



N'Dea Jackson
IST 659 - Winter 2020

Summary

Jackson Memorial Hospital is located in Fort Washington, Maryland and has been operating since December of 1995. In order to provide the best medical assistance to its patients, the hospital must have an efficient system in place. Jackson Memorial Hospital is currently working on developing a hospital management system database to keep track of their patient and doctor information. The data is currently being kept in a number of ways including on multiple siloed systems operated by individual departments and on physical paper charts. The way that things are currently being operated is making it difficult to keep information consistent and also making it difficult to view all patient information at once. By implementing this database, Jackson Memorial will be able to better track and update their patient information, as well as keep track of activities and better maintain all hospital records by minimizing paperwork and siloed applications.

Stakeholders

The stakeholders included in this project are the patients, doctors, staff, and current hospital owners. By implementing this database, all stakeholders will benefit in their own individual ways. When patient information needs to be updated, the information will be updated across all departments of the hospital, allowing everyone to have the most recent information to work with. This will allow for a streamlined treatment process for patients. Doctors will be able to have easier access to their patient's information in an amalgamated view. Hospital owners will be able to track the efficiency of various areas of the hospital, based on doctor productivity and room usage, by having a day-to-day view of hospital operations. Financial leaders for the hospital will be able to better track the status of patient invoices and insurance payment status in a consolidated view.

Data Questions

- ◆ What patients are assigned to which doctors?
- ◆ What is the average stay length of a patient that has been diagnosed with a particular illness?
- ◆ Who is the insurance provider of a given patient?

- ◆ What is the status of a given patient's invoice?
- ◆ What room is a given patient assigned to?
- ◆ How old is a given patient?

Business Rules

- ◆ A patient will have a unique Patient ID.
- ◆ A full description will be provided about each patient including full name, mailing address, phone number, date of birth, gender, insurance information, admittance, and discharge information.
- ◆ A doctor can treat one or more patients.
- ◆ A doctor will have a unique Doctor ID.
- ◆ A patient can be admitted to one hospital at a time.
- ◆ A hospital has a Building ID.
- ◆ A hospital has many rooms.
- ◆ Each room has a Room ID.
- ◆ A patient is assigned to exactly one room.
- ◆ Each room has a specific capacity.
- ◆ A patient has at least one associated invoice.
- ◆ An invoice has an Invoice ID.
- ◆ An invoice contains a total, a date billed, and patient insurance information.
- ◆ Every invoice has a status: paid or un-paid.
- ◆ A payment changes the status of an invoice.

Glossary

A **patient** is a person who is admitted to the hospital.

A **doctor** is a person who treats patients.

A **hospital** is a building in which doctors work and students are admitted.

An **invoice** is bill that is sent to the patient after receiving treatment at a hospital.

The **invoice status** is tagged to an invoice letting us know if it is either *PAID* or *UNPAID*.

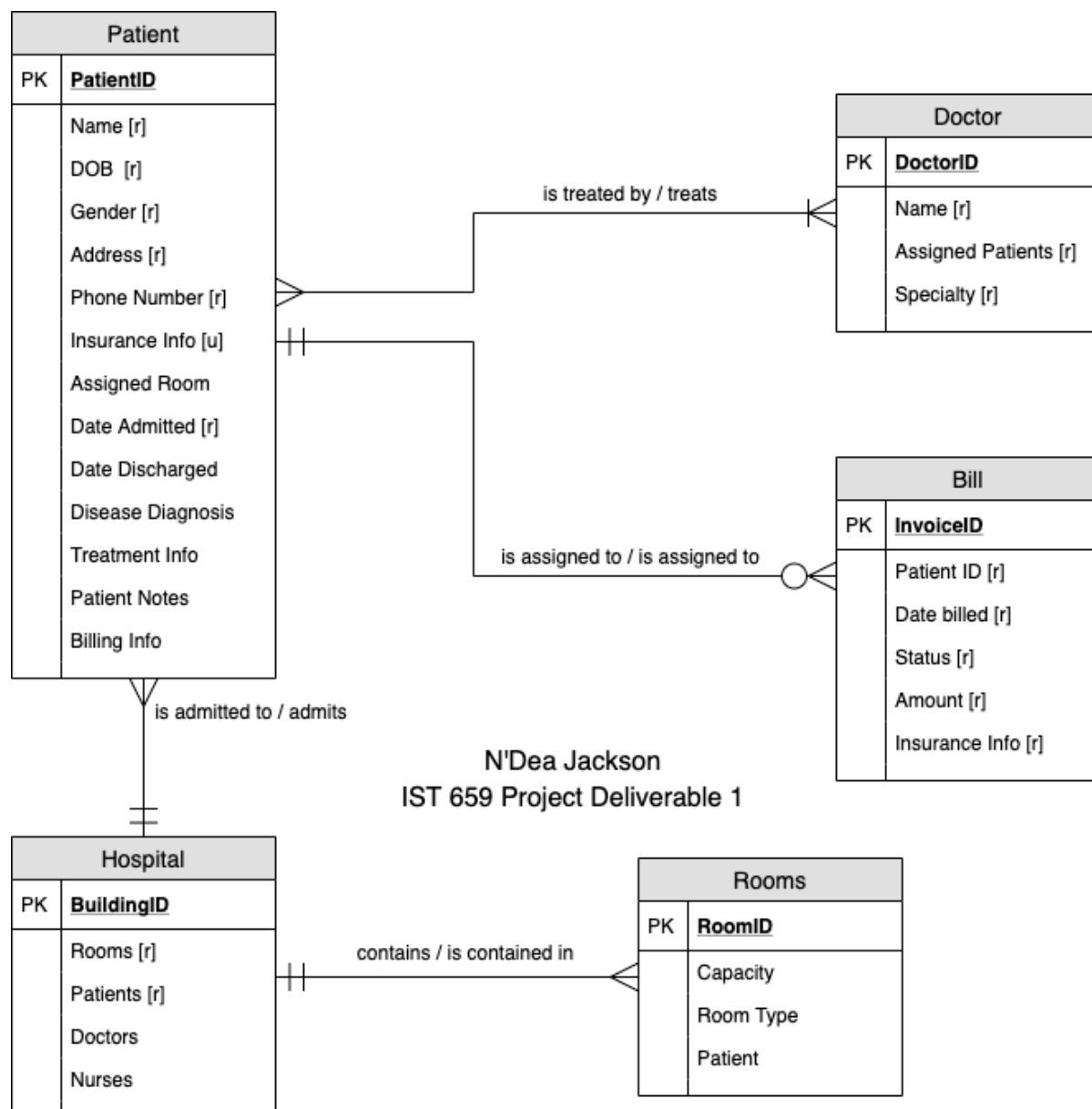
A **PAID invoice status** lets the billing department know when a customer has submitted the payment associated with the invoice.

An **UNPAID invoice status** lets the billing department know when a customer still needs to submit the payment associated with the invoice.

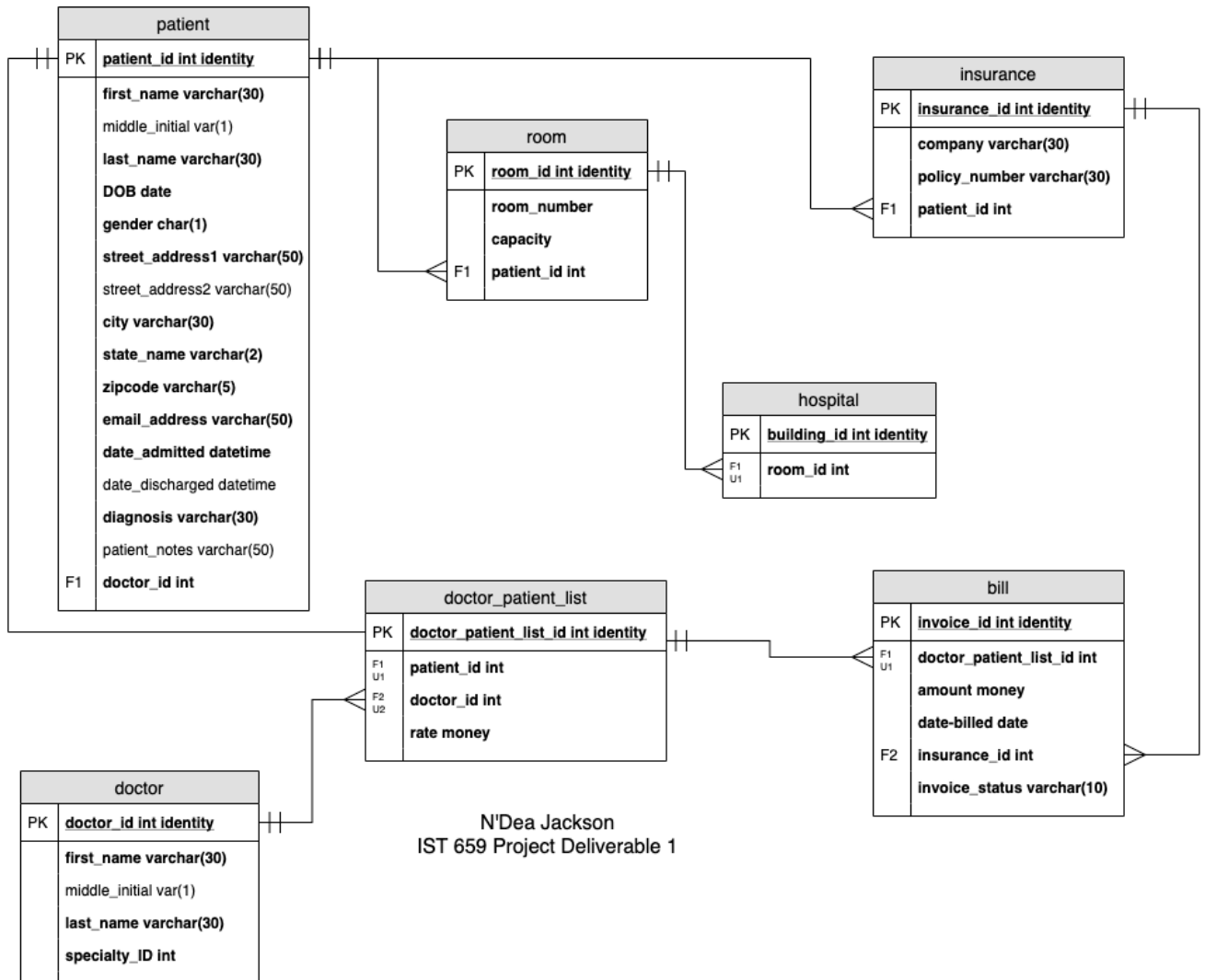
Date admitted refers to the first day of a patient's hospital stay.

Date Discharged refers to the final day of a patient's hospital stay.

Conceptual Model



Normalized Logical Model



Physical Database Design

```
/*
    Author: N'Dea Jackson
    Course: IST659 M402
    Term: January 2020
    Project Deliverable Two
*/
-- Creates a view for patient insurance information
GO
CREATE VIEW PatientInsuranceInfo AS
    SELECT
        patient.first_name AS PatientFirstName,
        patient.middle_initial AS PatientMiddleInitial,
        patient.last_name AS PatientLastName,
        insurance.company AS InsuranceCo,
        insurance.policy_number AS PolicyNumber
FROM insurance
JOIN patient ON insurance.patientID = patient.patientID
GO
SELECT * FROM PatientInsuranceInfo

-- Creates a view for room assignments
GO
CREATE VIEW PatientRoomAssignment AS
    SELECT
        patient.first_name AS PatientFirstName,
        patient.middle_initial AS PatientMiddleInitial,
        patient.last_name AS PatientLastName,
        room.room_number AS RoomNumber,
        room.capacity AS RoomCapacity
FROM room
JOIN patient ON room.patientID = patient.patientID
GO
SELECT * FROM PatientRoomAssignment

-- Creates a view for patient insurance information
GO
CREATE VIEW PatientBillStatus AS
    SELECT
        patient.first_name AS PatientFirstName,
        patient.middle_initial AS PatientMiddleInitial,
        patient.last_name AS PatientLastName,
        bill.invoiceID AS InvoiceID,
        bill.insuranceID AS InsuranceID,
        bill.amount AS InvoiceAmount,
        bill.date_billed AS InvoiceBillDate,
        bill.date_paid AS InvoicePaidDate,
        bill.invoice_status
FROM bill
```

```
JOIN patient ON bill.patientID = patient.patientID
GO
SELECT * FROM PatientBillStatus
```

--DATA CREATION--

--Creating the Doctor table

```
CREATE TABLE doctor(
    --Columns for the Doctor table
    doctorID int identity,
    first_name varchar(30) not null,
    middle_initial char(1),
    last_name varchar(30) not null,
    specialty_ID int not null,
    --Constraints on the Patient table
    CONSTRAINT PK_doctor PRIMARY KEY (doctorID),
    CONSTRAINT U1_doctor UNIQUE (doctorID)
)
```

-- Creating the Patient table

```
CREATE TABLE patient (
    --Columns for the Patient table
    patientID int identity,
    first_name varchar(30) not null,
    middle_initial char(1),
    last_name varchar(30) not null,
    DOB date not null,
    gender char(1),
    street_address1 varchar(50) not null,
    street_address2 varchar(50),
    city varchar(30) not null,
    state_name varchar(2) not null,
    zipcode varchar(5) not null,
    phone_number char(10) not null,
    email_address varchar(50) not null,
    date_admitted datetime not null,
    date_discharged datetime,
    diagnosis varchar(30),
    patient_notes varchar(50),
    doctorID int,
    --Constraints on the Patient table
    CONSTRAINT PK_patient PRIMARY KEY (patientID),
    CONSTRAINT U1_patient UNIQUE (patientID),
    CONSTRAINT FK_doctor FOREIGN KEY (doctorID) REFERENCES doctor(doctorID)
)
```

--Creating the Insurance table

```
CREATE TABLE insurance (
    --Columns for the Insurance table
    insuranceID int identity,
```

```

company varchar(30) not null,
policy_number varchar(30) not null,
patientID int not null,
--Constraints on the Insurance table
CONSTRAINT PK_insurance PRIMARY KEY (insuranceID),
CONSTRAINT U1_insurance UNIQUE (insuranceID),
CONSTRAINT FK_patient FOREIGN KEY (patientID) REFERENCES patient(patientID)
)

--Creating the Doctor Patient table
CREATE TABLE doctor_patient_list (
--Columns for the Doctor Patient table
doctor_patient_listID int identity,
patientID int,
doctorID int,
rate money,
--Constraints on the Insurance table
CONSTRAINT PK_doctor_patient_list PRIMARY KEY (doctor_patient_listID),
CONSTRAINT U1_doctor_patient_list UNIQUE (doctor_patient_listID),
CONSTRAINT FK2_patient FOREIGN KEY (patientID) REFERENCES patient(patientID),
CONSTRAINT FK2_doctor FOREIGN KEY (doctorID) REFERENCES doctor(doctorID)
)

--Creating the Room table
CREATE TABLE room (
--Columns for the Room table
roomID int identity,
room_number int,
patientID int not null,
capacity int not null,
--Constraints on the Room table
CONSTRAINT PK_room PRIMARY KEY (roomID),
CONSTRAINT U1_room UNIQUE (roomID),
CONSTRAINT FK1_patient FOREIGN KEY (patientID) REFERENCES patient(patientID)
)

--Creating the Hospital table
CREATE TABLE hospital (
--Columns for the Hospital table
buildingID int identity,
roomID int not null,
--Constraints on the Hospital table
CONSTRAINT PK_hospital PRIMARY KEY (buildingID),
CONSTRAINT FK1_room FOREIGN KEY (roomID) REFERENCES room(roomID)
)

--Creating the Bill table
CREATE TABLE bill (
--Columns for the Bill table
invoiceID int identity,

```



```

    doctor_patient_listID int,
    amount money not null,
    date_billed date not null,
    date_paid date,
    insuranceID int not null,
    invoice_status varchar(10) not null,
    patientID int not null,
    --Constraints on the Bill table
    CONSTRAINT PK_bill PRIMARY KEY (invoiceID),
    CONSTRAINT FK1_doctor_patient_list FOREIGN KEY (doctor_patient_listID) REFERENCES
doctor_patient_list(doctor_patient_listID),
    CONSTRAINT FK2_insurance FOREIGN KEY (insuranceID) REFERENCES
insurance(insuranceID),
    CONSTRAINT FK3_patient FOREIGN KEY (patientID) REFERENCES patient(patientID)
)

```

--DATA MANIPULATION--

--Insert doctor data values into the doctor table

```

INSERT INTO doctor(first_name, middle_initial, last_name, specialty_ID)
VALUES
    ('Jeffrey', 'L', 'Harper', 101),
    ('Bruce', 'R', 'Wayne', 101),
    ('Mariah', 'Z', 'France', 102),
    ('Jennifer', 'P', 'Son', 103),
    ('Robert', 'K', 'Lesley', 103),
    ('Mark', NULL, 'Manson', 104),
    ('Lillie', 'D', 'Baron', 105)

```

--View values from the doctor table

```
SELECT * FROM doctor
```

--Insert patient data values into the patient table

```

INSERT INTO patient(first_name, middle_initial, last_name, DOB, gender, street_address1,
street_address2, city, state_name, zipcode, phone_number, email_address, date_admitted,
date_discharged, diagnosis, patient_notes, doctorID)
VALUES
    ('Ariel', 'M', 'Love', '05/07/1996', 'F', '123 Rockledge Dr', NULL, 'Fort Washington', 'MD',
'20744', '2024451010', 'alovie@abc.com', '02/24/2020', '02/26/2020', 'Broken Arm', 'Patient has
allergies to Tylenol', 3),
    ('Tytiana', 'A', 'Christmas', '07/16/1997', 'F', '207 Frederick Ave', 'APT 255', 'Silver Spring',
'MD', '20815', '3015556006', 'itschristmastime@gmail.com', '02/28/2020', '02/28/2020', 'Allergic
Reaction', 'Patient has peanut allergy', 1),
    ('Kienna', 'M', 'Morris', '06/25/1996', 'F', '28 Backdoor Ln', NULL, 'Fort Washington', 'MD',
'20744', '2025102728', 'kmorris@gmail.com', '02/24/2020', '02/27/2020', 'Maternity', 'Delivered
healthy baby boy', 5),
    ('Cody', 'E', 'Mitchell', '08/30/1985', 'M', '315 Douglasville Blvd', NULL, 'Clinton', 'MD',
'20735', '4107160914', 'pcmitchell@scg.gov', '03/01/2020', '03/01/2020', 'Sun Burn', 'Large sun
burnt area on upper back and shoulders', 2),

```

('Scott', NULL, 'Disick', '12/19/1999', 'M', '824 Pine Oak Dr', NULL, 'Brandywine', 'MD',
 '20613', '3017099956', 'ilovekardashians@yahoo.com', '03/01/2020', '03/02/2020', 'Asthma
 Attack', 'Held overnight to monitor', 1),
 ('Charles', 'A', 'McCray', '09/09/1971', 'M', '9055 Ewing St', 'APT 7', 'Fort Washington', 'MD',
 '20744', '2021052563', 'camccray@gmail.com', '02/20/2020', '03/02/2020', 'Pneumonia', 'Needs
 follow up appy. scheduled', 4),
 ('Kevin', NULL, 'Smith', '03/22/1984', 'M', '4 Cherry Blossom Ave', NULL, 'Oxon Hill', 'MD',
 '20744', '3019715000', 'kevinsmith@nbc.com', '02/28/2020', '02/28/2020', 'Stiff Neck', 'Patient
 involved in a car accident', 6),
 ('Jason', NULL, 'Webster', '02/13/2008', 'M', '56 Arlington Ct', NULL, 'Clinton', 'MD', '20735',
 '4103348913', 'whatwouldjd@gmail.com', '03/01/2020', '03/03/2020', 'Asthma Attack', 'Needs
 inhaler prescription', 2),
 ('Jermaine', 'L', 'Cole', '10/29/1967', 'M', '2014 Forest Hills Dr', NULL, 'Waldorf', 'MD',
 '20601', '2407037081', 'alovie@abc.com', '03/02/2020', '03/04/2020', 'Fever', NULL, 4),
 ('Jimmy', NULL, 'Fallon', '06/04/2012', 'M', '1776 Independence St', NULL, 'Washington',
 'DC', '20003', '2020192018', 'tonightshow@nbc.com', '03/04/2020', '03/04/2020', 'Allergic
 Reaction', 'Patient has allergies to -cillin drugs', 1),
 ('Katherine', 'S', 'Peterson', '11/17/1993', 'F', '453 Macaulife Cir', NULL, 'Oxon Hill', 'MD',
 '20744', '2021014467', 'petersonk@yahoo.com', '02/28/2020', '02/29/2020', 'Maternity', NULL, 7),
 ('Jamaal', 'K', 'Andrews', '02/19/1998', 'M', '722 Summit Ave', NULL, 'Waldorf', 'MD',
 '20601', '2404455102', 'mrandrews2u@aol.com', '02/27/2020', '03/01/2020', 'Broken Arm',
 'Surgery required', 3),
 ('Hilary', 'V', 'Banks', '04/05/2006', 'F', '115 Bel-Air Rd', NULL, 'Temple Hills', 'MD', '20757',
 '2409920032', 'freshprincess@outlook.com', '02/24/2020', '02/24/2020', 'Gash on head', 'Stitches
 required', 6),
 --A null Discharged Date indicates that the patient is still admitted to the hospital ('Dwayne', 'X',
 'Snipes', '01/03/1997', 'M', '3111 Diggs Ter', NULL, 'Suitland', 'MD', '20746', '2403021672',
 'dwsni@yahoo.com', '02/23/2020', NULL, 'Coronavirus', 'Quarantine! Highly Contagious', 4),
 ('Cody', 'W', 'Carrington', '05/11/1950', 'M', '1911 Just Pl', NULL, 'Alexandria', 'VA', '22304',
 '7032173032', 'colemanlove@gmail.com', '03/04/2020', '03/05/2020', 'Chest Pain', 'Kept
 overnight to monitor', 6),
 ('Malik', 'P', 'Jones', '09/27/1988', 'M', '708 17th St', 'Suite 2300', 'White Plains', 'MD',
 '20695', '2402154113', 'hoopdreams47@yahoo.com', '02/29/2020', '02/29/2020', 'Rash', NULL,
 2),
 ('Michaela', 'A', 'Nelson', '05/13/2018', 'F', '6291 Ivy Pk', 'APT 6', 'Brandywine', 'MD',
 '20613', '3019920345', 'nellie1908@aol.com', '03/04/2020', '03/05/2020', 'Abdominal Pain',
 NULL, 6),
 ('Olivia', NULL, 'Benson', '01/01/1979', 'F', '334 14th St', 'Unit 5', 'Accokeek', 'MD', '20601',
 '2408028041', 'bensono@svu.gov', '02/21/2020', '02/24/2020', 'Flu', 'Tested for Coronavirus', 5),
 ('William', 'D', 'Campbell', '04/20/2000', 'M', '458 Girard Ln', NULL, 'Fort Washington', 'MD',
 '20744', '2025591855', 'wcampbell@abc.com', '03/04/2020', '03/05/2020', 'Concussion', NULL, 6)

--View values from the patient table

SELECT * FROM patient

--Update Olivia Benson's Email address using stored procedure

EXEC ChangeEmail 'Benson', 'bensolivia@svu.usa'

SELECT * FROM patient WHERE last_name = 'Benson'

--Insert room data values into the room table

```

INSERT INTO room(patientID, room_number, capacity)
VALUES
    (1, 207, 2), (2, 203, 2), (3, 419, 1), (4, 101, 1), (5, 306, 2),
    (6, 610, 1), (7, 110, 1), (8, 306, 2), (9, 412, 1), (10, 203, 2),
    (11, 427, 1), (12, 207, 2), (13, 104, 1), (14, 605, 1), (15, 123, 1),
    (16, 108, 4), (17, 401, 1), (18, 620, 1), (19, 117, 2)
SELECT * FROM room

```

--Insert insurance data values into the insurance table

```

INSERT INTO insurance(patientID, company, policy_number)
VALUES
    (1, 'Anthem', 'EZ11074'), (2, 'Wellcare', 'WC30912'),
    (3, 'Metropolitan', 'MP61738'), (4, 'Anthem', 'EZ90210'),
    (5, 'Wellcare', 'WC20845'), (6, 'Blue Cross', 'BC02396'),
    (7, 'Kaizer Permanente', 'KP14902'), (8, 'Independent Health', 'IH70229'),
    (9, 'Blue Cross', 'BC31846'), (10, 'United Health', 'UH10048'),
    (11, 'United Health', 'UH19472'), (12, 'Blue Cross', 'BC46291'),
    (13, 'Metropolitan', 'MP19920'), (14, 'Anthem', 'EZ40573'),
    (15, 'Metropolitan', 'MP51703'), (16, 'Centene Corp', 'CC30091'),
    (17, 'Wellcare', 'WC02358'), (18, 'Independent Health', 'IH81047'),
    (19, 'Blue Cross', 'BC15167')

```

--View values from the insurance table

```

SELECT * FROM insurance

```

--Insert billing data values into the bill table

```

INSERT INTO bill (patientID, date_billed, insuranceID, date_paid, invoice_status, amount)
VALUES
    (1, '02/27/2020', 102, '3/15/2020', 'PAID', '$3,000'),
    (2, '02/29/2020', 103, '3/05/2020', 'PAID', '$500'),
    (3, '02/28/2020', 104, NULL, 'UNPAID', '$6,000'),
    (4, '03/02/2020', 105, '3/02/2020', 'PAID', '$200'),
    (5, '03/02/2020', 106, '3/10/2020', 'PAID', '$1,000'),
    (6, '03/03/2020', 107, NULL, 'UNPAID', '$10,000'),
    (7, '02/29/2020', 108, '3/1/2020', 'PAID', '$250'),
    (8, '03/04/2020', 109, NULL, 'UNPAID', '$2,000'),
    (9, '03/05/2020', 110, '3/20/2020', 'PAID', '$3,500'),
    (10, '03/05/2020', 111, '3/15/2020', 'PAID', '$600'),
    (11, '03/01/2020', 112, NULL, 'UNPAID', '$5,000'),
    (12, '03/02/2020', 113, '3/26/2020', 'PAID', '$4,500'),
    (13, '02/25/2020', 114, '3/16/2020', 'PAID', '$600'),
    (14, '02/27/2020', 115, NULL, 'UNPAID', '$4,300'),
    (15, '03/06/2020', 116, '3/28/2020', 'PAID', '$1,000'),
    (16, '03/01/2020', 117, '3/06/2020', 'PAID', '$650'),
    (17, '03/06/2020', 118, '3/06/2020', 'PAID', '$1,000'),
    (18, '02/25/2020', 119, '3/15/2020', 'UNPAID', '$4,100'),
    (19, '03/05/2020', 120, '3/07/2020', 'PAID', '$900')

```

--View values from the bill table

```

SELECT * FROM bill

```

-- Answering questions --

--Question 1: What patients are assigned to Mark Manson?

```
SELECT * FROM DoctorsAndPatients WHERE DoctorLastName = 'Manson'
```

--Question 2:

```
SELECT
    patient.patientID,
    patient.first_name + ' ' + patient.middle_initial + ' ' + patient.last_name AS
PatientName,
    (DateDiff(day, date_admitted, date_discharged)) AS DaysInHospital
FROM patient
ORDER BY patient.last_name
```

--Question 3: Who is the insurance provider of Kienna Morris?

```
SELECT * FROM PatientInsuranceInfo WHERE PatientLastName = 'Morris'
```

--Question 4: What is the status of Jason Webster's invoice?

```
SELECT * FROM PatientBillStatus WHERE InvoiceID = 8
```

--Question 5: What room is Jermaine Cole patient assigned to?

```
SELECT * FROM PatientRoomAssignment WHERE PatientLastName = 'Cole'
```

-- Question 6: How old is Cody Carrington?

```
SELECT
    patient.patientID,
    patient.first_name AS PatientFirstName,
    patient.middle_initial AS PatientMiddleInitial,
    patient.last_name AS PatientLastName,
    (DateDiff(year, DOB, GETDATE())) AS AGE
FROM patient WHERE patient.last_name = 'Carrington'
```

Answering Data Questions:

Question 1: What patients are assigned to Mark Manson?

SELECT * FROM DoctorsAndPatients **WHERE** DoctorLastName = 'Manson'

	PatientFirstName	PatientMiddleInitial	PatientLastName	AssignedDoctor	DoctorFirstName	DoctorMiddleInitial	DoctorLastName
1	Kevin	NULL	Smith	6	Mark	NULL	Manson
2	Hilary	V	Banks	6	Mark	NULL	Manson
3	Cody	W	Carrington	6	Mark	NULL	Manson
4	Michaela	A	Nelson	6	Mark	NULL	Manson
5	William	D	Campbell	6	Mark	NULL	Manson

Question 2:

Question 3: Who is the insurance provider of Kienna Morris?

SELECT * FROM PatientInsuranceInfo **WHERE** PatientLastName = 'Morris'

	PatientFirstName	PatientMiddleInitial	PatientLastName	InsuranceCo	PolicyNumber
1	Kienna	M	Morris	Metropolitan	MP61738

Question 4: What is the status of Jason Webster's invoice?

SELECT * FROM PatientBillStatus **WHERE** InvoiceID = 8

	PatientFirstName	PatientMiddleInitial	PatientLastName	InvoiceID	InsuranceID	InvoiceAmount	InvoiceBillDate	InvoicePaidDate	invoice_status
1	Jason	NULL	Webster	8	109	2000.00	2020-03-04	NULL	UNPAID

Question 5: What room is Jermaine Cole patient assigned to?

SELECT * FROM PatientRoomAssignment **WHERE** PatientLastName = 'Cole'

	PatientFirstName	PatientMiddleInitial	PatientLastName	RoomNumber	RoomCapacity
1	Jermaine	L	Cole	412	1

Question 6: How old is Cody Carrington?

SELECT

patient.patientID,
patient.first_name AS PatientFirstName,
patient.middle_initial AS PatientMiddleInitial,
patient.last_name AS PatientLastName,
(DateDiff(year, DOB, GETDATE())) AS AGE

```
FROM patient WHERE patient.last_name = 'Carrington'
```

Results		Messages			
	patientID	PatientFirstName	PatientMiddleInitial	PatientLastName	AGE
1	15	Cody	W	Carrington	70

Implementation

Reflection

When I started the project, I assumed that the conceptual and logical models that I had were as fleshed out as they needed to be. As I began coding and setting up the tables in the database, I began to realize that I was missing some important information and had to periodically update my diagrams as I updated my database. While I had difficulty creating some of the diagrams earlier on in the course, I found them extremely helpful with structuring my hospital database. The next time I do this, I would like to be able to add security permissions. I was unable to create separate users because I did not have the permissions on the remote lab, but if I was able to, I would have created separate users for doctors/nurses, the finance department, and the receptionist. The doctors and nurses would have the same view upon logging in and would be able to see patient diagnosis and health history. The finance department would be able to see the billing and insurance information of patients as well as their mailing information, but not their health history. The receptionist view would show the room assignments of the patients and keep a log of their visitors.

Regardless to whether I go through these exact steps again, I think that doing this project will inform my approach to data as an information professional by first teaching me to map everything out first. This project has also shown be the importance of breaking data down into the lowest levels (no multi-value attributes) and creating views so that you can get much more information out of your data. It has also shown me how to connect multiple platforms together, which could be very useful in my career as a data scientist. I have coded

before but my favorite and the easiest to learn so far for me is SQL. I hope to be able to do some more in-depth database coding in the future.

Summary

The objective of this project was to create a hospital database management system for Jackson Memorial Hospital. The system was going to be used to keep track of patient and doctor information in a more efficient manner than their current method. The current method of data storing included multiple siloed applications and physical paper charts. After the creation of this database, hospital staff are now able to use one system that allows them the ability to view all of the information pertinent to their job role in one place. Some of the functionality of the database includes the ability to look up room assignments, patient-doctor assignments, and also view a patient's invoice status. Overall, implementing this database allowed Jackson Memorial Hospital to improve the healthcare experience for both their valued patients and staff members as well as develop a better way to manage their data.

By first creating the conceptual and logical models, I was able to create business rules for my hospital management system database that governed the code for database creation. Though the models and business rules are important for the creation of the physical database, this does not mean that they cannot be amended throughout the coding process as needed. If there was anything about this process that I would have done differently, it would have been to pick a project that had readily available data. Because this was a fictitious hospital, creating 19 unique patient records was a bit time consuming.