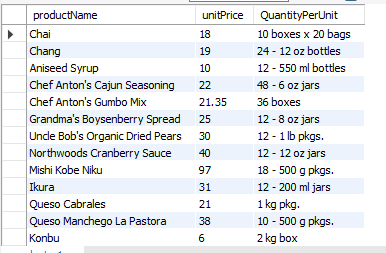
#1.Basic SELECT, WHERE, ORDER BY

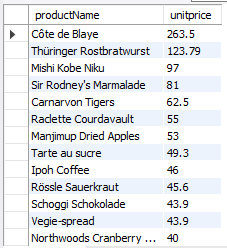
#List all products with their names, unit price, and units in stock.



#Show all customers from Germany, ordered by CompanyName.



#Retrieve the top 10 most expensive products (ProductName, UnitPrice), ordered by UnitPrice descending.

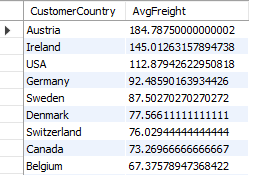


#2. GROUP BY + Aggregates

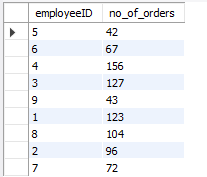
#Find the total number of customers per country.



#Show the average freight (AVG(Freight)) per ship country.

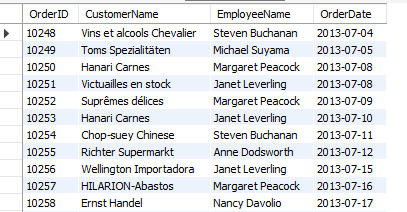


#For each employee, calculate the total number of orders they handled.



#3. JOINS

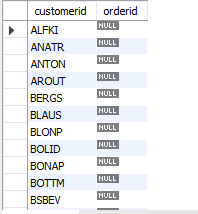
#List all orders with the customer’s company name and the employee’s full name. (INNER JOIN Orders, Customers, Employees)



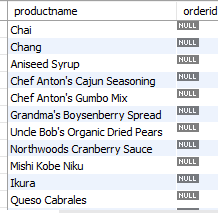
#Show all products and their category names (INNER JOIN Products, categories).



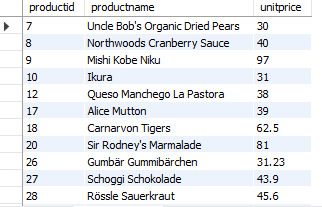
#List all customers and their orders (LEFT JOIN to include customers with no orders).



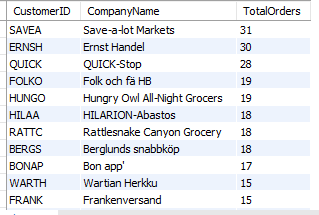
#Find products that have never been ordered (RIGHT JOIN / LEFT JOIN + IS NULL).



#Find all products that are more expensive than the average product price.



#List customers who have placed more orders than the average number of orders per customer.

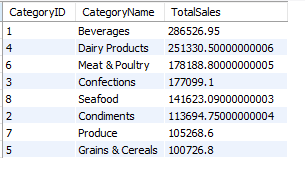


#Retrieve the employee(s) with the highest total sales amount (use OrderDetails.UnitPrice \* Quantity).

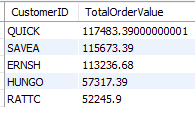


#5. Aggregates with Joins

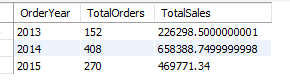
#Calculate the total sales per category (join Products, Categories, and OrderDetails).



#Find the top 5 customers by total order value.

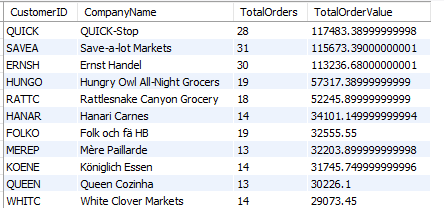


#For each year, compute the total number of orders and the total sales amount.

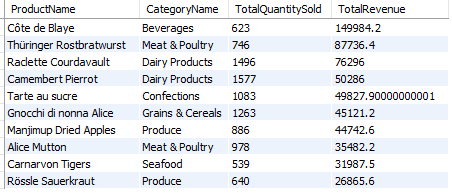


#6. Views

#Create a view CustomerOrderSummary showing: CustomerID, CompanyName, TotalOrders, TotalOrderValue.



#Create a view TopProductsBySales showing: ProductName, CategoryName, TotalQuantitySold, TotalRevenue.



#Create a view EmployeePerformance with EmployeeName, NumberOfOrders, TotalSales.

