|  |
| --- |
| **Problem 11** |
| Show the First Name, Last Name, OrderNumber and OrderDate for all Customers even if they do not have any orders. |
| Hint: The information for this query is in the Customer and Sales Order tables. Use a JOIN on these two tables. The "common column" is SalesOrderNumber. To get all Customers, use a LEFT JOIN. |

Select c.FirstName,c.LastName,o.OrderNumber , o.OrderDate from Customer c

left join SalesOrder o on o.SalesOrderNumber = c.SalesOrderNumber;

|  |
| --- |
| **Problem 12** |
| Show the first and last names of the customer as one column, and the City and State separated by a comma as one column. Order by the state, the city and the customer's last name. |
| Hint: To concatenate columns, use the '+' key, e.g., AreaCode + '-' + PhoneNumber. |

|  |
| --- |
| **Problem 13** |
| For each part, show the Sales Orders that sold that part. Show the Parts that have no sales as well. Display the PartID, Part Name, Sales Order #, the Quantity sold and the price for the order. |
| Hint: A Left Join returns everything already selected even if it doesn't match the tables following. An Inner Join only returns those rows that match the condition in both tables. A Right Join is the opposite of a Left Join: It returns rows in every table AFTER the join even if the previous rows are not matched. |

|  |
| --- |
| **Problem 14** |
| For each Part, show the PartID, Part Name, Total Quantity sold and the total sales, including Parts with no sales. Order by the quantity sold descending, then by the name of the part. |
| Hint: Aggregrate columns return results based on "groups" of rows. If you select a non-aggregrate column, your query must GROUP BY those non-aggregrate columns. To convert NULL to 0, use IsNull(column name, 0), for example IsNull(OrderCost, 0). |
| Problem 15 |
| For each Sales Order, show how many days it took to ship the order in order by the longest order, then by Sales Order Number. Display Sales Order Number and the number of days to ship. Include the orders that have not yet shipped. |
| Hint: Look at the DATEDIFF function to return the difference between two dates. The two dates in this problem are the Order Date and the Shipment Date.   |  | | --- | | **Problem 16** | | Show the orders that are partial shipments meaning that the quantity shipped is less than the quantity ordered. Display the SalesOrderNumber, PartID, Quantity Ordered and the Quantity Shipped in order by SalesOrderNumber, PartID. | | Hint: All of this information comes from two tables: SalesOrderPart and ShipmentPart. Use the WHERE statement to find the rows where the SalesOrderPart Quantity is greater than (use '>') the the ShipmentPart Quantity. |  |  | | --- | | **Problem 17** | | Show the Parts that were returned. Display the name of the part, the date it was returned and the reason. | | Hint: This simple sounding query requires several tables joined together. Start with the SalesReturn table and trace the path to the Part table, joining the tables along the way. There are four tables. |  |  | | --- | | **Problem 18** | | Show the total sales by year. Display the year and the total sales. Sort by year. | | Hint: Try using the YEAR( some date ) function and the SUM( some column ) aggregate function. Don't forget your GROUP BY. Note that YEAR( some date ) is a function and not an aggregate meaning that it must be in the GROUP BY expression. |  |  | | --- | |  | |  | |  |  |  | | --- | |  | |  | |  |  |  | | --- | |  | |  | |  | |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
|  |
|  |
|  |

|  |
| --- |
|  |
|  |
|  |

|  |
| --- |
|  |
|  |
|  |

|  |
| --- |
|  |
|  |
|  |