

Sentencia INSERT

La sentencia INSERT permite insertar nuevas filas en las tablas existentes.

Existen varias opciones para realizar la inserción: los valores pueden escribirse como un conjunto de valores cerrados entre paréntesis para cada nueva fila, una colección de expresiones de asignación, o como una sentencia SELECT.

Opción especificando los valores a insertar de forma explícita con VALUE:

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [INTO] nombre_tabla
[(lista_columnas)] {VALUES | VALUE} ({expresión | DEFAULT | NULL}, ...),
(...), ... [ON DUPLICATE KEY UPDATE col_name=expr] [, col_name=expr] ...]
```

Opción especificando los valores a insertar de forma explícita con SET:

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [INTO] nombre_tabla SET
col_name= {expresión | DEFAULT | NULL}, ... [ON DUPLICATE KEY UPDATE
col_name=expr] [, col_name=expr] ...]
```

Opción insertando filas con datos contenidos en otras tablas:

```
INSERT [LOW_PRIORITY | HIGH_PRIORITY] [INTO] nombre_tabla
[(lista_columnas)] SELECT ... [ON DUPLICATE KEY UPDATE col_name=expr] [,
col_name=expr] ...]
```

Consideraciones sobre la sintaxis anterior:

Las partes opcionales LOW_PRIORITY, DELAYED y HIGH_PRIORITY permiten indicar el nivel de prioridad que tiene la operación de inserción.

Cuando se utiliza la opción LOW_PRIORITY, el servidor espera a que no haya clientes leyendo la tabla para realizar la inserción. Cuando se utiliza la opción HIGH_PRIORITY se anula el efecto de la opción *–low-priority-updates* si estuviese habilitada. Estas opciones afectan solo a las tablas con motores que utilizan el bloqueo a nivel de tabla (MyISAM, MEMORY, MERGE).

Cuando se utiliza la opción DELAYED, el servidor guarda la sentencia en un buffer de memoria; realizando la inserción cuando la tabla quede libre. Esta opción está en desuso en la actualidad y ciertas bases de datos ya no la soportan.

nombre_tabla es el nombre de la tabla en la cual se van a insertar los datos.

lista_columnas es la relación de columnas en las que se van a insertar datos, separadas por comas. Se utilizan cuando se especifican los datos como una lista de valores (VALUES) o cuando se cogen los datos mediante una consulta (SELECT).

Las columnas que no figuran en la lista anterior toman el valor que tengan definido por defecto, o el valor por defecto implícito si no tienen definido un valor por

defecto. Los valores por defecto implícitos son 0 para los tipos numéricos y la cadena vacía (") para los tipos de datos de cadenas de caracteres.

Las columnas de tipo **AUTO_INCREMENT** no deben aparecer en la lista de columnas, y en este caso el servidor les asigna el valor siguiente al de la última que se insertó.

En el caso de no poner una lista de columnas a continuación del nombre de la tabla, se considera que se van a insertar datos en todas las columnas, **en el orden en el que se crearon**.

La lista de columnas se utiliza en el caso de que no se introduzcan valores para todas las columnas, o se escriban los valores en un orden distinto al que tienen las columnas en el esquema de la tabla.

En la opción 2, **SET** permitirá indicar explícitamente los nombres de las columnas y los valores que van a tomar mediante expresiones de asignación.

En la opción 1, VALUES | VALUE permitirán indicar explícitamente la lista de valores que van a tomar las columnas de la fila, separados por comas y cerrados entre paréntesis. VALUE es un sinónimo de VALUES.

La expresión correspondiente a cada columna debe devolver un valor del mismo tipo que la columna. En caso de no ser del mismo tipo, el SGBD realizará la conversión de tipos.

En una expresión se puede hacer referencia a una columna de las que aparecen antes en la lista de columnas. También se pueden utilizar las palabras reservadas NULL para hacer referencia al valor nulo, o DEFAULT para hacer referencia al valor por defecto.

Ejemplo:

```
INSERT INTO nombre_tabla (col1, col2, col3, col4) VALUES (15, col1*2, NULL, DEFAULT);
```

Una excepción en el uso de referencias y otras columnas son las columnas que tienen la propiedad AUTO_INCREMENT. Cualquier referencia a esas columnas toma el valor 0. El número de valores tiene que coincidir con el número de columnas relacionadas en la lista de columnas.

En la opción 3, las filas que se van a insertar y los valores que toman las columnas de esas filas, se obtienen de la ejecución de una sentencia SELECT.

Cuando se utiliza la cláusula ON DUPLICATE KEY UPDATE y se hace la inserción de una fila que puede provocar un valor duplicado en una clave PK o UNIQUE, se hace una actualización en la fila que ya existe, modificando los valores de las columnas que se indican en la cláusula. Ejemplo:

```
INSERT INTO tabla1 (a,b,c) VALUES (1,2,3) ON DUPLICATE KEY UPDATE c=c+1
```

La sentencia anterior modifica el contenido de la columna c, sumándole una unidad, en el caso que exista una fila con el valor 1 en la columna a, suponiendo que ésta es la clave primaria.