

SQL Analytics I

Módulo III



Temario

- SQL – Operadores
 - De comparación
 - Lógicos
 - Aritméticos
 - De concatenación de cadenas de caracteres
 - De conjunto

SQL

Operadores

De Comparación

- Al utilizar un operador de comparación se obtienen tres posibles resultados: TRUE (verdadero), FALSE (falso) , y UNKNOWN (desconocido – al trabajar con valores nulos)
- Ejemplo:
 - $5 > 3$ - TRUE
 - $6 = 10$ - FALSE
- Estos resultados devuelven un tipo de datos Boolean (TRUE, FALSE) y se conocen como expresiones booleanas

SQL

Operadores

De Comparación

- Expresiones con tipos de datos booleanos se utilizan en la cláusula WHERE para filtrar las filas que cumplen los requisitos para las condiciones de búsqueda
- Ejemplo:

```
SELECT FirstName, LastName
```

```
FROM Person.Person
```

```
WHERE FirstName = 'José'
```

SQL

Operadores

De Comparación

Operador	Significado (ejemplo)
=	Igual a (Nombre = 'José')
>	Mayor que (Salario > 10000)
<	Menor que (Edad < 18)
>=	Mayor o igual que (Precio >= 100)
<=	Menor o igual que (CantProductos <= 10)
<>	Distinto a (Departamento <> 'Montevideo')

SQL

Operadores

De Comparación

- Ejemplos:

```
SELECT TOP 10 ProductID
      ,Name
      ,ListPrice
FROM Production.Product
WHERE ListPrice >= 100
ORDER BY ListPrice
```

	ProductID	Name	ListPrice
1	995	ML Bottom Bracket	101,24
2	806	ML Headset	102,29
3	907	Rear Brakes	106,50
4	948	Front Brakes	106,50
5	826	LL Road Rear Wheel	112,565
6	876	Hitch Rack - 4-Bike	120,00
7	810	HL Mountain Handlebars	120,27
8	813	HL Road Handlebars	120,27
9	894	Rear Derailleur	121,46
10	996	HL Bottom Bracket	121,49

- En este caso se obtiene el ID del Producto, el nombre y precio de lista de los primeros 10 productos cuyo precios sean mayores o iguales a 100 ordenados por precio de menor a mayor (ORDER BY por defecto es ASC)

SQL

Operadores

De Comparación

- Ejemplos:

```
SELECT *  
FROM HumanResources.Employee  
WHERE JobTitle = 'Database Administrator'
```

- En este caso se obtienen todos los datos de los empleados que sean Administradores de Bases de Datos

	BusinessEntityID	NationalIDNumber	LoginID	OrganizationNode	OrganizationLevel	JobTitle	BirthDa
1	270	643805155	adventure-works\françois0	0x8E30	2	Database Administrator	1965-0
2	271	929666391	adventure-works\dan1	0x8E50	2	Database Administrator	1966-0

SQL

Operadores

Lógicos

Operador	Significado
OR	TRUE si cualquiera de las dos expresiones booleanas es TRUE.
AND	TRUE si ambas expresiones booleanas son TRUE.
NOT	Invierte el valor de cualquier otro operador booleano.
BETWEEN	TRUE si el operando está dentro de un intervalo.
IN	TRUE si el operando es igual a uno de la lista de expresiones.
LIKE	TRUE si el operando coincide con un patrón.

SQL

Operadores

Lógicos

Tabla de Verdad:

A	B	AND	OR	NOT A
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

Ejemplo:

A = (5 > 2)- True

B = (6 < 5)- False

(A) AND (B) -> False

(A) OR (B) -> True

SQL

Operadores

Lógicos

AND



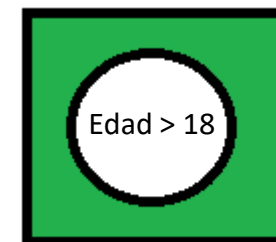
Edad > 18 Altura > 1.60

OR



Edad > 18 Altura > 1.60

NOT



AND – Se tienen que cumplir ambas condiciones (edad > 18 AND altura > 1.60)

OR – Se tiene que cumplir una de las condiciones por lo menos (edad > 18 OR altura > 1.60)

NOT – No se tiene que cumplir la condición establecida (NOT edad > 18)

SQL

Operadores

Lógicos

- Ejemplos:

```
SELECT *  
FROM HumanResources.Employee  
WHERE JobTitle = 'Database Administrator' AND MaritalStatus = 'S'
```

- En este caso se obtienen todos los datos de los empleados que sean Administradores de Bases de Datos Y sean Solteros/as

	BusinessEntityID	NationalID...	LoginID	OrganizationNode	Org...	Job Title	BirthDate	MaritalStatus	Gender
1	270	643805155	advent...	0x8E30	2	Database Administrator	1965-06-17	S	M

SQL

Operadores

Lógicos

BETWEEN:

- Evalúa si el valor de un campo se encuentra entre 2 valores determinados
- Ejemplo:

```
SELECT TOP 3 ProductID, Name, ListPrice  
FROM Production.Product  
WHERE ListPrice BETWEEN 1 AND 10  
ORDER BY ListPrice
```

	ProductID	Name	ListPrice
1	873	Patch Kit/8 Patches	2,29
2	922	Road Tire Tube	3,99
3	870	Water Bottle - 30 oz.	4,99
4	921	Mountain Tire Tube	4,99
5	923	Touring Tire Tube	4,99

SQL

Operadores

Lógicos

IN:

- Evalúa si el valor de un campo coincide con alguno de los valores de una lista
- Ejemplo:

```
SELECT ProductModelID, Name  
FROM Production.ProductModel  
WHERE ProductModelID IN (3, 4)
```

	ProductModelID	Name
1	3	Full-Finger Gloves
2	4	Half-Finger Gloves

SQL

Operadores

Lógicos

NOT IN:

- Evalúa si el valor de un campo no coincide con alguno de los valores de una lista
- Ejemplo:

```
SELECT ProductModelID, Name  
FROM Production.ProductModel  
WHERE ProductModelID NOT IN (3, 4)  
ORDER BY ProductModelID
```

	ProductModelID	Name
1	1	Classic Vest
2	2	Cycling Cap
3	5	HL Mountain Frame
4	6	HL Road Frame
5	7	HL Touring Frame
6	8	LL Mountain Frame

SQL

Operadores

Lógicos

LIKE:

- Evalúa si el valor de un campo coincide con un patrón establecido
- Ejemplo:

```
SELECT JobTitle, Gender, BirthDate  
FROM HumanResources.Employee  
WHERE JobTitle LIKE '%Financial%'
```

	Job Title	Gender	Birth Date
1	Chief Financial Officer	F	1966-02-06
2	Assistant to the Chief Financial Officer	M	1954-07-23

SQL

Operadores

Lógicos

LIKE:

- El carácter '%' oficia de comodín donde acepta cualquier cantidad de caracteres o ninguno en su lugar
- Ejemplo:
 - 'Pa%' - Palabras válidas: Papá, Paraguas, Palmera, etc
 - '%ro' - Palabras válidas: Gorro, pájaro, muro, etc
 - '%ar%' - Palabras válidas: Cartas, hablar, arriba, etc

SQL

Operadores

Lógicos

LIKE:

- El carácter '_' oficia de comodín donde acepta cualquier carácter individual
- Ejemplo:

WHERE name **LIKE** ' _ean ' - busca todos los nombres de cuatro letras que terminen en ean (Dean, Sean, etc.)

WHERE name **LIKE** ' _____ean' - busca todos los nombres de nueve letras que terminen en ean (tiene 6 _)

SQL

Operadores

Lógicos

LIKE:

- Los caracteres [] – establecen intervalos o conjunto de caracteres específicos que se deseen
- Ejemplo:

WHERE last_name **LIKE** '[C-P]arsen' - busca apellidos que terminen en arsen y empiecen por cualquier carácter individual entre C y P, como Carsen, Larsen, Karsen, etc.

WHERE last_name **LIKE** '[CP]arsen' - busca apellidos que terminen en arsen y empiecen por C o P, como Carsen o Parsen.

SQL

Operadores

Lógicos

LIKE:

- Los caracteres [^] – establecen intervalos o conjunto de caracteres específicos que **NO** se deseen
- Ejemplo:

WHERE last_name **LIKE** 'de[^c]%' - busca apellidos que empiecen con 'de' y que la siguiente letra no sea 'c'.

WHERE last_name **LIKE** 'de[^a-d]%' - busca apellidos que empiecen con 'de' y que la siguiente letra no sea 'a', 'b', 'c' o 'd'.

SQL

Operadores

Aritméticos

Operador	Significado
+	SUMA
-	RESTA
*	MULTIPLICACIÓN
/	DIVISIÓN
%	MÓDULO - Devuelve el resto de una división.

SQL

Operadores

Aritméticos

- Ejemplos:

```
SELECT BusinessEntityID,  
       VacationHours,  
       SickLeaveHours,  
       VacationHours + SickLeaveHours AS 'Total Hours Away'  
FROM HumanResources.Employee  
ORDER BY 'Total Hours Away' ASC
```

	BusinessEntityID	VacationHours	SickLeaveHours	Total Hours Away
1	97	0	20	20
2	139	0	20	20
3	234	0	20	20
4	141	1	20	21
5	95	1	20	21

- En este caso se obtiene el ID del Empleado, la cantidad de horas de vacaciones, la cantidad de horas que estuvo enfermo y las suma de estas dos últimas, ordenadas por las suma total (recordar que igualmente ORDER BY por defecto es ASC)

SQL

Operadores

Aritméticos

- Ejemplos:

```
SELECT ProductID, Name,  
ListPrice,  
ListPrice * 1.15 AS NewPrice  
FROM Production.Product  
WHERE Name LIKE 'Mountain-%'  
ORDER BY ProductID ASC
```

	ProductID	Name	ListPrice	NewPrice
1	771	Mountain-100 Silver, 38	3399,99	3909.988500
2	772	Mountain-100 Silver, 42	3399,99	3909.988500
3	773	Mountain-100 Silver, 44	3399,99	3909.988500
4	774	Mountain-100 Silver, 48	3399,99	3909.988500
5	775	Mountain-100 Black, 38	3374,99	3881.238500
6	776	Mountain-100 Black, 42	3374,99	3881.238500

- En este caso se obtiene el ID del Producto, su nombre, su precio de lista y un nuevo precio calculado con un 15% de incremento del precio de lista, para los productos cuyo nombre empieza con 'Mountain-', ordenados por el ID del Producto

- Ejemplos:

```
SELECT TOP(100)ProductID, OrderQty,  
OrderQty % 3 AS Modulo  
FROM Sales.SalesOrderDetail  
WHERE OrderQty > 3
```

	ProductID	OrderQty	Modulo
1	709	6	0
2	711	4	1
3	712	4	1
4	715	4	1
5	776	4	1
6	708	5	2
7	770	5	2

- En este caso se obtiene el ID del Producto, la cantidad pedida de ese producto y el resto de dividir la cantidad entre 3 para los 100 primeros productos donde la cantidad ordenada sea mayor a 3. Este caso o similar, puede servir cuando por ejemplo los productos se empaquen de a 3 y quiero saber en cada pedido cuantos productos irán 'sueños'.

- Ejemplo:

```
SELECT (LastName + ', ' + FirstName) AS Name
FROM Person.Person
ORDER BY LastName ASC, FirstName ASC
```

	Name
1	Abbas, Syed
2	Abel, Catherine
3	Abercrombie, Kim
4	Abercrombie, Kim
5	Abercrombie, Kim
6	Abolrous, Hazem
7	Abolrous, Sam
8	Acevedo, Humberto
9	Achong, Gustavo
10	Ackeman, Pilar

- En este caso se obtiene el apellido de la personas concatenado con una coma y el primer nombre como un único campo al que llamaremos 'Name', ordenándolo primero por el apellido y en caso de coincidencia por el primer nombre.

UNION:

- Combina los resultados de dos o más consultas en un solo conjunto de resultados
- Ejemplo:

```
SELECT ProductModelID, Name
FROM Production.ProductModel
WHERE ProductModelID IN (3, 4)
UNION
SELECT ProductModelID, Name
FROM Production.ProductModel
WHERE ProductModelID = 5
```

	ProductModelID	Name
1	3	Full-Finger Gloves
2	4	Half-Finger Gloves
3	5	HL Mountain Frame

Resumen Módulo III

- SQL – Operadores
 - De comparación
 - Lógicos
 - Aritméticos
 - De concatenación de cadenas
 - De conjunto