《数据库技术实验》报告

|  |  |  |
| --- | --- | --- |
| **第 八 次实验** | **日期：2022-11-26** | **得分：** |
| **学号：** | **姓名：*Steven*** | **专业：智能科学与技术** |

一、实验目的

1. 掌握实现存储过程的PL/SQL语句

2. 掌握实现函数的PL/SQL语句

3. 了解使用游标的PL/SQL语句

二、实验内容

（一）实验题

1. 在Student\_Course数据库上，创建存储过程实现：输入课程名，计算并输出该课程的平均分，并执行验证该存储过程。

2. 在Student\_Course数据库上，创建函数实现：输入课程名，计算并返回该课程的平均分，并执行验证该函数。

3. 在Student\_Course数据库上，创建存储过程实现：计算每门课程的平均分，将平均分低于70分的课程号、课程名、平均分存入表fail\_course中，并执行验证该存储过程。

4. 在TPC\_H数据库上，设计创建一个存储过程，并执行验证该存储过程。

5. 在TPC\_H数据库上，设计创建一个函数，并执行验证该函数。

（二）思考题

1. 存储过程与函数的区别是什么？

2. 为什么引入游标？如何使用游标？

三、实验过程及实验结果

1. 在Student\_Course数据库上，创建存储过程实现：输入课程名，计算并输出该课程的平均分，并执行验证该存储过程。

(1) 创建存储过程并调用

|  |
| --- |
| *-- 创建存储过程：输入课程名，计算并输出该课程的平均分* CREATE PROCEDURE *`course\_ave\_procedure`*(cname char(10)) BEGIN  declare grade\_avg float; *-- 课程平均分* declare cur\_grade\_avg cursor for  select cname, *avg*(sc.grade)  from sc  where sc.cno = (select cno from course where course.cname = cname); *-- 游标* create table if not exists tbl\_avg\_grade  (  cname char(10),  ave float  );*-- 创建暂存结果的表* open cur\_grade\_avg;  fetch cur\_grade\_avg into cname, grade\_avg;  close cur\_grade\_avg;  if *exists*(select *\** from tbl\_avg\_grade where tbl\_avg\_grade.cname = cname) then  update tbl\_avg\_grade  set ave = grade\_avg  where cname = cname;  else  insert into tbl\_avg\_grade()  values (cname, grade\_avg);  end if; *-- 将结果插入到表中* END; |
| CALL *course\_ave\_procedure*('C语言'); |

(2) 结果演示

图形用户界面, 文本, 应用程序

描述已自动生成表格

描述已自动生成

（成功创建course\_ave\_procedure，并且获取到了结果）

2. 在Student\_Course数据库上，创建函数实现：输入课程名，计算并返回该课程的平均分，并执行验证该函数。

(1) 创建函数并调用

|  |
| --- |
| *-- 创建函数实现：输入课程名，计算并返回该课程的平均分* set global log\_bin\_trust\_function\_creators = TRUE; CREATE FUNCTION *`course\_ave\_function`*(cname char(10)) RETURNS float BEGIN  declare grade\_avg float;  select *avg*(sc.grade)  into grade\_avg  from sc  where sc.cno = (select cno from course where course.cname = cname);  return grade\_avg; END; |
| SELECT *course\_ave\_function*('C语言'); |

(2) 结果演示

图形用户界面, 文本, 应用程序

描述已自动生成表格

低可信度描述已自动生成

（成功创建course\_ave\_function，并且获取到了结果）

3. 在Student\_Course数据库上，创建存储过程实现：计算每门课程的平均分，将平均分低于70分的课程号、课程名、平均分存入表fail\_course中，并执行验证该存储过程。

(1) 创建存储过程并调用

|  |
| --- |
| *-- 创建存储过程：计算每门课程的平均分，将平均分低于70分的课程号、课程名、平均分存入表中* CREATE PROCEDURE *`course\_ave\_procedure2`*() BEGIN  declare cno char(11);  declare cname char(10);  declare grade\_avg float;  declare flag int;  declare cur\_course\_avg cursor for  select course.cno, course.cname, *avg*(sc.grade)  from course,  sc  where course.cno = sc.cno  group by course.cno; *-- 游标* DECLARE CONTINUE HANDLER FOR NOT  FOUND SET flag = 1;  set flag = 0;  drop table if exists fail\_course;  create table fail\_course  (  cno char(11),  cname char(10),  avegrade float  );*-- 暂存结果的表* open cur\_course\_avg;  fetch cur\_course\_avg into cno, cname, grade\_avg;  while (not flag)  do  if (grade\_avg < 70) then  insert into fail\_course values (cno, cname, grade\_avg);  end if;  fetch cur\_course\_avg into cno, cname, grade\_avg;  end while;  close cur\_course\_avg; END; |
| CALL *course\_ave\_procedure2*(); |

(2) 结果演示

图形用户界面, 文本, 应用程序

描述已自动生成应用程序, 表格

描述已自动生成

（此时没有结果，因为所有科目的平均分都在70以上，为了验证该存储过程可用，将存储过程中的判断条件改为grade\_avg < 100，再次测试一下功能）

图形用户界面, 应用程序, 表格, Excel

描述已自动生成

（此时便有了所有课程的结果）

4. 在TPC\_H数据库上，创建存储过程：统计各个region中的customer人数

(1) 创建存储过程并调用

|  |
| --- |
| *-- 创建存储过程：统计各个region中的customer人数* CREATE PROCEDURE *`customer\_count\_by\_region\_procedure`*() BEGIN  declare region\_name text;  declare customer\_count int;  declare flag int;  declare cur\_customer\_count cursor for  select region.name, *count*(customer.custkey)  from region,  nation,  customer  where region.regionkey = nation.regionkey  and nation.nationkey = customer.nationkey  group by region.name;  *-- 游标* DECLARE CONTINUE HANDLER FOR NOT  FOUND SET flag = 1;  set flag = 0;  drop table if exists tbl\_customer\_count\_by\_region;  create table tbl\_customer\_count\_by\_region  (  region\_name text,  customer\_count int  );*-- 创建暂存结果的表* open cur\_customer\_count;  fetch cur\_customer\_count into region\_name, customer\_count;  while (not flag)  do  insert into tbl\_customer\_count\_by\_region values (region\_name, customer\_count);  fetch cur\_customer\_count into region\_name, customer\_count;  end while;  close cur\_customer\_count; END; |
| CALL *customer\_count\_by\_region\_procedure*(); |

(2) 结果演示

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成图形用户界面, 应用程序, 表格, Excel

描述已自动生成

（成功创建customer\_count\_by\_region\_procedure，并且获取到了结果）

5. 在TPC\_H数据库上，创建函数：返回某个region中的supplier个数

(1) 创建函数并调用

|  |
| --- |
| *-- 创建函数：返回某个region中的supplier个数* CREATE FUNCTION *`supplier\_count\_by\_region\_function`*(region\_name text) RETURNS int BEGIN  declare supplier\_count int;  select *count*(supplier.suppkey)  into supplier\_count  from supplier,  nation,  region  where supplier.nationkey = nation.nationkey  and nation.regionkey = region.regionkey  and region.name = region\_name;   return supplier\_count; END; |
| SELECT *supplier\_count\_by\_region\_function*('亚洲'); |

(2) 结果演示

图形用户界面, 文本, 应用程序

描述已自动生成图形用户界面, 文本, 应用程序

描述已自动生成

（成功创建supplier\_count\_by\_region\_function，并且获取到了结果）

四、思考题

1. 存储过程与函数的区别是什么？

① 函数的限制更多一些，比如不能用临时表、以及一些函数不可用；

② 函数必须要求有且仅有一个返回值，而存储过程不要求，或者说它将任意多的返回结果存储在表中。

③ 函数的参数只有一种，存储过程的参数类型有三种：IN参数、OUT参数、INOUT参数。

2. 为什么引入游标？如何使用游标？

由于MySQL在进行查询操作时结果有可能存在多行，此时便需要游标暂存这张表，以便于之后逐行处理。