✅ A. CREATE HIVE TABLES

sql
CopyEdit

```
CREATE TABLE customer_info (
  cust_id STRING,
  cust_name STRING,
  order_id STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

CREATE TABLE order_info (
  order_id STRING,
  item_id STRING,
  quantity INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

CREATE TABLE item_info (
  item_id STRING,
  item_name STRING,
  item_price FLOAT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

✅ B. CSV FILES AND LOAD TO HDFS

Step 1: Create CSV Files (Local)

📄 **customer_info.csv**

bash
CopyEdit

```
echo -e "001,ALICE,101\n002,BOB,102\n003,CHARLIE,103\n004,DIANA,104\n005,ETHAN,105\n006,FIONA,106" > ~/Desktop/customer_info.csv
```

📄 order_info.csv

bash
CopyEdit

```
echo -e "101,1,2\n102,2,1\n103,3,1\n104,4,2\n105,5,1\n106,1,2" > ~/Desktop/order_info.csv
```

📄 item_info.csv ✅ Fixed

bash
CopyEdit

```
echo -e "1,Monitor,200.0\n2,Mouse,20.0\n3,Keyboard,30.0\n4,Laptop,1500.0\n5,Webcam,70.0" > ~/Desktop/item_info.csv
```

Step 2: Upload to HDFS

bash
CopyEdit

```
hdfs dfs -mkdir -p /user/hive/inputfiles/
hdfs dfs -put ~/Desktop/customer_info.csv /user/hive/inputfiles/
hdfs dfs -put ~/Desktop/order_info.csv /user/hive/inputfiles/
hdfs dfs -put ~/Desktop/item_info.csv /user/hive/inputfiles/
```

Step 3: Load CSV into Hive Tables

sql
CopyEdit

```
LOAD DATA INPATH '/user/hive/inputfiles/customer_info.csv' INTO TABLE customer_info;
LOAD DATA INPATH '/user/hive/inputfiles/order_info.csv' INTO TABLE order_info;
LOAD DATA INPATH '/user/hive/inputfiles/item_info.csv' INTO TABLE item_info;
```

✅ C. JOIN TABLES

sql
CopyEdit

```
SELECT
  c.cust_id,
  c.cust_name,
  o.order_id,
  o.item_id,
  o.quantity,
  i.item_name,
  i.item_price
FROM
  customer_info c
LEFT JOIN
  order_info o ON c.order_id = o.order_id
LEFT JOIN
  item_info i ON o.item_id = i.item_id;
```

✅ D. CREATE INDEX on `cust_id`

sql
CopyEdit

```
CREATE INDEX idx_cust_id ON TABLE customer_info (cust_id)
AS 'COMPACT'
WITH DEFERRED REBUILD;

ALTER INDEX idx_cust_id ON customer_info REBUILD;
```

✅ E. TOTAL & AVERAGE SALES

sql
CopyEdit

```
-- Total sales
SELECT SUM(o.quantity * i.item_price) AS total_sales
FROM order_info o
JOIN item_info i ON o.item_id = i.item_id;

-- Average sales
SELECT AVG(o.quantity * i.item_price) AS avg_sales
```

```
FROM order_info o
JOIN item_info i ON o.item_id = i.item_id;
```

✅ F. ORDER DETAILS WITH MAX COST

sql
CopyEdit

```
SELECT o.order_id, o.item_id, o.quantity, i.item_name, i.item_price,
       (o.quantity * i.item_price) AS total_cost
FROM order_info o
JOIN item_info i ON o.item_id = i.item_id
ORDER BY total_cost DESC
LIMIT 1;
```

✅ G. HBASE TABLE + HIVE EXTERNAL LINK

In HBase Shell:

bash
CopyEdit

```
hbase shell
```

hbase
CopyEdit

```
create 'hbase_customer_info', 'info'
put 'hbase_customer_info', 'C001', 'info:cust_name', 'John'
put 'hbase_customer_info', 'C001', 'info:order_id', 'ORD001'
put 'hbase_customer_info', 'C002', 'info:cust_name', 'Kamal'
put 'hbase_customer_info', 'C002', 'info:order_id', 'ORD002'
put 'hbase_customer_info', 'C003', 'info:cust_name', 'Lando'
put 'hbase_customer_info', 'C003', 'info:order_id', 'ORD003'
exit
```

In Hive (External Table Link to HBase):

sql
CopyEdit

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

✅ H. VIEW HBASE TABLE DATA IN HIVE

sql
CopyEdit

```
SELECT * FROM hbase_customer;
```