

实验七 存储过程

1. 创建无参存储过程 p1 并调用

功能为获取 employees 表中的员工人数来初始化一个局部变量，如果小于 3 人输出“太少”， ≥ 3 且 < 10 输出“可以”， ≥ 10 输出“满员”。

```
1 • use yggl;
2 • drop procedure if exists p1;
3
4   delimiter ##
5
6 • create procedure p1()
7   begin
8       declare num int;
9       set num = (select count(*) from employees);
10      case
11      when num < 3 then
12          select num as "太少";
13      when num >= 10 then
14          select num as "满员";
15      else
16          select num as "可以";
17      end case;
18  end ##
19
20 • call p1();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Cor

满员
12

2. 创建无参存储过程 p2 并调用,

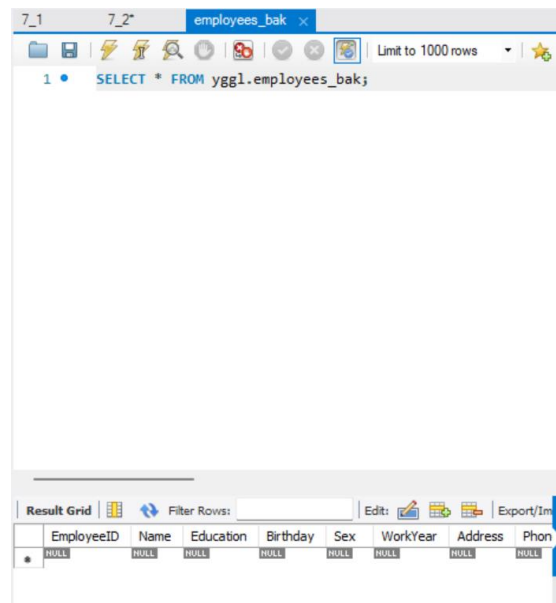
功能为创建与 employees 一样结构的表 employees_bak，将住在中山路的员工插入 employees_bak 后删除

```
1 • use yggl;
2 • drop procedure if exists p2;
3 • SET SQL_SAFE_UPDATES = 0;
4   delimiter ##
5
6 • create procedure p2()
7   begin
8       create table if not exists employees_bak like employees;
9       insert into employees_bak
10      select * from employees
11      where employees.Address like "%中山路%";
12      select * from employees_bak;
13      delete from employees_bak;
14  end ##
15
16 • call p2();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

EmployeeID	Name	Education	Birthday	Sex	WorkYear	Address	PhoneNumber	DepartmentID
000001	王浩	大专	1966-01-23	1	8	中山路32-1-508	83355668	2
108991	钟敏	硕士	1979-08-10	0	4	中山路10-3-105	83346722	3

插入 employees_bak 后删除





3.创建表 randnumber

字段: id 自增长,
data int;

创建无参存储过程 p3 并调用, 功能为: 向表中插入 50 个的随机数 (1-30), 但如果插入的数为 18, 则终止插入。

```
1 • use yggl;
2 • drop procedure if exists p3;
3
4   delimiter ##
5
6 • create procedure p3()
7   begin
8     declare t,n int;
9     set n = 1;
10    create table if not exists randnumber
11    (
12      `ID` int not null AUTO_INCREMENT primary key,
13      `data` int not null
14    );
15    loc: while n <= 50 do
16      set t = 1 + floor(29 * rand());
17      insert into randnumber values(n,t);
18      set n = n + 1;
19      if t = 18 then
20        leave loc;
21      end if;
22    end while;
23  end ##
24
25 • call p3();
```

Result Grid			 Filter Rows
	ID	data	
▶	1	15	
	2	29	
	3	13	
	4	6	
	5	20	
	6	25	
	7	7	
	8	16	
	9	2	
	10	17	
	11	21	
	12	27	
	13	12	
	14	9	
	15	6	
	16	6	
	17	8	
	18	24	
	19	8	
	20	26	
	21	18	
⊛	NULL	NULL	

4. 创建存储过程 p4(in name char(10),out income decimal(7,2))

计算一个员工的实际收入，并调用该存储过程，将员工朱骏的实际收入保存在一个用户变量中。

```
1 • use yggl;
2 • drop procedure if exists p4;
3
4   delimiter $$
5
6 • create procedure p4(in _name char(10),out _income decimal(7,2))
7   begin
8       declare id char(6);
9       set id = (select e.employeeid from employees e where e.`name` = _name);
10      set _income = (select s.income - s.outcome from salary s where s.employeeid = id);
11  end $$
12
13  delimiter ;
14 • set @employee_name="朱骏";
15 • set @actual_income=0.0;
16
17 • call p4(@employee_name,@actual_income);
18 • select @actual_income as "实际收入";
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	实际收入			
▶	2078.00			

5. 创建存储过程 p5(in edu char(6), in x decimal(5,1))

将所有某种学历的员工收入提高%x，并调用该存储过程，将所有硕士学历的员工收入提高 10%。

```
1 • use yggl;
2 • drop procedure if exists p5;
3
4   delimiter $$
5
6 • create procedure p5(in edu char(6),in x decimal(5,1))
7   begin
8       update salary, employees set income = income * (1 + x)
9       where salary.employeeid = employees.employeeid and employees.education = edu;
10  end $$
11
12  delimiter ;
13 • set @edu="硕士";
14 • set @x=1;
15
16 • call p5(@edu,@x);
17
```

EmployeeID	Name	Education
000001	王浩	大专
010008	伍容华	本科
020010	王向蓉	硕士
020018	李丽	大专
102201	刘明	本科
102208	朱骏	硕士
108991	钟敏	硕士
111006	张石兵	本科
210678	林涛	大专
302566	李玉珉	本科
308759	叶凡	本科
504209	陈林琳	大专

1 • SELECT * FROM yggl.salary;

1 • SELECT * FROM yggl.salary;

Result Grid

Filter Rows:

	EmployeeID	InCome	OutCome
▶	000001	2100.8	123.09
	010008	1582.62	88.03
	020010	2860	198
	020018	2347.68	180
	102201	2569.88	185.65
	102208	1980	100
	108991	3259.98	281.52
	111006	1987.01	79.58
	210678	2240	121
	302566	2980.7	210.2
	308759	2531.98	199.08
	504209	2066.15	108
✱	NULL	NULL	NULL

Result Grid

Filter Rows:

	EmployeeID	InCome	OutCome
▶	000001	2100.8	123.09
	010008	1582.62	88.03
	020010	3146	198
	020018	2347.68	180
	102201	2569.88	185.65
	102208	2178	100
	108991	3585.98	281.52
	111006	1987.01	79.58
	210678	2240	121
	302566	2980.7	210.2
	308759	2531.98	199.08
	504209	2066.15	108
✱	NULL	NULL	NULL