Table of Contents

Part A

Page No.

1. Overview of Business Environment………………………………………………………………………………………2

2. Project Objectives………………………………………………………………………………………………………………..3

3. ER Diagram…………………………………………………………………………………………………………………………..4

4. Normalization up to 3rd Normal Form…………………………………………………………………………………..5

5. Description of Tables with Screenshots………………………………………………………………………………..7

6. Q&A/Scenarios/Queries in Application Domain with Screenshots………………………………………11

Part B

1. Learning Experience……………………………………………………………………………………………………………16

2. Problems and Lessons Learned…………………………………………………………………………………………..16

3. Achievements and Outcomes……………………………………………………………………………………………..16

4. Future Plans, Thoughts and Comments…………………………………………………………………………...….16

Part A

**1. Overview of Business Environment:**

The idea of the project was to develop a database management system for a restaurant to operate both within the restaurant and online.

We wanted to build an integrated system to handle information corresponding to service providers at different levels and customers. It was targeted that the system must be flexible to suit to any changes made afterwards. All information must be easy to understand and readily available. Restaurant businesses require keeping track of orders, use and availability of resources, employee status, delivery status and end user feedback.

It was decided to start from scratch and begin with basics. The primary components were drafted to create a foundation. This draft was later modified and detailed to create a fundamental infrastructure for the database management system. The model is suitable for adjusting to changes. Priority was given to fetch data readily and clearly.

The business environment was developed to have a correlation between customers and employees. Considering this factor, the essential information required in both ends will be clearly stated. The data sourcing is such that data will be available to employees in the form of orders and customers in the form of a bill.

A descriptive idea of the business model is as follows:

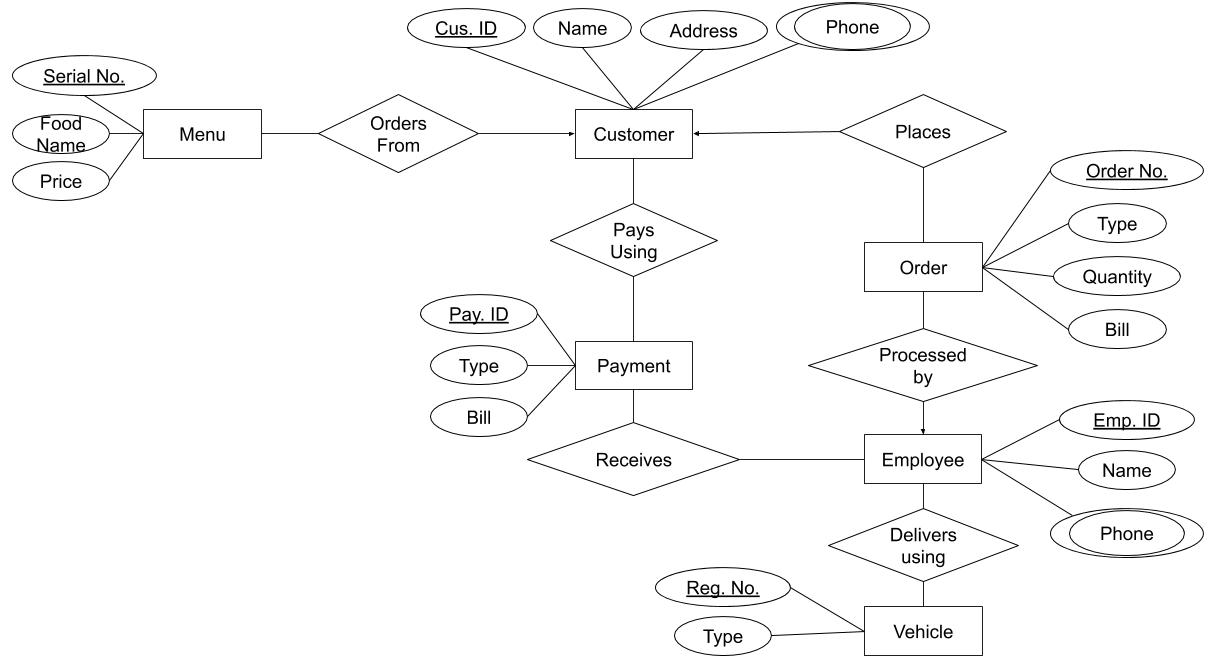
Customers can view food items from the menu. The menu will be available at the restaurant as well as online. They can place their desired items with respect to quantities in the form of an order. The orders will be received by employees for processing. Once confirmed, customers will receive a bill. If the order is placed online, the bill will include the information of the employee delivering that order. The payment can be done at the restaurant or online via cash, credit or mobile payment services. Customers will receive a confirmation of their payment.

Certain limitations were faced while constructing the infrastructure. These include availability of beverages items on online orders, vehicle information available to the customer, customization to the food items and keeping record of customer feedback. However, these issues were minimized as much as possible.

**2. Project Objectives:**

* To make all data readily available
* To make all data clear and uncomplicated
* To model the compiled specifications to the employees via order
* To model the compiled specifications to the customers via bill
* To make option for customers to order multiple food and in multiple quantities
* To assign delivery options to the employees
* To keep record of customers using a unique customer ID
* To keep track of employees using a unique employee ID
* To make employee and customer phone numbers available to each other for contact
* To make customer address available to employee for delivery

**3. ER Diagram:**



**4. Normalization up to 3rd Normal Form:**

**1. ORDER**

(SL\_NO,FOOD\_NAME,PRICE ,C\_ID,C\_NAME,C\_ADDRESS,C\_PHONE,ORDER\_NO ,

ORDER\_NO,ORDER\_TYPE , FOOD\_NAME,QUANTITY,BILL)

1NF : quantity,c\_phone,Food\_name is the multivalued attribute.

2NF: SL\_NO,FOOD\_NAME,PRICE ;

C\_ID,C\_NAME,C\_ADDRESS,C\_PHONE;

ORDER\_NO,ORDER\_TYPE , FOOD\_NAME,QUANTITY,BILL;

3NF : SL\_NO,FOOD\_NAME;

FOOD\_NAME,PRICE;

C\_ID,C\_NAME,C\_ADDRESS,C\_PHONE;

ORDER\_NO,ORDER\_TYPE ;

FOOD\_NAME,QUANTITY,BILL;

Table for prepared by:

C\_ID,C\_NAME,C\_ADDRESS,C\_PHONE;

SL\_NO,FOOD\_NAME;

FOOD\_NAME,PRICE;

ORDER\_NO,ORDER\_TYPE ;

ORDER\_NO FOOD\_NAME,QUANTITY,BILL ;

**2. Employee\_information(EMP\_ID,EMP\_NAME,EMP\_PHONE, REG\_NO,VEHICLE\_TYPE)**

1NF**:** EMP\_PHONE is the multivalued attribute.

2NF: EMP\_ID,EMP\_NAME,EMP\_PHONE;

REG\_NO,VEHICLE\_TYPE;

3NF : No transitive dependency.

Table for prepared by:

EMP\_ID,EMP\_NAME,EMP\_PHONE;

REG\_NO,VEHICLE\_TYPE, EMP\_PHONE;

**3. Payment (C\_ID,C\_NAME,C\_ADDRESS,C\_PHONE, PAY\_ID,PAY\_TYPE,BILL)**

1NF: C\_PHONE is the multivalued attribute.

2NF: C\_ID,C\_NAME,C\_ADDRESS,C\_PHONE;

PAY\_ID,PAY\_TYPE,BILL;

3NF: No transitive dependency.

Table for prepared by:

C\_ID, PAY\_ID,PAY\_TYPE,BILL;

**Final\_Table :**

1. Customer=**C\_ID**,C\_NAME,C\_ADDRESS,C\_PHONE,**ORDER\_NO**
2. Menu=**SL\_NO**,FOOD\_NAME,PRICE
3. Order=**ORDER\_NO**,FOOD\_NAME,QUANTITY,**SL\_NO**
4. Order\_T= **ORDER\_NO**,ORDER\_TYPE,T\_ITEM
5. Bill=**C\_ID,ORDER\_NO**,BILL
6. Employee = **EMP\_ID**,EMP\_NAME,EMP\_PHONE
7. Delivery =**C\_ID**,**EMP\_ID**,**REG\_NO**,VEHICLE\_TYPE
8. PAYMENT **= C\_ID**,**PAY\_ID**,PAY\_TYPE,BILL

**5. Description of Tables with Screenshots:**

1. CUSTOMER TABLE: This Table includes data of all customers and the orders they have placed. The primary key of this table is the Customer ID or C\_ID.

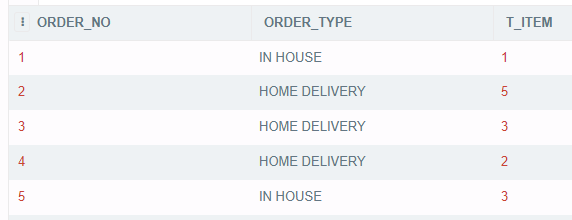


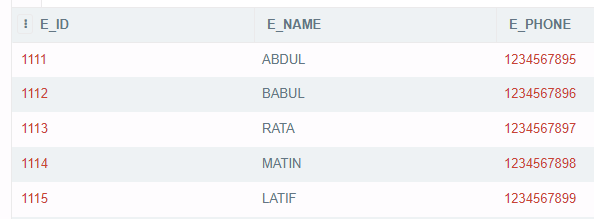
1. MENU TABLE: This table is the Menu of the restaurant.



1. ORDER TABLE: This table is the list of orders the customers have placed 

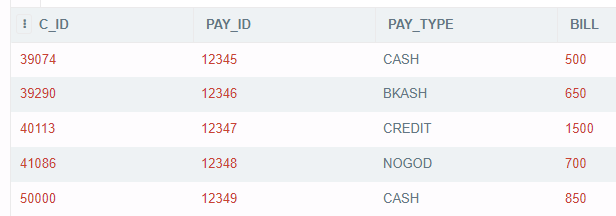
4. ORDER TYPE TABLE(ORDER\_T): This table contains the data of the type of order given by customers. IN HOUSE refers to orders in the restaurant while HOME DELIVERY refers to online orders.



5. EMPLOYEE TABLE: This table contains information of all employees. The primary key is the Employee ID or E\_ID.

6. BILL TABLE: This table contains the bills corresponding to the Customer ID numbers.



7. PAYMENT TABLE: This table contains the information of the payment done by the customers corresponding to a unique Payment ID or PAY\_ID.

8. DELIVERY TABLE: This Table contains the information of the delivery of online orders. Here, REG\_NO refers to vehicles assigned to the employees for delivery.



**6. Q&A/Scenarios/Queries in Application Domain with Screenshots:**

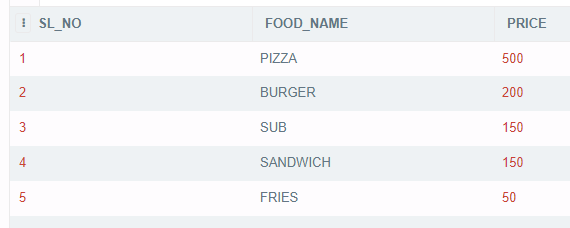
**1. Display the customer name and order from Customer table**

Command: SELECT C\_NAME,ORDERNO FROM CUSTOMER;



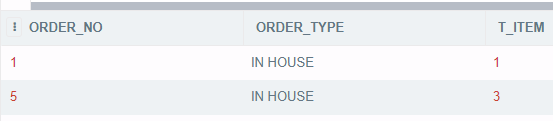
**2. Display the menu table**

Command: SELECT \* FROM MENU;



**3. Display the orders which are only IN HOUSE**

Command: SELECT \* FROM "ORDER" WHERE ORDER\_TYPE='IN HOUSE';



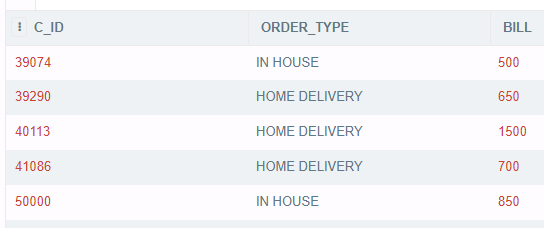
**4. Display order number, food name and quantity from order table**

Command: SELECT ORDER\_NO,FOOD\_NAME,QUANTITY FROM "ORDER";



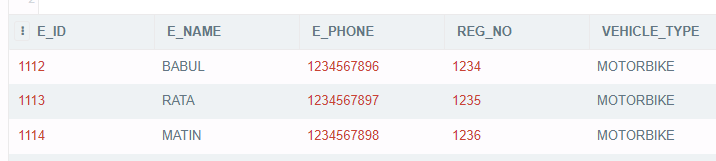
**5. Perform an equijoin between Order type table and Bill table and display Customer ID, Order Type and Bill**

Command: SELECT C\_ID , ORDER\_TYPE,BILL FROM "ORDER",BILL WHERE BILL.ORDER\_NO ="ORDER".ORDER\_NO;



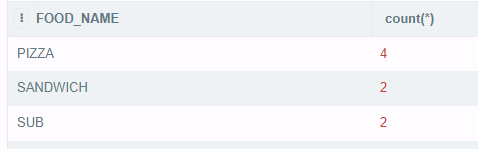
**6. Perform an equijoin between Employee table and Delivery table and show the Employee details and information of those who have been assigned vehicles for delivery**

Command: SELECT EMPLOYEE.\*,REG\_NO,VEHICLE\_TYPE FROM EMPLOYEE,DELIVERY WHERE EMPLOYEE.E\_ID=DELIVERY.E\_ID;



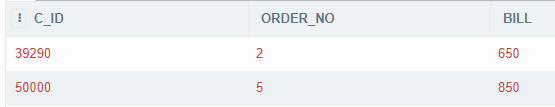
**7. Display the food name which food ordered more than one time, using group function**

Command: SELECT ORDER\_NO,COUNT(QUANTITY) FROM "ORDER" GROUP BY ORDER\_NO HAVING COUNT(\*)>1;



**8. Perform an equijoin between Order table and Bill table and show total bill of those who ordered sub and sandwich**

Command: SELECT BILL.\* FROM "ORDER",BILL WHERE BILL.ORDER\_NO ="ORDER".ORDER\_NO AND FOOD\_NAME IN('SUB','SANDWICH');



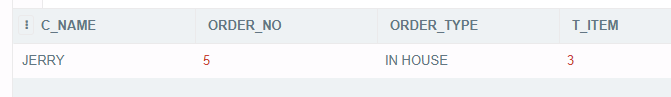
**9. Show the customer details of the customers who ordered max quantity of food items**

Command: SELECT CUSTOMER.\* FROM CUSTOMER,"ORDER" WHERE CUSTOMER.ORDERNO="ORDER".ORDER\_NO AND QUANTITY=(SELECT MAX(QUANTITY)FROM "ORDER");



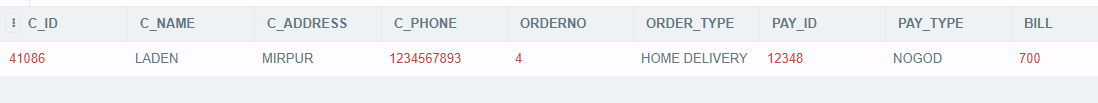
**10. Show the order details of the customer who paid the second highest bill**

Command: SELECT C\_NAME,ORDER\_T.\* FROM CUSTOMER,ORDER\_T WHERE CUSTOMER.ORDERNO=ORDER\_T.ORDER\_NO AND ORDER\_NO=(SELECT ORDER\_NO FROM BILL WHERE BILL=(SELECT MAX(BILL) FROM BILL WHERE BILL<(SELECT MAX(BILL) FROM BILL )));



**11. Show the customer information, order type and payment information of the customers who paid using bKash or Nogod and 3rd lowest bill.**

Command: SELECT CUSTOMER.\*,ORDER\_TYPE,PAY\_ID,PAY\_TYPE,BILL FROM CUSTOMER,PAYMENT,ORDER\_T WHERE CUSTOMER.ORDERNO=ORDER\_T.ORDER\_NO AND CUSTOMER.C\_ID=PAYMENT.C\_ID AND CUSTOMER.C\_ID=(SELECT C\_ID FROM PAYMENT WHERE PAY\_TYPE IN ('BKASH','NOGOD') AND BILL=(SELECT MIN(BILL) FROM BILL WHERE BILL>(SELECT MIN(BILL) FROM BILL WHERE BILL>(SELECT MIN(BILL) FROM BILL ))));



Part B

**1. Learning Experience:**

This project has helped us grow an insight towards database management systems in various businesses. The development and implementation of a database management system requires thorough planning and interrelation of data. Furthermore, data must be arranged in such a way that it may be easily readable and retrievable.

**2. Problems and Lessons Learned:**

Some of the problems that we faced during development of this database are including certain types of data due to environment restrictions and the facilities and resources to deliver them. This has retargeted our focus to simplify things and trim off parts which were not possible to implement with the knowledge we have.

**3. Achievements and Outcomes:**

We have been able to design and develop a database that can be used as a base model for a restaurant business. We have designed it in such a way that there are scopes for changes and extensions.

For example, a new segment can be added with relation to employees under the tag “resource management”. This can help with the overview of availability, use and sourcing raw materials required for the business.

**4. Future Plans, Thoughts and Comments:**

Future plans for the database management system include extension of the data tables to hold records of more information for better management. Doing so can help to collaborate with third party service providers and increase the domain of business. That is, one business can help another. However, with the limitation of our current knowledge and experience, we feel that it is important to learn, observe and apply more in the field of database management system.

This future plan concludes to our final thought, “experience can help with expertise”.