

## 一、实验题目：存储过程

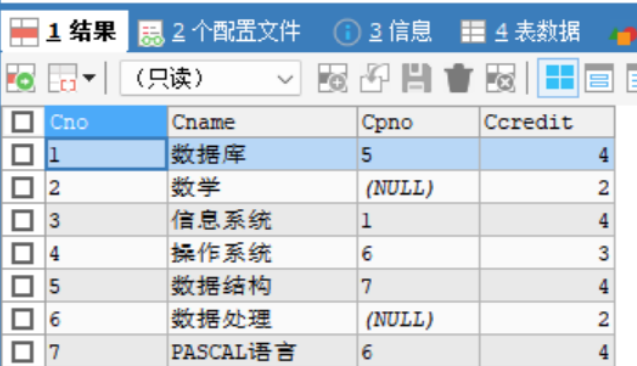
## 二、实验目的

1. 掌握过程化 SQL 语句;
2. 掌握存储过程创建、调用和删除的方法;

## 三、实验内容（利用截屏和文字对实验过程进行说明）

1. 创建存储过程 p\_AllCourse，用它显示所有的课程信息;


```
1 DELIMITER //
2 CREATE PROCEDURE p_AllCourse()
3 BEGIN
4     SELECT * FROM course;
5 END//
6 DELIMITER;
7 CALL p_AllCourse()
```



Cno	Cname	Cpno	Ccredit
1	数据库	5	4
2	数学	(NULL)	2
3	信息系统	1	4
4	操作系统	6	3
5	数据结构	7	4
6	数据处理	(NULL)	2
7	PASCAL语言	6	4

2. 创建存储过程 p\_FindCourse，根据用户输入的课程名进行模糊查询，显示相应的课程信息;

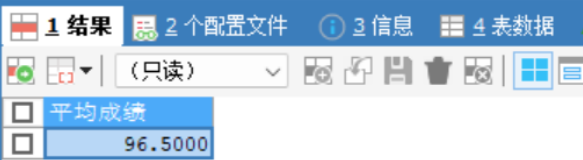
```
1 DELIMITER //
2 CREATE PROCEDURE p_FindCourse(IN c_name CHAR(20))
3 BEGIN
4     SELECT * FROM course
5     WHERE cname LIKE CONCAT('%', c_name, '%');
6 END//
7 DELIMITER;
8 CALL p_FindCourse('信息')
```



Cno	Cname	Cpno	Ccredit
3	信息系统	1	4

3. 创建存储过程 p\_CAvg，统计任意一门课程的平均成绩;

```
1 DELIMITER //
2 CREATE PROCEDURE p_CAvg(IN in_cno CHAR(4))
3 BEGIN
4     SELECT AVG(grade) AS '平均成绩'
5     FROM sc
6     WHERE cno=in_cno;
7 END//
8 DELIMITER;
9 CALL p_CAvg(2)
```



平均成绩
96.5000

4. 设计存储过程实现测试数据的批量插入：向用户表 users 增加指定条测试数据。用户表 users 包含三个属性：用户编号（系统自动编号）、用户名、密码。用户名为串 'yh' 及用户编号构成的字符串，密码为 8 位随机字符串。

```

1  DELIMITER //
2  CREATE PROCEDURE p_1(IN num INT)
3  BEGIN
4      DECLARE i INT DEFAULT 1;
5      WHILE (i<=num) DO
6          INSERT INTO USER(USER,authentication_string) VALUES (CONCAT('yh',i),LEFT(UUID(),8));
7          SET i=i+1;
8      END WHILE;
9  END//
10 DELIMITER;
11 CALL p_1(2)

```

1 信息 2 表数据 3 信息

限制行 第一行:

Host	User	Select_priv	Insert_priv	Update_priv	Delete_priv	Create_priv
	yh2	N	N	N	N	N
	yh1	N	N	N	N	N
localhost	root	Y	Y	Y	Y	Y
localhost	Tom	N	N	N	N	N
localhost	mysql.infoschema	Y	N	N	N	N
localhost	mysql.session	N	N	N	N	N
localhost	mysql.sys	N	N	N	N	N
localhost	Ben	N	N	N	N	N
*		N	N	N	N	N

5. 创建存储过程 p\_scdj, 将学生选课成绩从百分制改为等级制（即 A、B、C、D、E），数据保存在新表 SCDJ 中。新建表 SCDJ, 包含三个属性：学号、课号和等级。

```

1  DELIMITER //
2  CREATE PROCEDURE p_scdj()
3  BEGIN
4      CREATE TABLE SCDJ(
5          sno CHAR(9),
6          cno CHAR(9),
7          grade_level CHAR(4));
8      INSERT INTO SCDJ(sno,cno,grade_level)
9      SELECT sno,cno,
10         CASE
11             WHEN Grade>=90 THEN 'A'
12             WHEN Grade>=80 THEN 'B'
13             WHEN Grade>=70 THEN 'C'
14             WHEN Grade>=60 THEN 'D'
15             ELSE 'E'
16         END AS grade_level
17      FROM sc;
18  END//
19  DELIMITER;
20  CALL p_scdj();

```

1 信息 2 表数据 3 信息

sno	cno	grade_level
201215121	1	A
201215121	2	A
201215121	3	B
201215122	2	A
201215122	3	B

6. 创建存储过程 p\_UpdBadGrade, 给定学号、课号, 修改其成绩。要求如下:

- 若表中无相应选课记录, 显示“没有修改, 无此选课记录!”;
- 若表中相应记录成绩为空, 显示“没有修改, 还未考试!”;
- 若相应记录成绩小于 60 分, 则改为 60 分, 显示“已改为及格”;
- 若相应记录成绩大于 60 分且小于 80 分, 则改为 80 分, 显示“已改为良好”;
- 否则, 显示“成绩良好, 不需修改”。

```
1  DELIMITER //
2  CREATE PROCEDURE p_UpdBadGrade(IN in_sno CHAR(9), IN in_cno CHAR(9))
3  BEGIN
4      DECLARE grade_val INT;
5      SELECT grade INTO grade_val
6      FROM SC
7      WHERE sno=in_sno AND cno=in_cno;
8      IF grade_val IS NULL THEN
9          SELECT '没有修改, 还未考试!' AS '结论';
10     ELSEIF grade_val<60 THEN
11         UPDATE sc SET grade=60 WHERE sno=in_sno AND cno=in_cno;
12         SELECT '已改为及格' AS '结论';
13     ELSEIF grade_val>=60 AND grade_val<80 THEN
14         UPDATE SC SET grade=80 WHERE sno=in_sno AND cno=in_cno;
15         SELECT '已改为良好' AS '结论';
16     ELSE
17         SELECT '成绩良好, 不需修改' AS '结论';
18     END IF;
19     IF NOT EXISTS(SELECT * FROM sc WHERE sno=in_sno AND cno=in_cno) THEN
20         SELECT '没有修改, 无此选课记录!' AS '结论';
21     END IF;
22 END//
23 DELIMITER;
24 CALL p_UpdBadGrade('201215121', '2')
```

#### 四、实验分析与总结（遇到的困难及解决方法、对知识点的理解）

数据库中部分语法和 c 语言较为相像, 可以近似学习, 例如存储过程类似函数定义, 同时, 部分语法函数相同, 如 case 函数, 在使用时也要注意不能出现使用错误。复杂的数据库语言也是由基础的增删改查组成的, 所以也要注意基础的语言的使用, 在此基础上编写复杂的数据库语言才能解决问题。