FOOD ORDERING SYSTEM

"Final Project for SQL Module by Parth Patil"

Welcome to the Food Ordering System Project in MySQL, a digital platform designed to simplify the process of ordering food from restaurants. The MySQL database system, this project aims to transform the way food is ordered, managed, and delivered . Below, I'll provide a high-level overview of the key tables and their relationships in a basic food ordering system :

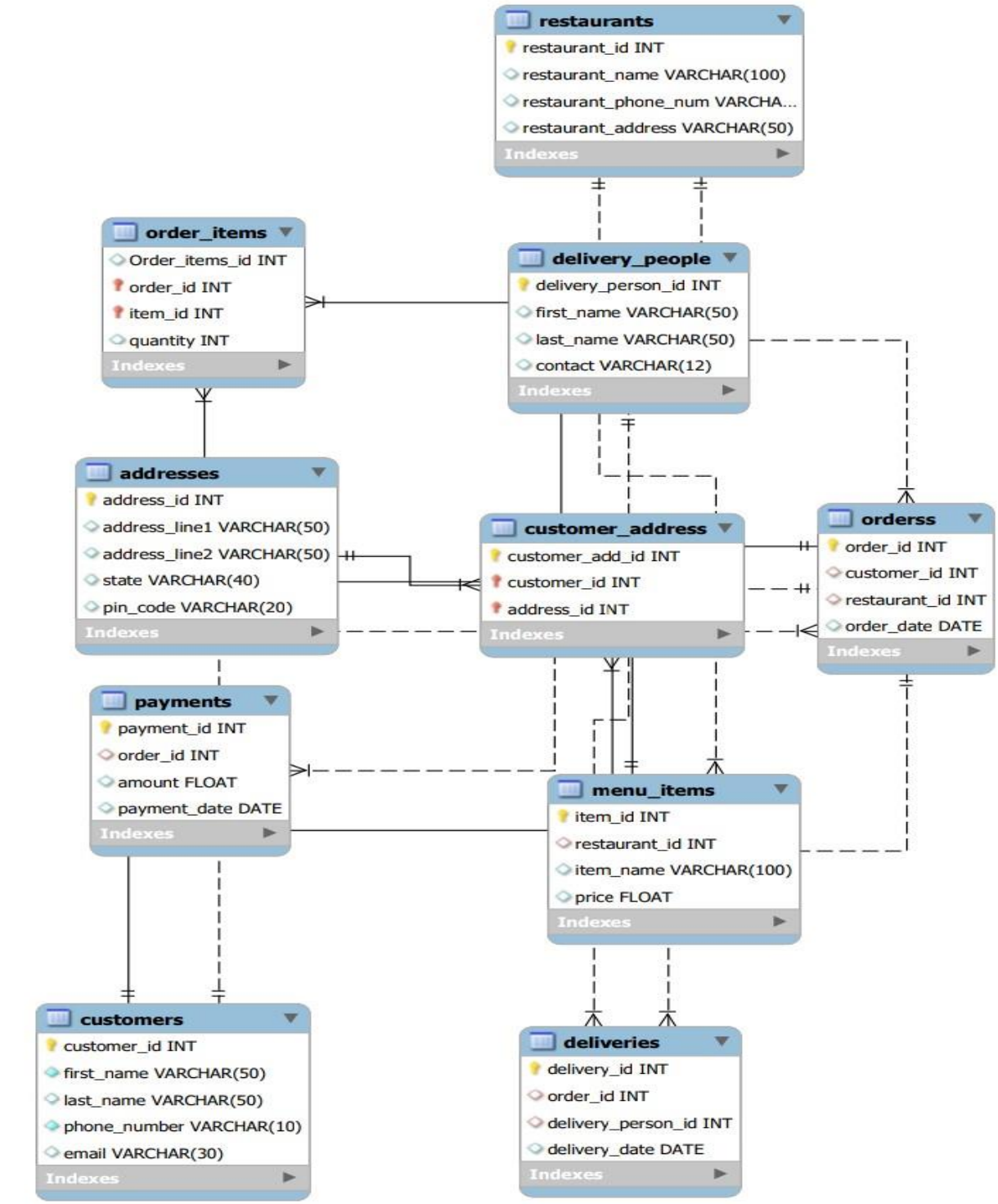
1. CUSTOMER TABLE.
2. RESTAURANTS TABLE.
3. ADDRESS TABLE.
4. CUSTOMER ADDRESS TABLE
5. MENU TABLE.
6. ORDER TABLE.
7. ORDER ITEM TABLE.
8. PAYMENT TABLE.
9. DELIVERY PEOPLE.
10. DELIVERIES.

These tables are interconnected through relationships, typically established using foreign keys. Here are some common relationships:

- A customer places an order, and the order is associated with a restaurant.
- An order can consist of multiple menu items, recorded in the order items table.
- Each order has a corresponding payment.
- A delivery person can handle multiple deliveries.

Remember to design your database based on your specific requirements and business logic, and always follow best practices for database normalization to ensure data integrity and efficiency.

2. ER-DIAGRAM FOR FOOD ORDERING SYSTEM:



3. TABLES DESCRIPTIONS:

1. CUSTOMER

Field	Type	Null	Key	Default	Extra
customer_id	int	NO	PRI	NULL	
first_name	varchar(50)	NO		NULL	
last_name	varchar(50)	YES		NULL	
phone_number	varchar(10)	NO		NULL	
email	varchar(30)	YES		NULL	

2. ADDRESS

Field	Type	Null	Key	Default	Extra
address_id	int	NO	PRI	NULL	
address_line1	varchar(50)	YES		NULL	
address_line2	varchar(50)	YES		NULL	
state	varchar(40)	YES		NULL	
pin_code	varchar(20)	YES		NULL	

3. CUSTOMER ADDRESS

Field	Type	Null	Key	Default	Extra
customer_add_id	int	NO	PRI	NULL	auto_increment
customer_id	int	NO	PRI	NULL	
address_id	int	NO	PRI	NULL	

4. REATAURANT

Field	Type	Null	Key	Default	Extra
restaurant_id	int	NO	PRI	NULL	
restaurant_name	varchar(100)	YES		NULL	
restaurant_phone_num	varchar(12)	YES		NULL	
restaurant_address	varchar(50)	YES		NULL	

5. ITEM

Field	Type	Null	Key	Default	Extra
item_id	int	NO	PRI	NULL	
restaurant_id	int	YES	MUL	NULL	
item_name	varchar(100)	YES		NULL	
price	float	YES		NULL	

6. ORDER

Field	Type	Null	Key	Default	Extra
order_id	int	NO	PRI	NULL	
customer_id	int	YES	MUL	NULL	
restaurant_id	int	YES	MUL	NULL	
order_date	date	YES		NULL	

7. ORDER ITEMS

Field	Type	Null	Key	Default	Extra
Order_items_id	int	YES		NULL	
order_id	int	NO	PRI	NULL	
item_id	int	NO	PRI	NULL	
quantity	int	YES		NULL	

8. PAYMENT

Field	Type	Null	Key	Default	Extra
payment_id	int	NO	PRI	NULL	
order_id	int	YES	MUL	NULL	
amount	float	YES		NULL	
payment_date	date	YES		NULL	

9. DELIVERY PERSON

Field	Type	Null	Key	Default	Extra
delivery_person_id	int	NO	PRI	NULL	
first_name	varchar(50)	YES		NULL	
last_name	varchar(50)	YES		NULL	
contact	varchar(12)	YES		NULL	

10. DELIVERIES

Field	Type	Null	Key	Default	Extra
delivery_id	int	NO	PRI	NULL	
order_id	int	YES	MUL	NULL	
delivery_person_id	int	YES	MUL	NULL	
delivery_date	date	YES		NULL	

4. COMMANDS:

- Create Database:

  create database foodorder;

- Select Database:

  use foodorder;

- Create table named customers

```
create table customers(
-> customer_id int primary key,
-> first_name varchar(50) not null,
-> last_name varchar(50),
-> phone_number varchar(10) not null,
-> email varchar(30),
-> );
```

- Create table named addresses

```
create table addresses(
-> address_id int primary key,
-> address_line1 varchar(50),
-> address_line2 varchar(50),
-> state varchar(40),
-> pin_code varchar(20)
-> );
```

- Create table named customer_address

```
create table customer_address(
-> customer_add_id int auto_increment,
-> customer_id int,
-> address_id int,
-> primary key (customer_add_id,customer_id,address_id),
->  foreign key (customer_id) references customers(customer_id),
-> foreign key (address_id) references addresses(address_id)
-> );
```

- Create table named restaurants

```
create table restaurants(
-> restaurant_id int primary key,
-> restaurant_name varchar(100),
-> restaurant_phone_num varchar(12),
-> restaurant_address varchar(50)
-> );
```

- Create table named menu_items

```
create table menu_items(
-> item_id int primary key,
-> restaurant_id int,
-> item_name varchar(100),
-> price float,
->  foreign key (restaurant_id) references restaurants(restaurant_id)
```

```
-> );
```

- Create table named orderss

```
create table orderss (
->    order_id INT primary key,
->    customer_id INT,
->    restaurant_id INT,
->    order_date DATE,
->   foreign key (customer_id) REFERENCES customers(customer_id),
->   foreign key (restaurant_id) REFERENCES restaurants(restaurant_id)
->  );
```

- Create table named order items

```
create table order_items (
->    Order_items_id INT,
->    order_id INT,
->    item_id INT,
->    quantity INT,
->    Primary key (order_id, item_id),
->    Foreign key(order_id) REFERENCES orderss(order_id),
->    Foreign key(item_id) REFERENCES menu_items(item_id)
-> );
```

- Create table named Payments

```
create table payments (
```

```
-> payment_id INT Primary key,
-> order_id INT,
-> amount FLOAT,
-> payment_date DATE,
-> Foreign key(order_id) REFERENCES orderss(order_id)
-> );
```

- Create table named Delivery people

```
create table delivery_people (
-> delivery_person_id INT auto_increment Primary key,
-> first_name VARCHAR(50),
-> last_name VARCHAR(50)
-> );
```

- Create table named Deliveries

```
create table deliveries (
-> delivery_id INT Primary key,
-> order_id INT,
-> delivery_person_id INT,
-> delivery_date DATE,
-> Foreign key(order_id) REFERENCES orderss(order_id),
-> Foreign key(delivery_person_id) REFERENCES
delivery_people(delivery_person_id)
-> );
```

5.INSERT DATA

- **Populate “Customers” Table:**

  insert into customers values (1,'Rakhi','Jain',9876543210,'rakhij@gmail.com'),

  (2,'Arjun','Singh',8876943910,'arjun89@gmail.com'),

  (3,'Karan','Das',8679543210,'daskaran@gmail.com'),

  (4,'shan','Kumar',8124537890,'kumars12@gmail.com'),

  (5,'Rani','Sharma',8325679013,'sharmarani12@gmail.com'),

  (6,'Rashi','Gore',9376543223,'rashi@gmail.com'),

  (7,'Anam','Ansari',9886943910,'anam099@gmail.com'),

  (8,'Vidya','Juvekar','9023451789','juvekarvi@gmail.com'),

  (9,'Aditya','Shelar',9735367728),

  (10,'Shruti','Singh',998367728),

  (13,'Bhumi','Mane',898367728);

- **Populate “Addresses” Table:**

  insert into addresses values(1,'123 Main st','Cityvilla Mumbai','Maharashtra',600078),

  (2,'456 Elm Rd','Townvilla Munger','Bihar',200089),

  (3,'23 marg','Lal kot','Delhi',300078),

  (4,'46 ram Rd','Mtown','Punjab',700089),

  (5,'2 swami samarth Rd','Akklkot marg','Akklkot',300099),

(6,'lbs marg','shreyas mumbai','maharashtra',400078),

(7,'326 shita Rd','Ptown','Pune',200089),

(8,'rr marg','tilak nagar mumbai','maharashtra',400079);

- **Populate “Customers Address” Table:**

insert into customer_address
values(101,1,1),(102,2,2),(103,3,3),(104,4,4),(105,5,5),(106,6,6),(107,7,7),(109,8,8);

- **Populate “Restaurants” Table:**

insert into restaurants values(1,'Chaat Masala',9808765432,'E-101 Laxmibaugh Ghatkopar Mumbai'),

(2,'Tandoori Flames',9808765432,'Ground-Floor-01 Samarth marg Bhandup Mumbai'),

(3,'Biryani Bliss',976542317,'A bobcat lane mumbai'),

(4,'Chai and Chaat',9398765432,'raghav marg Kurla Mumbai'),

(5,'Paneer Palace ',9765466017,'7th street thane mumbai'),

(6,'Samosa Spot',976165432,'Shri-baugh Delhi'),

(7,'Naan Nation',976165432,'Neelyok blg ground floor Bihar');

- **Populate “Menu Items” Table:**

insert into menu_items values(1,1,'Vegetable Pulao',200.00),

(2,1,'Aloo Pulao',190.00),

(3,2,'Chicken Tikka',250.00),

(4,3,'Biryani',300.00),

(5,4,'Chai and Vada Pav',50.00),

(6,5,'Panner Tikka',210.00),

(7,6,'Samosa Pav and Cheese Samosa Pav',150.00),

(8,7,'Naan Bread',10.00);

- **Populate “Menu Items” Table:**

  insert into orderss values(1, 1, 1, '2023-08-25'),

  (2, 2, 2, '2023-08-25'),

  (3, 3, 3, '2023-08-25'),

  (4, 4, 4, '2023-08-25'),

  (5, 5, 5, '2023-08-25'),

  (6, 6, 6, '2023-08-25'),

  (7, 7, 7, '2023-08-25'),

  (8, 8, 7, '2023-08-25'),

  (9,2,6,'2023-08-26'),

  (10,4,5,' 2023-08-25');,

  (11,6,5,' 2023-08-25');

- **Populate “Order Items” Table:**

  insert into order_items values(1,1,1,2),

  (2,2,2,3),

  (3,3,3,1),

  (4,4,4,2),

(5,5,1,2),

(6,6,6,1),

(7,7,7,5),

(8,8,8,1),

(9,9,4,5),

(10,10,5,2),

(11,11,5,3);

- **Populate “Payment” Table:**

  insert into payments values(1,1,200,'2023-08-25'),

  (2,1,190,'2023-08-25'),

  (3,2,250,'2023-08-25'),

  (4,3,300,'2023-08-25'),

  (5,4,50,'2023-08-25'),

  (6,5,210,'2023-08-25'),

  (7,6,150,'2023-08-25'),

  (8,7,10,'2023-08-25'),

  (9,9,100,2023-09-26),

  (10,10,200,'2023-09-26'),

  (11,11,200,'2023-09-26'),

  (12,8,10,’2023-09-26');

- **Populate “Delivery People” Table:**

```
insert into delivery_people values(1,'Rohan','Patel'),
(2,'Nimesh','Jha'),
(3,'Rohan','Patel'),
(4,'Amit','Shah'),
(5,'Aryan','Verma'),
(6,'Ritik','Kumar'),
(7,'Alex','Smith');
```

- **Populate “Deliveries” Table:**

```
insert into deliveries values(1,1,1,'2023-08-25'),
(2,2,2,'2023-08-25'),
(3,3,3,'2023-08-25'),
(4,4,4,'2023-08-25'),
(5,5,5,'2023-08-25'),
(6,6,6,'2023-08-25'),
(7,7,7,'2023-08-25'),
(8,8,7,'2023-08-25');
```

6.JOINS

1. **SQL query that retrieves customer information along with the restaurant names they have ordered**

select customers.customer_id,customers.first_name,customers.last_name,restaurants.restaurant_name from customers inner join orderss on customers.customer_id=orderss.customer_id inner join restaurants on orderss.restaurant_id = restaurants.restaurant_id;

RESULT:

customer_id	first_name	last_name	restaurant_name
1	Rakhi	Jain	Chaat Masala
2	Arjun	Singh	Tandoori Flames
3	Karan	Das	Biryani Bliss
4	shan	Kumar	Chai and Chaat
5	Rani	Sharma	Paneer Palace
4	shan	Kumar	Paneer Palace
6	Rashi	Gore	Paneer Palace
6	Rashi	Gore	Samosa Spot
2	Arjun	Singh	Samosa Spot
7	Anam	Ansari	Naan Nation
8	Vidya	Juvekar	Naan Nation

2. **SQL query that combines information about customers, restaurants, menu items, payments for each order:**

RESULT:

Select customers.customer_id,customers.first_name,customers.phone_number,restaurants.restaurant_name,menu_items.item_name, orderss.order_id,payments.Payment_id from customers left join orderss on customers.customer_id = orderss.customer_id left join restaurants on orderss.restaurant_id = restaurants.restaurant_id left join order_items on orderss.order_id = order_items.order_id left join menu_items on order_items.item_id = menu_items.item_id left join payments on orderss.order_id = payments.order_id;

RESULT:

customer_id	first_name	phone_number	restaurant_name	item_name	order_id	Payment_id
1	Rakhi	9876543210	Chaat Masala	Vegetable Pulao	1	1
1	Rakhi	9876543210	Chaat Masala	Vegetable Pulao	1	2
2	Arjun	8876943910	Tandoori Flames	Aloo Pulao	2	3
2	Arjun	8876943910	Samosa Spot	Biryani	9	9
3	Karan	8679543210	Biryani Bliss	Chicken Tikka	3	4
4	shan	8124537890	Chai and Chaat	Biryani	4	5
4	shan	8124537890	Paneer Palace	Chai and Vada Pav	10	10
5	Rani	8325679013	Paneer Palace	Vegetable Pulao	5	6
6	Rashi	9376543223	Samosa Spot	Panner Tikka	6	7
6	Rashi	9376543223	Paneer Palace	Chai and Vada Pav	11	11
7	Anam	9886943910	Naan Nation	Samosa Pav and Cheese Samosa Pav	7	8
8	Vidya	9023451789	Naan Nation	Naan Bread	8	12

3. **To retrieve customers who have placed orders for more than one item**

RESULT:

Select

customers.customer_id,customers.first_name,customers.last_name,COUNT(DISTINCT order_items.item_id) AS distinct_items_ordered FROM customers JOIN orderss ON customers.customer_id = orderss.customer_id JOIN order_items ON orderss.order_id = order_items.order_id GROUP BY customers.customer_id, customers.first_name, customers.last_name HAVING COUNT(DISTINCT order_items.item_id) > 1;

customer_id	first_name	last_name	distinct_items_ordered
2	Arjun	Singh	2
4	shan	Kumar	2
6	Rashi	Gore	2

7.SUBQUERY:

1. **Select Items with highest page Price**

select restaurant_id,item_name,price from menu_items where price=(select max(price) from menu_items);

restaurant_id	item_name	price
3	Biryani	300

2. Select customers who have not order any items

select customer_id,first_name from customers where customer_id not in(select customer_id from customer_address);

customer_id	first_name
9	Aditya
10	Shruti
13	Bhumi

3. Select customers name who order any items:

select first_name from customers where customer_id in (select customer_id from orderss where customer_id in (select customer_id from orderss));

first_name
Rakhi
Arjun
Karan
shan
Rani
Rashi
Anam
Vidya