Desi Dining: Final report

(Divi Joshi, Hetal Gala, Kunal Chopra)

Problem statement:

An online food ordering application that aims to provide its customers with a streamlined experience faces significant difficulties due to the absence of a centralized database that is both effective and efficient. There is a greater chance of errors and delays when there is no organized system in place to keep track of information about restaurants, menus, orders, payments, and deliveries. This necessitates a lot of manual labor. The user experience is further hampered by the inability to filter and sort restaurants and menu items based on customer preferences, which results in low retention rates and dissatisfaction. Additionally, in order to maintain the confidentiality and integrity of sensitive customer and payment information, a robust and secure database is required to address the growing concerns regarding data privacy and security. Subsequently, fostering a very much planned and improved data set that can uphold the functionalities of a web-based food-requesting application is significant to fulfill the developing needs of the market and give a problem-free encounter to the clients.

Functionality:

- A centralized and efficient database system that can store and manage information about restaurants, menus, orders, payments, and deliveries to streamline the online food ordering process and reduce errors and delays.
- Advanced filtering and sorting options to allow customers to easily find and select restaurants and menu items based on their preferences, enhancing the user experience and increasing retention rates. Additionally, ensure the security and privacy of sensitive customer and payment information by implementing robust security measures.

Entities:

1. **Restaurant:** contains information about all restaurants and uniquely identified by Rest_ID
2. **Item:** has a list of all items offered by restaurants and uniquely identified by Item_ID
3. **Customer:** contains information about customers who logged in using a username, password and email_id to place an order and is uniquely identified by Cust_ID
4. **Orders:** stores the date, amount and status of the order after it is placed by a customer for a restaurant and is uniquely identified by Order_ID
5. **Payment:** stores information about the date when the payment was made for an order by a customer, the payment status and the mode of payment i.e. credit card, debit card, paypal etc. It is uniquely identified by Payment_ID
6. **Order_Detail:** contains information about the item(s) included in an order, their quantity and additional instruction given by the customer while placing the order. It is uniquely identified by Order_Detail_ID
7. **Delivery:** has information regarding the delivery address, the delivery person's ID, any delivery instructions given by the customer, estimated time and status of the delivery of an order. It is uniquely identified by Delivery_ID
8. **Rider:** includes information of all delivery persons and is uniquely identified by Rider_ID

Entity Relationship:

The above mentioned entities have the following relationships:

1. Restaurant ⇄ Item
2. Restaurant ⇄ Orders
3. Item ⇄ Order_Detail
4. Orders ⇄ Order_Detail
5. Orders ⇄ Delivery
6. Delivery ⇄ Rider
7. Customer ⇄ Payment
8. Customer ⇄ Orders

Cardinalities:

1. Restaurant (Mandatory one) ⇄ Item (Mandatory many)
2. Restaurant (Mandatory one) ⇄ Orders (Mandatory many)
3. Item (Mandatory many) ⇄ Order_Detail (Optional many)
4. Orders (Mandatory one) ⇄ Order_Detail (Mandatory many)
5. Orders (Mandatory one) ⇄ Delivery (Mandatory one)
6. Delivery (Optional many) ⇄ Rider (Mandatory one)
7. Customer (Mandatory one) ⇄ Payment (Mandatory many)
8. Customer (Mandatory one) ⇄ Order_Detail (Mandatory many)

Attributes:

Initial list of attributes under each entity based on its given description:

Restaurant
Rest_ID
Rest_Name
Rest_Contact_No
Rest_Street_Number
Rest_Street_Name
Rest_City
Rest_State
Rest_Zipcode
Operating_Hours
Minimum_Delivery
Delivery_Fees

Payment
Payment_ID
Payment_Status
Payment_Type
Payment_Date

Item
Item_ID
Name
Category
Unit_Price
Type
Description

Delivery
Delivery_ID
Delivery_Status
Delivery_Address
Estimated_Time_of_Delivery
Delivery_Instructions

Orders
Order_ID
Order_Date
Order_Amount
Order_Status

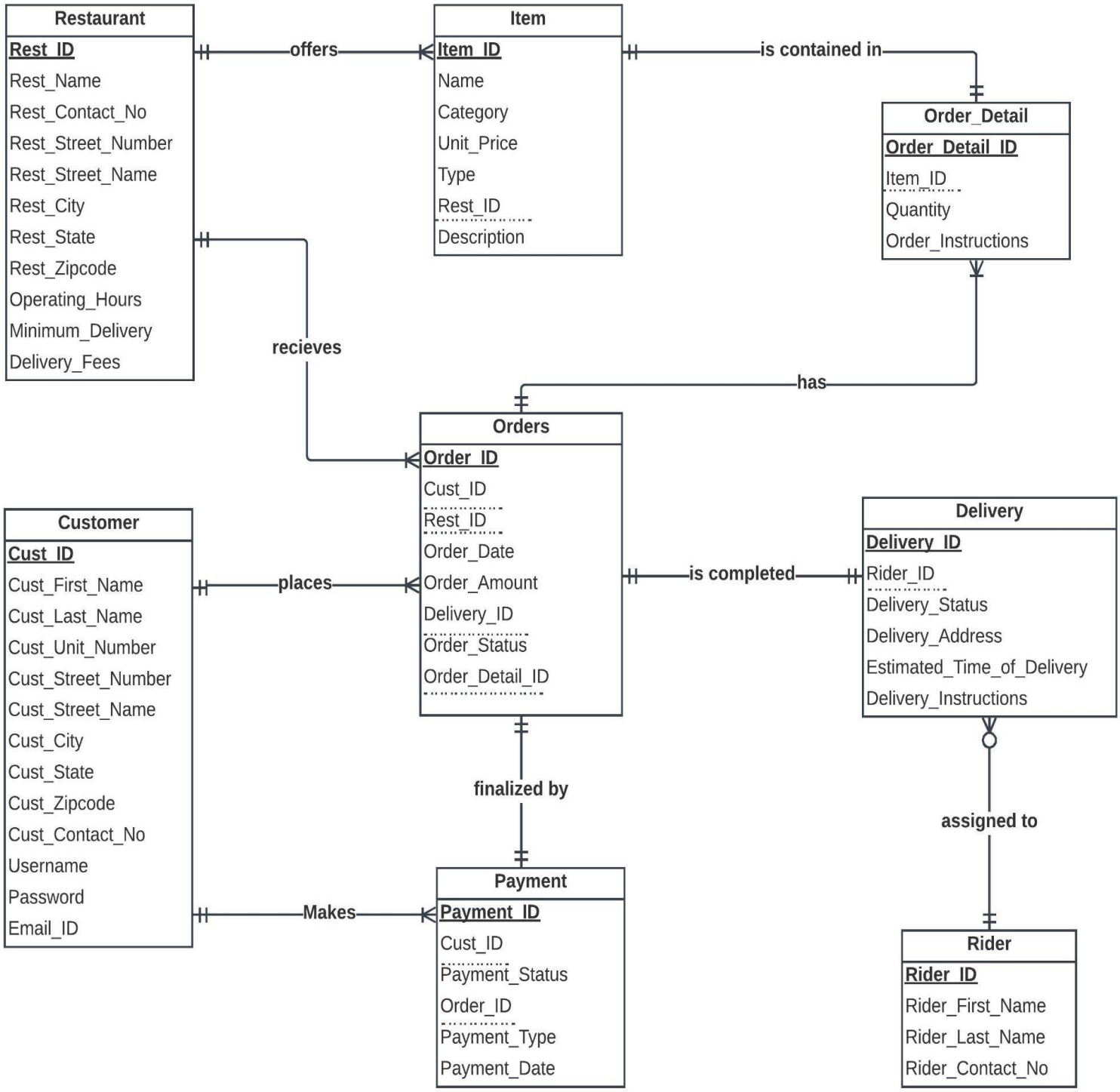
Rider
Rider_ID
Rider_First_Name
Rider_Last_Name
Rider_Contact_No

Order_Detail
Order_Detail_ID
Quantity
Order_Instructions

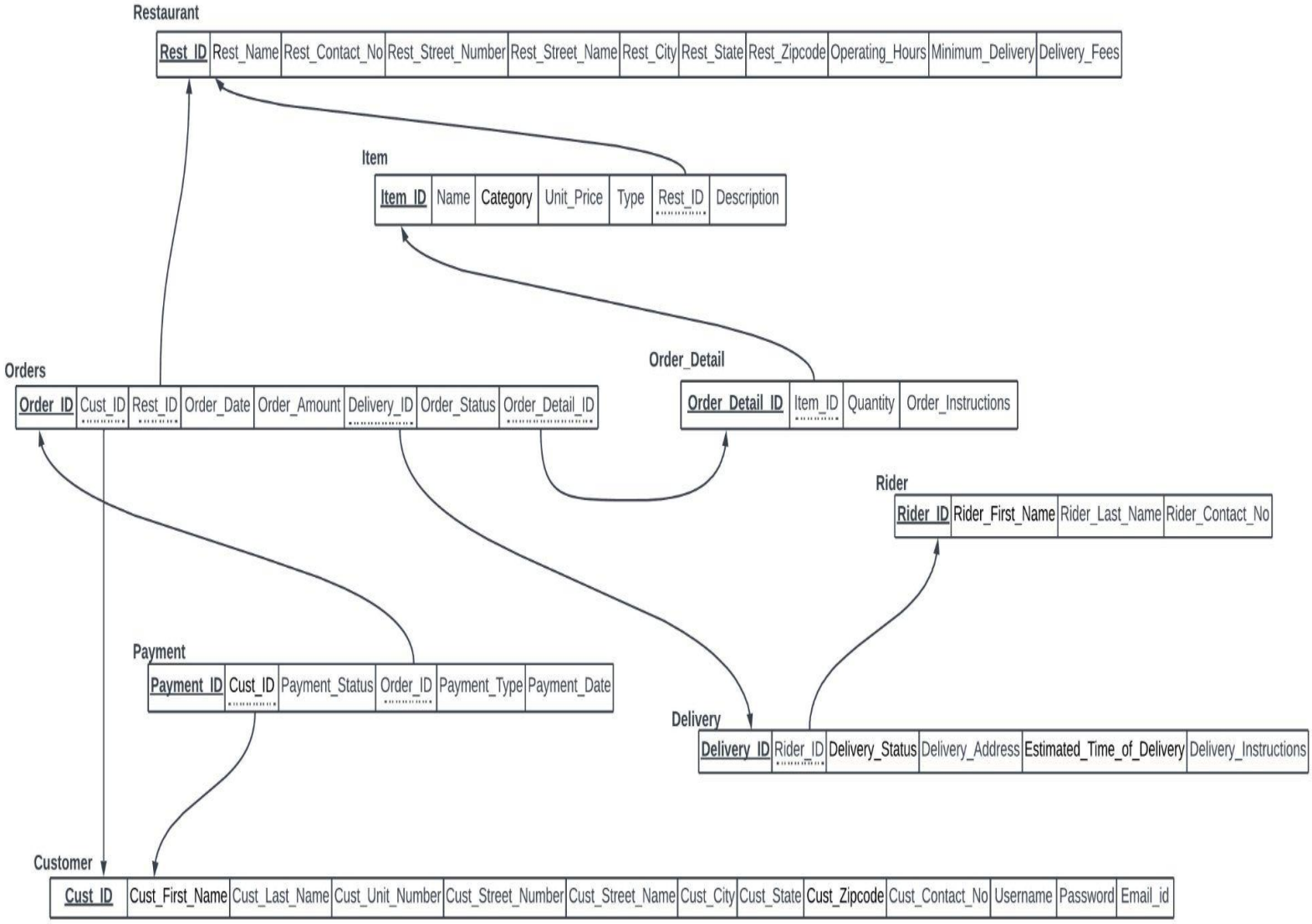
Customer
Cust_ID
Cust_First_Name
Cust_Last_Name
Cust_Unit_Number
Cust_Street_Number
Cust_Street_Name
Cust_City
Cust_State
Cust_Zipcode
Cust_Contact_No
Username
Password
Email_ID

ER Diagram:

The following ER Diagram is based on the attributes, relationships and cardinalities defined before.



ER Diagram to Relational Schema:



Data Normalization:

Since there are no multivalued attributes present, our initial single entity *Order_Details* already exists in 1NF. Partial dependencies are then eliminated from this entity by making three separate entities which were then again decomposed to eight entities by further removing transitive dependencies.

1st Normal Form (1NF):

Order_Details
Restaurant_ID
Restaurant_Name
Restaurant_Address (Street Number, Street Name, City, State, Zip Code)
Restaurant_Contact_No
Operating_Hours
Minimum_Delivery
Delivery_Fees
Customer_Name (First Name, Last Name)
Customer_Address (Unit Number, Street Name, City, State, Zip Code)
Customer_Contact_No
Username
Password
Email_ID
Order_Date
Order_Amount
Order_Status
Order_Quantity
Order_Instruction
Payment_Status
Payment_Type
Payment_Date
Delivery_Status
Delivery_Address
Delivery_Instructions
Estimated_Time_of_Delivery
Rider_Name (First Name, Last Name)
Rider_Contact_No
Item_Name
Item_Type
Item_Category
Item_Unit_Price
Item_Description

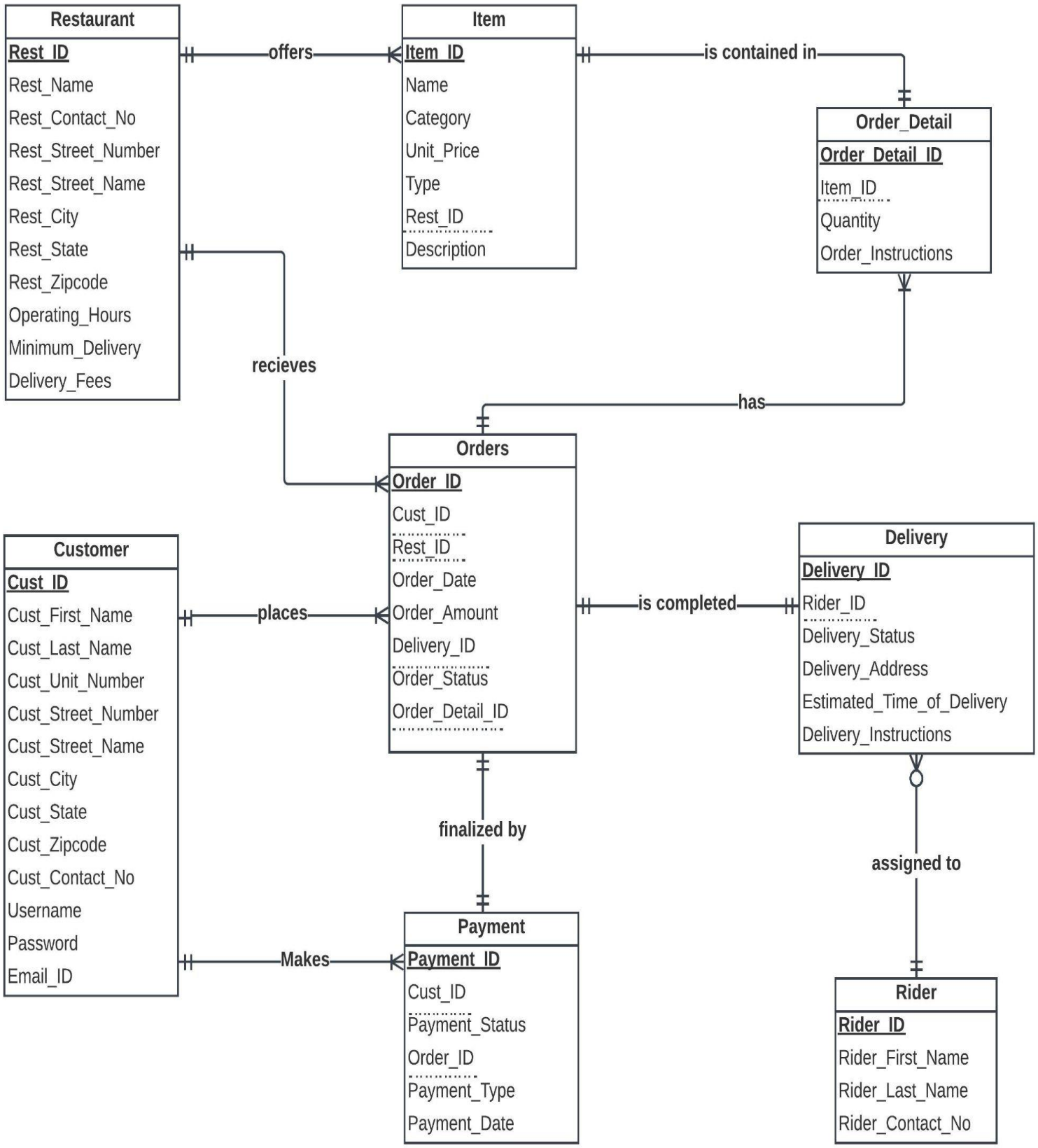
2nd Normal Form (2NF):

Restaurant
Rest_ID
Rest_Name
Rest_Contact_No
Rest_Street_Number
Rest_Street_Name
Rest_City
Rest_State
Rest_Zipcode
Operating_Hours
Minimum_Delivery
Delivery_Fees
Item_Name
Item_Type
Item_Category
Item_Unit_Price
Item_Description

Customer
Cust_ID
Cust_First_Name
Cust_Last_Name
Cust_Unit_Number
Cust_Street_Number
Cust_Street_Name
Cust_City
Cust_State
Cust_Zipcode
Cust_Contact_No
Username
Password
Email_ID
Order_Date
Order_Amount
Order_Status
Order_Quantity
Order_Instruction
Payment_Status
Payment_Type
Payment_Date

Delivery
Delivery_ID
Delivery_Status
Delivery_Address
Estimated_Time_of_Delivery
Delivery_Instructions
Rider_First_Name
Rider_Last_Name
Rider_Contact_No

3rd Normal Form (3NF):



Summary table for each entity:

Restaurant Table
Restaurant (Rest_ID, Rest_Name, Rest_Contact_No, Rest_Street_Number, Rest_Street_Name, Rest_City, Rest_ State, Rest_Zipcode, Operating_Hours, Minimum_Delivery, Delivery_Fees)
Datatype : INTEGER , VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), FLOAT (10), (FLOAT (10) respectively
Additional Details: All fields are required and Rest_ID is the primary key.

Customer Table
Customer (Cust_ID, Cust_First_Name, Cust_Last_Name, Cust_Unit_Number, Cust_Street_Number, Cust_Street_Name, Cust_City, Cust_State, Cust_Zipcode, Cust_Contact_No, Username, Password, Email_ID)
Datatype : INTEGER , VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20) respectively
Additional Details: All fields are required and Cust_ID is the primary key.

Orders Table
Orders (Order_ID, Cust_ID, Rest_ID, Order_Date, Order_Amount, Delivery_ID, Order_Status, Order_Detail_ID)
Datatype : INTEGER , INTEGER , INTEGER , TIMESTAMP, FLOAT (10), INTEGER , VARCHAR (20), INTEGER respectively
Additional Details: All fields are required. Order_ID is the primary key whereas Cust_ID, Rest_ID and Order_Detail_ID are the foreign key.

Item Table
Item (Item_ID, Name, Category, Unit_Price, Type, Rest_ID, Description)
Datatype : INTEGER , VARCHAR (20), VARCHAR (20), FLOAT (10), VARCHAR (20), INTEGER , VARCHAR (20) respectively
Additional Details: All fields are required. Item_ID is the primary key and Rest_ID is the foreign key.

Order Detail Table
Order (Order_Detail_ID, Item_ID, Quantity, Order_Instructions)
Datatype : INTEGER , INTEGER , INTEGER , VARCHAR (255) respectively
Additional Details: All fields are required except Order_Instructions. Order_Detail_ID is the primary key and Item_ID is the foreign key.

Delivery Table
Delivery (Delivery_ID, Rider_ID, Delivery_Status, Delivery_Address, Estimated_Time_of_Delivery, Delivery_Instructions)
Datatype : INTEGER , INTEGER , VARCHAR (20), VARCHAR (255), TIMESTAMP, VARCHAR (255) respectively
Additional Details: All fields are required except Delivery_Instructions. Delivery_ID is the primary key and Rider_ID is the foreign key.

Rider Table
Rider (Rider_ID, Rider_First_Name, Rider_Last_Name, Rider_Contact_No)
Datatype : INTEGER , VARCHAR (50), VARCHAR (50), VARCHAR (20), respectively
Additional Details: All fields are required and Rider_ID is the primary key.

Payment Table
Payment (Payment_ID, Cust_ID, Payment_Status, Order_ID, Payment_Type, Payment_Date)
Datatype : INTEGER , INTEGER , VARCHAR (20), INTEGER , VARCHAR (20), TIMESTAMP respectively
Additional Details: All fields are required. Payment_ID is the primary key whereas Cust_ID and Order_ID are the foreign key.

Creation of Tables:

Following SQL statements are used to finally create all the entities.

Restaurant Table

Restaurant (Rest_ID, Rest_Name, Rest_Contact_No, Rest_Street_Number, Rest_Street_Name, Rest_City, Rest_ State, Rest_Zipcode, Operating_Hours, Minimum_Delivery, Delivery_Fees)
Datatype : INTEGER , VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), FLOAT (10), (FLOAT (10) respectively
Additional Details: All fields are required and Rest_ID is the primary key.

```
CREATE TABLE Restaurant (
 Rest_ID INT(10) PRIMARY KEY,
 Rest_Name VARCHAR(20) NOT NULL,
 Rest_Contact_No VARCHAR(20) NOT NULL,
 Rest_Street_Number VARCHAR(20) NOT NULL,
 Rest_Street_Name VARCHAR(20) NOT NULL,
 Rest_City VARCHAR(20) NOT NULL,
 Rest_State VARCHAR(20) NOT NULL,
 Rest_Zipcode VARCHAR(20) NOT NULL,
 Operating_Hours VARCHAR(20) NOT NULL,
 Minimum_Delivery FLOAT(10) NOT NULL,
 Delivery_Fees FLOAT(10) NOT NULL
);
```

Customer Table

Customer (Cust_ID, Cust_First_Name, Cust_Last_Name, Cust_Unit_Number, Cust_Street_Number, Cust_Street_Name, Cust_City, Cust_State, Cust_Zipcode, Cust_Contact_No, Username, Password, Email_ID)
Datatype : INTEGER , VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20), VARCHAR (20) respectively
Additional Details: All fields are required and Cust_ID is the primary key.

```
CREATE TABLE Customer (
 Cust_ID INT PRIMARY KEY,
 Cust_First_Name VARCHAR(20) NOT NULL,
 Cust_Last_Name VARCHAR(20) NOT NULL,
 Cust_Unit_Number VARCHAR(20) NOT NULL,
 Cust_Street_Number VARCHAR(20) NOT NULL,
 Cust_Street_Name VARCHAR(20) NOT NULL,
 Cust_City VARCHAR(20) NOT NULL,
 Cust_State VARCHAR(20) NOT NULL,
 Cust_Zipcode VARCHAR(20) NOT NULL,
 Cust_Contact_No VARCHAR(20) NOT NULL,
 Username VARCHAR(20) NOT NULL,
 Password VARCHAR(20) NOT NULL,
 Email_ID VARCHAR(20) NOT NULL
);
```

Order Detail Table

Order (Order_Detail_ID, Item_ID, Quantity, Order_Instructions)
Datatype : INTEGER , INTEGER , INTEGER , VARCHAR (255) respectively
Additional Details: All fields are required except Order_Instructions. Order_Detail_ID is the primary key and Item_ID is the foreign key.

```
CREATE TABLE Order_Detail (
```

Item Table

Item (Item_ID, Name, Category, Unit_Price, Type, Rest_ID, Description)
Datatype : INTEGER , VARCHAR (20), VARCHAR (20), FLOAT (10), VARCHAR (20), INTEGER , VARCHAR (20) respectively
Additional Details: All fields are required. Item_ID is the primary key and Rest_ID is the foreign key.

```
CREATE TABLE Item (
```

```
  Order_Detail_ID INTEGER NOT NULL,
  Item_ID INTEGER NOT NULL,
  Quantity INTEGER NOT NULL,
  Order_Instructions VARCHAR(255),
  PRIMARY KEY (Order_Detail_ID),
  FOREIGN KEY (Item_ID) REFERENCES
Item(Item_ID)
);
```

```
  Item_ID INT PRIMARY KEY,
  Name VARCHAR(20) NOT NULL,
  Category VARCHAR(20) NOT NULL,
  Unit_Price FLOAT(10) NOT NULL,
  Type VARCHAR(20) NOT NULL,
  Rest_ID INT NOT NULL,
  Description VARCHAR(20) NOT NULL,
  FOREIGN KEY (Rest_ID) REFERENCES
Restaurant(Rest_ID)
);
```

Orders Table

Orders (Order_ID, Cust_ID, Rest_ID, Order_Date, Order_Amount, Delivery_ID, Order_Status, Order_Detail_ID)
Datatype : INTEGER , INTEGER , INTEGER , TIMESTAMP, FLOAT (10), INTEGER , VARCHAR (20), INTEGER respectively
Additional Details: All fields are required. Order_ID is the primary key whereas Cust_ID, Rest_ID and Order_Detail_ID are the foreign key.

```
CREATE TABLE Orders (
  Order_ID INTEGER NOT NULL,
  Cust_ID INTEGER NOT NULL,
  Rest_ID INTEGER NOT NULL,
  Order_Date TIMESTAMP NOT NULL,
  Order_Amount FLOAT(10) NOT NULL,
  Delivery_ID INTEGER NOT NULL,
  Order_Status VARCHAR(20) NOT NULL,
  Order_Detail_ID INTEGER NOT NULL,
  PRIMARY KEY(Order_ID),
  FOREIGN KEY(Cust_ID) REFERENCES
Customer(Cust_ID),
  FOREIGN KEY(Rest_ID) REFERENCES
Restaurant(Rest_ID),
  FOREIGN KEY(Order_Detail_ID) REFERENCES
Order_Detail(Order_Detail_ID)
);
```

Delivery Table

Delivery (Delivery_ID, Rider_ID, Delivery_Status, Delivery_Address, Estimated_Time_of_Delivery, Delivery_Instructions)
Datatype : INTEGER , INTEGER , VARCHAR (20), VARCHAR (255), TIMESTAMP, VARCHAR (255) respectively
Additional Details: All fields are required except Delivery_Instructions. Delivery_ID is the primary key and Rider_ID is the foreign key.

```
CREATE TABLE Delivery (
 Delivery_ID INT PRIMARY KEY,
 Rider_ID INT NOT NULL,
 Delivery_Status VARCHAR(20) NOT NULL,
 Delivery_Address VARCHAR(255) NOT NULL,
 Estimated_Time_of_Delivery TIMESTAMP NOT NULL,
 Delivery_Instructions VARCHAR(255),
 FOREIGN KEY (Rider_ID) REFERENCES
Rider(Rider_ID)
);
```

Rider Table

Rider (Rider_ID, Rider_First_Name, Rider_Last_Name, Rider_Contact_No)
Datatype : INTEGER , VARCHAR (50), VARCHAR (50), VARCHAR (20), respectively
Additional Details: All fields are required and Rider_ID is the primary key.

```
CREATE TABLE Rider (
 Rider_ID INT PRIMARY KEY,
 Rider_First_Name VARCHAR(50) NOT NULL,
 Rider_Last_Name VARCHAR(50) NOT NULL,
 Rider_Contact_No VARCHAR(20) NOT NULL
);
```

Payment Table

Payment (Payment_ID, Cust_ID, Payment_Status, Order_ID, Payment_Type, Payment_Date)
Datatype : INTEGER , INTEGER , VARCHAR (20), INTEGER , VARCHAR (20), TIMESTAMP respectively
Additional Details: All fields are required. Payment_ID is the primary key whereas Cust_ID and Order_ID are the foreign key.

```
CREATE TABLE Payment (
  Payment_ID INTEGER NOT NULL,
  Cust_ID INTEGER,
  Payment_Status VARCHAR(20) NOT NULL,
  Order_ID INTEGER NOT NULL,
  Payment_Type VARCHAR(20) NOT NULL,
  Payment_Date TIMESTAMP NOT NULL,
  PRIMARY KEY (Payment_ID),
  FOREIGN KEY (Cust_ID) REFERENCES
Customer(Cust_ID),
  FOREIGN KEY (Order_ID) REFERENCES
Orders(Order_ID)
);
```

Insertion of Data into Tables:

```
INSERT INTO Restaurant (Rest_ID, Rest_Name, Rest_Contact_No, Rest_Street_Number,
Rest_Street_Name, Rest_City, Rest_State, Rest_Zipcode, Operating_Hours,Minimum_Delivery,
Delivery_Fees)
VALUES
(1, 'Pizza Palace', '123-456-7890', '123', 'Main St', 'Anytown', 'NY', '12345', '10:00 AM - 10:00
PM', 10.0, 2.99),
(2, 'Burger Barn', '555-123-4567', '456', 'Highway Ave', 'Otherville', 'CA', '98765', '11:00 AM -
9:00 PM', 15.0, 4.99),
(3, 'Taco Town', '999-555-1212', '789', 'Oak Blvd', 'Smallville', 'TX', '54321', '9:00 AM - 11:00 PM',
20.0, 3.50),
(4, 'Sushi Spot', '888-222-3333', '246', 'Elm St', 'Big City', 'IL', '67890', '11:00 AM - 10:00 PM',
15.0, 5.99),
(5, 'Chicken Shack', '777-444-5555', '135', 'Maple Dr', 'Metropolis', 'CA', '98765', '12:00 PM -
8:00 PM', 12.0, 3.99);

INSERT INTO Item (Item_ID, Name, Category, Unit_Price, Type, Rest_ID, Description)
VALUES
(11, 'Margherita Pizza', 'Pizza', 9.99, 'Regular', 1, 'Classic pizza with tomato sauce and
mozzarella cheese'),
```

(12, 'Cheeseburger', 'Burgers', 8.99, 'Regular', 2, 'Juicy beef patty with melted cheese on a sesame bun'),
(13, 'Chicken Tacos', 'Tacos', 7.99, 'Regular', 3, 'Soft shell tacos with grilled chicken and salsa'),
(14, 'California Roll', 'Sushi', 11.99, 'Regular', 4, 'Crab, avocado, and cucumber wrapped in sushi rice and seaweed'),
(15, 'Fried Chicken Sandwich', 'Sandwiches', 6.99, 'Regular', 5, 'Crispy fried chicken breast with lettuce and mayo on a toasted bun');

INSERT INTO Customer (Cust_ID, Cust_First_Name, Cust_Last_Name, Cust_Unit_Number, Cust_Street_Number, Cust_Street_Name, Cust_City, Cust_State, Cust_Zipcode, Cust_Contact_No, Username, Password, Email_ID)
VALUES
(1001, 'John', 'Doe', 'Apt 101', '123', 'Main St', 'Los Angeles', 'CA', '90001', '555-1234', 'johndoe', 'password', 'johndoe@example.com'),
(1002, 'Jane', 'Smith', 'Unit 203', '456', 'Elm St', 'New York', 'NY', '10001', '555-5678', 'janesmith', 'password', 'janesmith@example.com'),
(1003, 'Mike', 'Johnson', 'Suite 301', '789', 'Oak St', 'Chicago', 'IL', '60601', '555-9012', 'mikejohnson', 'password', 'mikejohnson@example.com'),
(1004, 'Emily', 'Davis', 'Apt 102', '246', 'Cedar St', 'San Francisco', 'CA', '94101', '555-3456', 'emilydavis', 'password', 'emilydavis@example.com'),
(1005, 'David', 'Wilson', 'Unit 405', '135', 'Maple St', 'Boston', 'MA', '02101', '555-6789', 'davidwilson', 'password', 'davidwilson@example.com');

INSERT INTO Order_Detail (Order_Detail_ID, Item_ID, Quantity, Order_Instructions)
VALUES
(1, 101, 12, 'No onions please.'),
(2, 102, 11, 'Extra spicy.'),
(3, 103, 13, 'Extra cheese.'),
(4, 104, 12, 'Make it crispy.'),
(5, 105, 11, 'No tomato please.');

INSERT INTO `Orders` (Order_ID, Cust_ID, Rest_ID, Order_Date, Order_Amount, Delivery_ID, Order_Status, Order_Detail_ID)
VALUES
(10021, 1004, 5, '2023-03-15 12:34:56', 25.50, 20004, 'Pending', 1),
(10022, 1002, 1, '2023-03-14 14:15:16', 15.75, 20001, 'Delivered', 2),
(10023, 1001, 3, '2023-03-13 20:30:45', 32.00, 20005, 'Pending', 3),
(10024, 1005, 2, '2023-03-12 08:45:32', 18.00, 20002, 'Cancelled', 4),
(10025, 1004, 3, '2023-03-11 16:20:10', 8.50, 20003, 'Delivered', 5);

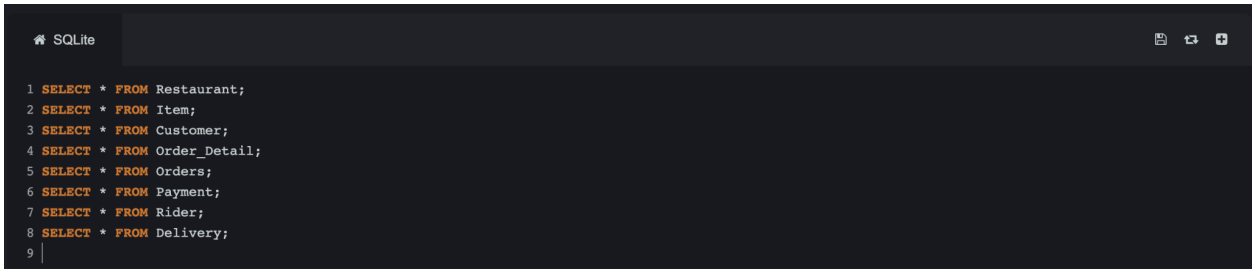
INSERT INTO Payment (Payment_ID, Cust_ID, Payment_Status, Order_ID, Payment_Type, Payment_Date)
VALUES (1, 1001, 'Paid', 10001, 'Credit Card', '2022-03-10 10:30:00'),
(2, 1002, 'Paid', 10002, 'Debit Card', '2022-03-11 11:45:00'),
(3, 1003, 'Pending', 10003, 'Cash', '2022-03-12 12:15:00'),

(4, 1004, 'Paid', 10004, 'PayPal', '2022-03-13 13:20:00'),
(5, 1005, 'Paid', 10005, 'Credit Card', '2022-03-14 14:30:00');

INSERT INTO Rider (Rider_ID, Rider_First_Name, Rider_Last_Name, Rider_Contact_No)
VALUES (101, 'John', 'Doe', '1234567890'),
(120, 'Jane', 'Smith', '2345678901'),
(123, 'Mike', 'Johnson', '3456789012'),
(182, 'Emily', 'Davis', '4567890123'),
(147, 'David', 'Brown', '5678901234');

INSERT INTO Delivery (Delivery_ID, Rider_ID, Delivery_Status, Delivery_Address, Estimated_Time_of_Delivery, Delivery_Instructions)
VALUES (20001, 147, 'In Transit', '123 Main St, Anytown, USA', '2023-03-15 13:00:00', 'Leave at front door'),
(20002, 120, 'Out for delivery', '456 Oak Ave, Anytown, USA', '2023-03-15 13:30:00', 'Call when arrived'),
(20003, 147, 'Delivered', '789 Pine St, Anytown, USA', '2023-03-15 14:00:00', ''),
(20004, 123, 'In Transit', '321 Elm St, Anytown, USA', '2023-03-15 14:30:00', 'Deliver to back door'),
(20005, 182, 'Out for delivery', '654 Maple Ave, Anytown, USA', '2023-03-15 15:00:00', '');

Data loaded in DB:



Rest_ID	Rest_Name	Rest_Contact_No	Rest_Street_Number	Rest_Street...	Rest_City	Rest_State	Rest_Zipcode	Operating_Hours	Mini...	Delive...
Rest_ID INT(10)	Pizza Palace	123-456-7890	123	Main St	Anytown	NY	12345	10:00 AM - 10:00 PM	10	2.99
2	Burger Barn	555-123-4567	456	Highway Ave	Otherville	CA	98765	11:00 AM - 9:00 PM	15	4.99
3	Taco Town	999-555-1212	789	Oak Blvd	Smallville	TX	54321	9:00 AM - 11:00 PM	20	3.5
4	Sushi Spot	888-222-3333	246	Elm St	Big City	IL	67890	11:00 AM - 10:00 PM	15	5.99
5	Chicken Shack	777-444-5555	135	Maple Dr	Metropolis	CA	98765	12:00 PM - 8:00 PM	12	3.99

Item_ID	Name	Category	Unit_Price	Type	Rest_ID	Description
11	Margherita Pizza	Pizza	9.99	Regular	1	Classic pizza with tomato sauce and mozzarella cheese
12	Cheeseburger	Burgers	8.99	Regular	2	Juicy beef patty with melted cheese on a sesame bun
13	Chicken Tacos	Tacos	7.99	Regular	3	Soft shell tacos with grilled chicken and salsa
14	California Roll	Sushi	11.99	Regular	4	Crab, avocado, and cucumber wrapped in sushi rice and seaweed
15	Fried Chicken Sandwich	Sandwiches	6.99	Regular	5	Crispy fried chicken breast with lettuce and mayo on a toasted bun

Cust_ID	Cust_Fir…	Cust_La…	Cust_Un…	Cust_St…	Cust_St…	Cust_City	Cust_St…	Cust_Zi…	Cust_Co…	Username	Password	Email_ID
1001	John	Doe	Apt 101	123	Main St	Los Angeles	CA	90001	555-1234	johndoe	password	johndoe@exam…
1002	Jane	Smith	Unit 203	456	Elm St	New York	NY	10001	555-5678	janesmith	password	janesmith@exa…
1003	Mike	Johnson	Suite 301	789	Oak St	Chicago	IL	60601	555-9012	mikejohns…	password	mikejohnson@e…
1004	Emily	Davis	Apt 102	246	Cedar St	San Fran…	CA	94101	555-3456	emilydavis	password	emilydavis@exa…
1005	David	Wilson	Unit 405	135	Maple St	Boston	MA	02101	555-6789	davidwilson	password	davidwilson@ex…

Order_Detail_ID	Item_ID	Quantity	Order_Instructions
1	101	12	No onions please.
2	102	11	Extra spicy.
3	103	13	Extra cheese.
4	104	12	Make it crispy.
5	105	11	No tomato please.

Order_ID	Cust_ID	Rest_ID	Order_Date	Order_Amount	Delivery_ID	Order_Status	Order_Detail_ID
10021	1004	5	2023-03-15 12:34:56	25.5	20004	Pending	1
10022	1002	1	2023-03-14 14:15:16	15.75	20001	Delivered	2
10023	1001	3	2023-03-13 20:30:45	32	20005	Pending	3
10024	1005	2	2023-03-12 08:45:32	18	20002	Cancelled	4
10025	1004	3	2023-03-11 16:20:10	8.5	20003	Delivered	5

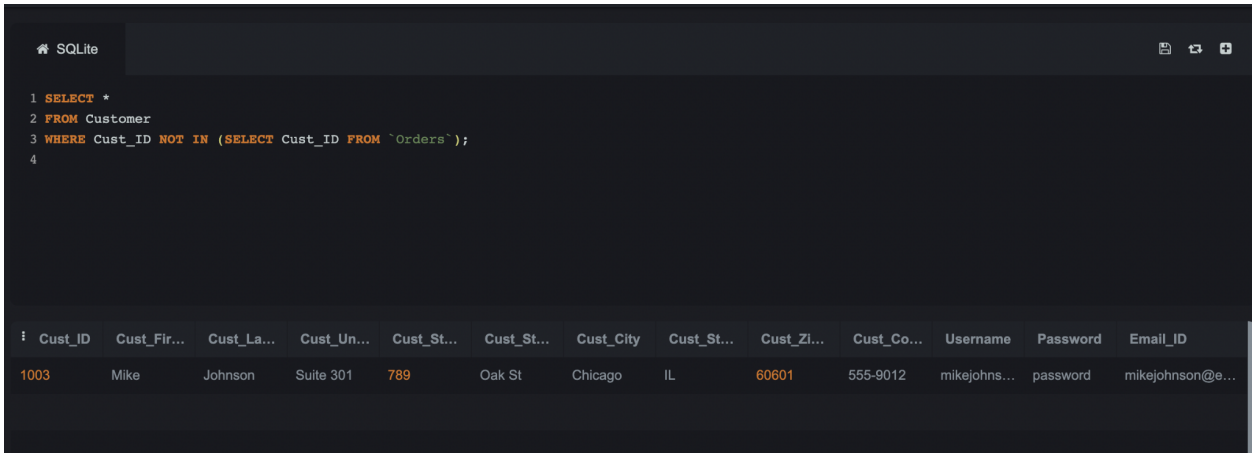
Payment_ID	Cust_ID	Paymen…	Order_ID	Payment_Type	Payment_Date
1	1001	Paid	10001	Credit Card	2022-03-10 10:30:00
2	1002	Paid	10002	Debit Card	2022-03-11 11:45:00
3	1004	Pending	10004	Cash	2022-03-12 12:15:00
4	1004	Paid	10004	PayPal	2022-03-13 13:20:00
5	1005	Paid	10005	Credit Card	2022-03-14 14:30:00

Rider_ID	Rider_First_Name	Rider_Last_Name	Rider_Contact_No
101	John	Doe	1234567890
120	Jane	Smith	2345678901
123	Mike	Johnson	3456789012
182	Emily	Davis	4567890123
147	David	Brown	5678901234

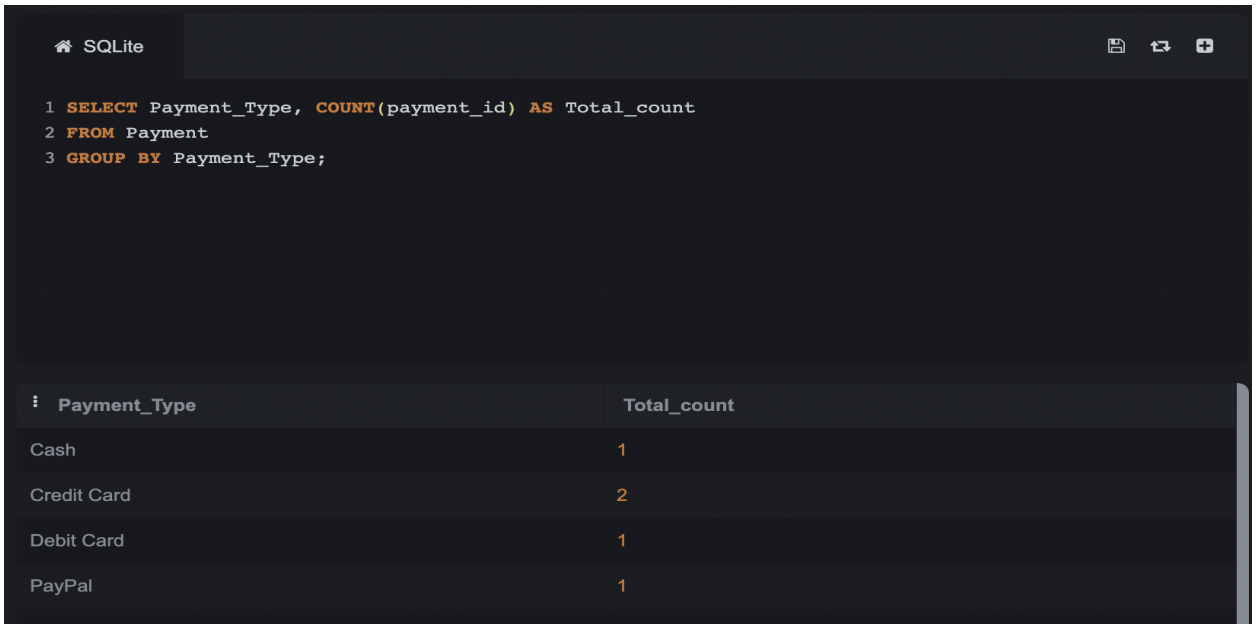
Delivery_ID	Rider_ID	Delivery_Status	Delivery_Address	Estimated_Time_of_Deliv…	Delivery_Instructions
20001	147	In Transit	123 Main St, Anytown, USA	2023-03-15 13:00:00	Leave at front door
20002	120	Out for delivery	456 Oak Ave, Anytown, USA	2023-03-15 13:30:00	Call when arrived
20003	147	Delivered	789 Pine St, Anytown, USA	2023-03-15 14:00:00	
20004	123	In Transit	321 Elm St, Anytown, USA	2023-03-15 14:30:00	Deliver to back door
20005	182	Out for delivery	654 Maple Ave, Anytown, U…	2023-03-15 15:00:00	

Queries:

Q1. To get the details of all the customers who have not placed any orders yet:



Q2. To retrieve the total number of payments corresponding to each payment mode/type



Q3. To get the total number of orders placed by a specific customer whose Cust_ID=’1004’ and share their name , address and contact information.

Learnings:

In general, this project taught us a lot about the significance of database design, advanced user experience features, and data security protocols. In order to construct a database system that is both effective and efficient enough to support the features of a web-based food ordering application, it brought to light the requirement for meticulous planning and careful attention to detail.

We have understood that the absence of a unified information base can prompt mistakes, postponements, and disappointment among clients. The need for advanced filtering and sorting options to improve the user experience was another important aspect of the project. By making it simple for customers to locate and select restaurants and menu items based on their preferences, this feature can significantly increase customer retention rates. We took in the significance of integrating these elements into our data set plan to give a smoothed out and easy to use insight. We also knew how important it was to keep sensitive customer and payment information safe and private. It is essential to implement robust security measures to protect customer data from unauthorized access and potential breaches in light of the growing concerns about data privacy and security.