**Title**: **KISAN STORE**

**Name**: Chandra Siva Charan Daggubati

**Github link**: <https://github.com/charanD09/KisanStore.git>

**Date**: 22-March-2022

**INTRODUCTION**

* 1. Project Description

I have created a web development application called Kisan Store. I'm trying to come up with a database system that keeps track of customer, product, and employee information. This web application will help farmers sell their products online, namely vegetables, fruits, and grains. The Kisan store will act as a bridge between farmers and customers. I’m interested in helping farmers' products sell online, which makes farmers profit as well as customers happy by providing them with organic products. I’m interested in getting data because I need to keep track of my business operations to make sure that I have the right components. I will manually enter employee data. My customers generate data when they produce orders. I will also manually add data from suppliers (farmers). I want to store my data in a database and do all kinds of operations whenever required. All kinds of records can be fetched from the database without any loss of data.

Week 2

For my business, I will have the following entities:

1. Product\_Info – product\_name, product\_id, Cost,brand\_id,customer\_id
2. Customer\_Info – cutomer\_Phone,customer\_id, first\_name, last\_name,address, zipcode,product\_Info\_product\_id,order\_id
3. Supplier\_Info – supplier\_ID,supplier\_Fname, supplier\_Lname, company\_name, address,Product\_Info\_product\_id
4. Sales\_Info – Item sold, date, price, profit,supplier\_Id

5. Store\_Info –store\_id, store\_name,phone,email,address,zipcode ,sales\_Id,sales\_Info\_sales\_Id

6. Order\_Info – order\_id, cutomer\_id, order\_status, order\_date,store\_id ,store\_Info\_store\_id

Tables:

|  |  |
| --- | --- |
| **Product Info** |  |
| product\_name | varChar[25] |
| product\_id | Integer |
| Cost | Float |
| brand\_id | Integer |
| Customer\_id | Integer |

|  |  |
| --- | --- |
| **Customer Info** |  |
| cutomer\_Phone | varChar[25] |
| customer\_id | Integer |
| first\_name | varChar[25] |
| last\_name | varChar[25] |
| address | varChar[50] |
| zipcode | Integer |
| Product\_Info\_product\_id | integer |

|  |  |
| --- | --- |
| **Supplier Info** |  |
| supplier\_ID | Integer |
| supplier\_Fname | varChar[25] |
| supplier\_Lname | varChar[25] |
| company\_name | varChar[20] |
| address | varChar[50] |
| Product\_Info\_product\_id | integer |

|  |  |
| --- | --- |
| **Sales Info** |  |
| Item sold | varChar[20] |
| date | DateTime |
| price | Float |
| profit | Float |
| Supplier\_ID | Integer |
| Sales\_Id | Integer |

|  |  |
| --- | --- |
| **Store Info** |  |
| store\_id | integer |
| store\_name | varChar[25] |
| phone | varChar[15] |
| email | varChar[25] |
| address | varChar[50] |
| zipcode | Integer |
| Sales\_id | integer |
| Sales\_Info\_Sales\_Id | integer |

|  |  |
| --- | --- |
| **Order Info** |  |
| order\_id | Integer |
| cutomer\_id | Integer |
| order\_status | varChar[10] |
| order\_date | DateTime |
| store\_id | Integer |
| Store\_info\_sales\_id | integer |

Also pushed files containing records of my data required for project in my github repository

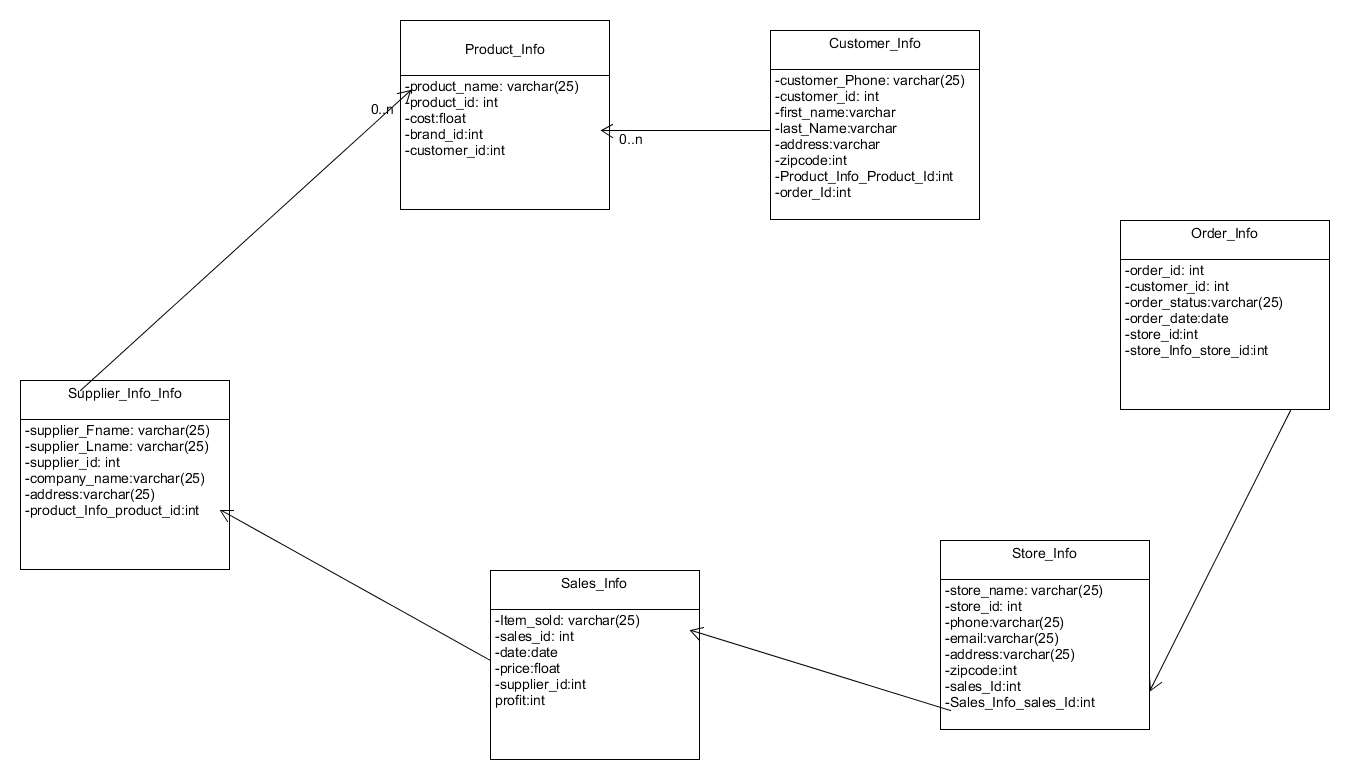
FileNames:

* Products\_Info.csv
* Customer\_Info.csv
* order\_Info.csv
* supplier\_Info.csv
* store\_Info.csv
* sales\_Info.csv

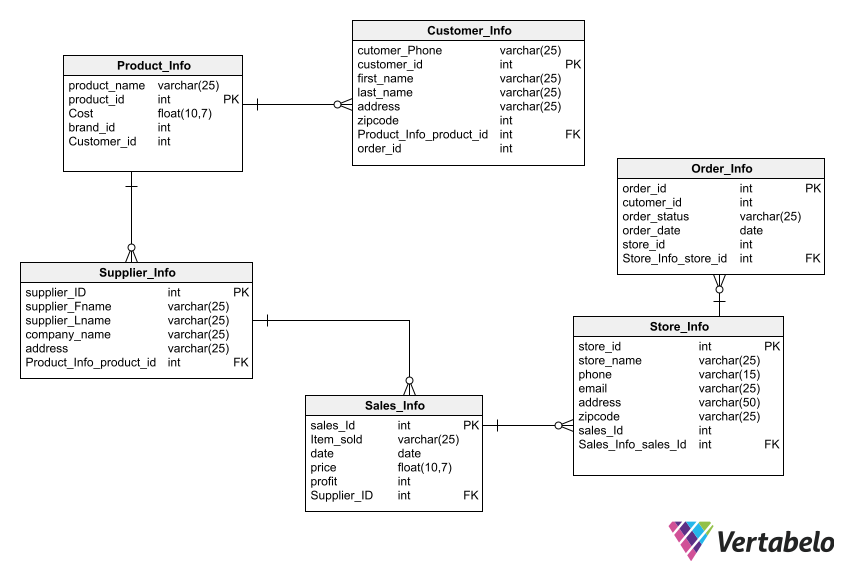
Week3

ER Diagrams

1. Umlet Free Utility Diagram



1. Crow’s foot model



**Week4:**

Scripts and Outputs:

1. **Create tables commands**
2. Customer\_Info

CREATE TABLE Customer\_Info (

cutomer\_Phone varchar(25) NOT NULL,

customer\_id int NOT NULL,

first\_name varchar(25) NOT NULL,

last\_name varchar(25) NOT NULL,

address varchar(25) NOT NULL,

zipcode int NOT NULL,

Product\_Info\_product\_id int NOT NULL,

order\_id int NOT NULL,

CONSTRAINT Customer\_Info\_pk PRIMARY KEY (customer\_id)

);

1. Product\_Info

CREATE TABLE Product\_Info (

product\_name varchar(25) NOT NULL,

product\_id int NOT NULL,

Cost float(10,7) NOT NULL,

brand\_id int NOT NULL,

Customer\_id int NOT NULL,

CONSTRAINT Product\_Info\_pk PRIMARY KEY (product\_id)

);

1. Supplier\_Info

CREATE TABLE Supplier\_Info (

supplier\_ID int NOT NULL,

supplier\_Fname varchar(25) NOT NULL,

supplier\_Lname varchar(25) NOT NULL,

company\_name varchar(25) NOT NULL,

address varchar(25) NOT NULL,

Product\_Info\_product\_id int NOT NULL,

CONSTRAINT Supplier\_Info\_pk PRIMARY KEY (supplier\_ID)

);

D)Sales\_Info

CREATE TABLE Sales\_Info (

sales\_Id int NOT NULL,

Item\_sold varchar(25) NOT NULL,

date date NOT NULL,

price float(10,7) NOT NULL,

profit int NOT NULL,

Supplier\_ID int NOT NULL,

CONSTRAINT Sales\_Info\_pk PRIMARY KEY (sales\_Id)

);

E)Store\_Info

CREATE TABLE Store\_Info (

store\_id int NOT NULL,

store\_name varchar(25) NOT NULL,

phone varchar(15) NOT NULL,

email varchar(25) NOT NULL,

address varchar(50) NOT NULL,

zipcode varchar(25) NOT NULL,

sales\_Id int NOT NULL,

Sales\_Info\_sales\_Id int NOT NULL,

CONSTRAINT Store\_Info\_pk PRIMARY KEY (store\_id)

);

1. Order\_Info

CREATE TABLE Order\_Info (

order\_id int NOT NULL,

cutomer\_id int NOT NULL,

order\_status varchar(25) NOT NULL,

order\_date date NOT NULL,

store\_id int NOT NULL,

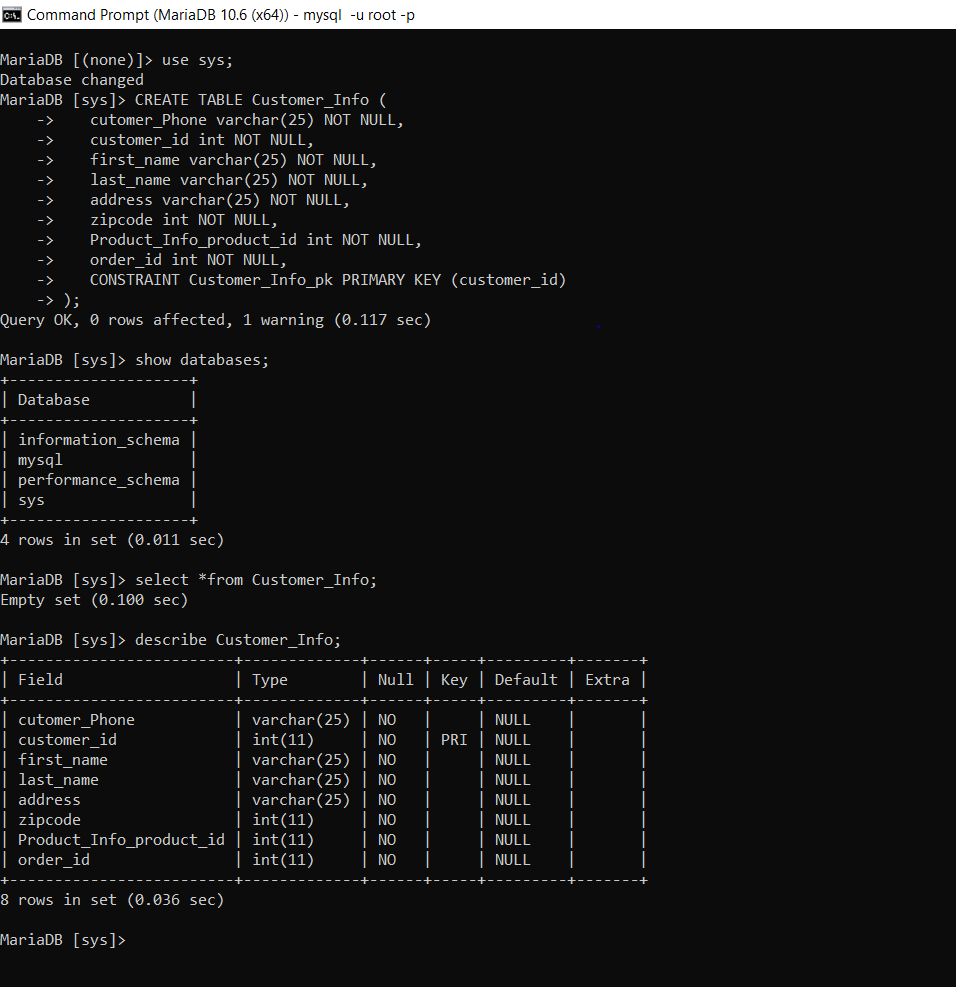
Store\_Info\_store\_id int NOT NULL,

CONSTRAINT Order\_Info\_pk PRIMARY KEY (order\_id)

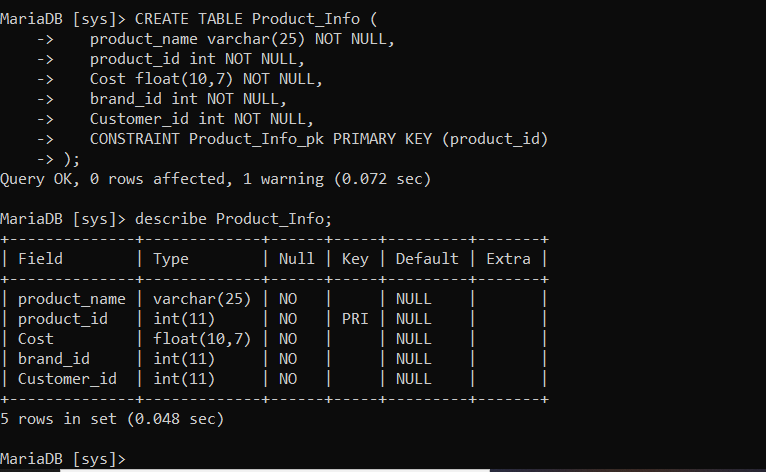
);

**OUTPUTS:**

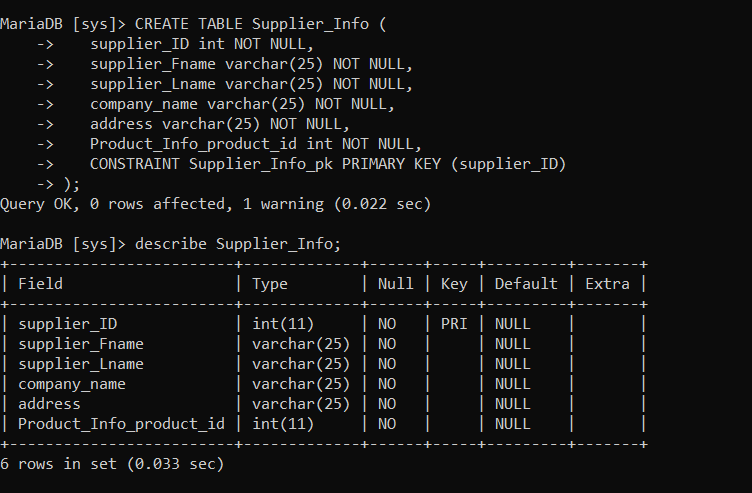
1. Create Customer\_Info table



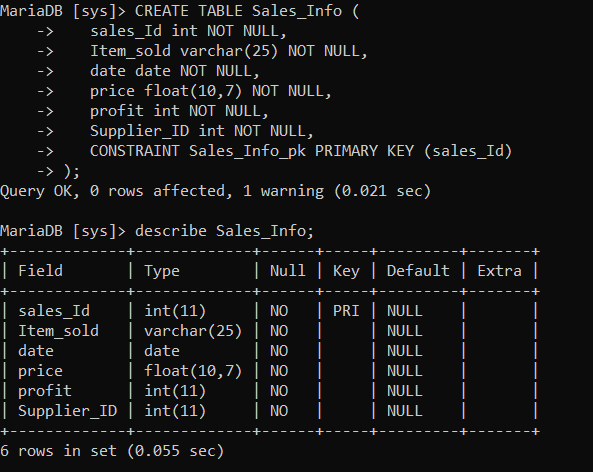
2)Create Product\_Info table



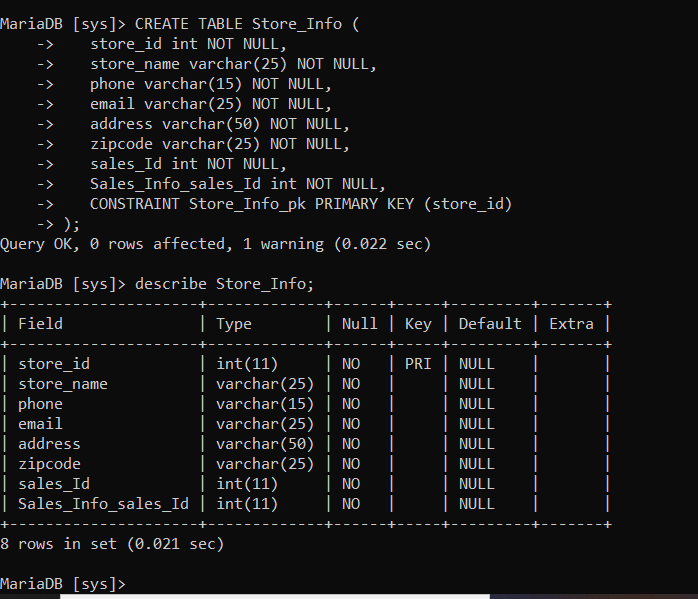
3)Create Supplier\_Info table



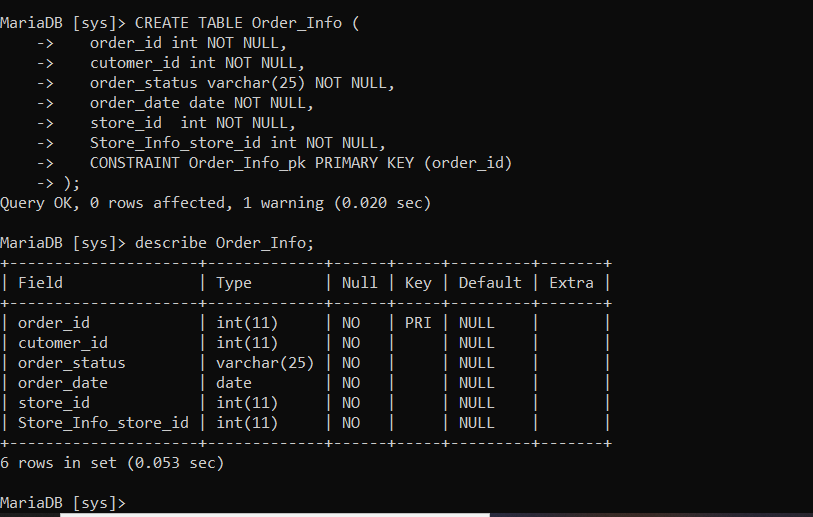
4)Create Sales\_Info table



5)Create Store\_Info table



6)Create Order\_Info table



1. **Loading the data files into tables**

**Scripts:**

1. LOAD DATA LOCAL INFILE 'c:/temp/Customer\_Info.csv'

-> INTO TABLE Customer\_Info

-> FIELDS TERMINATED BY ','

-> LINES TERMINATED BY '\r\n'

-> IGNORE 1 LINES

-> (cutomer\_Phone,customer\_id,first\_name,last\_name,address,zipcode,Product\_Info\_product\_id,order\_id);

1. LOAD DATA LOCAL INFILE 'c:/temp/Product\_Info.csv'

-> INTO TABLE Product\_Info

-> FIELDS TERMINATED BY ','

-> LINES TERMINATED BY '\r\n'

-> IGNORE 1 LINES

-> (product\_name,product\_id,Cost,brand\_id,customer\_id);

1. LOAD DATA LOCAL INFILE 'c:/temp/Supplier\_Info.csv'

-> INTO TABLE Supplier\_Info

-> FIELDS TERMINATED BY ','

-> LINES TERMINATED BY '\r\n'

-> IGNORE 1 LINES

-> (supplier\_ID,supplier\_Fname,supplier\_Lname,company\_name,address,Product\_Info\_product\_id);

1. LOAD DATA LOCAL INFILE 'c:/temp/Sales\_Info.csv'

-> INTO TABLE Sales\_Info

-> FIELDS TERMINATED BY ','

-> LINES TERMINATED BY '\r\n'

-> IGNORE 1 LINES

-> (sales\_Id,Item\_sold, date,price,profit,Supplier\_ID);

1. LOAD DATA LOCAL INFILE 'c:/temp/Store\_Info.csv'

-> INTO TABLE Store\_Info

-> FIELDS TERMINATED BY ','

-> LINES TERMINATED BY '\r\n'

-> IGNORE 1 LINES

-> (store\_id,store\_name,phone,email,address,zipcode,sales\_Id,Sales\_Info\_sales\_Id);

1. LOAD DATA LOCAL INFILE 'c:/temp/Order\_Info.csv'

-> INTO TABLE Order\_Info

-> FIELDS TERMINATED BY ','

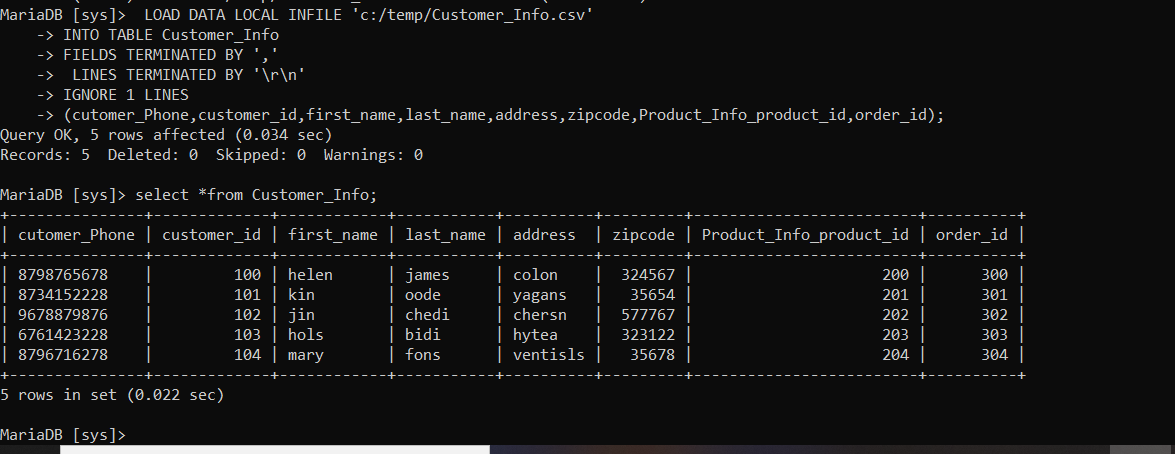
-> LINES TERMINATED BY '\r\n'

-> IGNORE 1 LINES

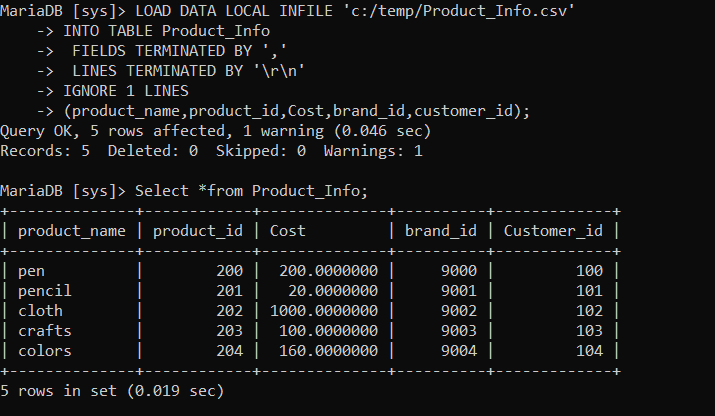
-> (order\_id,cutomer\_id,order\_status,order\_date,store\_id,store\_info\_store\_id);

**OUTPUTS:**

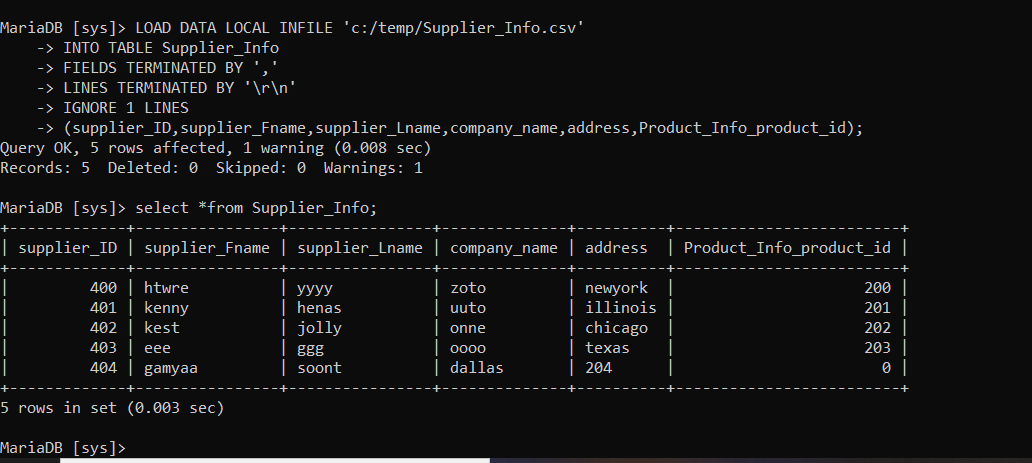
A)Loaded data present in Customer\_Info.csv( local to db)



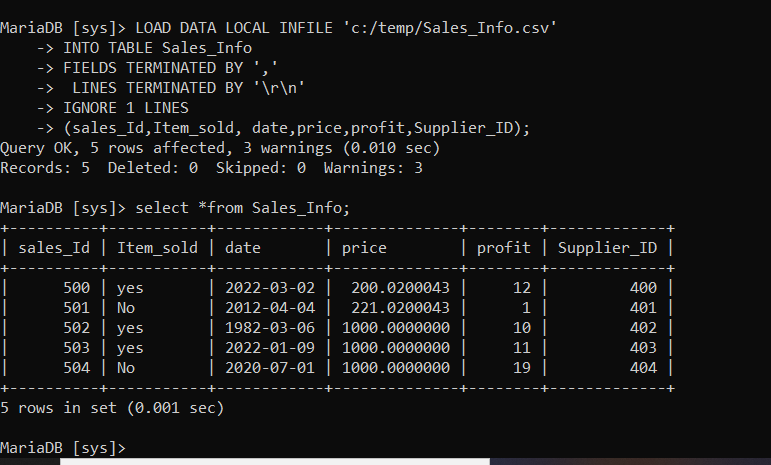
B)Loaded data present in Product\_Info.csv( local to db)



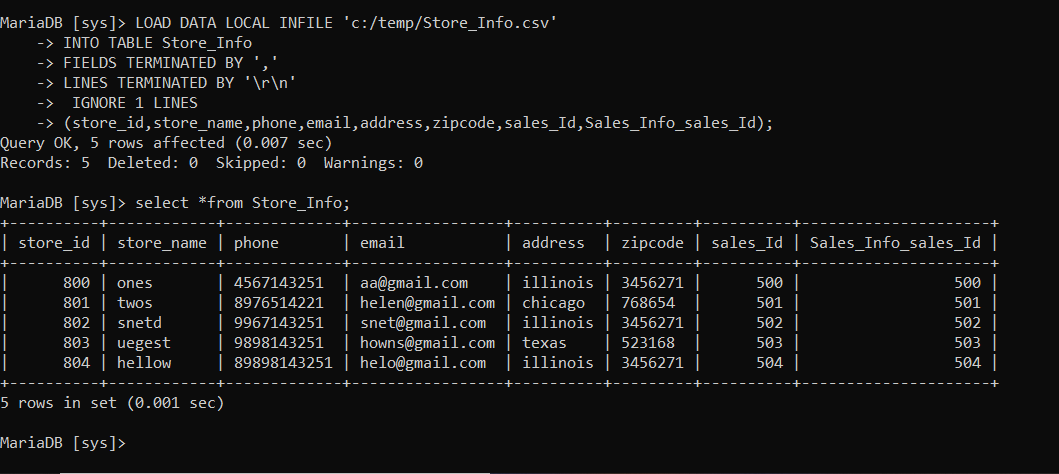
C)Loaded data present in Supplier\_Info.csv( local to db)



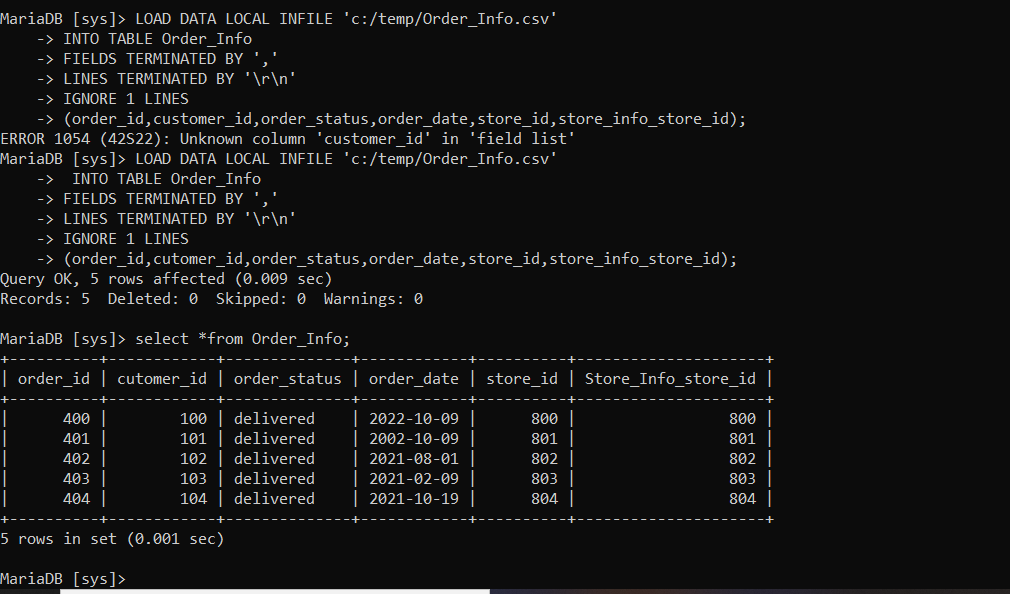
D)Loaded data present in Sales\_Info.csv( local to db)



E)Loaded data present in Store\_Info.csv( local to db)



F)Loaded data present in Order\_Info.csv( local to db)



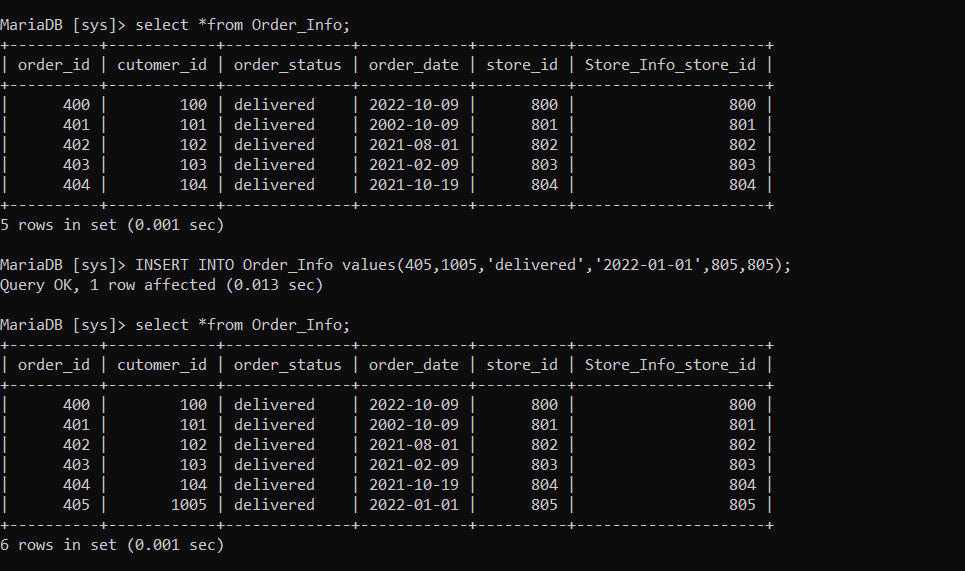
1. **DML Scripts:**
2. **Insert statements**

* Inserted a row in Order\_Info table which are related to orders data.

**Script:**

INSERT INTO Order\_Info values(405,1005,'delivered','2022-01-01',805,805);

**Output:**

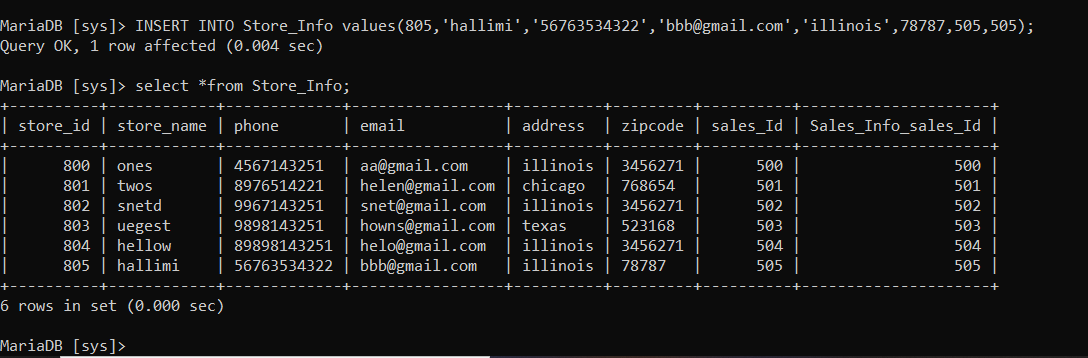


* Inserts a row in Store\_Info table related to stores information.

**Script:**

INSERT INTO Store\_Info values(805,'hallimi','56763534322','bbb@gmail.com','illinois',78787,505,505);

**Output:**



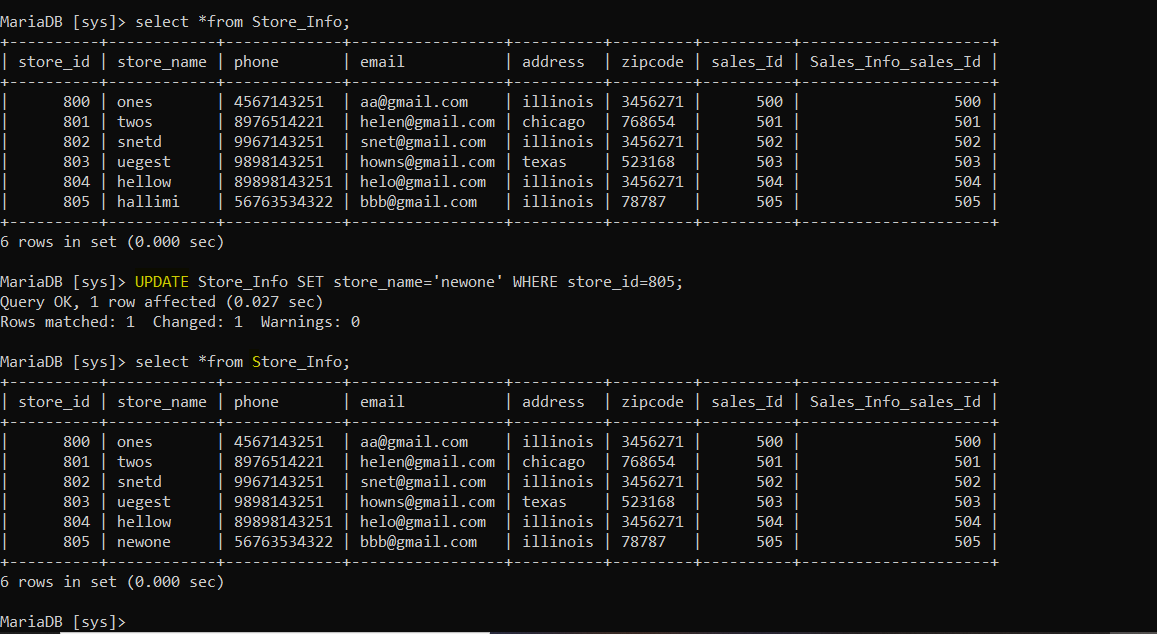
1. **Updation**

* Updates the store\_name from hallini to newone whose store\_id=805;

**Script:**

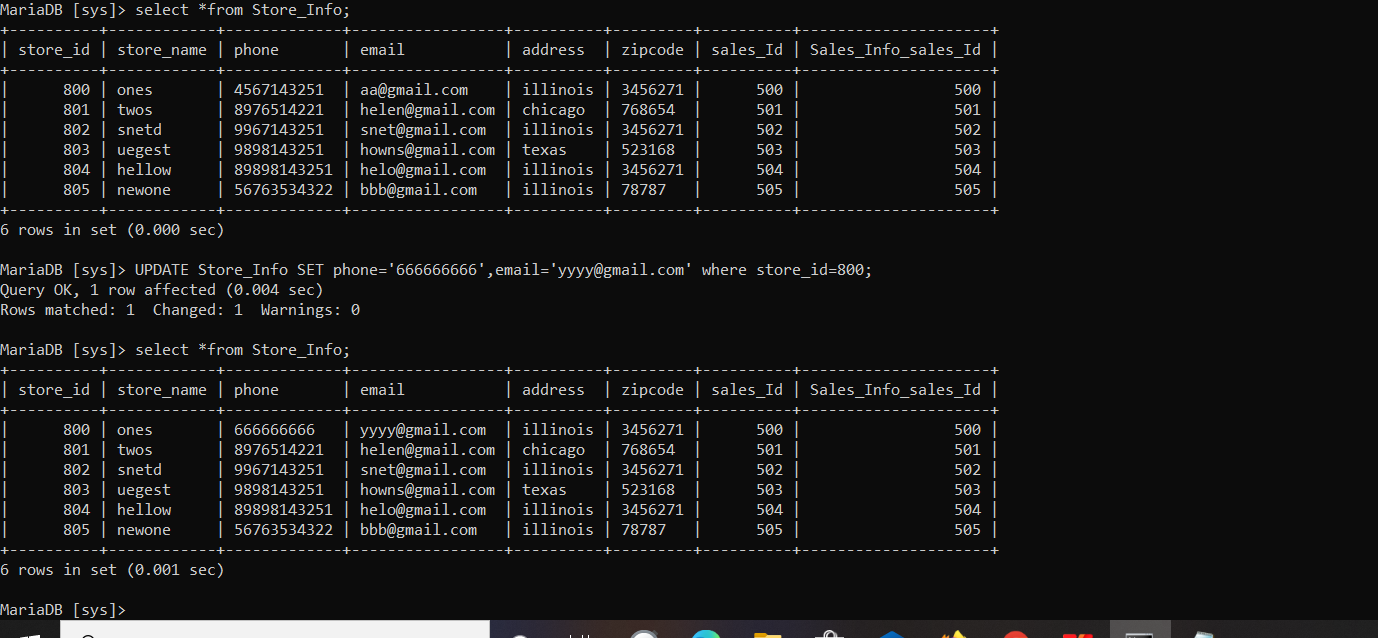
UPDATE Store\_Info SET store\_name='newone' WHERE store\_id=805;

**Output:**



* Updates the phone and email from [4567143251,aa@gmail.com](mailto:4567143251,aa@gmail.com) to 666666666, [yyyy@gmail.com](mailto:yyyy@gmail.com) whose store\_id=800;

**Script:**



**C)Deletion**

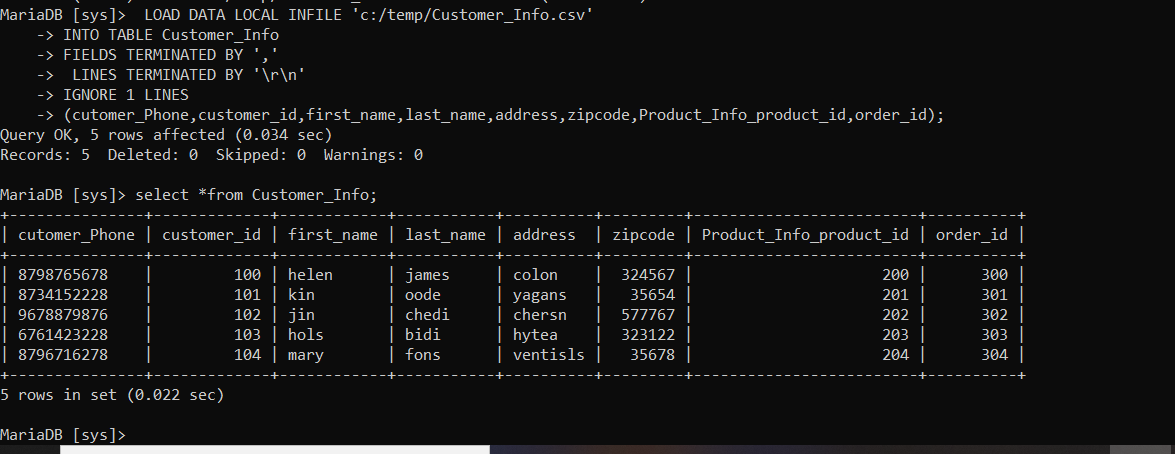
* Delete statement in customer table. It deleted the record whose customer\_id=101;

**Script:**

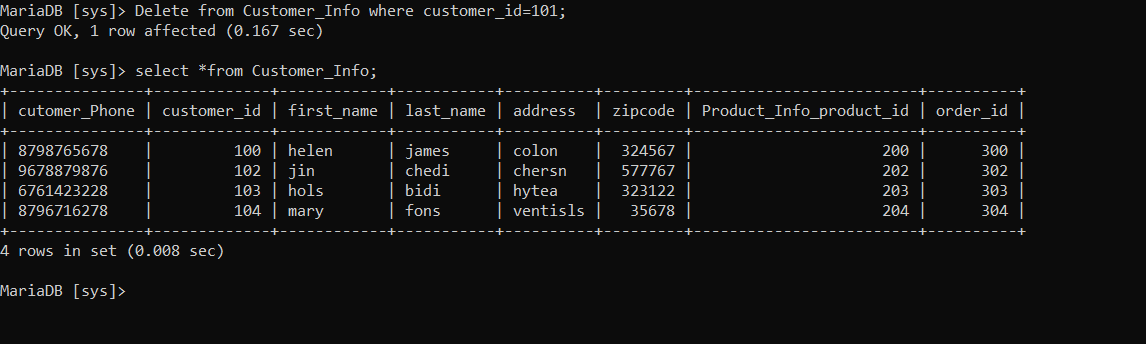
Delete from Customer\_Info where customer\_id=101;

**Output:**

Before:



After:



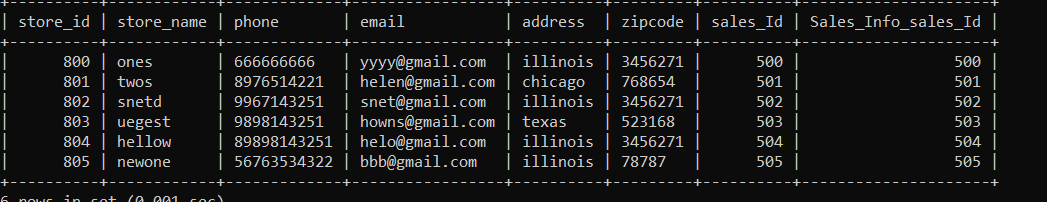
* Delete statement in store\_Infotable. It deleted the record whose store Id is in 801,802;

**Script:**

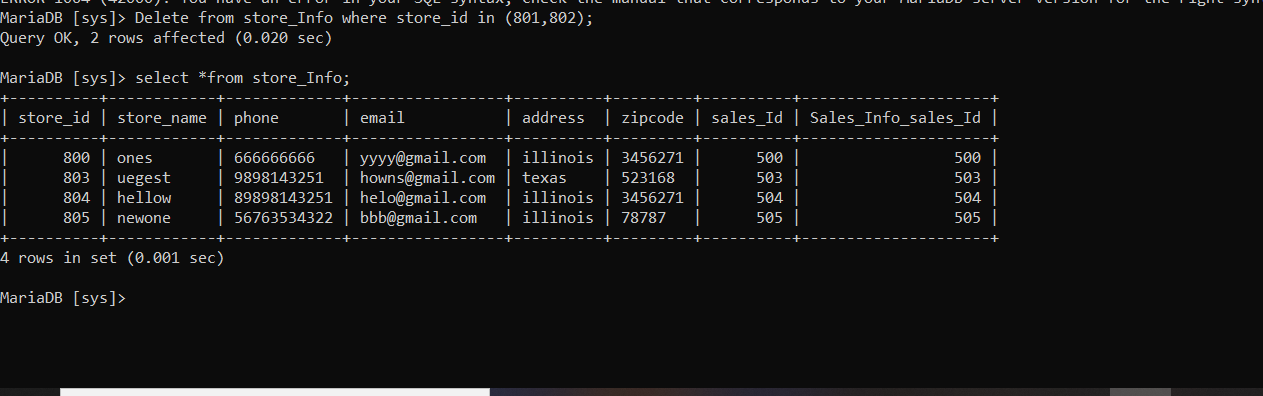
Delete from store\_Info where store\_id in (801,802);

**Output:**

Before:



After:



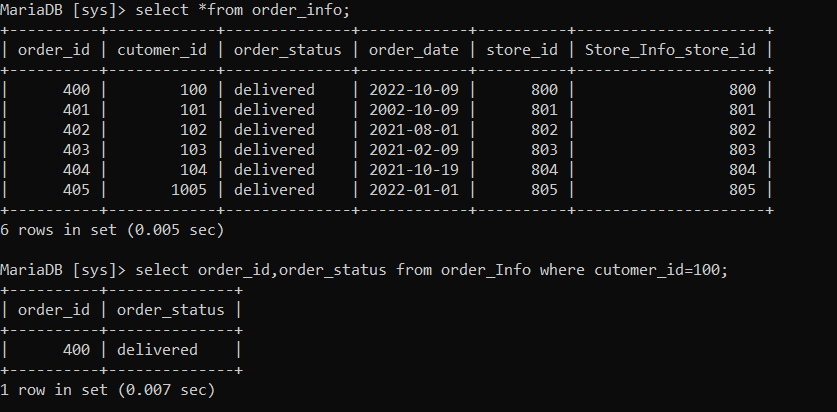
1. **Select statement**

* Selects the order\_id,order\_status of a customer having customer\_id=100;

**Script:**

Select order\_id,order\_status from order\_info where customer\_id=100;

**Output:**



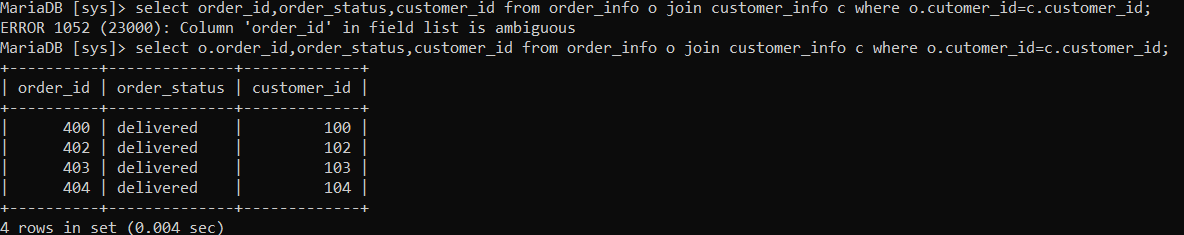
1. **Join statements**

* Used to get the columns ( order\_info,customer\_info ) after joining two different table.

**Script:**

select o.order\_id,order\_status,customer\_id from order\_info o join customer\_info c where o.cutomer\_id=c.customer\_id;

**Output:**

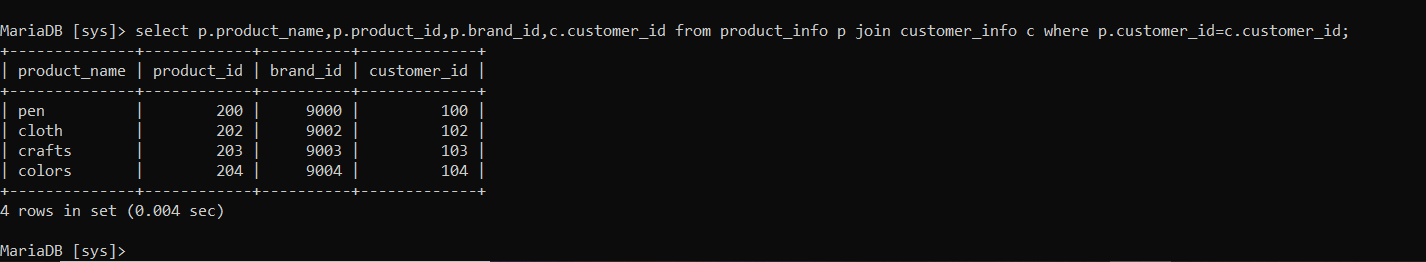


* Used to get the columns after joining product\_info,customer\_info table

**Script:**

select p.product\_name,p.product\_id,p.brand\_id,c.customer\_id from product\_info p join customer\_info c where p.customer\_id=c.customer\_id;

**Output:**



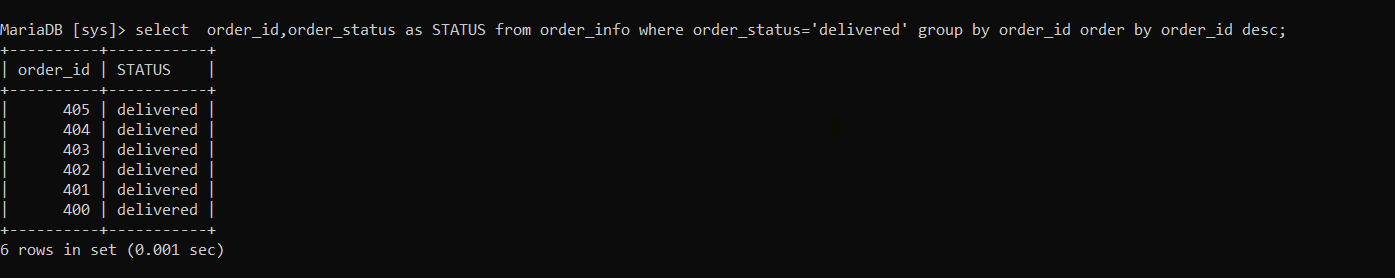
1. **Summary statements**

* Summary Query to get based on condition on table in descending order.

**Script:**

select order\_id,order\_status as STATUS from order\_info where order\_status='delivered' group by order\_id order by order\_id desc;

**Output:**

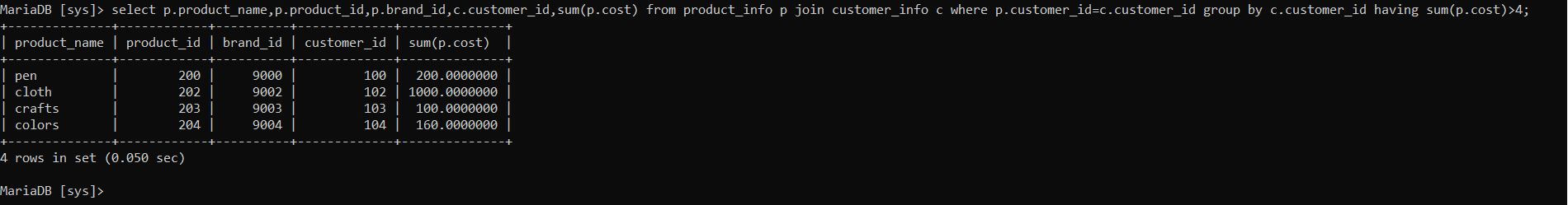


* Summary query to get information of products,customers

**Script:**

select p.product\_name,p.product\_id,p.brand\_id,c.customer\_id,sum(p.cost) from product\_info p join customer\_info c where p.customer\_id=c.customer\_id group by c.customer\_id having sum(p.cost)>4;

Output:

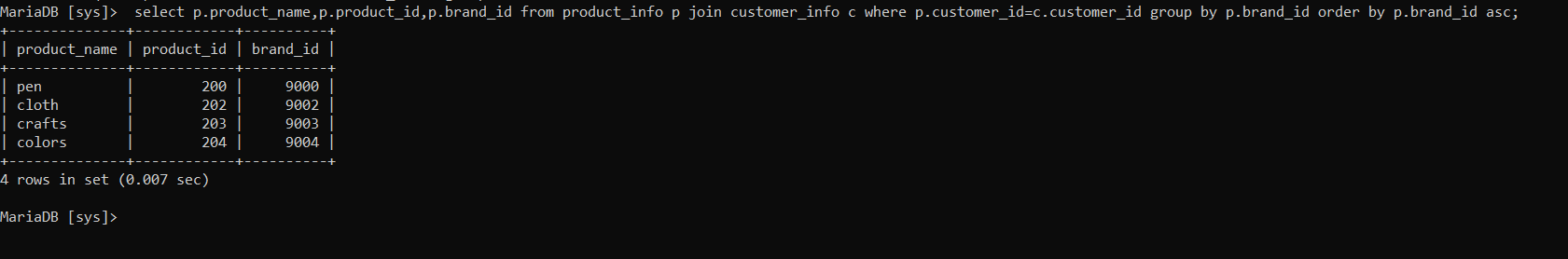


* Summary query to get information of products,customers in ascending order.

**Script:**

select p.product\_name,p.product\_id,p.brand\_id from product\_info p join customer\_info c where p.customer\_id=c.customer\_id group by p.brand\_id order by p.brand\_id asc;

**Output:**



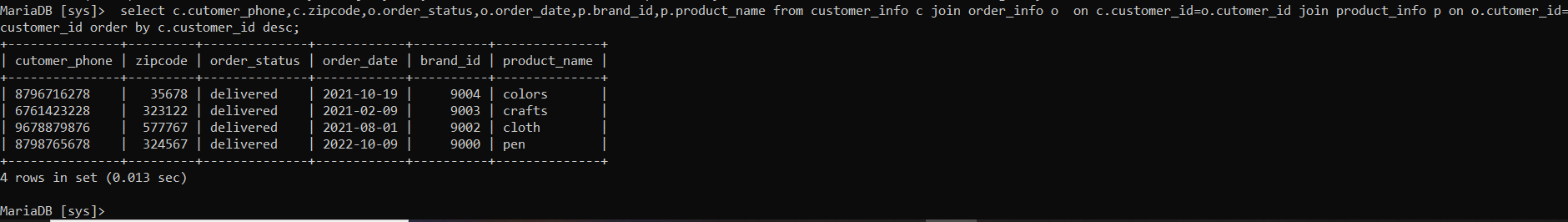
1. **Multi query**

* Query which gives required columns from 3 tables namely customer\_info,products\_info,order\_info in descending order

**Script:**

select c.cutomer\_phone,c.zipcode,o.order\_status,o.order\_date,p.brand\_id,p.product\_name from customer\_info c join order\_info o on c.customer\_id=o.cutomer\_id join product\_info p on o.cutomer\_id=p.customer\_id order by c.customer\_id desc;

**Output:**



**Overall Scripts For Reference**

CREATE TABLE Customer\_Info (

cutomer\_Phone varchar(25) NOT NULL,

customer\_id int NOT NULL,

first\_name varchar(25) NOT NULL,

last\_name varchar(25) NOT NULL,

address varchar(25) NOT NULL,

zipcode int NOT NULL,

Product\_Info\_product\_id int NOT NULL,

order\_id int NOT NULL,

CONSTRAINT Customer\_Info\_pk PRIMARY KEY (customer\_id)

);

CREATE TABLE Product\_Info (

product\_name varchar(25) NOT NULL,

product\_id int NOT NULL,

Cost float(10,7) NOT NULL,

brand\_id int NOT NULL,

Customer\_id int NOT NULL,

CONSTRAINT Product\_Info\_pk PRIMARY KEY (product\_id)

);

CREATE TABLE Supplier\_Info (

supplier\_ID int NOT NULL,

supplier\_Fname varchar(25) NOT NULL,

supplier\_Lname varchar(25) NOT NULL,

company\_name varchar(25) NOT NULL,

address varchar(25) NOT NULL,

Product\_Info\_product\_id int NOT NULL,

CONSTRAINT Supplier\_Info\_pk PRIMARY KEY (supplier\_ID)

);

CREATE TABLE Sales\_Info (

sales\_Id int NOT NULL,

Item\_sold varchar(25) NOT NULL,

date date NOT NULL,

price float(10,7) NOT NULL,

profit int NOT NULL,

Supplier\_ID int NOT NULL,

CONSTRAINT Sales\_Info\_pk PRIMARY KEY (sales\_Id)

);

CREATE TABLE Store\_Info (

store\_id int NOT NULL,

store\_name varchar(25) NOT NULL,

phone varchar(15) NOT NULL,

email varchar(25) NOT NULL,

address varchar(50) NOT NULL,

zipcode varchar(25) NOT NULL,

sales\_Id int NOT NULL,

Sales\_Info\_sales\_Id int NOT NULL,

CONSTRAINT Store\_Info\_pk PRIMARY KEY (store\_id)

);

CREATE TABLE Order\_Info (

order\_id int NOT NULL,

cutomer\_id int NOT NULL,

order\_status varchar(25) NOT NULL,

order\_date date NOT NULL,

store\_id int NOT NULL,

Store\_Info\_store\_id int NOT NULL,

CONSTRAINT Order\_Info\_pk PRIMARY KEY (order\_id)

);

LOAD DATA LOCAL INFILE 'c:/temp/Customer\_Info.csv'

-> INTO TABLE Customer\_Info

-> FIELDS TERMINATED BY ','

-> LINES TERMINATED BY '\r\n'

-> IGNORE 1 LINES

-> (cutomer\_Phone,customer\_id,first\_name,last\_name,address,zipcode,Product\_Info\_product\_id,order\_id);

LOAD DATA LOCAL INFILE 'c:/temp/Product\_Info.csv'

-> INTO TABLE Product\_Info

-> FIELDS TERMINATED BY ','

-> LINES TERMINATED BY '\r\n'

-> IGNORE 1 LINES

-> (product\_name,product\_id,Cost,brand\_id,customer\_id);

LOAD DATA LOCAL INFILE 'c:/temp/Supplier\_Info.csv'

-> INTO TABLE Supplier\_Info

-> FIELDS TERMINATED BY ','

-> LINES TERMINATED BY '\r\n'

-> IGNORE 1 LINES

-> (supplier\_ID,supplier\_Fname,supplier\_Lname,company\_name,address,Product\_Info\_product\_id);

LOAD DATA LOCAL INFILE 'c:/temp/Sales\_Info.csv'

-> INTO TABLE Sales\_Info

-> FIELDS TERMINATED BY ','

-> LINES TERMINATED BY '\r\n'

-> IGNORE 1 LINES

-> (sales\_Id,Item\_sold, date,price,profit,Supplier\_ID);

LOAD DATA LOCAL INFILE 'c:/temp/Store\_Info.csv'

-> INTO TABLE Store\_Info

-> FIELDS TERMINATED BY ','

-> LINES TERMINATED BY '\r\n'

-> IGNORE 1 LINES

-> (store\_id,store\_name,phone,email,address,zipcode,sales\_Id,Sales\_Info\_sales\_Id);

LOAD DATA LOCAL INFILE 'c:/temp/Order\_Info.csv'

-> INTO TABLE Order\_Info

-> FIELDS TERMINATED BY ','

-> LINES TERMINATED BY '\r\n'

-> IGNORE 1 LINES

-> (order\_id,cutomer\_id,order\_status,order\_date,store\_id,store\_info\_store\_id);

INSERT INTO Order\_Info values(405,1005,'delivered','2022-01-01',805,805);

INSERT INTO Store\_Info values(805,'hallimi','56763534322','bbb@gmail.com','illinois',78787,505,505);

UPDATE Store\_Info SET store\_name='newone' WHERE store\_id=805;

Updates the phone and email from 4567143251,aa@gmail.com to 666666666, yyyy@gmail.com whose store\_id=800;

Delete from Customer\_Info where customer\_id=101;

Delete from store\_Info where store\_id in (801,802);

Select order\_id,order\_status from order\_info where customer\_id=100;

select o.order\_id,order\_status,customer\_id from order\_info o join customer\_info c where o.cutomer\_id=c.customer\_id;

select p.product\_name,p.product\_id,p.brand\_id,c.customer\_id from product\_info p join customer\_info c where p.customer\_id=c.customer\_id;

select order\_id,order\_status as STATUS from order\_info where order\_status='delivered' group by order\_id order by order\_id desc;

select p.product\_name,p.product\_id,p.brand\_id,c.customer\_id,sum(p.cost) from product\_info p join customer\_info c where p.customer\_id=c.customer\_id group by c.customer\_id having sum(p.cost)>4;

select p.product\_name,p.product\_id,p.brand\_id from product\_info p join customer\_info c where p.customer\_id=c.customer\_id group by p.brand\_id order by p.brand\_id asc;

select c.cutomer\_phone,c.zipcode,o.order\_status,o.order\_date,p.brand\_id,p.product\_name from customer\_info c join order\_info o on c.customer\_id=o.cutomer\_id join product\_info p on o.cutomer\_id=p.customer\_id order by c.customer\_id desc;