

JSP

[11] 효율적인 데이터베이스 관리

```
public class EmpDTO {  
  
    private int empno;  
    private String ename;  
    private String job;  
    private int mgr;  
    private Date hiredate; //데이터 베이스에 있는걸 뿌리지만 할거니 string으로 받아도 됨  
    private int sal;  
    private int comm;  
    private int deptno;  
  
    public EmpDTO() {}  
  
    public EmpDTO(int empno, String ename, String job, int mgr, Date hiredate, int sal, int comm, int deptno) {  
        super();  
        this.empno = empno;  
        this.ename = ename;  
        this.job = job;  
        this.mgr = mgr;  
        this.hiredate = hiredate;  
        this.sal = sal;  
        this.comm = comm;  
        this.deptno = deptno;  
    }  
  
    public int getEmpno() {  
        return empno;  
    }  
  
    public void setEmpno(int empno) {  
        this.empno = empno;  
    }  
  
    public String getEname() {  
        return ename;  
    }  
  
    public void setName(String ename) {  
        this.ename = ename;  
    }  
  
    public String getJob() {  
        return job;  
    }  
}
```

DTO 에서 변수선언하고, 생성자 함수 만들어 내기 , getter setter 만들기

```
public class EmpDAO { //싱글톤 DAO로 만들것  
  
    private String driver = "oracle.jdbc.driver.OracleDriver";  
    private String url = "jdbc:oracle:thin:@localhost:1521:xe";  
  
    private static EmpDAO instance; //초기값이 null로 잡힘 //instance는 자기가 자기 클래스형 객체를 참조하는 변수  
    public static EmpDAO getInstance() {  
        if(instance == null) {  
            instance = new EmpDAO(); // instance가 null 이면 객체를 생성해라  
        }  
        return instance;  
    }  
}
```

DAO 에서 driver, url 그리고 싱글톤 객체 만들기

```
//싱글톤으로 만들것. 생성자 함수를 만들거야
private EmpDAO() {
    try {
        Class.forName(driver);
    } catch (ClassNotFoundException e) {
        System.out.println(e.getMessage());
    } //catch
} //try
```

생성자 함수 만들고 드라이버 열기

```
public ArrayList<EmpDTO> selectSearchName(String searchName){
    ArrayList<EmpDTO> dtos = new ArrayList<EmpDTO>();
    //2단계 - 7단계
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    String sql = "SELECT * FROM EMP WHERE ENAME LIKE '%"||UPPER(?)||'%'";

    try {
        conn = DriverManager.getConnection(url,"scott","tiger");
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, searchName);
        rs = pstmt.executeQuery();

        while(rs.next()) {
            int empno = rs.getInt("empno");
            String ename = rs.getString("ename");
            String job = rs.getString("job");
            int mgr = rs.getInt("mgr");
            Date hiredate = rs.getDate("hiredate");
            int sal = rs.getInt("sal");
            int comm = rs.getInt("comm");
            int deptno = rs.getInt("deptno");
            dtos.add(new EmpDTO(empno, ename, job, mgr, hiredate, sal, comm, deptno));
            EmpDTO emp = new EmpDTO();
            emp.setEmpno(rs.getInt("empno"));
            emp.setEname(rs.getString("ename"));
            emp.setJob(rs.getString("job"));
            emp.setMgr(rs.getInt("mgr"));
            emp.setHiredate(rs.getDate("hiredate"));
            emp.setSal(rs.getInt("sal"));
            emp.setComm(rs.getInt("comm"));
            emp.setDeptno(rs.getInt("deptno"));
            dtos.add(emp);
        }
    } catch (SQLException e) {
        System.out.println(e.getMessage());
    } finally {
        try {
```

ArrayList 로 dto 객체 만들고 그안에 dao 넣어주기

```
body>
```

```
<%
//파라미터 먼저 받자
String searchName = request.getParameter("searchName");
if(searchName == null){ //searchName이 null이면 searchName을 빈스트링으로 만들어라
    searchName = "";
}
%>
```

```
<form action="">
<p>이름
<input type="text" name="searchName" value="<%=searchName.toUpperCase()%>">
<input type="submit" value="검색">
</p>
</form>
<table>
```

```
<%
EmpDAO dao = EmpDAO.getInstance();//dao 함수 호출하는 것
ArrayList<EmpDTO> emps = dao.selectSearchName(searchName);
out.println("<tr><th>사원</th><th>이름</th><th>직책</th>"
```

```
            + "<th>성서씨번</th><th>입사일</th><th>급하</th><th>상하</th><th>부서번호</th></tr>");
```

```
if(!emps.isEmpty()){
    for(EmpDTO emp: emps){
        out.println("<tr><td>" + emp.getEmpno() + "</td><td>" + emp.getEname() + "</td><td>" +
            emp.getJob() + "</td><td>" + emp.getMgr() + "</td><td>" + emp.getHiredate() +
            "</td><td>" + emp.getSal() + "</td><td>" + emp.getComm() + "</td><td>" +
            emp.getDeptno() + "</td></tr>");
    }
}
```

```
}else{
    out.println("<tr><td colspan='8'>해당 사원이 존재하지 않습니다</td></tr>");
}
```

```
%>
</table>
```