**SQL EXERCISES**

EXERCISE 1

-- Exercise 1

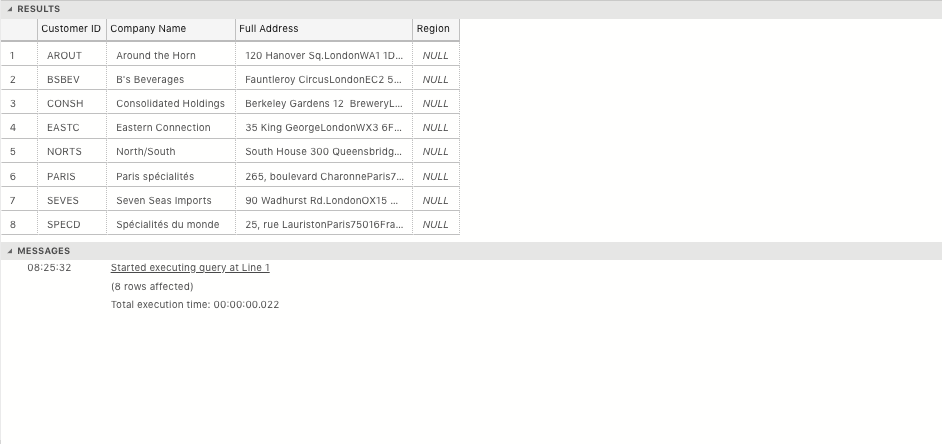
use Northwind;

-- 1.1

-- -- Select CustomerID AS 'Customer ID', CompanyName AS 'Company Name ', Address + City + Postalcode + Country AS 'Full Address', Region

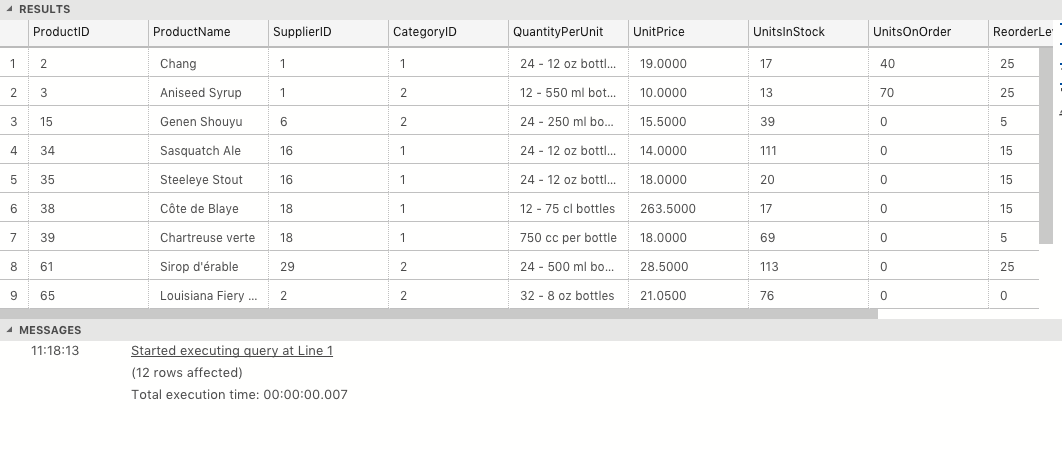
-- FROM Customers

-- WHERE City = 'Paris' OR City = 'LONDON';



-- 1.2

-- SELECT \* FROM Products WHERE QuantityPerUnit LIKE '%bottle%';



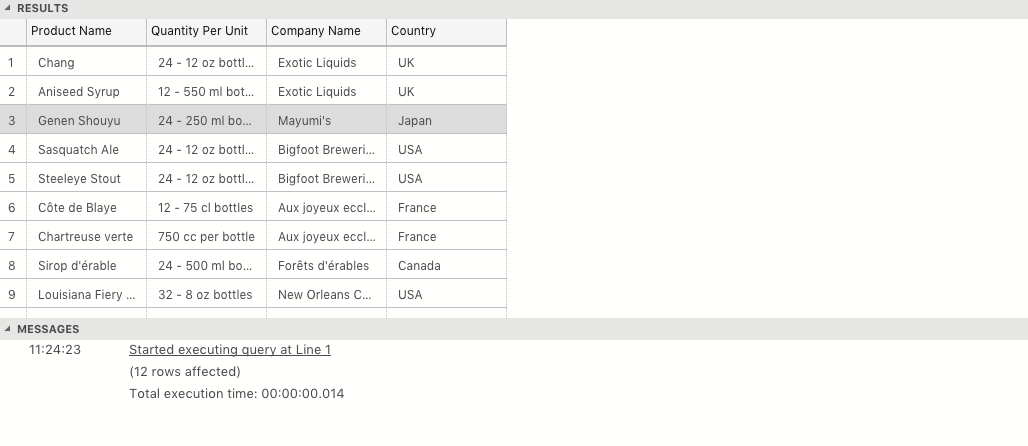
-- 1.3

-- SELECT p.ProductName AS 'Product Name', p.QuantityPerUnit AS 'Quantity Per Unit', s.CompanyName AS 'Company Name', s.Country AS 'Country' FROM Products p

-- INNER JOIN Suppliers s

-- ON p.SupplierID = s.SupplierID

-- WHERE QuantityPerUnit LIKE '%bottle%';



-- 1.4

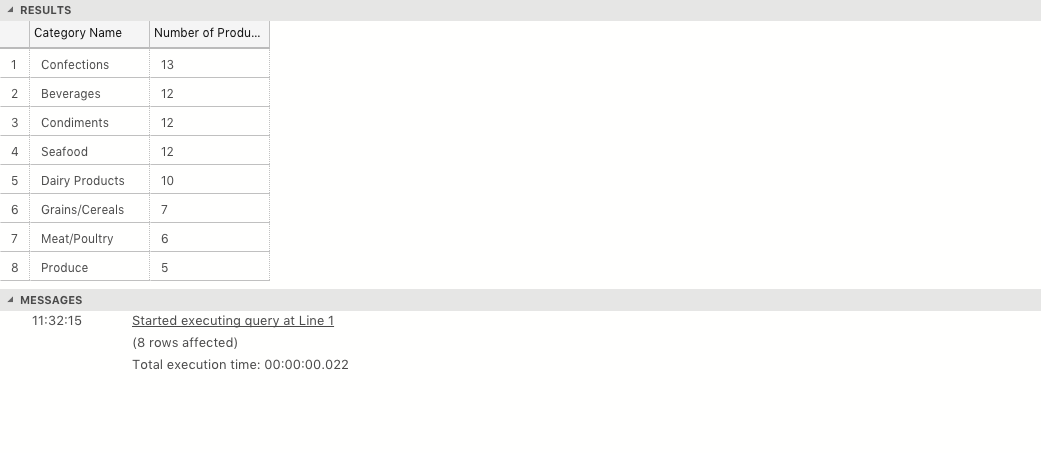
-- SELECT c.CategoryName AS 'Category Name' , COUNT(p.CategoryID) AS 'Number of Products in Category' FROM Products p

-- INNER JOIN Categories c

-- ON p.CategoryID = c.CategoryID

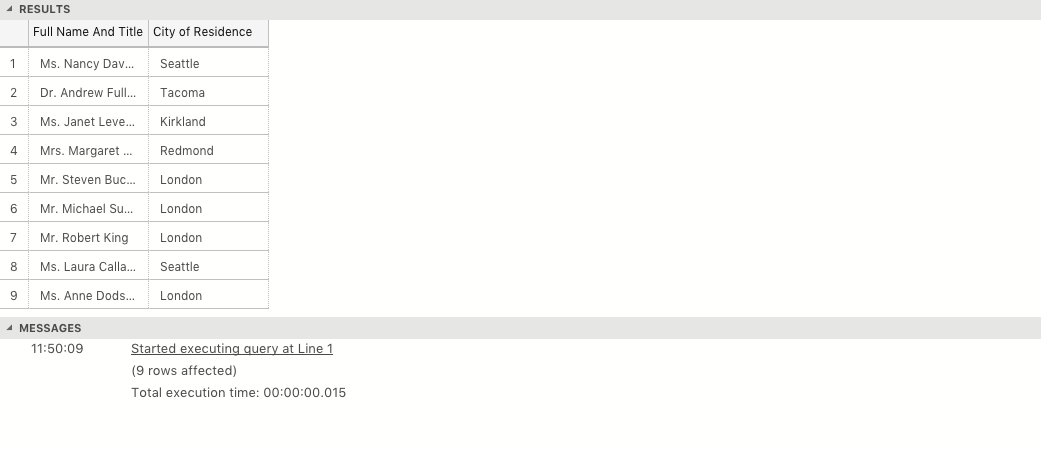
-- GROUP BY p.CategoryID,c.CategoryName

-- ORDER BY COUNT(p.CategoryID) DESC;



-- 1.5

-- SELECT TitleOfCourtesy + ' ' + FirstName + ' ' + LastName AS 'Full Name And Title', City AS 'City of Residence' FROM Employees;



-- 1.6

-- SELECT DISTINCT Region.RegionDescription AS 'Regions', FORMAT((SUM([Order Details].UnitPrice\*[Order Details].Quantity)), 'C', 'us-US') AS 'Sales Total'

-- FROM Territories

-- INNER JOIN Region

-- ON Region.RegionID = Territories.RegionID

-- INNER JOIN EmployeeTerritories

-- ON EmployeeTerritories.TerritoryID = Territories.TerritoryID

-- INNER JOIN Orders

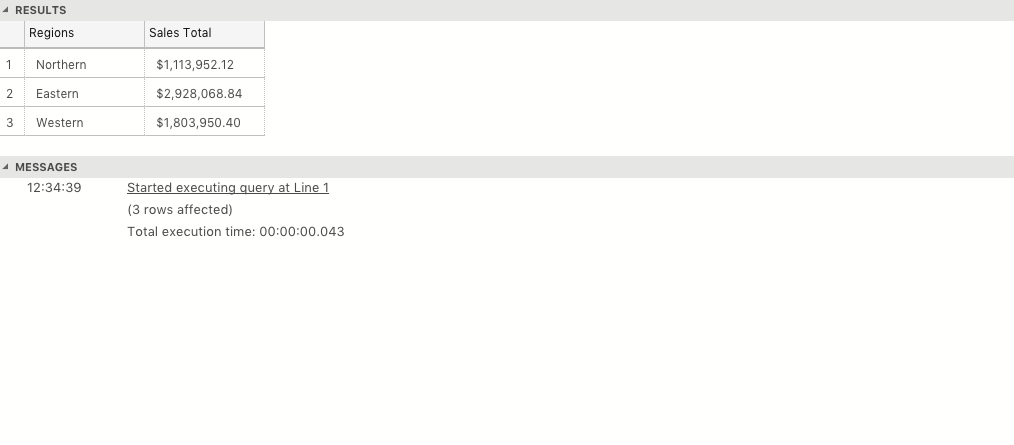
-- ON EmployeeTerritories.EmployeeID = Orders.EmployeeID

-- INNER JOIN [Order Details]

-- ON Orders.OrderID = [Order Details].[OrderID]

-- GROUP BY Region.RegionDescription

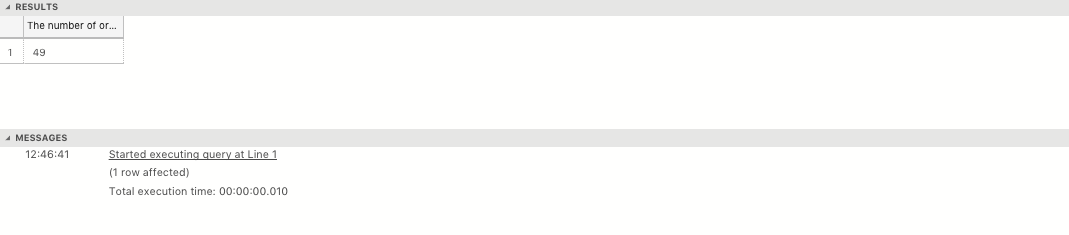
-- HAVING SUM([Order Details].UnitPrice\*[Order Details].Quantity) > 1000000;



--1.7

-- SELECT COUNT(\*) FROM Orders

-- WHERE Freight > 100 AND (ShipCountry = 'UK' OR ShipCountry = 'USA');



--1.8

-- SELECT TOP 1 OrderID AS 'Order ID', FORMAT(SUM(UnitPrice \* Quantity \* Discount), 'C', 'us-US') AS 'Total discount applied' FROM [Order Details]

-- GROUP BY OrderID, Discount

-- ORDER BY SUM(UnitPrice \* Quantity \* Discount) DESC;



EXERCISE 2

2.1

-- CREATE TABLE eng\_29 (

-- student\_id INT NOT NULL IDENTITY PRIMARY KEY,

-- title VARCHAR(5) NOT NULL,

-- first\_name VARCHAR(255) NOT NULL,

-- last\_name VARCHAR(255) NOT NULL,

-- university VARCHAR(255) NOT NULL,

-- course\_taken VARCHAR(255) NOT NULL,

-- mark\_acheived VARCHAR(255)

-- );

2.2

-- INSERT INTO eng\_29 (title, first\_name, last\_name, university, course\_taken, mark\_acheived)

-- VALUES ('Miss.', 'Maroua', 'Akkari', 'Queen Mary', 'Bioinformatics', 'First' ),

-- ('Mr', 'Ben', 'Owusu', 'London South Bank', 'Business Management With Finance', NULL ),

-- ('Mr.', 'Philip', 'Faboya', 'Queen Mary', 'Chemical Engineering', NULL ),

-- ('Mr.', 'Arthur', 'Hussey', 'Oxford', 'Materials Science', 'First' ),

-- ('Mr.', 'Taher', 'Khan', 'London South Bank', 'Information Technology', 'First' ),

-- ('Mr.', 'Aaron', 'Leslie', 'Queen Mary', 'Maths', NULL ),

-- ('Mr.', 'Christopher', 'Baker', 'UCL', 'Digital Humanities', 'First' ),

-- ('Mr.', 'Robert', 'Teall', 'Leeds Beckett', 'Sound And Music For Interactive Games', NULL ),

-- ('Miss.', 'Qamar', 'Aden', 'Portsmouth', 'Petroleum Engineering', NULL ),

-- ('Mr.', 'James', 'Bachen', 'Bournemouth', 'Music And Audio Technology', NULL ),

-- ('Mr.', 'Sebastion', 'Van Woerkom', 'Kent', 'English', NULL );

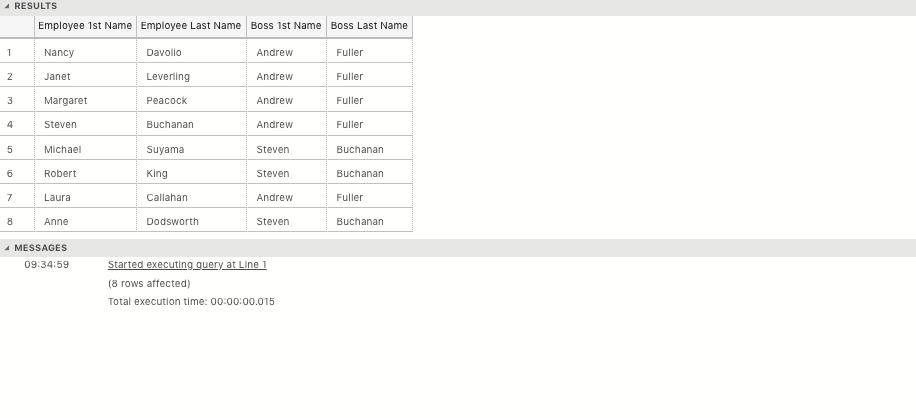


EXERCISE 3

-- SELECT a.FirstName AS 'Employee 1st Name', a.LastName 'Employee Last Name', b.FirstName AS 'Boss 1st Name', b.LastName AS 'Boss Last Name' FROM Employees a, Employees b

-- WHERE b.EmployeeID <> a.EmployeeID

-- AND b. EmployeeID = a.ReportsTo;



--3.2

-- SELECT s.CompanyName, SUM(o.UnitPrice \* o.Quantity ) FROM Suppliers s

-- INNER JOIN Products p

-- ON s.SupplierID = p.SupplierID

-- INNER JOIN [Order Details] o

-- ON p.ProductID = o.ProductID

-- GROUP BY s.CompanyName

-- HAVING SUM(o.UnitPrice \* o.Quantity ) > 10000

-- ORDER BY SUM(o.UnitPrice \* o.Quantity );

-- 3.3

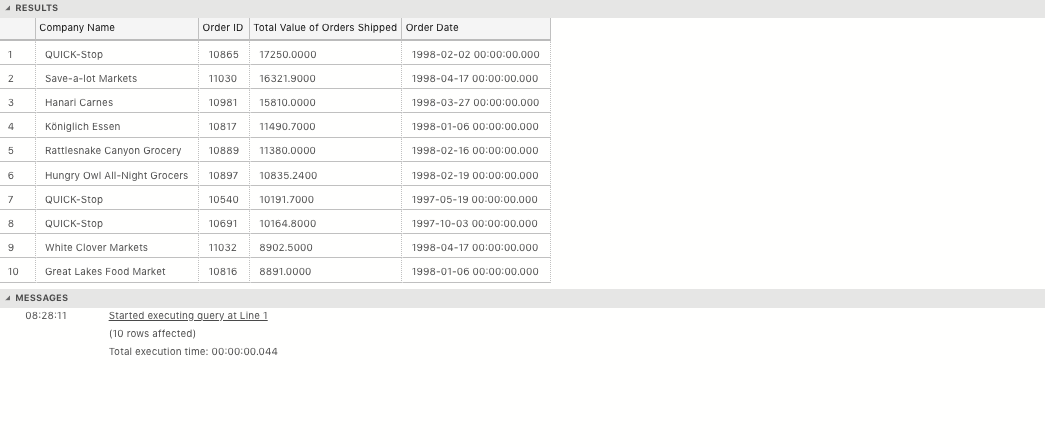
-- SELECT TOP 10 c.CompanyName AS 'Company Name', o.OrderID AS 'Order ID', SUM(od.UnitPrice \* od.Quantity) AS 'Total Value of Orders Shipped', o.OrderDate AS 'Order Date' FROM Orders o

-- INNER JOIN [Order Details] od ON o.OrderID = od.OrderID

-- INNER JOIN Customers c ON o.CustomerID = c.CustomerID

-- GROUP BY c.CompanyName, o.OrderID, o.OrderDate HAVING o.OrderDate > DATEADD(year, -1, '1998/05/06')

-- ORDER BY SUM(od.UnitPrice \* od.Quantity) DESC;



--3.4

-- SELECT MONTH(ShippedDate), YEAR(ShippedDate), AVG(DATEDIFF (dd, OrderDate, ShippedDate)) FROM Orders

-- GROUP BY MONTH(ShippedDate), YEAR(ShippedDate)

-- ORDER BY MONTH(ShippedDate), YEAR(ShippedDate);

­­­­­­­­