**Lab 2 – SQL II**

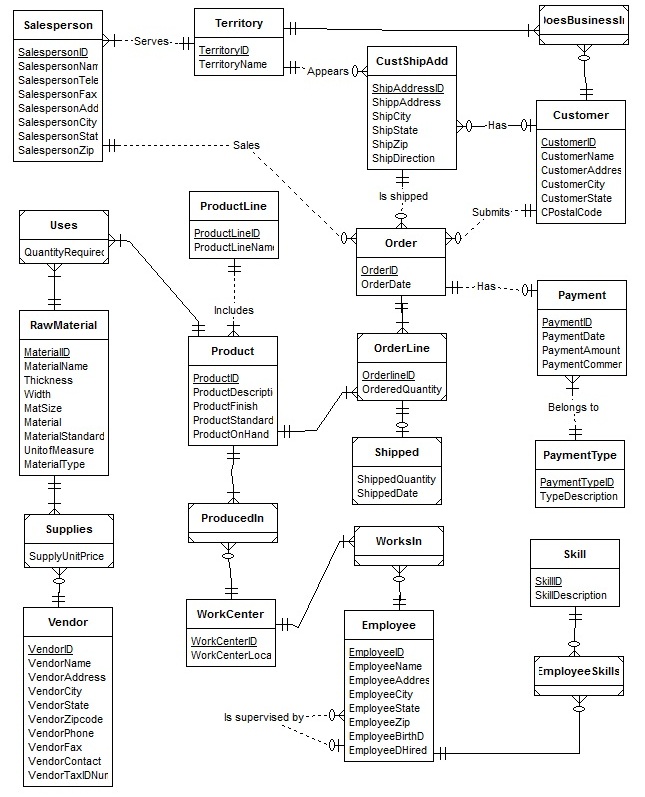
**Join Queries**

**Introduction**

The following SQL exercises are based on Pine Valley Furniture Company (PVFC) case study as described in the reference book. You can see the initial ERD in Figure 2-22. Based on the business processes at PVFC, the following entity types have been identified:

* The company sells a number of different furniture products. These products are grouped into several product lines. The identifier for product is Product ID, whereas the identifier for product line is Product Line ID. We identify the following additional attributes for product: Product Description, Product Finish, and Product Standard Price. Another attribute for product line is the 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 each order is Order ID, and another attribute is Order Date. A customer may submit any number of orders. Each order is submitted by exactly one costumer. The identifier for a costumer 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 PVF 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.
* PVF has established sales territories for its customers. Each customer may do business in any number of these sales territories or may not do business in any territory. A sales territory has one to many customers. The identifier for a sales territory is Territory ID and an attribute is Territory Name.
* PVF has several salespersons. The identifier for salespersons is Salesperson ID. Other attribute include Salesperson Name, Salesperson Telephone, and Salesperson Fax. A sales person serves exactly one sales territory. Each territory serves by one or more salespersons.
* Each product assembled from specified quantity of one or more raw materials. The identifier for the raw material entity is Material ID. Other attributes include Unite Of Measure, Material Name, Material Standard Cost. Each material 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 PVF. Supply Unit Price is the unit price at which a particular vendor supplies a particular raw material.
* PVF has established a number of work centres. The identifier for a work center is Work Center ID. Another attribute is Work Center Location. Each product is produced in one or more work centers. A work center may be used to produce any number of products or may not be used to produce any products.
* The company has more than 100 employees. The identifier for employee is Employee ID. Other attributes include Employee Name, Employee Address, and Skill. An employee may have more than one skill. Each employee may work in one or more work centers. A work center must have at least one employee woking in that center but may have any number of employee. A skill may be possessed by more than one employee or possibly no employees.
* Each employee has exactly one supervisor; however, a manager has no supervisor. An employee who is a supervisor may supervise any number of employees, but not all employees are supervisors.

**ERD**



**Database**

PVFC Database Tables are as follows:

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE Customer\_T

(CustomerID NUMERIC(4) NOT NULL,

CustomerName VARCHAR(25) ,

CustomerAddress VARCHAR(30) ,

CustomerCity VARCHAR(20) ,

CustomerState CHAR(2) ,

CustomerPostalCode VARCHAR(10) ,

CONSTRAINT Customer\_PK PRIMARY KEY (CustomerID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE Territory\_T

(TerritoryID NUMERIC(4) NOT NULL,

TerritoryName VARCHAR(50) ,

CONSTRAINT Territory\_PK PRIMARY KEY (TerritoryID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE DoesBusinessIn\_T

(CustomerID NUMERIC(4) NOT NULL,

TerritoryID NUMERIC(4) NOT NULL,

CONSTRAINT DoesBusinessIn\_PK PRIMARY KEY (CustomerID, TerritoryID),

CONSTRAINT DoesBusinessIn\_FK1 FOREIGN KEY (CustomerID) REFERENCES Customer\_T(CustomerID),

CONSTRAINT DoesBusinessIn\_FK2 FOREIGN KEY (TerritoryID) REFERENCES Territory\_T(TerritoryID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE Salesperson\_T

(SalespersonID NUMERIC(4) NOT NULL,

SalespersonName VARCHAR(25) ,

SalespersonTelephone VARCHAR(50) ,

SalespersonFax VARCHAR(50) ,

SalespersonAddress VARCHAR(30) ,

SalespersonCity VARCHAR(20) ,

SalespersonState CHAR(2) ,

SalespersonZip VARCHAR(20) ,

SalesTerritoryID NUMERIC(4) ,

CONSTRAINT Salesperson\_PK PRIMARY KEY (SalespersonID),

CONSTRAINT Salesperson\_FK1 FOREIGN KEY (SalesTerritoryID) REFERENCES Territory\_T(TerritoryID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE Skill\_T

(SkillID VARCHAR(12) NOT NULL,

SkillDescription VARCHAR(30) ,

CONSTRAINT Skill\_PK PRIMARY KEY (SkillID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE Employee\_T

(EmployeeID VARCHAR(10) NOT NULL,

EmployeeName VARCHAR(25) ,

EmployeeAddress VARCHAR(30) ,

EmployeeCity VARCHAR(20) ,

EmployeeState CHAR(2) ,

EmployeeZip VARCHAR(10) ,

EmployeeBirthDate DATE ,

EmployeeDateHired DATE ,

EmployeeSupervisor VARCHAR(10) ,

CONSTRAINT Employee\_PK PRIMARY KEY (EmployeeID),

CONSTRAINT Employeer\_FK FOREIGN KEY (EmployeeSupervisor) REFERENCES Employee\_T(EmployeeID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE EmployeeSkills\_T

(EmployeeID VARCHAR(10) NOT NULL,

SkillID VARCHAR(12) NOT NULL,

QualifyDate DATE ,

CONSTRAINT EmployeeSkills\_PK PRIMARY KEY (EmployeeID, SkillID),

CONSTRAINT EmployeeSkills\_FK1 FOREIGN KEY (EmployeeID) REFERENCES Employee\_T(EmployeeID),

CONSTRAINT EmployeeSkills\_FK2 FOREIGN KEY (SkillID) REFERENCES Skill\_T(SkillID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE WorkCenter\_T

(WorkCenterID VARCHAR(12) NOT NULL,

WorkCenterLocation VARCHAR(30) ,

CONSTRAINT WorkCenter\_PK PRIMARY KEY (WorkCenterID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE WorksIn\_T

(EmployeeID VARCHAR(10) NOT NULL,

WorkCenterID VARCHAR(12) NOT NULL,

CONSTRAINT WorksIn\_PK PRIMARY KEY (EmployeeID, WorkCenterID),

CONSTRAINT WorksIn\_FK1 FOREIGN KEY (EmployeeID) REFERENCES Employee\_T(EmployeeID),

CONSTRAINT WorksIn\_FK2 FOREIGN KEY (WorkCenterID) REFERENCES WorkCenter\_T(WorkCenterID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE ProductLine\_T

(ProductLineID NUMERIC(4) NOT NULL,

ProductLineName VARCHAR(50) ,

CONSTRAINT ProductLine\_PK PRIMARY KEY (ProductLineID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE Product\_T

(ProductID NUMERIC(4) NOT NULL,

ProductLineID NUMERIC(4) ,

ProductDescription VARCHAR(50) ,

ProductFinish VARCHAR(20) ,

ProductStandardPrice NUMERIC(6,2) ,

ProductOnHand NUMERIC(6) ,

CONSTRAINT Product\_PK PRIMARY KEY (ProductID),

CONSTRAINT Product\_FK1 FOREIGN KEY (ProductLineID) REFERENCES ProductLine\_T(ProductLineID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE ProducedIn\_T

(ProductID NUMERIC(4) NOT NULL,

WorkCenterID VARCHAR(12) NOT NULL,

CONSTRAINT ProducedInPK PRIMARY KEY (ProductID, WorkCenterID),

CONSTRAINT ProducedInFK1 FOREIGN KEY (ProductID) REFERENCES Product\_T(ProductID),

CONSTRAINT ProducedInFK2 FOREIGN KEY (WorkCenterID) REFERENCES WorkCenter\_T(WorkCenterID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE CustomerShipAddress\_T

(ShipAddressID NUMERIC(4) NOT NULL,

CustomerID NUMERIC(4) NOT NULL,

TerritoryID NUMERIC(4) NOT NULL,

ShipAddress VARCHAR(30) ,

ShipCity VARCHAR(20) ,

ShipState CHAR(2) ,

ShipZip VARCHAR(10) ,

ShipDirections VARCHAR(100) ,

CONSTRAINT CSA\_PK PRIMARY KEY (ShipAddressID),

CONSTRAINT CSA\_FK1 FOREIGN KEY (CustomerID) REFERENCES Customer\_T(CustomerID),

CONSTRAINT CSA\_FK2 FOREIGN KEY (TerritoryID) REFERENCES Territory\_T(TerritoryID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE Order\_T

(OrderID NUMERIC(5) NOT NULL,

CustomerID NUMERIC(4) ,

OrderDate DATE ,

FulfillmentDate DATE ,

SalespersonID NUMERIC(4) ,

ShipAdrsID NUMERIC(4) ,

CONSTRAINT Order\_PK PRIMARY KEY (OrderID),

CONSTRAINT Order\_FK1 FOREIGN KEY (CustomerID) REFERENCES Customer\_T(CustomerID),

CONSTRAINT Order\_FK2 FOREIGN KEY (SalespersonID) REFERENCES Salesperson\_T(SalespersonID),

CONSTRAINT Order\_FK3 FOREIGN KEY (ShipAdrsID) REFERENCES CustomerShipAddress\_T(ShipAddressID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE OrderLine\_T

(OrderLineID NUMERIC(4) NOT NULL,

OrderID NUMERIC(5) NOT NULL,

ProductID NUMERIC(4) NOT NULL,

OrderedQuantity NUMERIC(10) ,

CONSTRAINT OrderLine\_PK PRIMARY KEY (OrderLineID),

CONSTRAINT OrderLine\_FK1 FOREIGN KEY (OrderID) REFERENCES Order\_T(OrderID),

CONSTRAINT OrderLine\_FK2 FOREIGN KEY (ProductID) REFERENCES Product\_T(ProductID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE PaymentType\_T

(PaymentTypeID VARCHAR(50) NOT NULL,

TypeDescription VARCHAR(50) NOT NULL,

CONSTRAINT PaymentType\_PK PRIMARY KEY (PaymentTypeID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE Payment\_T

(PaymentID NUMERIC(5) NOT NULL,

OrderID NUMERIC(5) NOT NULL,

PaymentTypeID VARCHAR(50) NOT NULL,

PaymentDate DATE ,

PaymentAmount NUMERIC(6,2) ,

PaymentComment VARCHAR(255) ,

CONSTRAINT Payment\_PK PRIMARY KEY (PaymentID),

CONSTRAINT Payment\_FK1 FOREIGN KEY (OrderID) REFERENCES Order\_T(OrderID),

CONSTRAINT Payment\_FK2 FOREIGN KEY (PaymentTypeID) REFERENCES PaymentType\_T(PaymentTypeID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE Shipped\_T

(OrderLineId NUMERIC(4) NOT NULL,

ShippedQuantity NUMERIC(10) NOT NULL,

ShippedDate DATE,

CONSTRAINT Shipped\_PK PRIMARY KEY (OrderLineId),

CONSTRAINT Shipped\_FK1 FOREIGN KEY (OrderLineId) REFERENCES OrderLine\_T(OrderLineID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE Vendor\_T

(VendorID NUMERIC(4) NOT NULL,

VendorName VARCHAR(25) ,

VendorAddress VARCHAR(30) ,

VendorCity VARCHAR(20) ,

VendorState CHAR(2) ,

VendorZipcode VARCHAR(50) ,

VendorPhone VARCHAR(12) ,

VendorFax VARCHAR(12) ,

VendorContact VARCHAR(50) ,

VendorTaxIDNumber VARCHAR(50) ,

CONSTRAINT Vendor\_PK PRIMARY KEY (VendorID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE RawMaterial\_T

(MaterialID VARCHAR(12) NOT NULL,

MaterialName VARCHAR(30) ,

Thickness VARCHAR(50) ,

Width VARCHAR(50) ,

MatSize VARCHAR(50) ,

Material VARCHAR(15) ,

MaterialStandardPrice NUMERIC(6,2) ,

UnitOfMeasure VARCHAR(15) ,

MaterialType VARCHAR(50),

CONSTRAINT RawMaterial\_PK PRIMARY KEY (MaterialID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE Uses\_T

(MaterialID VARCHAR(12) NOT NULL,

ProductID NUMERIC(4) NOT NULL,

QuantityRequired NUMERIC(5) ,

CONSTRAINT Uses\_PK PRIMARY KEY (ProductID, MaterialID),

CONSTRAINT Uses\_FK1 FOREIGN KEY (ProductID) REFERENCES Product\_T(ProductID),

CONSTRAINT Uses\_FK2 FOREIGN KEY (MaterialID) REFERENCES RawMaterial\_T(MaterialID));

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE Supplies\_T

(VendorID NUMERIC(4) NOT NULL,

MaterialID VARCHAR(12) NOT NULL,

SupplyUnitPrice NUMERIC(6,2) ,

CONSTRAINT Supplies\_PK PRIMARY KEY (VendorID, MaterialID),

CONSTRAINT Supplies\_FK1 FOREIGN KEY (MaterialID) REFERENCES RawMaterial\_T(MaterialID),

CONSTRAINT Supplies\_FK2 FOREIGN KEY (VendorID) REFERENCES Vendor\_T(VendorID));;

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Exercises**

Load your database with the Pine Valley Furniture Company data and run the following queries:

1. Write an SQL command that will find any customers who have not placed orders.
2. List the names and number of employee supervised (label this value HeadCount) for each supervisor who supervises more than two employees.
3. Write a query to display the salespersons’ name in alphabetic order and the number of their orders.
4. Write a query to display each item ordered in order number 1, its standard price, and total price for each item ordered.
5. Write an SQL command to total the cost of order number 1.
6. List product descriptions and the number of raw materials used in them.
7. Calculate the total raw material cost (lable TotCost) for each product compared to its standard product price. Display product description, standard price and the total cost in the result.
8. For every order that has been received, display the order ID, the total dollar amount owed on that order, and the amount received in payments on that order. List the results in decreasing order of the difference between total due and amount paid.
9. Display the salesperson name, product finish, and total quantity sold for each finish by each salesperson.
10. Write a query to list the number of products produced in each work center. If a work center does not produce and products, display the result with a total of 0.
11. Produce a list of all products (product description) and the number of times each product has been ordered.