CREATE DATABASE landing\_v2;

USE landing\_v2;

CREATE TABLE IF NOT EXISTS cheque(

order\_code STRING,

product\_code STRING,

price STRING,

qty STRING,

currency\_code STRING,

shop STRING,

order\_comment STRING

)

COMMENT 'Table for raw cheque data'

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'

WITH SERDEPROPERTIES (

"quoteChar" = "\""

)

STORED AS TEXTFILE

TBLPROPERTIES("skip.header.line.count"="1");

LOAD DATA INPATH '/user/raj\_ops/TestTask1/cheque\_20200101.csv' INTO TABLE cheque;

CREATE TABLE IF NOT EXISTS Rate(

currency STRING,

rate STRING

)

COMMENT 'Table for raw rate data'

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'

WITH SERDEPROPERTIES (

"quoteChar" = "\""

)

STORED AS TEXTFILE

TBLPROPERTIES("skip.header.line.count"="1");

LOAD DATA INPATH '/user/raj\_ops/TestTask1/rate\_20200101.csv' INTO TABLE Rate;

CREATE TABLE IF NOT EXISTS Product(

product\_code STRING,

product\_name STRING,

product\_descritpion STRING

)

COMMENT 'Table for raw product data'

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'

WITH SERDEPROPERTIES (

"quoteChar" = "\""

)

STORED AS TEXTFILE

TBLPROPERTIES("skip.header.line.count"="1");

LOAD DATA INPATH '/user/raj\_ops/TestTask1/product\_20200101.csv' INTO TABLE product;

CREATE TABLE IF NOT EXISTS Currency(

currency\_code STRING,

currency\_name STRING

)

COMMENT 'Table for raw currency data'

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'

WITH SERDEPROPERTIES (

"quoteChar" = "\""

)

STORED AS TEXTFILE

TBLPROPERTIES("skip.header.line.count"="1");

LOAD DATA INPATH '/user/raj\_ops/TestTask1/Currency\_20200101.csv' INTO TABLE currency;

CREATE DATABASE Staging\_v2;

USE Staging\_v2;

CREATE TABLE IF NOT EXISTS Products(

product\_ID INT,

product\_code VARCHAR(4),

product\_name VARCHAR(50),

product\_description VARCHAR(50),

product\_price\_UAH DECIMAL(10,2)

)

STORED AS ORC;

INSERT INTO TABLE

staging\_v2.products

SELECT

row\_number() over() as id, product.product\_code, product.product\_name, product.product\_descritpion,

MAX(cheque.price\*rate.rate) as price\_in\_UAH

FROM

landing\_v2.product INNER JOIN landing\_v2.cheque ON product.product\_code = cheque.product\_code

INNER JOIN landing\_v2.rate ON rate.currency = cheque.currency\_code

GROUP BY

product.product\_code, product.product\_name, product.product\_descritpion;

CREATE TABLE IF NOT EXISTS Currency(

currency\_id INT,

currency\_code VARCHAR(3),

currency\_name VARCHAR(20)

)

COMMENT 'Handbook of currencies'

STORED AS ORC;

WITH cte\_currency AS(

SELECT row\_number() over() as id, \* FROM landing\_v2.Currency

)

INSERT INTO TABLE staging\_v2.currency

SELECT \* FROM cte\_currency;

CREATE TABLE IF NOT EXISTS Rates(

rate\_id INT,

currency VARCHAR(3),

rate DECIMAL(10,2),

rate\_date DATE

)

COMMENT 'Currency rate'

STORED AS ORC;

CREATE TEMPORARY TABLE temp\_rates AS SELECT row\_number() over() as id, \* , '2020-01-01' FROM landing\_v2.rate;

INSERT INTO TABLE staging\_v2.rates

SELECT row\_number() over() as id, \* FROM temp\_rates;

CREATE TABLE IF NOT EXISTS Countries(

country\_id INT,

country\_name VARCHAR(56)

)

COMMENT 'Country handbook'

STORED AS ORC;

WITH cte\_Location AS (

SELECT split(shop, ',')[0] FROM landing\_v2.cheque GROUP BY shop)

INSERT INTO TABLE Countries

SELECT row\_number() over() as id, \* FROM cte\_Location;

CREATE TABLE IF NOT EXISTS Cities(

city\_id INT,

city\_name VARCHAR(56)

)

COMMENT 'City handbook'

STORED AS ORC;

WITH cte\_Location AS (

SELECT trim(split(shop, ',')[1]) FROM landing\_v2.cheque GROUP BY shop)

INSERT INTO TABLE Cities

SELECT row\_number() over() as id, \* FROM cte\_Location;

CREATE TABLE IF NOT EXISTS Shops(

shop\_id INT,

city\_id INT,

country\_id INT

)

COMMENT 'Shop handbook'

STORED AS ORC;

WITH cte\_Location AS (SELECT city\_id, country\_id FROM Staging\_v2.cities as c INNER JOIN

(SELECT split(shop, ',')[0] as country, split(shop, ',')[1] as city

FROM landing\_v2.cheque GROUP BY shop) AS test ON c.city\_name = test.city INNER JOIN

Staging\_v2.countries as co ON co.country\_name = test.country )

INSERT INTO TABLE staging\_v2.shops

SELECT row\_number() over() as id, \* FROM cte\_Location;

CREATE TABLE IF NOT EXISTS Orders(

order\_id INT,

order\_code VARCHAR(4),

order\_date DATE,

shop\_id INT,

total\_sum DECIMAL(10,2)

)

COMMENT 'Orders list'

STORED AS ORC;

CREATE TEMPORARY TABLE temp\_Orders AS

SELECT

cheque.order\_code, '2020-01-01', shops.shop\_id,

SUM(cheque.price\*cheque.qty\*rate.rate)

FROM

Staging\_v2.shops JOIN Staging\_v2.cities

ON (shops.city\_id = cities.city\_id)

JOIN Staging\_v2.countries

ON(shops.country\_id = countries.country\_id)

JOIN Landing\_v2.cheque

ON (cities.city\_name = trim(split(cheque.shop, ',')[1])

AND (Countries.country\_name = trim(split(cheque.shop, ',')[0])))

JOIN Landing\_v2.rate

ON(rate.currency = cheque.currency\_code)

GROUP BY cheque.order\_code, shops.shop\_id;

INSERT INTO TABLE Orders

SELECT row\_number() over() as ID,\* FROM temp\_Orders;

CREATE TABLE IF NOT EXISTS order\_details (

order\_ID INT,

product\_ID INT,

quantity INT,

commentary VARCHAR(200),

currency\_ID INT,

Product\_price DECIMAL(10,2)

)

COMMENT 'Order details'

STORED AS ORC;

CREATE TEMPORARY TABLE temp\_Order\_details AS

SELECT

orders.order\_id, products.product\_id, cheque.qty,cheque.order\_comment,

currency.currency\_id, cheque.price

FROM

Landing\_v2.cheque JOIN Staging\_v2.orders

ON(cheque.order\_code = orders.order\_code)

JOIN Staging\_v2.Products

ON(cheque.product\_code = Products.product\_code)

JOIN Staging\_v2.currency

ON(cheque.currency\_code = currency.currency\_code);

INSERT INTO TABLE order\_details

SELECT row\_number() over() as ID,\* FROM temp\_Order\_details;