西安交通大学实验报告

课程名称：数据库基础与应用
学院：物理学院
班级：物试 2201
学号：2223711803

实验名称：存储过程
实验日期：2023.4.15
姓名：汪洋

1 实验目的

1. 理解存储过程和函数的概念。
2. 掌握一种 DBMS 中存储过程的使用方法。
3. 能将存储过程应用到数据的管理中。

2 实验任务

练习存储过程和函数的使用。

3 实验环境

1. 华为云服务器
2. openGauss 数据库
3. Windows 操作系统
4. Putty

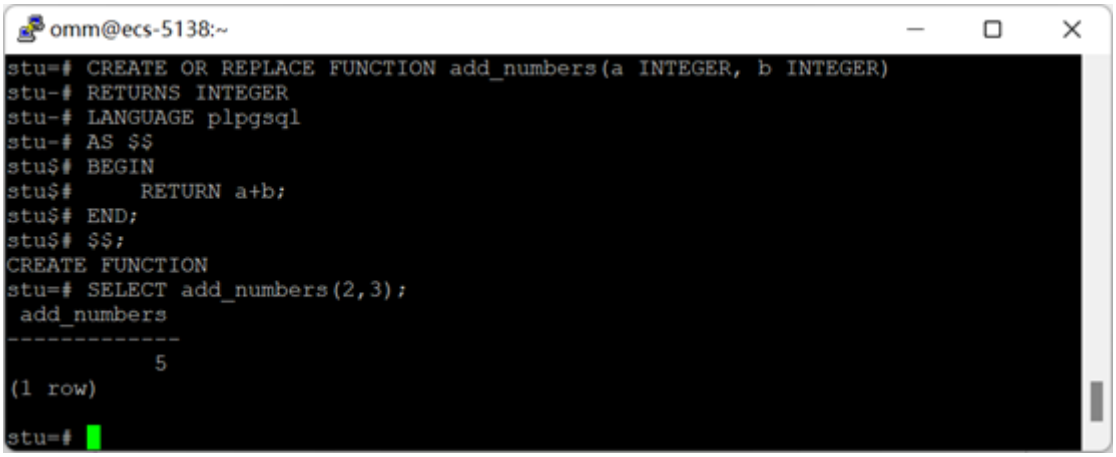
4 实验步骤与结果

1. 创建实验用数据表并插入 5 条实验数据，包含自己的名字。

```
create table student(number varchar(10), name varchar(20), birthday date);
insert into student values('001', '汪洋', '2004-9-5');
insert into student values('002', '张三', '2004-1-1');
insert into student values('003', '李四', '2004-2-2');
insert into student values('004', '王五', '2004-3-3');
insert into student values('005', '赵六', '2004-4-4');
```

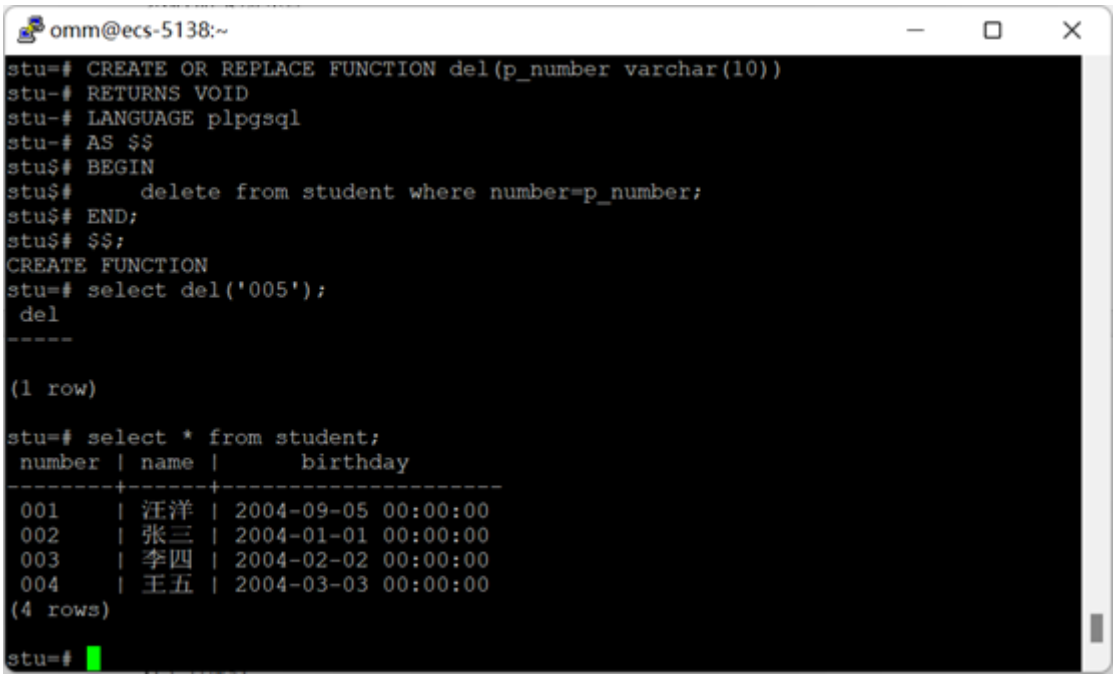
2. 编写函数，计算两个数的和，验证该函数。

```
CREATE OR REPLACE FUNCTION add_numbers(a INTEGER, b INTEGER)
RETURNS INTEGER
LANGUAGE plpgsql
AS $$
BEGIN
    RETURN a+b;
END;
$$;
SELECT add_numbers(2,3);
```



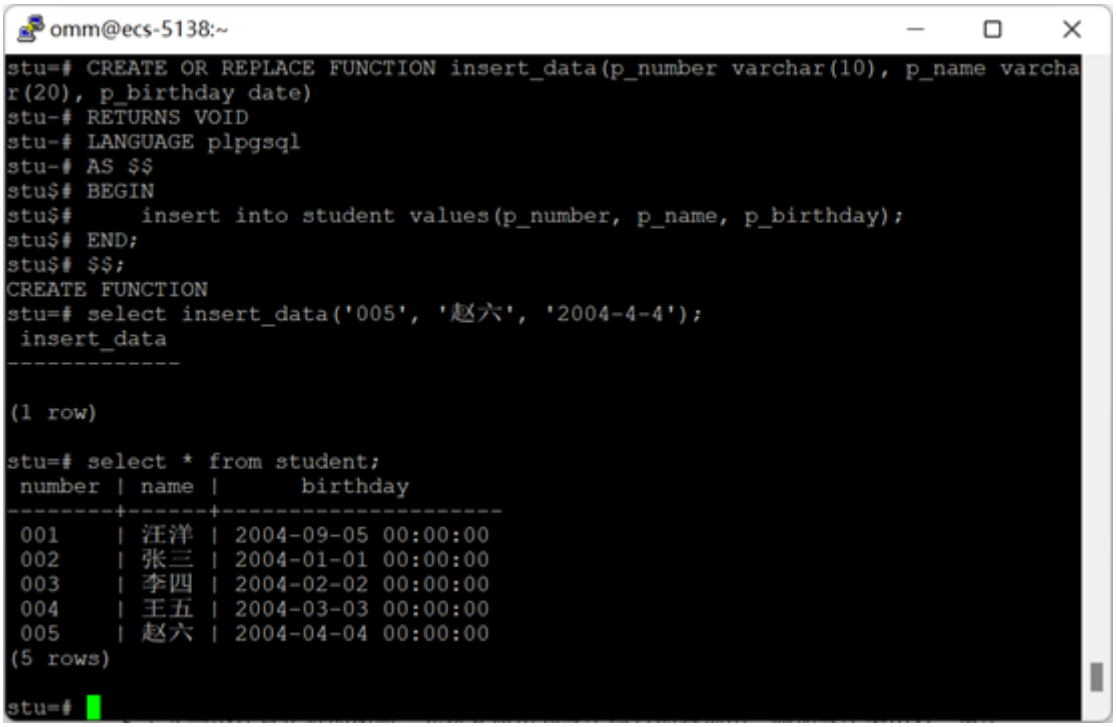
3. 编写函数，根据学号删除学生信息。输入参数为学号。验证该函数。

```
CREATE OR REPLACE FUNCTION del(p_number varchar(10))
RETURNS VOID
LANGUAGE plpgsql
AS $$
BEGIN
    delete from student where number=p_number;
END;
$$;
select del('005');
select * from student;
```



4. 编写函数，插入学生信息。输入参数为学号、姓名和出生日期。验证该函数。

```
CREATE OR REPLACE FUNCTION insert_data(p_number varchar(10), p_name varchar(20), p_birthday
    date)
RETURNS VOID
LANGUAGE plpgsql
AS $$
BEGIN
    insert into student values(p_number, p_name, p_birthday);
END;
$$;
select insert_data('005', '赵六', '2004-4-4');
select * from student;
```

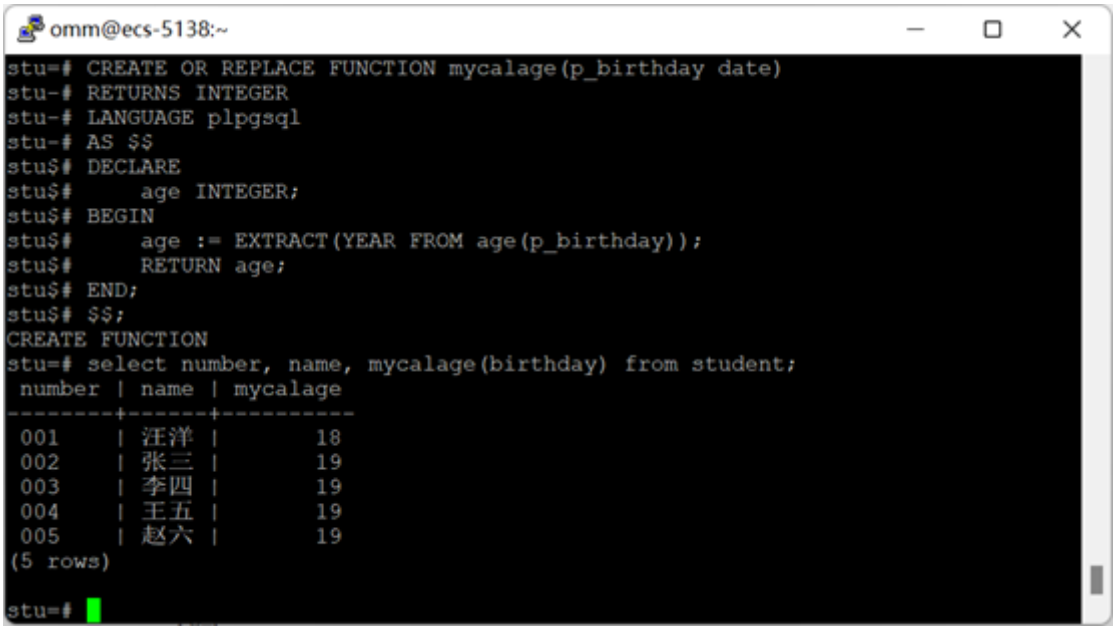


5. 定义一个计算年龄的函数，并将其应用到学生信息的查询中，例如学生表中有“出生日期”字段birth，日期型数据（含年月日），查询如下：

```
select numb, name, mycalage(birth)  from student;
```

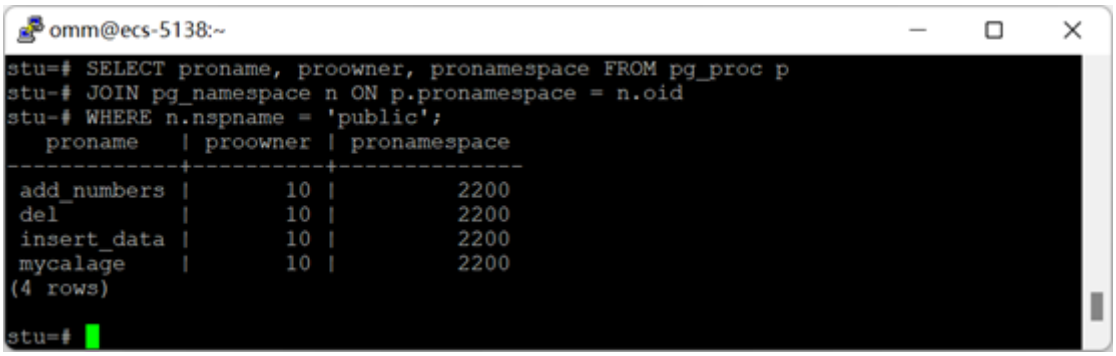
显示学生的学号、姓名和年龄。验证该函数。

```
CREATE OR REPLACE FUNCTION mycalage(p_birthday date)
RETURNS INTEGER
LANGUAGE plpgsql
AS $$
DECLARE
    age INTEGER;
BEGIN
    age := EXTRACT(YEAR FROM age(p_birthday));
    RETURN age;
END;
$$;
select number, name, mycalage(birthday) from student;
```



6. 查看所定义的函数。

```
SELECT proname, proowner, pronamespace FROM pg_proc p
JOIN pg_namespace n ON p.pronamespace = n.oid
WHERE n.nspname = 'public';
```



7. 自己做的其他函数练习。

(1) 计算 n!

```
#使用FOR循环
CREATE OR REPLACE FUNCTION factorial(n INTEGER)
RETURNS INTEGER
LANGUAGE plpgsql
AS $$
DECLARE
    result INTEGER := 1;
    i INTEGER;
BEGIN
    IF n < 0 THEN
        RAISE EXCEPTION 'Input must be a non-negative integer.';
    END IF;
    FOR i IN 1..n
    LOOP
        result := result * i;
    END LOOP;
    RETURN result;
END;
$$;

#使用WHILE循环
CREATE OR REPLACE FUNCTION factorial(n INTEGER)
RETURNS INTEGER
LANGUAGE plpgsql
AS $$
DECLARE
    result INTEGER := 1;
    i INTEGER := 1;
BEGIN
    IF n < 0 THEN
        RAISE EXCEPTION 'Input must be a non-negative integer.';
    END IF;
    WHILE i <= n
    LOOP
        result := result * i;
        i := i + 1;
    END LOOP;
    RETURN result;
END;
$$;
```

(2) 计算斐波那契数列第 n 项

```
#循环解法
CREATE OR REPLACE FUNCTION fibonacci(n INTEGER)
RETURNS INTEGER
```

```
LANGUAGE plpgsql
AS $$
DECLARE
    a INTEGER := 0;
    b INTEGER := 1;
    i INTEGER;
    temp INTEGER;
BEGIN
    IF n <= 1 THEN
        RETURN n;
    END IF;
    FOR i IN 2..n LOOP
        temp := a + b;
        a := b;
        b := temp;
    END LOOP;
    RETURN b;
END;
$$;

#递归解法
CREATE OR REPLACE FUNCTION fibonacci(n INTEGER)
RETURNS INTEGER
LANGUAGE plpgsql
AS $$
BEGIN
    IF n <= 1 THEN
        RETURN n;
    ELSE
        RETURN fibonacci(n - 1) + fibonacci(n - 2);
    END IF;
END;
$$;
```

(3) 计算 10 以内的奇数和

```
#使用FOR循环
CREATE OR REPLACE FUNCTION sum_odds(n TNTEGER)
RETURNS INTEGER
LANGUAGE plpgsql
AS $$
DECLARE
    sum_odds INTEGER :=0 ;
    i INTEGER;
BEGIN
    FOR i IN 1..10
    LOOP
        IF i % 2 != 0 THEN
            sum_odds := sum_odds + i;
        END IF;
    END LOOP;
    RETURN sum_odds;
END;
$$;

#使用WHILE循环
CREATE OR REPLACE FUNCTION sum_odds(n TNTEGER)
RETURNS INTEGER
LANGUAGE plpgsql
AS $$
DECLARE
    sum_odds INTEGER :=0 ;
    i INTEGER := 1;
BEGIN
```

```
    WHILE 1 LOOP  #此处1被视为真，创建了一个无限循环
        IF i >= 10 THEN
            EXIT;
        END IF;
        IF i % 2 != 0 THEN
            sum_odds := sum_odds + i;
            i = i + 1;
            CONTINUE;
        END IF;
        i = i + 1;
    END LOOP;
    RETURN sum_odds;
END;
$$;
```

8. 删除自定义的函数和数据表。

```
drop function add_numbers;
drop function del;
drop function insert_data;
drop function mycalage;
drop table student;
```

5 实验总结

1. 总体来说非常简单。
2. 当函数参数与列名重复的时候很容易报错，尽量要使用不重复的名称。
3. 在 FOR 循环中，i := i + 1 不需显式地写出；在 WHILE 循环中，i := i + 1 需要显式地写出。