You can find this on my github: <https://github.com/princesscorn/assignments-antra.git>

# Write queries for following scenarios

Using AdventureWorks2019

## 1.  How many products can you find in the Production.Product table?

Solution:

SELECT COUNT(ProductID) AS [Number of Products] FROM Production.Product

## 2.  Write a query that retrieves the number of products in the Production.Product table that are included in a subcategory. The rows that have NULL in column ProductSubcategoryID are considered to not be a part of any subcategory.

Solution:

SELECT COUNT(ProductID) AS "Number of Products in A Category"

FROM Production.Product p

WHERE p.ProductSubcategoryID IS NOT NULL;

## 3.  How many Products reside in each SubCategory? Write a query to display the results with the following titles.

ProductSubcategoryID CountedProducts

-------------------- ---------------

Solution:

SELECT ProductSubcategoryID, COUNT(ProductID) AS [CountedProducts]

FROM Production.Product p

WHERE p.ProductSubcategoryID IS NOT NULL

GROUP BY p.ProductSubcategoryID;

## 4.  How many products that do not have a product subcategory.

Solution:

SELECT COUNT(ProductID) AS [NumOfProduct]

FROM Production.Product

WHERE ProductSubcategoryID IS NULL;

## 5.  Write a query to list the sum of products quantity in the Production.ProductInventory table.

Solution:

SELECT ProductID, SUM(Quantity) AS 'Summary of Products'

FROM Production.ProductInventory

GROUP BY ProductID

ORDER BY 'Summary of Products' DESC

## 6.  Write a query to list the sum of products in the Production.ProductInventory table and LocationID set to 40 and limit the result to include just summarized quantities less than 100.

              ProductID    TheSum

              -----------        ----------

Solution:

SELECT ProductID, SUM(Quantity) AS 'Sum of Products'

FROM Production.ProductInventory

WHERE LocationID = 40

GROUP BY ProductID

HAVING SUM(Quantity) < 100

## 7.  Write a query to list the sum of products with the shelf information in the Production.ProductInventory table and LocationID set to 40 and limit the result to include just summarized quantities less than 100

    Shelf      ProductID    TheSum

----------   -----------        -----------

Solution:

SELECT Shelf, ProductID, SUM(Quantity) AS 'TheSum'

FROM Production.ProductInventory

WHERE LocationID = 40

GROUP BY Shelf, ProductID

HAVING SUM(Quantity) < 100

## 8.  Write the query to list the average quantity for products where column LocationID has the value of 10 from the table Production.ProductInventory table.

Solution:

SELECT ProductID, AVG(Quantity) AS 'TheAvg'

FROM Production.ProductInventory

WHERE LocationID = 10

GROUP BY ProductID

## 9.  Write query to see the average quantity of products by shelf from the table Production.ProductInventory

    ProductID   Shelf      TheAvg

----------- ---------- -----------

Solution:

SELECT ProductID, Shelf, AVG(Quantity) AS 'TheAvg'

FROM Production.ProductInventory

GROUP BY Shelf, ProductID

## 10.  Write query to see the average quantity of products by shelf excluding rows that has the value of N/A in the column Shelf from the table Production.ProductInventory

    ProductID   Shelf      TheAvg

----------- ---------- -----------

Solution:

SELECT ProductID, Shelf, AVG(Quantity) AS 'TheAvg'

FROM Production.ProductInventory

WHERE Shelf <> 'N/A'

GROUP BY Shelf, ProductID

## 11.  List the members (rows) and average list price in the Production.Product table. This should be grouped independently over the Color and the Class column. Exclude the rows where Color or Class are null.

    Color                        Class              TheCount          AvgPrice

    -------------- - -----    -----------            ---------------------

Solution:

SELECT Color, Class, Count(\*) AS TheCount, AVG(ListPrice) AS AvgPrice

FROM Production.Product

WHERE Color IS NOT NULL AND Class IS NOT NULL

GROUP BY Color, Class;

**Joins:**

## 12.   Write a query that lists the country and province names from person. CountryRegion and person. StateProvince tables. Join them and produce a result set similar to the following.

    Country                        Province

---------                          ----------------------

Solution:

SELECT c.Name AS Country, s.Name AS Province

FROM Person.CountryRegion c INNER JOIN Person.StateProvince s

ON c.CountryRegionCode = s.CountryRegionCode

## 13.  Write a query that lists the country and province names from person. CountryRegion and person. StateProvince tables and list the countries filter them by Germany and Canada. Join them and produce a result set similar to the following.

    Country                        Province

---------                          ----------------------

Solution:

SELECT c.Name AS Country, s.Name AS Province

FROM Person.CountryRegion c INNER JOIN Person.StateProvince s

ON c.CountryRegionCode = s.CountryRegionCode

WHERE c.Name IN ('Germany', 'Canada')

## **Using Northwnd Database: (Use aliases for all the Joins)**

## 14.  List all Products that has been sold at least once in last 25 years.

Solution:

SELECT DISTINCT p.ProductID, p.ProductName

FROM Orders o JOIN [Order Details] od ON o.OrderID = od.OrderID

JOIN Products p ON od.ProductID = p.ProductID

WHERE DATEDIFF(year, o.OrderDate, GETDATE())< 25;

## 15.  List top 5 locations (Zip Code) where the products sold most.

Solution:

SELECT TOP 5 o.ShipPostalCode, SUM(od.Quantity) AS TotalQuantity

FROM Orders o JOIN [Order Details] od ON o.OrderID = od.OrderID

WHERE o.ShipPostalCode IS NOT NULL

GROUP BY o.ShipPostalCode

ORDER BY TotalQuantity DESC

## 16.  List top 5 locations (Zip Code) where the products sold most in last 25 years.

Solution:

SELECT TOP 5 o.ShipPostalCode, SUM(od.Quantity) AS TotalQuantity

FROM Orders o JOIN [Order Details] od ON o.OrderID = od.OrderID

WHERE (o.ShipPostalCode IS NOT NULL) AND DATEDIFF(year, o.OrderDate, GETDATE())< 25

GROUP BY o.ShipPostalCode

ORDER BY TotalQuantity DESC

## 17.   List all city names and number of customers in that city.

Solution:

SELECT c.City, COUNT(c.CustomerID) AS NumOfCustomer FROM Customers c

GROUP BY c.City

## 18.  List city names which have more than 2 customers, and number of customers in that city

Solution:

SELECT City, COUNT(CustomerID) AS NumOfCustomer FROM Customers

GROUP BY City HAVING COUNT(CustomerID) > 2

## 19.  List the names of customers who placed orders after 1/1/98 with order date.

Solution:

SELECT DISTINCT c.CustomerID, c.ContactName, c.CompanyName

FROM Customers c INNER JOIN Orders o ON c.CustomerID = o.CustomerID

WHERE o.OrderDate > '1998-1-1';

## 20.  List the names of all customers with most recent order dates

Solution:

SELECT c.ContactName, MAX(o.OrderDate) AS MostRecentOrderDate

FROM Customers c LEFT JOIN Orders o ON c.CustomerId = o.CustomerId

GROUP BY c.ContactName

## 21.  Display the names of all customers along with the count of products they bought

Solution:

SELECT c.CustomerID, c.CompanyName, c.ContactName, SUM(od.Quantity) AS QTY

FROM Customers c LEFT JOIN Orders o ON c.CustomerID = o.CustomerID

LEFT JOIN [Order Details] od ON o.OrderID = od.OrderID

GROUP BY c.CustomerID, c.CompanyName, c.ContactName

ORDER BY QTY;

## 22.  Display the customer ids who bought more than 100 Products with count of products.

Solution:

SELECT c.CustomerID,

SUM(od.Quantity) AS QTY FROM

Customers c

LEFT JOIN

Orders o

ON c.CustomerID = o.CustomerID

LEFT JOIN

[Order Details] od

ON o.OrderID = od.OrderID

GROUP BY c.CustomerID

HAVING SUM(od.Quantity) > 100

ORDER BY QTY;

## 23.  List all of the possible ways that suppliers can ship their products. Display the results as below

    Supplier Company Name                Shipping Company Name

---------------------------------            ----------------------------------

Solution:

SELECT DISTINCT sup.CompanyName, ship.CompanyName

FROM Orders o LEFT JOIN [Order Details] od ON o.OrderID = od.OrderID

INNER JOIN Products p ON od.ProductID = p.ProductID

RIGHT JOIN Suppliers sup ON p.SupplierID = sup.SupplierID

INNER JOIN Shippers ship ON o.ShipVia = ship.ShipperID;

## 24.  Display the products order each day. Show Order date and Product Name.

Solution:

SELECT o.OrderDate, p.ProductName FROM

Orders o

LEFT JOIN

[Order Details] od

ON o.OrderID = od.OrderID

INNER JOIN

Products p

ON od.ProductID = p.ProductID

GROUP BY o.OrderDate, p.ProductName

ORDER BY o.OrderDate;

## 25.  Displays pairs of employees who have the same job title.

Solution:

SELECT e1.Title, e1.LastName + ' ' + e1.FirstName AS Name1, e2.LastName + ' ' + e2.FirstName AS Name2

FROM Employees e1

JOIN

Employees e2

ON e1.Title = e2.Title

WHERE e1.FirstName <> e2.FirstName OR e1.LastName <> e2.LastName

ORDER BY Title;

## 26.  Display all the Managers who have more than 2 employees reporting to them.

Solution:

SELECT e1.EmployeeId, e1.LastName, e1.FirstName, e2.ReportsTo, COUNT(e2.ReportsTo) AS Subordinate

FROM Employees e1 JOIN Employees e2 ON e1.EmployeeId = e2.ReportsTo

WHERE e2.ReportsTo IS NOT NULL

GROUP BY e1.EmployeeId, e1.LastName, e1.FirstName, e2.ReportsTo

HAVING COUNT(e2.ReportsTo) > 2

## 27.  Display the customers and suppliers by city. The results should have the following columns

City

Name

Contact Name,

Type (Customer or Supplier)

Solution:

SELECT City, CompanyName, ContactName, 'Type (Customer or Supplier)' = 'Customer'

FROM Customers

UNION

SELECT City, CompanyName, ContactName, 'Type (Customer or Supplier)' = 'Supplier'

FROM Suppliers

ORDER BY City, CompanyName;