Assignment Day2 –SQL: Comprehensive practice

# Answer following questions

1. What is a result set?

-Result set is output (rows) of a query which is generated by executing a statement of queries the database.

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

-Union will not have result set including duplicates, but Union All does.

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

There are Intersect, Except, Minus.

-Intersect Operator will return records that exist in both two dataset of two select statements.

-Except Operator will return records that are unique to one result set. It means one query but not in another query.

-Minus Operator combines result that includes only set of result from first table and remove all records from second table.

1. What is the difference between Union and Join?

-In Union, it combines the result set of two or more SELECT statement and data into rows. The number of column and data type selected must be same from each table.

-In Join, it combines data from many tables based on matched condition between them into columns. The number of data and data type selected can be different from each table.

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

-INNER JOIN returns the rows that have matching value in both tables. It is default join and same as JOIN.

-FULL JOIN returns all rows from the joined tables. It is combined ‘LEFT JOIN’ and ‘RIGHT JOIN’. LEFT JOIN returns all records from left table matched records from right table, for non-matched records in right table, return null value. RIGHT JOIN returns all records from right table matched records from left table, for non-matched records in left table, return null value.

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

-LEFT JOIN returns all records from left table matched records from right table, for non-matched records in right table, return null value.

-OUTER JOIN returns unmatched records from both tables.

1. What is cross join?

-CROS JOIN creates the Cartesian product of two tables. There is no join condition.

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

-WHERE clause applies to individual rows and goes before aggregation. It can be used with SELECT, UPDATE.

-HAVING clause only applies to groups as a whole and goes after aggregation. It is only with SELECT.

1. Can there be multiple groups by columns?

-Yes, using GROUP BY multiple columns, we can retrieve the summarized result set from the database using the sql query by defining the grouping criteria on more than one column.

# Write queries for following scenarios

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

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.

select count(ProductSubcategoryID)

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(\*) as ‘CountedProducts’

from production.Product

group by ProductSubcategoryID

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

select count(\*) as ‘noProductSubcategory’

from production.Product

where ProductSubcategoryID is null

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

select ProductID, sum(quantity) as ‘TheSum’

from production.ProductInventory

group by ProductID

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 ProductID, Shelf

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)

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', avg(ListPrice)

from production.Product

group by color, class

having color is not null and class is not null

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

inner 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

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)**

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

select distinct \*

from products p inner join(

select productid

from [order details] od inner join orders o on od.orderid = o.orderid

where datediff(year, o.orderdate, getdate()) <25) ot on p.productid = ot.ProductID

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

select top 5 c.PostalCode

FROM customers c JOIN Orders o ON c.CustomerID = o.CustomerID JOIN [order details] od ON od.OrderID = o.OrderID

where postalcode is not null

group by c.postalcode

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 as 'zip code', count(\*) as 'numofship'

from orders o

where datediff(year, o.orderdate, getdate()) < 25

group by shippostalcode

order by numofship desc

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

select City, count(ContactName) as 'numofCustomer'

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(contactName) as 'numofCustomer'

from Customers

group by city

having count(contactName) > 2

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

select distinct c.ContactName

from customers c inner join Orders o on c.CustomerID =o.CustomerID

where o.OrderDate > '1/1/98'

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

select c.contactname, max(orderdate) as 'recent date'

from orders o inner 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, td.sum

from Customers c inner join (

select CustomerID, sum(quantity) as 'sum'

from [Order Details] od inner join orders o on od.orderid = o.OrderID

group by CustomerID

) td on c.customerid = td.customerid

order by c.ContactName

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

select c.CustomerID, cp.sumofq

from customers c

inner join( select customerid, sum(quantity) as sumofq from [Order Details] od inner join orders o on od.orderid = o.orderid

group by CustomerID) cp

on c.customerID = cp.CustomerID

where sumofq > 100

order by c.customerid

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 su.CompanyName as 'Supplier Company Name', sh.companyname as 'Shipping Company Name'

from Suppliers su cross join shippers sh

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

select o.orderdate, p.productname

from orders o cross join products p

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

select distinct e1.EmployeeID, e2.EmployeeID

from employees e1 inner join employees e2 on e1.title = e2.title

where e1.EmployeeID <> e2.EmployeeID

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

select distinct e1.lastname, e1.firstname

from employees e1 inner join employees e2 on e1.employeeid = e2.reportsto

group by e1.lastname, e1.firstname

having count(e1.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, contactname

from customers

union

select city, contactname

from suppliers

GOOD LUCK.