LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title
CC4057NI Introduction to Information Systems

Assessment Weightage & Type
30% Individual Coursework 2

Year and Semester
2019-20 Autumn

Student Name: Mandip Thapa
Group: L1N6
London Met ID:
College ID: NP01NT4A190136

Assignment Due Date: 20 December, 2019 (week 8)

Assignment Submission Date: 20 December, 2019(week 8)

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 mark of zero will be awarded.

Table of contents

List of tables

List of figures

Introduction

Before the use of Database management system (DBMS), normal file systems were used in different organizations to keep various records. To search for data, various records, it is needed to be searched from bundle of files and it used to consume more time. After the introduction of DBMS, record keeping has been digitalized and it is way more convenient and less time consuming. Using different queries, the required data can be easily located in the database. As a result, the use of DBMS has been increasing and has become very important in Dealership Company and like this different organization. The simple database created to keep the records in a dealership is essential for present and future use. The record which is kept helps in various study and research.

The dealership has a number of departments; each department has their own department name and ID. The department information is kept in the table. There are many staffs working in different departments like sales department, management department, service department and financial department. The records of staff such as their ID and their department ID are also kept in the table. The database helps to find out which staff works on which department. The client's records are also kept in the database. Client personal information is important for staff and department to carry out various services. The records remain in the database and if there is an order form client, it is easier to fulfill their demands. In the same way, the records of orders and items are also kept and they are linked with other tables. If a research is to be done on the clients and their order, then database records are very useful.

The research on the clients and their order may be useful in finding out which items has high sales rate and which items has great demand in the market. This helps in fulfilling the demands of the clients. There are different other function of database.

Key Terms

1. Database

Database is the computer structure which saves, deliver, organize and protect the data (Machajewski, 2017). Database is essentially holders for information.

2. Primary Key

A primary key is an uncommon or special relational database table column or combination of columns assigned to uniquely recognize every table records.

A primary key's main features are:

- It doesn't contain null values.
- It must have a unique value for each row of data. (Techopedia, 2019)

3. Foreign key

A foreign key is the group of columns in a relational database table which provides a connection between information in two tables. It acts as a cross-reference between tables because it references the essential key of another table, in this way building up a connection between them. (Techopedia, 2019)

4. Constraints

Constraints are the guidelines implemented on the data columns of a table. These are used to limit the type of information that can go into a table. This guarantees the accuracy and reliability of the information in the database. Examples: NOT NULL, UNIQUE, etc.

Discussion and analysis

First five tables were created in this relational database system. Each table had a primary key which uniquely identify each row in table. The values in primary key where each points out to a unique value in the corresponding table .For a relational database system, the tables in the database must be connected using suitable foreign keys. Foreign keys were created so that we can relate different tables.

The primary keys in all tables were given “Auto_Increment” constraint so as to increase the numerical value automatically. The foreign keys in the tables referenced to the appropriate primary key of the other related tables in the database. The foreign keys used in my database are as follows:

- The column “department_id” of table staffs references to the column “department_id” of table departments.
- The column “staff_id” of table clients references to the column “staff_id” of table staffs.
- The column “client_id” of table orders references to the column “client_id” of table clients.
- The column “order_id” of table items references to the column “order_id” of table orders.

To store the values in tables in a database, the application “XAMPP” was installed in PC. MySQL was started and shell was opened. Then, we were ready to create and use databases and create and insert values into the tables. Thereafter, different queries were used to manipulate and update the values in the tables of the database. Similarly, the tables were illustrated with the help of Entity-Relationship (ER) diagram and Relational diagram drawn with the help of draw.io. Data dictionaries were also drawn to describe the columns of the entities (tables).

Database model

The database is about Dealership Company which consists of five different tables. The tables are departments, staffs, clients, orders and items. The table “departments” stores information like department ID, department name and address of the departments. The table “staffs” stores information like staff ID, staff’s contact, email and the department where they work. The table “clients” stores information like client ID, client phone number, client address, staff ID and their email. The table “orders” stores information like order ID, order date, ordered item name and client’s ID who has given order. The table “items” stores information like item ID, company name of item, price of item and its order ID.

Entity-Relation (ER) Diagram

An element relationship outline (ERD) is an information displaying method that graphically represents an information system entity and the connections between those elements. An ERD is a theoretical and illustrative model of information used to speak to the substance system framework. An entity-relationship diagram (ERD), also known as an entity-relationship model, is a graphical representation which describes relationships among people, objects, places, concepts or events within an information technology (IT) system (Techtarget, 2019).

In this ER diagram we have 5 entities which are departments, staffs, clients, orders and items. All entities are in relation. Here department has four attributes which are department_id, department_name, address and email. Here entities department and staffs has one to many relations. Here entity staffs have four attributes which are staff_id, contact, email and department_id. The entities staffs and clients have one to many relations. Here entity clients have five attributes which are client_id, address, email, phone_number and staff_id. The entities clients and orders have one to many relation. The entity order has four attributes which are order_id, item_name, order_date and client_id. The entities order and item has one to many relation. The entity item has four attributes which are item_id, price, company and order_id. The entity relation diagram is in next page.

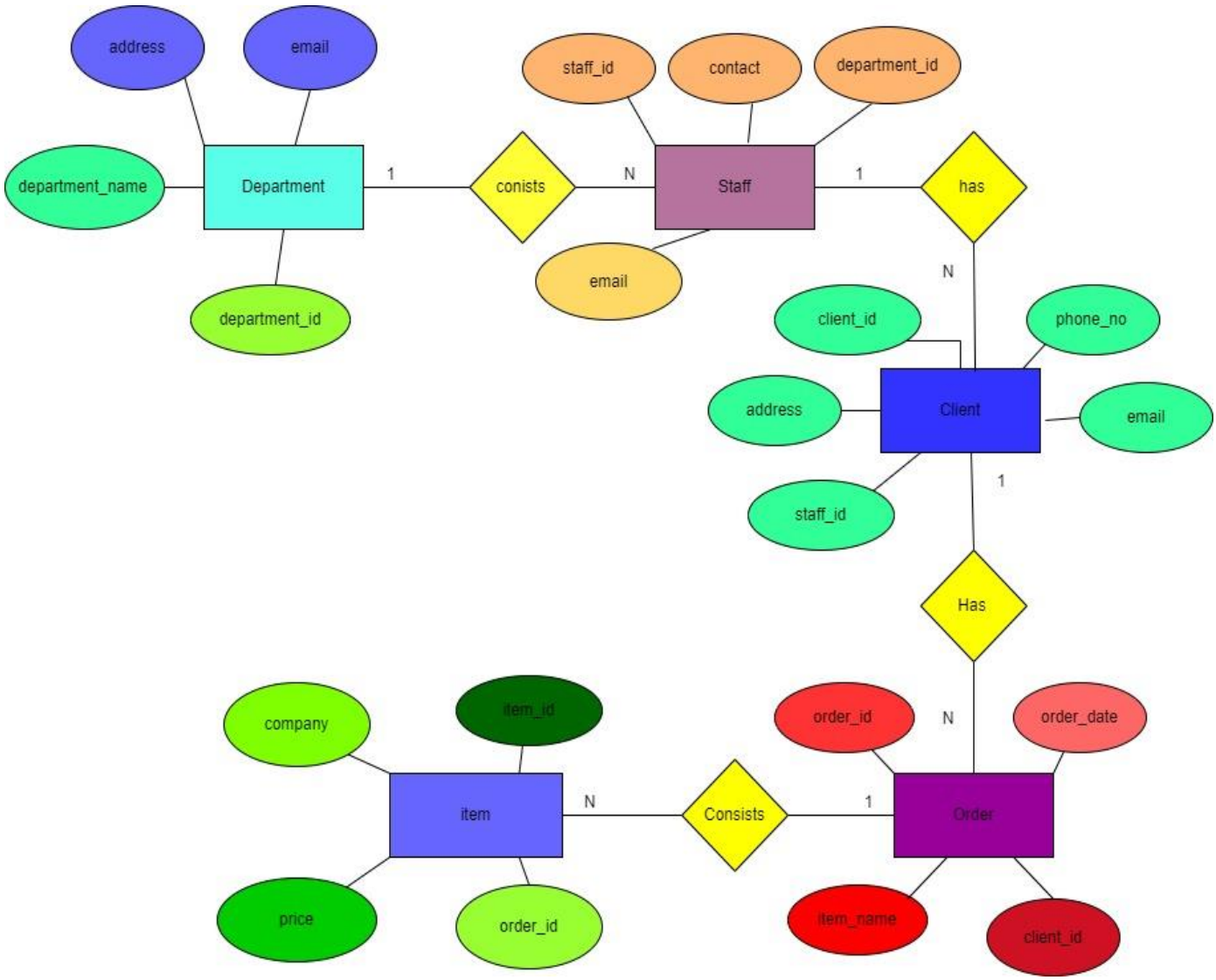

Figure 1: ER Diagram

Relational Diagram

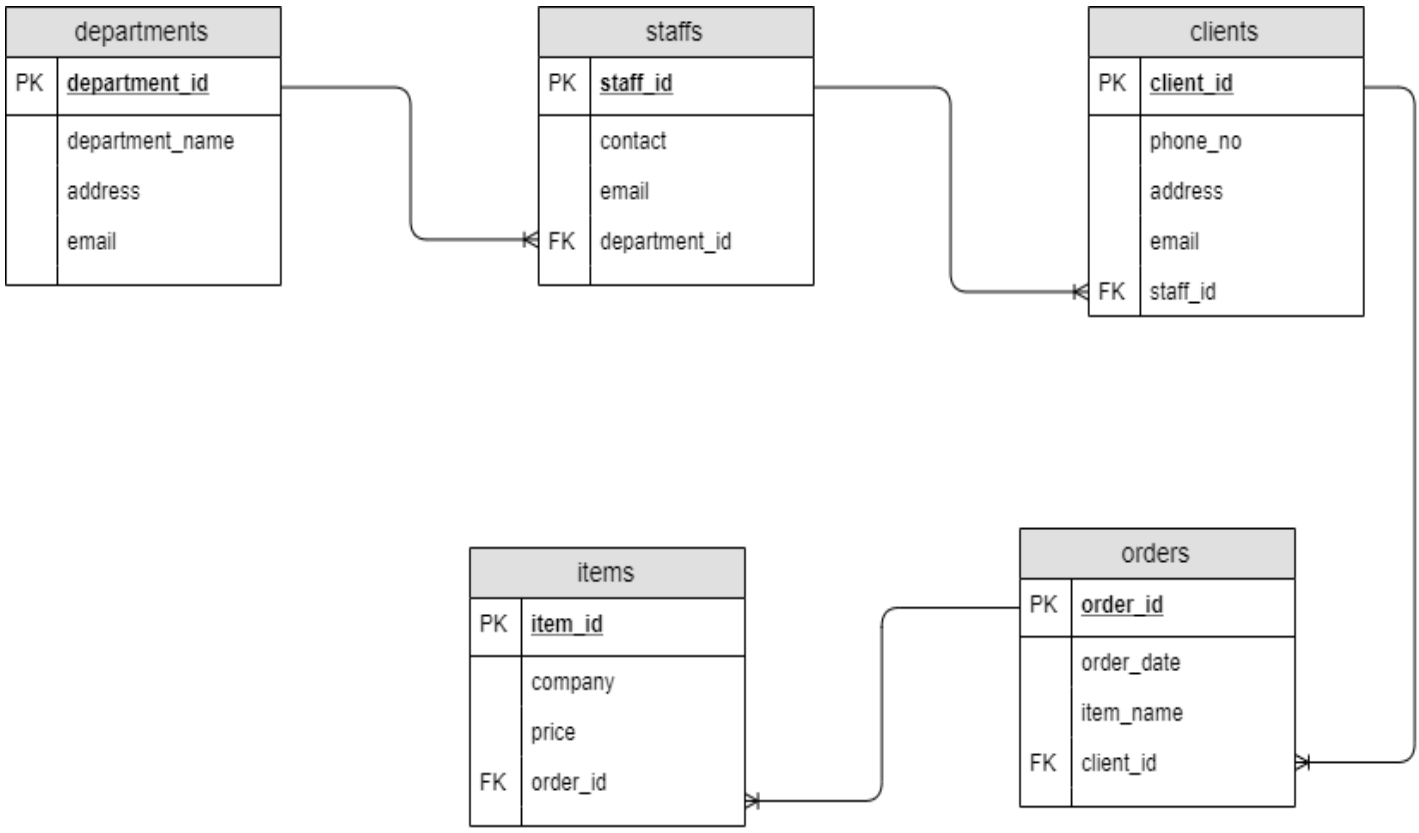

Figure 2: Relational Diagram

Table Information

Table-departments:

In this table there are 4 columns. They are department_id, department_name, address and email. Here department_id is primary key as it has unique value. The column department_name has all the names of the department of the company. The column address has information about the address of the departments. The column email has the email of the departments.

Table-staffs:

This table consists of 4 columns which are staff_id, contact, email and department_id. Here staff_id is primary key as it has unique value. Here department_id is foreign key as it references to the department_id of table departments. The column contact has the contact number of all the staffs and the column email has email of all the staff.

Table-clients:

In this table there are 5 columns which are client_id, phone_no, address, email and staff_id. Here client_id is primary key as it has unique value. Here staff_id is foreign key as it references to the staff_id of table staffs. The column address and email has address and email of all the clients. The column phone_no has the contact number of all the clients.

Table-orders:

This table consists of 4 columns which are order_id, order_date, item_name and client_id. Here order_id is primary key as it has unique value. Here client_id is foreign key as it references to the client_id of table clients. The column order_date has the date when order was placed. The column item_name has the name of items which has been ordered.

Table-items:

This table consists of 4 columns which are item_id, company, price and order_id. Here item_id is primary key as it has unique value. Here order_id is foreign key as it references to the order_id of table orders. The column company has the company name of products. The column price has the price of the products.

Creation and insertion of screenshots

Creation of database “dealer” and tables “departments” and “staff”.

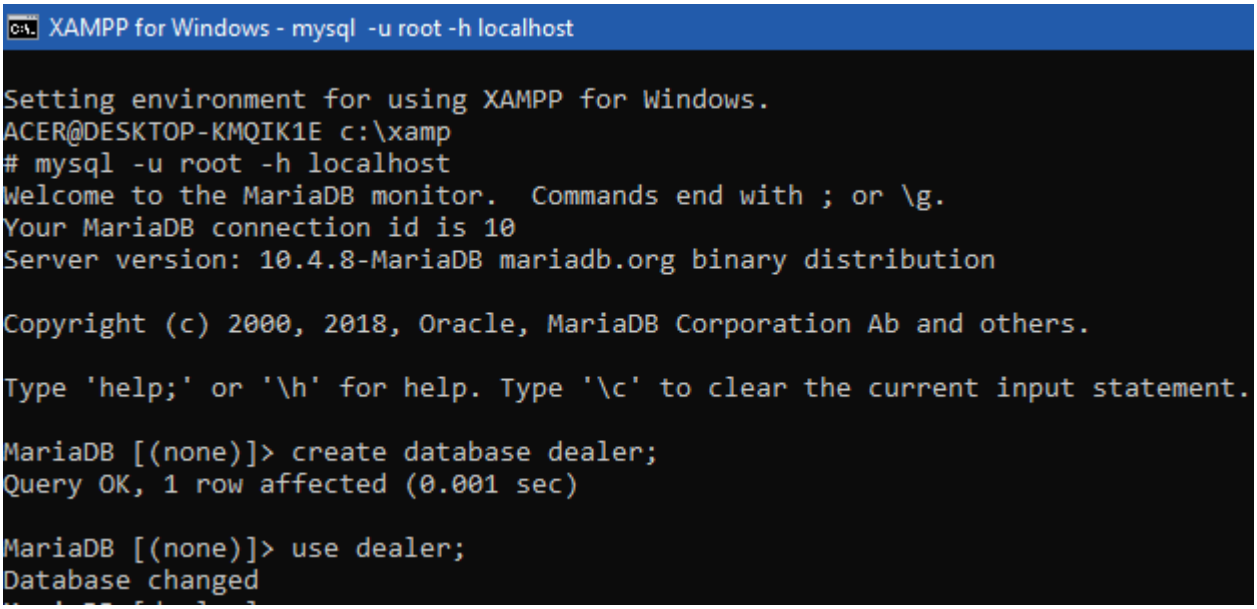

Figure 3: Creation of database

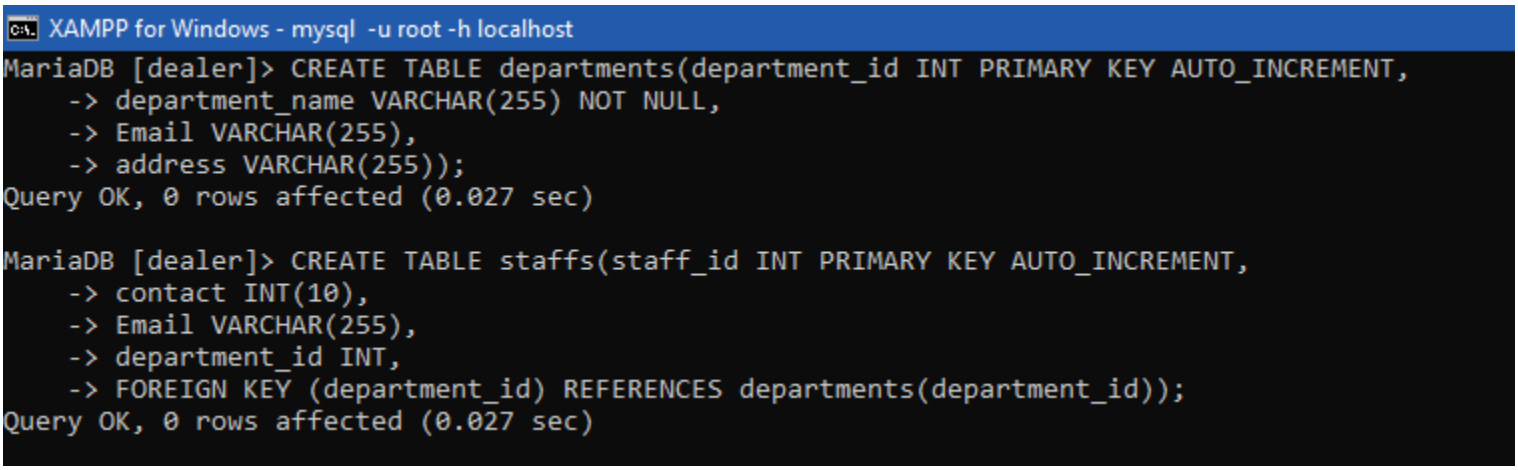

Figure 4: Creation of table-1

Creation of tables “clients”, “orders” and “items”.

```
XAMPP for Windows - mysql  -u root -h localhost
MariaDB [dealer]> CREATE TABLE clients(client_id INT PRIMARY KEY AUTO_INCREMENT,
    -> phone_no INT(10) NOT NULL,
    -> address VARCHAR(255),
    -> Email VARCHAR(255));
Query OK, 0 rows affected (0.020 sec)

MariaDB [dealer]> CREATE TABLE orders(order_id INT PRIMARY KEY AUTO_INCREMENT,
    -> order_date VARCHAR(255) NOT NULL,
    -> item_name VARCHAR(255) NOT NULL,
    -> client_id INT,
    -> FOREIGN KEY (client_id) REFERENCES clients(client_id));
Query OK, 0 rows affected (0.022 sec)

MariaDB [dealer]> CREATE TABLE items(item_id INT PRIMARY KEY AUTO_INCREMENT,
    -> company VARCHAR(255) NOT NULL,
    -> price INT,
    -> order_id INT,
    -> FOREIGN KEY (order_id) REFERENCES orders(order_id));
Query OK, 0 rows affected (0.023 sec)
```

Figure 5: Creation of table-2

Insertion of values into tables “departments” and “staffs”.

```
MariaDB [dealer]> INSERT INTO departments VALUES(001,"sales department","salesdept12@gmail.com","durbarmarg"),
    -> (002,"management department","managedept12@gmail.com","durbarmargh"),
    -> (003,"costumer care department","costdept12@gmail.com","lazimpath"),
    -> (004,"Technical department","techdept12@gmail.com","lazimpath"),
    -> (005,"financial department","techdept12@gmail.com","durbarmargh");
Query OK, 5 rows affected (0.033 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

Figure 6: Insertion into table-1

```
MariaDB [dealers]> INSERT INTO staffs VALUES(111,6634786,"harry11@gmail.com",1),
    -> (112,6656283,"sam33@gmail.com",2),
    -> (113,5582734,"raju44@gmail.com",3),
    -> (114,5092543,"ram88@gmail.com",4),
    -> (115,5592247,"erik67@gmail.com",5);
Query OK, 5 rows affected (0.015 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

Figure 7: Insertion into table-2

Insertion of values into tables “clients”, “orders” and “items”.

```
MariaDB [dealers]> INSERT INTO clients VALUES(991,5595243,"balkot","mandy99@gmail.com",111),
    -> (992,6631283,"kumaripati","bishal12@gmail.com",112),
    -> (993,6082637,"gausala","hari12@gmail.com",113),
    -> (994,6683056,"koteshwor","jack32@gmail.com",114),
    -> (995,5591254,"thimi","sumit77@gmail.com",115);
Query OK, 5 rows affected (0.019 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [dealers]>
```

```
MariaDB [dealer]> INSERT INTO orders Values(501,"march 2","Laptop","991"),
    -> (502,"april 22","mobile","992"),
    -> (503,"april 3","ipod","993"),
    -> (504,"july 1","ipad","994"),
    -> (505,"june 30","monitor","995");
Query OK, 5 rows affected (0.011 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [dealer]> INSERT INTO items Values(441,"apple",70000,501),
    -> (442,"samsung",50000,502),
    -> (443,"apple",50000,503),
    -> (444,"samsung",60000,504),
    -> (445,"samsung",90000,505);
Query OK, 5 rows affected (0.010 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

Figure 8: Insertion into table-3

Description of tables “departments”, ”staffs” and “clients”.

In table clients I used SQL syntax [ALTER TABLE <Table name> ADD <Column name><Data type>] to add column “staff_id” and used it as a foreign key in table “clients”.

```
MariaDB [dealer]> DESCRIBE departments;
+-----------------+--------------+------+-----+---------+----------------+
| Field           | Type         | Null | Key | Default | Extra          |
+-----------------+--------------+------+-----+---------+----------------+
| department_id   | int(11)      | NO   | PRI | NULL    | auto_increment |
| department_name | varchar(255) | NO   |     | NULL    |                |
| Email           | varchar(255) | YES  |     | NULL    |                |
| address         | varchar(255) | YES  |     | NULL    |                |
+-----------------+--------------+------+-----+---------+----------------+
4 rows in set (0.025 sec)

MariaDB [dealer]> describe staffs;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| staff_id      | int(11)      | NO   | PRI | NULL    | auto_increment |
| contact       | int(10)      | YES  |     | NULL    |                |
| Email         | varchar(255) | YES  |     | NULL    |                |
| department_id | int(11)      | YES  | MUL | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
4 rows in set (0.020 sec)

MariaDB [dealer]> describe clients;
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| client_id | int(11)      | NO   | PRI | NULL    | auto_increment |
| phone_no  | int(10)      | NO   |     | NULL    |                |
| address   | varchar(255) | YES  |     | NULL    |                |
| Email     | varchar(255) | YES  |     | NULL    |                |
| staff_id  | int(11)      | YES  | MUL | NULL    |                |
+-----------+--------------+------+-----+---------+----------------+
```

Figure 9: Description of table-1

Description of table “orders” and “items”.

```
MariaDB [dealer]> describe orders;
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| order_id   | int(11)      | NO   | PRI | NULL    | auto_increment |
| order_date | varchar(255) | NO   |     | NULL    |                |
| item_name  | varchar(255) | NO   |     | NULL    |                |
| client_id  | int(11)      | YES  | MUL | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
4 rows in set (0.014 sec)

MariaDB [dealer]> describe items;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| item_id  | int(11)      | NO   | PRI | NULL    | auto_increment |
| company  | varchar(255) | NO   |     | NULL    |                |
| price    | int(11)      | YES  |     | NULL    |                |
| order_id | int(11)      | YES  | MUL | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
4 rows in set (0.022 sec)
```

Figure 10: Description of table-2

Selection of data from table “departments”.

```
MariaDB [dealer]> select * from departments;
+---------------+--------------------------+-------------------------+-------------+
| department_id | department_name          | Email                   | address     |
+---------------+--------------------------+-------------------------+-------------+
|             1 | sales department         | salesdept12@gmail.com   | durbarmarg  |
|             2 | management department    | managedept12@gmail.com  | durbarmargh |
|             3 | costumer care department | costdept12@gmail.com    | lazimpath   |
|             4 | Technical department     | techdept12@gmail.com    | lazimpath   |
|             5 | financial department     | techdept12@gmail.com    | durbarmargh |
+---------------+--------------------------+-------------------------+-------------+
5 rows in set (0.001 sec)
```

Figure 11: Selection of table-1

Selection of data from table “staffs”.

```
MariaDB [dealer]> select * from staffs;
+----------+---------+--------------------+---------------+
| staff_id | contact | Email              | department_id |
+----------+---------+--------------------+---------------+
|      111 | 6634786 | harry12@gmail.com  |             1 |
|      112 | 6656283 | sam33@gmail.com    |             2 |
|      113 | 5582734 | raju44@gmail.com   |             3 |
|      114 | 5092543 | ram88@gmail.com    |             4 |
|      115 | 5592247 | erik67@gmail.com   |             5 |
+----------+---------+--------------------+---------------+
5 rows in set (0.001 sec)
```

Figure 12: Selection of table-2

Selection of data from table “clients”.

```
MariaDB [dealer]> select * from clients;
+-----------+----------+------------+--------------------+----------+
| client_id | phone_no | address    | Email              | staff_id |
+-----------+----------+------------+--------------------+----------+
|       991 |  5595243 | balkot     | mandy99@gmail.com  |      111 |
|       992 |  6631283 | kumaripati | bishal12@gmail.com |      112 |
|       993 |  6082637 | gausala    | hari12@gmail.com   |      113 |
|       994 |  6683656 | koteshwor  | jack32@gmail.com   |      114 |
|       995 |  5591254 | thimi      | sumit77@gmail.com  |      115 |
+-----------+----------+------------+--------------------+----------+
5 rows in set (0.001 sec)
```

Figure 13: Selection of table-3

Selection of data from table “orders” and “items”.

```
MariaDB [dealer]> select * from orders;
+----------+------------+-----------+-----------+
| order_id | order_date | item_name | client_id |
+----------+------------+-----------+-----------+
|      501 | march 2    | Laptop    |       991 |
|      502 | april 22   | mobile    |       992 |
|      503 | april 3    | ipod      |       993 |
|      504 | july 1     | ipad      |       994 |
|      505 | june 30    | monitor   |       995 |
+----------+------------+-----------+-----------+
5 rows in set (0.001 sec)

MariaDB [dealer]> select * from items;
+---------+---------+-------+----------+
| item_id | company | price | order_id |
+---------+---------+-------+----------+
|     441 | apple   | 70000 |      501 |
|     442 | samsung | 50000 |      502 |
|     443 | apple   | 50000 |      503 |
|     444 | samsung | 60000 |      504 |
|     445 | samsung | 90000 |      505 |
+---------+---------+-------+----------+
5 rows in set (0.000 sec)
```

Figure 14: Selection of table-4

Data dictionary

Data dictionary of table “departments”.

Entity name	Entity Description	Column name	Column description	Data type	length	Primary key	Foreign Key	Nullable	Unique	Notes
Departments	Departments are the organs of the company.	Department_id	Id of the department	INT		TRUE	FALSE	FALSE	TRUE	Auto increment
		Department_name	Name of the department	VARCHAR	255	FALSE	FALSE	FALSE	FALSE	
		address	Address of the department	VARCHAR	255	FALSE	FALSE	FALSE	FALSE	
		email	Email of the department	VARCHAR	255	FALSE	TRUE	FALSE	FALSE	

Table 1: Data dictionary of departments

Data dictionary of table “staffs”.

Entity name	Entity Description	Column name	Column description	Data type	length	Primary key	Foreign Key	Nullable	Unique	Notes
Staffs	Staff are the ones who works in the department.	Staff_id	Id of the staff	INT		TRUE	FALSE	FALSE	TRUE	Auto increment
		contact	Contact of staff	INT	10	FALSE	FALSE	FALSE	FALSE	
		email	Email of the staff	VARCHAR	255	FALSE	FALSE	FALSE	FALSE	
		Department_id	Id of the department	INT		FALSE	TRUE	FALSE	FALSE	Reference to department_id of departments table

Table 2: Data dictionary of staffs

Data dictionary of table “clients”.

Entity name	Entity Description	Column name	Column description	Data type	length	Primary key	Foreign Key	Nullable	Unique	Notes
Clients	Clients are someone who are using services from company.	Client_id	ID of the clients.	INT		TRUE	FALSE	FALSE	TRUE	Auto increment
		Phone_no	Contact number of clients	INT		FALSE	FALSE	FALSE	FALSE	
		address	Address of the clients	VARCHAR	255	FALSE	FALSE	FALSE	FALSE	
		Email	Email of the clients	VARCHAR	255	FALSE	FALSE	FALSE	FALSE	
		Staff_id	Id of the staff who is giving services to clients	INT		FALSE	TRUE	FALSE	FALSE	References to staff_id of staffs table

Table 3: Data dictionary of clients

Data dictionary of table “orders”.

Entity name	Entity Description	Column name	Column description	Data type	length	Primary key	Foreign Key	Nullable	Unique	Notes
Orders	Orders are the command given.	Order_id	Id of the given order	INT		TRUE	FALSE	FALSE	TRUE	Auto increment
		Item_name	Name of ordered item	VARCHAR	255	FALSE	FALSE	FALSE	FALSE	
		Order_date	Date of item ordered	VARCHAR	255	FALSE	FALSE	FALSE	FALSE	
		Client_id	Id of the client	INT		FALSE	TRUE	FALSE	FALSE	References to client_id of table clients

Table 4: Data dictionary of orders

Data dictionary of table “items”.

Entity name	Entity Description	Column name	Column description	Data type	length	Primary key	Foreign Key	Nullable	Unique	Notes
Items	Item is the part of the list.	Item_id	Id of the item ordered	INT		TRUE	FALSE	FALSE	TRUE	Auto increment
		Company	Company name of item	VARCHAR	255	FALSE	FALSE	FALSE	FALSE	
		Price	Price of item	INT		FALSE	FALSE	FALSE	FALSE	
		Order_id	Id of the given order	INT		FALSE	TRUE	FALSE	FALSE	References to order_id from table orders

Table 5: Data dictionary of Items

Queries

Query 1: Select Email from departments Where (department_id=1);

This query shows the email of the department whose department_id=1.

```
MariaDB [dealer]> select Email from departments Where (department_id=1);
+-----------------------+
| Email                 |
+-----------------------+
| salesdept12@gmail.com |
+-----------------------+
1 row in set (0.001 sec)
```

Figure 15: Query 1

Query 2: ALTER TABLE departments ADD COLUMN contact INT;

This query helps to add extra column in the table.

```
MariaDB [dealer]> ALTER TABLE departments ADD COLUMN contact INT;
Query OK, 0 rows affected (0.030 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [dealer]> describe departments;
+-----------------+--------------+------+-----+---------+----------------+
| Field           | Type         | Null | Key | Default | Extra          |
+-----------------+--------------+------+-----+---------+----------------+
| department_id   | int(11)      | NO   | PRI | NULL    | auto_increment |
| department_name | varchar(255) | NO   |     | NULL    |                |
| Email           | varchar(255) | YES  |     | NULL    |                |
| address         | varchar(255) | YES  |     | NULL    |                |
| contact         | int(11)      | YES  |     | NULL    |                |
+-----------------+--------------+------+-----+---------+----------------+
5 rows in set (0.007 sec)
```

Figure 16: Query 2

Query 3: UPDATE departments SET contact=66341717 WHERE department_id=1;

This query helps to update the column of a table.

```
MariaDB [dealer]> UPDATE departments SET contact=66341717 WHERE department_id=1;
Query OK, 1 row affected (0.012 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [dealer]> select * from departments;
+---------------+--------------------------+-----------------------+-------------+----------+
| department_id | department_name          | Email                 | address     | contact  |
+---------------+--------------------------+-----------------------+-------------+----------+
|             1 | sales department         | salesdept12@gmail.com | durbarmarg  | 66341717 |
|             2 | management department    | managedept12@gmail.com | durbarmargh |     NULL |
|             3 | costumer care department | costdept12@gmail.com  | lazimpath   |     NULL |
|             4 | Technical department     | techdept12@gmail.com  | lazimpath   |     NULL |
|             5 | financial department     | techdept12@gmail.com  | durbarmargh |     NULL |
+---------------+--------------------------+-----------------------+-------------+----------+
5 rows in set (0.001 sec)
```

Figure 17: Query 3

Query 4: SELECT * FROM items ORDER BY price desc;

This query select table and order it according to price in descending order.

```
MariaDB [dealer]> SELECT * FROM items ORDER BY price desc;
+---------+---------+-------+----------+
| item_id | company | price | order_id |
+---------+---------+-------+----------+
|     445 | samsung | 90000 |      505 |
|     441 | apple   | 70000 |      501 |
|     444 | samsung | 60000 |      504 |
|     442 | samsung | 50000 |      502 |
|     443 | apple   | 50000 |      503 |
+---------+---------+-------+----------+
5 rows in set (0.020 sec)
```

Figure 18: Query 4

Query 5: SELECT * FROM clients where (address="balkot");

This query selects data from table whose address is balkot.

```
MariaDB [dealer]> SELECT * FROM clients where (address="balkot");
+-----------+------------+---------+--------------------+-----------+
| client_id | phone_no   | address | Email              | staff_id |
+-----------+------------+---------+--------------------+-----------+
|       991 | 2147483647 | balkot  | mandy99@gmail.com |      111 |
+-----------+------------+---------+--------------------+-----------+
1 row in set (0.021 sec)
```

Figure 19: Query 5

Query 6: Select * from clients JOIN orders on clients.client_id=orders.client_id;

This query joints the data between two tables.

```
MariaDB [dealer]> Select * from clients JOIN orders on clients.client_id=orders.client_id;
+-----------+------------+-----------+--------------------+----------+----------+------------+-----------+-----------+
| client_id | phone_no   | address   | Email              | staff_id | order_id | order_date | item_name | client_id |
+-----------+------------+-----------+--------------------+----------+----------+------------+-----------+-----------+
|       991 | 2147483647 | balkot    | mandy99@gmail.com |      111 |      501 | march 2    | Laptop    |       991 |
|       992 | 2147483647 | kumaripati | bishal12@gmail.com |      112 |      502 | april 22   | mobile    |       992 |
|       993 | 2147483647 | gausala   | hari12@gmail.com  |      113 |      503 | april 3    | ipod      |       993 |
|       994 | 2147483647 | koteshwor | jack32@gmail.com  |      114 |      504 | july 1     | ipad      |       994 |
|       995 | 2147483647 | thimi     | sumit77@gmail.com |      115 |      505 | june 30    | monitor   |       995 |
+-----------+------------+-----------+--------------------+----------+----------+------------+-----------+-----------+
5 rows in set (0.017 sec)

MariaDB [dealer]>
```

Figure 20: Query 6

Query 7: Select * from items where (company="apple");

This query selects data from table whose company name is apple.

```
MariaDB [dealer]> Select * from items where (company="apple");
+---------+---------+-------+----------+
| item_id | company | price | order_id |
+---------+---------+-------+----------+
|     441 | apple   | 70000 |      501 |
|     443 | apple   | 50000 |      503 |
+---------+---------+-------+----------+
2 rows in set (0.011 sec)
```

Figure 21: Query 7

Query 8: SELECT SUM(price) from items;

This query finds the sum of price column in the items table.

```
MariaDB [dealer]> SELECT SUM(price) from items;
+-------------+
| SUM(price) |
+-------------+
|      320000 |
+-------------+
1 row in set (0.011 sec)

MariaDB [dealer]>
```

Figure 22: Query 8

Query 9: SELECT * FROM departments INNER JOIN staffs ON departments.department_id=staffs.department_id;

This query selects all rows from both tables as long as there is a match between the columns.

```
MariaDB [dealer]> SELECT * FROM departments INNER JOIN staffs ON departments.department_id=staffs.department_id;
+---------------+-------------------------+-----------------------+-------------+----------+------------+-------------------+---------------+
| department_id | department_name         | Email                 | address     | staff_id | contact    | Email             | department_id |
+---------------+-------------------------+-----------------------+-------------+----------+------------+-------------------+---------------+
|             1 | sales department        | salesdept12@gmail.com | durbarmarg  |      111 | 2147483647 | harry12@gmail.com |             1 |
|             2 | management department   | managedept12@gmail.com | durbarmargh |      112 | 2147483647 | sam33@gmail.com   |             2 |
|             3 | costumer care department | costdept12@gmail.com  | lazimpath   |      113 | 2147483647 | raju44@gmail.com  |             3 |
|             4 | Technical department    | techdept12@gmail.com  | lazimpath   |      114 | 2147483647 | ram88@gmail.com   |             4 |
|             5 | financial department    | techdept12@gmail.com  | durbarmargh |      115 | 2147483647 | erik67@gmail.com  |             5 |
+---------------+-------------------------+-----------------------+-------------+----------+------------+-------------------+---------------+
5 rows in set (0.000 sec)

MariaDB [dealer]>
```

Figure 23: Query 9

Query 10: SELECT * FROM orders RIGHT JOIN items ON orders.order_id = items.order_id;

This query returns all records from the right table (items), even if there are no matches in the left table (orders).

```
MariaDB [dealer]> SELECT * FROM orders RIGHT JOIN items ON orders.order_id = items.order_id;
+----------+------------+-----------+-----------+---------+---------+-------+----------+
| order_id | order_date | item_name | client_id | item_id | company | price | order_id |
+----------+------------+-----------+-----------+---------+---------+-------+----------+
|      501 | march 2    | Laptop    |       991 |     441 | apple   | 70000 |      501 |
|      502 | april 22   | mobile    |       992 |     442 | samsung | 50000 |      502 |
|      503 | april 3    | ipod      |       993 |     443 | apple   | 50000 |      503 |
|      504 | july 1     | ipad      |       994 |     444 | samsung | 60000 |      504 |
|      505 | june 30    | monitor   |       995 |     445 | samsung | 90000 |      505 |
+----------+------------+-----------+-----------+---------+---------+-------+----------+
5 rows in set (0.000 sec)

MariaDB [dealer]>
```

Figure 24: Query 10

Query 11: SELECT * FROM ITEMS WHERE price BETWEEN 40000 and 70000;

This query selects column (price) from items whose price is between 40000 and 70000.

```
MariaDB [dealer]> SELECT * FROM ITEMS WHERE price BETWEEN 40000 and 70000;
+---------+---------+-------+----------+
| item_id | company | price | order_id |
+---------+---------+-------+----------+
|     441 | apple   | 70000 |      501 |
|     442 | samsung | 50000 |      502 |
|     443 | apple   | 50000 |      503 |
|     444 | samsung | 60000 |      504 |
+---------+---------+-------+----------+
4 rows in set (0.008 sec)
```

Figure 25: Query 11

Conclusion

After the development of this coursework, I learned many things about the database management system (DBMS). All the study and research has helped me complete this coursework. At first, I created rough tables and corrected it in many ways. Then later, I created the planned tables by working in MySQL. I learned different queries required to create a table and insert values into it. Similarly I also learned many other queries while doing coursework like altering the table by adding columns, adding values into the new column and deleting rows and columns. Likewise, I learned to create Entity-Relationship diagram (ERD) and Relational diagram which are important part of database representation. The concept of data dictionary was also very clear for me only after going through lecture slides and after creating data dictionary of all the entities of the database. The values in the database can be changed and updated as required. Likewise, another important thing I learned in this relational database management system (RDBMS) is the use of primary and foreign key. Primary keys help uniquely identify each row in the column and foreign key connects two or more tables by referencing to the primary key of another table in the same database. Similarly, I also learned to use the attributes like Auto_Increment, Not Null, etc.

There were many queries used in this database. Going through lecture slides and research in the internet, I learned to type queries to get the required result in the database. I was able to work with the database which I have created.

Even though I faced some difficulties and problems, I did my best to overcome them through many research and practice. This assignment has helped me further in my research skill, report making and helped me understand how relational database management system works (RDBMS).

References

Anon. (2019) *Techopedia* [Online]. Available from: https://www.techopedia.com/definition/5547/primary-key.

Machajewski. (2017) *Information Management*.

Techopedia. (2019) [Online]. Available from: https://www.techopedia.com/definition/5547/primary-key.

Techtarget. (2019) [Online]. Available from: https://searchdatamanagement.techtarget.com/definition/entity-relationship-diagram-ERD.