**Ejemplos De Operadores Relacionales:**

* Seleccionamos los registros cuyo autor sea diferente de 'Borges':

SELECT titulo,autor,editorial,precio

FROM libros

WHERE autor <> 'Borges';

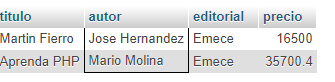


* Seleccionamos los registros cuyo precio supere o iguale los 20.000 pesos:

SELECT titulo,autor,editorial,precio

FROM libros

WHERE precio >= 20000;

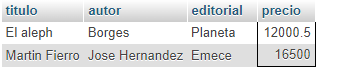


* Seleccionamos los libros cuyo precio es menor o igual a 30.000:

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');



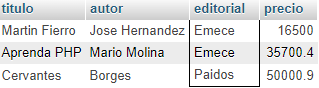
En la sentencia anterior usamos el operador "xor", indicamos que recupere los libros en los cuales el valor del campo "autor" sea "Borges" o el valor del campo "editorial" sea "Planeta", es decir, seleccionará los registros que cumplan con la primera condición o con la segunda condición pero no los que 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.

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:**

* 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)

);

* 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);

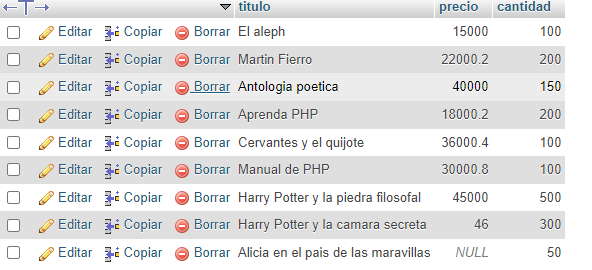
INSERT INTO libros (titulo,autor,editorial,precio,cantidad) VALUES('Harry Potter y la camara secreta','J.K. Rowling','Paidos',46.00,300);

INSERT INTO libros (titulo,autor,editorial,precio,cantidad)VALUES('Alicia en el pais de las maravillas','Lewis Carroll','Paidos',NULL,50);

* 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

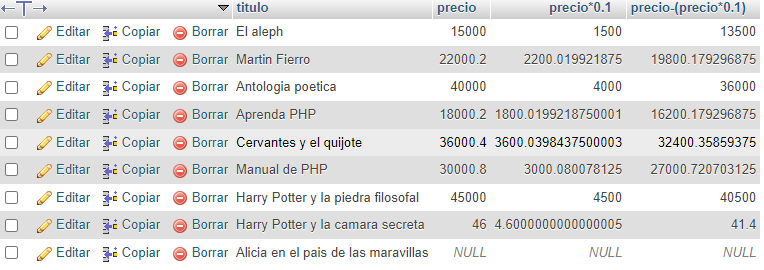
FROM libros;



* 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;



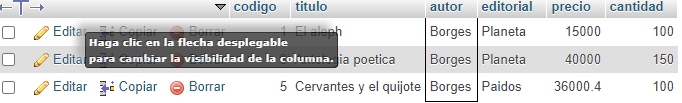
**Operador IN**

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

SELECT \*

FROM libros

WHERE autor = 'Borges' OR autor = 'Paenza';

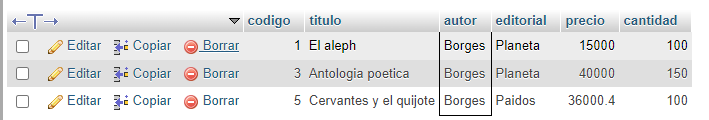


* Podemos usar "in":

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:

Actividad

INNER JOIN