

Universitatea Babeș-Bolyai
Facultatea de Științe Economice și Gestiunea Afacerilor
Proiect la Elemente Avansate de Baze de Date

SPITAL

Student: Curcean Teodor-Iulian
Specializare: Informatică Economică
Anul 3, Grupa 2, 2022/2023

Cuprins:

Crearea și popularea tabelelor.....	4
Creare tabelă PACIENTI.....	4
Creare tabelă ASISTENTE.....	5
Creare tabelă SALOANE.....	5
Creare tabelă ISTORIC_PACIENTI	5
Creare tabelă DOCTORI	5
Creare tabelă SECRETARE	6
Creare tabelă PROGRAMARI	6
Populare tabelă ANGAJATI.....	6
Populare tabelă SECRETARE.....	8
Populare tabelă DOCTORI.....	8
Populare tabelă ASISTENTE	9
Populare tabelă PACIENTI	9
Populare tabelă SALOANE.....	9
Populare tabelă ISTORIC_PACIENTI.....	10
Populare tabelă PROGRAMARI	10
Package Spital.....	12
Package HEAD	12
Package BODY.....	13
Triggers și secvențe.....	24
Secvențe	24
Triggers	24

Introducere

Un spital este o instituție foarte importanta pentru toți oamenii atunci când vine vorba de sănătate. Acest proiect își propune să vină în ajutorul atât al medicilor, cât și al pacienților atunci când vine vorba de programări, internare și chiar gestionarea unor conturi. Un spital are zilnic un număr mare de pacienți, fiecare venind dintr-un anumit motiv sau pentru o anumită problemă.

Un spital privat din Cluj-Napoca are nevoie de o mai bună organizare a programărilor și a camerelor în care pacienții sunt internați. Astfel, prin intermediul acestei baze de date aceste lucruri vor fi analizate și procesate în cel mai optim mod. De asemenea, se va realiza gestionarea activității medicilor și a programărilor pacienților într-un mod cât mai eficient. Pentru a realiza acest lucru avem nevoie de următoarele entități: angajat, doctor, secretara, asistenta, programare, salon, istoric pacient și pacient.

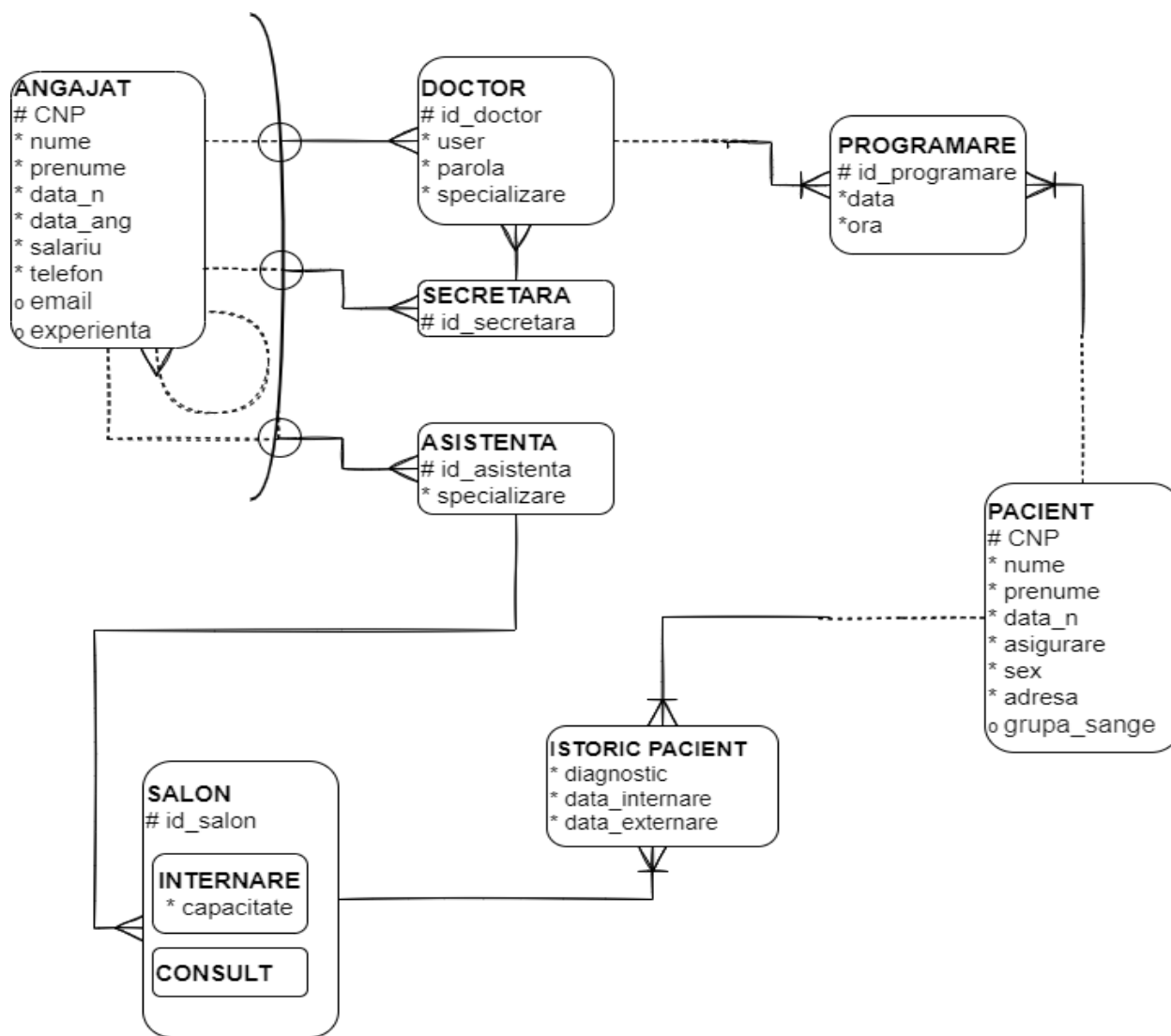


Fig. 1 – Diagrama ERD a bazei de date

Crearea și popularea tabelelor

Creare tabelă ANGAJATI

```
CREATE TABLE ANGAJATI (CNP CHAR(13) PRIMARY KEY NOT NULL,  
                        nume VARCHAR2(20) NOT NULL,  
                        prenume VARCHAR2(20) NOT NULL,  
                        data_n DATE NOT NULL,  
                        data_ang DATE NOT NULL,  
                        salariu NUMBER(4) NOT NULL,  
                        telefon CHAR(10) NOT NULL,  
                        email VARCHAR2(30),  
                        experienta NUMBER(2),  
                        CNP_drr CHAR(13),  
                        FOREIGN KEY (CNP_drr) REFERENCES ANGAJATI(CNP));  
  
ALTER TABLE ANGAJATI  
ADD CONSTRAINT nastere_inainte_angajare  
CHECK (data_n < data_ang);
```

Creare tabelă PACIENTI

```
CREATE TABLE PACIENTI (CNP CHAR(13) PRIMARY KEY NOT NULL,  
                        nume VARCHAR2(20) NOT NULL,  
                        prenume VARCHAR2(20) NOT NULL,  
                        data_n DATE NOT NULL,  
                        asigurare CHAR(2) NOT NULL,  
                        sex CHAR(1) NOT NULL,  
                        adresa VARCHAR2(30) NOT NULL,  
                        grupa_sange VARCHAR2(4));  
  
ALTER TABLE PACIENTI  
ADD CONSTRAINT sex_m_sau_f  
CHECK(sex = 'M' OR sex = 'F');
```

Creare tabelă ASISTENTE

```
CREATE TABLE ASISTENTE ( id_asistenta NUMBER(3) PRIMARY KEY NOT NULL,  
                           specializare VARCHAR2(20) NOT NULL,  
                           CNP_agt CHAR(13) NOT NULL,  
                           FOREIGN KEY (CNP_agt) REFERENCES ANGAJATI(CNP));
```

Creare tabelă SALOANE

```
CREATE TABLE SALOANE ( id_salon CHAR(3) PRIMARY KEY NOT NULL,  
                        tip_sln CHAR(1) NOT NULL,  
                        capacitate NUMBER(2),  
                        id_asistenta_asa NUMBER(3) NOT NULL,  
                        FOREIGN KEY (id_asistenta_asa) REFERENCES ASISTENTE(id_asistenta));
```

Creare tabelă ISTORIC_PACIENTI

```
CREATE TABLE ISTORIC_PACIENTI ( diagnostic VARCHAR2(40) NOT NULL,  
                                   data_internare DATE NOT NULL,  
                                   data_externare DATE NOT NULL,  
                                   CNP_pct CHAR(13) NOT NULL,  
                                   id_salon_sln CHAR(3) NOT NULL,  
                                   PRIMARY KEY(CNP_pct, id_salon_sln),  
                                   FOREIGN KEY (CNP_pct) REFERENCES PACIENTI(CNP),  
                                   FOREIGN KEY (id_salon_sln) REFERENCES SALOANE(id_salon));
```

Creare tabelă DOCTORI

```
CREATE TABLE DOCTORI ( id_doctor NUMBER(2) PRIMARY KEY NOT NULL,  
                        utilizator VARCHAR2(20) NOT NULL,  
                        parola VARCHAR2(20) NOT NULL,  
                        specializare VARCHAR2(20) NOT NULL,  
                        CNP_agt CHAR(13) NOT NULL,
```

```

        id_secretara_sca NUMBER(3) NOT NULL,
        FOREIGN KEY (CNP_agt) REFERENCES ANGAJATI(CNP),
        FOREIGN KEY (id_secretara_sca) REFERENCES SECRETARE(id_secretara));

```

ALTER TABLE DOCTORI

ADD CV CLOB;

Creare tabelă SECRETARE

```

CREATE TABLE SECRETARE ( id_secretara NUMBER(3) PRIMARY KEY NOT NULL,
                           CNP_agt CHAR(13) NOT NULL,
                           FOREIGN KEY (CNP_agt) REFERENCES ANGAJATI(CNP));

```

Creare tabelă PROGRAMARI

```

CREATE TABLE PROGRAMARI ( id_programare NUMBER(6) NOT NULL,
                           data DATE NOT NULL,
                           ora DATE NOT NULL,
                           id_doctor_dtr NUMBER(2) NOT NULL,
                           CNP_pct CHAR(13) NOT NULL,
                           PRIMARY KEY(id_programare, CNP_pct, id_doctor_dtr),
                           FOREIGN KEY (CNP_pct) REFERENCES PACIENTI(CNP),
                           FOREIGN KEY (id_doctor_dtr) REFERENCES DOCTORI(id_doctor));

```

Populare tabelă ANGAJATI

- ➔ INSERT INTO ANGAJATI VALUES ('2701002098245', 'Pop', 'Mara',
TO_DATE('20/10/1970','dd/mm/yyyy'), TO_DATE('20/1/2000','dd/mm/yyyy'), 4200,
'0749522792', 'pmara@gmail.com', 4, NULL);
- ➔ INSERT INTO ANGAJATI VALUES ('2701225229770', 'Titei', 'Nicoleta',
TO_DATE('25/12/1970','dd/mm/yyyy'), TO_DATE('29/1/2013','dd/mm/yyyy'), 8600,
'0745314149', 'tnicoleta@gmail.com', 7, NULL);
- ➔ INSERT INTO ANGAJATI VALUES ('2911214081477', 'Stan', 'Cristina',
TO_DATE('14/12/1971','dd/mm/yyyy'), TO_DATE('26/1/2010','dd/mm/yyyy'), 4200,
'0744957148', 'scristina@gmail.com', 5, NULL);

➔ INSERT INTO ANGAJATI VALUES ('1820521368762', 'Barbu', 'Ion',
TO_DATE('21/5/1982', 'dd/mm/yyyy'), TO_DATE('21/4/2000', 'dd/mm/yyyy'), 3600,
'0723680627', 'bion@gmail.com', 6, '2701225229770');

➔ INSERT INTO ANGAJATI VALUES ('2790622495943', 'Robu', 'Cristiana',
TO_DATE('22/6/1979', 'dd/mm/yyyy'), TO_DATE('22/3/2000', 'dd/mm/yyyy'), 3700,
'0756035178', 'rcristina@gmail.com', 3, '2701002098245');

➔ INSERT INTO ANGAJATI VALUES ('2741123184585', 'Barbu', 'Ana',
TO_DATE('23/11/1974', 'dd/mm/yyyy'), TO_DATE('14/6/2011', 'dd/mm/yyyy'), 6400,
'0788018388', NULL, 2, '2701002098245');

➔ INSERT INTO ANGAJATI VALUES ('2800124529844', 'Baciu', 'Daniela',
TO_DATE('24/1/1980', 'dd/mm/yyyy'), TO_DATE('30/12/2010', 'dd/mm/yyyy'), 7400,
'0745597951', 'bdaniela@gmail.com', 4, '2701225229770');

➔ INSERT INTO ANGAJATI VALUES ('1720326032009', 'David', 'Marcus',
TO_DATE('26/3/1972', 'dd/mm/yyyy'), TO_DATE('26/1/2010', 'dd/mm/yyyy'), 4300,
'0749288422', 'dmarcus@gmail.com', 2, '2701225229770');

➔ INSERT INTO ANGAJATI VALUES ('1820427902154', 'Serban', 'Ioan',
TO_DATE('27/4/1982', 'dd/mm/yyyy'), TO_DATE('27/5/2012', 'dd/mm/yyyy'), 5400,
'0749288423', 'sioan@gmail.com', 4, '2701225229770');

➔ INSERT INTO ANGAJATI VALUES ('170102875117', 'Vrajitoru', 'Nae',
TO_DATE('28/10/1970', 'dd/mm/yyyy'), TO_DATE('28/4/2009', 'dd/mm/yyyy'), 5900,
'0721286383', 'vnae@gmail.com', 3, '2701225229770');

➔ INSERT INTO ANGAJATI VALUES ('1880726010287', 'Rusu', 'George',
TO_DATE('26/7/1988', 'dd/mm/yyyy'), TO_DATE('20/1/2000', 'dd/mm/yyyy'), 3500,
'0762609709', 'rgeorge@gmail.com', 2, '2701225229770');

➔ INSERT INTO ANGAJATI VALUES ('1890225039752', 'Timis', 'Radu',
TO_DATE('25/2/1989', 'dd/mm/yyyy'), TO_DATE('21/4/2000', 'dd/mm/yyyy'), 3430,
'0722595682', 'tradu@gmail.com', NULL, '2701225229770');

➔ INSERT INTO ANGAJATI VALUES ('2941214142209', 'Todean', 'Delia',
TO_DATE('14/12/1994', 'dd/mm/yyyy'), TO_DATE('22/3/2019', 'dd/mm/yyyy'), 5400,
'0742082982', 'tdelia@gmail.com', 1, '2701002098245');

➔ INSERT INTO ANGAJATI VALUES ('2930409372433', 'Marginean', 'Andreea',
TO_DATE('9/4/1993', 'dd/mm/yyyy'), TO_DATE('14/6/2018', 'dd/mm/yyyy'), 4200,
'0726505305', 'mandreea@gmail.com', NULL, '2701002098245');

➔ INSERT INTO ANGAJATI VALUES ('2870415349451', 'Suciu', 'Ildiko',
TO_DATE('15/4/1987', 'dd/mm/yyyy'), TO_DATE('30/12/2010', 'dd/mm/yyyy'), 2870,
'0745754724', 'sildiko@gmail.com', 3, '2701002098245');

➔ INSERT INTO ANGAJATI VALUES ('2890129360697', 'Muntean', 'Debora',
TO_DATE('29/1/1989', 'dd/mm/yyyy'), TO_DATE('29/1/2013', 'dd/mm/yyyy'), 3700,
'0722858673', 'mdebora@gmail.com', 2, '2911214081477');

➔ INSERT INTO ANGAJATI VALUES ('2870102515115', 'Szabo', 'Irina',
TO_DATE('2/1/1987', 'dd/mm/yyyy'), TO_DATE('27/5/2012', 'dd/mm/yyyy'), 3900,
'0721205580', 'sirina@gmail.com', NULL, '2911214081477');

- ➔ INSERT INTO ANGAJATI VALUES ('1870308014200', 'Andreica', 'Stefan', TO_DATE('8/3/1987', 'dd/mm/yyyy'), TO_DATE('28/4/2009', 'dd/mm/yyyy'), 4800, '0722365248', 'astefan@gmail.com', 2, '2701225229770');
- ➔ INSERT INTO ANGAJATI VALUES ('2901024152754', 'Popa', 'Angela', TO_DATE('24/10/1990', 'dd/mm/yyyy'), TO_DATE('20/1/2000', 'dd/mm/yyyy'), 6500, '0745151130', 'pangela@gmail.com', NULL, '2911214081477');
- ➔ INSERT INTO ANGAJATI VALUES ('2870207250053', 'Costea', 'Adelina', TO_DATE('7/2/1987', 'dd/mm/yyyy'), TO_DATE('21/4/2000', 'dd/mm/yyyy'), 6100, '0788123781', 'cadelina@gmail.com', 1, '2911214081477');
- ➔ INSERT INTO ANGAJATI VALUES ('2910115119447', 'Popescu', 'Daria', TO_DATE('15/1/1991', 'dd/mm/yyyy'), TO_DATE('22/3/2000', 'dd/mm/yyyy'), 3500, '0744530067', 'pdaria@gmail.com', NULL, '2911214081477');

Populare tabelă SECRETARE

- ➔ INSERT INTO SECRETARE VALUES (111, '2790622495943');
- ➔ INSERT INTO SECRETARE VALUES (112, '2741123184585');
- ➔ INSERT INTO SECRETARE VALUES (113, '2701002098245');
- ➔ INSERT INTO SECRETARE VALUES (114, '2941214142209');
- ➔ INSERT INTO SECRETARE VALUES (115, '2930409372433');
- ➔ INSERT INTO SECRETARE VALUES (116, '2870415349451');

Populare tabelă DOCTORI

- ➔ INSERT INTO DOCTORI VALUES (1, 'dbaciu', 'dbaciu123', 'cardiologie', '2800124529844', 111);
- ➔ INSERT INTO DOCTORI VALUES (2, 'ntitei', 'ntitei123', 'chirurgie', '2701225229770', 116);
- ➔ INSERT INTO DOCTORI VALUES (3, 'mdavid', 'mdavid123', 'dermatologie', '1720326032009', 114);
- ➔ INSERT INTO DOCTORI VALUES (4, 'nvrajitoru', 'nvrajitoru123', 'pediatrie', '170102875117', 112);
- ➔ INSERT INTO DOCTORI VALUES (5, 'iserban', 'iserban123', 'ginecologie', '1820427902154', 113);
- ➔ INSERT INTO DOCTORI VALUES (6, 'grusu', 'grusu123', 'neurologie', '1880726010287', 115);
- ➔ INSERT INTO DOCTORI VALUES (7, 'rtimis', 'rtimis123', 'psihiatrie', '1890225039752', 111);
- ➔ INSERT INTO DOCTORI VALUES (8, 'ibarbu', 'ibarbu123', 'chirurgie', '1820521368762', 113);
- ➔ INSERT INTO DOCTORI VALUES (9, 'sandreica', 'sandreica123', 'ginecologie', '1870308014200', 114);

Populare tabelă ASISTENTE

- ➔ INSERT INTO ASISTENTE VALUES (221, 'chirurgie', '2890129360697');
- ➔ INSERT INTO ASISTENTE VALUES (222, 'cardiologie', '2911214081477');
- ➔ INSERT INTO ASISTENTE VALUES (223, 'dermatologie', '2870102515115');
- ➔ INSERT INTO ASISTENTE VALUES (224, 'neurologie', '2901024152754');
- ➔ INSERT INTO ASISTENTE VALUES (225, 'ginecologie', '2870207250053');
- ➔ INSERT INTO ASISTENTE VALUES (226, 'psihiatrie', '2910115119447');

Populare tabelă PACIENTI

- ➔ INSERT INTO PACIENTI VALUES ('1880717375061', 'Pop', 'Sabin', TO_DATE('17/7/1998','dd/mm/yyyy'), 'DA', 'M', 'str. Memo', 'OI');
- ➔ INSERT INTO PACIENTI VALUES ('1990912246794', 'Muresan', 'Eric', TO_DATE('12/9/1999','dd/mm/yyyy'), 'DA', 'M', 'str. Ion Barbu', 'AII');
- ➔ INSERT INTO PACIENTI VALUES ('2981029242005', 'Popa', 'Gina', TO_DATE('29/10/1998','dd/mm/yyyy'), 'NU', 'F', 'str. Gusteritei', 'ABIV');
- ➔ INSERT INTO PACIENTI VALUES ('1920628431862', 'Dumitru', 'Tudor', TO_DATE('28/6/1992','dd/mm/yyyy'), 'DA', 'M', 'str. Constanta', 'BIII');
- ➔ INSERT INTO PACIENTI VALUES ('6011125374844', 'Dinu', 'Iulia', TO_DATE('25/11/2001','dd/mm/yyyy'), 'NU', 'F', 'str. Fistic', NULL);
- ➔ INSERT INTO PACIENTI VALUES ('1971113164107', 'Mocanu', 'Matei', TO_DATE('13/11/1997','dd/mm/yyyy'), 'NU', 'M', 'str. Mica', 'OI');
- ➔ INSERT INTO PACIENTI VALUES ('1920119367325', 'Rotaru', 'Luca', TO_DATE('19/1/1992','dd/mm/yyyy'), 'DA', 'M', 'str. Vasile A', 'ABIV');
- ➔ INSERT INTO PACIENTI VALUES ('5030324448180', 'Cirdei', 'Raul', TO_DATE('24/3/2003','dd/mm/yyyy'), 'DA', 'M', 'str. Castanilor', 'BIII');
- ➔ INSERT INTO PACIENTI VALUES ('2950323358488', 'Nagy', 'Cristina', TO_DATE('23/3/1995','dd/mm/yyyy'), 'DA', 'F', 'str. Alunelor', 'OI');

Populare tabelă SALOANE

- ➔ INSERT INTO SALOANE VALUES ('S01', 'T', 4, 224);
- ➔ INSERT INTO SALOANE VALUES ('S02', 'T', 3, 226);
- ➔ INSERT INTO SALOANE VALUES ('S03', 'C', NULL, 222);
- ➔ INSERT INTO SALOANE VALUES ('S04', 'C', NULL, 223);
- ➔ INSERT INTO SALOANE VALUES ('S05', 'T', 2, 225);
- ➔ INSERT INTO SALOANE VALUES ('S06', 'C', NULL, 226);
- ➔ INSERT INTO SALOANE VALUES ('S07', 'T', 1, 221);
- ➔ INSERT INTO SALOANE VALUES ('S08', 'T', 4, 225);
- ➔ INSERT INTO SALOANE VALUES ('S09', 'C', NULL, 225);
- ➔ INSERT INTO SALOANE VALUES ('S10', 'T', 3, 224);
- ➔ INSERT INTO SALOANE VALUES ('S11', 'C', NULL, 226);

Populare tabelă ISTORIC_PACIENTI

- ➔ INSERT INTO ISTORIC_PACIENTI VALUES ('cardiopatie ischemica', TO_DATE('25/1/2022','dd/mm/yyyy'), TO_DATE('30/1/2022','dd/mm/yyyy'), '1920628431862', 'S02');
- ➔ INSERT INTO ISTORIC_PACIENTI VALUES ('consult ginecologie', TO_DATE('14/3/2022','dd/mm/yyyy'), TO_DATE('14/3/2022','dd/mm/yyyy'), '2950323358488', 'S11');
- ➔ INSERT INTO ISTORIC_PACIENTI VALUES ('AVC', TO_DATE('11/4/2019','dd/mm/yyyy'), TO_DATE('26/5/2019','dd/mm/yyyy'), '1880717375061', 'S07');
- ➔ INSERT INTO ISTORIC_PACIENTI VALUES ('alzheimer', TO_DATE('18/6/2021','dd/mm/yyyy'), TO_DATE('18/6/2021','dd/mm/yyyy'), '1880717375061', 'S04');
- ➔ INSERT INTO ISTORIC_PACIENTI VALUES ('migrena', TO_DATE('8/2/2022','dd/mm/yyyy'), TO_DATE('8/2/2022','dd/mm/yyyy'), '6011125374844', 'S09');
- ➔ INSERT INTO ISTORIC_PACIENTI VALUES ('hidrocefalie', TO_DATE('2/5/2022','dd/mm/yyyy'), TO_DATE('11/5/2022','dd/mm/yyyy'), '1920119367325', 'S10');
- ➔ INSERT INTO ISTORIC_PACIENTI VALUES ('cretinism', TO_DATE('10/9/2021','dd/mm/yyyy'), TO_DATE('15/9/2021','dd/mm/yyyy'), '5030324448180', 'S07');

Populare tabelă PROGRAMARI

- ➔ INSERT INTO PROGRAMARI VALUES (188796, TO_DATE('14/9/2022','dd/mm/yyyy'), TO_DATE('14:30','hh24:mi'), 5, '2981029242005');
- ➔ INSERT INTO PROGRAMARI VALUES (154975, TO_DATE('14/9/2022','dd/mm/yyyy'), TO_DATE('15:15','hh24:mi'), 8, '1880717375061');
- ➔ INSERT INTO PROGRAMARI VALUES (124769, TO_DATE('25/7/2022','dd/mm/yyyy'), TO_DATE('11:45','hh24:mi'), 2, '1990912246794');
- ➔ INSERT INTO PROGRAMARI VALUES (186432, TO_DATE('30/6/2022','dd/mm/yyyy'), TO_DATE('14:45','hh24:mi'), 1, '1920628431862');
- ➔ INSERT INTO PROGRAMARI VALUES (195648, TO_DATE('4/6/2022','dd/mm/yyyy'), TO_DATE('10:15','hh24:mi'), 1, '6011125374844');
- ➔ INSERT INTO PROGRAMARI VALUES (165785, TO_DATE('16/7/2022','dd/mm/yyyy'), TO_DATE('9:30','hh24:mi'), 3, '1971113164107');

- ➔ INSERT INTO PROGRAMARI VALUES (125498,
TO_DATE('20/7/2022','dd/mm/yyyy'), TO_DATE('16:15','hh24:mi'), 8,
'1920119367325');
- ➔ INSERT INTO PROGRAMARI VALUES (165321,
TO_DATE('14/8/2022','dd/mm/yyyy'), TO_DATE('12:45','hh24:mi'), 9,
'5030324448180');
- ➔ INSERT INTO PROGRAMARI VALUES (145987,
TO_DATE('26/8/2022','dd/mm/yyyy'), TO_DATE('13:30','hh24:mi'), 7,
'2950323358488');
- ➔ INSERT INTO PROGRAMARI VALUES (121585,
TO_DATE('8/9/2022','dd/mm/yyyy'), TO_DATE('15:00','hh24:mi'), 6,
'1880717375061');
- ➔ INSERT INTO PROGRAMARI VALUES (164235,
TO_DATE('23/9/2022','dd/mm/yyyy'), TO_DATE('14:00','hh24:mi'), 6,
'6011125374844');

Package Spital

Proceduri publice:

- ADD_DOCTOR
- AFF_PACIENTI_GRUPA_SANGE
- PROGRAM_DETAILS
- UPDATE_DOCTOR_SALARIES

Proceduri private:

- ADD_TBL_ANG
- ADD_TBL_DCT
- DOCTOR_INFO

Funcții publice:

- MAJORARE_SALARIU_ANG
- MAJORARE_SALARIU_DRR

Funcții private:

- GET_CNP_DRR
- GET_EMAIL
- GET_NEXT_ID

Package HEAD:

create or replace PACKAGE spital AS

-- Public variables

MIN_SAL NUMBER :=2000;

MAX_SAL NUMBER :=15000;

-- Public functions

FUNCTION majorare_salariu_ang RETURN NUMBER;

FUNCTION majorare_salariu_drr RETURN NUMBER;

-- Public procedures

PROCEDURE add_doctor(

```

v_CNP IN angajati.CNP%TYPE,
v_numa IN angajati.numa%TYPE,
v_prenume IN angajati.prenume%TYPE,
v_data_n IN angajati.data_n%TYPE,
v_salariu IN angajati.salariu%TYPE,
v_telefon IN angajati.telefon%TYPE,
v_experienta IN angajati.experienta%TYPE,
v_specializare IN doctori.specializare%TYPE,
v_id_secretara IN doctori.id_secretara_sca%TYPE
);

PROCEDURE aff_pacienti_grupa_sange(
    v_grupa_sange IN pacienti.grupa_sange%TYPE
);

PROCEDURE program_details(
    patient_cnp IN programari.CNP_pct%TYPE
);

PROCEDURE update_doctor_salaries(
    p_salary_increase NUMBER
);

END spital;

```

Package BODY:

```

create or replace PACKAGE BODY spital AS
-- Private functions and procedures

FUNCTION get_next_id
RETURN doctori.id_doctor%TYPE
IS
    v_id doctori.id_doctor%TYPE;

```

```

BEGIN

    SELECT max(id_doctor)

    INTO v_id

    FROM doctori;

    RETURN v_id+1;

END get_next_id;

PROCEDURE doctor_info(

    id_doctor_t IN programari.id_doctor_dtr%TYPE

) IS

    v_numa angajati.numa%TYPE;

    v_pnuma angajati.pnuma%TYPE;

BEGIN

    SELECT a.numa,a.pnuma INTO v_numa,v_pnuma

    FROM doctori d JOIN angajati a ON d.CNP_agt = a.CNP

    WHERE d.id_doctor = id_doctor_t;

    DBMS_OUTPUT.PUT_LINE( 'Medicul la care este programat: ' || v_numa || ' ' ||

v_pnuma);

    END;

    FUNCTION get_email (v_numa IN angajati.numa%TYPE, v_pnuma IN

angajati.pnuma%TYPE) RETURN VARCHAR2

    IS

        v_email angajati.email%TYPE;

BEGIN

    v_email := CONCAT(substr(lower(v_numa),1,1),lower(v_pnuma));

    v_email := CONCAT(v_email, '@gmail.com');

    RETURN v_email;

END get_email;

FUNCTION get_cnp_drr RETURN angajati.CNP_drr%TYPE

```

IS

```
v_cnp_drr angajati.CNP_drr%TYPE;  
BEGIN  
    SELECT a.CNP  
    INTO v_cnp_drr  
    FROM angajati a JOIN doctori d ON a.CNP = d.CNP_agt  
    WHERE a.CNP_drr IS NULL AND d.CNP_agt=a.CNP;  
    RETURN v_cnp_drr;  
END get_cnp_drr;
```

```
PROCEDURE add_tbl_ang(  
    v_CNP IN angajati.CNP%TYPE,  
    v_nume IN angajati.nume%TYPE,  
    v_prenume IN angajati.prenume%TYPE,  
    v_data_n IN angajati.data_n%TYPE,  
    v_salariu IN angajati.salariu%TYPE,  
    v_telefon IN angajati.telefon%TYPE,  
    v_experienta IN angajati.experienta%TYPE  
) IS  
    v_email angajati.email%TYPE;  
    v_cnp_drr angajati.CNP_drr%TYPE;  
BEGIN  
    v_email := get_email(v_nume, v_prenume);  
    v_cnp_drr := get_cnp_drr();  
    INSERT INTO angajati VALUES (v_CNP, v_nume, v_prenume, v_data_n,  
    SYSDATE, v_salariu, v_telefon, v_email, v_experienta, v_cnp_drr);  
END add_tbl_ang;
```

```
PROCEDURE add_tbl_dct(
```

```

v_CNP IN angajati.CNP%TYPE,
v_nume IN angajati.nume%TYPE,
v_prenume IN angajati.prenume%TYPE,
v_specializare IN doctori.specializare%TYPE
) IS
v_utilizator doctori.utilizator%TYPE;
v_id doctori.id_doctor%TYPE;
BEGIN
    v_id := get_next_id();
    v_utilizator := CONCAT(LOWER(SUBSTR(v_prenume, 1, 1)), LOWER(v_nume));
    INSERT INTO doctori VALUES (v_id, v_utilizator, CONCAT(v_utilizator,'123'),
v_specializare, v_CNP, dbms_random.value(111,116));
END add_tbl_dct;

```

-- Declare functions

```

FUNCTION majorare_salariu_ang RETURN NUMBER IS
BEGIN
    UPDATE angajati
    SET salariu = salariu + 100
    WHERE salariu > MIN_SAL AND salariu < MAX_SAL;
RETURN sql%ROWCOUNT;
END majorare_salariu_ang;

```

```

FUNCTION majorare_salariu_drr RETURN NUMBER IS
    v_numar NUMBER;
BEGIN
    UPDATE angajati
    SET salariu = salariu + 100
    WHERE CNP_drr IS NULL;

```



```

    RETURN sql%ROWCOUNT;

END majorare_salariu_drr;

-- Declare procedures

PROCEDURE add_doctor(
    v_CNP IN angajati.CNP%TYPE,
    v_numa IN angajati.numa%TYPE,
    v_prename IN angajati.prename%TYPE,
    v_data_n IN angajati.data_n%TYPE,
    v_salariu IN angajati.salariu%TYPE,
    v_telefon IN angajati.telefon%TYPE,
    v_experienta IN angajati.experienta%TYPE,
    v_specializare IN doctori.specializare%TYPE,
    v_id_secretara IN doctori.id_secretara_sca%TYPE
) IS
BEGIN
    add_tbl_ang( v_CNP, v_numa, v_prename, v_data_n, v_salariu, v_telefon,
v_experienta);
    add_tbl_dct(v_CNP, v_numa, v_prename, v_specializare);
END add_doctor;

PROCEDURE aff_pacienti_grupa_sange(
    v_grupa_sange IN pacienti.grupa_sange%TYPE
) IS
CURSOR pacienti_crs(v_grupa_sange pacienti.grupa_sange%TYPE) IS
    SELECT nume,prename
    FROM pacienti
    WHERE grupa_sange=UPPER(v_grupa_sange);
v_numa pacienti.numa%TYPE;

```

```

v_prenume pacienti.prenume%TYPE;

BEGIN

OPEN pacienti_crs(v_grupa_sange);

dbms_output.put_line('Pacientii cu grupa de sange'|| ' '|| v_grupa_sange||' sunt: ');

LOOP

    FETCH pacienti_crs

    INTO v_nume, v_prenume;

    EXIT WHEN pacienti_crs%NOTFOUND;

    dbms_output.put_line(v_nume ||' '||v_prenume);

END LOOP;

END aff_pacienti_grupa_sange;


PROCEDURE program_details(
    pacient_cnp IN programari.CNP_pct%TYPE
) IS

    CURSOR progr_cur IS

        SELECT p.data AS data_progr,p.ora AS ora_progr,p.id_doctor_dtr,pa.nume AS
nume_pct,pa.prenume AS prenume_pct

        FROM programari p JOIN pacienti pa ON p.CNP_pct = pa.CNP

        WHERE p.CNP_pct = pacient_cnp;

    TYPE progr_rec_type IS RECORD(data_progr programari.data%TYPE,
                                ora_progr programari.data%TYPE,
                                id_doctor_dtr programari.id_doctor_dtr%TYPE,
                                nume_pct pacienti.nume%TYPE,
                                prenume_pct pacienti.prenume%type);

    progr_rec progr_rec_type;

BEGIN

    OPEN progr_cur;

    LOOP

```

```

    FETCH progr_cur INTO progr_rec;

    EXIT WHEN progr_cur%NOTFOUND;

    DBMS_OUTPUT.PUT_LINE( 'Programarea pentru CNP-ul ' || pacient_cnp || ' este: ');

    DBMS_OUTPUT.PUT_LINE( 'Nume pacient: ' || progr_rec.ume_pct || ' ' ||
progr_rec.prenume_pct);

    DBMS_OUTPUT.PUT_LINE( 'Data si ora: ' || progr_rec.data_progr|| ', ' ||
TO_CHAR(progr_rec.ora_progr,'hh24:mm'));

    doctor_info(progr_rec.id_doctor_dtr);

    DBMS_OUTPUT.PUT_LINE(
'*****');

END LOOP;

IF(progr_cur%ROWCOUNT = 0) THEN

    DBMS_OUTPUT.PUT_LINE('Acest pacient nu are o programare!');

END IF;

CLOSE progr_cur;

END;

```

```

PROCEDURE update_doctor_salaries (p_salary_increase NUMBER)

```

```

IS

```

```

    l_doctor_ids SYS_REFCURSOR;
    l_doctor_salaries SYS_REFCURSOR;
    l_doctor_id NUMBER;
    l_doctor_cnp CHAR(13);
    l_doctor_salary NUMBER;

```

```

BEGIN

```

```

-- Open the first cursor

```

```

OPEN l_doctor_ids FOR

```

```

    SELECT id_doctor, CNP_agt
    FROM DOCTORI;

```

```

-- Open the second cursor
OPEN l_doctor_salaries FOR
    SELECT CNP, salariu
    FROM ANGAJATI;

-- Loop through the first cursor
LOOP
    FETCH l_doctor_ids INTO l_doctor_id, l_doctor_cnp;
    EXIT WHEN l_doctor_ids%NOTFOUND;

-- Loop through the second cursor
LOOP
    FETCH l_doctor_salaries INTO l_doctor_cnp, l_doctor_salary;
    EXIT WHEN l_doctor_salaries%NOTFOUND;

-- Update the salary in the second cursor
UPDATE ANGAJATI
    SET salariu = l_doctor_salary + p_salary_increase
    WHERE CNP = l_doctor_cnp;
END LOOP;
END LOOP;

-- Close the cursors
CLOSE l_doctor_ids;
CLOSE l_doctor_salaries;
END;
END spital;

```

- Procedura ADD_DOCTOR are rolul de adăuga un medic atât în tabela angajați cât și în tabela doctori, aceasta apelează două proceduri private ADD_TBL_ANG și ADD_TBL_DCT, care, la rândul lor, apelează funcțiile private GET_EMAIL, GET_CNP_DRR și GET_NEXT_ID :

```
begin
spital.add_doctor('5231687122398','Boian','Claudia', TO_DATE('26/3/1972', 'dd/mm/yyyy'), 3658, '0743090427', 5, 'neonatologie');
end;
```

ID_DOCTOR	UTILIZATOR	PAROLA	SPECIALIZARE	CNP_AGT	ID_SECRETARA_SCA
10	cboian	cboian123	neonatologie	5231687122398	115

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

- Procedura AFF_PACIENTI_GRUPA_SANGE este folosită pentru afișarea pașienților care au o anumită grupă de sânge.

```
begin
spital.aff_pacienti_grupa_sange('OI');
end;
```

Results	Explain	Describe	Saved SQL	History
Pacientii cu grupa de sange OI sunt: Pop Sabin Mocanu Matei Nagy Cristina				

- Procedura PROGRAM_DETAILS are rolul de a afișa detaliile programării (nume pacient, data, ora și medicul la care este făcută programarea), apelând procedura privată DOCTOR_INFO pentru a afișa numele doctorului la care este făcută programarea.

```
begin
spital.program_details('2981029242005');
end;
```

Results	Explain	Describe	Saved SQL	History
Programarea pentru CNP-ul 2981029242005 este: Nume pacient: Popa Gina Data si ora: 14-Sep-2022, 14:10 Medicul la care este programat: Serban Ioan ***** Statement processed.				

- Procedura UPDATE_DOCTOR_SALARIES are rolul de a crește salariul doctorilor cu o sumă introdusă de la tastatură.

CNP	NUME	PRENUME	DATA_N	DATA_ANG	SALARIU	TELEFON	EMAIL	EXPERIENTA	C
2701002098245	Pop	Mara	20-Oct-1970	20-Jan-2000	4300	0749522792	pmara@gmail.com	4	-
2701225229770	Titei	Nicoleta	25-Dec-1970	29-Jan-2013	8700	0745314149	tnicoleta@gmail.com	7	-
2911214081477	Stan	Cristina	14-Dec-1971	26-Jan-2010	4300	0744957148	scristina@gmail.com	5	-
1820521368762	Barbu	Ion	21-May-1982	21-Apr-2000	3600	0723680627	bion@gmail.com	6	270
2790622495943	Robu	Cristiana	22-Jun-1979	22-Mar-2000	3700	0756035178	rcristina@gmail.com	3	270

begin
spital.update_doctor_salaries(200);
end;

Results	Explain	Describe	Saved SQL	History					
CNP	NUME	PRENUME	DATA_N	DATA_ANG	SALARIU	TELEFON	EMAIL	EXPERIENTA	
2701002098245	Pop	Mara	20-Oct-1970	20-Jan-2000	4500	0749522792	pmara@gmail.com	4	-
2701225229770	Titei	Nicoleta	25-Dec-1970	29-Jan-2013	8900	0745314149	tnicoleta@gmail.com	7	-
2911214081477	Stan	Cristina	14-Dec-1971	26-Jan-2010	4500	0744957148	scristina@gmail.com	5	-
1820521368762	Barbu	Ion	21-May-1982	21-Apr-2000	3800	0723680627	bion@gmail.com	6	270
2790622495943	Robu	Cristiana	22-Jun-1979	22-Mar-2000	3900	0756035178	rcristina@gmail.com	3	270

- Funcția MAJORARE SALARIU_DRR crește salariul directorilor cu 100 și returnează numărul de linii afectate.

```
begin
dbms_output.put_line('Salariul a crescut la ' || spital.majorare_salariu_drr() || ' directori.');
```

Results	Explain	Describe	Saved SQL	History
Salariul a crescut la 3 directori.				

- Funcția MAJORARE SALARIU_ANG crește salariul angajaților cu 100 în cazul în care acesta se află între salariul minim și maxim și returnează numărul de linii afectate.

```
begin
  dbms_output.put_line('Salariul a crescut la ' || spital.majorare_salariu_ang() || ' angajati.');
```

```
end;
```

Results

[Explain](#)

[Describe](#)

[Saved SQL](#)

[History](#)

Salariul a crescut la 22 angajati.

Triggers și secvențe

1. Secvențe

```
CREATE SEQUENCE asistant_seq  
START WITH 226  
INCREMENT BY 1;
```

```
CREATE SEQUENCE appointment_seq  
START WITH 1  
INCREMENT BY 1;
```

2. Triggers

a) **appointment_id_trigger** - va genera un ID unic pentru coloana id_programare (ID-ul programării) a înregistrării noi, folosind secvența appointment_id_seq. Valoarea generată va fi atribuită variabilei :new.id_programare. Aceasta asigură că fiecare înregistrare în tabelul PROGRAMARI are un ID unic, care poate fi folosit pentru a identifica în mod unic o programare

```
CREATE OR REPLACE TRIGGER appointment_id_trigger  
BEFORE INSERT ON PROGRAMARI  
FOR EACH ROW  
BEGIN  
    SELECT appointment_id_seq.NEXTVAL INTO :new.id_programare FROM DUAL;  
END;
```

```
select * from programari ORDER BY 1;
```

Results	Explain	Describe	Saved SQL	History
ID_PROGRAMARE			DATA	
1				25-Jul-2023

b) **prevent_past_appointments** – acest trigger verifică dacă în momentul inserării în tabela programări data introdusă este în viitor, în caz contrar acesta va ridica excepția 20001

```
CREATE OR REPLACE TRIGGER prevent_past_appointments
```

```
BEFORE INSERT OR UPDATE ON PROGRAMARI
```

```
FOR EACH ROW
```

```
BEGIN
```

```
IF :new.data < SYSDATE THEN
```

```
RAISE_APPLICATION_ERROR(-20001, 'Appointment date cannot be in the past');
```

```
END IF;
```

```
END;
```

```
INSERT INTO PROGRAMARI VALUES (1887, TO_DATE('14/8/2021','dd/mm/yyyy'), TO_DATE('14:30','hh24:mi'), 5, '2981029242005');
```

Results

Explain

Describe

Saved SQL

History

ORA-20001: Appointment date cannot be in the past

c) **complex_assistant_trigger** - este activat înainte de inserarea sau actualizarea unei înregistrări în tabelul ASISTENTE și are următoarele acțiuni: verifică dacă valoarea coloanei CNP (Codul numeric personal) este unică în tabelul ASISTENTE și generează un ID unic folosind secvența assistant_id_seq și îl atribuie coloanei id_asistenta (ID-ul asistentei) a înregistrării noi sau actualizate

```
CREATE OR REPLACE TRIGGER complex_assistant_trigger
```

```
BEFORE INSERT OR UPDATE ON ASISTENTE
```

```
FOR EACH ROW
```

```
DECLARE
```

```
v_count NUMBER;
```

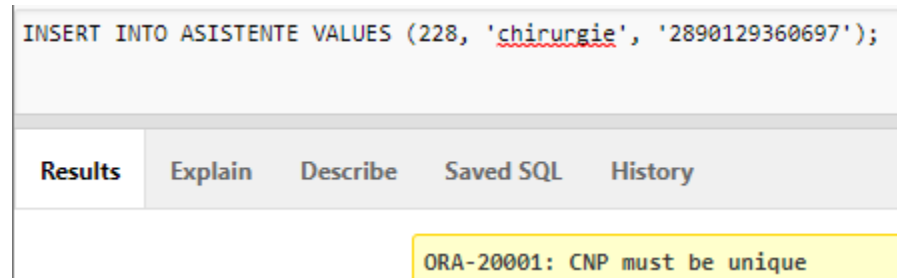
```
BEGIN
```

```
-- Generate a unique ID using the sequence
```

```
SELECT assistant_id_seq.NEXTVAL INTO :new.id_asistenta FROM DUAL;
```

```
-- Check if the CNP value is unique
SELECT COUNT(*) INTO v_count FROM ASISTENTE WHERE CNP = :new.CNP;

IF v_count > 0 THEN
    RAISE_APPLICATION_ERROR(-20001, 'CNP must be unique');
END IF;
END;
```



d) **update_doctor_salary** - acest declanșator este destinat să actualizeze salariul unui medic în tabelul ANGAJATI după ce o linie în tabelul DOCTORI este actualizată. Acesta va actualiza salariul medicului în tabelul ANGAJATI în funcție de specialitatea medicului. Dacă specialitatea este 'cardiologie', atunci salariul va fi incrementat cu 1000.

```
CREATE OR REPLACE TRIGGER update_doctor_salary
    AFTER UPDATE ON DOCTORI
    FOR EACH ROW
    DECLARE
        v_salary NUMBER; -- Declare a variable to hold the salary
    BEGIN
        SELECT salariu INTO v_salary FROM angajati WHERE CNP = :new.CNP_agt;
        -- Calculate the salary based on speciality
        IF :new.specializare = 'cardiologie' THEN
            v_salary := v_salary + 1000;
        END IF;
```

-- Update the salary in the ANGAJATI table

UPDATE ANGAJATI SET salariu = v_salary WHERE CNP = :new.CNP_agt;

END;

UPDATE DOCTORI
SET specializare = 'cardiologie'
WHERE id_doctor = 10;

select * from angajati where cnp=5231687122398;

Results

ExplainDescribeSaved SQLHistory

CNP	NUME	PRENUME	DATA_N	DATA_ANG	SALARIU
5231687122398	Boian	Claudia	26-Mar-1972	22-Dec-2022	4958

UPDATE DOCTORI
SET specializare = 'cardiologie'
WHERE id_doctor = 10;

select * from angajati where cnp=5231687122398;

Results

ExplainDescribeSaved SQLHistory

CNP	NUME	PRENUME	DATA_N	DATA_ANG	SALARIU
5231687122398	Boian	Claudia	26-Mar-1972	22-Dec-2022	5958