

## PROIECT BAZE DE DATE

### Cuprins:

1. Prezentarea bazei de date
2. Diagrama entitate-relație
3. Diagrama conceptuală
4. Definirea tabelelor în Oracle
5. Adăugarea de informații coerente în tabelele create
6. Interogări

### 1. Prezentarea bazei de date

În acest proiect am ales să evidențiez utilitatea și importanța unei baze de date în cadrul acțiunilor întreprinse într-un spital, mai exact, în ceea ce privește relația dintre medici și pacienți, gestiunea programărilor.

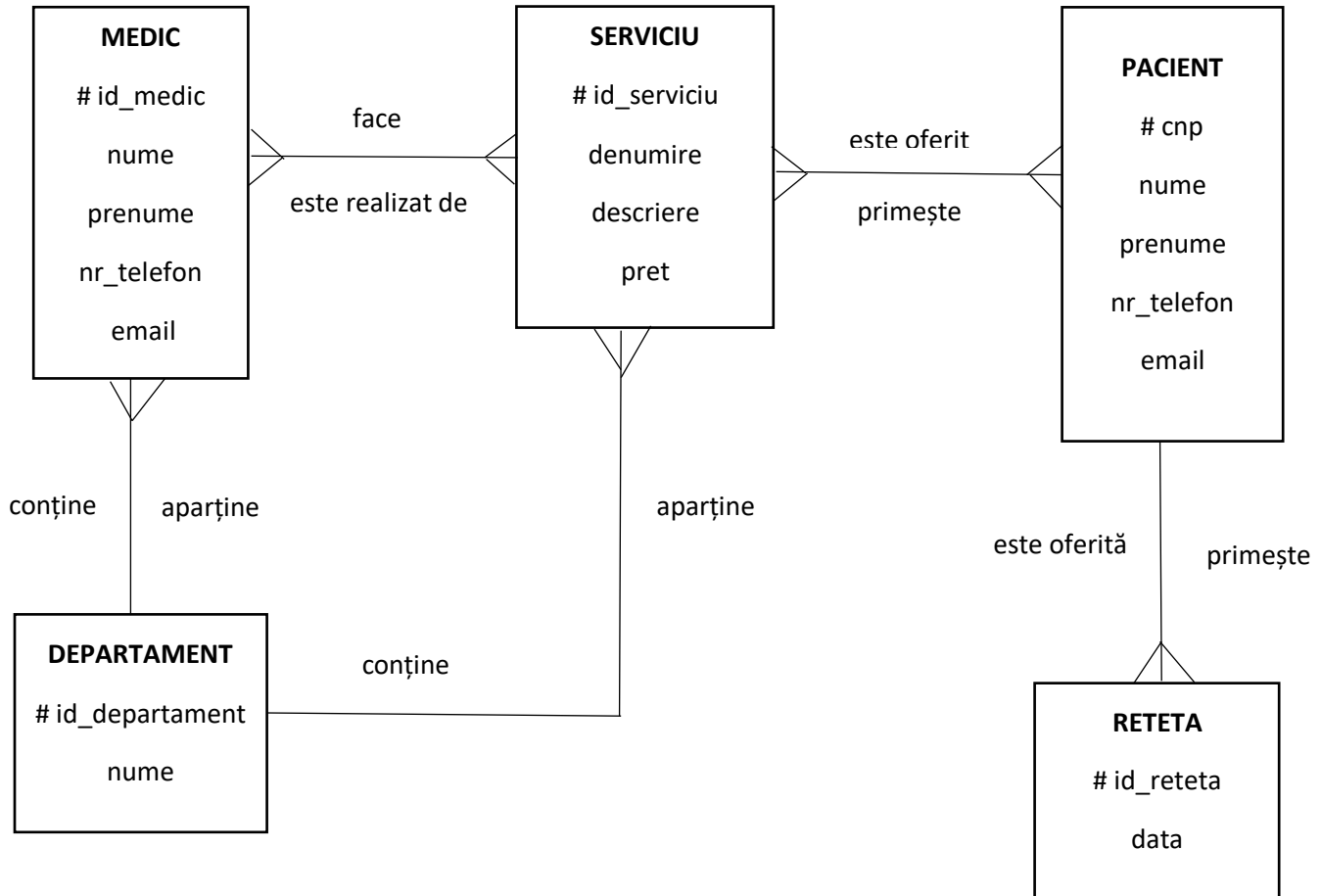
Primul pas a constat în realizarea diagramei entitate-relație și a diagramei conceptuale.

Acestea prezintă structura bazei de date cu tabelele și legăturile dintre acestea, ținând cont de regulile de aducere la forma normală 3. Baza de date conține în programările unor pacienți, programări care sunt preluate de medici din diferite departamente. În cadrul unei programări este realizat un anumit tip de serviciu (consultație, operație, etc.), iar în urma acesteia pacientul primește o rețetă.

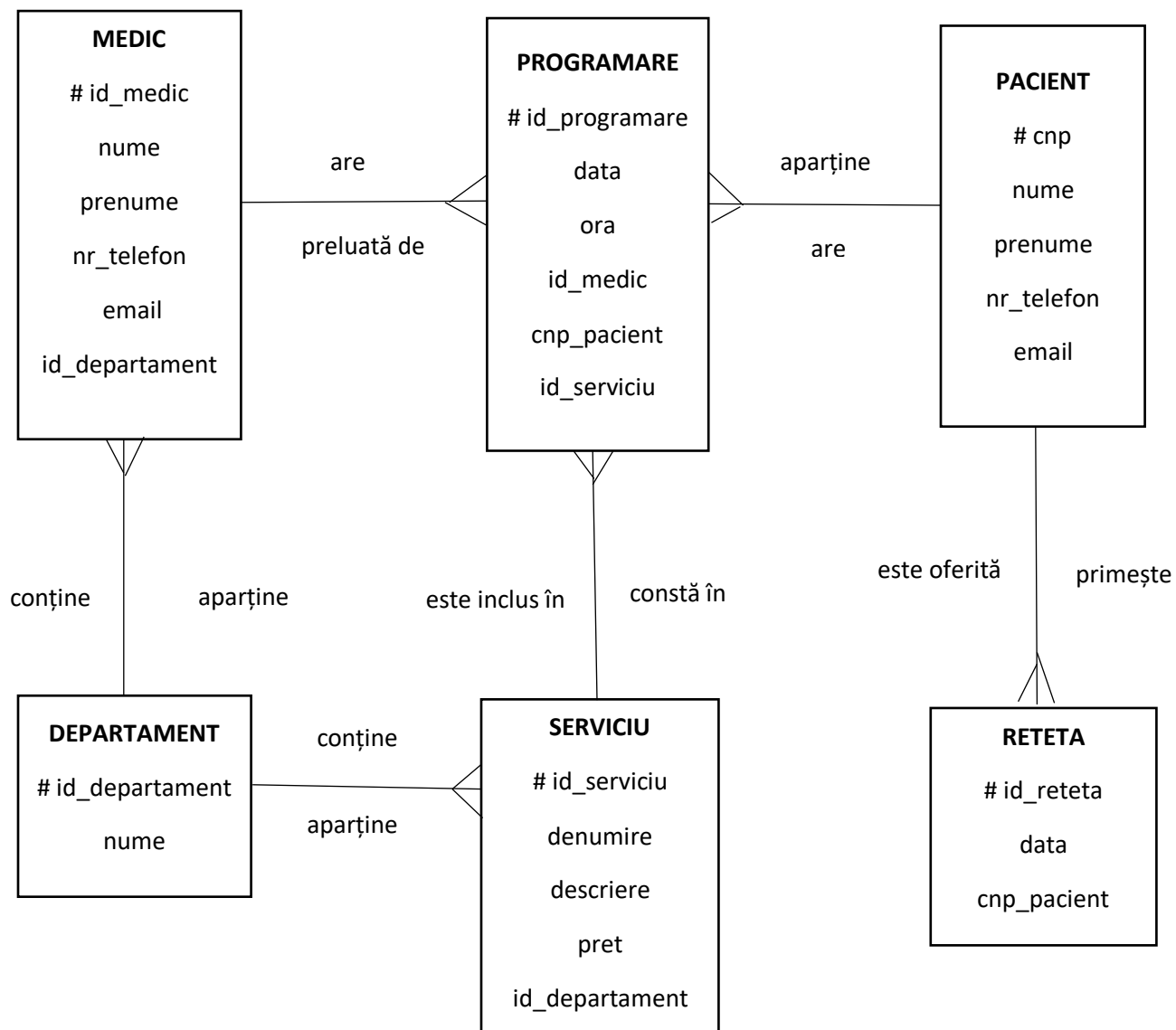
În a doua parte a proiectului am definit tabelele în Oracle, implementând toate constrângerile de integritate necesare. Urmând ca în a treia parte, să adaug informații coerente în tabelele create.

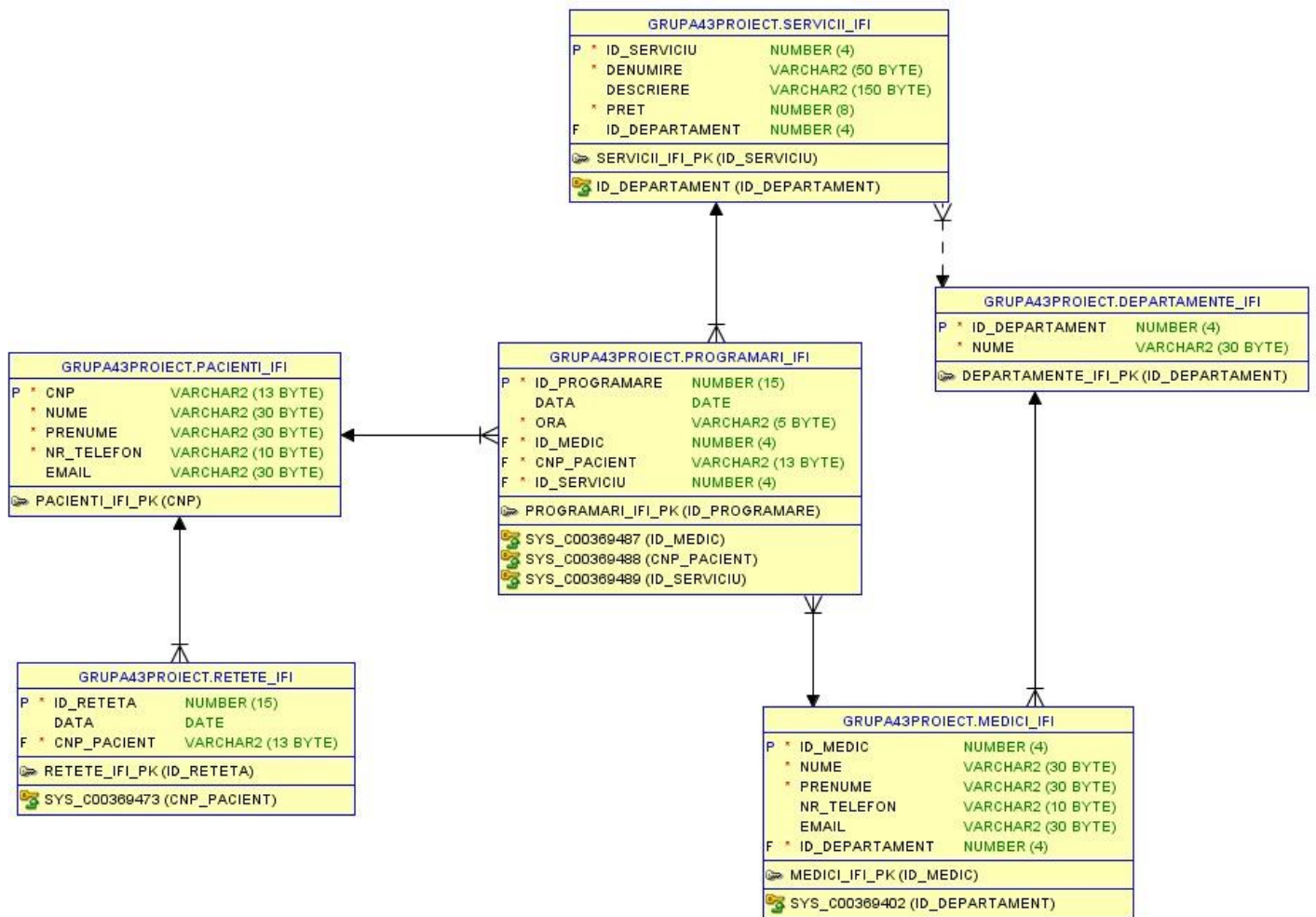
În cele din urmă am realizat diverse interogări asupra bazei de date.

## 2. Diagrama entitate-relație



### 3. Diagrama conceptuală





### Schema conceptuală:

**DEPARTAMENTE**(#id\_departament, nume)

**MEDICI**(#id\_medic, nume, prenume, nr\_telefon, email, id\_departament)

**SERVICII**(#id\_serviciu, denumire, descriere, pret, id\_departament)

**PACIENTI**(#cnp, nume, prenume, nr\_telefon, email)

**RETETE**(#id\_reteta, data, cnp\_pacient)

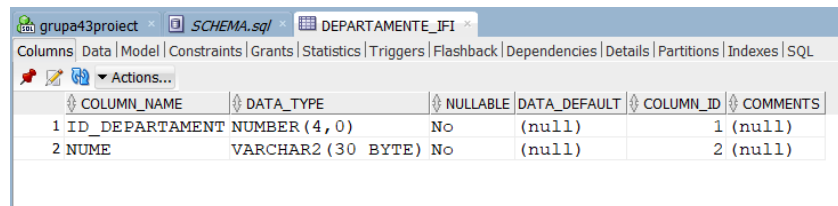
**PROGRAMARI**(#id\_programare, data, ora, id\_medic, cnp\_pacient, id\_serviciu)

## 4. Definirea tabelelor în Oracle

- Crearea tabelii **DEPARTAMENTE\_IFI**

```
CREATE TABLE DEPARTAMENTE_IFI  
(id_departament NUMBER(4) PRIMARY KEY,  
nume VARCHAR(30) NOT NULL);
```

Name	Null?	Type
-----	-----	-----
ID_DEPARTAMENT	NOT NULL	NUMBER (4)
NUME	NOT NULL	VARCHAR2 (30)



The screenshot shows the Oracle SQL Developer interface with the 'DEPARTAMENTE\_IFI' table selected. The 'Columns' tab is active, displaying the following table structure:

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
ID_DEPARTAMENT	NUMBER (4, 0)	No	(null)	1 (null)	
NUME	VARCHAR2 (30 BYTE)	No	(null)	2 (null)	

- Crearea tabelii **MEDICI\_IFI**

```
CREATE TABLE MEDICI_IFI  
(id_medic NUMBER(4) NOT NULL,  
nume VARCHAR2(30) NOT NULL,  
prenume VARCHAR(30) NOT NULL,  
nr_telefon VARCHAR(10),  
email VARCHAR(30),  
id_departament NUMBER(4) NOT NULL,  
PRIMARY KEY(id_medic),  
FOREIGN KEY(id_departament) REFERENCES DEPARTAMENTE_IFI(id_departament));
```

Name	Null?	Type
-----	-----	-----
ID_MEDIC	NOT NULL	NUMBER (4)
NUME	NOT NULL	VARCHAR2 (30)
PRENUME	NOT NULL	VARCHAR2 (30)
NR_TELEFON		VARCHAR2 (10)
EMAIL		VARCHAR2 (30)
ID_DEPARTAMENT	NOT NULL	NUMBER (4)

grupa43proiect x SCHEMA.sql x MEDICI_IFI x						
Columns   Data   Model   Constraints   Grants   Statistics   Triggers   Flashback   Dependencies   Details   Partitions   Indexes   SQL						
Actions...						
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	
1 ID_MEDIC	NUMBER (4, 0)	No	(null)	1	(null)	
2 NUME	VARCHAR2 (30 BYTE)	No	(null)	2	(null)	
3 PRENUME	VARCHAR2 (30 BYTE)	No	(null)	3	(null)	
4 NR_TELEFON	VARCHAR2 (10 BYTE)	Yes	(null)	4	(null)	
5 EMAIL	VARCHAR2 (30 BYTE)	Yes	(null)	5	(null)	
6 ID_DEPARTAMENT	NUMBER (4, 0)	No	(null)	6	(null)	

- Crearea tabelii **SERVICII\_IFI**

```
CREATE TABLE SERVICII_IFI
(id_serviciu NUMBER(4) PRIMARY KEY,
denumire VARCHAR(50) NOT NULL,
descriere VARCHAR(150),
pret NUMBER(8) NOT NULL);
```

```
ALTER TABLE SERVICII_IFI
ADD id_departament NUMBER(4);
```

```
ALTER TABLE SERVICII_IFI
ADD CONSTRAINT id_departament FOREIGN KEY (id_departament) REFERENCES
DEPARTAMENTE_IFI(id_departament);
```

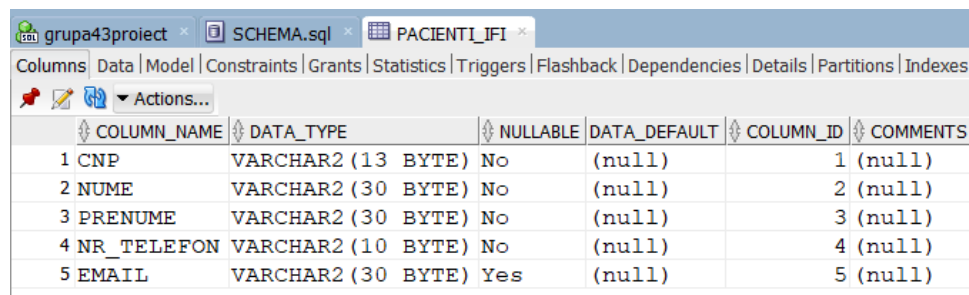
Name	Null?	Type
-----	-----	-----
ID_SERVICIU	NOT NULL	NUMBER (4)
DENUMIRE	NOT NULL	VARCHAR2 (50)
DESCRIERE		VARCHAR2 (150)
PRET	NOT NULL	NUMBER (8)
ID_DEPARTAMENT		NUMBER (4)

SCHEMA.sql x SQL.sql x SERVICII_IFI x						
Columns   Data   Model   Constraints   Grants   Statistics   Triggers   Flashback   Dependencies   Details   Partitions   Indexes   SQL						
Actions...						
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	
1 ID_SERVICIU	NUMBER (4, 0)	No	(null)	1	(null)	
2 DENUMIRE	VARCHAR2 (50 BYTE)	No	(null)	2	(null)	
3 DESCRIERE	VARCHAR2 (150 BYTE)	Yes	(null)	3	(null)	
4 PRET	NUMBER (8, 0)	No	(null)	4	(null)	
5 ID_DEPARTAMENT	NUMBER (4, 0)	Yes	(null)	5	(null)	

- Crearea tabelii **PACIENTI\_IFI**

```
CREATE TABLE PACIENTI_IFI
(cnp VARCHAR(13) PRIMARY KEY,
nume VARCHAR(30) NOT NULL,
prenume VARCHAR(30) NOT NULL,
nr_telefon VARCHAR(10) NOT NULL,
email VARCHAR(30));
```

Name	Null?	Type
CNP	NOT NULL	VARCHAR2 (13)
NUME	NOT NULL	VARCHAR2 (30)
PRENUME	NOT NULL	VARCHAR2 (30)
NR_TELEFON	NOT NULL	VARCHAR2 (10)
EMAIL		VARCHAR2 (30)



The screenshot shows the Oracle SQL Developer interface with the 'PACIENTI\_IFI' table selected. The 'Columns' tab is active, displaying the following table structure:

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 CNP	VARCHAR2 (13 BYTE)	No	(null)	1	(null)
2 NUME	VARCHAR2 (30 BYTE)	No	(null)	2	(null)
3 PRENUME	VARCHAR2 (30 BYTE)	No	(null)	3	(null)
4 NR_TELEFON	VARCHAR2 (10 BYTE)	No	(null)	4	(null)
5 EMAIL	VARCHAR2 (30 BYTE)	Yes	(null)	5	(null)

- Crearea tabelii **RETETE\_IFI**

```
CREATE TABLE RETETE_IFI
(id_reteta NUMBER(15) PRIMARY KEY,
data DATE DEFAULT SYSDATE,
cnp_pacient VARCHAR(13) NOT NULL,
FOREIGN KEY (cnp_pacient) REFERENCES PACIENTI_IFI(cnp));
```

Name	Null?	Type
ID_RETETA	NOT NULL	NUMBER (15)
DATA		DATE
CNP_PACIENT	NOT NULL	VARCHAR2 (13)

grupa43proiect x SCHEMA.sql x RETETE_IFI x					
Columns   Data   Model   Constraints   Grants   Statistics   Triggers   Flashback   Dependencies   Details   Partitions   Indexes   SQL					
Actions...					
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_RETETA	NUMBER (15, 0)	No	(null)	1	(null)
2 DATA	DATE	Yes	SYSDATE	2	(null)
3 CNP_PACIENT	VARCHAR2 (13 BYTE)	No	(null)	3	(null)

- Crearea tabelii **PROGRAMARI\_IFI**

```
CREATE TABLE PROGRAMARI_IFI
(id_programare NUMBER(15) PRIMARY KEY,
data DATE DEFAULT SYSDATE,
ora VARCHAR(5) NOT NULL,
id_medic NUMBER(4) NOT NULL,
cnp_pacient VARCHAR(13) NOT NULL,
id_serviciu NUMBER(4) NOT NULL,
FOREIGN KEY(id_medic) REFERENCES MEDICI_IFI(id_medic),
FOREIGN KEY (cnp_pacient) REFERENCES PACIENTI_IFI(cnp),
FOREIGN KEY(id_serviciu) REFERENCES SERVICII_IFI(id_serviciu));
```

Name	Null?	Type
-----	-----	-----
ID_PROGRAMARE	NOT NULL	NUMBER (15)
DATA		DATE
ORA	NOT NULL	VARCHAR2 (5)
ID_MEDIC	NOT NULL	NUMBER (4)
CNP_PACIENT	NOT NULL	VARCHAR2 (13)
ID_SERVICIU	NOT NULL	NUMBER (4)

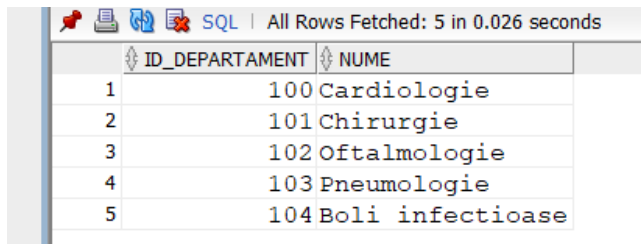
grupa43proiect x SCHEMA.sql x PROGRAMARI_IFI x					
Columns   Data   Model   Constraints   Grants   Statistics   Triggers   Flashback   Dependencies   Details   Partitions   Indexes   SQL					
Actions...					
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_PROGRAMARE	NUMBER (15, 0)	No	(null)	1	(null)
2 DATA	DATE	Yes	SYSDATE	2	(null)
3 ORA	VARCHAR2 (5 BYTE)	No	(null)	3	(null)
4 ID_MEDIC	NUMBER (4, 0)	No	(null)	4	(null)
5 CNP_PACIENT	VARCHAR2 (13 BYTE)	No	(null)	5	(null)
6 ID_SERVICIU	NUMBER (4, 0)	No	(null)	6	(null)



## 5. Adăugarea de informații coerente în tabelele create

- DEPARTAMNTE\_IFI

```
INSERT INTO DEPARTAMENTE_IFI
VALUES(100, 'Cardiologie');
INSERT INTO DEPARTAMENTE_IFI
VALUES(101, 'Chirurgie');
INSERT INTO DEPARTAMENTE_IFI
VALUES(102, 'Oftalmologie');
INSERT INTO DEPARTAMENTE_IFI
VALUES(103, 'Pneumologie');
INSERT INTO DEPARTAMENTE_IFI
VALUES(104, 'Boli infectioase');
```

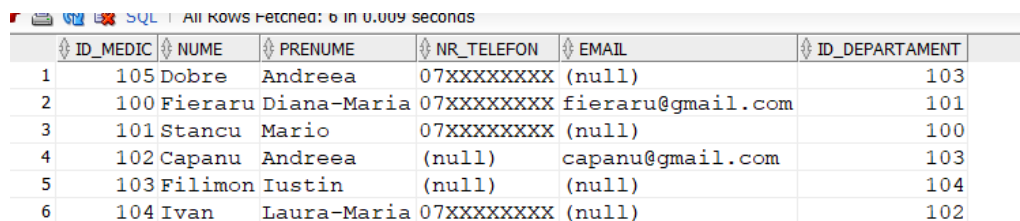


SQL | All Rows Fetched: 5 in 0.026 seconds

ID_DEPARTAMENT	NUME
1	100 Cardiologie
2	101 Chirurgie
3	102 Oftalmologie
4	103 Pneumologie
5	104 Boli infectioase

- MEDICI\_IFI

```
INSERT INTO MEDICI_IFI
VALUES(100, 'Fieraru', 'Diana-Maria', '07XXXXXXXX', 'fieraru@gmail.com', 101);
INSERT INTO MEDICI_IFI
VALUES(101, 'Stancu', 'Mario', '07XXXXXXXX', NULL, 100);
INSERT INTO MEDICI_IFI
VALUES(102, 'Capanu', 'Andreea', NULL, 'capanu@gmail.com', 103);
INSERT INTO MEDICI_IFI
VALUES(103, 'Filimon', 'Iustin', NULL, NULL, 104);
INSERT INTO MEDICI_IFI
VALUES(104, 'Ivan', 'Laura-Maria', '07XXXXXXXX', NULL, 102);
INSERT INTO MEDICI_IFI
VALUES(105, 'Dobre', 'Andreea', '07XXXXXXXX', NULL, 103);
```



SQL | All Rows Fetched: 6 in 0.009 seconds

ID_MEDIC	NUME	PRENUME	NR_TELEFON	EMAIL	ID_DEPARTAMENT
1	105 Dobre	Andreea	07XXXXXXXX	(null)	103
2	100 Fieraru	Diana-Maria	07XXXXXXXX	fieraru@gmail.com	101
3	101 Stancu	Mario	07XXXXXXXX	(null)	100
4	102 Capanu	Andreea	(null)	capanu@gmail.com	103
5	103 Filimon	Iustin	(null)	(null)	104
6	104 Ivan	Laura-Maria	07XXXXXXXX	(null)	102

- **SERVICII\_IFI**

```

INSERT INTO SERVICII_IFI
VALUES(100, 'Ecografie Doppler vascular', 'artere membre sup/inf/carotide', 250, 100);
INSERT INTO SERVICII_IFI
VALUES(101, 'Excizie tumori cutanate mici', NULL, 800, 101);
INSERT INTO SERVICII_IFI
VALUES(102, 'Excizie tumori cutanate medii', 'incl. ex.HP', 1200, 101);
INSERT INTO SERVICII_IFI
VALUES(103, 'Excizie tumori cutanate mari', 'reparare cu lambou cutanat; incl.HP', 1600, 101);
INSERT INTO SERVICII_IFI
VALUES(104, 'Consultatie completa oftalmologie', 'Examen digital biomicroscopic al polului anterior, tensiunea intraoculara, tehnologie laser autofluorescent, prescriptie de ochelari', 150, 102);
INSERT INTO SERVICII_IFI
VALUES(105, 'Pachet consultatie glaucom', 'tomografie nerv optic, gonioscopie, camp vizual Humphrey static + dinamic, pahimetrie + tonometrie', 400, 102);
INSERT INTO SERVICII_IFI
VALUES(106, 'Radiografie pulmonara', NULL, 150, 103);
INSERT INTO SERVICII_IFI
VALUES(107, 'Test FibroScan', NULL, 350, 104);

```

Query results

SQL | All Rows Fetched: 8 in 0.009 seconds

ID_SERVICIU	DENUMIRE	DESCRIERE	PRET	ID_
1	100Radiografie pulmonara	(null)	150	
2	101Excizie tumori cutanate mici	(null)	800	
3	102Excizie tumori cutanate medii	incl. ex.HP	1200	
4	103Excizie tumori cutanate mari	reparare cu lambou cutanat; incl.HP	1600	
5	104Consultatie completa oftalmologie	Examen digital biomicroscopic al polului anterior, tensiunea intraoculara, tehnologie laser autofluorescent, prescriptie de ochelari	150	
6	105Pachet consultatie glaucom	tomografie nerv optic, gonioscopie, camp vizual Humphrey static + dinamic, pahimetrie + tonometrie	400	
7	106Radiografie pulmonara	(null)	150	
8	107Test FibroScan	(null)	350	

- **PACIENTI\_IFI**

```

INSERT INTO PACIENTI_IFI
VALUES('1111111111111', 'Filip', 'Ioana', '07XXXXXXXX', 'filip@gmail.com');
INSERT INTO PACIENTI_IFI
VALUES('2222222222222', 'Filip', 'Denisa', '07XXXXXXXX', NULL);
INSERT INTO PACIENTI_IFI
VALUES('3333333333333', 'Simion', 'Andrei', '07XXXXXXXX', 'simion@gmail.com');
INSERT INTO PACIENTI_IFI
VALUES('4444444444444', 'Capitanu', 'Diana', '07XXXXXXXX', 'capitanu@gmail.com');
INSERT INTO PACIENTI_IFI
VALUES('5555555555555', 'Stanciu', 'Mihai', '07XXXXXXXX', NULL);

```

CNP	NUME	PRENUME	NR_TELEFON	EMAIL
1 11111111111111	Filip	Ioana	07XXXXXXXX	filip@gmail.com
2 22222222222222	Filip	Denisa	07XXXXXXXX	(null)
3 33333333333333	Simion	Andrei	07XXXXXXXX	simion@gmail.com
4 55555555555555	Stanciu	Mihai	07XXXXXXXX	(null)
5 44444444444444	Capitanu	Diana	07XXXXXXXX	capitanu@gmail.com

- **RETETE\_IFI**

```

INSERT INTO RETETE_IFI
VALUES(100, '20-Mar-2017', '11111111111111');
INSERT INTO RETETE_IFI
VALUES(101, '7-Dec-2015', '11111111111111');
INSERT INTO RETETE_IFI
VALUES(102, '25-Mar-2013', '33333333333333');
INSERT INTO RETETE_IFI
VALUES(103, '13-Jan-2017', '55555555555555');
INSERT INTO RETETE_IFI
VALUES(104, '9-Nov-2011', '22222222222222');
INSERT INTO RETETE_IFI
VALUES(105, '20-Mar-2014', '33333333333333');
INSERT INTO RETETE_IFI
VALUES(106, '2-Jun-2014', '11111111111111');
INSERT INTO RETETE_IFI
VALUES(107, '7-Jun-2014', '44444444444444');
INSERT INTO RETETE_IFI
VALUES(108, '20-Jan-2016', '55555555555555');
INSERT INTO RETETE_IFI
VALUES(109, '7-Sep-2012', '44444444444444');

```

SQL   All Rows Fetched: 10 in 0.114 seconds			
ID_RETETA	DATA	CNP_PACIENT	
1	100 20-MAR-17	11111111111111	
2	101 07-DEC-15	11111111111111	
3	102 25-MAR-13	33333333333333	
4	104 09-NOV-11	22222222222222	
5	103 13-JAN-17	55555555555555	
6	105 20-MAR-14	33333333333333	
7	106 02-JUN-14	11111111111111	
8	107 07-JUN-14	44444444444444	
9	108 20-JAN-16	55555555555555	
10	109 07-SEP-12	44444444444444	

- **PROGRAMARI\_IFI**

```

INSERT INTO PROGRAMARI_IFI
VALUES(100, '20-Mar-2017', '10:00', 103, '1111111111111', 107);
INSERT INTO PROGRAMARI_IFI
VALUES(101, '7-Dec-2015', '16:00', 100, '1111111111111', 103);
INSERT INTO PROGRAMARI_IFI
VALUES(102, '25-Mar-2013', '9:30', 102, '3333333333333', 106);
INSERT INTO PROGRAMARI_IFI
VALUES(103, '13-Jan-2017', '13:10', 104, '5555555555555', 104);
INSERT INTO PROGRAMARI_IFI
VALUES(104, '9-Nov-2011', '10:00', 100, '2222222222222', 101);
INSERT INTO PROGRAMARI_IFI
VALUES(105, '20-Mar-2014', '18:00', 100, '3333333333333', 103);
INSERT INTO PROGRAMARI_IFI
VALUES(106, '2-Jun-2014', '16:40', 101, '1111111111111', 100);
INSERT INTO PROGRAMARI_IFI
VALUES(107, '7-Jun-2014', '10:30', 104, '4444444444444', 105);
INSERT INTO PROGRAMARI_IFI
VALUES(108, '20-Jan-2016', '11:00', 103, '5555555555555', 107);
INSERT INTO PROGRAMARI_IFI
VALUES(109, '7-Sep-2012', '20:00', 100, '4444444444444', 102);

```

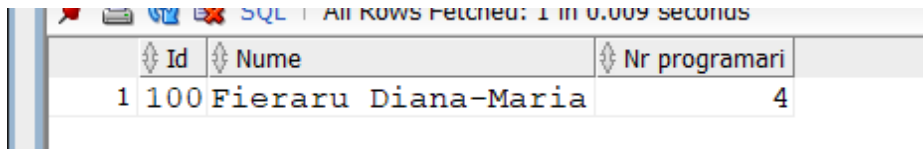
SQL | All Rows Fetched: 10 in 2.806 seconds

	ID_PROGRAMARE	DATA	ORA	ID_MEDIC	CNP_PACIENT	ID_SERVICIU
1	100	20-MAR-17	10:00	103	1111111111111	107
2	101	07-DEC-15	16:00	100	1111111111111	103
3	102	25-MAR-13	9:30	102	3333333333333	106
4	103	13-JAN-17	13:10	104	5555555555555	104
5	104	09-NOV-11	10:00	100	2222222222222	101
6	105	20-MAR-14	18:00	100	3333333333333	103
7	106	02-JUN-14	16:40	101	1111111111111	100
8	107	07-JUN-14	10:30	104	4444444444444	105
9	108	20-JAN-16	11:00	103	5555555555555	107
10	109	07-SEP-12	20:00	100	4444444444444	102

## 6. Interogări

- 1) Să se afișeze medicul/medicii cu numărul maxim de programări.

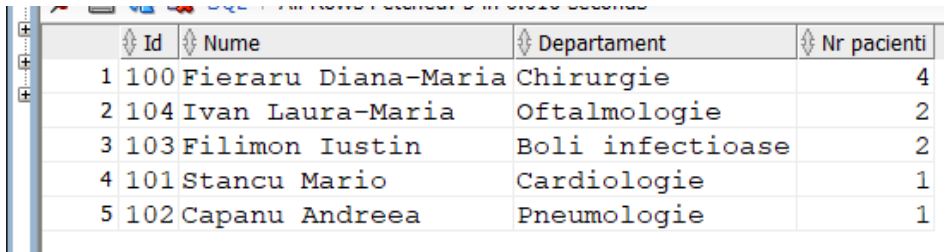
```
SELECT id_medic AS "Id", nume || ' ' || prenume AS "Nume", COUNT(id_programare) AS "Nr  
programari"  
FROM medici_ifi  
JOIN programari_ifi USING(id_medic)  
GROUP BY id_medic, nume, prenume  
HAVING COUNT(id_programare) = (SELECT MAX(COUNT(id_programare))  
FROM programari_ifi  
GROUP BY id_medic, nume, prenume);
```



Id	Nume	Nr programari
1 100	Fieraru Diana-Maria	4

- 2) Să se afișeze medicii, numele departamentelor din care fac parte și numărul total de pacienți pe care i-au avut. Ordonăți descrescător după numărul de pacienți.

```
SELECT m.id_medic AS "Id", m.nume || ' ' || m.prenume AS "Nume", d.nume AS "Departament",  
COUNT(pa.cnp) AS "Nr pacienti"  
FROM medici_ifi m  
JOIN programari_ifi p ON(m.id_medic = p.id_medic)  
JOIN pacienti_ifi pa ON(p.cnp_pacient = pa.cnp)  
JOIN departamente_ifi d ON(m.id_departament = d.id_departament)  
GROUP BY m.id_medic, m.nume, m.prenume, d.nume  
ORDER BY 4 DESC;
```



Id	Nume	Departament	Nr pacienti
1 100	Fieraru Diana-Maria	Chirurgie	4
2 104	Ivan Laura-Maria	Oftalmologie	2
3 103	Filimon Iustin	Boli infectioase	2
4 101	Stancu Mario	Cardiologie	1
5 102	Capanu Andreea	Pneumologie	1

- 3) Să se afișeze media prețurilor serviciilor pentru fiecare departament.

```
SELECT d.id_departament, d.nume, AVG(s.pret) AS "Media preturilor"  
FROM departamente_ifi d  
JOIN servicii_ifi s ON(s.id_departament = d.id_departament)  
GROUP BY d.id_departament, d.nume;
```

SQL   All Rows Fetched: 5 in 0.006 seconds			
	ID_DEPARTAMENT	NUME	Media preturilor
1	101	Chirurgie	1200
2	103	Pneumologie	150
3	100	Cardiologie	150
4	102	Oftalmologie	275
5	104	Boli infectioase	350

- 4) De câte luni nu a mai avut o programare pacientul cu numele Stanciu?

```
SELECT MIN(ROUND(MONTHS_BETWEEN(SYSDATE, pr.data))) AS "Nr luni"
FROM programari_ifi pr
JOIN pacienti_ifi pa ON(pr.cnp_pacient = pa.cnp)
WHERE UPPER(pa.numa) LIKE 'STANCIU';
```

SQL   All Rows Fetched: 1 in 0.000 seconds	
	Nr luni
1	41

- 5) Să se afișeze serviciile cu prețuri cuprinse între 150 și 500 sau serviciile care au fost oferite pacienților a căror nume încep cu 'F'.

```
SELECT s1.id_serviciu, s1.denumire, s1.descriere, s1.pret
FROM servicii_ifi s1
WHERE s1.pret BETWEEN 150 AND 500
UNION
SELECT s2.id_serviciu, s2.denumire, s2.descriere, s2.pret
FROM servicii_ifi s2
JOIN programari_ifi pr ON(pr.id_serviciu = s2.id_serviciu)
JOIN pacienti_ifi pa ON(pa.cnp = pr.cnp_pacient)
WHERE UPPER(pa.numa) LIKE 'F%';
```

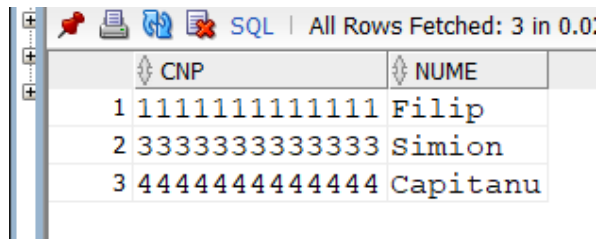
Query Result   All Rows Fetched: 5 in 0.014 seconds			
ID_SERVICIU	DENUMIRE	DESCRIERE	PRET
1	100 Radiografie pulmonara	(null)	150
2	104 Consultatie completa oftalmologie	Examen digital biomicroscopic al polului anterior, tensiunea intraoculara, tehnologie laser autofluorescent, prescriptie de ochelari	150
3	105 Pachet consultatie glaucom	tomografie nerv optic, gonioscopie, camp vizual Humphrey static + dinamic, pahimetrie + tonometrie	400
4	106 Radiografie pulmonara	(null)	150
5	107 Test FibroScan	(null)	350

- 6) Să se afișeze cnp-ul și numele pacienților care au fost tratați de cel puțin aceiași medici care l-au tratat pe pacientul cu cnp-ul '222222222222'. Exclueți-l pe acesta.

```

SELECT DISTINCT cnp, nume
FROM pacienti_ifi pa
WHERE NOT EXISTS(SELECT 1
                  FROM programari_ifi pr1
                  WHERE pr1.cnp_pacient LIKE '222222222222' AND
                  NOT EXISTS(SELECT 2
                              FROM programari_ifi pr2
                              WHERE pr2.cnp_pacient = pa.cnp
                              AND pr2.id_medic = pr1.id_medic))
AND cnp <> '222222222222';

```



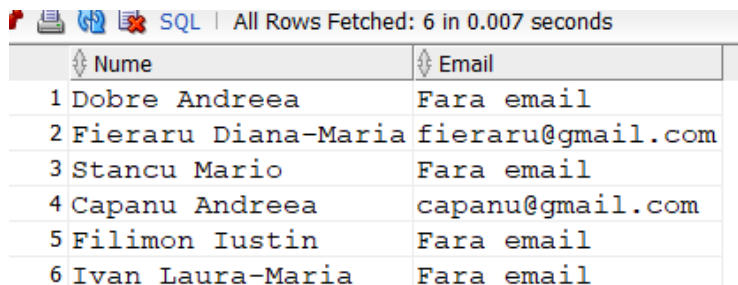
	CNP	NUME
1	11111111111111	Filip
2	33333333333333	Simion
3	44444444444444	Capitanu

- 7) Să se afișeze numele și adresele de email ale medicilor. Dacă un medic nu are adresa de email, sa se scrie "Fara email".

```

SELECT nume || ' ' || prenume AS "Nume", NVL(TO_CHAR(email), 'Fara email') AS "Email"
FROM medici_ifi;

```



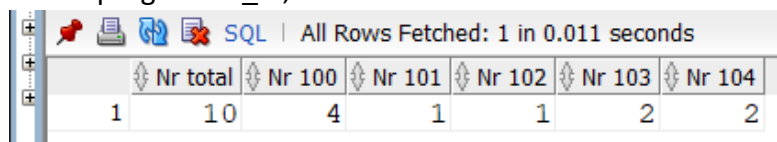
	Nume	Email
1	Dobre Andreea	Fara email
2	Fieraru Diana-Maria	fieraru@gmail.com
3	Stancu Mario	Fara email
4	Capanu Andreea	capanu@gmail.com
5	Filimon Iustin	Fara email
6	Ivan Laura-Maria	Fara email

- 8) Să se listeze numărul total de pacienți și numărul de pacienți ai fiecărui medic. Numiți coloanele astfel: Nr total, Nr id\_medic (unde id\_medic va lua id-ul fiecărui medic).

```

SELECT COUNT(cnp_pacient) AS "Nr total", SUM(DECODE(id_medic, 100, 1, 0)) AS "Nr 100",
      SUM(DECODE(id_medic, 101, 1, 0)) AS "Nr 101",
      SUM(DECODE(id_medic, 102, 1, 0)) AS "Nr 102",
      SUM(DECODE(id_medic, 103, 1, 0)) AS "Nr 103",
      SUM(DECODE(id_medic, 104, 1, 0)) AS "Nr 104"
FROM programari_ifi;

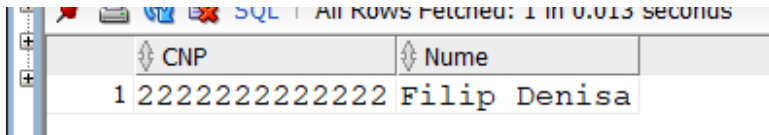
```



	Nr total	Nr 100	Nr 101	Nr 102	Nr 103	Nr 104
1	10	4	1	1	2	2

- 9) Să se afișeze lista pacienților care au avut programări preluate doar de medicul cu id-ul 100.

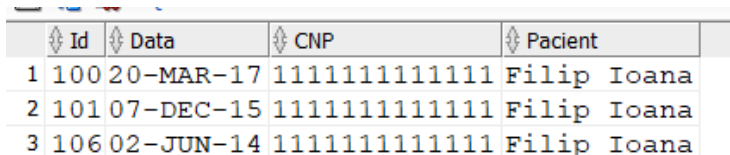
```
SELECT DISTINCT pr.cnp_pacient AS "CNP", pa.ume || ' ' || pa.prenume AS "Nume"
FROM programari_ifi pr
JOIN pacienti_ifi pa ON(pr.cnp_pacient = pa.cnp)
WHERE NOT EXISTS (SELECT id_medic
                   FROM programari_ifi pr1
                   WHERE pr.cnp_pacient = pr1.cnp_pacient
                   MINUS
                   SELECT id_medic
                   FROM programari_ifi pr2
                   WHERE id_medic = 100)
```



	CNP	Nume
1	222222222222	Filip Denisa

- 10) Să se listeze informații despre rețetele pacienților și numele acestora, pentru cei care au avut cel puțin 3 programări.

```
SELECT r.id_reteta AS "Id", r.data AS "Data", r.cnp_pacient AS "CNP", pa.ume || ' ' || pa.prenume
AS "Pacient"
FROM retete_ifi r
JOIN pacienti_ifi pa ON(r.cnp_pacient = pa.cnp)
WHERE r.cnp_pacient IN (SELECT cnp_pacient
                       FROM programari_ifi
                       GROUP BY cnp_pacient
                       HAVING COUNT(id_programare) >= 3);
```



	Id	Data	CNP	Pacient
1	100	20-MAR-17	111111111111	Filip Ioana
2	101	07-DEC-15	111111111111	Filip Ioana
3	106	02-JUN-14	111111111111	Filip Ioana



11) Să se afle de către care departament a fost preluat pacientul a cărei rețetă are id-ul 109.

```
SELECT r.id_reteta, pa.cnp, pr.id_medec, m.id_medec, m.id_departament, d.numa
FROM departamente_ifi d
JOIN medici_ifi m ON(m.id_departament = d.id_departament)
JOIN programari_ifi pr ON(pr.id_medec = m.id_medec)
JOIN pacienti_ifi pa ON(pa.cnp = pr.cnp_pacient)
JOIN retete_ifi r ON(r.cnp_pacient = pa.cnp)
WHERE r.id_reteta = 109 AND r.data = pr.data;
```

SQL | All Rows Fetched: 1 in 0.008 seconds

ID_RETETA	CNP	ID_MEDIC	ID_MEDIC_1	ID_DEPARTAMENT	NUME
1	1094444444444444	100	100	101	Chirurgie

12) Să se afișeze numele și prenumele medicilor, numele și preumele pacienților care au avut programări în luni impare ale anului. De asemenea să se afișeze și id-ul și luna programării.

```
SELECT m.numa || ' ' || m.prenume AS "Nume Medic", pa.numa || ' ' || pa.prenume AS "Nume
Pacient",
pr.id_programare AS "Id", TO_CHAR(pr.data, 'month') AS "Luna"
FROM medici_ifi m
JOIN programari_ifi pr ON(pr.id_medec = m.id_medec)
JOIN pacienti_ifi pa ON(pa.cnp = pr.cnp_pacient)
WHERE MOD(EXTRACT(MONTH FROM pr.data),2) = 1;
```

	Nume Medic	Nume Pacient	Id	Luna
1	Filimon Iustin	Filip Ioana	100	march
2	Capanu Andreea	Simion Andrei	102	march
3	Ivan Laura-Maria	Stanciu Mihai	103	january
4	Fieraru Diana-Maria	Filip Denisa	104	november
5	Fieraru Diana-Maria	Simion Andrei	105	march
6	Filimon Iustin	Stanciu Mihai	108	january
7	Fieraru Diana-Maria	Capitanu Diana	109	september

13) Să se afișeze id-ul și anul programărilor a căror dată e în același an în care au avut loc cele mai multe programări.

```
SELECT id_programare AS "Id", TO_CHAR(data, 'YYYY') AS "An"
FROM programari_ifi
WHERE TO_CHAR(data, 'YYYY') = (SELECT TO_CHAR(data, 'YYYY')
FROM programari_ifi
```

```

GROUP BY TO_CHAR(data, 'YYYY')
HAVING COUNT(id_programare) = (SELECT MAX(COUNT(id_programare))
FROM programari_ifi
GROUP BY TO_CHAR(data, 'YYYY')));

```

	Id	An
1	105	2014
2	106	2014
3	107	2014

- 14) Să se afișeze informații despre medici în formatul următor: "Medicul <nume prenume> face parte din departamentul <nume> si a avut <nr programari> programari".

```

SELECT 'Medicul ' || m.numa || ' ' || m.prenume || ' face parte din departamentul ' || d.numa ||
' si a avut nr de programari ' || CASE WHEN nr > 0 THEN nr
ELSE 0
END
FROM medici_ifi m
JOIN departamente_ifi d ON(d.id_departament = m.id_departament)
LEFT JOIN (SELECT id_medica, COUNT(id_programare) nr
FROM programari_ifi
GROUP BY id_medica) aux ON(m.id_medica = aux.id_medica);

```

	'MEDICUL'    M.NUMA    ' '    M.PRENUME    'FACE PARTE DIN DEPARTAMENTUL '    D.NUMA    ' SI A AVUT NR DE PROGRAMARI '    CASE WHEN NR > 0 THEN NR ELSE 0 END
1	Medicul Fieraru Diana-Maria face parte din departamentul Chirurgie si a avut nr de programari 4
2	Medicul Capanu Andreea face parte din departamentul Pneumologie si a avut nr de programari 1
3	Medicul Stancu Mario face parte din departamentul Cardiologie si a avut nr de programari 1
4	Medicul Ivan Laura-Maria face parte din departamentul Oftalmologie si a avut nr de programari 2
5	Medicul Filimon Iustin face parte din departamentul Boli infectioase si a avut nr de programari 2
6	Medicul Dobre Andreea face parte din departamentul Pneumologie si a avut nr de programari 0

- 15) Să se afișeze numele, prenumele și lungimea numelui pacienților, unde lungimea numelui este diferită de cea a prenumelui.

```

SELECT pa.numa || ' ' || pa.prenume AS "Nume", LENGTH(pa.numa) AS "Lungime nume"
FROM pacienti_ifi pa
WHERE NULLIF(LENGTH(pa.numa), LENGTH(pa.prenume)) IS NOT NULL;

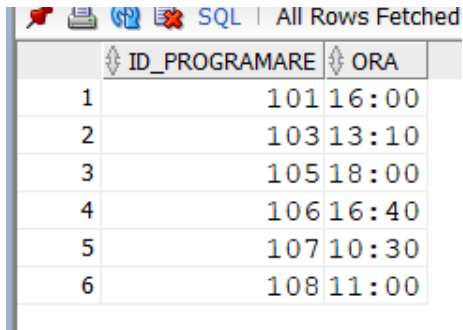
```

	Nume	Lungime nume
1	Filip Denisa	5
2	Stanciu Mihai	7
3	Capitanu Diana	8

- 16) Să se realizeze un script prin care să se afișeze id-ul programării și ora pentru programările a căror oră este între 2 ore introduse de utilizator.

```
ACCEPT p_or1 PROMPT 'ora de inceput(de forma 'HH:MM') '  
ACCEPT p_or2 PROMPT 'ora de sfarsit(de forma 'HH:MM') '  
SELECT id_programare, ora  
FROM programari_ifi  
WHERE ora BETWEEN &p_or1 AND &p_or2;
```

Pentru valorile introduse: '10:30' si '18:00' :



	ID_PROGRAMARE	ORA
1	101	16:00
2	103	13:10
3	105	18:00
4	106	16:40
5	107	10:30
6	108	11:00