Python 실습

Database Programming

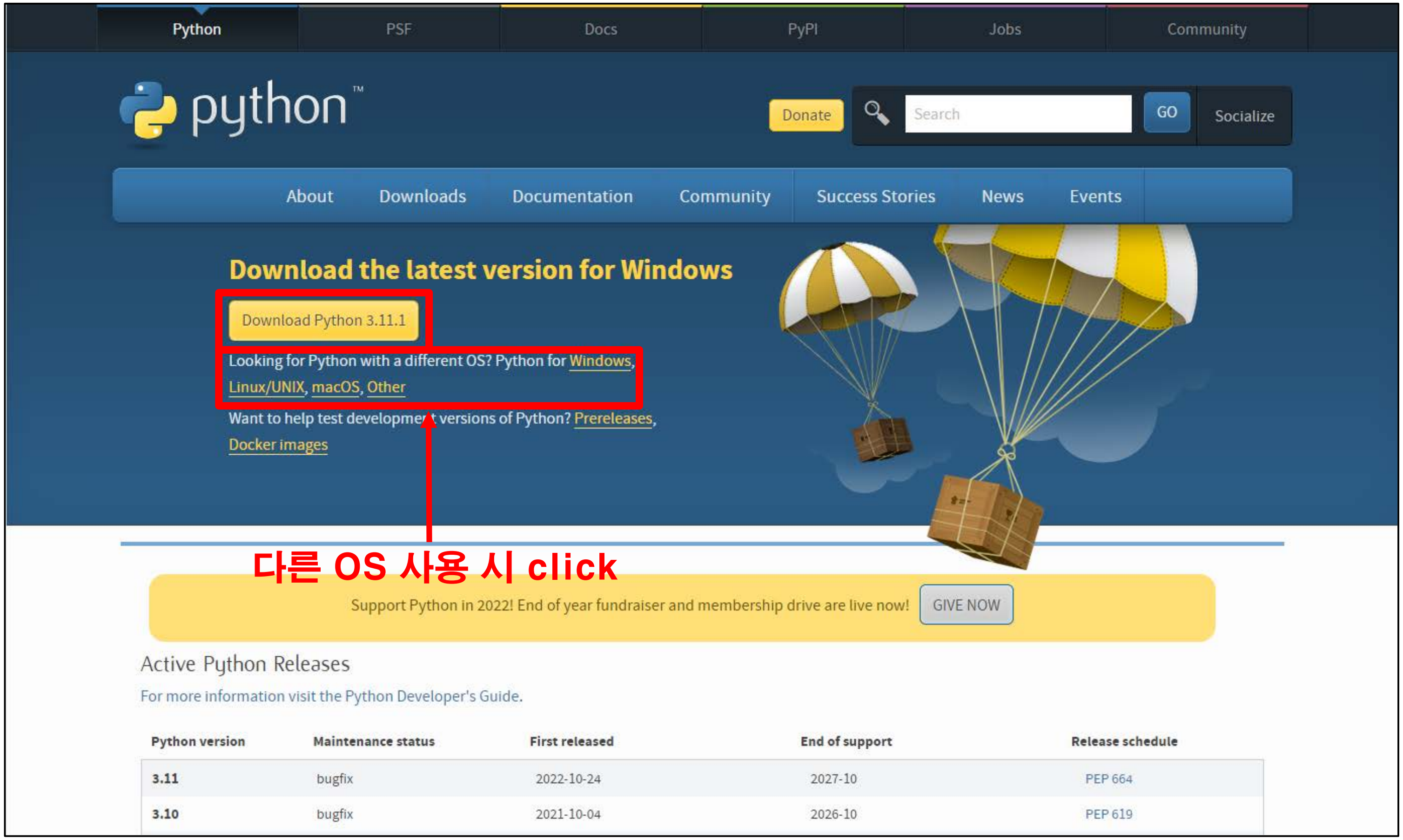
실습 순서

- Python 설치
- MySQL Connector 드라이버 설치
- Python과 데이터베이스 예제 실습

Python 설치(1/7)

1) https://www.python.org/downloads 접속

2) "Download Python 3.11.1" 클릭 (다른 OS 사용 시, "Linux/UNIX, macOS, Other" 을 선택하여 클릭)



Download the latest version for Windows

Download Python 3.11.1

Looking for Python with a different OS? Python for Windows, Linux/UNIX, macOS, Other

Want to help test development versions of Python? Prereleases, Docker images

다른 OS 사용 시 click

Support Python in 2022! End of year fundraiser and membership drive are live now!

Active Python Releases

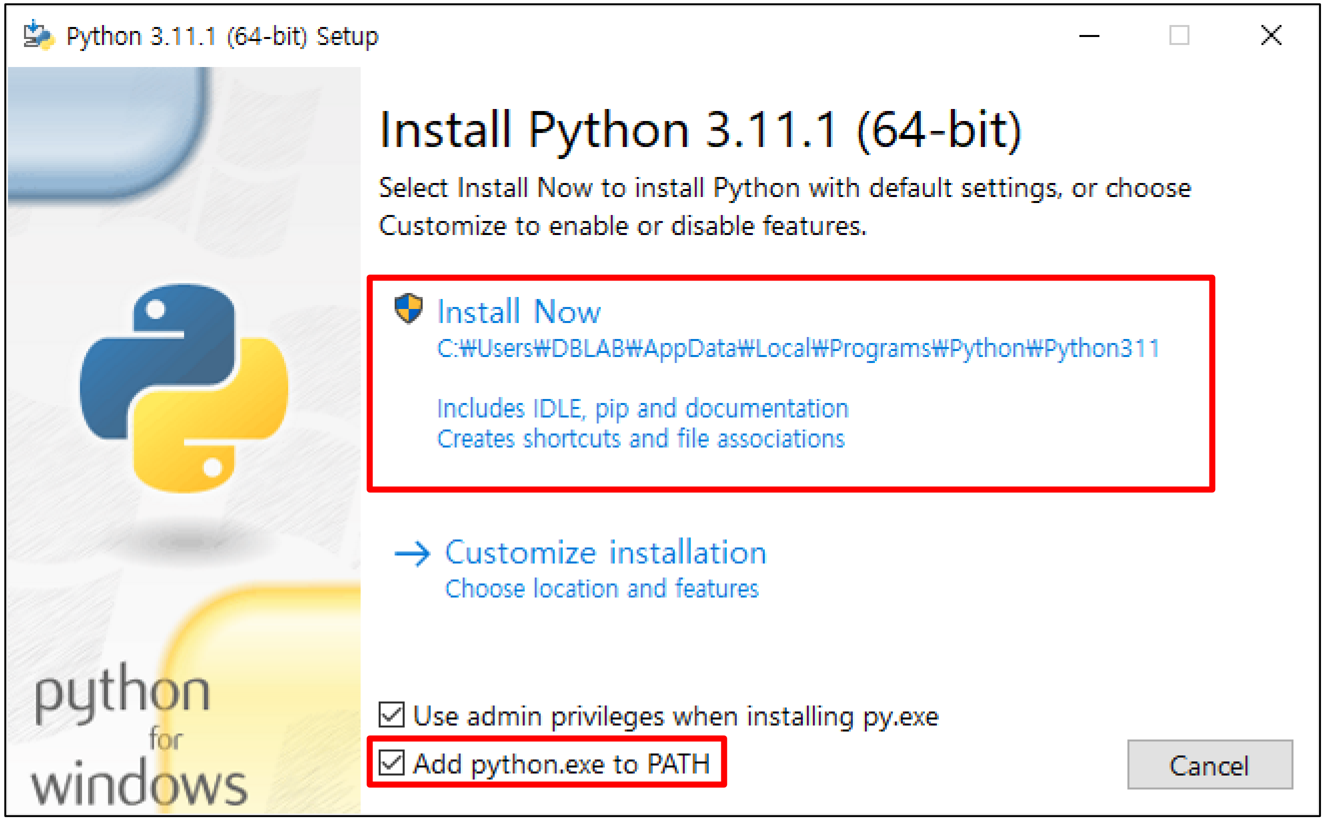
For more information visit the Python Developer's Guide.

Python version	Maintenance status	First released	End of support	Release schedule
3.11	bugfix	2022-10-24	2027-10	PEP 664
3.10	bugfix	2021-10-04	2026-10	PEP 619

Python 설치(2/7)

3) 다운로드 받은 설치 파일 실행

4) "Add python.exe to PATH" 체크 (환경 변수 자동 설정) → "Install Now" 클릭



Python 3.11.1 (64-bit) Setup

Install Python 3.11.1 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:₩Users₩DBLAB₩AppData₩Local₩Programs₩Python₩Python311

Includes IDLE, pip and documentation
Creates shortcuts and file associations

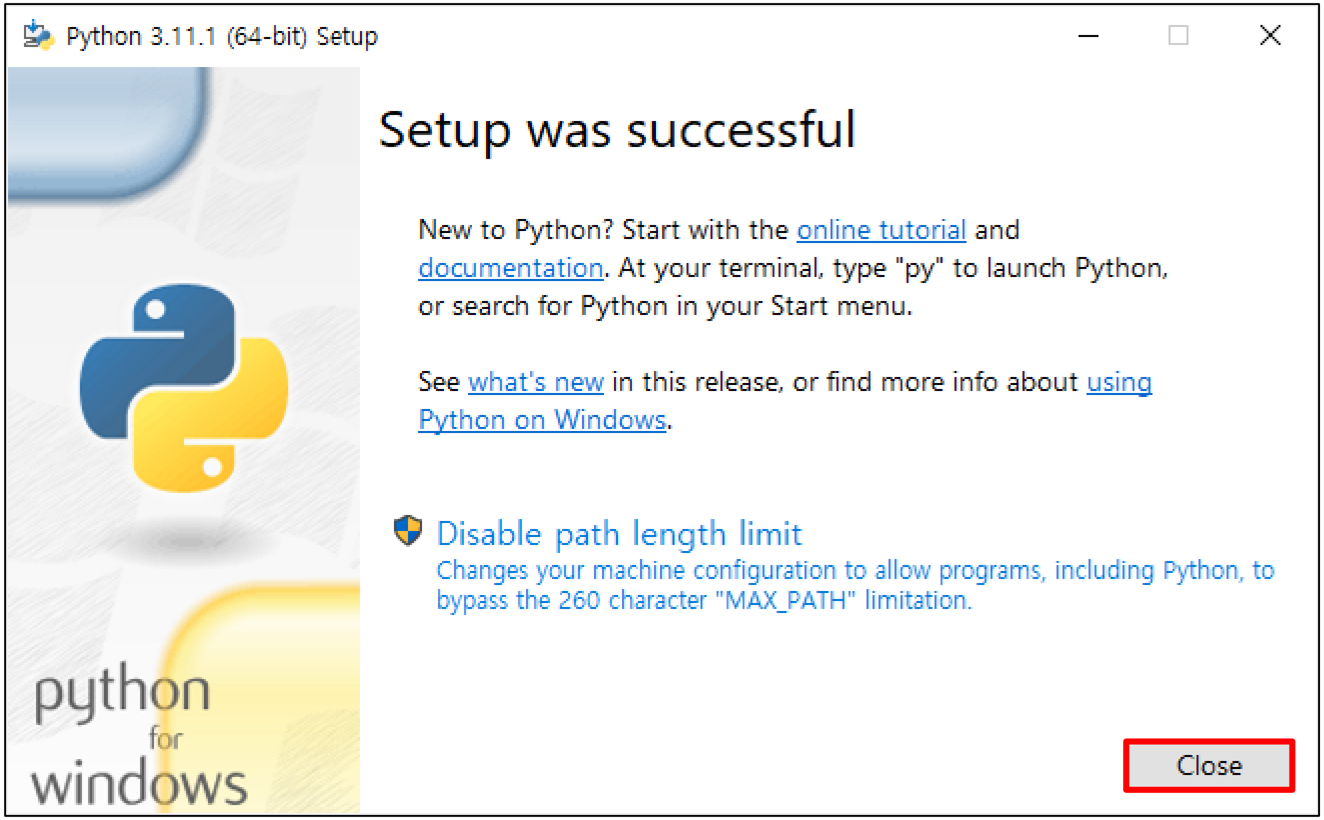
→ Customize installation
Choose location and features

☑ Use admin privileges when installing py.exe

☑ Add python.exe to PATH

Python 설치(3/7)

5) Close 클릭



Setup was successful

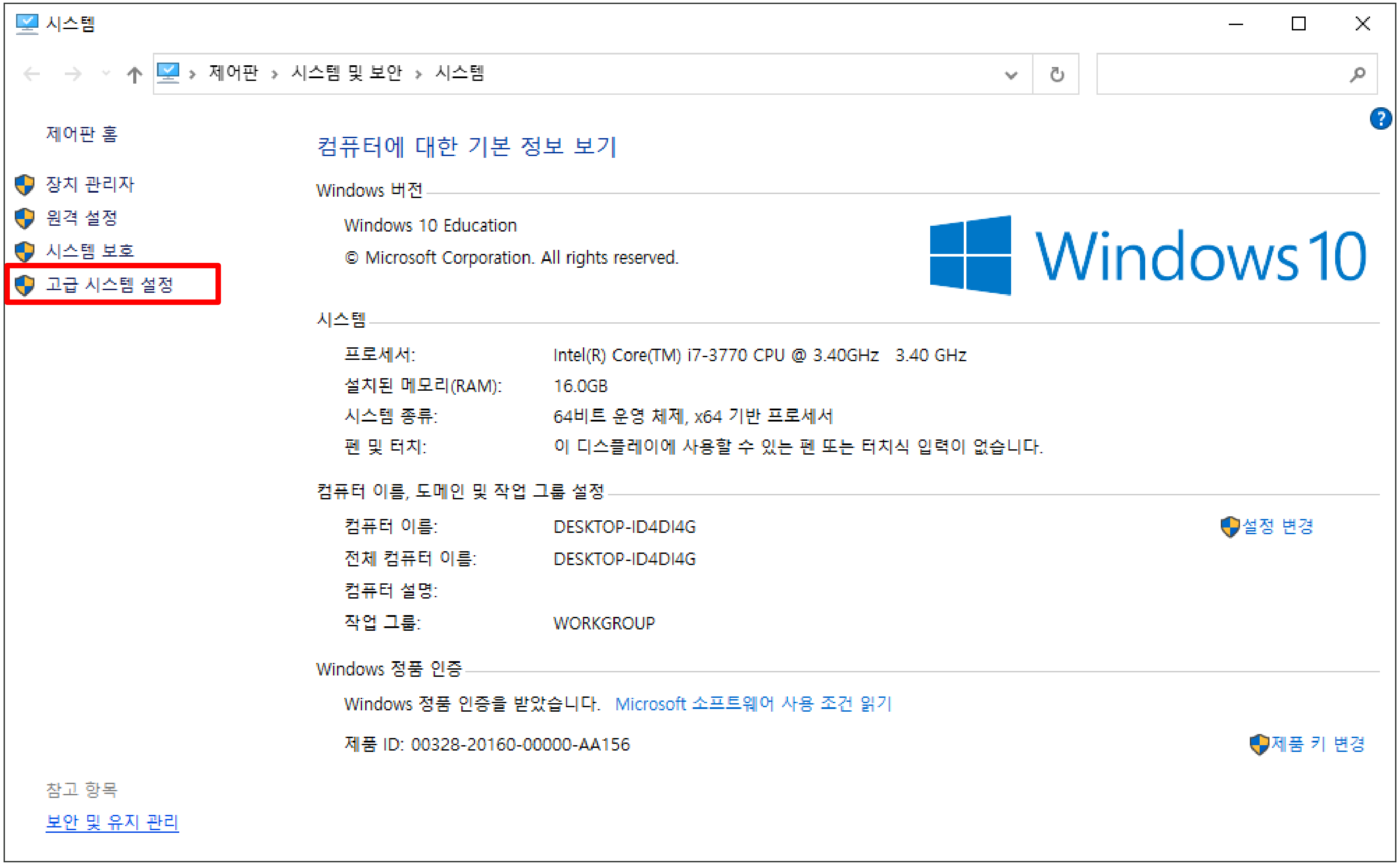
New to Python? Start with the online tutorial and documentation. At your terminal, type "py" to launch Python, or search for Python in your Start menu.

See what's new in this release, or find more info about using Python on Windows.

Disable path length limit
Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

Python 설치(4/7)

6) 내 PC 우클릭 → 속성 → 고급 시스템 설정 클릭



컴퓨터에 대한 기본 정보 보기

Windows 버전

Windows 10 Education

© Microsoft Corporation. All rights reserved.

시스템

프로세서:	Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz 3.40 GHz
설치된 메모리(RAM):	16.0GB
시스템 종류:	64비트 운영 체제, x64 기반 프로세서
펜 및 터치:	이 디스플레이에 사용할 수 있는 펜 또는 터치식 입력이 없습니다.

컴퓨터 이름, 도메인 및 작업 그룹 설정

컴퓨터 이름:	DESKTOP-ID4DI4G
전체 컴퓨터 이름:	DESKTOP-ID4DI4G
컴퓨터 설명:	
작업 그룹:	WORKGROUP

설정 변경

Windows 정품 인증

Windows 정품 인증을 받았습니다. Microsoft 소프트웨어 사용 조건 읽기

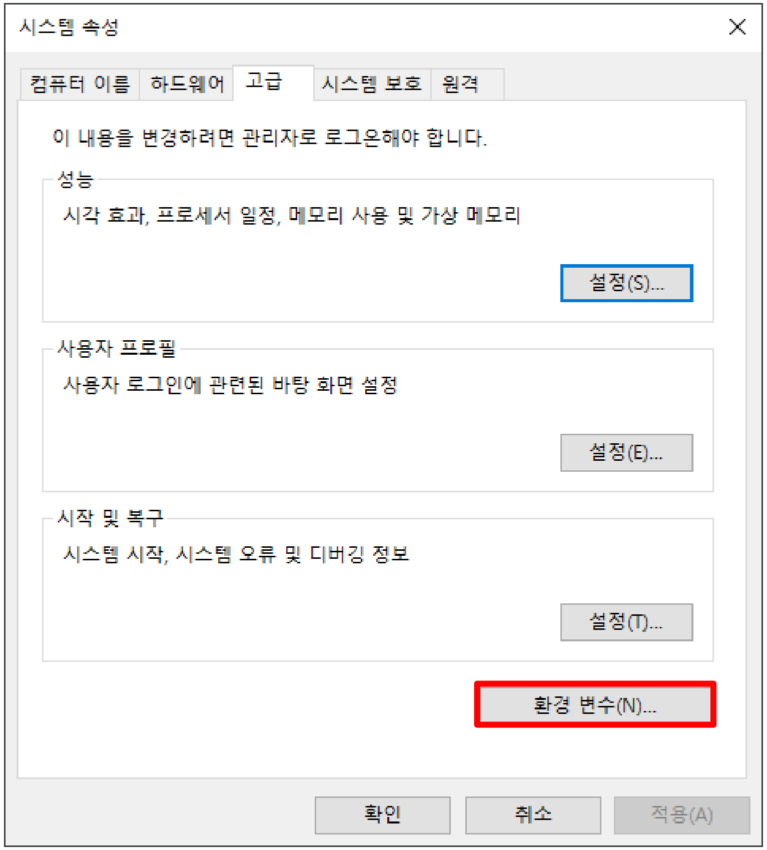
제품 ID: 00328-20160-00000-AA156

제품 키 변경

Python 설치(5/7)

7) 환경 변수 클릭

8) 시스템 변수 → Path 선택 → 편집 클릭

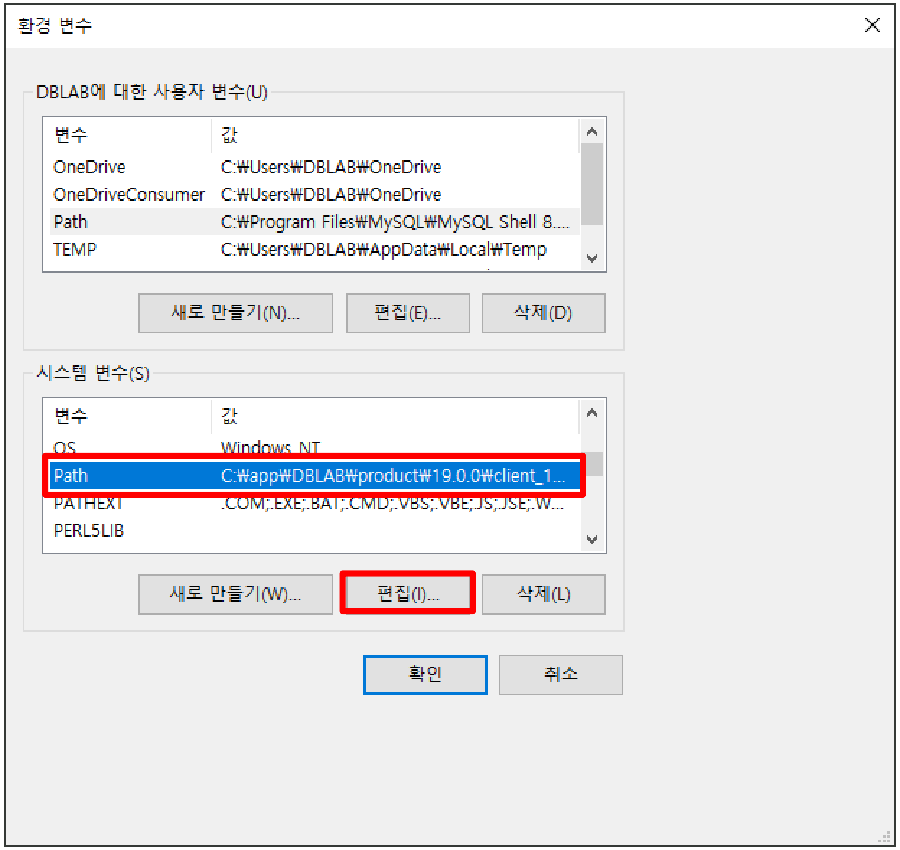


이 내용을 변경하려면 관리자로 로그온해야 합니다.

시각 효과, 프로세서 일정, 메모리 사용 및 가상 메모리

사용자 로그인에 관련된 바탕 화면 설정

시스템 시작, 시스템 오류 및 디버깅 정보



DBLAB에 대한 사용자 변수(U)

변수	값
OneDrive	C:₩Users₩DBLAB₩OneDrive
OneDriveConsumer	C:₩Users₩DBLAB₩OneDrive
Path	C:₩Program Files₩MySQL₩MySQL Shell 8....
TEMP	C:₩Users₩DBLAB₩AppData₩Local₩Temp

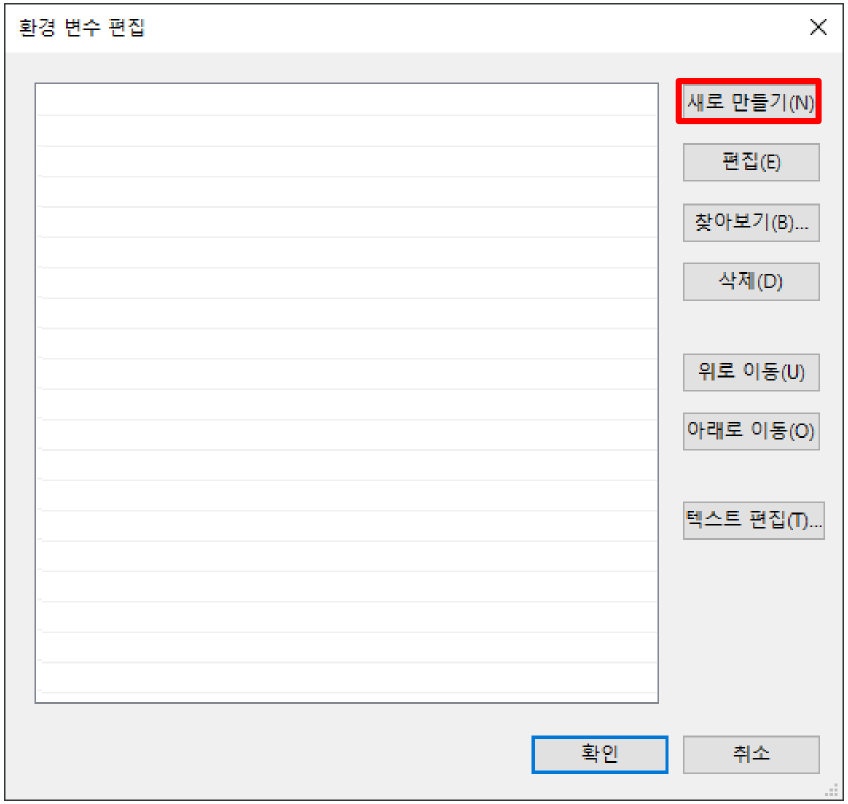
시스템 변수(S)

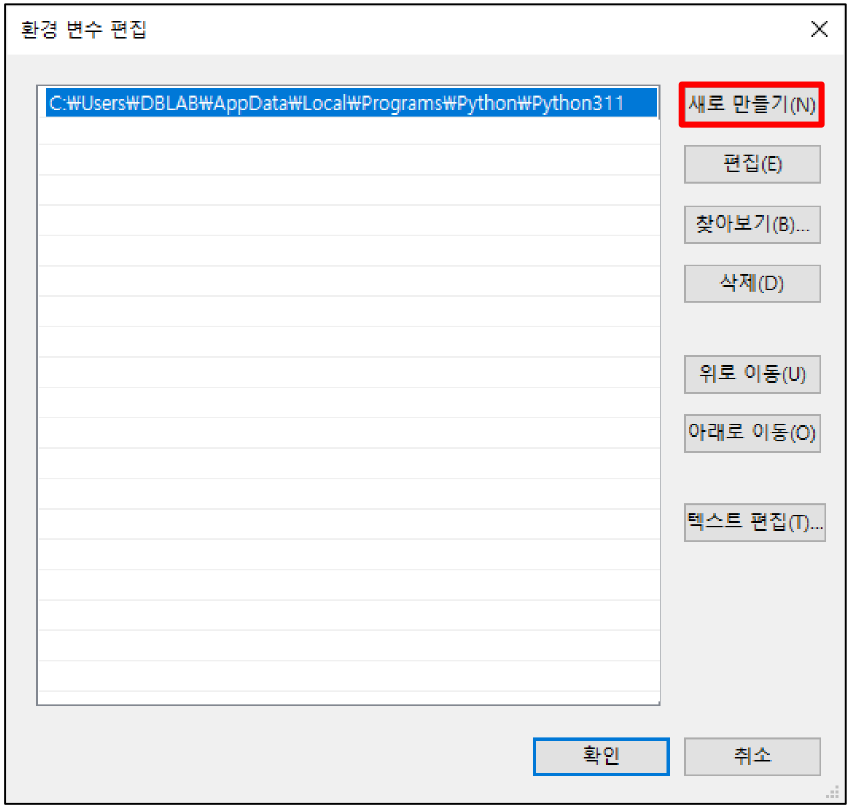
변수	값
OS	Windows_NT
Path	C:₩app₩DBLAB₩product₩19.0.0₩client_1...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.W...
PERL5LIB	

Python 설치(6/7)

9) 새로 만들기 클릭

10) "C:\Users\DBLAB\AppData\Local\Programs\Python\Python311" 입력 → 새로 만들기 클릭

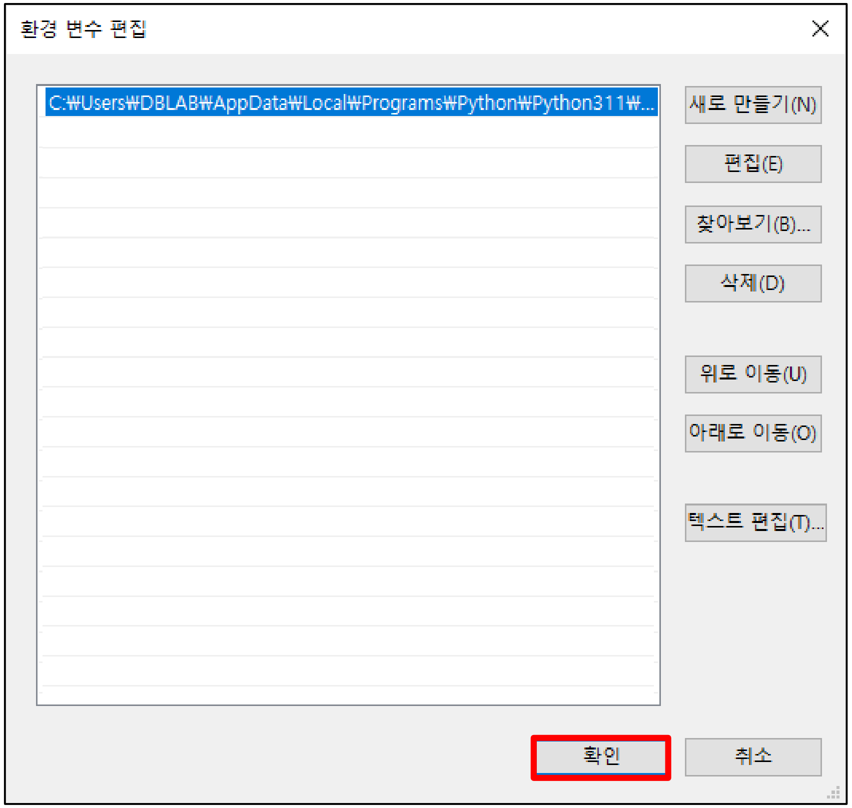




C:₩Users₩DBLAB₩AppData₩Local₩Programs₩Python₩Python311

Python 설치(7/7)

11)"C:\Users\DBLAB\AppData\Local\Programs\Python\Python311\Scripts"입력 → 확인 클릭

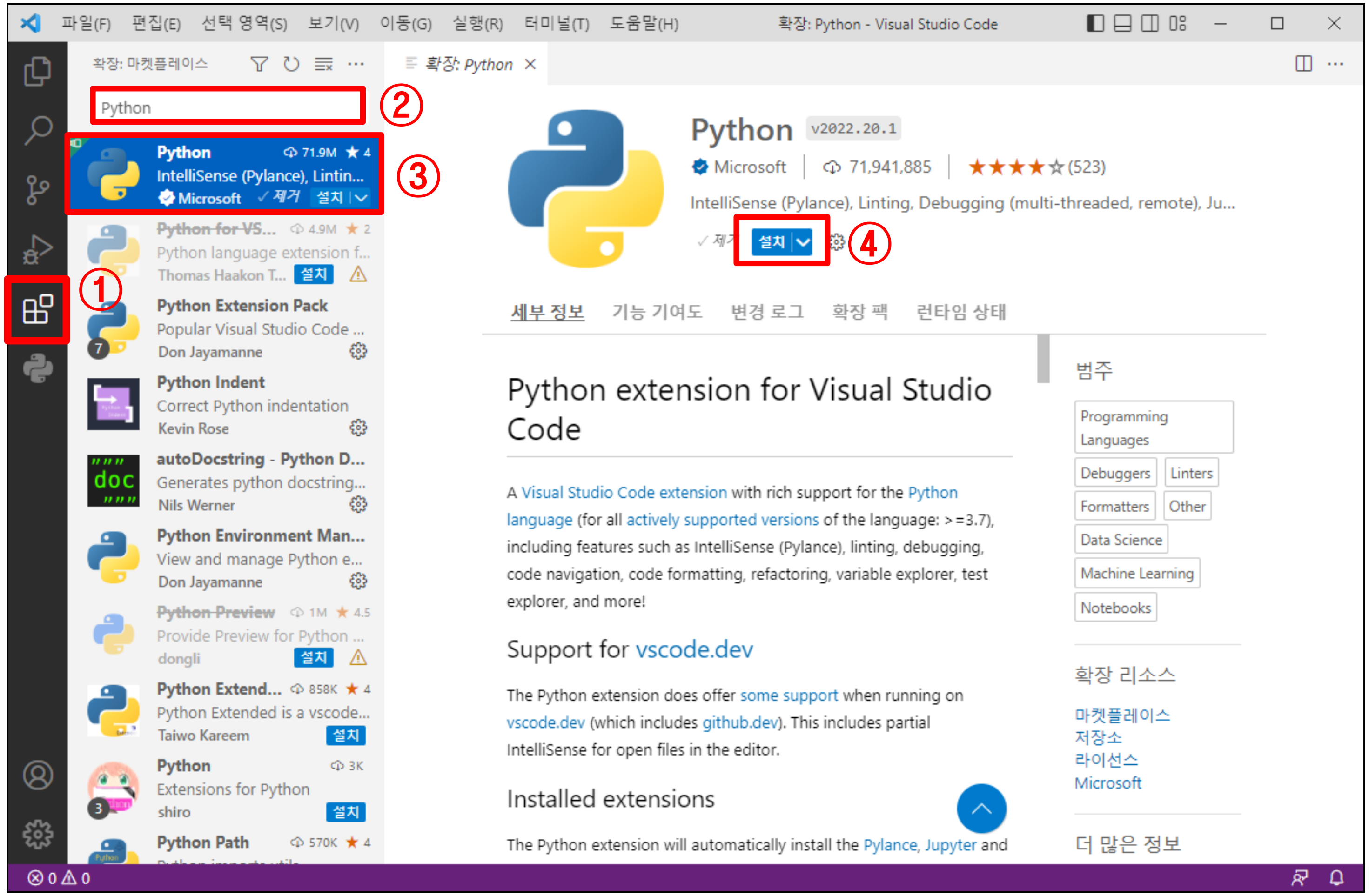


C:₩Users₩DBLAB₩AppData₩Local₩Programs₩Python₩Python311₩...

Python 실행

1) Visual Studio Code 실행

2) "확장" 클릭 → 검색창에 "python" 입력 → "Python" 선택 후 "설치" 클릭


파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) 실행(R) 터미널(T) 도움말(H)
확장: Python - Visual Studio Code
확장: 마켓플레이스
확장: Python
Python
②
Python 71.9M ★ 4
IntelliSense (Pylance), Lintin...
Microsoft 제거 설치
③
~~Python for VS...~~ 4.9M ★ 2
Python language extension f...
Thomas Haakon T... 설치
①
Python Extension Pack
Popular Visual Studio Code ...
Don Jayamanne
Python Indent
Correct Python indentation
Kevin Rose
autoDocstring - Python D...
Generates python docstring...
Nils Werner
Python Environment Man...
View and manage Python e...
Don Jayamanne
~~Python Preview~~ 1M ★ 4.5
Provide Preview for Python ...
dongli 설치
Python Extend... 858K ★ 4
Python Extended is a vscode...
Taiwo Kareem 설치
Python 3K
Extensions for Python
shiro 설치
Python Path 570K ★ 4
Python v2022.20.1
Microsoft | 71,941,885 | ★★★★☆ (523)
IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote), Ju...
제거 설치
④
세부 정보 기능 기여도 변경 로그 확장 팩 런타임 상태
범주
Programming Languages
Debuggers
Linters
Formatters
Other
Data Science
Machine Learning
Notebooks
확장 리소스
마켓플레이스
저장소
라이선스
Microsoft
더 많은 정보
⊗ 0 ⚠ 0

Python extension for Visual Studio Code

A Visual Studio Code extension with rich support for the Python language (for all actively supported versions of the language: >=3.7), including features such as IntelliSense (Pylance), linting, debugging, code navigation, code formatting, refactoring, variable explorer, test explorer, and more!

Support for vscode.dev

The Python extension does offer some support when running on vscode.dev (which includes github.dev). This includes partial IntelliSense for open files in the editor.

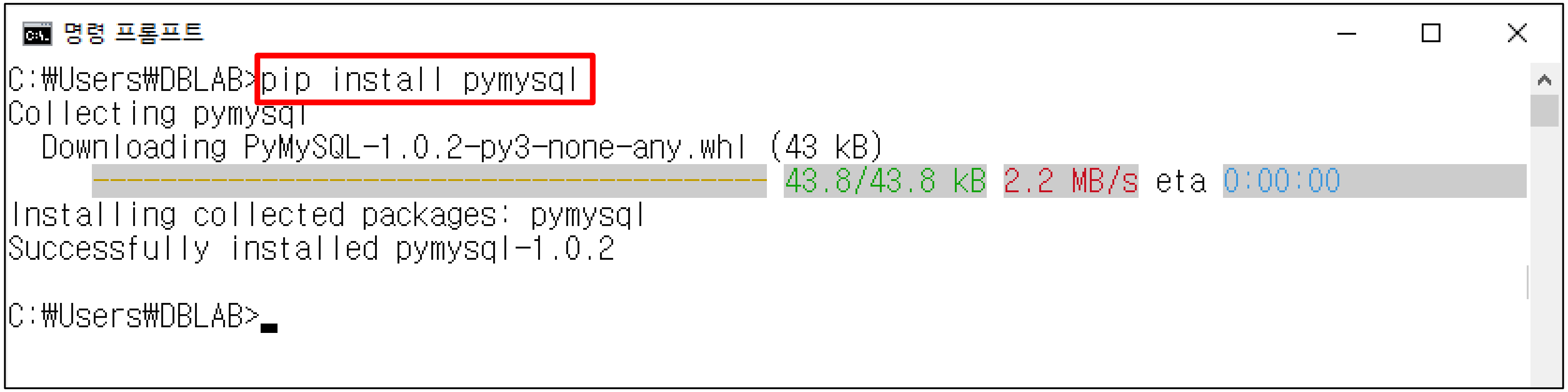
Installed extensions

The Python extension will automatically install the Pylance, Jupyter and

MySQL Connector 드라이버 설치(1/2)

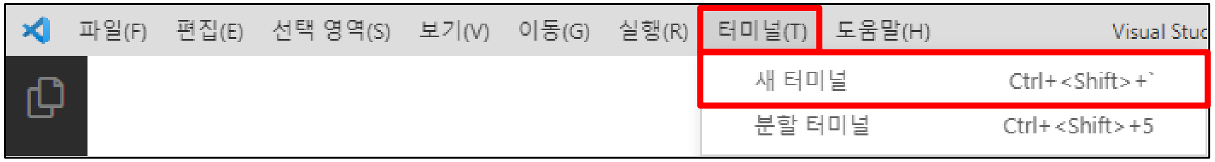
1) 명령 프롬프트 실행 (또는 Visual Studio Code 메뉴-터미널-새 터미널 실행)

2) "pip install pymysql" 입력



```
C:₩Users₩DBLAB>pip install pymysql
Collecting pymysql
  Downloading PyMySQL-1.0.2-py3-none-any.whl (43 kB)
     ---------------------------------------- 43.8/43.8 kB 2.2 MB/s eta 0:00:00
Installing collected packages: pymysql
Successfully installed pymysql-1.0.2

C:₩Users₩DBLAB>
```

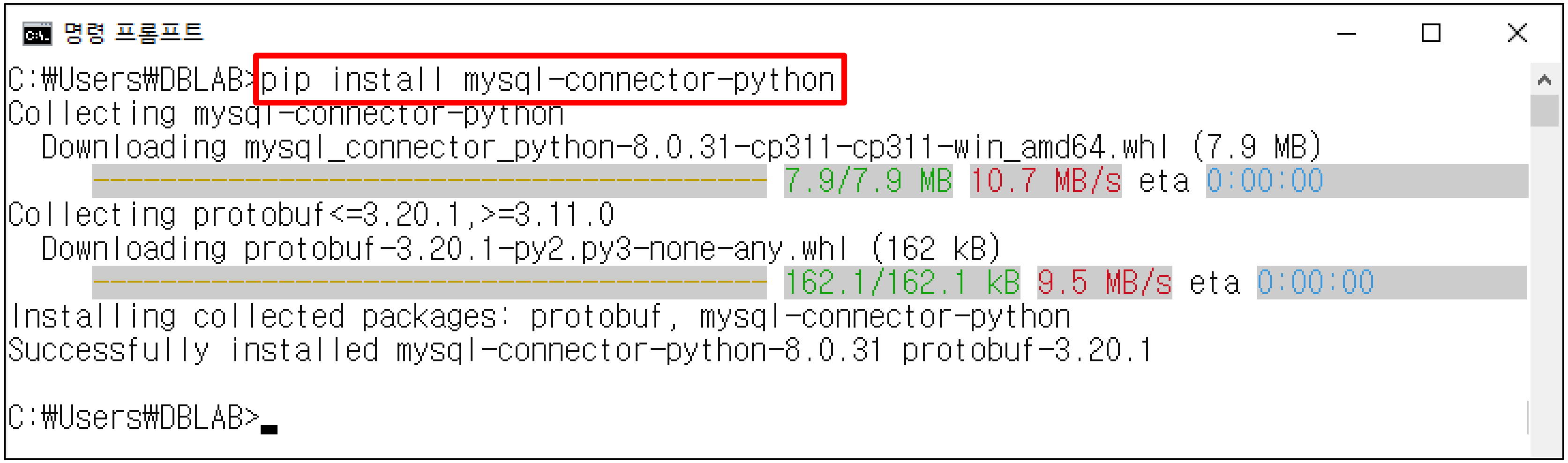
<명령 프롬프트 화면>



<Visual Studio Code 화면>

MySQL Connector 드라이버 설치(2/2)

3) "pip install mysql-connector-python" 입력
(pip 버전 오류가 발생할 경우, "python -m pip install --upgrade pip" 입력 후 MySQL Connector 드라이버 설치 진행)

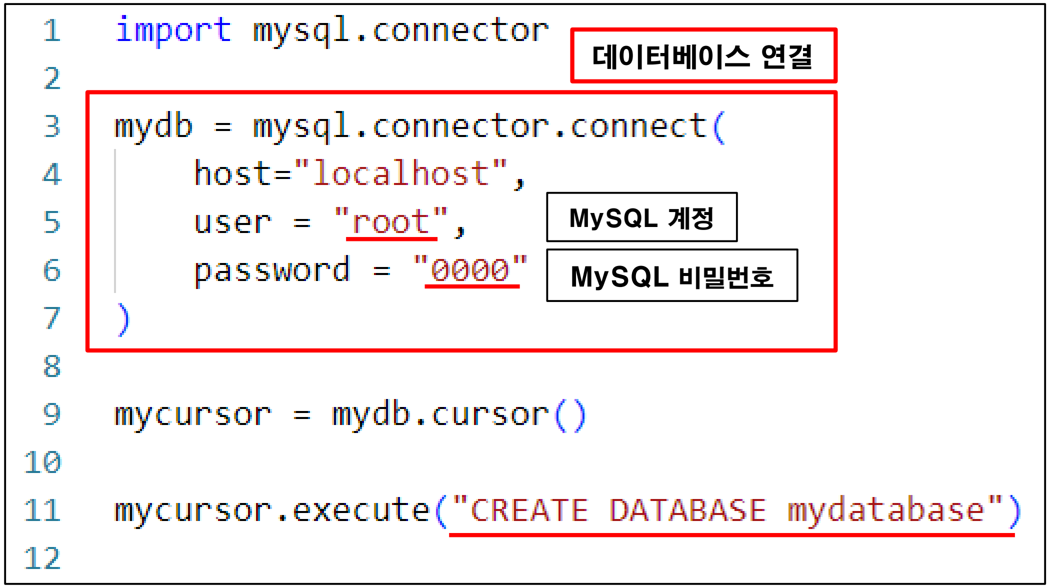


```
C:₩Users₩DBLAB>pip install mysql-connector-python
Collecting mysql-connector-python
  Downloading mysql_connector_python-8.0.31-cp311-cp311-win_amd64.whl (7.9 MB)
     ---------------------------------------- 7.9/7.9 MB 10.7 MB/s eta 0:00:00
Collecting protobuf<=3.20.1,>=3.11.0
  Downloading protobuf-3.20.1-py2.py3-none-any.whl (162 kB)
     ---------------------------------------- 162.1/162.1 kB 9.5 MB/s eta 0:00:00
Installing collected packages: protobuf, mysql-connector-python
Successfully installed mysql-connector-python-8.0.31 protobuf-3.20.1

C:₩Users₩DBLAB>
```

<명령 프롬프트 화면>

데이터베이스 생성

- MySQL의 계정 및 비밀번호로 데이터베이스 연결
- 데이터베이스 생성 예시

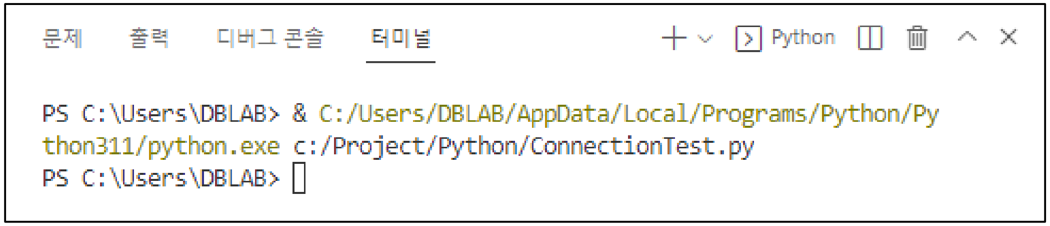


```python
import mysql.connector

mydb = mysql.connector.connect(
    host="localhost",
    user = "root",
    password = "0000"
)

mycursor = mydb.cursor()

mycursor.execute("CREATE DATABASE mydatabase")

```

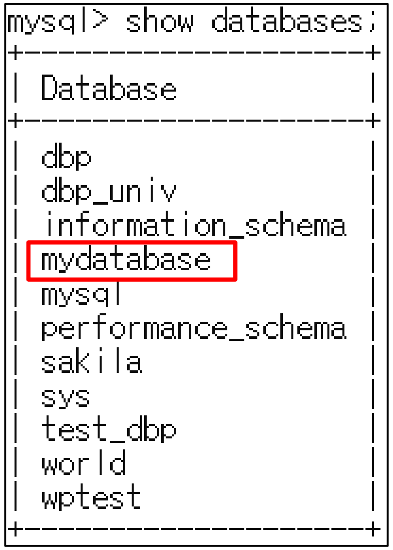
데이터베이스 연결

MySQL 계정

MySQL 비밀번호



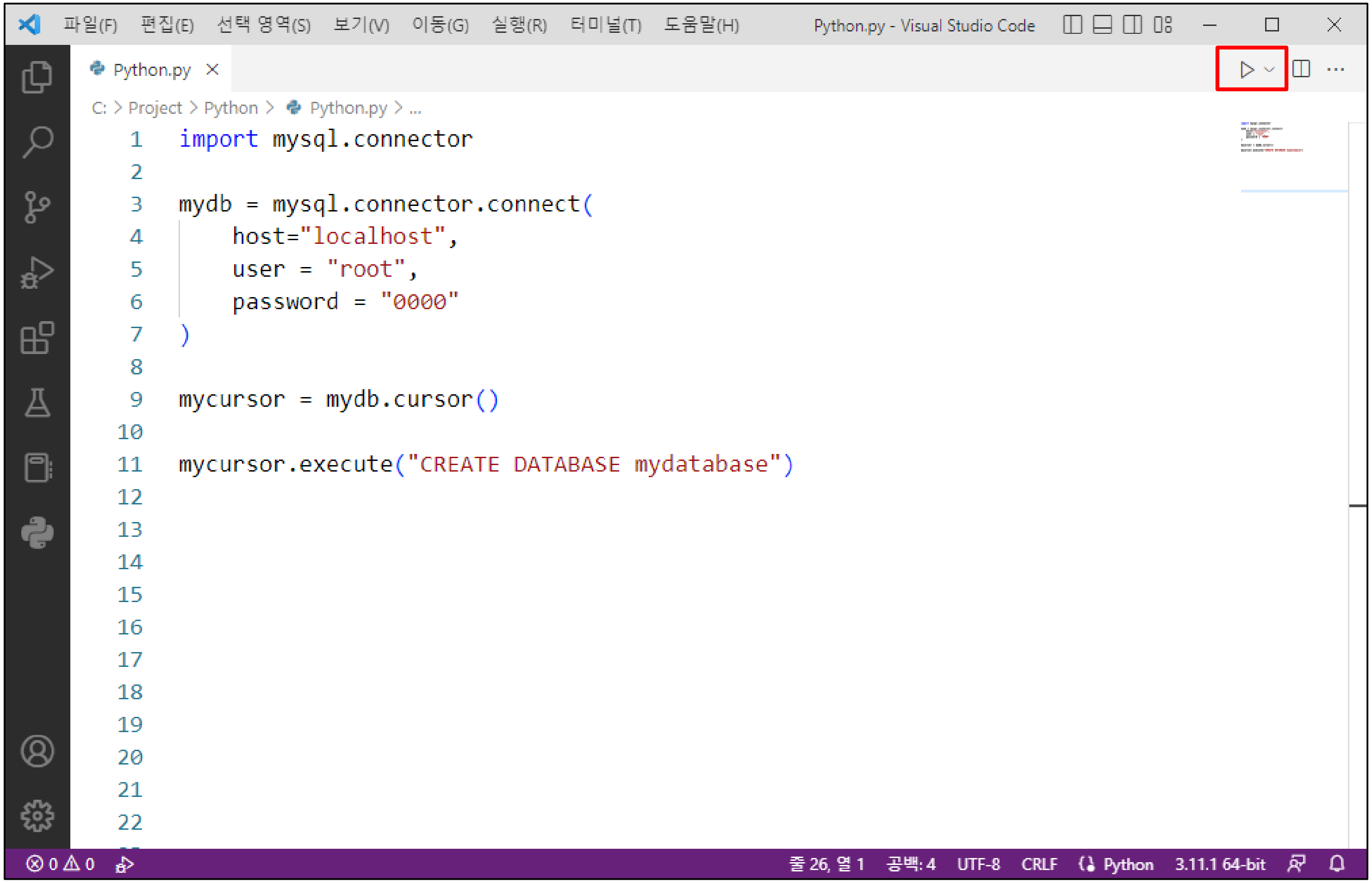
```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/Python/Py
thon311/python.exe c:/Project/Python/ConnectionTest.py
PS C:\Users\DBLAB> 
```



```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| dbp                |
| dbp_univ           |
| information_schema |
| mydatabase         |
| mysql              |
| performance_schema |
| sakila             |
| sys                |
| test_dbp           |
| world              |
| wptest             |
+--------------------+
```

Visual Studio Code – Python 실행

- 파일 실행 버튼을 클릭하여 작성한 코드 실행

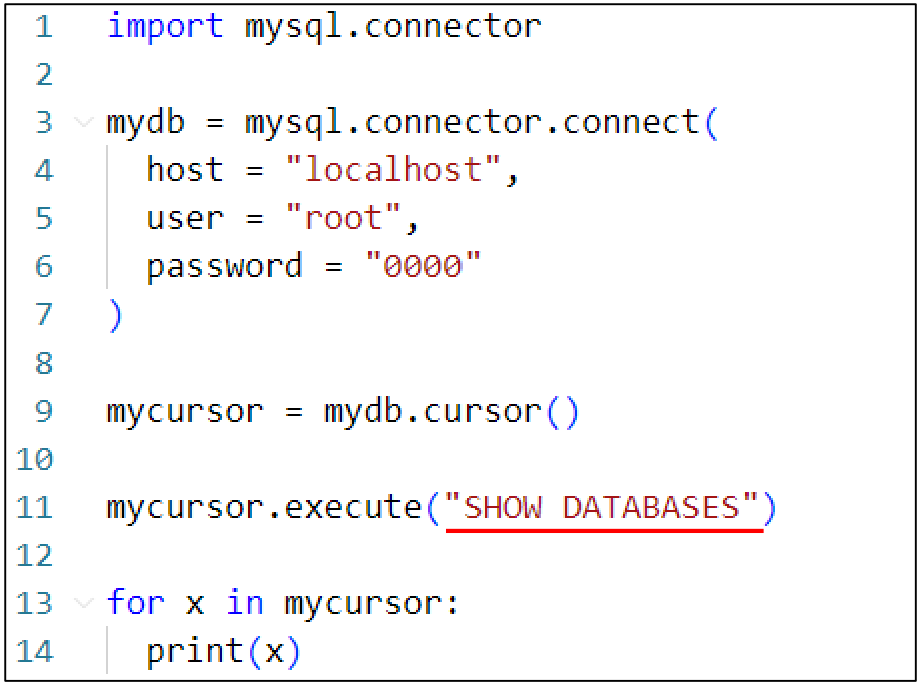


파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) 실행(R) 터미널(T) 도움말(H)
Python.py - Visual Studio Code
Python.py
C: > Project > Python > Python.py > ...
⊗ 0 ⚠ 0
줄 26, 열 1 공백: 4 UTF-8 CRLF {} Python 3.11.1 64-bit

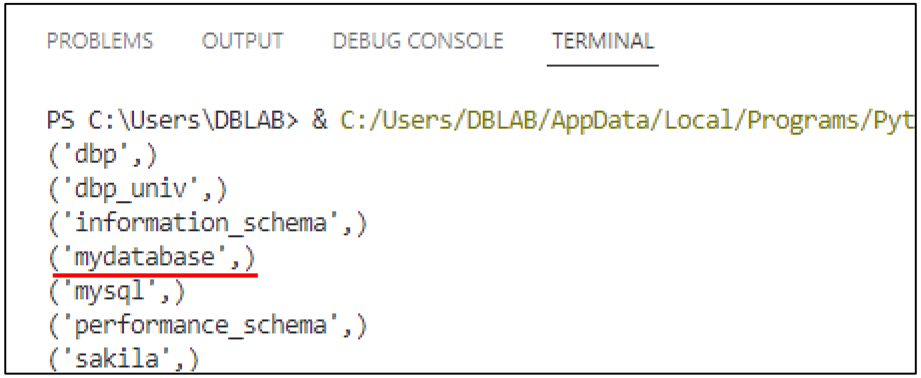
```python
import mysql.connector

mydb = mysql.connector.connect(
    host="localhost",
    user = "root",
    password = "0000"
)

mycursor = mydb.cursor()

mycursor.execute("CREATE DATABASE mydatabase")
```

데이터베이스 조회

- 생성한 데이터베이스 조회



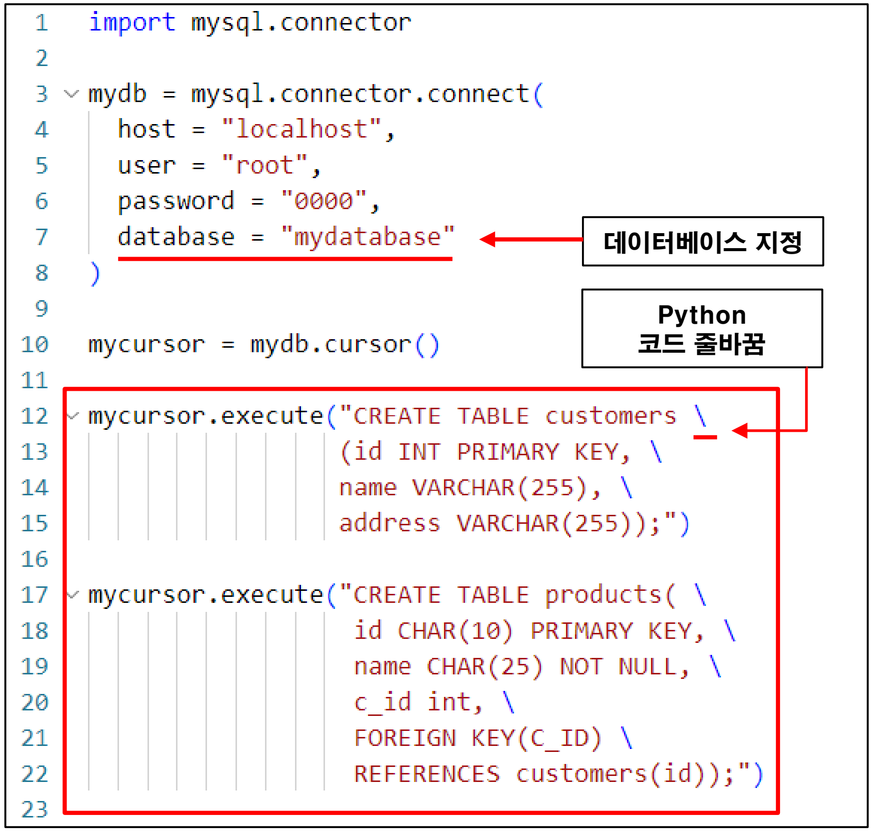
```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000"
)

mycursor = mydb.cursor()

mycursor.execute("SHOW DATABASES")

for x in mycursor:
  print(x)
```



```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/Pyt
('dbp',)
('dbp_univ',)
('information_schema',)
('mydatabase',)
('mysql',)
('performance_schema',)
('sakila',)
```

테이블 생성

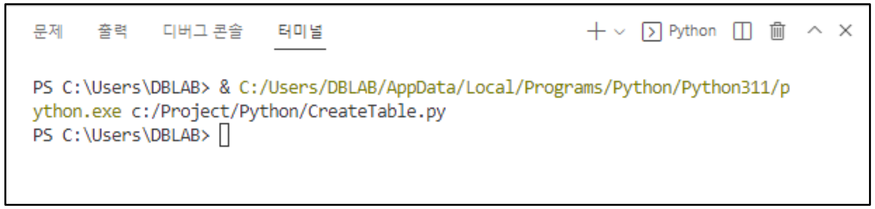
• 테이블 생성 예시



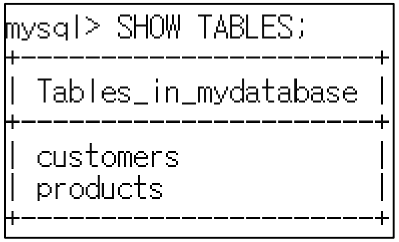
```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

mycursor.execute("CREATE TABLE customers \
                  (id INT PRIMARY KEY, \
                  name VARCHAR(255), \
                  address VARCHAR(255));")

mycursor.execute("CREATE TABLE products( \
                   id CHAR(10) PRIMARY KEY, \
                   name CHAR(25) NOT NULL, \
                   c_id int, \
                   FOREIGN KEY(C_ID) \
                   REFERENCES customers(id));")
```

데이터베이스 지정

Python
코드 줄바꿈

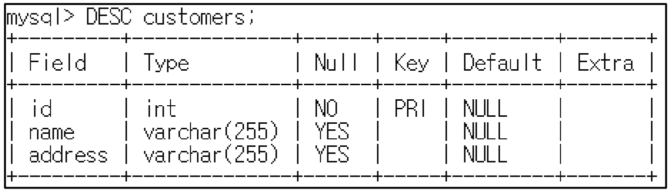


```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/Python/Python311/p
ython.exe c:/Project/Python/CreateTable.py
PS C:\Users\DBLAB> 
```



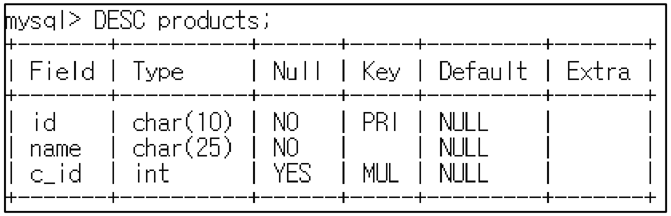
```
mysql> SHOW TABLES;
+----------------------+
| Tables_in_mydatabase |
+----------------------+
| customers            |
| products             |
+----------------------+
```

<테이블 목록>



```
mysql> DESC customers;
+---------+--------------+------+-----+---------+-------+
| Field   | Type         | Null | Key | Default | Extra |
+---------+--------------+------+-----+---------+-------+
| id      | int          | NO   | PRI | NULL    |       |
| name    | varchar(255) | YES  |     | NULL    |       |
| address | varchar(255) | YES  |     | NULL    |       |
+---------+--------------+------+-----+---------+-------+
```

<Customers 테이블 구조>

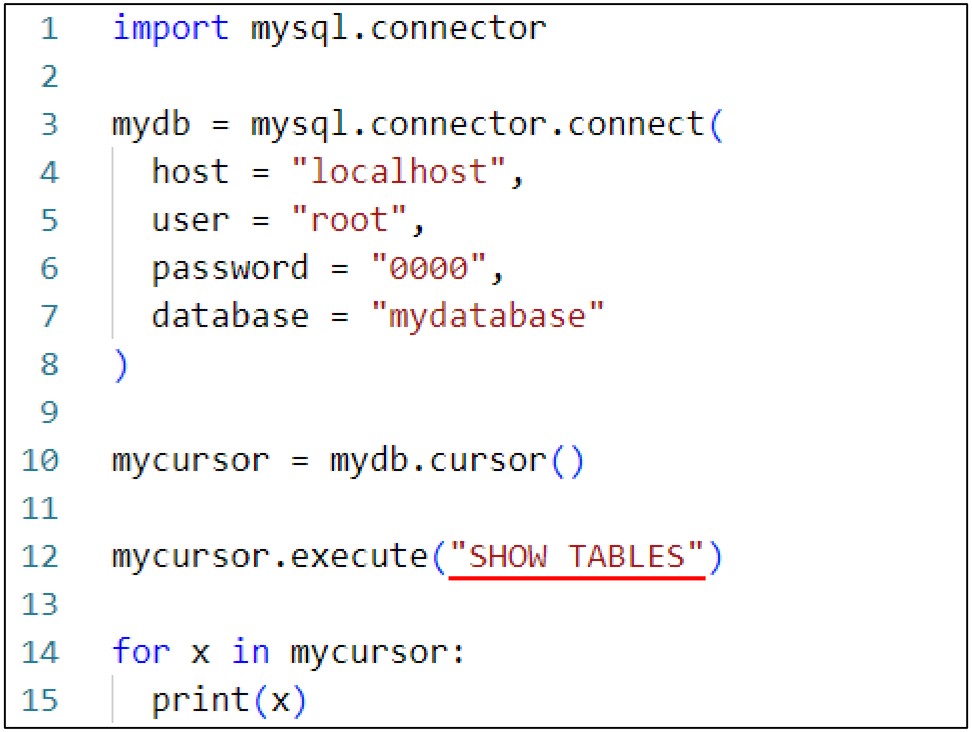


```
mysql> DESC products;
+-------+----------+------+-----+---------+-------+
| Field | Type     | Null | Key | Default | Extra |
+-------+----------+------+-----+---------+-------+
| id    | char(10) | NO   | PRI | NULL    |       |
| name  | char(25) | NO   |     | NULL    |       |
| c_id  | int      | YES  | MUL | NULL    |       |
+-------+----------+------+-----+---------+-------+
```

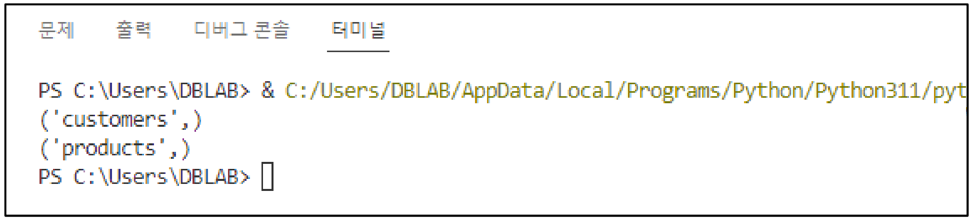
<Products 테이블 구조>

테이블 조회

- **생성한 테이블 조회**



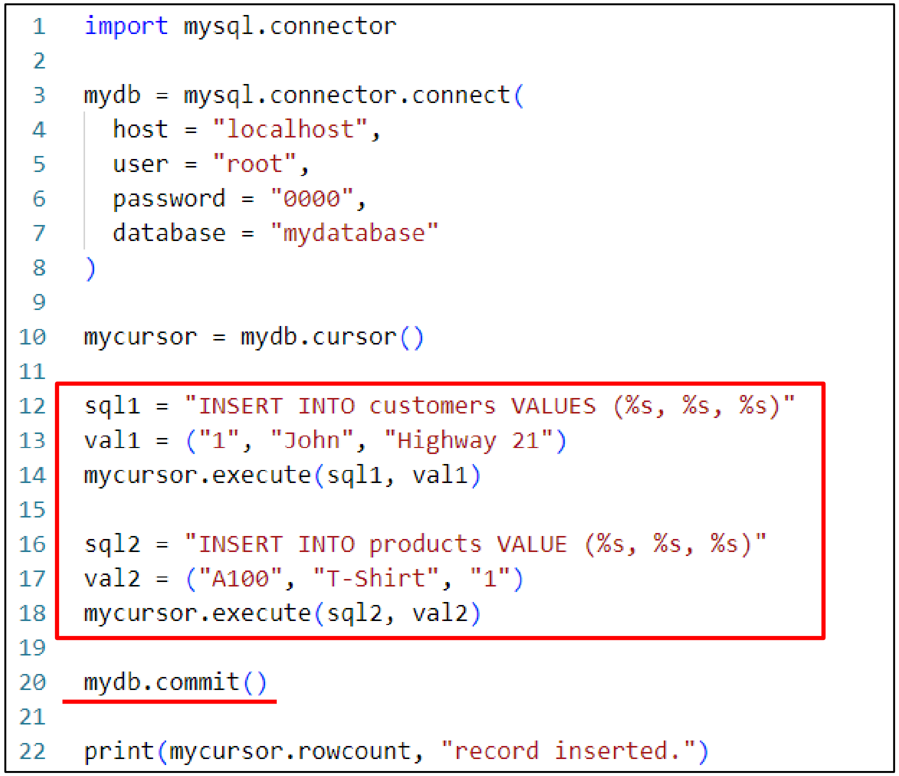
```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

mycursor.execute("SHOW TABLES")

for x in mycursor:
  print(x)
```



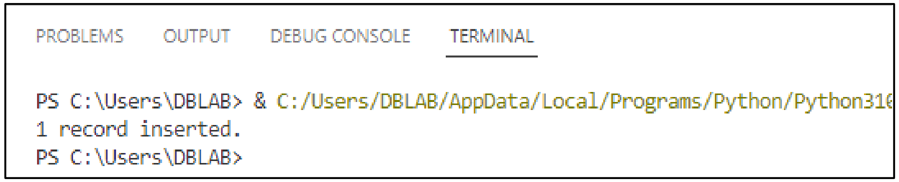
```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/Python/Python311/pyt
('customers',)
('products',)
PS C:\Users\DBLAB> 
```

데이터 삽입(1/2)

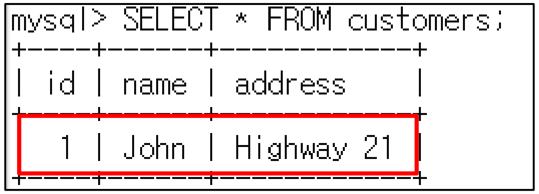
- **단일 행 삽입**



```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

sql1 = "INSERT INTO customers VALUES (%s, %s, %s)"
val1 = ("1", "John", "Highway 21")
mycursor.execute(sql1, val1)

sql2 = "INSERT INTO products VALUE (%s, %s, %s)"
val2 = ("A100", "T-Shirt", "1")
mycursor.execute(sql2, val2)

mydb.commit()

print(mycursor.rowcount, "record inserted.")
```

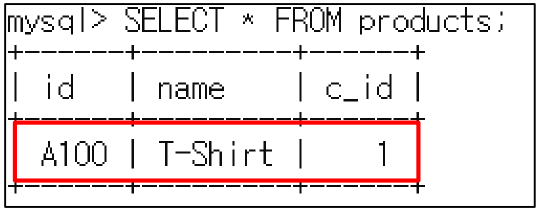


```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/Python/Python31
1 record inserted.
PS C:\Users\DBLAB>
```



```
mysql> SELECT * FROM customers;
+----+------+------------+
| id | name | address    |
+----+------+------------+
|  1 | John | Highway 21 |
+----+------+------------+
```

<Customers 테이블>

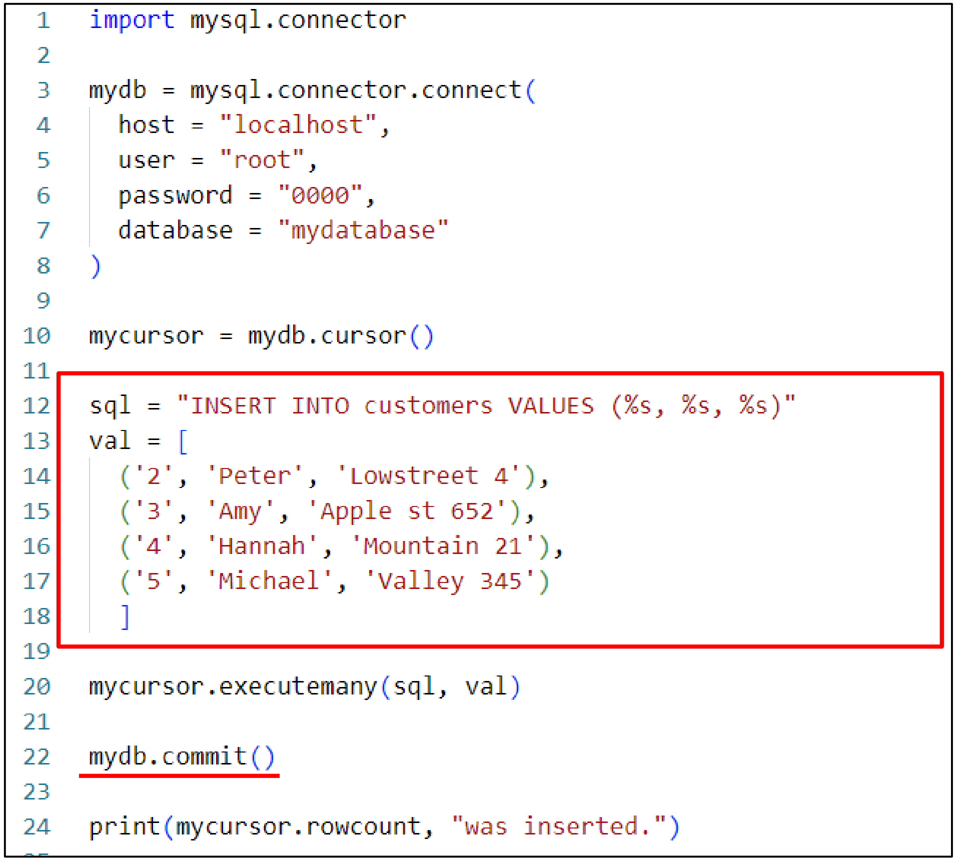


```
mysql> SELECT * FROM products;
+------+---------+------+
| id   | name    | c_id |
+------+---------+------+
| A100 | T-Shirt |    1 |
+------+---------+------+
```

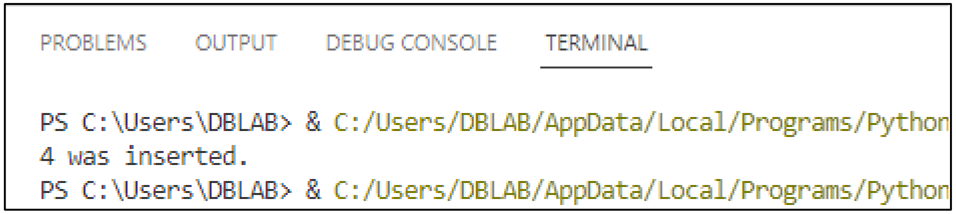
<Prodeucts 테이블>

데이터 삽입(2/2)

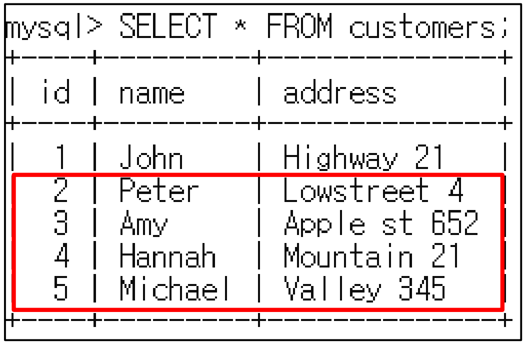
- **복수 행 삽입**



```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

sql = "INSERT INTO customers VALUES (%s, %s, %s)"
val = [
  ('2', 'Peter', 'Lowstreet 4'),
  ('3', 'Amy', 'Apple st 652'),
  ('4', 'Hannah', 'Mountain 21'),
  ('5', 'Michael', 'Valley 345')
  ]

mycursor.executemany(sql, val)

mydb.commit()

print(mycursor.rowcount, "was inserted.")
```



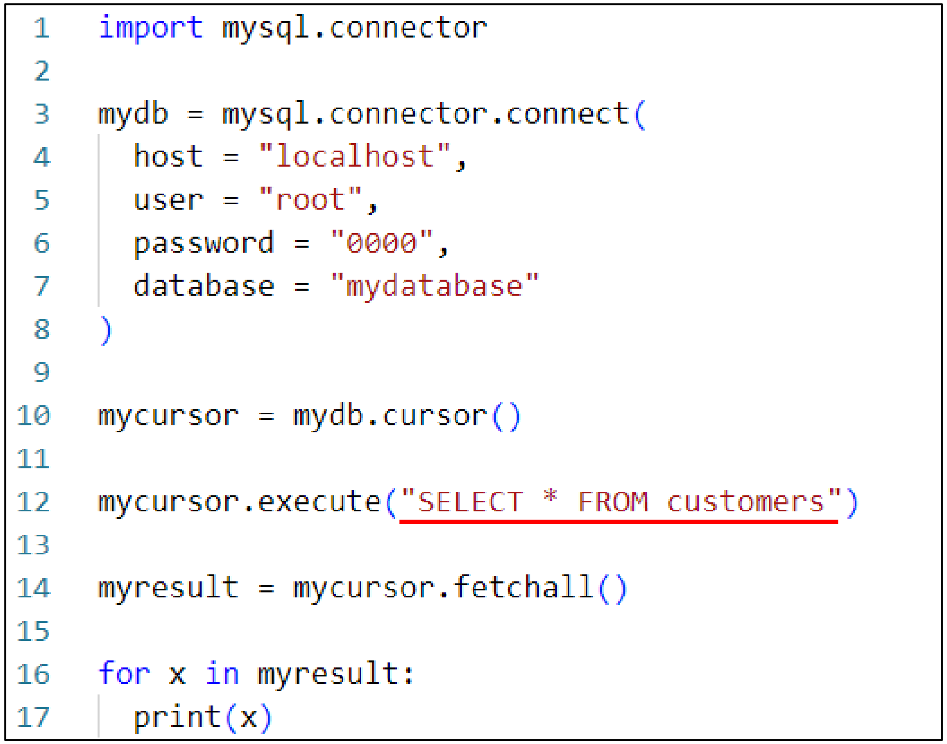
```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/Python
4 was inserted.
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/Python
```



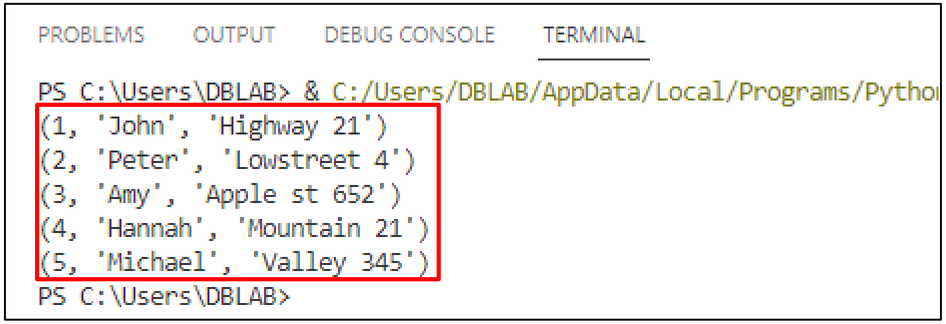
```
mysql> SELECT * FROM customers;
+----+---------+-----------------+
| id | name    | address         |
+----+---------+-----------------+
|  1 | John    | Highway 21      |
|  2 | Peter   | Lowstreet 4     |
|  3 | Amy     | Apple st 652    |
|  4 | Hannah  | Mountain 21     |
|  5 | Michael | Valley 345      |
+----+---------+-----------------+
```

데이터 조회(1/10)

- **전체 데이터 검색**



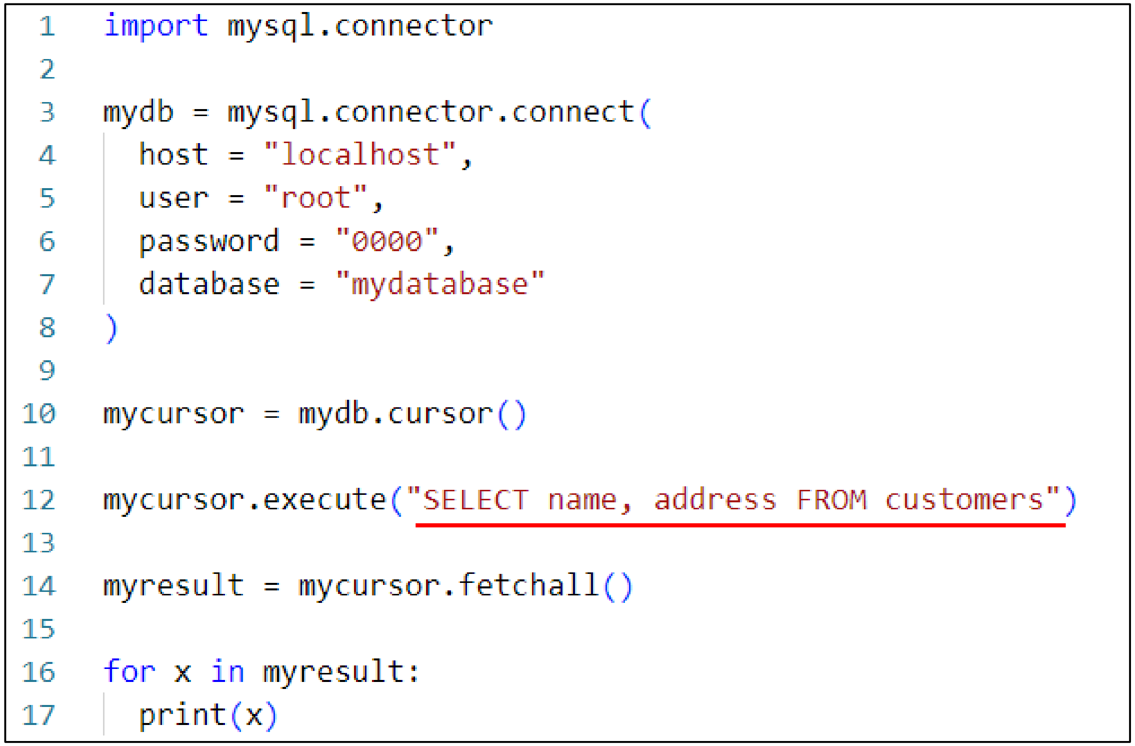
```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

mycursor.execute("SELECT * FROM customers")

myresult = mycursor.fetchall()

for x in myresult:
  print(x)
```



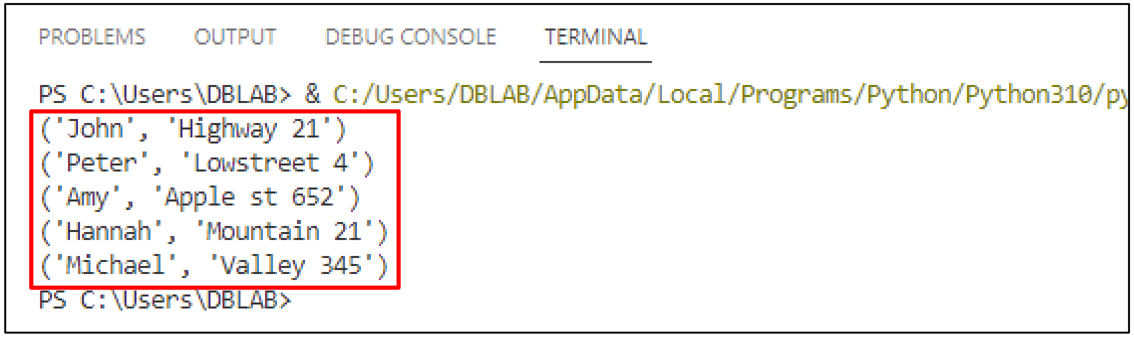
```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/Pytho
(1, 'John', 'Highway 21')
(2, 'Peter', 'Lowstreet 4')
(3, 'Amy', 'Apple st 652')
(4, 'Hannah', 'Mountain 21')
(5, 'Michael', 'Valley 345')
PS C:\Users\DBLAB>
```

데이터 조회(2/10)

- 일부 열 검색



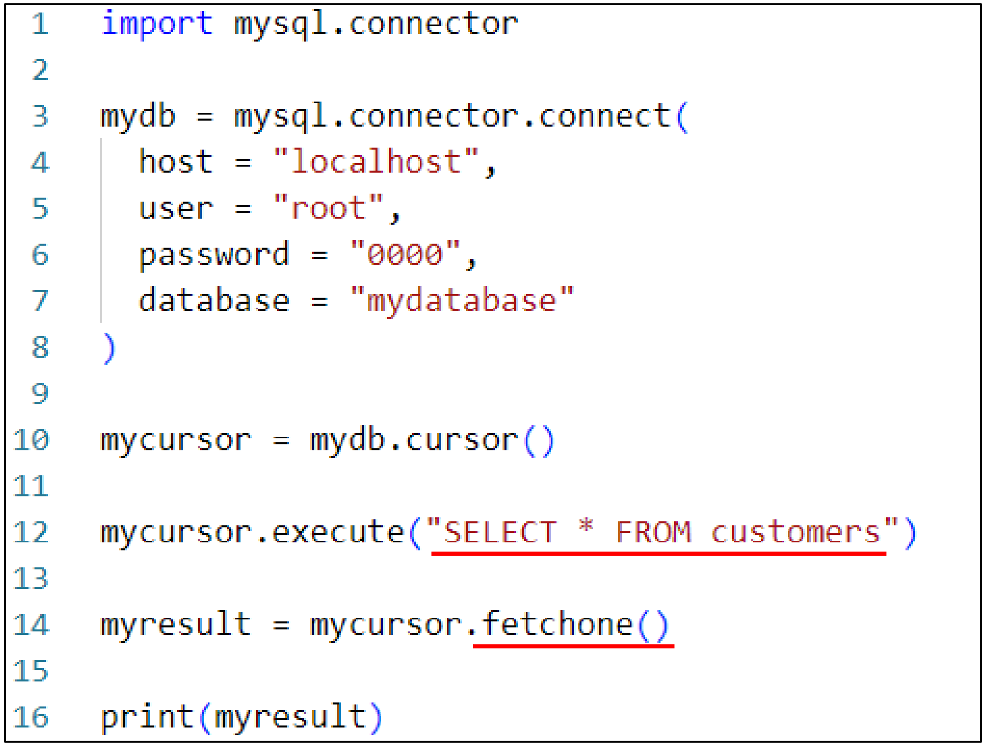
```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

mycursor.execute("SELECT name, address FROM customers")

myresult = mycursor.fetchall()

for x in myresult:
  print(x)
```



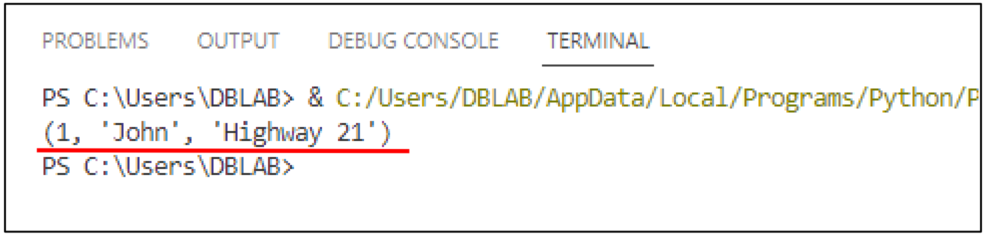
```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/Python/Python310/py
('John', 'Highway 21')
('Peter', 'Lowstreet 4')
('Amy', 'Apple st 652')
('Hannah', 'Mountain 21')
('Michael', 'Valley 345')
PS C:\Users\DBLAB>
```

데이터 조회(3/10)

- **단일 행 검색**



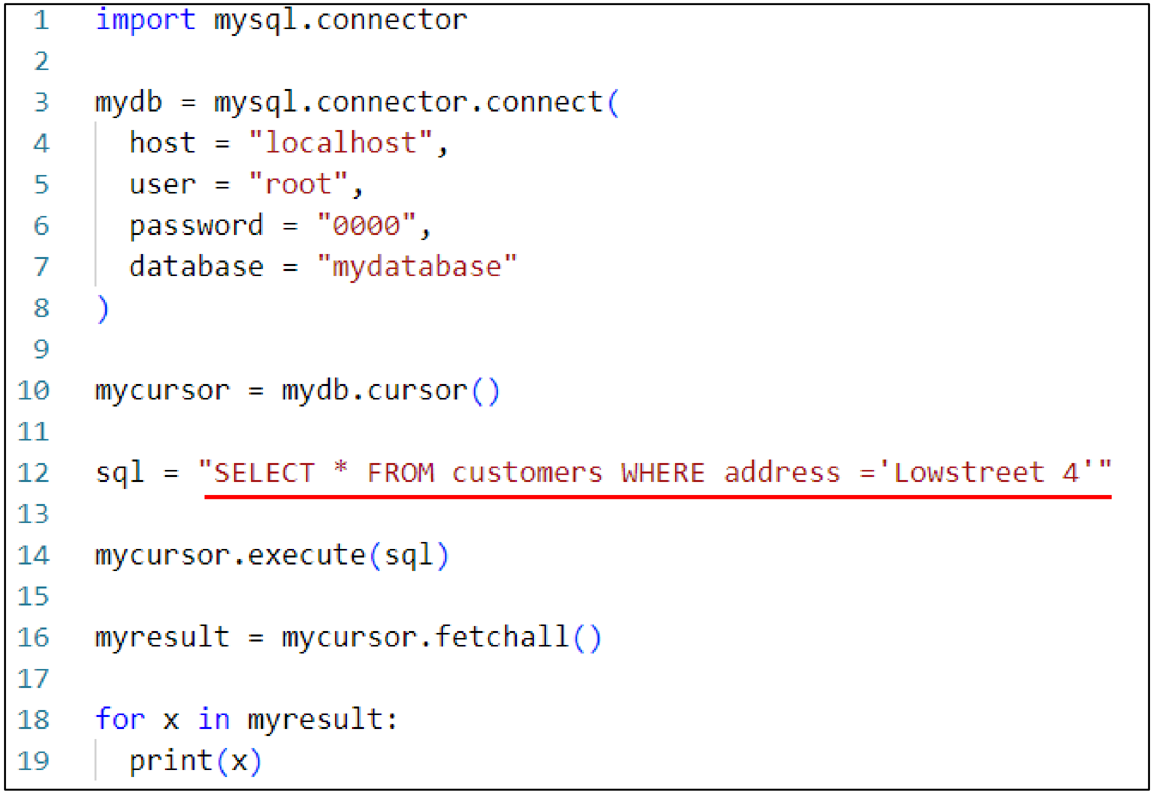
```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

mycursor.execute("SELECT * FROM customers")

myresult = mycursor.fetchone()

print(myresult)
```



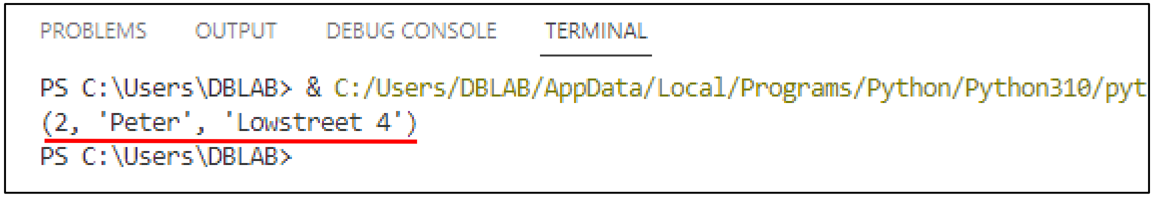
```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/Python/P
(1, 'John', 'Highway 21')
PS C:\Users\DBLAB>
```

데이터 조회(4/10)

- **조건을 만족하는 특정 행 검색**



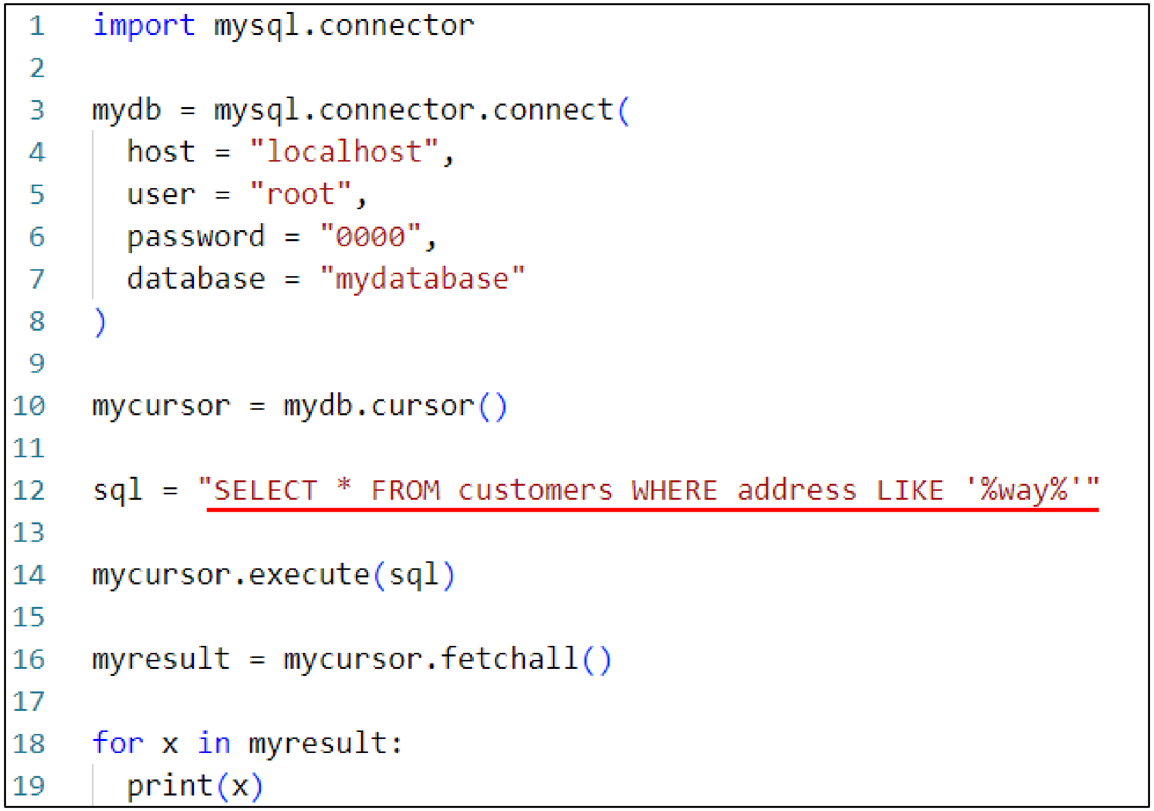
```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

sql = "SELECT * FROM customers WHERE address ='Lowstreet 4'"

mycursor.execute(sql)

myresult = mycursor.fetchall()

for x in myresult:
  print(x)
```



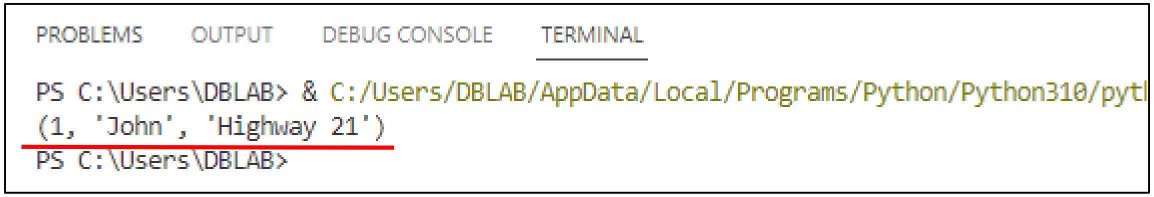
```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/Python/Python310/pyt
(2, 'Peter', 'Lowstreet 4')
PS C:\Users\DBLAB>
```

데이터 조회(5/10)

- **LIKE 연산자로 문자열을 비교하여 특정 행 검색**



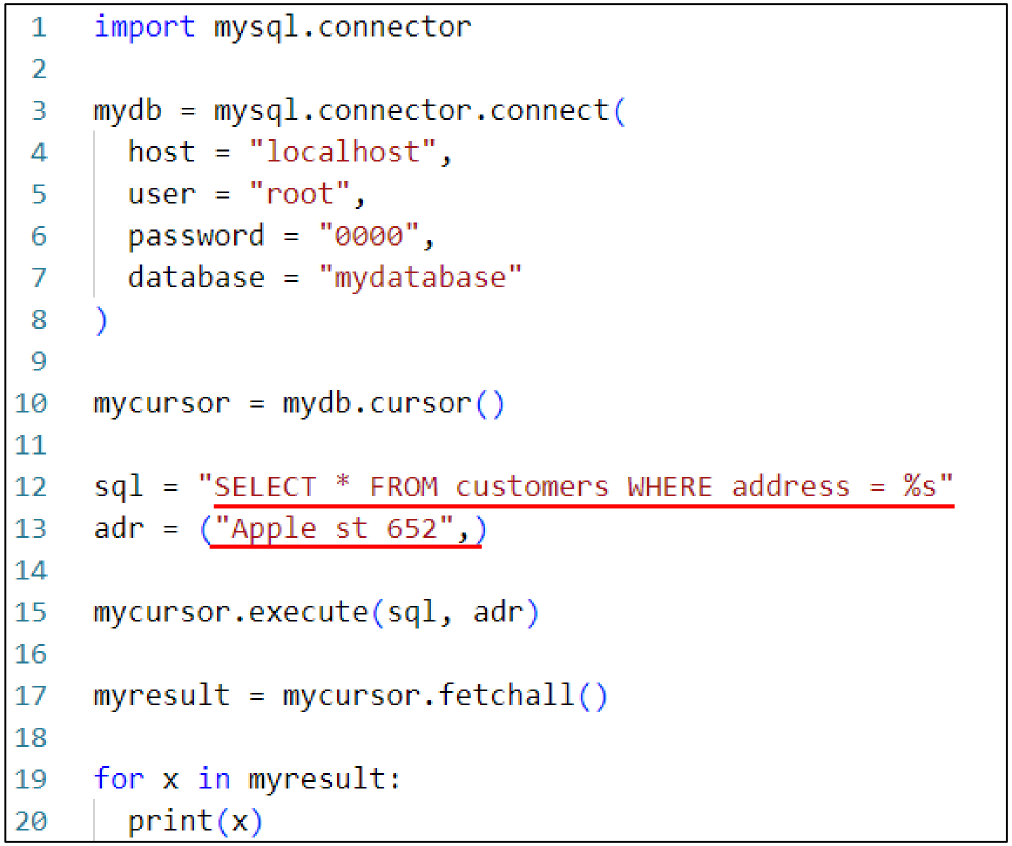
```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

sql = "SELECT * FROM customers WHERE address LIKE '%way%'"

mycursor.execute(sql)

myresult = mycursor.fetchall()

for x in myresult:
  print(x)
```



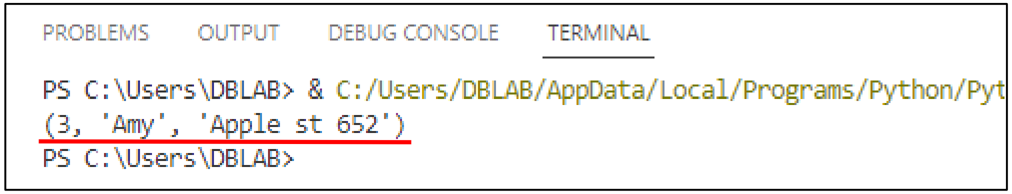
```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/Python/Python310/pytl
(1, 'John', 'Highway 21')
PS C:\Users\DBLAB>
```

데이터 조회(6/10)

- **파라미터 전달 방식으로 특정 행 검색**



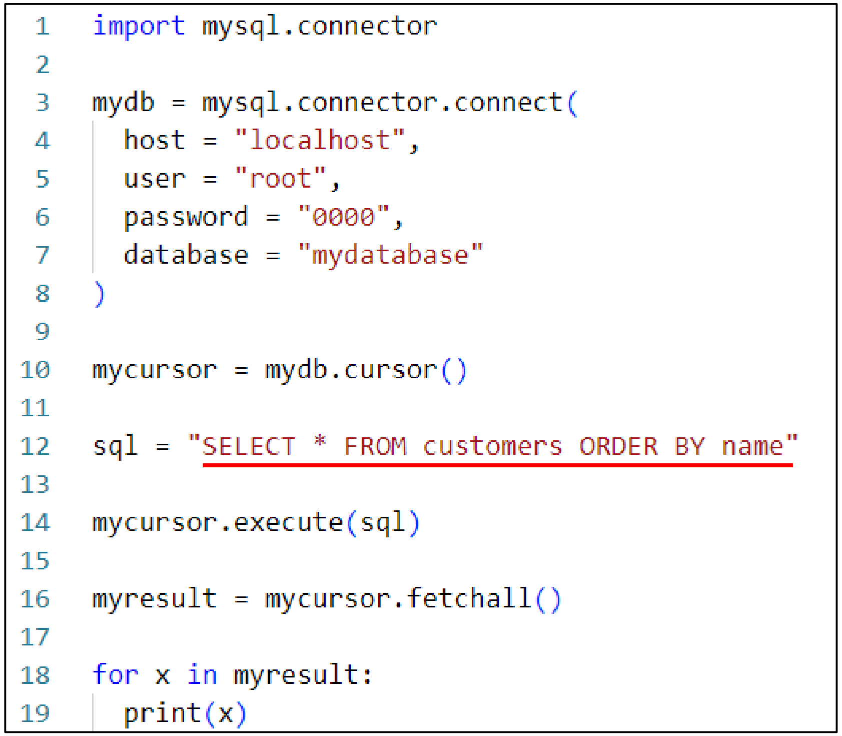
```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

sql = "SELECT * FROM customers WHERE address = %s"
adr = ("Apple st 652",)

mycursor.execute(sql, adr)

myresult = mycursor.fetchall()

for x in myresult:
  print(x)
```



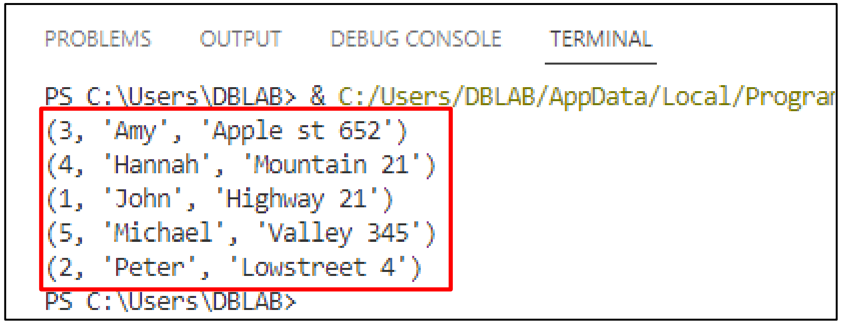
```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/Python/Pyt
(3, 'Amy', 'Apple st 652')
PS C:\Users\DBLAB>
```

데이터 조회(7/10)

- **Order by를 사용해 오름차순으로 정렬**



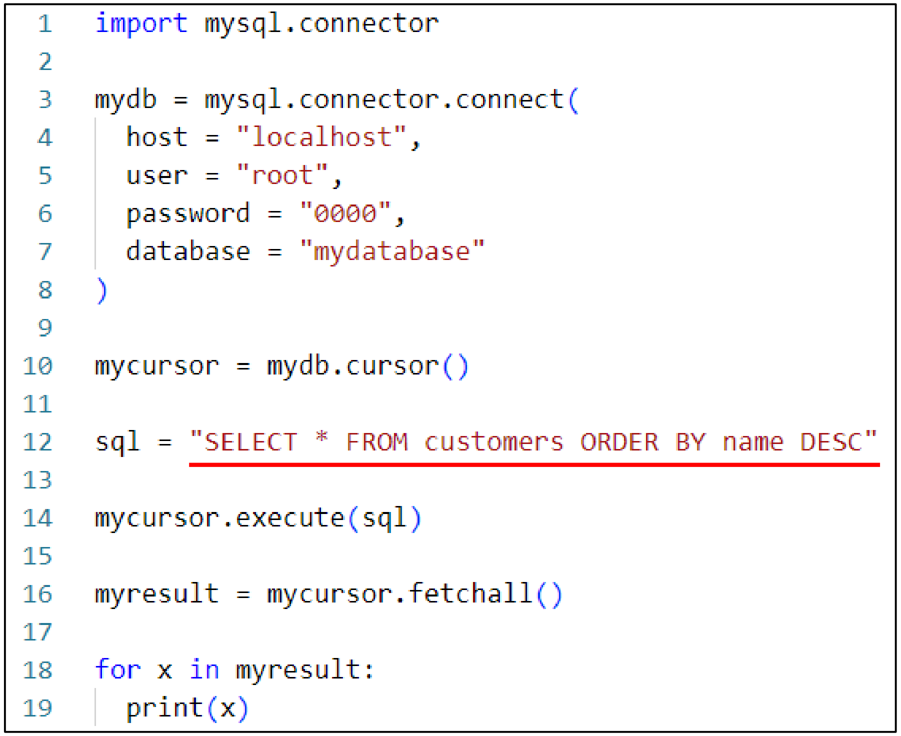
```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

sql = "SELECT * FROM customers ORDER BY name"

mycursor.execute(sql)

myresult = mycursor.fetchall()

for x in myresult:
  print(x)
```



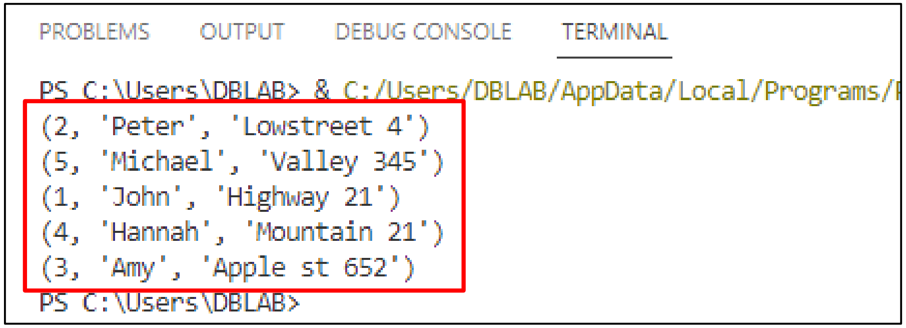
```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Progra
(3, 'Amy', 'Apple st 652')
(4, 'Hannah', 'Mountain 21')
(1, 'John', 'Highway 21')
(5, 'Michael', 'Valley 345')
(2, 'Peter', 'Lowstreet 4')
PS C:\Users\DBLAB>
```

데이터 조회(8/10)

- **Order by를 사용해 내림차순으로 정렬**



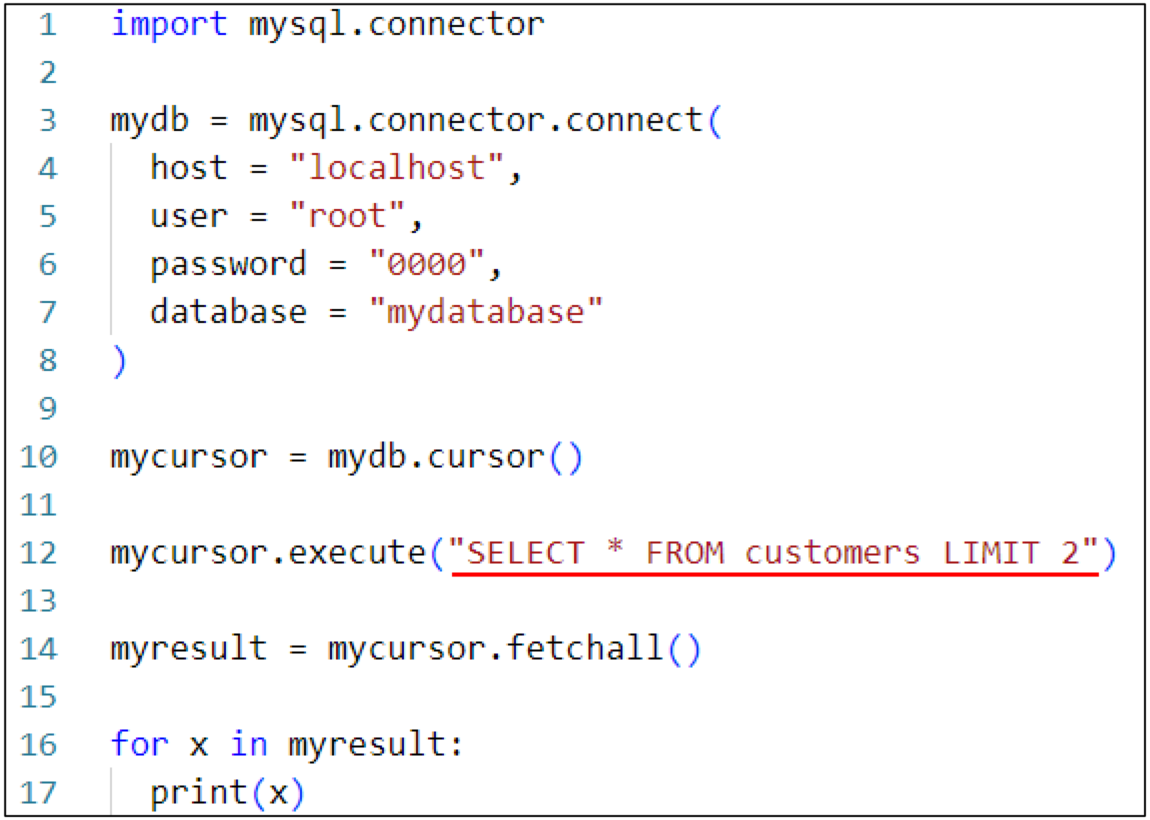
```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

sql = "SELECT * FROM customers ORDER BY name DESC"

mycursor.execute(sql)

myresult = mycursor.fetchall()

for x in myresult:
  print(x)
```



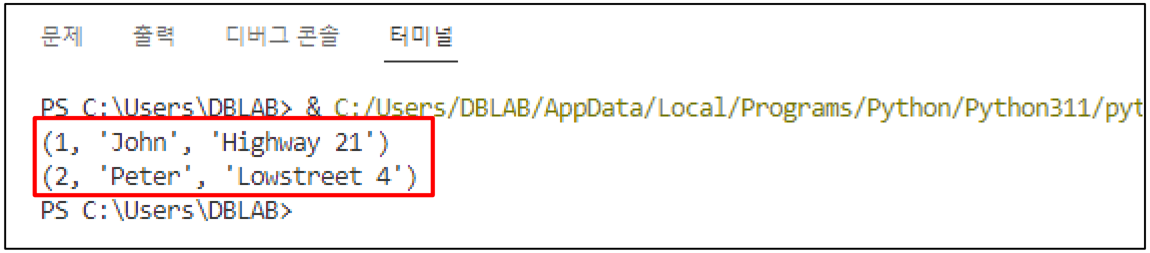
```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/P
(2, 'Peter', 'Lowstreet 4')
(5, 'Michael', 'Valley 345')
(1, 'John', 'Highway 21')
(4, 'Hannah', 'Mountain 21')
(3, 'Amy', 'Apple st 652')
PS C:\Users\DBLAB>
```

데이터 조회(9/10)

- 반환 개수를 지정하여 조회



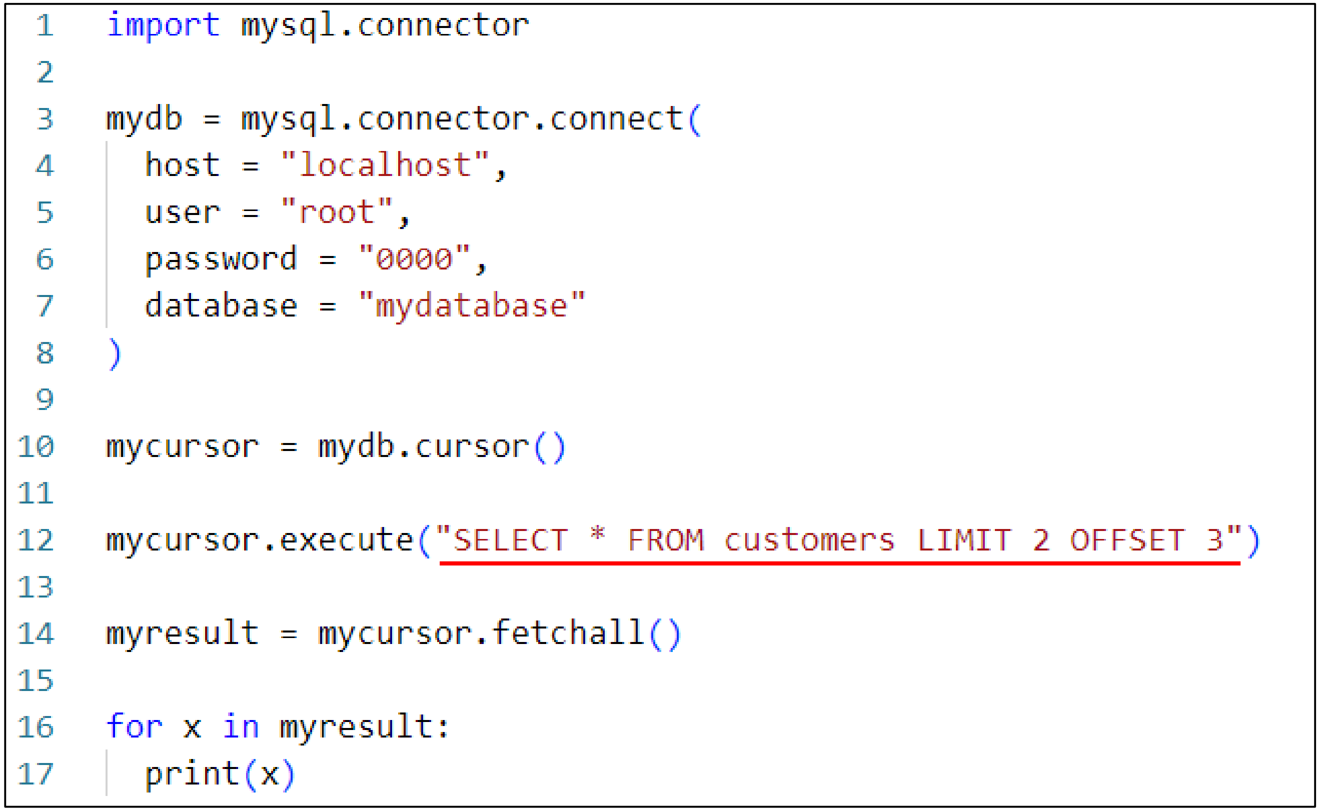
```
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

mycursor.execute("SELECT * FROM customers LIMIT 2")

myresult = mycursor.fetchall()

for x in myresult:
  print(x)
```



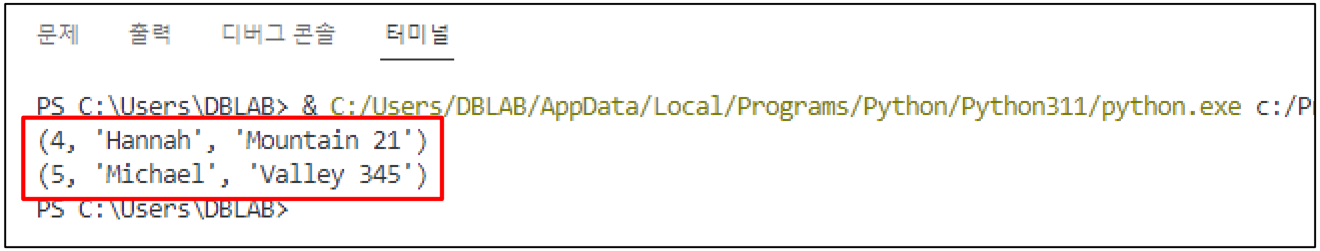
```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/Python/Python311/pyt
(1, 'John', 'Highway 21')
(2, 'Peter', 'Lowstreet 4')
PS C:\Users\DBLAB>
```

데이터 조회(10/10)

- 반환 시작 지점을 지정하여 조회



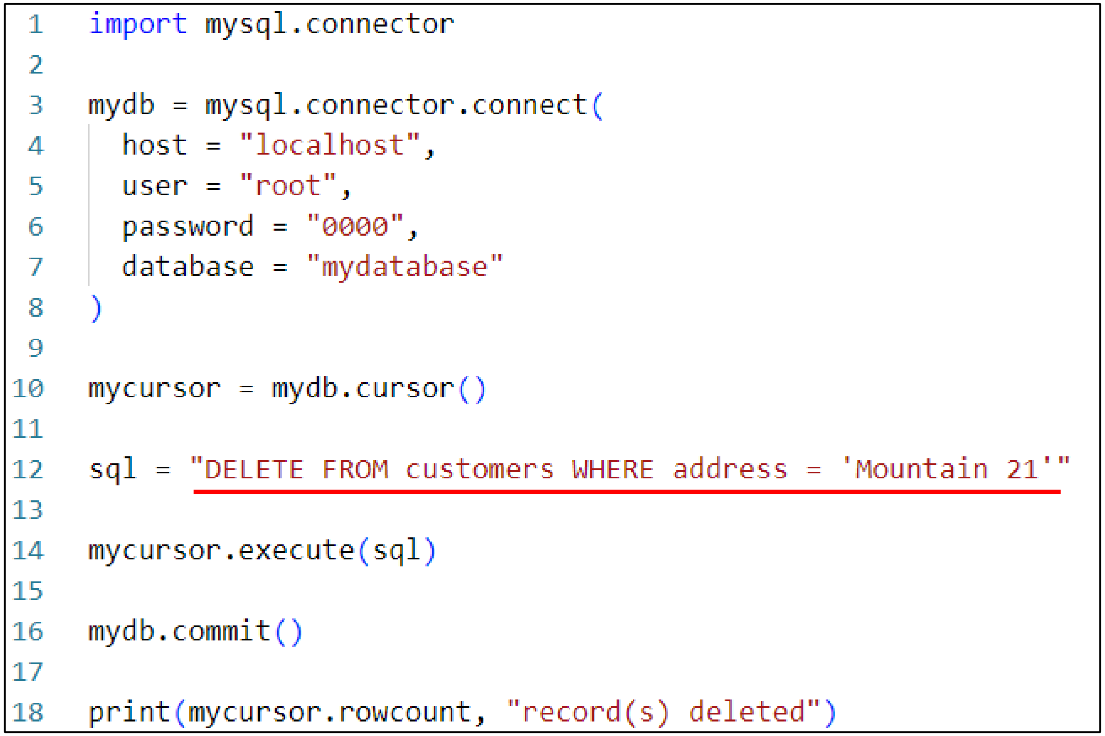
```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

mycursor.execute("SELECT * FROM customers LIMIT 2 OFFSET 3")

myresult = mycursor.fetchall()

for x in myresult:
  print(x)
```



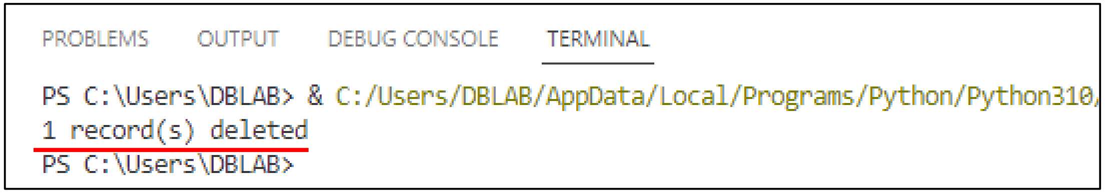
```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/Python/Python311/python.exe c:/P
(4, 'Hannah', 'Mountain 21')
(5, 'Michael', 'Valley 345')
PS C:\Users\DBLAB>
```

데이터 삭제(1/2)

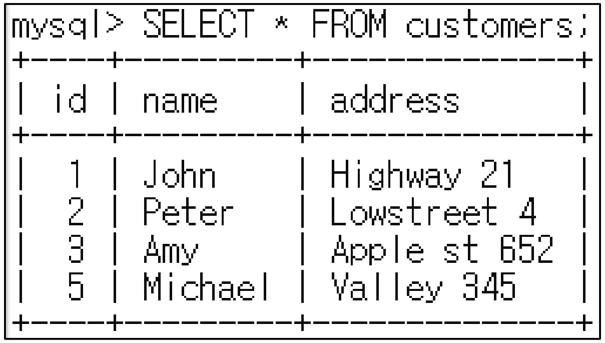
- **조건을 만족하는 특정 행 삭제**



```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

sql = "DELETE FROM customers WHERE address = 'Mountain 21'"

mycursor.execute(sql)

mydb.commit()

print(mycursor.rowcount, "record(s) deleted")
```



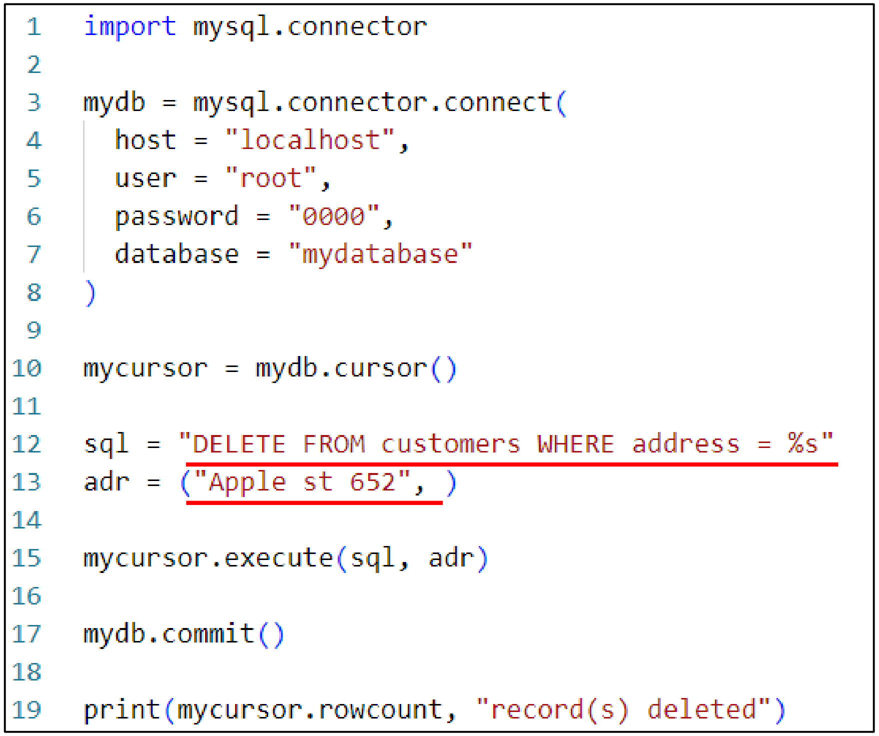
```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/Python/Python310/
1 record(s) deleted
PS C:\Users\DBLAB>
```



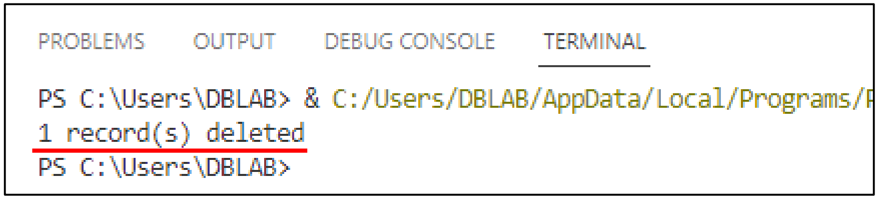
```
mysql> SELECT * FROM customers;
+----+---------+---------------+
| id | name    | address       |
+----+---------+---------------+
|  1 | John    | Highway 21    |
|  2 | Peter   | Lowstreet 4   |
|  3 | Amy     | Apple st 652  |
|  5 | Michael | Valley 345    |
+----+---------+---------------+
```

데이터 삭제(2/2)

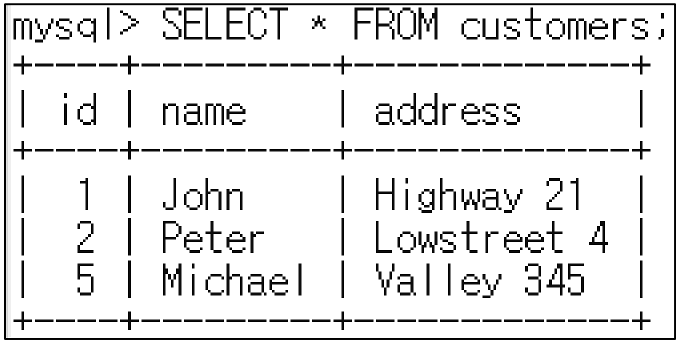
- **파라미터 전달 방식으로 특정 행 삭제**



```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

sql = "DELETE FROM customers WHERE address = %s"
adr = ("Apple st 652", )

mycursor.execute(sql, adr)

mydb.commit()

print(mycursor.rowcount, "record(s) deleted")
```



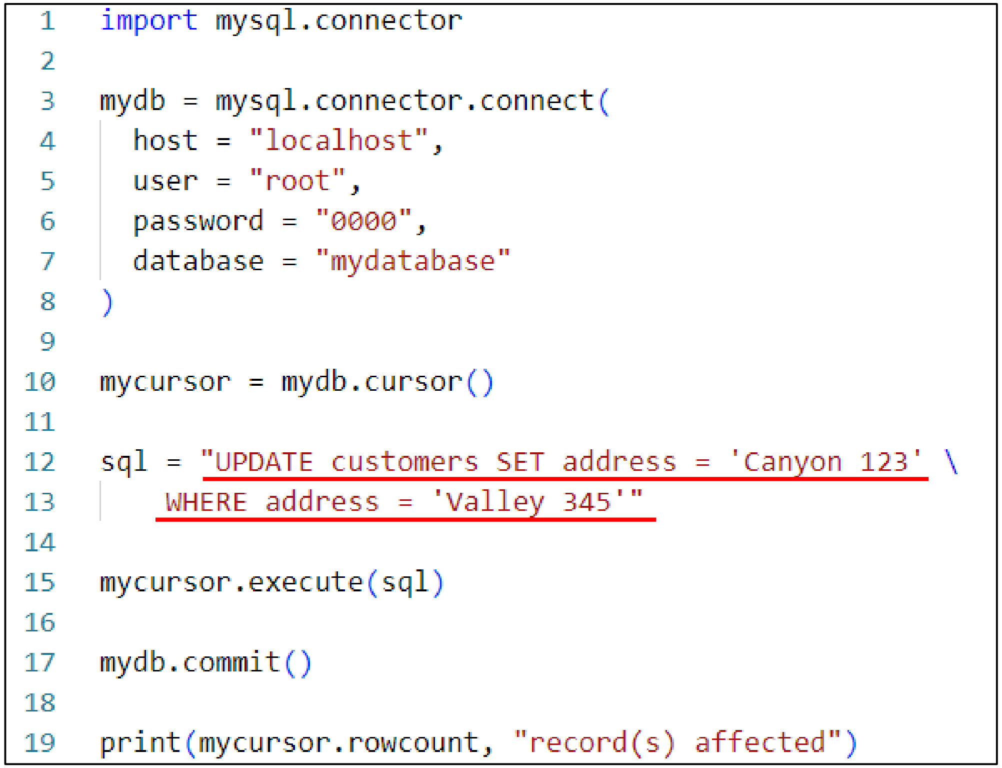
```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/
1 record(s) deleted
PS C:\Users\DBLAB>
```



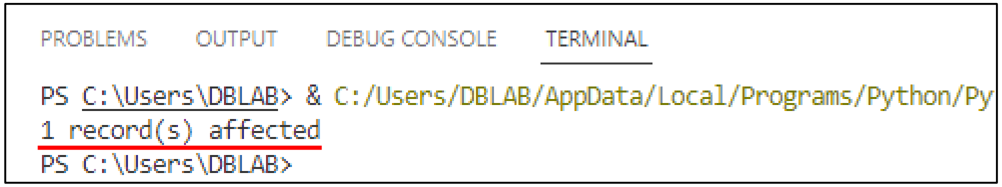
```
mysql> SELECT * FROM customers;
+----+---------+--------------+
| id | name    | address      |
+----+---------+--------------+
|  1 | John    | Highway 21   |
|  2 | Peter   | Lowstreet 4  |
|  5 | Michael | Valley 345   |
+----+---------+--------------+
```

데이터 수정(1/2)

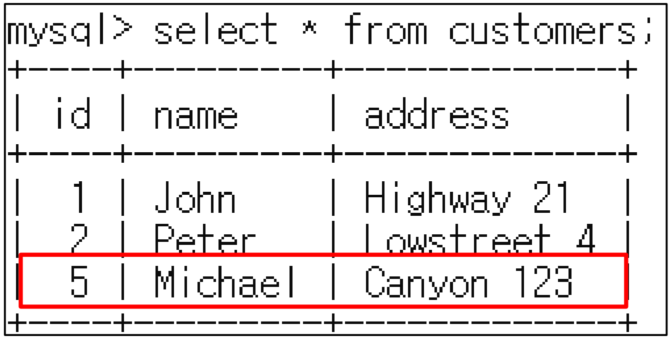
- **조건을 만족하는 특정 행 수정**



```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

sql = "UPDATE customers SET address = 'Canyon 123' \
   WHERE address = 'Valley 345'"

mycursor.execute(sql)

mydb.commit()

print(mycursor.rowcount, "record(s) affected")
```



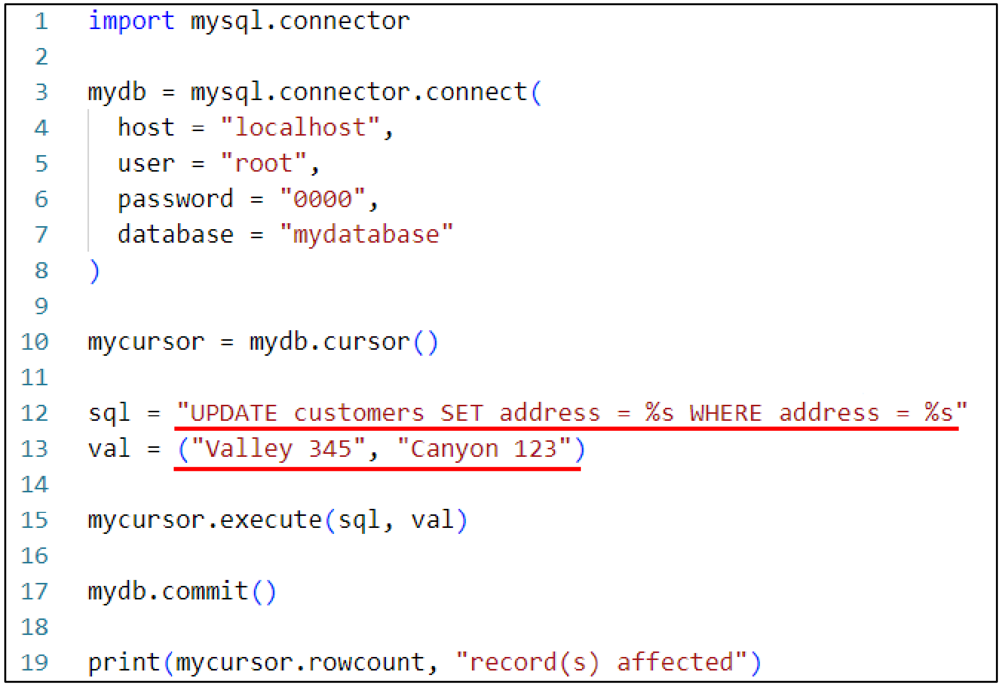
```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/Python/Py
1 record(s) affected
PS C:\Users\DBLAB>
```



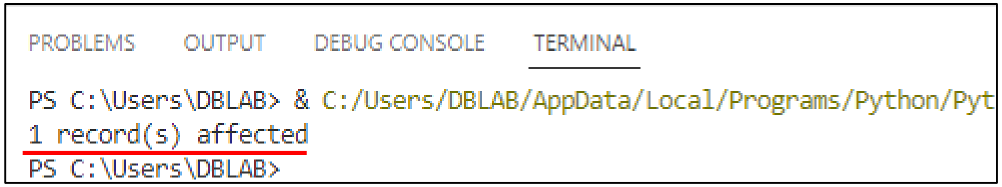
```
mysql> select * from customers;
+----+-----------+--------------+
| id | name      | address      |
+----+-----------+--------------+
|  1 | John      | Highway 21   |
|  2 | Peter     | Lowstreet 4  |
|  5 | Michael   | Canyon 123   |
+----+-----------+--------------+
```

데이터 수정(2/2)

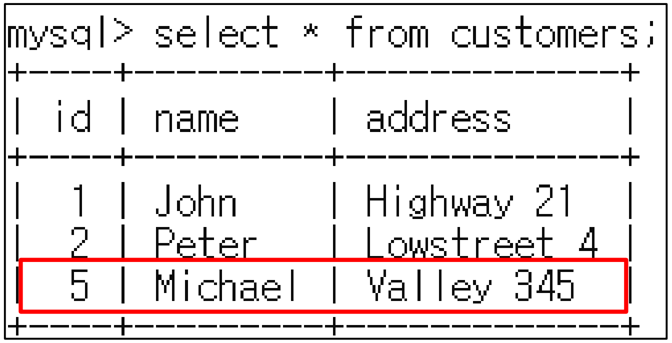
- **파라미터 전달 방식으로 특정 행 수정**



```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

sql = "UPDATE customers SET address = %s WHERE address = %s"
val = ("Valley 345", "Canyon 123")

mycursor.execute(sql, val)

mydb.commit()

print(mycursor.rowcount, "record(s) affected")
```



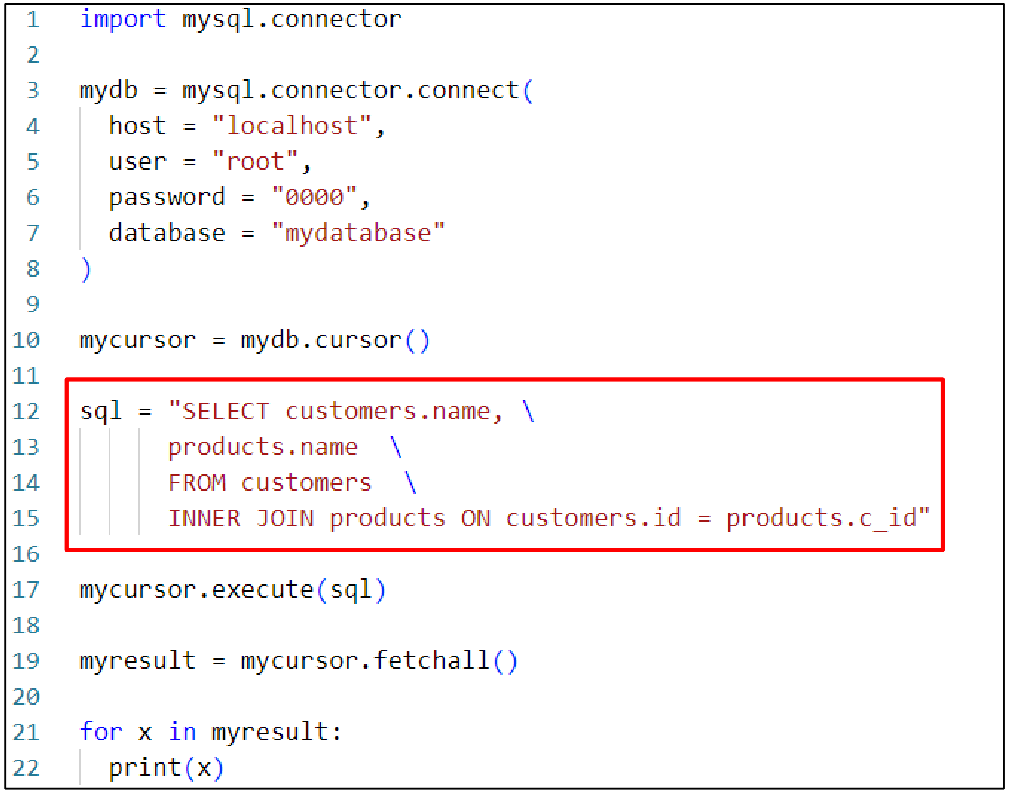
```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/Python/Pyt
1 record(s) affected
PS C:\Users\DBLAB>
```



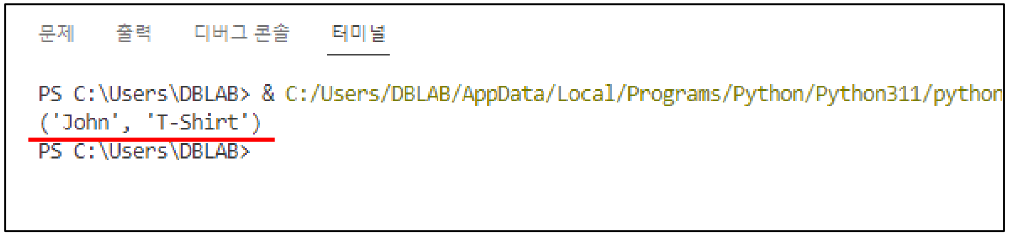
```
mysql> select * from customers;
+----+---------+--------------+
| id | name    | address      |
+----+---------+--------------+
|  1 | John    | Highway 21   |
|  2 | Peter   | Lowstreet 4  |
|  5 | Michael | Valley 345   |
+----+---------+--------------+
```

테이블 조인

- **두 개의 테이블을 조인하여 조회 (INNER JOIN)**



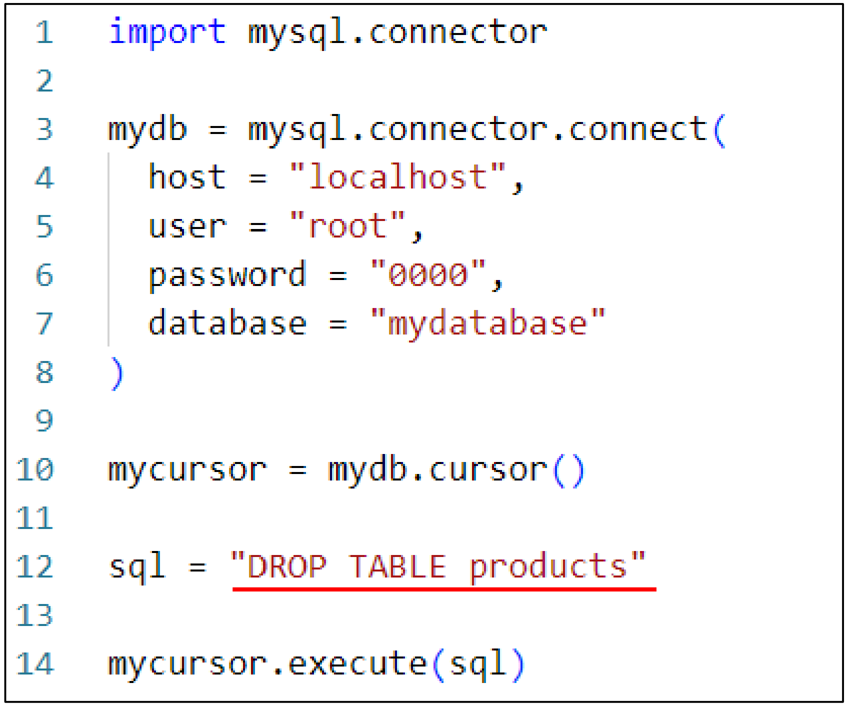
```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

sql = "SELECT customers.name, \
      products.name  \
      FROM customers  \
      INNER JOIN products ON customers.id = products.c_id"

mycursor.execute(sql)

myresult = mycursor.fetchall()

for x in myresult:
  print(x)
```



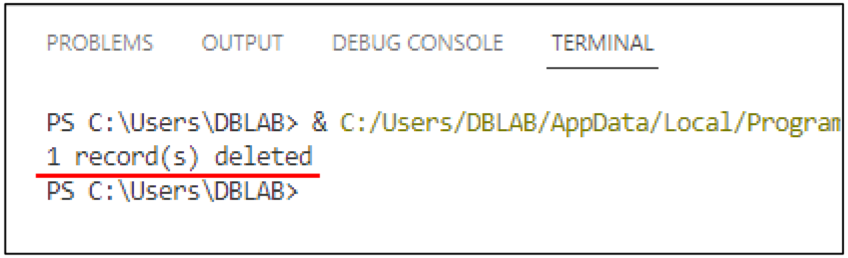
```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Programs/Python/Python311/python
('John', 'T-Shirt')
PS C:\Users\DBLAB>
```

테이블 삭제(1/2)

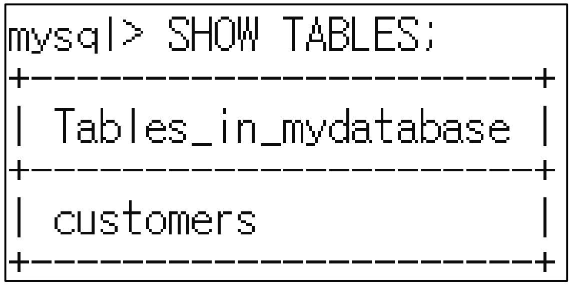
- **테이블 삭제 예시**



```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

sql = "DROP TABLE products"

mycursor.execute(sql)
```



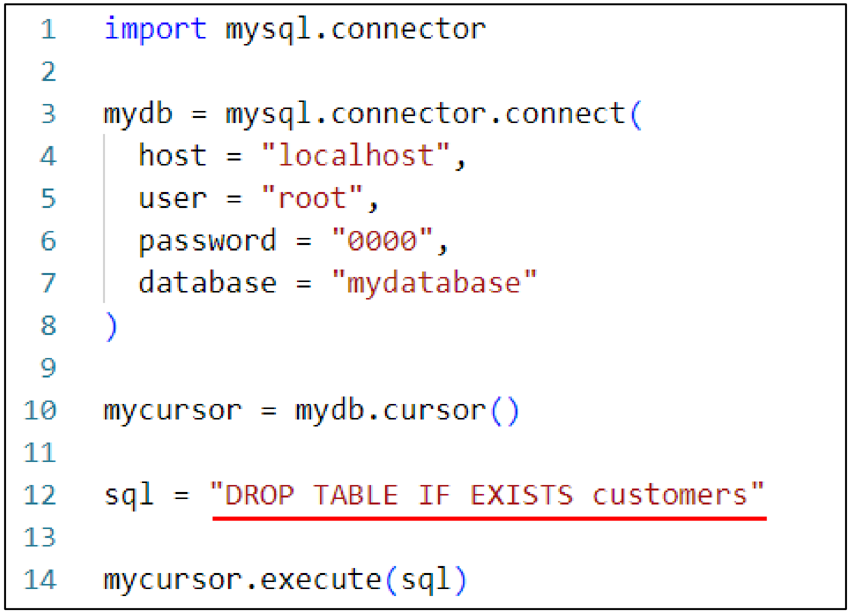
```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Program
1 record(s) deleted
PS C:\Users\DBLAB>
```



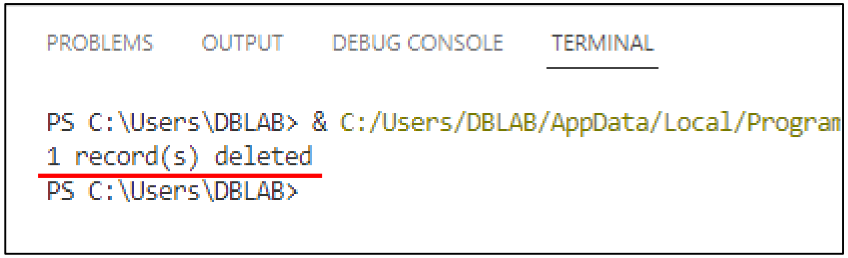
```
mysql> SHOW TABLES;
+----------------------+
| Tables_in_mydatabase |
+----------------------+
| customers            |
+----------------------+
```

테이블 삭제(2/2)

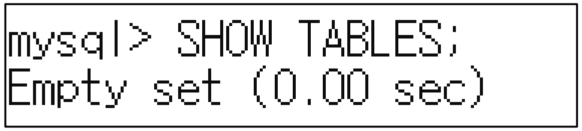
- **테이블 삭제 예시 – IF EXISTS 사용**



```python
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
  user = "root",
  password = "0000",
  database = "mydatabase"
)

mycursor = mydb.cursor()

sql = "DROP TABLE IF EXISTS customers"

mycursor.execute(sql)
```



```
PS C:\Users\DBLAB> & C:/Users/DBLAB/AppData/Local/Program
1 record(s) deleted
PS C:\Users\DBLAB>
```



```
mysql> SHOW TABLES;
Empty set (0.00 sec)
```