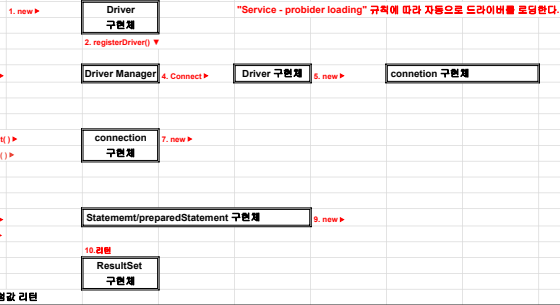
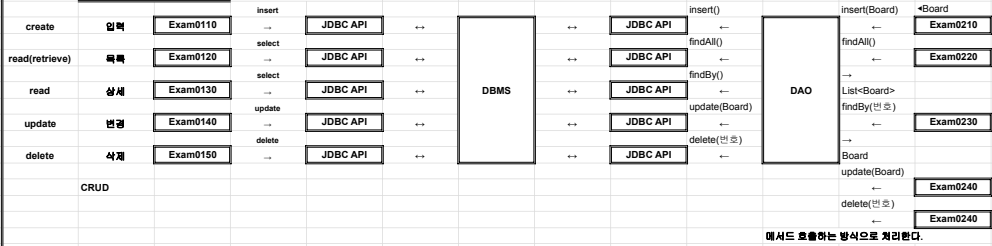


~JDBC API의 기본 객체 - 메서드 흐름



~JDBC API를 사용해 데이터를 처리 --- 학습목 --- DAO 클래스로 정의



PreparedStatement

문장이 같을
SQL 삽입 공격이 불가
비이너리 데이터 -> SQL문가 분리되어 있기 때문에 따로 관리.

Statement vs PreparedStatement

1) SQL 문장의 간결함
[Statement]
- 값을 가지고 문자열로 직접 SQL 문을 만들기 때문에 작성하거나 읽기 힘들다.
[PreparedStatement]
- SQL 문장과 값이 분리되어 있기 때문에 작성하거나 읽기 쉽다.

2) SQL 삽입 공격

[Statement]
- 사용자가 입력한 값을 가지고 SQL 문을 만들기 때문에 해킹되기 쉽다.
[PreparedStatement]
- SQL 문장과 값이 분리되어 다뤄지기 때문에 해킹할 수 없다.

3) 바이너리 데이터 다루기

[Statement]
- 문자열로 SQL 문장을 만들기 때문에 바이너리 타입의 컬럼 값을 설정할 수 없다.
[PreparedStatement]
- setXxx() 메서드를 호출하여 값을 설정하기 때문에 바이너리 타입의 컬럼 값을 설정할 수 있다.

4) 실행 속도

[Statement]
- executeUpdate()를 실행할 때 SQL 문을 파라미터로 전달한다.
- 호출할 때마다 SQL 문법을 분석하기 때문에 반복 실행하는 경우 SQL 문법도 반복 분석하므로 실행 속도가 느리다.
[PreparedStatement]
- 미리 SQL 문을 작성한 다음 DBMS 프로토콜에 맞게 파싱해 놓은 후, executeUpdate() 호출한다.
- 따라서 executeUpdate()를 호출할 때 마다 SQL 문법을 분석하지 않으므로 반복해서 실행하는 경우,

