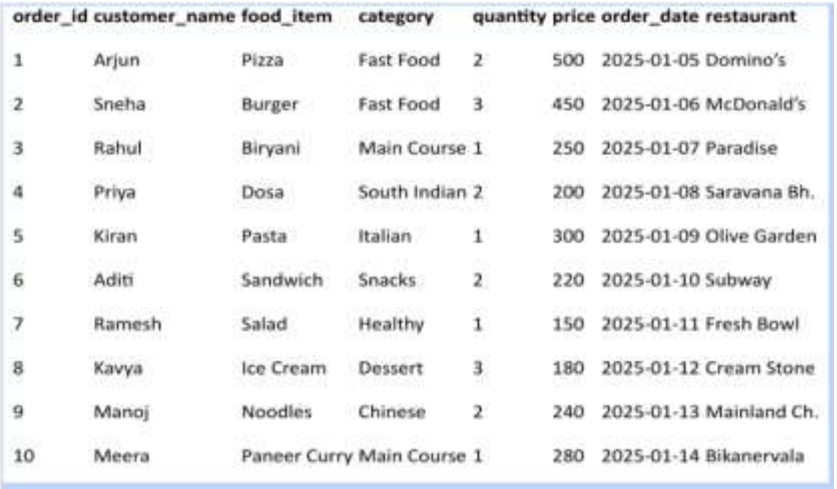

order_id	customer_name	food_item	category	quantity	price	order_date	restaurant
1	Arjun	Pizza	Fast Food	2	500	2025-01-05	Domino’s
2	Sneha	Burger	Fast Food	3	450	2025-01-06	McDonald’s
3	Rahul	Biryani	Main Course	1	250	2025-01-07	Paradise
4	Priya	Dosa	South Indian	2	200	2025-01-08	Saravana Bh.
5	Kiran	Pasta	Italian	1	300	2025-01-09	Olive Garden
6	Aditi	Sandwich	Snacks	2	220	2025-01-10	Subway
7	Ramesh	Salad	Healthy	1	150	2025-01-11	Fresh Bowl
8	Kavya	Ice Cream	Dessert	3	180	2025-01-12	Cream Stone
9	Manoj	Noodles	Chinese	2	240	2025-01-13	Mainland Ch.
10	Meera	Paneer Curry	Main Course	1	280	2025-01-14	Bikanervala

GROUP BY & Aggregates

1. Total sales amount for each restaurant

```
SELECT restaurant, SUM(price) AS total_sales

FROM FoodOrders

GROUP BY restaurant;
```

2. Average price of food items per category

```
SELECT category, AVG(price) AS avg_price

FROM FoodOrders

GROUP BY category;
```

3. Count how many orders were placed for each food category

```
SELECT category, COUNT(*) AS total_orders

FROM FoodOrders

GROUP BY category;
```

4. Maximum quantity ordered for each food item

```
SELECT food_item, MAX(quantity) AS max_quantity

FROM FoodOrders
```

```sql
GROUP BY food_item;
```

5. Total amount spent by each customer

```sql
SELECT customer_name, SUM(price) AS total_spent
FROM FoodOrders
GROUP BY customer_name;
```

HAVING

6. Restaurants with total sales greater than 800

```sql
SELECT restaurant, SUM(price) AS total_sales
FROM FoodOrders
GROUP BY restaurant
HAVING SUM(price) > 800;
```

7. Customers who spent more than 500 in total

```sql
SELECT customer_name, SUM(price) AS total_spent
FROM FoodOrders
GROUP BY customer_name
HAVING SUM(price) > 500;
```

```sql
-- 8. Food categories where the average item price > 250
SELECT category, AVG(price) AS avg_price
FROM FoodOrders
GROUP BY category
HAVING AVG(price) > 250;
```

```sql
-- 9. Restaurants with more than 2 orders
SELECT restaurant, COUNT(*) AS total_orders
```

```
FROM FoodOrders
GROUP BY restaurant
HAVING COUNT(*) > 2;

-- 10. Categories where total quantity ordered > 4
SELECT category, SUM(quantity) AS total_quantity
FROM FoodOrders
GROUP BY category
HAVING SUM(quantity) > 4;
```

ORDER BY

```
-- 11. All orders by price descending
SELECT *
FROM FoodOrders
ORDER BY price DESC;

-- 12. Customers ordered by their total spending
SELECT customer_name, SUM(price) AS total_spent
FROM FoodOrders
GROUP BY customer_name
ORDER BY total_spent DESC;

-- 13. Restaurants ordered by total quantity sold
SELECT restaurant, SUM(quantity) AS total_quantity
FROM FoodOrders
GROUP BY restaurant
ORDER BY total_quantity DESC;
```

```
-- 14. Top 3 highest-priced food items
SELECT food_item, price
FROM FoodOrders
ORDER BY price DESC
LIMIT 3;

-- 15. Orders sorted by order_date (latest first)
SELECT *
FROM FoodOrders
ORDER BY order_date DESC;
```

LIMIT & OFFSET

```
-- 16. First 5 orders
SELECT *
FROM FoodOrders
LIMIT 5;

-- 17. Top 3 most expensive orders
SELECT *
FROM FoodOrders
ORDER BY price DESC
LIMIT 3;

-- 18. Skip first 3 rows, show next 4
SELECT *
FROM FoodOrders
LIMIT 4 OFFSET 3;
```

```
-- 19. Second highest-priced food item
SELECT *
FROM FoodOrders
ORDER BY price DESC
LIMIT 1 OFFSET 1;

-- 20. Top 2 customers by total spending
SELECT customer_name, SUM(price) AS total_spent
FROM FoodOrders
GROUP BY customer_name
ORDER BY total_spent DESC
LIMIT 2;
```

Combined Questions

```
-- 1. Total spending by each customer (> ₹500), ordered by spending descending
SELECT customer_name, SUM(price) AS total_spent
FROM FoodOrders
GROUP BY customer_name
HAVING SUM(price) > 500
ORDER BY total_spent DESC;

-- 2. Top 3 food categories with highest total sales amount
SELECT category, SUM(price) AS total_sales
FROM FoodOrders
GROUP BY category
ORDER BY total_sales DESC
LIMIT 3;
```

```sql
-- 3. Restaurants where average price > 250, ordered by average price descending
SELECT restaurant, AVG(price) AS avg_price
FROM FoodOrders
GROUP BY restaurant
HAVING AVG(price) > 250
ORDER BY avg_price DESC;

-- 4. Top 2 customers who placed the highest quantity of food items
SELECT customer_name, SUM(quantity) AS total_quantity
FROM FoodOrders
GROUP BY customer_name
ORDER BY total_quantity DESC
LIMIT 2;

-- 5. Restaurant with 2nd highest total sales
SELECT restaurant, SUM(price) AS total_sales
FROM FoodOrders
GROUP BY restaurant
ORDER BY total_sales DESC
LIMIT 1 OFFSET 1;

-- 6. Categories with more than 2 total orders, ordered by total quantity sold
SELECT category, SUM(quantity) AS total_quantity, COUNT(*) AS total_orders
FROM FoodOrders
GROUP BY category
HAVING COUNT(*) > 2
ORDER BY total_quantity DESC;
```

```sql
-- 7. Top 3 food items by total sales amount
SELECT food_item, SUM(price) AS total_sales
FROM FoodOrders
GROUP BY food_item
ORDER BY total_sales DESC
LIMIT 3;

-- 8. Customers who ordered more than 2 different categories
SELECT customer_name, COUNT(DISTINCT category) AS category_count
FROM FoodOrders
GROUP BY customer_name
HAVING COUNT(DISTINCT category) > 2
ORDER BY category_count DESC;
```