**SQL Server**

A la izq estan los database en folders, selecciona NEW QUERY para poder programar dentro de ellos o mostrar en tablas la info que se tiene.

**SELECT indicar que columnas queremos que regrese**

SELECT ‘ Ana ‘ , ‘ Mendoza ‘

|  |  |
| --- | --- |
| Ana | Mendoza |

‘ ‘ : para escribir texto

Coma: separa en columnas

SELECT 5-3

|  |
| --- |
| 2 |

SELECT [Column 1], [column N]

FROM [Database Name], [Schema Name], [Table Name]

*I want to select this columns from this tables*

No se necesita especificar el database al que esta conectado porque ya esta en la barra blanca seleccionado.

SELECT FirstName

FROM Person.Person

|  |
| --- |
| Abi |
| Sam |
| adam |

SELECT FirstName, LastName

FROM Person.Person

|  |  |
| --- | --- |
| Abi | Abel |
| Sam | Ackerman |
| Adam | adams |

*Hay veces que hay millones de datos y se tardaria mucho si busco asi*

*Entonces escribe asi*

SELECT TOP 500 FirstName, MiddleName, LastName

FROM Person.Person

SELECT TOP 10 PERCENT FirstName, MiddleName, LastName

FROM Person.Person

SELECT TOP \*

FROM Person.Person

\*: muestra todas las columnas y todos los renglones que tiene la table Person.Person

SELECT TOP 100 \*  
FROM Production.Product

Muestra las primeras 100 filas pero con todas las columnas

SELECT TOP 100 FirstName AS [customer First Name], MiddleName, LastName

FROM Person.Person

|  |  |  |
| --- | --- | --- |
| Customer First Name | MiddleName | LastName |
| Abi | Anna | Adams |

AS [ ] o AS ‘ ‘: es para indicar como quieres remplazar el nombre de la columna por lo que escribes dentro de los corchetes

*Si seleccionas el texto asi y le picas execute entonces solo eso se corre*

**WHERE una especificación de la columna usando comparation operator**

WHERE [Column Name] [comparation operator] [some value]

SELECT \*

FROM Production.Product

WHERE ListPrice > 10

**>, < , >= , <= , = , < >**

SELECT \*

FROM HumanResources.vEmployee

WHERE FirstName = ‘ Chris ’

*Es lo mismo en mayúsculas que en minúsculas*

SELECT \*

FROM HumanResources.Employee

WHERE BirthDate >= ‘ 1-1-1980 ’

*Las fechas las identifica con – o con /*

SELECT \*

FROM HumanResources.Employee

WHERE BirthDate >= ‘ 1-1-1980 ’ AND Gender = ‘ F ‘

SELECT \*

FROM HumanResources.Employee

WHERE MaritalStatus = ‘ S ’ AND Gender = ‘ M ‘

SELECT \*

FROM HumanResources.Employee

WHERE MaritalStatus = ‘ S ’ OR Gender = ‘ M ‘

AND va primero que un OR

WHERE First Name IN ( ‘ Chris’ , ‘Thomas’)

*Es como un or, una lista de los que quieres que aparezcan*

WHERE AnnualSales BETWEEN 100000 AND 200000

*Significa que esta dentro*

WHERE FirstName LIKE ‘Mi%’

*Para que traiga nombres que empiecen con Mi: miguel, mike, mi*

*Espera un character cualquiera*

WHERE FirstName LIKE ‘Mi\_’

*Este es para que espere un carácter pero solo 1*

WHERE FirstName LIKE ‘%S’

*Cualquier nombre que termine con s*

WHERE FirstName LIKE ‘%h%’

*Cualquier cantidad de caracteres antes o despues de una h en el nombre*

WHERE FirstName LIKE ‘\_on’

*Un solo carácter antes*

WHERE FirstName LIKE ‘D[ a , o ]n’

*Te da Dan o Don*

WHERE FirstName LIKE ‘D[ a-f , r-z ]n’

*Conjunto de palabras posibles*

WHERE FirstName LIKE ‘D[ ^a - f ]n’

*Ese es el negativo, que el gorrito se ponga antes de una letra o un grupo te dice que no debe estar*

WHERE MiddleName is NULL

*Salen informacion que en blanco*

WHERE MiddleName = ‘NULL’

*No va a salir nada porque no esta registrado -> no data*

WHERE MiddleName IS NOT NULL

*Es nulo*

SELECT

FROM

WHERE

ORDER BY ColumnName, ColumnOrdinal, ColumnAlias [ASC/DESC]

SELECT FirstName, LastName

FROM Sales.vIndividualCustomer

ORDER BY 2

Si no viene especificado asc o desc lo hace asc

SELECT FirstName, LastName AS ]customer last name]

FROM Sales.vIndividualCustomer

ORDER BY [customer last name] DESC

**Como nosotros escribimos**

**SELECT**

**FROM**

**WHERE**

**ORDER BY**

**//////////////**

**Como sequel lee**

**FROM**

**WHERE**

**SELECT**

**ORDER BY**

**Por eso nomas el order by reconoce directamente lo que select le dice ocmo “alias”, sise pone en un where nos e va a reconocer por que esta antes**

SELECT

FROM

ORDER BY LastName, FirstName DESC

Xavier Adams

ORDER BY 2, FirstName

*Te lo dara en orden normal*

SELECT LastName, FirstName, SalesQuota

From Sales.vSalesPerson

WHERE SALESQUOTA >= 250000

ORDER BY SalesQuota DESC, LastName

INNER JOIN

SELECT

FROM Production.Product P

Es una table alias

SELECT P.Name, P.ProductNumber

FROM Production.Product P

Traeme el nombre y el product number de la table P

SELECT P.Name, P.ProductNumber, PS.Name

FROM Production.Product P

INNER JOIN Production.ProductSubcategory PS

ON P.ProductSubcategory ID = PS.ProductSubcategoryID

LEFT OUTER JOIN

RIGHT OUTER JOIN

Te devuelve todo lo de un lado

SELECT MAX (TotalDue)

FROM Sales.SalesOrderHeader

*Sale el valor mas grande*

SELECT MIN (TotalDue)

FROM Sales.SalesOrderHeader

*Sale el valor mas chico*

SELECT COUNT (\*)

FROM Sales.SalesOrderHeader

Te cuenta las filas en la table

SELECT COUNT (SalesPersonID)

FROM Sales.SalesOrderHeader

Los valores no NULL en la table de esa columna

SELECT COUNT (DISTINCT FirstName)

FROM Sales.SalesOrderHeader

Te muestra los nombres unicos, no repetidos en la table

SELECT AVG (TotalDue)

FROM Sales.SalesOrderHeader

El promedio de los valores en la columna

SELECT SUM (TotalDue)

FROM Sales.SalesOrderHeader

Suma los valores

SELECT SUM (TotalDue)

FROM Sales.SalesOrderHeader

WHERE OrderDate BETWEEN ‘1/1/2006’ AND ‘12/31/2006’

Sumas solo los total due que tengan esa fecha

**SELECT**

**FROM**

**WHERE**

**GROUP BY [Column Name]**

**ORDER BY**

SELECT SalesPersonID, SUM (TotalDue) AS Total Sales

FROM Sales.SalesPerson

GROUP BY SalesPersonID

Te suma la cantidad de sales juntando el ID de cada persona, si la persona con el ID 3 venido 5 sales, te va a sumar esas 5 y te va a agrupar en un solo 3.

SELECT P.FirstName + ‘ ‘ + P.LastName AS SalesPersonName

Es para juntar dos columnas en una sola

**SELECT**

**FROM**

**WHERE**

**GROUP BY**

**HAVING [Aggregate Function] {comparison operator} [filtering criteria]**

**ORDER BY**

El having te da un grupo (hecho por group by) filtrado con alguna condicion

SELECT

ST.Name AS [Territory Name].

SUM(TotalDue) AS [Total Sales – 2006]

FROM Sales.SalesOrderHeader SOH

INNER JOIN Sales.SalesTerritory ST

ON ST.TerritoryID = SOH.TerritoryID

WHERE OrderDate BETWEEN ‘1/1/2006’ AND ‘12/31/2006’

GROUP BY ST.Name

HAVING SUM (TotalDue) > 4000000

ORDER BY 1

Pone el total due de los grupos mayores de 4000000