islington college
(इस्लिङ्टन कलेज)

LONDON METROPOLITAN UNIVERSITY

Module Code & Module Title

CC4057NI Introduction to Information Systems

Assessment Weightage & Type

30% Individual Coursework

Year and Semester

2019-20 Autumn

Student Name: Himanshu Pandey

Group: N7

London Met ID: NP01NT4A190141

College ID: NP01NT4A190131

Assignment Due Date:28th NOVEMBER 2019

Assignment Submission Date:20th DECEMBER 2019

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Contents

Table of contents

.

List of Figures

Figure 1 Entity relationship diagram 8
Figure 2 Relational diagram 10
Figure 3 desc executive 11
Figure 4select from executive 11
Figure 5desc workers 12
Figure 6insert and select from workers 12
Figure 7 create customer 13
Figure 8 desc customer 13
Figure 9select from customer 13
Figure 10 create orders 14
Figure 11 desc order 14
Figure 12 select from orders 14
Figure 13create items 15
Figure 14 desc items 15
Figure 15insert items 15

Introduction

For the second assignment of Introduction to information system, we have been given a coursework to create database about an organization or company of our own choice. Following databases contain at least five relation tables with five different values for each table and five different relationship with each other and each relation must identified using a suitable primary key and the relation must contains a suitable attributes. For the attributes it constrained using suitable constraints (ie; unique, not null , auto increment, etc.) and the relation must be interlinked using suitable pairing of foreign keys. Here I have created a database of a shopping store and the shopping store start in a manner from Executive till to the bills. I have also created Entity Relation Diagram (ERD) with some research and interaction with teacher and friends. All together I've made six entities that are involved in this shopping store, they are as below.

1. Executive
2. Workers
3. Customer
4. Orders
5. Items
6. Bills

1) Executive: Executive are those management system of an organization or a company and in same manner my assume of executive contains five attributes like as name, phone no, executive id, address and email. In the attributes name store name of executive likewise phone no saves the phone number of executive person to call when required similarly, executive id is the id which helps to identify the executive person and lastly email attributes contains the email address of following executive.

2) Workers: Workers are those persons who works under the control of executive of any organization .so in the workers entity there are five attributes they are worker id ,address , name, contact number and email. An attributes workers id of entity workers stores the unique id of a workers which is a primary key and the attributes address stores the workers address likewise, name stores the name of worker in the column ,contact number stores the phone number of workers in the column similarly emails stores the email address of worker so that we can mail the executive when required. Executive id is an attributes of a workers which is a foreign key of the table executive which can identify the executive of shopping store.

3) Customer: Normally customer are those peoples of any organization who buys goods or items. So in this entity there are five attributes which are customer id,

4) name, address , workers id , contact no and email. The attribute customer id of entity customer stores unique id of customer also called as primary key , the attributes stores address of customer, likewise contact no stores the number of customers , the attributes email stores the email of customer so as to mail the customer when there is any mistake made by workers while sending the goods order by customers. The attributes worker id acts as foreign key of table workers which can identify the workers of the shopping store.

5) Orders: Orders are those which has benn ordered by customers to purchase their needy item in the shopping store. This entity contains total five attributes like order id , order code, name, date, quantity. The attributes order id contains the id of ordered items . the attributes order code stores the code of ordered items likewise, name attributes stores the name of goods ordered by customers and date attributes stores the date of ordered item on the date where customers ordered the goods. finally the quantity attributes strores the quantity of ordered item or how many goods items does the customed ordered .

6) Items: Items are those sold things which is burrowed or purchased from the orders given by customer in the store. This entity of items consists of five attributes like item name, item price, item id, ,quantity, customer id. The attributes item id is a unique id of a item which is a primary key. And the attributes item name stores name of items, similarly price stores the price of each items likewise, quantity determines the amount of items ordered and the customer id is an foreign key of table customer which can identify the item of shopping store.

7) Bills: It can be defined as a receipt or statement of the bill generally which are provided whenever we make any purchase either online or in physical presence of customer. This entity contains five attributes like bill id , price, amount, quantity, customer id .An attributes bill id of entity bill stores unique bill which acts as primary key. The attributes amount stores the amount of purchased items . The attributes quantity stores the quantity of an bill in the bill column. Likewise, price attributes stores price of bill item in the bill column. Customer id is an attribute of a bill which is foreign key of table customer which can identify the bill of shopping store.

INFORMATION SYSTEM

Information system is a set of components for collecting, creating, storing, processing the data and distributing useful information that can be used for decision making in an organization. Typically, Information system includes hardware, software and the data itself.

Hardware: It contains all the physical components where we can see and touch. Example; keyboards, iPads, Disk drives etc. Software: It is the set of instructions that tells the hardware what to do.it cannot be touched.

Data: It is defined as the collection of raw information or facts or figure. Data can be anything like phone numbers, address etc.

Some of the types of information system are as follows;

A).General purpose information system: let's take an example a database management system (DBMS) is a combination of software and data that makes it possible and analyze data. DBMS is typically not designed to work with specific organization or specific type of analysis. Rather, it is a general purpose information system.

B).Specialized information system : In contrast , there are a number of specialized information system that have been specifically designed to carry out very specific tasks For example; Enterprise resource planning(ERP) is an information system used to integrate all the internal and external information across an entire organization. (zwass, n.d.)

DATABASES

Database is a collection of information that is organized so that it can be easily accessed, managed and updated. Most databases use Structured Query Language (SQL) for writing and querying data. Example: SQL server, My SQL etc. Database management system plays vital role to the operation of different organizations because they help to manage an organization in various databases. This system allows users to easily retrieve, update and generally manage data relevant to a business operation. A database can track sales, expences and other financial information. Examples might be a shop's stock inventory or airline booking system. (oracle, n.d.)

ERD DIAGRAM OF SHOPPING STORE

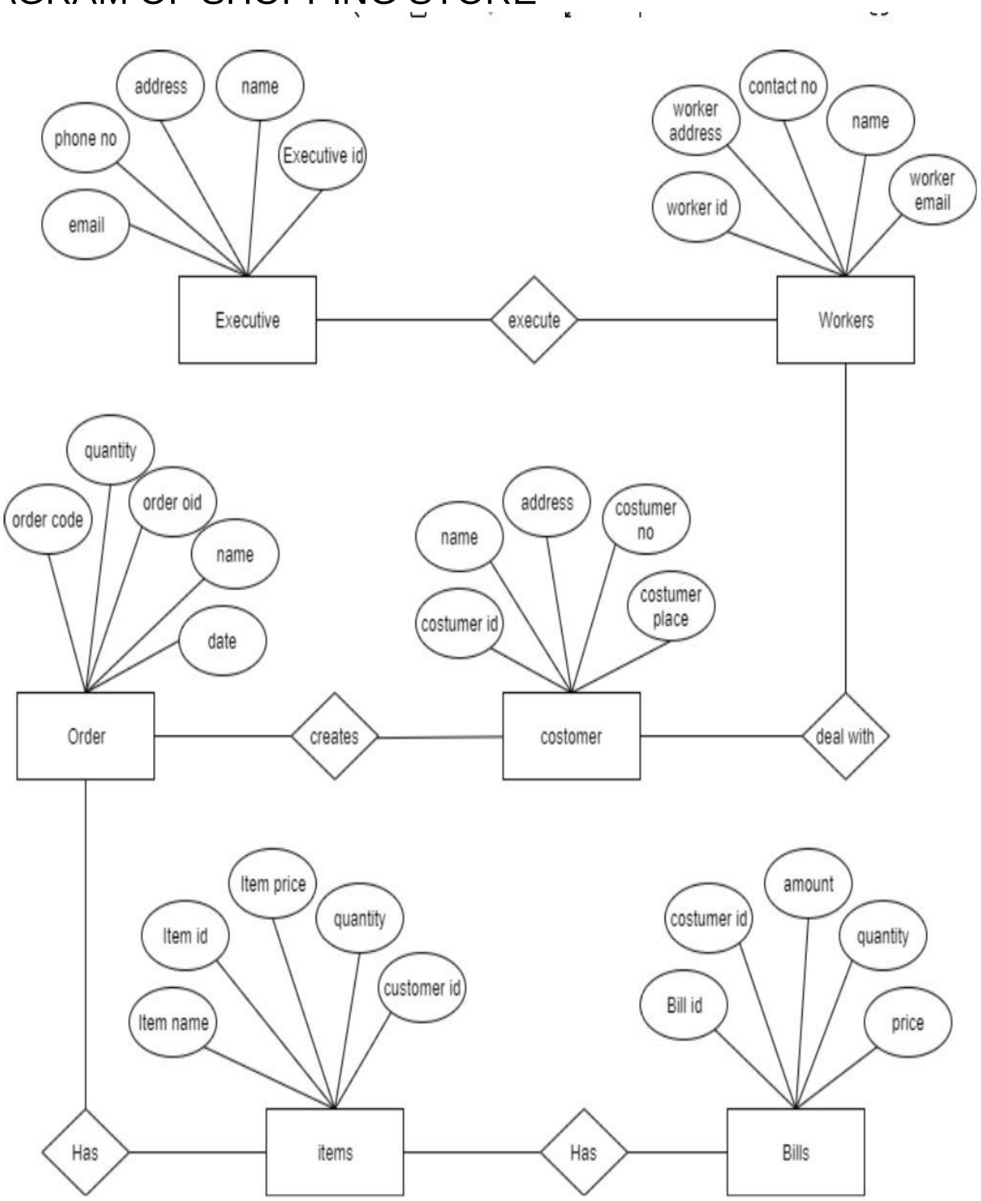


Figure 1 Entity relationship diagram

RELATIONAL DATABASES

Relational diagram,it refers to a visual representation of an relational database entity,relational diagram relates the entities and the attributes with in those entity. A relational database (RDB) is a collective set of multiple data sets organized by tables, records and columns. Relational databasess establish a well-defined relationship between database tables. Tables communicate and share information, which facilitates data searchability, organization and reporting. Relational databases organize data in different ways. Each table is known as a relation, which contains one or more data category columns. Each table record (or row) contains a unique data instance defined for a corresponding column category. One or more data or record characteristics relate to one or many records to form functional dependencies. These are classified as follows:

1) One-to-One Relation
2) One-to-Many Relation
3) Many-to-One Relation
4) Many-to-Many Relation

DEFINATION

- One to One: One table record relates to another record in another table.
- One to Many: One table record relates to many records in another table.
- Many to One: More than one table record relates to another table record.
- Many to Many: More than one table record relates to more than one record in another table.

Advantages of relational database

- Easy extendability, as new data may be added without modifying existing records. This is also known as scalability.
- New technology performance, power and flexibility with multiple data requirement capabilities.
- Data security, which is critical when data sharing is based on privacy. For example, management may share certain data privileges and access and block employees from other data, such as confidential salary or benefit information. (oracle, jun -10 -2016)

RELATIONAL DIAGRAM

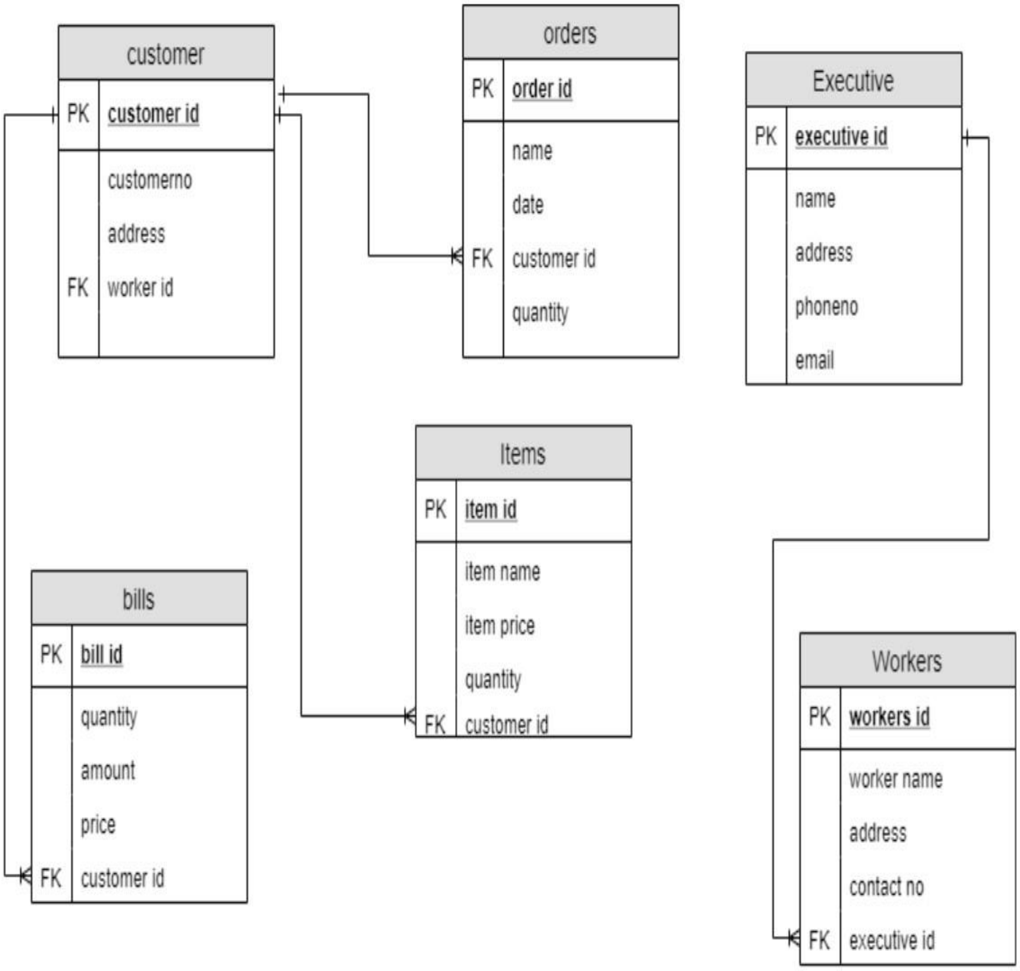


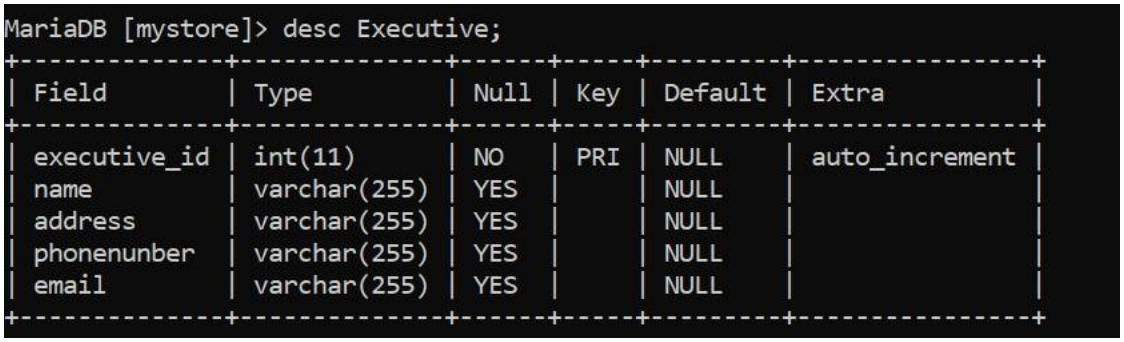
Figure 2 Relational diagram

CREATING TABLES ON DATABASES

1) Executive

Creating table of Executive:

(a) Create table Executive(executive id int primary key auto_increment not null, name varchar(255), address varchar(255), phoneno varchar(255), email varchar(255));



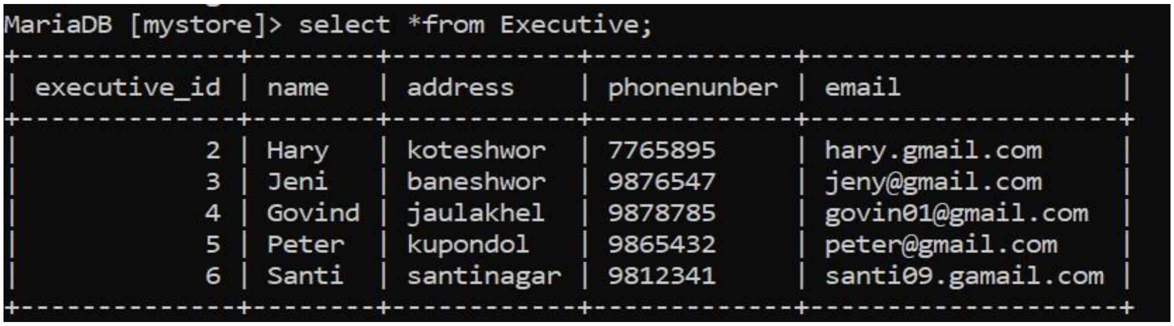
```
MariaDB [mystore]> desc Executive;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| executive_id | int(11)      | NO   | PRI | NULL    | auto_increment |
| name         | varchar(255) | YES  |     | NULL    |                |
| address      | varchar(255) | YES  |     | NULL    |                |
| phonenunber  | varchar(255) | YES  |     | NULL    |                |
| email        | varchar(255) | YES  |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
```

Figure 3 desc executive

Insert into Executive:

For eg: Insert into Executive values(2,“hary”,”koteshwor”,”7765895”,”hary.gmail,com”);
Insert into Executive values(3,“jeni”,”baneshwor”,”9876547”,”jenygmail,com”);

Select Executive:



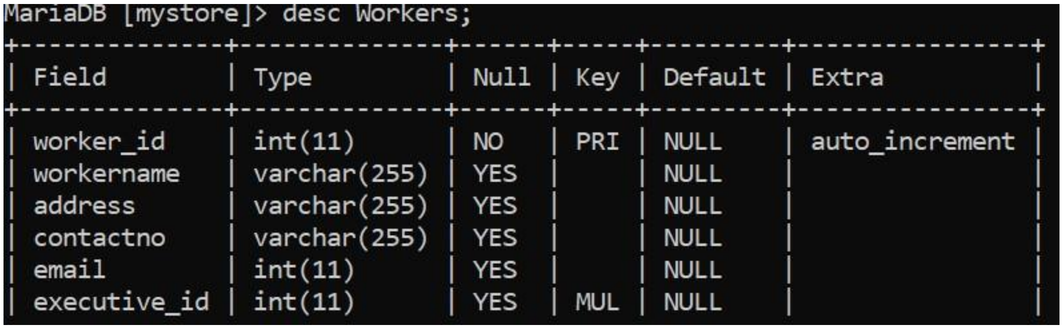
```
MariaDB [mystore]> select *from Executive;
+--------------+--------+------------+-------------+--------------------+
| executive_id | name   | address    | phonenunber | email              |
+--------------+--------+------------+-------------+--------------------+
|            2 | Hary   | koteshwor  | 7765895     | hary.gmail.com     |
|            3 | Jeni   | baneshwor  | 9876547     | jeny@gmail.com     |
|            4 | Govind | jaulakhel  | 9878785     | govin01@gmail.com  |
|            5 | Peter  | kupondol   | 9865432     | peter@gmail.com    |
|            6 | Santi  | santinagar | 9812341     | santi09.gamail.com |
+--------------+--------+------------+-------------+--------------------+
```

Figure 4select from executive

2) Workers:

Creating table of workers

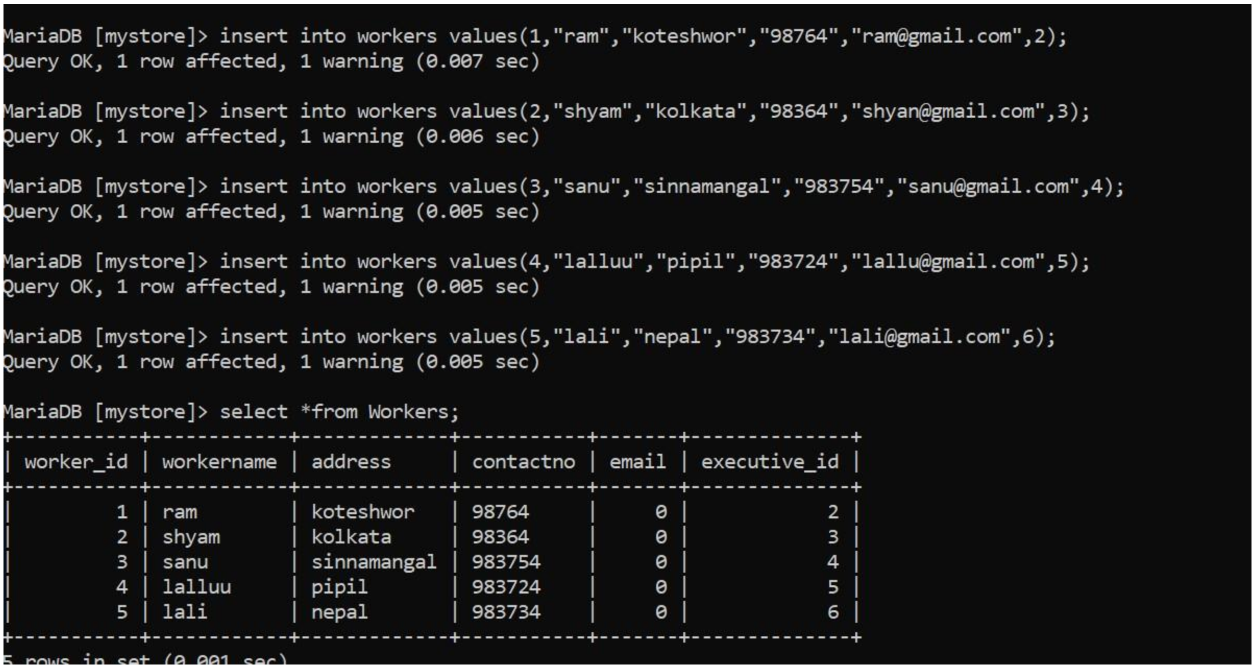
Create table Workers(worker_id int primary key auto_increment,workername varchar(255),address varchar(255),contactno int, email int, foreign key(executive_id)references Executive(executive_id));



```
MariaDB [mystore]> desc Workers;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| worker_id    | int(11)      | NO   | PRI | NULL    | auto_increment |
| workername   | varchar(255) | YES  |     | NULL    |                |
| address      | varchar(255) | YES  |     | NULL    |                |
| contactno    | varchar(255) | YES  |     | NULL    |                |
| email        | int(11)      | YES  |     | NULL    |                |
| executive_id | int(11)      | YES  | MUL | NULL    |                |
```

Figure 5desc workers

INSERT AND SELECTING WORKERS

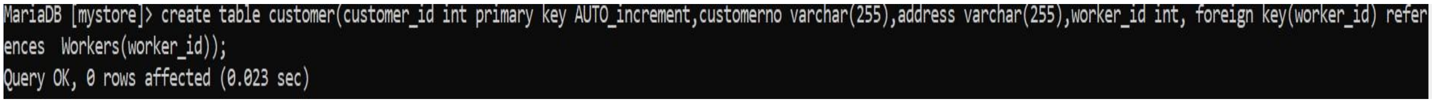


```
MariaDB [mystore]> insert into workers values(1,"ram","koteshwor","98764","ram@gmail.com",2);
Query OK, 1 row affected, 1 warning (0.007 sec)

MariaDB [mystore]> insert into workers values(2,"shyam","kolkata","98364","shyan@gmail.com",3);
Query OK, 1 row affected, 1 warning (0.006 sec)

MariaDB [mystore]> insert into workers values(3,"sanu","sinnamangal","983754","sanu@gmail.com",4);
Query OK, 1 row affected, 1 warning (0.005 sec)

MariaDB [mystore]> insert into workers values(4,"lalluu","pipil","983724","lallu@gmail.com",5);
Query OK, 1 row affected, 1 warning (0.005 sec)

MariaDB [mystore]> insert into workers values(5,"lali","nepal","983734","lali@gmail.com",6);
Query OK, 1 row affected, 1 warning (0.005 sec)

MariaDB [mystore]> select *from Workers;
+-----------+------------+-------------+-----------+-------+--------------+
| worker_id | workername | address     | contactno | email | executive_id |
+-----------+------------+-------------+-----------+-------+--------------+
|         1 | ram        | koteshwor   | 98764     |     0 |            2 |
|         2 | shyam      | kolkata     | 98364     |     0 |            3 |
|         3 | sanu       | sinnamangal | 983754    |     0 |            4 |
|         4 | lalluu     | pipil       | 983724    |     0 |            5 |
|         5 | lali       | nepal       | 983734    |     0 |            6 |
+-----------+------------+-------------+-----------+-------+--------------+
5 rows in set (0.001 sec)
```

Figure 6insert and select from workers

3) Customer:

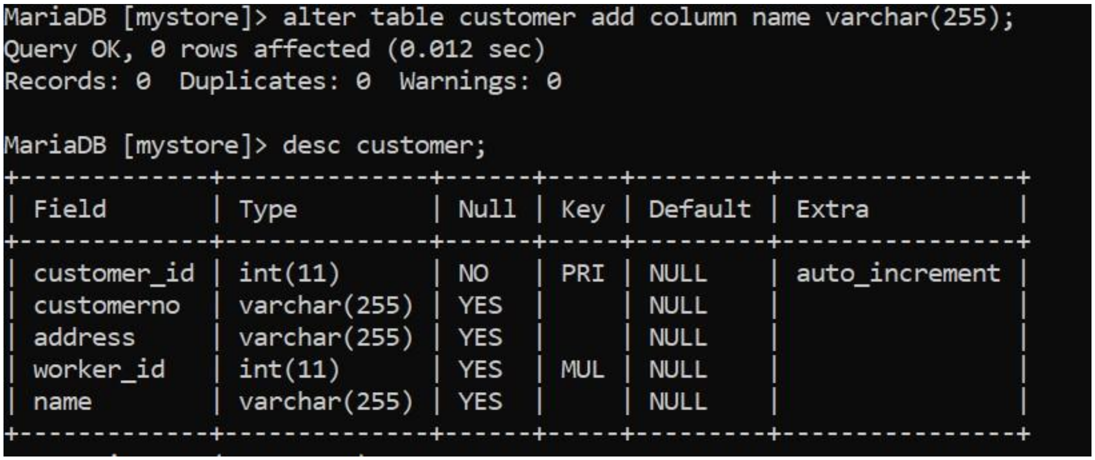
Creating customer table



```
MariaDB [mystore]> create table customer(customer_id int primary key AUTO_increment,customerno varchar(255),address varchar(255),worker_id int, foreign key(worker_id) refer
ences  Workers(worker_id));
Query OK, 0 rows affected (0.023 sec)
```

Figure 7 create customer

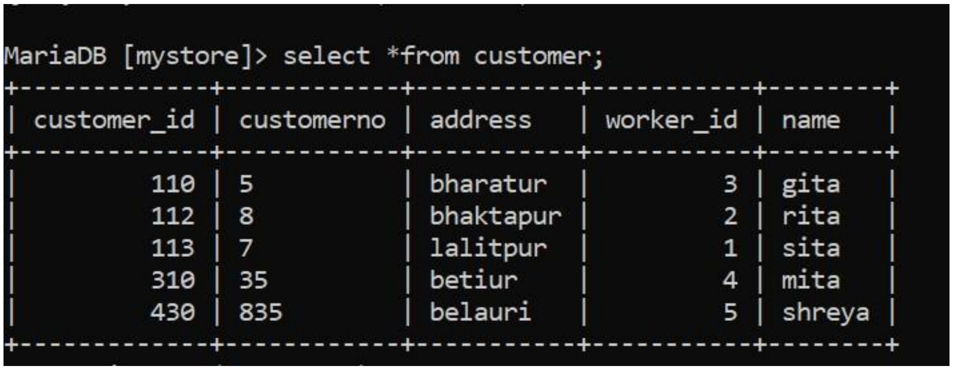
Desc customer table:



```
MariaDB [mystore]> alter table customer add column name varchar(255);
Query OK, 0 rows affected (0.012 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [mystore]> desc customer;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| customer_id | int(11)      | NO   | PRI | NULL    | auto_increment |
| customerno  | varchar(255) | YES  |     | NULL    |                |
| address     | varchar(255) | YES  |     | NULL    |                |
| worker_id   | int(11)      | YES  | MUL | NULL    |                |
| name        | varchar(255) | YES  |     | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
```

Figure 8 desc customer

Selecting customer tables:

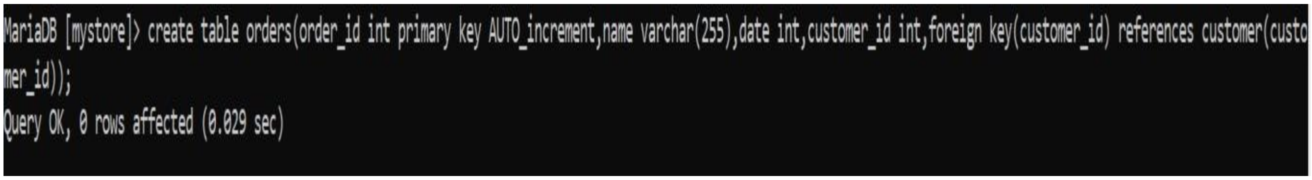


```
MariaDB [mystore]> select *from customer;
+-------------+------------+----------+-----------+--------+
| customer_id | customerno | address  | worker_id | name   |
+-------------+------------+----------+-----------+--------+
|         110 | 5          | bharatur |         3 | gita   |
|         112 | 8          | bhaktapur|         2 | rita   |
|         113 | 7          | lalitpur |         1 | sita   |
|         310 | 35         | betiur   |         4 | mita   |
|         430 | 835        | belauri  |         5 | shreya |
+-------------+------------+----------+-----------+--------+
```

Figure 9select from customer

4) Orders:

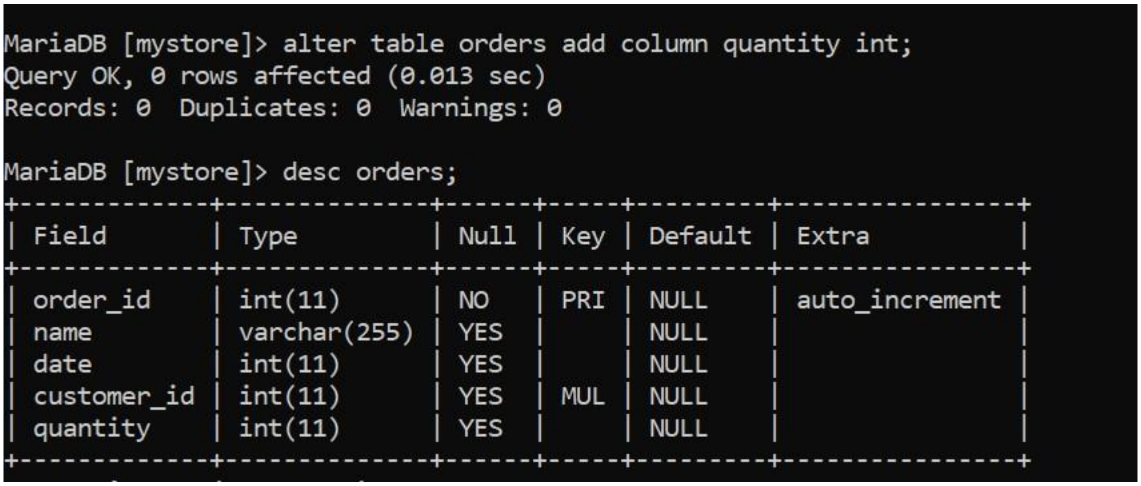
Creating orders table



```
MariaDB [mystore]> create table orders(order_id int primary key AUTO_increment,name varchar(255),date int,customer_id int,foreign key(customer_id) references customer(custo
mer_id));
Query OK, 0 rows affected (0.029 sec)
```

Figure 10 create orders

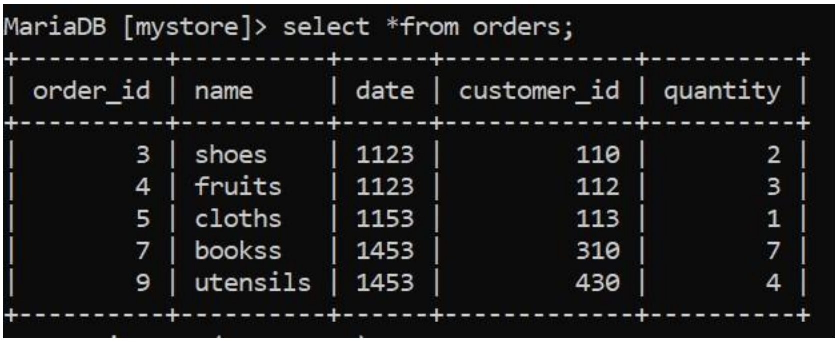
Desc orders table:



```
MariaDB [mystore]> alter table orders add column quantity int;
Query OK, 0 rows affected (0.013 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [mystore]> desc orders;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| order_id    | int(11)      | NO   | PRI | NULL    | auto_increment |
| name        | varchar(255) | YES  |     | NULL    |                |
| date        | int(11)      | YES  |     | NULL    |                |
| customer_id | int(11)      | YES  | MUL | NULL    |                |
| quantity    | int(11)      | YES  |     | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
```

Figure 11 desc order

Select from order table:

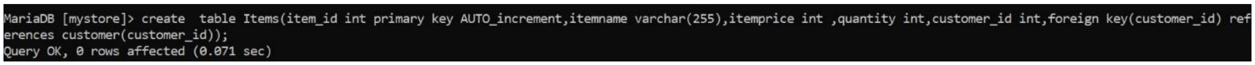


```
MariaDB [mystore]> select *from orders;
+----------+----------+------+-------------+----------+
| order_id | name     | date | customer_id | quantity |
+----------+----------+------+-------------+----------+
|        3 | shoes    | 1123 |         110 |        2 |
|        4 | fruits   | 1123 |         112 |        3 |
|        5 | cloths   | 1153 |         113 |        1 |
|        7 | bookss   | 1453 |         310 |        7 |
|        9 | utensils | 1453 |         430 |        4 |
+----------+----------+------+-------------+----------+
```

Figure 12 select from orders

5) Items:

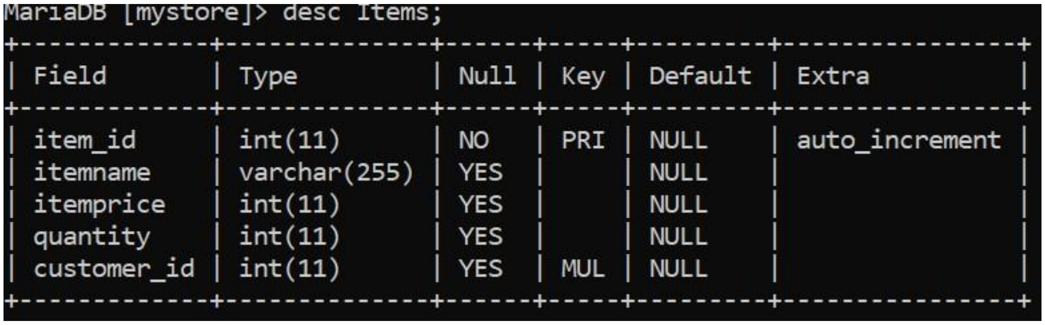
Creating items table:



```
MariaDB [mystore]> create  table Items(item_id int primary key AUTO_increment,itemname varchar(255),itemprice int ,quantity int,customer_id int,foreign key(customer_id) ref
erences customer(customer_id));
Query OK, 0 rows affected (0.071 sec)
```

Figure 13create items

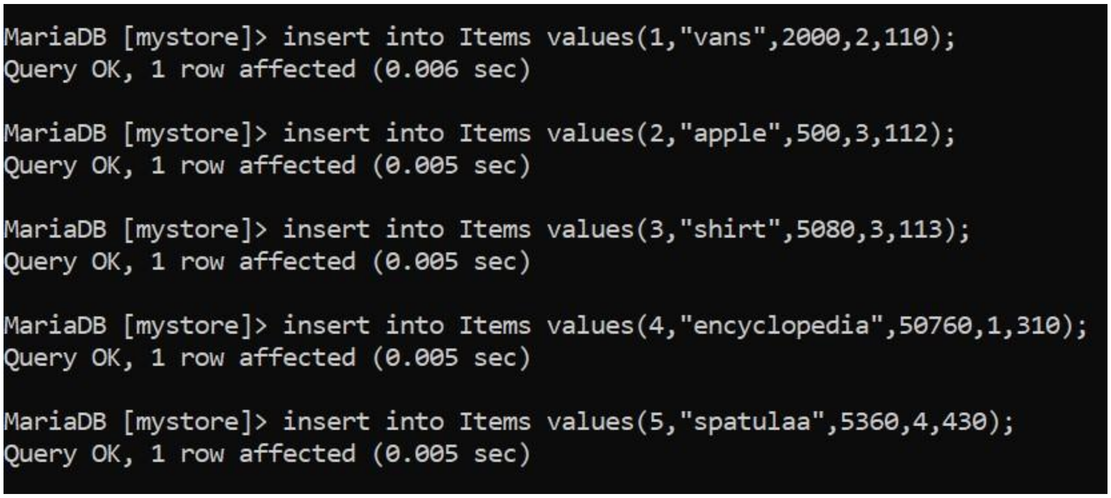
Desc items:



```
MariaDB [mystore]> desc Items;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| item_id     | int(11)      | NO   | PRI | NULL    | auto_increment |
| itemname    | varchar(255) | YES  |     | NULL    |                |
| itemprice   | int(11)      | YES  |     | NULL    |                |
| quantity    | int(11)      | YES  |     | NULL    |                |
| customer_id | int(11)      | YES  | MUL | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
```

Figure 14 desc items

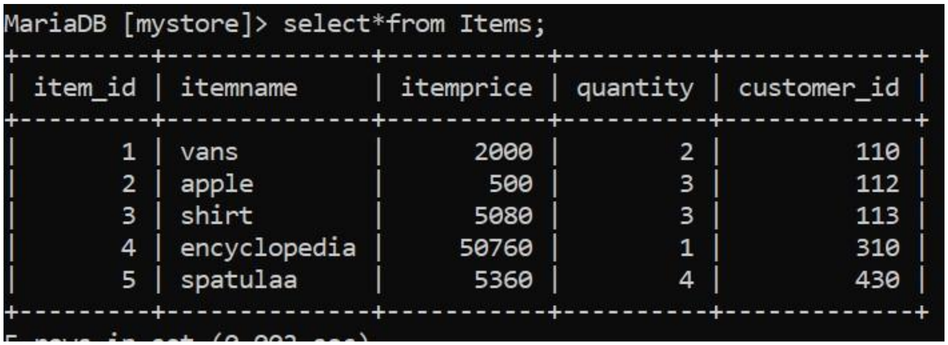
Inserting items values:



```
MariaDB [mystore]> insert into Items values(1,"vans",2000,2,110);
Query OK, 1 row affected (0.006 sec)

MariaDB [mystore]> insert into Items values(2,"apple",500,3,112);
Query OK, 1 row affected (0.005 sec)

MariaDB [mystore]> insert into Items values(3,"shirt",5080,3,113);
Query OK, 1 row affected (0.005 sec)

MariaDB [mystore]> insert into Items values(4,"encyclopedia",50760,1,310);
Query OK, 1 row affected (0.005 sec)

MariaDB [mystore]> insert into Items values(5,"spatulaa",5360,4,430);
Query OK, 1 row affected (0.005 sec)
```

Figure 15insert items

Select items table:

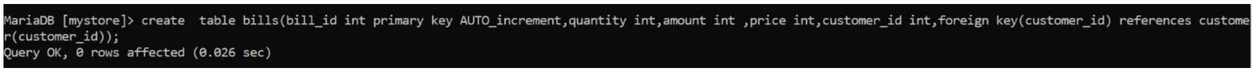


```
MariaDB [mystore]> select*from Items;
+---------+--------------+-----------+----------+-------------+
| item_id | itemname     | itemprice | quantity | customer_id |
+---------+--------------+-----------+----------+-------------+
|       1 | vans         |      2000 |        2 |         110 |
|       2 | apple        |       500 |        3 |         112 |
|       3 | shirt        |      5080 |        3 |         113 |
|       4 | encyclopedia |     50760 |        1 |         310 |
|       5 | spatulaa     |      5360 |        4 |         430 |
+---------+--------------+-----------+----------+-------------+
```

Figure 16 select items

6) Bills:

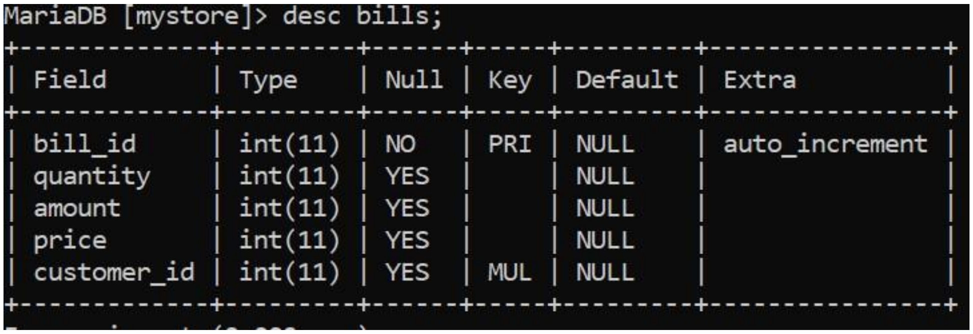
Create bills:



```
MariaDB [mystore]> create  table bills(bill_id int primary key AUTO_increment,quantity int,amount int ,price int,customer_id int,foreign key(customer_id) references custome
r(customer_id));
Query OK, 0 rows affected (0.026 sec)
```

Figure 17create bill

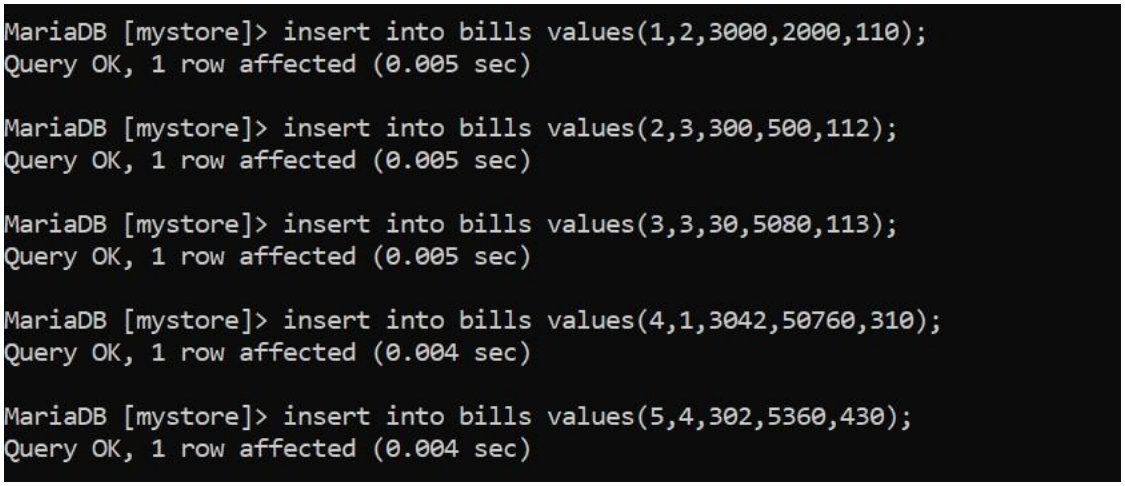
Desc bills



```
MariaDB [mystore]> desc bills;
+-------------+---------+------+-----+---------+----------------+
| Field       | Type    | Null | Key | Default | Extra          |
+-------------+---------+------+-----+---------+----------------+
| bill_id     | int(11) | NO   | PRI | NULL    | auto_increment |
| quantity    | int(11) | YES  |     | NULL    |                |
| amount      | int(11) | YES  |     | NULL    |                |
| price       | int(11) | YES  |     | NULL    |                |
| customer_id | int(11) | YES  | MUL | NULL    |                |
+-------------+---------+------+-----+---------+----------------+
```

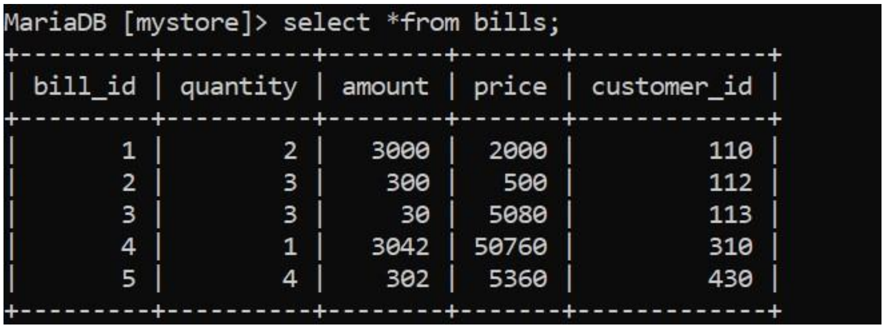
Figure 18desc bill

Inserting and selecting bill in tables



```
MariaDB [mystore]> insert into bills values(1,2,3000,2000,110);
Query OK, 1 row affected (0.005 sec)

MariaDB [mystore]> insert into bills values(2,3,300,500,112);
Query OK, 1 row affected (0.005 sec)

MariaDB [mystore]> insert into bills values(3,3,30,5080,113);
Query OK, 1 row affected (0.005 sec)

MariaDB [mystore]> insert into bills values(4,1,3042,50760,310);
Query OK, 1 row affected (0.004 sec)

MariaDB [mystore]> insert into bills values(5,4,302,5360,430);
Query OK, 1 row affected (0.004 sec)
```

Figure 19insert bill



```
MariaDB [mystore]> select *from bills;
+---------+----------+--------+-------+-------------+
| bill_id | quantity | amount | price | customer_id |
+---------+----------+--------+-------+-------------+
|       1 |        2 |   3000 |  2000 |         110 |
|       2 |        3 |    300 |   500 |         112 |
|       3 |        3 |     30 |  5080 |         113 |
|       4 |        1 |   3042 | 50760 |         310 |
|       5 |        4 |    302 |  5360 |         430 |
+---------+----------+--------+-------+-------------+
```

Figure 20select bill

DATA DICTIONARY

Data dictionary is a table with data elements (columns) as rows and their attributes as columns. Specific attributes vary depending on the purpose of the data dictionary.

Data dictionary has 2 essential elements:

1. List of tables (or entities)
2. List of columns (or fields, or attributes)

Relational database engines enable much more description of data models and provide this information through their data dictionaries. This information is:

- Data type of column,
- Default values for columns,
- Nullability of columns,
- Table relationships (foreign keys),
- Uniqueness of column values (primary and unique keys),
- Data elements descriptions (Anon., n.d.)

-

Here,I have made a data dictionary of the following above entities which I have used in above databases like as executive, workers, customers, orders, items and bills. These entities contain their own different attributes by describing their entity, column name, column description, data type length likewise, primary key, foreign key
Nullable, unique and notes of their entity.

1) Executive

Entity name	Entity description	Column name	Column description	Data type	Length	Primary key	Foreign key	Nullable	Unique	notes
Executive	A executive is a person who executes the material in the shopping mall.	Executive id	This column store id of executive person for unique identification of each executive.	int		True	false	false	True	Auto increment increases the number of executive id automatically .
		Name	This column stores the name of the executive	varchar	255	False	false	False	False	
		address	This column stores address of executive	varchar	255	False	False	False	False	
		Phone no	This column stores phone number of executive	varchar	255	False	False	False	False	
		email	This column stores email of executive	varchar	255	False	false	false	false	It is references of executive if from executive table in workers column

Figure 21data dictionary of executive

2) Worker:

Entity name	Entity description	Column name	Column description	Data type	Length	Primary key	Foreign key	Nullable	Unique	notes
Workers	A worker is someone who deals with the customer	Worker id	This column store id of worker person for unique identification of each workers.	int		true	False	False	true	Auto increment increases the number of executive id automatically.
		Name	This column stores the name of the workers	varchar	255	False	False	False	False	
		address	This column stores address of workers	varchar	255	False	False	False	False	
		Phone no	This column stores phone number of workers	varchar	255	False	false	False	False	
		email	This column stores email of workers	varchar		false	true	False	False	It is references of workers id from worker table in customer colum

Figure 22data dictionary workers

3) Customers

Entity name	Entity description	Column name	Column description	Data type	Length	Primary key	Foreign key	Nullable	Unique	notes
customers	A customer is a one who makes different order at shopping mall	Customer id	This column stores the id of customer	int		True	false	False	True	Auto increment increases the number of customer id automatically
		Customer no	This column stores no of customer in it.	varchar	255	False	False	False	False	
		Name	This column stores the name of customeri	varchar	255	False	False	False	False	
		Address	This column stores the address of customer	varchar	255	False	False	False	False	
		Workers id	This column stores id of workers in the customer tables	int		false	True	False	false	It is references of customer id from orders table in workers column

Figure 23datadictionary customer

4) Orders:

Entity name	Entity description	Column name	Column description	Data type	Length	Primary key	Foreign key	Nullable	Unique	notes
Orders	Order is something where customer gives order to the shopping mall	Order id	This column stores order in it.	int		True	false	False	True	An auto increment increases the number of order id automatically
		Name	This column stores name of order.	varchar	255	False	false	False	False	
		Date	This column stores the date of order.	int		False	False	False	False	
		Quantity	This column stores the quantity of orders.	int		False	False	False	False	
		Customer id	This column stores id of customer in order table.	int		false	True	False	False	It is references of customer id from orders table in workers column

Figure 24data dictionary orders

5) Items:

Entity name	Entity description	Column name	Column description	Data type	Length	Primary key	Foreign key	Nullable	Unique	notes
Items	An item is one who has the bill of items in the shopping store	Item id	This column store id of item in each item table.	int		True	False	False	true	Auto increment increases the number of item id automatically.
		Item name	This column stores the name of the items	varchar	255	False	False	False	False	
		Item price	This column price of items	int		False	False	False	False	
		quantity	This column stores quantity of item in table	int		False	False	False	False	
		Customer id	This column stores id of customer in the table	int		False	True	False	False	It is references of customer id from customer table in the item column

Figure 25data dictionary of item column

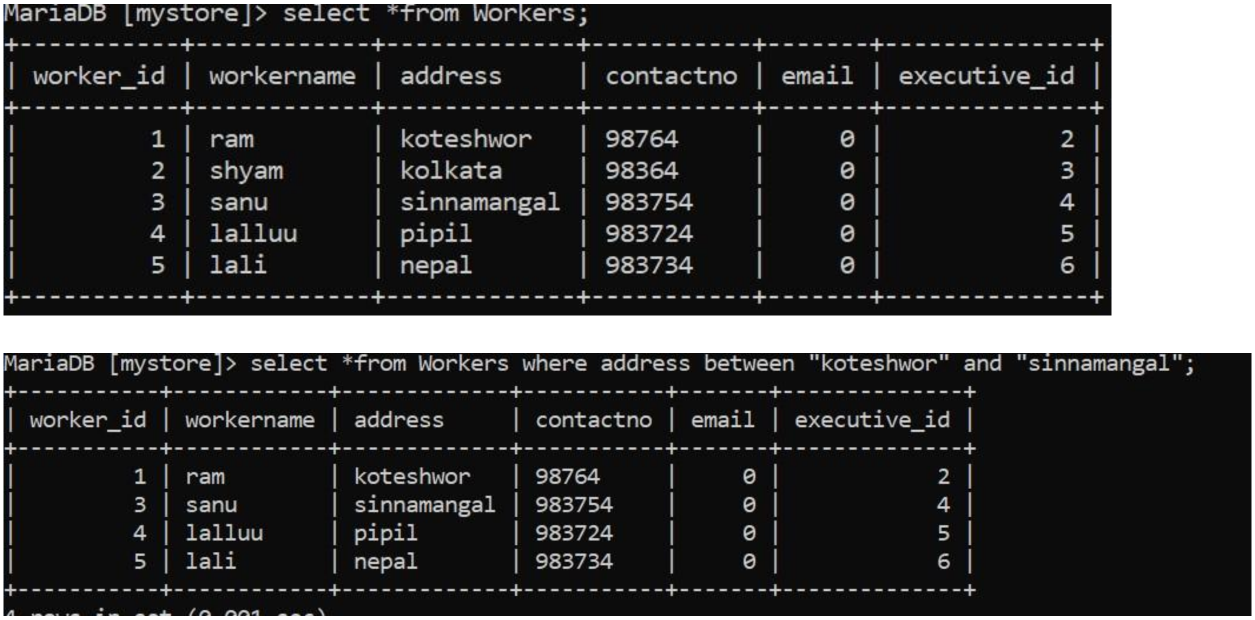
6) Bills:

Entity name Executive	Entity description	Column name	Column description	Data type	Length	Primary key	Foreign key	Nullable	Unique	notes
Bills	A bill is the statement of the order or items	Bill id	This column store id of bill in the table.	int		true	False	False	True	Auto increment increases the number of bill id automatically.
		Price	This column stores the price of the item	int		False	False	False	False	
		quantity	This column stores quantity of item	int		False	False	False	False	
		Item name	This column stores name of each item	varchar	255	False	False	False	False	
		Customer id	This column stores customer id from customer	int		False	True	False	False	It is references of customer id from customer table in bill column

Figure 26data dictionary of bill column

QUERIES

1. Show the details of workers whose are from koteshwor and sinnamangal .



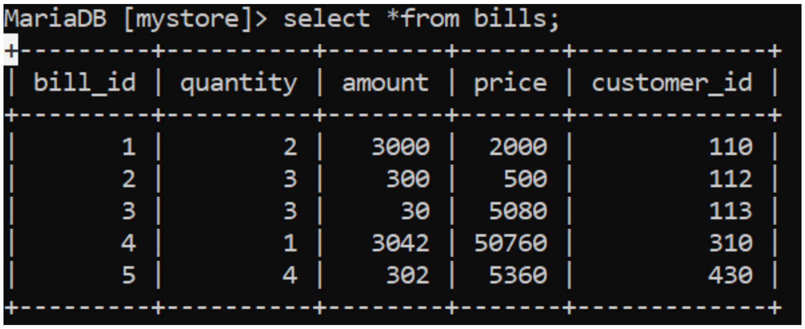
```
MariaDB [mystore]> select *from Workers;
+-----------+-----------+-------------+----------+-------+--------------+
| worker_id | workername | address    | contactno | email | executive_id |
+-----------+-----------+-------------+----------+-------+--------------+
|         1 | ram       | koteshwor   | 98764    |     0 |            2 |
|         2 | shyam     | kolkata     | 98364    |     0 |            3 |
|         3 | sanu      | sinnamangal | 983754   |     0 |            4 |
|         4 | lalluu    | pipil       | 983724   |     0 |            5 |
|         5 | lali      | nepal       | 983734   |     0 |            6 |
+-----------+-----------+-------------+----------+-------+--------------+
```

```
MariaDB [mystore]> select *from Workers where address between "koteshwor" and "sinnamangal";
+-----------+-----------+-------------+----------+-------+--------------+
| worker_id | workername | address    | contactno | email | executive_id |
+-----------+-----------+-------------+----------+-------+--------------+
|         1 | ram       | koteshwor   | 98764    |     0 |            2 |
|         3 | sanu      | sinnamangal | 983754   |     0 |            4 |
|         4 | lalluu    | pipil       | 983724   |     0 |            5 |
|         5 | lali      | nepal       | 983734   |     0 |            6 |
+-----------+-----------+-------------+----------+-------+--------------+
4 rows in set (0.001 sec)
```

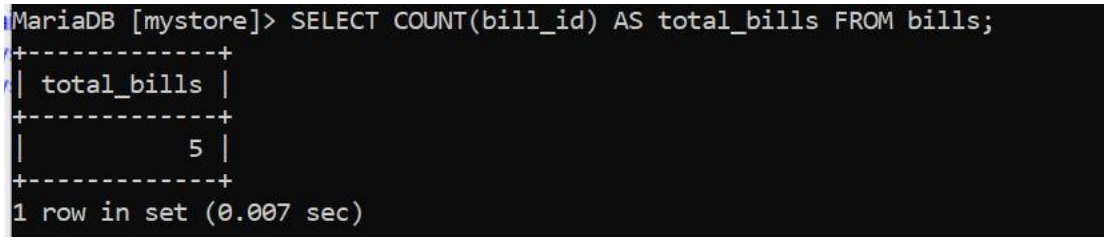
Figure 27queries of between

The designed query is Between which will lists all the between data in table format and will show all the workers according to their address between koteshwor and sinnamangal.

2. Show the count of the total bills from the bill_id of bills tables.



```
MariaDB [mystore]> select *from bills;
+---------+----------+--------+-------+-------------+
| bill_id | quantity | amount | price | customer_id |
+---------+----------+--------+-------+-------------+
|       1 |        2 |   3000 |  2000 |         110 |
|       2 |        3 |    300 |   500 |         112 |
|       3 |        3 |     30 |  5080 |         113 |
|       4 |        1 |   3042 | 50760 |         310 |
|       5 |        4 |    302 |  5360 |         430 |
+---------+----------+--------+-------+-------------+
```

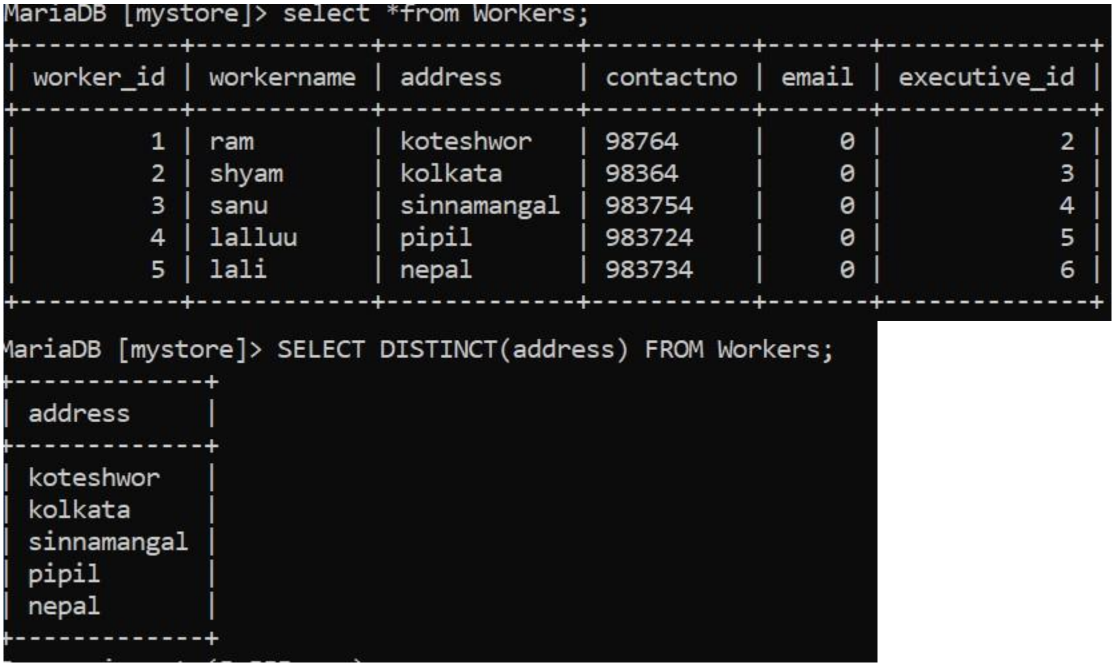


```
MariaDB [mystore]> SELECT COUNT(bill_id) AS total_bills FROM bills;
+-------------+
| total_bills |
+-------------+
|           5 |
+-------------+
1 row in set (0.007 sec)
```

Figure 28queries count

The designed query is for count which will count the number of customers ordered number of item.

3. Show the distinct (address) from workers column.

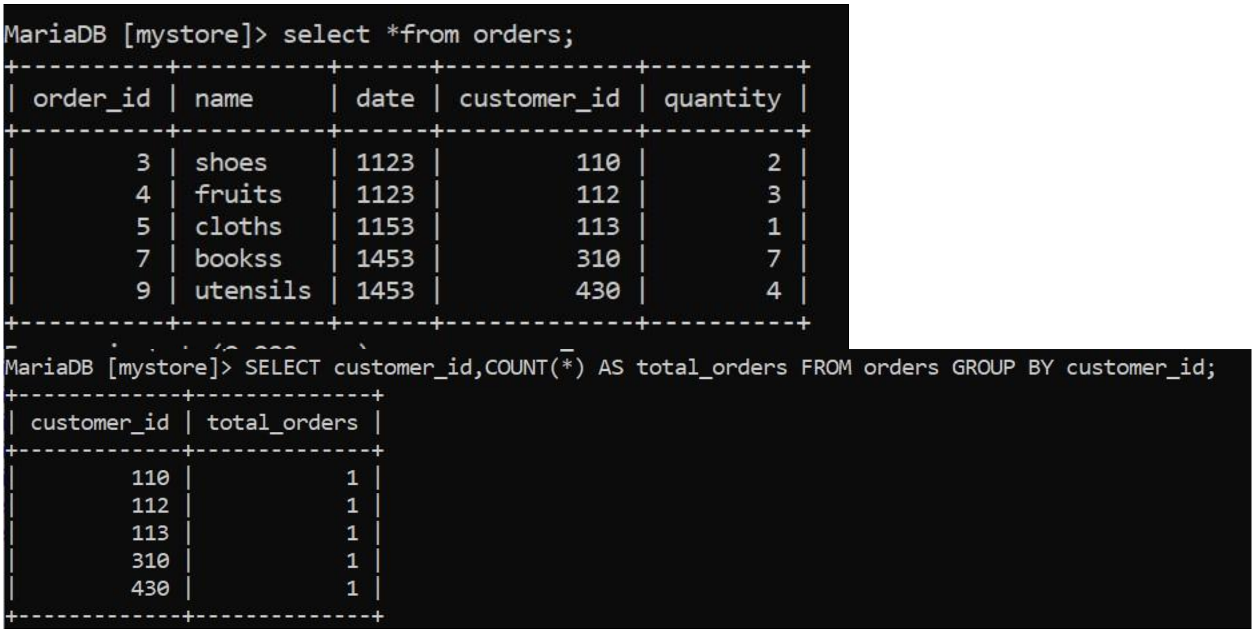


```
MariaDB [mystore]> select *from Workers;
+-----------+------------+-------------+-----------+-------+--------------+
| worker_id | workername | address     | contactno | email | executive_id |
+-----------+------------+-------------+-----------+-------+--------------+
|         1 | ram        | koteshwor   | 98764     |     0 |            2 |
|         2 | shyam      | kolkata     | 98364     |     0 |            3 |
|         3 | sanu       | sinnamangal | 983754    |     0 |            4 |
|         4 | lalluu     | pipil       | 983724    |     0 |            5 |
|         5 | lali       | nepal       | 983734    |     0 |            6 |
+-----------+------------+-------------+-----------+-------+--------------+

MariaDB [mystore]> SELECT DISTINCT(address) FROM Workers;
+-------------+
| address     |
+-------------+
| koteshwor   |
| kolkata     |
| sinnamangal |
| pipil       |
| nepal       |
+-------------+
```

Figure 29queries distinct

The designed query is used to distinct address of workers from the workers table .

4. Select total number of orders GROUP BY customer id from orders table.



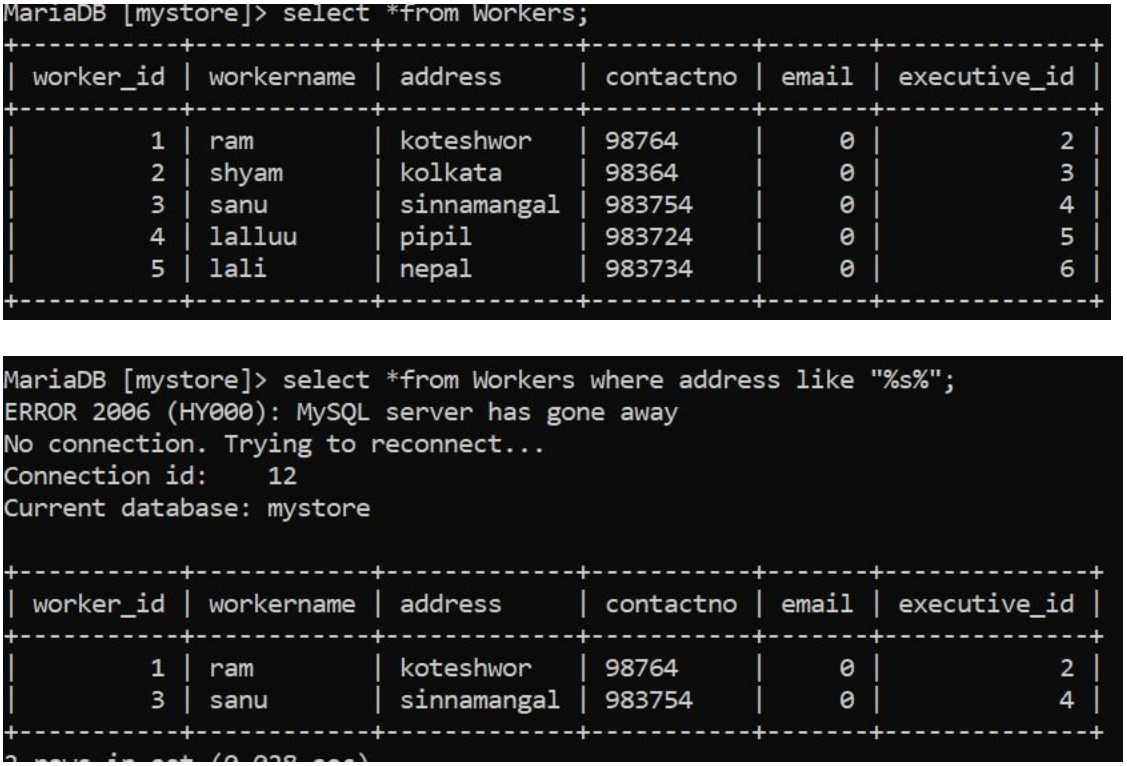
```
MariaDB [mystore]> select *from orders;
+----------+----------+------+-------------+----------+
| order_id | name     | date | customer_id | quantity |
+----------+----------+------+-------------+----------+
|        3 | shoes    | 1123 |         110 |        2 |
|        4 | fruits   | 1123 |         112 |        3 |
|        5 | cloths   | 1153 |         113 |        1 |
|        7 | bookss   | 1453 |         310 |        7 |
|        9 | utensils | 1453 |         430 |        4 |
+----------+----------+------+-------------+----------+
```

```
MariaDB [mystore]> SELECT customer_id,COUNT(*) AS total_orders FROM orders GROUP BY customer_id;
+-------------+--------------+
| customer_id | total_orders |
+-------------+--------------+
|         110 |            1 |
|         112 |            1 |
|         113 |            1 |
|         310 |            1 |
|         430 |            1 |
+-------------+--------------+
```

Figure 30limit queries

The designed query is used to limit name from the customer table

5. Select the address using like from workers on workers tables.



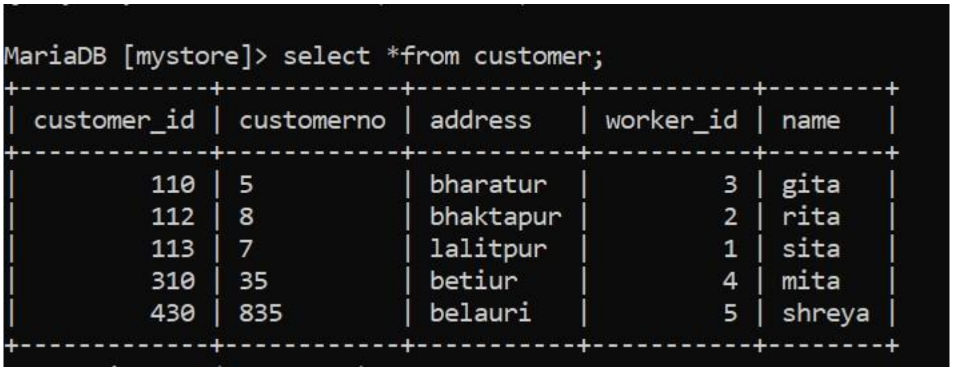
```
MariaDB [mystore]> select *from Workers;
+-----------+-----------+------------+-----------+-------+-------------+
| worker_id | workername | address    | contactno | email | executive_id |
+-----------+-----------+------------+-----------+-------+-------------+
|         1 | ram       | koteshwor  | 98764     |     0 |           2 |
|         2 | shyam     | kolkata    | 98364     |     0 |           3 |
|         3 | sanu      | sinnamangal | 983754   |     0 |           4 |
|         4 | lalluu    | pipil      | 983724    |     0 |           5 |
|         5 | lali      | nepal      | 983734    |     0 |           6 |
+-----------+-----------+------------+-----------+-------+-------------+
```

```
MariaDB [mystore]> select *from Workers where address like "%s%";
ERROR 2006 (HY000): MySQL server has gone away
No connection. Trying to reconnect...
Connection id:    12
Current database: mystore

+-----------+-----------+------------+-----------+-------+-------------+
| worker_id | workername | address    | contactno | email | executive_id |
+-----------+-----------+------------+-----------+-------+-------------+
|         1 | ram       | koteshwor  | 98764     |     0 |           2 |
|         3 | sanu      | sinnamangal | 983754   |     0 |           4 |
+-----------+-----------+------------+-----------+-------+-------------+
2 rows in set (0.028 sec)
```

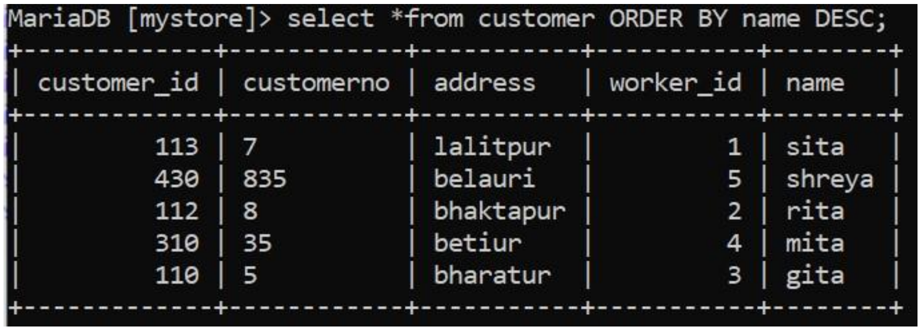
Figure 31like queries

The designed query is used to like address of workers from the workers table.

6. Select the order name from customer table.



```
MariaDB [mystore]> select *from customer;
+-------------+------------+----------+-----------+--------+
| customer_id | customerno | address  | worker_id | name   |
+-------------+------------+----------+-----------+--------+
|         110 | 5          | bharatur |         3 | gita   |
|         112 | 8          | bhaktapur |        2 | rita   |
|         113 | 7          | lalitpur |         1 | sita   |
|         310 | 35         | betiur   |         4 | mita   |
|         430 | 835        | belauri  |         5 | shreya |
+-------------+------------+----------+-----------+--------+
```

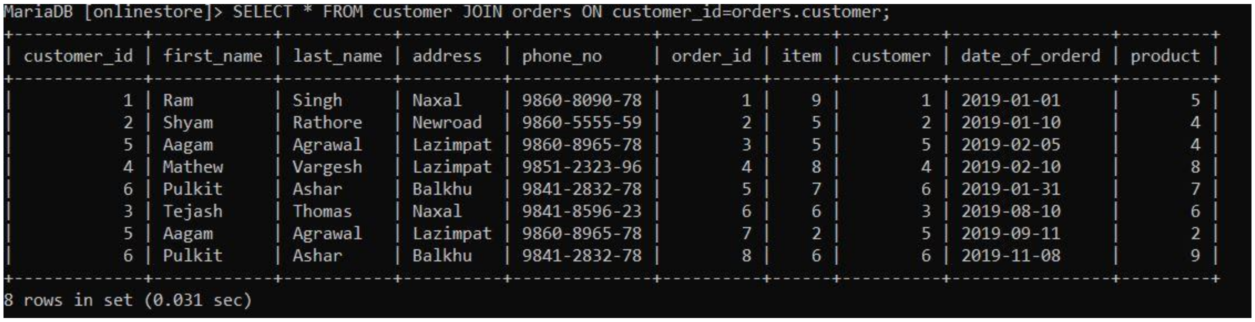


```
MariaDB [mystore]> select *from customer ORDER BY name DESC;
+-------------+------------+----------+-----------+--------+
| customer_id | customerno | address  | worker_id | name   |
+-------------+------------+----------+-----------+--------+
|         113 | 7          | lalitpur |         1 | sita   |
|         430 | 835        | belauri  |         5 | shreya |
|         112 | 8          | bhaktapur |        2 | rita   |
|         310 | 35         | betiur   |         4 | mita   |
|         110 | 5          | bharatur |         3 | gita   |
+-------------+------------+----------+-----------+--------+
```

Figure 32orderby queries

The designed query is used Order by name of customer from the customer table

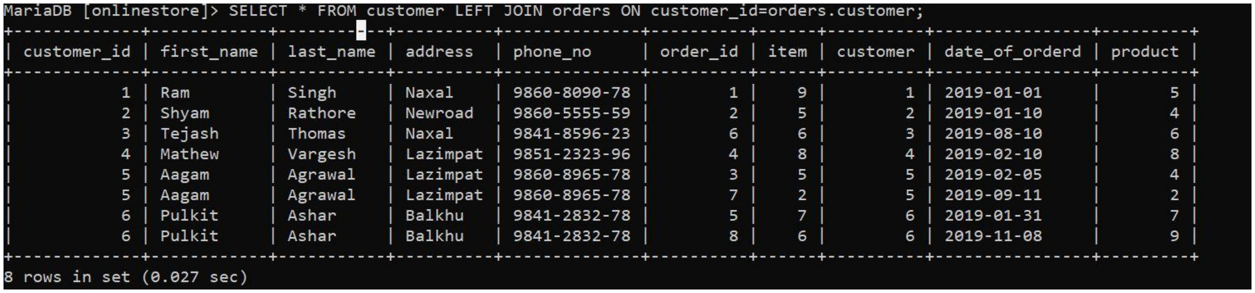
7. Select from Join query



```
MariaDB [onlinestore]> SELECT * FROM customer JOIN orders ON customer_id=orders.customer;
+-------------+------------+-----------+----------+--------------+----------+------+----------+----------------+---------+
| customer_id | first_name | last_name | address  | phone_no     | order_id | item | customer | date_of_orderd | product |
+-------------+------------+-----------+----------+--------------+----------+------+----------+----------------+---------+
|           1 | Ram        | Singh     | Naxal    | 9860-8090-78 |        1 |    9 |        1 | 2019-01-01     |       5 |
|           2 | Shyam      | Rathore   | Newroad  | 9860-5555-59 |        2 |    5 |        2 | 2019-01-10     |       4 |
|           5 | Aagam      | Agrawal   | Lazimpat | 9860-8965-78 |        3 |    5 |        5 | 2019-02-05     |       4 |
|           4 | Mathew     | Vargesh   | Lazimpat | 9851-2323-96 |        4 |    8 |        4 | 2019-02-10     |       8 |
|           6 | Pulkit     | Ashar     | Balkhu   | 9841-2832-78 |        5 |    7 |        6 | 2019-01-31     |       7 |
|           3 | Tejash     | Thomas    | Naxal    | 9841-8596-23 |        6 |    6 |        3 | 2019-08-10     |       6 |
|           5 | Aagam      | Agrawal   | Lazimpat | 9860-8965-78 |        7 |    2 |        5 | 2019-09-11     |       2 |
|           6 | Pulkit     | Ashar     | Balkhu   | 9841-2832-78 |        8 |    6 |        6 | 2019-11-08     |       9 |
+-------------+------------+-----------+----------+--------------+----------+------+----------+----------------+---------+
8 rows in set (0.031 sec)
```

Figure 33 join query

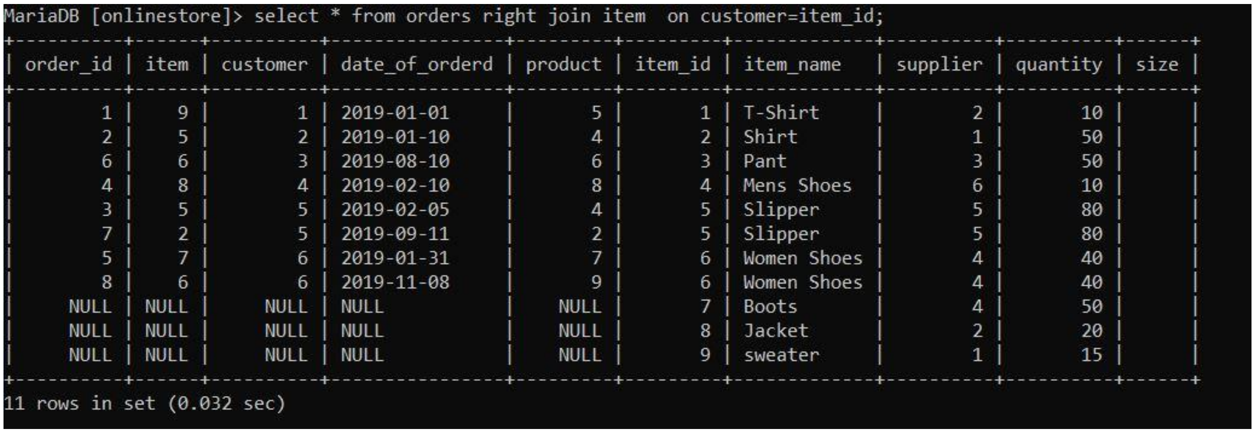
8. Select from Left join query



```
MariaDB [onlinestore]> SELECT * FROM customer LEFT JOIN orders ON customer_id=orders.customer;
+-------------+------------+-----------+----------+--------------+----------+------+----------+----------------+---------+
| customer_id | first_name | last_name | address  | phone_no     | order_id | item | customer | date_of_orderd | product |
+-------------+------------+-----------+----------+--------------+----------+------+----------+----------------+---------+
|           1 | Ram        | Singh     | Naxal    | 9860-8090-78 |        1 |    9 |        1 | 2019-01-01     |       5 |
|           2 | Shyam      | Rathore   | Newroad  | 9860-5555-59 |        2 |    5 |        2 | 2019-01-10     |       4 |
|           3 | Tejash     | Thomas    | Naxal    | 9841-8596-23 |        6 |    6 |        3 | 2019-08-10     |       6 |
|           4 | Mathew     | Vargesh   | Lazimpat | 9851-2323-96 |        4 |    8 |        4 | 2019-02-10     |       8 |
|           5 | Aagam      | Agrawal   | Lazimpat | 9860-8965-78 |        3 |    5 |        5 | 2019-02-05     |       4 |
|           5 | Aagam      | Agrawal   | Lazimpat | 9860-8965-78 |        7 |    2 |        5 | 2019-09-11     |       2 |
|           6 | Pulkit     | Ashar     | Balkhu   | 9841-2832-78 |        5 |    7 |        6 | 2019-01-31     |       7 |
|           6 | Pulkit     | Ashar     | Balkhu   | 9841-2832-78 |        8 |    6 |        6 | 2019-11-08     |       9 |
+-------------+------------+-----------+----------+--------------+----------+------+----------+----------------+---------+
8 rows in set (0.027 sec)
```

Figure 34 left join query

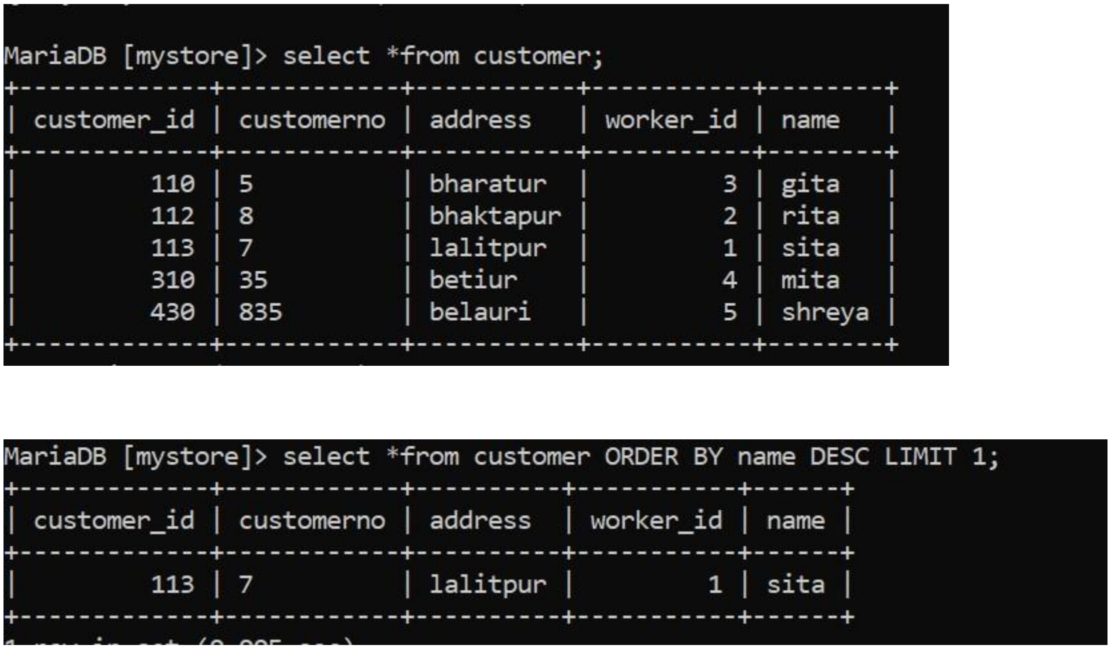
9. Select from Right join query



```
MariaDB [onlinestore]> select * from orders right join item  on customer=item_id;
+----------+------+----------+----------------+---------+---------+-------------+----------+----------+------+
| order_id | item | customer | date_of_orderd | product | item_id | item_name   | supplier | quantity | size |
+----------+------+----------+----------------+---------+---------+-------------+----------+----------+------+
|        1 |    9 |        1 | 2019-01-01     |       5 |       1 | T-Shirt     |        2 |       10 |      |
|        2 |    5 |        2 | 2019-01-10     |       4 |       2 | Shirt       |        1 |       50 |      |
|        6 |    6 |        3 | 2019-08-10     |       6 |       3 | Pant        |        3 |       50 |      |
|        4 |    8 |        4 | 2019-02-10     |       8 |       4 | Mens Shoes  |        6 |       10 |      |
|        3 |    5 |        5 | 2019-02-05     |       4 |       5 | Slipper     |        5 |       80 |      |
|        7 |    2 |        5 | 2019-09-11     |       2 |       5 | Slipper     |        5 |       80 |      |
|        5 |    7 |        6 | 2019-01-31     |       7 |       6 | Women Shoes |        4 |       40 |      |
|        8 |    6 |        6 | 2019-11-08     |       9 |       6 | Women Shoes |        4 |       40 |      |
|     NULL | NULL |     NULL | NULL           |    NULL |       7 | Boots       |        4 |       50 |      |
|     NULL | NULL |     NULL | NULL           |    NULL |       8 | Jacket      |        2 |       20 |      |
|     NULL | NULL |     NULL | NULL           |    NULL |       9 | sweater     |        1 |       15 |      |
+----------+------+----------+----------------+---------+---------+-------------+----------+----------+------+
11 rows in set (0.032 sec)
```

Figure 35 right join query

10. Select the limit of customer order by name from customer table.



```
MariaDB [mystore]> select *from customer;
+-------------+------------+----------+-----------+--------+
| customer_id | customerno | address  | worker_id | name   |
+-------------+------------+----------+-----------+--------+
|         110 | 5          | bharatur |         3 | gita   |
|         112 | 8          | bhaktapur |        2 | rita   |
|         113 | 7          | lalitpur |         1 | sita   |
|         310 | 35         | betiur   |         4 | mita   |
|         430 | 835        | belauri  |         5 | shreya |
+-------------+------------+----------+-----------+--------+
```

```
MariaDB [mystore]> select *from customer ORDER BY name DESC LIMIT 1;
+-------------+------------+----------+-----------+------+
| customer_id | customerno | address  | worker_id | name |
+-------------+------------+----------+-----------+------+
|         113 | 7          | lalitpur |         1 | sita |
+-------------+------------+----------+-----------+------+
```

Figure 36 limit query

The designed query is used to limit name from the customer table.

CONCLUSION

The project entitled **Online shopping store** was completed successfully.

The entire project has been developed with much care and free of errors at the same time it is efficient and less time consuming. The main purpose of this project is to develop an organization for purchasing items from a shopping mall through online source or physical apperence.

This project help in gaining valuable information and practical knowledge on several topics like, making databases, queries, data dictionary, ER diagram, relational database etc. Also the project helps us to understand about the entire process more clearly and efficiently. This project has given us great satisfaction by making a database which can be implemented to any nearby shops or store of branded shops selling various kind of products or goods by simple modifications.

A number of feature can be added to this system in the future if any informative entity is needed so that different offers can be given to each customers . System may keep track of history of purchases of each customer and provide suggestion based on their history. These feature could have been implemented unless the time did not limited us.

BIBLOGRAPHY

Bibliography

Anon., n.d. *www.bitdegree.org/.../what-is-a-relational-database.* [Online].

oracle, jun -10 -2016. s.l.: s.n.

oracle, n.d. database.

zwass, v., n.d. information system.

References

Anon., n.d. *www.bitdegree.org/.../what-is-a-relational-database.* [Online].

oracle, jun -10 -2016. s.l.: s.n.

oracle, n.d. database.

zwass, v., n.d. information system.