CSCI 340 01W Final Project

Replace this topic line with your designed Database Name

**Abiola Joda**

**Due Date: 5/8/2024 Wed. by 11:59pm**

Please specify which program you are in (**CS** or **CIS**)? Your answer: CS

Step 1: Create an imaginary scenario. Your scenario should satisfy the following conditions, explain the story behind the scenarios, and all your assumptions, which are required to support relationships given below.

1. Describe at least one one-to-one binary relationships:
2. Describe at least one one-to-many binary relationships:
3. Describe at least one many-to-many binary relationships:
4. Describe at least one intersection data (on many-to-many relationships):

Answer:

Modern Healthcare Hospital is a state-of-the-art medical facility that offers a wide range of specialized services to its patients. The hospital employs skilled medical professionals who work together to provide excellent healthcare services.

One-to-One Binary Relationship:

Each patient is assigned a primary doctor who oversees their overall medical care. This one-to-one relationship ensures personalized and continuous healthcare management for each patient.

One-to-Many Binary Relationship:

Within the hospital, there are various departments specializing in different medical fields such as Cardiology, Oncology, Radiology, Neurology, and Pediatrics. Each department has multiple doctors who specialize in different areas within that department. For example, the Cardiology department has doctors specializing in interventional cardiology, electrophysiology, and cardiac surgery.

Many-to-Many Binary Relationship:

Modern Healthcare Hospital offers a range of medical procedures and treatments to its patients. Patients may undergo multiple medical procedures during their treatment journey, and each procedure may involve multiple doctors and nurses. For instance, a patient undergoing heart surgery may require the expertise of a cardiac surgeon, an anesthesiologist, and several assisting nurses.

Intersection Data:

The intersection of patients, medical procedures, doctors, and nurses occurs during surgical procedures. The "SurgicalProcedures" table tracks details such as the patient undergoing the procedure, the medical procedure performed, the operating doctor, assisting nurses, and the date of the procedure. This intersection data ensures accurate documentation and coordination during surgical interventions.

Assumptions:

Each patient admitted to Modern Healthcare Hospital is assigned a unique PatientID.

Doctors and nurses are assigned unique DoctorID and NurseID, respectively.

Medical procedures are categorized and assigned unique ProcedureID.

Appointments are scheduled for patients to meet their primary doctors or specialists.

Patients can undergo multiple medical procedures during their treatment.

The hospital follows strict protocols for maintaining patient records and ensuring the continuity of care.

In this scenario, Modern Healthcare Hospital strives to deliver exceptional medical care by establishing clear relationships between patients, medical procedures, doctors, nurses, and departments, ensuring efficiency and effectiveness in healthcare delivery.

DROP TABLE IF EXISTS:

This command checks if a table exists in the database and drops it if it exists, preventing errors if the table already exists before creating it again.

CREATE TABLE:

This command creates a new table in the database with the specified columns and data types. It defines the structure of the table, including primary keys, foreign keys, and constraints.

INSERT INTO:

This command inserts new records into a table. It specifies the table name and columns to insert data into, followed by the values to be inserted into those columns.

DELETE FROM:

This command deletes records from a table based on specified conditions. It removes rows from the table that match the conditions specified in the WHERE clause.

UPDATE:

This command updates existing records in a table based on specified conditions. It modifies the values of specified columns in rows that match the conditions in the WHERE clause.

SELECT:

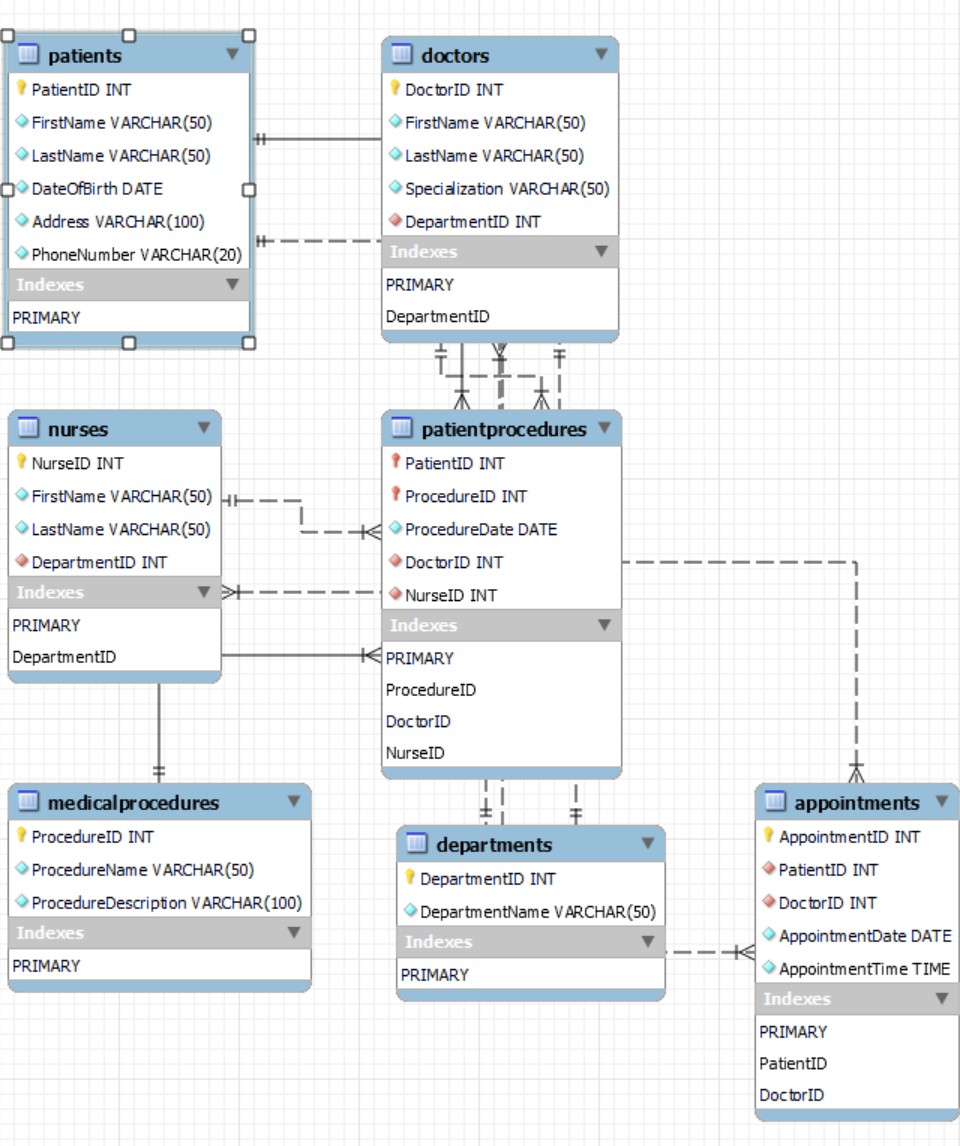
This command retrieves data from one or more tables in the database. It can be used to fetch specific columns, apply filtering conditions with the WHERE clause, group data with the GROUP BY clause, sort data with the ORDER BY clause, and filter grouped data with the HAVING clause.

JOIN:

This command is used to combine rows from two or more tables based on a related column between them. It allows you to fetch data from multiple tables simultaneously by specifying how the tables are related (e.g., INNER JOIN, LEFT JOIN, RIGHT JOIN).

CREATE VIEW:

This command creates a virtual table based on the result of a SELECT query. It allows you to save complex queries as a view, making it easier to access and reuse the query results without rewriting the entire query each time.

Step 2:

Step 3:

Answer:

Patients and Appointments:

When a patient is deleted from the system (DELETE CASCADE), all their related appointments should also be deleted automatically. This ensures that there are no orphaned appointments without a corresponding patient.

Doctors and Appointments:

If a doctor is deleted from the system, their appointments should not be automatically deleted. Instead, the appointments should be updated to reflect that the doctor is no longer available (UPDATE CASCADE). This maintains the historical record of appointments even if a doctor leaves the hospital.

Departments and Doctors/Nurses:

When a department is deleted (DELETE CASCADE), all doctors and nurses associated with that department should also be deleted automatically. This ensures that there are no doctors or nurses without a department affiliation.

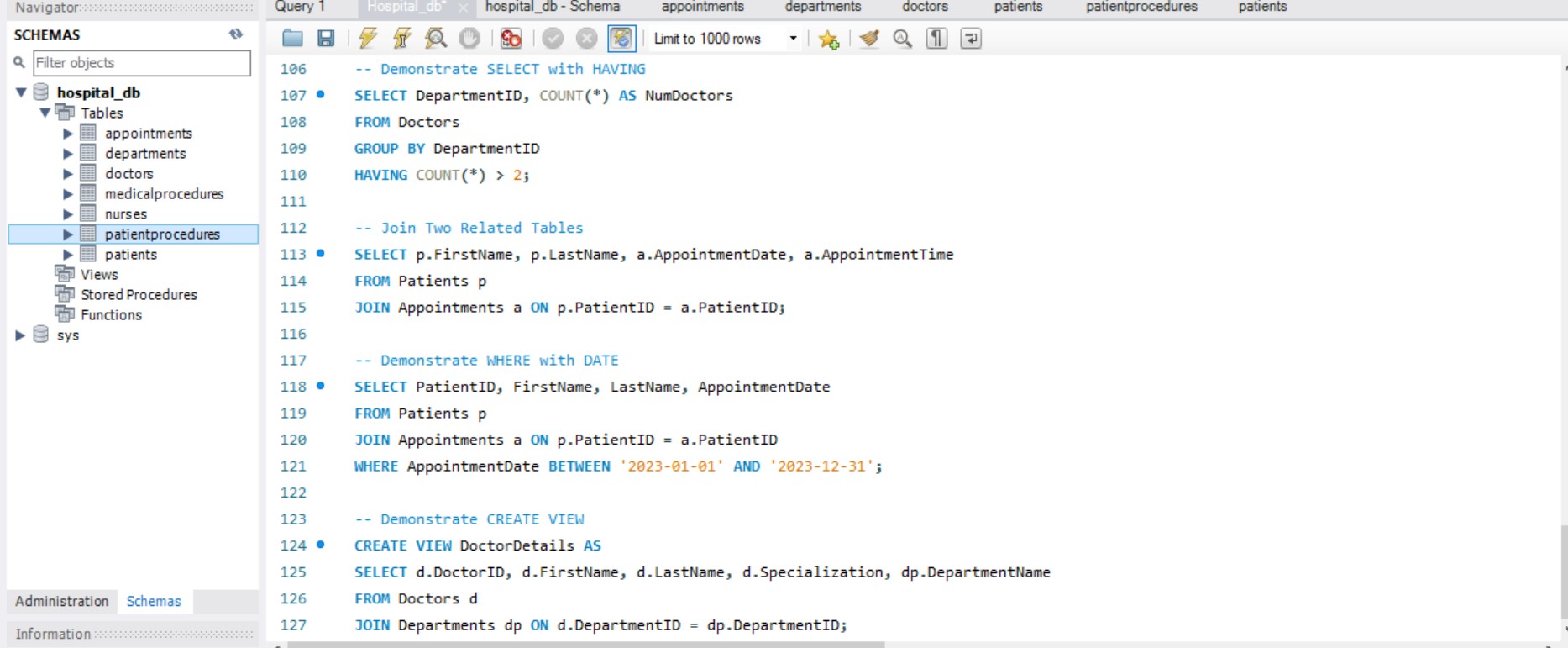
MedicalProcedures and PatientProcedures:

If a medical procedure is deleted from the system, any related patient procedures should also be deleted automatically (DELETE CASCADE). This prevents patient procedures from referencing non-existent medical procedures.

These referential integrity rules and constraints help maintain data consistency and prevent orphaned or invalid data in the database. They ensure that the relationships between entities are enforced and maintained correctly.

Step 4:

Answer:



Step 5:

Answer:

Patients Table:

Attributes: PatientID (PK), FirstName, LastName, DateOfBirth, Address, PhoneNumber

This table is in 1NF, 2NF, and 3NF as each attribute is atomic, there are no repeating groups, and all non-key attributes are functionally dependent on the primary key.

Departments Table:

Attributes: DepartmentID (PK), DepartmentName

This table is in 1NF, 2NF, and 3NF as it contains no repeating groups, and all non-key attributes are functionally dependent on the primary key.

Doctors Table:

Attributes: DoctorID (PK), FirstName, LastName, Specialization, DepartmentID (FK)

This table is in 1NF, 2NF, and 3NF as it contains no repeating groups, and all non-key attributes are functionally dependent on the primary key and do not have transitive dependencies.

Nurses Table:

Attributes: NurseID (PK), FirstName, LastName, DepartmentID (FK)

This table is in 1NF, 2NF, and 3NF as it contains no repeating groups, and all non-key attributes are functionally dependent on the primary key and do not have transitive dependencies.

Appointments Table:

Attributes: AppointmentID (PK), PatientID (FK), DoctorID (FK), AppointmentDate, AppointmentTime

This table is in 1NF, 2NF, and 3NF as it contains no repeating groups, and all non-key attributes are functionally dependent on the primary key.

MedicalProcedures Table:

Attributes: ProcedureID (PK), ProcedureName, ProcedureDescription

This table is in 1NF, 2NF, and 3NF as it contains no repeating groups, and all non-key attributes are functionally dependent on the primary key.

PatientProcedures Table:

Attributes: PatientID (FK), ProcedureID (FK), ProcedureDate, DoctorID (FK), NurseID (FK)

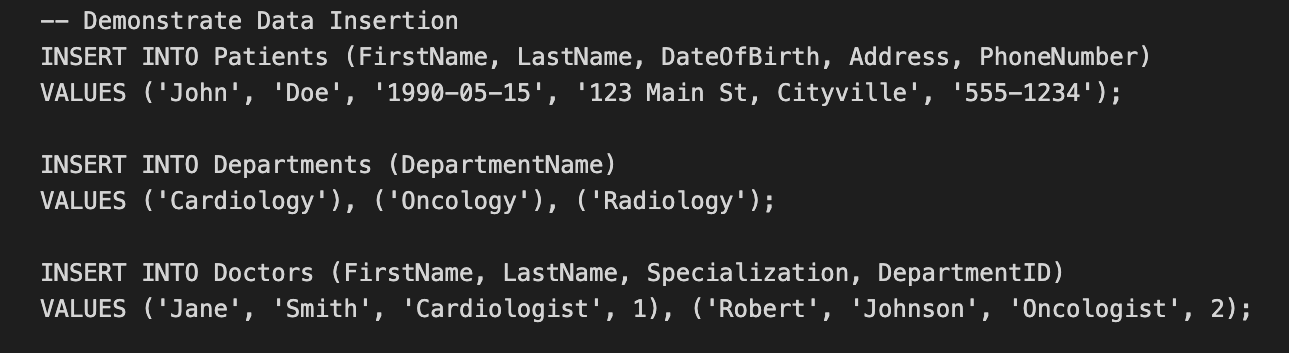
This table is in 1NF, 2NF, and 3NF as it contains no repeating groups, and all non-key attributes are functionally dependent on the primary key.

Step 6:

Answer:

Step 7:

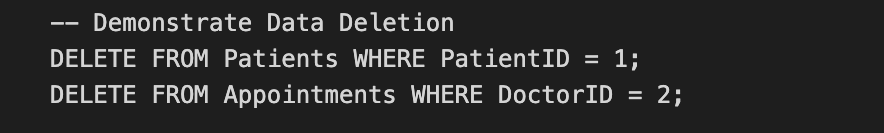
Answer:

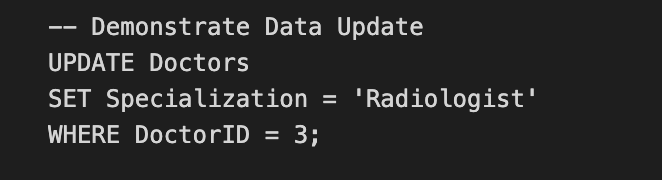


In summary, these SQL commands demonstrate how to insert data into specific tables, providing values for the corresponding columns to create new records or entries in the database.

Step 8:

Answer:

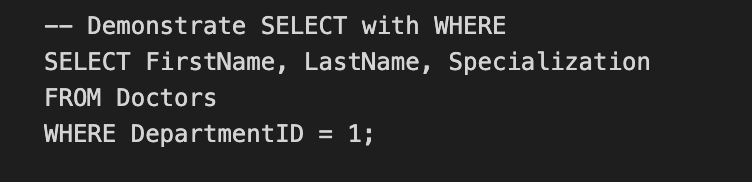
In summary, these SQL commands demonstrate how to delete specific records or multiple records based on a condition from the corresponding tables in the database.  
Step 9:



This SQL command updates the Specialization of the doctor with ID 3 to 'Radiologist' in the Doctors table.

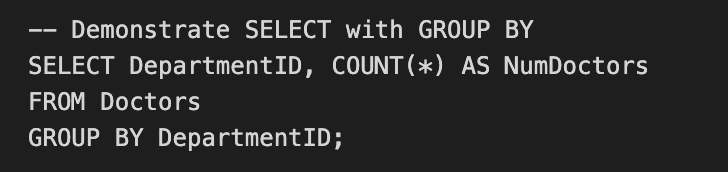
Step 10:

Answer:

  
This SQL command selects the FirstName, LastName, and Specialization columns from the Doctors table where the DepartmentID is equal to 1.

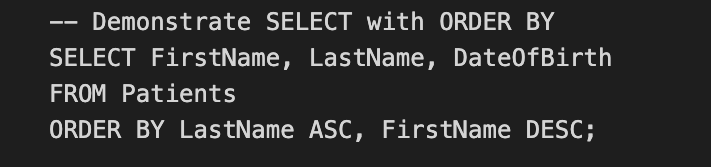
Step 11:

Answer:

  
This SQL command calculates the count of doctors (NumDoctors) for each DepartmentID by grouping the data in the Doctors table based on DepartmentID.

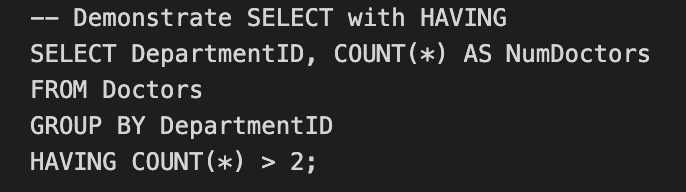
Step 12:

Answer:

  
This SQL command retrieves the FirstName, LastName, and DateOfBirth columns from the Patients table and orders the results first by LastName in ascending order and then by FirstName in descending order.

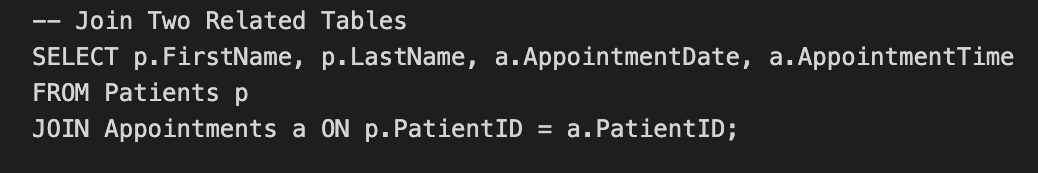
Step 13:

Answer:

  
This SQL command calculates the count of doctors (NumDoctors) for each DepartmentID from the Doctors table, groups the results by DepartmentID, and then filters the groups to include only those with a count greater than 2 using the HAVING clause.

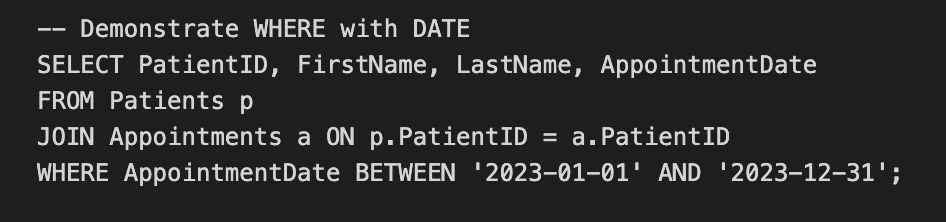
Step 14:

Answer:

  
This SQL command joins two related tables, Patients and Appointments, and selects the FirstName, LastName, AppointmentDate, and AppointmentTime columns from the Patients and Appointments tables based on the common PatientID column.

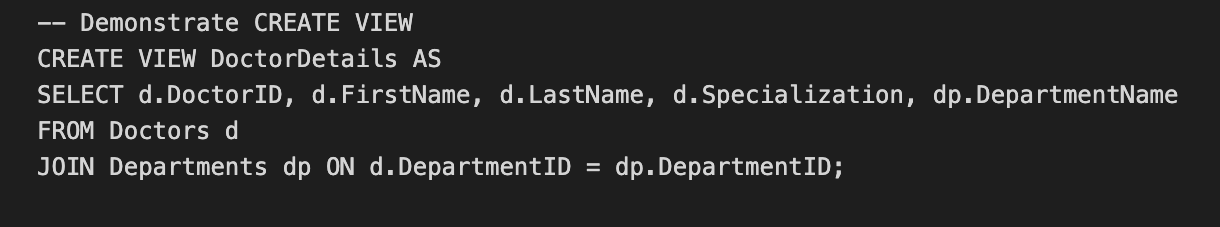
Step 15:

Answer:

  
This SQL command retrieves the PatientID, FirstName, LastName, and AppointmentDate columns from the Patients table joined with the Appointments table, filtering the results to include only appointments that fall between January 1, 2023, and December 31, 2023, using the WHERE clause with the BETWEEN operator.

Step 16:

Answer:



This SQL command creates a view named DoctorDetails, which is a virtual table that contains the DoctorID, FirstName, LastName, Specialization, and DepartmentName columns. The data in this view is obtained by joining the Doctors table (aliased as "d") with the Departments table (aliased as "dp") based on the DepartmentID column.