

Lawrence Chung  
1490133

OMIS 105 FINAL

-- 1. List the customers (ID and name) who live in California or New York. Order them by zip code, from high to low. (1 point)

```
SELECT CustomerId, CustomerName
FROM Customer_T
WHERE CustomerState='CA' OR CustomerState='NY'
ORDER BY CustomerPostalCode DESC
```

	CustomerId	CustomerName
1	5	Impressions
2	8	Dunkins Furniture
3	9	A Carpet
4	16	ABC Furniture Co.
5	3	Home Furnishings

-- 2. Write an SQL query that will find any salesperson (ID) who have not facilitated any order in 2017. (2 points)

```
SELECT SalespersonID
FROM Salesperson_T
WHERE SalesPersonID NOT IN
(
SELECT SalesPersonID
FROM Order_T
WHERE YEAR(OrderDate) = '2017'
)
```

	SalespersonID
1	1
2	4
3	6
4	8
5	9
6	10

-- 3. Display the product line ID, product line name and the average standard price for Walnut made products in those product lines that have an average standard price of at least \$500. (3 points)

```
SELECT Product_T.ProductLineID, ProductLineName, AVG(ProductStandardPrice) AS
AveragePrice
FROM Product_T INNER JOIN ProductLine_T ON Product_T.ProductLineID =
ProductLine_T.ProductLineID
WHERE ProductFinish = 'Walnut'
GROUP BY Product_T.ProductLineID, ProductLineName
HAVING AVG(ProductStandardPrice) >= 500
```

	ProductLineID	ProductLineName	AveragePrice
1	2	Antique	1200.00

-- 4. List, in alphabetical order, the names of all employees (managers) who are managing people in the SM1 work center. (3 points)

```
SELECT EmployeeName
FROM Employee_T INNER JOIN
(
SELECT WorksIn_T.EmployeeID, WorkCenterID, EmployeeSupervisor
FROM WorksIn_T INNER JOIN Employee_T On WorksIn_T.EmployeeID = Employee_T.EmployeeID
WHERE WorkCenterID = 'SM1'
) Temp_T ON Temp_T.EmployeeSupervisor = Employee_T.EmployeeID
ORDER BY EmployeeName
```

	EmployeeName
1	Robert Lewis

-- 5. Write a query to list the SalespersonID, SalespersonName and the number of orders facilitated (label as TotalOrders). If a salesperson does not facilitate any orders, display the result with a total of 0. (3 points)

```
SELECT Salesperson_T.SalespersonID, SalespersonName, COUNT(OrderID) AS TotalOrders
FROM Salesperson_T LEFT OUTER JOIN Order_T ON Salesperson_T.SalespersonID =
Order_T.SalesPersonID
GROUP BY Salesperson_T.SalespersonID, SalespersonName
```

	SalespersonID	SalespersonName	TotalOrders
1	1	Doug Henry	0
2	8	Fred Flinstone	1
3	5	Jacob Winslow	5
4	4	Julie Dawson	3
5	9	Mary James	1
6	10	Mary Smithson	0
7	6	Pepe Lepue	7
8	2	Robert Lewis	7
9	3	William Strong	23

-- 6. Retrieve a list of the customer states and the total profit earned from all of their orders on the Cherry and Birch products. Profit earned on a given order is the order total minus the total cost to produce the products sold. The cost of production for a given product is the total cost of all raw materials consumed in production. The query should output the state, total amount of all orders placed by its customers, total cost of production of all products sold in these orders, and total dollars profit earned. The output should be limited to the states with at least \$2000 profit. Sort the results in descending order of profit. (3 points)

```
SELECT CustomerState, SUM(OrderedQuantity * ProductStandardPrice) AS TotalOfAllOrders,
SUM(TotalCost * OrderedQuantity) AS TotalCost, (SUM(OrderedQuantity *
ProductStandardPrice) - SUM(TotalCost * OrderedQuantity)) AS TotalProfit
FROM Customer_T INNER JOIN Order_T ON Customer_T.CustomerID = Order_T.CustomerID INNER
JOIN OrderLine_T ON Order_T.OrderID = OrderLine_T.OrderID INNER JOIN Product_T ON
OrderLine_T.ProductID = Product_T.ProductID INNER JOIN
(
SELECT Product_T.ProductID, SUM(MaterialStandardPrice*QuantityRequired) AS TotalCost
FROM Product_T INNER JOIN Uses_T ON Product_T.ProductID = Uses_T.ProductID INNER JOIN
RawMaterial_T ON Uses_T.MaterialID = RawMaterial_T.MaterialID
WHERE ProductFinish IN ('Cherry', 'Birch')
GROUP BY Product_T.ProductID
) TotalCosts_T
ON Product_T.ProductID = TotalCosts_T.ProductID
GROUP BY CustomerState
HAVING (SUM(OrderedQuantity * ProductStandardPrice) - SUM(TotalCost *
OrderedQuantity)) >= 2000
ORDER BY (SUM(OrderedQuantity * ProductStandardPrice) - SUM(TotalCost * OrderedQuantity))
DESC
```

	CustomerState	TotalOfAllOrders	TotalCost	TotalProfit
1	NY	10825.00	6186.5	4638.5
2	NJ	9775.00	5423.7	4351.3

-- 7. Write an SQL query to list the order ID, product ID, and ordered quantity for all ordered products for which the ordered quantity is at least three times the average ordered quantity for that product. Also, the average ordered quantity for each product should appear at the end in a separate column. Limit the output to the products with the average ordered quantity of at least 2 . (3 points)

```
SELECT OrderID, OrderLine_T.ProductID, OrderedQuantity, ProductAverage
FROM OrderLine_T INNER JOIN
(
SELECT ProductID, AVG(OrderedQuantity) AS ProductAverage
FROM OrderLine_T
GROUP BY ProductID
)Temp_T ON OrderLine_T.ProductID = Temp_T.ProductID
WHERE OrderedQuantity >= ProductAverage * 3 AND ProductAverage >= 2
ORDER BY OrderID
```

	OrderID	ProductID	OrderedQuantity	ProductAverage
1	1	2	18	6
2	2	3	12	3

-- 8. For the those Ash materials that are provided by at least two vendors, write a query that lists the vendor ID, vendor name, material ID, material name, and size. (4 points)

```
SELECT Vendor_T.VendorID, VendorName, Supplies_T.MaterialID, MaterialName, Size
FROM Vendor_T INNER JOIN Supplies_T ON Vendor_T.VendorID = Supplies_T.VendorID INNER JOIN
RawMaterial_T ON Supplies_T.MaterialID = RawMaterial_T.MaterialID INNER JOIN
(
SELECT Supplies_T.MaterialID, COUNT(Vendor_T.VendorID) AS VendorCount
FROM Vendor_T INNER JOIN Supplies_T ON Vendor_T.VendorID = Supplies_T.VendorID INNER JOIN
RawMaterial_T ON Supplies_T.MaterialID = RawMaterial_T.MaterialID
GROUP BY Supplies_T.MaterialID
)Temp_T ON RawMaterial_T.MaterialID = Temp_T.MaterialID
WHERE VendorCount >= 2 AND Material = 'Ash'
```

	VendorID	VendorName	MaterialID	MaterialName	Size
1	2	Southern Lumber	1/21010ASH	1/2in X 10in X 10ft Ash	1/2in X 10in X 10ft
2	3	Pebbles Hardware	1/21010ASH	1/2in X 10in X 10ft Ash	1/2in X 10in X 10ft

-- 9. Display the salesperson names of all salespersons who have ordered at least one Cherry product and at least one Walnut product on the same or different orders. (4 points)

```
SELECT DISTINCT Salesperson_T.SalespersonName
FROM Salesperson_T INNER JOIN Order_T ON Salesperson_T.SalespersonID =
Order_T.SalesPersonID
WHERE
(Salesperson_T.SalespersonName IN
(
SELECT Salesperson_T.SalespersonName
FROM Salesperson_T INNER JOIN Order_T ON Salesperson_T.SalespersonID =
Order_T.SalesPersonID INNER JOIN OrderLine_T ON Order_T.OrderID = OrderLine_T.OrderID
INNER JOIN Product_T ON OrderLine_T.ProductID = Product_T.ProductID
WHERE ProductFinish = 'Cherry')) AND
(Salesperson_T.SalespersonName IN
(
SELECT Salesperson_T.SalespersonName
FROM Salesperson_T INNER JOIN Order_T ON Salesperson_T.SalespersonID =
Order_T.SalesPersonID INNER JOIN OrderLine_T ON Order_T.OrderID = OrderLine_T.OrderID
INNER JOIN Product_T ON OrderLine_T.ProductID = Product_T.ProductID
WHERE ProductFinish = 'Walnut'
))
```

	SalespersonName
1	Robert Lewis

-- 10. Retrieve a list of salespersons and the orders they have each facilitated, including order ID, order date, and order total price. Each order facilitated by a salesperson contributes to their total sales. Also provide the contribution of each order to the salesperson's total sales, as a percentage of total sales. The query should output the salesperson name, order ID, order date, order total price, distinct number of orders facilitated by salesperson, and percentage of salesperson total sales. Exclude orders with more than 30% contribution. Sort the list first in alphabetical order of salesperson name, then by ascending order date, and then with larger order totals listed first. (4 points)

```
SELECT SalespersonName, Order_T.OrderID, OrderDate, SUM(OrderedQuantity *
ProductStandardPrice) AS OrderTotalPrice, DistinctOrders, SUM((OrderedQuantity *
ProductStandardPrice) / TotalSales * 100) AS PercentageofSales
FROM Salesperson_T INNER JOIN Order_T ON Salesperson_T.SalespersonID =
Order_T.SalesPersonID INNER JOIN OrderLine_T ON Order_T.OrderID = OrderLine_T.OrderID
INNER JOIN Product_T ON OrderLine_T.ProductID = Product_T.ProductID INNER JOIN
(
SELECT SalesPersonID, SUM(OrderedQuantity * ProductStandardPrice) AS TotalSales, COUNT
(DISTINCT Order_T.OrderID) AS DistinctOrders
FROM Order_T INNER JOIN OrderLine_T ON Order_T.OrderID = OrderLine_T.OrderID INNER JOIN
Product_T ON OrderLine_T.ProductID = Product_T.ProductID
GROUP BY SalesPersonID
) TotalSales_T
ON Order_T.SalespersonID = TotalSales_T.SalespersonID
GROUP BY SalespersonName, Order_T.OrderID, OrderDate, DistinctOrders
HAVING SUM(((OrderedQuantity * ProductStandardPrice) / TotalSales) * 100) <= 30
ORDER BY SalespersonName ASC, OrderDate ASC, OrderTotalPrice DESC
```

	SalespersonName	OrderID	OrderDate	OrderTotalPrice	DistinctOrders	PercentageofSales
1	Jacob Winslow	66	2018-03-11 00:00:00.000	1650.00	3	25.88
2	Jacob Winslow	58	2018-03-11 00:00:00.000	750.00	3	11.76
3	Pepe Lepue	65	2018-03-11 00:00:00.000	0.00	2	0.00
4	Robert Lewis	3	2017-07-19 00:00:00.000	400.00	5	8.96
5	Robert Lewis	24	2018-03-10 00:00:00.000	0.00	5	0.00
6	Robert Lewis	69	2018-03-11 00:00:00.000	600.00	5	13.45
7	William Strong	5	2017-07-28 00:00:00.000	1675.00	8	7.77
8	William Strong	25	2018-03-10 00:00:00.000	1000.00	8	4.64
9	William Strong	26	2018-03-10 00:00:00.000	875.00	8	4.06
10	William Strong	28	2018-03-10 00:00:00.000	350.00	8	1.62
11	William Strong	39	2018-03-11 00:00:00.000	600.00	8	2.78
12	William Strong	49	2018-03-11 00:00:00.000	175.00	8	0.81