

Python with MySQL

Auto Mobile Robot

Exported on 06/25/2024

Table of Contents

1	실습환경 만들기	4
1.1	VSCoDe 실행.....	4
1.2	MySQL	4
1.3	데이터 삭제.....	4
1.4	python.ipynb 파일 생성.....	4
2	Python with MySQL.....	5
2.1	Install MySQL Driver - 1	5
2.2	Install MySQL Driver - 2	5
2.3	Create Connection	5
2.4	Create Connection 예제 1	5
2.5	Create Connection 예제 2	6
2.6	Close Database	6
2.7	Close Database 예제 1	6
2.8	Close Database 예제 2	7
2.9	Connect to Database	7
2.10	Create Connection 예제 1	7
2.11	Create Connection 예제 2	8
2.12	Execute SQL	8
2.13	Execute SQL 예제 1 - 1	8
2.14	Execute SQL 예제 1 - 2	9
2.15	Execute SQL 예제 2 - 1	9
2.16	Execute SQL 예제 2 - 2	9
2.17	Execute SQL File 1	10
2.18	Execute SQL File 1 예제 - 1	10
2.19	Execute SQL File 1 예제 - 2	10
2.20	Execute SQL File 1 예제 - 3	11
2.21	Execute SQL File 2	11

2.22 Execute SQL File 2 예제 - 1	11
2.23 Execute SQL File 2 예제 - 2	12
2.24 Execute SQL File 2 예제 - 3	12
2.25 Execute SQL File 2 예제 - 4	13
2.26 Fetch All	13
2.27 Fetch All 예제	13
2.28 참고. 검색결과를 Pandas 로 읽기	14
3 Python with CSV	15
3.1 csv 에 있는 데이터를 Python 으로 INSERT	15
3.2 Read CSV	15
3.3 amrbase 에 연결	15
3.4 Cursor 만들기	16
3.5 INSERT 문 만들기	16
3.6 데이터 입력	16
3.7 결과 확인	17
3.8 검색결과를 Pandas 로 읽기	17
3.9 Tip	17
4 Python with CSV 예제	19
4.1 AWS RDS(database-1) amrbase 에 접속	19
4.2 2020_crime.csv 데이터(encoding='euc-kr') 읽어오기	19
4.3 INSERT 쿼리 작성	19
4.4 데이터를 crime_status 테이블에 INSERT	20
4.5 crime_status 테이블의 데이터 조회	20
4.6 조회한 결과를 Pandas 로 변환해서 확인	20
5 연습	22

1 실습환경 만들기

1.1 VSCode 실행

VSCode 와 DBMS(mysql)는 모두 sql 폴더에서 시작합니다.

1.2 MySQL

AWS RDS 로 생성한 database-1 을 모두 사용합니다.

1.3 데이터 삭제

AWS RDS(database-1) amrbase 의 police_station 테이블의 데이터를 모두 지워줍니다.

```
% mysqldump --set-gtid-purged=OFF -h "database-1.cwzfvdk3vok.ap-northeast-2.rds.amazonaws.com" -P 3306 -u admin -p amrbase police_station > backup_police.sql
Enter password:
mysql> use amrbase;
Database changed
mysql> delete from police_station;
Query OK, 4 rows affected (0.02 sec)
```

1.4 python.ipynb 파일 생성

VSCode 에서 파일을 생성합니다.

2 Python with MySQL

Python 으로 MySQL 접속후 사용하는 방법

2.1 Install MySQL Driver - 1

- Python 에서 MySQL 을 사용하기 위해서는 먼저 MySQL Driver 를 설치한다. (eda 가상환경)

```
pip install mysql-connector-python
```

2.2 Install MySQL Driver - 2

- 설치확인

```
import mysql.connector
```

```
import mysql.connector  
✓ 0.5s
```

Python

2.3 Create Connection

MySQL 에 접속하기 위한 코드

```
mydb = mysql.connector.connect(  
    host = "<hostname>",  
    user = "<username>",  
    password = "<password>"  
)
```

2.4 Create Connection 예제 1

Local Database 연결

```
import mysql.connector

local = mysql.connector.connect(
    host = "localhost",
    user = "root",
    password = "*****"
)
```

✓ 0.4s

Python

2.5 Create Connection 예제 2

AWS RDS (database-1) 연결

```
remote = mysql.connector.connect(
    host = "database-1.cwzfvdk3vok.ap-northeast-2.rds.amazonaws.com",
    port = 3306,
    user = "admin",
    password = "*****"
)
```

✓ 0.9s

Python

2.6 Close Database

```
import mysql.connector

local = mysql.connector.connect(
    host = "<hostname>",
    user = "<username>",
    password = "<password>"
    database = "<databasename>"
)

local.close()
```

2.7 Close Database 예제 1

```
local.close()
```

✓ 0.1s

Python

2.8 Close Database 예제 2

```
remote.close()
```

✓ 0.1s

Python

2.9 Connect to Database

특정 Database 에 접속하기 위한 코드

```
import mysql.connector

mydb = mysql.connector.connect(
    host = "<hostname>",
    port = <port>,
    user = "<username>",
    password = "<password>",
    database = "<databasename>"
)
```

2.10 Create Connection 예제 1

Local MySQL 의 amrbase 에 연결

```
import mysql.connector

local = mysql.connector.connect(
    host = "localhost",
    port = 3306,
    user = "root",
    password = "*****"
    database = "amrbase"
)

local.close()
```

2.11 Create Connection 예제 2

AWS RDS (database-1)의 amrbase 에 연결

```
remote = mysql.connector.connect(
    host = "database-1.cwzevdhk3vok.ap-northeast-2.rds.amazonaws.com",
    port = 3306,
    user = "admin",
    password = "*****"
    database = "amrbase"
)

remote.close()
```

2.12 Execute SQL

Query 를 실행하기 위한 코드

```
import mysql.connector

mydb = mysql.connector.connect(
    host = "<hostname>",
    user = "<username>",
    password = "<password>"
    database = "<databasename>"
)

mycursor = mydb.cursor()
mycursor.execute(<query>);
```

2.13 Execute SQL 예제 1 - 1

테이블 생성

```
remote = mysql.connector.connect(
    host = "database-1.cwzevdhk3vok.ap-northeast-2.rds.amazonaws.com",
    port = 3306,
    user = "admin",
    password = "*****"
    database = "amrbase"
)
```



```
cur = remote.cursor()
cur.execute("CREATE TABLE sql_file (id int, filename varchar(16))")

remote.close()
```

2.14 Execute SQL 예제 1 - 2

- 결과 확인

```
mysql> desc sql_file;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int    | YES  |     | NULL    |       |
| filename | varchar(16) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

2.15 Execute SQL 예제 2 - 1

테이블 삭제

```
remote = mysql.connector.connect(
    host = "database-1.cwzevdhk3vok.ap-northeast-2.rds.amazonaws.com",
    port = 3306,
    user = "admin",
    password = "*****"
    database = "amrbase"
)

cur = remote.cursor()
cur.execute("DROP TABLE sql_file")

remote.close()
```

2.16 Execute SQL 예제 2 - 2

- 결과 확인

```
mysql> desc sql_file;
ERROR 1146 (42S02): Table 'amrbase.sql_file' doesn't exist
```

2.17 Execute SQL File 1

SQL File 을 실행하기 위한 코드

```
mydb = mysql.connector.connect(
    host = "<hostname>",
    user = "<username>",
    password = "<password>"
    database = "<databasename>"
)

mycursor = mydb.cursor()

sql = open("<filename>.sql").read()
mycursor.execute(sql)
```

2.18 Execute SQL File 1 예제 - 1

- test03.sql 생성

```
test03.sql
1 CREATE TABLE sql_file
2 (
3     id int,
4     filename varchar(16)
5 );
```

2.19 Execute SQL File 1 예제 - 2

- test03.sql 실행

```
remote = mysql.connector.connect(
    host = "database-1.cwzevdhk3vok.ap-northeast-2.rds.amazonaws.com",
    port = 3306,
    user = "admin",
```

```

        password = "*****"
        database = "amrbase"
    )

    cur = remote.cursor()
    sql = open("test03.sql").read()
    cur.execute(sql)

    remote.close()

```

2.20 Execute SQL File 1 예제 - 3

- 결과 확인

```

mysql> desc sql_file;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id     | int    | YES  |     | NULL    |       |
| filename | varchar(16) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

```

2.21 Execute SQL File 2

SQL File 내에 Query 가 여러개 존재하는 경우

```

mydb = mysql.connector.connect(
    host = "<hostname>",
    user = "<username>",
    password = "<password>",
    database = "<databasename>"
)

mycursor = mydb.cursor()

sql = open("<filename>.sql").read()
result = mycursor.execute(sql, multi=True)

```

2.22 Execute SQL File 2 예제 - 1

- test04.sql 생성

```

test04.sql
1  INSERT INTO sql_file VALUES (1, "test01.sql");
2  INSERT INTO sql_file VALUES (2, "test02.sql");
3  INSERT INTO sql_file VALUES (3, "test03.sql");
4  INSERT INTO sql_file VALUES (4, "test04.sql");

```

2.23 Execute SQL File 2 예제 - 2

- 실행 (에러)

```

remote = mysql.connector.connect(
    host = "database-1.cwzevdhk3vok.ap-northeast-2.rds.amazonaws.com",
    port = 3306,
    user = "admin",
    password = "*****"
    database = "amrbase"
)

cur = remote.cursor()
sql = open("test04.sql").read()
cur.execute(sql)

remote.close()

```

```
InterfaceError: Use multi=True when executing multiple statements
```

2.24 Execute SQL File 2 예제 - 3

- Multi=True

```

remote = mysql.connector.connect(
    host = "database-1.cwzevdhk3vok.ap-northeast-2.rds.amazonaws.com",
    port = 3306,
    user = "admin",
    password = "*****"
    database = "amrbase"
)

cur = remote.cursor()
sql = open("test04.sql").read()
for result_iterator in cur.execute(sql, multi=True):
    if result_iterator.with_rows:
        print(result_iterator.fetchall())

```

```
else:
    print(result_iterator.statement)
```

```
remote.commit()
remote.close()
```

```
INSERT INTO sql_file VALUES (1, "test01.sql")
INSERT INTO sql_file VALUES (2, "test02.sql")
INSERT INTO sql_file VALUES (3, "test03.sql")
INSERT INTO sql_file VALUES (4, "test04.sql")
```

2.25 Execute SQL File 2 예제 - 4

- 결과 확인

```
mysql> select * from sql_file;
+-----+-----+
| id   | filename |
+-----+-----+
| 1    | test01.sql |
| 2    | test02.sql |
| 3    | test03.sql |
| 4    | test04.sql |
+-----+-----+
4 rows in set (0.01 sec)
```

2.26 Fetch All

```
mycursor.execute(<query>)

result = mycursor.fetchall()
for data in result:
    print(data)
```

2.27 Fetch All 예제

celeb 테이블 조회 (읽어올 데이터 양이 많은 경우 buffered=True)

```
remote = mysql.connector.connect(
    host = "database-1.cwzevdhk3vok.ap-northeast-2.rds.amazonaws.com",
    port = 3306,
```

```

    user = "admin",
    password = "*****"
    database = "amrbase"
)

cur = remote.cursor(buffered=True)
cur.execute("SELECT * FROM sql_file")

result = cur.fetchall()
for result_iterator in result:
    print(result_iterator)

remote.close()

```

```

(1, 'test01.sql')
(2, 'test02.sql')
(3, 'test03.sql')
(4, 'test04.sql')

```

2.28 참고. 검색결과를 Pandas 로 읽기

```

import pandas as pd

df = pd.DataFrame(result)
df.head()

```

✓ 0.5s

Python

	0	1
0	1	test01.sql
1	2	test02.sql
2	3	test03.sql
3	4	test04.sql

3 Python with CSV

3.1 csv 에 있는 데이터를 Python 으로 INSERT

3.2 Read CSV

제공받은 police_station.csv 를 Pandas 로 읽어와서 데이터를 확인합니다.

```
import pandas as pd

df = pd.read_csv("police_station.csv")
df.head()
```

✓ 0.3s

Python

	서울특별시경찰청	서울시 종로구 사직로8길 31
0	서울중부경찰서	서울특별시 중구 수표로 27
1	서울종로경찰서	서울특별시 종로구 율곡로 46
2	서울남대문경찰서	서울특별시 중구 한강대로 410
3	서울서대문경찰서	서울특별시 서대문구 통일로 113
4	서울혜화경찰서	서울특별시 종로구 창경궁로 112-16

3.3 amrbase 에 연결

```
import mysql.connector

conn = mysql.connector.connect(
    host = "database-1.cwzevdhk3vok.ap-northeast-2.rds.amazonaws.com",
    port = 3306,
    user = "robot",
    password = "*****"
    database = "amrbase"
)
```

3.4 Cursor 만들기

읽어올 양이 많은 경우 cursor 생성 시 buffer 설정을 해준다.

```
cursor = conn.cursor(buffered=True)
```

✓ 0.4s

Python

3.5 INSERT 문 만들기

```
sql = "INSERT INTO police_station VALUES (%s, %s)"
```

✓ 0.3s

Python

3.6 데이터 입력

commit() 은 database 에 적용하기 위한 명령

```
for i, row in df.iterrows():
    cursor.execute(sql, tuple(row))
    print(tuple(row))
    conn.commit()
```

✓ 0.5s

Python

```
('서울중부경찰서', '서울특별시 중구 수표로 27')
('서울종로경찰서', '서울특별시 종로구 율곡로 46')
('서울남대문경찰서', '서울특별시 중구 한강대로 410')
```


3.7 결과 확인

```

cursor.execute("SELECT * FROM police_station")

result = cursor.fetchall()
for row in result:
    print(row)

```

✓ 0.4s Python

```

('서울특별시경찰청', '서울시 종로구 사직로8길 31')
('서울중부경찰서', '서울특별시 중구 수표로 27')
('서울종로경찰서', '서울특별시 종로구 율곡로 46')

```

3.8 검색결과를 Pandas 로 읽기

```

df = pd.DataFrame(result)
df

```

✓ 0.2s Python

	0	1
0	서울특별시경찰청	서울시 종로구 사직로8길 31
1	서울중부경찰서	서울특별시 중구 수표로 27
2	서울종로경찰서	서울특별시 종로구 율곡로 46
3	서울남대문경찰서	서울특별시 중구 한강대로 410

3.9 Tip

csv 한글이 깨지는 경우, encoding 값을 'euc-kr' 로 설정 (특히 우리나라 사이트에서 제공받은 csv 파일들.)

```
import pandas as pd

df = pd.read_csv('2020_crime.csv', encoding='euc-kr')
df.head()
```

✓ 0.5s

Python

	구분	죄종	발생검거	건수
0	중부	살인	발생	1
1	중부	살인	검거	1
2	중부	강도	발생	3
3	중부	강도	검거	4
4	중부	강간,추행	발생	113

4 Python with CSV 예제

crime_status 테이블에 아래 데이터를 입력하는 코드를 작성

4.1 AWS RDS(database-1) amrbase 에 접속

```
import mysql.connector

remote = mysql.connector.connect(
    host = "database-1.cwzevdhk3vok.ap-northeast-2.rds.amazonaws.com",
    port = 3306,
    user = "robot",
    password = "*****"
    database = "amrbase"
)
```

4.2 2020_crime.csv 데이터(encoding='euc-kr') 읽어오기

```
import pandas as pd

df = pd.read_csv('2020_crime.csv', encoding='euc-kr')
df.head()
```

✓ 0.2s

Python

	구분	죄종	발생검거	건수
0	중부	살인	발생	1
1	중부	살인	검거	1
2	중부	강도	발생	3
3	중부	강도	검거	4
4	중부	강간,추행	발생	113

4.3 INSERT 쿼리 작성

```
sql = """INSERT INTO crime_status VALUES ("2020", %s, %s, %s, %s)"""
cursor = conn.cursor(buffered=True)
```

✓ 0.3s

Python

4.4 데이터를 crime_status 테이블에 INSERT

```
for i, row in df.iterrows():
    cursor.execute(sql, tuple(row))
    print(tuple(row))
    conn.commit()
```

✓ 5.2s

Python

```
('중부', '살인', '발생', 1)
('중부', '살인', '검거', 1)
('중부', '강도', '발생', 3)
```

4.5 crime_status 테이블의 데이터 조회

```
cursor.execute("SELECT * FROM crime_status")

result = cursor.fetchall()
for row in result:
    print(row)
```

✓ 0.5s

Python

```
(2020, '중부', '살인', '발생', 1)
(2020, '중부', '살인', '검거', 1)
(2020, '중부', '강도', '발생', 3)
```

4.6 조회한 결과를 Pandas 로 변환해서 확인

```
df = pd.DataFrame(result)
df.head()
```

✓ 0.4s

Python

	0	1	2	3	4
0	2020	중부	살인	발생	1
1	2020	중부	살인	검거	1
2	2020	중부	강도	발생	3
3	2020	중부	강도	검거	4
4	2020	중부	강간,추행	발생	113

5 연습

1. AWS RDS (database-1) amrbase 접속
2. cctv Table 생성
3. cctv 데이터를 pandas 로 읽어오기 (Seoul_CCTV.csv)
4. 데이터를 cctv 테이블에 INSERT
5. cctv 테이블의 데이터 조회
6. pandas 로 변환하여 출력