 **(北京)**

# CHINA UNIVERSITY OF PETROLEUM

数据库原理上机报告

实T-SQL编程验三 **T-SQL编程**

院系名称： **信息学院**

专业名称： **计算机\_**

学生学号： 2019011777

学生姓名： 刘康来

完成日期 2021 年 11 月 17 日

**《数据库原理》上机报告**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **报告名称** | ***T-SQL编程*** | | | **粘贴**  **照片** |
| **姓 名** | 刘康来 | **班 级** | 计算机19-3 |
| **成 绩** |  | | |
| **1．上机实践目的与准备知识（简介，300字以内）**  **1.实验目的**  （1）掌握变量、基本数据类型、运算符、控制语句的基本语法和使用；  （2）掌握系统函数和用户自定义函数的使用；  （3）掌握存储过程的基本格式和使用。  **2. 实验准备**  （1）了解T-SQL支持的变量、基本数据类型、运算符、控制语句的基本语法和使用方法；  （2）了解系统函数的调用方法和用户自定义函数的使用步骤；  （3）了解存储过程的编写方法和使用过程。 | | | | |
| **2．主要实践内容与具体操作步骤（实践内容完成情况要有描述，如执行的SQL命令等，有运行结果截图，图大小以保证文字清晰为准）**  （1）自定义数据类型  CREATE TYPE Employee\_num  FROM char(6) NOT NULL  练习1：使用SQL语句创建表Employees3,结构与表Employees类似，只是EmployeeID使用上述自定义类型Employee\_num；  mysql 没有。。。  （2）使用变量  例如：DECLARE @student char(6)  例如：SELECT @var1 =  (  SELECT 姓名  FROM xsb  WHERE 学号= '191399'  )  SELECT @var1 AS 'NAME'  练习2： 定义一个变量，描述Salary表某一员工（员工号根据表中实际值选定）的实际收入（income-outcome），然后查询该变量。  set @wage = (  select Income - Outcome  from Salary  where EmployeeID = '1' );  select @wage as wage;    （3）流程控制  **分支结构练习3.1：** 判断姓名为“王琳”（或其他名字）的员工实际收入是否高于6000，如果是则显示其收入，否则显示“收入不高于6000”。  create procedure *judge*() begin  if @wagea > 6000  then select 'a', @wagea;  else  select 'a', '收入不高于6000';  end if; end;  call *judge*();    循环结构练习3.2：  （a）使用循环输出一个用’\*’组成的5行三角形。  -- mysql haven't print!!!  SET @NUMBER = 6; SELECT REPEAT('\* ', @NUMBER := @NUMBER - 1) as 'triangle'  FROM information\_schema.tables LIMIT 5;    （b）将员工收入低于4000的员工的收入使用循环修改到6000，每次只加50，并判断循环了多少次。  create table Income\_loop\_num(  ID char(6),  Loop\_num int );  create procedure *addIncome*(add\_num int) begin  declare number int;  declare i int;  declare count int;  declare tem\_income int;  declare tem\_id char(6);  select count(\*) from Salary into number;  set i = 0;  set count = 0;  while i < number do  select Income, EmployeeID from Salary1 LIMIT i, 1 into tem\_income, tem\_id;  if tem\_income < 4000 then  while tem\_income < 6000 do  set tem\_income = tem\_income + add\_num;  set count = count +1;  end while;  update Salary1 set Income = tem\_income where EmployeeID = tem\_id;  insert into Income\_loop\_num value (tem\_id, count);  end if;  set count = 0;  set i = i + 1;  end while; end;  select \* from Income\_loop\_num;  create table Salary1 like Salary; insert into Salary1 select \* from Salary;  call *addIncome*(50);    （4）自定义函数    练习4：a）编写一个函数用来对员工的工资进行分级，3000元以下为1级，3000-4000元为2级，…以此类推，每级相差1000元。调用该函数显示每个员工的工资及其级别。  create table grading like Salary; insert into grading select \* from Salary;  alter table grading add grade int;  create procedure *grading*() begin  declare number int;  declare i int;  declare tem\_income int;  declare tem\_id char(6);  declare tem\_grade int;  set i = 0;  select count(\*) from grading into number;  while(i < number) do  select Income, EmployeeID into tem\_income, tem\_id from grading limit i, 1;  if(tem\_income < 3000) then  set tem\_grade = 1;  else  set tem\_grade = (tem\_income - 1000) / 1000;  end if;  update grading set grade = tem\_grade where EmployeeID = tem\_id;  set i = i+1;  set tem\_grade = 0;  end while; end;  drop procedure *grading*; call *grading*();  select \* from grading;    b）编写一个函数，该函数的作用是统计公司各部门的员工人数和员工的最高收入、最低收入和平均收入（选做）  select DepartmentName, count(\*), max(Income), min(Income), avg(Income) from Employees natural join Departments natural join Salary group by DepartmentName;    5）存储过程  练习5.1：（a）创建一个存储过程，比较两个员工的实际收入，若前者比后者高就输出0，否则输出1  create procedure *compare\_income*(name1 char(6), name2 char(6)) begin  declare income1, income2 int;  select Income into income1 from Salary where EmployeeID = name1;  select Income into income2 from Salary where EmployeeID = name2;  if income1 >= income2 then  select 0;  else  select 1;  end if; end;  drop procedure *compare\_income*; call *compare\_income*('1', '0');  select \* from Salary;    （b）创建一个存储过程，将每个人的收入提高500。（如果是根据每个人的学历将收入提高，如大专及以下提高400，本科提高500，硕士提高650，博士提高800，又如何实现？选做）  create procedure *add\_income\_by\_education*() begin  declare tem\_education char(4);  declare number, i int;  declare tem\_income int;  declare tem\_id char(6);  set i = 0;  select count(\*) from Employees natural join Salary2 into number;  while(i < number) do  select Income, EmployeeID, Education into tem\_income, tem\_id, tem\_education  from Employees natural join Salary2 limit i, 1;  case tem\_education  when '小学' then set tem\_income = tem\_income + 400;  when '大学' then set tem\_income = tem\_income + 800;  when '高中' then set tem\_income = tem\_income + 1200;  end case;  update Salary2 set Income = tem\_income where EmployeeID = tem\_id;  set i = i+1;  end while; end;  select \* from Salary2;  select \* from Salary;  call *add\_income\_by\_education*();      练习5.2：（a）创建一个存储过程，要求一个员工的工作年份大于10年时将其转到经理办公室工作    create procedure *cursor\_move*() begin  declare done int default false;  declare tem\_id char(6);  declare tem\_workyer char(3);  declare need\_id char(3);  declare cursor\_i cursor for select EmployeeID, Workyer from Employees;  declare continue handler for not found set done = true;   select DepartmentID into need\_id from Departments where DepartmentName = '经理办公室';   open cursor\_i;  read\_loop: LOOP  fetch cursor\_i into tem\_id, tem\_workyer;  if done then  leave read\_loop;  END IF;  if tem\_workyer > 10 then  update Employees1 set DepartmentID = need\_id where EmployeeID = tem\_id;  end if;  end loop;  close cursor\_i; end;  create table Employees1 like Employees; insert into Employees1 select \* from Employees;  select \* from Employees1; call *cursor\_move*();    （b）创建一个存储过程，使用游标计算本科及以上学历的员工在总员工中所占比例。（选做）  create procedure *compute*() begin  declare done int default false;  declare tem\_education char(4);  declare total int;  declare number\_education int;  declare cursor\_education cursor for select Education from Employees;  declare continue handler for not found set done = true;   set number\_education = 0;  select count(\*) into total from Employees;  open cursor\_education;  label: loop  fetch cursor\_education into tem\_education;  if done then  leave label;  end if;  if tem\_education != '小学' then  set number\_education = number\_education + 1;  end if;  end loop;  close cursor\_education;  select number\_education / total; end;  call *compute*();    （c）创建存储过程，使用游标确定一个员工的实际收入是否排在前三名，结果为1表示是，结果为0表示否。（选做）  create procedure *judge\_rank*(id char(6)) begin  declare tem\_id char(6);  declare count int;  declare flag int;  declare cursor1 cursor for select EmployeeID from Salary order by Income desc;  set count = 0;  set flag = 0;   open cursor1;  label: loop  fetch cursor1 into tem\_id;  if count = 3 then  leave label;  end if;  if tem\_id = id then  set flag = 1;  leave label;  end if;  set count = count + 1;  end loop;  if flag then  select id, '1' as if\_first\_3;  else select id, '0' as if\_first\_3;  end if;  close cursor1; end;  drop procedure *judge\_rank*;  select \* from Salary order by Income desc; call *judge\_rank*('7'); call *judge\_rank*('6'); call *judge\_rank*('4'); | | | | |
| **3．总结与问题分析（100字以上）**  **语法难找，还有mysql 有一些语法不支持，像自定义数据类型，也没有print函数，只能找些代替品了，不是很全。**  **函数的定义必须要有返回值，一律用procedure代替，影响不大。而且函数里面不能select输出，报什么 value set 的错误。**  **在浏览一个表时，除了用 cursor 之外，limit 也挺好使的，加个循环。**  **Limit I 1， 从 I 行起，查一行 ，那在插入语句时怎样插入指定行的顺序呢，实现排序之类的。**  **在网上查的过程中，有一个label，好像挺重要的样子，还有什么 loop 之类的。** | | | | |
|  | | | | |