Handlers para manejar errores en MySQL

Seguimos con MySQL,hemos visto cómo hacer procedimientos almacenados y cómo controlar el flujo de estos. Ahora le toca el turno a los handlers…

(https://dev.mysql.com/doc/refman/8.0/en/declare-handler.html)

Cuando trabajamos con procedimientos almacenados en MySQL tenemos que tener en cuenta que durante la ejecución de nuestra aplicación **se pueden producir errores**.

Por ejemplo, si estamos trabajando con el motor de base de datos innoDB y definimos claves ajenas, podrán producirse **errores de integridad referencial** si intentamos hacer un insert en un campo que clave ajena y el valor que intentamos introducir no existe en la tabla que referencia.

Para **llevar el control de estos** errores podemos **definir handlers** en nuestros procedimientos almacenados.

Cada error en **MySQL desprende un código de error**, el cual tendremos que anotar para manejarlo en nuestro handler.

Sintaxis para declarar handlers:

```
DECLARE handler_action HANDLER
    FOR condition_value [, condition_value] ...
    statement

handler_action: {CONTINUE | EXIT}

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

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

La acción **exit** significa que cuando se acabe la ejecución del código del handler **se sale del procedimiento almacenado,** si ponemos **continue** en lugar de exit, **la ejecución de procedimiento almacenado proseguiría (habitualmente se utiliza con sqlwarning)**.

Ejemplo de un handler

Creamos la siguiente tabla:

```
create table t2(
    s1 int,
    primary key (s1)
)engine=innodb;
```

Después creamos esta otra que contiene una clave ajena que hace referencia a la anterior :

```
create table t3 (
    s1 int, key(s1),
    foreign key (s1) references t2 (s1)
```

```
)engine=innodB;
```

Después intentamos insertar un valor en la tabla t3, se trata de un valor que no se encuentra en la columna de t2 que referencia… lo que arroja el siguiente error en la consola de mysql:

```
insert into t3 values(5);
```

Error:

ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`pruebas/t3`, CONSTRAINT `t3_ibfk_1` FOREIGN KEY (`s1`) REFERENCES `t2` (`s1`))

Anota el código de error desprendido porque es lo que necesitamos para crear un handler que maneje este error, en este caso **1452**.

Para manejar el error **creamos un handler** que almacenara el error en una tabla que vamos a crear donde iremos guardando un log de errores…

```
 create table error_log(error_message varchar(80));
```

Ahora creamos el procedimiento almacenado que se encargara de introducir datos en la BD y manejar excepciones:

```
delimiter //
create procedure procedimientoConHandler(in parametro1 int)
begin
    declare exit handler for 1452
    begin
        insert into error_log values (concat('Time: ',current_date,'. Error
        de clave ajena para el valor= ', parametro1));
    end;
    insert into t3 values (parametro1);
end;
//
```

Creamos dentro del procedimiento un manejador de errores para el tipo de error deseado para que cuando se produzca el error introduzca en una tabla que indiquemos un log de errores y no falle la aplicación.

De esta manera para introducir valores en la base de datos lo haces llamando al procedimiento para que maneje los errores y tener un **mayor control sobre la aplicación**.

Ya tenemos nuestro **procedimiento almacenado creado con control de errores**.