

Sentencia DELETE

La sentencia DELETE permite eliminar una o varias ocurrencias (filas) de una o varias tablas.

Borrado de filas de una única tabla:

```
DELETE [LOW_PRIORITY] FROM nombre_tabla [WHERE condición] [ORDER BY expresión] [LIMIT] número_filas
```

- Ejemplo:

```
DELETE FROM empleado WHERE dni = 12345678;
```

Borrado de filas de varias tablas (opción 1):

```
DELETE [LOW_PRIORITY] [nombre_tabla[.*]] [, nombre_tabla[.*]]... {FROM | USING} referencia_tablas [WHERE condición]
```

- Ejemplo:

```
DELETE t1, t2 FROM t1 INNER JOIN t2 INNER JOIN t3 WHERE t1.id=t2.id AND t2.id=t3.id;
```

Borrado de filas de varias tablas (opción 2):

```
DELETE [LOW_PRIORITY] FROM nombre_tabla[.*] [, nombre_tabla[.*]]... USING referencia_tablas [WHERE condición]
```

- Ejemplo:

```
DELETE FROM t1, t2 USING t1 INNER JOIN t2 INNER JOIN t3 WHERE t1.id=t2.id AND t2.id=t3.id;
```

La opción LOW_PRIORITY permite indicar el nivel de prioridad que tiene la operación de borrado. Con esta opción el servidor espera que no haya clientes leyendo de la tabla para realizar la operación de borrado. Solo afecta a tablas que tengan motores que utilizan bloqueo a nivel de tabla (MyISAM, MEMORY, MERGE).

Cuando se realiza el borrado de filas de varias tablas, la sentencia DELETE borra las filas de las tablas nombradas en referencia_tablas que cumplen con las condiciones establecidas en la cláusula WHERE.

En la cláusula FROM se nombran las tablas en las que se van a borrar filas.

Cuando se van a borrar filas de varias tablas utilizando la opción 1, se establecen las relaciones entre las tablas mediante la sentencia JOIN. La cláusula WHERE es opcional y especifica las filas que deben borrarse.

Cuando se usan los métodos para borrar varias filas en varias tablas, se tienen que poner en primer lugar las condiciones de enlace entre las tablas y después el resto de las condiciones.

IMPORTANTE: si no se escribe la cláusula WHERE, se borran todas las filas indicadas.

La cláusula ORDER BY es opcional e indica el orden en el que se borran las filas. No se puede usar en el borrado en múltiples tablas.

La cláusula LIMIT establece el número de filas a borrar. No se puede utilizar en el borrado en múltiples tablas.



Borrado lógico de filas de una tabla

La sentencia DELETE borra físicamente las filas de las tablas. En las bases de datos existe otra opción para “eliminar” las filas de las tablas: el borrado lógico.

El borrado lógico consiste en marcar la fila poniendo en una columna un dato que la identifica como “no disponible”. Por ejemplo, puede ser una columna que contenga una fecha de baja o una columna que tome un valor lógico (verdadero o falso, 0 o 1) que indique si la columna está eliminada o no.

Ejemplo:

```
UPDATE cliente SET fecha_baja = CURDATE() WHERE id= 1234;
```

Este tipo de borrado puede resultar muy útil en ciertas circunstancias, por ejemplo, si el cliente vuelve a realizar una compra, podemos evitar borrar físicamente los datos del cliente y tener que volver a insertarlos cuando se quiere dar de alta de nuevo, además de poder conservar las referencias a ese cliente en el resto de las tablas.

El problema que produce este tipo de borrado es que en cada consulta que se hace en la que se utilice esa tabla hay que comprobar el estado de esa columna y comprobar las restricciones de integridad referencial. De no hacer esto, se puede producir una inconsistencia de datos, por ejemplo, porque puede haber filas en ventas que hacer referencia a clientes “eliminados”. También hay que tener en cuenta que ciertas legislaciones, como es el caso de la Ley Orgánica de Protección de Datos (LOPD) en España, puede obligar a realizar el borrado físico de ciertos tipos de datos.

Para gestionar las bajas lógicas es recomendable usar disparadores (*triggers*), vistas o procedimientos almacenados.