Final - Project Lab Report

Course No:	CSE
Course Title:	Database Design Lab
Project Name:	Online Food Ordering System
Submitted To:	Mr. Nahin Kumar Dey
Submitted By:	Sharia Emon Faisal (2020200010015)

Online Food Ordering System

Project Overview:

Online Food ordering system is a process in which one can order various foods and beverages from some local restaurant and hotels through the use of the internet, just by sitting at home or any place. And the order is delivered to the told location. So the food ordering system these days has one of the fastest growing markets, though being a new idea. Nowadays, people are more regular to dine-in at restaurants for their meals. The online food ordering system provides convenience for the customers that are nothing special but the general busy people of the society.

The online food ordering system provides the menu online and the customers can easily place the order through a website or mobile app.
Also with the food ordering system online, people can easily track their orders, and admin can maintain the customer's database and advance the food delivery system.
This food ordering system allows the user to select the desired food items from a list of available menu items provided by the local hotel or restaurant. The user can place orders for the food items of their liking from the list.

Project Entities described below:

Customer: This represents the set of customers, which are the clients who will be using this application. The customers are for whom the system is being designed.
This entity defines properties such as name, phone, email, address, payment details etc. Here the phone number must have to be unique.

Restaurant: A Restaurant entity defines properties such as the restaurant name, location address, description, phone number, and operating times

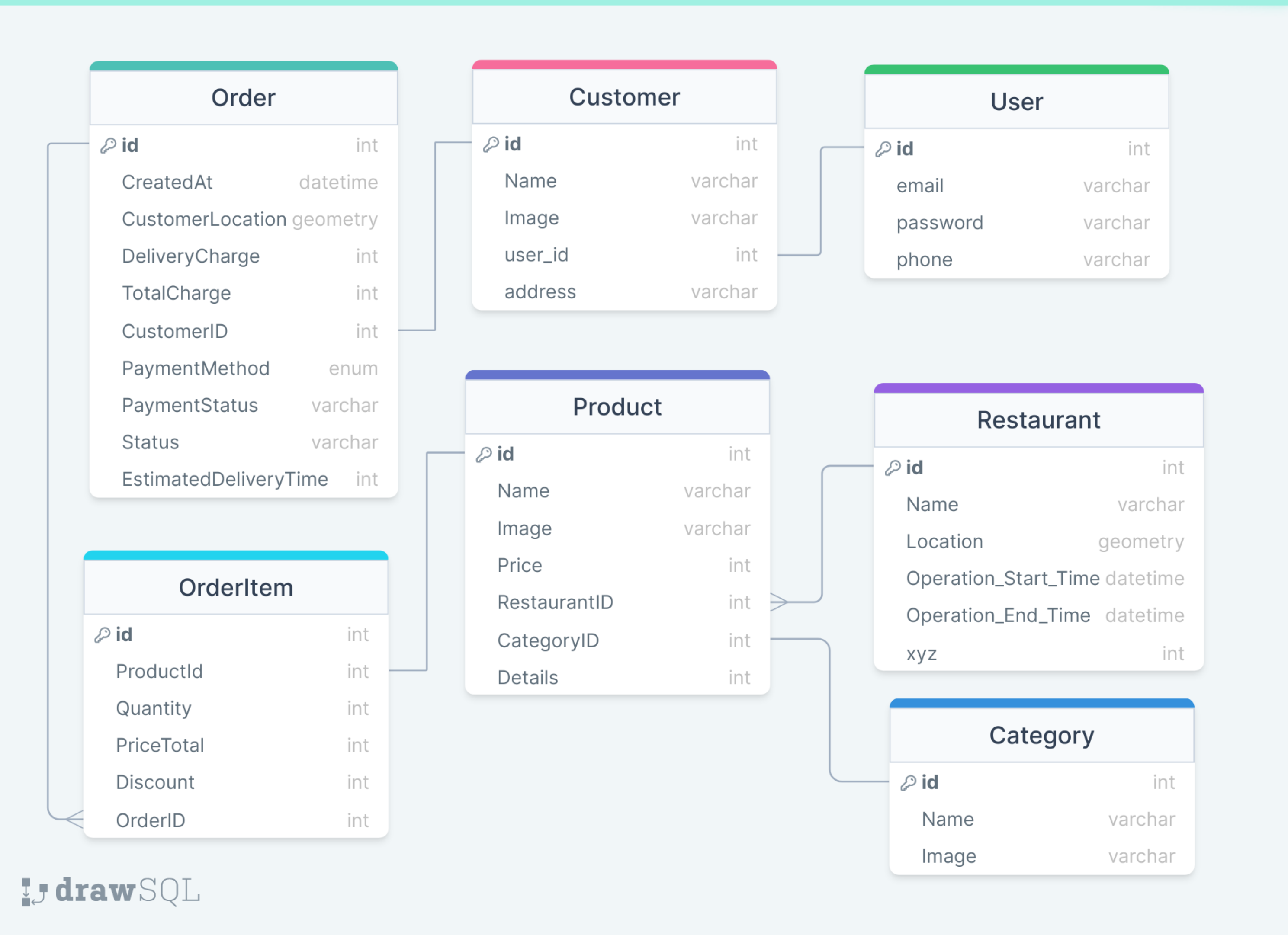
Category: Entity for categorizing products to make it easier for customers to find them on portals and through search.

Product: This entity holds information about a product. Attributes of this entity such as name, image, details, price, etc. Also restaurant and category id to know where it belongs to.

Order: Entity holds information about Customer, Charge, Payment information, Status, Customer present address for where to delivered

Order Item: An order item represents a purchasable entity inside of an order. It contains a reference to the purchasable entity, quantity, unit price and total price.

Schema Diagram:



Here we have 7 entities for this project. To display the menu of restaurants on the website we need three entities: Restaurant, Product and category. Products belong to certain restaurants and categories, to make relationships between entities we have to keep restaurant_id and category_id into product entities. Category will help customers to search and choose their desired menu items easily. There will be an authentication procedure before placing an order for customers. For this, customers need a profile so that they can track their ongoing orders status and order history later on. For customer profile and authentication I took another two entities Customer and User. Customer entity holds customer information and the user entity holds customers authentication information. To make a relationship between two entities customer and user, we kept user id into customer entity. The main purpose of this project was customers will place orders, so we have created another two entities Order and OrderItem to make it possible. Order entity holds information about Customer, Total Charge of order, Payment information, Status, Customer present address etc. And it has a relationship to the customer entity, here customer_id is the foreing key of a customer entity. OrderItem entity holds menu item information like product, quantity, price etc. OrderItem has a Many to One relationship with order entities.

Connect system

```
connect system;
Password: ******
```

Drop existing tables

```
DROP TABLE product;
DROP TABLE restaurants;
DROP TABLE categories;
DROP TABLE order_item;
DROP TABLE orders;
DROP TABLE customer;
DROP TABLE users;
```

Create tables

```
CREATE TABLE users (
        id number(10) NOT NULL,
        email varchar2(255) NOT NULL UNIQUE,
        password varchar2(255) NOT NULL,
        PRIMARY KEY (id)
);
CREATE TABLE customer (
    id number(10) NOT NULL,
    name varchar2(255) NOT NULL,
    user_id number(10) NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (user_id) REFERENCES users(id)
);
CREATE TABLE restaurants (
    id number(10) NOT NULL,
    name varchar2(255) NOT NULL,
    image varchar2(255) NOT NULL,
    user_id number(10) NOT NULL,
    opening_time varchar2(20) NOT NULL,
    closing_time varchar2(20) NOT NULL,
    availability varchar2(20) NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (user_id) REFERENCES users(id)
);
CREATE TABLE categories (
    id number(10) NOT NULL,
    name varchar2(255) NOT NULL,
    image varchar2(255) NOT NULL,
    PRIMARY KEY (id)
);
CREATE TABLE product (
    id number(10) NOT NULL,
    name varchar2(255) NOT NULL,
    image varchar2(255) NOT NULL,
    price number(10) NOT NULL,
    restaurant_id number(10) NOT NULL,
    category_id number(10) NOT NULL,
    details varchar2(255) NOT NULL,
    availability varchar2(20) NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (restaurant_id) REFERENCES restaurants(id),
    FOREIGN KEY (category_id) REFERENCES categories(id)
);

CREATE TABLE orders (
    id number(10) NOT NULL,
    customer_id number(10) NOT NULL,
    status varchar2(20) NOT NULL,
    payment_method varchar2(20) NOT NULL,
    payment_status varchar2(20) NOT NULL,
    total_charge number(10) NOT NULL,
    delivery_charge number(10) NOT NULL,
    address varchar2(255) NOT NULL,
    created_at date NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (customer_id) REFERENCES customer(id)
);

CREATE TABLE order_item (
    id number(10) NOT NULL,
    quantity number(10) NOT NULL,
    product_id number(10) NOT NULL,
    order_id number(10) NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (product_id) REFERENCES product(id),
    FOREIGN KEY (order_id) REFERENCES orders(id)
);
```

Add New Column to order

```
ALTER TABLE orders ADD(
    rider_note varchar(255),
    restaurant_note varchar(255)
);
```

Create user

```
INSERT INTO users (id, email, phone, password) VALUES (1, 'faisal@gmail.com','01234567892',
'123');
INSERT INTO users (id, email, phone, password) VALUES (2, 'rakib@gmail.com','012345455845',
'123');
INSERT INTO users (id, email, phone, password) VALUES (3, 'farhad@gmail.com','01234567878',
'123');
INSERT INTO users (id, email, phone, password) VALUES (4, 'hasan@gmail.com', '01234567889',
'123');
INSERT INTO users (id, email, phone, password) VALUES (5, 'robin@gmail.com', '01234567242',
'123');
INSERT INTO users (id, email, phone, password) VALUES (6, 'tanvir@gmail.com', '01234567972',
'123');

INSERT INTO customer (id, name, user_id) VALUES (1, 'Faisal', 1);
INSERT INTO customer (id, name, user_id) VALUES (2, 'Rakib', 2);
INSERT INTO customer (id, name, user_id) VALUES (3, 'Farhad', 3);
INSERT INTO customer (id, name, user_id) VALUES (4, 'Hasan', 4);
INSERT INTO customer (id, name, user_id) VALUES (5, 'Robin', 5);
INSERT INTO customer (id, name, user_id) VALUES (6, 'Tanvir', 6);
```

Create restaurants

```
INSERT INTO restaurants (id, name, image, user_id, opening_time,     closing_time,
availability) VALUES (1, 'KFC', 'banner.jpg', 3, '10:00', '22:00', 'open');
INSERT INTO restaurants (id, name, image, user_id, opening_time, closing_time, availability)
VALUES (2, 'Burger King', 'banner.jpg', 3, '10:00', '22:00', 'open');
```

```
INSERT INTO restaurants (id, name, image, user_id, opening_time, closing_time, availability)
VALUES (3, 'McDonalds', 'banner.jpg', 3, '10:00', '22:00', 'open');
INSERT INTO restaurants (id, name, image, user_id, opening_time, closing_time, availability)
VALUES (4, 'Pizza Hut', 'banner.jpg', 3, '10:00', '22:00', 'open');
INSERT INTO restaurants (id, name, image, user_id, opening_time, closing_time, availability)
VALUES (5, 'Dominos', 'banner.jpg', 3, '10:00', '22:00', 'open');
INSERT INTO restaurants (id, name, image, user_id, opening_time, closing_time, availability)
VALUES (6, 'Subway', 'banner.jpg', 3, '10:00', '22:00', 'open');
```

Create categories

```
INSERT INTO categories (id, name, image) VALUES (1, 'Burger', 'burger.jpg');
INSERT INTO categories (id, name, image) VALUES (2, 'Pizza', 'pizza.jpg');
INSERT INTO categories (id, name, image) VALUES (3, 'Drinks', 'drinks.jpg');
INSERT INTO categories (id, name, image) VALUES (4, 'Dessert', 'dessert.jpg');
INSERT INTO categories (id, name, image) VALUES (5, 'Chicken', 'chicken.jpg');
INSERT INTO categories (id, name, image) VALUES (6, 'Sandwich', 'sandwich.jpg');
```

Create some products

```
INSERT INTO product (id, name, image, price, restaurant_id, category_id, details, availability)
VALUES (1, 'Chicken Burger', 'burger.jpg', 100, 1, 1, 'Chicken Burger', 'available');
INSERT INTO product (id, name, image, price, restaurant_id, category_id, details, availability)
VALUES (2, 'Beef Burger', 'burger.jpg', 120, 1, 1, 'Beef Burger', 'available');
INSERT INTO product (id, name, image, price, restaurant_id, category_id, details, availability)
VALUES (3, 'Chicken Pizza', 'pizza.jpg', 150, 1, 2, 'Chicken Pizza', 'available');

INSERT INTO product (id, name, image, price, restaurant_id, category_id, details, availability)
VALUES (4, 'Beef Pizza', 'pizza.jpg', 200, 1, 2, 'Beef Pizza', 'available');
INSERT INTO product (id, name, image, price, restaurant_id, category_id, details, availability)
VALUES (5, 'Coke', 'drinks.jpg', 50, 1, 3, 'Coke', 'available');
INSERT INTO product (id, name, image, price, restaurant_id, category_id, details, availability)
VALUES (6, 'Fanta', 'drinks.jpg', 50, 1, 3, 'Fanta', 'available');
INSERT INTO product (id, name, image, price, restaurant_id, category_id, details, availability)
VALUES (7, 'Ice Cream', 'dessert.jpg', 50, 1, 4, 'Ice Cream', 'available');
INSERT INTO product (id, name, image, price, restaurant_id, category_id, details, availability)
VALUES (8, 'Chocolate Cake', 'dessert.jpg', 50, 1, 4, 'Chocolate Cake', 'available');
INSERT INTO product (id, name, image, price, restaurant_id, category_id, details, availability)
VALUES (9, 'Chicken Wings', 'chicken.jpg', 50, 1, 5, 'Chicken Wings', 'available');
INSERT INTO product (id, name, image, price, restaurant_id, category_id, details, availability)
VALUES (10, 'Chicken Nuggets', 'chicken.jpg', 50, 1, 5, 'Chicken Nuggets', 'available');
INSERT INTO product (id, name, image, price, restaurant_id, category_id, details, availability)
VALUES (11, 'Chicken Burger', 'burger.jpg', 100, 2, 1, 'Chicken Burger', 'available');
INSERT INTO product (id, name, image, price, restaurant_id, category_id, details, availability)
VALUES (12, 'Beef Burger', 'burger.jpg', 120, 2, 1, 'Beef Burger', 'available');
```

Place some order

```
INSERT INTO orders (id, customer_id, status, payment_method, payment_status, total_charge,
delivery_charge, address, created_at) VALUES (1, 1, 'pending', 'cash', 'paid', 100, 10,
'Dhaka', TO_DATE('2022-10-04', 'YYYY-MM-DD'));
INSERT INTO order_item (id, quantity, product_id, order_id) VALUES (1, 1, 1, 1);
INSERT INTO order_item (id, quantity, product_id, order_id) VALUES (2, 1, 2, 1);


INSERT INTO orders (id, customer_id, status, payment_method, payment_status, total_charge,
delivery_charge, address, created_at) VALUES (2, 1, 'pending', 'cash', 'paid', 100, 10,
'Dhaka', TO_DATE('2022-10-04', 'YYYY-MM-DD'));
INSERT INTO order_item (id, quantity, product_id, order_id) VALUES (3, 1, 3, 2);
INSERT INTO order_item (id, quantity, product_id, order_id) VALUES (4, 1, 4, 2);

INSERT INTO orders (id, customer_id, status, payment_method, payment_status, total_charge,
delivery_charge, address, created_at) VALUES (3, 1, 'pending', 'cash', 'paid', 100, 10,
'Dhaka', TO_DATE('2022-10-04', 'YYYY-MM-DD'));
INSERT INTO order_item (id, quantity, product_id, order_id) VALUES (5, 1, 5, 3);
INSERT INTO order_item (id, quantity, product_id, order_id) VALUES (6, 1, 6, 3);

INSERT INTO orders (id, customer_id, status, payment_method, payment_status, total_charge,
delivery_charge, address, created_at) VALUES (4, 1, 'pending', 'cash', 'paid', 100, 10,
'Dhaka', TO_DATE('2022-10-04', 'YYYY-MM-DD'));
INSERT INTO order_item (id, quantity, product_id, order_id) VALUES (7, 1, 7, 4);
INSERT INTO order_item (id, quantity, product_id, order_id) VALUES (8, 1, 8, 4);

INSERT INTO orders (id, customer_id, status, payment_method, payment_status, total_charge,
delivery_charge, address, created_at) VALUES (5, 1, 'pending', 'cash', 'paid', 100, 10,
'Dhaka', TO_DATE('2022-10-04', 'YYYY-MM-DD'));
INSERT INTO order_item (id, quantity, product_id, order_id) VALUES (9, 1, 9, 5);
INSERT INTO order_item (id, quantity, product_id, order_id) VALUES (10, 1, 10, 5);
```

Update order status to delivered for order id 1

```
UPDATE orders SET status = 'delivered' WHERE id = 1;
```

Get all orders for customer id 1

```
select * from orders where customer_id = 1;
```

Get order which order id is equal to 1

```
select * from orders where id=1;
```

Get all delivered order

```
SELECT * from orders where status='delivered';
```

Get orders with order item quantity and order status

```
SELECT o.status, oi.quantity, p.name FROM orders o JOIN order_item oi ON o.id = oi.order_id
INNER JOIN product p ON oi.product_id = p.id LIMIT 10;
```

Get average order value

```
SELECT avg(total_charge) FROM orders;
```

Get total item value for specific order by id

```
SELECT sum(price * quantity) FROM order_item oi INNER JOIN product p ON oi.product_id = p.id
WHERE oi.order_id = 1;
```

Get sold item quantity of specific product by id

```
SELECT sum(quantity) FROM order_item WHERE product_id = 9;
```

Order count for specific product by id

```
SELECT count(quantity) FROM order_item WHERE product_id = 9;
```

Fetch all customer, restaurant and categories

```
SELECT * FROM customer;
SELECT * FROM restaurant;
SELECT * FROM categories;
```

Fetch products including category name

```
SELECT p.name, c.name FROM product p INNER JOIN categories c ON p.category_id = c.id;
```

Fetch products including restaurant name

```
SELECT product.name, restaurants.name FROM product JOIN restaurants ON product.restaurant_id =
restaurants.id;
```

Create products table view

```
CREATE VIEW products AS SELECT p.name, c.name FROM product p INNER JOIN categories c ON
p.category_id = c.id;
```

Create restaurants table view

```
CREATE VIEW restaurants AS SELECT product.name, restaurants.name FROM product JOIN restaurants
ON product.restaurant_id = restaurants.id;
```