Logo

Description automatically generated

**Project Report**

**DATABASE MANAGEMENT SYSTEM**

**ONLINE DOCTOR APPOINTMENT SYSTEM**

**Department of Information Engineering Technology  
National Skills University Islamabad**

**Group Members**

**Ammar Aziz F21IETBS005**

**Ahtisham Qasim F21IETBS012**

**M. Ali Sikandar F21IETBS041**

**Ameer Ullah F21IETBS031**

**Supervised By**

**Dr. Bilal Shah Nawaz**

**Table of Contents**

[**1. Problem Statement** 3](#_Toc155817990)

[**2. Planning** 3](#_Toc155817991)

[**2.1 Business rules** 3](#_Toc155817992)

[**3. Analysis** 3](#_Toc155817993)

[**3.1 Fig: Entity Relation Diagram (ERD)** 4](#_Toc155817994)

[**4. Logical Design** 4](#_Toc155817995)

[**4.1 Fig: Logical Data Model** 5](#_Toc155817996)

[**5. Physical Design** 6](#_Toc155817997)

[**5.1 Fig: Physical Data Model** 6](#_Toc155817998)

[**6. Implementation** 7](#_Toc155817999)

[**7. Maintenance** 12](#_Toc155818000)

[**8. Conclusion** 13](#_Toc155818001)

# **1. Problem Statement**

Getting timely healthcare services can be tough because booking appointments the old way is a hassle. Patients struggle to book slots with their preferred doctors, causing long waits, inconvenience, and sometimes health risks. On the other side, doctors find it hard to manage their schedules well, leading to poor use of resources and unhappy patients. We need a better way to book appointments that’s easy for everyone and stops these problems.

Now we follow the SDLC model make the system online.

# **2. Planning**

An online doctor appointment system is a digital platform that facilitates the scheduling and management of appointments between patients and doctors. Our system will be used for managing appointments, patients, doctors, and payments across hospitals. While we’ll just create a small-scale example, a full hospital management system might be used for a nationwide health care service.

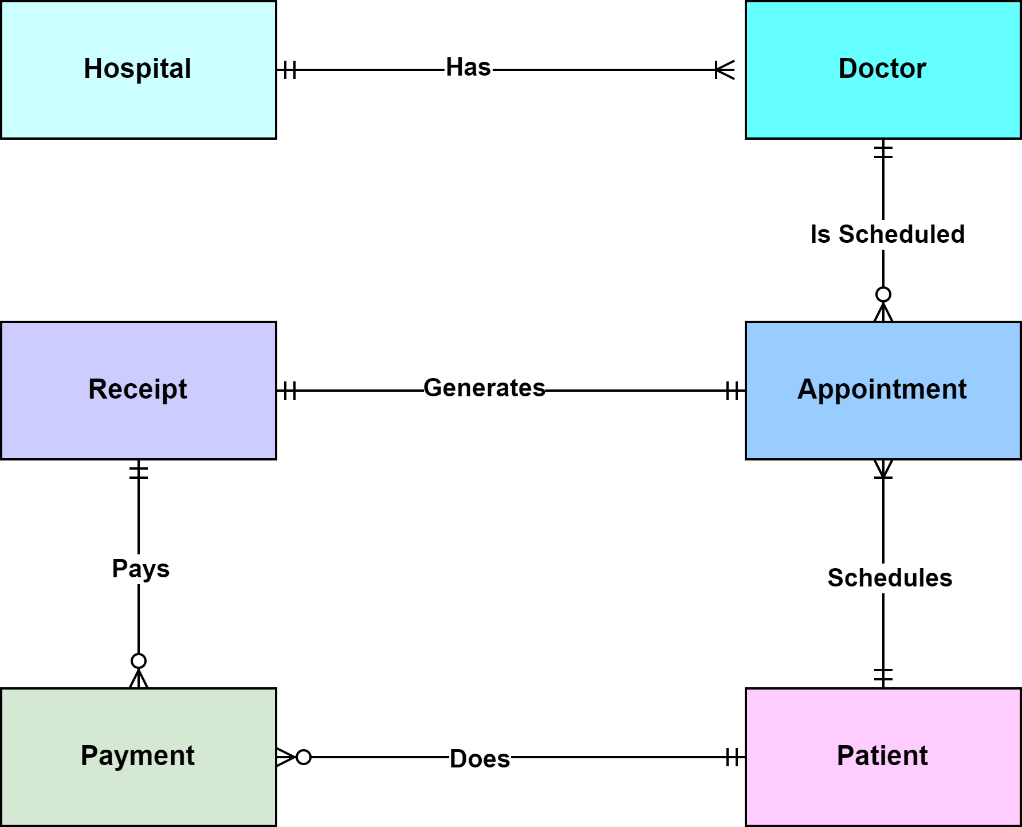
For making an online system there are several business rules we follow.

# **2.1 Business rules**

1. Hospital-Doctor: A hospital must have one or more doctors, but a doctor can only work for one hospital.
2. Doctor-Appointment: A doctor can be scheduled for many appointments, but an appointment can only be scheduled for one doctor.
3. Patient-Appointment: A patient can schedule many appointments, but an appointment can only be scheduled by one patient.
4. Patient-Payment: A patient can make many payments, but a payment can only be made by one patient.
5. Appointment-Receipt: An appointment must generate one receipt, but a receipt can only be generated by one appointment.
6. Payment-Receipt: A payment must have one receipt but a receipt may have one or more payments.

# **3. Analysis**

In this phase we convert the requirements into a conceptual model. This conceptual model is called Entity Relation Diagram (ERD). ERDs help everyone to understand the foundations of the data/information that is going to be stored within a database. Detailed conceptual data model, including all entities, relationships, and business rules. Going to be transformed and then implemented.

****

# **3.1 Fig: Entity Relation Diagram (ERD)**

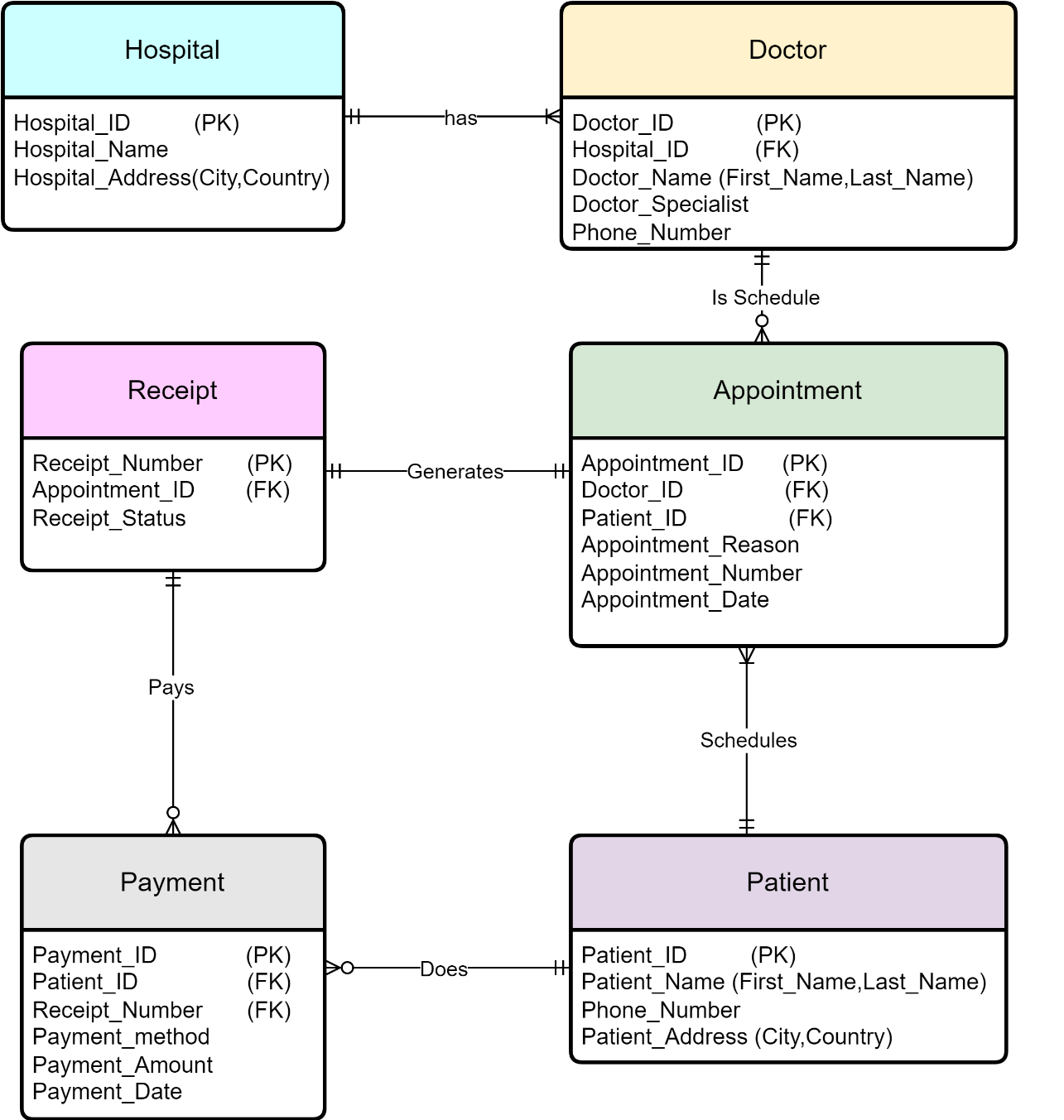
# **4. Logical Design**

The logical design phase focuses on creating a blueprint of how the system will work without concerning itself with the technical details or specific technologies. It involves outlining the system's functionality, data flow, and interactions without diving into coding or implementation specifics. It is Obtained by translating the conceptual database design into another data model.

In this we will define the Foreign key (FK) and Primary key (PK).

**Primary Key (PK):** This is a unique identifier for each record in a table. It ensures that each row in a table is distinct and identifiable. For example, in a table of students, the student ID might be the primary key.

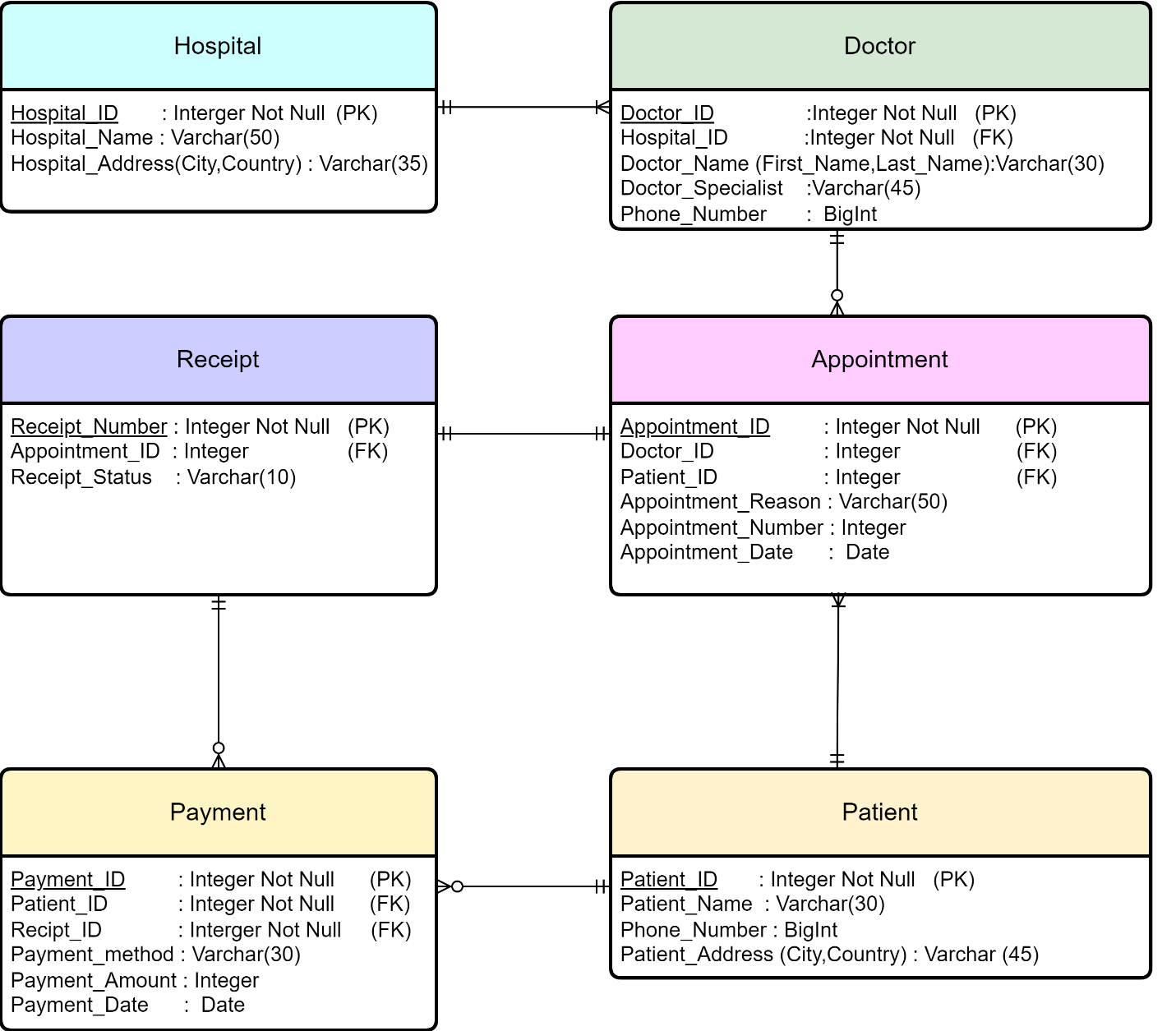
**Foreign Key (FK):** This is a field in a table that links to the primary key in another table. It establishes a relationship between two tables. For instance, in a database where students are associated with courses, the student ID in the courses table would be a foreign key referencing the student ID in the students table. This relationship allows for data consistency and integrity across related tables.

****

# **4.1 Fig: Logical Data Model**

# **5. Physical Design**

A physical data model is a detailed representation of how data is stored in a database management system. It defines the structure of the database, including tables, columns, data types, relationships between tables, indexes, constraints, and more. This model is specific to a particular database management system (DBMS) and considers implementation details like storage, performance, and optimization.

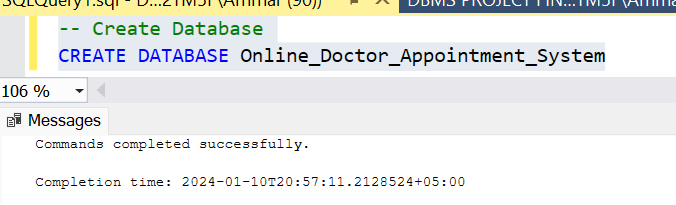


# **5.1 Fig: Physical Data Model**

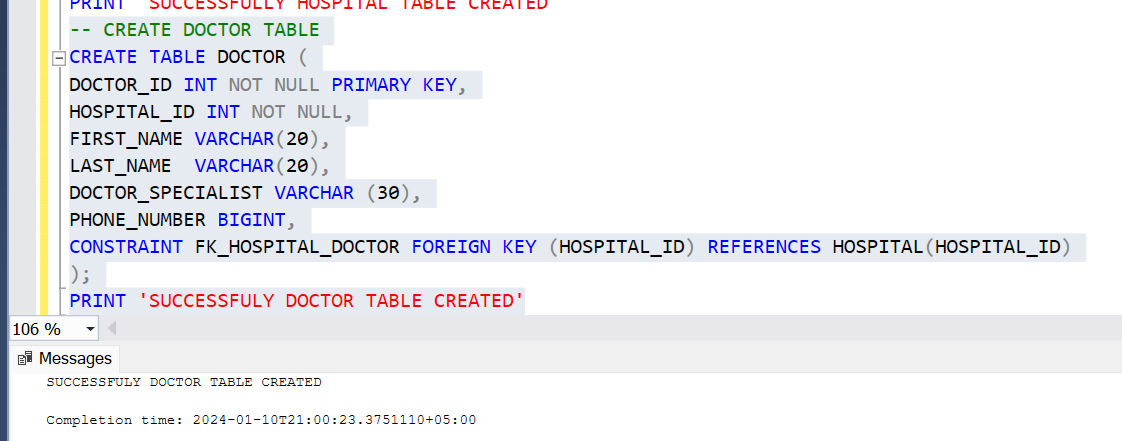
# **6. Implementation**

In Implementation we use Physical Data Model. Convert the Physical Data Model into actual database model using SQL.

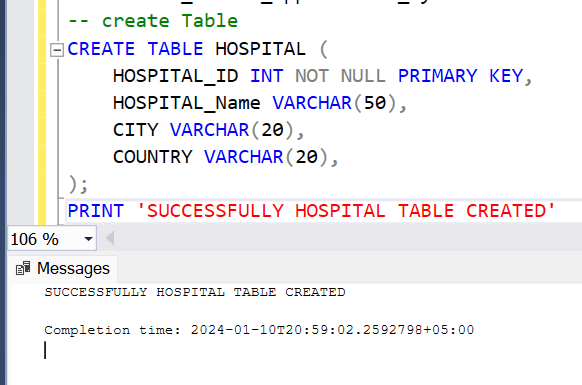
First, we create a new database in SQL server with name Online\_Doctor\_Appointment\_System1 and use it for creating tables in database.



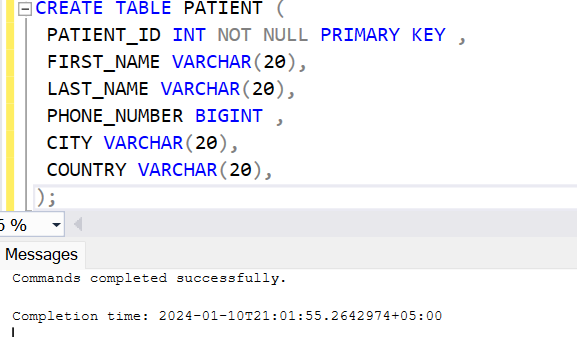
First, create a table with name Doctor with Primary key DOCTOR\_ID and Foreign Key HSOPITAL\_ID.



Create a table with name HOSPITAL with Primary key HOSTEL\_ID.



Create a table PATIENT with Primary key PATIENT\_ID.

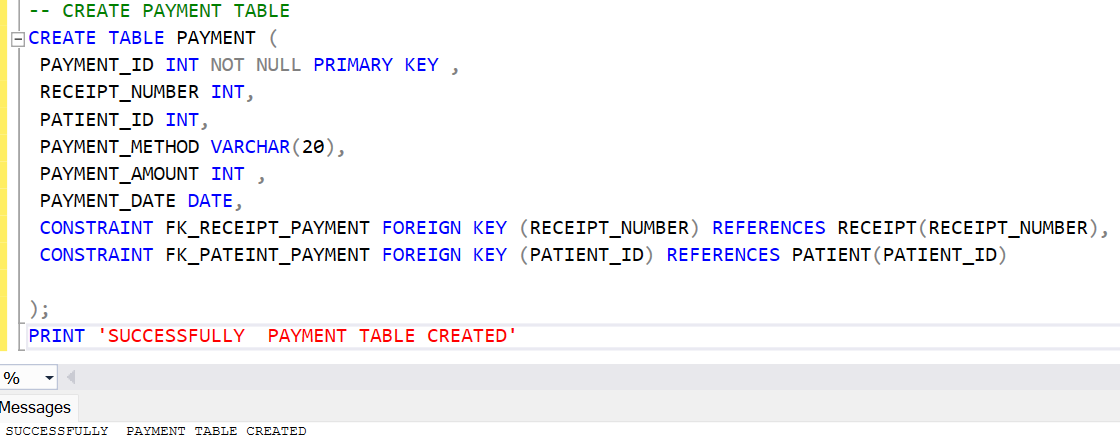


Create a table APPOINTMENT with Primary key APPOINTMENT\_ID and with Foreign Keys DOCTOR\_ID AND PATIENT\_ID.

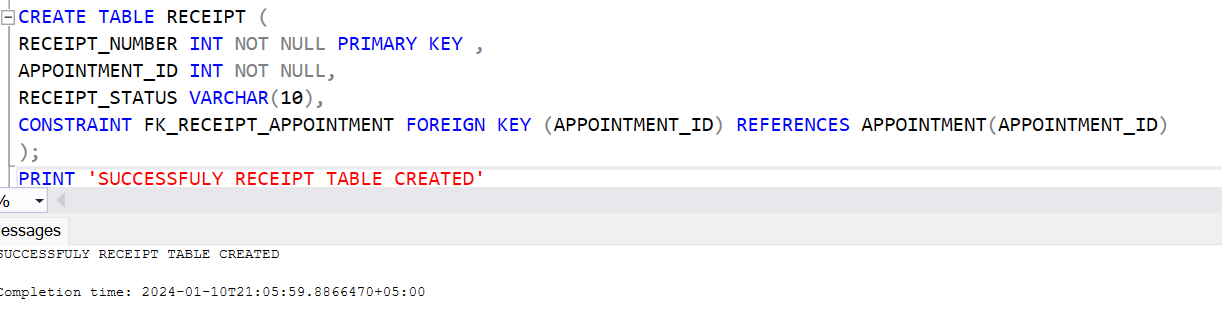
A screenshot of a computer

Description automatically generated

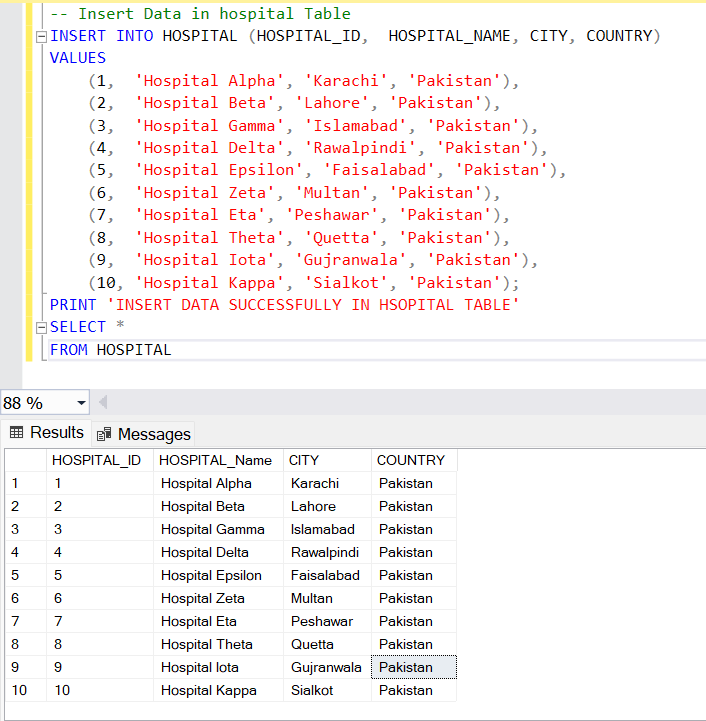
Create a table with the name PAYMENT with Primary key PAYMENT\_ID and Foreign Key RECEIPT\_ID and PATIENT\_ID.



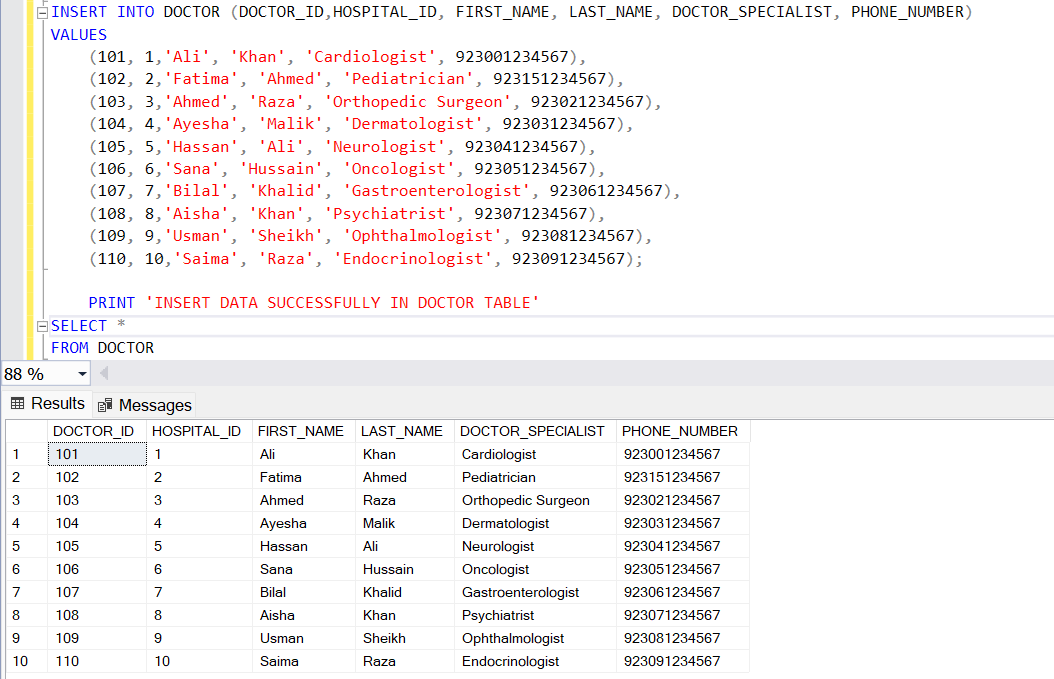
Create a table with name RECIPET with primary key PECIPET\_ID and Foreign Key APPOINTMENT\_ID.



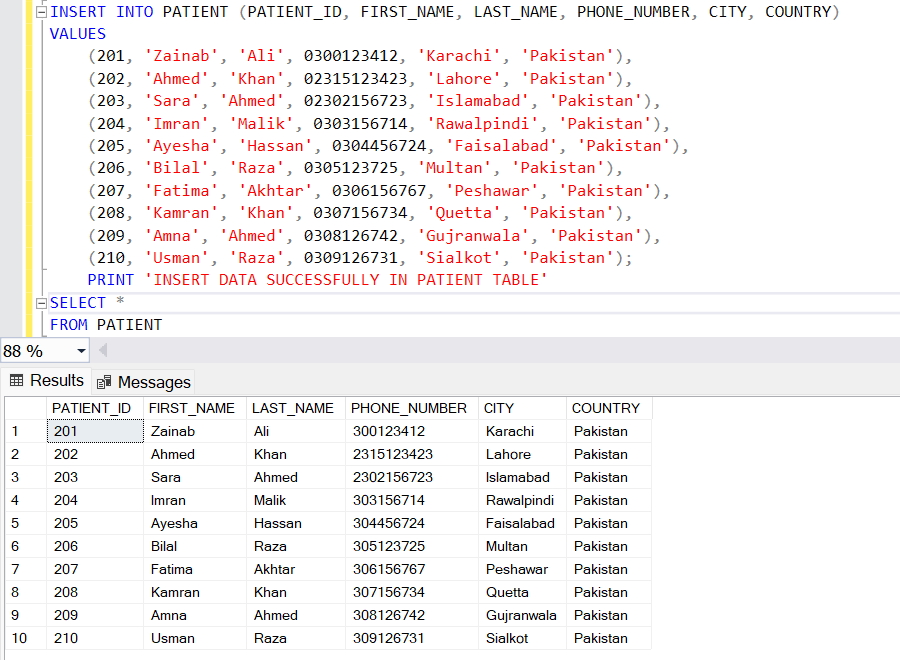
Insert and display data of HOSPITAL Table.



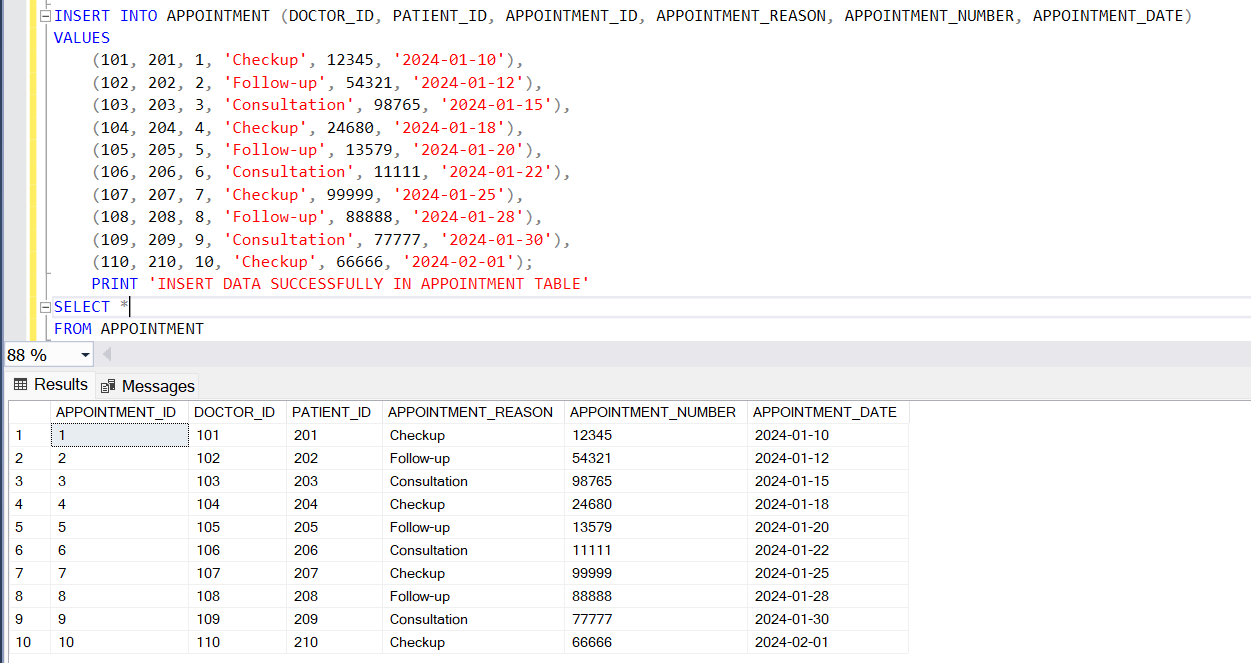
Insert and display data of DOCTOR Table.



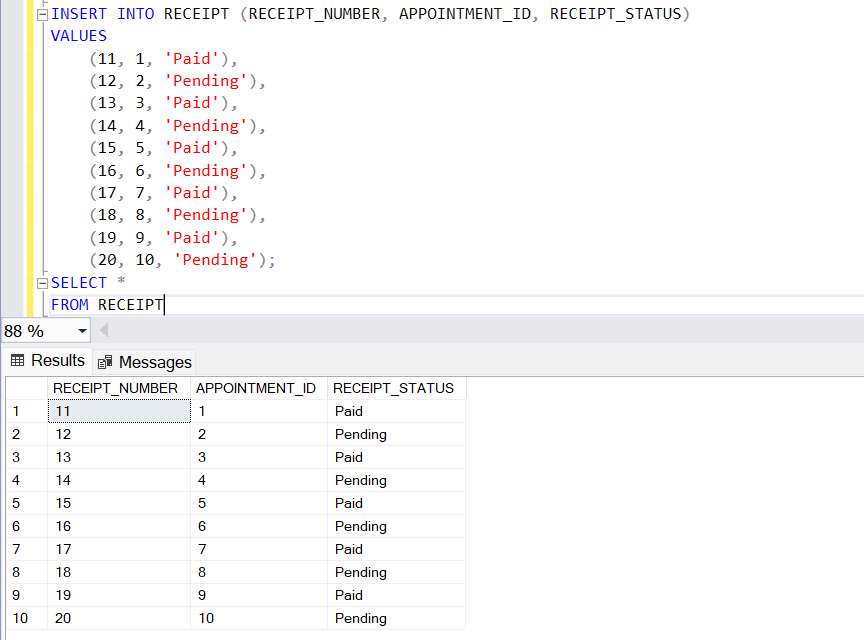
Insert and display data of PATIENT Table.



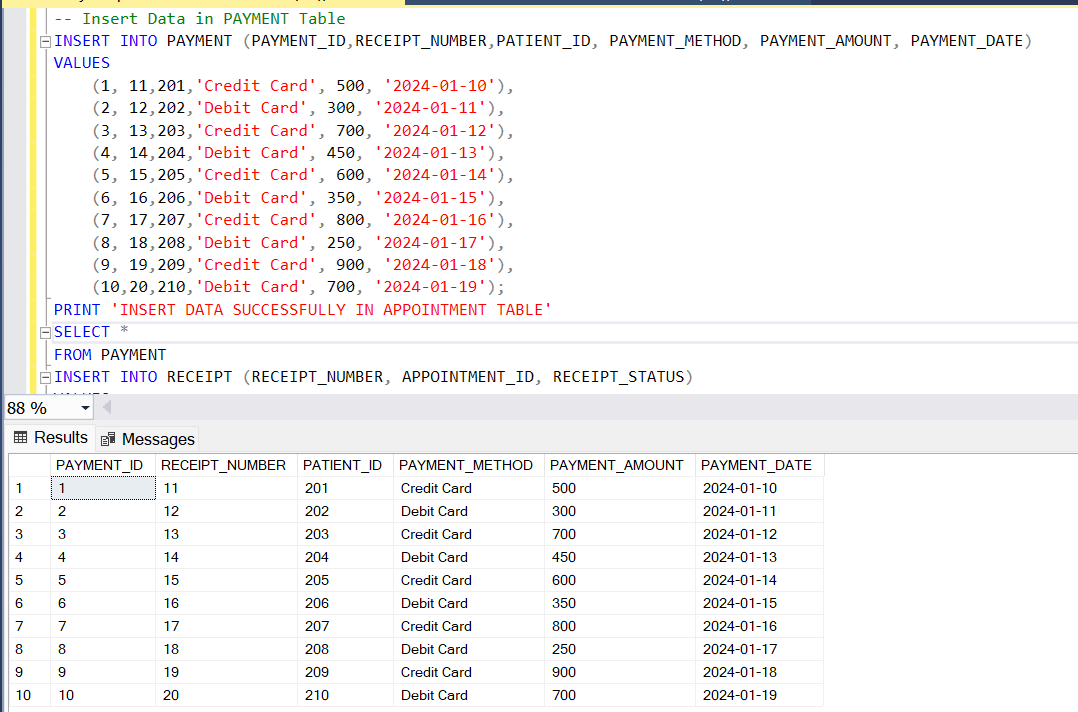
Insert and display data of APPOINTMENT Table.



Insert and display data of RECEIPT Table.



Insert and display data of PAYMENT Table.

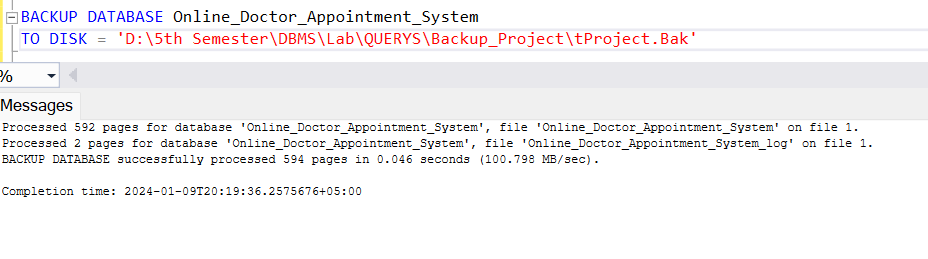


After successfully inserted data into Table we can say that in this way we make a database for a system use for online doctor appointment system.

# **7. Maintenance**

For system maintenance we must first follow the backup. Because, backing up data in an online doctor appointment system is crucial for preserving patient information and system functionality.

**7.1 Backup**

****

**7.2 Database Monitoring:** Monitor database performance, including query response times, database errors, and storage usage. Regularly optimize database performance to ensure efficient data retrieval and storage.

# **8. Conclusion**

An online doctor appointment system offers a multitude of benefits for both patients and healthcare providers. It streamlines the process of booking appointments, enhances accessibility to healthcare services, reduces waiting times, and allows for better organization of medical records. Additionally, it enables remote consultations, making healthcare more convenient for individuals who might have difficulty visiting a physical clinic due to distance, mobility issues, or time constraints. An online doctor appointment system represents a pivotal innovation in healthcare accessibility and efficiency. By leveraging digital platforms, it addresses the challenges of traditional appointment scheduling, offering patients convenient access to healthcare services while streamlining workflows for healthcare providers.

**--The End--**