SQL Exercises:

Please visit the below link to practice these questions,

<https://www.w3schools.com/sql/trymysql.asp?filename=trysql_asc>

**Questions:**

1. Write a SQL query to fetch “CustomerName” from Customers table using the alias name as “Customer”
2. Write an SQL query to fetch “CustomerName” from Customers table in upper case
3. Write an SQL query to fetch unique values of “City” from Suppliers table
4. Write an SQL query to print the first three characters of Country from Suppliers table
5. Write an SQL query to fetch OrderDate from Orders table in the format ‘dd-MM-yy’
6. Write an SQL query to fetch all columns from Employees, if the LastName is null then use FirstName in place of it
7. Write an SQL query to fetch unique list of full names from Employees by concatenating FirstName and LastName
8. Write an SQL query to fetch unique length of OrderID from Orders table, this is done as a quality check for OrderID length
9. SQL query that lists all customers with a NULL value in the "Address" field of Customers table
10. SQL statement that selects all the customers from the country "Mexico", in the "Customers" table
11. SQL statement that selects all fields from "Customers" where country is "Germany" AND city is "Berlin"
12. SQL statement that selects all fields from "Customers" where country is "Germany" OR "Spain"
13. SQL statement that selects all fields from "Customers" where country is NOT "Germany" and NOT "USA"
14. SQL statement selects all customers from the "Customers" table, sorted DESCENDING by the "Country" column
15. SQL statement that selects all customers from the "Customers" table, sorted by the "Country" and the "CustomerName" column. This means that it orders by Country, but if some rows have the same Country, it orders them by CustomerName
16. SQL statement that selects all customers from the "Customers" table, sorted ascending by the "Country" and descending by the "CustomerName" column
17. SQL statement that lists the number of customers in each country
18. Write an SQL query to select all columns from Customers table where country is “USA”, then append this to another query of all columns form Customers table where country is “UK”
19. SQL statement that returns the German cities (only distinct values) from both the "Customers" and the "Suppliers" table
20. SQL statement that returns the German cities (duplicate values also) from both the "Customers" and the "Suppliers" table
21. SQL statement that lists the ProductName if it finds ANY records in the OrderDetails table has Quantity equal to 10 (this will return TRUE because the Quantity column has some values of 10)
22. SQL statement that lists the ProductName if it finds ANY records in the OrderDetails table has Quantity larger than 1000 (this will return FALSE because the Quantity column has no values larger than 1000)
23. Write a SQL query to fetch the top customers for all the countries by number of orders made, sorted by ascending order of Country, followed by descending order of the Numbers of orders **(Use Customer Names** **and Country Names)**
24. Write a SQL query to fetch the top shippers for all the countries by number of unique orders delivered, sorted by ascending order of Country, followed by descending order of the Numbers of orders. **(Use Shipper Names and Country Names)**
25. Write a SQL query to fetch the top customers for all the countries by money spent, sorted by ascending order of Country, followed by descending order of the money spent by each customer **(Use Customer Names and Country Names)**
26. Write a SQL query to fetch the top products in all the countries by revenue generated, sorted by ascending order of Country, followed by descending order of the revenue generated. **(Use Product Names and Country Names)**
27. Write a SQL query to fetch the top products in all the product categories by revenue generated, sorted by ascending order of Product Category, followed by descending order of the revenue generated. **(Use Product Names, Category Names and Country Names)**
28. Write a SQL query to rank the top customers by the decreasing order of revenue generated (**Use Customer Names)**
29. Write a SQL query to rank the top products by the decreasing order of revenue generated (**Use:** **Product Names)**
30. Write a SQL query to rank the top customers at a country level by the decreasing order of revenue generated (**Use Customer Names)**

**Solutions:**

1. SELECT CustomerName as Customer FROM Customers
2. SELECT UPPER(CustomerName) FROM Customers;
3. SELECT DISTINCT City from Suppliers;
4. SELECT SUBSTRING(Country,1,3) FROM Suppliers;
5. SELECT FORMAT(OrderDate, 'dd-MM-yy') FROM Orders
6. SELECT EmployeeID, COALESCE(LastName, FirstName) as LastName, FirstName, BirthDate, Photo, Notes FROM Employees;
7. SELECT DISTINCT CONCAT(FirstName,' ',LastName) as Name FROM Employees;
8. SELECT DISTINCT LEN(OrderID) FROM Orders;
9. SELECT CustomerName, ContactName, Address FROM Customers WHERE Address IS NULL;
10. SELECT \* FROM Customers WHERE Country='Mexico';
11. SELECT \* FROM Customers WHERE Country='Germany' AND City='Berlin';
12. SELECT \* FROM Customers WHERE Country='Germany' OR Country='Spain';
13. SELECT \* FROM Customers WHERE NOT Country='Germany' AND NOT Country='USA';
14. SELECT \* FROM Customers ORDER BY Country DESC;
15. SELECT \* FROM Customers ORDER BY Country, CustomerName;
16. SELECT \* FROM Customers ORDER BY Country ASC, CustomerName DESC;
17. SELECT COUNT(CustomerID), Country FROM Customers GROUP BY Country;
18. A SELECT \* FROM Customers WHERE Country='USA' UNION SELECT \* FROM Customers WHERE Country='UK';
19. SELECT City, Country FROM Customers  
    WHERE Country='Germany'  
    UNION  
    SELECT City, Country FROM Suppliers  
    WHERE Country='Germany'  
    ORDER BY City;
20. SELECT City, Country FROM Customers  
    WHERE Country='Germany'  
    UNION ALL  
    SELECT City, Country FROM Suppliers  
    WHERE Country='Germany'  
    ORDER BY City;
21. SELECT ProductName  
    FROM Products  
    WHERE ProductID = ANY  
      (SELECT ProductID  
      FROM OrderDetails  
      WHERE Quantity = 10);
22. SELECT ProductName  
    FROM Products  
    WHERE ProductID = ANY  
      (SELECT ProductID  
      FROM OrderDetails  
      WHERE Quantity > 1000);

select A.CustomerID, A.CustomerName, A.Country, B.N\_Orders from

(select CustomerID, CustomerName, Country from Customers) as A

inner join (select count(distinct(OrderID)) as N\_Orders, CustomerID from Orders

group by CustomerID) as B

on A.CustomerID = B.CustomerID

order by A.Country, B.N\_Orders DESC

SELECT C.COUNTRY, C.SHIPPERID, COUNT(DISTINCT(C.ORDERID)) as N\_Orders

from

(select B.\*, A.Country from

(select CustomerID, Country from Customers) as A

inner join

(select \* from Orders) as B

on A.CustomerID = B.CustomerID) as C

group by C.ShipperID, C.Country

order by C.Country, Count(Distinct(C.OrderID)) desc

select Country, CustomerName, sum(Value) as Value\_Spent from

(select \*, (Price\*Quantity) as Value from

(select E.\*, F.Price from

(select D.Country, D.OrderID, C.ProductID, C.Quantity, D.CustomerID, D.CustomerName from

(select OrderID, ProductID, Quantity from Order\_Details) as C

inner join

(select A.CustomerID, A.CustomerName, A.Country, B.OrderID from

(select CustomerID, CustomerName, Country from Customers) as A

inner join

(select CustomerID, OrderID from Orders) as B

on A.CustomerID = B.CustomerID) as D

on C.OrderID = D.OrderID) as E

inner join

(select ProductID, Price from Products) as F

on E.ProductID = F.ProductID))

group by Country, CustomerName

order by Country, Value\_Spent desc

select Country, ProductName, sum(Value) as Value\_Spent from

(select \*, (Price\*Quantity) as Value from

(select E.\*, F.ProductName, F.Price from

(select D.Country, D.OrderID, C.ProductID, C.Quantity, D.CustomerID, D.CustomerName from

(select OrderID, ProductID, Quantity from Order\_Details) as C

inner join

(select A.CustomerID, A.CustomerName, A.Country, B.OrderID from

(select CustomerID, CustomerName, Country from Customers) as A

inner join

(select CustomerID, OrderID from Orders) as B

on A.CustomerID = B.CustomerID) as D

on C.OrderID = D.OrderID) as E

inner join

(select ProductID, ProductName, Price from Products) as F

on E.ProductID = F.ProductID))

group by Country, ProductName

order by Country, Value\_Spent desc

select CategoryName, ProductName, sum(Value) as Value\_Spent from

(select \*, (Price\*Quantity) as Value from

(select C.CategoryName, D.\* from

(select CategoryID, CategoryName from Categories) as C

inner join

(select B.\*, A.OrderID, A.Quantity from

(select OrderID, ProductID, Quantity from Order\_Details) as A

inner join

(select ProductID, ProductName, CategoryID, Price from Products) as B

on A.ProductID = B.ProductID) as D

on C.CategoryID = D.CategoryID))

group by CategoryName, ProductName

order by CategoryName, Value\_Spent desc

28.

select Country, CustomerName, sum(Value) as Value\_Spent, RANK() OVER (ORDER BY sum(Value) desc) AS Rank\_Num from

(select \*, (Price\*Quantity) as Value from

(select E.\*, F.Price from

(select D.Country, D.OrderID, C.ProductID, C.Quantity, D.CustomerID, D.CustomerName from

(select OrderID, ProductID, Quantity from Order\_Details) as C

inner join

(select A.CustomerID, A.CustomerName, A.Country, B.OrderID from

(select CustomerID, CustomerName, Country from Customers) as A

inner join

(select CustomerID, OrderID from Orders) as B

on A.CustomerID = B.CustomerID) as D

on C.OrderID = D.OrderID) as E

inner join

(select ProductID, Price from Products) as F

on E.ProductID = F.ProductID))

group by Country, CustomerName

order by Rank\_Num

29.

select Country, ProductName, sum(Value) as Value\_Spent , RANK() OVER (ORDER BY sum(Value) desc) AS Rank\_Num from

(select \*, (Price\*Quantity) as Value from

(select E.\*, F.ProductName, F.Price from

(select D.Country, D.OrderID, C.ProductID, C.Quantity, D.CustomerID, D.CustomerName from

(select OrderID, ProductID, Quantity from Order\_Details) as C

inner join

(select A.CustomerID, A.CustomerName, A.Country, B.OrderID from

(select CustomerID, CustomerName, Country from Customers) as A

inner join

(select CustomerID, OrderID from Orders) as B

on A.CustomerID = B.CustomerID) as D

on C.OrderID = D.OrderID) as E

inner join

(select ProductID, ProductName, Price from Products) as F

on E.ProductID = F.ProductID))

group by Country, ProductName

order by Rank\_Num

30.

select Country, ProductName, sum(Value) as Value\_Spent , RANK() OVER (PARTITION BY Country ORDER BY sum(Value) desc) AS Rank\_Num from

(select \*, (Price\*Quantity) as Value from

(select E.\*, F.ProductName, F.Price from

(select D.Country, D.OrderID, C.ProductID, C.Quantity, D.CustomerID, D.CustomerName from

(select OrderID, ProductID, Quantity from Order\_Details) as C

inner join

(select A.CustomerID, A.CustomerName, A.Country, B.OrderID from

(select CustomerID, CustomerName, Country from Customers) as A

inner join

(select CustomerID, OrderID from Orders) as B

on A.CustomerID = B.CustomerID) as D

on C.OrderID = D.OrderID) as E

inner join

(select ProductID, ProductName, Price from Products) as F

on E.ProductID = F.ProductID))

group by Country, ProductName

order by Rank\_Num