https://www.brandominus.com/manual-sentencias-basicas-en-mysql/

# [Manual] Sentencias básicas en MySQL

Publicado por Carlos Ramos Marquez el 28 , julio, 2015, en DISEÑO & PROGRAMACIÓN, 0 Comentarios

Pues hoy que he estado enredando un rato con MySQl, y me he acordado de aquellos comienzos en los que tenia que volver a mis antiguos apuntes para lograr entender las **sentencias básicas de MySQL** que debía utilizar para hacer llamadas a alguna base de datos.
Y como no, voy a representártelas en este post para hacerte el gran favor de no tener que revolver en tu *trastero de apuntes de programación*, y lo tengas a mano siempre a la hora de hacer alguna **consulta en SQL hacia una base de datos.**
Dejo el listado de las sentencias y más abajo un ejemplo y explicación de cada una de ellas.

## Listado de sentencias básicas en SQL

- **SELECT** se utiliza para consultar datos.
- **DISTINCT** sirve para eliminar los duplicados de las consultas de datos.
- **WHERE** se utiliza incluir las condiciones de los datos que queremos consultar.
- **AND y OR** se utilizan para incluir 2 o más condiciones a una consulta.
- **ORDER BY** se utiliza para ordenar los resultados de una consulta.
- **INSERT** se utiliza para insertar datos.
- **UPDATE** se utiliza actualizar o modificar datos ya existentes.
- **DELETE** se utiliza borrar datos.

## Ejemplos con sentencias básicas en SQL

### Ejemplo con SELECT

Sintaxis SQL SELECT

SELECT * FROM nombretabla

SELECT columna1, columna2 FROM nombretabla

Para los ejemplos, tendremos la siguiente tabla de personas denominada “personas”

Estos son los datos almacenados en la tabla “personas”

| nombre | apellido1 | apellido2 |
|---|---|---|
| ANTONIO | PEREZ | GOMEZ |
| ANTONIO | GARCIA | BENITO |

| | | |
|---|---|---|
| LUIS | LOPEZ | PEREZ |

Si queremos consultar todos los datos de la tabla “personas”

SELECT * FROM personas

Este será el resultado:

| nombre | apellido1 | apellido2 |
|---|---|---|
| ANTONIO | PEREZ | GOMEZ |
| ANTONIO | GARCIA | BENITO |
| LUIS | LOPEZ | PEREZ |

Si queremos consulta todos los nombres y primer apellido de todas las personas

SELECT nombre, apellido1 FROM personas

Este será el resultado:

| nombre | apellido1 |
|---|---|
| ANTONIO | PEREZ |
| ANTONIO | GARCIA |
| LUIS | LOPEZ |

---

## Ejemplo con Distinct

Al realizar una consulta puede ocurrir que existan valores repetidos para algunas columnas. Por ejemplo

SELECT nombre FROM personas

| nombre |
|---|
| ANTONIO |
| LUIS |
| ANTONIO |

Esto no es un problema, pero a veces queremos que no se repitan, por ejemplo, si queremos saber los nombre diferentes que hay en la tabla personas”, entonces utilizaremos **DISTINCT**.

SELECT DISTINCT nombre FROM personas

| nombre |
| --- |
| ANTONIO |
| LUIS |

---

## Ejemplo con WHERE

La cláusula WHERE se utiliza para hacer filtros en las consultas, es decir, seleccionar solamente algunas filas de la tabla que cumplan una determinada condición.

El valor de la condición debe ir entre comillas simples ”.

Por ejemplo:

Seleccionar las personas cuyo nombre sea ANTONIO

SELECT * FROM personas
WHERE nombre = ‘ANTONIO’

| nombre | apellido1 | apellido2 |
| --- | --- | --- |
| ANTONIO | PEREZ | GOMEZ |
| ANTONIO | GARCIA | BENITO |

---

## Ejemplo con AND y OR

Los **operadores AND y OR** se utilizan para filtrar resultados con 2 condiciones.
El operador **AND** mostrará los resultados cuando se cumplan las 2 condiciones.
Condición1 AND condición2

El operador **OR** mostrará los resultados cuando se cumpla alguna de las 2 condiciones.
Condicion1 OR condicion2

En la tabla personas

| nombre | apellido1 | apellido2 |
|---|---|---|
| ANTONIO | PEREZ | GOMEZ |
| ANTONIO | GARCIA | BENITO |
| LUIS | LOPEZ | PEREZ |

La siguiente sentencia (ejemplo AND) dará el siguiente resultado:

```
SELECT * FROM personas
WHERE nombre = 'ANTONIO'
AND apellido1 = 'GARCIA'
```

| nombre | apellido1 | apellido2 |
|---|---|---|
| ANTONIO | GARCIA | BENITO |

La siguiente sentencia (ejemplo OR) dará el siguiente resultado:

```
SELECT * FROM personas
WHERE nombre = 'ANTONIO'
OR apellido1 = 'GARCIA'
```

| nombre | apellido1 | apellido2 |
|---|---|---|
| ANTONIO | PEREZ | GOMEZ |
| ANTONIO | GARCIA | BENITO |

También se pueden combinar AND y OR, como el siguiente ejemplo:

```
SELECT * FROM personas
WHERE nombre = 'ANTONIO'
AND (apellido1 = 'GARCIA' OR apellido1 = 'LOPEZ)
```

| nombre | apellido1 | apellido2 |
|---|---|---|
| ANTONIO | GARCIA | BENITO |

---

## Ejemplo con ORDER BY

**ORDER BY** se utiliza para ordenar los resultados de una consulta, según el valor de la columna especificada.
Por defecto, se ordena de forma ascendente (ASC) según los valores de la columna.

Si se quiere ordenar por orden descendente se utiliza la palabra DES

```
SELECT nombre_columna(s)
FROM nombre_tabla
ORDER BY nombre_columna(s) ASC|DESC
```

Por ejemplo, en la tabla personas :

| nombre | apellido1 | apellido2 |
|---|---|---|
| ANTONIO | PEREZ | GOMEZ |
| LUIS | LOPEZ | PEREZ |
| ANTONIO | GARCIA | BENITO |

```
SELECT nombre, apellido1
FROM personas
ORDER BY apellido1 ASC
```

Esta es la consulta resultante:

| nombre | apellido1 |
|---|---|
| LUIS | LOPEZ |
| ANTONIO | GARCIA |
| ANTONIO | PEREZ |

Ejemplo de ordenación descendiente (DES)

```
SELECT nombre, apellido1
FROM personas
ORDER BY apellido1 DESC
```

Esta es la consulta resultante:

| nombre | apellido1 |
|---|---|
| ANTONIO | PEREZ |
| ANTONIO | GARCIA |
| LUIS | LOPEZ |

## Ejemplo con INSERT

La sentencia INSERT INTO se utiliza para insertar nuevas filas en una tabla.

Es posible insertar una nueva fila en una tabla de dos formas distintas:

```
INSERT INTO nombre_tabla
VALUES (valor1, valor2, valor3, .)

INSERT INTO nombre_tabla (columna1, columna2, columna3,.)
VALUES (valor1, valor2, valor3, .)
```

Ejemplo:

Dada la siguiente tabla personas:

| nombre | apellido1 | apellido2 |
|---|---|---|
| ANTONIO | PEREZ | GOMEZ |
| LUIS | LOPEZ | PEREZ |
| ANTONIO | GARCIA | BENITO |

Si queremos insertar una nueva fila en la tabla personas, lo podemos hacer con cualquiera de las dos sentencias siguientes:

```
INSERT INTO personas
VALUES ('PEDRO', 'RUIZ', 'GONZALEZ')
INSERT INTO personas (nombre, apellido1, apellido2)
VALUES ('PEDRO', 'RUIZ', 'GONZALEZ')
```

Cualquiera de estas sentencias anteriores produce que se inserte una nueva fila en la tabla personas, quedando así dicha tabla:

| nombre | apellido1 | apellido2 |
|---|---|---|
| ANTONIO | PEREZ | GOMEZ |
| LUIS | LOPEZ | PEREZ |
| ANTONIO | GARCIA | BENITO |
| PEDRO | RUIZ | GONZALEZ |

## Ejemplo con UPDATE

La sentencia **UPDATE** se utiliza para modificar valores en una tabla.
La sintaxis de SQL UPDATE es:

```
UPDATE nombre_tabla
SET columna1 = valor1, columna2 = valor2
WHERE columna3 = valor3
```

La cláusula SET establece los nuevos valores para las columnas indicadas.

La cláusula WHERE sirve para seleccionar las filas que queremos modificar.

Ojo: Si omitimos la cláusula WHERE, por defecto, modificará los valores en todas las filas de la tabla.

Ejemplo del uso de SQL UPDATE

| nombre | apellido1 | apellido2 |
|---|---|---|
| ANTONIO | PEREZ | GOMEZ |
| LUIS | LOPEZ | PEREZ |
| ANTONIO | GARCIA | BENITO |
| PEDRO | RUIZ | GONZALEZ |

Si queremos cambiar el apellido2 ‘BENITO’ por ‘RODRIGUEZ’ ejecutaremos:

```
UPDATE personas
SET apellido2 = ‘RODRIGUEZ’
WHERE nombre = ‘ANTONIO’
AND apellido1 = ‘GARCIA’
AND apellido2 = ‘BENITO’
```

Ahora la tabla ‘personas’ quedará así:

| nombre | apellido1 | apellido2 |
|---|---|---|
| ANTONIO | PEREZ | GOMEZ |
| LUIS | LOPEZ | PEREZ |
| ANTONIO | GARCIA | **RODRIGUEZ** |
| PEDRO | RUIZ | GONZALEZ |

## Ejemplo con DELETE

La sentencia **DELETE** sirve para borrar filas de una tabla.
La sintaxis de SQL DELETE es:

```
DELETE FROM nombre_tabla
WHERE nombre_columna = valor
```

Si queremos borrar todos los registros o filas de una tabla, se utiliza la sentencia:

```
DELETE * FROM nombre_tabla;
```

Ejemplo de SQL DELETE para borrar una fila de la tabla personas

| nombre | apellido1 | apellido2 |
|---|---|---|
| ANTONIO | PEREZ | GOMEZ |
| LUIS | LOPEZ | PEREZ |
| ANTONIO | GARCIA | **RODRIGUEZ** |
| PEDRO | RUIZ | GONZALEZ |

Si queremos borrar a la persona LUIS LOPEZ PEREZ, podemos ejecutar el comando:

```
DELETE FROM personas
WHERE nombre = 'LUIS'
AND apellido1 = 'LOPEZ'
AND apellido2 = 'PEREZ'
```

La tabla 'personas' resultante será:

| nombre | apellido1 | apellido2 |
|---|---|---|
| ANTONIO | PEREZ | GOMEZ |
| ANTONIO | GARCIA | **RODRIGUEZ** |
| PEDRO | RUIZ | GONZALEZ |