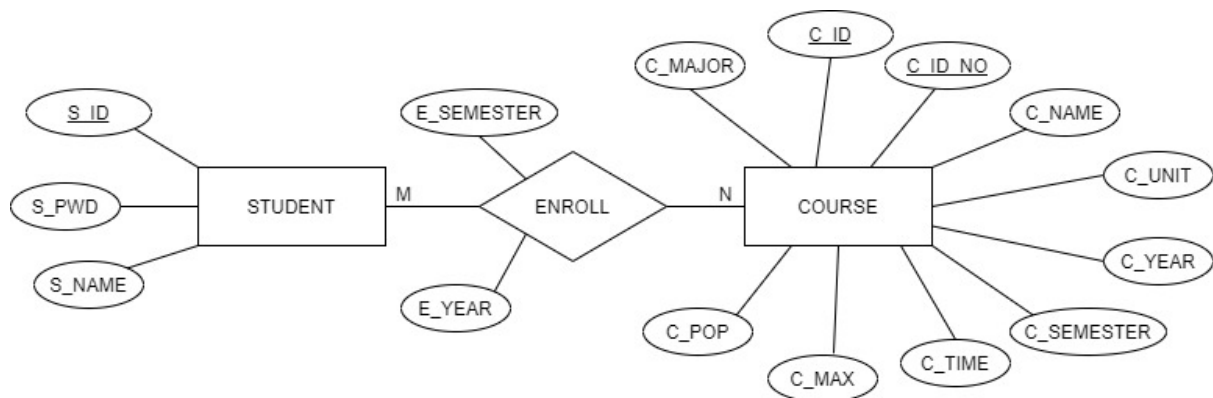


프로젝트 과제 보고서

연도	2021년	학기	1학기
수업명	데이터베이스 프로그래밍	분반	001분반
팀원	노세희(통계학과, 1816461), 이해원(통계학과, 1714121), 한하람(통계학과, 1710385)		

1. DB 설계도 및 테이블



<STUDENT> 테이블

변수	변수 설명	제약조건
S_ID	학생 학번	PRIMARY KEY
S_PWD	학생 비밀번호	
S_NAME	학생 이름	

<COURSE> 테이블

변수	변수 설명	제약조건
C_MAJOR	강의가 열리는 학과	
C_ID	강의 코드	PRIMARY KEY
C_ID_NO	강의 분반	PRIMARY KEY

C_NAME	강의 이름	
C_UNIT	강의 학점	
C_YEAR	강의가 열리는 년도	
C_SEMESTER	강의가 열리는 학기	
C_TIME	강의 시간 (교시)	
C_MAX	강의 수강 가능 인원	
C_POP	강의 수강 신청 인원	

<ENROLL> 테이블

변수	변수 설명	제약조건
S_ID	학생 학번	PRIMARY KEY FOREIGN KEY(STUDENT)
C_ID	강의 코드	PRIMARY KEY FOREIGN KEY(COURSE)
C_ID_NO	강의 분반	FOREIGN KEY(COURSE)
E_YEAR	수강신청한 강의의 년도	
E_SEMESTER	수강신청한 강의의 학기	

2. SQL질의문과 JSP 소스 코드 설명

table.sql
<pre> CREATE TABLE STUDENT(S_ID varchar(10), S_PWD varchar(10), S_NAME varchar(30), PRIMARY KEY(S_ID)); INSERT INTO STUDENT VALUES('1714121', 'abcde', '이혜원'); </pre>

```

INSERT INTO STUDENT VALUES('1816461', 'abcde', '노세희');
INSERT INTO STUDENT VALUES('1710385', 'abcde', '한하람');
INSERT INTO STUDENT VALUES('1712345', 'abcde', '서인국');
INSERT INTO STUDENT VALUES('1612345', 'abcde', '박서준');
INSERT INTO STUDENT VALUES('1512345', 'abcde', '장기용');

CREATE TABLE COURSE(
  C_MAJOR varchar(20),
  C_ID varchar(10),
  C_ID_NO number,
  C_NAME varchar(50),
  C_UNIT number,
  C_YEAR number,
  C_SEMESTER number,
  C_TIME number,
  C_MAX number,
  C_POP number,
  PRIMARY KEY (C_ID,C_ID_NO)
);
INSERT INTO COURSE VALUES('컴퓨터공학과','C700',1,'데이터베이스 프로그래밍',3,2021,1,1,5,0);
INSERT INTO COURSE VALUES('컴퓨터공학과','C800', 2, '데이터베이스 프로그래밍',3,2021,2,1,5,0);
INSERT INTO COURSE VALUES('컴퓨터공학과','C800', 3, '데이터베이스 프로그래밍',3,2021,2,1,5,0);
INSERT INTO COURSE VALUES('컴퓨터공학과','C900', 3, '멀티미디어 개론',3,2021,2,2,5,0);
INSERT INTO COURSE VALUES('컴퓨터공학과','M100', 3, '선형대수',3,2021,2,3,5,0);
INSERT INTO COURSE VALUES('컴퓨터공학과','M200', 3, '그래픽 활용',3,2021,2,4,5,0);
INSERT INTO COURSE VALUES('컴퓨터공학과','M300', 3, '윈도우 프로그래밍',3,2021,2,5,5,0);
INSERT INTO COURSE VALUES('컴퓨터공학과','M400', 3, '멀티미디어 처리',3,2021,2,6,5,0);
INSERT INTO COURSE VALUES('컴퓨터공학과','M500', 3, '게임프로그래밍',3,2021,2,7,5,0);
INSERT INTO COURSE VALUES('컴퓨터공학과','M600', 3, '컴퓨터 그래픽스',3,2021,2,1,5,0);
INSERT INTO COURSE VALUES('통계학과','S100',1,'기초통계학',3,2021,2,1,5,0);
INSERT INTO COURSE VALUES('통계학과','S200',1,'수리통계학',3,2021,2,1,5,0);

CREATE TABLE ENROLL(
  C_ID varchar(10),
  C_ID_NO number,
  S_ID varchar(10),
  E_YEAR NUMBER,
  E_SEMESTER NUMBER,
  PRIMARY KEY(S_ID, C_ID),
  FOREIGN KEY(S_ID) REFERENCES STUDENT,
  FOREIGN KEY(C_ID,C_ID_NO) REFERENCES COURSE
);

```

DB에 Student, Course, Enroll 테이블 생성

Main.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"%>
<!DOCTYPE html>
<html>
    <head> <meta charset="EUC-KR">
        <title>데이터베이스를 활용한 수강신청 시스템입니다.</title>
    </head>
    <body>
        <%@include file="top.jsp"%>
        <table width="75%" align="center" height="100%">
            <% if (session_id != null) { %>
            <tr>
            <br>
                <td align="center"> <%=session_id%> 님 방문을 환영합니다.</td>
            </tr>
            <% } else { %>
            <tr>
                <td align="center">로그인한 후 사용하세요.</td>
            </tr>
            <%
            }
            %>
        </table>
    </body>
</html>

```

; 조건문 사용

메인 화면을 보여주며, top.jsp를 포함하고 있다.

로그인 이전에는 "로그인 후 사용하세요" 가 화면에 출력되고, 로그인 이후에는 로그인 한 session_id(학번)에 대하여 "session_id님 방문을 환영합니다." 가 화면에 출력된다.

Top.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%
String session_id = (String)session.getAttribute("user");
String log;

if (session_id == null)
    log = "<a href=login.jsp>로그인</a>";
else
    log = "<a href=logout.jsp>로그아웃</a>";
%>

<table width="75%" align="center" bgcolor="#FFFF99" border>
<tr>
<td align="center"><b><%=log%></b></td>

```

```

<td align="center"><b><a href="update.jsp">사용자 정보 수정</a></b></td>
<td align="center"><b><a href="insert.jsp">수강신청</a></b></td>
<td align="center"><b><a href="delete.jsp">수강신청 삭제</a></b></td>
<td align="center"><b><a href="select.jsp">수강신청 조회</a></b></td>
</tr>
</table>

```

; 조건문 사용

로그인한 session_id(학번)을 이용하여 세션을 관리하고 메뉴를 관리한다.

Login.jsp

```

<%@ page contentType="text/html; charset=UTF-8"%>
<!DOCTYPE html>
<html>
<head> <meta charset="UTF-8">
    <title>수강신청 시스템 로그인 </title>
</head>
<body>
<%@ include file="top.jsp"%>

<br>
    <table width="50%" align="center" bgcolor="#FFFF99" border>
        <tr><td><div align="center"> 아이디와 패스워드를 입력하세요</div>
        </table>

<br>
    <table width="50%" align="center" border>
        <form method="post" action="login_verify.jsp">
            <tr>
                <td><div align="center">아이디</div></td>
                <td><div align="center"> <input type="text" name="userID"> </div></td>
            </tr>
            <tr>
                <td><div align="center">패스워드</div></td>
                <td><div align="center"> <input type="password" name="userPassword"> </div></td>
            </tr>
            <tr>
                <td colspan=2><div align="center">
                    <INPUT TYPE="SUBMIT" NAME="Submit" VALUE="로그인">
                    <INPUT TYPE="RESET" VALUE="취소">
                </div></td>
            </tr>
        </table>
    </form>
</body>
</html>

```

메인 화면에 [로그인]을 클릭함으로써 보여지는 로그인에 필요한 화면을 구성한다.

아이디(학번)와 암호 입력 후 올바른지 여부는 login_verify.jsp를 통해 확인한다.
로그인에 성공하면 top.jsp에 의해 메뉴에 [로그인]이 [로그아웃]으로 변경된다.

Login_Verify.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page import="java.sql.*"%>
<%

    String dbdriver = "oracle.jdbc.driver.OracleDriver";
    Class.forName(dbdriver);

    String dburl = "jdbc:oracle:thin:@localhost:1521:xe";
    String user = "db1816461"; String passwd = "oracle";

    Connection myConn = null;
    Statement stmt = null;
    String mySQL = null;

    String userID=request.getParameter("userID");
    String userPassword=request.getParameter("userPassword");

    myConn=DriverManager.getConnection(dburl, user, passwd);
    stmt = myConn.createStatement();
    mySQL="select s_id from student where s_id='" + userID + "'and s_pwd='" + userPassword + "'";
    ResultSet myResultSet = stmt.executeQuery(mySQL);

    if (myResultSet.next()) {
        session.putValue("user",userID);
        response.sendRedirect("main.jsp");
    }

    else {

        %>
        <script>
        alert("사용자아이디 혹은 암호가 틀렸습니다");
        location.href="login.jsp";
        </script>
        <%
        }
        stmt.close();
        myConn.close();

    %>
```

; JDBC드라이버 로딩후 데이터베이스 연결, 전진과 읽기만 가능한 ResultSet 생성, 조건문 사용

입력된 아이디(학번)와 암호가 연결된 DB의 Student테이블에 존재하는지 질의문을 통해 확인 후, 존재하는 경우에는 기타 기능을 사용할 수 있도록 main.jsp로 이동한다.
아이디와(학번)와 암호가 테이블에 존재하지 않은 경우 "아이디 혹은 비밀번호가 틀렸습니다" 팝업과 함께 다시 로그인을 시도하도록 login.jsp로 이동한다.

Logout.jsp

```
<%@ page contentType="text/html; charset=UTF-8" %>
<% session.invalidate(); %>
<script>
    alert("로그아웃 되었습니다.");
    location.href="main.jsp";
</script>
```

로그인된 상태에서 메뉴에 [로그아웃]을 클릭하면 top.jsp에 의해 [로그아웃]에서 [로그인]으로 변경되고 세션이 해제되며, main.jsp로 돌아간다.

Update.jsp

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ page import="java.sql.*" %>
<html>
<head><title> 수강신청 사용자 정보 수정 </title> </head>
<body>
    <%@ include file="top.jsp" %>
    <%
    if (session_id==null)
        response.sendRedirect("login.jsp");
    %>
    <%
        String dbdriver = "oracle.jdbc.driver.OracleDriver";
        String dburl = "jdbc:oracle:thin:@localhost:1521:xe";
        String user = "db1816461"; String passwd = "oracle";

        Connection myConn = null;
        Statement stmt = null;
        ResultSet myResultSet = null;
        String mySQL = "";

        String userPassword=""; String userName="";

        try {

            Class.forName(dbdriver);
            myConn = DriverManager.getConnection (dburl, user, passwd);
```

```

        stmt = myConn.createStatement();
        mySQL = "select s_name, s_pwd from student where s_id=" + session_id + " ";
        myResultSet = stmt.executeQuery(mySQL);
    }

    catch(SQLException ex) {
        System.err.println("SQLException: " + ex.getMessage());
    }

    finally{
        if(myResultSet != null) {
            if(myResultSet.next()) {
                userName = myResultSet.getString("s_name");
                userPassword = myResultSet.getString("s_pwd");
            }
        }
    }

    %>

    <form method= "post" action= "update_verify.jsp">
    <br>
    <table align = "center" border id= "update_table">
        <tr>
            <td align= "center" id= "update_td">아이디</td>
            <td align= "center" colspan= "3"><input id= "update_id_in" type= "text"
name= "id" size= "50" style="text-align: center;" value= "<%=session_id%> " ReadOnly></td>
            </tr>
            <tr>
            <td align= "center" id= "update_td">이름</td>
            <td align= "center" colspan= "3"><input id= "update_name_in"
type= "text" name= "name" size= "50" style="text-align: center;" value= "<%=userName%> "></td>
            </tr>
            <tr>
            <td align= "center" id= "update_td">비밀번호</td>
            <td><input id= "update_pw_in" type= "password" name= "password"
size= "20" value= "<%=userPassword%> "></td>
            <td id= "update_td">확인</td>
            <td><input id= "update_pw_in" type= "password"
name= "passwordConfirm" size= "20" ></td>
            </tr>
            <tr>
            <td colspan= "5"><div align= "center">
            <input id= "update_btn" type= "SUBMIT" NAME= "Submit" size= "30"
style="text-align:center;" value= "수정 완료" ></div></td>
            </tr>
        </table>
    </form>

    <%

```



```

        stmt.close();
        myConn.close();
    %>

</body>
</html>

```

; JDBC드라이버 로딩후 데이터베이스 연결, 전진과 읽기만 가능한 ResultSet 생성, 조건문 사용
로그인된 사용자의 정보를 수정하는 화면이다.

로그인 정보의 질의문을 통해 사용자 정보인 이름, 학번을 조회하여 화면에 보여준다.

<비밀번호>와 <확인>에 수정할 암호를 입력하거나, <이름> 에 수정할 이름을 입력하여
[수정완료] 버튼을 클릭하면 update_verify.jsp에 의해 입력된 값으로 사용자 정보를 수정한다.

Update_verify.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="java.io.*"%>
<%@ page import="java.sql.*"%>

<html>
<head>
<title>수강신청 사용자 정보 수정</title>
</head>
<body>
<%

    request.setCharacterEncoding("UTF-8");

    Connection myConn = null;
    String dburl = "jdbc:oracle:thin:@localhost:1521:xe";
    String user="db1816461"; String passwd="oracle";
    String dbdriver = "oracle.jdbc.driver.OracleDriver";
    PreparedStatement pstmt = null;

    String formId = request.getParameter("id");
    String formName = request.getParameter("name");
    String formPass = request.getParameter("password");
    String confirmPass = request.getParameter("passwordConfirm");

    if(!formPass.equals(confirmPass)) {
        %> <script>
        alert("비밀번호를 다시 확인해주세요.");
        location.href="update.jsp";
        </script> <%
    }
    else {
        try{
            Class.forName(dbdriver);
            myConn = DriverManager.getConnection(dburl, user, passwd);
            String SQL = "UPDATE student SET s_pwd=?, s_name=? WHERE s_id=?";

```

```

        pstmt = myConn.prepareStatement(SQL);

        pstmt.setString(1, formPass);
        pstmt.setString(2, formName);
        pstmt.setString(3, formId);
        pstmt.executeUpdate();

        %> <script>
            alert("성공적으로 수정했습니다.");
            location.href="main.jsp";
        </script> <%

    }catch(SQLException ex){
        String sMessage="";
        if (ex.getErrorCode() == 20002)
            sMessage = "암호는 4자리 이상이어야합니다.";
        else if (ex.getErrorCode() == 20003)
            sMessage = "암호에 공란은 입력되지 않습니다.";
        else if (ex.getErrorCode()==20004)
            sMessage="암호를 입력하셔야 합니다.";
        else
            sMessage = "잠시 후 다시 시도하십시오.";
        out.println("<script>");
        out.println("alert('"+sMessage+"');");
        out.println("location.href='update.jsp';");
        out.println("</script>");
        out.flush();
    }finally{
        if(pstmt != null)
            try{pstmt.close();
            }catch(SQLException sqle){}
        if(myConn != null)
            try{myConn.close();
            }catch(SQLException sqle){}
    }
}

%>
</body>
</html>

```

; JDBC드라이버 로딩후 데이터베이스 연결, Connection 메소드인 prepareStatement사용, 조건문 사용

Update.jsp에서 입력된 값으로 사용자 정보를 수정한다.

BeforeUpdateStudent 트리거를 통해 예외처리를 수행하고, 예외가 없다면 "성공적으로 수정했습니다" 팝업과 함께 UPDATE 질의문을 통해 Student 테이블의 사용자 정보를 수정하고 main.jsp로 이동한다.

BeforeUpdateStudent.sql

```
CREATE OR REPLACE TRIGGER BeforeUpdateStudent BEFORE
UPDATE ON student
FOR EACH ROW
DECLARE
    underflow_length EXCEPTION;
    invalid_value EXCEPTION;
    not_null_test EXCEPTION;
    nLength NUMBER;
    nBlank NUMBER;
    oldpwd VARCHAR(10);
    newpwd VARCHAR(10);
BEGIN
    nLength := Length(:new.s_pwd);
    nBlank := INSTR(:new.s_pwd, ' ',1,1);
    oldpwd := :old.s_pwd;
    newpwd := :new.s_pwd;

    IF nLength < 4 THEN
        RAISE underflow_length;
    ELSIF nBlank != 0 THEN
        RAISE invalid_value;
    ELSIF oldpwd IS NULL THEN
        RAISE not_null_test;
    ELSIF newpwd IS NULL THEN
        RAISE not_null_test;
    END IF;

    EXCEPTION
        WHEN underflow_length THEN
            RAISE_APPLICATION_ERROR(-20002, '암호는 4자리 이상이어야 합니다');
        WHEN invalid_value THEN
            RAISE_APPLICATION_ERROR(-20003, '암호에 공란은 입력되지 않습니다. ');
        WHEN not_null_test THEN
            RAISE_APPLICATION_ERROR(-20004, '암호를 입력하셔야 합니다. ');

END;
/
```

; 트리거와, null값 처리 사용, 조건문 사용

사용자 정보 수정에서 발생할 수 있는 예외를 처리하는 트리거이다.

1. 암호의 길이가 4자리 미만인 경우

-> 에러번호는 -20002, 에러설명은 '암호는 4자리 이상이어야 합니다.' 팝업과 함께 오류 발생

2. 암호에 공란이 포함된 경우

-> 에러번호는 -20003, 에러설명은 '암호에 공란은 입력되지 않습니다.' 팝업과 함께 오류 발생

3. 암호가 입력되지 않은 경우

-> 에러번호는 -20004, 에러설명은 '암호를 입력하셔야 합니다.' 팝업과 함께 오류가 발생.

4. 암호가 정상적인 경우

-> 에러 없이 수정이 완료된다.

Insert.jsp

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="java.sql.*" %>
<html>
  <head>
    <title>수강신청</title>
  </head>
  <body>
    <%@ include file="top.jsp" %>
    <%
      if (session_id==null)
        response.sendRedirect("login.jsp");
    %>

    <table width="75%" align="center" border>
    <br>
    <tr>
      <th>학과전공</th>
      <th>수업번호</th>
      <th>분반</th>
      <th>수업명</th>
      <th>시간</th>
      <th>학점</th>
      <th>신청인원</th>
      <th>수강신청</th>
    </tr>

    <%

    request.setCharacterEncoding("UTF-8");

    String search_major = request.getParameter("search_major");
    if(search_major == null) {
      search_major = "";
    }
    String dbdriver = "oracle.jdbc.driver.OracleDriver";
    String dburl = "jdbc:oracle:thin:@localhost:1521:xe";
```

```

String user = "db1816461"; String passwd = "oracle";

Connection myConn = null;
Statement stmt = null;
ResultSet myResultSet = null;

String SQL1 = ""; String SQL2 = ""; String SQL3="";
String mySQL = "";
String studentCount = "";

try {
    Class.forName(dbdriver);
    myConn = DriverManager.getConnection (dburl, user, passwd);
    stmt = myConn.createStatement();
} catch(SQLException ex) {
    System.err.println("SQLException: " + ex.getMessage());
}

SQL1 = "{? = call Date2EnrollYear(SYSDATE)}";
CallableStatement cstmt = myConn.prepareCall(SQL1);
cstmt.registerOutParameter(1, java.sql.Types.INTEGER);
cstmt.execute();
int nYear = cstmt.getInt(1);

SQL2 = "{? = call Date2EnrollSemester(SYSDATE)}";
CallableStatement cstmt2 = myConn.prepareCall(SQL2);
cstmt2.registerOutParameter(1, java.sql.Types.INTEGER);
cstmt2.execute();
int nSemester = cstmt2.getInt(1);

%>

<p align = "center" class="title">
<%=nYear%>년도
<%=nSemester%>학기 수강신청
</p>
<div align = "center">
<form method= "post" width= "75%" align= "center" action= "insert.jsp">
학과전공: <input type= "text" name= "search_major" value= "<%=search_major %>" size= "10"/>
<input type= "SUBMIT" NAME= "submit" size= "30" style="text-align:center;" value= "검색" >
<br/><br/>
</form>
</div>

<%
if (search_major.equals(""))

```

```

        mySQL = "select c_major,c_id,c_id_no,c_name,c_unit,c_time,c_pop,c_max from course where c_year =" + nYear +
        "and c_semester = " + nSemester
        + " and (c_id,c_id_no) not in (select c_id,c_id_no from enroll where s_id=" + session_id + ") order by
        c_major,c_id,c_id_no";
        else if (search_major.equals("컴퓨터공학과"))
            mySQL = "select c_major,c_id,c_id_no,c_name,c_unit,c_time,c_pop,c_max from comscience where c_year =" +
            nYear + "and c_semester = " + nSemester
            + " and (c_id,c_id_no) not in (select c_id,c_id_no from enroll where s_id=" + session_id + ") order by
            c_major,c_id,c_id_no";
        else if (search_major.equals("통계학과"))
            mySQL = "select c_major,c_id,c_id_no,c_name,c_unit,c_time,c_pop,c_max from statistics where c_year =" + nYear +
            "and c_semester = " + nSemester
            + " and (c_id,c_id_no) not in (select c_id,c_id_no from enroll where s_id=" + session_id + ") order by
            c_major,c_id,c_id_no";
        else
        {
            %>
            <script>
            alert("검색 결과가 없습니다.");
            </script>
            <%
            mySQL = "select c_major,c_id,c_id_no,c_name,c_unit,c_time,c_pop,c_max from course where c_year =" + nYear +
            "and c_semester = " + nSemester
            + " and (c_id,c_id_no) not in (select c_id,c_id_no from enroll where s_id=" + session_id + ") order by
            c_major,c_id,c_id_no";
            }

        myResultSet = stmt.executeQuery(mySQL);
        if (myResultSet != null) {
            while (myResultSet.next()) {
                String c_major = myResultSet.getString("c_major");
                String c_id = myResultSet.getString("c_id");
                int c_id_no = myResultSet.getInt("c_id_no");
                String c_name = myResultSet.getString("c_name");
                int c_time = myResultSet.getInt("c_time");
                int c_unit = myResultSet.getInt("c_unit");
                int c_pop = myResultSet.getInt("c_pop");
                int c_max = myResultSet.getInt("c_max");

            %>

            <tr>
            <td align="center"><%= c_major %></td>
            <td align="center"><%= c_id %></td>
            <td align="center"><%= c_id_no %></td>
            <td align="center"><%= c_name %></td>
            <td align="center"><%= c_time %></td>
            <td align="center"><%= c_unit %></td>

```

```

        <td align="center"><%= c_pop %> <%= "/" %> <%= c_max %> </td>
        <td align="center"><a href="insert_verify.jsp?c_id=<%= c_id %> &c_id_no=<%= c_id_no %> ">신청</a></td>
    </tr>
<%
    }

}

stmt.close();
myConn.close();
%>
</table>
</body>
</html>

```

; JDBC드라이버 로딩후 데이터베이스 연결, 전진과 읽기만 가능한 ResultSet 생성, 조건문 사용, subquery를 이용한 중첩질의 사용, Connection 메소드인 prepareCall()사용하여 리턴 값이 있는 함수 처리

연결된 세션 DB의 Course와 Enroll 테이블을 이용하여 수강신청 가능한 수업을 화면에 출력한다. Date2EnrollYear 과 Date2EnrollSemester 함수를 통해 현재 연도와 학기에 존재하는 수강 수업들만 표시되도록 한다.

<학과전공 검색>에 원하는 학과전공의 수업만 검색하면, ComScience와 Statistics VIEW를 통해 검색된 학과전공의 수업만 출력한다. 검색어를 잘못 입력한 경우에는 "검색 결과가 없습니다" 팝업이 출력된다.

<학과전공, 수업번호, 분반, 수업명, 시간, 학점, 신청인원, 수강신청> 변수를 로그인된 session_id의 Enroll 테이블에는 없지만 Course 테이블에는 존재하는 수업들, 즉, 수강신청 하지 않은 수업들만 출력된다.

원하는 수업의 <수강신청>의 [신청] 버튼을 클릭하면 insert_verify.jsp를 통해 확인 후 신청된다.

Insert_verify.jsp

```

<%@ page contentType="text/html; charset=UTF-8" %>
<%@ page import = "java.sql.*" %>

<html>
<head><title> 수강신청 </title>
</head>

<%@ include file="top.jsp" %>
<body>
    <%
        String s_id = (String)session.getAttribute("user");
        String c_id = request.getParameter("c_id");
        int c_id_no = Integer.parseInt(request.getParameter("c_id_no"));
    %>

```

```

String dbdriver = "oracle.jdbc.driver.OracleDriver";
String dburl = "jdbc:oracle:thin:@localhost:1521:xe";
String user = "db1816461"; String passwd = "oracle";

Connection myConn = null;
String result= "";

try {
    Class.forName(dbdriver);
    myConn = DriverManager.getConnection(dburl,user,passwd);
} catch (SQLException ex) {
    System.err.println("SQLException : " + ex.getMessage());
}

CallableStatement cstmt = myConn.prepareCall("{call InsertEnroll(?,?,?,?)");
cstmt.setString(1,s_id);
cstmt.setString(2,c_id);
cstmt.setInt(3,c_id_no);
cstmt.registerOutParameter(4,java.sql.Types.VARCHAR);

CallableStatement cstmt2 = myConn.prepareCall("{call countstudent(?,?)");
cstmt2.setString(1,c_id);
cstmt2.setInt(2,c_id_no);

try{
    cstmt.execute();
    cstmt2.execute();
    result = cstmt.getString(4);

%>
<script>
    alert("<%=result%>");
    location.href="insert.jsp";
</script>
<%
    } catch (SQLException ex){
        System.err.println("SQLException : " + ex.getMessage());
    } finally{
        if(cstmt != null)
            try{
                myConn.commit();
                cstmt.close();
                myConn.close();
            } catch (SQLException ex) {
            }
        }
    }
}
%>

```



```
</body>
</html>
```

; JDBC드라이버 로딩후 데이터베이스 연결, Connection 메소드인 prepareCall()사용하여 OUT, 조건문 사용, 파라미터가 존재하는 프로시저와 IN 파라미터만 있는 프로시저 처리
insert.jsp에서 신청한 수업에 대하여 InsertEnroll 프로시저를 통해 수강신청을 한다.
신청이 완료 되었다면 countstudent 프로시저로 수업별 수강 신청 인원을 새로 계산하여 Course 테이블의 c_pop변수를 UPDATE한다.

view.sql

```
CREATE OR REPLACE VIEW ComScience(c_major,c_id,c_id_no,c_name,c_unit,c_time,c_year,c_semester,c_pop,c_max)
AS
SELECT c_major,c_id,c_id_no,c_name,c_unit,c_time,c_year,c_semester,c_pop,c_max
FROM course
WHERE c_major='컴퓨터공학과';

CREATE OR REPLACE VIEW Statistics(c_major,c_id,c_id_no,c_name,c_unit,c_time,c_year,c_semester,c_pop,c_max)
AS
SELECT c_major,c_id,c_id_no,c_name,c_unit,c_time,c_year,c_semester,c_pop,c_max
FROM course
WHERE c_major='통계학과';
```

insert.jsp에서 <학과전공 검색>을 할 경우 검색한 학과전공의 수업만 출력하기 위해 학과전공마다 VIEW를 생성한다. Course테이블의 c_major 변수 가 '컴퓨터공학과'인 ComScience VIEW와 c_major 변수가 '통계학과'인 Statistics VIEW를 생성하였다.

Date_function.sql

```
CREATE OR REPLACE FUNCTION Date2EnrollYear(dDate IN DATE)
RETURN NUMBER
IS
    nMonth CHAR(2);
    nYear NUMBER;
BEGIN
    SELECT TO_NUMBER(TO_CHAR(dDate, 'YYYY')),TO_CHAR(dDate, 'MM')
    INTO nYear, nMonth
    FROM DUAL;
    IF (nMonth = '11' OR nMonth = '12')THEN
        nYear := nYear+1 ;
    END IF;
    RETURN nYear;
END;
/
```

```

CREATE OR Replace PROCEDURE DeleteEnroll (sStudentId IN VARCHAR2,
                                           sCourseId IN VARCHAR2,
                                           nCourseIdNo IN NUMBER,
                                           result OUT VARCHAR2)
IS
BEGIN
    result := '';

    DBMS_OUTPUT.put_line('#');
    DBMS_OUTPUT.put_line(sStudentId || '님이 수업번호 ' || sCourseId || ', 분반 ' || TO_CHAR(nCourseIdNo) || '의
수강 취소를 요청하였습니다.');
```

```

    DELETE
    FROM enroll
    WHERE s_id = sStudentId and c_id = sCourseId and c_id_no = nCourseIdNo;

    COMMIT;
    result := '수강취소가 완료되었습니다.';

EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        result := SQLCODE;

END;
/
```

; 리턴 값이 있는 저장함수 사용, 조건문 사용

Date2EnrollYear 함수에 SYSDATE(현재 날짜)의 연도 부분을 입력 파라미터로 받아 수강신청에 해당하는 연도를 리턴한다.

Date2EnrollSemester 함수에 SYSDATE(현재 날짜)의 월 부분을 입력 파라미터로 받아 수강신청에 해당하는 학기를 리턴 한다. (11월 ~ 4월: 1학기, 5월 ~ 10월 : 2학기)

insertEnroll.sql

```

CREATE OR REPLACE PROCEDURE InsertEnroll (
    sStudentId IN VARCHAR2,
    sCourseId IN VARCHAR2,
    nCourseIdNo IN NUMBER,
    result OUT VARCHAR2)
IS
    too_many_sumCourseUnit EXCEPTION;
    duplicate_courses EXCEPTION;
    too_many_students EXCEPTION;
    duplicate_time EXCEPTION;
```

```

nYear NUMBER;
nSemester NUMBER;
nSumCourseUnit NUMBER;
nCourseUnit NUMBER;
nCnt NUMBER;
nTeachMax NUMBER;

BEGIN
    result := '';

    DBMS_OUTPUT.put_line('#');
    DBMS_OUTPUT.put_line(sStudentId || '님이 수업번호 ' || sCourseId || ', 분반 ' || TO_CHAR(nCourseIdNo) || '의 수강 등록을
    요청하였습니다.');
```

/* 년도, 학기 알아내기 */

```

nYear := Date2EnrollYear(SYSDATE);
nSemester := Date2EnrollSemester(SYSDATE);

/* 에러 처리 1 : 최대학점 초과여부 */
SELECT SUM(c.c_unit)
INTO nSumCourseUnit
FROM   course c, enroll e
WHERE  e.s_id = sStudentId and e.e_year = nYear and e.e_semester = nSemester
       and e.c_id = c.c_id and e.c_id_no = c.c_id_no;

SELECT c_unit
INTO nCourseUnit
FROM   course
WHERE  c_id = sCourseId and c_id_no = nCourseIdNo;

IF (nSumCourseUnit + nCourseUnit > 18)
THEN
    RAISE too_many_sumCourseUnit;
END IF;

/* 에러 처리 2 : 동일한 수업 신청 여부 */
SELECT COUNT(*)
INTO   nCnt
FROM   enroll
WHERE  s_id = sStudentId and c_id = sCourseId and e_year = nYear and e_semester = nSemester;

IF (nCnt > 0)
THEN
    RAISE duplicate_courses;
END IF;

```

```

/* 에러 처리 3 : 수강신청 인원 초과 여부 */
SELECT c_max
INTO nTeachMax
FROM course
WHERE c_year= nYear and c_semester = nSemester
      and c_id = sCourseId and c_id_no= nCourseIdNo;

SELECT COUNT(*)
INTO nCnt
FROM enroll
WHERE e_year = nYear and e_semester = nSemester
      and c_id = sCourseId and c_id_no = nCourseIdNo;

IF (nCnt >= nTeachMax)
THEN
    RAISE too_many_students;
END IF;

/* 에러 처리 4 : 신청한 수업들 시간 중복 여부 */
SELECT COUNT(*)
INTO nCnt
FROM
(SELECT c_time
 FROM course
 WHERE c_id = sCourseId and c_id_no = nCourseIdNo
 INTERSECT
 SELECT c.c_time
 FROM course c, enroll e
 WHERE e.s_id=sStudentId and e.c_id=c.c_id and e.c_id_no=c.c_id_no and e.e_year = nYear and e.e_semester =
nSemester
);
IF(nCnt > 0)
THEN
    RAISE duplicate_time;
END IF;

/* 수강 신청 등록 */
INSERT INTO enroll(S_ID,C_ID,C_ID_NO,E_YEAR,E_SEMESTER)
VALUES (sStudentId, sCourseId, nCourseIdNo, nYear, nSemester);

COMMIT;
result := '수강신청 등록이 완료되었습니다.';

EXCEPTION
WHEN too_many_sumCourseUnit THEN

```

```

    result := '최대학점을 초과하였습니다';
WHEN duplicate_courses THEN
    result := '이미 등록된 수업을 신청하였습니다';
WHEN too_many_students THEN
    result := '수강신청 인원이 초과되어 등록이 불가능합니다';
WHEN duplicate_time THEN
    result := '이미 등록된 수업 중 중복되는 시간이 존재합니다';
WHEN no_data_found THEN
    result := '이번 학기 수업이 아닙니다.';
WHEN OTHERS THEN
    ROLLBACK;
    result := SQLCODE;
END;
/

```

;사용자 지정예외를 사용한 저장프로시저, 조건문 사용, Oracle함수(sum,count)사용, join 사용
수강신청 등록 시 발생하는 예외를 처리하고, 예외가 발생하지 않으면 수강 신청을 등록하여 해당 수업을 Enroll 테이블에 INSERT하는 프로시저이다.

1. Course와 Enroll 테이블로부터 전체 수강신청 수업의 총 학점이 18학점(최대 학점)이 초과되는지 검사하고, 18학점을 초과하면 'too_many_sumCourseUnit' 예외를 발생시켜 "최대학점을 초과하였습니다" 팝업을 출력하고, 예외 처리한다.
2. Enroll 테이블로부터 선택된 수업이 이전에 수강 신청한 수업인지 검사하고, 이미 수강 신청되어있는 경우, 'duplicate_courses' 예외를 발생시켜 "이미 등록된 수업을 신청하였습니다" 팝업창 출력하고, 예외 처리한다.
3. Course와 Enroll 테이블로부터 선택된 수업에 대한 최대 인원이 초과되었는지 검사하고, 최대 인원이 초과된 수업의 경우, 'too_many_students' 예외를 발생시켜 "수강신청 인원이 초과되어 등록이 불가능합니다" 팝업을 출력하고, 예외 처리한다.
4. Course와 Enroll 테이블로부터 동일한 시간에 다른 수업이 이미 수강 신청 되었는지 검사하고, 이미 수강 신청되어 있는 경우, 'duplicate_time' 예외를 발생시켜 "이미 등록된 수업 중 중복되는 시간이 존재합니다" 팝업을 출력하고, 예외 처리한다.
5. SELECT 질의문이 아무런 데이터 행을 반환하지 못할 경우, "이번 학기 수업이 아닙니다" 팝업을 출력한다.
6. 아무런 예외가 발생하지 않은 경우, 선택된 수업에 대한 정보를 Enroll테이블에 INSERT 한다.

countstudent.sql

```

CREATE OR REPLACE PROCEDURE countstudent
(cid IN course.c_id%TYPE,
 cidno IN course.c_id_no%TYPE)
IS
ncount NUMBER;
CURSOR course_list(cid course.c_id%TYPE, cidno course.c_id_no%TYPE) IS

```

```

SELECT s_id
FROM enroll
WHERE c_id = cid and c_id_no=cidno ;
BEGIN
ncount :=0 ;
FOR courselst IN course_list(cid,cidno) LOOP
    ncount :=ncount+1;
END LOOP;

UPDATE course
SET c_pop = ncount
WHERE c_id = cid and c_id_no=cidno ;
END;
/

```

; %TYPE형과 파라미터가 있는 커서 사용, 반복문 사용

insert.jsp에서 Course 테이블의 모든 c_id와 c_id_no 변수를 입력 파라미터로 받아 수업의 신청인원 수(c_pop)를 UPDATE하는 프로시저이다. 입력 파라미터의 데이터 형의 경우 %TYPE을 사용하여 정의한다. FOR-LOOP 반복문을 통하여 수업마다 신청한 총 인원수를 계산한다.

Delete.jsp

```

<%@ page contentType= "text/html; charset=UTF-8" %>
<%@ page import= "java.sql.*" %>
<html>
<head>
<title>수강신청취소 페이지입니다</title>
</head>
<body>
<%@ include file= "top.jsp"%>
<%
    if (session_id==null)
        response.sendRedirect("login.jsp");
%>

<table width = "75%" class= "classtable" border= "1" align= "center" >
<tr>
    <th>수업번호</th>
    <th>분반</th>
    <th>수업명</th>
    <th>학점</th>
    <th>수강취소</th>
</tr>
<%
    String dbdriver = "oracle.jdbc.driver.OracleDriver";
    String dburl = "jdbc:oracle:thin:@localhost:1521:xe";

```

```

String user = "db1816461"; String passwd = "oracle";

Connection myConn = null;
Statement stmt = null;
ResultSet myResultSet = null;
String SQL1 = "";
String SQL2 = "";
String mySQL = "";

try {
    Class.forName(dbdriver);
    myConn = DriverManager.getConnection (dburl, user, passwd);
    stmt = myConn.createStatement();
} catch(SQLException ex) {
    System.err.println("SQLException: " + ex.getMessage());
}

SQL1 = "{? = call Date2EnrollYear(SYSDATE)}";
CallableStatement cstmt1 = myConn.prepareCall(SQL1);
cstmt1.registerOutParameter(1,java.sql.Types.INTEGER);
cstmt1.execute();
int nYear = cstmt1.getInt(1);

SQL2 = "{? = call Date2EnrollSemester(SYSDATE)}";
CallableStatement cstmt2 = myConn.prepareCall(SQL2);
cstmt2.registerOutParameter(1,java.sql.Types.INTEGER);
cstmt2.execute();
int nSemester = cstmt2.getInt(1);

%>
<p class="title" align="center">
    <%=nYear%> 년도 <%=nSemester%> 학기 수강삭제
</p>
<%
    mySQL = "select c_id, c_id_no, c_name, c_unit from course where (c_id, c_id_no) in (select c_id, c_id_no from enroll
where e_year ="
        + nYear + " and e_semester =" + nSemester + " and s_id=" + session_id + "));";
myResultSet = stmt.executeQuery(mySQL);
if (myResultSet != null) {
    while (myResultSet.next()) {
        String c_id = myResultSet.getString("c_id");
        String c_name = myResultSet.getString("c_name");
        int c_id_no = myResultSet.getInt("c_id_no");
        int c_unit = myResultSet.getInt("c_unit");
    }
}

%>
<tr>
    <td align="center"><%= c_id %> </td>
    <td align="center"><%= c_id_no %> </td>
    <td align="center"><%= c_name %> </td>
    <td align="center"><%= c_unit %> </td>

```

```

        <td align="center"><a href="delete_verify.jsp?c_id=<%= c_id %> &c_id_no=<%= c_id_no %> ">삭제</a></td>
    </tr>
<%
}
}
stmt.close();
myConn.close();
%>
</table>
</body>
</html>

```

; JDBC드라이버 로딩후 데이터베이스 연결, 전진과 읽기만 가능한 ResultSet 생성, 조건문 사용
 Connection 메소드인 prepareCall()사용하여 리턴 값이 있는 함수 처리, subquery를 이용한
 중첩질의 사용

연결된 DB의 Course와 Enroll 테이블을 이용하여 수강신청한 수업을 출력하여, 수강취소를 할 수
 있도록 한다.

Date2EnrollYear 과 Date2EnrollSemester 함수를 통해 현재 연도와 학기에 신청한 수업들만
 표시되도록 한다. Enroll 테이블에 있는 수업들 중 s_id 변수가 session_id인 수업들, 즉, 수강신청을
 한 수업들에 대해서만 <수업번호, 분반, 수업명, 학점, 수강취소> 변수를 출력한다.

원하는 수업의 <수강취소>의 [삭제] 버튼을 클릭하면 delete_verify.jsp를 통해 취소된다.

Delete_verify.jsp

```

<%@ page contentType="text/html; charset=UTF-8" %>
<%@ page import="java.sql.*" %>
<html>
<head>
<title>수강신청 취소</title>
</head>

<%@ include file="top.jsp"%>
<body>
    <%

    Connection myConn = null;
    String result = "";
    String dbdriver = "oracle.jdbc.driver.OracleDriver";
    String dburl = "jdbc:oracle:thin:@localhost:1521:xe";
    String user = "db1816461"; String passwd = "oracle";
    CallableStatement cstmt;

    try{

```



```

Class.forName(dbdriver);
myConn = DriverManager.getConnection (dburl, user, passwd);
} catch(SQLException ex) {
    System.err.println("SQLException: " + ex.getMessage());
}
String s_id = (String)session.getAttribute("user");
String c_id = request.getParameter("c_id");
int c_id_no = Integer.parseInt(request.getParameter("c_id_no"));

cstmt = myConn.prepareCall("{call DeleteEnroll(?,?,?)");
cstmt.setString(1,s_id);
cstmt.setString(2,c_id);
cstmt.setInt(3,c_id_no);
cstmt.registerOutParameter(4,java.sql.Types.VARCHAR);

CallableStatement cstmt2 = myConn.prepareCall("{call countstudent(?,?)");
cstmt2.setString(1,c_id);
cstmt2.setInt(2,c_id_no);

try{
    cstmt.execute();
    cstmt2.execute();
    result = cstmt.getString(4);
} catch(SQLException ex) {
    System.err.println("SQLException : " + ex.getMessage());
} finally{
    if(cstmt != null) {
        try{
            myConn.commit();
            cstmt.close();
            myConn.close();
        } catch(SQLException ex) {}
    }
}

<%= result %>
<script>
    alert("<%=result%>");
    location.href="delete.jsp";
</script>
<%
} catch(SQLException ex) {
    System.err.println("SQLException : " + ex.getMessage());
} finally{
    if(cstmt != null) {
        try{
            myConn.commit();
            cstmt.close();
            myConn.close();
        } catch(SQLException ex) {}
    }
}

%>
</body>
</html>

```

; JDBC드라이버 로딩후 데이터베이스 연결, Connection 메소드인 prepareCall()사용하여 OUT ,

조건문 사용, 파라미터가 있는 프로시저 처리, IN 프로시저만 있는 프로시저 처리
delete.jsp에서 [삭제] 버튼을 클릭한 수업에 대하여 DeleteEnroll 프로시저를 통해 수강신청을 취소한다. 취소가 완료되었다면 countstudent 프로시저로 수업별 수강 신청 인원을 새로 계산하여 Course 테이블의 c_pop변수를 UPDATE한다.

Delete_enroll.sql

```
CREATE OR Replace PROCEDURE DeleteEnroll (sStudentId IN VARCHAR2,
                                           sCourseId IN VARCHAR2,
                                           nCourseIdNo IN NUMBER,
                                           result OUT VARCHAR2)

IS
BEGIN
    result := '';

    DBMS_OUTPUT.put_line('#');
    DBMS_OUTPUT.put_line(sStudentId || '님이 수업번호 ' || sCourseId || ', 분반 ' || TO_CHAR(nCourseIdNo) || '의
수강 취소를 요청하였습니다.');
```

```
    DELETE
    FROM enroll
    WHERE s_id = sStudentId and c_id = sCourseId and c_id_no = nCourseIdNo;

    COMMIT;
    result := '수강취소가 완료되었습니다.';

EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        result := SQLCODE;

END;
/
```

delete_verify.jsp에서 [삭제] 버튼을 클릭한 수업의 s_id, c_id, c_id_no 변수를 입력 파라미터로 받아 해당하는 데이터를 삭제한 후 "수강취소가 완료되었습니다." 팝업을 출력한다.

Select.jsp

```
<%@ page contentType="text/html; charset=UTF-8"%>
<%@ page import="java.sql.*"%>
<html>
<head>
<title>수강신청 조회</title>
</head>
```

```

<body>
<%@ include file= "top.jsp"%>
<%
    if (session_id==null)
        response.sendRedirect("login.jsp");
%>

<form id= "select_form" method= "post">
<table class= "classtable" width= "75%" align= "center" border>
    <tr>
        <th>수업번호</th>
        <th>분반</th>
        <th>수업명</th>
        <th>학점</th>
    </tr>
<%

    String result1 = null;
    String result2 = null;
    Connection myConn = null;
    Statement stmt = null;
    Statement stmt2 = null;
    ResultSet myResultSet = null;

    String SQL1 = "";
    String SQL2 = "";
    String mySQL = "";
    String dbdriver = "oracle.jdbc.driver.OracleDriver";
    String dburl = "jdbc:oracle:thin:@localhost:1521:xe";
    String user = "db1816461"; String passwd = "oracle";

    try {
        Class.forName(dbdriver);
        myConn = DriverManager.getConnection(dburl, user, passwd);
        stmt = myConn.createStatement();
    } catch (SQLException ex) {
        System.err.println("SQLException: " + ex.getMessage());
    }

    SQL1 = "{? = call Date2EnrollYear(SYSDATE)}";
    CallableStatement cstmt1 = myConn.prepareCall(SQL1);
    cstmt1.registerOutParameter(1,java.sql.Types.INTEGER);
    cstmt1.execute();
    int nYear = cstmt1.getInt(1);

    SQL2 = "{? = call Date2EnrollSemester(SYSDATE)}";
    CallableStatement cstmt2 = myConn.prepareCall(SQL2);
    cstmt2.registerOutParameter(1,java.sql.Types.INTEGER);
    cstmt2.execute();
    int nSemester = cstmt2.getInt(1);

```

```

        mySQL = "select c_id, c_name, c_id_no, c_unit from course where (c_id, c_id_no) in (select
c_id, c_id_no from enroll where e_year = "
                                + nYear + " and e_semester =" + nSemester + " and s_id=" +
session_id + ")";

myResultSet = stmt.executeQuery(mySQL);

if (myResultSet != null) {
    while (myResultSet.next()) {
        String c_id = myResultSet.getString("c_id");
        String c_name = myResultSet.getString("c_name");
        int c_id_no = myResultSet.getInt("c_id_no");
        int c_unit = myResultSet.getInt("c_unit");

        %>
        <tr>

            <td width = "5%" align="center"><%=c_id%> </td>
            <td width = "3%" align="center"><%=c_id_no%> </td>
            <td width = "10%" align="center"><%=c_name%> </td>
            <td width = "3%" align="center"><%=c_unit%> </td>

        </tr>

    <%
    }
    }

    CallableStatement cstmt = myConn.prepareCall("{call SelectTimeTable(?,?,?,?)");
    cstmt.setString(1, session_id);
    cstmt.setInt(2, nYear);
    cstmt.setInt(3, nSemester);
    cstmt.registerOutParameter(4, java.sql.Types.VARCHAR);
    cstmt.registerOutParameter(5, java.sql.Types.VARCHAR);

    try {
        cstmt.execute();
        result1 = cstmt.getString(4);
        result2 = cstmt.getString(5);
        %>
        <p align = "center" class="view">
            <%=nYear%> 년도 <%=nSemester%> 학기 수강조회
        </p>

        <table align = "center" class="simpletable">
            <br>
            <tr>

                <td>총 신청 수업 수</td>
                <td><%=result1%> </td>

            </tr>

            <tr>

                <td>총 신청 학점 수</td>

```

```

                                <td> <%=result2%> </td>
                                </tr>
                                </table>
                                <%
                                } catch (SQLException ex) {
                                    System.err.println("SQLException: " + ex.getMessage());
                                } finally {
                                    if (cstmt != null)
                                        try {
                                            myConn.commit();
                                            cstmt.close();
                                            myConn.close();
                                        } catch (SQLException ex) {
                                        }
                                    }
                                }
                                %>
</table>
</form>
</body>
</html>

```

; JDBC드라이버 로딩후 데이터베이스 연결, 전진과 읽기만 가능한 ResultSet생성, 조건문 사용
 Connection 메소드인 prepareCall()사용하여 리턴값이 있는 함수 처리, OUT파라미터가 있는
 프로시저 처리, subquery를 이용한 중첩질의 사용

연결된 DB의 Course와 Enroll 테이블을 이용하여 수강신청이된 수업들을 조회하여 화면에
 출력한다.

Date2EnrollYear 과 Date2EnrollSemester 함수를 통해 현재 연도와 학기에 신청한 수강 수업들만
 표시되도록 한다. Enroll 테이블에 있는 수업들 중 s_id 변수가 session_id인 수업들, 즉, 수강신청을
 한 수업들에 대해서만 <수업번호, 수업명, 분반, 학점> 변수를 출력한다.

SelectTimeTable 프로시저를 통해 현재 연도와 학기에 session_id가 신청한 전체 수업 개수와 총
 신청 학점 수를 계산하여 하단에 출력한다.

SelectTimeTable.sql

```

CREATE OR REPLACE PROCEDURE SelectTimeTable
(sStudentId IN enroll.s_id%TYPE,
nYear IN enroll.e_year%TYPE,
nSemester IN enroll.e_semester%TYPE,
tCourse OUT NUMBER,
tUnit OUT NUMBER)
IS
BEGIN
    DECLARE
        CURSOR timeTable IS
            SELECT c_time,c_id,c_name,c_id_no,c_unit

```

```

        FROM course
        WHERE (c_id, c_id_no) IN (SELECT c_id, c_id_no FROM enroll WHERE s_id=sStudentId AND e_year =
nYear AND e_semester = nSemester );
        curTab timeTable%ROWTYPE;
    BEGIN
        DBMS_OUTPUT.put_line('#');
        DBMS_OUTPUT.put_line(sStudentId || '님의 ' || nYear || '년도 ' || nSemester || '학기의 수강신청 시간표입니다.');
```

tCourse := 0;

tUnit := 0;

OPEN timeTable;

LOOP

 FETCH timeTable INTO curTab;

 EXIT WHEN timeTable%NOTFOUND;

 tCourse := tCourse + 1;

 tUnit := tUnit + curTab.c_unit;

 DBMS_OUTPUT.put_line('교시:' || curTab.c_time || ', 수업코드:' || curTab.c_id || ', 수업명 : ' || curTab.c_name || ', 분반 : ' || curTab.c_id_no || ', 학점 : ' || curTab.c_unit);

END LOOP;

DBMS_OUTPUT.put_line('총 ' || tCourse || '수업과 총 ' || tUnit || '학점을 신청하였습니다.');

END;

END;

/

DECLARE

tCourse NUMBER;

tUnit NUMBER;

BEGIN

SelectTimeTable('1816461',2021,2,tCourse,tUnit);

END;

/

; %TYPE, %ROWTYPE형식,파라미터가 없는 커서 사용

Enroll 테이블에서 s_id 변수가 session_id인 수업 중 현재 연도와 학기에 속하는 수업들의 리스트와 총 개수, 총 학점 수를 계산하여 리턴한다.

InsertTest.sql

```

Set ServerOutput On;
VAR tCourse NUMBER;
VAR tunit NUMBER;
DECLARE
result VARCHAR2(70) := '';
BEGIN

DBMS_OUTPUT.enable;

DBMS_OUTPUT.put_line('***** Insert 및 에러 처리 테스트 *****');
```

```

/* 에러 처리 2 : 동일한 수업 신청 여부 :데이터베이스프로그래밍 수업 신청 */
InsertEnroll('1710385', 'C800', 2, result);
DBMS_OUTPUT.put_line('결과 : ' || result);

/* 에러 처리 3 : 수강신청 인원 초과 여부 : 멀티미디어 개론 신청*/
InsertEnroll('1710385', 'C900', 3, result);
DBMS_OUTPUT.put_line('결과 : ' || result);

/* 에러 처리 4 : 신청한 수업들 시간 중복 여부 : 컴퓨터 그래픽스 신청 */
InsertEnroll('1710385', 'M600', 3, result);
DBMS_OUTPUT.put_line('결과 : ' || result);

/* 에러가 없는 경우 : 게임프로그래밍 신청 */
InsertEnroll('1710385', 'M500', 3, result);
DBMS_OUTPUT.put_line('결과 : ' || result);

/* 에러 처리 1 : 최대 학점 초과 여부 검사 : 기초통계 신청 */
InsertEnroll('1710385', 'S100', 1, result);
DBMS_OUTPUT.put_line('결과 : ' || result);

DBMS_OUTPUT.put_line('***** CURSOR를 이용한 SELECT 테스트 *****');

/* 최종 결과 확인 */
SelectTimeTable('1710385', 2021, 2, :tCourse, :tUnit);

delete from enroll where s_id='1710385' and c_id='C800' and c_id_no=3;

END;
/

SELECT s_id,s_pwd FROM STUDENT WHERE s_id='1710385';
UPDATE STUDENT SET s_pwd='12' WHERE s_id='1710385';
UPDATE STUDENT SET s_pwd='1 345' WHERE s_id='1710385';
UPDATE STUDENT SET s_pwd=NULL WHERE s_id='1710385';
UPDATE STUDENT SET s_pwd='12345' WHERE s_id='1710385';
SELECT s_id,s_pwd FROM STUDENT WHERE s_id='1710385';

```

작성한 모든 프로시저와 함수등에 대하여 잘 작동하는지 Test한다.

InsertEnroll, SelectTimeTable을 통하여 수강신청과 수강조회에 대한 예외처리와 실행이 잘 작동하는지 확인한다.

Student 테이블의 s_pwd 변수 UPDATE 질의문을 통해서 BeforeUpdateStudent 트리거가 잘

작동하는지 확인하고 변경 전 후의 사용자 정보를 확인한다.

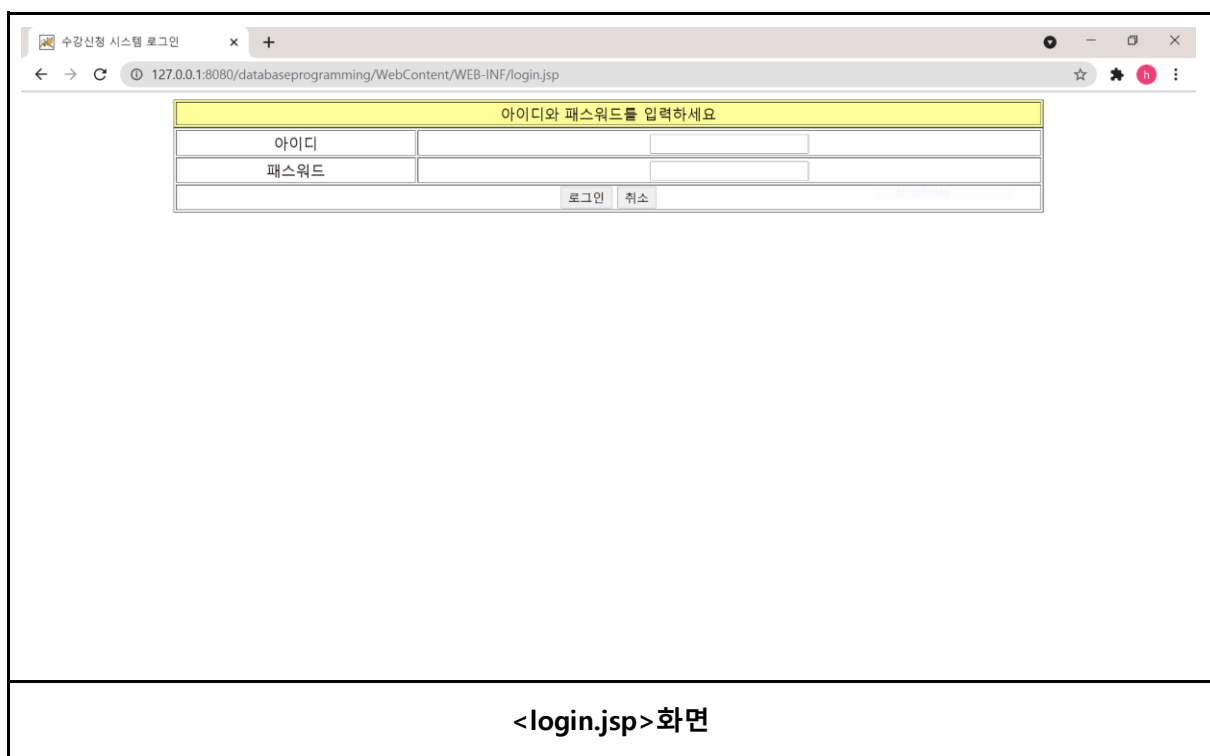
3. 화면 구성 설명



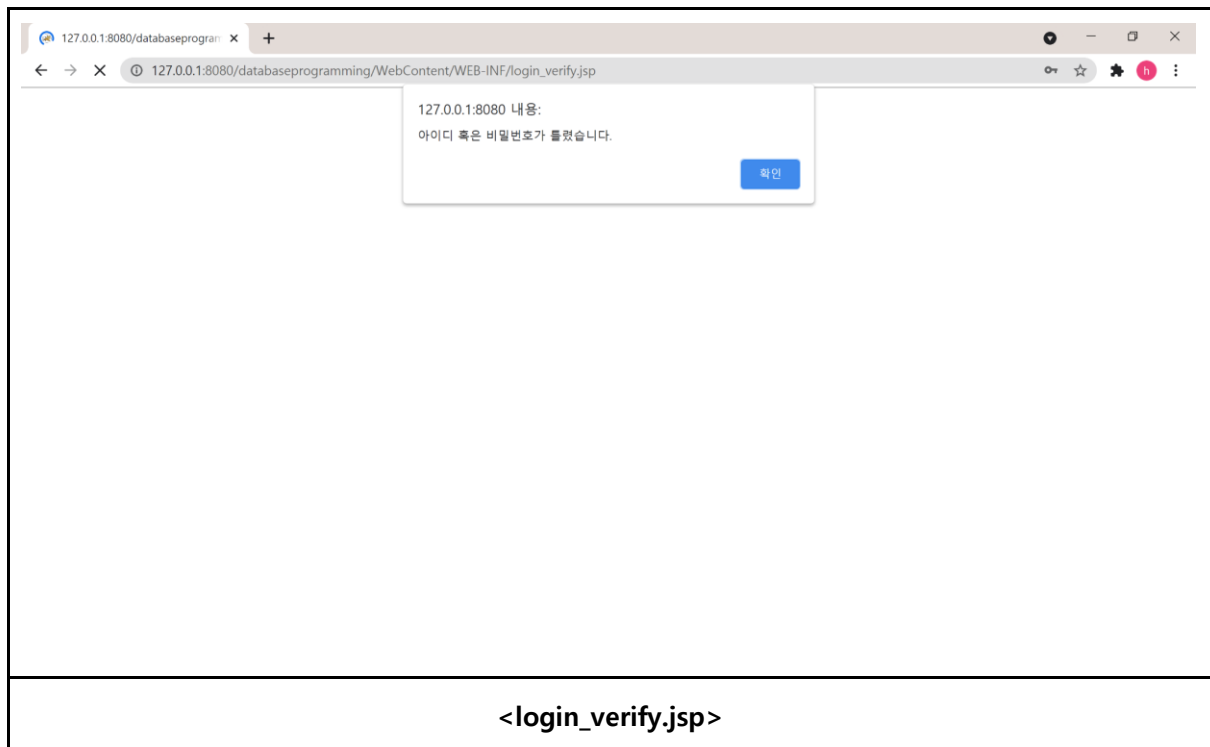
<main.jsp>화면에서 [로그인] 클릭 시 <login.jsp>화면으로 이동한다.



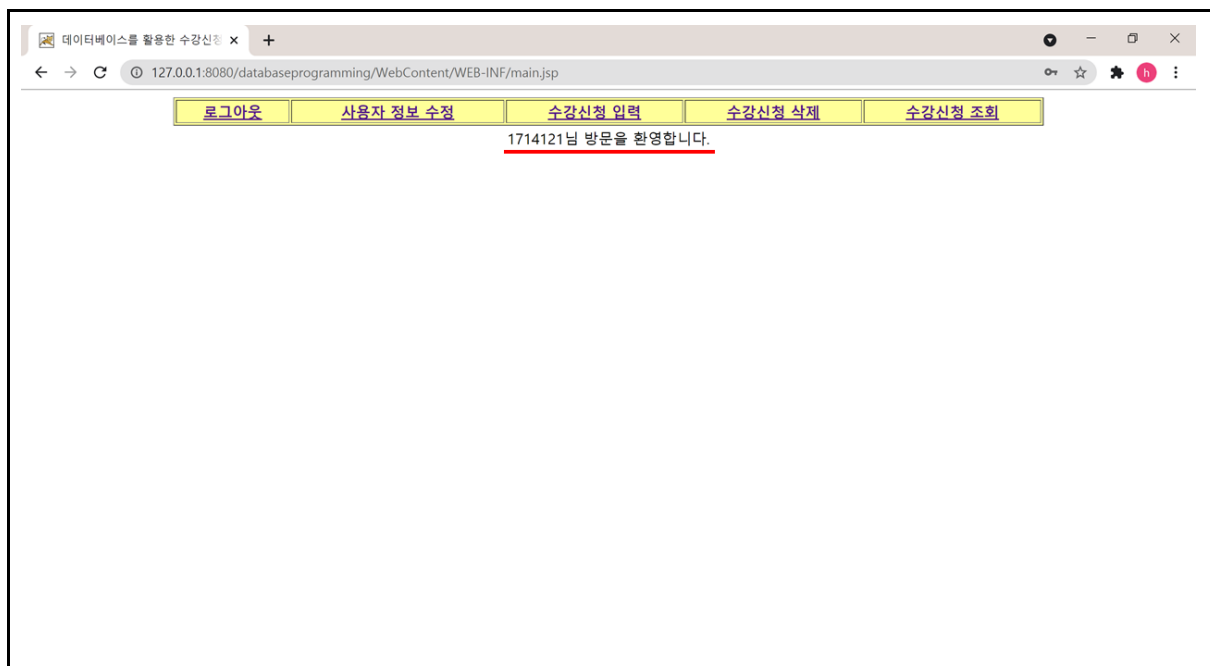
<main.jsp>화면에서 로그인 하지 않고 [사용자 정보 수정] / [수강신청 입력] / [수강신청 삭제] / [수강신청 조회] 클릭 시 <login.jsp>화면으로 이동한다.



<login.jsp> 화면에서 아이디(학번) 암호 입력시 <login_verify.jsp>에서 저장된 테이블에 입력된 값이 있는 지 확인한다.



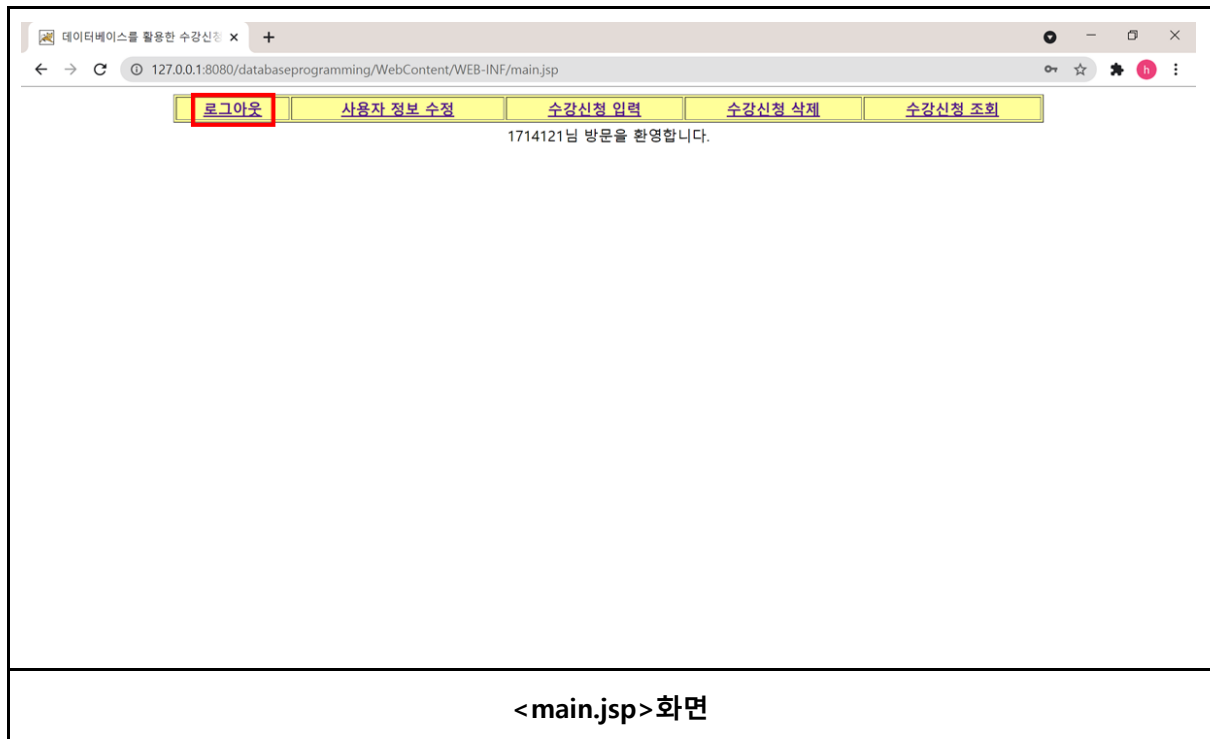
잘못된 아이디(학번) 혹은 암호 입력 시 "아이디 혹은 비밀번호가 틀렸습니다." 팝업을 보여주고 로그인에 실패하여 <login.jsp> 화면으로 돌아간다.



<main.jsp>화면

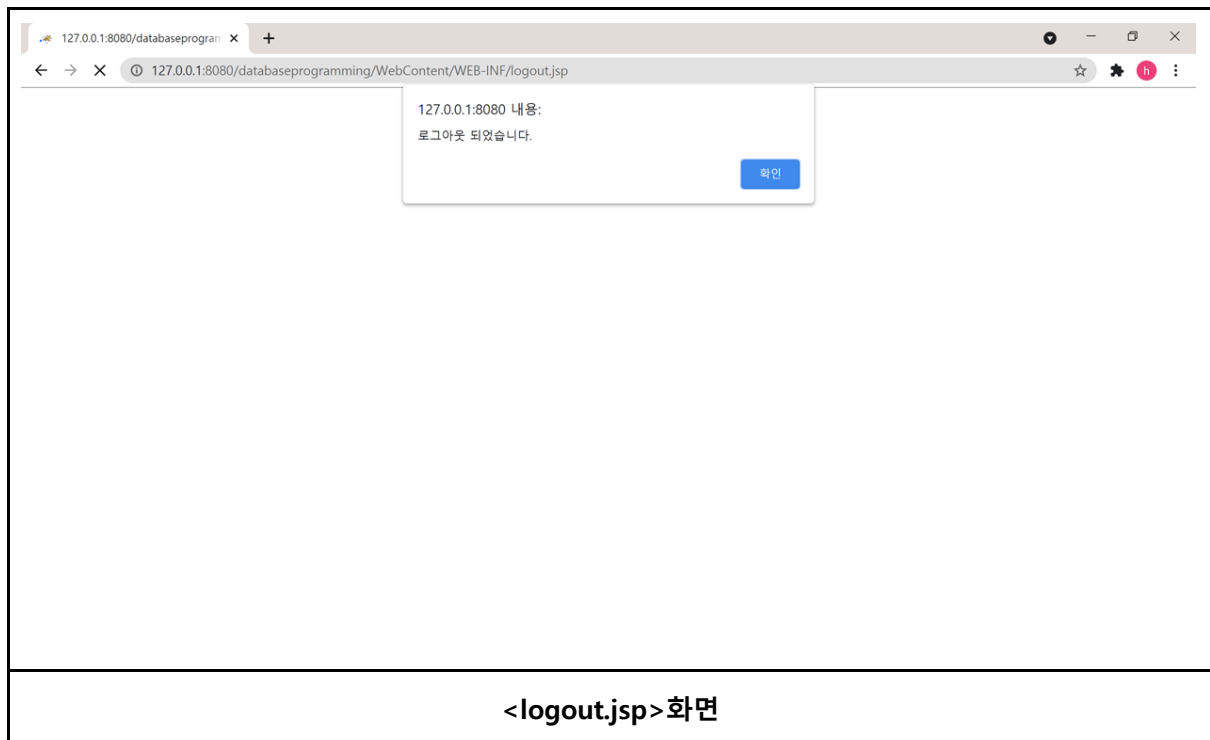
올바른 아이디(학번)와 암호 입력시, 로그인 성공 및 <main.jsp>화면으로 돌아간다.

<main.jsp>화면에 해당하는 아이디(학번)로 문구가 바뀔 것을 확인할 수 있다. 메뉴는 [로그인] 에서 [로그아웃]으로 변경된다.

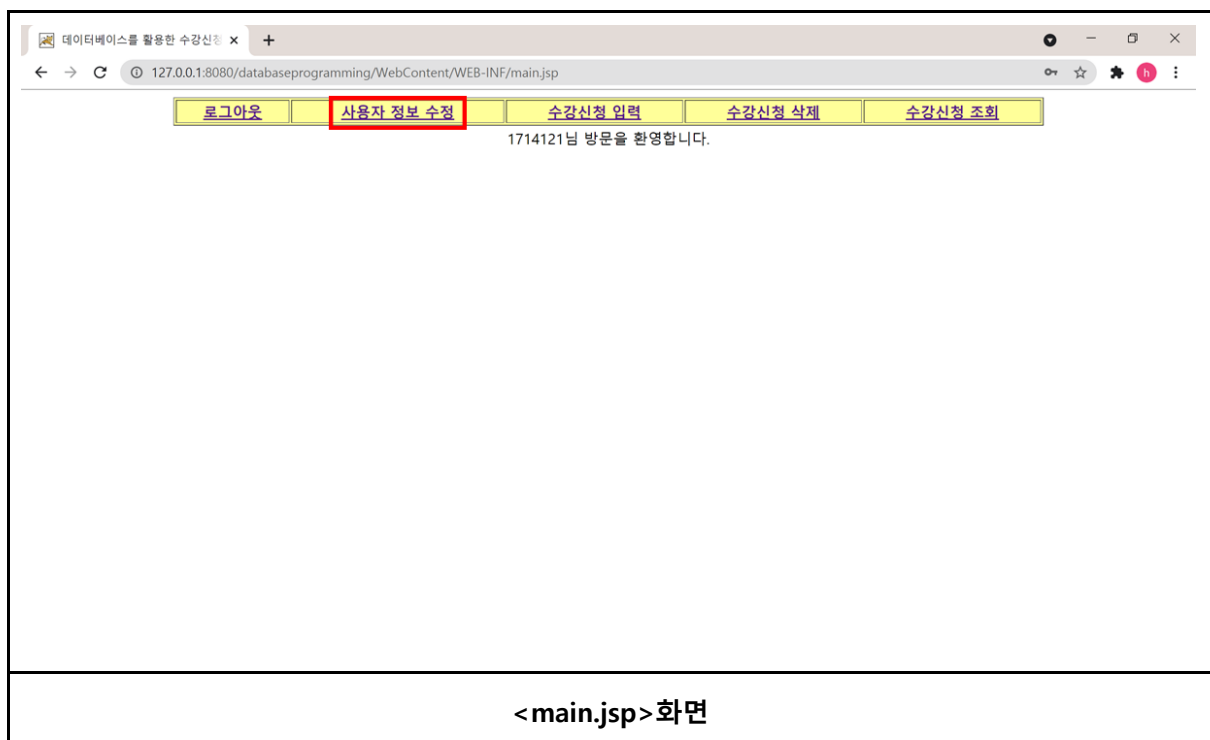


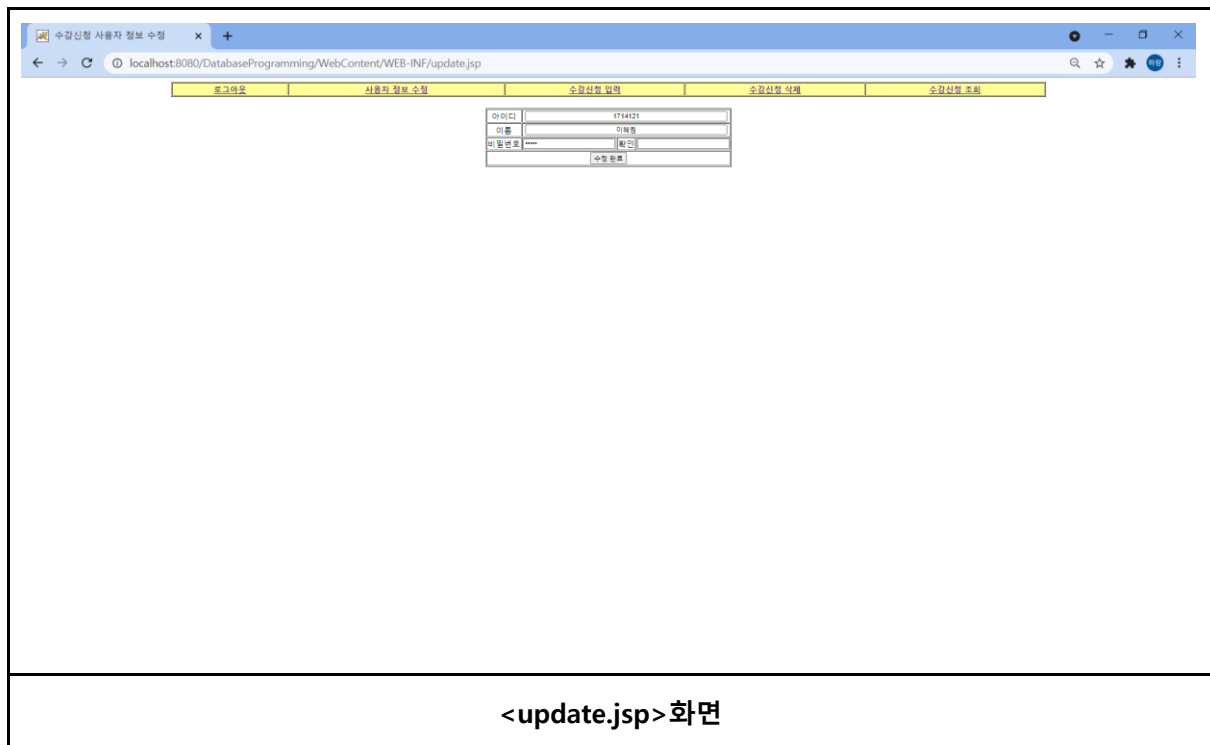
<main.jsp>화면

[로그아웃] 클릭 시 <logout.jsp> 실행된다.



[로그아웃] 클릭 후 성공적으로 로그아웃 되면 "로그아웃 되었습니다." 팝업을 보여준다.





[사용자 정보 수정]을 클릭하면 로그인된 사용자 정보를 조회하여 <update.jsp> 화면을 보여준다. 아이디와 이름을 보여주며 비밀번호는 가려져있다. 비밀번호와 이름을 수정하기를 요청하는 경우 <update_verify.jsp>를 호출한다.



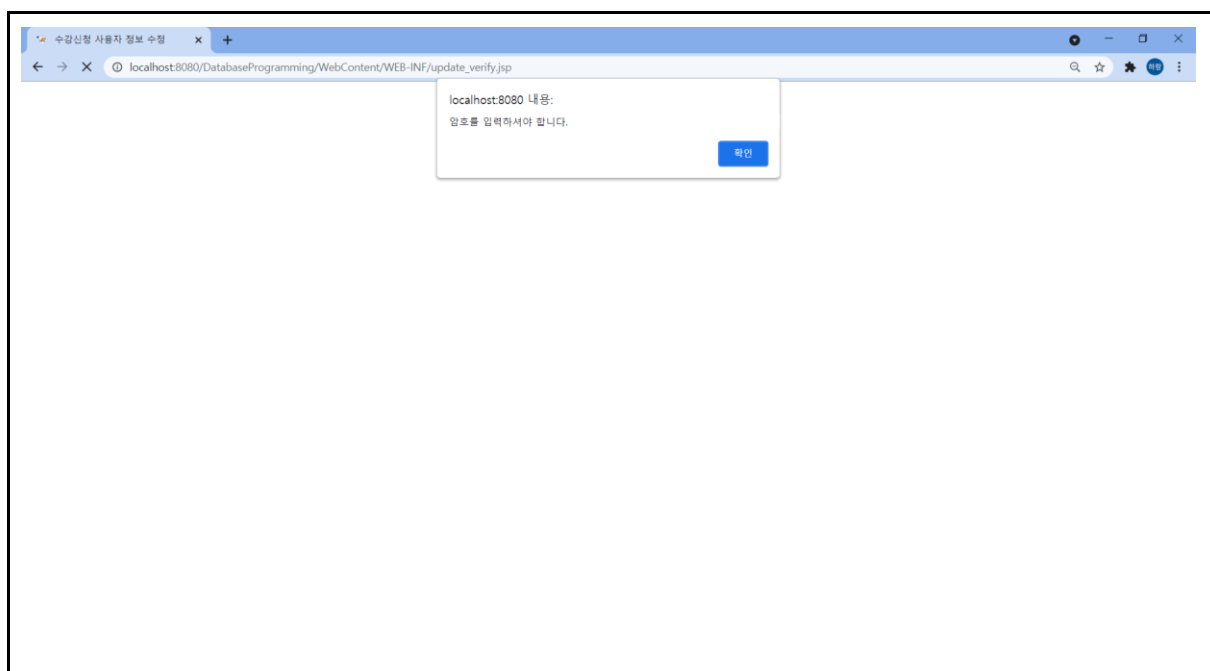
<update.jsp>화면에서 입력한 [비밀번호]와 [확인]이 다를 경우 "비밀번호를 다시 확인해주세요."
팝업을 보여주고, [확인] 클릭 시 <update.jsp>화면으로 돌아간다.



<update.jsp>화면에서 입력한 [비밀번호]와 [확인]이 같지만, 그 값이 4자리수 미만일 경우 BeforeUpdatedStudent 트리거에 의해 "암호는 4자리 이상이어야 합니다." 팝업이 뜨며 Student 테이블이 수정되지 않는다. 이때의 에러번호는 -20002이다. [확인]클릭 시 <update.jsp>화면으로 돌아간다.

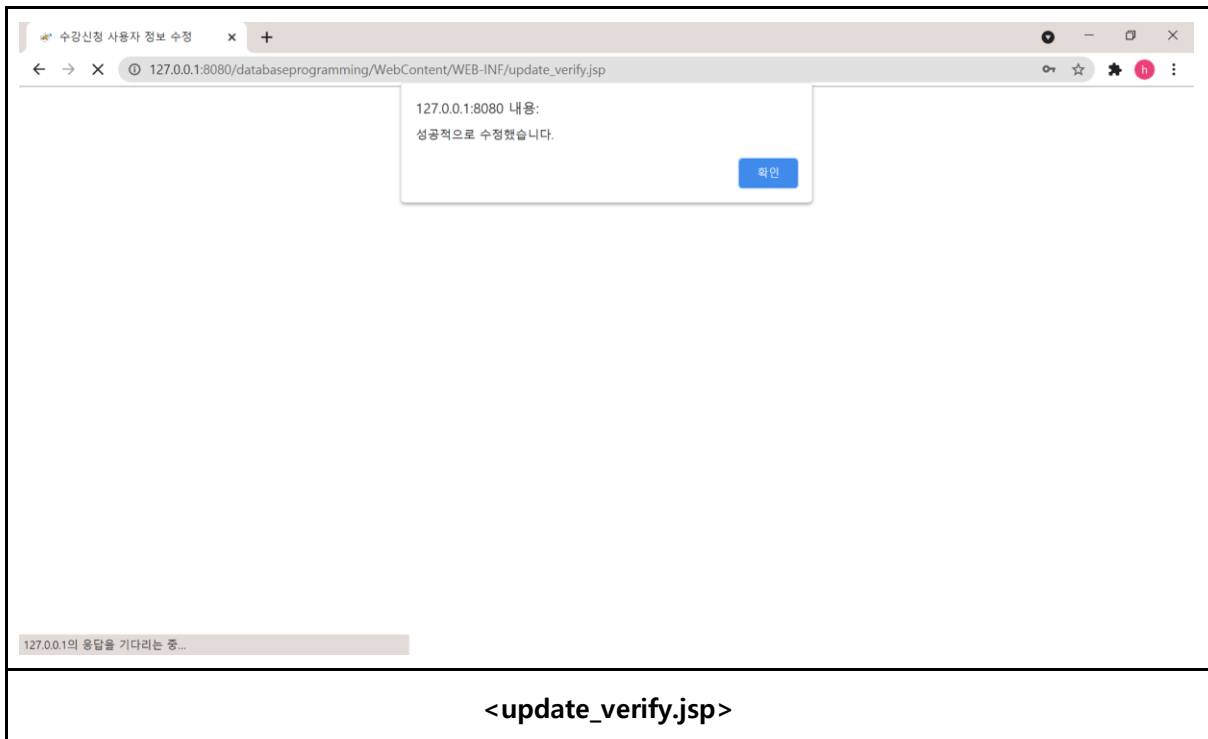


<update.jsp>화면에서 입력한 [비밀번호]와 [확인]이 같지만, 그 값에 공란이 포함되어 있을 경우 BeforeUpdatedStudent 트리거에 의해 “암호에 공란은 입력되지 않습니다.” 팝업이 뜨며 Student 테이블이 수정되지 않는다. 이때의 에러번호는 -20003이다. [확인]클릭 시 <update.jsp>화면으로 돌아간다.



<update_verify.jsp>

<update.jsp>화면에서 입력한 [비밀번호]와 [확인]이 모두 빈칸이면, BeforeUpdatedStudent 트리거에 의해 "암호를 입력하셔야 합니다." 팝업이 뜨며 Student 테이블이 수정되지 않는다. 이때의 에러번호는 -20004이다. [확인]클릭 시 <update.jsp>화면으로 돌아간다.



<update_verify.jsp>

<update.jsp>화면에서 입력한 [비밀번호]와 [확인]이 같고, 4자리 이상이며, 공란이 없고, NULL값이 아니라면 Student 테이블에 업데이트를 수행하고, "성공적으로 수정했습니다." 팝업이 뜬다. [확인]을 누르면 <main.jsp>화면으로 이동한다.

수강신청
127.0.0.1:8080/databaseprogramming/WebContent/WEB-INF/insert.jsp

로그아웃
사용자 정보 수정
수강신청 입력
수강신청 삭제
수강신청 조회

2021년도 2학기 수강신청

학과전공:
검색

학과전공	과목번호	분반	과목명	시간	학점	신청인원	수강신청
컴퓨터공학과	C800	2	데이터베이스 프로그래밍	1	3	0 / 5	신청
컴퓨터공학과	C800	3	데이터베이스 프로그래밍	1	3	0 / 5	신청
컴퓨터공학과	C900	3	멀티미디어 개론	2	3	0 / 5	신청
컴퓨터공학과	M100	3	선형대수	3	3	0 / 5	신청
컴퓨터공학과	M200	3	그래픽 활용	4	3	0 / 5	신청
컴퓨터공학과	M300	3	윈도우 프로그래밍	5	3	0 / 5	신청
컴퓨터공학과	M400	3	멀티미디어 처리	6	3	0 / 5	신청
컴퓨터공학과	M500	3	게임프로그래밍	7	3	0 / 5	신청
컴퓨터공학과	M600	3	컴퓨터 그래픽스	1	3	0 / 5	신청
통계학과	S100	1	기초통계학	1	3	0 / 5	신청
통계학과	S200	1	수리통계학	1	3	0 / 5	신청

<insert.jsp> 화면

[수강신청 입력]을 클릭하면 Course테이블에서 c_year = 2021, c_semester = 2 인 강의의 정보(학과전공, 수업번호, 분반, 수업명, 시간, 학점,신청인원,최대인원)를 조회한 <insert.jsp>화면을 보여준다. [신청]을 누르는경우 <insert_verify.jsp>를 호출한다.

수강신청 조회
127.0.0.1:8080/databaseprogramming/WebContent/WEB-INF/select.jsp
로그아웃
사용자 정보 수정
수강신청 입력
수강신청 삭제
수강신청 조회

2021년도 2학기 수강조회

과목번호	분반	과목명	학점
C800	2	데이터베이스 프로그래밍	3
M100	3	선형대수	3
M200	3	그래픽 활용	3
M300	3	윈도우 프로그래밍	3
M400	3	멀티미디어 처리	3
M500	3	게임프로그래밍	3

총 신청 과목 수 6
총 신청 학점 수 18

수강신청 조회
127.0.0.1:8080/databaseprogramming/WebContent/WEB-INF/select.jsp
로그아웃
사용자 정보 수정
수강신청
수강신청 삭제
수강신청 조회

2021년도 2학기 수강신청

학과전공:
검색

학과전공	과목번호	분반	과목명	시간	학점	신청인원	수강신청
컴퓨터공학과	C800	3	데이터베이스 프로그래밍	1	3	0 / 5	신청
컴퓨터공학과	C900	3	멀티미디어 개론	2	3	0 / 5	신청
컴퓨터공학과	M600	3	컴퓨터 그래픽스	1	3	0 / 5	신청
통계학과	S100	1	기초통계학	1	3	0 / 5	신청
통계학과	S200	1	수리통계학	1	3	0 / 5	신청



[신청]을 클릭하면 <insert_verify.jsp>를 호출하고, InsertEnroll프로시저가 호출된다.
Enroll 테이블에 있는 각 s_id가 신청한 수업들의 총 학점을 계산하고, 새로 신청하는 수업의 학점을 더하였을 때, 18학점(최대 학점)을 초과하면 'too_many_sumCourseUnit' 에러를 발생시켜 "최대학점을 초과하였습니다." 팝업을 출력한다.

수강신청
127.0.0.1:8080/databaseprogramming/WebContent/WEB-INF/insert.jsp
로그아웃
사용자 정보 수정
수강신청 입력
수강신청 삭제
수강신청 조회

2021년도 2학기 수강신청
학과전공:
검색

학과전공	과목번호	분반	과목명	시간	학점	신청인원	수강신청
컴퓨터공학과	C800	2	데이터베이스 프로그래밍	1	3	0 / 5	신청
컴퓨터공학과	C800	3	데이터베이스 프로그래밍	1	3	0 / 5	신청
컴퓨터공학과	C900	3	멀티미디어 개론	2	3	0 / 5	신청
컴퓨터공학과	M100	3	선형대수	3	3	4 / 5	신청
컴퓨터공학과	M200	3	그래픽 활용	4	3	0 / 5	신청
컴퓨터공학과	M300	3	윈도우 프로그래밍	5	3	0 / 5	신청
컴퓨터공학과	M400	3	멀티미디어 처리	6	3	0 / 5	신청
컴퓨터공학과	M500	3	게임프로그래밍	7	3	0 / 5	신청
컴퓨터공학과	M600	3	컴퓨터 그래픽스	1	3	0 / 5	신청
통계학과	S100	1	기초통계학	1	3	0 / 5	신청
통계학과	S200	1	수리통계학	1	3	0 / 5	신청

수강신청
127.0.0.1:8080/databaseprogramming/WebContent/WEB-INF/insert_verify.jsp?c_id=C800&c_id_no=2
127.0.0.1:8080 내용:
수강신청완료
확인

수강신청
127.0.0.1:8080/databaseprogramming/WebContent/WEB-INF/insert.jsp
로그아웃
사용자 정보 수정
수강신청 입력
수강신청 삭제
수강신청 조회

2021년도 2학기 수강신청
학과전공:
검색

학과전공	과목번호	분반	과목명	시간	학점	신청인원	수강신청
컴퓨터공학과	C800	3	데이터베이스 프로그래밍	1	3	0 / 5	신청
컴퓨터공학과	C900	3	멀티미디어 개론	2	3	0 / 5	신청
컴퓨터공학과	M100	3	선형대수	3	3	4 / 5	신청
컴퓨터공학과	M200	3	그래픽 활용	4	3	0 / 5	신청
컴퓨터공학과	M300	3	윈도우 프로그래밍	5	3	0 / 5	신청
컴퓨터공학과	M400	3	멀티미디어 처리	6	3	0 / 5	신청
컴퓨터공학과	M500	3	게임프로그래밍	7	3	0 / 5	신청
컴퓨터공학과	M600	3	컴퓨터 그래픽스	1	3	0 / 5	신청
통계학과	S100	1	기초통계학	1	3	0 / 5	신청
통계학과	S200	1	수리통계학	1	3	0 / 5	신청

수강신청
127.0.0.1:8080/databaseprogramming/WebContent/WEB-INF/insert_verify.jsp?c_id=C800&c_id_no=3
127.0.0.1:8080 내용:
이미 등록된 과목을 신청하였습니다
확인

127.0.0.1의 응답을 기다리는 중...

<insert.jsp>화면 ⇒ <insert_verify.jsp>

[신청]을 클릭하면 <insert_verify.jsp>를 호출하고, InsertEnroll프로시저가 호출된다.

InsertEnroll 프로시저에서 에러가 발생하지 않으면 "수강신청완료" 팝업이 출력되며, 신청한 수업의 정보가 Enroll테이블에 저장, countstudent 프로시저시저가 호출되어 수강신청인원(course.c_pop)을 계산하여 수정한다.

만약, Enroll 테이블에서 신청한 수업과 동일한 연도, 학기, 수업코드를 갖는 수업의 수가 0을 초과하면 'duplicate_courses' 에러를 발생시켜 "이미 등록된 수업을 신청하였습니다."라는 팝업을 출력한다.

수강신청

127.0.0.1:8080/databaseprogramming/WebContent/WEB-INF/insert.jsp

로그아웃
사용자 정보 수정
수강신청 입력
수강신청 삭제
수강신청 조회

2021년도 2학기 수강신청

학과전공:
검색

학과전공	과목번호	분반	과목명	시간	학점	신청인원	수강신청
컴퓨터공학과	C800	2	데이터베이스 프로그래밍	1	3	1 / 5	신청
컴퓨터공학과	C800	3	데이터베이스 프로그래밍	1	3	0 / 5	신청
컴퓨터공학과	C900	3	멀티미디어 개론	2	3	1 / 5	신청
컴퓨터공학과	M100	3	선형대수	3	3	5 / 5	신청
컴퓨터공학과	M200	3	그래픽 활용	4	3	1 / 5	신청
컴퓨터공학과	M300	3	윈도우 프로그래밍	5	3	1 / 5	신청
컴퓨터공학과	M400	3	멀티미디어 처리	6	3	1 / 5	신청
컴퓨터공학과	M500	3	게임프로그래밍	7	3	0 / 5	신청
컴퓨터공학과	M600	3	컴퓨터 그래픽스	1	3	0 / 5	신청
통계학과	S100	1	기초통계학	1	3	0 / 5	신청
통계학과	S200	1	수리통계학	1	3	0 / 5	신청

수강신청

127.0.0.1:8080/databaseprogramming/WebContent/WEB-INF/insert_verify.jsp?c_id=M100&c_id_no=3

127.0.0.1:8080 내용:
수강신청 인원이 초과되어 등록이 불가능합니다

확인

<insert.jsp>화면 ⇒ <insert_verify.jsp>

[신청]을 클릭하면 <insert_verify.jsp>를 호출하고, InsertEnroll프로시저가 호출된다.

Course테이블에서 c_max 변수를 조회하고 Enroll 테이블에서 해당 수업을 신청한 학생 수를 계산하여 c_max 값 이상이면 'too_many_students' 에러를 발생시켜 "수강신청 인원이 초과되어 등록이 불가능합니다" 팝업을 출력한다.

수강신청
127.0.0.1:8080/databaseprogramming/WebContent/WEB-INF/insert.jsp
로그아웃
사용자 정보 수정
수강신청 입력
수강신청 삭제
수강신청 조회

2021년도 2학기 수강신청
학과전공:
검색

학과전공	과목번호	분반	과목명	시간	학점	신청인원	수강신청
컴퓨터공학과	C800	2	데이터베이스 프로그래밍	1	3	0 / 5	신청
컴퓨터공학과	C800	3	데이터베이스 프로그래밍	1	3	0 / 5	신청
컴퓨터공학과	C900	3	멀티미디어 개론	2	3	0 / 5	신청
컴퓨터공학과	M100	3	선형대수	3	3	4 / 5	신청
컴퓨터공학과	M200	3	그래픽 활용	4	3	0 / 5	신청
컴퓨터공학과	M300	3	윈도우 프로그래밍	5	3	0 / 5	신청
컴퓨터공학과	M400	3	멀티미디어 처리	6	3	0 / 5	신청
컴퓨터공학과	M500	3	게임프로그래밍	7	3	0 / 5	신청
컴퓨터공학과	M600	3	컴퓨터 그래픽스	1	3	0 / 5	신청
통계학과	S100	1	기초통계학	1	3	0 / 5	신청
통계학과	S200	1	수리통계학	1	3	0 / 5	신청

수강신청
127.0.0.1:8080/databaseprogramming/WebContent/WEB-INF/insert_verify.jsp?c_id=C800&c_id_no=2
127.0.0.1:8080 내용:
수강신청완료
확인

수강신청

로그아웃

사용자 정보 수정

수강신청 입력

수강신청 삭제

수강신청 조회

2021년도 2학기 수강신청

학과전공: 검색

학과전공	과목번호	분반	과목명	시간	학점	신청인원	수강신청
컴퓨터공학과	C800	3	데이터베이스 프로그래밍	1	3	0 / 5	신청
컴퓨터공학과	C900	3	멀티미디어 개론	2	3	0 / 5	신청
컴퓨터공학과	M100	3	선형대수	3	3	4 / 5	신청
컴퓨터공학과	M200	3	그래픽 활용	4	3	0 / 5	신청
컴퓨터공학과	M300	3	윈도우 프로그래밍	5	3	0 / 5	신청
컴퓨터공학과	M400	3	멀티미디어 처리	6	3	0 / 5	신청
컴퓨터공학과	M500	3	게임프로그래밍	7	3	0 / 5	신청
컴퓨터공학과	M600	3	컴퓨터 그래픽스	1	3	0 / 5	신청
통계학과	S100	1	기초통계학	1	3	0 / 5	신청
통계학과	S200	1	수리통계학	1	3	0 / 5	신청

수강신청

127.0.0.1:8080/databaseprogramming/WebContent/WEB-INF/insert_verify.jsp?c_id=S100&c_id_no=1

127.0.0.1:8080 내용:
이미 등록된 과목 중 중복되는 시간이 존재합니다

확인

<insert.jsp>화면 ⇒ <insert_verify.jsp>

[신청]을 클릭하면 <insert_verify.jsp>를 호출하고, InsertEnroll프로시저가 호출된다.

Enroll 테이블에서 새로 신청한 수업과 동일한 수업 교시를 갖는 수업의 수가 0을 초과하면 'duplicate_time' 에러를 발생시켜 "이미 등록된 수업 중 중복되는 시간이 존재합니다." 팝업을 출력한다.

수강신청
127.0.0.1:8080/databaseprogramming/WebContent/WEB-INF/insert.jsp
로그아웃
사용자 정보 수정
수강신청 입력
수강신청 삭제
수강신청 조회

2021년도 2학기 수강신청
학과전공: 컴퓨터공학과
검색

학과전공	과목번호	분반	과목명	시간	학점	신청인원	수강신청
컴퓨터공학과	C800	2	데이터베이스 프로그래밍	1	3	0 / 5	신청
컴퓨터공학과	C800	3	데이터베이스 프로그래밍	1	3	0 / 5	신청
컴퓨터공학과	C900	3	멀티미디어 개론	2	3	0 / 5	신청
컴퓨터공학과	M100	3	선형대수	3	3	0 / 5	신청
컴퓨터공학과	M200	3	그래픽 활용	4	3	0 / 5	신청
컴퓨터공학과	M300	3	윈도우 프로그래밍	5	3	0 / 5	신청
컴퓨터공학과	M400	3	멀티미디어 처리	6	3	0 / 5	신청
컴퓨터공학과	M500	3	게임프로그래밍	7	3	0 / 5	신청
컴퓨터공학과	M600	3	컴퓨터 그래픽스	1	3	0 / 5	신청

수강신청
127.0.0.1:8080/databaseprogramming/WebContent/WEB-INF/insert.jsp
로그아웃
사용자 정보 수정
수강신청 입력
수강신청 삭제
수강신청 조회

2021년도 2학기 수강신청
학과전공: 통계학과
검색

학과전공	과목번호	분반	과목명	시간	학점	신청인원	수강신청
통계학과	S100	1	기초통계학	1	3	0 / 5	신청
통계학과	S200	1	수리통계학	1	3	0 / 5	신청

수강신청

127.0.0.1:8080/databaseprogramming/WebContent/WEB-INF/insert.jsp

로그아웃

사용자 정보 수정

수강신청 입력

수강신청 삭제

수강신청 조회

2021년도 2학기 수강신청

학과전공:

학과전공	과목번호	분반	과목명	시간	학점	신청인원	수강신청
컴퓨터공학과	C800	3	데이터베이스 프로그래밍	1	3	0 / 5	신청
컴퓨터공학과	C900	3	멀티미디어 개론	2	3	0 / 5	신청
컴퓨터공학과	M100	3	선형대수	3	3	4 / 5	신청
컴퓨터공학과	M200	3	그래픽 활용	4	3	0 / 5	신청
컴퓨터공학과	M300	3	윈도우 프로그래밍	5	3	0 / 5	신청
컴퓨터공학과	M400	3	멀티미디어 처리	6	3	0 / 5	신청
컴퓨터공학과	M500	3	게임프로그래밍	7	3	0 / 5	신청
컴퓨터공학과	M600	3	컴퓨터 그래픽스	1	3	0 / 5	신청
통계학과	S100	1	기초통계학	1	3	0 / 5	신청
통계학과	S200	1	수리통계학	1	3	0 / 5	신청

수강신청

127.0.0.1:8080/databaseprogramming/WebContent/WEB-INF/insert.jsp

127.0.0.1:8080 내용:
검색 결과가 없습니다.

확인

<insert.jsp>

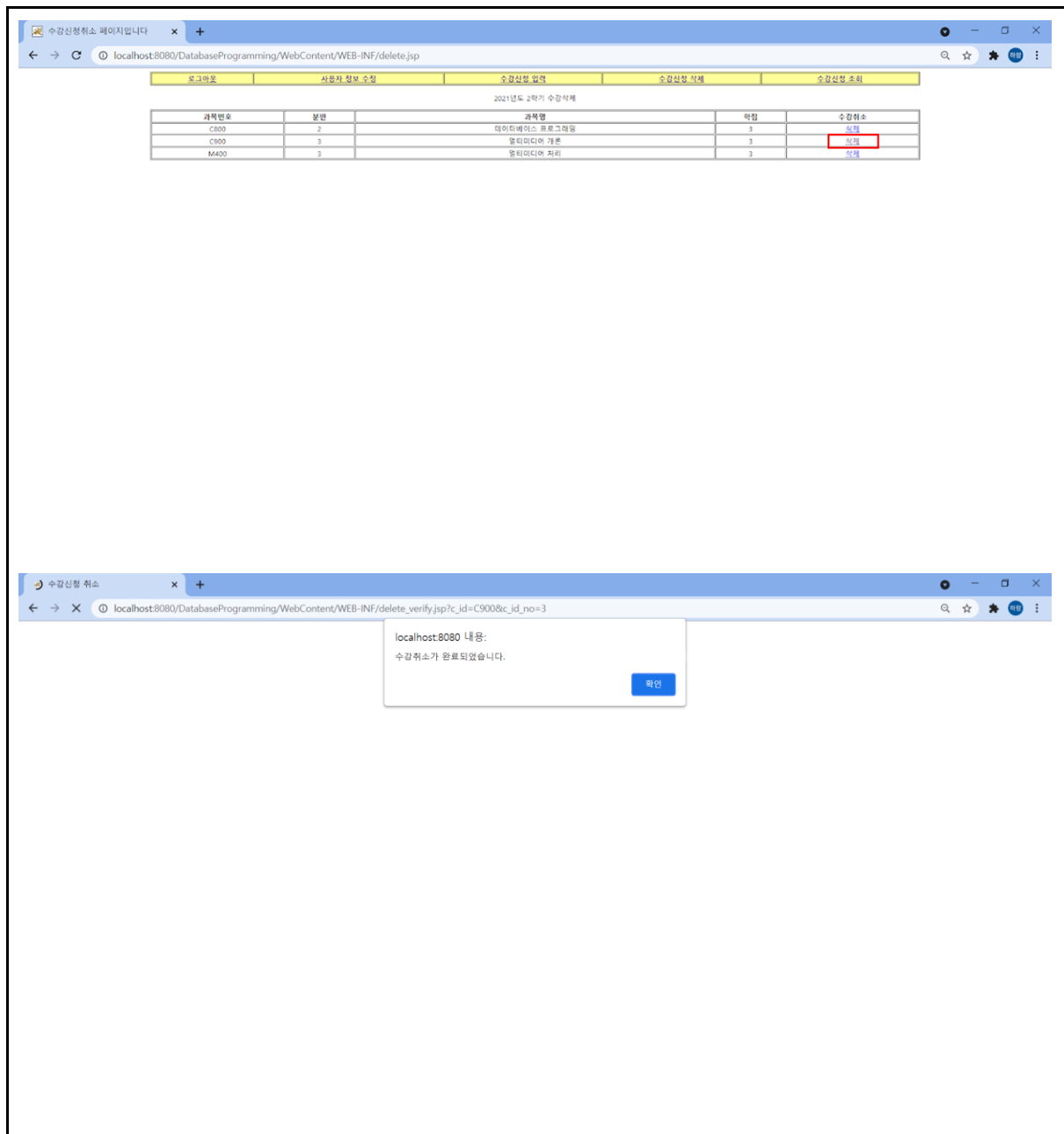
Course테이블에서 c_major = "컴퓨터공학과"인 ComScience 뷰와 c_major = "통계학과"인 Statistics 뷰를 생성하였다

학과전공 검색란에 '컴퓨터공학과'를 검색하면 ComScience 뷰에 있는 수업정보를, '통계학과'를 검색하면 Statistics 뷰에 있는 수업정보를 보여주고, 이외의 것을 검색하면 "검색 결과가 없습니다." 팝업이 출력된다.



로그인 후 [수강신청 삭제] 클릭 시 <delete.jsp>화면으로 이동한다.

<delete.jsp>화면에서는 로그인된 학번의 해당연도와 학기에 수강신청한 수업들의 수업번호, 분반, 수업명, 학점을 보여주고, 수강을 취소할 수 있는 수강취소 버튼도 함께 보여준다.



수강신청취소 페이지입니다

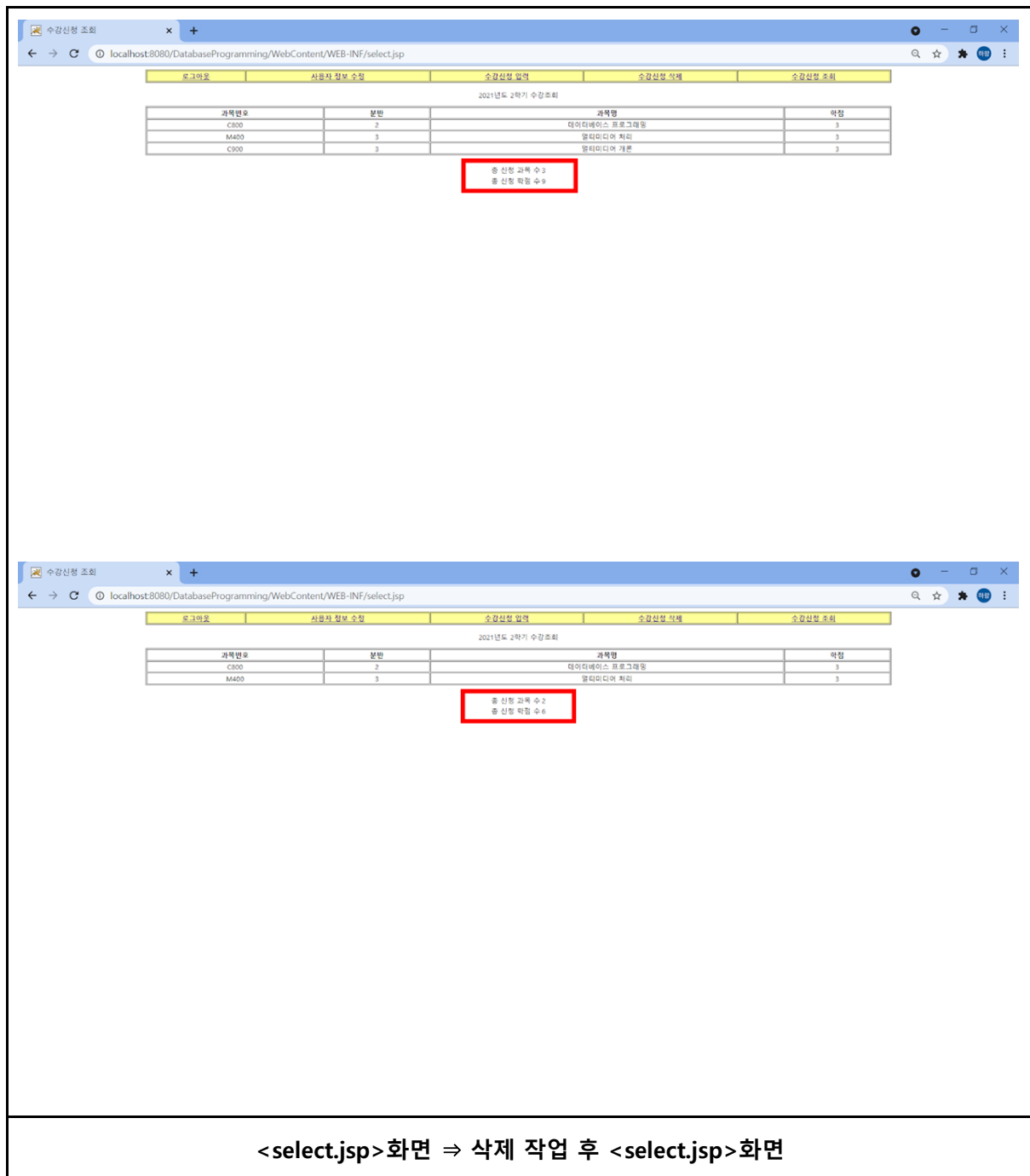
localhost:8080/DatabaseProgramming/WebContent/WEB-INF/delete.jsp

프로그램도	사용자 정보 수정	수강신청 입력	수강신청 삭제	수강신청 조회
2021년도 2학기 수강삭제				
과목번호	분반	과목명	학원	수강취소
C300	2	데이터베이스 프로그래밍	3	삭제
M400	3	멀티미디어 처리	3	삭제

<delete.jsp>화면 => <delete_verify.jsp> => <delete.jsp>화면

[삭제]를 클릭하면 <delete_verify.jsp>가 호출되어 DeleteEnroll 프로시저에 의해 해당하는 수업을 Enroll 테이블에서 삭제하고 "수강취소가 완료되었습니다." 팝업을 보여준다. [확인]을 누르면 <delete.jsp>화면으로 돌아가고, 업데이트된 <delete.jsp>화면을 보여준다.

또한 countstudent 프로시저에 의해 Course 테이블의 수강신청 인원(c_pop) 값이 업데이트 된다.



<delete_verify.jsp>가 성공적으로 완료되면 <select.jsp>화면에서도 해당하는 수업이 사라지고, 총 신청 수업 수와 총 신청 학점 수가 변경되는 것을 확인 할 수 있다.

4. SQL 서버를 통한 Test

1) InsertEnroll 프로시저와 SelectTimeTable 프로시저 Test

관리자: Start Database - sqlplus db1710385/oracle				
SQL> select * from enroll;				
C_ID	C_ID_NO	S_ID	E_YEAR	E_SEMESTER
C900	3	1714121	2021	2
C900	3	1816461	2021	2
C900	3	1712345	2021	2
C900	3	1612345	2021	2
C900	3	1512345	2021	2
C800	3	1710385	2021	2
M100	3	1710385	2021	2
M200	3	1710385	2021	2
M300	3	1710385	2021	2
M400	3	1710385	2021	2
10 rows selected.				
SQL>				
<insert_test.sql> 실행 이전 enroll테이블				

InsertEnroll 프로시저를 호출하여 사용자 정의 에러를 통하여 수강신청시 발생할 수 있는 에러를 확인한다.

1. 수업번호 'C800' 입력: 동일한 수업 신청 오류 발생
2. 수업번호 'C900' 입력: 수강신청 인원 초과 오류 발생
3. 수업번호 'M600' 입력: 신청한 수업들 시간 중복 발생
4. 수업번호 'M500' 입력: 정상적 입력
5. 수업번호 'S100' 입력: 최대 학점 초과 오류 발생

위의 오류를 확인 할 수 있는 질의문 입력 후, SelectTimeTable 프로시저 호출로 수강신청 결과 확인한다.

```
관리자: Start Database - sqlplus db1710385/oracle
SQL> VAR tCourse NUMBER;
SQL> VAR tUnit NUMBER;
SQL> DECLARE
2  result VARCHAR2(70) := '';
3  BEGIN
4
5  DBMS_OUTPUT.enable;
6
7  DBMS_OUTPUT.put_line('***** Insert 및 에러 처리 테스트 *****');
8
9  /* 에러 처리 2 : 동일한 과목 신청 여부 :데이터베이스프로그래밍 과목 신청 */
10 InsertEnroll('1710385', 'C800', 2, result);
11 DBMS_OUTPUT.put_line('결과 : ' || result);
12
13
14 /* 에러 처리 3 : 수강신청 인원 초과 여부 : 멀티미디어 개론 신청 */
15 InsertEnroll('1710385', 'C900', 3, result);
16 DBMS_OUTPUT.put_line('결과 : ' || result);
17
18
19 /* 에러 처리 4 : 신청한 과목들 시간 중복 여부 : 컴퓨터 그래픽스 신청 */
20 InsertEnroll('1710385', 'M600', 3, result);
21 DBMS_OUTPUT.put_line('결과 : ' || result);
22
23
24 /* 에러가 없는 경우 : 게임프로그래밍 신청 */
25 InsertEnroll('1710385', 'M500', 3, result);
26 DBMS_OUTPUT.put_line('결과 : ' || result);
27
28
29 /* 에러 처리 1 : 최대 학점 초과 여부 검사 : 기초통계 신청 */
30 InsertEnroll('1710385', 'S100', 1, result);
31 DBMS_OUTPUT.put_line('결과 : ' || result);
32
33
34 DBMS_OUTPUT.put_line('***** CURSOR를 이용한 SELECT 테스트 *****');
35
36 /* 최종 결과 확인 */
37 SelectTimeTable('1710385', 2021, 2, :tCourse, :tUnit);
38
39 delete from enroll where s_id='1710385' and c_id='C800' and c_id_no=3;
40
41 END;
42 /
PL/SQL procedure successfully completed.
```

```
관리자: Start Database - sqlplus db1710385/oracle
***** Insert 및 에러 처리 테스트 *****
#
1710385님이 과목번호 C800, 분반 2의 수강 등록을 요청하였습니다.
결과 : 이미 등록된 과목을 신청하였습니다
#
1710385님이 과목번호 C900, 분반 3의 수강 등록을 요청하였습니다.
결과 : 수강신청 인원이 초과되어 등록이 불가능합니다
#
1710385님이 과목번호 M600, 분반 3의 수강 등록을 요청하였습니다.
결과 : 이미 등록된 과목 중 중복되는 시간이 존재합니다
#
1710385님이 과목번호 M500, 분반 3의 수강 등록을 요청하였습니다.
결과 : 수강신청 등록이 완료되었습니다.
#
1710385님이 과목번호 S100, 분반 1의 수강 등록을 요청하였습니다.
결과 : 최대학점을 초과하였습니다
***** CURSOR를 이용한 SELECT 테스트 *****
#
1710385님의 2021년도 2학기의 수강신청 시간표입니다.
교시 : 1, 과목코드 : C800, 과목명 : 데이터베이스 프로그래밍, 분반 :3, 학점 : 3
교시 : 3, 과목코드 : M100, 과목명 : 선형대수, 분반 :3, 학점 : 3
교시 : 4, 과목코드 : M200, 과목명 : 그래프 활용, 분반 :3, 학점 : 3
교시 : 5, 과목코드 : M300, 과목명 : 윈도우 프로그래밍, 분반 :3, 학점 : 3
교시 : 6, 과목코드 : M400, 과목명 : 멀티미디어 처리, 분반 :3, 학점 : 3
교시 : 7, 과목코드 : M500, 과목명 : 게임 프로그래밍, 분반 :3, 학점 : 3
총 6과목과 총 18학점을 신청하였습니다.
PL/SQL procedure successfully completed.
SQL>
```

<insert_test.sql> 프로시저와 Outpoutline



수강신청과 수강신청 삭제 후 Enroll테이블 확인해보면 다음과 같다.

학번 '1710385'의 수강신청 목록에서 수업번호 'C800', '데이터베이스 프로그래밍' 수업이 삭제된것을 확인할 수 있고, 수업번호 'M500', '게임프로그래밍' 수업이 새로 신청된 것을 확인할 수 있다.

2) BeforeUpdateStudent 트리거 Test

BeforeUpdateStudent 트리거를 발생시켜 사용자 정보 수정시 발생할 수 있는 에러를 확인한다.

1. 암호의 길이가 4자리 미만인 경우

-> 에러번호는 -20002, 에러설명은 '암호는 4자리 이상이어야 합니다.' 오류 발생

2. 암호에 공란이 포함된 경우

-> 에러번호는 -20003, 에러설명은 '암호에 공란은 입력되지 않습니다.' 오류 발생

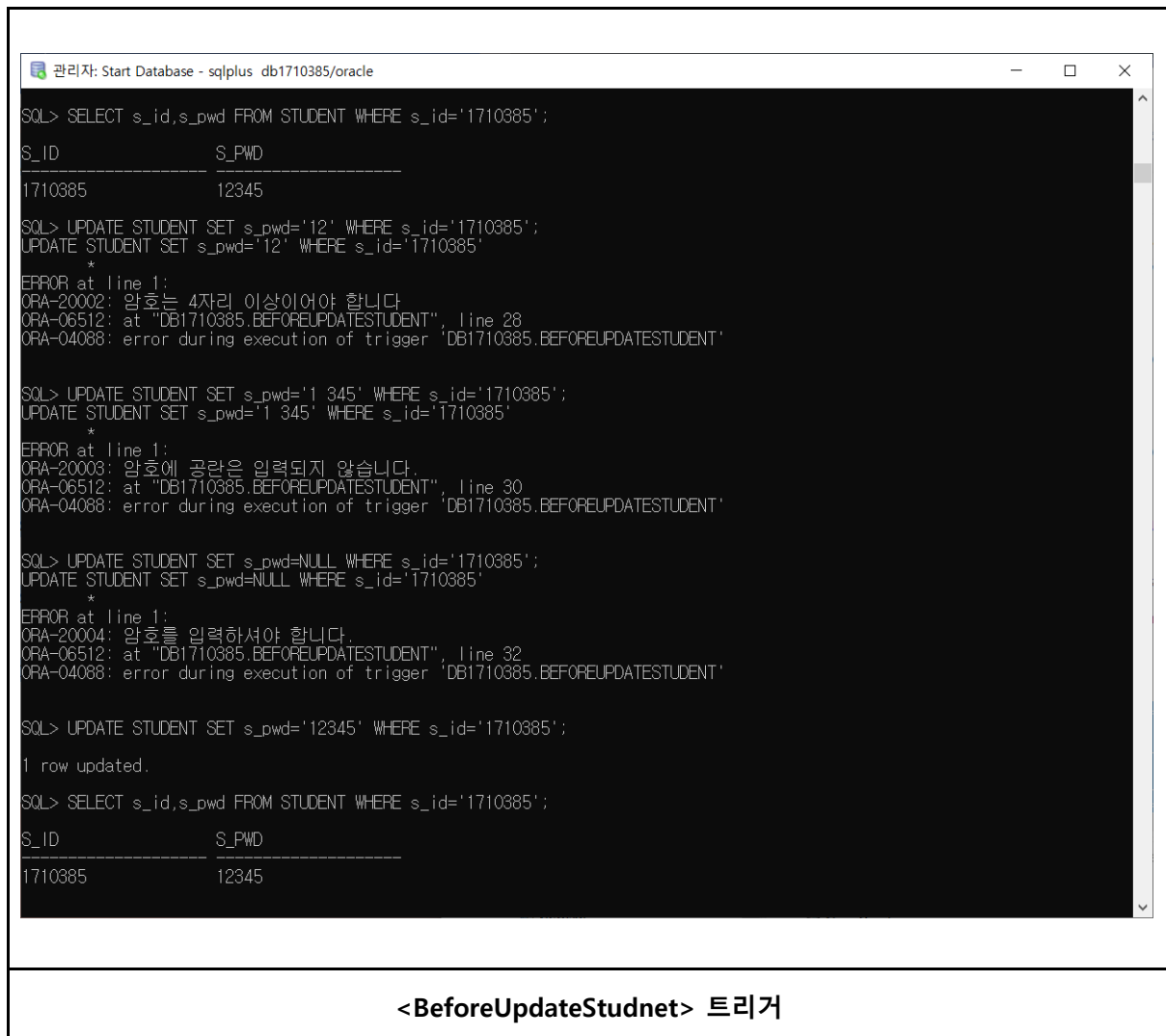
3. 암호가 입력되지 않은 경우

-> 에러번호는 -20004, 에러설명은 '암호를 입력하셔야 합니다.' 오류 발생

4. 암호가 정상적인 경우

-> 수정 완료

사용자 정보 수정 결과는 다음과 같다.



4자리 미만일때, 공란이 포함되어 있을때, NULL값일 때 s_pwd가 변경되지 않는 것을 확인할수 있다.

5. 기타(중요하다고 생각하는 부분 또는 새로 추가한 부분)

수강신청 인원에 관련 오류를 보여줄 때, 신청 한 후에 해당 과목의 인원이 초과된지 알려주는 것 보다는 신청하기 전에 신청한 인원이 몇 명인지를 보여주면 수강신청 할 때 더 편할 것 같다고 생각이 들어 수강신청 인원을 보여주는 변수와, 파라미터가 있는 커서를 추가하였다.

수강신청을 할 때 자신이 원하는 학과의 수업을 더 빠르고, 편리하게 찾을 수 있도록 view를 이용하여 학과를 검색할 수 있는 부분을 추가하였다.

6. 역할분담

학번 / 학과 / 이름	역할
1816461 / 통계학과 / 노세희	수강신청 입력 보고서 마무리
1714121 / 통계학과 / 이해원	수강신청 삭제 발표
1710385 / 통계학과 / 한하람	수강신청 조회 보고서 마무리
공통작업	테이블과 뷰 생성 프로시저, 함수, 트리거 등 sql문 작성과 테스트 보고서작성