

# **Subqueries and Views**

## **Project Title: Country and Customer Database**

### **Overview**

This project involves the creation and management of two primary tables: Country and Persons, along with a Customer table within a new database called Product. The goal is to manage demographic and customer information effectively, allowing for various data queries and operations.

### **Database Structure**

#### **1. Country Table**

- **Table Name:** Country
- **Fields:**
  - Id (INT, PRIMARY KEY): Unique identifier for each country.
  - Country\_name (VARCHAR(100)): Name of the country.
  - Population (INT): Total population of the country.
  - Area (INT): Total area of the country in square kilometers.

#### **2. Persons Table**

- **Table Name:** Persons
- **Fields:**
  - Id (INT, PRIMARY KEY): Unique identifier for each person.
  - Fname (VARCHAR(100)): First name of the person.
  - Lname (VARCHAR(100)): Last name of the person.
  - Population (INT): Population associated with the person (for context).
  - Rating (DECIMAL(3, 2)): Rating of the person (e.g., on a scale of 1 to 5).
  - Country\_Id (INT): Foreign key linking to the Country table.
  - Country\_name (VARCHAR(100)): Name of the country.

#### **3. Customer Table**

- **Database Name:** Product
- **Table Name:** Customer
- **Fields:**
  - Customer\_Id (INT, PRIMARY KEY): Unique identifier for each customer.
  - First\_name (VARCHAR(100)): Customer's first name.
  - Last\_name (VARCHAR(100)): Customer's last name.

- Email (VARCHAR(100)): Customer's email address.
- Phone\_no (VARCHAR(15)): Customer's phone number.
- Address (VARCHAR(255)): Customer's address.
- City (VARCHAR(100)): City of residence.
- State (VARCHAR(100)): State of residence.
- Zip\_code (VARCHAR(10)): Zip code of the address.
- Country (VARCHAR(100)): Country of residence.

## Views Created

1. **customer\_info**: Displays full name and email address of customers.
2. **US\_Customers**: Displays customers located in the United States.
3. **Customer\_details**: Provides a detailed view of customer information.

## SQL Operations

This project includes various SQL operations, such as:

- Counting persons by country.
- Calculating average ratings for persons.
- Updating customer phone numbers based on location.
- Filtering customers based on specific criteria.

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
1  ##Find the number of persons in each country:
2  • SELECT Country_name, COUNT(*) AS Number_of_Persons
3  FROM Persons
4  GROUP BY Country_name;
```

The results pane displays the output of the query in a table format:

Country_name	Number_of_Persons
USA	3
India	2
China	2
Brazil	1
Nigeria	1
Russia	1

Below the table, the 'Output' pane shows the execution details:

#	Time	Action	Message
1	11:51:43	SELECT Country_name, COUNT(*) AS Number_of_Persons FROM Persons GROUP BY Country_name	6 row(s) returned

Querying Data assignment sql\* SQL File 8\* SQL File 9 Sorting and Grouping data sql a... Joins and Union Functions SQL File 8\* SQL File 9\*

```

26 FROM Country
27 WHERE Population > (SELECT AVG(Population) FROM Country);
28
29 ##Create the Product Database and Customer Table
30 • CREATE DATABASE Product;
31
32 • USE Product;
33
34 • CREATE TABLE Customer1 (
35     Customer_Id INT PRIMARY KEY,
36     First_name VARCHAR(100),
37     Last_name VARCHAR(100),
38     Email VARCHAR(100),
39     Phone_no VARCHAR(15),
40     Address VARCHAR(255),
41     City VARCHAR(100),
42     State VARCHAR(100),
43     Zip_code VARCHAR(10),
44     Country VARCHAR(100)
45 );

```

Output

#	Time	Action	Message	Duration / Fetch
2	11:52:31	SELECT Country_name, COUNT(*) AS Number_of_Persons FROM Persons GROUP BY Country_name ORDER...	6 row(s) returned	0.000 sec / 0.000 sec
3	11:53:22	SELECT Country_name, AVG(Rating) AS Average_Rating FROM Persons GROUP BY Country_name HAVING ...	5 row(s) returned	0.000 sec / 0.000 sec
4	11:54:44	SELECT DISTINCT Country_name FROM Persons WHERE Rating = (SELECT AVG(Rating) FROM Persons W...	0 row(s) returned	0.000 sec / 0.000 sec
5	11:55:35	SELECT Country_name FROM Country WHERE Population > (SELECT AVG(Population) FROM Country)	2 row(s) returned	0.000 sec / 0.000 sec
6	11:56:25	CREATE TABLE Customer ( Customer_Id INT PRIMARY KEY, First_name VARCHAR(100), Last_name ...	Error Code: 1050. Table 'customer' already exists	0.000 sec
7	11:56:45	CREATE TABLE Customer1 ( Customer_Id INT PRIMARY KEY, First_name VARCHAR(100), Last_name...	0 row(s) affected	0.032 sec

3.

Querying Data assignment sql\* SQL File 8\* SQL File 9 Sorting and Grouping data sql a... Joins and Union Functions SQL File 8\* SQL File 9\*

```

44 Country VARCHAR(100)
45 );
46
47 ##Create a view named customer_info for the Customer table that displays Customer's full name and email address:
48 • CREATE VIEW customer_info AS
49 SELECT CONCAT(First_name, ' ', Last_name) AS Full_Name, Email
50 FROM Customer1;
51
52 • SELECT * FROM customer_info;
53
54 ##Create a view named US_Customers that displays customers located in the US:
55 • CREATE VIEW US_Customers AS
56 SELECT *
57 FROM Customer
58 WHERE Country = 'United States';
59
60 ##Create another view named Customer_details with specified columns:
61 • CREATE VIEW Customer_details AS
62 SELECT CONCAT(First_name, ' ', Last_name) AS Full_Name, Email, Phone_no, State
63 FROM Customer1;
64

```

Output

#	Time	Action	Message	Duration / Fetch
7	11:56:45	CREATE TABLE Customer1 ( Customer_Id INT PRIMARY KEY, First_name VARCHAR(100), Last_name...	0 row(s) affected	
8	11:59:04	SELECT * FROM customer_info	Error Code: 1146. Table 'sales.customer_info' doesn't exist	
9	12:00:18	CREATE VIEW customer_info AS SELECT CONCAT(First_name, ' ', Last_name) AS Full_Name, Email FROM Cu...	0 row(s) affected	
10	12:00:23	SELECT * FROM customer_info	0 row(s) returned	
11	12:01:19	CREATE VIEW US_Customers AS SELECT * FROM Customer WHERE Country = 'United States'	0 row(s) affected	
12	12:02:09	CREATE VIEW Customer_details AS SELECT CONCAT(First_name, ' ', Last_name) AS Full_Name, Email, Phon...	0 row(s) affected	

4.

Querying Data assignment.sql SQL File 8 SQL File 9 Sorting and Grouping data.sql a... Joins and Union Functions SQL File 8 SQL File 9

Don't Limit

```

51
52 • SELECT * FROM customer_info;
53
54 ##Create a view named US_Customers that displays customers located in the US:
55 • CREATE VIEW US_Customers AS
56 SELECT *
57 FROM Customer
58 WHERE Country = 'United States';
59
60 ##Create another view named Customer_details with specified columns:
61 • CREATE VIEW Customer_details AS
62 SELECT CONCAT(First_name, ' ', Last_name) AS Full_Name, Email, Phone_no, State
63 FROM Customer;
64
65 ##Update phone numbers of customers who live in California for Customer_details view:
66 • SET SQL_SAFE_UPDATES = 0;
67 • UPDATE Customer
68 SET Phone_no = '123-456-7890' -- Example new phone number
69 WHERE State = 'California';
70

```

Output

#	Time	Action	Message	Duration / Fe
10	12:00:23	SELECT * FROM customer_info	0 row(s) returned	0.000 sec / 0
11	12:01:19	CREATE VIEW US_Customers AS SELECT * FROM Customer WHERE Country = 'United States'	0 row(s) affected	0.016 sec
12	12:02:09	CREATE VIEW Customer_details AS SELECT CONCAT(First_name, ' ', Last_name) AS Full_Name, Email, Phon...	0 row(s) affected	0.016 sec
13	12:03:07	UPDATE Customer SET Phone_no = '123-456-7890' -- Example new phone number WHERE State = 'California'	Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses ...	0.000 sec
14	12:04:07	SET SQL_SAFE_UPDATES = 0	0 row(s) affected	0.000 sec
15	12:04:11	UPDATE Customer SET Phone_no = '123-456-7890' -- Example new phone number WHERE State = 'California'	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec

5.

Querying Data assignment.sql SQL File 8 SQL File 9 Sorting and Grouping data.sql a... Joins and Union Functions SQL File 8 SQL File 9

Don't Limit

```

69 WHERE State = 'California';
70
71 ##Count the number of customers in each state and show only states with more than 5 customers:
72 • SELECT State, COUNT(*) AS Customer_Count
73 FROM Customer
74 GROUP BY State
75 HAVING COUNT(*) > 5;
76

```

Result Grid

State	Customer_Count
-------	----------------

Result 7

Output

#	Time	Action	Message
11	12:01:19	CREATE VIEW US_Customers AS SELECT * FROM Customer WHERE Country = 'United States'	0 row(s) affected
12	12:02:09	CREATE VIEW Customer_details AS SELECT CONCAT(First_name, ' ', Last_name) AS Full_Name, Email, Phon...	0 row(s) affected
13	12:03:07	UPDATE Customer SET Phone_no = '123-456-7890' -- Example new phone number WHERE State = 'California'	Error Code: 1175. You are using safe update mode and you tried to update a table without a W
14	12:04:07	SET SQL_SAFE_UPDATES = 0	0 row(s) affected
15	12:04:11	UPDATE Customer SET Phone_no = '123-456-7890' -- Example new phone number WHERE State = 'California'	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0
16	12:07:22	SELECT State, COUNT(*) AS Customer_Count FROM Customer GROUP BY State HAVING COUNT(*) > 5	0 row(s) returned

6.

SQL File 8\* SQL File 9 Sorting and Grouping data sql a... Joins and Union Functions SQL File 8\* SQL File 9\*

FROM Customer\_details  
GROUP BY State;

##Return all columns from the customer\_details view, sorted by the "state" column in ascending order:

SELECT \*  
FROM Customer\_details  
ORDER BY State ASC;

Full Name Email Phone\_no State

#	Time	Action	Message
13	12:03:07	UPDATE Customer SET Phone_no = '123-456-7890' -- Example new phone number WHERE State = 'California'	Error Code: 1175. You are using safe update mode and you tried to update a table without a
14	12:04:07	SET SQL_SAFE_UPDATES = 0	0 row(s) affected
15	12:04:11	UPDATE Customer SET Phone_no = '123-456-7890' -- Example new phone number WHERE State = 'California'	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0
16	12:07:22	SELECT State, COUNT(*) AS Customer_Count FROM Customer GROUP BY State HAVING COUNT(*) > 5	0 row(s) returned
17	12:08:33	SELECT State, COUNT(*) AS Customer_Count FROM Customer_details GROUP BY State	0 row(s) returned
18	12:10:05	SELECT * FROM Customer_details ORDER BY State ASC	0 row(s) returned