Laboratory 3

Title of the Laboratory Exercise: data model to relational model

1. Introduction and Purpose of Experiment

   The ER schema is to be converted into a relational schema as data cannot be stored in an ER schema. A relation schema gives the basic information describing a table or relation. It is the logical definition of a table. This includes a set of column names, and the data types associated with each column. By doing this lab, students will be able to map ER schema to relational schema.

2. Aim and Objectives

   Aim

   - To map data model to relational model

   Objectives

   At the end of this lab, the student will be able to

   - Map ER schema to relational schema
   - Insert tuples using SQL commands for the developed database schema

3. Experimental Procedure

   i. Map all the components in the ER diagram to corresponding relation entities and instances
   ii. Insert tuples using SQL commands
   iii. Design SQL commands using aggregate functions in SQL
   iv. Execute SQL commands
   v. Test the executed commands
   vi. Document the Results
   vii. Analyse and discuss the outcomes of your experiment

4. Questions

   a. Consider the ER diagram you have drawn in Laboratory 2. Convert the ER diagram to corresponding relational database schema.
   b. Insert the tuples (minimum five) for the developed database schema using SQL commands. Perform aggregate functions in SQL based on the developed database schema.

5. Calculations/Computations/Algorithms

Updated ER diagram from Lab-2

The reason I've changed my ER diagram is there was lot of mistakes with my previous ER diagram so I wanted to correct it.

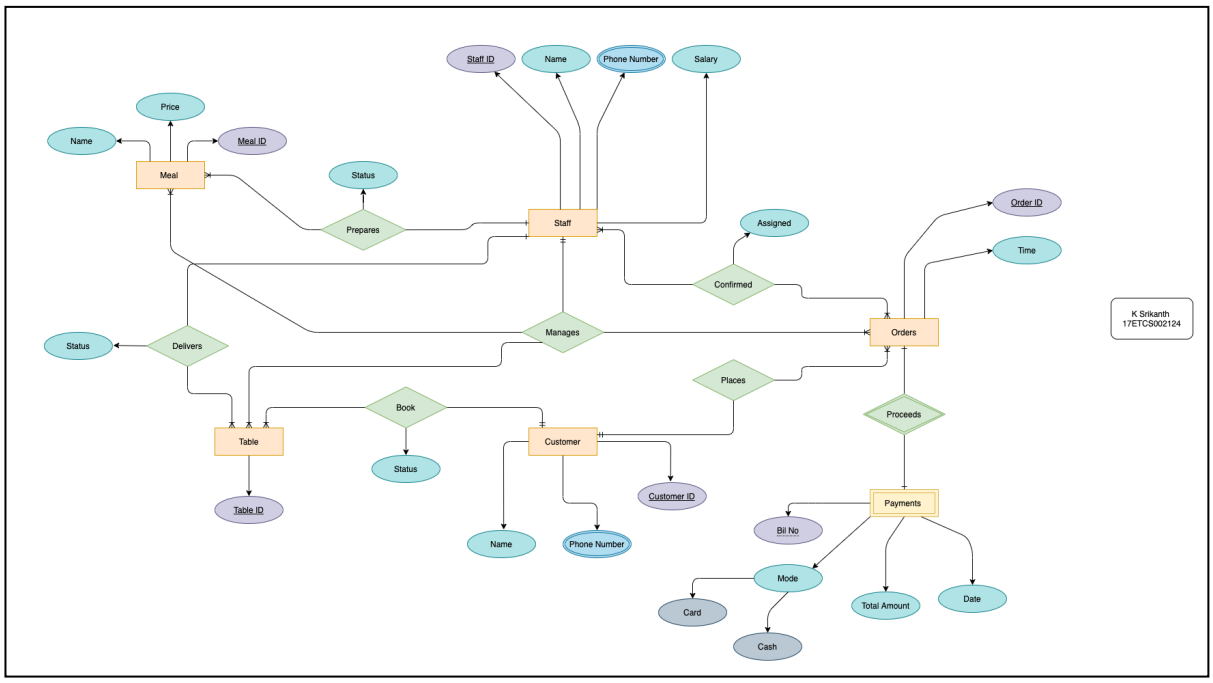

Image 1 Updated ER diagram from lab 2

Few changes from Lab 2 ER diagram

1. Added Attributes to **Book Relationship, Delivers Relationship, Prepares Relationship and Confirmed Relationship**
2. Added Multivalued Attributes for **Phone Numbers of Staff and Customer** entity.
3. Added Composite Attribute **(Mode)** to Payments entity.
4. Added N-Ary Relationship with **Staff, Meal, Orders** and table entity.
5. Added a relationship between orders and payments.
6. Changed Attribute Names to make it look Sensible and deleted few which were not needed.

ER Diagram to Relational Schema

Step - 1

Convert all the strong entities into tables

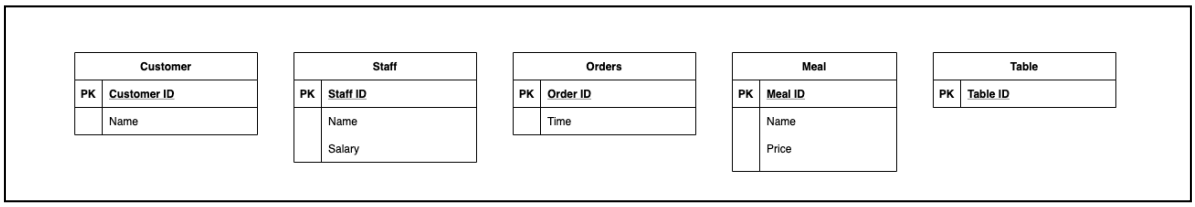

Image 2 Converting Strong Entities into table from Image 1

In Step 1 we just convert strong entities in to tables. If they have any multivalued or derived attributes will not be mentioned but multivalued attributes will have a separate table.

Step - 2

Convert all the weak entities into tables

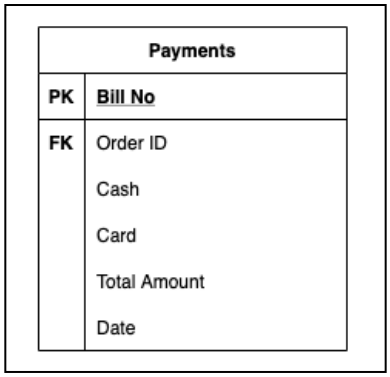

Image 3 Converting weak Entity into table from Image 1

In Step 2 we convert all the weak entities into a table in my case I only have not weak entity which is “Payments” and we also include a foreign key which inherits from its parent entity.

Step - 3

Convert all the 1:1 relations into tables

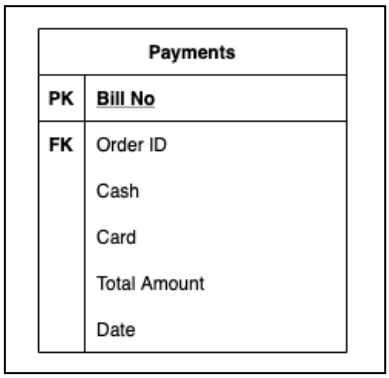

Image 4 Converting 1:1 Relation into table from Image 1

In Step 3 we would be converting all the 1:1 relations into table in my case we only have one relation 1:1 “Payments”.

Step - 4

Convert all the 1:N relations into tables

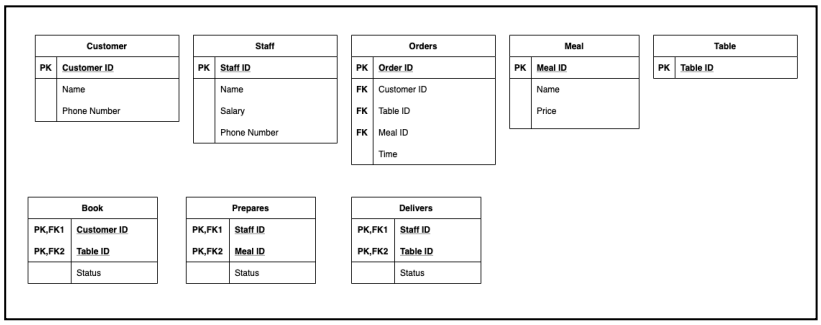

Image 5 Converting 1:N Relation into table from Image 1

In Step 4 we would be converting all the 1:N relations into table which basically means that adding foreign keys if your relation supports. In my case one customer can book only one table.

Step - 5

Convert all the M:N relations into tables

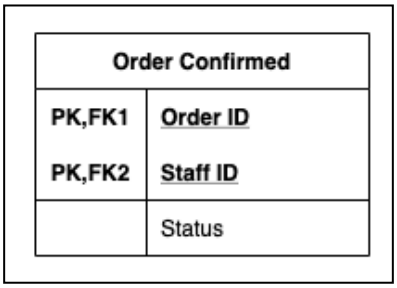

Image 6 Converting M:N Relation into table from Image 1

In Step 5 we would be converting all the M:N relations into table which has a separate table and have both the primary keys from the entities and also a attributes (if it has a attributes) in my case M number of orders can be assigned to N number of staff members. If an order assigns then it will have an Order ID and to whom its been assigned (Staff ID) with status.

Step - 6

Convert all the Multivalued attributes into tables

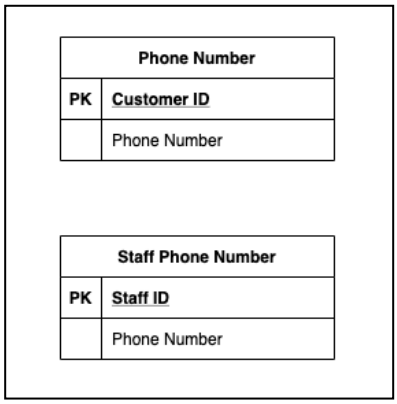

Image 7 Converting Multivalued Attributes into table from Image 1

In Step 6 we should convert all the multivalued attributes into a separate table so the primary key for the new table will be as a foreign key in the actual table just like shown in image 6.

Step - 7

Convert all the N-Ary relationships into tables

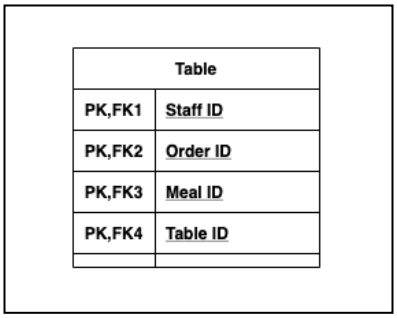

Image 8 Converting N-Ary
Relations into table from Image 1

In Step 7 we should convert all the N-Ary relationships into a table with all the primary keys from those entities which are connected to this relationship. Here only one Staff Member manages all the other entities.

Relational Schema

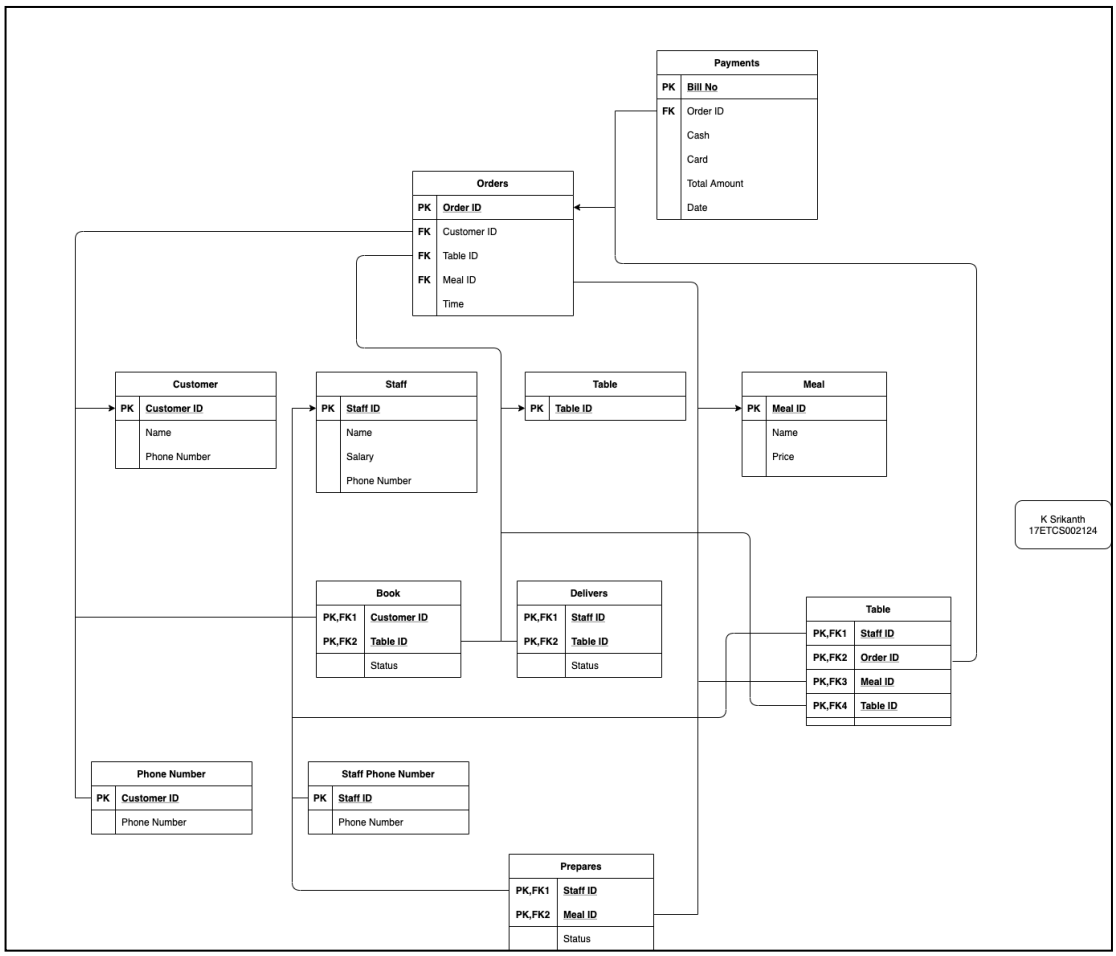

Image 8 Complete Relational Schema of ER diagram (Image 1)

6. Presentation of Results

Creating a Table **Staff in MySQL**

```
mysql> create table Staff(Staff_ID int auto_increment,Staff_Name varchar(30) not null,Salary int not null,primary key(Staff_ID));
Query OK, 0 rows affected (0.03 sec)

mysql> describe staff;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| Staff_ID   | int         | NO   | PRI | NULL    | auto_increment |
| Staff_Name | varchar(30) | NO   |     | NULL    |                |
| Salary     | int         | NO   |     | NULL    |                |
+------------+-------------+------+-----+---------+----------------+
3 rows in set (0.01 sec)

mysql> insert into staff(Staff_Name,Salary) values ("Ashwin",50000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into staff(Staff_Name,Salary) values ("Vignesh",60000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into staff(Staff_Name,Salary) values ("Sagar",40000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into staff(Staff_Name,Salary) values ("Pawan",45000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into staff(Staff_Name,Salary) values ("Raja",55000);
Query OK, 1 row affected (0.00 sec)

mysql> select * from staff;
+----------+------------+--------+
| Staff_ID | Staff_Name | Salary |
+----------+------------+--------+
|        1 | Ashwin     |  50000 |
|        2 | Vignesh    |  60000 |
|        3 | Sagar      |  40000 |
|        4 | Pawan      |  45000 |
|        5 | Raja       |  55000 |
+----------+------------+--------+
5 rows in set (0.00 sec)
```

Image 9 MySQL Commands to create table “Staff” and add values.

Creating a multivalued attribute table with a foreign key

Creating a Table **Staffnumber in MySQL**

```
mysql> create table staffphone( Staff_ID int not null, Phone_Number bigint not null, foreign key (Staff_ID)  references staff(Staff_ID));
Query OK, 0 rows affected (0.01 sec)

mysql> describe staffphone;
+--------------+--------+------+-----+---------+-------+
| Field        | Type   | Null | Key | Default | Extra |
+--------------+--------+------+-----+---------+-------+
| Staff_ID     | int    | NO   | MUL | NULL    |       |
| Phone_Number | bigint | NO   |     | NULL    |       |
+--------------+--------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

Image 10 MySQL Commands to create table “Staffnumber” with a foreign key.

```
mysql> insert into staffphone values(3,1234567890);
Query OK, 1 row affected (0.01 sec)

mysql> select * from staffphone;
+----------+--------------+
| Staff_ID | Phone_Number |
+----------+--------------+
|        1 |   9493364308 |
|        3 |   1234567890 |
+----------+--------------+
2 rows in set (0.00 sec)

mysql> delete from staff  where Staff_Id =3;
Query OK, 1 row affected (0.01 sec)

mysql> select * from staff;
+----------+------------+--------+
| Staff_ID | Staff_Name | Salary |
+----------+------------+--------+
|        1 | Ashwin     |  50000 |
|        2 | Vignesh    |  60000 |
+----------+------------+--------+
2 rows in set (0.00 sec)

mysql> select * from staffphone;
+----------+--------------+
| Staff_ID | Phone_Number |
+----------+--------------+
|        1 |   9493364308 |
+----------+--------------+
1 row in set (0.00 sec)
```

Image 11 MySQL Commands to add data to the table "Staffphone" and deleting a data from "Staff" table.

As you can see in image 11, we are adding phone number to the Staff ID = 3 and its getting updated on the table and if we delete it from the parent table "Staff" where Staff ID = 3 its gets deleted from Staffphone table also

Aggregate Functions on Staff table.

Aggregate functions perform a calculation on a set of values and return a single value.

Consider the schema,

Staff (name, salary)

1. Count(*)

Returns the number of rows

```
mysql> select count(*) from staff;
+----------+
| count(*) |
+----------+
|        5 |
+----------+
1 row in set (0.01 sec)
```

Image 12 MySQL Command to count the number of rows.

2. Average

Returns the average value of a given column

```
mysql> select avg(Salary) from staff;
+-------------+
| avg(Salary) |
+-------------+
|  50000.0000 |
+-------------+
1 row in set (0.01 sec)
```

Image 13 MySQL Command to Average value of column.

3. Minimum

Returns the smallest value

```
mysql> select min(Salary) from staff;
+-------------+
| min(Salary) |
+-------------+
|       40000 |
+-------------+
1 row in set (0.01 sec)
```

Image 14 MySQL Command to min value in a column.

4. Maximum

Returns the largest value

```
mysql> select max(Salary) from staff;
+-------------+
| max(Salary) |
+-------------+
|       60000 |
+-------------+
1 row in set (0.00 sec)
```

Image 15 MySQL Command to max value in a column.

5. Sum

Returns the sum of the values

```
mysql> select sum(Salary) from staff;
+-------------+
| sum(Salary) |
+-------------+
|      250000 |
+-------------+
1 row in set (0.00 sec)
```

Image 16 MySQL Command to sum of a column.

7. Conclusions

To start off with there were lot of changes from Lab-2 ER diagram that has been mentioned in page 2 of this document. There are 7 steps to convert a ER diagram to a Relational Schema

1. Map Strong Entities

Convert all the strong entities into tables

2. Map Weak Entities

Convert all the weak entities into tables

3. Map 1:1 Relations

Convert all the 1:1 relations into tables

4. Map 1:N Relations

Convert all the 1:N relations into tables

5. Map N:M Relations

Convert all the N:M relations into tables

6. Map N-ary Relations

Convert all the N-ary relations into tables

7. Map Multivalued Attributes

Convert all the Multivalued attributes into tables.

After converting all the entities and relations in to a **Relational Schema** then we perform queries on the tables in our database according to our needs.

For now, Created a table staff and inserted 5 data into the table were Staff ID is auto incremented which means that we don't have to specify the ID it increments from "0" at start. Then we created another table called "Staffphone" which was a multivalued attribute from staff entity. so, it has separate table and we have a foreign key which would be Staff ID so and then we have phone number type<bigint> and we enter the data if the Staff ID doesn't exist then it gives a error. Now if we try to delete the staff from the main table the data here will also be deleted as seen in image 11. Finally, we performed some aggregate functions

1. **Count :** which counts the number of rows in a column
2. **Average :** Which returns the average of the entire column(*if its a number)
3. **Minimum:** Which returns the minimum number in the entire column
4. **Maximum** : Which returns the maximum number in the entire column
5. **Sum :** Which returns the sum of all the values in entire column

8. Comments

1. Limitations of Experiments

ER diagram doesn't specify all the requirements need to make a perfect database so we drew relational schema

2. Limitations of Results

There are some Data types can be used to do some calculations like money but we used int to make it easy etc.

3. Learning happened

a. Transforming ER diagram to Relational Schema

b. Learned to map foreign key in a table on MySQL

c. Aggregate Functions in MySQL