Atitit mysql异常处理

目录

[1. 常见错误编码与异常 1](#_Toc21582)

[1.1. 四、异常处理的命名 2](#_Toc3641)

[1.2.](#_Toc5156) *[忽略异常](#_Toc5156)*[begin end块 2](#_Toc5156)

[1.3. 五、异常传播 3](#_Toc3281)

[2. 循环查询列表对方法 exit HANDLER +loop 4](#_Toc21289)

# 常见错误编码与异常

EXIT

condition\_value:

mysql\_error\_code

| SQLSTATE [VALUE] sqlstate\_value

| condition\_name

| SQLWARNING

| NOT FOUND

| SQLEXCEPTION

关于错误编号和SQLSTATE码：

　　每个MySQL错误都有一个唯一的数字错误编号(mysql\_error\_code)，每个错误又对应一个5字符的SQLSTATE码(ANSI SQL 采用)。

SQLSTATE码对应的处理程序：

　　1、SQLWARNING处理程序：以‘01’开头的所有sqlstate码与之对应；

　　2、NOT FOUND处理程序：以‘02’开头的所有sqlstate码与之对应；

　　3、SQLEXCEPTION处理程序：不以‘01’或‘02’开头的所有sqlstate码，也就是所有未被SQLWARNING或NOT FOUND捕获的SQLSTATE(常遇到的MySQL错误就是非‘01’、‘02’开头的)

注意：‘01’、‘02’开头和‘1’、‘2’开头是有区别的，是不一样的错误sqlsate码。

常遇到的MySQL错误SQLEXCEPTION

## 四、异常处理的命名

为了提高可读性，可以给某个sqlstate代码或mysql错误代码一个名字，并且在后面的异常处理程序中使用这个名字。

## *忽略异常*begin end块

*e.g：DECLARE CONTINUE HANDLER FOR SQLWARNING BEGIN END;*

也就是说，当遇到SQLWARNING的问题时，进行的异常处理是begin end块，因为里面什么都没有，就类同于直接忽略。

## 五、异常传播

　　在嵌套块的情况下，内部块中发生异常了，首先由本块的异常处理程序来处理，如果本块没有处理，则由外部块的异常处理程序来处理。

mysql> DELIMITER $$

mysql> CREATE PROCEDURE small\_mistake6()

-> BEGIN

-> 　　DECLARE CONTINUE HANDLER FOR SQLSTATE '23000'

-> 　　SET @processed = 100;   
 ->

-> 　　BEGIN

-> 　　　　DECLARE CONTINUE HANDLER FOR SQLSTATE '21000'

-> 　　　　SET @processed = 200;

-> 　　　　INSERT INTO TEAMS VALUES(2,27,'third');

-> 　　　　set @test=123321;

-> END;

-> END$$

mysql> DELIMITER ;

mysql> call small\_mistake6;

mysql> select @processed,@test;+------------+--------+

| @processed | @test |

+------------+--------+

| 300 | 123321 |

+------------+--------+

解析：@processed=100说明内部块里的异常传播到了外部块，交由外部块的异常处理程序进行的处理。

墙裂建议：

当有多层begin end的时候，每层都应该有自己完善的异常处理，做到：自己的异常，自己这层去处理。

# 循环查询列表对方法 exit HANDLER +loop

DECLARE exit HANDLER FOR NOT FOUND

BEGIN

select 'rows finish hadler ';-- leave loop\_name;

-- cant use leave sttmt..

end;