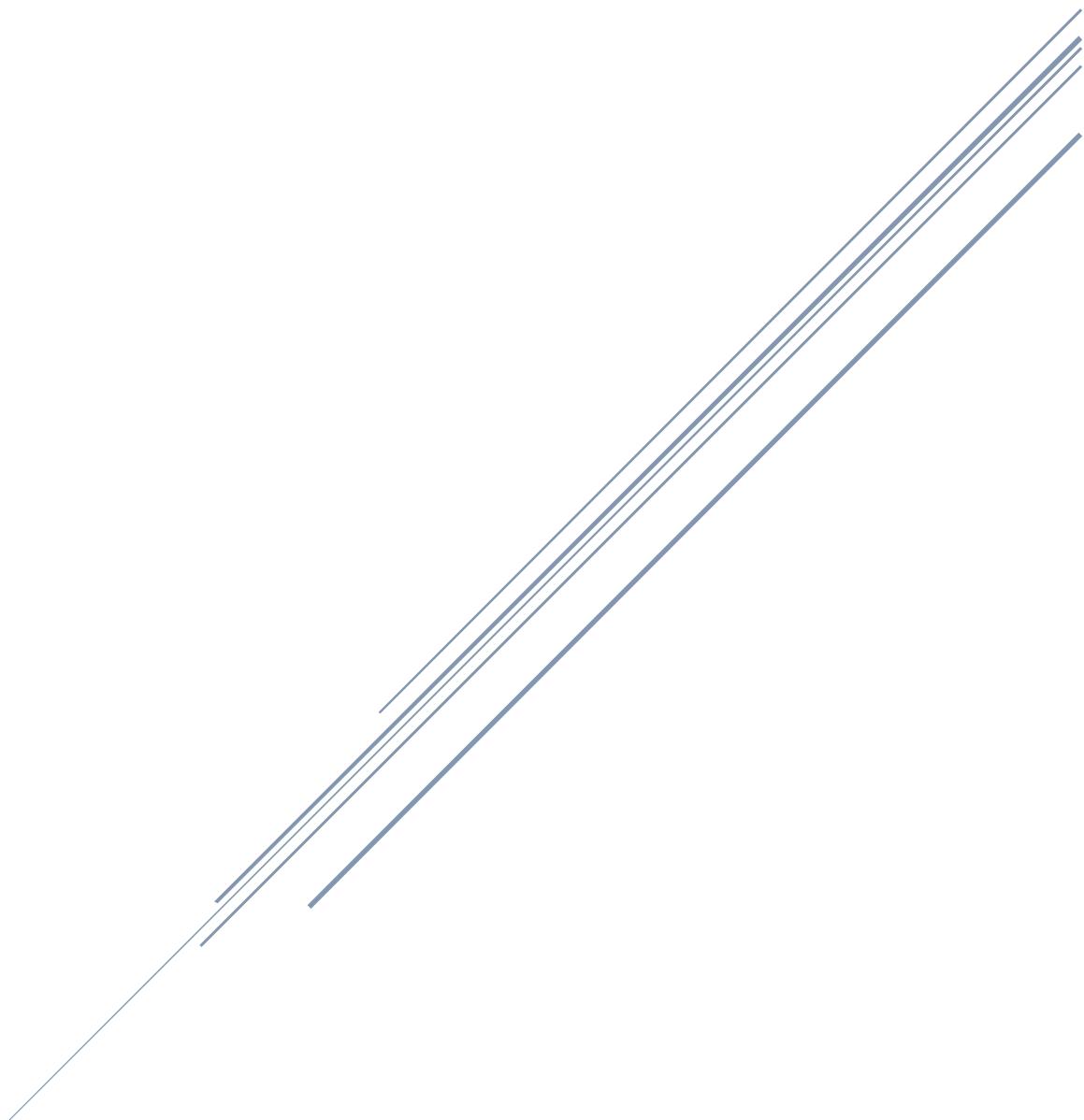


# EZMEDICALGROUP DATABASE APPLICATION DESIGN & IMPLEMENTATION

Hospital Management System



BY: Steve Carchi

## **Executive Summary**

In this project the purpose is to design & implement a database application for a Hospital Management System Application named EZMedicalGroup.com. The application is to automate the managing of patients and hospital personnel activities etc. This Hospital Management System is designed to allow hospital employees to conveniently manage hospital patient activities such as assigning physicians, medical care, registration etc., using this system. The system is a combination of both the Three-tiered Web and Two-tiered Windows client/server that will share the same Database Tier. To have this system the methodology being used will be Planning, Analysis, Design, Implementation, and Maintenance/Administration. The tables for the database will be done through this, the model diagrams, the data dictionaries, the development and implementation, development and implementation physical schema diagram, and development and implementation will also be done through this. In the upgrade done to the Database tier in the Hospital Management System Application named EZMedicalGroup.com, we followed the Project Management Methodology using the Database Application Development Lifecycle (Planning, Analysis, Design, Development/Implementation & Operations/Maintenance) to create business reports and improve the performance of the full working application by creating store procedures.

## **Problem Objectives**

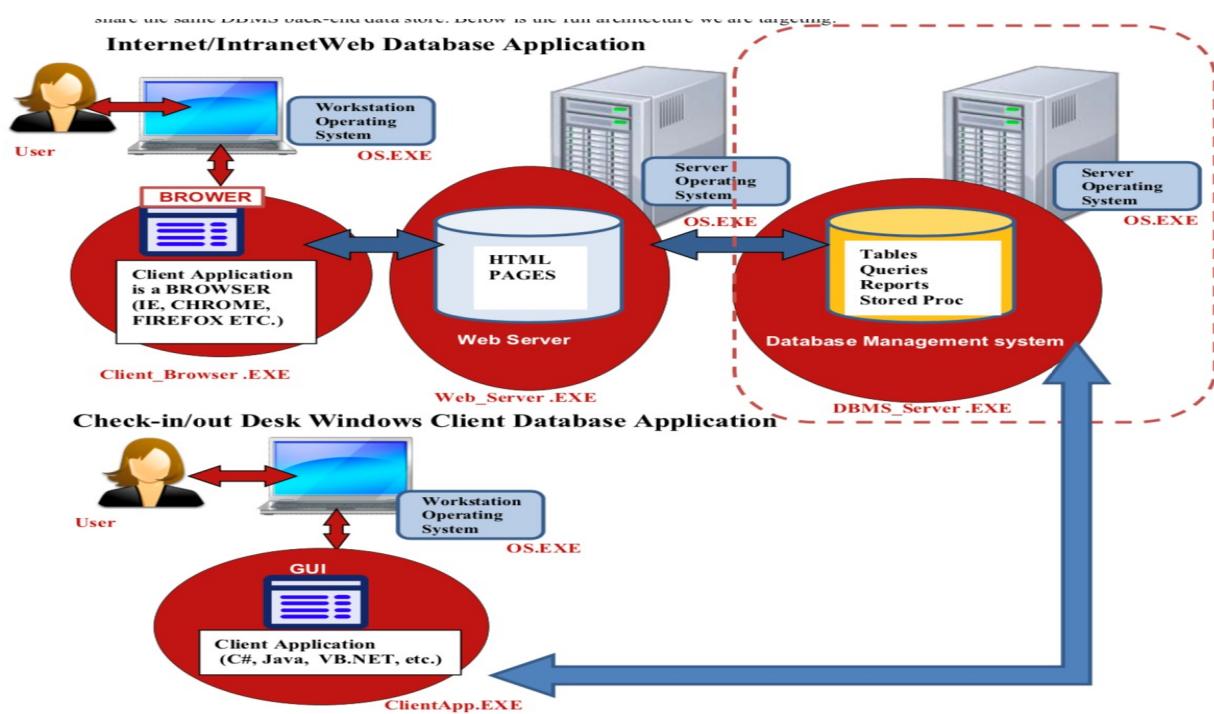
To design & implement a database application for a Hospital Management System Application named EZMedicalGroup.com. The application is to automate the managing of patients and hospital personnel activities etc. Basic objectives and architecture being targeted:

This Hospital Management System is designed to allow hospital employees to conveniently manage hospital patient activities such as assigning physicians, medical care, registration etc., using this system. The applications architecture being targeted is as follows:

- o Three-Tiered Web-based Client/Server – for customers Physicians, nurses and other medical professionals to manage the day activities in the hospital online via a browser.
  - o Two-Tiered Windows-Client Client/Server – Front line workers such as emergency room administrators, nurses, registration clerks etc., using the client application for speed and performance.
  - o Database Tier – both the Three-tiered Web and Two-tiered Windows client/server will share the same Database Tier.
- The application is to be designed to support dozens of major cities around the world. In addition, provide a great user experience in Hospital Reservations.

## Technical Application Architecture

- The system is a combination of 1) Three-Tiered Web Database Application with a web-based browser front-end for the users that want to use/employee that require the use of a browser-based application. In addition, we have an additional architecture: 2) Two-Tiered Windows Database Application, with a custom client front-end to for the application used by check-in desk at the hospital emergency and other location that require fast performance. Both these architecture (Web & Custom client) share the same DBMS back-end data store. Below is the full architecture being targeting:



## Project Management Methodology

- The following Project Management will be used for the Hospital Management System.
- Below is a listing of all the 5 PHASES & their main deliverables:



- High-level description of this methodology:
  - ▪ **Planning** – Plan the entire project from a high-level. Gather requirements from business. Decide on methodology etc.
  - ▪ **Analysis** – Analyze the business or organization in order to determine requirements for developing database. Deep analysis of requirements.
  - ▪ **Design** – Organize all the requirements and produce a detailed specification of all data, forms, reports, displays, and processing or business rules. This includes purchase of all hardware, software etc.
  - ▪ **Implementation** – Create the database, tables, view, stored procedures etc. Create Front-End Client programs. Test and install system, train users and document the installation and Standard Operation Procedures (SOP)
  - ▪ **Maintenance\Administration** – Monitor, backup & repair the system. Periodically audit system to ensure it continues to meet requirements. Have a disaster/recovery system in place.

## Business Requirements

In this section the business requirements are stated. All the information in here is what is wanted in the database.

- A Business Analyst was hired by Mr. Rodriguez to compile the list of requirements based on the results of interviews and conversations with the various business stakeholders.
- Below are the requirements captured by the Business Analyst:

- This EZMedicalGroup.com Hospital Management System is designed to allow hospital physicians, nurses and employees to manage outpatients and resident patient hospital activities.
- The application is to support our hospitals in major cities in the US and around the world. Both the client application and the website need to capture the following information:
  - o As a large service organization, EZ Medical Group depends on four major groups of persons for its continued success: employees, physicians, patients, and volunteers. Also note that a small number of persons in the hospital community do not belong to any of these four groups. A particular person may belong to two (or more) of these groups at a given time, for example, a volunteer or employee may also be a patient at the hospital at some point in time or an employee also be volunteer on their spare time.
  - o For our patients we need to store their person ID or identifier for that person, name composed of first, middle & last name. Also, address which include the following components: address, city, state, zip code and country. In addition, gender, contact phone & email. Additional attributes of patient include first contact date or date when patient first engaged with the hospital, emergency contact name, emergency contact number and the final attribute for a patient is a credit card. We accept major credit cards for customers to pay for their co-pay visit or services rendered. Therefore, we need to capture the following credit card information: credit card number, credit card merchant name such as AMX, Visa, Master Card etc., credit card owner name and credit card expiration date. Note that a patient can have more than one credit card, therefore, a patient can use any of the many types of major credit card he or she owns to pay for services or co-pay. In addition, the credit card can be owned by the patient, spouse or corporation who co-owns the credit card.
  - o For our physicians we need to store their person ID or identifier, name composed of first, middle & last name. Also, address composed of the following components: address, city, state, zip code and country. In addition, gender, contact phone & email. Other attributes of physicians are pager number and a DEA number (a unique DEA registration number from the Drug Enforcement Administration for a physician to be able to prescribe controlled substances).
  - o We have four main types of employees we need to track: RN Nurses, LPN Nurses, staff and technicians. In addition, to other types of employees that are of no interest now. For all four types of employees, we required the following information: person ID or identifier, name composed of first, middle & last name. Also, address which include the following components: address, city, state, zip code and country. In addition, gender, contact phone & email, in addition to the date hired.
  - o Finally, for our volunteer we are interested in capturing their person ID, name composed of first, middle & last name, address which include the following components: address, city, state, zip code and country, gender, contact phone, email, their many skills, interest ID which is an identifier we use to represent different interest, and finally interest description. The table below shows example of the interest id we use and matching description.

<i>Interest ID</i>	<i>Interest Description</i>
1	<b>Emergency Room</b>
2	<b>Patient Ward</b>
3	<b>Administrative work</b>
4	<b>Radiology work</b>
Etc..	<b>Etc..</b>

- o A care center is a treatment location within our hospitals. Example of care centers are maternity, emergency room, multiple sclerosis center, ambulatory, outpatient surgery, diagnostic services etc. Each care center has a care center ID, care center name and care center address.
- o We have two types of patients, outpatient and resident patient. for both outpatient & resident patient we need to store their person ID, name composed of first, middle & last name, address which include the following components: address, city, state, zip code and country. In addition, gender, contact phone & email. Other attributes patient includes first contact date or date when patient first engaged with the hospital, emergency contact name, emergency contact number, credit cards information: credit card number, credit card merchant name, credit card owner name and credit card expiration date.
- o What distinguish an outpatient is the attribute checkback date since an outpatient may come in for many reasons, including routine examinations at an outpatient care center. Each outpatient is scheduled for zero or more visits. A visit has several attributes: visit number which is a unique ID, visit date and visit time. Note that an instance of visit cannot exist without an outpatient instance. A visit must be scheduled for one or more care center. A care center can be visited by many outpatients or none.
- o For resident patients, we need to store the date admitted and the date discharged since resident patients stay in the hospital for a period of time. In our hospital we only have these two types of patients' outpatient and resident patient no other type of patient exists. A patient cannot be an outpatient and resident patient at the same time.
- o Each resident patient must be assigned a bed. Because the hospitals don't always fill all its beds, a bed may or may not have a resident patient assigned to it at a given time.
- o A care center can contain one or more beds (up to any number), but there are care center that do not have beds, and a bed must belong to at least one care center. Note that as with visit, without a care center a bed cannot exist. For our beds, we need to store their Bed ID which is composed of two components: Bed number and room number. In addition, we use a code to identify the bed type or bed type ID and a description for each bed type ID. The table below lists some of the bed type IDs and their descriptions.

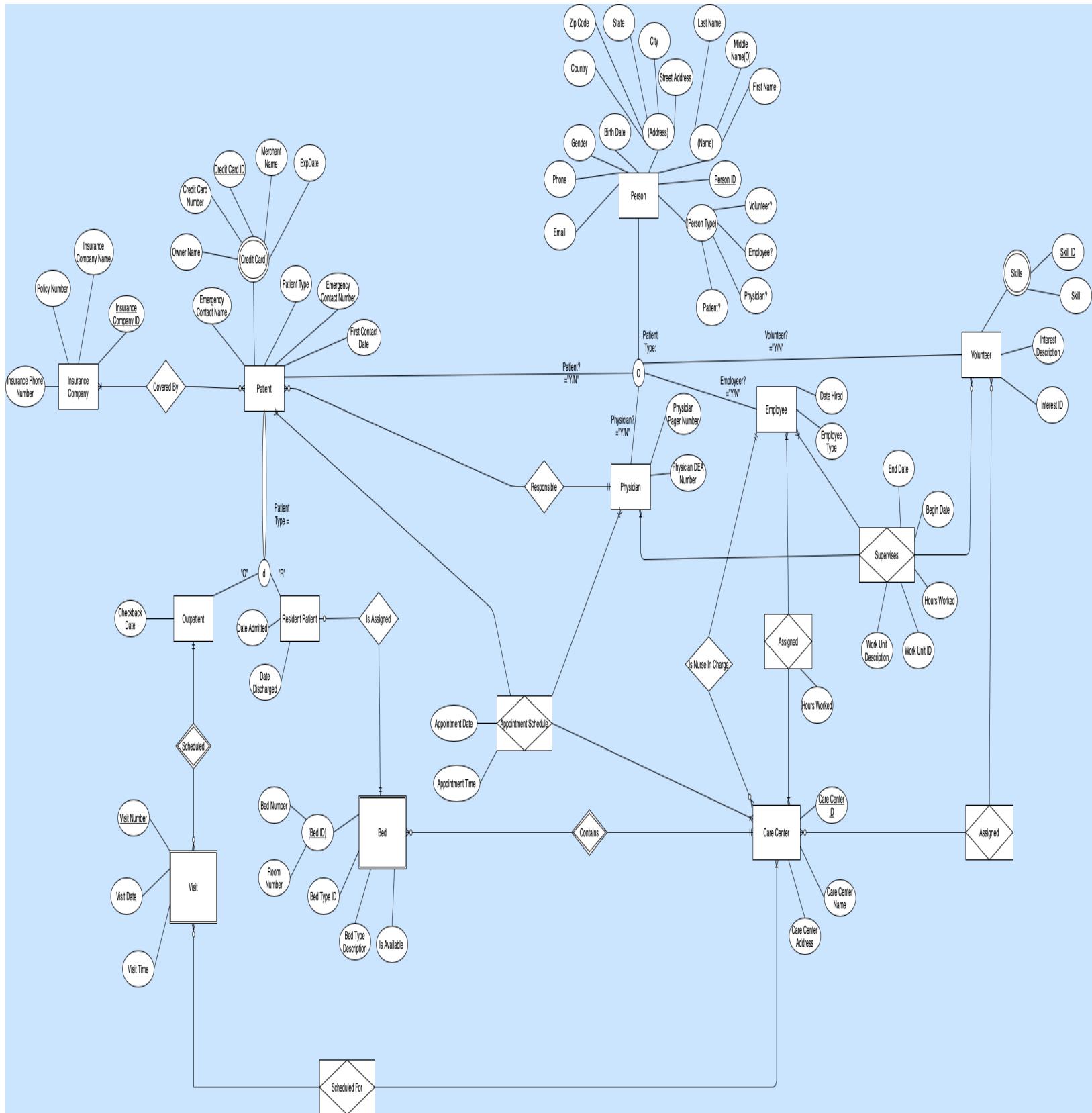
<i>Bed Type ID</i>	<i>Description</i>
1	<b>Standard Manual Bed</b>
2	<b>Electric Automatic Bed</b>
3	<b>Clinitron Air Bed</b>
4	<b>Stretcher</b>
Etc..	<b>Etc..</b>

- o Each patient has one and only one physician responsible for that patient. A given physician may not be responsible for a patient at a given time or may be responsible for one or more patients.
- o We also need to capture the insurance information for all patients. For insurance company, we need to store the insurance company ID, insurance company name, policy number, insurance company phone number. A patient can be covered by many insurance companies and insurance companies can cover many patients. Nevertheless, a patient must be covered by at least one insurance company.
- o Our employees must be assigned to a care center. Employees can be assigned to many care centers and many care centers assigned employees, but a care center must have at least one employee assigned. When an employee is assigned to a care center, we need to capture the hours worked.
- o Each care center is supervised by an employee who is the nurse in charge. A nurse in charge can supervise none or more care center, but a care center can only be supervised by one nurse in charge.
- o Appointments are captured for the patients that decide to schedule appointment(s) for the care centers they need to go to. We need to capture the day and time of the appointment scheduled as well as the physician assigned.
- o Each volunteer is supervised by an employee and a physician. Supervision requires participation of all three for a volunteer to be able to work. Note that not all employees and physicians supervise volunteer, and a volunteer can be supervised by one or more physicians and employee, but a volunteer must be supervised. When an employee(s) and physician(s) supervise a volunteer, we need to keep track of the begin date or date volunteer began to work and the end date of the job. In addition, the number of hours worked by the volunteer, the Unit ID or code we use for identifying a care center unit for volunteers to work and the unit description. The table below shows example of the code we use and the description.

<i>Unit ID</i>	<i>Unit Description</i>
1	<b>Emergency Room</b>
2	<b>Outpatient Care</b>
3	<b>Radiology</b>
4	<b>Pediatrics</b>
Etc..	<b>Etc..</b>

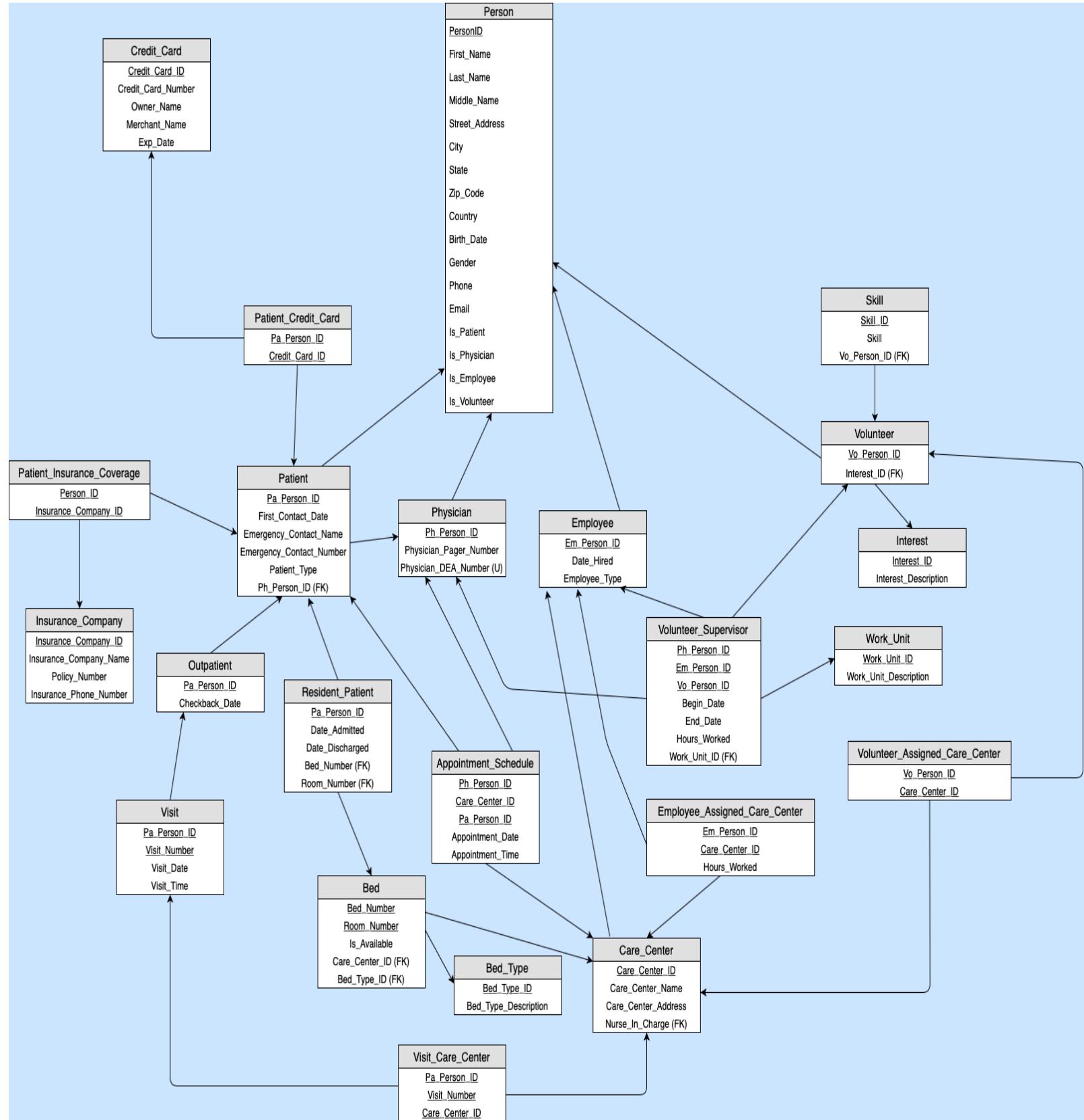
## ER/EER Conceptual Model

In this section the EER Conceptual Model is shown. This shows the Entities and attributes that are going to be created in the database.



## Normalized Logical Model

In this section the Normalized Logical Model is shown. This shows the entity tables and the attribute tables in it. Also showing primary keys and foreign keys.



## Physical Model Data Dictionary

In this section the data dictionary is shown. This shows the entities, attributes, and metadata with all information provided. This was done for all the entities.

PERSON						
Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
PERSON ID	Numeric	INT	Y	Default size of INT data type	NOT NULL IDENTITY(1111,1) PRIMARY KEY	Auto-generated unique identifier for person instance starting at 1111.
FIRST NAME	String	NVARCHAR(X)	Y	50	NOT NULL	Person first name
LAST NAME	String	NVARCHAR(X)	Y	50	NOT NULL	Person last name
MIDDLE NAME	String	NVARCHAR(X)	N	50	NULL	Person middle name(optional)
STREET ADDRESS	String	NVARCHAR(X)	Y	100	NOT NULL	Person address
CITY	String	NVARCHAR(X)	Y	60	NOT NULL	Person address city/town
STATE	Character	CHAR(X)	Y	2	NOT NULL	Person address state
ZIP CODE	String	VARCHAR(X)	Y	12	NOT NULL	Person address postal/zip code
COUNTRY	String	VARCHAR(X)	Y	60	NOT NULL	Person address country
BIRTH DATE	Date	DATE	Y	YYYY/MM/DD	NOT NULL	Person's Date of Birth
GENDER	Character	CHAR(X)	Y	1	NOT NULL	Person's gender
PHONE	String	VARCHAR(X)	Y	20	NOT NULL	Person's phone number
EMAIL	String	VARCHAR(X)	Y	50	NULL	Person's email
ISPATIENT	Character	CHAR(X)	Y	1	NOT NULL	Is the person a Patient(Y/N)
ISPHYSICIAN	Character	CHAR(X)	Y	1	NOT NULL	Is the person a Physician(Y/N)
ISEMployee	Character	CHAR(X)	Y	1	NOT NULL	Is the person an Employee(Y/N)
ISVOLUNTEER	Character	CHAR(X)	Y	1	NOT NULL	Is the person a Volunteer(Y/N)

INSURANCE COMPANY						
Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
INSURANCE COMPANY ID	Numeric	INT	Y	Default size of INT data type	NOT NULL IDENTITY(1111,1) PRIMARY KEY	Auto-generated unique identifier for insurance company instance starting at 1111.
INSURANCE COMPANY NAME	String	VARCHAR(X)	Y	40	NOT NULL	Insurance company's name
POLICY NUMBER	String	VARCHAR(X)	Y	15	NOT NULL	Insurance company policy number
INSURANCE PHONE NUMBER	String	VARCHAR(X)	Y	20	NOT NULL	Insurance company phone number

PATIENT INSURANCE COVERAGE						
Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
PERSON ID	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for person instance. Foreign key for patient entity
INSURANCE COMPANY ID	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for insurance company instance. Foreign key for insurance company entity

CREDIT CARD						
Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
CREDIT CARD ID	Numeric	INT	Y	Default size of INT data type	NOT NULL IDENTITY(1,1) PRIMARY KEY	Auto-generated unique identifier for credit card instance starting at 1.
CREDIT CARD NUMBER	String	VARCHAR(X)	Y	128	NOT NULL	credit card number of owner card
OWNER NAME	String	NVARCHAR(X)	Y	75	NOT NULL	Owner name of the credit card
MERCHANT NAME	String	VARCHAR(X)	Y	20	NOT NULL	Name of the bank company
EXPDATE	Date	DATE	Y	YYYY/MM/DD	NOT NULL	Expiration date of credit card

PATIENT CREDIT CARD						
Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
PA PERSON ID	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for patient person instance. Foreign key for patient entity
CREDIT CARD ID	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for credit card instance. Foreign key for credit card entity

PHYSICIAN						
Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
PH PERSON ID	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for Physician person instance. Foreign key for person entity
PHYSICIAN PAGER NUMBER	String	VARCHAR(X)	Y	20	NOT NULL	Physician pager number
PHYSICIAN DEA NUMBER	Character	CHAR(X)	Y	9	NOT NULL UNIQUE	Unique physician DEA number

PATIENT						
Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
PA PERSON ID	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for patient person instance. Foreign key for person entity
FIRST CONTACT DATE	Date	DATE	Y	YYYY/MM/DD	NOT NULL	Patient's first time attended
EMERGENCY CONTACT NAME	String	NVARCHAR(X)	Y	75	NOT NULL	Patient's emergency contact name
EMERGENCY CONTACT NUMBER	String	VARCHAR(X)	Y	20	NOT NULL	Patient's emergency contact number
PATIENT TYPE	Character	CHAR(X)	Y	1	NOT NULL	Patient type (R/O)
PH PERSON ID	Numeric	INT	Y	Default size of INT data type	NOT NULL FOREIGN KEY	ID of Physician who is treating patient (Foreign Key of Physician Table)

OUTPATIENT						
Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
PA PERSON ID	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for patient person instance. Foreign key for patient entity
CHECKBACK DATE	Date	DATE	Y	YYYY/MM/DD	NOT NULL	When patient comes back

RESIDENT PATIENT						
Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
PA PERSON ID	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for patient person instance. Foreign key for patient entity
DATE ADMITTED	Date	DATE	Y	YYYY/MM/DD	NOT NULL	Date when patient came in
DATE DISCHARGED	Date	DATE	Y	YYYY/MM/DD	NOT NULL	Date when patient got released
BED NUMBER	Numeric	INT	Y	Default size of INT data type	NOT NULL FOREIGN KEY	ID of Bed Number which is being occupied by patient (Foreign Key of Bed Table)
ROOM NUMBER	Numeric	INT	Y	Default size of INT data type	NOT NULL FOREIGN KEY	ID of Room Number which is being occupied by patient (Foreign Key of Bed Table)

VISIT						
Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
PA PERSON ID	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for patient person instance. Foreign key for outpatient entity
VISIT NUMBER	Numeric	INT	Y	Default size of INT data type	NOT NULL IDENTITY(1111,1) PRIMARY KEY	Auto-generated unique identifier for visit number instance starting at 1111.
VISIT DATE	Date	DATE	Y	YYYY/MM/DD	NOT NULL	Date visited
VISIT TIME	Time	TIME(X)	Y	0 HH:MM:SS	NOT NULL	Time visited

VISIT CARE CENTER						
Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
PA PERSON ID	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for patient person instance. Foreign key for visit entity
VISIT NUMBER	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for visit number instance. Foreign key for visit entity
CARE CENTER ID	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for care center instance. Foreign key for care center entity

BED						
Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
BED NUMBER	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for bed number instance starting at 1111.
ROOM NUMBER	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for room number instance starting at 1111.
IS_AVAILABLE	Character	CHAR(X)	Y	1	NOT NULL	Is the bed available (Y/N)
CARE CENTER ID	Numeric	INT	Y	Default size of INT data type	NOT NULL FOREIGN KEY	ID of Care Center where the beds are (Foreign Key of care center Table)
BED TYPE ID	Numeric	TINYINT	Y	Default size of TINYINT data type	NOT NULL FOREIGN KEY	ID of Bed Type (Foreign Key of Bed Type Table)

BED TYPE						
Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
BED TYPE ID	Numeric	TINYINT	Y	Default size of TINYINT data type	NOT NULL IDENTITY(1,1) PRIMARY KEY	Auto-generated unique identifier for bed type instance starting at 1.
BED TYPE DESCRIPTION	String	VARCHAR(X)	Y	25	NOT NULL	Description of the bed type

CARE CENTER						
Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
CARE CENTER ID	Numeric	INT	Y	Default size of INT data type	NOT NULL IDENTITY(1111,1) PRIMARY KEY	Auto-generated unique identifier for care center instance starting at 1111.
CARE CENTER NAME	String	VARCHAR(X)	Y	50	NOT NULL	Name of care center
CARE CENTER ADDRESS	String	NVARCHAR(X)	Y	75	NOT NULL	Address of care center
NURSE IN CHARGE	Numeric	INT	Y	Default size of INT data type	NOT NULL FOREIGN KEY	ID of employee who is in charge of the care center (Foreign Key of Employee Table)

EMPLOYEE						
Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
EM PERSON ID	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for Employee instance. Foreign key for person entity
DATE HIRED	Date	DATE	Y	YYYY/MM/DD	NOT NULL	The date the employee was hired
EMPLOYEE TYPE	String	VARCHAR(X)	Y	15	NOT NULL	Type of employee

VOLUNTEER						
Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
VO PERSON ID	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for volunteer instance. Foreign key for person entity
INTEREST ID	Numeric	TINYINT	Y	Default size of TINYINT data type	NOT NULL FOREIGN KEY	ID of Interest who is a volunteer (Foreign Key of Interest Table)

## VOLUNTEER ASSIGNED CARE CENTER

Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
VO PERSON ID	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for a volunteer instance. Foreign key for volunteer entity
CARE CENTER ID	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for care center instance. Foreign key for care center entity

## SKILL

Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
SKILL ID	Numeric	INT	Y	Default size of INT data type	NOT NULL IDENTITY(1,1) PRIMARY KEY	Auto-generated unique identifier for skill instance starting at 1.
SKILL	String	VARCHAR(X)	Y	20	NOT NULL	Skill set
VO PERSON ID	Numeric	INT	Y	Default size of INT data type	NOT NULL FOREIGN KEY	ID of volunteer person (Foreign Key of volunteer Table)

## INTEREST

Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
INTEREST ID	Numeric	TINYINT	Y	Default size of TINYINT data type	NOT NULL IDENTITY(1,1) PRIMARY KEY	Auto-generated unique identifier for interest instance starting at 1.
INTEREST DESCRIPTION	String	VARCHAR(X)	Y	25	NOT NULL	Description of interest

Volunteer Supervisor						
Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
PH PERSON ID	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for a Physician instance. Foreign key for physician entity
EM PERSON ID	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for Employee instance. Foreign key for employee entity
VO PERSON ID	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for Volunteer instance. Foreign key for volunteer entity
BEGIN DATE	Date	DATE	Y	YYYY/MM/DD	NOT NULL	Start date of volunteer
END DATE	Date	DATE	Y	YYYY/MM/DD	NOT NULL	End date of volunteer
HOURS WORKED	Numeric	INT	Y	Default size of INT data type	NOT NULL	Hours worked by volunteer
WORK UNIT ID	Numeric	TINYINT	Y	Default size of TINYINT data type	NOT NULL FOREIGN KEY	ID of Work Unit who is a volunteer (Foreign Key of Work Unit Table)

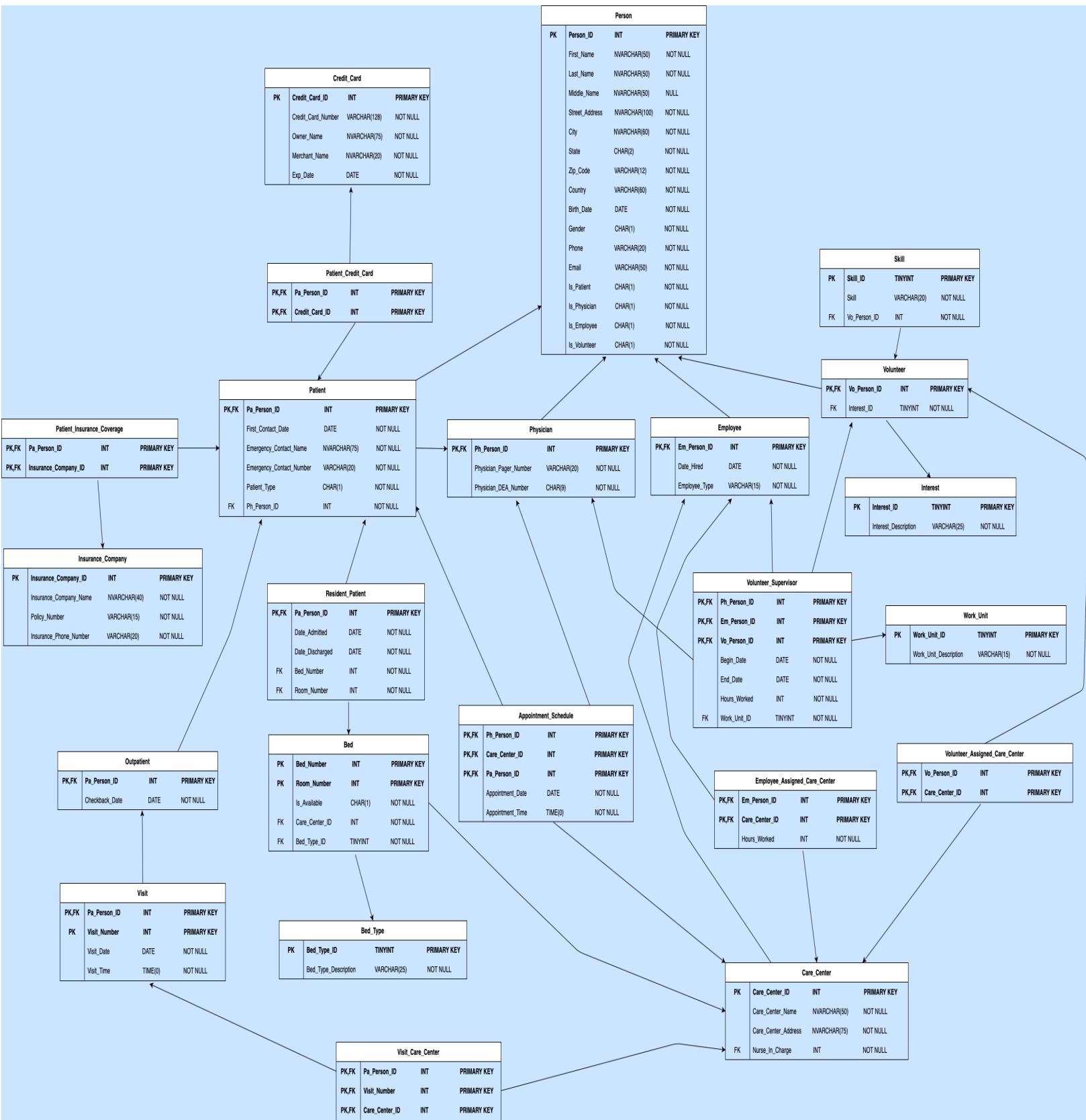
Employee Assigned Care Center						
Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
EM PERSON ID	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for Employee instance. Foreign key for employee entity
CARE CENTER ID	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for a care center instance. Foreign key for care center entity
HOURS WORKED	Numeric	INT	Y	Default size of INT data type	NOT NULL	Hours worked by employee

WORK UNIT						
Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
<b>WORK UNIT ID</b>	Numeric	TINYINT	Y	Default size of TINYINT data type	NOT NULL IDENTITY(1,1) PRIMARY KEY	Auto-generated unique identifier for work unit instance starting at 1.
<b>WORK UNIT DESCRIPTION</b>	String	VARCHAR(X)	Y	15	NOT NULL	Description of work unit

APPOINTMENT SCHEDULE						
Attribute Name	Data Type	MS SQL Data Type	Required?	Length/Size/Format	Constraints	Description/purpose
<b>PH PERSON ID</b>	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for physician instance. Foreign key for physician entity
<b>CARE CENTER ID</b>	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for care center instance. Foreign key for care center entity
<b>PA PERSON ID</b>	Numeric	INT	Y	Default size of INT data type	NOT NULL PRIMARY KEY	Unique identifier for patient instance. Foreign key for patient entity
<b>APPOINTMENT DATE</b>	Date	DATE	Y	YY/MM/DD	NOT NULL	Appointment date for patient, care center and doctor involved
<b>APPOINTMENT TIME</b>	Time	TIME(X)	Y	0 HH:MM:SS	NOT NULL	Appointment time for patient, care center and doctor involved

# Physical Model Schema Diagram

This section shows the Physical Model Schema Diagram. This has the entity tables with the attributes, data type and constraints as well as the primary keys and foreign keys.



## **Development & Implementation**

This section shows the create table statements for all the tables.

```
CREATE TABLE Person (
Person_ID      INT NOT NULL IDENTITY(1111,1),
First_Name     NVARCHAR(50) NOT NULL,
Last_Name      NVARCHAR(50) NOT NULL,
Middle_Name    NVARCHAR(50) NULL,
Street_Address NVARCHAR(100) NOT NULL,
City           NVARCHAR(60) NOT NULL,
State          CHAR(2) NOT NULL,
Zip_Code       VARCHAR(12) NOT NULL,
Country        VARCHAR(60) NOT NULL,
Birth_Date     DATE NOT NULL,
Gender         CHAR(1) NOT NULL,
Phone          VARCHAR(20) NOT NULL,
Email          VARCHAR(50) NULL,
Is_Patient     CHAR(1) NOT NULL,
Is_Physician   CHAR(1) NOT NULL,
Is_Employee    CHAR(1) NOT NULL,
Is_Volunteer   CHAR(1) NOT NULL,
PRIMARY KEY (Person_ID)
);
```

```
CREATE TABLE Physician (
Ph_Person_ID   INT NOT NULL,
Physician_Pager_Number  VARCHAR(20) NOT NULL,
Physician_DEA_Number   CHAR(9) NOT NULL,
PRIMARY KEY (Ph_Person_ID),
CONSTRAINT Ph_Person_ID_FK
FOREIGN KEY (Ph_Person_ID)
REFERENCES Person (Person_ID),
CONSTRAINT U DEA Num
UNIQUE (Physician_DEA_Number)
);
```

```
CREATE TABLE Employee (
    Em_Person_ID      INT NOT NULL,
    Date_Hired        DATE NOT NULL,
    Employee_Type     VARCHAR(15) NOT NULL,
    PRIMARY KEY (Em_Person_ID),
    CONSTRAINT Em_Person_ID_FK
    FOREIGN KEY (Em_Person_ID)
    REFERENCES Person (Person_ID)
);
```

```
CREATE TABLE Patient (
    Pa_Person_ID       INT NOT NULL,
    First_Contact_Date DATE NOT NULL,
    Emergency_Contact_Name NVARCHAR(75) NOT NULL,
    Emergency_Contact_Number VARCHAR(20) NOT NULL,
    Patient_Type       CHAR(1) NOT NULL,
    Ph_Person_ID       INT NOT NULL,
    PRIMARY KEY (Pa_Person_ID),
    CONSTRAINT Pa_Person_ID_FK
    FOREIGN KEY (Pa_Person_ID)
    REFERENCES Person (Person_ID),
    CONSTRAINT Pa_Ph_Person_ID_FK
    FOREIGN KEY (Ph_Person_ID)
    REFERENCES Physician (Ph_Person_ID)
);
```

```
CREATE TABLE Credit_Card (
    Credit_Card_ID      INT NOT NULL IDENTITY(1,1),
    Credit_Card_Number   VARCHAR(128) NOT NULL,
    Owner_Name           NVARCHAR(75) NOT NULL,
    Merchant_Name         VARCHAR(20) NOT NULL,
    Exp_Date             DATE NOT NULL,
    PRIMARY KEY (Credit_Card_ID)
);
```

```
CREATE TABLE Patient_Credit_Card (
    Pa_Person_ID           INT NOT NULL,
    Credit_Card_ID          INT NOT NULL,
    PRIMARY KEY (Pa_Person_ID, Credit_Card_ID),
    CONSTRAINT PCC_Pa_Person_ID_FK
        FOREIGN KEY (Pa_Person_ID)
        REFERENCES Patient (Pa_Person_ID),
    CONSTRAINT PCC_Credit_Card_ID_FK
        FOREIGN KEY (Credit_Card_ID)
        REFERENCES Credit_Card (Credit_Card_ID)
);
```

```
CREATE TABLE Insurance_Company (
    Insurance_Company_ID      INT NOT NULL IDENTITY(1111,1),
    Insurance_Company_Name     VARCHAR(40) NOT NULL,
    Policy_Number               VARCHAR(15) NOT NULL,
    Insurance_Phone_Number      VARCHAR(20) NOT NULL,
    PRIMARY KEY (Insurance_Company_ID)
);
```

```
CREATE TABLE Patient_Insurance_Coverage (
    Pa_Person_ID           INT NOT NULL,
    Insurance_Company_ID     INT NOT NULL,
    PRIMARY KEY (Pa_Person_ID, Insurance_Company_ID),
    CONSTRAINT PIC_Pa_Person_ID_FK
        FOREIGN KEY (Pa_Person_ID)
        REFERENCES Patient (Pa_Person_ID),
    CONSTRAINT PIC_Insurance_Company_ID_FK
        FOREIGN KEY (Insurance_Company_ID)
        REFERENCES Insurance_Company (Insurance_Company_ID)
);
```

```
CREATE TABLE Care_Center (  
Care_Center_ID           INT NOT NULL IDENTITY(1111,1),  
Care_Center_Name         VARCHAR(50) NOT NULL,  
Care_Center_Address      NVARCHAR(75) NOT NULL,  
Nurse_In_Charge          INT NOT NULL,  
  
PRIMARY KEY (Care_Center_ID),  
  
CONSTRAINT Nurse_In_Charge_FK  
FOREIGN KEY (Nurse_In_Charge)  
REFERENCES Employee (Em_Person_ID)  
);
```

```
CREATE TABLE Outpatient (  
Pa_Person_ID             INT NOT NULL,  
Checkback_Date            DATE NOT NULL,  
  
PRIMARY KEY (Pa_Person_ID),  
  
CONSTRAINT O_Pa_PersonID_FK  
FOREIGN KEY (Pa_Person_ID)  
REFERENCES Patient (Pa_Person_ID)  
);
```

```
CREATE TABLE Visit (  
Pa_Person_ID              INT NOT NULL,  
Visit_Number               INT NOT NULL IDENTITY(1111,1),  
Visit_Date                 DATE NOT NULL,  
Visit_Time                 TIME (0) NOT NULL,  
  
PRIMARY KEY (Visit_Number),  
  
CONSTRAINT V_Pa_Person_ID_FK  
FOREIGN KEY (Pa_Person_ID)  
REFERENCES Outpatient (Pa_Person_ID)  
);
```

```

CREATE TABLE Visit_Care_Center (
    Pa_Person_ID           INT NOT NULL,
    Visit_Number            INT NOT NULL,
    Care_Center_ID          INT NOT NULL,
    PRIMARY KEY (Pa_Person_ID, Visit_Number, Care_Center_ID),
    CONSTRAINT VCC_Pa_Person_ID_FK
    FOREIGN KEY (Pa_Person_ID)
    REFERENCES Visit (Pa_Person_ID),
    CONSTRAINT VCC_Visit_Number_FK
    FOREIGN KEY (Visit_Number)
    REFERENCES Visit (Visit_Number),
    CONSTRAINT VCC_Care_Center_ID_FK
    FOREIGN KEY (Care_Center_ID)
    REFERENCES Care_Center (Care_Center_ID)
);

```

```

CREATE TABLE Bed_Type (
    Bed_Type_ID              TINYINT NOT NULL IDENTITY(1,1),
    Bed_Type_Description       VARCHAR(25) NOT NULL,
    PRIMARY KEY (Bed_Type_ID)
);

```

```

CREATE TABLE Bed (
    Bed_Number                INT NOT NULL,
    Room_Number                INT NOT NULL,
    Is_Available               CHAR(1) NOT NULL,
    Care_Center_ID              INT NOT NULL,
    Bed_Type_ID                 TINYINT NOT NULL,
    PRIMARY KEY (Bed_Number, Room_Number),

```

```

    CONSTRAINT B_Bed_Type_ID_FK
    FOREIGN KEY (Bed_Type_ID)
    REFERENCES Bed_Type (Bed_Type_ID),

```

```

    CONSTRAINT B_Care_Center_ID_FK
    FOREIGN KEY (Care_Center_ID)
    REFERENCES Care_Center (Care_Center_ID)
);

```

```
CREATE TABLE Resident_Patient (
    Pa_Person_ID           INT NOT NULL,
    Date_Admitted          DATE NOT NULL,
    Date_Discharged         DATE NOT NULL,
    Bed_Number              INT NOT NULL,
    Room_Number             INT NOT NULL,
    PRIMARY KEY (Pa_Person_ID),
    CONSTRAINT RP_Bed_Room_Number_FK
        FOREIGN KEY (Bed_Number, Room_Number)
        REFERENCES Bed (Bed_Number, Room_Number),
    CONSTRAINT RP_Pa_Person_ID_FK
        FOREIGN KEY (Pa_Person_ID)
        REFERENCES Patient (Pa_Person_ID)
);
```

```
CREATE TABLE Employee_Assigned_Care_Center (
    Em_Person_ID            INT NOT NULL,
    Care_Center_ID          INT NOT NULL,
    Hours_Worked             INT NOT NULL,
    PRIMARY KEY (Em_Person_ID, Care_Center_ID),
    CONSTRAINT EACC_Employee_ID_FK
        FOREIGN KEY (Em_Person_ID)
        REFERENCES Employee (Em_Person_ID),
    CONSTRAINT EACC_Care_Center_ID_FK
        FOREIGN KEY (Care_Center_ID)
        REFERENCES Care_Center (Care_Center_ID)
);

CREATE TABLE Interest (
    Interest_ID              TINYINT NOT NULL IDENTITY(1,1),
    Interest_Description      VARCHAR(25) NOT NULL,
    PRIMARY KEY (Interest_ID)
);
```

```
CREATE TABLE Volunteer (
    Vo_Person_ID      INT NOT NULL,
    Interest_ID        TINYINT NOT NULL,
    PRIMARY KEY (Vo_Person_ID),
    CONSTRAINT V_Vo_Person_ID_FK
    FOREIGN KEY (Vo_Person_ID)
    REFERENCES Person (Person_ID),
    CONSTRAINT V_Interest_ID_FK
    FOREIGN KEY (Interest_ID)
    REFERENCES Interest (Interest_ID)
);
```

```
CREATE TABLE Skill (
    Skill_ID           TINYINT NOT NULL IDENTITY(1,1),
    Skill               VARCHAR(20) NOT NULL,
    Vo_Person_ID       INT NOT NULL,
    PRIMARY KEY (Skill_ID),
    CONSTRAINT S_Vo_Person_ID_FK
    FOREIGN KEY (Vo_Person_ID)
    REFERENCES Volunteer (Vo_Person_ID)
);
```

```
CREATE TABLE Volunteer_Assigned_Care_Center (
    Vo_Person_ID       INT NOT NULL,
    Care_Center_ID     INT NOT NULL,
    PRIMARY KEY (Vo_Person_ID, Care_Center_ID),
    CONSTRAINT VACC_Vo_Person_ID_FK
    FOREIGN KEY (Vo_Person_ID)
    REFERENCES Volunteer (Vo_Person_ID),
    CONSTRAINT VACC_Care_Center_ID_FK
    FOREIGN KEY (Care_Center_ID)
    REFERENCES Care_Center (Care_Center_ID)
);
```

```
CREATE TABLE Work_Unit (
    Work_Unit_ID           TINYINT NOT NULL IDENTITY(1,1),
    Work_Unit_Description   VARCHAR(15) NOT NULL,
    PRIMARY KEY (Work_Unit_ID)
);
```

```
CREATE TABLE Volunteer_Supervisor (
    Ph_Person_ID      INT NOT NULL,
    Em_Person_ID      INT NOT NULL,
    Vo_Person_ID      INT NOT NULL,
    Begin_Date        DATE NOT NULL,
    End_Date          DATE NOT NULL,
    Hours_Worked      INT NOT NULL,
    Work_Unit_ID      TINYINT NOT NULL,
```

```
PRIMARY KEY (Ph_Person_ID, Em_Person_ID, Vo_Person_ID),
```

```
CONSTRAINT VS_Ph_Person_ID_FK
FOREIGN KEY (Ph_Person_ID)
REFERENCES Physician (Ph_Person_ID),
```

```
CONSTRAINT VS_Employee_ID_FK
FOREIGN KEY (Em_Person_ID)
REFERENCES Employee (Em_Person_ID),
```

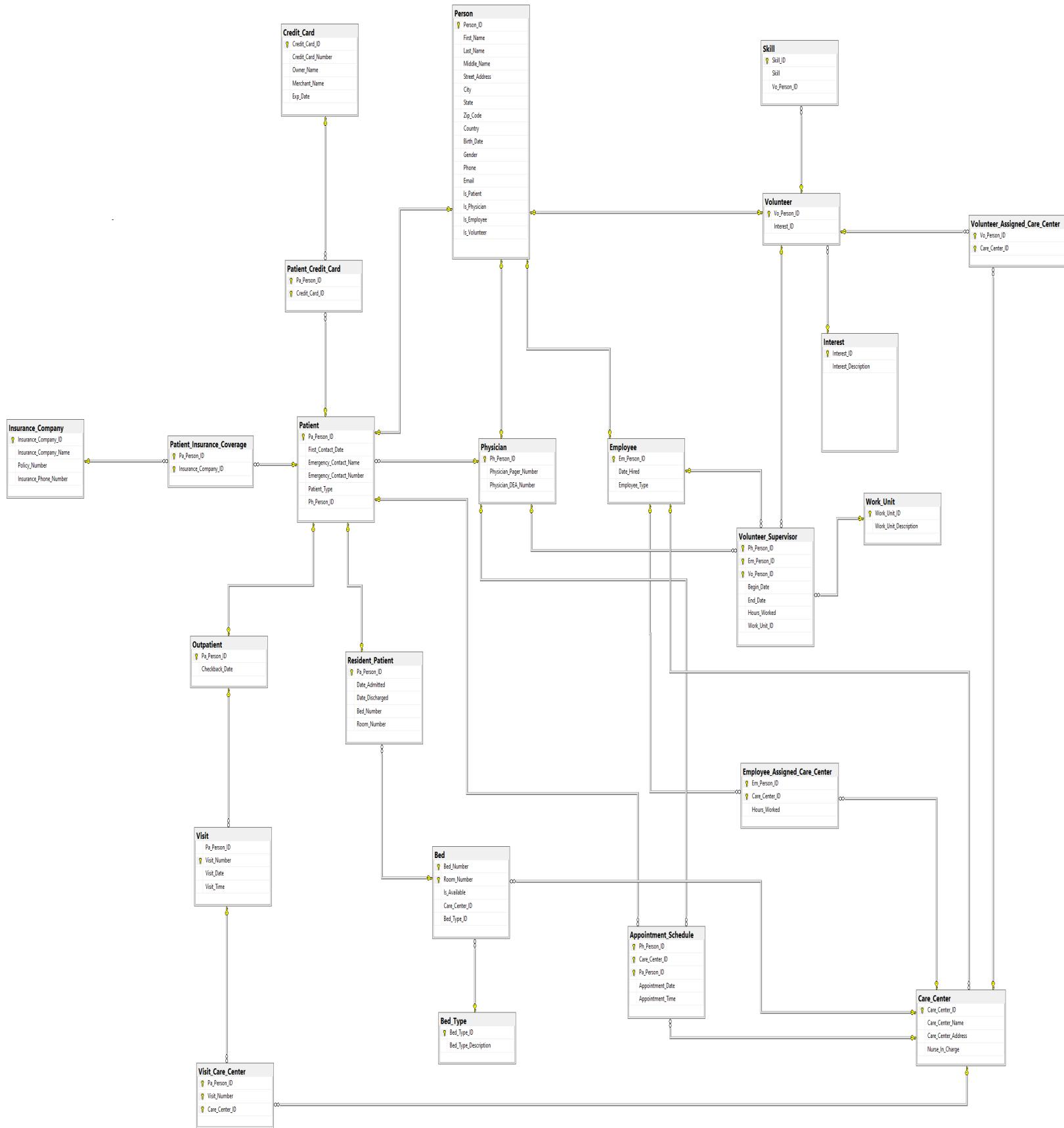
```
CONSTRAINT VS_Vo_Person_ID_FK
FOREIGN KEY (Vo_Person_ID)
REFERENCES Volunteer (Vo_Person_ID),
```

```
CONSTRAINT VS_Work_Unit_ID_FK
FOREIGN KEY (Work_Unit_ID)
REFERENCES Work_Unit (Work_Unit_ID)
);
```

```
CREATE TABLE Appointment_Schedule (
    Ph_Person_ID           INT NOT NULL,
    Care_Center_ID          INT NOT NULL,
    Pa_Person_ID             INT NOT NULL,
    Appointment_Date        DATE NOT NULL,
    Appointment_Time         TIME(0) NOT NULL,
    PRIMARY KEY (Ph_Person_ID, Care_Center_ID, Pa_Person_ID),
    CONSTRAINT AS_Ph_Person_ID_FK
    FOREIGN KEY (Ph_Person_ID)
    REFERENCES Physician (Ph_Person_ID),
    CONSTRAINT AS_Care_Center_ID_FK
    FOREIGN KEY (Care_Center_ID)
    REFERENCES Care_Center (Care_Center_ID),
    CONSTRAINT AS_Pa_Person_ID_FK
    FOREIGN KEY (Pa_Person_ID)
    REFERENCES Patient (Pa_Person_ID)
);
```

# Development & Implementation Physical Schema Diagram

This section shows the database schema table diagram for all the tables.



## Development & Implementation Testing

This section shows the SQL test queries which shows the insert, select, update, and delete statements.

Information is going to be inserted in the Person table.

**INSERT INTO** Person

(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country, Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)

**VALUES**

(N'Gerard', N'Piqué', N'Matheus', N'123 Parsons Avenue', N'Flushing', 'NY', '48433', 'United States', '1999-01-01', 'M', '212-946-1478', 'GerardPique@gmail.com', 'Y', 'Y', 'N', 'N');

**INSERT INTO** Person

(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country, Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)

**VALUES**

(N'Nelson', N'Semedo', N'Junior', N'456-01 01 Avenue', N'Corona', 'NY', '11368', 'United States', '1998-02-02', 'M', '212-456-1234', 'NelsonSemedo@outlook.com', 'Y', 'Y', 'N', 'N');

**INSERT INTO** Person

(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country, Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)

**VALUES**

(N'Paul', N'Pogba', N'Mike', N'789 Harlem Avenue', N'Manhattan', 'NY', '12345', 'United States', '1997-03-03', 'M', '212-923-2595', 'PaulPogba@gmail.com', 'Y', 'Y', 'N', 'N');

**INSERT INTO** Person

(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country, Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)

**VALUES**

(N'Fernan', N'Torres', N'Luis', N'234 Park Avenue', N'Manhattan', 'NY', '67891', 'United States', '1996-04-04', 'M', '212-369-1471', 'FernanTorres@yahoo.com', 'Y', 'Y', 'N', 'N');

**INSERT INTO** Person

(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country, Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)

**VALUES**

(N'Thomas', N'Bürki', N'Klopp', N'567 City Avenue', N'Manhattan', 'NY', '12345', 'United States', '1995-05-05', 'M', '212-729-2583', 'ThomasBurki@hotmail.com', 'Y', 'Y', 'N', 'N');

**INSERT INTO** Person

(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country, Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)

**VALUES**

(N'Erika', N'Gonzalez', N'Tapia', N'786 High Avenue', N'Flushing', 'NY', '48433', 'United States', '1990-06-06', 'F', '315-956-4242', 'ErikaGonzalez@gmail.com', 'Y', 'N', 'Y', 'N');

**INSERT INTO** Person

(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country, Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)

**VALUES**

(N'Stephany', N'Sánchez', N'María', N'238 Low Avenue', N'Corona', 'NY', '11368', 'United States', '1992-07-07', 'F', '315-454-5649', 'Stephanysanchez@outlook.com', 'Y', 'N', 'Y', 'N');

**INSERT INTO** Person

(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country, Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)

**VALUES**

(N'Lisa', N'Lainez', N'Cortez', N'986 Apple Avenue', N'Bronx', 'NY', '10458', 'United States', '1997-08-08', 'F', '315-275-3281', 'LisaCLainez@gmail.com', 'Y', 'N', 'Y', 'N');

**INSERT INTO** Person

(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country, Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)

**VALUES**

(N'Genesis', N'Gayà', N'Esteban', N'814 Marcus Avenue', N'Bronx', 'NY', '23456', 'United States', '1995-09-09', 'F', '315-418-7342', 'GenesisGaya@hotmail.com', 'Y', 'N', 'Y', 'N');

**INSERT INTO** Person

(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country, Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)

**VALUES**

(N'Eric', N'Wass', N'Albert', N'912 Orange Avenue', N'Brooklyn', 'NY', '11203', 'United States', '1998-10-10', 'M', '315-374-9523', 'EricWass@yahoo.com', 'Y', 'N', 'Y', 'N');

**INSERT INTO** Person

(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country, Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)

**VALUES**

(N'Melissa', N'Neves', N'Franco', N'915 Height Avenue', N'Manhattan', 'NY', '77665', 'United States', '1995-11-11', 'F', '347-867-5634', 'MelissaNeves@hotmail.com', 'N', 'N', 'Y', 'Y');

```
INSERT INTO Person
(First_Name, Last_Name, Middle_Name, Street_Address, City, State, Zip_Code, Country,
Birth_Date, Gender, Phone, Email, Is_Patient, Is_Physician, Is_Employee, Is_Volunteer)
VALUES
(N'Raúl', N'Guerrero', N'Jiménez', N'190 Town Avenue', N'Corona', 'NY', '99113', 'United
States', '1996-12-05', 'M', '347-729-2583', 'RaulGuerrero@yahoo.com', 'N', 'N', 'Y', 'Y');
```

```
INSERT INTO Person
(First_Name, Last_Name, Middle_Name, Street_Address, City, State, Zip_Code, Country,
Birth_Date, Gender, Phone, Email, Is_Patient, Is_Physician, Is_Employee, Is_Volunteer)
VALUES
(N'Delli', N'Alli', N'Neto', N'898 Pound Avenue', N'Brooklyn', 'NY', '33433', 'United States',
'1994-02-05', 'F', '347-234-8127', 'DelliAlli@gmail.com', 'N', 'N', 'Y', 'Y');
```

```
INSERT INTO Person
(First_Name, Last_Name, Middle_Name, Street_Address, City, State, Zip_Code, Country,
Birth_Date, Gender, Phone, Email, Is_Patient, Is_Physician, Is_Employee, Is_Volunteer)
VALUES
(N'Davinson', N'Sánchez', N'Frank', N'128 Pond Avenue', N'Queens', 'NY', '34781', 'United
States', '1993-04-05', 'M', '347-104-0390', 'DavidsonSanchez@outlook.com', 'N', 'N', 'Y', 'Y');
```

```
INSERT INTO Person
(First_Name, Last_Name, Middle_Name, Street_Address, City, State, Zip_Code, Country,
Birth_Date, Gender, Phone, Email, Is_Patient, Is_Physician, Is_Employee, Is_Volunteer)
VALUES
(N'Fernanda', N'Aurier', N'Maria', N'674 Middle Avenue', N'Corona', 'NY', '87288', 'United
States', '1996-06-05', 'F', '347-491-1022', 'FernandaAurier@hotmail.com', 'N', 'N', 'Y', 'Y');
```

```
INSERT INTO Person
(First_Name, Last_Name, Middle_Name, Street_Address, City, State, Zip_Code, Country,
Birth_Date, Gender, Phone, Email, Is_Patient, Is_Physician, Is_Employee, Is_Volunteer)
VALUES
(N'Victor', N'Wanyama', N'Eric', N'254 First Avenue', N'Queens', 'NY', '74472', 'United States',
'1993-08-05', 'M', '516-810-7077', 'VictorWanyama@gmail.com', 'Y', 'N', 'N', 'Y');
```

```
INSERT INTO Person
(First_Name, Last_Name, Middle_Name, Street_Address, City, State, Zip_Code, Country,
Birth_Date, Gender, Phone, Email, Is_Patient, Is_Physician, Is_Employee, Is_Volunteer)
VALUES
(N'Thomas', N'Lemar', N'Eriksen', N'752 Second Avenue', N'Bronx', 'NY', '64823', 'United
States', '1994-12-02', 'M', '516-104-9875', 'ThomasLemar@outlook.com', 'Y', 'N', 'N', 'Y');
```

```
INSERT INTO Person
(First_Name, Last_Name, Middle_Name, Street_Address, City, State, Zip_Code, Country,
Birth_Date, Gender, Phone, Email, Is_Patient, Is_Physician, Is_Employee, Is_Volunteer)
VALUES
(N'Cindy', N'Correa', N'Miguel', N'104 King Avenue', N'Queens', 'NY', '45754', 'United States',
'1991-02-05', 'F', '516-334-4404', 'CindyCorrea@hotmail.com', 'Y', 'N', 'N', 'Y);

INSERT INTO Person
(First_Name, Last_Name, Middle_Name, Street_Address, City, State, Zip_Code, Country,
Birth_Date, Gender, Phone, Email, Is_Patient, Is_Physician, Is_Employee, Is_Volunteer)
VALUES
(N'Álvaro', N'Morata', N'Maron', N'863 View Avenue', N'Manhattan', 'NY', '16783', 'United
States', '1992-03-01', 'M', '516-994-4103', 'AlvaroMorata@gmail.com', 'Y', 'N', 'N', 'Y);

INSERT INTO Person
(First_Name, Last_Name, Middle_Name, Street_Address, City, State, Zip_Code, Country,
Birth_Date, Gender, Phone, Email, Is_Patient, Is_Physician, Is_Employee, Is_Volunteer)
VALUES
(N'Jefferson', N'Montero', N'Mike', N'923 Brown Avenue', N'Bronx', 'NY', '78129', 'United
States', '1998-09-15', 'M', '516-564-9987', 'JeffersonMontero@yahoo.com', 'Y', 'N', 'N', 'Y);

INSERT INTO Person
(First_Name, Last_Name, Middle_Name, Street_Address, City, State, Zip_Code, Country,
Birth_Date, Gender, Phone, Email, Is_Patient, Is_Physician, Is_Employee, Is_Volunteer)
VALUES
(N'Erick', N'Angulo', N'Ibarra', N'812 Queen Avenue', N'Brooklyn', 'NY', '17878', 'United States',
'1990-04-13', 'M', '607-339-8856', 'ErickAngulo@gmail.com', 'N', 'Y', 'N', 'Y);

INSERT INTO Person
(First_Name, Last_Name, Middle_Name, Street_Address, City, State, Zip_Code, Country,
Birth_Date, Gender, Phone, Email, Is_Patient, Is_Physician, Is_Employee, Is_Volunteer)
VALUES
(N'Jennifer', N'Maddison', N'Sancho', N'723 Quart Avenue', N'Manhattan', 'NY', '34998', 'United
States', '1998-09-14', 'F', '607-221-7764', 'JenniferMaddison@hotmail.com', 'N', 'Y', 'N', 'Y);

INSERT INTO Person
(First_Name, Last_Name, Middle_Name, Street_Address, City, State, Zip_Code, Country,
Birth_Date, Gender, Phone, Email, Is_Patient, Is_Physician, Is_Employee, Is_Volunteer)
VALUES
(N'Mesut', N'Özil', N'Willock', N'296 Hill Avenue', N'Manhattan', 'NY', '93221', 'United States',
'1999-10-10', 'M', '607-302-4331', 'MesutOzil@gmail.com', 'N', 'Y', 'N', 'Y');
```

**INSERT INTO** Person  
(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country,  
Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)  
**VALUES**  
(N'Smith', N'Rowe', N'Paul', N'608 Key Avenue', N'Brooklyn', 'NY', '77331', 'United States',  
'1997-07-09', 'M', '607-799-3209', 'SmithRowe@outlook.com', 'N', 'Y', 'N', 'Y');

**INSERT INTO** Person  
(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country,  
Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)  
**VALUES**  
(N'Héctor', N'Bellerín', N'Ceballos', N'505 Fleet Avenue', N'Bronx', 'NY', '34331', 'United States',  
'1995-11-19', 'M', '607-192-0238', 'HectorBellerin@yahoo.com', 'N', 'Y', 'N', 'Y');

**INSERT INTO** Person  
(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country,  
Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)  
**VALUES**  
(N'Jamie', N'Vardy', N'Federick', N'909 Line Avenue', N'Bronx', 'NY', '33776', 'United States',  
'1990-03-07', 'M', '718-119-4588', 'JamieVardy@hotmail.com', 'Y', 'Y', 'N', 'Y');

**INSERT INTO** Person  
(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country,  
Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)  
**VALUES**  
(N'Cindy', N'Sánchez', N'Navas', N'202 Low Avenue', N'Manhattan', 'NY', '11885', 'United  
States', '1997-07-06', 'F', '718-229-8890', 'CindySanchez@gmail.com', 'Y', 'Y', 'N', 'Y');

**INSERT INTO** Person  
(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country,  
Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)  
**VALUES**  
(N'Sabrina', N'Costa', N'Paulo', N'404 High Avenue', N'Manhattan', 'NY', '22994', 'United States',  
'1998-11-11', 'F', '718-732-7788', 'SabrinaCosta@outlook.com', 'Y', 'Y', 'N', 'Y');

**INSERT INTO** Person  
(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country,  
Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)  
**VALUES**  
(N'Hugo', N'Sánchez', N'Alexander', N'805 Train Avenue', N'Queens', 'NY', '99447', 'United  
States', '1990-06-17', 'M', '718-223-5679', 'HugoSanchez@yahoo.com', 'Y', 'Y', 'N', 'Y');

**INSERT INTO** Person  
(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country,  
Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)  
**VALUES**  
(N'Raúl', N'Giménez', N'Cristian', N'679 Can Avenue', N'Queens', 'NY', '66994', 'United States',  
'1991-05-20', 'M', '718-445-1122', 'RaulGimenez@gmail.com', 'Y', 'Y', 'N', 'Y');

**INSERT INTO** Person  
(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country,  
Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)  
**VALUES**  
(N'Lisa', N'Pena', N'Saúl', N'750 Phone Avenue', N'Manhattan', 'NY', '67478', 'United States',  
'1996-07-23', 'F', '845-887-3399', 'LisaPena@hotmail.com', 'Y', 'N', 'Y', 'Y');

**INSERT INTO** Person  
(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country,  
Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)  
**VALUES**  
(N'Funes', N'Mori', N'Antonio', N'100 Cord Avenue', N'Manhattan', 'NY', '97834', 'United States',  
'1994-12-12', 'M', '845-708-7080', 'FunesMori@outlook.com', 'Y', 'N', 'Y', 'Y');

**INSERT INTO** Person  
(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country,  
Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)  
**VALUES**  
(N'Carlos', N'Pineda', N'Gallardo', N'200 Flow Avenue', N'Bronx', 'NY', '84312', 'United States',  
'1993-04-03', 'M', '845-339-5566', 'CarlosPineda@yahoo.com', 'Y', 'N', 'Y', 'Y');

**INSERT INTO** Person  
(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country,  
Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)  
**VALUES**  
(N'David', N'Neres', N'Borre', N'300 Power Avenue', N'Bronx', 'NY', '57382', 'United States',  
'1997-08-24', 'M', '845-691-4477', 'DavidNeres@gmail.com', 'Y', 'N', 'Y', 'Y');

**INSERT INTO** Person  
(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country,  
Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)  
**VALUES**  
(N'Diego', N'Carlos', N'Fernández', N'400 Inner Avenue', N'Bronx', 'NY', '89172', 'United States',  
'1998-05-27', 'M', '845-404-1357', 'DiegoCarlos@hotmail.com', 'Y', 'N', 'Y', 'Y');

**INSERT INTO** Person  
(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country,  
Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)  
**VALUES**  
(N'Iker', N'Casillas', N'Mark', N'150 Set Avenue', N'Bronx', 'NY', '47829', 'United States',  
'1993-06-01', 'M', '929-197-7453', 'IkerCasillas@outlook.com', 'Y', 'N', 'Y', 'Y');

**INSERT INTO** Person  
(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country,  
Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)  
**VALUES**  
(N'Cristina', N'Oblak', N'Cavani', N'250 Lincoln Avenue', N'Brooklyn', 'NY', '78237', 'United  
States', '1992-07-09', 'F', '914-682-8923', 'CristinaOblak@yahoo.com', 'Y', 'N', 'Y', 'Y');

**INSERT INTO** Person  
(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country,  
Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)  
**VALUES**  
(N'Calvin', N'Lewis', N'Martin', N'500 Leaf Avenue', N'Queens', 'NY', '78123', 'United States',  
'1992-06-17', 'M', '917-417-8293', 'CalvinLewis@outlook.com', 'N', 'N', 'N', 'N');

**INSERT INTO** Person  
(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country,  
Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)  
**VALUES**  
(N'Lucia', N'Hernández', N'Pineda', N'600 Shield Avenue', N'Brooklyn', 'NY', '17984', 'United  
States', '1992-09-18', 'F', '917-839-7568', 'LuciaHernandez@yahoo.com', 'N', 'N', 'N', 'N');

**INSERT INTO** Person  
(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country,  
Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)  
**VALUES**  
(N'Diego', N'Valdez', N'Córdova', N'700 Box Avenue', N'Queens', 'NY', '19876', 'United States',  
'1999-12-24', 'M', '917-334-8287', 'DiegoValdez@hotmail.com', 'N', 'N', 'N', 'N');

**INSERT INTO** Person  
(First\_Name, Last\_Name, Middle\_Name, Street\_Address, City, State, Zip\_Code, Country,  
Birth\_Date, Gender, Phone, Email, Is\_Patient, Is\_Physician, Is\_Employee, Is\_Volunteer)  
**VALUES**  
(N'Dani', N'Alves', N'Cáceres', N'800 Ford Avenue', N'Manhattan', 'NY', '79432', 'United States',  
'1991-01-25', 'M', '917-590-5221', 'DaniAlves@gmail.com', 'N', 'N', 'N', 'N');

```
INSERT INTO Person
(First_Name, Last_Name, Middle_Name, Street_Address, City, State, Zip_Code, Country,
Birth_Date, Gender, Phone, Email, Is_Patient, Is_Physician, Is_Employee, Is_Volunteer)
VALUES
(N'Carlos', N'Puyol', N'Mason', N'900 Flower Avenue', N'Manhattan', 'NY', '19432', 'United
States', '1997-01-05', 'M', '917-440-6947', 'CarlosPuyol@hotmail.com', 'N', 'N', 'N', 'N');
```

```
INSERT INTO Person
(First_Name, Last_Name, Middle_Name, Street_Address, City, State, Zip_Code, Country,
Birth_Date, Gender, Phone, Email, Is_Patient, Is_Physician, Is_Employee, Is_Volunteer)
VALUES
(N'João', N'Félix', N'Cancelo', N'350 Bush Avenue', N'Manhattan', 'NY', '17837', 'United States',
'1997-04-21', 'M', '718-198-1796', 'Joaofelix@gmail.com', 'Y', 'Y', 'N', 'N');
```

```
INSERT INTO Person
(First_Name, Last_Name, Middle_Name, Street_Address, City, State, Zip_Code, Country,
Birth_Date, Gender, Phone, Email, Is_Patient, Is_Physician, Is_Employee, Is_Volunteer)
VALUES
(N'Thomas', N'Partey', N'Tierney', N'450 Light Avenue', N'Queens', 'NY', '17293', 'United
States', '1996-07-22', 'M', '347-189-6973', 'ThomasPartley@outlook.com', 'Y', 'Y', 'N', 'N');
```

Information is going to be inserted in the Physician table.

```
INSERT INTO Physician
(Ph_Person_ID, Physician_Page_Number, Physician_DEA_Number)
VALUES
(1111, '292-378-5495', 'AB1249826');
```

```
INSERT INTO Physician
(Ph_Person_ID, Physician_Page_Number, Physician_DEA_Number)
VALUES
(1112, '894-514-8472', 'CD6789542');
```

```
INSERT INTO Physician
(Ph_Person_ID, Physician_Page_Number, Physician_DEA_Number)
VALUES
(1113, '439-857-6493', 'EF1249826');
```

```
INSERT INTO Physician
(Ph_Person_ID, Physician_Page_Number, Physician_DEA_Number)
VALUES
(1114, '782-345-3488', 'GH2846624');
```

```
INSERT INTO Physician
(Ph_Person_ID, Physician_Page_Number, Physician_DEA_Number)
VALUES
(1115, '387-531-9376', 'IJ2846623');
```

```
INSERT INTO Physician
(Ph_Person_ID, Physician_Page_Number, Physician_DEA_Number)
VALUES
(1131, '792-461-3554', 'KL3958331');
```

```
INSERT INTO Physician
(Ph_Person_ID, Physician_Page_Number, Physician_DEA_Number)
VALUES
(1132, '634-782-9225', 'MN1639281');
```

```
INSERT INTO Physician
(Ph_Person_ID, Physician_Page_Number, Physician_DEA_Number)
VALUES
(1133, '752-483-6833', 'OP5629442');
```

```
INSERT INTO Physician
(Ph_Person_ID, Physician_Page_Number, Physician_DEA_Number)
VALUES
(1134, '234-784-9567', 'QR9981772');
```

```
INSERT INTO Physician
(Ph_Person_ID, Physician_Page_Number, Physician_DEA_Number)
VALUES
(1135, '823-732-8521', 'ST3219334');
```

Information is going to be inserted in the Employee table.

```
INSERT INTO Employee
(Em_Person_ID, Date_Hired, Employee_Type)
VALUES
(1116, '2012-05-04', 'RN Nurse');
```

```
INSERT INTO Employee
(Em_Person_ID, Date_Hired, Employee_Type)
VALUES
(1117, '2014-01-29', 'LPN Nurse');
```

```
INSERT INTO Employee  
(Em_Person_ID, Date_Hired, Employee_Type)  
VALUES  
(1118, '2015-03-04', 'Technician');
```

```
INSERT INTO Employee  
(Em_Person_ID, Date_Hired, Employee_Type)  
VALUES  
(1119, '2018-10-10', 'Technician');
```

```
INSERT INTO Employee  
(Em_Person_ID, Date_Hired, Employee_Type)  
VALUES  
(1120, '2014-06-09', 'RN Nurse');
```

```
INSERT INTO Employee  
(Em_Person_ID, Date_Hired, Employee_Type)  
VALUES  
(1121, '2013-11-11', 'LPN Nurse');
```

```
INSERT INTO Employee  
(Em_Person_ID, Date_Hired, Employee_Type)  
VALUES  
(1122, '2011-12-01', 'Technician');
```

```
INSERT INTO Employee  
(Em_Person_ID, Date_Hired, Employee_Type)  
VALUES  
(1123, '2010-02-10', 'Technician');
```

```
INSERT INTO Employee  
(Em_Person_ID, Date_Hired, Employee_Type)  
VALUES  
(1124, '2015-07-20', 'RN Nurse');
```

```
INSERT INTO Employee  
(Em_Person_ID, Date_Hired, Employee_Type)  
VALUES  
(1125, '2016-04-26', 'LPN Nurse');
```

```
INSERT INTO Employee  
(Em_Person_ID, Date_Hired, Employee_Type)  
VALUES  
(1141, '2015-05-21', 'RN Nurse');
```

```
INSERT INTO Employee  
(Em_Person_ID, Date_Hired, Employee_Type)  
VALUES  
(1142, '2014-02-04', 'LPN Nurse');
```

```
INSERT INTO Employee  
(Em_Person_ID, Date_Hired, Employee_Type)  
VALUES  
(1143, '2016-04-26', 'Technician');
```

```
INSERT INTO Employee  
(Em_Person_ID, Date_Hired, Employee_Type)  
VALUES  
(1144, '2017-08-09', 'RN Nurse');
```

```
INSERT INTO Employee  
(Em_Person_ID, Date_Hired, Employee_Type)  
VALUES  
(1145, '2012-01-13', 'LPN Nurse');
```

```
INSERT INTO Employee  
(Em_Person_ID, Date_Hired, Employee_Type)  
VALUES  
(1146, '2013-02-13', 'Technician');
```

```
INSERT INTO Employee  
(Em_Person_ID, Date_Hired, Employee_Type)  
VALUES  
(1147, '2014-07-13', 'Technician');
```

Information is going to be inserted in the Patient table.

```
INSERT INTO Patient  
(Pa_Person_ID, First_Contact_Date, Emergency_Contact_Name, Emergency_Contact_Number,  
Patient_Type, Ph_Person_ID)  
VALUES  
(1126, '2020-11-03', N'Thomas Jefferson', '347-456-7893', 'O', 1111);
```

```
INSERT INTO Patient  
(Pa_Person_ID, First_Contact_Date, Emergency_Contact_Name, Emergency_Contact_Number,  
Patient_Type, Ph_Person_ID)  
VALUES  
(1127, '2020-11-10', N'Luca Modrić', '917-567-8915', 'O', 1112);
```

```
INSERT INTO Patient  
(Pa_Person_ID, First_Contact_Date, Emergency_Contact_Name, Emergency_Contact_Number,  
Patient_Type, Ph_Person_ID)  
VALUES  
(1128, '2020-11-15', N'Sergio Ramos', '914-678-9122', 'R', 1113);
```

```
INSERT INTO Patient  
(Pa_Person_ID, First_Contact_Date, Emergency_Contact_Name, Emergency_Contact_Number,  
Patient_Type, Ph_Person_ID)  
VALUES  
(1129, '2020-11-20', N'Gareth Bale', '646-789-1238', 'O', 1114);
```

```
INSERT INTO Patient  
(Pa_Person_ID, First_Contact_Date, Emergency_Contact_Name, Emergency_Contact_Number,  
Patient_Type, Ph_Person_ID)  
VALUES  
(1130, '2020-11-25', N'Karim Benzema', '929-891-2349', 'R', 1115);
```

```
INSERT INTO Patient  
(Pa_Person_ID, First_Contact_Date, Emergency_Contact_Name, Emergency_Contact_Number,  
Patient_Type, Ph_Person_ID)  
VALUES  
(1141, '2020-01-19', N'Kevin Dest', '929-198-9764', 'R', 1111);
```

```
INSERT INTO Patient  
(Pa_Person_ID, First_Contact_Date, Emergency_Contact_Name, Emergency_Contact_Number,  
Patient_Type, Ph_Person_ID)  
VALUES  
(1142, '2020-10-11', N'Dean Best', '646-782-9911', 'O', 1112);
```

```
INSERT INTO Patient  
(Pa_Person_ID, First_Contact_Date, Emergency_Contact_Name, Emergency_Contact_Number,  
Patient_Type, Ph_Person_ID)  
VALUES  
(1143, '2020-09-15', N'Sean Colbert', '914-124-1461', 'R', 1113);
```

```
INSERT INTO Patient  
(Pa_Person_ID, First_Contact_Date, Emergency_Contact_Name, Emergency_Contact_Number,  
Patient_Type, Ph_Person_ID)  
VALUES  
(1144, '2020-08-07', N'Paul Yedlin', '917-896-7522', 'O', 1114);
```

```
INSERT INTO Patient  
(Pa_Person_ID, First_Contact_Date, Emergency_Contact_Name, Emergency_Contact_Number,  
Patient_Type, Ph_Person_ID)  
VALUES  
(1145, '2020-06-23', N'Jean Weah', '347-678-1983', 'R', 1115);
```

```
INSERT INTO Patient  
(Pa_Person_ID, First_Contact_Date, Emergency_Contact_Name, Emergency_Contact_Number,  
Patient_Type, Ph_Person_ID)  
VALUES  
(1146, '2020-11-02', N'John Buffon', '929-981-8172', 'O', 1111);
```

```
INSERT INTO Patient  
(Pa_Person_ID, First_Contact_Date, Emergency_Contact_Name, Emergency_Contact_Number,  
Patient_Type, Ph_Person_ID)  
VALUES  
(1147, '2020-12-05', N'Robert Pirlo', '646-781-9714', 'R', 1112);
```

```
INSERT INTO Patient  
(Pa_Person_ID, First_Contact_Date, Emergency_Contact_Name, Emergency_Contact_Number,  
Patient_Type, Ph_Person_ID)  
VALUES  
(1116, '2020-04-13', N'Adam Rodrigo', '914-761-8912', 'O', 1113);
```

```
INSERT INTO Patient  
(Pa_Person_ID, First_Contact_Date, Emergency_Contact_Name, Emergency_Contact_Number,  
Patient_Type, Ph_Person_ID)  
VALUES  
(1117, '2020-11-11', N'Sam Fernández', '917-152-5612', 'R', 1114);
```

```
INSERT INTO Patient  
(Pa_Person_ID, First_Contact_Date, Emergency_Contact_Name, Emergency_Contact_Number,  
Patient_Type, Ph_Person_ID)  
VALUES  
(1118, '2020-10-10', N'Ben Reyna', '347-451-9612', 'O', 1115);
```

```
INSERT INTO Patient  
(Pa_Person_ID, First_Contact_Date, Emergency_Contact_Name, Emergency_Contact_Number,  
Patient_Type, Ph_Person_ID)  
VALUES  
(1119, '2020-04-17', N'Saúl González', '929-156-1234', 'R', 1111);
```

```
INSERT INTO Patient  
(Pa_Person_ID, First_Contact_Date, Emergency_Contact_Name, Emergency_Contact_Number,  
Patient_Type, Ph_Person_ID)  
VALUES  
(1153, '2020-07-16', N'Kate Ortiz', '646-786-1561', 'O', 1112);
```

```
INSERT INTO Patient  
(Pa_Person_ID, First_Contact_Date, Emergency_Contact_Name, Emergency_Contact_Number,  
Patient_Type, Ph_Person_ID)  
VALUES  
(1154, '2020-09-03', N'Katie Torró', '914-907-7567', 'R', 1113);
```

```
INSERT INTO Patient  
(Pa_Person_ID, First_Contact_Date, Emergency_Contact_Name, Emergency_Contact_Number,  
Patient_Type, Ph_Person_ID)  
VALUES  
(1111, '2020-11-21', N'Beth Quintero', '917-791-7031', 'O', 1114);
```

```
INSERT INTO Patient  
(Pa_Person_ID, First_Contact_Date, Emergency_Contact_Name, Emergency_Contact_Number,  
Patient_Type, Ph_Person_ID)  
VALUES  
(1112, '2020-11-22', N'Pablo Fornals', '347-891-6851', 'R', 1115);
```

Information is going to be inserted in the Credit Card table.

```
INSERT INTO Credit_Card  
(Credit_Card_Number, Owner_Name, Merchant_Name, Exp_Date)  
VALUES  
(‘6F0BAECBF0238928B7A62CFC94FFC2E16595855640981D13D79185C211662CF6137C9C  
A34360882B3B5C14C0064A8300A613E1E33BF4DEC934B1CF6E7A65F947’, N’Victor  
Wanyama’, ‘Visa’, ‘2021-06-30’);
```

```
INSERT INTO Credit_Card  
(Credit_Card_Number, Owner_Name, Merchant_Name, Exp_Date)  
VALUES  
(‘ADE848EC2D81CCAF6F7491A033A4A6A7CE9140DB2653A41459F103DC8A98F86D45B  
3301CA2FB69C5B05E25D805DF45B700798BE9B41329D019C7A1233EDF23BD’,  
N’Thomas Lemar’, ‘Mastercard’, ‘2022-07-31’);
```

```
INSERT INTO Credit_Card  
(Credit_Card_Number, Owner_Name, Merchant_Name, Exp_Date)  
VALUES  
(‘206A702FCB480C1D51711CB7DE70EA50FB4816BF8B81CBF41C531D4009D5E1F79EBC  
2C898184C455C7EF07F9F79D4CD93FBA165BC85C2BAB8C0ABAB1E3B58C21’, N’Cindy  
Correa’, ‘AMX’, ‘2023-08-31’);
```

```
INSERT INTO Credit_Card  
(Credit_Card_Number, Owner_Name, Merchant_Name, Exp_Date)  
VALUES  
(‘FF0A993F7AC6A151CF350A5A246018A3124F4FA8A0FC12DD04A945CD80A61FCF0636  
B89CA8E6E59A5AC6F2460F770CA943C006DB21E08BAAD88CE07C5DC6E399’, N’Álvaro  
Morata’, ‘Visa’, ‘2024-09-30’);
```

```
INSERT INTO Credit_Card
(Credit_Card_Number, Owner_Name, Merchant_Name, Exp_Date)
VALUES
('2C2E8ADD838831E03FF4A67F5483150AD73BC924AF445266D348E0A1DEC876FE05F66
D6658E607832584104725F1C3ADFD4F02C854E7E5DB04F3C5262B20B595', N'Jefferson
Montero', 'Mastercard', '2025-10-31');
```

Information is going to be inserted in the Patient Credit Card table.

```
INSERT INTO Patient_Credit_Card
(Pa_Person_ID, Credit_Card_ID)
VALUES
(1126, 1);
```

```
INSERT INTO Patient_Credit_Card
(Pa_Person_ID, Credit_Card_ID)
VALUES
(1127, 2);
```

```
INSERT INTO Patient_Credit_Card
(Pa_Person_ID, Credit_Card_ID)
VALUES
(1128, 3);
```

```
INSERT INTO Patient_Credit_Card
(Pa_Person_ID, Credit_Card_ID)
VALUES
(1129, 4);
```

```
INSERT INTO Patient_Credit_Card
(Pa_Person_ID, Credit_Card_ID)
VALUES
(1130, 5);
```

Information is going to be inserted in the Insurance Company table.

```
INSERT INTO Insurance_Company
(Insurance_Company_Name, Policy_Number, Insurance_Phone_Number)
VALUES
('United Health', 'ABC123456', '347-681-2464');
```

```
INSERT INTO Insurance_Company  
(Insurance_Company_Name, Policy_Number, Insurance_Phone_Number)  
VALUES  
('Blue Cross and Blue Shield Association', 'DEF789123', '929-792-3571');
```

```
INSERT INTO Insurance_Company  
(Insurance_Company_Name, Policy_Number, Insurance_Phone_Number)  
VALUES  
('American Family Insurance', 'GHI456789', '347-542-1864');
```

```
INSERT INTO Insurance_Company  
(Insurance_Company_Name, Policy_Number, Insurance_Phone_Number)  
VALUES  
('Golden Rule Insurance Company', 'JKL135681', '614-654-9878');
```

```
INSERT INTO Insurance_Company  
(Insurance_Company_Name, Policy_Number, Insurance_Phone_Number)  
VALUES  
('Liberty Medical', 'MNO246813', '646-975-5319');
```

Information is going to be inserted in the Patient Insurance Coverage table.

```
INSERT INTO Patient_Insurance_Coverage  
(Pa_Person_ID, Insurance_Company_ID)  
VALUES  
(1126, 1111);
```

```
INSERT INTO Patient_Insurance_Coverage  
(Pa_Person_ID, Insurance_Company_ID)  
VALUES  
(1127, 1112);
```

```
INSERT INTO Patient_Insurance_Coverage  
(Pa_Person_ID, Insurance_Company_ID)  
VALUES  
(1128, 1113);
```

```
INSERT INTO Patient_Insurance_Coverage  
(Pa_Person_ID, Insurance_Company_ID)  
VALUES  
(1129, 1114);
```

```
INSERT INTO Patient_Insurance_Coverage  
(Pa_Person_ID, Insurance_Company_ID)  
VALUES  
(1130, 1115);
```

Information is going to be inserted in the Care Center table.

```
INSERT INTO Care_Center  
(Care_Center_Name, Care_Center_Address, Nurse_In_Charge)  
VALUES  
('Emergency Room Center', N'123 Apple Ave Manhattan NY 12345', 1116);
```

```
INSERT INTO Care_Center  
(Care_Center_Name, Care_Center_Address, Nurse_In_Charge)  
VALUES  
('Outpatient Care Center', N'456 Maple Ave Queens NY 67891', 1120);
```

```
INSERT INTO Care_Center  
(Care_Center_Name, Care_Center_Address, Nurse_In_Charge)  
VALUES  
('Dermatology Center', N'123 Apple Ave Manhattan NY 12345', 1124);
```

```
INSERT INTO Care_Center  
(Care_Center_Name, Care_Center_Address, Nurse_In_Charge)  
VALUES  
('Urology Center', N'456 Maple Ave Queens NY 67891', 1141);
```

```
INSERT INTO Care_Center  
(Care_Center_Name, Care_Center_Address, Nurse_In_Charge)  
VALUES  
('Surgical Unit Care Center', N'123 Apple Ave Manhattan NY 12345', 1144);
```

Information is going to be inserted in the Outpatient table.

```
INSERT INTO Outpatient  
(Pa_Person_ID, Checkback_Date)  
VALUES  
(1126, '2019-11-01');
```

```
INSERT INTO Outpatient  
(Pa_Person_ID, Checkback_Date)  
VALUES  
(1127, '2019-10-11');
```

```
INSERT INTO Outpatient
(Pa_Person_ID, Checkback_Date)
VALUES
(1129, '2019-09-12');
```

```
INSERT INTO Outpatient
(Pa_Person_ID, Checkback_Date)
VALUES
(1142, '2019-08-16');
```

```
INSERT INTO Outpatient
(Pa_Person_ID, Checkback_Date)
VALUES
(1144, '2019-05-20');
```

```
INSERT INTO Outpatient
(Pa_Person_ID, Checkback_Date)
VALUES
(1146, '2019-05-21');
```

```
INSERT INTO Outpatient
(Pa_Person_ID, Checkback_Date)
VALUES
(1116, '2019-08-22');
```

```
INSERT INTO Outpatient
(Pa_Person_ID, Checkback_Date)
VALUES
(1118, '2019-09-23');
```

```
INSERT INTO Outpatient
(Pa_Person_ID, Checkback_Date)
VALUES
(1153, '2019-10-24');
```

```
INSERT INTO Outpatient
(Pa_Person_ID, Checkback_Date)
VALUES
(1111, '2019-12-27');
```

Information is going to be inserted in the Visit table.

```
INSERT INTO Visit  
(Pa_Person_ID, Visit_Date, Visit_Time)  
VALUES  
(1126, '2020-04-13', '07:00:00');
```

```
INSERT INTO Visit  
(Pa_Person_ID, Visit_Date, Visit_Time)  
VALUES  
(1127, '2020-04-14', '08:00:00');
```

```
INSERT INTO Visit  
(Pa_Person_ID, Visit_Date, Visit_Time)  
VALUES  
(1129, '2020-04-15', '09:00:00');
```

```
INSERT INTO Visit  
(Pa_Person_ID, Visit_Date, Visit_Time)  
VALUES  
(1142, '2020-04-16', '10:00:00');
```

```
INSERT INTO Visit  
(Pa_Person_ID, Visit_Date, Visit_Time)  
VALUES  
(1144, '2020-04-17', '11:00:00');
```

```
INSERT INTO Visit  
(Pa_Person_ID, Visit_Date, Visit_Time)  
VALUES  
(1146, '2020-04-18', '12:00:00');
```

```
INSERT INTO Visit  
(Pa_Person_ID, Visit_Date, Visit_Time)  
VALUES  
(1116, '2020-04-08', '01:00:00');
```

```
INSERT INTO Visit  
(Pa_Person_ID, Visit_Date, Visit_Time)  
VALUES  
(1118, '2020-04-19', '01:30:00');
```

```
INSERT INTO Visit
(Pa_Person_ID, Visit_Date, Visit_Time)
VALUES
(1153, '2020-04-06', '02:00:00');
```

```
INSERT INTO Visit
(Pa_Person_ID, Visit_Date, Visit_Time)
VALUES
(1111, '2020-04-07', '02:30:00');
```

Information is going to be inserted in the Visit Care Center table.

```
INSERT INTO Visit_Care_Center
(Pa_Person_ID, Visit_Number, Care_Center_ID)
VALUES
(1126, 1111, 1111);
```

```
INSERT INTO Visit_Care_Center
(Pa_Person_ID, Visit_Number, Care_Center_ID)
VALUES
(1127, 1112, 1112);
```

```
INSERT INTO Visit_Care_Center
(Pa_Person_ID, Visit_Number, Care_Center_ID)
VALUES
(1129, 1113, 1113);
```

```
INSERT INTO Visit_Care_Center
(Pa_Person_ID, Visit_Number, Care_Center_ID)
VALUES
(1142, 1114, 1114);
```

```
INSERT INTO Visit_Care_Center
(Pa_Person_ID, Visit_Number, Care_Center_ID)
VALUES
(1144, 1115, 1115);
```

```
INSERT INTO Visit_Care_Center
(Pa_Person_ID, Visit_Number, Care_Center_ID)
VALUES
(1146, 1116, 1111);
```

```
INSERT INTO Visit_Care_Center  
(Pa_Person_ID, Visit_Number, Care_Center_ID)  
VALUES  
(1116, 1117, 1112);
```

```
INSERT INTO Visit_Care_Center  
(Pa_Person_ID, Visit_Number, Care_Center_ID)  
VALUES  
(1118, 1118, 1113);
```

```
INSERT INTO Visit_Care_Center  
(Pa_Person_ID, Visit_Number, Care_Center_ID)  
VALUES  
(1153, 1119, 1114);
```

```
INSERT INTO Visit_Care_Center  
(Pa_Person_ID, Visit_Number, Care_Center_ID)  
VALUES  
(1111, 1120, 1115);
```

Information is going to be inserted in the Bed Type table.

```
INSERT INTO Bed_Type  
(Bed_Type_Description)  
VALUES  
('Standard Manuel Bed');
```

```
INSERT INTO Bed_Type  
(Bed_Type_Description)  
VALUES  
('Electric Automatic Bed');
```

```
INSERT INTO Bed_Type  
(Bed_Type_Description)  
VALUES  
('Clinitron Air Bed');
```

```
INSERT INTO Bed_Type  
(Bed_Type_Description)  
VALUES  
('Stretcher');
```

```
INSERT INTO Bed_Type  
(Bed_Type_Description)  
VALUES  
('Low Bed');
```

Information is going to be inserted in the Bed table.

```
INSERT INTO Bed  
(Bed_Number, Room_Number, Is_Available, Care_Center_ID, Bed_Type_ID)  
VALUES  
(1111, 1111, 'N', 1111, 1);
```

```
INSERT INTO Bed  
(Bed_Number, Room_Number, Is_Available, Care_Center_ID, Bed_Type_ID)  
VALUES  
(1112, 1112, 'Y', 1115, 2);
```

```
INSERT INTO Bed  
(Bed_Number, Room_Number, Is_Available, Care_Center_ID, Bed_Type_ID)  
VALUES  
(1113, 1113, 'Y', 1111, 3);
```

```
INSERT INTO Bed  
(Bed_Number, Room_Number, Is_Available, Care_Center_ID, Bed_Type_ID)  
VALUES  
(1114, 1114, 'Y', 1115, 4);
```

```
INSERT INTO Bed  
(Bed_Number, Room_Number, Is_Available, Care_Center_ID, Bed_Type_ID)  
VALUES  
(1115, 1115, 'Y', 1115, 5);
```

```
INSERT INTO Bed  
(Bed_Number, Room_Number, Is_Available, Care_Center_ID, Bed_Type_ID)  
VALUES  
(1116, 1116, 'Y', 1111, 1);
```

```
INSERT INTO Bed  
(Bed_Number, Room_Number, Is_Available, Care_Center_ID, Bed_Type_ID)  
VALUES  
(1117, 1117, 'Y', 1115, 2);
```

```
INSERT INTO Bed  
(Bed_Number, Room_Number, Is_Available, Care_Center_ID, Bed_Type_ID)  
VALUES  
(1118, 1118, 'Y', 1111, 3);
```

```
INSERT INTO Bed  
(Bed_Number, Room_Number, Is_Available, Care_Center_ID, Bed_Type_ID)  
VALUES  
(1119, 1119, 'Y', 1115, 4);
```

```
INSERT INTO Bed  
(Bed_Number, Room_Number, Is_Available, Care_Center_ID, Bed_Type_ID)  
VALUES  
(1120, 1120, 'Y', 1115, 5);
```

Information is going to be inserted in the Resident Patient table.

```
INSERT INTO Resident_Patient  
(Pa_Person_ID, Date_Admitted, Date_Discharged, Bed_Number, Room_Number)  
VALUES  
(1128, '2020-03-20', '2020-04-06', 1111, 1111);
```

```
INSERT INTO Resident_Patient  
(Pa_Person_ID, Date_Admitted, Date_Discharged, Bed_Number, Room_Number)  
VALUES  
(1130, '2020-03-21', '2020-04-20', 1112, 1112);
```

```
INSERT INTO Resident_Patient  
(Pa_Person_ID, Date_Admitted, Date_Discharged, Bed_Number, Room_Number)  
VALUES  
(1141, '2020-03-22', '2020-04-07', 1113, 1113);
```

```
INSERT INTO Resident_Patient  
(Pa_Person_ID, Date_Admitted, Date_Discharged, Bed_Number, Room_Number)  
VALUES  
(1143, '2020-03-23', '2020-04-21', 1114, 1114);
```

```
INSERT INTO Resident_Patient  
(Pa_Person_ID, Date_Admitted, Date_Discharged, Bed_Number, Room_Number)  
VALUES  
(1145, '2020-03-24', '2020-04-08', 1115, 1115);
```

```
INSERT INTO Resident_Patient  
(Pa_Person_ID, Date_Admitted, Date_Discharged, Bed_Number, Room_Number)  
VALUES  
(1147, '2020-03-25', '2020-04-22', 1116, 1116);
```

```
INSERT INTO Resident_Patient  
(Pa_Person_ID, Date_Admitted, Date_Discharged, Bed_Number, Room_Number)  
VALUES  
(1117, '2020-03-26', '2020-04-09', 1117, 1117);
```

```
INSERT INTO Resident_Patient  
(Pa_Person_ID, Date_Admitted, Date_Discharged, Bed_Number, Room_Number)  
VALUES  
(1119, '2020-03-27', '2020-04-23', 1118, 1118);
```

```
INSERT INTO Resident_Patient  
(Pa_Person_ID, Date_Admitted, Date_Discharged, Bed_Number, Room_Number)  
VALUES  
(1154, '2020-03-28', '2020-04-10', 1119, 1119);
```

```
INSERT INTO Resident_Patient  
(Pa_Person_ID, Date_Admitted, Date_Discharged, Bed_Number, Room_Number)  
VALUES  
(1112, '2020-03-29', '2020-04-24', 1120, 1120);
```

Information is going to be inserted in the Employee Assigned Care Center table.

```
INSERT INTO Employee_Assigned_Care_Center  
(Em_Person_ID, Care_Center_ID, Hours_Worked)  
VALUES  
(1116, 1111, 40);
```

```
INSERT INTO Employee_Assigned_Care_Center  
(Em_Person_ID, Care_Center_ID, Hours_Worked)  
VALUES  
(1117, 1112, 45);
```

```
INSERT INTO Employee_Assigned_Care_Center  
(Em_Person_ID, Care_Center_ID, Hours_Worked)  
VALUES  
(1118, 1113, 50);
```

```
INSERT INTO Employee_Assigned_Care_Center  
(Em_Person_ID, Care_Center_ID, Hours_Worked)  
VALUES  
(1119, 1114, 32);
```

```
INSERT INTO Employee_Assigned_Care_Center  
(Em_Person_ID, Care_Center_ID, Hours_Worked)  
VALUES  
(1120, 1115, 31);
```

```
INSERT INTO Employee_Assigned_Care_Center  
(Em_Person_ID, Care_Center_ID, Hours_Worked)  
VALUES  
(1121, 1111, 36);
```

```
INSERT INTO Employee_Assigned_Care_Center  
(Em_Person_ID, Care_Center_ID, Hours_Worked)  
VALUES  
(1122, 1112, 38);
```

```
INSERT INTO Employee_Assigned_Care_Center  
(Em_Person_ID, Care_Center_ID, Hours_Worked)  
VALUES  
(1123, 1113, 39);
```

```
INSERT INTO Employee_Assigned_Care_Center  
(Em_Person_ID, Care_Center_ID, Hours_Worked)  
VALUES  
(1124, 1114, 35);
```

```
INSERT INTO Employee_Assigned_Care_Center  
(Em_Person_ID, Care_Center_ID, Hours_Worked)  
VALUES  
(1125, 1115, 36);
```

```
INSERT INTO Employee_Assigned_Care_Center  
(Em_Person_ID, Care_Center_ID, Hours_Worked)  
VALUES  
(1141, 1111, 39);
```

```
INSERT INTO Employee_Assigned_Care_Center  
(Em_Person_ID, Care_Center_ID, Hours_Worked)  
VALUES  
(1142, 1112, 36);
```

```
INSERT INTO Employee_Assigned_Care_Center  
(Em_Person_ID, Care_Center_ID, Hours_Worked)  
VALUES  
(1143, 1113, 37);
```

```
INSERT INTO Employee_Assigned_Care_Center  
(Em_Person_ID, Care_Center_ID, Hours_Worked)  
VALUES  
(1144, 1114, 39);
```

```
INSERT INTO Employee_Assigned_Care_Center  
(Em_Person_ID, Care_Center_ID, Hours_Worked)  
VALUES  
(1145, 1115, 37);
```

```
INSERT INTO Employee_Assigned_Care_Center  
(Em_Person_ID, Care_Center_ID, Hours_Worked)  
VALUES  
(1146, 1111, 38);
```

```
INSERT INTO Employee_Assigned_Care_Center  
(Em_Person_ID, Care_Center_ID, Hours_Worked)  
VALUES  
(1147, 1112, 35);
```

Information is going to be inserted in the Interest table.

```
INSERT INTO Interest  
(Interest_Description)  
VALUES  
('Emergency Room');
```

```
INSERT INTO Interest  
(Interest_Description)  
VALUES  
('Outpatient Ward');
```

```
INSERT INTO Interest  
(Interest_Description)  
VALUES  
('Surgeon Assistant');
```

```
INSERT INTO Interest  
(Interest_Description)  
VALUES  
('Dermatology Work');
```

```
INSERT INTO Interest  
(Interest_Description)  
VALUES  
('Urology Work');
```

Information is going to be inserted in the Volunteer table.

```
INSERT INTO Volunteer  
(Vo_Person_ID, Interest_ID)  
VALUES  
(1141, 1);
```

```
INSERT INTO Volunteer  
(Vo_Person_ID, Interest_ID)  
VALUES  
(1142, 2);
```

```
INSERT INTO Volunteer  
(Vo_Person_ID, Interest_ID)  
VALUES  
(1143, 3);
```

```
INSERT INTO Volunteer  
(Vo_Person_ID, Interest_ID)  
VALUES  
(1144, 4);
```

```
INSERT INTO Volunteer  
(Vo_Person_ID, Interest_ID)  
VALUES  
(1145, 5);
```

```
INSERT INTO Volunteer  
(Vo_Person_ID, Interest_ID)  
VALUES  
(1146, 1);
```

```
INSERT INTO Volunteer  
(Vo_Person_ID, Interest_ID)  
VALUES  
(1147, 2);
```

```
INSERT INTO Volunteer  
(Vo_Person_ID, Interest_ID)  
VALUES  
(1126, 3);
```

```
INSERT INTO Volunteer  
(Vo_Person_ID, Interest_ID)  
VALUES  
(1127, 4);
```

```
INSERT INTO Volunteer  
(Vo_Person_ID, Interest_ID)  
VALUES  
(1128, 5);
```

Information is going to be inserted in the Skill table.

```
INSERT INTO Skill  
(Skill, Vo_Person_ID)  
VALUES  
('Patient Care', 1141);
```

```
INSERT INTO Skill  
(Skill, Vo_Person_ID)  
VALUES  
('Emergency', 1142);
```

```
INSERT INTO Skill  
(Skill, Vo_Person_ID)  
VALUES  
('Front Office', 1143);
```

```
INSERT INTO Skill  
(Skill, Vo_Person_ID)  
VALUES  
('Medical Records', 1144);
```

```
INSERT INTO Skill  
(Skill, Vo_Person_ID)  
VALUES  
('Surgery', 1145);
```

Information is going to be inserted in the Volunteer Assigned Care Center table.

```
INSERT INTO Volunteer_Assigned_Care_Center  
(Vo_Person_ID, Care_Center_ID)  
VALUES  
(1141, 1111);
```

```
INSERT INTO Volunteer_Assigned_Care_Center  
(Vo_Person_ID, Care_Center_ID)  
VALUES  
(1142, 1112);
```

```
INSERT INTO Volunteer_Assigned_Care_Center  
(Vo_Person_ID, Care_Center_ID)  
VALUES  
(1143, 1113);
```

```
INSERT INTO Volunteer_Assigned_Care_Center  
(Vo_Person_ID, Care_Center_ID)  
VALUES  
(1144, 1114);
```

```
INSERT INTO Volunteer_Assigned_Care_Center  
(Vo_Person_ID, Care_Center_ID)  
VALUES  
(1145, 1115);
```

```
INSERT INTO Volunteer_Assigned_Care_Center  
(Vo_Person_ID, Care_Center_ID)  
VALUES  
(1146, 1111);
```

```
INSERT INTO Volunteer_Assigned_Care_Center  
(Vo_Person_ID, Care_Center_ID)  
VALUES  
(1147, 1112);
```

```
INSERT INTO Volunteer_Assigned_Care_Center  
(Vo_Person_ID, Care_Center_ID)  
VALUES  
(1126, 1113);
```

```
INSERT INTO Volunteer_Assigned_Care_Center  
(Vo_Person_ID, Care_Center_ID)  
VALUES  
(1127, 1114);
```

```
INSERT INTO Volunteer_Assigned_Care_Center  
(Vo_Person_ID, Care_Center_ID)  
VALUES  
(1128, 1115);
```

Information is going to be inserted in the Work Unit table.

```
INSERT INTO Work_Unit  
(Work_Unit_Description)  
VALUES  
('Emergency Room');
```

```
INSERT INTO Work_Unit  
(Work_Unit_Description)  
VALUES  
('Outpatient Care');
```

```
INSERT INTO Work_Unit  
(Work_Unit_Description)  
VALUES  
('Dermatology');
```

```
INSERT INTO Work_Unit  
(Work_Unit_Description)  
VALUES  
('Urology');
```

```
INSERT INTO Work_Unit  
(Work_Unit_Description)  
VALUES  
('Surgical Unit');
```

Information is going to be inserted in the Volunteer\_Supervisor table.

```
INSERT INTO Volunteer_Supervisor  
(Ph_Person_ID, Em_Person_ID, Vo_Person_ID, Begin_Date, End_Date, Hours_Worked,  
Work_Unit_ID)  
VALUES  
(1111, 1116, 1141, '2020-01-05', '2020-10-05', 05, 1);
```

```
INSERT INTO Volunteer_Supervisor  
(Ph_Person_ID, Em_Person_ID, Vo_Person_ID, Begin_Date, End_Date, Hours_Worked,  
Work_Unit_ID)  
VALUES  
(1112, 1117, 1142, '2020-02-10', '2020-05-11', 05, 2);
```

```
INSERT INTO Volunteer_Supervisor  
(Ph_Person_ID, Em_Person_ID, Vo_Person_ID, Begin_Date, End_Date, Hours_Worked,  
Work_Unit_ID)  
VALUES  
(1113, 1118, 1143, '2019-11-20', '2020-06-21', 06, 3);
```

```
INSERT INTO Volunteer_Supervisor  
(Ph_Person_ID, Em_Person_ID, Vo_Person_ID, Begin_Date, End_Date, Hours_Worked,  
Work_Unit_ID)  
VALUES  
(1114, 1119, 1144, '2019-09-15', '2020-05-16', 06, 4);
```

```
INSERT INTO Volunteer_Supervisor  
(Ph_Person_ID, Em_Person_ID, Vo_Person_ID, Begin_Date, End_Date, Hours_Worked,  
Work_Unit_ID)  
VALUES  
(1115, 1120, 1145, '2019-10-12', '2020-07-13', 03, 5);
```

```
INSERT INTO Volunteer_Supervisor  
(Ph_Person_ID, Em_Person_ID, Vo_Person_ID, Begin_Date, End_Date, Hours_Worked,  
Work_Unit_ID)  
VALUES  
(1131, 1121, 1146, '2019-12-02', '2020-05-03', 03, 1);
```

```
INSERT INTO Volunteer_Supervisor  
(Ph_Person_ID, Em_Person_ID, Vo_Person_ID, Begin_Date, End_Date, Hours_Worked,  
Work_Unit_ID)  
VALUES  
(1132, 1122, 1147, '2019-08-22', '2020-08-23', 04, 2);
```

```
INSERT INTO Volunteer_Supervisor  
(Ph_Person_ID, Em_Person_ID, Vo_Person_ID, Begin_Date, End_Date, Hours_Worked,  
Work_Unit_ID)  
VALUES  
(1133, 1123, 1126, '2019-07-19', '2020-05-20', 04, 3);
```

```
INSERT INTO Volunteer_Supervisor  
(Ph_Person_ID, Em_Person_ID, Vo_Person_ID, Begin_Date, End_Date, Hours_Worked,  
Work_Unit_ID)  
VALUES  
(1134, 1124, 1127, '2020-03-26', '2020-09-27', 03, 4);
```

```
INSERT INTO Volunteer_Supervisor  
(Ph_Person_ID, Em_Person_ID, Vo_Person_ID, Begin_Date, End_Date, Hours_Worked,  
Work_Unit_ID)  
VALUES  
(1135, 1125, 1128, '2019-11-04', '2020-05-06', 04, 5);
```

Information is going to be inserted in the Appointment Schedule table.

```
INSERT INTO Appointment_Schedule  
(Ph_Person_ID, Care_Center_ID, Pa_Person_ID, Appointment_Date, Appointment_Time)  
VALUES  
(1111, 1111, 1126, '2020-05-28', '09:00:00');
```

```
INSERT INTO Appointment_Schedule  
(Ph_Person_ID, Care_Center_ID, Pa_Person_ID, Appointment_Date, Appointment_Time)  
VALUES  
(1112, 1112, 1127, '2020-05-05', '10:00:00');
```

```
INSERT INTO Appointment_Schedule  
(Ph_Person_ID, Care_Center_ID, Pa_Person_ID, Appointment_Date, Appointment_Time)  
VALUES  
(1113, 1113, 1128, '2020-04-20', '11:00:00');
```

```
INSERT INTO Appointment_Schedule  
(Ph_Person_ID, Care_Center_ID, Pa_Person_ID, Appointment_Date, Appointment_Time)  
VALUES  
(1114, 1114, 1129, '2020-04-25', '12:00:00');
```

```
INSERT INTO Appointment_Schedule  
(Ph_Person_ID, Care_Center_ID, Pa_Person_ID, Appointment_Date, Appointment_Time)  
VALUES  
(1115, 1115, 1130, '2020-04-23', '01:30:00');
```

```
INSERT INTO Appointment_Schedule  
(Ph_Person_ID, Care_Center_ID, Pa_Person_ID, Appointment_Date, Appointment_Time)  
VALUES  
(1111, 1111, 1141, '2020-05-22', '01:40:00');
```

```
INSERT INTO Appointment_Schedule  
(Ph_Person_ID, Care_Center_ID, Pa_Person_ID, Appointment_Date, Appointment_Time)  
VALUES  
(1112, 1112, 1142, '2020-05-02', '01:50:00');
```

```
INSERT INTO Appointment_Schedule  
(Ph_Person_ID, Care_Center_ID, Pa_Person_ID, Appointment_Date, Appointment_Time)  
VALUES  
(1113, 1113, 1143, '2020-04-24', '02:00:00');
```

```
INSERT INTO Appointment_Schedule  
(Ph_Person_ID, Care_Center_ID, Pa_Person_ID, Appointment_Date, Appointment_Time)  
VALUES  
(1114, 1114, 1144, '2020-04-26', '03:00:00');
```

```
INSERT INTO Appointment_Schedule  
(Ph_Person_ID, Care_Center_ID, Pa_Person_ID, Appointment_Date, Appointment_Time)  
VALUES  
(1115, 1115, 1145, '2020-04-21', '02:30:00');
```

Using select statements to see if the results appear and to test if the table is running correctly.

This is a select statement wanting all the rows from the Insurance Company table to show where Insurance Company ID is equal to 1111.

```
SELECT *  
FROM Insurance_Company  
WHERE Insurance_Company_ID = 1111;
```

	Insurance_Company_ID	Insurance_Company_Name	Policy_Number	Insurance_Phone_Number
1	1111	United Health	ABC123456	347-681-2464

This is a select statement wanting the rows from columns Pa Person ID and Patient Type from the Patient table which is then ordered by Person Type.

```
SELECT Pa_Person_ID, Patient_Type  
FROM Patient  
ORDER BY Patient_Type;
```

	Pa_Person_ID	Patient_Type
1	1111	O
2	1116	O
3	1118	O
4	1126	O
5	1127	O
6	1129	O
7	1142	O
8	1144	O
9	1146	O
10	1153	O
11	1154	R
12	1147	R
13	1145	R
14	1143	R
15	1130	R
16	1141	R
17	1128	R
18	1119	R
19	1117	R
20	1112	R

This is a select statement wanting the rows from the Credit Card table for the Owner Name, Merchant Name and the Expiration Date. Also Patient ID in the Patient table.

```
SELECT P.Pa_Person_ID, CC.Owner_Name, CC.Merchant_Name, CC.Exp_Date  
FROM Patient AS P  
INNER JOIN Patient_Credit_Card AS PCC  
ON P.Pa_Person_ID = PCC.Pa_Person_ID  
INNER JOIN Credit_Card AS CC  
ON PCC.Credit_Card_ID = CC.Credit_Card_ID;
```

	Pa_Person_ID	Owner_Name	Merchant_Name	Exp_Date
1	1126	Victor Wanyama	Visa	2021-06-30
2	1127	Thomas Lemar	Mastercard	2022-07-31
3	1128	Cindy Correa	AMX	2023-08-31
4	1129	Álvaro Morata	Visa	2024-09-30
5	1130	Jefferson Montero	Mastercard	2025-10-31

This is an update statement updating the table insurance company and setting the attributes to new information and using that new information to replace the information that was at the primary key Insurance Company ID 1111.

```
UPDATE Insurance_Company
SET Insurance_Company_Name = 'HCSC', Policy_Number = 'ZXC369147',
Insurance_Phone_Number = '928-384-7561'
WHERE Insurance_Company_ID = 1111;
```

	Insurance_Company_ID	Insurance_Company_Name	Policy_Number	Insurance_Phone_Number
1	1111	HCSC	ZXC369147	928-384-7561
2	1112	Blue Cross and Blue Shield Association	DEF789123	929-792-3571
3	1113	American Family Insurance	GHI456789	347-542-1864
4	1114	Golden Rule Insurance Company	JKL135681	614-654-9878
5	1115	Liberty Medical	MNO246813	646-975-5319

For the second update statement, one record had to be updated from a Associative Entity Table. The table named Appointment Schedule had Pa Person ID 1126 row updated with its date and time of appointment changed.

```
UPDATE Appointment_Schedule
SET Appointment_Date = '2020-05-29', Appointment_Time = '10:00:00'
WHERE Pa_Person_ID = 1126;
```

	Ph_Person_ID	Care_Center_ID	Pa_Person_ID	Appointment_Date	Appointment_Time
1	1111	1111	1126	2020-05-29	10:00:00
2	1111	1111	1141	2020-05-22	01:40:00
3	1112	1112	1127	2020-05-05	10:00:00
4	1112	1112	1142	2020-05-02	01:50:00
5	1113	1113	1128	2020-04-20	11:00:00
6	1113	1113	1143	2020-04-24	02:00:00
7	1114	1114	1129	2020-04-25	12:00:00
8	1114	1114	1144	2020-04-26	03:00:00
9	1115	1115	1130	2020-04-23	01:30:00
10	1115	1115	1145	2020-04-21	02:30:00

For the delete statement, one record had to be deleted from a Associative Entity Table. The table named Patient Insurance Coverage had the whole row from Pa Person ID 1002 deleted.

```
DELETE FROM Patient_Insurance_Coverage
WHERE Pa_Person_ID = 1126;
```

	Pa_Person_ID	Insurance_Company_ID
1	1127	1112
2	1128	1113
3	1129	1114
4	1130	1115

## **Conclusion**

In this project the design & implement a database application for a Hospital Management System Application named EZMedicalGroup.com was done. The application was done to automate the managing of patients and hospital personnel activities etc. The Hospital Management System was designed to allow hospital employees to conveniently manage hospital patient activities such as assigning physicians, medical care, registration etc., using this system. The system done was a combination of both the Three-tiered Web and Two-tiered Windows client/server that will share the same Database Tier. The system done was possible due to the use of this methodology being used which is Planning, Analysis, Design, Implementation, and Maintenance/Administration. The tables for the database were done through this, the model diagrams, the data dictionaries, the development and implementation, development and implementation physical schema diagram, and development and implementation were also done through this.

# **Hospital Management System Database Design &**

## **Implementation Version 2.0**

### **Upgrade Objectives**

To upgrade the Database tier to Hospital Management System Application named EZMedicalGroup.com. We will follow the Project Management Methodology using the Database Application Development Lifecycle (Planning, Analysis, Design, Development/Implementation & Operations/Maintenance) to create business reports and improve the performance of the full working application by creating store procedures.

### **Business Reports Queries**

This section shows 10 business reports queries that has a report objective/purpose, persona this report is targeting, decision this report will support, data needed to create the report as well as the query and the report of the query.

1.

Report Objective/Purpose: The business objective is to see how many volunteers are leaving next month and the employee and physician being freed up from supervising them. In addition, what is the volunteers work unit and what care center they are from.

Persona This Report Is Targeting: Hospital Supervisor

Decision This Report Is Targeting: The decision is to see how many more volunteers are going to be needed for next month after the volunteers leave next month.

Data You Will Need To Create The Report So The Decision Can Be Made: Vo Person ID, Em Person ID, Ph Person ID, Work Unit ID, Work Unit Description, Care Center ID, Care Center Name, and End Date

Additional Details:

Query:

```
SELECT VS.Vo_Person_ID, VS.Em_Person_ID, VS.Ph_Person_ID,  
VS.Work_Unit_ID, WU.Work_Unit_Description, CC.Care_Center_ID,  
CC.Care_Center_Name, VS.End_Date  
  
FROM Volunteer AS V  
  
INNER JOIN Volunteer_Assigned_Care_Center AS VACC  
  
ON V.Vo_Person_ID = VACC.Vo_Person_ID  
  
INNER JOIN Care_Center AS CC  
  
ON VACC.Care_Center_ID = CC.Care_Center_ID  
  
INNER JOIN Volunteer_Supervisor AS VS  
  
ON V.Vo_Person_ID = VS.Vo_Person_ID  
  
INNER JOIN Work_Unit AS WU  
  
ON VS.Work_Unit_ID = WU.Work_Unit_ID  
  
WHERE VS.End_Date BETWEEN '2020-05-01' AND '2020-05-31'  
  
ORDER BY VS.End_Date;
```

Report:

	Vo_Person_ID	Em_Person_ID	Ph_Person_ID	Work_Unit_ID	Work_Unit_Description	Care_Center_ID	Care_Center_Name	End_Date
1	1146	1121	1131	1	Emergency Room	1111	Emergency Room Center	2020-05-03
2	1128	1125	1135	5	Surgical Unit	1115	Surgical Unit Care Center	2020-05-06
3	1142	1117	1112	2	Outpatient Care	1112	Outpatient Care Center	2020-05-11
4	1144	1119	1114	4	Urology	1114	Urology Center	2020-05-16
5	1126	1123	1133	3	Dermatology	1113	Dermatology Center	2020-05-20

2.

Report Objective/Purpose: The business objective is to see how many maintenance people are needed this upcoming week to clean the resident patient rooms that are being discharged.

Persona This Report Is Targeting: Housekeeping Management

Decision This Report Will Support: The decision is to see what day, how many, and which staff to send to go clean the rooms that are going to clear this upcoming week.

Data You Will Need To Create The Report So The Decision Can Be Made – Pa Person ID, First Name, Last Name, Patient Type, Date Discharged, Care Center ID, Care Center Name, and Room Number

Additional Details:

Query:

```
SELECT RP.Pa_Person_ID, P.First_Name, P.Last_Name, PA.Patient_Type,  
RP.Date_Discharged, CC.Care_Center_ID, CC.Care_Center_Name, B.Room_Number  
FROM Person AS P  
INNER JOIN Patient AS PA  
ON P.Person_ID = PA.Pa_Person_ID  
INNER JOIN Resident_Patient AS RP  
ON PA.Pa_Person_ID = RP.Pa_Person_ID  
INNER JOIN Bed AS B  
ON RP.Bed_Number = B.Bed_Number  
INNER JOIN Care_Center AS CC  
ON B.Care_Center_ID = CC.Care_Center_ID  
WHERE RP.Date_Discharged BETWEEN '2020-04-20' AND '2020-04-26'  
ORDER BY RP.Date_Discharged;
```

Report:

	Pa_Person_ID	First_Name	Last_Name	Patient_Type	Date_Discharged	Care_Center_ID	Care_Center_Name	Room_Number
1	1130	Jefferson	Montero	R	2020-04-20	1115	Surgical Unit Care Center	1112
2	1143	Carlos	Pineda	R	2020-04-21	1115	Surgical Unit Care Center	1114
3	1147	Cristina	Oblak	R	2020-04-22	1111	Emergency Room Center	1116
4	1119	Genesis	Gayà	R	2020-04-23	1111	Emergency Room Center	1118
5	1112	Nélson	Semedo	R	2020-04-24	1115	Surgical Unit Care Center	1120

3.

Report Objective/Purpose: The business objective is to see how many outpatients visited each care center this week and the day they visited as well as the time they visited and the care center name they visited.

Persona This Report Is Targeting: Hospital Administrator

Decision This Report Will Support: The decision is to see if there is a need for more employees in the care center or not for next week.

Data You Will Need To Create The Report So The Decision Can Be Made: Pa Person ID, First Name, Last Name, Patient Type, Visit Number, Visit Date, Visit Time, Care Center ID and Care Center Name.

Additional Details: In this business report I assumed that in the previous weeks the amount of outpatients visiting kept on increasing and wanted to see if this report kept in correlation with the previous weeks to see if the decision of needing more employees was needed.

Query:

```
SELECT O.Pa_Person_ID, P.First_Name, P.Last_Name, PA.Patient_Type,  
V.Visit_Number, V.Visit_Date, V.Visit_Time, CC.Care_Center_ID,  
CC.Care_Center_Name  
FROM Person AS P  
INNER JOIN Patient AS PA  
ON P.Person_ID = PA.Pa_Person_ID  
INNER JOIN Outpatient AS O  
ON PA.Pa_Person_ID = O.Pa_Person_ID  
INNER JOIN Visit AS V  
ON O.Pa_Person_ID = V.Pa_Person_ID  
INNER JOIN Visit_Care_Center AS VCC  
ON V.Visit_Number = VCC.Visit_Number  
INNER JOIN Care_Center AS CC  
ON VCC.Care_Center_ID = CC.Care_Center_ID  
WHERE V.Visit_Date BETWEEN '2020-04-13' AND '2020-04-19'  
ORDER BY CC.Care_Center_Name;
```

Report:

	Pa_Person_ID	First_Name	Last_Name	Patient_Type	Visit_Number	Visit_Date	Visit_Time	Care_Center_ID	Care_Center_Name
1	1118	Lisa	Lainez	0	1118	2020-04-19	01:30:00	1113	Dermatology Center
2	1129	Álvaro	Morata	0	1113	2020-04-15	09:00:00	1113	Dermatology Center
3	1126	Victor	Wanyama	0	1111	2020-04-13	07:00:00	1111	Emergency Room Center
4	1146	Iker	Casillas	0	1116	2020-04-18	12:00:00	1111	Emergency Room Center
5	1127	Thomas	Lemar	0	1112	2020-04-14	08:00:00	1112	Outpatient Care Center
6	1144	David	Neres	0	1115	2020-04-17	11:00:00	1115	Surgical Unit Care Center
7	1142	Funes	Mori	0	1114	2020-04-16	10:00:00	1114	Urology Center

4.

Report Objective/Purpose: The business objective is to see how many employees there are who worked between 40 and 50 hours this week and what type of employees are working in the care centers and what care centers were they assigned to.

Persona This Report Is Targeting: Hospital Administrator

Decision This Report Will Support: The decision is to see which employees to give more hours and which ones to give less hours based on the hours worked this week for the upcoming week.

Data You Will Need To Create The Report So The Decision Can Be Made: Em Person ID, First Name, Last Name, Employee Type, Hours Worked, Care Center ID and Care Center Name.

Additional Details: In this business report I assumed what the hours worked were for this week.

Query:

```
SELECT E.Em_Person_ID, P.First_Name, P.Last_Name, E.Employee_Type,  
EACC.Hours_Worked, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Person AS P  
INNER JOIN Employee AS E  
ON P.Person_ID = E.Em_Person_ID  
INNER JOIN Employee_Assigned_Care_Center AS EACC  
ON E.Em_Person_ID = EACC.Em_Person_ID  
INNER JOIN Care_Center AS CC  
ON EACC.Care_Center_ID = CC.Care_Center_ID  
WHERE EACC.Hours_Worked BETWEEN 40 AND 50  
ORDER BY EACC.Hours_Worked;
```

Report:

	Em_Person_ID	First_Name	Last_Name	Employee_Type	Hours_Worked	Care_Center_ID	Care_Center_Name
1	1116	Erika	Gonzalez	RN Nurse	40	1111	Emergency Room Center
2	1117	Stephany	Sánchez	LPN Nurse	45	1112	Outpatient Care Center
3	1118	Lisa	Lainez	Technician	50	1113	Dermatology Center

5.

Report Objective/Purpose: The business objective is to see the resident patients who were discharged this week and go check the equipment to see any issues in those rooms.

Persona This Report Is Targeting: Head of Maintenance

Decision This Report Will Support: The decision is to see what day, which technicians and how many technicians are needed to do maintenance on the equipment in those rooms for this week.

Data You Will Need To Create The Report So The Decision Can Be Made: Pa Person ID, First Name, Last Name, Patient Type, Date Discharged, Care Center ID, Care Center Name, and Room Number

Additional Details:

Query:

```
SELECT RP.Pa_Person_ID, P.First_Name, P.Last_Name, PA.Patient_Type,  
RP.Date_Discharged, CC.Care_Center_ID, CC.Care_Center_Name, B.Room_Number  
FROM Person AS P  
INNER JOIN Patient AS PA  
ON P.Person_ID = PA.Pa_Person_ID  
INNER JOIN Resident_Patient AS RP  
ON PA.Pa_Person_ID = RP.Pa_Person_ID  
INNER JOIN Bed AS B  
ON RP.Room_Number = B.Room_Number  
INNER JOIN Care_Center AS CC  
ON B.Care_Center_ID = CC.Care_Center_ID  
WHERE RP.Date_Discharged BETWEEN '2020-04-06' AND '2020-04-12';
```

Report:

	Pa_Person_ID	First_Name	Last_Name	Patient_Type	Date_Discharged	Care_Center_ID	Care_Center_Name	Room_Number
1	1128	Cindy	Comea	R	2020-04-06	1111	Emergency Room Center	1111
2	1141	Lisa	Pena	R	2020-04-07	1111	Emergency Room Center	1113
3	1145	Diego	Carlos	R	2020-04-08	1115	Surgical Unit Care Center	1115
4	1117	Stephany	Sánchez	R	2020-04-09	1115	Surgical Unit Care Center	1117
5	1154	Thomas	Partey	R	2020-04-10	1115	Surgical Unit Care Center	1119

6.

Report Objective/Purpose: The business objective is to see how many volunteers there are currently working who have between 5 and 6 hours worked this week and their work unit description. In addition, what type of volunteers are working in the care centers and where they are assigned to.

Persona This Report Is Targeting: Hospital Supervisor

Decision This Report Will Support: The decision is to see which volunteers to give more hours and which ones to give less hours based on the hours worked this week for the upcoming week.

Data You Will Need To Create The Report So The Decision Can Be Made: Vo Person ID, First Name, Last Name, Work Unit ID, Work Unit Description, Hours Worked, Care Center ID, and Care Center Name

Additional Details: In this business report I assumed what the hours worked were for this week.

Query:

```
SELECT V.Vo_Person_ID, P.First_Name, P.Last_Name, VS.Work_Unit_ID,
WU.Work_Unit_Description, VS.Hours_Worked, CC.Care_Center_ID, CC.Care_Center_Name
FROM Volunteer_Supervisor AS VS
INNER JOIN Work_Unit AS WU
ON VS.Work_Unit_ID = WU.Work_Unit_ID
INNER JOIN Volunteer AS V
ON VS.Vo_Person_ID = V.Vo_Person_ID
INNER JOIN Person AS P
ON V.Vo_Person_ID = P.Person_ID
INNER JOIN Volunteer_Assigned_Care_Center AS VACC
ON V.Vo_Person_ID = VACC.Vo_Person_ID
INNER JOIN Care_Center AS CC
ON VACC.Care_Center_ID = CC.Care_Center_ID
WHERE VS.Hours_Worked BETWEEN 5 AND 6;
```

Report:

	Vo_Person_ID	First_Name	Last_Name	Work_Unit_ID	Work_Unit_Description	Hours_Worked	Care_Center_ID	Care_Center_Name
1	1141	Lisa	Pena	1	Emergency Room	5	1111	Emergency Room Center
2	1142	Funes	Mori	2	Outpatient Care	5	1112	Outpatient Care Center
3	1143	Carlos	Pineda	3	Dermatology	6	1113	Dermatology Center
4	1144	David	Neres	4	Urology	6	1114	Urology Center

7.

Report Objective/Purpose: The business objective is to see how many appointments there are going to be in the upcoming week for dermatology and the date and time of the appointments.

Person This Report Is Targeting: Health Administrator

Decision This Report Will Support: The decision is to determine what day, which and how many dermatologists have been assigned to treat the patients this upcoming week.

Data You Will Need To Create The Report So The Decision Can Be Made: Pa Person ID, Ph Person ID, Appointment Date, Appointment Time, Care Center ID, and Care Center Name.

Additional Details:

Query:

```
SELECT APPTS.Pa_Person_ID, APPTS.Ph_Person_ID, APPTS.Appointment_Date,  
APPTS.Appointment_Time, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Appointment_Schedule AS APPTS  
INNER JOIN Care_Center AS CC  
ON APPTS.Care_Center_ID = CC.Care_Center_ID  
WHERE APPTS.Appointment_Date BETWEEN '2020-04-20' AND '2020-04-26'  
AND CC.Care_Center_ID = 1113;
```

Report:

	Pa_Person_ID	Ph_Person_ID	Appointment_Date	Appointment_Time	Care_Center_ID	Care_Center_Name
1	1128	1113	2020-04-20	11:00:00	1113	Dermatology Center
2	1143	1113	2020-04-24	02:00:00	1113	Dermatology Center

8.

Report Objective/Purpose: The business objective is to see how many resident patients can be admitted next week based on how many beds there are, the availability, and how many of each types of beds there are available as well as where they are located.

Persona This Report Is Targeting: Health Administrator

Decision This Report Will Support: The decision is to see what is the limit of resident patients that can have a bed and the type of bed it has as well as what care center is it located in for next week. Also to see if there is a need to build more rooms and buy more beds starting next month.

Data You Will Need To Create The Report So The Decision Can Be Made: Bed Number, Room Number, Is Available, Bed Type Description, Care Center ID, and Care Center Name

Additional Details:

Query:

```
SELECT B.Bed_Number, B.Room_Number, B.Is_Available,  
BT.Bed_Type_Description, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Bed AS B  
INNER JOIN Care_Center AS CC  
ON B.Care_Center_ID = CC.Care_Center_ID  
INNER JOIN Bed_Type AS BT  
ON B.Bed_Type_ID = BT.Bed_Type_ID  
WHERE B.Is_Available = 'Y'  
ORDER BY CC.Care_Center_Name;
```

Report:

	Bed_Number	Room_Number	Is_Available	Bed_Type_Description	Care_Center_ID	Care_Center_Name
1	1113	1113	Y	Clinitron Air Bed	1111	Emergency Room Center
2	1116	1116	Y	Standard Manuel Bed	1111	Emergency Room Center
3	1118	1118	Y	Clinitron Air Bed	1111	Emergency Room Center
4	1119	1119	Y	Stretcher	1115	Surgical Unit Care Center
5	1120	1120	Y	Low Bed	1115	Surgical Unit Care Center
6	1112	1112	Y	Electric Automatic Bed	1115	Surgical Unit Care Center
7	1117	1117	Y	Electric Automatic Bed	1115	Surgical Unit Care Center
8	1114	1114	Y	Stretcher	1115	Surgical Unit Care Center
9	1115	1115	Y	Low Bed	1115	Surgical Unit Care Center

9.

Report Objective/Purpose: The business objective is to see how many patients are going to have surgery next week and the date and time of the appointments.

Persona This Report Is Targeting: Health Administrator

Decision This Report Will Support: The decision is to see what day, which surgeons and how many surgeons have been assigned for next week.

Data You Will Need To Create The Report So The Decision Can Be Made: Pa Person ID, Ph Person ID, Appointment Date, Appointment Time, Care Center ID, and Care Center Name.

Additional Details:

Query:

```
SELECT APPTS.Pa_Person_ID, APPTS.Ph_Person_ID, APPTS.Appointment_Date  
APPTS.Appointment_Time, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Appointment_Schedule AS APPTS  
INNER JOIN Care_Center AS CC  
ON APPTS.Care_Center_ID = CC.Care_Center_ID  
WHERE APPTS.Appointment_Date BETWEEN '2020-04-20' AND '2020-04-26'  
AND CC.Care_Center_ID = 1115  
ORDER BY APPTS.Appointment_Date;
```

Report:

	Pa_Person_ID	Ph_Person_ID	Appointment_Date	Appointment_Time	Care_Center_ID	Care_Center_Name
1	1145	1115	2020-04-21	02:30:00	1115	Surgical Unit Care Center
2	1130	1115	2020-04-23	01:30:00	1115	Surgical Unit Care Center

10.

Report Objective/Purpose: The business objective is to see how many appointments there are going to be in the upcoming week for Urology and the date and time of the appointments.

Persona This Report Is Targeting: Health Administrator

Decision This Report Will Support: The decision is to see what day, which urologists and how many urologists have been assigned to treat the patients.

Data You Will Need To Create The Report So The Decision Can Be Made: Pa Person ID, Ph Person ID, Appointment Date, Appointment Time, Care Center ID, and Care Center Name.

Additional Details:

Query:

```
SELECT APPTS.Pa_Person_ID, APPTS.Ph_Person_ID, APPTS.Appointment_Date,  
APPTS.Appointment_Time, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Appointment_Schedule AS APPTS  
INNER JOIN Care_Center AS CC  
ON APPTS.Care_Center_ID = CC.Care_Center_ID  
WHERE APPTS.Appointment_Date BETWEEN '2020-04-21' AND '2020-04-27'  
AND CC.Care_Center_ID = 1114;
```

Report:

	Pa_Person_ID	Ph_Person_ID	Appointment_Date	Appointment_Time	Care_Center_ID	Care_Center_Name
1	1129	1114	2020-04-25	12:00:00	1114	Urology Center
2	1144	1114	2020-04-26	03:00:00	1114	Urology Center

## **Business Reports Stored Procedures**

This section shows 10 business reports store procedures that has a store procedure objective/purpose, persona this store procedure is targeting, decision this store procedure will support, data needed to create the store procedure so the decision can be made as well as the store procedure and report of the store procedure.

1.

Store Procedure Objective/Purpose: The store procedure objective is to see how many volunteers are leaving next month and the employee and physician being freed up from supervising them. In addition, what is the volunteers work unit and what care center they are from.

Persona This Store Procedure Is Targeting: Hospital Supervisor

Decision This Store Procedure Will Support: The decision it will support is to see how many more volunteers are going to be needed for next month after the volunteers leave next month.

Data You Will Need To Create The Store Procedure So The Decision Can Be Made: Vo Person ID, Em Person ID, Ph Person ID, Work Unit ID, Work Unit Description, Care Center ID, Care Center Name, and End Date

Additional Details:

Store Procedure:

```
CREATE PROCEDURE Get_Volunteers_Leaving
(@Start_Date DATE, @End_Date DATE)
AS
BEGIN
SET COUNT ON

SELECT VS.Vo_Person_ID, VS.Em_Person_ID, VS.Ph_Person_ID, VS.Work_Unit_ID,
WU.Work_Unit_Description, CC.Care_Center_ID, CC.Care_Center_Name, VS.End_Date
FROM Volunteer AS V
INNER JOIN Volunteer_Assigned_Care_Center AS VACC
ON V.Vo_Person_ID = VACC.Vo_Person_ID
INNER JOIN Care_Center AS CC
ON VACC.Care_Center_ID = CC.Care_Center_ID
INNER JOIN Volunteer_Supervisor AS VS
ON V.Vo_Person_ID = VS.Vo_Person_ID
INNER JOIN Work_Unit AS WU
ON VS.Work_Unit_ID = WU.Work_Unit_ID
WHERE VS.End_Date BETWEEN @Start_Date AND @End_Date
ORDER BY VS.End_Date;

END
```

**EXEC** Get\_Volunteers\_Leaving @Start\_Date = '2020-05-01', @End\_Date = '2020-05-31'

Report:

	Vo_Person_ID	Em_Person_ID	Ph_Person_ID	Work_Unit_ID	Work_Unit_Description	Care_Center_ID	Care_Center_Name	End_Date
1	1146	1121	1131	1	Emergency Room	1111	Emergency Room Center	2020-05-03
2	1128	1125	1135	5	Surgical Unit	1115	Surgical Unit Care Center	2020-05-06
3	1142	1117	1112	2	Outpatient Care	1112	Outpatient Care Center	2020-05-11
4	1144	1119	1114	4	Urology	1114	Urology Center	2020-05-16
5	1126	1123	1133	3	Dermatology	1113	Dermatology Center	2020-05-20

2.

Store Procedure Objective/Purpose: The store procedure objective is to see how many maintenance people are needed this upcoming week to clean the resident patient rooms that are being discharged.

Persona This Store Procedure Is Targeting: Housekeeping Management

Decision This Store Procedure Will Support: The decision it will support is to see what day, how many, and which staff to send to go clean the rooms that are going to clear this upcoming week.

Data You Will Need To Create The Store Procedure So The Decision Can Be Made – Pa Person ID, First Name, Last Name, Patient Type, Date Discharged, Care Center ID, Care Center Name, and Room Number

Additional Details:

Store Procedure:

```
CREATE PROCEDURE Get_Resident_Patients_Discharged
(@Start_Date DATE, @End_Date DATE)
AS
BEGIN
SET COUNT ON

SELECT RP.Pa_Person_ID, P.First_Name, P.Last_Name, PA.Patient_Type,
RP.Date_Discharged, CC.Care_Center_ID, CC.Care_Center_Name, B.Room_Number
FROM Person AS P
INNER JOIN Patient AS PA
ON P.Person_ID = PA.Pa_Person_ID
INNER JOIN Resident_Patient AS RP
ON PA.Pa_Person_ID = RP.Pa_Person_ID
INNER JOIN Bed AS B
ON RP.Bed_Number = B.Bed_Number
INNER JOIN Care_Center AS CC
ON B.Care_Center_ID = CC.Care_Center_ID
WHERE RP.Date_Discharged BETWEEN @Start_Date AND @End_Date
ORDER BY RP.Date_Discharged;

END

EXEC Get_Resident_Patients_Discharged @Start_Date = '2020-04-20', @End_Date = '2020-
```

04-26'

Report:

	Pa_Person_ID	First_Name	Last_Name	Patient_Type	Date_Discharged	Care_Center_ID	Care_Center_Name	Room_Number
1	1130	Jefferson	Montero	R	2020-04-20	1115	Surgical Unit Care Center	1112
2	1143	Carlos	Pineda	R	2020-04-21	1115	Surgical Unit Care Center	1114
3	1147	Cristina	Oblak	R	2020-04-22	1111	Emergency Room Center	1116
4	1119	Genesis	Gayà	R	2020-04-23	1111	Emergency Room Center	1118
5	1112	Nélson	Semedo	R	2020-04-24	1115	Surgical Unit Care Center	1120

3.

Store Procedure Objective/Purpose: The store procedure objective is to see how many outpatients visited each care center this week and the day they visited as well as the time they visited and the care center name they visited.

Persona This Store Procedure Is Targeting: Hospital Administrator

Decision This Store Procedure Will Support: The decision it will support is to see if there is a need for more employees in the care center or not for next week.

Data You Will Need To Create The Store Procedure So The Decision Can Be Made: Pa Person ID, First Name, Last Name, Patient Type, Visit Number, Visit Date, Visit Time, Care Center ID and Care Center Name.

Additional Details: In this business report I assumed that in the previous weeks the amount of outpatients visiting kept on increasing and wanted to see if this report kept in correlation with the previous weeks to see if the decision of needing more employees was needed.

Store Procedure:

CREATE PROCEDURE Get\_Outpatients\_Visited

(@Start\_Date DATE, @End\_Date DATE)

AS

BEGIN

SET COUNT ON

SELECT O.Pa\_Person\_ID, P.First\_Name, P.Last\_Name, PA.Patient\_Type, V.Visit\_Number,

V.Visit\_Date, V.Visit\_Time, CC.Care\_Center\_ID, CC.Care\_Center\_Name

FROM Person AS P

INNER JOIN Patient AS PA

ON P.Person\_ID = PA.Pa\_Person\_ID

INNER JOIN Outpatient AS O

ON PA.Pa\_Person\_ID = O.Pa\_Person\_ID

INNER JOIN Visit AS V

ON O.Pa\_Person\_ID = V.Pa\_Person\_ID

INNER JOIN Visit\_Care\_Center AS VCC

ON V.Visit\_Number = VCC.Visit\_Number

INNER JOIN Care\_Center AS CC

ON VCC.Care\_Center\_ID = CC.Care\_Center\_ID

WHERE V.Visit\_Date BETWEEN @Start\_Date AND @End\_Date

ORDER BY CC.Care\_Center\_Name;

END

`EXEC Get_Outpatients_Visited @Start_Date = '2020-04-13', @End_Date = '2020-04-19'`

Report:

	Pa_Person_ID	First_Name	Last_Name	Patient_Type	Visit_Number	Visit_Date	Visit_Time	Care_Center_ID	Care_Center_Name
1	1118	Lisa	Lainez	0	1118	2020-04-19	01:30:00	1113	Dermatology Center
2	1129	Álvaro	Morata	0	1113	2020-04-15	09:00:00	1113	Dermatology Center
3	1126	Victor	Wanyama	0	1111	2020-04-13	07:00:00	1111	Emergency Room Center
4	1146	Iker	Casillas	0	1116	2020-04-18	12:00:00	1111	Emergency Room Center
5	1127	Thomas	Lemar	0	1112	2020-04-14	08:00:00	1112	Outpatient Care Center
6	1144	David	Neres	0	1115	2020-04-17	11:00:00	1115	Surgical Unit Care Center
7	1142	Funes	Mori	0	1114	2020-04-16	10:00:00	1114	Urology Center

4.

Store Procedure Objective/Purpose: The store procedure objective is to see how many employees there are who worked between 40 and 50 hours this week and what type of employees are working in the care centers and what care centers were they assigned to.

Persona This Store Procedure Is Targeting: Hospital Administrator

Decision This Store Procedure Will Support: The decision it will support is to see which employees to give more hours and which ones to give less hours based on the hours worked this week for the upcoming week.

Data You Will Need To Create The Report So The Decision Can Be Made: Em Person ID, First Name, Last Name, Employee Type, Hours Worked, Care Center ID and Care Center Name.

Additional Details: In this business report I assumed what the hours worked were for this week.

Store Procedure:

**CREATE PROCEDURE** Get\_Employees\_Hours\_Worked

(@Min\_Hours **INT**, @Max\_Hours **INT**)

**AS**

**BEGIN**

**SET COUNT ON**

**SELECT** E.Em\_Person\_ID, P.First\_Name, P.Last\_Name, E.Employee\_Type,

EACC.Hours\_Worked, CC.Care\_Center\_ID, CC.Care\_Center\_Name

**FROM** Person **AS** P

**INNER JOIN** Employee **AS** E

**ON** P.Person\_ID = E.Em\_Person\_ID

**INNER JOIN** Employee\_Assigned\_Care\_Center **AS** EACC

**ON** E.Em\_Person\_ID = EACC.Em\_Person\_ID

**INNER JOIN** Care\_Center **AS** CC

**ON** EACC.Care\_Center\_ID = CC.Care\_Center\_ID

**WHERE** EACC.Hours\_Worked **BETWEEN** @Min\_Hours **AND** @Max\_Hours

**ORDER BY** EACC.Hours\_Worked;

**END**

**EXEC** Get\_Employees\_Hours\_Worked @Min\_Hours = **40**, @Max\_Hours = **50**

Report:

	Em_Person_ID	First_Name	Last_Name	Employee_Type	Hours_Worked	Care_Center_ID	Care_Center_Name
1	1116	Erika	Gonzalez	RN Nurse	40	1111	Emergency Room Center
2	1117	Stephany	Sánchez	LPN Nurse	45	1112	Outpatient Care Center
3	1118	Lisa	Lainez	Technician	50	1113	Dermatology Center

5.

Store Procedure Objective/Purpose: The store procedure objective is to see the resident patients who were discharged this week and go check the equipment to see any issues in those rooms.

Persona This Store Procedure Is Targeting: Head of Maintenance

Decision This Store Procedure Will Support: The decision it will support is to see what day, which technicians and how many technicians are needed to do maintenance on the equipment in those rooms for this week.

Data You Will Need To Create The Store Procedure So The Decision Can Be Made: Person ID, First Name, Last Name, Patient Type, Date Discharged, Care Center ID, Care Center Name, and Room Number

Additional Details: Using the same stored procedure as number 2 since it has similar store procedure purpose.

Store Procedure:

```
CREATE PROCEDURE Get_Resident_Patients_Discharged
(@Start_Date DATE, @End_Date DATE)
AS
BEGIN
SET COUNT ON

SELECT RP.Pa_Person_ID, P.First_Name, P.Last_Name, PA.Patient_Type,
RP.Date_Discharged, CC.Care_Center_ID, CC.Care_Center_Name, B.Room_Number
FROM Person AS P
INNER JOIN Patient AS PA
ON P.Person_ID = PA.Pa_Person_ID
INNER JOIN Resident_Patient AS RP
ON PA.Pa_Person_ID = RP.Pa_Person_ID
INNER JOIN Bed AS B
ON RP.Room_Number = B.Room_Number
INNER JOIN Care_Center AS CC
ON B.Care_Center_ID = CC.Care_Center_ID
WHERE RP.Date_Discharged BETWEEN @Start_Date AND @End_Date
ORDER BY RP.Date_Discharged;

END

EXEC Get_Resident_Patients_Discharged @Start_Date = '2020-04-06', @End_Date = '2020-
```

04-12'

Report:

	Pa_Person_ID	First_Name	Last_Name	Patient_Type	Date_Discharged	Care_Center_ID	Care_Center_Name	Room_Number
1	1128	Cindy	Correa	R	2020-04-06	1111	Emergency Room Center	1111
2	1141	Lisa	Peña	R	2020-04-07	1111	Emergency Room Center	1113
3	1145	Diego	Carlos	R	2020-04-08	1115	Surgical Unit Care Center	1115
4	1117	Stephany	Sánchez	R	2020-04-09	1115	Surgical Unit Care Center	1117
5	1154	Thomas	Partey	R	2020-04-10	1115	Surgical Unit Care Center	1119

6.

Store Procedure Objective/Purpose: The store procedure objective is to see how many volunteers there are currently working who have between 5 and 6 hours worked this week and their work unit description. In addition, what type of volunteers are working in the care centers and where they are assigned to.

Persona This Store Procedure Is Targeting: Hospital Supervisor

Decision This Store Procedure Will Support: The decision it will support is to see which volunteers to give more hours and which ones to give less hours based on the hours worked this week for the upcoming week.

Data You Will Need To Create The Store Procedure So The Decision Can Be Made: Vo Person ID, First Name, Last Name, Work Unit ID, Work Unit Description, Hours Worked, Care Center ID, and Care Center Name

Additional Details: In this business report I assumed what the hours worked were for this week.

Store Procedure:

```
CREATE PROCEDURE Get_Volunteer_Hours_Worked
(@Min_Hours INT, @Max_Hours INT)
AS
BEGIN
SET COUNT ON

SELECT V.Vo_Person_ID, P.First_Name, P.Last_Name, VS.Work_Unit_ID,
WU.Work_Unit_Description, VS.Hours_Worked, CC.Care_Center_ID, CC.Care_Center_Name
FROM Volunteer_Supervisor AS VS
INNER JOIN Work_Unit AS WU
ON VS.Work_Unit_ID = WU.Work_Unit_ID
INNER JOIN Volunteer AS V
ON VS.Vo_Person_ID = V.Vo_Person_ID
INNER JOIN Person AS P
ON V.Vo_Person_ID = P.Person_ID
INNER JOIN Volunteer_Assigned_Care_Center AS VACC
ON V.Vo_Person_ID = VACC.Vo_Person_ID
INNER JOIN Care_Center AS CC
ON VACC.Care_Center_ID = CC.Care_Center_ID
WHERE VS.Hours_Worked BETWEEN @Min_Hours AND @Max_Hours;

END
```

`EXEC Get_Volunteer_Hours_Worked @Min_Hours = 5, @Max_Hours = 6`

Report:

	Vo_Person_ID	First_Name	Last_Name	Work_Unit_ID	Work_Unit_Description	Hours_Worked	Care_Center_ID	Care_Center_Name
1	1141	Lisa	Pena	1	Emergency Room	5	1111	Emergency Room Center
2	1142	Funes	Mori	2	Outpatient Care	5	1112	Outpatient Care Center
3	1143	Carlos	Pineda	3	Dermatology	6	1113	Dermatology Center
4	1144	David	Neres	4	Urology	6	1114	Urology Center

7.

Store Procedure Objective/Purpose: The Store Procedure objective is to see how many appointments there are going to be in the upcoming week for dermatology and the date and time of the appointments.

Person This Store Procedure Is Targeting: Health Administrator

Decision This Store Procedure Will Support: The decision it will support is to determine what day, which and how many dermatologists have been assigned to treat the patients this upcoming week.

Data You Will Need To Create The Store Procedure So The Decision Can Be Made: Pa Person ID, Ph Person ID, Appointment Date, Appointment Time, Care Center ID, and Care Center Name.

Additional Details:

Store Procedure:

```
CREATE PROCEDURE Get_Appointments
(@Start_Date DATE, @End_Date DATE, @Care_Center_ID INT)
AS
BEGIN
SET COUNT ON

SELECT APPTS.Pa_Person_ID, APPTS.Ph_Person_ID, APPTS.Appointment_Date,
APPTS.Appointment_Time, CC.Care_Center_ID, CC.Care_Center_Name
FROM Appointment_Schedule AS APPTS
INNER JOIN Care_Center AS CC
ON APPTS.Care_Center_ID = CC.Care_Center_ID
WHERE APPTS.Appointment_Date BETWEEN @Start_Date AND @End_Date
AND CC.Care_Center_ID = @Care_Center_ID;

END

EXEC Get_Appointments @Start_Date = '2020-04-20', @End_Date = '2020-04-26',
@Care_Center_ID = 1113
```

Report:

	Pa_Person_ID	Ph_Person_ID	Appointment_Date	Appointment_Time	Care_Center_ID	Care_Center_Name
1	1128	1113	2020-04-20	11:00:00	1113	Dermatology Center
2	1143	1113	2020-04-24	02:00:00	1113	Dermatology Center

8.

Store Procedure Objective/Purpose: The store procedure objective is to see how many resident patients can be admitted next week based on how many beds there are, the availability, and how many of each types of beds there are available as well as where they are located.

Persona This Store Procedure Is Targeting: Health Administrator

Decision This Store Procedure Will Support: The decision it will support is to see what is the limit of resident patients that can have a bed and the type of bed it has as well as what care center is it located in for next week. Also to see if there is a need to build more rooms and buy more beds starting next month.

Data You Will Need To Create The Store Procedure So The Decision Can Be Made: Bed Number, Room Number, Is Available, Bed Type Description, Care Center ID and Care Center Name

Additional Details:

Store Procedure:

```
CREATE PROCEDURE Get_Bed_Availability
(@Availability CHAR (1))
AS
BEGIN
SET COUNT ON

SELECT B.Bed_Number, B.Room_Number, B.Is_Available,
BT.Bed_Type_Description, CC.Care_Center_ID, CC.Care_Center_Name
FROM Bed AS B
INNER JOIN Care_Center AS CC
ON B.Care_Center_ID = CC.Care_Center_ID
INNER JOIN Bed_Type AS BT
ON B.Bed_Type_ID = BT.Bed_Type_ID
WHERE B.Is_Available = @Availability
ORDER BY CC.Care_Center_Name;

END

EXEC Get_Bed_Availability @Availability = 'Y'
```

Report:

	Bed_Number	Room_Number	Is_Available	Bed_Type_Description	Care_Center_ID	Care_Center_Name
1	1113	1113	Y	Clinitron Air Bed	1111	Emergency Room Center
2	1116	1116	Y	Standard Manuel Bed	1111	Emergency Room Center
3	1118	1118	Y	Clinitron Air Bed	1111	Emergency Room Center
4	1119	1119	Y	Stretcher	1115	Surgical Unit Care Center
5	1120	1120	Y	Low Bed	1115	Surgical Unit Care Center
6	1112	1112	Y	Electric Automatic Bed	1115	Surgical Unit Care Center
7	1117	1117	Y	Electric Automatic Bed	1115	Surgical Unit Care Center
8	1114	1114	Y	Stretcher	1115	Surgical Unit Care Center
9	1115	1115	Y	Low Bed	1115	Surgical Unit Care Center

9.

Store Procedure Objective/Purpose: The store procedure objective is to see how many patients are going to have surgery next week and the date and time of the appointments.

Persona This Store Procedure Is Targeting: Health Administrator

Decision This Store Procedure Will Support: The decision it will support is to see what day, which surgeons and how many surgeons have been assigned for next week.

Data You Will Need To Create The Store Procedure So The Decision Can Be Made: Pa Person ID, Ph Person ID, Appointment Date, Appointment Time, Care Center ID, and Care Center Name.

Additional Details: Using the same stored procedure as number 7 since it has similar store procedure purpose.

Store Procedure:

```
CREATE PROCEDURE Get_Appointments
(@Start_Date DATE, @End_Date DATE, @Care_Center_ID INT)
AS
BEGIN
SET COUNT ON

SELECT APPTS.Pa_Person_ID, APPTS.Ph_Person_ID, APPTS.Appointment_Date,
APPTS.Appointment_Time, CC.Care_Center_ID, CC.Care_Center_Name
FROM Appointment_Schedule AS APPTS
INNER JOIN Care_Center AS CC
ON APPTS.Care_Center_ID = CC.Care_Center_ID
WHERE APPTS.Appointment_Date BETWEEN @Start_Date AND @End_Date
AND CC.Care_Center_ID = @Care_Center_ID;

END

EXEC Get_Appointments @Start_Date = '2020-04-20', @End_Date = '2020-04-26',
@Care_Center_ID = 1115
```

Report:

	Pa_Person_ID	Ph_Person_ID	Appointment_Date	Appointment_Time	Care_Center_ID	Care_Center_Name
1	1130	1115	2020-04-23	01:30:00	1115	Surgical Unit Care Center
2	1145	1115	2020-04-21	02:30:00	1115	Surgical Unit Care Center

10.

Store Procedure Objective/Purpose: The store procedure objective is to see how many appointments there are going to be in the upcoming week for Urology and the date and time of the appointments.

Persona This Store Procedure Is Targeting: Health Administrator

Decision This Store Procedure Will Support: The decision it will support is to see what day, which urologists and how many urologists have been assigned to treat the patients.

Data You Will Need To Create The Store Procedure So The Decision Can Be Made: Pa Person ID, Ph Person ID, Appointment Date, Appointment Time, Care Center ID, and Care Center Name.

Additional Details: Using the same stored procedure as number 7 since it has similar store procedure purpose.

Store Procedure:

```
CREATE PROCEDURE Get_Appointments
(@Start_Date DATE, @End_Date DATE, @Care_Center_ID INT)
AS
BEGIN
SET COUNT ON

SELECT APPTS.Pa_Person_ID, APPTS.Ph_Person_ID, APPTS.Appointment_Date,
APPTS.Appointment_Time, CC.Care_Center_ID, CC.Care_Center_Name
FROM Appointment_Schedule AS APPTS
INNER JOIN Care_Center AS CC
ON APPTS.Care_Center_ID = CC.Care_Center_ID
WHERE APPTS.Appointment_Date BETWEEN @Start_Date AND @End_Date
AND CC.Care_Center_ID = @Care_Center_ID;

END

EXEC Get_Appointments @Start_Date = '2020-04-21', @End_Date = '2020-04-27',
@Care_Center_ID = 1114
```

Report:

	Pa_Person_ID	Ph_Person_ID	Appointment_Date	Appointment_Time	Care_Center_ID	Care_Center_Name
1	1129	1114	2020-04-25	12:00:00	1114	Urology Center
2	1144	1114	2020-04-26	03:00:00	1114	Urology Center

## Conclusion

In this project the database tier for a Hospital Management System Application named EZMedicalGroup.com was upgraded. The application was upgraded by creating business reports and improving the performance of the full working application by creating store procedures. The Hospital Management System was upgrades to allow hospital employees to conveniently manage hospital patient activities such as assigning physicians, medical care, registration etc., using this system. The system upgrade done was possible due to the use of this methodology being used which is Planning, Analysis, Design, Implementation, and Maintenance/Administration. The business reports and improvement of performance of the full working application by creating store procedures were also done through this.

# **Hospital Management System Database Design &**

## **Implementation Version 3.0**

### **Upgrade Objectives**

To enhance performance & security in the Hospital Management System Application named EZMedicalGroup.com some features need to be added. Some features are creating indexes for the reports such as non-clustered indexes to improve search performance. Also analyze, design, and create views for the queries in the reports as well as queries to test the views.

### **Non-Clustered Indexes**

1.

Description: Creating non-clustered indexes for the first report query which is to see how many volunteers are leaving next month and the employee and physician being freed up from supervising them. In addition, what is the volunteers work unit and what care center they are from.

Report Query:

```
SELECT VS.Vo_Person_ID, VS.Em_Person_ID, VS.Ph_Person_ID, VS.Work_Unit_ID,  
WU.Work_Unit_Description, CC.Care_Center_ID, CC.Care_Center_Name, VS.End_Date  
FROM Volunteer AS V  
  
INNER JOIN Volunteer_Assigned_Care_Center AS VACC  
  
ON V.Vo_Person_ID = VACC.Vo_Person_ID  
  
INNER JOIN Care_Center AS CC  
  
ON VACC.Care_Center_ID = CC.Care_Center_ID  
  
INNER JOIN Volunteer_Supervisor AS VS  
  
ON V.Vo_Person_ID = VS.Vo_Person_ID  
  
INNER JOIN Work_Unit AS WU  
  
ON VS.Work_Unit_ID = WU.Work_Unit_ID  
  
WHERE VS.End_Date BETWEEN '2020-05-01' AND '2020-05-31'  
  
ORDER BY VS.End_Date;
```

Indexes Syntax:

```
CREATE NONCLUSTERED INDEX IX_VACC_Care_Center_ID ON  
Volunteer_Assigned_Care_Center (Care_Center_ID);
```

```
CREATE NONCLUSTERED INDEX IX_VS_Volunteer_ID ON Volunteer_Supervisor  
(Vo_Person_ID);
```

```
CREATE NONCLUSTERED INDEX IX_VS_Work_Unit_ID ON Volunteer_Supervisor  
(Work_Unit_ID);
```

```
CREATE NONCLUSTERED INDEX IX_VS_End_Date ON Volunteer_Supervisor  
(End_Date);
```

Results:

The screenshot shows the Object Explorer pane of SQL Server Management Studio. It displays the schema for two tables: `dbo.Volunteer_Assigned_Care_Center` and `dbo.Volunteer_Supervisor`. Both tables have their primary keys highlighted with a blue border.

- dbo.Volunteer\_Assigned\_Care\_Center:**
  - Columns
  - Keys
  - Constraints
  - Triggers
  - Indexes
    - IX\_VACC\_Care\_Center\_ID (Non-Unique, Non-Clustered)
    - PK\_Voluntee\_5D1615536BB200D1 (Clustered)
- dbo.Volunteer\_Supervisor:**
  - Columns
  - Keys
  - Constraints
  - Triggers
  - Indexes
    - IX\_VS\_End\_Date (Non-Unique, Non-Clustered)
    - IX\_VS\_Volunteer\_ID (Non-Unique, Non-Clustered)
    - IX\_VS\_Work\_Unit\_ID (Non-Unique, Non-Clustered)
    - PK\_Voluntee\_C85DF247819FA2B1 (Clustered)

2.

Description: Creating non-clustered indexes for the second report query which is to see how many maintenance people are needed this upcoming week to clean the resident patient rooms that are being discharged.

Report Query:

```
SELECT RP.Pa_Person_ID, P.First_Name, P.Last_Name, PA.Patient_Type,  
RP.Date_Discharged, CC.Care_Center_ID, CC.Care_Center_Name, B.Room_Number  
FROM Person AS P  
INNER JOIN Patient AS PA  
ON P.Person_ID = PA.Pa_Person_ID  
INNER JOIN Resident_Patient AS RP  
ON PA.Pa_Person_ID = RP.Pa_Person_ID  
INNER JOIN Bed AS B  
ON RP.Bed_Number = B.Bed_Number  
INNER JOIN Care_Center AS CC  
ON B.Care_Center_ID = CC.Care_Center_ID  
WHERE RP.Date_Discharged BETWEEN '2020-04-20' AND '2020-04-26'  
ORDER BY RP.Date_Discharged;
```

Indexes Syntax:

```
CREATE NONCLUSTERED INDEX IX_RP_Bed_Number ON Resident_Patient  
(Bed_Number);  
  
CREATE NONCLUSTERED INDEX IX_RP_Date_Discharged ON Resident_Patient  
(Date_Discharged);  
  
CREATE NONCLUSTERED INDEX IX_B_Care_Center_ID ON Bed  
(Care_Center_ID);
```

Results:

[-]  dbo.Resident_Patient
[+]  Columns
[+]  Keys
[+]  Constraints
[+]  Triggers
[-]  Indexes
<span style="color: blue;">[+]</span> IX_RP_Bed_Number (Non-Unique, Non-Clustered)
<span style="color: blue;">[+]</span> IX_RP_Date_Discharged (Non-Unique, Non-Clustered)
<span style="color: red;">[-]</span> PK_Resident_3650475C3BFEA525 (Clustered)
[-]  dbo.Bed
[+]  Columns
[+]  Keys
[+]  Constraints
[+]  Triggers
[-]  Indexes
<span style="color: blue;">[+]</span> IX_B_Care_Center_ID (Non-Unique, Non-Clustered)
<span style="color: red;">[-]</span> PK_Bed_9DFCDAE2F37022DD (Clustered)

3.

Description: Creating non-clustered indexes for the third report query which is to see how many outpatients visited each care center this week and the day they visited as well as the time they visited and the care center name they visited.

Report Query:

```
SELECT O.Pa_Person_ID, P.First_Name, P.Last_Name, PA.Patient_Type,  
V.Visit_Number, V.Visit_Date, V.Visit_Time, CC.Care_Center_ID,  
CC.Care_Center_Name  
FROM Person AS P  
INNER JOIN Patient AS PA  
ON P.Person_ID = PA.Pa_Person_ID  
INNER JOIN Outpatient AS O  
ON PA.Pa_Person_ID = O.Pa_Person_ID  
INNER JOIN Visit AS V  
ON O.Pa_Person_ID = V.Pa_Person_ID  
INNER JOIN Visit_Care_Center AS VCC  
ON V.Visit_Number = VCC.Visit_Number  
INNER JOIN Care_Center AS CC  
ON VCC.Care_Center_ID = CC.Care_Center_ID  
WHERE V.Visit_Date BETWEEN '2020-04-13' AND '2020-04-19'  
ORDER BY CC.Care_Center_Name;
```

Indexes Syntax:

```
CREATE NONCLUSTERED INDEX IX_VCC_Visit_Number ON Visit_Care_Center  
(Visit_Number);
```

```
CREATE NONCLUSTERED INDEX IX_VCC_Care_Center_ID ON Visit_Care_Center  
(Care_Center_ID);
```

```
CREATE NONCLUSTERED INDEX IX_V_Visit_Date ON Visit  
(Visit_Date);
```

```
CREATE NONCLUSTERED INDEX IX_V_Pa_Person_ID ON Visit  
(Pa_Person_ID);
```

```
CREATE NONCLUSTERED INDEX IX_CC_Care_Center_Name ON Care_Center  
(Care_Center_Name);
```

Results:

[-]  dbo.Visit_Care_Center
[+]  Columns
[+]  Keys
[+]  Constraints
[+]  Triggers
[-]  Indexes
<span style="color: blue;">↳</span> IX_VCC_Care_Center_ID (Non-Unique, Non-Clustered)
<span style="color: blue;">↳</span> IX_VCC_Visit_Number (Non-Unique, Non-Clustered)
<span style="color: red;">➥</span> PK_Visit_Ca_07EF1358AB540861 (Clustered)
[-]  dbo.Visit
[+]  Columns
[+]  Keys
[+]  Constraints
[+]  Triggers
[-]  Indexes
<span style="color: blue;">↳</span> IX_V_Pa_Person_ID (Non-Unique, Non-Clustered)
<span style="color: blue;">↳</span> IX_V_Visit_Date (Non-Unique, Non-Clustered)
<span style="color: red;">➥</span> PK_Visit_8100804D2C04C959 (Clustered)
[-]  dbo.Care_Center
[+]  Columns
[+]  Keys
[+]  Constraints
[+]  Triggers
[-]  Indexes
<span style="color: blue;">↳</span> IX_CC_Care_Center_Name (Non-Unique, Non-Clustered)
<span style="color: red;">➥</span> PK_Care_Cen_AF5C00F85CA3D319 (Clustered)

4.

Description: Creating secondary key indexes for the fourth report query which is to see how many employees there are who worked between 40 and 50 hours this week and what type of employees are working in the care centers and what care centers were they assigned to.

Report Query:

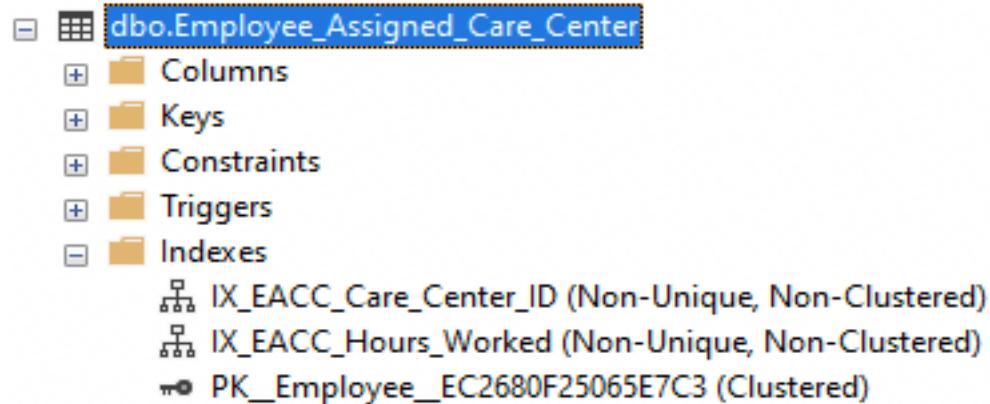
```
SELECT E.Em_Person_ID, P.First_Name, P.Last_Name, E.Employee_Type,  
EACC.Hours_Worked, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Person AS P  
INNER JOIN Employee AS E  
ON P.Person_ID = E.Em_Person_ID  
INNER JOIN Employee_Assigned_Care_Center AS EACC  
ON E.Em_Person_ID = EACC.Em_Person_ID  
INNER JOIN Care_Center AS CC  
ON EACC.Care_Center_ID = CC.Care_Center_ID  
WHERE EACC.Hours_Worked BETWEEN 40 AND 50  
ORDER BY EACC.Hours_Worked;
```

Indexes Syntax:

```
CREATE NONCLUSTERED INDEX IX_EACC_Care_Center_ID ON  
Employee_Assigned_Care_Center (Care_Center_ID);
```

```
CREATE NONCLUSTERED INDEX IX_EACC_Hours_Worked ON  
Employee_Assigned_Care_Center (Hours_Worked);
```

Results:



5.

Description: Creating non-clustered indexes for the fifth report query which is to see the resident patients who were discharged this week and go check the equipment to see any issues in those rooms.

Report Query:

```
SELECT RP.Pa_Person_ID, P.First_Name, P.Last_Name, PA.Patient_Type,  
RP.Date_Discharged, CC.Care_Center_ID, CC.Care_Center_Name, B.Room_Number  
FROM Person AS P  
INNER JOIN Patient AS PA  
ON P.Person_ID = PA.Pa_Person_ID  
INNER JOIN Resident_Patient AS RP  
ON PA.Pa_Person_ID = RP.Pa_Person_ID  
INNER JOIN Bed AS B  
ON RP.Room_Number = B.Room_Number  
INNER JOIN Care_Center AS CC  
ON B.Care_Center_ID = CC.Care_Center_ID  
WHERE RP.Date_Discharged BETWEEN '2020-04-06' AND '2020-04-12';
```

Indexes Syntax:

```
CREATE NONCLUSTERED INDEX IX_B_Room_Number ON Bed  
(Room_Number);
```

```
CREATE NONCLUSTERED INDEX IX_RP_Room_Number ON Resident_Patient  
(Room_Number);
```

Results:

dbo.Bed
+ Columns
+ Keys
+ Constraints
+ Triggers
- Indexes
IX_B_Care_Center_ID (Non-Unique, Non-Clustered)
IX_B_Room_Number (Non-Unique, Non-Clustered)
PK_Bed_9DFCDAE2F37022DD (Clustered)

  
| dbo.Resident\_Patient |
| + Columns |
| + Keys |
| + Constraints |
| + Triggers |
| - Indexes |
| IX\_RP\_Bed\_Number (Non-Unique, Non-Clustered) |
| IX\_RP\_Date\_Discharged (Non-Unique, Non-Clustered) |
| IX\_RP\_Room\_Number (Non-Unique, Non-Clustered) |
| PK\_Resident\_3650475C3BFEA525 (Clustered) |

6.

Description: Creating non-clustered indexes for the sixth report query which is to see how many volunteers there are currently working who have between 5 and 6 hours worked and work unit description. In addition, what type of volunteers are working in the care centers and where they are assigned to.

Report Query:

```
SELECT V.Vo_Person_ID, P.First_Name, P.Last_Name, VS.Work_Unit_ID,  
WU.Work_Unit_Description, VS.Hours_Worked, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Volunteer_Supervisor AS VS  
INNER JOIN Work_Unit AS WU  
ON VS.Work_Unit_ID = WU.Work_Unit_ID  
INNER JOIN Volunteer AS V  
ON VS.Vo_Person_ID = V.Vo_Person_ID  
INNER JOIN Person AS P  
ON V.Vo_Person_ID = P.Person_ID  
INNER JOIN Volunteer_Assigned_Care_Center AS VACC  
ON V.Vo_Person_ID = VACC.Vo_Person_ID  
INNER JOIN Care_Center AS CC  
ON VACC.Care_Center_ID = CC.Care_Center_ID  
WHERE VS.Hours_Worked BETWEEN 5 AND 6;
```

Indexes Syntax:

```
CREATE NONCLUSTERED INDEX IX_VS_Hours_Worked ON Volunteer_Supervisor  
(Hours_Worked)
```

Results:

The screenshot shows the object explorer in SQL Server Management Studio. A tree view is displayed under the node 'dbo.Volunteer\_Supervisor'. The tree includes categories for Columns, Keys, Constraints, Triggers, and Indexes. Under Indexes, there are four entries: IX\_VS\_End\_Date, IX\_VS\_Hours\_Worked, IX\_VS\_Volunteer\_ID, and IX\_VS\_Work\_Unit\_ID, all listed as Non-Unique, Non-Clustered. Below them is the PK\_Volunteer\_C85DF247819FA2B1 index, which is Clustered.

7.

Description: Creating non-clustered indexes for the seventh report query which is to see how many appointments there are going to be in the upcoming week for dermatology and the date and time of the appointments.

Report Query:

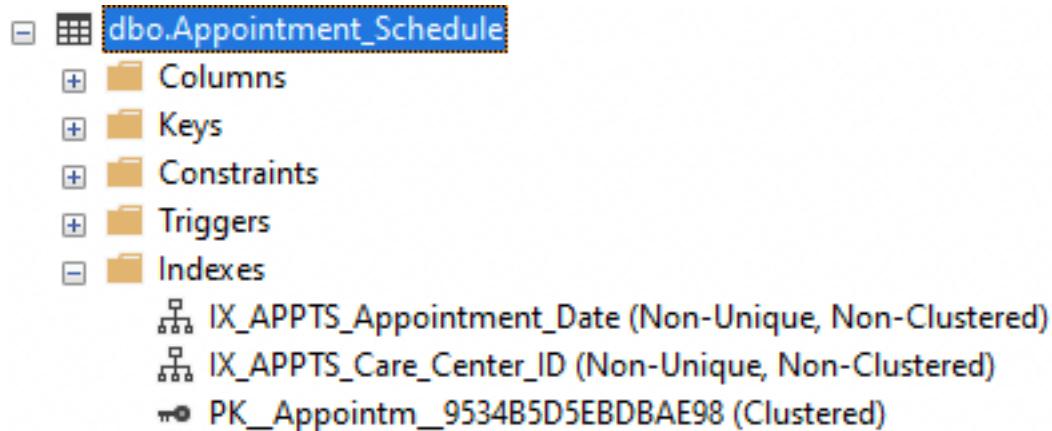
```
SELECT APPTS.Pa_Person_ID, APPTS.Ph_Person_ID, APPTS.Appointment_Date,  
APPTS.Appointment_Time, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Appointment_Schedule AS APPTS  
INNER JOIN Care_Center AS CC  
ON APPTS.Care_Center_ID = CC.Care_Center_ID  
WHERE APPTS.Appointment_Date BETWEEN '2020-04-20' AND '2020-04-26'  
AND CC.Care_Center_ID = 1113;
```

Indexes Syntax:

```
CREATE NONCLUSTERED INDEX IX_APPTS_Care_Center_ID ON Appointment_Schedule  
(Care_Center_ID);
```

```
CREATE NONCLUSTERED INDEX IX_APPTS_Appointment_Date ON  
Appointment_Schedule (Appointment_Date);
```

Results:



8.

Description: Creating non-clustered indexes for the eighth report query which is to see how many resident patients can be admitted next week based on how many beds there are, the availability, and how many of each types of beds there are available as well as where they are located.

Report Query:

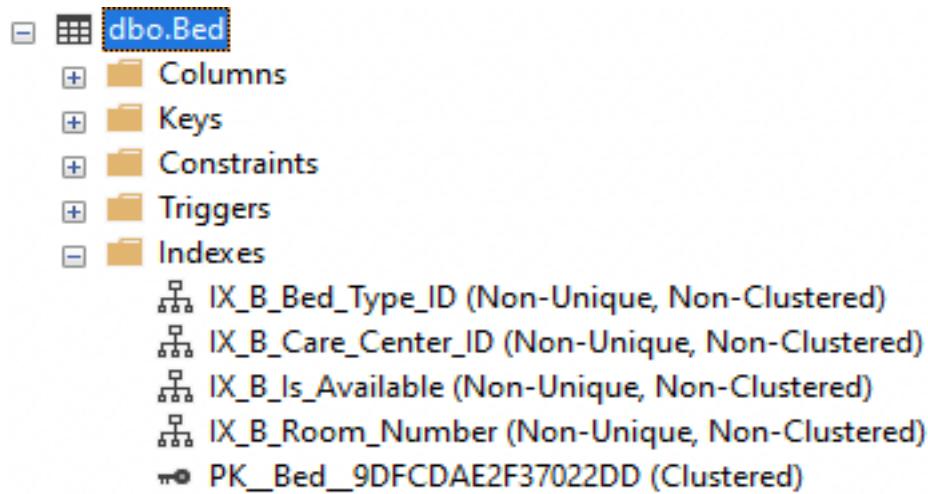
```
SELECT B.Bed_Number, B.Room_Number, B.Is_Available,  
BT.Bed_Type_Description, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Bed AS B  
INNER JOIN Care_Center AS CC  
ON B.Care_Center_ID = CC.Care_Center_ID  
INNER JOIN Bed_Type AS BT  
ON B.Bed_Type_ID = BT.Bed_Type_ID  
WHERE B.Is_Available = 'Y'  
ORDER BY CC.Care_Center_Name;
```

Indexes Syntax:

```
CREATE NONCLUSTERED INDEX IX_B_Bed_Type_ID ON Bed (Bed_Type_ID);
```

```
CREATE NONCLUSTERED INDEX IX_B_Is_Available ON Bed (Is_Available);
```

Results:



9.

Description: Creating non-clustered indexes for the ninth report query which is to see how many patients are going to have surgery next week and the date and time of the appointments.

Report Query:

```
SELECT APPTS.Pa_Person_ID, APPTS.Ph_Person_ID, APPTS.Appointment_Date  
APPTS.Appointment_Time, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Appointment_Schedule AS APPTS  
INNER JOIN Care_Center AS CC  
ON APPTS.Care_Center_ID = CC.Care_Center_ID  
WHERE APPTS.Appointment_Date BETWEEN '2020-04-20' AND '2020-04-26'  
AND CC.Care_Center_ID = 1115  
ORDER BY APPTS.Appointment_Date;
```

Indexes Syntax: Indexes already created in report 7.

Results: Already done in report 7.

10.

Description: Creating non-clustered indexes for the tenth report query which is to see how many appointments there are going to be in the upcoming week for Urology and the date and time of the appointments.

Report Query:

```
SELECT APPTS.Pa_Person_ID, APPTS.Ph_Person_ID, APPTS.Appointment_Date,  
APPTS.Appointment_Time, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Appointment_Schedule AS APPTS  
INNER JOIN Care_Center AS CC  
ON APPTS.Care_Center_ID = CC.Care_Center_ID  
WHERE APPTS.Appointment_Date BETWEEN '2020-04-21' AND '2020-04-27'  
AND CC.Care_Center_ID = 1114;
```

Indexes Syntax: Indexes already created in report 7.

Results: Already done in report 7.

## Views

1.

Description: Creating a view for the first report query which is about how many volunteers are leaving next month and the employee and physician being freed up from supervising them. In addition, what is the volunteers work unit and what care center they are from.

Report Query:

```
SELECT VS.Vo_Person_ID, VS.Em_Person_ID, VS.Ph_Person_ID,  
VS.Work_Unit_ID, WU.Work_Unit_Description, CC.Care_Center_ID,  
CC.Care_Center_Name, VS.End_Date  
FROM Volunteer AS V  
INNER JOIN Volunteer_Assigned_Care_Center AS VACC  
ON V.Vo_Person_ID = VACC.Vo_Person_ID  
INNER JOIN Care_Center AS CC  
ON VACC.Care_Center_ID = CC.Care_Center_ID  
INNER JOIN Volunteer_Supervisor AS VS  
ON V.Vo_Person_ID = VS.Vo_Person_ID  
INNER JOIN Work_Unit AS WU  
ON VS.Work_Unit_ID = WU.Work_Unit_ID  
WHERE VS.End_Date BETWEEN '2020-05-01' AND '2020-05-31'  
ORDER BY VS.End_Date;
```

View Created:

```
CREATE VIEW Volunteers_Supervisors_Ending AS  
  
SELECT VS.Vo_Person_ID, VS.Em_Person_ID, VS.Ph_Person_ID,  
VS.Work_Unit_ID, WU.Work_Unit_Description, CC.Care_Center_ID,  
CC.Care_Center_Name, VS.End_Date  
  
FROM Volunteer AS V  
  
INNER JOIN Volunteer_Assigned_Care_Center AS VACC  
  
ON V.Vo_Person_ID = VACC.Vo_Person_ID  
  
INNER JOIN Care_Center AS CC  
  
ON VACC.Care_Center_ID = CC.Care_Center_ID  
  
INNER JOIN Volunteer_Supervisor AS VS  
  
ON V.Vo_Person_ID = VS.Vo_Person_ID  
  
INNER JOIN Work_Unit AS WU  
  
ON VS.Work_Unit_ID = WU.Work_Unit_ID;
```

Results:



2.

Description: Creating a view for the second report query which is about how many maintenance people are needed this upcoming week to clean the resident patient rooms that are being discharged.

Report Query:

```
SELECT RP.Pa_Person_ID, P.First_Name, P.Last_Name, PA.Patient_Type,  
RP.Date_Discharged, CC.Care_Center_ID, CC.Care_Center_Name, B.Room_Number  
FROM Person AS P  
  
INNER JOIN Patient AS PA  
  
ON P.Person_ID = PA.Pa_Person_ID  
  
INNER JOIN Resident_Patient AS RP  
  
ON PA.Pa_Person_ID = RP.Pa_Person_ID  
  
INNER JOIN Bed AS B  
  
ON RP.Bed_Number = B.Bed_Number  
  
INNER JOIN Care_Center AS CC  
  
ON B.Care_Center_ID = CC.Care_Center_ID  
  
WHERE RP.Date_Discharged BETWEEN '2020-04-20' AND '2020-04-26'  
ORDER BY RP.Date_Discharged;
```

View Created:

```
CREATE VIEW Resident_Patient_Discharged AS  
SELECT RP.Pa_Person_ID, P.First_Name, P.Last_Name, PA.Patient_Type,  
RP.Date_Discharged, CC.Care_Center_ID, CC.Care_Center_Name, B.Room_Number  
FROM Person AS P  
INNER JOIN Patient AS PA  
ON P.Person_ID = PA.Pa_Person_ID  
INNER JOIN Resident_Patient AS RP  
ON PA.Pa_Person_ID = RP.Pa_Person_ID  
INNER JOIN Bed AS B  
ON RP.Bed_Number = B.Bed_Number  
INNER JOIN Care_Center AS CC  
ON B.Care_Center_ID = CC.Care_Center_ID;
```

Results:



3.

Description: Creating a view for the third report query which is about how many outpatients visited each care center this week and the day they visited as well as the time they visited and the care center name they visited.

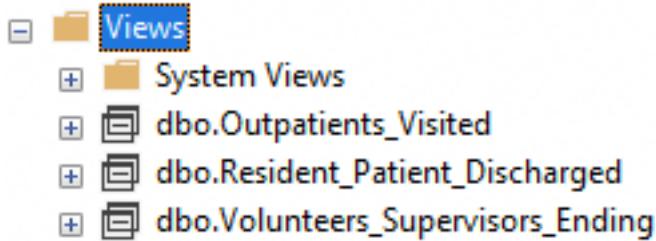
Report Query:

```
SELECT O.Pa_Person_ID, P.First_Name, P.Last_Name, PA.Patient_Type, V.Visit_Number,  
V.Visit_Date, V.Visit_Time, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Person AS P  
INNER JOIN Patient AS PA  
ON P.Person_ID = PA.Pa_Person_ID  
INNER JOIN Outpatient AS O  
ON PA.Pa_Person_ID = O.Pa_Person_ID  
INNER JOIN Visit AS V  
ON O.Pa_Person_ID = V.Pa_Person_ID  
INNER JOIN Visit_Care_Center AS VCC  
ON V.Visit_Number = VCC.Visit_Number  
INNER JOIN Care_Center AS CC  
ON VCC.Care_Center_ID = CC.Care_Center_ID  
WHERE V.Visit_Date BETWEEN '2020-04-13' AND '2020-04-19'  
ORDER BY CC.Care_Center_Name;
```

View Created:

```
CREATE VIEW Outpatients_Visited AS  
SELECT O.Pa_Person_ID, P.First_Name, P.Last_Name, PA.Patient_Type, V.Visit_Number,  
V.Visit_Date, V.Visit_Time, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Person AS P  
INNER JOIN Patient AS PA  
ON P.Person_ID = PA.Pa_Person_ID  
INNER JOIN Outpatient AS O  
ON PA.Pa_Person_ID = O.Pa_Person_ID  
INNER JOIN Visit AS V  
ON O.Pa_Person_ID = V.Pa_Person_ID  
INNER JOIN Visit_Care_Center AS VCC  
ON V.Visit_Number = VCC.Visit_Number  
INNER JOIN Care_Center AS CC  
ON VCC.Care_Center_ID = CC.Care_Center_ID;
```

Results:



4.

Description: Creating a view for the fourth report query which is about how many employees there are who worked between 40 and 50 hours this week and what type of employees are working in the care centers and what care centers were they assigned to.

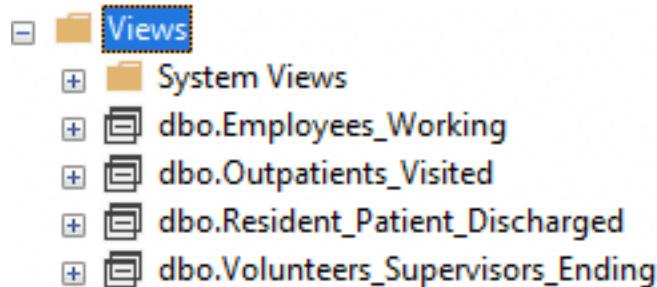
Report Query:

```
SELECT E.Em_Person_ID, P.First_Name, P.Last_Name, E.Employee_Type,  
EACC.Hours_Worked, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Person AS P  
INNER JOIN Employee AS E  
ON P.Person_ID = E.Em_Person_ID  
INNER JOIN Employee_Assigned_Care_Center AS EACC  
ON E.Em_Person_ID = EACC.Em_Person_ID  
INNER JOIN Care_Center AS CC  
ON EACC.Care_Center_ID = CC.Care_Center_ID  
WHERE EACC.Hours_Worked BETWEEN 40 AND 50  
ORDER BY EACC.Hours_Worked;
```

View Created:

```
CREATE VIEW Employees_Working AS  
SELECT E.Em_Person_ID, P.First_Name, P.Last_Name, E.Employee_Type,  
EACC.Hours_Worked, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Person AS P  
INNER JOIN Employee AS E  
ON P.Person_ID = E.Em_Person_ID  
INNER JOIN Employee_Assigned_Care_Center AS EACC  
ON E.Em_Person_ID = EACC.Em_Person_ID  
INNER JOIN Care_Center AS CC  
ON EACC.Care_Center_ID = CC.Care_Center_ID;
```

Results:



5.

Description: Creating a view for the fifth report query which is to see the resident patients who were discharged this week and go check the equipment to see any issues in those rooms.

Report Query:

```
SELECT RP.Pa_Person_ID, P.First_Name, P.Last_Name, PA.Patient_Type,  
RP.Date_Discharged, CC.Care_Center_ID, CC.Care_Center_Name, B.Room_Number  
FROM Person AS P  
INNER JOIN Patient AS PA  
ON P.Person_ID = PA.Pa_Person_ID  
INNER JOIN Resident_Patient AS RP  
ON PA.Pa_Person_ID = RP.Pa_Person_ID  
INNER JOIN Bed AS B  
ON RP.Room_Number = B.Room_Number  
INNER JOIN Care_Center AS CC  
ON B.Care_Center_ID = CC.Care_Center_ID  
WHERE RP.Date_Discharged BETWEEN '2020-04-06' AND '2020-04-12';
```

View Created:

Same view as Report Query 2

Results:

Same result as Report Query 2

6.

Description: Creating a view for the sixth report query which is about how many volunteers there are currently working who have between 5 and 6 hours worked and their work unit description. In addition, what type of volunteers are working in the care centers and where they are assigned to.

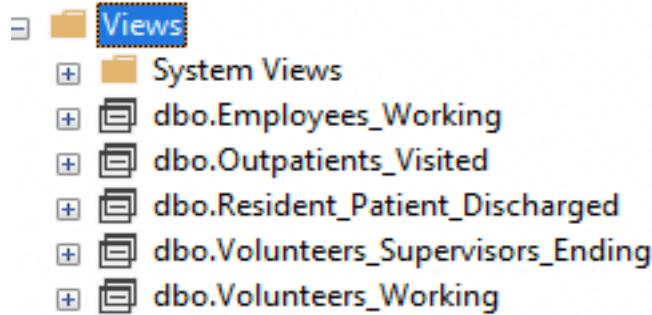
Report Query:

```
SELECT V.Vo_Person_ID, P.First_Name, P.Last_Name, VS.Work_Unit_ID,  
WU.Work_Unit_Description, VS.Hours_Worked, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Volunteer_Supervisor AS VS  
INNER JOIN Work_Unit AS WU  
ON VS.Work_Unit_ID = WU.Work_Unit_ID  
INNER JOIN Volunteer AS V  
ON VS.Vo_Person_ID = V.Vo_Person_ID  
INNER JOIN Person AS P  
ON V.Vo_Person_ID = P.Person_ID  
INNER JOIN Volunteer_Assigned_Care_Center AS VACC  
ON V.Vo_Person_ID = VACC.Vo_Person_ID  
INNER JOIN Care_Center AS CC  
ON VACC.Care_Center_ID = CC.Care_Center_ID  
WHERE VS.Hours_Worked BETWEEN 5 AND 6;
```

View Created:

```
CREATE VIEW Volunteers_Working AS  
SELECT V.Vo_Person_ID, P.First_Name, P.Last_Name, VS.Work_Unit_ID,  
WU.Work_Unit_Description, VS.Hours_Worked, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Volunteer_Supervisor AS VS  
INNER JOIN Work_Unit AS WU  
ON VS.Work_Unit_ID = WU.Work_Unit_ID  
INNER JOIN Volunteer AS V  
ON VS.Vo_Person_ID = V.Vo_Person_ID  
INNER JOIN Person AS P  
ON V.Vo_Person_ID = P.Person_ID  
INNER JOIN Volunteer_Assigned_Care_Center AS VACC  
ON V.Vo_Person_ID = VACC.Vo_Person_ID  
INNER JOIN Care_Center AS CC  
ON VACC.Care_Center_ID = CC.Care_Center_ID;
```

Results:



7.

Description: Creating a view for the seventh report query which is about how many appointments there are going to be in the upcoming week for dermatology and the date and time of the appointments.

Report Query:

```
SELECT APPTS.Pa_Person_ID, APPTS.Ph_Person_ID, APPTS.Appointment_Date,  
APPTS.Appointment_Time, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Appointment_Schedule AS APPTS  
INNER JOIN Care_Center AS CC  
ON APPTS.Care_Center_ID = CC.Care_Center_ID  
WHERE APPTS.Appointment_Date BETWEEN '2020-04-20' AND '2020-04-26'  
AND CC.Care_Center_ID = 1113;
```

View Created:

```
CREATE VIEW Appointments AS  
SELECT APPTS.Pa_Person_ID, APPTS.Ph_Person_ID, APPTS.Appointment_Date,  
APPTS.Appointment_Time, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Appointment_Schedule AS APPTS  
INNER JOIN Care_Center AS CC  
ON APPTS.Care_Center_ID = CC.Care_Center_ID;
```

Results:



8.

Description: Creating a view for the eighth report query which is about how many resident patients can be admitted next week based on how many beds there are, the availability, and how many of each types of beds there are available as well as where they are located.

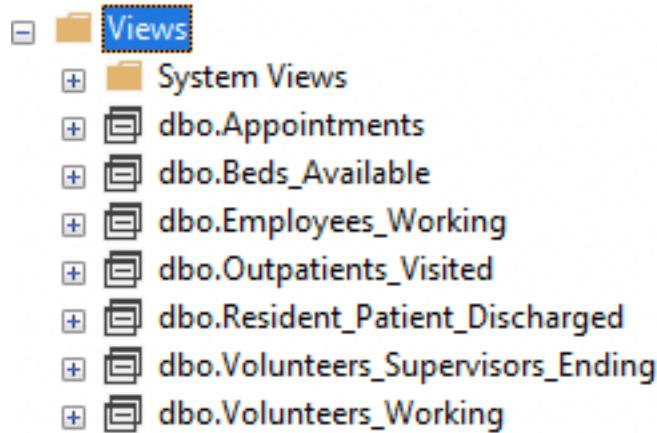
Report Query:

```
SELECT B.Bed_Number, B.Room_Number, B.Is_Available,  
BT.Bed_Type_Description, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Bed AS B  
INNER JOIN Care_Center AS CC  
ON B.Care_Center_ID = CC.Care_Center_ID  
INNER JOIN Bed_Type AS BT  
ON B.Bed_Type_ID = BT.Bed_Type_ID  
WHERE B.Is_Available = 'Y'  
ORDER BY CC.Care_Center_Name;
```

View Created:

```
CREATE VIEW Beds_Available AS  
SELECT B.Bed_Number, B.Room_Number, B.Is_Available,  
BT.Bed_Type_Description, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Bed AS B  
INNER JOIN Care_Center AS CC  
ON B.Care_Center_ID = CC.Care_Center_ID  
INNER JOIN Bed_Type AS BT  
ON B.Bed_Type_ID = BT.Bed_Type_ID;
```

Results:



9.

Description: Creating a view for the ninth report query which is about how many patients are going to have surgery next week and the date and time of the appointments.

Report Query:

```
SELECT APPTS.Pa_Person_ID, APPTS.Ph_Person_ID, APPTS.Appointment_Date  
APPTS.Appointment_Time, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Appointment_Schedule AS APPTS  
INNER JOIN Care_Center AS CC  
ON APPTS.Care_Center_ID = CC.Care_Center_ID  
WHERE APPTS.Appointment_Date BETWEEN '2020-04-20' AND '2020-04-26'  
AND CC.Care_Center_ID = 1115  
ORDER BY APPTS.Appointment_Date;
```

View Created:

Same view as Report Query 7

Results:

Same result as Report Query 7

10.

Description: Creating a view for the tenth report query which is about how many appointments there are going to be in the upcoming week for Urology and the date and time of the appointments.

Report Query:

```
SELECT APPTS.Pa_Person_ID, APPTS.Ph_Person_ID, APPTS.Appointment_Date,  
APPTS.Appointment_Time, CC.Care_Center_ID, CC.Care_Center_Name  
FROM Appointment_Schedule AS APPTS  
INNER JOIN Care_Center AS CC  
ON APPTS.Care_Center_ID = CC.Care_Center_ID  
WHERE APPTS.Appointment_Date BETWEEN '2020-04-21' AND '2020-04-27'  
AND CC.Care_Center_ID = 1114;
```

View Created:

Same view as Report Query 7

Results:

Same result as Report Query 7

## **New Business Reports That Use The New Views**

1.

Description: Creating a business report query for the first view of the report query.

What The Report Does: The report objective is to see how many volunteers are leaving next month and the employee and physician being freed up from supervising them. In addition, what is the volunteers work unit and what care center they are from.

View Query:

```
SELECT Vo_Person_ID, Em_Person_ID, Ph_Person_ID, Work_Unit_ID,  
Work_Unit_Description, Care_Center_ID, Care_Center_Name, End_Date  
FROM Volunteers_Supervisors_Ending  
  
WHERE End_Date BETWEEN '2020-05-01' AND '2020-05-31'  
  
ORDER BY End_Date;
```

Results:

	Vo_Person_ID	Em_Person_ID	Ph_Person_ID	Work_Unit_ID	Work_Unit_Description	Care_Center_ID	Care_Center_Name	End_Date
1	1146	1121	1131	1	Emergency Room	1111	Emergency Room Center	2020-05-03
2	1128	1125	1135	5	Surgical Unit	1115	Surgical Unit Care Center	2020-05-06
3	1142	1117	1112	2	Outpatient Care	1112	Outpatient Care Center	2020-05-11
4	1144	1119	1114	4	Urology	1114	Urology Center	2020-05-16
5	1126	1123	1133	3	Dermatology	1113	Dermatology Center	2020-05-20

2.

Description: Creating a business report query for the second view of the report query.

What The Report Does: The report objective is to see how many maintenance people are needed this upcoming week to clean the resident patient rooms that are being discharged.

View Query:

```
SELECT Pa_Person_ID, First_Name, Last_Name, Patient_Type, Date_Discharged,  
Care_Center_ID, Care_Center_Name, Room_Number  
FROM Resident_Patient_Discharged  
WHERE Date_Discharged BETWEEN '2020-04-20' AND '2020-04-26'  
ORDER BY Date_Discharged;
```

Results:

	Pa_Person_ID	First_Name	Last_Name	Patient_Type	Date_Discharged	Care_Center_ID	Care_Center_Name	Room_Number
1	1130	Jefferson	Montero	R	2020-04-20	1115	Surgical Unit Care Center	1112
2	1143	Carlos	Pineda	R	2020-04-21	1115	Surgical Unit Care Center	1114
3	1147	Cristina	Oblak	R	2020-04-22	1111	Emergency Room Center	1116
4	1119	Genesis	Gayà	R	2020-04-23	1111	Emergency Room Center	1118
5	1112	Nélson	Semedo	R	2020-04-24	1115	Surgical Unit Care Center	1120

3.

Description: Creating a business report for the third view of the report query.

What The Report Does: The report objective is to see how many outpatients visited each care center this week and the day they visited as well as the time they visited and the care center name they visited.

View Query:

```
SELECT Pa_Person_ID, First_Name, Last_Name, Patient_Type, Visit_Number, Visit_Date,  
Visit_Time, Care_Center_ID, Care_Center_Name  
FROM Outpatients_Visited  
WHERE Visit_Date BETWEEN '2020-04-13' AND '2020-04-19'  
ORDER BY Care_Center_Name;
```

Results:

	Pa_Person_ID	First_Name	Last_Name	Patient_Type	Visit_Number	Visit_Date	Visit_Time	Care_Center_ID	Care_Center_Name
1	1118	Lisa	Lainez	0	1118	2020-04-19	01:30:00	1113	Dermatology Center
2	1129	Álvaro	Morata	0	1113	2020-04-15	09:00:00	1113	Dermatology Center
3	1126	Victor	Wanyama	0	1111	2020-04-13	07:00:00	1111	Emergency Room Center
4	1146	Iker	Casillas	0	1116	2020-04-18	12:00:00	1111	Emergency Room Center
5	1127	Thomas	Lemar	0	1112	2020-04-14	08:00:00	1112	Outpatient Care Center
6	1144	David	Neres	0	1115	2020-04-17	11:00:00	1115	Surgical Unit Care Center
7	1142	Funes	Mori	0	1114	2020-04-16	10:00:00	1114	Urology Center

4.

Description: Creating a business report query for the fourth view of the report query.

What The Report Does: The report objective is to see how many employees there are who worked between 40 and 50 hours this week and what type of employees are working in the care centers and what care centers were they assigned to.

View Query:

```

SELECT Em_Person_ID, First_Name, Last_Name, Employee_Type, Hours_Worked,
Care_Center_ID, Care_Center_Name
FROM Employees_Working
WHERE Hours_Worked BETWEEN 40 AND 50
ORDER BY Hours_Worked;
```

Results:

	Em_Person_ID	First_Name	Last_Name	Employee_Type	Hours_Worked	Care_Center_ID	Care_Center_Name
1	1116	Erika	Gonzalez	RN Nurse	40	1111	Emergency Room Center
2	1117	Stephany	Sánchez	LPN Nurse	45	1112	Outpatient Care Center
3	1118	Lisa	Lainez	Technician	50	1113	Dermatology Center

5.

Description: Creating a business report query for the fifth view of the report query.

What The Report Does: The report objective is to see the resident patients who were discharged this week and go check the equipment to see any issues in those rooms.

View Query:

```
SELECT Pa_Person_ID, First_Name, Last_Name, Patient_Type, Date_Discharged,  
Care_Center_ID, Care_Center_Name, Room_Number  
FROM Resident_Patient_Discharged  
WHERE Date_Discharged BETWEEN '2020-04-06' AND '2020-04-12';
```

Results:

	Pa_Person_ID	First_Name	Last_Name	Patient_Type	Date_Discharged	Care_Center_ID	Care_Center_Name	Room_Number
1	1117	Stephany	Sánchez	R	2020-04-09	1115	Surgical Unit Care Center	1117
2	1128	Cindy	Correa	R	2020-04-06	1111	Emergency Room Center	1111
3	1141	Lisa	Pena	R	2020-04-07	1111	Emergency Room Center	1113
4	1145	Diego	Carlos	R	2020-04-08	1115	Surgical Unit Care Center	1115
5	1154	Thomas	Partey	R	2020-04-10	1115	Surgical Unit Care Center	1119

6.

Description: Creating a business report query for the sixth view of the report query.

What The Report Does: The report objective is to see how many volunteers there are currently working who have between 5 and 6 hours worked this week and their work unit description. In addition, what type of volunteers are working in the care centers and where they are assigned to.

View Query:

```
SELECT Vo_Person_ID, First_Name, Last_Name, Work_Unit_ID, Work_Unit_Description,  
Hours_Worked, Care_Center_ID, Care_Center_Name  
FROM Volunteers_Working  
WHERE Hours_Worked BETWEEN 5 AND 6;
```

Results:

	Vo_Person_ID	First_Name	Last_Name	Work_Unit_ID	Work_Unit_Description	Hours_Worked	Care_Center_ID	Care_Center_Name
1	1141	Lisa	Pena	1	Emergency Room	5	1111	Emergency Room Center
2	1142	Funes	Mori	2	Outpatient Care	5	1112	Outpatient Care Center
3	1143	Carlos	Pineda	3	Dermatology	6	1113	Dermatology Center
4	1144	David	Neres	4	Urology	6	1114	Urology Center

7.

Description: Creating a business report for the seventh view of the report query.

What The Report Does: The report objective is to see how many appointments there are going to be in the upcoming week for dermatology and the date and time of the appointments.

View Query:

```
SELECT Pa_Person_ID, Ph_Person_ID, Appointment_Date, Appointment_Time,  
Care_Center_ID, Care_Center_Name  
FROM Appointments  
WHERE Appointment_Date BETWEEN '2020-04-20' AND '2020-04-26'  
AND Care_Center_ID = 1113;
```

Results:

	Pa_Person_ID	Ph_Person_ID	Appointment_Date	Appointment_Time	Care_Center_ID	Care_Center_Name
1	1128	1113	2020-04-20	11:00:00	1113	Dermatology Center
2	1143	1113	2020-04-24	02:00:00	1113	Dermatology Center

8.

Description: Creating a business report query for the eighth view of the report query.

What The Report Does: The report objective is to see how many resident patients can be admitted next week based on how many beds there are, the availability, and how many of each types of beds there are available as well as where they are located.

View Query:

```
SELECT Bed_Number, Room_Number, Is_Available, Bed_Type_Description, Care_Center_ID,  
Care_Center_Name  
FROM Beds_Available  
WHERE Is_Available = 'Y'  
ORDER BY Care_Center_Name;
```

Results:

	Bed_Number	Room_Number	Is_Available	Bed_Type_Description	Care_Center_ID	Care_Center_Name
1	1113	1113	Y	Clinitron Air Bed	1111	Emergency Room Center
2	1116	1116	Y	Standard Manuel Bed	1111	Emergency Room Center
3	1118	1118	Y	Clinitron Air Bed	1111	Emergency Room Center
4	1119	1119	Y	Stretcher	1115	Surgical Unit Care Center
5	1120	1120	Y	Low Bed	1115	Surgical Unit Care Center
6	1112	1112	Y	Electric Automatic Bed	1115	Surgical Unit Care Center
7	1117	1117	Y	Electric Automatic Bed	1115	Surgical Unit Care Center
8	1114	1114	Y	Stretcher	1115	Surgical Unit Care Center
9	1115	1115	Y	Low Bed	1115	Surgical Unit Care Center

9.

Description: Creating a business report query for the ninth view of the report query.

What The Report Does: The report objective is to see how many patients are going to have surgery next week and the date and time of the appointments.

View Query:

```
SELECT Pa_Person_ID, Ph_Person_ID, Appointment_Date, Appointment_Time,  
Care_Center_ID, Care_Center_Name  
FROM Appointments  
WHERE Appointment_Date BETWEEN '2020-04-20' AND '2020-04-26'  
AND Care_Center_ID = 1115  
ORDER BY Appointment_Date;
```

Results:

	Pa_Person_ID	Ph_Person_ID	Appointment_Date	Appointment_Time	Care_Center_ID	Care_Center_Name
1	1145	1115	2020-04-21	02:30:00	1115	Surgical Unit Care Center
2	1130	1115	2020-04-23	01:30:00	1115	Surgical Unit Care Center

10.

Description: Creating a business report query for the tenth view of the report query.

What the Report Does: The report objective is to see how many appointments there are going to be in the upcoming week for Urology and the date and time of the appointments.

View Query:

```
SELECT Pa_Person_ID, Ph_Person_ID, Appointment_Date, Appointment_Time,  
Care_Center_ID, Care_Center_Name  
FROM Appointments  
WHERE Appointment_Date BETWEEN '2020-04-21' AND '2020-04-27'  
AND Care_Center_ID = 1114;
```

Results:

	Pa_Person_ID	Ph_Person_ID	Appointment_Date	Appointment_Time	Care_Center_ID	Care_Center_Name
1	1129	1114	2020-04-25	12:00:00	1114	Urology Center
2	1144	1114	2020-04-26	03:00:00	1114	Urology Center

## **Conclusion**

In this project to enhance performance & security in the Hospital Management System Application named EZMedicalGroup.com some features needed to be added. The application was upgraded by adding some features like creating indexes for the reports such as secondary key indexes to improve search performance. Also analyze, design, and create Views for the queries in the reports as well as queries to test the views. With the creation of indexes and views there was an improvement in performance and security.

