📂 Step 1: Launch Hive

bash
CopyEdit

```
hive
```

🛢️ Step 2: Create Database (Optional)

sql
CopyEdit

```
CREATE DATABASE IF NOT EXISTS customer_db;
USE customer_db;
```

🖊️ Step 3: Create Tables

sql
CopyEdit

```
CREATE TABLE IF NOT EXISTS customer_info (
    cust_id INT,
    cust_name STRING,
    order_id INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

CREATE TABLE IF NOT EXISTS order_info (
    order_id INT,
    item_id INT,
    quantity INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

CREATE TABLE IF NOT EXISTS item_info (
    item_id INT,
    item_name STRING,
    item_price FLOAT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

📄 Step 4: Create CSV Files in Terminal

bash

CopyEdit

```
cat <<EOF > customer_info.csv
101,John,1
102,Alice,2
103,Bob,3
EOF

cat <<EOF > order_info.csv
1,201,2
2,202,1
3,203,5
EOF

cat <<EOF > item_info.csv
201,Laptop,80000
202,Mouse,500
203,Keyboard,1000
EOF
```

📤 Step 5: Upload CSV Files to HDFS

bash

CopyEdit

```
hdfs dfs -mkdir -p /user/cloudera/customer

hdfs dfs -put customer_info.csv /user/cloudera/customer/
hdfs dfs -put order_info.csv /user/cloudera/customer/
hdfs dfs -put item_info.csv /user/cloudera/customer/
```

📥 Step 6: Load Data into Hive Tables

sql
CopyEdit

```
LOAD DATA INPATH '/user/cloudera/customer/customer_info.csv' INTO
TABLE customer_info;
LOAD DATA INPATH '/user/cloudera/customer/order_info.csv' INTO TABLE
order_info;
LOAD DATA INPATH '/user/cloudera/customer/item_info.csv' INTO TABLE
item_info;
```

🔗 Step 7: Perform Join Query

sql
CopyEdit

```
SELECT ci.cust_id, ci.cust_name, ii.item_name, oi.quantity,
ii.item_price,
       (oi.quantity * ii.item_price) AS total_cost
FROM customer_info ci
JOIN order_info oi ON ci.order_id = oi.order_id
JOIN item_info ii ON oi.item_id = ii.item_id;
```

📇 Step 8: Create Index on Customer Info Table

> Hive **indexing is deprecated** in recent versions, but for learning:

sql
CopyEdit

```
CREATE INDEX cust_idx ON TABLE customer_info (cust_id)
AS 'org.apache.hadoop.hive.ql.index.compact.CompactIndexHandler'
WITH DEFERRED REBUILD;

ALTER INDEX cust_idx ON customer_info REBUILD;
```

📊 Step 9: Total and Average Sales

sql

CopyEdit

```
SELECT
    SUM(oi.quantity * ii.item_price) AS total_sales,
    AVG(oi.quantity * ii.item_price) AS average_sales
FROM order_info oi
JOIN item_info ii ON oi.item_id = ii.item_id;
```

🏆 Step 10: Order with Maximum Cost

sql

CopyEdit

```
SELECT ci.cust_id, ci.cust_name, oi.order_id,
       (oi.quantity * ii.item_price) AS total_cost
FROM customer_info ci
JOIN order_info oi ON ci.order_id = oi.order_id
JOIN item_info ii ON oi.item_id = ii.item_id
ORDER BY total_cost DESC
LIMIT 1;
```

🌐 Step 11: External Hive Table for HBase (Integration)

sql

CopyEdit

```
CREATE EXTERNAL TABLE hbase_customer_info(
    cust_id STRING,
    cust_name STRING,
    order_id STRING
)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES (
    "hbase.columns.mapping" = ":key,info:name,info:order_id"
)
TBLPROPERTIES (
    "hbase.table.name" = "customer_info_hbase"
);
```

Make sure HBase is running and the table `customer_info_hbase` exists or will be created.

📋 Step 12: View Data from HBase Table

sql
CopyEdit

```sql
SELECT * FROM hbase_customer_info;
```