

## Navegación

Mostrar bases de datos:

SHOW DATABASES;

Mostrar tablas:

SHOW TABLES;

Mostrar la estructura de una tabla:

SHOW FIELDS FROM table / DESCRIBE table;

Mostrar la estructura de la creación de la tabla:

SHOW CREATE TABLE table;

Mostrar lista de procesos:

SHOW PROCESSLIST;

Matar proceso:

KILL process\_number;

## Condiciones

campo1 = valor1

campo1 <> valor1

campo1 LIKE 'valor \_ %'

campo1 IS NULL

campo1 IS NOT NULL

campo1 IS IN (valor1, valor2)

campo1 IS NOT IN (valor1, valor2)

condición1 AND condición2

condición1 OR condición2

## Crear / Abrir / Borrar base de datos

Crear base de datos llamada “ejemplo”:

CREATE DATABASE ejemplo;

Crear base de datos llamada “ejemplo” con codificación de caracteres “utf8”:

CREATE DATABASE DatabaseName CHARACTER SET utf8;

Seleccionar la base de datos “ejemplo”:

USE ejemplo;

Borrar base de datos llamada “ejemplo”:

DROP DATABASE ejemplo;

Modificar base de datos llamada “ejemplo” agregando la codificación de caracteres “utf8”:

ALTER DATABASE DatabaseName CHARACTER SET utf8;

## Copia de seguridad de la base de datos en un archivo SQL

mysqldump -u root -p ejemplo > ejemplo.sql

## Restaurar desde una copia de seguridad del archivo SQL

mysql -u root -p ejemplo < ejemplo.sql;

## Select

Consultar todos los campos de una tabla:

SELECT \* FROM tabla;

Consultar todos los campos de dos tablas:

SELECT \* FROM tabla1, tabla2;

Consultar dos campos de dos tablas:

SELECT campo1, campo2 FROM tabla1, tabla2;

Consultar con una condición:

SELECT ... FROM ... WHERE condición;

Consultar con una condición agrupando con un campo:

SELECT ... FROM ... WHERE condición GROUP BY campo;

Consultar con una condición agrupando con un campo con otra condición:

SELECT ... FROM ... WHERE condición GROUP BY campo

HAVING condición2;

Consultar con una condición ordenado con dos campos:

SELECT ... FROM ... WHERE condición ORDER BY

campo1, campo2;

Consultar con una condición ordenado con dos campos de forma descendiente:

SELECT ... FROM ... WHERE condición ORDER BY

campo1, campo2 DESC;

Consultar con una condición mostrando solamente 10 resultados:

SELECT ... FROM ... WHERE condición LIMIT 10;

Consultar devolviendo valores diferentes de un campo:

SELECT DISTINCT campo1 FROM ...

Consultar devolviendo valores diferentes de dos campos:

SELECT DISTINCT campo1, campo2 FROM ...

## Select - Join

Consultar tabla1 uniendo la tabla2 uniendo sus claves con una condición:

SELECT ... FROM t1 JOIN t2 ON t1.id1 = t2.id2 WHERE condición;

Consultar tabla1 uniendo la tabla2 uniendo sus claves con una condición y dando preferencia los resultados de la tabla1:

SELECT ... FROM t1 LEFT JOIN t2 ON t1.id1 = t2.id2

WHERE condición;

Consultar tabla1 uniendo con una subconsulta la tabla 2 y la tabla 3:

SELECT ... FROM t1 JOIN (t2 JOIN t3 ON ...) ON ...

## Reparación de tablas después de un cierre no limpio

mysqlcheck --all-databases;

mysqlcheck --all-databases --fast;

## Claves

Crear tabla con clave primaria en dos campos:

CREATE TABLE tabla (... PRIMARY KEY (campo1, campo2))

Crear tabla con clave foránea en dos campos referenciando a una tabla con dos campos:

CREATE TABLE tabla (... FOREIGN KEY (campo1, campo2)

REFERENCES tabla2 (t2\_campo1, t2\_campo2))

## Crear / Borrar / Modificar Tabla

**Crear tabla con dos campos, uno tipo INT, otro tipo VARCHAR:**

```
CREATE TABLE tabla (campo1 INT(2), campo2 VARCHAR(20));
```

**Crear tabla con dos campos, uno tipo INT, otro tipo VARCHAR, creando un índice:**

```
CREATE TABLE tabla (campo1 INT(2), campo2 VARCHAR(20), INDEX (campo1));
```

**Crear tabla con dos campos, uno tipo INT, otro tipo VARCHAR, creando una clave primaria:**

```
CREATE TABLE tabla (campo1 INT(2), campo2 VARCHAR(20), PRIMARY KEY (campo1));
```

**Crear tabla con dos campos, uno tipo INT, otro tipo VARCHAR, creando una clave primaria en dos campos:**

```
CREATE TABLE tabla (campo1 INT(2), campo2 VARCHAR(20), PRIMARY KEY (campo1, campo2));
```

**Crear tabla con dos campos, estableciendo una clave foránea que referencia a otra tabla de un campo con clave primaria:**

```
CREATE TABLE tabla1 (fk_campo1 INT(2), campo2 CHAR(10) FOREIGN KEY (fk_campo1) REFERENCES tabla2 (t2_campoA)) [ON UPDATE | ON DELETE] [CASCADE | SET NULL];
```

**Crear tabla con dos campos, estableciendo una clave foránea que referencia a otra tabla de un campo con clave primaria:**

```
CREATE TABLE tabla1 (fk_campo1 INT(2), fk_campo2 CHAR(10) FOREIGN KEY (fk_campo1, fk_campo2) REFERENCES tabla2 (t2_campoA, t2_campoB));
```

**Crear tabla si no existe:**

```
CREATE TABLE tabla IF NOT EXISTS;
```

**Crear una tabla temporal:**

```
CREATE TEMPORARY TABLE tabla;
```

**Borrar tabla:**

```
DROP TABLE tabla;
```

**Borrar tabla si existe:**

```
DROP TABLE IF EXISTS tabla;
```

**Borrar varias tablas:**

```
DROP TABLE tabla1, tabla2, ...
```

**Modificar un campo de una tabla:**

```
ALTER TABLE tabla MODIFY campo1 INT(2);
```

**Modificar un campo de una tabla y que no contenga valores nulos:**

```
ALTER TABLE tabla MODIFY campo1 INT(2) NOT NULL ...
```

**Modificar el nombre de un campo de una tabla:**

```
ALTER TABLE tabla CHANGE viejo_campo1 nuevo_campo1 INT(2);
```

**Modificar el nombre de un campo de una tabla y que no contenga valores nulos:**

```
ALTER TABLE tabla CHANGE viejo_campo1 nuevo_campo1 INT(2) NOT NULL ...
```

**Modificar un campo por valores por defecto:**

```
ALTER TABLE tablea ALTER campo1 SET DEFAULT ...
```

**Modificar un campo quitando los valores por defecto:**

```
ALTER TABLE tablea ALTER campo1 DROP DEFAULT;
```

**Añadir un campo a una tabla:**

```
ALTER TABLE tabla ADD nuevo_campo1 VARCHAR(20);
```

**Añadir un campo a una tabla y colocarlo como primer campo:**

```
ALTER TABLE tabla ADD nuevo_campo1 VARCHAR(20) FIRST;
```

**Añadir un campo a una tabla despues de otro campo:**

```
ALTER TABLE tabla ADD nuevo_campo1 VARCHAR(20) AFTER otro_campo;
```

**Borrar un campo de una tabla:**

```
ALTER TABLE tabla DROP campo1;
```

**Añadir un índice a un campo de una tabla:**

```
ALTER TABLE tabla ADD INDEX (campo);
```

**Modificar un campo a una tabla y colocarlo como primer campo:**

```
ALTER TABLE tabla MODIFY campo1 VARCHAR(20) FIRST;
```

**Modificar un campo a una tabla despues de otro campo:**

```
ALTER TABLE tabla MODIFY campo1 VARCHAR(20) AFTER otro_campo;
```

**Modificar el nombre de un campo a una tabla y colocarlo como primer campo:**

```
ALTER TABLE tabla CHANGE campo1 nuevo_campo1 VARCHAR(20) FIRST;
```

**Modificar el nombre campo a una tabla despues de otro campo:**

```
ALTER TABLE tabla CHANGE viejo_campo1 nuevo_campo1 VARCHAR(20) AFTER otro_campo;
```

## Insertar

Insertar valores en una tabla:

```
INSERT INTO tabla1 (campo1, campo2) VALUES (valor1, valor2);
```

## Actualizar

Actualizar un valor de un campo de una tabla:

```
UPDATE tabla1 SET campo1=nuevo_valor1 WHERE condición;
```

Actualizar valores de varias tablas:

```
UPDATE tabla1, tabla2 SET campo1=nuevo_valor1, campo2=nuevo_valor2 WHERE tabla1.id1 = tabla2.id2 AND condición;
```

## Borrar

Borrar una tabla:

```
DELETE FROM tabla;
```

Borrar todos los datos de una tabla:

```
TRUNCATE tabla;
```

Borrar datos de una tabla establecida de una condición:

```
DELETE FROM tabla WHERE condición;
```

Borrar datos de varias tablas:

```
DELETE FROM tabla1, tabla2 WHERE tabla1.id1 = tabla2.id2 AND condición;
```

## Principales tipos de datos

TINYINT (1byte: -128 to +127)	FLOAT (M,D)	TIME (HH:MM)
SMALLINT (2bytes: +-65 000)	DOUBLE (M,D)	YEAR (AAAA)
MEDIUMINT (3bytes: +-16 000 000)	FLOAT (D=0->53)	DATE (AAAA-MM-DD)
INT (4bytes: +- 2 000 000 000)		DATETIME (AAAA-MM-DD HH:MM; años 1000->9999)
BIGINT (8bytes: +-9.10 <sup>18</sup> )		TIMESTAMP (como DATETIME, pero 1970->2038, compatible con Unix)
TINY (máximo=255)		
MEDIUM (máximo=~16000)		
LONG (máximo=4Go)	ENUM ('valor1', 'valor2', ...) -- (por defecto NULL o NOT NULL)	
VARCHAR (una línea; tamaño explícito)		
TEXT (multilíneas; tamaño máximo=65535)		
BLOB (binario; tamaño máximo=65535)		

## Usuarios y permisos

Crear usuario:

```
CREATE USER 'usuario'@'localhost';
```

Establecer todos los permisos a un usuario:

```
GRANT ALL PRIVILEGES ON base.* TO 'usuario'@'localhost' IDENTIFIED BY 'contraseña';
```

Establecer permisos establecidos a un usuario:

```
GRANT SELECT, INSERT, DELETE ON base.* TO 'usuario'@'localhost' IDENTIFIED BY 'contraseña';
```

Quitar permisos establecidos a un usuario:

```
REVOKE ALL PRIVILEGES ON base.* FROM 'usuario'@'host'; -- menos el permisos de establecer permisos.
```

Quitar permisos establecidos a un usuario:

```
REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'usuario'@'host'; -- todos los permisos, incluso el de establecer permisos.
```

Volver a leer la tabla de permisos:

```
FLUSH PRIVILEGES;
```

## Restablecer la contraseña root

Detener el servicio:

```
/etc/init.d/mysql stop
```

Omitir tablas otorgadas, para modificar la tabla mysql:

```
mysqld_safe --skip-grant-tables
```

Establecer una nueva contraseña para root:

```
UPDATE mysql.user SET
```

```
password=PASSWORD('new_pass') WHERE user='root';
```

Arrancar el servicio:

```
/etc/init.d/mysql start
```