

# 파이썬 PyMySQL



# PyMySQL

- MySQL 데이터베이스를 사용하기 위해 우선 Python DB API 표준을 따르는 MySQL DB 모듈
  - pip install PyMySQL
  - 1. PyMySQL 모듈을 import
  - 2. MySQL Connection 연결  
`conn = pymysql.connect(host='localhost', user='root', password='root1234', db='mydb', charset='utf8')`
  - 3. Connection 으로부터 Cursor 생성  
`curs = conn.cursor()`
  - 4. SQL문 실행  
`sql = "select * from movie"`  
`curs.execute(sql)`
  - 5. 데이터를 서버로부터 가져온 후, Fetch 된 데이터를 사용  
`rows = curs.fetchall()`
  - 6. Connection 닫기



# PyMySQL - 쿼리

```
1  #1.PyMySQL 모듈을 import
2  import pymysql
3
4  #2.MySQL Connection 연결
5  conn = pymysql.connect(host='localhost', user='root', password='root1234',
6  | | | | | db='mydb', charset='utf8')
7
8  #3.Connection 으로부터 Cursor 생성 => Dictoionary Cursor 생성
9  curs = conn.cursor()
10
11 #4.SQL문 실행
12 sql = "select * from movie"
13 curs.execute(sql)
14
15 #5.데이터를 서버로부터 가져온 후, Fetch 된 데이터를 사용
16 rows = curs.fetchall()
17 print(rows)      # 전체 rows => 튜플
18
19 for row in rows :
20     for item in row :
21         print(item, end=' ')
22     print()
23
24 #6.Connection 닫기
25 conn.close()
```

((1, '스파이더맨: 홈 커밍', 4, 'http://t1.daumcdn.net/movie/4f9750f09dac0f1b3659ae03cfe9ed7938be8d30', 8.1, '2017.07.05', 2017), (3, '신과함께-인과 연', 1, 'http://t1.daumcdn.net/movie/e78737f6be1573f673b561f3fc0b32a8cffe820', 6.8, '2018.08.01', 2018), (4, '어벤저스: 인피니티 워', 2, 'http://t1.daumcdn.net/movie/dd84b905224c91225aa2a15203aba3a354197c1d', 7.6, '2018.04.25', 2018), (5, '보헤미안 랍소디', 3, 'http://t1.daumcdn.net/movie/ce3cd6a875284e8b96414ef3696756a11544772388211', 9.0, '2018.10.31', 2018), (6, '극한직업', 1, 'http://t1.daumcdn.net/movie/4e00e81f2b6f4d2eb65b3387240cc3c01547608409838', 7.4, '2019.01.23', 2019), (7, '어벤저스: 엔드게임', 2, 'http://t1.daumcdn.net/movie/5574fb2c20c844629aa9ad1d6043ee851555464908641', 7.8, '2019.04.24', 2019))

1 스파이더맨: 홈 커밍 4 http://t1.daumcdn.net/movie/4f9750f09dac0f1b3659ae03cfe9ed7938be8d30 8.1 2017.07.05 2017  
3 신과함께-인과 연 1 http://t1.daumcdn.net/movie/e78737f6be1573f673b561f3fc0b32a8cffe820 6.8 2018.08.01 2018  
4 어벤저스: 인피니티 워 2 http://t1.daumcdn.net/movie/dd84b905224c91225aa2a15203aba3a354197c1d 7.6 2018.04.25 2018  
5 보헤미안 랍소디 3 http://t1.daumcdn.net/movie/ce3cd6a875284e8b96414ef3696756a11544772388211 9.0 2018.10.31 2018  
6 극한직업 1 http://t1.daumcdn.net/movie/4e00e81f2b6f4d2eb65b3387240cc3c01547608409838 7.4 2019.01.23 2019  
7 어벤저스: 엔드게임 2 http://t1.daumcdn.net/movie/5574fb2c20c844629aa9ad1d6043ee851555464908641 7.8 2019.04.24 2019

# PyMySQL – 쿼리2

```
1  #1.PyMySQL 모듈을 import
2  import pymysql
3
4  #2.MySQL Connection 연결
5  conn = pymysql.connect(host='localhost', user='root', password='root1234',
6  | | | | | db='mydb', charset='utf8')
7
8  #3.Connection 으로부터 Cursor 생성 => Dictoionary Cursor 생성
9  curs = conn.cursor(pymysql.cursors.DictCursor)
10
11 #4.SQL문 실행
12 sql = "select * from movie"
13 curs.execute(sql)
14
15 #5.데이터를 서버로부터 가져온 후, Fetch 된 데이터를 사용
16 rows = curs.fetchall()
17 print(rows)      # 전체 rows =>튜플
18
19 for row in rows :
20     for key, value in row.items() :
21         print(row[key], end=" ")
22     print()
23
24 #6.Connection 닫기
25 conn.close()
```

[[{'id': 1, 'mtitle': '스파이더맨: 홈 커밍', 'rank': 4, 'imgurl': 'http://t1.daumcdn.net/movie/4f9750f09dac0f1b3659ae03cfe9ed7938be8d30', 'star': 8.1, 'cdate': '2017.07.05', 'cyy': 2017}, {'id': 3, 'mtitle': '신과함께-인과 연', 'rank': 1, 'imgurl': 'http://t1.daumcdn.net/movie/e78737feb e1573f673b561f3fcb0b32a8cffe820', 'star': 6.8, 'cdate': '2018.08.01', 'cyy': 2018}, {'id': 4, 'mtitle': '어벤저스: 인피니티 워', 'rank': 2, 'imgurl': 'http://t1.daumcdn.net/movie/dd84b905224c91225aa2a15203aba3a354197c1d', 'star': 7.6, 'cdate': '2018.04.25', 'cyy': 2018}, {'id': 5, 'mtitle': '보헤미안 랑소니', 'rank': 3, 'imgurl': 'http://t1.daumcdn.net/movie/ce3cd6a87528408b96414ef3696756a11544772388211', 'star': 9.0, 'cdate': '2018.10.31', 'cyy': 2018}, {'id': 6, 'mtitle': '극한직업', 'rank': 1, 'imgurl': 'http://t1.daumcdn.net/movie/4e00e81f2b6f4d2eb65b3387240cc3c01547608409838', 'star': 7.4, 'cdate': '2019.01.23', 'cyy': 2019}, {'id': 7, 'mtitle': '어벤저스: 엔드게임', 'rank': 2, 'imgurl': 'http://t1.daumcdn.net/movie/5574fb2c20c844629a9ad1d6043ee851555464908641', 'star': 7.8, 'cdate': '2019.04.24', 'cyy': 2019}]]

1 스파이더맨: 홈 커밍 4 http://t1.daumcdn.net/movie/4f9750f09dac0f1b3659ae03cfe9ed7938be8d30 8.1 2017.07.05 2017  
3 신과함께-인과 연 1 http://t1.daumcdn.net/movie/e78737feb e1573f673b561f3fcb0b32a8cffe820 6.8 2018.08.01 2018  
4 어벤저스: 인피니티 워 2 http://t1.daumcdn.net/movie/dd84b905224c91225aa2a15203aba3a354197c1d 7.6 2018.04.25 2018  
5 보헤미안 랑소니 3 http://t1.daumcdn.net/movie/ce3cd6a87528408b96414ef3696756a11544772388211 9.0 2018.10.31 2018  
6 극한직업 1 http://t1.daumcdn.net/movie/4e00e81f2b6f4d2eb65b3387240cc3c01547608409838 7.4 2019.01.23 2019  
7 어벤저스: 엔드게임 2 http://t1.daumcdn.net/movie/5574fb2c20c844629a9ad1d6043ee851555464908641 7.8 2019.04.24 2019

# PyMySQL – INSERT문 처리

```
#1.PyMySQL 모듈을 import
import pymysql
```

```
#2.MySQL Connection 연결
conn = pymysql.connect(host='localhost', user='root', password='root1234',
| | | | | db='mydb', charset='utf8')
```

```
#3.Connection 으로부터 Cursor 생성
curs = conn.cursor()
```

```
#4.SQL문 실행
```

```
sql = """INSERT INTO movie (mtitle, rank, imgurl, star, cdate, cyy)
| | | | | values
| | | | | ('알라딘', 4, 'http://t1.daumcdn.net/movie/3673a8a0c5ff4f5c8c25cc959fd6985b1558662957684', 8.4, '2019.05.23', 2019)"""

curs.execute(sql)
```

```
#5.Commit하지 않으면, 테이블의 데이터는 변경되지 않는다는 점
conn.commit() # 전체 rows => 튜플
```



```
#4.SQL문 실행
```

```
data = (
| ('신과함께-죄와 벌', 5, 'http://t1.daumcdn.net/movie/ff9d430c0d2df2a1c659ccba8b621ad2251f6f02', 7.0, '2017.12.20', 2018),
| ('쥬라기 월드: 폴른 킹덤', 6, 'http://t1.daumcdn.net/movie/d9b4fd04ea18b4db74f3ddfc9ebc3c22e8a0cd1b', 7.1, '2018.06.06', 20
| ('앳맨과 와스프', 7, 'http://t1.daumcdn.net/movie/f60e1438ee6ca99476e393258c41e11bf6e28138', 7.4, '2018.07.04', 2018),
| ('안시성', 8, 'http://t1.daumcdn.net/movie/8d6743476fdf438e846ccab9ac59b1491537322852509', 7.6, '2018.09.19', 2018)
| )
sql = """INSERT INTO movie (mtitle, rank, imgurl, star, cdate, cyy)
| | | | | values
| | | | | (%s, %s, %s, %s,%s,%s)"""

curs.executemany(sql, data)
```

```
#5.Commit하지 않으면, 테이블의 데이터는 변경되지 않는다는 점
conn.commit() # 전체 rows => 튜플
```

# PyMySQL – UPDATE/DELETE문 처리

```
#1.PyMySQL 모듈을 import
import pymysql
```

```
#2.MySQL Connection 연결
```

```
conn = pymysql.connect(host='localhost', user='root', password='root1234',
| | | | | db='mydb', charset='utf8')
```

```
#3.Connection 으로부터 Cursor 생성
```

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

```
#4.SQL문 실행
```

```
sql = """UPDATE movie SET
|         mtitle = %s
|         WHERE id = %s """
```

```
curs.execute(sql,('알라딘2', 10))
```

```
#4.SQL문 실행
```

```
sql = """DELETE FROM movie
|         WHERE id = %s """
```

```
curs.execute(sql, 10)
```

```
#5.Commit하지 않으면, 테이블의 데이터는 변경되지 않는다는 점
```

```
conn.commit() # 전체 rows => 튜플
```



# PyMySQL Connection Leak

- **Connection Leak**

- SQL Connection을 열고 프로그램 중간에서 에러가 발생하면, Connection은 그대로 열려 있는 상태로 있을 수 있는데 오픈되어 있는 Connection이 증가하여 새로운 Connection을 오픈할 수 없게 되는 경우

```
try:
    문장1
    문장2
except:
    예외처리
finally:
    마지막에 항상 수행
```

- **예외 처리**

- try 블록 수행 중 오류가 발생하면 except 블록이 수행
- 하지만 try 블록에서 오류가 발생하지 않는다면 except 블록은 수행되지 않음
- finally절은 try문 수행 도중 예외 발생 여부에 상관없이 항상 수행

- **with 구문**

- 파일을 다룰 때 with 블록을 통해 명시적으로 **close()** 메소드를 호출하지 않고도 파일을 닫을 수 있음



# PyMySQL Connection Leak

```
#1.PyMySQL 모듈을 import
import pymysql
```

```
#2.MySQL Connection 연결
conn = pymysql.connect(host='localhost', user='root', password='root1234',
                        db='mydb', charset='utf8')
```

```
#3.Connection 으로부터 Cursor 생성
```

```
try :
```

```
    with conn.cursor() as curs :
        #4.SQL문 실행
        sql = """INSERT INTO movie (mtitle, rank, imgurl, star, cdate, cyy)
        values
        (%s, %s, %s, %s, %s, %s)"""

        curs.execute(sql, ('알라딘', 4, 'http://t1.daumcdn.net/movie/3673a8a0c5ff4f5c8c25cc95'
```

```
#5.Commit하지 않으면, 테이블의 데이터는 변경되지 않는다는 점
conn.commit()    # 전체 rows =>튜플
```

```
#결과 확인
```

```
    with conn.cursor() as curs :
        sql = "select * from movie"
        curs.execute(sql)

        rows = curs.fetchall()
        print(rows)    # 전체 rows =>튜플

        for row in rows :
            for item in row :
                print(item, end=" ")
            print()
```

```
finally:
```

```
#6.Connection 닫기
conn.close()
```





# 실습

- 네이버 최신 1000건의 평점 데이터를 크롤링하여 테이블을 만들고 리뷰수가 10회 이상이면서 평점이 8이상인 영화를 추천하시오.