Atitit 存储过程 循环 游标

While最简单，，repeat也麻烦

#Routine body goes here...

set @i=0;

WHILE @i < 5 DO

SELECT @i;

SET @i=@i+1;

END WHILE;

CREATE DEFINER=`root`@`%` PROCEDURE `cursorT`()

BEGIN

#Routine body goes here...

DECLARE idx VARCHAR(30);

DECLARE bls VARCHAR(30);

DECLARE cur1 CURSOR FOR select id,balance from fs\_book\_t limit 3;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET @NOT\_FOUND=1;

set @NOT\_FOUND=0;

-- set @id=7;

OPEN cur1;

REPEAT

/\* this memberno must declare ,,not cant @ derfalut var mode\*/

FETCH cur1 INTO idx,bls;

select idx,bls;

-- print(11);

UNTIL @NOT\_FOUND END REPEAT;

CLOSE cur1;

select 'end';

END

fetch是获取游标当前指向的数据行，并将指针指向下一行，当游标已经指向最后一行时继续执行会造成游标溢出。  
使用loop循环游标时，他本身是不会监控是否到最后一条数据了，像下面代码这种写法，就会造成死循环；

read\_loop:loop

fetch cur into n,c; set total = total+c; end loop;

* 1
* 2
* 3
* 4

在MySql中，造成游标溢出时会引发mysql预定义的NOT FOUND错误，所以在上面使用下面的代码指定了当引发not found错误时定义一个continue 的事件，指定这个事件发生时修改done变量的值。

mysql游标的用法及作用 - ```求知若渴，虚怀若谷``` - CSDN博客.html