**Experiment No.: 03**

**Title:** To implement database for relational model in Experiment no. 2 using DDL statements.

**Batch:SY-IT(B3)** **Roll No.:16010423076** **Experiment No.: 03**

**Aim:** To implement database for relational model in experiment no. 2 using DDL statements (Virtual Lab).

---

**Resources needed:** PostgreSQL PgAdmin3

---

**Theory:**

The Data Definition Language (DDL) is used to create and modify the relational schema. Also it is used to add various constraints to the table like the primary key, foreign key, check constraint, not null constraint and unique constraint.

The DDL statements are:

CREATE

DROP

ALTER

PostgreSQL supports the standard SQL types int, smallint, real, double precision, char(N), varchar(N), date, time, timestamp, and interval for creating tables.

---

**Procedure:**

**Create Database and use it:**

$ createdb mydb

$ psql mydb

**Delete a database:** $

dropdb mydb

**Create table:**

```
CREATE TABLE my_first_table (
first_column text,
second_column integer
);

CREATE TABLE products (
product_no integer,
name text, price numeric);
```

**Drop Table:**

```
DROP TABLE my_first_table;
DROP TABLE products;
```

**Default Value:**

```
CREATE TABLE products (
product_no integer,
name text,
price numeric DEFAULT 9.99 );
```

**Constraints:**

1. **Primary Key**

```
CREATE TABLE products (
product_no integer PRIMARY KEY,
name text,
price numeric );
```

Primary keys can also constrain more than one column.

```
CREATE TABLE example (
a integer,
b integer,
c integer,
PRIMARY KEY (a, c)
);
```

2. **Check Constraint**

```
CREATE TABLE products (
product_no integer,
name text,
price numeric CHECK (price > 0) );
```

3. **Not Null Constraint**

```
CREATE TABLE products (
product_no integer NOT NULL,
name text NOT NULL,
price numeric );
```

4. **Unique Constraint**

```
CREATE TABLE products (
product_no integer UNIQUE,
name text,
price numeric );
```

5. **Foreign Key Constarint**

```
CREATE TABLE products (
product_no integer PRIMARY KEY,
name text,
price numeric );

CREATE TABLE orders (
order_id integer PRIMARY KEY,
product_no integer REFERENCES products (product_no),
quantity integer );
```

Here a foreign key constraint in the order table references the products table.

**Modifying table:**

**Adding column**
ALTER TABLE products ADD COLUMN description text;

**Removing column**
ALTER TABLE products DROP COLUMN description;

**Adding Constraint**
ALTER TABLE products ADD CONSTRAINT some_name UNIQUE (product_no); ALTER TABLE products ADD FOREIGN KEY (product_group_id) REFERENCES product_groups;

**Removing Constraint**
ALTER TABLE products DROP CONSTRAINT some_name;

**Adding Not Null Constraint**
ALTER TABLE products ALTER COLUMN product_no SET NOT NULL;

**Removing Not Null Constraint**
ALTER TABLE products ALTER COLUMN product_no DROP NOT NULL;

---

**Results: (Queries printout with output)**

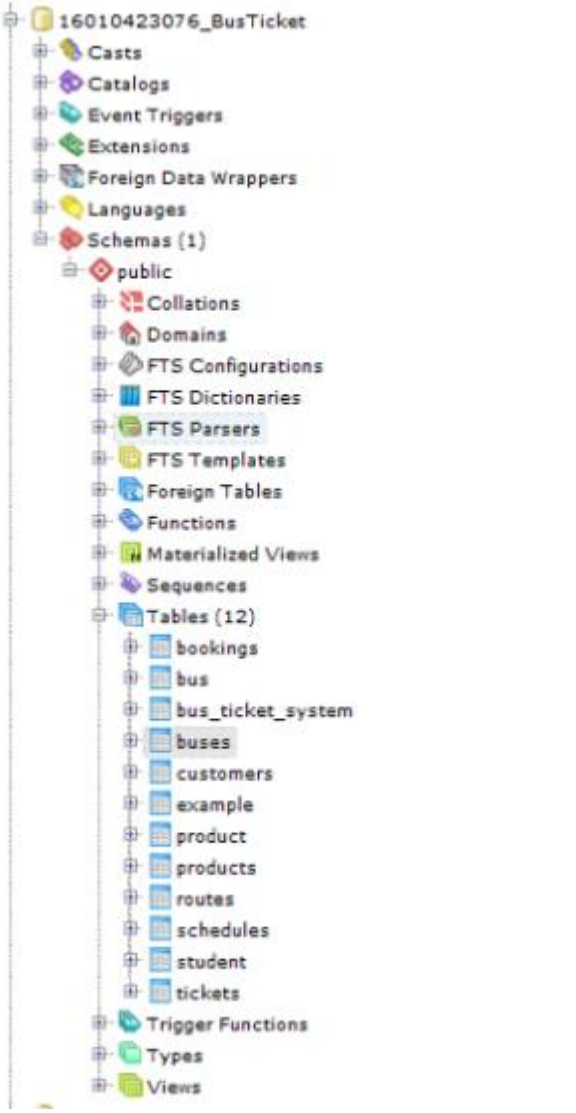
**Creating the tables :**

```
-- Create the buses table
CREATE TABLE bus (
bus_id INTEGER PRIMARY KEY,
bus_number VARCHAR(20) UNIQUE NOT NULL,
capacity INTEGER NOT NULL CHECK (capacity > 0)
);

-- Create the routes table
CREATE TABLE routes (
route_id INTEGER PRIMARY KEY,
origin VARCHAR(100) NOT NULL,
destination VARCHAR(100) NOT NULL,
distance INTEGER NOT NULL CHECK (distance > 0)
);
-- Create the schedules table
CREATE TABLE schedules (
schedule_id INTEGER PRIMARY KEY,
route_id INTEGER NOT NULL REFERENCES routes (route_id),
bus_id INTEGER NOT NULL REFERENCES buses (bus_id),
departure_time TIMESTAMP NOT NULL,
```

```
arrival_time TIMESTAMP NOT NULL,
CHECK (departure_time < arrival_time)
);

-- Create the customers table
CREATE TABLE customers (
customer_id INTEGER PRIMARY KEY,
name VARCHAR(100) NOT NULL,
email VARCHAR(100) UNIQUE NOT NULL,
phone_number VARCHAR(15) UNIQUE NOT NULL
);
-- Create the bookings table
CREATE TABLE bookings (
booking_id INTEGER PRIMARY KEY,
customer_id INTEGER NOT NULL REFERENCES customers (customer_id),
schedule_id INTEGER NOT NULL REFERENCES schedules (schedule_id),
booking_date TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
UNIQUE (customer_id, schedule_id, booking_date)
);
-- Create the tickets table
CREATE TABLE tickets (
ticket_id INTEGER PRIMARY KEY,
booking_id INTEGER NOT NULL REFERENCES bookings (booking_id),
seat_number VARCHAR(10) NOT NULL,
UNIQUE (booking_id, seat_number)
);
```

**Output Screenshots :**

Database -> Schemas ->Public->Tables->Constraints created

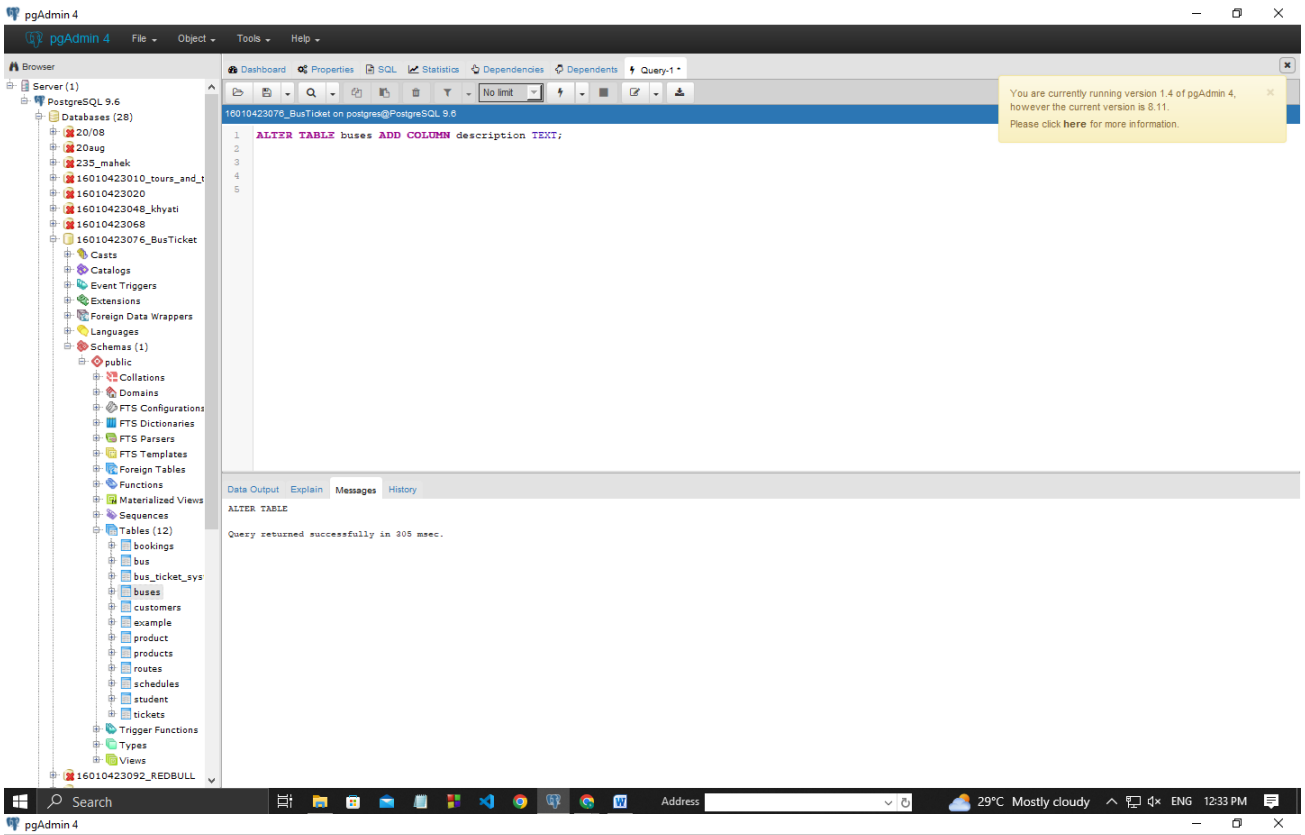


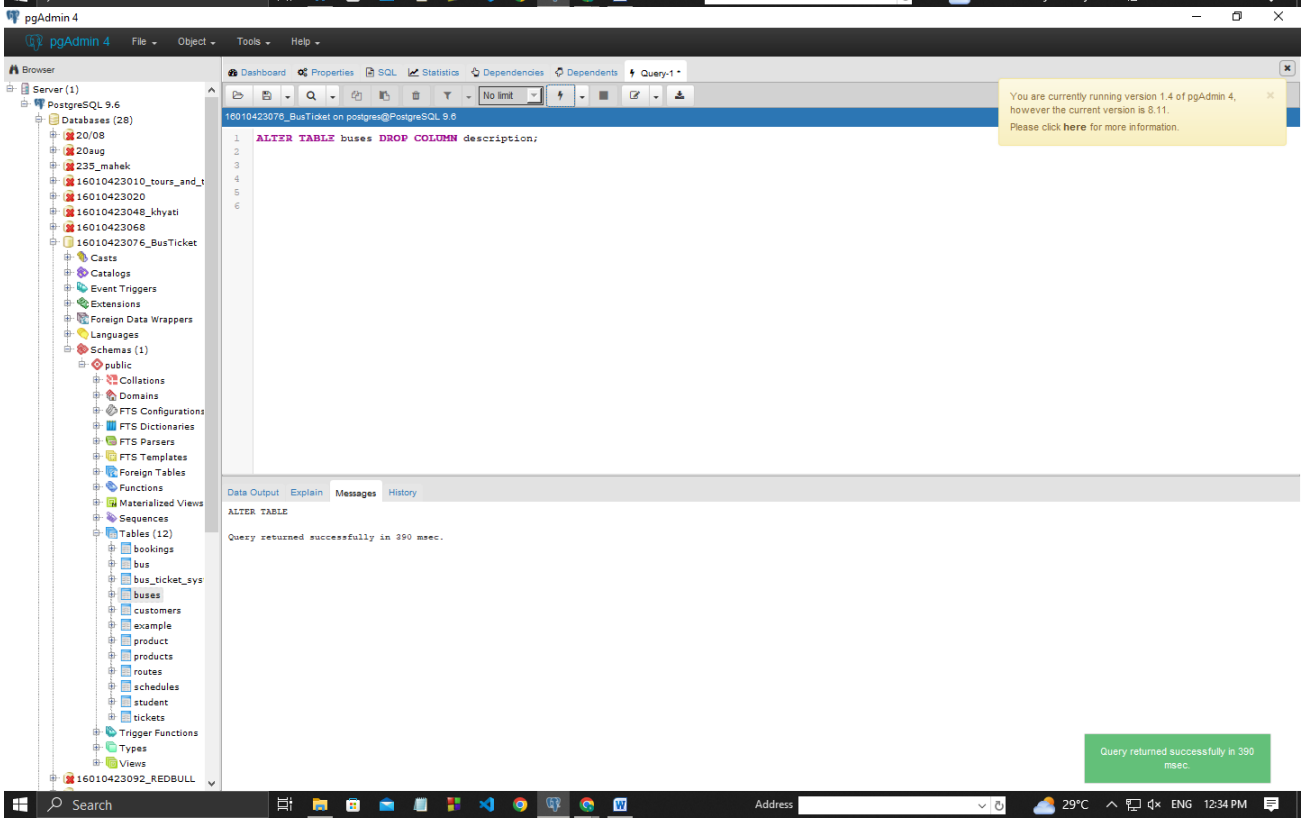

## Modifying the tables :

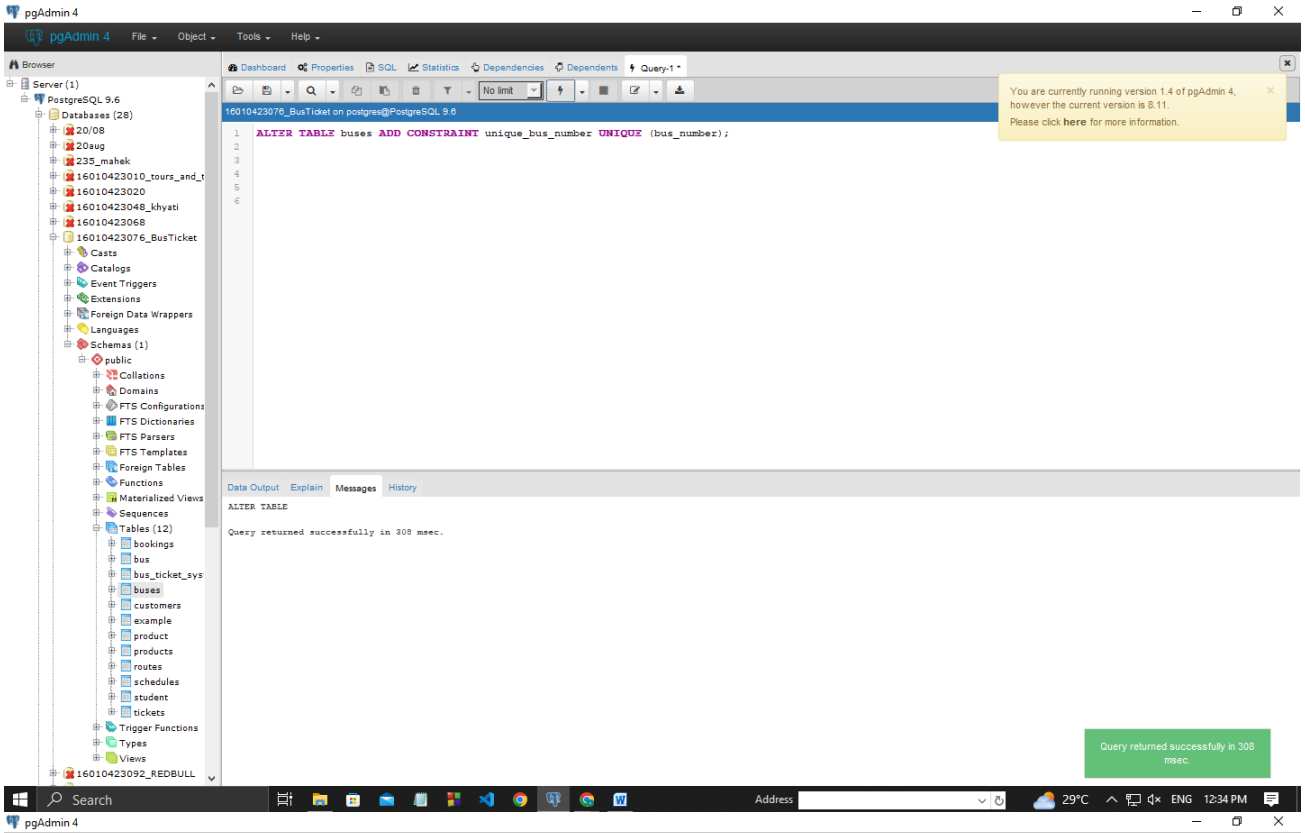
ALTER TABLE buses ADD COLUMN description TEXT;
ALTER TABLE buses DROP COLUMN description;
ALTER TABLE buses ADD CONSTRAINT unique_bus_number UNIQUE (bus_number);
ALTER TABLE schedules ADD CONSTRAINT fk_bus FOREIGN KEY (bus_id) REFERENCES buses (bus_id);
ALTER TABLE buses DROP CONSTRAINT unique_bus_number;

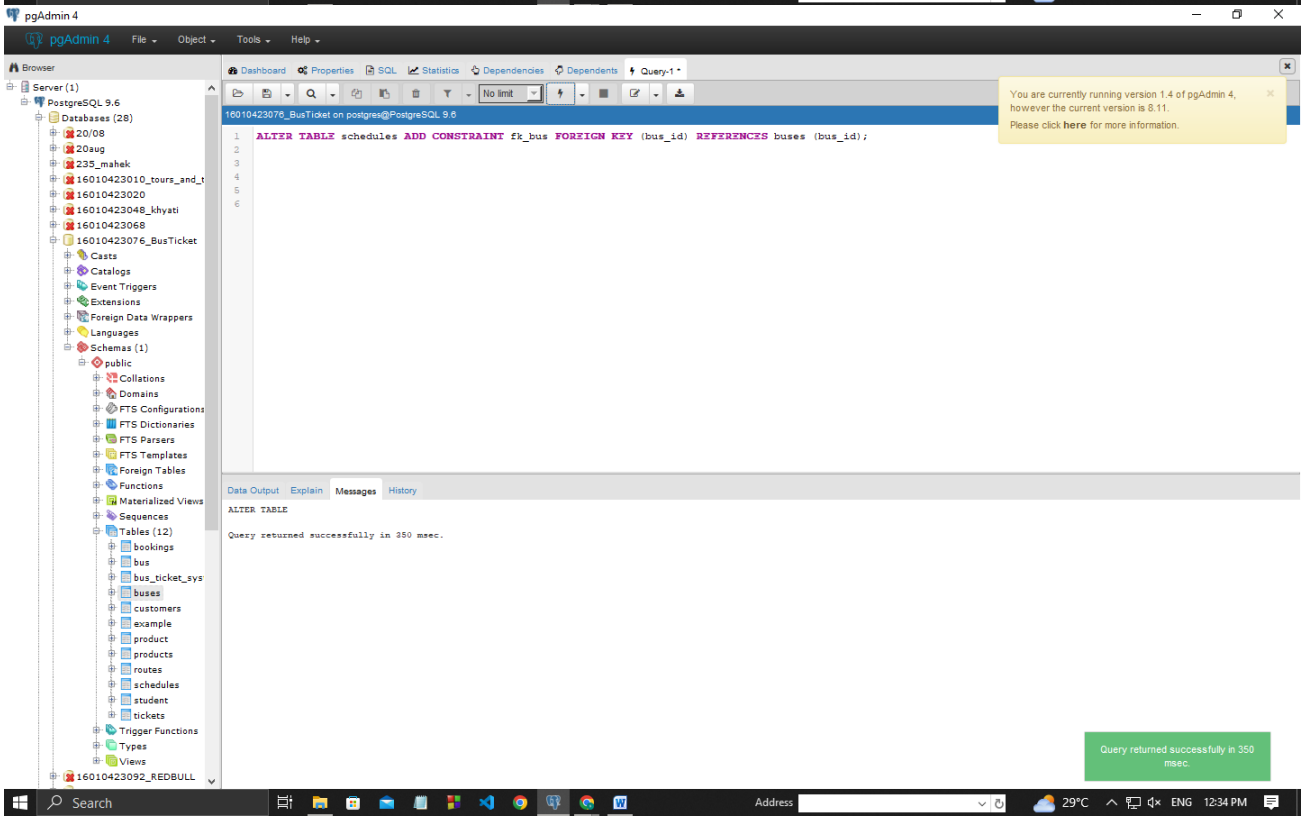
ALTER TABLE buses ALTER COLUMN bus_number SET NOT NULL;
ALTER TABLE buses ALTER COLUMN bus_number DROP NOT NULL;

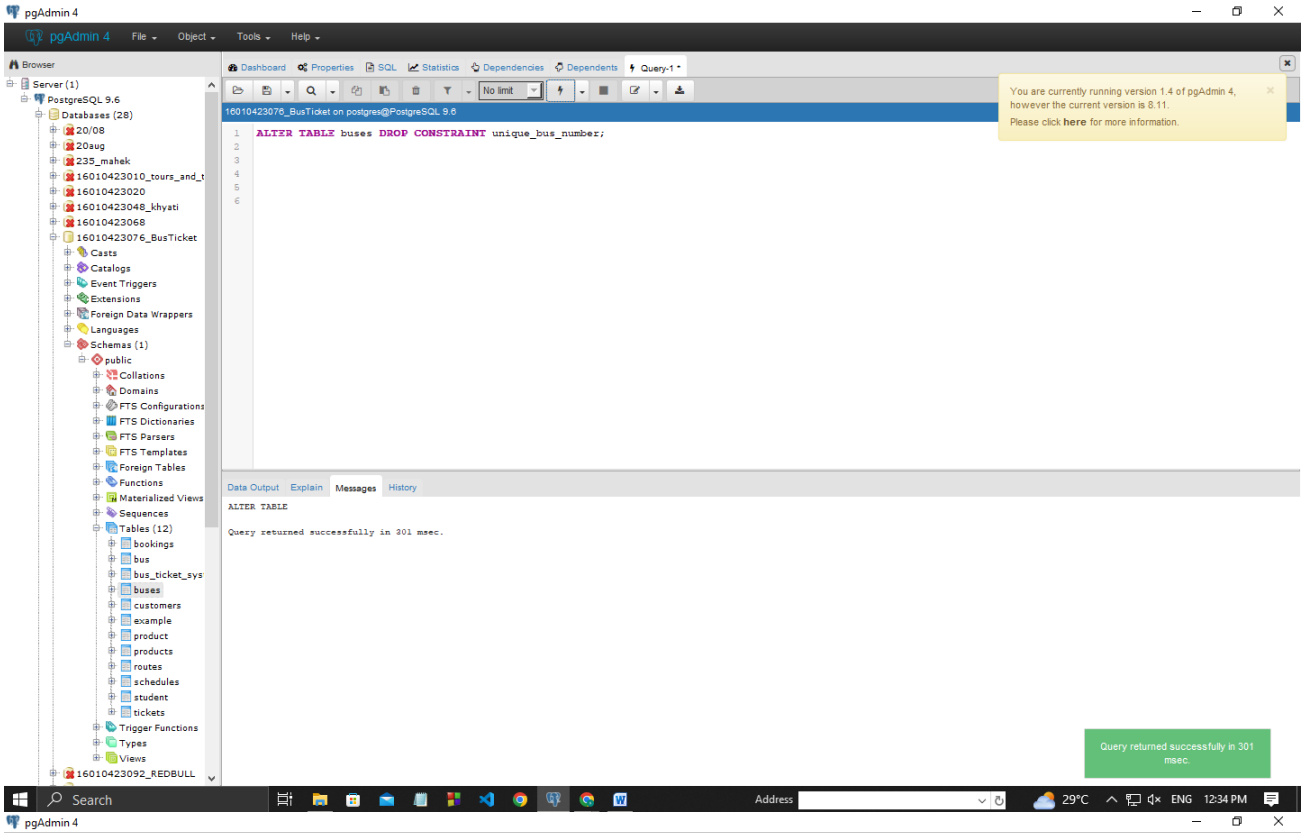
**Output Screenshots :**

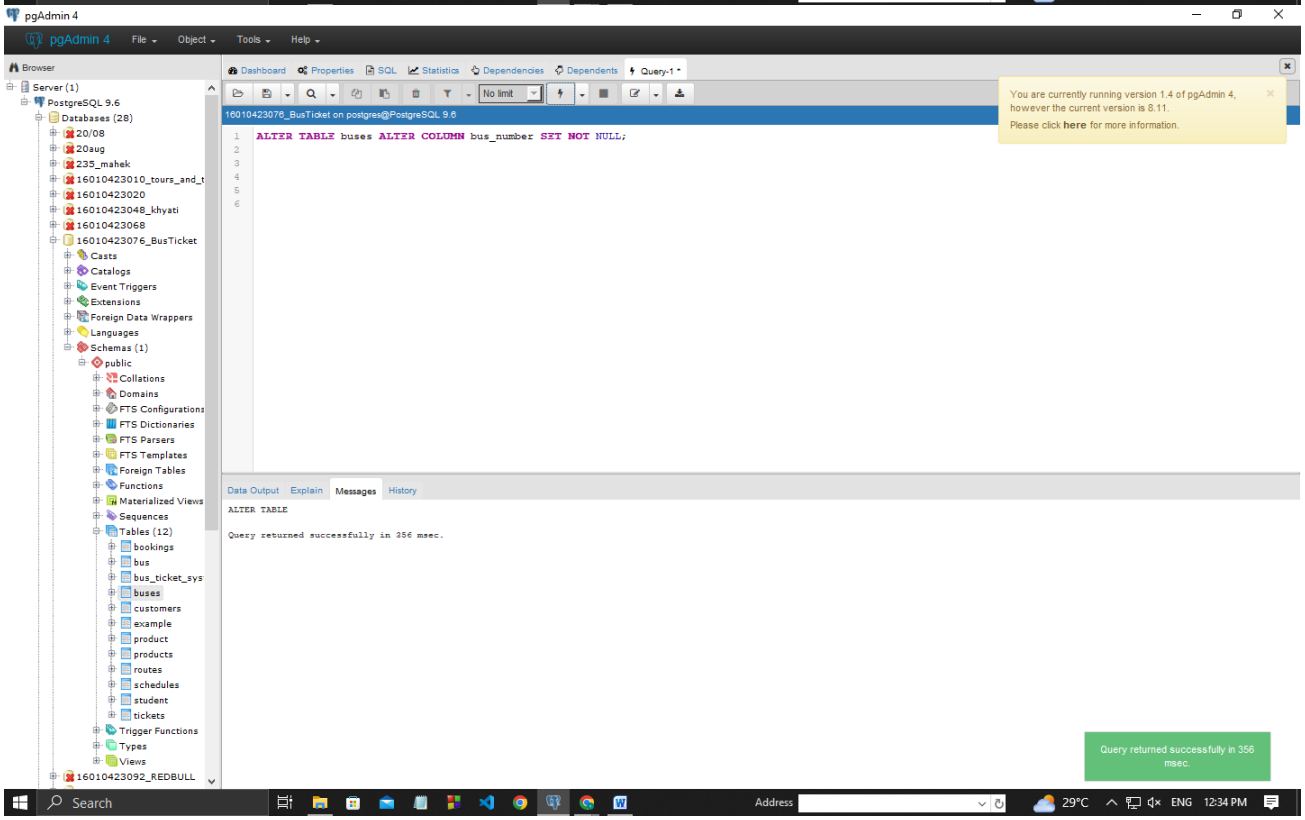

pgAdmin 4
File
Object
Tools
Help
Browser
Server (1)
PostgreSQL 9.6
Databases (28)
16010423076_BusTicket
Tables (12)
bookings
bus
bus_ticket_sys
buses
customers
example
product
products
routes
schedules
student
tickets
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Query-1
No limit
16010423076_BusTicket on postgres@PostgreSQL 9.6
ALTER TABLE buses ADD COLUMN description TEXT;
You are currently running version 1.4 of pgAdmin 4, however the current version is 8.11.
Please click here for more information.
Data Output
Explain
Messages
History
ALTER TABLE
Query returned successfully in 305 msec.
Search
Address
29°C Mostly cloudy
ENG
12:33 PM

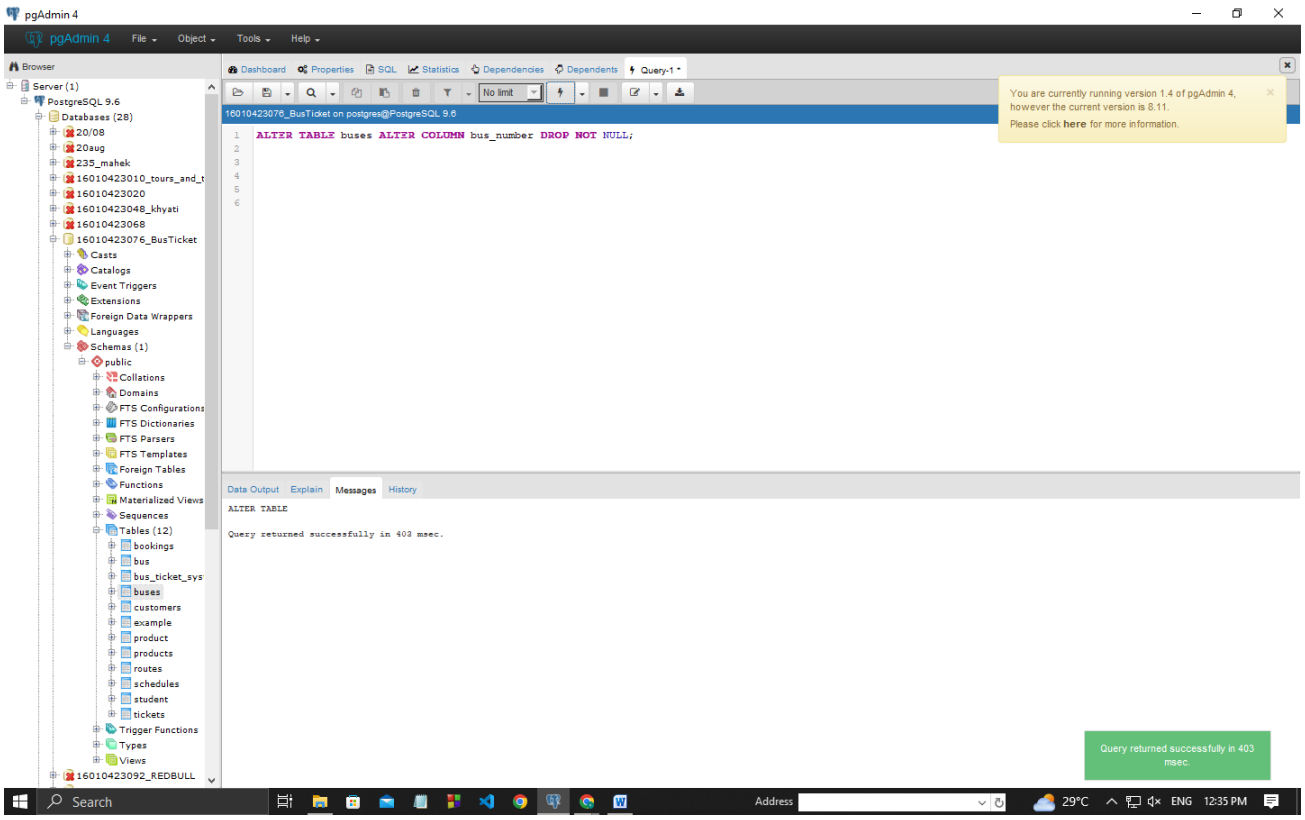

pgAdmin 4
File
Object
Tools
Help
Browser
Server (1)
PostgreSQL 9.6
Databases (28)
16010423076_BusTicket
Tables (12)
bookings
bus
bus_ticket_sys
buses
customers
example
product
products
routes
schedules
student
tickets
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Query-1
No limit
16010423076_BusTicket on postgres@PostgreSQL 9.6
ALTER TABLE buses DROP COLUMN description;
You are currently running version 1.4 of pgAdmin 4, however the current version is 8.11.
Please click here for more information.
Data Output
Explain
Messages
History
ALTER TABLE
Query returned successfully in 390 msec.
Query returned successfully in 390 msec.
Search
Address
29°C
ENG
12:34 PM


pgAdmin 4
File
Object
Tools
Help
Browser
Server (1)
PostgreSQL 9.6
Databases (28)
16010423076_BusTicket
Schemas (1)
public
Tables (12)
bookings
bus
bus_ticket_sys
buses
customers
example
product
products
routes
schedules
student
tickets
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Query-1
No limit
16010423076_BusTicket on postgres@PostgreSQL 9.6
ALTER TABLE buses ADD CONSTRAINT unique_bus_number UNIQUE (bus_number);
You are currently running version 1.4 of pgAdmin 4, however the current version is 8.11.
Please click here for more information.
Data Output
Explain
Messages
History
ALTER TABLE
Query returned successfully in 308 msec.
Query returned successfully in 308 msec.
Search
Address
29°C
ENG
12:34 PM


pgAdmin 4
File
Object
Tools
Help
Browser
Server (1)
PostgreSQL 9.6
Databases (28)
16010423076_BusTicket
Schemas (1)
public
Tables (12)
bookings
bus
bus_ticket_sys
buses
customers
example
product
products
routes
schedules
student
tickets
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Query-1
No limit
16010423076_BusTicket on postgres@PostgreSQL 9.6
ALTER TABLE schedules ADD CONSTRAINT fk_bus FOREIGN KEY (bus_id) REFERENCES buses (bus_id);
You are currently running version 1.4 of pgAdmin 4, however the current version is 8.11.
Please click here for more information.
Data Output
Explain
Messages
History
ALTER TABLE
Query returned successfully in 350 msec.
Query returned successfully in 350 msec.
Search
Address
29°C
ENG
12:34 PM


pgAdmin 4
File
Object
Tools
Help
Browser
Server (1)
PostgreSQL 9.6
Databases (28)
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Query-1
No limit
16010423076_BusTicket on postgres@PostgreSQL 9.6
ALTER TABLE buses DROP CONSTRAINT unique_bus_number;
You are currently running version 1.4 of pgAdmin 4, however the current version is 8.11.
Please click here for more information.
Data Output
Explain
Messages
History
ALTER TABLE
Query returned successfully in 301 msec.
Query returned successfully in 301 msec.
Search
Address
29°C
ENG
12:34 PM


pgAdmin 4
File
Object
Tools
Help
Browser
Server (1)
PostgreSQL 9.6
Databases (28)
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Query-1
No limit
16010423076_BusTicket on postgres@PostgreSQL 9.6
ALTER TABLE buses ALTER COLUMN bus_number SET NOT NULL;
You are currently running version 1.4 of pgAdmin 4, however the current version is 8.11.
Please click here for more information.
Data Output
Explain
Messages
History
ALTER TABLE
Query returned successfully in 356 msec.
Query returned successfully in 356 msec.
Search
Address
29°C
ENG
12:34 PM




**Insert 5 rows in 2 tables :**

-- Insert 5 rows into the buses table

INSERT INTO buses (bus_id, bus_number, capacity) VALUES

(1, 'BUS101', 50),

(2, 'BUS102', 45),

(3, 'BUS103', 60),

(4, 'BUS104', 55),

(5, 'BUS105', 40);

-- Insert 5 rows into the routes table

INSERT INTO routes (route_id, origin, destination, distance) VALUES

(1, 'City A', 'City B', 120),

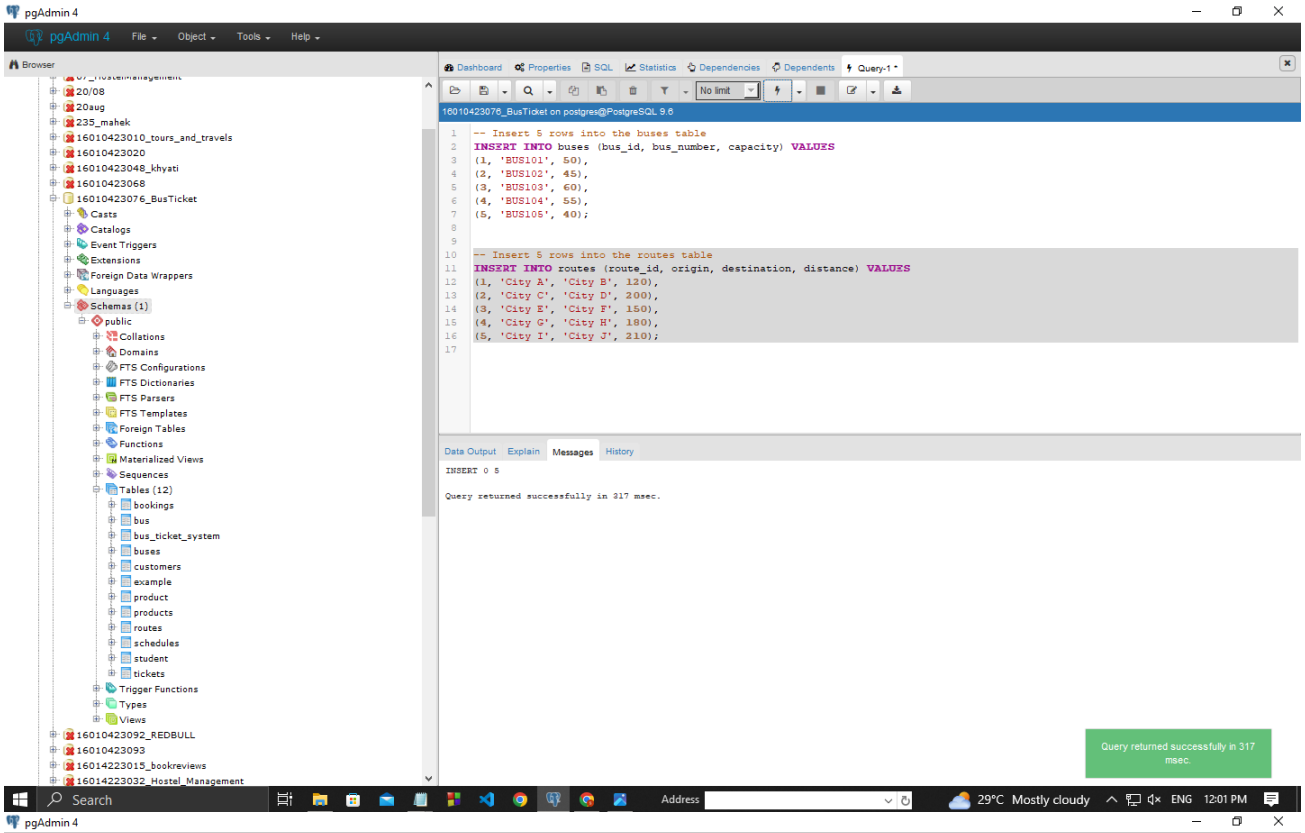
(2, 'City C', 'City D', 200),

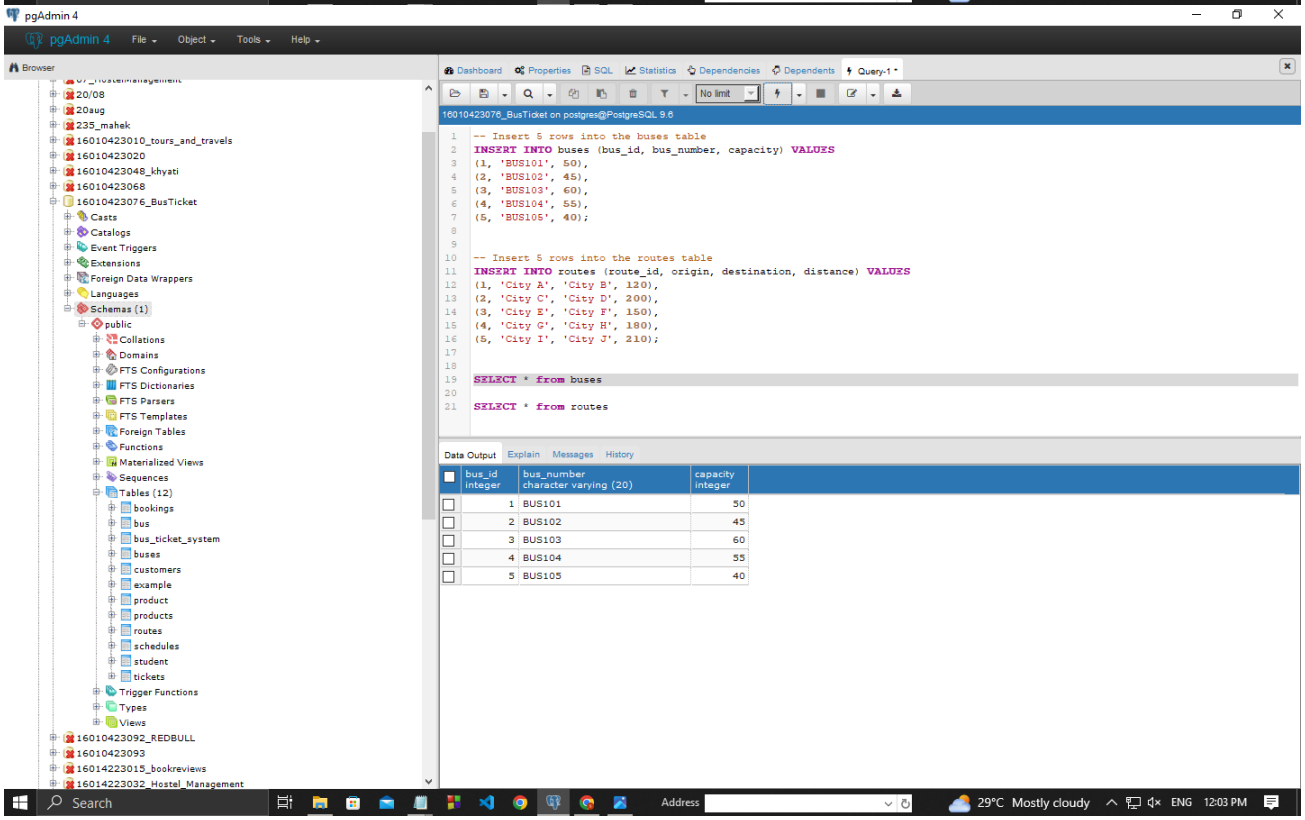
(3, 'City E', 'City F', 150),

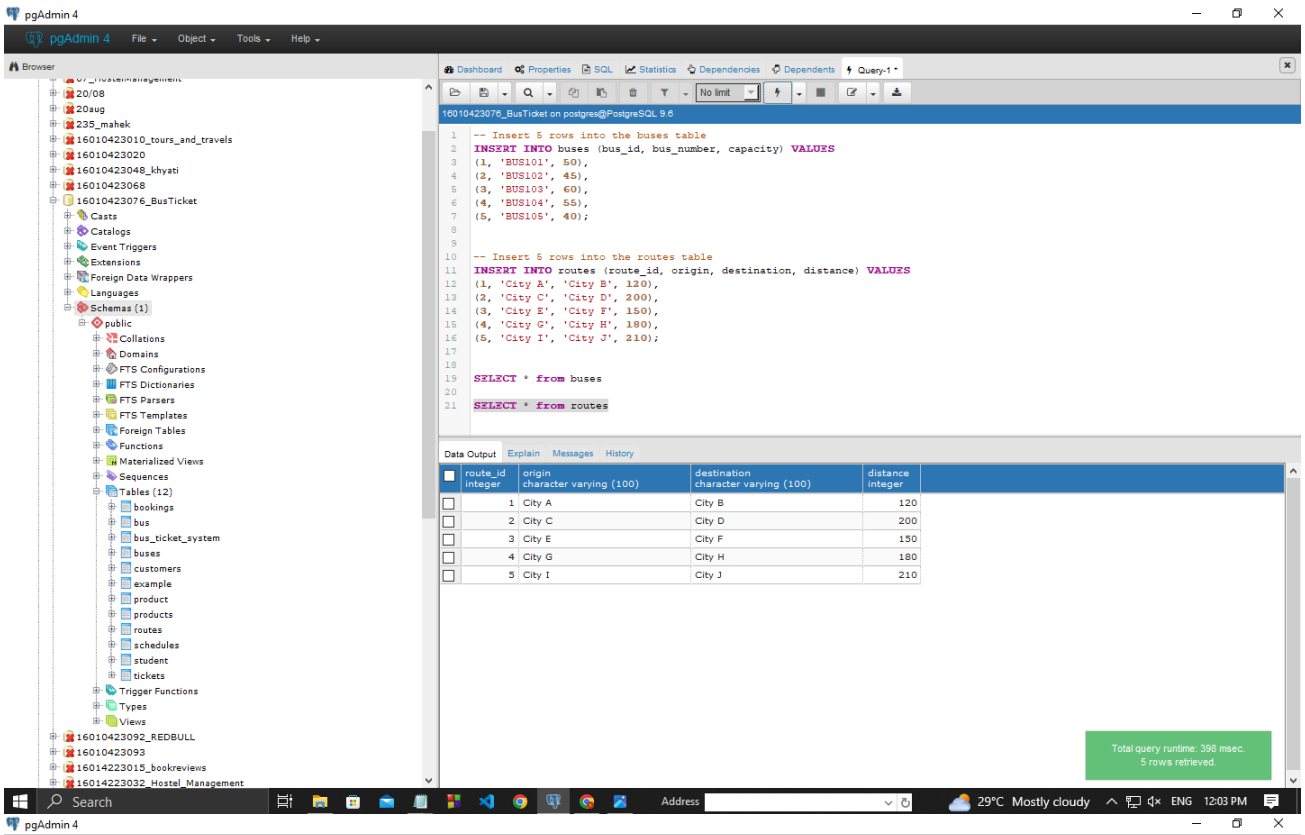
(4, 'City G', 'City H', 180),

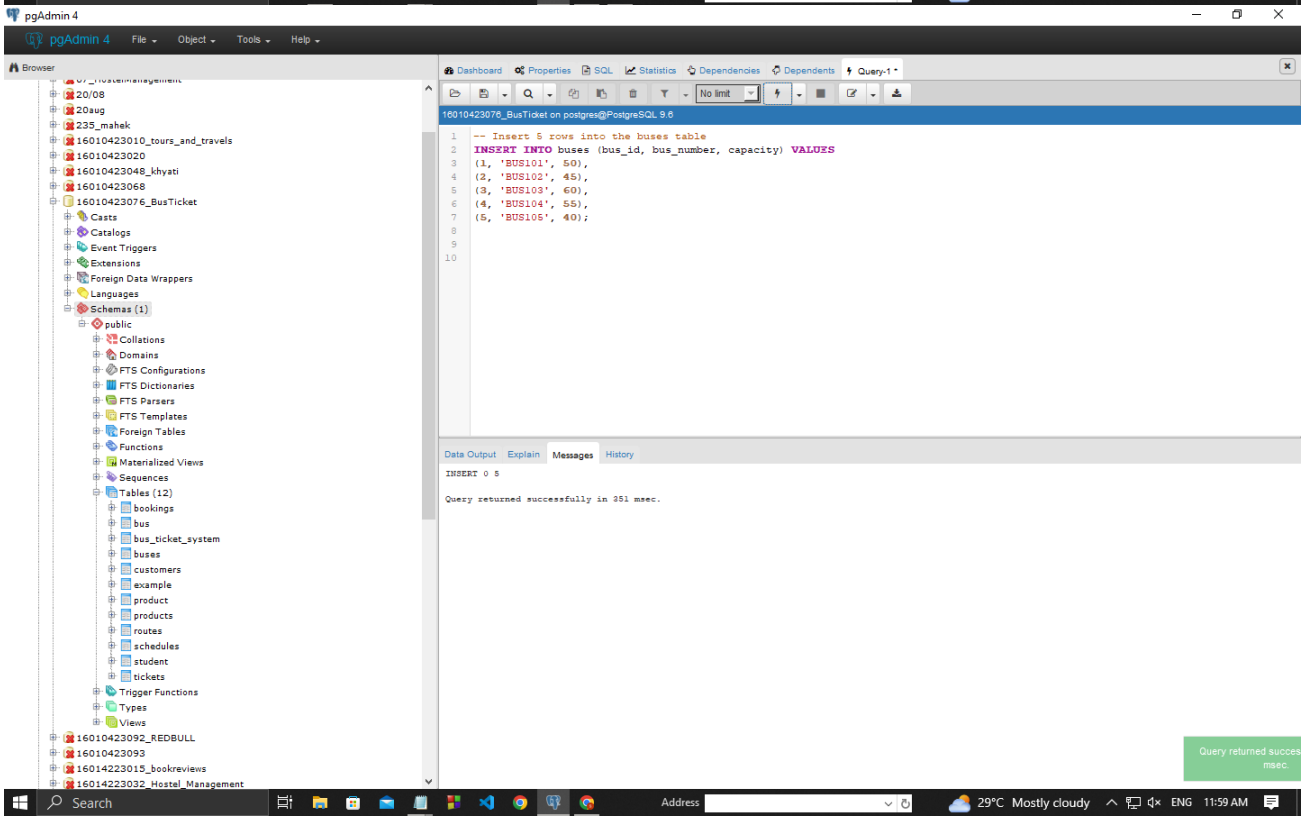
(5, 'City I', 'City J', 210);

**Output Screenshots :**


pgAdmin 4
File
Object
Tools
Help
Browser
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Query-1
No limit
16010423076_BusTicket on postgres@PostgreSQL 9.6
-- Insert 5 rows into the buses table
INSERT INTO buses (bus_id, bus_number, capacity) VALUES
(1, 'BUS101', 50),
(2, 'BUS102', 45),
(3, 'BUS103', 60),
(4, 'BUS104', 55),
(5, 'BUS105', 40);
-- Insert 5 rows into the routes table
INSERT INTO routes (route_id, origin, destination, distance) VALUES
(1, 'City A', 'City B', 120),
(2, 'City C', 'City D', 200),
(3, 'City E', 'City F', 150),
(4, 'City G', 'City H', 180),
(5, 'City I', 'City J', 210);
Data Output
Explain
Messages
History
INSERT 0 5
Query returned successfully in 317 msec.
Query returned successfully in 317 msec.
Search
Address
29°C Mostly cloudy
ENG
12:01 PM


pgAdmin 4
File
Object
Tools
Help
Browser
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Query-1
No limit
16010423076_BusTicket on postgres@PostgreSQL 9.6
-- Insert 5 rows into the buses table
INSERT INTO buses (bus_id, bus_number, capacity) VALUES
(1, 'BUS101', 50),
(2, 'BUS102', 45),
(3, 'BUS103', 60),
(4, 'BUS104', 55),
(5, 'BUS105', 40);
-- Insert 5 rows into the routes table
INSERT INTO routes (route_id, origin, destination, distance) VALUES
(1, 'City A', 'City B', 120),
(2, 'City C', 'City D', 200),
(3, 'City E', 'City F', 150),
(4, 'City G', 'City H', 180),
(5, 'City I', 'City J', 210);
SELECT * from buses
SELECT * from routes
Data Output
Explain
Messages
History
bus_id integer
bus_number character varying (20)
capacity integer
1 BUS101 50
2 BUS102 45
3 BUS103 60
4 BUS104 55
5 BUS105 40
Search
Address
29°C Mostly cloudy
ENG
12:03 PM


pgAdmin 4
File
Object
Tools
Help
Browser
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Query-1
No limit
16010423076_BusTicket on postgres@PostgreSQL 9.6
-- Insert 5 rows into the buses table
INSERT INTO buses (bus_id, bus_number, capacity) VALUES
(1, 'BUS101', 50),
(2, 'BUS102', 45),
(3, 'BUS103', 60),
(4, 'BUS104', 55),
(5, 'BUS105', 40);
-- Insert 5 rows into the routes table
INSERT INTO routes (route_id, origin, destination, distance) VALUES
(1, 'City A', 'City B', 120),
(2, 'City C', 'City D', 200),
(3, 'City E', 'City F', 150),
(4, 'City G', 'City H', 180),
(5, 'City I', 'City J', 210);
SELECT * from buses
SELECT * from routes
Data Output
Explain
Messages
History
route_id integer
origin character varying (100)
destination character varying (100)
distance integer
1 City A City B 120
2 City C City D 200
3 City E City F 150
4 City G City H 180
5 City I City J 210
Total query runtime: 398 msec.
5 rows retrieved.
Search
Address
29°C Mostly cloudy
ENG
12:03 PM


pgAdmin 4
File
Object
Tools
Help
Browser
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Query-1
No limit
16010423076_BusTicket on postgres@PostgreSQL 9.6
-- Insert 5 rows into the buses table
INSERT INTO buses (bus_id, bus_number, capacity) VALUES
(1, 'BUS101', 50),
(2, 'BUS102', 45),
(3, 'BUS103', 60),
(4, 'BUS104', 55),
(5, 'BUS105', 40);
Data Output
Explain
Messages
History
INSERT 0 5
Query returned successfully in 351 msec.
Search
Address
29°C Mostly cloudy
ENG
11:59 AM

**Checking for Constraint Violation error :**

-- 'BUS101' already exists.

INSERT INTO buses (bus_id, bus_number, capacity) VALUES

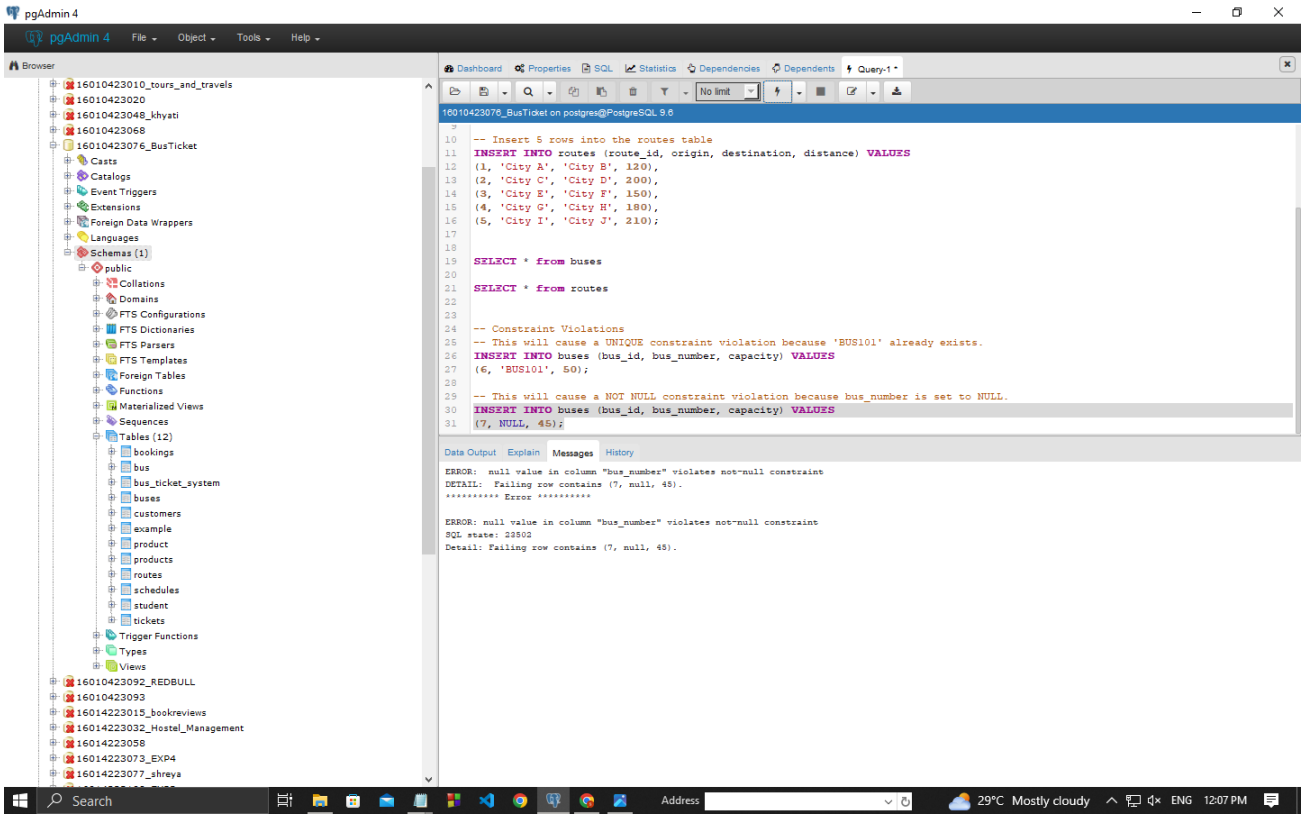
(6, 'BUS101', 50);

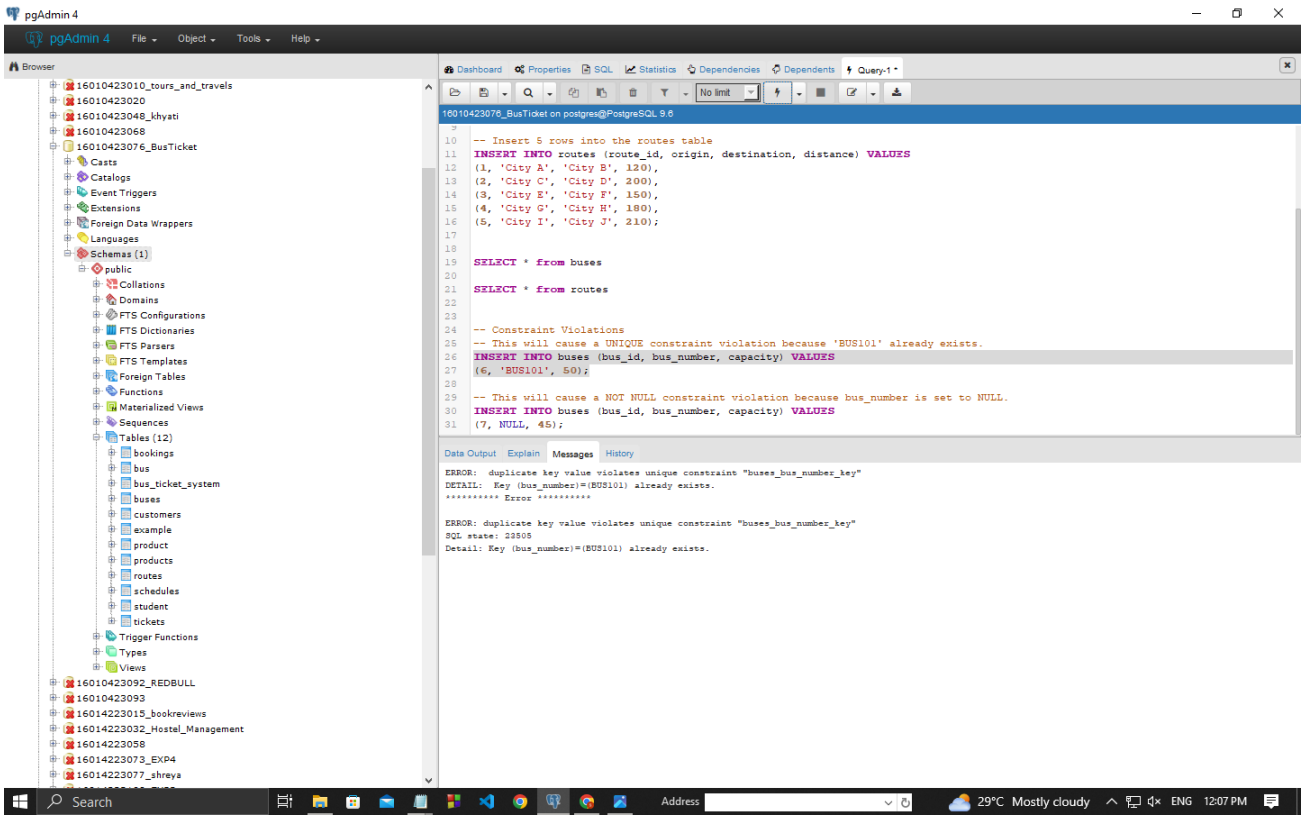
-- will cause a NOT NULL constraint violation because bus_number is set to NULL

INSERT INTO buses (bus_id, bus_number, capacity) VALUES

(7, NULL, 45);

**Output Screenshots :**






**Outcomes:**

CO1. Comprehend the features of relational database management systems.
CO2 Apply data models to real world scenarios.

**Questions:**

**Q1 what is difference between Truncate, Drop and delete? Explain with example**

TRUNCATE: This command removes all rows from a table, but it doesn't delete the table itself. It's like wiping the slate clean while keeping the slate. TRUNCATE is fast and efficient because it doesn't log individual row deletions, and it resets any auto-increment counters.

Example: If you have a table called Students with 1000 records, using TRUNCATE TABLE Students; will delete all records instantly, but the Students table will still exist, ready for new data.

● DROP: This command completely removes a table or database from the system. It's like throwing away the entire slate. Once you drop a table, it's gone permanently along with all its data and structure.

Example: If you use DROP TABLE Students;, the Students table and all its data are

permanently deleted from the database.

● DELETE: This command is used to remove specific rows from a table based on a condition. It's like erasing certain lines on the slate while keeping the rest intact. DELETE is slower than TRUNCATE because it logs each row deletion and allows you to specify exactly which rows to remove.

Example: If you want to delete only those students who are not enrolled anymore, you could use DELETE FROM Students WHERE enrolled = 'no';. This will remove only those rows where the condition is met, leaving the rest of the table intact.

---

**Conclusion:**

I successfully implemented a relational database model using DDL statements. By defining tables such as bus, routes, schedules, customers, bookings, and tickets, I applied key constraints like primary keys, foreign keys, unique constraints, and checks to ensure data integrity and consistency. I also practiced modifying tables by adding and removing columns and constraints.

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

**Signature of faculty in-charge with date**

---

**Reference books:**

1. Elmasri and Navathe, “Fundamentals of Database Systems”, 6th Edition, Pearson Education
2. Korth, Slberchatz,Sudarshan, :”Database System Concepts”, 6th Edition, McGraw – Hill.

**WebSite:**

1. http://www.tutorialspoint.com/postgresql/
2. http://sage.virtual-labs.ac.in/home/pub/21/