**프로젝트 보고서**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 연도 | 2020년 | 학기 | | 1학기 |
| 과목명 | 데이터베이스프로그래밍 | 분반 | | 002분반 |
| 조원 | 1715333 김민정 | | insert.jsp  insert\_verify.jsp | |
| 1716821 설민지 | | select.jsp  show\_enroll.jsp | |
| 1714152 이명지 | | delete.jsp  delete\_verify.jsp | |
| 공동작성  테이블 / login.jsp / login\_verify.jsp / update.jsp / update\_verify.jsp | | | |

|  |
| --- |
| DB 설계도 |
| -ERDCourseEnrollmentSystem_20200616_57_55.png  ERD2.png  -RDB   |  |  |  |  | | --- | --- | --- | --- | | CREATE TABLE student(  s\_id VARCHAR2(10) NOT NULL,  s\_pwd VARCHAR2(30) NOT NULL,  s\_name VARCHAR2(30),  s\_major VARCHAR2(30),  CONSTRAINT s\_pk PRIMARY KEY(s\_id)  );  student | | | | | s\_id | s\_pwd | s\_name | s\_major | | 1400015 | adee | 김민정 | 컴퓨터과학전공 | | 1400016 | cxzvbe | 설민지 | 컴퓨터과학전공 | | 1400017 | zbrwe32 | 이명지 | 컴퓨터과학전공 | | 1500018 | vcczha5 | 한고은 | 컴퓨터과학전공 | | 1500019 | cvxczb1 | 이예은 | 컴퓨터과학전공 | | 1500020 | cxzvw1sfa | 박소현 | 컴퓨터과학전공 | | 1500021 | xcvxgad | 정서희 | 컴퓨터과학전공 | | 1600022 | czvdga | 조소은 | 컴퓨터과학전공 | | 1600023 | cvztea | 김지원 | 컴퓨터과학전공 | | 1600024 | cxvzehj | 어진아 | 컴퓨터과학전공 | | 1600025 | kghvcb | 최윤민 | 컴퓨터과학전공 | | 1700026 | gdjgdh | 이정인 | 컴퓨터과학전공 |  |  |  |  | | --- | --- | --- | | CREATE TABLE course(  c\_id VARCHAR2(10) NOT NULL,  c\_name VARCHAR2(50) NOT NULL,  c\_credit NUMBER(1),  CONSTRAINT c\_pk PRIMARY KEY(c\_id)  );  course | | | | c\_id | c\_name | c\_credit | | 21003917 | 소프트웨어의이해 | 3 | | 21002147 | 컴퓨터수학 | 3 | | 21002144 | 프로그래밍개론 | 3 | | 21000540 | 자료구조 | 3 | | 21001710 | 디지털논리회로 | 3 | | 21000542 | 운영체제 | 3 | | 21000545 | 프로그래밍언어론 | 3 | | 21000549 | 알고리즘 | 3 | | 21003184 | 데이터베이스프로그래밍 | 3 | | 21003751 | 컴퓨터특강 | 3 | | 21000557 | 자바프로그래밍 | 3 | | 21001711 | 컴퓨터구조 | 3 | | 21102904 | 데이터사이언스개론 | 3 | | 21003758 | 시스템종합설계 | 3 | | 21000550 | 인공지능 | 3 |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | CREATE TABLE teach(  c\_id VARCHAR2(10) NOT NULL,  c\_class NUMBER(1),  t\_year NUMBER(4),  t\_semester NUMBER(1),  t\_day VARCHAR2(20),  t\_time1 VARCHAR2(20),  t\_time2 VARCHAR2(20),  t\_location VARCHAR2(20),  p\_name VARCHAR2(10),  t\_max NUMBER(3),  CONSTRAINT t\_fk FOREIGN KEY(c\_id) REFERENCES course(c\_id)  );  teach | | | | | | | | | | | c\_id | c\_class | t\_year | t\_semester | t\_day | t\_time1 | t\_time2 | t\_location | p\_name | t\_max | | 21003917 | 1 | 2020 | 1 | 월,수 | 09:00 | 10:15 | 명신관413 | 이광수 | 30 | | 21002144 | 1 | 2020 | 1 | 화,목 | 13:00 | 14:50 | 명신관305 | 박영훈 | 30 | | 21000540 | 1 | 2020 | 1 | 월,수 | 10:30 | 11:45 | 명신관309 | 유석종 | 40 | | 21000540 | 2 | 2020 | 1 | 월,수 | 13:30 | 14:45 | 명신관309 | 유석종 | 40 | | 21000545 | 1 | 2020 | 1 | 화,목 | 10:30 | 11:45 | 명신관413 | 창병모 | 40 | | 21003184 | 1 | 2020 | 1 | 월,수 | 10:30 | 11:45 | 명신관408 | 심준호 | 30 | | 21003184 | 2 | 2020 | 1 | 월,수 | 13:30 | 14:45 | 명신관408 | 심준호 | 30 | | 21003751 | 1 | 2020 | 1 | 수,금 | 16:30 | 17:45 | 명신관309 | 김윤진 | 30 | | 21002147 | 1 | 2020 | 2 | 화,목 | 10:30 | 11:45 | 명신관523 | 최영우 | 30 | | 21002144 | 1 | 2020 | 2 | 화,목 | 11:00 | 12:50 | 명신관615 | 이현자 | 30 | | 21001710 | 1 | 2020 | 2 | 화,목 | 13:30 | 14:45 | 명신관413 | 김주균 | 40 | | 21000549 | 1 | 2020 | 2 | 월,수 | 09:00 | 10:15 | 프라임관203 | 박영훈 | 40 | | 20000549 | 2 | 2020 | 2 | 화,목 | 09:00 | 10:15 | 프라임관203 | 박영훈 | 40 | | 21000545 | 1 | 2020 | 2 | 화,목 | 13:30 | 14:45 | 명신관516 | 창병모 | 30 | | 21000542 | 1 | 2020 | 2 | 수,금 | 16:30 | 17:45 | 명신관523 | 김주균 | 1 | | 21003917 | 1 | 2020 | 2 | 화,목 | 15:00 | 16:15 | 명신관615 | 이광수 | 10 | | 21000557 | 1 | 2020 | 2 | 월,수 | 09:00 | 10:15 | 명신관305 | 박숙영 | 20 | | 21000557 | 2 | 2020 | 2 | 월,수 | 12:00 | 13:15 | 명신관305 | 김윤희 | 20 | | 21000540 | 1 | 2020 | 2 | 화,목 | 16:30 | 17:45 | 명신관103 | 유석종 | 10 | | 21001711 | 1 | 2020 | 2 | 월,수 | 15:00 | 16:15 | 명신관615 | 문봉희 | 20 | | 21102904 | 1 | 2020 | 2 | 월,수 | 09:00 | 10:15 | 명신관615 | 박동철 | 5 | | 21000550 | 1 | 2020 | 2 | 금 | 12:00 | 13:15 | 명신관103 | 채희준 | 20 | | 21003758 | 1 | 2020 | 2 | 월,수 | 13:30 | 14:45 | 명신관103 | 박숙영 | 20 |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | CREATE TABLE enroll(  s\_id VARCHAR2(10) NOT NULL,  c\_id VARCHAR2(10) NOT NULL,  c\_class NUMBER(1),  e\_year NUMBER(4),  e\_semester NUMBER(1),  c\_credit NUMBER(1),  CONSTRAINT e\_fk1 FOREIGN KEY(s\_id) REFERENCES student(s\_id),  CONSTRAINT e\_fk2 FOREIGN KEY(c\_id) REFERENCES course(c\_id)  );  enroll | | | | | | | s\_id | c\_id | c\_class | e\_year | e\_semester | c\_credit | | 1400015 | 21003917 | 1 | 2020 | 1 | 3 | | 1400015 | 21002144 | 1 | 2020 | 1 | 3 | | 1400015 | 21000540 | 1 | 2020 | 1 | 3 | | 1400016 | 21003917 | 1 | 2020 | 1 | 3 | | 1400016 | 21002144 | 1 | 2020 | 1 | 3 | | 1400016 | 21000540 | 1 | 2020 | 1 | 3 | | 1400017 | 21003917 | 1 | 2020 | 1 | 3 | | 1400017 | 21002144 | 1 | 2020 | 1 | 3 | | 1400017 | 21000540 | 1 | 2020 | 1 | 3 | | 1500018 | 21000550 | 1 | 2020 | 2 | 3 | | 1500018 | 21003758 | 1 | 2020 | 2 | 3 | |
| 사용된 주요 SQL 질의문과 프로그래밍 기법 (프로젝트에 반드시 포함되어야 할 내용) |
| 1)login\_verify.jsp  line 21~line 25:  stmt = conn.createStatement();  String SQL="select s\_id from student where s\_id='" + userID + "'and s\_pwd='" + userPassword + "'";  rs = stmt.executeQuery(SQL);  - student 테이블에서 학번과 비밀번호를 불러오는 statement문.  2-1)update.jsp  line 27~line 29:  stmt = conn.createStatement();  String SQL = "select \* from student where s\_id = '" + session\_id + "'";  rs = stmt.executeQuery(SQL);  - 사용자 정보 수정을 위해, 현재 로그인 된 사용자의 정보를 student 테이블에서 가져옴.  2-2)update\_verify.jsp  line 25~line27 :  pstmt = conn.prepareStatement("select \* from students where s\_id=? and s\_pwd=?");  - prepared statement문을 활용하여, 현재 로그인한 사용자와 현재 작성한 비밀번호가 포함된 student테이블에 있는 지 확인  line 32 :  pstmt = conn.prepareStatement("update student set s\_pwd=? where s\_id=?");  - 위의 레코드가 존재한다면, prepared statement문을 활용하여, 현재 사용자의 비밀번호를 변경  line 51:  catch(SQLException ex) {  String sMessage;  if (ex.getErrorCode() == 20002)  sMessage = "암호는 4자리 이상이어야 합니다.";  else if (ex.getErrorCode() == 20003)  sMessage = "암호에 공란은 입력되지 않습니다.";  else  sMessage = "잠시 후 다시 시도하십시오.";  - 비밀번호 변경 시 BeforeUpdateStudent Trigger를 일으킴.  BeforeUpdateStudent Trigger  CREATE OR REPLACE TRIGGER BeforeUpdateStudent  BEFORE  UPDATE ON student  FOR EACH ROW  DECLARE  underflow\_length EXCEPTION;  invalid\_value EXCEPTION;  BEGIN  IF LENGTH( NVL(:new.s\_pwd, 0) ) < 4 THEN  RAISE underflow\_length;  END IF;  IF (INSTR(:new.s\_pwd, ' ') > 0) THEN  RAISE invalid\_value;  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');  END;  /  -오라클 NVL함수를 사용하여 생성할 비밀번호의 길이가 4 미만이거나, 공백이 포함된 경우 예외처리를 한다.  3-1)insert.jsp  line 37~line 42:  cstmt1 = myConn.prepareCall("{? = call Date2EnrollYear(SYSDATE)}");  cstmt2 = myConn.prepareCall("{? = call Date2EnrollSemester(SYSDATE)}");  cstmt1.registerOutParameter(1,java.sql.Types.INTEGER);  cstmt2.registerOutParameter(1,java.sql.Types.INTEGER);  cstmt1.execute();  cstmt2.execute();  - CallableStatement와 Stored Function인 Date2EnrollYear, Date2EnrollSemester로 현재 날짜를 받아 수강신청 년도와 학기 반환하기.  line 47~line 52:  String SQL= "select c.c\_id, t.c\_class, c.c\_name, c.c\_credit, t.t\_day, t.t\_time1, t.t\_time2, t.t\_location, t.p\_name, t.t\_max from course c, teach t where t\_year=? and t\_semester=? and t.c\_id =c.c\_id and (t.c\_id, t.c\_class) not in (select c\_id, c\_class from enroll where s\_id=?)";    pstmt =myConn.prepareStatement (SQL);  pstmt.setInt(1, year);  pstmt.setInt(2, semester);  pstmt.setString(3, session\_id);  - Prepared Statement와 중첩질의를 이용하여 강의 정보 받아오기.  line 71~line 78:  mysql ="select st\_num from enroll\_student e, teach t where e.c\_id = ? and e.c\_class = ? and e.e\_year=? and e.e\_semester = ?";  pstmt2 = myConn.prepareStatement(mysql);  pstmt2.setString(1, c\_id);  pstmt2.setInt(2, c\_class);  pstmt2.setInt(3, year);  pstmt2.setInt(4, semester);    rs = pstmt2.executeQuery();  - Prepared statement문과 view enroll\_student로 총 신청 학생 수 받아오기.  View enroll\_student  CREATE OR REPLACE VIEW enroll\_student  AS  SELECT e.c\_id, e.c\_class, e.e\_year, e.e\_semester, COUNT(e.s\_id) AS st\_num  FROM enroll e, teach t  WHERE e.c\_id = t.c\_id AND t.t\_year = e.e\_year AND t.t\_semester = e.e\_semester AND e.c\_class = t.c\_class  GROUP BY e.c\_id, e.c\_class, e.e\_year, e.e\_semester;  -해당 과목번호와 같은 강의를 수강신청한 학생들의 수를 count하는 enroll\_student view.  3-2)insert\_verify.jsp  line 28~line 32:  CallableStatement cstmt = myConn.prepareCall("{ call InsertEnroll(?, ?, ?, ?) }");  cstmt.setString(1, s\_id);  cstmt.setString(2, c\_id);  cstmt.setInt(3, c\_class);  cstmt.registerOutParameter(4, java.sql.Types.VARCHAR);  - Callable Statement문과 stored procedure InsertEnroll로 수강신청한 과목을 enroll 테이블에 넣어주기.  Stored Procedure InsertEnroll  CREATE OR REPLACE PROCEDURE InsertEnroll(sStudentId IN VARCHAR2,  sCourseId IN VARCHAR2,  nCourseIdNo IN NUMBER,  result OUT VARCHAR2)  IS  too\_many\_sumCourseUnit EXCEPTION;  too\_many\_courses EXCEPTION;  too\_many\_students EXCEPTION;  duplicate\_time EXCEPTION;  nYear NUMBER;  nSemester NUMBER;  nSumCourseUnit NUMBER;  v\_course course%ROWTYPE;  nSame NUMBER:=0;  nSameT NUMBER:=0;  nCnt NUMBER:=0;  nTeachMax teach.t\_max%TYPE;  nowday teach.t\_day%TYPE;  time1 DATE;  time2 DATE;  maxTime DATE;  minTime DATE;  i NUMBER;  leng1 number :=0;  day1 varchar2(12);  flag BOOLEAN NOT NULL:=false;  CURSOR time\_enroll(v\_sid enroll.s\_id%TYPE, v\_year enroll.e\_year%TYPE, v\_semester enroll.e\_semester%TYPE) IS  select t.t\_time1, t.t\_time2, t.t\_day  from enroll e, teach t  where e.s\_id=v\_sid AND e\_year=v\_year AND e\_year=t\_year AND e\_semester = v\_semester AND e\_semester = t\_semester AND  e.c\_id = t.c\_id AND e.c\_class = t.c\_class;  BEGIN  result := '';  DBMS\_OUTPUT.put\_line('#');  DBMS\_OUTPUT.put\_line(sStudentId || '님이 과목번호 ' || sCourseId ||  ', 분반 ' || TO\_CHAR(nCourseIdNo) || '의 수강 등록을 요청하였습니다.');  nYear := Date2EnrollYear(SYSDATE);  nSemester := Date2EnrollSemester(SYSDATE);  SELECT SUM(e.c\_credit)  INTO nSumCourseUnit  FROM enroll e  WHERE e.s\_id = sStudentId and e.e\_year = nYear and  e.e\_semester = nSemester;  SELECT c\_credit  INTO v\_course.c\_credit  FROM course  WHERE c\_id = sCourseId;  IF (nSumCourseUnit + v\_course.c\_credit > 18) THEN  RAISE too\_many\_sumCourseUnit;  END IF;  SELECT COUNT(\*)  INTO nSame  FROM enroll  WHERE s\_id = sStudentId and c\_id = sCourseId;  IF (nSame > 0) THEN  RAISE too\_many\_courses;  END IF;  SELECT t\_max  INTO nTeachMax  FROM teach  WHERE t\_year= nYear and t\_semester = nSemester  and c\_id = sCourseId and c\_class= nCourseIdNo;  SELECT COUNT(\*)  INTO nSameT  FROM enroll  WHERE e\_year = nYear and e\_semester = nSemester  and c\_id = sCourseId and c\_class = nCourseIdNo;  IF (nSameT >= nTeachMax) THEN  RAISE too\_many\_students;  END IF;  SELECT to\_date(t\_time1, 'HH24:MI'), to\_date(t\_time2, 'HH24:MI'), t\_day  INTO time1, time2, nowday  FROM teach  WHERE t\_year=nYear and t\_semester = nSemester and  c\_id = sCourseId and c\_class = nCourseIdNo;  nowday:=replace(nowday, ',', '');  leng1:=length(nowday);  FOR t\_e IN time\_enroll(sStudentId, nYear, nSemester) LOOP  i:=1;  flag:=false;  WHILE i<= leng1 LOOP  day1:=substr(nowday, i, 1);  IF instr(t\_e.t\_day, day1)>0 THEN  flag:=true;  END IF;  EXIT WHEN flag=true;  i:=i+1;  END LOOP;  IF flag= true THEN  maxTime := to\_date(t\_e.t\_time1, 'HH24:MI');  minTime := to\_date(t\_e.t\_time2, 'HH24:MI');  IF (time1>maxTime ) THEN  maxTime :=time1;  END IF;  IF (time2<minTime ) THEN  minTime :=time2;  END IF;  IF ((maxTime-minTime)\*60\*24 < 0) THEN  nCnt:=nCnt+1;  END IF;  END IF;  END LOOP;  IF (nCnt > 0) THEN  RAISE duplicate\_time;  END IF;  INSERT INTO enroll (S\_ID, C\_ID, C\_CLASS, E\_YEAR, E\_SEMESTER, C\_CREDIT)  VALUES (sStudentId, sCourseId, nCourseIdNo, nYear, nSemester, v\_course.c\_credit);  COMMIT;  result := '수강신청 등록이 완료되었습니다.';  EXCEPTION  WHEN too\_many\_sumCourseUnit THEN  result := '최대학점을 초과하였습니다.';  WHEN too\_many\_courses THEN  result := '이미 등록된 과목을 신청하였습니다.';  WHEN too\_many\_students THEN  result := '수강신청 인원이 초과되어 등록이 불가능합니다.';  WHEN duplicate\_time THEN  result := '이미 등록된 과목 중 중복되는 시간이 존재합니다.';  WHEN OTHERS THEN  ROLLBACK;  result := SQLCODE || ' ' || SQLERRM;  END;  /  - 1: 총 신청한 학점이 최대 학점을 초과하지 않았는지 2: 같은 과목을 신청하지 않았는지 3: 수강인원이 초과되지 않았는지 4: 신청한 강의 중 시간이 겹치는 강의가 있지 않은지 확인한 후 정상적인 경우 enroll 테이블에 수강신청한 강의를 추가하는 프로시저. length(), replace(), substr(), instr() 등 오라클 함수가 쓰임. 변수 선언에 %TYPE, %ROWTYPE 쓰임. 조건문, 반복문, 예외처리 외에도 for문에서 파라미터 있는 커서 쓰임.  4-1)delete.jsp  line 30~line 40:  String func = "{? = call Date2EnrollYear(SYSDATE)}";  cstmt = conn.prepareCall(func);  cstmt.registerOutParameter(1,java.sql.Types.INTEGER);  cstmt.execute();  int year = cstmt.getInt(1);  String func2 = "{? = call Date2EnrollSemester(SYSDATE)}";  cstmt2 = conn.prepareCall(func2);  cstmt2.registerOutParameter(1,java.sql.Types.INTEGER);  cstmt2.execute();  int semester = cstmt2.getInt(1);  - 현재 날짜를 받아 년도와 학기를 반환하는 함수 Date2EnrollYear와 Date2EnrollSemester를 CallableStatement를 이용해 사용.  line 45:  String SQL="select c\_id,c\_class from enroll where s\_id='" + session\_id + "' and e\_year=" + year + " and e\_semester=" + semester;  - statement문을 이용해 enroll 테이블에서 학번, 년도, 학기에 해당하는 강의 정보(과목번호, 분반) 가져오기.  line 56:  String SQL2 = "select c.c\_name, t.t\_day,t.t\_time1, t.t\_time2,t.t\_location, c.c\_credit,t.p\_name from course c,teach t where t.t\_year=" + year + " and t.t\_semester=" + semester + " and c.c\_id = '" + c\_id + "' and c.c\_id = t.c\_id and t.c\_class ="+ c\_class;  - statement문을 이용해 위 enroll 테이블에서 가져온 강의의 정보(과목명, 강의시간, 강의실, 담당교수) 가져오기.  line 90~line 100:  cstmt3 = conn.prepareCall("{call SumTable(?, ?, ?, ?, ?)}");  cstmt3.setString(1, session\_id);  cstmt3.setInt(2, year);  cstmt3.setInt(3, semester);  cstmt3.registerOutParameter(4, java.sql.Types.INTEGER);  cstmt3.registerOutParameter(5, java.sql.Types.INTEGER);  cstmt3.execute();    int classSum = cstmt3.getInt(4);  int creditSum = cstmt3.getInt(5);  - CallableStatement를 이용해 파라미터가 있는 프로시저 SumTable로 총 신청과목수와 총 학점 받아오기.  Stored Function  CREATE OR REPLACE FUNCTION Date2EnrollYear(eDate IN DATE)  RETURN NUMBER  IS  yearNow NUMBER;  monthNow NUMBER;  year NUMBER;  BEGIN  SELECT to\_number(to\_char(eDate, 'YYYY')) INTO yearNow FROM DUAL;  SELECT to\_number(to\_char(eDate, 'MM')) INTO monthNow FROM DUAL;  IF (monthNow = 11 OR monthNow = 12) THEN  year := yearNow + 1;  ELSE  year := yearNow;  END IF;  Return year;  END;  /  CREATE OR REPLACE FUNCTION Date2EnrollSemester(eDate IN DATE)  RETURN NUMBER  IS  monthNow NUMBER;  semester NUMBER;  BEGIN  SELECT to\_number(to\_char(eDate, 'MM')) INTO monthNow FROM DUAL;  IF (monthNow >= 5 and monthNow <= 10) THEN  semester := 2;  ELSE  semester := 1;  END IF;  Return semester;  END;  /  - 파라미터로 날짜를 받아 그 날짜에 해당하는 수강신청 년도와 학기를 반환하는 stored function.  4-2)delete\_verify.jsp  line 23~line 24:  String SQL = "DELETE FROM enroll WHERE s\_id='"+s\_id+"' and c\_id='"+c\_id+"' and c\_class=" + c\_class + " and e\_year="+year+"and e\_semester="+semester;  int resultSet = stmt.executeUpdate(SQL);  - delete.jsp에서 취소할 강의의 과목번호와 분반을 받아온 후, 과목번호와 분반, 현재 년도, 학기에 해당하는 강의 enroll 테이블에서 삭제하는 Statement문.  5)show\_enroll.jsp  line 49 :  String SQL = "select c\_id, c\_class from enroll where s\_id='" + session\_id + "' and e\_year='" + selec\_year + "' and e\_semester='" + selec\_semester + "'";  -statement문으로 현재 select box에서 선택된 연도,학기에 해당하는 사용자의 수강 정보를 enroll테이블로부터 받아옴.  line 64 :  String SQL2 = "select c.c\_id c\_id,c.c\_name c\_name, t.c\_class c\_class,t.t\_day t\_day,t.t\_time1 t\_time1, t.t\_time2 t\_time2,t.t\_location t\_location,c.c\_credit c\_credit,t.p\_name p\_name from course c,teach t where t.t\_year='" + selec\_year + "' and t.t\_semester='" + selec\_semester + "' and c.c\_id = '" + c\_id + "' and c.c\_id = t.c\_id and t.c\_class = " + c\_class;  -statement문으로 enroll테이블에서 얻어온 사용자의 수강 기본정보들의 값을 통해, teach테이블에서 강의 상세정보 레코드를 가져옴.  line 101 :  cstmt = Conn.prepareCall("{ call SumTable(?, ?, ?, ?, ?) }");  -총 신청 과목 및 총 신청 학점을 보여주기 위해, Callable Statement 를 활용하여, 오라클에서 정의한 SumTable procedure를 호출한다.  SumTable procedure  CREATE OR REPLACE PROCEDURE SumTable  (sStudentId IN VARCHAR2,  nYear IN NUMBER,  nSemester IN NUMBER,  nClass OUT NUMBER,  nUnit OUT NUMBER )  IS  v\_unit enroll.c\_credit%TYPE;  unitnum NUMBER;  CURSOR verify\_sum IS  SELECT e.c\_credit  FROM enroll e, teach t  WHERE e.s\_id=sStudentId AND e.e\_year=nYear AND e.e\_year=t.t\_year AND e.e\_semester=nSemester AND e.e\_semester=t.t\_semester AND  e.c\_id =t.c\_id AND e.c\_class =t.c\_class;  BEGIN  unitnum:=0;  OPEN verify\_sum;  LOOP  FETCH verify\_sum INTO v\_unit;  EXIT WHEN verify\_sum%NOTFOUND;  unitnum:=v\_unit+unitnum;  END LOOP;  nClass := verify\_sum%ROWCOUNT;  nUnit :=unitnum;  CLOSE verify\_sum;  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('에러 발생: ' || SQLERRM);  END;  /  - SumTable procedure는 LOOP문에서 명시적 커서 verify\_sum을 사용하여, enroll, teach들을 join해서 얻은 c\_credit를 통해 총 신청 학점 및 총 신청 과목을 계산한다. |
| 웹화면 구성과 jsp 소스 파일 관계 |
| 1) 로그인    main화면에서 ‘로그인’을 누르면 로그인 화면 페이지를 띄워준다. 아이디, 패스워드 정보를 입력하면 login\_verify.jsp에서 아이디, 패스워드가 맞는지 확인하고 맞으면 세션에 아이디 값을 저장한 뒤 main화면으로 다시 넘어가고, 틀리면 alert 창을 띄운다. 로그아웃 버튼을 누르면 logout.jsp가 실행되어 세션에 있는 로그인 정보를 무효화시켜 로그아웃한다.    아이디 또는 비밀번호 틀렸을 경우.    로그인 성공한 화면.  2) 사용자 정보 수정    -해당 페이지는 update.jsp로 구현하였고 top.jsp를 포함하고 있으며, db로부터 받아온 사용자의 이름, 학번, 학과를 웹페이지에 보여준다. 이후, 현재 사용자의 비밀번호를 올바르게 입력하고, 변경할 비밀번호를 제약조건에 맞춰 입력한 후, 비밀번호 변경 버튼을 누르면 update\_verify.jps에서 비밀번호에 대한 검증을 거쳐 다음과 같이 정상적으로 비밀번호 변경됨을 알려준다.    현재 비밀번호를 올바르게 입력하지 않거나, 변경할 비밀번호가 4자리 미만이거나, 공백이 포함되었을 경우 다음과 같은 경고 메시지를 보여주며, 비밀번호는 변경되지 않는다.        3) 수강신청    insert.jsp로 구현한 수강신청 페이지. ‘수강신청 입력’을 누르면 수강신청할 수 있는 창이 뜬다. 과목번호, 분반, 과목명 등 수강신청에 필요한 정보들을 보여준다. 현재 로그인한 학생이 신청하지 않은 과목을 보여준다. enroll\_student 뷰를 사용해서 여석이 얼마나 남았는지 확인할 수 있다. 과목의 신청 버튼을 누르면 신청한 과목번호와 분반 값이 insert\_verify.jsp로 전달된다. insert\_verify.jsp에서는 InsertEnroll 프로시저를 호출해서 최대학점을 초과하지 않는지, 이미 등록된 과목을 또 신청하지 않았는지, 해당 과목의 여석이 남아있는지, 이미 신청한 다른 과목들과 시간이 겹치지 않는지 검사하고 아무런 오류도 없다면 수강신청을 완료했다는 알림창을 띄우고, 조건 중 어느 하나라도 만족하지 않았다면 만족하지 않은 조건의 오류메세지를 띄운다.  E-1) 최대학점 초과여부    E-2) 동일한 과목 신청 여부    E-3) 수강신청 인원 초과 여부    E-4) 신청한 과목들 시간 중복 여부    수강 신청 등록 성공    4) 수강신청 취소    delete.jsp로 구성한 수강신청 취소 페이지. top.jsp를 포함하고 있다. 신청한 강의가 없는 경우,신청한 강의가 없음을 알리며 시간표가 뜨지 않는다.    현재 신청한 강의가 있는 경우, delete.jsp에서 해당 학생이 신청한 강의들을 보여준다. 각 강의마다 과목번호, 과목명, 분반, 강의시간, 강의실, 담당교수, 학점의 정보를 볼 수 있다. 취소하고 싶은 강의의 취소 버튼을 누르면 delete\_verify.jsp에서 해당 강의의 정보를 받아 다음과 같이 정상적으로 수강신청이 취소됨을 알려주며 테이블에서 해당 강의를 삭제한다.    정상적으로 수강취소되었다.    알고리즘 과목이 정상적으로 취소된 모습.  5) 수강신청 조회    - 해당 페이지는 select.jsp로 구현되었으며, select.jsp는 top.jsp를 포함하고 있다. 중앙에 위치한 select box를 통해 원하는 연도와 학기를 검색할 수 있다. 원하는 연도와 학기를 선택 후 “검색” 버튼을 누르면, show\_enroll.jsp를 통해 다음과 같은 화면을 보여준다.    show\_enroll.jsp는 select.jsp를 포함하고 있으며, 사용자가 해당 연도, 학기에 수강 신청한 과목 번호, 과목 이름, 분반, 요일, 시간, 장소 ,교수님, 학점을 조회 할 수 있다. 하단에는 해당 연도, 학기에 총 신청한 과목 개수와, 학점을 정리하여 보여준다. |
| 기타(중요하다고 생각하는 내용) |
| 김민정:  - 구현하고자 하는 기능과 조건에 맞게 PLSQL, 예외처리, 트리거, 프로시저, 함수, 커서 등을 적절히 이용하는 것.  - 원하는 조건을 만족하는 조인, 조건문을 제대로 사용하는 것.  - jsp와 oracle이 제대로 연동되도록 driver, 소스코드를 체크하는 것.  설민지:  - 합리적인 DB schema 설계  - PLSQL, 오라클 함수 등 다양한 기법을 사용하여 수강신청 시스템 구현  - 사용자 사용하기 편리한 웹 디자인 및 기능 구현  이명지:  - 실제 수강신청 시스템과 얼마나 비슷하게 구현했는지.  - 다른 사람이 봤을 때, 보기 쉬우며 해석하기 쉽고, 실용적인 코드를 작성하는 것.  - 역할 분담을 치우치지 않게 하는 것과 열심히 참여하려는 협동심, |
| 소스코드 |
| 1-1)main.jsp  <%@ page language=*"java"* contentType=*"text/html; charset=EUC-KR"* pageEncoding=*"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>  <td align=*"center"*><%=session\_id%>님 방문을 환영합니다.</td>  </tr>  <% } else { %>  <tr>  <td align=*"center"*>로그인한 후 사용하세요.</td>  </tr>  <%  }  %>  </table>  </body>  </html>  1-2)top.jsp  <%@ page language=*"java"* contentType=*"text/html; charset=EUC-KR"*  pageEncoding=*"EUC-KR"*%>  <%  String session\_id = (String) session.getAttribute("session\_id");  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</b> </td>  <td align=*"center"*> <b> <a href=*"insert.jsp"*>수강신청 입력</b> </td>  <td align=*"center"*> <b> <a href=*"delete.jsp"*>수강신청 삭제</b> </td>  <td align=*"center"*> <b> <a href=*"select.jsp"*>수강신청 조회</b> </td>  </tr>  </table>  2-1)login.jsp  <%@ page language=*"java"* contentType=*"text/html; charset=EUC-KR"* pageEncoding=*"UTF-8"*%>  <!DOCTYPE html>  <html>  <head> <meta charset=*"UTF-8"*> <title>수강신청 시스템 로그인</title> </head>  <body>  <table width=*"75%"* align=*"center"* bgcolor=*"#FFFF99"* border>  <tr> <td><div align=*"center"*>아이디와 패스워드를 입력하세요 </div></td></table>  <table width=*"75%"* 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>  </form>  </table>  </body> </html>  2-2)login\_verify.jsp  <%@ page language=*"java"* contentType=*"text/html; charset=EUC-KR"* pageEncoding=*"UTF-8"*%>  <%@page import=*"java.sql.\*"*%>  <%  String userID=request.getParameter("userID");  String userPassword=request.getParameter("userPassword");  String dbdriver = "oracle.jdbc.driver.OracleDriver";  String dburl = "jdbc:oracle:thin:@localhost:1521:orcl";  String user = "db1715333";  String passwd = "ss2";  Connection conn = null;  Statement stmt = null;  ResultSet rs = null;  try {  Class.forName(dbdriver);  conn=DriverManager.getConnection(dburl, user, passwd);  stmt = conn.createStatement();  String SQL="select s\_id from student where s\_id='" + userID + "'and s\_pwd='" + userPassword + "'";  rs = stmt.executeQuery(SQL);  if (rs.next()) {  session.setAttribute("session\_id", userID);  response.sendRedirect("main.jsp");  } else {  %>  <script>  alert("사용자 아이디 혹은 비밀번호가 틀렸습니다. ");  location.href="login.jsp";  </script>  <%  }    } catch (ClassNotFoundException e) {  System.out.println("jdbc driver 오류");  } catch (SQLException e) {  System.out.println("오라클 오류");  } finally {  if (rs != null) try { rs.close(); } catch(SQLException ex) {}  if (stmt != null) try { stmt.close(); } catch(SQLException ex) {}  if (conn != null) try { conn.close(); } catch(SQLException ex) {}  }  %>  2-3)logout.jsp  <%@ page contentType=*"text/html; charset=EUC-KR"* %>  <% session.invalidate(); %>  <script>  alert("로그아웃 되었습니다.");  location.href="main.jsp";  </script>  3-1)update.jsp  <%@ page contentType=*"text/html; charset=EUC-KR"* %>  <%@ 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"); %>  <table width=*"75%"* align=*"center"* border>  <%  String dbdriver = "oracle.jdbc.driver.OracleDriver";  String dburl = "jdbc:oracle:thin:@localhost:1521:orcl";  String user = "db1715333";  String passwd = "ss2";  Connection conn = null;  Statement stmt = null;  ResultSet rs = null;  try {  Class.forName(dbdriver);  conn=DriverManager.getConnection(dburl, user, passwd);  stmt = conn.createStatement();  String SQL = "select \* from student where s\_id = '" + session\_id + "'";  rs = stmt.executeQuery(SQL);    if(rs.next()){  %>  <tr>  <td><div align=*"center"*>이름</div></td>  <td><div><%=rs.getString("s\_name") %></div></td>  </tr>  <tr>  <td><div align=*"center"*>학번</div></td>  <td><div><%=rs.getString("s\_id") %></div></td>  </tr>  <tr>  <td><div align=*"center"*>학과</div></td>  <td><div><%=rs.getString("s\_major") %></div></td>  </tr>  <%  }  } catch (ClassNotFoundException e) {  System.out.println("jdbc driver 오류");  } catch (SQLException e) {  System.out.println("오라클 오류");  } finally {  if (rs != null) try { rs.close(); } catch(SQLException ex) {}  if (stmt != null) try { stmt.close(); } catch(SQLException ex) {}  if (conn != null) try { conn.close(); } catch(SQLException ex) {}  }  %>  <form method="post" action="update\_verify.jsp">  <tr>  <td><div align="center">현재 비밀번호</div></td>  <td><div align="left">  <input type="password" name="userPassword">  </div></td>  </tr>  <tr>  <td><div align="center">새 비밀번호</div></td>  <td><div align="left">  <input type="password" name="newPassword">  </div></td>  </tr>  <tr>  <td colspan=2><div align="center">  <INPUT TYPE="SUBMIT" NAME="Submit" VALUE="비밀번호 변경">  <INPUT TYPE="RESET" VALUE="취소">  </div></td>  </tr>  </form>  </table>  </body></html>  3-2)update\_verify.jsp  <%@ page contentType="text/html; charset=EUC-KR" %>  <%@ page import="java.sql.\*" %>  <html>  <head><title> 수강신청 사용자 정보 수정 </title></head>  <body>  <%  String s\_id = (String) session.getAttribute("session\_id");  String userPassword = request.getParameter("userPassword");  String newPassword = request.getParameter("newPassword");  String dbdriver = "oracle.jdbc.driver.OracleDriver";  String dburl = "jdbc:oracle:thin:@localhost:1521:orcl";  String user = "db1715333";  String passwd = "ss2";  Connection conn = null;  PreparedStatement pstmt = null;  PreparedStatement pstmt2 = null;  ResultSet rs = null;  try{  Class.forName(dbdriver);  conn=DriverManager.getConnection(dburl, user, passwd);    pstmt = conn.prepareStatement("select \* from student where s\_id=? and s\_pwd=?");  pstmt.setString(1, s\_id);  pstmt.setString(2, userPassword);  rs = pstmt.executeQuery();    if(rs.next()){  pstmt2 = conn.prepareStatement("update student set s\_pwd=? where s\_id=?");  pstmt2.setString(1, newPassword);  pstmt2.setString(2, s\_id);  pstmt2.executeUpdate();  %>  <script>  alert("비밀번호가 변경되었습니다. ");  location.href="update.jsp";  </script>  <%  }  else{  %>  <script>  alert("비밀번호가 틀렸습니다. ");  location.href="update.jsp";  </script>  <%  }  } catch(SQLException ex) {  String sMessage;  if (ex.getErrorCode() == 20002)  sMessage = "암호는 4자리 이상이어야 합니다.";  else if (ex.getErrorCode() == 20003)  sMessage = "암호에 공란은 입력되지 않습니다.";  else  sMessage = "잠시 후 다시 시도하십시오.";  %>  <script>  alert("<%=sMessage%>" );  history.back();  </script>  <%  } finally {  if (rs != null) try { rs.close(); } catch(SQLException ex) {}  if (pstmt != null) try { pstmt.close(); } catch(SQLException ex) {}  if (pstmt2 != null) try { pstmt2.close(); } catch(SQLException ex) {}  if (conn != null) try { conn.close(); } catch(SQLException ex) {}  }  %>  </body></html>  4-1)insert.jsp  <%@ page language=*"java"* contentType=*"text/html; charset=EUC-KR"*  pageEncoding=*"EUC-KR"*%>  <%@ page import=*"java.sql.\*"* %>  <html>  <head><title>수강신청 입력</title></head>  <body>  <%@ include file=*"top.jsp"*%>  <% if (session\_id == null) response.sendRedirect("login.jsp"); %>    <table width=*"75%"* align=*"center"* border>  <br><tr>  <th>과목번호</th><th>분반</th><th>과목명</th><th>학점</th><th>요일</th>  <th>시간</th><th>장소</th><th>담당교수</th><th>최대수강인원</th><th>여석</th><th>수강신청</th>  </tr>  <%  Connection myConn = null;  PreparedStatement pstmt = null;  CallableStatement cstmt1 = null;  CallableStatement cstmt2 = null;  ResultSet myResultSet = null;  String dbdriver = "oracle.jdbc.driver.OracleDriver";  String dburl = "jdbc:oracle:thin:@localhost:1521:orcl";  String user = "db1715333";  String password = "ss2";    PreparedStatement pstmt2 = null;  ResultSet rs=null;  String mysql=null;    int year=0; int semester=0;    try {  Class.forName(dbdriver);  myConn = DriverManager.getConnection(dburl, user, password);    cstmt1 = myConn.prepareCall("{? = call Date2EnrollYear(SYSDATE)}");  cstmt2 = myConn.prepareCall("{? = call Date2EnrollSemester(SYSDATE)}");  cstmt1.registerOutParameter(1,java.sql.Types.INTEGER);  cstmt2.registerOutParameter(1,java.sql.Types.INTEGER);  cstmt1.execute();  cstmt2.execute();    year = cstmt1.getInt(1);  semester = cstmt2.getInt(1);    String SQL= "select c.c\_id, t.c\_class, c.c\_name, c.c\_credit, t.t\_day, t.t\_time1, t.t\_time2, t.t\_location, t.p\_name, t.t\_max from course c, teach t where t\_year=? and t\_semester=? and t.c\_id =c.c\_id and (t.c\_id, t.c\_class) not in (select c\_id, c\_class from enroll where s\_id=?)";    pstmt =myConn.prepareStatement (SQL);  pstmt.setInt(1, year);  pstmt.setInt(2, semester);  pstmt.setString(3, session\_id);    myResultSet = pstmt.executeQuery();  if (myResultSet != null) {  while (myResultSet.next()) {  int st\_num = 0;  String c\_id = myResultSet.getString("c\_id");  int c\_class = myResultSet.getInt("c\_class");  String c\_name = myResultSet.getString("c\_name");  int c\_credit = myResultSet.getInt("c\_credit");  String t\_day = myResultSet.getString("t\_day");  String time1 = myResultSet.getString("t\_time1");  String time2 = myResultSet.getString("t\_time2");  String t\_location = myResultSet.getString("t\_location");  String p\_name = myResultSet.getString("p\_name");  int t\_max = myResultSet.getInt("t\_max");    //여석 가져오기  mysql ="select st\_num from enroll\_student e, teach t where e.c\_id = ? and e.c\_class = ? and e.e\_year=? and e.e\_semester = ?";  pstmt2 = myConn.prepareStatement(mysql);  pstmt2.setString(1, c\_id);  pstmt2.setInt(2, c\_class);  pstmt2.setInt(3, year);  pstmt2.setInt(4, semester);    rs = pstmt2.executeQuery();    if(rs.next() != false){  st\_num = rs.getInt("st\_num");  }  %>  <tr>  <td align=*"center"*><%=c\_id%></td>  <td align=*"center"*><%=c\_class%></td>  <td align=*"center"*><%=c\_name%></td>  <td align=*"center"*><%=c\_credit%></td>  <td align=*"center"*><%=t\_day%></td>  <td align=*"center"*><%=time1+"~"+time2%></td>  <td align=*"center"*><%=t\_location%></td>  <td align=*"center"*><%=p\_name%></td>  <td align=*"center"*><%=t\_max%></td>  <td align=*"center"*><%=t\_max-st\_num%></td>  <td align=*"center"*><a href=*"insert\_verify.jsp?c\_id=*<%=c\_id%>*&c\_class=*<%=c\_class%>*"*>신청</a></td>  </tr>  <%  }  }  } catch (SQLException ex) {  System.err.println("SQLException: " + ex.getMessage());  }    finally {  if (myResultSet != null) myResultSet.close();  if (pstmt != null) pstmt.close();  if (cstmt1 != null) cstmt1.close();  if (cstmt2 != null) cstmt2.close();  if (rs != null) rs.close();  if (pstmt2 != null) pstmt2.close();  if (myConn != null) myConn.close();  }  %>  </table>  </body>  </html>  4-2)insert\_verify.jsp  <%@ page language=*"java"* contentType=*"text/html; charset=EUC-KR"*  pageEncoding=*"EUC-KR"*%>  <%@ page import=*"java.sql.\*"* %>  <html><head><title> 수강신청 입력 </title></head>  <body>  <%  String s\_id = (String)session.getAttribute("session\_id");  String c\_id = request.getParameter("c\_id");  int c\_class = Integer.parseInt(request.getParameter("c\_class"));  %>  <%  Connection myConn = null; String result = null;  String dbdriver = "oracle.jdbc.driver.OracleDriver";  String dburl = "jdbc:oracle:thin:@localhost:1521:orcl";  String user = "db1715333";  String password = "ss2";  try {  Class.forName(dbdriver);  myConn = DriverManager.getConnection (dburl, user, password);  myConn.setAutoCommit(false);  } 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\_class);  cstmt.registerOutParameter(4, java.sql.Types.VARCHAR);  try {  cstmt.execute();  result = cstmt.getString(4);  System.out.println(" "+s\_id+" "+c\_id+" "+c\_class); //콘솔 창 출력  %>  <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>  5-1)delete.jsp  <%@ page language=*"java"* contentType=*"text/html; charset=EUC-KR"*  pageEncoding=*"UTF-8"*%><%@ page import=*"java.sql.\*"*%>    <html><head><title>수강신청 삭제</title></head>  <body>  <%@ include file=*"top.jsp"* %>  <% if (session\_id == null) response.sendRedirect("login.jsp"); %>  <table width=*"75%"* align=*"center"* border>  <br>  <tr><th>과목번호</th><th>과목명</th><th>분반</th>  <th>강의시간</th><th>강의실</th><th>담당교수</th><th>학점</th><th>수강취소</th></tr>  <%  String dbdriver = "oracle.jdbc.driver.OracleDriver";  String dburl = "jdbc:oracle:thin:@localhost:1521:orcl";  String user = "db1715333";  String passwd = "ss2";  Connection conn = null;  Statement stmt = null; Statement stmt2 = null;  CallableStatement cstmt = null; CallableStatement cstmt2 = null; CallableStatement cstmt3 = null;  ResultSet rs = null; ResultSet rs2 = null;  try {  Class.forName(dbdriver);  conn=DriverManager.getConnection(dburl, user, passwd);  stmt = conn.createStatement();    String func = "{? = call Date2EnrollYear(SYSDATE)}";  cstmt = conn.prepareCall(func);  cstmt.registerOutParameter(1,java.sql.Types.INTEGER);  cstmt.execute();  int year = cstmt.getInt(1);  String func2 = "{? = call Date2EnrollSemester(SYSDATE)}";  cstmt2 = conn.prepareCall(func2);  cstmt2.registerOutParameter(1,java.sql.Types.INTEGER);  cstmt2.execute();  int semester = cstmt2.getInt(1);    session.setAttribute("yearNow", year);  session.setAttribute("semesterNow", semester);    String SQL="select c\_id,c\_class from enroll where s\_id='" + session\_id + "' and e\_year=" + year + " and e\_semester=" + semester;      rs = stmt.executeQuery(SQL);    if(rs.next()) {  do {  String c\_id = rs.getString("c\_id");  int c\_class = rs.getInt("c\_class");    stmt2 = conn.createStatement();  String SQL2 = "select c.c\_name, t.t\_day,t.t\_time1, t.t\_time2,t.t\_location, c.c\_credit,t.p\_name from course c,teach t where t.t\_year=" + year + " and t.t\_semester=" + semester + " and c.c\_id = '" + c\_id + "' and c.c\_id = t.c\_id and t.c\_class ="+ c\_class;  rs2 = stmt2.executeQuery(SQL2);    rs2.next();  String c\_name = rs2.getString("c\_name");  String t\_day = rs2.getString("t\_day");  String t\_time1 = rs2.getString("t\_time1");  String t\_time2 = rs2.getString("t\_time2");  String t\_location = rs2.getString("t\_location");  int c\_credit = rs2.getInt("c\_credit");  String p\_name = rs2.getString("p\_name");  %>    <tr>  <td align=*"center"*><%=c\_id%></td>  <td align=*"center"*><%=c\_name%></td>  <td align=*"center"*><%=c\_class%></td>  <td align=*"center"*><%=t\_day %> <%= t\_time1%> - <%=t\_time2%></td>  <td align=*"center"*><%=t\_location%></td>  <td align=*"center"*><%=p\_name%></td>  <td align=*"center"*><%=c\_credit%></td>  <td align=*"center"*><a href=*"delete\_verify.jsp?c\_id=*<%=c\_id%>*&c\_class=*<%=c\_class%>*"*>취소</a></td>  </tr>    <%  } while(rs.next());  } else {  %>  <tr>  <td colspan=*8*><div align=*"center"*>신청한 강의가 없습니다. </div></td>  </tr>  <%  }    cstmt3 = conn.prepareCall("{call SumTable(?, ?, ?, ?, ?)}");  cstmt3.setString(1, session\_id);  cstmt3.setInt(2, year);  cstmt3.setInt(3, semester);  cstmt3.registerOutParameter(4, java.sql.Types.INTEGER);  cstmt3.registerOutParameter(5, java.sql.Types.INTEGER);  cstmt3.execute();    int classSum = cstmt3.getInt(4);  int creditSum = cstmt3.getInt(5);  %>  </table>  <br>    <table width=*"30%"* align=*"center"* border>  <tr>  <td align=*"center"*>총 신청과목</td>  <td align=*"center"*><%=classSum%></td>  <td align=*"center"*>총 신청 학점</td>  <td align=*"center"*><%=creditSum%></td>  </tr>  </table>  <%  } catch (ClassNotFoundException e) {  System.out.println("jdbc driver 오류");  } catch (SQLException e) {  System.out.println("오라클 오류 -" + e.getErrorCode());  } finally {  if (rs2 != null) try { rs2.close(); } catch(SQLException ex) {}  if (rs != null) try { rs.close(); } catch(SQLException ex) {}  if (stmt2 != null) try { stmt2.close(); } catch(SQLException ex) {}  if (stmt != null) try { stmt.close(); } catch(SQLException ex) {}  if (cstmt3 != null) try { cstmt2.close(); } catch(SQLException ex) {}  if (cstmt2 != null) try { cstmt2.close(); } catch(SQLException ex) {}  if (cstmt != null) try { cstmt.close(); } catch(SQLException ex) {}  if (conn != null) try { conn.close(); } catch(SQLException ex) {}  }  %>  </table>  </body>  </html>  5-2)delete\_verify.jsp  <%@ page language=*"java"* contentType=*"text/html; charset=EUC-KR"*  pageEncoding=*"UTF-8"*%><%@ page import=*"java.sql.\*"*%>  <html><head><title>수강신청 삭제</title></head>  <body>  <%  String s\_id = (String) session.getAttribute("session\_id");  String c\_id = request.getParameter("c\_id");  int c\_class = Integer.parseInt(request.getParameter("c\_class"));  Integer year = (Integer) session.getAttribute("yearNow");  Integer semester = (Integer) session.getAttribute("semesterNow");  String dburl = "jdbc:oracle:thin:@localhost:1521:orcl";  String user = "db1715333";  String passwd = "ss2";  String dbdriver = "oracle.jdbc.driver.OracleDriver";  Class.forName(dbdriver);  Connection conn = DriverManager.getConnection(dburl, user, passwd);  Statement stmt = conn.createStatement();  String SQL = "DELETE FROM enroll WHERE s\_id='"+s\_id+"' and c\_id='"+c\_id+"' and c\_class=" + c\_class + " and e\_year="+year+"and e\_semester="+semester;  int resultSet = stmt.executeUpdate(SQL);  stmt.close();  conn.close();  %>  <script>  alert("수강취소가 완료 되었습니다.");  location.href="delete.jsp";  </script>  </body>  </html>  6-1)select.jsp  <%@ page language=*"java"* contentType=*"text/html; charset=EUC-KR"*  pageEncoding=*"EUC-KR"*%>  <%@ page import=*"java.sql.\*"* %>  <html>  <head>  <title>수강 조회</title>  </head>  <body>  <%@ include file=*"top.jsp"* %>  <br>  <form method=*"post"* action=*"show\_enroll.jsp"*>  <center>  <select name=*"year"*>  <option value=*"2020"* selected>2020년</option>  <option value=*"2021"*>2021년</option>  </select>  <select name=*"semester"*>  <option value=*"1"* selected>1학기</option>  <option value=*"2"*>2학기</option>  </select>  <input type=*"submit"* value=*"검색"*>  </center>  </form>  </body>  </html>  6-2)show\_enroll.jsp  <%@ page language=*"java"* contentType=*"text/html; charset=EUC-KR"*  pageEncoding=*"UTF-8"*%>  <%@page import=*"java.sql.\*"*%>  <%@ include file=*"select.jsp"*%>  <html>  <head>  <meta charset=*"EUC-KR"*>  <title>수강 조회</title></head>  <body>  <table width=*"75%"* align=*"center"* border>  <th>과목 번호</th><th>과목 이름</th>  <th>분반</th><th>요일</th><th>시간</th>  <th>장소</th><th>교수님</th><th>학점</th>  <tbody id=*"enroll\_table"*>  <%  String selec\_year= (String) request.getParameter("year");  String selec\_semester= (String) request.getParameter("semester");    if (selec\_year == null ) selec\_year="2020";    if (selec\_semester ==null ) selec\_semester="1";    int nowyear=0;  int nowsem=0;  %>  <%  if (session\_id == null) response.sendRedirect("login.jsp");  String dbdriver = "oracle.jdbc.driver.OracleDriver";  String dburl = "jdbc:oracle:thin:@localhost:1521:orcl";  String user = "db1715333";  String passwd = "ss2";  Connection Conn = null;  Statement stmt = null;  Statement stmt2 = null;  ResultSet rs = null;  ResultSet rs2 = null;  CallableStatement cstmt=null;  try {  Class.forName(dbdriver);  Conn = DriverManager.getConnection(dburl, user, passwd);  stmt = Conn.createStatement();  String SQL = "select c\_id, c\_class from enroll where s\_id='" + session\_id + "' and e\_year='" + selec\_year  + "' and e\_semester='" + selec\_semester + "'";  rs = stmt.executeQuery(SQL);  while (rs.next() != false) {  String c\_id = "", c\_name = "", t\_day = "", t\_time1 = "", t\_time2 = "";  String t\_location = "", p\_name = "";  int c\_credit, c\_class;  c\_id = rs.getString("c\_id");  c\_class = rs.getInt("c\_class");  stmt2 = Conn.createStatement();  String SQL2 = "select c.c\_id c\_id,c.c\_name c\_name, t.c\_class c\_class,t.t\_day t\_day,t.t\_time1 t\_time1, t.t\_time2 t\_time2,t.t\_location t\_location,c.c\_credit c\_credit,t.p\_name p\_name from course c,teach t where t.t\_year='"  + selec\_year + "' and t.t\_semester='" + selec\_semester + "' and c.c\_id = '" + c\_id + "' and c.c\_id = t.c\_id and t.c\_class ='"+ c\_class+"'";  rs2 = stmt2.executeQuery(SQL2);  while (rs2.next() != false) {  c\_id = rs2.getString("c\_id");  c\_name = rs2.getString("c\_name");  c\_class = rs2.getInt("c\_class");  t\_day = rs2.getString("t\_day");  t\_time1 = rs2.getString("t\_time1");  t\_time2 = rs2.getString("t\_time2");  t\_location = rs2.getString("t\_location");  c\_credit = rs2.getInt("c\_credit");  p\_name = rs2.getString("p\_name");  %>    <tr>  <td align=*"center"*><%=c\_id%></td>  <td align=*"center"*><%=c\_name%></td>  <td align=*"center"*><%=c\_class%></td>  <td align=*"center"*><%=t\_day%></td>  <td align=*"center"*><%=t\_time1%> - <%=t\_time2%></td>  <td align=*"center"*><%=t\_location%></td>  <td align=*"center"*><%=p\_name%></td>  <td align=*"center"*><%=c\_credit%></td>  </tr>  </tbody>    <%  }  }    nowyear= Integer.parseInt(selec\_year);  nowsem = Integer.parseInt(selec\_semester);    int nclass=0, nunit=0;  cstmt = Conn.prepareCall("{ call SumTable(?, ?, ?, ?, ?) }");  cstmt.setString(1, session\_id);  cstmt.setInt(2, nowyear);  cstmt.setInt(3, nowsem);  cstmt.registerOutParameter(4, java.sql.Types.INTEGER);  cstmt.registerOutParameter(5, java.sql.Types.INTEGER);  cstmt.execute();    nclass=cstmt.getInt(4);  nunit=cstmt.getInt(5);  %>  </table>  <br>    <table width=*"30%"* align=*"center"* border>  <tr>  <td align=*"center"*>총 신청과목</td>  <td align=*"center"*><%=nclass%></td>  <td align=*"center"*>총 신청 학점</td>  <td align=*"center"*><%=nunit%></td>  </tr>  </table>  <%  } catch (ClassNotFoundException e) {  System.out.println("jdbc driver 오류");  } catch (SQLException e) {  System.out.println("오라클 오류");  System.err.println("SQLException: " + e.getMessage());  } finally {  if (rs2 != null) try { rs2.close(); } catch(SQLException ex) {}  if (rs != null) try { rs.close(); } catch(SQLException ex) {}  if (stmt2 != null) try { stmt2.close(); } catch(SQLException ex) {}  if (stmt != null) try { stmt.close(); } catch(SQLException ex) {}  if (cstmt != null) try { cstmt.close(); } catch(SQLException ex) {}  if (Conn != null) try { Conn.close(); } catch(SQLException ex) {}  }  %>    </body>  </html>  7) BeforeUpdateStudent (트리거)  CREATE OR REPLACE TRIGGER BeforeUpdateStudent  BEFORE  UPDATE ON student  FOR EACH ROW  DECLARE  underflow\_length EXCEPTION;  invalid\_value EXCEPTION;  BEGIN  IF LENGTH( NVL(:new.s\_pwd, 0) ) < 4 THEN  RAISE underflow\_length;  END IF;  IF (INSTR(:new.s\_pwd, ' ') > 0) THEN  RAISE invalid\_value;  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');  END;  /  8-1)Date2EnrollYear (함수)  CREATE OR REPLACE FUNCTION Date2EnrollYear(eDate IN DATE)  RETURN NUMBER  IS  yearNow NUMBER;  monthNow NUMBER;  year NUMBER;  BEGIN  SELECT to\_number(to\_char(eDate, 'YYYY')) INTO yearNow FROM DUAL;  SELECT to\_number(to\_char(eDate, 'MM')) INTO monthNow FROM DUAL;  IF (monthNow = 11 OR monthNow = 12) THEN  year := yearNow + 1;  ELSE  year := yearNow;  END IF;  Return year;  END;  /  8-2)Date2EnrollSemester (함수)  CREATE OR REPLACE FUNCTION Date2EnrollSemester(eDate IN DATE)  RETURN NUMBER  IS  monthNow NUMBER;  semester NUMBER;  BEGIN  SELECT to\_number(to\_char(eDate, 'MM')) INTO monthNow FROM DUAL;  IF (monthNow >= 5 and monthNow <= 10) THEN  semester := 2;  ELSE  semester := 1;  END IF;  Return semester;  END;  /  8-3) InsertEnroll (프로시저)  CREATE OR REPLACE PROCEDURE InsertEnroll(sStudentId IN VARCHAR2,  sCourseId IN VARCHAR2,  nCourseIdNo IN NUMBER,  result OUT VARCHAR2)  IS  too\_many\_sumCourseUnit EXCEPTION;  too\_many\_courses EXCEPTION;  too\_many\_students EXCEPTION;  duplicate\_time EXCEPTION;  nYear NUMBER;  nSemester NUMBER;  nSumCourseUnit NUMBER;  v\_course course%ROWTYPE;  nSame NUMBER:=0;  nSameT NUMBER:=0;  nCnt NUMBER:=0;  nTeachMax teach.t\_max%TYPE;  nowday teach.t\_day%TYPE;  time1 DATE;  time2 DATE;  maxTime DATE;  minTime DATE;  i NUMBER;  leng1 number :=0;  day1 varchar2(12);  flag BOOLEAN NOT NULL:=false;  CURSOR time\_enroll(v\_sid enroll.s\_id%TYPE, v\_year enroll.e\_year%TYPE, v\_semester enroll.e\_semester%TYPE) IS  select t.t\_time1, t.t\_time2, t.t\_day  from enroll e, teach t  where e.s\_id=v\_sid AND e\_year=v\_year AND e\_year=t\_year AND e\_semester = v\_semester AND e\_semester = t\_semester AND  e.c\_id = t.c\_id AND e.c\_class = t.c\_class;  BEGIN  result := '';  DBMS\_OUTPUT.put\_line('#');  DBMS\_OUTPUT.put\_line(sStudentId || '님이 과목번호 ' || sCourseId ||  ', 분반 ' || TO\_CHAR(nCourseIdNo) || '의 수강 등록을 요청하였습니다.');  nYear := Date2EnrollYear(SYSDATE);  nSemester := Date2EnrollSemester(SYSDATE);  SELECT SUM(e.c\_credit)  INTO nSumCourseUnit  FROM enroll e  WHERE e.s\_id = sStudentId and e.e\_year = nYear and  e.e\_semester = nSemester;  SELECT c\_credit  INTO v\_course.c\_credit  FROM course  WHERE c\_id = sCourseId;  IF (nSumCourseUnit + v\_course.c\_credit > 18) THEN  RAISE too\_many\_sumCourseUnit;  END IF;  SELECT COUNT(\*)  INTO nSame  FROM enroll  WHERE s\_id = sStudentId and c\_id = sCourseId;  IF (nSame > 0) THEN  RAISE too\_many\_courses;  END IF;  SELECT t\_max  INTO nTeachMax  FROM teach  WHERE t\_year= nYear and t\_semester = nSemester  and c\_id = sCourseId and c\_class= nCourseIdNo;  SELECT COUNT(\*)  INTO nSameT  FROM enroll  WHERE e\_year = nYear and e\_semester = nSemester  and c\_id = sCourseId and c\_class = nCourseIdNo;  IF (nSameT >= nTeachMax) THEN  RAISE too\_many\_students;  END IF;  SELECT to\_date(t\_time1, 'HH24:MI'), to\_date(t\_time2, 'HH24:MI'), t\_day  INTO time1, time2, nowday  FROM teach  WHERE t\_year=nYear and t\_semester = nSemester and  c\_id = sCourseId and c\_class = nCourseIdNo;  nowday:=replace(nowday, ',', '');  leng1:=length(nowday);  FOR t\_e IN time\_enroll(sStudentId, nYear, nSemester) LOOP  i:=1;  flag:=false;  WHILE i<= leng1 LOOP  day1:=substr(nowday, i, 1);  IF instr(t\_e.t\_day, day1)>0 THEN  flag:=true;  END IF;  EXIT WHEN flag=true;  i:=i+1;  END LOOP;  IF flag= true THEN  maxTime := to\_date(t\_e.t\_time1, 'HH24:MI');  minTime := to\_date(t\_e.t\_time2, 'HH24:MI');  IF (time1>maxTime ) THEN  maxTime :=time1;  END IF;  IF (time2<minTime ) THEN  minTime :=time2;  END IF;  IF ((maxTime-minTime)\*60\*24 < 0) THEN  nCnt:=nCnt+1;  END IF;  END IF;  END LOOP;  IF (nCnt > 0) THEN  RAISE duplicate\_time;  END IF;  INSERT INTO enroll (S\_ID, C\_ID, C\_CLASS, E\_YEAR, E\_SEMESTER, C\_CREDIT)  VALUES (sStudentId, sCourseId, nCourseIdNo, nYear, nSemester, v\_course.c\_credit);  COMMIT;  result := '수강신청 등록이 완료되었습니다.';  EXCEPTION  WHEN too\_many\_sumCourseUnit THEN  result := '최대학점을 초과하였습니다.';  WHEN too\_many\_courses THEN  result := '이미 등록된 과목을 신청하였습니다.';  WHEN too\_many\_students THEN  result := '수강신청 인원이 초과되어 등록이 불가능합니다.';  WHEN duplicate\_time THEN  result := '이미 등록된 과목 중 중복되는 시간이 존재합니다.';  WHEN OTHERS THEN  ROLLBACK;  result := SQLCODE || ' ' || SQLERRM;  END;  /  9) SumTable (프로시저)  CREATE OR REPLACE PROCEDURE SumTable  (sStudentId IN VARCHAR2,  nYear IN NUMBER,  nSemester IN NUMBER,  nClass OUT NUMBER,  nUnit OUT NUMBER )  IS  v\_unit enroll.c\_credit%TYPE;  unitnum NUMBER;  CURSOR verify\_sum IS  SELECT e.c\_credit  FROM enroll e, teach t  WHERE e.s\_id=sStudentId AND e.e\_year=nYear AND e.e\_year=t.t\_year AND e.e\_semester=nSemester AND e.e\_semester=t.t\_semester AND  e.c\_id =t.c\_id AND e.c\_class =t.c\_class;  BEGIN  unitnum:=0;  OPEN verify\_sum;  LOOP  FETCH verify\_sum INTO v\_unit;  EXIT WHEN verify\_sum%NOTFOUND;  unitnum:=v\_unit+unitnum;  END LOOP;  nClass := verify\_sum%ROWCOUNT;  nUnit :=unitnum;  CLOSE verify\_sum;  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('에러 발생: ' || SQLERRM);  END;  /  10) enroll\_student (뷰)  CREATE OR REPLACE VIEW enroll\_student  AS  SELECT e.c\_id, e.c\_class, e.e\_year, e.e\_semester, COUNT(e.s\_id) AS st\_num  FROM enroll e, teach t  WHERE e.c\_id = t.c\_id AND t.t\_year = e.e\_year AND t.t\_semester = e.e\_semester AND e.c\_class = t.c\_class  GROUP BY e.c\_id, e.c\_class, e.e\_year, e.e\_semester; |