

# Manejo de errores

# Índice

- 01 [Definición](#)
- 02 [Creación de handlers](#)
- 03 [Creación de conditions](#)



01

# Definición

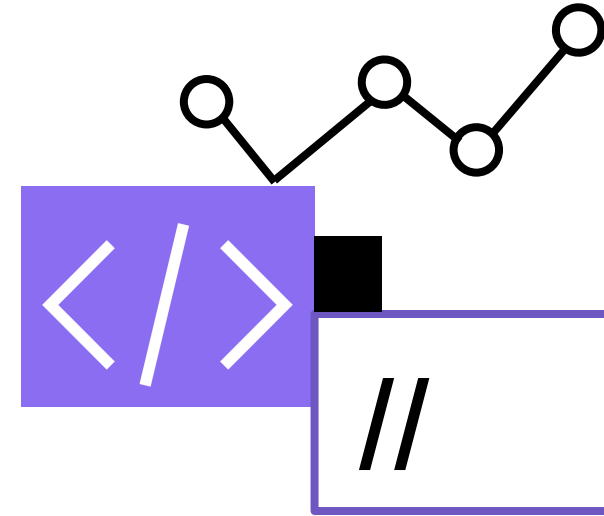
# Definición

Para manejar los errores de SQL en nuestros programas dentro de una base de datos, MySQL nos permite crear handlers y conditions.

- **Handlers:** es el manejador de errores. Por cada error que se ejecute, y si esa condición está definida en un handler, el código que definimos será ejecutado.
- **Conditions:** son nombres que podemos agregar a los errores para dejar más legible nuestro código.

02

# Creación de handlers



# Creación de handlers

SQL

```
DECLARE handler_action HANDLER
  FOR valorCondicion
  statement

valorCondicion: {
  mysql_error_code
| SQLSTATE [VALUE] sqlstate_value
| condition_name
| SQLWARNING
| NOT FOUND
| SQLEXCEPTION
}
```

# Creación de handlers

- **handler\_action:** puede ser los siguientes valores: continue, exit y undo.
- **Continue:** Ante la ejecución del handler, este continúa con la ejecución del bloque BEGin..End.
- **Exit:** termina la ejecución del bloque begin..End que disparó el handler.
- **valorCondicion:** Son los errores que vamos a capturar con nuestro handler. Ante alguno de estos errores, se ejecutará el código que definamos. Para ello, tenemos varios tipos de condiciones.

# Creación de handlers - Tipos de condiciones

**1- Mysql\_error\_code:** es un valor numérico que indica un error de sql. podemos definir por ejemplo el valor 1051. Son los que están en la siguiente documentación: clic [aquí](#).

**2- SQLState:** es un string que indica un error. por ejemplo el valor 42S02. Los valores posibles son los que están en la siguiente documentación: clic [aquí](#).

**3- Condition\_name:** Si definimos una condition, podemos agregar el nombre de la misma

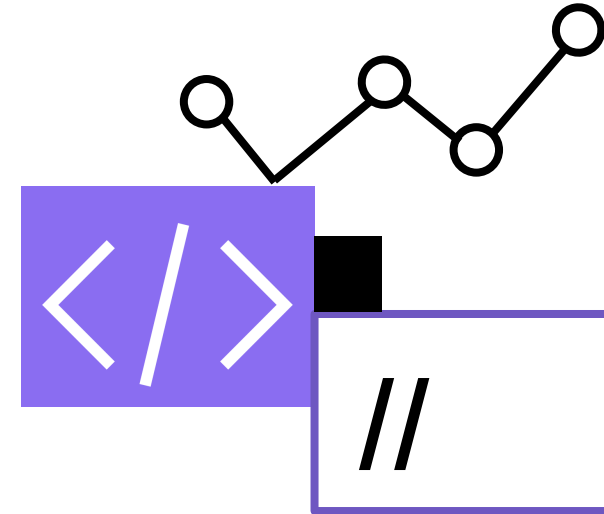


# Creación de handlers - Tipos de condiciones

**4- SQLWarning:** es un abreviado para incluir a todos los códigos de SQLSTATE que empiezan con '01'

**5- Not Found:** es un abreviado para incluir a todos los códigos de SQLSTATE que empiezan con '02'

**6- SQLEXCEPTION:** es un abreviado que excluye a todos los códigos de SQLSTATE que empiezan con '00', '01', '02'



# Ejemplo de handler

SQL

```
USE adventureworks;
DELIMITER $$
CREATE PROCEDURE usp_tabla_insertar(tablaId int)
BEGIN

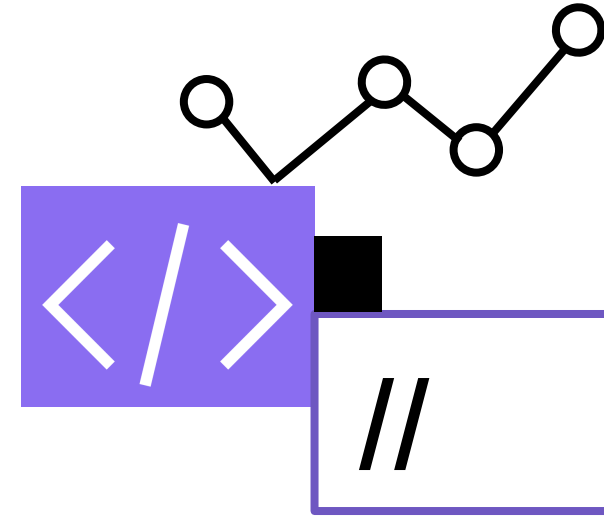
DECLARE EXIT HANDLER FOR SQLEXCEPTION
BEGIN
SELECT 'Se ejecuto KO - Error SQL';
END;

INSERT INTO tabla(id) VALUES (tablaId);
SELECT 'Se ejecuto OK';

END; $$
```

03

# Creación de conditions



# Creación de conditions

SQL

```
DECLARE nombreCondicion CONDITION FOR valorCondicion

valorCondicion: {
    mysql_error_code
| SQLSTATE [VALUE] sqlstate_value
}
```

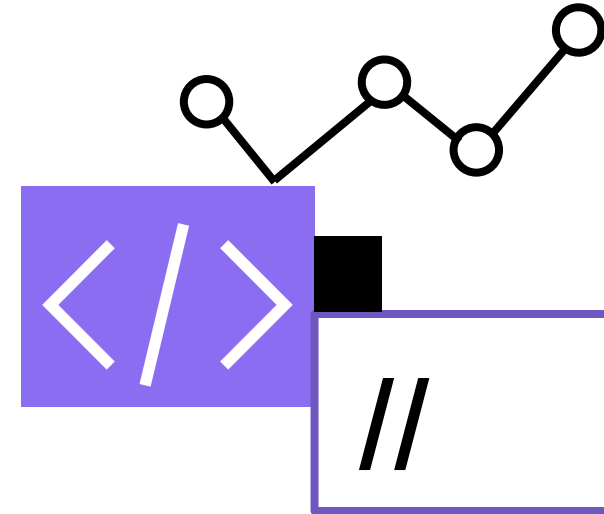
# Creación de conditions - Tipos de condiciones

→ **valorCondicion:** Son los errores que vamos a capturar con nuestro handler. Ante alguno de estos errores, se ejecutará el código que definamos.

Para ello, tenemos varios tipos de condiciones:

**1- [mysql\\_error\\_code](#):** es un valor numérico que indica un error de sql. Podemos definir, por ejemplo, el valor 1051. Son los que están en [esta](#) documentación.

**2- [SQLState](#):** es un string que indica un error. Por ejemplo, el valor 42S02. Los valores posibles son los que están en la documentación que podés acceder haciendo clic [aquí](#).



# Ejemplo de handler con conditions

SQL

```
USE adventureworks;
DELIMITER $$
CREATE PROCEDURE usp_tabla_insertar(tablaId int)
BEGIN

DECLARE no_existe_tabla CONDITION FOR 1146;
DECLARE EXIT HANDLER FOR no_existe_tabla
BEGIN
SELECT 'Se ejecuto KO - Error SQL';
END;

INSERT INTO tabla(id) VALUES (tablaId);
SELECT 'Se ejecuto OK';

END; $$
```

¡Muchas gracias!