

# Handlers en MySQL

Un handler en MySQL es invocado cuando se da un evento o condición definida con anterioridad. Este evento está siempre asociado con una condición de error, pero puedes tener tres formas de definir el error:

- Como código de error de MySQL
- Como código SQLSTATE ANSI-standard
- Como nombre de condición, ejemplo: SQLEXCEPTION, SQLWARNING y NOT FOUND.

MySQL tiene sus propios códigos de error que son únicos para MySQL Server. Por ejemplo:

```
DECLARE CONTINUE HANDLER FOR 1062 SET duplicate_key=1;
```

Un código de error SQLSTATE es definido por ANSI standard y son independientes de la base de datos, lo que significa que deberías tener el mismo valor de error para cualquier base de datos ANSI compatible.

Por ejemplo:

```
DECLARE CONTINUE HANDLER FOR SQLSTATE '23000' SET duplicate_key=1;
```

Así, Oracle, SQL Server, DB2, y MySQL reportarán el mismo SQLSTATEvalue (23000) cuando hay un error de primary key duplicada.

Sintaxis para declarar handlers:

```
declare {exit | continue } handler for {error-number | {SQLSTATE error-string}}
```

## SQLSTATE vs códigos de error MySQL

En teoría es mejor utilizar códigos SQLSTATE porque son independientes de la plataforma y hacen a nuestro código más portable. Sin embargo, hay algunas razones para utilizar el código de error de MySQL, específicamente en la escritura de Stored Procedures.

- El lenguaje para la escritura de procedimientos almacenados en Oracle y SQL Server es totalmente incompatible con el de MySQL. El de DB2, es algo más compatible (ambos basados en SQL:2003 standard).
- No todos los códigos de error MySQL tienen su equivalente en código SQLSTATE. Cada error de Mysql está relacionado con un código de error SQLState, pero no siempre esta relación es uno a uno. HY000 es un código de error SQLSTATE para propósitos generales que devuelve MySQL cuando su código de error no tiene asociado un código SQLSTATE.

## Errores comunes MySQL y códigos SQLSTATE

Código de error MySQL	Código SQLSTATE	Mensaje de error
1011	<b>HY000</b>	Error on delete of '%s' (errno: %d)
1021	HY000	Disk full (%s); waiting for someone to free some space . . .
1022	<b>23000</b>	Can't write; duplicate key in table '%s'
1027	HY000	'%s' is locked against change
1036	<b>HY000</b>	Table '%s' is read only
1048	<b>23000</b>	Column '%s' cannot be null
1062	<b>23000</b>	Duplicate entry '%s' for key %d
1099	<b>HY000</b>	Table '%s' was locked with a READ lock and can't be updated
1100	HY000	Table '%s' was not locked with LOCK TABLES
1104	<b>42000</b>	The SELECT would examine more than MAX_JOIN_SIZE rows; check your WHERE and use SET SQL_BIG_SELECTS=1 or SET SQL_MAX_JOIN_WHERE and use SET SQL_BIG_SELECTS=1 or SET SQL_MAX_JOIN_SIZE=# if the SELECT is okay
1106	<b>42000</b>	Incorrect parameters to procedure '%s'
1114	HY000	The table '%s' is full
1150	HY000	Delayed insert thread couldn't get requested lock for table %s
1165	HY000	INSERT DELAYED can't be used with table '%s' because it is locked with LOCK TABLES
1242	<b>21000</b>	Subquery returns more than 1 row
1263	22004	Column set to default value; NULL supplied to NOT NULL column '%s' at row %ld
1264	22003	Out of range value adjusted for column '%s' at row %ld
1265	1000	Data truncated for column '%s' at row %ld
1312	<b>0A000</b>	SELECT in a stored program must have INTO
1317	70100	Query execution was interrupted
1319	42000	Undefined CONDITION: %s
1325	<b>24000</b>	Cursor is already open

1329	<b>2000</b>	No data to FETCH
1336 1337	<b>42000 42000</b>	USE is not allowed in a stored program Variable or condition declaration after cursor or handler declaration
1338	<b>42000</b>	Cursor declaration after handler declaration
1339	20000	Case not found for CASE statement
1348 1357	HY000 HY000	Column '%s' is not updatable Can't drop a %s from within another stored routine
1358 1362	HY000 HY000	GOTO is not allowed in a stored program handler
1363	<b>HY000</b>	Updating of %s row is not allowed in %s trigger There is no %s row in %s trigger
To signal a generic <b>SQLSTATE</b> value, use ' <b>45000</b> ', which means "unhandled user-defined exception."		

Por ejemplo:

```
mysql> CREATE TABLE test.t (s1 int, primary key (s1));
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> delimiter |
```

```
mysql> CREATE PROCEDURE handlerdemo ()
-> BEGIN
->   DECLARE CONTINUE HANDLER FOR SQLSTATE '23000' SET @x2 = 1;
->   SET @x = 1;
->   INSERT INTO test.t VALUES (1);
->   SET @x = 2;
->   INSERT INTO test.t VALUES (1);
->   SET @x = 3;
-> END;
-> |
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> CALL handlerdemo() |
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> SELECT @x |
+-----+
| @x    |
+-----+
| 3     |
+-----+
1 row in set (0.00 sec)
```