use jxgl

--首先判断有没有已经建立up\_getallstudents存储过程，有则先删除

if exists

(select name from sysobjects where name = 'up\_getallstudents' and type = 'p' )

drop procedure up\_getallstudents

--编写存储过程up\_getallstudents,用于获取学生表students的所有记录

create procedure up\_getallstudents

as

select \* from students

--使用execute执行存储过程up\_getallstudents

exec up\_getallstudents

--也可写成

execute up\_getallstudents



--编写一个存储过程up\_insertstudent,完成学生表students数据的插入

--1、不带默认值的参数

create procedure up\_insertstudent

@sid varchar(15), @sname varchar(30), @ssex char(10),

@sbirth datetime, @sbirthplace varchar(300)

as

begin

insert into students

(stu\_id, stu\_name, stu\_sex, stu\_birth, stu\_birthplace)

values

(@sid, @sname, @ssex, @sbirth, @sbirthplace)

end

exec up\_insertstudent '200712110111', '肖玉峰', '男', '1975-02-05', '山东省滕州市木石镇'

--等同于

exec up\_insertstudent

@sname = '肖玉峰', @sid = '200712110111', @ssex = '男', @sbirth = '1975-02-05', @sbirthplace = '山东省滕州市木石镇'

drop procedure up\_insertstudent

delete students where stu\_name = '肖玉峰'

insert into students(stu\_id, stu\_name, stu\_sex, stu\_birth, stu\_birthplace)

values('200712110110', '马缪', '男', '1986-010-17 00:00:00.000', '广东广州')

--编写一个存储过程up\_delstudentbyname,根据输入的学生姓名，删除该学生记录

**if object\_id('up\_delstudentbyname', 'p') is not null**

**drop procedure up\_delstudentbyname**

go

create procedure up\_delstudentbyname

@sname varchar(30)

as

begin

delete from students

where stu\_name = @sname

end

--调用存储过程的代码如下：

exec up\_delstudentbyname '马缪'

select \* from students

--编写一个存储过程up\_getstuinformationbyname,根据输入的学生姓名，显示该学生的学号、姓名、课程名和成绩

if object\_id('up\_getstuinformationbyname', 'p') is not null

drop procedure up\_getstuinformationbyname

go

create procedure up\_getstuinformationbyname

@sname varchar(30)

as

begin

select "S".stu\_id, stu\_name, cour\_name, score

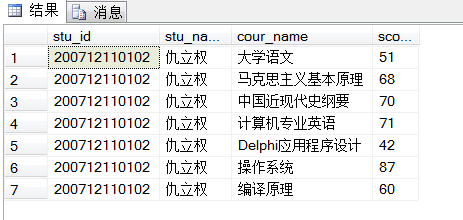
from students "S", courses "C", course\_score "CS"

where "S".stu\_id = "CS".stu\_id and "CS".cour\_id = "C".cour\_id and stu\_name = @sname

end

--调用存储过程的代码如下：

exec up\_getstuinformationbyname '仇立权' --ok



--2、带默认值的参数

--编写一个存储过程up\_insertstuwithdefault，给参数定义默认值，完成学生表students数据的插入

drop procedure up\_insertstuwithdefault

go

create procedure up\_insertstuwithdefault

@sid varchar(15), @sname varchar(30), @ssex char(10) = '男',

@sbirth datetime, @sbirthplace varchar(300) = '', @semail varchar(50) = ''

as

begin

insert into students

(stu\_id, stu\_name, stu\_sex, stu\_birth, stu\_birthplace, stu\_email)

values

(@sid, @sname, @ssex, @sbirth, @sbirthplace, @semail)

end

--调用存储过程的代码如下：

exec up\_insertstuwithdefault @sid = '11', @sname = '赵小乐', @sbirth = '1976-07-05'

select \* from students

--3、带输出参数

--在创建存储过程时，可以用关键字OUTPUT来创建一个输出参数，另外，调用时也必须给出OUTPUT关键字

--根据给定的学生姓名，获取该生的平均成绩

use jxgl

go

create procedure up\_getAvgScorebyname

@aname varchar(30), @avgscore int output

as

begin

select @avgscore = avg(score) from students "S", course\_score "CS"

where "S".stu\_id = "CS".stu\_id and "S".stu\_name = @aname

end

--调用过程代码

declare @avgscore int ---@avgscore可以改成任意变量名

exec up\_getAvgScorebyname '仇立权', @avgscore output --ok

print @avgscore

--不能写成print '@avgscore' 或print （'@avgscore'），这是输出字符串

D:\系统目录\桌面\YY截图20140510205028.png

--等同于

select avg(score) from students s, course\_score c

where s.stu\_id = c.stu\_id and s.stu\_name = '仇立权'

--4、带返回值的存储过程

--（）print语句可以将用户定义的消息返回给客户端

--编写一个存储过程up\_insertstudent2,在插入学生数据前，先判断一下学号是否存在，

--如果存在，输出“要插入的学生的学号已经存在”；否则，插入学生数据，并返回”恭喜，数据插入成功“

create procedure up\_insertstudent2

@sid varchar(15), @sname varchar (30), @ssex char(10) = '男', @sbirth datetime

as

begin

if exists(select \* from students where stu\_id = @sid)

print ('要插入的学生的学号已经存在')

--也可写成raiserror('要插入的学生的学号已经存在', 16, 1)

else

begin

insert into students

(stu\_id, stu\_name, stu\_sex, stu\_birth)

values

(@sid, @sname, @ssex, @sbirth)

print('恭喜，数据插入成功')

--raiserror('恭喜，数据插入成功', 16, 10)

end

end

drop procedure up\_insertstudent2

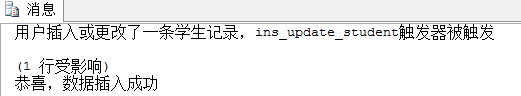
select \* from students

--调用过程代码如下：

exec up\_insertstudent2 @sid = '13', @sname = '张小飞', @sbirth = '1983-02-01' --ok



exec up\_insertstudent2 @sid = '10', @sname = '张小龙', @sbirth = '1988-01-08' --ok

D:\系统目录\桌面\YY截图20140510205028.png

--（）return 语句，return语句可以从过程、批处理或语句块中退出，不执行其后继语句

create procedure up\_delstudentbyname2

@sname varchar(30)

as

begin

delete from students

where stu\_name = @sname

return @@rowcount --行计数

end

drop procedure up\_delstudentbyname2

--调用存储过程的代码如下：

declare @ret\_val int

exec @ret\_val = up\_delstudentbyname2 '张小龙' --ok

select @ret\_val as "删除的行"

--也可写成print @ret\_val

--5、带变量的存储过程

/\*在存储过程可以定义变量，包括全局变量（@@变量名）和局部变量（@变量名）。

用于保存存储过程中的临时结果

编写存储过程up\_getavgscorebyname2，根据输入的学生姓名，计算该学生的平均成绩。

根据该生平局成绩与全体学生平均成绩的关系，返回相应信息\*/

use jxgl

go

create procedure up\_getavgscorebyname2

@aname varchar(30), @resStr varchar(30) output

as

begin

declare @curAvg decimal(18,2)

declare @totalAvg decimal(18,2)

select @totalAvg = avg(score) from course\_score

select curAvg = avg(score) from students "S", course\_score "CS"

where "S".stu\_id = "CS".stu\_id and "S".stu\_name = @aname

if @curAvg > @totalAvg

set @resStr = '高于平均分'

else

set @resStr = '低于平均分'

print ('总平均分为' + convert(varchar(18), @totalAvg))

print ('该生平均分' + convert(varchar(18), @curAvg))

print @resStr

drop procedure up\_getavgscorebyname2

--调用存储过程的代码

declare @resstring varchar(30)

exec up\_getavgscorebyname2 '甘明', @resstring output

--这句和要求不一样，不知道哪里出问题

--6、使用output游标参数

--output游标参数用来将存储过程的局部游标传递回执行调用的批处理、存储过程或触发器

/\*编写一个带有output型游标参数的存储过程up\_getstudent\_cursor, 再编写一个过程

up\_printstudentbycursor, 对游标中的数据进行显示\*/

--存储过程up\_getstudent\_cursor的代码如下

create procedure up\_getstudent\_cursor

@student\_cursor cursor varying output

as

begin

set @student\_cursor = cursor

forward\_only static for

select stu\_id, stu\_name, stu\_birth, stu\_sex

from students;

open @student\_cursor;

end

--存储过程printstudentbycursor代码如下：

create procedure printstudentbycursor

as

declare @Mycursor cursor

declare @axh varchar(15)

declare @asname varchar(30)

declare @asbir datetime

declare @assex char(10)

begin

exec getstudent\_cursor @student\_cursor = @Mycursor output

fetch next from @Mycursor into @axh, @asname, @asbir, @assex

while(@@fetch\_status = 0)

begin

print('学号:' + @axh + '姓名:' + @asname + '出生年月:' + convert(varchar(30), @asbir, 120)+ '性别' + @assex)

fetch next from @Mycursor into @axh, @asname, @asbir, @assex

end

close @Mycursor;

deallocate @Mycursor;

end

--调用存储过程的代码如下

exec printstudentbycursor