

# **MySQL DB 연동**

## **- 데이터 다루기**

# pymysql 모듈 이해

---

## pymysql 라이브러리 소개 및 설치

- mysql을 python에서 사용할 수 있는 라이브러리  
( pymysql 라이브러리 이외에도 MySQLdb(Mysql-pytion),  
MySQL connector 등 다양한 라이브러리 존재)
- 이 중에서 설치가 가장 쉬운 라이브러리
- !pip install pymysql

# pymysql 모듈 이해

---

## 일반적인 mysql 핸들링 코드 작성 순서

- PyMySQL 모듈 import
- pymysql.**connect()** 메소드를 사용하여 MySQL에 연결
- 호스트명, 포트, 로그인, 암호, 접속할 DB 등을 파라미터로 지정
- MySQL 접속이 성공하면, Connection 객체로부터 **cursor()** 메서드를 호출하여 Cursor 객체를 가져옴
- Cursor 객체의 **execute()** 메서드를 사용하여 SQL 문장을 DB 서버에 전송

# pymysql 모듈 이해

---

## 일반적인 mysql 핸들링 코드 작성 순서

- SQL 쿼리의 경우 Cursor 객체의 **fetchall()**, **fetchone()**, **fetchmany()** 등의 메서드를 사용하여 서버로부터 가져온 데이터를 코드에서 활용
- 삽입, 갱신, 삭제 등의 DML(Data Manipulation Language) 문장을 실행하는 경우, INSERT/UPDATE/DELETE 후 Connection 객체의 **commit()** 메서드를 사용하여 데이터를 확정
- Connection 객체의 **close()** 메서드를 사용하여 DB 연결을 닫음

# pymysql 모듈 이해

---

## PyMySQL 모듈 import

- !pip install pymysql
- PyMySQL 모듈 import

```
In [1]: !pip install pymysql
```

```
Requirement already satisfied: pymysql in c:\programdata\anaconda3\lib\site-packages (1.0.2)
```

```
In [2]: import pymysql
```

# pymysql 모듈 이해

---

## PyMySQL 모듈

- pymysql.connect() 메소드를 사용하여 MySQL에 연결
- 호스트명, 포트, 로그인, 암호, 접속할 DB 등을 파라미터로 지정
- 주요 파라미터
  - host : 접속할 mysql server 주소
  - port : 접속할 mysql server 의 포트 번호
  - user : mysql ID
  - passwd : mysql ID의 암호
  - db : 접속할 데이터베이스
  - charset='utf8' : mysql에서 select하여 데이터를 가져올 때 한글이 깨질 수 있으므로 연결 설정에 넣어줌

# pymysql 모듈 이해

---

## PyMySQL 모듈

- pymysql.connect() 메소드를 사용하여 MySQL에 연결

```
In [4]: # dbms 연결 코드
        db = pymysql.connect(host='localhost',port=3306,user='root',passwd='toor',charset='utf8')

In [5]: db

Out [5]: <pymysql.connections.Connection at 0x1d40e2a2af0>
```

# pymysql 모듈 이해

---

## PyMySQL 모듈

- MySQL 접속이 성공하면, Connection 객체로부터 `cursor()` 메서드를 호출하여 Cursor 객체를 가져옴
- Cursor 객체의 `execute()` 메서드를 사용하여 SQL 문장을 DB 서버에 전송
- DB 생성
  - Cursor Object 가져오기: `cursor = db.cursor()`
  - SQL 실행하기: `cursor.execute(SQL)`
  - 실행 mysql 서버에 확정 반영하기: `db.commit()`



# pymysql 모듈 이해

## PyMySQL 모듈

- MySQL 접속이 성공

```
In [8]: cursor=db.cursor()
```

```
In [17]: sql = "create database IF NOT EXISTS ecommerce"
         cursor.execute(sql)
```

```
Out[17]: 1
```

```
In [18]: sql = "show databases"
         cursor.execute(sql)
         result = cursor.fetchall()
         result
```

```
Out[18]: (('classicmodels',),
          ('ecommerce',),
          ('employees',),
          ('information_schema',),
          ('mysql',),
          ('performance_schema',),
          ('sakila',),
          ('shopdb',),
          ('sqldb',),
          ('sys',),
          ('world',))
```

# pymysql 모듈 이해

## PyMySQL 모듈

- MySQL 접속이 성공

```
In [8]: cursor=db.cursor()
```

```
In [17]: sql = "create database IF NOT EXISTS ecommerce"
         cursor.execute(sql)
```

```
Out[17]: 1
```

```
In [18]: sql = "show databases"
         cursor.execute(sql)
         result = cursor.fetchall()
         result
```

```
Out[18]: (('classicmodels',),
          ('ecommerce',),
          ('employees',),
          ('information_schema',),
          ('mysql',),
          ('performance_schema',),
          ('sakila',),
          ('shopdb',),
          ('sqldb',),
          ('sys',),
          ('world',))
```

# pymysql 모듈 이해

---

## PyMySql 모듈

- DB 변경 & 현재 DB 확인

```
In [19]: # db 변경  
sql = "use ecommerce"  
cursor.execute(sql)
```

```
Out [19]: 0
```

```
In [20]: # 현재 db 확인  
sql = "select database()"   
cursor.execute(sql)  
result = cursor.fetchone()  
result
```

```
Out [20]: ('ecommerce',)
```

# pymysql 모듈 이해

---

## PyMySQL 모듈

- 테이블 생성 SQL 실행하기: `cursor.execute(SQL)` 실행
- mysql 서버에 확정 반영하기: `db.commit()`

```
In [21]: cursor
```

```
Out [21]: <pymysql.cursors.Cursor at 0x1d40e2ba5e0>
```

- `cursor` 는 control structure of database 임 (연결된 객체)

# pymysql 모듈 이해

---

## PyMySQL 모듈

- SQL 실행 (Cursor 객체의 execute() 메서드를 사용하여 INSERT, UPDATE 혹은 DELETE 문장을 DB 서버에 보냄)

```
sql = """
CREATE TABLE product (
    PRODUCT_CODE VARCHAR(20) NOT NULL,
    TITLE VARCHAR(200) NOT NULL,
    ORI_PRICE INT,
    DISCOUNT_PRICE INT,
    DISCOUNT_PERCENT INT,
    DELIVERY VARCHAR(2),
    PRIMARY KEY(PRODUCT_CODE)
);
"""
```

# pymysql 모듈 이해

## PyMySQL 모듈

- SQL 실행 (Cursor 객체의 execute() 메서드를 사용하여 INSERT, UPDATE 혹은 DELETE 문장을 DB 서버에 보냄)

```
In [23]: cursor.execute(sql)
```

```
Out [23]: 0
```

```
In [24]: sql = 'show tables'
          cursor.execute(sql)
          result = cursor.fetchall()
          result
```

```
Out [24]: (('product',),)
```

- 삽입, 갱신, 삭제 등이 모두 끝났으면 Connection 객체의 commit() 메서드를 사용하여 데이터를 Commit

```
In [25]: db.close() #연결종료
```

# 패턴으로 익히는 pymysql

---

## 패턴으로 익히는 pymysql

1. 라이브러리 가져오기
2. 접속하기
3. 커서 가져오기
4. SQL 구문 만들기(CRUD SQL 구문 등)
5. SQL 구문 실행하기
6. DB에 Complete 하기
7. DB 연결 닫기

# 패턴으로 익히는 pymysql

## 패턴으로 익히는 pymysql

```
In [37]: # 1. 라이브러리 가져오기
import pymysql

# 2. 접속하기
db = pymysql.connect(host='localhost', port=3306, user='root', passwd='toor', db='ecommerce', charset='utf8')

In [38]: # 3. 커서 가져오기
cursor = db.cursor()

In [39]: ## 4. SQL 구문 만들기 (CRUD SQL 구문 등)
sql = '''
    CREATE TABLE product2 (
        PRODUCT_CODE VARCHAR(20) NOT NULL,
        TITLE VARCHAR(200) NOT NULL,
        ORI_PRICE INT,
        DISCOUNT_PRICE INT,
        DISCOUNT_PERCENT INT,
        DELIVERY VARCHAR(2),
        PRIMARY KEY(PRODUCT_CODE)
    );
    ...
```



# 패턴으로 익히는 pymysql

---

## 패턴으로 익히는 pymysql

In [40]:

```
# 5. SQL 구문 실행하기  
cursor.execute(sql)
```

Out [40]: 0

In [41]:

```
# 6. DB에 Complete 하기  
db.commit()
```

In [42]:

```
# 7. DB 연결 닫기  
db.close()
```

# 패턴으로 익히는 pymysql

---

## 데이터 삽입(INSERT)

- Cursor Object 가져오기: `cursor = db.cursor()`
- SQL 실행하기: `cursor.execute(SQL)`
- 실행 mysql 서버에 확정 반영하기: `db.commit()`

# 패턴으로 익히는 pymysql

---

## 데이터 삽입(INSERT)

1. 라이브러리 가져오기
2. 접속하기
3. 커서 가져오기
4. SQL 문 만들기 : INSERT INTO 테이블명 VALUES()
5. SQL 구문 실행하기
6. DB에 Complete 하기
7. DB 연결 닫기

# 패턴으로 익히는 pymysql

## 데이터 삽입(INSERT)

```
In [44]: # 1. 라이브러리 가져오기
import pymysql

# 2. 접속하기
db = pymysql.connect(host='localhost', port=3306, user='root', passwd='toor', db='ecommerce', charset='utf8')

In [45]: # 3. 커서 가져오기
cursor = db.cursor()

In [46]: for index in range(10):
    product_code = 216573140 + index + 1
    # print(product_code)
    sql = """INSERT INTO product2 VALUES(
    '"" + str(product_code) + ""', '스위트바니 여름신상5900원~홍원피스티셔츠/긴팔/반팔', 23000, 6900, 70, 'F'); """
    print (sql)
    cursor.execute(sql)
```

# 패턴으로 익히는 pymysql

## 데이터 삽입(INSERT)

```
'216573145', '스워트바니 여름신상5900원~통원피스티셔츠/긴팔/반팔', 23000, 6900, 70, 'F');
INSERT INTO product2 VALUES(
    '216573146', '스워트바니 여름신상5900원~통원피스티셔츠/긴팔/반팔', 23000, 6900, 70, 'F');
INSERT INTO product2 VALUES(
    '216573147', '스워트바니 여름신상5900원~통원피스티셔츠/긴팔/반팔', 23000, 6900, 70, 'F');
INSERT INTO product2 VALUES(
    '216573148', '스워트바니 여름신상5900원~통원피스티셔츠/긴팔/반팔', 23000, 6900, 70, 'F');
INSERT INTO product2 VALUES(
    '216573149', '스워트바니 여름신상5900원~통원피스티셔츠/긴팔/반팔', 23000, 6900, 70, 'F');
INSERT INTO product2 VALUES(
    '216573150', '스워트바니 여름신상5900원~통원피스티셔츠/긴팔/반팔', 23000, 6900, 70, 'F');
```

```
In [47]: db.commit() # 입력된 데이터를 db서버에 확정 시키기
```

```
In [48]: db.close()
```

# 패턴으로 익히는 pymysql

---

## 데이터 조회(SELECT)

- Cursor Object 가져오기: `cursor = db.cursor()`
- SQL 실행하기: `cursor.execute(SQL)`
- mysql 서버로부터 데이터 가져오기: fetch 메서드 사용
  - `fetchall()`: Fetch all the rows
  - `fetchmany(size=None)`: Fetch several rows
  - `fetchone()`: Fetch the next row

# 패턴으로 익히는 pymysql

## 데이터 조회(SELECT)

- Cursor Object 가져오기: `cursor = db.cursor()`
- SQL 실행하기: `cursor.execute(SQL)`

```
In [50]: import pymysql

db = pymysql.connect(host='localhost', port=3306, user='root', passwd='toor', db='ecommerce', charset='utf8')
cursor = db.cursor()
```

```
In [53]: sql = "select * from product2"
cursor.execute(sql)
```

```
Out[53]: 10
```

```
In [54]: result = cursor.fetchone() # 현재 커서를 다음 레코드로 이동시키고 해당 레코드를 반환
result
```

```
Out[54]: ('216573141', '스위트바니 여름신상5900원~통원피스티셔츠/긴팔/반팔', 23000, 6900, 70, 'F')
```

# 패턴으로 익히는 pymysql

## 데이터 조회(SELECT)

- Cursor Object 가져오기: `cursor = db.cursor()`
- SQL 실행하기: `cursor.execute(SQL)`

```
In [55]: result=cursor.fetchall()
```

```
In [56]: result
```

```
Out [56]: (('216573142', '스위트바니 여름신상5900원~통원피스티셔츠/긴팔/반팔', 23000, 6900, 70, 'F'),  
          ('216573143', '스위트바니 여름신상5900원~통원피스티셔츠/긴팔/반팔', 23000, 6900, 70, 'F'),  
          ('216573144', '스위트바니 여름신상5900원~통원피스티셔츠/긴팔/반팔', 23000, 6900, 70, 'F'),  
          ('216573145', '스위트바니 여름신상5900원~통원피스티셔츠/긴팔/반팔', 23000, 6900, 70, 'F'),  
          ('216573146', '스위트바니 여름신상5900원~통원피스티셔츠/긴팔/반팔', 23000, 6900, 70, 'F'),  
          ('216573147', '스위트바니 여름신상5900원~통원피스티셔츠/긴팔/반팔', 23000, 6900, 70, 'F'),  
          ('216573148', '스위트바니 여름신상5900원~통원피스티셔츠/긴팔/반팔', 23000, 6900, 70, 'F'),  
          ('216573149', '스위트바니 여름신상5900원~통원피스티셔츠/긴팔/반팔', 23000, 6900, 70, 'F'),  
          ('216573150', '스위트바니 여름신상5900원~통원피스티셔츠/긴팔/반팔', 23000, 6900, 70, 'F'))
```

```
In [57]: db.close()
```



# 패턴으로 익히는 pymysql

---

## 데이터 수정(UPDATE)

- Cursor Object 가져오기: `cursor = db.cursor()`
- SQL 실행하기: `cursor.execute(SQL)`
- 실행 mysql 서버에 확정 반영하기: `db.commit()`

# 패턴으로 익히는 pymysql

---

## 데이터 수정(UPDATE)

1. 라이브러리 가져오기
2. 접속하기
3. 커서 가져오기
4. SQL 구문 만들기
5. SQL 구문 실행하기
- 6.commit 하기
7. Update 확인
8. close()

# 패턴으로 익히는 pymysql

## 데이터 수정(UPDATE)

```
In [74]: # 1. 라이브러리 가져오기
import pymysql

# 2. 접속하기
db = pymysql.connect(host='localhost', port=3306, user='root', passwd='toor', db='ecommerce', charset='utf8')

# 3. 커서 가져오기
cursor = db.cursor()

In [75]: # 4. SQL 구문 만들기
sql = """
UPDATE product2 SET
    TITLE='하늘하늘 원피스 썸머 스페셜 가디건',
    ORI_PRICE=33000,
    DISCOUNT_PRICE=9900,
    DISCOUNT_PERCENT=70
WHERE PRODUCT_CODE='216573141'
"""
```

# 패턴으로 익히는 pymysql

## 데이터 수정(UPDATE)

```
# 5. sql 구문 실행하기  
cursor.execute(sql)
```

```
# 6. commit 하기  
db.commit()
```

```
In [76]: # 7.update확인  
sql = "select * from product2 where product_code='216573141'"  
  
cursor.execute(sql)  
  
result = cursor.fetchone()  
  
result
```

```
Out[76]: ('216573141', '하늘하늘 원피스 썸머 스페셜 가디건', 33000, 9900, 70, 'F')
```

```
In [77]: db.close()
```

# 패턴으로 익히는 pymysql

---

## 데이터 삭제(DELETE)

- Cursor Object 가져오기: `cursor = db.cursor()`
- SQL 실행하기: `cursor.execute(SQL)`
- 실행 mysql 서버에 확정 반영하기: `db.commit()`

# 패턴으로 익히는 pymysql

---

## 데이터 삭제(DELETE)

1. 라이브러리 가져오기
2. 접속하기
3. 커서 가져오기
4. SQL 구문 만들기
5. SQL 구문 실행하기
6. commit 하기
7. close()

# 패턴으로 익히는 pymysql

## 데이터 삭제(DELETE)

```
In [79]: import pymysql

db = pymysql.connect(host='localhost', port=3306, user='root', passwd='toor', db='ecommerce', charset='utf8')

cursor = db.cursor()

In [80]: sql = "DELETE FROM product WHERE PRODUCT_CODE='216573141'"
cursor.execute(sql)
db.commit()
db.close()
```

# 패턴으로 익히는 pymysql

## 데이터 삭제(DELETE)

```
In [82]: db = pymysql.connect(host='localhost', port=3306, user='root', passwd='toor', db='ecommerce', charset='utf8')

        cursor = db.cursor()

        SQL = """select * FROM product WHERE PRODUCT_CODE='216573141'"""

        cursor.execute(SQL)

        result = cursor.fetchone()
        print(result)

        db.commit()

        db.close()
```

None



# 정리

---

## 정리

- pymysql 모듈 이해
- 패턴으로 익히는 pymysql
- DB연동하여 데이터 삽입(INSERT)
- DB연동하여 데이터 조회(SELECT)
- DB연동하여 데이터 수정(UPDATE)
- DB연동하여 데이터 삭제(DELETE)