

DBMS Assignment 2

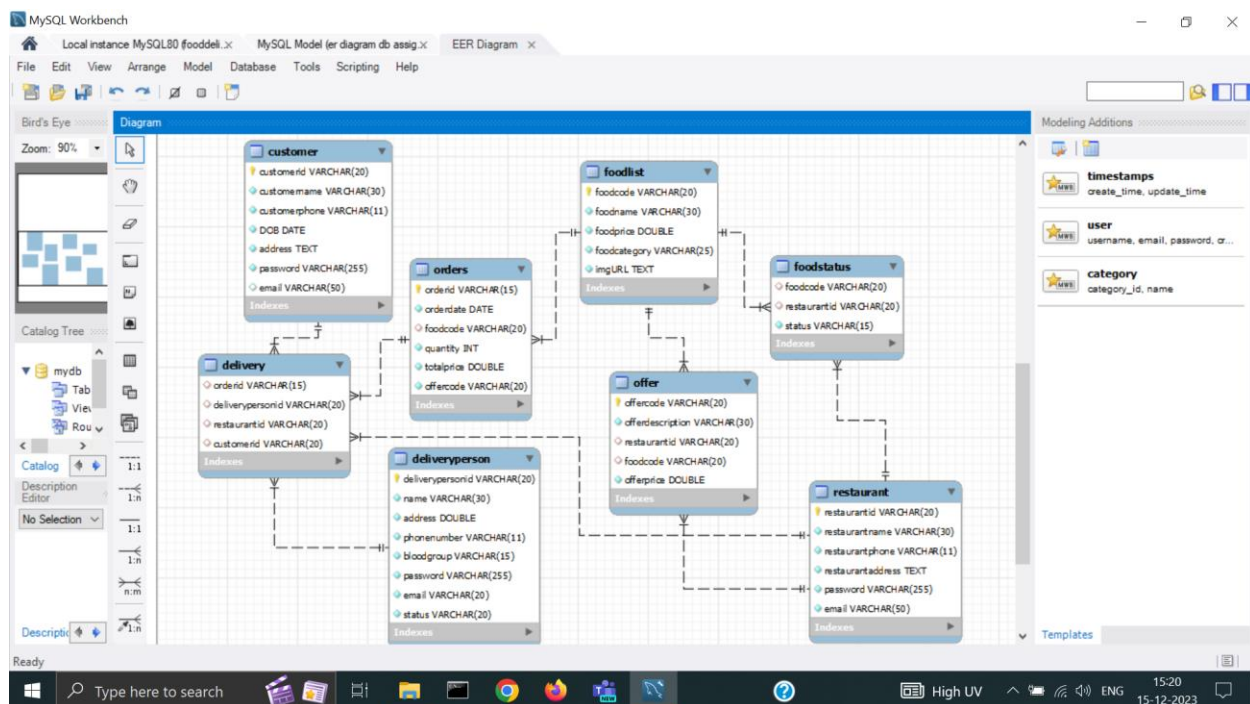
Binoy Barnabas

ILP Batch 1

Domain: Food delivery

This is the current trend or a method of purchasing food items through applications or websites which works as PaaS or SaaS. Food delivery services have become increasingly relevant in today's fast paced and digitally connected world. One of the main advantages of using these services is convenience where the users or customers can purchase food from any restaurants at any time anywhere without any need to go to be at the doorstep of restaurants. It also saves time and brings new economic opportunities in the form of new jobs for the delivery services, improves the sales at the restaurants and give equal opportunity to all the new and ongoing business to compete and provide the best food they possibly can. Food delivery services also increased accessibility in such a way that the users can buy or schedule their orders based on their need and location.

ER diagram of relations in the database foodelivery



Explanations of tables, their attributes and their relationships between another relations

Customer table

1. customerid (VARCHAR(20), PRIMARY KEY, NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 20 characters.

Constraint: Primary Key, which means it uniquely identifies each row in the table. It cannot be NULL.

2. customername (VARCHAR(30), NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 30 characters.

Constraint: Not NULL, indicating that this field must have a value.

3. customerphone (VARCHAR(11), NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 11 characters.

Constraint: Not NULL, indicating that this field must have a value.

4. DOB (DATE, NOT NULL):

Data Type: Date data type.

Constraint: Not NULL, indicating that this field must have a value.

5. address (TEXT, NOT NULL):

Data Type: TEXT, suitable for storing large amounts of text data.

Constraint: Not NULL, indicating that this field must have a value.

6. password (VARCHAR(255), NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 255 characters.

Constraint: Not NULL, indicating that this field must have a value.

7. email (VARCHAR(50), UNIQUE):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 50 characters.

Constraint: UNIQUE, ensuring that each email address in the table must be unique.

This table is designed to store information about customers. The customerid is the primary key, which uniquely identifies each customer. The email field is marked as UNIQUE, ensuring that each email address is unique across the table. The NOT NULL constraints ensure that certain fields must have values and cannot be left blank.

Restaurant table

1. restaurantid (VARCHAR(20), PRIMARY KEY, NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 20 characters.

Constraint: Primary Key, which means it uniquely identifies each row in the table. It cannot be NULL.

2. restaurantname (VARCHAR(30), NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 30 characters.

Constraint: Not NULL, indicating that this field must have a value.

3. restaurantphone (VARCHAR(11), NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 11 characters.

Constraint: Not NULL, indicating that this field must have a value.

4. restaurantaddress (TEXT, NOT NULL):

Data Type: TEXT, suitable for storing large amounts of text data.

Constraint: Not NULL, indicating that this field must have a value.

5. password (VARCHAR(255), NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 255 characters.

Constraint: Not NULL, indicating that this field must have a value.

6. email (VARCHAR(50), UNIQUE, NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 50 characters.

Constraint: UNIQUE, ensuring that each email address in the table must be unique.

Constraint: Not NULL, indicating that this field must have a value.

This table is designed to store information about restaurants. The restaurantid is the primary key, which uniquely identifies each restaurant. The email field is marked as UNIQUE, ensuring that each email address

is unique across the table. The NOT NULL constraints ensure that certain fields must have values and cannot be left blank

Foodlist table

1. foodcode (VARCHAR(20), PRIMARY KEY, NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 20 characters.

Constraint: Primary Key, which means it uniquely identifies each row in the table. It cannot be NULL.

2. foodname (VARCHAR(30), NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 30 characters.

Constraint: Not NULL, indicating that this field must have a value.

3. foodprice (DOUBLE, NOT NULL):

Data Type: Double-precision floating-point number for representing the price of the food.

Constraint: Not NULL, indicating that this field must have a value.

4. foodcategory (VARCHAR(25), NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 25 characters.

Constraint: Not NULL, indicating that this field must have a value.

5. imgURL (TEXT, NOT NULL):

Data Type: TEXT, suitable for storing large amounts of text data.

Constraint: Not NULL, indicating that this field must have a value.

This table is designed to store information about food items. The foodcode is the primary key, which uniquely identifies each food item. The foodprice is stored as a DOUBLE, suitable for representing prices with decimal values. The imgURL field is designed to store the URL of an image associated with the food item. The NOT NULL constraints ensure that certain fields must have values and cannot be left blank.

Deliveryperson table

1. deliverypersonid (VARCHAR(20), PRIMARY KEY, NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 20 characters.

Constraint: Primary Key, which means it uniquely identifies each row in the table. It cannot be NULL.

2. name (VARCHAR(30), NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 30 characters.

Constraint: Not NULL, indicating that this field must have a value.

3. address (DOUBLE, NOT NULL):

Data Type: Double-precision floating-point number. However, it seems like there might be an issue here; an address is typically a string, not a number. You might want to use a VARCHAR or TEXT data type here.

4. phonenumber (VARCHAR(11), NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 11 characters.

Constraint: Not NULL, indicating that this field must have a value.

5. bloodgroup (VARCHAR(15), NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 15 characters.

Constraint: Not NULL, indicating that this field must have a value.

6. password (VARCHAR(255), NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 255 characters.

Constraint: Not NULL, indicating that this field must have a value.

7. email (VARCHAR(20), NOT NULL UNIQUE):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 20 characters.

Constraint: Not NULL, indicating that this field must have a value.

Constraint: UNIQUE, ensuring that each email address in the table must be unique.

8. status (VARCHAR(20), NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 20 characters.

Constraint: Not NULL, indicating that this field must have a value.

Please note that I changed the data type of the address field to DOUBLE based on what you provided, but it seems unusual for an address. If an address is meant to be a textual value, consider using VARCHAR or TEXT instead. Adjust the data types according to your actual requirements.

Foodstatus table

1. foodcode (VARCHAR(20)):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 20 characters.

Note: This field is likely a reference to the foodcode in the foodlist table.

2. restaurantid (VARCHAR(20)):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 20 characters.

Note: This field is likely a reference to the restaurantid in the restaurant table.

3. status (VARCHAR(15), NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 15 characters.

Constraint: Not NULL, indicating that this field must have a value.

Foreign Key Constraints:

foodcode Foreign Key:

References the foodcode column in the foodlist table.

ON DELETE CASCADE: Specifies that if a referenced row in the foodlist table is deleted, the corresponding rows in the foodstatus table will also be deleted.

restaurantid Foreign Key:

References the restaurantid column in the restaurant table.

ON DELETE CASCADE: Specifies that if a referenced row in the restaurant table is deleted, the corresponding rows in the foodstatus table will also be deleted.

This table, foodstatus, is designed to maintain the status of food items in restaurants. The foodcode and restaurantid columns are used as foreign keys, establishing relationships with the foodlist and restaurant tables, respectively. The status column records the current status of the food item (e.g., available, out of stock). The ON DELETE CASCADE ensures referential integrity by automatically deleting related rows in the foodstatus table if corresponding rows in the referenced tables (foodlist or restaurant) are deleted.

Offer table

1. offercode (VARCHAR(20), PRIMARY KEY, NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 20 characters.

Constraint: Primary Key, which means it uniquely identifies each row in the table. It cannot be NULL.

2. offerdescription (VARCHAR(30), NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 30 characters.

Constraint: Not NULL, indicating that this field must have a value.

3. restaurantid (VARCHAR(20)):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 20 characters.

Note: This field is likely a reference to the restaurantid in the restaurant table.

4. foodcode (VARCHAR(20)):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 20 characters.

Note: This field is likely a reference to the foodcode in the foodlist table.

5. offerprice (DOUBLE, NOT NULL):

Data Type: Double-precision floating-point number for representing the price of the offer.

Constraint: Not NULL, indicating that this field must have a value.

Foreign Key Constraints:

foodcode Foreign Key:

References the foodcode column in the foodlist table.

ON DELETE CASCADE: Specifies that if a referenced row in the foodlist table is deleted, the corresponding rows in the offer table will also be deleted.

restaurantid Foreign Key:

References the restaurantid column in the restaurant table.

ON DELETE CASCADE: Specifies that if a referenced row in the restaurant table is deleted, the corresponding rows in the offer table will also be deleted.

This table, offer, is designed to store information about special offers provided by restaurants. The offercode is the primary key, uniquely identifying each offer. The offerdescription describes the details of the offer. The restaurantid and foodcode columns are used as foreign keys, establishing relationships with the restaurant and foodlist tables, respectively. The offerprice column represents the discounted price of the offer. The ON DELETE CASCADE ensures referential integrity by automatically deleting related rows in the offer table if corresponding rows in the referenced tables (foodlist or restaurant) are deleted.

Order table

1. orderid (VARCHAR(15), PRIMARY KEY, NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 15 characters.

Constraint: Primary Key, which means it uniquely identifies each row in the table. It cannot be NULL.

2. orderdate (DATE, NOT NULL):

Data Type: Date data type.

Constraint: Not NULL, indicating that this field must have a value.

3. foodcode (VARCHAR(20)):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 20 characters.

Note: This field is likely a reference to the foodcode in the foodlist table.

4. quantity (INT, NOT NULL):

Data Type: Integer for representing the quantity of the ordered food.

Constraint: Not NULL, indicating that this field must have a value.

5. totalprice (DOUBLE, NOT NULL):

Data Type: Double-precision floating-point number for representing the total price of the order.

Constraint: Not NULL, indicating that this field must have a value.

6. offercode (VARCHAR(20), NOT NULL):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 20 characters.

Constraint: Not NULL, indicating that this field must have a value.

Foreign Key Constraint:

foodcode Foreign Key:

References the foodcode column in the foodlist table.

This table, orders, is designed to store information about customer orders. The orderid is the primary key, uniquely identifying each order. The orderdate represents the date when the order was placed. The foodcode is likely a reference to the foodcode in the foodlist table, indicating the food item ordered. The quantity represents the quantity of the ordered food item. The totalprice represents the total cost of the order. The offercode represents the special offer code applied to the order. The foodcode is a foreign key, establishing a relationship with the foodlist table.

Delivery table

1. orderid (VARCHAR(15)):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 15 characters.

Note: This field is likely a reference to the orderid in the orders table.

2. deliverypersonid (VARCHAR(20)):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 20 characters.

Note: This field is likely a reference to the deliverypersonid in the deliveryperson table.

3. restaurantid (VARCHAR(20)):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 20 characters.

Note: This field is likely a reference to the restaurantid in the restaurant table.

4. customerid (VARCHAR(20)):

Data Type: Variable-length character string (VARCHAR) with a maximum length of 20 characters.

Note: This field is likely a reference to the customerid in the customer table.

Foreign Key Constraints:

orderid Foreign Key:

References the orderid column in the orders table.

deliverypersonid Foreign Key:

References the deliverypersonid column in the deliveryperson table.

restaurantid Foreign Key:

References the restaurantid column in the restaurant table.

customerid Foreign Key:

References the customerid column in the customer table.

This table, delivery, is designed to establish relationships between orders, delivery personnel, restaurants, and customers. The orderid references the orders table, linking a delivery to a specific order. The deliverypersonid references the deliveryperson table, associating a delivery with a particular delivery person. The restaurantid references the restaurant table, connecting a delivery to a specific restaurant.

The customerid references the customer table, linking a delivery to a specific customer. These foreign key constraints ensure referential integrity and maintain consistency in the relationships between the tables.

Normalization

Here the entire database is in 3NF level normalized.

There are no partial dependency in the foodlist, deliverperson, customer, restaurant tables. Since all the non prime attributes in each of the table are fully dependent on the primary key of each table, all these four tables results in 2NF. Then the transitive dependency that could have occurred in delivery table which holds the data for food delivery is solved by mapping orderid from order table, deliverpersonid from the deliverperson table, restaurantid from the restaurant table and customerid from the customer table. This is also applied in the offer table which could have had the transitive dependency, if the restaurantid from the restaurant table and foodcode from the foodlist table was not connected via foreign key constraints. Thus by removing the transitive dependency from these tables the integrity constraints are satisfied and thus the database is now currently in 3NF.