

# 实验 嵌入式 pgSQL/基于诺贝尔奖得主数据库

## 1. 实验目的

- (1) 掌握通过嵌入式 SQL（宿主语言为 C 语言）编程访问数据库的基本步骤，并对数据库进行各种数据操作。
- (2) 理解游标的概念并掌握游标的用法。
- (3) 弘扬科学精神，建设创新型国家，促进人类进步。

## 2. 实验环境

- (1) 数据库管理系统：PostgreSQL（含预处理程序 ECPG）。
- (2) 图形化管理工具：pgAdmin。
- (3) 应用程序开发环境：Dev C++。

## 3. 实验内容

- (1) 熟悉预处理程序 ECPG，掌握通过嵌入式 SQL 编程访问数据库的基本步骤。
- (2) 使用 C 语言编程实现对数据库的访问，进行表的创建、插入数据、更新、查询等操作。
- (3) 熟悉游标机制在嵌入式 SQL 编程中的使用方法。

## 4. 实验数据

诺贝尔奖（瑞典语：Nobel priset，英语：Nobel Prize）是指根据诺贝尔 1895 年的遗嘱而设立的五个奖项，包括：物理学奖、化学奖、和平奖、生理学或医学奖和文学奖，旨在表彰在物理学、化学、和平、生理学或医学以及文学上“对人类作出最大贡献”的人士；以及瑞典中央银行 1968 年设立的诺贝尔经济学奖，用于表彰在经济学领域杰出贡献的人。诺贝尔奖每年评选和颁发一次，诺贝尔奖包括一枚金牌、一份证书以及一笔奖金。截止至 2019 年，共授予 919 位个人和 24 个团体，其中 4 位个人以及 1 个团体两次获奖、1 个团体三次获奖。诺贝尔奖被普遍认为是在世界范围内，所有颁奖领域内（物理学、化学、和平、生理学或医学、文学和经济学）能够取得的最高荣誉。诺贝尔奖所要彰显的是科学精神。所谓科学精神就是人类科学实践长期积累下来的各种宝贵精神财富的总称。它们是实事求是的科学态度、开放的科学观念、顽强的科学毅力、远大的科学志向、博大的科学胸怀、崇高的科学理想、坚定的科学信念，科学的人生观、价值观和世界观等诸多精神要素的集合。诺贝尔就是上述科学精神的集大成者。他本人没有经过系统的学校教育，但在科学精神的强有力的支撑下，凭着顽强的科学毅力自学并成功研究了 350 余种专利成果，掌握了 6 门外语。他一向秉持开明开放的科学态度，将工厂分布在世界各地，这样的行为直接推动了人类社会发展的进程。他之所以坚持要研制炸药，正是为了帮助人类杜绝战争。时常萦绕在他耳际的问题是，人类都是兄弟，应该和睦相处，为什么要有战争呢？他倾其所有设立诺贝尔奖，就是为了“以促进人类进步和福利事业，以纯粹的理想主义为目的”，他要致力于人类生存与发展的终极关怀，而不是仅仅局限于科学的进步。于是有人这样评价诺贝尔奖，“迄今在作为个人以促进人类进步和福利事业，并以纯粹的理想主义为目的而留给人类的赠礼之中，这恐怕是空前的”。大力建设创新型国家的形势下，认识和弘扬诺贝尔及诺贝尔奖所凝练的科学精神不仅具有迫切的现实意义，也具有深远的历史意义。诺贝尔奖数据库(nobel prize)包括诺贝尔奖得主表(nobel\_laureate)，诺贝尔奖得主表包含诺贝尔奖得主编号（np\_id），诺贝

尔奖得主的姓名（np\_name），诺贝尔奖得主所获得的奖项（np\_award），诺贝尔奖得主的年龄（np\_age），诺贝尔奖得主的国籍（np\_nationality）。

## 5. 实验作业

(1) 创建数据表 `nobel_laureate`。

属性	类型	长度	是否为主键
np_id	int		是
np_name	character	20	否
np_award	character	20	否
np_age	int		否
np_nationality	character	20	否

首先打开 pgAdmin，创建 `nobel prize` 数据库

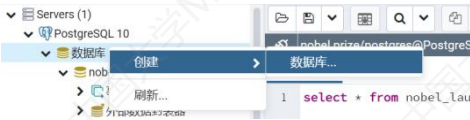


图 1

创建结果如下



图 2

接下来打开记事本，向记事本中输入以下语句

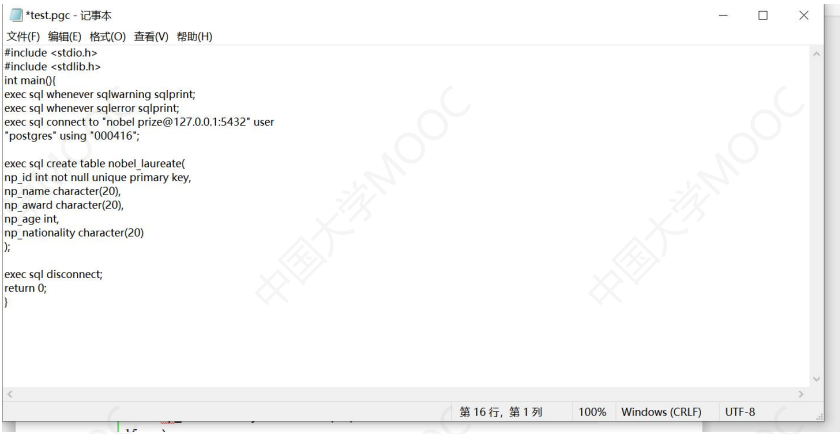


图 3

可执行代码如下


```
1. #include <stdio.h>
2. #include <stdlib.h>
3. int main(){
4.     exec sql whenever sqlwarning sqlprint;
```

```

5.  exec sql whenever sqlerror sqlprint;
6.  exec sql connect to "nobel prize@127.0.0.1:5432" user
7.  "postgres" using "000416";
8.
9.  exec sql create table nobel_laureate(
10. np_id int not null unique primary key,
11. np_name character(20),
12. np_award character(20),
13. np_age int,
14. np_nationality character(20)
15. );
16.
17. exec sql disconnect;
18. return 0;
19. }

```

保存文件名为 test 并修改文件后缀为.pgc 文件，运行 cmd，首先预编译源代码，在 cmd 中输入如下语句



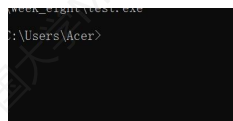
```

C:\Users\Acer>ecpg -t -c -I E:\PostgreSQL\10\include -o F:\数据库系统原理\week_eight\test.c F:\数据库系统原理\week_eight\test.pgc

```

图 4

语句执行返回为空，显示成功执行



```

ecpg -t -c -I E:\PostgreSQL\10\include -o F:\数据库系统原理\week_eight\test.c F:\数据库系统原理\week_eight\test.pgc
F:\数据库系统原理\week_eight\test.c
C:\Users\Acer>

```

图 5

接下来编译与执行代码，在 cmd 中输入如下语句



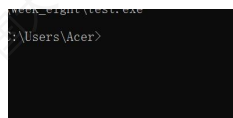
```

C:\Users\Acer>gcc -I E:\PostgreSQL\10\include -Wall -g F:\数据库系统原理\week_eight\test.c -L E:\PostgreSQL\10\lib -lecp -lpq -lpqtypes -o F:\数据库系统原理\week_eight\test.exe

```

图 6

语句执行返回为空，显示成功执行



```

gcc -I E:\PostgreSQL\10\include -Wall -g F:\数据库系统原理\week_eight\test.c -L E:\PostgreSQL\10\lib -lecp -lpq -lpqtypes -o F:\数据库系统原理\week_eight\test.exe
F:\数据库系统原理\week_eight\test.exe
C:\Users\Acer>

```

图 7

双击 test.exe 运行可执行文件，打开 pgAdmin，查看数据表创建情况

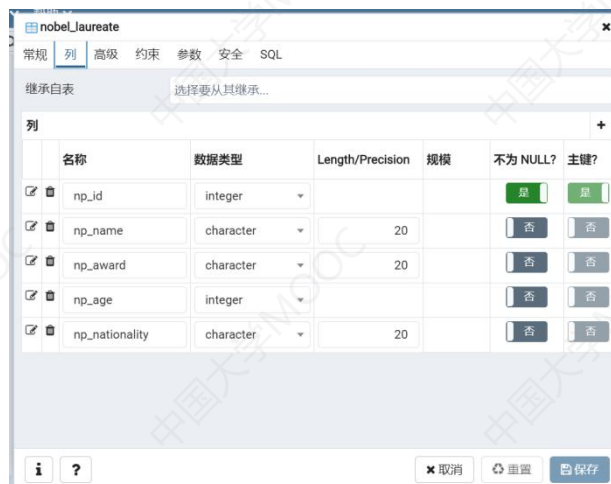


图 8

(2) 向 nobel\_laureate 表中插入数据。

np_id	np_name	np_award	np_age	np_nationality
202001	Roger_Penrose	Physics	89	GBR
202002	Jennifer_Doudna	Chemistry	56	USA
202003	Paul_R.Milgrom	Economic_Sciences	72	FRA
202004	Louise_Gluck	Literature	77	USA
201201	Mo_yan	Literature	65	CHN
201501	Tu_Youyou	Physiology	90	CHN

打开记事本，向记事本中输入以下语句

```

*test.pgc - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
#include <stdio.h>
#include <stdlib.h>
int main()
{
    exec sql whenever sqlwarning sqlprint;
    exec sql whenever sqlerror sqlprint;
    exec sql connect to "nobel prize@127.0.0.1:5432" user
    "postgres" using "000416";

    exec sql insert into nobel_laureate values
    (202001,'Roger_Penrose','Physics',89,'GBR'),
    (202002,'Jennifer_Doudna','Chemistry',56,'USA'),
    (202003,'Paul_R.Milgrom','Economic_Sciences',72,'FRA'),
    (202004,'Louise_Gluck','Literature',77,'USA'),
    (201201,'Mo_yan','Literature',65,'CHN'),
    (201501,'Tu_Youyou','Physiology',90,'CHN');
    exec sql disconnect;
    return 0;
}

```

图 9

可执行代码如下

1. #include <stdio.h>
2. #include <stdlib.h>
3. int main(){
4. exec sql whenever sqlwarning sqlprint;
5. exec sql whenever sqlerror sqlprint;
6. exec sql connect to "nobel prize@127.0.0.1:5432" user
7. "postgres" using "000416";

```

8.
9.  exec sql insert into nobel_laureate values
10. (202001,'Roger_Penrose','Physics',89,'GBR'),
11. (202002,'Jennifer_Doudna','Chemistry',56,'USA'),
12. (202003,'Paul_R.Milgrom','Economic_Sciences',72,'FRA'),
13. (202004,'Louise_Gluck','Literature',77,'USA'),
14. (201201,'Mo_yan','Literature',65,'CHN'),
15. (201501,'Tu_Youyou','Physiology',90,'CHN');
16. exec sql disconnect;
17. return 0;
18. }

```

保存文件名为 test 并修改文件后缀为.pgc 文件，运行 cmd，首先预编译源代码，在 cmd 中输入如下语句



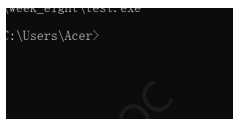
```

C:\Users\Acer>ecpg -t -c -I E:\PostgreSQL\10\include -o F:\数据库系统原理\week_eight\test.c F:\数据库系统原理\week_eight\test.pgc

```

图 10

语句执行返回为空，显示成功执行



```

C:\Users\Acer>

```

图 11

接下来编译与执行代码，在 cmd 中输入如下语句



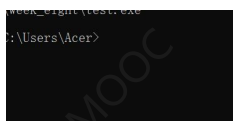
```

C:\Users\Acer>gcc -I E:\PostgreSQL\10\include -Wall -g F:\数据库系统原理\week_eight\test.c -L E:\PostgreSQL\10\lib -lecp -lpq -lpqtypes -o F:\数据库系统原理\week_eight\test.exe

```

图 12

语句执行返回为空，显示成功执行



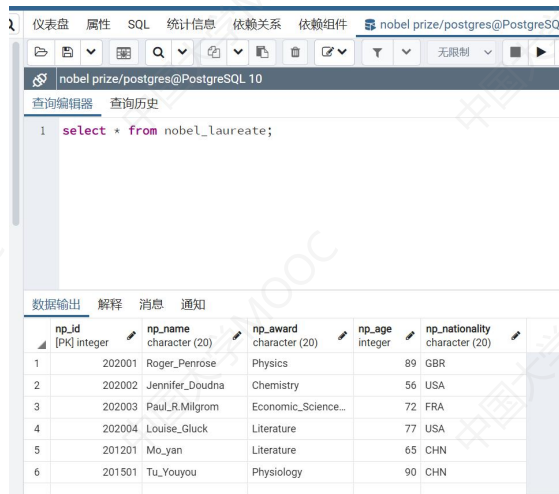
```

C:\Users\Acer>

```

图 13

双击 test.exe 运行可执行文件，打开 pgAdmin，查看数据插入情况



The screenshot shows a PostgreSQL query editor with the following query: `select * from nobel_laureate;`

The results are displayed in a table with the following columns: `np_id`, `np_name`, `np_award`, `np_age`, and `np_nationality`.

np_id	np_name	np_award	np_age	np_nationality
202001	Roger_Penrose	Physics	89	GBR
202002	Jennifer_Doudna	Chemistry	56	USA
202003	Paul_R_Milgrom	Economic_Science...	72	FRA
202004	Louise_Gluck	Literature	77	USA
201201	Mo_yan	Literature	65	CHN
201501	Tu_Youyou	Physiology	90	CHN

图 14

- (3) 查询 nobel\_laureate 表中 np\_id 号为 201201 的诺贝尔奖得主信息。  
打开记事本，向记事本中输入以下语句

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  int main()
4  {
5      exec sql whenever sqlwarning sqlprint;
6      exec sql whenever sqlerror sqlprint;
7      exec sql connect to "nobel prize@127.0.0.1:5432" user
8      "postgres" using "000416";
9
10     EXEC SQL BEGIN DECLARE SECTION;
11     int t_id;
12     char t_name[20];
13     char t_award[20];
14     int t_age;
15     char t_nationality[20];
16     EXEC SQL END DECLARE SECTION;
17
18     exec sql select *
19     into :t_id,:t_name,:t_award,:t_age,:t_nationality
20     from nobel_laureate
21     where np_id=201201;
22
23     exec sql disconnect;
24
25     printf("%d\t%s\t%s\t%d\t%s",t_id,t_name,t_award,t_age,t_nationality);
26     system("pause");
27 }
return 0;
}

```

图 15

可执行代码如下

```

1.  #include <stdio.h>
2.  #include <stdlib.h>
3.  int main(){
4.      exec sql whenever sqlwarning sqlprint;
5.      exec sql whenever sqlerror sqlprint;
6.      exec sql connect to "nobel prize@127.0.0.1:5432" user
7.      "postgres" using "000416";
8.
9.      EXEC SQL BEGIN DECLARE SECTION;
10.     int t_id;
11.     char t_name[20];
12.     char t_award[20];
13.     int t_age;
14.     char t_nationality[20];

```

```

15. EXEC SQL END DECLARE SECTION;
16.
17. exec sql select *
18. into :t_id,:t_name,:t_award,:t_age,:t_nationality
19. from nobel_laureate
20. where np_id=201201;
21.
22. exec sql disconnect;
23.
24. printf("%d\t%s\t%s\t%d\t%s",t_id,t_name,t_award,t_age,t_nationality);
25. system("pause");
26.
27. return 0;
28. }

```

保存文件名为 test 并修改文件后缀为.pgc 文件，运行 cmd，首先预编译源代码，在 cmd 中输入如下语句

```

C:\Users\Acer>ecpg -t -c -I E:\PostgreSQL\10\include -o F:\数据库系统原理\week_eight\test.c F:\数据库系统原理\week_eight\test.pgc

```

图 16

语句执行返回为空，显示成功执行

```

week_eight(test.exe)
C:\Users\Acer>

```

图 17

接下来编译与执行代码，在 cmd 中输入如下语句

```

C:\Users\Acer>gcc -I E:\PostgreSQL\10\include -Wall -g F:\数据库系统原理\week_eight\test.c -L E:\PostgreSQL\10\lib -lecp -lpq -lpqtypes -o F:\数据库系统原理\week_eight\test.exe

```

语句执行返回为空，显示成功执行

```

week_eight(test.exe)
C:\Users\Acer>

```

图 18

双击 test.exe 运行可执行文件，程序运行框显示结果如下

```

F:\数据库系统原理\week_eight\test.exe
201201 Mo yan Literature 65 CHN
请按任意键继续...

```



图 19

查询成功

- (4) 使用游标机制查询 nobel\_laureate 表中的所有记录。  
打开记事本，向记事本中输入以下语句

```
["]test.pgc
13 EXEC SQL BEGIN DECLARE SECTION;
14 int t_id;
15 char t_name[20];
16 char t_award[20];
17 int t_age;
18 char t_nationality[20];
19 EXEC SQL END DECLARE SECTION;
20
21 exec sql declare quesqr CURSOR for
22 select *
23 from nobel_laureate
24 for read only;
25
26 exec sql open quesqr;
27
28 while(1){
29 exec sql fetch next from quesqr
30 into :t_id,:t_name,:t_award,:t_age,:t_nationality;
31 if(No_more_tuples)break;
32 printf("%d\t%s\t%s\t%d\t%s\n",t_id,t_name,t_award,t_age,t_nationality);
33 }
34
35 exec sql close quesqr;
36
37
38 exec sql disconnect;
39
40 system("pause");
41
42 return 0;
43 }
```

图 20

可执行代码如下

```
1. #include <stdio.h>
2. #include <stdlib.h>
3. #define No_more_tuples !(strcmp(sqlca.sqlstate,"02000"))
4.
5. int main(){
6. exec sql whenever sqlwarning sqlprint;
7. exec sql whenever sqlerror sqlprint;
8. exec sql connect to "nobel prize@127.0.0.1:5432" user
9. "postgres" using "000416";
10. exec sql begin work;
11.
12.
13. EXEC SQL BEGIN DECLARE SECTION;
14. int t_id;
15. char t_name[20];
16. char t_award[20];
17. int t_age;
18. char t_nationality[20];
19. EXEC SQL END DECLARE SECTION;
20.
21. exec sql declare quesqr CURSOR for
22. select *
23. from nobel_laureate
24. for read only;
25.
```



```

26.  exec sql open queschr;
27.
28.  while(1){
29.  exec sql fetch next from queschr
30.  into :t_id,:t_name,:t_award,:t_age,:t_nationality;
31.  if(No_more_tuples)break;
32.  printf("%d\t%s\t%s\t%d\t%s\n",t_id,t_name,t_award,t_age,t_nationality);
33.  }
34.
35.  exec sql close queschr;
36.  exec sql disconnect;
37.
38.  system("pause");
39.
40.  return 0;
41.  }

```

保存文件名为 test 并修改文件后缀为.pgc 文件，运行 cmd，首先预编译源代码，在 cmd 中输入如下语句



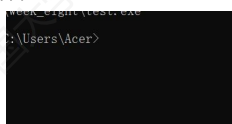
```

C:\Users\Acer>pgcc -t -I E:\PostgreSQL\10\include -o F:\数据库系统原理\week_eight\test.c F:\数据库系统原理\week_eight\test.pgc

```

图 21

语句执行返回为空，显示成功执行



```

C:\Users\Acer>

```

图 22

接下来编译与执行代码，在 cmd 中输入如下语句



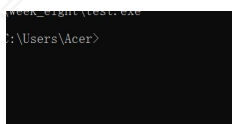
```

C:\Users\Acer>pgcc -t -I E:\PostgreSQL\10\include -Wall -g F:\数据库系统原理\week_eight\test.c -L E:\PostgreSQL\10\lib -lecp -lpq -lpqtypes -o F:\数据库系统原理\week_eight\test.exe

```

图 23

语句执行返回为空，显示成功执行



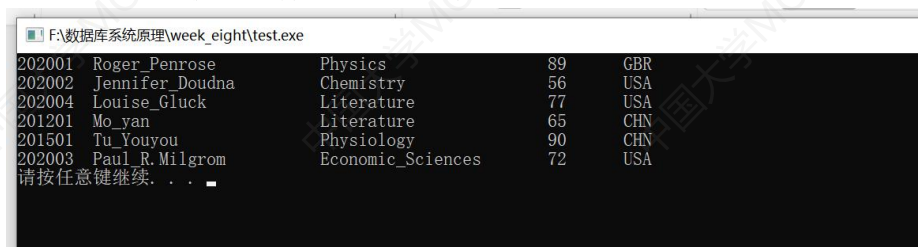
```

C:\Users\Acer>

```

图 24

双击 test.exe 运行可执行文件，程序运行框显示结果如下



202001	Roger_Penrose	Physics	89	GBR
202002	Jennifer_Doudna	Chemistry	56	USA
202004	Louise_Gluck	Literature	77	USA
201201	Mo_yan	Literature	65	CHN
201501	Tu_Youyou	Physiology	90	CHN
202003	Paul_R_Milgrom	Economic_Sciences	72	USA

请按任意键继续...

图 25

查询成功

- (5) 将 nobel\_laureate 表中 np\_id 号为 202003 的诺贝尔奖得主所在国籍改为美国(USA)。  
打开记事本，向记事本中输入以下语句

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #define No_more_tuples !(strcmp(SQLSTATE,"02000"))
4
5 int main(){
6     exec sql whenever sqlwarning sqlprint;
7     exec sql whenever sqlerror sqlprint;
8     exec sql connect to "nobel prize@127.0.0.1:5432" user
9     "postgres" using "000416";
10
11
12     exec sql update nobel_laureate
13     set np_nationality = 'USA'
14     where np_id = 202003;
15
16     //into :t_id,:t_name,:t_award,:t_age,:t_nationality;
17     //printf("%d\t%s\t%s\t%d\t%s",t_id,t_name,t_award,t_age,t_nationality);
18
19     exec sql disconnect;
20
21     //system("pause");
22
23     return 0;
24 }

```

图 26

可执行代码如下

```

1.  #include <stdio.h>
2.  #include <stdlib.h>
3.  #define No_more_tuples !(strcmp(SQLSTATE,"02000"))
4.
5.  int main(){
6.      exec sql whenever sqlwarning sqlprint;
7.      exec sql whenever sqlerror sqlprint;
8.      exec sql connect to "nobel prize@127.0.0.1:5432" user
9.      "postgres" using "000416";
10.
11.      exec sql update nobel_laureate
12.      set np_nationality = 'USA'
13.      where np_id = 202003;
14.
15.      //into :t_id,:t_name,:t_award,:t_age,:t_nationality;
16.      //printf("%d\t%s\t%s\t%d\t%s",t_id,t_name,t_award,t_age,t_nationality);
17.
18.      exec sql disconnect;
19.
20.      //system("pause");
21.
22.      return 0;
23.  }

```

保存文件名为 test 并修改文件后缀为.pgc 文件，运行 cmd，首先预编译源代码，在

cmd 中输入如下语句

```
C:\Users\Acer>gcc -t -c -I E:\PostgreSQL\10\include -o F:\数据库系统原理\week_eight\test.c F:\数据库系统原理\week_eight\test.pgc
```

图 27

语句执行返回为空，显示成功执行

```
week_eight\test.exe  
C:\Users\Acer>
```

图 28

接下来编译与执行代码，在 cmd 中输入如下语句

```
C:\Users\Acer>gcc -I E:\PostgreSQL\10\include -Wall -g F:\数据库系统原理\week_eight\test.c -L E:\PostgreSQL\10\lib -lcp  
g -lpq -lpqtypes -o F:\数据库系统原理\week_eight\test.exe
```

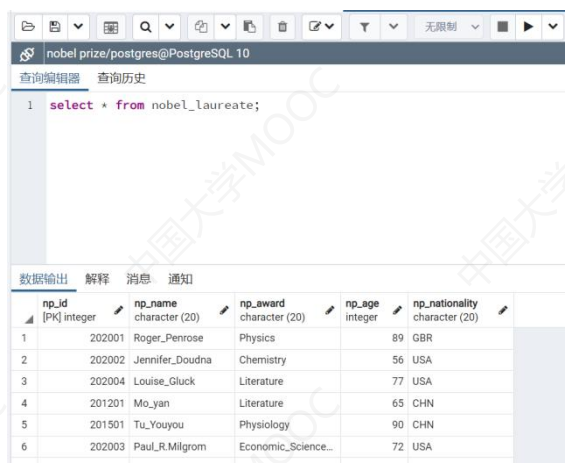
图 29

语句执行返回为空，显示成功执行

```
week_eight\test.exe  
C:\Users\Acer>
```

图 30

双击 test.exe 运行可执行文件，打开 pgAdmin，查看数据修改情况

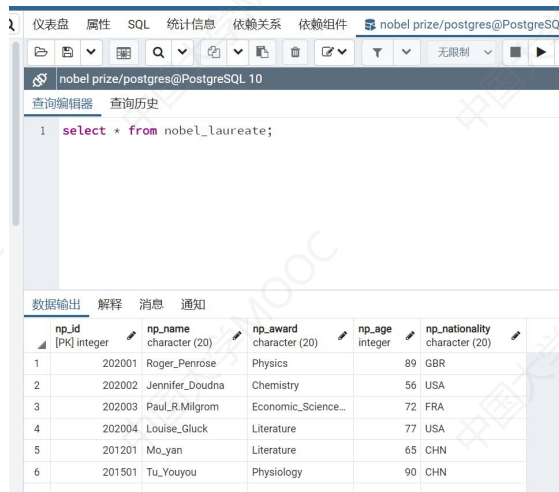


The screenshot shows the pgAdmin interface with a SQL query editor and a results table. The query is `select * from nobel_laureate;`. The results table has 6 rows and 5 columns: `np_id`, `np_name`, `np_award`, `np_age`, and `np_nationality`.

np_id	np_name	np_award	np_age	np_nationality
1	202001 Roger_Penrose	Physics	89	GBR
2	202002 Jennifer_Doudna	Chemistry	56	USA
3	202004 Louise_Gluck	Literature	77	USA
4	201201 Mo_yan	Literature	65	CHN
5	201501 Tu_Youyou	Physiology	90	CHN
6	202003 Paul_R_Milgrom	Economic_Science...	72	USA

图 31

可与之前数据查询情况进行对比，可以看到 np\_id 为 202003 的科学家国籍发生改变，数据修改成功



The screenshot shows a PostgreSQL query editor with the following query: `select * from nobel_laureate;`

The results are displayed in a table with the following columns: `np_id` (integer), `np_name` (character (20)), `np_award` (character (20)), `np_age` (integer), and `np_nationality` (character (20)).

np_id	np_name	np_award	np_age	np_nationality
202001	Roger_Penrose	Physics	89	GBR
202002	Jennifer_Doudna	Chemistry	56	USA
202003	Paul_R_Milgrom	Economic_Science...	72	FRA
202004	Louise_Gluck	Literature	77	USA
201201	Mo_yan	Literature	65	CHN
201501	Tu_Youyou	Physiology	90	CHN

图 32

- (6) 动态构造 pgSQL 语句，将 nobel\_laureate 表中 np\_id 号为 201201 的诺贝尔奖得主年龄增加 m（m 由键盘输入）。  
打开记事本，向记事本中输入以下语句

```

1 int main() {
2     exec sql whenever sqlwarning sqlprint;
3     exec sql whenever sqlerror sqlprint;
4     exec sql connect to "nobel prize@127.0.0.1:5432" user
5     "postgres" using "000416";
6     exec sql begin work;
7
8     EXEC SQL BEGIN DECLARE SECTION;
9     //int t_id;
10    //char t_name[20];
11    //char t_award[20];
12    //int t_age;
13    //char t_nationality[20];
14    int m;
15    EXEC SQL END DECLARE SECTION;
16
17    printf("Please input addition age of np_id=201201:");
18    scanf("%d",&m);
19    exec sql prepare t from "update nobel_laureate set np_age = (np_age+?) where np_id = 201201:";
20    exec sql execute t using :m;
21    exec sql execute immediate "commit;";
22
23    |
24    exec sql disconnect;
25
26    system("pause");
27
28    return 0;
29 }

```

图 33

可执行代码如下

```

1. #include <stdio.h>
2. #include <stdlib.h>
3. #define No_more_tuples !(strcmp(sqlca.sqlstate,"02000"))
4.
5. int main(){
6.     exec sql whenever sqlwarning sqlprint;
7.     exec sql whenever sqlerror sqlprint;
8.     exec sql connect to "nobel prize@127.0.0.1:5432" user
9.     "postgres" using "000416";
10.    exec sql begin work;
11.
12.
13.    EXEC SQL BEGIN DECLARE SECTION;

```

```

14. //int t_id;
15. //char t_name[20];
16. //char t_award[20];
17. //int t_age;
18. //char t_nationality[20];
19. int m;
20. EXEC SQL END DECLARE SECTION;
21.
22. printf("Please input addition age of np_id=201201:");
23. scanf("%d",&m);
24. exec sql prepare t from "update nobel_laureate set np_age = (n
    p_age+?) where np_id = 201201;";
25. exec sql execute t using :m;
26. exec sql execute immediate "commit;";
27.
28. exec sql disconnect;
29.
30. system("pause");
31.
32. return 0;
33. }

```

保存文件名为 test 并修改文件后缀为.pgc 文件，运行 cmd，首先预编译源代码，在 cmd 中输入如下语句

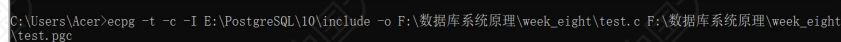


图 34

语句执行返回为空，显示成功执行

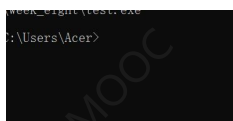


图 35

接下来编译与执行代码，在 cmd 中输入如下语句



图 36

语句执行返回为空，显示成功执行



图 37

双击 test.exe 运行可执行文件，弹出运行界面，在提示语 “Please input addition age of np\_id=201201” 后输入 “1”，回车运行，打开 pgAdmin，查看数据修改情况

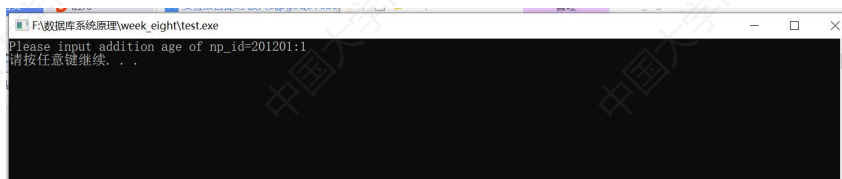


图 38

未修改前数据库数据如下：

np_id	np_name	np_award	np_age	np_nationality	
202001	Roger_Penrose	Physics	89	GBR	
202002	Jennifer_Doudna	Chemistry	56	USA	
202004	Louise_Gluck	Literature	77	USA	
201201	Mo_yan	Literature	65	CHN	
201501	Tu_Youyou	Physiology	90	CHN	
202003	Paul_R.Milgrom	Economic_Science...	72	USA	

图 39

修改后数据库数据如下：

np_id	np_name	np_award	np_age	np_nationality	
202001	Roger_Penrose	Physics	89	GBR	
202002	Jennifer_Doudna	Chemistry	56	USA	
202004	Louise_Gluck	Literature	77	USA	
201501	Tu_Youyou	Physiology	90	CHN	
202003	Paul_R.Milgrom	Economic_Science...	72	USA	
201201	Mo_yan	Literature	66	CHN	

图 40

- (7) 使用卷游标按顺序输出 nobel\_laureate 表中的第一条记录、下一条记录、最后一条记录和前一条记录。
- 打开记事本，向记事本中输入以下语句

```

["]test.pgc
19 EXEC SQL END DECLARE SECTION;
20
21 exec sql declare quesqr SCROLL CURSOR for
22 select *
23 from nobel_laureate
24 for read only;
25
26 exec sql open quesqr;
27
28 exec sql fetch first from quesqr
29 into :t_id,:t_name,:t_award,:t_age,:t_nationality;
30 printf("%d\t%s\t%s\t%d\t%s\n",t_id,t_name,t_award,t_age,t_nationality);
31
32 exec sql fetch next from quesqr
33 into :t_id,:t_name,:t_award,:t_age,:t_nationality;
34 printf("%d\t%s\t%s\t%d\t%s\n",t_id,t_name,t_award,t_age,t_nationality);
35
36 exec sql fetch absolute -1 from quesqr
37 into :t_id,:t_name,:t_award,:t_age,:t_nationality;
38 printf("%d\t%s\t%s\t%d\t%s\n",t_id,t_name,t_award,t_age,t_nationality);
39
40 exec sql fetch backward 1 from quesqr
41 into :t_id,:t_name,:t_award,:t_age,:t_nationality;
42 printf("%d\t%s\t%s\t%d\t%s\n",t_id,t_name,t_award,t_age,t_nationality);
43 exec sql close quesqr;
44 exec sql disconnect;
45
46 system("pause");
47
48 return 0;
49

```

图 41

可执行代码如下

```

1.  #include <stdio.h>
2.  #include <stdlib.h>
3.  #define No_more_tuples !(strcmp(sqlca.sqlstate,"02000"))
4.
5.  int main(){
6.      exec sql whenever sqlwarning sqlprint;
7.      exec sql whenever sqlerror sqlprint;
8.      exec sql connect to "nobel prize@127.0.0.1:5432" user
9.      "postgres" using "000416";
10.     exec sql begin work;
11.
12.
13.     EXEC SQL BEGIN DECLARE SECTION;
14.     int t_id;
15.     char t_name[20];
16.     char t_award[20];
17.     int t_age;
18.     char t_nationality[20];
19.     EXEC SQL END DECLARE SECTION;
20.
21.     exec sql declare quesqr SCROLL CURSOR for
22.     select *
23.     from nobel_laureate
24.     for read only;
25.
26.     exec sql open quesqr;
27.
28.     exec sql fetch first from quesqr
29.     into :t_id,:t_name,:t_award,:t_age,:t_nationality;

```



```

30. printf("%d\t%s\t%s\t%d\t%s\n",t_id,t_name,t_award,t_age,t_nati
    onality);
31.
32. exec sql fetch next from quesqr
33. into :t_id,:t_name,:t_award,:t_age,:t_nationality;
34. printf("%d\t%s\t%s\t%d\t%s\n",t_id,t_name,t_award,t_age,t_nati
    onality);
35.
36. exec sql fetch absolute -1 from quesqr
37. into :t_id,:t_name,:t_award,:t_age,:t_nationality;
38. printf("%d\t%s\t%s\t%d\t%s\n",t_id,t_name,t_award,t_age,t_nati
    onality);
39.
40. exec sql fetch backward 1 from quesqr
41. into :t_id,:t_name,:t_award,:t_age,:t_nationality;
42. printf("%d\t%s\t%s\t%d\t%s\n",t_id,t_name,t_award,t_age,t_nati
    onality);
43.
44.
45. exec sql close quesqr;
46. exec sql disconnect;
47.
48. system("pause");
49.
50. return 0;
51. }

```

保存文件名为 test 并修改文件后缀为.pgc 文件，运行 cmd，首先预编译源代码，在 cmd 中输入如下语句



图 42

语句执行返回为空，显示成功执行

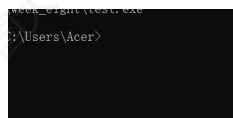


图 43

接下来编译与执行代码，在 cmd 中输入如下语句



图 44

语句执行返回为空，显示成功执行

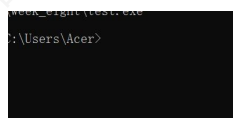


图 45

双击 test.exe 运行可执行文件，程序运行框显示结果如下

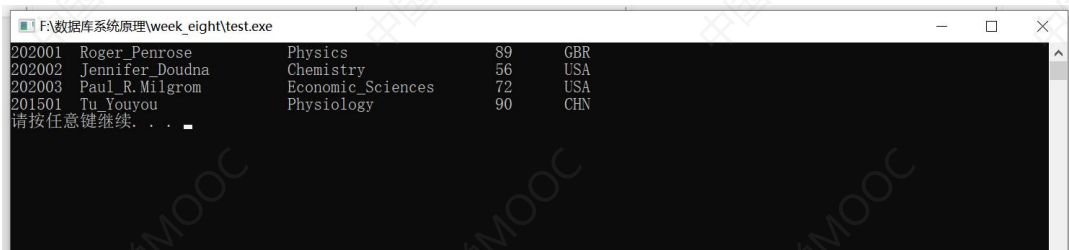


图 46