Assignment Day2 –SQL: Comprehensive practice

Zehua Pan

# Answer following questions

1. What is a result set?

An SQL result set is a set of rows from a database, as well as metadata about the query such as the column names, and the types and sizes of each column.

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

* Union drops duplicate records while Union All keeps duplicates.
* Union will sort the results set based on the first column of the first select statement.
* Union performs slower than Union All due to the sorting.
* Union cannot be used with recursive CTE, but Union All can be used with recursive CTE

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

INTERSECT returns any distinct values that are returned by both the query on the left and right sides of the INTERSECT operand.

EXCEPT query returns all rows which are in the first query but those are not returned in the second query.

1. What is the difference between Union and Join?

UNION in SQL is used to combine the result-set of two or more SELECT statements. The data combined using UNION statement is into results into new distinct rows. JOIN combines data from many tables based on a matched condition between them. It combines data into new columns.

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

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?

Left join is one of outer join. Outer join has three types: Left Outer Join, Right Outer Join, and Full Outer Join

1. What is cross join?

The CROSS JOIN is used to generate a paired combination of each row of the first table with each row of the second table. This join type is also known as cartesian join.

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

The difference between WHERE and HAVING clause are: The WHERE clause is used to filter rows before the grouping is performed. The HAVING clause is used to filter rows after the grouping is performed. It often includes the result of aggregate functions and is used with GROUP BY.

1. Can there be multiple group by columns?

Yes.

# Write queries for following scenarios

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

504

SELECT COUNT(\*) 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.

295

SELECT COUNT(\*)

FROM Production.Product

WHERE ProductSubcategoryID IS NOT NULL

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

ProductSubcategoryID CountedProducts

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

SELECT ProductSubcategoryID, COUNT(\*) AS 'CountedProducts'

FROM Production.Product

GROUP BY ProductSubcategoryID

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

209

SELECT COUNT(\*)

FROM Production.Product

WHERE ProductSubcategoryID IS NULL

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

SELECT ProductID, Count(\*) AS TheSum

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, Count(\*) AS TheSum

FROM Production.ProductInventory

WHERE LocationID = 40

GROUP BY ProductID

HAVING COUNT(\*) < 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, Count(\*) AS TheSum

FROM Production.ProductInventory

WHERE LocationID = 40

GROUP BY ProductID, Shelf

HAVING COUNT(\*) < 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 TheAvg

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 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 IS NOT NULL

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, 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 AS Country, s.Name AS Province

FROM person.CountryRegion c left 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 left join person.StateProvince s

ON c.CountryRegionCode = s.CountryRegionCode

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

**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 \*

FROM dbo.Products p RIGHT JOIN

(SELECT DISTINCT ProductID

FROM dbo.Orders a LEFT JOIN dbo.[Order Details] b

ON a.OrderID = b.OrderID

WHERE YEAR(a.OrderDate) >= YEAR(GETDATE()) - 25) o

ON p.ProductID = o.ProductID

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

SELECT TOP 5 ShipPostalCode, COUNT(\*) AS SoldCount

FROM dbo.Orders

GROUP BY ShipPostalCode

ORDER BY COUNT(\*) DESC

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

SELECT TOP 5 ShipPostalCode, COUNT(\*) AS SoldCount

FROM dbo.Orders

WHERE YEAR(OrderDate) >= YEAR(GETDATE()) - 20

GROUP BY ShipPostalCode

ORDER BY COUNT(\*) DESC

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

SELECT City, COUNT(\*) AS CustomerCount

FROM dbo.Customers

GROUP BY City

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

SELECT City, COUNT(\*) AS CustomerCount

FROM dbo.Customers

GROUP BY City

HAVING COUNT(\*) > 10

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

SELECT DISTINCT c.ContactName

FROM Orders o LEFT JOIN Customers c

ON o.CustomerID = c.CustomerID

WHERE DATEDIFF(d, o.OrderDate, '1998-01-01') > 0

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

SELECT c.ContactName

FROM Orders o LEFT JOIN Customers c

ON o.CustomerID = c.CustomerID

ORDER BY DATEDIFF(d, o.OrderDate, GETDATE())

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

SELECT c.ContactName, SUM(o.Quantity) AS BoughtCount

FROM

(SELECT c.ContactName, o.OrderID

FROM dbo.Orders o LEFT JOIN dbo.Customers c

ON o.CustomerID = c.CustomerID) c

LEFT JOIN dbo.[Order Details] o

ON c.OrderID = o.OrderID

GROUP BY c.ContactName

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

SELECT c.CustomerID, SUM(o.Quantity) AS BoughtCount

FROM

(SELECT c.CustomerID, o.OrderID

FROM dbo.Orders o LEFT JOIN dbo.Customers c

ON o.CustomerID = c.CustomerID) c

LEFT JOIN dbo.[Order Details] o

ON c.OrderID = o.OrderID

GROUP BY c.CustomerID

HAVING SUM(o.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 a.CompanyName AS 'Supplier Company Name',

b.CompanyName AS 'Shipping Company Name'

FROM dbo.Suppliers a CROSS JOIN dbo.Shippers b

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

SELECT o.OrderDate, p.ProductName

FROM

(SELECT o.OrderID, o.OrderDate, d.ProductID

FROM dbo.Orders o LEFT JOIN dbo.[Order Details] d

ON o.OrderID = d.OrderID) o

LEFT JOIN dbo.Products p

ON o.ProductID = o.ProductID

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

SELECT CONCAT(a.FirstName, ' ', a.LastName) AS employee1,

CONCAT(b.FirstName, ' ', b.LastName) AS employee2,

a.Title

FROM dbo.Employees a INNER JOIN dbo.Employees b

ON a.Title = b.Title

WHERE a.EmployeeID != b.EmployeeID

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

SELECT COUNT(a.EmployeeID) AS NumReportTo,a.EmployeeID, a.LastName, a.FirstName

FROM dbo.Employees a INNER JOIN dbo.Employees b

ON a.EmployeeID = b.ReportsTo

GROUP BY a.EmployeeID, a.LastName, a.FirstName

HAVING COUNT(a.EmployeeID) > 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, 'Customer' AS 'Type (Customer or Supplier)'

FROM dbo.Customers

UNION

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

FROM dbo.Suppliers

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 a.F1

FROM T1 a INNER JOIN T2 b

ON a.F1 = b.F2

the results will be

|F1|

|2 |

|3 |

1. 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 a.F1

FROM T1 a LEFT JOIN T2 b

ON a.F1 = b.F2

the result will be

|F1|

|1 |

|2 |

|3 |

GOOD LUCK.