SQL Exercises

***https://www.w3schools.com/sql/trysql.asp?filename=trysql\_desc***

1. **Show all rows for Customers**

SELECT \* FROM Customers;

1. **Show only Contact name information for Customers**

SELECT ContactName FROM Customers;

1. **Show all unique combinations between Cities and Countries for the customers**

SELECT DISTINCT City, Country FROM Customers;

1. **Insert 3 new Customers**

INSERT INTO Customers VALUES (92, ‘Petrov’, ‘Petar’, ‘12 Opalchenska Str’, ‘Sofia’, ‘1000’, ‘Bulgaria’);

INSERT INTO Customers (CustomerName, City, Country) VALUES (‘Ivanov’, ‘Plovdiv’, ‘Bulgaria’);

INSERT INTO Customers (CustomerName) VALUES (‘Dimitrov’);

1. **Move all orders made by Andrew Fuller to Nancy Davolio**

SELECT EmployeeID FROM Employees WHERE LastName = ‘Davolio’;

SELECT EmployeeID FROM Employees WHERE LastName = ‘Fuller’;

UPDATE Orders SET EmployeeID = 1 WHERE EmployeeID = 2;

1. **Group all products by category and show category name**

SELECT Products.ProductName, Categories.CategoryName FROM Products INNER JOIN Categories ON Products.CategoryID = Categories.CategoryID ORDER BY Products.CategoryID;

1. **Sort all employees by Last Name and delete the last one. Do not remember to move all his/her orders to another colleague**

SELECT \* FROM Employees ORDER BY LastName;

UPDATE Orders SET EmployeeID = 9 WHERE EmployeeID = 10;

DELETE FROM Employees WHERE EmployeeID = 10;

1. S**how all customers without orders**

SELECT CustomerName FROM Customers LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID WHERE Orders.CustomerID IS NULL

1. **Show all products including 'ch' in its name with price between 10 and 20**

SELECT \* FROM Products WHERE ProductName LIKE '%ch%' AND (Price >= 10 AND Price <= 20);

1. **Group all products from 9 by category and sort by count in descending order**

SELECT COUNT(ProductID), CategoryID FROM Products GROUP BY CategoryID ORDER BY COUNT(ProductID) DESC;

11. **Inner Join example**

– Let’s see in which city we have customers as well as suppliers

SELECT Customers.City, Customers.CustomerName, Suppliers.SupplierName FROM Customers INNER JOIN Suppliers ON Customers.City = Suppliers.City ORDER BY Customers.City;

12. **Left Outer Join Example**

– Let’s see in which city we have customers but we do not have suppliers

SELECT Customers.City, Customers.CustomerName FROM Customers LEFT JOIN Suppliers ON Customers.City = Suppliers.City WHERE Suppliers.City IS NULL;

**13. Right Outer Join Example**

– Let’s see in which city we have suppliers but we do not have customers

SELECT Customers.City FROM Customers RIGHT JOIN Suppliers ON Customers.City = Suppliers.City WHERE Customers.City IS NULL;

**14. Full Outer Join Example in MySQL**

– Let’s see the cities in which we have customers and/or suppliers

SELECT Customers.City FROM Customers LEFT JOIN Suppliers ON Customers.City = Suppliers.City

UNION

SELECT Customers.City FROM Customers RIGHT JOIN Suppliers ON Customers.City = Suppliers.City WHERE Customers.City IS NULL;

or simply:

SELECT DISTINCT City FROM Customers

UNION

SELECT DISTINCT City FROM Suppliers;