1. Find the number of persons in each country.

```
SELECT Country_name, COUNT(*) AS NumberOfPersons
FROM Persons
GROUP BY Country_name;
```

Country_name	NumberOfPersons
USA	2
Canada	2
Brazil	1
Germany	1
France	1
India	1
Australia	1
Mexico	1
Russia	1
China	1

2. Find the number of persons in each country sorted from high to low.

```
-- Find the number of persons in each country sorted from high to low.
SELECT Country_name, Count(*) AS Numberofpersons
FROM Persons
GROUP BY Country_name
ORDER BY Numberofpersons DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Country_name	Numberofpersons
USA	2
Canada	2
Brazil	1
Germany	1
France	1
India	1
Australia	1
Mexico	1
Russia	1
China	1

3. Find out an average rating for Persons in respective countries if the average is greater than 3.0

```
-- Find out an average rating for Persons in respective countries if the average is greater than 3.0
SELECT Country_name, AVG(Rating) AS AverageRating
FROM Persons
GROUP BY Country_name
HAVING AVG(Rating) > 3.0;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Country_name	AverageRating
USA	3.900000
Canada	3.350000
Brazil	4.200000
Germany	3.500000
France	4.000000
Australia	3.600000
Mexico	3.900000
Russia	4.700000
China	4.100000

4. Find the countries with the same rating as the USA. (Use Subqueries)

```
-- Find the countries with the same rating as the USA. (Use Subqueries)
SELECT DISTINCT Country_name
FROM Persons
WHERE Rating = (SELECT AVG(Rating) FROM Persons WHERE Country_name = 'USA');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Country_name
Mexico

5. Select all countries whose population is greater than the average population of all nations.

```
SELECT Country_name
FROM Country
WHERE Population > (SELECT AVG(Population) FROM Country);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

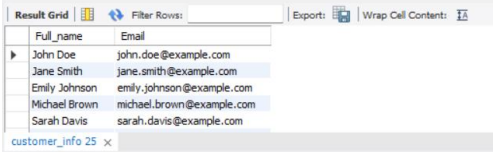
Country_name
India
China

6. Create a database named Product and create a table called Customer

```
-- Create a database product
CREATE DATABASE Product;

USE Product;

CREATE TABLE Customer (
    Customer_Id INT PRIMARY KEY,
    First_name VARCHAR(50),
    Last_name VARCHAR(50),
    Email VARCHAR(100),
    Phone_no VARCHAR(15),
    Address VARCHAR(255),
    City VARCHAR(100),
    State VARCHAR(50),
    Zip_code VARCHAR(10),
    Country VARCHAR(50)
);
```

7. Create a view named customer_info for the Customer table that displays Customer's Full name and email address. Then perform the SELECT operation for the customer_info view.

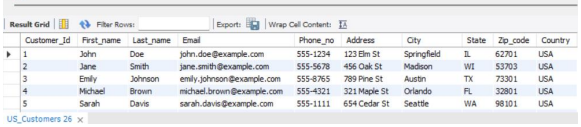
```
CREATE VIEW customer_info AS
SELECT CONCAT(First_name, ' ', Last_name) AS Full_name, Email
FROM Customer;

-- Perform the SELECT operation for the customer_info view
SELECT * FROM customer_info;
```



Full_name	Email
John Doe	john.doe@example.com
Jane Smith	jane.smith@example.com
Emily Johnson	emily.johnson@example.com
Michael Brown	michael.brown@example.com
Sarah Davis	sarah.davis@example.com

8. Create a view named US_Customers that displays customers located in the US.

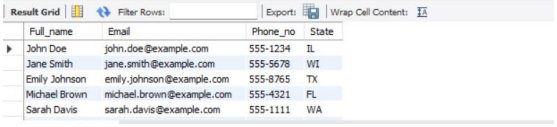
```
-- Create a view named US_Customers that displays customers located in the US.
CREATE VIEW US_Customers AS
SELECT * FROM customer WHERE Country = 'USA' ;

SELECT * FROM US_Customers;
```



Customer_Id	First_name	Last_name	Email	Phone_no	Address	City	State	Zip_code	Country
1	John	Doe	john.doe@example.com	555-1234	123 Elm St	Springfield	IL	62701	USA
2	Jane	Smith	jane.smith@example.com	555-5678	456 Oak St	Madison	WI	53703	USA
3	Emily	Johnson	emily.johnson@example.com	555-8765	789 Pine St	Austin	TX	73301	USA
4	Michael	Brown	michael.brown@example.com	555-4321	321 Maple St	Orlando	FL	32801	USA
5	Sarah	Davis	sarah.davis@example.com	555-1111	654 Cedar St	Seattle	WA	98101	USA

9. Create another view named Customer_details with columns full name(Combine first_name and last_name), email, phone_no, and state.

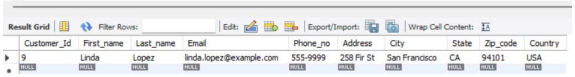
```
-- create another view named Customer_details with columns full name(Combine f
CREATE VIEW Customer_details AS
SELECT CONCAT(First_name, ' ', Last_name) AS Full_name, Email, Phone_no, State
FROM Customer;

SELECT * FROM Customer_details;
```



Full_name	Email	Phone_no	State
John Doe	john.doe@example.com	555-1234	IL
Jane Smith	jane.smith@example.com	555-5678	WI
Emily Johnson	emily.johnson@example.com	555-8765	TX
Michael Brown	michael.brown@example.com	555-4321	FL
Sarah Davis	sarah.davis@example.com	555-1111	WA

10. Update phone numbers of customers who live in California for Customer_details view.

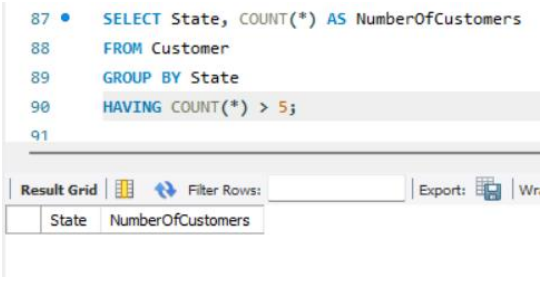
```
UPDATE Customer
SET Phone_no = '555-9999'  -- Replace with actual new phone number
WHERE State = 'CA';
SELECT * FROM Customer WHERE State = 'CA'
SET SQL_SAFE_UPDATES = 0;
```



Customer_Id	First_name	Last_name	Email	Phone_no	Address	City	State	Zip_code	Country
9	Linda	Lopez	linda.lopez@example.com	555-9999	258 Fir St	San Francisco	CA	94101	USA
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

11. Count the number of customers in each state and show only states with more than 5 customers.

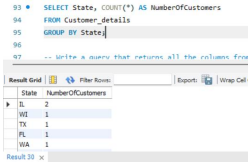
```
SELECT State, COUNT(*) AS NumberOfCustomers
FROM Customer
GROUP BY State
HAVING COUNT(*) > 5;
```



State	NumberOfCustomers

12. Write a query that will return the number of customers in each state, based on the "state" column in the "customer_details" view

```
SELECT State, COUNT(*) AS NumberOfCustomers
FROM Customer_details
GROUP BY State;
```



State	NumberOfCustomers
IL	2
WI	1
TX	1
FL	1
WA	1

13. Write a query that returns all the columns from the "customer_details" view, sorted by the "state" column in ascending order.

```
98 •  SELECT *
99    FROM Customer_details
100   ORDER BY State ASC;
101
102
```

Result Grid | Filter Rows: | Export: | Wrap Cell C

Full_name	Email	Phone_no	State
Laura Garcia	laura.garcia@example.com	555-3333	AZ
Linda Lopez	Laura Garcia ample.com	555-9999	CA
David Martinez	david.martinez@example.com	555-2222	CO
Michael Brown	michael.brown@example.com	555-4321	FL
John Doe	john.doe@example.com	555-1234	IL

Customer_details 31 ×