Mysql 도커로 하기

0. 준비하기

docker run -d -p 3306:3306 --name some-mysql -e MYSQL_ROOT_PASSWORD=my-secret-pw -d mysql:latest

docker exec -it some-mysql bash

1. mysql 실행하기

mysql -u root -p

(password: my-secret-pw)

사용자 권한 추가

`mysql> create user user2@localhost identified by 'test';`

`mysql> create user user2@'%' identified by 'test';`

```
mysql> create user user2@localhost identified by 'test';
Query OK, 0 rows affected (0.01 sec)

mysql> create user user2@'%' identified by 'test';
Query OK, 0 rows affected (0.01 sec)

mysql> select user, host from mysql.user;
+------------------+-----------+
| user             | host      |
+------------------+-----------+
| root             | %         |
| user1            | %         |
| user2            | %         |
| mysql.infoschema | localhost |
| mysql.session    | localhost |
| mysql.sys        | localhost |
| root             | localhost |
| user1            | localhost |
| user2            | localhost |
+------------------+-----------+
9 rows in set (0.00 sec)
```

데이터베이스 생성

`mysql> create database flask_test;`

```
mysql> create database flask_test;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| flask_test         |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)
```

테이블만들기

- User 테이블

```
mysql> create table User(
    -> id INT NOT NULL
    -> ,
    -> usermail VARCHAR(20) NOT NULL UNIQUE,
    -> email VARCHAR(120) NOT NULL UNIQUE,
    -> image_file VARCHAR(20),
    -> password VARCHAR(60) NOT NULL,
    -> PRIMARY KEY(id)
    -> );
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> DESC User
    -> ;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| id         | int          | NO   | PRI | NULL    |       |
| usermail   | varchar(20)  | NO   | UNI | NULL    |       |
| email      | varchar(120) | NO   | UNI | NULL    |       |
| image_file | varchar(20)  | YES  |     | NULL    |       |
| password   | varchar(60)  | NO   |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

- Post 테이블

```
mysql> create table Post(
    -> id INT NOT NULL AUTO_INCREMENT,
    -> user_id int,
    -> title VARCHAR(100),
    -> date_posted DATETIME DEFAULT NOW(),
    -> content TEXT NOT NULL,
    -> foreign key (user_id) references User(id) on update cascade,
    -> PRIMARY KEY(id));
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> DESC Post
    -> ;
+-------------+--------------+------+-----+-------------------+-------------------+
| Field       | Type         | Null | Key | Default           | Extra             |
+-------------+--------------+------+-----+-------------------+-------------------+
| id          | int          | NO   | PRI | NULL              | auto_increment    |
| user_id     | int          | YES  | MUL | NULL              |                   |
| title       | varchar(100) | YES  |     | NULL              |                   |
| date_posted | datetime     | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| content     | text         | NO   |     | NULL              |                   |
+-------------+--------------+------+-----+-------------------+-------------------+
5 rows in set (0.00 sec)
```

2. config.py 수정하기

```
import os
import pymysql
import mysql.connector
pymysql.install_as_MySQLdb()

db = {
    'user'     : 'user2',
    'password' : '+"test"+',
    'host'     : '127.17.0.1',
    'port'     : '3306',
    'database' : 'flask_test'
}

class Config:
  SECRET_KEY='301653b4421209309537996ef97b19e5'
  SQLALCHEMY_DATABASE_URI = f"mysql+mysqlconnector://{db['user']}:{db['password']}@{db['host']}:{db['port']}/{db['database']}?charset=
  MAIL_SERVER='smtp.googlemail.com'
  MAIL_PORT=587
  MAIL_USE_TLS=True
  MAIL_USERNAME=os.environ.get('EMAIL_USER')
  MAIL_PASSWORD=os.environ.get('EMAIL_PASS')
```
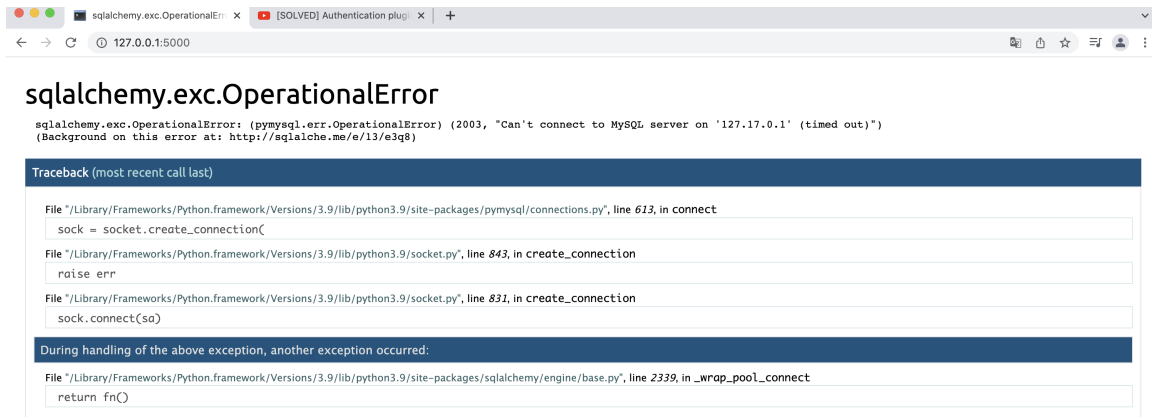
호스트 명:

```
^C(tutorial) EunSunui-MacBookAir:Backup_flask_blog eunsunkim$ python3 run.py
/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages/flask_sqlalchemy/__init__.py:833: FSADeprecationWarning:
 SQLALCHEMY_TRACK_MODIFICATIONS adds significant overhead and will be disabled by default in the future.  Set it to True or False to s
uppress this warning.
  warnings.warn(FSADeprecationWarning(
 * Serving Flask app "flaskblog" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages/flask_sqlalchemy/__init__.py:833: FSADeprecationWarning:
 SQLALCHEMY_TRACK_MODIFICATIONS adds significant overhead and will be disabled by default in the future.  Set it to True or False to s
uppress this warning.
  warnings.warn(FSADeprecationWarning(
 * Debugger is active!
 * Debugger PIN: 242-812-213
```

- mysql이 있는 컨테이너 실행 중임을 확인
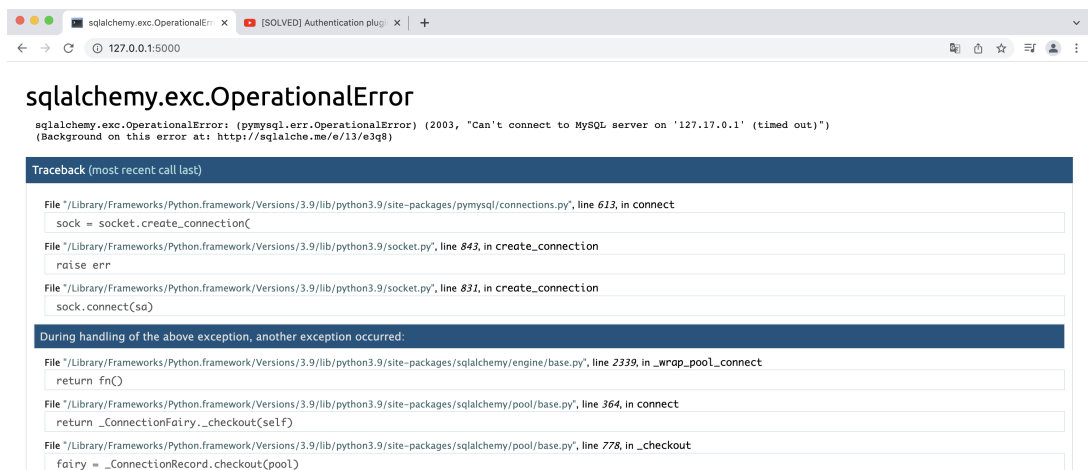
```
(base) EunSunui-MacBookAir:~ eunsunkim$ docker ps
CONTAINER ID   IMAGE          COMMAND                CREATED      STATUS         PORTS                                NAMES
7c47242ae53b   mysql:latest   "docker-entrypoint.s…" 2 days ago   Up 36 minutes  0.0.0.0:3306->3306/tcp, 33060/tcp    some-mysql
(base) EunSunui-MacBookAir:~ eunsunkim$
```

2. 에러 발생

- 'user' : 'root', 'password' : '+"my-secret-pw"+'일 때

- 'user' : 'user2', 'password' : '+"test"+'일 때 (같은 에러 발생)

참고:

https://nickjoit.tistory.com/144

https://boritea.tistory.com/84?category=901314

https://velog.io/@sungjun-jin/TIL-0524-Flask-mysql-연동하기