

# 实验 Psycopg2 与 PL/pgSQL

## 1. 实验目的

- (1) 掌握使用 Python 连接数据库的基本步骤，并对数据库进行各种数据操作。
- (2) 掌握 PL/pgSQL 函数的结构，并掌握 PL/pgSQL 函数的编写和调用。
- (3) 弘扬抗美援朝不朽精神，勠力中华民族伟大复兴。

## 2. 实验环境

- (1) 数据库管理系统：PostgreSQL。
- (2) 图形化管理工具：pgAdmin。
- (3) 应用程序开发环境：anaconda/pycharm/psycopg2。

## 3. 实验内容

- (1) 理解 psycopg2 连接 PostgreSQL 的基本步骤，使用 Python 语言编程实现对数据库的访问，并对数据库进行各种操作。
- (2) 编写 PL/pgSQL 函数，输入参数，定义变量，输出相应的结果。
- (3) Psycopg2 和 PL/pgSQL 函数综合练习

## 4. 实验数据

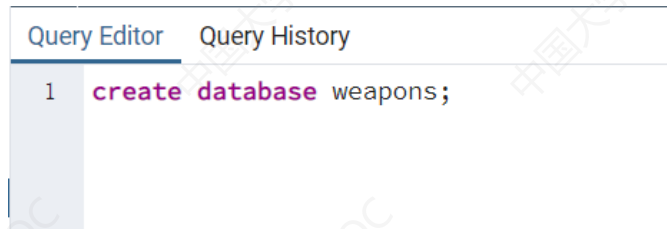
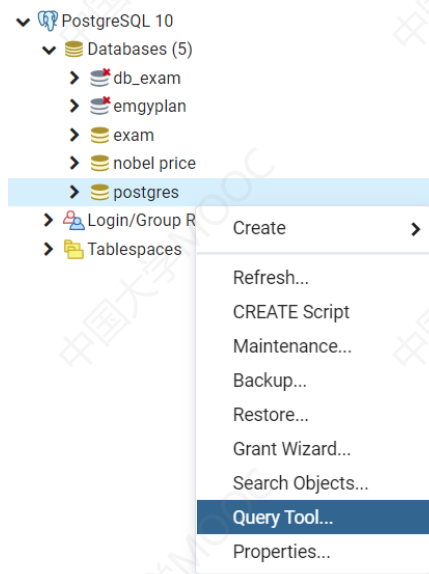
70 年来，伟大的抗美援朝精神一直是中国人民宝贵的精神财富，激励着中华儿女不畏牺牲，砥砺奋进，创造了一个又一个人间奇迹，谱写了中华民族伟大复兴道路上振奋人心的乐章。当前，世界正处于百年未有之大变局，我们党领导的伟大斗争、伟大工程、伟大事业、伟大梦想正在如火如荼进行，改革发展稳定任务艰巨繁重，我们面临着难得的历史机遇，也面临着一系列重大风险考验。因此，发扬伟大的抗美援朝精神，对我们今天坚定信念、勠力同心实现中华民族伟大复兴，具有十分重要的时代价值。抗美援朝是人类历史上以弱胜强的最经典战争，当时的中国人民志愿军依靠为了民族存亡而殊死搏斗的精神取得伟大胜利。我国与美国的军资力量悬殊，却凭着一股对祖国热爱、对友谊邻邦关爱，凭借着中国人民不屈不挠的血性，最终取得胜利。本次实验使用抗美援朝战争武器装备数据库 weapons，主要涉及 weapon 表，weapon 包括 weapon\_id，武器类型 (weapon\_type)，中国人民志愿军每个军配备数量 (num\_cn) 以及美国军队每个军配备数量 (num\_usa)。

## 5. 实验作业

- (1) 使用 psycopg2 编程，连接 weapons 数据库，新建数据表 weapon，weapon 结构如下：

属性	类型	是否为主键	是否外键
weapon_id	int	是	否
weapon_type	varchar	否	否
num_cn	int	否	否
num_usa	int	否	否

说明：在 pgAdmin 中新建数据库 weapons



可编辑格式的源代码：

```
1. # -*- coding: utf-8 -*-
2. """
3. Created on Mon Nov 2 18:14:39 2020
4.
5. @author: bayonet8163655
6. """
7.
8. import psycopg2
9.
10. #连接数据库
11. def connectPostgreSQL():
12.     conn = psycopg2.connect(database="weapons", user="postgres",
13.                             password="123456", host="127.0.0.1",
14.                             port="5432")
15.     print ("connect successful!")
16.     return conn
17. #断开数据库连接
```

```

18. def disconnectPostgreSQL(conn):
19.     conn.commit()#提交所有未提交操作到数据库
20.     conn.close()#断开数据库连接
21.
22. #创建数据表
23. def createtable():
24.     create_sql=("create table weapon("
25.                 "weapon_id int primary key,"
26.                 "weapon_type varchar(10),"
27.                 "num_cn int,"
28.                 "num_usa int)")
29.     conn=connectPostgreSQL()
30.     cur=conn.cursor()
31.     cur.execute(create_sql)
32.     print("数据表创建成功")
33.     disconnectPostgreSQL(conn)
34.
35.
36. if __name__=='__main__':
37.
38.     #创建数据表
39.     createtable()

```

**说明：**connectPostgreSQL()与 disconnectPostgreSQL() 分别是封装数据库连接和断连的函数，createtable()函数连接数据库 weapons 并创建数据表 weapon，然后与数据库断开连接。

**源代码截图：**

```
pg_test.py*
1  # -*- coding: utf-8 -*-
2  """
3  Created on Mon Nov  2 18:14:39 2020
4
5  @author: bayonet8163655
6  """
7
8  import psycopg2
9
10 # 连接数据库
11 def connectPostgreSQL():
12     conn = psycopg2.connect(database="weapons", user="postgres",
13                             password="123456", host="127.0.0.1", port="5432")
14     print ("connect successful!")
15     return conn
16
17 # 断开数据库连接
18 def disconnectPostgreSQL(conn):
19     conn.commit()#提交所有未提交操作到数据库
20     conn.close()#断开数据库连接
21
22 # 创建数据表
23 def createtable():
24     create_sql=("create table weapon("
25                "weapon_id int primary key,"
26                "weapon_type varchar(10),"
27                "num_cn int,"
28                "num_usa int)")
29     conn=connectPostgreSQL()
30     cur=conn.cursor()
31     cur.execute(create_sql)
32     print("数据表创建成功")
33     disconnectPostgreSQL(conn)
34
35
36 if __name__ == '__main__':
37
38     # 创建数据表
39     createtable()
```

执行结果:

```
Console 1/A
In [2]: runfile('D:/code/python_project/pg_test.py', wdir='D:/code/python_project')
connect successful!
数据表创建成功

Tables (1)
  weapon
    Columns (4)
      weapon_id
      weapon_type
      num_cn
      num_usa
    Constraints
    Indexes
    RLS Policies
    Rules
    Triggers
    Trigger Functions
    Types
    Views
```

Query Editor

Query History

1

select \* from weapon;

Data Output

Explain

Messages

Notifications

weapon_id	weapon_type	num_cn	num_usa
[PK] integer	character varying (10)	integer	integer

Tables (1)

weapon

Columns (4)

weapon\_id

weapon\_type

num\_cn

num\_usa

Constraints

Indexes

RLS Policies

Rules

Triggers

Trigger Functions

Types

Views

Group Roles

weapon

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Inherited from table(s)

Select to inherit from...

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
weapon_id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
weapon_type	character varying	10		<input type="checkbox"/>	<input type="checkbox"/>
num_cn	integer			<input type="checkbox"/>	<input type="checkbox"/>
num_usa	integer			<input type="checkbox"/>	<input type="checkbox"/>

(2) 使用 psycopg2 编程，向 weapon 中插入如下数据：

weapon_id	weapon_type	num_cn	num_usa
1	火炮	198	1428
2	汽车	120	7000
3	坦克	0	430

程序执行前状态：

Query Editor

Query History

1

select \* from weapon;

Data Output

Explain

Messages

Notifications

weapon_id	weapon_type	num_cn	num_usa
[PK] integer	character varying (10)	integer	integer

可编辑格式的源代码:

```
1. # -*- coding: utf-8 -*-
2. """
3. Created on Mon Nov  2 18:14:39 2020
4.
5. @author: bayonet8163655
6. """
7.
8. import psycopg2
9.
10. #连接数据库
11. def connectPostgreSQL():
12.     conn = psycopg2.connect(database="weapons", user="postgres",
13.                             password="123456", host="127.0.0.1",
14.                             port="5432")
15.     print ("connect successful!")
16.     return conn
17. #断开数据库连接
18. def disconnectPostgreSQL(conn):
19.     conn.commit()#提交所有未提交操作到数据库
20.     conn.close()#断开数据库连接
21.
22. #插入数据
23. def insertData():
24.     insert_sql=("insert into weapon values"
25.                "(1,'火炮',198,1428),"
26.                "(2,'汽车',120,7000),"
27.                "(3,'坦克',0,430)")
28.     conn=connectPostgreSQL()
29.     cur=conn.cursor()
30.     cur.execute(insert_sql)
31.     print("数据表插入成功")
32.     disconnectPostgreSQL(conn)
33.
34. if __name__ == '__main__':
35.
36.     #插入数据
37.     insertData()
```

说明: insertData()函数负责插入数据, 其中 insert\_sql 中为插入数据的 sql 语言

源代码截图:

```
pg_test.py
1  # -*- coding: utf-8 -*-
2  """
3  Created on Mon Nov  2 18:14:39 2020
4
5  @author: bayonet8163655
6  """
7
8  import psycopg2
9
10 #连接数据库
11 def connectPostgreSQL():
12     conn = psycopg2.connect(database="weapons", user="postgres",
13                             password="123456", host="127.0.0.1", port="5432")
14     print ("connect successful!")
15     return conn
16
17 #断开数据库连接
18 def disconnectPostgreSQL(conn):
19     conn.commit()#提交所有未提交操作到数据库
20     conn.close()#断开数据库连接
21
22 #插入数据
23 def insertData():
24     insert_sql=("insert into weapon values"
25                "(1,'火炮',198,1428),"
26                "(2,'汽车',120,7000),"
27                "(3,'坦克',0,430)")
28     conn=connectPostgreSQL()
29     cur=conn.cursor()
30     cur.execute(insert_sql)
31     print("数据表插入成功")
32     disconnectPostgreSQL(conn)
33
34 if __name__ == '__main__':
35     #插入数据
36     insertData()
37
```

执行效果:

```
Console 1/A
In [3]: runfile('D:/code/python_project/pg_test.py', wdir='D:/code/python_project')
connect successful!
数据表插入成功

In [4]:
```

weapons/postgres@PostgreSQL 10

Query Editor

Query History

1 select \* from weapon;

Data Output

Explain

Messages

Notifications

	weapon_id [PK] integer	weapon_type character varying (10)	num_cn integer	num_usa integer
1	1	火炮	198	1428
2	2	汽车	120	7000
3	3	坦克	0	430

(3) 在 PG 中，创建一个函数 add\_num\_cn (weapon\_id, num\_cn)，实现对中国人民志愿军每个军配备给定武器种类 id 的数量增加定值。例如：add\_num\_cn (2,1) 是给中国人民志愿军每个军配备汽车数量(num\_cn)添加 1。 其中，该函数的第一个形参对应 weapon\_id 的值，第二个形参对应 num\_cn 的值，形参名可自行定义或使用 \$1, \$2 代表。

可编辑格式的源代码：

```

1. create function add_num_cn(add_weapon_id int, add_num_cn int)
2. returns text
3. as
4. $$
5. Begin
6. update weapon set num_cn=num_cn+add_num_cn where weapon_id=add_weapon_id;
7. return '操作成功'
8. End;
9. $$
10. language plpgsql;

```

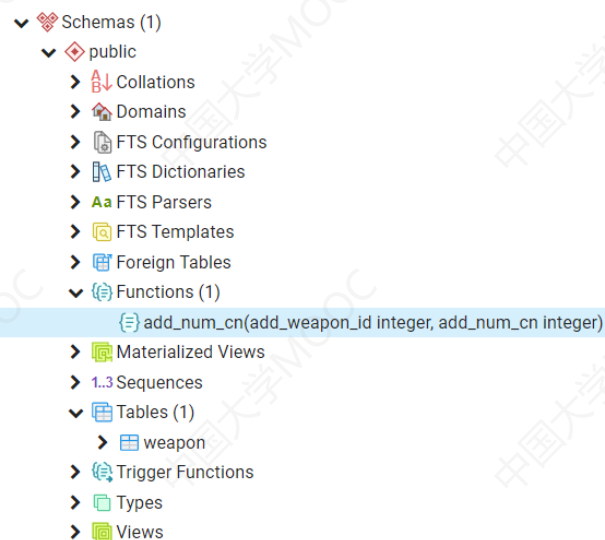
**说明：**在 pgAdmin 中创建 PL/pgSQL 函数，add\_num\_cn，若成功修改则返回操作成功字样，第一个参数表示修改的项目，第二个参数表示增加的数量。

源代码截图：



```
weapons/postgres@PostgreSQL 10
Query Editor  Query History
1  create function add_num_cn(add_weapon_id int, add_num_cn int)
2  returns text
3  as
4  $$
5  Begin
6  update weapon set num_cn=num_cn+add_num_cn where weapon_id=add_weapon_id;
7  return '操作成功'
8  End;
9  $$
10 language plpgsql;
```

执行结果:



(4) 在 psycopg2 中调用(3)中的函数, 输出给中国人民志愿军每个军配备汽车数量添加 1 后的结果 (psycopg2 输出最终得到数字即可)

程序执行前状态:

weapons/postgres@PostgreSQL 10				
Query Editor  Query History				
1 select * from weapon;				
Data Output	Explain	Messages	Notifications	
weapon_id [PK] integer	weapon_type character varying (10)	num_cn integer	num_usa integer	
1	1 火炮	198	1428	
2	2 汽车	120	7000	
3	3 坦克	0	430	

可编辑格式的源代码:

```
1. # -*- coding: utf-8 -*-
2. """
3. Created on Mon Nov  2 18:14:39 2020
4.
5. @author: bayonet8163655
6. """
7.
8. import psycopg2
9.
10. #连接数据库
11. def connectPostgreSQL():
12.     conn = psycopg2.connect(database="weapons", user="postgres",
13.                             password="123456", host="127.0.0.1",
14.                             port="5432")
15.     print ("connect successful!")
16.     return conn
17.
18. #断开数据库连接
19. def disconnectPostgreSQL(conn):
20.     conn.commit()#提交所有未提交操作到数据库
21.     conn.close()#断开数据库连接
22.
23. #添加
24. def add_plpgsql():
25.     select_sql="add_num_cn(2,1)"
26.     conn=connectPostgreSQL()
27.     cur=conn.cursor()
28.     cur.execute(select_sql)
29.     disconnectPostgreSQL(conn)
30.
31. #查询信息
32. def selectData():
33.     select_sql="select * from weapon"
34.     conn=connectPostgreSQL()
35.     cur=conn.cursor()
36.     cur.execute(select_sql)
37.     rows=cur.fetchall()
38.     for row in rows:
39.         print("weapon_id=",row[0],"\tweapon_type=",row[1],
40.               "\tnum_cn=",row[2],"\tnum_usa=",row[3],"\n")
41.     disconnectPostgreSQL(conn)
```

```

41.
42. if __name__ == '__main__':
43.
44.     #添加
45.     add_plpgsql()
46.     selectData()

```

说明：在 psycopg2 中调用 add\_plpgsql() 函数即可，add\_plpgsql()函数中嵌入 pgsql 语句 add\_num\_cn(2,1)

源代码截图：



```

pg_test.py
1  # -*- coding: utf-8 -*-
2  """
3  Created on Mon Nov  2 18:14:39 2020
4
5  @author: bayonet8163655
6  """
7
8  import psycopg2
9
10 # 连接数据库
11 def connectPostgreSQL():
12     conn = psycopg2.connect(database="weapons", user="postgres",
13                             password="123456", host="127.0.0.1", port="5432")
14     print ("connect successful!")
15     return conn
16
17 # 断开数据库连接
18 def disconnectPostgreSQL(conn):
19     conn.commit()#提交所有未提交操作到数据库
20     conn.close()#断开数据库连接
21
22 # 添加
23 def add_plpgsql():
24     select_sql="select add_num_cn(2,1)"
25     conn=connectPostgreSQL()
26     cur=conn.cursor()
27     cur.execute(select_sql)
28     disconnectPostgreSQL(conn)
29
30 # 查询信息
31 def selectData():
32     select_sql="select * from weapon"
33     conn=connectPostgreSQL()
34     cur=conn.cursor()
35     cur.execute(select_sql)
36     rows=cur.fetchall()
37     for row in rows:
38         print("weapon_id=",row[0],"\tweapon_type=",row[1],
39               "\tnum_cn=",row[2],"\tnum_usa=",row[3],"\n")
40     disconnectPostgreSQL(conn)
41
42 if __name__ == '__main__':
43
44     # 添加
45     add_plpgsql()
46     selectData()
47

```

执行结果:

```
Variable explorer  Help  Plots  Files
Console 1/A x
In [7]: runfile('D:/code/python_project/pg_test.py', wdir='D:/code/python_project')
connect successful!
connect successful!
weapon_id= 1    weapon_type= 火炮        num_cn= 198    num_usa= 1428
weapon_id= 3    weapon_type= 坦克        num_cn= 0      num_usa= 430
weapon_id= 2    weapon_type= 汽车        num_cn= 121   num_usa= 7000
```

weapons/postgres@PostgreSQL 10

Query Editor Query History

```
1 select * from weapon
```

Data Output Explain Messages Notifications

	weapon_id [PK] integer	weapon_type character varying (10)	num_cn integer	num_usa integer
1	1	火炮	198	1428
2	3	坦克	0	430
3	2	汽车	121	7000

(5) 使用 psycopg2 编程, 删除 weapon 数据表的所有记录。

程序执行前状态:

weapons/postgres@PostgreSQL 10

Query Editor Query History

```
1 select * from weapon
```

Data Output Explain Messages Notifications

	weapon_id [PK] integer	weapon_type character varying (10)	num_cn integer	num_usa integer
1	1	火炮	198	1428
2	3	坦克	0	430
3	2	汽车	121	7000

可编辑格式的源代码:

```
1. # -*- coding: utf-8 -*-
2. """
3. Created on Mon Nov  2 18:14:39 2020
4.
5. @author: bayonet8163655
6. """
7.
8. import psycopg2
9.
10. #连接数据库
11. def connectPostgreSQL():
12.     conn = psycopg2.connect(database="weapons", user="postgres",
13.                             password="123456", host="127.0.0.1",
14.                             port="5432")
15.     print ("connect successful!")
16.     return conn
17. #断开数据库连接
18. def disconnectPostgreSQL(conn):
19.     conn.commit()#提交所有未提交操作到数据库
20.     conn.close()#断开数据库连接
21.
22. #删除信息
23. def deletetData():
24.     select_sql="delete from weapon"
25.     conn=connectPostgreSQL()
26.     cur=conn.cursor()
27.     cur.execute(select_sql)
28.     disconnectPostgreSQL(conn)
29.
30. if __name__ == '__main__':
31.
32.     #删除
33.     deletetData()
```

说明: 在函数 `deletetData()` 中嵌入 `pgsql` 语句 “`delete from weapon`”, 从而删除 `weapon` 数据表的所有记录。


源代码截图:

```

pg_test.py
1  # -*- coding: utf-8 -*-
2  """
3  Created on Mon Nov  2 18:14:39 2020
4
5  @author: bayonet8163655
6  """
7
8  import psycopg2
9
10 # 连接数据库
11 def connectPostgreSQL():
12     conn = psycopg2.connect(database="weapons", user="postgres",
13                             password="123456", host="127.0.0.1", port="5432")
14     print("connect successful!")
15     return conn
16
17 # 断开数据库连接
18 def disconnectPostgreSQL(conn):
19     conn.commit()#提交所有未提交操作到数据库
20     conn.close()#断开数据库连接
21
22 # 删除信息
23 def deletetData():
24     select_sql="delete from weapon"
25     conn=connectPostgreSQL()
26     cur=conn.cursor()
27     cur.execute(select_sql)
28     disconnectPostgreSQL(conn)
29
30 if __name__=='__main__':
31     # 删除
32     deletetData()
33

```

执行结果:

 weapons/postgres@PostgreSQL 10

Query Editor

Query History

1






```
select * from weapon;
```

Data Output

Explain

Messages

Notifications

<div> weapon_id</div> <div>[PK] integer </div>	<div>weapon_type</div> <div>character varying (10) </div>	<div>num_cn</div> <div>integer </div>	<div>num_usa</div> <div>integer </div>