



Trabajo con datos

**Operadores\_DML**

## Operadores\_DML

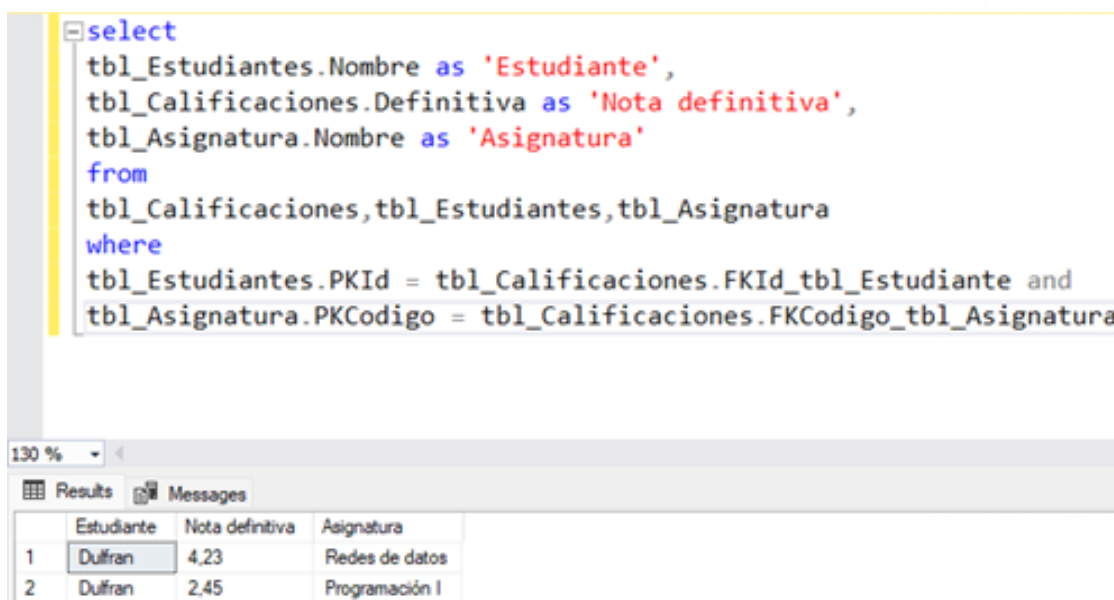
Los operadores DML permiten acompañar las labores de manipulación de los datos en una base de datos; dentro de estos, se encuentran varias instrucciones que hacen uso de dichos operadores para cambiar, comparar o realizar operaciones de tipo matemático para entregar al usuario los resultados que este requiere en los procesos a desarrollar. Aunque los operadores pueden combinarse, para realizar operaciones que llegan a ser de un nivel complejo, se debe entender que, en una operación, lo importante es la aplicación lógica de las operaciones y su integración con los valores de entrada y salida para mostrar al usuario resultados correctos; a continuación, veremos los diferentes operadores utilizados en DML.

### 1 Operadores de comparación

A continuación, trataremos con cada uno de estos operadores y su funcionalidad dentro del lenguaje DML y cómo es su aplicación dentro de dicho contexto.

#### Operador igual (=)

Este operador se utiliza cuando se desea comparar dos valores contenidos dentro de una instrucción DML y determinar las acciones que se deben realizar; esto es también aplicable al momento de realizar consultas y este se aplica para comparar si existe coincidencia entre las llaves foráneas y primarias.



```

select
tbl_Estudiantes.Nombre as 'Estudiante',
tbl_Calificaciones.Definitiva as 'Nota definitiva',
tbl_Asignatura.Nombre as 'Asignatura'
from
tbl_Calificaciones,tbl_Estudiantes,tbl_Asignatura
where
tbl_Estudiantes.PKId = tbl_Calificaciones.FKId_tbl_Estudiante and
tbl_Asignatura.PKCodigo = tbl_Calificaciones.FKCodigo_tbl_Asignatura
  
```

	Estudiante	Nota definitiva	Asignatura
1	Dulfran	4,23	Redes de datos
2	Dulfran	2,45	Programación I

Como se puede observar en la imagen anterior, se realiza el proceso de igualdad en la comparación de las llaves, tanto primarias como foráneas, para realizar la comparación de estos datos y arrojar el resultado correcto.

#### Operador Mayor que(>)

Este operador se utiliza cuando deseamos conocer si un valor supera el límite de comparación con otro ya establecido; esto es sumamente importante para los sistemas de información, puesto que, regularmente, se deben establecer límites en los valores de entrada o procesados en el sistema, y se deben tomar decisiones con base en estos resultados; a continuación, veremos un ejemplo de esto.

```

select
tbl_Estudiantes.Nombre as 'Estudiante',
tbl_Calificaciones.Definitiva as 'Nota definitiva',
tbl_Asignatura.Nombre as 'Asignatura'
from
tbl_Calificaciones,tbl_Estudiantes,tbl_Asignatura
where
tbl_Estudiantes.PKId = tbl_Calificaciones.FKId_tbl_Estudiante and
tbl_Asignatura.PKCodigo = tbl_Calificaciones.FKCodigo_tbl_Asignatura and
tbl_Calificaciones.Definitiva > 3.0

```

130 %

Results Messages

	Estudiante	Nota definitiva	Asignatura
1	Dulfran	4.23	Redes de datos

Como se puede observar en la imagen anterior, el operador mayor que establece un filtro, que solo puede mostrar a los estudiantes que tienen notas definitivas mayores a 3.0, lo que hace que descarte las demás; esto es muy práctico y se utiliza bastante en los sistemas de bases de datos para establecer condiciones mediante las cuales se limite la muestra de datos al usuario.

### Operador Menor que(<)

Este operador se utiliza de manera contraria al ya expuesto anteriormente; crea un filtro, pero teniendo en cuenta que, en la condición de este filtro, solo afecte los valores que se encuentran por debajo, como se ilustra a continuación en la imagen.

```

select
tbl_Estudiantes.Nombre as 'Estudiante',
tbl_Calificaciones.Definitiva as 'Nota definitiva',
tbl_Asignatura.Nombre as 'Asignatura'
from
tbl_Calificaciones,tbl_Estudiantes,tbl_Asignatura
where
tbl_Estudiantes.PKId = tbl_Calificaciones.FKId_tbl_Estudiante and
tbl_Asignatura.PKCodigo = tbl_Calificaciones.FKCodigo_tbl_Asignatura and
tbl_Calificaciones.Definitiva < 3.0

```

130 %

Results Messages

	Estudiante	Nota definitiva	Asignatura
1	Dulfran	2.45	Programación I

### Operador Mayor igual (>=)

Este operador cumple casi el mismo propósito del operador Mayor que, solo incluye el valor que se está comparando dentro del rango del filtro; por ejemplo, si tengo un valor de 3 y digo mayor o igual que 3, esto incluye todos los valores que se encuentren dentro del rango, incluyendo el 3 hacia adelante, como se ilustra en la imagen.



```

select
tbl_Estudiantes.Nombre as 'Estudiante',
tbl_Calificaciones.Definitiva as 'Nota definitiva',
tbl_Asignatura.Nombre as 'Asignatura'
from
tbl_Calificaciones,tbl_Estudiantes,tbl_Asignatura
where
tbl_Estudiantes.PKId = tbl_Calificaciones.FKId_tbl_Estudiante and
tbl_Asignatura.PKCodigo = tbl_Calificaciones.FKCodigo_tbl_Asignatura and
tbl_Calificaciones.Definitiva >= 3.0

```

	Estudiante	Nota definitiva	Asignatura
1	Dulfran	4,23	Redes de datos
2	Dulfran	4,6	Arquitectura de software

### Operador Menor que(<=)

Este operador se utiliza, en el caso contrario del operador mayor que, en lugar de realizar una comparación aumentando el valor del filtro, lo hace en forma de disminución; por eso trabaja con todos los datos que se encuentren, desde el dato ingresado, hasta los que se encuentran antes del mismo; a continuación, observamos un ejemplo del mismo.

```

select
tbl_Estudiantes.Nombre as 'Estudiante',
tbl_Calificaciones.Definitiva as 'Nota definitiva',
tbl_Asignatura.Nombre as 'Asignatura'
from
tbl_Calificaciones,tbl_Estudiantes,tbl_Asignatura
where
tbl_Estudiantes.PKId = tbl_Calificaciones.FKId_tbl_Estudiante and
tbl_Asignatura.PKCodigo = tbl_Calificaciones.FKCodigo_tbl_Asignatura and
tbl_Calificaciones.Definitiva <= 3.0

```

	Estudiante	Nota definitiva	Asignatura
1	Dulfran	2,45	Programación I
2	Dulfran	1,8	Bases de datos I

## 2 Operadores aritméticos

Los operadores aritméticos son instrucciones que permiten realizar operaciones que arrojan resultados de operaciones matemáticas; gracias a los operadores aritméticos podemos realizar las operaciones requeridas por los usuarios, dentro de los cuales encontramos la suma, resta, multiplicación y división. Ahora bien, debemos destacar que cada uno de ellos está diseñado para arrojar un resultado concreto al usuario; algunos de ellos deben tener una restricción como, por ejemplo, el operador de división el cual debe ser ejecutado siempre y cuando en su denominador no tenga un valor de cero; a continuación, veremos cada uno de ellos y su aplicación.

Todos conocemos las operaciones básicas, es decir: la suma, la multiplicación, la resta y la división. Todas estas operaciones pueden llevarse a cabo sin problemas en la mayoría de las bases de datos, sin necesidad de recurrir a datos que estén guardados en la base de datos.

La forma en la que vamos a hacer dichas operaciones es a través de la sentencia SELECT (otras de las bondades de esta sentencia), de la siguiente forma:

### Operador suma

La adición o suma es la operación matemática de composición que consiste en combinar o añadir dos números o más, para obtener una cantidad final o total. La suma también ilustra el proceso de juntar dos colecciones de objetos con el fin de obtener una sola colección.

PKN_Registro	FKId_tbl_Estud...	Nota1	Nota2
1	123	4,2	3,9

```

use dba_Calificaciones
go
select (tbl_Calificaciones.Nota1 + tbl_Calificaciones.Nota2)
from tbl_Calificaciones
where
tbl_Calificaciones.PKN_Registro = 1

```

1	8,1
---	-----

Como se observa en la imagen, a mano izquierda tenemos los registros de la base de datos donde se almacenan las calificaciones del estudiante y, a mano derecha, se observa la operación matemática que realiza la suma de ese proceso; como se puede observar, todo este proceso es llevado a cabo con el lenguaje DML, que permite, mediante la manipulación de la información, lograr mostrar el resultado esperado por el usuario.

### Operador resta

La resta o la sustracción es una operación aritmética que se representa con el signo; representa la operación de eliminación de objetos de una colección; a continuación, se ilustra la operación utilizada.

PKN_Registro	FKId_tbl_Estud...	Nota1	Nota2
1	123	4,2	3,9

```

use dba_Calificaciones
go
select (tbl_Calificaciones.Nota1 - tbl_Calificaciones.Nota2)
from tbl_Calificaciones
where
tbl_Calificaciones.PKN_Registro = 1

```

1	0,3
---	-----

Como se observa en la imagen, se realiza la operación resta y se procede a mostrar el resultado; de acuerdo con la imagen anterior, es importante tener en cuenta que se han utilizado los mismos valores para mantener la integridad del resultado.

### Operador multiplicación

La multiplicación es una operación binaria y derivada de la suma que se establece en un conjunto numérico. En aritmética, es una de las cuatro operaciones elementales, junto con la suma, la resta y la división, y es la operación inversa de esta última; a continuación, veremos un ejemplo de cómo es su aplicación.

PKN_Registro	FKId_tbl_Estud...	Nota1	Nota2
1	123	4,2	3,9

```

use dbs_Calificaciones
go
select (tbl_Calificaciones.Nota1 * tbl_Calificaciones.Nota2)
from tbl_Calificaciones
where
tbl_Calificaciones.PKN_Registro = 1

```

130 %

Results Messages

(No column name)

1 16,38

Como se observa en la imagen anterior, se aplica la operación de multiplicación, la cual tiene una dependencia con la suma, como se puede observar en el código; su representación es con el símbolo asterisco (\*), el cual le da a entender al motor de la base de datos que esta es la operación a realizar y procede a mostrarnos el resultado.

### Operador división

La división es una de las operaciones básicas de la aritmética, que consiste en separar en partes iguales un total. En matemáticas, el símbolo de la división es el signo /; esta operación es muy importante en el ámbito matemático y de la lógica porque permite realizar operaciones que pueden generar de valores enteros a valores con puntos decimales; un ejemplo de esto es el cálculo de los porcentajes de descuento para aplicar a un producto; es necesaria su aplicación, para lograr el resultado correcto. A continuación, veremos el resultado de su aplicación con el lenguaje DML.

PKN_Registro	FKId_tbl_Estud...	Nota1	Nota2
1	123	4,2	3,9

```

use dbs_Calificaciones
go
select (tbl_Calificaciones.Nota1 / tbl_Calificaciones.Nota2)
from tbl_Calificaciones
where
tbl_Calificaciones.PKN_Registro = 1

```

130 %

Results Messages

(No column name)

1 1,07692307692308

Como se puede observar, en la imagen anterior encontramos la aplicación del operador de división; algo importante a tener en cuenta es que si el número de división es cero, contempla la condición de la teoría matemática, que la división por cero no es posible; ahora bien, es algo que se debería realizar por programación para garantizar la integridad de la operación a realizar, como se observa en la siguiente imagen; por ello es necesario que se garanticen los mismos procesos matemáticos en la base de datos.

```

use dbs_Calificaciones
go
select (tbl_Calificaciones.Nota1 / tbl_Calificaciones.Nota2)
from tbl_Calificaciones
where
tbl_Calificaciones.PKN_Registro = 1

```

130 %

Results Messages

Msg 8134, Level 16, State 1, Line 3  
Divide by zero error encountered.

### 3 Operadores lógicos

Los operadores lógicos se usan para combinar dos valores Booleanos y devolver un resultado verdadero, falso o nulo.

Los operadores lógicos también se denominan operadores Booleanos.

Devuelve True cuando el valor 1 y valor 2 son verdaderos.

Devuelve True cuando valor 1 o valor 2 es verdadero, es decir, de acuerdo con la combinación establecida, así será el resultado esperado, el cual puede ser verdadero o falso, dependiendo de la combinación; a continuación, veremos un ejemplo de tabla de la verdad, la cual permite conocer el resultado de una operación lógica, de acuerdo a sus premisas.

VALOR 1	VALOR 2	AND	OR
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	F

Como se puede observar en la tabla, tenemos los principales operadores lógicos, los cuales permiten realizar las operaciones de comparación en las instrucciones, en el lenguaje DML; una de las más utilizadas es al momento de traer datos e información, debido a que estos operadores permiten comparar y anexar las diferentes validaciones que se deben realizar, para tener información; a continuación, veremos un ejemplo sobre esto.

```

select
tbl_Estudiantes.Nombre as 'Estudiante',
tbl_Calificaciones.Definitiva as 'Nota definitiva',
tbl_Asignatura.Nombre as 'Asignatura'
from
tbl_Calificaciones,tbl_Estudiantes,tbl_Asignatura
where
tbl_Estudiantes.PKId = tbl_Calificaciones.FKId_tbl_Estudiante and
tbl_Asignatura.PKCodigo = tbl_Calificaciones.FKCodigo_tbl_Asignatura and
tbl_Calificaciones.Definitiva <= 3.0

```

130 %

Results Messages

	Estudiante	Nota definitiva	Asignatura
1	Dulfran	2,45	Programación I
2	Dulfran	1,8	Bases de datos I



En la imagen, se encuentra el operador AND el cual tiene la particularidad de ser verdadero, según la tabla vista anteriormente; si ambos resultados son correctos, eso quiere decir que en esta consulta trae los datos; si la llave primaria del estudiante coincide con la llave foránea de la tabla `tbl_Calificaciones` y que el código de la asignatura coincida con el código que se encuentra en la llave foránea de la tabla `tbl_Calificaciones`, esto es coincidir los valores de las llaves para mostrar la información; en el caso que alguna de ellas no coincida, no traerá información alguna. Pero miremos desde el punto de vista del operador OR, este tiene la particularidad que si encuentra alguno de los valores verdadero, traerá la información solicitada; es decir, si alguna de estas llaves no es igual a la otra, él traerá los datos de igual manera. Y por último, tendríamos el operador NOT, el cual tiene la característica de negar los resultados finales arrojados por la operación; es decir, si el resultado final es verdadero, lo convierte a falso y si es falso lo convierte a verdadero.

#### 4 Operadores de conjuntos

Los operadores de conjuntos nos permiten realizar los procesos de relacionamiento de las diferentes entidades y las operaciones que estas pueden llegar a realizar, de acuerdo con esas relaciones establecidas; por eso, es importante entender un concepto llamado álgebra de conjuntos, el cual abordaremos a continuación para comprender mucho más el contexto de este proceso.

Álgebra relacional es un conjunto de operaciones simples sobre tablas relacionales, a partir de las cuales se definen operaciones más complejas mediante composición. Definen, por tanto, un pequeño lenguaje de manipulación de datos.

El elemento fundamental del modelo relacional de bases de datos es la tabla relacional. Una tabla relacional es una representación extensional de una relación definida sobre un cierto dominio.

Es un método que consiste, básicamente, en crear o construir nuevas relaciones, a partir de relaciones existentes.

Existen 2 tipos de operadores algebraicos:

##### Operadores básicos o primitivos

Se clasifican en:

- **Proyección ( $\pi$ ):** este operador permite extraer columnas de una relación y de esta manera crea un subconjunto de atributos de la relación, además elimina las filas duplicadas.
- **Selección ( $\sigma$ ):** este operador permite seleccionar un subconjunto de filas o registros de una relación y de acuerdo a la condición planteada, los registros serán seleccionados para formar parte de un nuevo subconjunto.
- **Unión ( $\cup$ ):** la unión de 2 relaciones R y S es otra relación la cual va a tener los registros de R en S o en ambas, además se eliminan los registros duplicados. En esta relación R y S deben ser compatibles, es decir que deben estar definidas sobre el mismo conjunto de atributos.
- **Diferencia ( $-$ ):** la diferencia de 2 relaciones R y S es otra relación la cual va a tener los registros que están en R, pero no están en S. En esta relación R y S deben ser compatibles.
- **Producto cartesiano ( $\times$ ):** es una relación que consiste en la concatenación de cada una de las filas de la relación R, con cada una de las filas de la relación S.





### Operadores no básicos o derivados

Se clasifican en:

- Intersección ( $\cap$ ): es una relación que contiene el conjunto de todas las filas que están tanto, en la relación R, como en S. R y S deben ser compatibles.
- Unión natural ( $\bowtie$ ): el resultado es una relación con los atributos de ambas relaciones y se obtiene combinando las filas de ambas relaciones que tengan el mismo valor en los atributos comunes. El join se usa entre los atributos comunes de las entidades o tablas que poseen la clave primaria de una tabla foránea correspondiente de otra entidad.
- División ( $/$ ): define una relación sobre el conjunto de atributos C, incluido en la relación R, y que contiene el conjunto de valores de S que en las filas de R están combinadas con cada una de las filas de S.

Este contexto matemático, nos permite entender cómo se encuentra relacionada la información en las entidades o tablas de la base de datos que tenemos y es un insumo valioso al momento de sustraer información de las mismas; por eso es importante tener siempre en cuenta, los límites y procedimientos que se pueden llegar a realizar de acuerdo a la teoría de conjuntos.