# 数据库第三次作业报告

## 樊漆亮

## 1712884

摘要：通过mysql的查询语句将数据库中的数据获取，并利用python编程，对所得数据进行进一步处理。其中将数据重定向到文件中时出现了编码问题。

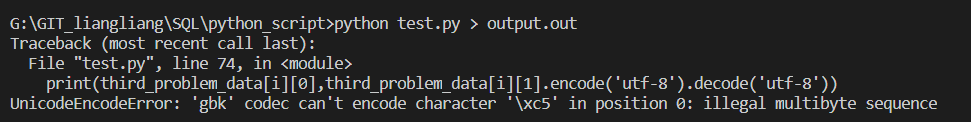
### 源码和输出结果：

##### 源码：

1. #-\*-coding:utf-8-\*-
2. import pymysql
3. import io
4. import sys
5. sys.stdout = io.TextIOWrapper(sys.stdout.buffer,encoding='utf8')
6. db = pymysql.connect("127.0.0.1", "test", "fanqiliang123456", "exercise",charset='utf8')
7. cursor = db.cursor()
8. def first\_problem():
9. first\_problem = '''
10. select dept\_name,count(\*)
11. from student
12. group by dept\_name
13. order by dept\_name DESC
14. '''
15. first\_problem=first\_problem.replace('\n', '')
16. cursor.execute(first\_problem)
17. first\_problem\_data = cursor.fetchall()
18. for i in range(0, len(first\_problem\_data)):
19. print(first\_problem\_data[i][0], first\_problem\_data[i][1])
20. def second\_problem():
21. second\_problem = '''
22. select takes.course\_id,course.title,semester,year
23. from course,takes
24. where(
25. course.course\_id=takes.course\_id
26. and
27. takes.course\_id=course.course\_id
28. )
29. group by takes.course\_id,year,semester
30. having count(distinct takes.ID)>=310
31. '''
32. second\_problem=second\_problem.replace('\n', '')
33. cursor.execute(second\_problem)
34. second\_problem\_data = cursor.fetchall()
35. for i in range(0, len(second\_problem\_data)):
36. print(second\_problem\_data[i][0], second\_problem\_data[i][1]
37. [0:5], second\_problem\_data[i][2], second\_problem\_data[i][3])
38. def third\_problem():
39. third\_problem = '''
40. select distinct ID,name
41. from (
42. select distinct student.ID,student.name
43. from student,takes
44. where (
45. student.ID=takes.ID
46. and
47. takes.course\_id in (
48. select distinct course\_id
49. from teaches
50. where (
51. teaches.ID = (
52. select distinct instructor.ID
53. from instructor
54. where instructor.name='Dale'
55. )
56. and
57. takes.year=teaches.year
58. and
59. takes.semester=teaches.semester
60. )
61. )
62. )
63. ) t;
64. '''
65. third\_problem=third\_problem.replace('\n', ' ')
66. cursor.execute(third\_problem)
67. third\_problem\_data=cursor.fetchall()
68. for i in range (0,len(third\_problem\_data)):
69. print(third\_problem\_data[i][0],third\_problem\_data[i][1].encode('utf-8').decode('utf-8'))
70. def forth\_problem():
71. forth\_problem='''
72. select distinct ID,course\_id,grade
73. from takes
74. where (
75. ID not in(
76. select t.ID
77. from takes t
78. where (
79. t.grade not like 'A%'
80. )
81. )
82. )
83. '''
84. forth\_problem=forth\_problem.replace('\n',' ')
85. cursor.execute(forth\_problem)
86. forth\_problem\_data=cursor.fetchall()
87. for i in range(0,len(forth\_problem\_data)):
88. print(forth\_problem\_data[i])
89. #def fifth\_problem():
90. def fifth\_problem():
91. fifth\_problem='''
92. select avg(budget) avg\_budget
93. from department
94. '''
95. fifth\_problem=fifth\_problem.replace('\n',' ')
96. cursor.execute(fifth\_problem)
97. fifth\_problem\_data=cursor.fetchall()
98. average\_number=fifth\_problem\_data[0][0]
99. fifth\_problem\_1='''
100. select budget
101. from department
102. order by budget ASC
103. '''
104. fifth\_problem\_1=fifth\_problem\_1.replace('\n',' ')
105. cursor.execute(fifth\_problem\_1)
106. budget=cursor.fetchall()
107. size=len(budget)
108. mid\_number=(budget[(size-1)//2][0]+budget[(size)//2][0])/2
109. print("average: ",average\_number)
110. print("median: ",mid\_number)
111. print("FIRST\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")
112. first\_problem()
113. print("SECOND\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")
114. second\_problem()
115. print("THIRD\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")
116. third\_problem()
117. print("FORTH\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")
118. forth\_problem()
119. print("FIFTH\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")
120. fifth\_problem()

##### b) 输出结果：由于太多，见附件-output.txt

### 2. 实验中发现的问题：

在尝试将输出结果重定向到文件中时，出现了如下的错误提示，尝试修改控制台窗口和

输出文件output.txt的编码方式以及test.py源文件的编码方式发现仍然报出同样的错。随后发现python默认标准输出的编码方式不是utf-8，添加这行代码后，编码问题便解决了：

sys.stdout = io.TextIOWrapper(sys.stdout.buffer,encoding='utf8')

### 3. 总结：

实验过程中，发现有的查询语句速度过慢，通过添加约束条件，几个单词的增删就让速度增长了不少，从而大大提高了查询速度。为此，日后写MySQL语句时，不要写完了就以为结束了，还需要再回顾一下代码，看是否有更好的实现方法，以提高查询效率。同时设置查询条件时一定要对目标的相关信息考虑周全，有时感觉写sql语句略微像写小说。