/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* SQL Queries: Practice your SQL Knowledge! \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Credit to Schema : https://github.com/AndrejPHP/w3schools-database \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Run w3schools.sql to set up database, tables and data\*/

/\*

----Schema----

Customers (CustomerID, CustomerName, ContactName, Address, City, PostalCode, Country)

Categories (CategoryID,CategoryName, Description)

Employees (EmployeeID, LastName, FirstName, BirthDate, Photo, Notes)

OrderDetails(OrderDetailID, OrderID, ProductID, Quantity)

Orders (OrderID, CustomerID, EmployeeID, OrderDate, ShipperID)

Products(ProductID, ProductName, SupplierID, CategoryID, Unit, Price)

Shippers (ShipperID, ShipperName, Phone)

\*/

/\*\*\*\* Advanced Level \*\*\*\*\*/

/\*1. Select customer name together with each order the customer made\*/

SELECT CustomerName, OrderID

FROM customers c

JOIN orders o

ON c.CustomerID = o.CustomerID;

/\*2. Select order id together with name of employee who handled the order\*/

SELECT o.OrderID, e.EmployeeID, e.FirstName, e.LastName

FROM orders o

JOIN employees e

ON o.EmployeeID = e.EmployeeID;

/\*3. Select customers who did not placed any order yet\*/

SELECT c.CustomerID, c.CustomerName, o.OrderID

FROM customers c

LEFT JOIN orders o

ON c.CustomerID = o.CustomerID

WHERE o.CustomerID IS NULL;

/\*4. Select order id together with the name of products\*/

SELECT o.OrderID, p.ProductID, p.ProductName

FROM orders o

JOIN order\_details od ON o.OrderID = od.OrderID

JOIN products p ON p.ProductID = od.ProductID

ORDER BY o.OrderID;

/\*5. Select products that no one bought\*/

SELECT p.ProductID, p.ProductName, od.OrderID

FROM products p

LEFT JOIN order\_details od ON p.ProductID = od.ProductID

WHERE od.OrderID IS NULL;

/\*6. Select customer together with the products that he bought\*/

SELECT c.CustomerID, c.CustomerName, p.ProductName

FROM customers c

JOIN orders o ON o.CustomerID = c.CustomerID

JOIN order\_details od ON od.OrderID = o.OrderID

JOIN products p ON p.ProductID = od.ProductID

ORDER BY c.CustomerID, p.ProductName ASC;

/\*7. Select product names together with the name of corresponding category\*/

SELECT p.ProductID, p.ProductName, c.CategoryName

FROM products p

JOIN categories c

ON p.CategoryID = c.CategoryID;

/\*8. Select orders together with the name of the shipping company\*/

SELECT o.OrderID, o.CustomerID, o.EmployeeID, o.OrderDate, shp.ShipperName

FROM orders o

JOIN shippers shp

ON o.ShipperID = shp.ShipperID

ORDER BY o.OrderID;

/\*9. Select customers with id greater than 50 together with each order they made\*/

SELECT c.CustomerID, c.CustomerName, o.OrderID

FROM customers c

JOIN orders o

ON c.CustomerID = o.CustomerID

WHERE c.CustomerID > 50;

/\*10. Select employees together with orders with order id greater than 10400\*/

SELECT o.OrderID, e.EmployeeID, e.FirstName, e.LastName

FROM orders o

JOIN employees e

ON o.EmployeeID = e.EmployeeID

WHERE o.OrderID > 10400;

/\*\*\*\*\*\*\*\*\*\*\*\* Expert Level \*\*\*\*\*\*\*\*\*\*\*\*/

/\*1. Select the most expensive product\*/

SELECT ProductID,ProductName,Price

FROM products

ORDER BY Price DESC

LIMIT 1;

/\*2. Select the second most expensive product\*/

/\*version 1\*/

SELECT ProductID,ProductName,Price

FROM products

ORDER BY Price DESC

LIMIT 1 OFFSET 1;

/\*version 2 (complex)\*/

WITH

tbl1 AS (SELECT ProductID,ProductName,Price

FROM products

ORDER BY Price DESC

LIMIT 2),

tbl2 AS (SELECT ProductID,ProductName,Price

FROM products

ORDER BY Price DESC

LIMIT 1)

SELECT tbl1.ProductID,tbl1.ProductName,tbl1.Price

FROM tbl1

LEFT JOIN tbl2 ON tbl1.ProductID = tbl2.ProductID

WHERE tbl2.ProductID IS NULL;

/\*3. Select name and price of each product, sort the result by price in decreasing order\*/

SELECT ProductID,ProductName,Price

FROM products

ORDER BY Price DESC;

/\*4. Select 5 most expensive products\*/

SELECT ProductID,ProductName,Price

FROM products

ORDER BY Price DESC

LIMIT 5;

/\*5. Select 5 most expensive products without the most expensive (in final 4 products)\*/

SELECT ProductID,ProductName,Price

FROM products

ORDER BY Price DESC

LIMIT 4 OFFSET 1;

/\*6. Select name of the cheapest product (only name) without using LIMIT and OFFSET\*/

WITH temp

AS (SELECT ProductID,ProductName,MIN(Price)

FROM products)

SELECT ProductName

FROM temp;

/\*7. Select name of the cheapest product (only name) using subquery\*/

SELECT ProductName

FROM products

WHERE Price IN (

SELECT MIN(Price) FROM products

);

/\*8. Select number of employees with LastName that starts with 'D'\*/

SELECT EmployeeID, LastName, FirstName

FROM employees

WHERE LastName LIKE 'D%';

/\* BONUS : same question for Customer this time \*/

SELECT CustomerName, SUBSTRING\_INDEX(CustomerName," ",1) AS firstName, SUBSTRING\_INDEX(CustomerName," ",-1) AS lastName

FROM customers

WHERE SUBSTRING\_INDEX(CustomerName," ",-1) LIKE 'D%';

/\*9. Select customer name together with the number of orders made by the corresponding customer

sort the result by number of orders in decreasing order\*/

SELECT c.CustomerID, c.CustomerName, COUNT(\*) AS 'TotalOder'

FROM customers c

JOIN Orders o

ON c.CustomerID = o.CustomerID

GROUP BY c.CustomerID

ORDER BY 3 DESC, 1 ASC;

/\*10. Add up the price of all products\*/

SELECT SUM(Price)

FROM products;

/\*11. Select orderID together with the total price of that Order, order the result by total price of order in increasing order\*/

SELECT od.OrderID, SUM((od.Quantity \* p.Price)) AS TotalValueOfOrder

FROM order\_details od

JOIN products p ON p.ProductID = od.ProductID

GROUP BY 1

ORDER BY 2 ASC;

/\*12. Select customer who spend the most money\*/

SELECT c.CustomerID, c.CustomerName, SUM(od.Quantity \* p.Price) AS TotalSpending

FROM orders o

JOIN customers c ON o.CustomerID = c.CustomerID

JOIN order\_details od ON o.OrderID = od.OrderID

JOIN products p ON p.ProductID = od.ProductID

GROUP BY c.CustomerID

ORDER BY 3 DESC

LIMIT 1;

/\*13. Select customer who spend the most money and lives in Canada\*/

SELECT c.CustomerID, c.CustomerName, SUM(od.Quantity \* p.Price) AS TotalSpending, c.Country

FROM orders o

JOIN customers c ON o.CustomerID = c.CustomerID

JOIN order\_details od ON o.OrderID = od.OrderID

JOIN products p ON p.ProductID = od.ProductID

WHERE c.Country LIKE 'Canada'

GROUP BY c.CustomerID

ORDER BY 3 DESC

LIMIT 1;

/\*14. Select customer who spend the second most money\*/

SELECT c.CustomerID, c.CustomerName, SUM(od.Quantity \* p.Price) AS TotalSpending

FROM orders o

JOIN customers c ON o.CustomerID = c.CustomerID

JOIN order\_details od ON o.OrderID = od.OrderID

JOIN products p ON p.ProductID = od.ProductID

GROUP BY c.CustomerID

ORDER BY 3 DESC

LIMIT 1 OFFSET 1;

/\*15. Select shipper together with the total price of proceed orders\*/

SELECT o.ShipperID, shp.ShipperName, SUM(od.Quantity \* p.Price) AS TotalValueOfOrder

FROM orders o

JOIN order\_details od ON o.OrderID = od.OrderID

JOIN products p ON p.ProductID = od.ProductID

JOIN shippers shp ON shp.ShipperID = o.ShipperID

GROUP BY 1

ORDER BY 2;

https://leetcode.com/problems/combine-two-tables/submissions/958964550