



## **CLASE**

# **CONCEPTOS**

---

### **Contenido:**

- **FUNCIONES PHP**

*Docente : Gonzalo Anchante Hurtado*

# EL LENGUAJE SQL Y PHP

En este capítulo nos dedicaremos a explicar el lenguaje SQL ya que posteriormente lo usaremos mucho en las conexiones de PHP con MySQL.

## Creación y modificación de Tablas en SQL

**MySQL** esta organizado a partir de tablas y dichas tablas contienen campos. Cada campo es capaz de contener un tipo de dato. Los tipos de datos que es posible crear en el lenguaje SQL son:

<b>Tipo</b>	<b>Descripción</b>
<b>Tinyint[Unsigned]</b>	Entero de 0 a 255 o de -128 a 128
<b>Smallint[Unsigned]</b>	Entero de 0 a 65535 o de -32768 a 32768
<b>Int o Integer</b>	Entero normal. Rango de -2147483648 a 214783648
<b>Float[ (M,D) ]</b>	Número de coma flotante de simple precisión si no se pasa ningún argumento M es el nº de dígitos y D el nº de decimales
<b>Double [ (M,D) ]</b>	Número de coma flotante de doble precisión. Siempre dispone de signo M y D
<b>Decimal [ (M [,D]) ]</b>	Número almacenado como cadena de caracteres M es el número total de dígitos y D el nº de decimales
<b>Date</b>	Tipo fecha. Admite formatos "AAAA-MM-DD" o "AA-MM-DD" o "AAMMDD"
<b>Time</b>	Tipo hora. Admite formato "HH:MM:SS" o "HHMMSS" o "HHMM" o "HH"
<b>Char(longitud)</b>	Cadena de caracteres de la longitud indicada. Se reserva el espacio en caracteres aunque no se usen
<b>Varchar(longitud)</b>	Cadena de caracteres de la longitud indicada que se almacena con su ocupación. Máxima longitud: 255 caracteres
<b>Blob</b>	Tipo destinado a almacenar bits sin interpretar. Se usa para almacenar texto más largo de 255 caracteres. Diferencia mayúsculas de minúsculas.
<b>Text</b>	Tipo destinado a almacenar bits sin interpretar. Se usa para almacenar texto más largo de 255 caracteres. No diferencia mayúsculas de minúsculas.

Para crear una tabla usaremos la siguiente sintaxis:

```
CREATE TABLE Nombre_tabla
(Campo1 Tipo_dato Not Null,
 Campo2 Tipo_dato,
 PRIMARY KEY (Campo3));
```

Esto nos crearía una tabla con 3 campos de los cuales Campo3 es un valor único, es decir, que no puede ser sobrescrito.

Para eliminar una tabla usaremos:

```
DROP TABLE Nombre_tabla;
```

Para modificar la estructura de la tabla usaremos la siguiente sintaxis:

```
ALTER TABLE Nombre_tabla
[ADD Nombre_atributo Definición] //Añadiría un nuevo campo
[CHANGE AntiguoNombreAtributo NuevoNombreAtributo Definición] //Cambiaría
un campo
[DROP NombreAtributo]; //Borraría un campo
```

Los índices son una estructura de acceso que permiten organizar los datos contenidos en una tabla. Para crear un índice usaríamos la siguiente sintaxis:

```
CREATE [UNIQUE] INDEX NombreIndice
ON Tabla (Campos);
```

## Manipulación de datos

### -Inserción de datos

Para insertar datos en la tabla se realiza mediante el comando insert y su sintaxis es la siguiente:

```
INSERT INTO NombreTabla [Campo1,Campo2...CampoN] VALUES
(Valor1,Valor2...ValorN);
```

### - Consultas de datos

Para esta acción usamos el comando **SELECT** y la sintaxis es la siguiente:

```
SELECT ([*]/[Atributos]) FROM Tabla/s [WHERE ListaCondiciones] [GROUP BY
Campo] [HAVING ListaCondiciones] [ORDER BY Campo]
```

Existen un conjunto de funciones dentro de las consultas de datos que nos permiten obtener información o realizar operaciones con respecto a las filas. Las funciones son:

función	Descripción
<b>COUNT(*)/DISTINCT Campo)</b>	Cuenta el numero de filas

<b>SUM(Campo)</b>	Suma los valores del atributo indicado
<b>AVG(Campo)</b>	Obtiene la media aritmética del atributo
<b>MAX(Campo)</b>	Obtiene el valor máximo del atributo
<b>MIN(Campo)</b>	Obtiene el valor mínimo del atributo

#### - Eliminación de datos

Para eliminar datos usamos la sentencia **DELETE** cuya sintaxis es la siguiente:

**DELETE FROM** NombreTabla [WHERE Condición];

MySQL es uno de los gestores de bases de datos mas utilizados en entornos en los cuales se emplea PHP ya que PHP dispone de numerosas funciones que se compaginan perfectamente con MySQL. La forma genérica de obtener información de tablas en Mysql es la siguiente:

- Conexión con el gestor.
- Preparación de la consulta SQL.
- Ejecución de la consulta.
- Procesamiento del resultado obtenido en el cursor.
- Liberación de recursos (esta es opcional, aunque es recomendable).
- Cierre de la conexión con el gestor.

.....

#### Función

**mysql\_connect("host","usuario","password")** Establece la conexión con el servidor. Recibe el host y el usuario y contraseña con el que debe conectar.

**mysql\_select\_db("base de datos",conexión)** Selecciona la base de datos sobre la cual se va a trabajar

**mysql\_query(consulta,conexión)** Ejecuta la consulta SQL indicada como primer parámetro.

**mysql\_num\_fields(cursor)** Devuelve el numero de atributos que figuran en el cursor que se le pasa como parámetro y en el que se almacena el resultado de la consulta

**mysql\_fetch\_row(cursor)** Avanza a la siguiente posición de la fila en cursor. Devuelve un array que contiene en sus celdas cada uno de los valores de los atributos de la fila.

**mysql\_free\_result(cursor)** Libera los recursos asociados al cursor.

**mysql\_close(conexion)** Cierra la conexion establecida con mysql\_connect.

#### Función

**mysql\_list\_dbs(conexion)** Devuelve en un cursor los nombres de las bases de datos disponibles en el servidor al que se haya conectado con mysql\_connect

**mysql\_list\_tables(base\_datos,conexion)** Devuelve en un cursor los nombres de las tablas disponibles en la base de datos.

**mysql\_tablename(cursor,numero\_fila)** Devuelve el nombre de la tabla o base de datos en la que esta el cursor indicado

**mysql\_field\_name(cursor,numero\_col)** Devuelve el nombre del campo cuyo índice se pasa como segundo parámetro

**mysql\_field\_type(cursor,numero\_Col)** Devuelve el tipo del campo cuyo índice se pasa como segundo parámetro

**mysql\_field\_len(cursor,numero\_col)** Devuelve la longitud del campo cuyo índice se pasa como segundo parámetro

**mysql\_field\_flags(cursor,numero\_col)** Devuelve una serie de indicativos correspondientes a características del atributo cuyo índice se pasa como segundo parámetro

**mysql\_affected\_rows(conexion)** Devuelve el numero de filas afectadas por una actualización o borrado

**mysql\_change\_user(usuario,password)** Cambia de usuario

**mysql\_create\_db(basedatos)** Crea una base de datos con el nombre pasado por parámetro

**mysql\_drop\_db(basedatos)** Elimina la base de datos pasada por parámetro

**mysql\_insert\_id(cursor)** Devuelve el valor generado para un AUTOINCREMENT