JSP와 데이터베이스 연동

안 화 수

JSP와 Oracle 연동

- 데이터베이스 접속 방식

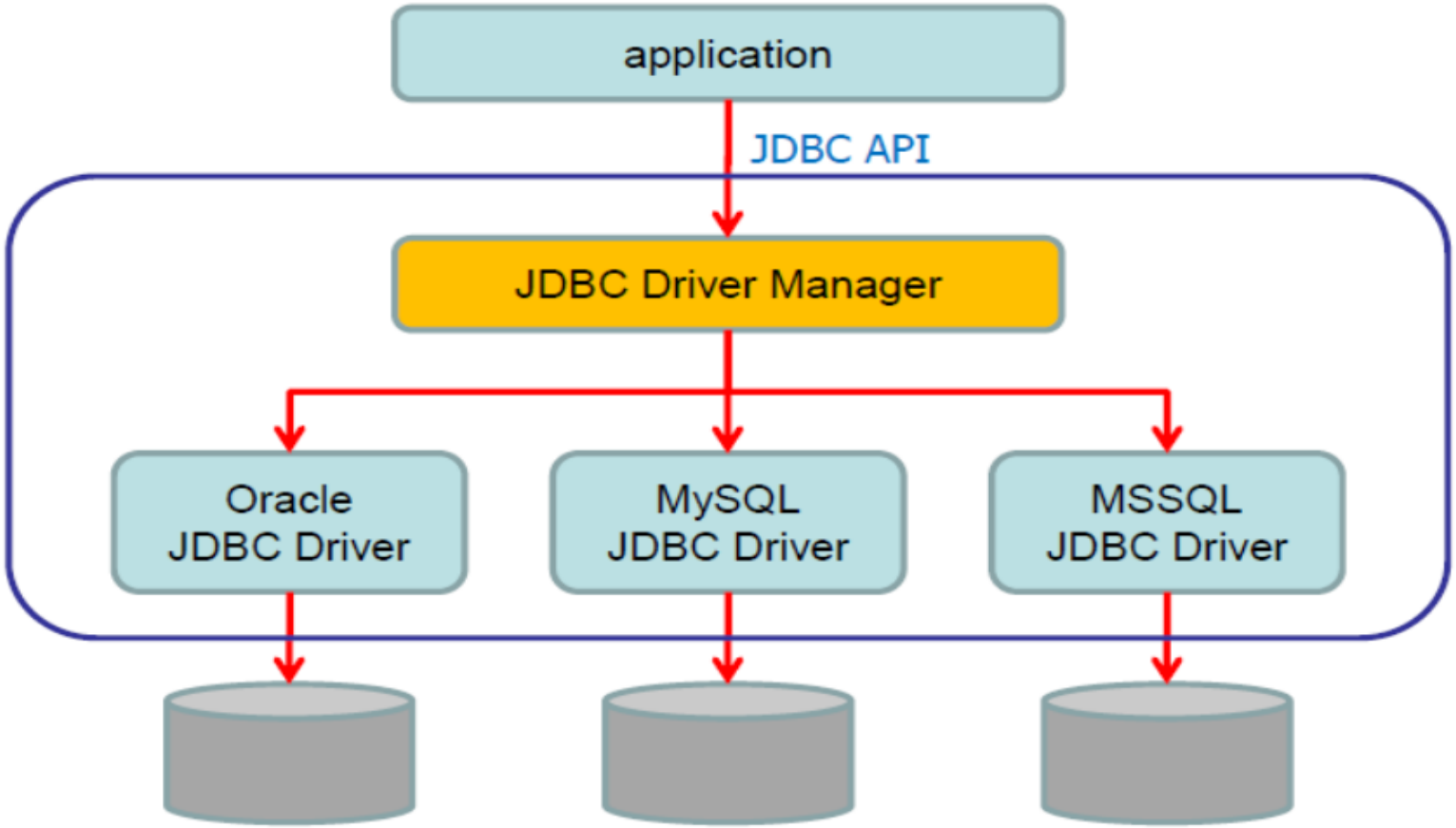
1. JDBC(Java DataBase Connectivity) 방식

2. DBCP(DataBase Connection Pool) 방식

3. ORM(Object Relational Mapping) 프레임워크
   ex) iBatis, MyBatis, hibernate, JPA etc

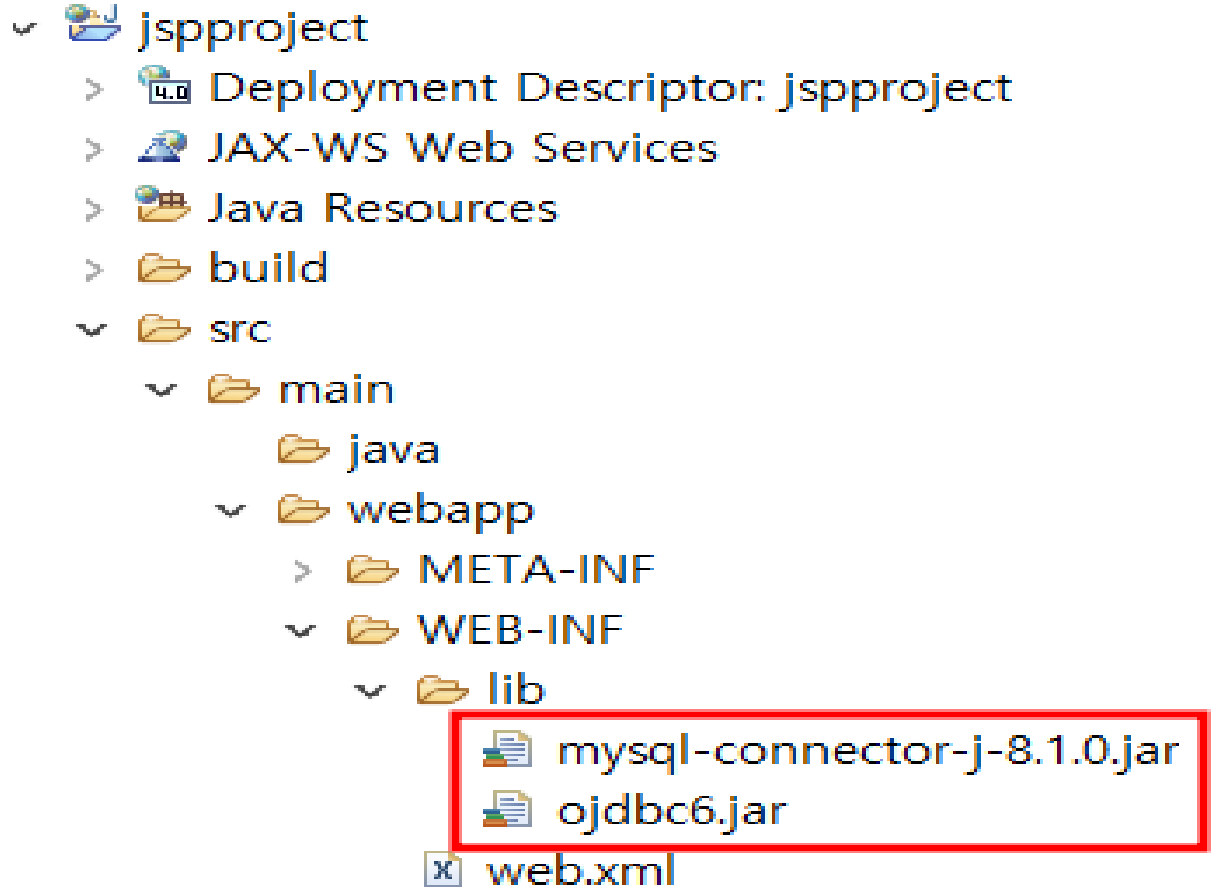
JDBC 방식

- JDBC Architecture



JDBC 방식

- JDBC Driver

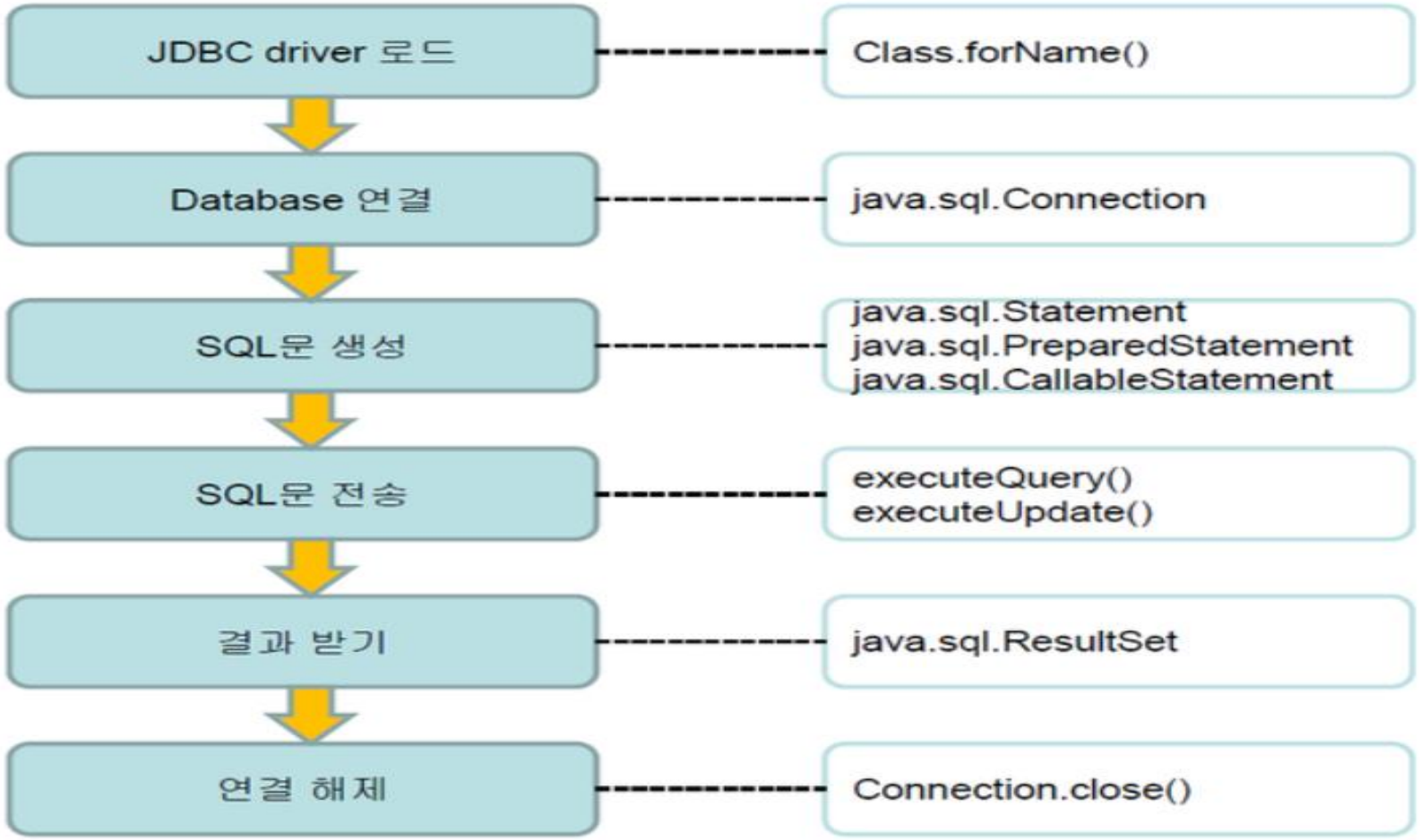


JDBC 방식 DB연동 테스트

```
<%@ page contentType="text/html; charset=utf-8" %>
<%@ page import="java.sql.*" %>
<%
	Connection con=null;
	try{
		String driver = "oracle.jdbc.driver.OracleDriver";

		String url = "jdbc:oracle:thin:@localhost:1521:xe";
		String user = "scott";
		String password = "tiger";

		Class.forName(driver);
		con=DriverManager.getConnection(url, user, password );

		out.println("제대로 연결되었습니다.");

	} catch(Exception e){
		e.printStackTrace();
	}
%>
```

JDBC 방식

- 데이터베이스 연동 절차



JDBC 방식

- 데이터베이스 연결

```
String driver = "oracle.jdbc.driver.OracleDriver";
String url = "jdbc:oracle:thin:@localhost:1521:xe";

Class.forName(driver);

con=DriverManager.getConnection(url, ”scott”, “tiger” );
```

JDBC 방식

- **SQL문 실행**

SQL문			Method
select	-	ResultSet	executeQuery()
insert update delete	-	int	executeUpdate()

JDBC 방식

- 데이터베이스 연동 예제

- 테이블 생성

```
create  table  member1(
        id varchar2(12) primary key,
       passwd varchar2(12) not null,
        name varchar2(10) not null,
        reg_date timestamp not null );
```

DBCP 방식

❖ DBCP(DataBase Connection Pool) 방식

- WAS 실행 시 미리 일정량의 DB Connection 객체를 생성하고 Pool 이라는 공간에 저장해 둡니다.
  그리고 DB 연결 요청이 있으면, 이 Pool 이라는 공간에서 Connection 객체를 가져다 쓰고 반환 하게 됩니다.

- 데이터베이스와 연결된 컨넥션을 미리 만들어서 풀(pool) 속에 저장해 두고 있다가 필요할 때에 컨넥션을 풀에서 가져다 쓰고 다시 풀에 반환하는 기법

❖ DBCP 특징

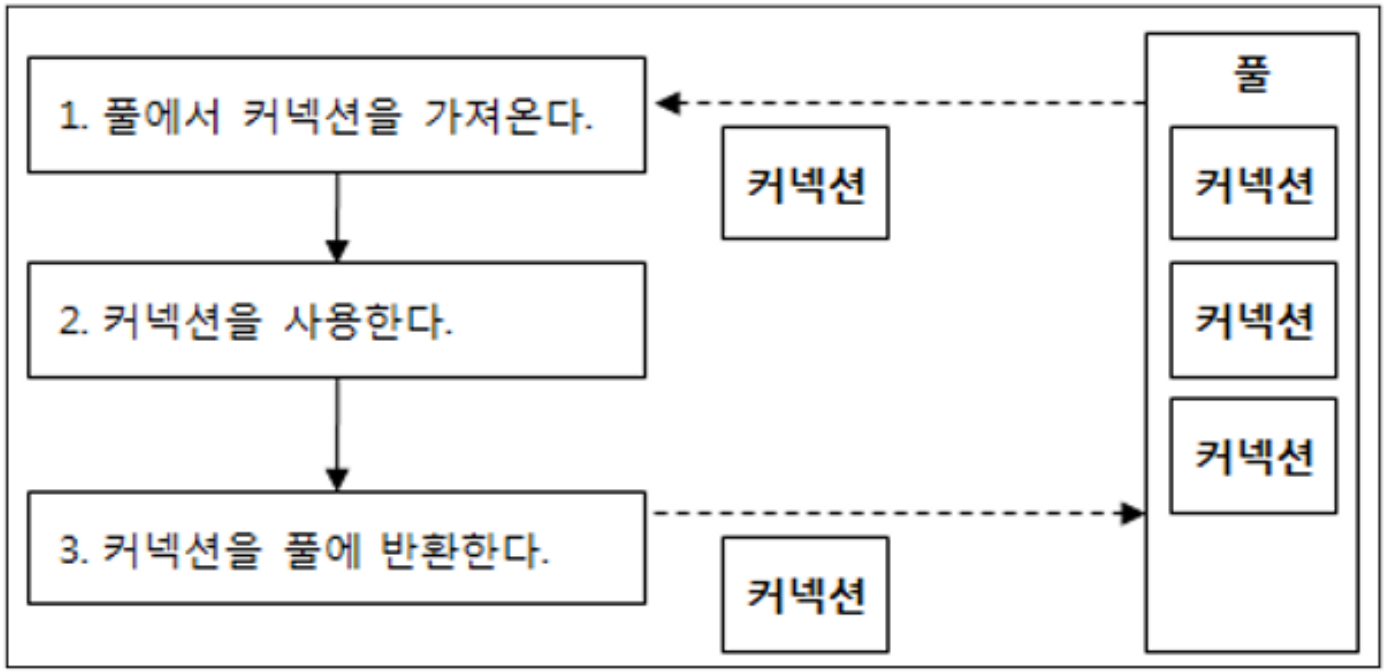
- 컨넥션을 생성하는 데 드는 연결 시간이 소비되지 않는다.

- 컨넥션풀에 저장된 컨넥션을 재사용 할 수 있다.

DBCP 방식

DBCP(DataBase Connection Pool) 방식

- 데이터베이스와 연결된 커넥션을 미리 만들어서 풀(pool) 속에 저장해 두고 있다가 필요할 때에 커넥션을 풀에서 가져다 쓰고 다시 풀에 반환하는 기법



DBCP 방식

- Connection Pool 환경 설정

dbcpTest - webapp - dbcpAPITest.jsp (테스트 파일)

- META-INF - context.xml (컨넥션풀 설정파일)

- WEB-INF - web.xml

- lib - ojdbc6.jar

DBCP 방식

- Connection Pool 환경 설정 파일 – context.xml

```
<Context>
  <Resource name="jdbc/OracleDB"
   auth="Container"
   type="javax.sql.DataSource"
   username="scott"
   password="tiger"
   driverClassName="oracle.jdbc.driver.OracleDriver"
   factory="org.apache.tomcat.dbcp.dbcp2.BasicDataSourceFactory"
   url="jdbc:oracle:thin:@localhost:1521:xe"
   maxActive="500"
   maxIdle="100"/>
</Context>

<!--
 maxActive="500"   컨넥션풀이 관리하는 컨넥션의 최대 갯수(기본값 : 8)
 maxIdle="100"     컨넥션풀이 관리하는 최대 유휴 갯수(기본값 : 8)
-->
```

DBCP 방식

- **Connection Pool 에서 컨넥션 구해오기**

```
<%@ page language="java" contentType="text/html; charset=utf-8"%>
<%@ page import="java.sql.*"%>
<%@ page import="javax.sql.*" %>
<%@ page import="javax.naming.*" %>
<%
    Connection conn = null;

    try {
        Context init = new InitialContext();
        DataSource ds = (DataSource) init.lookup("java:comp/env/jdbc/OracleDB");
        conn = ds.getConnection();

        out.println("<h3>연결되었습니다.</h3>");
    }catch(Exception e){
        out.println("<h3>연결에 실패하였습니다.</h3>");
        e.printStackTrace();
    }
%>
```