

Guía - Taller SQL 03 - Tipos de Operadores en MYSQL

Operadores relacionales

Los operadores relacionales vinculan un campo con un valor para que MySQL compare cada registro (el campo especificado) con el valor dado. Los operadores relacionales son los siguientes:

Operador	Descripción
=	igual
<>	distinto
>	mayor
<	menor
>=	mayor o igual
<=	menor o igual

El operador = compara dos expresiones, y da como resultado 1 si son iguales, o 0 si son diferentes.

MySQL dispone de dos operadores equivalente para comprobar desigualdades, <> y !=. Si las expresiones comparadas son diferentes, el resultado es verdadero, y si son iguales, el resultado es falso

Estos operadores también permiten comparar cadenas, fechas, y por supuesto, números: Cuando se comparan cadenas, se considera menor la cadena que aparezca antes por orden alfabético.

Si son fechas, se considera que es menor cuanto más antigua sea. Hay que tener en cuenta que si comparamos cadenas que contienen números, no hay conversión, y se comparan las cadenas tal como aparecen.

Operadores lógicos

Podemos establecer más de una condición con la cláusula "WHERE", para ello aprenderemos los operadores lógicos.

Son los siguientes:

Operador	Descripción
AND	significa "y"
OR	significa "y/o"
XOR	significa "o",

NOT	significa "no", invierte el resultado
()	paréntesis

Los operadores lógicos se usan para combinar condiciones. Los operadores lógicos se usan para crear expresiones lógicas complejas. Permiten el uso de álgebra booleana, y nos ayudarán a crear condiciones mucho más precisas.

En el álgebra booleana sólo existen dos valores posibles para los operandos y los resultados: verdadero y falso. **MySQL** dispone de dos constantes para esos valores: *TRUE* y *FALSE*, respectivamente.

MySQL añade un tercer valor: desconocido. Esto es para que sea posible trabajar con valores *NULL*. El valor verdadero se implementa como 1 o *TRUE*, el falso como 0 o *FALSE* y el desconocido como *NULL*.

Operador AND

En **MySQL** se puede usar tanto la forma *AND* como *&&*, es decir, ambas formas se refieren al mismo operador: “Y” lógico.

Se trata de un operador binario, es decir, requiere de dos operandos. El resultado es verdadero sólo si ambos operandos son verdaderos, y falso si cualquier operando es falso. Esto se representa mediante la siguiente tabla de verdad:

A	B	A AND B
falso	falso	falso
falso	verdadero	falso
verdadero	falso	falso
verdadero	verdadero	verdadero
falso	NULL	falso
NULL	falso	falso
verdadero	NULL	NULL
NULL	verdadero	NULL

Al igual que todos los operadores binarios que veremos, el operador “Y” se puede asociar, es decir, se pueden crear expresiones como *A AND B AND C*. El hecho de que se requieran dos operandos significa que las operaciones se realizan tomando los operandos dos a dos, y estas expresiones se evalúan de izquierda a derecha. Primero se evalúa *A AND B*, y el resultado, *R*, se usa como primer operando de la siguiente operación *R AND C*

Operador OR

En **MySQL** este operador también tiene dos formas equivalentes *OR* y *//*

El operador O también es binario. Si ambos operandos son distintos de *NULL* y el resultado es verdadero si cualquiera de ellos es verdadero, y falso si ambos son falsos. Si uno de los operandos es *NULL* el resultado es verdadero si el otro es verdadero, y *NULL* en el caso contrario. La tabla de verdad es:

A	B	A OR B
falso	falso	falso
falso	verdadero	verdadero
verdadero	falso	verdadero
verdadero	verdadero	verdadero
falso	NULL	NULL
NULL	falso	NULL
verdadero	NULL	verdadero
NULL	verdadero	verdadero

Operador NOT

El operador *NOT*, que también se puede escribir como *!*, es un operador unitario, es decir sólo afecta a un operando. Si el operando es verdadero devuelve falso, y viceversa. Si el operando es *NULL* el valor devuelto también es *NULL*.

A	NOT A
falso	verdadero
verdadero	falso
NULL	NULL

Entre los operadores de MySQL, hay uno para comprobar si una expresión está comprendida en un determinado rango de valores. La sintaxis es:

<expresión> BETWEEN mínimo AND máximo

<expresión> NOT BETWEEN mínimo AND máximo

El orden de prioridad de los operadores lógicos es el siguiente: "not" se aplica antes que "and" y "and" antes que "or", si no se especifica un orden de evaluación mediante el uso de paréntesis.

El orden en el que se evalúan los operadores con igual nivel de precedencia es indefinido, por ello se recomienda usar los paréntesis.

El valor NULL

A veces, puede desconocerse o no existir el dato correspondiente a algún campo de un registro. En estos casos decimos que el campo puede contener valores nulos.

Note que el valor "null" no es una cadena de caracteres, no se coloca entre comillas.

Si un campo acepta valores nulos, podemos ingresar: **null** cuando no conocemos el valor. Los campos establecidos como clave primaria no aceptan valores nulos. Si intentamos ingresar el valor "null" para este campo, no lo tomará.

Para recuperar los registros que contengan el valor "null" por ejemplo en un campo "precio" no podemos utilizar los operadores relacionales vistos anteriormente: = (igual) y <> (distinto); debemos utilizar los operadores "is null" (es igual a null) y "is not null" (no es null):

Los operadores IS NULL e IS NOT NULL sirven para verificar si una expresión determinada es o no nula. La sintaxis es:

<expresión> IS NULL

<expresión> IS NOT NULL

Operadores Aritméticos

Los operadores aritméticos se aplican a valores numéricos, ya sean enteros o en coma flotante. El resultado siempre es un valor numérico, entero o en coma flotante.

Puede realizar operaciones aritméticas en campos numéricos mediante operadores aritméticos, como se muestra a continuación:

Operador	Descripción
+	Suma
-	Resta
*	Multipliación
/	División
%	Modulo

Es posible obtener salidas en las cuales una columna sea el resultado de un cálculo y no un campo de una tabla.

A continuación se muestran algunos ejemplos de operadores en mySQL con base en la siguiente tabla:

libros
titulo: varchar(20)
autor: varchar(30)
editorial: varchar(15)
precio: float

Realizar las siguientes actividades:

1. Crear una base de datos que se llame: **libreria**
2. Ejecutar la siguiente sentencia SQL para crear la tabla:
create table libros(titulo varchar(20), autor varchar(30), editorial varchar(15), precio float);
3. Agregamos registros a la tabla. Note que al ingresar valores numéricos (en este caso float) no se utilizan comillas y para el separador de decimales se usa el punto(.)
insert into libros (titulo,autor,editorial,precio) values ('El aleph','Borges','Planeta',12000.50);
insert into libros (titulo,autor,editorial,precio) values ('Martin Fierro','Jose Hernandez','Emece',16500.00);
insert into libros (titulo,autor,editorial,precio) values ('Aprenda PHP','Mario Molina','Emece',35700.40);
insert into libros (titulo,autor,editorial,precio) values ('Cervantes','Borges','Paidos',50000.90);
4. Ejecutar y analizar los resultados de las siguientes consultas, las cuales están separadas por tipos de operadores con su respectiva explicación. Se debe evidenciar en el documento, los pantallazos de los resultados de las consultas

Ejemplos de Operadores relacionales:

- Seleccionamos los registros cuyo autor sea diferente de 'Borges':
select titulo,autor,editorial,precio from libros where autor<>'Borges';
- Seleccionamos los registros cuyo precio supere o iguale los 20.000 pesos:
select titulo,autor,editorial,precio from libros where precio>=20000;
- Seleccionamos los libros cuyo precio es menor o igual a 30.000:
select titulo,autor,editorial,precio from libros where precio<=30000;

Ejemplos de Operadores Lógicos:

- Queremos recuperar todos los registros cuyo autor sea igual a "Borges" y cuyo precio no supere los 20000 pesos, para ello necesitamos 2 condiciones:
select * from libros
where (autor='Borges') and
(precio<=20000);

Los registros recuperados en una sentencia que une 2 condiciones con el operador "and", cumplen con las 2 condiciones.

- Queremos ver los libros cuyo autor sea "Borges" y/o cuya editorial sea "Planeta":

```
select * from libros  
  where autor='Borges' or  
        editorial='Planeta';
```

En la sentencia anterior usamos el operador "or", indicamos que recupere los libros en los cuales el valor del campo "autor" sea "Borges" y/o el valor del campo "editorial" sea "Planeta", es decir, seleccionará los registros que cumplan con la primera condición, con la segunda condición o con ambas condiciones.

Los registros recuperados con una sentencia que une 2 condiciones con el operador "or", cumplen 1 de las condiciones o ambas.

- Queremos ver los libros cuyo autor sea "Borges" o cuya editorial sea "Planeta":

```
select * from libros  
  where (autor='Borges') xor  
        (editorial='Planeta');
```

En la sentencia anterior usamos el operador "xor", indicamos que recupere los libros en los cuales el valor del campo "autor" sea "Borges" o el valor del campo "editorial" sea "Planeta", es decir, seleccionará los registros que cumplan con la primera condición o con la segunda condición pero no los que cumplan con ambas condiciones. Los registros recuperados con una sentencia que une 2 condiciones con el operador "xor", cumplen 1 de las condiciones, no ambas.

- Queremos recuperar los libros que no cumplan la condición dada, por ejemplo, aquellos cuya editorial NO sea "Planeta":

```
select * from libros  
  where not (editorial='Planeta');
```

El operador "not" invierte el resultado de la condición a la cual antecede. Los registros recuperados en una sentencia en la cual aparece el operador "not", no cumplen con la condición a la cual afecta el "NO".

Los paréntesis se usan para encerrar condiciones, para que se evalúen como una sola expresión.

- Los siguientes ejemplos muestran cómo afecta el orden de los operadores (prioridad en que se ejecutan) y la utilización de los paréntesis. Cuando explicitamos varias condiciones con diferentes operadores lógicos (combinamos "and", "or") permite establecer el orden de prioridad de la evaluación; además permite diferenciar las expresiones más claramente.

Por ejemplo, las siguientes expresiones devuelven un resultado diferente:

```
select * from libros
where (autor='Borges') or
(editorial='Paidos' and precio<20000);
```

```
select*from libros
where (autor='Borges' or editorial='Paidos') and
(precio<20000);
```

Si bien los paréntesis no son obligatorios en todos los casos, se recomienda utilizarlos para evitar confusiones.

El orden de prioridad de los operadores lógicos es el siguiente: "not" se aplica antes que "and" y "and" antes que "or", si no se especifica un orden de evaluación mediante el uso de paréntesis.

El orden en el que se evalúan los operadores con igual nivel de precedencia es indefinido, por ello se recomienda usar los paréntesis.

Ejemplos de Operadores Aritméticos:

Es posible obtener salidas en las cuales una columna sea el resultado de un cálculo y no un campo de una tabla. A la tabla anterior, le adicionaremos el campo cantidad y nos quedará de la siguiente forma:

libros
titulo: varchar(20)
autor: varchar(30)
editorial: varchar(15)
precio: float
cantidad: int

Realizar las siguientes actividades:

1. Crear una base de datos que se llame: **libreria2**
2. Ejecutar la siguiente sentencia SQL para crear la tabla:
create table libros(codigo int unsigned auto_increment, titulo varchar(40) not null, autor varchar(30), editorial varchar(15), precio float, cantidad smallint unsigned, primary key (codigo));

3. Agregamos registros a la tabla.

```
insert into libros (titulo,autor,editorial,precio,cantidad) values('El aleph','Borges','Planeta',15000,100);
insert into libros (titulo,autor,editorial,precio,cantidad) values('Martin Fierro','Jose Hernandez','Emece',22000.20,200);
insert into libros (titulo,autor,editorial,precio,cantidad) values('Antologia poetica','Borges','Planeta',40000,150);
insert into libros (titulo,autor,editorial,precio,cantidad)values('Aprenda PHP','Mario Molina','Emece',18000.20,200);
insert into libros (titulo,autor,editorial,precio,cantidad)values('Cervantes y el quijote','Borges','Paidos',36000.40,100);
insert into libros (titulo,autor,editorial,precio,cantidad) values('Manual de PHP', 'J.C. Paez', 'Paidos',30000.80,100);
insert into libros (titulo,autor,editorial,precio,cantidad)values('Harry Potter y la piedra filosofal','J.K.
Rowling','Paidos',45000.00,500);
insert into libros (titulo,autor,editorial,precio,cantidad) values('Harry Potter y la camara secreta','J.K.
Rowling','Paidos',46.00,300);
insert into libros (titulo,autor,editorial,precio,cantidad)values('Alicia en el pais de las maravillas','Lewis
Carroll','Paidos',null,50);
```

4. Ejecutar y analizar los resultados de las siguientes consultas

- Si queremos ver los títulos, precio y cantidad de cada libro escribimos la siguiente sentencia:

```
select titulo,precio,cantidad from libros;
```

- Si queremos saber el monto total en dinero de un título podemos multiplicar el precio por la cantidad por cada título, pero también podemos hacer que MySQL realice el cálculo y lo incluya en una columna extra en la salida:

```
select titulo, precio,cantidad,precio*cantidad from libros;
```

- Si queremos saber el precio de cada libro con un 10% de descuento podemos incluir en la sentencia los siguientes cálculos:

```
select titulo, precio,precio*0.1,precio-(precio*0.1) from libros;
```

Operador IN

Para recuperar los libros cuyo autor sea 'Paenza' o 'Borges' usamos 2 condiciones:

```
select * from libros where autor='Borges' or autor='Paenza';
```

Podemos usar "in":

```
select * from libros where autor in('Borges','Paenza');
```

Con "in" averiguamos si el valor de un campo dado (autor) está incluido en la lista de valores especificada (en este caso, 2 cadenas).

Para recuperar los libros cuyo autor no sea 'Paenza' ni 'Borges' usamos:

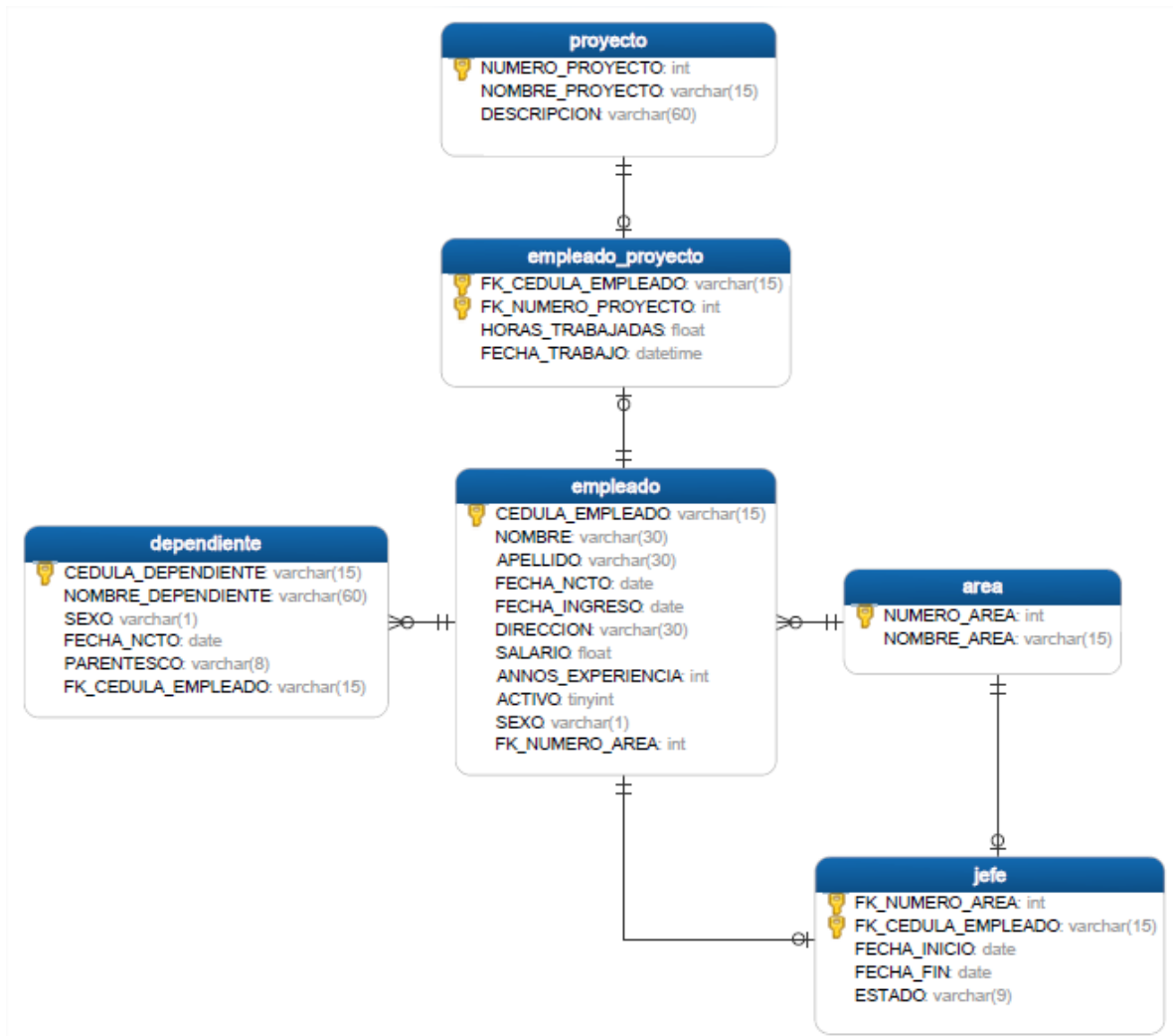
```
select * from libros where autor<>'Borges' and autor<>'Paenza';
```

También podemos usar "in" :


```
select * from libros where autor not in ('Borges','Paenza');
```

Con "in" averiguamos si el valor del campo está incluido en la lista, con "not" antecediendo la condición, invertimos el resultado.

A partir de este momento, con base a los ejemplos vistos anteriormente y al diagrama de tablas siguientes y los archivos proporcionados por el instructor, el aprendiz deberá realizar las siguientes actividades:



1. Crear una base de datos llamada: **empresa**
2. Ejecutar el script de SQL llamado Script BD-Taller SQL_03.sql proporcionado por el instructor
3. Corregir los posibles errores que se produzcan al ejecutar el script, se deben analizar los errores que salgan y corregirlos con la sintaxis correcta para ejecutarlo de nuevo. Se debe mostrar evidencia en el documento de los errores corregidos

4. Una vez el script para crear la base de datos se ejecute de forma correcta, el aprendiz deberá insertar en la base de datos, los registros que se encuentran en el archivo: Script datos BD empresa.sql, allí encontrará las sentencias SQL (INSERT INTO) para agregar los registros. Se debe tener en cuenta el orden de ingreso de los registros para evitar problemas con llaves foráneas
5. En el archivo anterior, no se encuentran sentencias SQL para poder ingresar datos en la tabla: EMPLEADO_PROYECTO, para ello el aprendiz deberá abrir el archivo: Datos BD empresa.xlsx y sacar de allí los datos de dicha tabla para poderlos ingresar a la BD. Pero para ello, se deberán realizar investigaciones en internet para saber cómo se pueden importar dichos datos desde un archivo de Excel. El aprendiz deberá realizar este proceso para poder ingresar los registros a la tabla EMPLEADO_PROYECTO
En caso de requerir registros adicionales para alguna tabla, estos se podrán ingresar a la Base de Datos sin problema.
6. Realizar pantallazos mostrando los registros que se ingresaron a la base de datos por cada tabla

Una vez se tenga la Base de datos lista y con los registros ingresados, el aprendiz procederá a realizar los siguientes ejercicios:

Ejercicios Operadores Relacionales

1. Seleccionar empleados cuya experiencia sea menor o igual a 2 años
2. Seleccionar dependientes menores a 5 años
3. Seleccionar empleados con salario mayor o igual a 1'000.000
4. Seleccionar los dependientes que no tengan parentesco "sobrino"

Ejercicios Operadores Lógicos

1. Seleccionar los empleados del área de ventas y del área de investigación
2. Seleccionar los empleados que estén activos o tengan un salario mayor a \$800.000 pero no las dos condiciones al tiempo
3. Seleccionar los empleados que tengan 10 años de experiencia o que el salario sea menos a \$1'000.000
4. Seleccionar los dependientes que no son mujeres

Ejercicios Operadores Aritméticos

1. Mostrar el 10% del salario del empleado Duvan Velez
2. Mostrar el resultado de sumarle dos años de experiencia al empleado Diego Orozco
3. Mostrar un cuarto (1/4) del salario de la empleada Luisa Marin
4. Mostrar el resultado de restarle \$500.000 al salario de la empleada Daniela Perez

Desarrollar las siguientes consultas

1. Obtener los empleados donde el nombre o el apellido tengan por primera letra una "D".
2. Obtener el nombre de los empleados que tienen 1, 2 o 3 años de experiencia.
3. Obtener todos los datos de aquellos empleados que no registran ninguna dirección
4. Obtener los números de cédula de los empleados que tienen salario mayor a 2.000.000 o que son mayores de 30 años;
5. Seleccionar los proyectos donde hayan trabajado empleados del área de ventas
6. Obtener el nombre de los empleados que están en el mismo proyecto que 'DANIEL MEJIA' y que sean mayores que él.
7. Obtener los empleados que tienen dependientes menores a 10 años y que ganan un salario mayor a 1.500.000
8. Obtener los dependientes que nacieron en el año 1988 y que no son dependientes del empleado 'DANIEL MEJIA' o que son de sexo masculino.
9. Obtener los empleados que tiene un salario mayor y 1.000.000 y menor 3.000.000 o que el salario sea igual a 5'000.000
10. Obtener los empleados que han trabajado 20 horas o más en un proyecto en la última semana
11. Obtener para cada empleado, el valor de las horas totales trabajadas del proyecto "Econo", teniendo en cuenta que el valor de la hora es de \$15.000
12. Obtener proyectos donde trabajan empleados diferentes a 'DANIEL MEJIA' y 'LUISA MARIN'
13. Obtener los dependientes que tengan sexo femenino o que el parentesco sea hijo, pero no los dos al tiempo, de los empleados cuya experiencia sea mayor a 5 años