**Batch: E-2** **Roll No.: 16010123325**

**Experiment / assignment / tutorial No. 3**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

**Title: :** Implementation of Database in SQL -DDL

---

**Objective:** Define/modify database definitions with proper constraints

---

**Expected Outcome of Experiment:**

CO 2: Convert entity-relationship diagrams into relational tables, populate a relational database and formulate SQL queries on the data Use SQL for creation and query the database.

CO 3: Define and apply integrity constraints and improve database design using normalization techniques.

---

**Pre Lab/ Prior Concepts**:

Resources used: Postgresql

**Theory:** The set of relations in a database must be specifies to the system by means of a data definition language (DDL). The SQL DDL allows specification of not only a set of relations but also specific information about the relation including,

1. The schema for each relation
2. The domain of values associated with each attribute
3. The integrity constraints
4. The set of indices to be maintained for each relation
5. The security and authorization information for each relation
6. The physical storage structure of each relation on disk

**Syntax Create Table:**

create table employee(ssn,fname varchar(10), mname varchar(10), lname varchar(10), desg

varchar(20), gender varchar(5), addr varchar(20), bdate datetime, sal float,primary key(ssn));

create table manages(ssn int, dept_code int, start_dt datetime, foreign key(ssn) references employee, foreign key(dept_code) references department, key(ssn,dept_code) ) on delete set null

**Data Constraints**

Business managers of the organization determine a set of rules that must be applied before the data is stored in the database. The application of such rules on raw data ensures **data integrity**.

**Eg:-** An employee belonging to the Sales department cannot have a salary higher than Rs. 1000.

An employee has an unique identification number.

**Applying Data Constraints**

Oracle permits data constraints to be attached to table columns using SQL syntax. Constraints can be attached to table columns using Alter table.

**Unique Constraint**

**Unique Constraint- At column level Syntax**

**<ColumnName><Datatype>(<size>) UNIQUE**

**Unique Constraint- At table level**

**CREATE TABLE<TableName>(**

**<ColumnName><Datatype>(<size>)**

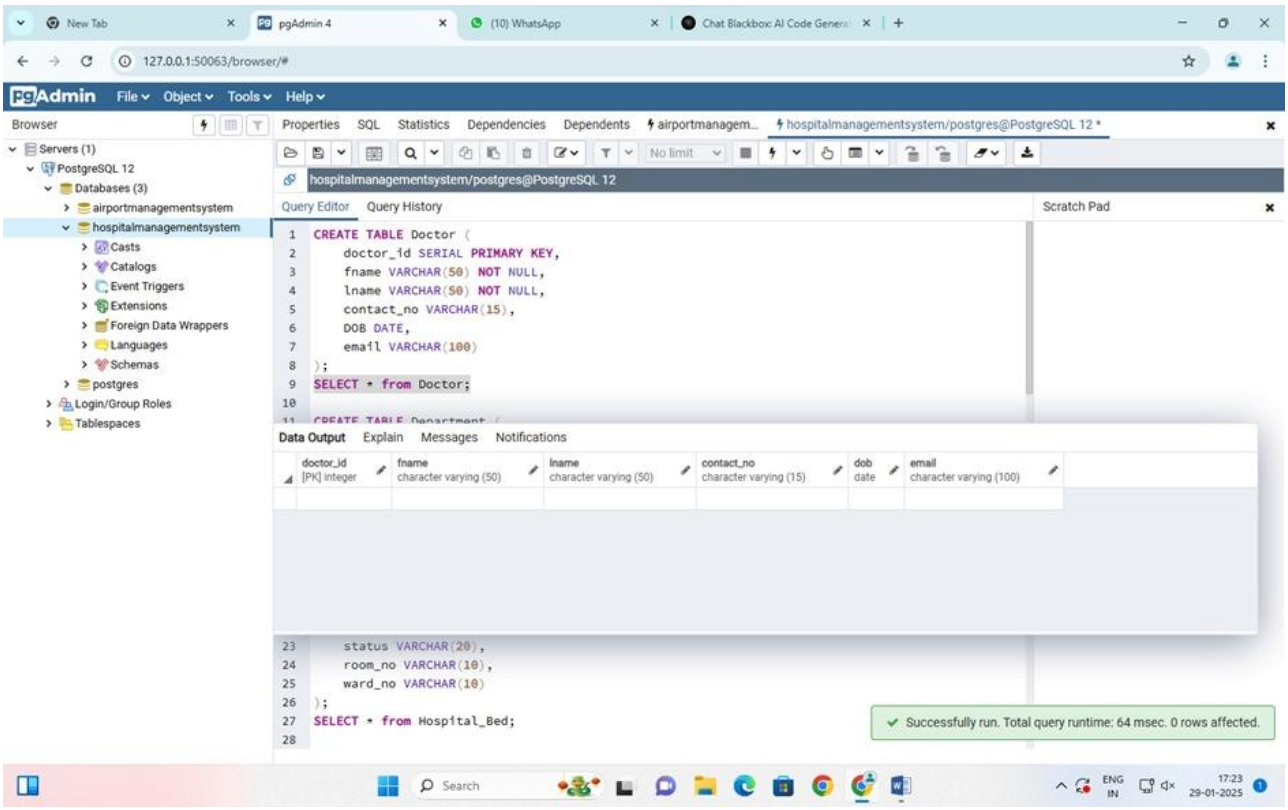
**<ColumnName><Datatype>(<size>)**

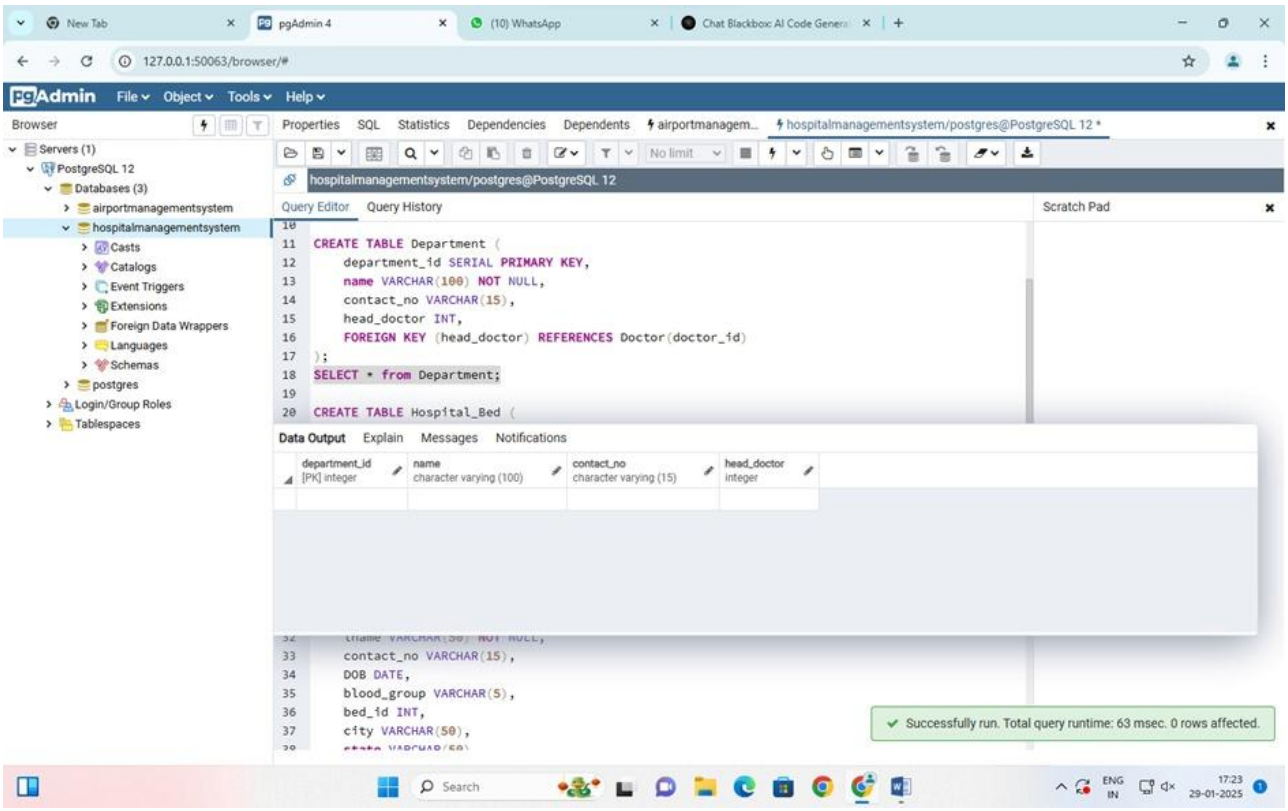
**<Columnname><Datatype>(<size>) UNIQUE(<ColumnName1>,<ColumnName2>);**

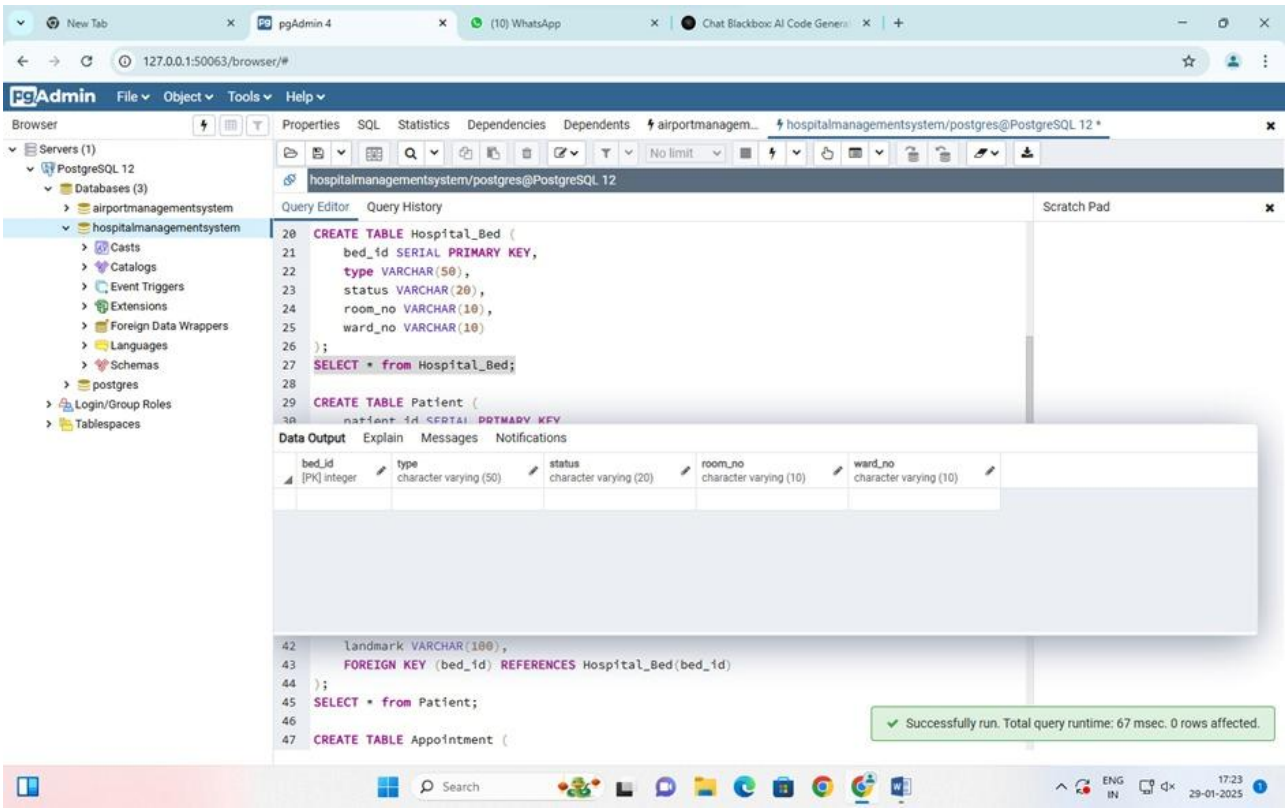
## Implementation Details (Problem Statement, Query and Screenshots of Results):

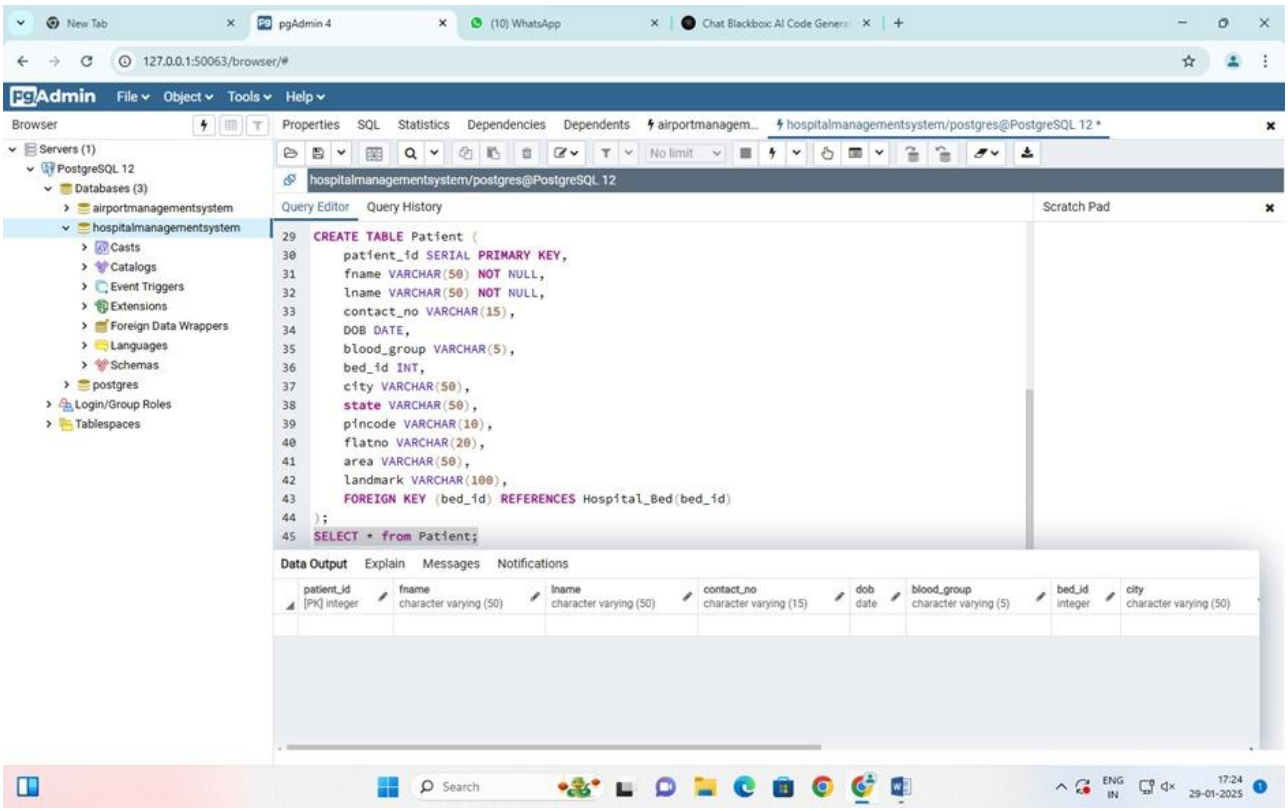
### Problem Statement

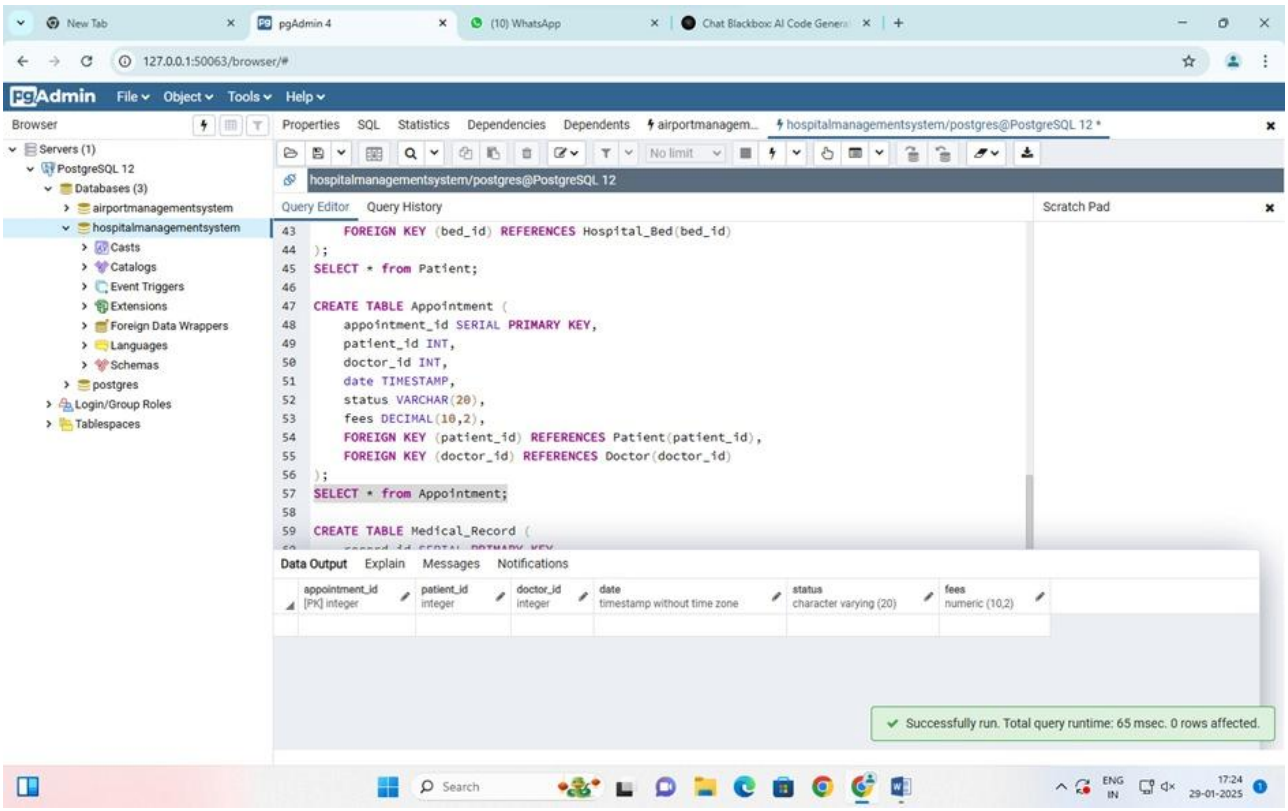
The problem revolves around designing a database for a Hospital Management System. It manages patient details, doctor information, department allocations, hospital bed assignments, appointment scheduling, medical records, prescriptions, and billing details, while maintaining essential constraints, specializations, generalizations, and relationships to ensure accurate and efficient management of hospital operations.

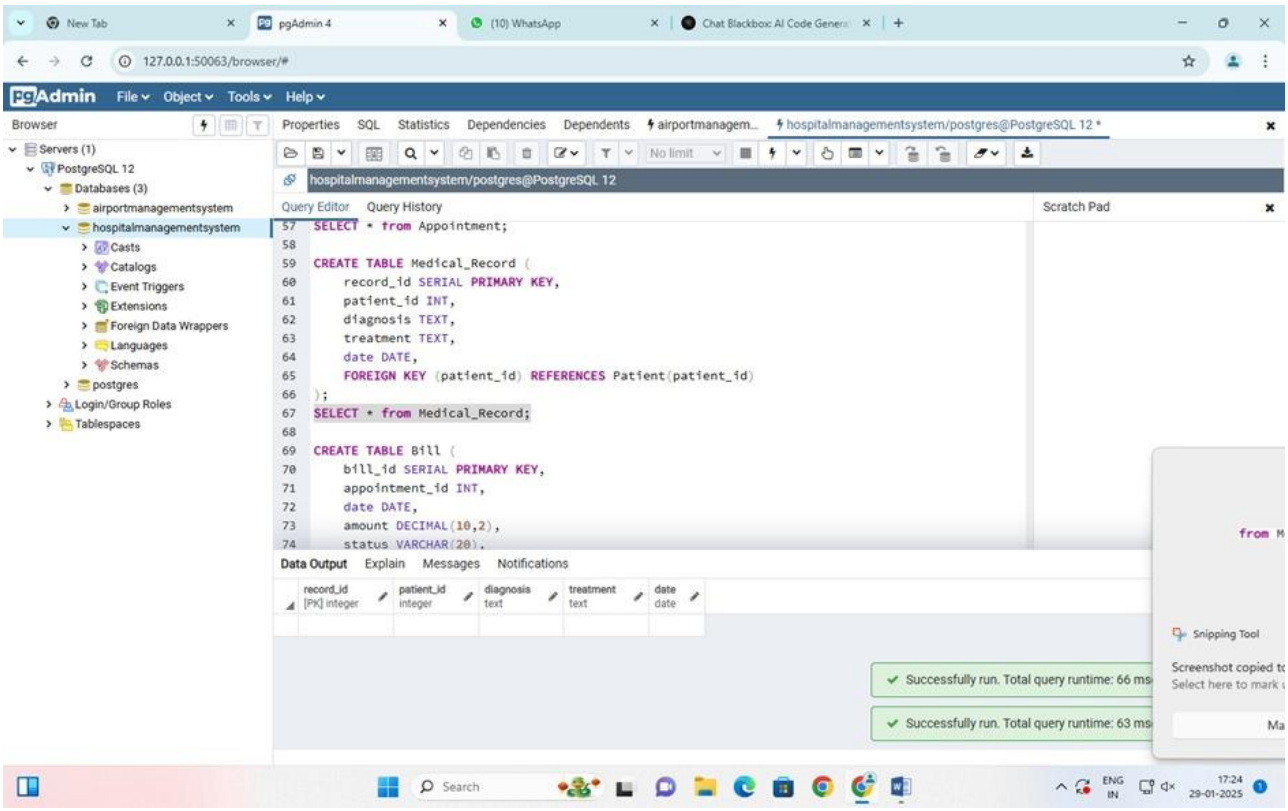


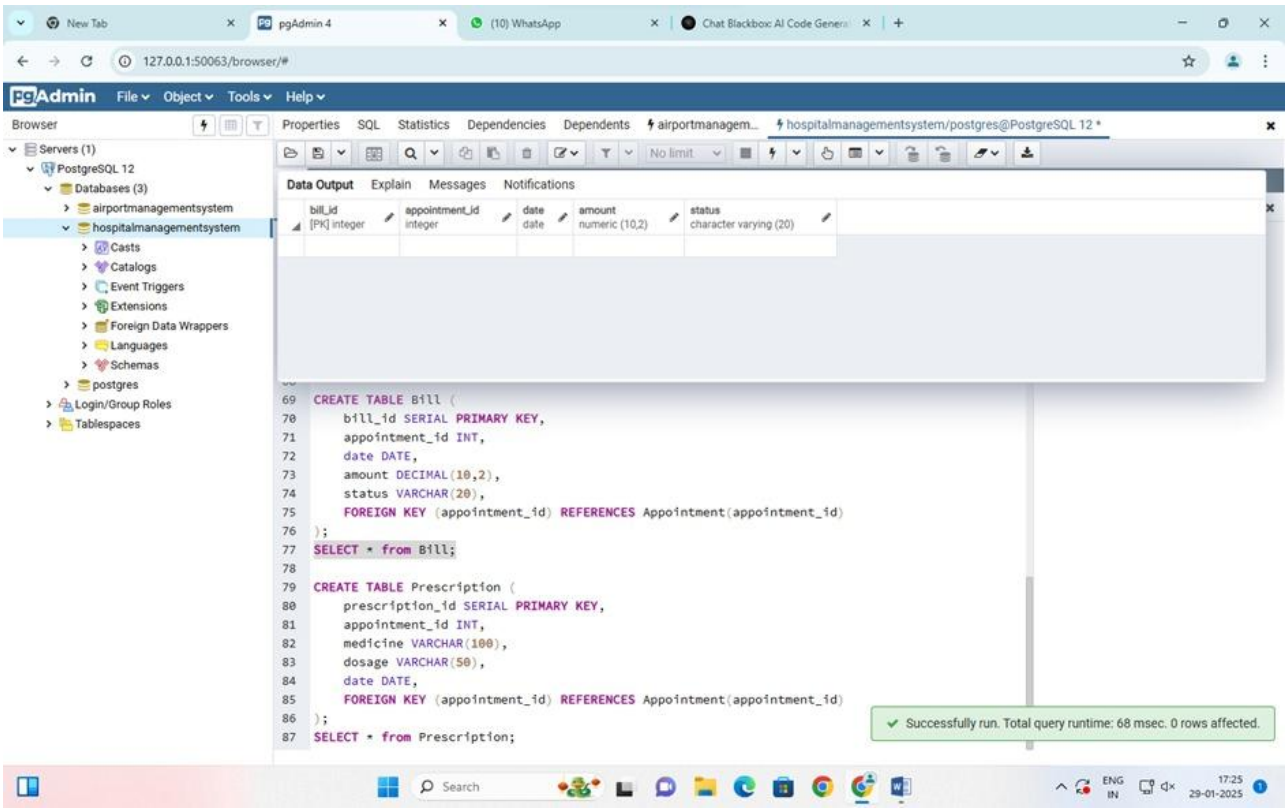

pgAdmin 4
127.0.0.1:50063/browser/#
pgAdmin File Object Tools Help
Browser
Servers (1)
PostgreSQL 12
Databases (3)
airportmanagementsystem
hospitalmanagementsystem
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas
postgres
Login/Group Roles
Tablespaces
Properties SQL Statistics Dependencies Dependents
hospitalmanagementsystem/postgres@PostgreSQL 12
Query Editor Query History
Scratch Pad
CREATE TABLE Department (
department_id SERIAL PRIMARY KEY,
name VARCHAR(100) NOT NULL,
contact_no VARCHAR(15),
head_doctor INT,
FOREIGN KEY (head_doctor) REFERENCES Doctor(doctor_id)
);
SELECT * from Department;
CREATE TABLE Hospital_Bed (
Data Output Explain Messages Notifications
department_id [PK] integer
name character varying (100)
contact_no character varying (15)
head_doctor integer
contact_no VARCHAR(15),
DOB DATE,
blood_group VARCHAR(5),
bed_id INT,
city VARCHAR(50),
Successfully run. Total query runtime: 63 msec. 0 rows affected.
Search
ENG IN
17:23 29-01-2025

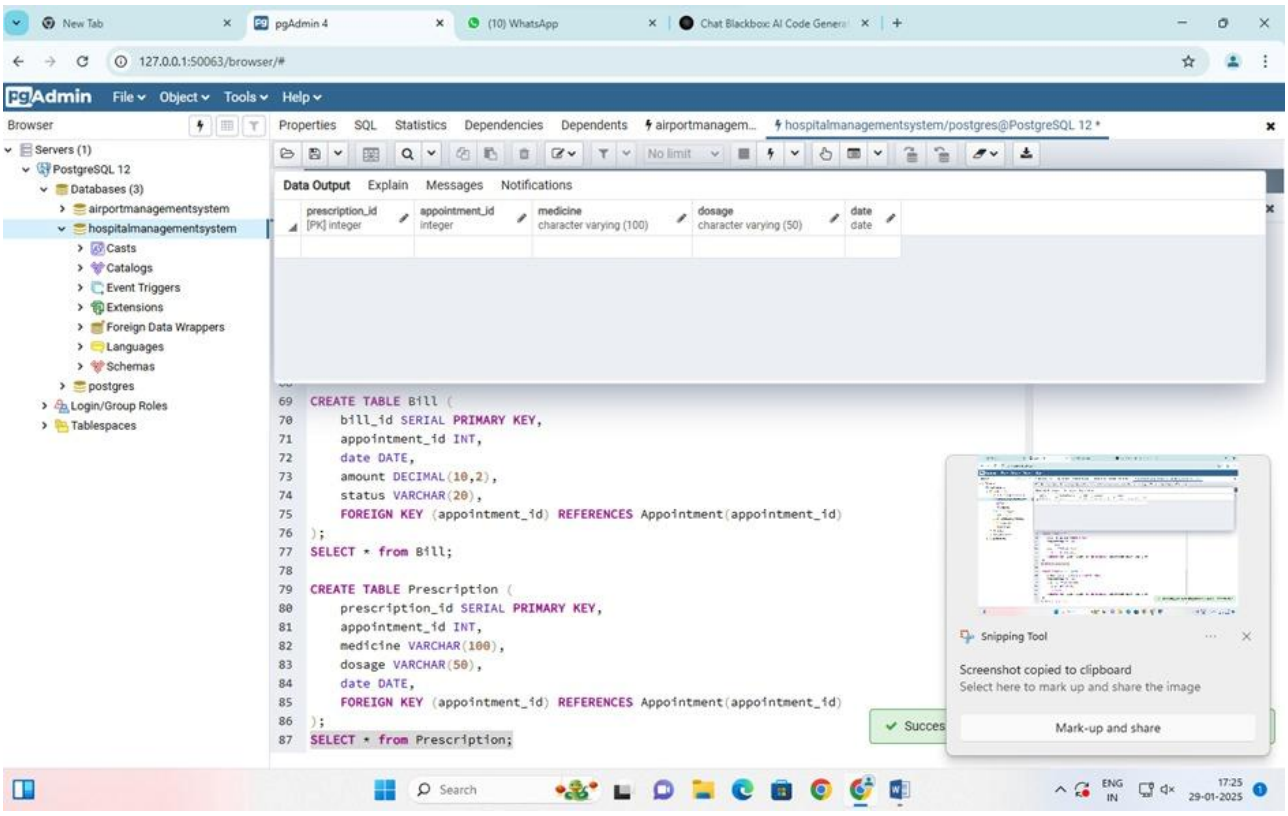

pgAdmin 4
127.0.0.1:50063/browser/#
pgAdmin File Object Tools Help
Browser
Properties SQL Statistics Dependencies Dependents airportmanagem... hospitalmanagementsystem/postgres@PostgreSQL 12 *
Servers (1)
PostgreSQL 12
Databases (3)
airportmanagementsystem
hospitalmanagementsystem
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas
postgres
Login/Group Roles
Tablespaces
hospitalmanagementsystem/postgres@PostgreSQL 12
Query Editor
Query History
Scratch Pad
CREATE TABLE Hospital_Bed (
bed_id SERIAL PRIMARY KEY,
type VARCHAR(50),
status VARCHAR(20),
room_no VARCHAR(10),
ward_no VARCHAR(10)
);
SELECT * from Hospital_Bed;
CREATE TABLE Patient (
Data Output
Explain
Messages
Notifications
bed_id [PK] integer
type character varying (50)
status character varying (20)
room_no character varying (10)
ward_no character varying (10)
landmark VARCHAR(100),
FOREIGN KEY (bed_id) REFERENCES Hospital_Bed(bed_id)
);
SELECT * from Patient;
CREATE TABLE Appointment (
Successfully run. Total query runtime: 67 msec. 0 rows affected.
Search
ENG IN
17:23 29-01-2025


pgAdmin 4
127.0.0.1:50063/browser/#
pgAdmin File Object Tools Help
Browser
Properties SQL Statistics Dependencies Dependents airportmanagem... hospitalmanagementsystem/postgres@PostgreSQL 12 *
Servers (1)
PostgreSQL 12
Databases (3)
airportmanagementsystem
hospitalmanagementsystem
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas
postgres
Login/Group Roles
Tablespaces
hospitalmanagementsystem/postgres@PostgreSQL 12
Query Editor
Query History
Scratch Pad
CREATE TABLE Patient (
patient_id SERIAL PRIMARY KEY,
fname VARCHAR(50) NOT NULL,
lname VARCHAR(50) NOT NULL,
contact_no VARCHAR(15),
DOB DATE,
blood_group VARCHAR(5),
bed_id INT,
city VARCHAR(50),
state VARCHAR(50),
pincode VARCHAR(10),
flatno VARCHAR(20),
area VARCHAR(50),
landmark VARCHAR(100),
FOREIGN KEY (bed_id) REFERENCES Hospital_Bed(bed_id)
);
SELECT * from Patient;
Data Output
Explain
Messages
Notifications
patient_id [PK] integer
fname character varying (50)
lname character varying (50)
contact_no character varying (15)
dob date
blood_group character varying (5)
bed_id integer
city character varying (50)
Search
ENG IN
17:24 29-01-2025


pgAdmin 4
127.0.0.1:50063/browser/#
pgAdmin File Object Tools Help
Browser
Properties SQL Statistics Dependencies Dependents airportmanagem... hospitalmanagementsystem/postgres@PostgreSQL 12 *
Servers (1)
PostgreSQL 12
Databases (3)
airportmanagementsystem
hospitalmanagementsystem
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas
postgres
Login/Group Roles
Tablespaces
hospitalmanagementsystem/postgres@PostgreSQL 12
Query Editor
Query History
Scratch Pad
FOREIGN KEY (bed_id) REFERENCES Hospital_Bed(bed_id)
);
SELECT * from Patient;
CREATE TABLE Appointment (
appointment_id SERIAL PRIMARY KEY,
patient_id INT,
doctor_id INT,
date TIMESTAMP,
status VARCHAR(20),
fees DECIMAL(10,2),
FOREIGN KEY (patient_id) REFERENCES Patient(patient_id),
FOREIGN KEY (doctor_id) REFERENCES Doctor(doctor_id)
);
SELECT * from Appointment;
CREATE TABLE Medical_Record (
Data Output
Explain
Messages
Notifications
appointment_id [PK] integer
patient_id integer
doctor_id integer
date timestamp without time zone
status character varying (20)
fees numeric (10,2)
Successfully run. Total query runtime: 65 msec. 0 rows affected.
Search
ENG IN
17:24 29-01-2025


pgAdmin 4
127.0.0.1:50063/browser/#
pgAdmin File Object Tools Help
Browser
Properties SQL Statistics Dependencies Dependents airportmanagem... hospitalmanagementsystem/postgres@PostgreSQL 12 *
Servers (1)
PostgreSQL 12
Databases (3)
airportmanagementsystem
hospitalmanagementsystem
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas
postgres
Login/Group Roles
Tablespaces
hospitalmanagementsystem/postgres@PostgreSQL 12
Query Editor
Query History
Scratch Pad
SELECT * from Appointment;
CREATE TABLE Medical_Record (
record_id SERIAL PRIMARY KEY,
patient_id INT,
diagnosis TEXT,
treatment TEXT,
date DATE,
FOREIGN KEY (patient_id) REFERENCES Patient(patient_id)
);
SELECT * from Medical_Record;
CREATE TABLE Bill (
bill_id SERIAL PRIMARY KEY,
appointment_id INT,
date DATE,
amount DECIMAL(10,2),
status VARCHAR(20),
Data Output
Explain
Messages
Notifications
record_id [PK] integer
patient_id integer
diagnosis text
treatment text
date date
Successfully run. Total query runtime: 66 ms
Successfully run. Total query runtime: 63 ms
Snipping Tool
Search
ENG IN
17:24 29-01-2025


pgAdmin 4
127.0.0.1:50063/browser/#
pgAdmin File Object Tools Help
Browser
Servers (1)
PostgreSQL 12
Databases (3)
airportmanagementsystem
hospitalmanagementsystem
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas
postgres
Login/Group Roles
Tablespaces
Properties SQL Statistics Dependencies Dependents
hospitalmanagementsystem/postgres@PostgreSQL 12 *
Data Output Explain Messages Notifications
bill_id [PK] integer
appointment_id integer
date date
amount numeric (10,2)
status character varying (20)
CREATE TABLE Bill (
bill_id SERIAL PRIMARY KEY,
appointment_id INT,
date DATE,
amount DECIMAL(10,2),
status VARCHAR(20),
FOREIGN KEY (appointment_id) REFERENCES Appointment(appointment_id)
);
SELECT * from Bill;
CREATE TABLE Prescription (
prescription_id SERIAL PRIMARY KEY,
appointment_id INT,
medicine VARCHAR(100),
dosage VARCHAR(50),
date DATE,
FOREIGN KEY (appointment_id) REFERENCES Appointment(appointment_id)
);
SELECT * from Prescription;
Successfully run. Total query runtime: 68 msec. 0 rows affected.
Search
ENG IN
17:25 29-01-2025


pgAdmin 4
127.0.0.1:50063/browser/#
pgAdmin
File
Object
Tools
Help
Browser
Servers (1)
PostgreSQL 12
Databases (3)
airportmanagementsystem
hospitalmanagementsystem
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas
postgres
Login/Group Roles
Tablespaces
Properties
SQL
Statistics
Dependencies
Dependents
Data Output
Explain
Messages
Notifications
prescription_id [PK] integer
appointment_id integer
medicine character varying (100)
dosage character varying (50)
date date
CREATE TABLE Bill (
bill_id SERIAL PRIMARY KEY,
appointment_id INT,
date DATE,
amount DECIMAL(10,2),
status VARCHAR(20),
FOREIGN KEY (appointment_id) REFERENCES Appointment(appointment_id)
);
SELECT * from Bill;
CREATE TABLE Prescription (
prescription_id SERIAL PRIMARY KEY,
appointment_id INT,
medicine VARCHAR(100),
dosage VARCHAR(50),
date DATE,
FOREIGN KEY (appointment_id) REFERENCES Appointment(appointment_id)
);
SELECT * from Prescription;
Snipping Tool
Screenshot copied to clipboard
Select here to mark up and share the image
Mark-up and share
Search
ENG
IN
17:25
29-01-2025

**Post Lab Questions:**

1. **Explain in brief the following terms:**

   a. **Database** - A database is an organized collection of data, so that it can be easily accessed and managed. You can organize data into tables, rows, columns, and index it to make it easier to find relevant information.

   b. **Types of databases**
   - Centralized Database
   - Distributed Database
   - NoSQL Database
   - Cloud Database
   - Relational Database
   - Network Database
   - Object-oriented Database
   - Hierarchical Database

   c. **SQL Data Types**
   - Numeric Data Types
   - Character and String Data Types
   - Date and Time Data Types
   - Binary Data Types
   - Boolean Data Types
   - Special Data Types

   d. **Foreign key -** In the relational databases, a foreign key is a field or a column that is used to establish a link between two tables. In simple words you can say that, a foreign key in one table used to point primary key in another table.

2. **What are the commands to:**

   a. **Delete an entire table**

   Ans. DROP TABLE table_name;

   b. **To view a database**

   Ans. SHOW DATABASES;

   USE database_name;

   SHOW TABLES;

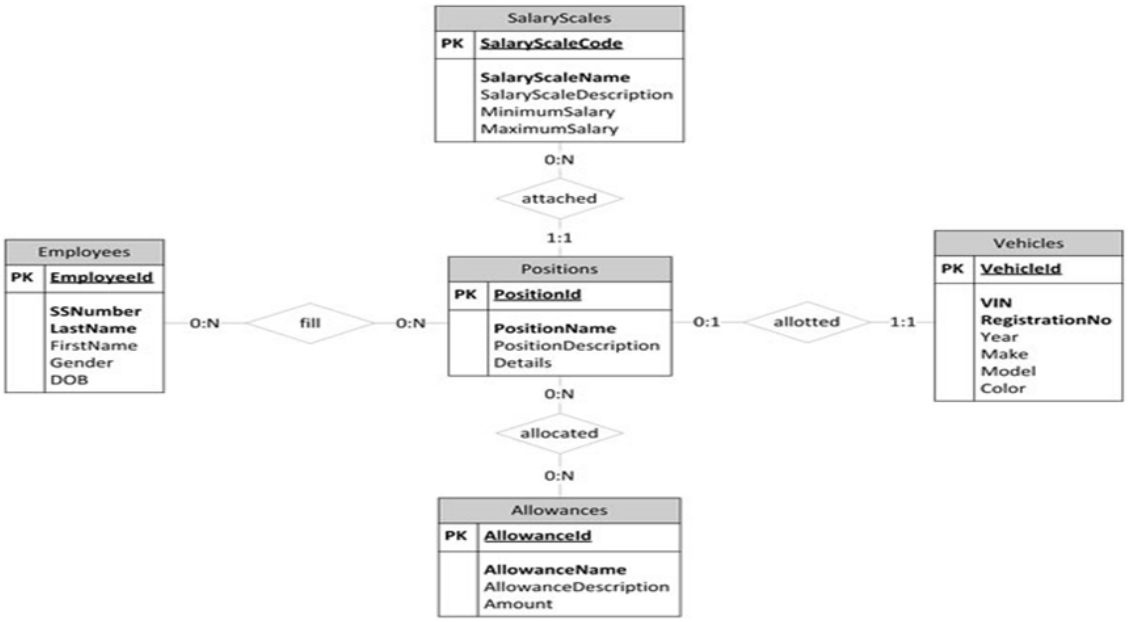
   c. **To select & view all the columns**

   Ans. SELECT * FROM table_name;

**3. For the given ER model, using DDL command: Write syntax to create CREATE Tables with all possible integrity constraints.**

**Problem Statement:**

A small accounting firm wants a simple HR application that will help it to keep track of its employees, their positions, allowances, salary scales, and which company vehicles their employees drive. The application must keep track of all the positions at the firm, the employees filling these positions, the allowances for these positions, the salary scales for these positions, and the company vehicles assigned to these positions.



## Code-

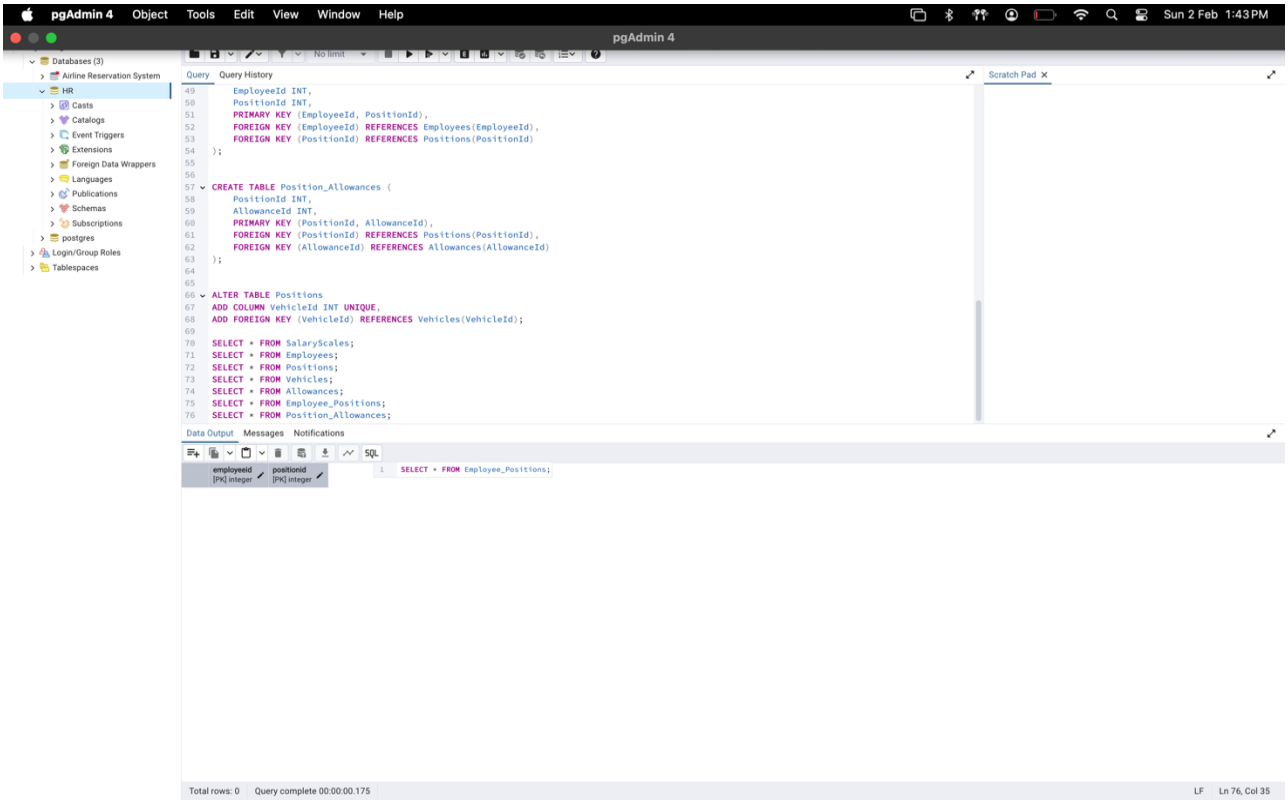
```
CREATE TABLE SalaryScales (
    SalaryScaleCode INT PRIMARY KEY,
    SalaryScaleName VARCHAR(100) NOT NULL,
    SalaryScaleDescription TEXT,
    MinimumSalary DECIMAL(10, 2) NOT NULL,
    MaximumSalary DECIMAL(10, 2) NOT NULL,
    CHECK (MinimumSalary <= MaximumSalary)
);

CREATE TABLE Employees (
    EmployeeId INT PRIMARY KEY,
    SSNumber VARCHAR(15) UNIQUE NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    FirstName VARCHAR(50) NOT NULL,
    Gender CHAR(1) CHECK (Gender IN ('M', 'F', 'O')),
    DOB DATE NOT NULL
);

CREATE TABLE Positions (
    PositionId INT PRIMARY KEY,
    PositionName VARCHAR(100) NOT NULL,
```

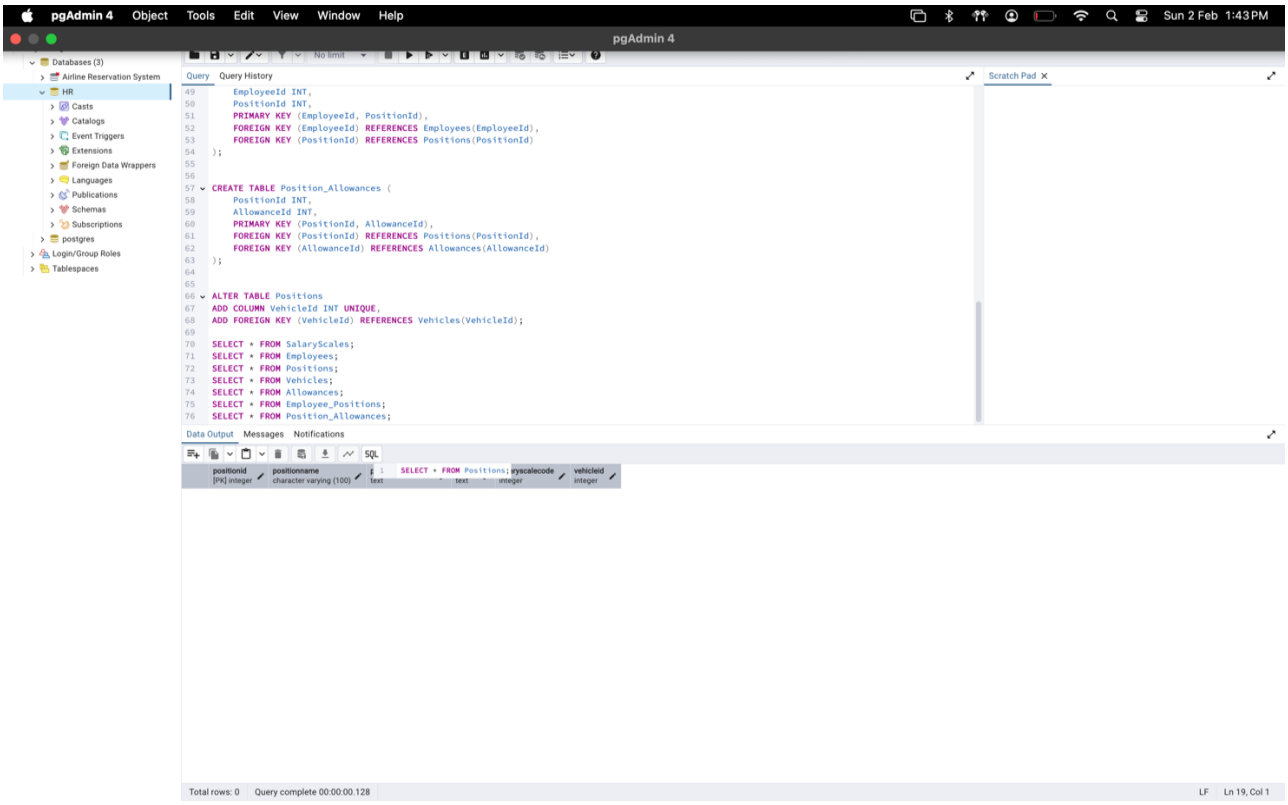
```
    PositionDescription TEXT,
    Details TEXT,
    SalaryScaleCode INT UNIQUE,
    FOREIGN KEY (SalaryScaleCode) REFERENCES SalaryScales(SalaryScaleCode)
);
CREATE TABLE Vehicles (
    VehicleId INT PRIMARY KEY,
    VIN VARCHAR(20) UNIQUE NOT NULL,
    RegistrationNo VARCHAR(20) UNIQUE NOT NULL,
    Year_R INT NOT NULL,
    Make VARCHAR(50) NOT NULL,
    Model VARCHAR(50) NOT NULL,
    Color VARCHAR(30) NOT NULL
);

CREATE TABLE Allowances (
    AllowanceId INT PRIMARY KEY,
    AllowanceName VARCHAR(100) NOT NULL,
    AllowanceDescription TEXT,
    Amount DECIMAL(10, 2) NOT NULL
);
CREATE TABLE Employee_Positions (
    EmployeeId INT,
```

```
    PositionId INT,
    PRIMARY KEY (EmployeeId, PositionId),
    FOREIGN KEY (EmployeeId) REFERENCES Employees(EmployeeId),
    FOREIGN KEY (PositionId) REFERENCES Positions(PositionId)
);

CREATE TABLE Position_Allowances (
    PositionId INT,
    AllowanceId INT,
    PRIMARY KEY (PositionId, AllowanceId),
    FOREIGN KEY (PositionId) REFERENCES Positions(PositionId),
    FOREIGN KEY (AllowanceId) REFERENCES Allowances(AllowanceId)
);

ALTER TABLE Positions
ADD COLUMN VehicleId INT UNIQUE,
ADD FOREIGN KEY (VehicleId) REFERENCES Vehicles(VehicleId);

SELECT * FROM SalaryScales;
SELECT * FROM Employees;
SELECT * FROM Positions;
SELECT * FROM Vehicles;
SELECT * FROM Allowances;
```

SELECT * FROM Employee_Positions;

SELECT * FROM Position_Allowances;




pgAdmin 4 Object Tools Edit View Window Help
Sun 2 Feb 1:43 PM
pgAdmin 4
Databases (3)
Airline Reservation System
HR
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas
Subscriptions
postgres
Login/Group Roles
Tablespaces
Query Query History
Scratch Pad
EmployeeId INT,
PositionId INT,
PRIMARY KEY (EmployeeId, PositionId),
FOREIGN KEY (EmployeeId) REFERENCES Employees(EmployeeId),
FOREIGN KEY (PositionId) REFERENCES Positions(PositionId)
);
CREATE TABLE Position_Allowances (
PositionId INT,
AllowanceId INT,
PRIMARY KEY (PositionId, AllowanceId),
FOREIGN KEY (PositionId) REFERENCES Positions(PositionId),
FOREIGN KEY (AllowanceId) REFERENCES Allowances(AllowanceId)
);
ALTER TABLE Positions
ADD COLUMN VehicleId INT UNIQUE,
ADD FOREIGN KEY (VehicleId) REFERENCES Vehicles(VehicleId);
SELECT * FROM SalaryScales;
SELECT * FROM Employees;
SELECT * FROM Positions;
SELECT * FROM Vehicles;
SELECT * FROM Allowances;
SELECT * FROM Employee_Positions;
SELECT * FROM Position_Allowances;
Data Output Messages Notifications
positionid [PK] integer
positionname character varying (100)
vehicleid integer
Total rows: 0 Query complete 00:00:00.128
Ln 19, Col 1

## Conclusion:

SQL DDL is essential for defining and managing database structures. Commands like CREATE, ALTER, and DROP help create, modify, and delete tables while maintaining data integrity through constraints.