Assignment Day2 –SQL: Comprehensive practice

# Answer following questions

1. What is a result set?

A: Result set is what the queries return.

1. What is the difference between Union and Union All?

A: 1. Union remove the duplicates in the result while union all does not.

2. Values in the first column will be stored in union.

3. Union cannot used in recursive cte union all can.

1. What are the other Set Operators SQL Server has?

A: 1. UNION

2. UNION All

3. INTERSECT

4. EXCEPT

1. What is the difference between Union and Join?

A: Join is used to combine columns from different tables, the union is used to combine rows.

1. What is the difference between INNER JOIN and FULL JOIN?

A: Inner join returns only the matching rows between both the tables, non-matching rows are eliminated. Full Join or Full Outer Join returns all rows from both the tables (left & right tables), including non-matching rows from both the tables.

1. What is difference between left join and outer join?

A: I assume the outer join means “left outer join.” And the difference between left join and left outer join is an empty set because they are all return the same result.

1. What is cross join?

A: The SQL CROSS JOIN produces a result set which is the number of rows in the first table multiplied by the number of rows in the second table if no WHERE clause is used along with CROSS JOIN. This kind of result is called as Cartesian Product.

1. What is the difference between WHERE clause and HAVING clause?

A: Where clause if for Select statement which will applies to all the rows in table. Having clause applies to the whole group. WHERE goes before aggregations, HAVING goes after the aggregations.

1. Can there be multiple group by columns?

A: I am not sure what this means, but I guess it is asking for “Can there be multiple group by clauses in one query?”

And my answer is no. Only one group by clauses can exist in one query, but you could sort more than one columns in one query.

# Write queries for following scenarios

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

504.

1. 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.

SELECT count(ProductSubcategoryID)

FROM Production.Product

295.

1. How many Products reside in each SubCategory? Write a query to display the results with the following titles. ProductSubcategoryID CountedProducts

SELECT ProductSubcategoryID, count(ProductSubcategoryID) AS CountedProducts

FROM Production.Product

Group by ProductSubcategoryID

1. How many products that do not have a product subcategory.

504 – 295 = 209. Or

SELECT count(\*)

FROM Production.Product

where ProductSubcategoryID is null

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

335974.

SELECT SUM(Quantity)

FROM Production.ProductInventory

1. 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

SELECT ProductID, SUM(Quantity) AS TheSum

FROM Production.ProductInventory

WHERE LocationID = 40

GROUP BY ProductID

HAVING SUM(Quantity) < 100

1. 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

SELECT shelf, ProductID, SUM(Quantity) AS TheSum

FROM Production.ProductInventory

WHERE LocationID = 40

GROUP BY shelf, ProductID

HAVING SUM(Quantity) < 100

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

SELECT AVG(Quantity) AS TheAvg

FROM Production.ProductInventory

WHERE LocationID = 10

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

ProductID Shelf TheAvg

SELECT ProductId, Shelf, AVG(Quantity) AS TheAvg

FROM Production.ProductInventory

GROUP BY ProductID, Shelf

1. 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

SELECT ProductId, Shelf, AVG(Quantity) AS TheAvg

FROM Production.ProductInventory

WHERE Shelf != 'N/A'

GROUP BY ProductID, Shelf

1. 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

SELECT Color, Class,

COUNT(\*) AS TheCount,

ROUND(AVG(ListPrice),2)

AS AvgPrice

FROM Production.Product

WHERE NOT(Color IS NULL OR Class IS NULL)

GROUP BY Color, Class

**Joins:**

1. 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

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

FROM Person.CountryRegion c JOIN Person.StateProvince s

ON c.CountryRegionCode = s.CountryRegionCode

1. 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

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

FROM Person.CountryRegion c JOIN Person.StateProvince s

ON c.CountryRegionCode = s.CountryRegionCode

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

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

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

SELECT DISTINCT ProductName

FROM Products p join [Order Details] od

ON p.ProductID = od.ProductID

JOIN Orders o

ON od.OrderID = o.OrderID

WHERE YEAR(GETDATE()) - YEAR(o.OrderDate) <= 25

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

SELECT TOP 5 ShipPostalCode

FROM Orders o JOIN [Order Details] od ON o.OrderID = od.OrderID JOIN Products P ON OD.ProductID = P.ProductID

WHERE O.ShipPostalCode IS NOT NULL

GROUP BY ShipPostalCode

Order BY sum(quantity) DESC

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

SELECT TOP 5 ShipPostalCode

FROM Products p JOIN [Order Details] od

ON p.ProductID = od.ProductID

JOIN Orders o

ON od.OrderID = o.OrderID

WHERE YEAR(GETDATE()) - YEAR(o.OrderDate) <= 25

AND O.ShipPostalCode IS NOT NULL

GROUP BY ShipPostalCode

Order BY sum(quantity) DESC

1. List all city names and number of customers in that city.

SELECT CITY, COUNT(CustomerID) AS NumOfCustomers

FROM Customers

GROUP BY City

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

SELECT CITY, COUNT(CustomerID) AS NumOfCustomers

FROM Customers

GROUP BY City

HAVING COUNT(CustomerID) > 2

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

SELECT DISTINCT c.ContactName

FROM Orders o JOIN Customers c ON

o.CustomerID = c.CustomerID

WHERE YEAR(o.OrderDate) >= 1998

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

SELECT c.ContactName, MAX(o.OrderDate)

FROM Orders o JOIN Customers c ON

o.CustomerID = c.CustomerID

GROUP BY c.ContactName

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

SELECT c.ContactName, SUM(Quantity)

AS CountOfProducts

FROM [Order Details] od JOIN Orders o

ON od.OrderID = o.OrderID JOIN

Customers c ON

o.CustomerID = c.CustomerID

GROUP BY c.ContactName

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

SELECT o.CustomerID, SUM(Quantity)

AS CountOfProducts

FROM [Order Details] od JOIN Orders o

ON od.OrderID = o.OrderID

GROUP BY o.CustomerID

HAVING SUM(Quantity) > 100

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

Supplier Company Name Shipping Company Name

SELECT DISTINCT S.CompanyName

AS "Supplier Company Name",

SH.CompanyName AS

"Shipping Company Name"

FROM Suppliers S JOIN Products P

ON S.SupplierID = P.SupplierID JOIN

[Order Details] OD ON

P.ProductID = OD.ProductID JOIN Orders O

ON OD.OrderID = O.OrderID JOIN Shippers SH

ON O.ShipVia = SH.ShipperID

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

SELECT O.OrderDate, P.ProductName

FROM ORDERS O JOIN [Order Details] OD

ON O.OrderID = OD.OrderID JOIN

Products P ON OD.ProductID =

P.ProductID

1. Displays pairs of employees who have the same job title.

SELECT A.FirstName, A.LastName, B.FirstName, B.LastName

FROM Employees A JOIN Employees B ON A.Title = B.Title

WHERE A.FirstName != B.FirstName AND A.LastName != B.LastName

AND A.FirstName > B.FirstName

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

SELECT A.FirstName, A.LastName

FROM Employees A JOIN Employees B

ON A.EmployeeID = B.ReportsTo

GROUP BY A.FirstName, A.LastName

HAVING COUNT (A.FirstName ) > 2

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

City

Name

Contact Name,

Type (Customer or Supplier)

SELECT City, CompanyName, ContactName, 'Supplier' AS Type

FROM Suppliers

UNION

SELECT City, CompanyName, ContactName, 'Customer' AS Type

FROM Customers