

# 복습하기

## ▶ 서블릿의 기본구조 : 자바클래스 형태로 구현

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class 서블릿클래스명 extends HttpServlet{

    public void service(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html; charset=utf-8");
        PrintWriter out = response.getWriter();
        out.println( " 웹 브라우저로 보낼 내용");
        ...
    }
}
```



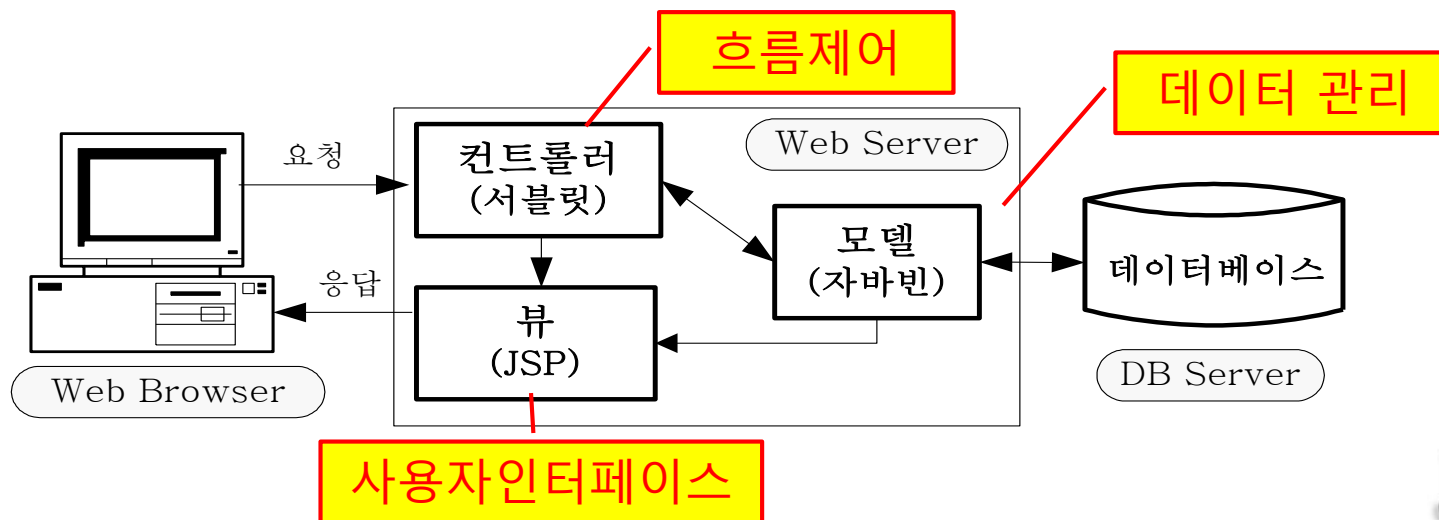
# 복습하기

- ▶ 서블릿은 클라이언트 요청에 대한 응답
  - ▶ doGet()과 doPost(), service() 메서드로 처리 내용 기술

메서드	설 명
doGet()	클라이언트가 GET 방식의 요청이 있을 때 처리
doPost()	클라이언트가 POST 방식의 요청이 있을 때 처리
doHead()	HEAD 요청을 처리
doPut()	PUT 방식의 요청이 있을 때 처리
doDelete()	DELETE 방식의 요청이 있을 때 처리
doOption()	OPTION 방식의 요청이 있을 때 처리
service()	요청의 종류와 관계없이 수행



# 복습하기



Java Servlets

- ▶ 컨트롤러는 클라이언트의 요청을 처리하기 위한 전체 흐름을 제어하는 역할 담당
- ▶ 다음 순서로 처리
  - ▶ ① 서블릿의 `doGet()` 또는 `doPost()` 메서드 등에서 클라이언트가 전송한 값을 검증
  - ▶ ② 모델에 관한 비즈니스 로직(business logic)을 호출한다.
  - ▶ ③ 결과를 `request` 또는 `session`의 `setAttribute()` 메서드로 저장한다.
  - ▶ ④ 뷰로 포워딩하여 jsp 페이지로 이동한다.

# Part2. 기초 프로그래밍

- ▶ Chapter4. JSP 기본문법
- ▶ Chapter5. JSP 내장객체
- ▶ Chapter6. JSP 입력 폼 설계
- ▶ Chapter7. JSP와 DB 연동
- ▶ Chapter8. 자바빈과 액션태그
- ▶ Chapter9. 쿠키와 세션
- ▶ Chapter10. 서블릿
- ▶ **Chapter11. DBCP**
- ▶ Chapter12. EL
- ▶ Chapter13. JSTL

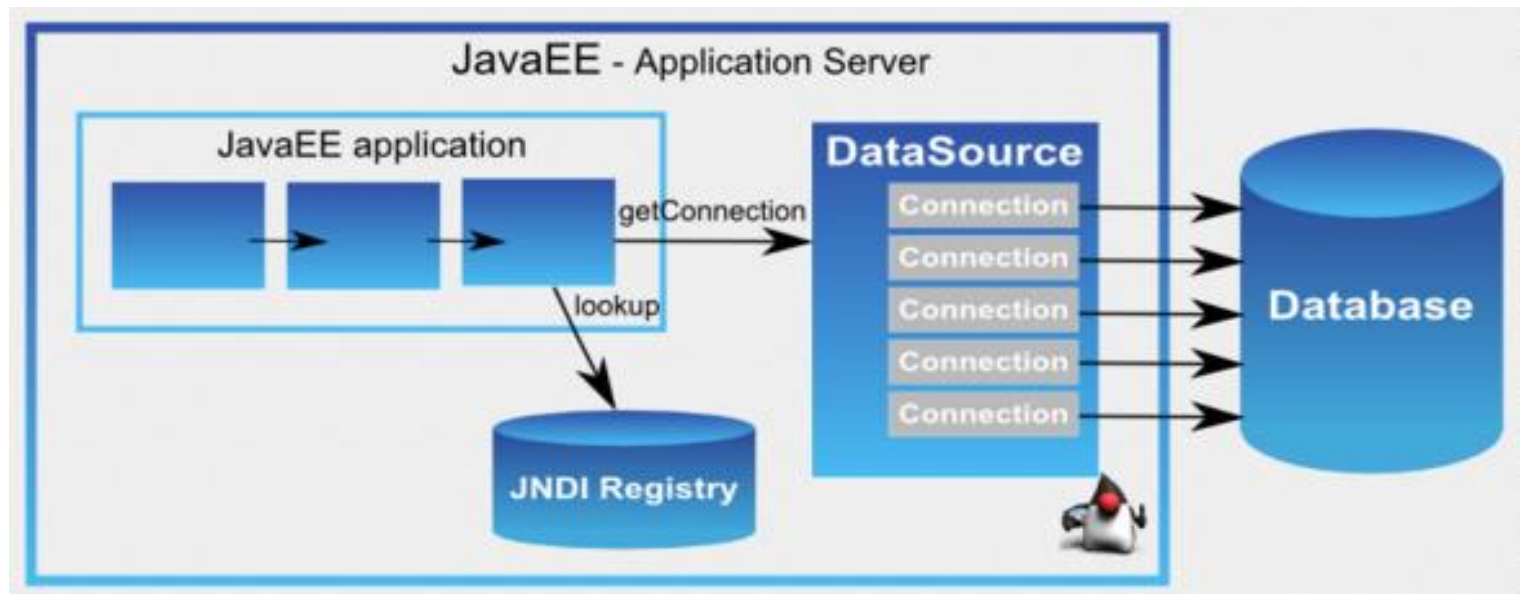
문법과  
기초 프로그래밍 실습

# 제11장 DBCP

## 11.1 JDBC 프로그래밍 절차의 코딩 문제

## 11.2 DBCP 개요

## 11.3 DBCP 기법



# 오늘의 수업 주제

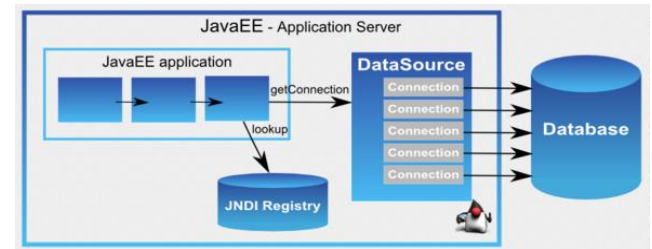
1. JDBC 프로그래밍 절차의 코딩 문제

2. DBCP 개요

3. DBCP 기법

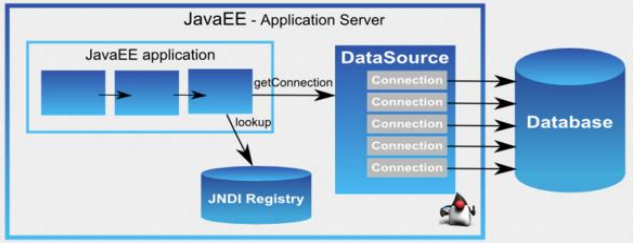
# 강의 목표

- ▶ 1. JDBC 프로그래밍 절차의 코딩 문제
- ▶ 2. DBCP 개요
- ▶ 3. DBCP 기법



## 11. DBCP

- 11.1 JDBC 프로그래밍 절차의 코딩 문제
- 11.2 DBCP 개요
- 11.3 DBCP 기법





# 1. 11.1 JDBC 프로그래밍 절차의 코딩 문제

```
1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2      pageEncoding="UTF-8"
3      import="java.sql.*" %>
4  <%
5      int i=0;
6      // SELECT문 문자열로 구성
7      String sql="SELECT * FROM Department";
8      // DB서버의 url
9      String url = "jdbc:oracle:thin:@220.67.2.3:1521:ora11";
10 try {
11     // 2단계: JDBC Driver 로드
12     Class.forName("oracle.jdbc.driver.OracleDriver");
13     // 3단계: DB서버 연결
14     Connection con = DriverManager.getConnection(url, "scott140", "pass140");
15     // 4단계: Statement 생성
16     Statement stmt = con.createStatement();
17     // 5단계: select문 실행 및 결과 반환
18     ResultSet rs = stmt.executeQuery(sql);
19 %>
20 |...
21 <%
22     // 6단계: 모든 행 반복 처리
23     while ( rs.next() ) {
24 %> <tr>
25         <td><%= ++i %></td>
26         <td><%= rs.getString(1) %> </td>
27         <td><%= rs.getString(2) %> </td>
28         <td><%= rs.getString(3) %> </td>
29     </tr>
30 %> <% } %>
31 </table>
32 <% // 7단계: 사용한 자원 해제
33     rs.close();
34     stmt.close();
35     con.close();
36 %>
37 ....
38 </body>
39 </html>
```

## 소스 코드 중복 코딩

- 데이터베이스 서버의 url
- JDBC Driver 로드
- DB 서버 접속
- 자원 해제

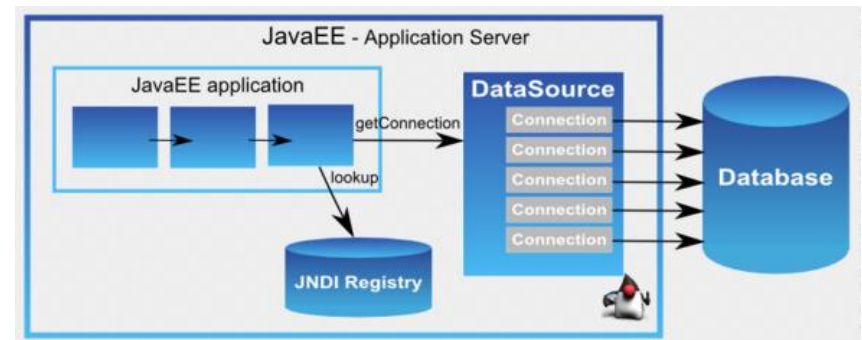
## 2. DBCP 개요

### ▶ DBCP(DataBase Connection Pool)

- ▶ 데이터베이스와 연결된 JNDI(Java Naming and Directory Interface)라는 커넥션(Connection)을 미리 만들어 풀(pool)속에 저장해 두고 있다가 필요할 때 커넥션 풀에서 가져와 쓰고, 사용한 후에 다시 풀에 반환하는 기법

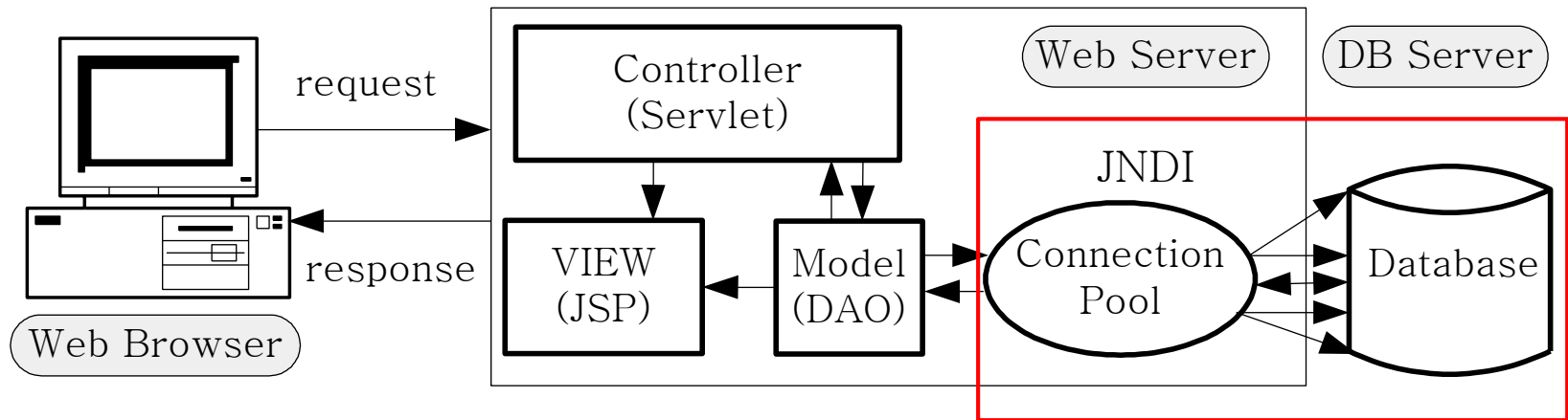
### ▶ 장점

- ▶ 데이터베이스 접속에 필요한 커넥션을 미리 생성하여 시간 절약
- ▶ 커넥션 재사용으로 생성되는 커넥션의 수가 많지 않아도 된다.
- ▶ 데이터베이스 서버 주소, 데이터베이스 드라이버, 사용자명, 암호의 보안 유지



## 2. DBCP 개요

- ▶ JNDI 등록 및 DBCP의 데이터베이스 연동에 필요한 작업
  - ▶ ① tomcat-dbcp.jar의 DBCP 관련 파일 준비
  - ▶ ② 데이터베이스 서버, 계정, 암호 등 "context.xml" 파일 생성
  - ▶ ③ 자원 참조명(resource-ref)의 정보를 "web.xml" 파일에 추가



## 2. DBCP 개요

- ▶ JNDI(Java Naming and Directory Interface)?
  - ▶ 자바 플랫폼 기반의 애플리케이션이 명칭 부여 및 디렉터리 서비스를 처리할 수 있는 응용 프로그램 인터페이스(API)
  - ▶ 디렉터리 서비스에서 제공하는 데이터 및 객체를 발견하고 참조(lookup)하기 위한 자바 API
  - ▶ 여러 대 서버간에 JNDI로 객체 등록하고, 이를 참조
  - ▶ JNDI는 "**javax.naming**" 패키지에 포함
  - ▶ 모든 자원들은 JNDI 네임스페이스의 "**java:com/env**"에 설정
- ▶ 이 기법은
  - ▶ 데이터베이스 서버 정보가 변경되더라도 소스 코드 변경없이 외부 파일의 정보만 수정하면 쉽게 변경 정보 적용 가능

# 3. DBCP 기법

## ▶ 기법 적용

- ▶ DBCP 파일 설치 (탐켓7.0이하)
- ▶ context.xml 파일에 정보 입력
- ▶ web.xml 파일에 JNDI 서비스 정보 추가 (탐켓7.0 이하)

## ▶ 3.1 DBCP의 jar 파일

### ▶ 탐켓 7.0 버전은 자체 포함

#### ▶ Tomcat-dbcp.jar 파일

### ▶ 탐켓 7.0 이하 버전

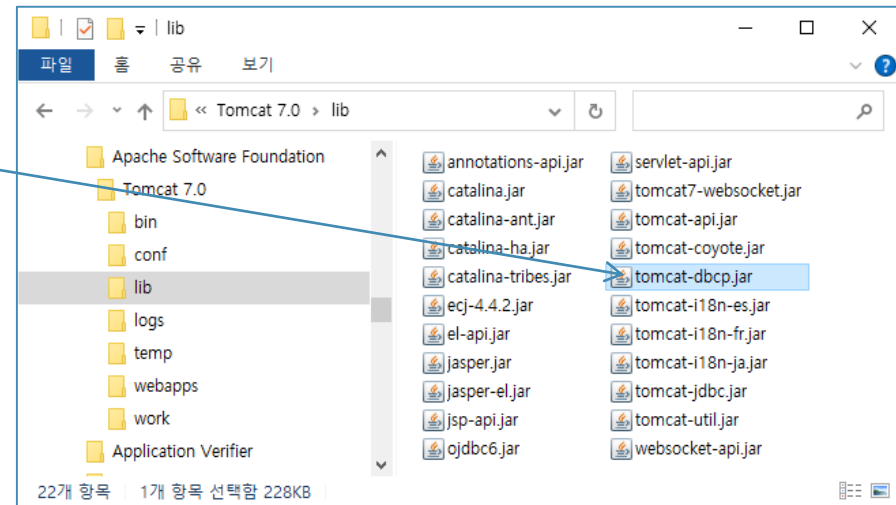
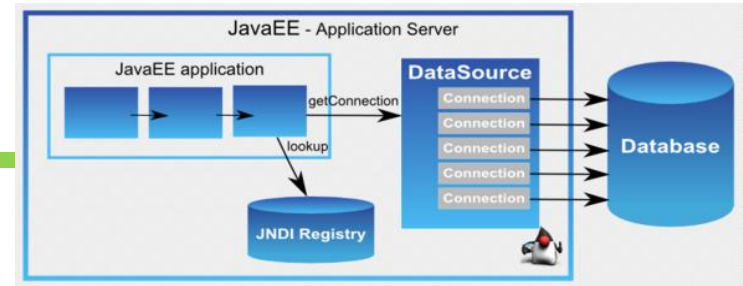
#### ▶ commons-collection-3.2.1.jar

#### ▶ commons-dbcp-1.4.jar

#### ▶ commons-pool-1.6.jar

### ▶ 저장 위치

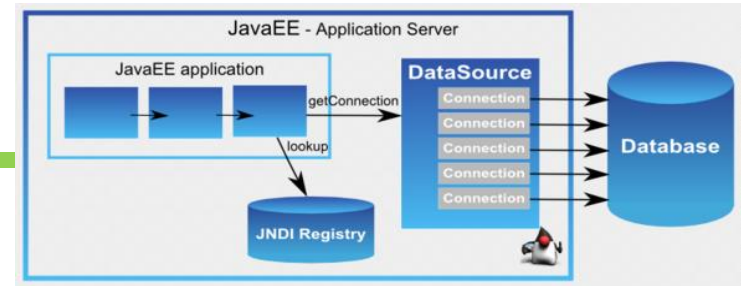
- Tomcat 폴더의 lib 또는 WEB-INF/lib



# 3. DBCP 기법

## ▶ 3.2 context.xml 파일 생성

### ▶ 기본적인 resource 요소



요 소	설 명
name	네임스페이스에 등록할 객체명
auth	DBCP를 관리할 자원관리자
type	해당 자원의 반환 객체타입(Connection 객체 반환)
driverClassName	JDBC 드라이버
factory	DBCP 사용 기본 파라메타
url	데이터베이스 서버의 url
username	데이터베이스 서버 접속 아이디(id)
password	데이터베이스 서버 접속 암호
maxActive	커넥션 풀의 최대 허용 개수, 0일 때 무제한
maxIdle	커넥션 풀의 여유분의 최대 개수, 0일 때 무제한

# 3. DBCP 기법

## ▶ 3.2 context.xml 파일 생성

### ▶ 정보 설정 [Oracle인 경우]

<context>

<Resource

name="jdbc/리소스명"

auth="Container"

type="javax.sql.DataSource"

username="아이디"

password="비밀번호"

driverClassName="oracle.jdbc.driver.OracleDriver"

factory="org.apache.tomcat.dbcp.BasicDataSourceFactory"

url="jdbc:oracle:thin:@ip주소:포트번호:SID"

maxActive="100"

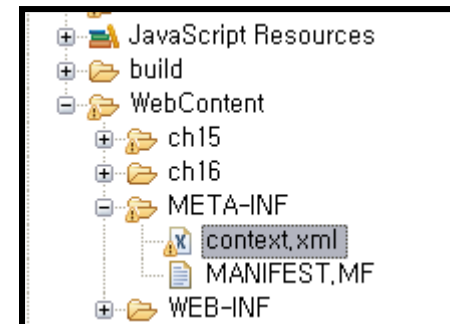
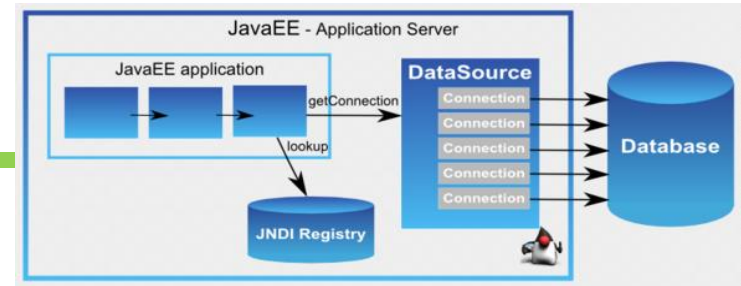
maxIdle="20"

</context>

소스코드에서 호출

### ▶ 파일 저장 위치 및 파일명

□ WebContent/META-INF/context.xml

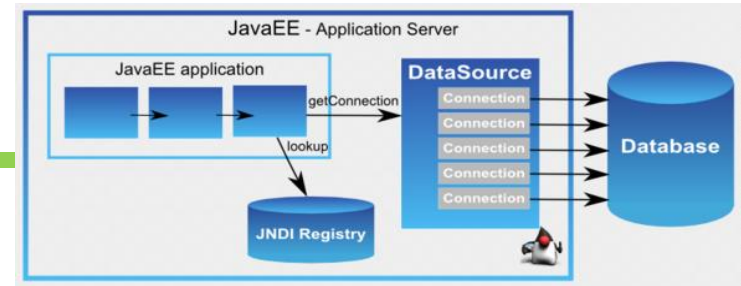


# 3. DBCP 기법

## ▶ 3.2 context.xml 파일 생성

### ▶ 정보 설정 [Oracle인 경우]

- 오라클 데이터베이스 서버의 ip 주소가 "222.66.2.3"이고, 포트번호가 1521, SID가 "oracle11", 사용자 아이디가 "stud", 암호가 "pass"일 때 DBCP의 정보를 "context.xml" 파일에 등록



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Context>
3   <Resource
4     name="jdbc/OracleDB"
5     auth="Container"
6     type="javax.sql.DataSource"
7     driverClassName="oracle.jdbc.driver.OracleDriver"
8     factory="org.apache.tomcat.dbcp.BasicDataSourceFactory"
9     url="jdbc:oracle:thin:@222.66.2.3:1521:oracle11"
10    username="stud"
11    password="pass"
12    maxActive="100"
13    maxIdle="20">
14  </Resource>
  </Context>
```



### 3. DBCP 기법

#### Tomcat 7.0 이상

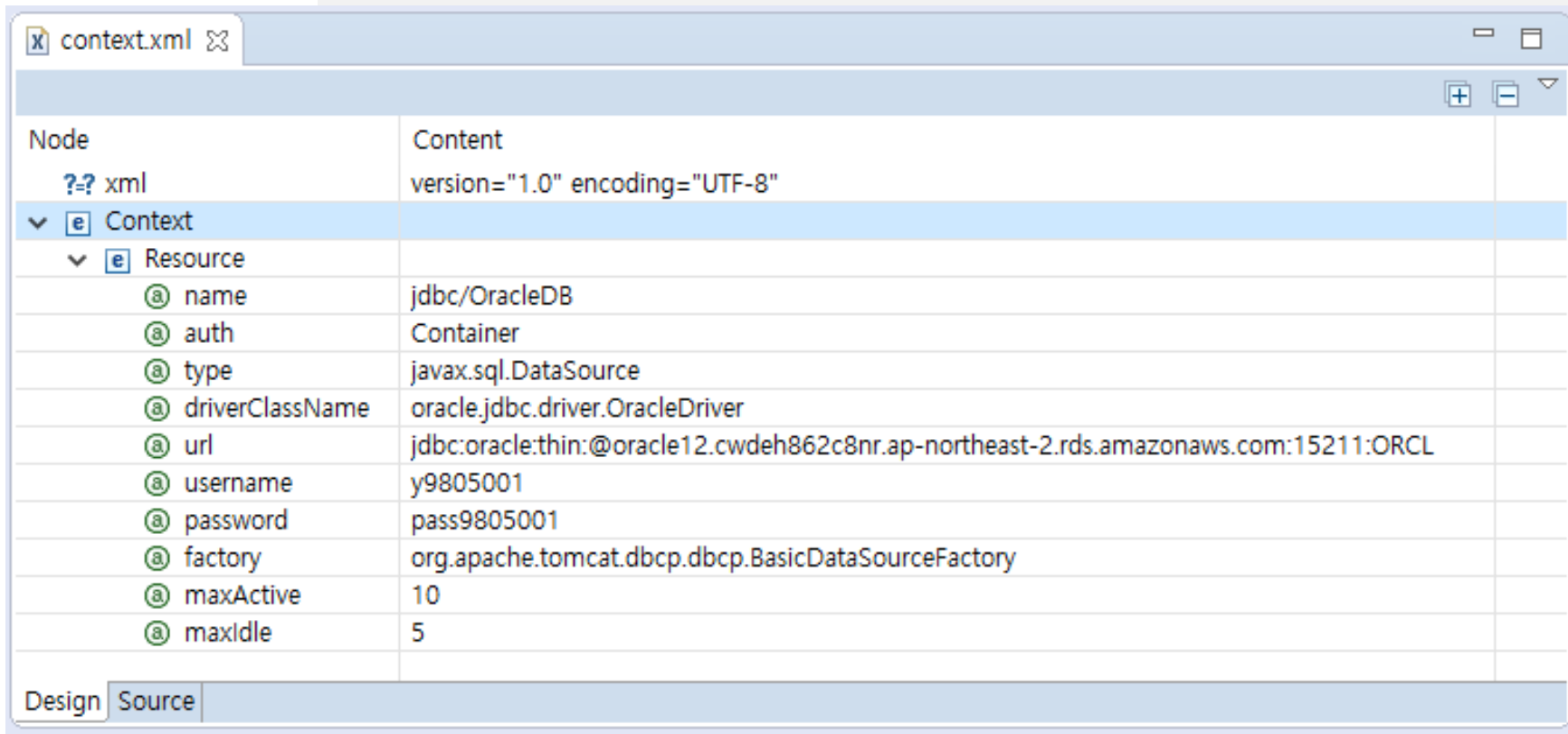
- ▶ **Tomcat7.0/lib**
  - ▶ Tomcat-dbcp.jar
  - ▶ 자동 설치
- ▶ **META-INF**
  - ▶ context.xml
  - ▶ 작성

#### Tomcat 7.0 이하

- ▶ **Tomcat x.x/lib**
  - ▶ commons-collection-3.2.1.jar
  - ▶ commons-dbcp-1.4.jar
  - ▶ commons-pool-1.6.jar
  - ▶ 별도 설치
- ▶ **META-INF**
  - ▶ context.xml
  - ▶ 작성
- ▶ **WEB-INF**
  - ▶ web.xml
  - ▶ Resource 추가

## 예제 11.1

오라클 서버의 호스트주소(아래url 참조), 포트번호(15211), SID(ORCL), 계정(y자신의학번), 암호(pass자신의학번 or 설정한 비번), 네임스페이스의 객체명(jdbc/OracleDB)을 참조하여 컨텍스트내의 META-INF 폴더에 context.xml 파일을 생성하시오.



**작성방법 : 교재 참조**

강의자료실 다운로드, 필요부분 수정후 사용

# 3. DBCP 기법

## ▶ 3.3 JNDI 리소스를 위한 web.xml 설정

### ▶ JNDI 리소스 사용 설정 [Oracle인 경우)

<resource-ref>

<description>Connection</description>

<res-ref-name>jdbc/리소스명</res-ref-name>

<res-type>javax.sql.DataSource</res-type>

<res-auth>Container</res-auth>

</resource-ref>

### ▶ 파일명

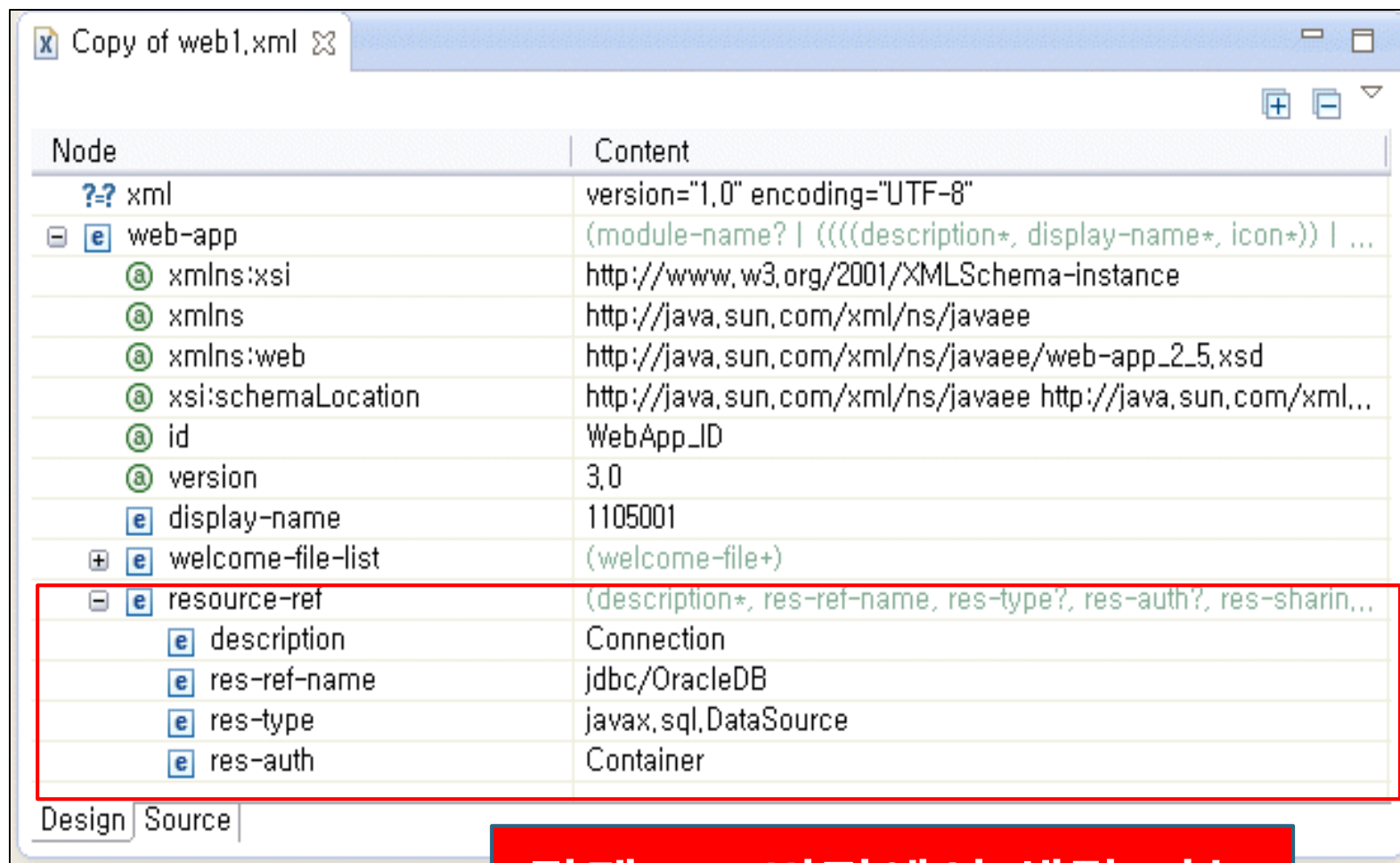
□ WEB-INF/web.xml

요 소	설 명
<res-ref-name>	context.xml 파일의 자원의 참조명
<res-type>	context.xml 파일의 자원의 타입
<res-auth>	context.xml 파일의 자원의 관리자

탐켓 7.0 버전에서 생략 가능

## 예제 11.2

예제 11.2와 같이 "web.xml" 파일에 resource-ref 노드를 추가하십시오. 단, res-ref-name은 예제 11.1을 참조한다.



Copy of web1.xml

Node	Content
xml	version="1.0" encoding="UTF-8"
web-app	(module-name?   (((description*, display-name*, icon*))   ...
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xmlns	http://java.sun.com/xml/ns/javaee
xmlns:web	http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd
xsi:schemaLocation	http://java.sun.com/xml/ns/javaee http://java.sun.com/xml...
id	WebApp_ID
version	3.0
display-name	1105001
welcome-file-list	(welcome-file+)
resource-ref	(description*, res-ref-name, res-type?, res-auth?, res-sharin...
description	Connection
res-ref-name	jdbc/OracleDB
res-type	javax.sql.DataSource
res-auth	Container

Design Source

**탐켓 7.0 버전에서 생략 가능**

## 3.4 DBCP을 이용한 데이터베이스 연동 프로그램

### 1단계

```
import="java.sql.*, javax.sql.*, javax.naming.*" %>
```

### 2,3단계

- ① `Context init = new InitialContext();`
- ② `Context dct = (Context)init.lookup("java:/comp/env");`
- ③ `DataSource ds = (DataSource)dct.lookup("jdbc/OracleDB");`
- ④ `Connection con = ds.getConnection();`

또는

- ① `Context init = new InitialContext();`
- ② `DataSource ds = (DataSource)init.lookup("java:/comp/env/jdbc/OracleDB");`
- ③ `Connection con = ds.getConnection();`

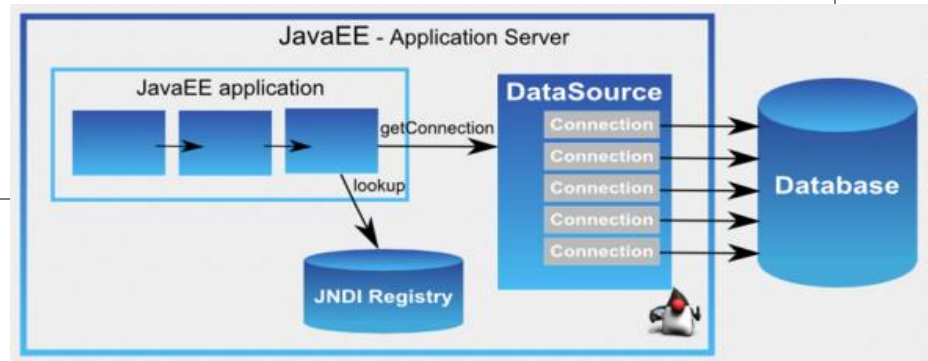
## 3.4 DBCP을 이용한 데이터베이스 연동 프로그램

### ▶ JSP 프로그램 코딩 예

```
<%@ page contentType="text/html; charset=utf8" pageEncoding="utf-8"
    pageEncoding="utf-8"
    import="java.sql.*,javax.sql.*,javax.naming.*" %>

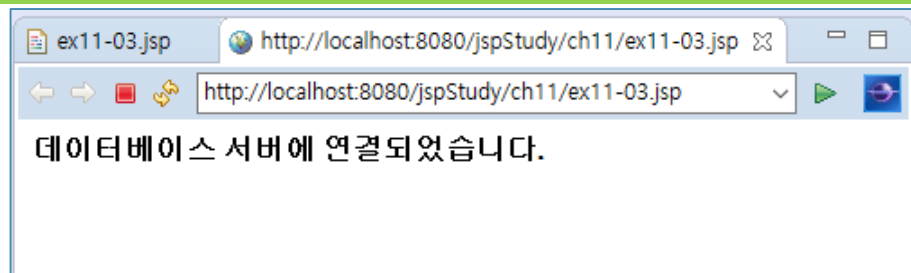
<%
try {
    Context init      = new InitialContext();
    DataSource ds      = (DataSource)init.lookup("java:/comp/env/jdbc/OracleDB");
    Connection con     = ds.getConnection();
    out.println("연결 성공 !!!");

} catch(SQLException e) {
    out.println("연결에 실패 !!!");
    e.printStackTrace(); }
%>
```



## 예제 11.3

"context.xml" 파일에 등록한 JNDI의 정보로 데이터베이스 서버 접속 유무를 출력하는 프로그램을 작성하시오.



```
1 <%@ page language="java" contentType="text/html; charset=utf-8"
2   pageEncoding="utf-8"
3   import="java.sql.*, javax.sql.*, javax.naming.*" %>
4 <%
5 try {
6     Context init = new InitialContext();
7     DataSource ds =
8         (DataSource) init.lookup("java:/comp/env/jdbc/OracleDB");
9     Connection con = ds.getConnection();
10    out.println("<h4>데이터베이스 서버에 연결되었습니다.</h4>");
11 } catch (Exception e) {
12     out.println("<h4>데이터베이스 서버에 연결이 되지 않았습니다.</h4>");
13     e.printStackTrace();
14 }
15 %>
```

## 예제 11.4

DBCP 기법을 이용하여 Course 테이블에서 추가 수강료가 0보다 큰 과목을 검색하여 출력하는 프로그램을 작성하시오.

ex11-04.jsp Course 테이블 검색

http://localhost:8080/jspStudy/ch11/ex11-04.jsp

순번	과목코드	과목명	학점 수	교수번호	추가수강료
1	L1012	웹디자인	2	null	20000
2	L1031	SQL	3	P12	30000
3	L1042	Delphi	3	P13	50000
4	L1052	전자상거래	3	P22	30000
5	L2031	게임이론	3	P23	50000

```
SELECT * FROM Course WHERE course_fees > 0;
```



```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"
3   import="java.sql.*, javax.sql.*, javax.naming.*" %>
4 <%
5   int i=0;
6   String sql="SELECT * FROM Course WHERE course_fees > 0";
7   try {
8     Context init = new InitialContext();
9     DataSource ds =
10 (DataSource) init.lookup("java:comp/env/jdbc/OracleDB");
11     Connection con = ds.getConnection();
12
13     Statement stmt = con.createStatement();
14     ResultSet rs = stmt.executeQuery(sql);
15 %>
16 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "">
17 <html>
18 <head>
19 <title>Course 테이블 검색</title>
20 </head>
21 <body>
22 <center>
23 <table border="1" cellspacing="1">
24 <tr>
25 <th colspan="6">Course 테이블</th>
26 </tr>
27 <tr>
28 <th>순번</th>
29 <th>과목코드</th>
30 <th>과목명</th>
31 <th>학점수</th>
32 <th>교수번호</th>
33 <th>추가수강료</th>
34 </tr>

```

```

35 <%
36   while ( rs.next() ) {
37 %> <tr>
38 <td><%= ++i %></td>
39 <td><%= rs.getString(1) %> </td>
40 <td><%= rs.getString(2) %> </td>
41 <td><%= rs.getInt(3) %> </td>
42 <td><%= rs.getString(4) %> </td>
43 <td><%= rs.getInt(5) %> </td>
44 </tr>
45 <%
46   }
47   } catch(Exception e){
48     out.println("<h4>데이터 가져오기에 실패하였습니다.</h4>");
49     e.printStackTrace();
50   }
51 %>
52 </table></center>
53 </body>
54 </html>

```

## 예제 11.5

DBCP 기법을 이용하여 교수번호로 개설과목을 출력하는 프로그램을 작성하시오.

ex11-05.jsp 교수번호 입력 화면

교수별 개설과목 조회 입력 화면

조회할 담당교수번호 : P12

ex11-05.jsp

ex11-05-1.jsp

ex11-05.jsp 교수의 개설과목

http://localhost:8080/jspStudy/ch11/ex11-05-1.jsp

순번	과목코드	과목명	학점수	추가수강료
1	L1021	데이터베이스	2	0
2	L1031	SQL	3	30000

Course 테이블에서 검색되었습니다.

```
SELECT * FROM Course WHERE Professor_ID = ?;
```

```

1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2      pageEncoding="UTF-8"%>
3  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
4  <html>
5  <head>
6  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7  <title>교수번호 입력화면</title>
8  <script language="JavaScript">
9      function In_Check() {
10         if (document.proinput.professor_id.value == "" ){
11             alert("담당교수를 입력하세요!!!");
12             return;
13         }
14         document.proinput.submit();
15     }
16 </script>
17 </head>
18 <body>
19     <center><h4>교수별 개설과목 조회 입력 화면</h4>
20     <form method="post" action="ex11-05-1.jsp" name="proinput">
21         <table border="1" cellspacing="1">
22             <tr>
23                 <td>조회할 담당교수번호 : </td>
24                 <td><input type="text" name="professor_id" size=5</td>
25             </tr>
26             <tr>
27                 <td colspan=2 align="center">
28                     <input type="button" name="confirm" value="조회" OnClick="In_Check()">
29                     <input type="reset" name="reset" value="취소">
30                 </td>
31             </tr>
32         </table>
33     </form>
34 </center>
35 </body>
36 </html>

```

ex11-05.jsp

```

1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2    pageEncoding="UTF-8"
3    import="java.sql.*, javax.sql.*, javax.naming.*" %>
4  <% request.setCharacterEncoding("UTF-8"); %>
5  <%
6    int i = 0;
7    String sql = "SELECT * FROM Course WHERE Professor_ID=?";
8    String professor_id = request.getParameter("professor_id");
9
10   try {
11     Context init = new InitialContext();
12     DataSource ds = (DataSource)init.lookup("java:comp/env/jdbc/OracleDB");
13     Connection con = ds.getConnection();
14
15     PreparedStatement pstmt = con.prepareStatement(sql);
16     pstmt.setString(1, professor_id);
17     ResultSet rs = pstmt.executeQuery();
18   } %>
19  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" ["http://www.w3.org/TR/html4/loose.dtd"]>
20  <html>
21  <head>
22    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
23    <title>교수의 개설과목</title>
24  </head>
25  <body>
26    <center>
27      <table border="1" cellspacing="1">
28        <tr>
29          <th colspan="5"><%= professor_id %> 교수의 개설과목</th>
30        </tr>
31        <tr>
32          <th>순번</th>
33          <th>과목코드</th>
34          <th>과목명</th>
35          <th>학점수</th>
36          <th>추가수강료</th>
37        </tr>

```

ex11-05-1.jsp

```

38  <%
39    while(rs.next()) {
40  >
41    <tr>
42      <td><%= ++i %></td>
43      <td><%= rs.getString(1) %></td>
44      <td><%= rs.getString(2) %></td>
45      <td><%= rs.getInt(3) %></td>
46      <td><%= rs.getInt(5) %></td>
47    </tr>
48  <%
49    }
50  > </table>
51  <p>Course 테이블에서 검색되었습니다.</center>
52  <%
53    } catch (Exception e) {
54      out.println("<h4> 데이터 가져오기에 실패하였습니다.</h4>");
55      e.printStackTrace();
56    }
57  >
58  </body>
59  </html>

```

# 과제

- ▶ 연습문제 1번, 2번, 3번, 4번을 작성하여 제출하시오.
  - ▶ 7장 예제 소스를 복사하여, DBCP 기법으로 수정함.
- ▶ 예제 소스 코드명
  - ▶ 예제 1번 : app11-01.jsp
  - ▶ 예제 2번 : app11-02.jsp, app11-02-2.jsp
  - ▶ 예제 3번 : app11-03.jsp, app11-03-1.jsp, app11-03-2.jsp
  - ▶ 예제 4번 : app11-04.jsp, app11-04-1.jsp, app11-04-3.jsp
- ▶ 제출방법
  - ▶ [YCLASS][과제제출]에 업로드
  - ▶ 제출파일 : pdf(실행결과 포함)
    - ▶ 추가로 전체 소스코드 제출 : 학번-11.zip으로 압축하여 제출
  - ▶ 수강 및 실습시 애로사항 또는 건의사항 등 기술

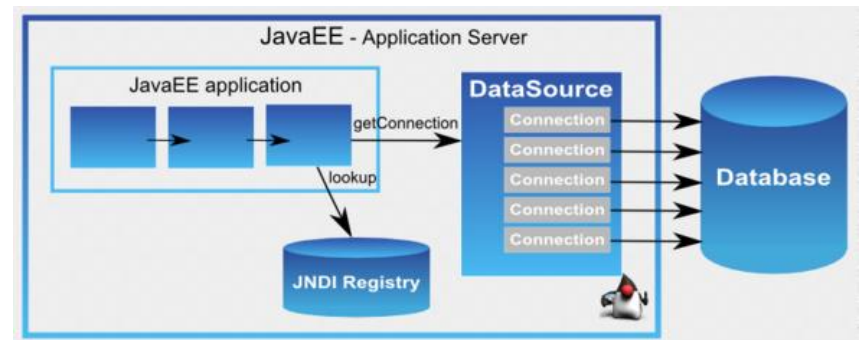
# 강 의 내 용 요약 정 리

## ▶ DBCP(DataBase Connection Pool)

- ▶ 데이터베이스와 연결된 JNDI(Java Naming and Directory Interface)라는 커넥션(Connection)을 미리 만들어 풀(pool)속에 저장해 두고 있다가 필요할 때 커넥션 풀에서 가져와 쓰고, 사용한 후에 다시 풀에 반환하는 기법

## ▶ 장점

- ▶ 데이터베이스 접속에 필요한 커넥션을 미리 생성하여 시간 절약
- ▶ 커넥션 재사용으로 생성되는 커넥션의 수가 많지 않아도 된다.
- ▶ 데이터베이스 서버 주소, 데이터베이스 드라이버, 사용자명, 암호의 보안 유지



# 강 의 내 용 요약 정 리

- ▶ JNDI 등록과 DBCP 이용한 데이터베이스 연동 작업
  - ▶ ① tomcat-dbc.jar의 DBCP 관련 파일 준비
  - ▶ ② 데이터베이스 서버, 계정, 암호 등의 "context.xml" 파일 생성
  - ▶ ③ 자원 참조명(resource-ref)의 정보를 "web.xml" 파일에 추가
- ▶ WebContent/META-INF/context.xml 파일 생성 [Oracle 예]

Node	Content
?? xml	version="1.0" encoding="UTF-8"
[-] [e] Context	
[-] [e] Resource	
[a] name	jdbc/OracleDB
[a] auth	Container
[a] type	javax.sql.DataSource
[a] driverClassName	oracle.jdbc.driver.OracleDriver
[a] url	jdbc:oracle:thin:@220.67.2.3:1521:ora11
[a] username	stud140
[a] password	pass140
[a] factory	org.apache.tomcat.dbcp.dbcp.BasicDataSourceFactory
[a] maxActive	10
[a] maxIdle	5

Design Source

# 강 의 내 용 요약 정 리

DBCP을 이용한 데이터베이스 연동 프로그램

1단계

```
import="java.sql.*, javax.sql.*, javax.naming.*" %>
```

2,3단계

- ㉠ Context init = new InitialContext();
- ㉡ DataSource ds = (DataSource)init.lookup("java:/comp/env/jdbc/OracleDB");
- ㉢ Connection con = ds.getConnection();





장 주제

12장. EL

