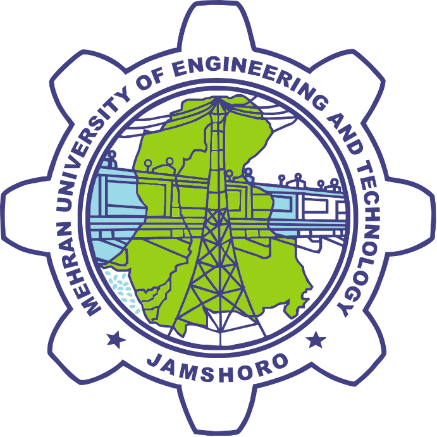
­­­

Date: 03-04-2024



**MEHRAN UNIVERSITY OF ENGINEERING AND TECHNOLOGY JAMSHORO**

**A Database Systems CEP**

**On**

PineValley Furniture Database

**Arbab Hussain Qureshi ( 22SW028 – Sec 1 )**

Submitted By

To

**Mam Rabeea Jafferi**

* **Problem Description:**

The PineValleyFurniture company requires a database to store details about its products, orders, raw materials, customers, and vendors. The entities for the required database can be identified via the business rules below:

• The company sells several different furniture products. These products are grouped into several product lines. The identifier for a product is Product ID, whereas the identifier for a product line is Product Line ID. We identify the following additional attributes for the product: Product Description, Product Finish, and Product Standard Price. Another attribute for the product line is Product Line Name. A product line may group any number of products but must group at least one product. Each product must belong to exactly one product line.

• Customers submit orders for products. The identifier for an order is Order ID, and another attribute is Order Date. A customer may submit any number of orders but need not submit any orders. Each order is submitted by exactly one customer. The identifier for a customer is Customer ID. Other attributes include Customer Name, Customer Address, and Customer Postal Code.

• A given customer order must request at least one product and only one product per order line item. Any product sold by Pine Valley Furniture may not appear on any order line item or may appear on one or more order line items. An attribute associated with each order line item is Ordered Quantity. Each product is assembled from a specified quantity of one or more raw materials. The identifier for the raw material entity is Material ID. Other attributes include Unit Of Measure, Material Name, and Material Standard Cost. Each raw material is assembled into one or more products, using a specified quantity of the raw material for each product.

• Raw materials are supplied by vendors. The identifier for a vendor is Vendor ID. Other attributes include Vendor Name and Vendor Address. Each raw material can be supplied by one or more vendors. A vendor may supply any number of raw materials or may not supply any raw materials to Pine Valley Furniture. Supply Unit Price is the unit price a particular vendor supplies a particular raw material.

You are required to create a logical entity relationship diagram (ERD) fulfilling the above requirements followed by the physical database implementation in any RDBMS of your choice. Perform the following actions on the created database.

1. Use meaningful names and datatypes during the database design.

2. Design and execute queries to create the tables. Assign keys and create the relationships between the tables.

3. Populate the database with at least 10 records including a record with your details as the customer. Create sequences wherever necessary.

4. Retrieve the order details for all customers who ordered on 24-03-2024 and delete the order details for the date 20-03-2024.

5. Create a complex view on the product, raw material, and vendor tables displaying the raw materials in each product and the corresponding vendors. Only include relevant columns from each table in the view.

Note: - **Submit the report with query output screenshots against each question along with the database file.**

After analyzing the system description, the following entities, attributes and relationships were identified.

* **Entities and Attributes**

1. **ProductLine**
   * productLineID (PK)
   * productLineName
2. **Products**
   * productID (PK)
   * productDescription
   * productFinish
   * productStandardPrice
   * productLineId(Fk referencing productLineID in ProductLine)

**3. Customers**

* + customerID (PK)
  + customerName
  + customerAddress
  + customerPostalCode

**4. Orders**

* + orderID (PK)
  + orderID
  + customerID (FK referencing customerID in Customers)

**5. OrderLine**

* + orderLineID (PK)
  + orderedQuantity
  + orderID (FK referencing orderID in Orders)
  + productID (FK referencing productID in Products)

**6. RawMaterial**

* + materialID (PK)
  + materialName
  + unitOFMeasure
  + materialStandardCost

**7. Product\_RawMaterial**

* + productID (FK referencing productID in Products)
  + materialID (FK referencing materialID in RawMaterial)
  + quantityRequired

(Composite PK productID, materialID)

**8. Vendors**

* + vendorID (PK)
  + vendorName
  + vendorAddress

**9. Vendors\_RawMaterial**

* + vendorID (FK referencing vendorID in Vendors)
  + materialID (FK referencing materialID in RawMaterial)
  + supplyUnitPrice

(Composite PK vendorID, materialID)

* **Relationships between Entities:**

**One – One Relationships:**

No any one – one relationship was identified through the CEP Description

**One – Many Relationships:**

1. **ProductLine -> Products**

One product line can have multiple products, but a product must belong to one product line only

2. **Customers -> Order**

One customer can place multiple orders, but each order belongs to a certain customer

3. **Order -> OrderLine**

One order can have multiple orderlines

4. **Products -> OrderLine**

One product can be associated with one orderline

**Many – Many Relationships:**

1. **Products <-> Products\_RawMaterial**

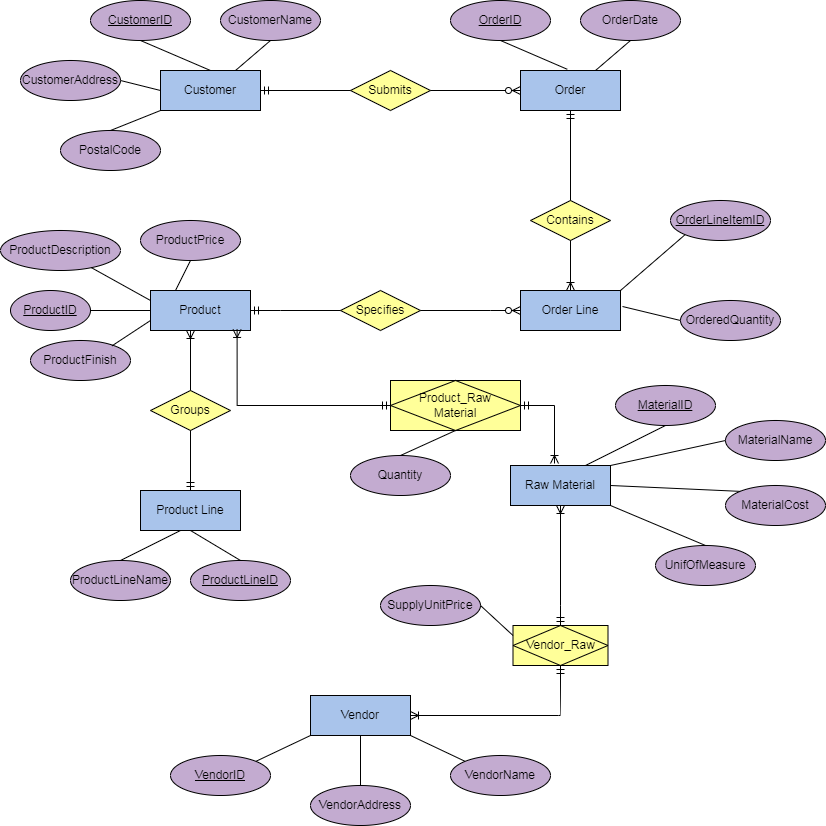
A product can be made up of multiple raw materials and one raw material can be used in multiple products.

2. **RawMaterial <-> Vendors\_RawMaterial**

One vendor can supply multiple raw materials and one raw material can be supplied by many vendors.

**ERD (Entity Relationship Diagram) Design**

After identifying the entities, attributes and relationships, The Entity Relationship Diagram was created using an online diagramming tool **draw.io**.



**Database Implementation**

MySQL is used for the implementation of the database. The CREATE statement of DDL, INSERT INTO statement of DML and SELECT statement of DQL is used to do the tasks mentioned in the CEP. All the commands and their respective outputs are given below.

**Data Definition Language (DDL) Commands**

Following are the DDL (Data Definition Language) commands used to create the database, tables and define the overall structure of the database system.

**Creation of Database:**

CREATE DATABASE IF NOT EXISTS PineValleyFurnitureDB;

use PineValleyFurnitureDB;

This will create the database and ensure all further commands to be applied on **PineValleyFurnitureDB**

**Creation of Tables:**

CREATE TABLE productLine(

productLineID INT PRIMARY KEY,

productLineName VARCHAR(50)

);

CREATE TABLE products(

productID INT PRIMARY KEY,

productDescription TEXT,

productFinish VARCHAR(50),

productStandardPrice FLOAT,

productLineID INT,

FOREIGN KEY (productLineID) REFERENCES productLine(productLineID)

);

CREATE TABLE customers(

customerID INT PRIMARY KEY,

customerName VARCHAR(50),

customerAddress TEXT,

customerPostalCode INT

);

CREATE TABLE orders(

orderID INT PRIMARY KEY,

orderDate DATE,

customerID INT,

FOREIGN KEY (customerID) REFERENCES customers(customerID)

);

CREATE TABLE orderLine(

orderLineID INT PRIMARY KEY,

orderedQuantity INT,

orderID INT,

productID INT,

FOREIGN KEY (orderID) REFERENCES orders(orderID),

FOREIGN KEY (productID) REFERENCES products(productID)

);

CREATE TABLE rawMaterial(

materialID INT PRIMARY KEY,

materialName VARCHAR(50),

unitOfMeasure VARCHAR(20),

materialStandardCost float

);

CREATE TABLE product\_rawMaterial(

productID INT,

materialID INT,

quantityRequired INT,

PRIMARY KEY(productID, materialID),

FOREIGN KEY (productID) REFERENCES products(productID),

FOREIGN KEY (materialID) REFERENCES rawMaterial(materialID)

);

CREATE TABLE vendors(

vendorID INT PRIMARY KEY,

vendorName VARCHAR(50),

vendorAddress TEXT

);

CREATE TABLE vendors\_rawMaterial(

vendorID INT,

materialID INT,

supplyUnitPrice FLOAT,

PRIMARY KEY (vendorID, materialID),

FOREIGN KEY (vendorID) REFERENCES vendors(vendorID),

FOREIGN KEY (materialID) REFERENCES rawMaterial(materialID) );

**Data Manipulation Language (DML) Commands**

The following are the DML commands that were used to insert 10 records into each of the created tables as mentioned in the CEP’s description.

INSERT INTO productLine (productLineID, productLineName)

VALUES

(1, 'Bedroom'),

(2, 'Bathroom'),

(3, 'Kitchen'),

(4, 'Office'),

(5, 'Garden'),

(6, 'Lounge'),

(7, 'Living Room'),

(8, 'Dining Room'),

(9, 'Library Furniture'),

(10, 'Hospital Furniture');



INSERT INTO products (productID, productDescription, productFinish, productStandardPrice, productLineID)

VALUES

(101, 'Sofa', 'Wood', 700.00, 6),

(102, 'Coffee Table', 'Glass', 400.00, 3),

(103, 'Bed Frame', 'Deco', 800.00, 1),

(104, 'Dining Table', 'Maple', 700.00, 8),

(105, 'Desk', 'Wood', 450.00, 4),

(106, 'Outdoor Chair', 'Aluminium', 200.00, 5),

(107, 'Toilet', 'Ceramic', 250.00, 2),

(108, 'Dining Chair', 'Wood', 140.00, 8),

(109, 'TV Stand', 'Wood', 300.00, 6),

(110, 'Bookshelf', 'Pine Wood', 170.00, 9);

A screenshot of a computer

Description automatically generated

INSERT INTO customers (customerID, customerName, customerAddress, customerPostalCode)

VALUES

(201, 'Arbab Hussain Qureshi', 'House No. B-28, Affandi Town, Hyderabad', 71000),

(202 ,'Muhammad Talha', 'House No. 361 TandoAllahyar', 70010),

(203, 'Maghan Das', 'House No. 79, Sattelite Town, MirpurKhas', 69000),

(204, 'Om Parkash', 'House No. 173, Qasimabad Hyderabad', 71000),

(205, 'Rudaba Khan', 'House No. 43, Iqbal Town Karachi', 75850),

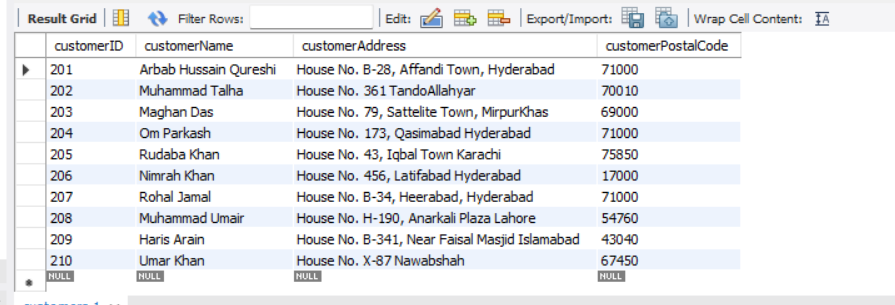
(206, 'Nimrah Khan', 'House No. 456, Latifabad Hyderabad', 17000),

(207, 'Rohal Jamal', 'House No. B-34, Heerabad, Hyderabad', 71000),

(208, 'Muhammad Umair', 'House No. H-190, Anarkali Plaza Lahore', 54760),

(209, 'Haris Arain', 'House No. B-341, Near Faisal Masjid Islamabad', 43040),

(210, 'Umar Khan', 'House No. X-87 Nawabshah', 67450);



INSERT INTO orders (orderID, orderDate, customerID)

VALUES

(301, '2024-03-16', 201),

(302, '2024-03-18', 202),

(303, '2024-03-20', 203),

(304, '2024-03-20', 204),

(305, '2024-03-20', 205),

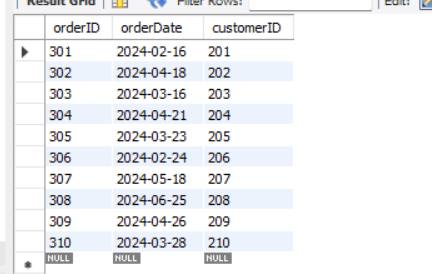
(306, '2024-03-23', 206),

(307, '2024-03-23', 207),

(308, '2024-03-23', 208),

(309, '2024-03-24', 209),

(310, '2024-03-24', 210);



INSERT INTO orderLine (orderLineID, orderedQuantity, orderID, productID)

VALUES

(401, 3, 301, 101),

(402, 1, 302, 102),

(403, 4, 303, 103),

(404, 6, 304, 104),

(405, 1, 305, 105),

(406, 3, 306, 106),

(407, 1, 307, 107),

(408, 2, 308, 108),

(409, 3, 309, 109),

(410, 3, 310, 110);



INSERT INTO rawMaterial (materialID, materialName, unitOfMeasure, materialStandardCost)

VALUES

(501, 'Wood', 'sqft', 15.00),

(502, 'Glass', 'sqft', 12.00),

(503, 'Deco', 'cubic ft', 7.00),

(504, 'Maple', 'sqft', 14.00),

(505, 'Aluminium', 'sqft', 8.00),

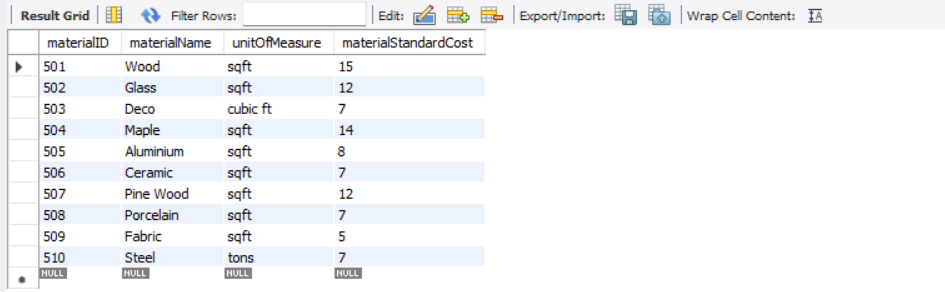
(506, 'Ceramic', 'sqft', 7.00),

(507, 'Pine Wood', 'sqft', 12.00),

(508, 'Porcelain', 'sqft', 7.00),

(509, 'Fabric', 'sqft', 5.00),

(510, 'Steel', 'tons', 7.00);



INSERT INTO product\_rawMaterial (productID, materialID, quantityRequired)

VALUES

(101, 501, 6),

(102, 502, 3),

(103, 503, 5),

(104, 504, 8),

(105, 505, 4),

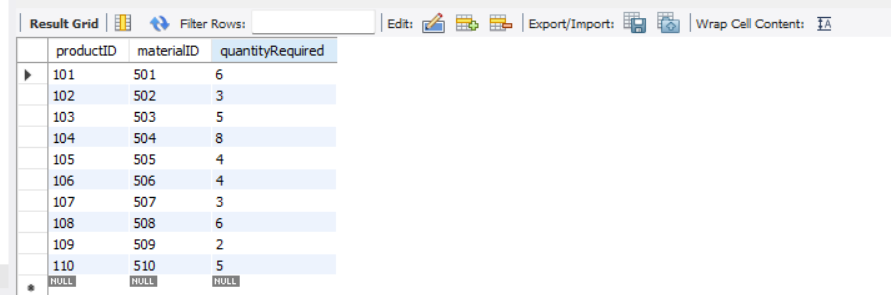
(106, 506, 4),

(107, 507, 3),

(108, 508, 6),

(109, 509, 2),

(110, 510, 5);



INSERT INTO vendors (vendorID, vendorName, vendorAddress)

VALUES

(601, 'Dream Home Furnishings', 'Hyderabad'),

(602, 'Bathing Accessories', 'Lahore'),

(603, 'Cooking Essentials', 'Multan'),

(604, 'Kids Planet', 'Sialkot'),

(605, 'Office Manufacturers', 'Hyderabad'),

(606, 'Library Zone Furnishings', 'Islamabad'),

(607, 'Dine in Furnitures', 'Lahore'),

(608, 'Emergency Equipments', 'Peshawar'),

(609, 'Room Decors', 'Islamabad'),

(610, 'Cinematic Furniture Manufacturers', 'Lahore');

A screenshot of a computer

Description automatically generated

INSERT INTO vendors\_rawMaterial (vendorID, materialID, supplyUnitPrice)

VALUES

(601, 501, 15.00),

(602, 502, 12.00),

(603, 503, 14.00),

(604, 504, 13.00),

(605, 505, 12.00),

(606, 506, 14.00),

(607, 507, 4.00),

(608, 508, 6.00),

(609, 509, 5.00),

(610, 510, 8.00);

A screenshot of a computer

Description automatically generated

**QUERY TASKS**

**1. Retrieve the order details for all customers who ordered on 24-03-2024 and delete the order details for the date 20-03-2024.**

Using join keyword to fetch details from orders, orderLine, products and customers who ordered on 24-03-2024

**Query:**

SELECT o.orderID, o.orderDate, c.customerName, p.productDescription, ol.orderedQuantity

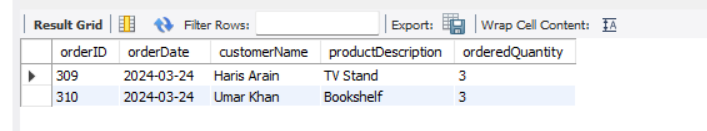
FROM orders o

JOIN customers c ON o.customerID = c.customerID

JOIN orderLine ol ON o.orderID = ol.orderID

JOIN products p ON ol.productID = p.productID

WHERE o.orderDate = '2024-03-24';

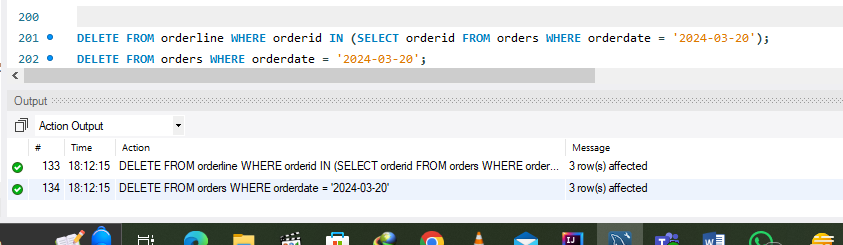


For deleting the order details for the date 20-03-2024, I have deleted both from orderLine and from orders table for maintaining referential integrity.

**Query:**

DELETE FROM orderLine WHERE orderID IN (SELECT orderID FROM orders WHERE orderDate = '2024-03-20');

DELETE FROM orders WHERE orderDate = '2024-03-20';



**2. Create a complex view on the product, raw material, and vendor tables displaying the raw materials in each product and the corresponding vendors. Only include relevant columns from each table in the view.**

Using the CREATE VIEW statement to create the view and join keyword to fetch data from multiple tables, then using select statement to see the output of the view

CREATE VIEW Vendor\_RawMaterial\_Product AS

SELECT p.productID, p.ProductDescription AS 'Product\'s Name', pl.ProductLineName, rmt.MaterialName, v.VendorName, v.vendorAddress

FROM products p

JOIN

productLine pl ON p.ProductLineID = pl.ProductLineID

JOIN

product\_rawMaterial prm ON p.ProductID = prm.ProductID

JOIN

rawMaterial rmt ON prm.MaterialID = rmt.MaterialID

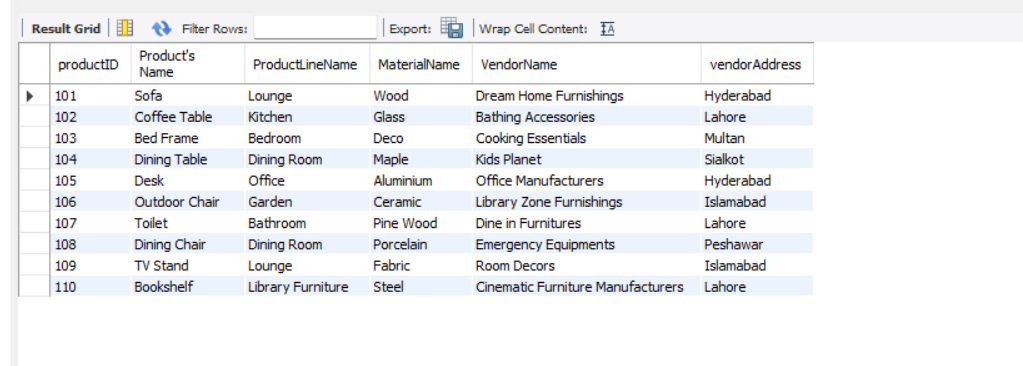
JOIN

vendors\_rawMaterial vrm ON rmt.MaterialID = vrm.MaterialID

JOIN

vendors v ON vrm.VendorID = v.VendorID;

SELECT \* FROM vendor\_rawMaterial\_Product;



**THE END**