

**ACADEMIA DE STUDII ECONOMICE BUCUREȘTI**  
**FACULTATEA DE CIBERNETICĂ, STATISTICĂ ȘI INFORMATICĂ ECONOMICĂ**

# **PROIECT BAZE DE DATE**

## **BANCĂ DE SÂNGE**

**Coordonator:**

**Niculae Andreea-Mihaela**

**Student: Pașcan Cosmin**

**Grupa: 1057**

**București**

**2023**

# CUPRINS

<b>TOPIC</b>	<b>NUMĂR PAGINĂ</b>
<b>DESCRIERE</b>	<b>3-4</b>
<b>SCHEMĂ CONCEPTUALĂ</b>	<b>5</b>
<b>TABELE ȘI RESTRICȚII</b>	<b>6-9</b>
<b>ACTUALIZARE ÎNREGISTRĂRI</b>	<b>10</b>
<b>INTEROGĂRI</b>	<b>11-17</b>
<b>COD</b>	<b>18-34</b>

# BANCĂ DE SÂNGE

## INTRODUCERE

Tema economică aleasă este gestiunea unei bănci de sânge prin intermediul unei baze de date. O bancă de sânge are scopul de a colecta și stoca sângele, precum și de a-l furniza tuturor pacienților spitalelor care au nevoie de transfuzii. Angajații băncii vor intermedia transferul de sânge dintre donatori și pacienți în felul următor: recoltorii vor prelua sângele de la donatori, sânge care va fi trimis ulterior la Departamentul de Cercetare. Aici, analiștii bacteriologi vor supune probele de sânge unor teste pentru a vedea dacă este sau nu contaminat, rezultate care vor fi trimise la analiștii sefi. La finalul fiecărei luni, după ce sângele este recoltat și analizat, acesta este introdus în baza de date. Spitalul de care aparține pacientul va contacta un manager pentru a verifica disponibilitatea grupei sanguine a pacientului, urmând realizarea transferului. Un astfel de sistem este net superior celui manual, în care datele sunt depozitate în caiete, agende, etc, deoarece salvează timp și este ușor de folosit.

## INFORMATII DESPRE BAZA DE DATE

Obiectivul proiectului este de a facilita interacțiunea pacient-donator, lucru realizat prin intermediul relațiilor dintre tabelele bazei de date create.

**FUNCTII\_BS: (Attribute: ID\_FUNCTIE, DENUMIRE\_FUNCTIE, DEPARTAMENT, SALARIU\_MIN, SALARIU\_MAX)**

Fiecărui angajat îi este atribuit un câmp numit ID\_FUNCTIE (primary key) care ne oferă informații despre departamentul în care lucrează și denumirea funcției acestuia în banca de sânge. De asemenea, avem informații despre salariul minim și maxim în care se încadrează funcția respectivă.

**ANGAJATI\_BS: (Attribute: ID\_ANGAJAT, NUME\_ANGAJAT, EMAIL\_ANGAJAT, DATA\_ANGAJARE, ID\_FUNCTIE, SALARIU, ID\_MANAGER)**

În baza de date, angajaților li se atribuie un ID\_ANGAJAT unic (primary key), precum și alte date: nume, email, data angajării. Prin intermediul atributului ID\_FUNCTIE (foreign key) se accesează detalii despre funcțiile fiecărui angajat. Totodată, avem în tabel attributele salariu și id-ul managerului (foreign key – id-ul managerului trebuie să existe în coloana ID\_ANGAJAT).

Functii – ID\_ANGAJAT clasificare: manager (100-199), recoltor sânge (200-299), analist general (300-399) și analist bacteriolog(400-499).

**SPITALE: (Attribute: ID\_SPITAL, DENUMIRE\_SPITAL, EMAIL\_SPITAL, TELEFON\_SPITAL, ID\_ANGAJAT)**

Acest tabel va stoca informații despre spitale: denumire, email, telefon. Fiecărui spital i se va repartiza un atribut unic ID\_SPITAL (primary key), precum și un manager ce se va ocupa de pacienții din spitalul respectiv prin atributul ID\_ANGAJAT (foreign key cu valori între 100 și 199).

**PACIENTI: (Attribute: ID\_PACIENT, NUME\_PACIENT, GRUPA\_SANGE\_NECESARA, DATA\_NASTERE\_PACIENT, SEX\_PACIENT, ID\_SPITAL)**

Tabelul PACIENTI va conține detalii personale ale pacientului: nume, data nașterii, sex, grupa de sânge necesară, precum și un id unic ID\_PACIENT (primary key). Pacienții sunt repartizați unui spital prin intermediul atributului ID\_SPITAL (foreign key).

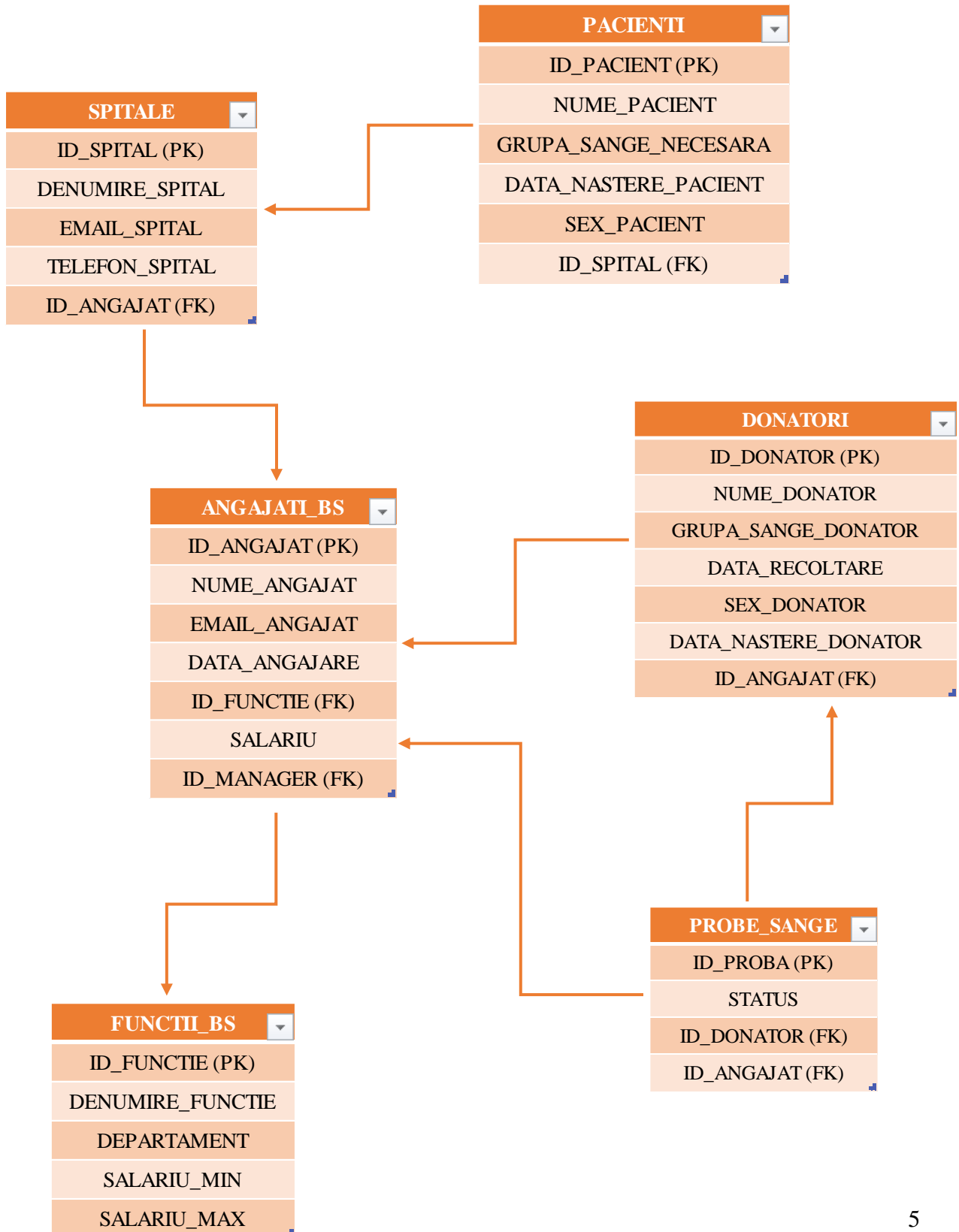
**DONATORI: (Attribute: ID\_DONATOR, NUME\_DONATOR, GRUPA\_SANGE\_DONATOR, DATA\_RECOLTARE, SEX\_DONATOR, DATA\_NASTERE\_DONATOR, ID\_ANGAJAT)**

În baza de date, donatorii au următoarele attribute: ID\_DONATOR unic (primary key), nume, grupa sanguină, data în care a avut loc recoltarea, sexul, data nașterii. De asemenea, li se atribuie un ID\_ANGAJAT (foreign key cu valori între 200-299) care corespunde unui recoltor.

**PROBE\_SANGE: (Attribute: ID\_PROBA, STATUS, ID\_DONATOR, ID\_ANGAJAT)**

Fiecărei probe de sânge i se alocă un ID\_PROBA unic (primary key), id-ul donatorului, alături de un status în urma testelor realizate de analiștii bacteriologi (ID\_ANGAJAT – foreign key cu valori între 400-499). Statusul are două valori: 0 – necontaminat, 1 – contaminat.

## SCHEMĂ CONCEPTUALĂ



# CREARE TABELE ȘI RESTRICȚII

## 1. TABEL FUNCTII\_BS

```
CREATE TABLE FUNCTII_BS
```

```
(  
    ID_FUNCTIE VARCHAR2(10) NOT NULL PRIMARY KEY,  
    DENUMIRE_FUNCTIE VARCHAR2(20) NOT NULL,  
    DEPARTAMENT VARCHAR2(30) NOT NULL,  
    SALARIU_MIN NUMBER(8, 2),  
    SALARIU_MAX NUMBER(8, 2)  
);
```

```
ALTER TABLE FUNCTII_BS
```

```
    ADD CONSTRAINT SALARIU_MIN_CK CHECK (SALARIU_MIN > 0);
```

```
ALTER TABLE FUNCTII_BS
```

```
    ADD CONSTRAINT SALARIU_MAX_CK CHECK (SALARIU_MAX > 0);
```

## 2. TABEL ANGAJATI\_BS

```
CREATE TABLE ANGAJATI_BS
```

```
(  
    ID_ANGAJAT NUMBER(6) NOT NULL PRIMARY KEY,  
    NUME_ANGAJAT VARCHAR2(50) NOT NULL,  
    EMAIL_ANGAJAT VARCHAR2(30) NOT NULL,  
    DATA_ANGAJARE DATE NOT NULL,  
    ID_FUNCTIE VARCHAR2(10) NOT NULL,  
    SALARIU NUMBER(8, 2) NOT NULL,  
    ID_MANAGER NUMBER(6)  
);
```

```
ALTER TABLE ANGAJATI_BS
```

```
    ADD CONSTRAINT ID_FUNCTIE_FK FOREIGN KEY (ID_FUNCTIE) REFERENCES FUNCTII_BS  
(ID_FUNCTIE);
```

```
ALTER TABLE ANGAJATI_BS
```

```
    ADD CONSTRAINT ID_ANGAJAT_CK CHECK
```

```
(
(ID_ANGAJAT >= 100 AND ID_ANGAJAT <= 199 AND ID_FUNCTIE LIKE 'MAN')
OR (ID_ANGAJAT >= 200 AND ID_ANGAJAT <= 299 AND ID_FUNCTIE LIKE 'REC')
OR (ID_ANGAJAT >= 300 AND ID_ANGAJAT <= 399 AND ID_FUNCTIE LIKE 'AN GEN')
OR (ID_ANGAJAT >= 400 AND ID_ANGAJAT <= 499 AND ID_FUNCTIE LIKE 'AN BACT')
);

ALTER TABLE ANGAJATI_BS
ADD CONSTRAINT SALARIU_CK CHECK
(
(SALARIU >= 6000 AND SALARIU <= 7000 AND ID_FUNCTIE LIKE 'MAN')
OR (SALARIU >= 3000 AND SALARIU <= 4000 AND ID_FUNCTIE LIKE 'REC')
OR (SALARIU >= 5000 AND SALARIU <= 6000 AND ID_FUNCTIE LIKE 'AN GEN')
OR (SALARIU >= 4000 AND SALARIU <= 5000 AND ID_FUNCTIE LIKE 'AN BACT')
);

ALTER TABLE ANGAJATI_BS
ADD CONSTRAINT ID_MANAGER_FK FOREIGN KEY (ID_MANAGER) REFERENCES ANGAJATI_BS
(ID_ANGAJAT);
```

### **3. TABEL SPITALE**

```
CREATE TABLE SPITALE
(
ID_SPITAL NUMBER(6) NOT NULL PRIMARY KEY,
DENUMIRE_SPITAL VARCHAR2(30) NOT NULL,
EMAIL_SPITAL VARCHAR2(30) NOT NULL,
TELEFON_SPITAL VARCHAR2(20) NOT NULL,
ID_ANGAJAT NUMBER(6)
);

ALTER TABLE SPITALE
ADD CONSTRAINT SP_ID_ANGAJAT_FK FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJATI_BS
(ID_ANGAJAT);

ALTER TABLE SPITALE
ADD CONSTRAINT SP_ID_ANGAJAT_CK CHECK (ID_ANGAJAT >= 100 AND ID_ANGAJAT <= 199);
```

#### **4. TABEL PACIENTI**

CREATE TABLE PACIENTI

```
(  
    ID_PACIENT NUMBER(6) NOT NULL PRIMARY KEY,  
    NUME_PACIENT VARCHAR2(50) NOT NULL,  
    GRUPA_SANGE_NECESARA VARCHAR2(10) NOT NULL,  
    DATA_NASTERE_PACIENT DATE NOT NULL,  
    SEX_PACIENT VARCHAR2(5) NOT NULL,  
    ID_SPITAL NUMBER(6) NOT NULL  
);
```

ALTER TABLE PACIENTI

ADD CONSTRAINT ID\_SPITAL\_FK FOREIGN KEY (ID\_SPITAL) REFERENCES SPITALE (ID\_SPITAL);

#### **5. TABEL DONATORI**

CREATE TABLE DONATORI

```
(  
    ID_DONATOR NUMBER(6) NOT NULL PRIMARY KEY,  
    NUME_DONATOR VARCHAR2(50) NOT NULL,  
    GRUPA_SANGE_DONATOR VARCHAR2(10) NOT NULL,  
    DATA_RECOLTARE DATE NOT NULL,  
    SEX_DONATOR VARCHAR2(5) NOT NULL,  
    DATA_NASTERE_DONATOR DATE NOT NULL,  
    ID_ANGAJAT NUMBER(6) NOT NULL  
);
```

ALTER TABLE DONATORI

ADD CONSTRAINT D\_ID\_ANGAJAT\_FK FOREIGN KEY (ID\_ANGAJAT) REFERENCES ANGAJATI\_BS (ID\_ANGAJAT);

ALTER TABLE DONATORI

ADD CONSTRAINT D\_ID\_ANGAJAT\_CK CHECK (ID\_ANGAJAT >= 200 AND ID\_ANGAJAT <= 299);

#### **6. TABEL PROBE\_SANGE**

CREATE TABLE PROBE\_SANGE

```
(  
    ID_PROBA NUMBER(6) NOT NULL PRIMARY KEY,
```



```
STATUS NUMBER(2),
ID_DONATOR NUMBER(6) NOT NULL,
ID_ANGAJAT NUMBER(6) NOT NULL
);
ALTER TABLE PROBE_SANGE
  ADD CONSTRAINT PS_ID_DONATOR_FK FOREIGN KEY (ID_DONATOR) REFERENCES DONATORI
(ID_DONATOR);
ALTER TABLE PROBE_SANGE
  ADD CONSTRAINT PS_ID_ANGAJAT_FK FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJATI_BS
(ID_ANGAJAT);
ALTER TABLE PROBE_SANGE
  ADD CONSTRAINT PS_ID_ANGAJAT_CK CHECK (ID_ANGAJAT >= 400 AND ID_ANGAJAT <= 499);
ALTER TABLE PROBE_SANGE
  ADD CONSTRAINT STATUS_CK CHECK (STATUS = 1 OR STATUS = 0);
```

## ACTUALIZARE ÎNREGISTRĂRI

UPDATE DONATORI

SET GRUPA\_SANGE\_DONATOR = 'B(III)-'

WHERE ID\_DONATOR = 15006;

UPDATE SPITALE

SET TELEFON\_SPITAL = '0748.558.397'

WHERE DENUMIRE\_SPITAL LIKE 'VICTOR BABES';

UPDATE ANGAJATI\_BS

SET ID\_MANAGER = 302

WHERE ID\_ANGAJAT = 402;

UPDATE PACIENTI

SET ID\_SPITAL = 1

WHERE NUME\_PACIENT LIKE '%LEC%';

UPDATE PROBE\_SANGE

SET STATUS = 0

WHERE ID\_PROBA = 111;

# INTEROGĂRI

1. Să se afișeze numele, emailul angajatului din departamentul administrare cu cel mai mic salariu.

```
SELECT ANG.NUME_ANGAJAT, ANG.EMAIL_ANGAJAT
FROM ANGAJATI_BS ANG, FUNCTII_BS F
WHERE ANG.ID_FUNCTIE = F.ID_FUNCTIE
AND F.DEPARTAMENT LIKE 'ADMINISTRARE'
AND ANG.SALARIU IN (SELECT MIN(ANG.SALARIU)
                     FROM ANGAJATI_BS ANG, FUNCTII_BS F
                     WHERE ANG.ID_FUNCTIE = F.ID_FUNCTIE
                     AND F.DEPARTAMENT LIKE 'ADMINISTRARE');
```

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0.016 seconds

NUME_ANGAJAT	EMAIL_ANGAJAT
1 PASCAN COSMIN PASCAN	

2. Să se afișeze numele donatorilor din ziua de 15 decembrie și numele recoltorilor lor.

```
SELECT D.NUME_DONATOR, BS.NUME_ANGAJAT
FROM DONATORI D, ANGAJATI_BS BS
WHERE D.ID_ANGAJAT = BS.ID_ANGAJAT
AND D.DATA_RECOLTARE = TO_DATE('15-12-2022', 'DD-MM-YYYY');
```

Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0.063 seconds

NUME_DONATOR	NUME_ANGAJAT
1 GEORGESCU IONELA	IFRIM MARIUS
2 CORINA CHITU	IFRIM MARIUS
3 MARIN OLGA	ANTON MATEI

3. Să se afișeze numărul probelor contaminate din intervalul 15-20 decembrie.

```
SELECT COUNT(PS.ID_PROBA) AS NR_PROBE_CONTAMINATE
FROM PROBE_SANGE PS, DONATORI D
WHERE PS.ID_DONATOR = D.ID_DONATOR
AND PS.STATUS = 1
AND EXTRACT(DAY FROM D.DATA_RECOLTARE) BETWEEN 15 AND 20;
```

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0.006 seconds

NR_PROBE_CONTAMINATE
1 3

4. Să se afișeze numele, departamentul în care lucrează și nivelul ierarhic pentru toți subordonații angajatului cu id 101 care nu au alți subordonați.

```

SELECT ANG.NUME_ANGAJAT, F.DEPARTAMENT, LEVEL
FROM ANGAJATI_BS ANG, FUNCTII_BS F
WHERE ANG.ID_FUNCTIE = F.ID_FUNCTIE
AND CONNECT_BY_ISLEAF = 1
CONNECT BY PRIOR ANG.ID_ANGAJAT = ANG.ID_MANAGER
START WITH ANG.ID_ANGAJAT = 101;

```

	NUME_ANGAJAT	DEPARTAMENT	LEVEL
1	ANTON MATEI	RECOLTARE	2
2	GEORGESCU MIHAI	RECOLTARE	2
3	SAVA BOGDAN	CERCETARE	3
4	DAVID DANIEL	CERCETARE	2

5. Să se afișeze denumirea spitalului și numărul pacienților asociați.

```

SELECT SP.DENUMIRE_SPITAL, COUNT(P.ID_SPITAL) AS NUMAR_PACIENTI
FROM SPITALE SP, PACIENTI P
WHERE P.ID_SPITAL = SP.ID_SPITAL
GROUP BY SP.DENUMIRE_SPITAL;

```

	DENUMIRE_SPITAL	NUMAR_PACIENTI
1	FLOREASCA	2
2	GEROTA	3
3	SFANTA MARIA	3
4	SFANTUL PANTELIMON	2
5	COLENTINA	4

6. Să se afișeze numele, grupa sanguină și data nașterii pacienților născuți după anul 2000.

```

SELECT NUME_DONATOR, GRUPA_SANGE_DONATOR, DATA_NASTERE_DONATOR
FROM DONATORI
WHERE EXTRACT(YEAR FROM DATA_NASTERE_DONATOR) >= 2000;

```

	NUME_DONATOR	GRUPA_SANGE_DONATOR	DATA_NASTERE_DONATOR
1	DINU FLORIN	A (II) +	26-NOV-02
2	CHIRILA ALEX	B (III) +	09-JUN-00

7. Să se afișeze numele, salariul inițial și salariul angajaților după ce este impozitat.

Impozitul se aplică astfel: 3000-4499 (10%), 4500-5999 (12%), 6000+ (14%).

```
SELECT NUME_ANGAJAT, SALARIU,
CASE
    WHEN SALARIU BETWEEN 3000 AND 4499 THEN 0.9*SALARIU
    WHEN SALARIU BETWEEN 4500 AND 5999 THEN 0.88*SALARIU
    WHEN SALARIU >= 6000 THEN 0.86*SALARIU
    ELSE 0
END AS SALARIU_IMPOZITAT
FROM ANGAJATI_BS;
```

Script Output x Query Result x

SQL | All Rows Fetched: 27 in 0.016 seconds

	NUME_ANGAJAT	SALARIU	SALARIU_IMPOZITAT
1	PASCAN COSMIN	6000	5160
2	STOICA LUCIAN	7000	6020
3	STAN EMILIAN	6500	5590
4	PAVEL OLIVIA	6800	5848
5	DUMITRU SABINA	7000	6020
6	IFRIM MARIUS	3300	2970
7	ANTON MATEI	3400	3060

8. Să se afișeze numele donatorilor a căror probă de sânge a fost analizată de analistul bacteriolog Dinu Eugen.

```
SELECT D.NUME_DONATOR
FROM ANGAJATI_BS ANG, PROBE_SANGE PS, DONATORI D
WHERE D.ID_DONATOR = PS.ID_DONATOR
AND PS.ID_ANGAJAT = ANG.ID_ANGAJAT
AND ANG.NUME_ANGAJAT LIKE 'DINU EUGEN';
```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.016 seconds

	NUME_DONATOR
1	GEORGESCU IONELA
2	CORINA CHITU
3	MARIN OLGA
4	TANASE ELENA
5	VIRGIL IONUT

9. Să se afișeze numele, emailul angajaților al căror nume începe cu litera 'D' din departamentul cercetare.

```
SELECT ANG.NUME_ANGAJAT, ANG.EMAIL_ANGAJAT
FROM ANGAJATI_BS ANG, FUNCTII_BS F
WHERE ANG.ID_FUNCTIE = F.ID_FUNCTIE
AND DEPARTAMENT LIKE 'CERCETARE'
AND ANG.NUME_ANGAJAT LIKE 'D%';
```

Script Output x Query Result x

SQL | All Rows Fetched: 4 in 0.013 seconds

	NUME_ANGAJAT	EMAIL_ANGAJAT
1	DRAGOMIR IULIAN	DRAGOMIRI
2	DAVID DANIEL	DAVIDD
3	DINU EUGEN	DINUE
4	DOBRE VICTOR	DOBRER

10. Să se afișeze media salariului angajaților cu vechime sub 10 ani.

```
SELECT ROUND (AVG (SALARIU), 2) AS SALARIU_MEDIU
FROM ANGAJATI_BS
WHERE EXTRACT (YEAR FROM SYSDATE) - 10 < EXTRACT (YEAR FROM DATA_ANGAJARE);
```

Script Output	Query Result
All Rows Fetched: 1 in 0.009 seconds	
SALARIU_MEDIU	
1 4566.67	

11. Să se afișeze numele recoltorului, numărul de donatori al căror nume începe cu 'G'.

```
SELECT ANG.NUME_ANGAJAT, COUNT (D.NUME_DONATOR) AS NR_DONATORI
FROM ANGAJATI_BS ANG, DONATORI D
WHERE ANG.ID_ANGAJAT = D.ID_ANGAJAT
GROUP BY ANG.NUME_ANGAJAT
HAVING ANG.NUME_ANGAJAT LIKE ('G%');
```

Script Output	Query Result
All Rows Fetched: 2 in 0.02 seconds	
NUME_ANGAJAT	NR_DONATORI
1 GEORGESCU MIHAI	2
2 GEAMANU FLORIN	1

12. Să se afișeze numele, salariul angajaților din departamentul recoltare cu vechime peste 10 ani.

```
SELECT ANG.NUME_ANGAJAT, ANG.SALARIU
FROM ANGAJATI_BS ANG, FUNCTII_BS F
WHERE ANG.ID_FUNCTIE = F.ID_FUNCTIE
AND F.DEPARTAMENT LIKE 'RECOLTARE'
AND EXTRACT (YEAR FROM SYSDATE) - 12 > EXTRACT (YEAR FROM DATA_ANGAJARE);
```

Script Output	Query Result
All Rows Fetched: 4 in 0.006 seconds	
NUME_ANGAJAT	SALARIU
1 ANTON MATEI	3400
2 RADU MONICA	3800
3 OLTEANU CARINA	3900
4 GEAMANU FLORIN	3300

13. Să se afișeze denumirea spitalului, grupa și numărul pacienților cu grupa necesara 0(II)- pentru spitatul Sfanta Maria.

```
SELECT SP.DENUMIRE_SPITAL, P.GRUPA_SANGE_NECESARA, COUNT(P.ID_PACIENT) AS NR_PACIENTI
FROM SPITALE SP, PACIENTI P
WHERE SP.ID_SPITAL = P.ID_SPITAL
AND P.GRUPA_SANGE_NECESARA LIKE '0(I)-'
GROUP BY SP.DENUMIRE_SPITAL, P.GRUPA_SANGE_NECESARA
HAVING SP.DENUMIRE_SPITAL LIKE 'SFANTA MARIA';
```

DENUMIRE_SPITAL	GRUPA_SANGE_NECESARA	NR_PACIENTI
1 SFANTA MARIA	0 (I) -	2

14. Să se afișeze numele donatorului, id-ul probei pentru probele de sânge necontaminate.

```
SELECT D.NUME_DONATOR, PS.ID_PROBA
FROM DONATORI D, PROBE_SANGE PS
WHERE D.ID_DONATOR = PS.ID_DONATOR
AND PS.STATUS = 0;
```

NUME_DONATOR	ID_PROBA
1 GEORGESCU IONELA	100
2 CORINA CHITU	101
3 TANASE ELENA	103
4 NEGRESU EUGEN	106
5 RADUCIOIU OCTAVIAN	107
6 CHIRILA ALEX	110

15. Să se afișeze numele, grupa și sexul pentru donatorii născuți între 1960-1980.

```
SELECT NUME_DONATOR, GRUPA_SANGE_DONATOR, SEX_DONATOR
FROM DONATORI
WHERE EXTRACT(YEAR FROM DATA_NASTERE_DONATOR) BETWEEN 1960 AND 1980;
```

NUME_DONATOR	GRUPA_SANGE_DONATOR	SEX_DONATOR
1 GEORGESCU IONELA	0 (I) +	F
2 CORINA CHITU	AB (IV) +	F
3 TANASE ELENA	B (III) -	F
4 NEAGOE NATALIA	A (II) -	F
5 MOISUC CIPRIAN	AB (IV) -	M

16. Să se afișeze numele și numărul pacienților de sex feminin ai spitalului Gerota.

```

SELECT SP.DENUMIRE_SPITAL, COUNT(P.SEX_PACIENT) AS NR_PACIENTI
FROM SPITALE SP, PACIENTI P
WHERE SP.ID_SPITAL = P.ID_SPITAL
AND P.SEX_PACIENT LIKE 'F'
GROUP BY SP.DENUMIRE_SPITAL
HAVING SP.DENUMIRE_SPITAL LIKE 'GEROTA';

```

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0.015 seconds

DENUMIRE_SPITAL	NR_PACIENTI
1 GEROTA	2

17. Să se afișeze numele managerului, spitalul și numărul de pacienți ajutați.

```

SELECT ANG.NUME_ANGAJAT, COUNT(P.ID_PACIENT) AS NR_PACIENTI
FROM ANGAJATI_BS ANG, SPITALE SP, PACIENTI P
WHERE SP.ID_ANGAJAT = ANG.ID_ANGAJAT
AND SP.ID_SPITAL = P.ID_SPITAL
GROUP BY ANG.NUME_ANGAJAT;

```

Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0.225 seconds

NUME_ANGAJAT	NR_PACIENTI
1 STOICA LUCIAN	6
2 PASCAN COSMIN	5
3 STAN EMILIAN	3

18. Să se afișeze numele recoltorului, numărul persoanelor de sex masculin care au donat sânge în luna decembrie.

```

SELECT ANG.NUME_ANGAJAT, COUNT(D.ID_DONATOR) AS NR_DONATORI
FROM ANGAJATI_BS ANG, DONATORI D
WHERE ANG.ID_ANGAJAT = D.ID_ANGAJAT
AND D.SEX_DONATOR LIKE 'M'
AND EXTRACT (MONTH FROM D.DATA_RECOLTARE) = 12
GROUP BY ANG.NUME_ANGAJAT;

```

Script Output x Query Result x

SQL | All Rows Fetched: 4 in 0.01 seconds

NUME_ANGAJAT	NR_DONATORI
1 OLTEANU CARINA	2
2 GEAMANU FLORIN	1
3 MANOLACHE LAURA	2
4 RADU MONICA	2



19. Să se afișeze numele, data recoltării și grupa persoanelor cu grupa diferită de 0(I).

```
SELECT NUME_DONATOR, DATA_RECOLTARE, GRUPA_SANGE_DONATOR
FROM DONATORI
WHERE GRUPA_SANGE_DONATOR <> '0(I) +'
AND GRUPA_SANGE_DONATOR <> '0(I) -';
```

Script Output x Query Result x

SQL | All Rows Fetched: 10 in 0.307 seconds

	NUME_DONATOR	DATA_RECOLTARE	GRUPA_SANGE_DONATOR
1	CORINA CHITU	15-DEC-22	AB (IV) +
2	TANASE ELENA	16-DEC-22	B (III) -
3	NEAGOE NATALIA	19-DEC-22	A (II) -
4	DOBRE DIANA	20-DEC-22	A (II) +
5	NEGRESCU EUGEN	20-DEC-22	B (III) -
6	DINU FLORIN	22-DEC-22	A (II) +

20. Să se afișeze numele, departamentul și nivelul ierarhic pentru angajații care au subordonați și au aceeași funcție ca Pavel Olivia.

```
SELECT ANG.NUME_ANGAJAT, F.DEPARTAMENT, LEVEL
FROM ANGAJATI_BS ANG, FUNCTII_BS F
WHERE ANG.ID_FUNCIE = F.ID_FUNCIE
AND CONNECT_BY_ISLEAF = 0
CONNECT BY PRIOR ANG.ID_ANGAJAT = ANG.ID_MANAGER
START WITH ANG.ID_FUNCIE IN (SELECT ID_FUNCIE FROM ANGAJATI_BS WHERE NUME_ANGAJAT = 'PAVEL OLIVIA');
```

Script Output x Query Result x

SQL | All Rows Fetched: 11 in 0.004 seconds

	NUME_ANGAJAT	DEPARTAMENT	LEVEL
1	PASCAN COSMIN	ADMINISTRARE	1
2	POPA MIREL	CERCETARE	2
3	FLORESCU DARIA	CERCETARE	2
4	STOICA LUCIAN	ADMINISTRARE	1
5	MARIN IULIAN	CERCETARE	2
6	STAN EMILIAN	ADMINISTRARE	1

## COD

```
DROP TABLE FUNCTII_BS CASCADE CONSTRAINTS;
DROP TABLE ANGAJATI_BS CASCADE CONSTRAINTS;
DROP TABLE SPITALE CASCADE CONSTRAINTS;
DROP TABLE PACIENTI CASCADE CONSTRAINTS;
DROP TABLE DONATORI CASCADE CONSTRAINTS;
DROP TABLE PROBE_SANGE CASCADE CONSTRAINTS;

--TABEL FUNCTII_BS
CREATE TABLE FUNCTII_BS
(
    ID_FUNCTIE VARCHAR2(10) NOT NULL PRIMARY KEY,
    DENUMIRE_FUNCTIE VARCHAR2(20) NOT NULL,
    DEPARTAMENT VARCHAR2(30) NOT NULL,
    SALARIU_MIN NUMBER(8, 2),
    SALARIU_MAX NUMBER(8, 2)
);
--CONSTRAINTS TABEL FUNCTII_BS
ALTER TABLE FUNCTII_BS
    ADD CONSTRAINT SALARIU_MIN_CK CHECK (SALARIU_MIN > 0);
ALTER TABLE FUNCTII_BS
    ADD CONSTRAINT SALARIU_MAX_CK CHECK (SALARIU_MAX > 0);
--VALORI TABEL FUNCTII_BS
INSERT INTO FUNCTII_BS (ID_FUNCTIE, DENUMIRE_FUNCTIE, DEPARTAMENT, SALARIU_MIN,
SALARIU_MAX)
VALUES('MAN', 'MANAGER', 'ADMINISTRARE', 6000, 7000);
INSERT INTO FUNCTII_BS (ID_FUNCTIE, DENUMIRE_FUNCTIE, DEPARTAMENT, SALARIU_MIN,
SALARIU_MAX)
VALUES('REC', 'RECOLTOR SANGE', 'RECOLTARE', 3000, 4000);
INSERT INTO FUNCTII_BS (ID_FUNCTIE, DENUMIRE_FUNCTIE, DEPARTAMENT, SALARIU_MIN,
SALARIU_MAX)
```

```

VALUES('AN GEN', 'ANALIST GENERAL', 'CERCETARE', 5000, 6000);

INSERT INTO FUNCTII_BS (ID_FUNCTIE, DENUMIRE_FUNCTIE, DEPARTAMENT, SALARIU_MIN,
SALARIU_MAX)

VALUES('AN BACT', 'ANALIST BACTERIOLOG', 'CERCETARE', 4000, 5000);

--TABEL ANGAJATI_BS
CREATE TABLE ANGAJATI_BS
(
    ID_ANGAJAT NUMBER(6) NOT NULL PRIMARY KEY,
    NUME_ANGAJAT VARCHAR2(50) NOT NULL,
    EMAIL_ANGAJAT VARCHAR2(30) NOT NULL,
    DATA_ANGAJARE DATE NOT NULL,
    ID_FUNCTIE VARCHAR2(10) NOT NULL,
    SALARIU NUMBER(8, 2) NOT NULL,
    ID_MANAGER NUMBER(6)
);

--CONSTRAINTS TABEL ANGAJATI_BS
ALTER TABLE ANGAJATI_BS

    ADD CONSTRAINT ID_FUNCTIE_FK FOREIGN KEY (ID_FUNCTIE) REFERENCES FUNCTII_BS
(ID_FUNCTIE);

ALTER TABLE ANGAJATI_BS

    ADD CONSTRAINT ID_ANGAJAT_CK CHECK
(
    (ID_ANGAJAT >= 100 AND ID_ANGAJAT <= 199 AND ID_FUNCTIE LIKE 'MAN')
    OR (ID_ANGAJAT >= 200 AND ID_ANGAJAT <= 299 AND ID_FUNCTIE LIKE 'REC')
    OR (ID_ANGAJAT >= 300 AND ID_ANGAJAT <= 399 AND ID_FUNCTIE LIKE 'AN GEN')
    OR (ID_ANGAJAT >= 400 AND ID_ANGAJAT <= 499 AND ID_FUNCTIE LIKE 'AN BACT')
);

ALTER TABLE ANGAJATI_BS

    ADD CONSTRAINT SALARIU_CK CHECK
(
    (SALARIU >= 6000 AND SALARIU <= 7000 AND ID_FUNCTIE LIKE 'MAN')
    OR (SALARIU >= 3000 AND SALARIU <= 4000 AND ID_FUNCTIE LIKE 'REC')

```

```

OR (SALARIU >= 5000 AND SALARIU <= 6000 AND ID_FUNCTIE LIKE 'AN GEN')
OR (SALARIU >= 4000 AND SALARIU <= 5000 AND ID_FUNCTIE LIKE 'AN BACT')
);
ALTER TABLE ANGAJATI_BS
ADD CONSTRAINT ID_MANAGER_FK FOREIGN KEY (ID_MANAGER) REFERENCES ANGAJATI_BS
(ID_ANGAJAT);
--VALORI TABEL ANGAJATI_BS
INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)
VALUES (100, 'PASCAN COSMIN', 'PASCANC', TO_DATE('15-08-2015', 'DD-MM-YYYY'), 'MAN', 6000,
NULL);
INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)
VALUES (101, 'STOICA LUCIAN', 'STOICAL', TO_DATE('29-09-2004', 'DD-MM-YYYY'), 'MAN', 7000, NULL);
INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)
VALUES (102, 'STAN EMILIAN', 'STANE', TO_DATE('30-03-2007', 'DD-MM-YYYY'), 'MAN', 6500, NULL);
INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)
VALUES (103, 'PAVEL OLIVIA', 'PAVELO', TO_DATE('14-07-2009', 'DD-MM-YYYY'), 'MAN', 6800, NULL);
INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)
VALUES (104, 'DUMITRU SABINA', 'DUMITRUS', TO_DATE('05-01-2011', 'DD-MM-YYYY'), 'MAN', 7000,
NULL);
INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)
VALUES (200, 'IFRIM MARIUS', 'IFRIMM', TO_DATE('19-06-2013', 'DD-MM-YYYY'), 'REC', 3300, 100);
INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)
VALUES (201, 'ANTON MATEI', 'ANTONM', TO_DATE('22-07-2008', 'DD-MM-YYYY'), 'REC', 3400, 101);
INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)
VALUES (202, 'GEORGESCU MIHAI', 'GEORGESCU', TO_DATE('08-09-2012', 'DD-MM-YYYY'), 'REC',
3200, 101);
INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)

```

```

VALUES (203, 'MANOLACHE LAURA', 'MANOLACHEL', TO_DATE('13-05-2014', 'DD-MM-YYYY'), 'REC',
3500, 100);

INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)

VALUES (204, 'RADU MONICA', 'RADUM', TO_DATE('19-07-2009', 'DD-MM-YYYY'), 'REC', 3800, 102);

INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)

VALUES (205, 'OLTEANU CARINA', 'OLTEANUC', TO_DATE('26-01-2009', 'DD-MM-YYYY'), 'REC', 3900,
103);

INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)

VALUES (206, 'GEAMANU FLORIN', 'GEAMANUF', TO_DATE('07-02-2010', 'DD-MM-YYYY'), 'REC', 3300,
100);

INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)

VALUES (300, 'POPA MIREL', 'POPAM', TO_DATE('31-01-2006', 'DD-MM-YYYY'), 'AN GEN', 5600, 100);

INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)

VALUES (301, 'MARIN IULIAN', 'MARINI', TO_DATE('18-08-2015', 'DD-MM-YYYY'), 'AN GEN', 5200, 101);

INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)

VALUES (302, 'DRAGOMIR IULIAN', 'DRAGOMIRI', TO_DATE('10-10-2007', 'DD-MM-YYYY'), 'AN GEN',
6000, 102);

INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)

VALUES (303, 'FLORESCU DARIA', 'FLORESCUD', TO_DATE('12-03-2011', 'DD-MM-YYYY'), 'AN GEN',
5800, 100);

INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)

VALUES (304, 'TOMULESCU ANA', 'TOMULESCUA', TO_DATE('04-05-2008', 'DD-MM-YYYY'), 'AN GEN',
5300, 103);

INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)

VALUES (305, 'SABOIU ELVIRA', 'SABOIUE', TO_DATE('29-11-2009', 'DD-MM-YYYY'), 'AN GEN', 5500,
104);

INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)

VALUES (306, 'DAVID DANIEL', 'DAVIDD', TO_DATE('11-09-2012', 'DD-MM-YYYY'), 'AN GEN', 5400, 101);

```

```

INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)

VALUES (400, 'DINU EUGEN', 'DINUE', TO_DATE('02-06-2006', 'DD-MM-YYYY'), 'AN BACT', 4500, 300);

INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)

VALUES (401, 'SAVA BOGDAN', 'SAVAB', TO_DATE('17-10-2010', 'DD-MM-YYYY'), 'AN BACT', 4700, 301);

INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)

VALUES (402, 'DOBRE VICTOR', 'DOBREV', TO_DATE('20-12-2014', 'DD-MM-YYYY'), 'AN BACT', 4200,
301);

INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)

VALUES (403, 'IONESCU ALBERT', 'IONESCUA', TO_DATE('31-03-2013', 'DD-MM-YYYY'), 'AN BACT',
4400, 302);

INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)

VALUES (404, 'PANDARU MIRELA', 'PANDARUM', TO_DATE('11-07-2005', 'DD-MM-YYYY'), 'AN BACT',
4700, 303);

INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)

VALUES (405, 'CRACIUN SMARANDA', 'CRACIUNS', TO_DATE('02-06-2009', 'DD-MM-YYYY'), 'AN BACT',
4800, 302);

INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)

VALUES (406, 'GABUREANU ADELINA', 'GABUREANUA', TO_DATE('14-11-2018', 'DD-MM-YYYY'), 'AN
BACT', 4300, 304);

INSERT INTO ANGAJATI_BS (ID_ANGAJAT, NUME_ANGAJAT, EMAIL_ANGAJAT, DATA_ANGAJARE,
ID_FUNCTIE, SALARIU, ID_MANAGER)

VALUES (407, 'VLASCEANU DOINA', 'VLASCEANUD', TO_DATE('29-09-2014', 'DD-MM-YYYY'), 'AN
BACT', 4200, 305);

```

--TABEL SPITALE

CREATE TABLE SPITALE

```

(
    ID_SPITAL NUMBER(6) NOT NULL PRIMARY KEY,
    DENUMIRE_SPITAL VARCHAR2(30) NOT NULL,
    EMAIL_SPITAL VARCHAR2(30) NOT NULL,

```

```

TELEFON_SPITAL VARCHAR2(20) NOT NULL,

ID_ANGAJAT NUMBER(6)

);

--CONSTRAINTS TABEL SPITALE

ALTER TABLE SPITALE

    ADD CONSTRAINT SP_ID_ANGAJAT_FK FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJATI_BS
(ID_ANGAJAT);

ALTER TABLE SPITALE

    ADD CONSTRAINT SP_ID_ANGAJAT_CK CHECK (ID_ANGAJAT >= 100 AND ID_ANGAJAT <= 199);

--VALORI TABEL SPITALE

INSERT INTO SPITALE (ID_SPITAL, DENUMIRE_SPITAL, EMAIL_SPITAL, TELEFON_SPITAL,
ID_ANGAJAT)

VALUES (1, 'FLOREASCA', 'SPFLOREASCA', '0728.040.761', 100);

INSERT INTO SPITALE (ID_SPITAL, DENUMIRE_SPITAL, EMAIL_SPITAL, TELEFON_SPITAL,
ID_ANGAJAT)

VALUES (2, 'SFANTA MARIA', 'SPMARIA', '0741.114.554', 100);

INSERT INTO SPITALE (ID_SPITAL, DENUMIRE_SPITAL, EMAIL_SPITAL, TELEFON_SPITAL,
ID_ANGAJAT)

VALUES (3, 'COLENTINA', 'SPCOLENTINA', '0743.886.975', 101);

INSERT INTO SPITALE (ID_SPITAL, DENUMIRE_SPITAL, EMAIL_SPITAL, TELEFON_SPITAL,
ID_ANGAJAT)

VALUES (4, 'SFANTUL PANTELIMON', 'SPPANTELIMON', '0766.716.263', 101);

INSERT INTO SPITALE (ID_SPITAL, DENUMIRE_SPITAL, EMAIL_SPITAL, TELEFON_SPITAL,
ID_ANGAJAT)

VALUES (5, 'GEROTA', 'SPGEROTA', '0723.342.412', 102);

INSERT INTO SPITALE (ID_SPITAL, DENUMIRE_SPITAL, EMAIL_SPITAL, TELEFON_SPITAL,
ID_ANGAJAT)

VALUES (6, 'ELIAS', 'SPELIAS', '0724.595.995', 102);

INSERT INTO SPITALE (ID_SPITAL, DENUMIRE_SPITAL, EMAIL_SPITAL, TELEFON_SPITAL,
ID_ANGAJAT)

VALUES (7, 'SFANTUL IOAN', 'SPIOAN', '0740.205.769', 103);

INSERT INTO SPITALE (ID_SPITAL, DENUMIRE_SPITAL, EMAIL_SPITAL, TELEFON_SPITAL,
ID_ANGAJAT)

VALUES (8, 'GOMOIU', 'SPGOMOIU', '0723.181.062', 103);

INSERT INTO SPITALE (ID_SPITAL, DENUMIRE_SPITAL, EMAIL_SPITAL, TELEFON_SPITAL,
ID_ANGAJAT)

```

```

VALUES (9, 'VICTOR BABES', 'SPBABES', '0744.507.778', 104);

INSERT INTO SPITALE (ID_SPITAL, DENUMIRE_SPITAL, EMAIL_SPITAL, TELEFON_SPITAL,
ID_ANGAJAT)

VALUES (10, 'MARIE S. CURIE', 'SPMSCURIE', '0723.553.915', 104);

--TABEL PACIENTI

CREATE TABLE PACIENTI

(
    ID_PACIENT NUMBER(6) NOT NULL PRIMARY KEY,
    NUME_PACIENT VARCHAR2(50) NOT NULL,
    GRUPA_SANGE_NECESARA VARCHAR2(10) NOT NULL,
    DATA_NASTERE_PACIENT DATE NOT NULL,
    SEX_PACIENT VARCHAR2(5) NOT NULL,
    ID_SPITAL NUMBER(6) NOT NULL
);

--CONSTRAINTS TABEL PACIENTI

ALTER TABLE PACIENTI

    ADD CONSTRAINT ID_SPITAL_FK FOREIGN KEY (ID_SPITAL) REFERENCES SPITALE (ID_SPITAL);

--VALORI TABEL PACIENTI

INSERT INTO PACIENTI (ID_PACIENT, NUME_PACIENT, GRUPA_SANGE_NECESARA,
DATA_NASTERE_PACIENT, SEX_PACIENT, ID_SPITAL)

VALUES (1000, 'DRAGAN ALEXANDRU', 'A(II)+', TO_DATE('30-05-2002', 'DD-MM-YYYY'), 'M', 1);

INSERT INTO PACIENTI (ID_PACIENT, NUME_PACIENT, GRUPA_SANGE_NECESARA,
DATA_NASTERE_PACIENT, SEX_PACIENT, ID_SPITAL)

VALUES (1001, 'ALECU CRISTIAN', 'B(III)+', TO_DATE('21-07-1999', 'DD-MM-YYYY'), 'M', 2);

INSERT INTO PACIENTI (ID_PACIENT, NUME_PACIENT, GRUPA_SANGE_NECESARA,
DATA_NASTERE_PACIENT, SEX_PACIENT, ID_SPITAL)

VALUES (1002, 'POPA CATALIN', '0(I)-', TO_DATE('15-11-1989', 'DD-MM-YYYY'), 'M', 2);

INSERT INTO PACIENTI (ID_PACIENT, NUME_PACIENT, GRUPA_SANGE_NECESARA,
DATA_NASTERE_PACIENT, SEX_PACIENT, ID_SPITAL)

VALUES (1003, 'DIACONU SERGIU', 'A(II)-', TO_DATE('01-04-2000', 'DD-MM-YYYY'), 'M', 2);

INSERT INTO PACIENTI (ID_PACIENT, NUME_PACIENT, GRUPA_SANGE_NECESARA,
DATA_NASTERE_PACIENT, SEX_PACIENT, ID_SPITAL)

VALUES (1004, 'STANESCU COSMIN', 'AB(IV)+', TO_DATE('06-02-1978', 'DD-MM-YYYY'), 'M', 3);

```



```

INSERT INTO PACIENTI (ID_PACIENT, NUME_PACIENT, GRUPA_SANGE_NECESARA,
DATA_NASTERE_PACIENT, SEX_PACIENT, ID_SPITAL)

VALUES (1005, 'GHITA ALEXANDRU', '0(I)+', TO_DATE('24-08-1986', 'DD-MM-YYYY'), 'M', 3);

INSERT INTO PACIENTI (ID_PACIENT, NUME_PACIENT, GRUPA_SANGE_NECESARA,
DATA_NASTERE_PACIENT, SEX_PACIENT, ID_SPITAL)

VALUES (1006, 'MATASARU CLARA', '0(I)-', TO_DATE('21-04-2001', 'DD-MM-YYYY'), 'F', 3);

INSERT INTO PACIENTI (ID_PACIENT, NUME_PACIENT, GRUPA_SANGE_NECESARA,
DATA_NASTERE_PACIENT, SEX_PACIENT, ID_SPITAL)

VALUES (1007, 'BANICA MIHAELA', 'AB(IV)-', TO_DATE('19-12-1994', 'DD-MM-YYYY'), 'F', 3);

INSERT INTO PACIENTI (ID_PACIENT, NUME_PACIENT, GRUPA_SANGE_NECESARA,
DATA_NASTERE_PACIENT, SEX_PACIENT, ID_SPITAL)

VALUES (1008, 'CORNEA RAMONA', 'B(III)-', TO_DATE('10-09-1999', 'DD-MM-YYYY'), 'F', 4);

INSERT INTO PACIENTI (ID_PACIENT, NUME_PACIENT, GRUPA_SANGE_NECESARA,
DATA_NASTERE_PACIENT, SEX_PACIENT, ID_SPITAL)

VALUES (1009, 'ANDREESCU FLORINA', 'A(II)+', TO_DATE('07-05-1978', 'DD-MM-YYYY'), 'F', 4);

INSERT INTO PACIENTI (ID_PACIENT, NUME_PACIENT, GRUPA_SANGE_NECESARA,
DATA_NASTERE_PACIENT, SEX_PACIENT, ID_SPITAL)

VALUES (1010, 'ZAMFIR LAURA', 'AB(IV)+', TO_DATE('16-08-1972', 'DD-MM-YYYY'), 'F', 5);

INSERT INTO PACIENTI (ID_PACIENT, NUME_PACIENT, GRUPA_SANGE_NECESARA,
DATA_NASTERE_PACIENT, SEX_PACIENT, ID_SPITAL)

VALUES (1011, 'GEORGESCU IONELA', '0(I)+', TO_DATE('19-01-1969', 'DD-MM-YYYY'), 'F', 5);

INSERT INTO PACIENTI (ID_PACIENT, NUME_PACIENT, GRUPA_SANGE_NECESARA,
DATA_NASTERE_PACIENT, SEX_PACIENT, ID_SPITAL)

VALUES (1012, 'MURGU GABRIEL', '0(I)-', TO_DATE('24-06-1977', 'DD-MM-YYYY'), 'M', 2);

INSERT INTO PACIENTI (ID_PACIENT, NUME_PACIENT, GRUPA_SANGE_NECESARA,
DATA_NASTERE_PACIENT, SEX_PACIENT, ID_SPITAL)

VALUES (1013, 'POPESCU ROBERT', 'B(III)+', TO_DATE('11-01-1999', 'DD-MM-YYYY'), 'M', 5);

```

--TABEL DONATORI

CREATE TABLE DONATORI

```

(
    ID_DONATOR NUMBER(6) NOT NULL PRIMARY KEY,
    NUME_DONATOR VARCHAR2(50) NOT NULL,
    GRUPA_SANGE_DONATOR VARCHAR2(10) NOT NULL,
    DATA_RECOLTARE DATE NOT NULL,

```

```

SEX_DONATOR VARCHAR2(5) NOT NULL,

DATA_NASTERE_DONATOR DATE NOT NULL,

ID_ANGAJAT NUMBER(6) NOT NULL

);

--CONSTRAINTS TABEL DONATORI

ALTER TABLE DONATORI

    ADD CONSTRAINT D_ID_ANGAJAT_FK FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJATI_BS
(ID_ANGAJAT);

ALTER TABLE DONATORI

    ADD CONSTRAINT D_ID_ANGAJAT_CK CHECK (ID_ANGAJAT >= 200 AND ID_ANGAJAT <= 299);

--VALORI TABEL DONATORI

INSERT INTO DONATORI (ID_DONATOR, NUME_DONATOR, GRUPA_SANGE_DONATOR,
DATA_RECOLTARE, SEX_DONATOR, DATA_NASTERE_DONATOR, ID_ANGAJAT)

VALUES (15000, 'GEORGESCU IONELA', '0(I)+', TO_DATE('15-12-2022', 'DD-MM-YYYY'), 'F', TO_DATE('18-
02-1966', 'DD-MM-YYYY'), 200);

INSERT INTO DONATORI (ID_DONATOR, NUME_DONATOR, GRUPA_SANGE_DONATOR,
DATA_RECOLTARE, SEX_DONATOR, DATA_NASTERE_DONATOR, ID_ANGAJAT)

VALUES (15001, 'CORINA CHITU', 'AB(IV)+', TO_DATE('15-12-2022', 'DD-MM-YYYY'), 'F', TO_DATE('22-05-
1977', 'DD-MM-YYYY'), 200);

INSERT INTO DONATORI (ID_DONATOR, NUME_DONATOR, GRUPA_SANGE_DONATOR,
DATA_RECOLTARE, SEX_DONATOR, DATA_NASTERE_DONATOR, ID_ANGAJAT)

VALUES (15002, 'MARIN OLGA', '0(I)+', TO_DATE('15-12-2022', 'DD-MM-YYYY'), 'F', TO_DATE('05-07-
1998', 'DD-MM-YYYY'), 201);

INSERT INTO DONATORI (ID_DONATOR, NUME_DONATOR, GRUPA_SANGE_DONATOR,
DATA_RECOLTARE, SEX_DONATOR, DATA_NASTERE_DONATOR, ID_ANGAJAT)

VALUES (15003, 'TANASE ELENA', 'B(III)-', TO_DATE('16-12-2022', 'DD-MM-YYYY'), 'F', TO_DATE('27-10-
1973', 'DD-MM-YYYY'), 201);

INSERT INTO DONATORI (ID_DONATOR, NUME_DONATOR, GRUPA_SANGE_DONATOR,
DATA_RECOLTARE, SEX_DONATOR, DATA_NASTERE_DONATOR, ID_ANGAJAT)

VALUES (15004, 'NEAGOE NATALIA', 'A(II)-', TO_DATE('19-12-2022', 'DD-MM-YYYY'), 'F', TO_DATE('19-
07-1978', 'DD-MM-YYYY'), 202);

INSERT INTO DONATORI (ID_DONATOR, NUME_DONATOR, GRUPA_SANGE_DONATOR,
DATA_RECOLTARE, SEX_DONATOR, DATA_NASTERE_DONATOR, ID_ANGAJAT)

VALUES (15005, 'DOBRE DIANA', 'A(II)+', TO_DATE('20-12-2022', 'DD-MM-YYYY'), 'F', TO_DATE('12-01-
1989', 'DD-MM-YYYY'), 202);

INSERT INTO DONATORI (ID_DONATOR, NUME_DONATOR, GRUPA_SANGE_DONATOR,
DATA_RECOLTARE, SEX_DONATOR, DATA_NASTERE_DONATOR, ID_ANGAJAT)

```

```
VALUES (15006, 'NEGRESCU EUGEN', 'B(III)+', TO_DATE('20-12-2022', 'DD-MM-YYYY'), 'M', TO_DATE('13-04-1994', 'DD-MM-YYYY'), 203);
```

```
INSERT INTO DONATORI (ID_DONATOR, NUME_DONATOR, GRUPA_SANGE_DONATOR, DATA_RECOLTARE, SEX_DONATOR, DATA_NASTERE_DONATOR, ID_ANGAJAT)
```

```
VALUES (15007, 'RADUCIOIU OCTAVIAN', '0(I)-', TO_DATE('21-12-2022', 'DD-MM-YYYY'), 'M', TO_DATE('21-07-1988', 'DD-MM-YYYY'), 203);
```

```
INSERT INTO DONATORI (ID_DONATOR, NUME_DONATOR, GRUPA_SANGE_DONATOR, DATA_RECOLTARE, SEX_DONATOR, DATA_NASTERE_DONATOR, ID_ANGAJAT)
```

```
VALUES (15008, 'DINU FLORIN', 'A(II)+', TO_DATE('22-12-2022', 'DD-MM-YYYY'), 'M', TO_DATE('26-11-2002', 'DD-MM-YYYY'), 204);
```

```
INSERT INTO DONATORI (ID_DONATOR, NUME_DONATOR, GRUPA_SANGE_DONATOR, DATA_RECOLTARE, SEX_DONATOR, DATA_NASTERE_DONATOR, ID_ANGAJAT)
```

```
VALUES (15009, 'MOISUC CIPRIAN', 'AB(IV)-', TO_DATE('23-12-2022', 'DD-MM-YYYY'), 'M', TO_DATE('30-08-1978', 'DD-MM-YYYY'), 204);
```

```
INSERT INTO DONATORI (ID_DONATOR, NUME_DONATOR, GRUPA_SANGE_DONATOR, DATA_RECOLTARE, SEX_DONATOR, DATA_NASTERE_DONATOR, ID_ANGAJAT)
```

```
VALUES (15010, 'CHIRILA ALEX', 'B(III)+', TO_DATE('24-12-2022', 'DD-MM-YYYY'), 'M', TO_DATE('09-06-2000', 'DD-MM-YYYY'), 205);
```

```
INSERT INTO DONATORI (ID_DONATOR, NUME_DONATOR, GRUPA_SANGE_DONATOR, DATA_RECOLTARE, SEX_DONATOR, DATA_NASTERE_DONATOR, ID_ANGAJAT)
```

```
VALUES (15011, 'VOICULET VASILE', 'AB(IV)+', TO_DATE('24-12-2022', 'DD-MM-YYYY'), 'M', TO_DATE('03-07-1980', 'DD-MM-YYYY'), 205);
```

```
INSERT INTO DONATORI (ID_DONATOR, NUME_DONATOR, GRUPA_SANGE_DONATOR, DATA_RECOLTARE, SEX_DONATOR, DATA_NASTERE_DONATOR, ID_ANGAJAT)
```

```
VALUES (15012, 'VIRGIL IONUT', 'A(II)+', TO_DATE('24-12-2022', 'DD-MM-YYYY'), 'M', TO_DATE('29-11-1983', 'DD-MM-YYYY'), 206);
```

```
--TABEL PROBE_SANGE
```

```
CREATE TABLE PROBE_SANGE
```

```
(
```

```
    ID_PROBA NUMBER(6) NOT NULL PRIMARY KEY,
```

```
    STATUS NUMBER(2),
```

```
    ID_DONATOR NUMBER(6) NOT NULL,
```

```
    ID_ANGAJAT NUMBER(6) NOT NULL
```

```
);
```

```
--CONSTRAINTS TABEL PROBE_SANGE
```

```
ALTER TABLE PROBE_SANGE
```

```
ADD CONSTRAINT PS_ID_DONATOR_FK FOREIGN KEY (ID_DONATOR) REFERENCES DONATORI
(ID_DONATOR);
```

```
ALTER TABLE PROBE_SANGE
```

```
ADD CONSTRAINT PS_ID_ANGAJAT_FK FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJATI_BS
(ID_ANGAJAT);
```

```
ALTER TABLE PROBE_SANGE
```

```
ADD CONSTRAINT PS_ID_ANGAJAT_CK CHECK (ID_ANGAJAT >= 400 AND ID_ANGAJAT <= 499);
```

```
ALTER TABLE PROBE_SANGE
```

```
ADD CONSTRAINT STATUS_CK CHECK (STATUS = 1 OR STATUS = 0);
```

```
--VALORI TABRL PROBE_SANGE
```

```
INSERT INTO PROBE_SANGE (ID_PROBA, STATUS, ID_DONATOR, ID_ANGAJAT) VALUES (100, 0,
15000, 400);
```

```
INSERT INTO PROBE_SANGE (ID_PROBA, STATUS, ID_DONATOR, ID_ANGAJAT) VALUES (101, 0,
15001, 400);
```

```
INSERT INTO PROBE_SANGE (ID_PROBA, STATUS, ID_DONATOR, ID_ANGAJAT) VALUES (102, 1,
15002, 400);
```

```
INSERT INTO PROBE_SANGE (ID_PROBA, STATUS, ID_DONATOR, ID_ANGAJAT) VALUES (103, 0,
15003, 400);
```

```
INSERT INTO PROBE_SANGE (ID_PROBA, STATUS, ID_DONATOR, ID_ANGAJAT) VALUES (104, 1,
15004, 401);
```

```
INSERT INTO PROBE_SANGE (ID_PROBA, STATUS, ID_DONATOR, ID_ANGAJAT) VALUES (105, 1,
15005, 401);
```

```
INSERT INTO PROBE_SANGE (ID_PROBA, STATUS, ID_DONATOR, ID_ANGAJAT) VALUES (106, 0,
15006, 401);
```

```
INSERT INTO PROBE_SANGE (ID_PROBA, STATUS, ID_DONATOR, ID_ANGAJAT) VALUES (107, 0,
15007, 401);
```

```
INSERT INTO PROBE_SANGE (ID_PROBA, STATUS, ID_DONATOR, ID_ANGAJAT) VALUES (108, 1,
15008, 402);
```

```
INSERT INTO PROBE_SANGE (ID_PROBA, STATUS, ID_DONATOR, ID_ANGAJAT) VALUES (109, 1,
15009, 402);
```

```
INSERT INTO PROBE_SANGE (ID_PROBA, STATUS, ID_DONATOR, ID_ANGAJAT) VALUES (110, 0,
15010, 402);
```

```
INSERT INTO PROBE_SANGE (ID_PROBA, STATUS, ID_DONATOR, ID_ANGAJAT) VALUES (111, 1,
15011, 402);
```

```
INSERT INTO PROBE_SANGE (ID_PROBA, STATUS, ID_DONATOR, ID_ANGAJAT) VALUES (112, 0,
15012, 400);
```

```
--ACTUALIZARE INREGISTRATI
```

```
UPDATE DONATORI
SET GRUPA_SANGE_DONATOR = 'B(III)-'
WHERE ID_DONATOR = 15006;
```

```
UPDATE SPITALE
SET TELEFON_SPITAL = '0748.558.397'
WHERE DENUMIRE_SPITAL LIKE 'VICTOR BABES';
```

```
UPDATE ANGAJATI_BS
SET ID_MANAGER = 302
WHERE ID_ANGAJAT = 402;
```

```
UPDATE PACIENTI
SET ID_SPITAL = 1
WHERE NUME_PACIENT LIKE '%LEC%';
```

```
UPDATE PROBE_SANGE
SET STATUS = 0
WHERE ID_PROBA = 111;
```

```
--INTEROGARI
```

```
--1-SA SE AFISEZE NUMELE, EMAILUL ANGAJATULUI DIN DEPARTAMENTUL ADMINISTRARE CU  
CEL MAI MIC SALARIU
```

```
SELECT ANG.NUME_ANGAJAT, ANG.EMAIL_ANGAJAT
FROM ANGAJATI_BS ANG, FUNCTII_BS F
WHERE ANG.ID_FUNCTIE = F.ID_FUNCTIE
AND F.DEPARTAMENT LIKE 'ADMINISTRARE'
AND ANG.SALARIU IN (SELECT MIN(ANG.SALARIU)
                     FROM ANGAJATI_BS ANG, FUNCTII_BS F
                     WHERE ANG.ID_FUNCTIE = F.ID_FUNCTIE
                     AND F.DEPARTAMENT LIKE 'ADMINISTRARE');
```

--2-SA SE AFISEZE NUMELE DONATORILOR DIN ZIUA DE 15 DECEMBRIE SI NUMELE RECOLTORILOR LOR

```
SELECT D.NUME_DONATOR, BS.NUME_ANGAJAT
FROM DONATORI D, ANGAJATI_BS BS
WHERE D.ID_ANGAJAT = BS.ID_ANGAJAT
AND D.DATA_RECOLTARE = TO_DATE('15-12-2022', 'DD-MM-YYYY');
```

--3-SA SE AFISEZE NUMARUL PROBELOR CONTAMINATE DIN INTERVALUL 15-20 DECEMBRIE

```
SELECT COUNT(PS.ID_PROBA) AS NR_PROBE_CONTAMINATE
FROM PROBE_SANGE PS, DONATORI D
WHERE PS.ID_DONATOR = D.ID_DONATOR
AND PS.STATUS = 1
AND EXTRACT(DAY FROM D.DATA_RECOLTARE) BETWEEN 15 AND 20;
```

--4-SA SE AFISEZE NUMELE, DEPARTAMENTUL IN CARE LUCREAZA SI NIVELUL IERARHIC PENTRU TOTI SUBORDONATI

--ANGAJATULUI CU ID 101, CARE NU AU ALTI SUBORDONATI

```
SELECT ANG.NUME_ANGAJAT, F.DEPARTAMENT, LEVEL
FROM ANGAJATI_BS ANG, FUNCTII_BS F
WHERE ANG.ID_FUNCTIE = F.ID_FUNCTIE
AND CONNECT_BY_ISLEAF = 1
CONNECT BY PRIOR ANG.ID_ANGAJAT = ANG.ID_MANAGER
START WITH ANG.ID_ANGAJAT = 101;
```

--5-SA SE AFISEZE DENUMIREA SPITALULUI SI NUMARUL PACIENTILOR ASOCIATI

```
SELECT SP.DENUMIRE_SPITAL, COUNT(P.ID_SPITAL) AS NUMAR_PACIENTI
FROM SPITALE SP, PACIENTI P
WHERE P.ID_SPITAL = SP.ID_SPITAL
GROUP BY SP.DENUMIRE_SPITAL;
```

--6-SA SE AFISEZE NUMELE, GRUPA SANGUINA SI DATA NASTERII PACIENTILOR NASCUTI DUPA ANUL 2000

```

SELECT NUME_DONATOR, GRUPA_SANGE_DONATOR, DATA_NASTERE_DONATOR
FROM DONATORI
WHERE EXTRACT(YEAR FROM DATA_NASTERE_DONATOR) >= 2000;

```

```

--7-SA SE AFISEZE NUMELE, SALARIUL INITIAL SI SALARIUL ANGAJATILOR DUPA CE ESTE
IMPOZITAT

```

```

--IMPOZITUL SE APLICA ASTFEL 3000-4499 10% 4500-5999 12% 6000+ 14%

```

```

SELECT NUME_ANGAJAT, SALARIU,
CASE
    WHEN SALARIU BETWEEN 3000 AND 4499 THEN 0.9*SALARIU
    WHEN SALARIU BETWEEN 4500 AND 5999 THEN 0.88*SALARIU
    WHEN SALARIU >= 6000 THEN 0.86*SALARIU
    ELSE 0
END AS SALARIU_IMPOZITAT
FROM ANGAJATI_BS;

```

```

--8-SA SE AFISEZE NUMELE DONATORILOR A CAROR PROBA DE SANGE A FOST ANALIZATA DE
ANALISTUL BACTERIOLOG

```

```

--CU NUMELE DINU EUGEN

```

```

SELECT D.NUME_DONATOR
FROM ANGAJATI_BS ANG, PROBE_SANGE PS, DONATORI D
WHERE D.ID_DONATOR = PS.ID_DONATOR
AND PS.ID_ANGAJAT = ANG.ID_ANGAJAT
AND ANG.NUME_ANGAJAT LIKE 'DINU EUGEN';

```

```

--9-SA SE AFISEZE NUMELE, EMAILUL ANGAJATILOR AL CAROR NUME INCEPE CU LITERA D DIN
DEPARTAMENTUL CERCETARE

```

```

SELECT ANG.NUME_ANGAJAT, ANG.EMAIL_ANGAJAT
FROM ANGAJATI_BS ANG, FUNCTII_BS F
WHERE ANG.ID_FUNCTIE = F.ID_FUNCTIE
AND DEPARTAMENT LIKE 'CERCETARE'
AND ANG.NUME_ANGAJAT LIKE 'D%';

```

--10-SA SE AFISEZE MEDIA SALARIULUI ANGAJATILOR CU VECHIME SUB 10 ANI

SELECT ROUND(AVG(SALARIU), 2) AS SALARIU\_MEDIU

FROM ANGAJATI\_BS

WHERE EXTRACT(YEAR FROM SYSDATE) - 10 < EXTRACT (YEAR FROM DATA\_ANGAJARE);

--11-SA SE AFISEZE NUMELE RECOLTORULUI, NUMARUL DE PERSOANE RECOLTATE AL CAROR NUME INCEP CU G

SELECT ANG.NUME\_ANGAJAT, COUNT(D.NUME\_DONATOR) AS NR\_DONATORI

FROM ANGAJATI\_BS ANG, DONATORI D

WHERE ANG.ID\_ANGAJAT = D.ID\_ANGAJAT

GROUP BY ANG.NUME\_ANGAJAT

HAVING ANG.NUME\_ANGAJAT LIKE ('G%');

--12-SA SE AFISEZE NUMELE, SALARIUL ANGAJATILOR DIN DEPARTAMENTUL RECOLTARE CU VECHIME PESTE 10 ANI

SELECT ANG.NUME\_ANGAJAT, ANG.SALARIU

FROM ANGAJATI\_BS ANG, FUNCTII\_BS F

WHERE ANG.ID\_FUNCTIE = F.ID\_FUNCTIE

AND F.DEPARTAMENT LIKE 'RECOLTARE'

AND EXTRACT(YEAR FROM SYSDATE) - 12 > EXTRACT (YEAR FROM DATA\_ANGAJARE);

--13-SA SE AFISEZE DENUMIREA SPITALULUI, GRUPA SI NUMARUL PACIENTILOR CU GRUPA NECESARA 0(II)-

--PENTRU SPITATUL SFANTA MARIA

SELECT SP.DENUMIRE\_SPITAL, P.GRUPA\_SANGE\_NECESARA, COUNT(P.ID\_PACIENT) AS NR\_PACIENTI

FROM SPITALE SP, PACIENTI P

WHERE SP.ID\_SPITAL = P.ID\_SPITAL

AND P.GRUPA\_SANGE\_NECESARA LIKE '0(I)-'

GROUP BY SP.DENUMIRE\_SPITAL, P.GRUPA\_SANGE\_NECESARA

HAVING SP.DENUMIRE\_SPITAL LIKE 'SFANTA MARIA';

--14-SA SE AFISEZE NUMELE DONATORULUI, ID-UL PROBEI PENTRU PROBELE DE SANGE NECONTAMINATE



```

SELECT D.NUME_DONATOR, PS.ID_PROBA
FROM DONATORI D, PROBE_SANGE PS
WHERE D.ID_DONATOR = PS.ID_DONATOR
AND PS.STATUS = 0;

```

--15-SA SE AFISEZE NUMELE, GRUPA SI SEXUL PENTRU DONATORII NASCUTII INTRE 1960-1980

```

SELECT NUME_DONATOR, GRUPA_SANGE_DONATOR, SEX_DONATOR
FROM DONATORI
WHERE EXTRACT(YEAR FROM DATA_NASTERE_DONATOR) BETWEEN 1960 AND 1980;

```

--16-SA SE AFISEZE NUMELE SI NUMARUL PACIENTILOR DE SEX FEMININ AI SPITALULUI GEROTA

```

SELECT SP.DENUMIRE_SPITAL, COUNT(P.SEX_PACIENT) AS NR_PACIENTI
FROM SPITALE SP, PACIENTI P
WHERE SP.ID_SPITAL = P.ID_SPITAL
AND P.SEX_PACIENT LIKE 'F'
GROUP BY SP.DENUMIRE_SPITAL
HAVING SP.DENUMIRE_SPITAL LIKE 'GEROTA';

```

--17-SA SE AFISEZE NUMELE MANAGERULUI SI NUMARUL DE PACIENTI AJUTATE

```

SELECT ANG.NUME_ANGAJAT, COUNT(P.ID_PACIENT) AS NR_PACIENTI
FROM ANGAJATI_BS ANG, SPITALE SP, PACIENTI P
WHERE SP.ID_ANGAJAT = ANG.ID_ANGAJAT
AND SP.ID_SPITAL = P.ID_SPITAL
GROUP BY ANG.NUME_ANGAJAT;

```

--18-SA SE AFISEZE NUMELE RECOLTORULUI, NUMARUL PERSOANELOR DE SEX MASCULIN CARE AU DONAT SANGE

--IN LUNA DECEMBRIE

```

SELECT ANG.NUME_ANGAJAT, COUNT(D.ID_DONATOR) AS NR_DONATORI
FROM ANGAJATI_BS ANG, DONATORI D
WHERE ANG.ID_ANGAJAT = D.ID_ANGAJAT
AND D.SEX_DONATOR LIKE 'M'

```

AND EXTRACT (MONTH FROM D.DATA\_RECOLTARE) = 12

GROUP BY ANG.NUME\_ANGAJAT;

--19-SA SE AFISEZE NUMELE, DATA RECOLTARII SI GRUPA PERSOANELOR CU GRUPA DIFERITA DE 0(I)

SELECT NUME\_DONATOR, DATA\_RECOLTARE, GRUPA\_SANGE\_DONATOR

FROM DONATORI

WHERE GRUPA\_SANGE\_DONATOR <> '0(I)+'

AND GRUPA\_SANGE\_DONATOR <> '0(I)-';

--20-SA SE AFISEZE NUMELE, DEPARTAMENTUL SI NIVELUL IERARHIC PENTRU ANGAJATII CARE AU SUBORDONATI SI AU

--ACEEASI FUNCTIE CA PAVEL OLIVIA

SELECT ANG.NUME\_ANGAJAT, F.DEPARTAMENT, LEVEL

FROM ANGAJATI\_BS ANG, FUNCTII\_BS F

WHERE ANG.ID\_FUNCTIE = F.ID\_FUNCTIE

AND CONNECT\_BY\_ISLEAF = 0

CONNECT BY PRIOR ANG.ID\_ANGAJAT = ANG.ID\_MANAGER

START WITH ANG.ID\_FUNCTIE IN (SELECT ID\_FUNCTIE FROM ANGAJATI\_BS WHERE NUME\_ANGAJAT = 'PAVEL OLIVIA');