

[Oracle]

oracle11g ex(OracleXEUniv.exe) 설치 후 SQLPlus에서 다음 실행

```
SQL> conn system/manager
SQL> @C:\Woraclexe\Wapp\Woracle\Wproduct\W11.2.0\server\WRDBMS\ADMIN\Wscott.sql
SQL> conn scott/tiger
SQL> select * from DEPT;
```

[JDBC]

JDBC(Java Database Connectivity)란 자바에서 데이터베이스를 사용하기 위해 필요한 API의 한 종류이다. Java.sql패키지에 있는 Connection, PreparedStatement, ResultSet, ResultSetMetaData, Statement, CallableStatement Interface들과 DriverManager 등의 클래스를 주로 사용

JDBC 사용하기

- JDBC 다운로드: 먼저 사용하고자 하는 DB 벤더사를 통해 JDBC를 다운로드 받는다.
Oracle의 경우, ojdbc14.jar파일을 jdk/jre/lib/ext 폴더 또는 WEB-INF/lib에 넣는다.
- JDBC드라이버 로딩
 - Class.forName(“드라이버명”);
 - : 매개변수로 넣은 클래스의 객체를 클래스 로더에게 전달해줌
 - 1. Oracle : oracle.jdbc.driver.OracleDriver
 - 2. SQL Servler : com.microsoft.jdbc.sqlserver.SQLServerDriver
- DB 접속 - Connection 객체 얻기
 - DriverManager.getConnection()메소드로 Connection객체를 얻음
 - 필요한 매개변수 형태
 - 1) URL : jdbc:<subprotocol>:<subname> 2) 사용자 ID 3) 사용자 암호
 - 예) "jdbc:oracle:thin:@127.0.0.1:1521:XE", "scott", "tiger"

```
package jdbcex;
import java.sql.*;

public class DBManager {
    public static Connection getConnection() {
        Connection conn = null;
        String driver = "oracle.jdbc.driver.OracleDriver";
        try { Class.forName(driver);
            String url = "jdbc:oracle:thin:@127.0.0.1:1521:xe";
            String user = "SCOTT";
            String password = "TIGER";
            conn = DriverManager.getConnection(url, user, password);
            System.out.println("연결 성공");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```

    }
    return conn;
}
public static void main(String[] args) {
    Connection conn = getConnection();
}
}

```

[VO(Value Object) or DTO(Data Transfer Object)]

테이블의 한 레코드를 저장하기 위한 용도의 객체



```
DeptVO vo = new DeptVO();
```

```
vo.setDeptNo(10); vo.setDname("ACCOUNTING"); vo.setLoc("NEW YORK");
```

```
package jdbcex.dept;
```

```
public class DeptVO {
    private int deptNo;
    private String dname;
    private String loc;

```

/*getter 메소드 용도: 필드(멤버 변수)의 값을 반환.

메소드 이름: get+필드명(첫글자 대문자로), 반환형: 필드타입, 매개인자없음 */

```

    public int getDeptNo() {
        return deptNo;
    }
    public String getDname() {
        return dname;
    }
    public String getLoc() {
        return loc;
    }

```

/*setter 메소드 용도: 필드(멤버 변수)의 값을 매개인자로 받은 값으로 변경.

메소드 이름: set+필드명(첫글자 대문자로), 반환형:void, 매개인자 : 필드타입 */

```

    public void setDeptNo(int deptNo) {
        this.deptNo = deptNo;
    }
    public void setDname(String dname) {
        this.dname = dname;
    }
}

```

```

    public void setLoc(String loc) {
        this.loc = loc;
    }
}

```

[DAO(Data Access Object) : 테이블에 접근하여 select, insert, delete, update 쿼리를 수행하는 메소드들로 구성되어 있는 객체]

```

package jdbcex.dept;

import java.sql.*;
import java.util.ArrayList;
import jdbcex.DBManager;
import jdbcex.dept.DeptVO;

public class DeptDAO {
    /* 매개인자로 받은 deptNo(부서번호)를 조회 조건으로 하여
    Dept 테이블에서 한 건의 데이터를 select한 후, 그 결과를 DeptVO 객체에 담아 반환 */
    public DeptVO getDept(int deptNo) {
        Connection conn = DBManager.getConnection();
        DeptVO vo = null;
        try {
            PreparedStatement stmt = null;
            ResultSet rs = null;
            stmt = conn.prepareStatement("SELECT  dname, loc FROM dept WHERE
deptno = ? ");
            stmt.setInt(1, deptNo);
            rs = stmt.executeQuery();
            while (rs.next()) {
                vo = new DeptVO();
                String dname = rs.getString("dname");
                String loc = rs.getString("loc");
                vo.setDeptNo(deptNo);
                vo.setDname(dname);
                vo.setLoc(loc);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            try { conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        return vo;
    }

    /* Dept 테이블의 모든 레코드를 select하여 한 건씩 DeptVO 객체에 담고,
    ArrayList에 각 DeptVO 객체를 저장하여 ArrayList를 반환 */
    public ArrayList<DeptVO> getDeptList() {
        Connection conn = DBManager.getConnection();
        ArrayList<DeptVO> arrayList = new ArrayList<DeptVO>();
        DeptVO vo = null;
        try {
            PreparedStatement stmt = null;

```

```

        ResultSet rs = null;
        stmt=conn.prepareStatement("SELECT deptNo, dname, loc FROM
dept");

        rs = stmt.executeQuery();
        while (rs.next()) {
            vo = new DeptVO();
            int deptNo = rs.getInt("deptNo"); // rs.getInt(1);
            String dname = rs.getString("dname");
            String loc = rs.getString("loc");
            vo.setDeptNo(deptNo);
            vo.setDname(dname);
            vo.setLoc(loc);
            arrayList.add(vo);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        try { conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    }
    return arrayList;
}

/* 매개인자로 받은 DeptVO 객체의 내용을 Dept테이블에 저장 */
public int insertDept(DeptVO vo) {
    Connection conn = DBManager.getConnection();
    int resultCnt = 0 ;
    try {
        PreparedStatement stmt = null;
        stmt = conn.prepareStatement("insert into dept ( deptNo, dname,
loc ) values (?,?,?) ");
        stmt.setInt(1,vo.getDeptNo());
        stmt.setString(2,vo.getDname());
        stmt.setString(3,vo.getLoc());
        resultCnt = stmt.executeUpdate();

    } catch (SQLException e) {
        e.printStackTrace();
    } finally{
        try { conn.close();
        }catch (SQLException e) { e.printStackTrace(); }
    }
    return resultCnt;
}

/* 매개 인자로 받은 deptNo 값을 갖는 레코드를 Dept테이블에서 삭제 */
public int deleteDept(int deptNo) {
    Connection conn = DBManager.getConnection();
    int resultCnt = 0 ;
    try {
        PreparedStatement stmt = null;
        stmt = conn.prepareStatement("delete from dept where deptNo
=?");

        stmt.setInt(1,deptNo);
        resultCnt = stmt.executeUpdate();

```

```

    } catch (SQLException e) {
        e.printStackTrace();
    } finally{ try { conn.close();
    }catch (SQLException e) { e.printStackTrace(); }
    }
    return resultCnt;
}

/* 매개인자로 받은 DeptVO 객체에서 deptno 값을 갖는 레코드의 내용을
매개인자로 받은 DeptVO 객체의 dname과 loc 값으로 갱신 */
public int updateDept(DeptVO vo) {
    Connection conn = DBManager.getConnection();
    int resultCnt = 0 ;
    try { PreparedStatement stmt = null;
        stmt = conn.prepareStatement("update dept set dname=? , loc=?
where deptNo = ? ");
        stmt.setString(1,vo.getDname());
        stmt.setString(2,vo.getLoc());
        stmt.setInt(3, vo.getDeptNo());
        resultCnt = stmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally{ try { conn.close();
    }catch (SQLException e) { e.printStackTrace(); }
    }
    return resultCnt;
}

public static void main(String[] args) {
    DeptDAO deptDao = new DeptDAO();
    //10번 부서 정보 select
    DeptVO vo1 = deptDao.getDept(10);
    System.out.println(vo1.getDeptNo()+":"+vo1.getDname()+":"+vo1.getLoc());

    //모든 부서 정보 select
    ArrayList<DeptVO> list = deptDao.getDeptList();
    for(DeptVO vo2 : list){
        System.out.println(vo2.getDeptNo()+":"+vo2.getDname()+":"+vo2.getLoc());
    }
    //DeptNo: 51 , dname: test , loc : Incheon 레코드를 insert
    DeptVO vo3 = new DeptVO();
    vo3.setDeptNo(51); vo3.setDname("test"); vo3.setLoc("Incheon");
    int insertCnt = deptDao.insertDept(vo3) ;
    System.out.println(insertCnt+" 건 insert 완료");

    //DeptNo: 51 , dname: test2, loc : Seoul 로 update
    DeptVO vo4 = new DeptVO();
    vo4.setDeptNo(51); vo4.setDname("test2"); vo3.setLoc("Seoul");
    int updateCnt =deptDao.updateDept(vo4);

```

```

        System.out.println(updateCnt+" 건 update 완료");

        //DeptNo가 51인 레코드 delete
        int deleteCnt =deptDao.deleteDept(51) ;
        System.out.println(deleteCnt+" 건 delete 완료");
    }
}

```

WebContent/jdbc/insertDept.html (부서 정보 등록 입력화면)

```

<!DOCTYPE html><html><head><meta charset="UTF-8">
<title> 부서 정보 등록 </title></head>
<body> <h1>부서 등록 화면 입니다.</h1>
    <form action="insertDept.jsp" method="post">
        <table border="1">
            <tr>
                <td>등록할 부서명 :</td>
                <td><input type="text" name="dname"></td>
            </tr>
            <tr>
                <td>등록할 부서사번 (숫자) :</td>
                <td><input type="text" name="deptno"></td>
            </tr>
            <tr>
                <td>부서 위치</td>
                <td><input type="text" name="loc"></td>
            </tr>
        </table>
        <input type="submit" value="등록" />
    </form>
</body>
</html>

```

WebContent/jdbc/*insertDept.jsp* (부서 정보 등록 처리)

```
<%@ page import="jdbcex.dept.DeptDAO"%>
<%@ page import="jdbcex.dept.DeptVO"%>
<%@ page contentType="text/html; charset=UTF-8" %>
<%
    String dname = request.getParameter("dname");
    String deptno = request.getParameter("deptno");
    String loc = request.getParameter("loc");

    DeptVO vo = new DeptVO();
    vo.setDeptNo(Integer.parseInt(deptno));
    vo.setDname(dname);
    vo.setLoc(loc);

    DeptDAO dao = new DeptDAO();
    dao.insertDept(vo);
%>
```

WebContent/jdbc/deptList.jsp (부서 정보 보기)

```

<%@ page import="java.util.ArrayList" %>
<%@ page import="jdbcex.dept.DeptDAO"%>
<%@ page import="jdbcex.dept.DeptVO"%>
<%@ page contentType="text/html; charset=UTF-8" %>
<html><body>
    <table border="1">
        <tr>
            <th>부서번호</th>
            <th>부서이름</th>
            <th>위치</th>
        </tr>
        <%
            DeptDAO deptDao = new DeptDAO();
            ArrayList<DeptVO> list = deptDao.getDeptList();

            for (DeptVO vo : list) {
                out.println("<tr>");
                out.println( "<td>" + vo.getDeptNo() + "</td>"
                    + "<td>" + vo.getDname() + "</td>"
                    + "<td>" + vo.getLoc()+ "</td>");
                out.println("</tr>");
            }
        %>
    </table>
</body></html>

```