**DB Object 스크립트(PL-SQL)**

| 요구사항번호 | A-01 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 특정 교사 정보 | 작성자 | 김태완 |
| DML | | | |
| create or replace procedure procTeacherInfo (  pname in varchar2    )  is  vtName tblTeacher.tname%type;  vsName tblSubject.subname%type;  vcirStart tblCurr.cirstart%type;  vcirEnd tblCurr.cirend%type;  vProgname tblProglist.progname%type;  vNowstart tblNow.nowstart%type;  vNowend tblNow.nowend%type;  vbTitle tblBook.btitle%type;  vclassNum tblClass.classnum%type;  vStat varchar2(15);    cursor vcursor is    select  t.tname as "교사명",  s.subname as "과목명",  c.cirstart as "과목시작일",  c.cirend as "과목종료일",  p.progname as "과정명",  n.nowstart as "과정시작일",  n.nowend as "과정종료일",  b.btitle as "교재명",  cl.classnum as "강의실명",  case  when c.cirstart < sysdate and sysdate > c.cirend then '강의종료'  when c.cirstart > sysdate then '강의예정'  when c.cirstart <= sysdate and sysdate <= c.cirend then '강의중'  end as "강의상태"  from tblbooklist l  inner join tblbook b  on l.booknum = b.booknum  inner join tblcurr c  on l.cirnum = c.cirnum  inner join tblsubject s  on c.subnum = s.subnum  inner join tblnow n  on c.prognum = n.prognum  inner join tblclass cl  on n.classnum = cl.classnum  inner join tblproglist p  on n.prognum = p.prognum  inner join tblteacher t  on t.tnum = p.tnum  where t.tname = pname;  begin  open vcursor;    loop  fetch vcursor into vtName, vsName, vcirStart, vcirEnd, vProgname, vNowstart, vNowend, vbTitle, vclassNum, vStat;  exit when vcursor%notfound;  dbms\_output.put\_line('-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  dbms\_output.put\_line('교사명: ' || vtName || '| 과목명: ' || vsName || '| 과목시작일: ' || vcirStart || '| 과목종료일: ' || vcirEnd || '| 과정명: ' || vProgname || '| 과정시작일: ' || vNowstart || '| 과정종료일: ' || vNowend || '| 교재명: ' || vbTitle || '| 강의실명: '|| vclassNum || '| 강의상태: ' || vStat);  dbms\_output.put\_line('-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  end loop;    close vcursor;    end; | | | |

| 요구사항번호 | A-02 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 특정 개설 과정 정보 | 작성자 | 김태완 |
| DML | | | |
| create or replace procedure procProgInfo (  pname in varchar2    )  is  vProgname tblProglist.progname%type;  vsName tblSubject.subname%type;  vWriteStat varchar2(15);  vPracStat char(1);  vScoreStat char(1);    cursor vcursor is    select  distinct  p.progname as "과정명",  s.subname as "과목명",  case  when wt.cirnum = cu.cirnum then 'Y'  else 'N'  end as "필기문제 등록 여부",  case  when pt.cirnum = cu.cirnum then 'Y'  else 'N'  end as "실기문제 등록 여부",  case  when sc.scnum is not null then 'Y'  else 'N'  end as "성적 등록 여부"  from tblnow n  inner join tblproglist p  on n.prognum = p.prognum  inner join tblcurr cu  on n.prognum = cu.prognum  inner join tblsubject s  on cu.subnum = s.subnum  left outer join tblwriteT wt  on cu.cirnum = wt.cirnum  left outer join tblpracT pt  on cu.cirnum = pt.cirnum  right outer join tblscore sc  on pt.pracTnum = sc.pracTnum  where p.progname = pname;  begin  open vcursor;    loop  fetch vcursor into vProgname, vsName, vWriteStat, vPracStat, vScoreStat;  exit when vcursor%notfound;  dbms\_output.put\_line('-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  dbms\_output.put\_line('과정명: ' || vProgname || ' | 과목명: ' || vsName || ' | 필기문제 등록 여부: ' || vWriteStat || ' | 실기문제 등록 여부: ' || vPracStat || ' | 성적 등록 여부: ' || vScoreStat);  dbms\_output.put\_line('-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  end loop;    close vcursor;    end; | | | |

| 요구사항번호 | A-03 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 특정 교육생 정보 | 작성자 | 김태완 |
| DML | | | |
| create or replace procedure procStuInfo (  pname in varchar2    )  is  vhname tblp.hname%type;  vProgname tblProglist.progname%type;  vSubStat varchar2(15);  vclassNum tblnow.classnum%type;  vcirStart tblCurr.cirstart%type;  vcirEnd tblCurr.cirend%type;  vprogStat varchar2(15);  vDate date;    cursor vcursor is    select  p.hname as "이름",  pl.progname as "과정명",  case  when n.nowstart < sysdate and n.nowend < sysdate then '수강했음'  when n.nowstart > sysdate and n.nowend > sysdate then '수강예정'  when n.nowstart < sysdate and n.nowend > sysdate then '수강중'  end as "과정상태",  n.classnum as "강의실",  n.nowstart as "과정시작일",  n.nowend as "과정종료일",  case  when st.compstat = 'Y' and st.fail is null then '수료'  when st.compstat = 'N' and st.fail is not null then '중도탈락'  when st.compstat is null and st.fail is null and n.nowstart < sysdate then '수강중'  when st.compstat is null and st.fail is null and n.nowstart > sysdate then '수강예정'  end as "수료 및 중도탈락",  case  when st.fail is not null then st.fail  when st.compstat = 'Y' then n.nowend  end as "수료(중도탈락) 날짜"  from tblp p  inner join tblstudent st  on p.hnum = st.hnum  inner join tblnow n  on st.prognum = n.prognum  inner join tblproglist pl  on n.prognum = pl.prognum  where p.hname = pname;  begin  open vcursor;  loop  fetch vcursor into vhname, vProgname, vSubStat, vclassNum, vcirStart, vcirEnd, vprogStat, vDate;  exit when vcursor%notfound;  dbms\_output.put\_line('-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  dbms\_output.put\_line('이름: ' || vhname || ' | 과정명: ' || vProgname || ' | 과정상태: ' || vSubStat || ' | 강의실: ' || vclassNum || ' | 과정시작일: ' || vcirStart || ' | 과정종료일: ' || vcirEnd || ' | 수료 및 중도탈락: ' || vprogStat || ' | 수료(중도탈락) 날짜: ' || vDate) ;  dbms\_output.put\_line('-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  end loop;    close vcursor;    end; | | | |

| 요구사항번호 | A-04 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 입력한 과목과 강의 가능 과목이 일치하는 교사 명단 | 작성자 | 김태완 |
| DML | | | |
| create or replace procedure procCanInfo (  pname in varchar2    )  is  vTname tblteacher.tname%type;  vSubName tblSubject.subname%type;    cursor vcursor is    select  distinct  t.tname, s.subname  from tblteacher t  inner join tblcan c  on t.tnum = c.tnum  right outer join tblsubject s  on c.subnum = s.subnum  right outer join tblcurr cu  on s.subnum = cu.subnum  right outer join tblnow n  on cu.prognum = n.prognum  where n.nowstart > sysdate and n.nowend > sysdate and s.subname = pname;  begin  open vcursor;  loop  fetch vcursor into vTname, vSubName;  exit when vcursor%notfound;  dbms\_output.put\_line('-----------------------------------------------------------------------------------');  dbms\_output.put\_line('이름: ' || vTname || ' | 과목명: ' || vSubName);  dbms\_output.put\_line('-----------------------------------------------------------------------------------');  end loop;    close vcursor;    end; | | | |

| 요구사항번호 | A-05 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 특정 개설 과정 정보 | 작성자 | 김태완 |
| DML | | | |
| create or replace procedure procProgDetail (  pname in varchar2    )  is  vSubName tblSubject.subname%type;  vcirStart tblCurr.cirstart%type;  vcirEnd tblCurr.cirend%type;  vbTitle tblBook.btitle%type;  vtName tblTeacher.tname%type;  vhName tblp.hname%type;  vSsn varchar2(15);  vTel varchar2(15);  vRegdate date;  vStat varchar2(15);    cursor vcursor is    select  s.subname as "과목명",  cu.cirstart as "과목시작일",  cu.cirend as "과목종료일",  b.btitle as "교재명",  t.tname as "교사명",  p.hname as "학생명",  substr(p.hssn, 8) as "주민번호뒷자리",  p.htel as "전화번호",  st.regdate as "등록일",  (case  when st.fail is not null then '중도탈락'  when st.compstat = 'Y' then '수료'  end) as "수료 여부"  from tblnow n  inner join tblproglist p  on n.prognum = p.prognum  inner join tblcurr cu  on n.prognum = cu.prognum  inner join tblsubject s  on cu.subnum = s.subnum  inner join tblbooklist bl  on cu.cirnum = bl.cirnum  inner join tblbook b  on bl.booknum = b.booknum  inner join tblteacher t  on p.tnum = t.tnum  right outer join tblstudent st  on n.prognum = st.prognum  right outer join tblP p  on st.hnum = p.hnum  where p.progname = pname and (st.compstat is not null or st.fail is not null)  order by p.prognum asc;  begin  open vcursor;  loop  fetch vcursor into vSubName, vcirStart, vcirEnd, vbTitle, vtName, vhName, vSsn, vTel, vRegdate, vStat;  exit when vcursor%notfound;  dbms\_output.put\_line('--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  dbms\_output.put\_line('과목명: ' || vSubName || ' | 과목시작일: ' || vcirStart || ' | 과목종료일: ' || vcirEnd || ' | 교재명: ' || vbTitle || ' | 교사명: ' || vtName || ' | 학생명: ' || vhName || ' | 주민번호: ' || vSsn || ' | 전화번호: ' || vTel || ' | 등록일: ' || vRegdate || ' | 수료여부: ' || vStat);  dbms\_output.put\_line('--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');    end loop;    close vcursor;    end; | | | |

| 요구사항번호 | A-06 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 과정 관리 | 작성자 | 김태완 |
| DML | | | |
| --과정 추가  create or replace procedure procProglistAdd(  pPrognum in tblproglist.prognum%type,  pProgname in tblproglist.progname%type,  pTnum in tblproglist.tnum%type  )  is  begin  insert into tblProglist(progNum,progName,tNum) values(pPrognum,pProgname,pTnum);  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end;  --과정 삭제  create or replace procedure procProglistDel(  pProgNum in tblProglist.progNum%type  )  is  begin  delete from tblProglist  where pProgNum = progNum;  commit;  end;  --과정 수정  create or replace procedure procProglistUpdate(  pPrognum in tblproglist.prognum%type,  pProgname in tblproglist.progname%type,  pTnum in tblproglist.tnum%type  )  is  begin  update tblProglist set progName = pProgname, tNum = pTnum where progNum = pPrognum;  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end; | | | |

| 요구사항번호 | A-07 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 과목 관리 | 작성자 | 김태완 |
| DML | | | |
| --과목 추가  create or replace procedure procSubAdd(  pSubNum in tblsubject.subnum%type,  pSubName in tblsubject.subname%type  )  is  begin  insert into tblSubject(subNum,subName) values(pSubNum, pSubName);  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end;  --과목 삭제  create or replace procedure procSubDel(  pSubNum in tblSubject.SubNum%type  )  is  begin  delete from tblSubject  where pSubNum = SubNum;  commit;  end;  --과목 수정  create or replace procedure procSubUpdate(  pSubNum in tblsubject.subnum%type,  pSubName in tblsubject.subname%type  )  is  begin  update tblSubject set subName = pSubname where subNum = pSubnum;  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end; | | | |

| 요구사항번호 | A-08 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 강의실 관리 | 작성자 | 김태완 |
| DML | | | |
| --강의실 추가  create or replace procedure procClassAdd(  pClassNum in tblclass.classnum%type,  pClassLimit in tblclass.classlimit%type  )  is  begin  insert into tblClass(ClassNum,ClassLimit) values(pClassNum, pClassLimit);  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end;  --강의실 삭제  create or replace procedure procClassDel(  pClassNum in tblClass.ClassNum%type  )  is  begin  delete from tblClass  where pClassNum = ClassNum;  commit;  end;  create or replace procedure procClassUpdate(  pClassNum in tblclass.classnum%type,  pClassLimit in tblclass.classlimit%type  )  is  begin  update tblClass set classlimit = pClassLimit where classNum = pClassnum;  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end; | | | |

| 요구사항번호 | A-09 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 교재 관리 | 작성자 | 김태완 |
| DML | | | |
| --교재 추가  create or replace procedure procBookAdd(  pBookNum in tblBook.bookNum%type,  pbTitle in tblBook.bTitle%type,  pbPub in tblBook.bPub%type,  pbAuth in tblBook.bAuth%type  )  is  begin  insert into tblBook(bookNum, bTitle, bPub, bAuth) values(pbookNum, pbTitle, pbPub, pbAuth);  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end;  --교재 삭제  create or replace procedure procBookDel(  pbookNum in tblBook.bookNum%type  )  is  begin  delete from tblBook  where pbookNum = bookNum;  commit;  end;  --교재 수정  create or replace procedure procBookUpdate(  pBookNum in tblBook.bookNum%type,  pbTitle in tblBook.bTitle%type,  pbPub in tblBook.bPub%type,  pbAuth in tblBook.bAuth%type  )  is  begin  update tblBook set bTitle = pbTitle, bPub = pbPub, bAuth = pbAuth where bookNum = pBookNum;  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end; | | | |

| 요구사항번호 | A-10 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 개강 과목 관리 | 작성자 | 김태완 |
| DML | | | |
| ---개강 과목 추가  create or replace procedure procCurrAdd(  pSubNum in tblCurr.subNum%type,  pcirStart in tblCurr.cirStart%type,  pcirEnd in tblCurr.cirEnd%type,  pProgNum in tblCurr.progNum%type  )  is  begin  insert into tblCurr(cirNum, SubNum, cirStart, cirEnd, ProgNum) values((select max(cirNum) from tblCurr), pSubNum, pcirStart, pcirEnd, pProgNum);  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end;  --개강 과목 삭제  create or replace procedure procCurrDel(  pCirNum in tblCurr.cirNum%type  )  is  begin  delete from tblCurr  where pCirNum = cirNum;  commit;  end;  --개강 과목 수정  create or replace procedure procCurrUpdate(  pCirNum in tblCurr.cirNum%type,  pSubNum in tblCurr.subNum%type,  pcirStart in tblCurr.cirStart%type,  pcirEnd in tblCurr.cirEnd%type,  pProgNum in tblCurr.progNum%type  )  is  begin  update tblCurr set subNum = pSubNum, cirStart = pcirStart, cirEnd = pcirEnd, progNum = pProgNum where cirNum = pCirNum;  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end; | | | |

| 요구사항번호 | A-11 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 교사 관리 | 작성자 | 김태완 |
| DML | | | |
| --교사 추가  create or replace procedure procTeacherAdd(  vTName in tblTeacher.TName%type,  vTtel in tblTeacher.Ttel%type,  vTSsn in tblTeacher.TSsn%type,  vsubNum in tblCan.subNum%type,  vTNum in tblTeacher.TNum%type  )  is  begin  insert into tblTeacher (TNum, TName, TTel, TSsn) values((select max(TNum)+1 from tblTeacher), vTName, vTtel, vTSsn);  insert into tblCan (TNum, subNum) values((select max(canSeq)+1 from tblCan), vsubNum);  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end;  --교사 삭제  create or replace procedure procTeacherDel(  vTNum in tblTeacher.TNum%type  )  is  begin  delete from tblTeacher  where vTNum = TNum;  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end;  --교사 수정  create or replace procedure procTeacherUpdate(  vTName in tblTeacher.TName%type,  vTtel in tblTeacher.Ttel%type,  vTSsn in tblTeacher.TSsn%type,  vTNum in tblTeacher.TNum%type,  vsubNum in tblCan.subNum%type  )  is  begin  update tblTeacher set TName = vTName, Ttel = vTtel, TSsn = vTSsn where TNum = vTNum;  update tblCan set subNum = vsubNum where TNum = vTNum;  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end; | | | |

| 요구사항번호 | A-12 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 관리 | 작성자 | 김태완 |
| DML | | | |
| --교육생 추가  create or replace procedure procStudentAdd(  vRegDate in tblStudent.regdate%type,  vEval in tblStudent.eval%type,  vPrognum in tblStudent.prognum%type,  vhNum in tblStudent.hNum%type  )  is  begin  insert into tblStudent (stNum, regdate, compstat, fail, eval, progNum, hNum) values((select max(TNum)+1 from tblTeacher), vRegDate, null, null, vEval, vPrognum, vhNum);  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end;  --교육생 삭제  create or replace procedure procStudentDel(  vStNum in tblStudent.stNum%type  )  is  begin  delete from tblStudent  where vStNum = stNum;  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end;  --교육생 수정  create or replace procedure procStudentUpdate(  vStNum in tblStudent.stNum%type,  vRegDate in tblStudent.regdate%type,  vCompStat in tblStudent.compstat%type,  vFail in tblStudent.fail%type,  vEval in tblStudent.eval%type,  vPrognum in tblStudent.prognum%type,  vhNum in tblStudent.hNum%type  )  is  begin  update tblStudent set regdate = vRegDate, compstat = vCompStat, fail = vFail, eval = vEval, prognum = vPrognum, hNum = vhNum where stNum = vStNum;  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end; | | | |

| 요구사항번호 | A-13 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 과정 정보 입력 | 작성자 | 김태완 |
| DML | | | |
| --과정 정보 (과정명, 과정기간(시작 년월일, 끝 년월일), 강의실 정보) => 입력  create or replace procedure procProgInfoAdd(  vProgNum in tblNow.progNum%type,  vNowStart in tblNow.nowStart%type,  vNowEnd in tblNow.nowEnd%type,  vClassNum in tblNow.classNum%type,  vProgName in tblProglist.progName%type,  vTNum in tblproglist.TNum%type  )  is  begin  insert into tblNow (progNum, nowStart, nowEnd, classNum) values((select max(progNum)+1 from tblNow), vNowStart, vNowEnd, vClassNum);  insert into tblProglist (progNum, progName, TNum) values((select max(progNum) from tblNow), vProgName, vTNum);  commit;  dbms\_output.put\_line('INSERT SUCCESS');  end; | | | |

| 요구사항번호 | A-14 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 정보 입력 | 작성자 | 김태완 |
| DML | | | |
| -- 교육생 정보 입력 => (교육생 이름, 주민번호 뒷자리, 전화번호,등록일) 등록일은 자동입력  create or replace procedure procStudentInfoAdd(  vhName in tblP.hName%type,  vhSsn in tblP.hSsn%type,  vhTel in tblP.hTel%type,  vhCity in tblP.hCity%type,  vRegDate in tblStudent.regdate%type,  vEval in tblStudent.eval%type,  vPrognum in tblStudent.prognum%type  )  is  begin  insert into tblP (hNum, hName, hSsn, hTel, hCity) values((select max(hNum)+1 from tblP), vhName, vhSsn, vhTel, vhCity);  insert into tblStudent (stNum, regdate, compstat, fail, eval, progNum, hNum) values((select max(stNum) from tblStudent), sysdate, null, null, vEval, vPrognum, (select max(hNum) from tblP));  end; | | | |

| 요구사항번호 | A-15 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 특정 개설 과정을 선택하는 경우 모든 교육생의 출결을 조회 | 작성자 | 유희윤 |
| DML | | | |
| create or replace procedure procheck\_allattend(  pnum varchar2  )  is  vstnum tblstudent.stnum%type;  vattDate varchar2(10);  vattStat tblAttend.attStat%type;  cursor vcursor is select  a.stnum as "학생번호",  a.attdate as "날짜",  a.attstat as "근태상황"  from tblAttend a  inner join tblStudent s  on a.stNum = s.stnum  inner join tblProgList p  on s.progNum = p.progNum  where p.progNum = pnum  order by s.stNum;  begin  open vcursor;  loop  fetch vcursor into vstnum, vattDate, vattStat;  exit when vcursor%notfound;  dbms\_output.put\_line('학생번호: ' || vstnum || '날짜:' || vattDate || '근태상황: '|| vattStat);  end loop;  close vcursor;  end procheck\_allattend; | | | |

| 요구사항번호 | A-16 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 특정 교육생의 출결을 조회 | 작성자 | 유희윤 |
| DML | | | |
| create or replace procedure procheck\_m\_attend(  phnum varchar2  )  is  vhname tblp.hname%type;  vattdate tblattend.attdate%type;  vattstat tblattend.attstat%type;  cursor vcursor is select  e.hname as "이름",  a.attdate as "날짜",  a.attstat as "근태상황"  from tblAttend a  inner join tblStudent s  on a.stNum = s.stnum  inner join tblProgList p  on s.progNum = p.progNum  inner join tblP e  on s.hNum = e.hnum  where s.stnum = phnum  order by a.attdate;  begin  open vcursor;  loop  fetch vcursor into vhname, vattdate, vattstat;  exit when vcursor%notfound;    dbms\_output.put\_line('이름: ' || vhname || '날짜: '|| vattdate || '근태상황: ' || vattstat);  end loop;  close vcursor;  end procheck\_m\_attend; | | | |

| 요구사항번호 | A-17 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 출결 현황을 기간별(년) 조회 | 작성자 | 유희윤 |
| DML | | | |
| create or replace procedure proSearchYear\_Attend(  pyear varchar2  )  is  vstnum tblstudent.stnum%type;  vattDate varchar2(4);  vattStat tblAttend.attStat%type;    cursor vcursor is select  a.stnum as "학생번호",  to\_char(a.attDate ,'yyyy') as "년도",  a.attstat as "근태"  from tblAttend a  where to\_char(attdate, 'yyyy') = pyear;  begin    open vcursor;  loop  fetch vcursor into vstnum, vattDate, vattStat;  exit when vcursor%notfound;  dbms\_output.put\_line('학생번호: ' || vstnum || '년도:' || vattDate || '근태: '|| vattStat);  end loop;  close vcursor;    end proSearchYear\_Attend;  set serveroutput on;  begin  proSearchYear\_Attend('2023');  end; | | | |

| 요구사항번호 | A-18 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 출결 현황을 기간별(월) 조회 | 작성자 | 유희윤 |
| DML | | | |
| create or replace procedure proSearchMonth\_Attend(  pyear varchar2,  pmonth varchar2  )  is  vstnum tblAttend.stnum%type;  vattDate1 varchar2(4);  vattDate2 varchar2(2);  vattstat tblAttend.attstat%type;    cursor vcursor is select  stnum as "학생번호",  to\_char(attDate ,'yyyy') as "년도",  to\_char(attDate ,'mm') as "월",  attstat as "근태"  from tblAttend  where to\_char(attdate, 'yyyy') = pyear and  to\_char(attdate, 'mm') = pmonth;    begin  open vcursor;  loop  fetch vcursor into vstnum, vattDate1, vattDate2, vattstat;  exit when vcursor%notfound;    dbms\_output.put\_line('학생번호: ' || vstnum || '년도:' || vattDate1 || '월:' || vattDate2 || '근태: '|| vattStat);  end loop;  close vcursor;    end proSearchMonth\_Attend; | | | |

| 요구사항번호 | A-19 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 출결 현황을 기간별(일) 조회 | 작성자 | 유희윤 |
| DML | | | |
| create or replace procedure proSearchDay\_Attend(  pyear varchar2,  pmonth varchar2,  pdate varchar2  )  is    vstnum tblAttend.stnum%type;  vattDate1 varchar2(2);  vattDate2 varchar2(2);  vattDate3 varchar2(4);  vattStat tblAttend.attStat%type;    cursor vcursor is select  stnum as "학생번호",  to\_char(attDate ,'yyyy') as "년도",  to\_char(attDate ,'mm') as "월",  to\_char(attDate, 'dd') as "일",  attstat as "근태"  from tblAttend  where to\_char(attdate, 'yyyy') = pyear and  to\_char(attdate, 'mm') = pmonth and  to\_char(attdate, 'dd') = pdate;  begin    open vcursor;  loop  fetch vcursor into vstnum, vattDate3, vattDate2, vattdate1, vattstat;  exit when vcursor%notfound;  dbms\_output.put\_line('학생번호: ' || vstnum || '년도: ' || vattDate3 || '월: ' || vattDate2 || '일: ' || vattDate1 || '근태: ' || vattStat);  end loop;  close vcursor;    end proSearchDay\_Attend; | | | |

| 요구사항번호 | A-20 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 특정 과정의 출결 현황을 조회 | 작성자 | 유희윤 |
| DML | | | |
| create or replace procedure procheck\_pro\_attend(  pprognum varchar2  )  is  vhname tblp.hname%type;  vattdate tblattend.attdate%type;  vattstat tblattend.attstat%type;  cursor vcursor is select  h.hname as "이름",  a.attdate as "날짜",  a.attstat as "근태상황"  from tblAttend a  inner join tblStudent s  on a.stNum = s.stnum  inner join tblProgList p  on s.progNum = p.progNum  inner join tblp h  on s.hNum = h.hnum  where p.progNum = pprognum  order by a.attdate, s.stnum;  begin  open vcursor;  loop  fetch vcursor into vhname, vattdate, vattstat;  exit when vcursor%notfound;    dbms\_output.put\_line('이름: ' || vhname || '날짜: '|| vattdate || '근태상황: ' || vattstat);  end loop;  close vcursor;  end procheck\_pro\_attend; | | | |

| 요구사항번호 | A-21 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 특정 교육생의 출결 현황을 조회 | 작성자 | 유희윤 |
| DML | | | |
| create or replace procedure procheck\_m\_attend(  phnum varchar2  )  is  vhname tblp.hname%type;  vattdate tblattend.attdate%type;  vattstat tblattend.attstat%type;  cursor vcursor is select  e.hname as "이름",  a.attdate as "날짜",  a.attstat as "근태상황"  from tblAttend a  inner join tblStudent s  on a.stNum = s.stnum  inner join tblProgList p  on s.progNum = p.progNum  inner join tblP e  on s.hNum = e.hnum  where s.stnum = phnum  order by a.attdate;  begin  open vcursor;  loop  fetch vcursor into vhname, vattdate, vattstat;  exit when vcursor%notfound;    dbms\_output.put\_line('이름: ' || vhname || '날짜: '|| vattdate || '근태상황: ' || vattstat);  end loop;  close vcursor;  end procheck\_m\_attend; | | | |

| 요구사항번호 | A-22 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 개근상 | 작성자 | 유희윤 |
| DML | | | |
| create or replace procedure proAward\_Attend(  pprogNum varchar2  )  is    vphname tblP.hname%type;    cursor vcursor is select  p.hname    from tblAttend a  inner join tblStudent s  on a.stNum = s.stNum  inner join tblP p  on s.hnum = p.hnum  where s.progNum = pprognum    group by s.stNum, p.hname, s.ProgNum  having count(attStat)  = (select max(count(\*))  from tblattend z  inner join tblStudent x  on z.stNum = x.stNum  where x.prognum = s.progNum and z.attdate between      (select nowstart from tblNow where prognum = s.progNum) and    (select nowend from tblNow where prognum = s.progNum)  group by z.stNum);    begin  open vcursor;  loop  fetch vcursor into vphname;  exit when vcursor%notfound;    dbms\_output.put\_line('이름: ' || vphname);  end loop;  close vcursor;  end proAward\_Attend; | | | |

| 요구사항번호 | A-23 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 요구사항명 | 출근 시간에 따른 출결 현황 상태 분류 | 작성자 | 김수민 |
| DML | | | |
| create or replace trigger trgtime  after  insert or update  on tblTime  for each row  declare  vresult number;  vmessage varchar2(1000);  begin  select count(\*) into vresult from tblAttend where attDate = sysdate and stNum = :old.stNum;  if inserting then  if to\_char(sysdate,'hh24:mi:ss') between '06:00:00' and '09:00:00' then  vmessage := '출근하셨습니다.';  elsif to\_char(sysdate,'hh24:mi:ss') > '09:00:00' then  insert into tblAttend values(seqAttNum.nextVal, :new.stNum, sysdate, '지각');  vmessage := '지각하셨습니다.';  end if;  elsif updating then  if to\_char(sysdate,'hh24:mi:ss') < '18:00:00' then  if vresult = 0 then  insert into tblAttend values (seqAttNum.nextVal, :old.stNum, sysdate, '조퇴');  vmessage := '조퇴하셨습니다.';  elsif vresult > 0 then  update tblAttend set attStat = '조퇴';  vmessage := '조퇴하셨습니다.';  end if;  elsif to\_char(sysdate, 'hh24:mi:ss') between '18:00:00' and '22:00:00' then  if vresult = 0 then  insert into tblAttend values (seqAttNum.nextVal, :old.stNum, sysdate, '출결');  vmessage := '출결하셨습니다.';  end if;  end if;  end if;  dbms\_output.put\_line(vmessage);  end trgtime; | | | |

| 요구사항번호 | A-24 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 요구사항명 | 출결현황 입력시 교육생의 현재 수료 진행여부와 중도탈락자 구분 | 작성자 | 김수민 |
| DML | | | |
| create or replace trigger trgClassEndCome  before  insert  on tblAttend  for each row  declare  vNum number;  fNum number;  cursor vcursor is select stNum from (select progNum from tblNow where nowStart >= sysdate or nowEnd <= sysdate) a inner join tblStudent s on a.prognum = s.progNum;  cursor fcursor is select stNum from tblStudent where fail is not null;  begin  if :new.stNum <>0 then  open fcursor;  loop  fetch fcursor into fNum;  exit when fcursor%notfound;    if :new.stNum = fNum then  dbms\_output.put\_line('중도탈락되셨습니다.');  raise\_application\_error(-20000,'중도탈락되셨습니다.');  end if;  end loop;  close fcursor;    open vcursor;  loop  fetch vcursor into vNum;  exit when vcursor%notfound;    if :new.stNum = vNum then  dbms\_output.put\_line('수료과정 중이 아닙니다.');  raise\_application\_error(-20000,'수료과정 중이 아닙니다.');  end if;  end loop;  close vcursor;  end if;  end; | | | |

| 요구사항번호 | A-25 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 요구사항명 | 출결시 중도탈락자 구분 | 작성자 | 김수민 |
| DML | | | |
| create or replace procedure procdropOut(  pstNum number  )  is  vabsence number;  vout number;  begin  select count(\*) into vabsence from tblAttend where stNum = pstNum and attStat = '결석' group by attStat;  select count(\*) into vout from tblAttend where stNum = pstNum and attStat in ('조퇴','지각','병가');  if vabsence > 10 then  update tblStudent set fail = sysdate where stNum = pstNum;  elsif (vabsence + (vout\*0.5)) > 10 then  update tblStudent set fail = sysdate where stNum = pstNum;  end if;  exception  when others then  dbms\_output.put\_line('잘못된 접근입니다.');  end procdropOut; | | | |

| 요구사항번호 | B-01 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 강의 스케줄 조회 | 작성자 | 유희윤 |
| DML | | | |
| create or replace procedure proSchedule\_t(  ptnum varchar2  )  is  vstate varchar2(30);  vprogNum tblProglist.prognum%type;  vprogName tblProglist.progName%type;  vnow varchar2(30);  vclassNum tblNow.classNum%type;  vsubname tblsubject.subname%type;  vcir varchar2(30);  vbtitle tblBook.btitle%type;  vcount number;  vhname tblp.hname%type;  vhtel tblp.htel%type;  vregdate tblStudent.regdate%type;  vcomp varchar2(30);  vfail tblStudent.fail%type;    cursor vcursor is select  case  when n.nowStart > sysdate then '강의 예정'  when n.nowStart < sysdate and n.nowEnd > sysdate then '강의 중'  when n.nowEnd < sysdate then '강의 종료'  end as "강의 진행 상태",  p.progNum as "과정번호",  p.progName as "과정명",  ( n.nowStart|| ' ~ ' || n.nowend) as "과정기간",  n.classNum as "강의실",  s.subname as "과목명",  (c.cirStart || ' ~ ' || c.cirend) as "과목기간",  k.btitle as "교재명",  (select count(\*) from tblstudent where prognum = p.prognum and fail is null) as "교육생인원",  pp.hname as "교육생 이름",  pp.htel as "전화번호",  sd.regdate as "등록일",  case  when sd.compStat is null then '진행중'  when sd.compStat is not null then sd.compStat  end "수료여부",  sd.fail as "탈락날짜"  from tblTeacher t  inner join tblProgList p  on t.tnum = p.tnum  left outer join tblNow n  on p.progNum = n.progNum  left outer join tblCurr c  on n.prognum = c.prognum  inner join tblsubject s  on c.subnum = s.subnum  inner join tblBookList b  on c.cirnum = b.cirnum  inner join tblBook k  on b.booknum = k.booknum  inner join tblStudent sd  on n.prognum = sd.prognum  inner join tblp pp  on sd.hnum = pp.hnum  where t.tnum = 1;  begin  open vcursor;  loop  fetch vcursor into vstate, vprogNum, vprogName, vnow, vclassNum, vsubname, vcir, vbtitle, vcount, vhname, vhtel, vregdate, vcomp, vfail;  exit when vcursor%notfound;  dbms\_output.put\_line('강의진행상태: ' || vstate || '과정번호: ' || vprognum || '과정명: ' || vprogName || '과정기간: ' || vnow  || '강의실: ' || vclassNum || '과목명: ' || vsubname || '과목기간: ' || vcir || '교재명: ' || vbtitle || '교육생인원: ' || vcount  || '교육생이름: '|| vhname ||'전화번호: ' || vhtel ||'등록일: ' || vregdate ||'수료여부: '|| vcomp ||'탈락날짜: '|| vfail );    end loop;  close vcursor;  end proSchedule\_t; | | | |

| 요구사항번호 | B-02 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 특정 과목을 과목번호로 선택 시 출결 배점, 필기 배점, 실기 배점 출력 | 작성자 | 유희윤 |
| DML | | | |
| create or replace procedure protestout\_pro(  pcirnum varchar2  )  is  vcirnum tblCurr.cirnum%type;  vsubnum tblcurr.cirnum%type;  vattpoint tblpoint.attpoint%type;  vwritepoint tblpoint.writepoint%type;  vpracpoint tblpoint.pracpoint%type;  cursor vcursor is select  a.cirNum as 과정번호,  a.subNum as 과목번호,  p.attPoint as 출결배점,  p.writePoint as 필기배점,  p.pracPoint as 실기배점  from tblCurr a  inner join tblPoint p  on a.cirNum = p.cirNum  where a.cirEnd < sysdate and  p.cirNum = pcirnum;  begin  open vcursor;  loop  fetch vcursor into vcirnum, vsubnum, vattpoint, vwritepoint, vpracpoint;  exit when vcursor%notfound;  dbms\_output.put\_line('과정번호: '|| vcirnum || '과목번호: ' || vsubnum || '출결배점: ' || vattpoint ||'필기배점: ' || vwritepoint ||'실기배점: ' || vpracpoint);  end loop;  close vcursor;  end protestout\_pro; | | | |

| 요구사항번호 | B-03 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 필기시험날짜, 필기시험문제 입력 | 작성자 | 유희윤 |
| DML | | | |
| create or replace procedure protestin\_write(  pwritenum varchar2,  pwritet varchar2,  ppronum varchar2,  pdate varchar2,  pwritename varchar2  )  is  begin  insert into tblWrite values((select max(writeNum)+1 from tblwrite), pwritenum, pwritet);  insert into tblWriteT values((select max(writeTNum)+1 from tblwriteT), ppronum, to\_date(pdate, 'yyyy-mm-dd'), pwritename);    end protestin\_write; | | | |

| 요구사항번호 | B-04 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 실기시험날짜, 실기시험문제 입력 | 작성자 | 유희윤 |
| DML | | | |
| create or replace procedure protestin\_prac(  ppracnum varchar2,  ppract varchar2,  ppronum varchar2,  pdate varchar2,  ppracname varchar2  )  is  begin  insert into tblPrac values((select max(pracNum)+1 from tblPrac), ppracnum, ppract);  insert into tblPracT values((select max(pracTNum)+1 from tblPracT), ppronum, to\_date(pdate, 'yyyy-mm-dd'), ppracname);    end protestin\_prac;  drop procedure protestin\_prac; | | | |

| 요구사항번호 | B-05 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 요구사항명 | 자신이 강의를 마친 과목의 목록 중에서 특정 과목을 선택하고 해당 과목의 배점 정보를 출결, 필기, 실기 나누어 입력 | 작성자 | 김수민 |
| DML | | | |
| create or replace procedure procTeaTest(  pTNum number,  pSubNum number,  pAttPoint number,  pWritePoint number,  pPracPoint number  )  is  vresult number;  begin  select  count(\*)  into vresult  from (select  \*  from tblCurr  where cirEnd < sysdate) a  inner join tblPoint p  on a.cirNum = p.cirNum  inner join tblProgList l  on a.progNum = l.progNum  where l.TNum = pTNum and p.cirNum = pSubNum;  if vresult > 0 then  insert into tblPoint values((select max(pointseq)+1 from tblPoint), pAttPoint, pWritePoint, pPracPoint);  end; | | | |

| 요구사항번호 | B-06 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 요구사항명 | 관리자 계정 | 작성자 | 최영진 |
| DML | | | |
| 과제와 관련된 프로시저와 뷰.  (프로시저)  --1. 과제별 출력(교육생번호, 교육생이름, 과제명, 제출 여부) 출력  --procTask('과제명')  create or replace procedure procTask(  ptaskname in varchar2  )  is  vstnum tblStudent.stNum%type;  vhname tblP.hName%type;  vtaskname tblTask.taskName%type;  vmitstat tblTask.mitstat%type;  cursor vcursor is  select  s.stNum as "교육생번호",  p.hName as "교육생이름",  t.taskName as "과제명",  t.mitStat as "제출여부"  into  vstnum,  vhname,  vtaskname,  vmitstat  from tblP p  inner join tblStudent s  on p.hnum = s.hnum  inner join tblTask t  on s.stNum = t.stNum  where t.taskName = ptaskname;  begin    open vcursor;  loop  fetch vcursor into vstnum, vhname, vtaskname, vmitstat;  exit when vcursor%notfound;    if vmitstat = 'y' then  dbms\_output.put\_line('교육생번호: ' || vstnum || ' | 교육생이름: ' || vhname || ' | 과제명: ' || vtaskname || ' | 제출O');  elsif vmitstat = 'n' then  dbms\_output.put\_line('교육생번호: ' || vstnum || ' | 교육생이름: ' || vhname || ' | 과제명: ' || vtaskname || ' | 제출X');  end if;  end loop;  close vcursor;    end procTask;  /  --2. 교육생별 출력(교육생 번호, 교육생이름, 과정명, 과제번호, 과제명, 제출 여부) 출력  --procTask(교육생번호)  create or replace procedure procTaskSt (  pstNum in number  )  is  vstnum tblStudent.stNum%type;  vhname tblP.hname%type;  vprogname tblProgList.progName%type;  vtasknum tblTask.taskNum%type;  vtaskname tblTask.taskName%type;  vmitstat tblTask.mitStat%type;  cursor vcursor is  select  s.stNum as "교육생번호",  p.hName as "교육생이름",  pl.progName as "과정명",  t.taskNum as "과제번호",  t.taskName as "과제명",  t.mitStat as "제출여부"  into  vstnum,  vhname,  vprogname,  vtasknum,  vtaskname,  vmitstat  from tblP p  inner join tblStudent s  on p.hnum = s.hnum  inner join tblTask t  on s.stNum = t.stNum  inner join tblNow n  on s.progNum = n.progNum  inner join tblProgList pl  on n.progNum = pl.progNum  where s.stNum = pstNum;  begin  open vcursor;  loop  fetch vcursor into vstnum, vhname, vprogname, vtasknum, vtaskname, vmitstat;  exit when vcursor%notfound;    if vmitstat = 'y' then  dbms\_output.put\_line('교육생번호: ' || vstnum || ' | 교육생이름: ' || vhname || ' | 과정명: ' || vprogname || ' | 과제번호: ' || vtasknum || ' | 과제명: ' || vtaskname || ' | 제출O');  elsif vmitstat = 'n' then  dbms\_output.put\_line('교육생번호: ' || vstnum || ' | 교육생이름: ' || vhname || ' | 과정명: ' || vprogname || ' | 과제번호: ' || vtasknum || ' | 과제명: ' || vtaskname || ' | 제출X');  end if;  end loop;  close vcursor;  end procTaskSt;  /  --3. 과제 추가  --procTaskAdd('과제명', 제출여부(y/n), 교육생번호, vresult)  create or replace procedure procTaskAdd(  ptaskname in varchar2,  pmitstat in char,  pstnum in number,  presult out number  )  is  vtasknum number;  begin  insert into tblTask (taskNum, taskName, mitStat, stNum)  values ((select nvl(max(tasknum), 0) + 1 from tblTask), ptaskname, pmitstat, pstnum);    presult := 1; --성공  exception  when others then  presult := 0; --실패  end procTaskAdd;  /  --호출 procTaskAdd('과제명', 제출여부(y/n), 교육생번호, vresult)  declare  vresult number; --결과  begin  procTaskAdd('네이버영화 DB 분석', 'y', 1, vresult);    if vresult = 1 then  dbms\_output.put\_line('과제 추가 완료');  else  dbms\_output.put\_line('과제 추가 실패');  end if;  end;  /  --4. 과제 제출여부 수정  -- procTaskUpdate(과제번호, 제출여부(y/n))  create or replace procedure procTaskUpdate(  ptasknum in number,  pmitstat in char  )  is  begin  update tblTask  set mitStat = pmitstat  where tasknum = ptasknum;  dbms\_output.put\_line('과제 수정 성공');  commit;  exception  when others then  dbms\_output.put\_line('과제 수정 실패');  rollback;  end procTaskUpdate;  /  --5. 과제 삭제  --procTaskRemove(과제번호)  create or replace procedure procTaskRemove (  ptasknum in number  )  is  begin  delete from tblTask where taskNum = ptasknum;  dbms\_output.put\_line('과제 삭제 성공');  commit;  exception  when others then  dbms\_output.put\_line('과제 삭제 실패');  rollback;  end procTaskRemove;  /  (뷰)  --1. 전체 출력(교육생번호, 교육생이름, 과제명, 제출 여부)  create or replace view vwTaskList  as  select  s.stNum as "교육생번호",  p.hName as "교육생이름",  t.taskName as "과제명",  t.mitStat as "제출여부"  from tblP p  inner join tblStudent s  on p.hnum = s.hnum  inner join tblTask t  on s.stNum = t.stNum;  --2. 교육생 전용 출력(교육생 번호, 교육생이름, 과정명, 과제번호, 과제명, 제출 여부)  create or replace view vwTaskSt  as  select  s.stNum as "교육생번호",  p.hName as "교육생이름",  pl.progName as "과정명",  t.taskNum as "과제번호",  t.taskName as "과제명",  t.mitStat as "제출여부"  from tblP p  inner join tblStudent s  on p.hnum = s.hnum  inner join tblTask t  on s.stNum = t.stNum  inner join tblNow n  on s.progNum = n.progNum  inner join tblProgList pl  on n.progNum = pl.progNum  where s.stNum = <교육생번호>; | | | |

| 요구사항번호 | B-07 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 요구사항명 | 관리자 계정 | 작성자 | 최영진 |
| DML | | | |
| 시험과 관련된 프로시저.  --1. 과정별 필기/실기시험명 출력 (과정번호, 과정명, 필기시험명, 필기시험날짜, 실기시험명, 실기시험날짜)  --procTest(과정번호)  create or replace procedure procTest(  pprognum in number  )  is  vprognum tblProgList.progNum%type;  vprogname tblProgList.progName%type;  vcirnum tblCurr.cirNum%type;  vwritename tblWriteT.writeName%type;  vwritedate tblWriteT.writeDate%type;  vpracName tblPracT.pracName%type;  vpracdate tblPracT.pracDate%type;  cursor vcursor is  select  p.progNum as "과정번호",  p.progName as "과정명",  c.cirNum as "커리번호",  wt.writeName as "필기시험명",  wt.writeDate as "필기시험날짜",  pt.pracName as "실기시험명",  pt.pracDate as "실기시험날짜"  into  vprognum,  vprogname,  vcirnum,  vwritename,  vwritedate,  vpracname,  vpracdate  from tblProgList p  inner join tblNow n  on p.progNum = n.progNum  inner join tblCurr c  on n.progNum = c.progNum  inner join tblWriteT wt  on c.cirNum = wt.cirNum  inner join tblPracT pt  on c.cirNum = pt.cirNum  where p.progNum = pprognum;    begin  open vcursor;  loop  fetch vcursor into vprognum, vprogname, vcirnum, vwritename, vwritedate, vpracname, vpracdate;  exit when vcursor%notfound;    dbms\_output.put\_line('과정번호: ' || vprognum || ' | 과정명: ' || vprogname || ' | 커리번호: ' || vcirnum || ' | 필기시험: ' || vwritename || '(' || vwritedate || ')' || ' | 실기시험: ' || vpracname || '(' || vpracdate || ')');  end loop;  close vcursor;  end procTest; | | | |

| 요구사항번호 | B-08 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 요구사항명 | 관리자 계정 | 작성자 | 최영진 |
| DML | | | |
| 성적, 성적우수장려금과 관련된 프로시저, 뷰, 트리거.  (프로시저)  --1. 전체목록 출력(교육생번호, 교육생이름, 성적번호, 금액)  -- procInsen  create or replace procedure procInsen  is  vstnum tblStudent.stNum%type;  vhname tblP.hName%type;  vscnum tblScore.scNum%type;  vinsentive tblInsentive.insentive%type;  cursor vcursor is  select  st.stNum as "교육생번호",  p.hname as "교육생이름",  sc.scNum as "성적번호",  i.insentive as "받은금액"  into  vstnum,  vhname,  vscnum,  vinsentive  from tblP p  inner join tblStudent st  on p.hNum = st.hnum  inner join tblScore sc  on st.stNum = sc.stNum  inner join tblInsentive i  on sc.scNum = i.scNum  order by i.insentive desc, st.stNum asc;  begin  open vcursor;  loop  fetch vcursor into vstnum, vhname, vscnum, vinsentive;  exit when vcursor%notfound;    dbms\_output.put\_line('교육생번호: ' || vstnum || ' | 교육생이름: ' || vhname || ' | 성적번호: ' || vscnum || ' | 받은금액: ' || vinsentive || '원');  end loop;  close vcursor;  end procInsen;  /  --2. 학생 장려금 조회  --procInsenSt(교육생번호)  create or replace procedure procInsenSt (  pstnum in varchar2  )  is  vstnum tblStudent.stNum%type;  vhname tblP.hName%type;  vinsentive tblInsentive.insentive%type;  cursor vcursor is  select  st.stNum as "교육생번호",  p.hname as "교육생이름",  i.insentive as "받은금액"  into  vstnum,  vhname,  vinsentive  from tblP p  inner join tblStudent st  on p.hNum = st.hnum  inner join tblScore sc  on st.stNum = sc.stNum  inner join tblInsentive i  on sc.scNum = i.scNum  where st.stNum = pstnum;  begin  open vcursor;  loop  fetch vcursor into vstnum, vhname, vinsentive;  exit when vcursor%notfound;    dbms\_output.put\_line('교육생번호: ' || vstnum || ' | 교육생이름: ' || vhname || ' | 받은금액: ' || vinsentive || '원');  end loop;  close vcursor;  end procInsenSt;  /  --3. 인센티브 추가  -- procInsenAdd(성적번호, 금액)  create or replace procedure procInsenAdd (  pscnum in number,  pinsentive in number  )  is  begin  insert into tblInsentive values ((select nvl(max(insenNum), 0) + 1 from tblInsentive), pscnum, pinsentive);  dbms\_output.put\_line('인센티브 추가 성공');  commit;  exception  when others then  dbms\_output.put\_line('인센티브 추가 실패');  rollback;  end procInsenAdd;  /  --4. 성적 인센티브 금액 수정  -- procInsenUpdate(인센티브번호, 금액)  create or replace procedure procInsenUpdate (  pinsennum in number,  pinsentive in number  )  is  begin  update tblInsentive  set insentive = pinsentive  where insenNum = pinsennum;  dbms\_output.put\_line('인센티브 금액 수정 성공');  commit;  exception  when others then  dbms\_output.put\_line('인센티브 금액 수정 실패');  rollback;  end procInsenUpdate;  /  --5. 인센티브 삭제  -- procInsenRemove(인센티브번호)  create or replace procedure procInsenRemove (  pinsennum in number  )  is  begin  delete from tblInsentive where insenNum = pinsenNum;  dbms\_output.put\_line('인센티브 삭제 성공');  commit;  exception  when others then  dbms\_output.put\_line('인센티브 삭제 실패');  rollback;  end procInsenRemove;  /  --6. 특정 개설 과정 선택 > 시험 문제 파일 등록 여부를 확인  --procTestApply(과정번호)  create or replace procedure procTestApply (  pprognum in number  )  is  vprognum tblProgList.progNum%type;  vprogname tblProgList.progName%type;  vnowend tblNow.nowEnd%type;  vnowstart tblNow.nowStart%type;  vclassNum tblNow.ClassNum%type;  vsubname tblSubject.subname%type;  vwritedate tblWriteT.writedate%type;  vtname tblTeacher.tName%type;  vbtitle tblBook.bTitle%type;  cursor vcursor is  select distinct  pl.progNum as "과정번호",  pl.progName as "개설과정명",  n.nowEnd,  n.nowStart,  n.classNum as "강의실명",  sb.subName as "개설 과목명",  wt.writeDate as "시험등록여부",  t.tName as "교사명",  b.btitle as "교재명"  into  vprognum,  vprogname,  vnowend,  vnowstart,  vclassNum,  vsubname,  vwritedate,  vtname,  vbtitle  from tblCurr c  inner join tblBookList bk  on c.cirNum = bk.cirNum  inner join tblBook b  on bk.bookNum = b.bookNum  inner join tblNow n  on c.progNum = n.progNum  inner join tblProgList pl  on n.progNum = pl.progNum  inner join tblTeacher t  on pl.TNum = t.TNum  inner join tblSubject sb  on c.subNum = sb.subNum  left outer join tblWriteT wt  on c.cirNum = wt.cirNum  inner join tblStudent st  on n.progNum = st.progNum  inner join tblScore sc  on st.stNum = sc.stNum  where c.progNum = pprognum;  begin  open vcursor;  loop  fetch vcursor into vprognum, vprogname, vnowend, vnowstart, vclassnum, vsubname, vwritedate, vtname, vbtitle;  exit when vcursor%notfound;    if vnowend - vnowstart in('166','167','168') then  if vwritedate is not null then  dbms\_output.put\_line('과정번호: ' || vprognum || ' | 개설과정명: ' || vprogname || ' | 개설과정기간: 5.5개월 | 강의실명: ' || vclassnum || '강의실' || ' | 개설과목명: ' || vsubname || ' | 시험등록여부: 시험등록O | 교사명: ' || vtname || ' | 교재명: ' || vbtitle);  elsif vwritedate is null then  dbms\_output.put\_line('과정번호: ' || vprognum || ' | 개설과정명: ' || vprogname || ' | 개설과정기간: 5.5개월 | 강의실명: ' || vclassnum || '강의실' || ' | 개설과목명: ' || vsubname || ' | 시험등록여부: 시험등록X | 교사명: ' || vtname || ' | 교재명: ' || vbtitle);  end if;    elsif vnowend - vnowstart in('181','182','184') then  if vwritedate is not null then  dbms\_output.put\_line('과정번호: ' || vprognum || ' | 개설과정명: ' || vprogname || ' | 개설과정기간: 6개월 | 강의실명: ' || vclassnum || '강의실' || ' | 개설과목명: ' || vsubname || ' | 시험등록여부: 시험등록O | 교사명: ' || vtname || ' | 교재명: ' || vbtitle);  elsif vwritedate is null then  dbms\_output.put\_line('과정번호: ' || vprognum || ' | 개설과정명: ' || vprogname || ' | 개설과정기간: 6개월 | 강의실명: ' || vclassnum || '강의실' || ' | 개설과목명: ' || vsubname || ' | 시험등록여부: 시험등록X | 교사명: ' || vtname || ' | 교재명: ' || vbtitle);  end if;  end if;  end loop;  close vcursor;  end procTestApply;  /  --7. 과목별 성적 출력(개설과정명, 개설과정기간, 강의실명 , 개설과목명 ,교사명, 교재명, 해당 과목의 교육생 성적(교육생 이름, 주민번호 뒷자리, 필기, 실기)  -- procScoreSub(과목번호);  create or replace procedure procScoreSub (  psubnum in number  )  is  vprognum tblNow.progNum%type;  vprogname tblProgList.progName%type;  vnowend tblNow.nowEnd%type;  vnowstart tblNow.nowStart%type;  vclassnum tblNow.classNum%type;  vsubname tblSubject.subname%type;  vtname tblTeacher.tName%type;  vbtitle tblBook.btitle%type;  vhname tblP.hname%type;  vhssn tblP.hSsn%type;  vwritescore tblScore.writeScore%type;  vpracscore tblScore.pracScore%type;  cursor vcursor is  select distinct  n.progNum as "과정번호",  pl.progName as "개설과정명",  n.nowEnd,  n.nowStart,  n.classNum as "강의실명",  sb.subName as "개설 과목명",  t.tName as "교사명",  b.btitle as "교재명",  p.hname as "교육생이름",  p.hSsn as "주민번호",  sc.writeScore as "필기점수",  sc.pracScore as "실기점수"  into  vprognum,  vprogname,  vnowend,  vnowstart,  vclassNum,  vsubname,  vtname,  vbtitle,  vhname,  vhssn,  vwritescore,  vpracscore  from tblCurr c  inner join tblBookList bk  on c.cirNum = bk.cirNum  inner join tblBook b  on bk.bookNum = b.bookNum  inner join tblNow n  on c.progNum = n.progNum  inner join tblProgList pl  on n.progNum = pl.progNum  inner join tblTeacher t  on pl.TNum = t.TNum  inner join tblSubject sb  on c.subNum = sb.subNum  left outer join tblWriteT wt  on c.cirNum = wt.cirNum  inner join tblStudent st  on n.progNum = st.progNum  inner join tblScore sc  on st.stNum = sc.stNum  inner join tblP p  on st.hNum = p.hNum    where sb.subnum = psubnum;  begin  open vcursor;  loop  fetch vcursor into vprognum,vprogname,vnowend,vnowstart,vclassNum,vsubname,vtname,vbtitle,vhname,vhssn,vwritescore,vpracscore;  exit when vcursor%notfound;    if vnowend - vnowstart in('166','167','168') then  dbms\_output.put\_line('과정번호: ' || vprognum || ' | 개설과정명: ' || vprogname || ' | 개설과정기간: 5.5개월 | 강의실명: ' || vclassnum || '강의실' || ' | 개설과목명: ' || vsubname || ' | 교사명: ' || vtname || ' | 교재명: ' || vbtitle || ' | 교육생이름: ' || vhname || ' | 주민번호뒷자리: ' || substr(vhssn, 8) || ' | 필기점수: ' || vwritescore || ' | 실기점수: ' || vpracscore);  elsif vnowend - vnowstart in('181','182','184') then  dbms\_output.put\_line('과정번호: ' || vprognum || ' | 개설과정명: ' || vprogname || ' | 개설과정기간: 6개월 | 강의실명: ' || vclassnum || '강의실' || ' | 개설과목명: ' || vsubname || ' | 교사명: ' || vtname || ' | 교재명: ' || vbtitle || ' | 교육생이름: ' || vhname || ' | 주민번호뒷자리: ' || substr(vhssn, 8) || ' | 필기점수: ' || vwritescore || ' | 실기점수: ' || vpracscore);  end if;  end loop;  close vcursor;  end procScoreSub;  /  --8. 교육생 개인별 성적 출력 (교육생 이름, 주민번호 뒷자리, 개설 과정명, 개설 과정기간, 강의실명, 성적 정보(개설 과목명, 개설 과목 기간, 교사명, 출결, 필기, 실기))  -- procScoreSt(교육생번호)  create or replace procedure procScoreSt (  pstnum in number  )  is  vprognum tblNow.progNum%type;  vstnum tblStudent.stNum%type;  vhname tblP.hname%type;  vhssn tblP.hSsn%type;  vprogname tblProgList.progName%type;  vclassnum tblNow.classNum%type;  vsubname tblSubject.subname%type;  vnowend tblNow.nowEnd%type;  vnowstart tblNow.nowStart%type;  vtname tblTeacher.tName%type;  vattscore tblScore.attScore%type;  vwritescore tblScore.writeScore%type;  vpracscore tblScore.pracScore%type;  cursor vcursor is  select  n.progNum as "과정번호",  st.stNum as "교육생번호",  p.hname as "교육생이름",  p.hSsn as "주민번호",  pl.progName as "개설과정명",  n.classNum as "강의실명",  sb.subName as "개설 과목명",  n.nowEnd,  n.nowStart,  t.tName as "교사명",  sc.attScore as "출결점수",  sc.writeScore as "필기점수",  sc.pracScore as "실기점수"  into  vprognum,  vstnum,  vhname,  vhssn,  vprogname,  vclassnum,  vsubname,  vnowend,  vnowstart,  vtname,  vattscore,  vwritescore,  vpracscore  from tblCurr c  inner join tblNow n  on c.progNum = n.progNum  inner join tblProgList pl  on n.progNum = pl.progNum  inner join tblTeacher t  on pl.TNum = t.TNum  inner join tblSubject sb  on c.subNum = sb.subNum  left outer join tblWriteT wt  on c.cirNum = wt.cirNum  inner join tblStudent st  on n.progNum = st.progNum  inner join tblScore sc  on st.stNum = sc.stNum  inner join tblP p  on st.hNum = p.hNum    where st.stNum = pstnum;  begin  open vcursor;  loop  fetch vcursor into vprognum,vstnum,vhname,vhssn,vprogname,vclassnum,vsubname,vnowend,vnowstart,vtname,vattscore,vwritescore,vpracscore;  exit when vcursor%notfound;    if vnowend - vnowstart in('166','167','168') then  dbms\_output.put\_line('과정번호: ' || vprognum || ' | 교육생번호: ' || vstnum || ' | 교육생이름: ' || vhname || ' | 주민번호뒷자리: ' || substr(vhSsn, 8) || ' | 개설과정명: ' || vprogname || ' | 개설과정기간: 5.5개월 | 강의실명: ' || vclassnum || '강의실' || ' | 개설과목명: ' || vsubname || ' | 교사명: ' || vtname || ' | 출결점수: ' || vattscore || ' | 필기점수: ' || vwritescore || ' | 실기점수: ' || vpracscore);  elsif vnowend - vnowstart in('181','182','184') then  dbms\_output.put\_line('과정번호: ' || vprognum || ' | 교육생번호: ' || vstnum || ' | 교육생이름: ' || vhname || ' | 주민번호뒷자리: ' || substr(vhSsn, 8) || ' | 개설과정명: ' || vprogname || ' | 개설과정기간: 6개월 | 강의실명: ' || vclassnum || '강의실' || ' | 개설과목명: ' || vsubname || ' | 교사명: ' || vtname || ' | 출결점수: ' || vattscore || ' | 필기점수: ' || vwritescore || ' | 실기점수: ' || vpracscore);  end if;  end loop;  close vcursor;  end procScoreSt;  /  (뷰)  --성적 우수 장려금  --1. 전체목록 출력(교육생번호, 교육생이름, 성적번호, 금액)  create or replace view vwInsenList  as  select  st.stNum as "교육생번호",  p.hname as "교육생이름",  sc.scNum as "성적번호",  i.insentive || '원' as "받은금액"  from tblP p  inner join tblStudent st  on p.hNum = st.hnum  inner join tblScore sc  on st.stNum = sc.stNum  inner join tblInsentive i  on sc.scNum = i.scNum  order by i.insentive desc, st.stNum asc;  (트리거)  --인센티브 입력시 금액이 0 이하일 때 실행되는 트리거.  create or replace trigger trgAddInsen  before  insert  on tblInsentive  declare  vinsentive number;  begin  dbms\_output.put\_line('인센티브 금액 트리거 실행');    if vinsentive <= 0 then  RAISE\_APPLICATION\_ERROR(-20100, '0원 이하의 금액은 입력이 불가능합니다.');  end if;  end trgAddInsen;  / | | | |

| 요구사항번호 | C-01 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 건의 게시판 입력 | 작성자 | 유희윤 |
| DML | | | |
| create or replace procedure proBoardIn(  pstnum number,  ptxt varchar  )  is  begin  insert into tblBoard values( (select NVL(Max(postnum), 0) +1 from tblBoard) , pstnum, '건의사항입니다.', ptxt, sysdate );  end proBoardIn; | | | |

| 요구사항번호 | C-02 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 요구사항명 | 로그인 성공시 = 교육생 개인정보 + 교육생이 수강한 과정명, 과정기간 (시작 년월일, 끝 년월일), 강의실 출력 | 작성자 | 김수민 |
| DML | | | |
| create or replace procedure ProcMyGrade(  pStNum number,  pStSsn varchar2  )  is  cursor vcursor is select  p.hName, p.hssn, p.htel, p.hcity, l.progName, n.nowStart, n.nowEnd, n.classNum  from tblStudent s  inner join tblP p  on s.hNum = p.hNum  inner join tblNow n  on s.progNum = n.progNum  inner join tblProgList l  on n.progNum = l.progNum  where s.stNum in (select b.stNum from tblP a inner join tblStudent b on a.hnum = b.hnum where b.hnum = s.hnum) and instr(p.hSsn, pStSsn)<>'0';  vresult number;  begin    select  count(\*) into vresult    from tblStudent s  inner join tblP p  on s.hNum = p.hNum  inner join tblNow n  on s.progNum = n.progNum  inner join tblProgList l  on n.progNum = l.progNum  where s.stNum in (select b.stNum from tblP a inner join tblStudent b on a.hnum = b.hnum where b.hnum = s.hnum) and instr(p.hSsn, pStSsn)<>'0';    if vresult > 0 then    for vrow in vcursor loop  dbms\_output.put\_line('교육생이름: '||vrow.hName||' 교육생 주민번호: '||vrow.hssn||' 교육생 전화번호: '||vrow.htel||' 교육생 거주지역: '||vrow.hcity||' 교육생이 수강한 과정명: '||vrow.progName||  ' 과정기간: '||'시작 '||to\_char(vrow.nowStart,'yyyy')||'년 '||to\_char(vrow.nowStart,'mm')||'월 '||to\_char(vrow.nowStart,'dd')||'일 '||'~ '||'종료 '||to\_char(vrow.nowEnd,'yyyy')||'년 '||to\_char(vrow.nowEnd,'mm')||'월 '||to\_char(vrow.nowStart,'dd')||'일 '  ||' 강의실 번호: '||vrow.classNum);  end loop;      else  dbms\_output.put\_line('입력 오류입니다.');  end if;  exception  when others then  dbms\_output.put\_line('오류로 로그인에 실패했습니다.');  end ProcMyGrade; | | | |

| 요구사항번호 | C-03 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 요구사항명 | 본인 출결 현황을 기간별(전체, 월,일) 조회할 수 있어야 한다. | 작성자 | 김수민 |
| DML | | | |
| create or replace procedure ProcMyAttData(  pStNum number  )  is    cursor vcursor is  select attdate,attstat from tblAttend where stNum = pStNum;  begin  for vrow in vcursor loop  dbms\_output.put\_line(to\_char(vrow.attDate, 'yyyy')||' 년 '||to\_char(vrow.attDate,'mm')||' 월 '||to\_char(vrow.attDate,'dd')||' 일 '||' 출결상황: '||vrow.attStat);  end loop;  end ProcMyAttData;  begin  ProcMyAttData(1);  end; | | | |

| 요구사항번호 | C-04 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 요구사항명 | 매일 근태관리를 기록할 수 있어야한다. (출근 1회, 퇴근 1회) | 작성자 | 김수민 |
| DML | | | |
| create or replace procedure procCheTime(  pstNum number,  pin date,  pGo date  )  is  vresult number;  vfeedback number;  cursor vcursor is select s.stnum from tblStudent s inner join tblNow n on s.progNum = n.progNum where n.nowend >sysdate and n.nowstart < sysdate and s.fail is null minus select a.stnum from tblAttend a where to\_char(a.attdate,'yyyy-mm-dd') =to\_char(sysdate,'yyyy-mm-dd');  vstNum number;  begin  select count(\*) into vresult from tblTime where stNum = pstNum;  select count(\*) into vfeedback from tblTime where stNum = pstNum and come is not null and go is not null;    if vfeedback <> 0 then  dbms\_output.put\_line('이미 퇴근하셨습니다.');  end if;    if pstNum <>0 then  --출근  if vfeedback = 0 and vresult = 0 and pin is not null and pgo is null then  insert into tblTime values (pstNum, to\_char(pin,'yyyy-mm-dd hh24:mi:ss'), null);    elsif vfeedback = 0 and vresult <> 0 and pin is not null and pgo is null then  dbms\_output.put\_line('이미 출근하셨습니다.');  end if;  --퇴근  if vfeedback = 0 and vresult <> 0 and pin is null and pgo is not null then  update tblTime set go = to\_char(pgo, 'yyyy-mm-dd hh24:mi:ss') where stNum = pstNum;  elsif vfeedback = 0 and vresult <> 0 and pin is null and pgo is not null then  dbms\_output.put\_line('이미 퇴근하셨습니다.');    elsif vfeedback = 0 and vresult = 0 and pin is null and pgo is not null then  dbms\_output.put\_line('잘못된 접근입니다.');  end if;  end if;  --관리자  if pstNum = 0 and pin is null and pgo is null and to\_char(sysdate,'hh24:mi:ss') between '22:00:00' and '23:59:59' then  open vcursor;  loop  fetch vcursor into vstNum;  exit when vcursor%notfound;    insert into tblAttend values (seqAttNum.nextVal, vstnum ,sysdate,'결석');    --중도탈락자  procdropout(vstNum);  end loop;  close vcursor;  delete from tblTime;  end if;    exception  when others then  dbms\_output.put\_line('출결에 실패하셨습니다.');  end; | | | |

| 요구사항번호 | C-04 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 요구사항명 | 교사 계정 | 작성자 | 최영진 |
| DML | | | |
| 시험과 관련된 프로시저와 뷰.  (프로시저)  --1. 커리별 필기문제출력(커리번호, 필기시험명, 필기문제명)  --procTestWQ(커리번호)  create or replace procedure procTestWQ(  pcirnum in number  )  is  num number := 1;  vcirnum tblCurr.cirnum%type;  vwritename tblWriteT.writeName%type;  vwriteq tblWrite.writeq%type;  cursor vcursor is  select  c.cirNum as "커리번호",  wt.writeName as "필기시험명",  wq.writeq as "필기시험문제"  into  vcirnum,  vwritename,  vwriteq  from tblCurr c  inner join tblWriteT wt  on c.cirNum = wt.cirNum  inner join tblWrite wq  on wt.writeTNum = wq.writeTNum  where c.cirNum = pcirnum;  begin  open vcursor;  loop  fetch vcursor into vcirnum, vwritename, vwriteq;  exit when vcursor%notfound;    dbms\_output.put\_line('커리번호(' || vcirnum || ') "' || vwritename || '" ' || num || '. ' || vwriteq);  num := num + 1;  end loop;  close vcursor;  end procTestWQ;  /  --2. 커리별 실기문제출력  --procTestPQ(커리번호)  create or replace procedure procTestPQ(  pcirnum in number  )  is  num number := 1;  vcirnum tblCurr.cirnum%type;  vpracname tblPracT.pracName%type;  vpracq tblPrac.pracq%type;  cursor vcursor is  select  c.cirNum as "커리번호",  pt.pracName as "실기시험명",  pq.Pracq as "실기시험문제"  into  vcirnum,  vpracname,  vpracq  from tblCurr c  inner join tblPracT pt  on c.cirNum = pt.cirNum  inner join tblPrac pq  on pt.PracTNum = pq.PracTNum  where c.cirNum = pcirnum;  begin  open vcursor;  loop  fetch vcursor into vcirnum, vpracname, vpracq;  exit when vcursor%notfound;    dbms\_output.put\_line('커리번호(' || vcirnum || ') "' || vpracname || '" ' || num || '. ' || vpracq);  num := num + 1;  end loop;  close vcursor;  end procTestPQ;  /  --3. 필기시험 추가  -- procTestWTAdd(커리번호, 시험날짜('YY/MM/DD'), '필기시험명')  create or replace procedure procTestWTAdd (  pcirnum in number,  pwritedate in date,  pwritename in varchar2  )  is  begin  insert into tblWriteT values ((select nvl(max(writeTNum), 0) + 1 from tblWriteT), pcirnum, pwritedate, pwritename);  dbms\_output.put\_line('필기시험 추가 성공');  commit;  exception  when others then  dbms\_output.put\_line('필기시험 추가 실패');  rollback;  end procTestWTAdd;  /  --4. 실기시험 추가  --호출 procTestPTAdd(커리번호, 시험날짜('YY/MM/DD'), '실기시험명')  create or replace procedure procTestPTAdd (  pcirnum in number,  ppracdate in date,  ppracname in varchar2  )  is  begin  insert into tblPracT values ((select nvl(max(pracTNum), 0) + 1 from tblPracT), pcirnum, ppracdate, ppracname);  dbms\_output.put\_line('필기시험 추가 성공');  commit;  exception  when others then  dbms\_output.put\_line('필기시험 추가 실패');  rollback;  end procTestPTAdd;  /  --5. 필기시험 날짜 수정  --호출 procTestWTUpdate(필기시험번호, 시험날짜('YY/MM/DD'));  create or replace procedure procTestWTUpdate(  pwritetnum in number,  pwritedate in date  )  is  begin  update tblWriteT  set writedate = pwritedate  where writeTNum = pwritetnum;  dbms\_output.put\_line('필기시험 날짜 수정 성공');  commit;  exception  when others then  dbms\_output.put\_line('필기시험 날짜 수정 실패');  rollback;  end procTestWtUpdate;  /  --6. 실기시험 날짜 수정  --호출 procTestPTUpdate(실기시험번호, 시험날짜('YY/MM/DD'));  create or replace procedure procTestPTUpdate(  ppractnum in number,  ppracdate in date  )  is  begin  update tblPracT  set pracdate = ppracdate  where pracTNum = ppractnum;  dbms\_output.put\_line('실기시험 날짜 수정 성공');  commit;  exception  when others then  dbms\_output.put\_line('실기시험 날짜 수정 실패');  rollback;  end procTestPTUpdate;  /  --7. 필기문제 추가  --호출 procTestWQAdd(필기시험번호, '필기문제');  create or replace procedure procTestWQAdd (  pwritetnum in number,  pwriteq in varchar2  )  is  begin  insert into tblWrite values ((select nvl(max(writeTNum), 0) + 1 from tblWrite), pwritetnum, pwriteq);  dbms\_output.put\_line('필기문제 추가 성공');  commit;  exception  when others then  dbms\_output.put\_line('필기문제 추가 실패');  rollback;  end procTestWQAdd;  /  --8. 실기문제 추가  --호출 procTestWQAdd(필기시험번호, '실기문제');  create or replace procedure procTestPQAdd (  ppractnum in number,  ppracq in varchar2  )  is  begin  insert into tblPrac values ((select nvl(max(pracTNum), 0) + 1 from tblPrac), ppractnum, ppracq);  dbms\_output.put\_line('실기문제 추가 성공');  commit;  exception  when others then  dbms\_output.put\_line('실기문제 추가 실패');  rollback;  end procTestPQAdd;  /  --9. 필기문제 내용 수정  --호출 procTestWQUpdate(필기문제번호, '필기문제')  create or replace procedure procTestWQUpdate(  pwritenum in number,  pwriteq in varchar2  )  is  begin  update tblWrite  set writeq = pwriteq  where writeNum = pwritenum;  dbms\_output.put\_line('필기문제 내용 수정 성공');  commit;  exception  when others then  dbms\_output.put\_line('필기문제 내용 수정 실패');  rollback;  end procTestWQUpdate;  /  --10. 실기문제 내용 수정  --호출 procTestPQUpdate(실기문제번호, '실기문제')  create or replace procedure procTestPQUpdate(  ppracnum in number,  ppracq in varchar2  )  is  begin  update tblPrac  set pracq = ppracq  where pracNum = ppracnum;  dbms\_output.put\_line('실기문제 내용 수정 성공');  commit;  exception  when others then  dbms\_output.put\_line('실기문제 내용 수정 실패');  rollback;  end procTestPQUpdate;  /  --11. 필기문제 삭제  --호출 procTestWQRemove(필기문제번호)  create or replace procedure procTestWQRemove (  pwritenum in number  )  is  begin  delete from tblWrite where writeNum = pwriteNum;  dbms\_output.put\_line('필기문제 삭제 성공');  commit;  exception  when others then  dbms\_output.put\_line('필기문제 삭제 실패');  rollback;  end procTestWQRemove;  /  --12. 실기문제 삭제  --호출 procTestPQRemove(실기문제번호)  create or replace procedure procTestPQRemove (  ppracnum in number  )  is  begin  delete from tblPrac where pracNum = ppracnum;  dbms\_output.put\_line('실기문제 삭제 성공');  commit;  exception  when others then  dbms\_output.put\_line('실기문제 삭제 실패');  rollback;  end procTestPQRemove;  /  (뷰)  --1. 전체 과정의 필기/실기시험명 출력 (과정번호, 과정명, 필기시험명, 필기시험날짜, 실기시험명, 실기시험날짜)  create or replace view vwTestList  as  select  p.progNum as "과정번호",  p.progName as "과정명",  c.cirNum as "커리번호",  wt.writeName as "필기시험명",  wt.writeDate as "필기시험날짜",  pt.pracName as "실기시험명",  pt.pracDate as "실기시험날짜"  from tblProgList p  inner join tblNow n  on p.progNum = n.progNum  inner join tblCurr c  on n.progNum = c.progNum  inner join tblWriteT wt  on c.cirNum = wt.cirNum  inner join tblPracT pt  on c.cirNum = pt.cirNum;  --2. 전체 필기 문제 출력(커리번호, 필기시험명, 필기시험문제)  create or replace view vwTestWQ  as  select  c.cirNum as "커리번호",  wt.writeName as "필기시험명",  wq.writeq as "필기시험문제"  from tblCurr c  inner join tblWriteT wt  on c.cirNum = wt.cirNum  inner join tblWrite wq  on wt.writeTNum = wq.writeTNum;  --3. 전체 실기 문제 출력(커리번호, 실기시험명, 실기시험문제)  create or replace view vwTestPQ  as  select  pt.pracName as "실기시험명",  pq.Pracq as "실기시험문제"  from tblCurr c  inner join tblPracT pt  on c.cirNum = pt.cirNum  inner join tblPrac pq  on pt.PracTNum = pq.PracTNum;  --4. 시험등록X & 개강 안한 반 확인  create or replace view vwCheckEmpty  as  select  c.cirnum as "커리번호",  c.subnum as "과목번호",  c.cirstart as "커리시작일",  c.cirend as "커리종료일",  c.prognum as "과정번호",  wt.writeTNum as "시험번호"  from tblCurr c  left outer join tblWriteT wt  on c.cirNum = wt.cirNum  where wt.writeTNum is null and c.cirstart >= '2024-01-01'; | | | |

| 요구사항번호 | C-05 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 요구사항명 | 교사 계정 | 작성자 | 최영진 |
| DML | | | |
| 교사의 성적 입출력과 관련된 프로시저.  --1. 강의를 마친 교사 본인 과목 목록 출력(과목번호, 과정명, 과정기간(시작 년월일, 끝 년월일), 강의실, 과목명, 과목기간(시작 년월일, 끝 년월일), 교재명, 출결, 필기, 실기 배점, 성적 등록 여부)  -- procTSub(교사번호)  create or replace procedure procTSub (  ptnum in number  )  is  vtnum tblTeacher.tnum%type;  vtname tblTeacher.tName%type;  vprogname tblProgList.progName%type;  vnowend tblNow.nowEnd%type;  vnowstart tblNow.nowStart%type;  vclassnum tblNow.classNum%type;  vcirnum tblCurr.cirNum%type;  vsubname tblSubject.subName%type;  vcirstart tblCurr.cirStart%type;  vcirend tblCurr.cirEnd%type;  vbtitle tblBook.bTitle%type;  vcirnumw tblWriteT.cirNum%type;  vattpoint number;  vwritepoint number;  vpracpoint number;  cursor vcursor is  select  t.TNum as "교사번호",  t.Tname as "교사명",  pl.progName as "과정명",  n.nowStart as 과정기간1,  n.nowEnd as 과정기간2,  n.classNum as "강의실",  c.cirNum as "과정과목번호",  sb.subName as "과목명",  c.cirStart as "과목기간1",  c.cirEnd as "과목기간2",  b.bTitle as "교재명",  (select attPoint from tblPoint where cirNum = c.cirNum) as 출결배점,  (select writePoint from tblPoint where cirNum = c.cirNum) as 필기배점,  (select pracPoint from tblPoint where cirNum = c.cirNum) as 실기배점,  wt.cirNum  into  vtnum, vtname,vprogname,vnowend,vnowstart,vclassnum,vcirnum,vsubname,vcirstart,vcirend,vbtitle,vcirnumw,vattpoint,vwritepoint,vpracpoint  from tblnow n  inner join tblProgList pl  on n.progNum = pl.progNum  inner join tblTeacher t  on pl.TNum = t.TNum  inner join tblCurr c  on n.progNum = c.progNum  inner join tblSubject sb  on c.subNum = sb.subNum  inner join tblBookList bk  on c.cirNum = bk.cirNum  inner join tblBook b  on bk.bookNum = b.bookNum  left outer join tblWriteT wt  on c.cirNum = wt.cirNum  where t.TNum = ptnum and c.cirEnd < sysdate  order by c.cirStart;  begin  open vcursor;  loop  fetch vcursor into vtnum, vtname,vprogname,vnowend,vnowstart,vclassnum,vcirnum,vsubname,vcirstart,vcirend,vbtitle,vcirnumw,vattpoint,vwritepoint,vpracpoint;  exit when vcursor%notfound;    if vcirnumw is not null then  dbms\_output.put\_line('교사번호: ' || vtnum || ' | 교사명: ' || vtname || ' | 과정명: ' || vprogname || ' | 과정기간: ' || vnowstart || ' ~ ' || vnowend || ' | 강의실: ' || vclassnum || '강의실 | 과정과목번호: ' || vcirnum || ' | 과목명: ' || vsubname || ' | 과목기간: ' || vcirstart || ' ~ ' || vcirend || ' | 교재명: ' || vbtitle || ' | 출결배점: ' || vattpoint || ' | 필기배점: ' || vwritepoint || ' | 실기배점: ' || vpracpoint || ' | 성적등록O');  elsif vcirnumw is null then  dbms\_output.put\_line('교사번호: ' || vtnum || ' | 교사명: ' || vtname || ' | 과정명: ' || vprogname || ' | 과정기간: ' || vnowstart || ' ~ ' || vnowend || ' | 강의실: ' || vclassnum || '강의실 | 과정과목번호: ' || vcirnum || ' | 과목명: ' || vsubname || ' | 과목기간: ' || vcirstart || ' ~ ' || vcirend || ' | 교재명: ' || vbtitle || ' | 출결배점: ' || vattpoint || ' | 필기배점: ' || vwritepoint || ' | 실기배점: ' || vpracpoint || ' | 성적등록X');  end if;    end loop;  close vcursor;  end procTSub;  /  --2. 과목번호로 특정과목 선택 시, (교육생 정보(이름, 전화번호, 수료 또는 중도탈락), 성적 (출결, 필기, 실기 점수) 출력  -- procTScore(커리번호)  create or replace procedure procTScore (  pcirnum in number  )  is  vhname tblP.hName%type;  vhtel tblP.htel%type;  vcompstat tblStudent.compStat%type;  vfail tblStudent.fail%type;  vsubname tblSubject.subname%type;  vattscore tblScore.attScore%type;  vwritescore tblScore.writeScore%type;  vpracscore tblScore.pracScore%type;  cursor vcursor is  select  p.hName as "교육생 명",  p.htel as "교육생 전화번호",  st.compStat,  st.fail as "탈락날짜",  sb.subName as "과목명",  sc.attScore as "출결점수",  sc.writeScore as "필기점수",  sc.pracScore as "실기점수"  into  vhname,vhtel,vcompstat,vfail,vsubname,vattscore,vwritescore,vpracscore  from tblnow n  inner join tblProgList pl  on n.progNum = pl.progNum  inner join tblCurr c  on n.progNum = c.progNum  inner join tblSubject sb  on c.subNum = sb.subNum  inner join tblStudent st  on n.progNum = st.progNum  inner join tblP p  on st.hNum = p.hNum  left outer join tblScore sc  on st.stNum = sc.stNum  where c.cirNum = pcirnum;  begin  open vcursor;  loop  fetch vcursor into vhname,vhtel,vcompstat,vfail,vsubname,vattscore,vwritescore,vpracscore;  exit when vcursor%notfound;    if vcompStat = 'Y' then  dbms\_output.put\_line('교육생이름: ' || vhname || ' | 교육생번호: ' || vhtel || ' | 수료 | 탈락날짜: ' || vfail || ' | 과목명: ' || vsubname || ' | 출결점수: ' || vattscore || ' | 필기점수: ' || vwritescore || ' | 실기점수: ' || vpracscore);  elsif vcompStat = 'N' then  dbms\_output.put\_line('교육생이름: ' || vhname || ' | 교육생번호: ' || vhtel || ' | 중도탈락 | 탈락날짜: ' || vfail || ' | 과목명: ' || vsubname || ' | 출결점수: ' || vattscore || ' | 필기점수: ' || vwritescore || ' | 실기점수: ' || vpracscore);  end if;  end loop;  close vcursor;  end procTScore;  / | | | |

| 요구사항번호 | C-06 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 요구사항명 | 교사 계정 | 작성자 | 최영진 |
| DML | | | |
| 성적과 성적장학금에 관련된 뷰.  --성적 정보  --1. 전체 시험 문제 파일 등록 여부  create or replace view vwTestApply  as  select distinct  pl.progNum as "과정번호",  pl.progName as "개설과정명",  case  when n.nowEnd - n.nowStart in('166','167','168') then '5.5개월'  when n.nowEnd - n.nowStart in('181','182','184') then '6개월'  end as "개설과정기간",  n.classNum || '강의실' as "강의실명",  sb.subName as "개설 과목명",  case  when wt.writeDate is not null then '시험등록O'  else '시험등록X'  end as "시험등록여부",t.tName as "교사명",  b.btitle as "교재명"  from tblCurr c  inner join tblBookList bk  on c.cirNum = bk.cirNum  inner join tblBook b  on bk.bookNum = b.bookNum  inner join tblNow n  on c.progNum = n.progNum  inner join tblProgList pl  on n.progNum = pl.progNum  inner join tblTeacher t  on pl.TNum = t.TNum  inner join tblSubject sb  on c.subNum = sb.subNum  left outer join tblWriteT wt  on c.cirNum = wt.cirNum  inner join tblStudent st  on n.progNum = st.progNum  inner join tblScore sc  on st.stNum = sc.stNum  order by pl.prognum asc;  --2. 전체과목 출력 >> 개설과정명, 개설과정기간, 강의실명 , 개설과목명 ,교사명, 교재명, 해당 과목의 교육생 성적(교육생 이름, 주민번호 뒷자리, 필기, 실기)  create or replace view vwScoreSub  as  select distinct  n.progNum as "과정번호",  pl.progName as "개설과정명",  case  when n.nowEnd - n.nowStart in('166','167','168') then '5.5개월'  when n.nowEnd - n.nowStart in('181','182','184') then '6개월'  end as "개설과정기간",  n.classNum || '강의실' as "강의실명",  sb.subName as "개설 과목명",  t.tName as "교사명",  b.btitle as "교재명",  p.hname as "교육생이름",  substr(p.hSsn, 8) as "주민번호뒷자리",  sc.writeScore as "필기점수",  sc.pracScore as "실기점수"  from tblCurr c  inner join tblBookList bk  on c.cirNum = bk.cirNum  inner join tblBook b  on bk.bookNum = b.bookNum  inner join tblNow n  on c.progNum = n.progNum  inner join tblProgList pl  on n.progNum = pl.progNum  inner join tblTeacher t  on pl.TNum = t.TNum  inner join tblSubject sb  on c.subNum = sb.subNum  left outer join tblWriteT wt  on c.cirNum = wt.cirNum  inner join tblStudent st  on n.progNum = st.progNum  inner join tblScore sc  on st.stNum = sc.stNum  inner join tblP p  on st.hNum = p.hNum  order by n.progNum;  --3. 강의를 마친 교사 본인 과목 목록  create or replace view vwTSubList  as  select  t.TNum as "교사번호",  t.Tname as "교사명",  pl.progName as "과정명",  n.nowStart || ' ~ ' || n.nowEnd as 과정기간,  n.classNum || '강의실' as "강의실",  c.cirNum as "과정과목번호",  sb.subName as "과목명",  c.cirStart || ' ~ ' || c.cirEnd "과목기간",  b.bTitle as "교재명",  (select attPoint from tblPoint where cirNum = c.cirNum) as 출결배점,  (select writePoint from tblPoint where cirNum = c.cirNum) as 필기배점,  (select pracPoint from tblPoint where cirNum = c.cirNum) as 실기배점,  case  when wt.cirNum is not null then '성적등록O'  else '성적등록X'  end as "성적등록여부"  from tblnow n  inner join tblProgList pl  on n.progNum = pl.progNum  inner join tblTeacher t  on pl.TNum = t.TNum  inner join tblCurr c  on n.progNum = c.progNum  inner join tblSubject sb  on c.subNum = sb.subNum  inner join tblBookList bk  on c.cirNum = bk.cirNum  inner join tblBook b  on bk.bookNum = b.bookNum  left outer join tblWriteT wt  on c.cirNum = wt.cirNum  where t.TNum = <교사번호> and c.cirEnd < sysdate  order by c.cirStart;  --4. 특정과목의 교육생정보 & 성적  create or replace view vwTSubScore  as  select  p.hName as "교육생 명",  p.htel as "교육생 전화번호",  case  when st.compStat = 'Y' then '수료'  when st.compStat = 'N' then '중도탈락'  else '과정중'  end as "수료여부",  st.fail as "탈락날짜",  sb.subName as "과목명",  (select attScore from tblScore where stNum = st.stNum and rownum = 1) as "출결성적",  (select writeScore from tblScore where stNum = st.stNum and rownum = 1) as "필기성적",  (select pracScore from tblScore where stNum = st.stNum and rownum = 1) as "실기성적"  from tblnow n  inner join tblProgList pl  on n.progNum = pl.progNum  inner join tblCurr c  on n.progNum = c.progNum  inner join tblSubject sb  on c.subNum = sb.subNum  inner join tblStudent st  on n.progNum = st.progNum  inner join tblP p  on st.hNum = p.hNum  where c.cirNum = <커리번호>; | | | |

| 요구사항번호 | D-01 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 계정 | 작성자 | 최영진 |
| DML | | | |
| 성적과 성적장학금에 관련된 뷰.  --1. 교육생 전용 장려금 조회  create or replace view vwInsenST  as  select  st.stNum as "교육생번호",  p.hname as "교육생이름",  i.insentive || '원' as "받은금액"  from tblP p  inner join tblStudent st  on p.hNum = st.hnum  inner join tblScore sc  on st.stNum = sc.stNum  inner join tblInsentive i  on sc.scNum = i.scNum  where st.stNum = <교육생번호>;  --2. 교육생 전용 출력 (교육생 이름, 주민번호 뒷자리, 개설 과정명, 개설 과정기간, 강의실명, 성적 정보(개설 과목명, 개설 과목 기간, 교사명, 출결, 필기, 실기))  create or replace view vwScoreSt  as  select distinct  n.progNum as "과정번호",  st.stNum as "교육생번호",  p.hname as "교육생이름",  substr(p.hSsn, 8) as "주민번호뒷자리",  pl.progName as "개설과정명",  n.classNum || '강의실' as "강의실명",  sb.subName as "개설 과목명",  case  when n.nowEnd - n.nowStart in('166','167','168') then '5.5개월'  when n.nowEnd - n.nowStart in('181','182','184') then '6개월'  end as "개설과정기간",  t.tName as "교사명",  sc.attScore as "출결점수",  sc.writeScore as "필기점수",  sc.pracScore as "실기점수"  from tblCurr c  inner join tblNow n  on c.progNum = n.progNum  inner join tblProgList pl  on n.progNum = pl.progNum  inner join tblTeacher t  on pl.TNum = t.TNum  inner join tblSubject sb  on c.subNum = sb.subNum  left outer join tblWriteT wt  on c.cirNum = wt.cirNum  inner join tblStudent st  on n.progNum = st.progNum  inner join tblScore sc  on st.stNum = sc.stNum  inner join tblP p  on st.hNum = p.hNum    where st.stNum = <교육생번호>; | | | |

| 요구사항번호 | D-02 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 요구사항명 | 특정 교사별 상담 내역을 조회하는 프로시저 | 작성자 | 윤혜린 |
| DML | | | |
| -- 특정 교사의 이름으로 해당 교사의 상담일지를 조회하는 프로시저  create or replace procedure procDiCheck(  teacherName in tblTeacher.tName%type  )  is  pstName tblP.hName%type;  pprogNum tblStudent.progNum%type;  pprogName tblProglist.progname%type;  pTName tblTeacher.Tname%type;  ptTopic tblTalk.talkTopic%type;  pTDate tblDiary.diDate%type;    cursor vcursor is  select  p.hName as 교육생이름,  s.prognum as 과정번호,  l.progName as 과정명,  a.TName as 담당교사,  t.talkTopic as 상담주제,  d.diDate as 상담일자    from tblTalk t  inner join tblDiary d on t.talkNum = d.talkNum  inner join tblStudent s on s.stNum = d.stNum  inner join tblP p on p.hNum = s.hNum  inner join tblProgList l on s.progNum = l.prognum  inner join tblTeacher a on l.TNum = a.TNum  where a.TName = teacherName  order by 상담일자;    begin  open vcursor;  loop  fetch vcursor into pstName, pprogNum, pprogName, pTName, ptTopic, pTDate;  exit when vcursor%notfound;  dbms\_output.put\_line('-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  dbms\_output.put\_line('교육생이름 : ' || pstName|| ' 과정번호 : ' || pprogNum || ' 과정명 : ' ||pprogName|| ' 담당교사 : ' || pTName || ' 상담주제 : ' || ptTopic || ' 상담일자 : ' || pTDate );  dbms\_output.put\_line('-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  end loop;  close vcursor;  end; | | | |

| 요구사항번호 | D-03 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 요구사항명 | 특정 학생의 거주지역에 속하는 회사 정보를 통해 추천 회사 목록을 출력하는 프로시저 | 작성자 | 윤혜린 |
| DML | | | |
| -- 특정 학생의 이름과 거주지역을 출력한 뒤, 추천 회사 목록을 출력한다.  -- 추천 회사 목록에는 해당 회사명과 지역 정보가 출력된다.  create or replace procedure procFirm(  studentName in tblP.hName%type  )  is  pstName tblP.hName%type;  pstCity tblP.hcity%type;  pfirmname tblFirm.firmName%type;  pfirmCity tblFirm.firmCity%type;  pnum number;  lnum number;          cursor vcursor is  select  f.firmname, f.firmcity into pfirmname, pfirmCity  from  tblP p  inner join tblStudent s on p.hNum = s.hNum  inner join tblFirm f on p.hCity = f.firmcity  where p.hname = studentName;        begin    select hcity, hName into pstCity, pstName from tblP where hName = studentName;  dbms\_output.put\_line('----------------------------------------------');  dbms\_output.put\_line(' ' || pstName || ' 교육생의 거주지역 : ' || pstCity );  dbms\_output.put\_line('----------------------------------------------');  dbms\_output.put\_line('----------------------------------------------');  dbms\_output.put\_line(' 추천 회사 목록 ');  dbms\_output.put\_line('----------------------------------------------');      lnum := 1;  open vcursor;  loop  fetch vcursor into pfirmName, pfirmCity;  exit when vcursor%notfound;  dbms\_output.put\_line('----------------------------------------------');  dbms\_output.put\_line('|' || lnum || '|' ||' 회사명 : ' || pfirmName || ' ' || '지역 : ' || pfirmCity);  dbms\_output.put\_line('----------------------------------------------');  lnum := lnum + 1;  end loop;  close vcursor;  end procFirm; | | | |

| 요구사항번호 | D-04 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 요구사항명 | 특정 학생이 보유한 자격증 개수와 자격증 목록을 출력하는 프로시저 | 작성자 | 윤혜린 |
| DML | | | |
| -- 학생의 이름을 통해 해당 학생이 보유하고 있는 자격증 개수와 자격증 정보 목록을 출력한다.  create or replace procedure procLicCnt(  studentName in tblP.hName%type  )  is  pstName tblP.hName%type;  plicName tblLic.licName%type;  pnum number;  lnum number;          cursor vcursor is  select li.licName into plicName  from  tblP p  inner join tblStudent s on p.hNum = s.hNum  inner join tblLicList l on s.stNum = l.stNum  inner join tblLic li on l.licNum = li.licNum  where p.hName = studentName;        begin    select p.hName, count(\*) into pstName, pnum  from  tblP p  inner join tblStudent s on p.hNum = s.hNum  inner join tblLicList l on s.stNum = l.stNum  inner join tblLic li on l.licNum = li.licNum  where p.hName = studentName  group by p.hName;  dbms\_output.put\_line('----------------------------------------------');  dbms\_output.put\_line(pstName || ' 교육생이 취득한 자격증 갯수 : ' || pnum );  dbms\_output.put\_line('----------------------------------------------');      lnum := 1;  open vcursor;  loop  fetch vcursor into plicName;  exit when vcursor%notfound;  dbms\_output.put\_line('----------------------------------------------');  dbms\_output.put\_line('|' || lnum || '|' || ' ' || plicName );  dbms\_output.put\_line('----------------------------------------------');  lnum := lnum + 1;  end loop;  close vcursor;  end procLicCnt; | | | |

| 요구사항번호 | D-05 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 요구사항명 | 특정 학생의 출결정보를 통해 출결급여를 계산하는 프로시저 | 작성자 | 윤혜린 |
| DML | | | |
| -- 특정 학생 이름을 통해 출결 정보들을 불러오고 결석 일수에 따라 출결급여를 계산한다.  -- 출결 급여는 현재 날짜(sysdate) 기준 바로 이전 달의 출결 정보들을 이용하여 계산한다.  -- 만약 해당 학생이 중도탈락생이라면 출결 급여를 출력하지 않고 중도탈락 일과 알림 메세지를 출력한다.  create or replace procedure procAttPay (  pname in varchar2  )  is    pfaildate tblStudent.fail%type;  ppremonth date;  pyear varchar2(4);  pmonth varchar2(2);  pnum number;      begin    pyear := to\_char(sysdate, 'yyyy');  pmonth := to\_char(ADD\_MONTHS(sysdate, -1), 'MM');  ppremonth := ADD\_MONTHS( TRUNC(sysdate,'MM'), -1 );    select count(\*) into pnum  from  tblAttend a  inner join tblStudent s on a.stNum = s.stNum  inner join tblNow n on n.progNum = s.progNum  inner join tblP p on p.hNum = s.hNum  where a.attStat = '결석' and  a.attDate between ppremonth and last\_day(ppremonth) and  p.hName = pname  group by p.hName;    select fail into pfaildate from tblStudent s inner join tblP p on s.hNum = p.hNum where p.hName = pname;    if pfaildate is null then  pnum := 20 - pnum;  dbms\_output.put\_line(pyear || '년 ' || pmonth || '월 ' || pname || ' 학생의 출결급여');  dbms\_output.put\_line('학생 이름 : ' || pname);  dbms\_output.put\_line('출결 급여 : ' || pnum || ' 만원');    elsif pfaildate is not null then  dbms\_output.put\_line(pfaildate || '자에 중도탈락한 교육생입니다.');    else pnum := 20;  end if;      exception  when no\_data\_found then pnum := 20;      dbms\_output.put\_line(pyear || '년 ' || pmonth || '월 ' || pname || ' 학생의 출결급여');  dbms\_output.put\_line('학생 이름 : ' || pname);  dbms\_output.put\_line('출결 급여 : ' || pnum || ' 만원');  end procAttPay; | | | |

| 요구사항번호 | D-06 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 요구사항명 | 한 과정을 수료한 학생이 해당 과정에 대해 강의 평가를 진행하는 프로시저 | 작성자 | 윤혜린 |
| DML | | | |
| -- 학생번호를 입력받아 해당 학생이 수료학생 테이블에도 존재하는지 확인한 뒤 레코드 값이 있다는 것이 확인되면 평가 테이블에 수료학생 번호와 평가 점수 입력 후 평가 대상 과정 정보를 출력한다.  -- 수료학생 테이블에 데이터가 존재하지 않을 때, 즉 수료 전인 상태일 땐 평가 정보를 입력하지 않고 과정 수료 후 강의 평가를 할 수 있다는 경고 메세지를 출력한다.  create or replace procedure procEvalUpdate(  EstNum in tblStudent.stNum%type, -- 교육생 번호(PK)  Esatisfy in tblTEval.satisfy%type -- 평가 점수  )  is  pprogNum tblProgList.progNum%type; -- 과정번호(PK)  pprogName tblProgList.progName%type; -- 과정명  pstuName tblP.hName%type; -- 교육생 이름  pstNum tblStudent.stNum%type; -- 교육생 번호(PK)  pcsNum tblCompStu.csNum%type; -- 수료생 번호(PK)  pcnt number;  begin  select count(\*) into pcnt  from (  select  s.stNum, c.csnum  from  tblStudent s  inner join tblCompStu c on c.stNum = s.stNum  where s.stNum = EstNum  );      if pcnt = 1 then    select  s.stNum, p.hName, l.progNum, l.progName, c.csNum into pstNum, pstuName, pprogNum, pprogName, pcsNum  from tblNow n  inner join tblStudent s on s.progNum = n.progNum  inner join tblCompStu c on c.stNum = s.stNum  inner join tblTEval e on e.CSNUm = c.CSNum  inner join tblProgList l on l.progNum = n.progNum  inner join tblP p on p.hNum = s.hNum    where s.stNum = EstNum;    insert into tblTEval values ((select nvl(max(evalNum), 0) + 1 from tblTEval), pcsNum, Esatisfy);  dbms\_output.put\_line('안녕하세요 ' || pstuName || '님, ' || pprogNum || '번 ' || pprogName || '과정에 대한 평가가 완료되었습니다.' );    else dbms\_output.put\_line('과정 수료 후 강의 평가를 할 수 있습니다.');  end if;    end procEvalUpdate; | | | |

| 인덱스번호 | Index01 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 인덱스명 | idxScore | 작성자 | 최영진 |
| DML | | | |
| --교육생별 성적을 조회하는 인덱스  create index idxScore on tblScore(stNum); | | | |

| 인덱스번호 | Index02 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 인덱스명 | idxWriteQ | 작성자 | 최영진 |
| DML | | | |
| --필기시험번호별 문제를 조회하는 인덱스  create index idxWriteQ on tblWrite(writeTNum); | | | |

| 인덱스번호 | Index03 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 인덱스명 | idxPracQ | 작성자 | 최영진 |
| DML | | | |
| --실기시험번호별 문제를 조회하는 인덱스  create index idxPracQ on tblPrac(PracTNum); | | | |

| 인덱스번호 | Index04 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 인덱스명 | idxTask | 작성자 | 최영진 |
| DML | | | |
| --교육생별 과제를 조회하는 인덱스  create index idxTask on tblTask(stNum); | | | |

| 인덱스번호 | Index05 | 작성일 | 2023-04-09 |
| --- | --- | --- | --- |
| 인덱스명 | idxStudent | 작성자 | 김수민 |
| DML | | | |
| --교육생별 과제를 조회하는 인덱스  create index idxStudent on tblStudent(stNum); | | | |