

EPET N° 7
MATERIA BASE DE DATOS

SQL básico

Qué es SQL

El lenguaje de consulta estructurado (SQL) es un **lenguaje de base de datos** normalizado, utilizado por los diferentes motores de bases de datos para realizar determinadas operaciones sobre los datos (consultar, modificar y eliminar) o sobre la estructura de los mismos.

Mysql: un gestor de BD

- Un gestor de base de datos.
 - Sencillo
 - Gratuito
 - Rápido y eficiente.
 - Para linux/windows

```
carlos@carlos-desktop:~$  
carlos@carlos-desktop:~$ mysql -u root -p svows  
Enter password:  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 46 to server version: 5.0.22-Debian_0ubuntu6.06.2-log  
  
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.  
  
mysql> █
```

Una base de datos

- Un conjunto de tablas relacionadas entre sí:
 - Un Club de un equipo de fútbol en internet.

pais_id	nombre
1	España
2	Portugal
3	Andorra
4	Bélgica
5	Holanda
6	Francia
7	Gran Bretaña
8	Irlanda
9	Islandia
10	Dinamarca

provincia_id	nombre
1	Pontevedra
2	Orense
3	La Coruña
4	Lugo
5	Asturias
6	Cantabria
7	Guipúzcoa
8	Vitoria
9	Vizcaya
10	Teruel

socio_id	nombre	email	pais_id	provincia_id
57	Mateo Byler.	Mbyler@worldonline.Es	1	23
58	Sergio Lan.	slan@conncoll.edu	61	0
59	Tomaás de Jesús Alvarez Aja.		61	0
60	Jesús Alvarez Arenas.		61	0
61	Amando Rio Loano	amando@lagloria.com.mx	61	0
62	Alvaro Fuentes García.		1	42
63	Sergio Sánchez Valencia.		1	41
64	Gabriela de la Peña Astorga.		61	0
65	Jorge García Nuñez de Cáceres.		61	0
66	Jorge Alvarez Arreola.	jalvarez@campus.mty.itesm.mx	61	0

Componentes del SQL

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

COMANDOS

Existen tres tipos de comandos SQL.

Los “DDL” (Data Definition Language), lenguaje de definición de datos, que permiten crear y definir nuevas bases de datos, campos e índices.

Los “DML” (Data Manipulation Language) lenguaje de manipulación de datos, que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.

Los “DCL” (Data Control Language), lenguaje de control de datos, contiene elementos útiles para trabajar en un entorno multiusuario, en el que es importante la protección de los datos, la seguridad de las tablas y el establecimiento de restricciones en el acceso, así como elementos para coordinar la compartición de datos por parte de usuarios concurrentes, asegurando que no interfieren unos con otros.

Comandos DDL

Comandos DDL

Comando	Descripción del Comando
CREATE	Utilizado para crear nuevas tablas, campos e índices
DROP	Empleado para eliminar tablas e índices
ALTER	Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.

Comandos DML

Comandos DML

Comando	Descripción del Comando
SELECT	Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado.-
INSERT	Utilizado para cargar lotes de datos en la base de datos en una única operación.-
UPDATE	Utilizado para modificar los valores de los campos y registros Especificados.-
DELETE	Utilizado para eliminar registros de una tabla de una base de datos.-

Cláusulas

Las cláusulas son condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular.

Cláusula	Descripción de la Cláusula
FROM	Utilizada para especificar la tabla de la cual se van a seleccionar los Registros.-
WHERE	Utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar.-
GROUP BY	Utilizada para separar los registros seleccionados en grupos específicos.-
HAVING	Utilizada para expresar la condición que debe satisfacer cada grupo.-
ORDER BY	Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico.-

Operadores Lógicos

Operadores Lógicos

Operador	Uso
AND	Es el "y" lógico. Evalúa dos condiciones y devuelve un valor de verdad sólo si ambas son ciertas.-
OR	Es el "o" lógico. Evalúa dos condiciones y devuelve un valor de verdad si alguna de las dos es cierta.-
NOT	Negación lógica. Devuelve el valor contrario de la expresión.-

Operadores de Comparación

Operadores de Comparación

Operador	Uso
<	Menor que.-
>	Mayor que.-
<>	Distinto de.-
<=	Menor ó Igual que
>=	Mayor ó Igual que.-
=	Igual que.-
BETWEEN	Utilizado para especificar un intervalo de valores.-
LIKE	Utilizado en la comparación de un modelo.-
IN	Utilizado para especificar registros de una base de datos.-

Funciones de Agregado

Las funciones de agregado se usan dentro de una cláusula SELECT en grupos de registros para devolver un único valor que se aplica a un grupo de registros.

Función	Descripción
AVG	Utilizada para calcular el promedio de los valores de un campo Determinado.-
COUNT	Utilizada para devolver el número de registros de la selección.-
SUM	Utilizada para devolver la suma de todos los valores de un campo Determinado.-
MAX	Utilizada para devolver el valor más alto de un campo especificado.-
MIN	Utilizada para devolver el valor más bajo de un campo especificado.-

Consultar la base de datos

```
SELECT ... FROM ... WHERE ...
```

```
SELECT (campos) FROM (tablas) WHERE (condiciones) ...
```

Consultar la base de datos

SELECT

[ALL | DISTINCT | DISTINCTROW]

select_expr, ...

[FROM table_references

[WHERE where_condition]

[GROUP BY {col_name | expr | position}]

[HAVING where_condition]

[ORDER BY {col_name | expr | position}

[ASC | DESC], ...]

[LIMIT {[offset,] row_count | row_count OFFSET offset}]

[INTO OUTFILE 'file_name' export_options]

SELECT: ejemplos

```
SELECT socio_id,nombre,email FROM socios WHERE pais_id=14;
```

```
mysql> SELECT socio_id,nombre,email FROM socios WHERE pais_id=14;
+-----+-----+-----+
| socio_id | nombre                | email                |
+-----+-----+-----+
| 1179 | Martin Putz.          | pum@advge.magwien.gv.at |
| 1471 | Antonio J. Salgado Alonso. |                    |
| 2628 | Martin Wohlfart       | wohlfart@gmx.at       |
| 3529 | Christian Jochum      |                    |
| 3618 | Guillermo Ludwig Sanz-Orrio | willi.ludwig@aon.at   |
| 5206 | Marco Antonio Bernal  | marco.bernal@chello.at |
+-----+-----+-----+
6 rows in set (0.05 sec)
```

```
SELECT socio_id,nombre,email FROM socios WHERE nombre like 'Alicia%';
```

```
+-----+-----+-----+
| socio_id | nombre                | email                |
+-----+-----+-----+
| 2662 | Alicia Rubio Espinosa. |                    |
| 2971 | Alicia Pedraz         |                    |
| 3311 | Alicia Blanco Serrano | aliblan@arrakis.es   |
| 3933 | Alicia Peretoca       | Alisorlino@Hotmail.Com |
| 5089 | Alicia Ma.Lema Costa  | olivosbaires@sinectis.com.ar |
| 5293 | Alicia Pérez Blázquez | cuca_@mixmail.com    |
+-----+-----+-----+
6 rows in set (0.01 sec)
```

SELECT: ejemplos

```
SELECT count(*) FROM socios;
```

```
+-----+
| count(*) |
+-----+
|      6447 |
+-----+
1 row in set (0.00 sec)
```

```
SELECT pais_id, count(*) as nsocios FROM socios
group by pais_id order by nsocios desc limit 10;
```

```
+-----+-----+
| pais_id | nsocios |
+-----+-----+
|      1 |    3828 |
|    173 |     464 |
|      61 |     460 |
|      52 |     288 |
|       7 |     139 |
|      43 |     101 |
|      12 |     100 |
|      59 |      99 |
|       6 |      86 |
|       5 |      61 |
+-----+-----+
10 rows in set (0.00 sec)
```

SELECT: ejemplos

```
SELECT socio_id,nombre,email,pais_id FROM socios WHERE nombre like 'Alicia%';
```

```
+-----+-----+-----+-----+
| socio_id | nombre                | email                | pais_id |
+-----+-----+-----+-----+
| 2662    | Alicia Rubio Espinosa. |                      | 1       |
| 2971    | Alicia Pedraz          |                      | 1       |
| 3311    | Alicia Blanco Serrano  | aliblan@arrakis.es  | 1       |
| 3933    | Alicia Peretoca        | Alisorlino@Hotmail.Com | 43      |
| 5089    | Alicia Ma.Lema Costa   | olivosbaires@sinectis.com.ar | 43      |
| 5293    | Alicia Pérez Blázquez  | cuca_@mixmail.com   | 1       |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

```
SELECT s.socio_id, s.nombre, s.email, p.nombre as pais
FROM socios s, países p
WHERE p.pais_id=s.pais_id and s.nombre like 'Alicia%';
```

```
+-----+-----+-----+-----+
| socio_id | nombre                | email                | pais    |
+-----+-----+-----+-----+
| 2662    | Alicia Rubio Espinosa. |                      | España  |
| 2971    | Alicia Pedraz          |                      | España  |
| 3311    | Alicia Blanco Serrano  | aliblan@arrakis.es  | España  |
| 3933    | Alicia Peretoca        | Alisorlino@Hotmail.Com | Argentina |
| 5089    | Alicia Ma.Lema Costa   | olivosbaires@sinectis.com.ar | Argentina |
| 5293    | Alicia Pérez Blázquez  | cuca_@mixmail.com   | España  |
+-----+-----+-----+-----+
6 rows in set (0.06 sec)
```

SELECT: ejemplos

```
SELECT pais_id,count(*) as nsocios
FROM socios
group by pais_id
order by nsocios desc
limit 10;
```

pais_id	nsocios
1	3828
173	464
61	460
52	288
7	139
43	101
12	100
59	99
6	86
5	61

10 rows in set (0.00 sec)

```
SELECT p.nombre as pais,count(*) as nsocios
FROM socios s, paises p
WHERE p.pais_id=s.pais_id
group by p.pais_id
order by nsocios desc
limit 10;
```

pais	count(*)
España	3828
Portugal	26
Andorra	13
Bélgica	40
Holanda	61
Francia	86
Gran Bretaña	139
Irlanda	6
Dinamarca	18
Alemania	54

10 rows in set (0.00 sec)

SELECT: ejemplos

```
SELECT p.nombre as pais,count(*) as nsocios
FROM socios s, paises p
WHERE p.pais_id=s.pais_id
group by p.pais_id
HAVING nsocios between 10 and 20
order by nsocios desc
limit 10;
```

```
+-----+-----+
| pais          | nsocios |
+-----+-----+
| Honduras     |      20 |
| Dinamarca    |      18 |
| Japón        |      15 |
| República Dominicana |      14 |
| Andorra      |      13 |
| Noruega      |      12 |
| Finlandia    |      11 |
| Ecuador      |      10 |
+-----+-----+
8 rows in set (0.10 sec)
```

Cambiar los datos

UPDATE ... SET ... WHERE ...

UPDATE (tabla) SET (campo=valor) WHERE (condiciones) ...

Cambiar los datos

```
UPDATE tbl_name  
  SET col_name1=expr1 [, col_name2=expr2 ...]  
  [WHERE where_condition]  
  [ORDER BY ...]  
  [LIMIT row_count]
```

UPDATE: ejemplos

```
mysql> select * from socios where socio_id=1;
+-----+-----+-----+-----+-----+
| socio_id | email          | pais_id | provincia_id | nombre          |
+-----+-----+-----+-----+-----+
|      1 | crb@laeff.esa.es |      1 |      28 | Carlos Rodrigo Blanco |
+-----+-----+-----+-----+-----+
1 row in set (0.16 sec)
```

UPDATE socios **SET** pais_id=162 **WHERE** socio_id=1;

```
mysql> update socios set pais_id=162 where socio_id=1;
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from socios where socio_id=1;
+-----+-----+-----+-----+-----+
| socio_id | email          | pais_id | provincia_id | nombre          |
+-----+-----+-----+-----+-----+
|      1 | crb@laeff.esa.es |     162 |      28 | Carlos Rodrigo Blanco |
+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

UPDATE: ejemplos

UPDATE socios **SET** email=**LOWER**(email);

```
mysql> UPDATE socios SET email=LOWER(email);  
Query OK, 716 rows affected (0.58 sec)  
Rows matched: 6447  Changed: 716  Warnings: 0
```

UPDATE socios **SET** puntos=puntos+2 **WHERE** puntos < 2;

```
mysql> UPDATE socios set puntos=puntos+2 where puntos <8;  
ERROR 1054 (42S22): Unknown column 'puntos' in 'where clause'  
mysql> █
```

Insertar Datos

```
INSERT INTO ... (...,...) VALUES (...,...)
```

```
INSERT INTO (tabla) (campo1,...,campoN) VALUES  
(valor1,...,valorN) ...
```

```
INSERT INTO ... SET ...
```

```
INSERT INTO (tabla) SET  
campo1=valor1,...,campoN=valorN
```

Insertar Datos

```
INSERT INTO socios  
(socio_id, email, pais_id, provincia_id, nombre)  
VALUES (7000,'pepe@austria.com',14,0,'Pepe Martínez');
```

```
mysql> insert into socios (socio_id,email,pais_id,provincia_id,nombre) values (7000,'pepe@austria.com',14,0,'Pepe Martínez');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from socios where socio_id=7000;  
+-----+-----+-----+-----+-----+  
| socio_id | email          | pais_id | provincia_id | nombre      |  
+-----+-----+-----+-----+-----+  
|      7000 | pepe@austria.com |      14 |              0 | Pepe Martínez |  
+-----+-----+-----+-----+-----+  
1 row in set (0.01 sec)
```

Borrar Datos

DELETE FROM ...WHERE ...

DELETE FROM tabla WHERE conditions

DELETE FROM socios WHERE socio_id=7000;

```
mysql> delete from socios where socio_id=7000;  
Query OK, 1 row affected (0.01 sec)
```

Ver la estructura de la Base de Datos

SHOW TABLES

```
mysql> show tables;
+-----+
| Tables_in_svows |
+-----+
| paises          |
| provincias      |
| socios          |
+-----+
3 rows in set (0.22 sec)
```

Ver la estructura de la Base de Datos

DESC tabla

```
mysql> desc socios;
```

Field	Type	Null	Key	Default	Extra
socio_id	int(7)	NO		0	
email	char(69)	NO		NULL	
pais_id	int(4)	NO		0	
provincia_id	int(4)	NO		0	
nombre	char(255)	NO		NULL	

5 rows in set (0.02 sec)