**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 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.

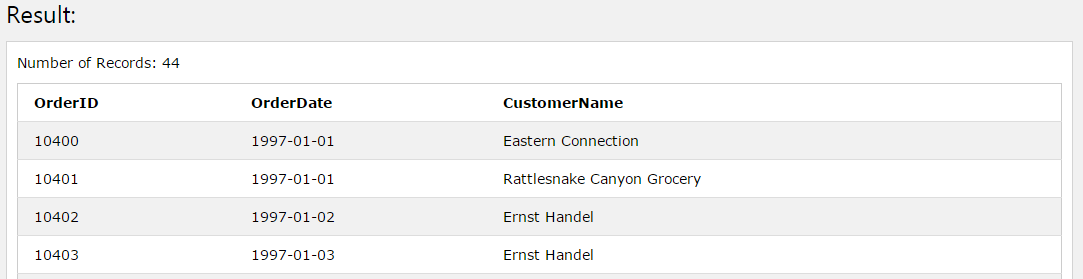
SELECT \* FROM Suppliers

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 Distinct(Country) FROM Customers;

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



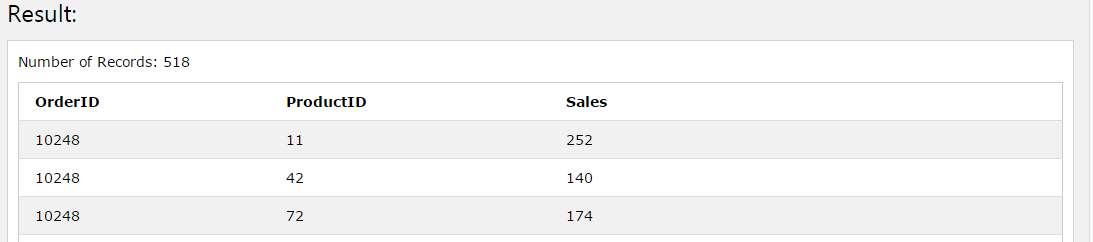
SELECT o.OrderId, o.Orderdate, c.CustomerName

FROM Orders o, Customers c

where o.orderdate > "1996-12-31"

and c.customerid = o.customerid;

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…



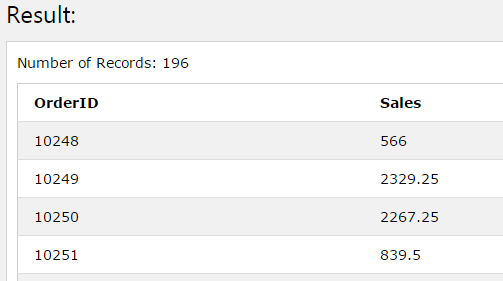
SELECT o.OrderId, p.productid, p.price\*o.quantity AS Sales

FROM orderdetails o

Join products p

On o.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…



SELECT o.OrderId, SUM(p.price\*o.quantity) AS Sales

FROM orderdetails o

Join products p

On o.productid = p.productid

GROUP BY o.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 o.OrderId, SUM(p.price\*o.quantity) AS [Total Sales]

FROM orderdetails o, products p

GROUP BY o.OrderId

Having SUM(p.price\*o.quantity) > 10000

And o.productid = p.productid;

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, productid

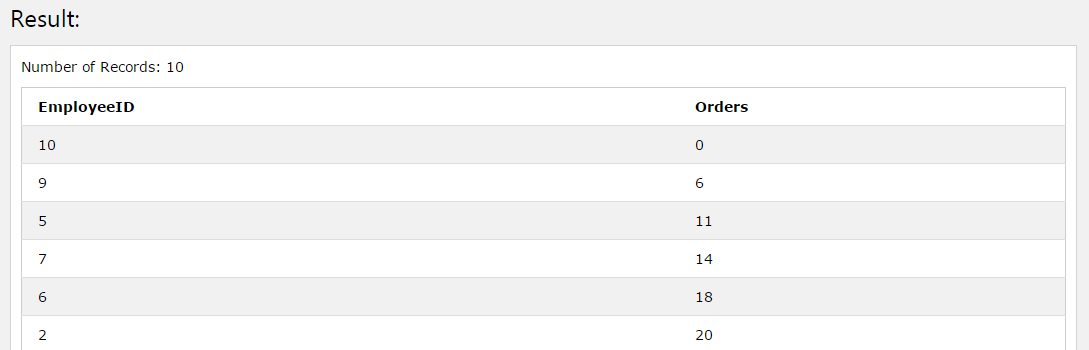
FROM orderdetails

Where productid IN (SELECT productid

FROM products

Where productid IN (19,35));

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 e.employeeid, COUNT(select count(\*) from Orders o where o.employeeid=e.employeeid) AS Orders FROM employees e;

SELECT e.employeeid, COUNT(o.employeeid) AS Orders FROM Orders o, employees e

Where e.employeeid = o.employeeid or e.employeeid not in (select employeeid from orders)

GROUP by e.employeeid