0. pymysql 라이브러리 설치 및 import

[mysql 참고]

공식 문서 : https://pypi.org/project/PyMySQL/#id3 유용한 블로그 : https://www.fun-coding.org/mysql_basic6.html

In [4]:
```
# 설치 안되어 있으면 설치 하기(python mysql 라이브러리)
!python -m pip install PyMySQL
```

```
Requirement already satisfied: PyMySQL in ./venv/lib/python3.8/site-packages (1.0.2)
```

In [1]:
```
import pymysql
```

1. DB연결(id, pw 잘 관리하기)

In [28]:
```
conn = pymysql.connect(host='localhost', port=3306, user='rapa01', passwd='1234', db='csm_db
```

2. DB cursor 생성

셔틀버스와 같은것,

In [29]:
```
cur = conn.cursor()
```

In [30]:
```
cur.execute("DROP TABLE items")
```

Out[30]:
```
0
```

3. sql 명령 실행

In [31]:
```
CREATE_SQL ="""
CREATE TABLE IF NOT EXISTS items(
    id integer primary key auto_increment,
    code text not null,
    name text not null,
    price integer null
)

"""
cur.execute(CREATE_SQL);
```

In [32]:
```
INSERT_SQL = "INSERT INTO items(code, name, price) VALUES (%s, %s, %s);"

cur.execute(INSERT_SQL, ('A0001', '게이밍 키보드', 38000))
```

Out[32]:
```
1
```

In [33]:
```
# 여러개의 데이터를 한꺼번에 넣기
data = (
    ('A0002', '에어컨 30평형', 2000000),
    ('A0003', '최신형 스마트폰', 1000000),
    ('A0004', '최신형 노트북', 500000)
)

cur.executemany(INSERT_SQL, data)
```

Out[33]:
```
3
```

In [34]:
```
# SELECT문(R)
SELECT_SQL1 = "SELECT * FROM items;"

SELECT_SQL2 = "SELECT * FROM items LIMIT 2;"

cur.execute(SELECT_SQL2)

rows = cur.fetchall()
for row in rows:
    print(row)
print('========')

# 현재에서 4개 가져오기
# print('fetch all -> \n', cur.fetchmany(size=4))
```

```
(1, 'A0001', '게이밍 키보드', 38000)
(2, 'A0002', '에어컨 30평형', 2000000)
========
```

In [46]:
```
cur.execute("select * from online_order limit 10;")
rows = cur.fetchall()
for row in rows:
    print(row)
```

```
(20210601, 82984454, 507734, 1, 20000, 1, 20000, 8000, 1000, 7000)
(20210601, 83086659, 280456, 1, 20000, 1, 20000, 8000, 1000, 7000)
(20210601, 83432330, 401841, 1, 20000, 1, 20000, 8000, 1000, 7000)
(20210601, 83515822, 708361, 1, 20000, 1, 20000, 8000, 1000, 7000)
(20210601, 83575844, 354070, 1, 20000, 1, 20000, 8000, 1000, 7000)
(20210601, 17604107, 832796, 1, 20000, 2, 40000, 16000, 2000, 14000)
(20210601, 37453043, 514972, 1, 20000, 2, 40000, 16000, 2000, 14000)
(20210601, 48290697, 191515, 1, 20000, 2, 40000, 16000, 2000, 14000)
(20210601, 63666257, 787667, 1, 20000, 2, 40000, 16000, 2000, 14000)
(20210601, 96701451, 500786, 1, 20000, 2, 40000, 16000, 2000, 14000)
```

In [55]:
```
slt_sql = """select ui.gender, sum(gmv) as tot_gmv, count(distinct oo.userid) as user_cnt
        from online_order oo
        join user_info ui on oo.userid = ui.userid
        group by 1
        order by 2 desc;
        """
cur.execute(slt_sql)
rows = cur.fetchall()
for row in rows:
    print(row)
```

```
('F', Decimal('125819000'), 4609)
('M', Decimal('345000'), 13)
```

In [65]:

```
# UPDATE문(U)
price = 1500000
code = 'A0001'
cur.execute("UPDATE items set price = %s WHERE code=%s;", (price, code))
```

Out[65]: 0

In [62]:
```
# DELETE 문(D)
DELETE_SQL = "DELETE FROM items WHERE code = 'A00001'"
cur.execute(DELETE_SQL)
```

Out[62]: 0

4. commit, rollback 실행

- 승인(=db 적용), 취소
- INSERT, UPDATE, DELETE는 반드시 commit()을 해야 실제 데이터베이스에 반영됨.

In [22]:
```
#conn.rollback()
conn.commit()
```

5. DB close()

In [23]:
```
conn.close()
```

In [ ]:
```

```