Practical 3

Shivam Tawari A-58

Aim: Project Table Identifications: Table Names, Table Attributes, Primary Keys, Foreign Keys, Integrity Constraints, Data Types, Referencing. Project Schema Creations: Formal joining of identified tables and checking of constraints

Theory:

SQL stands for Structured Query Language.

Why SQL:

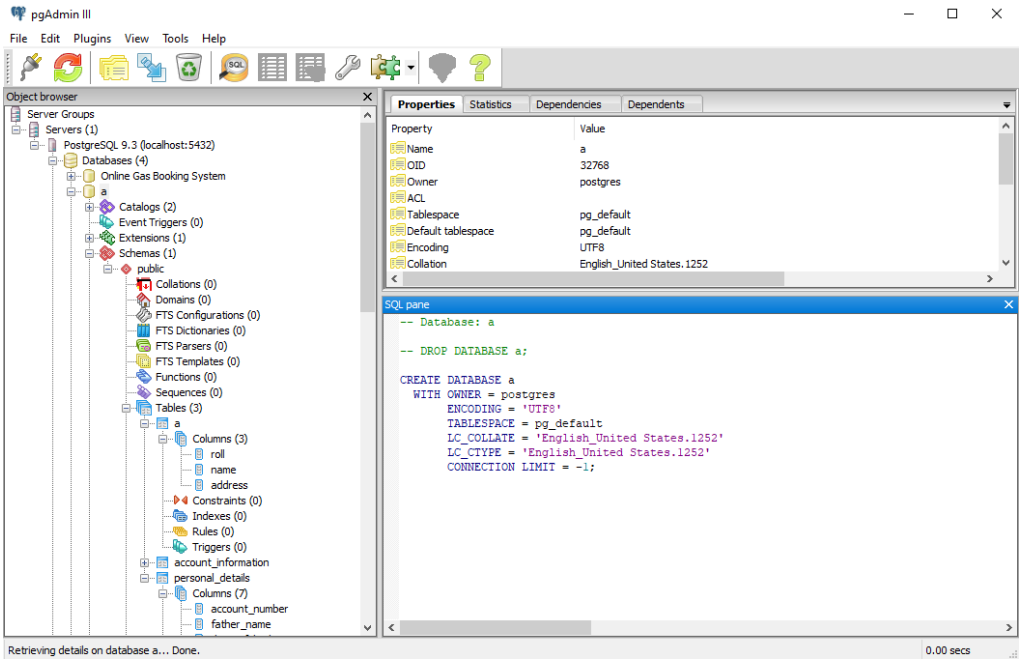
- Execute queries against a database
- Retrieve data from a database
- Insert records in a database
- Update records in a database
- Delete records from a database
- Create new databases
- Create new tables in a database
- Create stored procedures in a database
- Create views in a database

a) Create Database

It is used to create a new SQL database.

Syntax

CREATE DATABASE *databasename*;



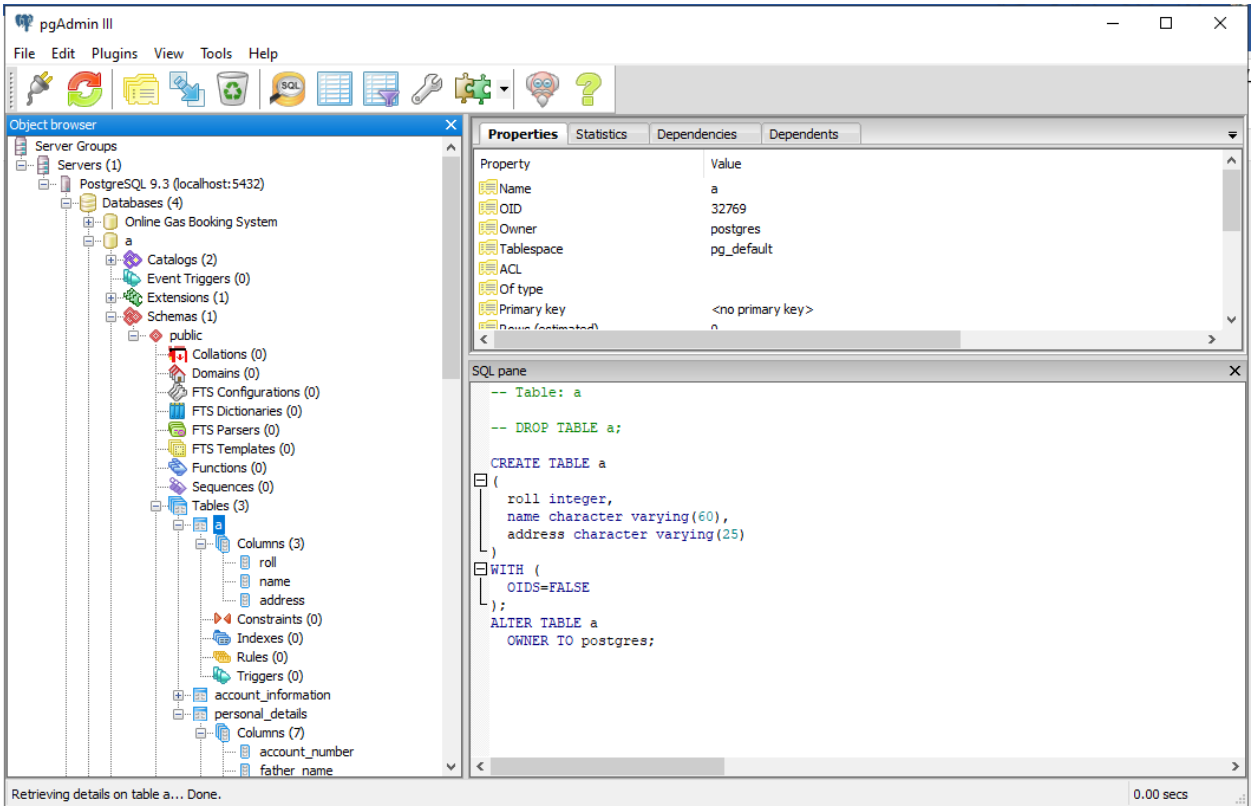
b) Create Table

Create Table (Without Constraints)

It is used to create a new table in a database.

Syntax:

CREATE TABLE table_name *(*
column1 datatype,
column2 datatype,
column3 datatype,
....
);



Create Table (With Constraints)

Constraints can be specified when the table is created with the CREATE TABLE statement, or after the table is created with the ALTER TABLE statement.

SQL constraints are used to specify rules for the data in a table.
Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.
Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.
The following constraints are commonly used in SQL:

- NOT NULL - Ensures that a column cannot have a NULL value
- UNIQUE - Ensures that all values in a column are different
- PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- FOREIGN KEY - Prevents actions that would destroy links between tables
- CHECK - Ensures that the values in a column satisfies a specific condition

Syntax:

CREATE TABLE sample_table

(

column1 data_type(size) constraint_name,

column2 data_type(size) constraint_name,

column3 data_type(size) constraint_name,

....

);

c) Inserting Record in table.

This statement is used to insert new records in a table.

Syntax:

INSERT INTO *table_name* (*column1*, *column2*, *column3*, ...)
VALUES (*value1*, *value2*, *value3*, ...);

or

INSERT INTO *table_name*
VALUES (*value1*, *value2*, *value3*, ...);

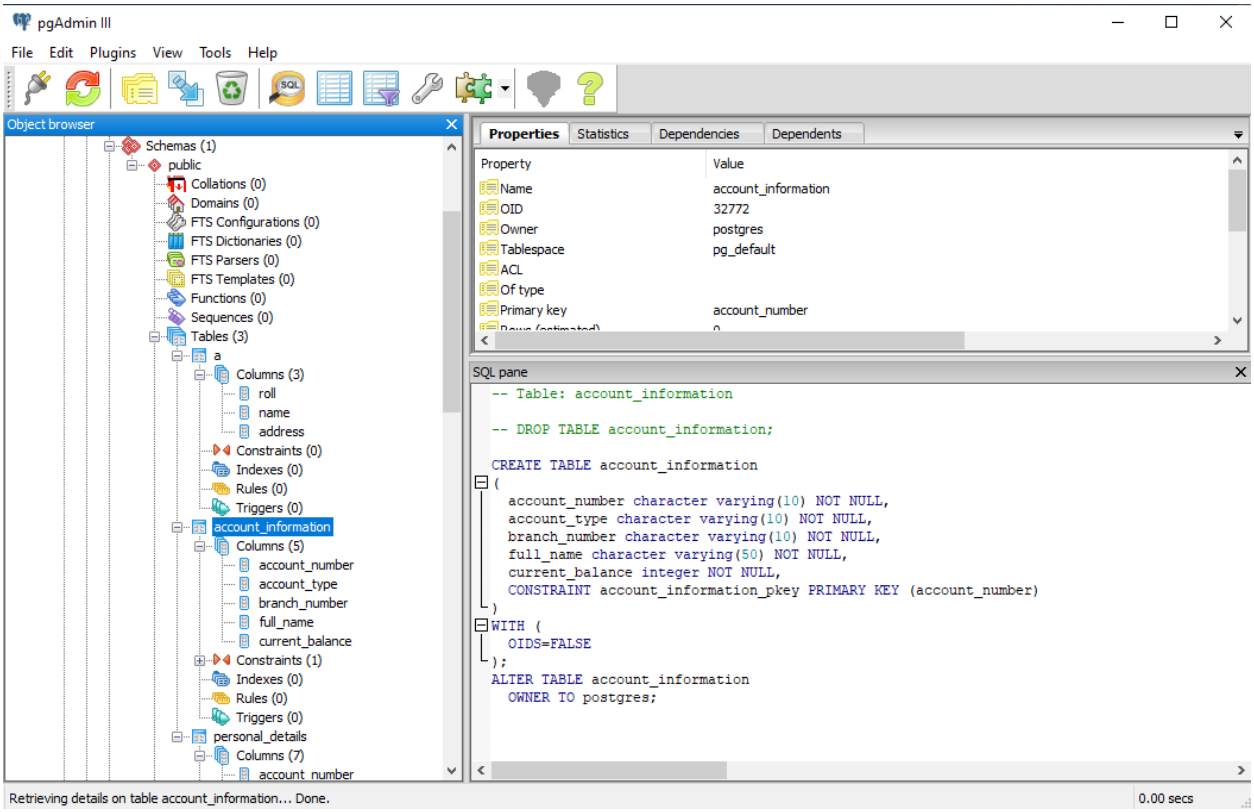
Program/Queries:

1. ***Creating a Database***

   CREATE DATABASE Company;

2. ***Creating Tables (Without Constraints)***

```
CREATE TABLE Persons (
    PersonID int,
    LastName varchar(255),
    FirstName varchar(255),
    Address varchar(255),
    City varchar(255)
);
```

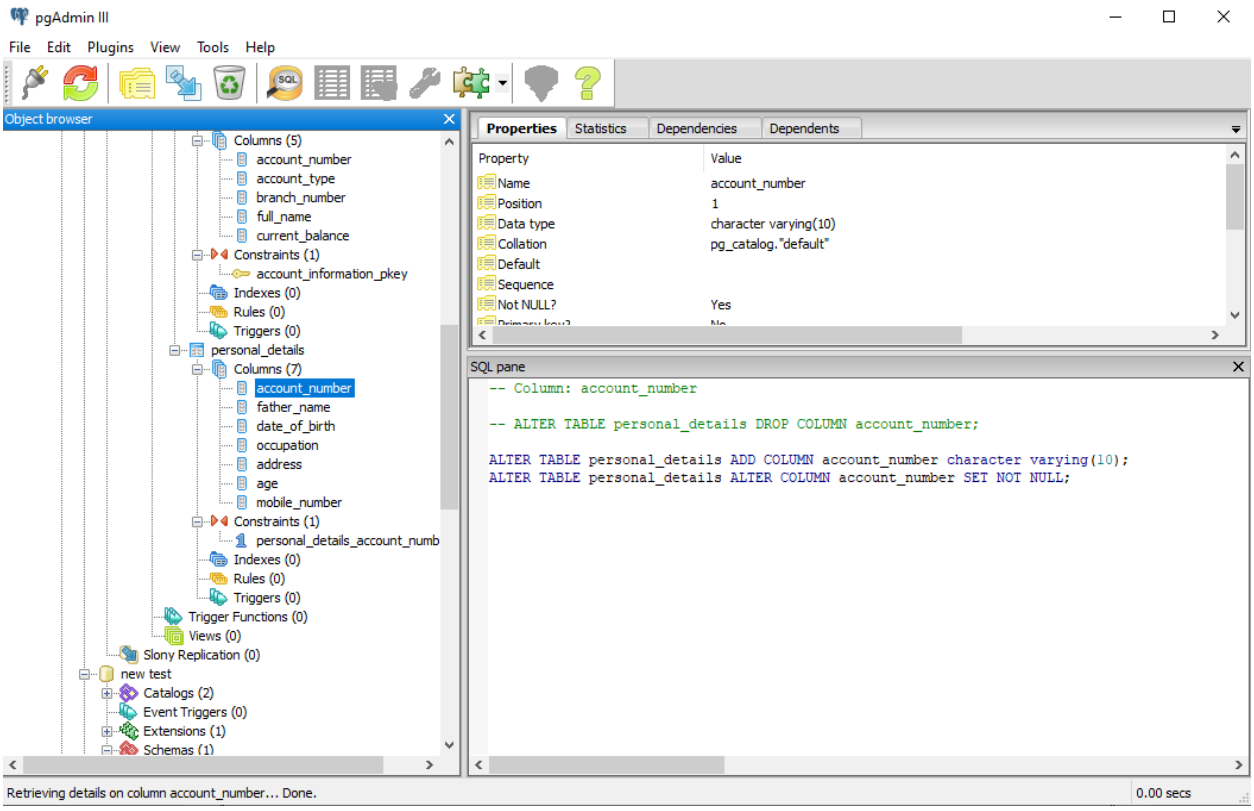


Creating Tables (With Constraints)

1. **NOT NULL –**

If we specify a field in a table to be NOT NULL then the field will never accept null value.

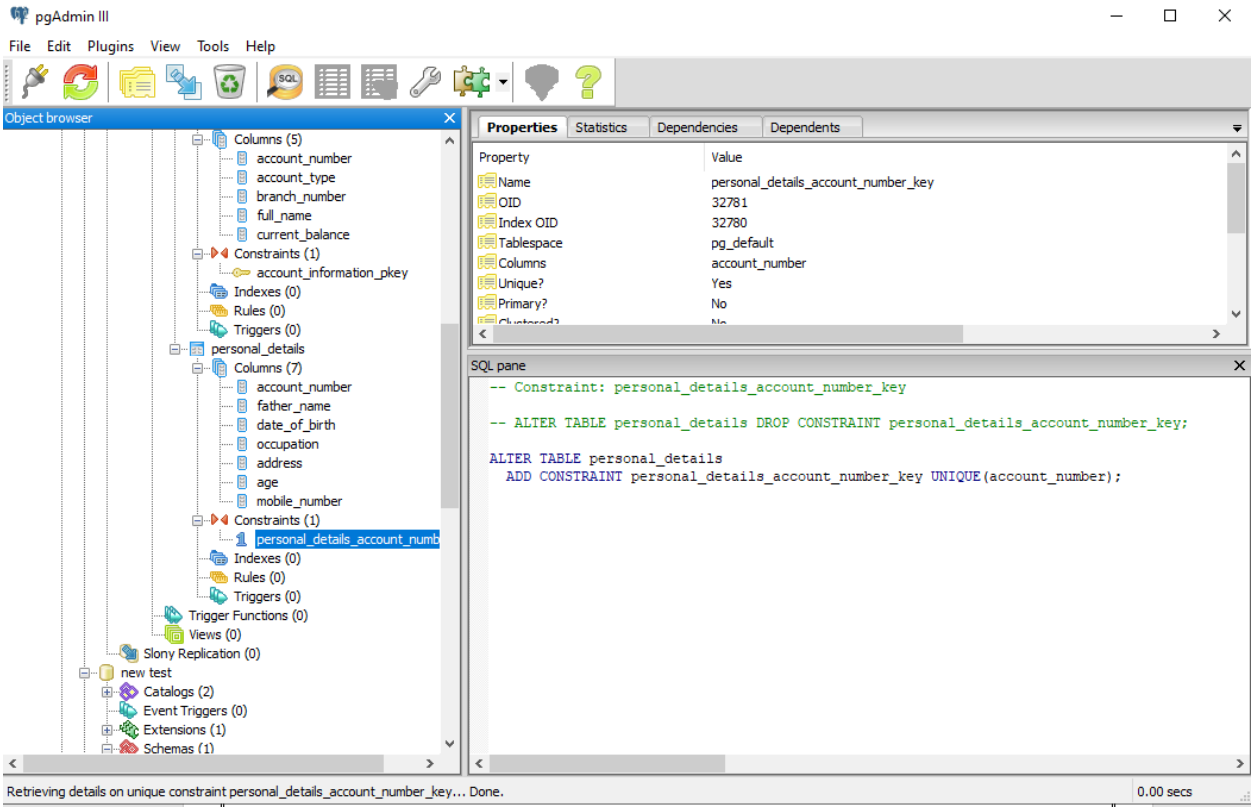
```
CREATE TABLE Persons (

    ID int NOT NULL,

    LastName varchar(255) NOT NULL,

    FirstName varchar(255) NOT NULL,

    Age int

);
```



2. UNIQUE –
This constraint helps to uniquely identify each row in the table. i.e. for a particular column, all the rows should have unique values.

For example, the below query creates a table Student where the field ID is specified as UNIQUE. i.e, no two students can have the same ID.

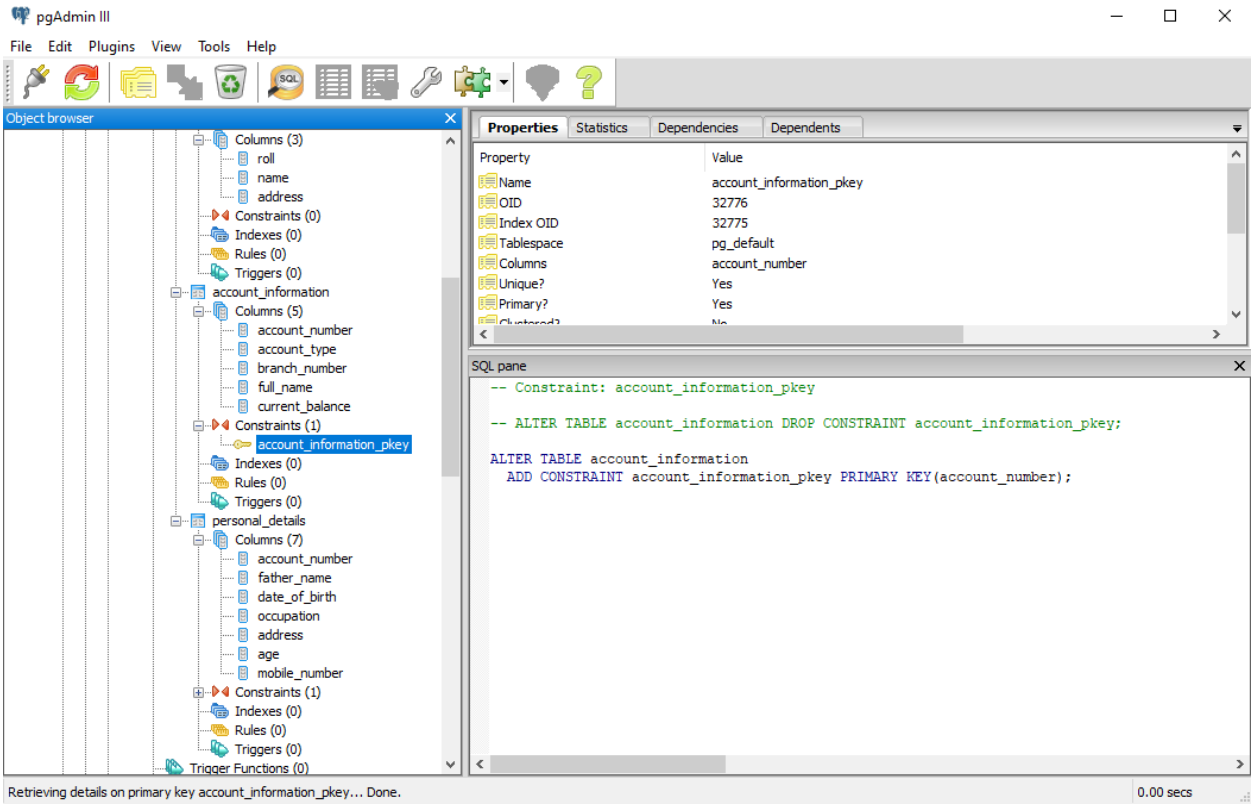
```
CREATE TABLE Student
(
ID int(6) NOT NULL UNIQUE,
NAME varchar(10),
ADDRESS varchar(20)
);
```



3. PRIMARY KEY –

Primary Key is a field which uniquely identifies each row in the table. If a field in a table as primary key, then the field will not be able to contain NULL values as well as all the rows should have unique values for this field. So, in other words we can say that this is combination of NOT NULL and UNIQUE constraints.
A table can have only one field as primary key. Below query will create a table named Student and specifies the field ID as primary key.

```
CREATE TABLE Student
(
ID int(6) NOT NULL UNIQUE,
NAME varchar(10),
ADDRESS varchar(20),
PRIMARY KEY(ID)
);
```



4. FOREIGN KEY –

Foreign Key is a field in a table which uniquely identifies each row of a another table. That is, this field points to primary key of another table. This usually creates a kind of link between the tables.

Consider the two tables as shown below:

Orders

O_ID	ORDER_NO	C_ID
1	2253	3
2	3325	3
3	4521	2
4	8532	1

Customers

C_ID	NAME	ADDRESS
1	RAMESH	DELHI
2	SURESH	NOIDA

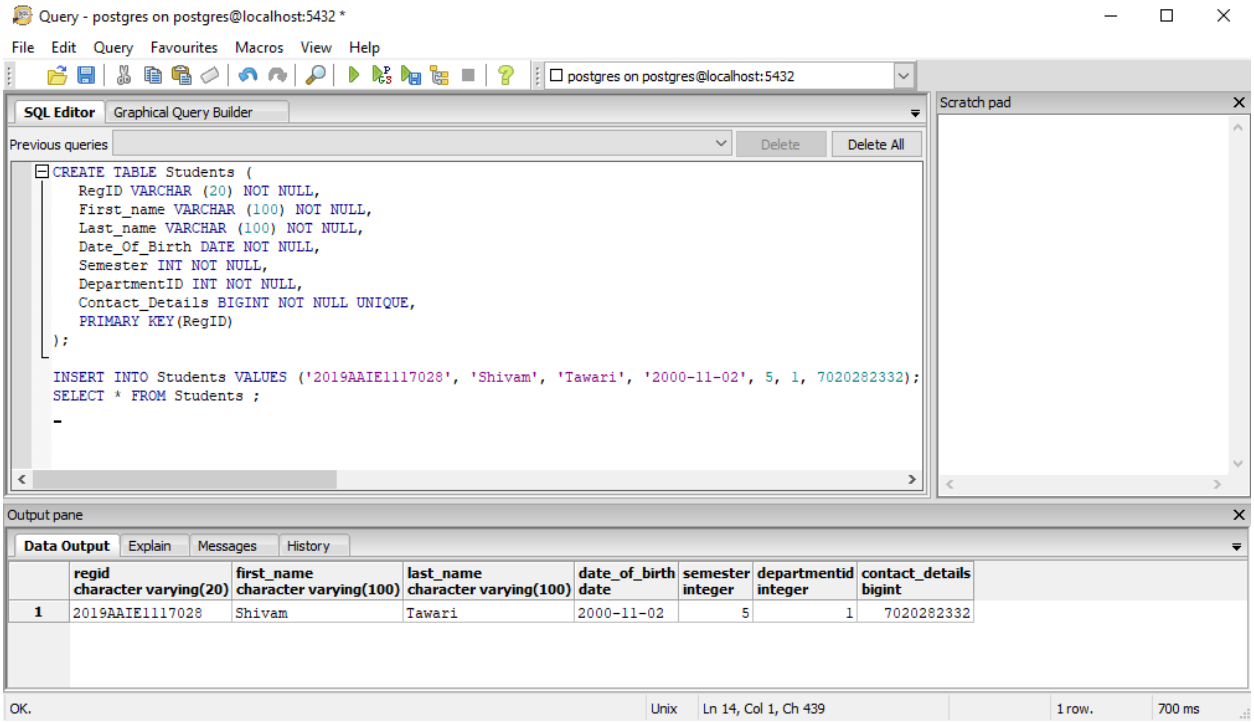
3	DHARMESH	GURGAON

We can see clearly that the field C_ID in Orders table is the primary key in Customers table, i.e. it uniquely identifies each row in the Customers table. Therefore, it is a Foreign Key in Orders table.
Syntax:

```
CREATE TABLE Orders
(
O_ID int NOT NULL,
ORDER_NO int NOT NULL,
C_ID int,
PRIMARY KEY (O_ID),
FOREIGN KEY (C_ID) REFERENCES Customers(C_ID)
)
```

3. Inserting Records in a Table,

INSERT INTO Students VALUES ('2019AAIE1117028', 'Shivam', 'Tawari', '2000-11-02', 5, 1, 7020282332);



Conclusion: Hence we have performed SQL query that demonstrated how to create database , table and insert records in the table.