

JSP

1. JDBC
2. Connector 설정
3. Connection객체
4. Statement객체
5. ResultSet객체

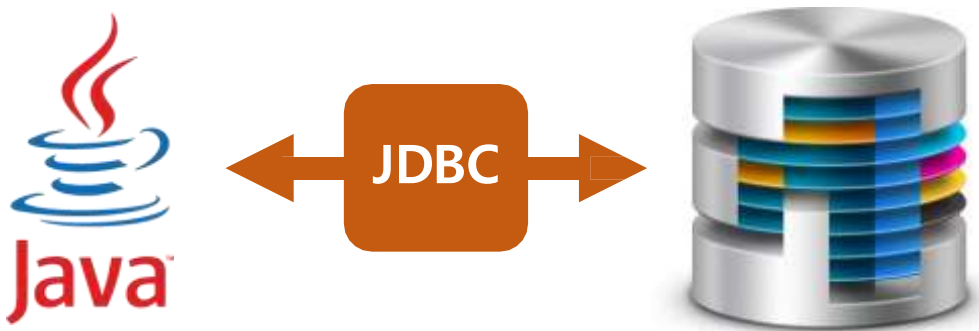
* JDBC(Java Database Connectivity) 프로그래밍

- JDBC란? 자바 프로그램에서 SQL문을 실행하여 데이터를 관리하기 위한 JAVA API입니다.

SQL문을 실행하기 위한 다양한 방법이 존재합니다

1. MyBatis (Spring에서 학습)
2. 하이버네이트 JPA 등

Java가 DB와 통신할수 있게 해주는 API



연결API connector가 필요하다

- MySQL을 사용한다면 MySQL용 JDBC를 사용합니다.
- ORACLE을 사용한다면 ORACLE용 JDBC를 사용합니다.



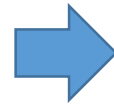
* JDBC라이브러리 추가 위치

▼ WEB-INF

▼ lib

ojdbc8.jar

web.xml



lib폴더에 넣어주면 됩니다

- 주요 DBMS의 JDBC 드라이버 클래스

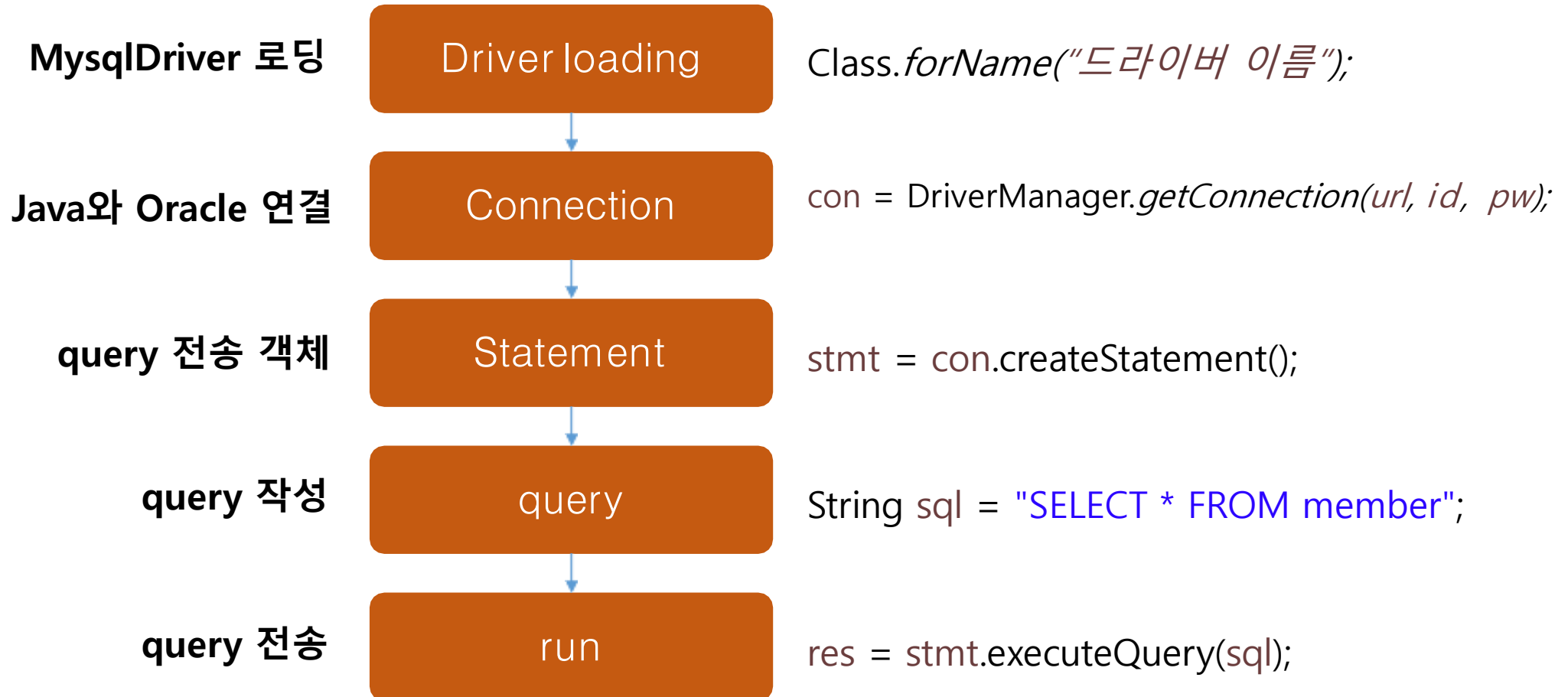
1. MySQL: "com.mysql.jdbc.Driver"
2. ORACLE: "oracle.jdbc.driver.OracleDriver"

외워 쓰는거 아닙니다~

- 주요 DBMS의 JDBC URL 패턴

1. MySQL: "jdbc:mysql://호스트이름:포트번호/DB이름"
2. ORACLE: "jdbc:oracle:thin:호스트이름:포트번호:DB이름"
PDB의경우: "jdbc:oracle:thin:호스트이름:포트번호/XEPDB1"

JDBC 실행순서



* 데이터베이스 연결을 위한 Connection 객체

- 데이터베이스와 JAVA 연결
- DriverManager.getConnection() 메서드를 사용하여 커넥션을 구할 수 있습니다.
- 연결이 끝난 뒤에는 반드시 close() 메서드를 호출하여 Connection 객체가 사용한 시스템 자원을 반환해야 합니다. 그렇지 않으면 시스템 자원이 불필요하게 소모되어 커넥션을 구할 수 없는 상황이 발생할 수도 있습니다.

• 쿼리문을 실행하기 위한 Statement 객체

- Connection 객체를 생성한 후에는 Connection 객체로부터 Statement를 생성하고 쿼리문을 실행

* 주요메서드

1. executeQuery() - Select 쿼리문을 실행합니다.
2. executeUpdate() - Insert, Update, Delete 쿼리문을 실행합니다.

* 쿼리 실행 결과 값을 읽어오는 ResultSet 객체

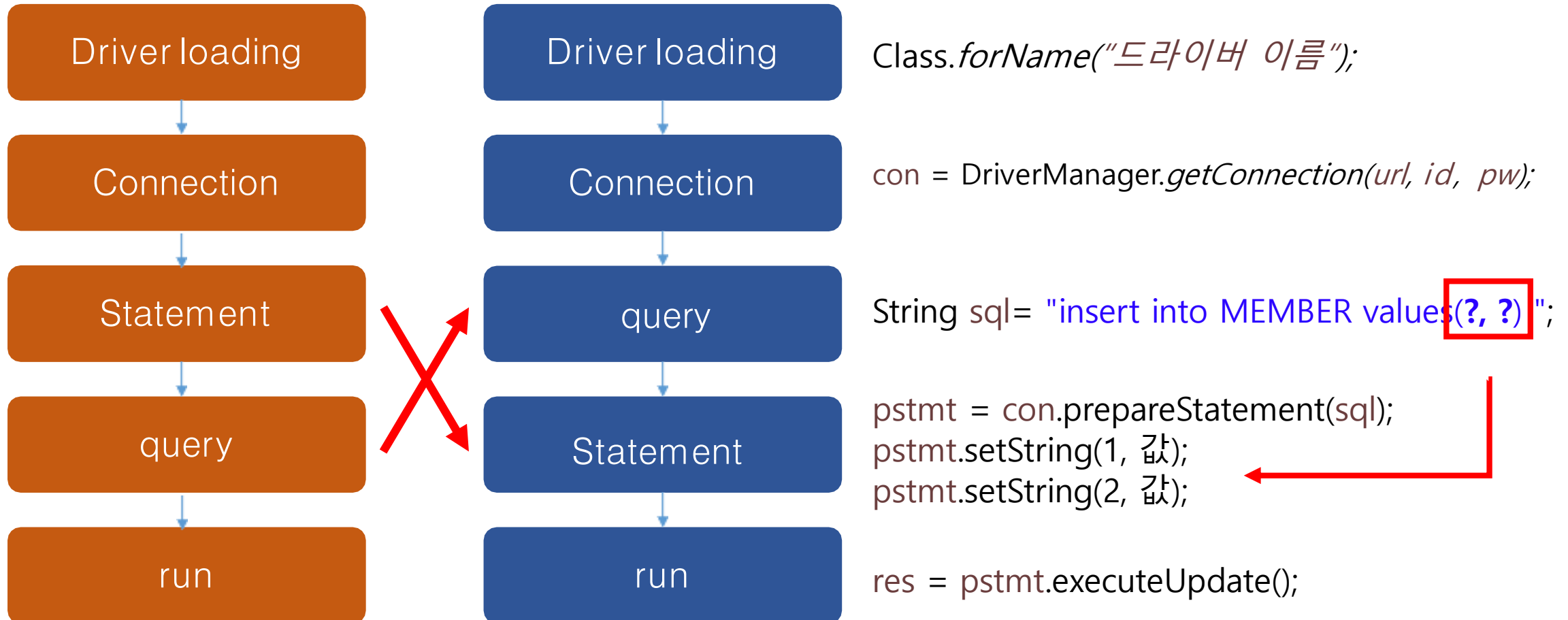
- Statement 객체의 executeQuery() 메서드는 Select 쿼리문의 결과를 ResultSet 객체에 담아서 리턴 합니다.

* ResultSet 주요 메서드

1. next() - 다음 값의 확인
2. getString() - 지정된 컬럼 값을 String으로 읽음
3. getInt() - 지정된 컬럼 값을 int 타입으로 읽음
4. getDouble() - 지정된 컬럼 값을 double 타입으로 읽음



PreparedStatement 실행순서



* **Statement** 객체를 **대신**하는 **PreparedStatement** 객체

- Statement 객체와 PreparedStatement 객체는 쿼리문을 실행하는 동일한 기능을 제공합니다.
- **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.