Write an application using HBase and HiveQL for OnlineRetail Dataset which will include

i. Create and Load table with Online Retail data in Hive.

j. Create Index on Online Retail Table in Hive.

k. Find the total, average sales in Hive

l. Find Order details with maximum cost.

m. Find Customer details with maximum order total.

n. Find the Country with maximum and minimum sale.

o. Creating an external Hive table to connect to the HBase for OnlineRetail.

p. Display records of OnlineRetail Table in Hbase.

Step 1: Environment Setup

Before you begin, ensure that you have the following:

1. **Hadoop, Hive, and HBase installed** and running on your system.
2. The **OnlineRetail dataset** (usually in CSV format).

Step 2: Create Hive Table and Load Data

a. Open Hive Shell

To start, open the Hive shell or Beeline:

bash

CopyEdit

hive

b. Create the Hive Table

Create the OnlineRetail table in Hive with appropriate columns:

sql

CopyEdit

```
CREATE TABLE OnlineRetail (
  InvoiceNo STRING,
  StockCode STRING,
```

Description STRING,

Quantity INT,

InvoiceDate STRING,

UnitPrice DOUBLE,

CustomerID INT,

Country STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\n'

STORED AS TEXTFILE;

This command creates a table with fields like InvoiceNo, StockCode, Description, etc.

c. Load Data into Hive Table

Next, load the data into the OnlineRetail table from a local CSV file (replace /path/to/OnlineRetail.csv with the actual path to your CSV file):

sql

CopyEdit

LOAD DATA LOCAL INPATH '/path/to/OnlineRetail.csv' INTO TABLE OnlineRetail;

This command loads the CSV file into the table.

Step 3: Create Index on the Table

Indexes can improve query performance for columns that are frequently used in WHERE conditions.

a. Create Index on Country Column

sql

CopyEdit

CREATE INDEX idx_country

ON TABLE OnlineRetail (Country)

AS 'COMPACT' WITH DEFERRED REBUILD;

This command creates an index on the Country column of the OnlineRetail table to optimize queries that filter by Country.

Step 4: Perform Queries on the OnlineRetail Data

a. Find Total Sales

To calculate the **total sales**, which is the sum of Quantity * UnitPrice:

sql

CopyEdit

SELECT SUM(Quantity * UnitPrice) AS TotalSales FROM OnlineRetail;

b. Find Average Sales

To calculate the **average sales**:

sql

CopyEdit

SELECT AVG(Quantity * UnitPrice) AS AverageSales FROM OnlineRetail;

c. Find Order Details with Maximum Cost

To find the order (InvoiceNo) with the **maximum cost**:

sql

CopyEdit

SELECT InvoiceNo, SUM(Quantity * UnitPrice) AS TotalCost

FROM OnlineRetail

GROUP BY InvoiceNo

ORDER BY TotalCost DESC

LIMIT 1;

d. Find Customer Details with Maximum Order Total

To find the **customer** (CustomerID) with the maximum total order value:

sql

CopyEdit

```
SELECT CustomerID, SUM(Quantity * UnitPrice) AS TotalOrderValue
FROM OnlineRetail
GROUP BY CustomerID
ORDER BY TotalOrderValue DESC
LIMIT 1;
```

e. Find Country with Maximum Sales

To find the **country** with the maximum sales:

sql

CopyEdit

```
SELECT Country, SUM(Quantity * UnitPrice) AS TotalSales
FROM OnlineRetail
GROUP BY Country
ORDER BY TotalSales DESC
LIMIT 1;
```

f. Find Country with Minimum Sales

To find the **country** with the minimum sales:

sql

CopyEdit

```
SELECT Country, SUM(Quantity * UnitPrice) AS TotalSales
FROM OnlineRetail
GROUP BY Country
ORDER BY TotalSales ASC
LIMIT 1;
```

Step 5: Create an External Hive Table to Connect to HBase

Now, we will create an **external Hive table** to connect to HBase. This allows Hive to read and write to HBase.

a. Create HBase Table

First, create a table in HBase to store the OnlineRetail data:

bash

CopyEdit

```
create 'online_retail', 'details'
```

This command creates an HBase table called online_retail with a column family details.

b. Create External Hive Table

Next, create an external Hive table to interact with the online_retail HBase table:

sql

CopyEdit

```
CREATE EXTERNAL TABLE OnlineRetail_HBase (
  InvoiceNo STRING,
  StockCode STRING,
  Description STRING,
  Quantity INT,
  InvoiceDate STRING,
  UnitPrice DOUBLE,
  CustomerID INT,
  Country STRING
)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
```

WITH SERDEPROPERTIES ("hbase.columns.mapping" = ":key,details:stockcode,details:description,details:quantity,details:invoicedate,details:unitprice,details:customerid,details:country")

TBLPROPERTIES ("hbase.table.name" = "online_retail");

This command creates an external Hive table OnlineRetail_HBase that points to the HBase table online_retail.

Step 6: Display Records from the OnlineRetail Table in HBase

a. Query Data in Hive from the External HBase Table

To retrieve records from the external Hive table that is connected to HBase:

sql

CopyEdit

SELECT * FROM OnlineRetail_HBase LIMIT 10;

This query will display the first 10 records from the HBase table using Hive.

b. Scan HBase Table Directly

If you want to view the data directly in HBase, you can use the following HBase shell command:

bash

CopyEdit

scan 'online_retail'

This command will display the rows stored in the online_retail HBase table.

Step 7: Verify Data

Finally, you should verify that the data is correctly loaded into Hive and HBase.

1. **Check Hive Queries**: Run queries in Hive to ensure that you get the expected results (e.g., total sales, average sales).
2. **Check HBase Data**: Use scan in HBase to verify that data is being correctly inserted and stored.

Complete List of Commands

Here's a **concise list** of all the commands used:

1. **Hive Table Creation**:

sql

CopyEdit

```
CREATE TABLE OnlineRetail (
  InvoiceNo STRING,
  StockCode STRING,
  Description STRING,
  Quantity INT,
  InvoiceDate STRING,
  UnitPrice DOUBLE,
  CustomerID INT,
  Country STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE;
```

2. **Load Data into Hive Table**:

sql

CopyEdit

```
LOAD DATA LOCAL INPATH '/path/to/OnlineRetail.csv' INTO TABLE OnlineRetail;
```

3. **Create Index on Country Column**:

sql

CopyEdit

```
CREATE INDEX idx_country
ON TABLE OnlineRetail (Country)
AS 'COMPACT' WITH DEFERRED REBUILD;
```

4. **Queries**:
    - Total Sales:

sql

CopyEdit

```
SELECT SUM(Quantity * UnitPrice) AS TotalSales FROM OnlineRetail;
```

    - Average Sales:

sql

CopyEdit

```
SELECT AVG(Quantity * UnitPrice) AS AverageSales FROM OnlineRetail;
```

    - Order with Maximum Cost:

sql

CopyEdit

```
SELECT InvoiceNo, SUM(Quantity * UnitPrice) AS TotalCost
FROM OnlineRetail
GROUP BY InvoiceNo
ORDER BY TotalCost DESC
LIMIT 1;
```

    - Customer with Maximum Order Total:

sql

CopyEdit

```
SELECT CustomerID, SUM(Quantity * UnitPrice) AS TotalOrderValue
FROM OnlineRetail
```

```
GROUP BY CustomerID
ORDER BY TotalOrderValue DESC
LIMIT 1;
```

- Country with Maximum Sales:

```
sql
CopyEdit
SELECT Country, SUM(Quantity * UnitPrice) AS TotalSales
FROM OnlineRetail
GROUP BY Country
ORDER BY TotalSales DESC
LIMIT 1;
```

- Country with Minimum Sales:

```
sql
CopyEdit
SELECT Country, SUM(Quantity * UnitPrice) AS TotalSales
FROM OnlineRetail
GROUP BY Country
ORDER BY TotalSales ASC
LIMIT 1;
```

5. **HBase Table Creation**:

```
bash
CopyEdit
create 'online_retail', 'details'
```

6. **External Hive Table Creation**:

```
sql
CopyEdit
```

```
CREATE EXTERNAL TABLE OnlineRetail_HBase (
  InvoiceNo STRING,
  StockCode STRING,
  Description STRING,
  Quantity INT,
  InvoiceDate STRING,
  UnitPrice DOUBLE,
  CustomerID INT,
  Country STRING
)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES ("hbase.columns.mapping" =
":key,details:stockcode,details:description,details:quantity,details:invoicedate,details:unitprice,details:customerid,details:country")
TBLPROPERTIES ("hbase.table.name" = "online_retail");
```

7. **Scan Data from HBase**:

```
bash
CopyEdit
scan 'online_retail'
```