**SQL Tutorial**

Use this [site](http://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all) (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 Customers

Order By ContactName ASC;

2. Display a list of all suppliers from France who but not from Paris.

SELECT \* FROM Customers

WHERE Country = 'France' AND NOT City = 'Paris';

3. Display all Suppliers whose SupplierName name begins with N.

SELECT \* FROM Suppliers

WHERE SupplierName

LIKE ‘N%’

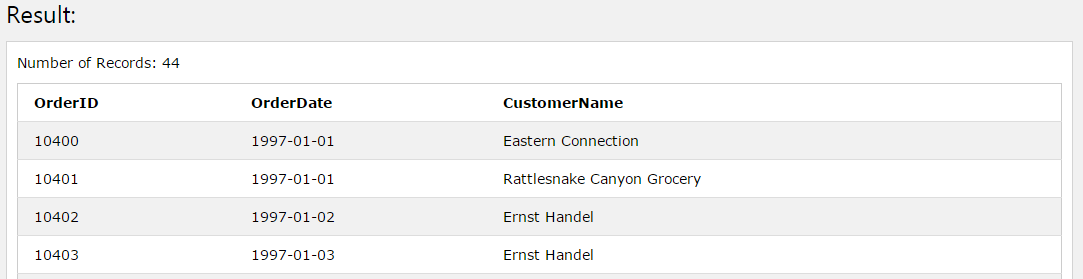
4. Display a list of each country where customers are located (N.B. Your list should not contain two of the same values).

SELECT Country, count(\*)

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:



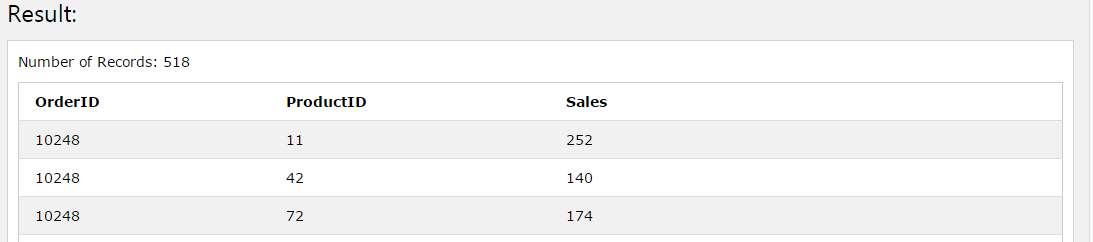
SELECT Orders.OrderID, Orders.OrderDate, Customers.CustomerName

FROM Orders

INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID

WHERE OrderDate >= '1997-01-01';

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…

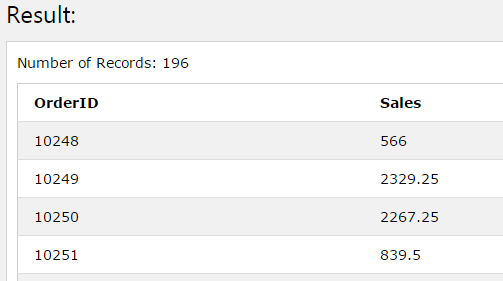


SELECT OrderDetails.OrderID, OrderDetails.ProductID, (Products.Price \* OrderDetails.Quantity) AS 'Sales'

FROM OrderDetails

INNER JOIN Products ON Products.ProductID = OrderDetails.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…



SELECT OrderDetails.OrderID, (Products.Price \* OrderDetails.Quantity) AS 'Sales'

FROM OrderDetails

INNER JOIN Products ON Products.ProductID = OrderDetails.ProductID;

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 OrderDetails.OrderID, (Products.Price \* OrderDetails.Quantity) AS 'Sales'

FROM OrderDetails

INNER JOIN Products ON Products.ProductID = OrderDetails.ProductID

WHERE (Products.Price \* OrderDetails.Quantity) > 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.

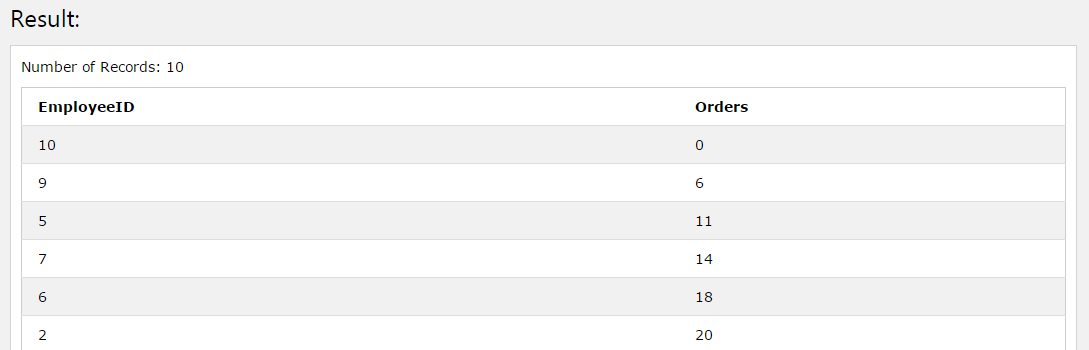
SELECT distinct OrderDetails.OrderID

FROM OrderDetails

WHERE OrderDetails.ProductID IN (19, 35)

Order By OrderDetails.OrderID;

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…



**SELECT EmployeeID, COUNT(\*) as Orders**

**FROM Orders**

**GROUP BY EmployeeID**

**ORDER BY EmployeeID DESC;**