College of Computer Technology (CCT)

Assessment Briefing Sheet

Student Name: Bianca Catalunha de Oliveira

Student Number: 2014305

Assignment Title: Structured Database Design

Module Title: Introduction to Databases

Lecturer: Colette Kiwan

Module Code: HC2060

Overall % of Final Grade: 30%

Due time: 11th December 2015

Assessment Descriptor

Learning outcomes assessed

- Have the skills required to use databases and improve data management.
- Be able to understand E-R-D Modelling and Normalisation concepts.
- Have an understanding of a Relational Database including DL and DML aspects.
- Produce normalised databases.
- Execute queries, reports and manipulate different objects.
- Design normalised relational databases.

Aim

The aim of this assignment is to design and implement a database using a structured approach.

Overview

You are required to design and implement a database for a GP Surgery using a structured approach based on the bellow requirements.

The submission for this module is a PDF report and the database create and implemented in your SQLite Server .

Requirements:

Design a database for a General Practice Surgery.

A General Practice Surgery needs to store information about its doctors, patients, and the medication being taken by each patient:-

You are given the following information

A doctor has a medical society registration number, name (first and last), date of birth and a speciality A patient has a unique number, name (first and last), date of birth, address, age, one contact number and a list of allergies.
Drugs have a unique id and a generic name.

You are given the following business rules

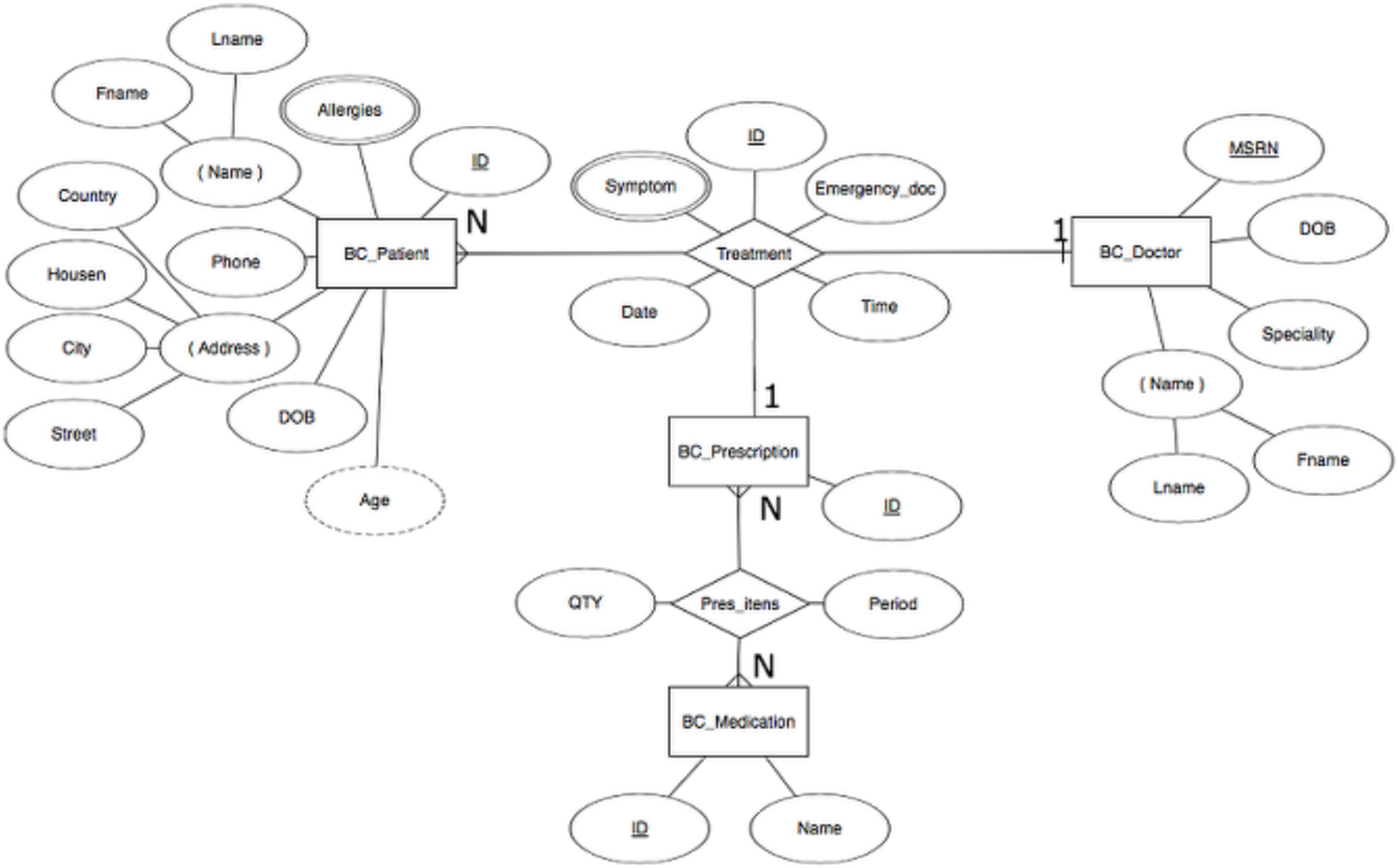
A patient is assigned to one doctor, but in emergencies can be seen by another a doctor. When a patient is treated by a doctor, the time, date and symptoms must be recorded. (The symptoms can be recorded as one long description)

Patients are prescribed drugs by doctors. The date (the drugs were prescribed), quantity, and time period (stated in days) of the drugs, prescribed by the doctor to the patient must be recorded. The Quantity is the dosage per day for e.g 125mg.

Design:

Conceptual:

ER Diagram



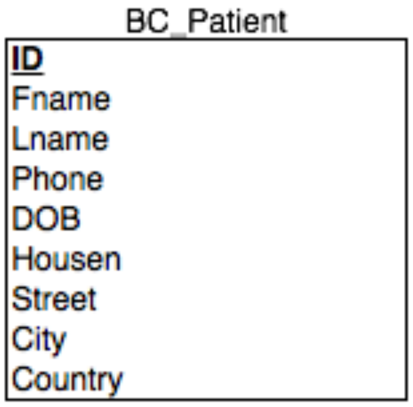
Assumptions:

- Emergency doctor is only assigned when treatment starts and patient can only be treated by one doctor at a time.
- Address is a composite attribute.
- Allergies is a multivalued attribute linked to the patient.
- Each prescription is related to each treatment because patients can have many appointments resulting in different prescriptions.
- In one appointment/treatment a doctor can prescribe one prescription with as many medications as needed with its own quantity and period of days.
- Age is derived of the date of birth attribute.
- Symptom is an attribute of treatment and is multivalued because a patient can have many symptoms.

Logical:

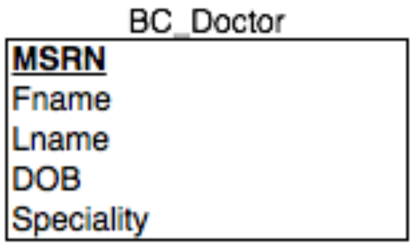
Transform the ER Model to Relational Model:

Step 1: Mapping of Regular Entity Types
▶For each regular (strong) entity type, create a relation R that includes all the simple attributes of E



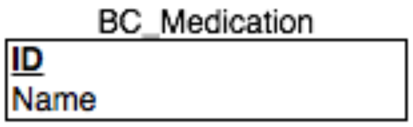
BC_Patient

ID
Fname
Lname
Phone
DOB
Housen
Street
City
Country



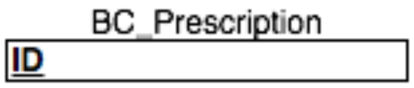
BC_Doctor

MSRN
Fname
Lname
DOB
Speciality



BC_Medication

ID
Name



BC_Prescription

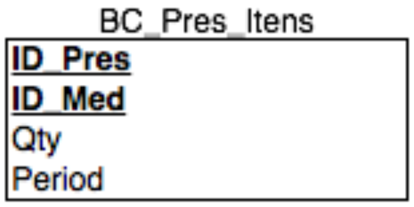
ID

Step 5: Mapping of Binary M:N Relationship Types ▶For each binary M:N relationship type

- Create a new relation

▶Include in the new relation as foreign keys, the primary

keys of the two participating entities.



BC_Pres_Itens

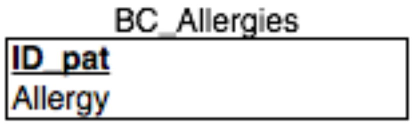
ID_Pres
ID_Med
Qty
Period

▶ Step 6: Mapping of Multivalued Attributes ▶For each multivalued attribute

- Create a new relation
- Primary key of R is the combination of A

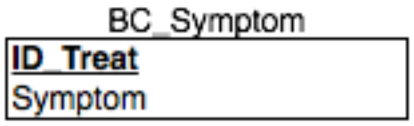
(attribute) and K(key)

- If the multivalued attribute is composite, include its simple components



BC_Allergies

ID_pat
Allergy



BC_Symptom

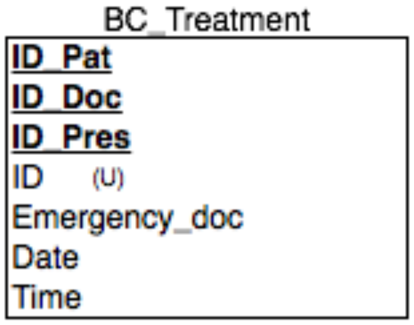
ID_Treat
Symptom

▶Step 7: Mapping of N-ary Relationship Types

▶For each n-ary relationship type R

- Create a new relation S to represent R
- Include primary keys of participating entity

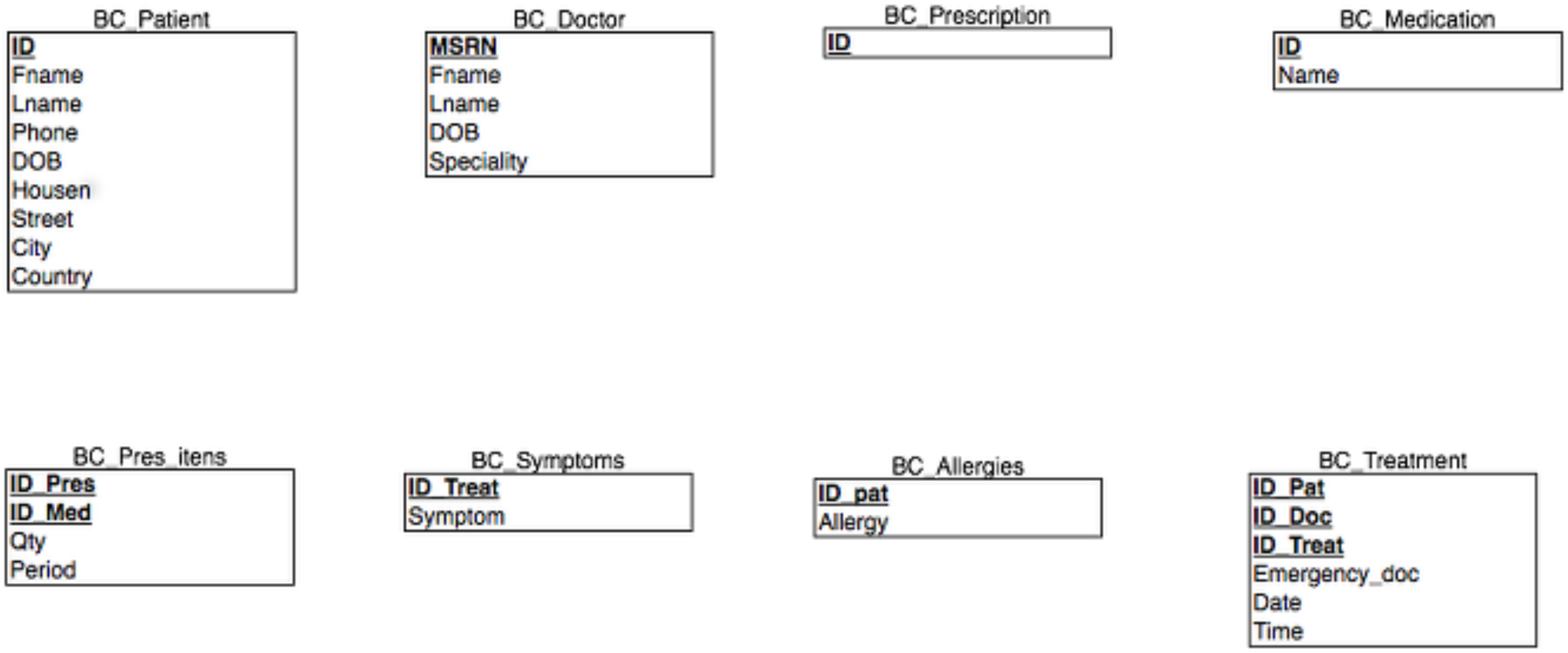
types as foreign keys
- Include any simple attributes as attributes



BC_Treatment

ID_Pat
ID_Doc
ID_Pres
ID (U)
Emergency_doc
Date
Time

Relational Model



BC_Patient
ID
Fname
Lname
Phone
DOB
Housen
Street
City
Country

BC_Doctor
MSRN
Fname
Lname
DOB
Speciality

BC_Prescription
ID

BC_Medication
ID
Name

BC_Pres_itens
ID_Pres
ID_Med
Qty
Period

BC_Symptoms
ID_Treat
Symptom

BC_Allergies
ID_pat
Allergy

BC_Treatment
ID_Pat
ID_Doc
ID_Treat
Emergency_doc
Date
Time

Normalisation

All Model were validated using the three Normal Forms of Normalisation.

1st Normal Form

All atomic data items, no repeating groups and a designed primary key (no duplicated rows).

2nd Formal Form (composite keys)

No attribute depends on a portion of primary key.

3rd Normal Form

Attributes are functionally dependent on a monkey attribute.

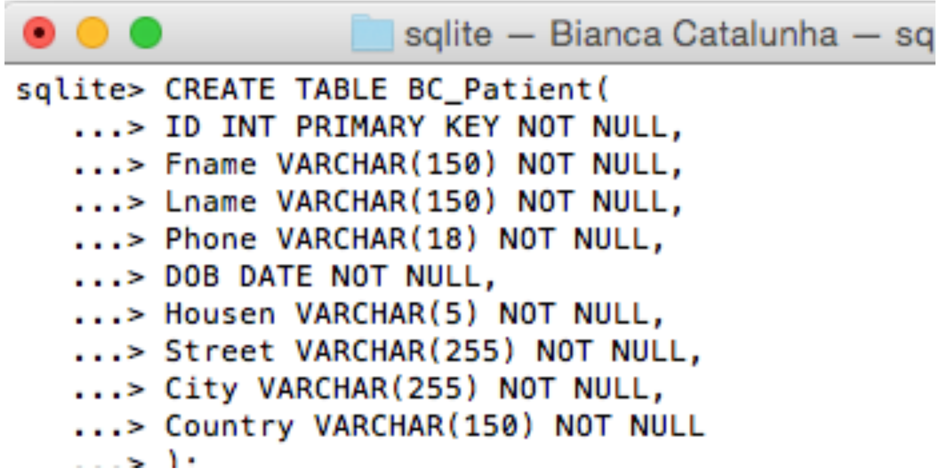
Data Dictionary

Table 1

Table name	Attribute name	Contents	Type	Formact	Required	PK or FK	FK referenced table	
BC_Patient	ID	Unique Identifier	INT	XXXX	TRUE	PK		
	Fname	First Name	VARCHAR	xxxx	TRUE			
	Lname	Last Name	VARCHAR	xxxx	TRUE			
	Phone	Contact number	VARCHAR	xxxx	TRUE			
	DOB	Date of Birth	DATE	YEAR/ MONTH/DAY	TRUE			
	Housen	House number	VARCHAR	XXXX	TRUE			
	Street	Street Name	VARCHAR	xxxx	TRUE			
	City	City Name	VARCHAR	xxxx	TRUE			
	Country	Country Name	VARCHAR	xxxx	TRUE			
BC_Doctor	MRSN	Medical Society Registration Number	INT	XXXX	TRUE	PK		
	Fname	First Name	VARCHAR	xxxx	TRUE			
	Lname	Last Name	VARCHAR	xxxx	TRUE			
	DOB	Date of Birth	DATE	xxxx	TRUE			
	Speciality	Doctor's Speciality	VARCHAR	xxxx	TRUE			
BC_Medication	ID	Unique Identifier	INT	XXXX	TRUE	PK		
	Name	Medication Name	VARCHAR	xxxx	TRUE			
BC_Prescription	ID	Unique Identifier	INT	XXXX	TRUE	PK		
BC_Pres_Itens	ID_Pres	Prescription's Unique Identifier	INT	XXXX	TRUE	FK,PK	BC_Prescription	
	ID_Med	Medication's Unique Identifier	INT	xxxx	TRUE	FK,PK	BC_Medication	

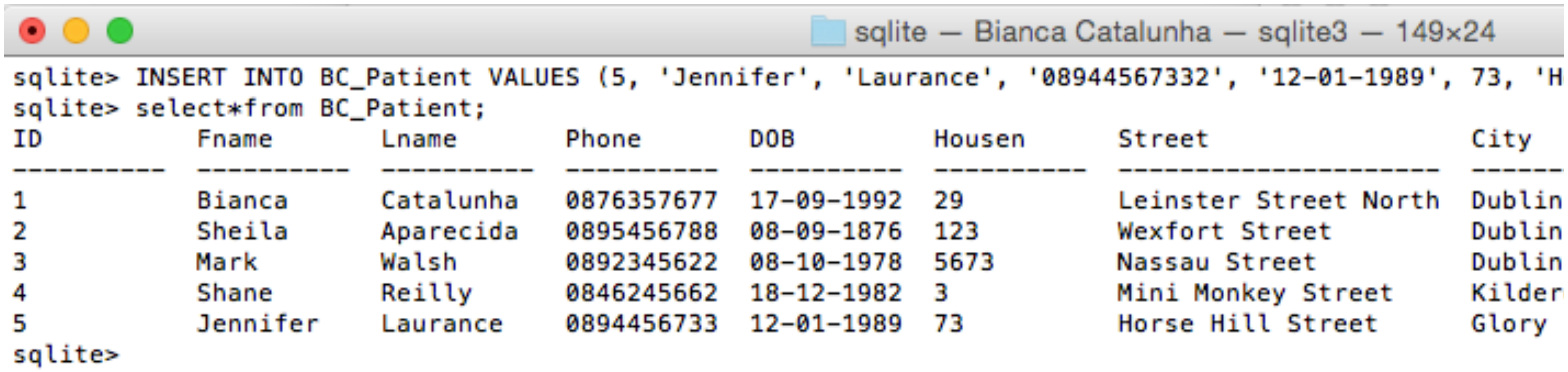
Implementation:
Inserts:

BC_Patient



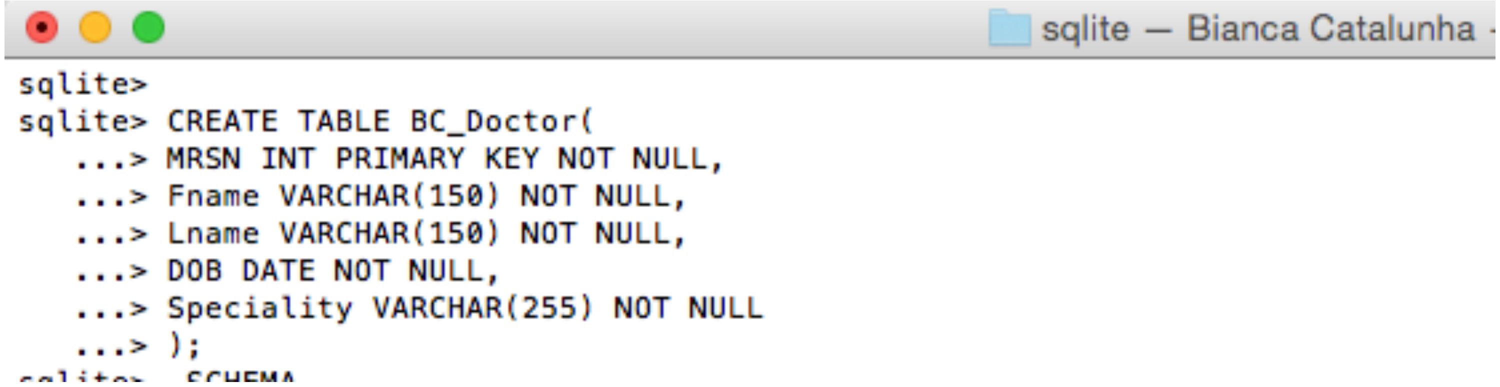
```
sqlite> CREATE TABLE BC_Patient(
   ...> ID INT PRIMARY KEY NOT NULL,
   ...> Fname VARCHAR(150) NOT NULL,
   ...> Lname VARCHAR(150) NOT NULL,
   ...> Phone VARCHAR(18) NOT NULL,
   ...> DOB DATE NOT NULL,
   ...> Housen VARCHAR(5) NOT NULL,
   ...> Street VARCHAR(255) NOT NULL,
   ...> City VARCHAR(255) NOT NULL,
   ...> Country VARCHAR(150) NOT NULL
   ...> );
```

```
INSERT INTO BC_Patient VALUES (00001, 'Bianca', 'Catalunha', '0876357677', '17-09-1992', 29,
'Leinster Street North', 'Dublin', 'Ireland');
INSERT INTO BC_Patient VALUES (2, 'Sheila', 'Aparecida', '0895456788', '08-09-1876', 123,
'Wexfort Street', 'Dublin', 'Ireland');
INSERT INTO BC_Patient VALUES (3, 'Mark', 'Walsh', '0892345622', '08-10-1978', 5673, 'Nassau
Street', 'Dublin', 'Ireland');
INSERT INTO BC_Patient VALUES (4, 'Shane', 'Reilly', '0846245662', '18-12-1982', 3, 'Mini Monkey
Street', 'Kildere', 'Ireland');
INSERT INTO BC_Patient VALUES (5, 'Jennifer', 'Laurance', '08944567332', '12-01-1989', 73, 'Horse
Hill Street', 'Glory', 'Ireland');
```



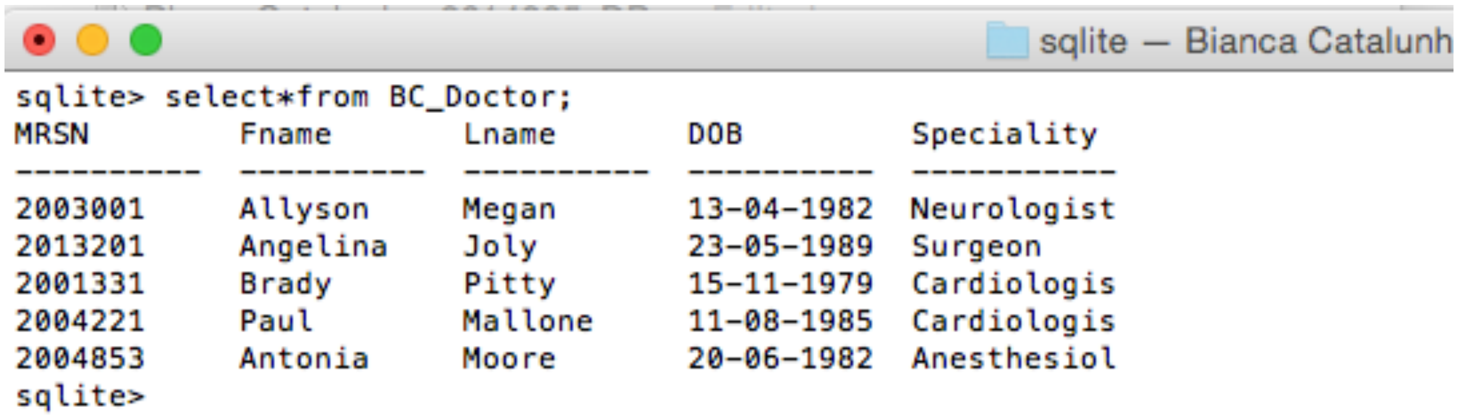
```
sqlite> INSERT INTO BC_Patient VALUES (5, 'Jennifer', 'Laurance', '08944567332', '12-01-1989', 73, 'H
sqlite> select*from BC_Patient;
ID          Fname       Lname       Phone       DOB         Housen      Street                 City
----------  ----------  ----------  ----------  ----------  ----------  ---------------------  ------
1           Bianca      Catalunha   0876357677  17-09-1992  29          Leinster Street North  Dublin
2           Sheila      Aparecida   0895456788  08-09-1876  123         Wexfort Street         Dublin
3           Mark        Walsh       0892345622  08-10-1978  5673        Nassau Street          Dublin
4           Shane       Reilly      0846245662  18-12-1982  3           Mini Monkey Street     Kilder
5           Jennifer    Laurance    0894456733  12-01-1989  73          Horse Hill Street      Glory
sqlite>
```

BC_Doctor



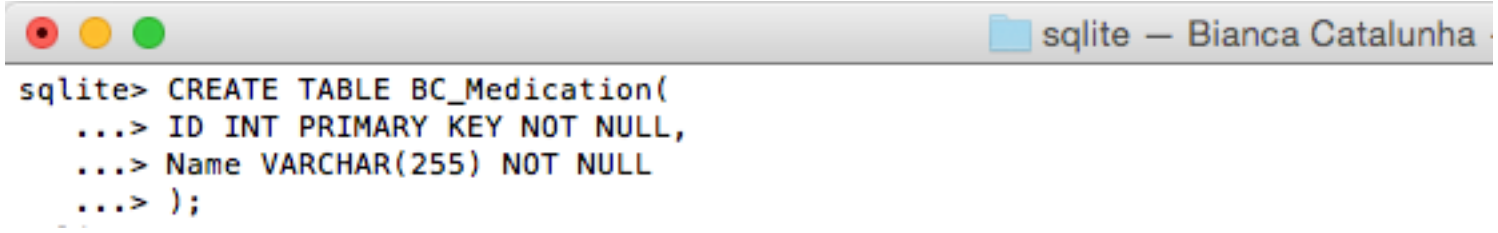
```
sqlite>
sqlite> CREATE TABLE BC_Doctor(
   ...> MRSN INT PRIMARY KEY NOT NULL,
   ...> Fname VARCHAR(150) NOT NULL,
   ...> Lname VARCHAR(150) NOT NULL,
   ...> DOB DATE NOT NULL,
   ...> Speciality VARCHAR(255) NOT NULL
   ...> );
sqlite>  SCHEMA
```

```
INSERT INTO BC_Doctor VALUES (2003001, 'Allyson', 'Megan', '13-04-1982', 'Neurologist');
INSERT INTO BC_Doctor VALUES (2013201, 'Angelina', 'Joly', '23-05-1989', 'Surgeon');
INSERT INTO BC_Doctor VALUES (2001331, 'Brady', 'Pitty', '15-11-1979', 'Cardiologist');
INSERT INTO BC_Doctor VALUES (2004221, 'Paul', 'Mallone', '11-08-1985', 'Cardiologist');
INSERT INTO BC_Doctor VALUES (2004853, 'Antonia', 'Moore', '20-06-1982',
'Anesthesiologist');
```



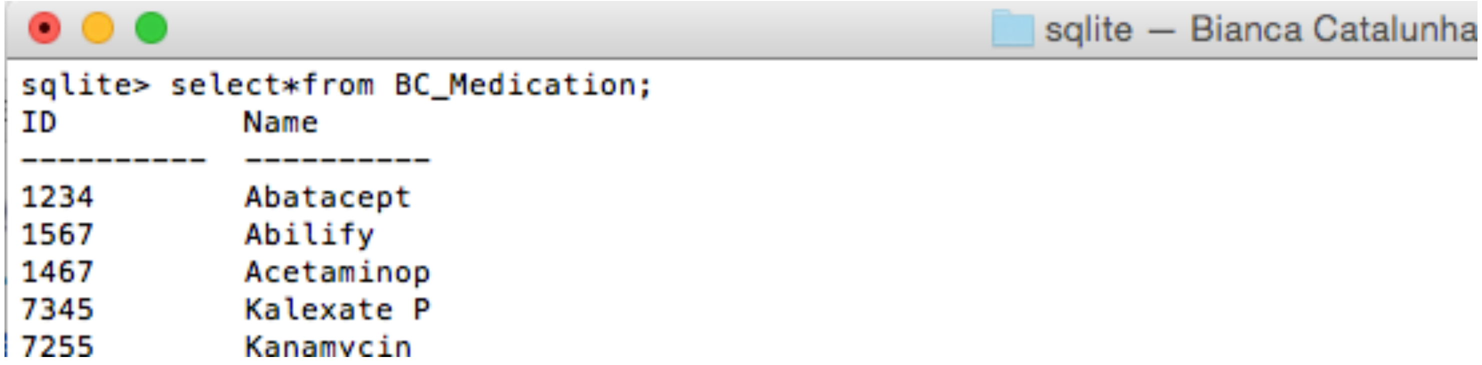
```
sqlite> select*from BC_Doctor;
MRSN        Fname       Lname       DOB         Speciality
----------  ----------  ----------  ----------  -----------
2003001     Allyson     Megan       13-04-1982  Neurologist
2013201     Angelina    Joly        23-05-1989  Surgeon
2001331     Brady       Pitty       15-11-1979  Cardiologis
2004221     Paul        Mallone     11-08-1985  Cardiologis
2004853     Antonia     Moore       20-06-1982  Anesthesiol
sqlite>
```

BC_Medication



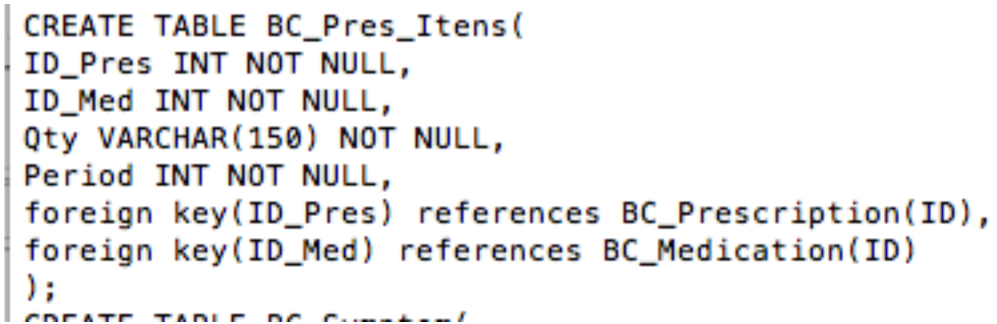
```
sqlite> CREATE TABLE BC_Medication(
   ...> ID INT PRIMARY KEY NOT NULL,
   ...> Name VARCHAR(255) NOT NULL
   ...> );
```

```
INSERT INTO BC_Medication VALUES (1234,'Abatacept');
INSERT INTO BC_Medication VALUES (1567,'Abilify');
INSERT INTO BC_Medication VALUES (1467,'Acetaminophen');
INSERT INTO BC_Medication VALUES (7345,'Kalexate Powder');
INSERT INTO BC_Medication VALUES (7255,'Kanamycin');
```



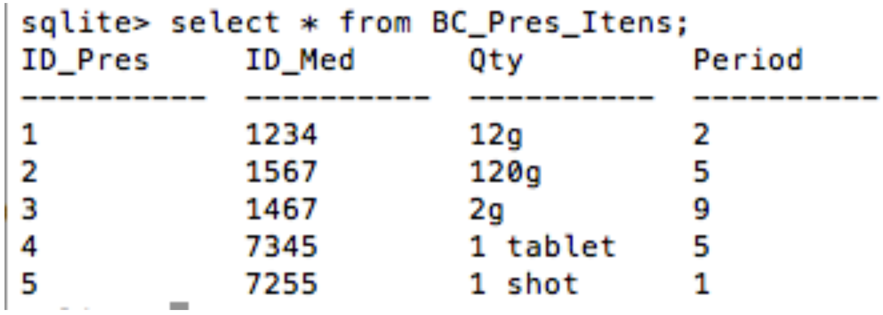
```
sqlite> select*from BC_Medication;
ID          Name
----------  ----------
1234        Abatacept
1567        Abilify
1467        Acetaminop
7345        Kalexate P
7255        Kanamycin
```

BC_Pres_Itens



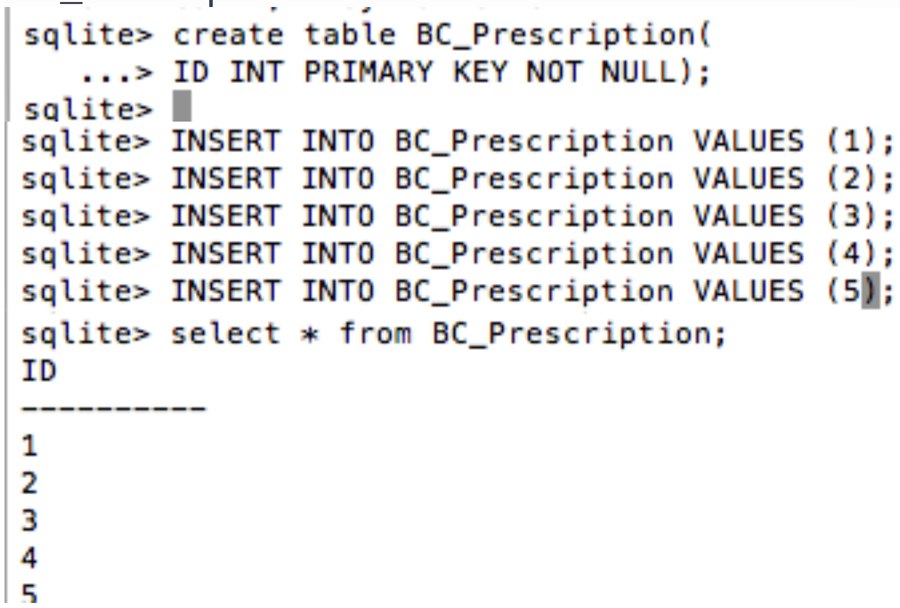
```
CREATE TABLE BC_Pres_Itens(
ID_Pres INT NOT NULL,
ID_Med INT NOT NULL,
Qty VARCHAR(150) NOT NULL,
Period INT NOT NULL,
foreign key(ID_Pres) references BC_Prescription(ID),
foreign key(ID_Med) references BC_Medication(ID)
);
```

```
INSERT INTO BC_Pres_Itens VALUES (1, 1234, '12g', 2);
INSERT INTO BC_Pres_Itens VALUES (2, 1567, '120g', 5);
INSERT INTO BC_Pres_Itens VALUES (3, 1467, '2g', 9);
INSERT INTO BC_Pres_Itens VALUES (4, 7345, '1 tablet', 5);
INSERT INTO BC_Pres_Itens VALUES (5, 7255, '1 shot', 1);
```



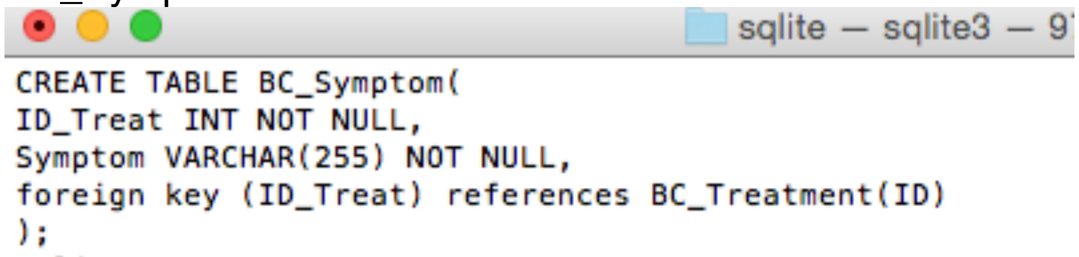
```
sqlite> select * from BC_Pres_Itens;
ID_Pres     ID_Med      Qty         Period
----------  ----------  ----------  ----------
1           1234        12g         2
2           1567        120g        5
3           1467        2g          9
4           7345        1 tablet    5
5           7255        1 shot      1
```

BC_Prescription



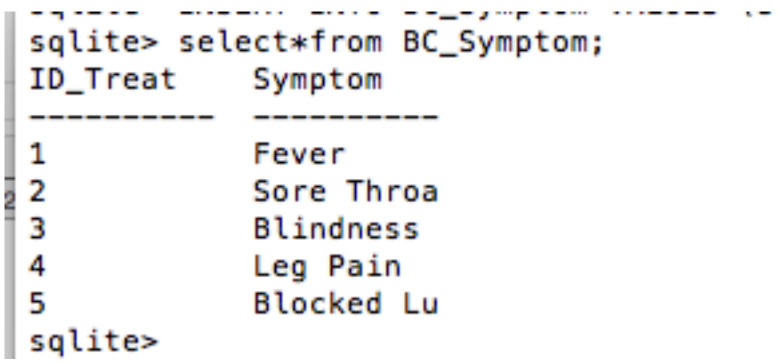
```
sqlite> create table BC_Prescription(
   ...> ID INT PRIMARY KEY NOT NULL);
sqlite> 
sqlite> INSERT INTO BC_Prescription VALUES (1);
sqlite> INSERT INTO BC_Prescription VALUES (2);
sqlite> INSERT INTO BC_Prescription VALUES (3);
sqlite> INSERT INTO BC_Prescription VALUES (4);
sqlite> INSERT INTO BC_Prescription VALUES (5);
sqlite> select * from BC_Prescription;
ID
----------
1
2
3
4
5
```

BC_Symptom



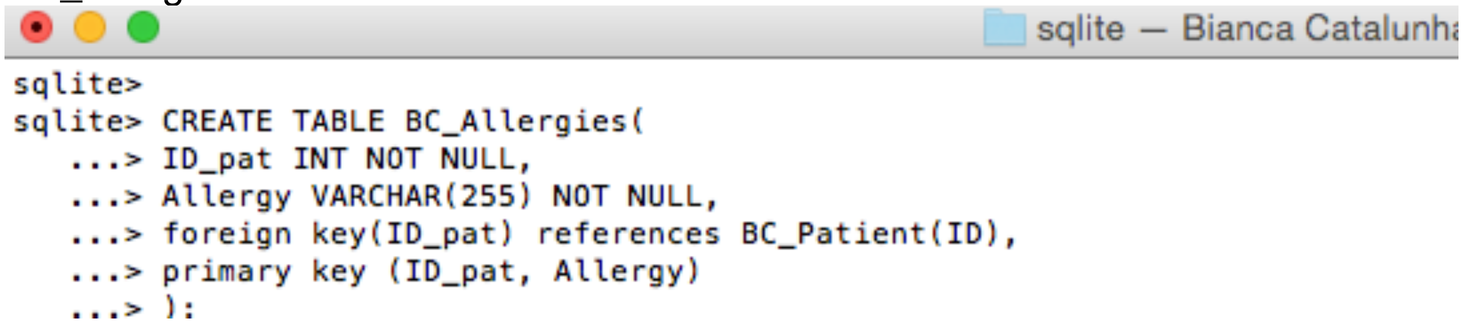
```
CREATE TABLE BC_Symptom(
ID_Treat INT NOT NULL,
Symptom VARCHAR(255) NOT NULL,
foreign key (ID_Treat) references BC_Treatment(ID)
);
```

```
INSERT INTO BC_Symptom VALUES (1, 'Fever');
INSERT INTO BC_Symptom VALUES (1, 'Sore Throat');
INSERT INTO BC_Symptom VALUES (2, 'Blindness');
INSERT INTO BC_Symptom VALUES (3, 'Leg Pain');
INSERT INTO BC_Symptom VALUES (4, 'Blocked Lungs');
```



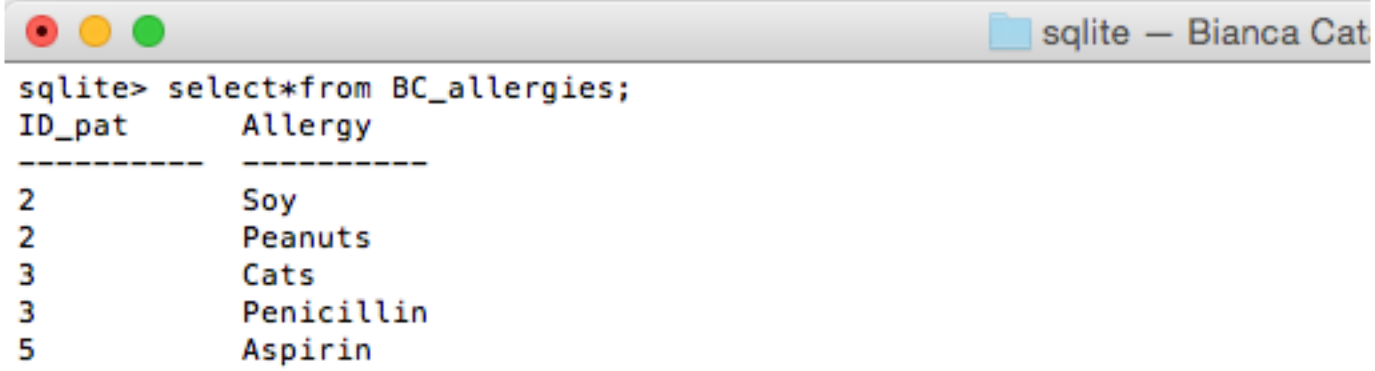
```
sqlite> select*from BC_Symptom;
ID_Treat    Symptom
----------  ----------
1           Fever
2           Sore Throa
3           Blindness
4           Leg Pain
5           Blocked Lu
sqlite>
```

BC_Allergies



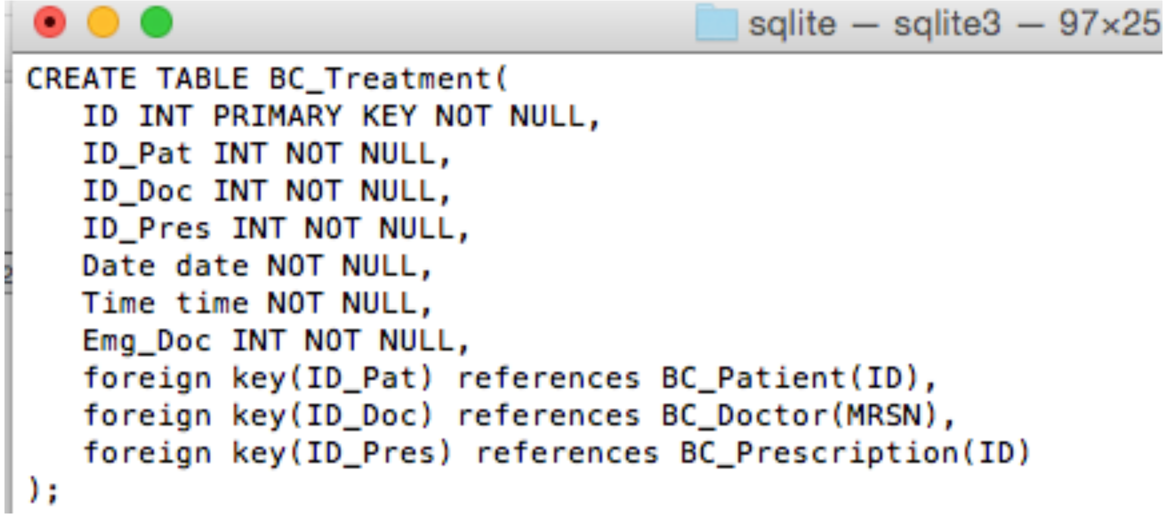
```
sqlite>
sqlite> CREATE TABLE BC_Allergies(
   ...> ID_pat INT NOT NULL,
   ...> Allergy VARCHAR(255) NOT NULL,
   ...> foreign key(ID_pat) references BC_Patient(ID),
   ...> primary key (ID_pat, Allergy)
   ...> );
```

```
INSERT INTO BC_Allergies VALUES (2,'Soy');
INSERT INTO BC_Allergies VALUES (2,'Peanuts');
INSERT INTO BC_Allergies VALUES (3,'Cats');
INSERT INTO BC_Allergies VALUES (3,'Penicillin');
INSERT INTO BC_Allergies VALUES (5,'Aspirin');
```



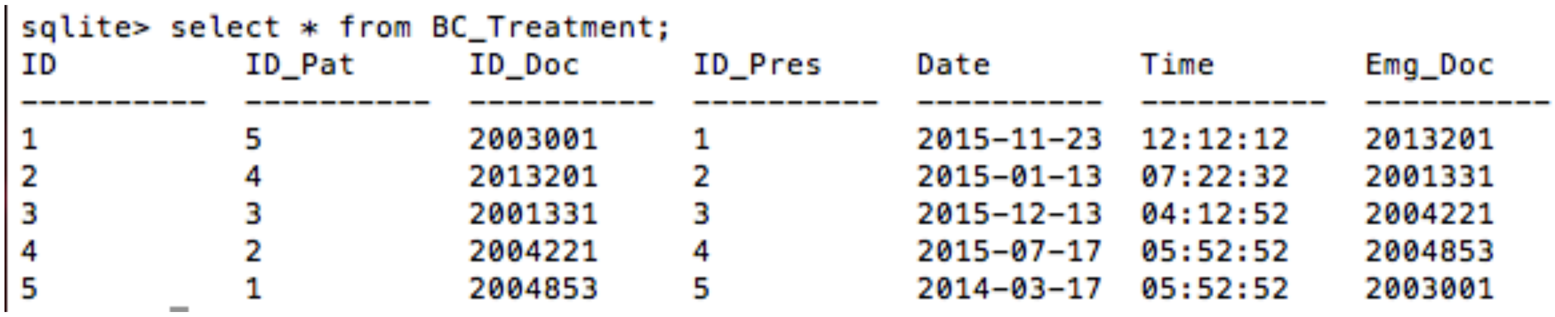
```
sqlite> select*from BC_allergies;
ID_pat      Allergy
----------  ----------
2           Soy
2           Peanuts
3           Cats
3           Penicillin
5           Aspirin
```

BC_Treatment



```
CREATE TABLE BC_Treatment(
   ID INT PRIMARY KEY NOT NULL,
   ID_Pat INT NOT NULL,
   ID_Doc INT NOT NULL,
   ID_Pres INT NOT NULL,
   Date date NOT NULL,
   Time time NOT NULL,
   Emg_Doc INT NOT NULL,
   foreign key(ID_Pat) references BC_Patient(ID),
   foreign key(ID_Doc) references BC_Doctor(MRSN),
   foreign key(ID_Pres) references BC_Prescription(ID)
);
```

```
INSERT INTO BC_Treatment VALUES (1,5,2003001,1, '2015-11-23', '12:12:12', 2013201);
INSERT INTO BC_Treatment VALUES (2,4,2013201,2, '2015-01-13', '07:22:32', 2001331);
INSERT INTO BC_Treatment VALUES (3,3,2001331,3, '2015-12-13', '04:12:52', 2004221);
INSERT INTO BC_Treatment VALUES (4,2,2004221,4, '2015-07-17', '05:52:52', 2004853);
INSERT INTO BC_Treatment VALUES (5,1,2004853,5, '2014-03-17', '05:52:52', 2003001);
```



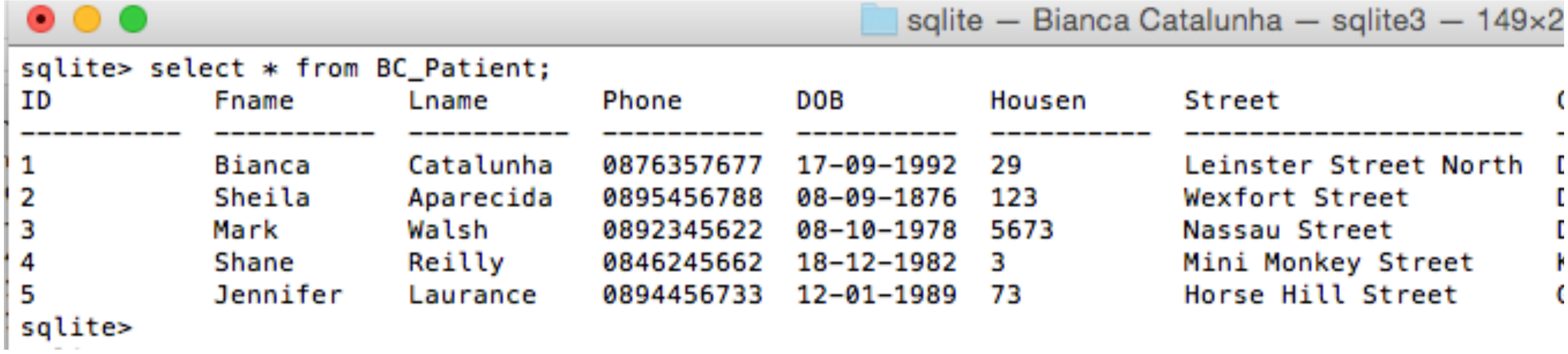
```
sqlite> select * from BC_Treatment;
ID          ID_Pat      ID_Doc      ID_Pres     Date        Time        Emg_Doc
----------  ----------  ----------  ----------  ----------  ----------  ----------
1           5           2003001     1           2015-11-23  12:12:12    2013201
2           4           2013201     2           2015-01-13  07:22:32    2001331
3           3           2001331     3           2015-12-13  04:12:52    2004221
4           2           2004221     4           2015-07-17  05:52:52    2004853
5           1           2004853     5           2014-03-17  05:52:52    2003001
```

Testing

Illustrate how your database has met its aforementioned requirements.

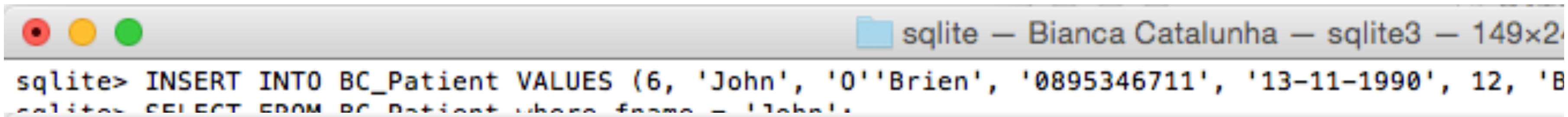
To do this you need to be able to write the SQL statements that will provide the following information.

A list of all patients at the surgery

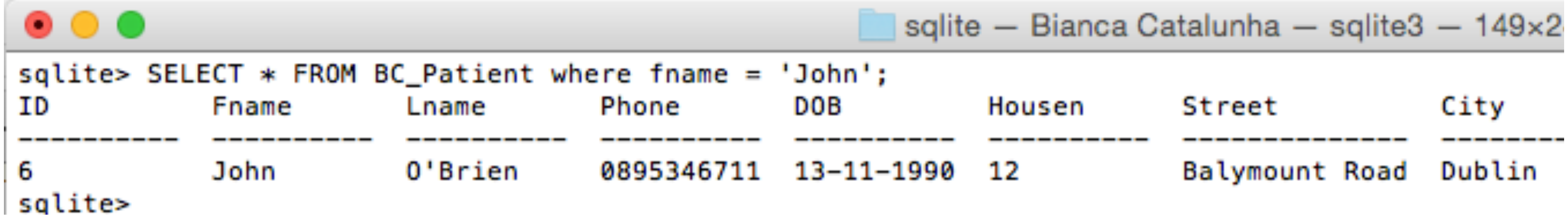


```
sqlite> select * from BC_Patient;
ID          Fname       Lname       Phone       DOB         Housen      Street
----------  ----------  ----------  ----------  ----------  ----------  ---------------------
1           Bianca      Catalunha   0876357677  17-09-1992  29          Leinster Street North
2           Sheila      Aparecida   0895456788  08-09-1876  123         Wexfort Street
3           Mark        Walsh       0892345622  08-10-1978  5673        Nassau Street
4           Shane       Reilly      0846245662  18-12-1982  3           Mini Monkey Street
5           Jennifer    Laurance    0894456733  12-01-1989  73          Horse Hill Street
sqlite>
```

Insert a new Patient into the Patients table whose name is John O'Brien. (Paste both the insert SQL and a select from the table)

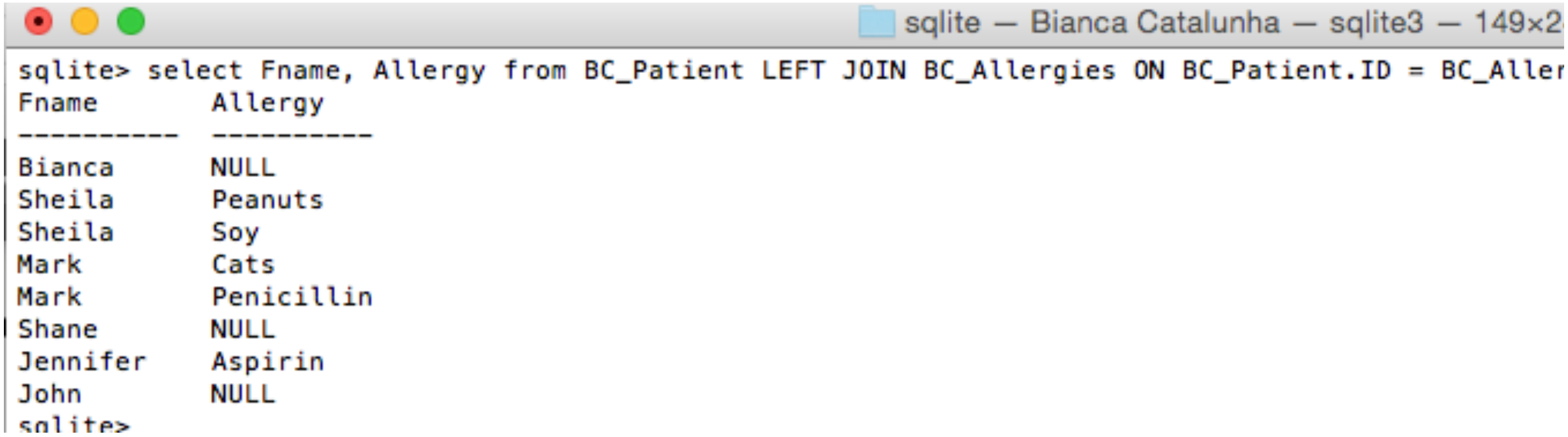


```
sqlite> INSERT INTO BC_Patient VALUES (6, 'John', 'O''Brien', '0895346711', '13-11-1990', 12, 'B
sqlite> SELECT FROM BC_Patient where fname = 'John';
```



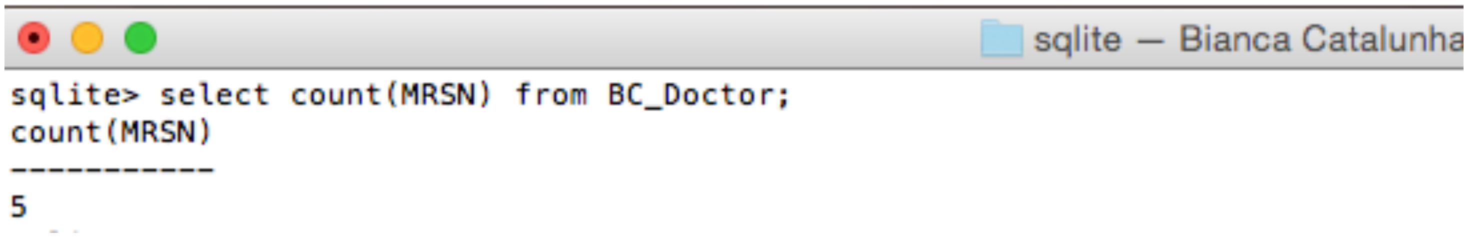
```
sqlite> SELECT * FROM BC_Patient where fname = 'John';
ID          Fname       Lname       Phone       DOB         Housen      Street          City
----------  ----------  ----------  ----------  ----------  ----------  --------------  --------
6           John        O'Brien     0895346711  13-11-1990  12          Balymount Road  Dublin
sqlite>
```

A list of all patients at the surgery and their allergies i.e. I want both the patient's name and the name of their allergy(ies)



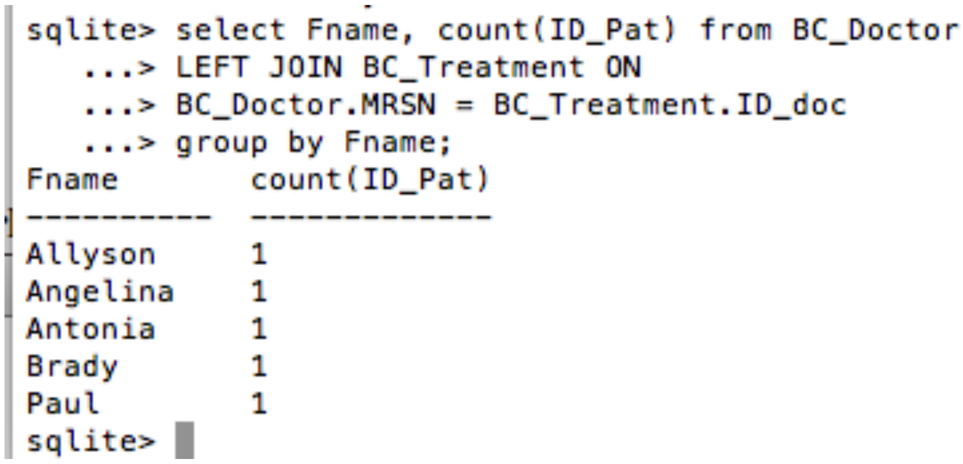
```
sqlite> select Fname, Allergy from BC_Patient LEFT JOIN BC_Allergies ON BC_Patient.ID = BC_Aller
Fname       Allergy
----------  ----------
Bianca      NULL
Sheila      Peanuts
Sheila      Soy
Mark        Cats
Mark        Penicillin
Shane       NULL
Jennifer    Aspirin
John        NULL
sqlite>
```

How many doctors work in the surgery?



```
sqlite> select count(MRSN) from BC_Doctor;
count(MRSN)
-----------
5
```

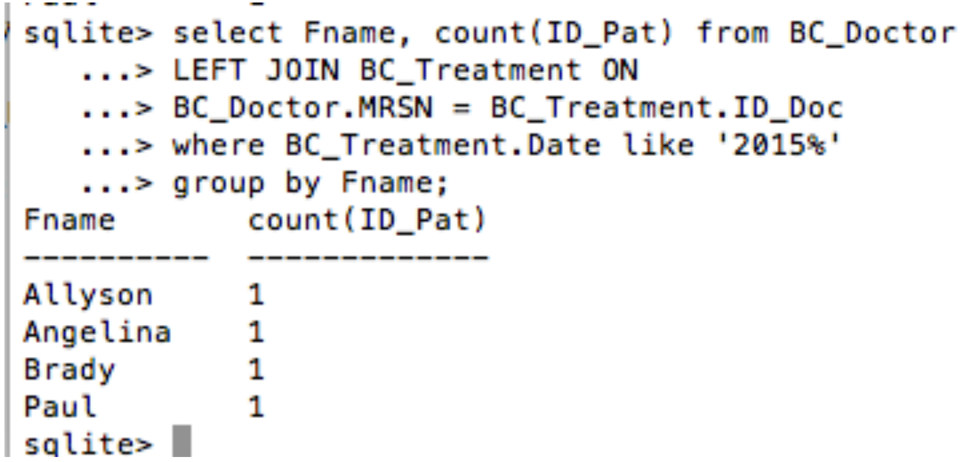
For each doctor, the number of patients they have treated.



```
sqlite> select Fname, count(ID_Pat) from BC_Doctor
   ...> LEFT JOIN BC_Treatment ON
   ...> BC_Doctor.MRSN = BC_Treatment.ID_doc
   ...> group by Fname;
Fname       count(ID_Pat)
----------  -------------
Allyson     1
Angelina    1
Antonia     1
Brady       1
Paul        1
sqlite>
```

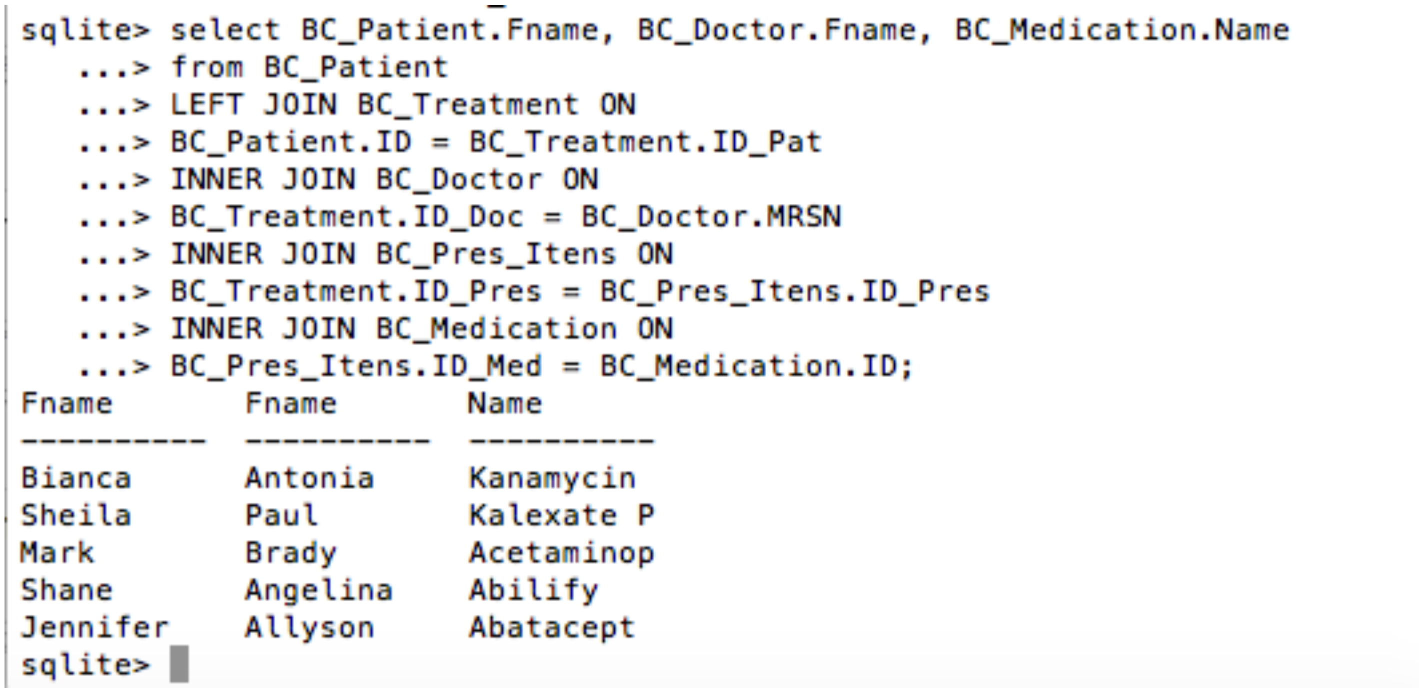
```
select Fname, count(ID_Pat) from BC_Doctor

    LEFT JOIN BC_Treatment ON
    BC_Doctor.MRSN = BC_Treatment.ID_doc
     group by Fname;
```

For each doctor, the number of patients they have treated this year. You will need to do some research into sqlite date functions



```
sqlite> select Fname, count(ID_Pat) from BC_Doctor
   ...> LEFT JOIN BC_Treatment ON
   ...> BC_Doctor.MRSN = BC_Treatment.ID_Doc
   ...> where BC_Treatment.Date like '2015%'
   ...> group by Fname;
Fname       count(ID_Pat)
----------  -------------
Allyson     1
Angelina    1
Brady       1
Paul        1
sqlite> 
```

The name of all patients, with the drug name and the doctor who prescribed the drug



```
sqlite> select BC_Patient.Fname, BC_Doctor.Fname, BC_Medication.Name
   ...> from BC_Patient
   ...> LEFT JOIN BC_Treatment ON
   ...> BC_Patient.ID = BC_Treatment.ID_Pat
   ...> INNER JOIN BC_Doctor ON
   ...> BC_Treatment.ID_Doc = BC_Doctor.MRSN
   ...> INNER JOIN BC_Pres_Itens ON
   ...> BC_Treatment.ID_Pres = BC_Pres_Itens.ID_Pres
   ...> INNER JOIN BC_Medication ON
   ...> BC_Pres_Itens.ID_Med = BC_Medication.ID;
Fname       Fname       Name
----------  ----------  ----------
Bianca      Antonia     Kanamycin
Sheila      Paul        Kalexate P
Mark        Brady       Acetaminop
Shane       Angelina    Abilify
Jennifer    Allyson     Abatacept
sqlite> 
```