

미 니 프 로젝 트

(Web Programming)

학사관리 프로그램

학사관리 시스템

교수목록 교수등록 학생목록 학생등록 강좌목록 강좌등록

교 수 목 록

교수번호 ▼		3행 ▼	검색	검색수: 4건	교수번호 ▼	오름차순 ▼
교수번호	교수이름	교수학과	교수직급	임용일자	급여	교수정보
221	이병렬	전산	정교수	1975-04-03	3,000,000	교수정보
228	이재광	전산	부교수	1991-09-19	2,500,000	교수정보
311	강승일	전자	부교수	1994-06-09	2,300,000	교수정보

◀ ▶ [1/2]

Copyright © ICIA All Right Reserved.

학사관리 시스템

교수목록 교수등록 학생목록 학생등록 강좌목록 강좌등록

교 수 정 보

교수번호	228	교수학과	전산 ▼	임용일자	1991 년 09 월 19 일
교수이름	이재광	교수급여	2,500,000	교수직급	<input type="radio"/> 정교수 <input checked="" type="radio"/> 부교수 <input type="radio"/> 조교수

저장 삭제 취소

담 당 강 좌

강좌번호	강좌이름	강의실	강의시수	최대수강인원	수강신청인원	강좌정보
C312	자료구조	510	2	100	60	강좌정보
C421	알고리즘	510	3	80	72	강좌정보

담 당 학 생

학생번호	학생이름	학과	학년	생년월일	학생정보
92414029	서연우	전산	3	1973-10-06	학생정보
92414037	심청이	전산	1	1991-10-05	학생정보
92414038	김은경	전산	1	1991-04-05	학생정보
96414404	김수정	전산	3	1977-12-22	학생정보

Copyright © ICIA All Right Reserved.

학사관리 시스템

교수목록 교수등록 학생목록 학생등록 강좌목록 강좌등록

학 생 목 록

학생번호 ▼ 5행 ▼ 검색 검색수: 11건
 학생번호 ▼ 오름차순 ▼

학생번호	학생이름	학생학과	학년	생년월일	지도교수	학생정보
92414029	서연우	전산	3	1973-10-06	이재광 228	학생정보
92414033	김창덕	전산	4	1991-02-03	이병렬 221	학생정보
92414037	심청이	전산	1	1991-10-05	이재광 228	학생정보
92414038	김은경	전산	1	1991-04-05	이재광 228	학생정보
92414048	한호주	전산	1	1973-02-03	강승일 311	학생정보

◀ ▶ [1/3]

Copyright © ICIA All Right Reserved.

학사관리 시스템

교수목록 교수등록 학생목록 학생등록 강좌목록 강좌등록

학 생 정 보

학생번호	<input type="text" value="92414029"/>	학생학과	<input type="text" value="전산"/>
학생이름	<input type="text" value="서연우"/>	생년월일	<input type="text" value="1973"/> 년 <input type="text" value="10"/> 월 <input type="text" value="06"/> 일
학생학년	<input type="radio"/> 1학년 <input type="radio"/> 2학년 <input checked="" type="radio"/> 3학년 <input type="radio"/> 4학년	지도교수	<input type="text" value="228"/> <input type="text" value="이재광"/>

[저장](#)
[삭제](#)
[취소](#)

수 강 신 청

수강신청 과목: ▼ [수강신청](#)

강좌번호	강좌이름	수강신청일	담당교수	강의실	강의시수	수강취소
C301	파일처리론	2020-05-26	이병렬	506	3	수강취소
C401	데이터베이스	2018-12-02	이병렬	414	3	수강취소
C312	자료구조	2020-05-26	이재광	510	2	수강취소

Copyright © ICIA All Right Reserved.

학사관리 시스템

교수목록 교수등록 학생목록 학생등록 강좌목록 강좌등록

강좌 목록

강좌번호 ▼

3행 ▼

검색

검색수: 7건

강좌번호 ▼

오름차순 ▼

강좌번호	강좌이름	강의시간	강의실	담당교수	최대인원	수강인원	강좌정보
C312	자료구조	2	510	이재광	100	60	강좌정보
C401	데이터베이스	3	414	이병렬	80	80	강좌정보
C421	알고리즘	3	510	이재광	80	72	강좌정보

◀ ▶ [2/3]

Copyright © ICIA All Right Reserved.

학사관리 시스템

교수목록 교수등록 학생목록 학생등록 강좌목록 강좌등록

강좌 정보

강좌번호	C401	강의실	414	강의시수	3
강좌이름	데이터베이스			최대수강인원	80
담당교수	221	이병렬		수강신청인원	80

저장 삭제 취소

성적 입력

선택저장

<input type="checkbox"/>	학생번호	학생이름	학과	지도교수	학년	수강신청일	점수	
<input type="checkbox"/>	92414033	김창덕	전산	이병렬	4	2020-05-26	0	수정
<input type="checkbox"/>	95414058	박혜경	전산	이병렬	4	1998-03-03	90	수정
<input type="checkbox"/>	92414029	서연우	전산	이재광	3	2018-12-02	0	수정
<input type="checkbox"/>	92454018	이원구	건축	오문환	3	2018-07-21	0	수정
<input type="checkbox"/>	95454003	이재영	건축	오문환	4	2018-07-21	0	수정

Copyright © ICIA All Right Reserved.

- 데이터베이스 생성

쇼핑몰 user, database 생성

```
create user 'haksa'@'localhost' identified by 'pass';
create database haksadb;
grant all privileges on haksadb.* to 'haksa'@'localhost';
```

교수테이블 생성

```
create table professors(
    pcode char(3) not null,
    pname varchar(15) not null,
    dept varchar(30),
    hiredate date,
    title varchar(15),
    salary int default 0,
    primary key(pcode)
);
```

학생테이블 생성

```
create table students(
    scode char(8) not null,
    sname varchar(15) not null,
    dept varchar(30),
    year char(1),
    birthday date,
    advisor char(3),
    primary key(scode),
    foreign key(advisor) references professors(pcode)
);
```

강좌테이블생성

```
create table courses(
    lcode char(4) not null,
    lname varchar(50) not null,
    hours int, room char(3),
    instructor char(3),
    capacity int default 0,
    persons int default 0,
    primary key(lcode),
    foreign key(instructor) references professors(pcode)
);
```

수강신청테이블 생성

```
create table enrollments(
    lcode char(4) not null,
    scode char(8) not null,
    edate date, grade int default 0,
    primary key(lcode, scode),
    foreign key(lcode) references courses(lcode),
    foreign key(scode) references students(scode)
);
```

교수테이블 데이터 입력

```
insert into professors(pcode,pname,dept,hiredate,title,salary) values('221','이병렬','전산','75/04/03','정교수',3000000);
insert into professors(pcode,pname,dept,hiredate,title,salary) values('228','이재광','전산','91/09/19','부교수',2500000);
insert into professors(pcode,pname,dept,hiredate,title,salary) values('311','강승일','전자','94/06/09','부교수',2300000);
insert into professors(pcode,pname,dept,hiredate,title,salary) values('509','오문환','건축','92/10/14','조교수',2000000);
```

학생테이블 데이터 입력

```
insert into students(scode,sname,dept,year,birthday,advisor) values('92414029','서연우','전산',3,'73/10/06','228');
insert into students(scode,sname,dept,year,birthday,advisor) values('92414033','김창덕','전산',4,'73/10/26','221');
insert into students(scode,sname,dept,year,birthday,advisor) values('92514009','이지행','전자',4,'73/11/16','311');
insert into students(scode,sname,dept,year,birthday,advisor) values('92514023','김형명','전자',4,'73/08/29','311');
insert into students(scode,sname,dept,year,birthday,advisor) values('92454018','이원구','건축',3,'74/09/30','509');
insert into students(scode,sname,dept,year,birthday,advisor) values('95454003','이재영','건축',4,'76/02/06','509');
insert into students(scode,sname,dept,year,birthday,advisor) values('95414058','박혜경','전산',4,'76/03/12','221');
insert into students(scode,sname,dept,year,birthday,advisor) values('96414404','김수정','전산',3,'77/12/22','228');
```

강좌테이블 데이터 입력

```
insert into courses(lcode,lname,hours,room,instructor,capacity,persons) values('C301','파일처리론',3,'506','221',100,80);
insert into courses(lcode,lname,hours,room,instructor,capacity,persons) values('C401','데이터베이스',3,'414','221',80,80);
insert into courses(lcode,lname,hours,room,instructor,capacity,persons) values('C421','알고리즘',3,'510','228',80,72);
insert into courses(lcode,lname,hours,room,instructor,capacity,persons) values('C312','자료구조',2,'510','228',100,60);
insert into courses(lcode,lname,hours,room,instructor,capacity,persons) values('E221','논리회로',3,'304','311',100,80);
insert into courses(lcode,lname,hours,room,instructor,capacity,persons) values('A109','한국외건축문화',2,'101','509',120,36);
```

수강신청 테이블 데이터 입력

```
insert into enrollments(lcode, scode, edate, grade) values('C401','92414033','98/03/02',85);
insert into enrollments(lcode, scode, edate, grade) values('C301','92414033','98/03/02',80);
insert into enrollments(lcode, scode, edate, grade) values('C421','92414033','98/03/02',0);
insert into enrollments(lcode, scode, edate, grade) values('C401','95414058','98/03/03',90);
insert into enrollments(lcode, scode, edate, grade) values('C301','95414058','98/03/03',80);
insert into enrollments(lcode, scode, edate, grade) values('C312','95414058','98/03/03',80);
insert into enrollments(lcode, scode, edate, grade) values('C401','92514023','98/03/03',70);
insert into enrollments(lcode, scode, edate, grade) values('C301','92514023','98/03/03',70);
insert into enrollments(lcode, scode, edate, grade) values('C421','92514023','98/03/03',70);
insert into enrollments(lcode, scode, edate, grade) values('C301','92414029','98/03/03',90);
insert into enrollments(lcode, scode, edate, grade) values('C421','92414029','98/03/03',0);
insert into enrollments(lcode, scode, edate, grade) values('C312','92414029','98/03/03',70);
insert into enrollments(lcode, scode, edate, grade) values('E221','92414029','98/03/03',75);
insert into enrollments(lcode, scode, edate, grade) values('A109','92414029','98/03/03',90);
insert into enrollments(lcode, scode, edate, grade) values('C301','92514009','98/03/03',70);
insert into enrollments(lcode, scode, edate, grade) values('C401','92514009','98/03/03',85);
insert into enrollments(lcode, scode, edate, grade) values('E221','92514009','98/03/03',85);
insert into enrollments(lcode, scode, edate, grade) values('C301','96414404','98/03/04',75);
insert into enrollments(lcode, scode, edate, grade) values('C401','96414404','98/03/04',75);
insert into enrollments(lcode, scode, edate, grade) values('C421','96414404','98/03/04',75);
insert into enrollments(lcode, scode, edate, grade) values('C312','92454018','98/03/04',90);
insert into enrollments(lcode, scode, edate, grade) values('E221','92454018','98/03/04',90);
insert into enrollments(lcode, scode, edate, grade) values('A109','95454003','98/03/05',85);
insert into enrollments(lcode, scode, edate, grade) values('E221','95454003','98/03/05',85);
```

- 목록 출력을 위한 Stored Procedure

```
call list('professors', 'pcode', '', 'pcode', '', 1, 5);
```

```
CREATE DEFINER='haksa'@'localhost' PROCEDURE `LIST`(  
  in i_table varchar(1000),  
  in i_key varchar(50),  
  in i_word varchar(50),  
  in i_order varchar(50),  
  in i_desc varchar(5),  
  in i_page int,  
  in i_perPage int)  
BEGIN  
  SET @v_word=concat('%',i_word,'%');  
  SET @v_start=(i_page-1) * i_perPage;  
  SET @v_perPage=i_perPage;  
  SET @v_sql=concat('select * from ',i_table,' where ',i_key,' like ? order by ',i_order,' ',i_desc,' limit ?, ?');  
  PREPARE ps from @v_sql;  
  EXECUTE ps using @v_word, @v_start, @v_perPage;  
  SET @v_sql=concat('select count(*) total from ',i_table,' where ',i_key,' like ?');  
  PREPARE ps from @v_sql;  
  EXECUTE ps using @v_word;  
END
```

- 데이터베이스 접속

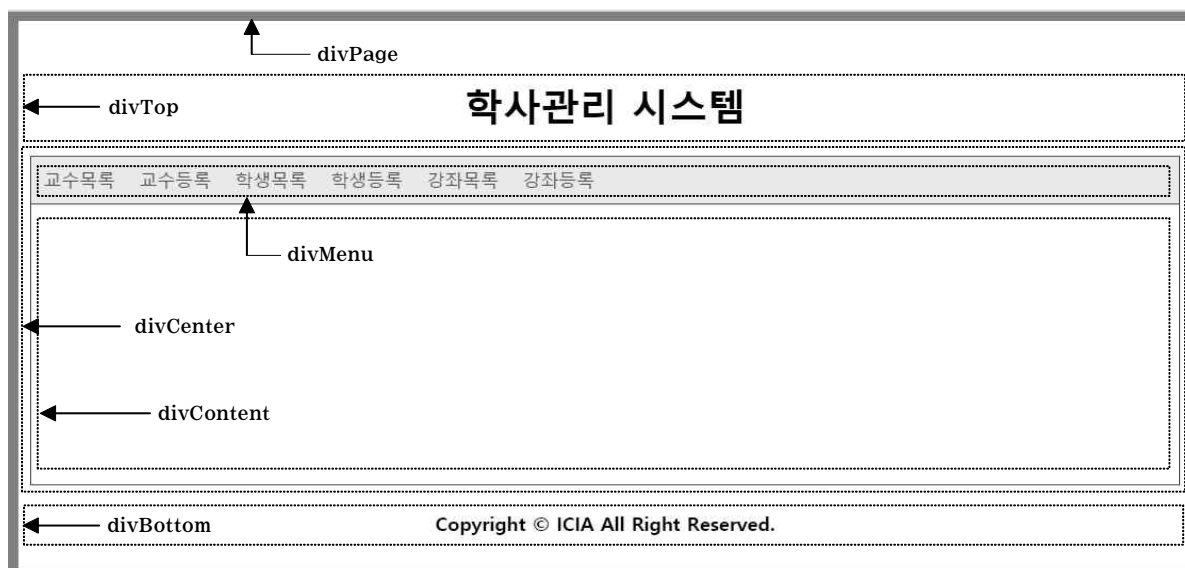
```
[Java Resources]-[src]-[database]-[Database.java]
```

```
public class Database {  
  public static Connection CON;  
  static {  
    try {  
      Class.forName("com.mysql.jdbc.Driver");  
      CON=DriverManager.getConnection("jdbc:mysql://localhost:3306/haksadb", "haksa", "pass");  
      System.out.println("접속성공");  
    } catch (Exception e) { System.out.println("접속실패:" + e.toString()); }  
  }  
}
```

```
[Java Resources]-[src]-[database]-[SqlVO.java]
```

```
public class SqlVO {  
  private String key;  
  private String word;  
  private String order;  
  private String desc;  
  private int page;  
  private int perPage;  
  //... setter()/getter()/toString()  
}
```

- 학사관리 템플릿



- 학사관리 StyleSheet

[webContent]-[home.css]

```
@charset "UTF-8";
body{background:gray;}
#divPage{width:980px; background:white; box-shadow:10px 10px 10px blak; margin:auto;}
#divTop{padding:30px 0px 0px 0px; text-align:center;}
#divCenter{border:1px solid green; margin:0px 10px 0px 10px; padding-bottom:20px;}
#divMenu{overflow:hidden; background:#EAEAEA; margin:0px 0px 20px 0px; padding:10px; border-bottom:1px solid green;}
#divMenu .menuItem{width:80px;float:left;}
#divContent{width:980px;}
#divBottom{padding:0px 0px 10px 0px; text-align:center;}
#divHeader{width:980px; text-align:center;}
#divCondition{width:800px;overflow:hidden; margin:auto; margin-bottom:5px;}
#divSearch{ float:left; }
#divSort{ float:right; }
#pagination{width:980px; text-align:center;}
.row .pcode, .row .pname, .row .hiredate, .row .dept, .row .ptitle{text-align:center;}
.row .scode, .row .sname, .row .year, .row .birthday, .row .edate{text-align:center;}
.row .lcode, .row .room, .row .hours, .row .capacity, .row .persons{text-align:center;}
.row .salary, .row .grade{text-align:right;}
a{text-decoration:none; color:green;font-size:13px;}
a:hover{font-weight:bold;}
select, input, button {height:23px; font-size:12px;}
input[type="radio"]{height:12px;}
table {border-collapse:collapse; margin:auto; margin-bottom:10px;}
table .title{background:#EAEAEA; text-align:center;}
table .row:hover{background:gray; color:white;}
td{border:1px solid black; height:30px; font-size:12px;padding:0px 5px 0px 5px;}
td button{text-align:center;}
input[type="radio"]{height:12px;}
input[name="yy"], input[name="mm"], input[name="dd"], #yy,#mm,#dd {text-align:center;}
input[name="salary"], input[name="capacity"], input[name="persons"], input[name="hours"]{text-align:right;}
```

[webContent]-[header.jsp]

```
<h1>학사관리 시스템</h1>
```

[webContent]-[menu.jsp]

```
<div class="menuItem"><a href="/haksa/professors/list.jsp">교수목록</a></div>
<div class="menuItem"><a href="/haksa/professors/insert.jsp">교수등록</a></div>
<div class="menuItem"><a href="/haksa/students/list.jsp">학생목록</a></div>
<div class="menuItem"><a href="/haksa/students/insert.jsp">학생등록</a></div>
<div class="menuItem"><a href="/haksa/courses/list.jsp">강좌목록</a></div>
<div class="menuItem"><a href="/haksa/courses/insert.jsp">강좌등록</a></div>
```

[webContent]-[footer.jsp]

```
<h4>Copyright © ICIA All Rights Reserved.</h4>
```

[webContent]-[index.jsp]

```
<html>
<head>
  <title>학사관리</title>
  <link rel="stylesheet" href="home.css">
</head>
<body>
  <div id="divPage">
    <div id="divTop"><jsp:include page="header.jsp"/></div>
    <div id="divCenter">
      <div id="divMenu"><jsp:include page="menu.jsp"/></div>
      <div id="divContent">
        <!-- 여기에 내용 등록 시작----->

        <!-- 여기에 내용 등록 종료----->
      </div>
    </div>
    <div id="divBottom"><jsp:include page="footer.jsp"/></div>
  </div>
</body>
</html>
```


- 교수 목록 프로그램

교수 목록

교수번호 ▼		3행 ▼	검색	검색수: 4건	교수번호 ▼	오름차순 ▼
교수번호	교수이름	교수학과	교수직급	임용일자	급여	교수정보
221	이병렬	전산	정교수	1975-04-03	3,000,000	교수정보
228	이재광	전산	부교수	1991-09-19	2,500,000	교수정보
311	강승일	전자	부교수	1994-06-09	2,300,000	교수정보

◀ ▶ [1/2]

[Java Resources]-[src]-[professors]-[ProfessorsVO.java]

```
public class ProfessorsVO {
    private String pcode;
    private String pname;
    private String dept;
    private Date hireDate;
    private String title;
    private int salary;
    // ... setter() getter() toString();
}
```

[Java Resources]-[src]-[professors]-[ProfessorsDAO.java]

```
public class ProfessorsDAO {
    public JSONObject list(SqlVO sqlVO) {
        JSONObject jobject=new JSONObject();
        try {
            String sql="call list(?,?,?,?);";
            CallableStatement cs=Database.CON.prepareCall(sql);
            cs.setString(1, "professors");
            cs.setString(2, sqlVO.getKey());
            cs.setString(3, sqlVO.getWord());
            cs.setString(4, sqlVO.getOrder());
            cs.setString(5, sqlVO.getDesc());
            cs.setInt(6, sqlVO.getPage());
            cs.setInt(7, sqlVO.getPerPage());
            cs.execute();

            ResultSet rs=cs.getResultSet();
            JSONArray jArray=new JSONArray();
            DecimalFormat df=new DecimalFormat("#,###");
            SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd");
            while(rs.next()) {
                JSONObject obj=new JSONObject();
                obj.put("pcode", rs.getString("pcode"));
                obj.put("pname", rs.getString("pname"));
                obj.put("dept", rs.getString("dept"));
                obj.put("title", rs.getString("title"));
                try {
                    obj.put("salary", df.format(rs.getInt("salary")));
                    obj.put("hiredate", sdf.format(rs.getDate("hiredate")));
                }catch(Exception e) {}
                jArray.add(obj);
            }
            jobject.put("array", jArray);
            cs.getMoreResults();
            rs=cs.getResultSet();
            int count=0;
            if(rs.next()) { count=rs.getInt("total"); }
            int perPage = sqlVO.getPerPage();
            int totPage = count%perPage==0?count/perPage:count/perPage+1;
            jobject.put("count", count);
            jobject.put("page", sqlVO.getPage());
            jobject.put("perPage", sqlVO.getPerPage());
            jobject.put("totPage", totPage);
        }catch(Exception e) { System.out.println("교수목록:" + e.toString()); }
        return jobject;
    }
}
```

```

@WebServlet(value= {"/professors/list"})
public class ProfessorsServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();

        ProfessorsDAO pdao=new ProfessorsDAO();
        switch(request.getServletPath()) {
            case "/professors/list":
                SqlVO sqlVO=new SqlVO();
                String key=request.getParameter("key")==null?"pcode":request.getParameter("key");
                String word=request.getParameter("word")==null?"":request.getParameter("word");
                String order=request.getParameter("order")==null?"pcode":request.getParameter("order");
                String desc=request.getParameter("key")==null?"":request.getParameter("desc");
                String page=request.getParameter("page")==null?"1":request.getParameter("page");
                String perPage=request.getParameter("perPage")==null?"5":request.getParameter("perPage");

                sqlVO.setKey(key);
                sqlVO.setWord(word);
                sqlVO.setOrder(order);
                sqlVO.setDesc(desc);
                sqlVO.setPage(Integer.parseInt(page));
                sqlVO.setPerPage(Integer.parseInt(perPage));
                out.println(pdao.list(sqlVO));
                break;
        }
    }
}

```

```

<html>
<head>
    <title>학사관리</title>
    <link rel="stylesheet" href="../home.css">
    <script src="http://code.jquery.com/jquery-3.1.1.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/handlebars.js/3.0.1/handlebars.js"></script>
</head>
<body>
    ...
    <div id="divHeader"><h2>교 수 목 록</h2></div>
    <div id="divCondition">
        <div id="divSearch">
            <select id="selKey">
                <option value="pcode">번호</option>
                <option value="pname">이름</option>
                <option value="dept">학과</option>
            </select>
            <input type="text" id="txtWord">
            <select id="selPerPage">
                <option value="3">3행</option>
                <option value="5">5행</option>
                <option value="10">10행</option>
            </select>
            <input type="button" id="btnSearch" value="검색">
            <span style="font-size:12px;">검색수: <b id="count"></b>건</span>
        </div>
        <div id="divSort">
            <select id="selOrder">
                <option value="pcode">번호</option>
                <option value="pname">이름</option>
                <option value="dept">학과</option>
            </select>
            <select id="selDesc">
                <option value="">오름차순</option>
                <option value="desc">내림차순</option>
            </select>
        </div>
    </div>

```

```

<table id="tbl"></table>
<script id="temp" type="text/x-handlebars-template">
  <tr class="title">
    <td width=130>교수번호</td>
    <td width=130>교수이름</td>
    <td width=130>교수학과</td>
    <td width=130>교수직급</td>
    <td width=130>임용일자</td>
    <td width=150>급여</td>
  </tr>
  {{#each array}}
  <tr class="row">
    <td class="pcode">{{pcode}}</td>
    <td class="pname">{{pname}}</td>
    <td class="dept">{{dept}}</td>
    <td class="ptitle">{{title}}</td>
    <td class="hiredate">{{hiredate}}</td>
    <td class="salary">{{salary}}</td>
  </tr>
  {{/each}}
</script>
<div id="pagination">
  <button id="btnPre">◀</button> <button id="btnNext">▶</button>
  [<span id="curPage"></span></span id="totPage"></span>]
</div>
</body>
<script>
  var key, word, order, desc, perPage, page=1;
  getList();
  $("#btnSearch").on("click", function(){ page=1; getList(); });
  $("#selOrder, #selDesc, #selPerPage").change(function(){ page=1; getList(); });
  $("#txtWord").keydown(function(key){
    if(key.keyCode==13){ page=1; getList(); }
  });
  $("#btnNext").click(function(){ page++; getList(); });
  $("#btnPre").click(function(){ page--; getList(); });

  //교수목록 출력을 위한 함수
  function getList(){
    key=$("#selKey").val();
    word=$("#txtWord").val();
    order=$("#selOrder").val();
    desc=$("#selDesc").val();
    perPage=$("#selPerPage").val();
    $.ajax({
      type:"get",
      url:"list",
      data:{'key':key,'word':word,'page':page,'perPage':perPage,'order':order,'desc':desc},
      dataType:"json",
      success:function(data){
        var template = Handlebars.compile($("#temp").html());
        $("#tbl").html(template(data));

        if(data.page==1){
          $("#btnPre").attr("disabled",true);
        }else{ $("#btnPre").attr("disabled",false); }

        if(data.page==data.totPage){
          $("#btnNext").attr("disabled",true);
        }else{ $("#btnNext").attr("disabled",false);}

        $("#curPage").html(data.page);
        $("#totPage").html(data.totPage);
        $("#count").html(data.count);
      }
    });
  }
</script>
</html>

```

- 학생 목록 프로그램

학 생 목 록

학생번호 ▼

3행 ▼

검색

검색수: 13건

학생번호 ▼

오름차순 ▼

학생번호	학생이름	학생학과	학년	생년월일	지도교수	학생정보
92414029	서연우	전산	3	1973-10-06	이재광 228	학생정보
92414033	김창덕	전산	4	1991-02-03	이병렬 221	학생정보
92414034	심청이	전산	1	1991-02-03	강승일 311	학생정보

◀ ▶

[1/5]

[Java Resources]-[src]-[students]-[StudentsVO.java]

```
public class StudentsVO extends ProfessorsVO{
    private String scode;
    private String sname;
    private String dept;
    private String year;
    private Date birthday;
    private String advisor;
    // ... setter()/ getter()/ toString();
}
```

[Java Resources]-[src]-[students]-[StudentsDAO.java]

```
public class StudentsDAO {
    //학생목록
    public JSONObject list(SqlVO sqlVO) {
        JSONObject jObject=new JSONObject();
        try {
            String sql="call list(?,?,?,?,?,?)";
            CallableStatement cs=Database.CON.prepareCall(sql);
            cs.setString(1, "(select s.*, pname from students s, left join professors on advisor=pcode) tbl");
            cs.setString(2, sqlVO.getKey());
            cs.setString(3, sqlVO.getWord());
            cs.setString(4, sqlVO.getOrder());
            cs.setString(5, sqlVO.getDesc());
            cs.setInt(6, sqlVO.getPage());
            cs.setInt(7, sqlVO.getPerPage());
            cs.execute();

            ResultSet rs=cs.getResultSet();
            JSONArray jArray=new JSONArray();
            while(rs.next()) {
                JSONObject obj=new JSONObject();
                ...
                jArray.add(obj);
            }
            jObject.put("array", jArray);
            cs.getMoreResults();
            rs=cs.getResultSet();

            int count=0;
            if(rs.next()) { count=rs.getInt("total"); }
            int perPage = sqlVO.getPerPage();
            int totPage = count%sqlVO.getPerPage()==0?count/sqlVO.getPerPage():count/sqlVO.getPerPage()+1;

            jObject.put("count", count);
            jObject.put("page", sqlVO.getPage());
            jObject.put("perPge", sqlVO.getPerPage());
            jObject.put("totPage", totPage);
        }catch(Exception e) { System.out.println("학생목록:" + e.toString()); }
        return jObject;
    }
}
```

[Java Resources]-[src]-[students]-[StudentsServlet.java]

```
@WebServlet(value= {"/students/list"})
public class StudentsServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        StudentsDAO sdao=new StudentsDAO();

        switch(request.getServletPath()) {
            case "/students/list":
                SqlVO sqlVO=new SqlVO();
                ...
                out.println(sdao.list(sqlVO));
                break;
        }
    }
}
```

[webContent]-[students]-[list.jsp]

```
<script src="../../home.js"></script>
<script>
    $("#tbl").on("click", ".btnRead", function(){
        var row=$(this).parent().parent();
        var scode=row.find(".scode").html();
        location.href="read?scode=" + scode;
    });
</script>
```

[webContent]-[home.jsp]

```
var key, word, order, desc, perPage, page=1;
getList();

$("#btnSearch").on("click", function(){ page=1; getList(); });
$("#selOrder, #selDesc, #selPerPage").change(function(){ page=1; getList(); });
$("#txtWord").keydown(function(key){
    if(key.keyCode==13){ page=1; getList(); }
});
$("#btnNext").click(function(){ page++; getList(); });
$("#btnPre").click(function(){ page--; getList(); });

function getList(){
    key=$("#selKey").val();
    word=$("#txtWord").val();
    order=$("#selOrder").val();
    desc=$("#selDesc").val();
    perPage=$("#selPerPage").val();

    //목록 출력을 위한 함수
    $.ajax({
        type:"get",
        url:"list",
        data:{key:key, word:word, "page":page, "perPage":perPage, "order":order, "desc":desc},
        dataType:"json",
        success:function(data){
            var template = Handlebars.compile($("#temp").html());
            $("#tbl").html(template(data));

            if(data.page==1){ $("#btnPre").attr("disabled",true);
            }else{ $("#btnPre").attr("disabled",false); }

            if(data.page==data.totPage){
                $("#btnNext").attr("disabled",true);
            }else{ $("#btnNext").attr("disabled",false); }

            $("#curPage").html(data.page);
            $("#totPage").html(data.totPage);
            $("#count").html(data.count);
        }
    });
}
```

- 강좌 목록 프로그램

강좌 목록

강좌번호 ▼

3행 ▼

검색

검색수: 7건

강좌번호 ▼

오름차순 ▼

강좌번호	강좌이름	강의시간	강의실	담당교수	최대인원	수강인원	강좌정보
C312	자료구조	2	510	이재광	100	60	강좌정보
C401	데이터베이스	3	414	이병렬	80	80	강좌정보
C421	알고리즘	3	510	이재광	80	72	강좌정보

◀ ▶

[2/3]

[Java Resources]-[src]-[courses]-[CoursesVO.java]

```
public class CoursesVO extends ProfessorsVO{
    private String lcode;
    private String lname;
    private int hours;
    private String room;
    private String instructor;
    private int capacity;
    private int persons
    // ... setter()/ getter()/ toString();
}
```

[Java Resources]-[src]-[courses]-[CoursesDAO.java]

```
public class CoursesDAO {
    public JSONObject list(SqlVO sqlVO) {
        JSONObject jObject=new JSONObject();
        try {
            String sql="call list(?,?,?,?.?.?.?)";
            CallableStatement cs=Database.CON.prepareCall(sql);
            cs.setString(1, "(select c.*, pname from courses c left join professors on instructor=pcode) tbl");
            ...
        } catch (Exception e) { System.out.println("강좌목록:" + e.toString()); }
        return jObject;
    }
}
```

[Java Resources]-[src]-[courses]-[CoursesServlet.java]

```
@WebServlet(value = {"/courses/list"})
public class StudentsServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        CoursesDAO cdao=new CoursesDAO();
        switch(request.getServletPath()) {
            case "/courses/list":
                SqlVO sqlVO=new SqlVO();
                ...
                out.println(cdao.list(sqlVO));
                break;
        }
    }
}
```

[webContent]-[courses]-[list.jsp]

```
<script>
    $("#tbl").on("click", ".btnRead", function(){
        var row=$(this).parent().parent();
        var lcode=row.find(".lcode").html();
        location.href="read?lcode=" + lcode;
    });
</script>
<script src="../home.js"></script>
</html>
```

- 학생 등록 프로그램

학 생 등 록

학생번호	<input type="text" value="92414030"/>	학생이름	<input type="text" value="학생이름"/>
학생학과	<input type="text" value="전산 ▼"/>	생년월일	<input type="text" value="yyyy"/> 년 <input type="text" value="mm"/> 월 <input type="text" value="dd"/> 일
학생학년	<input checked="" type="radio"/> 1학년 <input type="radio"/> 2학년 <input type="radio"/> 3학년 <input type="radio"/> 4학년	지도교수	<input type="text" value="교수번호"/> <input type="text" value="교수명"/>

[Java Resources]-[src]-[students]-[StudentsDAO.java]

```
public void insert(StudentsVO vo) {
    try {
        String sql="insert into students(scode,sname,dept,year,birthday,advisor) values(?,?,?,?,?,?)";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, vo.getScode());
        ps.setString(2, vo.getSname());
        ps.setString(3, vo.getDept());
        ps.setString(4, vo.getYear());
        ps.setDate(5, vo.getBirthday());
        ps.setString(6, vo.getAdvisor());
        ps.execute();
    } catch (Exception e) {
        System.out.println("학생등록:" + e.toString());
    }
}

public StudentsVO read(String scode) {
    StudentsVO vo=new StudentsVO();
    try {
        String sql="select s.*, pname from students s left join professors on advisor=pcode where scode=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, scode);
        ResultSet rs=ps.executeQuery();
        if(rs.next()) {
            vo.setScode(rs.getString("scode"));
            vo.setSname(rs.getString("sname"));
            vo.setDept(rs.getString("dept"));
            vo.setYear(rs.getString("year"));
            vo.setAdvisor(rs.getString("advisor"));
            vo.setPname(rs.getString("pname"));
            vo.setBirthday(rs.getDate("birthday"));
        }
    } catch (Exception e) {
        System.out.println("학생읽기:" + e.toString());
    }
    return vo;
}

public int delete(String scode) {
    int count = -1;
    try {
        String sql="call del_students(?,?)";
        CallableStatement cs=Database.CON.prepareCall(sql);
        cs.setString(1, scode);
        cs.registerOutParameter(2, java.sql.Types.INTEGER);
        cs.execute();
        count = cs.getInt(2);
    } catch (Exception e) {
        System.out.println("학생삭제:" + e.toString());
    }
    return count;
}
```

```
exec del_students('92414029', @count); select @count;
```

```
CREATE PROCEDURE `DEL_STUDENTS`(in i_scode char(8), out o_count int)
BEGIN
    select count(*) into o_count from enrollments where scode = i_scode;
    IF(o_count = 0)THEN
        delete from students where scode=i_scode;
    END IF;
END
```

[Java Resources]-[src]-[students]-[StudentsServlet.java]

```
@WebServlet(value= ("/students/list","/students/insert"))
public class StudentsServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException{
        request.setCharacterEncoding("UTF-8");
        PrintWriter out=response.getWriter();
        JSONObject jObject=new JSONObject();
        StudentsDAO sdao= new StudentsDAO();

        StudentsVO vo=new StudentsVO();
        vo.setScode(request.getParameter("scode"));
        vo.setSname(request.getParameter("sname"));
        vo.setDept(request.getParameter("dept"));
        vo.setYear(request.getParameter("year"));
        vo.setAdvisor(request.getParameter("pcode"));
        try {
            java.sql.Date birthday = Date.valueOf(request.getParameter("birthday"));
            vo.setBirthday(birthday);
        } catch (Exception e) { }

        switch(request.getServletPath()){
        case "/students/insert":
            StudentsVO v=sdao.read(vo.getScode());
            if(v.getScode()==null) { sdao.insert(vo); jObject.put("count", 0);
            }else { jObject.put("count", 1); }
            out.println(jObject);
            break;
        }
    }
}
```

```
[wecontent--[students]-[insert.jsp]
```

[illegible]

[wecontent--[students]-[insert.jsp]

```
<script>
    $("#btnInsert").on("click", function(){
        if(!confirm("학생을 등록하시겠습니까?")) return;
        var scode=$(frm.scode).val();
        var sname=$(frm.sname).val();
        var dept=$(frm.dept).val();
        var year=$(input[name="year"]:checked).val();
        var birthday=$(frm.yy).val() + "-" + $(frm.mm).val() + "-" + $(frm.dd).val();
        var advisor=$(frm.pcode).val();

        var fmtDate = /^(19|20)\d{2}-(0[1-9]|1[012])-(0[1-9]|12)[0-9]|3[0-1])$/;
        if(scode==""||sname==""||dept==""||advisor==""){
            alert("모든 항목들을 입력해주세요."); return;
        }else if(!fmtDate.test(birthday)){
            alert("날짜는 yyyy-mm-dd 형식으로 입력해주세요."); return;
        }
        $.ajax({
            type:"post",
            url:"insert",
            data:{"scode":scode,"sname":sname,"dept":dept,"year":year,"birthday":birthday,"advisor":advisor},
            dataType:"json",
            success:function(data){
                if(data.count==1){ alert("학번이 이미 존재합니다.");
                }else{ alert("등록되었습니다."); location.href="/haksa/students/list.jsp"; }
            }
        });
    });
    $(frm.pcode).click(function(){
        window.open('/haksa/professors/listAll.jsp','advisor','width=320,height=300,top=200,left=900');
    });
</script>
```

[webContent]-[professors]-[listAll.jsp]

```
<style>
    td{border:1px solid black; height:30px; font-size:12px;padding:0px 5px 0px 5px;}
    table .title{background:#EAEAEA; text-align:center;}table .row:hover{background:gray; color:white;}
    table {border-collapse:collapse;}.row:hover{background:gray; color:white; cursor:pointer;}
    #pagination, h2{text-align:center;padding-top:5px;}
</style>
</head>
<body>
    <h2>교 수 목 록 </h2>
    <table id="tbl" width=300></table>
    <script id="temp" type="text/x-handlebars-template">
        <tr class="title">
            <td width=100>교수번호</td>
            <td width=100>교수이름</td>
            <td width=100>교수학과</td>
        </tr>
        {{#each array}}
        <tr class="row">
            <td class="pcode">{{pcode}}</td>
            <td class="pname">{{pname}}</td>
            <td class="dept">{{dept}}</td>
        </tr>
        {{/each}}
    </script>
    <div id="pagination"><button id="btnPre">◀</button> <button id="btnNext">▶</button></div>
</body>
<script src="../home.js"></script>
<script>
    $("#tbl").on("click", ".row", function(){
        var pcode=$(this).find(".pcode").html();
        var pname=$(this).find(".pname").html();
        $(opener.frm.pcode).val(pcode);
        $(opener.frm.pname).val(pname);
        window.close();
    });
</script>
```

교수 목록

교수번호	교수이름	교수학과
221	이병렬	전산
228	이재광	전산
311	강승일	전자
509	오문환	건축



- 학생 정보 읽기, 수정, 삭제 프로그램

학 생 정 보

학생번호	<input type="text" value="92414029"/>	학생학과	<input type="text" value="전산 ▼"/>
학생이름	<input type="text" value="서연우"/>	생년월일	<input type="text" value="1973"/> 년 <input type="text" value="10"/> 월 <input type="text" value="06"/> 일
학생학년	<input type="radio"/> 1학년 <input type="radio"/> 2학년 <input checked="" type="radio"/> 3학년 <input type="radio"/> 4학년	지도교수	<input type="text" value="228"/> <input type="text" value="이재광"/>

[Java Resources]-[src]-[students]-[StudentsDAO.java]

```
public void update(StudentsVO vo) {
    try {
        String sql="update students set sname=?,dept=?,year=?,birthday=?,advisor=? where scode=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, vo.getSname());
        ps.setString(2, vo.getDept());
        ps.setString(3, vo.getYear());
        ps.setDate(4, vo.getBirthday());
        ps.setString(5, vo.getAdvisor());
        ps.setString(6, vo.getScode());
        ps.execute();
    }catch(Exception e) { System.out.println("학생수정:" + e.toString()); }
}

public int delete(String scode) {
    int count = -1;
    try {
        String sql="call del_students(?,?)";
        CallableStatement cs=Database.CON.prepareCall(sql);
        cs.setString(1, scode);
        cs.registerOutParameter(2, java.sql.Types.INTEGER);
        cs.execute();
        count = cs.getInt(2);
    }catch(Exception e) { System.out.println("학생삭제:" + e.toString()); }
    return count;
}
```

[Java Resources]-[src]-[students]-[StudentsServlet.java]

```
@WebServlet(value= {"/students/list","/students/insert","/students/read","/students/update","/students/delete"})
public class StudentsServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        StudentsDAO sdao=new StudentsDAO();

        switch(request.getServletPath()) {
            case "/students/read":
                request.setAttribute("vo", sdao.read(request.getParameter("scode")));
                RequestDispatcher dis=request.getRequestDispatcher("read.jsp");
                dis.forward(request, response);
                break;
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        ...
        switch(request.getServletPath()) {
            case "/students/update":
                sdao.update(vo);
                response.sendRedirect("list.jsp");
                break;
            case "/students/delete":
                jObject.put("count", sdao.delete(vo.getScode()));
                out.println(jObject);
                break;
        }
    }
}
```

```

<%@taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions"%>
<div id="divHeader"><h2>학 생 정 보</h2></div>
<form name="frm" action="update" method="post">
  <table width=800>
    ...
    <tr>
      <td width=100 class="title">학생학과</td>
      <td width=300>
        <select name="dept"><option <c:out value="\${vo.dept=='전산'? 'selected': ''}"/>>전산</option>...</select>
      </td>
      <td width=100 class="title" width=150>생년월일</td>
      <td width=300>
        <input type="hidden" name="birthday">
        <input type="text" size=1 id="yy" placeholder="yyyy" maxlength=4 value="\${fn:substring(vo.birthday,0,4)}">년
        <input type="text" size=1 id="mm" placeholder="mm" maxlength=2 value="\${fn:substring(vo.birthday,5,7)}">월
        <input type="text" size=1 id="dd" placeholder="dd" maxlength=2 value="\${fn:substring(vo.birthday,8,10)}">일
      </td>
    </tr>
    <tr>
      <td class="title">학생학년</td>
      <td><input type="radio" name="year" value=1 <c:out value="\${vo.year=='1'? 'checked': ''}"/>>1학년 ...</td>
      <td class="title">지도교수</td>
      <td>
        <input type="text" name="pcode" size=5 readonly value="\${vo.advisor}"/>
        <input type="text" name="pname" size=5 readonly value="\${vo.pname}"/>
      </td>
    </tr>
  </table>
  <div id="pagination">
    <input type="submit" value="저장" id="btnInsert">
    <input type="button" value="삭제" id="btnDelete"><input type="reset" value="취소">
  </div>
</form>
<script>
  $(frm).submit(function(e){
    e.preventDefault();
    if(!confirm("학생 정보를 수정하시겠습니까?")) return;
    var scode=$(frm.scode).val();
    var sname=$(frm.sname).val();
    var dept=$(frm.dept).val();
    var year=$(frm.year).val();
    var birthday=$(frm.yy).val() + "-" + $(frm.mm).val() + "-" + $(frm.dd).val();
    $(frm.birthday).val(birthday);
    var advisor=$(frm.pcode).val();
    var fmtDate = /^(19|20)\d{2}-(0[1-9]|1[012])-(0[1-9]|1[12])[0-9]|3[0-1])$/;

    if(scode=="||sname=="||dept=="||advisor==" ){ alert("모든 항목들을 입력해주세요."); return;
    }else if(!fmtDate.test(birthday)){ alert("날짜는 yyyy-mm-dd 형식으로 입력해주세요."); return; }
    frm.submit();
  });
  //삭제 버튼을 클릭한 경우
  $("#btnDelete").on("click", function(){
    var scode=$(frm.scode).val();
    if(!confirm(scode + " 학생을 삭제하시겠습니까?")) return;
    $.ajax({
      type:"post",
      url:"delete",
      data:{"scode":scode},
      dataType:"json",
      success:function(data){
        if(data.count==0){
          alert("삭제되었습니다.");
          location.href="list.jsp";
        }else(data.count > 0){
          alert("수강신청 데이터가 있어 삭제할 수 없습니다.");
        }
      }
    });
  });
</script>

```

- 수강 신청, 취소 프로그램

수강신청

수강신청 과목:	C301 : 파일처리론 : 이병렬 ▼	수강신청
----------	----------------------	------

강좌번호	강좌이름	수강신청일	담당교수	강의실	강의시수	수강취소
C301	파일처리론	2018-12-02	이병렬	506	3	수강취소
C401	데이터베이스	2020-05-26	이병렬	414	3	수강취소
C312	자료구조	2020-05-26	이재광	510	2	수강취소

[Java Resources]-[src]-[courses]-[CoursesDAO.java]

```
public ArrayList<CoursesVO> listAll(){
    ArrayList<CoursesVO> list= new ArrayList<CoursesVO>();
    try {
        String sql="select c.*, pname from courses c left join professors on instructor=pcode";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ResultSet rs=ps.executeQuery();
        while(rs.next()) {
            CoursesVO vo=new CoursesVO();
            vo.setLcode(rs.getString("lcode"));
            vo.setLname(rs.getString("lname"));
            vo.setRoom(rs.getString("room"));
            vo.setPname(rs.getString("pname"));
            list.add(vo);
        }
    } catch (Exception e) { System.out.println("모든 강좌데이터:" + e.toString()); }
    return list;
}
```

[Java Resources]-[src]-[enrollments]-[EnrollmentsDAO.java]

```
public class EnrollmentsDAO {
    public JSONObject enrollCourses(String scode) {
        JSONObject jObject=new JSONObject();
        try {
            String sql="select e.*, pname from enrollCourses e left join professors on instructor=pcode where e.scode=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, scode);
            ResultSet rs=ps.executeQuery();

            JSONArray jArray=new JSONArray();
            while(rs.next()) {
                JSONObject obj=new JSONObject();
                obj.put("lcode", rs.getString("lcode"));
                obj.put("lname", rs.getString("lname"));
                obj.put("grade", rs.getInt("grade"));
                String edate="";
                try {
                    SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd");
                    edate=sdf.format(rs.getDate("edate"));
                } catch (Exception e) {}
                obj.put("edate", rs.getString("edate"));
                obj.put("pname", rs.getString("pname"));
                obj.put("room", rs.getString("room"));
                obj.put("hours", rs.getInt("hours"));
                jArray.add(obj);
            }
            jObject.put("array", jArray);
        } catch (Exception e) {
            System.out.println("특정학생이 신청한 강좌목록:" + e.toString());
        }
        return jObject;
    }
}
```

```
create view enrollCourses as
(select e.*, lname, room, hours, instructor from enrollments e, courses c
where e.lcode=c.lcode);
```

[Java Resources]-[src]-[enrollments]-[EnrollmentsServlet.java]

```
@WebServlet(value= {"/enroll/courses"})
public class EnrollmentsServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        EnrollmentsDAO edao=new EnrollmentsDAO();

        switch(request.getServletPath()) {
        case "/enroll/courses":
            JSONObject jObject=edao.enrollCourses(request.getParameter("scode"));
            out.println(jObject);
            break;
        }
    }
}
```

[Java Resources]-[src]-[enrollments]-[EnrollmentsDAO.java]

```
public int insert(String scode, String lcode) {
    int count=-1;
    try {
        String sql="call add_enroll(?,?,?)";
        CallableStatement cs=Database.CON.prepareCall(sql);
        cs.setString(1, scode);
        cs.setString(2, lcode);
        cs.registerOutParameter(3, java.sql.Types.INTEGER);
        cs.execute();
        count = cs.getInt(3);
    }catch(Exception e) {
        System.out.println("수강신청:" + e.toString());
    }
    return count;
}

public void delete(String lcode, String scode) {
    try {
        String sql="delete from enrollments where scode=? and lcode=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, scode);
        ps.setString(2, lcode);
        ps.execute();
    }catch(Exception e) {
        System.out.println("수강취소:" + e.toString());
    }
}
```

```
CREATE DEFINER='haksa'@'localhost' PROCEDURE `ADD_ENROLL`(
    in i_scode char(8),
    in i_lcode char(4),
    out o_count int
)
BEGIN
    select count(*) into o_count from enrollments
    where scode=i_scode and lcode=i_lcode;
    IF(o_count=0)THEN
        insert into enrollments(scode,lcode,edate,grade)
        values(i_socde, i_lcode, now(), 0);
    END IF;
END

//결과 확인
call add_enroll('92414029', 'C301', @cnt)
select @cnt
```

[Java Resources]-[src]-[enrollments]-[EnrollmentsServlet.java]

```
@WebServlet(value= {"/enroll/students", "/enroll/insert", "/enroll/delete"})
public class EnrollmentsServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        PrintWriter out=response.getWriter();
        JSONObject jObject = new JSONObject();

        EnrollmentsDAO edao=new EnrollmentsDAO();
        String scode=request.getParameter("scode");
        String lcode=request.getParameter("lcode");

        switch(request.getServletPath()) {
        case "/enroll/insert":
            jObject.put("count", edao.insert(scode, lcode));
            out.println(jObject);
            break;
        case "/enroll/delete":
            edao.delete(lcode, scode);
            break;
        }
    }
}
```

```

<div id="divCondition">
  <table>
    <tr height=40>
      <td class="title" width=120>수강신청 과목:</td>
      <td>
        <select id="selLcode">
          <c:forEach items="${clist}" var="cvo">
            <option value="${cvo.lcode}" lname="${vo.lname}">${cvo.lcode} : ${cvo.lname} : ${cvo.pname}</option>
          </c:forEach>
        </select>
        <input type="button" value="수강신청" id="btnEnroll">
      </td>
    </tr>
  </table>
<table id="tbl" width=800></table>
<script id="temp" type="text/x-handlebars-template">
  <tr class="title">
    <td width=100>강좌번호</td><td width=230>강좌이름</td><td width=100>수강신청일</td>
    <td width=100>담당교수</td><td width=100>강의실</td><td width=100>강의시수</td><td width=70>수강취소</td>
  </tr>
  {{#each array}}
  <tr class="row">
    <td class="lcode">{{lcode}}</td><td class="name">{{lname}}</td><td class="edate">{{edate}}</td>
    <td class="pname">{{pname}}</td><td class="room">{{room}}</td><td class="hours">{{hours}}</td>
    <td><button class="btnDelete">수강취소</td>
  </tr>
  {{/each}}
</script>
</div>
<script>
  var scode="${vo.scode}";
  getList();
  //특정 학생이 수강신청한 강좌목록
  function getList(){
    $.ajax({
      type:"get",
      url:"/haksa/enroll/courses",
      data:{"scode":scode},
      dataType:"json",
      success:function(data){
        var template = Handlebars.compile($("#temp").html());
        $("#tbl").html(template(data));
      }
    });
  }
  //수강신청 버튼을 클릭한 경우
  $("#btnEnroll").on("click", function(){
    var lcode=$("#selLcode").val();
    var lname=$("#lcode option:selected").attr("lname");
    if(!confirm(lcode + " 강좌를 수강신청하시겠습니까?")) return;
    $.ajax({
      type:"post",
      url:"/haksa/enroll/insert",
      data:{"lcode":lcode, "scode":scode},
      dataType:"json",
      success:function(data){
        if(data.count==0){ alert( "수강신청이 완료되었습니다."); getList(); }
        else if(data.count==1){ alert("이미 수강신청한 강좌입니다."); }
      }
    });
  });
  //각행의 수강신청 취소 버튼을 클릭한 경우
  $("#tbl").on("click",".row button", function(){
    var row=$(this).parent().parent();
    var lcode=row.find(".lcode").html();
    if(!confirm(lcode + " 강좌를 수강취소 하겠습니까?")) return;
    $.ajax({
      type:"post",
      url:"/haksa/enroll/delete",
      data:{"scode":scode, "lcode":lcode},
      success:function(){ alert("수강이 취소되었습니다."); getList(); }
    });
  });
</script>

```

- 강좌 정보 읽기 프로그램

강좌 정보

강좌번호	<input type="text" value="A109"/>	강의실	<input type="text" value="101"/>	강의시수	<input type="text" value="2"/>
강좌이름	<input type="text" value="한국의건축문화"/>			최대수강인원	<input type="text" value="120"/>
담당교수	<input type="text" value="509"/>	<input type="text" value="오문환"/>		수강신청인원	<input type="text" value="36"/>

[Java Resources]-[src]-[courses]-[CoursesDAO.java]

```
public CoursesVO read(String lcode) {
    CoursesVO vo = new CoursesVO();
    try {
        String sql="select c.*, pname from courses c left join professors on instructor=pcode where lcode=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, lcode);
        ResultSet rs=ps.executeQuery();

        if(rs.next()) {
            vo.setLcode(rs.getString("lcode"));
            vo.setLname(rs.getString("lname"));
            ...
        }
    } catch (Exception e) { System.out.println("강좌정보 읽기:" + e.toString()); }
    return vo;
}
```

[Java Resources]-[src]-[courses]-[CoursesServlet.java]

```
@WebServlet(value= {"/courses/list", "/courses/read"})
public class CoursesServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        ...
        switch(request.getServletPath()) {
            case "/courses/read":
                request.setAttribute("vo", cdao.read(request.getParameter("lcode")));
                RequestDispatcher dis=request.getRequestDispatcher("read.jsp");
                dis.forward(request, response);
                break;
        }
    }
}
```

[webContent]-[courses]-[read.jsp]

```
<form name="frm" action="update" method="post">
<table>
<tr>
<td class="title">강좌번호</td><td><input type="text" size=10 name="lcode" value="{vo.lcode}" readonly></td>
<td class="title" width=150>강의실</td><td><input type="text" size=5 name="room" value="{vo.room}"></td>
<td class="title" width=150>강의시수</td><td><input type="text" size=5 name="hours" value="{vo.hours}"></td>
</tr>
<tr>
<td class="title">강좌이름</td><td colspan=3><input type="text" size=60 name="lname" value="{vo.lname}"></td>
<td class="title">최대수강인원</td><td><input type="text" size=5 name="capacity" value="{vo.capacity}"></td>
</tr>
<tr>
<td width=90 class="title">담당교수</td>
<td width=400 colspan=3>
<input type="text" size=5 name="pcode" value="{vo.instructor}">
<input type="text" size=10 name="pname" value="{vo.pname}">
</td>
<td class="title">수강신청인원</td><td><input type="text" size=5 name="persons" value="{vo.persons}"></td>
</tr>
</table>
<div id="pagination">
<input type="submit" value="저장" id="btnInsert">
<input type="button" value="삭제" id="btnDelete">
<input type="reset" value="취소">
</div>
</form>
```

- 수강 신청, 강좌 점수 입력 프로그램

성 적 입 력

선택저장

<input type="checkbox"/>	학생번호	학생이름	학과	지도교수	학년	수강신청일	점수
<input type="checkbox"/>	96414404	김수정	전산	이재광	3	2018-12-02	90 수정
<input type="checkbox"/>	92514009	이지행	전자	강승일	4	2018-12-02	60 수정
<input type="checkbox"/>	92514023	김형명	전자	강승일	4	2018-07-21	0 수정
<input type="checkbox"/>	92454018	이원구	건축	오문환	3	2018-07-21	0 수정

[Java Resources]-[src]-[enrollments]-[EnrollmentsDAO.java]

//특정강좌를 신청한 학생목록

```
public JSONObject enrollStudents(String lcode) {
    JSONObject jobject=new JSONObject();
    try {
        String sql="select e.*, pname from enrollStudents e left join professors on advisor=pcode where e.lcode=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, lcode);
        ResultSet rs=ps.executeQuery();

        JSONArray jArray=new JSONArray();
        while(rs.next()) {
            JSONObject obj=new JSONObject();
            obj.put("scode", rs.getString("scode"));
            obj.put("sname", rs.getString("sname"));
            obj.put("dept", rs.getString("dept"));
            obj.put("pname", rs.getString("pname"));
            obj.put("year", rs.getString("year"));
            obj.put("grade", rs.getInt("grade"));
            String edate="";
            try {
                SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd");
                edate=sdf.format(rs.getDate("edate"));
            } catch (Exception e) {}
            obj.put("edate", edate);
            jArray.add(obj);
        }
        jobject.put("array", jArray);
    } catch (Exception e) {
        System.out.println("특정강좌를 신청한 학생목록:" + e.toString());
    }
    return jobject;
}
```

```
create view enrollStudents as
(select e.*, sname, dept, year, advisor from enrollments e, students s
where e.scode=s.scode);
```

[Java Resources]-[src]-[enrollments]-[EnrollmentsServlet.java]

```
@WebServlet(value= {"/enroll/courses", "/enroll/insert", "/enroll/delete", "/enroll/students", "/enroll/update"})
public class EnrollmentsServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();

        EnrollmentsDAO edao=new EnrollmentsDAO();

        switch(request.getServletPath()) {
            case "/enroll/students":
                jobject=edao.enrollStudents(request.getParameter("lcode"));
                out.println(jobject);
                break;
        }
    }
}
```



```

<div id="divHeader"><h2>성 적 입 력</h2></div>
<div id="divCondition">
  <button id="btnUpdate" style="margin-bottom:5px;">선택저장</button>
  <table id="tbl"></table>
  <script id="temp" type="text/x-handlebars-template">
    <tr class="title">
      <td width=10><input type="checkbox" id="chkAll"></td>
      <td width=100>학생번호</td><td width=100>학생이름</td><td width=100>학과</td>
      <td width=100>지도교수</td><td width=100>학년</td>
      <td width=150>수강신청일</td><td width=100>점수</td>
    </tr>
    {{#each array}}
    <tr class="row">
      <td><input type="checkbox" class="chk"></td>
      <td class="scode">{{scode}}</td>
      <td class="sname">{{sname}}</td>
      <td class="dept">{{dept}}</td>
      <td class="pname">{{pname}}</td>
      <td class="year">{{year}}</td>
      <td class="edate">{{edate}}</td>
      <td><input class="grade" size=2 value="{{grade}}" maxlength=3> <button class="btnUpdate">수정</button></td>
    </tr>
    {{/each}}
  </script>
</div>
<script>
  var lcode="{{vo.lcode}}";
  getList();
  //특정 강좌를 수강신청 한 학생목록
  function getList(){
    $.ajax({
      type:"get",
      url:"/haksa/enroll/students",
      data:{"lcode":lcode},
      dataType:"json",
      success:function(data){
        var template = Handlebars.compile($("#temp").html());
        $("#tbl").html(template(data));
      }
    });
  }
</script>

```

```

public void update(String scode, String lcode, int grade) {
  try {
    String sql="update enrollments set grade=? where scode=? and lcode=?";
    PreparedStatement ps=Database.CON.prepareStatement(sql);
    ps.setInt(1, grade);
    ps.setString(2, scode);
    ps.setString(3, lcode);
    ps.execute();
  }catch(Exception e) {
    System.out.println("점수수정:" + e.toString());
  }
}

```

```

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
  switch(request.getServletPath()) {
    case "/enroll/update":
      String strGrade=request.getParameter("grade");
      int grade=Integer.parseInt(strGrade);
      edao.update(scode, lcode, grade);
      break;
  }
}

```

```

<script>
//점수를 잘못 입력했을 때 alert출력 후 원래값으로 돌려놓기
var preGrade;
$("#tbl").on("focus", ".row .grade", function(){
    preGrade=$(this).val();
});
$("#tbl").on("change", ".row .grade", function(e){
    var row=$(this).parent().parent();
    var grade=row.find(".grade").val();
    if(grade<0 || grade>100){
        alert("점수를 0~100 사이의 값을 입력하세요.");
        $(this).val(preGrade);
        $(this).focus();
    }
});
//전체 체크 버튼을 클릭했을 경우
$("#tbl").on("click", "#chkAll", function(){
    if($("#tbl").is(":checked")){
        $("#tbl .row .chk").each(function(){ $(this).prop("checked",true); });
    }else{
        $("#tbl .row .chk").each(function(){ $(this).prop("checked",false); });
    }
});

//체크 버튼을 클릭한 경우 전체선택인지를 체크
$("#tbl").on("click", ".row .chk", function(){
    var isChkAll=true;
    $("#tbl .row .chk").each(function(){ if(!$(this).is(":checked")) isChkAll=false; });
    if(isChkAll){
        $("#tbl #chkAll").prop("checked", true);
    }else{
        $("#tbl #chkAll").prop("checked", false);
    }
});

//선택 저장 버튼을 클릭한 경우
$("#btnUpdate").on("click", function(){
    if(!confirm("선택한 학생들의 점수를 수정하시겠습니까?")) return;
    $("#tbl .row .chk:checked").each(function(){
        var row=$(this).parent().parent();
        var scode=row.find(".scode").html();
        var grade=row.find(".grade").val();
        $.ajax({
            type:"post",
            url:"/haksa/enroll/update",
            data:{"scode":scode, "lcode":lcode, "grade":grade},
            success:function(){ }
        });
    });
    alert("수정이 완료되었습니다.");
    getList();
});

//각 행의 수정버튼을 클릭한 경우
$("#tbl").on("click", ".row button", function(){
    var row=$(this).parent().parent();
    var scode=row.find(".scode").html();
    var grade=row.find(".grade").val();
    if(!confirm(scode + " 학생의 점수를 수정하시겠습니까?")) return;
    $.ajax({
        type:"post",
        url:"/haksa/enroll/update",
        data:{"scode":scode, "lcode":lcode, "grade":grade},
        success:function(){ alert("수정이 완료되었습니다."); getList(); }
    });
});
$(frm.pcode).click(function(){
    window.open('/haksa/professors/advisor.jsp','advisor','width=320,height=300,top=200,left=900');
});
</script>

```

- 교수 정보 읽기 프로그램

교수 정보

교수번호	<input type="text" value="221"/>	교수학과	<input type="text" value="전산"/>	임용일자	<input type="text" value="1975"/> 년 <input type="text" value="04"/> 월 <input type="text" value="03"/> 일
교수이름	<input type="text" value="이병렬"/>	교수급여	<input type="text" value="3,000,000"/>	교수직급	<input checked="" type="radio"/> 정교수 <input type="radio"/> 부교수 <input type="radio"/> 조교수

[Java Resources]-[src]-[professors]-[ProfessorsDAO.java]

```

public ProfessorsVO read(String pcode) {
    ProfessorsVO vo=new ProfessorsVO();
    try {
        String sql="select * from professors where pcode=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, pcode);
        ResultSet rs=ps.executeQuery();
        while(rs.next()) {
            vo.setPcode(rs.getString("pcode"));
            vo.setPname(rs.getString("pname"));
            vo.setTitle(rs.getString("title"));
            vo.setHireDate(rs.getDate("hiredate"));
            vo.setSalary(rs.getInt("salary"));
        }
    } catch(Exception e) { System.out.println("교수정보:" + e.toString()); }
    return vo;
}

```

[Java Resources]-[src]-[professors]-[ProfessorsServlet.java]

```

@WebServlet(value= {"/professors/list", "/professors/read"})
public class ProfessorsServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        ...
        switch(request.getServletPath()) {
            case "/professors/read":
                request.setAttribute("vo", pdao.read(request.getParameter("pcode")));
                RequestDispatcher dis=request.getRequestDispatcher("read.jsp");
                dis.forward(request, response);
                break;
        }
    }
}

```

[webContent]-[professors]-[list.jsp]

```

<form name="frm" action="update" method="post">
    <table>
        <tr>
            <td width=100 class="title" width=150>교수번호</td>
            <td width=100><input type="text" name="pcode" value="{vo.pcode}" size=10></td>
            <td width=100 class="title" width=150>교수학과</td>
            <td width=100>
                <select name="dept">
                    <option <c:out value="{vo.dept}" selected="" /></option> ...
                </select>
            </td>
            <td width=100 class="title" width=150>임용일자</td>
            <td><input type="text" name="yy" value="{fn:substring(vo.hireDate,0,4)}" maxlength=4 size=2>년...</td>
        </tr>
        <tr>
            <td width=100 class="title" width=150>교수이름</td>
            <td width=100><input type="text" name="pname" value="{vo.pname}" size=10></td>
            <td width=100 class="title" width=150>교수급여</td>
            <td><input type="text" name="salary" value="{fn:formatNumber value="{vo.salary}" pattern="#,###" />></td>
            <td width=100 class="title" width=150>교수직급</td>
            <td><input type="radio" name="title" value="정교수" <c:out value="{vo.title}" checked="" />>정교수 ...</td>
        </tr>
    </table>
</form>

```

- 교수의 담당과목 목록, 담당학생 목록 프로그램

담당강좌

강좌번호	강좌이름	강의실	강의시수	최대수강인원	수강신청인원	강좌정보
C301	파일처리론	506	3	100	80	강좌정보
C401	데이터베이스	414	3	80	80	강좌정보

담당학생

학생번호	학생이름	학과	학년	생년월일	학생정보
92414033	김창덕	전산	4	1991-02-03	학생정보
95414058	박혜경	전산	4	1976-03-12	학생정보

[Java Resources]-[src]-[professors]-[ProfessorsDAO.java]

//특정교수가 담당하는 강좌목록, 학생목록

```
public JSONObject infoProfessors(String pcode) {
    JSONObject jObject=new JSONObject();
    try {
        String sql="call list_courses_students(?)";
        CallableStatement cs=Database.CON.prepareCall(sql);
        cs.setString(1, pcode);
        cs.execute();

        ResultSet rs=cs.getResultSet();
        JSONArray jArray=new JSONArray();
        while(rs.next()) {
            JSONObject obj=new JSONObject();
            ....
            jArray.add(obj);
        }
        jObject.put("cArray", jArray);

        cs.getMoreResults();
        rs=cs.getResultSet();

        jArray=new JSONArray();
        while(rs.next()) {
            JSONObject obj=new JSONObject();
            ...
            try {
                SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd");
                obj.put("birthday", sdf.format(rs.getDate("birthday")));
            }catch(Exception e) {}
            jArray.add(obj);
        }
        jObject.put("sArray", jArray);
    }catch(Exception e) { System.out.println("특정교수가 담당하는 강좌목록, 학생목록:" + e.toString()); }
    return jObject;
}
```

```
CREATE DEFINER='root'@'localhost' PROCEDURE `LIST_COURSES_STUDENTS`(
    in i_pcode char(3))
BEGIN
    select * from courses where instructor=i_pcode;
    select * from students where advisor=i_pcode;
END
```

[Java Resources]-[src]-[professors]-[ProfessorsServlet.java]

```
@WebServlet(value= {"/professors/list", "/professors/read", "/professors/info"})
public class ProfessorsServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        ...
        switch(request.getServletPath()) {
            case "/professors/info":
                out.println(pdao.infoProfessors(request.getParameter("pcode")));
                break;
        }
    }
}
```

```

<div id="divHeader"><h2>답 당 강 좌</h2></div>
<div id="divCondition">
  <table id="tblCourses"></table>
  <script id="tempCourses" type="text/x-handlebars-template">
    <tr class="title"> ... </tr>
    {{#each cArray}}
    <tr class="row">
      <td class="lcode">{{lcode}}</td>
      <td class="name">{{lname}}</td>
      <td class="edate">{{room}}</td>
      <td class="pname">{{hours}}</td>
      <td class="room">{{capacity}}</td>
      <td class="hours">{{persons}}</td>
      <td><button>강좌정보</td>
    </tr>
    {{/each}}
  </script>
</div>
<div id="divHeader"><h2>답 당 학 생</h2></div>
<div id="divCondition">
  <table id="tblStudents"></table>
  <script id="tempStudents" type="text/x-handlebars-template">
    <tr class="title"> ... </tr>
    {{#each sArray}}
    <tr class="row">
      <td class="scode">{{scode}}</td>
      <td class="sname">{{sname}}</td>
      <td class="dept">{{dept}}</td>
      <td class="year">{{year}}</td>
      <td class="birthday">{{birthday}}</td>
      <td><button>학생정보</td>
    </tr>
    {{/each}}
  </script>
</div>
</body>
<script>
  var pcode="{{vo.pcode}}";
  getList();

  //담당강좌목록, 담당학생목록 출력
  function getList(){
    $.ajax({
      type:"get",
      url:"info",
      data:{"pcode":pcode},
      dataType:"json",
      success:function(data){
        var template = Handlebars.compile($("#tempCourses").html());
        $("#tblCourses").html(template(data));
        var template = Handlebars.compile($("#tempStudents").html());
        $("#tblStudents").html(template(data));
      }
    });
  }

  //강좌정보 버튼을 클릭한 경우
  $("#tblCourses").on("click", ".row button", function(){
    var row=$(this).parent().parent();
    var lcode=row.find(".lcode").html();
    location.href="/haksa/courses/read?lcode=" + lcode;
  });

  //학생정보 버튼을 클릭한 경우
  $("#tblStudents").on("click", ".row button", function(){
    var row=$(this).parent().parent();
    var scode=row.find(".scode").html();
    location.href="/haksa/students/read?scode=" + scode;
  });
</script>
</html>

```

쇼핑몰 프로그램

Home 업체목록 상품목록 상품등록 주문목록 장바구니

상 품 목 록

검색수: 12건



P101
아모라 아쿠아텍스 이지클린 침대
2200000원
판매중지



P102
모던 라운지 낮은 퀸 침대
2100000원
CART담기



P103
도로시 아쿠아텍스 기능성 퀸 침대
3000000원
CART담기



P104
모리엘 호텔형 침대
1500000원
CART담기



P105
글로리 모던 서랍장 와이드
600000원
CART담기



P106
카밀라 4인용 극세사가죽 소파
2500000원
CART담기



P107
루진 6인 원목 식탁세트 (벤치형)
3000000원
CART담기



P108
베스트 10.5자 장롱 시리즈
400000원
CART담기

◀ ▶ [1/2]

Home 업체목록 상품목록 상품등록 주문목록 장바구니

업 체 목 록

검색수: 5건

업체코드	업체명	주소	전화번호	이메일
M101	에몬스 가구	서울 강서구 대치동 561번지	02-522-6898	Blllee@hanmail.net
M102	벨라지오 퍼니처	서울 강북구 수유동 134번지	02-897-6690	Jglee@netian.com
M103	까사미아	인천 부평구 계산동 104번지	032-502-9599	Ywshu@netian.com

◀ ▶ [1/2]

Home 업체목록 상품목록 상품등록 주문목록 장바구니

상 품 목 록

검색수: 12건

코드	상품명	업체명	제조사	일반가	판매가	장바구니	상품정보
P101	아모라 아쿠아텍스 이지클린 침대	에몬스 가구	가구나라	2200000	2200000	<input type="button" value="장바구니"/>	<input type="button" value="상품정보"/>
P102	모던 라운지 낮은 퀸 침대	에몬스 가구	삼익 가구나라	2100000	2100000	<input type="button" value="장바구니"/>	<input type="button" value="상품정보"/>
P103	도로시 아쿠아텍스 기능성 퀸 침대	벨라지오 퍼니처	한샘가구	3000000	3000000	<input type="button" value="장바구니"/>	<input type="button" value="상품정보"/>
P104	모리엘 호텔형 침대	벨라지오 퍼니처	한샘가구	1500000	1500000	<input type="button" value="장바구니"/>	<input type="button" value="상품정보"/>
P105	글로리 모던 서랍장 와이드	에몬스 가구	한샘가구	600000	600000	<input type="button" value="장바구니"/>	<input type="button" value="상품정보"/>

◀ ▶ [1/3]

상 품 등 록

상품코드	P113	제조원/수입원	가구코리아	판매가격	250000
업체코드				일반가격	250000
상품명	킹침대(진주 매트포함)				
상품이미지	150 x 120				
상품설명					

저장

취소

상 품 정 보

상품코드	P101	제조원/수입원	가구나라	판매가격	2200000
업체코드	M101	에몬스 가구		일반가격	2300000
상품명	아모라 아쿠아텍스 이지클린 침대			판매상태	<input checked="" type="checkbox"/> 판매중지
상품이미지					
상품설명					

저장

삭제

취소

업 체 목 록

업체코드	업체명	전화번호
M101	에몬스 가구	02-522-6898
M102	벨라지오 퍼니처	02-897-6690
M103	까사미아	032-502-9599
M104	레이디 가구	02-602-9876
M105	가구 나라	032-503-9021



[1/1]

주문 목록

주문코드 ▼ 3행 ▼ 검색 검색수: 10건 주문코드 ▼ 오름자순 ▼

주문번호	주문자명	주소	전화번호	구매정보
R107	이준호	인천 계양구 효성동	032-522-5678	구매정보
R108	김연호	서울 종로구 관훈동 관훈빌딩	02-688-7703	구매정보
R109	김수정	서울 강남구 대치1동	02-503-9090	구매정보

◀ ▶ [3/4]

이름	이준호	전화	032-522-5678	이메일	jhlee@hanmil.net	구매일	2014-08-15 00:00:00
주소	인천 계양구 효성동			결제	무통장	상태	처리중

상품코드	상품명	제조사	가격	수량	합계
P109	줄리아 화이트 10자/12자 장농	동서가구	520000	1	520000
P111	리트머스 1500 거실장	한샘가구	120000	1	120000
P112	엘린 4인용 카우치 가죽소파	가구나라	150000	1	150000

장 바 구 니

<input type="checkbox"/>	코드	상품명	단가	수량	합계	삭제
<input type="checkbox"/>	P102	모던 라운지 낮은 퀸 침대	2100000	<input type="text" value="1"/> 수정	2100000	삭제
<input type="checkbox"/>	P103	도로시 아쿠아텍스 기능성 퀸 침대	3000000	<input type="text" value="1"/> 수정	3000000	삭제
<input type="checkbox"/>	P104	모리엘 호텔형 침대	1500000	<input type="text" value="1"/> 수정	1500000	삭제

[전체상품주문](#) [선택상품주문](#)

주문 정보

상품코드	상품명	상품가격	상품수량	합계
P102	모던 라운지 낮은 퀸 침대	2100000	1	2100000
P103	도로시 아쿠아텍스 기능성 퀸 침대	3000000	1	3000000
P104	모리엘 호텔형 침대	1500000	1	1500000
합계				6600000

주문자 성명	<input type="text" value="홍길동"/>
배송지 주소	<input type="text" value="인천 서구 경서동 등문 굿모닝힐"/>
이메일 주소	<input type="text" value="hong@daum.net"/>
전화번호	<input type="text" value="010-0210-1220"/>
결제방법	<input checked="" type="radio"/> 무통장 <input type="radio"/> 카드

[주문하기](#)

- 데이터베이스 생성

쇼핑몰 user 생성

```
create user shop identified by pass;  
grant connect, resource, dba to shop;
```

업체 테이블 생성

```
create table mall(  
    mall_id char(4) not null,  
    mall_name varchar(50) not null,  
    manager varchar(15),  
    address varchar(100),  
    tel varchar(15),  
    email varchar(20),  
    detail varchar(100),  
    primary key(mall_id)  
);
```

업체 테이블 데이터입력

```
insert into mall(mall_id,mall_name,manager,address,tel,email)  
values('M101','에온스 가구','이병렬','서울 강서구 대치동 561번지','02-522-6898','Bllee@hanmail.net');  
insert into mall(mall_id,mall_name,manager,address,tel,email)  
values('M102','벨라지오 퍼니처','이재광','서울 강북구 수유동 134번지','02-897-6690','Jglee@netian.com');  
insert into mall(mall_id,mall_name,manager,address,tel,email)  
values('M103','까사미아','서연우','인천 부평구 계산동 104번지','032-502-9599','Ywshu@netian.com');  
insert into mall(mall_id,mall_name,manager,address,tel,email)  
values('M104','레이디 가구','김성철','서울 강남구 압구정동 456번지','02-602-9876','Sigang@netian.com');  
insert into mall(mall_id,mall_name,manager,address,tel,email)  
values('M105','가구 나라','이원구','인천 남동구 만수동 567번지','032-503-9021','Wglee@hanmail.net');
```

상품 테이블 생성

```
create table product(  
    prod_id char(4) not null,  
    prod_name varchar(50) not null,  
    mall_id char(4),  
    company varchar(50),  
    pricel int,  
    price2 int,  
    detail varchar(3000),  
    image varchar(100),  
    prod_del char(1) default '0',  
    primary key(prod_id),  
    foreign key(mall_id) references mall(mall_id)  
);
```

상품 테이블 데이터입력

```
insert into product(prod_id,prod_name,mall_id,company,pricel,price2)  
values('P101','아모라 아쿠아텍스 이지클린 침대','M101','가구나라',2200000,2300000);  
insert into product(prod_id,prod_name,mall_id,company,pricel,price2)  
values('P102','모던 라운지 낮은 쿤 침대','M101','삼익 가구나라',2100000,2200000);  
insert into product(prod_id,prod_name,mall_id,company,pricel,price2)  
values('P103','도로시 아쿠아텍스 기능성 쿤 침대','M102','한샘가구',3000000,3200000);  
insert into product(prod_id,prod_name,mall_id,company,pricel,price2)  
values('P104','모리엘 호텔형 침대','M102','한샘가구',1500000,1600000);  
insert into product(prod_id,prod_name,mall_id,company,pricel,price2)  
values('P105','글로리 모던 서랍장 와이드','M101','한샘가구',600000,650000);  
insert into product(prod_id,prod_name,mall_id,company,pricel,price2)  
values('P106','카밀라 4인용 극세사가죽 소파','M102','까사미아',2500000,2600000);  
insert into product(prod_id,prod_name,mall_id,company,pricel,price2)  
values('P107','루진 6인 원목 식탁세트 (벤치형)','M101','까사미아',300000,320000);  
insert into product(prod_id,prod_name,mall_id,company,pricel,price2)  
values('P108','베스트 10.5자 장롱 시리즈','M103','동서가구',400000,420000);  
insert into product(prod_id,prod_name,mall_id,company,pricel,price2)  
values('P109','줄리아 화이트 10자/12자 장롱','M104','동서가구',520000,550000);  
insert into product(prod_id,prod_name,mall_id,company,pricel,price2)  
values('P110','글로리 모던 서랍장 830','M105','가구나라',100000,120000);  
insert into product(prod_id,prod_name,mall_id,company,pricel,price2)  
values('P111','리트머스 1500 거실장','M102','한샘가구',120000,130000);  
insert into product(prod_id,prod_name,mall_id,company,pricel,price2)  
values('P112','엘런 4인용 카우치 가죽소파','M102','가구나라',1500000,1670000);
```

구매 테이블 생성

```
create table purchase(
  order_id char(4) not null,
  name varchar(15) not null,
  address varchar(100),
  email varchar(20),
  tel varchar(15),
  pdate date,
  payType char(1), /* 0:무통장, 1:카드 */
  status char(1) default '0', /* 0:처리중, 1:처리완료 */
  primary key(order_id)
);
```

구매 테이블 데이터입력

```
insert into purchase(order_id, name, address, email, tel, pdate, payType, status)
  values('R101', '김수정', '서울 강남구 대치1동', 'Sjkim@netian.com', '02-503-9090', '2014/08/10', '1', '1');
insert into purchase(order_id, name, address, email, tel, pdate, payType, status)
  values('R102', '박혜경', '인천 부평구 부평5동', 'Hkpark@hanmail.net', '032-609-9987', '2014/08/10', '1', '1');
insert into purchase(order_id, name, address, email, tel, pdate, payType, status)
  values('R103', '임미숙', '인천 서구 청라2동', 'mslim@hanmail.net', '02-688-7703', '2014/08/13', '1', '1');
insert into purchase(order_id, name, address, email, tel, pdate, payType, status)
  values('R104', '김연호', '서울 종로구 관훈동 관훈빌딩', 'Yhkim@yahoo.com', '02-688-7703', '2014/08/13', '1', '0');
insert into purchase(order_id, name, address, email, tel, pdate, payType, status)
  values('R105', '한보연', '서울 종로구 인사동', 'byhan@yahoo.com', '02-230-2044', '2014/08/13', '1', '1');
insert into purchase(order_id, name, address, email, tel, pdate, payType, status)
  values('R106', '김수정', '서울 강남구 대치1동', 'Sjkim@netian.com', '02-503-9090', '2014/08/13', '0', '0');
insert into purchase(order_id, name, address, email, tel, pdate, payType, status)
  values('R107', '이준호', '인천 계양구 효성동', 'Jhlee@hanmil.net', '032-522-5678', '2014/08/14', '0', '1');
insert into purchase(order_id, name, address, email, tel, pdate, payType, status)
  values('R108', '김연호', '서울 종로구 관훈동 관훈빌딩', 'Yhkim@yahoo.com', '02-688-7703', '2014/08/15', '1', '1');
insert into purchase(order_id, name, address, email, tel, pdate, payType, status)
  values('R109', '김수정', '서울 강남구 대치1동', 'Sjkim@netian.com', '02-503-9090', '2014/08/15', '1', '1');
insert into purchase(order_id, name, address, email, tel, pdate, payType, status)
  values('R110', '이준호', '인천 계양구 효성동', 'Jhlee@hanmil.net', '032-522-5678', '2014/08/15', '0', '0');
```

주문상품 테이블 생성

```
create table orders(
  order_id char(4) not null,
  prod_id char(4) not null,
  price int,
  quantity int,
  primary key(order_id, prod_id),
  foreign key (order_id) references purchase(order_id),
  foreign key (prod_id) references product(prod_id)
);
```

주문 테이블 데이터입력

```
insert into orders(order_id, prod_id, price, quantity) values('R101', 'P109', 520000, 1);
insert into orders(order_id, prod_id, price, quantity) values('R101', 'P112', 150000, 1);
insert into orders(order_id, prod_id, price, quantity) values('R101', 'P107', 300000, 2);
insert into orders(order_id, prod_id, price, quantity) values('R102', 'P106', 2500000, 1);
insert into orders(order_id, prod_id, price, quantity) values('R102', 'P112', 150000, 1);
insert into orders(order_id, prod_id, price, quantity) values('R103', 'P107', 300000, 2);
insert into orders(order_id, prod_id, price, quantity) values('R103', 'P103', 3000000, 1);
insert into orders(order_id, prod_id, price, quantity) values('R104', 'P109', 520000, 2);
insert into orders(order_id, prod_id, price, quantity) values('R104', 'P107', 300000, 1);
insert into orders(order_id, prod_id, price, quantity) values('R104', 'P111', 120000, 1);
insert into orders(order_id, prod_id, price, quantity) values('R105', 'P110', 100000, 1);
insert into orders(order_id, prod_id, price, quantity) values('R105', 'P103', 3000000, 1);
insert into orders(order_id, prod_id, price, quantity) values('R106', 'P112', 150000, 2);
insert into orders(order_id, prod_id, price, quantity) values('R107', 'P112', 150000, 4);
insert into orders(order_id, prod_id, price, quantity) values('R107', 'P103', 3000000, 1);
insert into orders(order_id, prod_id, price, quantity) values('R108', 'P104', 1500000, 2);
insert into orders(order_id, prod_id, price, quantity) values('R109', 'P103', 3000000, 1);
insert into orders(order_id, prod_id, price, quantity) values('R110', 'P109', 520000, 1);
insert into orders(order_id, prod_id, price, quantity) values('R110', 'P111', 120000, 1);
insert into orders(order_id, prod_id, price, quantity) values('R110', 'P112', 150000, 1);
```

- List Stored Procedure 작성 (SET SERVEROUTPUT ON; /* 결과를 화면에 출력하기 위한 설정 */)

```
variable cur refcursor; variable cnt number; exec list('mall','mall_id','1,2','mall_id',:cur, :cnt); print cur; print cnt;
```

```
create or replace PROCEDURE LIST(
```

```
    i_table in varchar,
    i_key in varchar,
    i_word in varchar,
    i_page in int,
    i_perPage in int,
    i_order in varchar,
    i_desc in varchar,
    o_cur out sys_refcursor,
    o_count out int)
```

```
select * from
(select rownum rn, tbl.* from
(select * from purchase where address like '%서울%' order by order_id desc) tbl)
where rn between 3 and 4;
```

```
AS
```

```
    v_tbl1 varchar(1000);
    v_tbl2 varchar(1000);
    v_sql varchar(1000);
    v_word varchar(100) := '' || i_word || '';
    v_start int := (i_page-1)*i_perPage +1;
    v_end int:=i_page * i_perPage;
```

```
BEGIN
```

```
    v_tbl1:=(select * from '||i_table||' where '||i_key||' like '||v_word||' order by '||i_order||' '||i_desc||') tbl';
    v_tbl2:=(select rownum rn, tbl.* from ' || v_tbl1 || ');
    v_sql:='select * from ' || v_tbl2 || ' where rn between ' || v_start || ' and ' || v_end;
    DBMS_OUTPUT.PUT_LINE(v_sql);
    OPEN o_cur FOR v_sql;
    v_sql := 'select count(*) from '|| v_tbl1;
    EXECUTE IMMEDIATE v_sql INTO o_count;
```

```
END LIST;
```

[Java Resources]-[src]-[mall]-[Database.java]

```
public class Database{
    public static Connection CON;
    static {
        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
            CON=DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1:1521:xe", "shop", "pass");
            System.out.println("접속성공");
        } catch(Exception e) { System.out.println("접속실패:" + e.toString()); }
    }
}
```

[Java Resources]-[src]-[mall]-[SqlVO.java]

```
public class SQLVO {
    private String key;
    private String word;
    private String order;
    private String desc;
    private int page;
    private int perPage;
    //...getter(),setter(),toString();
}
```

- 쇼핑몰 템플릿



- 쇼핑몰 StyleSheet

[webContent]-[home.css]

```
body{background:gray;}
#divPage{width:980px;
  background:white;
  box-shadow:10px 10px 10px blak;
  margin:auto;
  padding:0px 0px 20px 0px;}
#divMenu{margin:auto;
  overflow:hidden;
  background:#EAEAEA;
  padding:15px 0px 15px 20px;
  border-bottom:1px solid gray;}
#divMenu .item{width:100px;
  float:left;
  text-align:center;}
#divHeader{text-align:center;}
#divCondition{width:780px;
  margin:auto; overflow:
  hidden;padding:5px 10px 5px 10px;
  border:1px solid green;
  margin-bottom:10px;
  background:#EAEAEA;}
#divSearch{float:left; }
#divSort{float:right; }
#tbl{width:800px;
  margin:auto;
  overflow:hidden;}
#pagination{width:800px; margin:auto; text-align:center; padding:10px 0px 10px 0px;}
.price, .quantity, .sum {text-align:right;}
a{text-decoration:none; color:green;}
a:hover{font-weight:bold;}
select, input, button {height:23px;}
table {border-collapse:collapse;}
td{border:1px solid black; height:30px; font-size:12px; padding:0px 5px 0px 5px;}
table .title{background:#EAEAEA; text-align:center;}
table .row:hover{background:gray; color:white;}
input, button, select, textarea {font-size:12px;}
input[type='radio'], input[type='checkbox'] {height:12px;}
td button{text-align:center;}
textarea{margin:5px 0px 5px 0px;}
```

[WebContent]-[menu.jsp]

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<div class="item"><a href="">Home</a></div>
<div class="item"><a href="">업체목록</a></div>
<div class="item"><a href="">상품목록</a></div>
<div class="item"><a href="">상품등록</a></div>
<div class="item"><a href="">주문목록</a></div>
<div class="item"><a href="">장바구니</a></div>
```

[webContent]-[index.jsp]

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<html>
<head>
  <title>쇼핑몰</title>
  <link rel="stylesheet" href="home.css">
</head>
<body>
<div id="divPage" style="height:300px;">
  <div id="divMenu"><jsp:include page="menu.jsp"/></div>
  <!--내용 입력 시작 ----->

  <!--내용 입력 종료 ----->
</div>
</body>
</html>
```

- 업체 목록 프로그램

업체 목록

업체코드 ▼

3행 ▼

검색

검색수: 5건

업체코드 ▼

오름차순 ▼

업체코드	업체명	주소	전화번호	이메일
M101	에몬스 가구	서울 강서구 대치동 561번지	02-522-6898	Blllee@hanmail.net
M102	벨라지오 퍼니처	서울 강북구 수유동 134번지	02-897-6690	Jglee@netian.com
M103	까사미아	인천 부평구 계산동 104번지	032-502-9599	Ywshu@netian.com

◀ ▶ [1/2]

[Java Resources]-[src]-[mall]-[MallVO.java]

```
public class MallVO {
    private String mall_id;
    private String mall_name;
    private String manager;
    private String address;
    private String tel;
    private String email;
    private String detail;
    //... getter() getter() toString();
}
```

[Java Resources]-[src]-[mall]-[MallDAO.java]

```
public class MallDAO {
    public JSONObject list(SqlVO sqlVO) {
        JSONObject jObject=new JSONObject();
        try {
            String sql="call list(?,?,?, ?, ?, ?, ?)";
            CallableStatement cs=Database.CON.prepareCall(sql);
            cs.setString(1, "mall");
            cs.setString(2, sqlVO.getKey());
            cs.setString(3, sqlVO.getWord());
            cs.setInt(4, sqlVO.getPage());
            cs.setInt(5, sqlVO.getPerPage());
            cs.setString(6, sqlVO.getOrder());
            cs.setString(7, sqlVO.getDesc());
            cs.registerOutParameter(8, oracle.jdbc.OracleTypes.CURSOR);
            cs.registerOutParameter(9, oracle.jdbc.OracleTypes.INTEGER);

            cs.execute();
            ResultSet rs=(ResultSet)cs.getObject(8);

            JSONArray jArray=new JSONArray();
            while(rs.next()) {
                JSONObject obj=new JSONObject();
                obj.put("mall_id", rs.getString("mall_id"));
                obj.put("mall_name", rs.getString("mall_name"));
                obj.put("manager", rs.getString("manager"));
                obj.put("address", rs.getString("address"));
                obj.put("tel", rs.getString("tel"));
                obj.put("email", rs.getString("email"));
                jArray.add(obj);
            }
            jObject.put("array", jArray);

            int count=cs.getInt(9);
            int perPage=sqlVO.getPerPage();
            int totPage=count % perPage==0 ? count / perPage:(count / perPage) + 1;

            jObject.put("count", count);
            jObject.put("totPage", totPage);
            jObject.put("page", sqlVO.getPage());
        }catch(Exception e) { System.out.println("업체목록:" + e.toString()); }
        return jObject;
    }
}
```

```

@WebServlet(value= ("/mall/list"))
public class MallServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        MallDAO dao=new dao();

        switch(request.getServletPath()) {
        case "/mall/list":
            String key=request.getParameter("key")==null?"mall_id":request.getParameter("key");
            String word=request.getParameter("word")==null?"":request.getParameter("word");
            String strPage=request.getParameter("page")==null?"1":request.getParameter("page");
            int page=Integer.parseInt(strPage);
            String strPerPage=request.getParameter("perPage")==null?"5":request.getParameter("perPage");
            int perPage=Integer.parseInt(strPerPage);
            String order=request.getParameter("order")==null?"mall_id":request.getParameter("order");
            String desc=request.getParameter("desc")==null?"":request.getParameter("desc");

            SqlVO sqlVO=new SqlVO();
            sqlVO.setKey(key);
            sqlVO.setWord(word);
            sqlVO.setPage(page);
            sqlVO.setPerPage(perPage);
            sqlVO.setOrder(order);
            sqlVO.setDesc(desc);
            JSONObject jObject=dao.list(sqlVO);
            out.println(jObject);
            break;
        }
    }
}

```

```

<html>
<head>
    <title>쇼핑몰</title>
    <script src="http://code.jquery.com/jquery-3.1.1.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/handlebars.js/3.0.1/handlebars.js"></script>
    <link rel="stylesheet" href="../home.css"/>
</head>
<body>
<div id="divPage">
    <div id="divMenu"><jsp:include page="../menu.jsp"/></div>
    <div id="divHeader"><h2>업 체 목 록 </h2></div>
    <div id="divCondition">
        <div id="divSearch">
            <select id="selKey">
                <option value="mall_id">업체코드</option>
                <option value="mall_name">업체명</option>
                <option value="address">주소</option>
            </select>
            <input type="text" id="txtWord">
            <select id="selPerPage">
                <option value="3">3행</option><option value="5">5행</option><option value="10">10행</option>
            </select>
            <input type="button" id="btnSearch" value="검색">
            <span style="font-size:12px;">검색수: <b id="count"></b>건</span>
        </div>
        <div id="divSort">
            <select id="selOrder">
                <option value="mall_id">업체코드</option>
                <option value="mall_name">업체명</option>
                <option value="address">주소</option>
            </select>
            <select id="selDesc">
                <option value="">오름차순</option><option value="desc">내림차순</option>
            </select>
        </div>
    </div>
</div>

```

```

<table id="tbl"></table>
<script id="temp" type="text/x-handlebars-template">
  <tr class="title">
    <td width=100>업체코드</td>
    <td width=100>업체명</td>
    <td width=300>주소</td>
    <td width=150>전화번호</td>
    <td width=150>이메일</td>
  </tr>
  {{#each array}}
  <tr class="row">
    <td>{{mall_id}}</td>
    <td>{{mall_name}}</td>
    <td>{{address}}</td>
    <td>{{tel}}</td>
    <td>{{email}}</td>
  </tr>
  {{/each}}
</script>
<div id="pagination">
  <button id="btnPre">◀</button> <button id="btnNext">▶</button>
  [<span id="curPage"></span> / <span id="totPage"></span>]
</div>
</div>
</body>
<script>
  var url="/shop/mall/list";
</script>
<script src="../../home.js"></script>
</html>

```

[webContent]-[home.js]

```

var key, word, order, desc, perPage, page=1;
getList();

$("#btnSearch").on("click", function(){ page=1; getList(); });
$("#selOrder, #selDesc, #selPerPage").change(function(){ page=1; getList(); });
$("#txtWord").keydown(function(key){ if(key.keyCode==13){ page=1; getList(); } });
$("#btnNext").click(function(){ page++; getList(); });
$("#btnPre").click(function(){ page--; getList(); });

function getList(){
  key=$("#selKey").val();
  word=$("#txtWord").val();
  order=$("#selOrder").val();
  desc=$("#selDesc").val();
  perPage=$("#selPerPage").val();

  $.ajax({
    type:"get",
    url:url,
    data:{'key':key,'word':word,'page':page,'perPage':perPage,'order':order,'desc':desc},
    dataType:"json",
    success:function(data){
      var template = Handlebars.compile($("#temp").html());
      $("#tbl").html(template(data));
      if(data.page==1){
        $("#btnPre").attr("disabled",true);
      }else{ $("#btnPre").attr("disabled",false); }
      if(data.page==data.totPage){
        $("#btnNext").attr("disabled",true);
      }else{ $("#btnNext").attr("disabled",false); }
      $("#curPage").html(data.page);
      $("#totPage").html(data.totPage);
      $("#count").html(data.count);
    }
  });
}

```

- 상품 목록 프로그램

상 품 목 록

상품코드 ▼

3행 ▼

검색

검색수: 12건

상품코드 ▼

오름차순 ▼

코드	상품명	업체명	제조사	일반가	판매가	장바구니	상품정보
P101	아모라 아쿠아텍스 이지클린 침대	에몬스 가구	가구나라	2200000	2200000	장바구니	상품정보
P102	모던 라운지 낮은 퀸 침대	에몬스 가구	삼익 가구나라	2100000	2100000	장바구니	상품정보
P103	도로시 아쿠아텍스 기능성 퀸 침대	벨라지오 퍼니처	한샘가구	3000000	3000000	장바구니	상품정보

◀

▶

[1/4]

[Java Resources]-[src]-[product]-[ProductVO.java]

```
public class ProductVO extends MallVO{
    private String prod_id;
    private String prod_name;
    private String company;
    private int price1;
    private int price2;
    private String detail;
    private String image;
    private String prod_del;
    // ... setter()/ getter()/ toString();
}
```

[Java Resources]-[src]-[product]-[ProductDAO.java]

```
public JSONObject list(SqlVO sqlVO){
    JSONObject jobject=new JSONObject();
    CallableStatement cs=null;
    ResultSet rs=null;
    try {
        String sql="{call list(?,?,?,?,?,?,?,?)}";
        cs=Database.CON.prepareCall(sql);
        String tbl="(select p.*, mall_name from product p, mall m where p.mall_id=m.mall_id)";
        cs.setString(1, tbl);
        ...
        cs.registerOutParameter(8, oracle.jdbc.OracleTypes.CURSOR);
        cs.registerOutParameter(9, oracle.jdbc.OracleTypes.INTEGER);
        cs.execute();
        rs=(ResultSet)cs.getObject(8);

        JSONArray jArray=new JSONArray();
        while(rs.next()) {
            JSONObject obj=new JSONObject();
            ...
            jArray.add(obj);
        }
        jobject.put("array", jArray);

        int count=cs.getInt(9);
        int perPage=sqlVO.getPerPage();
        int totPage=count%perPage==0?count/perPage:(count/perPage)+1;
        jobject.put("count", count);
        jobject.put("totPage", totPage);
        jobject.put("page", sqlVO.getPage());
    }catch(Exception e) { System.out.println("상품목록:" + e.toString());
    }finally{
        try {
            rs.close();
            cs.close();
        }catch(Exception e) {}
    }
    return jobject;
}
```



```

@WebServlet(value= "/product/list")
public class ProductServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        ProductDAO dao=new ProductDAO();

        switch(request.getServletPath()) {
        case "/product/list":
            String key=request.getParameter("key")==null?"prod_id":request.getParameter("key");
            String word=request.getParameter("word")==null?"":request.getParameter("word");
            String strPage=request.getParameter("page")==null?"1":request.getParameter("page");
            int page=Integer.parseInt(strPage);
            String strPerPage=request.getParameter("perPage")==null?"5":request.getParameter("perPage");
            int perPage=Integer.parseInt(strPerPage);
            String order=request.getParameter("order")==null?"prod_id":request.getParameter("order");
            String desc=request.getParameter("desc")==null?"":request.getParameter("desc");

            SqlVO sqlVO=new SqlVO();
            sqlVO.setKey(key);
            sqlVO.setWord(word);
            sqlVO.setPage(page);
            sqlVO.setPerPage(perPage);
            sqlVO.setOrder(order);
            sqlVO.setDesc(desc);
            JSONObject jObject=dao.list(sqlVO);
            out.println(jObject);
            break;
        }
    }
}

```

```

<body>
<div id="divPage">
    <div id="divMenu"><jsp:include page="../menu.jsp"/></div>
    <div id="divHeader"><h2>상 품 목 록 </h2></div>
    <div id="divCondition">
        ...
    </div>
    <table id="tbl"></table>
    <script id="temp" type="text/x-handlebars-template">
        <tr class="title">
            <td width=30>코드</td><td width=210>상품명</td><td width=150>업체명</td><td width=150>제조사</td>
            <td width=50>일판가</td><td width=50>판매가</td><td width=70>장바구니</td><td width=70>상품정보</td>
        </tr>
        {{#each array}}
        <tr class="row">
            <td class="id">{{prod_id}}</td><td>{{prod_name}}</td><td>{{mall_name}}</td><td>{{company}}</td>
            <td class="price">{{price}}</td>
            <td class="price">{{price}}</td>
            <td><button class="btnCart">장바구니</button></td>
            <td><button class="btnRead">상품정보</button></td>
        </tr>
        {{/each}}
    </script>
    <div id="pagination">
        <button id="btnPre">◀</button><button id="btnNext">▶</button>
        [<span id="curPage"></span></span><span id="totPage"></span>]
    </div>
</div>
</body>

<script>
    var url="/shop/product/list";
</script>
<script src="../home.js"></script>

```

- 주문 목록 프로그램

주문 목록

주문코드 ▼	<input type="text"/>	3행 ▼	검색	검색수: 10건	주문코드 ▼	오름차순 ▼
--------	----------------------	------	----	----------	--------	--------

주문번호	주문자명	주소	전화번호	구매정보
R101	김수정	서울 강남구 대치1동	02-503-9090	구매정보
R102	박혜경	인천 부평구 부평5동	032-609-9987	구매정보
R103	임미숙	인천 서구 청라2동	02-688-7703	구매정보

◀ ▶ [1/4]

이름	김수정	전화	02-503-9090	이메일	Sjkim@netian.com	구매일	2014-08-10 00:00:00
주소	서울 강남구 대치1동			결제	카드	상태	완료

상품코드	상품명	제조사	가격	수량	합계
P107	루진 6인 원목 식탁세트 (벤치형)	까사미아	300000	2	600000
P109	줄리아 화이트 10자/12자 장농	동서가구	520000	1	520000
P112	엘린 4인용 카우치 가죽소파	가구나라	150000	1	150000

[Java Resources]-[src]-[purchase]-[PurchaseVO.java]

```
public class PurchaseVO {
    private String order_id;
    private String name;
    private String address;
    private String email;
    private String tel;
    private String pdate;
    private String payType;
    private String status;
    // ... setter()/ getter()/ toString();
}
```

[Java Resources]-[src]-[purchase]-[PurchaseDAO.java]

```
public JSONObject list(SqlVO sqlVO) {
    JSONObject jobject=new JSONObject();
    try {
        String sql="{call list(?,?,?,?,?,?,?,?)}";
        CallableStatement cs=Database.CON.prepareCall(sql);
        ...
        cs.registerOutParameter(8, oracle.jdbc.OracleTypes.CURSOR);
        cs.registerOutParameter(9, oracle.jdbc.OracleTypes.INTEGER);
        cs.execute();
        ResultSet rs=(ResultSet)cs.getObject(8);
        JSONArray jArray=new JSONArray();
        while(rs.next()) {
            JSONObject obj=new JSONObject();
            ...
            jArray.add(obj);
        }
        jobject.put("array", jArray);
        int count=cs.getInt(9);
        int perPage=sqlVO.getPerPage();
        int totPage=count%perPage==0?count/perPage:(count/perPage)+1;
        jobject.put("count", count);
        jobject.put("totPage", totPage);
        jobject.put("page", sqlVO.getPage());
    }catch(Exception e) { System.out.println("구매목록:" + e.toString()); }
    return jobject;
}
```

[Java Resources]-[src]-[purchase]-[PurchaseDAO.java]

```
public PurchaseVO read(String id) {
    PurchaseVO vo=new PurchaseVO();
    try {
        String sql="select * from purchase where order_id=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, id);
        ResultSet rs=ps.executeQuery();
        if(rs.next()) {
            vo.setOrder_id("order_id");
            vo.setName(rs.getString("name"));
            vo.setAddress(rs.getString("address"));
            vo.setEmail(rs.getString("email"));
            vo.setPayType(rs.getString("payType"));
            vo.setStatus(rs.getString("status"));
            vo.setTel(rs.getString("tel"));
            vo.setPdate(rs.getString("pdate"));
        }
    }catch(Exception e) { System.out.println("구매정보 읽기:" + e.toString()); }
    return vo;
}
```

[Java Resources]-[src]-[purchase]-[PurchaseServlet.java]

```
@WebServlet(value= {"/purchase/list", "/purchase/read", "/purchase/insert"})
public class PurchaseServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        PurchaseDAO dao=new PurchaseDAO();

        switch(request.getServletPath()) {
            case "/purchase/list":
                ...
                SqlVO sqlVO=new SqlVO();
                sqlVO.setKey(key);
                sqlVO.setWord(word);
                sqlVO.setPage(page);
                sqlVO.setPerPage(perPage);
                sqlVO.setOrder(order);
                sqlVO.setDesc(desc);
                jObject=dao.list(sqlVO);
                out.println(jObject);
                break;
            case "/purchase/read":
                String id=request.getParameter("id");
                PurchaseVO vo=dao.read(id);
                JSONObject jObject=new JSONObject();
                jObject.put("order_id", vo.getOrder_id());
                jObject.put("name", vo.getName());
                jObject.put("address", vo.getAddress());
                jObject.put("tel", vo.getTel());
                jObject.put("email", vo.getEmail());
                jObject.put("pdate", vo.getPdate());

                if(vo.getPayType().equals("0")){
                    jObject.put("payType", "무통장");
                }else {
                    jObject.put("payType", "카드");
                }
                if(vo.getStatus().equals("0")){
                    jObject.put("status", "처리중");
                }else {
                    jObject.put("status", "완료");
                }
                out.println(jObject);
                break;
        }
    }
}
```

- 주문 상품 정보 프로그램

[Java Resources]-[src]-[order]-[OrderVO.java]

```
public class OrderVO extends ProductVO{
    private String order_id;
    private String prod_id;
    private int price;
    private int quantity;
    // ...setter()/ getter()/ toString();
}
```

[Java Resources]-[src]-[order]-[OrderDAO.java]

```
public class OrderDAO {
    public ArrayList<OrderVO> list(String id) {
        ArrayList<OrderVO> list=new ArrayList<OrderVO>();
        try {
            String sql="select o.*, prod_name, company from orders o,product p where o.order_id=? and o.prod_id=p.prod_id";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, id);
            ResultSet rs=ps.executeQuery();

            while(rs.next()) {
                OrderVO vo=new OrderVO();
                vo.setProd_id(rs.getString("prod_id"));
                vo.setProd_name(rs.getString("prod_name"));
                vo.setPrice(rs.getInt("price"));
                vo.setQuantity(rs.getInt("quantity"));
                vo.setCompany(rs.getString("company"));
                list.add(vo);
            }
        }catch(Exception e) {
            System.out.println("구매목록:" + e.toString());
        }
        return list;
    }
}
```

[Java Resources]-[src]-[order]-[OrderServlet.java]

```
@WebServlet(value= {"/order/list", "/order/insert"})
public class OrderServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        OrderDAO dao=new OrderDAO();

        switch(request.getServletPath()) {
            case "/order/list":
                String id=request.getParameter("id");
                ArrayList<OrderVO> list=dao.list(id);
                JSONArray jArray=new JSONArray();

                for(OrderVO vo:list) {
                    JSONObject obj=new JSONObject();
                    obj.put("prod_id", vo.getProd_id());
                    obj.put("prod_name", vo.getProd_name());
                    obj.put("company", vo.getCompany());
                    obj.put("price", vo.getPrice());
                    obj.put("quantity", vo.getQuantity());
                    obj.put("sum", vo.getQuantity()*vo.getPrice());
                    jArray.add(obj);
                }
                out.println(jArray);
                break;
        }
    }
}
```

```

<style>
    #tblInfo, #tblOrder{width:800px; margin:auto; overflow:hidden; margin-bottom:10px; margin-top:10px;}
</style>
<div id="divInfo">
    <table id="tblInfo" style="margin-top:30px;"></table>
    <script id="tempInfo" type="text/x-handlebars-template">
        <tr>
            <td class="title" width=50>이름</td>
            <td width=100>{{name}}</td>
            <td class="title" width=50>전화</td>
            <td width=200>{{tel}}</td>
            <td class="title" width=50>이메일</td>
            <td width=150>{{email}}</td>
            <td class="title" width=50>구매일</td>
            <td width=150>{{pdate}}</td>
        </tr>
        <tr>
            <td class="title" width=50>주소</td>
            <td id="address" colspan=3>{{address}}</td>
            <td class="title" width=50>결제</td>
            <td id="payType" width=100>{{payType}}</td>
            <td class="title" width=50>상태</td>
            <td id="status" width=100>{{status}}</td>
        </tr>
    </script>
    <table id="tblOrder"></table>
    <script id="tempOrder" type="text/x-handlebars-template">
        <tr class="title">
            <td width=100>상품코드</td>
            <td width=300>상품명</td>
            <td width=100>제조사</td>
            <td width=100>가격</td>
            <td width=100>수량</td>
            <td width=100>합계</td>
        </tr>
        {{#each .}}
        <tr class="row">
            <td>{{prod_id}}</td>
            <td>{{prod_name}}</td>
            <td>{{company}}</td>
            <td>{{price}}</td>
            <td>{{quantity}}</td>
            <td>{{sum}}</td>
        </tr>
        {{/each}}
    </script>
</div>
<script>
    $("#tblPurchase").on("click", ".row button", function(){
        var row=$(this).parent().parent();
        var id=row.find(".id").html();
        $.ajax({
            type:"get",
            url:"read",
            data:{"id":id},
            dataType:"json",
            success:function(data){
                var template = Handlebars.compile($("#tempInfo").html());
                $("#tblInfo").html(template(data));
            }
        });
        $.ajax({
            type:"get",
            url:"/order/list",
            data:{"id":id},
            dataType:"json",
            success:function(data){
                var template = Handlebars.compile($("#tempOrder").html());
                $("#tblOrder").html(template(data));
            }
        });
    });
</script>

```

- 장바구니 프로그램

장 바 구 니

<input type="checkbox"/>	코드	상품명	단가	수량	합계	삭제
<input type="checkbox"/>	P101	아모라 아쿠아텍스 이지클린 침대	2200000	1 <input type="button" value="수정"/>	2200000	<input type="button" value="삭제"/>
<input type="checkbox"/>	P102	모던 라운지 낮은 퀸 침대	2100000	2 <input type="button" value="수정"/>	4200000	<input type="button" value="삭제"/>

[Java Resources]-[src]-[product]-[CartServlet.java]

```
@WebServlet(value= {"/cart/list", "/cart/insert", "/cart/update", "/cart/delete"})
public class CartServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        HttpSession session=request.getSession();
        ArrayList<OrderVO> listCart=(ArrayList<OrderVO>)session.getAttribute("listCart");

        ProductDAO dao=new ProductDAO();
        ProductVO productVO=dao.read(request.getParameter("id"));
        cartVO=new OrderVO();
        cartVO.setProd_id(id);
        cartVO.setProd_name(productVO.getProd_name());
        cartVO.setPrice(productVO.getPrice());
        cartVO.setQuantity(1);
        cartVO.setSum(cartVO.getPrice() * cartVO.getQuantity());

        switch(request.getServletPath()){
            case "/cart/insert":
                boolean isFind=false; //카트에 상품 존재 여부 체크
                if(listCart == null) {
                    listCart=new ArrayList<OrderVO>();
                } else {
                    for(OrderVO oldVO:listCart) {
                        if(oldVO.getProd_id().equals(cartVO.getProd_id())) {
                            oldVO.setQuantity(oldVO.getQuantity() + 1);
                            oldVO.setSum(oldVO.getPrice() * oldVO.getQuantity());
                            isFind = true;
                        }
                    }
                }
                if(!isFind) listCart.add(cartVO);
                session.setAttribute("listCart", listCart);
                break;
            case "/cart/update":
                String quantity=request.getParameter("quantity");
                for(OrderVO oldVO:listCart) {
                    if(oldVO.getProd_id().equals(oldVO.getProd_id())) {
                        oldVO.setQuantity(Integer.parseInt(quantity));
                        oldVO.setSum(oldVO.getPrice() * oldVO.getQuantity());
                    }
                }
                session.setAttribute("listCart", listCart);
                break;
            case "/cart/delete":
                for(int i=0; i< listCart.size(); i++) {
                    OrderVO oldVO=listCart.get(i);
                    if(oldVO.getProd_id().equals(cartVO.getProd_id())){
                        listCart.remove(i);
                        break;
                    }
                }
                session.setAttribute("listCart", listCart);
                break;
        }
    }
}
```

```
public ProductVO read(String prod_id) {
    ProductVO vo=new ProductVO();
    try {
        String sql="select * from product where prod_id=?";
        PreparedStatement ps=
            Database.CON.prepareStatement(sql);
        ps.setString(1, prod_id);
        ResultSet rs=ps.executeQuery();
        if(rs.next()) {
            vo.setProd_id(rs.getString("prod_id"));
            vo.setProd_name(rs.getString("prod_name"));
            vo.setPrice(rs.getInt("price1"));
        }
    } catch (Exception e) {
        System.out.println("상품정보:" + e.toString());
    }
    return vo;
}
```

[webContent]-[product]-[list.jsp]

```
//각행의 장바구니 버튼을 클릭한 경우
$("#tbl").on("click", ".row .btnCart", function(){
    var row=$(this).parent().parent();
    var id=row.find(".id").html();
    if(!confirm(id + "상품을 장바구니에 담겠습니까?")) return;
    $.ajax({
        type:"post",
        url:"/shop/cart/insert",
        data:{"id":id},
        success:function(){
            alert("장바구니에 " + id + " 상품이 담겼습니다.");
        }
    });
});
```

[webContent]-[cart]-[list.jsp]

```
<html>
<head>
    <title>쇼핑몰</title>
    <script src="http://code.jquery.com/jquery-3.1.1.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/handlebars.js/3.0.1/handlebars.js"></script>
    <link rel="stylesheet" href="../home.css"/>
</head>
<body>
<div id="divPage">
    <div id="divMenu"><jsp:include page="../menu.jsp"/></div>
    <div id="divCart">
        <div id="divHeader"><h2>장 바 구 니</h2></div>
        <c:if test="${listCart.size() > 0}">
            <table id="tbl">
                <tr class="title">
                    <td width=20><input type="checkbox" id="chkAll"></td>
                    <td width=100>코드</td>
                    <td width=330>상품명</td><td width=100>단가</td>
                    <td width=100>수량</td>
                    <td width=100>합계</td>
                    <td width=50>삭제</td>
                </tr>
                <c:forEach items="${listCart}" var="vo">
                    <tr class="row">
                        <td><input type="checkbox" class="chk"></td>
                        <td class="id">${vo.prod_id}</td>
                        <td class="name">${vo.prod_name}</td>
                        <td class="price">${vo.price}</td>
                        <td>
                            <input type="text" value="${vo.quantity}" class="quantity" size=1>
                            <button class="btnUpdate">수정</button>
                        </td>
                        <td class="sum">${vo.sum}</td>
                        <td><button class="btnDelete">삭제</button>
                    </tr>
                </c:forEach>
            </table>
            <div id="pagination">
                <button id="btnAll">전체상품주문</button><button id="btnSel">선택상품주문</button>
            </div>
        </c:if>
        <!-- 장바구니가 비어있는 경우----->
        <c:if test="${listCart==null || listCart.size()==0}">
            <table id="tbl">
                <tr class="title"><td class="title">장바구니가 비어있습니다.</td></tr>
            </table>
        </c:if>
    </div>
</div>
</body>
</html>
```

```

<script>
//모두 선택버튼을 클릭한 경우
$("#chkAll").on("click", function(){
    if($(this).prop("checked")){
        $("#tbl .row .chk").each(function(){
            $(this).prop("checked", true)
        });
    }else{
        $("#tbl .row .chk").each(function(){
            $(this).prop("checked", false)
        });
    }
});

//각 행의 체크 버튼을 클릭한 경우 전체선택인지를 체크
$("#tbl").on("click", ".row .chk", function(){
    var isChkAll=true;
    $("#tbl .row .chk").each(function(){
        if(!$(this).is(":checked")) isChkAll=false;
    });
    if(isChkAll){
        $("#tbl #chkAll").prop("checked", true);
    }else{
        $("#tbl #chkAll").prop("checked", false);
    }
});

//장바구니 수량 수정버튼을 클릭한 경우
$("#tbl").on("click", ".row .btnUpdate", function(){
    var row=$(this).parent().parent();
    var id=$(row).find(".id").html();
    var quantity=$(row).find(".quantity").val();
    var price=$(row).find(".price").html();
    if(!confirm(id + " 상품의 수량을 " + quantity + "개로 수정하시겠습니까?")) return;

    $.ajax({
        type:"post",
        url:"/cart/update",
        data:{"id":id, "quantity":quantity},
        success:function(){
            alert("수정되었습니다.");
            location.href="list.jsp";
        }
    });
});

//장바구니 상품 삭제버튼을 클릭한 경우
$("#tbl").on("click", ".row .btnDelete", function(){
    var row=$(this).parent().parent();
    var id=$(row).find(".id").html();
    if(!confirm(id + " 상품을 장바구니에서 삭제하시겠습니까?")) return;

    $.ajax({
        type:"post",
        url:"/shop/cart/delete",
        data:{"id":id},
        success:function(){
            alert("삭제 되었습니다.");
            location.href="list.jsp";
        }
    });
});
</script>

```


- 주문정보 프로그램 작성

주문 정보

상품코드	상품명	상품가격	상품수량	합계
P101	아모라 아쿠아텍스 이지클린 침대	2200000	1	2200000
P102	모던 라운지 낮은 퀸 침대	2100000	2	4200000
합계				6400000

주문자 성명	<input type="text" value="홍길동"/>
배송지 주소	<input type="text" value="인천 서구 경서동 등문 굿모닝힐"/>
이메일 주소	<input type="text" value="hong@daum.net"/>
전화번호	<input type="text" value="010-0210-1220"/>
결제방법	<input checked="" type="radio"/> 무통장 <input type="radio"/> 카드

주문하기

[webContent]-[cart]-[list.jsp]

```
<style>
#tblPurchase, #tblOrder{width:800px; margin:auto; overflow:hidden;margin-bottom:10px;}
</style>
<div id="divPage">
  <div id="divOrderInfo">
    <div id="divHeader"><h2>주문 정보</h2></div>
    <table id="tblOrder"></table>
    <script id="tempOrder" type="text/x-handlebars-template">
      <tr class="title">
        <td width=100>상품코드</td>
        <td width=400>상품명</td>
        <td width=100>상품가격</td>
        <td width=100>상품수량</td>
        <td width=100>합계</td>
      </tr>
      {{#each .}}
      <tr class="row">
        <td class="id">{{id}}</td>
        <td class="name">{{name}}</td>
        <td class="price">{{price}}</td>
        <td class="quantity">{{quantity}}</td>
        <td class="sum">{{sum}}</td>
      </tr>
      {{/each}}
      <tr class="title"><td colspan=4>합계</td><td id="totSum" class="sum"></td></tr>
    </script>
    <form name="frm">
      <table id="tblPurchase">
        <tr><td class="title" width=100>주문자 성명</td><td><input type="text" id="name" size=10></td></tr>
        <tr><td class="title" width=100>배송지 주소</td><td><input type="text" id="address" size=80></td></tr>
        <tr><td class="title" width=100>이메일 주소</td><td><input type="text" id="email" size=40></td></tr>
        <tr><td class="title" width=100>전화번호</td><td><input type="text" id="tel" size=40></td></tr>
        <tr>
          <td class="title" width=100>결제방법</td>
          <td>
            <input type="radio" id="payType" checked value="0">무통장
            <input type="radio" id="payType" value="1">카드
          </td>
        </tr>
      </table>
      <div id="pagination"><button id="btnOrder">주문하기</button></div>
    </form>
  </div>
</div>
```

```

<script>
...
$("#divOrderInfo").hide();

//전체상품 주문버튼을 클릭한 경우
$("#btnAll").on("click", function(){ showOrder($("#tbl .row")); });

//선택상품 주문버튼을 클릭한 경우
$("#btnSel").on("click", function(){
    var array=[];
    if($("#tbl .row .chk:checked").length > 0){
        $("#tbl .row .chk:checked").each(function(){
            var row=$(this).parent().parent();
            array.push(row);
        });
        showOrder(array);
    }else{ alert("선택한 상품이 없습니다."); }
});

//주문정보 화면 보기
function showOrder(row){
    var totSum=0, array=[];
    $(row).each(function(){
        var id=$(this).find(".id").html();
        var name=$(this).find(".name").html();
        var price=$(this).find(".price").html();
        var quantity=$(this).find(".quantity").val();
        sum=$(this).find(".sum").html();
        totSum += parseInt(sum);
        var data={"id":id,"name":name,"price":price,"quantity":quantity,"sum":sum};
        array.push(data);
    });

    var template = Handlebars.compile($("#tempOrder").html());
    $("#tblOrder").html(template(array));
    $("#totSum").html(totSum);
    $("#divCart").hide();
    $("#divOrderInfo").show();
}
</script>

```

• 주문하기 프로그램

```

public String insert(PurchaseVO vo) {
    String order_id="";
    try {
        String sql="insert into purchase(order_id, name, address, email, tel, paytype, pdate)
            values('R'||seq_purchase.nextval, ?, ?, ?, ?, sysdate)";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, vo.getName());
        ps.setString(2, vo.getAddress());
        ps.setString(3, vo.getEmail());
        ps.setString(4, vo.getTel());
        ps.setString(5, vo.getPayType());
        ps.execute();

        //현재 주문 ID 구하기
        sql="select 'R'||seq_purchase.currval order_id from dual";
        ps=Database.CON.prepareStatement(sql);
        ResultSet rs=ps.executeQuery();
        if(rs.next()) {
            order_id=rs.getString("order_id");
        }
    } catch (Exception e) { System.out.println("구매정보입력:" + e.toString()); }
    return order_id;
}

```

[Java Resources]-[src]-[purchase]-[PurchaseServlet.java]

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    request.setCharacterEncoding("UTF-8");
    PrintWriter out=response.getWriter();
    PurchaseVO vo=new PurchaseVO();
    ...
    PurchaseDAO dao=new PurchaseDAO();
    switch(request.getServletPath()) {
    case "/purchase/insert":
        String order_id=dao.insert(vo);
        JSONObject jObject=new JSONObject();
        jObject.put("order_id", order_id);
        out.print(jObject);
        break;
    }
}
```

[Java Resources]-[src]-[purchase]-[OrderDAO.java]

```
public void insert(OrderVO vo) {
    try {
        String sql="insert into orders(order_id,prod_id,price,quantity) values(?,?,?,?)";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ...
        ps.execute();
    }catch(Exception e) { System.out.println("주문상품 입력:" + e.toString()); }
}
```

[Java Resources]-[src]-[purchase]-[OrderServlet.java]

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    request.setCharacterEncoding("UTF-8");
    OrderVO vo=new OrderVO();
    ...
    OrderDAO dao=new OrderDAO();
    switch(request.getServletPath()) {
    case "/order/insert":
        dao.insert(vo);
        break;
    }
}
```

[webContent]-[cart]-[list.jsp]

```
$("#btnOrder").on("click", function(){
    if(!confirm("선택한 상품들을 주문하시겠습니까?")) return;
    var name=$(frm.name).val(); var address=$(frm.address).val();
    var email=$(frm.email).val(); var tel=$(frm.tel).val();
    var payType=$('#input[name="payType"]:checked').val();
    $.ajax({
        type:"post",
        url:"/purchase/insert",
        data:{"name":name,"address":address,"email":email,"tel":tel,"payType":payType},
        dataType:"json",
        success:function(data){
            var order_id=data.order_id;
            $("#tblOrder .row").each(function(){
                var prod_id=$(this).find(".id").html();
                var price=$(this).find(".price").html();
                var quantity=$(this).find(".quantity").html();
                $.ajax({
                    type:"post",
                    url:"/order/insert",
                    data:{"order_id":order_id,"prod_id":prod_id,"price":price,"quantity":quantity},
                    success:function(){ }
                });
            });
            alert("주문이 완료 되었습니다.");
            location.href="/shop/purchase/list.jsp";
        }
    });
});
```

- 상품 등록 프로그램

상 품 등 록

상품코드	P113	제조원/수입원	가구코리아	판매가격	250000																		
업체코드				일반가격	250000																		
상품명	킹침대(진주 매트포함)																						
상품이미지	<div>150 x 120</div> <div>업 체 목 록</div> <table border="1"> <thead> <tr> <th>업체코드</th> <th>업체명</th> <th>전화번호</th> </tr> </thead> <tbody> <tr> <td>M101</td> <td>에몬스 가구</td> <td>02-522-6898</td> </tr> <tr> <td>M102</td> <td>벨라지오 퍼니처</td> <td>02-897-6690</td> </tr> <tr> <td>M103</td> <td>까사미아</td> <td>032-502-9599</td> </tr> <tr> <td>M104</td> <td>레이디 가구</td> <td>02-602-9876</td> </tr> <tr> <td>M105</td> <td>가구 나라</td> <td>032-503-9021</td> </tr> </tbody> </table> <div> <div>◀ ▶</div> <div>[1/1]</div> </div>					업체코드	업체명	전화번호	M101	에몬스 가구	02-522-6898	M102	벨라지오 퍼니처	02-897-6690	M103	까사미아	032-502-9599	M104	레이디 가구	02-602-9876	M105	가구 나라	032-503-9021
업체코드	업체명	전화번호																					
M101	에몬스 가구	02-522-6898																					
M102	벨라지오 퍼니처	02-897-6690																					
M103	까사미아	032-502-9599																					
M104	레이디 가구	02-602-9876																					
M105	가구 나라	032-503-9021																					
상품설명																							

저장 취소

[Java Resources]-[src]-[product]-[ProductDAO.java]

//상품코드 마지막값 출력

```
public JSONObject getID() {
    JSONObject jObject=new JSONObject();
    try {
        String sql="select max(prod_id) id from product";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ResultSet rs=ps.executeQuery();
        if(rs.next()) {
            String lastID=(rs.getString("id")).substring(1);
            int newID = Integer.parseInt(lastID) + 1;
            String id="P" + newID;
            jObject.put("id", id);
        }
    }catch(Exception e) {
        System.out.println("상품코드 생성:" + e.toString());
    }
    return jObject;
}
```

//상품 등록

```
public void insert(ProductVO vo) {
    try {
        String sql="insert into product(prod_id,prod_name,company,mall_id,price1,price2,detail,image) values(?,?,?,?,?,?,?,?)";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, vo.getProd_id());
        ps.setString(2, vo.getProd_name());
        ps.setString(3, vo.getCompany());
        ps.setString(4, vo.getMall_id());
        ps.setInt(5, vo.getPrice1());
        ps.setInt(6, vo.getPrice2());
        ps.setString(7, vo.getDetail());
        ps.setString(8, vo.getImage());
        ps.execute();
    }catch(Exception e) {
        System.out.println("상품 등록:" + e.toString());
    }
}
```

[Java Resources]-[src]-[product]-[ProductServlet.java]

```
@WebServlet(value= {"/product/list", "/product/id", "/product/insert"})
public class ProductServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        ProductDAO dao=new ProductDAO();

        switch(request.getServletPath()) {
            case "/product/id":
                out.println(dao.getID());
                break;
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException{
        request.setCharacterEncoding("UTF-8");
        ProductDAO dao=new ProductDAO();

        //이미지파일 업로드
        String uploadPath="c:" + File.separator + "shop" + File.separator + "product" + File.separator;
        File mdPath=new File(uploadPath);
        if(!mdPath.exists()) mdPath.mkdir();

        MultipartRequest multi = new MultipartRequest(request,
                                                    uploadPath,
                                                    1024*1024*10,
                                                    "UTF-8",
                                                    new DefaultFileRenamePolicy());

        ProductVO vo=new ProductVO();
        String image=multi.getFilesystemName("image");
        String prod_id=multi.getParameter("prod_id");

        vo.setProd_id(prod_id);
        vo.setProd_name(multi.getParameter("prod_name"));
        vo.setCompany(multi.getParameter("company"));
        vo.setMall_id(multi.getParameter("mall_id"));
        vo.setPrice1(Integer.parseInt(multi.getParameter("price1")));
        vo.setPrice2(Integer.parseInt(multi.getParameter("price2")));
        vo.setDetail(multi.getParameter("detail"));

        switch(request.getServletPath()) {
            case "/product/insert":
                if(image != null) {
                    //Product ID로 이미지파일명 변경
                    File file=new File(uploadPath + image);
                    String newName = prod_id + image.substring(image.indexOf("."));
                    File newFile=new File(uploadPath + newName);
                    file.renameTo(newFile);
                    vo.setImage(newName);
                }

                dao.insert(vo);
                response.sendRedirect("list.jsp");
                break;
        }
    }
}
```

- c:/shop 폴더를 새로 생성한다.

[servers]-[sever.xml]

```
<host>
...
<Context docBase="c:/shop" path="/img" reloadable="true"/>
</host>
```

```

<div id="divPage">
  <div id="divMenu"><jsp:include page="../menu.jsp"/></div>
  <div id="divHeader"><h2>상 품 등 록 </h2></div>
  <form name="frm" action="insert" method="post" enctype="multipart/form-data">
    <table id="tbl">
      <tr>
        <td class="title" width=100>상품코드</td>
        <td width=100><input type="text" name="prod_id" size=10></td>
        <td class="title" width=100>제조원/수입원</td>
        <td width=200><input type="text" name="company" size=27 ></td>
        <td class="title" width=100>판매가격</td>
        <td width=100><input type="text" name="price1" size=10></td>
      </tr>
      <tr>
        <td class="title">업체코드</td>
        <td colspan=3><input type="text" name="mall_id" size=10><input type="text" name="mall_name" size=47></td>
        <td class="title">일반가격</td><td><input type="text" name="price2" size=10 ></td>
      </tr>
      <tr>
        <td class="title">상품명</td><td colspan=5><input type="text" name="prod_name" size=99></td>
      </tr>
      <tr>
        <td class="title">상품이미지</td>
        <td colspan=5>
          
          <input type="file" name="image" accept="image/*" style="visibility:hidden:">
        </td>
      </tr>
      <tr>
        <td class="title">상품설명</td>
        <td colspan=5><textarea rows="10" cols="100" name="detail"></textarea></td>
      </tr>
    </table>
    <div id="pagination">
      <input type="submit" value="저장"> <input type="reset" value="취소">
    </div>
  </form>
</div>
...
<script>
  //저장버튼을 클릭한 경우
  $(frm).submit(function(e){
    e.preventDefault();
    if(!confirm("새로운 상품을 등록하시겠습니까?")) return;
    frm.submit();
  });

  //이미지 미리보기
  $("#image").on("click", function(){ $(frm.image).click(); });
  $(frm.image).on("change", function(e){
    var reader = new FileReader();
    reader.onload=function(e){ $("#image").attr("src", e.target.result); }
    reader.readAsDataURL(this.files[0]);
  });

  //새로운 상품코드 출력
  $.ajax({
    type:"get",
    url:"id",
    dataType:"json",
    success:function(data){
      $(frm.prod_id).val(data.id);
    }
  })

  //업체목록 불러오기
  $(frm.mall_id).click(function(){
    window.open('/shop/mall/listAll.jsp', 'mall', 'width=400,height=300,top=200,left=900');
  });
</script>

```

```

<style>
    #tbl{width:350px; margin:auto;}
    #pagination, h2{width:350px; margin:auto; text-align:center; margin-top:10px; margin-bottom:10px;}
    table {border-collapse:collapse;}
    td{border:1px solid black; height:30px; font-size:12px; padding:0px 5px 0px 5px;}
    table .title{background:#EAEAEA; text-align:center;}
    table .row:hover{background:gray; color:white;}
</style>
<div id="divAll">
    <h2>업 체 목 록 </h2>
    <input id="selPerPage" type="hidden" value=5>
    <table id="tbl"></table>
    <script id="temp" type="text/x-handlebars-template">
        <tr class="title"><td width=100>업체코드</td><td width=100>업체명</td><td width=150>전화번호</td></tr>
        {{#each array}}
        <tr class="row">
            <td class="mall_id">{{mall_id}}</td>
            <td class="mall_name">{{mall_name}}</td>
            <td>{{tel}}</td>
        </tr>
        {{/each}}
    </script>
    <div id="pagination">
        <button id="btnPre">◀</button><button id="btnNext">▶</button>
        [<span id="curPage"></span><span id="totPage"></span>]
    </div>
</div>
</body>
<script>
    var url="list";
    $("#tbl").on("click", ".row", function(){
        var mall_id=$(this).find(".mall_id").html();
        var mall_name=$(this).find(".mall_name").html();
        $(opener.frm.mall_id).val(mall_id);
        $(opener.frm.mall_name).val(mall_name);
        window.close();
    });
</script>
<script src="../home.js"></script>

```

- 상품 정보 읽기, 수정 삭제 프로그램

상 품 정 보

상품코드	P101	제조원/수입원	가구나라	판매가격	2200000																		
업체코드	M101	에몬스 가구		일반가격	2300000																		
상품명	아모라 아쿠아텍스 이지클린 침대			판매상태	<input checked="" type="checkbox"/> 판매중지																		
상품이미지	<div style="display: flex; align-items: center;">  <div style="margin-left: 20px;"> <h3>업 체 목 록</h3> <table border="1"> <thead> <tr> <th>업체코드</th> <th>업체명</th> <th>전화번호</th> </tr> </thead> <tbody> <tr> <td>M101</td> <td>에몬스 가구</td> <td>02-522-6898</td> </tr> <tr> <td>M102</td> <td>벨라지오 퍼니처</td> <td>02-897-6690</td> </tr> <tr> <td>M103</td> <td>까사미아</td> <td>032-502-9599</td> </tr> <tr> <td>M104</td> <td>레이디 가구</td> <td>02-602-9876</td> </tr> <tr> <td>M105</td> <td>가구 나라</td> <td>032-503-9021</td> </tr> </tbody> </table> </div> </div>					업체코드	업체명	전화번호	M101	에몬스 가구	02-522-6898	M102	벨라지오 퍼니처	02-897-6690	M103	까사미아	032-502-9599	M104	레이디 가구	02-602-9876	M105	가구 나라	032-503-9021
업체코드	업체명	전화번호																					
M101	에몬스 가구	02-522-6898																					
M102	벨라지오 퍼니처	02-897-6690																					
M103	까사미아	032-502-9599																					
M104	레이디 가구	02-602-9876																					
M105	가구 나라	032-503-9021																					
상품설명	<div style="display: flex;"> <div style="flex: 1;"> <p>크기 : 싱글침대 재질 : 원목 색상 : 나무색</p> </div> <div style="flex: 1; border: 1px solid gray; padding: 5px;"> <p>[1/1]</p> </div> </div>																						

저장 삭제 취소

```

public ProductVO read(String id) {
    ProductVO vo=new ProductVO();
    try {
        String sql="select p.*, mall_name from product p left join mall m on p.mall_id= m.mall_id where prod_id=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, id);
        ResultSet rs=ps.executeQuery();
        if(rs.next()) {
            vo.setProd_id(rs.getString("prod_id"));
            vo.setProd_name(rs.getString("prod_name"));
            vo.setMall_id(rs.getString("mall_id"));
            vo.setMall_name(rs.getString("mall_name"));
            vo.setCompany(rs.getString("company"));
            vo.setImage(rs.getString("image"));
            vo.setPrice1(rs.getInt("price1"));
            vo.setPrice2(rs.getInt("price2"));
            vo.setDetail(rs.getString("detail"));
            vo.setProd_del(rs.getString("prod_del"));
        }
    }catch(Exception e) {
        System.out.println("상품정보 읽기:" + e.toString());
    }
    return vo;
}

public int delete(String id) {
    int count=-1;
    try {
        String sql="{call del_product(?, ?)}";
        CallableStatement cs=Database.CON.prepareCall(sql);
        cs.setString(1, id);
        cs.registerOutParameter(2, oracle.jdbc.OracleTypes.INTEGER);
        cs.execute();
        count=cs.getInt(2);
    }catch(Exception e) { System.out.println("상품삭제:" + e.toString()); }
    return count;
}

public void update(ProductVO vo) {
    try {
        String sql="update product set prod_name=?,company=?,mall_id=?,price1=?,price2=?,detail=?,image=?,prod_del=? ";
        sql += "where prod_id=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, vo.getProd_name());
        ps.setString(2, vo.getCompany());
        ps.setString(3, vo.getMall_id());
        ps.setInt(4, vo.getPrice1());
        ps.setInt(5, vo.getPrice2());
        ps.setString(6, vo.getDetail());
        ps.setString(7, vo.getImage());
        ps.setString(8, vo.getProd_del());
        ps.setString(9, vo.getProd_id());
        ps.execute();
    }catch(Exception e) {
        System.out.println("상품 수정:" + e.toString());
    }
}
}

```

```

variable count number; exec del_product('P101', :count); print count;

```

```

create or replace PROCEDURE DEL_PRODUCT(
    i_prod_id in product.prod_id%TYPE,
    o_count out int
)
AS
BEGIN
    select count(*) into o_count from orders where prod_id=i_prod_id;
    IF(o_count=0)THEN
        delete from product where prod_id=i_prod_id;
    END IF;
END DEL_PRODUCT;

```

```

@WebServlet(value= {"/product/list", "/product/id", "/product/insert", "/product/read", "/product/update", "/product/delete"})
public class ProductServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        ProductDAO dao=new ProductDAO();

        switch(request.getServletPath()) {
            case "/product/read":
                request.setAttribute("vo", dao.read(request.getParameter("id")));
                RequestDispatcher dis=request.getRequestDispatcher("read.jsp");
                dis.forward(request, response);
                break;
            case "/product/delete":
                ProductVO oldVO = dao.read(request.getParameter("prod_id"));
                int count=dao.delete(request.getParameter("prod_id"));
                if(count==0 && oldVO.getImage()!=null) {
                    if(oldVO.getImage() != null) {
                        System.gc();
                        String uploadPath="c:" + File.separator + "shop" + File.separator + "product" + File.separator;
                        File file=new File(uploadPath + oldVO.getImage());
                        file.delete();
                    }
                }
                JSONObject=new JSONObject();
                JSONObject.put("count", count);
                out.println(JSONObject);
                break;
        }
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException{
        ...
        ProductVO vo=new ProductVO();
        String image=multi.getFilesystemName("image");
        String prod_id=multi.getParameter("prod_id");

        vo.setProd_id(prod_id);
        vo.setProd_name(multi.getParameter("prod_name"));
        vo.setCompany(multi.getParameter("company"));
        vo.setMall_id(multi.getParameter("mall_id"));
        vo.setPrice1(Integer.parseInt(multi.getParameter("price1")));
        vo.setPrice2(Integer.parseInt(multi.getParameter("price2")));
        vo.setDetail(multi.getParameter("detail"));
        vo.setProd_del(multi.getParameter("prod_del"));

        switch(request.getServletPath()) {
            case "/product/insert":
                vo.setImage(newName);
                dao.insert(vo);
                response.sendRedirect("list.jsp");
                break;
            case "/product/update":
                ProductVO oldVO=dao.read(prod_id);
                if(image != null) {
                    if(oldVO.getImage() != null) {
                        System.gc();
                        File oldImage=new File(uploadPath + oldVO.getImage());
                        oldImage.delete();
                    }
                    vo.setImage(image);
                } else {
                    vo.setImage(oldVO.getImage());
                }
                dao.update(vo);
                response.sendRedirect("list.jsp");
                break;
        }
    }
}

```

```

<div id="divHeader"><h2>상 품 정 보</h2></div>
<form name="frm" action="update" method="post" enctype="multipart/form-data">
  <table id="tbl">
    ...
    <td><input type="checkbox" name="prod_del" value=1 <c:out value="{vo.prod_del=='1'?checked:''}"/>> 판매중지</td>
  </tr>
  <tr>
    <td class="title">상품이미지</td>
    <td colspan=5>
      <c:if test="{vo.image==null}"></c:if>
      <c:if test="{vo.image!=null}"></c:if>
      <input type="file" name="image" accept="image/*" style="visibility:hidden;">
    </td>
  </tr>
  <tr>
    <td class="title">상품설명</td>
    <td colspan=5><textarea rows="10" cols="100" name="detail">{vo.detail}</textarea></td>
  </tr>
</table>
<div id="pagination">
  <input type="submit" value="저장"><input type="button" value="삭제" id="btnDelete"> <input type="reset" value="취소">
</div>
</form>
...
<script>
  var prod_id="{vo.prod_id}";
  //저장버튼을 클릭한 경우
  $(frm).submit(function(e){
    e.preventDefault();
    if(!confirm("상품 내용을 수정하시겠습니까?")) return;
    frm.submit();
  });
  //판매중지 체크
  $(frm.prod_del).on("click",function(){
    if($(frm.prod_del).is(":checked")){ $(frm.prod_del).val(1); alert("판매를 중지합니다."); }
    else{ $(frm.prod_del).val(0); alert("판매중지를 해제합니다."); }
  });
  //이미지 미리보기
  $("#image").on("click", function(){ $(frm.image).click(); });
  $(frm.image).on("change", function(e){
    var reader = new FileReader();
    reader.onload=function(e){ $("#image").attr("src", e.target.result); }
    reader.readAsDataURL(this.files[0]);
  });

  //삭제버튼을 클릭한 경우
  $("#btnDelete").on("click", function(){
    if(!confirm(prod_id + " 상품을 삭제하시겠습니까?")) return;
    $.ajax({
      type:"get",
      url:"delete",
      data:{"prod_id":prod_id},
      dataType:"json",
      success:function(data){
        if(data.count==0){
          alert("상품이 삭제되었습니다.");
          location.href="/shop/product/list.jsp";
        }else{
          alert("구매내역에 상품이 있어 삭제할 수 없습니다.");
        }
      }
    });
  });

  //업체목록 불러오기
  $(frm.mall_id).click(function(){
    window.open('/shop/mall/listAll.jsp','mall',{width=400,height=300,top=200,left=900});
  });
</script>

```

상품코드 ▾

4개 ▾

검색

검색수: 12건

상품코드 ▾

오름차순 ▾



P101
아모라 아쿠아텍스 이지클린 침대
2200000원
판매중지



P102
모던 라운지 낮은 퀸 침대
2100000원
CART담기



P103
도로시 아쿠아텍스 기능성 퀸 침대
3000000원
CART담기



P104
모리엘 호텔형 침대
1500000원
CART담기

◀ ▶ [1/3]

[webContent]-[index.jsp]

```
<style>
    .box{width:200px;float:left;text-align:center; margin:auto;}
    .price1, .prod_id, .prod_name{font-size:12px; width:200px; text-overflow:ellipsis; white-space:nowrap; overflow:hidden;}
    .cart0{font-size:12px; width:200px; cursor:pointer; color:green; font-weight:bold;}
    .cart1{font-size:12px; width:200px; color:red; font-weight:bold;}
    .image img{width:150px; height:120px; margin:auto; margin-bottom:5px;}
</style>
</head>
<body>
<div id="divPage">
    ...
    <div id="tbl"></div>
    <script id="temp" type="text/x-handlebars-template">
        {{#each array}}
        <div class="box">
            <div class="image"><img src={{printImage image}}></div>
            <div class="prod_id">{{prod_id}}</div>
            <div class="prod_name">{{prod_name}}</div>
            <div class="price1">{{price1}}원</div>
            <div class="cart">{{prod_del}}>{{printDelete prod_del}}</div>
        </div>
        {{/each}}
    </script>
    ...
</div>
</body>
<script>
    var url="/shop/product/list";

    //이미지가 없을경우 출력형태 변경
    Handlebars.registerHelper("printImage", function(image){
        var src;
        if(!image) src="http://placeholder.it/300x200"; else src="/img/product/" + image;
        return src;
    });
    //상품이 판매중지된경우 출력형태 변경
    Handlebars.registerHelper("printDelete", function(del){
        var src;
        if(del==1) src="판매중지"; else src="CART담기";
        return src;
    });
    //각행의 장바구니 버튼을 클릭한 경우
    $("#tbl").on("click", ".box .cart0", function(){
        var box=$(this).parent();
        var id=box.find(".prod_id").html();
        if(!confirm(id + "상품을 장바구니에 담겠습니까?")) return;
        $.ajax({
            type:"post",
            url:"/shop/cart/insert",
            data:{"id":id},
            success:function(){ alert("장바구니에 " + id + " 상품이 담겼습니다."); }
        });
    });
</script>
<script src="home.js"></script>
```

- 상품정보 입력 (첨부 이미지추가)

상 품 등 록

상품코드	P113	제조원/수입원	가우코리아	판매가격	250000
업체코드				일반가격	250000
상품명	킹침대(진주 매트포함)				
대표이미지			상품설명을 입력하세요... <div></div>		
<div>상품이미지</div>					

저장

취소

상품이미지 테이블 생성

```
create table images(
  image varchar(200),
  prod_id char(4),
  primary key(image),
  foreign key(prod_id) references product(prod_id)
);
```

[Java Resources]-[src]-[product]-[ProductDAO.java]

```
//상품이미지 입력
public void insertImage(ArrayList<String> images, String prod_id) {
  try {
    for(String image:images) {
      String sql="insert into images(image, prod_id) values(?,?)";
      PreparedStatement ps=Database.CON.prepareStatement(sql);
      ps.setString(1, image);
      ps.setString(2, prod_id);
      ps.execute();
    }
  } catch(Exception e) {
    System.out.println("상품이미지입력:" + e.toString());
  }
}
```

[Java Resources]-[src]-[product]-[ProductServlet.java]

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
  request.setCharacterEncoding("UTF-8");
  ProductDAO dao=new ProductDAO();
  ProductVO vo=new ProductVO();

  //업로드한 상품이미지 저장변수
  ArrayList<String> images=new ArrayList<String>();

  //이미지파일 업로드 Path
  String uploadPath="c:" + File.separator + "shop" + File.separator + "product" + File.separator;

  //업로드 패스가 존재하지 않을 경우 패스를 생성한다.
  File mdPath=new File(uploadPath);
  if(!mdPath.exists()) mdPath.mkdir();
}
```

```

//1.업로드 데이터를 보관하는 FileItem의 Factory 설정
DiskFileItemFactory fatory = new DiskFileItemFactory();

//2.업로드 요청을 처리하는 ServletFileUpload 생성
ServletFileUpload upload = new ServletFileUpload(fatory);

//3.파일명 한글 처리
upload.setHeaderEncoding("UTF-8");

try {
    //4.업로드 요청시 파싱해서 FileItem 목록 구함
    List<FileItem> items = upload.parseRequest(request);

    for(FileItem item:items) {
        //폼필드인 경우
        if(item.isFormField()) {
            switch(item.getFieldName()) {
                case "prod_id":
                    vo.setProd_id(item.getString("UTF-8"));
                    break;
                case "prod_name":
                    vo.setProd_name(item.getString("UTF-8"));
                    break;
                case "company":
                    vo.setCompany(item.getString("UTF-8"));
                    break;
                case "mall_id":
                    vo.setMall_id(item.getString("UTF-8"));
                    break;
                case "pricel":
                    vo.setPricel(Integer.parseInt(item.getString("UTF-8")));
                    break;
                case "price2":
                    vo.setPrice2(Integer.parseInt(item.getString("UTF-8")));
                    break;
                case "detail":
                    vo.setDetail(item.getString("UTF-8"));
                    break;
                case "prod_del":
                    vo.setProd_del(item.getString("UTF-8"));
                    break;
            }
            //업로드 파일인 경우
        }else{
            String fileName=item.getName();

            if(!fileName.equals("")) {
                if(item.getFieldName().equals("image")) {
                    vo.setImage(fileName);
                }else if(item.getFieldName().equals("images")){
                    images.add(fileName);
                }

                File file = new File(uploadPath + fileName);
                item.write(file);
            }
        }
    }
}catch(Exception e) {
    System.out.println(e.toString());
}

switch(request.getServletPath()) {
case "/product/insert":
    dao.insert(vo);
    dao.insertImage(images, vo.getProd_id());
    response.sendRedirect("list.jsp");
    break;
}
}

```

```

<style>
    #listFile img {width:150px;margin-left:20px;}
    input[type='file'] {visibility:hidden; height:0px;}
</style>

...
<div id="divPage">
    <div id="divMenu"><jsp:include page="../menu.jsp"/></div>
    <div id="divHeader"><h2>상 품 등 록 </h2></div>
    <form name="frm" action="insert" method="post" enctype="multipart/form-data">
        <table id="tbl">
            <tr>
                <td width=100 class="title">상품코드</td>
                <td width=100><input type="text" name="prod_id" size=10></td>
                <td width=100 class="title">제조원/수입원</td>
                <td width=200><input type="text" name="company" size=20 value="가구코리아"></td>
                <td class="title" width=100>판매가격</td>
                <td width=100><input type="text" name="price1" size=5 value="250000"></td>
            </tr>
            <tr>
                <td class="title">업체코드</td>
                <td colspan=3><input type="text" name="mall_id" size=10><input type="text" name="mall_name" size=47></td>
                <td class="title">일반가격</td><td><input type="text" name="price2" size=5 value="250000"></td>
            </tr>
            <tr>
                <td class="title">상품명</td>
                <td colspan=5><input type="text" name="prod_name" size=100 value="킹침대"></td>
            </tr>
            <tr>
                <td class="title">대표이미지</td>
                <td colspan=2>
                    
                    <input type="file" name="image" accept="image/*">
                </td>
                <td colspan=3><textarea rows="12" cols="65" name="detail" placeholder="상품설명을 입력하세요..."></textarea></td>
            </tr>
            <tr>
                <td class="title"><input type="button" id="btnImage" value="상품이미지"></td>
                <td colspan=5 style="padding:5px;height:150px;">
                    <input type="file" name="images" accept="image/*" multiple>
                    <div id="listFile"></div>
                </td>
            </tr>
        </table>
        <div id="pagination"><input type="submit" value="저장"> <input type="reset" value="취소"></div>
    </form>
</div>
<script>
    $(frm).submit(function(e){ //저장버튼을 클릭한 경우
        e.preventDefault();
        if(!confirm("새로운 상품을 등록하시겠습니까?")) return;
        if($(frm.image)[0].files[0]==null){ alert("대표이미지를 등록하세요!");
        }else{ frm.submit(); }
    });

    $("#image").on("click", function(){ $(frm.image).click(); }); //대표이미지 미리보기
    $(frm.image).on("change", function(){
        var file=$(frm.image)[0].files[0];
        $("#image").attr("src",URL.createObjectURL(file));
    });

    $("#btnImage").on("click", function(){ $(frm.images).click(); }); //상품이미지 미리보기
    $(frm.images).change(function(){
        var target = $(frm.images);
        var html = "";
        $.each(target[0].files, function(index, file){
            html += "";
        });
        $('#listFile').html(html);
    });
</script>

```

- 상품정보 읽기, 수정, 삭제 (첨부 이미지추가)

상 품 정 보

상품코드	P112	제조원/수입원	삼익가구	판매가격	1500000
업체코드	M102	벨라지오 퍼니처		일반가격	1670000
상품명	엘린 4인용 카우치 가죽소파			판매상태	<input type="checkbox"/> 판매중지
대표이미지				상품설명을 입력하세요...	
상품이미지	<div>   </div>				

[Java Resources]-[src]-[product]-[ProductDAO.java]

```

public ArrayList<String> readImage(String prod_id) {
    ArrayList<String> list=new ArrayList<String>();
    try {
        String sql="select * from images where prod_id=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, prod_id);
        ResultSet rs=ps.executeQuery();
        while(rs.next()) {
            list.add(rs.getString("image"));
        }
    } catch (Exception e) { System.out.println("상품이미지읽기:" + e.toString()); }
    return list;
}

public void deleteImage(String prod_id) {
    try {
        String sql="delete from images where prod_id=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, prod_id);
        ps.execute();
    } catch (Exception e) { System.out.println("상품이미지삭제:" + e.toString()); }
}

public void delete(String id) {
    try {
        String sql="delete from product where prod_id=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, id);
        ps.execute();
    } catch (Exception e) { System.out.println("상품삭제:" + e.toString()); }
}

public int chkOrder(String prod_id) {
    int count=-1;
    try {
        String sql="select count(*) cnt from orders where prod_id=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, prod_id);
        ResultSet rs=ps.executeQuery();
        if(rs.next()) { count=rs.getInt("cnt"); }
    } catch (Exception e) { System.out.println("주문상품 유무 체크:" + e.toString()); }
    return count;
}
  
```

```

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out=response.getWriter();
    ProductDAO dao=new ProductDAO();

    switch(request.getServletPath()) {
    case "/product/read":
        request.setAttribute("vo", dao.read(request.getParameter("id")));
        request.setAttribute("images", dao.readImage(request.getParameter("id")));
        RequestDispatcher dis=request.getRequestDispatcher("read.jsp");
        dis.forward(request, response);
        break;
    case "/product/delete":
        String uploadPath="c:" + File.separator + "shop" + File.separator + "product" + File.separator;
        String prod_id=request.getParameter("prod_id");
        int count=dao.chkOrder(prod_id);
        if(count==0) {
            ProductVO oldVO = dao.read(prod_id);
            System.gc();
            //대표이미지 삭제
            File file=new File(uploadPath + oldVO.getImage());
            file.delete();

            //상품추가이미지 삭제
            ArrayList<String> images=dao.readImage(prod_id);
            for(String image:images) {
                file=new File(uploadPath + image);
                file.delete();
            }
            dao.deleteImage(request.getParameter("prod_id"));
            dao.delete(prod_id);
        }
        JSONObject=new JSONObject();
        JSONObject.put("count", count);
        out.println(JSONObject);
        break;
    }
}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    ...
    switch(request.getServletPath()) {
    case "/product/update":
        ProductVO oldVO=dao.read(vo.getProd_id());
        //대표이미지 수정
        if(vo.getImage()==null) {
            vo.setImage(oldVO.getImage());
        } else {
            System.gc();
            File file=new File(uploadPath + oldVO.getImage());
            file.delete();
        }
        //상품추가이미지 수정
        if(images.size()>0) {
            ArrayList<String> oldImages=dao.readImage(vo.getProd_id());
            for(String oldImage:oldImages) {
                System.gc();
                File file=new File(uploadPath + oldImage);
                file.delete();
            }
            dao.deleteImage(vo.getProd_id());
            dao.insertImage(images, vo.getProd_id());
        }

        dao.update(vo);
        response.sendRedirect("list.jsp");
        break;
    }
}

```



```

<form name="frm" action="update" method="post" enctype="multipart/form-data">
  <table id="tbl">
    ...
    <tr>
      <td class="title">상품명</td>
      <td colspan=3><input type="text" name="prod_name" size=63 value="{vo.prod_name}"></td>
      <td class="title">판매상태</td>
      <td>
        <input type="checkbox" name="prod_del" value="{vo.prod_del}" <c:out value="{vo.prod_del}='1'?checked:''"/>
        판매중지
      </td>
    </tr>
    <tr>
      <td class="title">대표이미지</td>
      <td colspan=2>
        
        <input type="file" name="image" accept="image/*">
      </td>
      <td colspan=3>
        <textarea rows="12" cols="65" name="detail" placeholder="상품설명을 입력하세요...">{vo.detail}</textarea>
      </td>
    </tr>
    <tr>
      <td colspan=5>
        <input type="button" id="btnImage" value="상품이미지">
        <input type="file" name="images" accept="image/*" multiple>
        <div id="listFile">
          <c:forEach items="{images}" var="image">
            
          </c:forEach>
        </div>
      </td>
    </tr>
  </table>
  <div id="pagination">
    <input type="submit" value="저장"><input type="button" value="삭제" id="btnDelete"> <input type="reset" value="취소">
  </div>
</form>
...
<script>
  var prod_id="{vo.prod_id}";
  //저장버튼을 클릭한 경우
  $(frm).submit(function(e){
    e.preventDefault();
    if(!confirm("상품 내용을 수정하시겠습니까?")) return;
    frm.submit();
  });
  //판매중지 체크
  $(frm.prod_del).on("click",function(){
    if($(frm.prod_del).is(":checked")){ $(frm.prod_del).val(1); alert("판매를 중지합니다."); }
    else{ $(frm.prod_del).val(0); alert("판매중지를 해제합니다."); }
  });
  //삭제버튼을 클릭한 경우
  $("#btnDelete").on("click", function(){
    if(!confirm(prod_id + " 상품을 삭제하시겠습니까?")) return;
    $.ajax({
      type:"get",
      url:"delete",
      data:{prod_id:prod_id},
      dataType:"json",
      success:function(data){
        if(data.count==0){
          alert("상품이 삭제되었습니다.");
          location.href="/shop/product/list.jsp";
        }else{ alert("구매내역에 상품이 있어 삭제할수 없습니다."); }
      }
    });
  });
</script>

```

- 로그인 프로그램

로그인	
아이디	<input type="text" value="user02"/>
비밀번호	<input type="password" value="****"/>
<input type="button" value="로그인"/>	

사용자 테이블 생성

```
create table users(
    id varchar(20),
    pass varchar(20),
    name varchar(20),
    address varchar(200),
    tel varchar(50),
    email varchar(50),
    primary key(id)
);
```

사용자 Sample 데이터 입력

```
insert into users(id, pass, name, address, tel, email)
values('user01', 'pass', '홍길동', '인천 서구 경서동 동문굿모닝힐', '010-1010-2020', 'user01@shop.com');
insert into users(id, pass, name, address, tel, email)
values('user02', 'pass', '심청이', '인천 부평구 부평5동 104번지', '010-2020-1010', 'user02@shop.com');
```

[Java Resources]-[src]-[user]-[UserVO.java]

```
public class UserVO {
    private String id;
    private String pass;
    private String name;
    private String address;
    private String tel;
    private String email;
    // ... setter()/getter() to String()
}
```

[Java Resources]-[src]-[user]-[UserDAO.java]

```
public class UserDAO {
    public UserVO login(String id) {
        UserVO vo=new UserVO();
        try {
            String sql="select * from users where id=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, id);
            ResultSet rs=ps.executeQuery();
            if(rs.next()) {
                vo.setId(rs.getString("id"));
                vo.setPass(rs.getString("pass"));
                vo.setName(rs.getString("name"));
                vo.setAddress(rs.getString("address"));
                vo.setTel(rs.getString("tel"));
                vo.setEmail(rs.getString("email"));
            }
        } catch (Exception e) {
            System.out.println("로그인체크:" + e.toString());
        }
        return vo;
    }
}
```

[Java Resources]-[src]-[user]-[UserServlet.java]

```
@WebServlet(value= {"/user/login", "/user/logout"})
public class UserServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        switch(request.getServletPath()) {
            case "/user/logout":
                HttpSession session=request.getSession();
                session.invalidate();
                response.sendRedirect("login.jsp");
            }
        }
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        PrintWriter out=response.getWriter();
        String id=request.getParameter("id");
        String pass=request.getParameter("pass");
        UserDAO udao=new UserDAO();

        switch(request.getServletPath()) {
            case "/user/login":
                UserVO vo=udao.login(id);
                int check = 0; //id가 없는 경우
                if(vo.getId()!=null) {
                    if(vo.getPass().equals(pass)) {
                        check=2; //pass가 일치하는 경우
                        HttpSession session=request.getSession();
                        session.setAttribute("vo", vo);
                    }else {
                        check=1; //pass가 일치하지 않는 경우
                    }
                }
                JSONObject jObject=new JSONObject();
                jObject.put("check", check);
                out.println(jObject);
            }
        }
    }
}
```

[webContent]-[menu.jsp]

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
...
<c:if test="${vo.id!=null}">
    <div class="item"><a href="#" id="btnChat">채팅창</a></div>
    <div class="item" style="float:right;width:200px;">
        <span style="font-size:15px">[${vo.name}]</span>
        <a href="/shop/user/logout">로그아웃</a>
    </div>
</c:if>
<c:if test="${vo.id==null}">
    <div class="item" style="float:right;width:200px;">
        <a href="/shop/user/login.jsp">로그인</a>
    </div>
</c:if>
```

[webContent]-[cart]-[list.jsp]

```
<table id="tblPurchase">
    <tr>
        <td class="title" width=100>주문자 성명</td><td><input type="text" name="name" size=10 value="${vo.name}"></td>
    </tr>
    <tr>
        <td class="title" width=100>배송지 주소</td><td><input type="text" name="address" size=80 value="${vo.address}"></td>
    </tr>
    <tr>
        <td class="title" width=100>이메일 주소</td><td><input type="text" name="email" size=40 value="${vo.email}"></td>
    </tr>
    <tr>
        <td class="title" width=100>전 화 번 호</td><td><input type="text" name="tel" size=40 value="${vo.tel}"></td>
    </tr>
    ...
</table>
```

```

<html>
<head>
  <title>학사관리시스템</title>
  <link rel="stylesheet" href="../../home.css">
  <style>
    #divContent{height:300px;}
    #divContent table{margin:auto; margin-top:100px;}
    input[type="text"] {height:15px; font-size:10px;}
    input[type="password"] {height:15px; font-size:10px;}
  </style>
  <script src="http://code.jquery.com/jquery-1.9.1.js"></script>
</head>
<body>
<div id="divPage">
  <div id="divMenu"><jsp:include page="../../menu.jsp"/></div>
  <div id="divContent">
    <form name="frm" action="login" method="post">
      <table>
        <tr>
          <td class="title" width=300 colspan=2><h1>로그인</h1></td>
        </tr>
        <tr>
          <td class="title" width=150>아이디</td>
          <td><input type="text" name="id" size=15></td>
        </tr>
        <tr>
          <td class="title" width=150>비밀번호</td>
          <td><input type="password" name="pass" size=15></td>
        </tr>
        <tr>
          <td colspan=2 style="text-align:center;height:50px;">
            <input type="submit" value="로그인"></td>
          </tr>
        </table>
      </form>
    </div>
  </div>
  <script>
    $(frm).submit(function(e){
      var id=$(frm.id).val();
      var pass=$(frm.pass).val();
      e.preventDefault();
      $.ajax({
        type:"post",
        url:"login",
        data:{"id":id,"pass":pass},
        dataType:"json",
        success:function(data){
          if(data.check==0){
            alert("아이디가 존재하지 않습니다.");
            $(frm.id).focus();
          }else if(data.check==1){
            alert("아이디와 비밀번호가 일치하지 않습니다.");
            $(frm.pass).focus();
          }else{
            location.href="/shop/index.jsp";
          }
        }
      });
    });
  </script>
</html>

```