

ASSIGNMENT 2

Question 1

STEP 1 – Creating a Database

```
1 create database assignment2
```

Data Output Messages Notifications

CREATE DATABASE

Query returned successfully in 216 msec.

STEP 2 – Creating the table

assignment2/postgres@Assignment2_Siddharth

No limit

Query Query History

```
1 -- Create the "World" table
2 CREATE TABLE World (
3     name VARCHAR(255) PRIMARY KEY,
4     continent VARCHAR(255),
5     area INT,
6     population INT,
7     gdp BIGINT
8 );
9
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 34 msec.

STEP 3 – Inserting the values in the tables

assignment2/postgres@Assignment2_Siddharth

Query Query History

```
8 );
9
10 -- Insert sample data into the "World" table
11 INSERT INTO World (name, continent, area, population, gdp)
12 VALUES
13     ('Afghanistan', 'Asia', 652230, 25500100, 20343000000),
14     ('Albania', 'Europe', 28748, 2831741, 12960000000),
15     ('Algeria', 'Africa', 2381741, 37100000, 188681000000),
16     ('Andorra', 'Europe', 468, 78115, 3712000000),
17     ('Angola', 'Africa', 1246700, 20609294, 100990000000);
18
```

Data Output Messages Notifications

INSERT 0 5

Query returned successfully in 35 msec.

STEP 4 – Displaying the table

assignment2/postgres@Assignment2_Siddharth

Query Query History

```
12 VALUES
13     ('Afghanistan', 'Asia', 652230, 25500100, 20343000000),
14     ('Albania', 'Europe', 28748, 2831741, 12960000000),
15     ('Algeria', 'Africa', 2381741, 37100000, 188681000000),
16     ('Andorra', 'Europe', 468, 78115, 3712000000),
17     ('Angola', 'Africa', 1246700, 20609294, 100990000000);
18
19 -- Checking the Table we created
20 SELECT *
21 FROM World
22
```

Data Output Messages Notifications

	name [PK] character varying (255)	continent character varying (255)	area integer	population integer	gdp bigint
1	Afghanistan	Asia	652230	25500100	20343000000
2	Albania	Europe	28748	2831741	12960000000
3	Algeria	Africa	2381741	37100000	188681000000
4	Andorra	Europe	468	78115	3712000000
5	Angola	Africa	1246700	20609294	100990000000

STEP 5 – OUTPUT

assignment2/postgres@Assignment2_Siddharth

Query Query History

```

18
19 -- Checking the Table we created
20 SELECT *
21 FROM World
22
23 -- Solving to get our output as given in the question
24 SELECT name, population, area
25 FROM World
26 WHERE area >= 3000000 OR population >= 25000000;
27
28

```

Data Output Messages Notifications

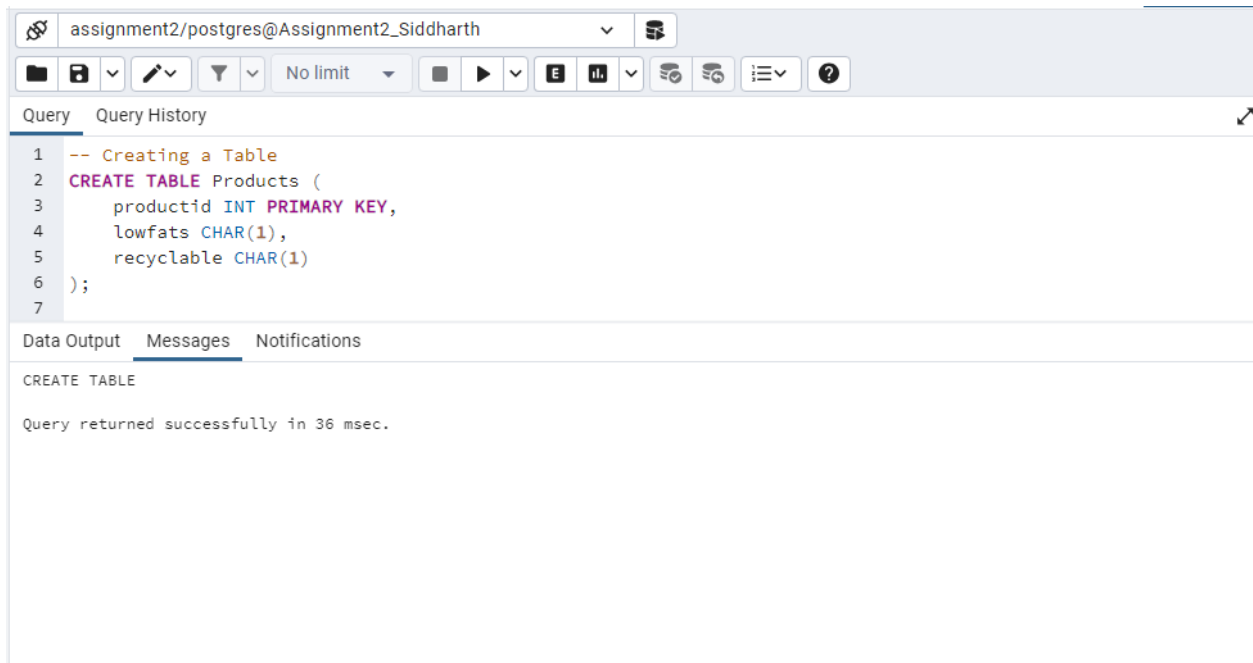
	name [PK] character varying (255)	population integer	area integer
1	Afghanistan	25500100	652230
2	Algeria	37100000	2381741

OUTPUT SNAPSHOT FOR QUESTION 1

	name [PK] character varying (255)	population integer	area integer
1	Afghanistan	25500100	652230
2	Algeria	37100000	2381741

QUESTION 2

Step 1 - Creating the table

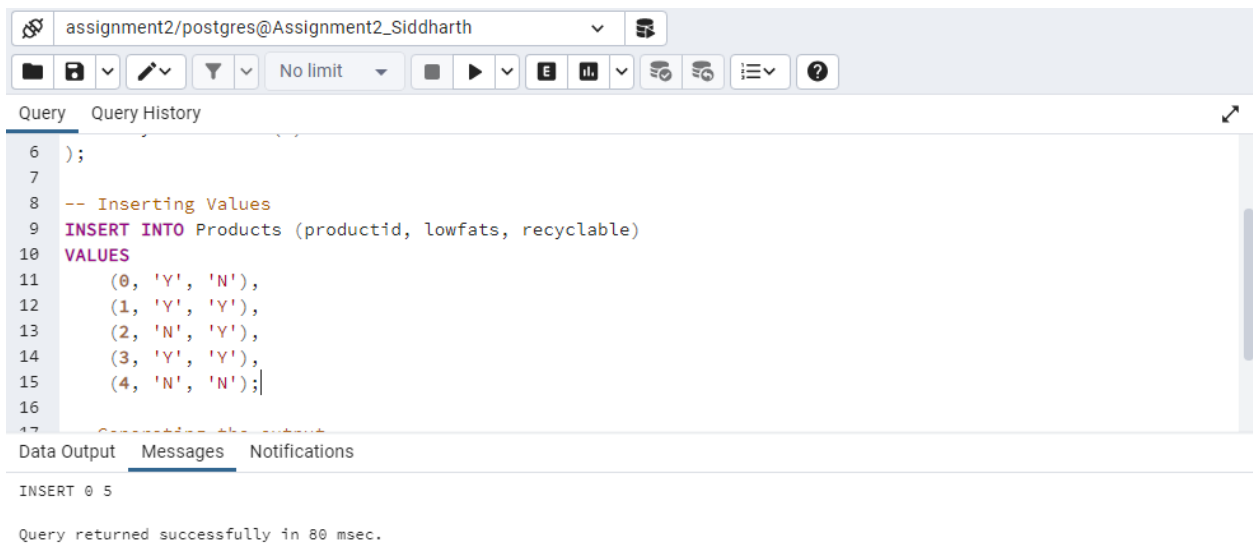


The screenshot shows a PostgreSQL query editor interface. At the top, the connection is set to 'assignment2/postgres@Assignment2_Siddharth'. Below the connection bar is a toolbar with icons for file operations, query execution, and settings. The 'Query' tab is active, displaying a SQL script to create a table named 'Products'. The script includes a comment '-- Creating a Table' and the following SQL code:
`CREATE TABLE Products (
 productid INT PRIMARY KEY,
 lowfats CHAR(1),
 recyclable CHAR(1)
);`
The 'Data Output' tab is selected, showing the message 'CREATE TABLE' and 'Query returned successfully in 36 msec.'

```
-- Creating a Table
CREATE TABLE Products (
    productid INT PRIMARY KEY,
    lowfats CHAR(1),
    recyclable CHAR(1)
);
```

CREATE TABLE
Query returned successfully in 36 msec.

Step 2 - Inserting values into the table



The screenshot shows the same PostgreSQL query editor interface. The 'Query' tab is active, displaying a SQL script to insert values into the 'Products' table. The script includes a comment '-- Inserting Values' and the following SQL code:
`INSERT INTO Products (productid, lowfats, recyclable)
VALUES
 (0, 'Y', 'N'),
 (1, 'Y', 'Y'),
 (2, 'N', 'Y'),
 (3, 'Y', 'Y'),
 (4, 'N', 'N');`
The 'Data Output' tab is selected, showing the message 'INSERT 0 5' and 'Query returned successfully in 80 msec.'

```
-- Inserting Values
INSERT INTO Products (productid, lowfats, recyclable)
VALUES
    (0, 'Y', 'N'),
    (1, 'Y', 'Y'),
    (2, 'N', 'Y'),
    (3, 'Y', 'Y'),
    (4, 'N', 'N');
```

INSERT 0 5
Query returned successfully in 80 msec.

Step 3 - Checking the table we created

assignment2/postgres@Assignment2_Siddharth

Query Query History

```

11  (0, 'Y', 'N'),
12  (1, 'Y', 'Y'),
13  (2, 'N', 'Y'),
14  (3, 'Y', 'Y'),
15  (4, 'N', 'N');
16
17  -- Checking the table we created
18  SELECT *
19  FROM Products
20

```

Data Output Messages Notifications

	productid [PK] integer	lowfats character	recyclable character
1	0	Y	N
2	1	Y	Y
3	2	N	Y
4	3	Y	Y
5	4	N	N

Step 4 - Generating the output

assignment2/postgres@Assignment2_Siddharth

Query Query History

```


16
17  -- Checking the table we created
18  SELECT *
19  FROM Products
20
21  -- Generating the output
22  SELECT productid
23  FROM Products
24  WHERE lowfats = 'Y' AND recyclable = 'Y';
25

```

Data Output Messages Notifications

	productid [PK] integer
1	1
2	3

OUTPUT SNAPSHOT FOR QUESTION 2

	productid [PK] integer 
1	1
2	3