

Easy Eatery - An Online Food Delivery System

Course: Database Systems
Course Code: CSE 304 (Section 2)

Submitted By

* Jaohar Raihan (193014035)
Sadia Afrin Ridita (193014035)
Aditi Bhattacharya (193014026)

Submitted To

Muhammad Golam Kibria, PhD
Department of Computer Science and Engineering

Summer 2021
Department of Computer Science and Engineering

Table of Index

	Section title	Page No.
1.	Introduction of the Database	3
1.1	Easy Eatery	3
1.2	Scopes Of Our Proposed Database	3
1.3	Motivation	3
1.4	Entities of the proposed DB	4
2	Database project objectives	4
3	ERD of the Database	5
4	Normalization	6
5	Relationship diagram	7
6	Screenshots of the Tables	8
7	View results	9
7.1	Complex View (SQL JOIN)	10
7.2	Simple View	12
8	BONUS	15

1. Introduction of the Database

A database is a collection of interconnected data that allows for efficient retrieving data, insertion, and deletion from a database and organizes the data into tables, views, schemas, and reports. A database management system (DBMS) is a software program that allows to create and manage databases.

Easy Eatery :

We named our project as “Easy Eatery” .It’s An Online Food Delivery System system.

- This system is a bunch of benefits from various points of view. This online application enables the users to register to the system online, select the food items of their choice from the menu list, and order food online. Also, the payment can be made through online mode or at the time of home delivery depending upon the customer’s choice and convenience.
- The selection made by the customers will be available to the hotel reception or to the person handling the work assignment. Now this same person will assign the orders to the specialist chef to be completed within a fixed duration of time.
- There will be a system administrator who will have the right to add and manage user accounts.

Scopes Of Our Proposed Database :

- Our Food Ordering app can sale Food product, preferred brands, kitchen needs, essential restaurant supplies and more through this online, onestop Food store.
- Restaurant owners can use this app as one big super market app to sale product of their store.
- This app make easy for user to buy product from store with easy steps and store can get easy order.
- There is a great oppurtunity for the delivery persons as well to work with the system

Motivation :

- Easy Eatery can be very useful application for people during this pandemic period undoubtedly as our motive is to serve the food all over the country maintaining all the safety,hygene and precautions.

- Customers can avail the discount ,offers and the prices of food will be accurate .So there's no chance to be cheated
- As there is Admin Panel and delivery panel ,Easy Eatery can be a perfect workplace for the applicants who're seeking this type of jobs

Entities of the proposed DB :

- User Information
- Delivery Person
- Delivery process
- Food Item
- Order information
- Payment information
- Offer Information

2. Database project objectives

There are some objectives that must be achieved by developing this system. These objectives are implemented through the use of some sql queries by using XAMPP software. These objectives are as follows:

1. Provide the Searching Facility based on food category, location.
2. Customers will be able to gather more and reliable information about different foods and relevant offers, discounts.
3. Track the order and delivery process
4. Provide the customer and Restaurant information properly
5. Low cost software implementation and maintain
6. Easy, Reliable and hassle free Payment system

There are also some business utilities of this project. According to [Dhaka Tribune](#) ,There are about 60,000 big/small restaurants in all over the Bangladesh.Out of them, about 10000 restaurants are in Dhaka([Financial Express](#)) . So we can implement our project properly in atleast some popular ,specific area , there would be huge profit during this pandemic and busy day-life as well.

3. ERD of the Database

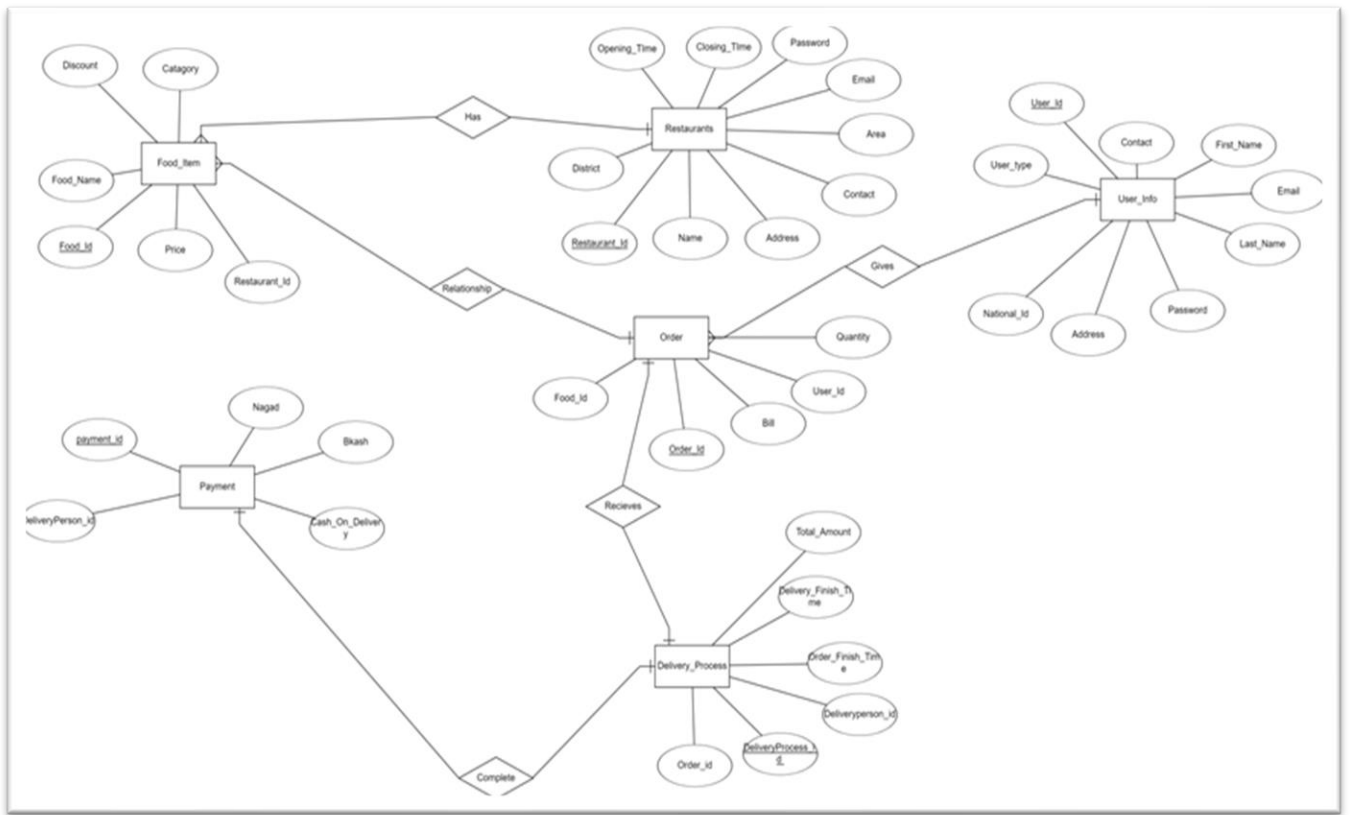
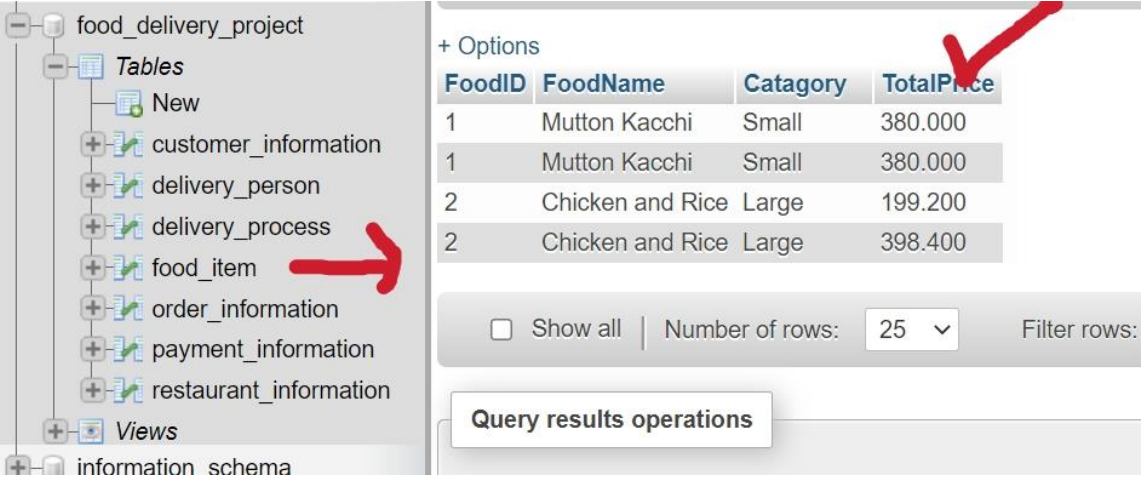


Diagram 1 – ERD Of Online Food Delivery System

4. Normalization:

All table of the database have created considering First step of normalization (1NF)

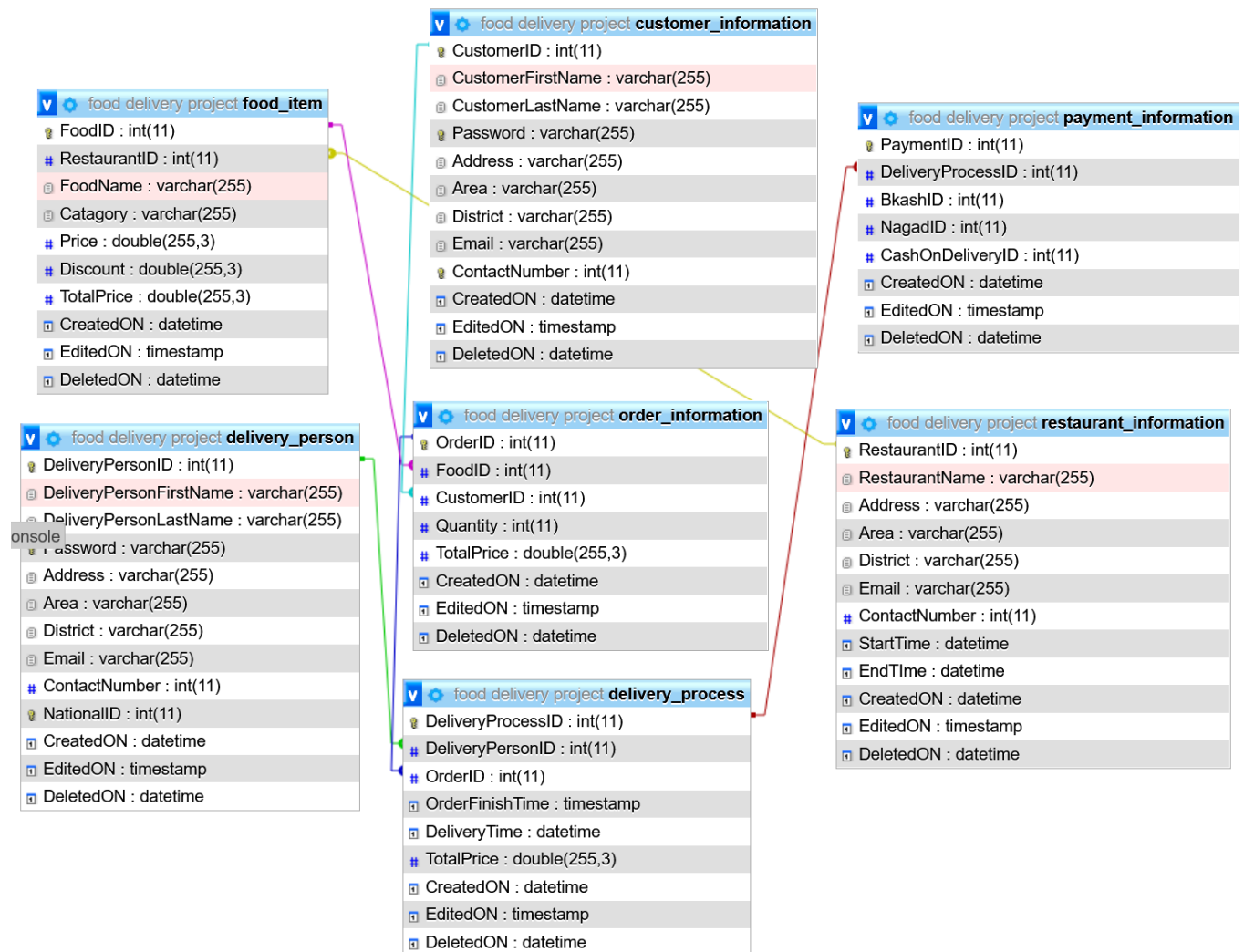
Screenshot



The screenshot displays a database management interface. On the left, a tree view shows the 'food_delivery_project' database containing a 'Tables' folder with a 'New' button and several existing tables: 'customer_information', 'delivery_person', 'delivery_process', 'food_item', 'order_information', 'payment_information', and 'restaurant_information'. A red arrow points to the 'food_item' table. On the right, the 'Query results' pane shows a table with 4 columns: 'FoodID', 'FoodName', 'Catagory', and 'TotalPrice'. A red arrow points to the 'TotalPrice' column. The table contains 6 rows of data. Below the table, there are controls for 'Show all', 'Number of rows' (set to 25), and 'Filter rows'. A 'Query results operations' button is also visible.

FoodID	FoodName	Catagory	TotalPrice
1	Mutton Kacchi	Small	380.000
1	Mutton Kacchi	Small	380.000
2	Chicken and Rice	Large	199.200
2	Chicken and Rice	Large	398.400

5. Relationship diagram :



6. Screenshots of the Tables

Customer_information table:

CustomerID	CustomerFirstName	CustomerLastName	Password	Address	Area	District	Email	ContactNumber
1	Jaohar	Raihan	000	8/4A	Dhanmondi	Dhaka	j.raihan@ulab.edu.bd	1605625602
2	Rupom	Saha	111	1/8	mohammadpur	Dhaka	rupom.saha.cse@ulab.edu.bd	1805625502
3	Sheikh	Sadia	222	2/8	tajmahal road	Dhaka	s.sadia.cse@ulab.edu.bd	1905625555
4	Md.	Ridita	333	5/15	adabor	Dhaka	ridita.chowdhury.cse@ulab.edu.bd	1305626265

Delivery_person table:

DeliveryPersonID	DeliveryPersonFirstName	DeliveryPersonLastName	Password	Address	Area	District	Email	ContactNumber
1	Tasnim	Alam	D555	20/6A	Dhanmondi	Dhaka	tasnim@ulab.edu.bd	1105625602
2	Ripon	Biswas	D666	21/7	mohammadpur	Dhaka	ripom@ulab.edu.bd	1005626622
3	Tajrean	Reza	D777	12/18	tajmahal road	Dhaka	tajrean@ulab.edu.bd	1905625555
4	Shimul	Chowdhury	D888	25/1	adabor	Dhaka	shimul.c@ulab.edu.bd	1990555555

Food_item table:

FoodID	RestaurantID	FoodName	Catagory	Price	Discount	TotalPrice
1	1	Mutton Kacchi	Small	399.990	19.990	380.000
2	1	Chicken and Rice	Large	199.200	0.000	199.200
3	2	Mutton Kacchi	Large	599.000	50.000	549.000
4	2	Murag Polao	Small	140.000	20.500	119.500

Order_information table:

OrderID	FoodID	CustomerID	Quantity	TotalPrice
1	1	1	1	380.000
2	1	2	1	380.000
3	2	2	1	199.200
4	2	3	2	398.400

Payment_information table:

PaymentID	DeliveryProcessID	BkashID	NagadID	CashOnDeliveryID
1	1	1683659364	0	0
2	2	0	1552854693	0
3	3	0	0	1
4	4	1956935888	0	0

Restaurant_information table:

RestaurantID	RestaurantName	Address	Area	District	Email	ContactNumber
1	Sultan Dine	2/6C	Dhanmondi	Dhaka	sultan@gmail.com	1682565953
2	Basmati Kacchi	12/18B	shamoly	Dhaka	Basmati@gmail.com	1925688888
3	Basmati Bhai	2/19A	Dhanmondi	Dhaka	Kacchi@gmail.com	1889988256
4	TakeOut2.O	2/15A	dhanmondi	Dhaka	takeout2.O@gmail.com	1335566932

7. View results:

There are 2 types of **Views** in SQL: **Simple View** and **Complex View**. **Simple views** can only contain a single base table. **Complex views** can be constructed on more than one base table. In particular, **complex views** can contain: join conditions, a group by clause, a order by clause.

I execute some Simple and Complex views from my database directly from 'Create View' with sql queries.

7.1 Complex View (SQL JOIN):

1. [SELECT](#) * FROM `customer_details_view`

CustomerID	CustomerFirstName	CustomerLastName	PASSWORD	Address	TotalPrice
1	Jaohar	Raihan	000	8/4A	380.000
2	Aditi	Bhattacharya	111	1/8	380.000
2	Aditi	Bhattacharya	111	1/8	199.200
3	Sadia	Afrin	222	2/8	398.400

Sql Queries :

SELECT

i.CustomerID,

i.CustomerFirstName,

i.CustomerLastName,

i.Password,

i.Address,

f.TotalPrice

FROM

customer_information AS i

JOIN order_information AS f

ON

i.CustomerID = f.CustomerID;

CREATE ALGORITHM = UNDEFINED VIEW `customer_information`

(CustomerID, CustomerFirstName, CustomerLastName, PASSWORD, Address, TotalPrice)

As SELECT i.CustomerID, i.CustomerFirstName, i.CustomerLastName, i.PASSWORD,
i.Address, f.TotalPrice FROM customer_information as i JOIN order_information as f ON
i.CustomerID = f.CustomerID;

2. [SELECT](#) * FROM `delivery_person_view`

ID	NAME	PASSWORD	Address	Area	OrderFinishTime	DeliveryTime
1	Washim	D555	20/6A	Dhanmondi	2021-05-15 15:14:44	2021-05-15 16:14:44
2	Faria	D666	21/7	mohammadpur	2021-05-24 00:30:33	2021-05-04 16:14:44
3	Tajrean	D777	12/18	tajmahal road	2021-05-24 00:31:14	2014-05-14 17:14:44
3	Tajrean	D777	12/18	tajmahal road	2021-05-24 00:27:51	2021-05-04 16:14:44

Sql Queries:

SELECT

i.DeliveryPersonID AS ID,
i.DeliveryPersonFirstName AS NAME,
i.Password,
i.Address,
i.Area,
s.OrderFinishTime,
s.DeliveryTime

FROM

`delivery_person` AS i

JOIN delivery_process AS s

ON

i.DeliveryPersonID = s.DeliveryPersonID;

CREATE ALGORITHM = UNDEFINED VIEW `delivery_person_view`(
ID,
NAME,
PASSWORD,
Address,
Area,
OrderFinishTime,
DeliveryTime
) AS SELECT

i.DeliveryPersonID AS ID,
i.DeliveryPersonFirstName AS NAME,
i.Password,
i.Address,
i.Area,
s.OrderFinishTime,
s.DeliveryTime

FROM

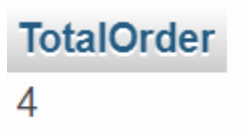
```

        delivery_person AS i
    JOIN delivery_process AS s
    ON
        i.DeliveryPersonID = s.DeliveryPersonID;

```

7.2 Simple View:

1. [SELECT](#) * FROM `passenger_simple_view`



TotalOrder
4

Sql Queries:

```
SELECT
```

```
    COUNT(OrderID)
```

```
FROM
```

```
    `order_information`;
```

```
CREATE ALGORITHM = UNDEFINED VIEW `order_view` (TotalOrder) AS SELECT
```







```
    COUNT(OrderID)
```

```
FROM
```

```
    order_information;
```

2. [SELECT](#) * FROM `food_item_view`

+ Options

				ID	NAME	Catagory
<input type="checkbox"/>	 Edit	 Copy	 Delete	Mutton Kacchi	1	Small
<input type="checkbox"/>	 Edit	 Copy	 Delete	Murag Polao	4	Small

Sql Queries :

```

SELECT
FoodID AS ID,
FoodName AS NAME,
Catagory AS Catagory
FROM
    food_item
WHERE
    Catagory = 'Small';

CREATE ALGORITHM = UNDEFINED VIEW 'food_item_view'(ID, NAME, Catagory) AS
SELECT
    FoodName AS NAME,
    FoodID AS ID,
    Catagory AS Catagory
FROM
    food_item
WHERE
    Catagory = 'Small';

```

3.[SELECT](#) * FROM `Food_item_simple_view`

FoodID	FoodName
2	Chicken and Rice
4	Murag Polao
1	Mutton Kacchi

Sql Queries:

SELECT

FoodID,

FoodName

FROM

food_item

GROUP BY

FoodName;

CREATE ALGORITHM = UNDEFINED VIEW `food_item_simple_view` (Total_FoodID, Food)

AS SELECT

FoodID,

FoodName

FROM

food_item

GROUP BY

FoodName;

8 BONUS:

(1)

FoodName

Chicken and Rice

SELECT

__food_item.FoodName

FROM

__food_item,

__order_information

WHERE

__food_item.FoodID = order_information.FoodID AND order_information.Quantity > 1;

2

`food_item_Order ID` simple view

+ Options

FoodID	RestaurantID	FoodName	Catagory	Price	OrderID
1	1	Mutton Kacchi	Small	399.990	1
1	1	Mutton Kacchi	Small	399.990	2
2	1	Chicken and Rice	Large	199.200	3
2	1	Chicken and Rice	Large	199.200	4

SELECT

```

    i.FoodID,
    i.RestaurantID,
    i.FoodName,
    i.Catagory,
    i.Price,
    f.OrderID
FROM
    food_item AS i
JOIN order_information AS f
ON
    i.FoodID = f.FoodID;
CREATE ALGORITHM = UNDEFINED VIEW `food_item_OrderID`(
    FoodID,
    RestaurantID,
    FoodName,
    Catagory,
    Price,
    OrderID
) AS SELECT
    i.FoodID,
    i.RestaurantID,
    i.FoodName,
    i.Catagory,
    i.Price,
    f.OrderID

```


FROM

food_item AS i

JOIN order_information AS f

ON

i.FoodID = f.FoodID;

3.

``restaurant_information_View``

RestaurantID	RestaurantName
3	Basmati Bhai
2	Basmati Kacchi
1	Sultan Dine
4	TakeOut2.O

SELECT

RestaurantID,

RestaurantName

FROM

restaurant_information

GROUP BY

RestaurantName;

CREATE ALGORITHM = UNDEFINED VIEW

``restaurant_information_View``(Total_RestaurantID, Restaurant) AS SELECT

```
    RestaurantID,  
    RestaurantName  
FROM  
    restaurant_information  
GROUP BY  
    RestaurantName;
```

4.

'payment_information_View'

ID	NAME	BkashID
2	2	0
3	3	0

```
SELECT  
    PaymentID AS ID,  
    DeliveryProcessID AS NAME,  
    BkashID AS BkashID  
FROM  
    payment_information  
WHERE  
    BkashID = '0';  
  
CREATE ALGORITHM = UNDEFINED VIEW 'payment_information_View'(ID, NAME,  
BkashID) AS SELECT
```

```

PaymentID AS ID,
DeliveryProcessID AS ID,
BkashID AS BkashID
FROM
    payment_information
WHERE
    BkashID = '0';

```

5. 'Total_restaurant_information_View'

+ Options

COUNT(RestaurantID)

4

```

SELECT

```

```

    COUNT(RestaurantID)

```

```

FROM

```

```

    `restaurant_information`;

```

```

CREATE ALGORITHM = UNDEFINED VIEW

```

```

'Total_restaurant_information_View'(TotalRestauran) AS SELECT

```

```
COUNT(RestaurantID)
```

```
FROM
```

```
restaurant_information;
```

```
END
```