From the above given tables perform the following queries:

Part – A

1. Create a view that displays the top 5 countries with the highest population, along with their population figures.
   CREATE VIEW top5country AS
   SELECT TOP 5
           Country_Name,
           Population
   FROM Country
   ORDER BY Population DESC;

2. Create a view that lists countries that do not have any states.
   CREATE VIEW List_Country_not_state AS
   select Country.Country_ID,Country.Country_Name
   FROM Country
   INNER JOIN State
   on Country.Country_ID = State.Country_ID
   WHERE State.State_ID IS NULL

3. Create a view that displays the state with the highest population for each country, along with its population figure.
   ALTER VIEW List_Country_not_state AS
   SELECT
           Country.Country_ID,
           Country.Country_Name
   FROM Country
   INNER JOIN State
   on Country.Country_ID = State.Country_ID
   where State.State_ID IS NULL

4. Create a view that lists states that do not have a designated capital.
   CREATE VIEW StatesWithoutCapital AS
   SELECT
           State_ID,
           State_Name,
           Population,
           Area_sq_km,
           Country_ID
   FROM State
   WHERE Capital IS NULL OR Capital = '';

5. Create a view that displays countries with more than one capital city.
   CREATE VIEW country_with_morethan_one_city

```
AS
SELECT Country_Name,Country_ID,COUNT(Capital) AS total
FROM Country
GROUP BY Country_Name,Country_ID
HAVING COUNT(Capital) > 1
```

Part – B

6. Create a view AllOrdersView to Get All Orders with customer name.
```
CREATE VIEW ge_all_order AS
SELECT
        Customer.FirstName,
        Customer.LastName,
        Orders.OrderID,
        Orders.OrderDate,
        Orders.TotalAmount
FROM Customer INNER JOIN Orders
ON Customer.CustomerID = Orders.CustomerID
```

7. Create a view to Get Customers with No Email Addresses.
```
CREATE VIEW get_customer_no_email
AS
SELECT * FROM Customer
WHERE Email IS NULL
```

8. Create a view to return sum of total amount of order as total_amount.
```
CREATE VIEW total_amount_order AS
SELECT SUM(TotalAmount) AS total_amount
FROM Orders;
```

9. Create a view to Get Customers with Their Total Order Amount.
```
CREATE VIEW customer_with_their_total_amount
AS
SELECT
        Customer.FirstName,
        Customer.LastName,
        Sum(Orders.TotalAmount) as total_amount
FROM Customer INNER JOIN Orders on
Customer.CustomerID = Orders.CustomerID
GROUP BY Customer.FirstName,Customer.LastName
```

10. Create a view to Get Customers with Their Latest Order Date.

```
CREATE VIEW Customer_with_latest_date as
SELECT
        Customer.FirstName,
        Customer.Email,
        Customer.LastName,
```

```
        Customer.Phone,
        Orders.OrderDate
FROM Customer INNER JOIN Orders
on Customer.CustomerID = Orders.CustomerID
```

Part – C

11. Create a view to Get Customers with No Orders.

```
CREATE VIEW Customer_no_order AS
SELECT
        Customer.CustomerID,
        Customer.FirstName,
        Customer.LastName,
        Customer.Email,
        Customer.Phone,
        Orders.OrderID
FROM Customer LEFT JOIN Orders
on Customer.CustomerID = Orders.CustomerID
WHERE Orders.OrderID IS NULL
```

12. Create a view to Get Customers with Their Total Number of Orders.

```
CREATE VIEW customer_with_total_order AS
SELECT
        Customer.CustomerID,
        Customer.FirstName,
        Customer.LastName,
        Customer.Email,
        Customer.Phone,
count(Orders.OrderID) as totalorder
FROM Customer LEFT JOIN Orders
on Customer.CustomerID = Orders.CustomerID
GROUP BY Customer.CustomerID,Customer.FirstName,Customer.LastName
```

13. Create a view to Get Customers with High-Value Orders.

```
CREATE VIEW Customer_higest_order AS
SELECT
        Customer.CustomerID,
        Customer.FirstName,
        Customer.LastName,
        Customer.Email,
        Customer.Phone
FROM Customer LEFT JOIN Orders
on Customer.CustomerID = Orders.CustomerID
WHERE Orders.TotalAmount >200
```

14. Getting Customers with more than 1 order Placed.

```
CREATE VIEW CustomersWithMoreThanOneOrder AS
SELECT
        Customer.CustomerID,
        Customer.FirstName,
        Customer.LastName,
        Customer.Email,
        Customer.Phone,
        COUNT(Orders.OrderID) AS NumberOfOrders
FROM  Customer INNER JOIN Orders
ON Customer.CustomerID = Orders.CustomerID
GROUP BY Customer.CustomerID, Customer.FirstName, Customer.LastName
HAVING  COUNT(Orders.OrderID) > 1;
```

15. Getting Customers with Orders in Date Range 2023-07-01 to 2023-07-04.

```
CREATE VIEW CustomersWithOrdersInDateRange AS
SELECT
        Customer.CustomerID,
        Customer.FirstName,
        Customer.LastName,
        Customer.Email,
        Customer.Phone,
        Orders.OrderID,
        CONVERT(VARCHAR, Orders.OrderDate, 120) AS OrderDate,
        -- Convert DATE to VARCHAR
        Orders.TotalAmount
FROM Customer INNER JOIN Orders
ON Customer.CustomerID = Orders.CustomerID
WHERE Orders.OrderDate BETWEEN '2023-07-01' AND '2023-07-04';
```