**12/2/2019**

H8 F2019:

**Embedded SQL in Python**

Write a simple Python program, h8q1.py, to connect to the database toyu in dcm.uhcl.edu server to print out product information about a product line with number provided as the only command line argument. You must use PyMySQL driver for this assignment. For example:

…>python h8q1.py ITEC
Classes enrolled by students majoring in dept ITEC:
----------------------------------------------------
Catherine Lim: 0.
Larry Johnson: 2.

…>python h8q1.py CSCI

Classes enrolled by students majoring in dept CSCI:

----------------------------------------------------

David Hawk: 3.

Mary Hawk: 2.

Tony Hawk: 5.

…>python h8q1.py CINF

Classes enrolled by students majoring in dept CINF:

----------------------------------------------------

Lillian Johnson: 2.

Linda Johnson: 4.

Analysis:

Input: ITEC

Output:

Classes enrolled by students majoring in dept ITEC:
----------------------------------------------------
Catherine Lim: 0.
Larry Johnson: 2.

Dynamic content: from input or MySQL.

Not highlight: static content.

ITEC: from input

Catherine Lim: 0.
Larry Johnson: 2.

From MySQL





SELECT DISTINCT s.stuId, s.fname, s.lname, e.classsId

FROM Student s INNER JOIN Enroll e ON (s.stuId = e.stuId)

WHERE s.Major = 'ITEC';

SELECT DISTINCT s.stuId, s.fname, s.lname, e.classId

FROM Student s LEFT JOIN Enroll e ON (s.stuId = e.stuId)

WHERE s.Major = 'ITEC';

SELECT DISTINCT s.stuId, s.fname, s.lname, COUNT(e.classId) as n\_classes

FROM Student s LEFT JOIN Enroll e ON (s.stuId = e.stuId)

WHERE s.Major = 'ITEC'

GROUP BY s.stuId;

SELECT DISTINCT s.stuId, s.fname, s.lname, COUNT(e.classId) as n\_classes

FROM Student s LEFT JOIN Enroll e ON (s.stuId = e.stuId)

WHERE s.Major = 'CSCI' -- 'CSCI' replaced by input

GROUP BY s.stuId;

Solution:

from dbconfig import \*

import pymysql

import sys

# import sys object

# Get command line argument of product line id.

# For python h8q1.py ITEC

# sys.argv[1] = ‘ITEC’
# sys.argv[0] = ‘h8q1.py’

if len(sys.argv) > 1:

 dept = sys.argv[1]

else:

 dept = 'ITEC'

# Connect to big\_pvfc of the MySQL server and

# set up SQL statement to obtain product information

# in the product line: quite standard for config.py and config.ini

db = get\_mysql\_param()

cnx = pymysql.connect(user=db['user'], password=db['password'],

 host=db['host'],

 database=db['database'])

cursor = cnx.cursor()

# Parametrized query: %s: placeholder

query = '''

select concat(s.fname, ' ', s.lname) as student,

 count(e.classId) as num\_classes

from department d join student s on (d.deptCode = s.major)

 left join enroll e on (s.stuId = e.stuId)

where d.deptCode = %s

group by student

order by student

'''

# (dept): one variable, the same as dept

# (dept,): a list with one element, dept

cursor.execute(query,(dept,))

# Print product report.

print('Classes enrolled by students majoring in dept ' + dept + ':')

print('----------------------------------------------------')

for (student, num\_classes) in cursor:

 print(student + ': ' + str(num\_classes) + '.')

# housekeeping

cursor.close()

cnx.close()

<https://pymysql.readthedocs.io/en/latest/modules/cursors.html>:

**execute(query, args=None)**

Execute a query

|  |  |
| --- | --- |
| **Parameters:** | * **query** ([str](https://docs.python.org/3/library/stdtypes.html#str)) – Query to execute.
* **args** ([tuple](https://docs.python.org/3/library/stdtypes.html#tuple), [list](https://docs.python.org/3/library/stdtypes.html#list) or [dict](https://docs.python.org/3/library/stdtypes.html#dict)) – parameters used with query. (optional)
 |
| **Returns:** | Number of affected rows |
| **Return type:** | [int](https://docs.python.org/3/library/functions.html#int) |

If args is a list or tuple, %s can be used as a placeholder in the query. If args is a dict, %(name)s can be used as a placeholder in the query.

**System Catalog Examples:**

**SELECT** t.\*

**FROM** Information\_schema.**TABLES** t

**WHERE** t.TABLE\_SCHEMA = 'toyu';

*-- number of tables in toyu*

**SELECT** **COUNT**(t.**TABLE\_NAME**)

**FROM** Information\_schema.**TABLES** t

**WHERE** t.TABLE\_SCHEMA = 'toyu';

*-- all schema and their number of tables.*

**SELECT** t.TABLE\_SCHEMA, **COUNT**(t.**TABLE\_NAME**)

**FROM** Information\_schema.**TABLES** t

**GROUP** **BY** t.TABLE\_SCHEMA;