



THE iSCHOOL

Syracuse University

PROJECT FINAL REPORT

TOPIC: TELEMEDICINE DATABASE MANAGEMENT SYSTEM



Professor Benjamin Nichols
IST659: Data Administration Concepts &
Database Management

Sankalp Singh
ssingh56@syr.edu
SUID: 974006655
Fall 2019



S. No.	Author	Description	Date
1	Sankalp Singh ssingh56@syr.edu	Telemedicine Database Management System Final Report (Fall 2019)	12/10/2019



Table of Contents

SECTION I: Title Page	4
Introduction.....	4
SECTION II: Project Summary	5
Background	5
Designed Solution.....	5
SECTION III: Glossary Table - Entities and Attributes.....	6
ER Diagram – Visio	10
ER Diagram – SQL Server Management Studio.....	11
Relational Data Model – Access.....	12
SECTION IV: SQL Script – Creation & Insertion	13
Creating Tables	13
Inserting data into Tables.....	15
SECTION V: Major Data Questions – SQL Statements.....	18
Why Doctors have major data questions	18
Why Patients have major data questions	19
Why Chemists have major data questions.....	20
Reports for major data questions	21
SECTION VI: Interfaces – Forms & Reports	31
Forms.....	31
Reports.....	37
SECTION VII: Trigger & Stored Procedure.....	42
Trigger.....	42
Stored Procedure.....	44



SECTION I: Title Page

TELEMEDICINE DATABASE MANAGEMENT SYSTEM



Introduction

Telemedicine Database Management System will be leveraging the concept of telemedicine to provide consultation to patients from certified doctors via video conferencing calls. The system will be having an integrated database consisting of doctors, patients, chemists and medicines based on their location. This system will eventually let patients get doorstep consultation from renowned doctors which will be hassle free. This system will also help people living in remote parts of town who do not have easy access to hospitals and clinics. Financially weaker sections of the society will also be benefitted by this system as our organization will be providing them consultation at nominal rates.

Sankalp Singh
ssingh56@syr.edu
SUID: 974006655
Fall 2019



SECTION II: Project Summary

Background

Currently, most of the people prefer to visit a doctor's clinic or hospitals even for minor ailments. It is just a matter of time that people will be switching to telemedicine apps and websites with the advancement in technology. Telemedicine is the concept of distributing health related services via internet and telecommunication technologies. Our organization will be bringing together all the different bodies including – American medical association registered doctors, patients, chemists and medicines – at one single place based on their location. All the doctors, patients and chemists in a particular city will be able to interact with each other for the sole purpose of providing remedies to people for their minor ailments via video calls. This will also be beneficial for the people living in remote parts of cities who generally do not have easy access to clinics and hospitals. Our organization will also be helping the financially weaker sections of society by providing them online consultation at discounted rates.

Telemedicine is the next big revolution that will be happening in the healthcare sector. Our organization plans to leverage the concept of telemedicine to provide hassle free treatments to people and benefit the healthcare industry.

Designed Solution

- All the AMA registered doctors who are willing to work with our organization will connect with our system.
- The customers will be patients who will register themselves on our system to get consultations from doctor at their doorstep.
- Chemists in an area will also sign up with our system.
- Patients will provide a brief description and history about their ailments while booking for a new consultation.
- Patients can opt for a free delivery of medicines as well from their nearby chemists.
- Doctors will select the time slots when they are free and available. They can just switch on the online button during that time slot so they will be visible to customers when patients are booking the appointments online.
- Once the patient provides information of ailment and selects the doctor they want to consult with, they will be prompted to pay a nominal fee.
- After the patient finish their transaction, they will be directly connected with the doctor via video conferencing. The patient can call back again in the next half hour if there is a call drop due to any reason.
- Once the doctor is done with the consultation, he will prescribe medicines to the patient. The list of medicines will be texted and emailed to the patient. If the patient has opted for a free home delivery of the medicines, the same prescription will be emailed to the chemist as well.



SECTION III: Glossary Table - Entities and Attributes

DATA OBJECT: TELEMEDICINE_MASTER	Contains all the tables and relations that together build the entire Telemedicine Database Management System.
doctor_master	Stores all the details and records for AMA registered doctors.
doctor_id	PRIMARY KEY: Each doctor will have a unique AMA registration ID.
doctor(firstName	Doctor's firstname
doctor(lastName	Doctor's lastname
doctor.specialization	Describes the specialization of doctor – specialist, super specialist, general physician, ENT specialist, skin specialist, child specialist, eye specialist etc.
doctor.fees	Doctor's fees
doctor.availabilitySlot	Doctor's available time slot
doctor.address	Doctor's address
doctor.emailID	Doctor's email address
doctor.phoneNumber	Doctor's phone number
patient_master	Stores all records and details for patients.
patient_id	PRIMARY KEY: Each patient will have a unique ID to identify them.
patient.firstName	Patient's firstname
patient.lastName	Patient's lastname
patient.age	Patient's age
patient.address	Patient's address
patient.salary	Patient's salary
patient.emailID	Patients email address
patient.phoneNumber	Patients phone number



chemist_master	Stores all records and details for chemists.
chemist_id	PRIMARY KEY: Each chemist will have its own unique ID to identify them.
chemist(firstName	Chemist's firstname
chemist(lastName	Chemist's lastname
pharmacy_name	Pharmacy's name
pharmacy_address	Pharmacy's address
pharmacy_emailID	Pharmacy's email address
pharmacy_phoneNumber	Pharmacy's phone number
medicine_master	Stores all records and details related to medicines.
medicine_id	PRIMARY KEY: Each medicine will have a unique ID to identify them.
medicine_name	Medicine's name
medicine_price	Medicine's price
medicine_expiryDate	Medicine's expiry date
medicine_quantity	Medicine's quantity left in the stock
transaction_master	Stores all records and details related to the transactions between doctors and patients
transaction_id	PRIMARY KEY: Each transaction between a doctor and patient will have a unique ID to identify them.
transaction_date	Date of the transaction
transaction_time	Time of the transaction
transaction_amount	Transaction amount
patient_id	FOREIGN KEY: Associated with primary key of 'patient_master' table
doctor_id	FOREIGN_KEY: Associated with primary key of 'doctor_master' table



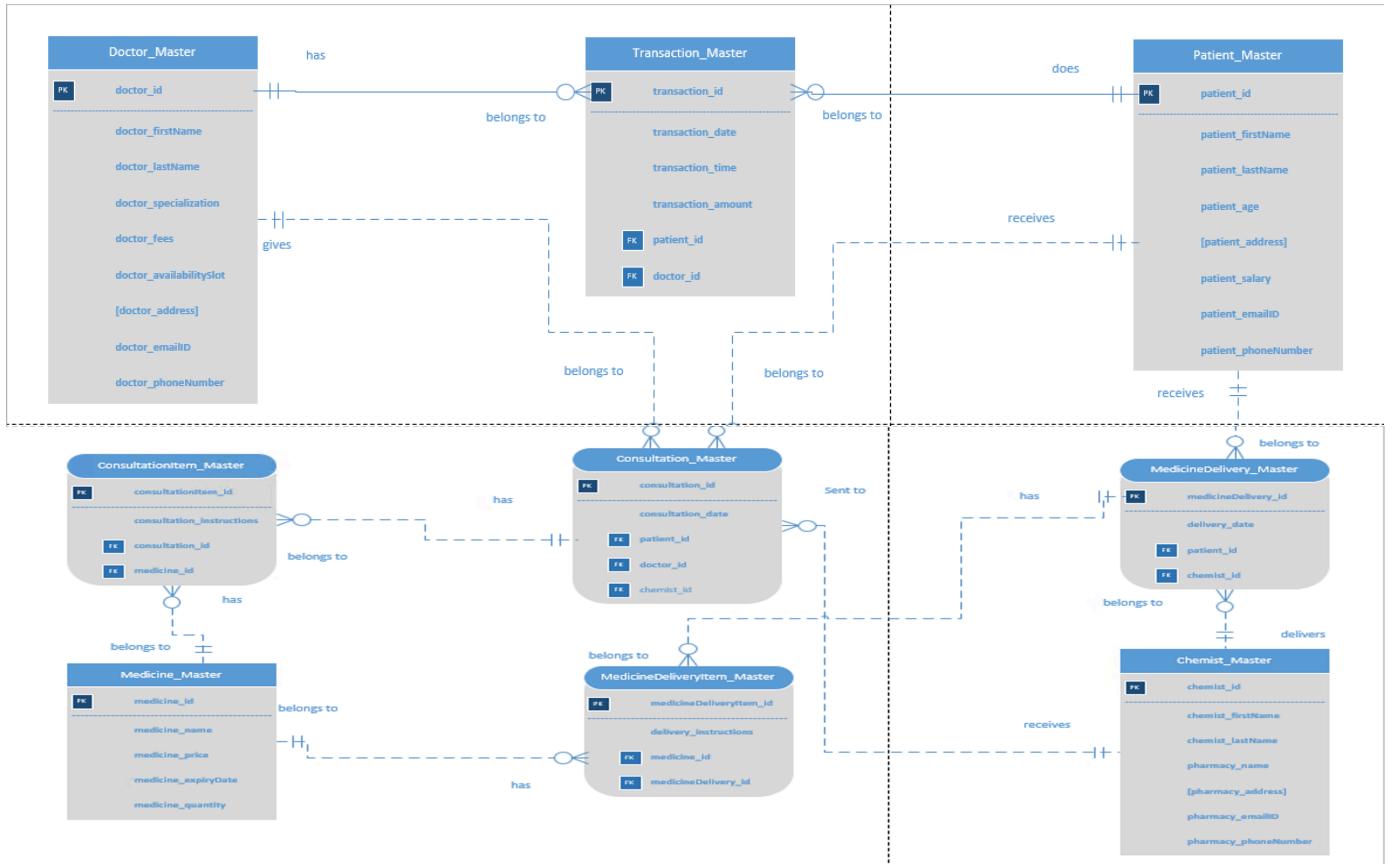
consultation_master	Associative table that keeps track of each consultation between a patient and a doctor.
consultation_id	PRIMARY KEY: Each consultation will have a unique ID to identify them.
consultation_date	Date of consultation
patient_id	FOREIGN KEY: Associated with primary key of 'patient_master' table.
doctor_id	FOREIGN KEY: Associated with primary key of 'doctor_master' table.
Chemist_id	FOREIGN KEY: Associated with primary key of 'chemist_master' table.
consultationItem_master	Associative entity between 'consultation_master' and 'medicine_master'
consultationItem_id	PRIMARY KEY: Each consultationItem will have a unique ID to identify them.
consultation_instructions	Instructions mentioned with the consultation.
consultation_id	FOREIGN KEY: Associated with primary key of 'consultation_master' table.
medicine_id	FOREIGN KEY: Associated with primary key of 'medicine_master' table.
medicineDelivery_master	Associative table that keeps record of all the medicine delivery data between the patients and chemists.
medicineDelivery_id	PRIMARY KEY: Each medicineDelivery has a unique ID to identify them.
delivery_date	Date for the medicine delivery
patient_id	FOREIGN KEY: Associated with primary key of 'patient_master' table.
chemist_id	FOREIGN KEY: Associated with primary key of 'chemist_master' table.



medicineDeliveryItem_master	Associative entity between ‘medicineDelivery_master’ and ‘medicine_master’.
medicineDeliveryItem_id	PRIMARY KEY: Each medicineDeliveryItem has a unique ID to identify them.
delivery_instructions	Instructions mentioned for the home delivery of medicines.
medicine_id	FOREIGN KEY: Associated with primary key of ‘medicine_master’ table.
medicineDelivery_id	FOREIGN KEY: Associated with primary key of ‘medicineDelivery_master’ table.

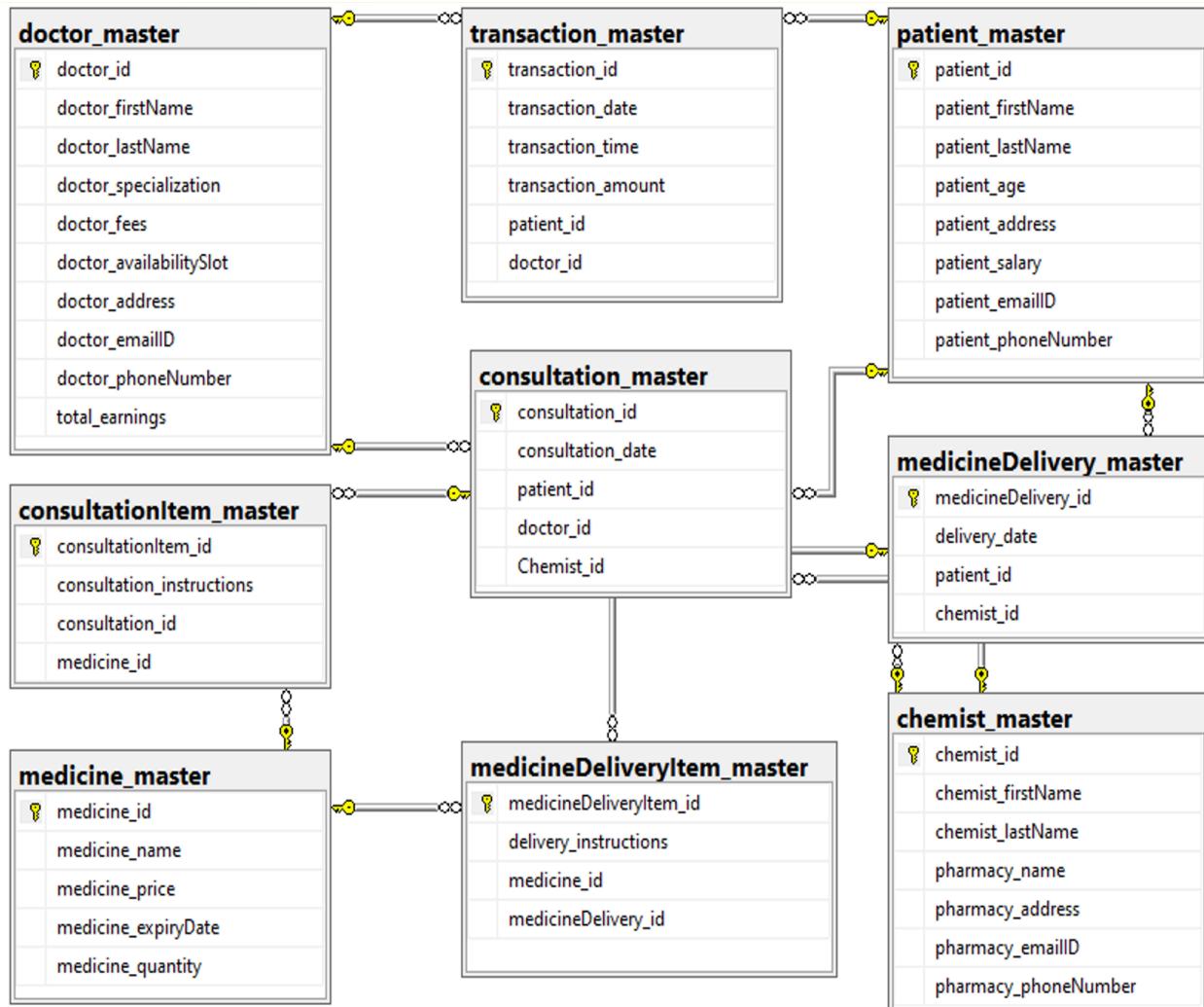


ER Diagram – Visio



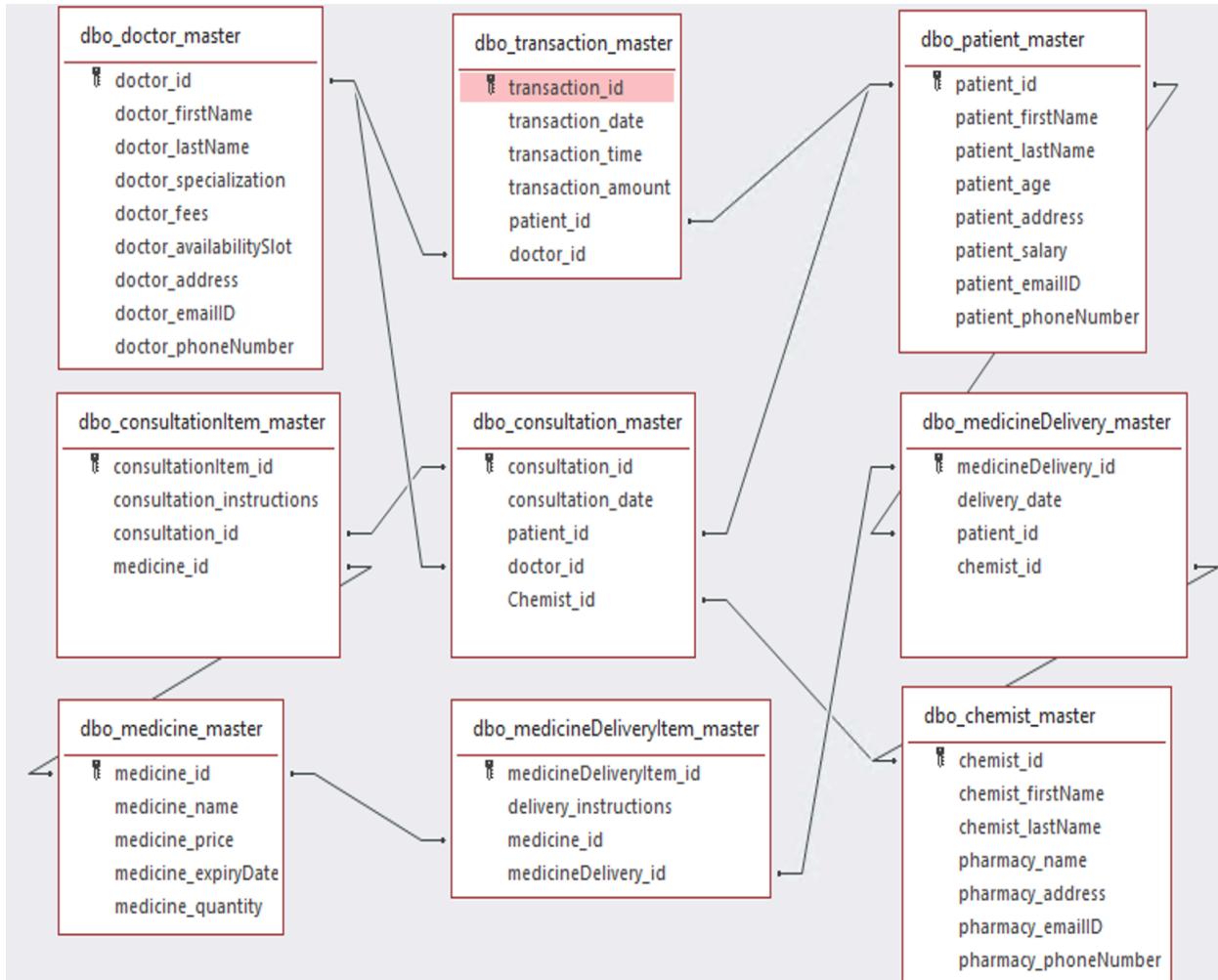


ER Diagram – SQL Server Management Studio





Relational Data Model – Access





SECTION IV: SQL Script – Creation & Insertion

Creating Tables

```

/*Creating a new doctor_master table*/
CREATE TABLE doctor_master (
doctor_id INT PRIMARY KEY,
doctor(firstName VARCHAR(30) NOT NULL,
doctor.lastName VARCHAR(30) NOT NULL,
doctor.specialization VARCHAR(30),
doctor.fees INT,
doctor.availabilitySlot TIME,
doctor.address VARCHAR(50),
doctor.emailID VARCHAR(30),
doctor.phoneNumber INT
);

/*Adding new column to the doctor_master table*/
ALTER TABLE doctor_master ADD total_earnings INT;

/*Creating a new patient_master table*/
CREATE TABLE patient_master (
patient_id INT PRIMARY KEY,
patient.firstName VARCHAR(30) NOT NULL,
patient.lastName VARCHAR(30) NOT NULL,
patient.age INT,
patient.address VARCHAR(50),
patient.salary INT,
patient.emailID VARCHAR(30),
patient.phoneNumber INT
);

/*Creating a new chemist_master table*/
CREATE TABLE chemist_master(
chemist_id INT PRIMARY KEY,
chemist.firstName VARCHAR(30) NOT NULL,
chemist.lastName VARCHAR(30) NOT NULL,
pharmacy.name VARCHAR(30) NOT NULL,
pharmacy.address VARCHAR(50) NOT NULL,
pharmacy.emailID VARCHAR(30) NOT NULL,
pharmacy.phoneNumber INT
);

/*Creating a new medicine_master table*/
CREATE TABLE medicine_master (
medicine_id INT PRIMARY KEY,
medicine.name VARCHAR(30) NOT NULL,
medicine.price INT,
medicine.expiryDate DATE,
medicine.quantity INT
);

```



```
/*Creating a new transaction_master table*/
CREATE TABLE transaction_master (
transaction_id INT PRIMARY KEY,
transaction_date DATE,
transaction_time TIME,
transaction_amount INT,
patient_id INT FOREIGN KEY REFERENCES patient_master(patient_id),
doctor_id INT FOREIGN KEY REFERENCES doctor_master(doctor_id)
);

/*Creating a new consultation_master table*/
CREATE TABLE consultation_master (
consultation_id INT PRIMARY KEY,
consultation_date DATE,
patient_id INT FOREIGN KEY REFERENCES patient_master(patient_id),
doctor_id INT FOREIGN KEY REFERENCES doctor_master(doctor_id),
Chemist_id INT FOREIGN KEY REFERENCES chemist_master(chemist_id)
);

/*Creating a new consultationItem_master table*/
CREATE TABLE consultationItem_master (
consultationItem_id INT PRIMARY KEY,
consultation_instructions VARCHAR(50),
consultation_id INT FOREIGN KEY REFERENCES consultation_master(consultation_id),
medicine_id INT FOREIGN KEY REFERENCES medicine_master(medicine_id)
);

/*Creating a new medicineDelivery_master table*/
CREATE TABLE medicineDelivery_master (
medicineDelivery_id INT PRIMARY KEY,
delivery_date DATE,
patient_id INT FOREIGN KEY REFERENCES patient_master(patient_id),
chemist_id INT FOREIGN KEY REFERENCES chemist_master(chemist_id)
);

/*Creating a new medicineDeliveryItem_master table*/
CREATE TABLE medicineDeliveryItem_master (
medicineDeliveryItem_id INT PRIMARY KEY,
delivery_instructions VARCHAR(50),
medicine_id INT FOREIGN KEY REFERENCES medicine_master(medicine_id),
medicineDelivery_id INT FOREIGN KEY REFERENCES
medicineDelivery_master(medicineDelivery_id)
);
```



Inserting data into Tables

```

/*Inserting data into doctor_master table*/
INSERT INTO doctor_master VALUES (101, 'Jason', 'Shapiro', 'General Physician', 350, '09:00:00', 'Clarendon Street', 'shapiro@doc.com', 2113114111);
INSERT INTO doctor_master VALUES (102, 'Matt', 'LeBlanc', 'Cardiologist', 350, '11:00:00', 'Ackerman Street', 'leblanc@doc.com', 2113114222);
INSERT INTO doctor_master VALUES (103, 'Ross', 'Geller', 'Dermatologist', 350, '12:00:00', 'Lancaster Street', 'geller@doc.com', 2113114333);
INSERT INTO doctor_master VALUES (104, 'Michael', 'Scott', 'ENT Specialist', 350, '13:00:00', 'Westcott Street', 'scott@doc.com', 2113114444);
INSERT INTO doctor_master VALUES (105, 'Dwight', 'Schrute', 'Pediatrician', 350, '14:00:00', 'South Beech Street', 'schrute@doc.com', 2113114555);
INSERT INTO doctor_master VALUES (106, 'Jim', 'Halper', 'Dentist', 350, '15:00:00', 'Ostrom Street', 'halper@doc.com', 2113114666);
INSERT INTO doctor_master VALUES (107, 'Pam', 'Beasley', 'Eye Specialist', 350, '16:00:00', 'Glendale Street', 'beasley@doc.com', 2113114777);
INSERT INTO doctor_master VALUES (108, 'Rachel Green', 'Shapiro', 'Psychiatrist', 350, '17:00:00', 'Marshall Street', 'green@doc.com', 2113114888);
INSERT INTO doctor_master VALUES (109, 'Angela', 'Thomson', 'General Physician', 350, '18:00:00', 'Baker Street', 'thomson@doc.com', 2113114999);
INSERT INTO doctor_master VALUES (110, 'Chandler', 'Bing', 'Mindfulness Specialist', 350, '19:00:00', 'Euclid Street', 'bing@doc.com', 2113114000);
INSERT INTO doctor_master VALUES (111, 'Richard', 'Roy', 'Eye Specialist', 350, '18:00:00', 'Concord Street', 'richard@doc.com', 2119114000);
INSERT INTO doctor_master VALUES (113, 'Andy', 'Bernard', 'Dentist', 350, '14:00:00', 'Laker Street', 'andy@doc.com', 2119114123,0);

/*Inserting data into patient_master table*/
INSERT INTO patient_master VALUES (201, 'Sankalp', 'Singh', '27', '437 Clarendon Street', '100000', 'ssingh@syr.edu', 2013014011);
INSERT INTO patient_master VALUES (202, 'Stanley', 'Cooper', '43', '237 Westcott Street', '120000', 'cooper@pat.com', 2013014911);
INSERT INTO patient_master VALUES (203, 'Kevin', 'Donas', '45', '123 Lancaster Street', '800000', 'donas@pat.com', 2013014811);
INSERT INTO patient_master VALUES (204, 'Toby', 'Mcguire', '32', '734 Ostrom Street', '700000', 'toby@syr.edu', 2013014711);
INSERT INTO patient_master VALUES (205, 'Chris', 'Hemsworth', '35', '444 Ackerman Street', '130000', 'hems@syr.edu', 2013014611);
INSERT INTO patient_master VALUES (206, 'Robert', 'Jr', '44', '40 Baker Street', '140000', 'robert@syr.edu', 2013014511);
INSERT INTO patient_master VALUES (207, 'Evan', 'Smith', '22', '22 Baker Street', '150000', 'smith@syr.edu', 2013014411);
INSERT INTO patient_master VALUES (208, 'Tommy', 'Lee', '50', '333 Euclid Av', '160000', 'lee@syr.edu', 2013014311);
INSERT INTO patient_master VALUES (209, 'Brad', 'Pitt', '53', '99 Manchester Street', '170000', 'pitt@syr.edu', 2013014211);
INSERT INTO patient_master VALUES (210, 'John', 'Rambo', '66', '101 Stron Av', '180000', 'rambo@syr.edu', 2013014114);

```



```

/*Inserting data into chemist_master table*/
INSERT INTO chemist_master VALUES (301, 'Katie', 'Holmes', 'CVS Pharmacy', 'Marshall Street', 'cvs@gmail.com', 2113114114);
INSERT INTO chemist_master VALUES (302, 'Tom', 'Cruise', 'Chase Pharmacy', 'Ackerman Street', 'chase@gmail.com', 2113114111);
INSERT INTO chemist_master VALUES (303, 'Barney', 'Stinson', 'Jet Pharmacy', 'Lancaster Street', 'jet@gmail.com', 2113114112);
INSERT INTO chemist_master VALUES (304, 'Lily', 'Dawson', 'Blue Pharmacy', 'Clarendon Street', 'blue@gmail.com', 2113114113);
INSERT INTO chemist_master VALUES (305, 'Robin', 'Dannings', 'Green Pharmacy', 'Euclid Street', 'green@gmail.com', 2113114115);
INSERT INTO chemist_master VALUES (306, 'Ted', 'Mosby', 'Yellow Pharmacy', 'Ostrom Street', 'yellow@gmail.com', 2113114116);
INSERT INTO chemist_master VALUES (307, 'Marshall', 'Anderson', 'Red Cross Pharmacy', 'Comstock Street', 'redc@gmail.com', 2113114117);
INSERT INTO chemist_master VALUES (308, 'Kevin', 'Donas', 'Health Plus Pharmacy', 'Gilbert Street', 'healthp@gmail.com', 2113114118);
INSERT INTO chemist_master VALUES (309, 'Margeret', 'Robbie', 'Trust Pharmacy', 'Westcott Street', 'trust@gmail.com', 2113114119);
INSERT INTO chemist_master VALUES (310, 'James', 'Blunt', 'CVS Pharmacy', 'Lyra Street', 'cvsp@gmail.com', 2113114110);

/*Inserting data into medicine_master table*/
INSERT INTO medicine_master VALUES (401, 'Paracep', 5, '2020-05-05', 10);
INSERT INTO medicine_master VALUES (402, 'TZ', 3, '2021-05-05', 20);
INSERT INTO medicine_master VALUES (403, 'Benadryl', 4, '2022-05-05', 30);
INSERT INTO medicine_master VALUES (404, 'Dcold', 6, '2020-10-05', 40);
INSERT INTO medicine_master VALUES (405, 'Chestcold', 1, '2020-08-05', 50);
INSERT INTO medicine_master VALUES (406, 'Norflox', 2, '2020-06-05', 60);
INSERT INTO medicine_master VALUES (407, 'Crocin', 8, '2021-03-05', 70);
INSERT INTO medicine_master VALUES (408, 'Vicks', 10, '2023-05-20', 20);
INSERT INTO medicine_master VALUES (409, 'BandAid', 2, '2021-10-10', 90);
INSERT INTO medicine_master VALUES (410, 'Azithromycine', 4, '2022-07-07', 80);

/*Inserting data into transaction_master table*/
INSERT INTO transaction_master VALUES (501, '2019-12-02', '09:00:00', 350, 201, 101);
INSERT INTO transaction_master VALUES (502, '2019-12-03', '10:00:00', 350, 202, 102);
INSERT INTO transaction_master VALUES (503, '2019-12-04', '11:00:00', 350, 203, 103);
INSERT INTO transaction_master VALUES (504, '2019-12-05', '12:00:00', 350, 204, 104);
INSERT INTO transaction_master VALUES (505, '2019-12-06', '13:00:00', 350, 205, 105);
INSERT INTO transaction_master VALUES (506, '2019-12-07', '14:00:00', 350, 206, 106);
INSERT INTO transaction_master VALUES (507, '2019-12-08', '15:00:00', 350, 207, 107);
INSERT INTO transaction_master VALUES (508, '2019-12-09', '16:00:00', 350, 208, 108);
INSERT INTO transaction_master VALUES (509, '2019-12-10', '17:00:00', 350, 209, 109);
INSERT INTO transaction_master VALUES (510, '2019-12-11', '18:00:00', 350, 210, 110);
INSERT INTO transaction_master VALUES (511, '2019-12-13', '10:00:00', 350, 202, 111);
INSERT INTO transaction_master VALUES (512, '2019-12-13', '11:00:00', 350, 203, 112);
INSERT INTO transaction_master VALUES (513, '2019-12-14', '12:00:00', 350, 204, 112);
INSERT INTO transaction_master VALUES (514, '2019-12-15', '13:00:00', 350, 205, 112);
INSERT INTO transaction_master VALUES (515, '2019-12-16', '14:00:00', 350, 207, 112);
INSERT INTO transaction_master VALUES (516, '2019-12-17', '15:00:00', 350, 209, 112);
INSERT INTO transaction_master VALUES (517, '2019-12-18', '16:00:00', 350, 206, 107);
INSERT INTO transaction_master VALUES (518, '2019-12-13', '14:00:00', 350, 205, 104);
INSERT INTO transaction_master VALUES (519, '2019-12-15', '13:00:00', 350, 202, 104);
INSERT INTO transaction_master VALUES (520, '2019-12-13', '12:00:00', 350, 206, 113);

```



```

/*Inserting data into consultation_master table*/
INSERT INTO consultation_master VALUES (601, '2019-12-02', 201, 101, 301);
INSERT INTO consultation_master VALUES (602, '2019-12-03', 202, 102, 302);
INSERT INTO consultation_master VALUES (603, '2019-12-04', 203, 103, 303);
INSERT INTO consultation_master VALUES (604, '2019-12-05', 204, 104, 304);
INSERT INTO consultation_master VALUES (605, '2019-12-06', 205, 105, 305);
INSERT INTO consultation_master VALUES (606, '2019-12-07', 206, 106, 306);
INSERT INTO consultation_master VALUES (607, '2019-12-08', 207, 107, 307);
INSERT INTO consultation_master VALUES (608, '2019-12-09', 208, 108, 308);
INSERT INTO consultation_master VALUES (609, '2019-12-10', 209, 109, 309);
INSERT INTO consultation_master VALUES (610, '2019-12-11', 210, 110, 310);
INSERT INTO consultation_master VALUES (611, '2019-12-11', 207, 105, 310);
INSERT INTO consultation_master VALUES (612, '2019-12-11', 207, 104, 310);
INSERT INTO consultation_master VALUES (613, '2019-12-11', 207, 103, 310);
INSERT INTO consultation_master VALUES (614, '2019-12-13', 204, 112, 305);

/*Inserting data into consultationItem_master table*/
INSERT INTO consultationItem_master VALUES (701, 'Take 3 times a day', 601, 401);
INSERT INTO consultationItem_master VALUES (702, 'Take 2 times a day', 602, 402);
INSERT INTO consultationItem_master VALUES (703, 'Take 1 times a day', 603, 403);
INSERT INTO consultationItem_master VALUES (704, 'Take 4 times a day', 604, 404);
INSERT INTO consultationItem_master VALUES (705, 'Take 5 times a day', 605, 405);
INSERT INTO consultationItem_master VALUES (706, 'Apply daily', 606, 406);
INSERT INTO consultationItem_master VALUES (707, 'Apply 2 times daily', 607, 407);
INSERT INTO consultationItem_master VALUES (708, 'Apply 3 times daily', 608, 408);
INSERT INTO consultationItem_master VALUES (709, 'Apply 4 times daily', 609, 409);
INSERT INTO consultationItem_master VALUES (710, 'Apply 5 times daily', 610, 410);

/*Inserting data into medicineDelivery_master table*/
INSERT INTO medicineDelivery_master VALUES (801, '2019-12-03', 201, 301);
INSERT INTO medicineDelivery_master VALUES (802, '2019-12-04', 202, 302);
INSERT INTO medicineDelivery_master VALUES (803, '2019-12-05', 203, 303);
INSERT INTO medicineDelivery_master VALUES (804, '2019-12-06', 204, 304);
INSERT INTO medicineDelivery_master VALUES (805, '2019-12-07', 205, 305);
INSERT INTO medicineDelivery_master VALUES (806, '2019-12-08', 206, 306);
INSERT INTO medicineDelivery_master VALUES (807, '2019-12-09', 207, 307);
INSERT INTO medicineDelivery_master VALUES (808, '2019-12-10', 208, 308);
INSERT INTO medicineDelivery_master VALUES (809, '2019-12-11', 209, 309);
INSERT INTO medicineDelivery_master VALUES (810, '2019-12-12', 210, 310);
INSERT INTO medicineDelivery_master VALUES (811, '2019-12-08', 203, 305);
INSERT INTO medicineDelivery_master VALUES (812, '2019-12-07', 205, 305);
INSERT INTO medicineDelivery_master VALUES (813, '2019-12-06', 207, 307);
INSERT INTO medicineDelivery_master VALUES (814, '2019-12-05', 209, 307);
INSERT INTO medicineDelivery_master VALUES (815, '2019-12-07', 201, 307);

/*Inserting data into medicineDeliveryItem_master table*/
INSERT INTO medicineDeliveryItem_master VALUES (901, 'Deliver today', 401, 801);
INSERT INTO medicineDeliveryItem_master VALUES (902, 'Deliver tomorrow', 402, 802);
INSERT INTO medicineDeliveryItem_master VALUES (903, 'Deliver tomorrow', 403, 803);
INSERT INTO medicineDeliveryItem_master VALUES (904, 'Deliver today', 404, 804);
INSERT INTO medicineDeliveryItem_master VALUES (905, 'Deliver tomorrow', 405, 805);
INSERT INTO medicineDeliveryItem_master VALUES (906, 'Deliver tomorrow', 406, 806);
INSERT INTO medicineDeliveryItem_master VALUES (907, 'Deliver today', 407, 807);
INSERT INTO medicineDeliveryItem_master VALUES (908, 'Deliver tomorrow', 408, 808);
INSERT INTO medicineDeliveryItem_master VALUES (909, 'Deliver today', 409, 809);
INSERT INTO medicineDeliveryItem_master VALUES (910, 'Deliver tomorrow', 410, 810);

```



SECTION V: Major Data Questions – SQL Statements

- What is the total number of consultations provided by a doctor?
- How many consultations have been taken by a patient?
- How many transactions have been carried out by a doctor?
- Find out the total transactions performed by a patient.
- Find out the total earnings for a doctor.
- How many medicine deliveries have been carried out by a chemist?

Following will be the users of the Telemedicine database management system:

1. Doctors
2. Patients
3. Chemists

Why Doctors have major data questions

- Doctors who have enrolled with our system would like to know how many consultations they have provided at the end of the day along with consultation details. So, our system will be generating a report from ‘consultation_master’ table that contains this information for the doctors.
- Doctors would also want to know the number of transactions they have carried out at the end of the day. We will be generating a report from ‘transaction_master’ that contains this information.
- Doctors would also like to know their total earnings for every fortnight. So, we will be creating a report from ‘transaction_master’ with this information as well.

Furthermore, doctors would like to query our database:

- To Add / Modify / Delete a doctor profile.
- When the doctor adds, update or delete their profile, the ‘doctor_master’ table is updated.
- One constraint over here is that the doctor can only modify their own profile as they will need to login into their profiles with their password in order to update.
- To get the information regarding the patients – Doctors can get information related to patients by looking at records from the ‘patient_master’ table.
- To get information related to chemists - Doctors can get information related to chemists by looking at records from the ‘chemist_master’ table.
- To get medicine information - Doctors can get information related to medicines by looking at records from the ‘medicine_master’ table.



Why Patients have major data questions

- Patient who are using the services of the Telemedicine database management system would like to know the total number of consultations they have taken up in a month along with the consultation details. Our system will be generating a monthly report from ‘consultation_master’ table with this information for the patients.
- Patients would also like to know how many transactions they have made in the past one month. So, our system will be generating a monthly report from ‘transaction_master’ table for patients.

Furthermore, patients would like to query our database:

- To Add / Modify / Delete a patient profile.
- When the patient adds, update or delete their profile, the ‘patient_master’ table is updated.
- One constraint over here is that the patient can only modify their own profile as they will need to login into their profiles with their password in order to update.
- To get details related to doctors - Patients can get information related to doctors by looking at records from the ‘doctor_master’ table.
- To get information regarding chemists – Patients can get information related to chemists by looking at records from the ‘chemist_master’ table.
- To get details regarding the medicines - Patients can get information related to medicines by looking at records from the ‘medicine_master’ table.



Why Chemists have major data questions

- Chemists who have enrolled with our system would like to know how many medicine deliveries have been carried out by them in a week. So, our system will be generating a weekly report from ‘medicineDelivery_master’ table.

Furthermore, chemists would like to query our database:

- To Add / Modify / Delete a chemist profile.
- When the chemist adds, update or delete their profile, the ‘chemist_master’ table is updated.
- One constraint over here is that the chemist can only modify their own profile as they will need to login into their profiles with their password in order to update.
- To get information regarding a particular doctor – Chemist can get information related to doctors by looking at records from the ‘doctor_master’ table.
- To get information related to a particular patient – Chemist can get information related to patients by looking at records from the ‘patient_master’ table.
- To get information related to medicine stock – Chemists can get information related to medicines and their stock by looking at records from the ‘medicine_master’ table.
- To get information related to consultation– Chemists can get information related to consultations by looking at records from the ‘consultation_master’ table.



Reports for major data questions

What is the total number of consultations provided by a doctor?

Report:

This report uses the count aggregate function to display the total consultations provided by a doctor.

Doctor Consultations

Total Consultations	doctor_id	FirstName	LastName	doctor_specialization
1	101	Jason	Shapiro	General Physician
1	102	Matt	LeBlanc	Cardiologist
1	106	Jim	Halper	Dentist
1	107	Pam	Beasley	Eye Specialist
1	108	Rachel Green	Shapiro	Psychiatrist
1	109	Angela	Thomson	General Physician
1	110	Chandler	Bing	Mindfulness Specialis
1	112	Biffy	Clyro	ENT Specialist
2	103	Ross	Geller	Dermatologist
2	104	Michael	Scott	ENT Specialist
2	105	Dwight	Schrute	Pediatrician

Monday, December 2, 2019

Page 1 of 1



SQL Statement:

```

/*Query to find out the total number of consultations
provided by a doctor*/
SELECT dm.doctor_id, dm.doctor(firstName, dm.doctor.lastName,
dm.doctor.specialization,
COUNT(consultation_id) AS 'Total Doctor Consultations'
FROM doctor_master dm
FULL OUTER JOIN consultation_master cm
ON dm.doctor_id=cm.doctor_id
GROUP BY dm.doctor_id, dm.doctor.firstName,
dm.doctor.lastName, dm.doctor.specialization;

```

Screen Copy:

	doctor_id	doctor(firstName	doctor.lastName	doctor.specialization	Total Doctor Consultations
1	101	Jason	Shapiro	General Physician	1
2	102	Matt	LeBlanc	Cardiologist	1
3	103	Ross	Geller	Dermatologist	2
4	104	Michael	Scott	ENT Specialist	2
5	105	Dwight	Schrute	Pediatrician	2
6	106	Jim	Halper	Dentist	1
7	107	Pam	Beasley	Eye Specialist	1
8	108	Rachel Green	Shapiro	Psychiatrist	1
9	109	Angela	Thomson	General Physician	1
10	110	Chandler	Bing	Mindfulness Specialist	1
11	111	Richard	Roy	Eye Specialist	0
12	112	Biffy	Clyro	ENT Specialist	1



How many consultations have been taken by a patient?

Report:

This report uses the count aggregate function to display the total consultations taken up by a patient.

Patient Consultations					
Total Consultations	patient_id	First Name	Last Name	Age	Address
4	207	Evan	Smith	22	22 Baker Street
2	204	Toby	Mcguire	32	734 Ostrom Street
1	205	Chris	Hemsworth	35	444 Ackerman Street
1	206	Robert	Jr	44	40 Baker Street
1	201	Sankalp	Singh	27	437 Clarendon
1	202	Stanley	Cooper	43	237 Westcott Street
1	203	Kevin	Donas	45	123 Lancaster Street
1	208	Tommy	Lee	50	333 Euclid Avenue
1	209	Brad	Pitt	53	99 Manchester Street
1	210	John	Rambo	66	101 Stroh Avenue

**SQL Statement:**

```
/*Query to find out the total number of consultations
taken up by a patient*/
SELECT pm.patient_id, pm.patient(firstName, pm.patient.lastName,
pm.patient.age,
COUNT(consultation_id) AS 'Total Patient Consultations'
FROM patient_master pm
FULL OUTER JOIN consultation_master cm
ON pm.patient_id=cm.patient_id
GROUP BY pm.patient_id, pm.patient.firstName,
pm.patient.lastName,
pm.patient.age;
```

Screen Copy:

	patient_id	patient.firstName	patient.lastName	patient.age	Total Patient Consultations
1	201	Sankalp	Singh	27	1
2	202	Stanley	Cooper	43	1
3	203	Kevin	Donas	45	1
4	204	Toby	Mcquire	32	2
5	205	Chris	Hemsworth	35	1
6	206	Robert	Jr	44	1
7	207	Evan	Smith	22	4
8	208	Tommy	Lee	50	1
9	209	Brad	Pitt	53	1
10	210	John	Rambo	66	1



How many transactions have been carried out by a doctor and their total earnings?

Report:

This report uses the count aggregate function to display the total transactions carried out by a doctor and their total earnings.

Doctor Transactions and Earnings

Total Transactions	doctor_id	FirstName	LastName	Fees	Earnings
5	112	Biffy	Clyro	350	1750
2	107	Pam	Beasley	350	700
1	108	Rachel Green	Shapiro	350	350
1	109	Angela	Thomson	350	350
1	110	Chandler	Bing	350	350
1	111	Richard	Roy	350	350
1	101	Jason	Shapiro	350	350
1	102	Matt	LeBlanc	350	350
1	103	Ross	Geller	350	350
1	104	Michael	Scott	350	350
1	105	Dwight	Schrute	350	350
1	106	Jim	Halper	350	350



SQL Statement:

```
/*Query to find out the total transactions & earnings for a doctor*/
SELECT dm.doctor_id, dm.doctor(firstName, dm.doctor.lastName,
dm.doctor.fees, dm.total_earnings,
COUNT(transaction_id) AS 'Total Doctor Transactions'
FROM doctor_master dm
FULL OUTER JOIN transaction_master tm
ON dm.doctor_id=tm.doctor_id
GROUP BY dm.doctor_id, dm.doctor.firstName,
dm.doctor.lastName, dm.doctor.fees, dm.total_earnings;
```

Screen Copy:

	doctor_id	doctor(firstName	doctor.lastName	doctor.fees	total_earnings	Total Doctor Transactions
1	101	Jason	Shapiro	350	350	1
2	102	Matt	LeBlanc	350	350	1
3	103	Ross	Geller	350	350	1
4	104	Michael	Scott	350	1050	3
5	105	Dwight	Schrute	350	350	1
6	106	Jim	Halper	350	350	1
7	107	Pam	Beasley	350	700	2
8	108	Rachel Green	Shapiro	350	350	1
9	109	Angela	Thomson	350	350	1
10	110	Chandler	Bing	350	350	1
11	111	Richard	Roy	350	350	1
12	112	Biffy	Clyro	350	1750	5
13	113	Andy	Bernard	350	350	1



Find out the total transactions performed by a patient.

Report:

This report uses the count aggregate function to display the total transactions performed by a patient.

Patient Transactions

Total Transactions	patient_id	FirstName	LastName	PhoneNumber
2	202	Stanley	Cooper	2013014911
2	203	Kevin	Donas	2013014811
2	204	Toby	Mcguire	2013014711
2	205	Chris	Hemsworth	2013014611
2	206	Robert	Jr	2013014511
2	207	Evan	Smith	2013014411
2	209	Brad	Pitt	2013014211
1	210	John	Rambo	2013014114
1	201	Sankalp	Singh	2013014011
1	208	Tommy	Lee	2013014311

Monday, December 2, 2019

Page 1 of 1

**SQL Statement:**

```
/*Query to find out the total transactions performed by a patient*/
SELECT pm.patient_id, pm.patient(firstName, pm.patient.lastName,
pm.patient.age,
COUNT(transaction_id) AS 'Total Patient Transactions'
FROM patient_master pm
FULL OUTER JOIN transaction_master tm
ON pm.patient_id=tm.patient_id
GROUP BY pm.patient_id, pm.patient.firstName,
pm.patient.lastName, pm.patient.age;
```

Screen Copy:

	patient_id	patient.firstName	patient.lastName	patient.age	Total Patient Transactions
1	201	Sankalp	Singh	27	1
2	202	Stanley	Cooper	43	3
3	203	Kevin	Donas	45	2
4	204	Toby	Mcguire	32	2
5	205	Chris	Hemsworth	35	3
6	206	Robert	Jr	44	3
7	207	Evan	Smith	22	2
8	208	Tommy	Lee	50	1
9	209	Brad	Pitt	53	2
10	210	John	Rambo	66	1



How many medicine deliveries have been carried out by a chemist?

Report:

This report uses the count aggregate function to display the total medicine deliveries carried out by a chemist.

Chemist Medicine Deliveries					
Total Deliveries	chemist_id	First Name	Last Name	Pharmacy Name	Pharmacy Address
4	307	Marshall	Anderson	Red Cross Pharmacy	Comstock Street
3	305	Robin	Dannings	Green Pharmacy	Euclid Street
1	306	Ted	Mosby	Yellow Pharmacy	Ostrom Street
1	308	Kevin	Donas	Health Plus Pharmacy	Gilbert Street
1	309	Margeret	Robbie	Trust Pharmacy	Westcott Street
1	310	James	Blunt	CVS Pharmacy	Lyra Street
1	301	Katie	Holmes	CVS Pharmacy	Marshall Street
1	302	Tom	Cruise	Chase Pharmacy	Ackerman Street
1	303	Barney	Stinson	Jet Pharmacy	Lancaster Street
1	304	Lily	Dawson	Blue Pharmacy	Clarendon Street

Monday, December 2, 2019

Page 1 of 1



SQL Statement:

```
/*Query to find out total medicine deliveries carried out by a chemist*/
SELECT cm.chemist_id, cm.chemist(firstName, cm.chemist(lastName,
cm.pharmacy_name, cm.pharmacy_address,
COUNT(medicineDelivery_id) AS 'Total Medicine Deliveries'
FROM chemist_master cm
FULL OUTER JOIN medicineDelivery_master mdm
ON cm.chemist_id=mdm.chemist_id
GROUP BY cm.chemist_id, cm.chemist(firstName, cm.chemist(lastName,
cm.pharmacy_name, cm.pharmacy_address;
```

Screen Copy:

	chemist_id	chemist(firstName	chemist(lastName	pharmacy_name	pharmacy_address	Total Medicine Deliveries
1	301	Katie	Holmes	CVS Pharmacy	Marshall Street	1
2	302	Tom	Cruise	Chase Pharmacy	Ackerman Street	1
3	303	Bamey	Stinson	Jet Pharmacy	Lancaster Street	1
4	304	Lily	Dawson	Blue Pharmacy	Clarendon Street	1
5	305	Robin	Dannings	Green Pharmacy	Euclid Street	3
6	306	Ted	Mosby	Yellow Pharmacy	Ostrom Street	1
7	307	Marshall	Anderson	Red Cross Pharmacy	Comstock Street	4
8	308	Kevin	Donas	Health Plus Pharmacy	Gilbert Street	1
9	309	Margeret	Robbie	Trust Pharmacy	Westcott Street	1
10	310	James	Blunt	CVS Pharmacy	Lyra Street	1



SECTION VI: Interfaces – Forms & Reports

Forms

Doctor Information Form – allows you to add and view doctor records

Doctor Details

doctor_id	101	
doctor(firstName	Jason	
doctor.lastName	Shapiro	
doctor.specialization	General Physician	
doctor.fees	350	
doctor.availabilitySlot	09:00:00.0000000	
doctor.address	Clarendon Street	
doctor.emailID	shapiro@doc.com	
doctor.phoneNumber	2113114111	
total_earnings	350	

Combo Box Form – allows you to select the doctor_id from a combo box

Doctor Details

doctor_id	101	
doctor.firstName	101	
doctor.lastName	102	
doctor.specialization	103	
doctor.fees	104	
doctor.availabilitySlot	105	
	an	
	00	



Patient Details Form – allows you to add and view patient records

Patient Details

patient_id	201
patient(firstName	Sankalp
patient(lastName	Singh
patient_age	27
patient_address	437 Clarendon
patient_salary	100000
patient_emailID	ssingh@syr.edu
patient_phoneNumber	2013014011



Chemist Info Form – allows you view and add chemist records

Chemist Information

chemist_id	301
chemist(firstName	Katie
chemist(lastName	Holmes
pharmacy_name	CVS Pharmacy
pharmacy_address	Marshall Street
pharmacy_emailID	cvs@gmail.com
pharmacy_phoneNumber	2113114114



Medicine Details Form – allows you to add and view medicine records

Medicine Info

medicine_id

medicine_name

medicine_price

medicine_expiryDate

medicine_quantity



Transaction Form – allows you to view and add transaction details

Transaction Details

transaction_id

transaction_date

transaction_time

transaction_amount

patient_id

doctor_id



Consultation Details Form – allows you to view and add consultation information

Consultation Details

consultation_id

601

consultation_date

12/2/2019

patient_id

201

doctor_id

101

Chemist_id

301



Consultation Instructions Form – allows you to add and view the consultation instructions

Consultation Instructions

consultationItem_id

701

consultation_instructions

Take 3 times a day

consultation_id

601

medicine_id

401





Medicine Delivery Form – allows you to view and add medicine delivery records

Medicine Delivery Details

medicineDelivery_id	<input type="text" value="801"/>
delivery_date	<input type="text" value="12/3/2019"/>
patient_id	<input type="text" value="201"/>
chemist_id	<input type="text" value="301"/>



Delivery Instructions Form – allows you to view and add delivery instruction records

Medicine Delivery Instructions

medicineDeliveryItem_id	<input type="text" value="901"/>
delivery_instructions	<input type="text" value="Deliver today"/>
medicine_id	<input type="text" value="401"/>
medicineDelivery_id	<input type="text" value="801"/>





Consultation Sub form – allows you to view patient and their respective doctors and consultation records from three tables namely – patient_master, doctor_master, consultation_master

patient_id	207																				
patient(firstName	Evan																				
patient(lastName	Smith																				
Consultation Sub Form	<table border="1"> <thead> <tr> <th>doctor_firstName</th> <th>doctor_specialization</th> <th>consultation_id</th> <th>consultation_date</th> </tr> </thead> <tbody> <tr> <td>Pam</td> <td>Eye Specialist</td> <td>607</td> <td>12/8/2019</td> </tr> <tr> <td>Dwight</td> <td>Pediatrician</td> <td>611</td> <td>12/11/2019</td> </tr> <tr> <td>Michael</td> <td>ENT Specialist</td> <td>612</td> <td>12/11/2019</td> </tr> <tr> <td>Ross</td> <td>Dermatologist</td> <td>613</td> <td>12/11/2019</td> </tr> </tbody> </table>	doctor_firstName	doctor_specialization	consultation_id	consultation_date	Pam	Eye Specialist	607	12/8/2019	Dwight	Pediatrician	611	12/11/2019	Michael	ENT Specialist	612	12/11/2019	Ross	Dermatologist	613	12/11/2019
doctor_firstName	doctor_specialization	consultation_id	consultation_date																		
Pam	Eye Specialist	607	12/8/2019																		
Dwight	Pediatrician	611	12/11/2019																		
Michael	ENT Specialist	612	12/11/2019																		
Ross	Dermatologist	613	12/11/2019																		

Medicine Delivery Sub form – allows you to view medicine delivery details and instruction records from two tables namely – medicineDelivery_master, medicineDeliveryItem_master

medicineDelivery_id	808									
delivery_date	12/10/2019									
patient_id	208									
chemist_id	308									
Medicine Delivery Sub Form	<table border="1"> <thead> <tr> <th>medicineDeliveryItem_id</th> <th>delivery_instructions</th> <th>medicine_id</th> </tr> </thead> <tbody> <tr> <td>908</td> <td>Deliver tomorrow</td> <td>408</td> </tr> <tr> <td>*</td> <td></td> <td></td> </tr> </tbody> </table>	medicineDeliveryItem_id	delivery_instructions	medicine_id	908	Deliver tomorrow	408	*		
medicineDeliveryItem_id	delivery_instructions	medicine_id								
908	Deliver tomorrow	408								
*										



Reports

Total consultations provided by a doctor – this report shows total number of consultations provided by a doctor using a count aggregate function.

Doctor Consultations

Total Consultations	doctor_id	FirstName	LastName	doctor_specialization
1	101	Jason	Shapiro	General Physician
1	102	Matt	LeBlanc	Cardiologist
1	106	Jim	Halper	Dentist
1	107	Pam	Beasley	Eye Specialist
1	108	Rachel Green	Shapiro	Psychiatrist
1	109	Angela	Thomson	General Physician
1	110	Chandler	Bing	Mindfulness Specialis
1	112	Biffy	Clyro	ENT Specialist
2	103	Ross	Geller	Dermatologist
2	104	Michael	Scott	ENT Specialist
2	105	Dwight	Schrute	Pediatrician

Monday, December 2, 2019

Page 1 of 1



Total consultations taken by a patient – this report shows the total number of consultations taken up by a patient using a count aggregate function.

Patient Consultations

Total Consultations	patient_id	First Name	Last Name	Age	Address
4	207	Evan	Smith	22	22 Baker Street
2	204	Toby	Mcguire	32	734 Ostrom Street
1	205	Chris	Hemsworth	35	444 Ackerman Street
1	206	Robert	Jr	44	40 Baker Street
1	201	Sankalp	Singh	27	437 Clarendon
1	202	Stanley	Cooper	43	237 Westcott Street
1	203	Kevin	Donas	45	123 Lancaster Street
1	208	Tommy	Lee	50	333 Euclid Avenue
1	209	Brad	Pitt	53	99 Manchester Street
1	210	John	Rambo	66	101 Stroh Avenue



Total transactions carried out by a doctor & their total earnings – this report uses a count aggregate function to show the total number of transactions carried out by a doctor along with their total earnings till the present date.

Doctor Transactions and Earnings

Total Transactions	doctor_id	FirstName	LastName	Fees	Earnings
5	112	Biffy	Clyro	350	1750
2	107	Pam	Beasley	350	700
1	108	Rachel Green	Shapiro	350	350
1	109	Angela	Thomson	350	350
1	110	Chandler	Bing	350	350
1	111	Richard	Roy	350	350
1	101	Jason	Shapiro	350	350
1	102	Matt	LeBlanc	350	350
1	103	Ross	Geller	350	350
1	104	Michael	Scott	350	350
1	105	Dwight	Schrute	350	350
1	106	Jim	Halper	350	350

Monday, December 2, 2019

Page 1 of 1



Total transactions performed by a patient - this report gives a count of total transactions carried out by a customer using a count aggregate function.

Patient Transactions

Total Transactions	patient_id	FirstName	LastName	PhoneNumber
2	202	Stanley	Cooper	2013014911
2	203	Kevin	Donas	2013014811
2	204	Toby	Mcguire	2013014711
2	205	Chris	Hemsworth	2013014611
2	206	Robert	Jr	2013014511
2	207	Evan	Smith	2013014411
2	209	Brad	Pitt	2013014211
1	210	John	Rambo	2013014114
1	201	Sankalp	Singh	2013014011
1	208	Tommy	Lee	2013014311

Monday, December 2, 2019

Page 1 of 1



Total medicine deliveries carried out by a chemist – this report shows the total medicine deliveries carried out by a chemist using a count aggregate function.

Chemist Medicine Deliveries

Total Deliveries	chemist_id	First Name	Last Name	Pharmacy Name	Pharmacy Address
4	307	Marshall	Anderson	Red Cross Pharmacy	Comstock Street
3	305	Robin	Dannings	Green Pharmacy	Euclid Street
1	306	Ted	Mosby	Yellow Pharmacy	Ostrom Street
1	308	Kevin	Donas	Health Plus Pharmacy	Gilbert Street
1	309	Margeret	Robbie	Trust Pharmacy	Westcott Street
1	310	James	Blunt	CVS Pharmacy	Lyra Street
1	301	Katie	Holmes	CVS Pharmacy	Marshall Street
1	302	Tom	Cruise	Chase Pharmacy	Ackerman Street
1	303	Barney	Stinson	Jet Pharmacy	Lancaster Street
1	304	Lily	Dawson	Blue Pharmacy	Clarendon Street

Monday, December 2, 2019

Page 1 of 1



SECTION VII: Trigger & Stored Procedure

Trigger

This trigger automatically updates the ‘total_earnings’ column for a particular doctor in the ‘doctor_master’ table whenever there is a new transaction carried out for that doctor in ‘transaction_master’ table.

SQL Statement:

```
/* Trigger to update total_earnings in doctor_master
on insert in transaction_master */
CREATE TRIGGER update_doctor_earnings
ON transaction_master
FOR INSERT
AS
IF @@ROWCOUNT>=1
BEGIN
UPDATE doctor_master
SET total_earnings = (transactionCount.total_earnings)*350
FROM
(
SELECT doctor_id, COUNT(DISTINCT transaction_id) 'total_earnings'
FROM transaction_master
GROUP BY doctor_id
) AS transactionCount
WHERE doctor_master.doctor_id=transactionCount.doctor_id
END;
```



Before execution of trigger:

- SELECT * FROM doctor_master;
- ‘doctor_id’ = ‘113’ has ‘total_earnings’ = ‘0’ currently in the ‘doctor_master’ table

doctor_id	doctor(firstName)	doctor(lastName)	doctor_specialization	doctor_fees	doctor_availabilitySlot	doctor_address	doctor_emailID	doctor_phone...	total_earnings
103	Ross	Geller	Dermatologist	350	12:00:00.0000000	Lancaster Street	geller@doc.com	2113114333	350
104	Michael	Scott	ENT Specialist	350	13:00:00.0000000	Westcott Street	scott@doc.com	2113114444	1050
105	Dwight	Schrute	Pediatrician	350	14:00:00.0000000	South Beech Street	schrute@doc.com	2113114555	350
106	Jim	Halper	Dentist	350	15:00:00.0000000	Ostrom Street	halper@doc.com	2113114666	350
107	Pam	Beasley	Eye Specialist	350	16:00:00.0000000	Glendale Street	beasley@doc.com	2113114777	700
108	Rachel Green	Shapiro	Psychiatrist	350	17:00:00.0000000	Marshall Street	green@doc.com	2113114888	350
109	Angela	Thomson	General Physician	350	18:00:00.0000000	Baker Street	thomson@doc.com	2113114999	350
110	Chandler	Bing	Mindfulness Specialist	350	19:00:00.0000000	Euclid Street	bing@doc.com	2113114000	350
111	Richard	Roy	Eye Specialist	350	18:00:00.0000000	Concord Street	richard@doc.com	2119114000	350
112	Biffy	Clyro	ENT Specialist	350	11:00:00.0000000	Comstock Avenue	biffy@doc.com	2013014015	1750
113	Andy	Bernard	Dentist	350	14:00:00.0000000	Laker Street	andy@doc.com	2119114123	0

After execution of trigger:

- INSERT INTO transaction_master VALUES (520, '2019-12-13', '12:00:00', 350, 206, 113);
- This insert in the ‘transaction_master’ table triggers the update in the ‘doctor_master’ table.
- ‘total_earnings’ column for ‘doctor_master’ table is updated to ‘350’ for the ‘doctor_id’ = ‘113’ when there is insertion in ‘transaction_master’ table for ‘doctor_id’ = ‘113’

doctor_id	doctor(firstName)	doctor(lastName)	doctor_specialization	doctor_fees	doctor_availabilitySlot	doctor_address	doctor_emailID	doctor_phone...	total_earnings
104	Michael	Scott	ENT Specialist	350	13:00:00.0000000	Westcott Street	scott@doc.com	2113114444	1050
105	Dwight	Schrute	Pediatrician	350	14:00:00.0000000	South Beech Street	schrute@doc.com	2113114555	350
106	Jim	Halper	Dentist	350	15:00:00.0000000	Ostrom Street	halper@doc.com	2113114666	350
107	Pam	Beasley	Eye Specialist	350	16:00:00.0000000	Glendale Street	beasley@doc.com	2113114777	700
108	Rachel Green	Shapiro	Psychiatrist	350	17:00:00.0000000	Marshall Street	green@doc.com	2113114888	350
109	Angela	Thomson	General Physician	350	18:00:00.0000000	Baker Street	thomson@doc.com	2113114999	350
110	Chandler	Bing	Mindfulness Specialist	350	19:00:00.0000000	Euclid Street	bing@doc.com	2113114000	350
111	Richard	Roy	Eye Specialist	350	18:00:00.0000000	Concord Street	richard@doc.com	2119114000	350
112	Biffy	Clyro	ENT Specialist	350	11:00:00.0000000	Comstock Avenue	biffy@doc.com	2013014015	1750
113	Andy	Bernard	Dentist	350	14:00:00.0000000	Laker Street	andy@doc.com	2119114123	350



Stored Procedure

This procedure updates the ‘total_earnings’ for a doctor in ‘doctor_master’ table upon execution.

SQL Statement:

```
/* Procedure to update earnings for doctor_master table */
CREATE PROCEDURE doctor_earnings
AS
BEGIN
UPDATE doctor_master
SET total_earnings = (transactionCount.total_earnings)*350
FROM
(
SELECT doctor_id, COUNT(DISTINCT transaction_id) 'total_earnings'
FROM transaction_master
GROUP BY doctor_id
) AS transactionCount
WHERE doctor_master.doctor_id=transactionCount.doctor_id
END;

/*Executing the stored procedure*/
EXEC doctor_earnings;
```