Atitit mysql 死锁检测与恢复

BEGIN

#Routine body goes here...

-- 定义变量

declare timeout int default 15; -- sql执行超时时间

DECLARE exittag int DEFAULT 0;

DECLARE pid int;

DECLARE sec int;DECLARE sec2 int;

-- 定义游标，并将sql结果集赋值到游标中

DECLARE CURSOR11 CURSOR FOR select trx\_mysql\_thread\_id,now()-trx\_started as secs from information\_schema.innodb\_trx;

-- select id,bal from test.user;

-- select trx\_mysql\_thread\_id,now()-trx\_started as secs from information\_schema.innodb\_trx;

-- select id,bal from test.user;

-- select trx\_mysql\_thread\_id,now()-trx\_started as secs from information\_schema.innodb\_trx;

-- 声明当游标遍历完后将标志变量置成某个值

-- DECLARE CONTINUE HANDLER FOR NOT FOUND SET exittag=1;

-- 打开游标

open CURSOR11;

-- fetch CURSOR11 into pid,sec;

SELECT exittag,PID,SEC;

loop\_name:loop

BEGIN

-- this just like derfer...

DECLARE exit HANDLER FOR NOT FOUND

BEGIN

set exittag=1 ;

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

-- cant use leave sttmt..

end;

if exittag=1 THEN

select 'exitag is 1 will leave';

leave loop\_name;

end if;

-- select 'enter while';

-- 将游标中的值赋值给变量，注意：变量名不要和返回的列名同名，变量顺序要和sql结果列的顺序一致

-- set @pid=0;

-- set @sec=0;

fetch CURSOR11 into pid,sec;

-- 当s不等于1，也就是未遍历完时，会一直循环

SELECT exittag,pid,sec;

-- 执行业务逻辑

if sec>timeout then

begin

DECLARE exit HANDLER FOR SQLEXCEPTION set @e=1;

select CONCAT(' will kill pid:',pid);

-- insert logx set log= CONCAT(' will kill pid:',pid);

kill pid;

insert logx set log= CONCAT(' kill pid:',pid);

end;

end if ;

-- dbg ,use select or log

-- select pid,sec;

end ;

end loop;

select 'finish';

-- 关闭游标

close CURSOR11;

END