**PROG8080**

**PROGRAMMING: DATABASE MANAGEMENT**

**Assignment: 5**

**Name:** Shravani Gandla

**Student Number:** 8697289

1. We need to know the number of products we have in the PurchaseOrderDetailtable. (count the number of un-repeated productid).

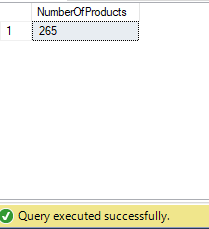
**Query:**

SELECT COUNT (DISTINCT ProductID) AS NumberOfProducts FROM Purchasing.PurchaseOrderDetail ;

**Description:**

The above query shows that, the ProductID column is selected from the Purchasing.PurcgaseOrderDetail table. The Distinct is used to extract the different values of product id instead of selecting all the same product id’s and also the count is used to add all the extracted product id’s.

**Output:**



1. Write a query to show the productID of the most profitable product(ignore

production costs) after price and order quantity are considered(maximum amount of money gained for each product id).

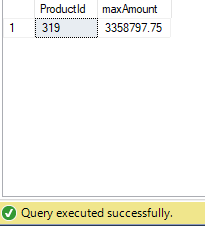
**Query:**

SELECT TOP 1 ProductID AS ProductId, SUM(UnitPrice\*OrderQty) AS maxAmount FROM Purchasing.PurchaseOrderDetail Group by ProductID ORDER BY maxAmount DESC;

**Description:**

The above query shows that, the ProductID is selected from the Purchasing.PurchaseOrderDetail table and returned as ProductId using alias ‘AS’. Here we need the maximum profit so ,I have multiplied the unitPrice ,OrderQty columns and summed them by using “SUM” ,Grouped them by using “GROUP BY “. I have used “ORDER BY DESC” to get all the maximum profits in descending order and then returned the top value among them by using “TOP 1”.

**Output:**



1. Write a query to show the names of the top 5 most profitable products, as in question 2.Remember to take both price and quantity sold into account.

**Query:**

SELECT TOP 5 Production.Product.Name,SUM(UnitPrice\*OrderQty) AS maxAmount FROM Purchasing.PurchaseOrderDetail JOIN Production.Product

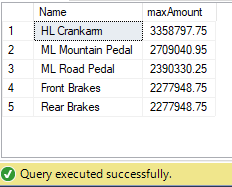
ON Production.Product.ProductID = Purchasing.PurchaseOrderDetail.ProductID

Group by Production.Product.ProductID,Production.Product.Name ORDER BY maxAmount DESC ;

**Description:**

The above query shows that, the Name is selected from the Production.Product table. Here we need the maximum profit so ,I have multiplied the unitPrice ,OrderQty columns and summed them by using “SUM” ,Joinned the two tables by using “JOIN” followed by “ON” clause where equating the ProductIds of both tables to get the matched products Ids . Grouped ProductID from Purchasing.PurchaseOrderDetail and Name column from Production.Product by using “GROUP BY “. I have used “ORDER BY DESC” to get all the maximum profits in descending order and then returned the top 5 values among them by using “TOP 5”.

**Output:**



1. Write a query to show all product ID with the stock quantity less than average stock quantity.

**Query:**

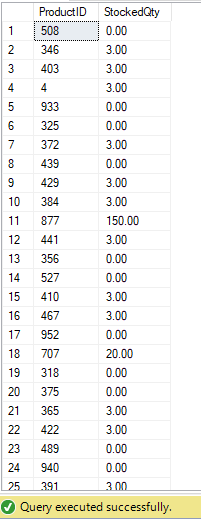
SELECT DISTINCT ProductID,StockedQty FROM Purchasing.PurchaseOrderDetail WHERE

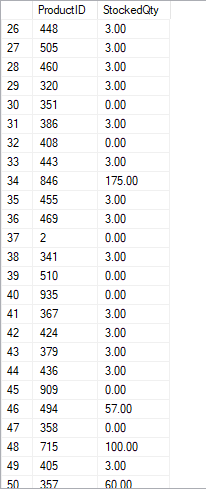
StockedQty <(SELECT Avg(StockedQty) FROM Purchasing.PurchaseOrderDetail);

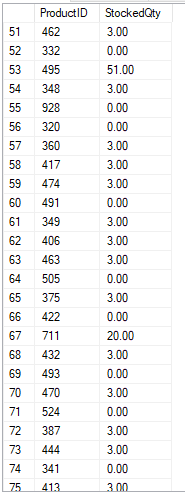
**Description:**

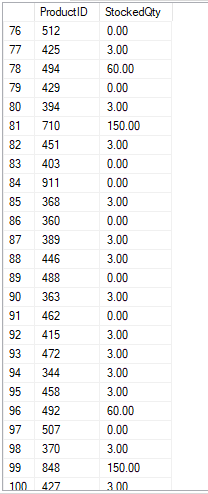
The above query shows that, the ProductID and StockedQty columns are selected from the Purchasing.PurchaseOrderDetail table. The Distinct is used to extract the different values of product id instead of selecting all the same product id’s. Here I used a “WHERE” condition and a sub query to return the stock quantity less than average stock quantity,

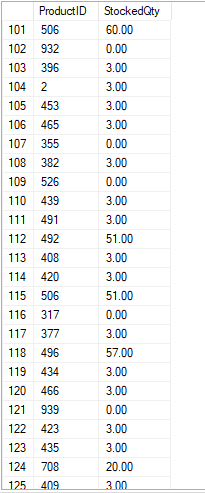
**Output:**

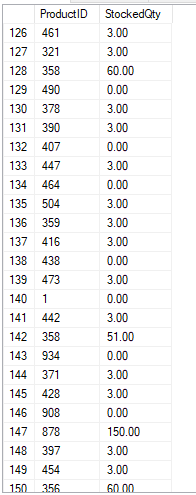


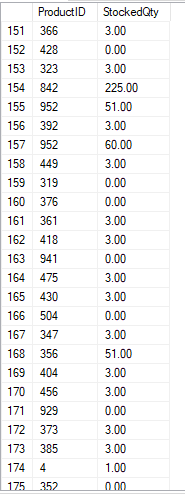


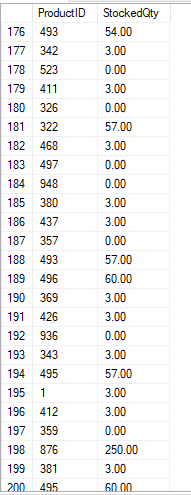


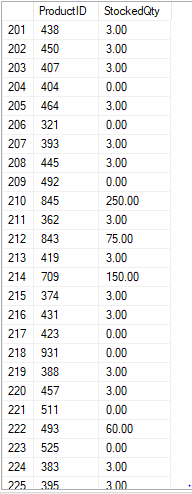


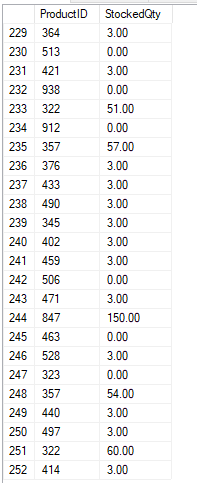












1. We need to know the product id and the modified date of the products with special offer “Half-Price Pedal Sale”.

**Query:**

SELECT ProductID,Sales.SpecialOffer.ModifiedDate FROM Sales.SpecialOfferProduct JOIN Sales.SpecialOffer ON Sales.SpecialOffer.SpecialOfferID = Sales.SpecialOfferProduct.SpecialOfferID AND Sales.SpecialOffer.Description = 'Half-Price Pedal sale';

**Description:**

The above query shows that, the ProductID column ,modifiedDate is selected from the Sales.SpecialOfferProduct, Sales.SpecialOffer tables. I have joined these two tables using “JOIN” and an “ON” clause to compare both SpecialofferID’s from both the tables by Sales.SpecialOffer.SpecialOfferID = Sales.SpecialOfferProduct.SpecialOfferID and also “AND “ to get the half price pedal sale from the Description column.

**Output:**

