IST 659: PROJECT FINAL REPORT:

HOSPITAL MANAGEMENT SYSTEM

Abstract: The project focuses on enhancing the hospital management by keeping in mind the patient allergies and previous medications. There are many hospital management systems currently being used in hospitals but this management system gives a way to minimize overdose and reduce complications to provide better healthcare for patients. It will give the doctors a better way to organize patient details and save them from untimely deaths due to negligence and carelessness. The project will also help in research for doctors as it will give reports where doctors can find the medicines that cannot be given to maximum patients and what allergies do maximum patients have.

Aditi Chawla

SUID: 214847310

Index

Project Summary 2

Entities and Attributes 3

Relationship Data Model-Visio 6

Relationship Data Model- Access..6

Relationship Data Model-SQL Server 7

Business Rules 8

Screenshots and Sql query for creating and inserting 9

Major Data questions with reports 18

Interfaces 26

Trigger…. 35

Project Summary:

Since a lot of people lose their lives every year, because of negligence, overdose and allergies, I find it necessary to have a hospital management system that takes care of all this data. The new solution being real time will have the latest information of the patient and will even provide recommendations of medications that might not suit the patient. The database will also store the allergies of various patients and avoid giving food to them that might have those substances. The database can also be further utilized for research in the field of medicine and healthcare because of the information it would have gathered over the years.

The doctors will have access to the patient data and the medications to make swift decisions when it comes to the health of the patient. The central function of the data is to help patients know their prescription and also know about their allergies to take better care of their health. The doctors, being the admin of the database will have access to all the data and can modify it too.

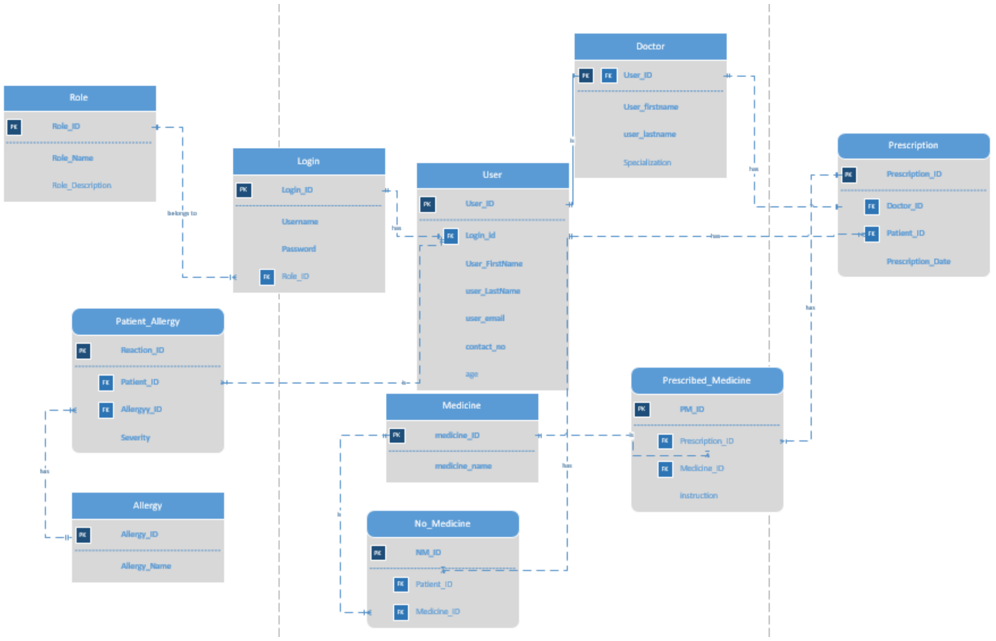
Entity & Attributes:

Data Objects: **Hospital Management System**	**This Database contains all the tables, their attributes and the relationship that build the hospital management system**
Role	Stores the details of the roles present in the application
- Role ID - Role Name - Role Description	Role ID: ID of the role Role Name: Either Doctor or Patient Primary Key- Each role has unique **Role ID,** so it is the primary key
User	Stores the details of the user
- User ID - Login_ID - User_FirstName - User_LastName - User_Email - ContactNo - Age	Primary Key: User ID will uniquely identify each user of the system. The 3 categories of users would be I. Doctor II. Patient Primary Key: Each user has a unique user_ID Foreign Key: Login ID is referenced from the login table
Login	Stores the login details of the users with the respective role details
- Login ID - User Name - Password - Role ID	Primary Key: Login ID is unique and hence the primary key Foreign Key: Role ID referenced from Roles table indicating whether the user is a doctor or a patient
Doctor	Stores the details of the doctors in the hospital
- User_ID - User_firstname - User_Lastname - Specialization **Prescription**	Primary Key and Foreign Key: User_ID is unique for each record and is being referenced from the user table
- Prescripion ID - Doctor ID - Patient ID - Prescription_Date	Stores the prescription details of the patient Primary Key: Prescription ID Foreign Key: i) Doctor ID referenced from User table indicating which doctor is giving the prescription

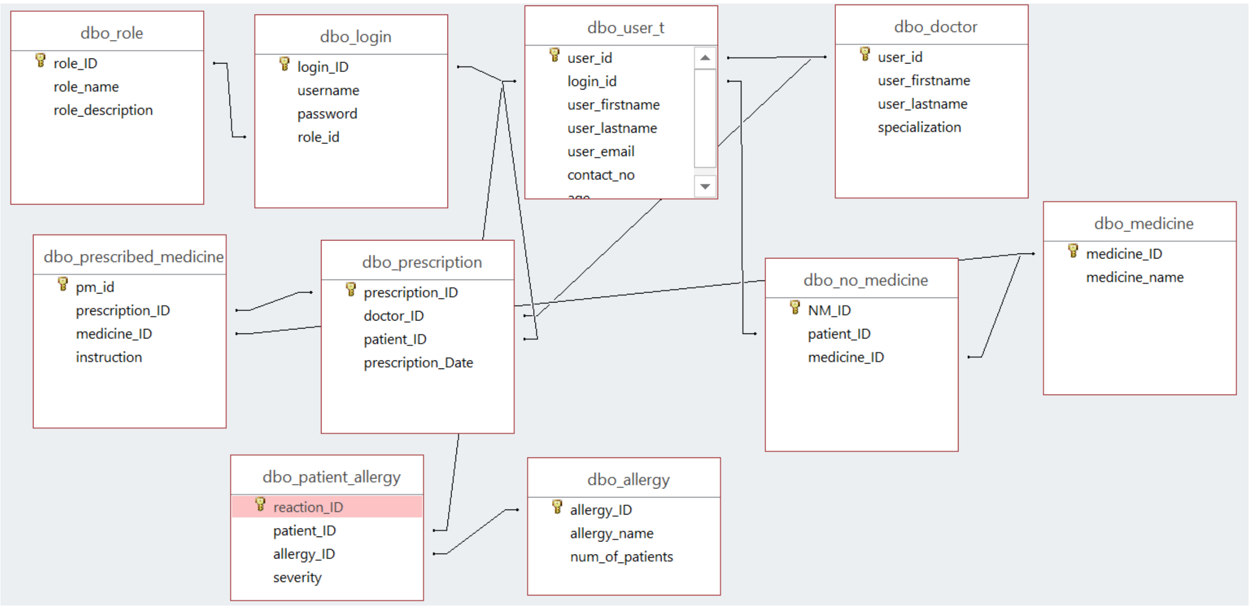
	ii) Patient ID referenced from Users table indicating the patient who gets the prescription
Prescribed_Medicine	Stores the room details of the prescribed_medicine
• PM_ID • Prescription_ID • Medicine_ID • Instruction	Primary Key: PM ID is unique for each prescribed medicine hence the primary key Foreign Key: Prescription_ID referenced from prescription table Medicine_ID being refenced form the medicine table
Medicine	Stores the medicine details
• Medicine ID • Medicine_name	Primary Key: Medicine ID is unique for the medicines and will unique identify each of them hence the primary key
No Medicine	Stores the no medicine details
• NM ID • Patient_ID • Medicine_ID	Primary_Key: NM_ID is unique for each record Foreign Key: Patient_ID being referenced from the user table Medicine_ID being referenced from the medicine table
Patient_Allergy	Allows to track the patients and their allergies
• Reaction_ID • Patient_ID • Allergy_ID • Severity	This table acts as an associative table for Patients and Allergies table having a many-to-many relationship Primary Key: Reaction_ID is an identifier uniquely identifying each work order assignment combination, hence primary key Foreign Key: i)Patient ID referenced from the Users table indicating what patients have allergies ii)AllergyID referenced from the Allergy table indicating the allergies that the patients have
Allergy	Stores the names of the various allergies
• Allergy_ID • Allergy Name	Primary Key –Allergy ID is the unique ID identifying the various allergies Allergy name gives the name of the various allergies

Relational Data Model:

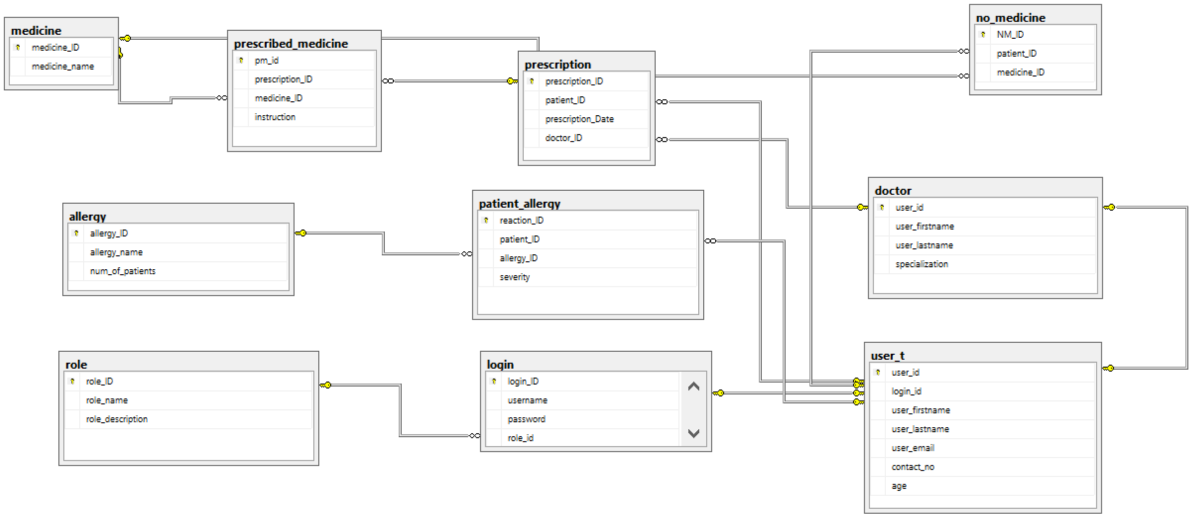
Visio ER Diagram:



Access Relationship Diagram:



SQL Server ER Diagram:



Business Rules:

There are 2 different user roles – Doctor and Patient, they will have separate flow in the database and are treated as separate entities

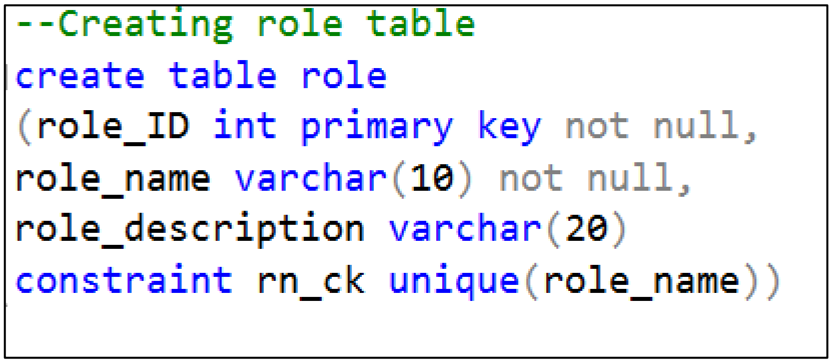
- Doctor is the admin of the system who can manage the medication and allergy details of the patients and track down the prescription along with the allergies and medicines that have been given to the patient and the medicines that cannot be given
- Patients are the users of the systems who have consulted the doctor. They can check their reports on allergies, no medicines and prescriptions.
- Each User has one and only one login id to the login page.
- Each user has exactly one assigned role; Each role must have atleast one user. Here I have assumed that a role has minimum one user.

Each user id from the user table has one to one relationship with the user_id in the doctor table.

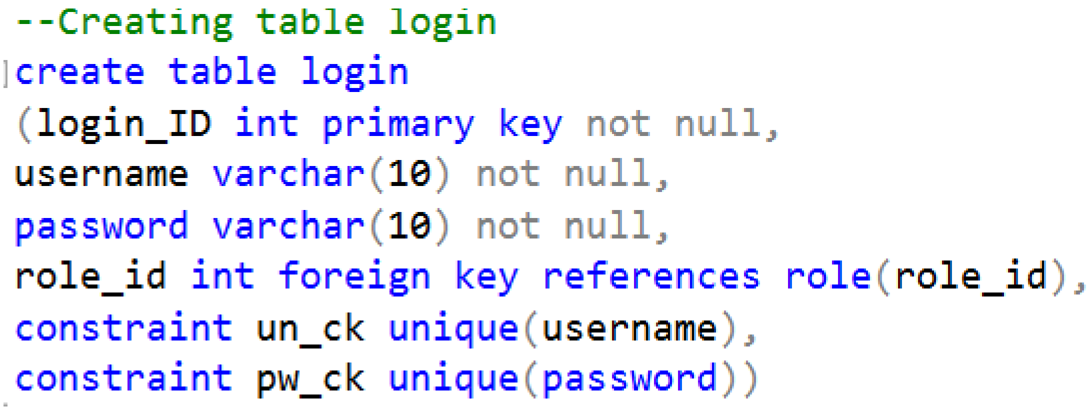
- Each user(doctor) gives one or more prescriptions; Each prescription is signed by one and only one doctor. Here I have assumed that a doctor must give at least one prescription
- Each patient may have several prescriptions; There is one and only one patient for each prescription at any given time.
- A prescription has at least one prescribed_medicine; a medicine may or may not be a part of several prescripions.
- No_medicine table lists the medicines that cannot be given to a patient.
- A medicine can have many no_medicines associated to it.
- A patient may or may not have no_medicines associated to it.
- A patient may have zero or multiple allergies.
- An allergy can be associated with zero or many patients.

Screenshots for creating tables

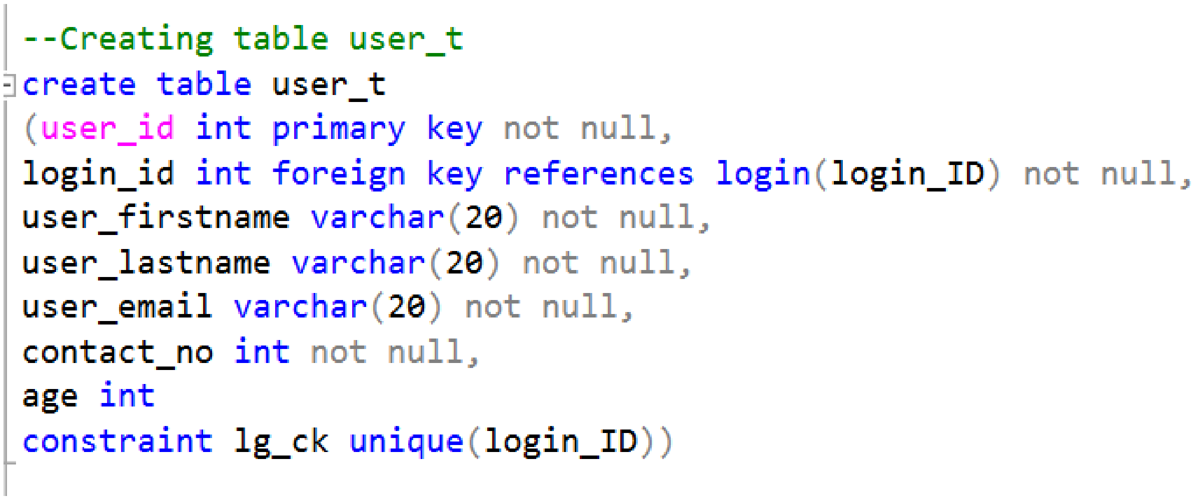
SQL Scripts for Creating and Inserting Sample Data



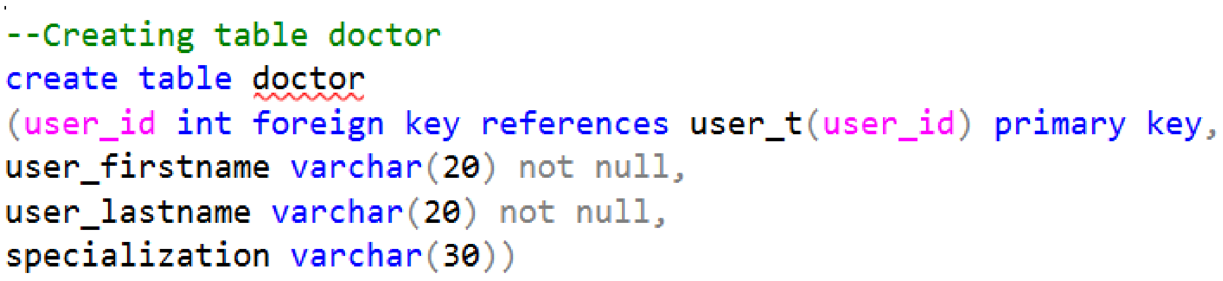
```
--Creating role table
create table role
(role_ID int primary key not null,
role_name varchar(10) not null,
role_description varchar(20)
constraint rn_ck unique(role_name))
```



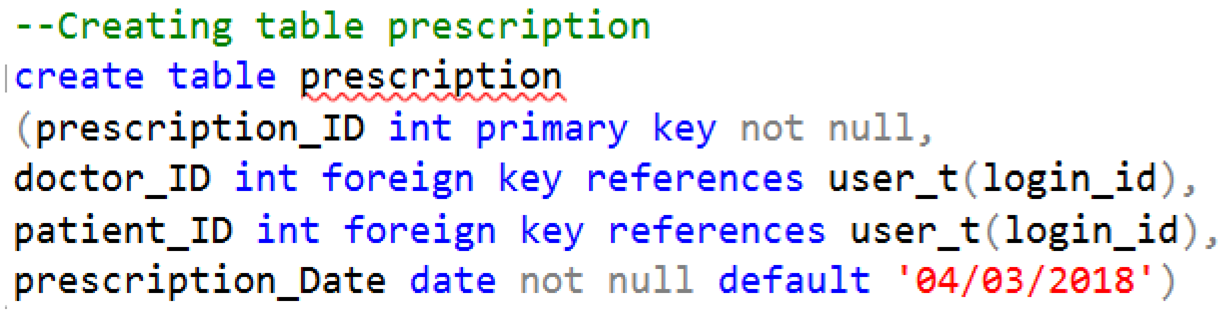
```
--Creating table login
create table login
(login_ID int primary key not null,
username varchar(10) not null,
password varchar(10) not null,
role_id int foreign key references role(role_id),
constraint un_ck unique(username),
constraint pw_ck unique(password))
```



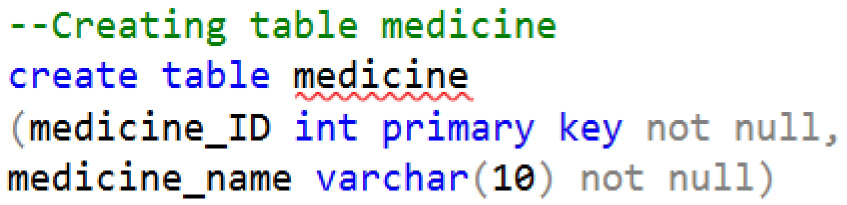
```
--Creating table user_t
create table user_t
(user_id int primary key not null,
login_id int foreign key references login(login_ID) not null,
user_firstname varchar(20) not null,
user_lastname varchar(20) not null,
user_email varchar(20) not null,
contact_no int not null,
age int
constraint lg_ck unique(login_ID))
```



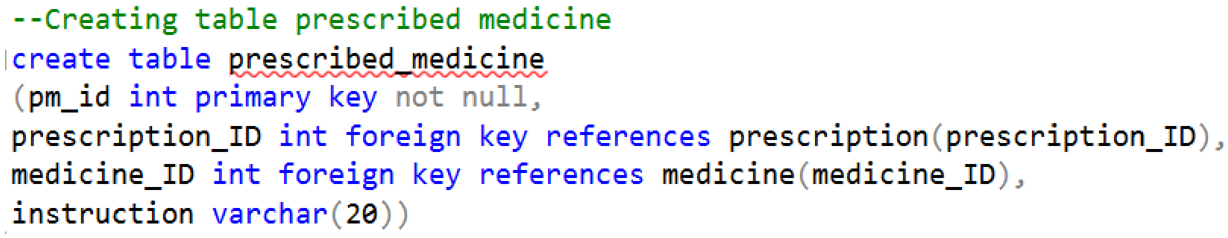
```
--Creating table doctor
create table doctor
(user_id int foreign key references user_t(user_id) primary key,
user_firstname varchar(20) not null,
user_lastname varchar(20) not null,
specialization varchar(30))
```



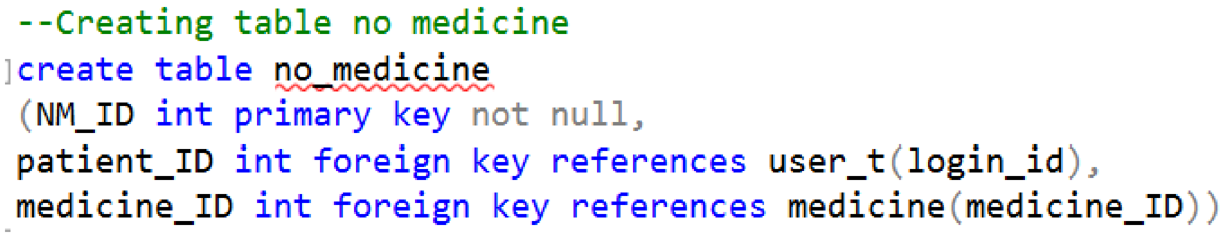
```
--Creating table prescription
create table prescription
(prescription_ID int primary key not null,
doctor_ID int foreign key references user_t(login_id),
patient_ID int foreign key references user_t(login_id),
prescription_Date date not null default '04/03/2018')
```



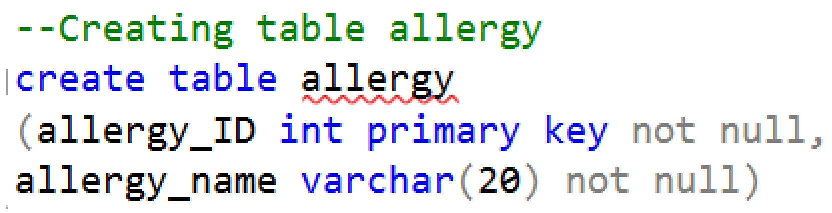
```
--Creating table medicine
create table medicine
(medicine_ID int primary key not null,
medicine_name varchar(10) not null)
```



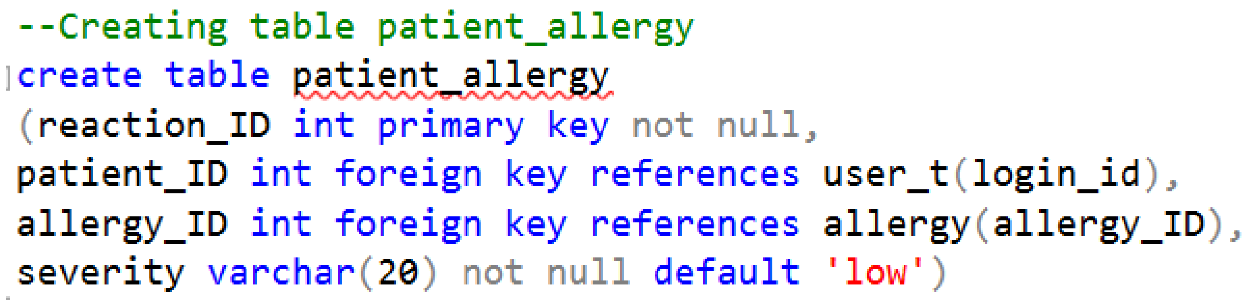
```
--Creating table prescribed medicine
create table prescribed_medicine
(pm_id int primary key not null,
prescription_ID int foreign key references prescription(prescription_ID),
medicine_ID int foreign key references medicine(medicine_ID),
instruction varchar(20))
```



```
--Creating table no medicine
create table no_medicine
(NM_ID int primary key not null,
patient_ID int foreign key references user_t(login_id),
medicine_ID int foreign key references medicine(medicine_ID))
```



```
--Creating table allergy
create table allergy
(allergy_ID int primary key not null,
allergy_name varchar(20) not null)
```



```
--Creating table patient_allergy
create table patient_allergy
(reaction_ID int primary key not null,
patient_ID int foreign key references user_t(login_id),
allergy_ID int foreign key references allergy(allergy_ID),
severity varchar(20) not null default 'low')
```

SQL Query for creating tables

CREATE: Role table

```
create table role
(role_ID int primary key not null,
role_name varchar(10) not null,
role_description varchar(20)
constraint rn_ck unique(role_name))
```

Create: Login table

```
create table login
(login_ID int primary key not null,
username varchar(10) not null,
password varchar(10) not null,
role_id int foreign key references role(role_id),
constraint un_ck unique(username),
constraint pw_ck unique(password))
```

Create: User_t table

```
create table user_t
(user_id int primary key not null,
login_id int foreign key references login(login_ID) not null,
user_firstname varchar(20) not null,
user_lastname varchar(20) not null,
user_email varchar(20) not null,
contact_no int not null,
age int
constraint lg_ck unique(login_ID))
```

Create: Doctor table

```
create table doctor
(user_id int foreign key references user_t(user_id) primary key,
user_firstname varchar(20) not null,
user_lastname varchar(20) not null,
specialization varchar(30))
```

Create: Prescription table

```
create table prescription
(prescription_ID int primary key not null,
doctor_ID int foreign key references user_t(login_id),
patient_ID int foreign key references user_t(login_id),
prescription_Date date not null default '04/03/2018')
```

Create: Medicine table

```
create table medicine
(medicine_ID int primary key not null,
medicine_name varchar(10) not null)
```

Create: Prescribed_Medicine

```
create table prescribed_medicine

prescription_ID int foreign key references prescription(prescription_ID),
medicine_ID int foreign key references medicine(medicine_ID),
instruction varchar(20))
```

Create: No Medicine table

```
create table no_medicine
(NM_ID int primary key not null,
patient_ID int foreign key references user_t(login_id),
medicine_ID int foreign key references medicine(medicine_ID))
```

Create: Allergy table

```
create table allergy
(allergy_ID int primary key not null,
allergy_name varchar(20) not null)
```

Create: Patient_Allergy table

```
create table patient_allergy
(reaction_ID int primary key not null,
patient_ID int foreign key references user_t(login_id),
allergy_ID int foreign key references allergy(allergy_ID),
severity varchar(20) not null default 'low')
```

Inserting into tables

Inserting into Role

```
insert into role(role_ID, role_name, role_description)
values(101, 'doctor', 'prescribes'),
(102, 'patient', 'gets prescribed')
```

Inserting into Login

```
insert into login(login_ID, username, password, role_ID)
values(201, 'adchawla', 'aaa', 101),
(202, 'nasaluja', 'bbb', 101),
(203,'anhanda', 'ccc', 102),
(204,'adchauhan', 'ddd', 102),
(205, 'prmatnani', 'eee', 102),
(206, 'arsuri', 'fff', 102),
(207, 'manav', 'mehta', 102)
```

Inserting into user_t

```
insert into user_t(user_ID,login_ID, user_firstname, user_lastname,
user_email,contact_no, age)
values(1, 201, 'aditi','chawla', 'adchawla@doc.com', 7402, 24),
(2,202, 'niti', 'saluja', 'nasaluja@doc.com', 4752, 28),
(3, 203, 'anmol', 'handa', 'anhanda@pat.com', 4949, 31),
(4, 204, 'aditya', 'chauhan', 'achauhan@pat.com', 4953, 5),
(5, 205, 'priya', 'matnani', 'prmatnani@pat.com', 2222, 29),
(6, 206, 'arjun', 'suri', 'arsuri@pat.com', 3333, 12)
```

Inserting into Doctor

```
insert into doctor(user_id, user_firstname, user_lastname, specialization)
values(1, 'aditi','chawla', 'psychiatrist'),
(2, 'niti', 'saluja', 'immunologist')
```

Inserting into Prescription

```
insert into prescription(prescription_ID, patient_ID, prescription_Date, doctor_ID)
values(301, 203, '04/05/2018',2),
(302,206, '05/06/2018',1)
```

Inserting into Medicine

```
insert into medicine(medicine_ID, medicine_name)
values
(403, 'Tylenol'),
(404, 'Benadryl'),
(405, 'Advil'),
(406, 'Codeine')
```

Inserting into Prescribed Medicine

```
insert into prescribed_medicine(pm_id, prescription_ID, medicine_ID, instruction)
values(501, 301, 402, 'after dinner'),
(502, 302, 401, 'before breakfast')
```

Inserting into No_medicine

```
insert into no_medicine(NM_ID, patient_ID, medicine_ID)
values(601, 204, 402),
(602, 206, 401)
```

Inserting into allergies

```
insert into allergy(allergy_ID, allergy_name)
values(701, 'tuna'),
(702, 'eggs'),
(703, 'peanuts')
```

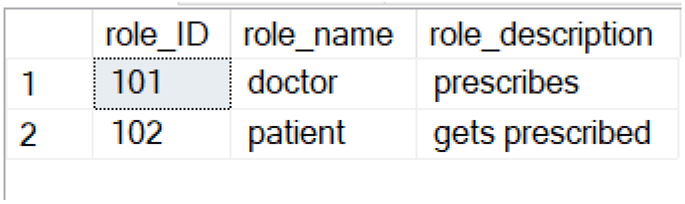
Inserting into Patient_allergy

```
insert into patient_allergy(reaction_ID, patient_ID,allergy_ID, severity)
values
(816, 209, 705, 'high'),
(801, 203, 701, 'medium'),
(802, 204, 702, 'high'),
(803, 207, 703, 'low'),
(804, 206, 705, 'low')
```

Displaying Data from Tables

Role Table

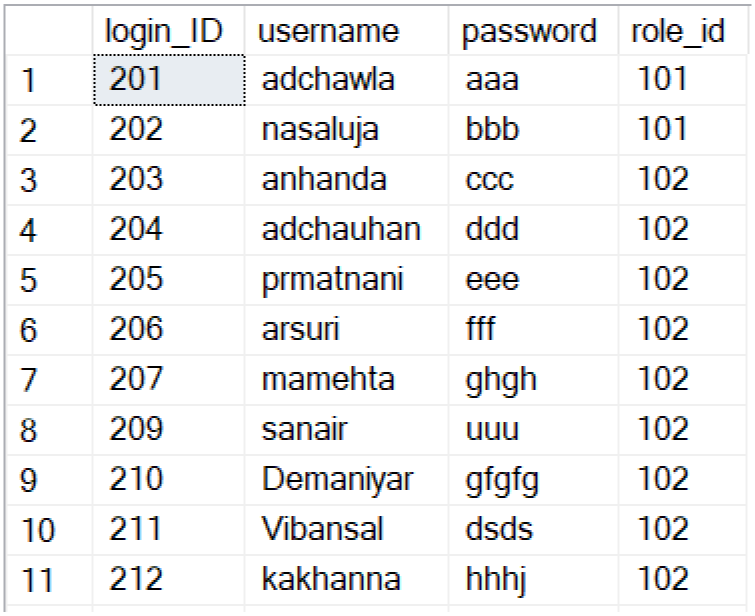
```
select * from role
```



	role_ID	role_name	role_description
1	101	doctor	prescribes
2	102	patient	gets prescribed

Login Table

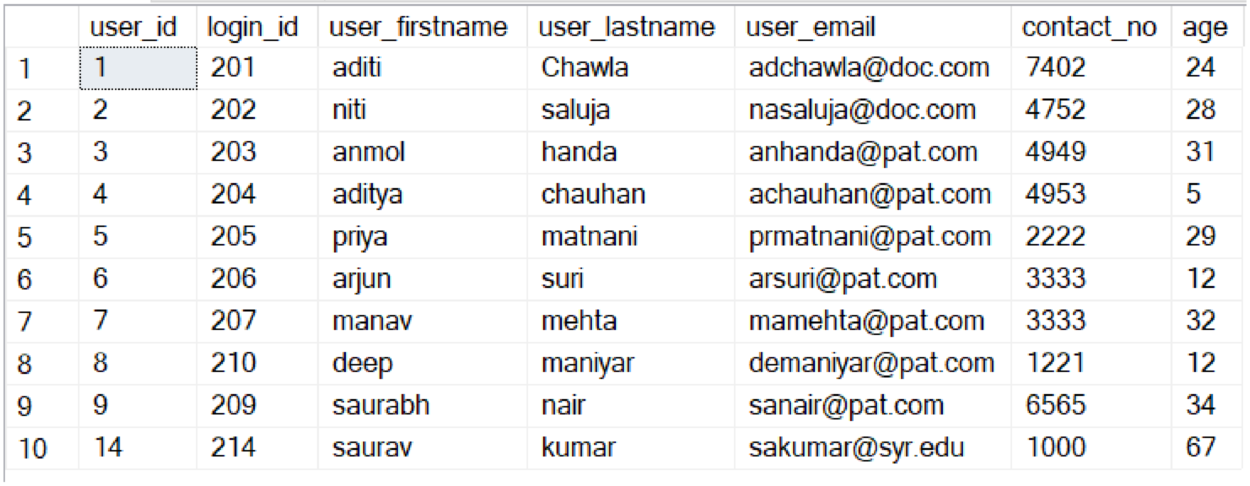
```
select * from login
```



	login_ID	username	password	role_id
1	201	adchawla	aaa	101
2	202	nasaluja	bbb	101
3	203	anhanda	ccc	102
4	204	adchauhan	ddd	102
5	205	prmatnani	eee	102
6	206	arsuri	fff	102
7	207	mamehta	ghgh	102
8	209	sanair	uuu	102
9	210	Demaniyar	gfgfg	102
10	211	Vibansal	dsds	102
11	212	kakhanna	hhhj	102

User_t Table

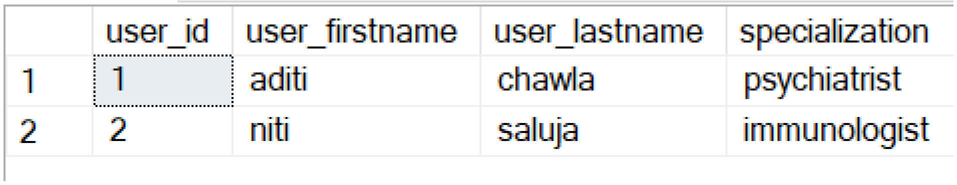
```
select * from user_t
```



	user_id	login_id	user_firstname	user_lastname	user_email	contact_no	age
1	1	201	aditi	Chawla	adchawla@doc.com	7402	24
2	2	202	niti	saluja	nasaluja@doc.com	4752	28
3	3	203	anmol	handa	anhanda@pat.com	4949	31
4	4	204	aditya	chauhan	achauhan@pat.com	4953	5
5	5	205	priya	matnani	prmatnani@pat.com	2222	29
6	6	206	arjun	suri	arsuri@pat.com	3333	12
7	7	207	manav	mehta	mamehta@pat.com	3333	32
8	8	210	deep	maniyar	demaniyar@pat.com	1221	12
9	9	209	saurabh	nair	sanair@pat.com	6565	34
10	14	214	saurav	kumar	sakumar@syr.edu	1000	67

Doctor Table

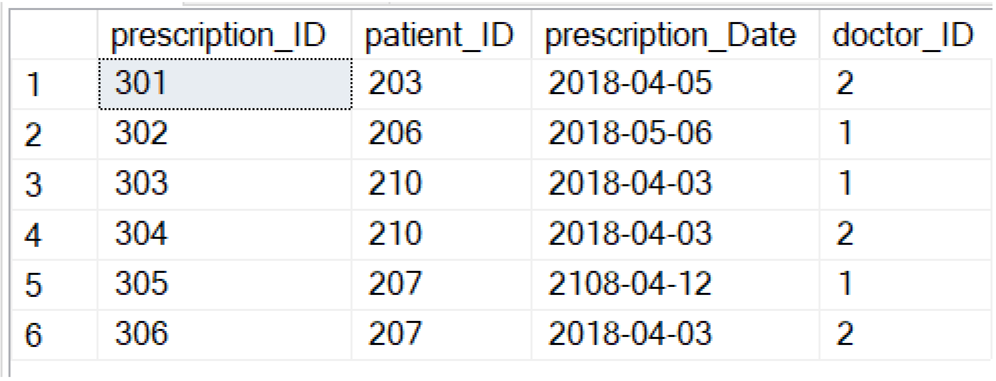
```
select * from doctor
```



	user_id	user_firstname	user_lastname	specialization
1	1	aditi	chawla	psychiatrist
2	2	niti	saluja	immunologist

Prescription Table

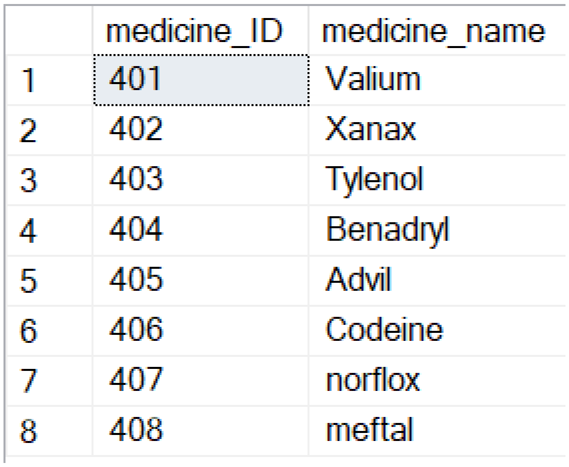
```
select * from prescription
```



	prescription_ID	patient_ID	prescription_Date	doctor_ID
1	301	203	2018-04-05	2
2	302	206	2018-05-06	1
3	303	210	2018-04-03	1
4	304	210	2018-04-03	2
5	305	207	2108-04-12	1
6	306	207	2018-04-03	2

Medicine Table

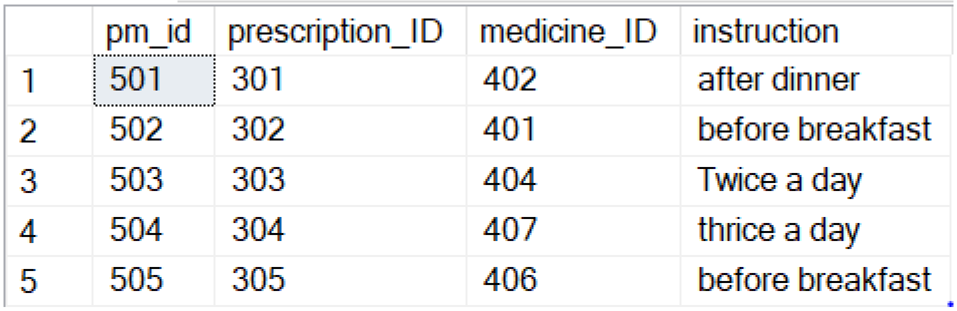
```
select * from medicine
```



	medicine_ID	medicine_name
1	401	Valium
2	402	Xanax
3	403	Tylenol
4	404	Benadryl
5	405	Advil
6	406	Codeine
7	407	norflox
8	408	meftal

Prescribed_Medicine Table

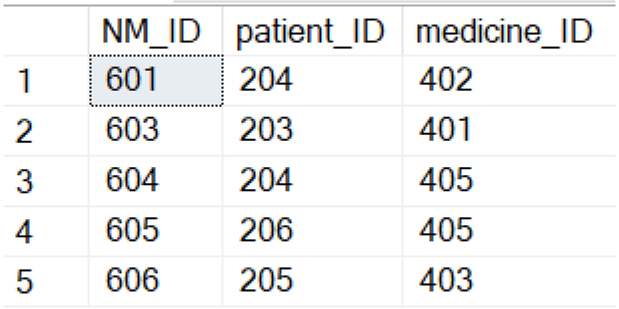
```
select * from prescribed_medicine
```



	pm_id	prescription_ID	medicine_ID	instruction
1	501	301	402	after dinner
2	502	302	401	before breakfast
3	503	303	404	Twice a day
4	504	304	407	thrice a day
5	505	305	406	before breakfast

No_Medicine Table

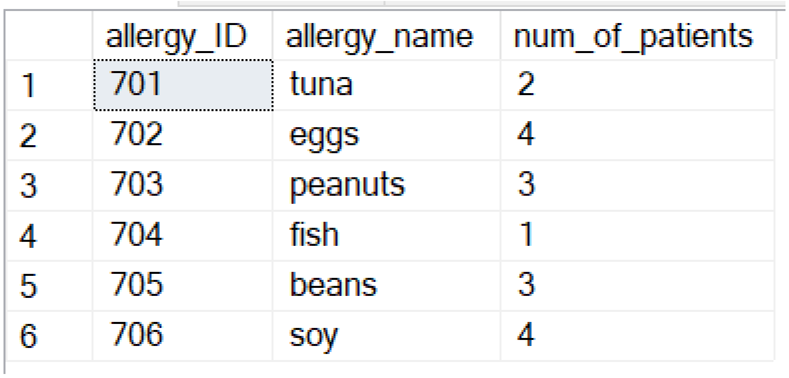
```
select * from no_medicine
```



	NM_ID	patient_ID	medicine_ID
1	601	204	402
2	603	203	401
3	604	204	405
4	605	206	405
5	606	205	403

Allergy table

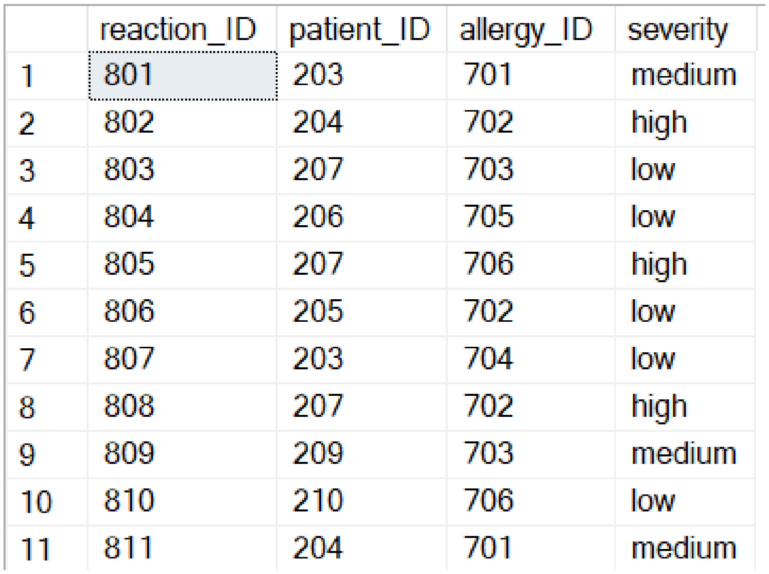
```
select * from allergy
```



	allergy_ID	allergy_name	num_of_patients
1	701	tuna	2
2	702	eggs	4
3	703	peanuts	3
4	704	fish	1
5	705	beans	3
6	706	soy	4

Patient_Allergy Table

```
select * from patient_allergy
```



	reaction_ID	patient_ID	allergy_ID	severity
1	801	203	701	medium
2	802	204	702	high
3	803	207	703	low
4	804	206	705	low
5	805	207	706	high
6	806	205	702	low
7	807	203	704	low
8	808	207	702	high
9	809	209	703	medium
10	810	210	706	low
11	811	204	701	medium

Major Data Questions:

The hospital Management system manages all the activities of the doctor centrally. It also caters to the needs of the patients by streamlining the complex process of keeping a track of medicines and allergies.. There are two users of the system:

i)Doctor
ii)Patient

The below list highlights upon some of the data questions which are answered by the users of the proposed system:

i) <u>Doctor:</u>
The Doctor would have the administrator rights on the system. He can not only retrieve information about his patient and their medicines but also make modifications wherever needed. He will use the system for:

- Prescribing medicines to patients:
  The doctor will have the details and can easily prescribe medicines to the patients. They can give many prescription items to the same patient and same items to different patients.
- The special function of the database is that the doctor can also look at the no_medicine: the medicine that should not be given to the patient because of some earlier medicines that have already been given.
- The doctor, being the admin of the system, can also view and edit the various allergies that patients have. They can see what allergies are associated with which patient and can avoid those things while taking care of the patient at the hospital.

ii) <u>Patient:</u>
The patient can use the database to look at its prescription and prescribed items and also the no medicine and the allergies. But the patient will not have any access to change the data in any form.

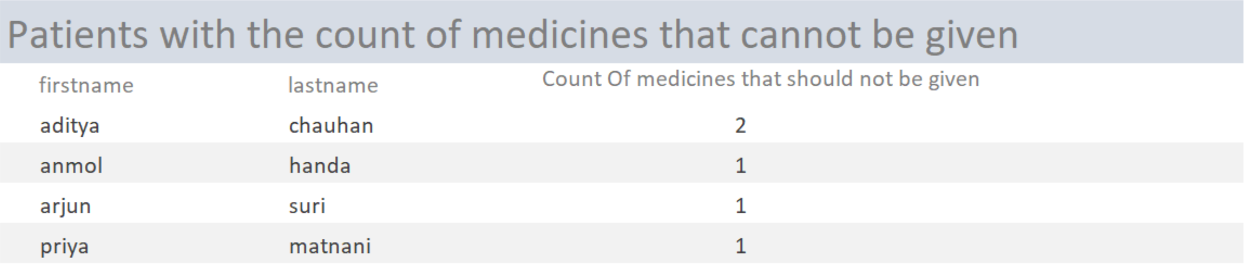
By looking at the allergies that patients might have. He can avoid those foods having allergic substances to take care of his/her health in the future. Similarly with no medicines patients can know about what medicines they should not take

Reports

1. What is the count of medicines that cannot be given to each patient?

   This data question will give the doctors an idea about how many medicines cannot be given to certain patients.

SQL Script:
SELECT Count(dbo_medicine.medicine_ID) AS CountOfmedicine_ID,
dbo_user_t.user_firstname, dbo_user_t.user_lastname
FROM (dbo_medicine INNER JOIN dbo_no_medicine ON dbo_medicine.[medicine_ID] =
dbo_no_medicine.[medicine_ID]) INNER JOIN dbo_user_t ON dbo_no_medicine.patient_ID =
dbo_user_t.login_id
GROUP BY dbo_user_t.user_firstname, dbo_user_t.user_lastname;

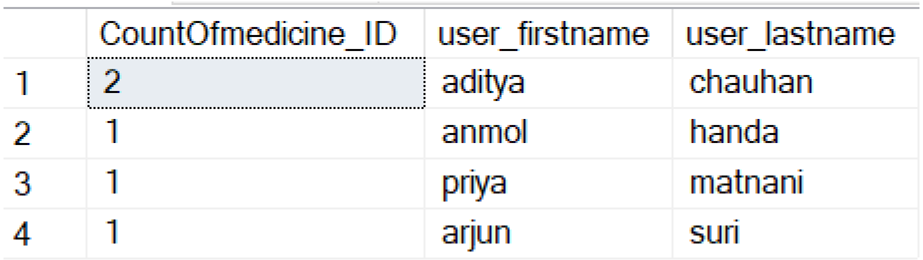


Patients with the count of medicines that cannot be given

firstname	lastname	Count Of medicines that should not be given
aditya	chauhan	2
anmol	handa	1
arjun	suri	1
priya	matnani	1

SQL SERVER QUERY

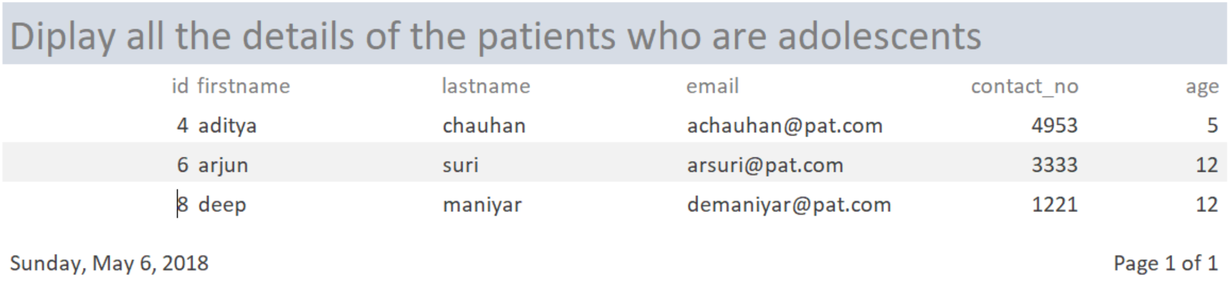
```
SELECT Count(medicine.medicine_ID) AS CountOfmedicine_ID, user_t.user_firstname,
user_t.user_lastname
FROM (medicine INNER JOIN no_medicine ON medicine.[medicine_ID] =
no_medicine.[medicine_ID]) INNER JOIN user_t ON no_medicine.patient_ID = user_t.login_id
GROUP BY user_t.user_firstname, user_t.user_lastname;
```



	CountOfmedicine_ID	user_firstname	user_lastname
1	2	aditya	chauhan
2	1	anmol	handa
3	1	priya	matnani
4	1	arjun	suri

2. How many patients are adolescents that use the hospital services?

SQL Script:
SELECT dbo_user_t.user_id, dbo_user_t.user_firstname, dbo_user_t.user_lastname,
dbo_user_t.user_email, dbo_user_t.contact_no, dbo_user_t.age
FROM dbo_user_t
WHERE (((dbo_user_t.age)<20));



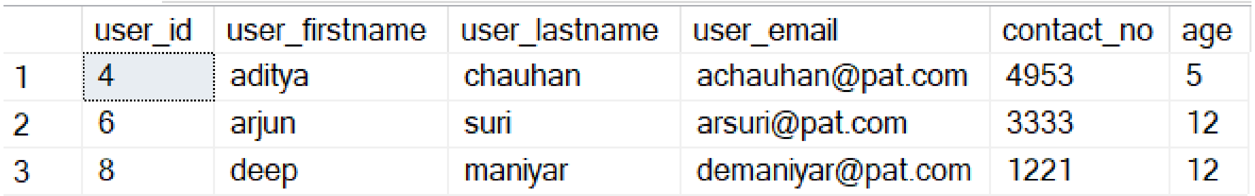
Diplay all the details of the patients who are adolescents

id	firstname	lastname	email	contact_no	age
4	aditya	chauhan	achauhan@pat.com	4953	5
6	arjun	suri	arsuri@pat.com	3333	12
8	deep	maniyar	demaniyar@pat.com	1221	12

Sunday, May 6, 2018 Page 1 of 1

SQL SERVER QUERY

```
SELECT user_t.user_id, user_t.user_firstname, user_t.user_lastname, user_t.user_email,
user_t.contact_no, user_t.age
FROM user_t
WHERE (((user_t.age)<20));
```



	user_id	user_firstname	user_lastname	user_email	contact_no	age
1	4	aditya	chauhan	achauhan@pat.com	4953	5
2	6	arjun	suri	arsuri@pat.com	3333	12
3	8	deep	maniyar	demaniyar@pat.com	1221	12

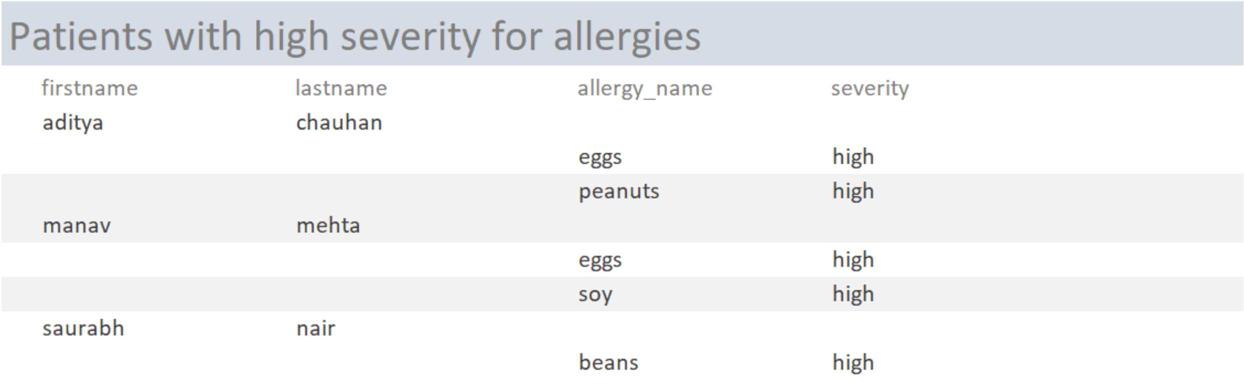
3. Which patients have a high severity to allergies

SQL Script:

SELECT dbo_user_t.user_firstname, dbo_user_t.user_lastname, dbo_allergy.allergy_name, dbo_patient_allergy.severity

FROM (dbo_allergy INNER JOIN dbo_patient_allergy ON dbo_allergy.allergy_ID = dbo_patient_allergy.allergy_ID) INNER JOIN (dbo_login INNER JOIN dbo_user_t ON dbo_login.login_ID = dbo_user_t.login_id) ON dbo_patient_allergy.patient_ID = dbo_user_t.login_id

WHERE (((dbo_patient_allergy.severity)='high'));

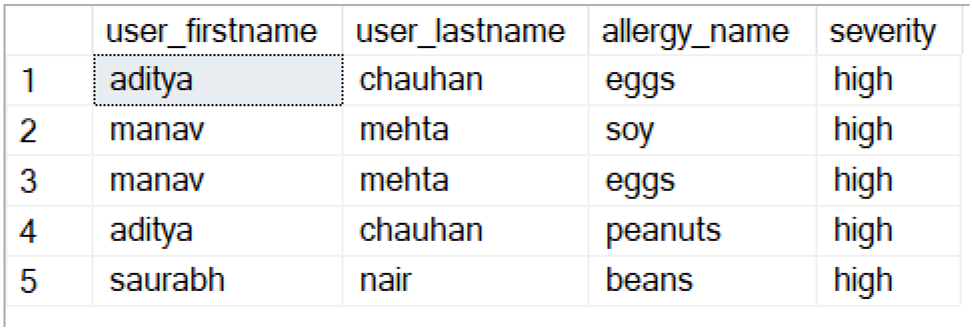


Patients with high severity for allergies

firstname	lastname	allergy_name	severity
aditya	chauhan		
		eggs	high
		peanuts	high
manav	mehta		
		eggs	high
		soy	high
saurabh	nair		
		beans	high

SQL Server query

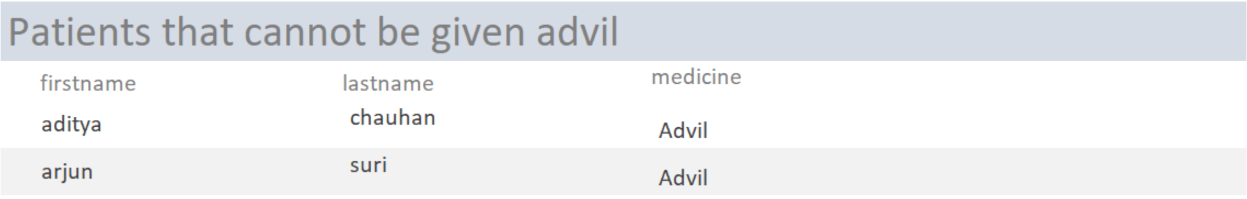
```
SELECT user_t.user_firstname, user_t.user_lastname, allergy.allergy_name,
patient_allergy.severity
FROM (allergy INNER JOIN patient_allergy ON allergy.allergy_ID =
patient_allergy.allergy_ID) INNER JOIN (login INNER JOIN user_t ON login.login_ID =
user_t.login_id) ON patient_allergy.patient_ID = user_t.login_id
WHERE (((patient_allergy.severity)='high'));
```



	user_firstname	user_lastname	allergy_name	severity
1	aditya	chauhan	eggs	high
2	manav	mehta	soy	high
3	manav	mehta	eggs	high
4	aditya	chauhan	peanuts	high
5	saurabh	nair	beans	high

4. What is the list of patients that cannot be prescribed ‘advil’?

SQL Script:
SELECT dbo_user_t.user_firstname, dbo_user_t.user_lastname,
dbo_no_medicine.medicine_ID, dbo_medicine.medicine_name
FROM dbo_medicine INNER JOIN (dbo_no_medicine INNER JOIN dbo_user_t ON
dbo_no_medicine.patient_ID = dbo_user_t.login_id) ON dbo_medicine.medicine_ID =
dbo_no_medicine.medicine_ID
WHERE (((dbo_medicine.medicine_name)='advil'));

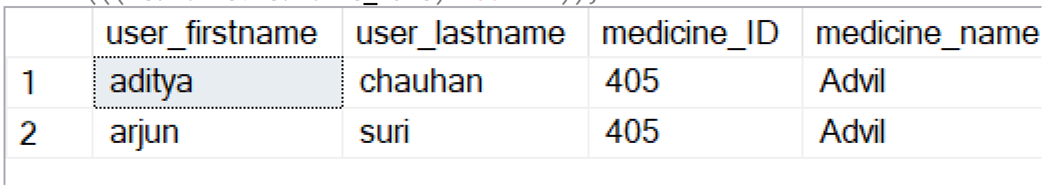


Patients that cannot be given advil

firstname	lastname	medicine
aditya	chauhan	Advil
arjun	suri	Advil

SQL server query

```
SELECT user_t.user_firstname, user_t.user_lastname, no_medicine.medicine_ID,
medicine.medicine_name
FROM medicine INNER JOIN (no_medicine INNER JOIN user_t ON no_medicine.patient_ID =
user_t.login_id) ON medicine.medicine_ID = no_medicine.medicine_ID
WHERE (((medicine.medicine_name)='advil'));
```

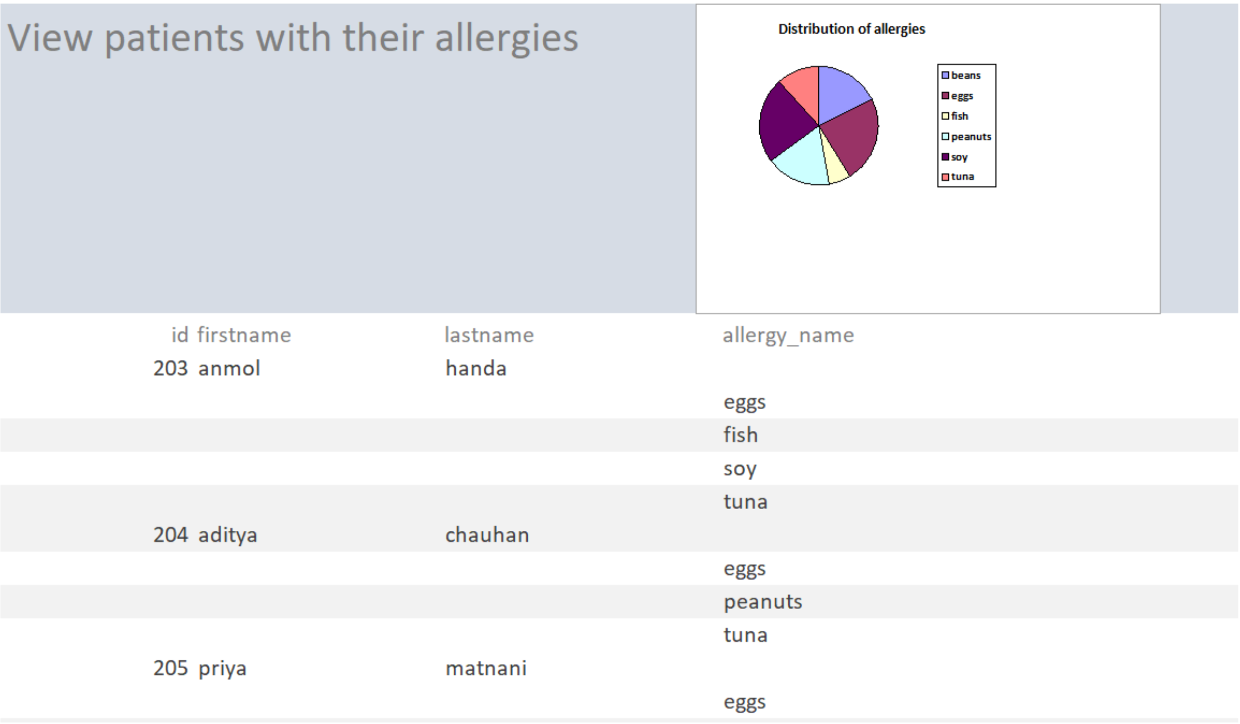


	user_firstname	user_lastname	medicine_ID	medicine_name
1	aditya	chauhan	405	Advil
2	arjun	suri	405	Advil

5. What allergies do the patients have?

SQL Script:

```
SELECT dbo_user_t.login_id, dbo_user_t.user_firstname, dbo_user_t.user_lastname,
dbo_allergy.allergy_name
FROM (dbo_allergy INNER JOIN dbo_patient_allergy ON dbo_allergy.allergy_ID =
dbo_patient_allergy.allergy_ID) INNER JOIN dbo_user_t ON dbo_patient_allergy.patient_ID =
dbo_user_t.login_id;
```



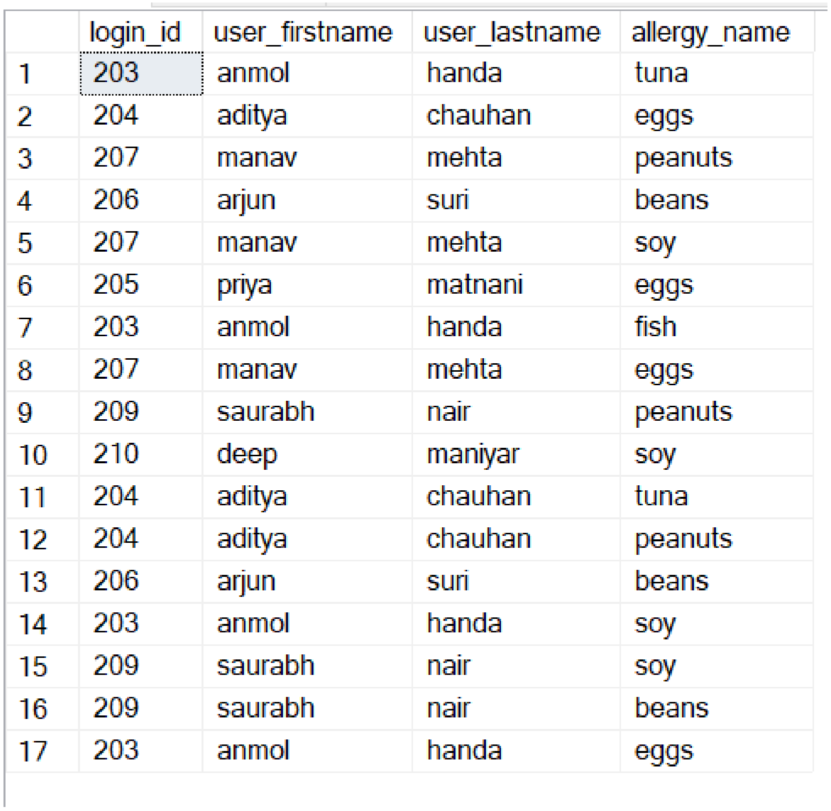
View patients with their allergies

id	firstname	lastname	allergy_name
203	anmol	handa	
			eggs
			fish
			soy
			tuna
204	aditya	chauhan	
			eggs
			peanuts
			tuna
205	priya	matnani	
			eggs

206	arjun	suri	
			beans
			beans
207	manav	mehta	
			eggs
			peanuts
			soy
209	saurabh	nair	
			beans
			peanuts
			soy
210	deep	maniyar	
			soy

SQL server query

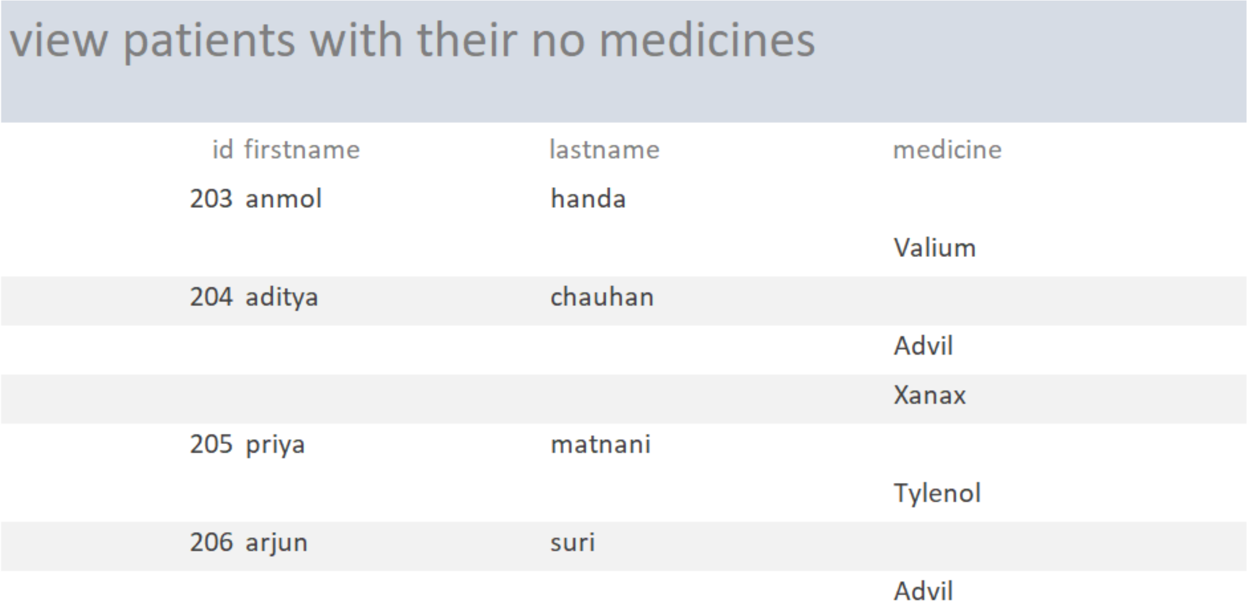
```
SELECT user_t.login_id, user_t.user_firstname, user_t.user_lastname, allergy.allergy_name
FROM (allergy INNER JOIN patient_allergy ON allergy.allergy_ID =
patient_allergy.allergy_ID) INNER JOIN user_t ON patient_allergy.patient_ID =
user_t.login_id;
```



	login_id	user_firstname	user_lastname	allergy_name
1	203	anmol	handa	tuna
2	204	aditya	chauhan	eggs
3	207	manav	mehta	peanuts
4	206	arjun	suri	beans
5	207	manav	mehta	soy
6	205	priya	matnani	eggs
7	203	anmol	handa	fish
8	207	manav	mehta	eggs
9	209	saurabh	nair	peanuts
10	210	deep	maniyar	soy
11	204	aditya	chauhan	tuna
12	204	aditya	chauhan	peanuts
13	206	arjun	suri	beans
14	203	anmol	handa	soy
15	209	saurabh	nair	soy
16	209	saurabh	nair	beans
17	203	anmol	handa	eggs

6. Which medicines cannot be given to patients?

SQL Script:
SELECT dbo_user_t.login_id, dbo_user_t.user_firstname, dbo_user_t.user_lastname, dbo_medicine.medicine_name
FROM dbo_medicine INNER JOIN (dbo_no_medicine INNER JOIN dbo_user_t ON dbo_no_medicine.patient_ID = dbo_user_t.login_id) ON dbo_medicine.medicine_ID = dbo_no_medicine.medicine_ID;

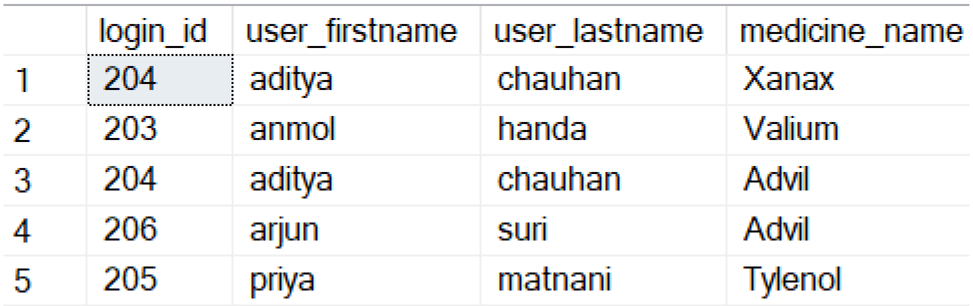


view patients with their no medicines

id	firstname	lastname	medicine
203	anmol	handa	
			Valium
204	aditya	chauhan	
			Advil
			Xanax
205	priya	matnani	
			Tylenol
206	arjun	suri	
			Advil

SQL server query

```
SELECT user_t.login_id, user_t.user_firstname, user_t.user_lastname,
medicine.medicine_name
FROM medicine INNER JOIN (no_medicine INNER JOIN user_t ON no_medicine.patient_ID =
user_t.login_id) ON medicine.medicine_ID = no_medicine.medicine_ID;
```

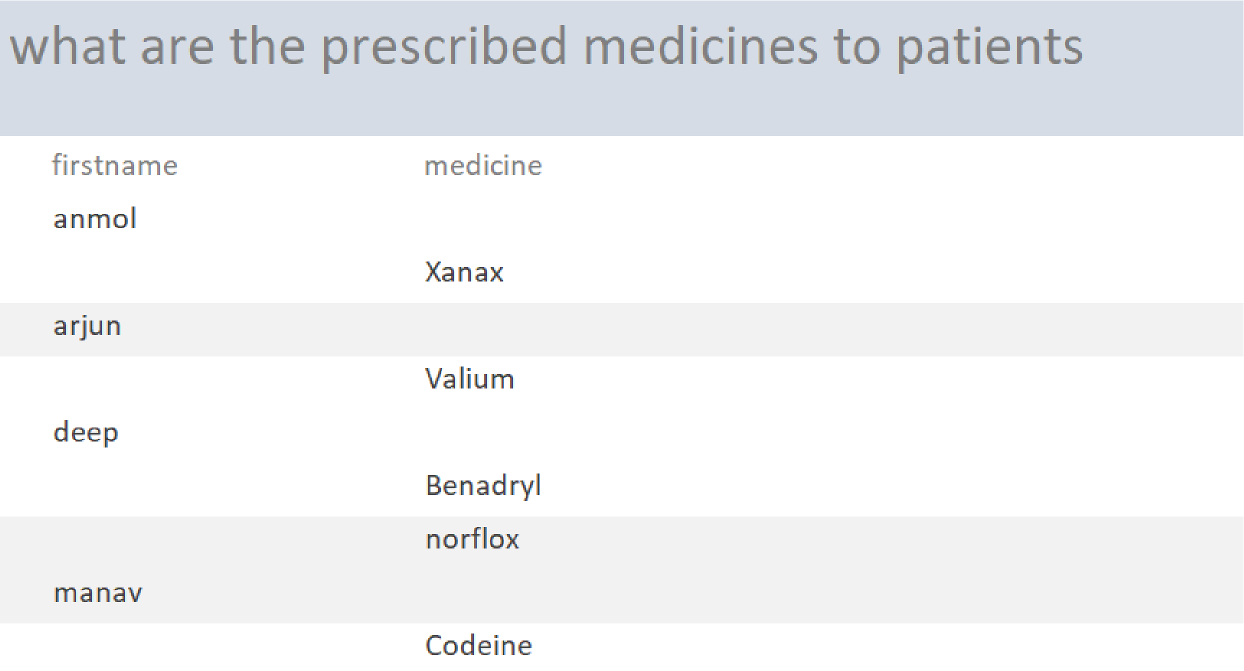


	login_id	user_firstname	user_lastname	medicine_name
1	204	aditya	chauhan	Xanax
2	203	anmol	handa	Valium
3	204	aditya	chauhan	Advil
4	206	arjun	suri	Advil
5	205	priya	matnani	Tylenol

7. What medicines are prescribed to the patients?

SQL Script:
SELECT dbo_user_t.user_firstname, dbo_medicine.medicine_name

FROM (dbo_prescription INNER JOIN (dbo_medicine INNER JOIN dbo_prescribed_medicine ON dbo_medicine.medicine_ID = dbo_prescribed_medicine.medicine_ID) ON dbo_prescription.prescription_ID = dbo_prescribed_medicine.prescription_ID) INNER JOIN dbo_user_t ON (dbo_user_t.login_id = dbo_prescription.patient_ID) AND (dbo_prescription.patient_ID = dbo_user_t.login_id);

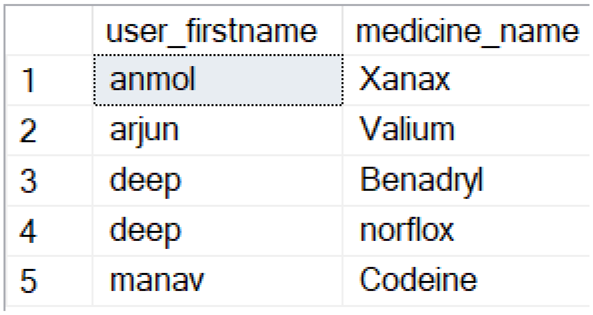


what are the prescribed medicines to patients

firstname	medicine
anmol	
	Xanax
arjun	
	Valium
deep	
	Benadryl
	norflox
manav	
	Codeine

SQL server query

```
SELECT user_t.user_firstname, medicine.medicine_name
FROM (prescription INNER JOIN (medicine INNER JOIN prescribed_medicine ON
medicine.medicine_ID = prescribed_medicine.medicine_ID) ON prescription.prescription_ID =
prescribed_medicine.prescription_ID) INNER JOIN user_t ON (user_t.login_id =
prescription.patient_ID) AND (prescription.patient_ID = user_t.login_id);
```



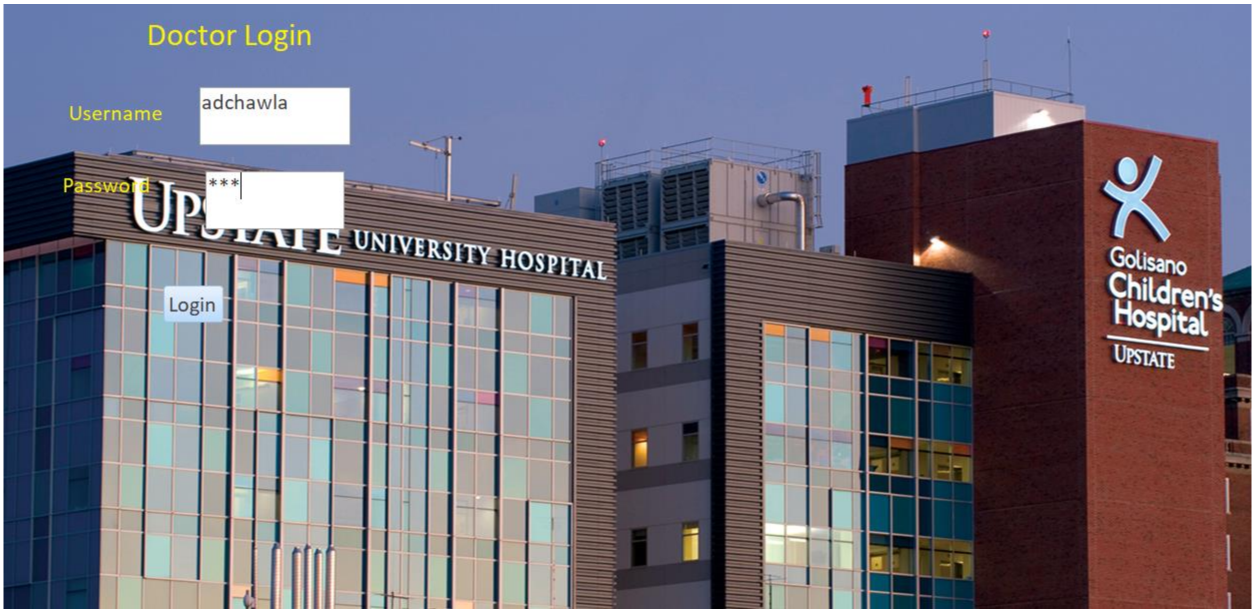
	user_firstname	medicine_name
1	anmol	Xanax
2	arjun	Valium
3	deep	Benadryl
4	deep	norflox
5	manav	Codeine

INTERFACES

FORMS:

DOCTOR

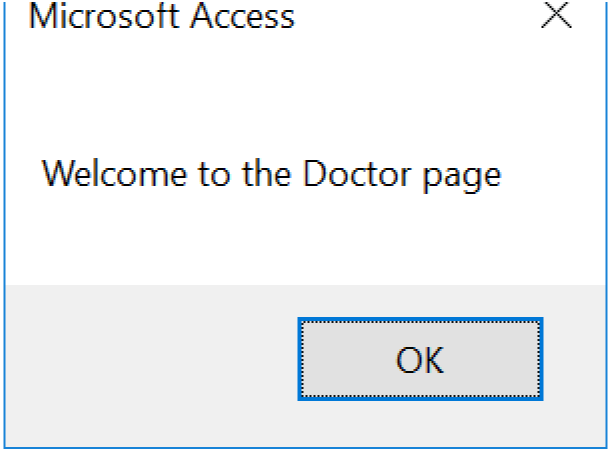
1. Doctor Login



Doctor Login

Username: adchawla

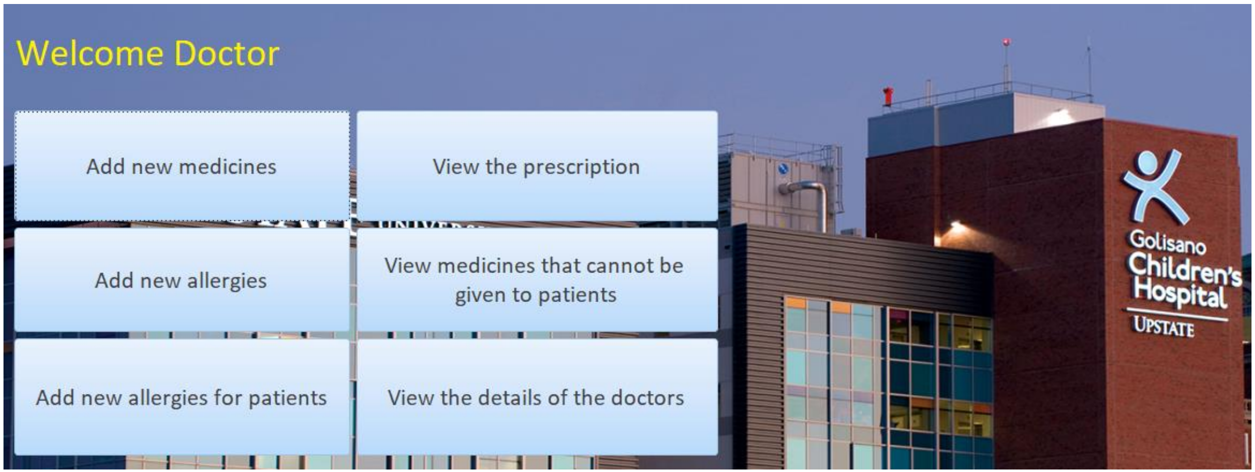
Password: ***

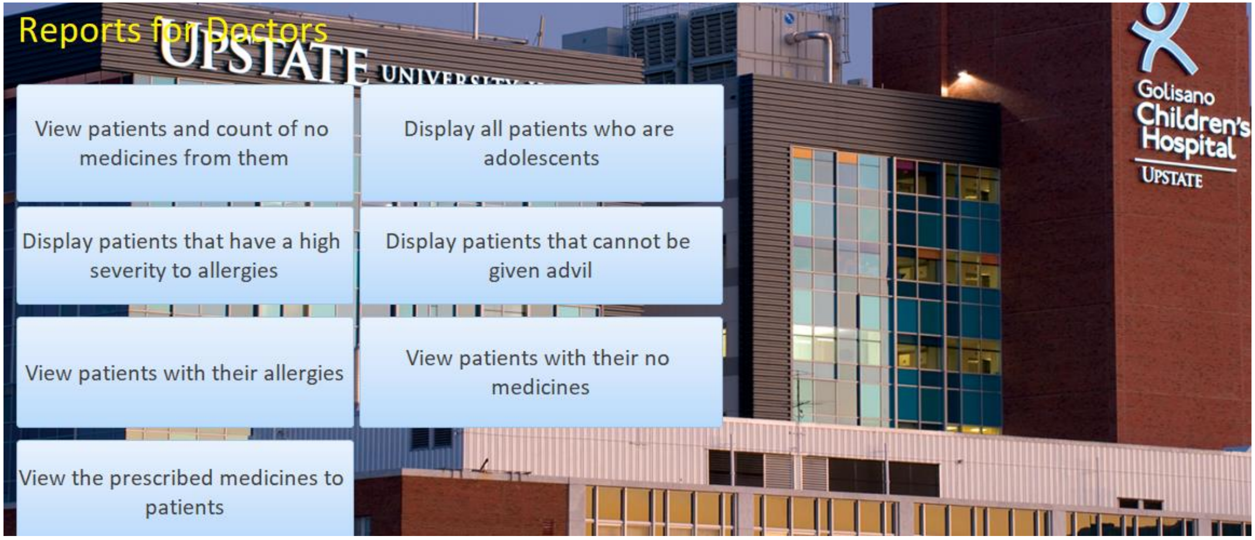


Microsoft Access

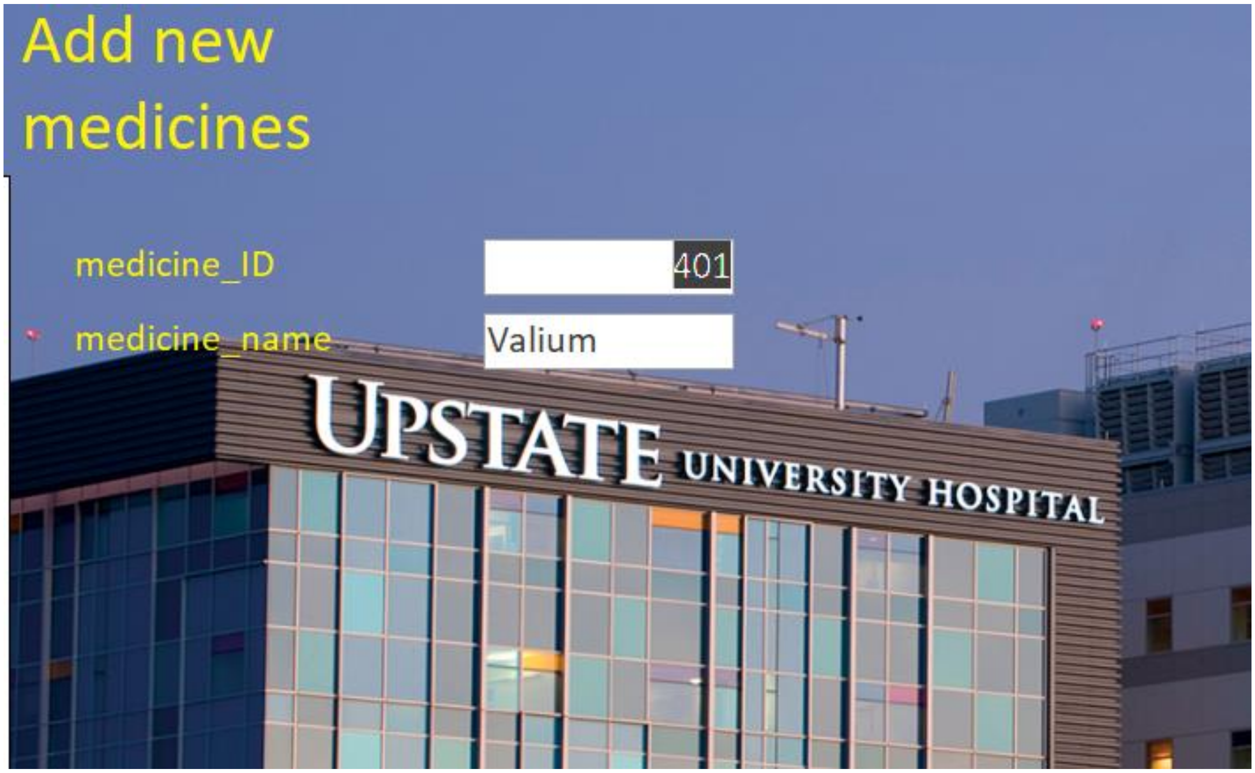
Welcome to the Doctor page

2. Doctor Welcome Page with various forms and reports


Welcome Doctor
Add new medicines
View the prescription
Add new allergies
View medicines that cannot be given to patients
Add new allergies for patients
View the details of the doctors


Reports for Doctors
View patients and count of no medicines from them
Display all patients who are adolescents
Display patients that have a high severity to allergies
Display patients that cannot be given advil
View patients with their allergies
View patients with their no medicines
View the prescribed medicines to patients

3. Adding New medicines

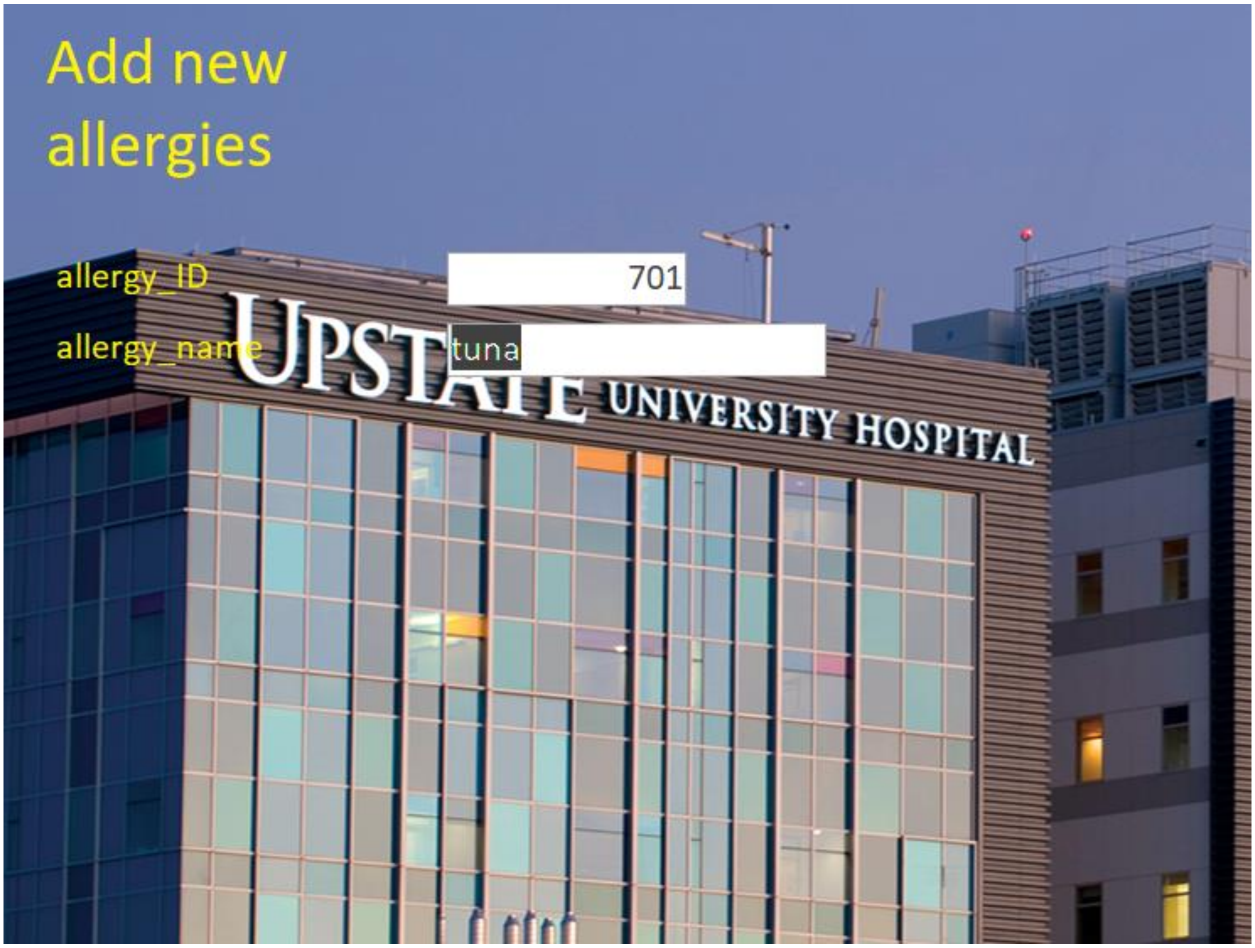


Add new medicines

medicine_ID 401

medicine_name Valium

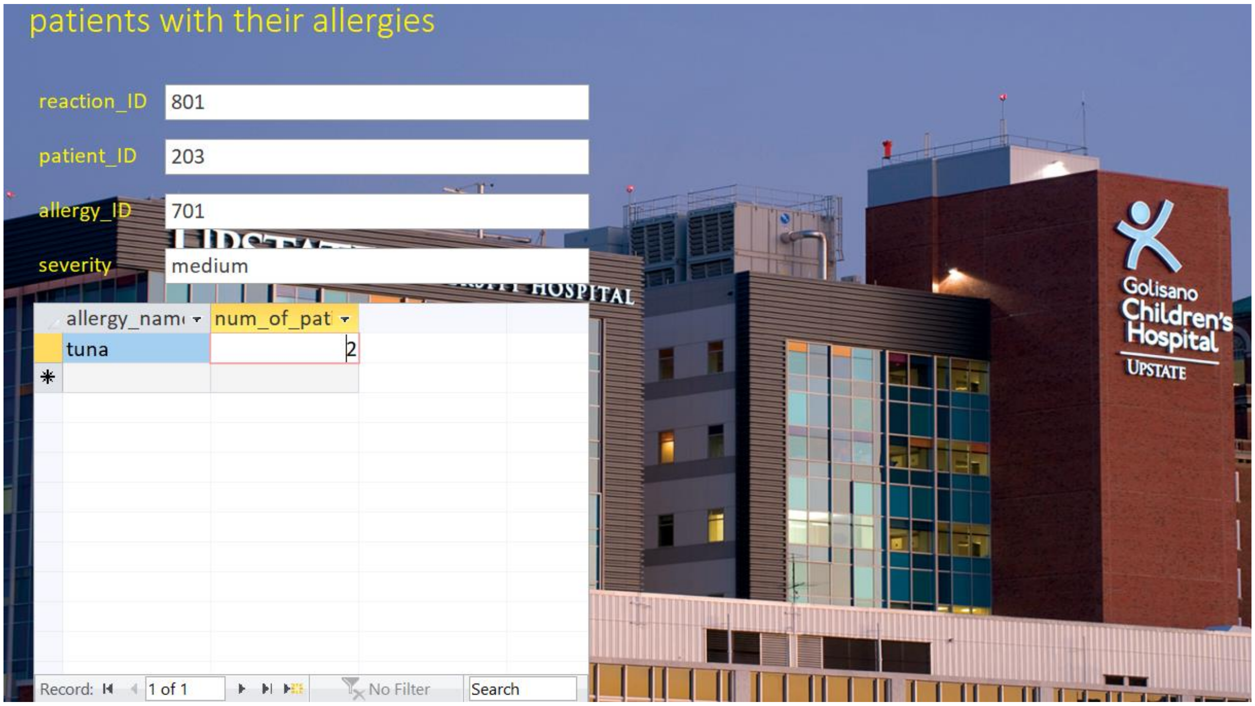
4. Adding new allergies



Add new allergies

allergy_ID	701
allergy_name	tuna

5. Adding new allergies to specific patients

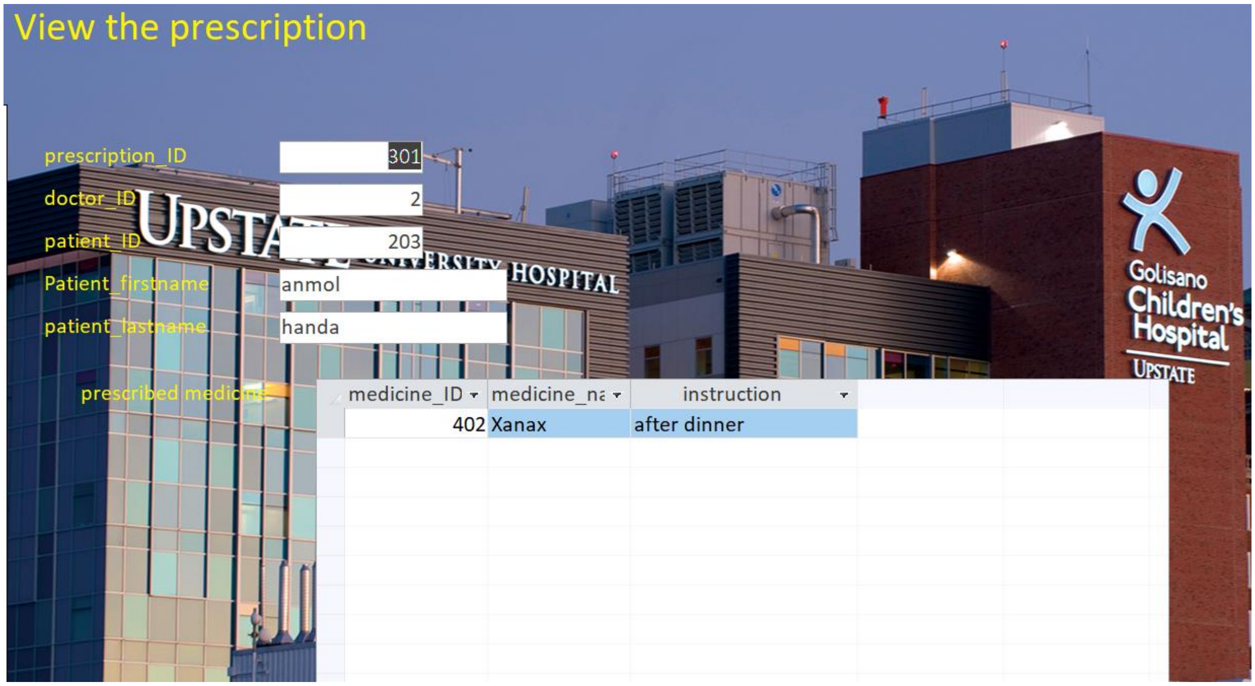


patients with their allergies

reaction_ID	801
patient_ID	203
allergy_ID	701
severity	medium

allergy_nam	num_of_pat
tuna	2

6. Doctors viewing the various prescriptions



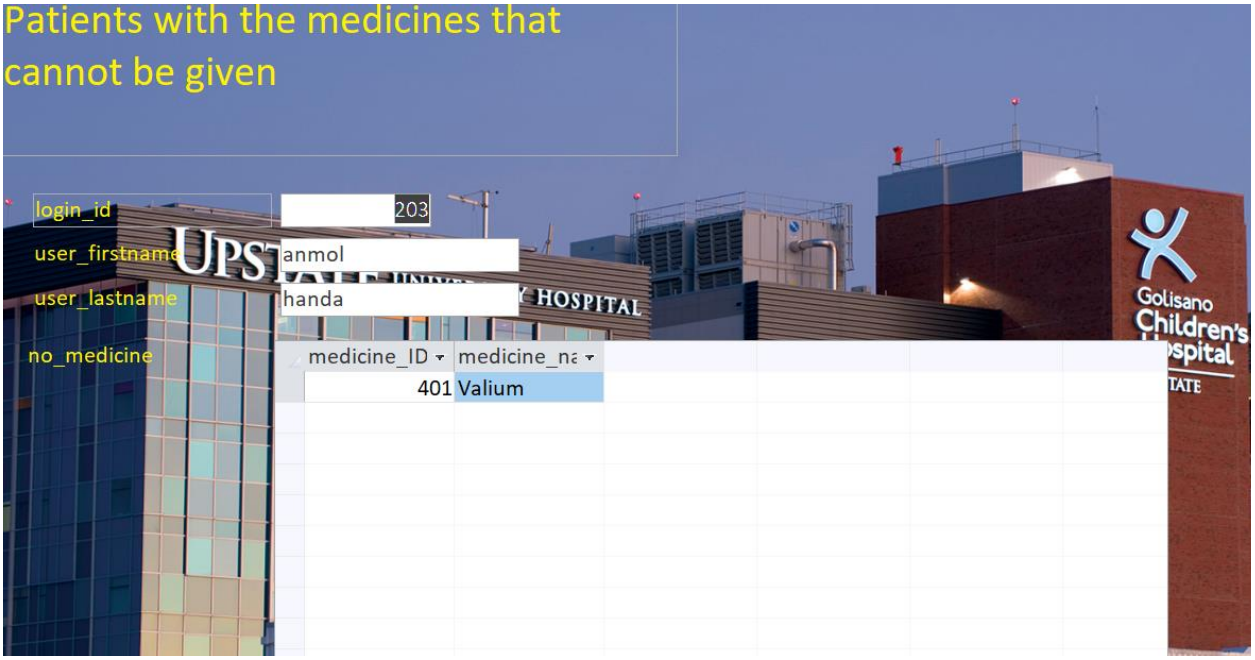
View the prescription

prescription_ID	301
doctor_ID	2
patient_ID	203
Patient_firstname	anmol
patient_lastname	handa

prescribed medicine

medicine_ID	medicine_na	instruction
402	Xanax	after dinner

7. Patients cannot be given the below medicines



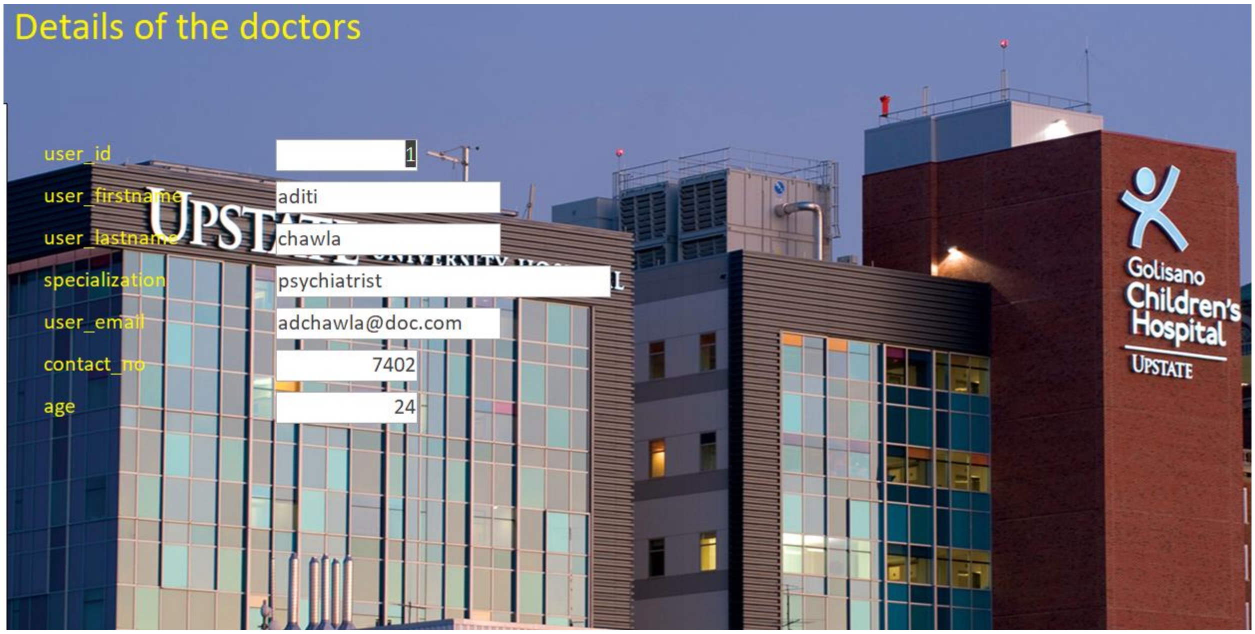
Patients with the medicines that cannot be given

login_id	203
user_firstname	anmol
user_lastname	handa

no_medicine

medicine_ID	medicine_na
401	Valium

8. Details of the doctors present at the hospital



Details of the doctors

user_id: 1

user_firstname: aditi

user_lastname: chawla

specialization: psychiatrist

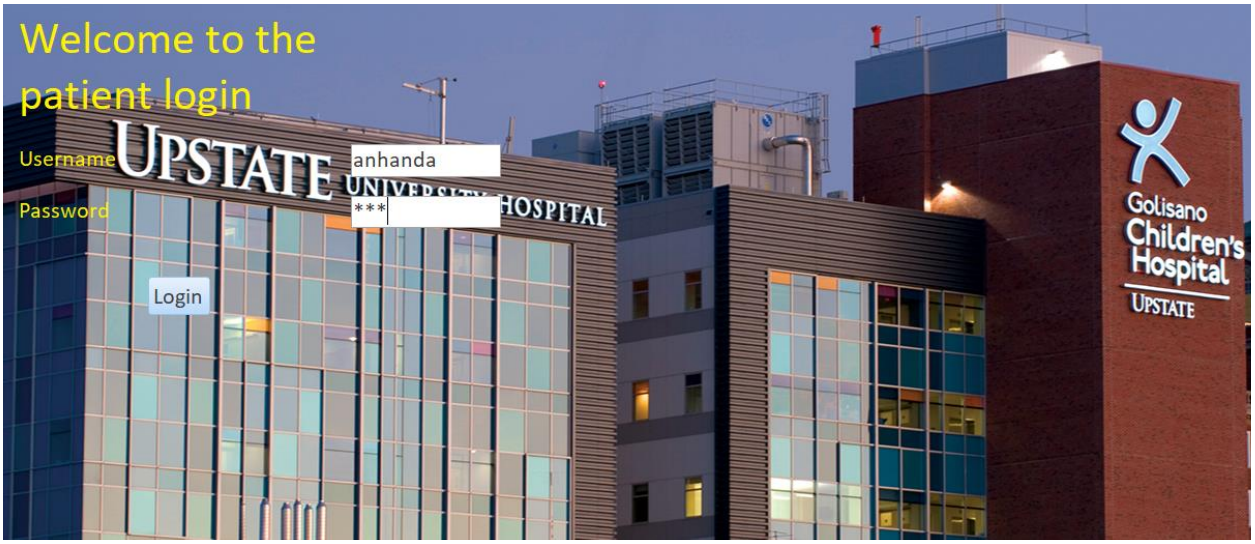
user_email: adchawla@doc.com

contact_no: 7402

age: 24

PATIENT

1. Patient login page

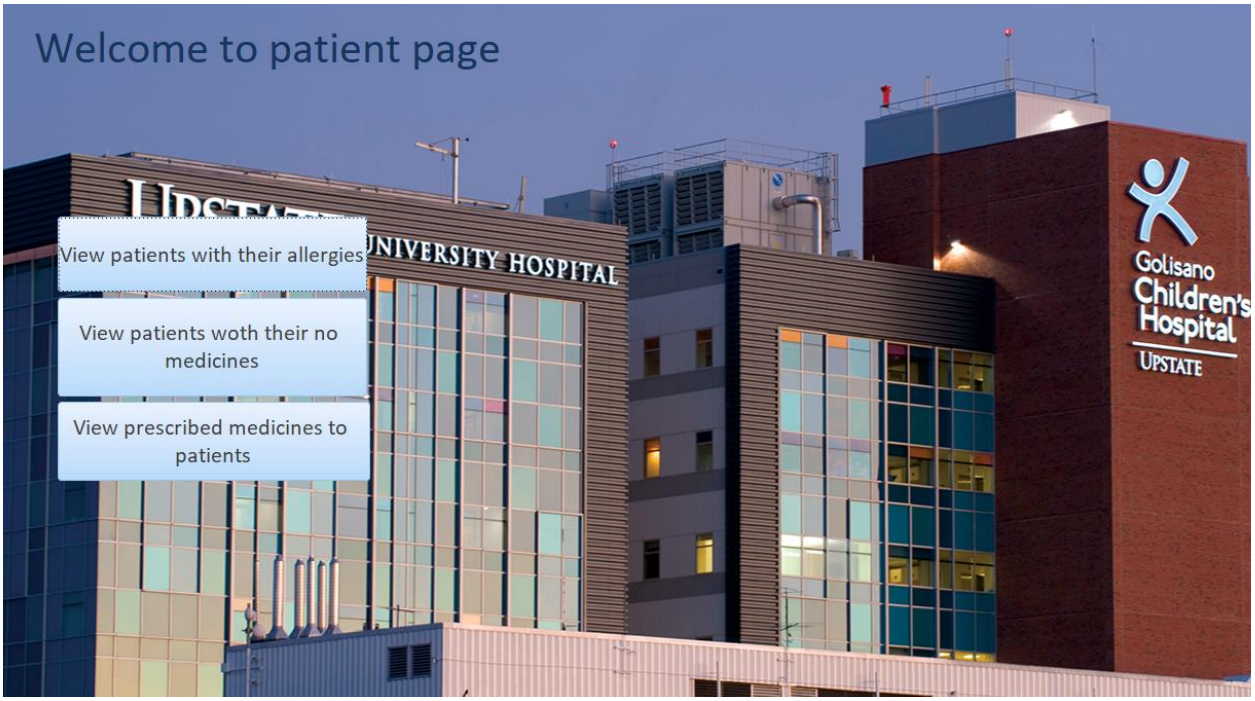


Welcome to the patient login

Username: anhanda

Password: ***

2. Patient access to various reports



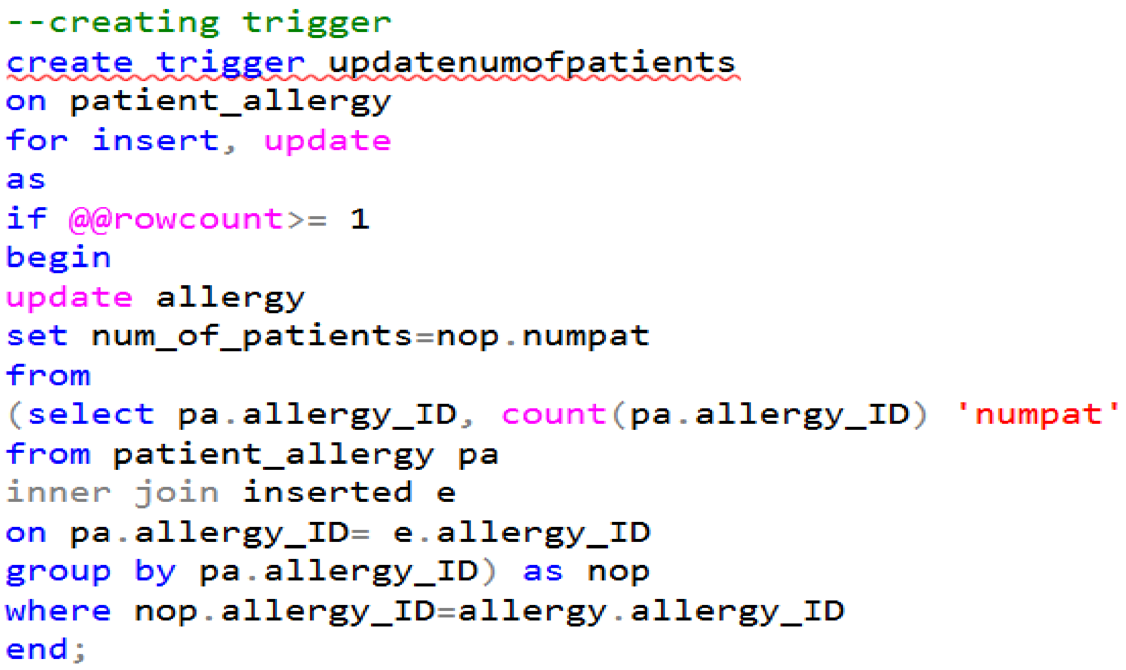
Welcome to patient page

View patients with their allergies
View patients woth their no medicines
View prescribed medicines to patients
UPSTATE
UNIVERSITY HOSPITAL
Golisano Children's Hospital
UPSTATE

TRIGGER

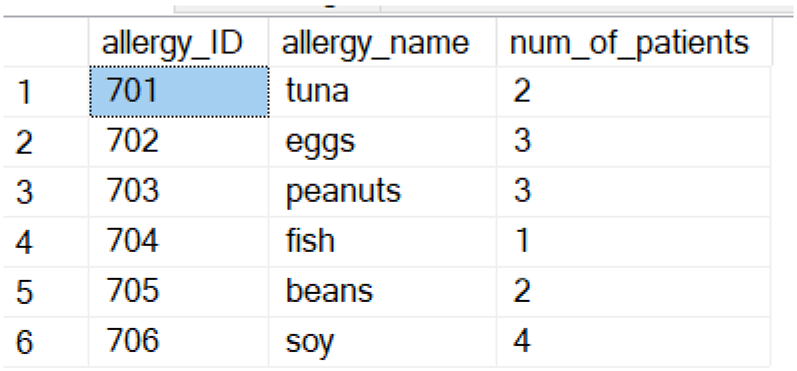
The trigger updates the count of allergies for patients when a new allergy is added to the database

```
create trigger updatenumofpatients
on patient_allergy
for insert, update
as
if @@rowcount>= 1
begin
update allergy
set num_of_patients=nop.numpat
from
(select pa.allergy_ID, count(pa.allergy_ID) 'numpat'
from patient_allergy pa
inner join inserted e
on pa.allergy_ID= e.allergy_ID
group by pa.allergy_ID) as nop
where nop.allergy_ID=allergy.allergy_ID
end;
```



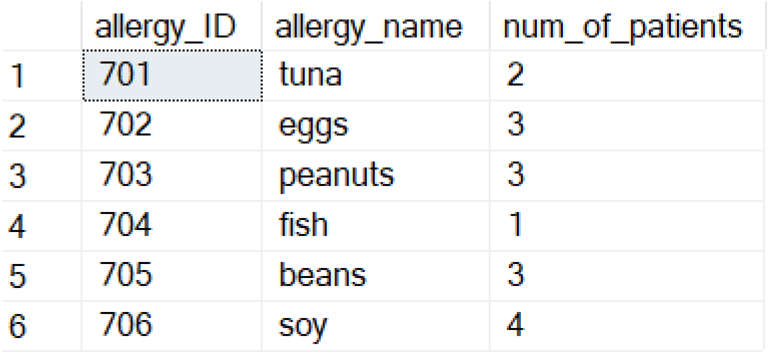
```
--creating trigger
create trigger updatenumofpatients
on patient_allergy
for insert, update
as
if @@rowcount>= 1
begin
update allergy
set num_of_patients=nop.numpat
from
(select pa.allergy_ID, count(pa.allergy_ID) 'numpat'
from patient_allergy pa
inner join inserted e
on pa.allergy_ID= e.allergy_ID
group by pa.allergy_ID) as nop
where nop.allergy_ID=allergy.allergy_ID
end;
```

Allergy table before the trigger



	allergy_ID	allergy_name	num_of_patients
1	701	tuna	2
2	702	eggs	3
3	703	peanuts	3
4	704	fish	1
5	705	beans	2
6	706	soy	4

Allergy table after insert into patient_allergy(reaction_ID, patient_ID,allergy_ID, severity) values(816, 209, 705, 'high')



	allergy_ID	allergy_name	num_of_patients
1	701	tuna	2
2	702	eggs	3
3	703	peanuts	3
4	704	fish	1
5	705	beans	3
6	706	soy	4