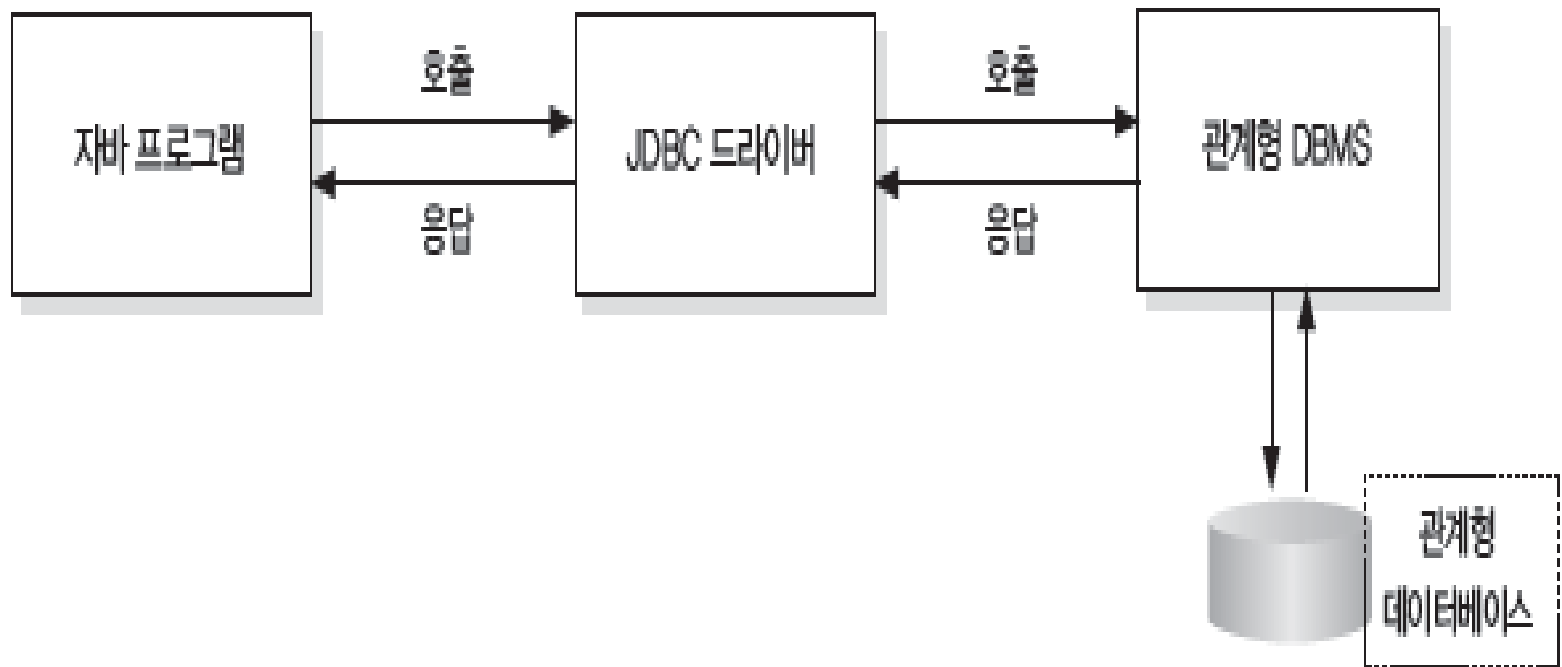


10장. Database

강사 강태광

JDBC

1. 데이터베이스(database)는 파일과 마찬가지로 보조기억장치에 데이터를 저장하는 수단입니다.
2. 자바 프로그램에서 데이터베이스를 사용하려면 JDBC 드라이버가 필요합니다



JDBC

1. JDBC 드라이버 : DBMS와 통쉬을 담당하는 자바 클래스
2. DBMS 별로 알맞은 JDBC 드라이버 필요
보통 jar 파일로 제공
2. JDBC 드라이버 로딩
 - DBMS와 통신하기 위해서는 먼저 로딩해 주어야 함
 - 로딩 코드
 - Class.forName("JDBC드라이버 클래스의 완전한 이름");
 - 주요 DBMS의 JDBC 드라이버
 - MySQL - com.mysql.jdbc.Driver
 - 오라클 - oracle.jdbc.driver.OracleDriver
 - MS SQL 서버 - com.microsoft.sqlserver.jdbc.SQLServerDriver

JDBC

3, DBMS URL: DBMS와의 연결을 위한 식별 값

4. JDBC 드라이버에 따라 형식 다름

5. 일반적인 구성

jdbc:[DBMS]:[데이터베이스식별자]

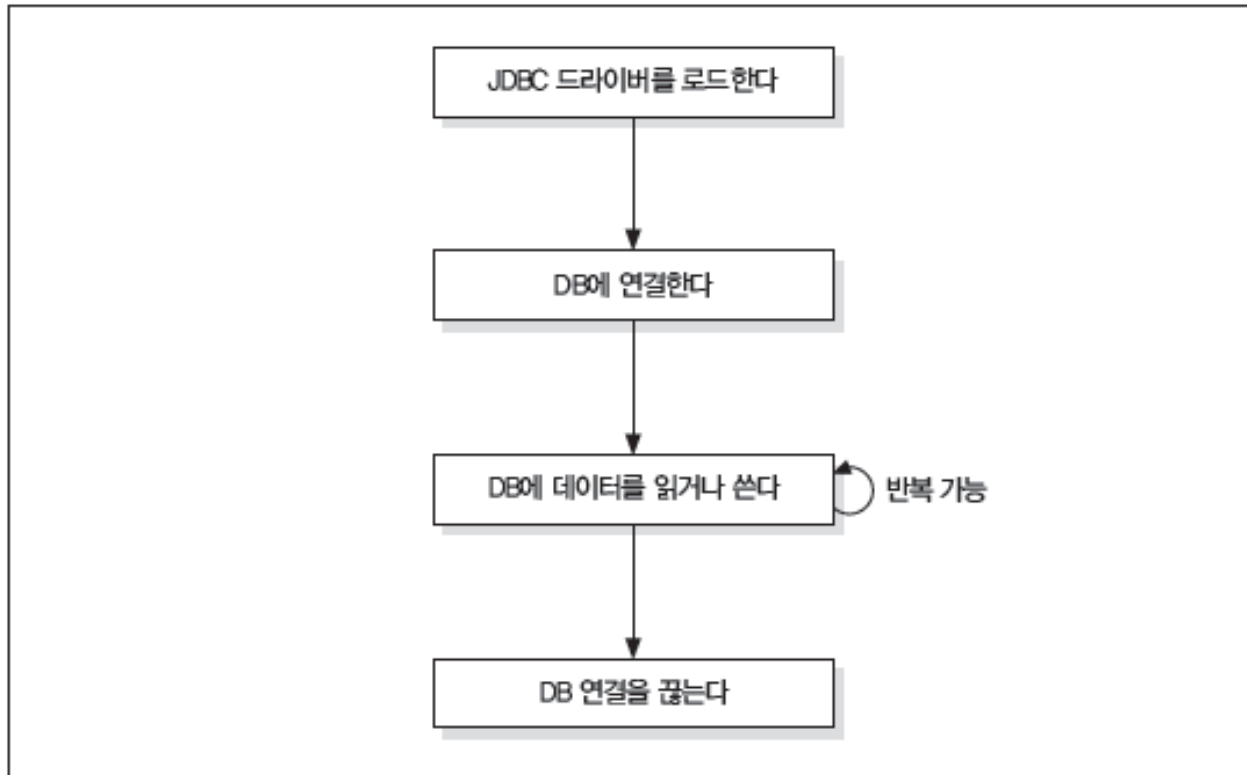
6. 주요 DBMS의 JDBC URL 구성

MySQL : jdbc:mysql://HOST[:PORT]/DBNAME

Oracle: jdbc:oracle:thin:@HOST:PORT:SID

MSSQL : jdbc:sqlserver://HOST[:PORT];databaseName=DB

데이터베이스 사용



데이터베이스 사용

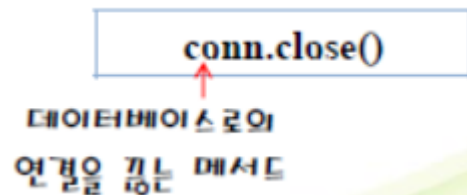
1. JDBC 드라이버를 로드하고, 데이터베이스로 연결하고, 연결을 끊는 방법

- 1] 자바 프로그램에서 관계형 데이터베이스로 연결을 하기 위해서는 먼저 다음과 같은 방법으로 **JDBC 드라이버를 로드해야 한다.**
- 2] `Class.forName('oracle.jdbc.driver.OracleDriver');`
Mysql : `com.mysql.jdbc.Driver`
- 3] 데이터베이스로 연결을 하기 위해서는 먼저 프로토콜, 서브프로토콜, 서브네임으로 이루어진 데이터베이스 **URL을 알아두어야 한다.**
 - Mysql : `jdbc:mysql://localhost:3306/db명`
 - `"jdbc:oracle:thin:@127.0.0.1:1521:orcl";`

2. 서브네임의 작성 방법은 DBMS의 종류마다 다르므로 다른 DBMS를 사용할 때는 관련 매뉴얼을 찾아보아야 한다.

데이터베이스 사용

3. JDBC 드라이버를 로드하고, 데이터베이스로 연결하고, 연결을 끊는 방법
- 데이터베이스로 연결을 맺기 위해서는 **java.sql.DriverManager** 클래스의 **getConnection** 메서드를 호출해야 한다.
 - `con = DriverManager.getConnection(url,"scott","tiger");`
 - 이 메서드는 데이터베이스로의 연결에 성공하면 **java.sql.Connection** 인터페이스 타입의 객체를 만들어서 리턴.
 - 데이터베이스로의 연결을 끊기 위해서는 **Connection** 객체에 대해 **close** 메서드를 호출



JDBC 드라이버를 로드하고, 데이터베이스로 연결하고, 연결을 끊는 방법

JDBC 드라이버의 로드, 데이터베이스로 연결하고 연결 끊기 DBTest.jsp

```
<%@page contentType= "text/html; charset=euc-kr "%>
<%@page import= "java.sql.* "%>
<HTML>
<HEAD> <TITLE> 데이터베이스로 연결하기 </TITLE> </HEAD>
<BODY>
<H3> 데이터베이스 연결 테스트 </H3>
<%
Class.forName( "oracle.jdbc.driver.OracleDriver");
Connection conn =
DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1:1521:orcl",
"scott ", "tiger ");
if (conn != null) {
out.println( "webdb 데이터베이스로 연결했습니다.<BR> ");
conn.close();
out.println( "webdb 데이터베이스로의 연결을 끊었습니다.<BR> ");
}
else {
out.println( "webdb 데이터베이스로 연결할 수 없습니다.<BR> ");
}
%>
</BODY>
</HTML>
```


데이터베이스의 데이터를 읽어오는 방법

1. 데이터베이스에 있는 데이터를 읽어오려면 우선 **Connection** 객체에 대해 **createStatement** 메서드를 호출해서 **java.sql.Statement** 타입 객체를 구해야 함

```
Statement stmt = conn.createStatement();
```

getConnection 메서드가
리턴한 Connection 객체

Statement 객체를
만들어서 리턴하는 메서드

2. Statement 객체에 대해 **executeQuery** 메서드를 호출하면 데이터베이스에 있는 데이터를 읽어올 수 있다.

```
ResultSet rs = stmt.executeQuery( "select * from goodsinfo where code='10002'; ");
```

select 문을 실행하는 메서드

데이터베이스의 데이터를 읽어오는 방법

3. `executeQuery` 메서드가 리턴한 `ResultSet` 객체에 대해 `next` 메서드를 호출하면 데이터베이스로부터 읽은 데이터를 순서대로 가져올 수 있다.
 - 이 메서드는 다음 위치에 데이터가 있을 때는 `True`, 없을 때는 `False`를 리턴

```
boolean exists = rs.next();
```

데이터베이스로부터 읽은 데이터의
첫 번째/다음 행 위치로 이동하는 메서드

4. `next` 메서드를 호출한 다음에 `ResultSet` 객체에 대해 `getInt`, `getString`, `getFloat` 등의 메서드를 호출하면 특정 데이터 항목 값을 가져올 수 있다

```
String code = rs.getString("code");
```

문자 데이터를 가져오는 메서드

```
int price = rs.getInt("price");
```

정수 데이터를 가져오는 메서드

oraTest3.jsp

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"    import="java.sql.*"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html> <head> <meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title> </head>
<body>
    <%
        String dno = request.getParameter("deptno");
        String sql = String.format("select * from dept where deptno=%s",dno);
        String driver = "oracle.jdbc.driver.OracleDriver";
        String url = "jdbc:oracle:thin:@localhost:1521:xe";
        try {
            Class.forName(driver);
            Connection conn = DriverManager.getConnection(url, "scott", "tiger");
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery(sql);
            if (rs.next()) {
                request.setAttribute("deptno", rs.getInt(1));
                request.setAttribute("dname", rs.getString(2));
                request.setAttribute("loc", rs.getString(3));
            }
            Else    out.println("그런 부서 없는데 ㅠㅠ");
            rs.close(); stmt.close(); conn.close();
            RequestDispatcher rd = request.getRequestDispatcher("oracleTest3Result.jsp");
            rd.forward(request, response); }
        catch(Exception e) {
            out.println("DB 연결에 실패했습니다" + e.getMessage());
        }
    %>
</body> </html>
```

oraTest3resul.jsp

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
pageEncoding="EUC-KR"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=EUC-
    KR">
    <title>Insert title here</title>
  </head>
  <body>
    <H2>부서정보</H2>
    부서코드 : ${dno}<p>
    부서명 : ${dname}<p>
    전화번호 : ${phone}<p>
    근무지 : ${position}
  </body>
</html>
```

데이터베이스에 데이터를 입력하는 방법

회원 정보를 입력받는 HTML 문서 dataInput.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html> <head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title>
<script type="text/javascript">
function chk() {
    if (!frm.deptno.value) { alert("부서코드를 입력하십시오");
    frm.deptno.focus(); return false;
    }
    if (!frm.dname.value) { alert("부서명을 입력하십시오");
    frm.dname.focus(); return false;
    }
    if (!frm.loc.value) { alert("근무지를 입력하십시오");
    frm.loc.focus(); return false;
    }
    return true;
}
</script> </head>
```

데이터베이스에 데이터를 입력하는 방법

회원 정보를 입력받는 HTML 문서 dataInput.html

```
<
<body>
  <h2>부서 정보를 입력하세요</h2>
  <form action="oraDeptInput2.jsp" name="frm" onsubmit="chk()">
    부서코드 : <input type="text" name="deptno"> <p>
    부서명 : <input type="text" name="dname"> <p>
    근무지 : <input type="text" name="loc"> <p>
    <input type="submit" value="입력완료"> <p>
    <input type="reset" value="입력취소"> <p>
  </form>
</body>
</html>
```

데이터베이스에 데이터를 입력하는 방법

회원 정보를 데이터베이스에 입력하는 JSP 페이지 **oraDeptInput.jsp**

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
pageEncoding="EUC-KR" import="java.sql.*"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html> <head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title> </head>
<body>
    <%
        String deptno = request.getParameter("deptno");
        String dname = request.getParameter("dname");
        String loc = request.getParameter("loc");
        String sql = String.format("insert into dept values(%s,%s,%s)",
            deptno, dname, loc);
        String driver = "oracle.jdbc.driver.OracleDriver";
        String url = "jdbc:oracle:thin:@localhost:1521:xe";
        Connection conn = null; Statement stmt = null;
        try { Class.forName(driver);
            conn=DriverManager.getConnection(url,"scott","tiger");
            stmt = conn.createStatement();
            int result = stmt.executeUpdate(sql);
            out.println("입력 성공 ㅋㅋㅋㅋ");
        } catch(Exception e) {
            out.println("입력실패 ㅠㅠ 메세지 :"+e.getMessage());
        } finally { stmt.close(); conn.close();
        }
    %>
</body> </html>
```

데이터베이스에 데이터를 입력하는 방법

회원 정보를 데이터베이스에 입력하는 JSP 페이지 oraDeptInput2.jsp

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
pageEncoding="EUC-KR" import="java.sql.*"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html> <head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title> </head>
<body>
    <%
        int deptno = Integer.parseInt(request.getParameter("deptno"));
        String dname = request.getParameter("dname");
        String loc = request.getParameter("loc");
        String sql = "insert into dept values(?,?,?)";
        String driver = "oracle.jdbc.driver.OracleDriver";
        String url = "jdbc:oracle:thin:@localhost:1521:xe";
        Connection conn = null; PreparedStatement pstmt = null;
        try {    Class.forName(driver);
                conn=DriverManager.getConnection(url,"scott","tiger");
                pstmt = conn.prepareStatement(sql);
                pstmt.setInt(1, deptno); pstmt.setString(2, dname);
                pstmt.setString(3, loc);
                int result = pstmt.executeUpdate();
                out.println("입력 성공 ㅋㅋㅋㅋ");
            }catch(Exception e) {
                out.println("입력실패 ㅠㅠ 메세지 :"+e.getMessage());
            } finally { pstmt.close(); conn.close();
            }
    %>
</body> </html>
```


데이터베이스의 데이터를 수정하고 삭제하는 방법

1. 데이터베이스에 있는 데이터를 수정할 때는 `executeUpdate` 메서드에 `update` 문을 파라미터로 넘겨줘야 한다.

```
int rowNum = stmt.executeUpdate( "update userinfo set password := "dalek"  
where id = "rose " );
```

2. 데이터베이스에 있는 데이터 삭제할 때는 `executeUpdate` 메서드에 `delete` 문을 파라미터로 넘겨줘야 한다.

```
int rowNum = stmt.executeUpdate( "delete from userinfo where id = 'rose ' " );
```

데이터베이스의 데이터를 수정하고 삭제하는 방법

상품코드를 입력받는 HTML 문서 dataInput3.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html> <head> <meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title>
<script type="text/javascript">
function chk() {
    if (!frm.deptno.value) { alert("부서코드를 입력하십시오");
    frm.deptno.focus(); return false;
    }
    return true;
}
</script> </head>
<body>
    <h2>수정할 부서코드를 입력하세요</h2>
    <form action="oraUpdateData.jsp" name="frm" onsubmit="chk()">
    부서코드 : <input type="text" name="deptno"> <p>
        <input type="submit" value="입력완료"> <p>
        <input type="reset" value="입력취소"> <p>
    </form>
</body>
</html>
```

데이터베이스의 데이터를 수정하고 삭제하는 방법

상품 정보 테이블을 읽는 JSP 페이지 `oraUpdateData.jsp`

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
pageEncoding="EUC-KR" import="java.sql.*"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html> <head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title> </head>
<body>
    <%
        String dno = request.getParameter("deptno");
        String sql = String.format("select * from dept where deptno=%s",dno);
        String driver = "oracle.jdbc.driver.OracleDriver";
        String url = "jdbc:oracle:thin:@localhost:1521:xe";
        try {
            Class.forName(driver);
            Connection conn = DriverManager.getConnection(url, "scott", "tiger");
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery(sql);
            if (rs.next()) { request.setAttribute("deptno", rs.getInt(1));
                            request.setAttribute("dname", rs.getString(2));
                            request.setAttribute("loc", rs.getString(3));
                        } else out.println("그런 부서 없는데 ㅠㅠ");
            rs.close(); stmt.close(); conn.close();
            RequestDispatcher rd = request.getRequestDispatcher("oraUpdateForm.jsp");
            rd.forward(request, response);
        } catch (Exception e) { out.println("DB 연결에 실패했습니다" + e.getMessage()); }
    %>
</body>
</html>
```

데이터베이스의 데이터를 수정하고 삭제하는 방법

oraUpdateForm.jsp

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
pageEncoding="EUC-KR"%><!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html><head><
meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title>
<script type="text/javascript">
function chk() {
    if (!frm.dname.value) { alert("부서명을 입력하십시오");
    frm.dname.focus(); return false; }
    if (!frm.loc.value) { alert("근무지를 입력하십시오");
    frm.loc.focus(); return false; }
    return true;
}
</script></head>
<body>
<h2>수정할 부서내용을 입력하세요</h2>
<form action="oraUpdate.jsp" name="frm" onsubmit="return chk()">
    <table border=1 bgcolor=yellow>
        <tr><td>부서코드 </td><td><input type="text"
name="deptno" value="${deptno}" readonly="readonly"> </td></tr>
        <tr><td>부서명 </td><td><input type="text" name="dname"
value="${dname}"> </td></tr>
        <tr><td>근무지 </td><td><input type="text"
name="loc" value="${loc}"> </td></tr>
        <tr><td><input type="submit" value="입력완료"> </td></tr>
    </table>
</form>
</body>
</html>
```

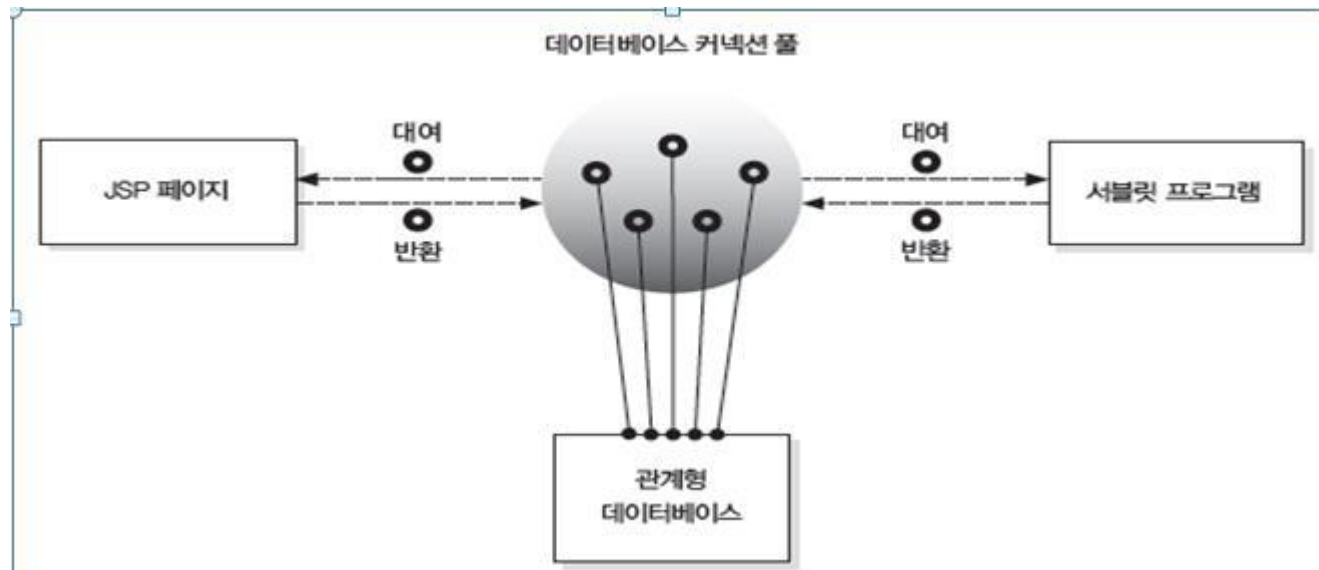
데이터베이스의 데이터를 수정하고 삭제하는 방법

oraUpdate.jsp

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
pageEncoding="EUC-KR" import="java.sql.*" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head> <meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title> </head>
<body>
    <%
        String deptno = request.getParameter("deptno");
        String dname = request.getParameter("dname");
        String loc = request.getParameter("loc");
        String sql = String.format("update dept set dname='%s', loc='%s' " +
            " where deptno=%s",dname, loc, deptno);
        String driver = "oracle.jdbc.driver.OracleDriver";
        String url = "jdbc:oracle:thin:@localhost:1521:xe";
        Connection conn = null; Statement stmt = null;
        try {
            Class.forName(driver);
            conn=DriverManager.getConnection(url,"scott","tiger");
            stmt = conn.createStatement();
            int result = stmt.executeUpdate(sql);
            out.println("수정 성공 ㅋㅋㅋㅋ");
        } catch (Exception e) { out.println("수정실패 ㅠㅠ 메세지 :"+e.getMessage());
        } finally {
            stmt.close(); conn.close();
        }
    %>
</body>
</html>
```

데이터베이스 커넥션 풀의 설치와 사용

1. 데이터베이스에 동시에 접속할 수 있는 사용자 수는 한정되어 있는데, 웹 서버에는 동시에 수백, 수천의 사용자들이 접속할 수 있다.
2. 웹 애플리케이션이 실행될 때마다 데이터베이스로 새로운 접속을 맺는 것은 현실적으로 불가능하므로, 데이터베이스 몇 개의 접속을 맺어서 데이터베이스 커넥션 풀(Database Connection Pool)에 저장해놓고, 필요한 웹 애플리케이션이 빌려 쓰고 반환하는 방식을 사용해야 한다.



커넥션 풀의 사용

1. context.xml 파일에 데이터베이스 연결과 관련된 코드를 resource로 작성하고 web.xml 파일에서 불러들인 후 사용 가능
2. Context.xml – WebContent 폴더의 META-INF 폴더에 작성

<Context>

<Resource name="리소스 이름"

auth="권한"

type="리소스 종류" - 데이터베이스는 javax.sql.DataSource

username="계정"

password= " 비밀번호"

driverClassName="드라이버이름"

factory="팩토리클래스 이름"

url= " 자원의 위치"

maxActive="최대사용개수"

maxIdle="사용되지 않고 풀에 저장될 수 있는 최대 커넥션 개수"/>

minIdle="사용되지 않고 풀에 저장될 수 있는 최소 커넥션 개수"

whenExhaustedAction= " 커넥션 풀에서 가져올 수 없을 때의 동작으로 0이면

maxWait 만큼 기다리고 0이면 에러를 발생시키며 2이면 일시적으로 커넥션을 생성해서 사용" maxWait="대기시간으로 1/1000초 단위"

timeBetweenEvictionRunsMills="사용되지 않는 커넥션의 추출 주기"

/>

</Context>

커넥션 풀의 사용

3.web.xml에서 앞에서 작성한 context.xml의 내용 사용하기

```
<resource-ref>  
    <description>설명</description>  
    <res-ref-name>리소스이름</res-ref-name>  
    <res-type>리소스 종류</res-type>  
    <res-auth>권한</res-auth>  
</resource-ref>  
JSP나 서블릿에서 사용
```

```
Context 변수1= new InitialContext();  
DataSource 변수2 = (DataSource) 변수1.lookup("java:comp/env/리소스이름");  
Connection변수 = 변수.getConnection();
```


context.xml 활용 : META-INF에 context.xml을 작성하여 넣음

1

```
<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.dbcp.BasicDataSourceFactory"  
    url="jdbc:oracle:thin:@127.0.0.1:1521:ORCL"  
    maxActive="500"  
    maxIdle="100"/>
```

```
</Context>
```

%. 과거에는 web.xml에 다음을 추가했으나 현재는 생략해도 됨

```
<resource-ref>  
<description>Connection</description>  
<res-ref-name>jdbc/OracleDB</res-ref-name>  
<res-type>javax.sql.DataSource</res-type>  
<res-auth>Container</res-auth> </resource-ref>
```