Database Management System Project

ONLINE FOOD ORDERING SYSTEM

PROBLEM STATEMENT :

Online Food Ordering Database Design aims to provide a comprehensive platform that enables customers to order their favorite food from various restaurants and have it delivered to their location. It helps to eliminate the difficulty of conventional queuing mechanisms. This system enhances the process of food order by a particular customer, from the ordering of food, looking at the menu being provided online, to the payment of the money as they wish, be it an online payment or the payment of cash, after the food is delivered.

It efficiently manages and organizes the details such as customer profiles, restaurant menus, order details, delivery logistics, and payment transactions, making the process for customers as easy as that being done in person. The customer is also provided with a facility of giving feedback regarding the food item received and the hotel as well, along with the rating. The database administrator can insert and delete data like restaurant details, delivery person details, food items, etc.,

ASSUMPTIONS:

- A customer with a unique customer id can order any number of different food items with a single order id.
- Several orders with different order ids can also be placed by a customer.
- Multiple order ids can have a single customer id.
- Each order id has a unique transaction id.
- Each type of food ordered has a unique id that links to give information on the name of the food item and its price per unit value.
- Each restaurant has its unique id to give information regarding its name, location and rating.
- Each delivery partner has a unique id to give information regarding their name, contact number and rating.
- Each customer has a unique id to give information regarding their name, contact number and address.
- Each customer can only order from one restaurant.

TABLES:

1. CUSTOMER

ATTRIBUTES	DATA TYPES	CONSTRAINTS
customer_id	INT (5)	PRIMARY KEY
customer_name	VARCHAR (30)	NOT NULL
customer_contact_number	VARCHAR (10)	NOT NULL
customer_address	VARCHAR (40)	NOT NULL

2. RESTAURANT

ATTRIBUTES	DATA TYPES	CONSTRAINTS
res_id	INT (3)	PRIMARY KEY
res_name	VARCHAR (20)	NOT NULL
res_location	VARCHAR (20)	NOT NULL
res_rating	DECIMAL (2,1)	NOT NULL

3. ORDER DETAIL

ATTRIBUTES	DATA TYPES	CONSTRAINTS
order_id	INT(4)	PRIMARY KEY
customer _id	INT (5)	NOT NULL, FOREIGN KEY
res_id	INT(3)	NOT NULL, FOREIGN KEY
deliverp_id	INT(4)	NOT NULL, FOREIGN KEY
order_status	VARCHAR(10)	NOT NULL
ordered_time	TIMESTAMP	NOT NULL
delivered_time	TIMESTAMP	NOT NULL

4. DELIVERY PARTNER

ATTRIBUTES	DATA TYPES	CONSTRAINTS
deliveryp_id	INT(4)	PRIMARY KEY
deliveryp_name	VARCHAR (30)	NOT NULL
deliveryp_contact_number	VARCHAR (10)	NOT NULL
deliveryp_avg_rating	DECIMAL(2,1)	NOT NULL

5. FOODS

ATTRIBUTES	DATA TYPES	CONSTRAINTS
food_id	INT (7)	PRIMARY KEY
food_name	VARCHAR (20)	NOT NULL
price_per_unit	DECIMAL (5,2)	NOT NULL

6. ORDER FOOD

ATTRIBUTES	DATA TYPES	CONSTRAINTS
order_id	INT (4)	PRIMARY KEY, FOREIGN KEY
food_id	INT (7)	PRIMARY KEY, FOREIGN KEY
quantity	INT (3)	NOT NULL

7. PAYMENT TABLE

ATTRIBUTES	DATA TYPES	CONSTRAINTS
payment_id	INT (8)	PRIMARY KEY
order_id	INT (4)	NOT NULL, FOREIGN KEY
payment_type	VARCHAR (20)	NOT NULL
payment_status	VARCHAR (10)	NOT NULL

FUNCTIONAL DEPENDENCIES AND PRIMARY KEY:

1. CUSTOMER:

customer_id -> { customer_name,contact_number,address}

As all the fields are depending on customer_id, (customer_id) +->R.

Hence, **customer_id** is a primary key.

2. RESTAURANT:

res_id -> {res_name,res_location,res_rating}

As all the fields are depending on restaurant_id, (restaurant_id) +->R.

Hence, **restaurant_id** is a primary key.

3. FOODS:

food_id -> {food_name,price_per_unit}

As all the fields are depending on food_id, (food_id) +->R.

Hence, **food_id** is a primary key.

4. DELIVERY PARTNER:

deliveryp_id
->{deliveryp_name,deliveryp_contact_number,deliveryp_avg_rating}

As all the fields are depending on deliveryp_id, (deliveryp_id) +->R.

Hence, **deliveryp_id** is a primary key.

5. ORDER DETAIL:

order_id -> {customer_id,res_id,deliverp_id,order_status,order_time,delivered_time}

As all the fields are depending on order_id, (order_id) +->R.

Hence, **order_id** is a primary key.

Also, customer_id,res_id, and deliveryp_id are foreign keys from customer, restaurant, and delivery_partner tables respectively.

6. ORDER FOOD:

{order_id + food_id} -> quantity

As all the fields are depending on (order_id + food_id), (order_id + food_id) +->R.

The combination of **order_id** and **food_id** acts as a composite key to uniquely identify each row.

Also, order_id and food_id are foreign keys from the order_detail and foods table respectively.

7. PAYMENT TABLE:

payment_id -> { payment_type,payment_status,order_id}

As all the fields are depending on payment_id, (payment_id) +->R.

Hence, **payment_id** is a primary key.

Also, order_id is a foreign key from the order_detail table.

NORMALISATION:

1. CUSTOMER:

Primary key: customer_id

All attributes depend on the customer_id, hence the table is in 2NF.

All attributes depend directly on customer_id hence the table is in 3NF.

All dependencies are from Candidate Key. Therefore the table is in BCNF.

2. RESTAURANT:

Primary key: res_id

All attributes depend on the res_id, hence the table is in 2NF.

All attributes depend directly on res_id hence the table is in 3NF.

All dependencies are from Candidate Key. Therefore the table is in BCNF.

3. FOODS:

Primary key: food_id

All attributes depend on the food_id, hence the table is in 2NF.

All attributes depend directly on food_id hence the table is in 3NF.

All dependencies are from Candidate Key. Therefore the table is in BCNF.

4. DELIVERY PARTNER:

Primary key: deliveryp_id

All attributes depend on the deliveryp_id, hence the table is in 2NF.

All attributes depend directly on deliveryp_id hence the table is in 3NF.

All dependencies are from Candidate Key. Therefore the table is in BCNF.

5. ORDER DETAIL:

Primary key: order_id

All attributes depend on the order_id, hence the table is in 2NF.

All attributes depend directly on order_id hence the table is in 3NF.

All dependencies are from Candidate Key. Therefore the table is in BCNF.

6. ORDER FOOD:

Primary key: order_id, food_id

All attributes depend on the order_id and food_id, hence the table is in 2NF.

All attributes depend directly on order_id and food_id, hence the table is in 3NF.

All dependencies are from Candidate Key. Therefore the table is in BCNF.

7. PAYMENT TABLE:

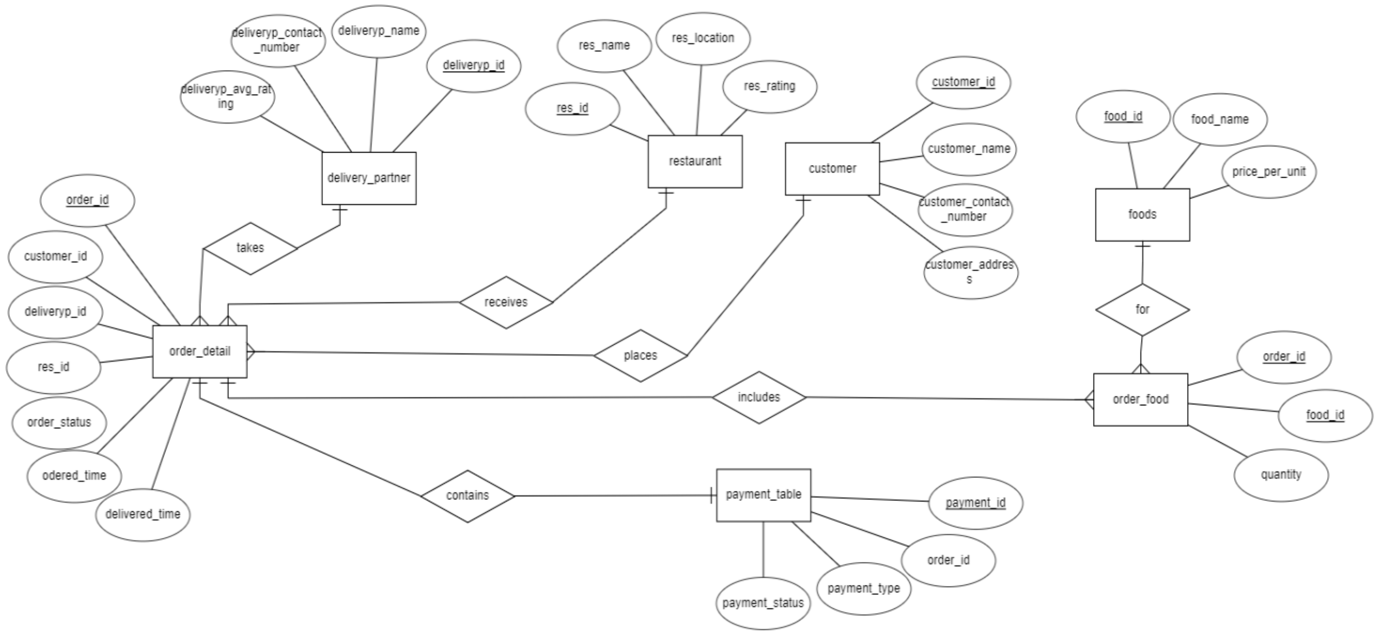
Primary key: payment_id

All attributes depend on the payment_id, hence the table is in 2NF.

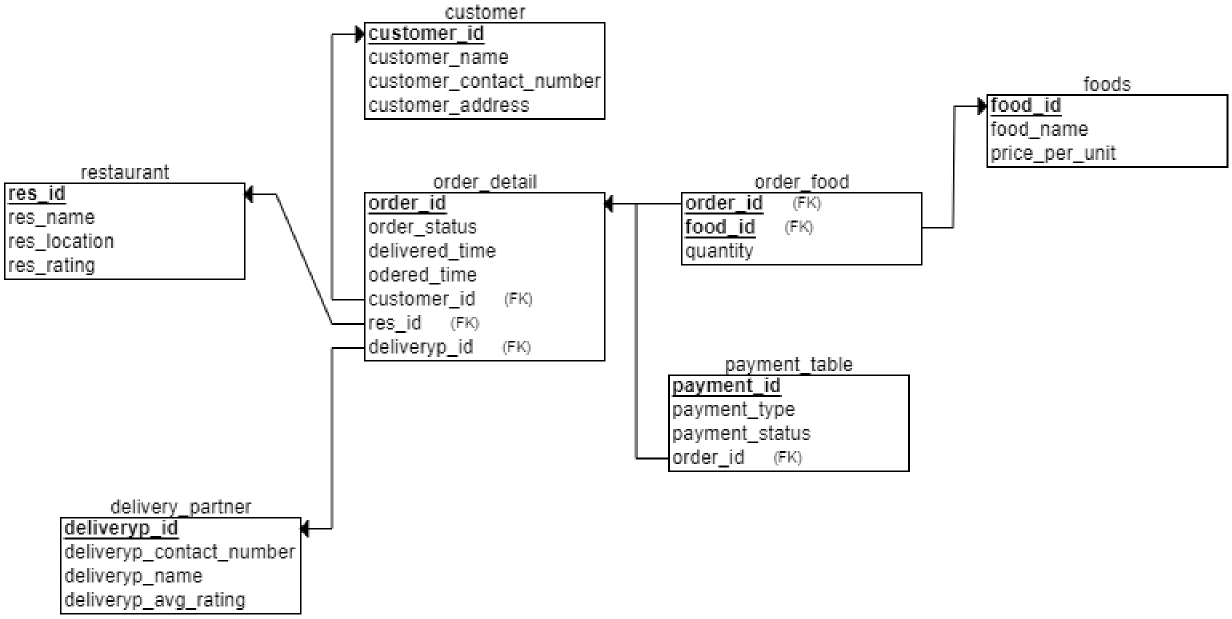
All attributes depend directly on payment_id hence the table is in 3NF.

All dependencies are from Candidate Key. Therefore the table is in BCNF.

Entity Relationship (ER) Diagram :



Relational Schema :



customer
customer_id
customer_name
customer_contact_number
customer_address

foods
food_id
food_name
price_per_unit

restaurant
res_id
res_name
res_location
res_rating

order_detail
order_id
order_status
delivered_time
odered_time
customer_id (FK)
res_id (FK)
deliveryp_id (FK)

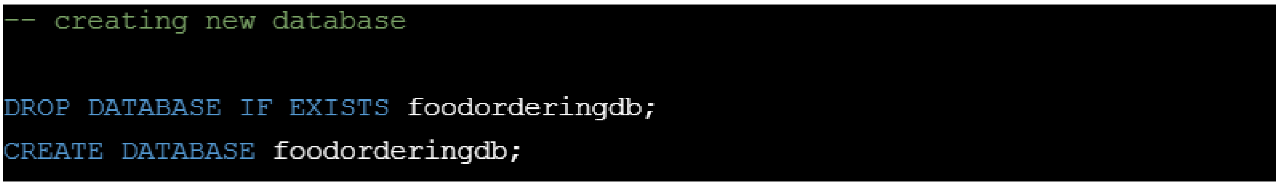
order_food
order_id (FK)
food_id (FK)
quantity

payment_table
payment_id
payment_type
payment_status
order_id (FK)

delivery_partner
deliveryp_id
deliveryp_contact_number
deliveryp_name
deliveryp_avg_rating

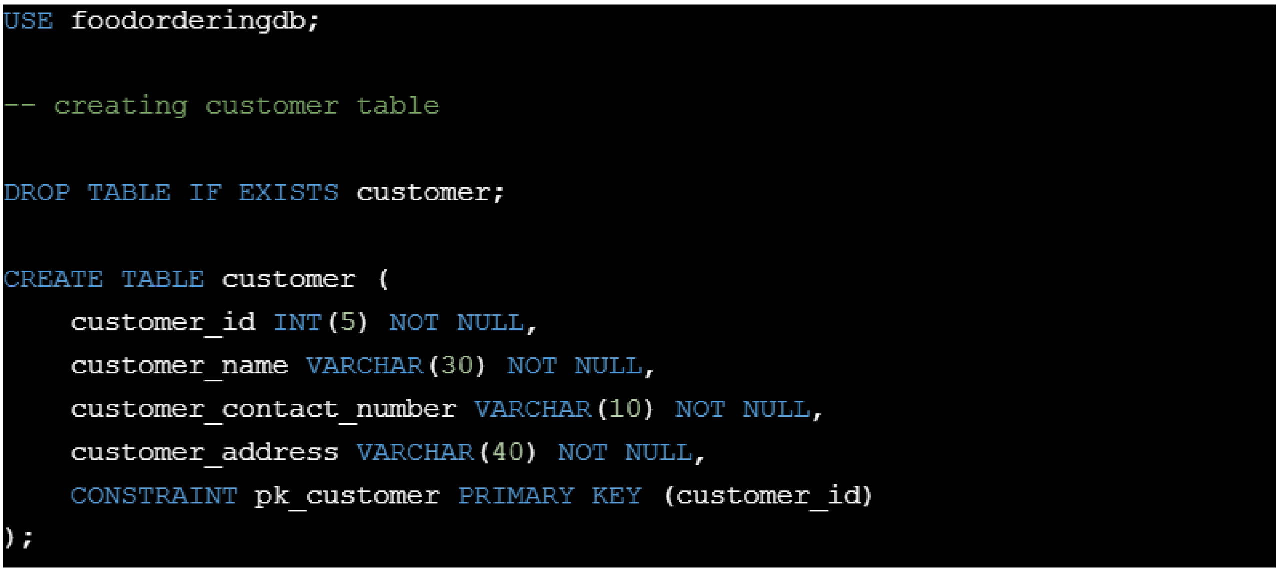
SQL CODE:

Creating Database :

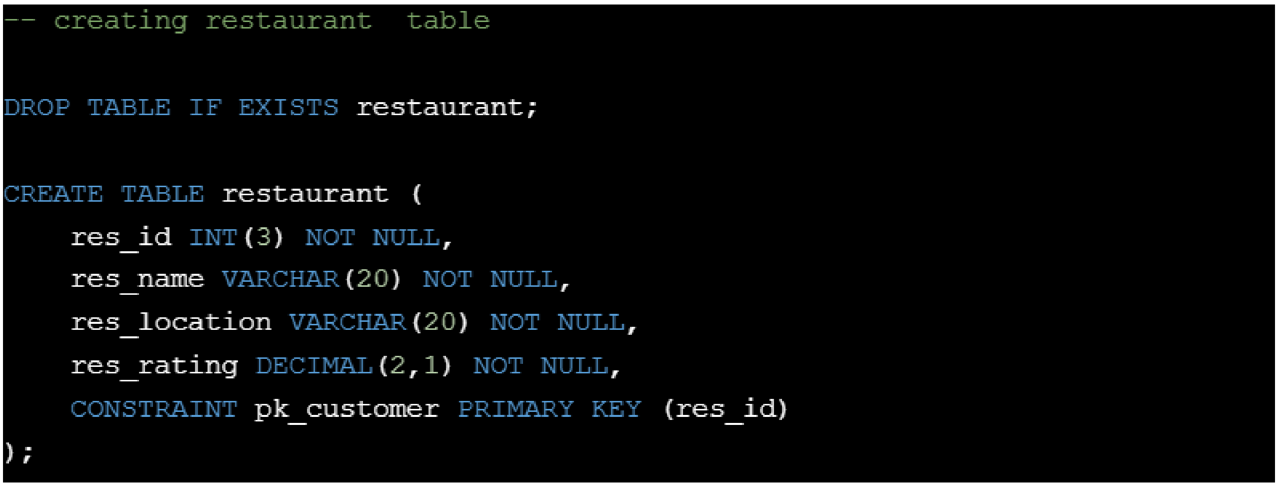


```
-- creating new database

DROP DATABASE IF EXISTS foodorderingdb;
CREATE DATABASE foodorderingdb;
```

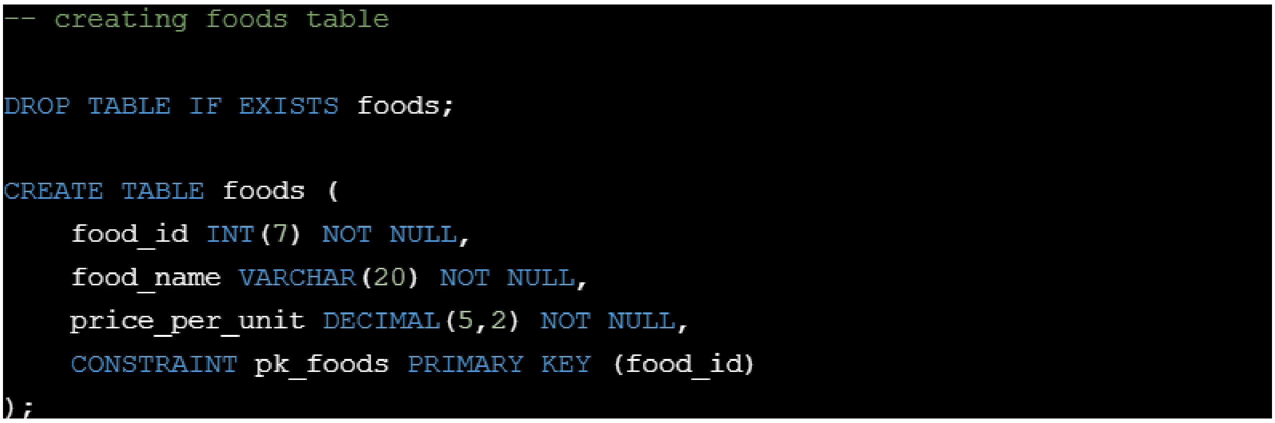
Creating Tables :



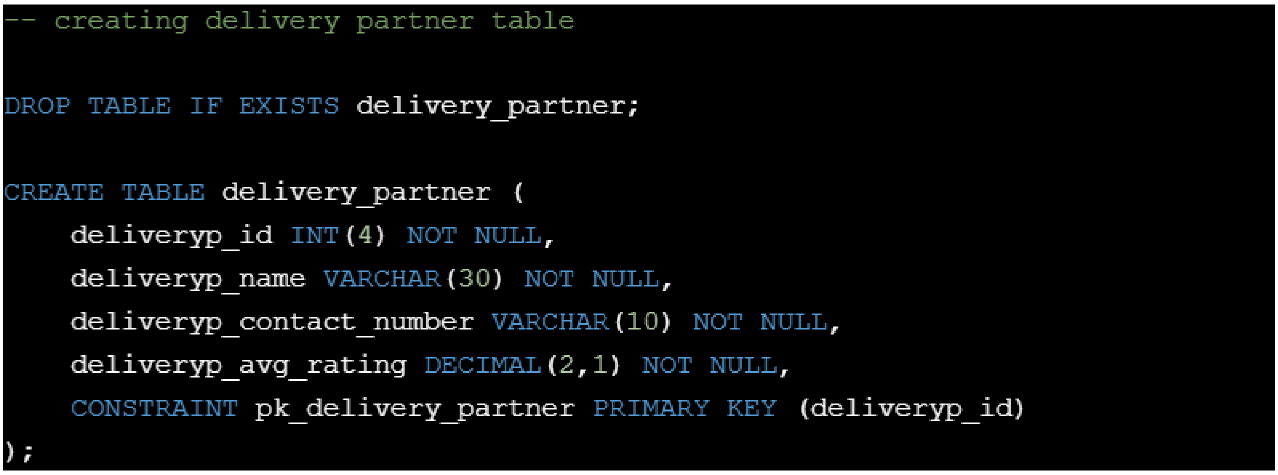
```
USE foodorderingdb;

-- creating customer table

DROP TABLE IF EXISTS customer;

CREATE TABLE customer (
    customer_id INT(5) NOT NULL,
    customer_name VARCHAR(30) NOT NULL,
    customer_contact_number VARCHAR(10) NOT NULL,
    customer_address VARCHAR(40) NOT NULL,
    CONSTRAINT pk_customer PRIMARY KEY (customer_id)
);
```



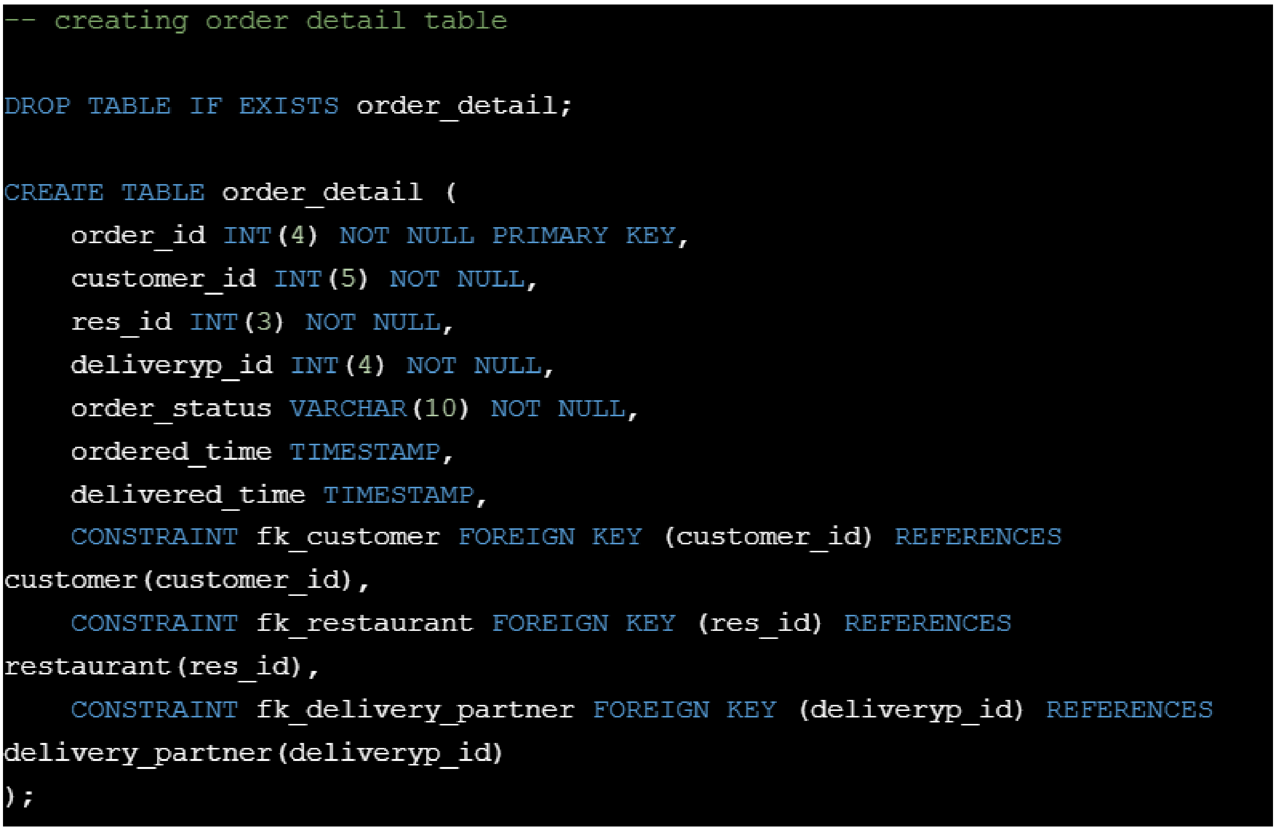
```
-- creating restaurant  table

DROP TABLE IF EXISTS restaurant;

CREATE TABLE restaurant (
    res_id INT(3) NOT NULL,
    res_name VARCHAR(20) NOT NULL,
    res_location VARCHAR(20) NOT NULL,
    res_rating DECIMAL(2,1) NOT NULL,
    CONSTRAINT pk_customer PRIMARY KEY (res_id)
);
```



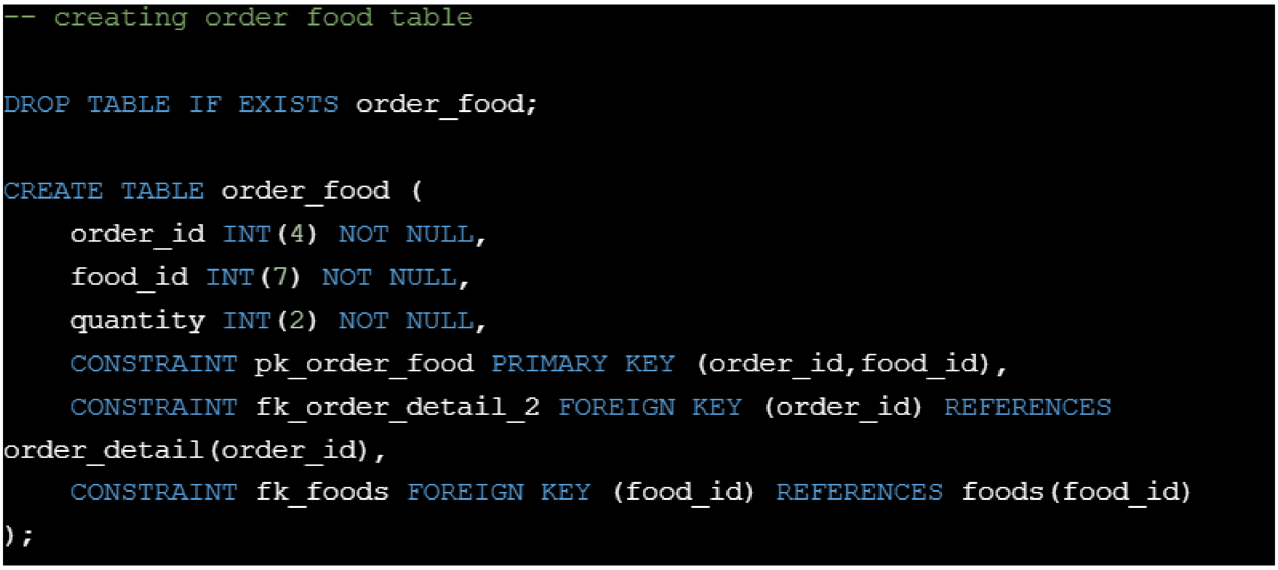
```
-- creating foods table

DROP TABLE IF EXISTS foods;

CREATE TABLE foods (
    food_id INT(7) NOT NULL,
    food_name VARCHAR(20) NOT NULL,
    price_per_unit DECIMAL(5,2) NOT NULL,
    CONSTRAINT pk_foods PRIMARY KEY (food_id)
);
```



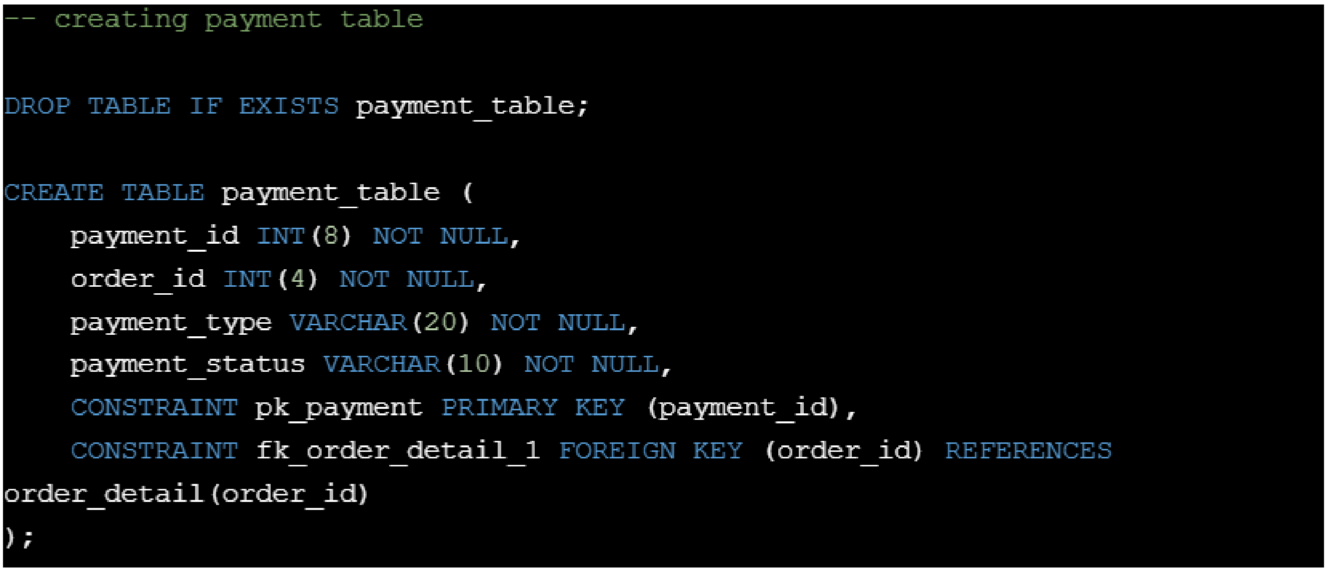
```sql
-- creating delivery partner table

DROP TABLE IF EXISTS delivery_partner;

CREATE TABLE delivery_partner (
    deliveryp_id INT(4) NOT NULL,
    deliveryp_name VARCHAR(30) NOT NULL,
    deliveryp_contact_number VARCHAR(10) NOT NULL,
    deliveryp_avg_rating DECIMAL(2,1) NOT NULL,
    CONSTRAINT pk_delivery_partner PRIMARY KEY (deliveryp_id)
);
```



```sql
-- creating order detail table

DROP TABLE IF EXISTS order_detail;

CREATE TABLE order_detail (
    order_id INT(4) NOT NULL PRIMARY KEY,
    customer_id INT(5) NOT NULL,
    res_id INT(3) NOT NULL,
    deliveryp_id INT(4) NOT NULL,
    order_status VARCHAR(10) NOT NULL,
    ordered_time TIMESTAMP,
    delivered_time TIMESTAMP,
    CONSTRAINT fk_customer FOREIGN KEY (customer_id) REFERENCES
customer(customer_id),
    CONSTRAINT fk_restaurant FOREIGN KEY (res_id) REFERENCES
restaurant(res_id),
    CONSTRAINT fk_delivery_partner FOREIGN KEY (deliveryp_id) REFERENCES
delivery_partner(deliveryp_id)
);
```

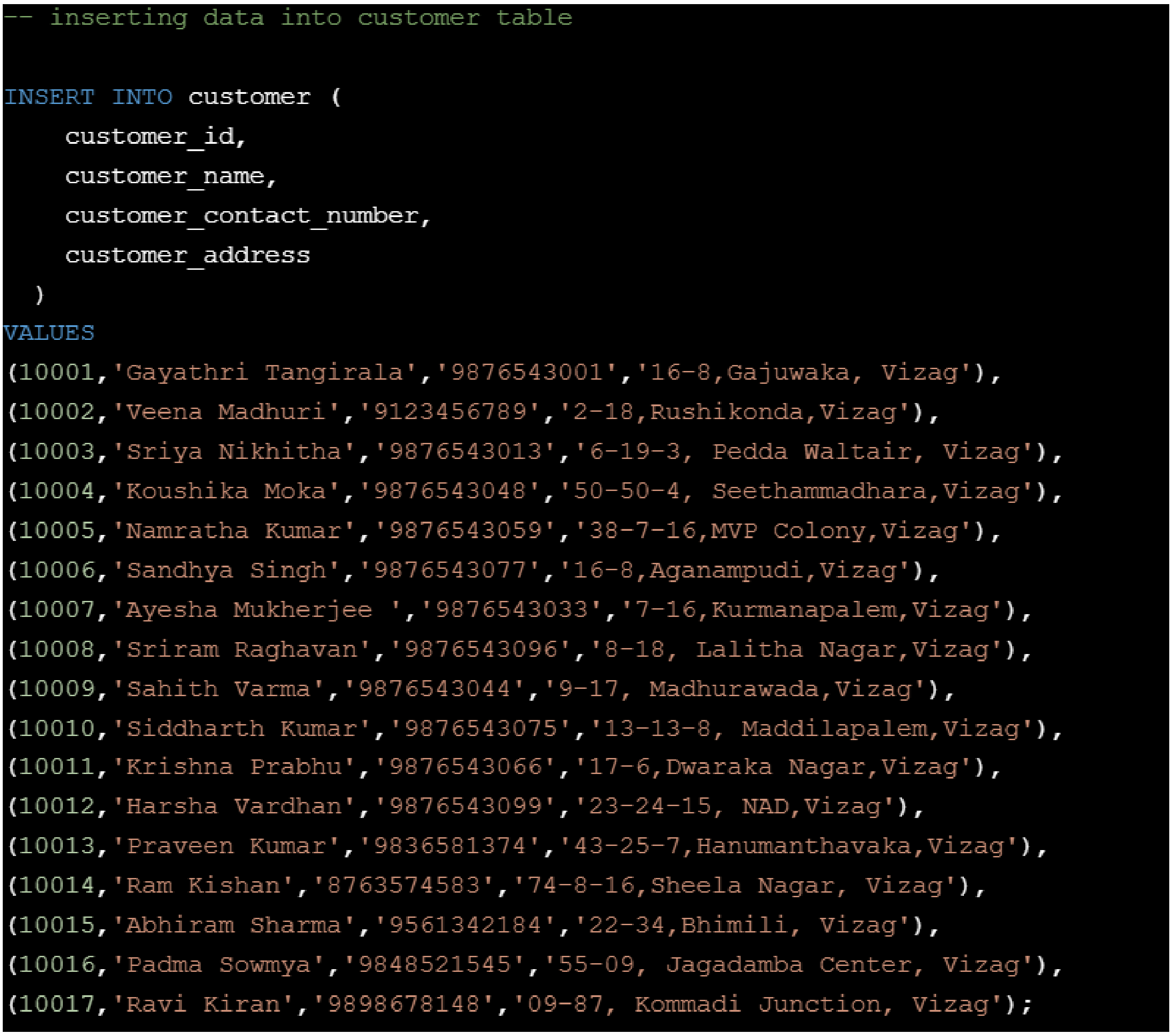


```sql
-- creating order food table

DROP TABLE IF EXISTS order_food;

CREATE TABLE order_food (
    order_id INT(4) NOT NULL,
    food_id INT(7) NOT NULL,
    quantity INT(2) NOT NULL,
    CONSTRAINT pk_order_food PRIMARY KEY (order_id,food_id),
    CONSTRAINT fk_order_detail_2 FOREIGN KEY (order_id) REFERENCES
order_detail(order_id),
    CONSTRAINT fk_foods FOREIGN KEY (food_id) REFERENCES foods(food_id)
);
```

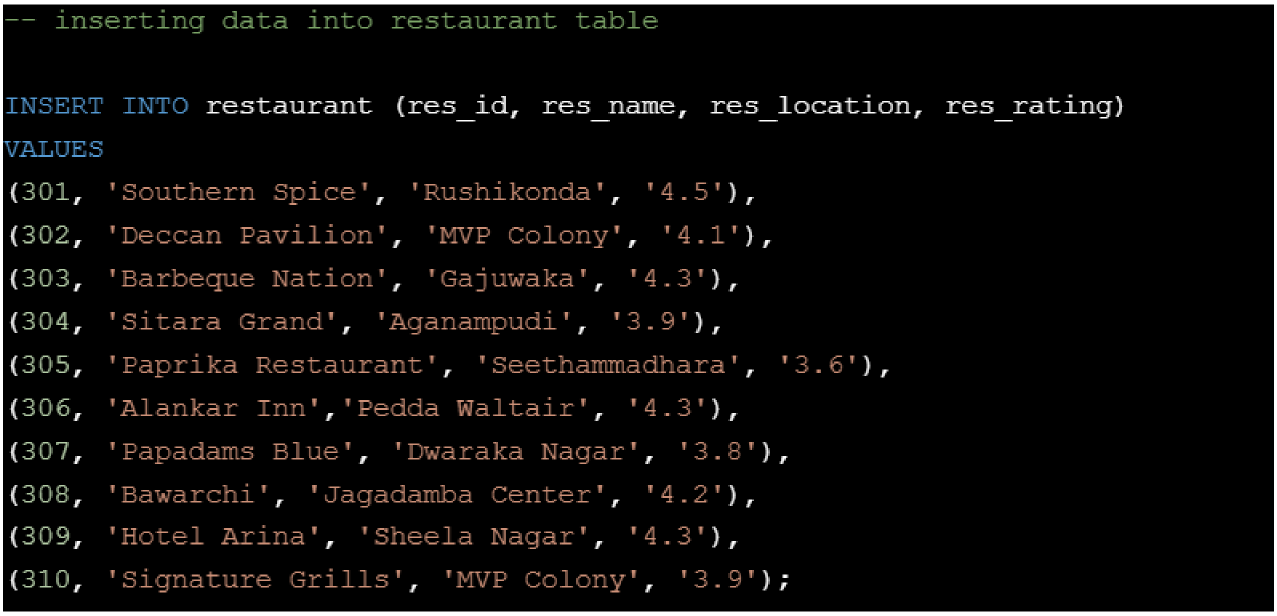


```
-- creating payment table

DROP TABLE IF EXISTS payment_table;

CREATE TABLE payment_table (
    payment_id INT(8) NOT NULL,
    order_id INT(4) NOT NULL,
    payment_type VARCHAR(20) NOT NULL,
    payment_status VARCHAR(10) NOT NULL,
    CONSTRAINT pk_payment PRIMARY KEY (payment_id),
    CONSTRAINT fk_order_detail_1 FOREIGN KEY (order_id) REFERENCES
order_detail(order_id)
);
```

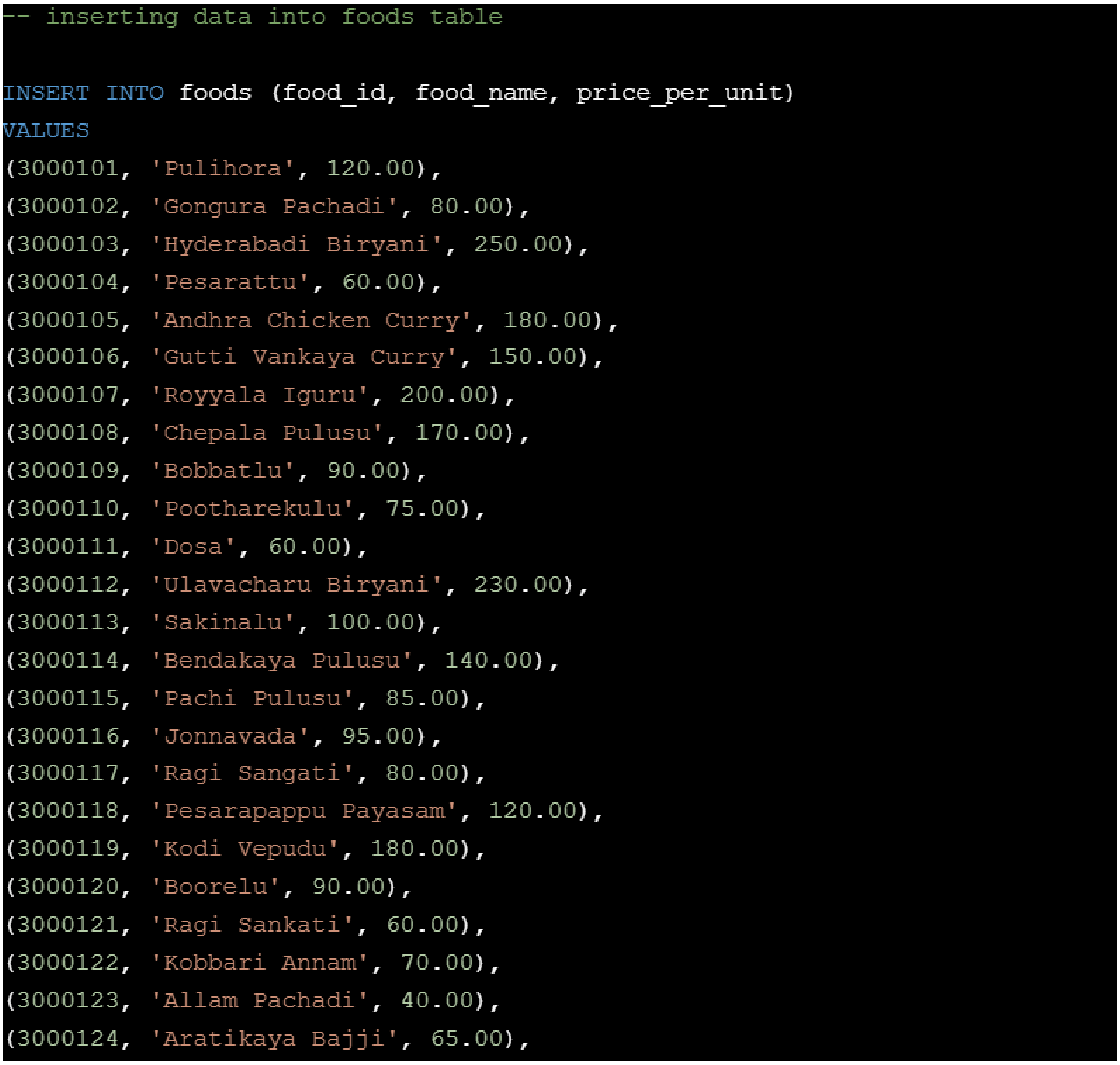
Inserting Data Into Tables :



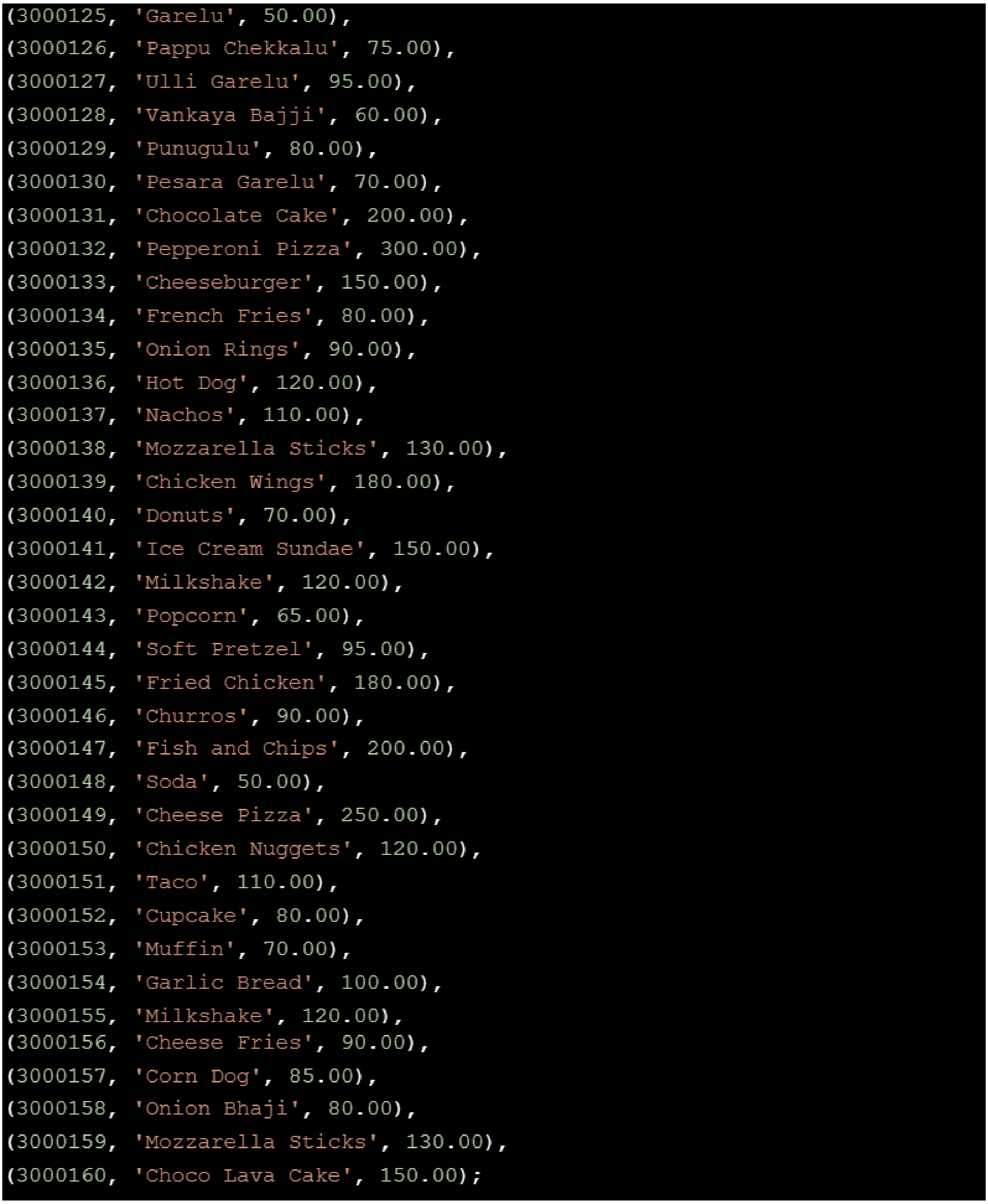
```
-- inserting data into customer table

INSERT INTO customer (
    customer_id,
    customer_name,
    customer_contact_number,
    customer_address
  )
VALUES
(10001,'Gayathri Tangirala','9876543001','16-8,Gajuwaka, Vizag'),
(10002,'Veena Madhuri','9123456789','2-18,Rushikonda,Vizag'),
(10003,'Sriya Nikhitha','9876543013','6-19-3, Pedda Waltair, Vizag'),
(10004,'Koushika Moka','9876543048','50-50-4, Seethammadhara,Vizag'),
(10005,'Namratha Kumar','9876543059','38-7-16,MVP Colony,Vizag'),
(10006,'Sandhya Singh','9876543077','16-8,Aganampudi,Vizag'),
(10007,'Ayesha Mukherjee ','9876543033','7-16,Kurmanapalem,Vizag'),
(10008,'Sriram Raghavan','9876543096','8-18, Lalitha Nagar,Vizag'),
(10009,'Sahith Varma','9876543044','9-17, Madhurawada,Vizag'),
(10010,'Siddharth Kumar','9876543075','13-13-8, Maddilapalem,Vizag'),
(10011,'Krishna Prabhu','9876543066','17-6,Dwaraka Nagar,Vizag'),
(10012,'Harsha Vardhan','9876543099','23-24-15, NAD,Vizag'),
(10013,'Praveen Kumar','9836581374','43-25-7,Hanumanthavaka,Vizag'),
(10014,'Ram Kishan','8763574583','74-8-16,Sheela Nagar, Vizag'),
(10015,'Abhiram Sharma','9561342184','22-34,Bhimili, Vizag'),
(10016,'Padma Sowmya','9848521545','55-09, Jagadamba Center, Vizag'),
(10017,'Ravi Kiran','9898678148','09-87, Kommadi Junction, Vizag');
```



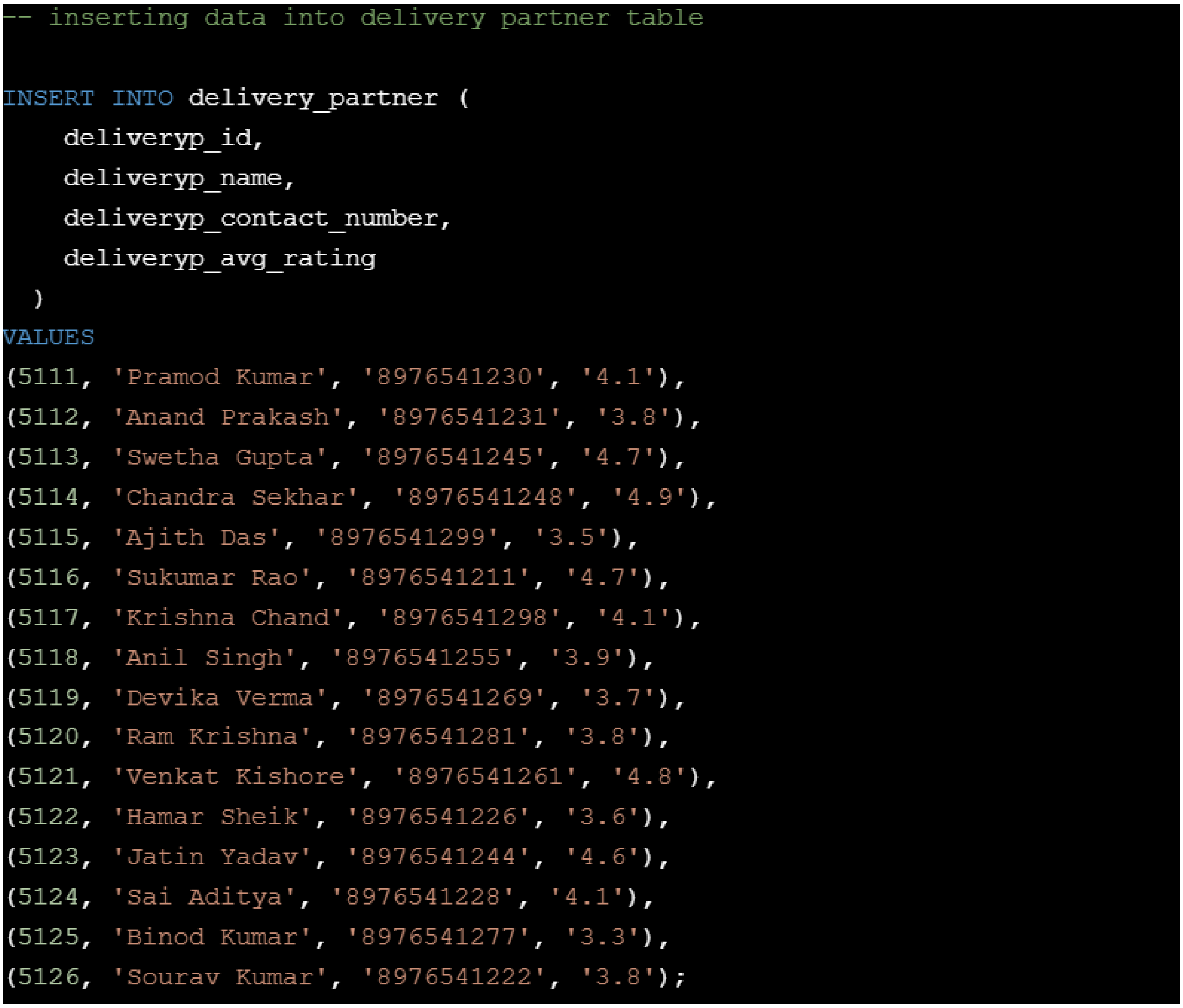
```sql
-- inserting data into restaurant table

INSERT INTO restaurant (res_id, res_name, res_location, res_rating)
VALUES
(301, 'Southern Spice', 'Rushikonda', '4.5'),
(302, 'Deccan Pavilion', 'MVP Colony', '4.1'),
(303, 'Barbeque Nation', 'Gajuwaka', '4.3'),
(304, 'Sitara Grand', 'Aganampudi', '3.9'),
(305, 'Paprika Restaurant', 'Seethammadhara', '3.6'),
(306, 'Alankar Inn','Pedda Waltair', '4.3'),
(307, 'Papadams Blue', 'Dwaraka Nagar', '3.8'),
(308, 'Bawarchi', 'Jagadamba Center', '4.2'),
(309, 'Hotel Arina', 'Sheela Nagar', '4.3'),
(310, 'Signature Grills', 'MVP Colony', '3.9');
```



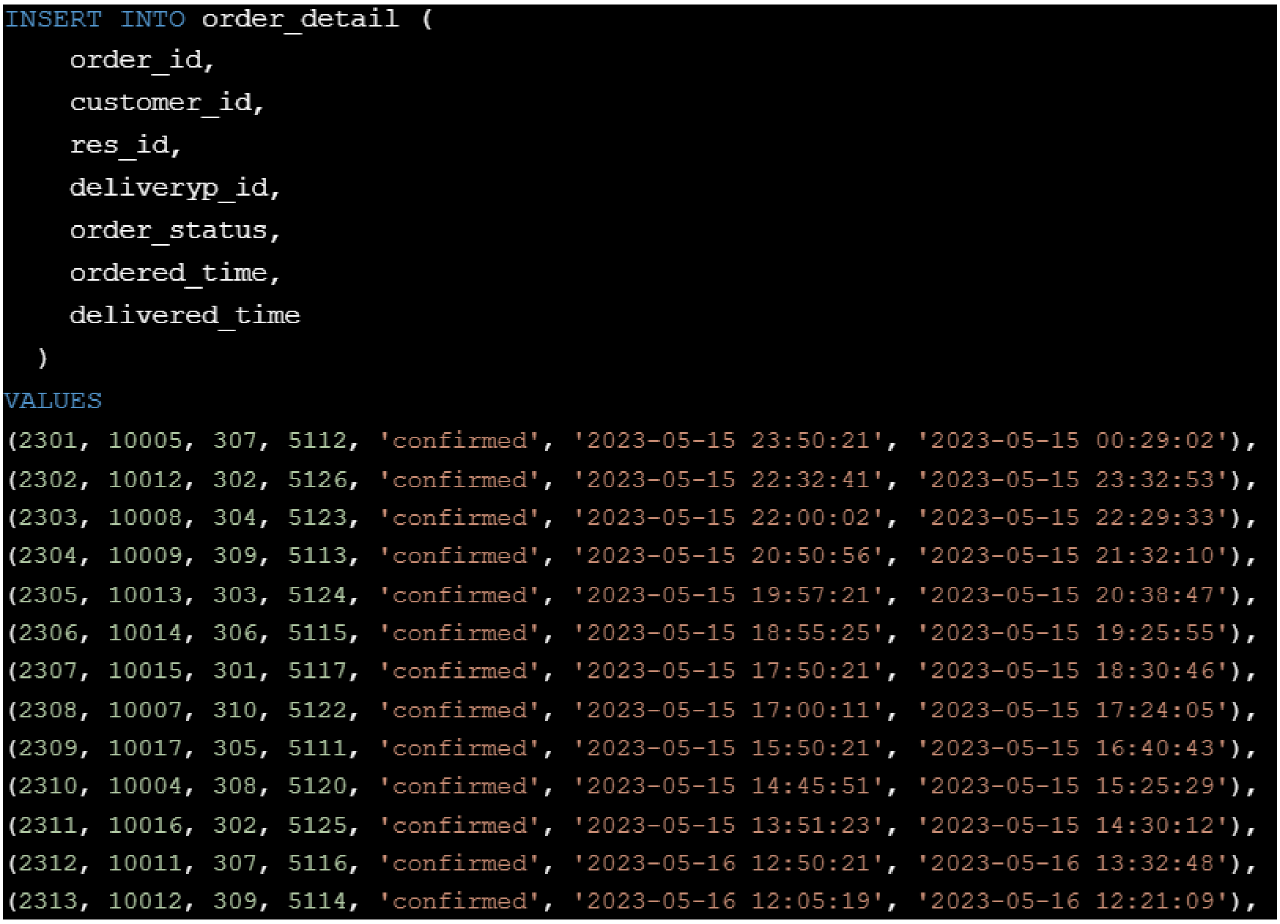
```sql
-- inserting data into foods table

INSERT INTO foods (food_id, food_name, price_per_unit)
VALUES
(3000101, 'Pulihora', 120.00),
(3000102, 'Gongura Pachadi', 80.00),
(3000103, 'Hyderabadi Biryani', 250.00),
(3000104, 'Pesarattu', 60.00),
(3000105, 'Andhra Chicken Curry', 180.00),
(3000106, 'Gutti Vankaya Curry', 150.00),
(3000107, 'Royyala Iguru', 200.00),
(3000108, 'Chepala Pulusu', 170.00),
(3000109, 'Bobbatlu', 90.00),
(3000110, 'Pootharekulu', 75.00),
(3000111, 'Dosa', 60.00),
(3000112, 'Ulavacharu Biryani', 230.00),
(3000113, 'Sakinalu', 100.00),
(3000114, 'Bendakaya Pulusu', 140.00),
(3000115, 'Pachi Pulusu', 85.00),
(3000116, 'Jonnavada', 95.00),
(3000117, 'Ragi Sangati', 80.00),
(3000118, 'Pesarapappu Payasam', 120.00),
(3000119, 'Kodi Vepudu', 180.00),
(3000120, 'Boorelu', 90.00),
(3000121, 'Ragi Sankati', 60.00),
(3000122, 'Kobbari Annam', 70.00),
(3000123, 'Allam Pachadi', 40.00),
(3000124, 'Aratikaya Bajji', 65.00),
```



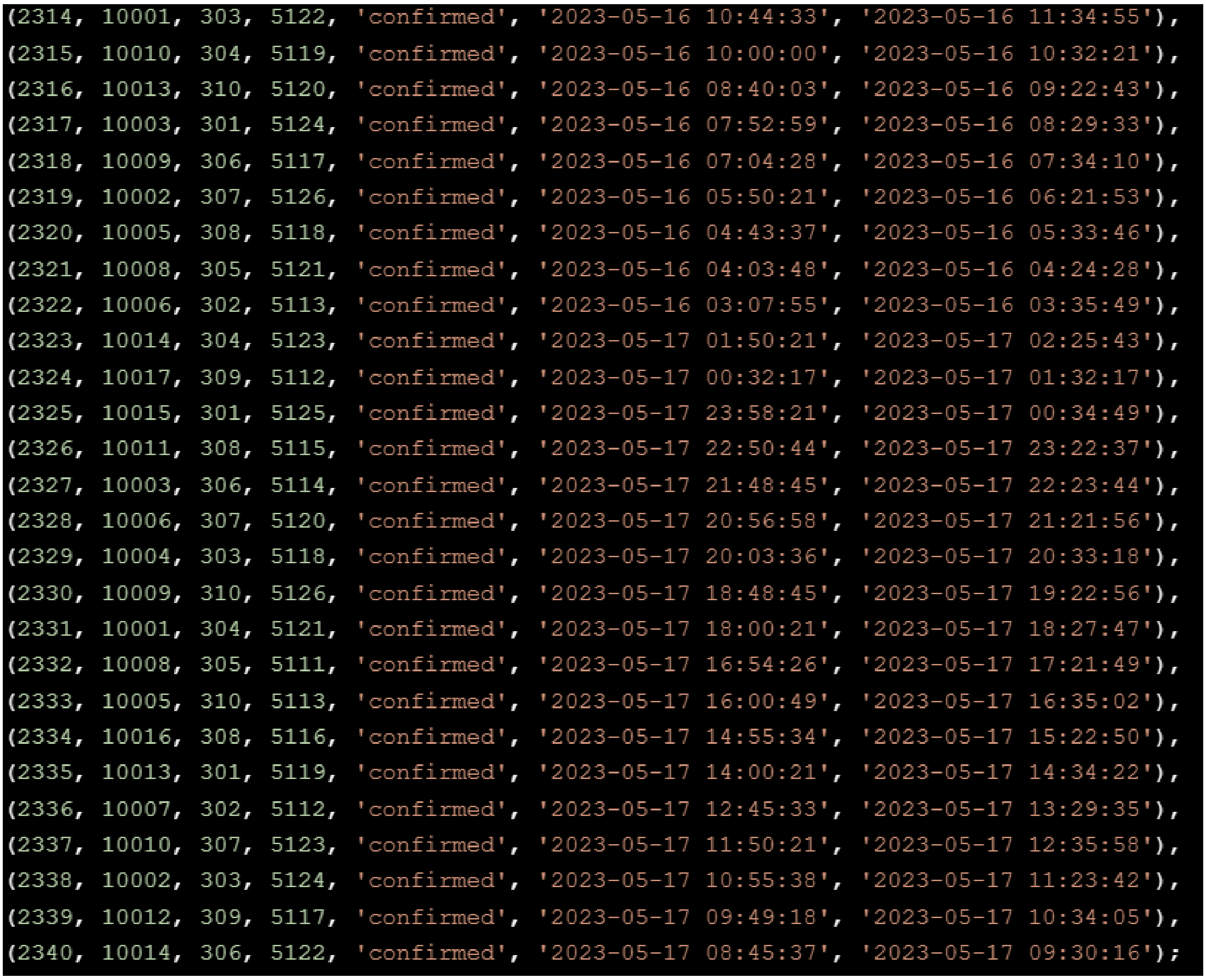
```
(3000125, 'Garelu', 50.00),
(3000126, 'Pappu Chekkalu', 75.00),
(3000127, 'Ulli Garelu', 95.00),
(3000128, 'Vankaya Bajji', 60.00),
(3000129, 'Punugulu', 80.00),
(3000130, 'Pesara Garelu', 70.00),
(3000131, 'Chocolate Cake', 200.00),
(3000132, 'Pepperoni Pizza', 300.00),
(3000133, 'Cheeseburger', 150.00),
(3000134, 'French Fries', 80.00),
(3000135, 'Onion Rings', 90.00),
(3000136, 'Hot Dog', 120.00),
(3000137, 'Nachos', 110.00),
(3000138, 'Mozzarella Sticks', 130.00),
(3000139, 'Chicken Wings', 180.00),
(3000140, 'Donuts', 70.00),
(3000141, 'Ice Cream Sundae', 150.00),
(3000142, 'Milkshake', 120.00),
(3000143, 'Popcorn', 65.00),
(3000144, 'Soft Pretzel', 95.00),
(3000145, 'Fried Chicken', 180.00),
(3000146, 'Churros', 90.00),
(3000147, 'Fish and Chips', 200.00),
(3000148, 'Soda', 50.00),
(3000149, 'Cheese Pizza', 250.00),
(3000150, 'Chicken Nuggets', 120.00),
(3000151, 'Taco', 110.00),
(3000152, 'Cupcake', 80.00),
(3000153, 'Muffin', 70.00),
(3000154, 'Garlic Bread', 100.00),
(3000155, 'Milkshake', 120.00),
(3000156, 'Cheese Fries', 90.00),
(3000157, 'Corn Dog', 85.00),
(3000158, 'Onion Bhaji', 80.00),
(3000159, 'Mozzarella Sticks', 130.00),
(3000160, 'Choco Lava Cake', 150.00);
```



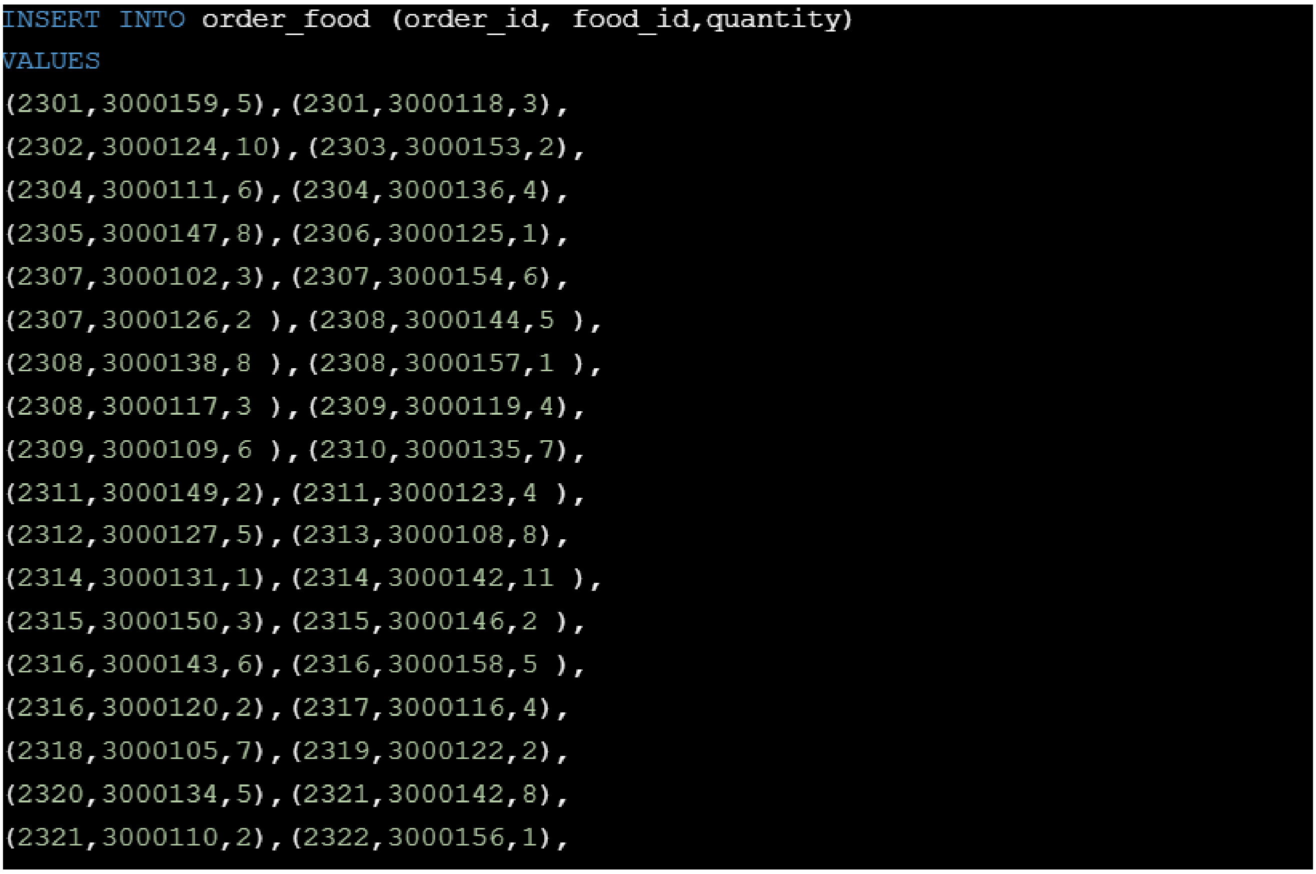
```sql
-- inserting data into delivery partner table

INSERT INTO delivery_partner (
    deliveryp_id,
    deliveryp_name,
    deliveryp_contact_number,
    deliveryp_avg_rating
  )
VALUES
(5111, 'Pramod Kumar', '8976541230', '4.1'),
(5112, 'Anand Prakash', '8976541231', '3.8'),
(5113, 'Swetha Gupta', '8976541245', '4.7'),
(5114, 'Chandra Sekhar', '8976541248', '4.9'),
(5115, 'Ajith Das', '8976541299', '3.5'),
(5116, 'Sukumar Rao', '8976541211', '4.7'),
(5117, 'Krishna Chand', '8976541298', '4.1'),
(5118, 'Anil Singh', '8976541255', '3.9'),
(5119, 'Devika Verma', '8976541269', '3.7'),
(5120, 'Ram Krishna', '8976541281', '3.8'),
(5121, 'Venkat Kishore', '8976541261', '4.8'),
(5122, 'Hamar Sheik', '8976541226', '3.6'),
(5123, 'Jatin Yadav', '8976541244', '4.6'),
(5124, 'Sai Aditya', '8976541228', '4.1'),
(5125, 'Binod Kumar', '8976541277', '3.3'),
(5126, 'Sourav Kumar', '8976541222', '3.8');
```



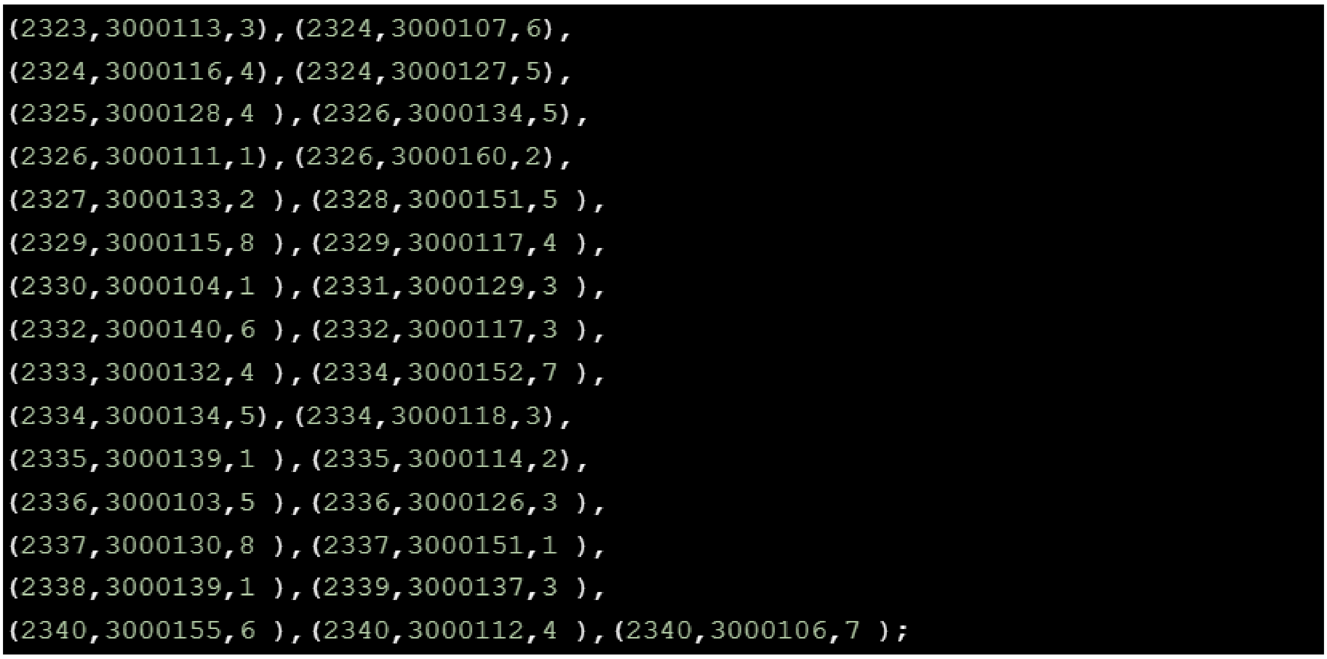
```sql
INSERT INTO order_detail (
    order_id,
    customer_id,
    res_id,
    deliveryp_id,
    order_status,
    ordered_time,
    delivered_time
  )
VALUES
(2301, 10005, 307, 5112, 'confirmed', '2023-05-15 23:50:21', '2023-05-15 00:29:02'),
(2302, 10012, 302, 5126, 'confirmed', '2023-05-15 22:32:41', '2023-05-15 23:32:53'),
(2303, 10008, 304, 5123, 'confirmed', '2023-05-15 22:00:02', '2023-05-15 22:29:33'),
(2304, 10009, 309, 5113, 'confirmed', '2023-05-15 20:50:56', '2023-05-15 21:32:10'),
(2305, 10013, 303, 5124, 'confirmed', '2023-05-15 19:57:21', '2023-05-15 20:38:47'),
(2306, 10014, 306, 5115, 'confirmed', '2023-05-15 18:55:25', '2023-05-15 19:25:55'),
(2307, 10015, 301, 5117, 'confirmed', '2023-05-15 17:50:21', '2023-05-15 18:30:46'),
(2308, 10007, 310, 5122, 'confirmed', '2023-05-15 17:00:11', '2023-05-15 17:24:05'),
(2309, 10017, 305, 5111, 'confirmed', '2023-05-15 15:50:21', '2023-05-15 16:40:43'),
(2310, 10004, 308, 5120, 'confirmed', '2023-05-15 14:45:51', '2023-05-15 15:25:29'),
(2311, 10016, 302, 5125, 'confirmed', '2023-05-15 13:51:23', '2023-05-15 14:30:12'),
(2312, 10011, 307, 5116, 'confirmed', '2023-05-16 12:50:21', '2023-05-16 13:32:48'),
(2313, 10012, 309, 5114, 'confirmed', '2023-05-16 12:05:19', '2023-05-16 12:21:09'),
```



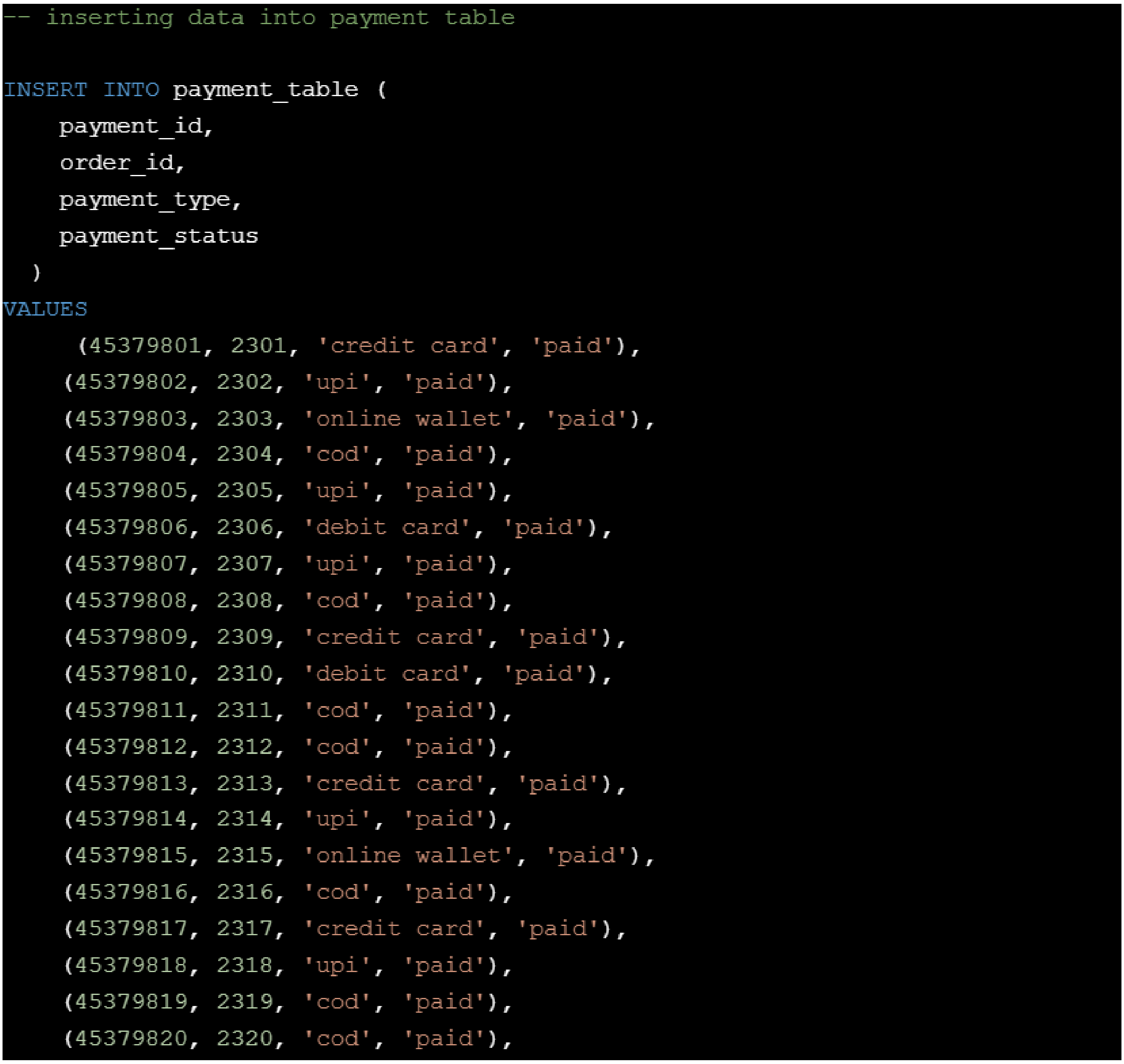
```
(2314, 10001, 303, 5122, 'confirmed', '2023-05-16 10:44:33', '2023-05-16 11:34:55'),
(2315, 10010, 304, 5119, 'confirmed', '2023-05-16 10:00:00', '2023-05-16 10:32:21'),
(2316, 10013, 310, 5120, 'confirmed', '2023-05-16 08:40:03', '2023-05-16 09:22:43'),
(2317, 10003, 301, 5124, 'confirmed', '2023-05-16 07:52:59', '2023-05-16 08:29:33'),
(2318, 10009, 306, 5117, 'confirmed', '2023-05-16 07:04:28', '2023-05-16 07:34:10'),
(2319, 10002, 307, 5126, 'confirmed', '2023-05-16 05:50:21', '2023-05-16 06:21:53'),
(2320, 10005, 308, 5118, 'confirmed', '2023-05-16 04:43:37', '2023-05-16 05:33:46'),
(2321, 10008, 305, 5121, 'confirmed', '2023-05-16 04:03:48', '2023-05-16 04:24:28'),
(2322, 10006, 302, 5113, 'confirmed', '2023-05-16 03:07:55', '2023-05-16 03:35:49'),
(2323, 10014, 304, 5123, 'confirmed', '2023-05-17 01:50:21', '2023-05-17 02:25:43'),
(2324, 10017, 309, 5112, 'confirmed', '2023-05-17 00:32:17', '2023-05-17 01:32:17'),
(2325, 10015, 301, 5125, 'confirmed', '2023-05-17 23:58:21', '2023-05-17 00:34:49'),
(2326, 10011, 308, 5115, 'confirmed', '2023-05-17 22:50:44', '2023-05-17 23:22:37'),
(2327, 10003, 306, 5114, 'confirmed', '2023-05-17 21:48:45', '2023-05-17 22:23:44'),
(2328, 10006, 307, 5120, 'confirmed', '2023-05-17 20:56:58', '2023-05-17 21:21:56'),
(2329, 10004, 303, 5118, 'confirmed', '2023-05-17 20:03:36', '2023-05-17 20:33:18'),
(2330, 10009, 310, 5126, 'confirmed', '2023-05-17 18:48:45', '2023-05-17 19:22:56'),
(2331, 10001, 304, 5121, 'confirmed', '2023-05-17 18:00:21', '2023-05-17 18:27:47'),
(2332, 10008, 305, 5111, 'confirmed', '2023-05-17 16:54:26', '2023-05-17 17:21:49'),
(2333, 10005, 310, 5113, 'confirmed', '2023-05-17 16:00:49', '2023-05-17 16:35:02'),
(2334, 10016, 308, 5116, 'confirmed', '2023-05-17 14:55:34', '2023-05-17 15:22:50'),
(2335, 10013, 301, 5119, 'confirmed', '2023-05-17 14:00:21', '2023-05-17 14:34:22'),
(2336, 10007, 302, 5112, 'confirmed', '2023-05-17 12:45:33', '2023-05-17 13:29:35'),
(2337, 10010, 307, 5123, 'confirmed', '2023-05-17 11:50:21', '2023-05-17 12:35:58'),
(2338, 10002, 303, 5124, 'confirmed', '2023-05-17 10:55:38', '2023-05-17 11:23:42'),
(2339, 10012, 309, 5117, 'confirmed', '2023-05-17 09:49:18', '2023-05-17 10:34:05'),
(2340, 10014, 306, 5122, 'confirmed', '2023-05-17 08:45:37', '2023-05-17 09:30:16');
```



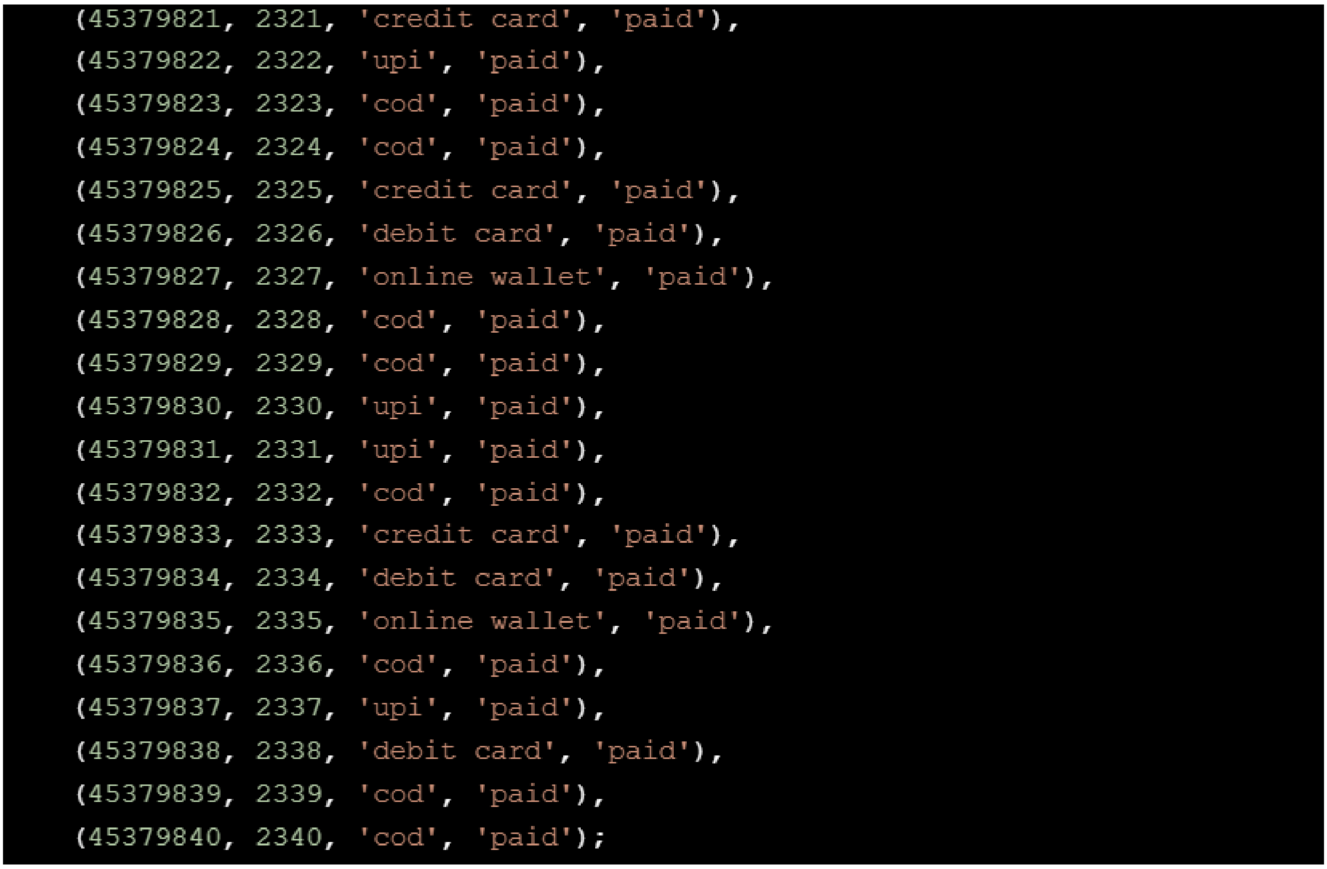
```
INSERT INTO order_food (order_id, food_id,quantity)
VALUES
(2301,3000159,5),(2301,3000118,3),
(2302,3000124,10),(2303,3000153,2),
(2304,3000111,6),(2304,3000136,4),
(2305,3000147,8),(2306,3000125,1),
(2307,3000102,3),(2307,3000154,6),
(2307,3000126,2 ),(2308,3000144,5 ),
(2308,3000138,8 ),(2308,3000157,1 ),
(2308,3000117,3 ),(2309,3000119,4),
(2309,3000109,6 ),(2310,3000135,7),
(2311,3000149,2),(2311,3000123,4 ),
(2312,3000127,5),(2313,3000108,8),
(2314,3000131,1),(2314,3000142,11 ),
(2315,3000150,3),(2315,3000146,2 ),
(2316,3000143,6),(2316,3000158,5 ),
(2316,3000120,2),(2317,3000116,4),
(2318,3000105,7),(2319,3000122,2),
(2320,3000134,5),(2321,3000142,8),
(2321,3000110,2),(2322,3000156,1),
```



```
(2323,3000113,3),(2324,3000107,6),
(2324,3000116,4),(2324,3000127,5),
(2325,3000128,4 ),(2326,3000134,5),
(2326,3000111,1),(2326,3000160,2),
(2327,3000133,2 ),(2328,3000151,5 ),
(2329,3000115,8 ),(2329,3000117,4 ),
(2330,3000104,1 ),(2331,3000129,3 ),
(2332,3000140,6 ),(2332,3000117,3 ),
(2333,3000132,4 ),(2334,3000152,7 ),
(2334,3000134,5),(2334,3000118,3),
(2335,3000139,1 ),(2335,3000114,2),
(2336,3000103,5 ),(2336,3000126,3 ),
(2337,3000130,8 ),(2337,3000151,1 ),
(2338,3000139,1 ),(2339,3000137,3 ),
(2340,3000155,6 ),(2340,3000112,4 ),(2340,3000106,7 );
```



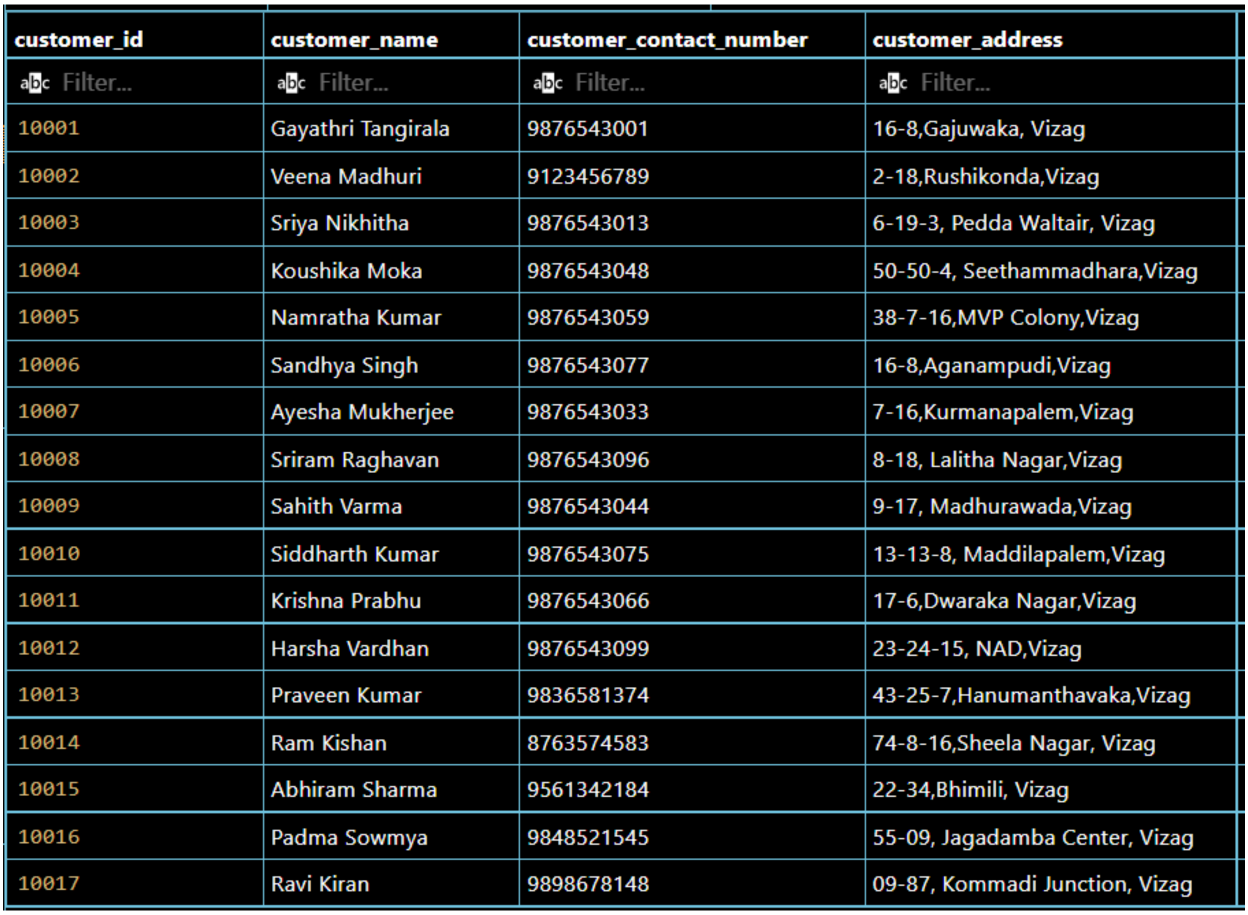
```
-- inserting data into payment table

INSERT INTO payment_table (
    payment_id,
    order_id,
    payment_type,
    payment_status
  )
VALUES
     (45379801, 2301, 'credit card', 'paid'),
    (45379802, 2302, 'upi', 'paid'),
    (45379803, 2303, 'online wallet', 'paid'),
    (45379804, 2304, 'cod', 'paid'),
    (45379805, 2305, 'upi', 'paid'),
    (45379806, 2306, 'debit card', 'paid'),
    (45379807, 2307, 'upi', 'paid'),
    (45379808, 2308, 'cod', 'paid'),
    (45379809, 2309, 'credit card', 'paid'),
    (45379810, 2310, 'debit card', 'paid'),
    (45379811, 2311, 'cod', 'paid'),
    (45379812, 2312, 'cod', 'paid'),
    (45379813, 2313, 'credit card', 'paid'),
    (45379814, 2314, 'upi', 'paid'),
    (45379815, 2315, 'online wallet', 'paid'),
    (45379816, 2316, 'cod', 'paid'),
    (45379817, 2317, 'credit card', 'paid'),
    (45379818, 2318, 'upi', 'paid'),
    (45379819, 2319, 'cod', 'paid'),
    (45379820, 2320, 'cod', 'paid'),
```



```
    (45379821, 2321, 'credit card', 'paid'),
    (45379822, 2322, 'upi', 'paid'),
    (45379823, 2323, 'cod', 'paid'),
    (45379824, 2324, 'cod', 'paid'),
    (45379825, 2325, 'credit card', 'paid'),
    (45379826, 2326, 'debit card', 'paid'),
    (45379827, 2327, 'online wallet', 'paid'),
    (45379828, 2328, 'cod', 'paid'),
    (45379829, 2329, 'cod', 'paid'),
    (45379830, 2330, 'upi', 'paid'),
    (45379831, 2331, 'upi', 'paid'),
    (45379832, 2332, 'cod', 'paid'),
    (45379833, 2333, 'credit card', 'paid'),
    (45379834, 2334, 'debit card', 'paid'),
    (45379835, 2335, 'online wallet', 'paid'),
    (45379836, 2336, 'cod', 'paid'),
    (45379837, 2337, 'upi', 'paid'),
    (45379838, 2338, 'debit card', 'paid'),
    (45379839, 2339, 'cod', 'paid'),
    (45379840, 2340, 'cod', 'paid');
```

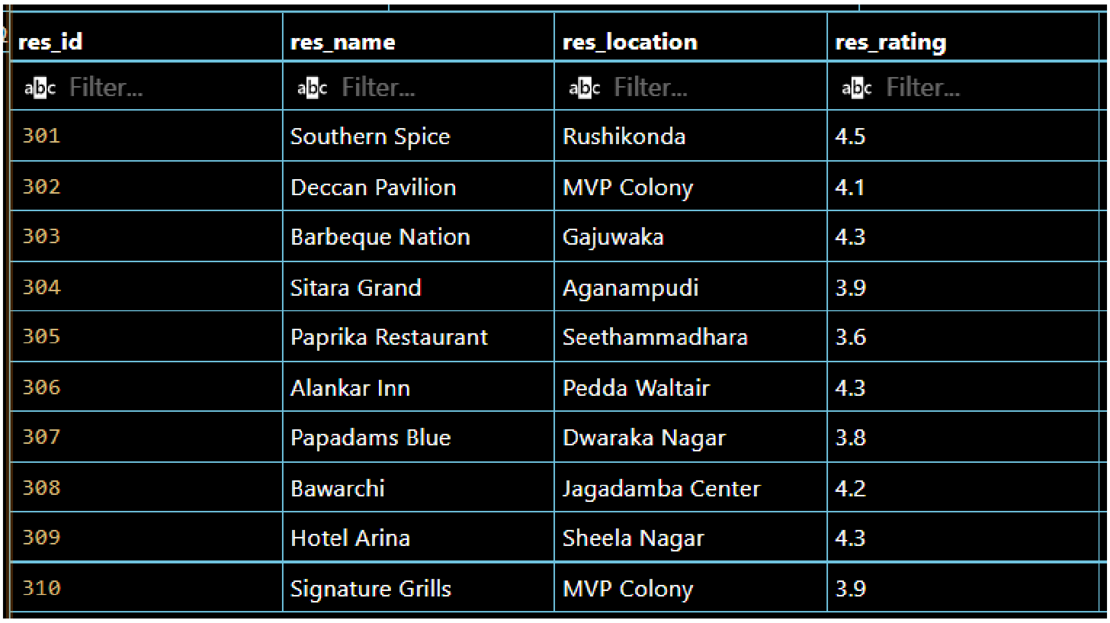
TABLES :

Customer Table :



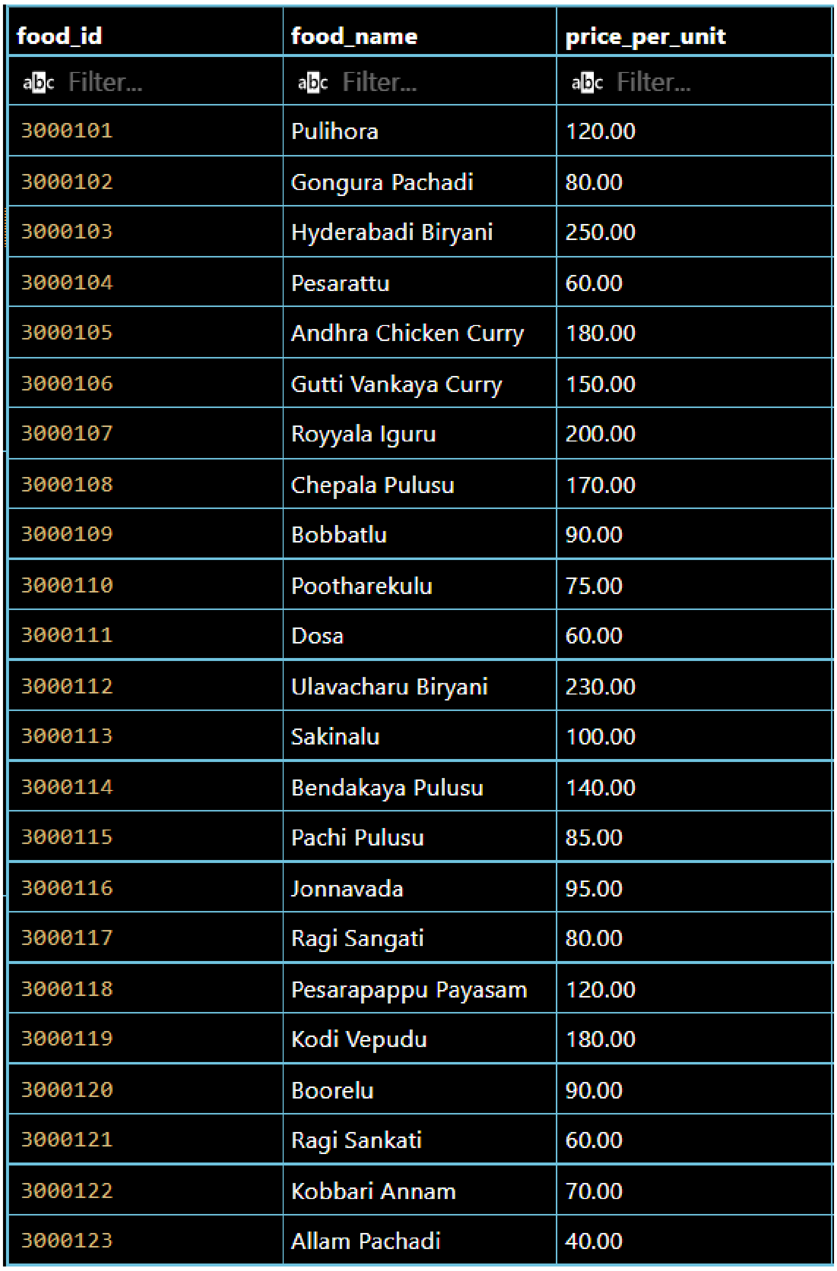
customer_id	customer_name	customer_contact_number	customer_address
Filter...	Filter...	Filter...	Filter...
10001	Gayathri Tangirala	9876543001	16-8,Gajuwaka, Vizag
10002	Veena Madhuri	9123456789	2-18,Rushikonda,Vizag
10003	Sriya Nikhitha	9876543013	6-19-3, Pedda Waltair, Vizag
10004	Koushika Moka	9876543048	50-50-4, Seethammadhara,Vizag
10005	Namratha Kumar	9876543059	38-7-16,MVP Colony,Vizag
10006	Sandhya Singh	9876543077	16-8,Aganampudi,Vizag
10007	Ayesha Mukherjee	9876543033	7-16,Kurmanapalem,Vizag
10008	Sriram Raghavan	9876543096	8-18, Lalitha Nagar,Vizag
10009	Sahith Varma	9876543044	9-17, Madhurawada,Vizag
10010	Siddharth Kumar	9876543075	13-13-8, Maddilapalem,Vizag
10011	Krishna Prabhu	9876543066	17-6,Dwaraka Nagar,Vizag
10012	Harsha Vardhan	9876543099	23-24-15, NAD,Vizag
10013	Praveen Kumar	9836581374	43-25-7,Hanumanthavaka,Vizag
10014	Ram Kishan	8763574583	74-8-16,Sheela Nagar, Vizag
10015	Abhiram Sharma	9561342184	22-34,Bhimili, Vizag
10016	Padma Sowmya	9848521545	55-09, Jagadamba Center, Vizag
10017	Ravi Kiran	9898678148	09-87, Kommadi Junction, Vizag

Restaurant Table :

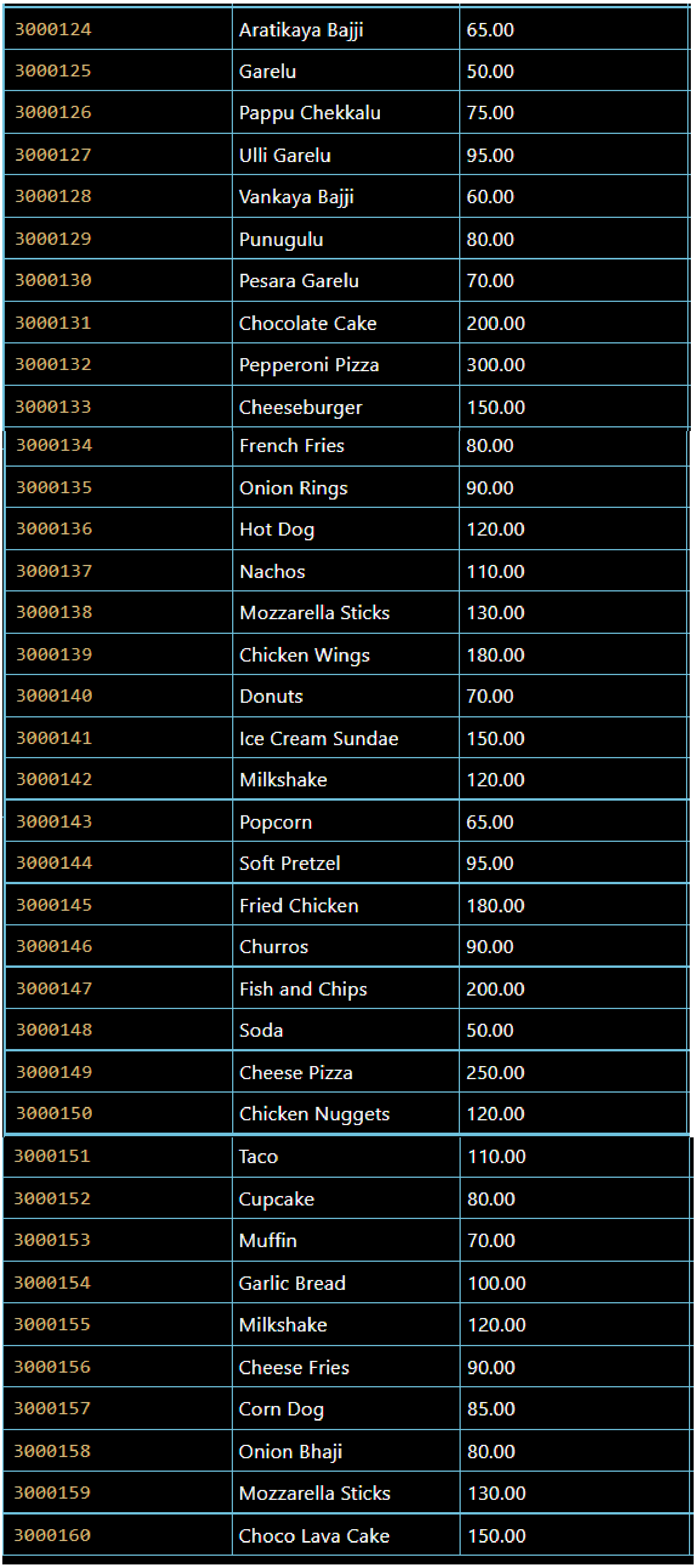


res_id	res_name	res_location	res_rating
abc Filter...	abc Filter...	abc Filter...	abc Filter...
301	Southern Spice	Rushikonda	4.5
302	Deccan Pavilion	MVP Colony	4.1
303	Barbeque Nation	Gajuwaka	4.3
304	Sitara Grand	Aganampudi	3.9
305	Paprika Restaurant	Seethammadhara	3.6
306	Alankar Inn	Pedda Waltair	4.3
307	Papadams Blue	Dwaraka Nagar	3.8
308	Bawarchi	Jagadamba Center	4.2
309	Hotel Arina	Sheela Nagar	4.3
310	Signature Grills	MVP Colony	3.9

Foods Table :

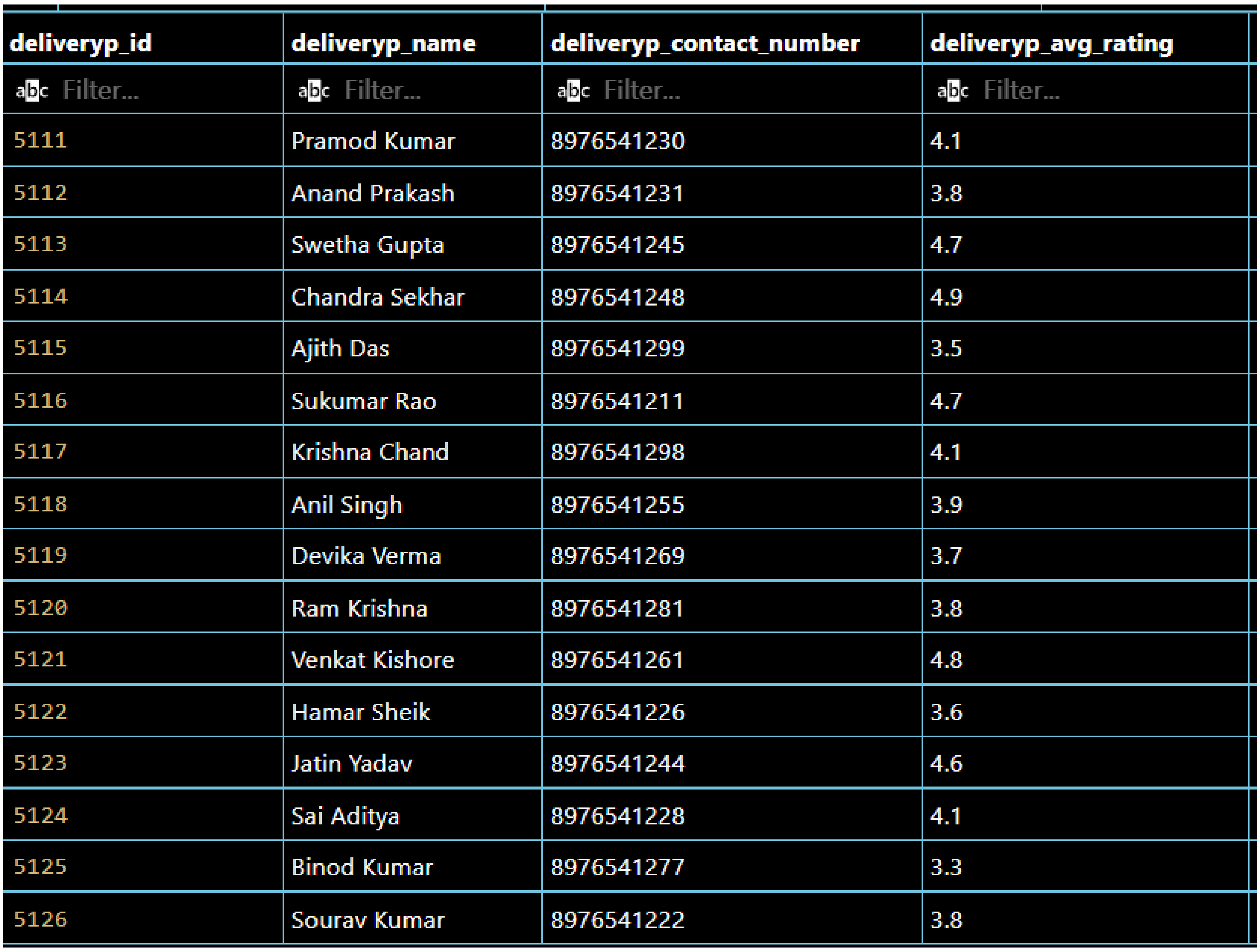


food_id	food_name	price_per_unit
abc Filter...	abc Filter...	abc Filter...
3000101	Pulihora	120.00
3000102	Gongura Pachadi	80.00
3000103	Hyderabadi Biryani	250.00
3000104	Pesarattu	60.00
3000105	Andhra Chicken Curry	180.00
3000106	Gutti Vankaya Curry	150.00
3000107	Royyala Iguru	200.00
3000108	Chepala Pulusu	170.00
3000109	Bobbatlu	90.00
3000110	Pootharekulu	75.00
3000111	Dosa	60.00
3000112	Ulavacharu Biryani	230.00
3000113	Sakinalu	100.00
3000114	Bendakaya Pulusu	140.00
3000115	Pachi Pulusu	85.00
3000116	Jonnavada	95.00
3000117	Ragi Sangati	80.00
3000118	Pesarapappu Payasam	120.00
3000119	Kodi Vepudu	180.00
3000120	Boorelu	90.00
3000121	Ragi Sankati	60.00
3000122	Kobbari Annam	70.00
3000123	Allam Pachadi	40.00



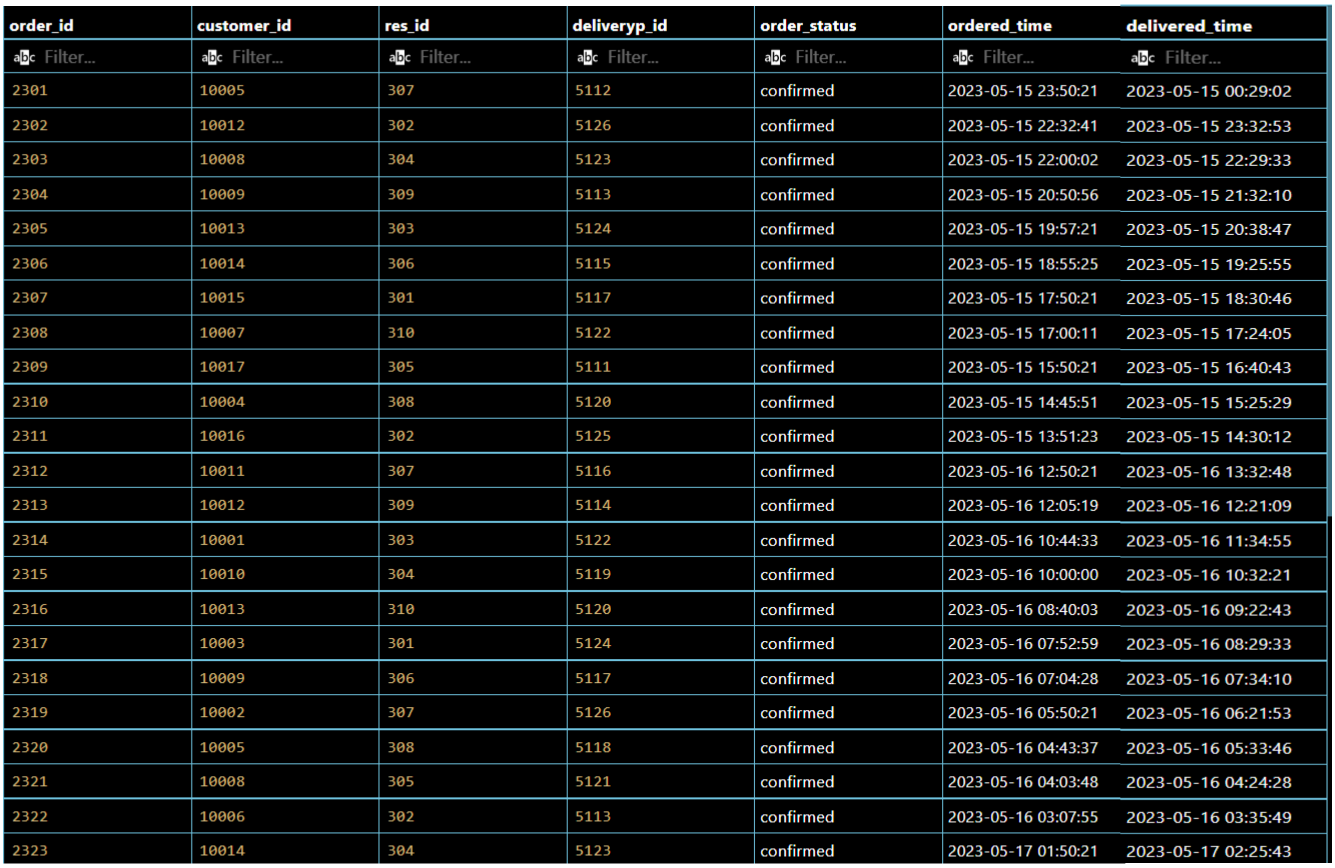
3000124	Aratikaya Bajji	65.00
3000125	Garelu	50.00
3000126	Pappu Chekkalu	75.00
3000127	Ulli Garelu	95.00
3000128	Vankaya Bajji	60.00
3000129	Punugulu	80.00
3000130	Pesara Garelu	70.00
3000131	Chocolate Cake	200.00
3000132	Pepperoni Pizza	300.00
3000133	Cheeseburger	150.00
3000134	French Fries	80.00
3000135	Onion Rings	90.00
3000136	Hot Dog	120.00
3000137	Nachos	110.00
3000138	Mozzarella Sticks	130.00
3000139	Chicken Wings	180.00
3000140	Donuts	70.00
3000141	Ice Cream Sundae	150.00
3000142	Milkshake	120.00
3000143	Popcorn	65.00
3000144	Soft Pretzel	95.00
3000145	Fried Chicken	180.00
3000146	Churros	90.00
3000147	Fish and Chips	200.00
3000148	Soda	50.00
3000149	Cheese Pizza	250.00
3000150	Chicken Nuggets	120.00
3000151	Taco	110.00
3000152	Cupcake	80.00
3000153	Muffin	70.00
3000154	Garlic Bread	100.00
3000155	Milkshake	120.00
3000156	Cheese Fries	90.00
3000157	Corn Dog	85.00
3000158	Onion Bhaji	80.00
3000159	Mozzarella Sticks	130.00
3000160	Choco Lava Cake	150.00

Delivery Partner Table :

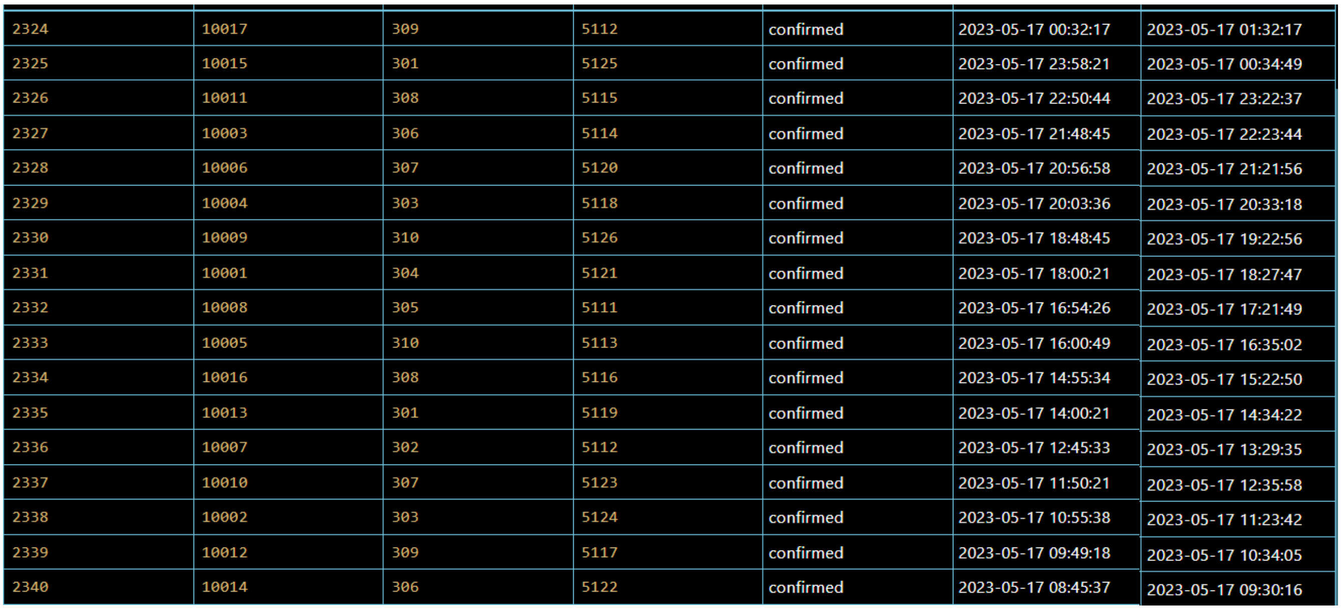


deliveryp_id	deliveryp_name	deliveryp_contact_number	deliveryp_avg_rating
Filter...	Filter...	Filter...	Filter...
5111	Pramod Kumar	8976541230	4.1
5112	Anand Prakash	8976541231	3.8
5113	Swetha Gupta	8976541245	4.7
5114	Chandra Sekhar	8976541248	4.9
5115	Ajith Das	8976541299	3.5
5116	Sukumar Rao	8976541211	4.7
5117	Krishna Chand	8976541298	4.1
5118	Anil Singh	8976541255	3.9
5119	Devika Verma	8976541269	3.7
5120	Ram Krishna	8976541281	3.8
5121	Venkat Kishore	8976541261	4.8
5122	Hamar Sheik	8976541226	3.6
5123	Jatin Yadav	8976541244	4.6
5124	Sai Aditya	8976541228	4.1
5125	Binod Kumar	8976541277	3.3
5126	Sourav Kumar	8976541222	3.8

Order Details Table :

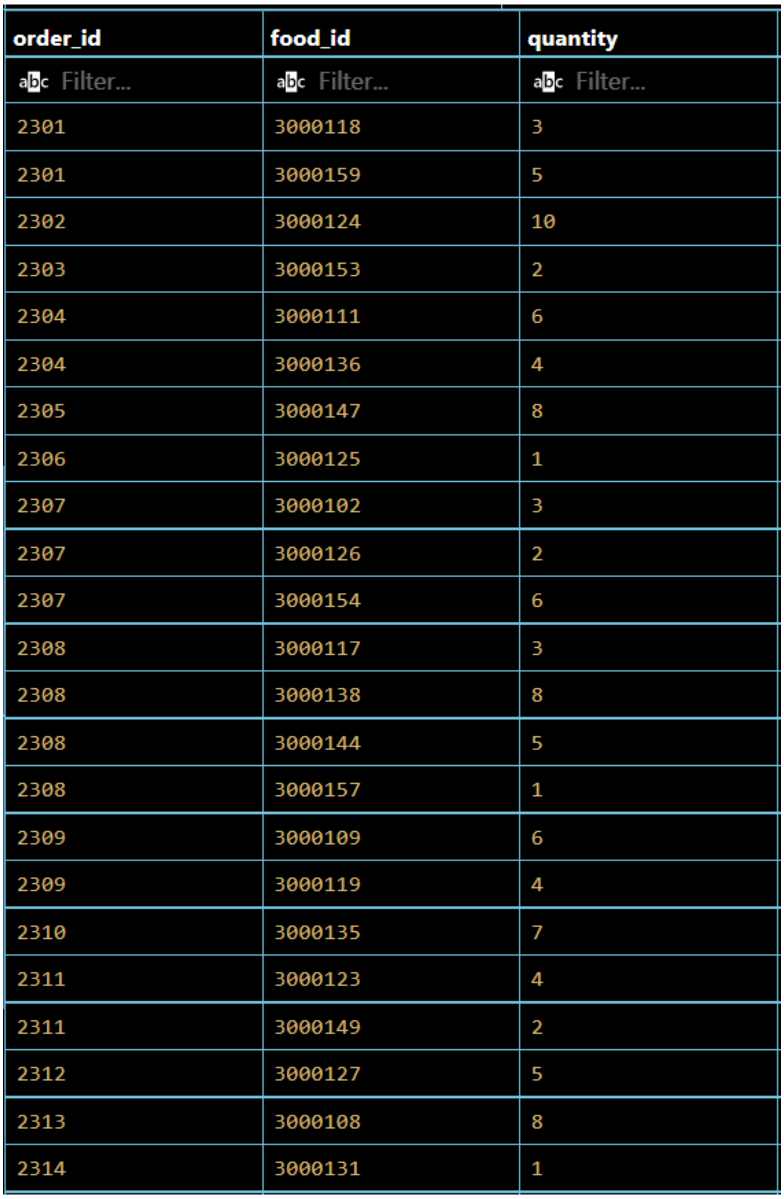


order_id	customer_id	res_id	deliveryp_id	order_status	ordered_time	delivered_time
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
2301	10005	307	5112	confirmed	2023-05-15 23:50:21	2023-05-15 00:29:02
2302	10012	302	5126	confirmed	2023-05-15 22:32:41	2023-05-15 23:32:53
2303	10008	304	5123	confirmed	2023-05-15 22:00:02	2023-05-15 22:29:33
2304	10009	309	5113	confirmed	2023-05-15 20:50:56	2023-05-15 21:32:10
2305	10013	303	5124	confirmed	2023-05-15 19:57:21	2023-05-15 20:38:47
2306	10014	306	5115	confirmed	2023-05-15 18:55:25	2023-05-15 19:25:55
2307	10015	301	5117	confirmed	2023-05-15 17:50:21	2023-05-15 18:30:46
2308	10007	310	5122	confirmed	2023-05-15 17:00:11	2023-05-15 17:24:05
2309	10017	305	5111	confirmed	2023-05-15 15:50:21	2023-05-15 16:40:43
2310	10004	308	5120	confirmed	2023-05-15 14:45:51	2023-05-15 15:25:29
2311	10016	302	5125	confirmed	2023-05-15 13:51:23	2023-05-15 14:30:12
2312	10011	307	5116	confirmed	2023-05-16 12:50:21	2023-05-16 13:32:48
2313	10012	309	5114	confirmed	2023-05-16 12:05:19	2023-05-16 12:21:09
2314	10001	303	5122	confirmed	2023-05-16 10:44:33	2023-05-16 11:34:55
2315	10010	304	5119	confirmed	2023-05-16 10:00:00	2023-05-16 10:32:21
2316	10013	310	5120	confirmed	2023-05-16 08:40:03	2023-05-16 09:22:43
2317	10003	301	5124	confirmed	2023-05-16 07:52:59	2023-05-16 08:29:33
2318	10009	306	5117	confirmed	2023-05-16 07:04:28	2023-05-16 07:34:10
2319	10002	307	5126	confirmed	2023-05-16 05:50:21	2023-05-16 06:21:53
2320	10005	308	5118	confirmed	2023-05-16 04:43:37	2023-05-16 05:33:46
2321	10008	305	5121	confirmed	2023-05-16 04:03:48	2023-05-16 04:24:28
2322	10006	302	5113	confirmed	2023-05-16 03:07:55	2023-05-16 03:35:49
2323	10014	304	5123	confirmed	2023-05-17 01:50:21	2023-05-17 02:25:43

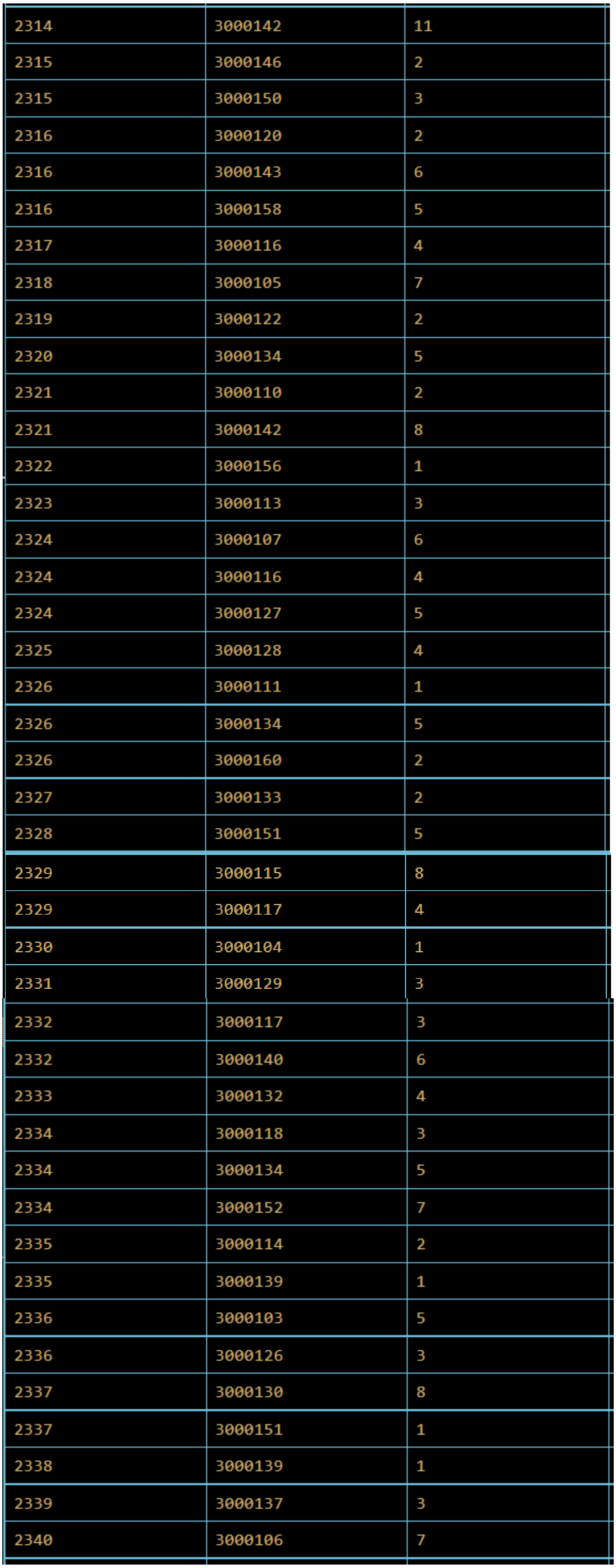


2324	10017	309	5112	confirmed	2023-05-17 00:32:17	2023-05-17 01:32:17
2325	10015	301	5125	confirmed	2023-05-17 23:58:21	2023-05-17 00:34:49
2326	10011	308	5115	confirmed	2023-05-17 22:50:44	2023-05-17 23:22:37
2327	10003	306	5114	confirmed	2023-05-17 21:48:45	2023-05-17 22:23:44
2328	10006	307	5120	confirmed	2023-05-17 20:56:58	2023-05-17 21:21:56
2329	10004	303	5118	confirmed	2023-05-17 20:03:36	2023-05-17 20:33:18
2330	10009	310	5126	confirmed	2023-05-17 18:48:45	2023-05-17 19:22:56
2331	10001	304	5121	confirmed	2023-05-17 18:00:21	2023-05-17 18:27:47
2332	10008	305	5111	confirmed	2023-05-17 16:54:26	2023-05-17 17:21:49
2333	10005	310	5113	confirmed	2023-05-17 16:00:49	2023-05-17 16:35:02
2334	10016	308	5116	confirmed	2023-05-17 14:55:34	2023-05-17 15:22:50
2335	10013	301	5119	confirmed	2023-05-17 14:00:21	2023-05-17 14:34:22
2336	10007	302	5112	confirmed	2023-05-17 12:45:33	2023-05-17 13:29:35
2337	10010	307	5123	confirmed	2023-05-17 11:50:21	2023-05-17 12:35:58
2338	10002	303	5124	confirmed	2023-05-17 10:55:38	2023-05-17 11:23:42
2339	10012	309	5117	confirmed	2023-05-17 09:49:18	2023-05-17 10:34:05
2340	10014	306	5122	confirmed	2023-05-17 08:45:37	2023-05-17 09:30:16

Order Food Table :

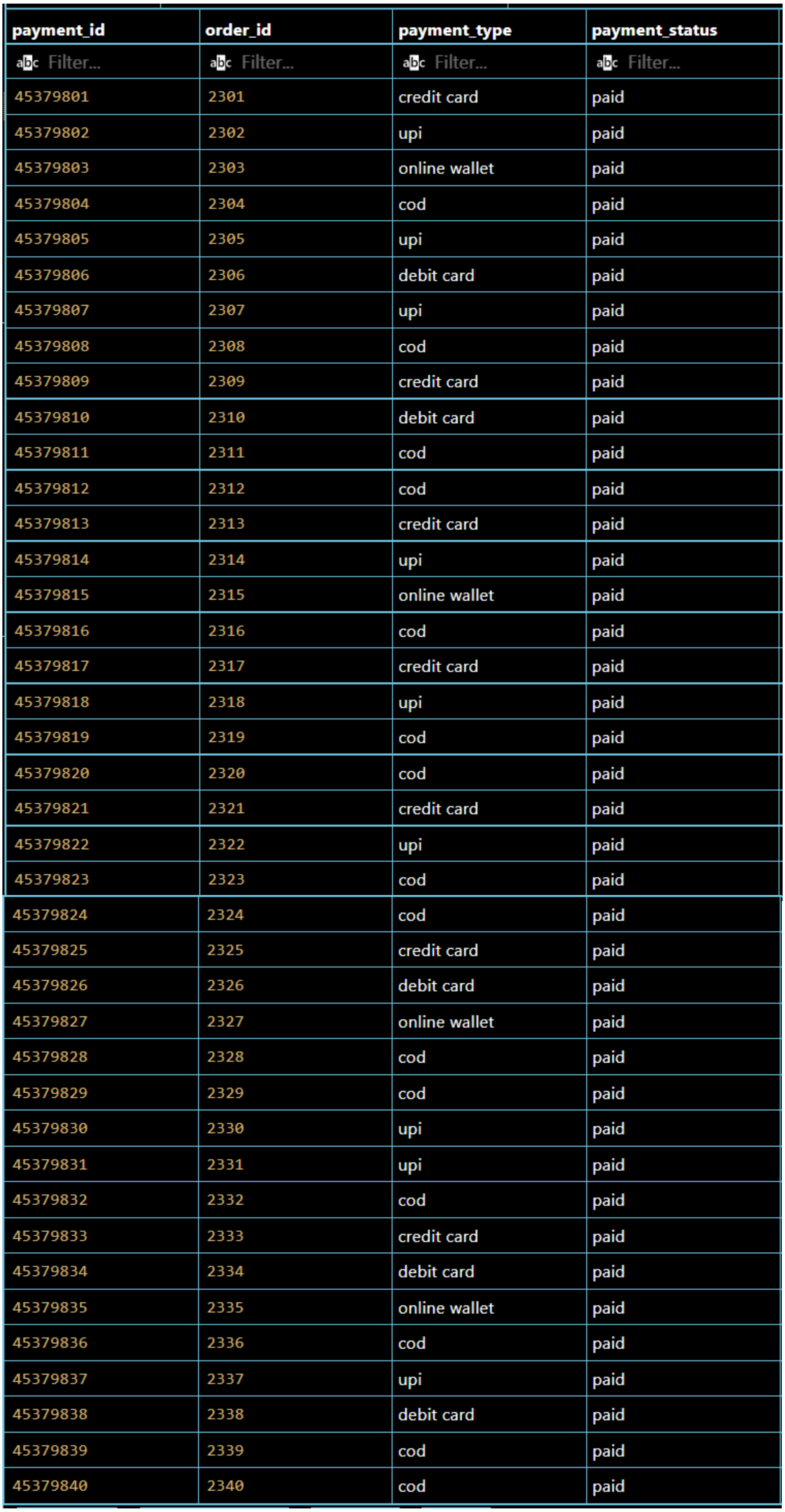


order_id	food_id	quantity
abc Filter...	abc Filter...	abc Filter...
2301	3000118	3
2301	3000159	5
2302	3000124	10
2303	3000153	2
2304	3000111	6
2304	3000136	4
2305	3000147	8
2306	3000125	1
2307	3000102	3
2307	3000126	2
2307	3000154	6
2308	3000117	3
2308	3000138	8
2308	3000144	5
2308	3000157	1
2309	3000109	6
2309	3000119	4
2310	3000135	7
2311	3000123	4
2311	3000149	2
2312	3000127	5
2313	3000108	8
2314	3000131	1



2314	3000142	11
2315	3000146	2
2315	3000150	3
2316	3000120	2
2316	3000143	6
2316	3000158	5
2317	3000116	4
2318	3000105	7
2319	3000122	2
2320	3000134	5
2321	3000110	2
2321	3000142	8
2322	3000156	1
2323	3000113	3
2324	3000107	6
2324	3000116	4
2324	3000127	5
2325	3000128	4
2326	3000111	1
2326	3000134	5
2326	3000160	2
2327	3000133	2
2328	3000151	5
2329	3000115	8
2329	3000117	4
2330	3000104	1
2331	3000129	3
2332	3000117	3
2332	3000140	6
2333	3000132	4
2334	3000118	3
2334	3000134	5
2334	3000152	7
2335	3000114	2
2335	3000139	1
2336	3000103	5
2336	3000126	3
2337	3000130	8
2337	3000151	1
2338	3000139	1
2339	3000137	3
2340	3000106	7

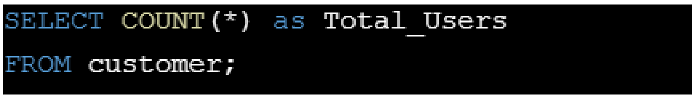
Payment Table :



payment_id	order_id	payment_type	payment_status
45379801	2301	credit card	paid
45379802	2302	upi	paid
45379803	2303	online wallet	paid
45379804	2304	cod	paid
45379805	2305	upi	paid
45379806	2306	debit card	paid
45379807	2307	upi	paid
45379808	2308	cod	paid
45379809	2309	credit card	paid
45379810	2310	debit card	paid
45379811	2311	cod	paid
45379812	2312	cod	paid
45379813	2313	credit card	paid
45379814	2314	upi	paid
45379815	2315	online wallet	paid
45379816	2316	cod	paid
45379817	2317	credit card	paid
45379818	2318	upi	paid
45379819	2319	cod	paid
45379820	2320	cod	paid
45379821	2321	credit card	paid
45379822	2322	upi	paid
45379823	2323	cod	paid
45379824	2324	cod	paid
45379825	2325	credit card	paid
45379826	2326	debit card	paid
45379827	2327	online wallet	paid
45379828	2328	cod	paid
45379829	2329	cod	paid
45379830	2330	upi	paid
45379831	2331	upi	paid
45379832	2332	cod	paid
45379833	2333	credit card	paid
45379834	2334	debit card	paid
45379835	2335	online wallet	paid
45379836	2336	cod	paid
45379837	2337	upi	paid
45379838	2338	debit card	paid
45379839	2339	cod	paid
45379840	2340	cod	paid

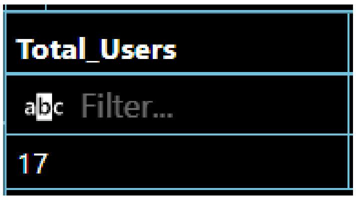
QUERIES :

- Write a SQL query to find the total number of users :



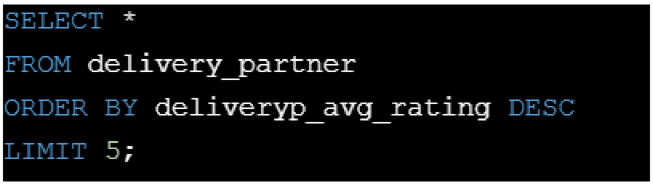
```
SELECT COUNT(*) as Total_Users
FROM customer;
```

Output:



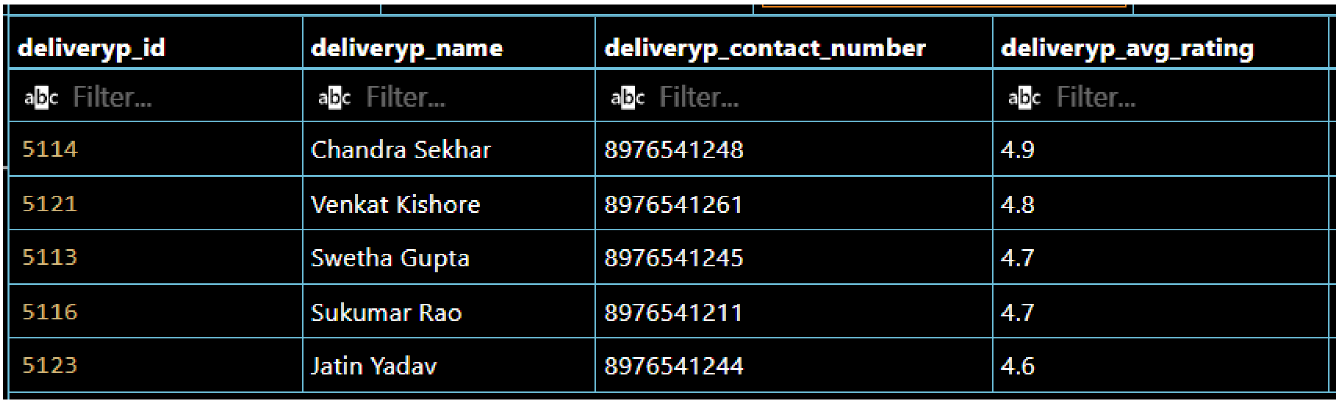
Total_Users
Filter...
17

- Write a SQL query to find the top 5 delivery partners based on the ratings given by customers :



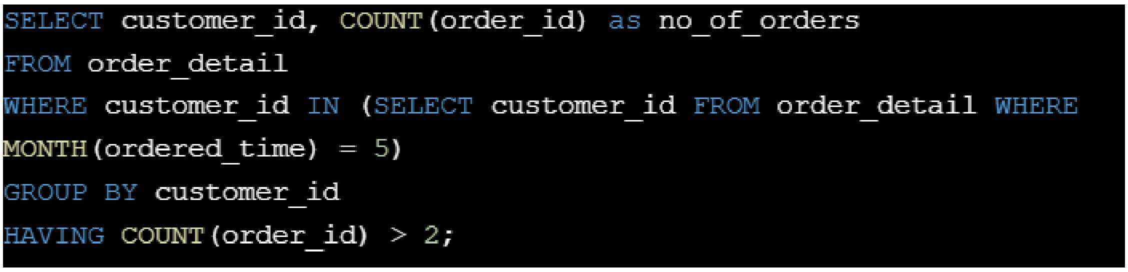
```
SELECT *
FROM delivery_partner
ORDER BY deliveryp_avg_rating DESC
LIMIT 5;
```

Output:



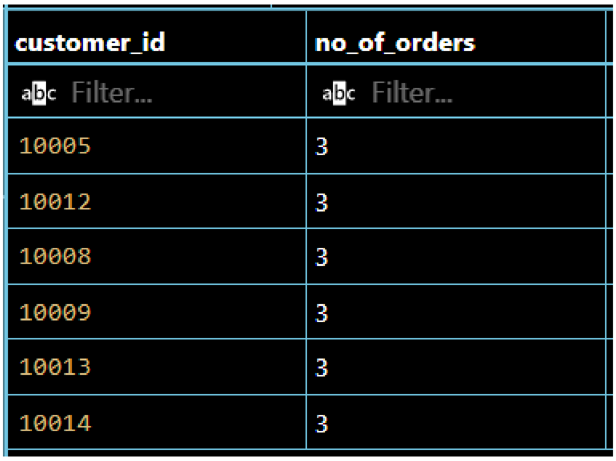
deliveryp_id	deliveryp_name	deliveryp_contact_number	deliveryp_avg_rating
Filter...	Filter...	Filter...	Filter...
5114	Chandra Sekhar	8976541248	4.9
5121	Venkat Kishore	8976541261	4.8
5113	Swetha Gupta	8976541245	4.7
5116	Sukumar Rao	8976541211	4.7
5123	Jatin Yadav	8976541244	4.6

- Write a SQL query to find the list of users who made more than 2 orders in a May month :



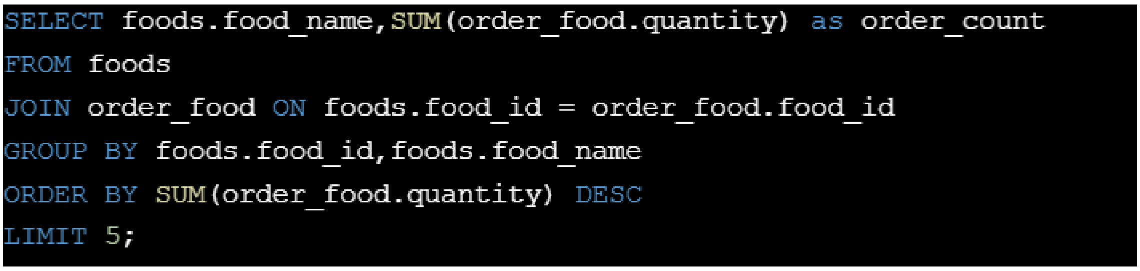
```
SELECT customer_id, COUNT(order_id) as no_of_orders
FROM order_detail
WHERE customer_id IN (SELECT customer_id FROM order_detail WHERE
MONTH(ordered_time) = 5)
GROUP BY customer_id
HAVING COUNT(order_id) > 2;
```

Output:



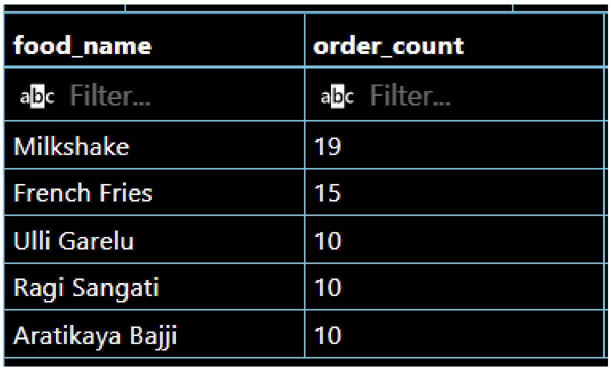
customer_id	no_of_orders
abc Filter...	abc Filter...
10005	3
10012	3
10008	3
10009	3
10013	3
10014	3

- Write an SQL query to find the top 5 food items based on the quantity placed :



```
SELECT foods.food_name,SUM(order_food.quantity) as order_count
FROM foods
JOIN order_food ON foods.food_id = order_food.food_id
GROUP BY foods.food_id,foods.food_name
ORDER BY SUM(order_food.quantity) DESC
LIMIT 5;
```

Output:



food_name	order_count
abc Filter...	abc Filter...
Milkshake	19
French Fries	15
Ulli Garelu	10
Ragi Sangati	10
Aratikaya Bajji	10