SQL Tutorial

Use this site (please use Chrome Browser) to write your SQL queries and fill in your final answer below each question). The answers are in the compressed zip file. The password is the last 3 EmployeeIDs of the last question's answer (without spaces or quotes) e.g. "8410".

1. Display a list of all employees sorted according to Last Name in ascending order i.e. A-Z.
  SELECT * FROM Employees Order By LastName asc;

2. Display a list of all suppliers from France who but not from Paris.
  SELECT * FROM [Suppliers] where Country = 'France' and City != 'Paris';

3. Display all Suppliers whose SupplierName name begins with N.
  SELECT * FROM [Suppliers] where SupplierName like 'N%';

4. Display a list of each country where customers are located (N.B. Your list should not contain two of the same values).
  SELECT Country FROM [Customers] group by Country;

5. Display a list of all Customers and their order dates that made orders after 1996. Your result should look as follows:

Result:

Number of Records: 44

OrderID	OrderDate	CustomerName
10400	1997-01-01	Eastern Connection
10401	1997-01-01	Rattlesnake Canyon Grocery
10402	1997-01-02	Ernst Handel
10403	1997-01-03	Ernst Handel

  SELECT o.OrderId, o.OrderDate, c.CustomerName
  FROM Orders o JOIN Customers c ON o.CustomerID = c.CustomerID
  and o.OrderDate > '1996-12-31';

6. Display each Order and Product ID sold as well as the total sales for each product (sales = productprice*quantity). Hint: You will have to join the Products table to get the price of each product. Your result should look as follows…

Result:

Number of Records: 518

OrderID	ProductID	Sales
10248	11	252
10248	42	140
10248	72	174

SELECT od.OrderId, p.ProductID, ROUND(p.Price*od.Quantity, 2) as Sales FROM OrderDetails od JOIN Products p ON od.ProductID = p.ProductID;

7. Edit your previous query to display the Total Sales for each order. Note that orders may contain multiple products sold however we want to display the sum of all the sales for each order. Your result should look as follows…

Result:

Number of Records: 196

OrderID	Sales
10248	566
10249	2329.25
10250	2267.25
10251	839.5

SELECT od.OrderId, ROUND(SUM(p.Price*od.Quantity), 2) as Sales FROM OrderDetails od JOIN Products p ON od.ProductID = p.ProductID Group by od.OrderId;

8. Edit your previous query to display all Order ID's as well as their Total Sales where the Total Sales for the whole order is greater than 10000.

SELECT od.OrderId, ROUND(SUM(p.Price*od.Quantity), 2) as Sales FROM OrderDetails od JOIN Products p ON od.ProductID = p.ProductID Group by od.OrderId Having Sales > 10000;

9. Select all order IDs that sold Products with IDs 19 and 35 on the same order i.e. for each order listed, it needs to contain product ID 19 and product ID 35. N.B we're just looking for the OrderID to be returned. HINT: You can use a sub-query within your query.

SELECT OrderID FROM OrderDetails Where ProductID in (19, 35) group by OrderID Having COUNT(*) = 2;

10. Write a query to list all Employees as well as how many orders they have sold even if they have not made any orders and order the result by number of orders. Your list should like the below diagram…

Result:

Number of Records: 10

EmployeeID	Orders
10	0
9	6
5	11
7	14
6	18
2	20

SELECT e.EmployeeId, Count(o.OrderID) as Orders FROM Employees e Left Join Orders o ON e.EmployeeID = o.EmployeeID Group By e.EmployeeId Order by Orders asc