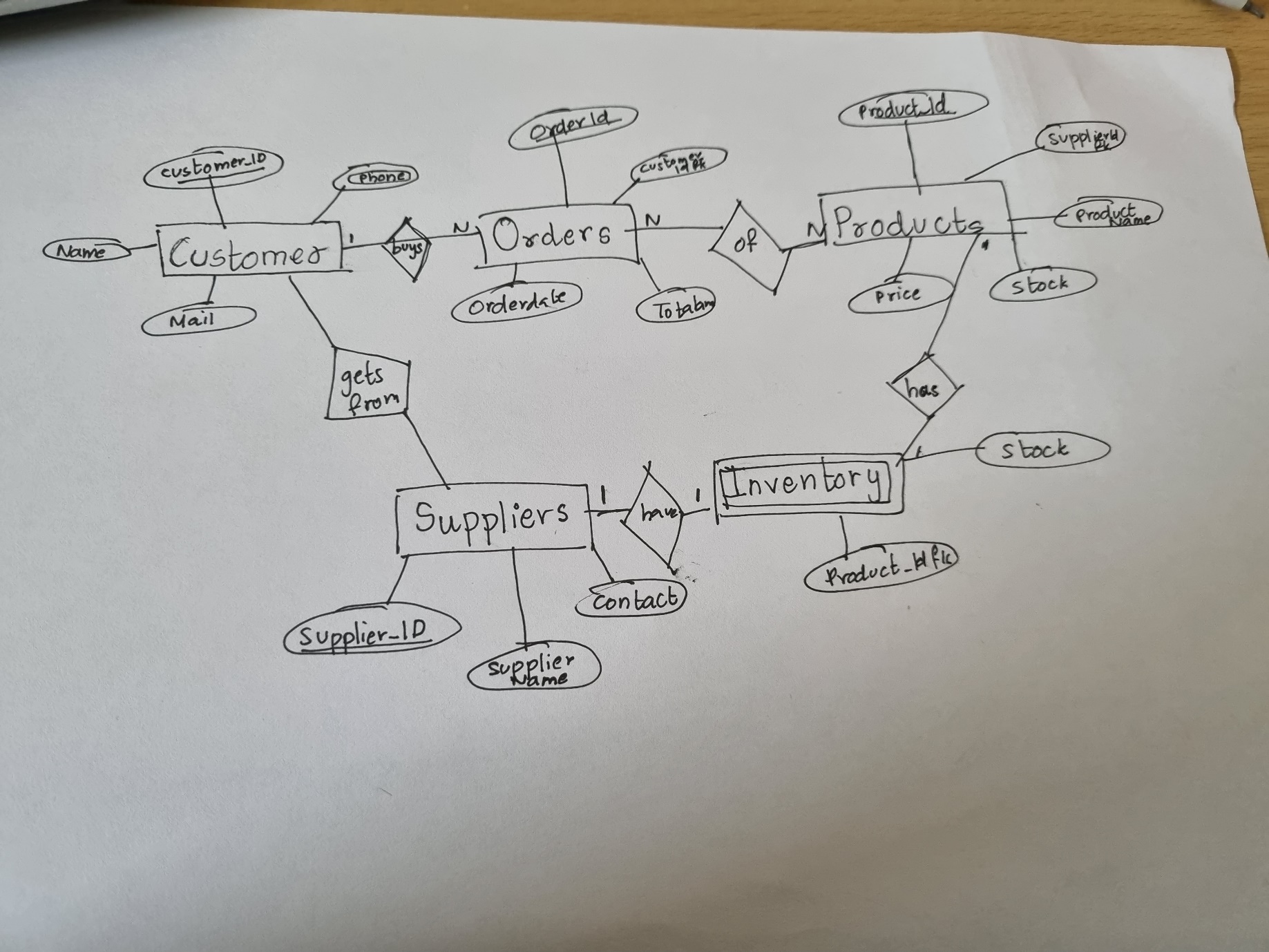
**CIA-3**

****

The textile industry database is designed to manage and streamline operations related to fabric production, distribution, and sales. It consists of five key tables:

1. Customers: Stores information about customers, including their contact details.

2. Suppliers: Contains details about suppliers providing raw materials and fabrics.

3. Products: Lists various textile products, linking them to suppliers, along with pricing and stock information.

4. Orders: Records customer orders, including order dates and total amounts.

5. Inventory: Manages stock levels for each product, ensuring effective supply chain management.

mysql> use textile;

Database changed

mysql> CREATE TABLE Customers (

-> CustomerID INT PRIMARY KEY,

-> Name VARCHAR(100),

-> Email VARCHAR(100),

-> Phone VARCHAR(15)

-> );

Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE Suppliers (

-> SupplierID INT PRIMARY KEY,

-> SupplierName VARCHAR(100),

-> ContactInfo VARCHAR(100)

-> );

Query OK, 0 rows affected (0.06 sec)

mysql> CREATE TABLE Products (

-> ProductID INT PRIMARY KEY,

-> SupplierID INT,

-> ProductName VARCHAR(100),

-> Price DECIMAL(10, 2),

-> Stock INT,

-> FOREIGN KEY (SupplierID) REFERENCES Suppliers(SupplierID)

-> );

Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE Orders (

-> OrderID INT PRIMARY KEY,

-> CustomerID INT,

-> OrderDate DATE,

-> TotalAmount DECIMAL(10, 2),

-> FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

-> );

Query OK, 0 rows affected (0.07 sec)

mysql> CREATE TABLE Inventory (

-> InventoryID INT PRIMARY KEY,

-> ProductID INT,

-> Stock INT,

-> FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

-> );

Query OK, 0 rows affected (0.08 sec)

mysql> INSERT INTO Customers (CustomerID, Name, Email, Phone) VALUES

-> (1, 'John Doe', 'john@example.com', '1234567890'),

-> (2, 'Jane Smith', 'jane@example.com', '0987654321'),

-> (3, 'Alice Johnson', 'alice@example.com', '1112223333'),

-> (4, 'Bob Williams', 'bob@example.com', '4445556666'),

-> (5, 'Eve Adams', 'eve@example.com', '7778889999');

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

mysql> INSERT INTO Suppliers (SupplierID, SupplierName, ContactInfo) VALUES

-> (1, 'Textile Corp', 'contact@textilecorp.com'),

-> (2, 'Fabric Suppliers', 'contact@fabricsuppliers.com'),

-> (3, 'Premium Textiles', 'premium@textiles.com'),

-> (4, 'Global Fabrics', 'info@globalfabrics.com'),

-> (5, 'Silk Masters', 'support@silkmasters.com');

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

mysql> INSERT INTO Products (ProductID, SupplierID, ProductName, Price, Stock) VALUES

-> (1, 1, 'Cotton Fabric', 10.50, 100),

-> (2, 2, 'Silk Fabric', 25.00, 50),

-> (3, 3, 'Wool Fabric', 15.75, 70),

-> (4, 4, 'Polyester Fabric', 8.99, 200),

-> (5, 5, 'Linen Fabric', 18.50, 120);

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

mysql> INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES

-> (1, 1, '2024-10-01', 105.00),

-> (2, 2, '2024-10-02', 50.00),

-> (3, 3, '2024-10-03', 85.00),

-> (4, 4, '2024-10-04', 150.00),

-> (5, 5, '2024-10-05', 230.00);

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

mysql> INSERT INTO Inventory (InventoryID, ProductID, Stock) VALUES

-> (1, 1, 100),

-> (2, 2, 50),

-> (3, 3, 70),

-> (4, 4, 200),

-> (5, 5, 120);

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE Customers ADD CONSTRAINT UC\_Email UNIQUE (Email);

Query OK, 0 rows affected (0.10 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> CREATE TABLE Payments (

-> PaymentID INT PRIMARY KEY,

-> OrderID INT,

-> PaymentDate DATE,

-> Amount DECIMAL(10, 2),

-> FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),

-> CHECK (Amount > 0) -- Ensure that the payment amount is positive

-> );

Query OK, 0 rows affected (0.08 sec)

mysql> INSERT INTO Payments (PaymentID, OrderID, PaymentDate, Amount) VALUES

-> (1, 1, '2024-10-02', 105.00),

-> (2, 2, '2024-10-03', 50.00),

-> (3, 3, '2024-10-04', 85.00),

-> (4, 4, '2024-10-05', 150.00),

-> (5, 5, '2024-10-06', 230.00);

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

**Clauses**

mysql> SELECT \* FROM Products

-> WHERE Price > 15.00;

+-----------+------------+--------------+-------+-------+

| ProductID | SupplierID | ProductName | Price | Stock |

+-----------+------------+--------------+-------+-------+

| 2 | 2 | Silk Fabric | 25.00 | 50 |

| 3 | 3 | Wool Fabric | 15.75 | 70 |

| 5 | 5 | Linen Fabric | 18.50 | 120 |

+-----------+------------+--------------+-------+-------+

3 rows in set (0.02 sec)

mysql> SELECT SupplierID, COUNT(\*) AS ProductCount

-> FROM Products

-> GROUP BY SupplierID;

+------------+--------------+

| SupplierID | ProductCount |

+------------+--------------+

| 1 | 1 |

| 2 | 1 |

| 3 | 1 |

| 4 | 1 |

| 5 | 1 |

+------------+--------------+

5 rows in set (0.02 sec)

mysql> SELECT SupplierID, COUNT(\*) AS ProductCount

-> FROM Products

-> GROUP BY SupplierID;

+------------+--------------+

| SupplierID | ProductCount |

+------------+--------------+

| 1 | 1 |

| 2 | 1 |

| 3 | 1 |

| 4 | 1 |

| 5 | 1 |

+------------+--------------+

5 rows in set (0.00 sec)

mysql> SELECT SupplierID, COUNT(\*) AS ProductCount

-> FROM Products

-> GROUP BY SupplierID

-> HAVING ProductCount > 2;

Empty set (0.00 sec)

mysql> SELECT \* FROM Customers

-> ORDER BY Name ASC;

+------------+---------------+-------------------+------------+

| CustomerID | Name | Email | Phone |

+------------+---------------+-------------------+------------+

| 3 | Alice Johnson | alice@example.com | 1112223333 |

| 4 | Bob Williams | bob@example.com | 4445556666 |

| 5 | Eve Adams | eve@example.com | 7778889999 |

| 2 | Jane Smith | jane@example.com | 0987654321 |

| 1 | John Doe | john@example.com | 1234567890 |

+------------+---------------+-------------------+------------+

5 rows in set (0.01 sec)

mysql> SELECT DISTINCT ProductName

-> FROM Products;

+------------------+

| ProductName |

+------------------+

| Cotton Fabric |

| Silk Fabric |

| Wool Fabric |

| Polyester Fabric |

| Linen Fabric |

+------------------+

5 rows in set (0.01 sec)

mysql> SELECT \* FROM Customers

-> WHERE Name LIKE 'J%';

+------------+------------+------------------+------------+

| CustomerID | Name | Email | Phone |

+------------+------------+------------------+------------+

| 1 | John Doe | john@example.com | 1234567890 |

| 2 | Jane Smith | jane@example.com | 0987654321 |

+------------+------------+------------------+------------+

2 rows in set (0.01 sec)

JOINS

mysql> SELECT p.ProductName, s.SupplierName

-> FROM Products p

-> INNER JOIN Suppliers s ON p.SupplierID = s.SupplierID;

+------------------+------------------+

| ProductName | SupplierName |

+------------------+------------------+

| Cotton Fabric | Textile Corp |

| Silk Fabric | Fabric Suppliers |

| Wool Fabric | Premium Textiles |

| Polyester Fabric | Global Fabrics |

| Linen Fabric | Silk Masters |

+------------------+------------------+

5 rows in set (0.00 sec)

mysql> SELECT p.ProductName, s.SupplierName

-> FROM Products p

-> LEFT JOIN Suppliers s ON p.SupplierID = s.SupplierID;

+------------------+------------------+

| ProductName | SupplierName |

+------------------+------------------+

| Cotton Fabric | Textile Corp |

| Silk Fabric | Fabric Suppliers |

| Wool Fabric | Premium Textiles |

| Polyester Fabric | Global Fabrics |

| Linen Fabric | Silk Masters |

+------------------+------------------+

5 rows in set (0.00 sec)

mysql> SELECT s.SupplierName, p.ProductName

-> FROM Suppliers s

-> RIGHT JOIN Products p ON s.SupplierID = p.SupplierID;

+------------------+------------------+

| SupplierName | ProductName |

+------------------+------------------+

| Textile Corp | Cotton Fabric |

| Fabric Suppliers | Silk Fabric |

| Premium Textiles | Wool Fabric |

| Global Fabrics | Polyester Fabric |

| Silk Masters | Linen Fabric |

+------------------+------------------+

5 rows in set (0.00 sec)

mysql> SELECT o.OrderID, c.Name AS CustomerName, p.ProductName, o.TotalAmount

-> FROM Orders o

-> INNER JOIN Customers c ON o.CustomerID = c.CustomerID

-> INNER JOIN Products p ON p.ProductID = o.OrderID; -- Note: Adjusted this part for meaningful join

+---------+---------------+------------------+-------------+

| OrderID | CustomerName | ProductName | TotalAmount |

+---------+---------------+------------------+-------------+

| 1 | John Doe | Cotton Fabric | 105.00 |

| 2 | Jane Smith | Silk Fabric | 50.00 |

| 3 | Alice Johnson | Wool Fabric | 85.00 |

| 4 | Bob Williams | Polyester Fabric | 150.00 |

| 5 | Eve Adams | Linen Fabric | 230.00 |

+---------+---------------+------------------+-------------+

5 rows in set (0.01 sec)

mysql> SELECT o.OrderID, c.Name AS CustomerName, p.ProductName

-> FROM Orders o

-> LEFT JOIN Customers c ON o.CustomerID = c.CustomerID

-> LEFT JOIN Products p ON o.OrderID = p.ProductID; -- Again, should relate order to products

+---------+---------------+------------------+

| OrderID | CustomerName | ProductName |

+---------+---------------+------------------+

| 1 | John Doe | Cotton Fabric |

| 2 | Jane Smith | Silk Fabric |

| 3 | Alice Johnson | Wool Fabric |

| 4 | Bob Williams | Polyester Fabric |

| 5 | Eve Adams | Linen Fabric |

+---------+---------------+------------------+

5 rows in set (0.00 sec)

mysql> SELECT c.Name AS CustomerName, o.OrderID

-> FROM Customers c

-> RIGHT JOIN Orders o ON c.CustomerID = o.CustomerID;

+---------------+---------+

| CustomerName | OrderID |

+---------------+---------+

| John Doe | 1 |

| Jane Smith | 2 |

| Alice Johnson | 3 |

| Bob Williams | 4 |

| Eve Adams | 5 |

+---------------+---------+

5 rows in set (0.00 sec)

mysql> CREATE TABLE OrderDetails (

-> DetailID INT PRIMARY KEY,

-> OrderID INT,

-> ProductID INT,

-> Quantity INT,

-> FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),

-> FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

-> );

Query OK, 0 rows affected (0.09 sec)

mysql> INSERT INTO OrderDetails (DetailID, OrderID, ProductID, Quantity) VALUES

-> (1, 1, 1, 5),

-> (2, 1, 2, 3),

-> (3, 2, 3, 2),

-> (4, 3, 4, 1),

-> (5, 4, 5, 4);

Query OK, 5 rows affected (0.02 sec)

Records: 5 Duplicates: 0 Warnings: 0

mysql> SELECT od.DetailID, c.Name AS CustomerName, p.ProductName, od.Quantity

-> FROM OrderDetails od

-> INNER JOIN Orders o ON od.OrderID = o.OrderID

-> INNER JOIN Customers c ON o.CustomerID = c.CustomerID

-> INNER JOIN Products p ON od.ProductID = p.ProductID;

+----------+---------------+------------------+----------+

| DetailID | CustomerName | ProductName | Quantity |

+----------+---------------+------------------+----------+

| 1 | John Doe | Cotton Fabric | 5 |

| 2 | John Doe | Silk Fabric | 3 |

| 3 | Jane Smith | Wool Fabric | 2 |

| 4 | Alice Johnson | Polyester Fabric | 1 |

| 5 | Bob Williams | Linen Fabric | 4 |

+----------+---------------+------------------+----------+

5 rows in set (0.01 sec)

mysql> SELECT od.DetailID, c.Name AS CustomerName, p.ProductName, od.Quantity

-> FROM OrderDetails od

-> LEFT JOIN Orders o ON od.OrderID = o.OrderID

-> LEFT JOIN Customers c ON o.CustomerID = c.CustomerID

-> LEFT JOIN Products p ON od.ProductID = p.ProductID;

+----------+---------------+------------------+----------+

| DetailID | CustomerName | ProductName | Quantity |

+----------+---------------+------------------+----------+

| 1 | John Doe | Cotton Fabric | 5 |

| 2 | John Doe | Silk Fabric | 3 |

| 3 | Jane Smith | Wool Fabric | 2 |

| 4 | Alice Johnson | Polyester Fabric | 1 |

| 5 | Bob Williams | Linen Fabric | 4 |

+----------+---------------+------------------+----------+

5 rows in set (0.00 sec)

FUNCTIONS

mysql> -- String Functions

mysql> SELECT Name, UPPER(Name) AS UppercaseName

-> FROM Customers;

+---------------+---------------+

| Name | UppercaseName |

+---------------+---------------+

| John Doe | JOHN DOE |

| Jane Smith | JANE SMITH |

| Alice Johnson | ALICE JOHNSON |

| Bob Williams | BOB WILLIAMS |

| Eve Adams | EVE ADAMS |

+---------------+---------------+

5 rows in set (0.01 sec)

mysql>

mysql> SELECT Name, CONCAT(Name, ' Doe') AS FullName

-> FROM Customers;

+---------------+-------------------+

| Name | FullName |

+---------------+-------------------+

| John Doe | John Doe Doe |

| Jane Smith | Jane Smith Doe |

| Alice Johnson | Alice Johnson Doe |

| Bob Williams | Bob Williams Doe |

| Eve Adams | Eve Adams Doe |

+---------------+-------------------+

5 rows in set (0.00 sec)

mysql>

mysql> SELECT ProductName, SUBSTRING(ProductName, 1, 3) AS ShortName

-> FROM Products;

+------------------+-----------+

| ProductName | ShortName |

+------------------+-----------+

| Cotton Fabric | Cot |

| Silk Fabric | Sil |

| Wool Fabric | Woo |

| Polyester Fabric | Pol |

| Linen Fabric | Lin |

+------------------+-----------+

5 rows in set (0.00 sec)

mysql>

mysql> -- Numeric Functions

mysql> SELECT ProductName, Price, ROUND(Price) AS RoundedPrice

-> FROM Products;

+------------------+-------+--------------+

| ProductName | Price | RoundedPrice |

+------------------+-------+--------------+

| Cotton Fabric | 10.50 | 11 |

| Silk Fabric | 25.00 | 25 |

| Wool Fabric | 15.75 | 16 |

| Polyester Fabric | 8.99 | 9 |

| Linen Fabric | 18.50 | 19 |

+------------------+-------+--------------+

5 rows in set (0.01 sec)

mysql>

mysql> SELECT AVG(Price) AS AveragePrice

-> FROM Products;

+--------------+

| AveragePrice |

+--------------+

| 15.748000 |

+--------------+

1 row in set (0.01 sec)

mysql>

mysql> SELECT SUM(Stock) AS TotalStock

-> FROM Products;

+------------+

| TotalStock |

+------------+

| 540 |

+------------+

1 row in set (0.00 sec)

mysql>

mysql> -- Date-Time Functions

mysql> SELECT CURDATE() AS CurrentDate;

+-------------+

| CurrentDate |

+-------------+

| 2024-10-16 |

+-------------+

1 row in set (0.00 sec)

mysql>

mysql> SELECT OrderID, OrderDate, DATEDIFF(CURDATE(), OrderDate) AS DaysSinceOrder

-> FROM Orders;

+---------+------------+----------------+

| OrderID | OrderDate | DaysSinceOrder |

+---------+------------+----------------+

| 1 | 2024-10-01 | 15 |

| 2 | 2024-10-02 | 14 |

| 3 | 2024-10-03 | 13 |

| 4 | 2024-10-04 | 12 |

| 5 | 2024-10-05 | 11 |

+---------+------------+----------------+

5 rows in set (0.00 sec)

mysql>

mysql> SELECT OrderID, DATE\_FORMAT(OrderDate, '%d-%m-%Y') AS FormattedOrderDate

-> FROM Orders;

+---------+--------------------+

| OrderID | FormattedOrderDate |

+---------+--------------------+

| 1 | 01-10-2024 |

| 2 | 02-10-2024 |

| 3 | 03-10-2024 |

| 4 | 04-10-2024 |

| 5 | 05-10-2024 |

+---------+--------------------+

5 rows in set (0.00 sec)

mysql>

mysql> SELECT c.Name AS CustomerName,

-> (SELECT COUNT(od.Quantity)

-> FROM Orders o

-> INNER JOIN OrderDetails od ON o.OrderID = od.OrderID

-> WHERE o.CustomerID = c.CustomerID) AS TotalProductsOrdered

-> FROM Customers c;

+---------------+----------------------+

| CustomerName | TotalProductsOrdered |

+---------------+----------------------+

| John Doe | 2 |

| Jane Smith | 1 |

| Alice Johnson | 1 |

| Bob Williams | 1 |

| Eve Adams | 0 |

+---------------+----------------------+

5 rows in set (0.01 sec)

mysql> SELECT c.Name,

-> (SELECT SUM(o.TotalAmount)

-> FROM Orders o

-> WHERE o.CustomerID = c.CustomerID) AS CustomerTotal

-> FROM Customers c

-> HAVING CustomerTotal > (SELECT AVG(TotalAmount) FROM Orders);

+--------------+---------------+

| Name | CustomerTotal |

+--------------+---------------+

| Bob Williams | 150.00 |

| Eve Adams | 230.00 |

+--------------+---------------+

2 rows in set (0.00 sec)

mysql> SELECT ProductName, Stock

-> FROM Products

-> WHERE Stock > (SELECT AVG(Stock) FROM Products);

+------------------+-------+

| ProductName | Stock |

+------------------+-------+

| Polyester Fabric | 200 |

| Linen Fabric | 120 |

+------------------+-------+

2 rows in set (0.00 sec)

mysql> SELECT ProductName, Price

-> FROM Products

-> WHERE Price > ANY (SELECT Price FROM Products WHERE SupplierID = 1);

+--------------+-------+

| ProductName | Price |

+--------------+-------+

| Silk Fabric | 25.00 |

| Wool Fabric | 15.75 |

| Linen Fabric | 18.50 |

+--------------+-------+

3 rows in set (0.00 sec)

mysql> SELECT ProductName, Price

-> FROM Products

-> WHERE Price > ALL (SELECT Price FROM Products WHERE Price < 20);

+-------------+-------+

| ProductName | Price |

+-------------+-------+

| Silk Fabric | 25.00 |

+-------------+-------+

1 row in set (0.00 sec)

mysql> SELECT ProductName FROM Products

-> UNION

-> SELECT SupplierName FROM Suppliers;

+------------------+

| ProductName |

+------------------+

| Cotton Fabric |

| Silk Fabric |

| Wool Fabric |

| Polyester Fabric |

| Linen Fabric |

| Textile Corp |

| Fabric Suppliers |

| Premium Textiles |

| Global Fabrics |

| Silk Masters |

+------------------+

10 rows in set (0.00 sec)

mysql> SELECT ProductName

-> FROM Products

-> WHERE ProductName IN (SELECT SupplierName FROM Suppliers);

Empty set (0.00 sec)

mysql> SELECT ProductName

-> FROM Products

-> WHERE ProductName IN (SELECT SupplierName FROM Suppliers);

Empty set (0.00 sec)

JOINS

mysql> SELECT \*

-> FROM Customers

-> CROSS JOIN Products;

+------------+---------------+-------------------+------------+-----------+------------+------------------+-------+-------+

| CustomerID | Name | Email | Phone | ProductID | SupplierID | ProductName | Price | Stock |

+------------+---------------+-------------------+------------+-----------+------------+------------------+-------+-------+

| 5 | Eve Adams | eve@example.com | 7778889999 | 1 | 1 | Cotton Fabric | 10.50 | 100 |

| 4 | Bob Williams | bob@example.com | 4445556666 | 1 | 1 | Cotton Fabric | 10.50 | 100 |

| 3 | Alice Johnson | alice@example.com | 1112223333 | 1 | 1 | Cotton Fabric | 10.50 | 100 |

| 2 | Jane Smith | jane@example.com | 0987654321 | 1 | 1 | Cotton Fabric | 10.50 | 100 |

| 1 | John Doe | john@example.com | 1234567890 | 1 | 1 | Cotton Fabric | 10.50 | 100 |

| 5 | Eve Adams | eve@example.com | 7778889999 | 2 | 2 | Silk Fabric | 25.00 | 50 |

| 4 | Bob Williams | bob@example.com | 4445556666 | 2 | 2 | Silk Fabric | 25.00 | 50 |

| 3 | Alice Johnson | alice@example.com | 1112223333 | 2 | 2 | Silk Fabric | 25.00 | 50 |

| 2 | Jane Smith | jane@example.com | 0987654321 | 2 | 2 | Silk Fabric | 25.00 | 50 |

| 1 | John Doe | john@example.com | 1234567890 | 2 | 2 | Silk Fabric | 25.00 | 50 |

| 5 | Eve Adams | eve@example.com | 7778889999 | 3 | 3 | Wool Fabric | 15.75 | 70 |

| 4 | Bob Williams | bob@example.com | 4445556666 | 3 | 3 | Wool Fabric | 15.75 | 70 |

| 3 | Alice Johnson | alice@example.com | 1112223333 | 3 | 3 | Wool Fabric | 15.75 | 70 |

| 2 | Jane Smith | jane@example.com | 0987654321 | 3 | 3 | Wool Fabric | 15.75 | 70 |

| 1 | John Doe | john@example.com | 1234567890 | 3 | 3 | Wool Fabric | 15.75 | 70 |

| 5 | Eve Adams | eve@example.com | 7778889999 | 4 | 4 | Polyester Fabric | 8.99 | 200 |

| 4 | Bob Williams | bob@example.com | 4445556666 | 4 | 4 | Polyester Fabric | 8.99 | 200 |

| 3 | Alice Johnson | alice@example.com | 1112223333 | 4 | 4 | Polyester Fabric | 8.99 | 200 |

| 2 | Jane Smith | jane@example.com | 0987654321 | 4 | 4 | Polyester Fabric | 8.99 | 200 |

| 1 | John Doe | john@example.com | 1234567890 | 4 | 4 | Polyester Fabric | 8.99 | 200 |

| 5 | Eve Adams | eve@example.com | 7778889999 | 5 | 5 | Linen Fabric | 18.50 | 120 |

| 4 | Bob Williams | bob@example.com | 4445556666 | 5 | 5 | Linen Fabric | 18.50 | 120 |

| 3 | Alice Johnson | alice@example.com | 1112223333 | 5 | 5 | Linen Fabric | 18.50 | 120 |

| 2 | Jane Smith | jane@example.com | 0987654321 | 5 | 5 | Linen Fabric | 18.50 | 120 |

| 1 | John Doe | john@example.com | 1234567890 | 5 | 5 | Linen Fabric | 18.50 | 120 |

+------------+---------------+-------------------+------------+-----------+------------+------------------+-------+-------+

25 rows in set (0.00 sec)

mysql> SELECT \*

-> FROM Customers, Products;

+------------+---------------+-------------------+------------+-----------+------------+------------------+-------+-------+

| CustomerID | Name | Email | Phone | ProductID | SupplierID | ProductName | Price | Stock |

+------------+---------------+-------------------+------------+-----------+------------+------------------+-------+-------+

| 5 | Eve Adams | eve@example.com | 7778889999 | 1 | 1 | Cotton Fabric | 10.50 | 100 |

| 4 | Bob Williams | bob@example.com | 4445556666 | 1 | 1 | Cotton Fabric | 10.50 | 100 |

| 3 | Alice Johnson | alice@example.com | 1112223333 | 1 | 1 | Cotton Fabric | 10.50 | 100 |

| 2 | Jane Smith | jane@example.com | 0987654321 | 1 | 1 | Cotton Fabric | 10.50 | 100 |

| 1 | John Doe | john@example.com | 1234567890 | 1 | 1 | Cotton Fabric | 10.50 | 100 |

| 5 | Eve Adams | eve@example.com | 7778889999 | 2 | 2 | Silk Fabric | 25.00 | 50 |

| 4 | Bob Williams | bob@example.com | 4445556666 | 2 | 2 | Silk Fabric | 25.00 | 50 |

| 3 | Alice Johnson | alice@example.com | 1112223333 | 2 | 2 | Silk Fabric | 25.00 | 50 |

| 2 | Jane Smith | jane@example.com | 0987654321 | 2 | 2 | Silk Fabric | 25.00 | 50 |

| 1 | John Doe | john@example.com | 1234567890 | 2 | 2 | Silk Fabric | 25.00 | 50 |

| 5 | Eve Adams | eve@example.com | 7778889999 | 3 | 3 | Wool Fabric | 15.75 | 70 |

| 4 | Bob Williams | bob@example.com | 4445556666 | 3 | 3 | Wool Fabric | 15.75 | 70 |

| 3 | Alice Johnson | alice@example.com | 1112223333 | 3 | 3 | Wool Fabric | 15.75 | 70 |

| 2 | Jane Smith | jane@example.com | 0987654321 | 3 | 3 | Wool Fabric | 15.75 | 70 |

| 1 | John Doe | john@example.com | 1234567890 | 3 | 3 | Wool Fabric | 15.75 | 70 |

| 5 | Eve Adams | eve@example.com | 7778889999 | 4 | 4 | Polyester Fabric | 8.99 | 200 |

| 4 | Bob Williams | bob@example.com | 4445556666 | 4 | 4 | Polyester Fabric | 8.99 | 200 |

| 3 | Alice Johnson | alice@example.com | 1112223333 | 4 | 4 | Polyester Fabric | 8.99 | 200 |

| 2 | Jane Smith | jane@example.com | 0987654321 | 4 | 4 | Polyester Fabric | 8.99 | 200 |

| 1 | John Doe | john@example.com | 1234567890 | 4 | 4 | Polyester Fabric | 8.99 | 200 |

| 5 | Eve Adams | eve@example.com | 7778889999 | 5 | 5 | Linen Fabric | 18.50 | 120 |

| 4 | Bob Williams | bob@example.com | 4445556666 | 5 | 5 | Linen Fabric | 18.50 | 120 |

| 3 | Alice Johnson | alice@example.com | 1112223333 | 5 | 5 | Linen Fabric | 18.50 | 120 |

| 2 | Jane Smith | jane@example.com | 0987654321 | 5 | 5 | Linen Fabric | 18.50 | 120 |

| 1 | John Doe | john@example.com | 1234567890 | 5 | 5 | Linen Fabric | 18.50 | 120 |

+------------+---------------+-------------------+------------+-----------+------------+------------------+-------+-------+

25 rows in set (0.00 sec)

mysql> Implement all TCL commands on the tables of your chosen domain.

->

-> Start transaction

-> Auto commit (on/off)

-> Commit

-> Savepoint

-> Rollback etc;

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 'Implement all TCL commands on the tables of your chosen domain.

Start transacti' at line 1

mysql> -- Start a new transaction

mysql> START TRANSACTION;

Query OK, 0 rows affected (0.00 sec)

mysql>

mysql> -- Insert a new customer

mysql> INSERT INTO Customers (CustomerID, Name, Email, Phone) VALUES

-> (3, 'Alice Johnson', 'alice@example.com', '1112223333');

ERROR 1062 (23000): Duplicate entry '3' for key 'customers.PRIMARY'

mysql>

mysql> -- Set a savepoint

mysql> SAVEPOINT Savepoint1;

Query OK, 0 rows affected (0.00 sec)

mysql>

mysql> -- Insert a new order for the new customer

mysql> INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES

-> (3, 3, '2024-10-03', 85.00);

ERROR 1062 (23000): Duplicate entry '3' for key 'orders.PRIMARY'

mysql>

mysql> -- Intentionally cause an error (if you try to insert a duplicate OrderID)

mysql> -- INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES

mysql> -- (3, 2, '2024-10-03', 85.00); -- Uncomment to simulate error

mysql>

mysql> -- If everything is correct, commit the changes

mysql> COMMIT;

Query OK, 0 rows affected (0.00 sec)

mysql>

mysql> -- If there was an error, we would rollback to the savepoint

mysql> -- ROLLBACK TO Savepoint1;

mysql>

mysql> -- Or rollback the entire transaction

mysql> -- ROLLBACK;

mysql> -- Start a new transaction

mysql> START TRANSACTION;

Query OK, 0 rows affected (0.00 sec)

mysql>

mysql> -- Insert a new customer

mysql> INSERT INTO Customers (CustomerID, Name, Email, Phone) VALUES

-> (3, 'Alice Johnson', 'alice@example.com', '1112223333');

ERROR 1062 (23000): Duplicate entry '3' for key 'customers.PRIMARY'

mysql>

mysql> -- Set a savepoint

mysql> SAVEPOINT Savepoint1;

Query OK, 0 rows affected (0.00 sec)

mysql>

mysql> -- Insert a new order for the new customer

mysql> INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES

-> (3, 3, '2024-10-03', 85.00);

ERROR 1062 (23000): Duplicate entry '3' for key 'orders.PRIMARY'

mysql>

mysql> -- Intentionally cause an error (if you try to insert a duplicate OrderID)

mysql> -- INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES

mysql> -- (3, 2, '2024-10-03', 85.00); -- Uncomment to simulate error

mysql>

mysql> -- If everything is correct, commit the changes

mysql> COMMIT;

Query OK, 0 rows affected (0.00 sec)

mysql>

mysql> -- If there was an error, we would rollback to the savepoint

mysql> -- ROLLBACK TO Savepoint1;

mysql>

mysql> -- Or rollback the entire transaction

mysql> -- ROLLBACK;