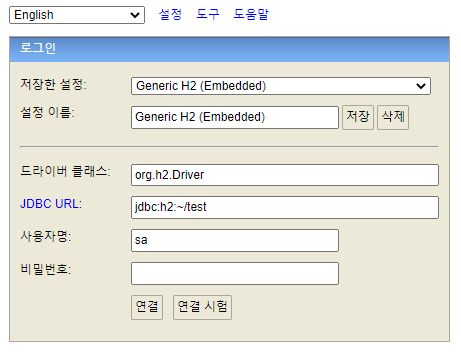
**H2 데이터베이스 설치**

1) <https://www.h2database.com/html/download-archive.html>

**1.4.200 버전 설치**

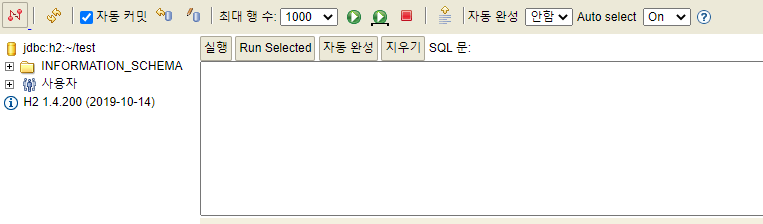
2) h2\bin 디렉토리에서 h2.bat 실행

h2\bin> h2.bat



만약, 접속이 안될 경우 주소창의 IP 주소를 localhost로 바꾸어주면 된다.

3) 연결 클릭

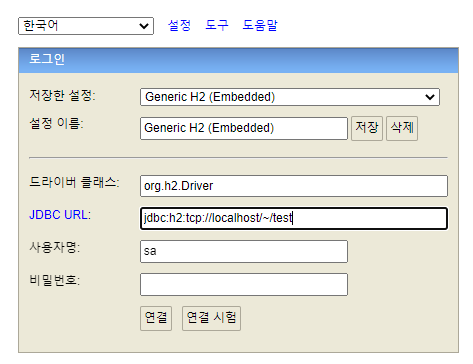


4) 새 cmd 창을 열어 dir

test.mv.db 가 생성되어있는 것을 확인

5) 소켓을 통해 접근할 수 있도록 url 변경

jdbc:h2:tcp://localhost/~/test



🔸**TABLE 생성**

CREATE TABLE member

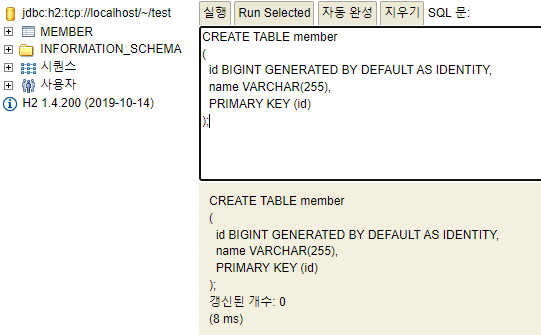
(

id BIGINT GENERATED BY DEFAULT AS IDENTITY,

name VARCHAR(255),

PRIMARY KEY (id)

);



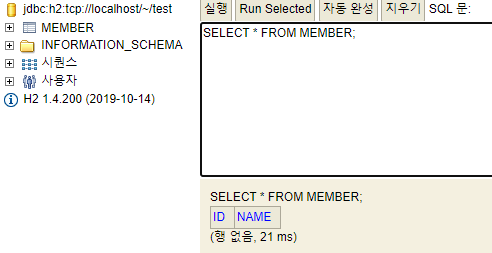
MEMBER table이 생성된 것을 확인

**GENERATED BY DEFAULT AS IDENTITY**

id BIGINT GENERATED BY DEFAULT AS IDENTITY : id 값이 들어오지 않으면 DB가 자동으로 Default 값을 채워넣어줌

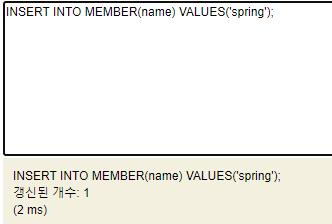
🔸**TABLE 조회**

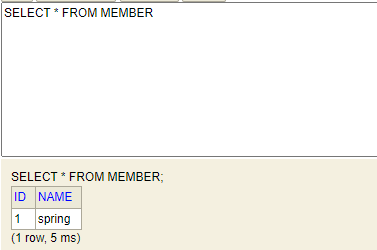
SELECT \* FROM MEMBER;



🔸 **TABLE에 데이터 넣기**

INSERT INTO MEMBER(name) VALUES('spring');





📚**순수 Jdbc**

build.gradle의 dependencies에 추가

spring.datasource.url=jdbc:h2:tcp://localhost/~/test

spring.datasource.driver-class-name=org.h2.Driver

spring.datasource.username=sa

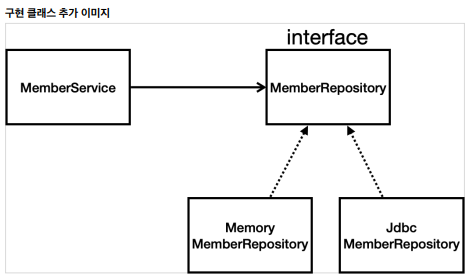
🔸**Jdbc**

자바 프로그램과 DB를 연결할 때 필요한 driver

🔸 **repository에서 DB로 메모리를 변경하는 방법**

1) MemberRepository interface를 구현한 JdbcMemberRepository class 생성

public class JdbcMemberRepository implements MemberRepository



2) configuration만 수정하면 된다!

spring에서 제공하는 DataSource를 생성하여 memberRepository에 DI해준다.

private DataSource dataSource;

@Autowired

public SpringConfig(DataSource dataSource) {

this.dataSource = dataSource;

}

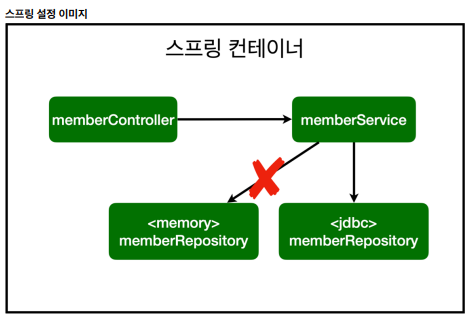
@Bean

public MemberRepository memberRepository() {

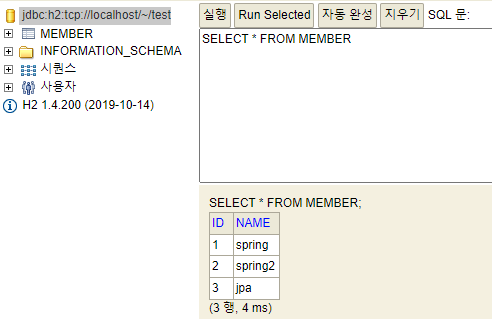
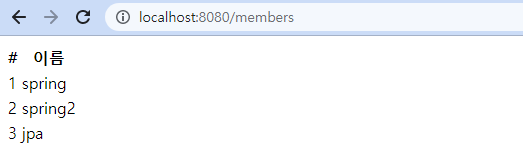
*// return new MemoryMemberRepository();*

return new JdbcMemberRepository(dataSource);

}



실행 결과 : DB에 잘 연결된다.



🔸 **Spring (객체 기반)**

* 개방-폐쇄 원칙확장에는 열려있고, 수정과 변경에는 닫혀있다.
* spring의 DI를 상요하면 기존 코드의 수정 없이 설정만으로 interface 구현 클래스를 변경할 수 있다.

📚 **스프링 통합 테스트**

스프링과 연결된 DB까지 통합된 테스트 진행

**@SpringBootTest**

: 스프링 컨테이너와 테스트를 함께 실행

**@Transactional**

: 테스트 케이스에서 사용하면 test를 시작하기 전에 transaction을 시작하고, test가 완료되면 rollback  
  
▶ test 후에 DB에 데이터가 남지 않으므로 다음 테스트에 영향을 주지않고 반복 가능