



# Database Systems Mini Project

A report submitted to the

Department of Electrical and Information Engineering  
Faculty of Engineering  
University of Ruhuna  
Sri Lanka

On 11<sup>th</sup> of September 2023

In completing an assignment for the module EE4350  
Database Systems

By

EG/2020/3903	:	Dissanayaka D.M.C.L.
EG/2020/3905	:	Dissanayaka D.M.S.C.
EG/2020/3909	:	Dissanayaka R.P.L.M.

## Table of Content

Table of Content.....	2
List of Figures .....	3
1 Chapter 1 - Requirement Analysis.....	5
1.1 Functional Requirement .....	5
1.2 Data Requirement.....	6
2 Chapter 2 - Conceptual Design .....	8
3 Chapter 3 – UML Diagram.....	9
4 Chapter 4 – Implementation .....	10
4.1 Database creation .....	10
4.2 Table Definitions.....	11
4.3 Insert data .....	13
4.4 Update data.....	17
4.5 Delete data.....	21
5 Chapter 5 – Transactions .....	24
5.1 Simple queries .....	24
5.2 Complex queries.....	26
6 Chapter 6 - Database Tuning .....	29

## List of Figure

Figure 1:ER diagram of garment database system.....	8
Figure 2:UML diagram of garmrnt database system .....	9
Figure 3:Schema creation.....	10
Figure 4:Create DepartmenT table, Employee table and child table .....	11
Figure 5:Create customer , cus_phone_number and customer_order tables.....	11
Figure 6:Create delivery , payment and product tables.....	12
Figure 7:Create order_item, supplier and material tables .....	12
Figure 8:Insert data to department, employee and child TABLES .....	13
Figure 9:Insert data to custormer,cus_phone_number and customer_order tables .....	13
Figure 10:Inser data into delivery, payment and product tables .....	14
Figure 11:Insert data into order_item, supplier and material tables.....	14
Figure 12:Data inserted table set_1 .....	15
Figure 13:DATA INSERTED TABLE SET_2.....	15
Figure 14:DATA INSERTED TABLE SET_3 .....	16
Figure 15:DATA INSERTED TABLE SET_4.....	16
Figure 16:Update two rows in department, employee and child tables .....	17
Figure 17:Update two rows in custormer, cus_phone_number and customer_order tables .....	17
Figure 18:update two rows in delivery ,payment and product tables .....	18
Figure 19:Update two rwoes in order_item , supplier and Material tables.....	18
Figure 20:updated table set-2 .....	19
Figure 21:Updated table set-1 .....	19
Figure 22:updated table set-4 .....	20
Figure 23:updated table set-3 .....	20
Figure 24:Delete one row from all tables.....	21
Figure 25:Deleted table set-2 .....	22
Figure 26:deleted table set-1 .....	22
Figure 27:deleted table set-4 .....	23
Figure 28:Deleted table set-3 .....	23
Figure 29:cartesian product operation and creating a user view .....	24
Figure 30: Select Operation and Project operation .....	24
Figure 31:Renaming operation , aggregation function and use of like keyword .....	25
Figure 32:Union operation and intersection operation.....	26
Figure 33: set difference operation and division operation.....	26
Figure 34:Inner join and natural join .....	27
Figure 35:Left outer join and right outer join .....	27
Figure 36:full outer join and outer union .....	28
Figure 37:nested queries .....	28
Figure 38:After TUNING UNION OPERATION .....	29
Figure 39:Before tuning Union operation .....	29
Figure 40:Before tuning intersection operation .....	30
Figure 41:AFTER TUNING INTERSECTION OPERATION .....	30
Figure 42:Before tuning set difference operation .....	31
Figure 43:After tuning set difference operation.....	31
Figure 44:Before tuning division operation .....	32
Figure 45:After tuning division operation.....	32
Figure 46:Before tuning inner join.....	33
Figure 47:After tuning inner join .....	33

Figure 48:Before tuning natural join.....	34
Figure 49:After joining natural join .....	34
Figure 50:Before tuning left outer join .....	35
Figure 51:After tuning left outer join.....	35
Figure 52:Before tuning right join .....	36
Figure 53:After tuning right join.....	36
Figure 54:Before tuning full outer join .....	37
Figure 55:After tuning full outer join.....	37

# **1 Chapter 1 - Requirement Analysis**

## **1.1 Functional Requirement**

This document is contained about the garment database system. First identified and derived required entities, attributes and relationships regarding to the garment database system. Then representing them using Entity Relationship diagram. Then ER diagram is converted to UML diagram of the database through normalization process. At the last, the database is physically implemented in the database management system by using MySQL workbench and MySQL Command Line client.

- In a garment factory, many products are made such as frocks, Trousers, T-shirts etc. To make this product they use many materials such as buttons, fabrics, zippers, laces, elastic etc.
- Materials are supplied by suppliers to the garment factory.
- There are many departments in a garment factory such as cutting department, design department, sewing department, pattern making department etc. Employees are worked in that departments.
- Employees have their own children and children are depended on the employees.
- Some Employees such as managers, supervisors worked as leaders to other employees. They manage other employees.
- Customers place orders and an order includes one or more items.
- An order consists of one or many products that made in the garment factory.
- An order is associated to an employee. That employee should responsible for that order.
- After order is ready it place delivery to the given places.
- Customers should make their payment for their order to get their order.

## **1.2 Data Requirement**

### **SUPPLIER**

- Name
- Email
- Contact Person
- Supplier ID
- Address
- Phone Number

### **MATERIALS**

- Material ID
- Colour
- Name
- Supplier ID
- Type
- Price per unit

### **PRODUCT**

- Product ID
- Size
- Category
- Price
- Name
- Description

### **ORDER**

- Order Date
- Status
- Delivery Date
- Customer ID
- Employee ID
- Order ID
- Total Amount

### **ORDER\_ITEM**

- Item ID
- Quantity
- Price

**DELIVERY**

- Delivery Date
- Delivery Address
- Order ID
- Delivery ID

**EMPLOYEE**

- Employee Name
- Employee ID
- Phone Number
- Email
- Date of Birth
- Age
- Salary
- Position

**DEPARTMENT**

- Department ID
- Name
- Department Head
- Description

**CHILD**

- Age
- Child Name
- Gender

**CUSTOMER**

- Customer ID
- Customer Name
- Address
- Email
- Phone number

**PAYMENT**

- Payment ID
- Payment Date
- Payment method
- Amount
- Customer ID

## 2 Chapter 2 - Conceptual Design

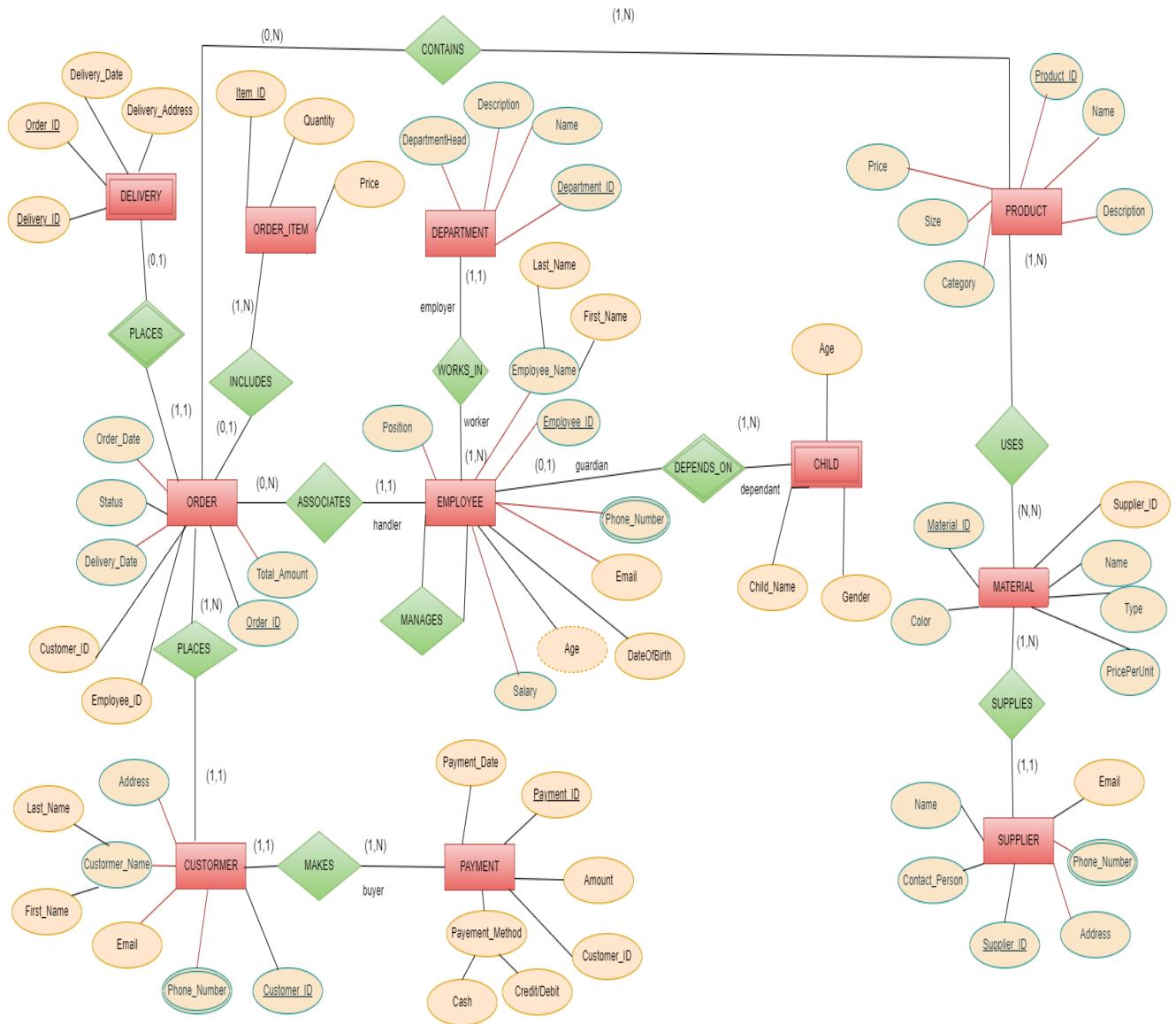
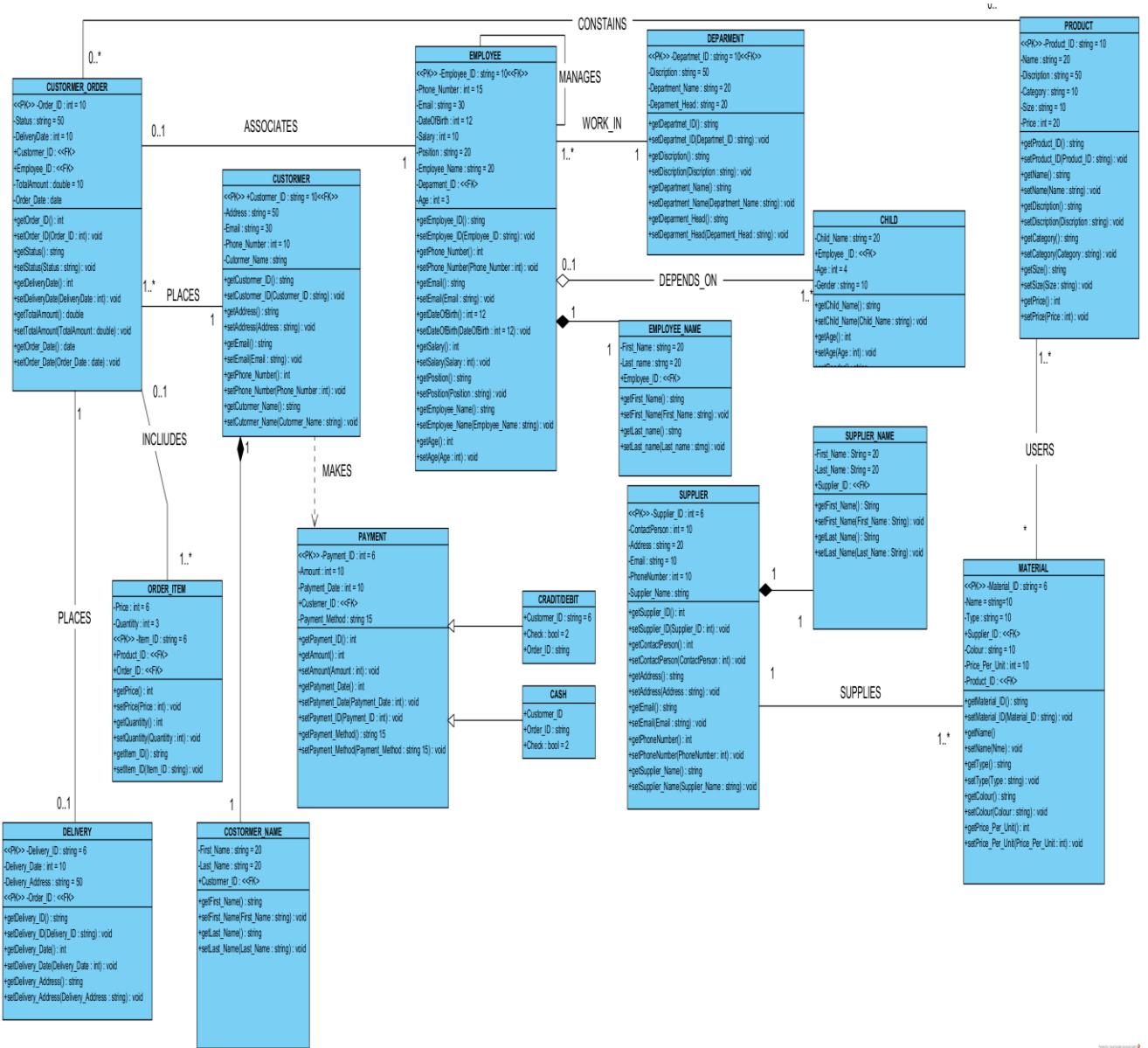


FIGURE 1:ER DIAGRAM OF GARMENT DATABASE SYSTEM

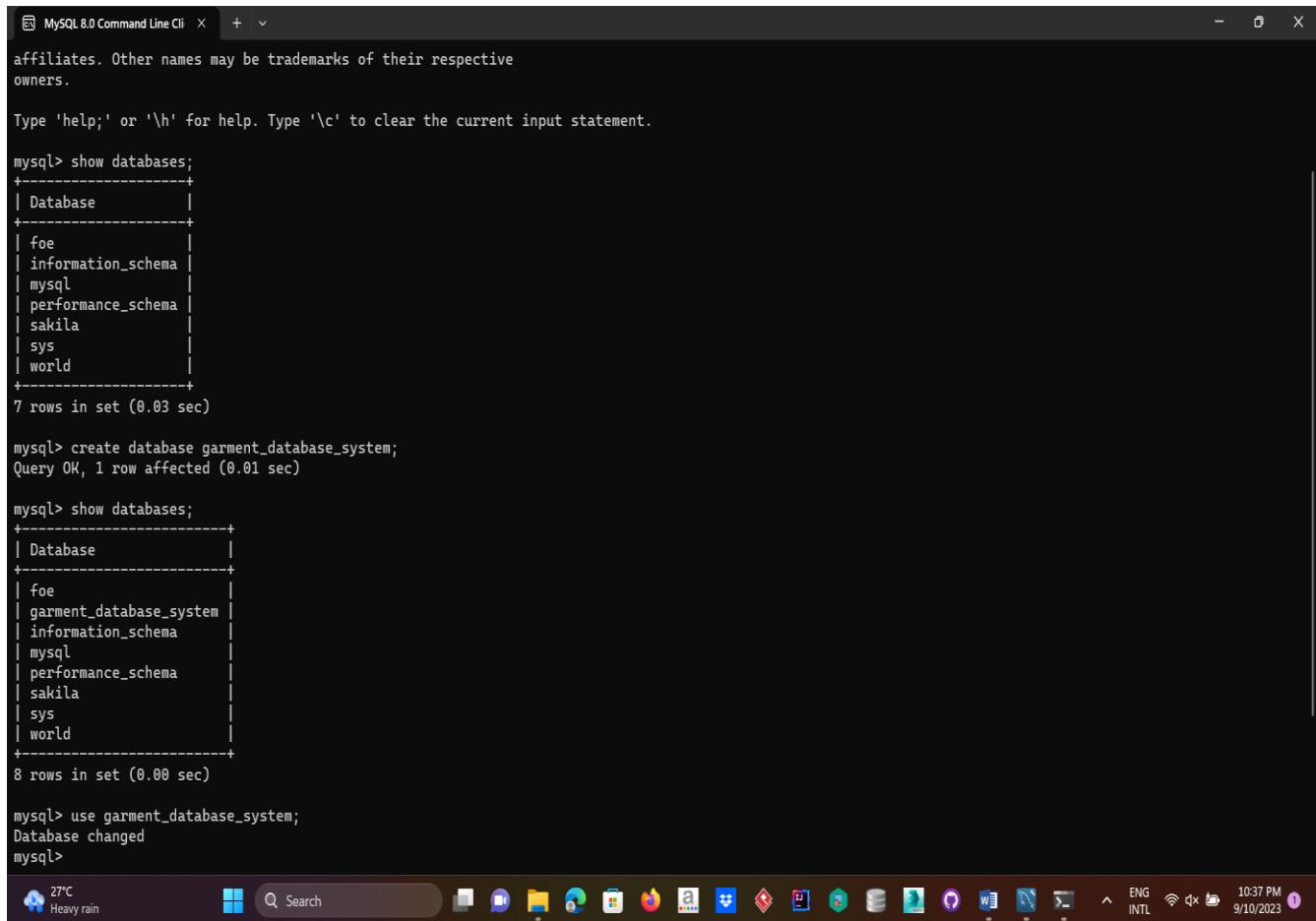
### 3 Chapter 3 – UML Diagram



**FIGURE 2:UML DIAGRAM OF GARMRNT DATABASE SYSTEM**

## 4 Chapter 4 – Implementation

### 4.1 Database creation



The screenshot shows a terminal window titled "MySQL 8.0 Command Line Cli". The session starts with a notice about trademarks, followed by a help message. The user runs "show databases;" which lists the default MySQL databases: "foe", "information\_schema", "mysql", "performance\_schema", "sakila", "sys", and "world". Then, the user creates a new database named "garment\_database\_system" with the command "create database garment\_database\_system;". After the creation, another "show databases;" command is run, showing the newly created database alongside the others. Finally, the user switches to the new database with "use garment\_database\_system;" and confirms the change with "Database changed". The system tray at the bottom shows weather information ("27°C Heavy rain"), a search bar, and various application icons.

```
MySQL 8.0 Command Line Cli + X
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| foe      |
| information_schema |
| mysql    |
| performance_schema |
| sakila   |
| sys      |
| world    |
+-----+
7 rows in set (0.03 sec)

mysql> create database garment_database_system;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| foe      |
| garment_database_system |
| information_schema |
| mysql    |
| performance_schema |
| sakila   |
| sys      |
| world    |
+-----+
8 rows in set (0.00 sec)

mysql> use garment_database_system;
Database changed
mysql>

27°C Heavy rain Q Search ENG INTL 10:37 PM 9/10/2023
```

FIGURE 3:SCHEMA CREATION

## 4.2 Table Definitions

```
mysql> use garment_database_system;
Database changed
mysql> create table DEPARTMENT(
    -> Department_ID varchar(10) not null ,
    -> Department_Name varchar(20),
    -> Description varchar (50),
    -> Department_Head varchar(20),
    -> primary key(Department_ID));
Query OK, 0 rows affected (0.01 sec)

mysql> create table EMPLOYEE(Employee_ID varchar (10) not null primary KEY ,
    -> Department_ID varchar(10),
    -> Email varchar(25),
    -> Born_Year int(4),
    -> Born_Month int(4),
    -> Born_Date int(4),
    -> Age int(3),
    -> First_Name varchar(20),
    -> Last_Name varchar(20),
    -> Salary int (10),
    -> Position varchar(20),
    -> foreign key (Department_ID) references DEPARTMENT(Department_ID)on delete set null on update cascade);
Query OK, 0 rows affected, 5 warnings (0.01 sec)

mysql> CREATE TABLE CHILD (
    -> Child_Name VARCHAR(20) NOT NULL,
    -> Employee_ID VARCHAR(10),
    -> Age INT(5),
    -> Gender VARCHAR(8),
    -> PRIMARY KEY (Child_Name,Employee_ID),
    -> FOREIGN KEY (Employee_ID) REFERENCES EMPLOYEE(Employee_ID) ON DELETE CASCADE ON UPDATE CASCADE);
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

FIGURE 4:CREATE DEPARTMENT TABLE, EMPLOYEE TABLE AND CHILD TABLE

```
mysql> CREATE TABLE CUSTOMER (
    -> First_Name VARCHAR(20),
    -> Last_Name VARCHAR(20),
    -> Customer_ID VARCHAR(10) not null PRIMARY KEY,
    -> Street_Name VARCHAR(20),
    -> City VARCHAR(20),
    -> State VARCHAR(20),
    -> PostalCode VARCHAR(20),
    -> Country VARCHAR(30),
    -> Email VARCHAR(30));
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE CUS_PHONE_NUMBER (
    -> Phone_Number CHAR(15) not null,
    -> Customer_ID VARCHAR(10),
    -> PRIMARY KEY (Phone_Number, Customer_ID),
    -> FOREIGN KEY (Customer_ID) REFERENCES CUSTOMER(Customer_ID) ON DELETE cascade ON UPDATE CASCADE);
Query OK, 0 rows affected (0.01 sec)

mysql> create table Custormer_Order(Order_ID varchar (10) not null ,
    -> Customer_ID varchar (10),
    -> Employee_ID varchar (10) ,
    -> Order_Date date ,
    -> Delivery_Date date ,
    -> Order_status varchar (30),
    -> Total_Amount int(10),
    -> foreign key (Employee_ID) REFERENCES EMPLOYEE(Employee_ID) on delete set null on update cascade ,
    -> foreign key (Customer_ID) REFERENCES CUSTOMER(Customer_ID) on delete set null on update cascade ,
    -> primary key ( Order_ID));
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> CREATE TABLE DELIVERY (
    -> Order_ID VARCHAR(10),
```

FIGURE 5:CREATE CUSTOMER , CUS\_PHONE\_NUMBER AND CUSTOMER\_ORDER TABLES

```
MySQL 8.0 Command Line Cli X + v

mysql> CREATE TABLE DELIVERY (
->     Order_ID VARCHAR(10),
->     Delivery_ID VARCHAR(10) NOT NULL,
->     Street_Name VARCHAR(20),
->     City VARCHAR(20),
->     State VARCHAR(20),
->     PostalCode VARCHAR(20),
->     Country VARCHAR(30),
->     Delivery_Date DATE,
->     PRIMARY KEY (Delivery_ID, Order_ID),
->     FOREIGN KEY (Order_ID) REFERENCES Customer_Order(Order_ID) ON DELETE cascade ON UPDATE CASCADE);
Query OK, 0 rows affected (0.01 sec)

mysql> create table PAYMENT(
->     Customer_ID varchar (10),
->     Payment_ID varchar (10) not null ,
->     Payment_Date date ,
->     cash boolean ,
->     Online_payment boolean,
->     foreign key (Customer_ID) REFERENCES CUSTOMER(Customer_ID) on delete set null on update cascade ,
->     primary key ( Payment_ID));
Query OK, 0 rows affected (0.01 sec)

mysql> create table PRODUCT(
->     Product_ID varchar (10) not null ,
->     Product_Name varchar (20),
->     Product_Description varchar(50),
->     Size varchar (10),
->     Category varchar (20),
->     Price int(20),
->     primary key (Product_ID));
Query OK, 0 rows affected, 1 warning (0.01 sec)

Cloudy 25°C Search ENG INTL 3:06 AM 9/11/2023
```

FIGURE 6:CREATE DELIVERY , PAYMENT AND PRODUCT TABLES

```
MySQL 8.0 Command Line Cli X + v

->     Size varchar (10),
->     Category varchar (20),
->     Price int(20),
->     primary key (Product_ID));
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> create table ORDER_ITEM(
->     Customer_ID varchar (10),
->     Item_ID varchar (10) not null ,
->     Product_ID varchar (10) ,
->     Quantity int(10) ,
->     Price int(20) ,
->     foreign key (Customer_ID) REFERENCES CUSTOMER(Customer_ID) on delete set null on update cascade ,
->     foreign key (Product_ID) REFERENCES PRODUCT(Product_ID) on delete set null on update cascade ,
->     primary key (Item_ID));
Query OK, 0 rows affected, 2 warnings (0.01 sec)

mysql> create table SUPPLIER(
->     First_Name varchar (20),
->     Last_Name varchar (20),
->     Supplier_ID varchar (10) not null ,
->     Contact_Person varchar (30),
->     Email varchar (30),
->     Phone_Number int(15),
->     primary key (Supplier_ID));
Query OK, 0 rows affected, 1 warning (0.06 sec)

mysql> create table MATERIAL(
->     Material_ID varchar (10) not null ,
->     Supplier_ID varchar (10),
->     Material_Name varchar (30) ,
->     Product_ID varchar (10),
->     Colour varchar (10) ,
->     Material_Type varchar (10) ,
->     Price_Per_Unit int(10),
->     foreign key (Product_ID) REFERENCES PRODUCT(Product_ID) on delete set null on update cascade ,
->     foreign key (Supplier_ID) REFERENCES SUPPLIER(Supplier_ID) on delete set null on update cascade ,
->     primary key (Material_ID));
Query OK, 0 rows affected, 1 warning (0.02 sec)

Cloudy 25°C Search ENG INTL 3:07 AM 9/11/2023
```

FIGURE 7:CREATE ORDER\_ITEM, SUPPLIER AND MATERIAL TABLES

## 4.3 Insert data

```
MySQL 8.0 Command Line Cli X + - X

mysql> INSERT INTO DEPARTMENT (Department_ID, Department_Name, Description, Department_Head)
-> VALUES
->     ('D001', 'HR', 'Human Resources', 'John Doe'),
->     ('D002', 'IT', 'Information Technology', 'Alice Smith'),
->     ('D003', 'Finance', 'Finance and Accounting', 'Bob Johnson'),
->     ('D004', 'Sales', 'Sales and Marketing', 'Eva Williams'),
->     ('D005', 'Production', 'Production Department', 'Mike Brown'),
->     ('D006', 'Research', 'Research and Development', 'Lisa Davis');
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> INSERT INTO EMPLOYEE (Employee_ID, Department_ID, Email, Born_Year, Born_Month, Born_Date, Age, First_Name, Last_Name, Salary, Position)
-> VALUES
->     ('E001', 'D001', 'john@example.com', 1980, 5, 15, 43, 'John', 'Doe', 50000, 'HR Manager'),
->     ('E002', 'D002', 'alice@example.com', 1990, 8, 20, 32, 'Alice', 'Smith', 60000, 'Software Engineer'),
->     ('E003', 'D002', 'bob@example.com', 1985, 3, 10, 37, 'Bob', 'Johnson', 55000, null),
->     ('E004', 'D004', 'eva@example.com', 1992, 11, 5, 29, 'Eva', 'Williams', 52000, 'Sales Executive'),
->     ('E005', 'D004', 'mike@example.com', 1988, 6, 25, 34, 'Mike', 'Brown', 48000, 'Sales Executive'),
->     ('E006', 'D006', 'lisa@example.com', 1987, 9, 30, 35, 'Lisa', 'Davis', 56000, 'Research Scientist');
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> INSERT INTO CHILD (Child_Name, Employee_ID, Age, Gender)
-> VALUES
->     ('Child1', 'E001', 5, 'Male'),
->     ('Child2', 'E002', 3, 'Female'),
->     ('Child3', 'E004', 6, 'Male'),
->     ('Child4', 'E006', 4, 'Female'),
->     ('Child5', 'E003', 7, 'Male'),
->     ('Child6', 'E005', 5, 'Female');
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> INSERT INTO CUSTOMER (First_Name, Last_Name, Customer_ID, Street_Name, City, State, PostalCode, Country, Email)

```

FIGURE 8:INSERT DATA TO DEPARTMENT, EMPLOYEE AND CHILD TABLES

```
MySQL 8.0 Command Line Cli X + - X

Records: 6 Duplicates: 0 Warnings: 0

mysql> INSERT INTO CUSTOMER (First_Name, Last_Name, Customer_ID, Street_Name, City, State, PostalCode, Country, Email)
-> VALUES
->     ('Saman', 'Perera', 'C001', '123 Main St', 'Colombo', 'Western', '10000', 'Sri Lanka', 'saman@example.com'),
->     ('Nimal', 'Fernando', 'C002', '456 Elm St', 'Kandy', 'Central', '20000', 'Sri Lanka', 'nimal@example.com'),
->     ('Kamal', 'Silva', 'C003', '789 Oak St', 'Galle', 'Southern', '30000', 'Sri Lanka', 'kamal@example.com'),
->     ('Priya', 'Rajapakse', 'C004', '101 Pine St', 'Jaffna', 'Northern', '40000', 'Sri Lanka', 'priya@example.com'),
->     ('Rohan', 'Bandara', 'C005', '202 Cedar St', 'Negombo', 'Western', '50000', 'Sri Lanka', 'rohan@example.com'),
->     ('Sujitha', 'Fernando', 'C006', '303 Birch St', 'Matara', 'Southern', '60000', 'Sri Lanka', 'sujitha@example.com');
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> INSERT INTO CUS_PHONE_NUMBER (Phone_Number, Customer_ID)
-> VALUES
->     (0772635432, 'C001'),
->     (0775643245, 'C002'),
->     (0778598786, 'C003'),
->     (0765667788, 'C004'),
->     (0756778899, 'C005'),
->     (0727889900, 'C006');
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> INSERT INTO Customer_Order (Order_ID, Customer_ID, Employee_ID, Order_Date, Delivery_Date, Order_Status, Total_Amount)
-> VALUES
->     ('O001', 'C001', 'E001', '2023-09-08', '2023-09-15', 'Pending', 5000),
->     ('O002', 'C002', 'E002', '2023-09-08', '2023-09-16', 'Shipped', 7500),
->     ('O003', 'C003', 'E003', '2023-09-09', '2023-09-17', 'Delivered', 6000),
->     ('O004', 'C004', 'E004', '2023-09-10', '2023-09-18', 'Pending', 5500),
->     ('O005', 'C004', 'E005', '2023-09-11', '2023-09-19', 'Shipped', 8000),
->     ('O006', 'C006', 'E006', '2023-09-12', '2023-09-20', 'Delivered', 7000);
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

FIGURE 9:INSERT DATA TO CUSTOMER.CUS\_PHONE\_NUMBER AND CUSTOMER\_ORDER TABLES

```

MySQL 8.0 Command Line Cli + 
Records: 6 Duplicates: 0 Warnings: 0

mysql> INSERT INTO DELIVERY (Order_ID, Delivery_ID, Street_Name, City, State, PostalCode, Country, Delivery_Date)
-> VALUES
->     ('D001', 'D001', '123 Main St', 'Colombo', 'Western', '10000', 'Sri Lanka', '2023-09-15'),
->     ('D002', 'D002', '456 Elm St', 'Kandy', 'Central', '20000', 'Sri Lanka', '2023-09-16'),
->     ('D003', 'D003', '789 Oak St', 'Galle', 'Southern', '30000', 'Sri Lanka', '2023-09-17'),
->     ('D004', 'D004', '101 Pine St', 'Jaffna', 'Northern', '40000', 'Sri Lanka', '2023-09-18'),
->     ('D005', 'D005', '202 Cedar St', 'Negombo', 'Western', '50000', 'Sri Lanka', '2023-09-19'),
->     ('D006', 'D006', '303 Birch St', 'Matara', 'Southern', '60000', 'Sri Lanka', '2023-09-20');
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> INSERT INTO PAYMENT (Customer_ID, Payment_ID, Payment_Date, cash, Online_payment)
-> VALUES
->     ('C001', 'P001', '2023-09-10', true, false),
->     ('C002', 'P002', '2023-09-11', false, true),
->     ('C003', 'P003', '2023-09-12', true, false),
->     ('C004', 'P004', '2023-09-13', false, true),
->     ('C005', 'P005', '2023-09-14', true, false),
->     ('C006', 'P006', '2023-09-15', false, true);
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> INSERT INTO PRODUCT (Product_ID, Product_Name, Product_Description, Size, Category, Price)
-> VALUES
->     ('PRD001', 'Shirt', 'Cotton shirt for men', 'M', 'Clothing', 2000),
->     ('PRD002', 'Dress', 'Summer dress for women', 'S', 'Clothing', 2500),
->     ('PRD003', 'Shoes', 'Leather shoes for men', '10', 'Footwear', 3000),
->     ('PRD004', 'Handbag', 'Leather handbag for women', 'Medium', 'Accessories', 1500),
->     ('PRD005', 'Hat', 'Sun hat for all', 'One Size', 'Accessories', 800),
->     ('PRD006', 'Jeans', 'Denim jeans for men', '32', 'Clothing', 1800);
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

```

FIGURE 10:INSER DATA INTO DELIVERY, PAYMENT AND PRODUCT TABLES

```

MySQL 8.0 Command Line Cli + 
Records: 6 Duplicates: 0 Warnings: 0

mysql> INSERT INTO ORDER_ITEM (Customer_ID, Item_ID, Product_ID, Quantity, Price)
-> VALUES
->     ('C001', 'I001', 'PRD001', 2, 4000),
->     ('C002', 'I002', 'PRD001', 1, 2500),
->     ('C003', 'I003', 'PRD003', 1, 3000),
->     ('C004', 'I004', 'PRD003', 3, 4500),
->     ('C005', 'I005', 'PRD005', 2, 1600),
->     ('C006', 'I006', 'PRD006', 2, 3600);
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> INSERT INTO SUPPLIER (First_Name, Last_Name, Suppllier_ID, Contact_Person, Email, Phone_Number)
-> VALUES
->     ('ABC Textiles', 'Private Ltd', 'S001', 'John Supplier', 'johnsupplier@example.com', 772345678),
->     ('XYZ Footwear', 'Suppliers', 'S002', 'Alice Supplier', 'alicesupplier@example.com', 773456789),
->     ('Leather World', 'Exports', 'S003', 'Bob Supplier', 'bobsupplier@example.com', 774567890),
->     ('FashionHub', 'Supplies', 'S004', 'Eva Supplier', 'evasupplier@example.com', 775678901),
->     ('Accessories Plus', 'Imports', 'S005', 'Mike Supplier', 'mikesupplier@example.com', 776789012),
->     ('Jeans Inc', 'International', 'S006', 'Lisa Supplier', 'lisasupplier@example.com', 777890123);
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> INSERT INTO MATERIAL (Material_ID, Suppllier_ID, Material_Name, Product_ID, Colour, Material_Type, Price_Per_Unit)
-> VALUES
->     ('M001', 'S001', 'Cotton Fabric', 'PRD001', 'White', 'Fabric', 500),
->     ('M002', 'S002', 'Leather', 'PRD003', 'Black', 'Leather', 1000),
->     ('M003', 'S003', 'Leather', 'PRD004', 'Brown', 'Leather', 800),
->     ('M004', 'S004', 'Straw', 'PRD005', 'Beige', 'Straw', 300),
->     ('M005', 'S005', 'Denim Fabric', 'PRD006', 'Blue', 'Fabric', 600),
->     ('M006', 'S006', 'Cotton Fabric', 'PRD002', 'Red', 'Fabric', 550);
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

```

FIGURE 11:INSERT DATA INTO ORDER ITEM, SUPPLIER AND MATERIAL TABLES

```

mysql> select *from department;
+-----+-----+-----+-----+
| Department_ID | Department_Name | Description | Department_Head |
+-----+-----+-----+-----+
| D001 | HR | Human Resources | John Doe |
| D002 | IT | Information Technology | Alice Smith |
| D003 | Finance | Finance and Accounting | Bob Johnson |
| D004 | Sales | Sales and Marketing | Eva Williams |
| D005 | Production | Production Department | Mike Brown |
| D006 | Research | Research and Development | Lisa Davis |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select *from employee;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Employee_ID | Department_ID | Email | Born_Year | Born_Month | Born_Date | Age | First_Name | Last_Name | Salary | Position |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| E001 | D001 | john@example.com | 1980 | 5 | 15 | 43 | John | Doe | 50000 | HR Manager |
| E002 | D002 | alice@example.com | 1990 | 8 | 20 | 32 | Alice | Smith | 60000 | Software Engineer |
| E003 | D002 | bob@example.com | 1985 | 3 | 10 | 37 | Bob | Johnson | 55000 | NULL |
| E004 | D004 | eva@example.com | 1992 | 11 | 5 | 29 | Eva | Williams | 52000 | Sales Executive |
| E005 | D004 | mike@example.com | 1988 | 6 | 25 | 34 | Mike | Brown | 48000 | Sales Executive |
| E006 | D006 | lisa@example.com | 1987 | 9 | 30 | 35 | Lisa | Davis | 56000 | Research Scientist |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select *from child;
+-----+-----+-----+-----+
| Child_Name | Employee_ID | Age | Gender |
+-----+-----+-----+-----+
| Child1 | E001 | 5 | Male |
| Child2 | E002 | 3 | Female |
| Child3 | E004 | 6 | Male |
| Child4 | E006 | 4 | Female |
| Child5 | E003 | 7 | Male |
| Child6 | E005 | 5 | Female |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

Windows taskbar at the bottom:

- Cloudy (25°C)
- Search bar
- Icons for File, Copy, Paste, Find, etc.
- System status: ENG INTL, battery level, date (9/11/2023), time (3:11 AM)

FIGURE 12:DATA INSERTED TABLE SET \_1

```

mysql> select *from customer;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| First_Name | Last_Name | Customer_ID | Street_Name | City | State | PostalCode | Country | Email |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Saman | Perera | C001 | 123 Main St | Colombo | Western | 10000 | Sri Lanka | saman@example.com |
| Nimal | Fernando | C002 | 456 Elm St | Kandy | Central | 20000 | Sri Lanka | nimal@example.com |
| Kamal | Silva | C003 | 789 Oak St | Galle | Southern | 30000 | Sri Lanka | kamal@example.com |
| Priya | Rajapakse | C004 | 101 Pine St | Jaffna | Northern | 40000 | Sri Lanka | priya@example.com |
| Rohan | Bandara | C005 | 202 Cedar St | Negombo | Western | 50000 | Sri Lanka | rohan@example.com |
| Sujitha | Fernando | C006 | 303 Birch St | Matara | Southern | 60000 | Sri Lanka | sujitha@example.com |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select *from cus_phone_number;
+-----+-----+
| Phone_Number | Customer_ID |
+-----+-----+
| 772635432 | C001 |
| 775643245 | C002 |
| 778598786 | C003 |
| 765667788 | C004 |
| 756778899 | C005 |
| 727889900 | C006 |
+-----+-----+
6 rows in set (0.00 sec)

mysql> select *from customer_order;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Order_ID | Customer_ID | Employee_ID | Order_Date | Delivery_Date | Order_Status | Total_Amount |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 0001 | C001 | E001 | 2023-09-08 | 2023-09-15 | Pending | 5000 |
| 0002 | C002 | E002 | 2023-09-08 | 2023-09-16 | Shipped | 7500 |
| 0003 | C003 | E003 | 2023-09-09 | 2023-09-17 | Delivered | 6000 |
| 0004 | C004 | E004 | 2023-09-10 | 2023-09-18 | Pending | 5500 |
| 0005 | C004 | E005 | 2023-09-11 | 2023-09-19 | Shipped | 8000 |
| 0006 | C006 | E006 | 2023-09-12 | 2023-09-20 | Delivered | 7000 |
+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

Windows taskbar at the bottom:

- Cloudy (25°C)
- Search bar
- Icons for File, Copy, Paste, Find, etc.
- System status: ENG INTL, battery level, date (9/11/2023), time (3:12 AM)

FIGURE 13:DATA INSERTED TABLE SET \_2

```

MySQL 8.0 Command Line Cli × + v

mysql> select *from delivery;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Order_ID | Delivery_ID | Street_Name | City | State | PostalCode | Country | Delivery_Date |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 0001 | D001 | 123 Main St | Colombo | Western | 10000 | Sri Lanka | 2023-09-15 |
| 0002 | D002 | 456 Elm St | Kandy | Central | 20000 | Sri Lanka | 2023-09-16 |
| 0003 | D003 | 789 Oak St | Galle | Southern | 30000 | Sri Lanka | 2023-09-17 |
| 0004 | D004 | 101 Pine St | Jaffna | Northern | 40000 | Sri Lanka | 2023-09-18 |
| 0005 | D005 | 202 Cedar St | Negombo | Western | 50000 | Sri Lanka | 2023-09-19 |
| 0006 | D006 | 303 Birch St | Matara | Southern | 60000 | Sri Lanka | 2023-09-20 |
+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select *from payment;
+-----+-----+-----+-----+-----+
| Customer_ID | Payment_ID | Payment_Date | cash | Online_payment |
+-----+-----+-----+-----+-----+
| C001 | P001 | 2023-09-10 | 1 | 0 |
| C002 | P002 | 2023-09-11 | 0 | 1 |
| C003 | P003 | 2023-09-12 | 1 | 0 |
| C004 | P004 | 2023-09-13 | 0 | 1 |
| C005 | P005 | 2023-09-14 | 1 | 0 |
| C006 | P006 | 2023-09-15 | 0 | 1 |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select *from product;
+-----+-----+-----+-----+-----+-----+
| Product_ID | Product_Name | Product_Description | Size | Category | Price |
+-----+-----+-----+-----+-----+-----+
| PRD001 | Shirt | Cotton shirt for men | M | Clothing | 2000 |
| PRD002 | Dress | Summer dress for women | S | Clothing | 2500 |
| PRD003 | Shoes | Leather shoes for men | 10 | Footwear | 3000 |
| PRD004 | Handbag | Leather handbag for women | Medium | Accessories | 1500 |
| PRD005 | Hat | Sun hat for all | One Size | Accessories | 800 |
| PRD006 | Jeans | Denim jeans for men | 32 | Clothing | 1800 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

FIGURE 14:DATA INSERTED TABLE SET \_3

```

MySQL 8.0 Command Line Cli × + v

mysql> select *from order_item;
+-----+-----+-----+-----+-----+
| Customer_ID | Item_ID | Product_ID | Quantity | Price |
+-----+-----+-----+-----+-----+
| C001 | I001 | PRD001 | 2 | 4000 |
| C002 | I002 | PRD001 | 1 | 2500 |
| C003 | I003 | PRD003 | 1 | 3000 |
| C004 | I004 | PRD003 | 3 | 4500 |
| C005 | I005 | PRD005 | 2 | 1600 |
| C006 | I006 | PRD006 | 2 | 3600 |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

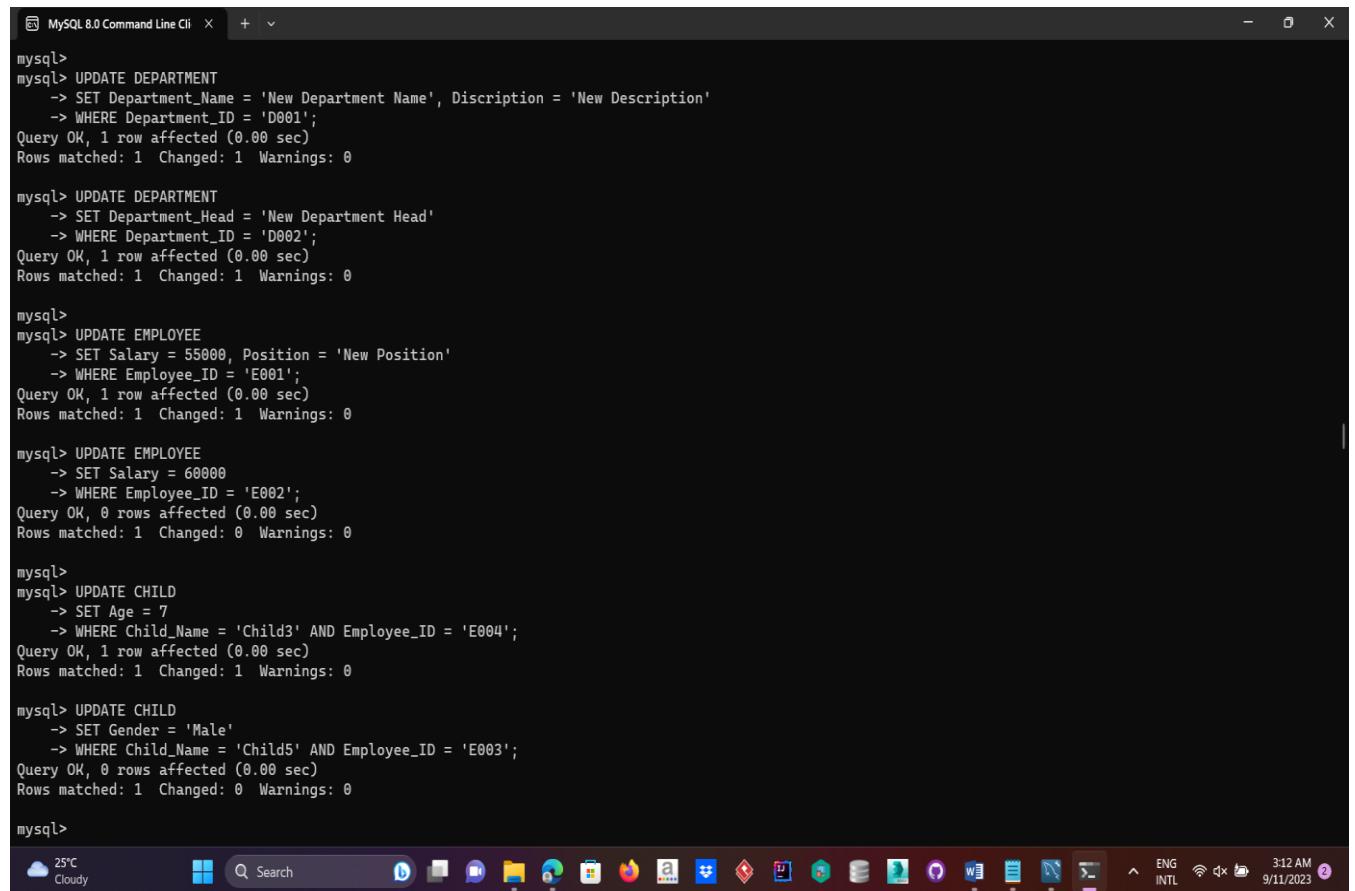
mysql> select *from supplier;
+-----+-----+-----+-----+-----+-----+
| First_Name | Last_Name | Suppllier_ID | Contact_Person | Email | Phone_Number |
+-----+-----+-----+-----+-----+-----+
| ABC Textiles | Private Ltd | S001 | John Supplier | johnsupplier@example.com | 772345678 |
| XYZ Footwear | Suppliers | S002 | Alice Supplier | alicesupplier@example.com | 773456789 |
| Leather World | Exports | S003 | Bob Supplier | bobsupplier@example.com | 774567890 |
| FashionHub | Supplies | S004 | Eva Supplier | evasupplier@example.com | 775678901 |
| Accessories Plus | Imports | S005 | Mike Supplier | mikessupplier@example.com | 776789012 |
| Jeans Inc | International | S006 | Lisa Supplier | lisasupplier@example.com | 777890123 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select *from material;
+-----+-----+-----+-----+-----+-----+
| Material_ID | Suppllier_ID | Material_Name | Product_ID | Colour | Material_Type | Price_Per_Unit |
+-----+-----+-----+-----+-----+-----+-----+
| M001 | S001 | Cotton Fabric | PRD001 | White | Fabric | 500 |
| M002 | S002 | Leather | PRD003 | Black | Leather | 1000 |
| M003 | S003 | Leather | PRD004 | Brown | Leather | 800 |
| M004 | S004 | Straw | PRD005 | Beige | Straw | 300 |
| M005 | S005 | Denim Fabric | PRD006 | Blue | Fabric | 600 |
| M006 | S006 | Cotton Fabric | PRD002 | Red | Fabric | 550 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

FIGURE 15:DATA INSERTED TABLE SET \_4

## 4.4 Update data



The screenshot shows the MySQL 8.0 Command Line Client interface. The command line window displays several UPDATE queries being executed against different tables. The queries are as follows:

```
mysql> UPDATE DEPARTMENT
-> SET Department_Name = 'New Department Name', Description = 'New Description'
-> WHERE Department_ID = 'D001';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE DEPARTMENT
-> SET Department_Head = 'New Department Head'
-> WHERE Department_ID = 'D002';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql>
mysql> UPDATE EMPLOYEE
-> SET Salary = 55000, Position = 'New Position'
-> WHERE Employee_ID = 'E001';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE EMPLOYEE
-> SET Salary = 60000
-> WHERE Employee_ID = 'E002';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1 Changed: 0 Warnings: 0

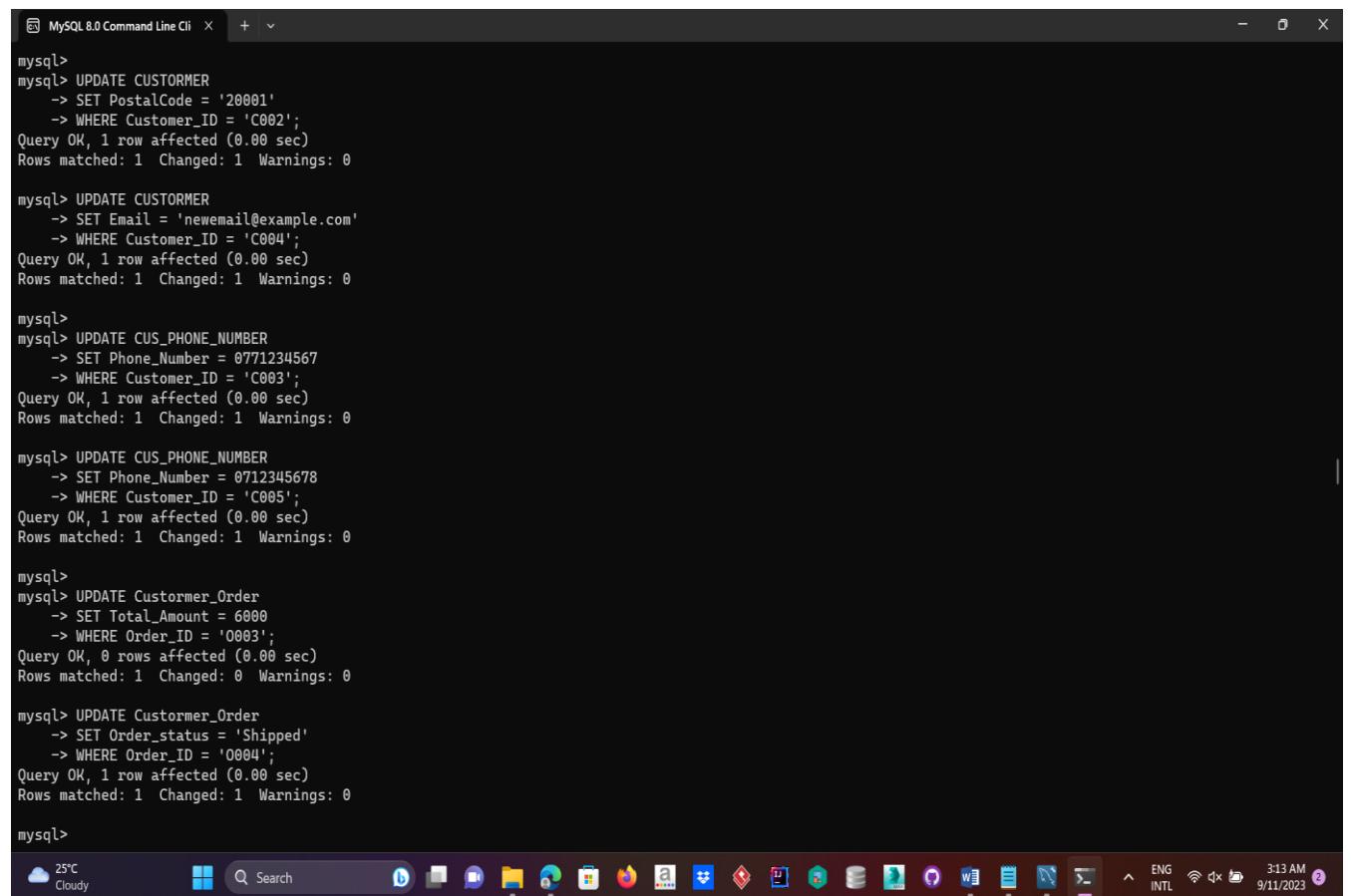
mysql>
mysql> UPDATE CHILD
-> SET Age = 7
-> WHERE Child_Name = 'Child3' AND Employee_ID = 'E004';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE CHILD
-> SET Gender = 'Male'
-> WHERE Child_Name = 'Child5' AND Employee_ID = 'E003';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1 Changed: 0 Warnings: 0

mysql>
```

The taskbar at the bottom of the screen shows various application icons, and the system tray indicates it's 9:11 AM on 11/9/2023.

FIGURE 16:UPDATE TWO ROWS IN DEPARTMENT, EMPLOYEE AND CHILD TABLES



The screenshot shows the MySQL 8.0 Command Line Client interface. The command line window displays several UPDATE queries being executed against three different tables. The queries are as follows:

```
mysql>
mysql> UPDATE CUSTOMER
-> SET PostalCode = '20001'
-> WHERE Customer_ID = 'C002';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE CUSTOMER
-> SET Email = 'newemail@example.com'
-> WHERE Customer_ID = 'C004';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql>
mysql> UPDATE CUS_PHONE_NUMBER
-> SET Phone_Number = 0771234567
-> WHERE Customer_ID = 'C003';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql>
mysql> UPDATE CUS_PHONE_NUMBER
-> SET Phone_Number = 0712345678
-> WHERE Customer_ID = 'C005';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

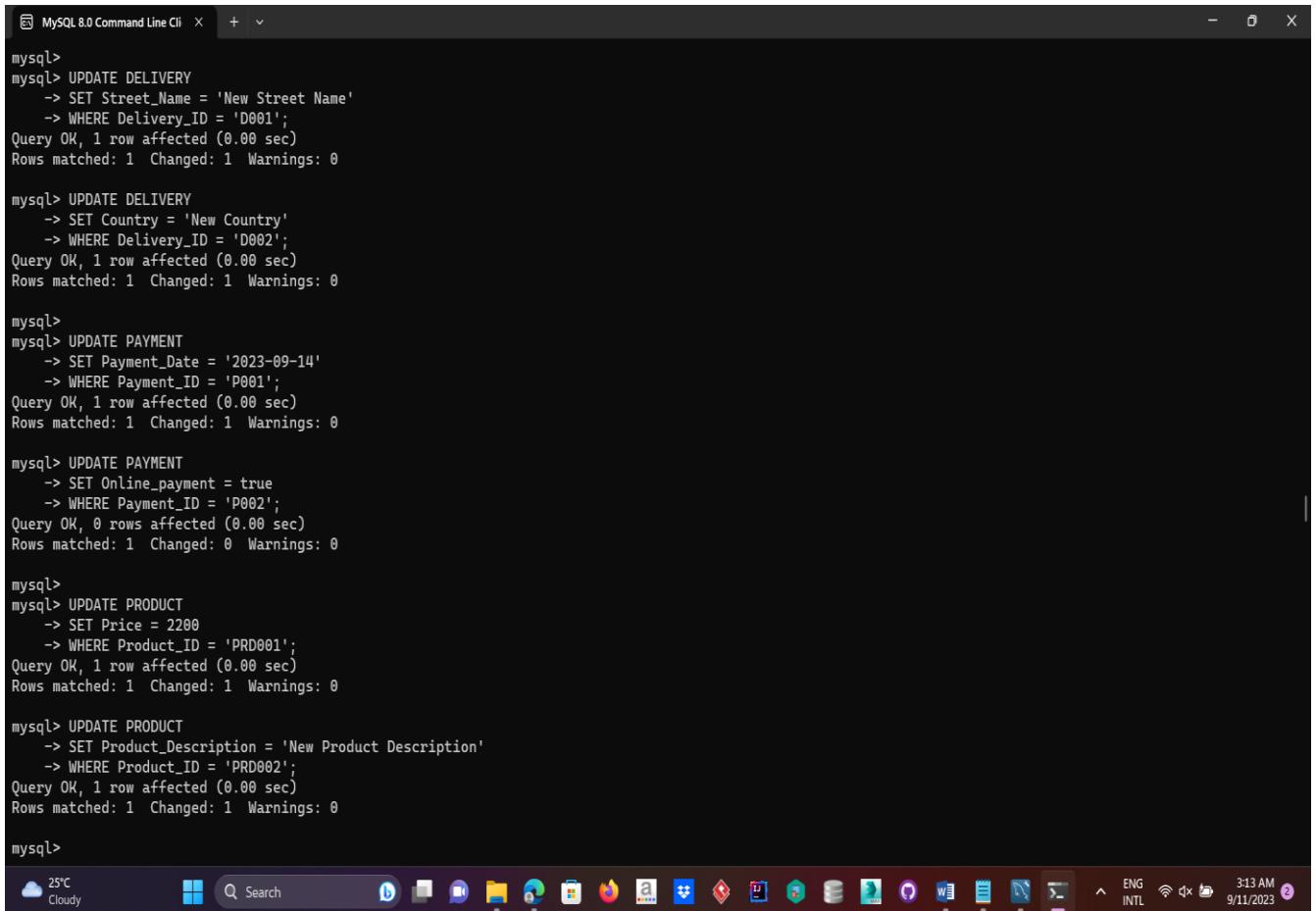
mysql>
mysql> UPDATE Customer_Order
-> SET Total_Amount = 6000
-> WHERE Order_ID = 'O003';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1 Changed: 0 Warnings: 0

mysql> UPDATE Customer_Order
-> SET Order_Status = 'Shipped'
-> WHERE Order_ID = 'O004';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql>
```

The taskbar at the bottom of the screen shows various application icons, and the system tray indicates it's 9:11 AM on 11/9/2023.

FIGURE 17:UPDATE TWO ROWS IN CUSTOMER, CUS\_PHONE\_NUMBER AND CUSTOMER\_ORDER TABLES



```
mysql> UPDATE DELIVERY
    -> SET Street_Name = 'New Street Name'
    -> WHERE Delivery_ID = 'D001';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE DELIVERY
    -> SET Country = 'New Country'
    -> WHERE Delivery_ID = 'D002';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> UPDATE PAYMENT
    -> SET Payment_Date = '2023-09-14'
    -> WHERE Payment_ID = 'P001';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

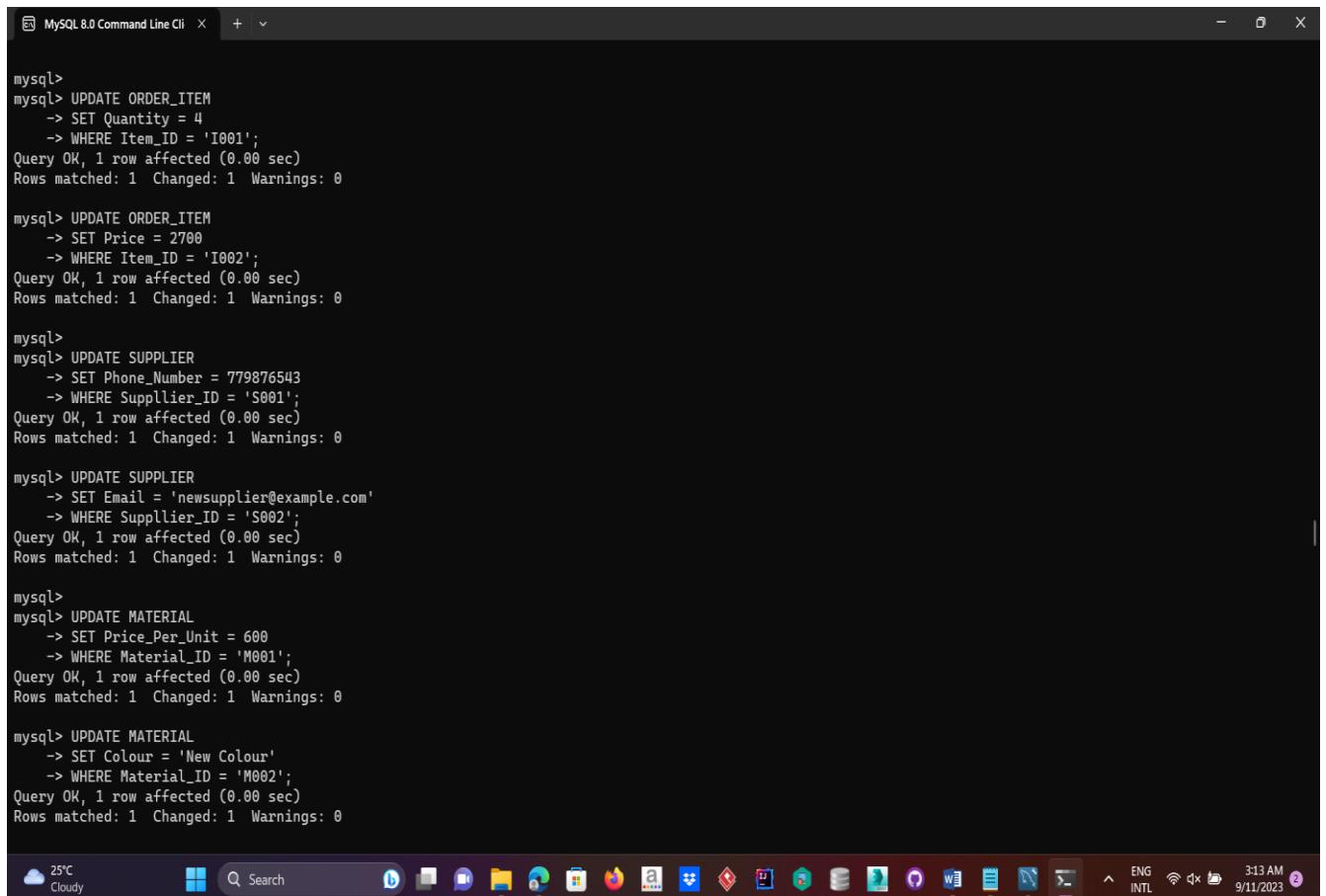
mysql> UPDATE PAYMENT
    -> SET Online_payment = true
    -> WHERE Payment_ID = 'P002';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0

mysql>
mysql> UPDATE PRODUCT
    -> SET Price = 2200
    -> WHERE Product_ID = 'PRD001';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE PRODUCT
    -> SET Product_Description = 'New Product Description'
    -> WHERE Product_ID = 'PRD002';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
```

FIGURE 18:UPDATE TWO ROWS IN DELIVERY ,PAYMENT AND PRODUCT TABLES



```
mysql>
mysql> UPDATE ORDER_ITEM
    -> SET Quantity = 4
    -> WHERE Item_ID = 'I001';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE ORDER_ITEM
    -> SET Price = 2700
    -> WHERE Item_ID = 'I002';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> UPDATE SUPPLIER
    -> SET Phone_Number = 779876543
    -> WHERE Supplier_ID = 'S001';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE SUPPLIER
    -> SET Email = 'newsupplier@example.com'
    -> WHERE Supplier_ID = 'S002';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> UPDATE MATERIAL
    -> SET Price_Per_Unit = 600
    -> WHERE Material_ID = 'M001';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE MATERIAL
    -> SET Colour = 'New Colour'
    -> WHERE Material_ID = 'M002';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

FIGURE 19:UPDATE TWO RWOS IN ORDER ITEM , SUPPLIER AND MATERIAL TABLES

```

MySQL 8.0 Command Line Cli + X

mysql> select *from department;
+-----+-----+-----+-----+
| Department_ID | Department_Name | Description | Department_Head |
+-----+-----+-----+-----+
| D001 | New Department Name | New Description | John Doe
| D002 | IT | Information Technology | New Department Head
| D003 | Finance | Finance and Accounting | Bob Johnson
| D004 | Sales | Sales and Marketing | Eva Williams
| D005 | Production | Production Department | Mike Brown
| D006 | Research | Research and Development | Lisa Davis
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select *from employee;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Employee_ID | Department_ID | Email | Born_Year | Born_Month | Born_Date | Age | First_Name | Last_Name | Salary | Position |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| E001 | D001 | john@example.com | 1980 | 5 | 15 | 43 | John | Doe | 55000 | New Position
| E002 | D002 | alice@example.com | 1990 | 8 | 20 | 32 | Alice | Smith | 60000 | Software Engineer
| E003 | D002 | bob@example.com | 1985 | 3 | 10 | 37 | Bob | Johnson | 55000 | NULL
| E004 | D004 | eva@example.com | 1992 | 11 | 5 | 29 | Eva | Williams | 52000 | Sales Executive
| E005 | D004 | mike@example.com | 1988 | 6 | 25 | 34 | Mike | Brown | 48000 | Sales Executive
| E006 | D006 | lisa@example.com | 1987 | 9 | 30 | 35 | Lisa | Davis | 56000 | Research Scientist
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select *from child;
+-----+-----+-----+-----+
| Child_Name | Employee_ID | Age | Gender |
+-----+-----+-----+-----+
| Child1 | E001 | 5 | Male |
| Child2 | E002 | 3 | Female |
| Child3 | E004 | 7 | Male |
| Child4 | E006 | 4 | Female |
| Child5 | E003 | 7 | Male |
| Child6 | E005 | 5 | Female |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

Cloudy 25°C Search ENG INTL 3:14 AM 9/11/2023

FIGURE 21:UPDATED TABLE SET-1

```

MySQL 8.0 Command Line Cli + X

mysql> select *from customer;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| First_Name | Last_Name | Customer_ID | Street_Name | City | State | PostalCode | Country | Email |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Saman | Perera | C001 | 123 Main St | Colombo | Western | 10000 | Sri Lanka | saman@example.com |
| Nimal | Fernando | C002 | 456 Elm St | Kandy | Central | 20001 | Sri Lanka | nimal@example.com |
| Kamal | Silva | C003 | 789 Oak St | Galle | Southern | 30000 | Sri Lanka | kamal@example.com |
| Priya | Rajapakse | C004 | 101 Pine St | Jaffna | Northern | 40000 | Sri Lanka | newemail@example.com |
| Rohan | Bandara | C005 | 202 Cedar St | Negombo | Western | 50000 | Sri Lanka | rohan@example.com |
| Sujitha | Fernando | C006 | 303 Birch St | Matara | Southern | 60000 | Sri Lanka | sujitha@example.com |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select *from cus_phone_number;
+-----+-----+
| Phone_Number | Customer_ID |
+-----+-----+
| 772635432 | C001 |
| 775643245 | C002 |
| 771234567 | C003 |
| 765667788 | C004 |
| 712345678 | C005 |
| 727889900 | C006 |
+-----+-----+
6 rows in set (0.00 sec)

mysql> select *from customer_order;
+-----+-----+-----+-----+-----+-----+-----+
| Order_ID | Customer_ID | Employee_ID | Order_Date | Delivery_Date | Order_Status | Total_Amount |
+-----+-----+-----+-----+-----+-----+-----+
| 0001 | C001 | E001 | 2023-09-08 | 2023-09-15 | Pending | 5000 |
| 0002 | C002 | E002 | 2023-09-08 | 2023-09-16 | Shipped | 7500 |
| 0003 | C003 | E003 | 2023-09-09 | 2023-09-17 | Delivered | 6000 |
| 0004 | C004 | E004 | 2023-09-10 | 2023-09-18 | Shipped | 5500 |
| 0005 | C004 | E005 | 2023-09-11 | 2023-09-19 | Shipped | 8000 |
| 0006 | C006 | E006 | 2023-09-12 | 2023-09-20 | Delivered | 7000 |
+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

Cloudy 25°C Search ENG INTL 3:14 AM 9/11/2023

FIGURE 20:UPDATED TABLE SET-2

```

MySQL 8.0 Command Line Cli  + 
mysql> select *from delivery;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Order_ID | Delivery_ID | Street_Name | City | State | PostalCode | Country | Delivery_Date |
+-----+-----+-----+-----+-----+-----+-----+-----+
| D001 | D001 | New Street Name | Colombo | Western | 10000 | Sri Lanka | 2023-09-15 |
| D002 | D002 | 456 Elm St | Kandy | Central | 20000 | New Country | 2023-09-16 |
| D003 | D003 | 789 Oak St | Galle | Southern | 30000 | Sri Lanka | 2023-09-17 |
| D004 | D004 | 101 Pine St | Jaffna | Northern | 40000 | Sri Lanka | 2023-09-18 |
| D005 | D005 | 202 Cedar St | Negombo | Western | 50000 | Sri Lanka | 2023-09-19 |
| D006 | D006 | 303 Birch St | Matara | Southern | 60000 | Sri Lanka | 2023-09-20 |
+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select *from payment;
+-----+-----+-----+-----+-----+
| Customer_ID | Payment_ID | Payment_Date | cash | Online_payment |
+-----+-----+-----+-----+-----+
| C001 | P001 | 2023-09-14 | 1 | 0 |
| C002 | P002 | 2023-09-11 | 0 | 1 |
| C003 | P003 | 2023-09-12 | 1 | 0 |
| C004 | P004 | 2023-09-13 | 0 | 1 |
| C005 | P005 | 2023-09-14 | 1 | 0 |
| C006 | P006 | 2023-09-15 | 0 | 1 |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select *from product;
+-----+-----+-----+-----+-----+-----+
| Product_ID | Product_Name | Product_Description | Size | Category | Price |
+-----+-----+-----+-----+-----+-----+
| PRD001 | Shirt | Cotton shirt for men | M | Clothing | 2200 |
| PRD002 | Dress | New Product Description | S | Clothing | 2500 |
| PRD003 | Shoes | Leather shoes for men | 10 | Footwear | 3000 |
| PRD004 | Handbag | Leather handbag for women | Medium | Accessories | 1500 |
| PRD005 | Hat | Sun hat for all | One Size | Accessories | 800 |
| PRD006 | Jeans | Denim jeans for men | 32 | Clothing | 1800 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

FIGURE 23:UPDATED TABLE SET-3

```

MySQL 8.0 Command Line Cli  + 
mysql> select *from order_item;
+-----+-----+-----+-----+
| Customer_ID | Item_ID | Product_ID | Quantity | Price |
+-----+-----+-----+-----+
| C001 | I001 | PRD001 | 4 | 4000 |
| C002 | I002 | PRD001 | 1 | 2700 |
| C003 | I003 | PRD003 | 1 | 3000 |
| C004 | I004 | PRD003 | 3 | 4500 |
| C005 | I005 | PRD005 | 2 | 1600 |
| C006 | I006 | PRD006 | 2 | 3600 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select *from supplier;
+-----+-----+-----+-----+-----+-----+
| First_Name | Last_Name | Suppllier_ID | Contact_Person | Email | Phone_Number |
+-----+-----+-----+-----+-----+-----+
| ABC Textiles | Private Ltd | S001 | John Supplier | johnsupplier@example.com | 779876543 |
| XYZ Footwear | Suppliers | S002 | Alice Supplier | newsupplier@example.com | 773456789 |
| Leather World | Exports | S003 | Bob Supplier | bobsupplier@example.com | 774567890 |
| FashionHub | Supplies | S004 | Eva Supplier | evasupplier@example.com | 775678901 |
| Accessories Plus | Imports | S005 | Mike Supplier | mikesupplier@example.com | 776789012 |
| Jeans Inc | International | S006 | Lisa Supplier | lisasupplier@example.com | 777890123 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select *from material;
+-----+-----+-----+-----+-----+-----+
| Material_ID | Suppllier_ID | Material_Name | Product_ID | Colour | Material_Type | Price_Per_Unit |
+-----+-----+-----+-----+-----+-----+-----+
| M001 | S001 | Cotton Fabric | PRD001 | White | Fabric | 600 |
| M002 | S002 | Leather | PRD003 | New Colour | Leather | 1000 |
| M003 | S003 | Leather | PRD004 | Brown | Leather | 800 |
| M004 | S004 | Straw | PRD005 | Beige | Straw | 300 |
| M005 | S005 | Denim Fabric | PRD006 | Blue | Fabric | 600 |
| M006 | S006 | Cotton Fabric | PRD002 | Red | Fabric | 550 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

FIGURE 22:UPDATED TABLE SET-4

## 4.5 Delete data

```
MySQL 8.0 Command Line Cli  X  +  v
6 rows in set (0.00 sec)

mysql>
mysql> DELETE FROM DEPARTMENT WHERE Department_ID = 'D001';
Query OK, 1 row affected (0.00 sec)

mysql> DELETE FROM EMPLOYEE WHERE Employee_ID = 'E001';
Query OK, 1 row affected (0.00 sec)

mysql> DELETE FROM CHILD WHERE Child_Name = 'Child2' AND Employee_ID = 'E002';
Query OK, 1 row affected (0.00 sec)

mysql> DELETE FROM CUSTOMER WHERE Customer_ID = 'C001';
Query OK, 1 row affected (0.00 sec)

mysql> DELETE FROM CUS_PHONE_NUMBER WHERE Phone_Number = 0775643245 AND Customer_ID = 'C002';
Query OK, 1 row affected (0.00 sec)

mysql> DELETE FROM Customer_Order WHERE Order_ID = 'O001';
Query OK, 1 row affected (0.00 sec)

mysql> DELETE FROM DELIVERY WHERE Order_ID = 'O002' AND Delivery_ID = 'D002';
Query OK, 1 row affected (0.00 sec)

mysql> DELETE FROM PAYMENT WHERE Payment_ID = 'P001';
Query OK, 1 row affected (0.00 sec)

mysql> DELETE FROM PRODUCT WHERE Product_ID = 'PRD001';
Query OK, 1 row affected (0.00 sec)

mysql> DELETE FROM ORDER_ITEM WHERE Item_ID = 'I001';
Query OK, 1 row affected (0.00 sec)

mysql> DELETE FROM SUPPLIER WHERE Supplier_ID = 'S001';
Query OK, 1 row affected (0.00 sec)

mysql> DELETE FROM MATERIAL WHERE Material_ID = 'M001';
Query OK, 1 row affected (0.00 sec)

mysql> select *from department;
```



FIGURE 24:DELETE ONE ROW FROM ALL TABLES

```

MySQL 8.0 Command Line Cli  +  -
Query OK, 1 row affected (0.00 sec)

mysql> DELETE FROM MATERIAL WHERE Material_ID = 'M001';
Query OK, 1 row affected (0.00 sec)

mysql> select *from department;
+-----+-----+-----+-----+
| Department_ID | Department_Name | Description | Department_Head |
+-----+-----+-----+-----+
| D002 | IT | Information Technology | New Department Head |
| D003 | Finance | Finance and Accounting | Bob Johnson |
| D004 | Sales | Sales and Marketing | Eva Williams |
| D005 | Production | Production Department | Mike Brown |
| D006 | Research | Research and Development | Lisa Davis |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select *from employee;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Employee_ID | Department_ID | Email | Born_Year | Born_Month | Born_Date | Age | First_Name | Last_Name | Salary | Position |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| E002 | D002 | alice@example.com | 1999 | 8 | 20 | 32 | Alice | Smith | 60000 | Software Engineer |
| E003 | D002 | bob@example.com | 1985 | 3 | 10 | 37 | Bob | Johnson | 55000 | NULL |
| E004 | D004 | eva@example.com | 1992 | 11 | 5 | 29 | Eva | Williams | 52000 | Sales Executive |
| E005 | D004 | mike@example.com | 1988 | 6 | 25 | 34 | Mike | Brown | 48000 | Sales Executive |
| E006 | D006 | lisa@example.com | 1987 | 9 | 30 | 35 | Lisa | Davis | 56000 | Research Scientist |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select *from child;
+-----+-----+-----+-----+
| Child_Name | Employee_ID | Age | Gender |
+-----+-----+-----+-----+
| Child3 | E004 | 7 | Male |
| Child4 | E006 | 4 | Female |
| Child5 | E003 | 7 | Male |
| Child6 | E005 | 5 | Female |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

Windows taskbar at the bottom showing various icons and system status.

FIGURE 26:DELETED TABLE SET-1

```

MySQL 8.0 Command Line Cli  +  -
Query OK, 1 row affected (0.00 sec)

mysql> select *from custormer;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| First_Name | Last_Name | Customer_ID | Street_Name | City | State | PostalCode | Country | Email |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Nimal | Fernando | C002 | 456 Elm St | Kandy | Central | 20001 | Sri Lanka | nimal@example.com |
| Kamal | Silva | C003 | 789 Oak St | Galle | Southern | 30000 | Sri Lanka | kamal@example.com |
| Priya | Rajapakse | C004 | 101 Pine St | Jaffna | Northern | 40000 | Sri Lanka | newemail@example.com |
| Rohan | Bandara | C005 | 202 Cedar St | Negombo | Western | 50000 | Sri Lanka | rohan@example.com |
| Sujitha | Fernando | C006 | 303 Birch St | Matara | Southern | 60000 | Sri Lanka | sujitha@example.com |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select *from cus_phone_number;
+-----+-----+
| Phone_Number | Customer_ID |
+-----+-----+
| 771234567 | C003 |
| 765667788 | C004 |
| 712345678 | C005 |
| 727889900 | C006 |
+-----+-----+
4 rows in set (0.00 sec)

mysql> select *from customer_order;
+-----+-----+-----+-----+-----+-----+-----+
| Order_ID | Customer_ID | Employee_ID | Order_Date | Delivery_Date | Order_Status | Total_Amount |
+-----+-----+-----+-----+-----+-----+-----+
| 0002 | C002 | E002 | 2023-09-08 | 2023-09-16 | Shipped | 7500 |
| 0003 | C003 | E003 | 2023-09-09 | 2023-09-17 | Delivered | 6000 |
| 0004 | C004 | E004 | 2023-09-10 | 2023-09-18 | Shipped | 5500 |
| 0005 | C004 | E005 | 2023-09-11 | 2023-09-19 | Shipped | 8000 |
| 0006 | C006 | E006 | 2023-09-12 | 2023-09-20 | Delivered | 7000 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

Windows taskbar at the bottom showing various icons and system status.

FIGURE 25:DELETED TABLE SET-2

```

mysql> select *from delivery;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Order_ID | Delivery_ID | Street_Name | City | State | PostalCode | Country | Delivery_Date |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 0003 | D003 | 789 Oak St | Galle | Southern | 30000 | Sri Lanka | 2023-09-17 |
| 0004 | D004 | 101 Pine St | Jaffna | Northern | 40000 | Sri Lanka | 2023-09-18 |
| 0005 | D005 | 202 Cedar St | Negombo | Western | 50000 | Sri Lanka | 2023-09-19 |
| 0006 | D006 | 303 Birch St | Matara | Southern | 60000 | Sri Lanka | 2023-09-20 |
+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select *from payment;
+-----+-----+-----+-----+-----+
| Customer_ID | Payment_ID | Payment_Date | cash | Online_payment |
+-----+-----+-----+-----+-----+
| C002 | P002 | 2023-09-11 | 0 | 1 |
| C003 | P003 | 2023-09-12 | 1 | 0 |
| C004 | P004 | 2023-09-13 | 0 | 1 |
| C005 | P005 | 2023-09-14 | 1 | 0 |
| C006 | P006 | 2023-09-15 | 0 | 1 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select *from product;
+-----+-----+-----+-----+-----+-----+
| Product_ID | Product_Name | Product_Description | Size | Category | Price |
+-----+-----+-----+-----+-----+-----+
| PRD002 | Dress | New Product Description | S | Clothing | 2500 |
| PRD003 | Shoes | Leather shoes for men | 10 | Footwear | 3000 |
| PRD004 | Handbag | Leather handbag for women | Medium | Accessories | 1500 |
| PRD005 | Hat | Sun hat for all | One Size | Accessories | 800 |
| PRD006 | Jeans | Denim jeans for men | 32 | Clothing | 1800 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

FIGURE 28:DELETED TABLE SET-3

```

mysql> select *from order_item;
+-----+-----+-----+-----+-----+
| Customer_ID | Item_ID | Product_ID | Quantity | Price |
+-----+-----+-----+-----+-----+
| C002 | I002 | NULL | 1 | 2700 |
| C003 | I003 | PRD003 | 1 | 3000 |
| C004 | I004 | PRD003 | 3 | 4500 |
| C005 | I005 | PRD005 | 2 | 1600 |
| C006 | I006 | PRD006 | 2 | 3600 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select *from supplier;
+-----+-----+-----+-----+-----+-----+
| First_Name | Last_Name | Supplier_ID | Contact_Person | Email | Phone_Number |
+-----+-----+-----+-----+-----+-----+
| XYZ Footwear | Suppliers | S002 | Alice Supplier | newsupplier@example.com | 773456789 |
| Leather World | Exports | S003 | Bob Supplier | bobsupplier@example.com | 774567890 |
| FashionHub | Supplies | S004 | Eva Supplier | evasupplier@example.com | 775678901 |
| Accessories Plus | Imports | S005 | Mike Supplier | mikesupplier@example.com | 776789012 |
| Jeans Inc | International | S006 | Lisa Supplier | lisasupplier@example.com | 777890123 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select *from material;
+-----+-----+-----+-----+-----+-----+-----+
| Material_ID | Supplier_ID | Material_Name | Product_ID | Colour | Material_Type | Price_Per_Unit |
+-----+-----+-----+-----+-----+-----+-----+
| M002 | S002 | Leather | PRD003 | New Colour | Leather | 1000 |
| M003 | S003 | Leather | PRD004 | Brown | Leather | 800 |
| M004 | S004 | Straw | PRD005 | Beige | Straw | 300 |
| M005 | S005 | Denim Fabric | PRD006 | Blue | Fabric | 600 |
| M006 | S006 | Cotton Fabric | PRD002 | Red | Fabric | 550 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

FIGURE 27:DELETED TABLE SET-4

## 5 Chapter 5 – Transactions

## 5.1 Simple queries

```
MySQL 8.0 Command Line Cli x + v
mysql>
mysql> SELECT First_Name, Email FROM CUSTOMER;
+-----+-----+
| First_Name | Email
+-----+-----+
| Nimal      | nimal@example.com
| Kamal      | kamal@example.com
| Priya      | newemail@example.com
| Rohan      | rohan@example.com
| Sujitha    | sujitha@example.com
+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT Customer_ID, Order_Status FROM Customer_Order;
+-----+-----+
| Customer_ID | Order_Status |
+-----+-----+
| C002        | Shipped
| C003        | Delivered
| C004        | Shipped
| C004        | Shipped
| C006        | Delivered
+-----+-----+
5 rows in set (0.00 sec)
```

FIGURE 30: SELECT OPERATION AND PROJECT OPERATION

```
MySQL>
MySQL>
MySQL> SELECT * FROM EMPLOYEE CROSS JOIN CHILD;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Employee_ID | Department_ID | Email           | Born_Year | Born_Month | Born_Date | Age   | First_Name | Last_Name | Salary | Position        | Child_Name | Employee_ID | Age   | Gender |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| E002       | D002          | alice@example.com | 1990     | 8          | 28      | 32   | Alice      | Smith    | 60000 | Software Engineer | Child6    | E005      | 5     | Female |
| E002       | D002          | alice@example.com | 1990     | 8          | 28      | 32   | Alice      | Smith    | 60000 | Software Engineer | Child5    | E003      | 7     | Male   |
| E002       | D002          | alice@example.com | 1999     | 8          | 28      | 32   | Alice      | Smith    | 60000 | Software Engineer | Child4    | E006      | 4     | Female |
| E002       | D002          | alice@example.com | 1999     | 8          | 28      | 32   | Alice      | Smith    | 60000 | Software Engineer | Child3    | E004      | 7     | Male   |
| E003       | D002          | bob@example.com   | 1985     | 3          | 10      | 37   | Bob        | Johnson  | 55000 | NULL             | Child6    | E005      | 5     | Female |
| E003       | D002          | bob@example.com   | 1985     | 3          | 10      | 37   | Bob        | Johnson  | 55000 | NULL             | Child5    | E003      | 7     | Male   |
| E003       | D002          | bob@example.com   | 1985     | 3          | 10      | 37   | Bob        | Johnson  | 55000 | NULL             | Child4    | E006      | 4     | Female |
| E003       | D002          | bob@example.com   | 1985     | 3          | 10      | 37   | Bob        | Johnson  | 55000 | NULL             | Child3    | E004      | 7     | Male   |
| E004       | D004          | eva@example.com   | 1992     | 11         | 5       | 29   | Eva        | Williams | 52000 | Sales Executive  | Child6    | E005      | 5     | Female |
| E004       | D004          | eva@example.com   | 1992     | 11         | 5       | 29   | Eva        | Williams | 52000 | Sales Executive  | Child5    | E003      | 7     | Male   |
| E004       | D004          | eva@example.com   | 1992     | 11         | 5       | 29   | Eva        | Williams | 52000 | Sales Executive  | Child4    | E006      | 4     | Female |
| E004       | D004          | eva@example.com   | 1992     | 11         | 5       | 29   | Eva        | Williams | 52000 | Sales Executive  | Child3    | E004      | 7     | Male   |
| E005       | D004          | mike@example.com | 1988     | 6          | 25      | 34   | Mike       | Brown    | 48000 | Sales Executive  | Child6    | E005      | 5     | Female |
| E005       | D004          | mike@example.com | 1988     | 6          | 25      | 34   | Mike       | Brown    | 48000 | Sales Executive  | Child5    | E003      | 7     | Male   |
| E005       | D004          | mike@example.com | 1988     | 6          | 25      | 34   | Mike       | Brown    | 48000 | Sales Executive  | Child4    | E006      | 4     | Female |
| E005       | D004          | mike@example.com | 1988     | 6          | 25      | 34   | Mike       | Brown    | 48000 | Sales Executive  | Child3    | E004      | 7     | Male   |
| E006       | D006          | lisa@example.com | 1987     | 9          | 30      | 35   | Lisa       | Davis    | 56000 | Research Scientist | Child6    | E005      | 5     | Female |
| E006       | D006          | lisa@example.com | 1987     | 9          | 30      | 35   | Lisa       | Davis    | 56000 | Research Scientist | Child5    | E003      | 7     | Male   |
| E006       | D006          | lisa@example.com | 1987     | 9          | 30      | 35   | Lisa       | Davis    | 56000 | Research Scientist | Child4    | E006      | 4     | Female |
| E006       | D006          | lisa@example.com | 1987     | 9          | 30      | 35   | Lisa       | Davis    | 56000 | Research Scientist | Child3    | E004      | 7     | Male   |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
20 rows in set (0.00 sec)

MySQL> CREATE VIEW CustomerOrders AS
-> SELECT C.First_Name, CO.Order_Status
-> FROM CUSTOMER C
-> JOIN Customer_Order CO ON C.Customer_ID = CO.Customer_ID;
Query OK, 0 rows affected (0.01 sec)

MySQL> select * from customerOrders;
+-----+-----+
| First_Name | Order_Status |
+-----+-----+
| Nimal     | Shipped    |
| Kamal    | Delivered  |
| Priya    | Shipped    |
| Priya    | Shipped    |
| Sujitha  | Delivered  |
+-----+-----+
5 rows in set (0.00 sec)

MySQL> |
```

FIGURE 29: CARTESIAN PRODUCT OPERATION AND CREATING A USER VIEW

```
MySQL 8.0 Command Line Cli X + v - O X
5 rows in set (0.00 sec)

mysql> alter table SUPPLIER
    -> RENAME COLUMN EMAIL to Customer_Email;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> select *from supplier;
+-----+-----+-----+-----+-----+-----+
| First_Name | Last_Name | Supplier_ID | Contact_Person | Customer_Email | Phone_Number |
+-----+-----+-----+-----+-----+-----+
| XYZ Footwear | Suppliers | S002 | Alice Supplier | newsupplier@example.com | 773456789 |
| Leather World | Exports | S003 | Bob Supplier | bobsupplier@example.com | 774567890 |
| FashionHub | Supplies | S004 | Eva Supplier | evasupplier@example.com | 775678901 |
| Accessories Plus | Imports | S005 | Mike Supplier | mikesupplier@example.com | 776789012 |
| Jeans Inc | International | S006 | Lisa Supplier | lisasupplier@example.com | 777890123 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

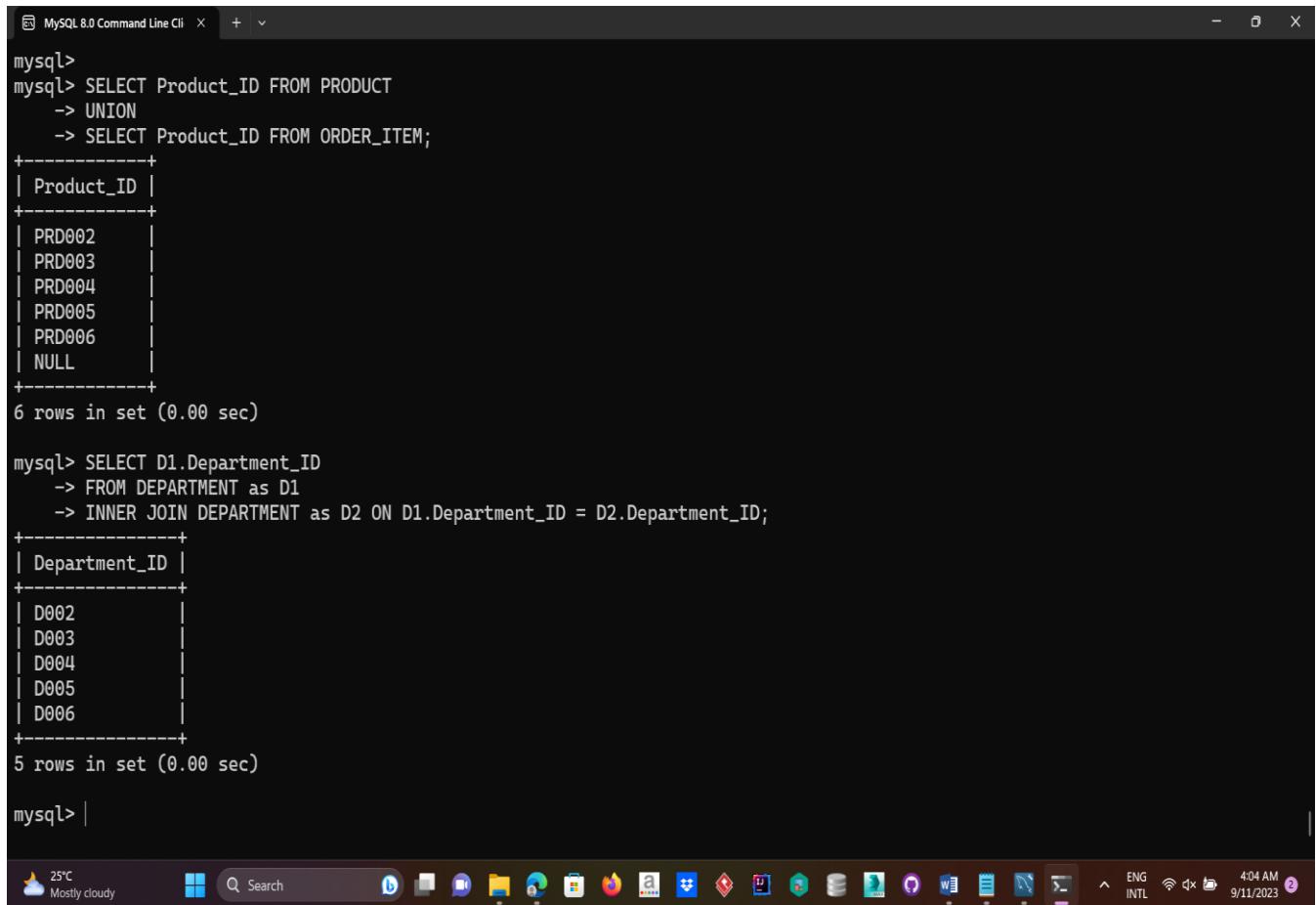
mysql> SELECT AVG(Salary) AS Average_Salary FROM EMPLOYEE;
+-----+
| Average_Salary |
+-----+
| 54200.0000 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT Email FROM CUSTOMER WHERE Email LIKE "%example.com%";
+-----+
| Email |
+-----+
| nimal@example.com |
| kamal@example.com |
| newemail@example.com |
| rohan@example.com |
| sujitha@example.com |
+-----+
5 rows in set (0.00 sec)

mysql> |
```

FIGURE 31:RENAMING OPERATION , AGGREGATION FUNCTION AND USE OF LIKE KEYWORD

## 5.2 Complex queries



MySQL 8.0 Command Line Cli

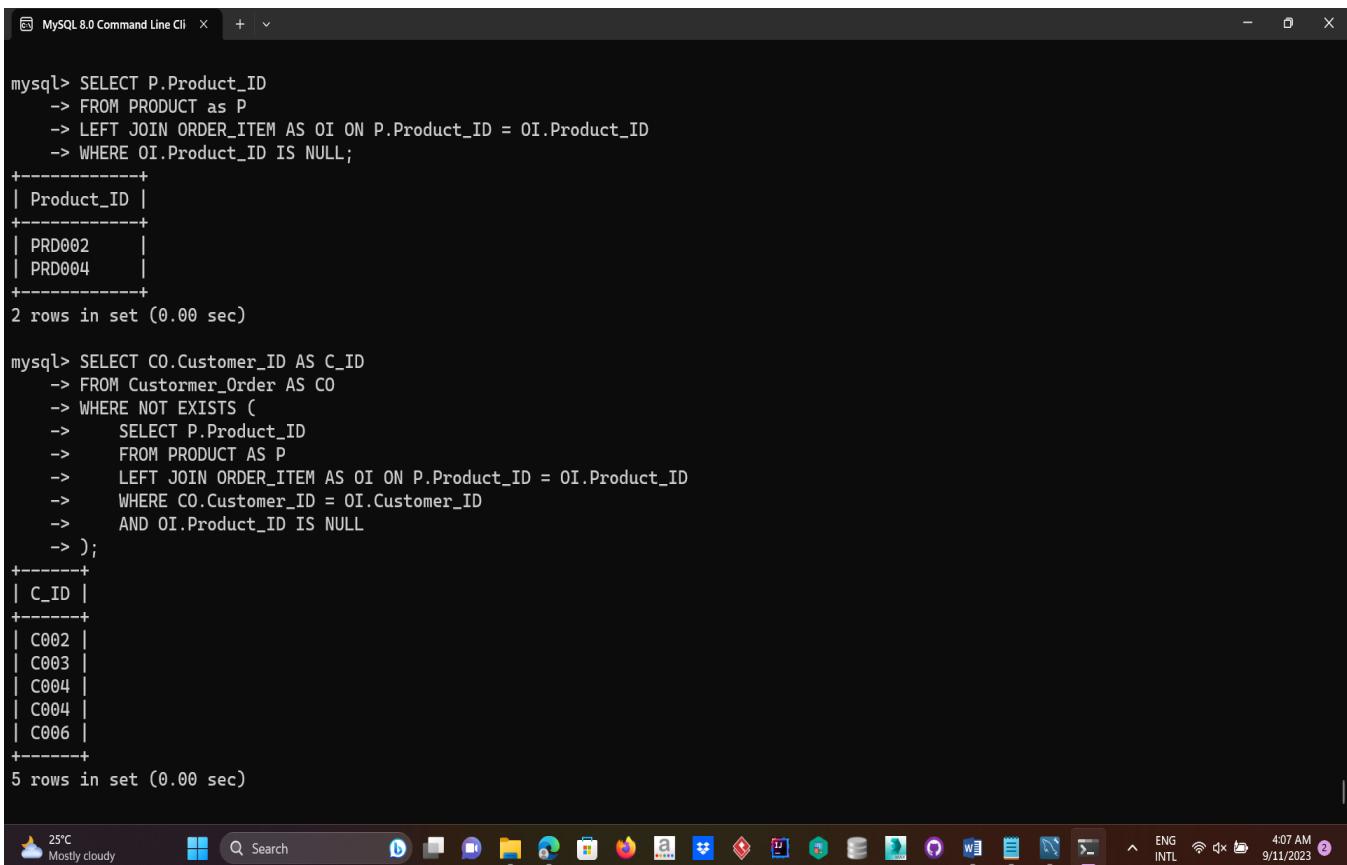
```
mysql> SELECT Product_ID FROM PRODUCT
-> UNION
-> SELECT Product_ID FROM ORDER_ITEM;
+-----+
| Product_ID |
+-----+
| PRD002    |
| PRD003    |
| PRD004    |
| PRD005    |
| PRD006    |
| NULL      |
+-----+
6 rows in set (0.00 sec)

mysql> SELECT D1.Department_ID
-> FROM DEPARTMENT as D1
-> INNER JOIN DEPARTMENT as D2 ON D1.Department_ID = D2.Department_ID;
+-----+
| Department_ID |
+-----+
| D002          |
| D003          |
| D004          |
| D005          |
| D006          |
+-----+
5 rows in set (0.00 sec)

mysql> |
```

25°C Mostly cloudy Search ENG INTL 4:04 AM 9/11/2023

FIGURE 32:UNION OPERATION AND INTERSECTION OPERATION



MySQL 8.0 Command Line Cli

```
mysql> SELECT P.Product_ID
-> FROM PRODUCT as P
-> LEFT JOIN ORDER_ITEM AS OI ON P.Product_ID = OI.Product_ID
-> WHERE OI.Product_ID IS NULL;
+-----+
| Product_ID |
+-----+
| PRD002    |
| PRD004    |
+-----+
2 rows in set (0.00 sec)

mysql> SELECT CO.Customer_ID AS C_ID
-> FROM Custormer_Order AS CO
-> WHERE NOT EXISTS (
->     SELECT P.Product_ID
->     FROM PRODUCT AS P
->     LEFT JOIN ORDER_ITEM AS OI ON P.Product_ID = OI.Product_ID
->     WHERE CO.Customer_ID = OI.Customer_ID
->     AND OI.Product_ID IS NULL
-> );
+-----+
| C_ID |
+-----+
| C002 |
| C003 |
| C004 |
| C006 |
+-----+
5 rows in set (0.00 sec)
```

25°C Mostly cloudy Search ENG INTL 4:04 AM 9/11/2023

FIGURE 33: SET DIFFERENCE OPERATION AND DIVISION OPERATION

```

MySQL 8.0 Command Line Cli  X  +  v

5 rows in set (0.00 sec)

mysql> CREATE VIEW EmployeeDepartment AS
-> SELECT E.First_Name, E.Last_Name, D.Department_Name
-> FROM EMPLOYEE E
-> INNER JOIN DEPARTMENT D ON E.Department_ID = D.Department_ID;
Query OK, 0 rows affected (0.01 sec)

mysql> select *from employeedepartment;
+-----+-----+-----+
| First_Name | Last_Name | Department_Name |
+-----+-----+-----+
| Alice      | Smith     | IT           |
| Bob        | Johnson   | IT           |
| Eva        | Williams  | Sales        |
| Mike       | Brown     | Sales        |
| Lisa       | Davis     | Research    |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> CREATE VIEW cus_order AS
-> SELECT C.First_Name, C.Last_Name, CO.Order_Status , CO.Total_Amount
-> FROM CUSTOMER as C
-> NATURAL JOIN Custormer_Order as CO;
Query OK, 0 rows affected (0.00 sec)

mysql> select *from cus_order;
+-----+-----+-----+-----+
| First_Name | Last_Name | Order_Status | Total_Amount |
+-----+-----+-----+-----+
| Nimal      | Fernando  | Shipped     | 7500          |
| Kamal      | Silva     | Delivered   | 6000          |
| Priya      | Rajapakse | Shipped     | 5500          |
| Priya      | Rajapakse | Shipped     | 8000          |
| Sujitha   | Fernando  | Delivered   | 7000          |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>

```

FIGURE 34:INNER JOIN AND NATURAL JOIN

```

MySQL 8.0 Command Line Cli  X  +  v

mysql> CREATE VIEW employeeDep AS
-> SELECT D.Department_Name,E.First_Name, E.Last_Name
-> FROM DEPARTMENT as D
-> LEFT JOIN EMPLOYEE as E ON E.Department_ID = D.Department_ID;
Query OK, 0 rows affected (0.01 sec)

mysql> select *from employeeDep;
+-----+-----+-----+
| Department_Name | First_Name | Last_Name |
+-----+-----+-----+
| IT             | Alice      | Smith      |
| IT             | Bob        | Johnson    |
| Finance        | NULL       | NULL       |
| Sales           | Eva        | Williams   |
| Sales           | Mike       | Brown      |
| Production     | NULL       | NULL       |
| Research        | Lisa       | Davis      |
+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> CREATE VIEW employeePosition AS
-> SELECT D.Department_Name, E.First_Name, IFNULL(E.Position, 'No position') AS Position
-> FROM DEPARTMENT D
-> RIGHT JOIN EMPLOYEE E ON D.Department_ID = E.Department_ID;
Query OK, 0 rows affected (0.00 sec)

mysql> select *from employeePosition;
+-----+-----+-----+
| Department_Name | First_Name | Position   |
+-----+-----+-----+
| IT             | Alice      | Software Engineer |
| IT             | Bob        | No position |
| Sales          | Eva        | Sales Executive |
| Sales          | Mike       | Sales Executive |
| Research        | Lisa       | Research Scientist |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> |

```

FIGURE 35:LEFT OUTER JOIN AND RIGHT OUTER JOIN

```

MySQL 8.0 Command Line Cli X + - O X

mysql> CREATE VIEW FullOuterJoinView AS
-> SELECT CO.Order_ID AS CO_Order_ID, CO.Customer_ID AS CO_Customer_ID, CO.Employee_ID AS CO_Employee_ID, CO.Order_Date AS CO_Order_Date, CO.Delivery_Date AS CO_Delivery_Date, CO.Order_Status AS CO_Order_Status, CO.Total_Amount AS CO_Total_Amount,
-> C.First_Name AS C_First_Name, C.Last_Name AS C_Last_Name, C.Street_Name AS C_Street_Name, C.City AS C_City, C.State AS C_State, C.PostalCode AS C_PostalCode, C.Country AS C_Country, C.Email AS C_Email
-> FROM Customer_Order CO
-> LEFT JOIN CUSTOMER C ON CO.Customer_ID = C.Customer_ID
-> UNION ALL
-> SELECT CO.Order_ID AS CO_Order_ID, CO.Customer_ID AS CO_Customer_ID, CO.Employee_ID AS CO_Employee_ID, CO.Order_Date AS CO_Order_Date, CO.Delivery_Date AS CO_Delivery_Date, CO.Order_Status AS CO_Order_Status, CO.Total_Amount AS CO_Total_Amount,
-> C.First_Name AS C_First_Name, C.Last_Name AS C_Last_Name, C.Street_Name AS C_Street_Name, C.City AS C_City, C.State AS C_State, C.PostalCode AS C_PostalCode, C.Country AS C_Country, C.Email AS C_Email
-> FROM Customer_Order CO
-> RIGHT JOIN CUSTOMER C ON CO.Customer_ID = C.Customer_ID;
Query OK, 9 rows affected (1.51 sec)

mysql> select *from fullouterjoinview;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| CO.Order_ID | CO.Customer_ID | CO.Employee_ID | CO.Order_Date | CO.Delivery_Date | CO.Order_Status | CO.Total_Amount | C.First_Name | C.Last_Name | C.Street_Name | C.City | C.State | C.PostalCode | C.Country | C.Email |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 0002 | C002 | E002 | 2023-09-08 | 2023-09-16 | Shipped | 7500 | Nimal | Fernando | 456 Elm St | Kandy | Central | 200001 | Sri Lanka | nimal@example.com |
| 0003 | C003 | E003 | 2023-09-09 | 2023-09-17 | Delivered | 6000 | Kamal | Silva | 789 Oak St | Galle | Southern | 300000 | Sri Lanka | kamal@example.com |
| 0004 | C004 | E004 | 2023-09-10 | 2023-09-18 | Shipped | 5500 | Priya | Rajapakse | 101 Pine St | Jaffna | Northern | 400000 | Sri Lanka | newemail@example.com |
| 0005 | C005 | E005 | 2023-09-11 | 2023-09-19 | Shipped | 8000 | Priya | Rajapakse | 101 Pine St | Jaffna | Northern | 400000 | Sri Lanka | newemail@example.com |
| 0006 | C006 | E006 | 2023-09-12 | 2023-09-20 | Delivered | 7000 | Sujitha | Fernando | 303 Birch St | Matara | Southern | 600000 | Sri Lanka | sujitha@example.com |
| 0007 | C007 | E007 | 2023-09-08 | 2023-09-16 | Shipped | 7500 | Nimal | Fernando | 456 Elm St | Kandy | Central | 200001 | Sri Lanka | nimal@example.com |
| 0008 | C008 | E008 | 2023-09-09 | 2023-09-17 | Delivered | 6000 | Kamal | Silva | 789 Oak St | Galle | Southern | 300000 | Sri Lanka | kamal@example.com |
| 0009 | C009 | E009 | 2023-09-10 | 2023-09-18 | Shipped | 5500 | Priya | Rajapakse | 101 Pine St | Jaffna | Northern | 400000 | Sri Lanka | newemail@example.com |
| 0010 | C010 | E010 | 2023-09-11 | 2023-09-19 | Shipped | 8000 | Priya | Rajapakse | 101 Pine St | Jaffna | Northern | 400000 | Sri Lanka | newemail@example.com |
| NULL |
| 0006 | C006 | E006 | 2023-09-12 | 2023-09-20 | Delivered | 7000 | Sujitha | Fernando | 303 Birch St | Matara | Southern | 600000 | Sri Lanka | sujitha@example.com |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql> CREATE VIEW SupplierMaterial AS
-> SELECT S.Supplier_ID, S.First_Name AS Supplier_First_Name, S.Last_Name AS Supplier_Last_Name,
-> M.Material_ID, M.Material_Name, M.Product_ID, M.Colour, M.Material_Type, M.Price_Per_Unit
-> FROM SUPPLIER S
-> UNION ALL
-> SELECT S.Supplier_ID, S.First_Name AS Supplier_First_Name, S.Last_Name AS Supplier_Last_Name,
-> M.Material_ID, M.Material_Name, M.Product_ID, M.Colour, M.Material_Type, M.Price_Per_Unit
-> FROM SUPPLIER S
-> RIGHT JOIN MATERIAL M ON S.Supplier_ID = M.Supplier_ID;
Query OK, 0 rows affected (0.01 sec)

mysql> select *from suppliermaterial;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Supplier_ID | Supplier_First_Name | Supplier_Last_Name | Material_ID | Material_Name | Product_ID | Colour | Material_Type | Price_Per_Unit |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 0002 | XYZ Footwear | Suppliers | M002 | Leather | PRD0001 | New Colour | Leather | 1000 |
| 0003 | Leather World | Exports | M003 | Leather | PRD0004 | Brown | Leather | 800 |
| 0004 | FashionHub | Supplies | M004 | Straw | PRD0005 | Beige | Fabric | 300 |
| 0005 | Accessories Plus | Imports | M005 | Denim Fabric | PRD0006 | Blue | Fabric | 600 |
| 0006 | Jeans Inc | International | M006 | Cotton Fabric | PRD0002 | Red | Fabric | 550 |
| 0002 | XYZ Footwear | Suppliers | M002 | Leather | PRD0003 | New Colour | Leather | 1000 |
| 0003 | Leather World | Exports | M003 | Leather | PRD0004 | Brown | Leather | 800 |
| 0004 | FashionHub | Supplies | M004 | Straw | PRD0005 | Beige | Fabric | 300 |
| 0005 | Accessories Plus | Imports | M005 | Denim Fabric | PRD0006 | Blue | Fabric | 600 |
| 0006 | Jeans Inc | International | M006 | Cotton Fabric | PRD0002 | Red | Fabric | 550 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>


```

**FIGURE 36:FULL OUTER JOIN AND OUTER UNION**

```

MySQL 8.0 Command Line Cli X + - O X

mysql> SELECT First_Name, Email
-> FROM CUSTOMER
-> WHERE Email LIKE '%example.com'
-> UNION
-> SELECT C.First_Name, 'Order Placed' AS Email
-> FROM CUSTOMER C
-> WHERE C.Customer_ID IN (SELECT DISTINCT Customer_ID FROM Customer_Order);
+-----+-----+
| First_Name | Email |
+-----+-----+
| Nimal | nimal@example.com |
| Kamal | kamal@example.com |
| Priya | newemail@example.com |
| Rohan | rohan@example.com |
| Sujitha | sujitha@example.com |
| Nimal | Order Placed |
| Kamal | Order Placed |
| Priya | Order Placed |
| Sujitha | Order Placed |
+-----+-----+
9 rows in set (0.00 sec)

mysql>
mysql>
mysql> SELECT C.First_Name, CO.Order_Status
-> FROM CUSTOMER C
-> JOIN Customer_Order CO ON C.Customer_ID = CO.Customer_ID
-> WHERE CO.Order_ID IN (SELECT Order_ID FROM PAYMENT);
+-----+-----+
| First_Name | Order_Status |
+-----+-----+
| Nimal | Shipped |
| Kamal | Delivered |
| Priya | Shipped |
| Priya | Shipped |
| Sujitha | Delivered |
+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT C.First_Name, CO.Order_Status
-> FROM CUSTOMER C
-> JOIN Customer_Order CO ON C.Customer_ID = CO.Customer_ID
-> WHERE CO.Order_ID IN (
->     SELECT Order_ID
->     FROM PAYMENT
-> );
+-----+-----+
| First_Name | Order_Status |
+-----+-----+
| Nimal | Shipped |
| Kamal | Delivered |
| Priya | Shipped |
| Priya | Shipped |
| Sujitha | Delivered |
+-----+-----+
5 rows in set (0.00 sec)

mysql>


```

**FIGURE 37:NESTED QUERIES**

## 6 Chapter 6 - Database Tuning

mysql> explain SELECT Product\_ID FROM PRODUCT  
-> UNION  
-> SELECT Product\_ID FROM ORDER\_ITEM;  
+-----+  
| id | select\_type | table | partitions | type | possible\_keys | key | key\_len | ref | rows | filtered | Extra |  
+-----+  
1	PRIMARY	PRODUCT	NULL	index	NULL	PRIMARY	42	NULL	5	100.00	Using index
2	UNION	ORDER\_ITEM	NULL	index	NULL	Product\_ID	43	NULL	5	100.00	Using index
3	UNION RESULT	<union1,2>	NULL	ALL	NULL	NULL	NULL	NULL	NULL	NULL	Using temporary
+-----+  
3 rows in set, 1 warning (0.00 sec)  
  
mysql>

FIGURE 39: BEFORE TUNING UNION OPERATION

MySQL Workbench  
Local instance MySQL80 X  
File Edit View Query Database Server Tools Scripting Help  
Navigator Schemas  
Filter objects  
final  
551  
552 -- after tuning  
553 • explain SELECT Product\_ID FROM PRODUCT  
554 UNION  
555 SELECT Product\_ID FROM ORDER\_ITEM;

Result Grid | Filter Rows: Export: Wrap Cell Content:  Result Grid Form Editor Field Types Query Stats Execution Plan  
1 PRIMARY PRODUCT NULL index NULL PRIMARY 42 NULL 5 100.00 Using index  
2 UNION ORDER\_ITEM NULL index NULL Product\_ID 43 NULL 5 100.00 Using index  
3 UNION RESULT <union1,2> NULL ALL NULL NULL NULL NULL NULL NULL Using temporary

No object selected

Result 58 X Read Only Context Help Snippets  
Object Info Session Output Action Output # Time Action Message Duration / Fetch  
27°C Mostly cloudy Search ENG INTL 245 PM 9/11/2023

FIGURE 38: AFTER TUNING UNION OPERATION

```

MySQL> explain SELECT D1.Department_ID
    -> FROM DEPARTMENT as D1
    -> INNER JOIN DEPARTMENT as D2 ON D1.Department_ID = D2.Department_ID;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id  | select_type | table | partitions | type  | possible_keys | key   | key_len | ref
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1   | SIMPLE     | D1   | NULL      | index | PRIMARY       | PRIMARY | 42    | NULL
| 1   | SIMPLE     | D2   | NULL      | eq_ref | PRIMARY       | PRIMARY | 42    | garment_database_system.D1.Department_ID
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>

```

The screenshot shows the MySQL 8.0 Command Line Cli window. The command `explain` is used to analyze the execution plan for a query involving two tables, `DEPARTMENT` (aliased as D1 and D2). The output shows two rows in the result set, indicating the execution plan for each table. The first row (D1) uses an index for the `Department\_ID` column. The second row (D2) uses an equal reference (`eq\_ref`) to the primary key of D1. The `Extra` column indicates the use of indexes for both rows.

FIGURE 40:BEFORE TUNING INTERSECTION OPERATION

```

563  -- after tuning
564 •  explain SELECT D1.Department_ID
565   FROM DEPARTMENT as D1
566   INNER JOIN DEPARTMENT as D2 ON D1.Department_ID = D2.Department_ID;
567

```

The screenshot shows the MySQL Workbench interface. A query editor window titled "final" contains the same `explain` command as Figure 40, but with the prefix `-- after tuning`. The results grid shows the same execution plan as Figure 40, with two rows and the same analysis for using indexes.

FIGURE 41:AFTER TUNING INTERSECTION OPERATION

```

mysql> explain SELECT P.Product_ID
   > FROM PRODUCT as P
   > LEFT JOIN ORDER_ITEM AS OI ON P.Product_ID = OI.Product_ID
   > WHERE OI.Product_ID IS NULL;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE     | P      | NULL       | index | NULL          | PRIMARY | 42    | NULL  | 5    | 100.00  | Using index |
| 1 | SIMPLE     | OI     | NULL       | ref   | Product_ID   | Product_ID | 43    |garment_database_system.P.Product_ID| 1    | 100.00  | Using where; Using index |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql> |

```

The screenshot shows the MySQL Command Line Client interface. A query is being executed to explain the execution plan for a SELECT statement. The output shows two rows in the result set, both using simple select types. The first row uses an index for the table 'P' and the second row uses a ref for the table 'OI'. Both rows are using the primary key for their respective tables. The 'Extra' column indicates the use of an index for the first row and the use of a where clause and index for the second row.

FIGURE 42:BEFORE TUNING SET DIFFERENCE OPERATION

```

570 • explain SELECT P.Product_ID
571   FROM PRODUCT as P
572   LEFT JOIN ORDER_ITEM AS OI ON P.Product_ID = OI.Product_ID
573   WHERE OI.Product_ID IS NULL;

```

The screenshot shows the MySQL Workbench interface. A query is being explained in the SQL editor. The output shows the same execution plan as Figure 42, with two rows in the result set. The first row is a simple select from 'P' using an index, and the second row is a simple select from 'OI' using a ref. The 'Extra' column indicates the use of an index for the first row and the use of a where clause and index for the second row.

FIGURE 43:AFTER TUNING SET DIFFERENCE OPERATION

```

MySQL> explain SELECT CO.Customer_ID AS C_ID
   > FROM Customer_Order AS CO
   > WHERE NOT EXISTS (
   >   SELECT P.Product_ID
   >   FROM PRODUCT AS P
   >   LEFT JOIN ORDER_ITEM AS OI ON P.Product_ID = OI.Product_ID
   >   WHERE CO.Customer_ID = OI.Customer_ID
   >   AND OI.Product_ID IS NULL
   > );

```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	CO	NULL	index	NULL	Customer_ID	43	NULL	5	100.00	Using index
1	SIMPLE	<subquery2>	NULL	eq_ref	<auto_distinct_key>	<auto_distinct_key>	43	garment_database_system.CO.Customer_ID	1	100.00	Using where; Not exists
2	MATERIALIZED	OI	NULL	ref	Customer_ID,Product_ID	Product_ID	43	const	1	100.00	Using index condition
2	MATERIALIZED	P	NULL	eq_ref	PRIMARY	PRIMARY	42	garment_database_system.OI.Product_ID	1	100.00	Using index

4 rows in set, 2 warnings (0.00 sec)

mysql>

FIGURE 44: BEFORE TUNING DIVISION OPERATION

```

explain SELECT CO.Customer_ID AS C_ID
   > FROM Customer_Order AS CO
   > WHERE NOT EXISTS (
   >   SELECT P.Product_ID
   >   FROM PRODUCT AS P
   >   LEFT JOIN ORDER_ITEM AS OI ON P.Product_ID = OI.Product_ID
   >   WHERE CO.Customer_ID = OI.Customer_ID
   >   AND OI.Product_ID IS NULL
   > );

```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	CO	NULL	index	NULL	idx_customer_order_customer_id2	43	NULL	5	1	Using index
1	SIMPLE	<subquery2>	NULL	eq_ref	<auto_distinct_key>	<auto_distinct_key>	43	garment_database_system.CO.Customer_ID	1	1	Using where; Not exists
2	MATERIALIZED	OI	NULL	ref	Customer_ID,Product_ID	Product_ID	43	const	1	1	Using index condition
2	MATERIALIZED	P	NULL	eq_ref	PRIMARY	PRIMARY	42	garment_database_system.OI.Product_ID	1	1	Using index

FIGURE 45: AFTER TUNING DIVISION OPERATION

```

mysql> explain SELECT E.First_Name, E.Last_Name, D.Department_Name
   FROM EMPLOYEE E
   INNER JOIN DEPARTMENT D ON E.Department_ID = D.Department_ID;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE     | E    | NULL      | ALL  | Department_ID | PRIMARY | 42    | NULL  |
| 1 | SIMPLE     | D    | NULL      | eq_ref| PRIMARY       | PRIMARY | 42    | garment_database_system.E.Department_ID |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>

```

The screenshot shows the MySQL Command Line Client interface. The command `explain` is used to analyze a query. The output shows one simple select type query (id 1) that performs a full table scan (type ALL) on the EMPLOYEE table and uses a primary key (type eq\_ref) from the DEPARTMENT table. The query retrieves columns First\_Name, Last\_Name, and Department\_Name. The results indicate 2 rows in set and 1 warning.

FIGURE 46:BEFORE TUNING INNER JOIN

```

-- after tuning
601 • CREATE INDEX idx_employee_Dip_ID ON EMPLOYEE (Department_ID);
602 • CREATE INDEX idx_employee_first_name ON EMPLOYEE (First_Name);
603 • CREATE INDEX idx_employee_last_name ON EMPLOYEE (Last_Name);
604
605 • explain SELECT E.First_Name, E.Last_Name, D.Department_Name
606   FROM EMPLOYEE E
607   INNER JOIN DEPARTMENT D ON E.Department_ID = D.Department_ID;
608

```

The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, three new indexes are created on the EMPLOYEE table: `idx\_employee\_Dip\_ID` on `Department\_ID`, `idx\_employee\_first\_name` on `First\_Name`, and `idx\_employee\_last\_name` on `Last\_Name`. Below the indexes, the `explain` command is run again with the same query. The output shows the query now using the newly created indexes for both tables, resulting in a more efficient execution plan.

FIGURE 47:AFTER TUNING INNER JOIN

```

MySQL> explain SELECT C.First_Name, C.Last_Name, CO.Order_Status , CO.Total_Amount
   > FROM CUSTOMER AS C
   > NATURAL JOIN Customer_Order AS CO;
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE      | C     | NULL       | ALL  | PRIMARY        | NULL | NULL    | NULL  |
| 1 | SIMPLE      | CO    | NULL       | ref  | Customer_ID   | Customer_ID | 43    | garment_database_system.C.Customer_ID |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>

```

The screenshot shows the MySQL Command Line Client interface. The command `explain` is used to analyze the query `SELECT C.First\_Name, C.Last\_Name, CO.Order\_Status , CO.Total\_Amount FROM CUSTOMER AS C NATURAL JOIN Customer\_Order AS CO;`. The output shows two rows in the result set, with one warning. The `rows` column indicates 5 rows, and the `filtered` column indicates 100.00%. The `Extra` column shows `Using index`.

FIGURE 48: BEFORE TUNING NATURAL JOIN

```

-- after tuning
616 • CREATE INDEX idx_customer_order_customer_id2 ON Customer_Order (Customer_ID);
617 • CREATE INDEX idx_customer_customer_name ON CUSTOMER (First_Name);
618 • CREATE INDEX idx_customer_customer_name2 ON CUSTOMER (Last_Name);
619 • CREATE INDEX idx_customer_order_customer_id3 ON Customer_Order (Customer_ID);
620 • CREATE INDEX idx_customer_order_order_status2 ON Customer_Order (Order_Status);
621 • CREATE INDEX idx_customer_order_total_amount2 ON Customer_Order (Total_Amount);
622
623 • explain SELECT C.First_Name, C.Last_Name, CO.Order_Status , CO.Total_Amount
   FROM CUSTOMER AS C
   NATURAL JOIN Customer_Order AS CO;
624
625
626

```

The screenshot shows the MySQL Workbench interface. In the SQL editor, several indexes are created on the `Customer` and `Customer\_Order` tables. Then, the same query as in Figure 48 is run again, but this time it uses an index. The results show the same 5 rows and 100.00% filtering, but the `Extra` column now shows `Using index`.

FIGURE 49: AFTER JOINING NATURAL JOIN

```

MySQL 8.0 Command Line Cli + ×
mysql> explain
--> SELECT D.Department_Name,E.First_Name, E.Last_Name
--> FROM DEPARTMENT as D
--> LEFT JOIN EMPLOYEE as E ON E.Department_ID = D.Department_ID;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id  | select_type | table | partitions | type | possible_keys | key | key_len | ref  | rows | filtered | Extra
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1   | SIMPLE      | D    | NULL       | ALL  | NULL          | NULL | NULL    | NULL  | 5    | 100.00    | Using where; Using join buffer (hash join)
| 1   | SIMPLE      | E    | NULL       | ALL  | Department_ID | NULL | NULL    | NULL  | 5    | 100.00    | Using where; Using join buffer (hash join)
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql> |

```

The screenshot shows the MySQL Command Line Client interface. A query is being executed, starting with 'explain'. The output shows two rows in the result set, both of which are 'SIMPLE' type and use an 'ALL' type join. The first row has a key of 'NULL' and a ref of 'NULL', while the second row has a key of 'Department\_ID' and a ref of 'NULL'. Both rows have a 'rows' value of 5 and a 'filtered' value of 100.00. The 'Extra' column indicates 'Using where; Using join buffer (hash join)' for both rows.

FIGURE 50:BEFORE TUNING LEFT OUTER JOIN

The screenshot shows the MySQL Workbench interface. In the central pane, there is a SQL editor window containing the following code:

```

-- AFTER tuning
CREATE INDEX idx_employee_Dip_ID2 ON EMPLOYEE (Department_ID);
CREATE INDEX idx_employee_fname ON EMPLOYEE (First_Name);
CREATE INDEX idx_employee_fnam2 ON EMPLOYEE (Last_Name);

explain
SELECT D.Department_Name,E.First_Name, E.Last_Name
FROM DEPARTMENT as D
LEFT JOIN EMPLOYEE as E ON E.Department_ID = D.Department_ID;

```

Below the SQL editor is a 'Result Grid' pane displaying the explain plan. The results are identical to Figure 50, showing two rows with 'SIMPLE' type and 'ALL' type joins, both using 'NULL' keys and refs, and both having a 'rows' value of 5 and a 'filtered' value of 100.00. The 'Extra' column again indicates 'Using where; Using join buffer (hash join)'.

FIGURE 51:AFTER TUNING LEFT OUTER JOIN

```

MySQL> explain
--> SELECT D.Department_Name, E.First_Name, IFNULL(E.Position, 'No position') AS Position
--> FROM DEPARTMENT D
--> RIGHT JOIN EMPLOYEE E ON D.Department_ID = E.Department_ID;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id  | select_type | table | partitions | type  | possible_keys | key   | key_len | ref      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1   | SIMPLE     | E    | NULL       | ALL   | NULL          | NULL  | NULL    | NULL     |
| 1   | SIMPLE     | D    | NULL       | eq_ref | PRIMARY       | PRIMARY | 42     | garment_database_system.E.Department_ID |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>

```

The screenshot shows the MySQL Command Line Client interface. The command `explain` is run on a query involving the `DEPARTMENT` and `EMPLOYEE` tables. The output shows two rows in the result set, one warning, and a duration of 0.00 seconds. The execution plan indicates a simple scan of both tables.

FIGURE 52:BEFORE TUNING RIGHT JOIN

```

651 -- after tuning
652 • CREATE INDEX idx_employee_Position ON EMPLOYEE (Position);
653
654 • explain
655 SELECT D.Department_Name, E.First_Name, IFNULL(E.Position, 'No position') AS Position
656 FROM DEPARTMENT D
657 RIGHT JOIN EMPLOYEE E ON D.Department_ID = E.Department_ID;

```

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	SIMPLE	E	NULL	ALL	NULL	NULL	NULL	NULL	5	100.00	NULL
1	SIMPLE	D	NULL	eq_ref	PRIMARY	PRIMARY	42	garment_database_system.E.Department_ID	1	100.00	NULL

The screenshot shows the MySQL Workbench interface. A new index `idx\_employee\_Position` is created on the `EMPLOYEE` table. The `explain` command is run on the same query as in Figure 52. The resulting execution plan shows the same two rows and one warning, but the duration is now 2.51 PM, indicating a significant performance improvement.

FIGURE 53:AFTER TUNING RIGHT JOIN

```

MySQL> explain SELECT CO.Order_ID AS CO_Order_ID, CO.Customer_ID AS CO_Customer_ID, CO.Employee_ID AS CO_Employee_ID, CO.Order_Date AS CO_Order_Date, CO.Delivery_Date AS CO_Delivery_Date, CO.Order_Status AS CO_Order_Status, CO.Total_Amount AS CO_Total_Amount,
    >     C.First_Name AS C_First_Name, C.Last_Name AS C_Last_Name, C.Street_Name AS C_Street_Name, C.City AS C_City, C.State AS C_State, C.PostalCode AS C_PostalCode, C.Country AS C_Country, C.Email AS C_Email
    > FROM Customer_Order CO
    > LEFT JOIN CUSTOMER C ON CO.Customer_ID = C.Customer_ID
    >
    > UNION ALL
    > SELECT CO.Order_ID AS CO_Order_ID, CO.Customer_ID AS CO_Customer_ID, CO.Employee_ID AS CO_Employee_ID, CO.Order_Date AS CO_Order_Date, CO.Delivery_Date AS CO_Delivery_Date, CO.Order_Status AS CO_Order_Status, CO.Total_Amount AS CO_Total_Amount,
    >     C.First_Name AS C_First_Name, C.Last_Name AS C_Last_Name, C.Street_Name AS C_Street_Name, C.City AS C_City, C.State AS C_State, C.PostalCode AS C_PostalCode, C.Country AS C_Country, C.Email AS C_Email
    > FROM Customer_Order CO
    > RIGHT JOIN CUSTOMER C ON CO.Customer_ID = C.Customer_ID;
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | CO | NULL | ALL | NULL | NULL | NULL | NULL |
| 1 | PRIMARY | C | NULL | eq_ref | PRIMARY | PRIMARY | 42 | garment_database_system.CO.Customer_ID |
| 2 | UNION | C | NULL | ALL | NULL | NULL | NULL | NULL |
| 2 | UNION | CO | NULL | ref | Customer_ID | Customer_ID | 43 | garment_database_system.C.Customer_ID |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set, 1 warning (0.00 sec)

mysql>

```

System tray icons: USD/JPY -0.97%, Search, File Explorer, Task View, Edge, Firefox, File, Print, Power, Battery, ENG INTEL, 2:08 PM, 9/11/2023.

FIGURE 54: BEFORE TUNING FULL OUTER JOIN

```

CREATE INDEX idx_CUSTOMER_Customer_ID ON CUSTOMER(Customer_ID);
CREATE INDEX idx_Customer_Order_Customer_ID ON Customer_Order(Customer_ID);
CREATE INDEX idx_Customer_Order_Order_ID ON Customer_Order(Order_ID);
CREATE INDEX idx_Customer_Order_Employee_ID ON Customer_Order(Employee_ID);
explain SELECT CO.Order_ID AS CO_Order_ID, CO.Customer_ID AS CO_Customer_ID, CO.Employee_ID AS CO_Employee_ID, CO.Order_Date AS CO_Order_Date, CO.Delivery_Date AS CO_Delivery_Date, CO.Order_Status AS CO_Order_Status, CO.Total_Amount AS CO_Total_Amount,
    >     C.First_Name AS C_First_Name, C.Last_Name AS C_Last_Name, C.Street_Name AS C_Street_Name, C.City AS C_City, C.State AS C_State, C.PostalCode AS C_PostalCode, C.Country AS C_Country, C.Email AS C_Email
    > FROM Customer_Order CO
    > LEFT JOIN CUSTOMER C ON CO.Customer_ID = C.Customer_ID
    >
    > UNION ALL
    > SELECT CO.Order_ID AS CO_Order_ID, CO.Customer_ID AS CO_Customer_ID, CO.Employee_ID AS CO_Employee_ID, CO.Order_Date AS CO_Order_Date, CO.Delivery_Date AS CO_Delivery_Date, CO.Order_Status AS CO_Order_Status, CO.Total_Amount AS CO_Total_Amount,
    >     C.First_Name AS C_First_Name, C.Last_Name AS C_Last_Name, C.Street_Name AS C_Street_Name, C.City AS C_City, C.State AS C_State, C.PostalCode AS C_PostalCode, C.Country AS C_Country, C.Email AS C_Email
    > FROM Customer_Order CO
    > RIGHT JOIN CUSTOMER C ON CO.Customer_ID = C.Customer_ID;

```

Result Grid:

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Len	Ref
1	PRIMARY	CO	NULL	ALL	NULL	NULL	NULL	NULL
1	PRIMARY	C	NULL	eq_ref	PRIMARY, idx_CUSTOMER_Customer_ID	PRIMARY	42	garment_database_system.CO.Customer_ID
2	UNION	C	NULL	ALL	NULL	NULL	NULL	NULL
2	UNION	CO	NULL	ref	idx_Customer_Order_Customer_ID2, idx_Customer_Order_Order_ID2	idx_Customer_Order_Customer_ID2	43	garment_database_system.C.Customer_ID

Object Info: Session

System tray icons: SENSEX +0.60%, Search, File Explorer, Task View, Edge, Firefox, File, Print, Power, Battery, ENG INTEL, 2:53 PM, 9/11/2023.

FIGURE 55: AFTER TUNNING FULL OUTER JOIN