✅ Step 1: Create `onlineretail.csv` via `echo` (sample row)

Here's one valid sample row from the dataset:

bash
CopyEdit

```
echo "536365,85123A,WHITE HANGING HEART T-LIGHT HOLDER,6,01-12-2010
08:26,2.55,17850,United Kingdom" > ~/Desktop/onlineretail.csv
```

✔ This creates the file on your **Desktop**.

> 📝 You can add more rows manually or extend with additional `echo` and `>>` commands.

✅ Step 2: Put the file into HDFS

bash
CopyEdit

```
hdfs dfs -mkdir -p /user/hive
hdfs dfs -put ~/Desktop/onlineretail.csv /user/hive
```

✅ Step 3: Create Hive Table

sql
CopyEdit

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

✅ Step 4: Load data into Hive table

sql
CopyEdit

```
LOAD DATA LOCAL INPATH '/home/cloudera/Desktop/onlineretail.csv'
INTO TABLE online_retail;
```

✅ Step 5: Create and rebuild index

sql
CopyEdit

```
CREATE INDEX idx_invoice ON TABLE online_retail (InvoiceNo)
AS 'COMPACT'
WITH DEFERRED REBUILD;

ALTER INDEX idx_invoice ON online_retail REBUILD;

SHOW INDEXES ON online_retail;
```

✅ Step 6: Sales Analysis Queries

Total & Average Sales

sql
CopyEdit

```
SELECT
  SUM(Quantity * UnitPrice) AS total_sales,
  AVG(Quantity * UnitPrice) AS average_sales
FROM online_retail;
```

Max Cost Order

sql
CopyEdit

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

Max Order Total by Customer

sql
CopyEdit

```sql
SELECT CustomerID, SUM(Quantity * UnitPrice) AS customer_total
FROM online_retail
GROUP BY CustomerID
ORDER BY customer_total DESC
LIMIT 1;
```

Country with Max Sale

sql
CopyEdit

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

Country with Min Sale

sql
CopyEdit

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

✅ Step 7: Create HBase Table and Insert Row

bash
CopyEdit

```bash
hbase shell
```

Inside shell:

hbase
CopyEdit

```
create 'online_retail', 'cf'

put 'online_retail', '536365_85123A', 'cf:InvoiceNo', '536365'
```

```
put 'online_retail', '536365_85123A', 'cf:StockCode', '85123A'
put 'online_retail', '536365_85123A', 'cf:Description', 'WHITE
HANGING HEART T-LIGHT HOLDER'
put 'online_retail', '536365_85123A', 'cf:Quantity', '6'
put 'online_retail', '536365_85123A', 'cf:InvoiceDate', '01-12-2010
08:26'
put 'online_retail', '536365_85123A', 'cf:UnitPrice', '2.55'
put 'online_retail', '536365_85123A', 'cf:CustomerID', '17850'
put 'online_retail', '536365_85123A', 'cf:Country', 'United Kingdom'
```

✅ Step 8: Create Hive External Table for HBase

sql
CopyEdit

```
CREATE EXTERNAL TABLE retail_hive_view (
  rowkey STRING,
  InvoiceNo STRING,
  StockCode STRING,
  Description STRING,
  Quantity INT,
  InvoiceDate STRING,
  UnitPrice DOUBLE,
  CustomerID STRING,
  Country STRING
)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES (
  "hbase.columns.mapping" =
":key,cf:InvoiceNo,cf:StockCode,cf:Description,cf:Quantity,cf:Invoic
eDate,cf:UnitPrice,cf:CustomerID,cf:Country"
)
TBLPROPERTIES (
  "hbase.table.name" = "online_retail"
);
```

✅ Step 9: View records in Hive from HBase

sql
CopyEdit

```
SELECT * FROM retail_hive_view;
```