

Condition Handling

Condition Handling (Manejador de excepciones)

La gestión de excepciones en MySQL consta de dos partes:

- Declaración de la condición de excepción.
- Declaración del manejador de la excepción.

Declaración de la condición de excepcion

```
DECLARE condition_name CONDITION FOR condition_value
```

```
condition_value: mysql_error_code | SQLSTATE [VALUE] sqlstate_value
```

- Declaraciones de condiciones deben de aparecer antes de las declaraciones de cursores y antes de las declaraciones de los manejadores.
- *condition_value* puede ser o un *error_code* (número) o un SQLSTATE (cadena de 5 caracteres). [Listado de errores](#)
- Realmente, lo que estamos haciendo es renombrar el error, con ello ganamos legibilidad.

Declaración del manejador

DECLARE handler_action HANDLER FOR condition_value [, condition_value] ...
Statement

handler_action: CONTINUE | EXIT | UNDO

condition_value: mysql_error_code | SQLSTATE [VALUE] sqlstate_value |
condition_name | SQLWARNING | NOT FOUND | SQLEXCEPTION

- Cualquiera de las condiciones "activan" el manejador y se ejecutan las instrucciones albergadas en el "statement".
- "Statement" puede estar formado por una sola instrucción o todo un bloque BEGIN...END.
- Debe ser declarado después de las variables, después de la declaración de condiciones y después de los cursores.

- "handler_action" indica cuál es la acción a realizar después de ejecutar el "statement". Las opciones son:
 - CONTINUE: la ejecución del programa continúa.
 - EXIT: la ejecución del bloque donde el manejador esté declarado termina.
 - UNDO: no está implementado.
- "condition_value" puede ser:
 - mysql_error_code: número que identifica un error. Ya lo hemos visto en la declaración de condiciones.
 - SQLSTATE: bis
 - SQLWARNING: todos los errores SQLSTATE que empiecen por 01.
 - NOT FOUND: todos los SQLSTATE que empiecen por 02. Por ejemplo, los cursores cuando no hay más registros disponibles lanzan un error "No data" que tiene asociado el SQLSTATE 02000.
 - SQLEXCEPTION: todos los errores que no empiezan ni por 00, 01 ó 02.
- Para ignorar un error:
- ```
DECLARE CONTINUE HANDLER FOR SQLWARNING BEGIN END
```
- Dentro del bloque de instrucciones de un manejador no puede haber referencias a etiquetas externas al bloque. Mirar manual.

# Problemas: un poco de todo

1. Reescribir el procedimiento que imprimía los usuarios del sistema utilizando el manejador de excepciones.
2. Crea una función que determine si un número es primo devolviendo 0 o 1.
3. Usando la función anterior crea otra que calcule la suma de los primeros m números primos empezando en el 1.
4. Crea un procedimiento para generar y almacenar en la tabla primos (primos(id, número)) de la base test los números primos comprendidos entre 1 y m (parámetro de entrada). Modifica posteriormente el procedimiento para almacenar en la variable de salida @np el número de primos almacenado
5. Crea un disparador para que, cada vez que un cliente alquile 5 películas se le otorgue un préstamo gratuito. (bbdd videoclub)

# Generar “errores”: SIGNAL

```
SIGNAL condition_value [SET signal_information, signal information ...]
```

```
signal_information: condition_item = simple_value
```

- SIGNAL devuelve un error.
- *condition\_value* indica el error que debe ser devuelto. Puede ser cualquiera de los vistos anteriormente en la declaración del manejador.
- Se puede utilizar el valor SQLSTATE '45000' (“unhandled user-definition exception”) para devolver un error genérico
- *condition\_item*: MESSAGE\_TEXT | MYSQL\_ERRNO | SCHEMA\_NAME | TABLE\_NAME
- Podemos utilizar esta instrucción desde terminal. Ejemplo:

```
mysql> signal SQLSTATE '45000' set message_text='Se ha producido un error';
```

Nos devuelve : ERROR 1644 (45000): Se ha producido un error