

Chap 12

Mysql Python 연동

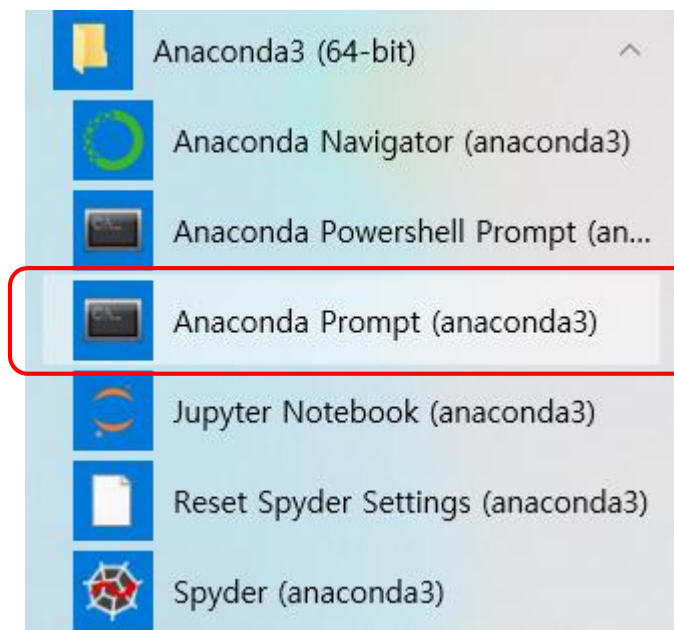
실습환경 : Anaconda3 2020.11, MySQL 8.0

1. Prepare

- Anaconda install
 - It includes Python and spyder (IDE)
 - <https://www.anaconda.com/products/individual>
Anaconda3 2020.11
- Install python-mysql connection module

1. Prepare

- Install python-mysql connection library



conda install pymysql

```
(base) C:\Users\WDKJ>conda install pymysql
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: D:\Wanaconda3

added / updated specs:
  - pymysql

The following packages will be downloaded:



| package       | build          | size   |
|---------------|----------------|--------|
| conda-4.10.1  | py38haa95532_1 | 2.9 MB |
| pymysql-1.0.2 | py38haa95532_1 | 77 KB  |
| Total:        |                | 3.0 MB |



The following NEW packages will be INSTALLED:

  pymysql          pkgs/main/win-64::pymysql-1.0.2-py38haa95532_1

The following packages will be UPDATED:

  conda             4.9.2-py38haa95532_0 --> 4.10.1-py38haa95532_1

Proceed ([y]/n)? y

Downloading and Extracting Packages
conda-4.10.1          | 2.9 MB | #####
pymysql-1.0.2        | 77 KB  | #####
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(base) C:\Users\WDKJ>
```

2. Mysql 연결 테스트

- Spyder 에서 프로그램 작성

```
# mysql - python connection test
```

```
import pymysql
```

```
#필요한 기본 DB 정보
```

```
host = "localhost" #접속할 db의 host명
```

```
user = "root"      #접속할 db의 user명
```

```
pw = "xxxx"       #접속할 db의 password
```

```
db = "my_db"      #접속할 db명
```

```
#DB에 접속
```

```
conn = pymysql.connect( host= host, user = user, password = pw, db = db)
```

```
#실제 사용 될 sql쿼리 문
```

```
sql = "SELECT * FROM emp LIMIT 10"
```

2. Mysql 연결 테스트

- Spyder 에서 프로그램 작성

#sql문 실행 / 데이터 받기

```
curs = conn.cursor()
```

```
curs.execute(sql)
```

```
data = curs.fetchall()
```

```
type(data)
```

```
data
```

```
data[0]
```

```
str(data[0][0])
```

sql 실행 결과 모두 가져오기

data 의 자료구조

data 의 내용출력

첫번째 행

첫번째 행의 첫번째 컬럼

#db 접속 종료

```
curs.close()
```

```
conn.close()
```

2. Mysql 연결 테스트

```
In [43]: type(data)
```

```
Out[43]: tuple
```

```
In [44]: data
```

```
Out[44]:
```

```
((Decimal('7369'),  
  'SMITH',  
  'CLERK',  
  Decimal('7902'),  
  datetime.date(1980, 12, 17),  
  Decimal('800.00'),  
  None,  
  Decimal('20')),  
(Decimal('7499'),  
  'ALLEN',  
  'SALESMAN',  
  Decimal('7698'),  
  datetime.date(1981, 2, 20),  
  Decimal('1600.00'),  
  Decimal('300.00'),  
  Decimal('30'),  
  (Decimal('7521'),
```

History

IPython console

2. Mysql 연결 테스트

```
In [45]: data[0] # 첫번째행
```

```
Out[45]:  
(Decimal('7369'),  
 'SMITH',  
 'CLERK',  
 Decimal('7902'),  
 datetime.date(1980, 12, 17),  
 Decimal('800.00'),  
 None,  
 Decimal('20'))
```

```
In [46]: str(data[0][0]) # 첫번째행의 첫번째 컬럼
```

```
Out[46]: '7369'
```

3. 실행결과를 하나씩 가져오기

... (slide 4와 동일)

#실제 사용 될 sql쿼리 문

```
sql = "SELECT * FROM emp LIMIT 10"
```

#sql문 실행 / 데이터 받기

```
curs = conn.cursor()
```

```
curs.execute(sql)
```

```
row = curs.fetchone()
```

```
while(row):
```

```
    print(row)
```

```
    row = curs.fetchone()
```

#db 접속 종료

```
curs.close()
```

```
conn.close()
```

cursor
→

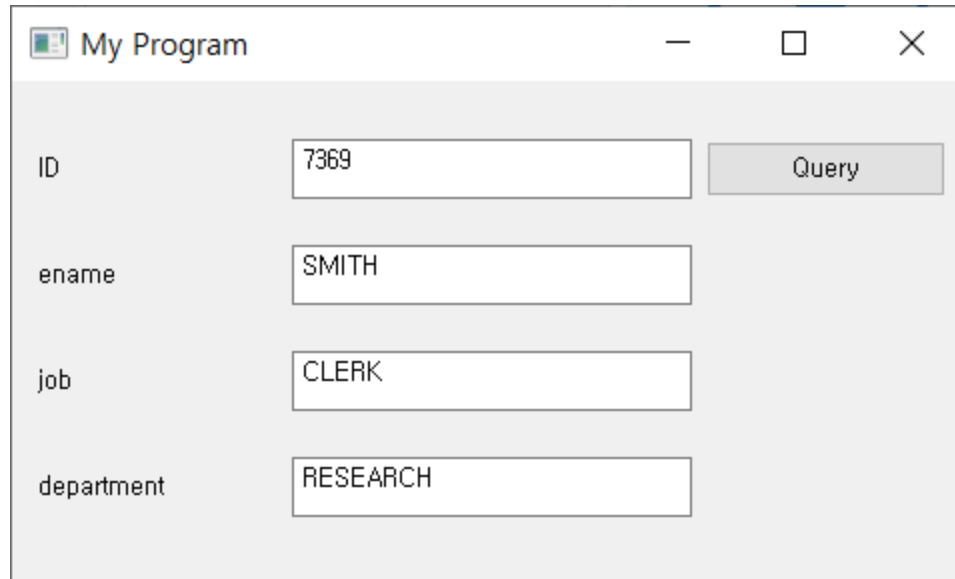
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	1980-12-17	800.00	NULL	20
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30
7566	JONES	MANAGER	7839	1981-04-02	2975.00	NULL	20
7654	MARTIN	SALESMAN	7698	1981-08-28	1250.00	1400.00	30
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00	NULL	30
7782	CLARK	MANAGER	7839	1981-06-09	2450.00	NULL	10
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00	NULL	20
7839	KING	PRESIDENT	NULL	1981-11-17	5000.00	NULL	10
7844	TURNER	SALESMAN	7698	1981-08-08	1500.00	0.00	30

3. 실행결과를 하나씩 가져오기

```
(Decimal('7369'), 'SMITH', 'CLERK', Decimal('7902'), datetime.date(1980, 12, 17), Decimal('800.00'), None, Decimal('20'))  
(Decimal('7499'), 'ALLEN', 'SALESMAN', Decimal('7698'),  
datetime.date(1981, 2, 20), Decimal('1600.00'), Decimal('300.00'),  
Decimal('30'))  
(Decimal('7521'), 'WARD', 'SALESMAN', Decimal('7698'),  
datetime.date(1981, 2, 22), Decimal('1250.00'), Decimal('500.00'),  
Decimal('30'))  
(Decimal('7566'), 'JONES', 'MANAGER', Decimal('7839'),  
datetime.date(1981, 4, 2), Decimal('2975.00'), None, Decimal('20'))  
(Decimal('7654'), 'MARTIN', 'SALESMAN', Decimal('7698'),
```

4. PyQt 를 이용한 GUI 프로그램

- 사원번호를 입력하면 사원의 이름, 담당업무, 부서명을 윈도우에 보이는 프로그램



The image shows a window titled "My Program" with standard Windows window controls (minimize, maximize, close). Inside the window, there is a form with four labels and corresponding text input fields:

- ID: 7369
- ename: SMITH
- job: CLERK
- department: RESEARCH

To the right of the ID input field is a button labeled "Query".

```
import sys
import pymysql
from PyQt5.QtWidgets import *

def connectDB():
    #필요한 기본 DB 정보
    host = "localhost" #접속할 db의 host명
    user = "root"       #접속할 db의 user명
    pw = "*****"       #접속할 db의 password
    db = "my_db"        #접속할 db의 이름

    #DB에 접속
    conn = pymysql.connect( host= host, user = user, password = pw, db = db)
    return(conn)

def disconnectDB(conn):
    conn.close()
```

```
class MyApp(QWidget):
```

```
    def __init__(self):
```

```
        super().__init__()
```

```
        self.initUI()
```

```
# UI 디자인 함수
```

```
def initUI(self):
```

```
    label1 = QLabel('ID')
```

```
    label2 = QLabel('ename')
```

```
    label3 = QLabel('job')
```

```
    label4 = QLabel('department')
```

global 변수

```
self.text_id = QTextEdit()
```

```
self.text_id.setFixedWidth(200)
```

```
self.text_id.setFixedHeight(30)
```

```
btn_1 = QPushButton('Query')
```

```
btn_1.clicked.connect(self.btn_1_clicked)
```

사용자 정의
함수

클릭하면 이 함수 실행

```
self.text_ename = QTextEdit()
```

```
self.text_ename.setFixedWidth(200)
```

```
self.text_ename.setFixedHeight(30)
```

```
self.text_job = QTextEdit()
```

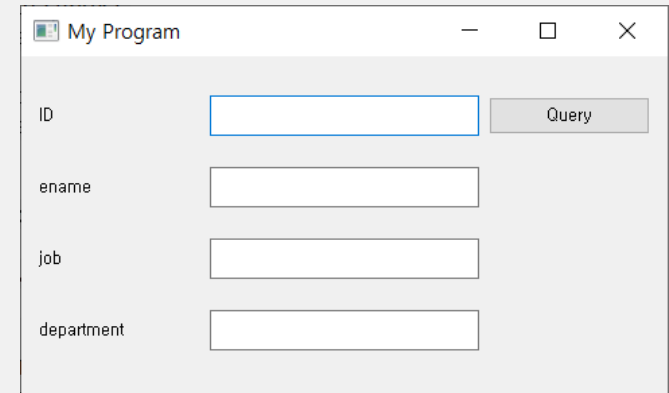
```
self.text_job.setFixedWidth(200)
```

```
self.text_job.setFixedHeight(30)
```

```
self.text_dept = QTextEdit()
```

```
self.text_dept.setFixedWidth(200)
```

```
self.text_dept.setFixedHeight(30)
```



```

gbox = QGridLayout() 배치-방식
gbox.addWidget(label1, 0, 0)
gbox.addWidget(self.text_id, 0, 1)
gbox.addWidget(btn_1, 0, 2)
gbox.addWidget(label2, 1, 0)
gbox.addWidget(self.text_ename, 1, 1)
gbox.addWidget(label3, 2, 0)
gbox.addWidget(self.text_job, 2, 1)
gbox.addWidget(label4, 3, 0)
gbox.addWidget(self.text_dept, 3, 1)

```

```

self.setLayout(gbox)
self.setWindowTitle('My Program')
self.setGeometry(300,300, 480,250)
self.show()

```

0.0	0.1	0.2
ID	<input type="text"/>	Query
ename	<input type="text"/>	
job	<input type="text"/>	
department	<input type="text"/>	

버튼 클릭시 처리 함수

```
def btn_1_clicked(self):
```

```
    empno = self.text_id.toPlainText()
```

* text-id에 넣은 값은 가져옴

sql쿼리 문

```
    sql = "SELECT ename, job, dname \
          FROM emp e, dept d \
          where e.deptno = d.deptno \
          and empno = " + empno
```

* chame = "'" + ename + "',"

```
    conn = connectDB()
```

```
    curs = conn.cursor()
```

```
    curs.execute(sql)
```

```
    result = curs.fetchone() # sql 실행 결과 가져오기
```

```
    self.text_ename.setText(result[0])
```

```
    self.text_job.setText(result[1])
```

```
    self.text_dept.setText(result[2])
```

```
# QMessageBox.about(self, "message", "qqq")
```

```
curs.close()
```

```
disconnect(conn)
```

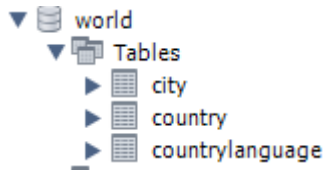
END Class

프로그램 실행

```
if (__name__ == '__main__'):  
    app = QApplication(sys.argv)  
    ex = MyApp()  
    sys.exit(app.exec_())
```

[과제]

- 1. Mysql 의 my_db 와 연동하여 다음 내용을 실행하는 python 프로그램을 작성하고 실행 결과를 보이시오 (하나의 프로그램으로 작성)
 - 사원번호가 7521인 **사원의 이름을** 출력한다
 - 이름이 SCOTT 인 사원의 **부서이름**을 출력한다
 - 담당업무가 SALESMAN 인 **모든사원의 이름을** 출력한다.
 - **모든 부서의 이름을** 출력한다.
- 2. Mysql 의 world 데이터베이스와 연동하여 다음과 같이 국가명을 입력하면 해당 국가의 정보를 출력하는 윈도우 프로그램을 작성하시오
- 매치되는 국가가 없으면 모든 항목에 공백 출력



Country Info	
Country name	<input type="text"/>
Continent	<input type="text"/>
Population	<input type="text"/>
GNP	<input type="text"/>
Capital city	<input type="text"/>
Language	<input type="text"/>

Query