1. Download vehicle sales data

2. Store raw data in the hdfs location.

```
#Create a new directory
hadoop fs -mkdir /sales

#store data in that directory
hadoop fs -put sales_order_data.csv /sales
```

3. Create an internal hive table "sales_order_csv" which will store csv data sales_order_csv. Make sure to skip the header row while creating a table.

```
#Create a new database
create database hive_db;
use hive_db;

create table sales_order_csv(
ORDERNUMBER int,
QUANTITYORDERED int,
PRICEEACH float,
ORDERLINENUMBER int,
SALES float,
STATUS string,
QTR_ID int,
MONTH_ID int,
YEAR_ID int,
PRODUCTLINE string,
MSRP int,
PRODUCTCODE string,
PHONE string,
CITY string,
STATE string,
POSTALCODE string,
COUNTRY string,
TERRITORY string,
CONTACTLASTNAME string,
CONTACTFIRSTNAME string,
DEALSIZE string
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
TBLPROPERTIES ("skip.header.line.count"="1");   #skipping header row
```

4. Load data from the hdfs path into "sales_order_csv".

```
>Load data inpath '/sales/sales_order_data.csv' into table sales_order_csv;

>set hive.cli.print.header =true;      #to display data along with column names

>select * from sales_order_csv limit 10;
```

5. Create an internal hive table which will store data in ORC format "sales_order_orc".

```
create table sales_order_orc(
ORDERNUMBER int,
QUANTITYORDERED int,
PRICEEACH float,
ORDERLINENUMBER int,
SALES float,
STATUS string,
QTR_ID int,
MONTH_ID int,
YEAR_ID int,
PRODUCTLINE string,
MSRP int,
PRODUCTCODE string,
PHONE string,
CITY string,
STATE string,
POSTALCODE string,
COUNTRY string,
TERRITORY string,
CONTACTLASTNAME string,
CONTACTFIRSTNAME string,
DEALSIZE string
)
Stored as ORC;
```

6. Load data from "sales_order_csv" into "sales_order_orc".

```
>Insert into table sales_order_orc
>Select * from sales_order_csv;
>select * from sales_order_orc limit 10;
```

Perform below mentioned queries on the "sales_order_orc" table :

a. Calculate total sales per year

```
>SELECT sales_order_orc.year_id as sales_year, SUM(sales_order_orc.sales) as
>total_sales
>FROM sales_order_orc
>GROUP BY sales_order_orc.year_id;
```

b. Find a product for which maximum orders were placed.

```
>SELECT sales_order_orc.productcode as product, MAX(sales_order_orc.quantityordered)
>as order_placed
>FROM sales_order_orc
>GROUP BY sales_order_orc.productcode
```

c. Calculate the total sales for each quarter.

```
>SELECT sales_order_orc.qtr_id as sales_quarter, sum(sales_order_orc.sales) as
>total_sales
>FROM sales_order_orc
>GROUP BY sales_order_orc.qtr_id;
```

d. In which quarter sales were minimum.

```
>SELECT sales_quarter
>FROM (
> SELECT sales_order_orc.qtr_id as sales_quarter, sum(sales_order_orc.sales) as
>total_sales
>FROM sales_order_orc
>GROUP BY sales_order_orc.qtr_id
>ORDER BY sales_order_orc.qtr_id limit 1;
>) as sales_quater;
```

e. In which country sales were maximum and in which country sales were minimum.

```
>SELECT sales_order_orc.country as max_sales_country, SUM(sales_order_orc.sales) as
>total_sales
>FROM sales_order_orc
>GROUP BY sales_order_orc.country
>ORDER BY total_sales DESC
>LIMIT 1;
```

```
>SELECT sales_order_orc.country as min_sales_country, SUM(sales_order_orc.sales) as
>total_sales
>FROM sales_order_orc
>GROUP BY sales_order_orc.country
>ORDER BY total_sales ASC
>LIMIT 1;
```

f. Calculate quarterly sales for each city.

```
>SELECT sales_order_orc.state as city, sales_order_orc.qtr_id as sales_quarter,
>sum(sales_order_orc.sales) as total_sales
>FROM sales_order_orc
>GROUP BY city, qtr_id;
```

h. Find a month for each year in which the maximum number of quantities were sold.

```
>select year, month from (select year_id as year, month_id as month, sum(quantityordered)
as total_quantity, dense_rank() over(partition by year_id order by sum(quantityordered)
desc) as r from sales_order_orc group by year_id, month_id)tab where r=1;
```