

Curso:

Programador Web Inicial-Front End Developer



Módulo 4:

Node.js y base de datos

Unidad 2:

Introducción al lenguaje SQL



Presentación

En esta unidad vemos el language SQL, utilizado en bases de datos tanto para realizar consultas como para escribir o eliminar datos.



Objetivos

Que los participantes logren...

- Hacer consultas a la base de datos.
- Escribir y modificar datos en las tablas.
- Eliminar registros de nuestra base de datos.



Bloques temáticos

1. ¿Qué es?
2. Consultas
3. Agregar, Modificar o Eliminar registros

1. ¿Qué es?

El lenguaje de consulta estructurado o SQL (por sus siglas en inglés Structured Query Language) es un **lenguaje declarativo de acceso a bases de datos relacionales** que permite especificar diversos tipos de operaciones en ellas.

El SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales y permite así gran variedad de operaciones.

Características generales de SQL

Es un lenguaje declarativo de "alto nivel" o "de no procedimiento" que, gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros —y no a registros individuales— permite una alta productividad en codificación y la orientación a objetos. De esta forma, una sola sentencia puede equivaler a uno o más programas que se utilizarían en un lenguaje de bajo nivel orientado a registros. SQL también tiene las siguientes características:

- **Lenguaje de definición de datos:** El LDD de SQL proporciona comandos para la definición de esquemas de relación, borrado de relaciones y modificaciones de los esquemas de relación.
- **Lenguaje interactivo de manipulación de datos:** El LMD de SQL incluye lenguajes de consultas basado tanto en álgebra relacional como en cálculo relacional de tuplas.
- **SQL incorporado y dinámico:** Esto quiere decir que se pueden incorporar instrucciones de SQL en lenguajes de programación como: C++, C, Java, PHP, COBOL, Pascal y Fortran.
- **Autorización:** El LDD incluye comandos para especificar los derechos de acceso a las relaciones y a las vistas.

Tipos de datos

- **Varchar:** Recibe cadena de palabras compuestas de letras, números y caracteres especiales.
- **Int** es el principal tipo de datos de valores enteros de SQL Server. Con números enteros con o sin signo
- **Date:** una fecha de calendario que contiene el año (de cuatro cifras), el mes y el día.
- **Time:** La hora del día en horas minutos segundos (el valor predeterminado es 0).

2. Consultas

Es el lenguaje de consulta universal para bases de datos.

Los mandatos de SQL se dividen en tres grandes grupos diferenciados.

- **DDL (Data Definition Language):** es el encargado de la definición de Bases de Datos, tablas, vistas e índices entre otros.
- **DML (Data Manipulation Language):** cuya misión es la manipulación de datos. A través de él podemos seleccionar, y actualizar insertar, eliminar datos.
- **DCL (Data Control Language):** encargado de la seguridad de la base de datos, en todo lo referente al control de accesos y privilegios entre los usuarios.

Lenguaje de definición de datos de mysql

Para **crear una base** de datos es la consulta es la siguiente: CREATE DATABASE, se puede escribir en mayúscula o minúscula.



```
CREATE DATABASE NombreDeBaseDeDatos;
```

Para **crear una tabla**:



```
CREATE TABLE 'alumnos' ('nombre' VARCHAR(30), 'apellido'  
VARCHAR(30), 'FechaNac' DATETIME, 'email' VARCHAR(30));
```

En primer momento colocamos el nombre de la tabla en este caso alumnos y entre paréntesis se debe especificar el nombre y el tipo de dato de cada columna de la tabla.

Para **seleccionar datos de una tabla** y devolver un resultado, que es una matriz de datos (filas y columnas).



```
SELECT nombre_de_columna(s)  
FROM nombre_de_tabla
```

Para seleccionar los datos de todas las columnas de una tabla, se usa un asterisco en lugar de los nombres de las columnas:



```
SELECT *  
FROM nombre_de_la_tabla
```

Para seleccionar datos de las tablas en donde sólo queramos recuperar registros en donde una columna tenga un valor en particular, usamos la **cláusula WHERE**.



```
SELECT nombre_de_columna(s)
FROM nombre_de_tabla
WHERE nombre_de_columna operador valor
```

Si se quisiera recuperar registros que cumplan con más de una condición, se utiliza AND u OR entre las condiciones del WHERE:



```
SELECT nombre_de_columna(s)
FROM nombre_de_tabla
WHERE nombre_de_columna operador valor
AND|OR nombre_de_columna operador valor
```

Ejemplos:



```
SELECT Nombre, Apellido
FROM alumnos
```

| Nombre | Apellido |
|----------|----------|
| Laura | Gomez |
| Victoria | Sanchez |



```
SELECT sueldo
FROM empleados
WHERE Puesto='Puesto1'
```

| Sueldo |
|--------|
| 20000 |

Operadores que se pueden usarse con WHERE:

| Operador | Descripción |
|----------------|---|
| = | Igual a |
| <> | Diferente de |
| > | Mayor que |
| < | Menor que |
| >= | Mayor que o igual a |
| <= | Menor que o igual a |
| BETWEEN | Entre cierto rango |
| LIKE | <p>Busca cierto patrón de caracteres en los datos. Al patrón buscado se lo precede o sucede con “%” según se desee:</p> <p>LIKE ‘%ma%’: Busca campos que contengan la sílaba “ma”.</p> <p>LIKE ‘%ma’: Busca campos que terminen con la sílaba “ma”.</p> <p>LIKE ‘ma%’: Busca campos que empiecen con la sílaba “ma”.</p> |

Consultas auxiliares


ORDER BY

Para lograr un ordenamiento específico de la respuesta podemos utilizar la palabra clave "ORDER BY"



```
ORDER BY nombre_de_columna ASC | DESC
```

Para dar dos ordenamientos sucesivos (por edad y por apellido, por ejemplo), vamos agregando las instrucciones una detrás de la otra en el orden en que queramos que se apliquen:



```
SELECT Nombre, Apellido, Edad  
FROM Empleados  
ORDER BY Edad ASC, Apellido ASC
```

LIMIT

Se utiliza para cuando sólo queremos una cantidad limitada de datos



```
SELECT nombre_de_columna(s)
FROM nombre_de_tabla
LIMIT numero_fila_inicial, numero_de_filas
```

número_fila_inicial es desde qué fila de las que nos devolvería queremos empezar

número_de_filas es el es la cantidad de filas que queremos recibir



```
SELECT ID_Empleado, Nombre, Apellido
FROM Empleados
LIMIT 2,1
```

| ID_Empleado | Nombre | Apellido |
|-------------|--------|----------|
| 2 | Laura | Conti |
| 3 | Juana | Díaz |

3. Agregar, Modificar o Eliminar registros

El **lenguaje de manipulación de datos** consiste en 3 consultas para manipular los datos.

INSERT

Inserta nuevos registros en una tabla



```
INSERT INTO nombre_de_tabla  
VALUES (valor1, valor2,....)
```



```
INSERT INTO Empleados VALUES (03, 'Carlos', 'García');
```

| ID_Empleado | Nombre | Apellido |
|-------------|--------|----------|
| 1 | Laura | Conti |
| 2 | Juana | Diaz |
| 3 | Carlos | García |

O también se pueden especificar las columnas a las que se les agrega datos:



```
INSERT INTO nombre_de_tabla (columna1, columna2,...)  
VALUES (valor1, valor2,...);
```



```
INSERT INTO Empleados (ID_Empleado, Nombre, Apellido) VALUES (03,  
'Carlos', 'García');
```

DELETE

Sirve para eliminar registros de una tabla.



```
DELETE FROM nombre_de_tabla  
WHERE nombre_de_columna operador valor
```



```
DELETE FROM Empleados  
WHERE ID_Empleado=2
```



| ID_Empleado | Nombre | Apellido |
|-------------|--------|----------|
| 1 | Laura | Conti |
| 3 | Carlos | García |

También se pueden eliminar todas las filas de una tabla:



```
DELETE *  
FROM nombre_de_tabla
```

UPDATE

Sirve para modificar datos de una tabla



```
UPDATE nombre_de_tabla  
SET nombre_de_columna=valor  
WHERE nombre_de_columna operador valor
```



```
UPDATE Empleados  
SET nombre=Lara  
WHERE ID_Empleado= 1;
```



| ID_Empleado | Nombre | Apellido |
|-------------|--------|----------|
| 1 | Lara | Conti |
| 3 | Carlos | García |

Instalación de MySQL y otras herramientas

Para poder probar de forma local las consultas y operaciones que acabamos de ver necesitamos tener instalado MySQL y algún cliente o gestor que nos permita tanto administrar las bases de datos como las tablas. Para esto utilizaremos un paquete de aplicaciones llamado **Wamp Server** que incluye MySQL como motor de base de datos, a la vez que instala y configura de forma automática el servidor Apache y el intérprete del lenguaje PHP. Tanto PHP como Apache nos serán de utilidad para ejecutar una herramienta llamada **phpMyAdmin**. Esta aplicación nos permite administrar bases de datos MySQL de forma muy sencilla.

WampServer puede ser descargado para windows de forma gratuita desde <https://www.wampserver.com/en/>. En caso de utilizar la plataforma de Mac, su equivalente es MAMP y puede ser descargado desde <https://www.mamp.info/>. Cualquiera sea el que instalemos deberemos seguir las instrucciones de instalación provistas en ambos sitios.

Una vez instalado WampServer, nos aseguramos de que los servicios están ejecutándose y abriremos la dirección **http://localhost**. Una vez ahí, en la sección de Tools o herramientas buscaremos el link a phpMyAdmin.


WampServer

Version 3.1.9 - 64bit english

Server Configuration

Apache Version: 2.4.39 - [Documentation](#)
Server Software: Apache/2.4.39 (Win64) PHP/7.2.18 - Port defined for Apache: 80
PHP Version: 7.2.18 - [Documentation](#)

Loaded Extensions :

- apache2handler
- com_dotnet
- date
- filter
- hash
- json
- mysqli
- PDO
- readline
- soap
- standard
- xml
- xsl
- bcmath
- Core
- dom
- gd
- iconv
- ldap
- mysqlnd
- pdo_mysql
- Reflection
- sockets
- tokenizer
- xmlreader
- Zend OPcache
- bz2
- ctype
- exif
- gettext
- imap
- libxml
- openssl
- pdo_sqlite
- session
- SPL
- wddx
- xmlrpc
- zip
- calendar
- curl
- fileinfo
- gmp
- intl
- mbstring
- pcre
- Phar
- SimpleXML
- sqlite3
- xdebug
- xmlwriter
- zlib

MySQL Version: 5.7.26 - Port defined for MySQL: 3306 - Default DBMS - [Documentation](#)
MariaDB Version: 10.3.14 - Port defined for MariaDB: 3307 - [Documentation](#)

Tools

- [phpinfo\(\)](#)
- [phpmyadmin](#)
- [Add a Virtual Host](#)

Your Projects

- [C:\wamp\www\...](#)
- [C:\wamp\www\...](#)
- [C:\wamp\www\...](#)
- [C:\wamp\www\...](#)

Your Aliases

- [adminer](#)
- [phpmyadmin](#)
- [phpsysinfo](#)

Your VirtualHost

- [localhost](#)

En caso de que nos pida las credenciales de acceso al servidor de MySQL, y no las hayamos modificado, ingresamos root como usuario y dejaremos vacío el campo de contraseña (MAMP generalmente usa también root como contraseña).

Una vez que ingresamos al servidor podemos ejecutar las consultas en la pestaña SQL.



Bibliografía utilizada y sugerida

Artículos de revista en formato electrónico:

SQL Commands Disponible desde la URL:

<http://www.postgresql.org/docs/9.1/static/sql-commands.html>

Tutorial de SQL Disponible desde la URL:

<http://www.unalmed.edu.co/~mstabare/Sql.pdf>

Wampserver Disponible desde la URL: <http://www.wampserver.es/>