Roll No	***22SW040 --> section: 01***
Subject	***DBS***
Project Report + Project	***10 marks***
Teacher	***Ma'am Rabeea***

Problem Description

The Pine Valley Furniture 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.

- Use meaningful names and datatypes during the database design.
- Design and execute queries to create the tables. Assign keys and create the relationships between the tables.

- Populate the database with at least 10 records including a record with your details as the customer. Create sequences wherever necessary.
- Retrieve the order details for all customers who ordered on 24-03-2024 and delete the order details for the date 20-03-2024.
- 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.

My Proposed Solution

- Creating a Database for Pine Valley Furniture will help them in many aspects of their business.

1.Keep Products Organized: The database helps Pine Valley Furniture keep track of all their different furniture products and group them neatly into categories.

2. Manage Orders Smoothly: Whenever customers place orders, the database makes sure all the details are captured accurately, making it easier to process and manage orders.

3. Track Raw Materials: It helps Pine Valley Furniture keep tabs on all the materials they use to make their furniture, where they come from, and how much they cost.

4. Know Your Customers: The database stores information about customers, like their names and addresses, so Pine Valley Furniture can provide better service and keep them happy.

5. Work Well with Vendors: Pine Valley Furniture can use the database to manage their relationships with vendors, making sure they get the materials they need on time.

6. Make Smart Decisions: By analyzing the data in the database, Pine Valley Furniture can make better decisions about what products to sell, how much to charge, and how to grow their business.

Aims and Objectives of the Proposed Solution

Aims

1. Organize Products: Arrange furniture products into categories for easy management.

2. Efficient Order Processing: Simplify the process of handling customer orders.

3. Effective Inventory Management: Keep track of raw materials used for furniture production.

4. Enhanced Customer Service: Provide better service by understanding customer needs.

5. Streamlined Supplier Relations: Maintain good relationships with material suppliers.

6. Informed Decision Making: Use data to make smart business decisions.

Objectives

1. Categorize Products: Group furniture items into product lines.

2. Capture Order Details: Record customer orders accurately.

3. Monitor Raw Material Usage: Track materials used in furniture production.

4.Store Customer Information: Keep records of customer names and addresses.

5. Manage Supplier Information: Maintain a database of material suppliers.

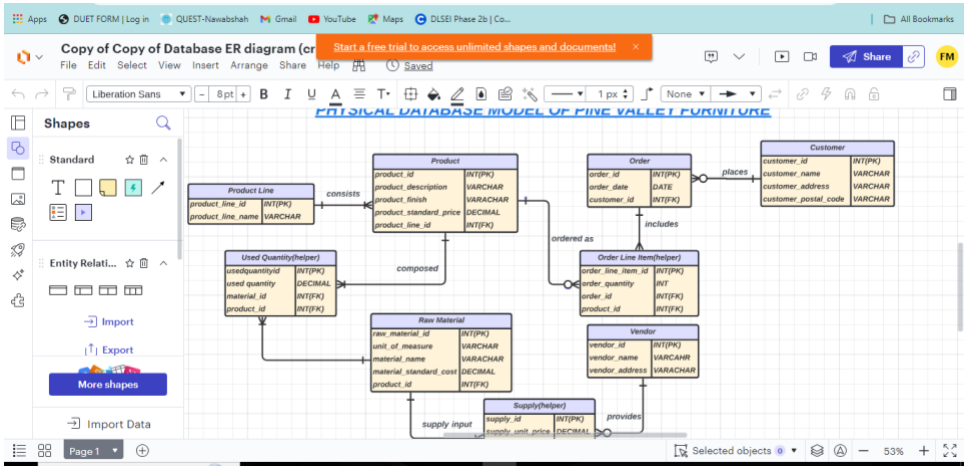
6. Analyze Data: Use database insights to guide business strategies.

Tools Required

Lucid Chart: Lucid Chart is an essential tool for developing conceptual, logical, and physical models for my project. It offers a user-friendly interface with a wide range of diagramming features, allowing me to visually represent

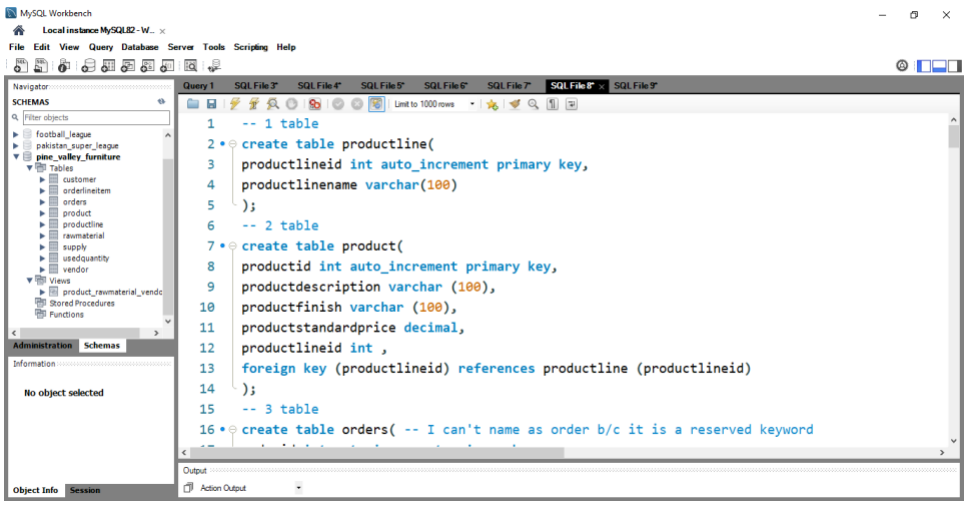
my database schema, entity-relationship diagrams, and more. With Lucid Chart, I can collaborate with team members in real-time, making it ideal for designing complex database structures.

See the Screen Shot



MySQL Workbench and Server: MySQL Workbench and server are indispensable for developing my project by writing queries and code. It provides a comprehensive set of tools for database design, development, and administration. With these tools, I can create and manage database schemas, design tables, write SQL queries, and debug code efficiently. Its intuitive interface makes it suitable for both beginners and experienced developers alike.

See the Screen Shot



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.

- **Use meaningful names and datatypes during the database design.**
- **Design and execute queries to create the tables. Assign keys and create the relationships between the tables.**

All these has done by me see below

Specification and Design

Table names and attributes

Customer

- customer_id (INT, PK)

- customer_name (VARCHAR)

- customer_address (VARCHAR)

- customer_postal_code (VARCHAR)

Order

- order_id (INT, PK)

- customer_id (INT, FK references Customer.customer_id)

- order_date (DATE)

Product

- product_id (INT, PK)

- product_line_id (INT, FK references Product Line.product_line_id)

- product_description (VARCHAR)

- product_finish (VARCHAR)

- product_standard_price (DECIMAL)

Product Line

- product_line_id (INT, PK)

- product_line_name (VARCHAR)

Order Line Item (helper table)

- order_line_item_id (INT, PK)

- order_id (INT, FK references Order.order_id)

- product_id (INT, FK references Product.product_id)

- order_quantity (INT)

Used Quantity (helper table)

- used_quantity_id (INT, PK)

- order_line_item_id (INT, FK references Order Line Item.order_line_item_id)

- product_id (INT, FK references Product.product_id)

- material_id (INT, FK references Raw Material.raw_material_id)

- used_quantity (DECIMAL)

Raw Material

- raw_material_id (INT, PK)

- material_name (VARCHAR)

- unit_of_measure (VARCHAR)

- material_standard_cost (DECIMAL)

Vendor

- vendor_id (INT, PK)

- vendor_name (VARCHAR)

- vendor_address (VARCHAR)

Supply (helper table)

- supply_id (INT, PK)

- vendor_id (INT, FK references Vendor.vendor_id)

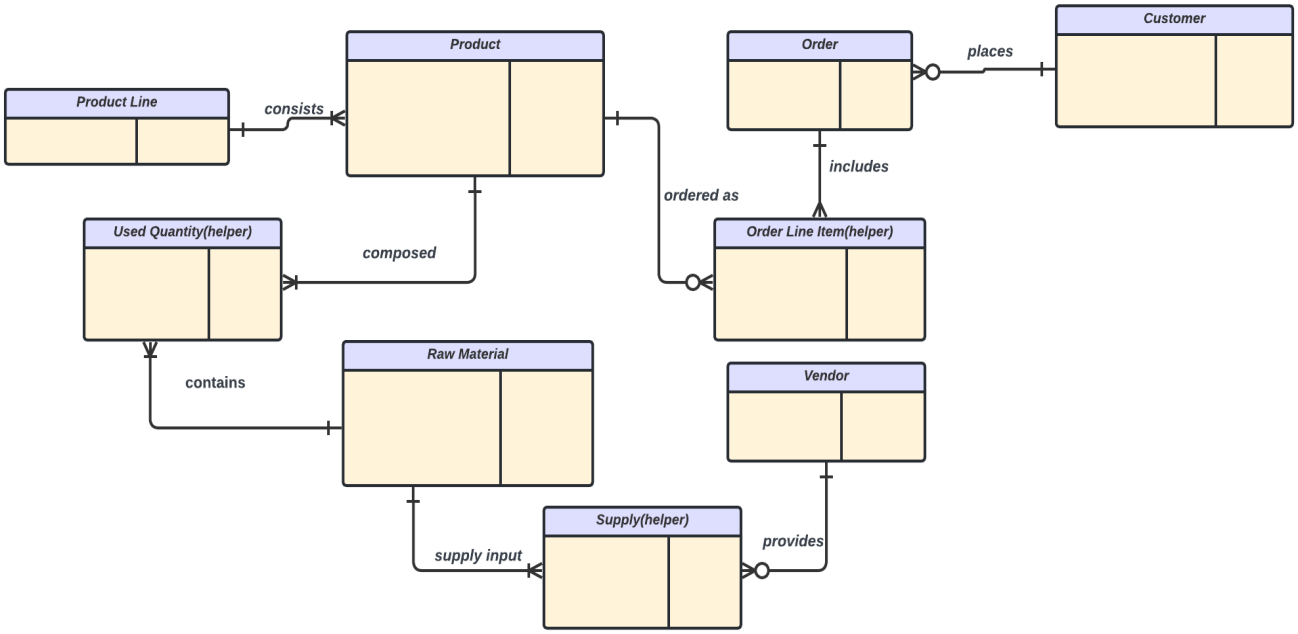
- material_id (INT, FK references Raw Material.raw_material_id)

- supply_unit_price (DECIMAL)

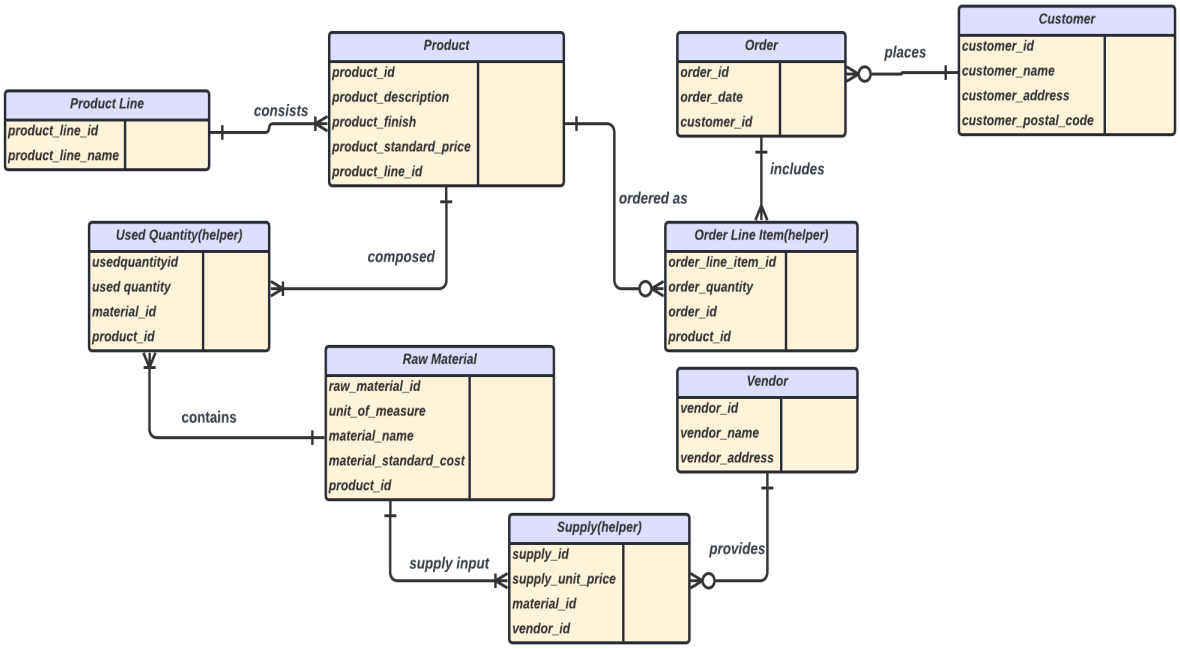
Relation Type

- ***Customer to Order:*** One-to-Many
- ***Order to Customer:*** Many-to-One (inverse of One-to-Many)
- ***Product to Product Line:*** One-to-Many
- ***Product Line to Product:*** Many-to-One (inverse of One-to-Many)
- ***Order to Order Line Item:*** One-to-Many
- ***Order Line Item to Order:*** Many-to-One (inverse of One-to-Many)
- ***Order Line Item to Product (through Used Quantity):*** Many-to-Many (requires a helper table)
- ***Order Line Item to Raw Material (through Used Quantity):*** Many-to-Many (requires a helper table)
- **Raw Material to Order Line Item:** Many-to-Many (inverse of above Many-to-Many)
- ***Raw Material to Vendor (through Supply):*** Many-to-Many (requires a helper table)
- ***Vendor to Raw Material (through Supply):*** Many-to-Many (inverse of above Many-to-Many)

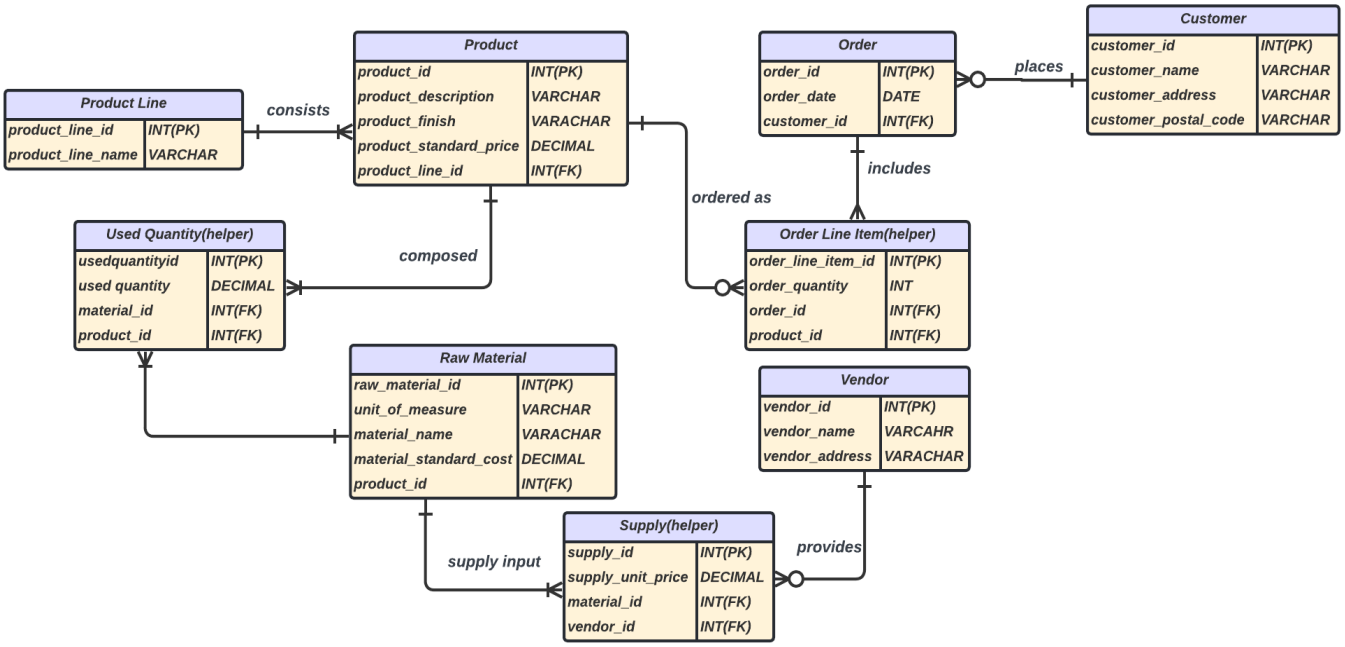
CONCEPTUAL DATABASE MODEL OF PINE VALLEY FURNITURE



LOGICAL DATABASE MODEL OF PINE VALLEY FURNITURE



PHYSICAL DATABASE MODEL OF PINE VALLEY FURNITURE



DDL Commands (Create Statement)

```
-- 1 table
create table productline(
productlineid int auto_increment primary key,
productlinename varchar(100)
);
-- 2 table
create table product(
productid int auto_increment primary key,
productdescription varchar (100),
```

```
productfinish varchar (100),
productstandardprice decimal,
productlineid int ,
foreign key (productlineid) references productline (productlineid)
);
-- 3 table
create table orders( -- I can't name as order b/c it is a reserved keyword
orderid int auto_increment primary key,
orderdate date,
customerid int,
foreign key (customerid) references customer (customerid)
);
-- 4 table
create table customer(
customerid int auto_increment primary key,
customername varchar (100),
customeraddress varchar(100),
customerpostalcode varchar(100)
);
-- 5 table
create table orderlineitem(  -- helper table for solving (M --> M) relation
orderlineitemid int auto_increment primary key,
orderquantity int ,
```

customerid int ,

orderid int,

foreign key (customerid) references customer (customerid),

foreign key (orderid) references orders (orderid)

);

-- 6 table

create table rawmaterial(

rawmaterialid int auto_increment primary key,

unitofmeasure varchar(100),

materialname varchar(100),

materialstandardcost decimal,

productid int ,

foreign key (productid) references product (productid)

);

-- 7 table

create table vendor(

vendorid int auto_increment primary key,

vendorname varchar(100),

vendoraddress varchar(100)

);

-- 8 table

create table supply(-- helper table

```
supplyid int auto_increment primary key,
supplyunitprice decimal,
rawmaterialid int,
vendorid int,
foreign key (rawmaterialid) references rawmaterial(rawmaterialid),
foreign key (vendorid) references vendor(vendorid)
);
-- 9 table
create table usedquantity( -- helper table
usedquantityid int auto_increment primary key,
usedquantity decimal,
rawmaterialid int ,
productid int,
foreign key (rawmaterialid) references rawmaterial(rawmaterialid),
foreign key (productid) references product(productid)
);
```


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

Sorry but MYSQL does not support creating sequence instead of sequences it supports auto increment, so I have done all the primary key to auto increment --> Because it is the alternative of sequence in MYSQL.

DML Commands (Insert Data into Tables)

-- inserting in table 1

insert into usedquantity(usedquantity,rawmaterialid,productid)
values

(10,1,1),

(12,2,2),

(9,3,3),

(8,4,4),

(3,5,6),

(2,6,7),

(4,7,1),

(11,8,2)

;

-- inserting data in table 2

insert into productline (productlinename) values

("Modern Living Room"),

("Cozy Bedroom"),

("Elegant Dinning Room"),

("Sleek Kitchen"),

("Functional Office"),

("Relaxing Outdoor"),

("ColorFul Kids Room"),

("Organized Storage"),

("Luxury Bathroom");

-- inserting data in table 3

insert into
product(productdescription,productfinish,productstandardprice,productlineid)

values

("Sofa","Leather",900,1),

("Single Bed","Wood",2000,2),

("Double Bed","Wood",3000,3),

("Dining Table","Glass",1000,4),

("Single Bed","Wood",2000,5),

("Kitchen Kabinet","Stainless Steel",1300,6),

("Desk","Metal",200,7);

-- inserting data in table 4

-- My SQL does not support sequence it's support auto increment

```
insert into customer
(customername,customeraddress,customerpostalcode) values
("Farooque","Jhuddo","KKM Word:02"),
("Sajjad","Jhuddo","KKM Word:02"),
("Talha","Tando Allayar","AR Word:09"),
("Rafy","TMK","MM Word:04"),
("Rohit","UK","U Word:03"),
("Fakhar","Lahore","LH Word:01"),
("Imam","Karachi","KR Word:08"),
("Babar","Peshawar","PW Word:11"),
("Saim","Multan","ML Word:17"),
("Amir","Karachi","KR Word:02");
```



```
-- inserting data in table 5
insert into orders (orderdate,customerid) values
("2024-03-20",1),
("2024-03-21",2),
("2024-03-22",3),
("2024-03-23",4),
("2024-03-24",5),
("2024-03-25",6),
```

("2024-03-26",7),

("2024-03-27",1),

("2024-03-28",2);

-- inserting data in table 6

insert into orderlineitem(orderquantity,customerid,orderid) values

(2,1,1),

(1,1,8),

(3,2,2),

(4,2,9),

(5,3,3),

(10,4,4),

(11,5,5),

(10,6,6),

(20,7,7);

-- inserting into table 3 again

insert into product(productdescription,productfinish,productstandardprice,productlineid)

values

("Chair","Wood",200,5);

-- inserting into the table 7

insert into rawmaterial (unitofmeasure, materialname, materialstandardcost, productid) values

("sq.ft", "Leather", 15, 1),

("kg", "Wood", 10, 2),

("kg", "Wood", 11, 3),

("kg", "Wood", 11, 8),

("sq.m", "Glass", 12, 4),

("m", "Stainless Steel", 18, 6),

("kg", "Metal", 8, 7);

-- inserting into table 8

insert into vendor (vendorname, vendoraddress)

values

('Leather Supplier A', '123 Main Street karachi'),

('Wood Supplier B', '456 Fateh Chowk Street Hyderabad'),

('Glass Supplier C', '789 M.A Street Lahore'),

('Stainless Steel Supplier D', '101 Pine Street Islamabad'),

('Metal Supplier E', '202 Maple Street Multan');

Inserting into table 9

insert into supply(supplyunitprice,rawmaterialid,vendorid) values

(100,1,1),

(120,2,2),

(120,3,2),

(130,4,2),

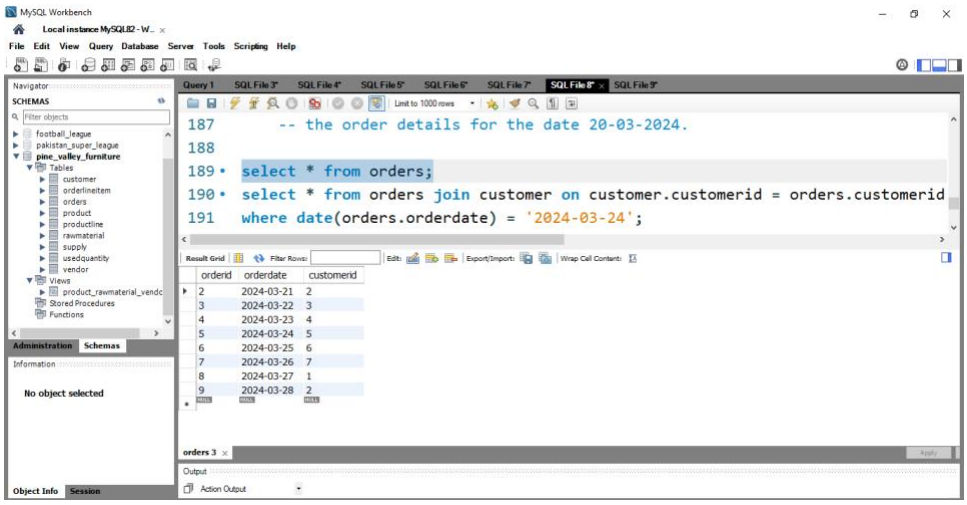
(110,5,3),

(140,6,4),

(199,7,5);

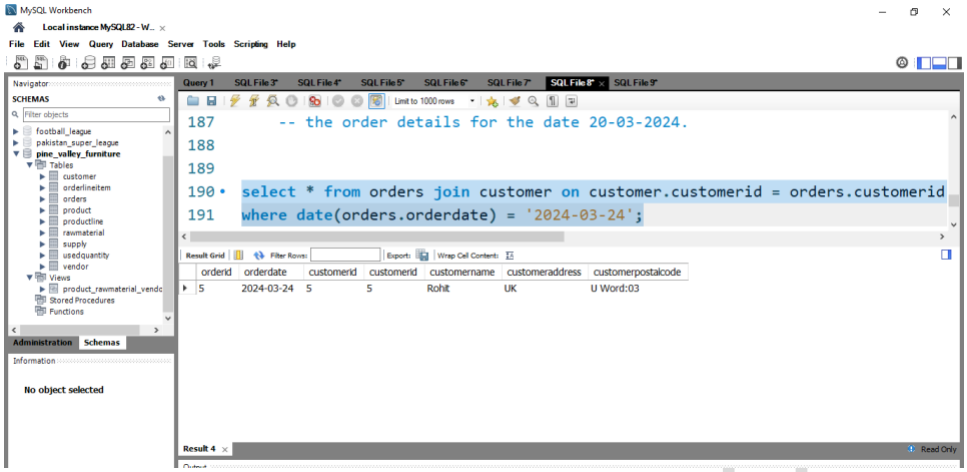
Retrieve the order details for all customers who ordered on 24-03-2024 and delete the order details for the date 20-03-2024.

First Retrieving all the details

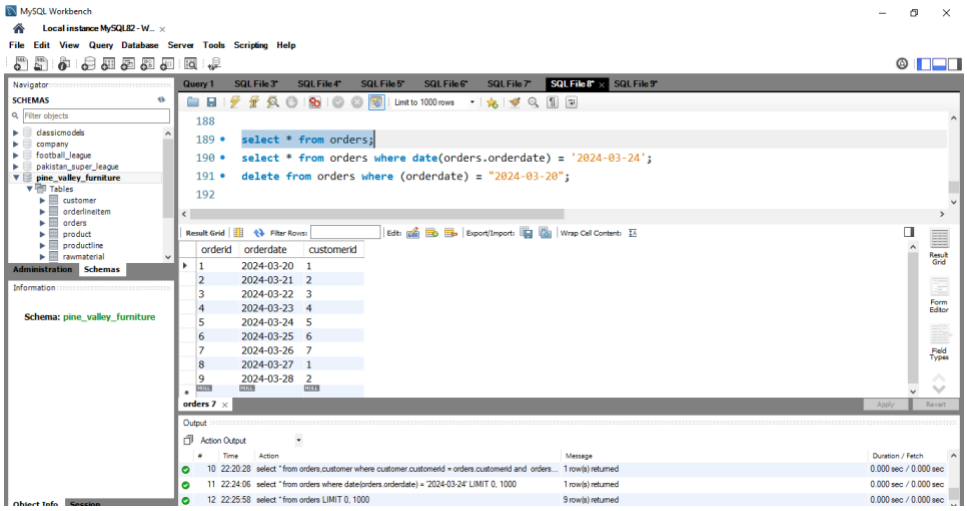


Now Applying Condition (24-03-2024)

```
select * from orders join customer on customer.customerid =
orders.customerid where date(orders.orderdate) = '2024-03-24';
```



Before Deleting (date 20-03-2024.)

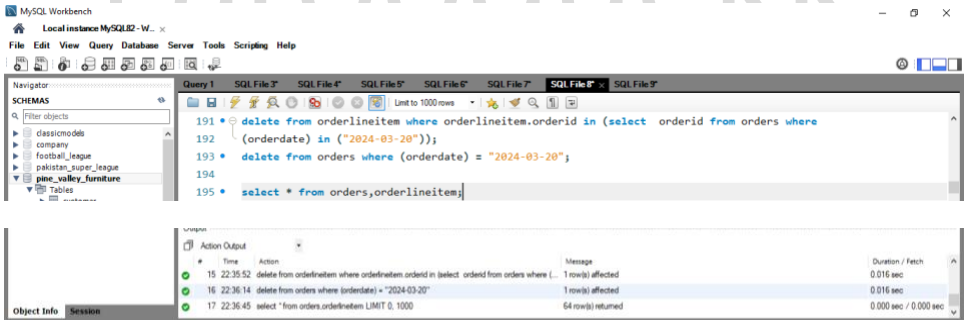


Applying Delete Statement

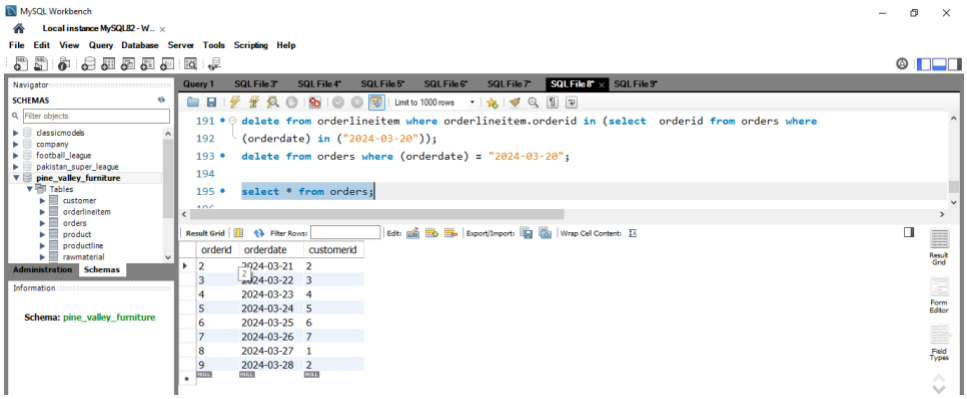
Note: If you are deleting a row in any table but if any of the column in that row is used as a foreign key it will show you referential error so before deleting that row you have to delete the subsequent column which is used in another table first then you are able to delete in that table in my case I have to delete first from order line item then I will be able to delete in the order table

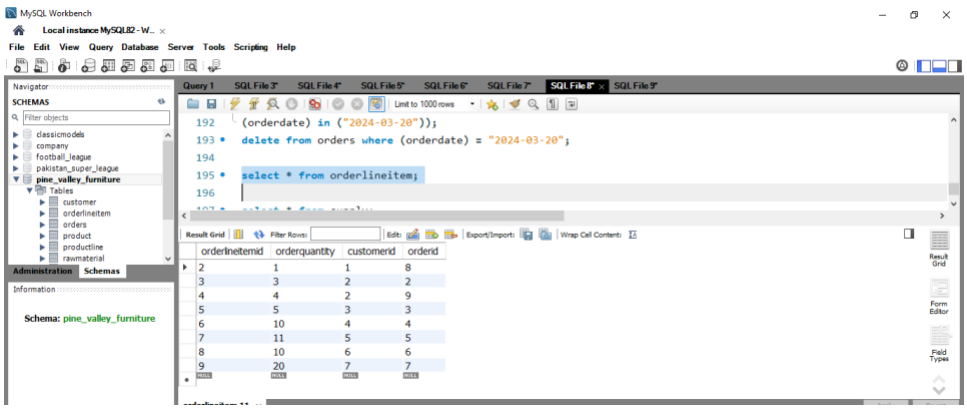
delete from orderlineitem where orderlineitem.orderid in (select orderid from orders where (orderdate) in ("2024-03-20"));

delete from orders where (orderdate) = "2024-03-20";



After Deleting See the Order and Order line-item Table

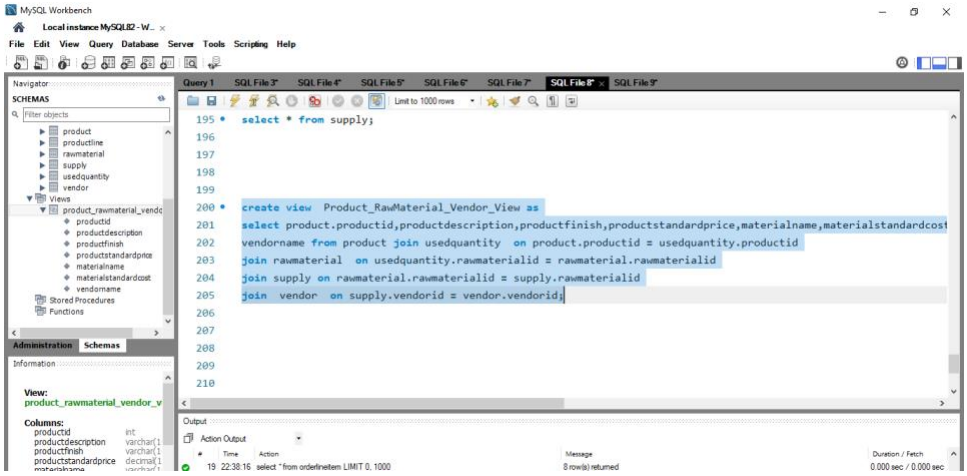


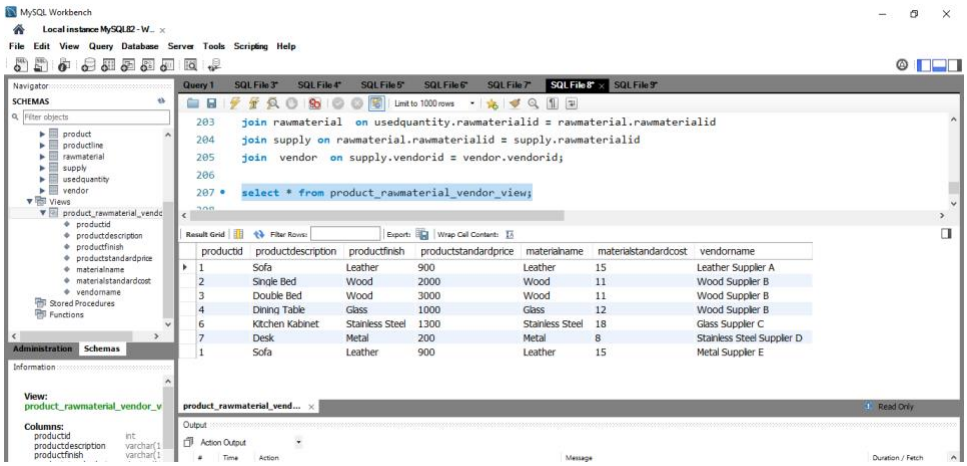


Create a complex view on the product, raw material, and vendor tables displaying the raw materials in each product and the corresponding vendors. Only include relevant columns from each table in the view.


```
create view  Product_RawMaterial_Vendor_View as select
product.productid,productdescription,productfinish,productstandardpri
ce,materialname,materialstandardcost,

vendorname from product join usedquantity  on product.productid =
usedquantity.productid

join rawmaterial  on usedquantity.rawmaterialid =
rawmaterial.rawmaterialid

join supply on rawmaterial.rawmaterialid = supply.rawmaterialid

join  vendor  on supply.vendorid = vendor.vendorid;
```

select * from product_rawmaterial_vendor_view;





Conclusion

In conclusion, the Pine Valley Furniture database project successfully established a robust relational database system to manage product inventory, customer orders, and vendor relationships. This solution enhances data organization, accessibility, and analysis, providing the company with valuable insights for informed decision-making and improved operational efficiency. With opportunities for future enhancements, the database lays the groundwork for continued growth and innovation within Pine Valley Furniture's business operations.