JSP

1. PrepareStatement
2. Model
3. MVC패턴1
4. MVC패턴2

* **Statement** 객체를 대신하는 **PreparedStatement** 객체

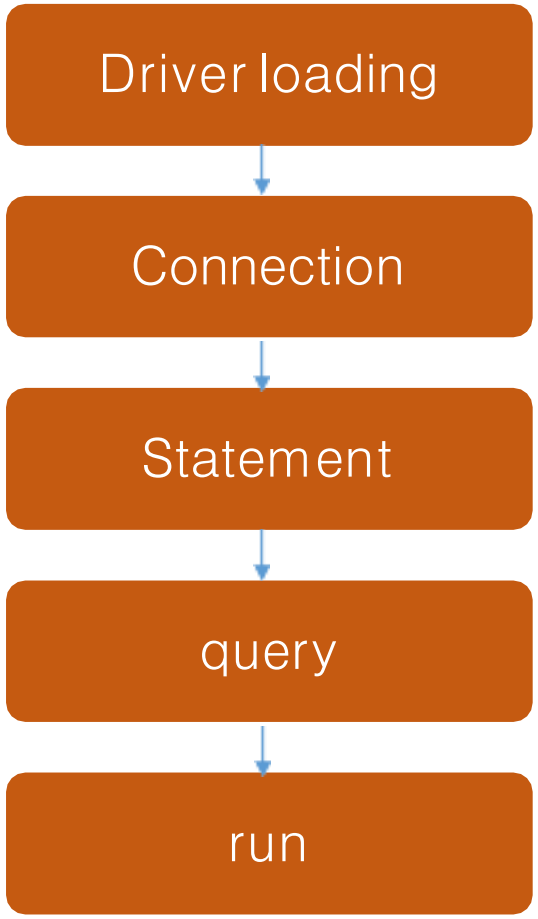
- Statement 객체와 PreparedStatement객체는 쿼리문을 실행하는 동일한 기능을 제공합니다.

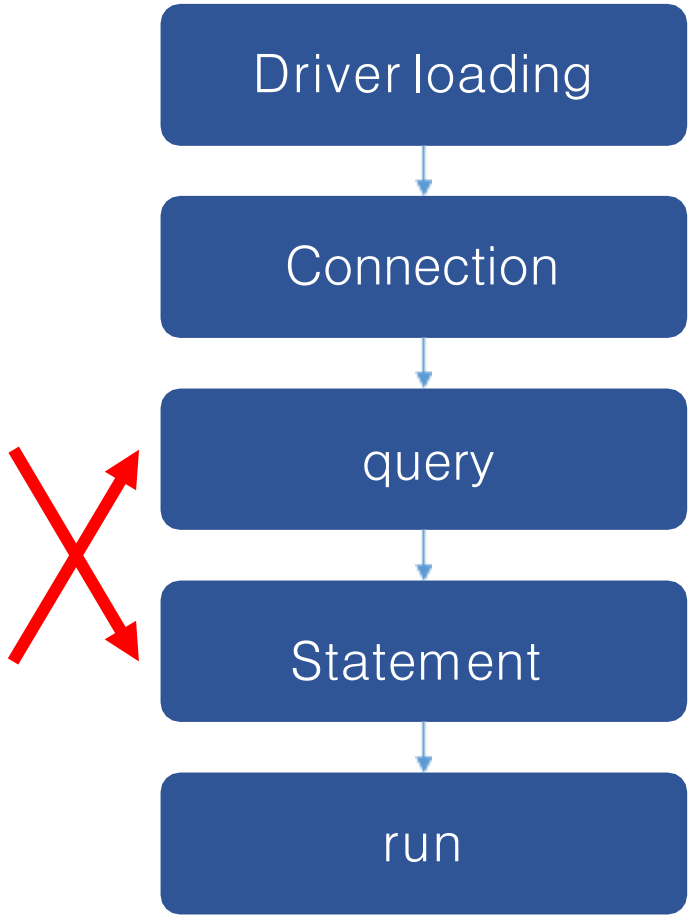
- 그런데 **PreparedStatement** 객체를 사용하는 이유는 이 객체가 값 변환을 자동으로 해주는 기능을 제공하고, 간결한 코드를 만들 수 있기 때문입니다.

- Statement객체는 지정할 값이 많아질 경우 따옴표가 복잡하게 얽히기 때문에 코드 작성에서 오류가 발생할 수도 있고, 코드 수정시에도 어려움이 발생합니다.

- 그러나 **PreparedStatement** 객체는 값을 지정할 때 값 부분을 물음표(?)로 처리하기 때문에 간단히 값을 지정할 수 있습니다. 이 때 첫번째 물음표의 인덱스는 1이며, 이후 물음표의 인덱스는 나오는 순서대로 인덱스 값이 1씩 증가합니다.

PrepareStatement 실행순서





Class.*forName("드라이버 이름");*

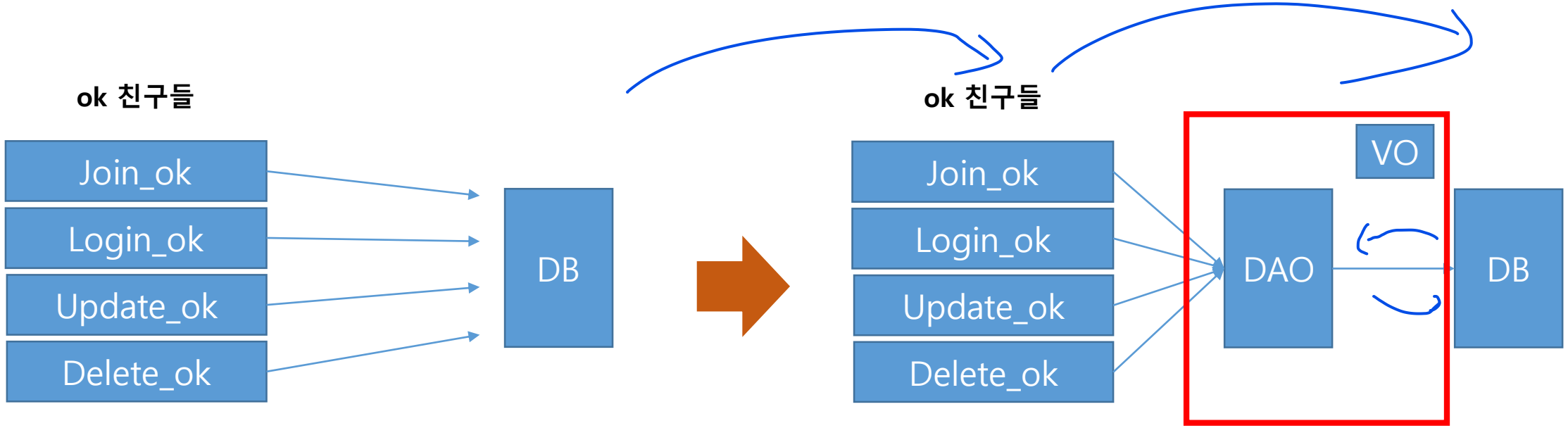
con = DriverManager.*getConnection(url, id, pw);*

String sql= "insert into MEMBER values(**?, ?**) ";

pstmt = con.prepareStatement(sql);
pstmt.setString(1, 값);
pstmt.setString(2, 값);

res = pstmt.executeUpdate();

Model - DAO 클래스와 VO클래스



* MVC 패턴에서의 Model

1. **DAO 클래스**(Data Access Object)

- 데이터베이스에 접속해서 데이터의 추가, 삭제, 수정 등의 작업을 하는 클래스입니다.

- 일반적으로 JSP 혹은 Servlet에서 위의 로직을 함께 기술할 수도 있지만, 유지보수 및 코드의 모듈화를 위해 별도의 DAO 클래스를 만들어 사용합니다.

- 보통 한 개의 테이블마다 한 개의 DAO 클래스를 작성합니다.

- DAO 클래스는 테이블로부터 데이터를 읽어와 자바 객체로 변환하거나 자바 객체의 값을 테이블에 저장합니다.

- 따라서 DAO를 구현하면 테이블의 컬럼과 매핑되는 값을 갖는 자바빈 클래스를 항상 작성해야 합니다. 이 자바빈 클래스를 VO클래스라 부릅니다.

---

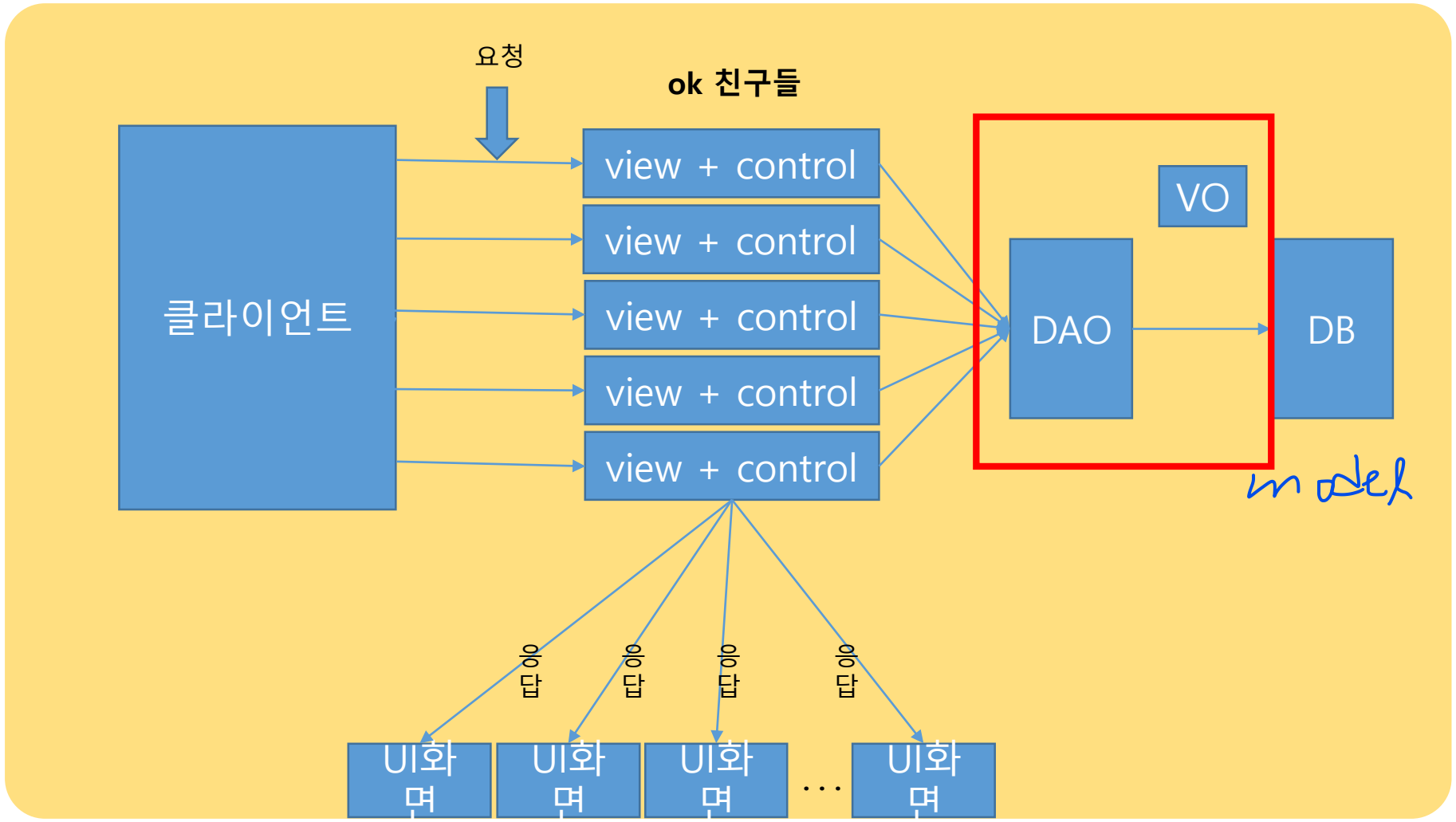
1. **VO 클래스**(Value Object) / DTO 클래스(Data Transfer Object)

- DAO 클래스를 이용하여 데이터베이스에서 데이터를 관리할 때 데이터를 일반적인 변수에 할당하여 작업할 수도 있지만, 별도의 VO 클래스를 작성하여 데이터베이스와 관련된 변수들의 모음 역할을 합니다.

- VO클래스는 자바빈 클래스로 생성합니다.

MVC1 전체적인 컴포넌트 설계

M <- model
V <-view
C <- controller



MVC2 전체적인 컴포넌트 설계