Assignment Day2 –SQL: Comprehensive practice

# Answer following questions

1. What is a result set?

A set of rows from a database

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

Union All returns all duplicate records, Union removes duplicate records .

Union’s first column sorts the data, Union All can’t

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

Exclude union and union all, there are intersect and minus.

1. What is the difference between Union and Join?

Join combine data into new columns, but unions combine data into new rows.

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

Inner join returns only matched row, but full join returns rows from both tables even if there are no matching rows in the other table.

1. What is difference between left join and outer join

Outer join compares two tables and returns data when a match is available or NULL

values otherwise. But left join keep all records from left table no mater what and insert

NULL values when right table doesn’t match.

1. What is cross join?

Create the Cartesian product of two tables, irrespective of any filter criterial or any condition

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

Where applies to individual rows.

Having applies only to groups as a whole, as only filter aggregated field.

Where is before aggregation

Having after aggregation

Where can be with Select, Update, Delete statements

Having is only with select statement

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?

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(\*) 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)

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)'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)'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) 'TheAvg'

from Production.ProductInventory

group by shelf,Productid

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) from production.ProductInventory

where shelf<> 'N/A'

group by Shelf, productid

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(\*) 'The Count', avg(listprice) 'AvgPrice'

from Production.Product

group by color, class

having color is not null and class is not null

**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 'Country', s.name '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 'Country', s.name '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 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

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

select o.shippostalcode, sum(od.quantity) as qty

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

where o.shippostalcode is not null

group by o.shippostalcode

order by qty desc

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

select top 5 o.shippostalcode, sum(od.quantity) as qty

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 qty desc

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

select city, count(customerid)

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 NumOfCustomer

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.customerid, c.companyname, c.contactname

from customers c join orders o on o.customerid = c.customerid

where orderdate >'1998-1-1'

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

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

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

GROUP BY c.ContactName

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

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;

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

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;

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 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;

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

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;

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

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;

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

SELECT T1.EmployeeId, T1.LastName, T1.FirstName,T2.ReportsTo, COUNT(T2.ReportsTo) AS Subordinate

FROM Employees T1 JOIN Employees T2 ON T1.EmployeeId = T2.ReportsTo

WHERE T2.ReportsTo IS NOT NULL

GROUP BY T1.EmployeeId, T1.LastName, T1.FirstName,T2.ReportsTo

HAVING COUNT(T2.ReportsTo) > 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.CompanyName, c.ContactName, 'Customer' as Type

FROM Customers c

UNION

SELECT s.City, s.CompanyName, s.ContactName, 'Supplier' as Type

FROM Suppliers s;

GOOD LUCK.