FOOD MANAGEMENT SYSTEM

Database Design Document

V 2.0

By

1	*ZUNAIRA AKBAR*	*NUM-BSCS-2022-34*
2	*DANISH ABDULLAH KHAN*	*NUM-BSCS-2022-05*
3	*ABUBAKAR*	*NUM-BSCS-2022-41*

Department of Computer Sciences

Namal University

Mianwali, Pakistan

Submission Date: 10th June, 2024

REVISION HISTORY

Date	Version	Description	Approved by
10/6/24	*V.2.0*	*Change the ERD* *Give the attributes datatypes according to their nature* *Define primary keys and foreign keys* *Define the Relations between the tables*	
22/04/24	*V 1.0*	*Specify the changes implemented subsequent to the submission of the previous document. These changes should be based on the suggestions given by the person who approved the document.*	

Instructions:

- *Place the latest revisions at the top of the table.*
- *The Revision History pertains only to changes in the document's content or any updates made after a suggestion from the approving authority. It does not apply to the template's formatting.*

TABLE OF CONTENTS

CHAPTER 1: PROJECT OVERVIEW

1.1. INTRODUCTION:

The "Food Management System" is a project that aims to simplify the handling of food items in places like restaurants or food banks. It helps in keeping track of food stock, including when items go bad and how they are used. This is important because the traditional methods of managing food can be slow, prone to mistakes, and result in food wastage. With this system, everyone involved, from the kitchen staff to the managers and suppliers, can effectively manage food items, reduce waste, and optimize resource utilization.and suppliers - can manage food items better, waste less food, and use resources more effectively.

1.2. PROBLEM STATEMENT:

Currently, managing food items can be quite challenging. People often have to manually keep track of everything, there's no central place to store all the information, and it's difficult to know when food will spoil. These problems can lead to errors, food wastage, and operational inefficiencies. Most of the time, people rely on paper records or separate spreadsheets, which can be hard to keep up-to-date and accurate.

The Food Management System aims to address these issues by providing a centralized database where all the information can be stored. This makes it easier to track the available food stock and automatically receive alerts when items are about to expire. Additionally, it enables the generation of detailed reports to keep everyone informed about what's happening.

1.3. PROJECT OBJECTIVES:

Centralized Database Creation:

- Objective: Develop a centralized database to store food inventory data, including item details, quantities, and expiration dates.
- Measurable: Complete the setup of the database within the semester.
- Achievable: Feasible with the available resources.
- Relevance: Directly addresses the need for organized data storage.

Reporting Functionality:

- Objective: Generate reports on food usage, wastage, and inventory levels for data-driven decision-making.
- Measurable: Develop and validate reporting functionalities within the semester.
- Achievable: Aligned with project resources.
- Relevance: Facilitates informed decision-making.

Inventory Management Efficiency Improvement:

- Objective: Enhance overall inventory management processes by providing tools for tracking, ordering, and replenishment.
- Measurable: Achieve improvement in efficiency within the semester.
- Achievable: Realistic given the project context.
- Relevance: Solves identified problems related to food management.

1.4. DOCUMENT OBJECTIVES:

Introduction:

- Purpose: Provide an overview of the project and its significance.
- Content: Briefly explain the need for efficient food inventory management and introduce the objectives of the system.

Problem Statement:

- Purpose: Clearly state the challenges or issues the system aims to address.
- Content: Explain the existing problems related to food inventory tracking, wastage, and data management.

Project Objectives:

- Purpose: Specify the goals of the system.
- Content:
  - Centralized Database Creation: Describe the objective of creating a centralized database for food inventory data.
  - Automated Expiration Date Tracking: Detail the goal of efficient expiration date tracking.
  - Reporting Functionality: Highlight the importance of generating relevant reports.
  - Inventory Management Efficiency Improvement: Discuss the objective of enhancing overall inventory management processes.

Database Schema:

- Purpose: Explain the structure of the database.
- Content: Present the tables, fields, and relationships relevant to food inventory data.

CHAPTER 2: DETAILED DATABASE DESIGN

2.1. ENTITY:

Sr. No	Entity Name	Description
01	Customer	This entity represents the customer who places the order.
02	Order	An order represents a transaction made by a customer.
03	Payment	A payment records the transaction details of an order.
04	Menu	A menu item represents the food or drink options available.
05	MenuType	A menu type categorizes the menu items.
06	OrderDetail	Order detail captures specific items and quantities in an order.
07	Rating	A rating provides customer feedback on menu items.

2.2. DATA DICTIONARY:

Customer:

Sr. No	Name	Data Type	Constraint	Description
01	CustomerID	INT	PK	Unique identifier for each customer.
02	CustomerType	VARCHAR(20)		Type of customer (e.g., regular, VIP).
03	Email	VARCHAR(50)	UNIQUE	Email address of the customer.
04	Phone	VARCHAR(20)		Phone number of the customer.
05	Address	VARCHAR(100)		Address of the customer.

Orders:

Sr. No	Name	Data Type	Constraint	Description
01	OrderID	INT	PK	Unique identifier for each order.
02	CustomerID	INT	FK	Reference to the customer who placed the order.
03	OrderDate	DATE		Date when the order was placed.

Payment:

Sr. No	Name	Data Type	Constraint	Description
01	PaymentID	INT	PK	Unique identifier for each payment.
02	OrderID	INT	FK	Reference to the order for the payment.
03	PaymentAmount	DECIMAL(10,2)		Amount paid.
04	PaymentDate	DATE		Date of the payment.
05	PaymentMethod	VARCHAR(50)		Method used for the payment.

Menu:

Sr. No	Name	Data Type	Constraint	Description
01	MenuItemID	INT	PK	Unique identifier for each menu item.
02	MenuName	VARCHAR(50)		Name of the menu item.
03	Price	DECIMAL(10,2)		Price of the menu item.
04	Description	VARCHAR(255)		Description of the menu item.
05	MenuTypeID	INT	FK	Reference to the type of menu item.

MenuType:

Sr. No	Name	Data Type	Constraint	Description
01	MenuTypeID	INT	PK	Unique identifier for each menu type.
02	TypeName	VARCHAR(50)		Name of the menu type.

OrderDetail:

Sr. No	Name	Data Type	Constraint	Description
01	OrderDetailID	INT	PK	Unique identifier for each order detail.
02	OrderID	INT	FK	Reference to the order.
03	MenuItemID	INT	FK	Reference to the menu item.
04	Quantity	INT		Quantity of the menu item ordered.
05	Price	DECIMAL(10,2)		Price of the ordered item.
06	SpecialInstructions	VARCHAR(255)		Any special instructions for the order.

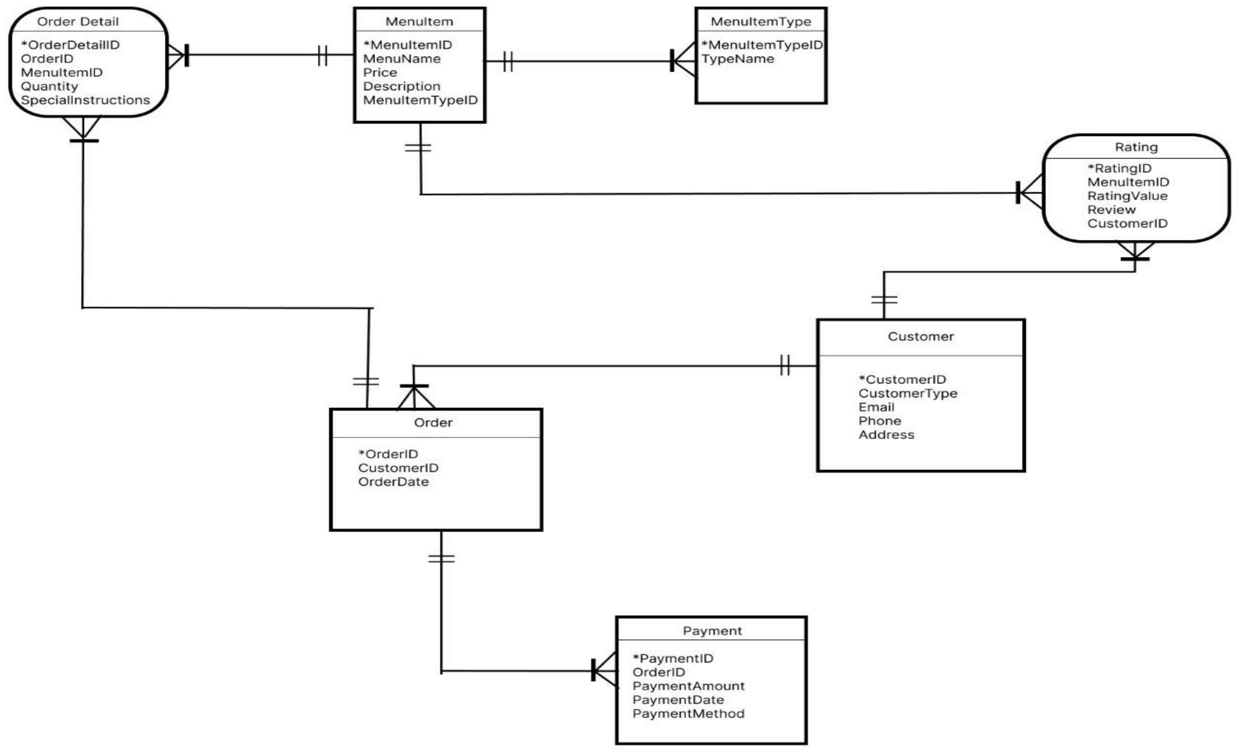
Rating:

Sr. No	Name	Data Type	Constraint	Description
01	RatingID	INT	PK	Unique identifier for each rating.
02	MenuItemID	INT	FK	Reference to the rated menu item.
03	RatingValue	INT		Rating value given by the customer.
04	Review	VARCHAR(255)		Customer review for the menu item.
05	CustomerID	INT	FK	Reference to the customer who gave the rating.

2.3. RELATIONSHIPS:

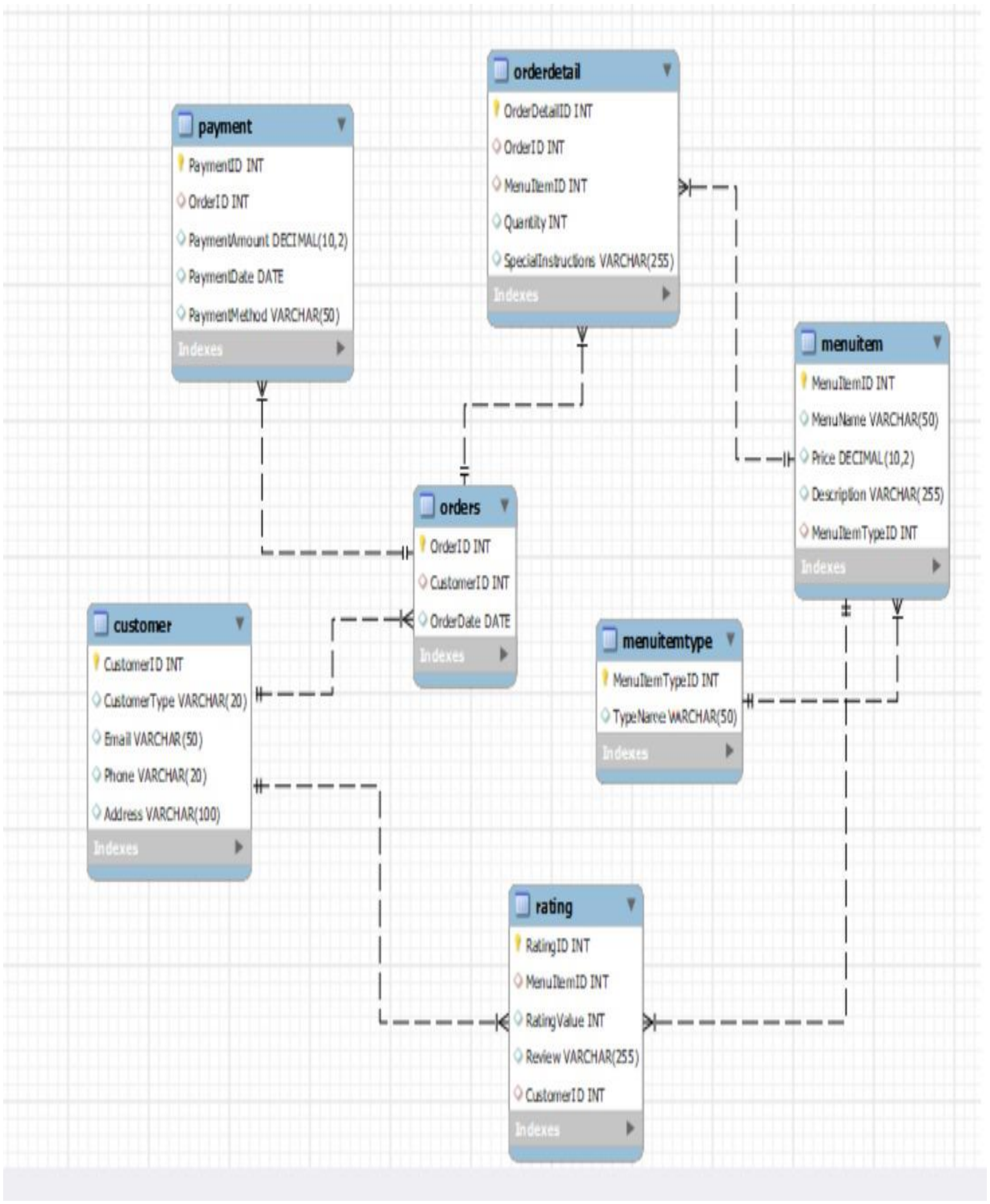
Sr. No	Participating Entities	Relation	Business Rule
01	User, Order	User places Order	A user may place multiple orders. An order is placed by exactly one user.
02	Order, Payment	Order has Payment	An order may have multiple payments. A payment is for exactly one order.
03	Order, OrderDetail	Order has OrderDetail	An order has multiple order details. An order detail belongs to exactly one order.
04	Menu, OrderDetail	Menu is in OrderDetail	A menu item may appear in multiple order details. An order detail references exactly one menu item.
05	Menu, Rating	Menu receives Rating	A menu item may receive multiple ratings. A rating is for exactly one menu item.
06	User, Rating	User gives Rating	A user may give multiple ratings. A rating is given by exactly one user.
07	MenuType, Menu	MenuType categorizes Menu	A menu type may categorize multiple menu items. A menu item belongs to exactly one menu type.

2.4. ENTITY RELATIONSHIP DIAGRAM:



CHAPTER 3 : LOGICAL DATABASE DESIGN

3.1. RELATIONAL SCHEMA:



3.2. FUNCTIONAL DEPENDENCIES:

1. Customer table:

- CustomerID → CustomerType, Email, Phone, Address

Example: If CustomerID is 1, it determines the CustomerType as 'Registered', Email as 'john@example.com', Phone as '123-456-7890', and Address as '123 Main St, Anytown'.

2. MenuType table:

- MenuTypeID → TypeName

Example: If MenuTypeID is 1, it determines the TypeName as 'Pizza'.

3. Menu table:

- MenuItemID → MenuName, Price, Description, MenuTypeID

Example: If MenuItemID is 301, it determines the MenuName as 'Margherita Pizza', Price as 12.99, Description as 'Classic pizza with tomatoes', and MenuTypeID as 1 (belonging to the 'Pizza' menu type).

4. Orders table:

- OrderID → CustomerID, OrderDate

Example: If OrderID is 101, it determines the CustomerID as 1 (the customer who placed the order) and the OrderDate as '2024-06-01'.

5. OrderDetail table:

- OrderDetailID → OrderID, MenuItemID, Quantity, Price, SpecialInstructions

Example: If OrderDetailID is 401, it determines the OrderID as 101 (the order it belongs to), MenuItemID as 301 (the specific menu item ordered), Quantity as 2, Price as 25.98 (the price of the menu item), and SpecialInstructions as 'No onions'.

6. Payment table:

- PaymentID → OrderID, PaymentAmount, PaymentDate, PaymentMethod

Example: If PaymentID is 201, it determines the OrderID as 101 (the order for which the payment was made), PaymentAmount as 50.00, PaymentDate as '2024-06-01', and PaymentMethod as 'Credit Card'.

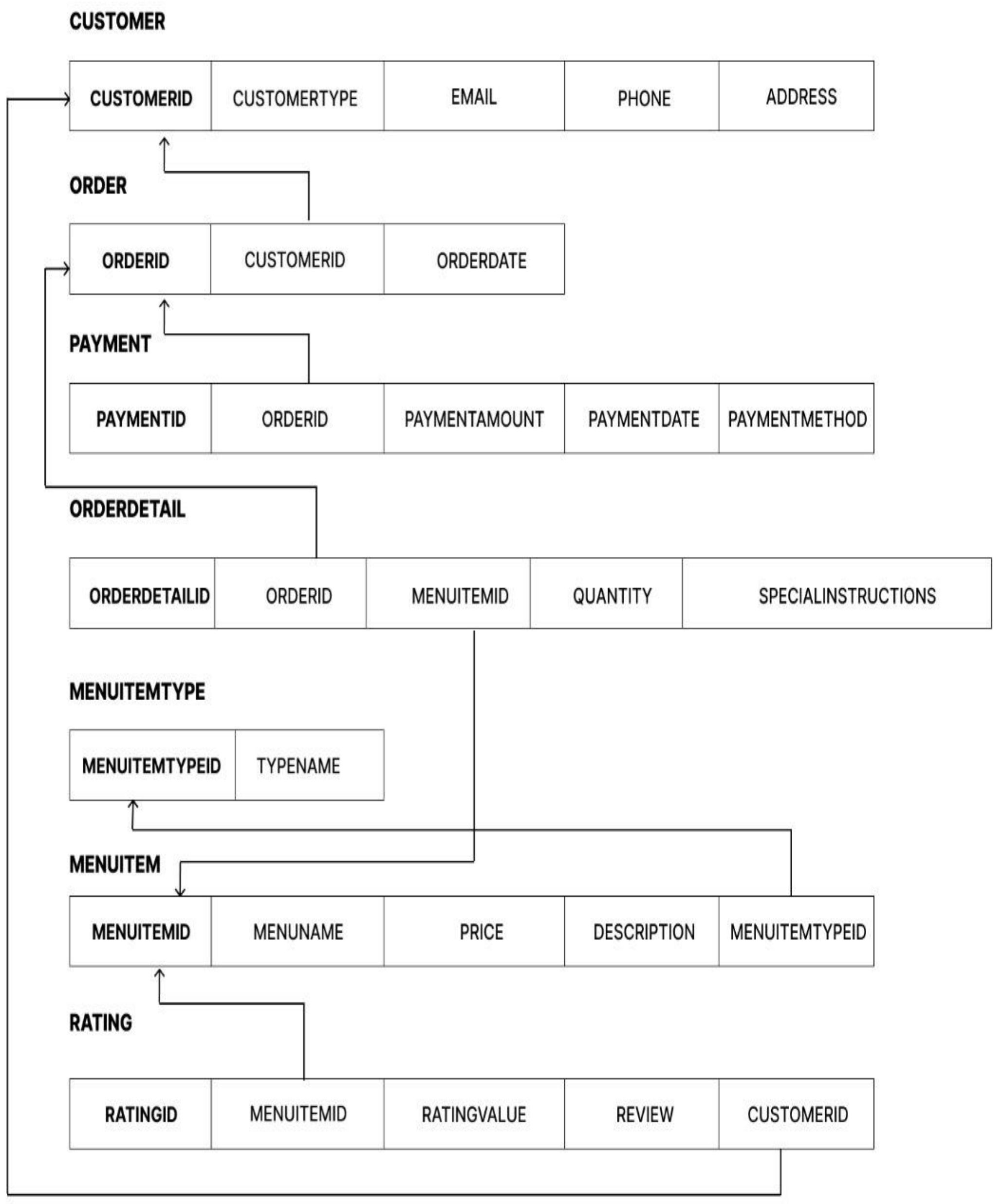
7. Rating table:

- RatingID → MenuItemID, RatingValue, Review, CustomerID

Example: If RatingID is 501, it determines the MenuItemID as 301 (the menu item being rated), RatingValue as 5 (out of 5), Review as 'Excellent taste!', and CustomerID as 1 (the customer who provided the rating).

3.3. NORMALIZATION:

Our ERD doesn't contain any anomaly so we only draw the 3NF Normalization.



CHAPTER 4 : PHYSICAL DATABASE DESIGN

4.1. STRUCTURE OF THE TABLES:

DESCRIBE CUSTOMER;

DESCRIBE ORDERS;

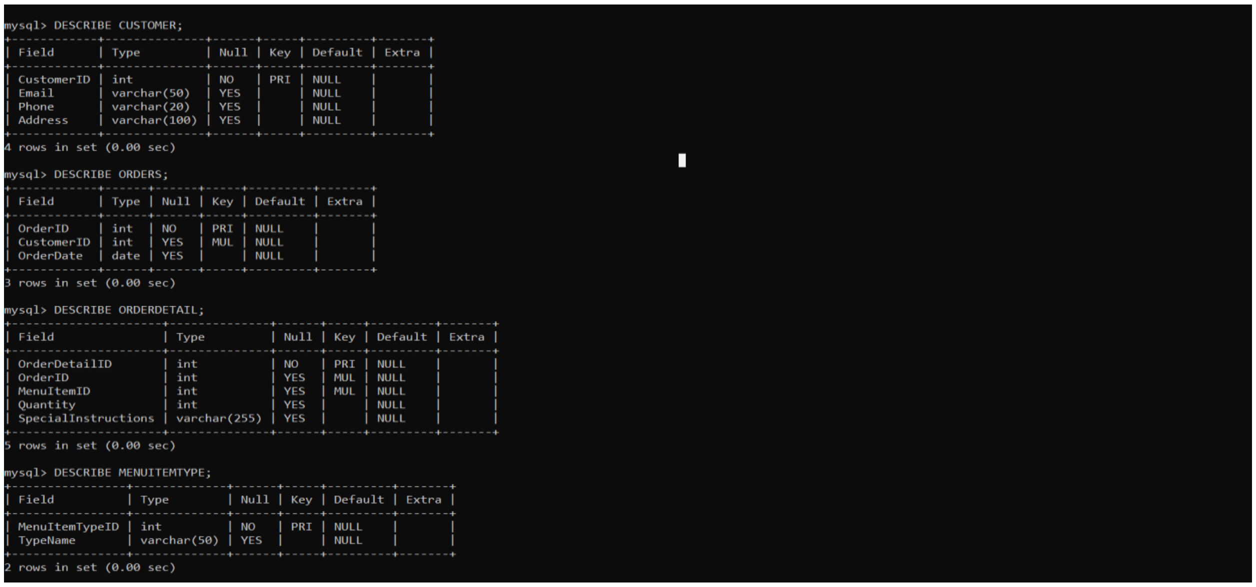
DESCRIBE ORDERDETAIL;

DESCRIBE MENUITEMTYPE;

DESCRIBE MENUITEM;

DESCRIBE PAYMENT;

DESCRIBE RATING;

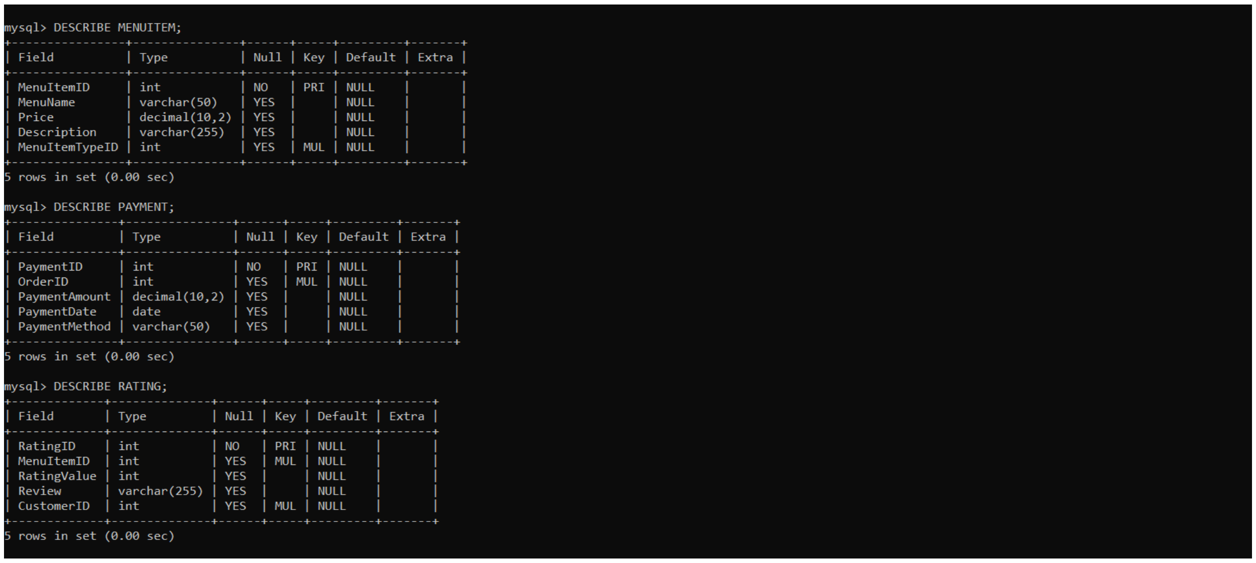


```
mysql> DESCRIBE CUSTOMER;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| CustomerID | int          | NO   | PRI | NULL    |       |
| Email      | varchar(50)  | YES  |     | NULL    |       |
| Phone      | varchar(20)  | YES  |     | NULL    |       |
| Address    | varchar(100) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> DESCRIBE ORDERS;
+------------+------+------+-----+---------+-------+
| Field      | Type | Null | Key | Default | Extra |
+------------+------+------+-----+---------+-------+
| OrderID    | int  | NO   | PRI | NULL    |       |
| CustomerID | int  | YES  | MUL | NULL    |       |
| OrderDate  | date | YES  |     | NULL    |       |
+------------+------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> DESCRIBE ORDERDETAIL;
+---------------------+--------------+------+-----+---------+-------+
| Field               | Type         | Null | Key | Default | Extra |
+---------------------+--------------+------+-----+---------+-------+
| OrderDetailID       | int          | NO   | PRI | NULL    |       |
| OrderID             | int          | YES  | MUL | NULL    |       |
| MenuItemID          | int          | YES  | MUL | NULL    |       |
| Quantity            | int          | YES  |     | NULL    |       |
| SpecialInstructions | varchar(255) | YES  |     | NULL    |       |
+---------------------+--------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> DESCRIBE MENUITEMTYPE;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| MenuItemTypeID | int         | NO   | PRI | NULL    |       |
| TypeName       | varchar(50) | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```



```
mysql> DESCRIBE MENUITEM;
+----------------+---------------+------+-----+---------+-------+
| Field          | Type          | Null | Key | Default | Extra |
+----------------+---------------+------+-----+---------+-------+
| MenuItemID     | int           | NO   | PRI | NULL    |       |
| MenuName       | varchar(50)   | YES  |     | NULL    |       |
| Price          | decimal(10,2) | YES  |     | NULL    |       |
| Description    | varchar(255)  | YES  |     | NULL    |       |
| MenuItemTypeID | int           | YES  | MUL | NULL    |       |
+----------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> DESCRIBE PAYMENT;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| PaymentID     | int           | NO   | PRI | NULL    |       |
| OrderID       | int           | YES  | MUL | NULL    |       |
| PaymentAmount | decimal(10,2) | YES  |     | NULL    |       |
| PaymentDate   | date          | YES  |     | NULL    |       |
| PaymentMethod | varchar(50)   | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> DESCRIBE RATING;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| RatingID    | int          | NO   | PRI | NULL    |       |
| MenuItemID  | int          | YES  | MUL | NULL    |       |
| RatingValue | int          | YES  |     | NULL    |       |
| Review      | varchar(255) | YES  |     | NULL    |       |
| CustomerID  | int          | YES  | MUL | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

4.2. DATA SAMPLES INSIDE TABLES:

SELECT * FROM Customer;

SELECT * FROM Orders;

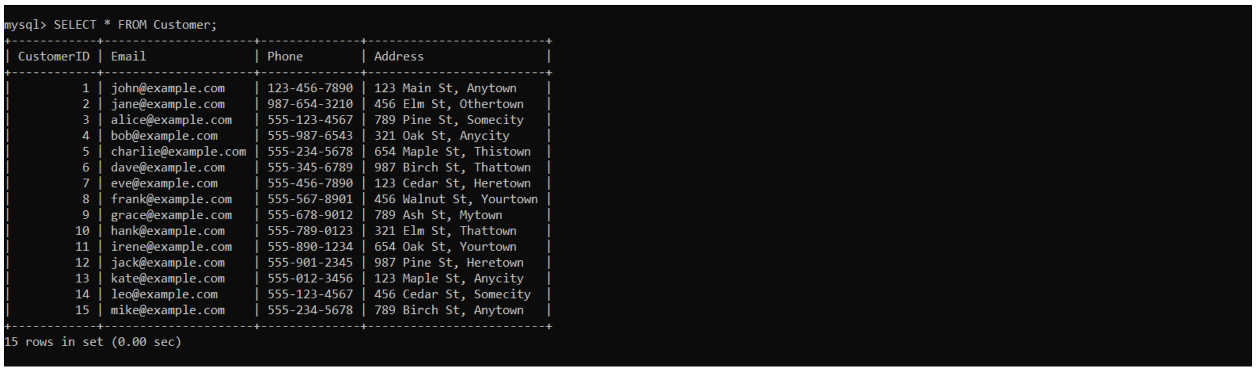
SELECT * FROM OrderDetail;

SELECT * FROM MenuItemType;

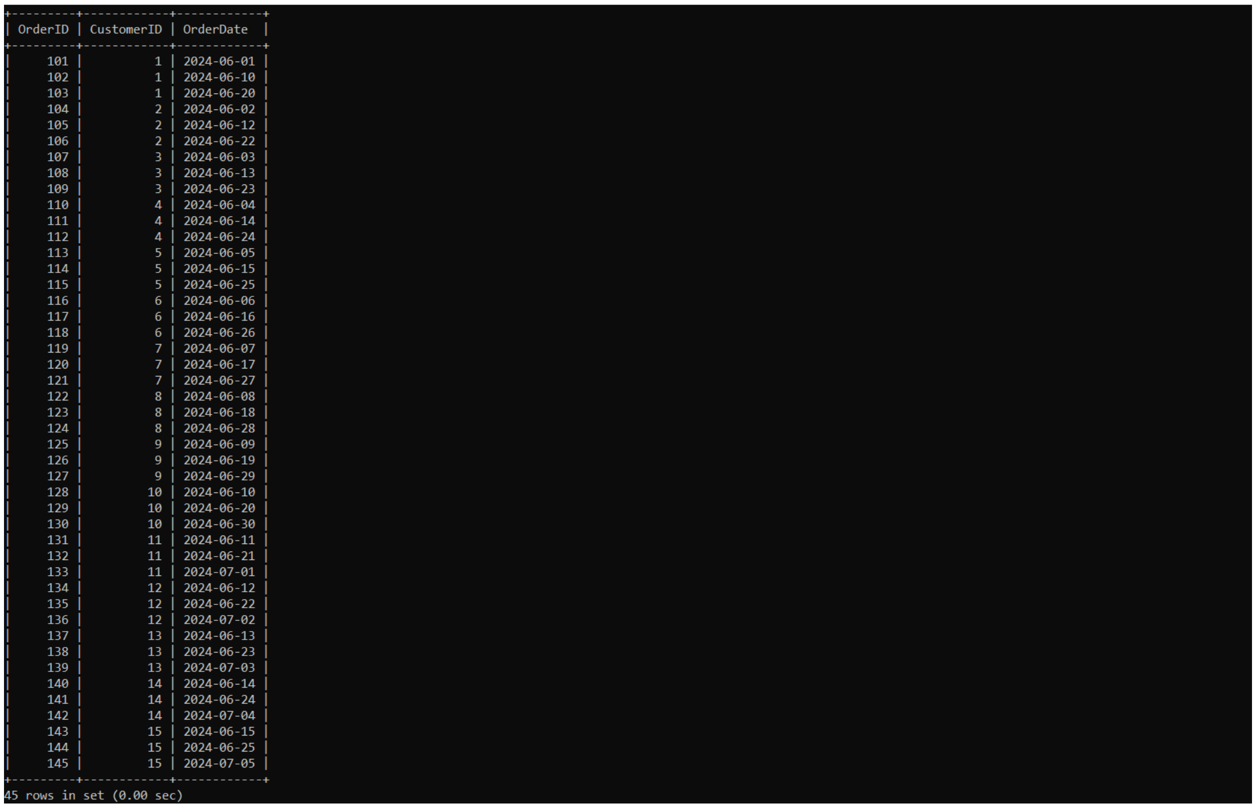
SELECT * FROM MenuItem;

SELECT * FROM Payment;

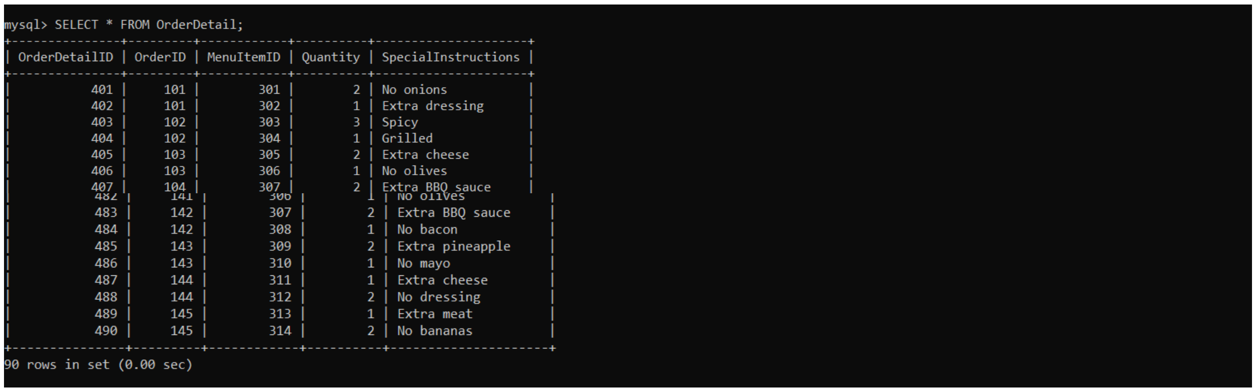
SELECT * FROM Rating;



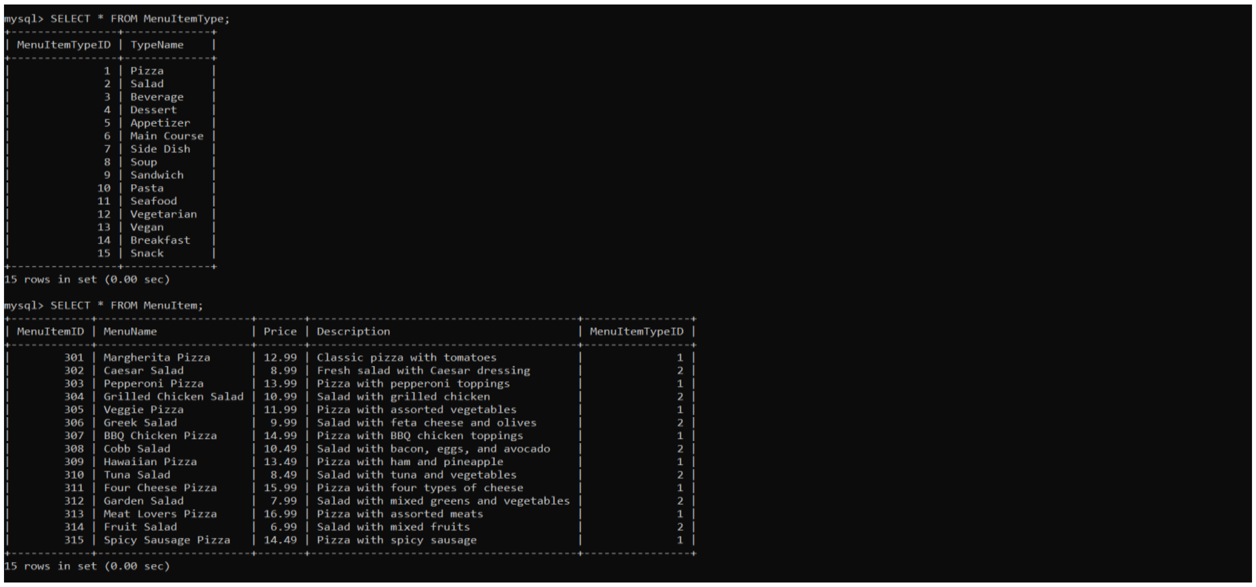
```
mysql> SELECT * FROM Customer;
+------------+---------------------+--------------+-------------------------+
| CustomerID | Email               | Phone        | Address                 |
+------------+---------------------+--------------+-------------------------+
|          1 | john@example.com    | 123-456-7890 | 123 Main St, Anytown    |
|          2 | jane@example.com    | 987-654-3210 | 456 Elm St, Othertown   |
|          3 | alice@example.com   | 555-123-4567 | 789 Pine St, Somecity   |
|          4 | bob@example.com     | 555-987-6543 | 321 Oak St, Anycity     |
|          5 | charlie@example.com | 555-234-5678 | 654 Maple St, Thistown  |
|          6 | dave@example.com    | 555-345-6789 | 987 Birch St, Thattown  |
|          7 | eve@example.com     | 555-456-7890 | 123 Cedar St, Heretown  |
|          8 | frank@example.com   | 555-567-8901 | 456 Walnut St, Yourtown |
|          9 | grace@example.com   | 555-678-9012 | 789 Ash St, Mytown      |
|         10 | hank@example.com    | 555-789-0123 | 321 Elm St, Thattown    |
|         11 | irene@example.com   | 555-890-1234 | 654 Oak St, Yourtown    |
|         12 | jack@example.com    | 555-901-2345 | 987 Pine St, Heretown   |
|         13 | kate@example.com    | 555-012-3456 | 123 Maple St, Anycity   |
|         14 | leo@example.com     | 555-123-4567 | 456 Cedar St, Somecity  |
|         15 | mike@example.com    | 555-234-5678 | 789 Birch St, Anytown   |
+------------+---------------------+--------------+-------------------------+
15 rows in set (0.00 sec)
```



```
+---------+------------+------------+
| OrderID | CustomerID | OrderDate  |
+---------+------------+------------+
|     101 |          1 | 2024-06-01 |
|     102 |          1 | 2024-06-10 |
|     103 |          1 | 2024-06-20 |
|     104 |          2 | 2024-06-02 |
|     105 |          2 | 2024-06-12 |
|     106 |          2 | 2024-06-22 |
|     107 |          3 | 2024-06-03 |
|     108 |          3 | 2024-06-13 |
|     109 |          3 | 2024-06-23 |
|     110 |          4 | 2024-06-04 |
|     111 |          4 | 2024-06-14 |
|     112 |          4 | 2024-06-24 |
|     113 |          5 | 2024-06-05 |
|     114 |          5 | 2024-06-15 |
|     115 |          5 | 2024-06-25 |
|     116 |          6 | 2024-06-06 |
|     117 |          6 | 2024-06-16 |
|     118 |          6 | 2024-06-26 |
|     119 |          7 | 2024-06-07 |
|     120 |          7 | 2024-06-17 |
|     121 |          7 | 2024-06-27 |
|     122 |          8 | 2024-06-08 |
|     123 |          8 | 2024-06-18 |
|     124 |          8 | 2024-06-28 |
|     125 |          9 | 2024-06-09 |
|     126 |          9 | 2024-06-19 |
|     127 |          9 | 2024-06-29 |
|     128 |         10 | 2024-06-10 |
|     129 |         10 | 2024-06-20 |
|     130 |         10 | 2024-06-30 |
|     131 |         11 | 2024-06-11 |
|     132 |         11 | 2024-06-21 |
|     133 |         11 | 2024-07-01 |
|     134 |         12 | 2024-06-12 |
|     135 |         12 | 2024-06-22 |
|     136 |         12 | 2024-07-02 |
|     137 |         13 | 2024-06-13 |
|     138 |         13 | 2024-06-23 |
|     139 |         13 | 2024-07-03 |
|     140 |         14 | 2024-06-14 |
|     141 |         14 | 2024-06-24 |
|     142 |         14 | 2024-07-04 |
|     143 |         15 | 2024-06-15 |
|     144 |         15 | 2024-06-25 |
|     145 |         15 | 2024-07-05 |
+---------+------------+------------+
45 rows in set (0.00 sec)
```

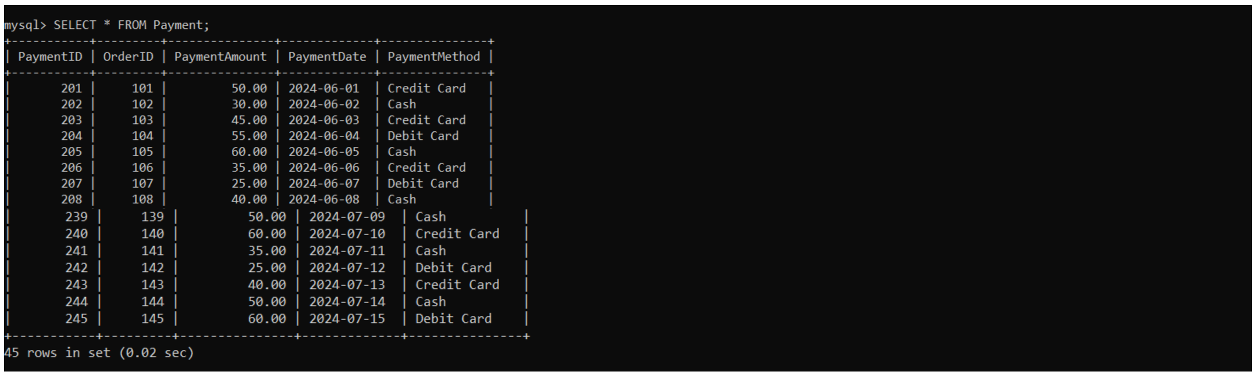


```
mysql> SELECT * FROM OrderDetail;
+---------------+---------+------------+----------+---------------------+
| OrderDetailID | OrderID | MenuItemID | Quantity | SpecialInstructions |
+---------------+---------+------------+----------+---------------------+
|           401 |     101 |        301 |        2 | No onions           |
|           402 |     101 |        302 |        1 | Extra dressing      |
|           403 |     102 |        303 |        3 | Spicy               |
|           404 |     102 |        304 |        1 | Grilled             |
|           405 |     103 |        305 |        2 | Extra cheese        |
|           406 |     103 |        306 |        1 | No olives           |
|           407 |     104 |        307 |        2 | Extra BBQ sauce     |
|           482 |     141 |        306 |        1 | No olives           |
|           483 |     142 |        307 |        2 | Extra BBQ sauce     |
|           484 |     142 |        308 |        1 | No bacon            |
|           485 |     143 |        309 |        2 | Extra pineapple     |
|           486 |     143 |        310 |        1 | No mayo             |
|           487 |     144 |        311 |        1 | Extra cheese        |
|           488 |     144 |        312 |        2 | No dressing         |
|           489 |     145 |        313 |        1 | Extra meat          |
|           490 |     145 |        314 |        2 | No bananas          |
+---------------+---------+------------+----------+---------------------+
90 rows in set (0.00 sec)
```

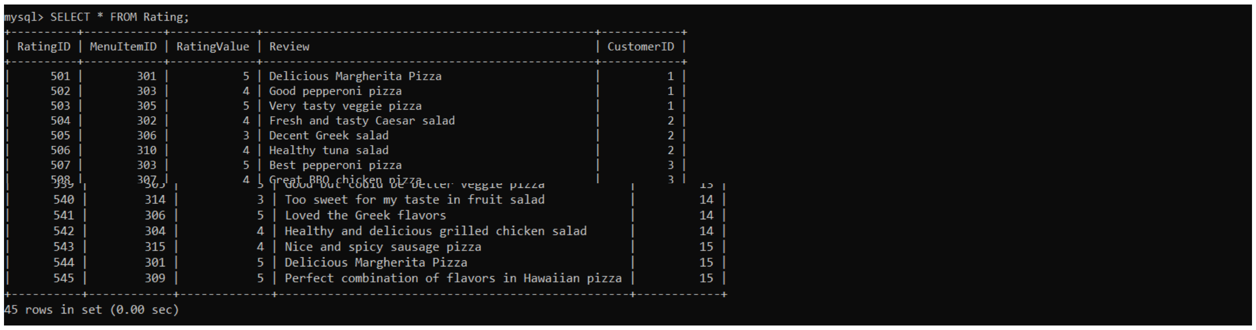


```
mysql> SELECT * FROM MenuItemType;
+----------------+-------------+
| MenuItemTypeID | TypeName    |
+----------------+-------------+
|              1 | Pizza       |
|              2 | Salad       |
|              3 | Beverage    |
|              4 | Dessert     |
|              5 | Appetizer   |
|              6 | Main Course |
|              7 | Side Dish   |
|              8 | Soup        |
|              9 | Sandwich    |
|             10 | Pasta       |
|             11 | Seafood     |
|             12 | Vegetarian  |
|             13 | Vegan       |
|             14 | Breakfast   |
|             15 | Snack       |
+----------------+-------------+
15 rows in set (0.00 sec)

mysql> SELECT * FROM MenuItem;
+------------+----------------------+-------+-----------------------------------------+----------------+
| MenuItemID | MenuName             | Price | Description                             | MenuItemTypeID |
+------------+----------------------+-------+-----------------------------------------+----------------+
|        301 | Margherita Pizza     | 12.99 | Classic pizza with tomatoes             |              1 |
|        302 | Caesar Salad         |  8.99 | Fresh salad with Caesar dressing        |              2 |
|        303 | Pepperoni Pizza      | 13.99 | Pizza with pepperoni toppings           |              1 |
|        304 | Grilled Chicken Salad| 10.99 | Salad with grilled chicken              |              2 |
|        305 | Veggie Pizza         | 11.99 | Pizza with assorted vegetables          |              1 |
|        306 | Greek Salad          |  9.99 | Salad with feta cheese and olives       |              2 |
|        307 | BBQ Chicken Pizza    | 14.99 | Pizza with BBQ chicken toppings         |              1 |
|        308 | Cobb Salad           | 10.49 | Salad with bacon, eggs, and avocado     |              2 |
|        309 | Hawaiian Pizza       | 13.49 | Pizza with ham and pineapple            |              1 |
|        310 | Tuna Salad           |  8.49 | Salad with tuna and vegetables          |              2 |
|        311 | Four Cheese Pizza    | 15.99 | Pizza with four types of cheese         |              1 |
|        312 | Garden Salad         |  7.99 | Salad with mixed greens and vegetables  |              2 |
|        313 | Meat Lovers Pizza    | 16.99 | Pizza with assorted meats               |              1 |
|        314 | Fruit Salad          |  6.99 | Salad with mixed fruits                 |              2 |
|        315 | Spicy Sausage Pizza  | 14.49 | Pizza with spicy sausage                |              1 |
+------------+----------------------+-------+-----------------------------------------+----------------+
15 rows in set (0.00 sec)
```



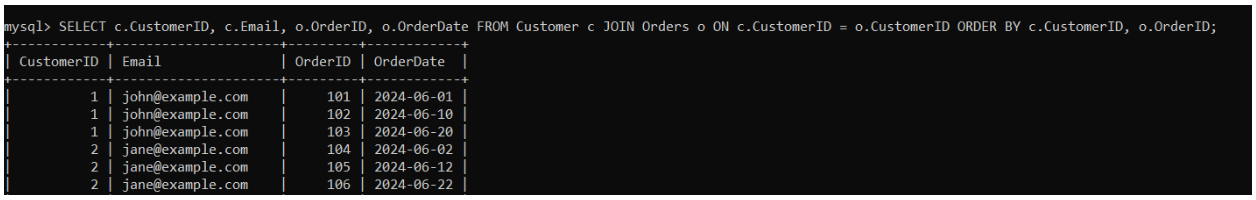
```
mysql> SELECT * FROM Payment;
+-----------+---------+---------------+-------------+---------------+
| PaymentID | OrderID | PaymentAmount | PaymentDate | PaymentMethod |
+-----------+---------+---------------+-------------+---------------+
|       201 |     101 |         50.00 | 2024-06-01  | Credit Card   |
|       202 |     102 |         30.00 | 2024-06-02  | Cash          |
|       203 |     103 |         45.00 | 2024-06-03  | Credit Card   |
|       204 |     104 |         55.00 | 2024-06-04  | Debit Card    |
|       205 |     105 |         60.00 | 2024-06-05  | Cash          |
|       206 |     106 |         35.00 | 2024-06-06  | Credit Card   |
|       207 |     107 |         25.00 | 2024-06-07  | Debit Card    |
|       208 |     108 |         40.00 | 2024-06-08  | Cash          |
|       239 |     139 |         50.00 | 2024-07-09  | Cash          |
|       240 |     140 |         60.00 | 2024-07-10  | Credit Card   |
|       241 |     141 |         35.00 | 2024-07-11  | Cash          |
|       242 |     142 |         25.00 | 2024-07-12  | Debit Card    |
|       243 |     143 |         40.00 | 2024-07-13  | Credit Card   |
|       244 |     144 |         50.00 | 2024-07-14  | Cash          |
|       245 |     145 |         60.00 | 2024-07-15  | Debit Card    |
+-----------+---------+---------------+-------------+---------------+
45 rows in set (0.02 sec)
```



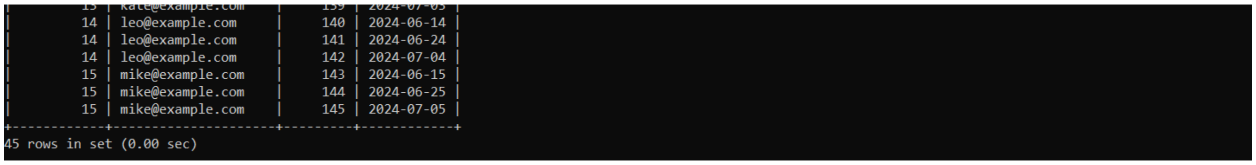
```
mysql> SELECT * FROM Rating;
+----------+------------+-------------+----------------------------------------------+------------+
| RatingID | MenuItemID | RatingValue | Review                                       | CustomerID |
+----------+------------+-------------+----------------------------------------------+------------+
|      501 |        301 |           5 | Delicious Margherita Pizza                   |          1 |
|      502 |        303 |           4 | Good pepperoni pizza                         |          1 |
|      503 |        305 |           5 | Very tasty veggie pizza                      |          1 |
|      504 |        302 |           4 | Fresh and tasty Caesar salad                 |          2 |
|      505 |        306 |           3 | Decent Greek salad                           |          2 |
|      506 |        310 |           4 | Healthy tuna salad                           |          2 |
|      507 |        303 |           5 | Best pepperoni pizza                         |          3 |
|      508 |        307 |           4 | Great BBQ chicken pizza                      |          3 |
|      539 |        305 |           5 | Good but could be better veggie pizza        |         13 |
|      540 |        314 |           3 | Too sweet for my taste in fruit salad        |         14 |
|      541 |        306 |           5 | Loved the Greek flavors                      |         14 |
|      542 |        304 |           4 | Healthy and delicious grilled chicken salad  |         14 |
|      543 |        315 |           4 | Nice and spicy sausage pizza                 |         15 |
|      544 |        301 |           5 | Delicious Margherita Pizza                   |         15 |
|      545 |        309 |           5 | Perfect combination of flavors in Hawaiian pizza |     15 |
+----------+------------+-------------+----------------------------------------------+------------+
45 rows in set (0.00 sec)
```

4.3. QUERIES RESULTS:

- SELECT c.CustomerID, c.Email, o.OrderID, o.OrderDate FROM Customer c JOIN Orders o ON c.CustomerID = o.CustomerID ORDER BY c.CustomerID, o.OrderID;

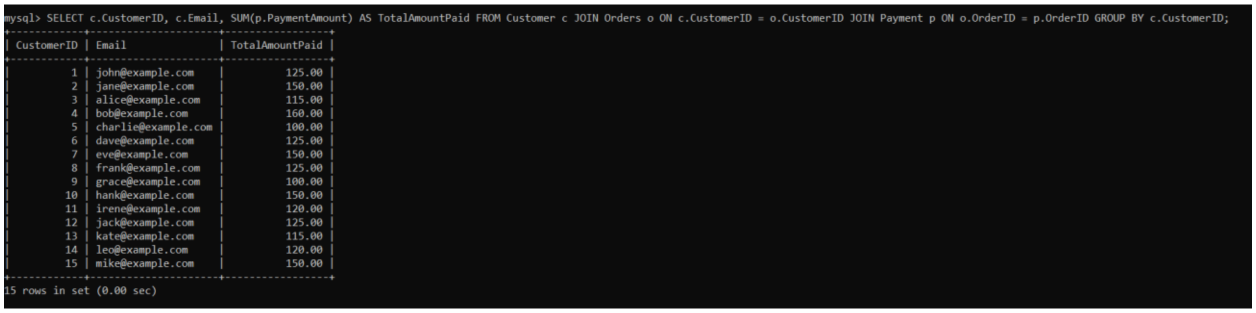


```
mysql> SELECT c.CustomerID, c.Email, o.OrderID, o.OrderDate FROM Customer c JOIN Orders o ON c.CustomerID = o.CustomerID ORDER BY c.CustomerID, o.OrderID;
+------------+--------------------+---------+------------+
| CustomerID | Email              | OrderID | OrderDate  |
+------------+--------------------+---------+------------+
|          1 | john@example.com   |     101 | 2024-06-01 |
|          1 | john@example.com   |     102 | 2024-06-10 |
|          1 | john@example.com   |     103 | 2024-06-20 |
|          2 | jane@example.com   |     104 | 2024-06-02 |
|          2 | jane@example.com   |     105 | 2024-06-12 |
|          2 | jane@example.com   |     106 | 2024-06-22 |
```



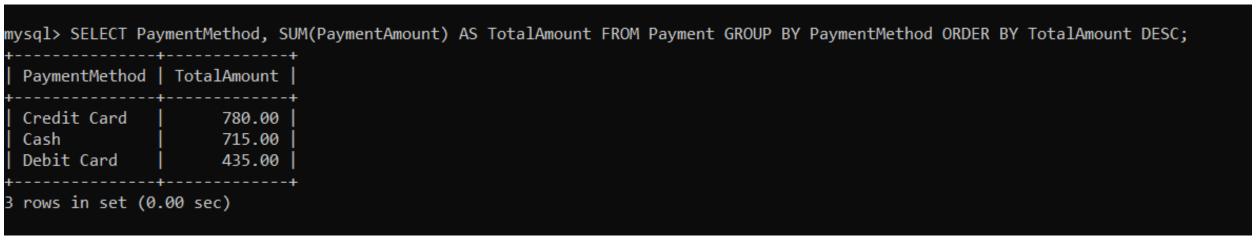
```
|         13 | kate@example.com   |     139 | 2024-07-03 |
|         14 | leo@example.com    |     140 | 2024-06-14 |
|         14 | leo@example.com    |     141 | 2024-06-24 |
|         14 | leo@example.com    |     142 | 2024-07-04 |
|         15 | mike@example.com   |     143 | 2024-06-15 |
|         15 | mike@example.com   |     144 | 2024-06-25 |
|         15 | mike@example.com   |     145 | 2024-07-05 |
+------------+--------------------+---------+------------+
45 rows in set (0.00 sec)
```

- SELECT c.CustomerID, c.Email, SUM(p.PaymentAmount) AS TotalAmountPaid FROM Customer c JOIN Orders o ON c.CustomerID = o.CustomerID JOIN Payment p ON o.OrderID = p.OrderID GROUP BY c.CustomerID;



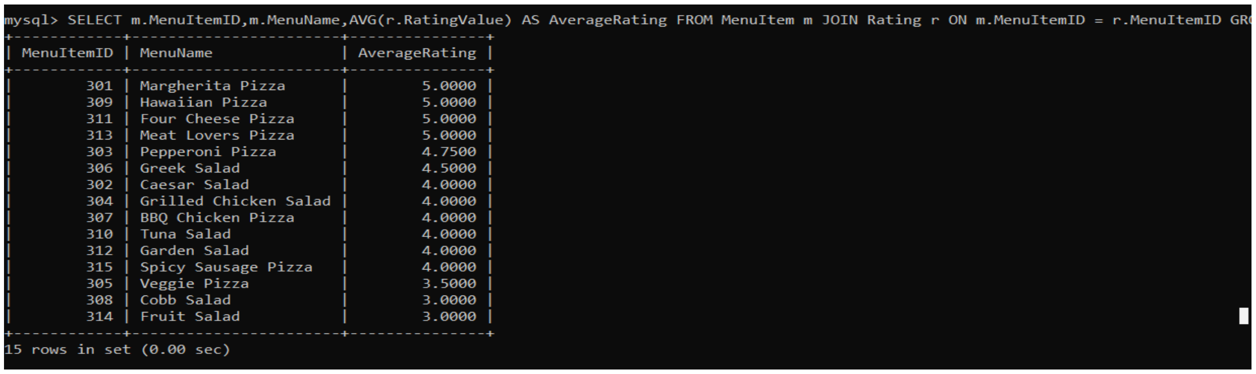
```
mysql> SELECT c.CustomerID, c.Email, SUM(p.PaymentAmount) AS TotalAmountPaid FROM Customer c JOIN Orders o ON c.CustomerID = o.CustomerID JOIN Payment p ON o.OrderID = p.OrderID GROUP BY c.CustomerID;
+------------+---------------------+-----------------+
| CustomerID | Email               | TotalAmountPaid |
+------------+---------------------+-----------------+
|          1 | john@example.com    |          125.00 |
|          2 | jane@example.com    |          150.00 |
|          3 | alice@example.com   |          115.00 |
|          4 | bob@example.com     |          160.00 |
|          5 | charlie@example.com |          100.00 |
|          6 | dave@example.com    |          125.00 |
|          7 | eve@example.com     |          150.00 |
|          8 | frank@example.com   |          125.00 |
|          9 | grace@example.com   |          100.00 |
|         10 | hank@example.com    |          150.00 |
|         11 | irene@example.com   |          120.00 |
|         12 | jack@example.com    |          125.00 |
|         13 | kate@example.com    |          115.00 |
|         14 | leo@example.com     |          120.00 |
|         15 | mike@example.com    |          150.00 |
+------------+---------------------+-----------------+
15 rows in set (0.00 sec)
```

- SELECT PaymentMethod, SUM(PaymentAmount) AS TotalAmount FROM Payment GROUP BY PaymentMethod ORDER BY TotalAmount DESC;



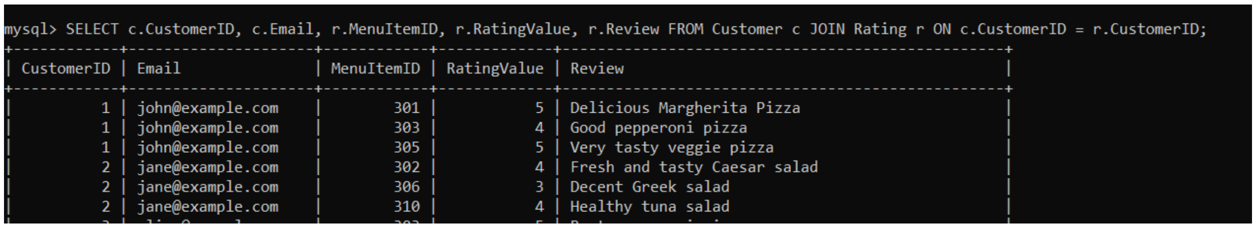
```
mysql> SELECT PaymentMethod, SUM(PaymentAmount) AS TotalAmount FROM Payment GROUP BY PaymentMethod ORDER BY TotalAmount DESC;
+---------------+-------------+
| PaymentMethod | TotalAmount |
+---------------+-------------+
| Credit Card   |      780.00 |
| Cash          |      715.00 |
| Debit Card    |      435.00 |
+---------------+-------------+
3 rows in set (0.00 sec)
```

- SELECT m.MenuItemID,m.MenuName,AVG(r.RatingValue) AS AverageRating FROM Menu m JOIN Rating r ON m.MenuItemID = r.MenuItemID GROUP BY m.MenuItemID, m.MenuName ORDER BY AverageRating DESC;

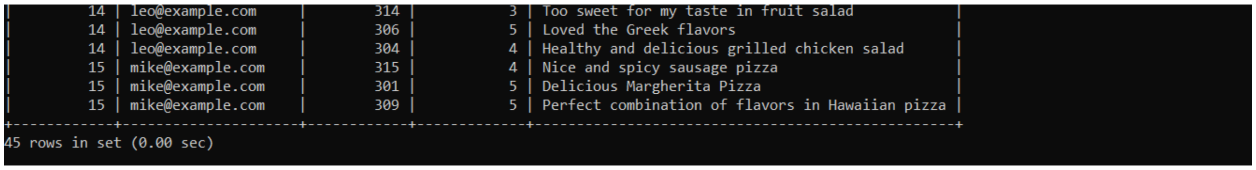


```
mysql> SELECT m.MenuItemID,m.MenuName,AVG(r.RatingValue) AS AverageRating FROM MenuItem m JOIN Rating r ON m.MenuItemID = r.MenuItemID GR
+------------+-----------------------+---------------+
| MenuItemID | MenuName              | AverageRating |
+------------+-----------------------+---------------+
|        301 | Margherita Pizza      |        5.0000 |
|        309 | Hawaiian Pizza        |        5.0000 |
|        311 | Four Cheese Pizza     |        5.0000 |
|        313 | Meat Lovers Pizza     |        5.0000 |
|        303 | Pepperoni Pizza       |        4.7500 |
|        306 | Greek Salad           |        4.5000 |
|        302 | Caesar Salad          |        4.0000 |
|        304 | Grilled Chicken Salad |        4.0000 |
|        307 | BBQ Chicken Pizza     |        4.0000 |
|        310 | Tuna Salad            |        4.0000 |
|        312 | Garden Salad          |        4.0000 |
|        315 | Spicy Sausage Pizza   |        4.0000 |
|        305 | Veggie Pizza          |        3.5000 |
|        308 | Cobb Salad            |        3.0000 |
|        314 | Fruit Salad           |        3.0000 |
+------------+-----------------------+---------------+
15 rows in set (0.00 sec)
```

- SELECT c.CustomerID, c.Email, r.MenuItemID, r.RatingValue, r.Review FROM Customer c JOIN Rating r ON c.CustomerID = r.CustomerID;

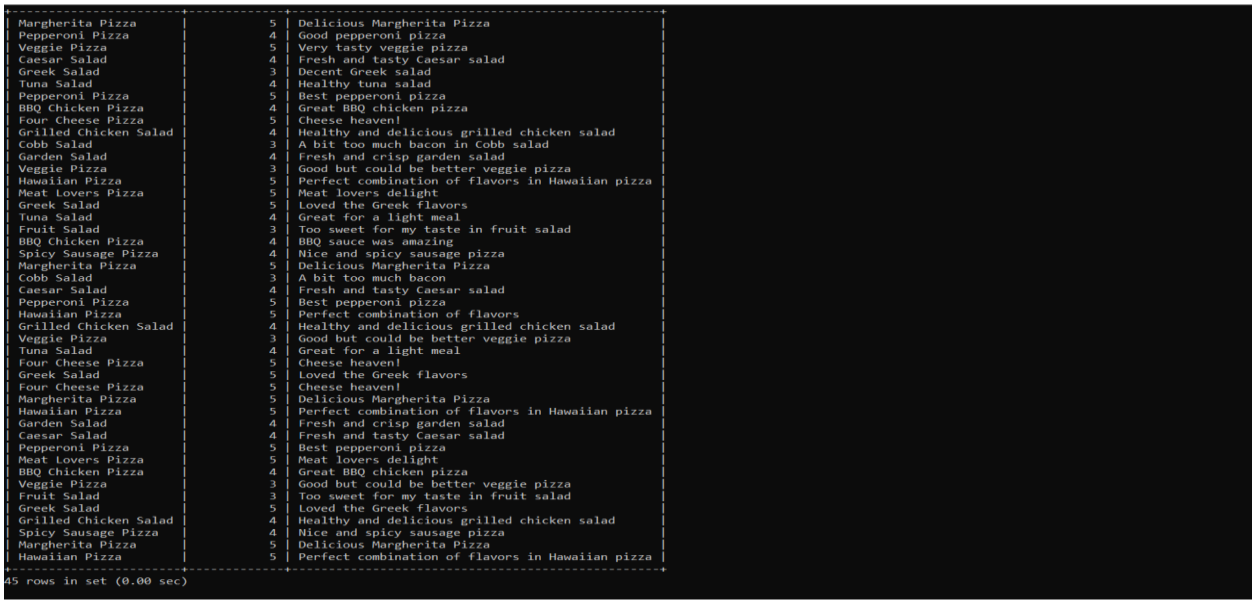


```
mysql> SELECT c.CustomerID, c.Email, r.MenuItemID, r.RatingValue, r.Review FROM Customer c JOIN Rating r ON c.CustomerID = r.CustomerID;
+------------+--------------------+------------+-------------+----------------------------------------------+
| CustomerID | Email              | MenuItemID | RatingValue | Review                                       |
+------------+--------------------+------------+-------------+----------------------------------------------+
|          1 | john@example.com   |        301 |           5 | Delicious Margherita Pizza                   |
|          1 | john@example.com   |        303 |           4 | Good pepperoni pizza                         |
|          1 | john@example.com   |        305 |           5 | Very tasty veggie pizza                      |
|          2 | jane@example.com   |        302 |           4 | Fresh and tasty Caesar salad                 |
|          2 | jane@example.com   |        306 |           3 | Decent Greek salad                           |
|          2 | jane@example.com   |        310 |           4 | Healthy tuna salad                           |
```



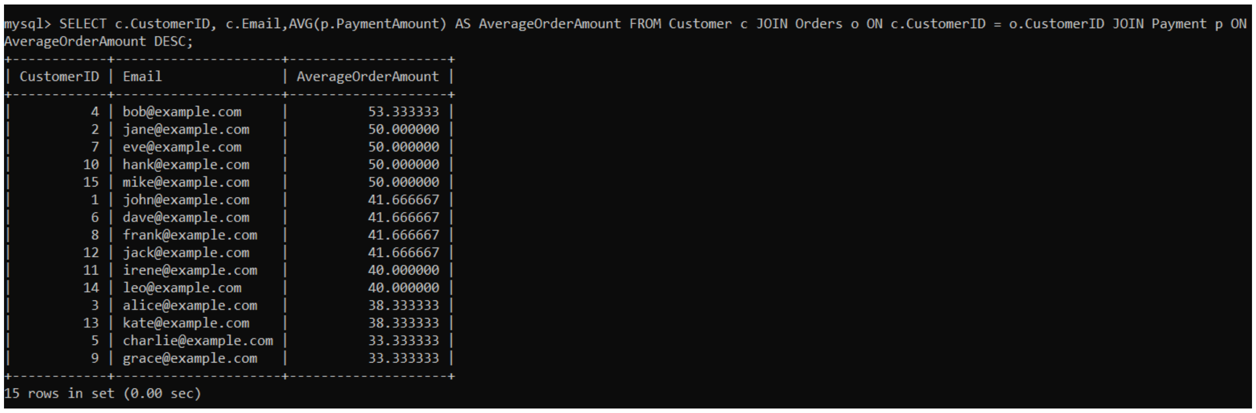
```
|         14 | leo@example.com    |        314 |           3 | Too sweet for my taste in fruit salad        |
|         14 | leo@example.com    |        306 |           5 | Loved the Greek flavors                      |
|         14 | leo@example.com    |        304 |           4 | Healthy and delicious grilled chicken salad  |
|         15 | mike@example.com   |        315 |           4 | Nice and spicy sausage pizza                 |
|         15 | mike@example.com   |        301 |           5 | Delicious Margherita Pizza                   |
|         15 | mike@example.com   |        309 |           5 | Perfect combination of flavors in Hawaiian pizza |
+------------+--------------------+------------+-------------+----------------------------------------------+
45 rows in set (0.00 sec)
```

- SELECT m.MenuName, r.RatingValue, r.Review FROM Menu m JOIN Rating r ON m.MenuItemID = r.MenuItemID WHERE r.RatingValue >= 3;



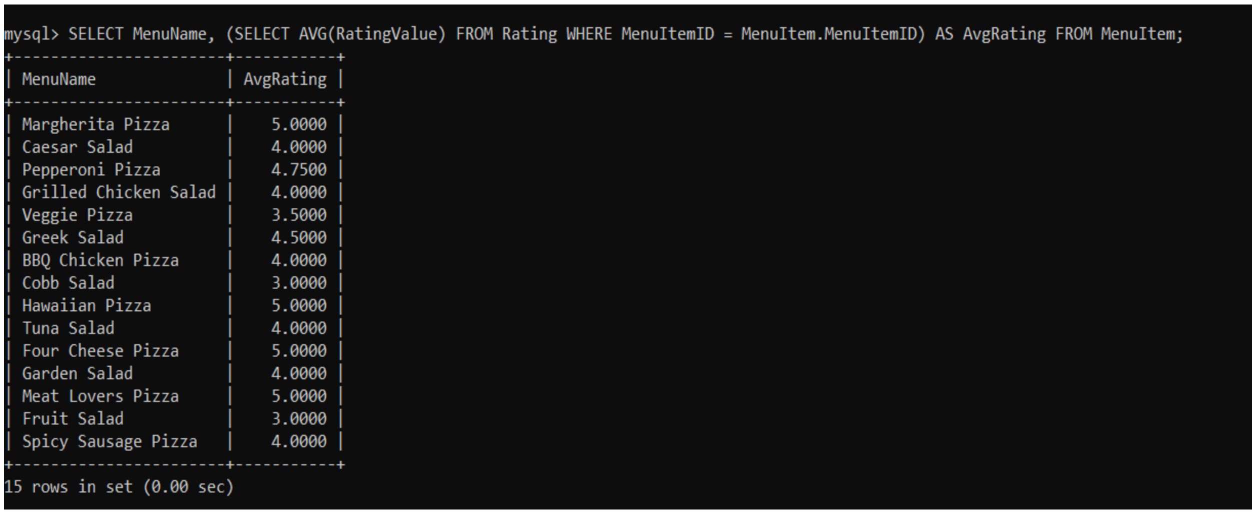
```
+-----------------------+-------------+--------------------------------------------------+
| Margherita Pizza      |           5 | Delicious Margherita Pizza                       |
| Pepperoni Pizza       |           4 | Good pepperoni pizza                             |
| Veggie Pizza          |           5 | Very tasty veggie pizza                          |
| Caesar Salad          |           4 | Fresh and tasty Caesar salad                     |
| Greek Salad           |           3 | Decent Greek salad                               |
| Tuna Salad            |           4 | Healthy tuna salad                               |
| Pepperoni Pizza       |           5 | Best pepperoni pizza                             |
| BBQ Chicken Pizza     |           4 | Great BBQ chicken pizza                          |
| Four Cheese Pizza     |           5 | Cheese heaven!                                   |
| Grilled Chicken Salad |           4 | Healthy and delicious grilled chicken salad      |
| Cobb Salad            |           3 | A bit too much bacon in Cobb salad               |
| Garden Salad          |           4 | Fresh and crisp garden salad                     |
| Veggie Pizza          |           3 | Good but could be better veggie pizza            |
| Hawaiian Pizza        |           5 | Perfect combination of flavors in Hawaiian pizza |
| Meat Lovers Pizza     |           5 | Meat lovers delight                              |
| Greek Salad           |           5 | Loved the Greek flavors                          |
| Tuna Salad            |           4 | Great for a light meal                           |
| Fruit Salad           |           3 | Too sweet for my taste in fruit salad            |
| BBQ Chicken Pizza     |           4 | BBQ sauce was amazing                            |
| Spicy Sausage Pizza   |           4 | Nice and spicy sausage pizza                     |
| Margherita Pizza      |           5 | Delicious Margherita Pizza                       |
| Cobb Salad            |           3 | A bit too much bacon                             |
| Caesar Salad          |           4 | Fresh and tasty Caesar salad                     |
| Pepperoni Pizza       |           5 | Best pepperoni pizza                             |
| Hawaiian Pizza        |           5 | Perfect combination of flavors                   |
| Grilled Chicken Salad |           4 | Healthy and delicious grilled chicken salad      |
| Veggie Pizza          |           3 | Good but could be better veggie pizza            |
| Tuna Salad            |           4 | Great for a light meal                           |
| Four Cheese Pizza     |           5 | Cheese heaven!                                   |
| Greek Salad           |           5 | Loved the Greek flavors                          |
| Four Cheese Pizza     |           5 | Cheese heaven!                                   |
| Margherita Pizza      |           5 | Delicious Margherita Pizza                       |
| Hawaiian Pizza        |           5 | Perfect combination of flavors in Hawaiian pizza |
| Garden Salad          |           4 | Fresh and crisp garden salad                     |
| Caesar Salad          |           4 | Fresh and tasty Caesar salad                     |
| Pepperoni Pizza       |           5 | Best pepperoni pizza                             |
| Meat Lovers Pizza     |           5 | Meat lovers delight                              |
| BBQ Chicken Pizza     |           4 | Great BBQ chicken pizza                          |
| Veggie Pizza          |           3 | Good but could be better veggie pizza            |
| Fruit Salad           |           3 | Too sweet for my taste in fruit salad            |
| Greek Salad           |           5 | Loved the Greek flavors                          |
| Grilled Chicken Salad |           4 | Healthy and delicious grilled chicken salad      |
| Spicy Sausage Pizza   |           4 | Nice and spicy sausage pizza                     |
| Margherita Pizza      |           5 | Delicious Margherita Pizza                       |
| Hawaiian Pizza        |           5 | Perfect combination of flavors in Hawaiian pizza |
+-----------------------+-------------+--------------------------------------------------+
45 rows in set (0.00 sec)
```

- SELECT c.CustomerID,c.CustomerType,c.Email,AVG(p.PaymentAmount) AS AverageOrderAmount FROM Customer c JOIN Orders o ON c.CustomerID = o.CustomerID JOIN Payment p ON o.OrderID = p.OrderID GROUP BY c.CustomerID ORDER BY AverageOrderAmount DESC;



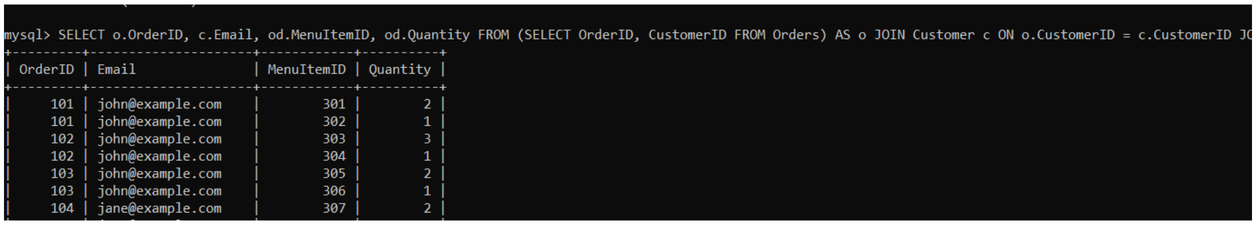
```
mysql> SELECT c.CustomerID, c.Email,AVG(p.PaymentAmount) AS AverageOrderAmount FROM Customer c JOIN Orders o ON c.CustomerID = o.CustomerID JOIN Payment p ON
AverageOrderAmount DESC;
+------------+---------------------+--------------------+
| CustomerID | Email               | AverageOrderAmount |
+------------+---------------------+--------------------+
|          4 | bob@example.com     |          53.333333 |
|          2 | jane@example.com    |          50.000000 |
|          7 | eve@example.com     |          50.000000 |
|         10 | hank@example.com    |          50.000000 |
|         15 | mike@example.com    |          50.000000 |
|          1 | john@example.com    |          41.666667 |
|          6 | dave@example.com    |          41.666667 |
|          8 | frank@example.com   |          41.666667 |
|         12 | jack@example.com    |          41.666667 |
|         11 | irene@example.com   |          40.000000 |
|         14 | leo@example.com     |          40.000000 |
|          3 | alice@example.com   |          38.333333 |
|         13 | kate@example.com    |          38.333333 |
|          5 | charlie@example.com |          33.333333 |
|          9 | grace@example.com   |          33.333333 |
+------------+---------------------+--------------------+
15 rows in set (0.00 sec)
```

- SELECT MenuName, (SELECT AVG(RatingValue) FROM Rating WHERE MenuItemID = Menu.MenuItemID) AS AvgRating FROM Menu;

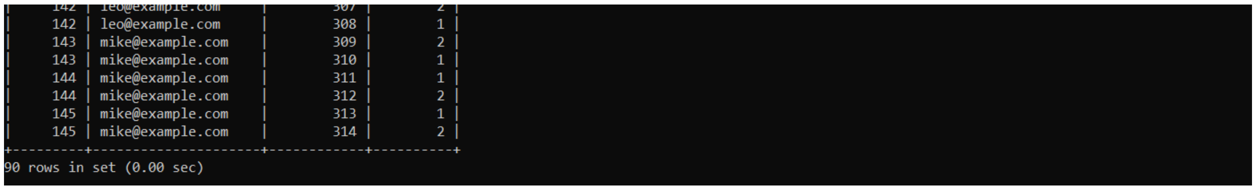


```
mysql> SELECT MenuName, (SELECT AVG(RatingValue) FROM Rating WHERE MenuItemID = MenuItem.MenuItemID) AS AvgRating FROM MenuItem;
+----------------------+-----------+
| MenuName             | AvgRating |
+----------------------+-----------+
| Margherita Pizza     |    5.0000 |
| Caesar Salad         |    4.0000 |
| Pepperoni Pizza      |    4.7500 |
| Grilled Chicken Salad |   4.0000 |
| Veggie Pizza         |    3.5000 |
| Greek Salad          |    4.5000 |
| BBQ Chicken Pizza    |    4.0000 |
| Cobb Salad           |    3.0000 |
| Hawaiian Pizza       |    5.0000 |
| Tuna Salad           |    4.0000 |
| Four Cheese Pizza    |    5.0000 |
| Garden Salad         |    4.0000 |
| Meat Lovers Pizza    |    5.0000 |
| Fruit Salad          |    3.0000 |
| Spicy Sausage Pizza  |    4.0000 |
+----------------------+-----------+
15 rows in set (0.00 sec)
```

- SELECT o.OrderID, c.Email, od.MenuItemID, od.Quantity, od.Price FROM (SELECT OrderID, CustomerID FROM Orders) AS o JOIN Customer c ON o.CustomerID = c.CustomerID JOIN OrderDetail od ON o.OrderID = od.OrderID;

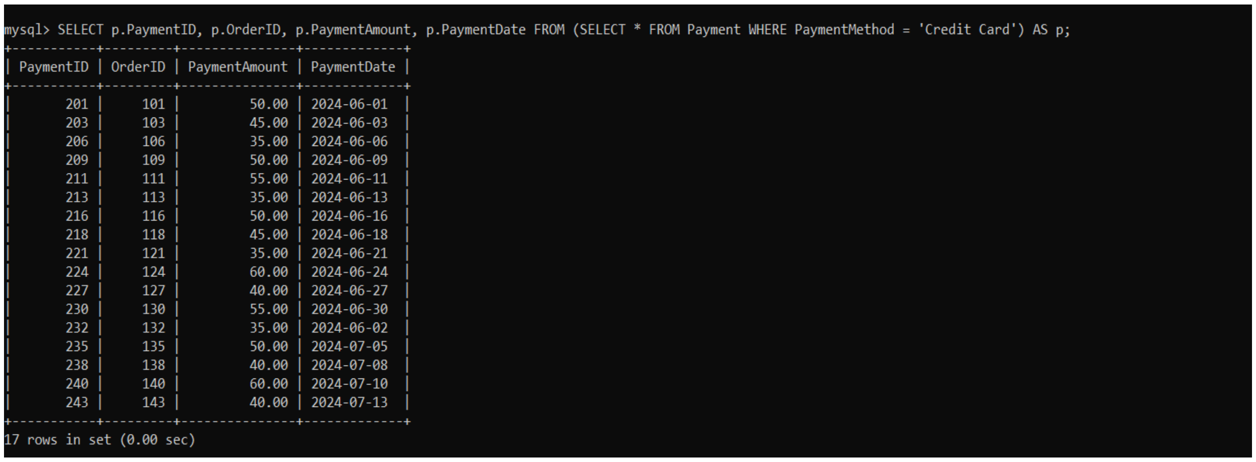


```
mysql> SELECT o.OrderID, c.Email, od.MenuItemID, od.Quantity FROM (SELECT OrderID, CustomerID FROM Orders) AS o JOIN Customer c ON o.CustomerID = c.CustomerID JO
+---------+--------------------+------------+----------+
| OrderID | Email              | MenuItemID | Quantity |
+---------+--------------------+------------+----------+
|     101 | john@example.com   |        301 |        2 |
|     101 | john@example.com   |        302 |        1 |
|     102 | john@example.com   |        303 |        3 |
|     102 | john@example.com   |        304 |        1 |
|     103 | john@example.com   |        305 |        2 |
|     103 | john@example.com   |        306 |        1 |
|     104 | jane@example.com   |        307 |        2 |
```



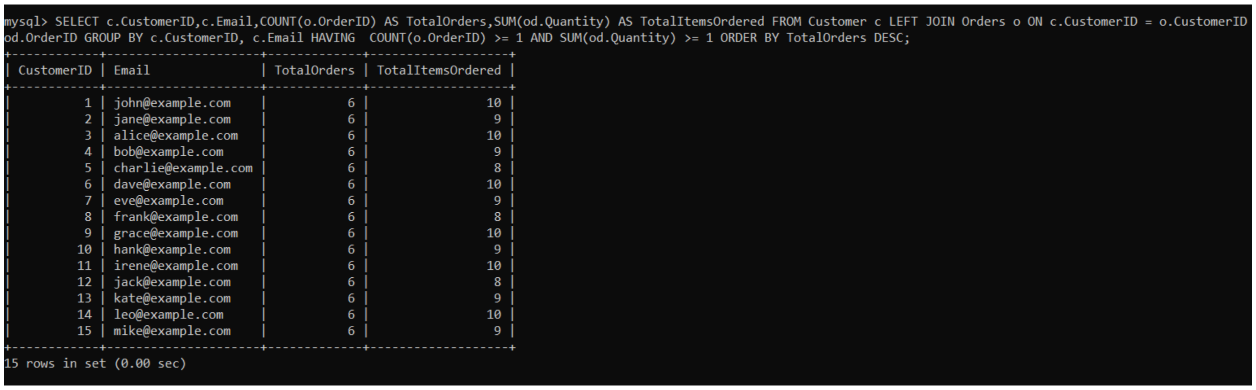
```
|     142 | leo@example.com    |        307 |        2 |
|     142 | leo@example.com    |        308 |        1 |
|     143 | mike@example.com   |        309 |        2 |
|     143 | mike@example.com   |        310 |        1 |
|     144 | mike@example.com   |        311 |        1 |
|     144 | mike@example.com   |        312 |        2 |
|     145 | mike@example.com   |        313 |        1 |
|     145 | mike@example.com   |        314 |        2 |
+---------+--------------------+------------+----------+
90 rows in set (0.00 sec)
```

- SELECT p.PaymentID, p.OrderID, p.PaymentAmount, p.PaymentDate FROM (SELECT * FROM Payment WHERE PaymentMethod = 'Credit Card') AS p;



```
mysql> SELECT p.PaymentID, p.OrderID, p.PaymentAmount, p.PaymentDate FROM (SELECT * FROM Payment WHERE PaymentMethod = 'Credit Card') AS p;
+-----------+---------+---------------+-------------+
| PaymentID | OrderID | PaymentAmount | PaymentDate |
+-----------+---------+---------------+-------------+
|       201 |     101 |         50.00 | 2024-06-01  |
|       203 |     103 |         45.00 | 2024-06-03  |
|       206 |     106 |         35.00 | 2024-06-06  |
|       209 |     109 |         50.00 | 2024-06-09  |
|       211 |     111 |         55.00 | 2024-06-11  |
|       213 |     113 |         35.00 | 2024-06-13  |
|       216 |     116 |         50.00 | 2024-06-16  |
|       218 |     118 |         45.00 | 2024-06-18  |
|       221 |     121 |         35.00 | 2024-06-21  |
|       224 |     124 |         60.00 | 2024-06-24  |
|       227 |     127 |         40.00 | 2024-06-27  |
|       230 |     130 |         55.00 | 2024-06-30  |
|       232 |     132 |         35.00 | 2024-06-02  |
|       235 |     135 |         50.00 | 2024-07-05  |
|       238 |     138 |         40.00 | 2024-07-08  |
|       240 |     140 |         60.00 | 2024-07-10  |
|       243 |     143 |         40.00 | 2024-07-13  |
+-----------+---------+---------------+-------------+
17 rows in set (0.00 sec)
```

- SELECT c.CustomerID,c.Email,COUNT(o.OrderID) AS TotalOrders,SUM(od.Quantity) AS TotalItemsOrdered FROM Customer c LEFT JOIN Orders o ON c.CustomerID = o.CustomerID LEFT JOIN OrderDetail od ON o.OrderID = od.OrderID GROUP BY c.CustomerID, c.Email HAVING COUNT(o.OrderID) >= 1 AND SUM(od.Quantity) >= 1 ORDER BY TotalOrders DESC;



```
mysql> SELECT c.CustomerID,c.Email,COUNT(o.OrderID) AS TotalOrders,SUM(od.Quantity) AS TotalItemsOrdered FROM Customer c LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
od.OrderID GROUP BY c.CustomerID, c.Email HAVING  COUNT(o.OrderID) >= 1 AND SUM(od.Quantity) >= 1 ORDER BY TotalOrders DESC;
+------------+---------------------+-------------+-------------------+
| CustomerID | Email               | TotalOrders | TotalItemsOrdered |
+------------+---------------------+-------------+-------------------+
|          1 | john@example.com    |           6 |                10 |
|          2 | jane@example.com    |           6 |                 9 |
|          3 | alice@example.com   |           6 |                10 |
|          4 | bob@example.com     |           6 |                 9 |
|          5 | charlie@example.com |           6 |                 8 |
|          6 | dave@example.com    |           6 |                10 |
|          7 | eve@example.com     |           6 |                 9 |
|          8 | frank@example.com   |           6 |                 8 |
|          9 | grace@example.com   |           6 |                10 |
|         10 | hank@example.com    |           6 |                 9 |
|         11 | irene@example.com   |           6 |                10 |
|         12 | jack@example.com    |           6 |                 8 |
|         13 | kate@example.com    |           6 |                 9 |
|         14 | leo@example.com     |           6 |                10 |
|         15 | mike@example.com    |           6 |                 9 |
+------------+---------------------+-------------+-------------------+
15 rows in set (0.00 sec)
```

REFERENCES

Provide a list of all the sources cited or consulted during the development of database project. This section serves to acknowledge the contributions of other authors and researchers, as well as to enable readers to locate the sources for further information. Ensure that references are formatted according to IEEE referencing style. Also provide the in-text citations of the references

Formatting Guidelines:

- *Level 1 Heading: Font Style: Times New Roman, Font Size: 18, Color: Black, Case: All Caps, Align: Right, Numbering Style: CHAPTER 1… Should appear in Table of Content*
- *Level 2 Heading: Font Style: Times New Roman, Font Size 16, Color Black, Case: All Caps, Align: Lef,t Numbering Style: 1.1, 1.2 … Should appear in Table of Content*
- *Level 3 Heading: Font Style: Times New Roman, Font Size 14, Color Black, Case: Capitalize each word, Align: Left, Numbering Style: 1.1.1, 1.1.2 … Should Not appear in Table of Content*
- *Paragraph: Font Style: Times New Roman, Font Size 12, Color Black, Case: Sentence Case, Align: Justified, Should Not appear in Table of Content, Add space before and after paragraph.*
- *Insert caption to the picture with figure number.*
- *Insert caption to table with table number.*

Any text that is in blue and italicized is for your understanding only and should not be included in your submission document.