애플리케이션 테스트 관리

*** 단위 테스트**

- 개발 과정에서 진행되는 테스트로

  구현된 모듈(함수, 서브루틴, 컴포넌트 등)의 **기능 수행 여부를 판정**하고,

  내부에 존재하는 논리적 오류를 검출하는 테스트

*** 테스트 케이스 설계**

- 테스트 진행 하기 위한 산출물로 개발된 기능 또는 애플리케이션이 요구사항을 준수하는지 확인하기 위해 설계된 입력 값, 실행 조건, 기대결과 등을 작성한 명세서

*** 테스트 케이스 기본 작성 방법(반드시 작성되어야 될 내용)**

1. **테스트 케이스 ID** : 테스트 케이스 고유 식별 ID
2. **테스트 시나리오** : 테스트 케이스의 동작 순서를 기술
3. **테스트 데이터** : 테스트를 진행하기 위해 특별히 개발된 데이터

   꼭 필요한 데이터만을 작성할 것
4. **시작 조건** : 테스트 수행을 위한 시작 조건
5. **종료 조건** : 테스트 완료를 위한 종료 조건
6. **예상 결과** : 테스트가 성공적으로 수행 되었을 때의 예상 결과

ex) 단위 테스트용 테이스 케이스 예시

1. 테스트 케이스 ID - TEST101
2. 테스트 시나리오 - 로그인 화면에서 아이디와 비밀번호를 입력하여 로그인을 테스트한다.
3. 테스트 데이터 -

```
CREATE TABLE MEMBER(
    MEMBER_NO NUMBER PRIMARY KEY,
    MEMBER_ID VARCHAR2(30) NOT NULL,
    MEMBER_PWD VARCHAR2(80) NOT NULL,
    MEMBER_NAME VARCHAR2(15) NOT NULL
);
INSERT INTO MEMBER VALUES(1, 'user01', 'pass01', '유관순');
INSERT INTO MEMBER VALUES(2, 'user02', 'pass02', '홍길동');
INSERT INTO MEMBER VALUES(3, 'user03', 'pass03', '신사임당');
```

4. 시작 조건 - 화면 개발 및 서버 모듈 연동이 완료 되었을 때
5. 종료 조건 - 샘플 회원(사용자 3명)의 로그인이 성공하였으며,
   정상적인 로그인 정보가 서버에 전달 되었을 때
6. 예상 결과 - 로그인 되어진 회원의 아이디를 콘솔에 출력하도록 함.
7. 실제 결과 - 로그인 되어진 회원의 아이디가 콘솔에 출력된 것을 확인함.

* 위의 테스트 결과에 대한 성능을 분석하여 성능을 개선하는 방안을 제시한다면?

현재 실행한 SQL은 PreparedStatement 이므로 CallableStatement(데이터 베이스 내에 프로시저를 선언하여, 저장 된 프로시저를 실행만 해주면 그 프로시저 내용이 처리 되며 실행 속도가 빠르고 부하가 적다는 장점이 있다.)로의 구현도 고려해야 한다.

* 데이터 베이스와 통신하는 과정에서 지나치게 많은 정보를 받아와 애플리케이션의 성능이 저하되는 문제가 발생한다면?

'select * from ... ' 와 같은 쿼리문은 현재 서비스에서 요구하는 정보 외의 부가적인 데이터를 함께 받아오기 때문에 성능 저하를 유발할 수 있다. 이를 해결하기 위해서는 Mapper에서 'select user_id, user_pwd, user_name from ... '와 같이 필요로 하는 정보만을 받도록 수정해야 한다.