

RESUMEN BÁSICO OPERADORES MYSQL

OPERADORES DE COMPARACION	
=	Equal operator
!=, <>	Not equal operator
>=	Greater than or equal operator
>	Greater than operator
<=	Less than or equal operator
<	Less than operator
BETWEEN ... AND ...	Check whether a value is within a range of values
NOT BETWEEN ... AND ...	Check whether a value is not within a range of values
IN (v1,v2,v3, ...)	Check whether a value is within a set of values
NOT IN (v1,v2,v3, ...)	Check whether a value is not within a set of values
LIKE	Simple pattern matching
NOT LIKE	Negation of simple pattern matching
IS NULL	NULL value test
IS NOT NULL	NOT NULL value test
OPERADORES ARITMETICOS	
+	Addition operator
-	Minus operator
*	Multiplication operator
/	Division operator
DIV	Integer division
MOD, %	Modulo operator
OPERADORES LOGICOS	
AND, &&	Logical AND
NOT, !	Negates value
OR,	Logical OR

Funciones matemáticas.

OBJETIVO	EJEMPLO	RESULTADO
Potencia	POW (2,3)	8
Raíz cuadrada	SQRT (25)	5
Redondear al entero inferior	FLOOR (1.7)	1
Redondear al entero superior	CEIL (1.2)	2
Redondear al entero más próximo	ROUND (3.56)	4
Redondear con decimales	ROUND (3.56,1)	3.6

Funciones de cadenas.

OBJETIVO	EJEMPLO	RESULTADO
Obtener la longitud de una cadena	CHARACTER_LENGTH ('hola')	4
Concatenar cadenas	CONCAT ('h','ol','a')	hola
Obtener el principio de una cadena	LEFT ('hola',2)	ho
Obtener el final de una cadena	RIGHT ('hola',2)	la
Obtener una subcadena	SUBSTRING ('hola',2,2)	ol
	SUBSTRING ('hola',2)	ola
Obtener la posición de una subcadena	INSTR ('cucaracha','a')	4
	LOCATE ('a','cucaracha')	4
	LOCATE ('a','cucaracha',5)	6
Reemplazar una subcadena por otra	REPLACE ('hola','ol','x')	hxa
Convertir a minúsculas	LOWER ('Hola')	hola
Convertir a mayúsculas	UPPER ('Hola')	HOLA
Rellenar una cadena por la izquierda	LPAD ('Hola',8,'-')	----Hola
Rellenar una cadena por la derecha	RPAD ('Hola',8,'-')	Hola----
Eliminar los espacios de los extremos	TRIM (' Hola ')	Hola

Funciones de fechas.

OBJETIVO	EJEMPLO	RESULTADO
Obtener la fecha actual	CURDATE()	Fecha actual
Obtener la hora actual	CURTIME()	Hora actual
Obtener la fecha y hora actual	NOW()	Fecha y hora actual
Obtener el año de una fecha	YEAR('1984-10-21')	1984
Obtener el mes de una fecha	MONTH('1984-10-21')	10
Obtener el día de una fecha	DAY('1984-10-21')	21
Obtener el día de la semana (0=Lunes)	WEEKDAY('1984-10-21')	6
Obtener el nombre del día de la semana	DAYNAME('1984-10-21')	Sunday
Sumar un intervalo de tiempo a una fecha	DATE_ADD('1984-10-21', INTERVAL 3 DAY)	1984-10-24
	DATE_ADD('1984-10-21', INTERVAL 3 MONTH)	1985-01-21
	DATE_ADD('1984-10-21', INTERVAL 3 YEAR)	1987-10-21
	'1984-10-21' + INTERVAL 3 DAY	1984-10-24
Restar un intervalo de tiempo a una fecha	DATE_SUB('1984-10-21', INTERVAL 3 DAY)	1984-10-18
	DATE_ADD('1984-10-21', INTERVAL 3 MONTH)	1985-07-21
	DATE_ADD('1984-10-21', INTERVAL 3 YEAR)	1981-10-21
	'1984-10-21' - INTERVAL 3 DAY	1984-10-18
Obtener la diferencia en días entre dos fechas	DATEDIFF('1984-10-21', '1984-10-15')	6

PROCEDIMIENTO ALMACENADO

```
drop procedure if exists nombre_procedimiento;
delimiter $$
create procedure nombre_procedimiento ( lista_parámetros )
begin
    declaraciones variables locales
    instrucciones
end$$
delimiter ;
```

FUNCION

```
drop function if exists nombre_funcion;
delimiter $$
create function nombre_funcion ( lista_parametros ) returns tipo_retorno
begin
    declaraciones variables locales
    instrucciones
end$$
delimiter ;
```

INSTRUCCION IF

```
if condicion then
    instrucciones
elseif condicion then
    instrucciones
else
    instrucciones
end if;
```

INSTRUCCION CASE 1

```
case expresion
    when valor1 then
        instrucciones
    when valor2 then
        instrucciones
    else
        instrucciones
end case;
```

INSTRUCCION CASE 2

```
case
    when condicion1 then
        instrucciones
    when condicion2 then
        instrucciones
    else
        instrucciones
end case;
```

OPERADOR CASE

```
set var = case expresion
    when valor1 then valor2
    when valor3 then valor4
    else
        valor_defecto
end case;
```

INSTRUCCION WHILE

```
while condicion do
    instrucciones
end while;
```

INSTRUCCION REPEAT

```
repeat
    instrucciones
until condicion
end repeat;
```

INSTRUCCION LOOP

```
bucle : LOOP
    instrucciones
    if condicion then
        leave bucle;
    end if;
    instrucciones
end loop bucle;
```

GENERACION DE ERRORES

```
signal sqlstate '45000' set message_text ='mensaje de error'
```

CURSOR MYSQL

```
declare _fin boolean default false;

declare _cursor cursor for [consulta];

declare continue handler for not found set _fin = true;

open _cursor;

bucle: loop

    fetch _cursor into [variables];

    if _fin then leave bucle; end if;

    [instrucciones]

end loop bucle;

close _cursor;
```

TRIGGER MYSQL

```
drop trigger if exists nombre_trigger;

delimiter $$

create trigger nombre_trigger [before|after] [insert|update|delete]
on nombre_tabla for each row
begin
    [declaraciones variables locales]
    [instrucciones]
end$$

delimiter ;
```