**1조 DB Object 스크립트(PL-SQL)**

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-01 | 작성일 | 2022-05-19 |
| 요구사항명 | 관리자 계정 | 작성자 | 김성태 |
| DML | | | |
| 계정으로 로그인 후 관리자 기본정보 간단하게 출력  create or replace procedure admin\_P(  p\_admin\_ssn in number,  p\_result out sys\_refcursor  )  is  begin  open p\_result  for  select  admin.seq,  admin.name as "이름",  admin.tel as "연락처"  from admin  where ssn = p\_admin\_ssn;  end admin\_P; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-02 | 작성일 | 2022-05-19 |
| 요구사항명 | 기초정보 관리기능 | 작성자 | 김성태 |
| DML | | | |
| --과정 명 관리  select \* from curriculum;  --과정 명 추가  create or replace procedure c\_insert\_p(p\_seq in curriculum.seq%type, p\_name in curriculum.name%type)  is  begin  insert into curriculum(seq,name) values(p\_seq,p\_name);  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end c\_insert\_p;  --과정 명 수정  CREATE OR REPLACE PROCEDURE c\_update\_p ( cur\_seq\_p IN NUMBER, cur\_name\_p IN VARCHAR2)  IS  BEGIN  UPDATE curriculum  SET seq = cur\_seq\_p,  name = cur\_name\_p  WHERE seq = cur\_seq\_p;  EXCEPTION WHEN OTHERS THEN  DBMS\_OUTPUT.put\_line('추가에 실패했습니다.');  ROLLBACK;  END;  --과정 명 삭제  create or replace procedure c\_delete\_p(p\_seq in curriculum.seq%type)  is  begin  delete from curriculum  where p\_seq = seq;  commit;  end c\_delete\_p;  --과목 명 관리  select \* from subject;  --과목 명 추가  create or replace procedure s\_insert\_p(p\_seq in subject.seq%type, p\_subject\_name IN subject.name%type)  is  begin  insert into subject(seq, subject\_name) values(p\_seq,p\_subject\_name);  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end s\_insert\_p;  --과목 명 수정  create or replace procedure s\_update\_p ( sub\_seq\_p in number, sub\_subject\_name\_p in varchar2)  is  begin  update subject  set seq = sub\_seq\_p,  subject\_name = sub\_subject\_name\_p  where seq = sub\_seq\_p;  exception when others then  dbms\_output.put\_line('추가에 실패했습니다.');  rollback;  end;  --과목 명 삭제  create or replace procedure s\_delete\_p(p\_seq in subject.seq%type)  is  begin  delete from subject  where p\_seq = seq;  commit;  end s\_delete\_p;  --강의실 명 관리  select \* from lectureroom;  --강의실 명 추가  create or replace procedure l\_insert\_p(p\_seq in lectureroom.seq%type, p\_rname IN lectureroom.room\_name%type, p\_sl in lectureroom.student\_limit%type)  is  begin  insert into lectureroom(seq, room\_name, student\_limit) values(p\_seq,p\_rname, p\_sl);  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end l\_insert\_p;  --강의실 명 수정  create or replace procedure l\_update\_p ( lect\_seq\_p in number, lect\_room\_name\_p in varchar2, lect\_student\_limit\_p in number)  is  begin  update lectureroom  set seq = lect\_seq\_p,  room\_name = lect\_room\_name\_p,  student\_limit = lect\_student\_limit\_p  where seq = lect\_seq\_p;  exception when others then  dbms\_output.put\_line('추가에 실패했습니다.');  rollback;  end;  --강의실 명 삭제  create or replace procedure l\_delete\_p(p\_seq in lectureroom.seq%type)  is  begin  delete from lectureroom  where p\_seq = seq;  commit;  end l\_delete\_p;  --교재 명 관리  select \* from book;  --교재 명 추가  create or replace procedure b\_insert\_p(p\_seq in book.seq%type, p\_sub IN book.subject%type, p\_aut in book.author%type,  p\_pub in book.publisher%type, p\_puye in book.pub\_year%type)  is  begin  insert into book(seq, subject, author, publisher, pub\_year) values(p\_seq, p\_sub, p\_aut, p\_pub, p\_puye);  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end b\_insert\_p;  --교재 명 수정  create or replace procedure b\_update\_p ( b\_seq\_p in number, b\_subject\_p in varchar2, b\_author\_p in varchar2,  b\_publisher\_p in varchar2, b\_pub\_year\_p in date)  is  begin  update book  set seq = b\_seq\_p,  subject = b\_subject\_p,  author = b\_author\_p,  publisher = b\_publisher\_p,  pub\_year = b\_pub\_year\_p  where seq = b\_seq\_p;  exception when others then  dbms\_output.put\_line('추가에 실패했습니다.');  rollback;  end;  --교재 명 삭제  create or replace procedure b\_delete\_p(p\_seq in book.seq%type)  is  begin  delete from book  where p\_seq = seq;  commit;  end b\_delete\_p; | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 요구사항번호 | B-03 | | 작성일 | | | 2022-05-19 |
| 요구사항명 | 교사계정 관리기능 | | 작성자 | | | 김성태 |
| DML | | | | | | |
| --교사정보 등록  create or replace procedure teacher\_insert\_p(p\_seq in teacher.seq%type, p\_name IN teacher.name%type, p\_ssn in teacher.ssn%type,  p\_tel in teacher.tel%type)  is  begin  insert into teacher(seq, name, ssn, tel) values(p\_seq, p\_name, p\_ssn, p\_tel);  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end teacher\_insert\_p;  --교사정보 수정  create or replace procedure t\_update\_p ( t\_seq\_p in number, t\_name\_p in varchar2, t\_ssn\_p in number, t\_tel\_p in varchar2)  is  begin  update teacher  set seq = t\_seq\_p,  name = t\_name\_p,  ssn = t\_ssn\_p,  tel = t\_tel\_p  where seq = t\_seq\_p;  exception when others then  dbms\_output.put\_line('추가에 실패했습니다.');  rollback;  end;  --교사정보 삭제  create or replace procedure teacher\_delete\_p(p\_seq in teacher.seq%type)  is  begin  delete from teacher  where p\_seq = seq;  commit;  end teacher\_delete\_p;  --교사의 강의과목에 대한 정보 조회  create or replace procedure open\_subject\_book\_P (  p\_open\_subject\_seq in number,  p\_result out sys\_refcursor  )  is  begin  open p\_result  for  select subject  from book b  inner join subject\_book s  on b.seq = s.book\_seq  where s.subject\_seq = (select subject\_seq from open\_subject where open\_subject. p\_open\_subject\_seq);  end open\_subject\_book\_P;  set SERVEROUTPUT ON;  declare  v\_result sys\_refcursor;  v\_open\_subject\_seq open\_subject.seq%type;  v\_open\_subject\_state varchar2(15);  v\_curriculum\_name curriculum.name%type;  v\_open\_curs\_begin open\_curs.begin\_date%type;  v\_open\_curs\_end open\_curs.end\_date%type;  v\_lectureroon\_name lectureroom.room\_name%type;  v\_open\_subject\_begin open\_subject.begin\_date%type;  v\_open\_subject\_end open\_subject.end\_date%type;  v\_book\_name sys\_refcursor;  v\_subject\_name subject.subject\_name%type;  v\_student\_num number;  v\_seq number;  begin  teacher\_subject\_P(1, v\_result);  v\_seq :=1;  dbms\_output.put\_line('==================================강의 스케줄 조회===================================');    loop  fetch v\_result into v\_open\_subject\_seq, v\_open\_subject\_state, v\_curriculum\_name, v\_open\_curs\_begin, v\_open\_curs\_end, v\_lectureroon\_name,  v\_subject\_name, v\_open\_subject\_begin, v\_open\_subject\_end, v\_student\_num;  exit when v\_result%notfound;          dbms\_output.put\_line('-----------------------------------------------------------------------------------');  dbms\_output.put\_line(v\_seq ||'. '||'['||v\_open\_subject\_state||'] '||v\_subject\_name||' (과목번호: '||v\_open\_subject\_seq||')');  dbms\_output.put\_line(' - 기간: '||v\_open\_subject\_begin||' ~ '||v\_open\_subject\_end);  dbms\_output.put\_line(' - 인원: '||v\_student\_num||'명');  dbms\_output.put\_line(' - 교재: '||v\_book\_name);  dbms\_output.put\_line(' - 과정: '||v\_curriculum\_name || ' ('||v\_open\_curs\_begin||' ~ '||v\_open\_curs\_end||')');  dbms\_output.put\_line(' - 강의실: '||v\_lectureroon\_name);    v\_seq := v\_seq + 1;  end loop;  close v\_result;  dbms\_output.put\_line('-----------------------------------------------------------------------------------');  dbms\_output.put\_line('===================================================================================');  end; | | | | | | |
| 요구사항번호 | | B-04 | | 작성일 | 2022-05-20 | |
| 요구사항명 | | 과정 수료 처리 | | 작성자 | 임종운 | |
| DML | | | | | | |
| SELECT      s.seq, m.name, s.drop\_out, s.drop\_out\_date  FROM open\_curs oc      INNER JOIN student s ON oc.seq = s.open\_curs\_seq      INNER JOIN member m ON s.member\_seq = m.seq  WHERE oc.seq = 12;  CREATE OR REPLACE PROCEDURE change\_complete\_date\_P(       course\_seq in  number,  -- 수료 완료과정       complete\_date  in date  )  AS  BEGIN      UPDATE student s SET s.drop\_out = '수료', s.drop\_out\_date = complete\_date      WHERE          EXISTS(              SELECT \*              FROM member m              WHERE s.member\_seq = m.seq              AND s.open\_curs\_seq = course\_seq          )          AND          (s.drop\_out IS NULL          OR          s.drop\_out<>'중도탈락');  END;  /  --과정번호, 수료일자 입력  EXECUTE change\_complete\_date\_P(12, '2022-05-29'); | | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-04 | 작성일 | 2022-05-20 |
| 요구사항명 | 과정 등록인원 출력 | 작성자 | 임종운 |
| DML | | | |
| CREATE OR REPLACE TYPE curs\_member\_obj AS OBJECT(      name varchar2(50),      ssn varchar2(50),      tel varchar2(50),      regdate date,      drop\_out varchar2(15)  );  /  CREATE OR REPLACE TYPE member\_tables AS TABLE OF curs\_member\_obj;  /  CREATE OR REPLACE FUNCTION curs\_member\_list\_F  (      input\_curs\_seq NUMBER  ) RETURN member\_tables  IS      v\_rtn member\_tables;  BEGIN      SELECT          curs\_member\_obj(rs.name, rs.ssn, rs.tel, rs.regdate, rs.drop\_out)          BULK COLLECT INTO v\_rtn      FROM (SELECT          m.name, m.ssn, m.tel, s.regdate, s.drop\_out      FROM (SELECT \*      FROM open\_curs oc      WHERE seq = input\_curs\_seq) rs      --seq가 확일할 과정 번호          INNER JOIN enrollment e ON rs.seq = e.open\_curs\_seq          INNER JOIN member m ON e.member\_seq = m.seq          INNER JOIN student s ON m.seq = s.member\_seq) rs;        RETURN v\_rtn;    END curs\_member\_list\_F;  /  DROP TYPE member\_tables;  DROP TYPE curs\_member\_obj;  DROP FUNCTION curs\_member\_list;  SELECT \* FROM TABLE(curs\_member\_list\_F(12)); | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-05 | 작성일 | 2022-05-20 |
| 요구사항명 | 개설 과정의 과목 정보 출력 | 작성자 | 임종운 |
| DML | | | |
| CREATE OR REPLACE TYPE curs\_info\_obj AS OBJECT(      subject\_name varchar2(50),      begin\_date date,      end\_date date,      subject varchar2(100),      name varchar2(15)  );  /  CREATE OR REPLACE TYPE curs\_info\_tables AS TABLE OF curs\_info\_obj;  /  CREATE OR REPLACE FUNCTION search\_curs\_info\_list\_F  (      input\_curs\_seq NUMBER  ) RETURN curs\_info\_tables  IS      v\_rtn curs\_info\_tables;  BEGIN        SELECT          curs\_info\_obj(info.subject\_name, info.begin\_date, info.end\_date, info.subject, info.name)          BULK COLLECT INTO v\_rtn      FROM (SELECT          s.subject\_name, os.begin\_date, os.end\_date, b.subject ,t.name      FROM (SELECT seq  FROM open\_curs oc WHERE seq = 1) sr          INNER JOIN open\_subject os ON sr.seq = os.open\_curs\_seq          INNER JOIN subject s ON os.subject\_seq = s.seq          INNER JOIN subject\_book sb ON s.seq = sb.subject\_seq          INNER JOIN book b ON sb.book\_seq = b.seq          INNER JOIN teacher t ON os.teacher\_seq = t.seq) info;        RETURN v\_rtn;    END search\_curs\_info\_list\_F;  /  DROP TYPE curs\_info\_obj;  DROP TYPE curs\_info\_tables;  DROP FUNCTION search\_curs\_info\_list\_F;  SELECT \* FROM TABLE(search\_curs\_info\_list\_F(12)); | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-07 | 작성일 | 2022-05-19 |
| 요구사항명 | 시험 관리 및 성적 조회 기능 | 작성자 | 강지윤 |
| DML | | | |
| **--개설 과목별(수강종료, 수강중인 개설과목만)로 시험 문제 파일 등록 여부(필기,실기),  성적 등록 여부(필기,실기,출결)확인**  --프로시저 호출시 원하는 개설과목명을 입력  --프로지서명 : subject\_info\_P  --subject\_info\_P(<개설과목명>)으로 호출.  create or replace procedure subject\_info\_P (  psubject\_name varchar2 := ''  )  is  jname subject.subject\_name%type;  osseq open\_subject.seq%type;  ossubject\_seq open\_subject.subject\_seq%type;  ocseq open\_curs.seq%type;  cname curriculum.name%type;  a1 varchar2(1);  a2 varchar2(1);  b1 varchar2(1);  b2 varchar2(1);  b3 varchar2(1);  cursor vcursor is  select  DISTINCT j.subject\_name as 과목명,  os.seq as 개설과목번호,  os.subject\_seq as 과목번호,  oc.seq as 개설과정번호,  c.name as 개설과정명,  (case when rq.written\_test\_question = '시험문제등록안함' then 'N' else 'Y' end) as 필기시험문제등록여부,  (case when sq.skill\_test\_question = '시험문제등록안함' then 'N' else 'Y' end) as 실기시험문제등록여부,  (case when ss.written\_score is not null then 'Y' else 'N' end) as 필기성적등록여부,  (case when ss.skill\_score is not null then 'Y' else 'N' end) as 실기성적등록여부,  (case when ss.attendance\_score is not null then 'Y' else 'N' end) as 출결성적등록여부    from open\_subject os  inner join subject\_score ss on ss.open\_subject\_seq = os.seq  inner join subject j on j.seq = os.subject\_seq  inner join open\_curs oc on oc.seq = os.open\_curs\_seq  inner join curriculum c on c.seq = oc.curs\_seq  inner join written\_test r on r.open\_subject\_seq = os.seq  inner join skill\_test s on s.open\_subject\_seq = os.seq  inner join written\_test\_question rq on rq.written\_test\_seq = r.seq  inner join skill\_test\_question sq on sq.skill\_test\_seq = s.seq    where j.subject\_name = psubject\_name  order by oc.seq;  begin  open vcursor;  loop  fetch vcursor into jname, osseq, ossubject\_seq, ocseq, cname, a1, a2, b1, b2, b3;  exit when vcursor%notfound;    dbms\_output.put\_line('-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  dbms\_output.put\_line('과목명: ' || jname || ' | 개설과목번호 : ' || osseq || ' | 과목번호: ' || ossubject\_seq || ' | 개설과정번호: ' || ocseq || ' | 개설과정명: ' || cname || ' | 필기시험문제 등록여부: ' || a1 ||  ' | 실기시험문제 등록여부: ' || a2 || ' | 필기성적등록여부 : ' || b1 || ' | 실기성적등록여부 : ' || b2 || ' | 출결성적등록여부 : ' || b3);  dbms\_output.put\_line('-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');    end loop;  close vcursor;  end;  --호출  --subject\_info\_P('개설 과목명')  --ex. ‘java’의 개설과목정보와 시험문제파일등록여부, 성적등록여부를 알고싶을때  begin  subject\_info\_P('java');  end;  **--개설 과목별 성적 출력시 개설과목별 수강생의 성적정보 (교육생 이름, 주민번호 뒷자리, 필기점수, 실기점수,출결점수) 출력**  --프로시저 호출시 원하는 개설과목명을 입력  --프로지서명 subject\_score\_check\_P  -- subject\_score\_check\_P(<개설과목명>)으로 호출.  create or replace procedure subject\_score\_check\_P (  popen\_subject\_name varchar2 := ''  )  is  stseq student.seq%type;  mname member.name%type;  mssn member.ssn%type;  ssubject\_name subject.subject\_name%type;  sswritten\_score subject\_score.written\_score%type;  ssskill\_score subject\_score.skill\_score%type;  ssattendance\_score subject\_score.attendance\_score%type;  cursor vcursor is  select  st.seq as "교육생 번호",  m.name as "교육생 이름",  m.ssn as "주민번호 뒷자리",  s.subject\_name as "개설과목명",  ss.written\_score as 필기점수,  ss.skill\_score as 실기점수,  ss.attendance\_score as 출결점수  from student st  inner join member m on m.seq = st.member\_seq  inner join subject\_score ss on ss.student\_seq = st.seq  inner join open\_subject os on os.seq = ss.open\_subject\_seq  inner join subject s on s.seq = os.subject\_seq  where s.subject\_name = popen\_subject\_name;  begin  open vcursor;  loop  fetch vcursor into stseq, mname, mssn, ssubject\_name, sswritten\_score, ssskill\_score, ssattendance\_score;    exit when vcursor%notfound;    dbms\_output.put\_line('---------------------------------------------------------------------------------------');  dbms\_output.put\_line('교육생번호: ' || stseq || ' | 교육생이름: ' || mname || ' | 개설과목명: ' || ssubject\_name || ' | 필기점수 : ' || sswritten\_score || '점'  || ' | 실기점수 : ' || ssskill\_score || '점' || ' | 출결점수 : ' || ssattendance\_score || '점');  dbms\_output.put\_line('---------------------------------------------------------------------------------------');  end loop;  close vcursor;  end;  --호출시 subject\_score\_check\_P('개설과목명')  --ex. ‘java’를 수강하는 모든 교육생의 성적정보를 알고싶을때  begin  subject\_score\_check\_P('java');  end;  **--4.교육생 개인별 출력시 교육생정보+교육생이 수강한 모든 개설과목에 대한 성적 정보**  --(교육생번호, 교육생 이름, 주민번호 뒷자리, 개설과정명, 개설과정시작, 개설과정종료, 강의실, 개설 과목명, 교사명, 개설과목시작, 개설과목종료날짜, 필기, 실기, 출결)  --프로시저 호출시 원하는 교육생이름을 입력  --프로지서명 : student\_score\_check\_P  -- student\_score\_check\_P (<교육생이름>)으로 호출.  create or replace procedure student\_score\_check\_P (  pstudent\_name varchar2 := ''  )  is  ssubject\_name subject.subject\_name%type;  stseq student.seq%type;  mname member.name%type;  mssn member.ssn%type;  cname curriculum.name%type;  ocbegin\_date open\_curs.begin\_date%type;  ocend\_date open\_curs.end\_date%type;  lroom\_name lectureroom.room\_name%type;  tname teacher.name%type;  osbegin\_date open\_subject.begin\_date%type;  osend\_date open\_subject.end\_date%type;  sswritten\_score subject\_score.written\_score%type;  ssskill\_score subject\_score.skill\_score%type;  ssattendance\_score subject\_score.attendance\_score%type;  cursor vcursor is    select  distinct s.subject\_name as "개설과목명",  st.seq as "교육생 번호",  m.name as "교육생 이름",  m.ssn as "주민번호 뒷자리",  c.name as 과정명,  oc.begin\_date as "과정시작날짜",  oc.end\_date as "과정종료날짜",  l.room\_name as 강의실명,  t.name as 교사명,  os.begin\_date as "과목시작날짜",  os.end\_date as "과목종료날짜",  ss.written\_score as 필기점수,  ss.skill\_score as 실기점수,  ss.attendance\_score as 출결점수  from student st  inner join member m on m.seq = st.member\_seq  inner join subject\_score ss on ss.student\_seq = st.seq  inner join open\_subject os on os.seq = ss.open\_subject\_seq  inner join subject s on s.seq = os.subject\_seq  inner join open\_curs oc on oc.seq = os.open\_curs\_seq  inner join curriculum c on oc.curs\_seq = c.seq  inner join lectureroom l on oc.room\_seq = l.seq  inner join teacher t on os.teacher\_seq = t.seq  inner join subject\_book sb on sb.subject\_seq = s.seq  inner join book b on sb.book\_seq = b.seq    where m.name = pstudent\_name;  begin  open vcursor;  loop  fetch vcursor into ssubject\_name, stseq, mname, mssn, cname, ocbegin\_date, ocend\_date, lroom\_name, tname, osbegin\_date, osend\_date, sswritten\_score, ssskill\_score, ssattendance\_score;  exit when vcursor%notfound;  dbms\_output.put\_line('-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  dbms\_output.put\_line('교육생번호: ' || stseq || ' | 교육생이름: ' || mname || ' | 주민번호 뒷자리: ' || mssn || ' | 개설과정명: ' || cname || ' | 개설과정시작: ' || ocbegin\_date || ' | 개설과정종료: ' || ocend\_date ||  ' | 강의실명: ' || lroom\_name || ' | 개설과목명: ' || ssubject\_name || ' | 교사명: ' || tname || ' | 개설과목시작: ' || osbegin\_date || ' | 개설과목종료: ' || osend\_date ||  ' | 필기점수: ' || sswritten\_score || '점' || ' | 실기점수: ' || ssskill\_score || '점' || ' | 출결점수: ' || ssattendance\_score || '점');    dbms\_output.put\_line('-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');    end loop;  close vcursor;  end;  --호출시 student\_score\_check\_P('교육생이름');  --ex. 교육생 ‘홍영화’의 기본 정보와 ‘홍영화’가 수강한 모든 개설과목에 대한 성적 정보를 알고 싶을 때  begin  student\_score\_check\_P('홍영화');  end; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-08 | 작성일 | 2022-05-19 |
| 요구사항명 | 출결 관리 및 출결 조회 | 작성자 | 고민지 |
| DML | | | |
| **-- 1. 개설과정별 교육생 출결 조회**  create or replace procedure admin\_attendance\_inquiry\_P(        p\_open\_curs\_seq in number,        p\_result      out sys\_refcursor  )  is  begin      open p\_result      for      select          "학생번호",          count(decode("출석", '출석', 1)) as "출석일수",          count(decode("지각조퇴", '출석..', 1)) as "정상일수",          count(decode("지각조퇴", '출석지각.', 1)) as "지각일수",          count(decode("지각조퇴", '출석.조퇴', 1)) as "조퇴일수",          count(decode("지각조퇴", '출석지각조퇴', 1)) as "지각 및 조퇴일수",          count(decode("지각조퇴", '외출', 1)) as "외출일수",          count(decode("지각조퇴", '결석', 1)) as "결석일수",          count(decode("출석", '병가', 1)) as "병가일수",          count(decode("출석", '기타', 1)) as "기타일수"      from       (select i.\*, 출석||지각||조퇴 as "지각조퇴" from attendance\_info i              where "학생번호" in (select seq from student where open\_curs\_seq = p\_open\_curs\_seq))                  group by "학생번호";    end admin\_attendance\_inquiry\_P;  **-- 1. 개설과정별 교육생 출결 조회 > 실행문**  declare      v\_student\_name  member.name%type;      v\_student\_seq   number; --학생번호      v\_attendance    number; --출석일수      v\_normal        number; --정상일수      v\_tardy         number; --지각일수      v\_early\_leaving number; --조퇴일수      v\_tardy\_leaving number; --지각 및 조퇴일수      v\_out           number; --외출일수      v\_absence       number; --결석일수      v\_sickness      number; --병가일수      v\_warrant       number; --기타일수      v\_result sys\_refcursor;      v\_seq           number;      v\_open\_curs\_seq     number;   v\_curs\_name      curriculum.name%type;      v\_curs\_begin     open\_curs.begin\_date%type;      v\_curs\_end       open\_curs.end\_date%type;  begin      v\_open\_curs\_seq := 12;        select c.name, o.begin\_date, o.end\_date into v\_curs\_name, v\_curs\_begin, v\_curs\_end      from open\_curs o         inner join curriculum c on o.curs\_seq = c.seq              where o.seq = v\_open\_curs\_seq;          dbms\_output.put\_line('===================================== 과정 정보 =======================================');      dbms\_output.put\_line('[과목] ' || v\_curs\_name ||'(과정번호: '||v\_open\_curs\_seq||')');      dbms\_output.put\_line('     - 기간: '||v\_curs\_begin||' ~ '||v\_curs\_end);      --dbms\_output.put\_line('     - 인원: '||subject\_student\_num(v\_open\_curs\_seq)||'명');      dbms\_output.put\_line('');        dbms\_output.put\_line('================================== 교육생 출결 조회 ===================================');        dbms\_output.put\_line('번호. 이름(학생번호)                                         단위: 일');      dbms\_output.put\_line('              [출석] [정상] [지각] [조퇴] [지각+조퇴] [결석] [외출] [병가] [기타]');      dbms\_output.put\_line('---------------------------------------------------------------------------------------');        admin\_attendance\_inquiry\_P(v\_open\_curs\_seq, v\_result);      v\_seq := 1;        loop          fetch v\_result into v\_student\_seq,          v\_attendance, v\_normal, v\_tardy, v\_early\_leaving, v\_tardy\_leaving,          v\_out, v\_absence, v\_sickness, v\_warrant;          exit when v\_result%notfound;          select name into v\_student\_name from member where seq = (select member\_seq from student where seq = v\_student\_seq);            dbms\_output.put\_line(v\_seq ||'.'||v\_student\_name||'('||v\_student\_seq||')   '||v\_attendance||'     '||v\_normal||'      '||v\_tardy||'     '||v\_early\_leaving||'      '||                              v\_tardy\_leaving||'            '||v\_absence||'      '||v\_out||'      '||v\_sickness||'      '||v\_warrant);            dbms\_output.put\_line('---------------------------------------------------------------------------------------');          v\_seq := v\_seq + 1;        end loop;      dbms\_output.put\_line('=======================================================================================');        close v\_result;  end;    **3. 특정 교육생의 출결현황 조회**  create or replace procedure student\_attendance\_inquiry\_P(      p\_student\_seq in number,      p\_result      out sys\_refcursor  )  is  begin      open p\_result      for      select \* from attendance\_info          where "학생번호" = p\_student\_seq              order by "날짜";  end;  **3. 특정 교육생의 출결현황 조회 > 실행문**  declare      v\_student\_seq       number;      v\_result            sys\_refcursor;      v\_attendance\_seq    attendance.seq%type;      v\_attendance\_date   attendance.attendance\_date%type;      v\_attendance        varchar2(15);      v\_tardy             varchar2(10);      v\_early\_leaving     varchar2(10);      v\_out               varchar2(10);      v\_temp              number;      v\_member\_info       member%rowtype;      v\_curs\_name         curriculum.name%type;      v\_regdate           date;      v\_seq               number;  begin      v\_student\_seq := 3;        student\_attendance\_inquiry\_P(v\_student\_seq, v\_result);        dbms\_output.put\_line('================================== 교육생 출결 조회 ===================================');        select \* into v\_member\_info from member where seq = (select member\_seq from student where seq = v\_student\_seq);        select c.name, s.regdate into v\_curs\_name, v\_regdate from student s          inner join open\_curs o on s.open\_curs\_seq = o.seq              inner join curriculum c on o.curs\_seq = c.seq                  where s.seq = v\_student\_seq;        dbms\_output.put\_line('이름: '||v\_member\_info.name||' (교육생 번호:'||v\_student\_seq||')');      dbms\_output.put\_line('등록일: '||v\_regdate);      dbms\_output.put\_line('전화번호: '||v\_member\_info.tel);      dbms\_output.put\_line('수강과정: '||v\_curs\_name);      dbms\_output.put\_line('');      dbms\_output.put\_line('================================== 교육생 세부 출결 ===================================');        v\_seq := 1;        loop          fetch v\_result into v\_temp, v\_attendance\_seq, v\_attendance\_date, v\_attendance, v\_tardy, v\_early\_leaving, v\_out;          exit when v\_result%notfound;            v\_tardy := replace(v\_tardy, '.', '');          v\_early\_leaving := replace(v\_early\_leaving, '.', '');          v\_out := replace(v\_out, '.', '');            if v\_attendance <> '(주말)' and  v\_attendance <> '(공휴일)'          then              dbms\_output.put\_line('---------------------------------------------------------------------------------------');              dbms\_output.put\_line(v\_seq||'. '||v\_attendance\_date||'    '||v\_attendance||' '||v\_tardy||' '||v\_early\_leaving||' '||v\_out);              v\_seq := v\_seq + 1;          end if;        end loop;        dbms\_output.put\_line('=======================================================================================');        close v\_result;    end; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-09 | 작성일 | 2022-05-19 |
| 요구사항명 | 취업 공고 관리 및 조회 | 작성자 | 강지윤 |
| DML | | | |
| ▶ where절 < > 위치에는 <>안에 있는 내용에 해당하는 데이터 입력  **--1. 전체 취업 공고조회(회사명, 지역, 분야, 계약형태, 모집인원, 모집시작날짜, 모집종료날짜)**  --뷰이름 : job\_announce\_V  create or replace view job\_announce\_V  as  (  select  a.company as 회사명,  city.city as 지역,  job\_field.field\_name as 분야,  contract\_type.contract\_type as 계약형태,  a.people as 모집인원,  a.begin\_date as 모집시작날짜,  a.end\_date as 모집종료날짜  from job\_announce a  inner join city on city.seq = a.city\_seq  inner join job\_field on job\_field.seq = a.field\_seq  inner join contract\_type on contract\_type.seq = a.contract\_seq  )  order by 모집시작날짜;  --전체 취업 공고를 조회하고 싶을 때  select \* from job\_announce\_V;  **--2. 기간별 조회(모집종료날짜가 'yyyy-mm-dd'부터 'yyyy-mm-dd' 사이인 기간 조회)**  --프로시저 호출시 원하는 기간(기간시작날짜, 기간종료날짜) 입력  --프로지서명 : period\_job\_check\_P  -- period\_job\_check\_P (<’기간시작날짜’>, <’기간종료날짜’>)으로 호출.  create or replace procedure period\_job\_check\_P (  start\_period varchar2 := '',  end\_period varchar2 := ''  )  is  acompany job\_announce.company%type;  ccity city.city%type;  ffield\_name job\_field.field\_name%type;  ctcontract\_type contract\_type.contract\_type%type;  apeople job\_announce.people%type;  abegin\_date job\_announce.begin\_date%type;  aend\_date job\_announce.end\_date%type;  cursor vcursor is  select  a.company as 회사명,  c.city as 지역,  f.field\_name as 분야,  ct.contract\_type as 계약형태,  a.people as 모집인원,  a.begin\_date as 모집시작날짜,  a.end\_date as 모집종료날짜  from job\_announce a  inner join city c on c.seq = a.city\_seq  inner join job\_field f on f.seq = a.field\_seq  inner join contract\_type ct on ct.seq = a.contract\_seq  where a.end\_date between start\_period and end\_period order by 모집시작날짜;  begin  open vcursor;  loop  fetch vcursor into acompany, ccity, ffield\_name, ctcontract\_type, apeople, abegin\_date, aend\_date;  exit when vcursor%notfound;  dbms\_output.put\_line('-------------------------------------------------------------------------------------------------------------');  dbms\_output.put\_line('회사명: ' || acompany || ' | 지역: ' || ccity || ' | 분야: ' || ffield\_name || ' | 계약형태: ' || ctcontract\_type || ' | 모집인원: ' || apeople || '명' || ' | 모집시작날짜: ' ||  abegin\_date || ' | 모집종료날짜: ' || aend\_date);  dbms\_output.put\_line('-------------------------------------------------------------------------------------------------------------'); end loop;  close vcursor;  end;  --모집종료날짜의 기간 입력  -- period\_job\_check\_P('기간시작날짜', '기간종료날짜')  ex. 모집종료 날짜가 2022-05-07~2022-05-31 사이인 취업공고를 알고싶을때  begin  period\_job\_check\_P('2022-05-07', '2022-05-31');  end;  **--3.분야별 (분야명으로 검색 ex. 웹개발) 조회**  --프로시저 호출시 원하는 분야명을 입력  --프로지서명 : field\_job\_check\_P  -- field\_job\_check\_P (<’분야명’>)으로 호출.  create or replace procedure field\_job\_check\_P (  pjob\_field varchar2 := ''  )  is  acompany job\_announce.company%type;  ccity city.city%type;  ffield\_name job\_field.field\_name%type;  ctcontract\_type contract\_type.contract\_type%type;  apeople job\_announce.people%type;  abegin\_date job\_announce.begin\_date%type;  aend\_date job\_announce.end\_date%type;  cursor vcursor is  select  a.company as 회사명,  c.city as 지역,  f.field\_name as 분야,  ct.contract\_type as 계약형태,  a.people as 모집인원,  a.begin\_date as 모집시작날짜,  a.end\_date as 모집종료날짜  from job\_announce a  inner join city c on c.seq = a.city\_seq  inner join job\_field f on f.seq = a.field\_seq  inner join contract\_type ct on ct.seq = a.contract\_seq    where f.field\_name = pjob\_field order by 모집시작날짜;  begin  open vcursor;  loop  fetch vcursor into acompany, ccity, ffield\_name, ctcontract\_type, apeople, abegin\_date, aend\_date;  exit when vcursor%notfound;    dbms\_output.put\_line('-------------------------------------------------------------------------------------------------------------');    dbms\_output.put\_line('회사명: ' || acompany || ' | 지역: ' || ccity || ' | 분야: ' || ffield\_name || ' | 계약형태: ' || ctcontract\_type || ' | 모집인원: ' || apeople || '명' || ' | 모집시작날짜: ' ||  abegin\_date || ' | 모집종료날짜: ' || aend\_date);  dbms\_output.put\_line('-------------------------------------------------------------------------------------------------------------');  end loop;  close vcursor;    end;  --호출시 field\_job\_check\_P('분야명');  --ex. ‘웹 개발’분야의 취업공고를 확인하고싶을때  begin  field\_job\_check\_P('웹 개발');  end;  **--4. 지역별(지역명으로 검색 - ex. 서울) 조회**  --프로시저 호출시 원하는 지역명을 입력  --프로지서명 : city\_job\_check\_P  -- city\_job\_check\_P (<’지역명’’>)으로 호출.  create or replace procedure city\_job\_check\_P (  pcity varchar2 := ''  )  is  acompany job\_announce.company%type;  ccity city.city%type;  ffield\_name job\_field.field\_name%type;  ctcontract\_type contract\_type.contract\_type%type;  apeople job\_announce.people%type;  abegin\_date job\_announce.begin\_date%type;  aend\_date job\_announce.end\_date%type;  cursor vcursor is  select  a.company as 회사명,  c.city as 지역,  f.field\_name as 분야,  ct.contract\_type as 계약형태,  a.people as 모집인원,  a.begin\_date as 모집시작날짜,  a.end\_date as 모집종료날짜  from job\_announce a  inner join city c on c.seq = a.city\_seq  inner join job\_field f on f.seq = a.field\_seq  inner join contract\_type ct on ct.seq = a.contract\_seq    where c.city = pcity order by 모집시작날짜;    begin  open vcursor;  loop  fetch vcursor into acompany, ccity, ffield\_name, ctcontract\_type, apeople, abegin\_date, aend\_date;  exit when vcursor%notfound;    dbms\_output.put\_line('-------------------------------------------------------------------------------------------------------------');      dbms\_output.put\_line('회사명: ' || acompany || ' | 지역: ' || ccity || ' | 분야: ' || ffield\_name || ' | 계약형태: ' || ctcontract\_type || ' | 모집인원: ' || apeople || '명' || ' | 모집시작날짜: ' ||  abegin\_date || ' | 모집종료날짜: ' || aend\_date);      dbms\_output.put\_line('-------------------------------------------------------------------------------------------------------------');    end loop;  close vcursor;    end;  --호출시 city\_job\_check\_P(‘<지역명>’)  --ex. ‘서울’지역의 취업공고를 확인하고싶을 때  begin    city\_job\_check\_P('서울');    end; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-02 | 작성일 | 2022-05-19 |
| 요구사항명 | 강의 스케줄 조회 기능 | 작성자 | 고민지 |
| DML | | | |
| **-- 교육생 등록인원 function**      create or replace function subject\_student\_num(          pseq open\_subject.seq%type      ) return number      is          v\_student\_num number;      begin          select case                  when s.begin\_date > sysdate  -- 강의예정                      then (select count(\*) from student where open\_curs\_seq = c.seq                          and (drop\_out\_date is null or (drop\_out\_date >= s.begin\_date)))                  when s.end\_date < sysdate    -- 강의종료                      then (select count(\*) from student where open\_curs\_seq = c.seq                          and (drop\_out\_date is null or (drop\_out = '수료') or ((drop\_out = '중도탈락') and (drop\_out\_date > s.end\_date))))                  else (select count(\*) from student where open\_curs\_seq = c.seq  --'강의 중'                          and (drop\_out\_date is null or (drop\_out\_date > sysdate)))              end into v\_student\_num          from open\_subject s              inner join open\_curs c                  on s.open\_curs\_seq = c.seq                      where s.seq = pseq;            return v\_student\_num;      end;  **-- 1. 로그인한 교사의 강의정보를 강의예정, 강의 중, 강의 종료로 구분하여 조회**  create or replace procedure teacher\_subject\_inquiry\_P(      p\_teacher\_seq in number,      p\_result out sys\_refcursor  )  is  begin      open p\_result      for      select          s.seq as "과목 번호",          case              when s.begin\_date > sysdate then '강의 예정'              when s.end\_date < sysdate then '강의 종료'              else '강의 중'          end as "진행 상태",         (select name from curriculum where seq = c.curs\_seq) as "과정명",          c.begin\_date as "과정 시작일",          c.end\_date as "과정 종료일",          (select room\_name from lectureroom where seq = c.room\_seq) as "강의실",          (select subject\_name from subject where seq = s.subject\_seq) as "과목명",          s.begin\_date as "과목 시작일",          s.end\_date as "과목 종료일",          subject\_student\_num(s.seq) as "교육생 등록인원"      from open\_subject s          inner join open\_curs c              on s.open\_curs\_seq = c.seq                  where teacher\_seq = p\_teacher\_seq                      order by s.begin\_date;  end teacher\_subject\_inquiry\_P;  **-- 1. 과목별 교재명을 조회**  create or replace procedure open\_subject\_book\_P (      p\_open\_subject\_seq in number,      p\_result out sys\_refcursor  )  is  begin      open p\_result      for      select subject      from book b          inner join subject\_book s              on b.seq = s.book\_seq                  where s.subject\_seq = (select subject\_seq from open\_subject where open\_subject.seq = p\_open\_subject\_seq);  end open\_subject\_book\_P;  **-- 1. 로그인한 교사의 강의정보를 강의예정, 강의 중, 강의 종료로 구분하여 조회 > 출력문**  declare      v\_result sys\_refcursor;      v\_open\_subject\_seq open\_subject.seq%type;      v\_open\_subject\_state varchar2(15);      v\_curriculum\_name curriculum.name%type;      v\_open\_curs\_begin open\_curs.begin\_date%type;      v\_open\_curs\_end open\_curs.end\_date%type;      v\_lectureroon\_name lectureroom.room\_name%type;      v\_open\_subject\_begin open\_subject.begin\_date%type;      v\_open\_subject\_end open\_subject.end\_date%type;      v\_subject\_name subject.subject\_name%type;      v\_student\_num number;      v\_seq number;      v\_book\_name sys\_refcursor;      v\_temp book.subject%type;      v\_book\_seq number;  begin      teacher\_subject\_inquiry\_P(1, v\_result);      v\_seq :=1;      dbms\_output.put\_line('==================================강의 스케줄 조회===================================');        loop          fetch v\_result into v\_open\_subject\_seq, v\_open\_subject\_state, v\_curriculum\_name, v\_open\_curs\_begin, v\_open\_curs\_end, v\_lectureroon\_name,                              v\_subject\_name, v\_open\_subject\_begin, v\_open\_subject\_end, v\_student\_num;          exit when v\_result%notfound;                  dbms\_output.put\_line('-----------------------------------------------------------------------------------');          dbms\_output.put\_line(v\_seq ||'. '||'['||v\_open\_subject\_state||'] '||v\_subject\_name||' (과목번호: '||v\_open\_subject\_seq||')');          dbms\_output.put\_line('              - 기간: '||v\_open\_subject\_begin||' ~ '||v\_open\_subject\_end);          dbms\_output.put\_line('              - 인원: '||v\_student\_num||'명');            open\_subject\_book\_P (v\_open\_subject\_seq, v\_book\_name);            v\_book\_seq := 1;          loop              fetch v\_book\_name into v\_temp;              exit when v\_book\_name%notfound;              dbms\_output.put\_line('              - 교재'||v\_book\_seq||': '||v\_temp);              v\_book\_seq := v\_book\_seq + 1;          end loop;          close v\_book\_name;            dbms\_output.put\_line('              - 과정: '||v\_curriculum\_name || ' ('||v\_open\_curs\_begin||' ~ '||v\_open\_curs\_end||')');          dbms\_output.put\_line('              - 강의실: '||v\_lectureroon\_name);             v\_seq := v\_seq + 1;      end loop;      close v\_result;      dbms\_output.put\_line('-----------------------------------------------------------------------------------');      dbms\_output.put\_line('===================================================================================');  end; | | | |
| **-- 2. 로그인한 교사의 특정 강의과목의 수강생 정보 조회**  create or replace procedure teacher\_student\_inquiry\_P(      p\_open\_subject\_seq in number,      p\_result out sys\_refcursor  )  is  begin      open p\_result      for      select          m.name as "이름",          m.tel as "전화번호",          t.regdate as "등록일",          case              when t.drop\_out = '중도탈락' and t.drop\_out\_date < s.end\_date then '중도탈락'              when s.begin\_date >= sysdate then '수강예정'              when (s.begin\_date < sysdate) and (sysdate < s.end\_date) then '수강중'              when s.end\_date <= sysdate then '수료'          end as "수강 상태"      from open\_subject s          inner join open\_curs c              on s.open\_curs\_seq = c.seq                  inner join student t                      on t.open\_curs\_seq = c.seq                          inner join member m                              on t.member\_seq = m.seq      where s.seq = p\_open\_subject\_seq;  end teacher\_student\_inquiry\_P;  **-- 2. 로그인한 교사의 특정 강의과목의 수강생 정보 조회 > 실행문**  declare      v\_seq number;      v\_name member.name%type;      v\_tel member.tel%type;      v\_regdate student.regdate%type;      v\_state varchar(15);      v\_result sys\_refcursor;      v\_open\_subject\_seq open\_subject.seq%type;      v\_subject\_name subject.subject\_name%type;      v\_subject\_begin open\_subject.begin\_date%type;      v\_subject\_end open\_subject.end\_date%type;      v\_book\_name sys\_refcursor;      v\_book book.subject%type;      v\_book\_seq number;  begin      v\_open\_subject\_seq := 1;        teacher\_student\_inquiry\_P(v\_open\_subject\_seq, v\_result);          select s.subject\_name, o.begin\_date, o.end\_date into v\_subject\_name, v\_subject\_begin, v\_subject\_end      from open\_subject o         inner join subject s on o.subject\_seq = s.seq              where o.seq = v\_open\_subject\_seq;          dbms\_output.put\_line('===================================== 과목 정보 =======================================');      dbms\_output.put\_line('[과목] ' || v\_subject\_name ||'(과목번호: '||v\_open\_subject\_seq||')');      dbms\_output.put\_line('     - 기간: '||v\_subject\_begin||' ~ '||v\_subject\_end);      dbms\_output.put\_line('     - 인원: '||subject\_student\_num(v\_open\_subject\_seq)||'명');        open\_subject\_book\_P (v\_open\_subject\_seq, v\_book\_name);        v\_book\_seq := 1;      loop          fetch v\_book\_name into v\_book;          exit when v\_book\_name%notfound;          dbms\_output.put\_line('     - 교재'||v\_book\_seq||': '||v\_book);          v\_book\_seq := v\_book\_seq + 1;      end loop;      close v\_book\_name;        dbms\_output.put\_line('');      dbms\_output.put\_line('================================== 교육생 정보 조회 ===================================');      dbms\_output.put\_line('---------------------------------------------------------------------------------------');      dbms\_output.put\_line('번호   이름          전화번호           등록일        수강상태');      v\_seq := 1;      loop          fetch v\_result into v\_name, v\_tel, v\_regdate, v\_state;          exit when v\_result%notfound;          dbms\_output.put\_line('---------------------------------------------------------------------------------------');          dbms\_output.put\_line(v\_seq||'     '||v\_name||'       '||v\_tel||'       '||v\_regdate||'       '||v\_state);          v\_seq := v\_seq + 1;      end loop;     if v\_result%rowcount = 0      then dbms\_output.put\_line('---------------------------------------------------------------------------------------');           dbms\_output.put\_line('등록한 교육생이 존재하지 않습니다.');      end if;      close v\_result;        dbms\_output.put\_line('===================================================================================');    end; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-03 | 작성일 | 2022-05-19 |
| 요구사항명 | 평가 관리 및 조회 | 작성자 | 고민지 |
| DML | | | |
| **-- 1. 배점 테이블에 배점 수정, 삭제시 성적데이터와의 무결성 유지**  create or replace trigger teacher\_point\_T      before update or delete      on point      for each row  declare      v\_score\_num number;      v\_subject\_seq number;  begin        if updating then          v\_subject\_seq := :new.open\_subject\_seq;      elsif deleting then          v\_subject\_seq := :old.open\_subject\_seq;      end if;        select count(\*) into v\_score\_num from subject\_score s where s.open\_subject\_seq = v\_subject\_seq;        if v\_score\_num > 0      then raise\_application\_error(-20100, '학생의 성적 데이터가 존재하여 배점 수정에 실패하였습니다.');      end if;  end;  **-- 2. 성적 테이블에 성적 입력, 수정시 배점데이터와의 무결성 유지**  create or replace trigger teacher\_score\_T      before insert or update      on subject\_score      for each row  declare      v\_attend\_point point.attend\_point%type;      v\_written\_point point.written\_point%type;      v\_skill\_point point.skill\_point%type;  begin        select p.attend\_point, p.written\_point, p.skill\_point          into v\_attend\_point, v\_written\_point, v\_skill\_point      from point p where p.open\_subject\_seq = :new.open\_subject\_seq;          if :new.attendance\_score > v\_attend\_point          or :new.written\_score > v\_written\_point          or :new.skill\_score > v\_skill\_point      then raise\_application\_error(-20101, '배점이상의 성적은 입력하여 성적 등록 및 수정에 실패하였습니다.');      end if;    end; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-04 | 작성일 | 2022-05-19 |
| 요구사항명 | 시험 관리 및 성적 조회 기능 | 작성자 | 권민채 |
| DML | | | |
| --과목 목록 출력  create or replace view subject\_catalog\_v as  select             open\_subject.seq as 과목번호,             curriculum.name as 과정명,              subject.subject\_name as 개설과목,              open\_subject.begin\_date as "과목 시작일",              open\_subject.end\_date as "과목 종료일",              lectureroom.room\_name as 강의실명,              teacher.name as 교사명  from open\_curs      inner join open\_subject on open\_subject.open\_curs\_seq = open\_curs.seq      inner join curriculum on open\_curs.curs\_seq = curriculum.seq      inner join subject on subject.seq = open\_subject.subject\_seq      inner join lectureroom on open\_curs.room\_seq = lectureroom.seq      inner join teacher on open\_subject.teacher\_seq = teacher.seq                order by open\_subject.seq;      -- 특정 과목번호의 교육생 정보 및 성적 조회  create or replace procedure member\_info\_score\_p(  msubjectseq number :=''  )  is  msubseq open\_subject.seq%type;  msubname subject.subject\_name%type;  mmemname member.name%type;  mmemtel member.tel%type;  mdrop student.drop\_out%type;  mwritten subject\_score.written\_score%type;  mskill subject\_score.skill\_score%type;  mattend subject\_score.attendance\_score%type;  cursor vcursor is  select  open\_subject.seq as 과목번호,  subject.subject\_name as 과목명,  member.name as 이름,  member.tel as 전화번호,  student.drop\_out as "중도탈락 여부",  subject\_score.written\_score as 필기점수,  subject\_score.skill\_score as 실기점수,  subject\_score.attendance\_score as 출결점수  from student  inner join member on member.seq = student.member\_seq  inner join subject\_score on subject\_score.student\_seq = student.seq  inner join open\_subject on open\_subject.seq = subject\_score.open\_subject\_seq  inner join subject on subject.seq = open\_subject.subject\_seq  where open\_subject.seq = msubjectseq;  begin  open vcursor;  loop  fetch vcursor into msubseq,msubname,mmemname,mmemtel,mdrop,mwritten,mskill,mattend;  exit when vcursor%notfound;  dbms\_output.put\_line(' | 과목번호: ' || msubseq || '  | 과목명 : ' || msubname || ' | 이름: ' || mmemname || ' | 전화번호: ' || mmemtel || ' | 중도탈락여부: ' || mdrop || ' | 필기점수: ' || mwritten || '점' ||                                                          ' | 실기점수: ' || mskill || '점' || ' | 출결점수 : ' || mattend || '점');  end loop;  close vcursor;  end;  --점수가 없는 학생들  create or replace view score\_null\_v as   select  open\_subject.seq as 과목번호,  subject.subject\_name as 과목명,  member.name as 이름,  student.seq as 교육생번호,  student.drop\_out as "중도탈락 여부",  subject\_score.written\_score as 필기점수,  subject\_score.skill\_score as 실기점수,  subject\_score.attendance\_score as 출결점수  from student  inner join member on member.seq = student.member\_seq  inner join subject\_score on subject\_score.student\_seq = student.seq  inner join open\_subject on open\_subject.seq = subject\_score.open\_subject\_seq  inner join subject on subject.seq = open\_subject.subject\_seq  where subject\_score.written\_score is null;  --시험점수 등록  create or replace procedure subject\_score\_p  (  w\_score in number  s\_score in number  a\_score in number  )  is  begin  update subject\_score  set  written\_score = w\_score  skill\_score = s\_score  attendance\_score = a\_score;  end; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-05 | 작성일 | 2022-05-19 |
| 요구사항명 | 출결관리 및 출결 조회 | 작성자 | 권민채 |
| DML | | | |
| --전체 출결 조회  create or replace procedure attendance\_check\_p(        pseq in number,        popencurs  in number,        pdatestart in date,        pdateend in date  is      vstudentnum student.seq%type;      출석일수  student.seq%type;      정상일수   student.seq%type;      지각일수   student.seq%type;      조퇴일수     student.seq%type;      지각조퇴     student.seq%type;      외출     student.seq%type;      결석     student.seq%type;       병가      student.seq%type;       기타      student.seq%type;      cursor vcursor is      select      "학생번호",      count(decode("출석", '출석', 1)) as "출석일수",      count(decode("지각조퇴", '출석..', 1)) as "정상일수",      count(decode("지각조퇴", '출석지각.', 1)) as "지각일수",      count(decode("지각조퇴", '출석.조퇴', 1)) as "조퇴일수",      count(decode("지각조퇴", '출석지각조퇴', 1)) as "지각 및 조퇴일수",      count(decode("지각조퇴", '외출', 1)) as "외출일수",      count(decode("지각조퇴", '결석', 1)) as "결석일수",      count(decode("출석", '병가', 1)) as "병가일수",      count(decode("출석", '기타', 1)) as "기타일수"  from      (select i.\*,      출석||지각||조퇴 as "지각조퇴"      from attendance\_info i where "학생번호" in (select seq from student where open\_curs\_seq =  popencurs )              and "날짜" between  pdatestart  and  pdateend)              where "학생번호"=pseq              group by "학생번호";        begin      open vcursor;          loop              fetch vcursor into   vstudentnum,  출석일수, 정상일수,  지각일수,  조퇴일수, 지각조퇴 ,외출,결석,병가,기타;              exit when vcursor%notfound;              dbms\_output.put\_line('-------------------------------------------------------------------------------------------------------------------------------------------------------------');   dbms\_output.put\_line('교육생번호|  '  ||vstudentnum||  '   출석일수|  ' ||출석일수||   '   정상일수|  '  ||정상일수||   '   지각일수|  ' ||지각일수  );              dbms\_output.put\_line('지각조퇴   |  '    ||지각조퇴||    '   외출일수|  '  ||외출||    '     결석일수|  ' ||결석|| '    조퇴일수|  '  ||조퇴일수   );              dbms\_output.put\_line('병가일수   |  '  ||병가||    '   기타일수|  '  ||기타  );              dbms\_output.put\_line('--------------------------------------------------------------------------------------------------------------------------------------------------------------');          end loop;      close vcursor;  end; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-06 | 작성일 | 2022-05-19 |
| 요구사항명 | 상담일지 작성 및 조회 | 작성자 | 권민채 |
| DML | | | |
| --상담일지 조회   create or replace view counsel\_v   as   select      member.name as 교육생이름,      counsel\_topic.counsel\_topic as 상담주제,      counsel\_date as 상담날짜,      curriculum.name as 수강과정  from counsel      inner join counsel\_topic on counsel\_topic.seq = counsel\_topic\_seq      inner join student on student.seq = student\_seq      inner join member on member.seq = student.member\_seq      inner join teacher on teacher.seq = counsel.teacher\_seq    inner join open\_curs on open\_curs.seq = student.open\_curs\_seq    inner join curriculum on curriculum.seq = open\_curs.curs\_seq    order by counsel\_date;     -- 특정 교육생의 상담현황 조회   create or replace procedure counsel\_member\_p(    cname varchar2:='',    )   is    cmember member.name%type;    ctopic counsel\_topic.counsel\_topic%type;    cdate counsel.counsel\_date%type;    cuname curriculum.name%type;    cursor vcursor is   select    member.name as 교육생이름,      counsel\_topic.counsel\_topic as 상담주제,      counsel\_date as 상담날짜,      curriculum.name as 수강과정   from counsel    inner join counsel\_topic on counsel\_topic.seq = counsel\_topic\_seq      inner join student on student.seq = student\_seq      inner join member on member.seq = student.member\_seq      inner join teacher on teacher.seq = counsel.teacher\_seq    inner join open\_curs on open\_curs.seq = student.open\_curs\_seq    inner join curriculum on curriculum.seq = open\_curs.curs\_seq    where member.name = cname order by counsel\_date;   begin    open vcursor;    loop  fetch vcursor into cmember, ctopic, cdate, cuname;  exit when vcursor%notfound;  dbms\_output.put\_line('교육생이름: ' || cmember || ' | 상담주제: ' || ctopic || ' | 상담날짜: ' || cdate || ' | 상담주제: ' || cuname);    end loop    close vcursor;  end;  -- 특정 과정별 상담현황 조회   create or replace procedure counsel\_curriculum\_p(    ccurri varchar2:='',    )   is    cmember member.name%type;    ctopic counsel\_topic.counsel\_topic%type;    cdate counsel.counsel\_date%type;    cuname curriculum.name%type;    cursor vcursor is   select    member.name as 교육생이름,      counsel\_topic.counsel\_topic as 상담주제,      counsel\_date as 상담날짜,      curriculum.name as 수강과정   from counsel    inner join counsel\_topic on counsel\_topic.seq = counsel\_topic\_seq      inner join student on student.seq = student\_seq      inner join member on member.seq = student.member\_seq      inner join teacher on teacher.seq = counsel.teacher\_seq    inner join open\_curs on open\_curs.seq = student.open\_curs\_seq    inner join curriculum on curriculum.seq = open\_curs.curs\_seq    where curriculum.name = ccurri order by counsel\_date;   begin    open vcursor;    loop  fetch vcursor into cmember, ctopic, cdate, cuname;  exit when vcursor%notfound;  dbms\_output.put\_line('교육생이름: ' || cmember || ' | 상담주제: ' || ctopic || ' | 상담날짜: ' || cdate || ' | 개설과정: ' || cuname);    end loop    close vcursor;  end;  --상담일지 등록  create or replace procedure insert\_counsel\_p(  cseq in varchar2,  ctopic in varchar2,  cdate in date  )  is  begin  insert into counsel values((select max(seq)+1 from counsel), cseq, ctopic, cdate);  end;  --상담일지 수정  create or replace procedure modify\_counsel\_p(  cseq in varchar2,  ctopic in varchar2,  cdate in date  )  is  begin  update counsel  set  student\_seq = cseq  counsel\_topic\_seq = ctopic  counsel\_date = cdate;  end;  --상담일지 삭제  create or replace procedure delete\_counsel\_p(  ccseq in number  )  is  begin  delete from counsel values where seq = ccseq;  end; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C07 | 작성일 | 2022-05-19 |
| 요구사항명 | 과제 등록 및 제출 현황 조회 | 작성자 | 주혜원 |
| DML | | | |
| **--과제 등록하기**  create or replace view task\_list\_V(인서트확인용 뷰)  as  select\*from task\_list;  create or replace procedure add\_assgin\_P(       psubject in  varchar2,       passgin  in varchar2,       deadline  in date  )  is  begin      insert into task\_list(seq,open\_curs\_seq,open\_subject\_seq,task\_description,summit\_limit) values      ((select max(seq) + 1 from task\_list),12, psubject, passgin,deadline);  --테이블에 원하는 값 넣기        exception      when others then      dbms\_output.put\_line('---------------------------------------------------------------------------------------');      dbms\_output.put\_line('              잘못 입력하였습니다.');      dbms\_output.put\_line('---------------------------------------------------------------------------------------');  end;    begin      add\_assgin\_P(1, '자바 열심히해라', '2022-01-01');  --원하는 과목명,과제,제출기한 입력하기  end; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | D01 | 작성일 | 2022-05-19 |
| 요구사항명 | 교육생 계정 | 작성자 | 주혜원 |
| DML | | | |
| **-로그인 된 교육생 계정 출력(기본정보)**  **--교육생 계정으로 로그인 후 본인의 개인정보와 본인이 수강한 과정명, 과정기간(시작 년월일, 끝 년월일), 강의실 출력**  create or replace procedure student\_info\_P(        ppw in member.ssn%type  )  is      vstudentNum student.seq%type;      vstudentName member.name%type;      vtel member.tel%type;      vcursName curriculum.name%type;      vcurTerm open\_curs.current\_term%type;      vcursStart open\_curs.begin\_date%type;      vcursFinish open\_curs.end\_date%type;      vlectureRoom lectureroom.room\_name%type;      cursor vcursor is select                             s.seq as "교육생번호",                             m.name as "이름",                             m.tel as "전화번호",                             c.name as"과정이름",                             o.current\_term as "과정기간",                             o.begin\_date  as "과정시작일",                             o.end\_date as "과정종료일",                             l.room\_name as "강의실"                  from student  s                   inner join member m  on m.seq = s.member\_seq                   inner join open\_curs   o on o.seq = s.open\_curs\_seq                   inner join curriculum c  on c.seq = o.curs\_seq                   inner join lectureroom l   on l.seq = o.room\_seq                   where m.ssn = (select ssn from member where ssn = ppw);      begin      open vcursor;          loop              fetch vcursor into  vstudentNum,  vstudentName, vtel, vcursName,  vcurTerm,  vcursStart, vcursFinish,vlectureRoom;              exit when vcursor%notfound;              dbms\_output.put\_line('-------------------------------------------------------------------------------------------------------------------------------------------------------------');              dbms\_output.put\_line('교육생번호 :  '  || vstudentNum||   '이름 :  '  || vstudentName ||   '전화번호 :  ' ||vtel||   '과정이름:  '  || vcursName||  '강의실:  ' ||vlectureRoom );              dbms\_output.put\_line('과정기간  :  '  ||vcurTerm||    '과정시작일 :  '  ||vcursStart||        '과정종료일   : ' ||vcursFinish );              dbms\_output.put\_line('--------------------------------------------------------------------------------------------------------------------------------------------------------------');          end loop;      close vcursor;      exception --3 아이디와 비밀번호가 일치하지 않다면 예외처리      when others then      dbms\_output.put\_line('             일치하는 정보가 없습니다. 비밀번호를 다시 확인하세요!');      dbms\_output.put\_line('');      dbms\_output.put\_line('--------------------------------------------------------------------------------------------------------------------------------------------------------------');  end;  begin       student\_info\_P('4256385');  end; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 명 | D-02 | 작성일 | 2022-05-19 |
| System | 성적 조회 | 작성자 | 주혜원 |
| Query | | | |
| **1.제목출력(D-01) 쿼리와 같음**  **2.학생번호,과목번호, 과목명, 과목기간, 교재명, 교사명, 과목별 배점 정보(출결,필기,실기 배점), 과목별 성적정보(출결,필기,실기 점수)**  select\*from  student\_score\_V--뷰  create or replace procedure student\_score\_P(        ppw in member.ssn%type  )  is      vstudentNum  student.seq%type;      vsubjectNum   subject.seq%type;      vjumin             member.ssn%type;      vsubjectName   subject.subject\_name%type;      vsubjectTerm    subject.seq%type;      vbookName     book.subject%type;      vteacherName   teacher.name%type;      vattendPoint   point.attend\_point%type;      vwrittenPoint   point.written\_point%type;      vskillPoint   point.skill\_point%type;      vattendScore    subject\_score.written\_score%type;      vwrittenScore   subject\_score.skill\_score%type;      vskillScore       subject\_score.attendance\_score%type;      cursor vcursor is (select\*from  student\_score\_V  where 주민번호 = (select ssn from member where ssn =4256385 ));      begin      open vcursor;          loop              fetch vcursor into  vstudentNum,  vsubjectNum ,  vjumin, vsubjectName,  vsubjectTerm ,  vbookName ,  vteacherName ,   vattendPoint , vwrittenPoint  ,vskillPoint ,  vattendScore,  vwrittenScore ,vskillScore;              exit when vcursor%notfound;              dbms\_output.put\_line('-------------------------------------------------------------------------------------------------------------------------------------------------------------');              dbms\_output.put\_line('교육생번호  : '  || vstudentNum ||    ' 과목 이름 :  '  || vsubjectName ||   '  과목기간 :  ' ||vsubjectTerm||   '  교재이름:  '  || vbookName ||  '  교사명:  ' || vteacherName );              dbms\_output.put\_line('출석배점  :  '  || vattendPoint||    '필기배점 :  '  ||vwrittenPoint||        '실기배점   : ' ||vskillPoint );              dbms\_output.put\_line('출석점수  :  '  ||  vattendScore||    '필기점수 :  '  || vwrittenScore||      '실기점수   : ' ||vskillScore );              dbms\_output.put\_line('--------------------------------------------------------------------------------------------------------------------------------------------------------------');          end loop;      close vcursor;      exception --3 아이디와 비밀번호가 일치하지 않다면 예외처리      when others then      dbms\_output.put\_line('             일치하는 정보가 없습니다. 비밀번호를 다시 확인하세요!');      dbms\_output.put\_line('');      dbms\_output.put\_line('--------------------------------------------------------------------------------------------------------------------------------------------------------------');  end;  begin       student\_score\_P('4256385');  end;    **--필기시험 목록 조회**  create or replace procedure written\_test\_P( ----원하는 과목명 입력      psubject  in written\_test.open\_subject\_seq%type  )  is      vsubjectNum    written\_test.open\_subject\_seq%type; --과목번호      vsubjectName   subject.subject\_name%type;  --과목이름      vsubjectTerm    subject.seq%type; --과목기간      vwrittenNum    written\_test.seq%type; --필기시험번호      vwrittenName   written\_test.written\_name%type;  --필기시험이름      vwrittenQuestion   written\_test\_question.written\_test\_question%type;   --필기시험 문제        cursor vcursor is      select      wt.open\_subject\_seq as "과목번호",      s.subject\_name as "과목이름",      os.end\_date - os.begin\_date as "과목기간",      wt.seq as "필기시험번호",      wt.written\_name as "필기시험 이름",      q.written\_test\_question as "필기시험문제"          from written\_test wt           inner join written\_test\_question q on wt.seq = q.written\_test\_seq           inner join open\_subject os   on os.seq = wt.open\_subject\_seq           inner join subject s  on  s.seq = os.subject\_seq           where  wt.open\_subject\_seq= psubject;          begin---------2      open vcursor;          LOOP              fetch vcursor into   vsubjectNum,  vsubjectName , vsubjectTerm,  vwrittenNum, vwrittenName ,  vwrittenQuestion  ;              exit when vcursor%notfound;                 dbms\_output.put\_line('-------------------------------------------------------------------------------------------------------------------------------------------------------------');              dbms\_output.put\_line('과목번호 : '  || vsubjectNum ||    ' 과목 이름 :  '  || vsubjectName ||   '  과목기간(일) :  ' ||vsubjectTerm );              dbms\_output.put\_line('필기시험번호  :  '  ||  vwrittenNum||    '필기시험 이름 :  '  ||vwrittenName );              dbms\_output.put\_line('필기시험문제 :  '  ||  vwrittenQuestion  );              dbms\_output.put\_line('--------------------------------------------------------------------------------------------------------------------------------------------------------------');          end loop;      close vcursor;        exception ----------3      when others then      dbms\_output.put\_line('              권한이 없습니다. 로그인 정보를 다시 확인하세요');      dbms\_output.put\_line('');      dbms\_output.put\_line('          —--------------------------------------------------------------------');      end; ------------------4    begin      written\_test\_P(57);  --필기시험 과목 번호 입력  end;    **-- 실기시험 목록 조회**  create or replace procedure skill\_test\_P( ----원하는 과목명 입력      psubject  in skill\_test.open\_subject\_seq%type  )  is      vsubjectNum    skill\_test.open\_subject\_seq%type; --과목번호      vsubjectName   subject.subject\_name%type;  --과목이름      vsubjectTerm    subject.seq%type; --과목기간      vskillNum         skill\_test.seq%type; --실기시험번호      vskillName        skill\_test.skill\_name%type;  --실기시험이름      vskillQuestion   skill\_test\_question.skill\_test\_question%type;   --실기시험 문제          cursor vcursor is    select      st.open\_subject\_seq as "과목번호",      s.subject\_name as "과목이름",      os.end\_date - os.begin\_date as "과목기간",      st.seq as "실기시험번호",      st.skill\_name as "실기시험 이름",      q.skill\_test\_question as "실기시험문제"  from skill\_test st          inner join skill\_test\_question q   on st.seq = q.skill\_test\_seq          inner join open\_subject os       on os.seq = st.open\_subject\_seq          inner join subject s     on  s.seq = os.subject\_seq           where  st.open\_subject\_seq= psubject;      begin---------2      open vcursor;          LOOP              fetch vcursor into   vsubjectNum,  vsubjectName , vsubjectTerm,  vskillNum, vskillName ,  vskillQuestion  ;              exit when vcursor%notfound;                 dbms\_output.put\_line('-------------------------------------------------------------------------------------------------------------------------------------------------------------');              dbms\_output.put\_line('과목번호 : '  || vsubjectNum ||    ' 과목 이름 :  '  || vsubjectName ||   '  과목기간(일) :  ' ||vsubjectTerm );              dbms\_output.put\_line('필기시험번호  :  '  ||  vskillNum||    '필기시험 이름 :  '  ||vskillName );              dbms\_output.put\_line('필기시험문제 :  '  ||  vskillQuestion  );              dbms\_output.put\_line('--------------------------------------------------------------------------------------------------------------------------------------------------------------');          end loop;      close vcursor;      exception ----------3      when others then      dbms\_output.put\_line('              권한이 없습니다. 로그인 정보를 다시 확인하세요');      dbms\_output.put\_line('');      dbms\_output.put\_line('        —-----------------------------------------------------------------------------');  end;  begin      skill\_test\_P(57);  --실기시험 과목 번호 입력  end; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 명 | D-03 | 작성일 | 2022-05-19 |
| System | 출결 관리 및 출결 조회 | 작성자 | 주혜원 |
| Query | | | |
| **--출결사항 조회하기**  create or replace procedure attendance\_check\_P(          pseq in number,   --학생번호        popenCurs  in number,  --특정개설과정번호        pdateStart in Date, --과정시작날짜        pdateEnd in Date --끝날짜  )  is      vstudentNum student.seq%type; --학생번호      출석일수  student.seq%type;  --출석일수      정상일수   student.seq%type; --정상일수      지각일수   student.seq%type;  --지각일수      조퇴일수     student.seq%type; --조퇴일수      지각조퇴     student.seq%type;  --지각 및 조퇴일수      외출     student.seq%type;   --외출일수      결석     student.seq%type;  --결석일수       병가      student.seq%type;  --병가일수       기타      student.seq%type;  --기타일수      cursor vcursor is      select      "학생번호",      count(decode("출석", '출석', 1)) as "출석일수",      count(decode("지각조퇴", '출석..', 1)) as "정상일수",      count(decode("지각조퇴", '출석지각.', 1)) as "지각일수",      count(decode("지각조퇴", '출석.조퇴', 1)) as "조퇴일수",      count(decode("지각조퇴", '출석지각조퇴', 1)) as "지각 및 조퇴일수",      count(decode("지각조퇴", '외출', 1)) as "외출일수",      count(decode("지각조퇴", '결석', 1)) as "결석일수",      count(decode("출석", '병가', 1)) as "병가일수",      count(decode("출석", '기타', 1)) as "기타일수"  from      (select i.\*,      출석||지각||조퇴 as "지각조퇴"      from attendance\_info i where "학생번호" in (select seq from student where open\_curs\_seq =  popenCurs )              and "날짜" between  pdateStart  and  pdateEnd)              where "학생번호"=pseq              group by "학생번호";        begin      open vcursor;          loop              fetch vcursor into   vstudentNum,  출석일수, 정상일수,  지각일수,  조퇴일수, 지각조퇴 ,외출,결석,병가,기타;              exit when vcursor%notfound;              dbms\_output.put\_line('-------------------------------------------------------------------------------------------------------------------------------------------------------------');   dbms\_output.put\_line('교육생번호|  '  ||vstudentNum||  '   출석일수|  ' ||출석일수||   '   정상일수|  '  ||정상일수||   '   지각일수|  ' ||지각일수  );              dbms\_output.put\_line('지각조퇴   |  '    ||지각조퇴||    '   외출일수|  '  ||외출||    '     결석일수|  ' ||결석|| '    조퇴일수|  '  ||조퇴일수   );              dbms\_output.put\_line('병가일수   |  '  ||병가||    '   기타일수|  '  ||기타  );              dbms\_output.put\_line('--------------------------------------------------------------------------------------------------------------------------------------------------------------');          end loop;      close vcursor;      exception --3 아이디와 비밀번호가 일치하지 않다면 예외처리      when others then      dbms\_output.put\_line('             일치하는 정보가 없습니다. 비밀번호를 다시 확인하세요!');      dbms\_output.put\_line('');      dbms\_output.put\_line('--------------------------------------------------------------------------------------------------------------------------------------------------------------');  end;--실행  begin    attendance\_check\_P(11, 11, '2022-02-02','2022-03-03');  end; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 명 | D-04 | 작성일 | 2022-05-19 |
| System | 상담일지조회 | 작성자 | 주혜원 |
| Query | | | |
| **–상담일지조회**  **–학생이름,상담날짜,상담주제,교사명**  create or replace procedure counsel\_check\_P(        ppw in member.ssn%type  )  is      vstudentName member.name%type;  --학생이름      vcounselDate  counsel.counsel\_date%type;    --상담날짜      vcounselTopic counsel\_topic.counsel\_topic%type;   --상담 주제      vteacherName teacher.name%type;  --교사명      cursor vcursor is      select          m.name as 이름,          c.counsel\_date as 상담날짜,          t.counsel\_topic as 상담주제,          t.name  as 교사명     from counsel c          inner join counsel\_topic t  on t.seq = c.counsel\_topic\_seq          inner join student s  on s.seq = c.student\_seq          inner join member m   on m.seq = s.member\_seq          inner join teacher t  on t.seq = c.teacher\_seq          where m.ssn = (select ssn from member where ssn = ppw);      begin      open vcursor;          loop              fetch vcursor into   vstudentName,   vcounselDate, vcounselTopic , vteacherName;              exit when vcursor%notfound;              dbms\_output.put\_line('-------------------------------------------------------------------------------------------------------------------------------------------------------------');              dbms\_output.put\_line('교육생이름 :  '  ||  vstudentName||   '  상담날짜 :  '  || vcounselDate ||   '  상담주제 :  ' ||vcounselTopic||   '  교사명:  '  || vteacherName );              dbms\_output.put\_line('--------------------------------------------------------------------------------------------------------------------------------------------------------------');          end loop;      close vcursor;      exception --3 아이디와 비밀번호가 일치하지 않다면 예외처리      when others then      dbms\_output.put\_line('             일치하는 정보가 없습니다. 비밀번호를 다시 확인하세요!');      dbms\_output.put\_line('');      dbms\_output.put\_line('--------------------------------------------------------------------------------------------------------------------------------------------------------------');  end;  begin      counsel\_check\_P(2660833);  end; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | D-05 | 작성일 | 2022-05-19 |
| 요구사항명 | 과제 조회 및 제출 | 작성자 | 김성태 |
| DML | | | |
| --자신이 제출한 과제 목록 조회  create or replace procedure summit\_P(  p\_member\_ssn in number,  p\_result out sys\_refcursor  )  is  begin  open p\_result  for  select  s.seq,  t.task\_description as 과제설명,  m.name as 제출자,  s.summit\_date as 제출날짜,  t.summit\_limit as 제출기한  from summit s  inner join task\_list t  on s.task\_seq = t.seq  inner join student stu  on s.student\_seq = stu.seq  inner join member m  on stu.member\_seq = m.seq  where m.ssn = p\_member\_ssn;  end summit\_P; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | D-06 | 작성일 | 2022-05-20 |
| 요구사항명 | 주로젝트 제목 검색기능 | 작성자 | 임종운 |
| DML | | | |
| CREATE OR REPLACE TYPE study\_obj AS OBJECT(      topic varchar2(50),      study\_name varchar2(50),      study\_description varchar2(150),      name varchar(20),      email varchar2(50),      tel varchar2(15),      begin\_date date,      end\_date date,      completement varchar2(5)  );  /  CREATE OR REPLACE TYPE study\_tables AS TABLE OF study\_obj;  /  CREATE OR REPLACE FUNCTION study\_list\_F  (      input\_topic\_or\_description varchar2  ) RETURN study\_tables  IS      v\_rtn study\_tables;  BEGIN      SELECT          study\_obj(rs.topic, rs.study\_name, rs.study\_description, rs.name, rs.email, rs.tel, rs.begin\_date, rs.end\_date, rs.completement)          BULK COLLECT INTO v\_rtn      FROM (SELECT          st.topic, sr.study\_name, sr.study\_description, m.name, sr.email, m.tel, sr.begin\_date, sr.end\_date, sr.completement      FROM study\_recruit sr          INNER JOIN study\_topic st ON sr.study\_topic\_seq = st.seq          INNER JOIN student s ON s.seq = sr.student\_seq          INNER JOIN member m ON s.member\_seq = m.seq      WHERE          sr.study\_name LIKE '%'||input\_topic\_or\_description||'%') rs;        RETURN v\_rtn;    END study\_list\_F;  /  DROP TYPE study\_tables;  DROP TYPE study\_obj;  DROP FUNCTION study\_list\_F;  --스터디 제목으로 검색 -- 공백입력시 전체 스터디 모집내역 출력됨.  SELECT \* FROM TABLE(study\_list\_F('웹')); | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | D-07 | 작성일 | 2022-05-19 |
| 요구사항명 | 취업 공고 조회 | 작성자 | 권민채 |
| DML | | | |
| --전체 취업공고 조회  create or replace view job\_list\_v  as(  select  city.city as "지역",  job\_announce.company as "회사명",  job\_field.field\_name as "분야",  contract\_type.contract\_type as "직위",  job\_announce.people as "모집인원",  job\_announce.begin\_date as "모집 시작일",  job\_announce.end\_date as "모집 종료일"  from job\_announce  inner join city on job\_announce.city\_seq = city.seq  inner join job\_field on job\_announce.field\_seq= job\_field.seq  inner join contract\_type on job\_announce.contract\_seq = contract\_type.seq);    --기간별 취업공고 조회  create or replace procedure job\_period\_p (      start\_period varchar2 := '',      end\_period varchar2 := ''  )  is    acompany job\_announce.company%type;          ccity city.city%type;          ffield\_name job\_field.field\_name%type;          ctcontract\_type contract\_type.contract\_type%type;          apeople job\_announce.people%type;          abegin\_date job\_announce.begin\_date%type;          aend\_date job\_announce.end\_date%type;          cursor vcursor is  select          a.company as 회사명,          c.city as 지역,          f.field\_name as 분야,          ct.contract\_type as 계약형태,          a.people as 모집인원,          a.begin\_date as 모집시작날짜,          a.end\_date as 모집종료날짜  from job\_announce a          inner join city c on c.seq = a.city\_seq          inner join job\_field f on f.seq = a.field\_seq          inner join contract\_type ct on ct.seq = a.contract\_seq           where a.end\_date between start\_period and end\_period order by 모집시작날짜;  begin          open vcursor;              loop                  fetch vcursor into acompany, ccity, ffield\_name, ctcontract\_type, apeople, abegin\_date, aend\_date;              exit when vcursor%notfound;              dbms\_output.put\_line('회사명: ' || acompany || ' | 지역: ' || ccity || ' | 분야: ' || ffield\_name || ' | 계약형태: ' || ctcontract\_type || ' | 모집인원: ' || apeople || '명' || ' | 모집시작날짜: ' ||                                                          abegin\_date || ' | 모집종료날짜: ' || aend\_date);              end loop;          close vcursor;  end;  --지역별 취업공고 조회  create or replace procedure city\_job\_check\_P (      pcity varchar2 := ''  )  is          acompany job\_announce.company%type;          ccity city.city%type;          ffield\_name job\_field.field\_name%type;          ctcontract\_type contract\_type.contract\_type%type;          apeople job\_announce.people%type;          abegin\_date job\_announce.begin\_date%type;          aend\_date job\_announce.end\_date%type;          cursor vcursor is  select          a.company as 회사명,          c.city as 지역,          f.field\_name as 분야,          ct.contract\_type as 계약형태,          a.people as 모집인원,          a.begin\_date as 모집시작날짜,          a.end\_date as 모집종료날짜  from job\_announce a          inner join city c on c.seq = a.city\_seq          inner join job\_field f on f.seq = a.field\_seq          inner join contract\_type ct on ct.seq = a.contract\_seq           where c.city = pcity order by 모집시작날짜;    begin          open vcursor;              loop                  fetch vcursor into acompany, ccity, ffield\_name, ctcontract\_type, apeople, abegin\_date, aend\_date;              exit when vcursor%notfound;                    dbms\_output.put\_line('회사명: ' || acompany || ' | 지역: ' || ccity || ' | 분야: ' || ffield\_name || ' | 계약형태: ' || ctcontract\_type || ' | 모집인원: ' || apeople || '명' || ' | 모집시작날짜: ' ||                                                          abegin\_date || ' | 모집종료날짜: ' || aend\_date);                end loop;          close vcursor;  end;  --분야별 취업공고 조회  create or replace procedure field\_job\_check\_P (      pjob\_field varchar2 := ''  )  is          acompany job\_announce.company%type;          ccity city.city%type;          ffield\_name job\_field.field\_name%type;          ctcontract\_type contract\_type.contract\_type%type;          apeople job\_announce.people%type;          abegin\_date job\_announce.begin\_date%type;          aend\_date job\_announce.end\_date%type;          cursor vcursor is  select          a.company as 회사명,          c.city as 지역,          f.field\_name as 분야,          ct.contract\_type as 계약형태,          a.people as 모집인원,          a.begin\_date as 모집시작날짜,          a.end\_date as 모집종료날짜  from job\_announce a          inner join city c on c.seq = a.city\_seq          inner join job\_field f on f.seq = a.field\_seq          inner join contract\_type ct on ct.seq = a.contract\_seq           where f.field\_name = pjob\_field order by 모집시작날짜;  begin          open vcursor;              loop                  fetch vcursor into acompany, ccity, ffield\_name, ctcontract\_type, apeople, abegin\_date, aend\_date;              exit when vcursor%notfound;                  dbms\_output.put\_line('회사명: ' || acompany || ' | 지역: ' || ccity || ' | 분야: ' || ffield\_name || ' | 계약형태: ' || ctcontract\_type || ' | 모집인원: ' || apeople || '명' || ' | 모집시작날짜: ' ||                                                          abegin\_date || ' | 모집종료날짜: ' || aend\_date);              end loop;          close vcursor;    end; | | | |