



Faculty of Engineering - University of Ruhuna

## **EE4350 - Database Systems**

Mini Project Title

### **Online Retail Store management system**

On 9<sup>th</sup> of April

Group 66 - Members

EG/2021/4875 – Wijesinghe D.M.A.B.

EG/2021/4877 – Wijesinghe S.A.

EG/2021/4890 – Wijegurusinghe O.W.D.K.M.

# Content

<b>Relational Database .....</b>	5
<b>Chapter 1 – Requirement Analysis .....</b>	5
<b>Chapter 2 – Conceptual Design .....</b>	8
<b>Chapter 3 – Implementation .....</b>	9
3.1 Creating operation.....	9
3.2 Inserting operation .....	18
3.3 Updating operation .....	26
3.4. Deleting operation.....	35
<b>Chapter 4 – Transaction.....</b>	44
4.1 Simple queries.....	44
4.1.1 Select operation.....	44
4.1.2 Project operation .....	44
4.1.3 Cartesian operation .....	45
4.1.4. Creating a user view .....	45
4.1.5. Renaming operation.....	46
4.1.6. Aggregation function .....	46
4.1.7. Demonstrating the use of LIKE keyword .....	47
4.2. Complex queries .....	47
4.2.1. Basic set operations (union, intersection, set difference, division) without user views.....	47
4.2.2. Inner join, natural join, left outer join, right outer join, full outer join, outer union relational algebraic operations by using user views .....	49
<b>Chapter 5 – Database Tunning.....</b>	54

## List of Figures

Figure 1. Entity - relationship diagram.....	8
Figure 2. UML class diagram.....	8
Figure 3. Creating database .....	9
Figure 4. Creating delivery_man table .....	9
Figure 5. Creating customer table .....	10
Figure 6. creating order_details table .....	10
Figure 7. creating customer_email table .....	11
Figure 8. creating shipping_address table .....	11
Figure 9. creating payment table .....	12
Figure 10. creating warehouse table .....	12
Figure 11. creating invetory table.....	13
Figure 12. creating product table.....	13
Figure 13. creating order_item table .....	14
Figure 14. creating supplier table .....	14
Figure 15. creating product_supplier table .....	15
Figure 16. creating product_warehouse table.....	15
Figure 17. creating employee table .....	16
Figure 18. creating product_features table .....	16
Figure 19. creating employee_inventory table .....	17
Figure 20. creating employee_email table.....	17
Figure 21. Inserting delivery_man table.....	18
Figure 22. Inserting customer table .....	18
Figure 23. Inserting order_details table.....	19
Figure 24. Inserting customer_email table .....	19
Figure 25. Inserting shipping_address table .....	20
Figure 26. Inserting payment table .....	20
Figure 27. Inserting warehouse table.....	21
Figure 28. Inserting inventory table .....	21
Figure 29. Inserting product table .....	22
Figure 30. Inserting order_item table .....	22
Figure 31. Inserting supplier table.....	23
Figure 32. Inserting product_supplier table.....	23
Figure 33. Inserting product_warehouse table .....	24
Figure 34. Inserting employee table .....	24
Figure 35. Inserting product_feature table .....	25
Figure 36. Inserting employee_email table .....	25
Figure 37. Inserting employee_inventory table .....	26
Figure 38. Updating delivery_Man table.....	26
Figure 39. Updating customer table .....	27
Figure 40. Updating order_details table .....	27
Figure 41. Updating customer_email table .....	28
Figure 42. Updating shipping_address table .....	28
Figure 43. Updating payment table .....	29
Figure 44. Updating warehouse table .....	29
Figure 45. Updating inventory table.....	30
Figure 46. Updating product_table .....	30
Figure 47. Updating order_item table .....	31
Figure 48. Updating supplier table .....	31
Figure 49. Updating product_supplier table .....	32

Figure 50. Updating product_warehouse table .....	32
Figure 51. Updating employee table .....	33
Figure 52. Updating product_feactures table .....	33
Figure 53. Updating employee_email table.....	34
Figure 54. Updating employee_inventory table .....	34
Figure 55. deleting from delivery_man table .....	35
Figure 56. deleting from customer table.....	35
Figure 57. deleting from order_details table .....	36
Figure 58. deleting from customer_email table.....	36
Figure 59. deleting from shipping_address table .....	37
Figure 60. deleting from payment table .....	37
Figure 61. deleting from warehouse table .....	38
Figure 62. deleting from inventory table.....	38
Figure 63. deleting from product table .....	39
Figure 64. deleting from order item table.....	39
Figure 65. deleting from supplier table .....	40
Figure 66. deleting from product supplier table .....	40
Figure 67. deleting from product_warehouse table .....	41
Figure 68. deleting from employee table.....	41
Figure 69. deleting from product_feactures table.....	42
Figure 70. deleting from employee_email table.....	42
Figure 71. deleting from employee_inventory table .....	43
Figure 72.simple queries - select operation.....	44
Figure 73, simple queries - project operation.....	44
Figure 74. simple queries - cartesian operation.....	45
Figure 75. simple queries - creating a new user view .....	45
Figure 76. simple queries - renaming operation.....	46
Figure 77. simple queries - aggregation function.....	46
Figure 78. simple queries - demonstrating the use of like keyword.....	47
Figure 79. Basic set operation - Union.....	47
Figure 80. Basic set operation – Set Difference .....	48
Figure 81. Basic set operation – Division .....	48
Figure 82. Basic set operation – Intersection .....	49
Figure 83. Inner join with User view.....	49
Figure 84. Natural join with User view .....	50
Figure 85. Left Outer join with User view .....	50
Figure 86. Right Outer join with User view .....	51
Figure 87. Full outer join with user view .....	51
Figure 88. Outer Union with User view .....	52
Figure 89. Nested queries_1 .....	52
Figure 90. Nested queries_2 .....	53
Figure 91. Nested queries_3 .....	53
Figure 92. Database Tuning Figure_1 .....	54
Figure 93. Database Tuning Figure_2 .....	54
Figure 94. Database Tuning Figure_3 .....	55
Figure 95. Database Tuning Figure_4 .....	55
Figure 96. Database Tuning Figure_5 .....	56
Figure 97. Database Tuning Figure_6 .....	56
Figure 98. Database Tuning Figure_7 .....	57
Figure 99. Database Tuning Figure_8 .....	57
Figure 100. Database Tuning Figure_9 .....	58
Figure 101. Database Tuning Figure_10 .....	58
Figure 102. Database Tuning Figure_11 .....	59

## **Relational Database**

### **Chapter 1 – Requirement Analysis**

#### **1.1. Functional Requirements**

Functional requirements in a database specify essential capabilities like data storage, retrieval, manipulation, integrity, concurrency control, security etc. These requirements ensure the database effectively supports user needs and organizational objectives while maintaining data accuracy, reliability and accessibility.

This schema was created in order to represent a database of an Online Retail Store management system. In this retail store schema suitable entities are chosen in order to represent the schema. Customer, Order, Deliver man, product, payment, warehouse, Inventory are the areas covered by this schema. The data spread between those areas can be retrieved by using this database.

#### **1.2. Data Requirements**

##### **Strong entities and attributes**

###### **Customer**

- Name
- CustomerID
- Permanent Address
- PhoneNo
- DOB
- Age
- Email

###### **Order**

- OrderID
- OrderDate
- Total\_Amount
- Status

###### **Order\_Item**

- OrderitemID
- Quantity
- Price

### Deliverman

- Contact\_Number
- DeliverymanID
- Name

### Payment

- PaymentID
- Amount
- PaymentData
- PaymentMethod

### Product

- ProductID
- Name
- Description
- Price
- Stock\_Quantity

### Supplier

- SupplierID
- Name
- ConractPerson
- ContactNumber

### Employee

- EmployeeID
- City
- District
- StreetName
- PhoneNo
- DOB
- Email
- Age

### Inventory

- InventoryID
- StockQuantity
- LastStockUpdate

### Warehouse

- Capacity
- Location
- WarehouseID

Weak entities and attributes

### Shipping Address

- Street
- City
- State
- Zip\_code

### Product Features

- ProductName
- ProductValue
- ProductColor
- ProductSize

## Chapter 2 – Conceptual Design

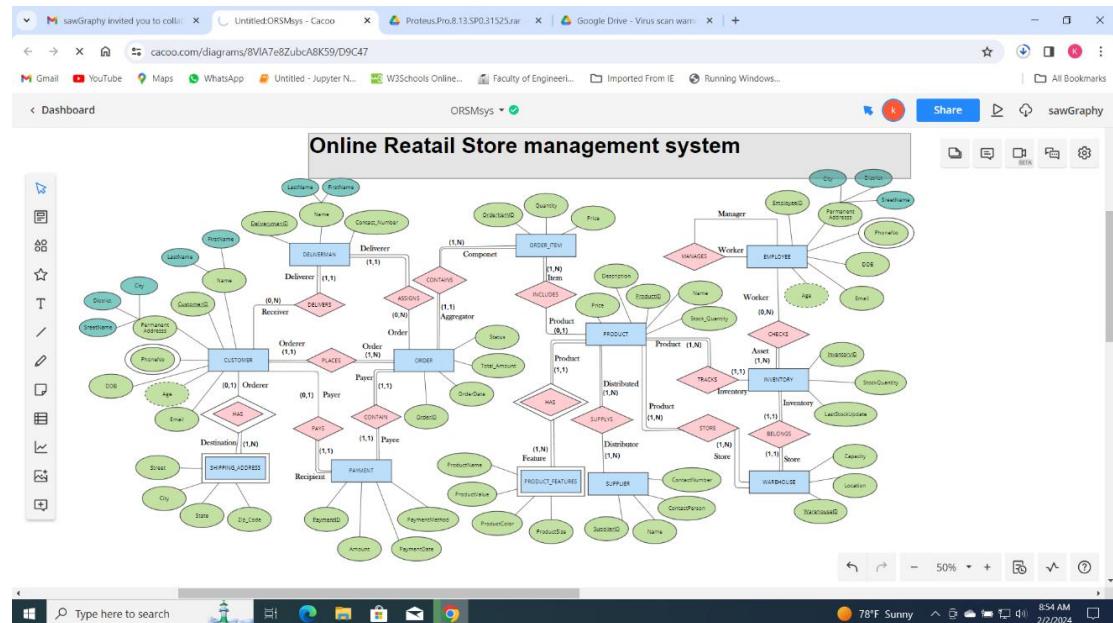


Figure 1. Entity - relationship diagram

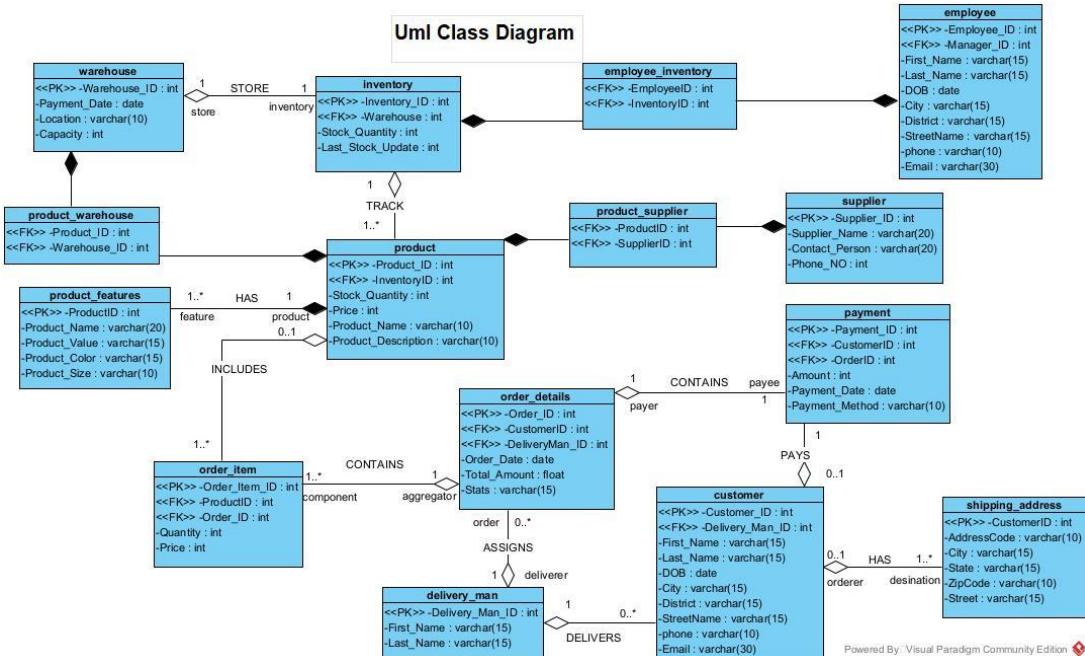
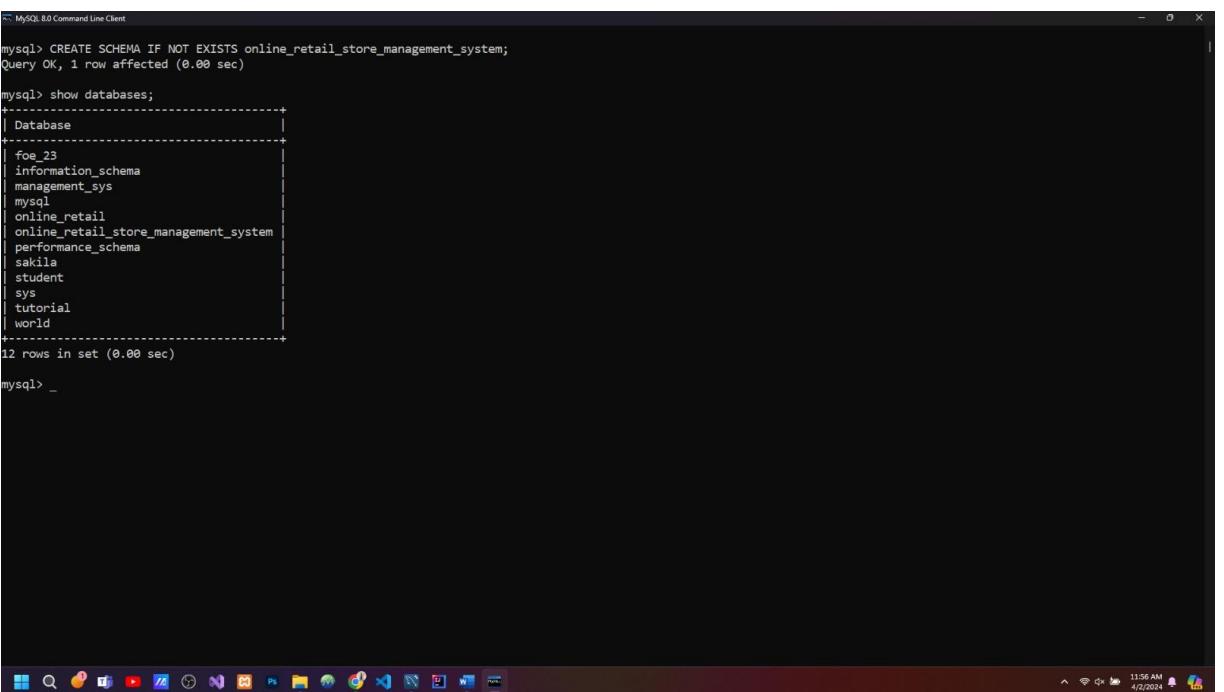


Figure 2. UML class diagram

# Chapter 3 – Implementation

## 3.1 Creating operation

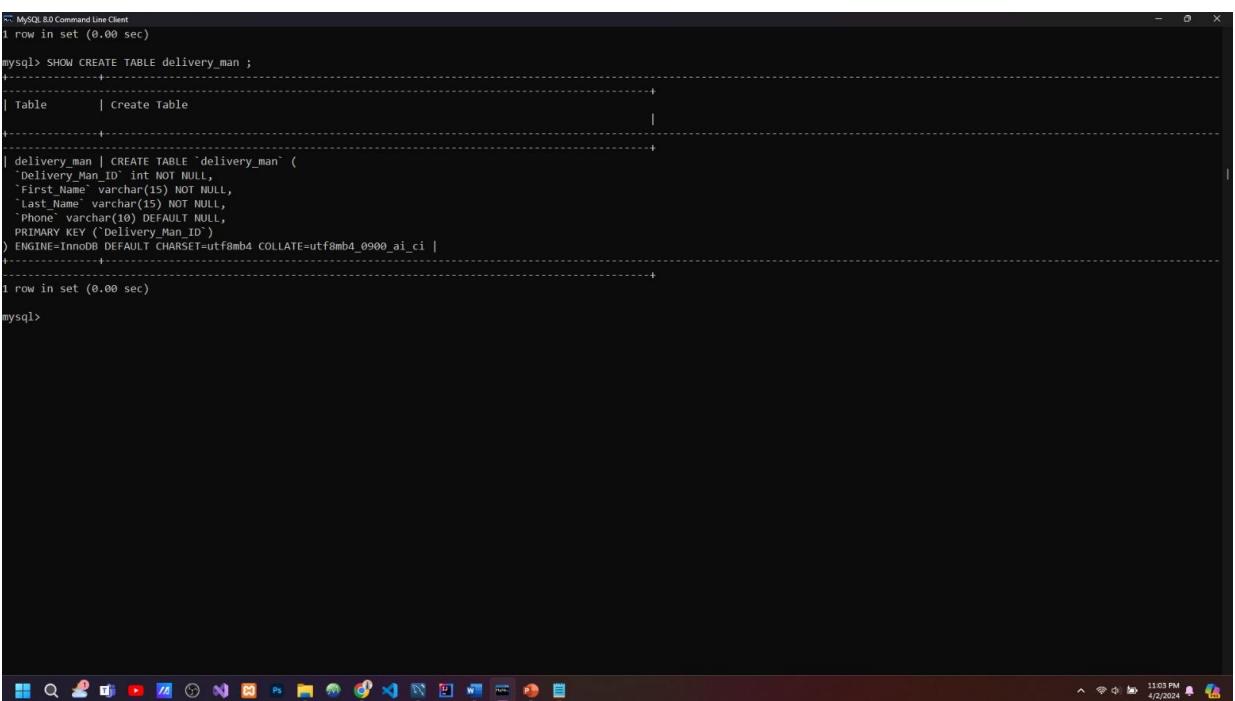


```
-- MySQL 8.0 Command Line Client
mysql> CREATE SCHEMA IF NOT EXISTS online_retail_store_management_system;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| foe_23   |
| information_schema |
| management_sys |
| mysql      |
| online_retail |
| online_retail_store_management_system |
| performance_schema |
| sakila     |
| student    |
| sys        |
| tutorial   |
| world      |
+-----+
12 rows in set (0.00 sec)

mysql> _
```

Figure 3. Creating database



```
-- MySQL 8.0 Command Line Client
1 row in set (0.00 sec)

mysql> SHOW CREATE TABLE delivery_man ;
+-----+
| Table      | Create Table
+-----+
| delivery_man | CREATE TABLE `delivery_man` (
  `Delivery_Man_ID` int NOT NULL,
  `First Name` varchar(15) NOT NULL,
  `Last Name` varchar(15) NOT NULL,
  `Phone` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`Delivery_Man_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+
1 row in set (0.00 sec)

mysql>
```

Figure 4. Creating delivery\_man table

```
MySQL 8.0 Command Line Client

mysql> SHOW CREATE TABLE customer;
+-----+-----+
| Table | Create Table |
+-----+-----+
| customer | CREATE TABLE `customer` (
  `Customer_ID` int NOT NULL,
  `First_Name` varchar(15) NOT NULL,
  `Last_Name` varchar(15) NOT NULL,
  `DOB` date DEFAULT NULL,
  `Deliver_Man_ID` int NOT NULL,
  PRIMARY KEY (`Customer_ID`),
  KEY `fk_deliver` (`Deliver_Man_ID`),
  CONSTRAINT `fk_deliver` FOREIGN KEY (`Deliver_Man_ID`) REFERENCES `delivery_man` (`Delivery_Man_ID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

Figure 5. Creating customer table

```
MySQL 8.0 Command Line Client

mysql> SHOW CREATE TABLE order_details;
+-----+-----+
| Table | Create Table |
+-----+-----+
| order_details | CREATE TABLE `order_details` (
  `Order_ID` int NOT NULL,
  `Order_Date` date NOT NULL,
  `Total_Amount` float DEFAULT NULL,
  `CustomerID` int NOT NULL,
  `DeliverMan_ID` int NOT NULL,
  `Status` varchar(15) DEFAULT NULL,
  PRIMARY KEY (`Order_ID`),
  KEY `fk_deliver1` (`DeliverMan_ID`),
  KEY `fk_customer` (`CustomerID`),
  CONSTRAINT `fk_customer` FOREIGN KEY (`CustomerID`) REFERENCES `customer` (`Customer_ID`) ON DELETE CASCADE,
  CONSTRAINT `fk_deliver1` FOREIGN KEY (`DeliverMan_ID`) REFERENCES `delivery_man` (`Delivery_Man_ID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Figure 6. creating order\_details table

```
-- MySQL 8.0 Command Line Client
mysql> SHOW CREATE TABLE customer_email ;
+-----+-----+
| Table | Create Table |
+-----+-----+
| customer_email | CREATE TABLE `customer_email` (
  `Email` varchar(30) NOT NULL,
  `City` varchar(15) DEFAULT NULL,
  `District` varchar(15) DEFAULT NULL,
  `Street_Name` varchar(15) DEFAULT NULL,
  `Phone` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`Email`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

Figure 7. creating customer\_email table

```
-- MySQL 8.0 Command Line Client
mysql> SHOW CREATE TABLE shipping_address ;
+-----+-----+
| Table | Create Table |
+-----+-----+
| shipping_address | CREATE TABLE `shipping_address` (
  `CustomerID` int NOT NULL,
  `Addresscode` varchar(10) NOT NULL,
  `City` varchar(15) DEFAULT NULL,
  `State` varchar(15) DEFAULT NULL,
  `Zipcode` varchar(10) DEFAULT NULL,
  `Street` varchar(15) DEFAULT NULL,
  PRIMARY KEY (`CustomerID`),
  CONSTRAINT `fk_customer` FOREIGN KEY (`CustomerID`) REFERENCES `customer` (`Customer_ID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Figure 8. creating shipping\_address table

```
-- MySQL 8.0 Command Line Client
mysql> SHOW CREATE TABLE payment;
+-----+-----+
| Table | Create Table
+-----+-----+
| payment | CREATE TABLE `payment` (
  `Payment_ID` int NOT NULL,
  `CustomerID` int NOT NULL,
  `OrderID` int NOT NULL,
  `Amount` int DEFAULT NULL,
  `Payment_Date` date DEFAULT NULL,
  `Payment_Method` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`Payment_ID`),
  KEY `fk_order` (`OrderID`),
  KEY `fk_customer2` (`CustomerID`),
  CONSTRAINT `fk_customer2` FOREIGN KEY (`CustomerID`) REFERENCES `customer` (`Customer_ID`) ON DELETE CASCADE,
  CONSTRAINT `fk_order` FOREIGN KEY (`OrderID`) REFERENCES `order_details` (`Order_ID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Figure 9. creating payment table

```
-- MySQL 8.0 Command Line Client
1 row in set (0.00 sec)
mysql> SHOW CREATE TABLE warehouse;
+-----+-----+
| Table | Create Table
+-----+-----+
| warehouse | CREATE TABLE `warehouse` (
  `Warehouse_ID` int NOT NULL,
  `Payment_Date` date DEFAULT NULL,
  `Location` varchar(10) DEFAULT NULL,
  `Capacity` int DEFAULT NULL,
  PRIMARY KEY (`Warehouse_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

Figure 10. creating warehouse table

```
-- MySQL 8.0 Command Line Client
1 row in set (0.00 sec)

mysql> SHOW CREATE TABLE inventory;
+-----+-----+
| Table | Create Table
+-----+-----+
| inventory | CREATE TABLE `inventory` (
  `Inventory_ID` int NOT NULL,
  `WarehouseID` int NOT NULL,
  `Stock_Quantity` int DEFAULT NULL,
  `Last_Stock_Update` int DEFAULT NULL,
  PRIMARY KEY (`Inventory_ID`),
  KEY `fk_whouse` (`WarehouseID`),
  CONSTRAINT `fk_whouse` FOREIGN KEY (`WarehouseID`) REFERENCES `warehouse` (`Warehouse_ID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Figure 11. creating inventory table

```
-- MySQL 8.0 Command Line Client
mysql> SHOW CREATE TABLE product;
+-----+-----+
| Table | Create Table
+-----+-----+
| product | CREATE TABLE `product` (
  `Product_ID` int NOT NULL,
  `InventoryID` int NOT NULL,
  `Stock_Quantity` int DEFAULT NULL,
  `Price` int DEFAULT NULL,
  `Product_Name` varchar(20) DEFAULT NULL,
  `Product_Description` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`Product_ID`),
  KEY `fk_invent` (`InventoryID`),
  CONSTRAINT `fk_invent` FOREIGN KEY (`InventoryID`) REFERENCES `inventory` (`Inventory_ID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Figure 12. creating product table

```
-- MySQL 8.0 Command Line Client
1 row in set (0.00 sec)

mysql> SHOW CREATE TABLE order_item ;
+-----+-----+
| Table | Create Table |
+-----+-----+
| order_item | CREATE TABLE `order_item` (
  `Order_Item_ID` int NOT NULL,
  `ProductID` int NOT NULL,
  `OrderID` int NOT NULL,
  `Quantity` int DEFAULT NULL,
  `Price` int DEFAULT NULL,
  PRIMARY KEY (`Order_Item_ID`),
  KEY `fk_product` (`ProductID`),
  KEY `fk_order1` (`OrderID`),
  CONSTRAINT `fk_order1` FOREIGN KEY (`OrderID`) REFERENCES `order_details` (`Order_ID`) ON DELETE CASCADE,
  CONSTRAINT `fk_product` FOREIGN KEY (`ProductID`) REFERENCES `product` (`Product_ID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Figure 13. creating order\_item table

```
-- MySQL 8.0 Command Line Client
1 row in set (0.00 sec)

mysql> SHOW CREATE TABLE supplier;
+-----+-----+
| Table | Create Table |
+-----+-----+
| supplier | CREATE TABLE `supplier` (
  `Supplier_ID` int NOT NULL,
  `Supplier_Name` varchar(20) DEFAULT NULL,
  `Contact_Person` varchar(20) DEFAULT NULL,
  `Phone_NO` int DEFAULT NULL,
  PRIMARY KEY (`Supplier_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

Figure 14. creating supplier table

```
-- MySQL 8.0 Command Line Client
mysql> SHOW CREATE TABLE product_supplier;
+-----+-----+
| Table | Create Table |
+-----+-----+
| product_supplier | CREATE TABLE `product_supplier` (
  `ProductID` int NOT NULL,
  `SupplierID` int NOT NULL,
  PRIMARY KEY (`ProductID`,`SupplierID`),
  KEY `fk_supplier` (`SupplierID`),
  CONSTRAINT `fk_product` FOREIGN KEY (`ProductID`) REFERENCES `product` (`Product ID`) ON DELETE CASCADE,
  CONSTRAINT `fk_supplier` FOREIGN KEY (`SupplierID`) REFERENCES `supplier` (`Supplier_ID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)

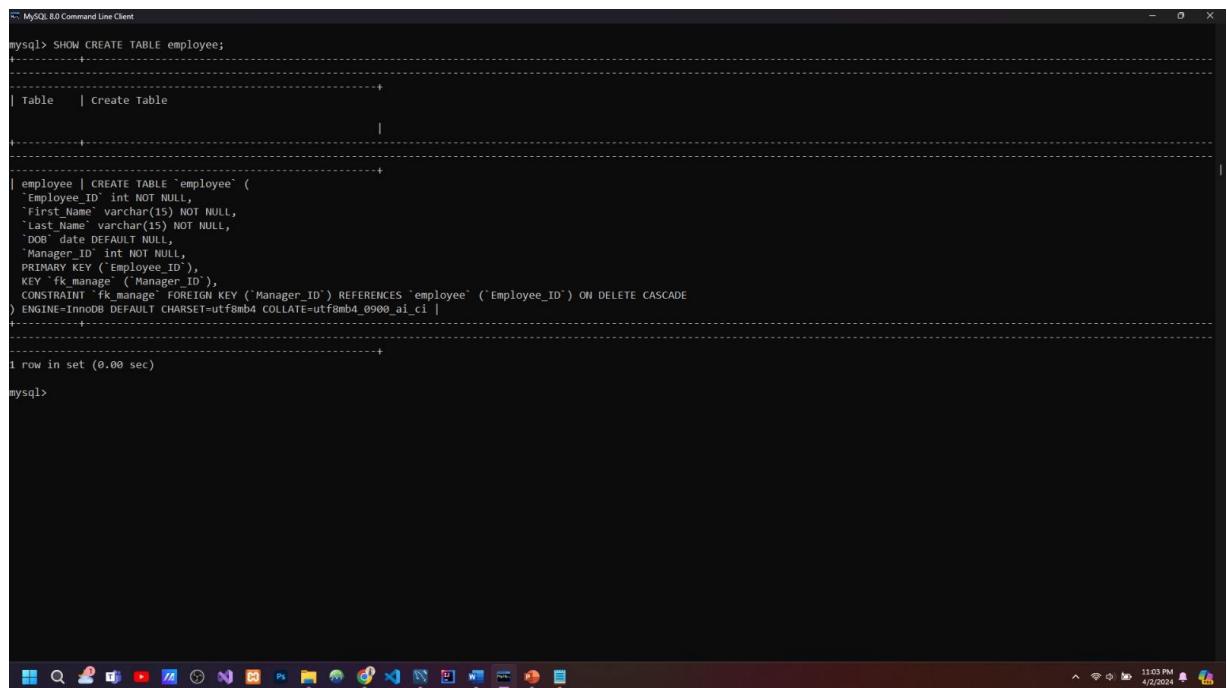
mysql> _
```

Figure 15. creating product\_supplier table

```
-- MySQL 8.0 Command Line Client
mysql> SHOW CREATE TABLE product_warehouse ;
+-----+-----+
| Table | Create Table |
+-----+-----+
| product_warehouse | CREATE TABLE `product_warehouse` (
  `ProductID` int NOT NULL,
  `WarehouseID` int NOT NULL,
  PRIMARY KEY (`ProductID`, `WarehouseID`),
  KEY `fk_whouse1` (`WarehouseID`),
  CONSTRAINT `fk_product2` FOREIGN KEY (`ProductID`) REFERENCES `product` (`Product ID`) ON DELETE CASCADE,
  CONSTRAINT `fk_whouse1` FOREIGN KEY (`WarehouseID`) REFERENCES `warehouse` (`Warehouse_ID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

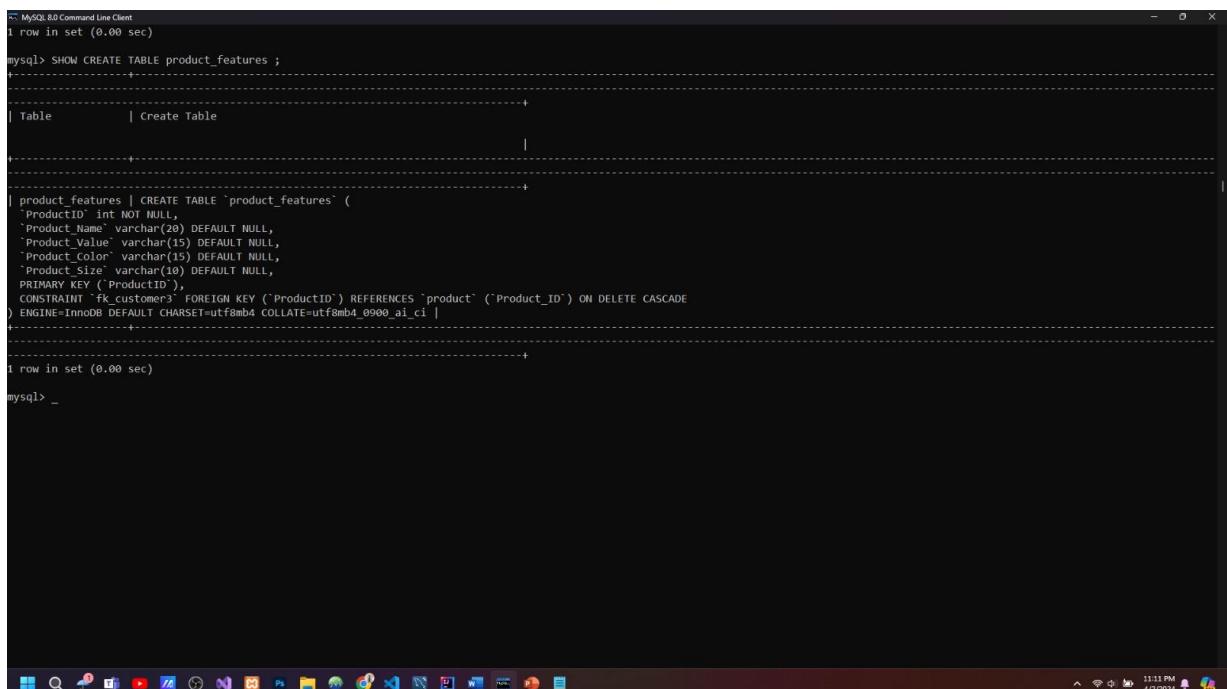
Figure 16. creating product\_warehouse table



```
MySQL> SHOW CREATE TABLE employee;
+-----+-----+
| Table | Create Table |
+-----+-----+
| employee | CREATE TABLE `employee` (
  `Employee_ID` int NOT NULL,
  `First_Name` varchar(15) NOT NULL,
  `Last_Name` varchar(15) NOT NULL,
  `DOB` date DEFAULT NULL,
  `Manager_ID` int NOT NULL,
  PRIMARY KEY (`Employee_ID`),
  KEY `fk_manager` (`Manager_ID`),
  CONSTRAINT `fk_manager` FOREIGN KEY (`Manager_ID`) REFERENCES `employee` (`Employee_ID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)

MySQL>
```

Figure 17. creating employee table



```
MySQL> SHOW CREATE TABLE product_features ;
+-----+-----+
| Table | Create Table |
+-----+-----+
| product_features | CREATE TABLE `product_features` (
  `ProductID` int NOT NULL,
  `Product_Name` varchar(20) DEFAULT NULL,
  `Product_Value` varchar(15) DEFAULT NULL,
  `Product_Color` varchar(15) DEFAULT NULL,
  `Product_Size` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`ProductID`),
  CONSTRAINT `fk_customer3` FOREIGN KEY (`ProductID`) REFERENCES `product` (`Product_ID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)

MySQL> _
```

Figure 18. creating product\_features table

```
MySQL> SHOW CREATE TABLE employee_inventory ;
+-----+-----+
| Table | Create Table |
+-----+-----+
| employee_inventory | CREATE TABLE `employee_inventory` (
  `EmployeeID` int NOT NULL,
  `InventoryID` int NOT NULL,
  PRIMARY KEY (`EmployeeID`, `InventoryID`),
  KEY `fk_inven` (`InventoryID`),
  CONSTRAINT `fk_employee` FOREIGN KEY (`EmployeeID`) REFERENCES `employee` (`Employee_ID`) ON DELETE CASCADE,
  CONSTRAINT `fk_inven` FOREIGN KEY (`InventoryID`) REFERENCES `inventory` (`Inventory_ID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

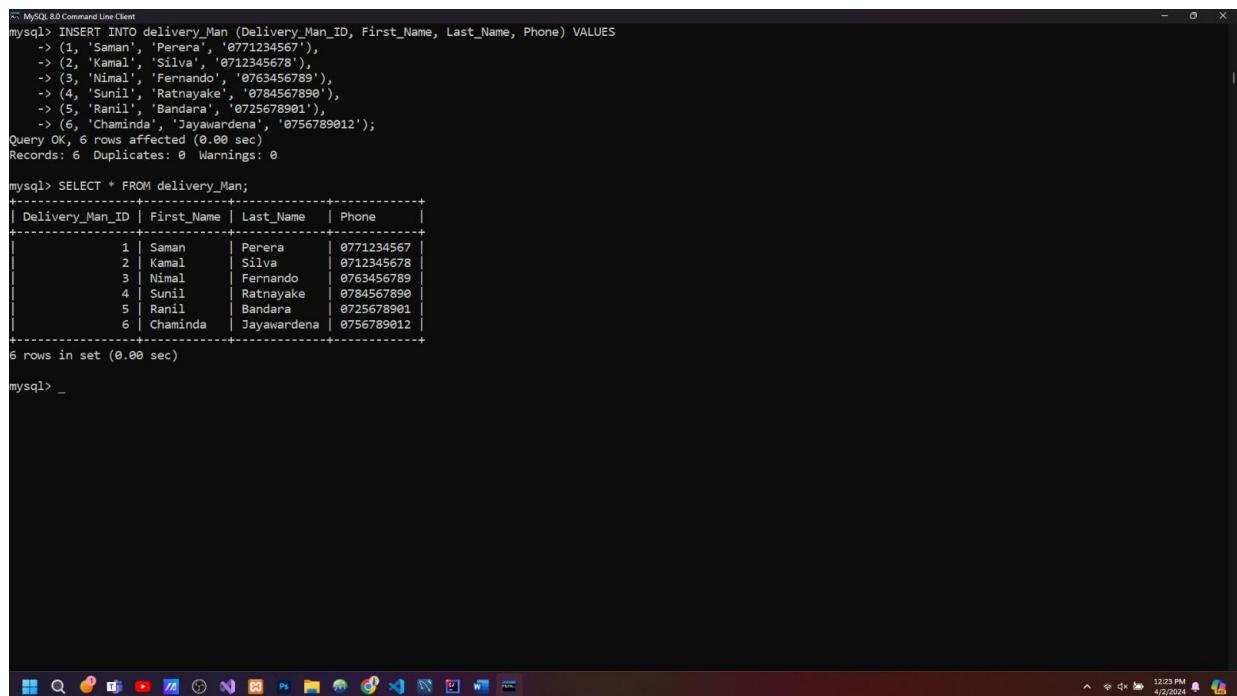
Figure 19. creating employee\_inventory table

```
MySQL> SHOW CREATE TABLE employee_email ;
+-----+-----+
| Table | Create Table |
+-----+-----+
| employee_email | CREATE TABLE `employee_email` (
  `Email` varchar(30) NOT NULL,
  `City` varchar(15) DEFAULT NULL,
  `District` varchar(15) DEFAULT NULL,
  `Street_Name` varchar(15) DEFAULT NULL,
  `Phone` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`Email`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

Figure 20. creating employee\_email table

### 3.2 Inserting operation

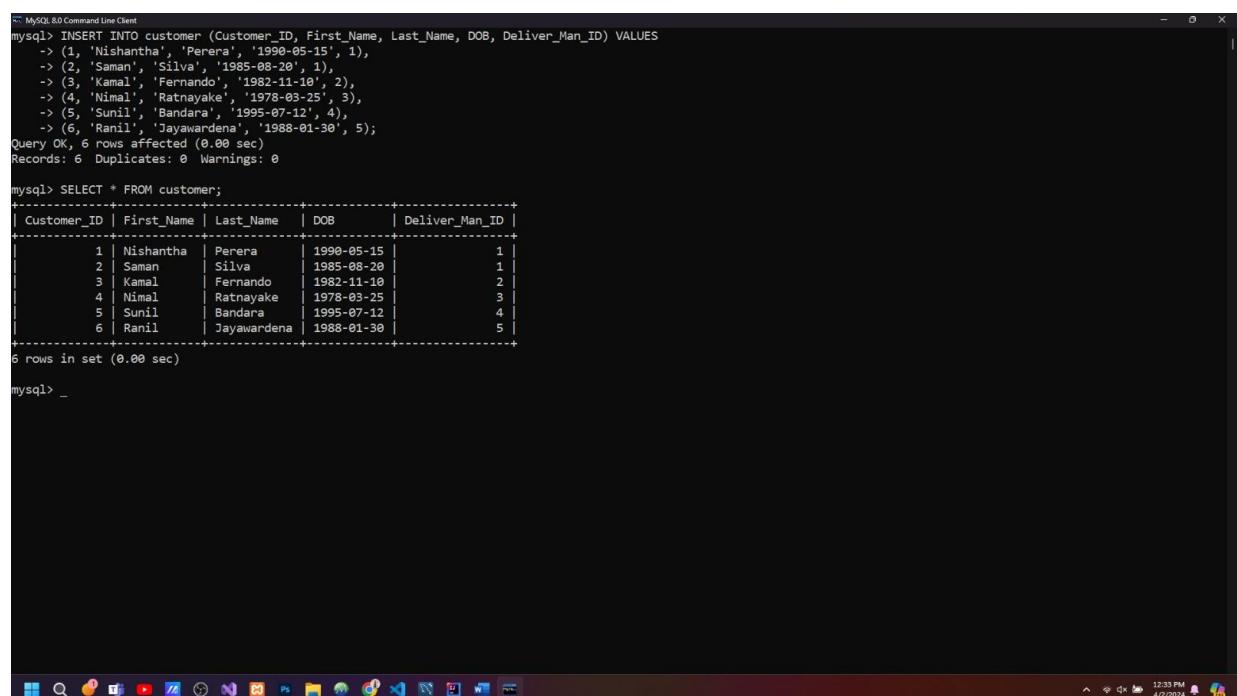


```
mysql> INSERT INTO delivery_Man (Delivery_Man_ID, First_Name, Last_Name, Phone) VALUES
-> (1, 'Saman', 'Perera', '0771234567'),
-> (2, 'Kamal', 'Silva', '0712345678'),
-> (3, 'Nimal', 'Fernando', '0763456789'),
-> (4, 'Sunil', 'Ratnayake', '0784567890'),
-> (5, 'Ranil', 'Bandara', '0725678901'),
-> (6, 'Chaminda', 'Jayawardena', '0756789012');
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM delivery_Man;
+-----+-----+-----+
| Delivery_Man_ID | First_Name | Last_Name | Phone |
+-----+-----+-----+
| 1 | Saman | Perera | 0771234567 |
| 2 | Kamal | Silva | 0712345678 |
| 3 | Nimal | Fernando | 0763456789 |
| 4 | Sunil | Ratnayake | 0784567890 |
| 5 | Ranil | Bandara | 0725678901 |
| 6 | Chaminda | Jayawardena | 0756789012 |
+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> _
```

Figure 21. Inserting delivery\_man table



```
mysql> INSERT INTO customer (Customer_ID, First_Name, Last_Name, DOB, Deliver_Man_ID) VALUES
-> (1, 'Nishantha', 'Perera', '1990-05-15', 1),
-> (2, 'Saman', 'Silva', '1985-08-20', 1),
-> (3, 'Kamal', 'Fernando', '1982-11-10', 2),
-> (4, 'Nimal', 'Ratnayake', '1978-03-25', 3),
-> (5, 'Sunil', 'Bandara', '1995-07-12', 4),
-> (6, 'Ranil', 'Jayawardena', '1988-01-30', 5);
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM customer;
+-----+-----+-----+-----+
| Customer_ID | First_Name | Last_Name | DOB | Deliver_Man_ID |
+-----+-----+-----+-----+
| 1 | Nishantha | Perera | 1990-05-15 | 1 |
| 2 | Saman | Silva | 1985-08-20 | 1 |
| 3 | Kamal | Fernando | 1982-11-10 | 2 |
| 4 | Nimal | Ratnayake | 1978-03-25 | 3 |
| 5 | Sunil | Bandara | 1995-07-12 | 4 |
| 6 | Ranil | Jayawardena | 1988-01-30 | 5 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> _
```

Figure 22. Inserting customer table

```

MySQL> INSERT INTO order_details (Order_ID, Order_Date, Total_Amount, CustomerID, DeliverMan_ID, Stats) VALUES
-> (1, '2024-03-01', 2500.50, 1, 1, 'delivered'),
-> (2, '2024-03-05', 1500.25, 2, 1, 'pending'),
-> (3, '2024-03-10', 3200.75, 3, 2, 'delivered'),
-> (4, '2024-03-15', 1800.30, 4, 3, 'delivered'),
-> (5, '2024-03-20', 2000.00, 5, 4, 'pending'),
-> (6, '2024-03-25', 2800.80, 6, 5, 'delivered');
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

MySQL> SELECT * FROM order_details;
+-----+-----+-----+-----+-----+-----+
| Order_ID | Order_Date | Total_Amount | CustomerID | DeliverMan_ID | Stats
+-----+-----+-----+-----+-----+-----+
| 1 | 2024-03-01 | 2500.5 | 1 | 1 | delivered
| 2 | 2024-03-05 | 1500.25 | 2 | 1 | pending
| 3 | 2024-03-10 | 3200.75 | 3 | 2 | delivered
| 4 | 2024-03-15 | 1800.3 | 4 | 3 | delivered
| 5 | 2024-03-20 | 2000.0 | 5 | 4 | pending
| 6 | 2024-03-25 | 2800.8 | 6 | 5 | delivered
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

MySQL> truncate online_retail_store_management_system.order_details;
ERROR 1701 (42000): Cannot truncate a table referenced in a foreign key constraint ('online_retail_store_management_system`.`payment`, CONSTRAINT `fk_order`)
MySQL>

```

Figure 23. Inserting order\_details table

```

MySQL>
MySQL>
mysql> INSERT INTO customer_email (Email, City, District, Street_Name, Phone) VALUES
-> ('nishantha@gmail.com', 'Colombo', 'Colombo', 'Galle Road', '0711112222'),
-> ('samar@gmail.com', 'Kandy', 'Kandy', 'Peradeniya Road', '0772223333'),
-> ('kamal@gmail.com', 'Galle', 'Galle', 'Fort Road', '0763334444'),
-> ('nimal@gmail.com', 'Jaffna', 'Jaffna', 'Main Street', '0754445555'),
-> ('sunil@gmail.com', 'Matara', 'Matara', 'Beach Road', '0725556666'),
-> ('ranil@gmail.com', 'Kurunegala', 'Kurunegala', 'Station Road', '0786667777');
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

MySQL> SELECT * FROM customer_email;
+-----+-----+-----+-----+-----+
| Email | City | District | Street_Name | Phone
+-----+-----+-----+-----+-----+
| kama1@gmail.com | Galle | Galle | Fort Road | 0763334444 |
| nimal@gmail.com | Jaffna | Jaffna | Main Street | 0754445555 |
| nishantha@gmail.com | Colombo | Colombo | Galle Road | 0711112222 |
| ranil@gmail.com | Kurunegala | Kurunegala | Station Road | 0786667777 |
| samar@gmail.com | Kandy | Kandy | Peradeniya Road | 0772223333 |
| sunil@gmail.com | Matara | Matara | Beach Road | 0725556666 |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

MySQL> _

```

Figure 24. Inserting customer\_email table

```

MySQL> INSERT INTO shipping_address (CustomerID, AddressCode, City, State, ZipCode, Street) VALUES
-> (1, '00100', 'Colombo', 'Western', '00100', 'Galle Road'),
-> (2, '20000', 'Kandy', 'Central', '20000', 'Peradeniya Road'),
-> (3, '80000', 'Galle', 'Southern', '80000', 'Fort Road'),
-> (4, '40000', 'Jaffna', 'Northern', '40000', 'Main Street'),
-> (5, '90000', 'Matara', 'Southern', '90000', 'Beach Road'),
-> (6, '60000', 'Kurunegala', 'North Western', '60000', 'Station Road');
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

MySQL> SELECT * FROM shipping_address;
+-----+-----+-----+-----+-----+-----+
| CustomerID | AddressCode | City      | State    | ZipCode | Street
+-----+-----+-----+-----+-----+-----+
| 1 | 00100 | Colombo | Western | 00100 | Galle Road
| 2 | 20000 | Kandy   | Central | 20000 | Peradeniya Road
| 3 | 80000 | Galle   | Southern | 80000 | Fort Road
| 4 | 40000 | Jaffna  | Northern | 40000 | Main Street
| 5 | 90000 | Matara  | Southern | 90000 | Beach Road
| 6 | 60000 | Kurunegala | North Western | 60000 | Station Road
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

MySQL> SELECT * FROM shipping_address;
+-----+-----+-----+-----+-----+-----+
| CustomerID | AddressCode | City      | State    | ZipCode | Street
+-----+-----+-----+-----+-----+-----+
| 1 | 00100 | Colombo | Western | 00100 | Galle Road
| 2 | 20000 | Kandy   | Central | 20000 | Peradeniya Road
| 3 | 80000 | Galle   | Southern | 80000 | Fort Road
| 4 | 40000 | Jaffna  | Northern | 40000 | Main Street
| 5 | 90000 | Matara  | Southern | 90000 | Beach Road
| 6 | 60000 | Kurunegala | North Western | 60000 | Station Road
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

MySQL> _

```

Figure 25. Inserting shipping\_address table

```

MySQL> INSERT INTO payment (Payment_ID, CustomerID, OrderID, Amount, Payment_Date, Payment_Method) VALUES
-> (1, 1, 1, 2500, '2024-03-02', 'Cash'),
-> (2, 1, 6, 1500, '2024-03-06', 'Credit Card'),
-> (3, 3, 3, 3200, '2024-03-11', 'Cash'),
-> (4, 2, 6, 1800, '2024-03-16', 'Credit Card'),
-> (5, 5, 5, 2000, '2024-03-21', 'Cash'),
-> (6, 6, 6, 2800, '2024-03-26', 'Credit Card');
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

MySQL> SELECT * FROM payment;
+-----+-----+-----+-----+-----+-----+
| Payment_ID | CustomerID | OrderID | Amount | Payment_Date | Payment_Method |
+-----+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 2500 | 2024-03-02 | Cash
| 2 | 1 | 6 | 1500 | 2024-03-06 | Credit Card
| 3 | 3 | 3 | 3200 | 2024-03-11 | Cash
| 4 | 2 | 6 | 1800 | 2024-03-16 | Credit Card
| 5 | 5 | 5 | 2000 | 2024-03-21 | Cash
| 6 | 6 | 6 | 2800 | 2024-03-26 | Credit Card
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

MySQL> _

```

Figure 26. Inserting payment table

```
MySQL> INSERT INTO warehouse (Warehouse_ID, Payment_Date, Location, Capacity) VALUES
-> (1, '2024-03-01', 'Colombo', 5000),
-> (2, '2024-03-05', 'Kandy', 3000),
-> (3, '2024-03-10', 'Galle', 4000),
-> (4, '2024-03-15', 'Jaffna', 2000),
-> (5, '2024-03-20', 'Matara', 3500),
-> (6, '2024-03-25', 'Kurunegala', 2500);
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0

MySQL> SELECT * FROM warehouse;
+-----+-----+-----+-----+
| Warehouse_ID | Payment_Date | Location | Capacity |
+-----+-----+-----+-----+
| 1 | 2024-03-01 | Colombo | 5000 |
| 2 | 2024-03-05 | Kandy | 3000 |
| 3 | 2024-03-10 | Galle | 4000 |
| 4 | 2024-03-15 | Jaffna | 2000 |
| 5 | 2024-03-20 | Matara | 3500 |
| 6 | 2024-03-25 | Kurunegala | 2500 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

MySQL>
```

Figure 27. Inserting warehouse table

```
MySQL> INSERT INTO inventory (Inventory_ID, WarehouseID, Stock_Quantity, Last_Stock_Update) VALUES
-> (1, 1, 1000, UNIX_TIMESTAMP('2024-03-01')),
-> (2, 1, 800, UNIX_TIMESTAMP('2024-03-05')),
-> (3, 3, 1200, UNIX_TIMESTAMP('2024-03-10')),
-> (4, 2, 600, UNIX_TIMESTAMP('2024-03-15')),
-> (5, 5, 900, UNIX_TIMESTAMP('2024-03-20')),
-> (6, 6, 1100, UNIX_TIMESTAMP('2024-03-25'));
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0

MySQL> SELECT * FROM inventory;
+-----+-----+-----+-----+
| Inventory_ID | WarehouseID | Stock_Quantity | Last_Stock_Update |
+-----+-----+-----+-----+
| 1 | 1 | 1000 | 1709231000 |
| 2 | 1 | 800 | 1709577000 |
| 3 | 3 | 1200 | 1710089000 |
| 4 | 2 | 600 | 1710441000 |
| 5 | 5 | 900 | 1710873000 |
| 6 | 6 | 1100 | 1711305000 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

MySQL> _
```

Figure 28. Inserting inventory table

```
-- MySQL 8.0 Command Line Client
6 rows in set (0.00 sec)

mysql> INSERT INTO product (Product_ID, InventoryID, Stock_Quantity, Price, Product_Name, Product_Description) VALUES
-> (1, 1, 1000, 150, 'Rice', 'High-quality rice from local farms'),
-> (2, 2, 800, 250, 'Tea', 'Premium Ceylon tea'),
-> (3, 1, 1200, 300, 'Spices', 'Assorted spices for cooking'),
-> (4, 4, 600, 200, 'Coconut', 'Organic coconut oil'),
-> (5, 2, 900, 180, 'Apparel', 'Traditional Sri Lankan clothing'),
-> (6, 6, 1100, 350, 'Handicraft', 'Handmade crafts from local artisans');
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> SELECT * FROM product;
+-----+-----+-----+-----+-----+-----+
| Product_ID | InventoryID | Stock_Quantity | Price | Product_Name | Product_Description |
+-----+-----+-----+-----+-----+-----+
| 1 | 1 | 1000 | 150 | Rice | High-quality rice from local farms |
| 2 | 2 | 800 | 250 | Tea | Premium Ceylon tea |
| 3 | 1 | 1200 | 300 | Spices | Assorted spices for cooking |
| 4 | 4 | 600 | 200 | Coconut | Organic coconut oil |
| 5 | 2 | 900 | 180 | Apparel | Traditional Sri Lankan clothing |
| 6 | 6 | 1100 | 350 | Handicraft | Handmade crafts from local artisans |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> _
```

Figure 29. Inserting product table

```
-- MySQL 8.0 Command Line Client
mysql> INSERT INTO order_item (Order_Item_ID, ProductID, OrderID, Quantity, Price) VALUES
-> (1, 1, 1, 2, 300),
-> (2, 2, 2, 3, 750),
-> (3, 3, 3, 1, 300),
-> (4, 1, 2, 2, 300),
-> (5, 2, 3, 4, 720),
-> (6, 6, 6, 1, 350);
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> SELECT * FROM order_item;
+-----+-----+-----+-----+-----+
| Order_Item_ID | ProductID | OrderID | Quantity | Price |
+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 2 | 300 |
| 2 | 2 | 2 | 3 | 750 |
| 3 | 3 | 3 | 1 | 300 |
| 4 | 1 | 2 | 2 | 300 |
| 5 | 2 | 3 | 4 | 720 |
| 6 | 6 | 6 | 1 | 350 |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Figure 30. Inserting order\_item table

```
-- MySQL 8.0 Command Line Client
6 rows in set (0.00 sec)

mysql> INSERT INTO supplier (Supplier_ID, Supplier_Name, Contact_Person, Phone_NO) VALUES
-> (1, 'ABC Suppliers', 'John Doe', '071112222'),
-> (2, 'XYZ Suppliers', 'Jane Smith', '0762223333'),
-> (3, 'PQR Suppliers', 'David Brown', '0753334444'),
-> (4, 'LMN Suppliers', 'Emma Johnson', '0784445555'),
-> (5, 'DEF Suppliers', 'Michael Williams', '0715556666'),
-> (6, 'GHI Suppliers', 'Sophia Brown', '0726667777');
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM supplier;
+-----+-----+-----+-----+
| Supplier_ID | Supplier_Name | Contact_Person | Phone_NO |
+-----+-----+-----+-----+
| 1 | ABC Suppliers | John Doe | 771112222 |
| 2 | XYZ Suppliers | Jane Smith | 762223333 |
| 3 | PQR Suppliers | David Brown | 753334444 |
| 4 | LMN Suppliers | Emma Johnson | 784445555 |
| 5 | DEF Suppliers | Michael Williams | 715556666 |
| 6 | GHI Suppliers | Sophia Brown | 726667777 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Figure 31. Inserting supplier table

```
-- MySQL 8.0 Command Line Client
mysql> INSERT INTO product_supplier (ProductID, SupplierID) VALUES
-> (1, 1),
-> (2, 3),
-> (3, 3),
-> (4, 4),
-> (5, 1),
-> (6, 6);
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM product_supplier;
+-----+-----+
| ProductID | SupplierID |
+-----+-----+
| 1 | 1 |
| 5 | 1 |
| 2 | 3 |
| 3 | 3 |
| 4 | 4 |
| 6 | 6 |
+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Figure 32. Inserting product\_supplier table

```
MySQL 8.0 Command Line Client
mysql> INSERT INTO product_warehouse (ProductID, WarehouseID) VALUES
-> (1, 1),
-> (2, 1),
-> (3, 1),
-> (4, 2),
-> (5, 2),
-> (6, 2);
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM product_warehouse;
+-----+-----+
| ProductID | WarehouseID |
+-----+-----+
|       1    |        1     |
|       2    |        1     |
|       3    |        1     |
|       4    |        2     |
|       5    |        2     |
|       6    |        2     |
+-----+-----+
6 rows in set (0.00 sec)

mysql> _
```

Figure 33. Inserting product\_warehouse table

```
MySQL 8.0 Command Line Client
mysql> INSERT INTO employee (Employee_ID, First_Name, Last_Name, DOB, Manager_ID) VALUES
-> (1, 'Kasun', 'Perera', '1988-05-15', 1),
-> (2, 'Nimal', 'Silva', '1975-08-20', 1),
-> (3, 'Sunil', 'Fernando', '1972-11-10', 2),
-> (4, 'Ranil', 'Ratnayake', '1968-03-25', 2),
-> (5, 'Chaminda', 'Bandara', '1985-07-12', 1),
-> (6, 'Kamal', 'Jayawardena', '1978-01-30', 1);
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM employee;
+-----+-----+-----+-----+-----+
| Employee_ID | First_Name | Last_Name | DOB      | Manager_ID |
+-----+-----+-----+-----+-----+
|       1    |   Kasun   |   Perera  | 1988-05-15 |        1    |
|       2    |   Nimal   |   Silva   | 1975-08-20 |        1    |
|       3    |   Sunil   |   Fernando | 1972-11-10 |        2    |
|       4    |   Ranil   |   Ratnayake | 1968-03-25 |        2    |
|       5    |   Chaminda |   Bandara  | 1985-07-12 |        1    |
|       6    |   Kamal   |   Jayawardena | 1978-01-30 |        1    |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Figure 34. Inserting employee table

```
-- MySQL 8.0 Command Line Client
mysql> INSERT INTO product_features (ProductID, Product_Name, Product_Value, Product_Color, Product_Size) VALUES
-> (1, 'Rice', 'Premium', 'White', '5kg'),
-> (2, 'Tea', 'Premium', 'Black', '200g'),
-> (3, 'Spices', 'Assorted', 'Various', NULL),
-> (4, 'Coconut', 'Organic', 'Brown', NULL),
-> (5, 'Apparel', 'Traditional', 'Various', NULL),
-> (6, 'Handicraft', 'Artisanal', 'Various', NULL);
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> SELECT * FROM product_features;
+-----+-----+-----+-----+
| ProductID | Product_Name | Product_Value | Product_Color | Product_Size |
+-----+-----+-----+-----+
| 1 | Rice | Premium | White | 5kg |
| 2 | Tea | Premium | Black | 200g |
| 3 | Spices | Assorted | Various | NULL |
| 4 | Coconut | Organic | Brown | NULL |
| 5 | Apparel | Traditional | Various | NULL |
| 6 | Handicraft | Artisanal | Various | NULL |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Figure 35. Inserting product\_feature table

```
-- MySQL 8.0 Command Line Client
mysql> INSERT INTO employee_email (Email, City, District, Street_Name, Phone) VALUES
-> ('kasun@example.com', 'Colombo', 'Colombo', 'Galle Road', '0711112222'),
-> ('nimal@example.com', 'Kandy', 'Kandy', 'Peradeniya Road', '0772223333'),
-> ('sunil@example.com', 'Galle', 'Galle', 'Fort Road', '0763334444'),
-> ('ranil@example.com', 'Jaffna', 'Jaffna', 'Main Street', '0754445555'),
-> ('chaminda@example.com', 'Matara', 'Matara', 'Beach Road', '0725556666'),
-> ('kamal@example.com', 'Kurunegala', 'Kurunegala', 'Station Road', '0786667777');
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> SELECT * FROM employee_email;
+-----+-----+-----+-----+
| Email | City | District | Street_Name | Phone |
+-----+-----+-----+-----+
| chaminda@example.com | Matara | Matara | Beach Road | 0725556666 |
| kamal@example.com | Kurunegala | Kurunegala | Station Road | 0786667777 |
| kasun@example.com | Colombo | Colombo | Galle Road | 0711112222 |
| nimal@example.com | Kandy | Kandy | Peradeniya Road | 0772223333 |
| ranil@example.com | Jaffna | Jaffna | Main Street | 0754445555 |
| sunil@example.com | Galle | Galle | Fort Road | 0763334444 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Figure 36. Inserting employee\_email table

```
MySQL> INSERT INTO employee_inventory (EmployeeID, InventoryID) VALUES
-> (1, 1),
-> (2, 1),
-> (3, 3),
-> (4, 4),
-> (5, 2),
-> (6, 6);
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> SELECT * FROM employee_inventory;
+-----+-----+
| EmployeeID | InventoryID |
+-----+-----+
| 1 | 1 |
| 2 | 1 |
| 5 | 2 |
| 3 | 3 |
| 4 | 4 |
| 6 | 6 |
+-----+
6 rows in set (0.00 sec)

mysql> _
```

Figure 37. Inserting employee\_inventory table

### 3.3 Updating operation

```
MySQL> select * from delivery_Man;
+-----+-----+-----+
| Delivery_Man_ID | First_Name | Last_Name | Phone |
+-----+-----+-----+
| 1 | Saman | Perera | 0771234567 |
| 2 | Kamal | Fernando | 0712345678 |
| 3 | Nimal | Fernando | 0763456789 |
| 4 | Sunil | Ratnayake | 0784567890 |
| 5 | Ranil | Bandara | 0725678901 |
| 6 | Chaminda | Jayawardena | 0756789012 |
+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> UPDATE delivery_Man
-> SET Phone = '0715555555'
-> WHERE Delivery_Man_ID = 1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE delivery_Man
-> SET Last_Name = 'Wijesinghe'
-> WHERE Delivery_Man_ID = 2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from delivery_Man;
+-----+-----+-----+
| Delivery_Man_ID | First_Name | Last_Name | Phone |
+-----+-----+-----+
| 1 | Saman | Perera | 0715555555 |
| 2 | Kamal | Wijesinghe | 0712345678 |
| 3 | Nimal | Fernando | 0763456789 |
| 4 | Sunil | Ratnayake | 0784567890 |
| 5 | Ranil | Bandara | 0725678901 |
| 6 | Chaminda | Jayawardena | 0756789012 |
+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Figure 38. Updating delivery\_Man table

```

C:\ MySQL 8.0 Command Line Client
mysql> select * from customer;
+-----+-----+-----+-----+
| Customer_ID | First_Name | Last_Name | DOB       | Deliver_Man_ID |
+-----+-----+-----+-----+
| 1 | Nishantha | Perera    | 1990-05-15 | 1          |
| 2 | Saman     | Silva     | 1985-08-20 | 1          |
| 3 | Kamal     | Fernando   | 1982-11-10 | 2          |
| 4 | Nimal     | Ratnayake  | 1978-03-25 | 3          |
| 5 | Sunil     | Bandara   | 1995-07-12 | 4          |
| 6 | Ranil     | Jayawardena| 1988-01-30 | 5          |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> UPDATE customer
-> SET Deliver_Man_ID = 3
-> WHERE Customer_ID = 1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE customer
-> SET DOB = '1986-09-22'
-> WHERE Customer_ID = 2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from customer;
+-----+-----+-----+-----+
| Customer_ID | First_Name | Last_Name | DOB       | Deliver_Man_ID |
+-----+-----+-----+-----+
| 1 | Nishantha | Perera    | 1990-05-15 | 3          |
| 2 | Saman     | Silva     | 1986-09-22 | 1          |
| 3 | Kamal     | Fernando   | 1982-11-10 | 2          |
| 4 | Nimal     | Ratnayake  | 1978-03-25 | 3          |
| 5 | Sunil     | Bandara   | 1995-07-12 | 4          |
| 6 | Ranil     | Jayawardena| 1988-01-30 | 5          |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> _

```

Figure 39. Updating customer table

```

C:\ MySQL 8.0 Command Line Client
mysql> select * from order_details;
+-----+-----+-----+-----+-----+-----+
| Order_ID | Order_Date | Total_Amount | CustomerID | DeliverMan_ID | stats      |
+-----+-----+-----+-----+-----+-----+
| 1 | 2024-03-01 | 2500.5 | 1 | 1 | delivered |
| 2 | 2024-03-05 | 1500.25 | 2 | 1 | pending    |
| 3 | 2024-03-10 | 3200.75 | 3 | 2 | delivered |
| 4 | 2024-03-15 | 1800.3 | 4 | 3 | delivered |
| 5 | 2024-03-20 | 2000 | 5 | 4 | pending    |
| 6 | 2024-03-25 | 2800.8 | 6 | 5 | delivered |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> UPDATE order_details
-> SET Stats = 'shipped'
-> WHERE Order_ID = 2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE order_details
-> SET Order_Date = '2024-03-09'
-> WHERE Order_ID = 3;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from order_details;
+-----+-----+-----+-----+-----+-----+
| Order_ID | Order_Date | Total_Amount | CustomerID | DeliverMan_ID | stats      |
+-----+-----+-----+-----+-----+-----+
| 1 | 2024-03-01 | 2500.5 | 1 | 1 | delivered |
| 2 | 2024-03-05 | 1500.25 | 2 | 1 | shipped    |
| 3 | 2024-03-09 | 3200.75 | 3 | 2 | delivered |
| 4 | 2024-03-15 | 1800.3 | 4 | 3 | delivered |
| 5 | 2024-03-20 | 2000 | 5 | 4 | pending    |
| 6 | 2024-03-25 | 2800.8 | 6 | 5 | delivered |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>

```

Figure 40. Updating order\_details table

```

-- MySQL 8.0 Command Line Client
mysql> select * from customer_email;
+-----+-----+-----+-----+-----+
| Email | City | District | Street_Name | Phone |
+-----+-----+-----+-----+-----+
| kamal@gmail.com | Galle | Galle | Fort Road | 0763334444 |
| nimal@gmail.com | Jaffna | Jaffna | Main Street | 0754445555 |
| nishantha@gmail.com | Colombo | Colombo | Galle Road | 0711112222 |
| ranil@gmail.com | Kurunegala | Kurunegala | Station Road | 0786667777 |
| saman@gmail.com | Kandy | Kandy | Peradeniya Road | 0772223333 |
| sunil@gmail.com | Matara | Matara | Beach Road | 0725556666 |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> UPDATE customer_email
      -> SET Phone = '0777777777'
      -> WHERE Email = 'saman@gmail.com';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE customer_email
      -> SET District = 'Nuwara Eliya'
      -> WHERE Email = 'nimal@gmail.com';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from customer_email;
+-----+-----+-----+-----+-----+
| Email | City | District | Street_Name | Phone |
+-----+-----+-----+-----+-----+
| kamal@gmail.com | Galle | Galle | Fort Road | 0763334444 |
| nimal@gmail.com | Jaffna | Nuwara Eliya | Main Street | 0754445555 |
| nishantha@gmail.com | Colombo | Colombo | Galle Road | 0711112222 |
| ranil@gmail.com | Kurunegala | Kurunegala | Station Road | 0786667777 |
| saman@gmail.com | Kandy | Kandy | Peradeniya Road | 0772223333 |
| sunil@gmail.com | Matara | Matara | Beach Road | 0725556666 |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> _

```

Figure 41. Updating customer\_email table

```

-- MySQL 8.0 Command Line Client
6 rows in set (0.00 sec)

mysql> select * from shipping_address;
+-----+-----+-----+-----+-----+
| CustomerID | AddressCode | city | State | ZipCode | Street |
+-----+-----+-----+-----+-----+
| 1 | 00100 | colombo | Western | 00100 | Galle Road |
| 2 | 20000 | Kandy | Central | 20000 | Peradeniya Road |
| 3 | 80000 | Galle | Southern | 80000 | Fort Road |
| 4 | 40000 | Jaffna | Northern | 40000 | Main Street |
| 5 | 90000 | Matara | Southern | 90000 | Beach Road |
| 6 | 60000 | Kurunegala | North Western | 60000 | Station Road |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> UPDATE shipping_address
      -> SET City = 'Gampaha'
      -> WHERE CustomerID = 2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE shipping_address
      -> SET State = 'Western Province'
      -> WHERE CustomerID = 4;
ERROR 1486 (22001): Data too long for column 'State' at row 1
mysql> UPDATE shipping_address
      -> SET State = 'Western'
      -> WHERE CustomerID = 4;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from shipping_address;
+-----+-----+-----+-----+-----+
| CustomerID | AddressCode | city | State | ZipCode | Street |
+-----+-----+-----+-----+-----+
| 1 | 00100 | colombo | Western | 00100 | Galle Road |
| 2 | 20000 | Gampaha | Central | 20000 | Peradeniya Road |
| 3 | 80000 | Galle | Southern | 80000 | Fort Road |
| 4 | 40000 | Jaffna | Western | 40000 | Main Street |
| 5 | 90000 | Matara | Southern | 90000 | Beach Road |
| 6 | 60000 | Kurunegala | North Western | 60000 | Station Road |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> _

```

Figure 42. Updating shipping\_address table

```

MySQL> select * from payment;
+-----+-----+-----+-----+-----+-----+
| Payment_ID | CustomerID | OrderID | Amount | Payment_Date | Payment_Method |
+-----+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 2500 | 2024-03-02 | Cash |
| 2 | 1 | 6 | 1500 | 2024-03-06 | Credit Card |
| 3 | 3 | 3 | 3200 | 2024-03-11 | Cash |
| 4 | 2 | 6 | 1800 | 2024-03-16 | Credit Card |
| 5 | 5 | 5 | 2000 | 2024-03-21 | Cash |
| 6 | 6 | 6 | 2800 | 2024-03-26 | Credit Card |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> UPDATE payment
      -> SET Amount = 2200
      -> WHERE Payment_ID = 3;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE payment
      -> SET Payment_Method = 'Debit Card'
      -> WHERE Payment_ID = 5;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from payment;
+-----+-----+-----+-----+-----+-----+
| Payment_ID | CustomerID | OrderID | Amount | Payment_Date | Payment_Method |
+-----+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 2500 | 2024-03-02 | Cash |
| 2 | 1 | 6 | 1500 | 2024-03-06 | Credit Card |
| 3 | 3 | 3 | 2200 | 2024-03-11 | Cash |
| 4 | 2 | 6 | 1800 | 2024-03-16 | Credit Card |
| 5 | 5 | 5 | 2000 | 2024-03-21 | Debit Card |
| 6 | 6 | 6 | 2800 | 2024-03-26 | Credit Card |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>

```

Figure 43. Updating payment table

```

MySQL> select * from warehouse;
+-----+-----+-----+-----+
| Warehouse_ID | Payment_Date | Location | Capacity |
+-----+-----+-----+-----+
| 1 | 2024-03-01 | Colombo | 5000 |
| 2 | 2024-03-05 | Kandy | 3000 |
| 3 | 2024-03-10 | Galle | 4000 |
| 4 | 2024-03-15 | Jaffna | 2000 |
| 5 | 2024-03-20 | Matara | 3500 |
| 6 | 2024-03-25 | Kurunegala | 2500 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> UPDATE warehouse
      -> SET Location = 'Colombo 03'
      -> WHERE Warehouse_ID = 1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE warehouse
      -> SET Capacity = 4500
      -> WHERE Warehouse_ID = 3;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from warehouse;
+-----+-----+-----+-----+
| Warehouse_ID | Payment_Date | Location | Capacity |
+-----+-----+-----+-----+
| 1 | 2024-03-01 | Colombo 03 | 5000 |
| 2 | 2024-03-05 | Kandy | 3000 |
| 3 | 2024-03-10 | Galle | 4500 |
| 4 | 2024-03-15 | Jaffna | 2000 |
| 5 | 2024-03-20 | Matara | 3500 |
| 6 | 2024-03-25 | Kurunegala | 2500 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>

```

Figure 44. Updating warehouse table

```
-- MySQL 8.0 Command Line Client
mysql> select * from inventory;
+-----+-----+-----+-----+
| Inventory_ID | WarehouseID | Stock_Quantity | Last_Stock_Update |
+-----+-----+-----+-----+
| 1 | 1 | 1000 | 1709231400 |
| 2 | 1 | 800 | 1709577000 |
| 3 | 3 | 1200 | 1710069000 |
| 4 | 2 | 600 | 1710441000 |
| 5 | 5 | 900 | 1710873000 |
| 6 | 6 | 1100 | 1711305000 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> UPDATE inventory
    -> SET Stock_Quantity = 1500
    -> WHERE Inventory_ID = 3;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE inventory
    -> SET Last_Stock_Update = UNIX_TIMESTAMP('2024-04-01')
    -> WHERE Inventory_ID = 5;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from inventory;
+-----+-----+-----+-----+
| Inventory_ID | WarehouseID | Stock_Quantity | Last_Stock_Update |
+-----+-----+-----+-----+
| 1 | 1 | 1000 | 1709231400 |
| 2 | 1 | 800 | 1709577000 |
| 3 | 3 | 1500 | 1710069000 |
| 4 | 2 | 600 | 1710441000 |
| 5 | 5 | 900 | 1711909800 |
| 6 | 6 | 1100 | 1711305000 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Figure 45. Updating inventory table

```
-- MySQL 8.0 Command Line Client
mysql> select * from product;
+-----+-----+-----+-----+-----+-----+
| Product_ID | InventoryID | Stock_Quantity | Price | Product_Name | Product_Description |
+-----+-----+-----+-----+-----+-----+
| 1 | 1 | 1000 | 150 | Rice | High-quality rice from local farms |
| 2 | 2 | 800 | 250 | Tea | Premium Ceylon tea |
| 3 | 1 | 1200 | 300 | Spices | Assorted spices for cooking |
| 4 | 4 | 600 | 200 | Coconut | Organic coconut oil |
| 5 | 2 | 900 | 180 | Apparel | Traditional Sri Lankan clothing |
| 6 | 6 | 1100 | 350 | Handicraft | Handmade crafts from local artisans |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> UPDATE product
    -> SET Price = 220
    -> WHERE Product_ID = 4;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE product
    -> SET Product_Description = 'Handmade crafts from local artisans, perfect for home decor'
    -> WHERE Product_ID = 6;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from product;
+-----+-----+-----+-----+-----+-----+
| Product_ID | InventoryID | Stock_Quantity | Price | Product_Name | Product_Description |
+-----+-----+-----+-----+-----+-----+
| 1 | 1 | 1000 | 150 | Rice | High-quality rice from local farms |
| 2 | 2 | 800 | 250 | Tea | Premium Ceylon tea |
| 3 | 1 | 1200 | 300 | Spices | Assorted spices for cooking |
| 4 | 4 | 600 | 220 | Coconut | Organic coconut oil |
| 5 | 2 | 900 | 180 | Apparel | Traditional Sri Lankan clothing |
| 6 | 6 | 1100 | 350 | Handicraft | Handmade crafts from local artisans, perfect for home decor |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Figure 46. Updating product table

```
MySQL 8.0 Command Line Client
mysql> select * from order_item;
+-----+-----+-----+-----+-----+
| Order_Item_ID | ProductID | OrderID | Quantity | Price |
+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 2 | 300 |
| 2 | 2 | 2 | 3 | 750 |
| 3 | 3 | 3 | 1 | 500 |
| 4 | 1 | 2 | 2 | 300 |
| 5 | 2 | 3 | 4 | 720 |
| 6 | 6 | 6 | 1 | 350 |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> UPDATE order_item
    -> SET Quantity = 3
    -> WHERE Order_Item_ID = 4;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE order_item
    -> SET Price = 280
    -> WHERE Order_Item_ID = 5;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from order_item;
+-----+-----+-----+-----+-----+
| Order_Item_ID | ProductID | OrderID | Quantity | Price |
+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 2 | 300 |
| 2 | 2 | 2 | 3 | 750 |
| 3 | 3 | 3 | 1 | 500 |
| 4 | 1 | 2 | 3 | 300 |
| 5 | 2 | 3 | 4 | 280 |
| 6 | 6 | 6 | 1 | 350 |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Figure 47. Updating order\_item table

```
MySQL 8.0 Command Line Client
mysql> select * from supplier;
+-----+-----+-----+-----+
| Supplier ID | Supplier Name | Contact Person | Phone NO |
+-----+-----+-----+-----+
| 1 | ABC Suppliers | John Doe | 771112222 |
| 2 | XYZ Suppliers | Jane Smith | 762223333 |
| 3 | PQR Suppliers | David Brown | 753334444 |
| 4 | LMN Suppliers | Emma Johnson | 784445555 |
| 5 | DEF Suppliers | Michael Williams | 715556666 |
| 6 | GHI Suppliers | Sophia Brown | 726667777 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> UPDATE supplier
    -> SET Contact_Person = 'Emma Watson'
    -> WHERE Supplier_ID = 4;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE supplier
    -> SET Phone_NO = '0719998888'
    -> WHERE Supplier_ID = 1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from supplier;
+-----+-----+-----+-----+
| Supplier ID | Supplier Name | Contact Person | Phone NO |
+-----+-----+-----+-----+
| 1 | ABC Suppliers | John Doe | 719998888 |
| 2 | XYZ Suppliers | Jane Smith | 762223333 |
| 3 | PQR Suppliers | David Brown | 753334444 |
| 4 | LMN Suppliers | Emma Watson | 784445555 |
| 5 | DEF Suppliers | Michael Williams | 715556666 |
| 6 | GHI Suppliers | Sophia Brown | 726667777 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Figure 48. Updating supplier table

```
-- MySQL 8.0 Command Line Client
mysql> select * from product_supplier;
+-----+-----+
| ProductID | SupplierID |
+-----+-----+
|       1    |      1     |
|       5    |      1     |
|       2    |      3     |
|       3    |      3     |
|       4    |      4     |
|       6    |      6     |
+-----+-----+
6 rows in set (0.00 sec)

mysql> UPDATE product_supplier
-> SET SupplierID = 5
-> WHERE ProductID = 6;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE product_supplier
-> SET ProductID = 4, SupplierID = 3
-> WHERE ProductID = 1 AND SupplierID = 1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from product_supplier;
+-----+-----+
| ProductID | SupplierID |
+-----+-----+
|       5    |      1     |
|       2    |      3     |
|       3    |      3     |
|       4    |      3     |
|       4    |      4     |
|       6    |      5     |
+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Figure 49. Updating product\_supplier table

```
-- MySQL 8.0 Command Line Client
mysql> select * from product_warehouse;
+-----+-----+
| ProductID | WarehouseID |
+-----+-----+
|       1    |      1     |
|       2    |      1     |
|       3    |      1     |
|       4    |      2     |
|       5    |      2     |
|       6    |      2     |
+-----+-----+
6 rows in set (0.00 sec)

mysql> UPDATE product_warehouse
-> SET WarehouseID = 4
-> WHERE ProductID = 5 AND WarehouseID = 2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE product_warehouse
-> SET ProductID = 2
-> WHERE ProductID = 4 AND WarehouseID = 2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from product_warehouse;
+-----+-----+
| ProductID | WarehouseID |
+-----+-----+
|       1    |      1     |
|       2    |      1     |
|       3    |      1     |
|       2    |      2     |
|       6    |      2     |
|       5    |      4     |
+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Figure 50. Updating product\_warehouse table

```

-- MySQL 8.0 Command Line Client
mysql> select * from employee;
+-----+-----+-----+-----+
| Employee_ID | First_Name | Last_Name | DOB      | Manager_ID |
+-----+-----+-----+-----+
| 1 | Kasun    | Perera   | 1980-05-15 | 1          |
| 2 | Nimal    | Silva    | 1975-08-20 | 1          |
| 3 | Sunil    | Fernando | 1972-11-10 | 2          |
| 4 | Raniil   | Ratnayake | 1968-03-25 | 2          |
| 5 | Chaminda | Bandara  | 1985-07-12 | 1          |
| 6 | Kamal    | Jayawardena | 1978-01-30 | 1          |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> UPDATE employee
-> SET Manager_ID = 3
-> WHERE Employee_ID = 2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE employee
-> SET DOB = '1978-02-15'
-> WHERE Employee_ID = 4;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from employee;
+-----+-----+-----+-----+
| Employee_ID | First_Name | Last_Name | DOB      | Manager_ID |
+-----+-----+-----+-----+
| 1 | Kasun    | Perera   | 1980-05-15 | 1          |
| 2 | Nimal    | Silva    | 1975-08-20 | 3          |
| 3 | Sunil    | Fernando | 1972-11-10 | 2          |
| 4 | Raniil   | Ratnayake | 1978-02-15 | 2          |
| 5 | Chaminda | Bandara  | 1985-07-12 | 1          |
| 6 | Kamal    | Jayawardena | 1978-01-30 | 1          |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> _

```

Figure 51. Updating employee table

```

-- MySQL 8.0 Command Line Client
mysql> select * from product_features;
+-----+-----+-----+-----+
| ProductID | Product_Name | Product_Value | Product_Color | Product_Size |
+-----+-----+-----+-----+
| 1 | Rice       | Premium     | White        | 5kg          |
| 2 | Tea        | Premium     | Black        | 200g         |
| 3 | Spices     | Assorted    | Various      | NULL         |
| 4 | Coconut    | Organic    | Brown        | NULL         |
| 5 | Apparel    | Traditional | Various      | NULL         |
| 6 | Handicraft | Artisanal  | Various      | NULL         |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> UPDATE product_features
-> SET Product_Value = 'Organic', Product_Color = 'Brown', Product_Size = 'Medium'
-> WHERE ProductID = 4;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE product_features
-> SET Product_Name = 'Brown Rice'
-> WHERE ProductID = 1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from product_features;
+-----+-----+-----+-----+
| ProductID | Product_Name | Product_Value | Product_Color | Product_Size |
+-----+-----+-----+-----+
| 1 | Brown Rice | Premium     | White        | 5kg          |
| 2 | Tea        | Premium     | Black        | 200g         |
| 3 | Spices     | Assorted    | Various      | NULL         |
| 4 | Coconut    | Organic    | Brown        | Medium       |
| 5 | Apparel    | Traditional | Various      | NULL         |
| 6 | Handicraft | Artisanal  | Various      | NULL         |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> _

```

Figure 52. Updating product\_features table

```

c:\ MySQL 8.0 Command Line Client
mysql> select * from employee_email;
+-----+-----+-----+-----+-----+
| Email | City  | District | Street_Name | Phone   |
+-----+-----+-----+-----+-----+
| chaminda@example.com | Matara | Matara   | Beach Road | 0725556666 |
| kamal@example.com    | Kurunegala | Kurunegala | Station Road | 0786667777 |
| kasun@example.com   | Colombo | Colombo  | Galle Road  | 0711112222 |
| nimal@example.com   | Kandy   | Kandy    | Peradeniya Road | 0772223333 |
| ranil@example.com   | Jaffna  | Jaffna   | Main Street | 0754445555 |
| sunil@example.com   | Galle   | Galle    | Fort Road   | 0763334444 |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> UPDATE employee_email
      -> SET Phone = '0788889999'
      -> WHERE Email = 'nimal@example.com';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE employee_email
      -> SET City = 'Gampaha'
      -> WHERE Email = 'sunil@example.com';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from employee_email;
+-----+-----+-----+-----+-----+
| Email | City  | District | Street_Name | Phone   |
+-----+-----+-----+-----+-----+
| chaminda@example.com | Matara | Matara   | Beach Road | 0725556666 |
| kamal@example.com    | Kurunegala | Kurunegala | Station Road | 0786667777 |
| kasun@example.com   | Colombo | Colombo  | Galle Road  | 0711112222 |
| nimal@example.com   | Kandy   | Kandy    | Peradeniya Road | 0788889999 |
| ranil@example.com   | Jaffna  | Jaffna   | Main Street | 0754445555 |
| sunil@example.com   | Gampaha | Galle    | Fort Road   | 0763334444 |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>

```

Figure 53. Updating employee\_email table

```

c:\ MySQL 8.0 Command Line Client
mysql> select * from employee_inventory;
+-----+-----+
| EmployeeID | InventoryID |
+-----+-----+
| 1           | 1           |
| 2           | 1           |
| 5           | 2           |
| 3           | 3           |
| 4           | 4           |
| 6           | 6           |
+-----+-----+
6 rows in set (0.00 sec)

mysql> UPDATE employee_inventory
      -> SET InventoryID = 2
      -> WHERE EmployeeID = 3 AND InventoryID = 3;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE employee_inventory
      -> SET EmployeeID = 5
      -> WHERE EmployeeID = 6 AND InventoryID = 6;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from employee_inventory;
+-----+-----+
| EmployeeID | InventoryID |
+-----+-----+
| 1           | 1           |
| 2           | 1           |
| 3           | 2           |
| 5           | 2           |
| 4           | 4           |
| 5           | 6           |
+-----+-----+
6 rows in set (0.00 sec)

mysql>

```

Figure 54. Updating employee\_inventory table

### 3.4. Deleting operation

The screenshot shows a MySQL Command Line Client window. The session starts with a SELECT query to view all records in the delivery\_Man table. The result set shows 6 rows. Then, a DELETE query is executed to remove the record where Delivery\_Man\_ID = 6. A confirmation message "Query OK, 1 row affected (0.00 sec)" is displayed. Finally, another SELECT query is run to show the updated state of the table, which now has 5 rows.

```
MySQL> select * from delivery_Man;
+-----+-----+-----+
| Delivery_Man_ID | First_Name | Last_Name | Phone |
+-----+-----+-----+
| 1 | Saman | Perera | 0715555555 |
| 2 | Kamal | Wijesinghe | 0712345678 |
| 3 | Nimal | Fernando | 0763456789 |
| 4 | Sunil | Ratnayake | 0784567890 |
| 5 | Ranil | Bandara | 0725678901 |
| 6 | Chaminda | Jayawardena | 0756789012 |
+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> DELETE FROM delivery_Man
-> WHERE Delivery_Man_ID = 6;
Query OK, 1 row affected (0.00 sec)

mysql> select * from delivery_Man;
+-----+-----+-----+
| Delivery_Man_ID | First_Name | Last_Name | Phone |
+-----+-----+-----+
| 1 | Saman | Perera | 0715555555 |
| 2 | Kamal | Wijesinghe | 0712345678 |
| 3 | Nimal | Fernando | 0763456789 |
| 4 | Sunil | Ratnayake | 0784567890 |
| 5 | Ranil | Bandara | 0725678901 |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> _
```

Figure 55. deleting from delivery\_man table

The screenshot shows a MySQL Command Line Client window. It begins with a SELECT query to list all records from the customer table. The result set contains 6 rows. Next, a DELETE query is run to delete the record where Customer\_ID = 6. A message "Query OK, 1 row affected (0.00 sec)" is shown. A final SELECT query is performed to verify that the row with Customer\_ID = 6 has been removed, leaving 5 rows in the table.

```
MySQL> select * from customer;
+-----+-----+-----+-----+
| Customer_ID | First_Name | Last_Name | DOB | Deliver_Man_ID |
+-----+-----+-----+-----+
| 1 | Nishantha | Perera | 1990-05-15 | 3 |
| 2 | Saman | Silva | 1986-09-22 | 1 |
| 3 | Kamal | Fernando | 1982-11-10 | 2 |
| 4 | Nimal | Ratnayake | 1978-03-25 | 3 |
| 5 | Sunil | Bandara | 1995-07-12 | 4 |
| 6 | Ranil | Jayawardena | 1988-01-30 | 5 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> DELETE FROM customer
-> WHERE Customer_ID = 6;
Query OK, 1 row affected (0.00 sec)

mysql> select * from customer;
+-----+-----+-----+-----+
| Customer_ID | First_Name | Last_Name | DOB | Deliver_Man_ID |
+-----+-----+-----+-----+
| 1 | Nishantha | Perera | 1990-05-15 | 3 |
| 2 | Saman | Silva | 1986-09-22 | 1 |
| 3 | Kamal | Fernando | 1982-11-10 | 2 |
| 4 | Nimal | Ratnayake | 1978-03-25 | 3 |
| 5 | Sunil | Bandara | 1995-07-12 | 4 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> _
```

Figure 56. deleting from customer table

```
c:\ MySQL 8.0 Command Line Client
mysql> select * from order_details;
+-----+-----+-----+-----+-----+
| Order_ID | Order_Date | Total_Amount | CustomerID | DeliverMan_ID | Stats
+-----+-----+-----+-----+-----+
| 1 | 2024-03-01 | 2500.5 | 1 | 1 | delivered
| 2 | 2024-03-05 | 1500.25 | 2 | 1 | shipped
| 3 | 2024-03-09 | 3200.75 | 3 | 2 | delivered
| 4 | 2024-03-15 | 1800.3 | 4 | 3 | delivered
| 5 | 2024-03-20 | 2000 | 5 | 4 | pending
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> DELETE FROM order_details
-> WHERE Order_ID = 6;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from order_details;
+-----+-----+-----+-----+-----+
| Order_ID | Order_Date | Total_Amount | CustomerID | DeliverMan_ID | Stats
+-----+-----+-----+-----+-----+
| 1 | 2024-03-01 | 2500.5 | 1 | 1 | delivered
| 2 | 2024-03-05 | 1500.25 | 2 | 1 | shipped
| 3 | 2024-03-09 | 3200.75 | 3 | 2 | delivered
| 4 | 2024-03-15 | 1800.3 | 4 | 3 | delivered
| 5 | 2024-03-20 | 2000 | 5 | 4 | pending
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> _
```

Figure 57. deleting from order\_details table

```
c:\ MySQL 8.0 Command Line Client
mysql> select * from customer_email;
+-----+-----+-----+-----+-----+
| Email | City | District | Street Name | Phone
+-----+-----+-----+-----+-----+
| kamal@gmail.com | Galle | Galle | Fort Road | 0763334444
| nimal@gmail.com | Jaffna | Nuwara Eliya | Main Street | 0754445555
| nishantha@gmail.com | Colombo | Colombo | Galle Road | 0711112222
| ranil@gmail.com | Kurunegala | Kurunegala | Station Road | 0786667777
| saman@gmail.com | Kandy | Kandy | Peradeniya Road | 0777777777
| sunil@gmail.com | Matara | Matara | Beach Road | 0725556666
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> DELETE FROM customer_email
-> WHERE Email = 'sunil@gmail.com';
Query OK, 1 row affected (0.00 sec)

mysql> select * from customer_email;
+-----+-----+-----+-----+-----+
| Email | City | District | Street Name | Phone
+-----+-----+-----+-----+-----+
| kamal@gmail.com | Galle | Galle | Fort Road | 0763334444
| nimal@gmail.com | Jaffna | Nuwara Eliya | Main Street | 0754445555
| nishantha@gmail.com | Colombo | Colombo | Galle Road | 0711112222
| ranil@gmail.com | Kurunegala | Kurunegala | Station Road | 0786667777
| saman@gmail.com | Kandy | Kandy | Peradeniya Road | 0777777777
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

Figure 58. deleting from customer\_email table

```
c:\ MySQL 8.0 Command Line Client
mysql> select * from shipping_address;
+-----+-----+-----+-----+-----+
| CustomerID | AddressCode | City   | State  | ZipCode | Street      |
+-----+-----+-----+-----+-----+
| 1 | 00100 | Colombo | Western | 00100 | Galle Road |
| 2 | 20000 | Gampaha | Central | 20000 | Peradeniya Road |
| 3 | 80000 | Galle   | Southern | 80000 | Fort Road   |
| 4 | 40000 | Jaffna  | Western  | 40000 | Main Street |
| 5 | 90000 | Matara  | Southern | 90000 | Beach Road  |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> DELETE FROM shipping_address
    -> WHERE CustomerID = 6;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> select * from shipping_address;
+-----+-----+-----+-----+-----+
| CustomerID | AddressCode | City   | State  | ZipCode | Street      |
+-----+-----+-----+-----+-----+
| 1 | 00100 | Colombo | Western | 00100 | Galle Road |
| 2 | 20000 | Gampaha | Central | 20000 | Peradeniya Road |
| 3 | 80000 | Galle   | Southern | 80000 | Fort Road   |
| 4 | 40000 | Jaffna  | Western  | 40000 | Main Street |
| 5 | 90000 | Matara  | Southern | 90000 | Beach Road  |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> _
```

Figure 59. deleting from shipping\_address table

```
c:\ MySQL 8.0 Command Line Client
mysql> select * from payment;
+-----+-----+-----+-----+-----+
| Payment_ID | CustomerID | OrderID | Amount | Payment_Date | Payment_Method |
+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 2500 | 2024-03-02 | Cash |
| 3 | 3 | 3 | 2200 | 2024-03-11 | Cash |
| 5 | 5 | 5 | 2000 | 2024-03-21 | Debit Card |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> DELETE FROM payment
    -> WHERE Payment_ID = 6;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> select * from payment;
+-----+-----+-----+-----+-----+
| Payment_ID | CustomerID | OrderID | Amount | Payment_Date | Payment_Method |
+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 2500 | 2024-03-02 | Cash |
| 3 | 3 | 3 | 2200 | 2024-03-11 | Cash |
| 5 | 5 | 5 | 2000 | 2024-03-21 | Debit Card |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> _
```

Figure 60. deleting from payment table

```
MySQL 8.0 Command Line Client
mysql> select * from warehouse;
+-----+-----+-----+-----+
| Warehouse_ID | Payment_Date | Location | Capacity |
+-----+-----+-----+-----+
| 1 | 2024-03-01 | Colombo 03 | 5000 |
| 2 | 2024-03-05 | Kandy | 3000 |
| 3 | 2024-03-10 | Galle | 4500 |
| 4 | 2024-03-15 | Jaffna | 2000 |
| 5 | 2024-03-20 | Matara | 3500 |
| 6 | 2024-03-25 | Kurunegala | 2500 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> DELETE FROM warehouse
-> WHERE Warehouse_ID = 6;
Query OK, 1 row affected (0.00 sec)

mysql> select * from warehouse;
+-----+-----+-----+-----+
| Warehouse_ID | Payment_Date | Location | Capacity |
+-----+-----+-----+-----+
| 1 | 2024-03-01 | Colombo 03 | 5000 |
| 2 | 2024-03-05 | Kandy | 3000 |
| 3 | 2024-03-10 | Galle | 4500 |
| 4 | 2024-03-15 | Jaffna | 2000 |
| 5 | 2024-03-20 | Matara | 3500 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

Figure 61. deleting from warehouse table

```
MySQL 8.0 Command Line Client
5 rows in set (0.00 sec)

mysql> select * from inventory;
+-----+-----+-----+-----+
| Inventory_ID | WarehouseID | Stock_Quantity | Last_Stock_Update |
+-----+-----+-----+-----+
| 1 | 1 | 1000 | 1709231400 |
| 2 | 1 | 800 | 1709577000 |
| 3 | 3 | 1500 | 1710090000 |
| 4 | 2 | 600 | 1710441000 |
| 5 | 5 | 900 | 1711909800 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> DELETE FROM inventory
-> WHERE Inventory_ID = 4;
Query OK, 1 row affected (0.00 sec)

mysql> select * from inventory;
+-----+-----+-----+-----+
| Inventory_ID | WarehouseID | Stock_Quantity | Last_Stock_Update |
+-----+-----+-----+-----+
| 1 | 1 | 1000 | 1709231400 |
| 2 | 1 | 800 | 1709577000 |
| 3 | 3 | 1500 | 1710090000 |
| 5 | 5 | 900 | 1711909800 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

Figure 62. deleting from inventory table

```
MySQL 8.0 Command Line Client

mysql> select * from product;
+-----+-----+-----+-----+-----+-----+
| Product_ID | InventoryID | Stock_Quantity | Price | Product_Name | Product_Description |
+-----+-----+-----+-----+-----+-----+
| 1 | 1 | 1000 | 150 | Rice | High-quality rice from local farms |
| 2 | 2 | 800 | 250 | Tea | Premium Ceylon tea |
| 3 | 1 | 1200 | 300 | Spices | Assorted spices for cooking |
| 5 | 2 | 900 | 180 | Apparel | Traditional Sri Lankan clothing |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> DELETE FROM product
-> WHERE Product_ID = 5;
Query OK, 1 row affected (0.00 sec)

mysql> select * from product;
+-----+-----+-----+-----+-----+-----+
| Product_ID | InventoryID | Stock_Quantity | Price | Product_Name | Product_Description |
+-----+-----+-----+-----+-----+-----+
| 1 | 1 | 1000 | 150 | Rice | High-quality rice from local farms |
| 2 | 2 | 800 | 250 | Tea | Premium Ceylon tea |
| 3 | 1 | 1200 | 300 | Spices | Assorted spices for cooking |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> _
```

Figure 63. deleting from product table

```
MySQL 8.0 Command Line Client

mysql> select * from order_item;
+-----+-----+-----+-----+-----+
| Order_Item_ID | ProductID | OrderID | Quantity | Price |
+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 2 | 300 |
| 2 | 2 | 2 | 3 | 750 |
| 3 | 3 | 3 | 1 | 300 |
| 4 | 1 | 2 | 3 | 300 |
| 5 | 2 | 3 | 4 | 280 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> DELETE FROM order_item
-> WHERE Order_Item_ID = 6;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from order_item;
+-----+-----+-----+-----+-----+
| Order_Item_ID | ProductID | OrderID | Quantity | Price |
+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 2 | 300 |
| 2 | 2 | 2 | 3 | 750 |
| 3 | 3 | 3 | 1 | 300 |
| 4 | 1 | 2 | 3 | 300 |
| 5 | 2 | 3 | 4 | 280 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

Figure 64. deleting from order item table

```
MySQL 8.0 Command Line Client
mysql> select * from supplier;
+-----+-----+-----+
| Supplier_ID | Supplier_Name | Contact_Person | Phone_NO |
+-----+-----+-----+
| 1 | ABC Suppliers | John Doe | 719998888 |
| 2 | XYZ Suppliers | Jane Smith | 762223333 |
| 3 | PQR Suppliers | David Brown | 753334444 |
| 4 | LMN Suppliers | Emma Watson | 784445555 |
| 5 | DEF Suppliers | Michael Williams | 715556666 |
| 6 | GHI Suppliers | Sophia Brown | 726667777 |
+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> DELETE FROM supplier
-> WHERE Supplier_ID = 6;
Query OK, 1 row affected (0.00 sec)

mysql> select * from supplier;
+-----+-----+-----+
| Supplier_ID | Supplier_Name | Contact_Person | Phone_NO |
+-----+-----+-----+
| 1 | ABC suppliers | John Doe | 719998888 |
| 2 | XYZ Suppliers | Jane Smith | 762223333 |
| 3 | PQR Suppliers | David Brown | 753334444 |
| 4 | LMN Suppliers | Emma Watson | 784445555 |
| 5 | DEF Suppliers | Michael Williams | 715556666 |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> _
```

Figure 65. deleting from supplier table

```
MySQL 8.0 Command Line Client
5 rows in set (0.00 sec)

mysql> select * from product_supplier;
+-----+-----+
| ProductID | SupplierID |
+-----+-----+
| 2 | 3 |
| 3 | 3 |
+-----+-----+
2 rows in set (0.00 sec)

mysql> DELETE FROM product_supplier
-> WHERE ProductID = 2 AND SupplierID = 3;
Query OK, 1 row affected (0.00 sec)

mysql> select * from product_supplier;
+-----+-----+
| ProductID | supplierID |
+-----+-----+
| 3 | 3 |
+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

Figure 66. deleting from product supplier table

```
MySQL 8.0 Command Line Client

mysql> select * from product_warehouse;
+-----+-----+
| ProductID | WarehouseID |
+-----+-----+
| 1          | 1          |
| 2          | 1          |
| 3          | 1          |
| 2          | 2          |
+-----+-----+
4 rows in set (0.00 sec)

mysql> DELETE FROM product_warehouse
-> WHERE ProductID = 2 AND WarehouseID = 2;
Query OK, 1 row affected (0.00 sec)

mysql> select * from product_warehouse;
+-----+-----+
| ProductID | WarehouseID |
+-----+-----+
| 1          | 1          |
| 2          | 1          |
| 3          | 1          |
+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

Figure 67. deleting from product\_warehouse table

```
MySQL 8.0 Command Line Client

mysql> select * from employee;
+-----+-----+-----+-----+
| Employee_ID | First_Name | Last_Name | DOB      | Manager_ID |
+-----+-----+-----+-----+
| 1           | Kasun      | Perera    | 1980-05-15 | 1          |
| 2           | Nimal      | Silva     | 1975-08-20 | 3          |
| 3           | Sunil      | Fernando  | 1972-11-10 | 2          |
| 4           | Ranil      | Ratnayake | 1978-02-15 | 2          |
| 5           | Chaminda   | Bandara   | 1985-07-12 | 1          |
| 6           | Kamal      | Jayawardena | 1978-01-30 | 1          |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> DELETE FROM employee
-> WHERE Employee_ID = 6;
Query OK, 1 row affected (0.00 sec)

mysql> select * from employee;
+-----+-----+-----+-----+
| Employee_ID | First_Name | Last_Name | DOB      | Manager_ID |
+-----+-----+-----+-----+
| 1           | Kasun      | Perera    | 1980-05-15 | 1          |
| 2           | Nimal      | Silva     | 1975-08-20 | 3          |
| 3           | Sunil      | Fernando  | 1972-11-10 | 2          |
| 4           | Ranil      | Ratnayake | 1978-02-15 | 2          |
| 5           | Chaminda   | Bandara   | 1985-07-12 | 1          |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> _
```

Figure 68. deleting from employee table

```
MySQL 8.0 Command Line Client
mysql> select * from product_features;
+-----+-----+-----+-----+-----+
| ProductID | Product_Name | Product_Value | Product_Color | Product_Size |
+-----+-----+-----+-----+-----+
| 1 | Brown Rice | Premium | White | 5kg |
| 2 | Tea | Premium | Black | 200g |
| 3 | Spices | Assorted | Various | NULL |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> DELETE FROM product_features
-> WHERE ProductID = 2;
Query OK, 1 row affected (0.00 sec)

mysql> select * from product_features;
+-----+-----+-----+-----+-----+
| ProductID | Product_Name | Product_Value | Product_Color | Product_Size |
+-----+-----+-----+-----+-----+
| 1 | Brown Rice | Premium | White | 5kg |
| 3 | Spices | Assorted | Various | NULL |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Figure 69. deleting from product\_features table

```
MySQL 8.0 Command Line Client
mysql> select * from employee_email;
+-----+-----+-----+-----+-----+
| Email | City | District | Street_Name | Phone |
+-----+-----+-----+-----+-----+
| chaminda@example.com | Matara | Matara | Beach Road | 0725556666 |
| kamal@example.com | Kurunegala | Kurunegala | Station Road | 0786667777 |
| kasun@example.com | Colombo | Colombo | Galle Road | 0711112222 |
| nimil@example.com | Kandy | Kandy | Peradeniya Road | 0788889999 |
| ranil@example.com | Jaffna | Jaffna | Main Street | 0754445555 |
| sunil@example.com | Gampaha | Gampaha | Fort Road | 0763334444 |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> DELETE FROM employee_email
-> WHERE Email = 'kasun@example.com';
Query OK, 1 row affected (0.00 sec)

mysql> select * from employee_email;
+-----+-----+-----+-----+-----+
| Email | City | District | Street_Name | Phone |
+-----+-----+-----+-----+-----+
| chaminda@example.com | Matara | Matara | Beach Road | 0725556666 |
| kamal@example.com | Kurunegala | Kurunegala | Station Road | 0786667777 |
| nimil@example.com | Kandy | Kandy | Peradeniya Road | 0788889999 |
| ranil@example.com | Jaffna | Jaffna | Main Street | 0754445555 |
| sunil@example.com | Gampaha | Gampaha | Fort Road | 0763334444 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> _
```

Figure 70. deleting from employee\_email table

```
MySQL 8.0 Command Line Client

mysql> select * from employee_inventory;
+-----+-----+
| EmployeeID | InventoryID |
+-----+-----+
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 5 | 2 |
+-----+-----+
4 rows in set (0.00 sec)

mysql> DELETE FROM employee_inventory
-> WHERE EmployeeID = 1 AND InventoryID = 1;
Query OK, 1 row affected (0.00 sec)

mysql> select * from employee_inventory;
+-----+-----+
| EmployeeID | InventoryID |
+-----+-----+
| 2 | 1 |
| 3 | 2 |
| 5 | 2 |
+-----+-----+
3 rows in set (0.00 sec)

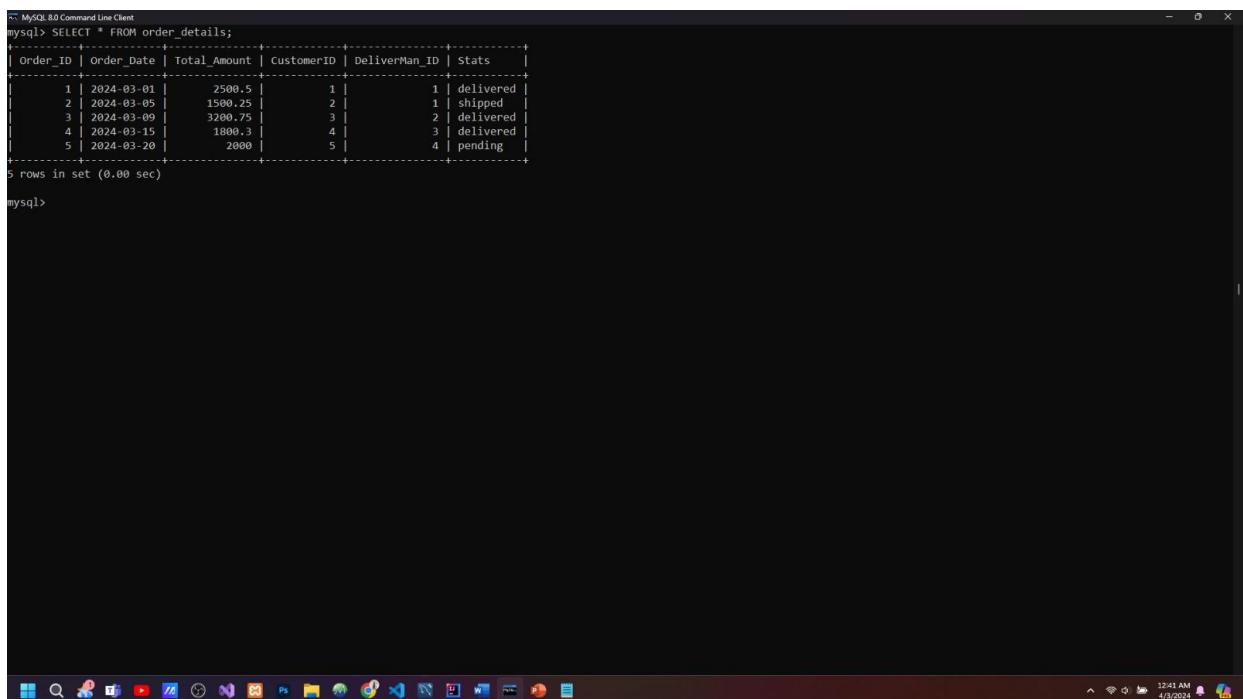
mysql>
```

Figure 71. deleting from employee\_inventory table

# Chapter 4 – Transaction

## 4.1 Simple queries

### 4.1.1 Select operation



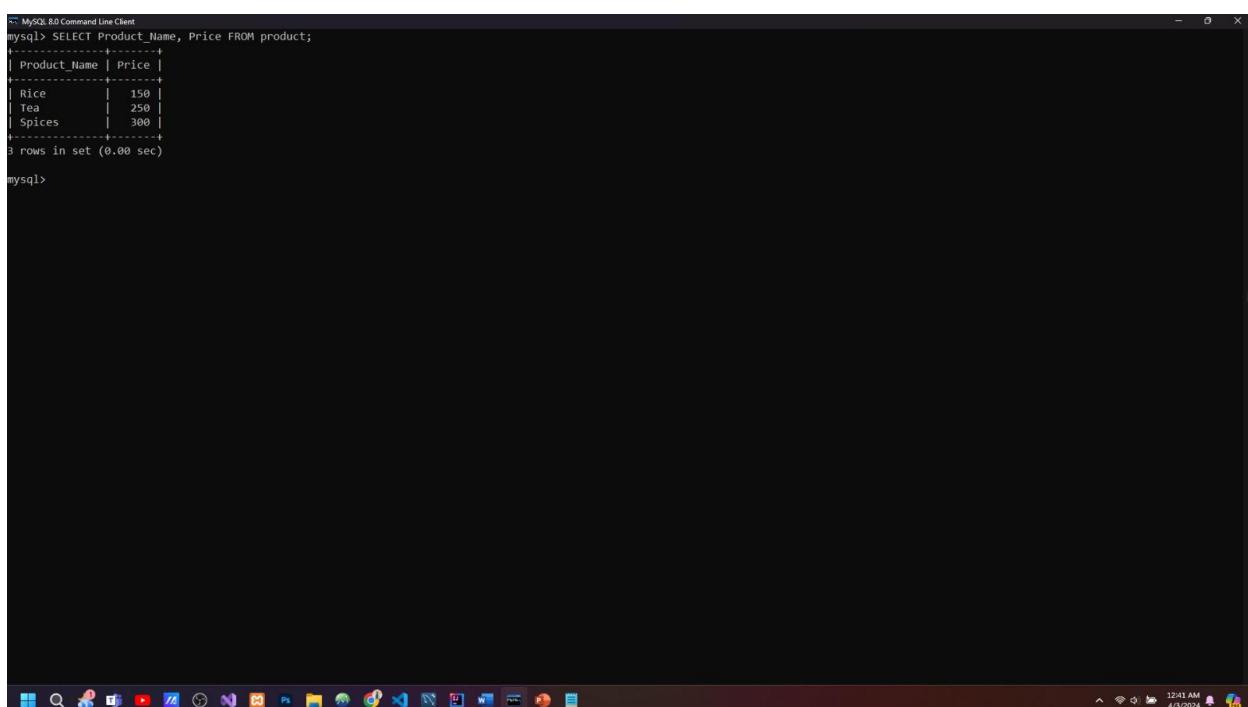
MySQL 8.0 Command Line Client

```
mysql> SELECT * FROM order_details;
+-----+-----+-----+-----+-----+
| Order_ID | Order_Date | Total_Amount | CustomerID | DeliverMan_ID | Stats      |
+-----+-----+-----+-----+-----+
| 1       | 2024-03-01 | 2500.5      | 3          | 1             | delivered   |
| 2       | 2024-03-05 | 1500.25     | 2          | 1             | shipped     |
| 3       | 2024-03-09 | 3200.75     | 3          | 2             | delivered   |
| 4       | 2024-03-15 | 1800.3      | 4          | 3             | delivered   |
| 5       | 2024-03-20 | 2000        | 5          | 4             | pending     |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

mysql>

Figure 72.simple queries - select operation

### 4.1.2 Project operation



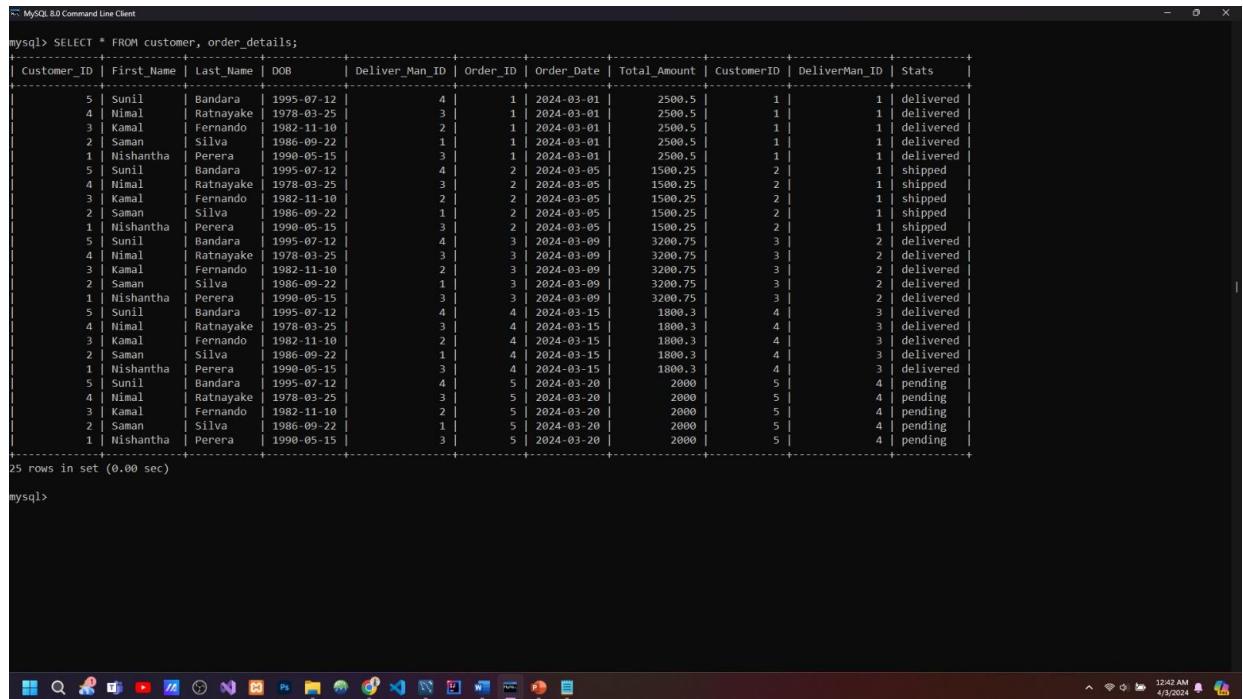
MySQL 8.0 Command Line Client

```
mysql> SELECT Product_Name, Price FROM product;
+-----+-----+
| Product_Name | Price |
+-----+-----+
| Rice         | 150   |
| Tea          | 250   |
| Spices       | 300   |
+-----+-----+
3 rows in set (0.00 sec)
```

mysql>

Figure 73, simple queries - project operation

#### 4.1.3 Cartesian operation



MySQL 8.0 Command Line Client

```
mysql> SELECT * FROM customer, order_details;
```

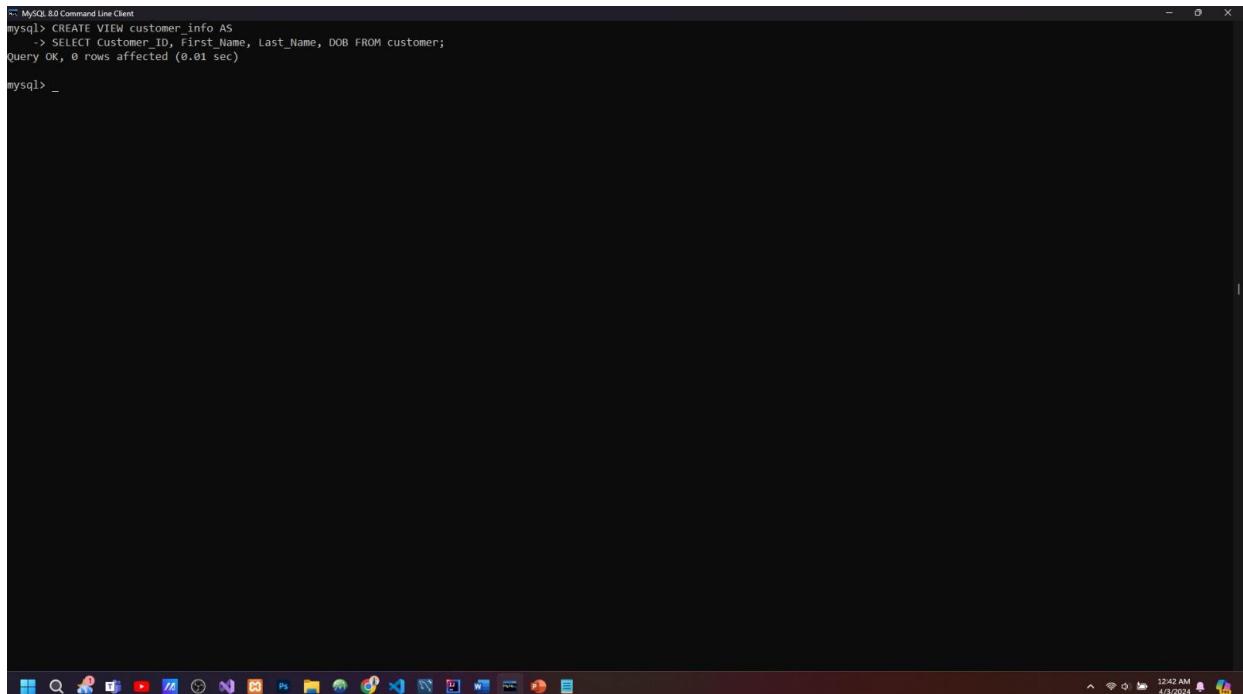
Customer_ID	First_Name	Last_Name	DOB	Deliver_Man_ID	Order_ID	Order_Date	Total_Amount	CustomerID	DeliverMan_ID	Stats
5	Sunil	Bandara	1995-07-12	4	1	2024-03-01	2500.5	1	1	delivered
4	Nimal	Ratnayake	1978-03-25	3	1	2024-03-01	2500.5	1	1	delivered
3	Kamal	Fernando	1982-11-10	2	1	2024-03-01	2500.5	1	1	delivered
2	Saman	Silva	1986-09-22	1	1	2024-03-01	2500.5	1	1	delivered
1	Nishantha	Perera	1990-05-15	3	1	2024-03-01	2500.5	1	1	delivered
5	Sunil	Bandara	1995-07-12	4	2	2024-03-05	1500.25	2	1	shipped
4	Nimal	Ratnayake	1978-03-25	3	2	2024-03-05	1500.25	2	1	shipped
3	Kamal	Fernando	1982-11-10	2	2	2024-03-05	1500.25	2	1	shipped
2	Saman	Silva	1986-09-22	1	2	2024-03-05	1500.25	2	1	shipped
1	Nishantha	Perera	1990-05-15	3	2	2024-03-05	1500.25	2	1	shipped
5	Sunil	Bandara	1995-07-12	4	3	2024-03-05	3200.75	3	2	delivered
4	Nimal	Ratnayake	1978-03-25	3	3	2024-03-09	3200.75	3	2	delivered
3	Kamal	Fernando	1982-11-10	2	3	2024-03-09	3200.75	3	2	delivered
2	Saman	Silva	1986-09-22	1	3	2024-03-09	3200.75	3	2	delivered
1	Nishantha	Perera	1990-05-15	3	3	2024-03-09	3200.75	3	2	delivered
5	Sunil	Bandara	1995-07-12	4	4	2024-03-15	1800.3	4	3	delivered
4	Nimal	Ratnayake	1978-03-25	3	4	2024-03-15	1800.3	4	3	delivered
3	Kamal	Fernando	1982-11-10	2	4	2024-03-15	1800.3	4	3	delivered
2	Saman	Silva	1986-09-22	1	4	2024-03-15	1800.3	4	3	delivered
1	Nishantha	Perera	1990-05-15	3	4	2024-03-15	1800.3	4	3	delivered
5	Sunil	Bandara	1995-07-12	4	5	2024-03-20	2000	5	4	pending
4	Nimal	Ratnayake	1978-03-25	3	5	2024-03-20	2000	5	4	pending
3	Kamal	Fernando	1982-11-10	2	5	2024-03-20	2000	5	4	pending
2	Saman	Silva	1986-09-22	1	5	2024-03-20	2000	5	4	pending
1	Nishantha	Perera	1990-05-15	3	5	2024-03-20	2000	5	4	pending

25 rows in set (0.00 sec)

```
mysql>
```

Figure 74. simple queries - cartesian operation

#### 4.1.4. Creating a user view



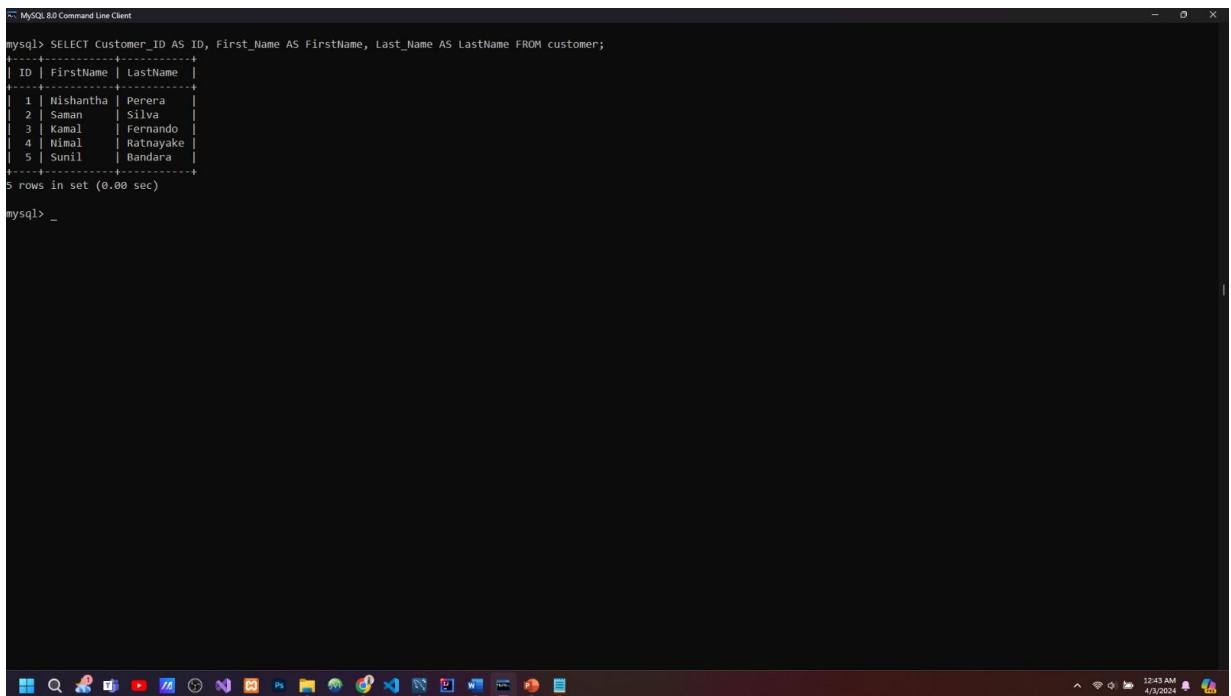
```
MySQL 8.0 Command Line Client
```

```
mysql> CREATE VIEW customer_info AS
-> SELECT Customer_ID, First_Name, Last_Name, DOB FROM customer;
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> _
```

Figure 75. simple queries - creating a new user view

#### 4.1.5. Renaming operation



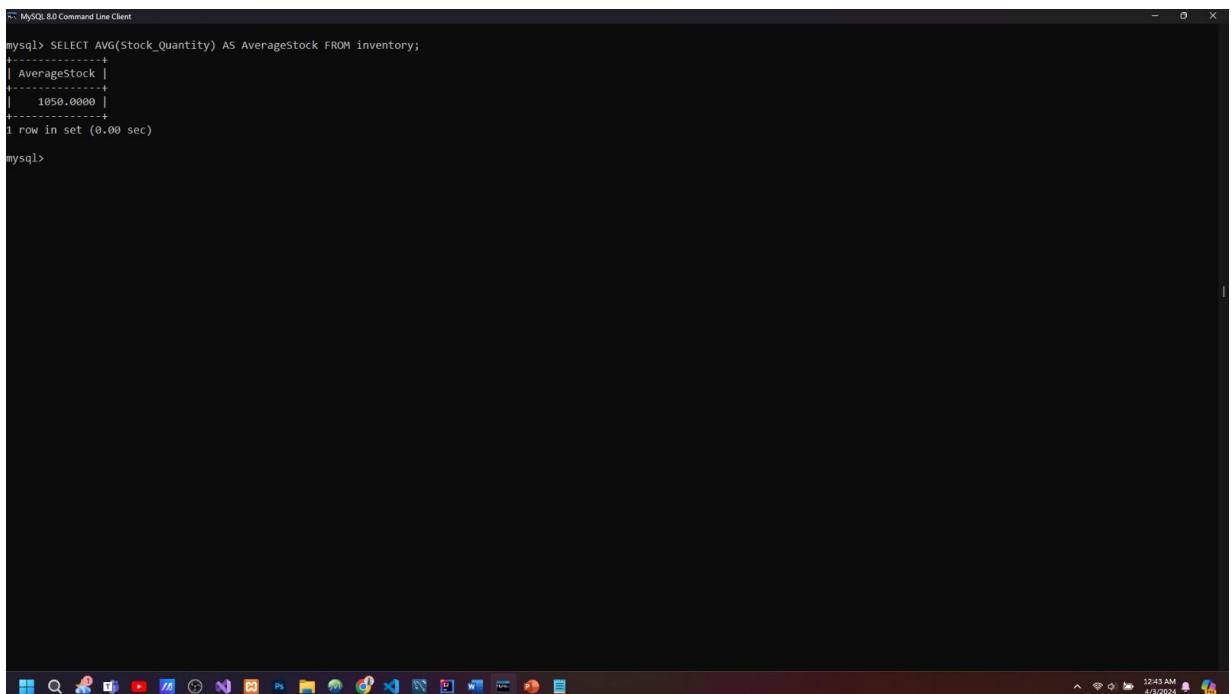
```
MySQL 8.0 Command Line Client

mysql> SELECT Customer_ID AS ID, First_Name AS FirstName, Last_Name AS LastName FROM customer;
+----+-----+-----+
| ID | Firstname | Lastname |
+----+-----+-----+
| 1  | Nishantha | Perera |
| 2  | Saman     | Silva   |
| 3  | Kamal     | Fernando |
| 4  | Nimal     | Ratnayake |
| 5  | Sunil     | Bandara |
+----+-----+-----+
5 rows in set (0.00 sec)

mysql> _
```

Figure 76. simple queries - renaming operation

#### 4.1.6. Aggregation function



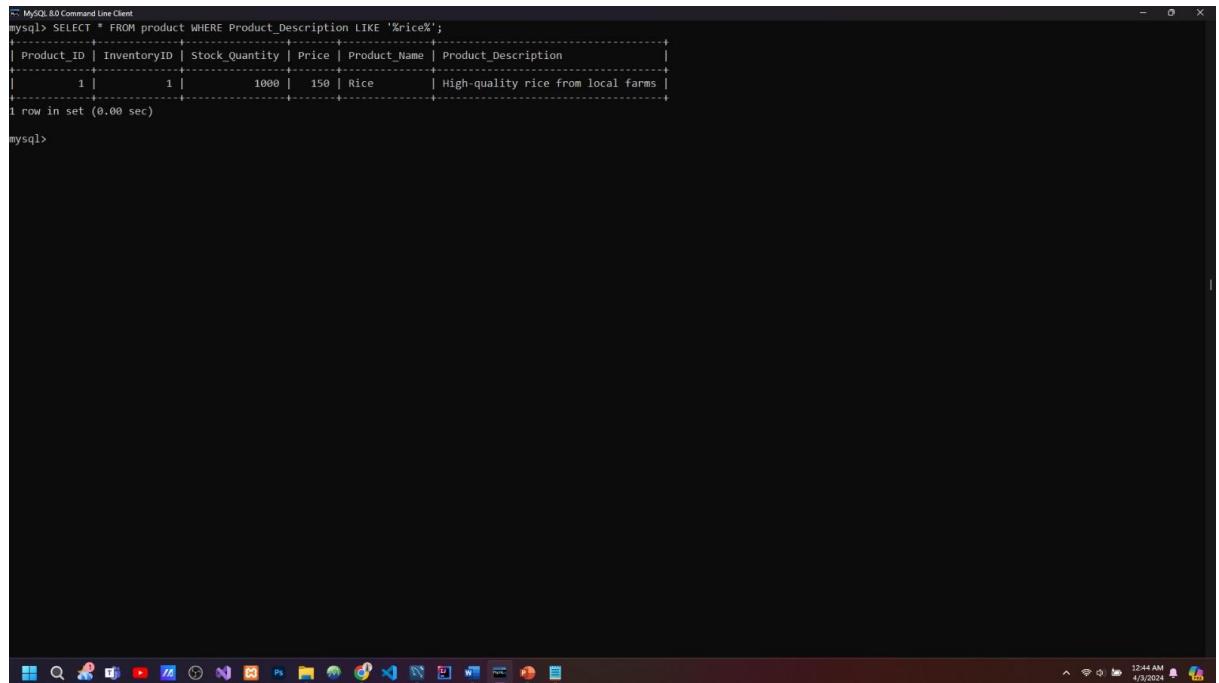
```
MySQL 8.0 Command Line Client

mysql> SELECT AVG(Stock_Quantity) AS AverageStock FROM inventory;
+-----+
| AverageStock |
+-----+
| 1050.0000 |
+-----+
1 row in set (0.00 sec)

mysql>
```

Figure 77. simple queries - aggregation function

#### 4.1.7. Demonstrating the use of LIKE keyword



MySQL 8.0 Command Line Client

```
mysql> SELECT * FROM product WHERE Product_Description LIKE '%rice%';
+-----+-----+-----+-----+-----+
| Product_ID | InventoryID | Stock_Quantity | Price | Product_Name | Product_Description |
+-----+-----+-----+-----+-----+
| 1 | 1 | 1000 | 150 | Rice | High-quality rice from local farms |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

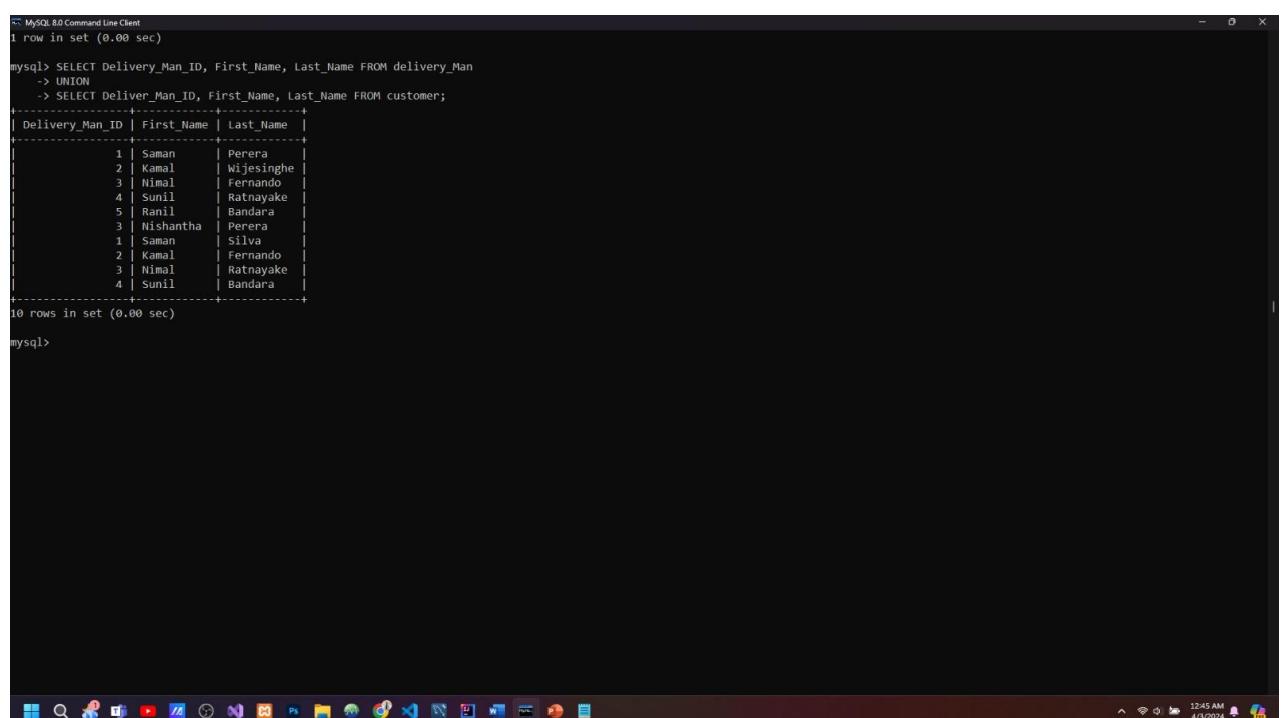
mysql>
```

Figure 78. simple queries - demonstrating the use of like keyword

#### 4.2. Complex queries

##### 4.2.1. Basic set operations (union, intersection, set difference, division) without user views

###### Basic set operation - Union



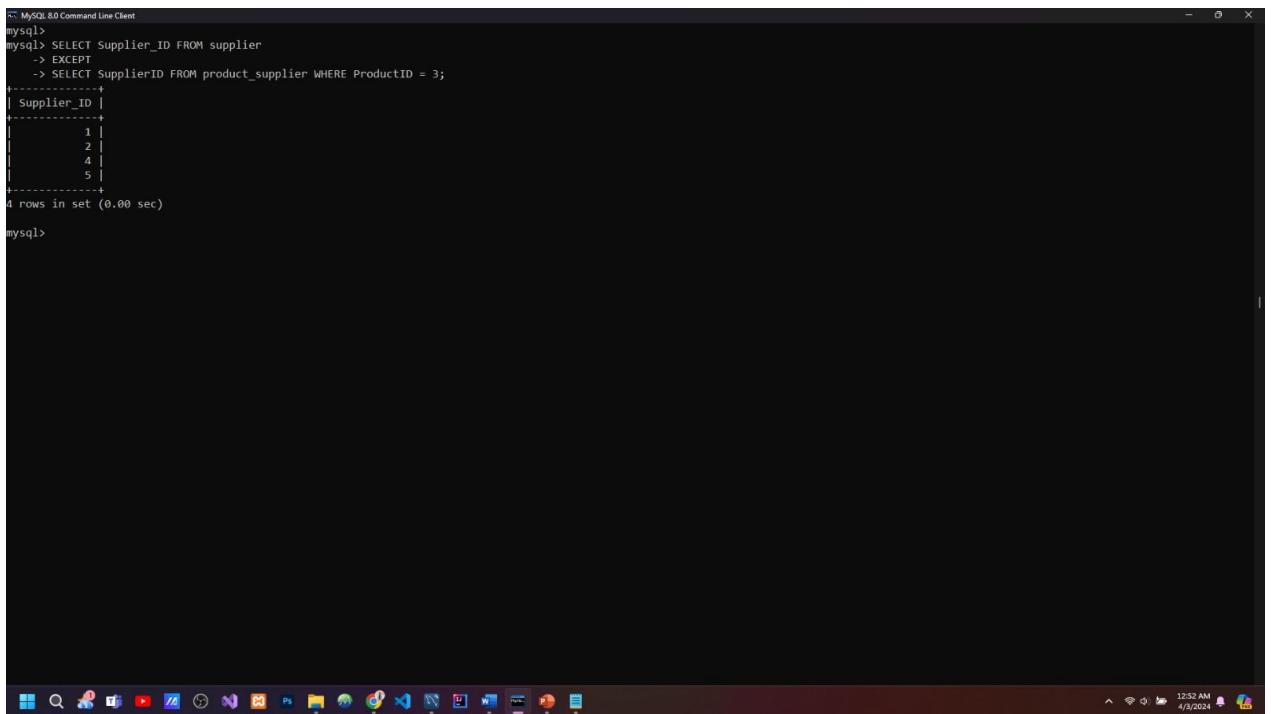
```
MySQL 8.0 Command Line Client
1 row in set (0.00 sec)

mysql> SELECT Delivery_Man_ID, First_Name, Last_Name FROM delivery_Man
    -> UNION
    -> SELECT Deliver_Man_ID, First_Name, Last_Name FROM customer;
+-----+-----+-----+
| Delivery_Man_ID | First_Name | Last_Name |
+-----+-----+-----+
| 1 | Saman | Perera |
| 2 | Kamal | Wijesinghe |
| 3 | Nimal | Fernando |
| 4 | Sunil | Ratnayake |
| 5 | Ranil | Bandara |
| 3 | Nishantha | Perera |
| 1 | Saman | Silva |
| 2 | Kamal | Fernando |
| 3 | Nimal | Ratnayake |
| 4 | Sunil | Bandara |
+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
```

Figure 79. Basic set operation - Union

## Basic set operation – Set Difference

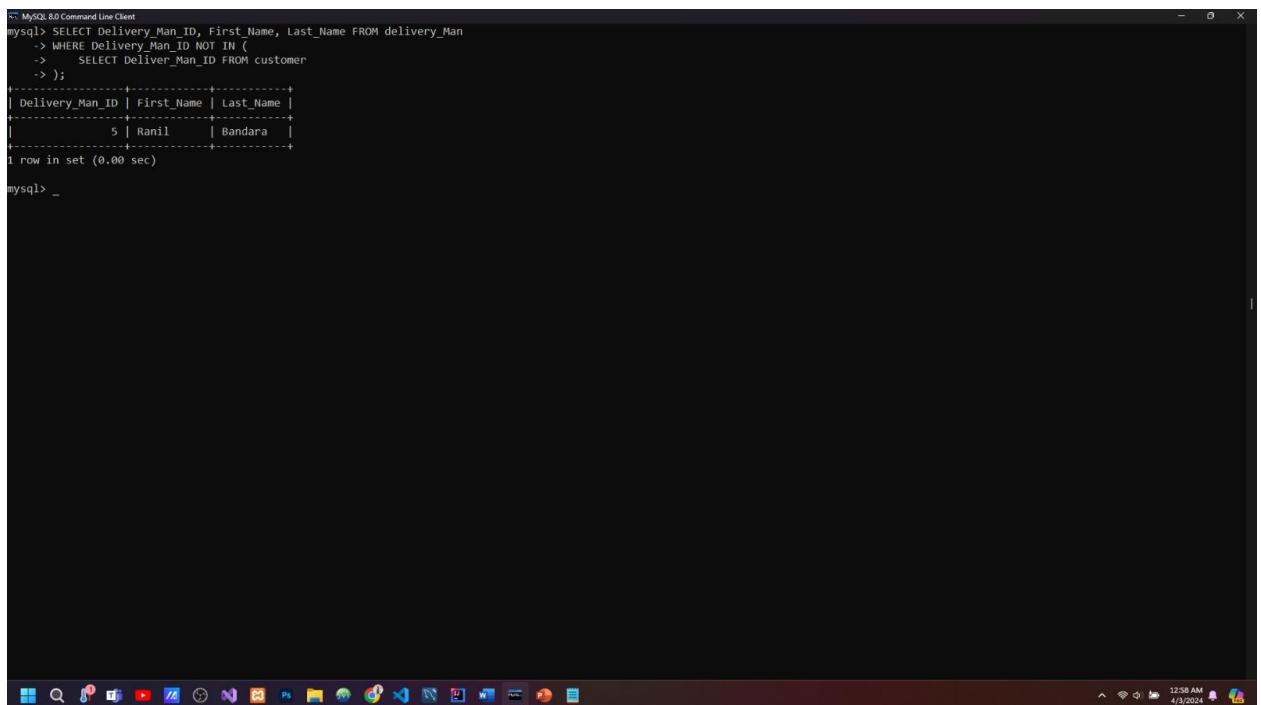


```
-- MySQL 8.0 Command Line Client
mysql> SELECT Supplier_ID FROM supplier
    -> EXCEPT
    ->   SELECT SupplierID FROM product_supplier WHERE ProductID = 3;
+-----+
| supplier_ID |
+-----+
|      1      |
|      2      |
|      4      |
|      5      |
+-----+
4 rows in set (0.00 sec)

mysql>
```

Figure 80. Basic set operation – Set Difference

## Basic set operation – Division



```
-- MySQL 8.0 Command Line Client
mysql> SELECT Delivery_Man_ID, First_Name, Last_Name FROM delivery_Man
    -> WHERE Delivery_Man_ID NOT IN (
    ->     SELECT Deliver_Man_ID FROM customer
    -> );
+-----+-----+-----+
| Delivery_Man_ID | First_Name | Last_Name |
+-----+-----+-----+
|      5          | Ranil     | Bandara   |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

Figure 81. Basic set operation – Division

## Basic set operation – Intersection (by using innerjoin)

The screenshot shows a Windows desktop with the MySQL 8.0 Command Line Client window open. The command entered is:

```
mysql> SELECT w.Warehouse_ID, w.Location  
    -> FROM warehouse w  
    -> INNER JOIN inventory i ON w.Warehouse_ID = i.WarehouseID;
```

The resulting table output is:

Warehouse_ID	Location
1	Colombo 03
1	Colombo 03
3	Galle
5	Matara

4 rows in set (0.00 sec)

mysql> \_

Figure 82. Basic set operation – Intersection

### 4.2.2. Inner join, natural join, left outer join, right outer join, full outer join, outer union relational algebraic operations by using user views

#### Inner join with User view

The screenshot shows a Windows desktop with the MySQL 8.0 Command Line Client window open. The commands entered are:

```
mysql> CREATE VIEW OrderCustomer AS  
    -> SELECT od.Order_ID, c.First_Name, c.Last_Name  
    -> FROM order_details od  
    -> INNER JOIN customer c ON od.CustomerID = c.Customer_ID;  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql>  
mysql> SELECT * FROM OrderCustomer;
```

The resulting table output is:

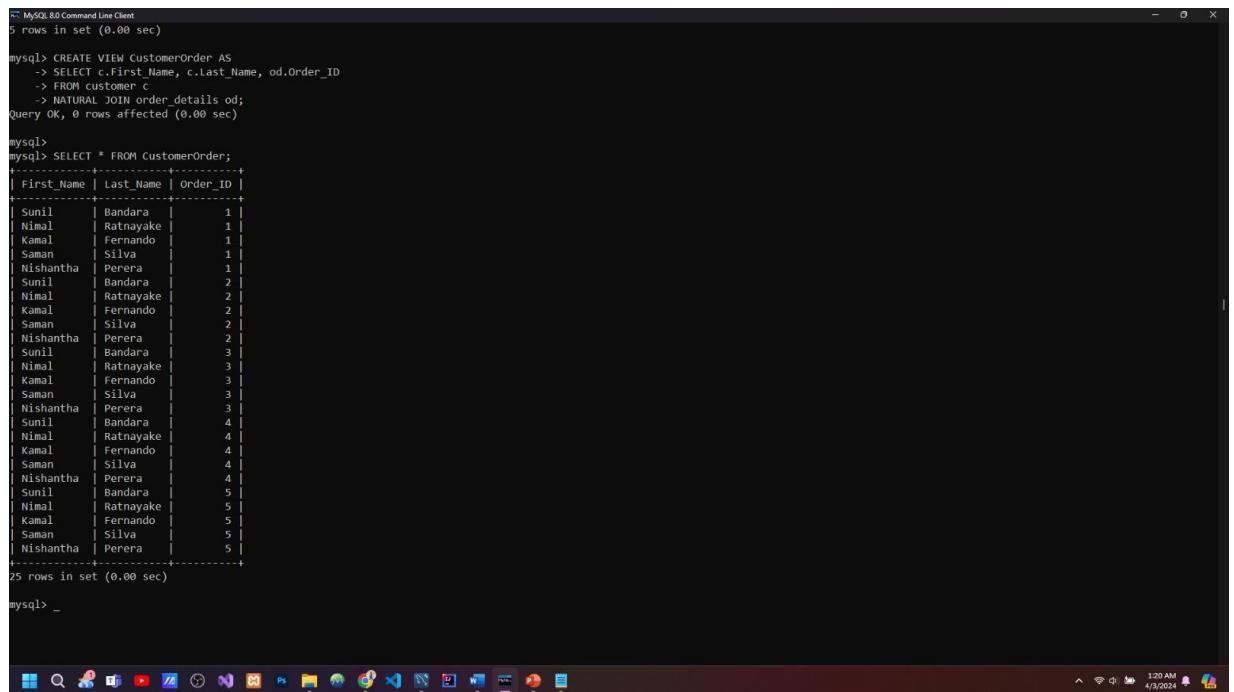
Order_ID	First_Name	Last_Name
1	Nishantha	Perera
2	Saman	Silva
3	Kamal	Fernando
4	Nimal	Ratnayake
5	Sunil	Bandara

5 rows in set (0.00 sec)

mysql> \_

Figure 83. Inner join with User view

## Natural join with User view



MySQL 8.0 Command Line Client  
5 rows in set (0.00 sec)

```
mysql> CREATE VIEW CustomerOrder AS
-> SELECT c.First_Name, c.Last_Name, od.Order_ID
-> FROM customer c
-> NATURAL JOIN order_details od;
Query OK, 0 rows affected (0.00 sec)

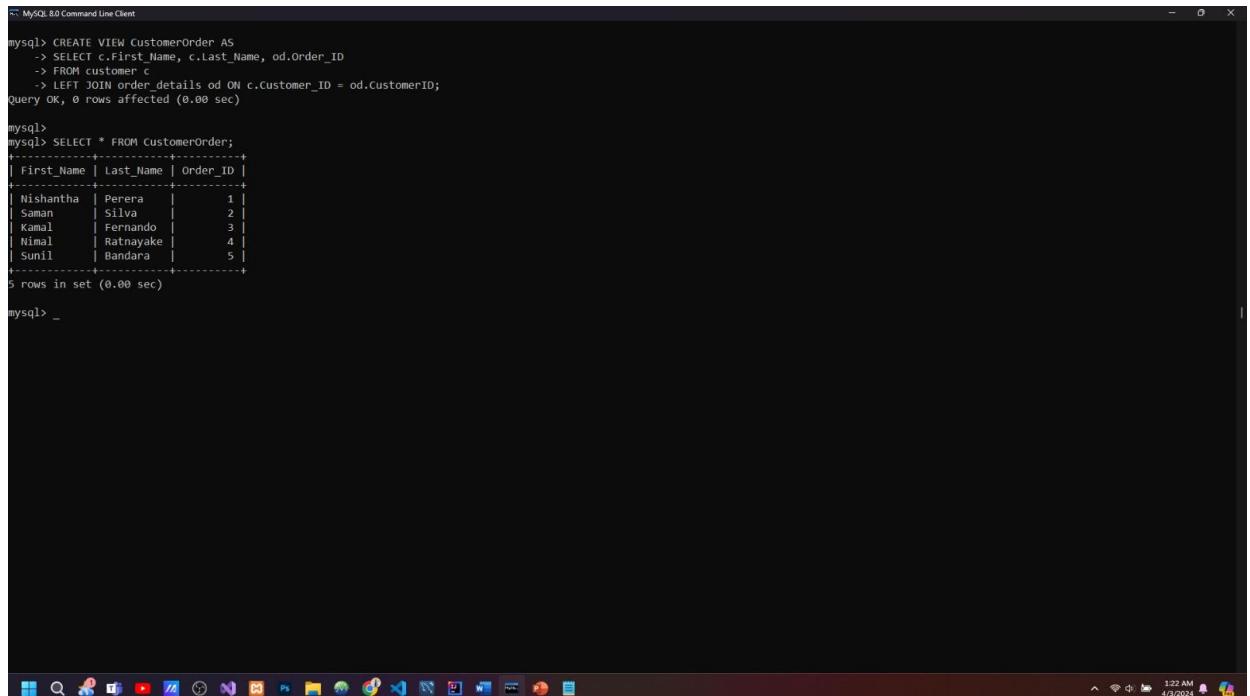
mysql> SELECT * FROM CustomerOrder;
+-----+-----+-----+
| First_Name | Last_Name | Order_ID |
+-----+-----+-----+
| Sunil      | Bandara   | 1        |
| Nimal      | Ratnayake | 1        |
| Kamal      | Fernando  | 1        |
| Saman      | Silva     | 1        |
| Nishantha  | Perera    | 1        |
| Sunil      | Bandara   | 2        |
| Nimal      | Ratnayake | 2        |
| Kamal      | Fernando  | 2        |
| Saman      | Silva     | 2        |
| Nishantha  | Perera    | 2        |
| Sunil      | Bandara   | 3        |
| Nimal      | Ratnayake | 3        |
| Kamal      | Fernando  | 3        |
| Saman      | Silva     | 3        |
| Nishantha  | Perera    | 3        |
| Sunil      | Bandara   | 4        |
| Nimal      | Ratnayake | 4        |
| Kamal      | Fernando  | 4        |
| Saman      | Silva     | 4        |
| Nishantha  | Perera    | 4        |
| Sunil      | Bandara   | 5        |
| Nimal      | Ratnayake | 5        |
| Kamal      | Fernando  | 5        |
| Saman      | Silva     | 5        |
| Nishantha  | Perera    | 5        |
+-----+-----+-----+
25 rows in set (0.00 sec)

mysql> _
```

The screenshot shows the MySQL Command Line Client interface. The command `CREATE VIEW CustomerOrder AS` is issued to create a view named `CustomerOrder` that selects `First\_Name`, `Last\_Name`, and `Order\_ID` from the `customer` and `order\_details` tables via a natural join. The resulting table has 25 rows, each containing a first name, last name, and order ID. The MySQL prompt `mysql>` appears at the bottom.

Figure 84. Natural join with User view

## Left Outer join with User view



MySQL 8.0 Command Line Client  
5 rows in set (0.00 sec)

```
mysql> CREATE VIEW CustomerOrder AS
-> SELECT c.First_Name, c.Last_Name, od.Order_ID
-> FROM customer c
-> LEFT JOIN order_details od ON c.Customer_ID = od.CustomerID;
Query OK, 0 rows affected (0.00 sec)

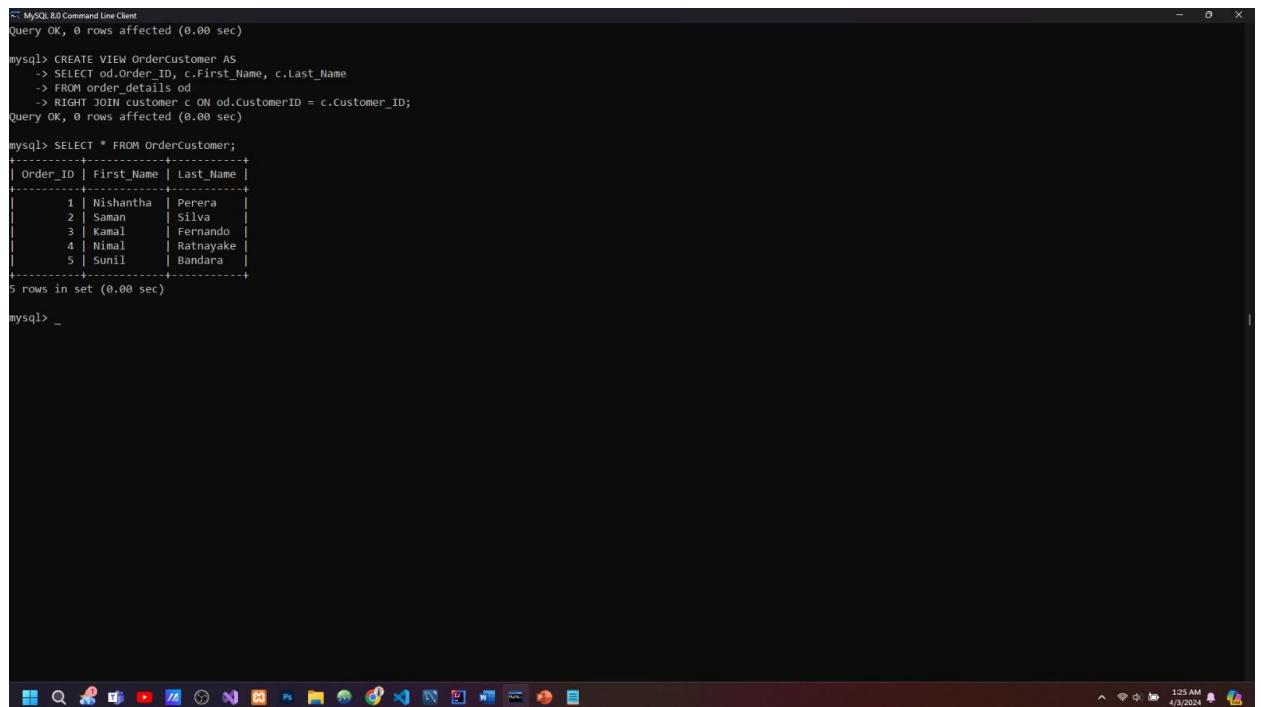
mysql> SELECT * FROM CustomerOrder;
+-----+-----+-----+
| First_Name | Last_Name | Order_ID |
+-----+-----+-----+
| Nishantha | Perera    | 1        |
| Saman     | Silva     | 2        |
| Kamal     | Fernando  | 3        |
| Nimal     | Ratnayake | 4        |
| Sunil     | Bandara   | 5        |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> _
```

The screenshot shows the MySQL Command Line Client interface. The command `CREATE VIEW CustomerOrder AS` is issued to create a view named `CustomerOrder` that selects `First\_Name`, `Last\_Name`, and `Order\_ID` from the `customer` and `order\_details` tables via a left outer join on the `Customer\_ID` column. The resulting table has 5 rows, each containing a first name, last name, and order ID. The MySQL prompt `mysql>` appears at the bottom.

Figure 85. Left Outer join with User view

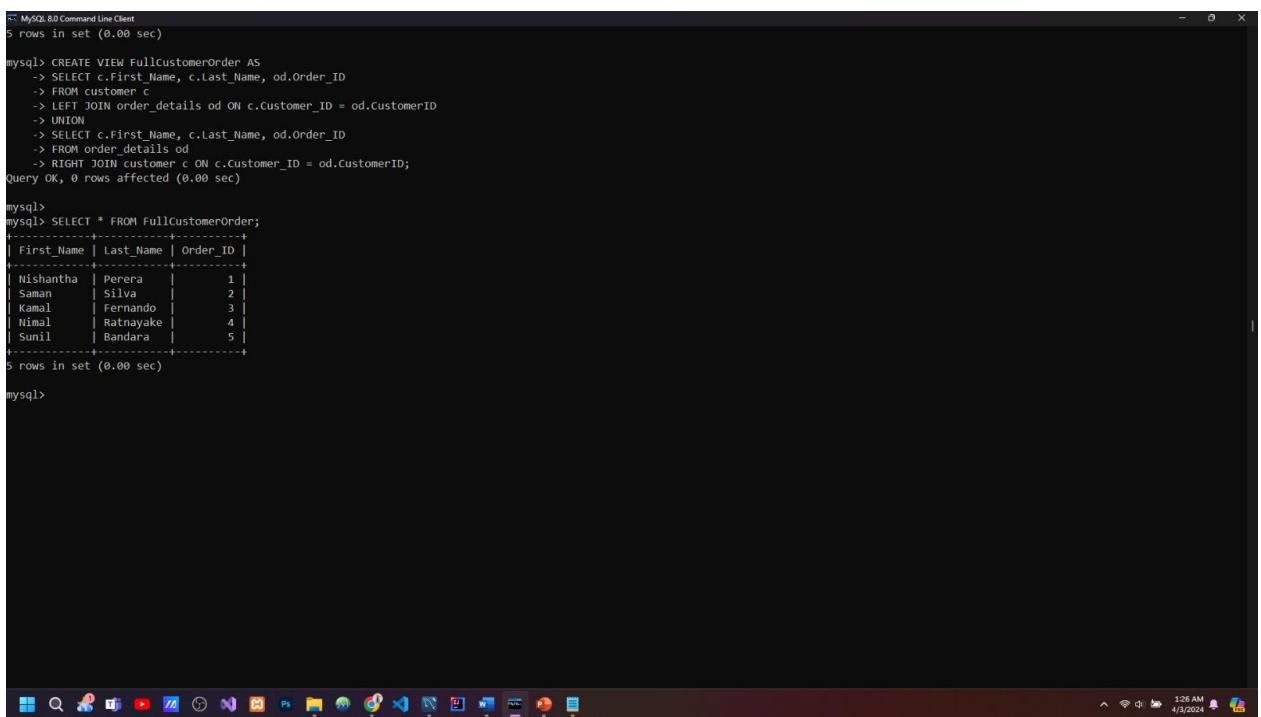
## Right Outer join with User view



MySQL 8.0 Command Line Client  
Query OK, 0 rows affected (0.00 sec)  
mysql> CREATE VIEW OrderCustomer AS  
-> SELECT od.Order\_ID, c.First\_Name, c.Last\_Name  
-> FROM order\_details od  
-> RIGHT JOIN customer c ON od.CustomerID = c.Customer\_ID;  
Query OK, 0 rows affected (0.00 sec)  
mysql> SELECT \* FROM OrderCustomer;  
+-----+-----+-----+  
| Order\_ID | First\_Name | Last\_Name |  
+-----+-----+-----+  
1	Nishantha	Perera
2	Saman	Silva
3	Kamal	Fernando
4	Nimal	Ratnayake
5	Sunil	Bandara
+-----+-----+-----+  
5 rows in set (0.00 sec)  
mysql> \_

Figure 86. Right Outer join with User view

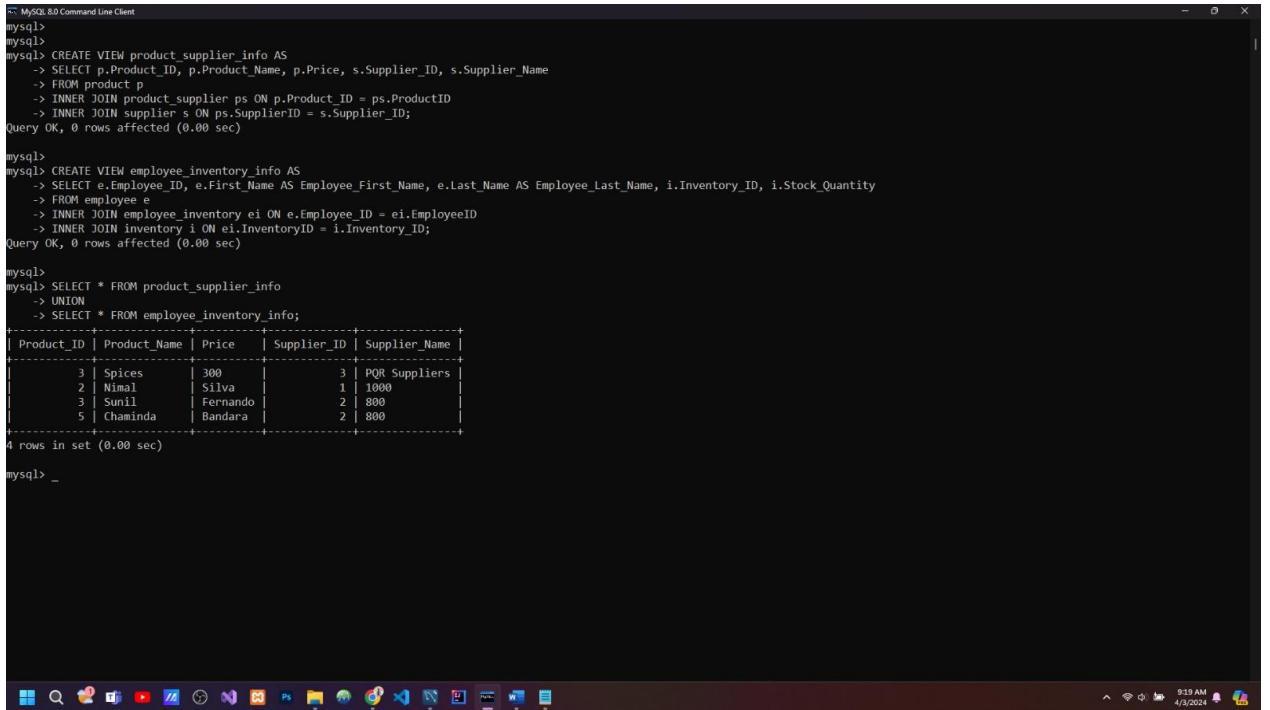
## Full outer join with user view



MySQL 8.0 Command Line Client  
5 rows in set (0.00 sec)  
mysql> CREATE VIEW FullCustomerOrder AS  
-> SELECT c.First\_Name, c.Last\_Name, od.Order\_ID  
-> FROM customer c  
-> LEFT JOIN order\_details od ON c.Customer\_ID = od.CustomerID  
-> UNION  
-> SELECT c.First\_Name, c.Last\_Name, od.Order\_ID  
-> FROM order\_details od  
-> RIGHT JOIN customer c ON c.Customer\_ID = od.CustomerID;  
Query OK, 0 rows affected (0.00 sec)  
mysql> SELECT \* FROM FullCustomerOrder;  
+-----+-----+-----+  
| First\_Name | Last\_Name | Order\_ID |  
+-----+-----+-----+  
Nishantha	Perera	1
Saman	Silva	2
Kamal	Fernando	3
Nimal	Ratnayake	4
Sunil	Bandara	5
+-----+-----+-----+  
5 rows in set (0.00 sec)  
mysql> \_

Figure 87. Full outer join with user view

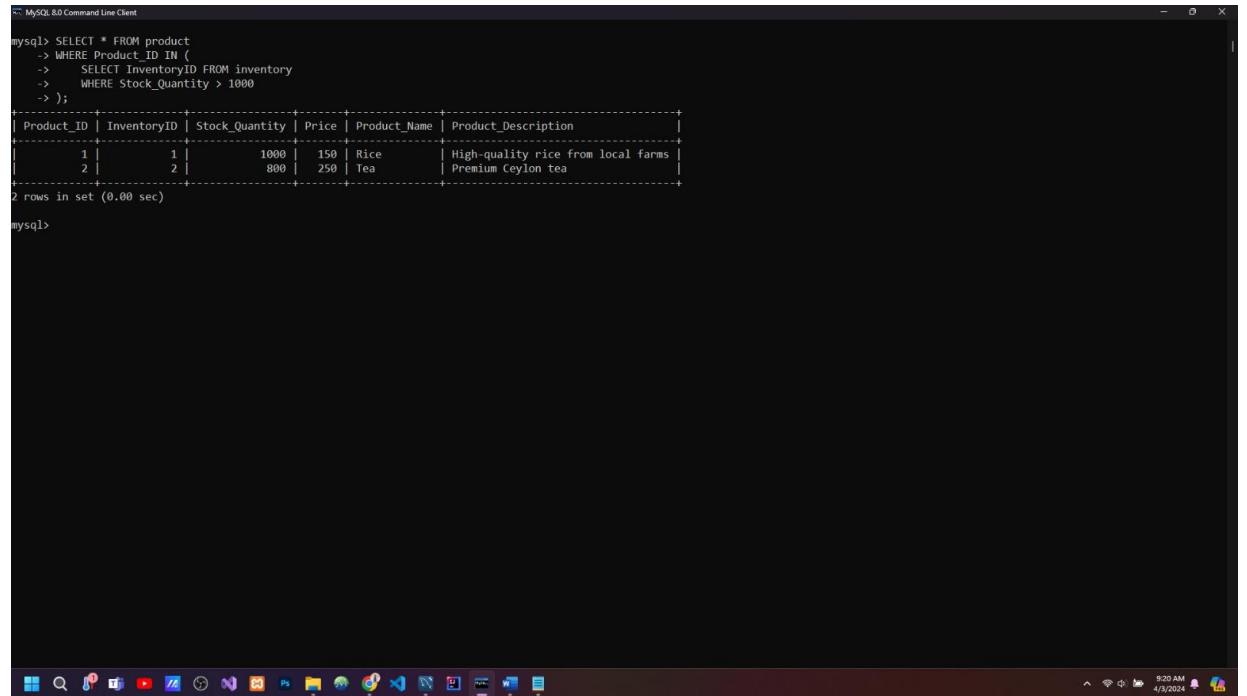
## Outer Union with User view



MySQL 8.0 Command Line Client  
mysql> CREATE VIEW product\_supplier\_info AS  
-> SELECT p.Product\_ID, p.Product\_Name, p.Price, s.Supplier\_ID, s.Supplier\_Name  
-> FROM product p  
-> INNER JOIN product\_supplier ps ON p.Product\_ID = ps.ProductID  
-> INNER JOIN supplier s ON ps.SupplierID = s.Supplier\_ID;  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> CREATE VIEW employee\_inventory\_info AS  
-> SELECT e.Employee\_ID, e.First\_Name AS Employee\_First\_Name, e.Last\_Name AS Employee\_Last\_Name, i.Inventory\_ID, i.Stock\_Quantity  
-> FROM employee e  
-> INNER JOIN employee\_inventory ei ON e.Employee\_ID = ei.EmployeeID  
-> INNER JOIN inventory i ON ei.InventoryID = i.Inventory\_ID;  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> SELECT \* FROM product\_supplier\_info  
-> UNION  
-> SELECT \* FROM employee\_inventory\_info;  
+-----+-----+-----+-----+  
| Product\_ID | Product\_Name | Price | Supplier\_ID | Supplier\_Name |  
+-----+-----+-----+-----+  
3	Spices	300	3	PQR Suppliers
2	Nimal	Silva	1	1000
3	Sunil	Fernando	2	800
5	Chaminda	Bandara	2	800
+-----+-----+-----+-----+  
4 rows in set (0.00 sec)  
  
mysql> \_

Figure 88. Outer Union with User view

## Nested queries



MySQL 8.0 Command Line Client  
mysql> SELECT \* FROM product  
-> WHERE Product\_ID IN (  
-> SELECT InventoryID FROM inventory  
-> WHERE Stock\_Quantity > 1000  
-> );  
+-----+-----+-----+-----+-----+  
| Product\_ID | InventoryID | Stock\_Quantity | Price | Product\_Name | Product\_Description |  
+-----+-----+-----+-----+-----+  
| 1 | 1 | 1000 | 150 | Rice | High-quality rice from local farms |  
| 2 | 2 | 800 | 250 | Tea | Premium Ceylon tea |  
+-----+-----+-----+-----+-----+  
2 rows in set (0.00 sec)  
  
mysql> \_

Figure 89. Nested queries\_1

```
-- MySQL 8.0 Command Line Client
2 rows in set (0.00 sec)

mysql> SELECT * FROM employee
   -> WHERE Employee_ID IN (
   ->     SELECT e.Employee_ID FROM employee e
   ->     INNER JOIN warehouse w ON e.Employee_ID = w.Warehouse_ID
   ->     WHERE w.Capacity < 3000
   -> );
+-----+-----+-----+-----+
| Employee_ID | First_Name | Last_Name | DOB      | Manager_ID |
+-----+-----+-----+-----+
|        4    | Ranil     | Ratnayake | 1978-02-15 |         2   |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

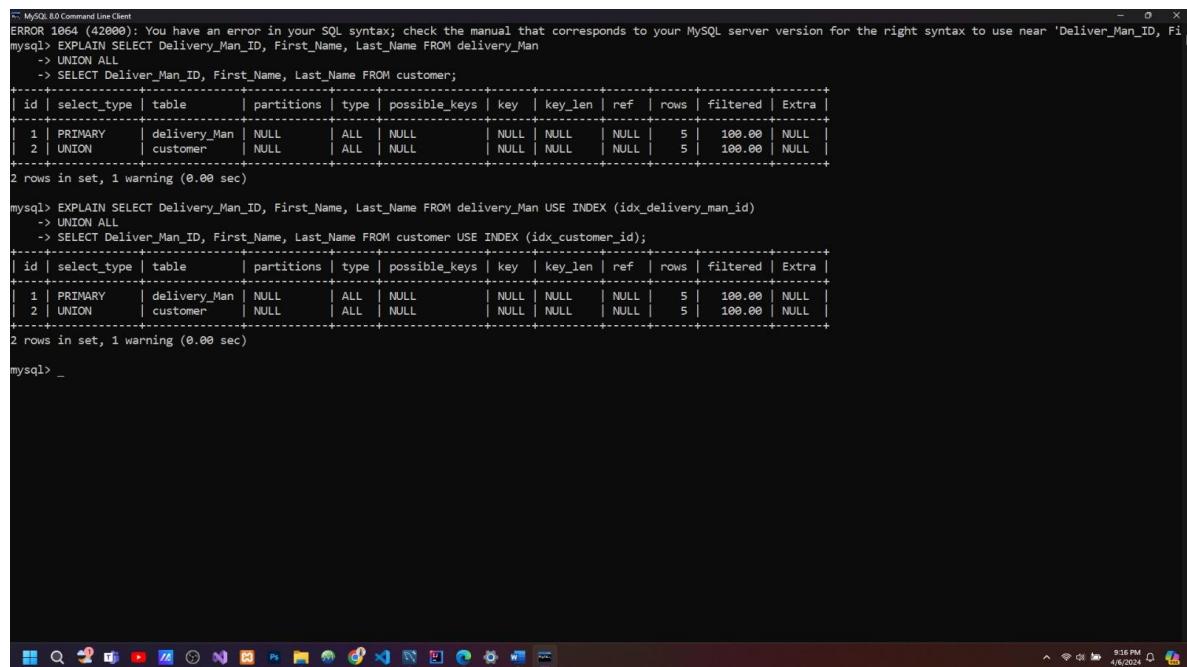
Figure 90. Nested queries\_2

```
-- MySQL 8.0 Command Line Client
mysql> SELECT * FROM customer
   -> WHERE Customer_ID IN (
   ->     SELECT CustomerID FROM order_details
   ->     WHERE Total_Amount > 2000
   -> );
+-----+-----+-----+-----+
| Customer_ID | First_Name | Last_Name | DOB      | Deliver_Man_ID |
+-----+-----+-----+-----+
|        1    | Mishantha | Perera   | 1990-05-15 |             3   |
|        3    | Kamal     | Fernando  | 1982-11-10 |             2   |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> _
```

Figure 91. Nested queries\_3

# Chapter 5 – Database Tuning

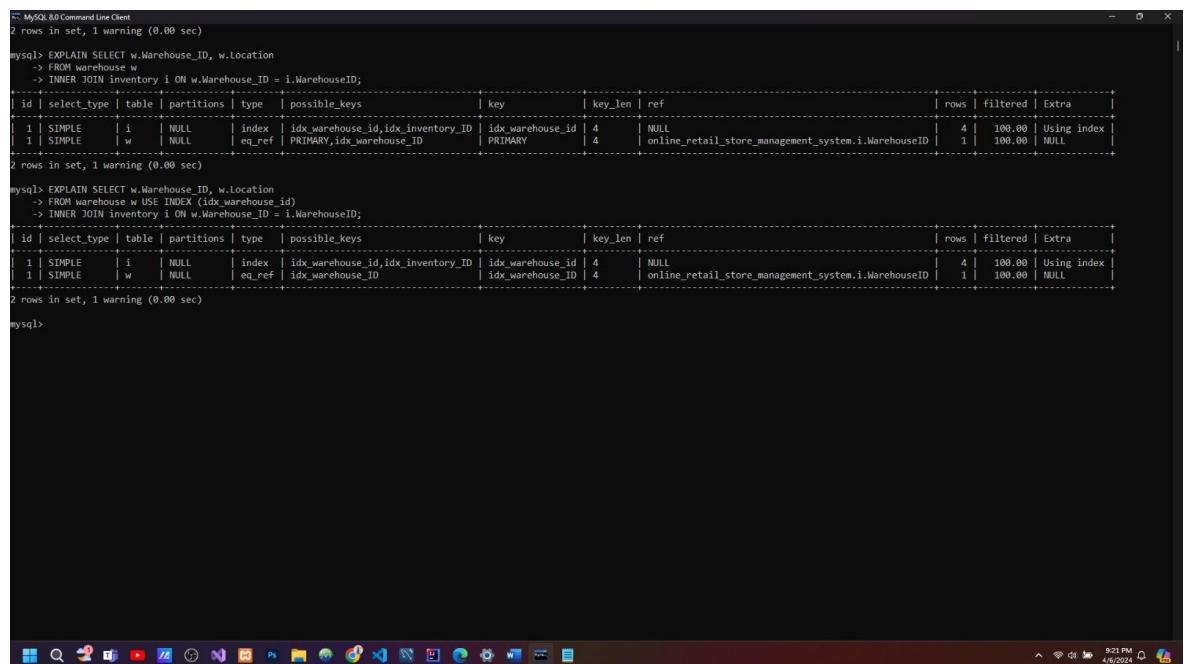


```
MySQL 8.0 Command Line Client
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'Deliver_Man_ID, Fi
mysql> EXPLAIN SELECT Deliver_Man_ID, First_Name, Last_Name FROM delivery_Man
   >- UNION ALL
   >- SELECT Deliver_Man_ID, First_Name, Last_Name FROM customer;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table      | partitions | type  | possible_keys | key    | key_len | ref   | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  | PRIMARY     | delivery_Man | NULL      | ALL   | NULL          | NULL   | NULL    | NULL   | 5    | 100.00  | NULL   |
| 2  | UNION        | customer    | NULL      | ALL   | NULL          | NULL   | NULL    | NULL   | 5    | 100.00  | NULL   |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql> EXPLAIN SELECT Delivery_Man_ID, First_Name, Last_Name FROM delivery_Man USE INDEX (idx_delivery_man_id)
   >- UNION ALL
   >- SELECT Delivery_Man_ID, First_Name, Last_Name FROM customer USE INDEX (idx_customer_id);
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table      | partitions | type  | possible_keys | key    | key_len | ref   | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  | PRIMARY     | delivery_Man | NULL      | ALL   | NULL          | NULL   | NULL    | NULL   | 5    | 100.00  | NULL   |
| 2  | UNION        | customer    | NULL      | ALL   | NULL          | NULL   | NULL    | NULL   | 5    | 100.00  | NULL   |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>
```

Figure 92. Database Tuning Figure\_1



```
MySQL 8.0 Command Line Client
2 rows in set, 1 warning (0.00 sec)

mysql> EXPLAIN SELECT w.Warehouse_ID, w.location
   > FROM warehouse w
   >- INNER JOIN Inventory i ON w.Warehouse_ID = i.WarehouseID;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type  | possible_keys | key    | key_len | ref   | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  | SIMPLE     | i      | NULL      | index | idx_warehouse_id, idx_inventory_ID | idx_warehouse_id | 4    | NULL   | 4    | 100.00  | Using index |
| 1  | SIMPLE     | w      | NULL      | eq_ref | PRIMARY, idx_warehouse_ID          | PRIMARY          | 4    | online_retail_store_management_system.i.WarehouseID | 1    | 100.00  | NULL   |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql> EXPLAIN SELECT w.Warehouse_ID, w.location
   > FROM warehouse w USE INDEX (idx_warehouse_id)
   >- INNER JOIN Inventory i ON w.Warehouse_ID = i.WarehouseID;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type  | possible_keys | key    | key_len | ref   | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  | SIMPLE     | i      | NULL      | index | idx_warehouse_id, idx_inventory_ID | idx_warehouse_id | 4    | NULL   | 4    | 100.00  | Using index |
| 1  | SIMPLE     | w      | NULL      | eq_ref | idx_warehouse_id                  | idx_warehouse_id | 4    | online_retail_store_management_system.i.WarehouseID | 1    | 100.00  | NULL   |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>
```

Figure 93. Database Tuning Figure\_2

```

-- MySQL 8.0 Command Line Client
2 rows in set, 1 warning (0.00 sec)

mysql> EXPLAIN (
    ->     SELECT Delivery_Man_ID, First_Name, Last_Name FROM delivery_Man
    ->     UNION
    ->     EXCEPT
    ->     (
    ->         SELECT Deliver_Man_ID, First_Name, Last_Name FROM customer
    ->     );
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY   | delivery_Man | NULL | ALL | NULL | NULL | NULL | NULL | 5 | 100.00 | NULL |
| 2 | EXCEPT    | customer    | NULL | ALL | NULL | NULL | NULL | NULL | 5 | 100.00 | NULL |
| 3 | EXCEPT RESULT | <except1,2> | NULL | ALL | NULL | NULL | NULL | NULL | NULL | NULL | Using temporary |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set, 1 warning (0.00 sec)

mysql> EXPLAIN (
    ->     SELECT Delivery_Man_ID, First_Name, Last_Name FROM delivery_Man USE INDEX (idx_delivery_man_id)
    ->     UNION
    ->     EXCEPT
    ->     (
    ->         SELECT Deliver_Man_ID, First_Name, Last_Name FROM customer USE INDEX (idx_customer_id)
    ->     );
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY   | delivery_Man | NULL | ALL | NULL | NULL | NULL | NULL | 5 | 100.00 | NULL |
| 2 | EXCEPT    | customer    | NULL | ALL | NULL | NULL | NULL | NULL | 5 | 100.00 | NULL |
| 3 | EXCEPT RESULT | <except1,2> | NULL | ALL | NULL | NULL | NULL | NULL | NULL | NULL | Using temporary |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set, 1 warning (0.00 sec)

mysql> _

```

Figure 94. Database Tuning Figure\_3

```

-- MySQL 8.0 Command Line Client
mysql>
mysql> EXPLAIN SELECT Delivery_Man_ID, First_Name, Last_Name FROM delivery_Man
    -> WHERE Delivery_Man_ID NOT IN (
    ->     SELECT Deliver_Man_ID FROM customer
    -> );
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE    | delivery_Man | NULL | ALL | NULL | NULL | NULL | NULL | 5 | 100.00 | NULL |
| 2 | SIMPLE    | customer    | NULL | ALL | NULL | NULL | NULL | fk_deliver | 4 | online_retail_store_management_system.delivery_Man.Delivery_Man_ID |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql> EXPLAIN SELECT Delivery_Man_ID, First_Name, Last_Name FROM delivery_Man USE INDEX (idx_delivery_man_id)
    -> WHERE Delivery_Man_ID NOT IN (
    ->     SELECT Deliver_Man_ID FROM customer USE INDEX (idx_customer_id)
    -> );
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE    | delivery_Man | NULL | ALL | NULL | NULL | NULL | NULL | 5 | 100.00 | NULL |
| 2 | SIMPLE    | customer    | NULL | ALL | NULL | NULL | NULL | fk_deliver | 4 | online_retail_store_management_system.delivery_Man.Delivery_Man_ID |
| 3 | MATERIALIZED | <subquery2> | NULL | ALL | auto_distinct_keys | auto_distinct_key | NULL | NULL | 1 | 100.00 | Using where; Not exists |
| 4 | EXPLAIN   | customer    | NULL | ALL | NULL | NULL | NULL | NULL | 5 | 100.00 | NULL |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set, 1 warning (0.00 sec)

mysql> _

```

Figure 95. Database Tuning Figure\_4

```

-- MySQL 8.0 Command Line Client
4 rows in set, 1 warning (0.00 sec)

mysql> EXPLAIN SELECT w.Warehouse_ID, w.location
   > FROM warehouse w
   > INNER JOIN inventory i ON w.Warehouse_ID = i.WarehouseID;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE     | i      | NULL       | Index | idx_warehouse_id, idx_inventory_ID | idx_warehouse_id | 4      | NULL |
| 1 | SIMPLE     | w      | NULL       | eq_ref | PRIMARY, idx_warehouse_id          | PRIMARY          | 4      | online_retail_store_management_system.i.WarehouseID |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql> CREATE INDEX idx_inventory_ID ON inventory(WarehouseID);
ERROR 1061 (42000): Duplicate key name 'idx_inventory_ID'
mysql> CREATE INDEX idx_warehouse_ID ON warehouse(WarehouseID);
ERROR 1061 (42000): Duplicate key name 'idx_warehouse_ID'
mysql> SELECT w.Warehouse_ID, w.Location
   > FROM warehouse w
   > INTERSECT
   > SELECT i.WarehouseID, w.location
   > FROM inventory i
   > INNER JOIN warehouse w ON i.WarehouseID = w.Warehouse_ID;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY    | w      | NULL       | ALL    | NULL          | NULL | NULL    | NULL |
| 2 | INTERSECT  | i      | NULL       | Index  | idx_warehouse_id, idx_inventory_ID | idx_warehouse_id | 4      | NULL |
| 2 | INTERSECT  | w      | NULL       | eq_ref  | PRIMARY, idx_warehouse_id          | PRIMARY          | 4      | online_retail_store_management_system.i.WarehouseID |
| 3 | INTERSECT RESULT | <intersect1,2> | NULL    | ALL    | NULL          | NULL | NULL    | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set, 1 warning (0.00 sec)

mysql>

```

Figure 96. Database Tuning Figure\_5

```

-- MySQL 8.0 Command Line Client
2 rows in set, 2 warnings (0.00 sec)

mysql> EXPLAIN SELECT e.Employee_ID, e.First_Name, e.Last_Name
   > FROM employee e
   > WHERE e.EmployeeID =
   >   (SELECT 1
   >   FROM employee_inventory ei
   >   WHERE ei.EmployeeID = e.Employee_ID
   >   );
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE     | el    | NULL       | index | PRIMARY, idx_employee_inventory_EmployeeID | idx_employee_inventory_EmployeeID | 4      | NULL |
| 1 | SIMPLE     | e     | NULL       | eq_ref | PRIMARY          | PRIMARY          | 4      | online_retail_store_management_system.el.EmployeeID |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 2 warnings (0.00 sec)

mysql> CREATE INDEX idx_employee_inventory_EmployeeID ON employee_inventory(EmployeeID);
ERROR 1061 (42000): Duplicate key name 'idx_employee_inventory_EmployeeID'
mysql> EXPLAIN SELECT DISTINCT e.Employee_ID, e.First_Name, e.Last_Name
   > FROM employee e
   > INNER JOIN employee_inventory ei ON e.Employee_ID = ei.EmployeeID;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE     | el    | NULL       | index | PRIMARY, idx_employee_inventory_EmployeeID | fk_inven | 4      | NULL |
| 1 | SIMPLE     | e     | NULL       | eq_ref | PRIMARY          | PRIMARY          | 4      | online_retail_store_management_system.el.EmployeeID |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql> _

```

Figure 97. Database Tuning Figure\_6

```

MySQL 8.0 Command Line Client
  SELECT InventoryID ` at line 1
mysql> EXPLAIN SELECT * FROM product
   > WHERE Product_ID IN (
   >   ->   SELECT InventoryID FROM inventory
   >   ->   WHERE Stock_Quantity > 1000
   > );
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE     | product |      NULL | ALL  |    NULL       | NULL |    NULL | NULL |
| 1 | SIMPLE     | inventory |      NULL | ALL  |    NULL       | NULL |    NULL | NULL |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 2 warnings (0.00 sec)

mysql> EXPLAIN SELECT * FROM product USE INDEX (PRIMARY) -- Use the primary key index
   > WHERE Product_ID IN (
   >   ->   SELECT InventoryID FROM inventory USE INDEX (PRIMARY) -- Use the primary key index of inventory table
   >   ->   WHERE Stock_Quantity > 1000
   > );
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE     | product |      NULL | ALL  |    NULL       | NULL |    NULL | NULL |
| 1 | SIMPLE     | inventory |      NULL | ALL  |    NULL       | NULL |    NULL | NULL |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 2 warnings (0.00 sec)

mysql> _

```

Figure 98. Database Tuning Figure\_7

```

MySQL 8.0 Command Line Client
 2 rows in set, 2 warnings (0.00 sec)

mysql> EXPLAIN SELECT * FROM employee
   > WHERE Employee_ID IN (
   >   ->   SELECT e.Employee_ID FROM employee e
   >   ->   INNER JOIN warehouse w ON e.Employee_ID = w.Warehouse_ID
   >   ->   WHERE w.Capacity < 3000
   > );
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE     | w    |      NULL | ALL  | PRIMARY,Idx_warehouse_ID | PRIMARY | 4       | eq_ref |
| 1 | SIMPLE     | employee |      NULL | eq_ref | PRIMARY          | PRIMARY | 4       | eq_ref |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set, 1 warning (0.00 sec)

mysql> EXPLAIN SELECT * FROM employee USE INDEX (PRIMARY)
   > WHERE Employee_ID IN (
   >   ->   SELECT e.Employee_ID FROM employee e USE INDEX (PRIMARY)
   >   ->   INNER JOIN warehouse w ON e.Employee_ID = w.Warehouse_ID
   >   ->   WHERE w.Capacity < 3000
   > );
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE     | w    |      NULL | ALL  | PRIMARY,Idx_warehouse_ID | PRIMARY | 4       | NULL |
| 1 | SIMPLE     | employee |      NULL | eq_ref | PRIMARY          | PRIMARY | 4       | eq_ref |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set, 1 warning (0.00 sec)

mysql>

```

Figure 99. Database Tuning Figure\_8

```

-- MySQL 8.0 Command Line Client

mysql> EXPLAIN SELECT * FROM product_supplier_info
--> UNION
--> SELECT * FROM employee.inventory_info;
+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys |
+-----+-----+-----+-----+-----+
| 1 | PRIMARY | ps | NULL | index | PRIMARY,fk_supplier |
| 1 | PRIMARY | p | NULL | eq_ref | PRIMARY |
| 1 | PRIMARY | s | NULL | eq_ref | PRIMARY |
| 2 | UNION | ei | NULL | Index | PRIMARY,fk_inven,IDX_employee_inventory_EmployeeID |
| 2 | UNION | i | NULL | eq_ref | PRIMARY |
| 2 | UNION | e | NULL | eq_ref | PRIMARY |
| 3 | UNION RESULT | union1_2 | NULL | ALL | NULL |
+-----+-----+-----+-----+-----+
7 rows in set, 1 warning (0.00 sec)

mysql> EXPLAIN SELECT * FROM product_supplier_info -- Use the appropriate index name for product_supplier_info table
--> UNION
--> SELECT * FROM employee.inventory_info;
+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys |
+-----+-----+-----+-----+-----+
| 1 | PRIMARY | ps | NULL | index | PRIMARY,fk_supplier |
| 1 | PRIMARY | p | NULL | eq_ref | PRIMARY |
| 1 | PRIMARY | s | NULL | eq_ref | PRIMARY |
| 2 | UNION | ei | NULL | Index | PRIMARY,fk_inven,IDX_employee_inventory_EmployeeID |
| 2 | UNION | i | NULL | eq_ref | PRIMARY |
| 2 | UNION | e | NULL | eq_ref | PRIMARY |
| 3 | UNION RESULT | union1_2 | NULL | ALL | NULL |
+-----+-----+-----+-----+-----+
7 rows in set, 1 warning (0.00 sec)

mysql>

```

Figure 100. Database Tuning Figure\_9

```

-- MySQL 8.0 Command Line Client

mysql> CREATE VIEW CustomerOrder3 AS
--> SELECT c.First_Name, c.Last_Name, od.Order_ID
--> FROM customer c
--> LEFT JOIN order_details od ON c.Customer_ID = od.CustomerID;
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> EXPLAIN SELECT * FROM CustomerOrder3;
+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys |
+-----+-----+-----+-----+-----+
| 1 | SIMPLE | c | NULL | ALL | NULL |
| 1 | SIMPLE | od | NULL | ref | idx_order_details_CustomerID |
+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql> EXPLAIN SELECT c.First_Name, c.Last_Name, od.Order_ID
--> FROM customer c
--> LEFT JOIN order_details od ON c.Customer_ID = od.CustomerID;
+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys |
+-----+-----+-----+-----+-----+
| 1 | SIMPLE | c | NULL | ALL | NULL |
| 1 | SIMPLE | od | NULL | ref | idx_order_details_CustomerID |
+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql> _

```

Figure 101. Database Tuning Figure\_10

```

MySQL 8.0 Command Line Client
mysql> CREATE VIEW FullCustomerOrder1 AS
-> SELECT c.FirstName, c.LastName, od.Order_ID
-> FROM customer c
-> LEFT JOIN order_details od ON c.Customer_ID = od.CustomerID
-> UNION
-> SELECT c.FirstName, c.LastName, od.Order_ID
-> FROM order_details od
-> RIGHT JOIN customer c ON c.Customer_ID = od.CustomerID;
Query OK, 0 rows affected (0.00 sec)

mysql> EXPLAIN SELECT * FROM FullCustomerOrder1;
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | <derived2> | NULL | ALL | NULL | NULL | NULL | NULL |
| 2 | DERIVED | c | NULL | ALL | NULL | NULL | NULL | NULL |
| 3 | DERIVED | od | NULL | ref | idx_order_details_CustomerID | idx_order_details_CustomerID | 4 | NULL |
| 4 | UNION | c | NULL | ALL | NULL | NULL | NULL | NULL |
| 5 | UNION | od | NULL | ref | idx_order_details_CustomerID | idx_order_details_CustomerID | 4 | NULL |
| 6 | UNION RESULT | union1,3> | NULL | ALL | NULL | NULL | NULL | NULL |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set, 1 warning (0.00 sec)

mysql> EXPLAIN (SELECT c.FirstName, c.LastName, od.Order_ID
-> FROM customer c
-> LEFT JOIN order_details od ON c.Customer_ID = od.CustomerID
-> UNION
-> SELECT c.FirstName, c.LastName, od.Order_ID
-> FROM order_details od
-> RIGHT JOIN customer c ON c.Customer_ID = od.CustomerID);
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | c | NULL | ALL | NULL | NULL | NULL | NULL |
| 2 | PRIMARY | od | NULL | ref | idx_order_details_CustomerID | idx_order_details_CustomerID | 4 | NULL |
| 3 | UNION | c | NULL | ALL | NULL | NULL | NULL | NULL |
| 4 | UNION | od | NULL | ref | idx_order_details_CustomerID | idx_order_details_CustomerID | 4 | NULL |
| 5 | UNION RESULT | union1,2> | NULL | ALL | NULL | NULL | NULL | NULL |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set, 1 warning (0.00 sec)

mysql> -

```

Figure 102. Database Tuning Figure\_11