

CSCI 3430.2 - Principles of Programming Languages

Solution for Assignment #5

Question 1 – SQL

In order to learn and test the SQL sentences, I use MySql which is installed in my laptop.

The first step is to create the tables and insert the test data, the screen shot is as below:

tbllabmark: 6 rows total (approximately)

studentNo	labNo	labMark
79,004,651	1	6
79,004,651	2	7
96,000,001	1	6
96,000,002	2	7
96,000,003	2	8
96,000,005	2	7

tblstudentno: 5 rows total (approximately)

studentNo	lastName	firstName
79,004,651	Somers	Curtis
96,000,001	Unknown	(NULL)
96,000,002	Day	Green
96,000,003	Oven	Beth
96,000,004	Lisp	Franz

tbltestmark: 8 rows total (approximately)

studentNo	testNo	testMark
79,004,651	1	70
79,004,651	2	60
96,000,001	1	50
96,000,001	2	50
96,000,002	1	60
96,000,003	1	70
96,000,003	2	80
96,000,002	2	90

The below SQL sentences have been tested in my MySQL environment.

a. Print the last name, a comma, then first name from concatenated together as one column from the the tblStudentNo table sorted by firstName in ascending order.

```
select concat_ws(',',lastname , firstName) as name from  
tblstudentno order by firstName asc;
```

b. Show the number of student records in the tblStudentNo table.

```
select count(*) from tblstudentno
```

c. Select the first and last names of the lowest numbered studentNo in the tblStudentNo table

```
select firstname, lastname from tblstudentno where studentno in  
(select min(studentno) from tblstudentno)
```

d. Select the name of the student who did not pass in a lab

```
select a.firstName, a.lastName from tblstudentno a left join  
tbllabmark b on a.studentNo = b.studentNo  
where (IFNULL(b.labMark, 0))<5
```

e. Select the student firstName, lastName that had the highest lab mark

```
select firstname, lastname from tblstudentno where studentno in  
(select studentno from tbllabmark where labmark in (select  
max(labMark) from tbllabmark))
```

f. Select the average of lab #2 (limited of course to those four lab #2 actually passed in).

```
select avg(labmark) from tbllabmark where labno=2
```

g. Calculate term marks assuming the equation:

$$((\text{Lab\#1}/10)*15) + ((\text{Lab\#2}/10))*15) + ((\text{Midterm\#1}/100)*30) + ((\text{Midterm\#2}/100)*40)$$

Remember to replace nulls with zeroes.

```
select t.studentno,sum(t.score) from  
(  
(
```

```
select studentno,sum((labmark/10)*15) as score from tbllabmark
group by studentno
)
```

union

(

```
select t.studentno, sum(t.test) as score from
```

(

```
select studentno, ((testmark/100)*30) as test from tbltestmark
where testno=1
```

union

```
select studentno, ((testmark/100)*40)as test from tbltestmark
where (testno=2)
```

) as t

```
group by t.studentno
```

)

) as t group by t.studentno

h. Select the firstName, lastName and final mark of the student with the highest final mark using the equation in g.

```
select s2.firstname, s2.lastname from tblstudentno as s2 where
s2.studentno in (
```

```
select s1.studentno from (
```

```
select s.studentno, max(s.score) from (
```

```
select t.studentno,sum(t.score) as score from
```

```

(
(select studentno,sum((labmark/10)*15) as score from tbllabmark
group by studentno)
union
(
select t.studentno, sum(t.test)  as score from
(
select studentno, ((testmark/100)*30) as test from tbltestmark
where testno=1
union
select studentno, ((testmark/100)*40)as test from tbltestmark
where (testno=2)
) as t group by t.studentno
)
) as t group by t.studentno
) as s
) as s1
)

```

Question 2 – C and Pascal

Source code List:

sum.p

sum.c

exercise2.c

sort.c

input.dat

2.1 Write a RECURSIVE factorial program in both C (not C++) and Pascal.

I choose to use the free pascal compiler (fpc) in my computer downloaded from <http://www.freepascal.org/>.

In order to calculate the summation, the source code of pascal is as below:

```
1  program Sum;
2  var a : integer;
3  function Sum(num : integer) : integer;
4  begin
5      if num = 1
6      then Sum := 1
7      else Sum := Sum(num-1) + num
8  end;
9  begin
10     a := Sum(3);
11     writeln(a);
12 end.
```

The screen shot of running result is as below:

```
C:\FPC\3.0.2\bin\i386-win32>fpc.exe Sum.p
Free Pascal Compiler version 3.0.2 [2017/02/13] for i386
Copyright (c) 1993-2017 by Florian Klaempfl and others
Target OS: Win32 for i386
Compiling Sum.p
Linking Sum.exe
11 lines compiled, 0.1 sec, 25760 bytes code, 1252 bytes data

C:\FPC\3.0.2\bin\i386-win32>Hello.exe
Hello, world.

C:\FPC\3.0.2\bin\i386-win32>
```

In order to implement the same function, the c language source code is as below:

```
1 #include <stdio.h>
2 int main()
3 {
4     printf("Sum = %d", addNumbers(3));
5 }
6 int addNumbers(int n)
7 {
8     if(n!=0)        return n + addNumbers(n-1);
9     else return n;
10 }
```

The running result is as below:

```
$ gcc Sum.c
Sum.c: In function 'main':
Sum.c:4:23: warning: implicit declaration of function 'addNumbers' [-Wimplicit-function-declaration]
    printf("Sum = %d",addNumbers(3));
                        ^
$ ./a.exe
Sum = 6
```

From the above result, the c language source code has the less number of source code.

2.2 Convert the following Pascal program into C (not C++). Using comments in the code, Identify at least 4 places where the code is shorter in Pascal or C.

The running result of Pascal is as below:

```
C:\FPC\3.0.2\bin\i386-win32>exercise2.exe
Enter an integer:
3
27
8
1
Well that was easy!
```

In order to implement the same function, the source code of c language is as below:

```
1 //the line 1, #include command is moer than Pascal language
2 #include <stdio.h>
3 int main()
4 {
5     int num;
6     printf("Enter an integer: ");
7     scanf("%d", &num);
8     // in Pascal language, the for loop has less sentences
9     for(int i=num;i>0;i--){
10         printf("%d\n",i*i*i);
11     }
12     printf("Well that was easy!");
13     //in Psacal language, the return sentence is not a must
14     return 0;
15 }
```

The running result is as below:

```
$ gcc exercise2.c
```

```
$ ./a.exe
Enter an integer: 3
27
8
1
Well that was easy!
```

3. Write the equivalent C program translation (NOT C++) for the following Pascal program. The program must compile using gcc (not g++) on the cs machine. Note the use of nested functions for min and swap. This can be translated directly since nested functions are now legal in C. Note “sArray” is passed into the functions by ref. Your C code should hand them by ref as well. A good reference is the Wikipedia Page - Comparison of Pascal and C.

The running result is as below, more details please finding in the source code.

```
$ gcc sort.c
```

```
$ ./a.exe
This program reads numbers from a file called input.dat,
sorts the numbers.
Sorted numbers are as follows:
Please input the filename:input.dat
1.00
2.00
3.00
4.00
5.00
6.00
7.00
8.00
9.00
21.00
22.00
23.00
24.00
25.00
26.00
27.00
28.00
29.00
30.00
100.00
```