Assignment Day2 –SQL: Comprehensive practice

# Answer following questions

1. What is a result set?

* A result set is a set of data returned by a select statement from the database, and it is saved in RAM or displayed on the screen.

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

* The difference is that Union extracts the rows that are being specified in the query while Union All extracts all the rows including the duplicates from both the queries.

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

* SQL Server also has Intersect and Except Set Operators.

1. What is the difference between Union and Join?

* Join is used to combine columns from different tables, whereas Union is used to combine rows using two or more SELECT statement.

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

* INNER JOIN returns only the matching rows between two tables, non-matching rows are eliminated. FULL JOIN returns all rows from both tables, including non-matching rows.

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

* Left Join returns all rows from the left table plus the rows that the right table had in common. Full Outer Join returns all the rows from both tables, including non-matching rows.

1. What is cross join?

* Cross Join returns the Cartesian product of the sets of records from the two joined tables.

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

* WHERE clause is used in the selection of rows according to given conditions whereas HAVING clause is used in column operations and is applied to aggregated rows.

1. Can there be multiple group by columns?

* Yes, GROUP BY clause can contain two or more columns, which means the result is grouped on two or more columns.

# Write queries for following scenarios

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

SELECT COUNT(1) AS 'TotalProduct' FROM Production.Product

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) AS 'TotalSubcategoryProduct'

FROM Production.Product

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.

SELECT count(1) - count(ProductSubcategoryID) AS 'CountedNonSubcategoryProducts'

FROM Production.Product

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

SELECT ProductID, SUM(Quantity)

FROM Production.ProductInventory

GROUP BY ProductID

1. Write a query to list the summary 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 summary 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 ProductID, AVG(Quantity) AS Average

FROM Production.ProductInventory

WHERE LocationID = 10

GROUP BY ProductID

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 Average

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 Average

FROM Production.ProductInventory

WHERE Shelf NOT like '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(ListPrice) AS TheCount, AVG(ListPrice) AS AvgPrice

FROM Production.Product

WHERE Color is NOT NULL AND Class is NOT 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, s.Name

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, s.Name

FROM Person.CountryRegion c JOIN Person.StateProvince s

ON c.CountryRegionCode = s.CountryRegionCode

WHERE c.Name like 'Canada' OR c.Name like 'Germany'

**Using Northwnd 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 JOIN [Order Details]

ON Products.ProductID = [Order Details].ProductID

JOIN Orders

ON [Order Details].OrderID = Orders.OrderID

WHERE OrderDate > '1996-01-01'

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

SELECT TOP 5 Orders.ShipPostalCode, COUNT(Orders.ShipPostalCode) AS CountProducts

FROM Products JOIN [Order Details]

ON Products.ProductID = [Order Details].ProductID

JOIN Orders

ON [Order Details].OrderID = Orders.OrderID

Group By Orders.ShipPostalCode

Order By CountProducts

DESC

1. List top 5 locations (Zip Code) where the products sold most in last 20 years.

SELECT TOP 5 Orders.ShipPostalCode, COUNT(Orders.ShipPostalCode) AS CountProducts

FROM Products JOIN [Order Details]

ON Products.ProductID = [Order Details].ProductID

JOIN Orders

ON [Order Details].OrderID = Orders.OrderID

WHERE Orders.OrderDate > '2001-01-01'

Group By Orders.ShipPostalCode

Order By CountProducts

DESC

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

SELECT City, COUNT(1)

FROM Customers

GROUP BY City

1. List city names which have more than 10 customers, and number of customers in that city

SELECT City, COUNT(1) AS CountCustomers

FROM Customers

GROUP BY City

HAVING COUNT(1) > 10

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

SELECT ContactName, OrderDate

FROM Customers JOIN Orders

ON Customers.CustomerID = Orders.CustomerID

WHERE OrderDate > '1998-01-01'

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

SELECT ContactName, OrderDate

FROM Customers JOIN Orders

ON Customers.CustomerID = Orders.CustomerID

Order By OrderDate

DESC

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

SELECT ContactName, COUNT(OrderID)

FROM Customers LEFT OUTER JOIN Orders

ON Customers.CustomerID = Orders.CustomerID

GROUP BY ContactName

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

SELECT c.CustomerID, COUNT(OrderID)

FROM Customers c LEFT OUTER JOIN Orders

ON c.CustomerID = Orders.CustomerID

GROUP BY c.CustomerID

HAVING COUNT(OrderID) > 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 Suppliers.CompanyName, Shippers.CompanyName

FROM Suppliers CROSS JOIN Shippers

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

SELECT ProductName, OrderDate

FROM Products JOIN [Order Details]

ON Products.ProductID = [Order Details].ProductID

JOIN Orders

ON [Order Details].OrderID = Orders.OrderID

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

SELECT a.FirstName, b.FirstName, a.Title

FROM Employees a, Employees b

WHERE a.EmployeeID <> b.EmployeeID

AND a.Title = b.Title

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

SELECT a.FirstName, Count(b.FirstName)

FROM Employees a, Employees b

WHERE a.EmployeeID = b.ReportsTo

GROUP BY a.FirstName

HAVING Count(b.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 c.City, c.ContactName

FROM Customers c FULL OUTER JOIN Suppliers s

ON c.City = s.City AND c.ContactName = s.ContactName

WHERE c.City is NOT NULL AND c.ContactName is NOT NULL

28. Have two tables T1 and T2

|  |  |
| --- | --- |
| F1.T1 | F2.T2 |
| 1 | 2 |
| 2 | 3 |
| 3 | 4 |

Please write a query to inner join these two tables and write down the result of this query.

SELECT T1.id FROM F1.T1 INNER JOIN F2.T2 ON T1.id = T2.id

Result:

|  |
| --- |
| T1.id |
| 2 |
| 3 |

29. Based on above two table, Please write a query to left outer join these two tables and write down the result of this query.

SELECT T1.id FROM F1.T1 LEFT OUTER JOIN F2.T2 ON T1.id = T2.id

Result:

|  |
| --- |
| T1.id |
| 1 |
| 2 |
| 3 |

GOOD LUCK.