## INNER JOIN

**1**

ALTER VIEW [dbo].[vAllHumanResourcesEmployee]

AS

SELECT

EDH.[BusinessEntityID]

,EDH.[Title]

,EDH.[FirstName]

,EDH.[MiddleName]

,EDH.[LastName]

,EDH.[Suffix]

,EDH.[Shift]

,EDH.[Department]

,EDH.[GroupName]

,EDH.[StartDate]

,EDH.[EndDate]

,ED.[JobTitle]

,E.[PhoneNumber]

,E.[PhoneNumberType]

,E.[EmailAddress]

,E.[EmailPromotion]

,E.[AddressLine1]

,E.[AddressLine2]

,E.[City]

,E.[StateProvinceName]

,E.[PostalCode]

,E.[CountryRegionName]

,E.[AdditionalContactInfo]

FROM [AdventureWorks].[HumanResources].[vEmployeeDepartmentHistory] AS EDH

INNER JOIN [AdventureWorks].[HumanResources].[vEmployeeDepartment] AS ED

ON EDH.BusinessEntityID = ED.BusinessEntityID

INNER JOIN [AdventureWorks].[HumanResources].[vEmployee] AS E

ON ED.BusinessEntityID = E.BusinessEntityID

GO

**2**

ALTER VIEW [Sales].[vStoreWithAddresses] AS

SELECT

s.[BusinessEntityID]

,s.[Name]

,at.[Name] AS [AddressType]

,a.[AddressLine1]

,a.[AddressLine2]

,a.[City]

,sp.[Name] AS [StateProvinceName]

,a.[PostalCode]

,cr.[Name] AS [CountryRegionName]

FROM [Sales].[Store] s

INNER JOIN [Person].[BusinessEntityAddress] bea

ON bea.[BusinessEntityID] = s.[BusinessEntityID]

INNER JOIN [Person].[Address] a

ON a.[AddressID] = bea.[AddressID]

INNER JOIN [Person].[StateProvince] sp

ON sp.[StateProvinceID] = a.[StateProvinceID]

INNER JOIN [Person].[CountryRegion] cr

ON cr.[CountryRegionCode] = sp.[CountryRegionCode]

INNER JOIN [Person].[AddressType] at

ON at.[AddressTypeID] = bea.[AddressTypeID];

GO

## SUB-QUERIES

**SELECT**

ALTER VIEW [dbo].[vSubQuerySELECT]

AS

SELECT SWA.Name, (SELECT AVG(SWD.AnnualSales) FROM [AdventureWorks].[Sales].[vStoreWithDemographics] AS SWD WHERE SWD.Specialty = 'Mountain') AS AverageMountainSales

FROM [AdventureWorks].[Sales].[vStoreWithAddresses] as SWA

GO

**FROM**

ALTER VIEW [dbo].[vSubQueryFROM]

AS

SELECT SalesOrderID

FROM

(SELECT SalesOrderID FROM [AdventureWorks].[Sales].[SalesOrderDetail]

WHERE CarrierTrackingNumber LIKE '%4DFB-%')

t

GO

**WHERE**

ALTER VIEW [dbo].[vSubQueryWHERE]

AS

SELECT Name

FROM [AdventureWorks].[Sales].[vStoreWithContacts]

WHERE BusinessEntityID IN

(SELECT BusinessEntityID FROM [AdventureWorks].[Sales].[vStoreWithAddresses] WHERE AddressType = 'Shipping')

GO

**WHERE2**

ALTER VIEW [dbo].[vSubQueryWHERE2]

AS

SELECT \*

FROM [AdventureWorks].[dbo].[vAllSalesStore]

WHERE [BusinessEntityID] IN

(SELECT [ProductID] FROM [AdventureWorks].[Production].[vProductAndDescription]

WHERE [ProductModel] LIKE '%Mountain%')

GO

**WHERE\_NOT\_IN**

ALTER VIEW [dbo].[vSubQueryWHERE\_NOT\_IN]

AS

SELECT Name

FROM [AdventureWorks].[Sales].[vStoreWithContacts]

WHERE BusinessEntityID NOT IN

(SELECT BusinessEntityID FROM [AdventureWorks].[Sales].[vStoreWithAddresses] WHERE AddressType = 'Shipping')

GO

## DATEDIFF

**BETWEEN**

ALTER VIEW [dbo].[vDATEDIFF\_Between]

AS

SELECT year(ShipDate) as ShipYear, sum(SubTotal) as TotalSumsPerYear

FROM [AdventureWorks].[Sales].[SalesOrderHeader]

WHERE DATEDIFF(year,ShipDate,GETDATE()) between 8 and 9

GROUP BY year(ShipDate)

GO

**GREATER THAN**

ALTER VIEW [dbo].[vDATEDIFF\_Greater\_Than]

AS

SELECT year(ShipDate) as ShipYear, sum(SubTotal) as TotalSumsPerYear

FROM [AdventureWorks].[Sales].[SalesOrderHeader]

WHERE DATEDIFF(year,ShipDate,GETDATE()) > 6

GROUP BY year(ShipDate)

GO

**LESS THAN**

ALTER VIEW [dbo].[vDATEDIFF\_Less\_Than]

AS

SELECT year(ShipDate) as ShipYear, sum(SubTotal) as TotalSumsPerYear

FROM [AdventureWorks].[Sales].[SalesOrderHeader]

WHERE DATEDIFF(year,ShipDate,GETDATE()) < 9

GROUP BY year(ShipDate)

GO

**DAYS BETWEEN**

ALTER VIEW [dbo].[vDaysBetweenOrderAndShip]

AS

SELECT DATEDIFF(dd, [OrderDate], [ShipDate]) AS DaysBetweenOrderAndShip

FROM [AdventureWorks].[Sales].[SalesOrderHeader]

GO

**DAYS: WHERE FILTER**

ALTER VIEW [dbo].[vDaysBetweenOrderAndShip\_Where\_Filter]

AS

SELECT DATEDIFF(dd, [OrderDate], [ShipDate]) AS DaysBetweenOrderAndShip

FROM [AdventureWorks].[Sales].[SalesOrderHeader]

WHERE DATEDIFF(dd, [OrderDate], [ShipDate]) = 7

GO

## SHOW DUPLICATES

ALTER VIEW [dbo].[vDuplicateIDs]

AS

SELECT [PurchaseOrderID], sum([ReceivedQty]) AS TotalReicedQuantity

FROM [AdventureWorks].[Purchasing].[PurchaseOrderDetail]

GROUP BY [PurchaseOrderID]

HAVING count(\*) > 1

GO

## DaysBetweenOrderAndShip

**GENERAL**

ALTER VIEW [dbo].[vDaysBetweenOrderAndShip]

AS

SELECT DATEDIFF(dd, [OrderDate], [ShipDate]) AS DaysBetweenOrderAndShip

FROM [AdventureWorks].[Sales].[SalesOrderHeader]

GO

**WHERE**

ALTER VIEW [dbo].[vDaysBetweenOrderAndShip\_Where\_Filter]

AS

SELECT DATEDIFF(dd, [OrderDate], [ShipDate]) AS DaysBetweenOrderAndShip

FROM [AdventureWorks].[Sales].[SalesOrderHeader]

WHERE DATEDIFF(dd, [OrderDate], [ShipDate]) = 7

## NULL

**IS**

ALTER VIEW [dbo].[vWHERE\_IS\_NULL]

AS

SELECT Name, Suffix, EmailAddress

FROM [AdventureWorks].[dbo].[vAllSalesStore]

WHERE Suffix IS NULL

GO

**IS NOT**

ALTER VIEW [dbo].[vWHERE\_IS\_NOT\_NULL]

AS

SELECT Name, Suffix, EmailAddress

FROM [AdventureWorks].[dbo].[vAllSalesStore]

WHERE Suffix IS NOT NULL

GO

## TOP PERCENT

ALTER VIEW [dbo].[vTOP\_PERCENT]

AS

SELECT TOP 30 PERCENT [BusinessEntityID], [SalesYTD]

FROM [AdventureWorks].[Sales].[vSalesPerson]

ORDER BY [SalesYTD] DESC

GO

## EXCEPT

The SQL EXCEPT statement returns those records from the left SELECT query, that are not present in the results returned by the SELECT query on the right side of the EXCEPT statement.

ALTER VIEW [dbo].[vEXCEPT]

AS

SELECT ProductID

FROM Production.Product

EXCEPT

SELECT ProductID

FROM Production.WorkOrder ;

GO

## INTERSECT

The SQL INTERSECT clause/operator is used to combine two SELECT statements, but returns rows only from the first SELECT statement that are identical to a row in the second SELECT statement. This means INTERSECT returns only common rows returned by the two SELECT statements.

ALTER VIEW [dbo].[vINTERSECT]

AS

SELECT ProductID

FROM Production.Product

INTERSECT

SELECT ProductID

FROM Production.WorkOrder ;

GO

## UNION

ALTER VIEW [dbo].[vStackedTable]

AS

SELECT \*

FROM [Invoice Matching Problem].[dbo].[INVOICENO]

UNION

SELECT \*

FROM [Invoice Matching Problem].[dbo].[RXNUMBER]

GO

## End of Month

CREATE VIEW [dbo].[vEOMONTH]

AS

SELECT Department, EOMONTH(StartDate) AS EndOfMonth

FROM [AdventureWorks].[dbo].[vAllHumanResourcesEmployee]

GO

## CARTESIAN JOIN

ALTER VIEW [dbo].[vCartesianJoin]

AS

SELECT

distinct

t3.[Department]

,t3.[GroupName]

,t3.[PhoneNumberType]

FROM [AdventureWorks].[dbo].[vAllHumanResourcesEmployee] t1,

[AdventureWorks].[dbo].[vAllHumanResourcesEmployee] t2,

[AdventureWorks].[dbo].[vAllHumanResourcesEmployee] t3

GO

## CASE

**Field Categories Between**

ALTER VIEW [dbo].[vCase\_FieldCategories\_BETWEEN]

AS

SELECT

[CountryRegionName],

sum([AnnualSales]) AS TotalAnnualSales,

(CASE

WHEN sum([AnnualSales]) between 0 and 200000000 THEN 'Low'

WHEN sum([AnnualSales]) between 200000001 and 300000000 THEN 'Medium'

WHEN sum([AnnualSales]) > 300000000 THEN 'High'

END) AS 'SalesGroups'

FROM [AdventureWorks].[dbo].[vAllSalesStore]

GROUP BY [CountryRegionName]

GO

**ORDER BY LEN**

ALTER VIEW [dbo].[vCase\_OrderBy\_LEN]

AS

SELECT TOP (100) PERCENT ContactType, Name, PhoneNumber, EmailAddress, AddressType

FROM [AdventureWorks].[dbo].[vAllSalesStore]

ORDER BY LEN(ContactType) DESC, LEN(PhoneNumber) DESC,

(CASE

WHEN PhoneNumber IS NOT NULL THEN 1

ELSE 0

END),

(CASE

WHEN EmailAddress IS NOT NULL THEN 1

ELSE 0

END)

GO

**COUNTS OF VALUES**

ALTER VIEW [dbo].[vCase1]

AS

SELECT

CountryRegionName,

COUNT(CASE

WHEN [AnnualSales] > 2000000 THEN 1

END) AS OverTwoMillion,

COUNT(CASE

WHEN [AnnualSales] <= 2000000 THEN 1

END) AS TwoMillionOrUnder

FROM [AdventureWorks].[dbo].[vAllSalesStore]

GROUP BY CountryRegionName

GO

## DISTINCT

Only displays one of each string

ALTER VIEW [dbo].[vDISTINCT]

AS

SELECT DISTINCT [AddressType], [SquareFeet]

FROM [AdventureWorks].[dbo].[vAllSalesStore]

GO

## PIVOT

**1**

ALTER VIEW [dbo].[vPivot]

AS

SELECT 'AverageSales' AS AverageSalesByCountry,

[Germany], [United States], [Australia], [United Kingdom], [Canada], [France]

FROM

(SELECT CountryRegionName, AnnualSales

FROM [AdventureWorks].[dbo].[vAllSalesStore]) AS SourceTable

PIVOT

(

AVG(AnnualSales)

FOR CountryRegionName IN ([Germany], [United States], [Australia], [United Kingdom], [Canada], [France])

) AS PivotTable;

GO

**2**

ALTER VIEW [dbo].[vPivot2]

AS

SELECT Specialty,

[Germany], [United States], [Australia], [United Kingdom], [Canada], [France]

FROM

(SELECT CountryRegionName, AnnualSales, Specialty

FROM [AdventureWorks].[dbo].[vAllSalesStore]) AS SourceTable

PIVOT

(

AVG(AnnualSales)

FOR CountryRegionName IN ([Germany], [United States], [Australia], [United Kingdom], [Canada], [France])

) AS PivotTable;

GO

## Weighted Average

ALTER VIEW [dbo].[vWeighted\_Average]

AS

SELECT [BusinessEntityID], Name, EmailAddress, sum([NumberEmployees] \* [SquareFeet]) / SUM([SquareFeet]) AS EmployeeAveragePerSquareFoot

FROM [AdventureWorks].[dbo].[vAllSalesStore]

GROUP BY [BusinessEntityID], Name, EmailAddress

GO

## COALESCE

ALTER VIEW [dbo].[vCOALESCE]

AS

SELECT Title, FirstName, COALESCE(MiddleName, 'N/A') AS MiddleNameNA, LastName

FROM [AdventureWorks].[dbo].[vAllSalesStore]

GO

## CONCAT

ALTER VIEW [dbo].[vCONCAT]

AS

SELECT CONCAT(Title, ' ',FirstName, ' ', LastName) AS FullName

FROM [AdventureWorks].[dbo].[vAllSalesStore]

GO

## CONVERT

ALTER VIEW [dbo].[vCONVERT]

AS

SELECT BusinessEntityID, EmailAddress, CONVERT(int, SalesYTD) as SalesYTD\_int, CONVERT(int, SalesLastYear) AS SalesLastYear\_int

FROM [AdventureWorks].[Sales].[vSalesPerson]

GO

## WINDOW FUCTION: ROW – WHERE

ALTER VIEW [dbo].[vWindowFunctionROW\_NUMBER\_CTE\_WHERE\_filter]

AS

WITH RowNumberTable

AS

(

SELECT FirstName, MiddleName, LastName, EmailAddress, TerritoryGroup, SalesYTD,

ROW\_NUMBER() OVER (PARTITION BY TerritoryGroup ORDER BY SalesYTD DESC)

AS RowNumber

FROM [AdventureWorks].[Sales].[vSalesPerson]

)

SELECT FirstName, MiddleName, LastName, EmailAddress, TerritoryGroup, SalesYTD, RowNumber

FROM RowNumberTable

WHERE RowNumber = 1

GO

## WINDOW FUNCTION: ORDER DATE DIFFERENCE

ALTER VIEW [dbo].[vWindowFunctionLEAD\_OrderDay\_Difference]

AS

WITH OrderDateDifferenceTable AS

(

SELECT

[CustomerID]

,[OrderDate]

,LEAD(OrderDate, 1) OVER (PARTITION BY CustomerID ORDER BY OrderDate) AS NextOrderDate

FROM [AdventureWorks].[Sales].[SalesOrderHeader]

)

SELECT

[CustomerID]

,[OrderDate]

,NextOrderDate

,DATEDIFF(dd, OrderDate, NextOrderDate) as OrderDayDifference

FROM OrderDateDifferenceTable

GO