**DB Object 스크립트(PL-SQL)**

| 요구사항번호 | A-01 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 주변 맛집 추천 | 작성자 | 김지우 |
| DML | | | |
| -- 맛집 정보  create or replace view vwRM  as  select      r.name as "가게명",      r.address as "주소",      r.time as "거리",      r.likeCount as "추천수",      m.name as "메뉴",      m.price as "가격"  from tblRestaurant r      inner join tblMenu m          on r.restaurantSeq = m.menuSeq;  select \* from vwRM;      -- 새로운 맛집 및 메뉴 추가  select \* from tblRestaurant;  select \* from tblMenu;  create or replace procedure r\_insert\_p (      p\_seq in tblRestaurant.restaurantSeq%type,          -- 맛집seq      p\_name in tblRestaurant.name%type,                 -- 가게명      p\_address in tblRestaurant.address%type,            -- 주소      p\_time in tblRestaurant.time%type,                    -- 거리      p\_likeCount in tblRestaurant.likeCount%type        -- 추천수  )  is  begin          insert into tblRestaurant(restaurantSeq, name, address, time, likeCount) values (p\_seq, p\_name, p\_address, p\_time, p\_likeCount);            dbms\_output.put\_line('맛집seq' || '가게명' || '주소' || '거리' || '추천수');  end r\_insert\_p;  create or replace procedure m\_insert\_p (      p\_mseq in tblMenu.menuSeq%type,                    -- 메뉴 seq      p\_rseq in tblMenu.restaurantSeq%type,                -- 맛집 seq      p\_name in tblMenu.name%type,                        -- 메뉴명      p\_price in tblMenu.price%type                           -- 가격  )  is  begin      insert into tblMenu(menuSeq, restaurantSeq, name, price) values (p\_mseq, p\_rseq, p\_name, p\_price);        dbms\_output.put\_line('메뉴seq' || '맛집seq' || '메뉴명' || '가격');    end m\_insert\_p;  -- 특정 맛집 추천 > 추천수 증가  create or replace procedure r\_update\_p (      r\_seq\_p in tblRestaurant.restaurantSeq%type,          -- 맛집seq      r\_name\_p in tblRestaurant.name%type,                 -- 가게명      r\_address\_p in tblRestaurant.address%type,            -- 주소      r\_time\_p in tblRestaurant.time%type,                    -- 거리      r\_likeCount\_p in tblRestaurant.likeCount%type        -- 추천수  )  is  begin      update tblRestaurant      set likeCount = r\_likeCount\_p            where restaurantSeq = r\_seq\_p;          dbms\_output.put\_line('추천 성공');        exception when others then        dbms\_output.put\_line('추천 실패');    end r\_update\_p;  -- 해당 맛집 가격 및 메뉴 정보 수정 기능  create or replace procedure m\_update\_p (      m\_seq\_p in tblMenu.menuSeq%type,                      -- 메뉴 seq      m\_rseq\_p in tblMenu.restaurantSeq%type,                -- 맛집 seq      m\_name\_p in tblMenu.name%type,                        -- 메뉴명      m\_price\_p in tblMenu.price%type                           -- 가격  )  is  begin      update tblMenu      set name = m\_name\_p,          price = m\_price\_p        where menuSeq = m\_seq\_p;      dbms\_output.put\_line('메뉴seq' || '맛집seq' || '메뉴명' || '가격');        exception when others then        dbms\_output.put\_line('메뉴정보 수정 실패');    end m\_update\_p; | | | |

| 요구사항번호 | B-01 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 기초정보 관리 | 작성자 | 김지우 |
| DML | | | |
| -- 과정 관리  select \* from tblCourse;  -- 과정 추가  create or replace procedure c\_insert\_p (      p\_seq in tblCourse.courseSeq%type,              -- 과정번호      p\_name in tblCourse.courseTitle%type           -- 과정명  )  is  begin        insert into tblCourse(courseSeq, courseTitle) values (p\_seq, p\_name);        dbms\_output.put\_line('추가 완료');    end c\_insert\_p;  -- 과정 수정  create or replace procedure c\_update\_p (      c\_seq\_p in tblCourse.courseSeq%type,      c\_name\_p in tblCourse.courseTitle%type  )  is  begin      update tblCourse      set courseSeq = c\_seq\_p,          courseTitle = c\_name\_p        where courseSeq = c\_seq\_p;        exception when others then        dbms\_output.put\_line('수정 실패');        rollback;    end c\_update\_p;  -- 과정 삭제  create or replace procedure c\_delete\_p(      p\_seq in tblCourse.courseSeq%type,      p\_name in tblCourse.courseTitle%type  )  is  begin        delete from tblCourse      where p\_seq = courseSeq;        dbms\_output.put\_line('삭제 완료');    end c\_delete\_p;  -- 과정 테이블(tblCourse)에서 데이터 추가, 수정, 삭제 시 동작하는 트리거  create or replace trigger trgCourse  after  insert or update or delete  on tblCourse  begin  dbms\_output.put\_line('트리거 실행');  if inserting then  dbms\_output.put\_line('새로운 과정이 추가되었습니다.');  elsif updating then  dbms\_output.put\_line('과정명이 수정되었습니다');  elsif deleting then  dbms\_output.put\_line('과정이 삭제되었습니다.');  end if;  end trgCourse;  -- 과목 관리  select \* from tblSubject;  -- 과목 추가  create or replace procedure s\_insert\_p (      p\_seq in tblSubject.subjectSeq%type,      p\_name in tblSubject.subjectTitle%type  )  is  begin      insert into tblSubject(subjectSeq, subjectTitle) values (p\_seq, p\_name);        dbms\_output.put\_line('추가 완료');  end s\_insert\_p;  -- 과목 수정  create or replace procedure s\_update\_p (      s\_seq\_p in tblSubject.subjectSeq%type,      s\_name\_p in tblSubject.subjectTitle%type  )  is  begin      update tblSubject      set subjectSeq = s\_seq\_p,          subjectTitle = s\_name\_p        where subjectSeq = s\_seq\_p;        exception when others then        dbms\_output.put\_line('실패');    end s\_update\_p;  -- 과목 삭제  create or replace procedure s\_delete\_p(      p\_seq in tblSubject.subjectSeq%type,      p\_name in tblSubject.subjectTitle%type  )  is  begin        delete from tblSubject      where p\_seq = subjectSeq;        dbms\_output.put\_line('삭제 완료');    end s\_delete\_p;  -- 과목 테이블(tblSubject)에서 데이터 추가, 수정, 삭제 시 동작하는 트리거  create or replace trigger trgSubject  after  insert or update or delete  on tblSubject  begin  dbms\_output.put\_line('트리거 실행');  if inserting then  dbms\_output.put\_line('새로운 과목이 추가되었습니다.');  elsif updating then  dbms\_output.put\_line('과목명이 수정되었습니다');  elsif deleting then  dbms\_output.put\_line('과목이 삭제되었습니다.');  end if;  end trgSubject;  -- 강의실 관리  select \* from tblClassroom;  -- 강의실 추가  create or replace procedure cl\_insert\_p (      p\_seq in tblClassroom.classroomSeq%type,      p\_name in tblClassroom.classroomTitle%type,      p\_sc in tblClassroom.capacity%type  )  is  begin      insert into tblClassroom(classroomSeq, classroomTitle, capacity) values (p\_seq, p\_name, p\_sc);        dbms\_output.put\_line('추가 완료');  end cl\_insert\_p;  -- 강의실 수정  create or replace procedure cl\_update\_p (      cl\_seq\_p in tblClassroom.classroomSeq%type,      cl\_name\_p in tblClassroom.classroomTitle%type,      cl\_sc\_p in tblClassroom.capacity%type  )  is  begin      update tblClassroom      set classroomSeq = cl\_seq\_p,          classroomTitle = cl\_name\_p,          capacity = cl\_sc\_p        where classroomSeq = cl\_seq\_p;        exception when others then        dbms\_output.put\_line('실패');    end cl\_update\_p;  -- 강의실 삭제  create or replace procedure cl\_delete\_p(      p\_seq in tblClassroom.classroomSeq%type,      p\_name in tblClassroom.classroomTitle%type,      p\_sc in tblClassroom.capacity%type  )  is  begin        delete from tblClassroom      where p\_seq = classroomSeq;        dbms\_output.put\_line('삭제 완료');    end cl\_delete\_p;  -- 강의실 테이블(tblClassroom)에서 데이터 추가, 수정, 삭제 시 동작하는 트리거  create or replace trigger trgClassroom  after  insert or update or delete  on tblClassroom  begin  dbms\_output.put\_line('트리거 실행');  if inserting then  dbms\_output.put\_line('새로운 강의실이 추가되었습니다.');  elsif updating then  dbms\_output.put\_line('강의실명이 수정되었습니다');  elsif deleting then  dbms\_output.put\_line('강의실이 삭제되었습니다.');  end if;  end trgClassroom;  -- 교재 관리  select \* from tblBook;  -- 교재 추가  create or replace procedure b\_insert\_p (      p\_seq in tblBook.bookSeq%type,      p\_name in tblBook.bookTitle%type,      p\_ps in tblBook.publisher%type  )  is  begin      insert into tblBook(bookSeq, bookTitle, publisher) values (p\_seq, p\_name, p\_ps);        dbms\_output.put\_line('추가 완료');  end b\_insert\_p;  -- 교재 수정  create or replace procedure b\_update\_p (      b\_seq\_p in tblBook.bookSeq%type,      b\_name\_p in tblBook.bookTitle%type,      b\_ps\_p in tblBook.publisher%type  )  is  begin      update tblBook      set bookSeq = b\_seq\_p,          bookTitle = b\_name\_p,          publisher = b\_ps\_p      where bookSeq = b\_seq\_p;        exception when others then        dbms\_output.put\_line('실패');    end b\_update\_p;  -- 교재 삭제  create or replace procedure b\_delete\_p(      p\_seq in tblSubject.subjectSeq%type,      p\_name in tblSubject.subjectTitle%type  )  is  begin        delete from tblBook      where p\_seq = BookSeq;        dbms\_output.put\_line('삭제 완료');    end b\_delete\_p;  -- 교재 테이블(tblBook)에서 데이터 추가, 수정, 삭제 시 동작하는 트리거  create or replace trigger trgBook  after  insert or update or delete  on tblBook  begin  dbms\_output.put\_line('트리거 실행');  if inserting then  dbms\_output.put\_line('새로운 교재가 추가되었습니다.');  elsif updating then  dbms\_output.put\_line('교재명이 수정되었습니다');  elsif deleting then  dbms\_output.put\_line('교재가 삭제되었습니다.');  end if;  end trgBook; | | | |

| 요구사항번호 | B-02 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 교사 계정 관리 | 작성자 | 백정석 |
| DML | | | |
| 1. 전체 교사정보 출력   CREATE OR REPLACE VIEW vwTeacher  AS  SELECT      t.name as 교사명,      t.ssn as "주민번호 뒷자리",      t.tel as 전화번호,      s.subjecttitle as 강의강의과목  FROM tblteacher t      INNER JOIN tblteachersubject ts ON t.teacherseq = ts.teacherseq      INNER JOIN tblsubject s ON ts.subjectseq = s.subjectseq;     1. 특정 교사정보 출력   CREATE OR REPLACE PROCEDURE procTeacherInfo  (      pTeacherName IN VARCHAR2  )  IS      sSubjectTitle       tblsubject.subjecttitle%type;      osStartDate         tblopensubject.startdate%type;      osEndDate           tblopensubject.enddate%type;      cCoursetitle        tblcourse.coursetitle%type;      ocStartDate         tblopencourse.startdate%type;      ocEndDate           tblopencourse.enddate%type;      bbooktitle          tblbook.booktitle%type;      crclassroomtitle    tblclassroom.classroomtitle%type;      sLectureStatus      VARCHAR2(20);      CURSOR vcursor IS      SELECT          s.subjecttitle AS "배정된 개설 과목",          os.startdate AS "과목 시작 일",          os.enddate AS "과목 종료 일",          c.coursetitle AS "과정명",          oc.startdate AS "과정 시작 일",          oc.enddate AS "과정 종료 일",          b.booktitle AS "교재명",          cr.classroomtitle AS "강의실",          CASE              WHEN oc.enddate <= sysdate THEN '강의종료'              WHEN oc.startdate <= sysdate THEN '강의중'              WHEN oc.startdate >= sysdate THEN '예정'          END AS "강의진행여부"      FROM tblopensubject os          INNER JOIN tblteacher t ON os.teacherseq = t.teacherseq          INNER JOIN tblsubject s ON os.subjectseq = s.subjectseq          INNER JOIN tblopencourse oc ON os.opencourseseq = oc.opencourseseq          INNER JOIN tblcourse c ON oc.courseseq = c.courseseq          INNER JOIN tblbook b ON os.bookseq = b.bookseq          INNER JOIN tblclassroom cr ON oc.classroomseq = cr.classroomseq      WHERE t.name = pTeacherName;  BEGIN      OPEN vcursor;          LOOP              FETCH vcursor INTO sSubjectTitle, osStartDate, osEndDate, cCoursetitle,                                  ocStartDate, ocEndDate, bbooktitle, crclassroomtitle, sLectureStatus;              EXIT WHEN vcursor%notfound;              dbms\_output.put\_line('=====================================================================');              dbms\_output.put\_line('배정된 개설 과목: ' || sSubjectTitle);              dbms\_output.put\_line('과목 기간: ' || osStartDate || ' ~ ' || osEndDate);              dbms\_output.put\_line('과정명: ' || cCoursetitle);              dbms\_output.put\_line('과정 기간: ' || ocStartDate || ' ~ ' || ocEndDate);              dbms\_output.put\_line('교재명: ' || bbooktitle);              dbms\_output.put\_line('강의실: ' || crclassroomtitle);              dbms\_output.put\_line('강의진행여부: ' || sLectureStatus);              dbms\_output.put\_line('=====================================================================');              dbms\_output.put\_line('');          END LOOP;      CLOSE vcursor;  END procTeacherInfo; | | | |

| 요구사항번호 | B-03 | 작성일 | 2022-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 개설 과정 관리 | 작성자 | 이재성 |
| DML | | | |
| **- 개설 과정 전체 출력**  create or replace view vwOpenCourse  as  select  distinct  c.coursetitle as "과정명",  oc.startdate as "과정시작일",  oc.enddate as "과정종료일",  cr.classroomtitle as "강의실명",  case  when os.subjectseq is not null then '등록'  end as "개설과목등록여부",  oc.capacity as "교육생 등록 인원"  from tblOpenCourse oc  inner join tblCourse c  on oc.courseseq = c.courseseq  inner join tblStudent s  on oc.opencourseseq = s.opencourseseq  inner join tblClassroom cr  on cr.classroomseq = oc.classroomseq  inner join tblOpenSubject os  on os.opencourseseq = oc.opencourseseq  order by oc.startdate;  **- 특정 개설 과정 선택 과목 출력**  create or replace view vwSelectOC  as  select  oc.opencourseseq,  s.subjecttitle as "과목명",  oc.startdate as "과목시작일",  oc.enddate as "과목종료일",  b.booktitle as "교재명",  t.name as "교사명",  a.name as "교육생 이름",  a.ssn as "주민번호 뒷자리",  a.tel as "전화번호",  st.registerdate as "등록일",  case  when st.completeorfail = 1 then '수료'  when st.completeorfail = 0 then '중도탈락'  when st.completeorfail is null then '수강중'  end as "수료 및 중도탈락"  from tblOpencourse oc  inner join tblOpenSubject os  on os.opencourseseq = oc.opencourseseq  inner join tblSubject s  on os.subjectseq = s.subjectseq  inner join tblBook b  on os.bookseq = b.bookseq  inner join tblteacher t  on os.teacherseq = t.teacherseq  inner join tblStudent st  on oc.opencourseseq = st.opencourseseq  inner join tblApplicant a  on st.applicantseq = a.applicantseq;  **- 특정 과정 선택 시 과목 정보 출력**  create or replace procedure procCourseSubject(  pOpenCoureseSeq number  )  is  vseq number;  vsubjecttitle tblSubject.subjecttitle%type;  vstartdate tblOpenSubject.startdate%type;  venddate tblOpenSubject.enddate%type;  vbooktitle tblBook.booktitle%type;  vtname tblTeacher.name%type;  cursor vcursor is select distinct 과목명, 과목시작일, 과목종료일, 교재명, 교사명  from vwSelectOC where opencourseseq = pOpenCoureseSeq;    begin  vseq := 1;  dbms\_output.put\_line('=============================개설과목=============================');  open vcursor;    loop  fetch vcursor into vsubjecttitle, vstartdate, venddate, vbooktitle, vtname;  exit when vcursor%notfound;  dbms\_output.put\_line('-----------------------------------------------------------------');  dbms\_output.put\_line(vseq || '.' || vsubjecttitle);  dbms\_output.put\_line(' - 기간: ' || vstartdate || ' ~ ' || venddate);  dbms\_output.put\_line(' - 교재: ' || vbooktitle);  dbms\_output.put\_line(' - 교사: ' || vtname);  vseq := vseq + 1;  end loop;    dbms\_output.put\_line('-----------------------------------------------------------------');  dbms\_output.put\_line('=================================================================');  dbms\_output.put\_line('');  close vcursor;    end procCourseSubject;  **- 과정번호 입력**  begin  procCourseSubject(8)  end;  **- 특정 과목 선택 시 과목 정보 및 학생 정보 출력**  create or replace procedure procCourseSubjectStudent(  pOpenCoureseSeq number  )  is  vsname tblApplicant.name%type;  vssn tblApplicant.ssn%type;  vtel tblApplicant.tel%type;  vregdate tblStudent.registerdate%type;  vcompleteorfail varchar2(30);  cursor vcursor2 is select distinct "교육생 이름","주민번호 뒷자리", 전화번호, 등록일, "수료 및 중도탈락"  from vwSelectOC where opencourseseq = pOpenCoureseSeq;  begin  procCourseSubject(pOpenCoureseSeq);  dbms\_output.put\_line('==============================교육생==============================');  open vcursor2;  loop  fetch vcursor2 into vsname, vssn, vtel, vregdate, vcompleteorfail;  exit when vcursor2%notfound;  dbms\_output.put\_line('이름: ' || vsname || ' | 주민번호 뒷자리: ' || vssn || ' | 전화번호: ' || vtel  || ' | 등록일: ' || vregdate || ' | 수료 및 중도탈락: ' || vcompleteorfail);  end loop;    close vcursor2;  end procCourseSubjectStudent;  **- 과정 번호 입력**  begin  procCourseSubjectStudent(8);  end;  **- 수료 날짜 및 수료 지정**  create or replace procedure procComplete(  popencourseseq number,  pcompletedate date  )  is  begin  update tblStudent  set completeorfaildate = pcompletedate, completeorfail = 1  where opencourseseq = popencourseseq  and (completeorfail is null or completeorfail <> 0);  end procComplete;  **- 과정번호 및 수료날짜 입력**  begin  procComplete(13, '23-11-11');  end; | | | |

| 요구사항번호 | B-04 | 작성일 | 2022-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 개설 과목 관리 | 작성자 | 이재성 |
| DML | | | |
| **- 과목 정보 출력**  create or replace procedure procOpenSubject(  pOpenCoureseSeq number  )  is  vcoursetitle tblCourse.coursetitle%type;  vstartdate tblOpenCourse.startdate%type;  venddate tblOpenCourse.enddate%type;  vclassroomtitle tblClassroom.classroomtitle%type;  begin  select distinct coursetitle, startdate, enddate, classroomtitle  into vcoursetitle, vstartdate, venddate, vclassroomtitle  from tblOpenCourse oc  inner join tblCourse c  on oc.courseseq = c.courseseq  inner join tblClassroom cr  on oc.classroomseq = cr.classroomseq  where opencourseseq = pOpenCoureseSeq;    dbms\_output.put\_line('====================개설 과정====================');  dbms\_output.put\_line('- 과정명: ' || vcoursetitle);  dbms\_output.put\_line('- 과정기간: ' || vstartdate || ' ~ ' || venddate);  dbms\_output.put\_line('- 강의실명: ' || vclassroomtitle);  dbms\_output.put\_line('=================================================');  dbms\_output.put\_line('');  procCourseSubject(pOpenCoureseSeq);    end procOpenSubject;  **- 과정 번호 입력**  begin  procOpenSubject(8);  end;  /  **- 교사 강의 가능 과목 출력**  create or replace procedure procTeacherSubject(  psubjecttitle varchar2  )  is  vsubjecttitle tblSubject.subjecttitle%type;  vname tblTeacher.name%type;  cursor vcursor is select distinct 과목명, 교사명  from (select  rownum,  s.subjecttitle as "과목명",  t.name as "교사명"  from tblSubject s  inner join tblTeacherSubject ts  on s.subjectseq = ts.subjectseq  inner join tblTeacher t  on ts.teacherseq = t.teacherseq  where subjecttitle = psubjecttitle);  begin  open vcursor;  loop    fetch vcursor into vsubjecttitle, vname;  exit when vcursor%notfound;  dbms\_output.put\_line('교사명: ' || vname || ' | ' || '과목명: ' || vsubjecttitle);  end loop;  close vcursor;  end procTeacherSubject;  **- 과목명 입력**  begin  procTeacherSubject('자바 프로그래밍');  end; | | | |

| 요구사항번호 | B-05 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 관리 | 작성자 | 백정석 |
| DML | | | |
| 1. 전체 교육생 정보 출력   CREATE OR REPLACE VIEW vwStudent  AS  SELECT      a.name AS "이름",      a.ssn AS "주민번호",      a.tel AS "전화번호",      s.registerdate AS "등록일",      (          SELECT COUNT(\*)          FROM tblstudent s2          WHERE s2.applicantseq = s.applicantseq      ) AS "수강횟수"  FROM tblapplicant a      INNER JOIN tblstudent s ON a.applicantseq = s.applicantseq      INNER JOIN tblOpenCourse oc ON oc.openCourseSeq = s.openCourseSeq;     1. 특정 교육생 선택 출력   CREATE OR REPLACE PROCEDURE procStudentInfo  (      pStudentSeq IN NUMBER  )  IS      cCourseTitle        tblcourse.coursetitle%type;      ocStartDate         tblopencourse.startdate%type;      ocEnddate           tblopencourse.enddate%type;      crClassRoomTitle    tblclassroom.classroomtitle%type;      sCompleteOrFail     VARCHAR2(20);      CURSOR vcursor IS      SELECT          c.coursetitle AS 과정명,          oc.startdate AS "과정 시작 일",          oc.enddate AS "과정 종료 일",          cr.classroomtitle AS "강의실",          CASE              WHEN s.completeOrFail = 0 THEN '중도탈락'              WHEN s.completeOrFail = 1 THEN '수료'              ELSE '수강중'          END AS "수료여부"      FROM tblstudent s          INNER JOIN tblopencourse oc ON s.opencourseseq = oc.opencourseseq          INNER JOIN tblcourse c ON oc.courseseq = c.courseseq          INNER JOIN tblclassroom cr ON oc.classroomseq = cr.classroomseq      WHERE s.studentseq = pStudentSeq;  BEGIN      OPEN vcursor;          LOOP              FETCH vcursor INTO cCourseTitle, ocStartDate, ocEnddate, crClassRoomTitle,                                  sCompleteOrFail;              EXIT WHEN vcursor%notfound;              dbms\_output.put\_line('==========================================================');              dbms\_output.put\_line('과정명: '|| cCourseTitle);              dbms\_output.put\_line('과정 기간: '|| ocStartDate || ' ~ ' || ocEnddate);              dbms\_output.put\_line('강의실: '|| crClassRoomTitle);              dbms\_output.put\_line('수료여부: '|| sCompleteOrFail);              dbms\_output.put\_line('==========================================================');          END LOOP;      CLOSE vcursor;  END procStudentInfo; | | | |

| 요구사항번호 | B-06 | 작성일 | 2022-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 시험 관리 및 성적 조회 | 작성자 | 이재성 |
| DML | | | |
| **- 전체 과정 및 과목 출력**  create or replace view vwCourseScore  as  select  oc.opencourseseq,  s.subjecttitle as "과목명",  os.startdate as "과목시작일",  os.enddate as "과목종료일",  c.coursetitle as "개설과정명",  cr.classroomtitle as "강의실명",  t.name as "교사명",  b.booktitle as "교재명"  from tblOpenCourse oc  inner join tblOpenSubject os  on oc.opencourseseq = os.opencourseseq  left outer join tblexamregister e  on os.opensubjectseq = e.opensubjectseq  left outer join (select opensubjectseq from tblScore group by opensubjectseq) oi  on os.opensubjectseq = oi.opensubjectseq  inner join tblSubject s  on s.subjectseq = os.subjectseq  inner join tblCourse c  on c.courseseq = oc.courseseq  inner join tblTeacher t  on t.teacherseq = os.teacherseq  inner join tblBook b  on b.bookseq = os.bookseq  inner join tblClassroom cr  on cr.classroomseq = oc.classroomseq  order by opencourseseq;  **- 성적, 필기시험, 실기시험 등록여부 출력**  create or replace view vwRegister  as  select  oc.opencourseseq,  s.subjectseq as "과목번호",  s.subjecttitle as "과목명",  case  when oi.opensubjectseq is not null then 'Y'  else 'N'  end as "성적등록여부",  case  when writtendate is not null then 'Y'  else 'N'  end as "필기시험등록여부",  case  when practicaldate is not null then 'Y'  else 'N'  end as "실기시험등록여부"  from tblOpenSubject os  inner join tblOpenCourse oc  on os.opencourseseq = oc.opencourseseq  inner join tblSubject s  on os.subjectseq = s.subjectseq  inner join tblExamRegister e  on os.opensubjectseq = e.opensubjectseq  left outer join (select opensubjectseq from tblScore group by opensubjectseq) oi  on os.opensubjectseq = oi.opensubjectseq  order by oc.opencourseseq;  **- 특정 개설 과정 선택 시 성적등록여부 및 시험 문제 파일 등록 여부 출력**  create or replace procedure procCourseScore(  popencourseseq number  )  is  vname tblSubject.subjecttitle%type;  vregscore varchar(2);  vregwritten varchar(2);  vregpractical varchar(2);    cursor vcursor is select 과목명, 성적등록여부, 필기시험등록여부, 실기시험등록여부  from vwRegister  where opencourseseq = popencourseseq;  begin  procCourseSubject(popencourseseq);    open vcursor;  dbms\_output.put\_line('-시험 관리-');  loop  fetch vcursor into vname, vregscore, vregwritten, vregpractical;  exit when vcursor%notfound;  dbms\_output.put\_line('과목명: ' || vname);  dbms\_output.put\_line(' - 성적등록여부: ' || vregscore ||  ' | 필기시험등록여부: ' || vregwritten || ' | 실기시험등록여부: ' || vregpractical);    end loop;    close vcursor;    end procCourseScore;  **- 과정 번호 입력**  begin  procCourseScore(8);  end;  **- 과목별 성적 출력**  create or replace view vwSubjectScore  as  select  os.opensubjectseq as "과목번호",  c.coursetitle as "개설과정명",  oc.startdate || '~' || oc.enddate as "개설과정기간",  cr.classroomtitle as "강의실명",  sj.subjecttitle as "개설 과목명",  t.name as "교사명",  b.booktitle as "교재명",  a.name as "교육생명",  a.ssn as "주민번호 뒷자리",  s.writtenscore as "필기점수",  s.practicalscore as "실기점수"  from tblOpenSubject os  inner join tblScore s  on os.opensubjectseq = s.opensubjectseq  inner join tblOpenCourse oc  on oc.opencourseseq = os.opencourseseq  inner join tblCourse c  on oc.courseseq = c.courseseq  inner join tblClassroom cr  on oc.classroomseq = cr.classroomseq  inner join tblTeacher t  on os.teacherseq = t.teacherseq  inner join tblBook b  on os.bookseq = b.bookseq  inner join tblStudent st  on s.studentseq = st.studentseq  inner join tblApplicant a  on st.applicantseq = a.applicantseq  inner join tblSubject sj  on os.subjectseq = sj.subjectseq;  **- 특정 과목 선택 시 수강하는 교육생들의 성적 출력**  create or replace procedure procSubjectScore(  popensubjectseq number  )  is  vcoursetitle tblCourse.coursetitle%type;  vdate varchar2(100);  vclassroomtitle tblClassroom.classroomtitle%type;  vsubjecttitle tblSubject.subjecttitle%type;  vtname tblTeacher.name%type;  vbooktitle tblBook.booktitle%type;    vsname tblApplicant.name%type;  vssn tblApplicant.ssn%type;  vwscore tblScore.writtenscore%type;  vpscore tblScore.practicalscore%type;    cursor vcursor is select 교육생명, "주민번호 뒷자리", 필기점수, 실기점수  from vwSubjectScore where 과목번호 = popensubjectseq;    begin  select distinct 개설과정명, 개설과정기간, 강의실명, "개설 과목명", 교사명, 교재명  into vcoursetitle, vdate, vclassroomtitle, vsubjecttitle, vtname, vbooktitle  from vwSubjectScore where 과목번호 = popensubjectseq;    dbms\_output.put\_line('====================개설 과정====================');  dbms\_output.put\_line('- 과정명: ' || vcoursetitle);  dbms\_output.put\_line('- 과정기간: ' || vdate);  dbms\_output.put\_line('- 강의실명: ' || vclassroomtitle);  dbms\_output.put\_line('- 개설과목명: ' || vsubjecttitle);  dbms\_output.put\_line('- 교사명: ' || vtname);  dbms\_output.put\_line('- 교재명: ' || vbooktitle);  dbms\_output.put\_line('=================================================');  dbms\_output.put\_line('');    dbms\_output.put\_line('-성적-');  open vcursor;  loop  fetch vcursor into vsname, vssn, vwscore, vpscore;  exit when vcursor%notfound;  dbms\_output.put\_line('이름 : ' || vsname || ' | 주민번호 뒷자리: ' || vssn || ' | 필기점수 : ' || vwscore || ' | 실기점수: ' || vpscore);  end loop;  close vcursor;  exception  when others then  dbms\_output.put\_line('배점이 등록되지 않은 과목을 입력하셨습니다.');  end procSubjectScore;  **- 개설 과목 번호 입력**  begin  procSubjectScore(20);  end;  **- 특정 교육생 선택시 교육생이 들었더 과목들의 성적 출력**  create or replace procedure procStudentScore(  pstudentseq number  )  is  vname tblApplicant.name%type;  vssn tblApplicant.ssn%type;  vcoursetitle tblCourse.coursetitle%type;  vdate varchar2(100);  vclassroomtitle tblClassroom.classroomtitle%type;    vsubjecttitle tblSubject.subjecttitle%type;  vsdate varchar2(100);  vtname tblTeacher.name%type;  vascore tblScore.attendancescore%type;  vwscore tblScore.writtenscore%type;  vpscore tblScore.practicalscore%type;    cursor vcursor is select 개설과목명, 개설과목기간, 교사명, 출석점수, 필기점수, 실기점수  from vwStudentScore where 교육생번호 = pstudentseq;    begin  select distinct "교육생 이름", "주민번호 뒷자리", 개설과정명, 개설과정기간, 강의실명  into vname, vssn, vcoursetitle, vdate, vclassroomtitle  from vwStudentScore where 교육생번호 = pstudentseq and rownum = 1;    dbms\_output.put\_line('====================교육생 정보====================');  dbms\_output.put\_line('- 이름: ' || vname);  dbms\_output.put\_line('- 주민번호: ' || vssn);  dbms\_output.put\_line('- 개설과정명: ' || vcoursetitle);  dbms\_output.put\_line('- 개설 과정기간: ' || vdate);  dbms\_output.put\_line('- 강의실명: ' || vclassroomtitle);  dbms\_output.put\_line('=================================================');  dbms\_output.put\_line('');    open vcursor;  loop  fetch vcursor into vsubjecttitle, vsdate, vtname, vascore, vwscore, vpscore;  exit when vcursor%notfound;    dbms\_output.put\_line('과목명 : ' || vsubjecttitle || ' | 개설과목기간: ' || vsdate || ' | 교사명: ' || vtname || ' | 출석점수: ' || vascore || ' | 필기점수 : ' || vwscore || ' | 실기점수: ' || vpscore);    end loop;  close vcursor;  exception  when others then  dbms\_output.put\_line('잘못된 번호를 입력하셨습니다.');  end procStudentScore;  **- 학생 번호 입력**  begin  procStudentScore(75);  end; | | | |

| 요구사항번호 | B-07 (관리자) | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 출결 관리 및 출결 조회 | 작성자 | 남건욱 |
| DML | | | |
| **학생 출결 세부정보 뷰> 조회시 참조**  create or replace view vwStudentAttendanceInfo\_total  as  select       s.studentseq as "학생번호",      to\_char(a.checkout,'yyyy-mm-dd') as "날짜",      case when a.checkin is null then          '안옴'      else          to\_char(a.checkin,'yyyy-mm-dd hh24:mi')      end          as "입실시간",         case when a.checkin is null then          '안옴'      else          to\_char(a.checkout,'yyyy-mm-dd hh24:mi')      end          as "퇴실시간",      case when record is null and a.checkin is not null and to\_char(a.checkin,'hh24:mi') < '09:01' then '정상'           when  to\_char(a.checkin,'hh24:mi') > '09:00'  then '지각'           when a.checkin is null and checkout is null then record           when to\_char(a.checkout,'hh24:mi') < '14:00' then '조퇴'           else record end as "분류"  from tblattendance a inner join tblstudent s      on a.studentseq = s.studentseq  order by 날짜;  **- 관리자가 원하는 과정에 모든 교육생 출결 확인> 해당 과정의 모든 교육생 > 전체기간단위 출석조회**  create or replace procedure procPrintAttendance (      -- pteacherseq number,      popencourse number  )  is   cursor vcursor is      select studentseq from tblStudent where opencourseseq = popencourse  and sysdate >  registerdate;   vseq number;   vabsent number;   vnormal number;   vsick number;   vearly number;   vetc number;   vout number;   vlate number;   vtotal number;   vname varchar2(20);   vnum number:=1;  begin      open vcursor;      loop          fetch vcursor into vseq;          exit when vcursor%notfound;            select name into vname from tblapplicant a inner join tblstudent t on a.applicantseq = t.applicantseq and t.studentseq = vseq;          select          count(case                  when record is null and to\_char(checkin,'hh24:mi') < '09:01' and to\_char(checkout,'hh24:mi') > '17:49' then 1              end) as "정상",          count(case              when record = '결석' then 1          end) as "결석",          count(case                  when record = '병가' then 1                end) as "병가",          count(case                  when record = '조퇴' then 1                end) as "조퇴",          count(case  when record = '기타' then 1                end) as "기타",          count(case                  when to\_char(checkin,'hh24:mi') > '09:00' then 1                end) as "지각",          count(case                  when to\_char(checkout,'hh24:mi') < '15:00' then 1                end) as "외출",          count(\*) as 출석일수          into  vnormal, vabsent, vsick, vearly, vetc, vlate, vout, vtotal          from tblattendance where studentseq = vseq ;          dbms\_output.put\_line(vnum ||'.' || '학생시퀀스 : ' || vseq || ', 이름 : '|| vname||' / ' || '정상 : '|| vnormal ||' / '||' 결석 : '|| vabsent||' / '|| '지각 : ' || vlate ||  ' / ' || '병가 : '|| vsick || ' / '||'조퇴 : '||vearly||' / ' || '기타 : '||vetc ||' / '||'외출 : '||vout  );          vnum := vnum+1;      end loop;      close vcursor;  end procPrintAttendance;  begin      procPrintAttendance(7);  end;  **- 관리자가 원하는 과정의 모든 교육생 출결 확인 > 지정한  기간의 출석조회**  create or replace procedure **procPrintAttendancePeriod** (      -- pteacherseq number,      popencourse number,      pmonth1 varchar2,      pmonth2 varchar2  )  is   cursor vcursor is      select studentseq from tblStudent where opencourseseq = popencourse  and sysdate >  registerdate;   vseq number;   vabsent number;   vnormal number;   vsick number;   vearly number;   vetc number;   vout number;   vlate number;   vtotal number;   vname varchar2(20);   vnum number:=1;  begin      open vcursor;      loop          fetch vcursor into vseq;          exit when vcursor%notfound;            select name into vname from tblapplicant a inner join tblstudent t on a.applicantseq = t.applicantseq and t.studentseq = vseq;          select          count(case                  when record is null and to\_char(checkin,'hh24:mi') < '09:01' and to\_char(checkout,'hh24:mi') > '17:49' then 1              end) as "정상",          count(case              when record = '결석' then 1          end) as "결석",          count(case                  when record = '병가' then 1                end) as "병가",          count(case                  when record = '조퇴' then 1                end) as "조퇴",          count(case  when record = '기타' then 1                end) as "기타",          count(case                  when to\_char(checkin,'hh24:mi') > '09:00' then 1                end) as "지각",          count(case                  when to\_char(checkout,'hh24:mi') < '15:00' then 1                end) as "외출",          count(\*) as 출석일수          into  vnormal, vabsent, vsick, vearly, vetc, vlate, vout, vtotal          from tblattendance where studentseq = vseq  and checkout >= to\_char ( to\_date(pmonth1, 'yyyy-mm-dd'), 'yyyy-mm-dd') and checkout <= to\_char ( to\_date(pmonth2, 'yyyy-mm-dd'), 'yyyy-mm-dd');          dbms\_output.put\_line(vnum ||'.' || '학생시퀀스 : ' || vseq || ', 이름 : '|| vname||' / ' || '정상 : '|| vnormal ||' / '||' 결석 : '|| vabsent||' / '|| '지각 : ' || vlate ||  ' / ' || '병가 : '|| vsick || ' / '||'조퇴 : '||vearly||' / ' || '기타 : '||vetc ||' / '||'외출 : '||vout  );          vnum := vnum+1;      end loop;      close vcursor;  end **procPrintAttendancePeriod**;  begin      procPrintAttendancePeriod(2, '2022-08-01', '2022-09-01'); -- 기간단위로 검색  end;  **- 관리자가 원하는 과정, 해당 당일 출결조회 !!**  create or replace procedure **procC\_A\_D**(      popencourseseq number,      pmonth in varchar2 -->> 일자 선택  )  is      cursor vcursor is select studentseq from tblStudent where opencourseseq = popencourseseq;      studentseq number;      vrow vwStudentAttendanceInfo\_total%rowtype;  begin      open vcursor;          loop              fetch vcursor into studentseq;              exit when vcursor%notfound;              select \* into vrow from vwStudentAttendanceInfo\_total where 학생번호 = studentseq and 날짜 = to\_char ( to\_date(pmonth , 'yyyy-mm-dd'), 'yyyy-mm-dd') ;              dbms\_output.put\_line('학생번호 : '||vrow.학생번호 || ', 날짜 : '|| vrow.날짜 || ', 입실시간 : '|| vrow.입실시간 || ', 퇴실시간 : ' ||vrow.퇴실시간 || ', 분류 : ' || vrow.분류);          end loop;      close vcursor;  exception      when NO\_DATA\_FOUND then          dbms\_output.put\_line('기록이 없습니다.');  end **procC\_A\_D**;  begin  **procC\_A\_D**(4 ,'2022-08-25' );  end; | | | |

| 요구사항번호 | B-08 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 선발 과정 | 작성자 | 백정석 |
| DML | | | |
| 1. 선발과정 정보   CREATE OR REPLACE VIEW vwSelectionStudent  AS  SELECT      a.name as "이름",      a.ssn as "주민번호",      a.tel as "전화번호",      r.score as "면접점수",      r.recruitdate as "면접날짜",      r.interviewer as "면접관"  FROM tblRecruit r      INNER JOIN tblApplicant a ON a.applicantseq = r.applicantseq          ORDER BY r.recruitdate;     1. 지원자 인적사항 출력   CREATE OR REPLACE VIEW vwPassStatus  AS  SELECT      a.name as "이름",      a.ssn as "주민번호",      a.tel as "전화번호",      r.recruitdate as "면접날짜",      r.score as "면접점수",      CASE          WHEN pass = 'pass' THEN '합격'          ELSE '탈락'      END AS "합격여부"  FROM tblRecruit r      INNER JOIN tblApplicant a ON a.applicantseq = r.applicantseq          ORDER BY r.recruitdate; | | | |

| 요구사항번호 | B-09 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 세미나 초청 | 작성자 | 백정석 |
| DML | | | |
| 1. 세미나 초청 정보 출력   CREATE OR REPLACE VIEW vwSeminar  AS  SELECT      s.teacher AS 강사명,      s.title AS 강의명,      s.seminardate AS 강연날짜,      c.name AS 기업명,      c.companysize AS 기업규모  FROM tblseminar s      INNER JOIN tblcompany c ON s.companyseq = c.companyseq; | | | |

| 요구사항번호 | B-10. | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 취업결과 관리 (추가요구사항) | 작성자 | 오소엽 |
| DML | | | |
| **/\***  **1. 취업결과 작성**  **2. 취업결과 조회 (특정 교육생, 연도별, 기간별, 직무별, 기업규모별, 기업별, 만족도별)**  **\*/**  **-- 취업결과 전체 목록 조회**  create or replace view vwJobResult  as  select      stdn.studentSeq as studentSeq,      substr(a.name,1,1) || lpad('\*', length(a.name)-2, '\*') ||  substr(a.name, length(a.name), 1) as name,      jr.hireDate as hireDate,      jr.job as job,      jr.company as company,      jr.companySize as companySize,      jr.satisfaction as satisfaction,      c.courseTitle as courseTitle  from tblStudent stdn      inner join tbljobResult jr          on stdn.studentSeq = jr.studentSeq              inner join tblApplicant a                  on a.applicantSeq = stdn.applicantSeq                      inner join tblOpenCourse openC                          on stdn.openCourseSeq = openC.openCourseSeq                              inner join tblCourse c                                  on openC.courseSeq = c.courseSeq                                      order by studentSeq, hireDate, company;  – 취업결과 전체 조회  select  studentSeq as 교육생번호,  name as 이름,  hireDate as 취업날짜,  job as 직무,  company as 기업명,  companySize as 기업규모,  satisfaction as 만족도,  courseTitle as 수강과정  from vwJobResult;  -- 특정 교육생 취업결과 조회  select  studentSeq as 교육생번호,  name as 이름,  hireDate as 취업날짜,  job as 직무,  company as 기업명,  companySize as 기업규모,  satisfaction as 만족도,  courseTitle as 수강과정  from vwJobResult where studentSeq = 94;  -- 특정 기간 내 취업결과 조회  select  studentSeq as 교육생번호,  name as 이름,  hireDate as 취업날짜,  job as 직무,  company as 기업명,  companySize as 기업규모,  satisfaction as 만족도,  courseTitle as 수강과정  from vwJobResult where hireDate between '2023-01-01' and '2023-04-01' order by hireDate;  -- 특정 직무별 취업결과 조회  select  studentSeq as 교육생번호,  name as 이름,  hireDate as 취업날짜,  job as 직무,  company as 기업명,  companySize as 기업규모,  satisfaction as 만족도,  courseTitle as 수강과정  from vwJobResult where job like '%개발%' order by job;  -- 특정 기업의 취업결과 조회  select  studentSeq as 교육생번호,  name as 이름,  hireDate as 취업날짜,  job as 직무,  company as 기업명,  companySize as 기업규모,  satisfaction as 만족도,  courseTitle as 수강과정  from vwJobResult where company like '더존%';  -- 만족도별 취업결과 조회  select  studentSeq as 교육생번호,  name as 이름,  hireDate as 취업날짜,  job as 직무,  company as 기업명,  companySize as 기업규모,  satisfaction as 만족도,  courseTitle as 수강과정  from vwJobResult where satisfaction >=7;  -- procInsertJobresult  -- tblJobResult에 데이터 추가하는 프로시저  -- exec procInsertJobResult(**<교육생seq>**, **‘<입사일>’**, ‘**<직무>’**, ‘**<기업명>’**, ‘**<기업규모>’**, **<만족도>**);  create or replace procedure procInsertJobResult(      pstudentSeq tblStudent.studentSeq%type,      phireDate tblJobResult.hireDate%type,      pjob tblJobResult.job%type,      pcompany tblJobResult.company%type,      pcompanySize tblJobResult.companySize%type,      psatisfaction tblJobResult.satisfaction%type  )  is  begin      insert into tblJobResult values((select max(jobResultSeq) + 1 from tbljobResult), pstudentSeq, phireDate, pjob, pcompany, pcompanySize, psatisfaction);      commit;      dbms\_output.put\_line('취업결과를 추가하였습니다.');  end procInsertJobResult;    -- procDeleteJobresult  -- tblJobResult에서 데이터 삭제하는 프로시저  -- exec procDeleteJobResult(**<취업결과seq>**);  create or replace procedure procDeleteJobResult(      pjobResultSeq tblJobResult.jobResultSeq%type  )  is      vcount number;  begin      select count(\*) into vcount from tblJobResult where jobResultSeq = pjobResultSeq;      if vcount = 0 then          raise NO\_DATA\_FOUND;      elsif vcount = 1 then          delete from tblJobResult where jobResultSeq = pjobResultSeq;         commit;          dbms\_output.put\_line('취업결과를 삭제하였습니다.');      end if;  exception      when NO\_DATA\_FOUND then          dbms\_output.put\_line('일치하는 취업결과가 없습니다.');  end procDeleteJobResult;  -- 취업결과 테이블(tblJobResult)에서 데이터 삽입 또는 삭제시 동작하는 트리거  create or replace trigger trgInsDelJobResult  after  insert or delete  on tblJobResult  for each row  begin  if inserting then  dbms\_output.put\_line('추가항목: ' || :new.jobResultSeq || '번');  elsif deleting then  dbms\_output.put\_line('삭제항목: ' || :old.jobResultSeq || '번');  end if;  end; | | | |

| 요구사항번호 | C-01 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 강의 스케줄 조회 | 작성자 | 백정석 |
| DML | | | |
| 1. 특정 교사 과목별 스케줄 조회   CREATE OR REPLACE PROCEDURE procTeacherSchedule  (      pteacherseq IN NUMBER  )  IS      osSubjectSeq       tblopensubject.subjectseq%type;      cCourseTitle       tblcourse.coursetitle%type;      ocStartDate        tblopencourse.startdate%type;      ocEndDate          tblopencourse.enddate%type;      crclassRoomTitle   tblclassroom.classroomtitle%type;      osStartDate        tblopensubject.startdate%type;      osEndDate          tblopensubject.enddate%type;      bBooktitle         tblbook.booktitle%type;      Personnel          VARCHAR2(10);      CURSOR vcursor IS      SELECT          os.subjectseq AS "과목번호",          c.coursetitle AS "과정명",          oc.startdate AS "과정 시작 일",          oc.enddate AS "과정 종료 일",          cr.classroomtitle AS "강의실",          os.startdate AS "과목 시작 일",          os.enddate AS "과목 종료 일",          b.booktitle AS "교재명",          (              SELECT COUNT(\*)              FROM tblstudent t1              WHERE t1.opencourseseq = oc.opencourseseq          ) AS "등록 인원"      FROM tblteacher t          INNER JOIN tblopensubject os ON t.teacherseq = os.teacherseq          INNER JOIN tblopencourse oc ON os.opencourseseq = oc.opencourseseq          INNER JOIN tblcourse c ON oc.courseseq = c.courseseq          INNER JOIN tblclassroom cr ON oc.classroomseq = cr.classroomseq          INNER JOIN tblbook b ON b.bookseq = os.bookseq      WHERE t.teacherseq = 1          ORDER BY os.subjectseq;  BEGIN      OPEN vcursor;          LOOP              FETCH vcursor INTO osSubjectSeq, cCourseTitle, ocStartDate, ocEndDate, crclassRoomTitle,                                  osStartDate, osEndDate, bBooktitle, Personnel;              EXIT WHEN vcursor%notfound;              dbms\_output.put\_line('========================='|| '과목 ' || osSubjectSeq || '=============================');              dbms\_output.put\_line('과정명 : ' || cCourseTitle);              dbms\_output.put\_line('과정 기간 : ' || ocStartDate || ' ~ ' || ocenddate);              dbms\_output.put\_line('강의실 : ' || crclassRoomTitle);              dbms\_output.put\_line('과목 기간 : ' || osStartDate || ' ~ ' || osEndDate);              dbms\_output.put\_line('교재명 : ' || bBooktitle);              dbms\_output.put\_line('등록 인원 : ' || Personnel || '명');              dbms\_output.put\_line('============================================================');              dbms\_output.put\_line('');          END LOOP;      CLOSE vcursor;  END procTeacherSchedule;     1. 교육중인 과정 교육생 조회   CREATE OR REPLACE PROCEDURE procCourseStudent  (      pteacherseq IN NUMBER  )  IS      sStudentSeq         tblstudent.studentseq%type;      aName               tblapplicant.name%type;      aTel                tblapplicant.tel%type;      sRegisterdate       tblstudent.registerdate%type;      status              VARCHAR2(20);      CURSOR vcursor IS      SELECT DISTINCT          s.studentseq AS "교육생 번호",          a.name AS "교육생 이름",          a.tel AS "전화번호",          s.registerdate AS "등록일",          CASE              WHEN s.completeOrFail = 0 THEN '중도탈락'              WHEN s.completeOrFail = 1 THEN '수료'              ELSE '수강중'          END AS "수료여부"      FROM tblapplicant a          INNER JOIN tblstudent s ON a.applicantseq = s.applicantseq          INNER JOIN tblopencourse oc ON s.opencourseseq = oc.opencourseseq          INNER JOIN tblopensubject os ON oc.opencourseseq = os.opencourseseq          INNER JOIN tblteacher t ON os.teacherseq = t.teacherseq      WHERE t.teacherseq = pteacherseq          AND oc.startdate <= sysdate          AND oc.enddate >= sysdate;  BEGIN      OPEN vcursor;          LOOP              FETCH vcursor INTO sStudentSeq, aName, aTel, sRegisterdate, status;              EXIT WHEN vcursor%notfound;              dbms\_output.put\_line('========================================================================================================');              dbms\_output.put\_line('교육생 번호: ' || sStudentSeq || ' | ' || '교육생 이름: ' || aName || ' | ' || '전화번호: ' || aTel || ' | ' || '등록일: ' || sRegisterdate || ' | ' || '수료여부: ' || status);              dbms\_output.put\_line('========================================================================================================');          END LOOP;      CLOSE vcursor;  END procCourseStudent; | | | |

| 요구사항번호 | C-02 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 배점 입출력 | 작성자 | 백정석 |
| DML | | | |
| 1. 과목별 배점 출력   CREATE OR REPLACE VIEW vwSubjectPoint  AS  SELECT      s.subjectseq AS "과목번호",      c.coursetitle AS "과정명",      oc.startdate AS "과정 시작 일",      oc.enddate AS "과정 종료 일",      cr.classroomtitle AS "강의실",      s.subjecttitle AS "과목명",      os.startdate AS "과목 시작 일",      os.enddate AS "과목 종료 일",      b.booktitle AS "교재명",      p.attendance AS "출결배점",      p.written AS "필기배점",      p.practical AS "실기배점"  FROM tblpoints p      INNER JOIN tblopensubject os ON p.pointsseq = os.pointsseq      INNER JOIN tblsubject s ON os.subjectseq = s.subjectseq      INNER JOIN tblopencourse oc ON os.opencourseseq = oc.opencourseseq      INNER JOIN tblcourse c ON oc.courseseq = c.courseseq      INNER JOIN tblclassroom cr ON oc.classroomseq = cr.classroomseq      INNER JOIN tblbook b ON os.bookseq = b.bookseq;     1. 특정 과목 배점 선택 출력   CREATE OR REPLACE PROCEDURE procSubjectSelect  (      psubjectSeq IN NUMBER  )  IS      pAttendance         tblpoints.attendance%type;      pWritten            tblpoints.written%type;      pPractical          tblpoints.practical%type;      erWrittenDate       tblexamregister.writtendate%type;      erPracticalDate     tblexamregister.practicaldate%type;      CURSOR vcursor IS      SELECT          p.attendance AS "출결배점",          p.written AS "필기배점",          p.practical AS "실기배점",          er.writtendate AS "필기 시험날짜",          er.practicaldate AS "실기 시험날짜"      FROM tblopensubject os          INNER JOIN tblsubject s ON os.subjectseq = s.subjectseq          INNER JOIN tblpoints p ON os.pointsseq = p.pointsseq          INNER JOIN tblexamregister er ON os.opensubjectseq = er.opensubjectseq      WHERE s.subjectseq = psubjectSeq;  BEGIN      OPEN vcursor;          LOOP              FETCH vcursor INTO pAttendance, pWritten, pPractical, erWrittenDate, erPracticalDate;              EXIT WHEN vcursor%notfound;              dbms\_output.put\_line('=================================================================================================================');              dbms\_output.put\_line('출결점수: ' || pAttendance || ' | ' || '필기점수: ' || pWritten || ' | ' || '실기점수: ' || pPractical || ' | ' || '시험 기간: ' || erWrittenDate || ' ~ ' || erPracticalDate);              dbms\_output.put\_line('=================================================================================================================');          END LOOP;      CLOSE vcursor;  END procSubjectSelect; | | | |

| 요구사항번호 | C-03 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 성적 입출력 | 작성자 | 남건욱 |
| DML | | | |
| **시험 점수 레코드 추가 > -- 개강시에 한 과정 안의 전체 학생들 성적 레코드를 추가**  create or replace procedure **procInsertAddAllScoreRecord**(      popensubjectseq number  )  is    -- 해당 강의를 듣는 학생들을 찾기      cursor studentcursor is          select studentseq from tblstudent where opencourseseq =  ( select opencourseseq from tblopensubject where opensubjectseq = popensubjectseq );      vstartdate date; -- open      studentseq number;      cnt number;  begin      -- 해당 과목과,      select startdate into vstartdate from tblopensubject where opensubjectseq = popensubjectseq;      select count(\*) into cnt from tblscore where opensubjectseq = popensubjectseq;      open studentcursor;      loop          fetch studentcursor into studentseq;          exit when studentcursor%notfound;          if to\_char (vstartdate,'yyyy-mm-dd') = to\_char ( sysdate, 'yyyy-mm-dd') and cnt = 0 then              --insert into tblscore values( (select nvl( max(scoreseq), 0) + 1 from tblscore) , 1 , 1, null, null, null);              dbms\_output.put\_line('추가 '); -- 이때만 추가          else              raise\_application\_error(-20000, '에러');          end if;      end loop;      close studentcursor;  exception      when others then          dbms\_output.put\_line('과목이 개설되지 않아 추가가 불가능합니다.과목이 개설된 당일날 레코드가 추가됩니다. ');  end procInsertAddAllScoreRecord;  begin      procInsertAddAllScoreRecord(10);  end;  **출석 점수 입력 or 필기 점수 입력  or 실기 점수 입력  > 출석배점, 필기 배점 실기배점의 범위를 넘어서면 에러출력**  - **모두 입력 혹은 개별입력 가능**  create or replace procedure **procInputAttendScore**(      pstudentseq number,      popensubjectseq number,      pas number, -- 출석점수      pws number, -- 필기점수      pps number  -- 실기점수  )  is      vrow tblPoints%rowtype;      visWsNull number;      visPsNull number;  begin        -- 시험등록 확인      select count(writtendate) into visWsNull from tblExamRegister where opensubjectseq = popensubjectseq;      select count(practicaldate) into visPsNull from tblExamRegister where opensubjectseq = popensubjectseq;        -- 배점 받아옴      select \* into vrow from tblPoints where pointsseq = (select pointsseq from tblopensubject where opensubjectseq = popensubjectseq);        -- DBMS\_OUTPUT.PUT\_LINE(vrow.attendance || vrow.written || vrow.practical) ;      if pas is not null then          -- 예외처리 > 배점을 넘으면 에러          if 0 > pas or pas > vrow.attendance then              raise\_application\_error(-20000, '출석배점 범위 에러');          update tblScore set attendancescore = pas  where opensubjectseq = popensubjectseq and studentseq = pstudentseq;          end if;      end if;      -- 필기시험이 등록되었으면 실행, 등록 안되었으면 실행 x      if visWsNull = 0 then          raise\_application\_error(-20000, ' 필기시험 등록 에러');      elsif visWsNull <> 0  and pws is not null then          -- 예외처리 > 배점을 넘으면 에러 > 트리거로          if 0 > pws or pws > vrow.written then              raise\_application\_error(-20000, '필기배점 범위 에러');          end if;          update tblScore set  writtenscore = pws  where opensubjectseq = popensubjectseq and studentseq = pstudentseq;      end if;      if visPsNull = 0 then          raise\_application\_error(-20000, '실기시험 등록 에러');      elsif visPsNull <> 0 and pps is not null then          if 0 > pps or pps > vrow.practical then              raise\_application\_error(-20000, '실기배점 범위 에러');          end if;          update tblScore set  practicalscore = pps  where opensubjectseq = popensubjectseq and studentseq = pstudentseq;      end if;  exception      when others then          DBMS\_OUTPUT.PUT\_LINE(SQLERRM) ; -- 에러 메세지 받음  end procInputAttendScore;    begin      procInputAttendScore(239, 34, 20, 40, 30);  end ;  **개별입력 가능 > 실기점수만 입력**  begin      procInputAttendScore(239, 34, 20, null, null);  -- 출석점수 개별입력  end ;  begin      procInputAttendScore(239, 34, null, 30, null);  -- 필기점수 개별입력  end ;  begin      procInputAttendScore(239, 34, null, null, 20);  -- 실기점수 개별입력  end ;  **- 성적 등록 여부를 출력하는 함수 프로시저 > 해당 과목을 수강하는 학생의 성적이 성적 테이블에 모두 null이 아닌 상태로 있으면 1을 반환함**  create or replace **function** **fnScoreAssign** (      psubjectSeq number  ) return number  is  cursor vcursor is          select \* from tblstudent where opencourseseq = (select opencourseseq from tblopensubject where opensubjectseq = psubjectSeq);  vrow tblstudent%rowtype;  flag number;  begin  open vcursor;      loop          fetch vcursor into vrow;          exit when vcursor%notfound;          select count(\*) into flag from tblscore where studentseq = vrow.studentseq and opensubjectseq = psubjectSeq and attendancescore is not null and writtenscore is not null and practicalscore is not null;          if flag = 0 then              return 0;          end if;      end loop;  close vcursor;  return 1;  end;  **-- 성적 등록 여부 확인**  declare  vnum number;  begin      vnum := fnScoreAssign(1);      dbms\_output.put\_line('성적 등록여부 결과 : ' || vnum);  end;  **- 과목 목록 출력시 과목번호, 과정명, 과정기간(시작 년월일, 끝 년월일) 강의실 과목명, 과목기간(시작년월일, 끝년월일), 교재명, 출결, 필기, 실기배점, 성적등록여부 출력**  **-- 교사가 강의를 마친 과목을 출력**  create or replace procedure procTeacherEndSub(      pTeacherseq number  )  is  cursor vcursor is      select          t.opensubjectseq as 개설과목번호,          ( select subjecttitle from tblsubject where subjectseq = t .subjectseq) as 과목명,          t.startdate as 시작기간,          t.enddate as 끝기간,          ( select classroomtitle from tblclassroom where classroomseq = ( select classroomseq from tblopencourse where opencourseseq = t.opencourseseq)) as 강의실,          ( select booktitle from tblbook where bookseq = t.bookseq) as 교재명,          (select attendance from tblPoints p where p.pointsseq = t.pointsseq) as 출결배점,          (select written from tblPoints p where p.pointsseq = t.pointsseq) as 필기배점,          (select practical from tblPoints p where p.pointsseq = t.pointsseq) as 실기배점,          case when fnScoreAssign(t.opensubjectseq) = 1 then '등록됨' else '등록안됨' end as 성적등록여부          from tblopensubject t where t.teacherseq = pTeacherseq and t.startdate < sysdate and t.enddate < sysdate;  vrow vcursor%rowtype;  begin      open vcursor;      loop          fetch vcursor into vrow;          exit when vcursor%notfound;          dbms\_output.put\_line('개설과목번호 : '||vrow.개설과목번호 || ', 과목명 : '|| vrow.과목명 ||', 시작년월일 : '|| vrow.시작기간 ||', 끝년월일 : '|| vrow.끝기간 );          dbms\_output.put\_line('강의실명 : '|| vrow.강의실 || ', 교재명 : '|| vrow.교재명);          dbms\_output.put\_line('출결배점 : ' || vrow.출결배점 ||', 필기배점 : ' || vrow.필기배점 || ', 실기배점 : ' || vrow.실기배점 || ', 성적등록여부 : ' || vrow.성적등록여부);          dbms\_output.put\_line(' ');      end loop;      close vcursor;  end procTeacherEndSub;  begin      procTeacherEndSub(4);  end;  **특정과목을 과목번호로 선택시 교육생 정보(이름, 전화번호, 수료 또는 중도탈락)  및 성적이 출결, 필기, 실기 점수로 구분되어서 출력**  create or replace function **fnPrintNo**( -- 입력값이 null인 경우 ‘미입력’ 출력을 위한 함수      pdate number  )return varchar2  is  begin      if pdate is null then          return '미입력';      else          return pdate;      end if;  end fnPrintNo;    **-- 해당 요구사항의 내용을 출력하기 위한 프로시저**  create or replace procedure procTeacherSelSub\_P\_student(      pOpenSubjectSeq number  )  is  cursor vcursor is      select      (select name from tblapplicant a inner join tblstudent s on a.applicantseq = s.applicantseq where s.studentseq=sos.studentseq) as "이름",      (select tel from tblapplicant a inner join tblstudent s on a.applicantseq = s.applicantseq where s.studentseq=sos.studentseq) as "전화번호",      sos.studentseq as "학생번호",      sos.opensubjectseq as "과목번호",      case          when sos.completeorfail = 0 then '중도포기자'          when sos.completeorfail = 1 then '수료자'          else '교육수강중'      end as "중도포기여부",      case          when sos.completeorfail = 0 then to\_char(sos.completeorfaildate,'yyyy-mm-dd')          else '해당사항없음'      end as "중도포기날짜",      case          when              sos.completeorfail is null or sos.completeorfail = 1  then                  sc.attendancescore      end as "출결점수",      case      when sos.completeorfail is null or sos.completeorfail = 1  then              sc.writtenscore end      as "필기점수",      case      when sos.completeorfail is null or sos.completeorfail = 1 then      sc.practicalscore end as "실기점수"      from (select \* from  tblstudent s              inner join  tblopensubject os                  on s.opencourseseq = os.opencourseseq and os.opensubjectseq= pOpenSubjectSeq) sos          inner join tblscore sc              on sos.studentseq=sc.studentseq and sc.opensubjectseq = pOpenSubjectSeq;  begin      for vrow in vcursor loop          dbms\_output.put\_line('학생번호 : ' || vrow.학생번호|| ', 이름 : ' || vrow.이름 ||', 전화번호 : '|| vrow.전화번호);          dbms\_output.put\_line('중도포기여부 : ' || vrow.중도포기여부 || ', 중도포기날짜 : ' || vrow.중도포기날짜);          dbms\_output.put\_line('출결점수 : ' || fnPrintNo(vrow.출결점수) || ', 필기점수 : '|| fnPrintNo(vrow.필기점수) ||', 실기점수 : ' || fnPrintNo(vrow.실기점수));          dbms\_output.put\_line(' ');      end loop;  end procTeacherSelSub\_P\_student;  **- 특정 과목 번호를 프로시저에 입력**  begin      procTeacherSelSub\_P\_student(1);  end; | | | |

| 요구사항번호 | C-04(교사) | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 출결 관리 및 출결 조회 | 작성자 | 남건욱 |
| DML | | | |
| **학생 출결 세부정보 뷰**  create or replace view **vwStudentAttendanceInfo\_total**  as  select       s.studentseq as "학생번호",      to\_char(a.checkout,'yyyy-mm-dd') as "날짜",      case when a.checkin is null then          '안옴'      else          to\_char(a.checkin,'yyyy-mm-dd hh24:mi')      end          as "입실시간",         case when a.checkin is null then          '안옴'      else          to\_char(a.checkout,'yyyy-mm-dd hh24:mi')      end          as "퇴실시간",      case when record is null and a.checkin is not null and to\_char(a.checkin,'hh24:mi') < '09:01' then '정상'           when  to\_char(a.checkin,'hh24:mi') > '09:00'  then '지각'           when a.checkin is null and checkout is null then record           when to\_char(a.checkout,'hh24:mi') < '14:00' then '조퇴'           else record end as "분류"  from tblattendance a inner join tblstudent s      on a.studentseq = s.studentseq  order by 날짜;  **- 교사가 강의하는 전체과정 조회**  create or replace procedure procReadTeacherCourse\_(      pteacher number  )  is  cursor vcursor is      select          oc.opencourseseq as "과정번호",          (select coursetitle from tblcourse where tblcourse.courseseq=oc.courseseq ) as "강의과정",          oc.startdate as "시간기간",          oc.enddate as "끝기간",          case              when oc.startdate < sysdate and oc.enddate > sysdate then '진행중'              when oc.enddate < sysdate then '강의종료'              else '강의예정'          end as 강의여부                                                                                   -- 선생의 seq를 입력          from (select distinct opencourseseq from tblopensubject where teacherseq = pteacher) toc inner join tblopencourse  oc on toc.opencourseseq = oc.opencourseseq ;      vrow vcursor%rowtype;  begin      open vcursor;      loop          fetch vcursor into vrow;          exit when vcursor%notfound;          dbms\_output.put\_line('과정번호 : '||vrow.과정번호 || ', 강의과정 : '|| vrow.강의과정 || ', 시간기간 : '|| vrow.시간기간 || ', 끝기간 : ' ||vrow.끝기간 || ', 강의여부 : ' || vrow.강의여부);      end loop;      close vcursor;  end procReadTeacherCourse\_;  begin   procReadTeacherCourse\_(4);  end;  **- 교사가 강의하는 과정에 모든 교육생 출결 확인> 해당 과정의 모든 교육생 > 전체기간단위 출석조회**  create or replace procedure procPrintAttendance (      -- pteacherseq number,      popencourse number  )  is   cursor vcursor is      select studentseq from tblStudent where opencourseseq = popencourse  and sysdate >  registerdate;   vseq number;   vabsent number;   vnormal number;   vsick number;   vearly number;   vetc number;   vout number;   vlate number;   vtotal number;   vname varchar2(20);   vnum number:=1;  begin      open vcursor;      loop          fetch vcursor into vseq;          exit when vcursor%notfound;            select name into vname from tblapplicant a inner join tblstudent t on a.applicantseq = t.applicantseq and t.studentseq = vseq;          select          count(case                  when record is null and to\_char(checkin,'hh24:mi') < '09:01' and to\_char(checkout,'hh24:mi') > '17:49' then 1              end) as "정상",          count(case              when record = '결석' then 1          end) as "결석",          count(case                  when record = '병가' then 1                end) as "병가",          count(case                  when record = '조퇴' then 1                end) as "조퇴",          count(case  when record = '기타' then 1                end) as "기타",          count(case                  when to\_char(checkin,'hh24:mi') > '09:00' then 1                end) as "지각",          count(case                  when to\_char(checkout,'hh24:mi') < '15:00' then 1                end) as "외출",          count(\*) as 출석일수          into  vnormal, vabsent, vsick, vearly, vetc, vlate, vout, vtotal          from tblattendance where studentseq = vseq ;          dbms\_output.put\_line(vnum ||'.' || '학생시퀀스 : ' || vseq || ', 이름 : '|| vname||' / ' || '정상 : '|| vnormal ||' / '||' 결석 : '|| vabsent||' / '|| '지각 : ' || vlate ||  ' / ' || '병가 : '|| vsick || ' / '||'조퇴 : '||vearly||' / ' || '기타 : '||vetc ||' / '||'외출 : '||vout  );          vnum := vnum+1;      end loop;      close vcursor;  end procPrintAttendance;  begin      procPrintAttendance(7);  end;  **- 교사가 강의하는 과정에 모든 교육생 출결 확인 > 지정기간 단위 출석조회**  create or replace procedure **procPrintAttendancePeriod** (      -- pteacherseq number,      popencourse number,      pmonth1 varchar2,      pmonth2 varchar2  )  is   cursor vcursor is      select studentseq from tblStudent where opencourseseq = popencourse  and sysdate >  registerdate;   vseq number;   vabsent number;   vnormal number;   vsick number;   vearly number;   vetc number;   vout number;   vlate number;   vtotal number;   vname varchar2(20);   vnum number:=1;  begin      open vcursor;      loop          fetch vcursor into vseq;          exit when vcursor%notfound;            select name into vname from tblapplicant a inner join tblstudent t on a.applicantseq = t.applicantseq and t.studentseq = vseq;          select          count(case                  when record is null and to\_char(checkin,'hh24:mi') < '09:01' and to\_char(checkout,'hh24:mi') > '17:49' then 1              end) as "정상",          count(case              when record = '결석' then 1          end) as "결석",          count(case                  when record = '병가' then 1                end) as "병가",          count(case                  when record = '조퇴' then 1                end) as "조퇴",          count(case  when record = '기타' then 1                end) as "기타",          count(case                  when to\_char(checkin,'hh24:mi') > '09:00' then 1                end) as "지각",          count(case                  when to\_char(checkout,'hh24:mi') < '15:00' then 1                end) as "외출",          count(\*) as 출석일수          into  vnormal, vabsent, vsick, vearly, vetc, vlate, vout, vtotal          from tblattendance where studentseq = vseq  and checkout >= to\_char ( to\_date(pmonth1, 'yyyy-mm-dd'), 'yyyy-mm-dd') and checkout <= to\_char ( to\_date(pmonth2, 'yyyy-mm-dd'), 'yyyy-mm-dd');          dbms\_output.put\_line(vnum ||'.' || '학생시퀀스 : ' || vseq || ', 이름 : '|| vname||' / ' || '정상 : '|| vnormal ||' / '||' 결석 : '|| vabsent||' / '|| '지각 : ' || vlate ||  ' / ' || '병가 : '|| vsick || ' / '||'조퇴 : '||vearly||' / ' || '기타 : '||vetc ||' / '||'외출 : '||vout  );          vnum := vnum+1;      end loop;      close vcursor;  end **procPrintAttendancePeriod**;    begin      procPrintAttendancePeriod(2, '2022-08-01', '2022-09-01'); -- 기간단위로 검색  end;  **- 교사가 원하는 과정, 해당 당일 출결조회 !!**  create or replace procedure **procC\_A\_D**(      popencourseseq number,      pmonth in varchar2 -->> 일자 선택  )  is      cursor vcursor is select studentseq from tblStudent where opencourseseq = popencourseseq;      studentseq number;      vrow vwStudentAttendanceInfo\_total%rowtype;  begin      open vcursor;          loop              fetch vcursor into studentseq;              exit when vcursor%notfound;              select \* into vrow from vwStudentAttendanceInfo\_total where 학생번호 = studentseq and 날짜 = to\_char ( to\_date(pmonth , 'yyyy-mm-dd'), 'yyyy-mm-dd') ;              dbms\_output.put\_line('학생번호 : '||vrow.학생번호 || ', 날짜 : '|| vrow.날짜 || ', 입실시간 : '|| vrow.입실시간 || ', 퇴실시간 : ' ||vrow.퇴실시간 || ', 분류 : ' || vrow.분류);          end loop;      close vcursor;  exception      when NO\_DATA\_FOUND then          dbms\_output.put\_line('기록이 없습니다.');  end **procC\_A\_D**;  begin  **procC\_A\_D**(4 ,'2022-08-25' );  end; | | | |

| 요구사항번호 | D-01 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 계정 | 작성자 | 남건욱 |
| DML | | | |
| 1. **교육생 로그인**   **- 교육생이 로그인 시 교육생 개인정보와 과정기간, 강의실 정보가 출력된다.**  **> 뷰로 생성하여 필요시 재활용**  create or replace view vwStudentInfo  as      select a.name "이름", a.ssn "주민번호", a.tel "전화번호", c.coursetitle "과정이름", o.startdate "시간기간", o.enddate "종료기간"      from tblStudent s          inner join tblApplicant a              on a.applicantseq = s.applicantseq                  inner join tblopencourse o                      on o.opencourseseq = s.opencourseseq                          inner join tblcourse c                              on c.courseseq = o.courseseq;     1. **성적 정보 출력**   **- 성적정보 출력 : 개설과목번호, 과목명, 시작년월일, 끝 년월일, 출결배점, 필기배점, 실기배점, 출석점수, 필기점수, 실기점수, 필기시험날짜, 실기시험날짜**  **> 프로시저로 작성**  create or replace procedure procCurrentScoreInfo (      pname in varchar2,      pssn in varchar2  )  is      cursor vcursor(pname tblapplicant.name%type, pssn tblapplicant.ssn%type)      is      select              'SUB' || sc.opensubjectseq as 개설과목번호,              (select subjecttitle from tblsubject where subjectseq = ( select subjectseq from tblopensubject os where os.opensubjectseq = sc.opensubjectseq )) as 과목명,              (select startdate from tblopensubject where opensubjectseq = sc.opensubjectseq ) as 시작년월일,              (select startdate from tblopensubject where opensubjectseq = sc.opensubjectseq ) as 끝년월일,              (select b.booktitle from tblopensubject os inner join tblbook b on os.bookseq = b.bookseq where os.opensubjectseq = sc.opensubjectseq) as 교재명,              (select t.name from tblteacher t inner join tblopensubject os on t.teacherseq = os.teacherseq  where os.opensubjectseq = sc.opensubjectseq) as 교사명,              -- 출결 배점              (select p.attendance from tblopensubject os inner join tblpoints p on os.pointsseq = p.pointsseq where os.opensubjectseq = sc.opensubjectseq) as 출결배점,              -- 필기 배점              (select p.written from tblopensubject os inner join tblpoints p on os.pointsseq = p.pointsseq where os.opensubjectseq = sc.opensubjectseq) as 필기배점,              -- 실기배점              (select p.practical from tblopensubject os inner join tblpoints p on os.pointsseq = p.pointsseq where os.opensubjectseq = sc.opensubjectseq) as 실기배점,              sc.attendancescore as 출결점수,              sc.writtenscore as 필기점수,              sc.practicalscore as 실기점수,              (select writtendate from tblExamRegister e where e.opensubjectseq = sc.opensubjectseq) as 필기시험날짜,              (select practicaldate from tblExamRegister e where e.opensubjectseq = sc.opensubjectseq) as 실기시험날짜      from (select \* from tblscore where studentseq = (select s.studentseq from tblapplicant a inner join tblstudent s on a.applicantseq = s.applicantseq and a.name=pname and a.ssn=pssn )) sc order by 시작년월일;      vrow vcursor%rowtype;  begin      open vcursor(pname,pssn);      loop          fetch vcursor into vrow;          exit when vcursor%notfound;          dbms\_output.put\_line('과목번호 : '||vrow.개설과목번호 || ', 과목명 : '|| vrow.과목명 ||', 시작년월일 : '|| vrow.시작년월일 ||', 끝년월일 : '|| vrow.끝년월일 ||', 교재명 : '|| vrow.교재명 || ', 교사명 : ' || vrow.교사명);          dbms\_output.put\_line('출결배점 : ' || vrow.출결배점 ||', 필기배점 : ' || vrow.필기배점 || ', 실기배점 : ' || vrow.실기배점 );          dbms\_output.put\_line('출결점수 : ' || vrow.출결점수 ||', 필기점수 : ' || vrow.필기점수 || ', 실기점수 : ' || vrow.실기점수 );          dbms\_output.put\_line('필기시험날짜 : ' || vrow.필기시험날짜 || ', 실기시험날짜 : ' || vrow.실기시험날짜 );          dbms\_output.put\_line(' ');      end loop;      close vcursor;  end procCurrentScoreInfo;    **- 실행코드**  begin      procCurrentScoreInfo('이순옥', '1066962');  end;     1. **출결 관리 및 출결 조회**   **- 매일 근태 관리를 기록할 수 있어야 한다. (**출근 1회, 퇴근 1회**)**  - 새로운 근태 기록을 추가 > 병가를 기록  create or replace procedure insertAttendanceRecord(  pname in tblapplicant.name%type,  pssn in tblapplicant.name%type,  precord in tblattendance.record%type,  pdate in varchar2   )  is  begin      if precord in (null, '결석', '병가', '기타', '조퇴', '외출') then          insert into tblattendance values( (select nvl( max(attendanceseq), 0) + 1 from tblattendance),          ( select studentseq from tblstudent where applicantseq = (select applicantseq from tblapplicant where name = pname and ssn = pssn)) ,           precord, null , to\_date( pdate ||'17:50', 'yyyy-mm-dd hh24 mi' ));      else          dbms\_output.put\_line('입력에러!!  결석, 병가, 기타, 조퇴, 외출만 기록해주세요.');  end if;    end insertAttendanceRecord;  **- 결석을 기록**  begin   insertAttendanceRecord('홍민석', '1541919', '결석', '2023-04-05');  end;  **- 새로운 근태 기록을 추가 > 정상출석인 경우**  begin   insertAttendanceRecord('홍민석', '1541919', null, '2023-04-05 08:59');  end;  **- 본인의 출결 현황 조회 > 전체 출결 현황 조회**  **- 뷰 생성**  create or replace view **vwStudentAttendanceInfo\_total**  as  select       s.studentseq as "학생번호",      to\_char(a.checkout,'yyyy-mm-dd') as "날짜",      case when a.checkin is null then          '안옴'      else          to\_char(a.checkin,'yyyy-mm-dd hh24:mi')      end          as "입실시간",         case when a.checkin is null then          '안옴'      else          to\_char(a.checkout,'yyyy-mm-dd hh24:mi')      end          as "퇴실시간",      case when record is null and a.checkin is not null and to\_char(a.checkin,'hh24:mi') < '09:01' then '정상'           when  to\_char(a.checkin,'hh24:mi') > '09:00'  then '지각'           when a.checkin is null and checkout is null then record           when to\_char(a.checkout,'hh24:mi') < '14:00' then '조퇴'           else record end as "분류"  from tblattendance a inner join tblstudent s      on a.studentseq = s.studentseq  order by 날짜;  **- 프로시저 생성**  create or replace procedure **procReadAttendanceRecord** (      pstudentseq in number  )  is      cursor vcursor is select \* from vwStudentAttendanceInfo\_total where 학생번호 = pstudentseq;      vrow vwStudentAttendanceInfo\_total%rowtype;  begin      open vcursor;          loop              fetch vcursor into vrow;              exit when vcursor%notfound;              dbms\_output.put\_line('학생번호 : '||vrow.학생번호 || ', 날짜 : '|| vrow.날짜 || ', 입실시간 : '|| vrow.입실시간 || ', 퇴실시간 : ' ||vrow.퇴실시간 || ', 분류 : ' || vrow.분류);          end loop;      close vcursor;  end procReadAttendanceRecord;    begin   procReadAttendanceRecord(89);  end;  - **본인의 출결 현황 조회 > 달 조회**  create or replace procedure **procReadAttendanceRecordMonth**(      pstudentseq in number,      pmonth in varchar2 -->> 월 선택  )  is      cursor vcursor is select \* from vwStudentAttendanceInfo\_total where 학생번호 = pstudentseq and 날짜 >= to\_char ( to\_date(pmonth, 'yyyy-mm-dd'), 'yyyy-mm-dd') and 날짜 < to\_char(ADD\_MONTHS(to\_date('2022-08-01', 'yyyy-mm-dd'),1),'yyyy-mm-dd');      vrow vwStudentAttendanceInfo\_total%rowtype;  begin      open vcursor;          loop              fetch vcursor into vrow;              exit when vcursor%notfound;              dbms\_output.put\_line('학생번호 : '||vrow.학생번호 || ', 날짜 : '|| vrow.날짜 || ', 입실시간 : '|| vrow.입실시간 || ', 퇴실시간 : ' ||vrow.퇴실시간 || ', 분류 : ' || vrow.분류);          end loop;      close vcursor;  end **procReadAttendanceRecordMonth**;  **-- 실행구문**  begin  **procReadAttendanceRecordMonth**(89,'2022-08-01' );  end;  **- 본인의 출결 현황 조회 > 일 조회**  create or replace procedure **procReadAttendanceRecordDay**(      pstudentseq in number,      pmonth in varchar2 -->> 일자 선택  )  is      cursor vcursor is select \* from vwStudentAttendanceInfo\_total where 학생번호 = pstudentseq and 날짜 = to\_char ( to\_date(pmonth , 'yyyy-mm-dd'), 'yyyy-mm-dd') ;      vrow vwStudentAttendanceInfo\_total%rowtype;  begin      open vcursor;          loop              fetch vcursor into vrow;              if vrow.학생번호 is null then                  dbms\_output.put\_line('기록이 없습니다.');              end if;              exit when vcursor%notfound;              dbms\_output.put\_line('학생번호 : '||vrow.학생번호 || ', 날짜 : '|| vrow.날짜 || ', 입실시간 : '|| vrow.입실시간 || ', 퇴실시간 : ' ||vrow.퇴실시간 || ', 분류 : ' || vrow.분류);            end loop;      close vcursor;  end procReadAttendanceRecordDay;  begin   procReadAttendanceRecordDay (89,'2022-08-25' );  end; | | | |

| 요구사항번호 | D-03. | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 스터디 신청 (추가요구사항) | 작성자 | 오소엽 |
| DML | | | |
| -- 스터디 목록 전체 조회  create or replace view vwStudy  as  select      study.studySeq as studySeq,      stdn.studentSeq as studentSeq,      a.name as name,      opc.courseSeq as courseSeq,      study.studyTitle as studyTitle,      study.registerDate as registerDate,      study.studyDate as studyDate,      study.capacity || '명' as capacity  from tblStudent stdn      inner join tblStudy study          on stdn.studentSeq = study.studentSeq              inner join tblApplicant a                  on stdn.applicantSeq = a.applicantSeq                      inner join tblOpenCourse opc                          on stdn.openCourseSeq = opc.openCourseSeq                              order by study.studySeq, study.registerDate, study.studyDate;  -- 특정 교육생별, 날짜별, 스터디 주제별 조회  -- 스터디 목록 전체 조회  select  studySeq as 스터디번호,  studentSeq as 교육생번호,  name as 이름,  courseSeq as 수강과정번호,  studyTitle as 스터디주제,  registerDate as 신청날짜,  studyDate as 스터디날짜,  capacity as 정원  from vwStudy;  -- 특정 교육생 스터디 조회  select  studySeq as 스터디번호,  studentSeq as 교육생번호,  name as 이름,  courseSeq as 수강과정번호,  studyTitle as 스터디주제,  registerDate as 신청날짜,  studyDate as 스터디날짜,  capacity as 정원  from vwStudy where studentSeq = **<교육생seq>**;  -- 기간별 스터디 조회  select  studySeq as 스터디번호,  studentSeq as 교육생번호,  name as 이름,  courseSeq as 수강과정번호,  studyTitle as 스터디주제,  registerDate as 신청날짜,  studyDate as 스터디날짜,  capacity as 정원  from vwStudy where registerDate between '**<신청날짜 or 스터디날짜>**' and '**<신청날짜 or 스터디날짜>**';  -- 스터디 주제 검색하여 조회  select  studySeq as 스터디번호,  studentSeq as 교육생번호,  name as 이름,  courseSeq as 수강과정번호,  studyTitle as 스터디주제,  registerDate as 신청날짜,  studyDate as 스터디날짜,  capacity as 정원  from vwStudy where studyTitle like '**%<검색할주제>%**';  -- procInsertStudy  -- tblStudy에 데이터 추가하는 프로시저  -- exec procStudyInsert(**<교육생seq>**, ‘**<스터디주제>**’, ‘**<신청날짜>**’, ‘**<스터디날짜>**’, ‘**<정원>**’);  create or replace procedure procInsertStudy(      pstudentSeq tblStudent.studentSeq%type,      pstudyTitle tblStudy.studyTitle%type,      pregisterDate tblStudy.registerDate%type,      pstudyDate tblStudy.studyDate%type,      pcapacity tblStudy.capacity%type  )  is  begin      insert into tblStudy values((select max(studySeq)+1 from tblStudy), pstudentSeq, pstudyTitle, pregisterDate, pstudyDate, pcapacity);      commit;      dbms\_output.put\_line('스터디를 추가하였습니다.');  end procInsertStudy;  -- procDeleteStudy  -- tblStudy에서 데이터 삭제하는 프로시저  -- exec procDeleteStudy(**<스터디seq>**);  create or replace procedure procDeleteStudy(      pstudySeq tblStudy.studySeq%type  )  is      vcount number;  begin      select count(\*) into vcount from tblStudy where studySeq = pstudySeq;      if vcount = 1 then          delete from tblStudy where studySeq = pstudySeq;          commit;          dbms\_output.put\_line('스터디를 삭제하였습니다.');      elsif vcount = 0 then          raise NO\_DATA\_FOUND;      end if;  exception      when NO\_DATA\_FOUND then          dbms\_output.put\_line('일치하는 스터디가 없습니다.');  end procDeleteStudy;  -- 스터디 테이블(tblStudy)에서 데이터 삽입 또는 삭제시 동작하는 트리거  create or replace trigger trgInsDelStudy  after  insert or delete  on tblStudy  for each row  begin  if inserting then  dbms\_output.put\_line('추가항목: ' || :new.studySeq || '번');  elsif deleting then  dbms\_output.put\_line('삭제항목: ' || :old.studySeq || '번');  end if;  end; | | | |

| 요구사항번호 | D-04. | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 취업 히스토리 (추가요구사항) | 작성자 | 오소엽 |
| DML | | | |
| -- 취업 히스토리 전체 조회하는 뷰뷰  create or replace view vwJobHistory  as  select  stdn.studentSeq as studentSeq,  substr(a.name,1,1) || lpad('\*', length(a.name)-2, '\*') ||  substr(a.name, length(a.name), 1) as name,ㅍ  jh.applyDate as applyDate,  case  when jr.hireDate is null then '미취업'  when jr.hireDate is not null then to\_char(jr.hireDate)  end as hireDate,  jh.companyName as company,  jh.job as job,  jh.companySize as companySize,  jh.review as review,  jh.pass as pass,  case  when jr.satisfaction is not null then jr.satisfaction || '점'  when jr.satisfaction is null then '-'  end as satisfaction,  spec.education as education,  spec.major as major,  case  when cf.certificateName is not null then cf.certificateName  when cf.certificateName is null then '없음'  end as certificate,  c.courseTitle as courseTitle,  case  when  stdn.completeOrFail is null and sysdate > opc.startDate  and sysdate < stdn.completeOrFailDate then '수강중'  when stdn.completeOrFail = 1  and sysdate > opc.endDate then stdn.completeOrFailDate ||' (수료)'  when stdn.completeOrFail = 0 then stdn.completeOrFailDate || ' (중도탈락)'  when stdn.completeOrFail is null and sysdate < opc.startDate then '개강전'  end as completeOrFail  from tblStudent stdn  inner join tblSpec spec  on stdn.studentSeq = spec.studentSeq  inner join tblApplicant a  on a.applicantSeq = stdn.applicantSeq  inner join tblOpenCourse opc  on stdn.openCourseSeq = opc.openCourseSeq  inner join tblCourse c  on opc.courseSeq = c.courseSeq  inner join tblJobHistory jh  on stdn.studentSeq = jh.studentSeq  left outer join tblJobResult jr  on stdn.studentSeq = jr.studentSeq  left outer join tblCerificate cf  on spec.specSeq = cf.specSeq  order by applyDate, hireDate, studentSeq, completeOrFailDate;  -- 취업 히스토리 전체 목록 조회  select  studentSeq as 교육생번호,  name as 이름,  applyDate as 지원날짜,  hireDate as 취업날짜,  company as 기업명,  job as 직무,  companySize as 기업규모,  review as 지원후기,  pass as 합격여부,  education as 학력,  major as 전공,  certificate as 자격증,  courseTitle as 수강과정,  completeOrFail as 수료날짜  from vwJobHistory;  -- procInsertJobHistory  -- tblJobHistory에 데이터 추가하는 프로시저  -- exec procInsertJobHistory(**<교육생seq>**, '**<직무>**', ‘**<기업명>**', '**<기업규모>**', '**<지원후기>**', '**<합격불합격>**', '**<지원날짜>**');  create or replace procedure procInsertJobHistory(  pstudentSeq tblStudent.studentSeq%type,  pjob tblJobHistory.job%type,  pcompanyName tblJobHistory.companyName%type,  pcompanySize tblJobHistory.companySize%type,  preview tblJobHistory.review%type,  ppass tblJobHistory.pass%type,  papplyDate tblJobHistory.applyDate%type  )  is  begin  insert into tblJobHistory values ((select max(jobHistorySeq)+1 from tblJobHistory), pstudentSeq, pjob, pcompanyName, pcompanySize, preview, ppass, papplyDate);  dbms\_output.put\_line('취업 히스토리를 추가하였습니다.');  end procInsertJobHistory;      -- procDeleteJobHistory  -- tblJobHistory에서 데이터 삭제하는 프로시저  -- exec procDeleteJobHistory(**<취업히스토리seq>**);  create or replace procedure procDeleteJobHistory(  pjobHistorySeq tblJobHistory.jobHistorySeq%type  )  is  vcount number;  begin  select count(\*) into vcount from tblJobHistory where jobHistorySeq = pjobHistorySeq;  if vcount = 0 then  raise NO\_DATA\_FOUND;  elsif vcount = 1 then  delete from tblJobHistory where jobHistorySeq = pjobHistorySeq;  dbms\_output.put\_line('취업 히스토리를 삭제하였습니다.');  end if;  exception  when NO\_DATA\_FOUND then  dbms\_output.put\_line('일치하는 취업 히스토리가 없습니다.');  end procDeleteJobHistory;    -- procCompanyReview  -- 기업별 리뷰 통계 조회하는 프로시저  -- exec procCompanyReview;  create or replace procedure procCompanyReview  is  pcompany tblJobHistory.companyName%type;  preview tblJobHistory.review%type;  pcount number;  cursor vcursor  is  select company, review, count(\*) as cnt  from vwJobHistory  group by company, review  order by company;  begin  dbms\_output.put\_line(rpad('기업명', 15) || rpad('지원자리뷰', 15) || rpad('리뷰수', 15));  dbms\_output.put\_line('----------------------------------------');  open vcursor;  loop  fetch vcursor into pcompany, preview, pcount;  exit when vcursor%notfound;    dbms\_output.put\_line(rpad(pcompany,15) || rpad(preview, 15) || rpad(pcount || '명',15));  end loop;  close vcursor;  end procCompanyReview;      -- procCompanySizeReview  -- 기업규모별 리뷰 통계 조회하는 프로시저  -- exec procCompanySizeReview;  create or replace procedure procCompanySizeReview  is  pcompanySize tblJobHistory.companySize%type;  preview tblJobHistory.review%type;  pcount number;  cursor vcursor  is  select companySize, review, count(\*) as cnt  from vwJobHistory  group by companySize, review  order by companySize;  begin  open vcursor;  dbms\_output.put\_line(rpad('기업규모', 15) || rpad('지원자리뷰', 15) || rpad('리뷰수', 15));  dbms\_output.put\_line('----------------------------------------');  loop  fetch vcursor into pcompanySize, preview, pcount;  exit when vcursor%notfound;  dbms\_output.put\_line(rpad(pcompanySize,15) || rpad(preview, 15) || rpad(pcount || '명',15));  end loop;  close vcursor;  end procCompanySizeReview;      -- procJobReview  -- 직무별 리뷰 통계 조회하는 프로시저  -- exec procJobReview;  create or replace procedure procJobReview  is  pjob tblJobHistory.job%type;  preview tblJobHistory.review%type;  pcount number;  cursor vcursor  is  select job, review, count(\*) as cnt from vwJobHistory  group by job, review  order by job;  begin  open vcursor;  dbms\_output.put\_line(rpad('지원직무', 30) || rpad('지원자리뷰', 15) || rpad('리뷰수', 15));  dbms\_output.put\_line('---------------------------------------------------');  loop  fetch vcursor into pjob, preview, pcount;  exit when vcursor%notfound;  dbms\_output.put\_line(rpad(pjob,30) || rpad(preview, 15) || rpad(pcount || '명',15));  end loop;  close vcursor;  end procJobReview;      -- 취업 히스토리 테이블(tblJobHistory)에서 데이터 삽입 또는 삭제시 동작하는 트리거  create or replace trigger trgInsDelJobHistory  after  insert or delete  on tblJobHistory  for each row  begin  if inserting then  dbms\_output.put\_line('추가항목: ' || :new.jobHistorySeq || '번');  elsif deleting then  dbms\_output.put\_line('삭제항목: ' || :old.jobHistorySeq || '번');  end if;  end; | | | |