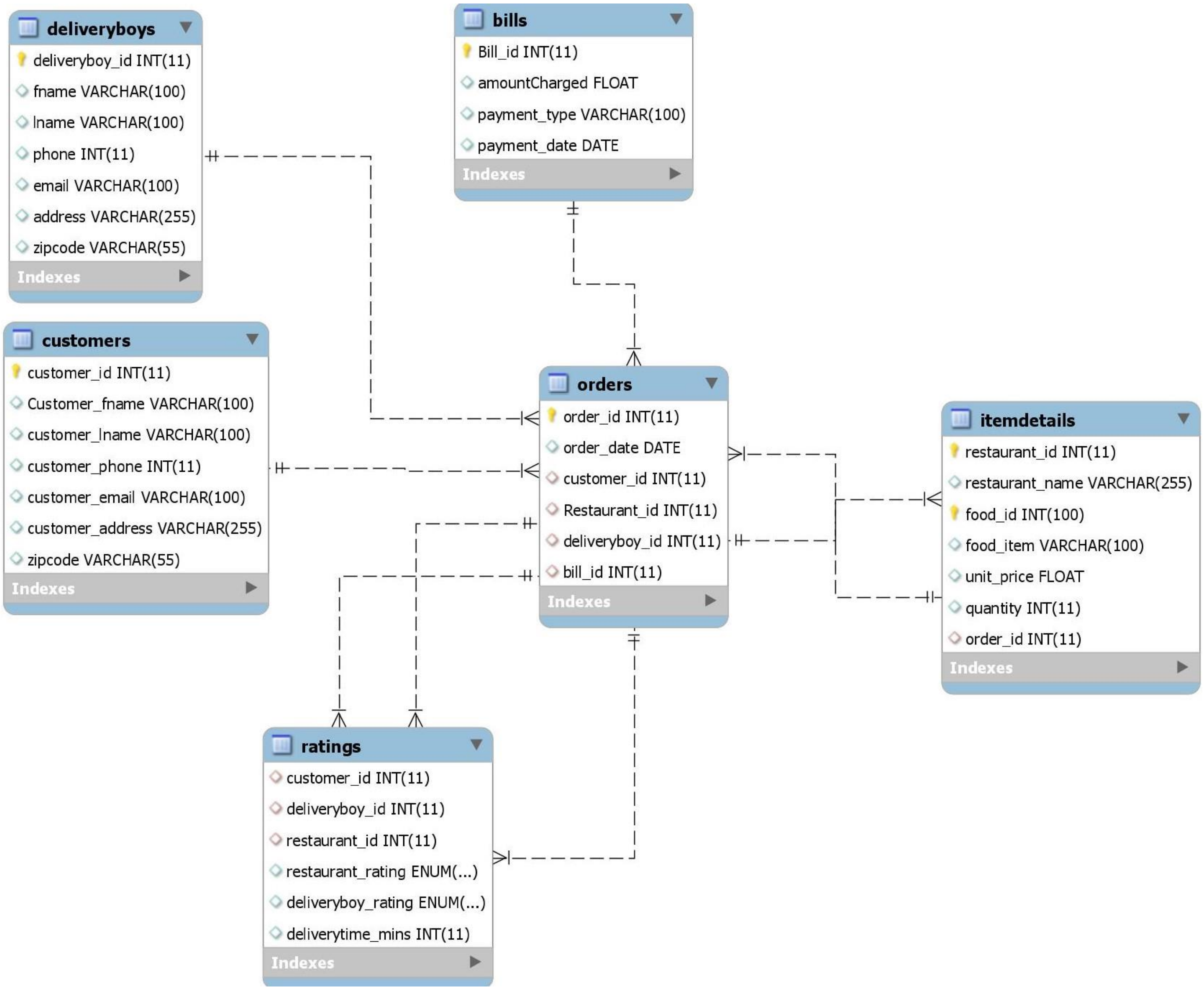
Following ER Diagram has been drawn using MySQL workbench:



STEPS TO CREATE SWIGGY DATABASE AND TABLES:

CREATE DATABASE Swiggy;

USE Swiggy;

```
CREATE TABLE IF NOT EXISTS customers (
        customer_id INT PRIMARY KEY,
        Customer_fname VARCHAR(100), customer_lname VARCHAR(100), customer_phone INT,
        customer_email VARCHAR(100),  customer_address VARCHAR(255), zipcode VARCHAR(55));



CREATE TABLE IF NOT EXISTS deliveryBoys(
        deliveryboy_id INT PRIMARY KEY, fname VARCHAR(100),
        lname VARCHAR(100), phone INT, email VARCHAR(100),
        address VARCHAR(255),zipcode varchar(55));


CREATE TABLE IF NOT EXISTS itemDetails(
        restaurant_id INT , restaurant_name varchar(255),
        food_id int(100), food_item varchar(100),
        unit_price FLOAT, quantity INT, order_id INT,
        PRIMARY KEY (Restaurant_id,food_id),
        FOREIGN KEY (order_id) REFERENCES orders(order_id));


CREATE TABLE IF NOT EXISTS Bills(
        Bill_id INT PRIMARY KEY, amountCharged FLOAT,
        payment_type varchar(100), payment_date DATE);


CREATE TABLE IF NOT EXISTS orders(
        order_id INT PRIMARY KEY, order_date DATE,
        customer_id INT, Restaurant_id INT, deliveryboy_id INT,bill_id INT unique,
        FOREIGN KEY (customer_id) REFERENCES customers(customer_id),
        FOREIGN KEY (deliveryboy_id) REFERENCES deliveryBoys(deliveryboy_id),
        FOREIGN KEY(restaurant_id) REFERENCES itemDetails(restaurant_id), FOREIGN KEY(bill_id)
        REFERENCES bills(bill_id));
```

```
CREATE TABLE IF NOT EXISTS order_items(

        order_item_id INT , food_id int(100),

        food_item varchar(100), unit_price FLOAT,

        PRIMARY KEY (Restaurant_id,food_id));
```

```
CREATE TABLE IF NOT EXISTS ratings(

        customer_id INT,deliveryboy_id INT, restaurant_id INT, restaurant_rating ENUM
        ('1','1.5','2','2.5','3','3.5','4','4.5','5'),

        deliveryboy_rating ENUM('1','1.5','2','2.5','3','3.5','4','4.5','5'), deliverytime_mins INT,

        FOREIGN KEY (customer_id) REFERENCES orders(customer_id),

        FOREIGN KEY (restaurant_id) REFERENCES orders(restaurant_id),

        FOREIGN KEY (deliveryboy_id) REFERENCES orders(deliveryboy_id));
```

1. Write a SQL query to find the number of Swiggy users

```
        SELECT COUNT(DISTINCT customer_id) as numberofSwiggyUsers FROM orders;
```

2. Write a SQL query to find details of Swiggy delivery Boy

```
                -- To find details of all delivery boy;

                        SELECT * FROM deliveryBoys;

                -- To find details of particular delivery boy;


                        SELECT * FROM deliveryBoys WHERE deliveryboy_id = 25;

                (or)

                        SELECT * FROM deliveryBoys WHERE fname like 'ram%';
```

3. Write a SQL query to find the list of Swiggy users who made more than 10 orders in a particular month

```
SELECT customer_id,month(order_date) as Month, count(*) AS total_orders_monthly FROM
orders GROUP BY customer_id,Month HAVING total_orders_monthly > 10  and Month = 2;
```

4. Write a SQL query to find top 10 Swiggy delivery Boy on basis of customer rating and time to deliver the item

```
SELECT A.deliveryboy_id,CONCAT(A.fname,' ',A.lname) AS fullName, B.average_deliveryRating,
B.average_deliveryTime FROM deliveryboys  AS A JOIN

(SELECT deliveryboy_id, AVG(deliveryboy_rating) average_deliveryRating,
AVG(deliverytime_mins) average_deliveryTime FROM ratings GROUP BY deliveryboy_id ORDER
BY average_deliveryRating DESC, average_deliveryTime DESC LIMIT 10) B WHERE
A.deliveryboy_id = B.deliveryboy_id ;
```

5. Write a SQL query to find the list of Swiggy users who order food from the same restaurants more than 3 times in a week

```
SELECT customer_id,week(order_date) as week_order, count(*) AS total_orders_weekly FROM
orders GROUP BY customer_id,week_order HAVING total_orders_weekly > 3;
```