DB세팅

1) DB생성

db설치가 잘 되었다면 root 계정으로 mysql로 접속이 가능 할 것이다. cmd나 워크벤치를 통해 root계정에 접속한다. DB에 어떻게 접근하든 상관은 없다. IntelIJ에서도 DB로 접근이 가능하도록 기능을 제공하고 있다.

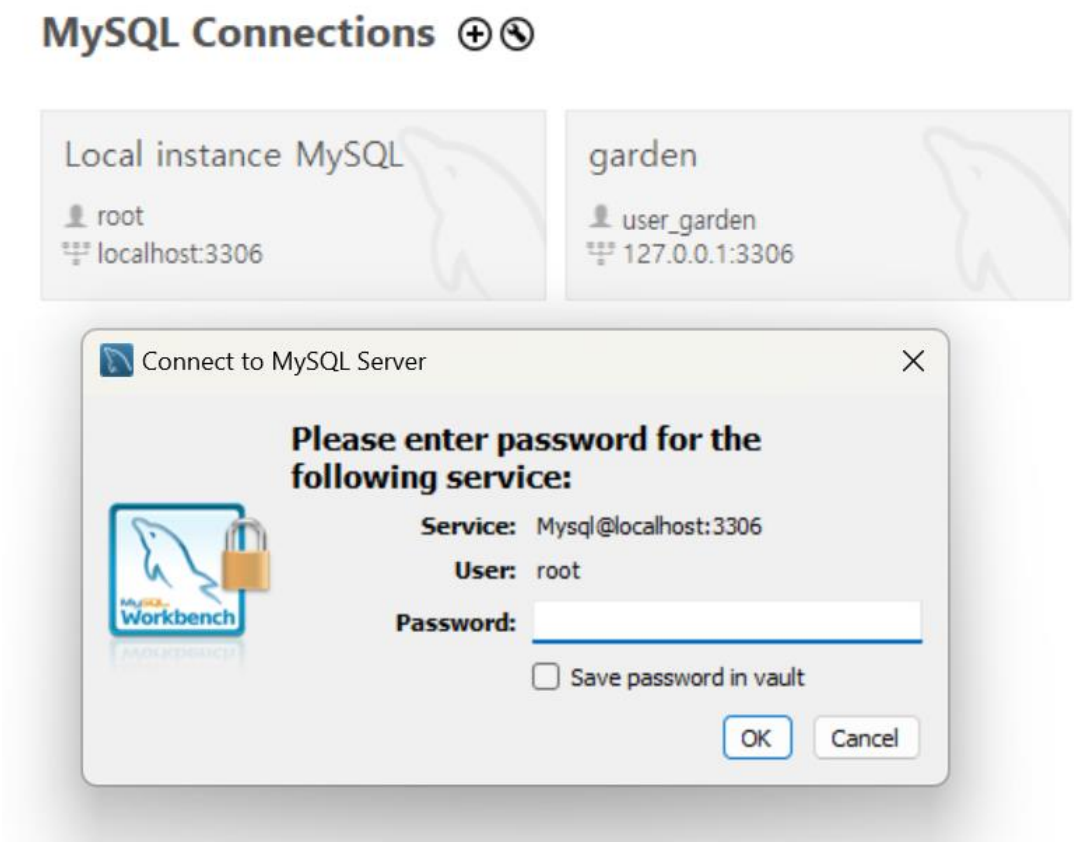
```
C:\Users\977me>mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 401
Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```



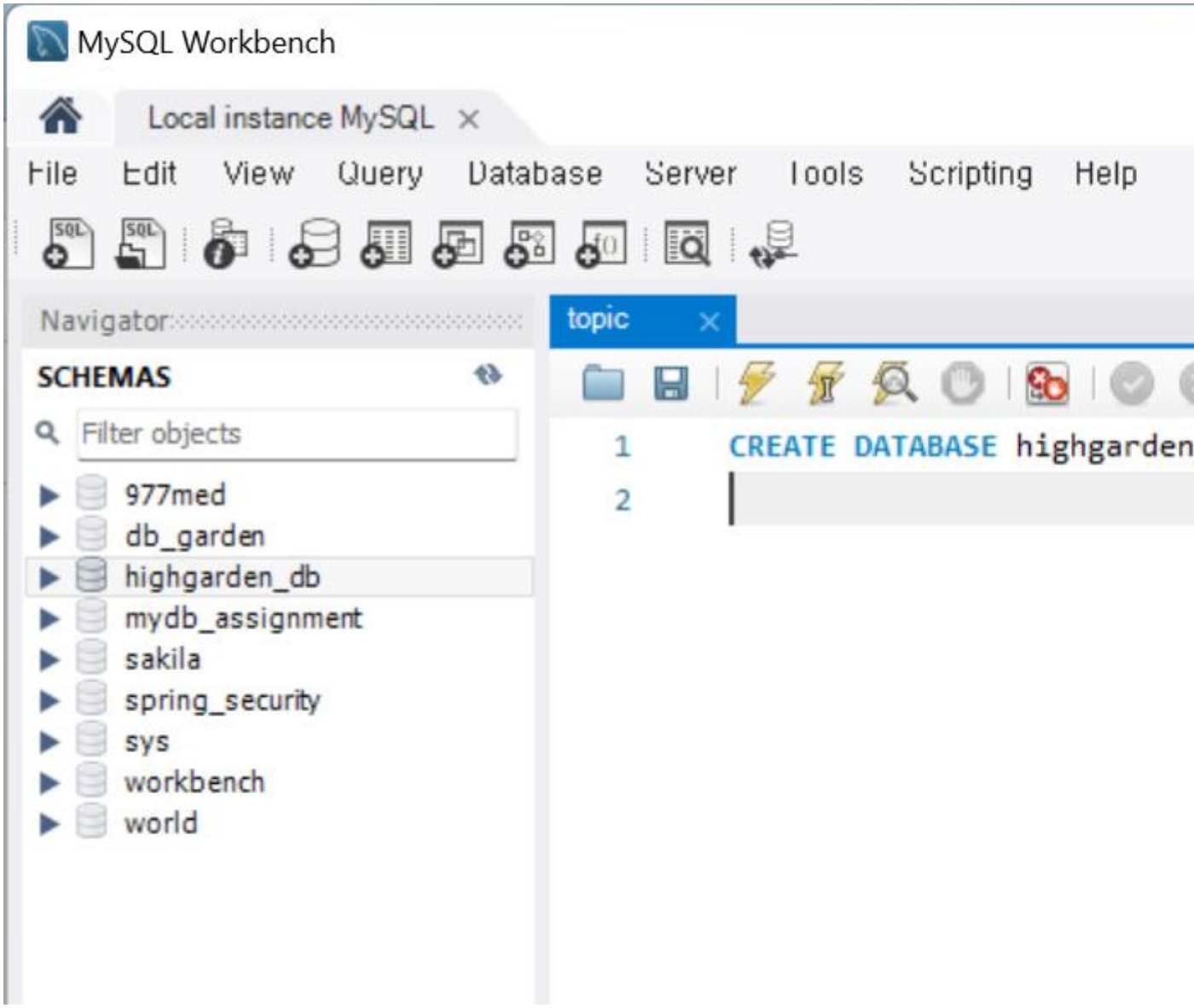
root 계정 접속

root계정으로 db에 접속하게 되면 이제 명령어를 사용할 수 있게된다.

CREATE DATABASE highgarden_db default CHARACTER SET UTF8;

SHOW DATABASES; --현재 DB목록을 보여줌

CREATE명령어로 일단 작업을 하게될 DB를 생성한다. 나같은 경우 DB명을 "highgarden_db"로 정하였다. 나중에 spring boot에서 DB와 연동하기위해 DB명이 사용되기 때문에 잘 기억하도록 하자.



DB생성 확인

SHOW DATABASES 명령어를 사용해 DB가 잘 생성되었는지 확인해보자. cmd환경과 워크벤치 환경에서 모두 잘 생성된 것을 볼 수 있다.

2) 사용자 생성

DB가 생성되면 사용자도 생성해준다. 보통 root계정으로 개발을 하지는 않기 때문에 게시판 작업을 하게 될 사용자를 따로 생성해준다.

CREATE USER '{username}'@'localhost' IDENTIFIED BY '{password}';

유저생성은 다음과 같은 방식으로 가능하다. 생성할 유저의 아이디와 DB로 접근 할 수 있는 IP, 그리고 해당유저를 식별할 수 있도록 비밀번호를 설정한다.

CREATE USER 'highgarden_user'@'localhost' IDENTIFIED BY '1234';

나는 'highgarden_user' 라는 유저를 생성하고 비밀번호를 1234로 설정하였다. 본 프로젝트는 로컬환경에서 구현할 예정이기 때문에 ip는 localhost로 설정하였다. 만약 db서버를 분리하고 싶거나 aws같은 클라우드 환경에서 구축하고 싶다면 localhost부분을 해당 db로 접속가능한 ip로 변경해주면 된다.

3) 권한부여

사용자를 처음 생성하고나면 어떠한 권한도 갖고있지 않다.

GRANT ALL PRIVILEGES ON {database}.* TO '{username}'@'localhost';

```
FLUSH PRIVILEGES;
```

위와 같은 방식으로 권한을 명시적으로 부여할 수 있는데 ALL PRIVILEGES는 모든 권한을 의미한다.

GRANT ALL PRIVILEGES ON highgarden_db.* TO 'highgarden_user'@'localhost';

앞서 DB와 사용자를 생성하는 과정을 잘 수행했다면 그 때 생성한 이름들을 이용해 권한을 부여할 수 있다.

권한이 생성된 후에는 'FLUSH PRIVILEGES' 명령을 수행한다.

FLUSH PRIVILEGES 명령어는 MySQL 서버의 메모리에 저장된 권한 테이블을 다시 로드하여, 최근에 수행된 권한 변경 사항이 즉시 반영되도록 한다. MySQL에서는 사용자 계정 및 권한 정보가 mysql 데이터베이스의 테이블에 저장되는데, 이 테이블의 변경 사항은 MySQL 서버가 재시작되기 전까지 자동으로 반영되지 않기 때문이다.

권한이 잘 부여되었는지 확인하기 위해 아래 명령어를 수행한다.

SHOW GRANTS FOR 'highgarden_user'@'localhost';

```
mysql> show grants for 'highgarden_user'@'localhost';
+-----------------------------------------------------------------------+
| Grants for highgarden_user@localhost                                  |
+-----------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `highgarden_user`@`localhost`                   |
| GRANT ALL PRIVILEGES ON `highgarden_db`.* TO `highgarden_user`@`localhost` |
+-----------------------------------------------------------------------+
2 rows in set (0.00 sec)
```

권한생성 확인

highgarden_user가 highgarden_db에 대한 모든 권한을 부여받았음을 알 수 있다.

4) 테이블 생성

테이블은 일단 두 개를 생성하도록 한다. 게시물의 정보를 담을 테이블인 board_table와 첨부파일 정보를 담게될 board_file_table을 생성한다.

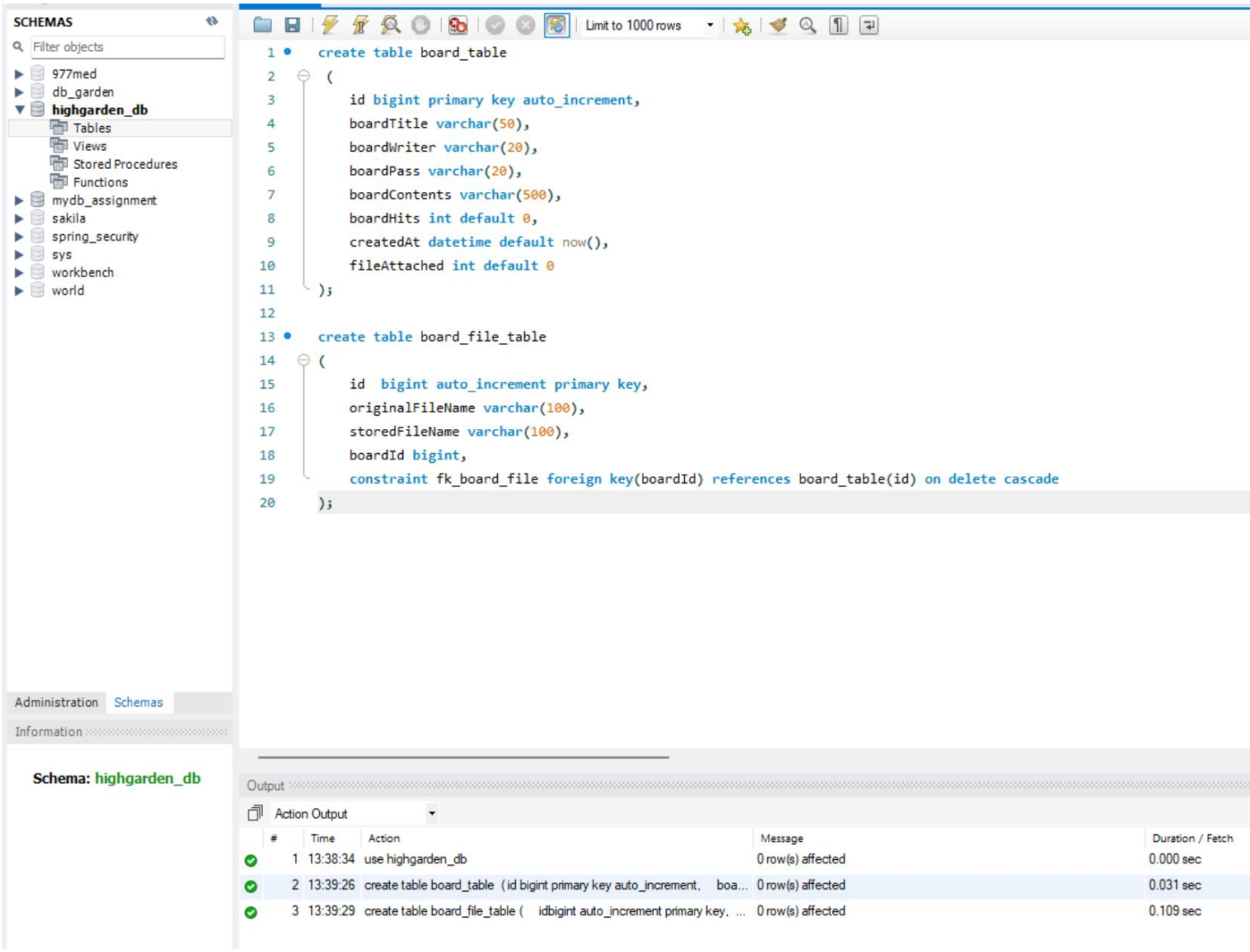
```
--board_table 생성

create table board_table
 (
        id bigint primary key auto_increment,
    boardTitle varchar(50),
    boardWriter varchar(20),
```

```
    boardPass varchar(20),
    boardContents varchar(500),
    boardHits int default 0,
    createdAt datetime default now(),
    fileAttached int default 0
);

--board_file_table 생성
create table board_file_table
(
    id    bigint auto_increment primary key,
    originalFileName varchar(100),
    storedFileName varchar(100),
    boardId bigint,
    constraint fk_board_file foreign key(boardId) references board_table(id) on delete cascade
);
```

아주 기본적인 컬럼들로 구성된 테이블이다. 게시물에 파일이 첨부되기 때문에 외래키 관계를 맺어준다. PK인 id는 auto_increment를 사용해서 자동으로 생성 되도록 하였다.



워크 벤치에서 테이블을 생성하였고 아래 성공적으로 수행되었다는 메시지를 확인할 수 있다.

여기까지 하면 DB세팅은 끝이 난다. 다음 시간에는 지난번 세팅했던 spring boot 서버와 DB를 연동해보자.

공유하기

게시글 관리

구독하기