '1_B' ,TO_DATE('03/12/2020', 'DD/MM/YYYY'), TO_DATE('03/12/2020',
'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (4, 'Bucur', 0712484248, 'ADULT', 58 ,null, '1_B' ,TO_DATE('03/12/2020', 'DD/MM/YYYY'), TO_DATE('03/12/2020',
'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (5, 'Dumitru', 0700456274, 'ADULT', 71 , '1_Hidric', '2_A' ,TO_DATE('04/12/2020', 'DD/MM/YYYY'),TO_DATE('05/12/2020',
'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (6, 'Andrei', 0783912172, 'ADULT', 66 , '1_Hidric', '2_A' ,TO_DATE('04/12/2020', 'DD/MM/YYYY'),TO_DATE('05/12/2020',
'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (7, 'Savu', 0767705377, 'ADULT', 65 , '1_Hidric', '2_A' ,TO_DATE('06/12/2020', 'DD/MM/YYYY'),TO_DATE('07/12/2020',
'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (8, 'Turcu', 0718134895, 'ADULT', 68, '1_Hidric', '2_B' ,TO_DATE('06/12/2020', 'DD/MM/YYYY'),TO_DATE('07/12/2020',
'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (9, 'Necula', 0747646792, 'ADULT', 44, '4_Post_Operator', '51_A' ,TO_DATE('07/12/2020',
'DD/MM/YYYY'),TO_DATE('08/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (10, 'Cozma', 0712082458, 'ADULT', 34, '4_Post_Operator', '52_B' ,TO_DATE('07/12/2020',
'DD/MM/YYYY'),TO_DATE('08/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (11, 'Galod', 0724265883, 'ADULT', 36, '2_Renal', '61_B' ,TO_DATE('08/12/2020', 'DD/MM/YYYY'),TO_DATE('09/12/2020',
'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (12, 'Muntean', 0724501973, 'ADULT', 48, '5_Cortizon ', '62_A' ,TO_DATE('08/12/2020', 'DD/MM/YYYY'),TO_DATE('09/12/2020',
'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (13, 'Voicu', 0748544014, 'ADULT', 69, '6_Hepatic ', '71_A' ,TO_DATE('10/12/2020', 'DD/MM/YYYY'),TO_DATE('11/12/2020',
'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (14, 'Paduraru', 0733690452, 'ADULT', 65, '7_Lactofainos_Vegetarian', '72_A' ,TO_DATE('10/12/2020',
'DD/MM/YYYY'),TO_DATE('11/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (15, 'Trandafir', 0704589585, 'ADULT', 55, '3_Enterocolita', '81_A' ,TO_DATE('09/12/2020',
'DD/MM/YYYY'),TO_DATE('10/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (16, 'Dascalu', 0764345133, 'ADULT', 64, '1_Hidric', '81_B' ,TO_DATE('09/12/2020', 'DD/MM/YYYY'),TO_DATE('10/12/2020',
'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (17, 'Vasilescu', 0713958128, 'ADULT', 53, '2_Renal', '82_A' ,TO_DATE('10/12/2020', 'DD/MM/YYYY'),TO_DATE('11/12/2020',
'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (18, 'Diaconu', 0791305635, 'ADULT', 72, '6_Hepatic', '82_B' ,TO_DATE('10/12/2020', 'DD/MM/YYYY'),TO_DATE('11/12/2020',
'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
```

```

VALUES (19, 'Cretu', 0716194689, 'ADULT', 52, '9_Comun', '83_A' ,TO_DATE('12/12/2020', 'DD/MM/YYYY'),TO_DATE('13/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (20, 'Dragomirescu', 0703350719, 'ADULT', 66, '1_Hidric', '91_A' ,TO_DATE('12/12/2020', 'DD/MM/YYYY'),TO_DATE('13/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (21, 'Oancea', 0793016885, 'ADULT', 68, '4_Post_Operator', '92_A' ,TO_DATE('12/12/2020', 'DD/MM/YYYY'),TO_DATE('13/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (22, 'Radu', 0724034328, 'ADULT', 55, '8_Diabet', '102_A' ,TO_DATE('13/12/2020', 'DD/MM/YYYY'),TO_DATE('13/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (23, 'Achim', 0733161752, 'ADULT', 47, '9_Comun', '102_B' ,TO_DATE('13/12/2020', 'DD/MM/YYYY'),TO_DATE('13/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (24, 'Albu', 0784484586, 'ADULT', 46, '7_Lactofainos_Vegetarian', '112_A' ,TO_DATE('14/12/2020', 'DD/MM/YYYY'),TO_DATE('15/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (25, 'Preda', 0791797298, 'ADULT', 32, '9_Comun', '112_B' ,TO_DATE('14/12/2020', 'DD/MM/YYYY'),TO_DATE('15/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (26, 'Pavel', 0718119017, 'ADULT', 37, '9_Comun', '122_A' ,TO_DATE('15/12/2020', 'DD/MM/YYYY'),TO_DATE('16/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (27, 'Popescu', 0706002938, 'ADULT', 44, '7_Lactofainos_Vegetarian', '122_B' ,TO_DATE('15/12/2020', 'DD/MM/YYYY'),TO_DATE('16/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (28, 'Tanase', 0741746308, 'COPIL', 7, '5_Cortizon', '131_A' ,TO_DATE('16/12/2020', 'DD/MM/YYYY'),TO_DATE('17/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (29, 'Ion', 0792511103, 'COPIL', 2, '1_Hidric', '131_B' ,TO_DATE('17/12/2020', 'DD/MM/YYYY'),TO_DATE('18/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (30, 'Stanciu', 0734206439, 'COPIL', 5, '3_Enterocolita', '132_A' ,TO_DATE('17/12/2020', 'DD/MM/YYYY'),TO_DATE('18/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (31, 'Botezatu', 0745803076, 'SUGAR', 1, '1_Hidric', '132_B' ,TO_DATE('18/12/2020', 'DD/MM/YYYY'),TO_DATE('19/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (32, 'Coman', 0768758836, 'ADULT', 28, '4_Post_Operator', '142_A' ,TO_DATE('18/12/2020', 'DD/MM/YYYY'),TO_DATE('19/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (33, 'Radulescu', 0735602870, 'ADULT', 33, '9_Comun', '142_B' ,TO_DATE('18/12/2020', 'DD/MM/YYYY'),TO_DATE('19/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (34, 'Tanciu', 0744291543, 'SUGAR', 0.5, '1_Hidric', '152_A' ,TO_DATE('19/12/2020', 'DD/MM/YYYY'),TO_DATE('20/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (35, 'Marginean', 0783814491, 'SUGAR', 0.25, '1_Hidric', '152_B' ,TO_DATE('19/12/2020', 'DD/MM/YYYY'),TO_DATE('20/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (36, 'Gheorghiu', 0726479471, 'ADULT', 60, '1_Hidric', '161_A' ,TO_DATE('20/12/2020', 'DD/MM/YYYY'),TO_DATE('21/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (37, 'Stanescu', 0765497118, 'ADULT', 63, '2_Renal', '161_B' ,TO_DATE('20/12/2020', 'DD/MM/YYYY'),TO_DATE('21/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (38, 'Zaharia', 0716791660, 'ADULT', 54, '3_Enterocolita ', '162_A' ,TO_DATE('20/12/2020', 'DD/MM/YYYY'),TO_DATE('21/12/2020', 'DD/MM/YYYY'))

```

```

INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (39, 'Mihalache', 0713362774, 'ADULT', 61, '7_Lactofainos_Vegetarian', '172_A', TO_DATE('21/12/2020',
'DD/MM/YYYY'), TO_DATE('22/12/2020', 'DD/MM/YYYY'))
INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (40, 'Prodan', 0708561029, 'ADULT', 58, '8_Diabet', '172_B', TO_DATE('21/12/2020', 'DD/MM/YYYY'), TO_DATE('22/12/2020',
'DD/MM/YYYY'))
SELECT * FROM dual;
select * from PACIENTI;
commit;

```

ID_PACIENT	NUME	TELEFON	CATEGORIE	VARSTA	REGIM	ID_SALON	INTERNARE	EXTERNARE
1	1 Grecu	719657127	ADULT	60 (null)	1_A	02-DEC-20	02-DEC-20	
2	2 Stroe	784582055	ADULT	73 (null)	1_A	02-DEC-20	02-DEC-20	
3	3 Bunea	758008790	ADULT	61 (null)	1_B	03-DEC-20	03-DEC-20	
4	4 Bucur	712484248	ADULT	58 (null)	1_B	03-DEC-20	03-DEC-20	
5	5 Dumitru	700456274	ADULT	71	1_Hidric	2_A	04-DEC-20	05-DEC-20
6	6 Andrei	783912172	ADULT	66	1_Hidric	2_A	04-DEC-20	05-DEC-20
7	7 Savu	767705377	ADULT	65	1_Hidric	2_A	06-DEC-20	07-DEC-20
8	8 Turcu	718134895	ADULT	68	1_Hidric	2_B	06-DEC-20	07-DEC-20
9	9 Necula	747646792	ADULT	44	4_Post_Operator	51_A	07-DEC-20	08-DEC-20

----- INGRIJESC -----

INSERT ALL

```

INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (1, 37, TO_DATE('02/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (2, 38, TO_DATE('02/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (3, 39, TO_DATE('03/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (4, 40, TO_DATE('03/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (5, 41, TO_DATE('04/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (5, 41, TO_DATE('05/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (6, 41, TO_DATE('04/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (6, 41, TO_DATE('05/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (7, 42, TO_DATE('06/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (7, 42, TO_DATE('07/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (8, 43, TO_DATE('06/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (8, 43, TO_DATE('07/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (9, 47, TO_DATE('07/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (9, 47, TO_DATE('08/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (10, 48, TO_DATE('08/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (11, 49, TO_DATE('08/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (11, 49, TO_DATE('09/12/2020', 'DD/MM/YYYY') )

```



```

VALUES (30, 64, TO_DATE('17/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (30, 64, TO_DATE('18/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (31, 64, TO_DATE('18/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (31, 64, TO_DATE('19/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (32, 65, TO_DATE('18/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (33, 66, TO_DATE('19/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (34, 67, TO_DATE('19/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (35, 68, TO_DATE('19/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (36, 69, TO_DATE('20/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (36, 69, TO_DATE('21/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (37, 70, TO_DATE('20/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (37, 70, TO_DATE('21/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (38, 70, TO_DATE('20/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (38, 70, TO_DATE('21/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (39, 72, TO_DATE('21/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (39, 72, TO_DATE('22/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (40, 72, TO_DATE('21/12/2020', 'DD/MM/YYYY') )
INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
VALUES (40, 72, TO_DATE('22/12/2020', 'DD/MM/YYYY') )
SELECT * FROM dual;
select * from INGRIJESC;
commit;

```

ID_PACIENT	ID_ASISTENT	DATA_INGRIJIRE
60	38	70 21-DEC-20
61	39	72 21-DEC-20
62	39	72 22-DEC-20
63	40	72 21-DEC-20
64	40	72 22-DEC-20
65	41	70 21-DEC-20
66	41	72 21-DEC-20
67	41	72 22-DEC-20

ID_PACIENT	ID_ASISTENT	DATA_INGRIJIRE
1	41	67 02-JAN-21
2	41	67 03-JAN-21
3	41	68 02-JAN-21
4	41	70 21-DEC-20
5	41	72 21-DEC-20
6	41	72 22-DEC-20

----- TRATEAZA -----

```

INSERT ALL
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (1 ,1, 21, TO_DATE('02/12/2020', 'DD/MM/YYYY'))
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (2 ,2, 22, TO_DATE('02/12/2020', 'DD/MM/YYYY'))

```



```

VALUES (29 ,33, 33, TO_DATE('19/12/2020', 'DD/MM/YYYY'))
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (29 ,33, 36, TO_DATE('19/12/2020', 'DD/MM/YYYY'))
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (31 ,34, 35, TO_DATE('19/12/2020', 'DD/MM/YYYY'))
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (32 ,35, 31, TO_DATE('19/12/2020', 'DD/MM/YYYY'))
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (33 ,36, 12, TO_DATE('20/12/2020', 'DD/MM/YYYY'))
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (33 ,36, 2, TO_DATE('20/12/2020', 'DD/MM/YYYY'))
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (33 ,36, 12, TO_DATE('21/12/2020', 'DD/MM/YYYY'))
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (33 ,36, 2, TO_DATE('21/12/2020', 'DD/MM/YYYY'))
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (33 ,37, 17, TO_DATE('20/12/2020', 'DD/MM/YYYY'))
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (33 ,37, 2, TO_DATE('20/12/2020', 'DD/MM/YYYY'))
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (33 ,37, 17, TO_DATE('21/12/2020', 'DD/MM/YYYY'))
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (33 ,37, 2, TO_DATE('21/12/2020', 'DD/MM/YYYY'))
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (34 ,38, 13, TO_DATE('20/12/2020', 'DD/MM/YYYY'))
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (34 ,38, 3, TO_DATE('20/12/2020', 'DD/MM/YYYY'))
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (34 ,38, 13, TO_DATE('21/12/2020', 'DD/MM/YYYY'))
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (34 ,38, 3, TO_DATE('21/12/2020', 'DD/MM/YYYY'))
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (35 ,39, 54, TO_DATE('21/12/2020', 'DD/MM/YYYY'))
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (35 ,39, 54, TO_DATE('22/12/2020', 'DD/MM/YYYY'))
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (35 ,40, 52, TO_DATE('21/12/2020', 'DD/MM/YYYY'))
INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
VALUES (35 ,40, 52, TO_DATE('22/12/2020', 'DD/MM/YYYY'))
SELECT * FROM dual;
select * from TRATEAZA;
commit;

```

ID_DOCTOR	ID_PACIENT	ID_TRATAMENT	DATA_TRATAMENT
64	25	27	35 15-DEC-20
65	27	28	18 16-DEC-20
66	27	28	1 16-DEC-20
67	27	28	35 16-DEC-20
68	27	28	18 17-DEC-20
69	27	28	1 17-DEC-20
70	27	28	35 17-DEC-20
71	27	29	13 17-DEC-20
72	27	29	2 17-DEC-20
73	27	29	61 17-DEC-20

ID_DOCTOR	ID_PACIENT	ID_TRATAMENT	DATA_TRATAMENT
1	33	41	1 02-JAN-21
2	33	41	2 02-JAN-21
3	33	41	2 03-JAN-21
4	33	41	3 03-JAN-21
5	34	41	1 02-JAN-21
6	35	41	52 21-DEC-20
7	35	41	52 22-DEC-20
8	35	41	54 21-DEC-20
9	35	41	54 22-DEC-20

```

1335 VALUES (37, 'Stanescu', 0765497118, 'ADULT', 63, '1_Renal', '161_B', TO_DATE('20/12/2020', 'DD/MM/YYYY'), TO_DATE('21/12/2020', 'DD/MM/YYYY'))
1336 INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
1337 VALUES (38, 'Zaharia', 0716791660, 'ADULT', 54, '3_Enterocolita ', '162_A', TO_DATE('20/12/2020', 'DD/MM/YYYY'), TO_DATE('21/12/2020', 'DD/MM/YYYY'))
1338 INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
1339 VALUES (39, 'Mihalache', 0713362774, 'ADULT', 61, '7_Lactofainos_Vegetarian', '172_A', TO_DATE('21/12/2020', 'DD/MM/YYYY'), TO_DATE('22/12/2020', 'DD/MM/YYYY'))
1340 INTO PACIENTI (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
1341 VALUES (40, 'Prodan', 0708561029, 'ADULT', 58, '8_Diabet', '172_B', TO_DATE('21/12/2020', 'DD/MM/YYYY'), TO_DATE('22/12/2020', 'DD/MM/YYYY'))
1342
1343 SELECT * FROM dual;
1344 select * from PACIENTI;
1345 commit;
1346

```

Script Output | Query Result | Task completed in 0.046 seconds

40 rows inserted.

```

1501 INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
1502 VALUES (40, 72, TO_DATE('21/12/2020', 'DD/MM/YYYY') )
1503 INTO INGRIJESC (id_pacient, id_asistent, data_ingrijire)
1504 VALUES (40, 72, TO_DATE('22/12/2020', 'DD/MM/YYYY') )
1505
1506 SELECT * FROM dual;
1507 select * from INGRIJESC;
1508 commit;

```

Script Output | Query Result | Task completed in 0.05 seconds

64 rows inserted.

```

1759 INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
1760 VALUES (35 ,40, 52, TO_DATE('21/12/2020', 'DD/MM/YYYY'))
1761 INTO TRATEAZA (id_doctor, id_pacient, id_tratament, data_tratament)
1762 VALUES (35 ,40, 52, TO_DATE('22/12/2020', 'DD/MM/YYYY'))
1763
1764 SELECT * FROM dual;
1765 select * from TRATEAZA;
1766 commit;

```

Script Output | Query Result | Task completed in 0.06 seconds

110 rows inserted.

Ex 6. Definiți un subprogram stocat care să utilizeze **un tip de COLECTIE studiat**. Apelați subprogramul.

-- o procedura prin care se actualizeaza cu o valoare data ca parametru salariul unui angajat (doctor sau asistent) al carui nume este dat ca parametru:
-- > se va verifica daca angajatul este doar doctor sau asistent;
-- > daca sunt mai multi angajati care au acelasi nume, atunci se va afisa un mesaj corespunzator si de asemenea se va afisa lista acestora;
-- > daca nu exista angajati cu numele dat, atunci se va afisa un mesaj corespunzator;

```
DROP PROCEDURE marire_salariu;
/
CREATE OR REPLACE PROCEDURE marire_salariu (ang_nume angajati.nume%TYPE,
                                             nou_salariu angajati.salariu%TYPE)
IS
    ang_id angajati.id_angajat%TYPE := find_emp_id(ang_nume);
    doctor_id_minim doctori.id_doctor%TYPE;
    doctor_id_maxim doctori.id_doctor%TYPE;
    asistent_id_minim asistente.id_asistent%TYPE;
    asistent_id_maxim asistente.id_asistent%TYPE;

    SALARY_NOT_IN_RANGE EXCEPTION;
    PRAGMA exception_init(SALARY_NOT_IN_RANGE, -20002);

BEGIN
    SELECT min(id_doctor) -- 1
    INTO doctor_id_minim
    FROM doctori dr
        JOIN angajati ang ON (dr.id_doctor = ang.id_angajat);

    -----
    SELECT max(id_doctor) -- 36
    INTO doctor_id_maxim
    FROM doctori dr
        JOIN angajati ang ON (dr.id_doctor = ang.id_angajat);

    -----
    SELECT min(id_asistent) -- 37
    INTO asistent_id_minim
    FROM asistente a
        JOIN angajati ang ON (a.id_asistent = ang.id_angajat);

    -----
    SELECT max(id_asistent) -- 72
    INTO asistent_id_maxim
    FROM asistente a
        JOIN angajati ang ON (a.id_asistent = ang.id_angajat);

    --
    DBMS_OUTPUT.PUT_LINE( 'doctor_id_minim' || doctor_id_minim );
    DBMS_OUTPUT.PUT_LINE( 'doctor_id_maxim' || doctor_id_maxim );
    DBMS_OUTPUT.PUT_LINE( 'asistent_id_minim' || asistent_id_minim );
    DBMS_OUTPUT.PUT_LINE( 'asistent_id_maxim' || asistent_id_maxim );
```

```

IF not( (ang_id >= doctor_id_minim and ang_id <= doctor_id_maxim)
      or (ang_id >= asistent_id_minim and ang_id <= asistent_id_maxim) ) THEN
    RAISE SALARY_NOT_IN_RANGE;
END IF;

UPDATE angajati
SET salariu = nou_salariu
WHERE id_angajat = ang_id;

-- DBMS_OUTPUT.PUT_LINE ('Angajatul cu id ' || ang_id || ' are salariul modificat.' );

IF (ang_id >= doctor_id_minim and ang_id <= doctor_id_maxim) THEN
  DBMS_OUTPUT.PUT_LINE('Angajatul DOCTOR cu id: ' || ang_id || ' are salariul modificat: '
                      || nou_salariu || ' ron.');
ELSIF (ang_id >= asistent_id_minim and ang_id <= asistent_id_maxim) THEN
  DBMS_OUTPUT.PUT_LINE('Angajatul ASISTENT cu id: ' || ang_id || ' are salariul modificat: '
                      || nou_salariu || ' ron.');
END IF;

EXCEPTION
  WHEN SALARY_NOT_IN_RANGE THEN
    DBMS_OUTPUT.PUT_LINE('Salariul dat nu ii este atribuit unui doctor sau asistent! ');
    RAISE_APPLICATION_ERROR(-20002, 'Salariul dat nu ii este atribuit unui doctor sau asistent! ');

END marire_salariu;
/

```

```

DROP FUNCTION find_emp_id;
/
CREATE OR REPLACE FUNCTION find_emp_id(ang_nume angajati.nume%TYPE)
  RETURN angajati.id_angajat%TYPE
IS
  TYPE tabel_agă IS TABLE OF angajati.id_angajat%TYPE;
  ang tabel_agă;
  emp_id angajati.id_angajat%TYPE;

BEGIN
  SELECT id_angajat
  BULK COLLECT INTO ang
  FROM angajati
  WHERE UPPER(nume) = UPPER(ang_nume);

  IF ang.count = 0 THEN

```

```

        RAISE NO_DATA_FOUND;
ELSIF ang.count > 1 THEN
    RAISE TOO_MANY_ROWS;
END IF;

emp_id := ang(1);
RETURN emp_id;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista un angajat cu numele dat!');
        RAISE_APPLICATION_ERROR(-20000, 'Nu exista un angajat cu numele dat');

    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE('Mai multi angajati cu numele dat');

    FOR i IN ang.FIRST..ang.LAST LOOP
        DBMS_OUTPUT.PUT_LINE( i || ' Id angajat: ' || ang(i));
    END LOOP;

    RAISE_APPLICATION_ERROR(-20001, 'Mai multi angajati cu numele dat');

END find_emp_id;
/

```

The screenshot shows the Oracle SQL Developer interface with a script editor and a script output window.

Script Editor Content:

```

76     DBMS_OUTPUT.PUT_LINE('Salariul dat nu ii este atribuit unui doctor sau asistent!');
77     RAISE_APPLICATION_ERROR(-20002, 'Salariul dat nu ii este atribuit unui doctor sau asistent!');
78
79 END marire_salariu;
80 /
81
82 DROP FUNCTION find_emp_id;
83 /
84 CREATE OR REPLACE FUNCTION find_emp_id(ang_nume angajati.nume%TYPE)
85     RETURN angajati.id_angajat%TYPE
86 IS
87     TYPE tabel_agă IS TABLE OF angajati.id_angajat%TYPE;
88 
```

Script Output Window:

Script Output x

Task completed in 0.037 seconds

Function FIND_EMP_ID compiled

Procedure MARIRE_SALARIU compiled

```

select * from angajati where nume = 'Ciuchi'; --doctor
rollback;
BEGIN
    marire_salariu('Ciuchi',5010); -- salariu vechi: 5002
END;
/

```

```

select * from angajati where nume = 'Ghiuta'; --asistent
rollback;
BEGIN
    marire_salariu('Ghiuta',3510); -- salariu vechi: 3503
END;
/

```

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab contains the PL/SQL code. The 'Script Output' tab shows the message 'PL/SQL procedure successfully completed.' twice, once for each update. The 'Query Result' tab shows two messages: 'Angajatul DOCTOR cu id: 10 are salariul modificat: 5010 ron.' and 'Angajatul ASISTENT cu id: 60 are salariul modificat: 3510 ron.'

```

123
124 select * from angajati where nume = 'Ciuchi'; --doctor
125 rollback;
126 BEGIN
127     marire_salariu('Ciuchi',5010); -- salariu vechi: 5002
128 END;
129 /
130 -----
131 select * from angajati where nume = 'Ghiuta'; --asistent
132 rollback;
133 BEGIN
134     marire_salariu('Ghiuta',3510); -- salariu vechi: 3503
135 END;
136 /
137

```

```

select * from angajati where nume = 'Chiuca'; -- nu exista niciun angajat cu acest nume
BEGIN
    marire_salariu('Chiuca',3200); -- Eroare: Nu exista un angajat cu numele dat!
END;
/

```

Worksheet | Query Builder

```

135 END;
136 /
137 -----
138 select * from angajati where nume = 'Chiuca'; -- nu exista niciun angajat
139 BEGIN
140     marire_salariu('Chiuca',3200); -- Eroare: Nu exista un angajat cu nume
141 END;
142 /
143 -----
144 select * from angajati where nume = 'Hangantu'; -- 2 angajati cu acest nume

```

Script Output | Query Result | Task completed in 0.057 seconds

```

BEGIN
    marire_salariu('Chiuca',3200); -- Eroare: Nu exista un angajat cu numele dat!
END;
Error report -
ORA-20000: Nu exista un angajat cu numele dat!
ORA-06512: at "SYSTEM.FIND_EMP_ID", line 26
ORA-06512: at "SYSTEM.MARIRE_SALARIU", line 4
ORA-06512: at line 2
20000. 00000 - "%s"
*Cause: The stored procedure 'raise_application_error'
was called which causes this error to be generated.
*Action: Correct the problem as described in the error message or contact
the application administrator or DBA for more information.

```

```

select * from angajati where nume = 'Hangantu'; -- 2 angajati cu acest nume
BEGIN
    marire_salariu('Hangantu',3700);
    -- Eroare: Salariul dat nu ii este atribuit unui doctor sau asistent!
END ;
/

```

Worksheet | Query Builder

```

141 END;
142 /
143 -----
144 select * from angajati where nume = 'Hangantu'; -- 2 angajati cu acest nume
145 BEGIN
146     marire_salariu('Hangantu',3700);
147     -- Eroare: Salariul dat nu ii este atribuit unui doctor sau asistent!
148 END;
149 /
150

```

Script Output | Query Result | Task completed in 0.203 seconds

```

Error starting at line : 145 in command -
BEGIN
    marire_salariu('Hangantu',3700);
    -- Eroare: Salariul dat nu ii este atribuit unui doctor sau asistent!
END;
Error report -
ORA-20001: Mai multi angajati cu numele dat
ORA-06512: at "SYSTEM.FIND_EMP_ID", line 35
ORA-06512: at "SYSTEM.MARIRE_SALARIU", line 4
ORA-06512: at line 2

```

```
BEGIN
```

```
    marire_salariu('Racu',2500); -- Eroare: Salariul dat nu ii este atribuit unui doctor sau asistent!
```

```
END;
```

```
/
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a 'Worksheet' tab containing a PL/SQL block. The code is as follows:

```
147 -- Eroare: Salariul dat nu ii este atribuit unui doctor sau asistent!
148 END;
149 /
150
151 BEGIN
152     marire_salariu('Racu',2500); -- Eroare: Salariul dat nu ii este atribuit unui doctor sau asistent!
153 END;
154 /
155
```

In the top-right pane, titled 'Andreea_SGBD', the output of the script is displayed:

```
Angajatul DOCTOR cu id: 10 are salariul modificat: 5010 ron.
Angajatul ASISTENT cu id: 60 are salariul modificat: 3510 ron.
Mai multi angajati cu numele dat
1 Id angajat: 42
2 Id angajat: 106
Nu exista un angajat cu numele dat!
Salariul dat nu ii este atribuit unui doctor sau asistent!
```

Below the worksheet, there is a 'Script Output' tab showing the error message:

```
Action: Correct the problem as described in the error message or contact the application administrator or DBA for more information.

Error starting at line : 151 in command -
BEGIN
    marire_salariu('Racu',2500); -- Eroare: Salariul dat nu ii este atribuit unui doctor sau asistent!
Error report -
ORA-20002: Salariul dat nu ii este atribuit unui doctor sau asistent!
ORA-06512: at "SYSTEM.MARIRE_SALARIU", line 64
ORA-06512: at line 2
```

Ex 7. Definiți un subprogram stocat care să utilizeze **un tip de CURSOR studiat**. Apelați subprogramul.

-- o procedura care utilizeaza 2 cursoare:

-- (-> un cursor care obtine lista angajatilor care lucreaza pe o sectie al carei id este dat ca parametru;

-- -> un cursor care obtine lista tuturor sectiilor din spital;)

-- si obtine pt fiecare sectie: codul, numele acesteia si zona din care face parte

```
DROP PROCEDURE gaseste_ang;
```

```
/
```

```
CREATE OR REPLACE PROCEDURE gaseste_ang
IS
```

```
CURSOR c (sec_id sectii.id_sectie%TYPE)
```

```
    RETURN angajati%ROWTYPE
```

```
IS
```

```
    SELECT *
```

```
    FROM angajati
```

```
    WHERE id_sectie = sec_id;
```

```
-----
```

```
CURSOR d
```

```
    RETURN sectii%ROWTYPE
```

```
IS
```

```

SELECT *
FROM sectii;

-----  

nr number(4);
BEGIN
FOR i in d LOOP
    nr := 0;
    DBMS_OUTPUT.PUT_LINE ('Sectie: ' || i.nume || ' Zona: ' || i.zona || ' Id_sectie: ' || i.id_sectie);
    DBMS_OUTPUT.PUT_LINE ('-----');
    DBMS_OUTPUT.PUT_LINE ('Angajati: ' || ' Id_angajat | Nume_angajat:');
    DBMS_OUTPUT.PUT_LINE ('-----');

    FOR j IN c (i.id_sectie) LOOP
        DBMS_OUTPUT.PUT_LINE(j.id_angajat || ' ' || j.nume);
        nr := nr + 1;
    END LOOP;

    IF nr = 0 THEN
        DBMS_OUTPUT.PUT_LINE('--> Sectia nu are momentan angajati... ');
    END IF;

    DBMS_OUTPUT.PUT_LINE('');
END LOOP;
END;
/

```

```

BEGIN
    gaseste_ang;
END;
/

```

```

Andreea_SGBD
Sectie: UPU Zona: ROSIE Id_sectie: 1
-----
Angajati: Id_angajat | Nume_angajat:
-----
1 Seceleanu
2 Geru
3 Tarcuta
4 Marin
37 Iacob
38 Mihai
39 Jitea
40 Copacel
73 Adavanei
74 Duta
75 Alupei
76 Rogoza

Sectie: ATI Zona: ROSIE Id_sectie: 2
-----
Angajati: Id_angajat | Nume_angajat:
-----
5 Hirjanu
6 Zarzu
7 Dogaru
41 Ratusanu
42 Hanganu
43 Manas
77 Maran
78 Palaghita
79 Girileanu

Sectie: LABORATOR Zona: VERDE Id_sectie: 3
-----
Angajati: Id_angajat | Nume_angajat:
-----
8 Andronic
44 Nechita

```

Script Output X | Task completed in 0.054 seconds

Procedure GASESTE_ANG compiled

PL/SQL procedure successfully completed.

Compiler - Log

Messages | Statements | Compiler

```

-----CURSOR C: -----
DECLARE
    val varchar2(10);
    lista angajati%ROWTYPE;

    CURSOR c (sec_id sectii.id_sectie%TYPE)
    RETURN angajati%ROWTYPE
    IS
        SELECT *
        FROM angajati
        WHERE id_sectie = sec_id;
BEGIN
    val := '1';
    DBMS_OUTPUT.PUT_LINE('Lista ang care lucreaza pe sectia: ' || val);
    DBMS_OUTPUT.PUT_LINE('-----');
    FOR v_cursor IN c (val) LOOP
        DBMS_OUTPUT.PUT_LINE( 'Id_angajat: ' || v_cursor.id_angajat || ' -> Nume_angajat: ' || v_cursor.nume);
    END LOOP;
END;
/

```

Worksheet | Query Builder

```

-----CURSOR C: -----
DECLARE
    val varchar2(10);
    lista_angajati%ROWTYPE;

    CURSOR c (sec_id sectii.id_sectie%TYPE)
    RETURN angajati%ROWTYPE
    IS
        SELECT *
        FROM angajati
        WHERE id_sectie = sec_id;
BEGIN
    val := '1';
    DBMS_OUTPUT.PUT_LINE('Lista ang care lucreaza pe sectia: ' || val);
    DBMS_OUTPUT.PUT_LINE('-----');
    FOR v_cursor IN  c (val) LOOP
        DBMS_OUTPUT.PUT_LINE('Id_angajat: ' || v_cursor.id_angajat || ' -> Nume_angajat: '|| v_cursor.nume);
    END LOOP;
END;

```

Script Output X

```

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

-----cursor c: -----

```

```

-----cursor d: -----
DECLARE
    lista_sectii%ROWTYPE;

CURSOR d
RETURN sectii%ROWTYPE
IS
    SELECT *
    FROM sectii;
BEGIN
    FOR v_cursor IN  d () LOOP
        DBMS_OUTPUT.PUT_LINE( 'Id_sectie: ' || v_cursor.id_sectie || ' -> Nume_sectie: '|| v_cursor.nume);
        DBMS_OUTPUT.PUT_LINE( ' -> Categorie: ' || v_cursor.categorie || ' -> Zona: ' || v_cursor.zona );
        DBMS_OUTPUT.PUT_LINE('-----');
    END LOOP;
END;
/

```

```

Andreea_SGBD x
Id_sectie: 1      -> Nume_sectie: UPU
-> Categorie:   -> Zona: ROSIE
-----
Id_sectie: 2      -> Nume_sectie: ATI
-> Categorie:   -> Zona: ROSIE
-----
Id_sectie: 3      -> Nume_sectie: LABORATOR
-> Categorie:   -> Zona: VERDE
-----
Id_sectie: 41     -> Nume_sectie: RADIOLOGIE
-> Categorie:   -> Zona: ROSIE
-----
Id_sectie: 42     -> Nume_sectie: RADIOLOGIE
-> Categorie:   -> Zona: VERDE
-----
Id_sectie: 51     -> Nume_sectie: CHIRURGIE
-> Categorie: adulti -> Zona: ROSIE
-----
Id_sectie: 52     -> Nume_sectie: CHIRURGIE
-> Categorie: adulti -> Zona: VERDE
-----
Id_sectie: 53     -> Nume_sectie: CHIRURGIE
-> Categorie: adulti -> Zona: TAMPON
-----
Id_sectie: 61     -> Nume_sectie: ORTOPEDIE
-> Categorie: adulti -> Zona: ROSIE
-----
Id_sectie: 62     -> Nume_sectie: ORTOPEDIE
-> Categorie: adulti -> Zona: VERDE
-----
Id_sectie: 63     -> Nume_sectie: ORTOPEDIE
-> Categorie: adulti -> Zona: TAMPON
-----
Id_sectie: 71     -> Nume_sectie: NEUROLOGIE
-> Categorie: adulti -> Zona: ROSIE
-----
Id_sectie: 72     -> Nume_sectie: NEUROLOGIE

```

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

Ex 8. Definiți un subprogram stocat de tip FUNCTIE care să utilizeze 3 dintre tabelele definite.

Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

--Definiți un subprogram stocat de tip FUNCTIE care determine numarul de pacienti
-- care în prezent sunt internați într-un salon dat ca parametru.
--Detaliati informatiile referitoare la locatia acestor pacienti internati:
-- id-ul si numele acestora, sectia in care se afla salonul, si zona din care face parte sectia

```

DROP FUNCTION salon_nr_pacienti;
/
CREATE OR REPLACE FUNCTION salon_nr_pacienti
    (v_salon saloane.id_salon%TYPE)
RETURN NUMBER IS
    t saloane.id_salon%TYPE;
    nr_pac number;
BEGIN
    select id_salon into t
    from saloane
    where id_salon = v_salon;

    select COUNT(id_pacient)
    into nr_pac
    from pacienti p
        join saloane sal on p.id_salon = sal.id_salon
        join sectii sec on sal.id_sectie = sec.id_sectie
    where upper(sal.id_salon) = upper(v_salon);

```

```

dbms_output.put_line('Nr pacienti internati in salonul respectiv: ' || nr_pac);
dbms_output.put_line('-----');

FOR e IN (
    select p.id_pacient, p.nume pacient_nume, sec.id_sectie, sec.nume, sec.zona,
          sal.id_salon, sal.nr_nume
    from saloane sal
    join sectii sec on sec.id_sectie = sal.id_sectie
    join pacienti p on p.id_salon = sal.id_salon
    where upper(sal.id_salon) = upper(v_salon)      ) LOOP
DBMS_OUTPUT.PUT_LINE
( 'Id_pacient: ' || e.id_pacient || ' -> Nume_pacient: ' || e.pacient_nume );
DBMS_OUTPUT.PUT_LINE
( 'Id_salon: ' || e.id_salon || ' Nr_salon: ' || e.nr_nume );
DBMS_OUTPUT.PUT_LINE
( 'Id_sectie: ' || e.id_sectie || ' -> Nume sectie: ' || e.nume || ' -> Zona_sectiei: ' || e.zona );
DBMS_OUTPUT.PUT_LINE ('');
END LOOP;

DBMS_OUTPUT.PUT_LINE('');

IF nr_pac = 0 THEN
    insert into info_agă
    values (seq_info_agă.nextval, user, sysdate, 'SELECT', 0, 'Nu există PACIENTI în salonul dat ca parametru');
    dbms_output.put_line('Nu există PACIENTI în salonul dat ca parametru');
    return 0;
END IF;

insert into info_agă values (seq_info_agă.nextval, user, sysdate, 'SELECT', 1, null);
return nr_pac;

EXCEPTION
WHEN NO_DATA_FOUND THEN
    insert into info_agă
    values (seq_info_agă.nextval, user, sysdate, 'SELECT', -1, 'Nu există acest salon al sectiei cu codul dat');
    dbms_output.put_line('Nu există acest salon al sectiei cu codul dat');
    RAISE_APPLICATION_ERROR(-20000, 'Nu există salon cu numele dat');
    return -1;

WHEN TOO_MANY_ROWS THEN
-- nu va intra niciodata pe aceasta exceptie deoarece saloanele au id-ul unic
-- (este cheie primara) deci nu vor exista niciodata 2 saloane cu acelasi cod
    insert into info_agă
    values(seq_info_agă.nextval, user, sysdate, 'SELECT', -2, 'Există mai multe saloane cu codul dat');
    dbms_output.put_line(SQL%ROWCOUNT || 'Există mai multe saloane cu codul dat');
    RAISE_APPLICATION_ERROR(-20001, 'Există mai multe saloane cu codul dat');
    return -2;

```

```

WHEN OTHERS THEN
    insert into info_ag
    values(seq_info_ag.nextval, user, sysdate, 'SELECT', -3, 'Alta eroare!');
    dbms_output.put_line('Alta eroare!');
    RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
    return -3;

```

```

END salon_nr_pacienti;
/

```

```

CREATE OR REPLACE FUNCTION salon_nr_pacienti
    (v_salon saloane.id_salon%TYPE)
RETURNS NUMBER IS
    t saloane.id_salon%TYPE;
    nr_pac number;
BEGIN
    select id_salon into t
    from saloane
    where id_salon = v_salon;

    select COUNT(id_pacient)
    into nr_pac
    from pacienti p
    join saloane sal on p.id_salon = sal.id_salon
    join sectii sec on sal.id_sectie = sec.id_sectie
    ;

```

Script Output X | Task completed in 0.052 seconds

Function SALON_NR_PACIENTI compiled

```

DROP TABLE info_ag;
CREATE TABLE info_ag (
    id number(3)      primary key,
    utilizator        VARCHAR2(30),
    data              DATE,
    comanda           VARCHAR2(20),
    nr_linii          NUMBER(3),
    eroare            VARCHAR2(255)
);
COMMIT;

```

```

drop sequence seq_info_ag;
create sequence seq_info_ag
start with 1
increment by 1;

```

```

select seq_info_ag.nextval from dual;
select seq_info_ag.curval from dual;
select * from info_ag;

```

```

DECLARE
    val number;
BEGIN
    val := salon_nr_pacienti('1_A'); --Functioneaza
    IF val > 0 THEN
        dbms_output.put_line ('Rezultat: ' || val || ' pacienti internati in acest salon. ');
    ELSE
        dbms_output.put_line ('Rezultat: ' || val );
    END IF;
END;
/

```

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Worksheet' tab displays a PL/SQL block. This block first selects the current value of the sequence 'seq_info_ag.a.currrval' from the 'dual' table. Then, it declares a variable 'val' of type 'number'. Inside the 'BEGIN' block, 'val' is assigned the value returned by the function 'salon_nr_pacienti('1_A')'. A conditional statement checks if 'val' is greater than 0. If true, it outputs the result 'Rezultat: ' followed by 'val' and a space, then 'pacienti internati in acest salon. '. If false, it simply outputs 'Rezultat: ' followed by 'val'. The block ends with an 'END IF;' and an 'END;' statement, followed by a slash '/' on a new line.

The right side of the interface shows the 'Andreea_SGBD' session. It contains two tabs: 'Andreea_SGBD x' and 'Script Output x'. The 'Andreea_SGBD x' tab shows the execution results:

```

Nr pacienti internati in salonul respectiv: 2
Id_pacient: 1 -> Nume_pacient: Grecu
Id_salon: 1_A           Nr_salon: A
Id_sectie: 1            -> Nume sectie: UPU -> Zona_sectiei: ROSIE

Id_pacient: 2 -> Nume_pacient: Stroe
Id_salon: 1_A           Nr_salon: A
Id_sectie: 1            -> Nume sectie: UPU -> Zona_sectiei: ROSIE

Rezultat: 2 pacienti internati in acest salon.

```

The 'Script Output x' tab shows the compilation and execution status:

```

Function SALON_NR_PACIENTII compiled
PL/SQL procedure successfully completed.

```

```

DECLARE
    val number;
BEGIN
    val := salon_nr_pacienti('1_AAA'); --Nu exista acest salon...
    IF val > 0 THEN
        dbms_output.put_line ('Rezultat: ' || val || ' pacienti internati in acest salon.');
    ELSE
        dbms_output.put_line ('Rezultat: ' || val );
    END IF;
END;
/

```

The screenshot shows a PL/SQL script in the Worksheet tab:

```

DECLARE
  val number;
BEGIN
  val :=    salon_nr_pacienti('1_AAA'); --Nu exista acest salon...
  IF val > 0 THEN
    dbms_output.put_line ('Rezultat: ' || val || ' pacienti internati in acest salon.');
  ELSE
    dbms_output.put_line ('Rezultat: ' || val );
  END IF;
END;
/

```

The output in the Script Output tab shows the execution results and an error report:

```

Script Output x Query Result x
Task completed in 0.036 seconds
dbms_output.put_line ('Rezultat: ' || val || ' pacienti internati in acest salon.');
ELSE
dbms_output.put_line ('Rezultat: ' || val );
END IF;
END;

Error report -
ORA-20000: Nu exista salon cu numele dat
ORA-06512: at "SYSTEM.SALON_NR_PACIENTI", line 55
ORA-06512: at line 4
20000. 00000 - "%s"
*Cause: The stored procedure 'raise_application_error'
  was called which causes this error to be generated.
*Action: Correct the problem as described in the error message or contact
  the application administrator or DBA for more information.

```

```

select * from info_ag;
select * from saloane;
INSERT INTO saloane (id_salon, nr_nume, id_sectie) VALUES ('174_A', 'A', 173 );

```

```

DECLARE
  val number;
BEGIN
  val :=    salon_nr_pacienti('174_A'); --Salonul nou adaugat nu are pacienti
  IF val > 0 THEN
    dbms_output.put_line ('Rezultat: ' || val || ' pacienti internati in acest salon.');
  ELSE
    dbms_output.put_line ('Rezultat: ' || val );
  END IF;
END;
/

```

```

select * from info_ag;
DELETE FROM saloane WHERE id_salon = '174_A' ;

```

Worksheet | Query Builder | Andreea_SGBD

```

select * from saloane;
INSERT INTO saloane (id_salon, nr_numar, id_sectie) VALUES ('174_A', 'A', 173);

--DECLARE
val number;
BEGIN
val :=    salon_nr_pacient('174_A'); --Salonul nou adaugat nu are pacienti
IF val > 0 THEN
dbms_output.put_line ('Rezultat: ' || val || ' pacienti internati in acest salon.');
ELSE
dbms_output.put_line ('Rezultat: ' || val );
END IF;
END;
/
select * from info_agă;
DELETE FROM saloane WHERE id_salon = '174_A' ;

```

Script Output | Task completed in 0.031 seconds

1 row inserted.

PL/SQL procedure successfully completed.

```

END;
/
select * from info_agă;
DELETE FROM saloane WHERE id_salon = '174_A' ;

```

Script Output | Query Result | All Rows Fetched: 10 in 0.003 seconds

ID	UTILIZATOR	DATA	COMANDA	NR_LINII	EROARE
1	1 SYSTEM	24-DEC-20	SELECT	1 (null)	
2	4 SYSTEM	24-DEC-20	SELECT	-1	Nu exista acest salon al sectiei cu codul dat
3	5 SYSTEM	24-DEC-20	SELECT	-1	Nu exista acest salon al sectiei cu codul dat
4	6 SYSTEM	27-DEC-20	SELECT	1 (null)	
5	9 SYSTEM	30-DEC-20	SELECT	1 (null)	
6	10 SYSTEM	30-DEC-20	SELECT	1 (null)	
7	12 SYSTEM	30-DEC-20	SELECT	0	Nu exista PACIENTI in salonul dat ca parametru
8	13 SYSTEM	30-DEC-20	SELECT	0	Nu exista PACIENTI in salonul dat ca parametru
9	18 SYSTEM	02-JAN-21	SELECT	1 (null)	

Ex 9. Definiți un **subprogram stocat de tip PROCEDURA** care să utilizeze **5** dintre **tabelele definite**.

Tratați toate exceptiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

--Definiți un subprogram stocat de tip PROCEDURA care să realizeze un buletin cu toate informațiile pacientului.
--Se vor afisa informații referitoare la codul doctorului ce l-a tratat, data internarii și externării,
--codul tratamentului administrat și codul asistentei ce l-a îngrijit, dar și date referitoare la
--locatia pacientului internat în salon, respectiv sectia din care face parte salonul.

```

DROP PROCEDURE detalii_pacienti;
/
CREATE OR REPLACE PROCEDURE PROCEDURE detalii_pacienti (pac_nume pacienti.nume%TYPE)
IS
    pac_id pacienti.id_pacient%TYPE := find_pac_id(pac_nume);
BEGIN
DBMS_OUTPUT.PUT_LINE ('Pacientul ' || ' detine urmatoarele info: ');
DBMS_OUTPUT.PUT_LINE ('');
DBMS_OUTPUT.PUT_LINE ('Id_pacient: ' || pac_id || ' -> nume_pacient: ' || pac_nume );
DBMS_OUTPUT.PUT_LINE ('-----');
DBMS_OUTPUT.PUT_LINE ('');

FOR e IN (
    select p.id_pacient, p.nume patient_nume,
           t.id_doctor, t.id_tratament, t.data_tratament, p.internare, p.externare,
           i.id_asistent, i.data_ingrijire,
           sal.id_salon, sal.nr_nume,
           sec.id_sectie, sec.nume sectie_nume, sec.zona
      from pacienti p
     join trateaza t on t.id_pacient = p.id_pacient
     join ingrijesc i on i.id_pacient = p.id_pacient
     join saloane sal on sal.id_salon = p.id_salon
     join sectii sec on sec.id_sectie = sal.id_sectie
     where p.id_pacient = pac_id and t.data_tratament = i.data_ingrijire ) LOOP

    DBMS_OUTPUT.PUT_LINE ( 'DATA: ' || e.data_ingrijire );
    DBMS_OUTPUT.PUT_LINE ( 'Id_doctor: ' || e.id_doctor );
    DBMS_OUTPUT.PUT_LINE ( 'Id_asistent: ' || e.id_asistent);

    DBMS_OUTPUT.PUT_LINE ( ' -> id_tratament: ' || e.id_tratament );
    DBMS_OUTPUT.PUT_LINE ( ' -> internare: ' || e.internare || ' -> externare: ' || e.externare );

    DBMS_OUTPUT.PUT_LINE ( ' -> Id_salon: ' || e.id_salon || ' -> nr_salon: ' || e.nr_nume );
    DBMS_OUTPUT.PUT_LINE ( ' -> Id_sectie: ' || e.id_sectie || ' -> nume_sectie: ' || e.sectie_nume
                           || ' -> zona_sectiei: ' || e.zona );
    DBMS_OUTPUT.PUT_LINE ('-----');
    DBMS_OUTPUT.PUT_LINE ('');

END LOOP;

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Alta eroare!');
        RAISE_APPLICATION_ERROR(-20003,'Alta eroare!');

END detalii_pacienti;
/

```

```

DROP FUNCTION find_pac_id;
/
CREATE OR REPLACE FUNCTION find_pac_id (pac_nume pacienti.nume%TYPE)
    RETURN pacienti.id_pacient%TYPE
IS
    TYPE tabel_agă IS TABLE OF pacienti.id_pacient%TYPE;
    pac tabel_agă;
    pac_id pacienti.id_pacient%TYPE;
BEGIN
    SELECT id_pacient
    BULK COLLECT INTO pac
    FROM pacienti
    WHERE UPPER(nume) = UPPER(pac_nume);

    IF pac.count = 0 THEN
        RAISE NO_DATA_FOUND;
    ELSIF pac.count > 1 THEN
        RAISE TOO_MANY_ROWS;
    END IF;

    pac_id := pac(1);
    RETURN pac_id;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE ('Nu exista un pacient cu numele dat!');
        RAISE_APPLICATION_ERROR (-20000, 'Nu exista un pacient cu numele dat');

    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE ('Mai multi pacienti cu numele dat');

        FOR i IN pac.FIRST..pac.LAST LOOP
            DBMS_OUTPUT.PUT_LINE ( i || ' Id pacient: ' || pac(i));
        END LOOP;

        RAISE_APPLICATION_ERROR (-20001, 'Mai multi pacienti cu numele dat');
END find_pac_id;
/

```

```

        DDMSO_OUIPOI.FOI_DINE\ Alta eroare: /
        RAISE_APPLICATION_ERROR(-20003,'Alta eroare!');

    END detalii_pacienti;
    /

```

```

DROP FUNCTION find_pac_id;
/
CREATE OR REPLACE FUNCTION find_pac_id (pac_nume_pacienti.nume%TYPE)
RETURN pacienti.id_pacient%TYPE
IS
    TYPE tabel_agă IS TABLE OF pacienti.id_pacient%TYPE;
    pac tabel_agă;
    pac_id pacienti.id_pacient%TYPE;
BEGIN
    SELECT id_pacient
    BULK COLLECT INTO pac
    FROM pacienti

```

Script Output X

Task completed in 0.034 seconds

Function FIND_PAC_ID compiled

Procedure DETALII_PACIENTI compiled

```
select * from trateaza;
select * from ingrijesc;
```

```
select * from pacienti where nume = 'Andrei';
BEGIN
    detalii_pacienti('Andrei');
END;
```

```
/
```

Worksheet Query Builder

```

    END LOOP;

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

    END find_pac_id;
    /

```

```

select * from trateaza;
select * from ingrijesc;

select * from pacienti where nume = 'Andrei';
BEGIN
    detalii_pacienti('Andrei');
END;
/

```

```

select * from pacienti where nume = 'Breaca'; -- nu exista niciun angajat cu acest nume
BEGIN

```

Script Output X

Task completed in 0.034 seconds

Procedure DETALII_PACIENTI compiled

PL/SQL procedure successfully completed.

Pacientul detine urmatoarele info:

```

Id_pacient: 6 -> nume_pacient: Andrei
-----
```

```

DATA: 04-DEC-20
Id_doctor: 6
Id_asistent: 41
-> id_tratament: 12
-> internare: 04-DEC-20 -> externare: 05-DEC-20
> Id_salon: 2_A      -> nr_salon: A
> Id_sectie: 2       -> nume_sectie: ATI -> zona_sectiei: ROSIE
-----
```

```

DATA: 04-DEC-20
Id_doctor: 6
Id_asistent: 41
-> id_tratament: 2
-> internare: 04-DEC-20 -> externare: 05-DEC-20
> Id_salon: 2_A      -> nr_salon: A
> Id_sectie: 2       -> nume_sectie: ATI -> zona_sectiei: ROSIE
-----
```

```

DATA: 05-DEC-20
Id_doctor: 6
Id_asistent: 41
-> id_tratament: 12
-> internare: 04-DEC-20 -> externare: 05-DEC-20
> Id_salon: 2_A      -> nr_salon: A
> Id_sectie: 2       -> nume_sectie: ATI -> zona_sectiei: ROSIE
-----
```

```

DATA: 05-DEC-20
Id_doctor: 6
Id_asistent: 41
-> id_tratament: 2
-> internare: 04-DEC-20 -> externare: 05-DEC-20

```

```

select * from pacienti where nume = 'Breaca'; -- nu exista niciun angajat cu acest nume
BEGIN
    detalii_pacienti('Breaca'); -- Eroare: Nu exista un angajat cu numele dat!
END;
/

```

The screenshot shows the Oracle SQL Developer interface. In the top right, a message window titled "Andreea_SGBD" displays the error message: "Nu exista un pacient cu numele dat!". The main area shows a query in the "Worksheet" tab:

```

END;
/
-----  

select * from pacienti where nume = 'Breaca'; -- nu exista niciun angajat cu acest nume
BEGIN
    detalii_pacienti('Breaca'); -- Eroare: Nu exista un angajat cu numele dat!
END;
/

```

In the bottom left, the "Script Output" tab shows the execution results and errors:

```

Error starting at line : 511 in command -
BEGIN
    detalii_pacienti('Breaca'); -- Eroare: Nu exista un angajat cu numele dat!
END;
Error report -
ORA-20000: Nu exista un pacient cu numele dat!
ORA-06512: at "SYSTEM.FIND_PAC_ID", line 25
ORA-06512: at "SYSTEM.DETALII_PACIENTI", line 3
ORA-06512: at line 2
20000. 00000 - "%s"
*Cause:  The stored procedure 'raise_application_error'
          was called which causes this error to be generated.
*Action: Correct the problem as described in the error message or contact
          the application administrator or DBA for more information.

```

```

INSERT INTO pacienti (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (41, 'Stroe', 0717994689, 'ADULT', 62, null, '1_A',
        TO_DATE('21/12/2020', 'DD/MM/YYYY'),TO_DATE('22/12/2020', 'DD/MM/YYYY') );

```

```

select * from pacienti where nume = 'Stroe'; -- Mai multi pacienti cu numele dat
BEGIN
    detalii_pacienti('Stroe'); -- Eroare: Mai multi pacienti cu numele dat
END;
/

```

SQL Worksheet | Query Builder

```

2292 END;
2293 /
2294
2295 INSERT INTO pacienti (id_pacient, nume, telefon, categorie, varsta, regim, id_salon
2296 VALUES (41, 'Stroe', 0717994689, 'ADULT', 62, null, '1_A', TO_DATE('21/12/2020', 'DD
2297
2298 select * from pacienti where nume = 'Stroe'; -- Mai multi pacienti cu numele dat
2299 BEGIN
2300     detalii_pacienti('Stroe'); -- Eroare: Mai multi pacienti cu numele dat
2301 END;
2302 /
2303

```

Script Output | Query Result | Task completed in 0.036 seconds

```

BEGIN
    detalii_pacienti('Stroe'); -- Eroare: Mai multi pacienti cu numele dat
END;
Error report -
ORA-20001: Mai multi pacienti cu numele dat
ORA-06512: at "SYSTEM.FIND_PAC_ID", line 34
ORA-06512: at "SYSTEM.DETALII_PACIENTI", line 3
ORA-06512: at line 2

```

DELETE FROM pacienti WHERE id_pacient = 41;

SQL Worksheet | History

Worksheet | Query Builder

```

select * from pacienti where nume = 'Stroe'; -- Mai multi pacienti cu numele dat
BEGIN
    detalii_pacienti('Stroe'); -- Eroare: Mai multi pacienti cu numele dat
END;
/

```

```

DELETE FROM pacienti WHERE id_pacient = 41;

```

Script Output | Query Result | Task completed in 0.027 seconds

```

detalii_pacienti('Stroe'); -- Eroare: Mai multi pacienti cu numele dat
END;
Error report -
ORA-20001: Mai multi pacienti cu numele dat
ORA-06512: at "SYSTEM.FIND_PAC_ID", line 34
ORA-06512: at "SYSTEM.DETALII_PACIENTI", line 3
ORA-06512: at line 2

1 row deleted.

PL/SQL procedure successfully completed.

```

Andreea_SGBD x

```

Mai multi pacienti cu numele dat
1 Id pacient: 2
2 Id pacient: 41

Pacientul detine urmatoarele info:
Id_pacient: 2 -> nume_pacient: Stroe
-----  

DATA: 02-DEC-20
Id_doctor: 2
Id_asistent: 38
-> id_tratament: 22
-> internare: 02-DEC-20 -> externare: 02-DEC-20
> Id_salon: 1_A          -> nr_salon: A
> Id_sectie: 1           -> nume_sectie: UPU -> zona_sectiei: ROSIE
-----  


```

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

-- Definiti un declansator la nivel de comanda care sa actualizeze numarul pacientilor internati in fiecare zona
-- Stocati aceste variabile intr-un pachet nou.

```
DROP PACKAGE pacienti_zone_spital;
/
CREATE OR REPLACE PACKAGE pacienti_zone_spital
AS
    pac_zrosie number;
    pac_zverde number;
    pac_ztampon number;
END pacienti_zone_spital;
/
-----
DROP TRIGGER tr_nr_pac_zone_spital;
/
CREATE OR REPLACE TRIGGER tr_nr_pac_zone_spital
AFTER INSERT OR UPDATE OR DELETE ON pacienti
BEGIN
    SELECT COUNT(*)
    INTO pacienti_zone_spital.pac_zrosie
    FROM (
        select p.id_pacient
        from sectii sec
        join saloane sal on sec.id_sectie = sal.id_sectie
        join pacienti p on p.id_salon = sal.id_salon
        where zona = 'ROSIE'  );
    SELECT COUNT(*)
    INTO pacienti_zone_spital.pac_zverde
    FROM (
        select p.id_pacient
        from sectii sec
        join saloane sal on sec.id_sectie = sal.id_sectie
        join pacienti p on p.id_salon = sal.id_salon
        where zona = 'VERDE'  );
    SELECT COUNT(*)
    INTO pacienti_zone_spital.pac_ztampon
    FROM (
        select p.id_pacient
        from sectii sec
        join saloane sal on sec.id_sectie = sal.id_sectie
        join pacienti p on p.id_salon = sal.id_salon
        where zona = 'TAMPON'  );
END;
/
```

```

BEGIN
    DBMS_OUTPUT.PUT_LINE( 'Nr pacienti in zona ROSIE: ' || pacienti_zone_spital.pac_zrosie);
    DBMS_OUTPUT.PUT_LINE( 'Nr pacienti in zona VERDE: ' || pacienti_zone_spital.pac_zverde);
    DBMS_OUTPUT.PUT_LINE( 'Nr pacienti in zona TAMPON: ' || pacienti_zone_spital.pac_ztampon);
END;
/

```

The screenshot shows the Oracle SQL Developer interface. In the top right, there's a window titled "Andreea_SGBD" displaying the output of the PL/SQL code: "Nr pacienti in zona ROSIE: 18", "Nr pacienti in zona VERDE: 21", and "Nr pacienti in zona TAMPON: 1". Below this, the "Script Output" tab shows the results of the query: "Trigger TR_NR_PAC_ZONE_SPITAL compiled" and "PL/SQL procedure successfully completed.".

```

select COUNT(*)
from (
    select p.id_pacient, p.nume pacient_nume, sec.id_sectie, sec.nume, sec.zona, sal.id_salon, sal.nr_nume
    from sectii sec
        join saloane sal on sec.id_sectie = sal.id_sectie
        join pacienti p on p.id_salon = sal.id_salon
    where zona = 'TAMPON' ); --1 pacient in zona TAMPON in acest moment

```

```

INSERT INTO pacienti (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (41, 'Creval', 0717994689, 'ADULT', 62, '9_Cunun', '83_A',
        TO_DATE('21/12/2020', 'DD/MM/YYYY'), TO_DATE('22/12/2020', 'DD/MM/YYYY') );
BEGIN
    DBMS_OUTPUT.PUT_LINE( 'Nr pacienti in zona TAMPON: ' || pacienti_zone_spital.pac_ztampon);
END;
/

```

Worksheet Query Builder

```

3216 select COUNT(*)
3217 from (
3218 select p.id_pacient, p.nume pacient_nume, sec.id_sectie, sec.nume, sec.zona, sal.id_salon, sal.nr_nume
3219 from sectii sec
3220 join saloane sal on sec.id_sectie = sal.id_sectie
3221 join pacienti p on p.id_salon = sal.id_salon
3222 where zona = 'TAMPON'
3223 ); --l pacient in zona TAMPON in acest moment
3224
3225 INSERT INTO pacienti (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
3226 VALUES (41, 'Creval', 0717994689, 'ADULT', 62, '9_Cunun', '83_A' ,TO_DATE('21/12/2020', 'DD/MM/YYYY'),TO_I
3227
3228 BEGIN
3229 DBMS_OUTPUT.PUT_LINE( 'Nr pacienti in zona| TAMPON: ' || pacienti_zone_spital.pac_ztampon);
3230 END;

```

Script Output x | Query Result x

PL/SQL procedure successfully completed.

Task completed in 0.034 seconds

PL/SQL procedure successfully completed.

Nr pacienti in zona ROSIE: 18
Nr pacienti in zona VERDE: 21
Nr pacienti in zona TAMPON: 1
Nr pacienti in zona TAMPON: 2

DELETE FROM pacienti WHERE id_pacient = 41;

BEGIN

```

DBMS_OUTPUT.PUT_LINE( 'Nr pacienti in zona TAMPON: ' || pacienti_zone_spital.pac_ztampon);
END;
/

```

Worksheet Query Builder

```

3226 VALUES (41, 'Creval', 0717994689, 'ADULT', 62, '9_Cunun', '83_A' ,TO_DATE('21/12/2020', 'DD/MM/YYYY'),TO_I
3227
3228 BEGIN
3229 DBMS_OUTPUT.PUT_LINE( 'Nr pacienti in zona TAMPON: ' || pacienti_zone_spital.pac_ztampon);
3230 END;
3231 /
3232
3233 DELETE FROM pacienti WHERE id_pacient = 41;
3234
3235 BEGIN
3236 DBMS_OUTPUT.PUT_LINE( 'Nr pacienti in zona TAMPON: ' || pacienti_zone_spital.pac_ztampon);
3237 END;
3238 /
3239 -----
3240

```

Script Output x | Query Result x

PL/SQL procedure successfully completed.

Task completed in 0.025 seconds

PL/SQL procedure successfully completed.

Nr pacienti in zona ROSIE: 18
Nr pacienti in zona VERDE: 21
Nr pacienti in zona TAMPON: 1
Nr pacienti in zona TAMPON: 2
Nr pacienti in zona TAMPON: 1

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

-- Definiti un declansator cu ajutorul caruia sa se implementeze restrictia conform careia
-- intr-o sectie nu pot lucra mai mult de 12 persoane.

```
DROP TRIGGER tr_limit_nr_ang_on_sectii;
/
CREATE OR REPLACE TRIGGER tr_limit_nr_ang_on_sectii
    BEFORE INSERT ON angajati
FOR EACH ROW
DECLARE
    -- Nr de angajati din sectia in care s-ar insera
    nr_ang NUMBER;

    NO_SPACE EXCEPTION;
    PRAGMA exception_init(NO_SPACE, -20002);
BEGIN
    SELECT COUNT(1)
        INTO nr_ang
        FROM angajati
        WHERE :NEW.id_sectie = id_sectie;

    IF nr_ang = 12 THEN
        RAISE NO_SPACE;
    END IF;

EXCEPTION
    WHEN NO_SPACE THEN
        DBMS_OUTPUT.PUT_LINE('Aceasta sectie are deja nr maxim de angajati.');
        RAISE_APPLICATION_ERROR(-20002, 'Aceasta sectie are deja nr maxim de angajati.');
END;
/
```

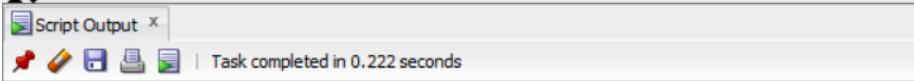
Worksheet Query Builder

```

DROP TRIGGER tr_limit_nr_ang_on_sectii;
/
CREATE OR REPLACE TRIGGER tr_limit_nr_ang_on_sectii
    BEFORE INSERT ON angajati
FOR EACH ROW
DECLARE
    -- Nr de angajati din sectia in care s-ar insera
    nr_ang NUMBER;

    NO_SPACE EXCEPTION;
    PRAGMA exception_init(NO_SPACE, -20002);
BEGIN

```



Trigger TR_LIMIT_NR_ANG_ON_SECTII compiled

```

SELECT id_sectie, COUNT(*) AS nr_ang -- care sectii sunt deja pline
FROM angajati
GROUP BY id_sectie
HAVING COUNT(*) >= 12;

```

```

select id_sectie, id_angajat
from angajati
where id_sectie = 1;

```

```

-- Eroare deoarece sunt deja prea multi angajati in sectia respectiva
INSERT INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (109, 'Irimiaca', 0785771821, 3734, TO_DATE('13/05/1996', 'DD/MM/YYYY'), null, 1);
/

```

```

Worksheet | Query Builder
239 SELECT id_sectie, COUNT(*) AS nr_ang
240 FROM angajati
241 GROUP BY id_sectie
242 HAVING COUNT(*) >= 12;
243
244 select id_sectie, id_angajat
245 from angajati
246 where id_sectie = 1;
247
248 -- Eroare deoarece sunt deja prea multi angajati in sectia respectiva
249 INSERT INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
250 VALUES (109, 'Irimiaca',0785771821 , 3734 , TO_DATE('13/05/1996', 'DD/MM/YYYY'),null,1);
251 /
252

```

Script Output X | Query Result X

Task completed in 0.037 seconds

PL/SQL procedure successfully completed.

Error starting at line : 249 in command -
 INSERT INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
 VALUES (109, 'Irimiaca',0785771821 , 3734 , TO_DATE('13/05/1996', 'DD/MM/YYYY'),null,1)
 Error report -
 ORA-20002: Aceasta sectie are deja nr maxim de angajati.
 ORA-06512: at "SYSTEM.TR_LIMIT_NR_ANG_ON_SECTII", line 20
 ORA-04088: error during execution of trigger 'SYSTEM.TR_LIMIT_NR_ANG_ON_SECTII'

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

-- Definiti un declansator care sa introduca date intr-un tabel nou creat si sa afiseze in consola aceste informatii
-- dupa ce utilizatorul a folosit o comanda LDD (declansator sistem - la nivel de schema).

```

DROP TABLE audit_spital;
CREATE TABLE audit_spital
    (utilizator      VARCHAR2(30),
     nume_bd        VARCHAR2(50),
     eveniment      VARCHAR2(20),
     nume_object   VARCHAR2(30),
     data           DATE );

```

```

DROP TRIGGER tr_audit_spital;
/
CREATE OR REPLACE TRIGGER tr_audit_spital
    AFTER CREATE OR DROP OR ALTER ON SCHEMA
BEGIN

```

```

    INSERT INTO audit_spital
    VALUES ( SYS.LOGIN_USER,  SYS.DATABASE_NAME,  SYS.SYSEVENT,
             SYS.DICTIONARY_OBJ_NAME,  SYSDATE);

```

```

    DBMS_OUTPUT.PUT_LINE( 'Utilizator: ' || SYS.LOGIN_USER );
    DBMS_OUTPUT.PUT_LINE( 'Nume_bd: ' || SYS.DATABASE_NAME );
    DBMS_OUTPUT.PUT_LINE( 'Eveniment: ' || SYS.SYSEVENT );

```

```

DBMS_OUTPUT.PUT_LINE( 'Nume_object: ' || SYS.DICTIONARY_OBJ_NAME );
DBMS_OUTPUT.PUT_LINE( 'Data: '           || SYSDATE );
END;
/

```

```

CREATE INDEX index_ang ON angajati(nume);
DROP INDEX index_ang;

```

The screenshot shows the Oracle SQL Developer interface. In the top right, a message window titled "Andreea_SGBD" displays log entries for creating and dropping an index. Below it, the "Script Output" window shows the execution of a script containing a trigger definition and the creation/dropping of an index.

```

Andreea_SGBD x
Utilizator: SYSTEM
Nume_bd: ORCL
Eveniment: CREATE
Nume_object: INDEX_ANG
Data: 04-JAN-21

Utilizator: SYSTEM
Nume_bd: ORCL
Eveniment: DROP
Nume_object: INDEX_ANG
Data: 04-JAN-21

Script Output x | Query Result x
Task completed in 0.046 seconds
Trigger TR_AUDIT_SPITAL compiled

Index INDEX_ANG created.

Index INDEX_ANG dropped.

```

```

SELECT * FROM audit_spital;

```

The screenshot shows the "Query Result" window of Oracle SQL Developer displaying the contents of the "audit_spital" table. The table has columns: UTILIZATOR, NUME_BD, EVENIMENT, NUME_OBJECT, and DATA. The data shows 12 rows of events, all occurring on 29-DEC-20.

UTILIZATOR	NUME_BD	EVENIMENT	NUME_OBJECT	DATA
1	SYSTEM	ORCL	CREATE	INDEX_ANG
2	SYSTEM	ORCL	DROP	INDEX_ANG
3	SYSTEM	ORCL	CREATE	TR_LIMIT_NR_ANG_ON_SECTII
4	SYSTEM	ORCL	CREATE	TR_NR_PAC_ZONE_SPITAL
5	SYSTEM	ORCL	CREATE	DISTRIBUTIE_PACIENTI
6	SYSTEM	ORCL	CREATE	SPITAL_PROJECT
7	SYSTEM	ORCL	CREATE	SPITAL_PROJECT
8	SYSTEM	ORCL	CREATE	SPITAL_PROJECT
9	SYSTEM	ORCL	CREATE	SPITAL_PROJECT
10	SYSTEM	ORCL	CREATE	SPITAL_PROJECT
11	SYSTEM	ORCL	CREATE	SPITAL_PROJECT
12	SYSTEM	ORCL	CREATE	SPITAL_PROJECT

Ex 13. Definiți un **pachet** care să conțină **toate obiectele definite în cadrul proiectului**.

```
--PACKAGE SPITAL_PROJECT

53 PROCEDURE detalii_pacienti (pac_nume pacienti.nume%TYPE);
54 -----
55 FUNCTION find_pac_id (pac_nume pacienti.nume%TYPE)
56     RETURN pacienti.id_pacient%TYPE;
57 -----
58 END SPITAL_PROJECT;
59 /
60 CREATE OR REPLACE PACKAGE BODY SPITAL_PROJECT
61 AS
62     --7
63     CURSOR c (sec_id sectii.id_sectie%TYPE)
64         RETURN angajati%ROWTYPE
65     IS
66         SELECT *
67         FROM angajati
68         WHERE id_sectie = sec_id;
69 -----
70     CURSOR d
71         RETURN sectii%ROWTYPE
72     TC
```

Script Output X | Task completed in 0.061 seconds

```
Package SPITAL_PROJECT compiled

Package Body SPITAL_PROJECT compiled
```

```
DROP PACKAGE SPITAL_PROJECT;
```

```
/
```

```
CREATE OR REPLACE PACKAGE SPITAL_PROJECT
```

```
AS
```

```
--7
-- o procedura care utilizeaza 2 cursoare:
-- ( -> un cursor care obtine lista angajatilor care lucreaza pe o sectie al carei id este dat ca parametru;
-- -> un cursor care obtine lista tuturor sectiilor din spital; )
-- si obtine pt fiecare sectie: codul, numele acesteia si zona din care face parte
```

```
CURSOR c (sec_id sectii.id_sectie%TYPE)
    RETURN angajati%ROWTYPE;
```

```

CURSOR d
  RETURN sectii%ROWTYPE;
-----
PROCEDURE gaseste_ang;
-----
--6
-- o procedura prin care se actualizeaza cu o valoare data ca parametru salariul unui
-- angajat (doctor sau asistent) al carui nume este dat ca parametru:
-- -> se va verifica daca angajatul este doar doctor sau asistent;
-- -> daca sunt mai multi angajati care au acelasi nume, atunci se va afisa un mesaj corespunzator si
-- de asemenea se va afisa lista acestora;
-- -> daca nu exista angajati cu numele dat, atunci se va afisa un mesaj corespunzator;

PROCEDURE marire_salariu (ang_nume angajati.nume%TYPE, nou_salariu angajati.salariu%TYPE);
-----
FUNCTION find_emp_id(ang_nume angajati.nume%TYPE)
  RETURN angajati.id_angajat%TYPE;
-----
--8
--Definiti un subprogram stocat de tip FUNCTIE care determine numarul de pacienti care
--în prezent sunt internati într-un salon dat ca parametru.
--Detaliati informatiile referitoare la locatia acestor pacienti internati:
-- id-ul si numele acestora, sectia in care se afla salonul, si zona din care face parte sectia.

FUNCTION salon_nr_pacienti
  (v_salon saloane.id_salon%TYPE)
RETURN NUMBER;
-----
--9
--Definiti un subprogram stocat de tip PROCEDURA care sa realizeze un buletin
-- cu toate informatiile pacientului.
--Se vor afisa informatii referitoare la codul doctorului ce l-a tratat, data internarii si externarii,
-- codul tratamentului administrat si codul asistentei ce l-a ingrijit, dar si date referitoare la
-- locatia pacientului internat in salon, respectiv sectia din care face parte salonul.

PROCEDURE detalii_pacienti (pac_nume pacienti.nume%TYPE);
-----
FUNCTION find_pac_id (pac_nume pacienti.nume%TYPE)
  RETURN pacienti.id_pacient%TYPE;
-----
END SPITAL_PROJECT;
/
=====
```

```
CREATE OR REPLACE PACKAGE BODY SPITAL_PROJECT
```

```
AS
```

```
--7
```

```
CURSOR c (sec_id sectii.id_sectie%TYPE)
    RETURN angajati%ROWTYPE
```

```
IS
```

```
    SELECT *
        FROM angajati
       WHERE id_sectie = sec_id;
```

```
CURSOR d
```

```
    RETURN sectii%ROWTYPE
```

```
IS
```

```
    SELECT *
        FROM sectii;
```

```
--6
```

```
PROCEDURE marire_salariu (ang_nume angajati.nume%TYPE,
                           nou_salariu angajati.salariu%TYPE)
```

```
IS
```

```
    ang_id angajati.id_angajat%TYPE := find_emp_id(ang_nume);
    doctor_id_minim doctori.id_doctor%TYPE;
    doctor_id_maxim doctori.id_doctor%TYPE;
    asistent_id_minim asistente.id_asistent%TYPE;
    asistent_id_maxim asistente.id_asistent%TYPE;
```

```
SALARY_NOT_IN_RANGE EXCEPTION;
```

```
PRAGMA exception_init(SALARY_NOT_IN_RANGE, -20002);
```

```
BEGIN
```

```
    SELECT min(id_doctor) -- 1
        INTO doctor_id_minim
        FROM doctori dr
            JOIN angajati ang ON (dr.id_doctor = ang.id_angajat);
```

```
    SELECT max(id_doctor) -- 36
```

```
        INTO doctor_id_maxim
        FROM doctori dr
            JOIN angajati ang ON (dr.id_doctor = ang.id_angajat);
```

```
    SELECT min(id_asistent) -- 37
```

```
        INTO asistent_id_minim
        FROM asistente a
            JOIN angajati ang ON (a.id_asistent = ang.id_angajat);
```

```
    SELECT max(id_asistent) -- 72
```

```
        INTO asistent_id_maxim
        FROM asistente a
            JOIN angajati ang ON (a.id_asistent = ang.id_angajat);
```

```

-- DBMS_OUTPUT.PUT_LINE( 'doctor_id_minim ' || doctor_id_minim );
-- DBMS_OUTPUT.PUT_LINE( 'doctor_id_maxim ' || doctor_id_maxim );
-- DBMS_OUTPUT.PUT_LINE( 'asistent_id_minim ' || asistent_id_minim );
-- DBMS_OUTPUT.PUT_LINE( 'asistent_id_maxim ' || asistent_id_maxim );

IF not( (ang_id >= doctor_id_minim and ang_id <= doctor_id_maxim)
        or (ang_id >= asistent_id_minim and ang_id <= asistent_id_maxim) ) THEN
    RAISE SALARY_NOT_IN_RANGE;
END IF;

UPDATE angajati
SET salariu = nou_salariu
WHERE id_angajat = ang_id;

-- DBMS_OUTPUT.PUT_LINE ('Angajatul cu id ' || ang_id || ' are salariul modificat.' );

IF (ang_id >= doctor_id_minim and ang_id <= doctor_id_maxim) THEN
    DBMS_OUTPUT.PUT_LINE('Angajatul DOCTOR cu id: ' || ang_id || ' are salariul modificat: '
                         || nou_salariu || ' ron.');
ELSIF (ang_id >= asistent_id_minim and ang_id <= asistent_id_maxim) THEN
    DBMS_OUTPUT.PUT_LINE('Angajatul ASISTENT cu id: ' || ang_id || ' are salariul modificat: '
                         || nou_salariu || ' ron.');
END IF;

EXCEPTION
    WHEN SALARY_NOT_IN_RANGE THEN
        DBMS_OUTPUT.PUT_LINE('Salariul dat nu ii este atribuit unui doctor sau asistent!');
        RAISE_APPLICATION_ERROR(-20002, 'Salariul dat nu ii este atribuit unui doctor sau asistent!');
END;
-----


FUNCTION find_emp_id(ang_nume angajati.nume%TYPE)
    RETURN angajati.id_angajat%TYPE
IS
    TYPE tabel_agă IS TABLE OF angajati.id_angajat%TYPE;
    ang tabel_agă;
    emp_id angajati.id_angajat%TYPE;

BEGIN
    SELECT id_angajat
    BULK COLLECT INTO ang
    FROM angajati
    WHERE UPPER(nume) = UPPER(ang_nume);

    IF ang.count = 0 THEN

```

```

        RAISE NO_DATA_FOUND;
ELSIF ang.count > 1 THEN
    RAISE TOO_MANY_ROWS;
END IF;

emp_id := ang(1);
RETURN emp_id;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista un angajat cu numele dat!');
        RAISE_APPLICATION_ERROR(-20000, 'Nu exista un angajat cu numele dat');

    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE('Mai multi angajati cu numele dat');

    FOR i IN ang.FIRST..ang.LAST LOOP
        DBMS_OUTPUT.PUT_LINE( i || ' Id angajat: ' || ang(i));
    END LOOP;

    RAISE_APPLICATION_ERROR(-20001, 'Mai multi angajati cu numele dat');
END;

```

```

--7
PROCEDURE gaseste_ang
IS
    nr number(4);
BEGIN
    FOR i in d LOOP
        nr := 0;
        DBMS_OUTPUT.PUT_LINE ('Sectie: ' || i.nume || ' Zona: ' || i.zona || ' Id_sectie: ' || i.id_sectie);
        DBMS_OUTPUT.PUT_LINE ('-----');
        DBMS_OUTPUT.PUT_LINE ('Angajati: ' || 'Id_angajat | Nume_angajat:');
        DBMS_OUTPUT.PUT_LINE ('-----');

        FOR j IN c (i.id_sectie) LOOP
            DBMS_OUTPUT.PUT_LINE(j.id_angajat || ' ' || j.nume);
            nr := nr + 1;
        END LOOP;

        IF nr = 0 THEN
            DBMS_OUTPUT.PUT_LINE('--> Sectia nu are momentan angajati...');

        END IF;

        DBMS_OUTPUT.PUT_LINE('');
    END LOOP;
END;

```

```

--8
FUNCTION salon_nr_pacienti
    (v_salon saloane.id_salon%TYPE)
RETURN NUMBER IS
    t saloane.id_salon%TYPE;
    nr_pac number;
BEGIN
    select id_salon into t
    from saloane
    where id_salon = v_salon;

    select COUNT(id_pacient)
    into nr_pac
    from pacienti p
        join saloane sal on p.id_salon = sal.id_salon
        join sectii sec on sal.id_sectie = sec.id_sectie
    where upper(sal.id_salon) = upper(v_salon);

    dbms_output.put_line('Nr pacienti internati in salonul respectiv: ' || nr_pac);
    dbms_output.put_line('-----');

    FOR e IN (
        select p.id_pacient, p.nume_pacient_nume, sec.id_sectie, sec.nume, sec.zona,
               sal.id_salon, sal.nr_nume
        from saloane sal
            join sectii sec on sec.id_sectie = sal.id_sectie
            join pacienti p on p.id_salon = sal.id_salon
        where upper(sal.id_salon) = upper(v_salon)
    ) LOOP
        DBMS_OUTPUT.PUT_LINE
            ( 'Id_pacient: ' || e.id_pacient || ' -> Nume_pacient: ' || e.nume_pacient_nume );
        DBMS_OUTPUT.PUT_LINE
            ( 'Id_salon: ' || e.id_salon || ' Nr_salon: ' || e.nr_nume );
        DBMS_OUTPUT.PUT_LINE
            ( 'Id_sectie: ' || e.id_sectie || ' -> Nume_sectie: ' || e.nume_sectie || ' -> Zona_sectiei: ' || e.zona );
        DBMS_OUTPUT.PUT_LINE ('');
    END LOOP;

    DBMS_OUTPUT.PUT_LINE('');

    IF nr_pac = 0 THEN
        insert into info_agă
        values (seq_info_agă.nextval, user, sysdate, 'SELECT', 0, 'Nu există PACIENTI în salonul dat ca parametru');
        dbms_output.put_line('Nu există PACIENTI în salonul dat ca parametru');
        return 0;
    END IF;

    insert into info_agă values (seq_info_agă.nextval, user, sysdate, 'SELECT', 1, null);
    return nr_pac;

```

EXCEPTION

```
WHEN NO_DATA_FOUND THEN
    insert into info_ag
    values (seq_info_ag.nextval, user, sysdate, 'SELECT', -1, 'Nu exista acest salon al sectiei cu codul dat');
    dbms_output.put_line('Nu exista acest salon al sectiei cu codul dat');
    RAISE_APPLICATION_ERROR(-20000, 'Nu exista salon cu numele dat');
    return -1;
```

WHEN TOO_MANY_ROWS THEN

```
-- nu va intra niciodata pe aceasta exceptie deoarece saloanele au id-ul unic
-- (este cheie primara) deci nu vor exista niciodata 2 saloane cu acelasi cod
    insert into info_ag
    values(seq_info_ag.nextval, user, sysdate, 'SELECT', -2, 'Exista mai multe saloane cu codul dat');
    dbms_output.put_line(SQL%ROWCOUNT || 'Exista mai multe saloane cu codul dat');
    RAISE_APPLICATION_ERROR(-20001, 'Exista mai multe saloane cu codul dat');
    return -2;
```

WHEN OTHERS THEN

```
    insert into info_ag
    values(seq_info_ag.nextval, user, sysdate, 'SELECT', -3, 'Alta eroare!');
    dbms_output.put_line('Alta eroare!');
    RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
    return -3;
```

```
END;
```

```
--9
```

```
PROCEDURE detalii_pacienti (pac_nume pacienti.nume%TYPE)
IS
    pac_id pacienti.id_pacient%TYPE := find_pac_id(pac_nume);
BEGIN
    DBMS_OUTPUT.PUT_LINE ('Pacientul ' || ' detine urmatoarele info: ');
    DBMS_OUTPUT.PUT_LINE ('');
    DBMS_OUTPUT.PUT_LINE ( 'Id_pacient: ' || pac_id || ' -> nume_pacient: ' || pac_nume );
    DBMS_OUTPUT.PUT_LINE ( '-----' );
    DBMS_OUTPUT.PUT_LINE ('');
```

```
FOR e IN (
    select p.id_pacient, p.nume pacient_nume,
           t.id_doctor, t.id_tratament, t.data_tratament, p.internare, p.externare,
           i.id_asistent, i.data_ingrijire,
           sal.id_salon, sal.nr_numar,
           sec.id_sectie, sec.nume sectie_nume, sec.zona
      from pacienti p
     join trateaza t on t.id_pacient = p.id_pacient
     join ingrijesc i on i.id_pacient = p.id_pacient
     join saloane sal on sal.id_salon = p.id_salon
     join sectii sec on sec.id_sectie = sal.id_sectie
   where p.id_pacient = pac_id and t.data_tratament = i.data_ingrijire ) LOOP
```

```

DBMS_OUTPUT.PUT_LINE ( 'DATA: ' || e.data_ingrijire );
DBMS_OUTPUT.PUT_LINE ( 'Id_doctor: ' || e.id_doctor );
DBMS_OUTPUT.PUT_LINE ( 'Id_asistent: ' || e.id_asistent);

DBMS_OUTPUT.PUT_LINE ( '-> id_tratament: ' || e.id_tratament );
DBMS_OUTPUT.PUT_LINE ( '-> internare: ' || e.internare || '-> externare: ' || e.externare );

DBMS_OUTPUT.PUT_LINE ( '> Id_salon: ' || e.id_salon || '-> nr_salon: ' || e.nr_nume );
DBMS_OUTPUT.PUT_LINE ( '> Id_sectie: ' || e.id_sectie || '-> nume_sectie: ' || e.sectie_nume
    || '-> zona_sectiei: ' || e.zona );
DBMS_OUTPUT.PUT_LINE ('-----');
DBMS_OUTPUT.PUT_LINE ('');

END LOOP;

```

EXCEPTION

WHEN OTHERS THEN

```

DBMS_OUTPUT.PUT_LINE('Alta eroare!');
RAISE_APPLICATION_ERROR(-20003,'Alta eroare!');

```

END;

FUNCTION find_pac_id (pac_nume pacienti.nume%TYPE)

 RETURN pacienti.id_pacient%TYPE

IS

 TYPE tabel_agă IS TABLE OF pacienti.id_pacient%TYPE;

 pac tabel_agă;

 pac_id pacienti.id_pacient%TYPE;

BEGIN

 SELECT id_pacient

 BULK COLLECT INTO pac

 FROM pacienti

 WHERE UPPER(nume) = UPPER(pac_nume);

 IF pac.count = 0 THEN

 RAISE NO_DATA_FOUND;

 ELSIF pac.count > 1 THEN

 RAISE TOO_MANY_ROWS;

 END IF;

 pac_id := pac(1);

 RETURN pac_id;

EXCEPTION

WHEN NO_DATA_FOUND THEN

 DBMS_OUTPUT.PUT_LINE ('Nu există un pacient cu numele dat!');

 RAISE_APPLICATION_ERROR (-20000, 'Nu există un pacient cu numele dat!');

WHEN TOO_MANY_ROWS THEN

 DBMS_OUTPUT.PUT_LINE ('Mai mulți pacienți cu numele dat');

```

FOR i IN pac.FIRST..pac.LAST LOOP
    DBMS_OUTPUT.PUT_LINE ( i || ' Id pacient: ' || pac(i));
END LOOP;

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

END;
-----

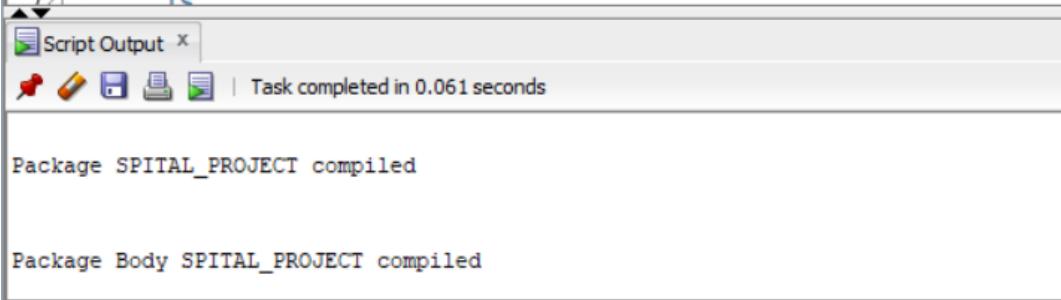
END SPITAL_PROJECT;
/

```

```

53     PROCEDURE detalii_pacienti (pac_nume pacienti.nume%TYPE);
54     -----
55     FUNCTION find_pac_id (pac_nume pacienti.nume%TYPE)
56         RETURN pacienti.id_pacient%TYPE;
57     -----
58 END SPITAL_PROJECT;
59 /
60 CREATE OR REPLACE PACKAGE BODY SPITAL_PROJECT
61 AS
62     --7
63     CURSOR c (sec_id sectii.id_sectie%TYPE)
64         RETURN angajati%ROWTYPE
65     IS
66         SELECT *
67         FROM angajati
68         WHERE id_sectie = sec_id;
69     -----
70     CURSOR d
71         RETURN sectii%ROWTYPE
72     TS

```



The screenshot shows the Oracle SQL Developer interface with the 'Script Output' tab selected. The output window displays the following messages:

- Package SPITAL_PROJECT compiled
- Package Body SPITAL_PROJECT compiled

Below the output window, there is a message indicating the task completed in 0.061 seconds.

Ex 6. Definiți un subprogram stocat care să utilizeze **un tip de COLECTIE studiat**. Apelați subprogramul.

-- o procedura prin care se actualizeaza cu o valoare data ca parametru salariul unui angajat (doctor sau asistent) al carui nume este dat ca parametru:
-- > se va verifica daca angajatul este doar doctor sau asistent;
-- > daca sunt mai multi angajati care au acelasi nume, atunci se va afisa un mesaj corespunzator si de asemenea se va afisa lista acestora;
-- > daca nu exista angajati cu numele dat, atunci se va afisa un mesaj corespunzator;

```
select * from angajati where nume = 'Ciuchi'; --doctor
rollback;
BEGIN
    SPITAL_PROJECT.marire_salariu('Ciuchi',5010); -- salariu vechi: 5002
END;
/
```

```
select * from angajati where nume = 'Ghiuta'; --asistent
rollback;
BEGIN
    SPITAL_PROJECT.marire_salariu('Ghiuta',3510); -- salariu vechi: 3503
END;
/
```

```
select * from angajati where nume = 'Chiuca'; -- nu exista niciun angajat cu acest nume
BEGIN
    SPITAL_PROJECT.marire_salariu('Chiuca',3200); -- Eroare: Nu exista un angajat cu numele dat!
END;
/
```

```
select * from angajati where nume = 'Hangantu'; -- 2 angajati cu acest nume
BEGIN
    SPITAL_PROJECT.marire_salariu('Hangantu',3700);
    -- Eroare: Salariul dat nu ii este atribuit unui doctor sau asistent!
END;
/
```

```
BEGIN
    SPITAL_PROJECT.marire_salariu('Racu',2500); -- Eroare: Salariul dat nu ii este atribuit unui doctor sau asistent!
END;
/
```

Worksheet | Query Builder

```

14 BEGIN
15   SPITAL_PROJECT.marire_salariu('Ciuchi',5010); -- salariu vechi: 5002
16 END;
17 /
18 -----
19 select * from angajati where nume = 'Chiuta'; --asistent
20 rollback;
21 BEGIN
22   SPITAL_PROJECT.marire_salariu('Ghiuta',3510); -- salariu vechi: 3503
23 END;
24 /
25 -----
26 select * from angajati where nume = 'Chiucă'; -- nu există niciun angajat cu acest nume
27 BEGIN
28   SPITAL_PROJECT.marire_salariu('Chiucă',3200); -- Eroare: Nu există un angajat cu numele dat!
29 END;
30 /

```

Script Output X | Task completed in 0.073 seconds

Error starting at line : 27 in command -

```

BEGIN
  SPITAL_PROJECT.marire_salariu('Chiucă',3200); -- Eroare: Nu există un angajat cu numele dat!
END;
Error report -
ORA-20000: Nu există un angajat cu numele dat!
ORA-06512: at "SYSTEM.SPITAL_PROJECT", line 105

```

Worksheet | Query Builder

```

23 END;
24 /
25 -----
26 select * from angajati where nume = 'Chiucă'; -- nu există niciun angajat cu acest nume
27 BEGIN
28   SPITAL_PROJECT.marire_salariu('Chiucă',3200); -- Eroare: Nu există un angajat cu numele dat!
29 END;
30 /
31 -----
32 select * from angajati where nume = 'Hanganu'; -- 2 angajati cu acest nume
33 BEGIN
34   SPITAL_PROJECT.marire_salariu('Hanganu');
35   -- Eroare: Salariul dat nu îl este atribuit unui doctor sau asistent!
36 END;
37 /

```

Script Output X | Task completed in 0.062 seconds

Error starting at line : 33 in command -

```

BEGIN
  SPITAL_PROJECT.marire_salariu('Hanganu',3700);
  -- Eroare: Salariul dat nu îl este atribuit unui doctor sau asistent!
END;
Error report -
ORA-20001: Mai mulți angajati cu numele dat
ORA-06512: at "SYSTEM.SPITAL_PROJECT", line 114
ORA-06512: at "SYSTEM.SPITAL_PROJECT", line 21
ORA-06512: at line 2

```

Worksheet | Query Builder

```

32 select * from angajati where nume = 'Hanganu'; -- 2 angajati cu acest nume
33 BEGIN
34   SPITAL_PROJECT.marire_salariu('Hanganu',3700);
35   -- Eroare: Salariul dat nu îl este atribuit unui doctor sau asistent!
36 END;
37 /
38 -----
39 select * from angajati where nume = 'Racu'; -- infirmiera
40 BEGIN
41   SPITAL_PROJECT.marire_salariu('Racu',2500); -- Eroare: Salariul dat nu îl este atribuit unui
42 END;
43 /
44 -----
45

```

Script Output X | Task completed in 0.046 seconds

Error starting at line : 40 in command -

```

BEGIN
  SPITAL_PROJECT.marire_salariu('Racu',2500); -- Eroare: Salariul dat nu îl este atribuit unui doctor sau asistent!
END;
Error report -
ORA-20002: Salariul dat nu îl este atribuit unui doctor sau asistent!
ORA-06512: at "SYSTEM.SPITAL_PROJECT", line 77
ORA-06512: at line 2

```

Angajatul DOCTOR cu id: 10 are salariul modificat: 5010 ron.
 Angajatul ASISTENT cu id: 60 are salariul modificat: 3510 ron.
 Nu există un angajat cu numele dat!

Angajatul DOCTOR cu id: 10 are salariul modificat: 5010 ron.
 Angajatul ASISTENT cu id: 60 are salariul modificat: 3510 ron.
 Nu există un angajat cu numele dat!
 Mai mulți angajati cu numele dat
 1 Id angajat: 42
 2 Id angajat: 106

Angajatul DOCTOR cu id: 10 are salariul modificat: 5010 ron.
 Angajatul ASISTENT cu id: 60 are salariul modificat: 3510 ron.
 Nu există un angajat cu numele dat!
 Mai mulți angajati cu numele dat
 1 Id angajat: 42
 2 Id angajat: 106
 Salariul dat nu îl este atribuit unui doctor sau asistent!

Ex 7. Definiți un subprogram stocat care să utilizeze **un tip de CURSOR studiat**. Apelați subprogramul.

-- o procedura care utilizeaza 2 cursoare:

-- (-> un cursor care obtine lista angajatilor care lucreaza pe o sectie al carei id este dat ca parametru;

-- -> un cursor care obtine lista tuturor sectiilor din spital;)

-- si obtine pt fiecare sectie: codul, numele acesteia si zona din care face parte

BEGIN

SPITAL_PROJECT.gaseste_ang;

END;

/

PL/SQL procedure successfully completed.

Sectie	Zona	Id_sectie
UPU	ROSIE	1
AII	ROSIE	2
LABORATOR	VERDE	3

Angajati	Id_angajat Nume_angajat
1	Seceleanu
2	Geru
3	Tarcuta
4	Marin
37	Iacob
38	Mihai
39	Jitea
40	Copacei
73	Adavanei
74	Duta
75	Alupei
76	Rogoza
5	Hirjanu
6	Zarzu
7	Dogaru
41	Ratusanu
42	Hanganu
43	Manas
77	Maran
78	Palaghita
79	Girileanu
8	Andronic
44	Nechita

-----CURSOR C: -----

DECLARE

val varchar2(10);

lista_angajati%ROWTYPE;

BEGIN

val := '1';

DBMS_OUTPUT.PUT_LINE('Lista ang care lucreaza pe sectia: ' || val);

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

FOR v_cursor IN spital_project.c (val) LOOP

DBMS_OUTPUT.PUT_LINE('Id_angajat: ' || v_cursor.id_angajat || ' -> Nume_angajat: ' || v_cursor.nume);

END LOOP;

END;

/

Worksheet Query Builder

```

56 -- cursor c: -----
57 DECLARE
58   val varchar2(10);
59   lista_angajati%ROWTYPE;
60 BEGIN
61   val := '1';
62   DBMS_OUTPUT.PUT_LINE('Lista ang care lucreaza pe sectia: ' || val);
63   DBMS_OUTPUT.PUT_LINE('-----');
64   FOR v_cursor IN spital_project.c (val) LOOP
65     DBMS_OUTPUT.PUT_LINE('Id_angajat: ' || v_cursor.id_angajat || ' -> Nume_angajat: ' || v_cursor.nume);
66   END LOOP;
67 END;
68 /

```

Script Output Query Result

Task completed in 0.041 seconds

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

Andreea_SGBD x

Lista ang care lucreaza pe sectia: 1

Id_angajat: 1 -> Nume_angajat: Seceleanu
 Id_angajat: 2 -> Nume_angajat: Geru
 Id_angajat: 3 -> Nume_angajat: Tarcuta
 Id_angajat: 4 -> Nume_angajat: Marin
 Id_angajat: 37 -> Nume_angajat: Iacob
 Id_angajat: 38 -> Nume_angajat: Mihai
 Id_angajat: 39 -> Nume_angajat: Jitea
 Id_angajat: 40 -> Nume_angajat: Copacel
 Id_angajat: 73 -> Nume_angajat: Adavanei
 Id_angajat: 74 -> Nume_angajat: Duta
 Id_angajat: 75 -> Nume_angajat: Alupei
 Id_angajat: 76 -> Nume_angajat: Rogoza

-----cursor d:-----

DECLARE

 lista_sectii%ROWTYPE;

BEGIN

```

    FOR v_cursor IN spital_project.d () LOOP
      DBMS_OUTPUT.PUT_LINE( 'Id_sectie: ' || v_cursor.id_sectie || ' -> Nume_sectie: ' || v_cursor.nume);
      DBMS_OUTPUT.PUT_LINE( ' -> Categorie: ' || v_cursor.categorie || ' -> Zona: ' || v_cursor.zona );
      DBMS_OUTPUT.PUT_LINE('-----');
    END LOOP;
  
```

END;

/

Worksheet Query Builder

```

67 /-----/
68 -- cursor d: -----
69
70 DECLARE
71   lista_sectii%ROWTYPE;
72 BEGIN
73   FOR v_cursor IN spital_project.d () LOOP
74     DBMS_OUTPUT.PUT_LINE( 'Id_sectie: ' || v_cursor.id_sectie || ' -> Nume_sectie: ' || v_cursor.nume);
75     DBMS_OUTPUT.PUT_LINE( ' -> Categorie: ' || v_cursor.categorie || ' -> Zona: ' || v_cursor.zona );
76     DBMS_OUTPUT.PUT_LINE('-----');
77   END LOOP;
78 END;
79 /
80

```

Script Output Query Result

Task completed in 0.041 seconds

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

Andreea_SGBD x

Id_sectie: 1 -> Nume_sectie: UPU
 -> Categorie: -> Zona: ROSIE

Id_sectie: 2 -> Nume_sectie: ATI
 -> Categorie: -> Zona: ROSIE

Id_sectie: 3 -> Nume_sectie: LABORATOR
 -> Categorie: -> Zona: VERDE

Id_sectie: 41 -> Nume_sectie: RADIOLOGIE
 -> Categorie: -> Zona: ROSIE

Id_sectie: 42 -> Nume_sectie: RADIOLOGIE
 -> Categorie: -> Zona: VERDE

Id_sectie: 51 -> Nume_sectie: CHIRURGIE
 -> Categorie: adulti -> Zona: ROSIE

Id_sectie: 52 -> Nume_sectie: CHIRURGIE
 -> Categorie: adulti -> Zona: VERDE

Id_sectie: 53 -> Nume_sectie: CHIRURGIE
 -> Categorie: adulti -> Zona: TAMPON

Id_sectie: 61 -> Nume_sectie: ORTOPEDIE
 -> Categorie: adulti -> Zona: ROSIE

Id_sectie: 62 -> Nume_sectie: ORTOPEDIE
 -> Categorie: adulti -> Zona: VERDE

Id_sectie: 63 -> Nume_sectie: ORTOPEDIE
 -> Categorie: adulti -> Zona: TAMPON

Id_sectie: 71 -> Nume_sectie: NEUROLOGIE
 -> Categorie: adulti -> Zona: ROSIE

Id_sectie: 72 -> Nume_sectie: NEUROLOGIE

Ex 8. Definiți un subprogram stocat de tip FUNCTIE care să utilizeze 3 dintre tabelele definite.

Tratați toate exceptiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

--Definii un subprogram stocat de tip FUNCTIE care determine numarul de pacienti
-- care în prezent sunt internati într-un salon dat ca parametru.
--Detaliati informatiile referitoare la locatia acestor pacienti internati:
-- id-ul si numele acestora, sectia in care se afla salonul, si zona din care face parte sectia

```
DROP TABLE info_agă;
CREATE TABLE info_agă (
    id number(3)      primary key,
    utilizator        VARCHAR2(30),
    data              DATE,
    comanda           VARCHAR2(20),
    nr_linii          NUMBER(3),
    eroare            VARCHAR2(255)
);
COMMIT;
```

```
drop sequence seq_info_agă;
create sequence seq_info_agă
start with 1
increment by 1;
```

```
select seq_info_agă.nextval from dual;
select seq_info_agă.curval from dual;
select * from info_agă;
```

```
DECLARE
    val number;
BEGIN
    val := spital_project.salon_nr_pacienti('1_A'); --Functioneaza
    IF val > 0 THEN
        dbms_output.put_line ('Rezultat: ' || val || ' pacienti internati in acest salon. ');
    ELSE
        dbms_output.put_line ('Rezultat: ' || val );
    END IF;
END;
/
```

```

104
105 select seq_info_ag.a.nextval from dual;
106 select seq_info_ag.a.curval from dual;
107 select * from info_ag;
108
109 DECLARE
110   val number;
111 BEGIN
112   val := spital_project.salon_nr_pacienti('1_A'); --Functioneaza
113   IF val > 0 THEN
114     dbms_output.put_line ('Rezultat: ' || val || ' pacienti internati in acest salon.');
115   ELSE
116     dbms_output.put_line ('Rezultat: ' || val );
117   END IF;
118 END;
119 /
120
121 DECLARE

```

Script Output | Query Result | Task completed in 0.047 seconds

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

Andreea_SGBD x
Nr pacienti internati in salonul respectiv: 2

Id_pacient: 1 -> Nume_pacient: Grecu
Id_salon: 1_A Nr_salon: A
Id_sectie: 1 -> Nume sectie: UPU -> Zona_sectiei: ROSIE

Id_pacient: 2 -> Nume_pacient: Stroe
Id_salon: 1_A Nr_salon: A
Id_sectie: 1 -> Nume sectie: UPU -> Zona_sectiei: ROSIE

Rezultat: 2 pacienti internati in acest salon.

```

DECLARE
  val number;
BEGIN
  val := spital_project.salon_nr_pacienti('1_AAA'); --Nu exista acest salon...
  IF val > 0 THEN
    dbms_output.put_line ('Rezultat: ' || val || ' pacienti internati in acest salon.');
  ELSE
    dbms_output.put_line ('Rezultat: ' || val );
  END IF;
END;
/

```

```

119 /
120 -----
121 DECLARE
122   val number;
123 BEGIN
124   val := spital_project.salon_nr_pacienti('1_AAA'); --Nu exista acest salon...
125   IF val > 0 THEN
126     dbms_output.put_line ('Rezultat: ' || val || ' pacienti internati in acest salon.');
127   ELSE
128     dbms_output.put_line ('Rezultat: ' || val );
129   END IF;
130 END;
131 /
132 select * from info_ag;

```

Script Output | Query Result | Task completed in 0.076 seconds

END;
Error report -
ORA-20000: Nu exista salon cu numele dat
ORA-06512: at "SYSTEM.SPITAL_PROJECT", line 195
ORA-06512: at line 4
20000. 00000 - "%s"
*Cause: The stored procedure 'raise_application_error'
was called which causes this error to be generated.
*Action: Correct the problem as described in the error message or contact
the application administrator or DBA for more information.

Andreea_SGBD x
Nu exista acest salon al sectiei cu codul dat

```

select * from info_ag;
select * from saloane;
INSERT INTO saloane (id_salon, nr_nume, id_sectie) VALUES ('174_A', 'A', 173);

DECLARE
val number;
BEGIN
val := spital_project.salon_nr_pacienti('174_A'); --Salonul nou adaugat nu are pacienti
IF val > 0 THEN
dbms_output.put_line ('Rezultat: ' || val || ' pacienti internati in acest salon.');
ELSE
dbms_output.put_line ('Rezultat: ' || val );
END IF;
END;
/
select * from info_ag;
DELETE FROM saloane WHERE id_salon = '174_A';

```

Andreea_SGBD

```

132 select * from info_ag;
133 select * from saloane;
134
135 INSERT INTO saloane (id_salon, nr_nume, id_sectie)
VALUES ('174_A', 'A', 173 );
136
137 DECLARE
138     val number;
139 BEGIN
140     val := spital_project.salon_nr_pacienti('174_A'); --Salonul nou adaugat nu are pacienti
141     IF val > 0 THEN
142         dbms_output.put_line ('Rezultat: ' || val || ' pacienti internati in acest salon.');
143     ELSE
144         dbms_output.put_line ('Rezultat: ' || val );
145     END IF;
146 END;
147 /
148
149

```

Script Output | Query Result

Nr pacienti internati in salonul respectiv: 0

Nu exista PACIENTI in salonul dat ca parametru

Rezultat: 0

Task completed in 0.032 seconds

1 row inserted.

PL/SQL procedure successfully completed.

Andreea_SGBD

```

149 select * from info_ag;
150
151
152 DELETE FROM saloane WHERE id_salon = '174_A';

```

Script Output | Query Result

All Rows Fetched: 9 in 0.002 seconds

ID	UTILIZATOR	DATA	COMANDA	NR_LINII	EROARE
1	1 SYSTEM	24-DEC-20	SELECT	1	(null)
2	4 SYSTEM	24-DEC-20	SELECT	-1	Nu exista acest salon al sectiei cu codul dat
3	5 SYSTEM	24-DEC-20	SELECT	-1	Nu exista acest salon al sectiei cu codul dat
4	6 SYSTEM	27-DEC-20	SELECT	1	(null)
5	9 SYSTEM	30-DEC-20	SELECT	1	(null)
6	10 SYSTEM	30-DEC-20	SELECT	1	(null)
7	12 SYSTEM	30-DEC-20	SELECT	0	Nu exista PACIENTI in salonul dat ca parametru
8	13 SYSTEM	30-DEC-20	SELECT	0	Nu exista PACIENTI in salonul dat ca parametru
9	14 SYSTEM	31-DEC-20	SELECT	1	(null)

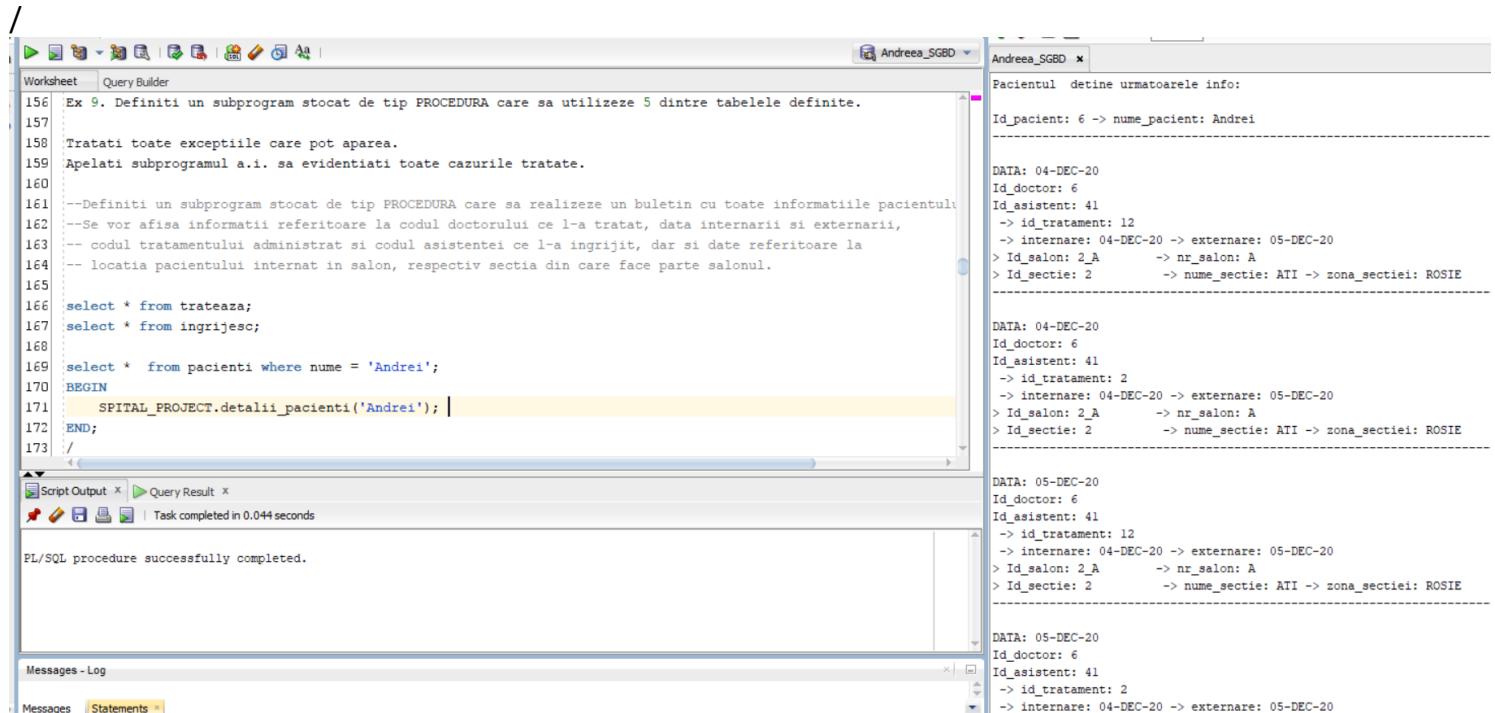
Ex 9. Definiți un subprogram stocat de tip PROCEDURA care să utilizeze 5 dintre tabelele definite.

Tratați toate exceptiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

--Definiți un subprogram stocat de tip PROCEDURA care să realizeze un buletin cu toate informațiile pacientului.
--Se vor afisa informații referitoare la codul doctorului ce l-a tratat, data internarii și externării,
-- codul tratamentului administrat și codul asistentei ce l-a ingrijit, dar și date referitoare la
-- locația pacientului internat în salon, respectiv secția din care face parte salonul.

```
select * from trateaza;  
select * from ingrijesc;
```

```
select * from pacienti where nume = 'Andrei';  
BEGIN  
    SPITAL_PROJECT.detalii_pacienti('Andrei');  
END;
```



```
156 Ex 9. Definiți un subprogram stocat de tip PROCEDURA care să utilizeze 5 dintre tabelele definite.  
157  
158 Tratați toate exceptiile care pot apărea.  
159 Apelați subprogramul a.i. să evidențiați toate cazurile tratate.  
160  
161 --Definiți un subprogram stocat de tip PROCEDURA care să realizeze un buletin cu toate informațiile pacientului  
162 --Se vor afisa informații referitoare la codul doctorului ce l-a tratat, data internarii și externării,  
163 -- codul tratamentului administrat și codul asistentei ce l-a ingrijit, dar și date referitoare la  
164 -- locația pacientului internat în salon, respectiv secția din care face parte salonul.  
165  
166 select * from trateaza;  
167 select * from ingrijesc;  
168  
169 select * from pacienti where nume = 'Andrei';  
170 BEGIN  
171     SPITAL_PROJECT.detalii_pacienti('Andrei'); |  
172 END;  
173 /
```

Pacientul detine următoarele info:
Id_pacient: 6 -> nume_pacient: Andrei

DATA: 04-DEC-20
Id_doctor: 6
Id_asistent: 41
-> id_tratament: 12
-> internare: 04-DEC-20 -> externare: 05-DEC-20
> Id_salon: 2_A -> nr_salon: A
> Id_sectie: 2 -> nume_sectie: ATI -> zona_sectiei: ROSIE

DATA: 04-DEC-20
Id_doctor: 6
Id_asistent: 41
-> id_tratament: 2
-> internare: 04-DEC-20 -> externare: 05-DEC-20
> Id_salon: 2_A -> nr_salon: A
> Id_sectie: 2 -> nume_sectie: ATI -> zona_sectiei: ROSIE

DATA: 05-DEC-20
Id_doctor: 6
Id_asistent: 41
-> id_tratament: 12
-> internare: 04-DEC-20 -> externare: 05-DEC-20
> Id_salon: 2_A -> nr_salon: A
> Id_sectie: 2 -> nume_sectie: ATI -> zona_sectiei: ROSIE

DATA: 05-DEC-20
Id_doctor: 6
Id_asistent: 41
-> id_tratament: 2
-> internare: 04-DEC-20 -> externare: 05-DEC-20

PL/SQL procedure successfully completed.

Messages - Log

Messages Statements

```
select * from pacienti where nume = 'Breaca'; -- nu există niciun angajat cu acest nume
```

```
BEGIN  
    SPITAL_PROJECT.detalii_pacienti('Breaca'); -- Eroare: Nu există un angajat cu numele dat!  
END;
```

```

170 BEGIN
171   SPITAL_PROJECT.detalii_pacienti('Andrei');
172 END;
173 /
174 select * from pacienti where nume = 'Breaca'; -- nu există niciun angajat cu acest nume
175 BEGIN
176   SPITAL_PROJECT.detalii_pacienti('Breaca'); -- Eroare: Nu există un angajat cu numele dat!
177 END;
178 /

```

Script Output | Query Result | Task completed in 0.041 seconds
PL/SQL procedure successfully completed.

Error starting at line : 175 in command -
BEGIN
SPITAL_PROJECT.detalii_pacienti('Breaca'); -- Eroare: Nu există un angajat cu numele dat!
END;
Error report -
ORA-20000: Nu există un pacient cu numele dat!
ORA-06512: at "SYSTEM.SPITAL_PROJECT", line 289
ORA-06512: at "SYSTEM.SPITAL_PROJECT", line 216
ORA-06512: at line 2
20000. 00000 - "%s"
*Cause: The stored procedure 'raise_application_error'
was called which causes this error to be generated.
*Action: Correct the problem as described in the error message or contact
the application administrator or DBA for more information.

```

INSERT INTO pacienti (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
VALUES (41, 'Stroe', 0717994689, 'ADULT', 62, null, '1_A'
       ,TO_DATE('21/12/2020', 'DD/MM/YYYY'),TO_DATE('22/12/2020', 'DD/MM/YYYY') );

```

```

select * from pacienti where nume = 'Stroe'; -- Mai multi pacienti cu numele dat
BEGIN
  SPITAL_PROJECT.detalii_pacienti('Stroe'); -- Eroare: Mai multi pacienti cu numele dat
END;
/

```

```

3188
3189 INSERT INTO pacienti (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
3190 VALUES (41, 'Stroe', 0717994689, 'ADULT', 62, null, '1_A', TO_DATE('21/12/2020', 'DD/MM/YYYY'), TO_DATE('22/
3191
3192
3193 select * from pacienti where nume = 'Stroe'; -- Mai multi pacienti cu numele dat
3194 BEGIN
3195   SPITAL_PROJECT.detalii_pacienti('Stroe'); -- Eroare: Mai multi pacienti cu numele dat
3196 END;
3197 /
3198 DELETE FROM pacienti WHERE id_pacient = 41;
3199

```

Script Output | Query Result | Task completed in 0.04 seconds
1 row inserted.

Error starting at line : 3,194 in command -
BEGIN
SPITAL_PROJECT.detalii_pacienti('Stroe'); -- Eroare: Mai multi pacienti cu numele dat
END;
Error report -
ORA-20001: Mai multi pacienti cu numele dat
ORA-06512: at "SYSTEM.SPITAL_PROJECT", line 298
ORA-06512: at "SYSTEM.SPITAL_PROJECT", line 216
ORA-06512: at line 2

```
DELETE FROM pacienti WHERE id_pacient = 41;
```

Worksheet Query Builder

```
3185  SPITAL_PROJECT.detalii_pacienti('Breaca'); -- Eroare: Nu exista un angajat cu numele dat!
3186 END;
3187 /
3188
3189 INSERT INTO pacienti (id_pacient, nume, telefon, categorie, varsta, regim, id_salon, internare, externare)
3190 VALUES (41, 'Stroe', 0717994689, 'ADULT', 62, null, '1_A', TO_DATE('21/12/2020', 'DD/MM/YYYY'), TO_DATE('22/
3191
3192
3193 select * from pacienti where nume = 'Stroe'; -- Mai multi pacienti cu numele dat
3194 BEGIN
3195  SPITAL_PROJECT.detalii_pacienti('Stroe'); -- Eroare: Mai multi pacienti cu numele dat
3196 END;
3197 /
3198 DELETE FROM pacienti WHERE id_pacient = 41;
3199
```

Script Output | Task completed in 0.027 seconds

```
ORA-06512: at "SYSTEM.SPITAL_PROJECT", line 216
ORA-06512: at line 2
```

1 row deleted.

PL/SQL procedure successfully completed.

Mai multi pacienti cu numele dat
1 Id pacient: 2
2 Id pacient: 41

Pacientul detine urmatoarele info:

Id_pacient: 2 -> nume_pacient: Stroe

DATA: 02-DEC-20
Id_doctor: 2
Id_asistent: 38
-> id_tratament: 22
-> internare: 02-DEC-20 -> externare: 02-DEC-20
> Id_salon: 1_A -> nr_salon: A
> Id_sectie: 1 -> nume_sectie: UPU -> zona_sectiei: ROSIE

Ex 14. Definiți un pachet care să includă tipuri de date complexe și obiecte necesare pentru acțiuni integrate.

- Creează un pachet în care vor fi incluse funcții, cu ajutorul cărora să aflați numărul pacienților tratați de un anumit doctor, respectiv numărul pacienților îngrijiti de o anumita asistență.
- Tratati toate exceptiile care pot apărea și apelați subprogramele astfel încât să evidențiați cazurile tratate.

--PACKAGE distributie_pacienti

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there are tabs for 'Worksheet' and 'Query Builder'. The main area displays the PL/SQL code for the package. The code includes two functions: 'pac_asistente' and 'find_id_asis', and a body for 'pac_doctori'. The package is named 'distributie_pacienti'. The code is as follows:

```
540  FUNCTION pac_asistente(ang_nume angajati.nume%TYPE)
541      RETURN NUMBER;
542
543  FUNCTION find_id_asis(ang_nume angajati.nume%TYPE)
544      RETURN angajati.id_angajat%TYPE;
545  END;
546 /
547 CREATE OR REPLACE PACKAGE BODY distributie_pacienti|
548 AS
549     FUNCTION pac_doctori(ang_nume angajati.nume%TYPE)
550         RETURN NUMBER
551     IS
552         ang_id angajati.id_angajat%TYPE := find_id_dr(ang_nume);
553         nr NUMBER;
554     BEGIN
555         SELECT COUNT(*)
556         INTO nr
557         FROM ( SELECT distinct id_pacient
```

Below the code editor, there are tabs for 'Script Output' and 'Query Result'. The 'Script Output' tab shows the message: 'Task completed in 0.063 seconds'. The 'Query Result' tab shows the output: 'Package DISTRIBUTIE_PACIENTI compiled' and 'Package Body DISTRIBUTIE_PACIENTI compiled'.

CREATE OR REPLACE PACKAGE distributie_pacienti

AS

```
FUNCTION pac_doctori(ang_nume angajati.nume%TYPE)
    RETURN NUMBER;
```

```
FUNCTION find_id_dr(ang_nume angajati.nume%TYPE)
    RETURN angajati.id_angajat%TYPE;
```

```
FUNCTION pac_asistente(ang_nume angajati.nume%TYPE)
    RETURN NUMBER;
```

```
FUNCTION find_id_asis(ang_nume angajati.nume%TYPE)
    RETURN angajati.id_angajat%TYPE;
```

END;

/

```
CREATE OR REPLACE PACKAGE BODY distributie_pacienti
AS
    FUNCTION pac_doctori (ang_nume angajati.nume%TYPE)
        RETURN NUMBER
    IS
        ang_id angajati.id_angajat%TYPE := find_id_dr(ang_nume);
        nr NUMBER;
    BEGIN
        SELECT COUNT(*)
        INTO nr
        FROM (  SELECT distinct id_pacient
                  FROM trateaza
                 WHERE id_doctor = ang_id );
        RETURN nr;
    END;
```

```
FUNCTION find_id_dr(ang_nume angajati.nume%TYPE)
    RETURN angajati.id_angajat%TYPE
IS
    TYPE tabel_agă IS TABLE OF angajati.id_angajat%TYPE;
    ang tabel_agă;
    emp_id angajati.id_angajat%TYPE;

    NOT_DOCTOR EXCEPTION;
    PRAGMA exception_init(NOT_DOCTOR, -20002);
BEGIN
    SELECT id_angajat
    BULK COLLECT INTO ang
    FROM angajati
    WHERE UPPER(nume) = UPPER(ang_nume);

    FOR i IN ang.FIRST..ang.LAST LOOP
        IF ang(i) > 36 THEN
            RAISE NOT_DOCTOR;
        END IF;
    END LOOP;

    IF ang.count = 0 THEN
        RAISE NO_DATA_FOUND;
    ELSIF ang.count > 1 THEN
        RAISE TOO_MANY_ROWS;
    END IF;

    emp_id := ang(1);
    RETURN emp_id;
```

EXCEPTION

```
WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Nu exista un doctor cu numele dat!');
    RAISE_APPLICATION_ERROR(-20000, 'Nu exista doctor cu numele dat');
```

```
WHEN VALUE_ERROR THEN
    DBMS_OUTPUT.PUT_LINE('Nu exista un doctor cu numele dat!');
    RAISE_APPLICATION_ERROR(-20000, 'Nu exista doctor cu numele dat');
```

```
WHEN TOO_MANY_ROWS THEN
    DBMS_OUTPUT.PUT_LINE('Mai multi doctori cu numele dat');
```

```
FOR i IN ang.FIRST..ang.LAST LOOP
    DBMS_OUTPUT.PUT_LINE( i || ' Id angajat: ' || ang(i) );
END LOOP;
```

```
RAISE_APPLICATION_ERROR(-20001, 'Mai multi doctori cu numele dat');
```

```
WHEN NOT_DOCTOR THEN
    DBMS_OUTPUT.PUT_LINE('Angajatul NU este doctor!');
    RAISE_APPLICATION_ERROR(-20002, 'Angajatul NU este doctor!');
```

```
END;
```

```
FUNCTION pac_asistente(ang_nume angajati.nume%TYPE)
```

```
    RETURN NUMBER
```

```
IS
```

```
    ang_id angajati.id_angajat%TYPE := find_id_asis(ang_nume);
    nr NUMBER;
```

```
BEGIN
```

```
    SELECT COUNT(*)
    INTO nr
    FROM ( SELECT distinct id_pacient
            FROM ingrijesc
            WHERE id_asistent = ang_id );
```

```
    RETURN nr;
```

```
END;
```

```
FUNCTION find_id_asis(ang_nume angajati.nume%TYPE)
```

```
    RETURN angajati.id_angajat%TYPE
```

```
IS
```

```
    TYPE tabel_agă IS TABLE OF angajati.id_angajat%TYPE;
    ang tabel_agă;
    emp_id angajati.id_angajat%TYPE;
```

```
NOT_ASSISTENT EXCEPTION;
```

```
PRAGMA exception_init(NOT_ASSISTENT, -20002);
```

```

BEGIN
  SELECT id_angajat
  BULK COLLECT INTO ang
  FROM angajati
  WHERE UPPER(nume) = UPPER(ang_nume);

  FOR i IN ang.FIRST..ang.LAST LOOP
    IF NOT (ang(i) >= 37 AND ang(i) <= 72) THEN
      RAISE NOT_ASSISTENT;
    END IF;
  END LOOP;

  IF ang.count = 0 THEN
    RAISE NO_DATA_FOUND;
  ELSIF ang.count > 1 THEN
    RAISE TOO_MANY_ROWS;
  END IF;

  emp_id := ang(1);
  RETURN emp_id;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Nu exista un asistent cu numele dat!');
    RAISE_APPLICATION_ERROR(-20000, 'Nu exista asistent cu numele dat');

  WHEN VALUE_ERROR THEN
    DBMS_OUTPUT.PUT_LINE('Nu exista un asistent cu numele dat!');
    RAISE_APPLICATION_ERROR(-20000, 'Nu exista asistent cu numele dat');

  WHEN TOO_MANY_ROWS THEN
    DBMS_OUTPUT.PUT_LINE('Mai multi asistenti cu numele dat');
    RAISE_APPLICATION_ERROR(-20001, 'Mai multi asistenti cu numele dat');

  WHEN NOT_ASSISTENT THEN
    DBMS_OUTPUT.PUT_LINE('Angajatul NU este asistent!');
    RAISE_APPLICATION_ERROR(-20002, 'Angajatul NU este asistent!');
END;

```

```

END distributie_pacienti;
/

```

Worksheet Query Builder

```

540     FUNCTION pac_asistente(ang_nume angajati.nume%TYPE)
541         RETURN NUMBER;
542     -----
543     FUNCTION find_id_asis(ang_nume angajati.nume%TYPE)
544         RETURN angajati.id_angajat%TYPE;
545     END;
546 /
547 CREATE OR REPLACE PACKAGE BODY distributie_pacienti|
548 AS
549     FUNCTION pac_doctori(ang_nume angajati.nume%TYPE)
550         RETURN NUMBER
551     IS
552         ang_id angajati.id_angajat%TYPE := find_id_dr(ang_nume);
553         nr NUMBER;
554     BEGIN
555         SELECT COUNT(*)
556         INTO nr
557         FROM ( SELECT distinct id_pacient

```

Script Output Query Result

Task completed in 0.063 seconds

Package DISTRIBUTIE_PACIENTI compiled

Package Body DISTRIBUTIE_PACIENTI compiled

```
select * from angajati;
rollback;
```

```
select distinct id_doctor, id_pacient from trateaza;
```

```
SELECT distinct id_pacient FROM trateaza WHERE id_doctor = 17;
```

```
declare
    val number;
begin
    val := distributie_pacienti.pac_doctori ('Androne'); -- id 17 Dr Androne - are 3 pacienti
    dbms_output.put_line ('Rezultat: ' || val || ' pacienti');
end;
/
```

Worksheet Query Builder

```

279
280 select * from angajati;
281 rollback;
282
283 select distinct id_doctor, id_pacient from trateaza;
284
285 SELECT distinct id_pacient FROM trateaza WHERE id_doctor = 17;
286
287 declare
288   val number;
289 begin
290   val := distributie_pacienti.pac_doctori ('Androne'); -- id 17 Dr Androne - are 3 pacienti
291   dbms_output.put_line ('Rezultat: ' || val || ' pacienti');
292 end;
293 /
294
295 declare

```

Script Output x | Query Result x

Task completed in 0.033 seconds

PL/SQL procedure successfully completed.

Andreea_SGBD x | Resultat: 3 pacienti

```

declare
  val number;
begin
  val := distributie_pacienti.pac_doctori ('Iacob'); --Eroare: ang cu id 37 nu este DOCTOR
  dbms_output.put_line ('Rezultat: ' || val || ' pacienti');
end;
/

```

Worksheet Query Builder

```

294 /
295
296 declare
297   val number;
298 begin
299   val := distributie_pacienti.pac_doctori ('Iacob'); --Eroare: ang cu id 37 nu este DOCTOR
300   dbms_output.put_line ('Rezultat: ' || val || ' pacienti');
301 end;
302 /
303
304 declare

```

Script Output x | Query Result x

Task completed in 0.05 seconds

Error starting at line : 295 in command -

```

declare
  val number;
begin
  val := distributie_pacienti.pac_doctori ('Iacob'); --Eroare: ang cu id 37 nu este DOCTOR
  dbms_output.put_line ('Rezultat: ' || val || ' pacienti');
end;
Error report -
ORA-20002: Angajatul NU este doctor!
ORA-06512: at "SYSTEM.DISTRIBUTIE_PACIENTI", line 68
ORA-06512: at "SYSTEM.DISTRIBUTIE_PACIENTI", line 6
ORA-06512: at line 4

```

Andreea_SGBD x | Angajatul NU este doctor!

```

declare
val number;
begin
  val := distributie_pacienti.pac_doctori ('Ancab'); --Eroare: Nu exista doctor cu numele dat
  dbms_output.put_line ('Rezultat: ' || val || ' pacienti');
end;
/

```

The screenshot shows the Oracle SQL Developer interface. In the top right, a message box displays the error: "Nu există un doctor cu numele dat!". The main window shows the PL/SQL code and its execution results. The code attempts to call a procedure that does not exist, resulting in an error.

```

303 declare
304   val number;
305 begin
306   val := distributie_pacienti.pac_doctori ('Acab'); --Eroare: Nu există doctor cu numele dat
307   dbms_output.put_line ('Rezultat: ' || val || ' pacienti');
308 end;
309 /
310 -----
311 select * from angajati;
312 rollback;
4

```

Script Output

```

val number;
begin
  val := distributie_pacienti.pac_doctori ('Acab'); --Eroare: Nu există doctor cu numele dat
  dbms_output.put_line ('Rezultat: ' || val || ' pacienti');
end;
Error report -
ORA-20000: Nu există doctor cu numele dat
ORA-06512: at "SYSTEM.DISTRIBUTIE_PACIENTI", line 55
ORA-06512: at "SYSTEM.DISTRIBUTIE_PACIENTI", line 6
ORA-06512: at line 4
20000. 00000 - "%s"
*Cause:  The stored procedure 'raise_application_error'
         was called which causes this error to be generated.
*Action: Correct the problem as described in the error message or contact
         the application administrator or DBA for more information.

```

```

INSERT INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (0, 'Seceleanu', 0745697802, 7551, TO_DATE('07/03/1994', 'DD/MM/YYYY'),null, 2 );

```

```

select * from angajati where nume = 'Seceleanu'; -- 2 doctori cu acest nume

```

```

DELETE from angajati where id_angajat = 0;

```

```

declare
  val number;
begin
  val := distributie_pacienti.pac_doctori ('Seceleanu'); -- Eroare: 2 doctori cu acest nume
  dbms_output.put_line ('Rezultat: ' || val || ' pacienti');
end;
/

```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, titled 'Worksheet', there is a 'Query Builder' tab. The code entered is:

```

INSERT INTO angajati (id_angajat, nume, telefon, salariu, data_angajarii, id_sef, id_sectie)
VALUES (0, 'Seceleanu', 0745697802, 7551, TO_DATE('07/03/1994', 'DD/MM/YYYY'),null, 2 );

select * from angajati where nume = 'Seceleanu'; -- 2 doctori cu acest nume

DELETE from angajati where id_angajat = 0;

declare
    val number;
begin
    val := distributie_pacienti.pac_doctori ('Seceleanu'); --Eroare: 2 doctori cu acest nume
    dbms_output.put_line ('Rezultat: ' || val || ' pacient');
end;
/

```

In the top-right pane, titled 'Andreea_SGBD', the output is:

```

Mai multi doctori cu numele dat
1 Id angajat: 1
2 Id angajat: 0

```

Below the Worksheet, the 'Script Output' pane shows the executed code and its results:

```

Task completed in 0.077 seconds
var := distributie_pacienti.pac_doctori ('Seceleanu'); --Eroare: 2 doctori cu acest nume
dbms_output.put_line ('Rezultat: ' || val || ' pacient');
end;
Error report -
ORA-20001: Mai multi doctori cu numele dat
ORA-06512: at "SYSTEM.DISTRIBUTIE_PACIENTI", line 64
ORA-06512: at "SYSTEM.DISTRIBUTIE_PACIENTI", line 6
ORA-06512: at line 4

```

```
select * from angajati;
rollback;
```

```
select distinct id_asistent, id_pacient from ingrijesc;
```

```
SELECT distinct id_pacient FROM ingrijesc WHERE id_asistent = 41;
```

```
declare
    val number;
begin
    val := distributie_pacienti.pac_asistente ('Ratusanu'); -- id 41 As Ratusanu - are 2 pacienti
    dbms_output.put_line ('Rezultat: ' || val || ' pacienti');
end;
/
```

Worksheet Query Builder

```

317
318 declare
319     val number;
320 begin
321     val := distributie_pacienti.pac_asistente ('Ratusanu'); -- id 41 As Ratusanu - are 2 pacienti
322     dbms_output.put_line ('Rezultat: ' || val || ' pacienti');
323 end;
324 /
325
326 declare

```

Script Output x | Query Result x

PL/SQL procedure successfully completed.

Andreea_SGBD x | Rezultat: 2 pacienti

```

declare
    val number;
begin
    val := distributie_pacienti.pac_asistente ('Racu'); --Eroare: ang cu id 107 nu este ASISTENT
    dbms_output.put_line ('Rezultat: ' || val || ' pacienti');
end;
/

```

Worksheet Query Builder

```

326 declare
327     val number;
328 begin
329     val := distributie_pacienti.pac_asistente ('Racu'); --Eroare: ang cu id 107 nu este ASISTENT
330     dbms_output.put_line ('Rezultat: ' || val || ' pacienti');
331 end;
332 /
333
334 declare
335     val number;

```

Script Output x | Query Result x

PL/SQL procedure successfully completed.

Andreea_SGBD x | Angajatul NU este asistent!

Error starting at line : 326 in command -

```

declare
    val number;
begin
    val := distributie_pacienti.pac_asistente ('Racu'); --Eroare: ang cu id 107 nu este ASISTENT
    dbms_output.put_line ('Rezultat: ' || val || ' pacienti');
end;
Error report -
ORA-20002: Angajatul NU este asistent!
ORA-06512: at "SYSTEM.DISTRIBUTIE_PACIENTI", line 137
ORA-06512: at "SYSTEM.DISTRIBUTIE_PACIENTI", line 75
ORA-06512: at line 4

```

```

declare
    val number;
begin
    val := distributie_pacienti.pac_asistente ('Ancab'); --Eroare: Nu exista asistent cu numele dat
    dbms_output.put_line ('Rezultat: ' || val || ' pacienti');
end;
/

```

```

334 declare
335   val number;
336 begin
337   val := distributie_pacienti.pac_asistente ('Ancab'); --Eroare: Nu exista asistent cu numele dat
338   dbms_output.put_line ('Rezultat: ' || val || ' pacienti');
339 end;
340 /
341
342
343

```

Script Output | Query Result | Task completed in 0.04 seconds

```

val number;
begin
  val := distributie_pacienti.pac_asistente ('Ancab'); --Eroare: Nu exista asistent cu numele dat
  dbms_output.put_line ('Rezultat: ' || val || ' pacienti');
end;
Error report -
ORA-20000: Nu exista asistent cu numele dat
ORA-06512: at "SYSTEM.DISTRIBUTIE_PACIENTI", line 124
ORA-06512: at "SYSTEM.DISTRIBUTIE_PACIENTI", line 75
ORA-06512: at line 4
20000. 00000 - "%s"
*Cause: The stored procedure 'raise_application_error'
  was called which causes this error to be generated.
>Action: Correct the problem as described in the error message or contact
  the application administrator or DBA for more information.

```

```

update angajati set nume = 'Iacob' where id_angajat = 38;
select * from angajati where id_angajat = 37 or id_angajat = 38; -- 2 asistenti cu acest nume

```

```

declare
  val number;
begin
  val := distributie_pacienti.pac_asistente ('Iacob'); -- Eroare: 2 asistenti cu acest nume
  dbms_output.put_line ('Rezultat: ' || val || ' pacienti');
end;
/

```

```

update angajati set nume = 'Iacob' where id_angajat = 38;

select * from angajati where id_angajat = 37 or id_angajat = 38; -- 2 asistenti cu acest nume

declare
  val number;
begin
  val := distributie_pacienti.pac_asistente ('Iacob'); --Eroare: 2 asistenti cu acest nume
  dbms_output.put_line ('Rezultat: ' || val || ' pacient');
end;
Error report -
ORA-20001: Mai multi asistenti cu numele dat
ORA-06512: at "SYSTEM.DISTRIBUTIE_PACIENTI", line 133
ORA-06512: at "SYSTEM.DISTRIBUTIE_PACIENTI", line 75
ORA-06512: at line 4

```

Script Output | Query Result | Task completed in 0.072 seconds
