| DB Object 스크립트 | 문서번호 |  |
| --- | --- | --- |
| 작성자 | 4조 |
| 작성일자 | 24.03.08 |

| 요구사항번호 | B-03(관리자) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 특정 교사 정보 출력 | 작성자 | 김희연 |
| DML | | | |
| create or replace procedure procTeacherInfo (  pname in varchar2  )  is  vtName tblTeacher.name%type;  vsName tblSubject.name%type;  vsStart tblopensubjectlist.startdate%type;  vsEnd tblopensubjectlist.enddate%type;  vcName tblcurriculum.name%type;  vcstart tblopencurriculum.startdate%type;  vcend tblopencurriculum.enddate%type;  vbname tblTextbook.name%type;  vrname tblroom.name%type;  vStat varchar2(15);    cursor vcursor is  select  s\_name as "개설 과목명",  osl\_startdate as 과목시작일,  osl\_enddate as 과목종료일,  c\_name as 과정명,  oc\_startdate as 과정시작일,  oc\_enddate as 과정종료일,  b.name as 교재명,  r\_name as 강의실,  case  when osl\_startdate < sysdate and sysdate > osl\_enddate then '강의종료'  when osl\_startdate > sysdate then '강의예정'  when osl\_startdate <= sysdate and sysdate <= osl\_enddate then '강의중'  end as "강의상태"  from vwCurriculum v  inner join tblTextbook b  on b.seq\_textbook = v.seq\_textbook  inner join tblcurriculumprogress g  on v.seq\_curriculumprogress = g.seq\_curriculumprogress  inner join tblteacher t  on v.seq\_teacher = t.seq\_teacher  where t.name = pname  order by 과목시작일;  begin  open vcursor;  loop  fetch vcursor into vsName, vsStart, vsEnd, vcname, vcstart, vcend, vbname, vrname, vStat;  exit when vcursor%notfound;  dbms\_output.put\_line('교사명: ' || pName || '| 과목명: ' || vsName || '| 과목시작일: ' || vsStart || '| 과목종료일: ' || vsEnd || '| 과정명: ' || vcname || '| 과정시작일: ' || vcstart || '| 과정종료일: ' || vcend || '| 교재명: ' || vbname || '| 강의실명: '|| vrname || '| 강의상태: ' || vStat);  dbms\_output.put\_line('-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  end loop;  close vcursor;  end procTeacherInfo;  /  begin  procTeacherInfo('김희연');  end;  /  create or replace procedure procTeacherWhoCan(  pname in varchar2  )  is  vTname tblteacher.name%type;  vSubName tblsubject.name%type;  cursor vcursor is  select  distinct t.name, s.name  from tblteacher t  inner join tblAvailableSubjectList asl  on t.seq\_teacher = asl.seq\_teacher  inner join tblsubject s  on s.seq\_subject = asl.seq\_subject  where s.name = pname;  begin  open vcursor;  loop  fetch vcursor into vTname, vSubName;  exit when vcursor%notfound;  dbms\_output.put\_line('이름: ' || vTname || ' | 과목명: ' || vSubName);  dbms\_output.put\_line('-----------------------------------------------');  end loop;  close vcursor;  end procTeacherWhoCan;  /  begin  procTeacherWhoCan('자바');  end;  / | | | |

| 요구사항번호 | B-04(관리자) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 개설 교육 과정 번호를 입력하여  개설 과목 정보 확인 | 작성자 | 김희연 |
| DML | | | |
| /\* 특정 개설 과정 선택 \*/  --개설 교육 과정 번호를 입력하여 개설 과목 정보 확인  create or replace procedure procCurriDetail (  pnum in number  )  is  vSName tblSubject.name%type;  vsStart tblOpensubjectlist.startdate%type;  vsEnd tblOpensubjectlist.enddate%type;  vtName tbltrainees.name%type;  vTel varchar2(15);  vRegdate date;  vStat varchar2(30);    cursor vcursor is    select  v.s\_name as "과목명",  v.osl\_startdate as "과목시작일",  v.osl\_enddate as "과목종료일",  t.name as "학생명",  t.tel as "전화번호",  t.registrationDate as "등록일",  (case  when tl.status is not null then tl.status  when v.seq\_curriculumprogress = 1 then '강의 진행 예정'  when v.seq\_curriculumprogress = 2 then '강의 진행 중'  end) as "수료 여부"  from vwcurriculum v  left outer join tbltraineelist tl  on tl.seq\_opencurriculum = v.seq\_opencurriculum  inner join tbltrainees t  on t.seq\_trainee = tl.seq\_trainee  where v.seq\_opencurriculum = pnum;  begin  open vcursor;  loop  fetch vcursor into vSName, vsStart, vsEnd, vtName, vTel, vRegdate, vStat;  exit when vcursor%notfound;  dbms\_output.put\_line('과목명: ' || vSName || ' | 과목시작일: ' || vsStart || ' | 과목종료일: ' || vsEnd || ' | 학생명: ' || vtName || ' | 전화번호: ' || vTel || ' | 등록일: ' || vRegdate || ' | 수료여부: ' || vStat);  dbms\_output.put\_line('----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  end loop;  close vcursor;  end procCurriDetail;  /  begin  procCurriDetail(1); --개설교육과정 번호  end;  / | | | |

| 요구사항번호 | B-07(관리자) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 시험 관리 및 성적 조회 | 작성자 | 장원준 |
| DML | | | |
| -- b-7  --과목별  CREATE OR REPLACE PROCEDURE procCurriculumBySubject (  p\_subject\_name IN VARCHAR2  )  IS  v\_output VARCHAR2(4000);  BEGIN  FOR cur IN (  SELECT vc.c\_name || ', ' || cp.period || ', ' || vc.r\_name || ', ' || vc.s\_name || ', ' || tea.name || ', ' || tb.name || ', ' || vt.t\_name || ', ' || vt.t\_ssn || ', ' || vg.writtengrade || ', ' || vg.practicalgrade AS output\_line  FROM vwcurriculum vc  INNER JOIN vwtrainees vt ON vt.seq\_opencurriculum = vc.seq\_opencurriculum  INNER JOIN vwgrades vg ON vg.seq\_traineelist = vt.seq\_traineelist  INNER JOIN tblteacher tea ON tea.seq\_teacher = vc.seq\_teacher  INNER JOIN tbltextbook tb ON tb.seq\_textbook = vc.seq\_textbook  INNER JOIN tblcourseperiod cp ON cp.seq\_courseperiod = vc.seq\_courseperiod  WHERE vc.c\_name = p\_subject\_name  )  LOOP  v\_output := cur.output\_line;  DBMS\_OUTPUT.PUT\_LINE(v\_output);  END LOOP;  END procCurriculumBySubject;  /  /  /  BEGIN  procCurriculumBySubject ('자바(Java)기반 Web과 스마트플랫폼 Full-Stack 과정(A)');  END;  /  -- 특정 개설 과정  CREATE OR REPLACE PROCEDURE procGradesCourseInfo(  p\_course IN vwcurriculum.c\_name%TYPE  ) IS  CURSOR c\_grades IS  SELECT vt.t\_name, vc.c\_name, vc.s\_name, t.name AS teacher\_name,  vg.writtengrade, vg.practicalgrade,  (CASE WHEN vg.writtengrade IS NOT NULL THEN '필기성적입력완료' END) AS written,  (CASE WHEN vg.practicalgrade IS NOT NULL THEN '실기성적입력완료' END) AS practical  FROM vwgrades vg  INNER JOIN vwtrainees vt ON vt.seq\_traineelist = vg.seq\_traineelist  INNER JOIN vwcurriculum vc ON vc.seq\_subject = vg.seq\_subject  INNER JOIN tblteacher t ON t.seq\_teacher = vc.seq\_teacher  WHERE vc.c\_name = p\_course  GROUP BY vt.t\_name, vc.c\_name, vc.s\_name, t.name, vg.writtengrade, vg.practicalgrade;  v\_grades c\_grades%ROWTYPE;  written VARCHAR2(50);  practical VARCHAR2(50);  BEGIN  OPEN c\_grades;  LOOP  FETCH c\_grades INTO v\_grades;  EXIT WHEN c\_grades%NOTFOUND;    written := v\_grades.written;  practical := v\_grades.practical;  DBMS\_OUTPUT.PUT\_LINE('과정 이름: ' || v\_grades.c\_name || ', 주제: ' || v\_grades.s\_name ||  ', 강사 이름: ' || v\_grades.teacher\_name || ', 필기 성적: ' ||  v\_grades.writtengrade || ', 실기 성적: ' || v\_grades.practicalgrade ||  ', 필기성적입력 ' || written || ', 실기성적입력 ' || practical);    dbms\_output.put\_line('----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  END LOOP;  CLOSE c\_grades;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('해당 교육생의 성적 정보를 찾을 수 없습니다.');  WHEN OTHERS THEN  ROLLBACK;  RAISE;  END procGradesCourseInfo;  /  BEGIN  procGradesCourseInfo('AWS와 Docker를 활용한 Java Full-Stack 과정(B)');  END;  /  --교육생 개인 별  CREATE OR REPLACE PROCEDURE procGradesInfoByTrainee(  p\_trainee\_name IN VARCHAR2  ) IS  BEGIN  FOR rec IN (  SELECT vt.t\_name AS trainee\_name, vt.t\_ssn, vc.c\_name AS course\_name, cp.period,  vc.r\_name AS classroom\_name, vc.s\_name AS subject\_name, vc.period AS subject\_period,  tea.name AS teacher\_name, ass.situation, vg.writtengrade, vg.practicalgrade  FROM vwgrades vg  INNER JOIN vwtrainees vt ON vt.seq\_traineelist = vg.seq\_traineelist  INNER JOIN vwcurriculum vc ON vc.seq\_subject = vg.seq\_subject  INNER JOIN tblcourseperiod cp ON cp.seq\_courseperiod = vc.seq\_courseperiod  INNER JOIN tblattendancestatus ass ON ass.seq\_attendancestatus = vt.seq\_attendancestatus  INNER JOIN tblteacher tea ON tea.seq\_teacher = vc.seq\_teacher  WHERE vt.t\_name = p\_trainee\_name  GROUP BY vt.t\_name, vt.t\_ssn, vc.c\_name, cp.period, vc.oc\_startdate, vc.oc\_enddate,  vc.r\_name, vc.c\_name, vc.period, tea.name, ass.situation, vc.s\_name, vg.writtengrade, vg.practicalgrade  )  LOOP  DBMS\_OUTPUT.PUT\_LINE('교육생 이름: ' || rec.trainee\_name || ', 주민등록번호: ' || rec.t\_ssn ||  ', 과정명: ' || rec.course\_name || ', 교시: ' || rec.period ||  ', 강의실: ' || rec.classroom\_name || ', 주제: ' || rec.subject\_name ||  ', 기간: ' || rec.subject\_period || ', 교사 이름: ' || rec.teacher\_name ||  ', 출결상태: ' || rec.situation || ', 필기성적: ' ||  rec.writtengrade || ', 실기성적: ' || rec.practicalgrade);  END LOOP;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('해당 교육생의 성적 정보를 찾을 수 없습니다.');  WHEN OTHERS THEN  ROLLBACK;  RAISE;  END procGradesInfoByTrainee;  /  /  BEGIN  procGradesInfoByTrainee('천유서');  END;  / | | | |

| 요구사항번호 | B-08(관리자) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 개설 과정 특정 기간 출결 조회 | 작성자 | 장원준 |
| DML | | | |
| CREATE OR REPLACE PROCEDURE procCourseInfo(  p\_course IN tblCurriculum.name%TYPE,  p\_start\_date IN DATE,  p\_end\_date IN DATE    ) IS  CURSOR c\_course IS  SELECT t.name AS trainee\_name, a.day, c.name AS course\_name, ad.situation  FROM tblAttendance a  INNER JOIN tblTraineeList tl ON a.seq\_traineeList = tl.seq\_traineelist  INNER JOIN tblopencurriculum oc ON oc.seq\_opencurriculum = tl.seq\_opencurriculum  INNER JOIN tblCurriculum c ON c.seq\_curriculum = oc.seq\_opencurriculum  INNER JOIN tblattendancestatus ad ON ad.seq\_attendancestatus = a.seq\_attendancestatus  INNER JOIN tblTrainees t ON t.seq\_trainee = tl.seq\_trainee  WHERE c.name = p\_course  AND a.day BETWEEN p\_start\_date AND p\_end\_date  GROUP BY t.name, a.day, c.name, ad.situation;    v\_course c\_course%ROWTYPE;  BEGIN  OPEN c\_course;  LOOP  FETCH c\_course INTO v\_course;  EXIT WHEN c\_course%NOTFOUND;    DBMS\_OUTPUT.PUT\_LINE('교육생 이름: ' || v\_course.trainee\_name || ', 날짜: ' || v\_course.day || ', 과정 이름: ' || v\_course.course\_name || ', 출석 상황: ' || v\_course.situation);  dbms\_output.put\_line('----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  END LOOP;  CLOSE c\_course;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('해당 과정의 정보를 찾을 수 없습니다.');  WHEN OTHERS THEN  ROLLBACK;  RAISE;  END procCourseInfo;  /  BEGIN  procCourseInfo('AWS 클라우드와 Elasticsearch를 활용한 Java Full-Stack 과정(B)', TO\_DATE('2023-11-01', 'YYYY-MM-DD'), TO\_DATE('2023-12-30', 'YYYY-MM-DD'));  END;  /  -- 2. 특정 인원  CREATE OR REPLACE PROCEDURE procTraineeAttendanceInfo(  p\_name IN vwtrainees.t\_name%TYPE  ) IS  CURSOR c\_attendance IS  SELECT vt.t\_name, vt.a\_day, vc.c\_name, vt.situation  FROM vwtrainees vt  INNER JOIN vwcurriculum vc ON vt.seq\_opencurriculum = vc.seq\_opencurriculum  WHERE vt.t\_name = p\_name  GROUP BY vt.t\_name, vt.a\_day, vc.c\_name, vt.situation;  v\_attendance c\_attendance%ROWTYPE;  BEGIN  OPEN c\_attendance;  LOOP  FETCH c\_attendance INTO v\_attendance;  EXIT WHEN c\_attendance%NOTFOUND;  DBMS\_OUTPUT.PUT\_LINE('교육생 이름: ' || v\_attendance.t\_name || ', 날짜: ' || v\_attendance.a\_day || ', 과정 이름: ' || v\_attendance.c\_name || ', 상태: ' || v\_attendance.situation);  dbms\_output.put\_line('----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  END LOOP;  CLOSE c\_attendance;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('해당 이름을 가진 교육생의 출석 정보를 찾을 수 없습니다.');  WHEN OTHERS THEN  ROLLBACK;  RAISE;  END procTraineeAttendanceInfo;  /  BEGIN  procTraineeAttendanceInfo('제류혁');  END;  / | | | |

| 요구사항번호 | B-13(관리자) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 출결 인정 서류 관리 | 작성자 | 정원혁 |
| DML | | | |
| create or replace procedure procInsertattendancePapers (  p\_seq\_attendancePapers tblAttendancePapers.seq\_attendancePapers%type,  p\_document tblAttendancePapers.document%type  )  is  vdocument varchar2(50);  begin  vdocument := p\_document;  if (vdocument like '%코로나%' or  vdocument like '%사망%' or  vdocument like '%입원%' or  vdocument like '%의사%' or  vdocument like '%출생 신고서%' or  vdocument like '%국가 자격증 시험%' or  vdocument like '%예비군%' or  vdocument like '%면접%' or  vdocument like '%국가%' or  vdocument like '%병가%') then  update tblAttendancePapers  set admitattendance = '출석 인정'  where seq\_attendancePapers = p\_seq\_attendancePapers;  dbms\_output.put\_line('출석이 인정되었습니다.');  else  update tblAttendancePapers  set admitattendance = '출석 미인정'  where seq\_attendancePapers = p\_seq\_attendancePapers;  dbms\_output.put\_line('출석으로 인정되지 않는 서류입니다.');  end if;  end procInsertattendancePapers;  /  begin  procInsertattendancePapers(1, '경찰서');  end;  / | | | |

| 요구사항번호 | B-14(관리자) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 기자재 & 사물함 관리 | 작성자 | 정원혁 |
| DML | | | |
| create or replace procedure procManagelocker (  p\_status tblTraineeList.status%type,  p\_seq\_locker tblLocker.seq\_locker%type  )  is  begin  if p\_status = '수료' then  update tblLocker  set seq\_traineeList = null  where seq\_locker = p\_seq\_locker;  dbms\_output.put\_line('교육생이 수료했습니다. 사물함이 비었습니다.');  end if;  end procManagelocker;  /  declare  v\_seq\_locker tblLocker.seq\_locker%type;  begin  v\_seq\_locker := 111;  procManagelocker('수료', v\_seq\_locker);  end;  /  -- 교육생 목록에 리스트가 들어갔을 때 사물함이 자동 배정  create or replace procedure procAutoInsertlocker (  p\_seq\_traineeList tblTraineeList.seq\_traineeList%type,  p\_seq\_trainee tblTraineeList.seq\_trainee%type,  p\_seq\_openCurriculum tblTraineeList.seq\_openCurriculum%type  )  is  v\_seq\_locker tblLocker.seq\_locker%type;    begin  select seq\_locker into v\_seq\_locker  from tblLocker  where seq\_traineeList is null  and seq\_locker between 21 and 178  and rownum = 1;    update tblLocker  set seq\_traineelist = p\_seq\_traineeList  where seq\_locker = v\_seq\_locker;    -- 배정 결과를 출력합니다.  dbms\_output.put\_line('교육생이 사물함에 자동으로 배정되었습니다.');  exception  when no\_data\_found then  dbms\_output.put\_line('비어있는 사물함이 없습니다.');  when others then  dbms\_output.put\_line('오류가 발생했습니다.');  end procAutoInsertlocker;  /  begin  procAutoInsertlocker(71, 51, 3);  end;  / | | | |

| 요구사항번호 | B-16(관리자) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 취업현황 조회 및 관리 | 작성자 | 정원혁 |
| DML | | | |
| create or replace procedure procSelectemploymentStatus (  p\_curriculum in number  )  is  p\_seq\_employmentStatus tblEmploymentStatus.seq\_employmentStatus%type;  p\_seq\_traineeList tblTraineeList.seq\_traineeList%type;  p\_seq\_openCurriculum tblOpenCurriculum.seq\_openCurriculum%type;  p\_name tblCurriculum.name%type;  p\_estatus tblEmploymentStatus.status%type;  p\_city tblEmploymentStatus.city%type;  p\_field tblEmploymentStatus.field%type;  p\_salary tblEmploymentStatus.salary%type;  cursor vcursor is  select  es.seq\_employmentStatus as "번호",  tl.seq\_traineeList as "교육생 목록 번호",  oc.seq\_openCurriculum as "개설 교육과정 번호",  c.name as "교육 과정",  es.status as "취업 여부",  es.city as "지역",  es.field as "분야",  es.salary as "연봉"  from tblEmploymentStatus es  left outer join tblTraineeList tl  on tl.seq\_traineeList = es.seq\_traineeList  inner join tblOpenCurriculum oc  on oc.seq\_openCurriculum = tl.seq\_openCurriculum  inner join tblCurriculum c  on oc.seq\_curriculum = c.seq\_curriculum  where oc.seq\_openCurriculum = p\_curriculum  order by es.seq\_employmentStatus;  begin  open vcursor;  loop  fetch vcursor into p\_seq\_employmentStatus, p\_seq\_traineeList, p\_seq\_openCurriculum, p\_name, p\_estatus, p\_city, p\_field, p\_salary;  exit when vcursor%notfound;  dbms\_output.put\_line('번호: ' || p\_seq\_employmentStatus || '| 교육생 목록 번호: ' || p\_seq\_traineeList || '| 교육 과정: ' || p\_seq\_openCurriculum || '| 취업 여부: ' || p\_estatus || '| 지역: ' || p\_city || '| 분야: ' || p\_field || '| 연봉: ' || p\_salary);  dbms\_output.put\_line('-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  end loop;  close vcursor;  end procSelectemploymentStatus;  /  begin  procSelectemploymentStatus(1);  end;  / | | | |

| 요구사항번호 | C-1(교사) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 교사의 강의 스케줄 조회 | 작성자 | 김민곤 |
| DML | | | |
| CREATE OR REPLACE PROCEDURE proctschedule(  ptname VARCHAR2  )  IS  vc\_name VARCHAR2(200);  vcp\_status varchar2(30);  voc\_startdate date;  voc\_enddate date;  vs\_seq\_subject NUMBER;  vs\_name varchar2(200);  vos\_startdate DATE;  vos\_enddate DATE;  vtb\_name VARCHAR2(200);  vtl\_seq\_opencurriculum number;    CURSOR vcursor IS  select distinct  c.name as 과정명,  cp.status as 과정상태,  oc.startDate as 시작일,  oc.endDate as 종료일,  s.seq\_subject as 과목번호,  s.name as 과목명,  os.startdate as 과목시작일,  os.enddate as 과목종료일,  tb.name as 교재명,  count(tl.seq\_opencurriculum) as "교육생 등록 인원"  from tblTeacher t  inner join tblOpenSubjectList os  on t.seq\_teacher = os.seq\_teacher  inner join tblOpenCurriculum oc  on oc.seq\_openCurriculum = os.seq\_openCurriculum  inner join tblCurriculumProgress cp  on cp.seq\_curriculumProgress = oc.seq\_curriculumProgress  inner join tblTraineeList tl  on oc.seq\_openCurriculum = tl.seq\_openCurriculum  inner join tblTrainees tr  on tr.seq\_trainee = tl.seq\_trainee  inner join tblsubjectlist sl  on sl.seq\_subjectList = os.seq\_subjectList  inner join tblsubject s  on s.seq\_subject = sl.seq\_subject  inner join tblRoom r  on r.seq\_room = oc.seq\_room  inner join tblCurriculum c  on c.seq\_Curriculum = oc.seq\_Curriculum  inner join tblCoursePeriod cop  on cop.seq\_coursePeriod = c.seq\_coursePeriod  inner join tblTextBook tb  on tb.seq\_textbook = os.seq\_textbook  where t.name = ptname  group by c.name, cp.status, oc.startDate,oc.endDate,s.seq\_subject,s.name,  os.startdate,os.enddate,tb.name;    BEGIN  OPEN vcursor;  LOOP  FETCH vcursor INTO vc\_name,vcp\_status, voc\_startdate,voc\_enddate,vs\_seq\_subject, vs\_name, vos\_startdate, vos\_enddate, vtb\_name, vtl\_seq\_opencurriculum;  EXIT WHEN vcursor%NOTFOUND;  DBMS\_OUTPUT.PUT\_LINE( '과정명: ' || vc\_name || '| 과정상태' || vcp\_status|| '| 과정시작일: ' || voc\_startdate || ' | 과정종료일: ' || voc\_enddate ||  '| 과목번호'||vs\_seq\_subject|| '| 과목명: ' ||vs\_name ||'| 과목시작일: ' || vos\_startdate || '| 과목종료일: ' || vos\_enddate ||  '| 교재명'|| vtb\_name||'| 등록인원' || vtl\_seq\_opencurriculum);  dbms\_output.put\_line ('----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  END LOOP;  CLOSE vcursor;  END proctschedule;  /  begin  proctschedule('김민곤');  end;  /  --c-1.2교육중인 과정에 등록된 교육생 정보(교육생 이름, 전화번호, 등록일, 수료 또는 중도탈락)을 확인할 수 있어야 한다.  CREATE OR REPLACE PROCEDURE proctinfo(  ptname VARCHAR2  )  IS  vc\_name varchar2(200);  vtr\_name varchar2(10);  vtr\_tel varchar2(15);  vtr\_registrationDate date;  vtl\_status varchar2(15);  CURSOR vcursor IS  select distinct  c.name as 과정명,  tr.name as 교육생이름,  tr.tel as 교육생전화번호,  tr.registrationDate as 교육생등록일,  tl.status as 교육생상태  from tblTeacher t  inner join tblOpenSubjectList os  on t.seq\_teacher = os.seq\_teacher  inner join tblOpenCurriculum oc  on oc.seq\_openCurriculum = os.seq\_openCurriculum  inner join tblCurriculumProgress cp  on cp.seq\_curriculumProgress = oc.seq\_curriculumProgress  inner join tblTraineeList tl  on oc.seq\_openCurriculum = tl.seq\_openCurriculum  inner join tblTrainees tr  on tr.seq\_trainee = tl.seq\_trainee  inner join tblsubjectlist sl  on sl.seq\_subjectList = os.seq\_subjectList  inner join tblsubject s  on s.seq\_subject = sl.seq\_subject  inner join tblRoom r  on r.seq\_room = oc.seq\_room  inner join tblCurriculum c  on c.seq\_Curriculum = oc.seq\_Curriculum  inner join tblCoursePeriod cp  on cp.seq\_coursePeriod = c.seq\_coursePeriod  inner join tblTextBook tb  on tb.seq\_textbook = os.seq\_textbook  where t.name = ptname;    BEGIN  OPEN vcursor;  LOOP  FETCH vcursor INTO vc\_name, vtr\_name,vtr\_tel, vtr\_registrationDate, vtl\_status;  EXIT WHEN vcursor%NOTFOUND;  DBMS\_OUTPUT.PUT\_LINE( '과정명: ' || vc\_name || '| 교육생이름: ' || vtr\_name|| '| 전화번호: ' || vtr\_tel || ' | 등록일: ' || vtr\_registrationDate ||  '| 상태:'||vtl\_status);  dbms\_output.put\_line ('----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  END LOOP;  CLOSE vcursor;  END proctinfo;  /  begin  proctinfo('김민곤');  end;  / | | | |

| 요구사항번호 | C-2(교사) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 배점 입출력 및 시험 문제,날짜 등록 | 작성자 | 정혜정 |
| DML | | | |
| create or replace procedure procTestInfoIn(  pSeq\_teacher in number,  pSeq\_testInfo in number,  pSeq\_openSubjectList in number,  pWrittenpoints in number,  pPracticalpoints in number,  pAtendancepoints in number  )  is  vEnddate date;  begin  -- tblopensubjectlist 테이블의 enddate 확인  select osl.enddate into vEnddate  from tblOpenSubjectList osl  inner join tblTestInfo ti  on ti.seq\_openSubjectList = osl.seq\_openSubjectList  where osl.seq\_teacher = pSeq\_teacher  and osl.seq\_OpenSubjectList = pSeq\_openSubjectList;  -- enddate가 오늘 이전인 경우에만 배점 입력  if vEnddate < sysdate then  -- 배점 입력  insert into tblTestInfo (seq\_testInfo, seq\_openSubjectList, writtenpoints, practicalpoints, attendancepoints)  values (pSeq\_testInfo, pSeq\_openSubjectList, pWrittenpoints, pPracticalpoints, pAtendancepoints);  if sql%rowcount = 0 then  dbms\_output.put\_line('해당 시험 정보가 존재하지 않습니다.');  else  dbms\_output.put\_line('배점이 등록되었습니다.');  end if;  else  dbms\_output.put\_line('해당 과목은 아직 종료되지 않았습니다.');  end if;  end procTestInfoIn;  /  begin  procTestInfoIn(1, 1, 1, 40, 40, 20);  end;  /  -- 교사 번호를 입력하면 본인이 수강 중인 과목 목록 다 출력  -- 과목 목록 출력 시 과목번호, 과정명, 과정기간(시작 년월일, 끝 년월일), 강의실, 과목명, 과목기간(시작 년월일, 끝 년월일), 교재명, 출결, 필기, 실기 배점 등이 출력되고,  -- 특정 과목을 과목번호로 선택 시 출결 배점, 필기 배점, 실기 배점, 시험 날짜, 시험 문제를 입력할 수 있는 화면으로 연결되어야 한다. (자바에서 구현해야할 듯)  -- 배점 등록이 안 된 과목인 경우는 과목 정보가 출력될 때 배점 부분은 null 값으로 출력한다.  create or replace procedure procSubjectListOut (  pSeq\_teacher in number  )  is  cursor vcursor is  select  vc.seq\_openSubjectList "과정 번호",  vc.seq\_subject "과목 번호",  vc.c\_name "과정명",  vc.oc\_startDate "과정 시작일",  vc.oc\_enddate "과정 종료일",  vc.r\_name 강의실,  vc.s\_name 과목명,  vc.osl\_startdate "과목 시작일",  vc.osl\_enddate "과목 종료일",  b.name 교재명,  ti.attendancePoints "출결 배점",  ti.writtenPoints "필기 배점",  ti.practicalPoints "실기 배점"  from vwCurriculum vc  inner join tblTextbook b  on b.seq\_textbook = vc.seq\_textbook  inner join tblTestInfo ti  on ti.seq\_openSubjectList = vc.seq\_openSubjectList  where vc.seq\_teacher = pSeq\_teacher  and vc.osl\_enddate < sysdate;    vrecord vcursor%rowtype;    begin  -- 정보 출력  open vcursor;  loop  fetch vcursor into vrecord;  exit when vcursor%notfound;  dbms\_output.put\_line('과정 번호: ' || vrecord."과정 번호");  dbms\_output.put\_line('과목 번호: ' || vrecord."과목 번호");  dbms\_output.put\_line('과정명: ' || vrecord."과정명");  dbms\_output.put\_line('과정 시작일: ' || vrecord."과정 시작일");  dbms\_output.put\_line('과정 종료일: ' || vrecord."과정 종료일");  dbms\_output.put\_line('강의실: ' || vrecord.강의실);  dbms\_output.put\_line('과목명: ' || vrecord.과목명);  dbms\_output.put\_line('과목 시작일: ' || vrecord."과목 시작일");  dbms\_output.put\_line('과목 종료일: ' || vrecord."과목 종료일");  dbms\_output.put\_line('교재명: ' || vrecord.교재명);  dbms\_output.put\_line('출결 배점: ' || vrecord."출결 배점");  dbms\_output.put\_line('필기 배점: ' || vrecord."필기 배점");  dbms\_output.put\_line('실기 배점: ' || vrecord."실기 배점");  dbms\_output.put\_line('-----------------------------------------------------------------------------------');  end loop;  close vcursor;  exception  when no\_data\_found then  dbms\_output.put\_line('해당 과목이 존재하지 않습니다.');  when others then  dbms\_output.put\_line('오류 발생: ' || SQLERRM);  end procSubjectListOut;  /  begin  procSubjectListOut(1);  end;  /  -- 2. 교사는 자신이 강의를 마친 과목의 목록 중에서 특정 과목을 선택하고 해당 과목의 배점 정보를 출결, 필기, 실기로 구분해서 등록할 수 있어야 한다.(위에서 구현 완료) 시험 날짜, 시험 문제를 추가할 수 있어야 한다.  -- 2.5. 시험 날짜 추가  create or replace procedure procTestDateIn (  pSeq\_teacher in number,  pSeq\_testInfo in number,  pWrittenDate in date,  pPracticalDate in date  )  is  vEnddate date;  begin  -- tblopensubjectlist 테이블의 enddate 확인  select osl.enddate into vEnddate  from tblOpenSubjectList osl  inner join tblTestInfo ti  on ti.seq\_openSubjectList = osl.seq\_openSubjectList  where osl.seq\_teacher = pSeq\_teacher  and ti.seq\_testinfo = pSeq\_testInfo;  -- 시험 날짜 추가  update tblTestInfo  set writtenDate = TO\_DATE(pWrittenDate, 'YYYY-MM-DD'),  practicalDate = TO\_DATE(pPracticalDate, 'YYYY-MM-DD')  where seq\_testInfo = pSeq\_testInfo;  if sql%rowcount = 0 then  dbms\_output.put\_line('해당 시험 정보가 존재하지 않습니다.');  else  dbms\_output.put\_line('시험 날짜가 추가되었습니다.');  end if;  end procTestDateIn;  /  begin  procTestDateIn(1, 1, TO\_DATE('2023-10-03', 'YYYY-MM-DD'), TO\_DATE('2023-10-04', 'YYYY-MM-DD'));  end;  /  -- 2.6. 시험 문제 추가  create or replace procedure procTestQuestionIn (  pSeq\_teacher in number,  pSeq\_question in number,  pSeq\_openSubjectList in number,  pSeq\_examPaper in number,  pQuestion in varchar2,  pAnswer in varchar2,  pKind in varchar2  )  is  vEnddate date;  begin  -- tblopensubjectlist 테이블의 enddate 확인  select osl.enddate into vEnddate  from tblOpenSubjectList osl  inner join tblTestInfo ti  on ti.seq\_openSubjectList = osl.seq\_openSubjectList  where osl.seq\_teacher = pSeq\_teacher  and ti.seq\_openSubjectList = pSeq\_openSubjectList;  -- 시험 문제 추가  insert into tblQuestion (seq\_question, question, answer)  values (pSeq\_question, pQuestion, pAnswer);    insert into Tblexampaper (seq\_examPaper, seq\_question, seq\_openSubjectList, kind)  values (pSeq\_examPaper, pSeq\_question, pSeq\_openSubjectList, pKind);  if sql%rowcount = 0 then  dbms\_output.put\_line('해당 시험 정보가 존재하지 않습니다.');  else  dbms\_output.put\_line('시험 문제가 추가되었습니다.');  end if;  end procTestQuestionIn;  /  begin  procTestQuestionIn(1, 1, 1, 1, '자바 소스 확장자는?', '3', '필기');  end;  / | | | |

| 요구사항번호 | C-3(교사) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 성적 입 출력 | 작성자 | 정혜정 |
| DML | | | |
| create or replace procedure procTestGradesOut (  pSeq\_teacher in number,  pSeq\_openSubjectList in number  )  is  cursor vcursor is  select  distinct tl.seq\_traineeList "교육생 목록 번호",  t.name 이름,  t.tel 전화번호,  tl.status 상태,  tl.day "상태 완료 날짜",  g.attendanceGrade "출결 점수",  g.writtenGrade "필기 점수",  g.practicalGrade "실기 점수"  from tblTraineeList tl  left outer join tblOpenSubjectList osl  on osl.seq\_openCurriculum = tl.seq\_openCurriculum  left outer join tblGrades g  on g.seq\_traineeList = tl.seq\_traineeList  left outer join tblTrainees t  on t.seq\_trainee = tl.seq\_trainee  where osl.seq\_teacher = pSeq\_teacher  and osl.seq\_openSubjectList = pSeq\_openSubjectList  and osl.endDate < sysdate;    vrecord vcursor%rowtype;  vCompletion varchar2(10);    begin  -- 정보 출력  open vcursor;  loop  fetch vcursor into vrecord;  exit when vcursor%notfound;    -- 출결, 필기, 실기 점수가 모두 입력된 경우에만 "완료" 출력  if vrecord."출결 점수" is not null and vrecord."필기 점수" is not null and vrecord."실기 점수" is not null then  vCompletion := '완료';  else  vCompletion := '미완료';  end if;    -- 정보 출력  dbms\_output.put\_line('교육생 목록 번호: ' || vrecord."교육생 목록 번호");  dbms\_output.put\_line('이름: ' || vrecord."이름");  dbms\_output.put\_line('전화번호: ' || vrecord."전화번호");  dbms\_output.put\_line('상태: ' || vrecord."상태");  dbms\_output.put\_line('상태 완료 날짜: ' || vrecord."상태 완료 날짜");  dbms\_output.put\_line('출결 점수: ' || vrecord."출결 점수");  dbms\_output.put\_line('필기 점수: ' || vrecord."필기 점수");  dbms\_output.put\_line('실기 점수: ' || vrecord."실기 점수");  dbms\_output.put\_line('성적 등록 여부: ' || vCompletion);  dbms\_output.put\_line('-----------------------------------------------------------------------------------');  end loop;    close vcursor;  exception  when no\_data\_found then  dbms\_output.put\_line('해당 정보가 없습니다.');  when others then  dbms\_output.put\_line('오류 발생: ' || SQLERRM);  end procTestGradesOut;  /  begin  procTestGradesOut (1, 1);  end;  / | | | |

| 요구사항번호 | C-4(관리자) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 출결 현황 기간별로 구분 후 조회 | 작성자 | 정혜정 |
| DML | | | |
| -- 4.1. 기간별로, 출결 상황을 구분하여 조회한다.  -- 출결 현황을 기간별(전체, 월, 일) 조회할 수 있어야 한다.  -- 모든 출결 조회는 근태 상황을 구분할 수 있어야 한다.(정상, 지각, 조퇴, 외출, 병가, 기타)  -- 교사가 강의한 과정에 한해 선택하는 경우 모든 교육생의 출결을 조회할 수 있어야 한다.  create or replace procedure procTraineeAttendanceOut (  pSeq\_teacher in number,  pSeq\_openSubjectList in number,  pStartDate in date,  pEndDate in date  )  is  cursor vcursor is  select  distinct  vc.seq\_openCurriculum "개설 교육과정 번호",  asl.seq\_subject "과목 번호",  tl.seq\_trainee "교육생 번호",  a.day 날짜,  to\_char(a.inTime, 'HH24:MI:SS') "출근 시간",  to\_char(a.outTime, 'HH24:MI:SS') "퇴근 시간",  tas.situation "상태"  from vwCurriculum vc  inner join tblAvailableSubjectList asl  on asl.seq\_subject = vc.seq\_subject  inner join tblTraineeList tl  on tl.seq\_openCurriculum = vc.seq\_openCurriculum  inner join tblAttendance a  on a.seq\_traineeList = tl.seq\_traineeList  inner join tblAttendanceStatus tas  on tas.seq\_attendanceStatus = a.seq\_attendanceStatus  where a.inTime between pStartDate and pEndDate  and vc.seq\_teacher = pSeq\_teacher  and vc.seq\_openSubjectList = pSeq\_openSubjectList  order by tl.seq\_trainee;  vrecord vcursor%rowtype;    begin  -- 정보 출력  open vcursor;  loop  fetch vcursor into vrecord;  exit when vcursor%notfound;    -- 정보 출력  dbms\_output.put\_line('개설 교육과정 번호: ' || vrecord."개설 교육과정 번호");  dbms\_output.put\_line('과목 번호: ' || vrecord."과목 번호");  dbms\_output.put\_line('교육생 번호: ' || vrecord."교육생 번호");  dbms\_output.put\_line('날짜: ' || vrecord.날짜);  dbms\_output.put\_line('출근 시간: ' || vrecord."출근 시간");  dbms\_output.put\_line('퇴근 시간: ' || vrecord."퇴근 시간");  dbms\_output.put\_line('상태: ' || vrecord.상태);  dbms\_output.put\_line('-----------------------------------------------------------------------------------');  end loop;    close vcursor;  exception  when no\_data\_found then  dbms\_output.put\_line('해당 정보가 없습니다.');  when others then  dbms\_output.put\_line('오류 발생: ' || SQLERRM);  end procTraineeAttendanceOut;  /  begin  procTraineeAttendanceOut (1, 1, to\_date('2023-09-04', 'YYYY-MM-DD'), to\_date('2023-10-04', 'YYYY-MM-DD'));  end;  /  -- 4.1.2. 특정(특정 과정, 특정 인원) 출결 현황을 조회할 수 있어야 한다.  -- 특정 과정  create or replace procedure procSpecificCurriculumAttendanceOut (  pSeq\_teacher in number,  pSeq\_openSubjectList in number,  pSeq\_openCurriculum in number  )  is  cursor vcursor is  select  vc.seq\_openCurriculum "개설 교육과정 번호",  asl.seq\_subject "과목 번호",  tl.seq\_trainee "교육생 번호",  a.day 날짜,  to\_char(a.inTime, 'HH24:MI:SS') "출근 시간",  to\_char(a.outTime, 'HH24:MI:SS') "퇴근 시간"  from vwCurriculum vc  inner join tblAvailableSubjectList asl  on asl.seq\_subject = vc.seq\_subject  inner join tblTraineeList tl  on tl.seq\_openCurriculum = vc.seq\_openCurriculum  inner join tblAttendance a  on a.seq\_traineeList = tl.seq\_traineeList  where vc.seq\_openCurriculum = pSeq\_openCurriculum  and vc.seq\_teacher = pSeq\_teacher  and vc.seq\_openSubjectList = pSeq\_openSubjectList  order by tl.seq\_trainee;  vrecord vcursor%rowtype;    begin  -- 정보 출력  open vcursor;  loop  fetch vcursor into vrecord;  exit when vcursor%notfound;    -- 정보 출력  dbms\_output.put\_line('개설 교육과정 번호: ' || vrecord."개설 교육과정 번호");  dbms\_output.put\_line('과목 번호: ' || vrecord."과목 번호");  dbms\_output.put\_line('교육생 번호: ' || vrecord."교육생 번호");  dbms\_output.put\_line('날짜: ' || vrecord.날짜);  dbms\_output.put\_line('출근 시간: ' || vrecord."출근 시간");  dbms\_output.put\_line('퇴근 시간: ' || vrecord."퇴근 시간");  dbms\_output.put\_line('-----------------------------------------------------------------------------------');  end loop;    close vcursor;  exception  when no\_data\_found then  dbms\_output.put\_line('해당 정보가 없습니다.');  when others then  dbms\_output.put\_line('오류 발생: ' || SQLERRM);  end procSpecificCurriculumAttendanceOut;  /  begin  procSpecificCurriculumAttendanceOut (1, 1, 7);  end;  /  -- 특정 인원  create or replace procedure procSpecificTraineeAttendanceOut (  pSeq\_teacher in number,  pSeq\_openSubjectList in number,  pSeq\_trainee in number  )  is  cursor vcursor is  select  vc.seq\_openCurriculum "개설 교육과정 번호",  asl.seq\_subject "과목 번호",  tl.seq\_trainee "교육생 번호",  a.day 날짜,  to\_char(a.inTime, 'HH24:MI:SS') "출근 시간",  to\_char(a.outTime, 'HH24:MI:SS') "퇴근 시간"  from vwCurriculum vc  inner join tblAvailableSubjectList asl  on asl.seq\_subject = vc.seq\_subject  inner join tblTraineeList tl  on tl.seq\_openCurriculum = vc.seq\_openCurriculum  inner join tblAttendance a  on a.seq\_traineeList = tl.seq\_traineeList  where vc.seq\_openSubjectList = pSeq\_openSubjectList  and vc.seq\_teacher = pSeq\_teacher  and tl.seq\_trainee = pSeq\_trainee  order by tl.seq\_trainee;  vrecord vcursor%rowtype;    begin  -- 정보 출력  open vcursor;  loop  fetch vcursor into vrecord;  exit when vcursor%notfound;    -- 정보 출력  dbms\_output.put\_line('개설 교육과정 번호: ' || vrecord."개설 교육과정 번호");  dbms\_output.put\_line('과목 번호: ' || vrecord."과목 번호");  dbms\_output.put\_line('교육생 번호: ' || vrecord."교육생 번호");  dbms\_output.put\_line('날짜: ' || vrecord.날짜);  dbms\_output.put\_line('출근 시간: ' || vrecord."출근 시간");  dbms\_output.put\_line('퇴근 시간: ' || vrecord."퇴근 시간");  dbms\_output.put\_line('-----------------------------------------------------------------------------------');  end loop;    close vcursor;  exception  when no\_data\_found then  dbms\_output.put\_line('해당 정보가 없습니다.');  when others then  dbms\_output.put\_line('오류 발생: ' || SQLERRM);  end procSpecificTraineeAttendanceOut;  /  begin  procSpecificTraineeAttendanceOut (1, 1, 1);  end;  / | | | |

| 요구사항번호 | C-5(교사) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 강사가 자신의 이름 검색 하면 수료한 학생들이 작성한 평가 확인 | 작성자 | 김민곤 |
| DML | | | |
| create or replace procedure procTce(  ptname varchar2  )  is  vt\_name varchar2(10);  vce\_grade number;  vce\_content varchar2(10000);  CURSOR vcursor IS  select distinct  t.name as 강사명,  ce.grade as 점수,  ce.content as 내용  from tblCurriculumEvaluation ce  inner join tblTraineeList tl  on tl.seq\_traineeList = ce.seq\_traineeList  inner join tblOpenCurriculum oc  on oc.seq\_openCurriculum = tl.seq\_openCurriculum  inner join tblOpenSubjectList sl  on oc.seq\_openCurriculum = sl.seq\_openCurriculum  inner join tblTeacher t  on t.seq\_teacher = sl.seq\_teacher  where t.name = ptname;  begin  open vcursor;  loop  fetch vcursor into vt\_name, vce\_grade, vce\_content;  exit when vcursor%notfound;  DBMS\_OUTPUT.PUT\_LINE('점수: ' || vce\_grade || '| 내용: ' || vce\_content);  dbms\_output.put\_line('----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  end loop;  close vcursor;  end procTce;  /  begin  proctce('김민곤');  end;  / | | | |

| 요구사항번호 | D-1(교육생) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 강사명 검색시 과목별 정보와 배점,시험문제 조회 | 작성자 | 김민곤 |
| DML | | | |
| CREATE OR REPLACE PROCEDURE proctga(  ptname VARCHAR2  )  IS  vtc\_name VARCHAR2(15); -- 강사명  vg\_seq\_subject NUMBER; -- 과목 번호  vg\_s\_name VARCHAR2(100); -- 과목명  vb\_name VARCHAR2(100); -- 교재명  vosl\_startdate DATE; -- 과목 시작일  vosl\_enddate DATE; -- 과목 종료일  vg\_writtenDate DATE; -- 필기날짜  vg\_practicalDate DATE; -- 실기날짜  vg\_WRITTENPOINTS NUMBER; -- 필기배점  vg\_practicalPoints NUMBER; -- 실기배점  vg\_attendancePoints NUMBER; -- 출결배점  vq\_question VARCHAR2(5000); -- 시험 문제  CURSOR vcursor IS  SELECT DISTINCT  tc.name AS 강사명,  g.seq\_subject AS 과목번호,  g.s\_name AS 과목명,  b.name AS 교재명,  osl.startdate AS 과목시작일,  osl.enddate AS 과목종료일,  g.writtenPoints AS 필기배점,  g.practicalPoints AS 실기배점,  g.attendancePoints AS 출결배점,  g.writtenDate AS 필기날짜,  g.practicalDate AS 실기날짜,  q.question AS 문제  FROM  vwgrades g  INNER JOIN tbltraineelist tl ON g.seq\_traineelist = tl.seq\_traineelist  INNER JOIN tbltrainees t ON tl.seq\_trainee = t.seq\_trainee  INNER JOIN tblteacher tc ON g.seq\_teacher = tc.seq\_teacher  INNER JOIN tblopensubjectlist osl ON g.seq\_opensubjectlist = osl.seq\_opensubjectlist  INNER JOIN tbltextbook b ON osl.seq\_textbook = b.seq\_textbook  INNER JOIN tblopencurriculum oc ON g.seq\_opencurriculum = oc.seq\_opencurriculum  INNER JOIN tblroom r ON oc.seq\_room = r.seq\_room  INNER JOIN tblexampaper ep ON osl.seq\_openSubjectList = ep.seq\_openSubjectList  INNER JOIN tblquestion q ON ep.seq\_question = q.seq\_question  WHERE  tc.name = ptname  ORDER BY  g.seq\_subject asc;  BEGIN  OPEN vcursor;  LOOP  FETCH vcursor INTO vtc\_name, vg\_seq\_subject, vg\_s\_name, vb\_name, vosl\_startdate, vosl\_enddate, vg\_writtenPoints,  vg\_practicalPoints, vg\_attendancePoints, vg\_writtenDate, vg\_practicalDate, vq\_question;  EXIT WHEN vcursor%NOTFOUND;    DBMS\_OUTPUT.PUT\_LINE('강사명: ' || vtc\_name || ' | 과목번호: ' || vg\_seq\_subject || '| 과목명: ' || vg\_s\_name || '| 교재명: ' || vb\_name ||  '| 과목시작일: ' || vosl\_startdate || '| 과목종료일: ' || vosl\_enddate || '| 필기배점: ' || vg\_writtenPoints || '| 실기배점: ' || vg\_practicalPoints ||  '| 출결배점: ' || vg\_attendancePoints || '| 필기날짜:'|| vg\_writtenDate || '| 실기날짜:'|| vg\_practicalDate ||'| 시험문제:'|| vq\_question);  DBMS\_OUTPUT.PUT\_LINE('----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  END LOOP;  CLOSE vcursor;  END proctga;  /  begin  proctga('김민곤');  end;  /  --D-1.2교육생 이름으로 성적 검색 프로시저  CREATE OR REPLACE PROCEDURE proctg(  ptname VARCHAR2  )  IS  vt\_name VARCHAR2(500);  vs\_name VARCHAR2(500);  vosl\_startdate DATE;  vosl\_enddate DATE;  vg\_writtenDate DATE;  vg\_practicalDate DATE;  vg\_writtenGrade NUMBER;  vg\_practicalGrade NUMBER;  vg\_attendanceGrade NUMBER;  CURSOR vcursor IS  SELECT DISTINCT  t.name AS 교육생이름,  g.s\_name AS 과목명,  osl.startdate AS 과목시작일,  osl.enddate AS 과목종료일,  g.writtenDate AS 필기날짜,  g.practicalDate AS 실기날짜,  g.writtenGrade AS 필기점수,  g.practicalGrade AS 실기점수,  g.attendanceGrade AS 출결점수  FROM  vwgrades g  INNER JOIN tbltraineelist tl ON g.seq\_traineelist = tl.seq\_traineelist  INNER JOIN tbltrainees t ON tl.seq\_trainee = t.seq\_trainee  INNER JOIN tblteacher tc ON g.seq\_teacher = tc.seq\_teacher  INNER JOIN tblopensubjectlist osl ON g.seq\_opensubjectlist = osl.seq\_opensubjectlist  INNER JOIN tbltextbook b ON osl.seq\_textbook = b.seq\_textbook  INNER JOIN tblopencurriculum oc ON g.seq\_opencurriculum = oc.seq\_opencurriculum  INNER JOIN tblroom r ON oc.seq\_room = r.seq\_room  INNER JOIN tblexampaper ep ON osl.seq\_opensubjectlist = ep.seq\_opensubjectlist  INNER JOIN tblquestion q ON ep.seq\_question = q.seq\_question    WHERE  t.name = ptname;  BEGIN  OPEN vcursor;  LOOP  FETCH vcursor INTO vt\_name, vs\_name, vosl\_startdate, vosl\_enddate, vg\_writtenDate, vg\_practicalDate, vg\_writtenGrade, vg\_practicalGrade, vg\_attendanceGrade;  EXIT WHEN vcursor%NOTFOUND;  DBMS\_OUTPUT.PUT\_LINE('수강생이름: ' || vt\_name || ' | 과목명: ' || vs\_name || '| 과목시작일: ' || vosl\_startdate || '| 과목종료일: ' || vosl\_enddate ||  '| 필기날짜: ' || vg\_writtenDate || '| 실기날짜: ' || vg\_practicalDate || '| 필기점수: ' || vg\_writtenGrade || '| 실기점수: ' || vg\_practicalGrade ||  '| 출결점수: ' || vg\_attendanceGrade);  dbms\_output.put\_line('----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  END LOOP;  CLOSE vcursor;  END proctg;  /  begin  proctg('곤백안');  end;  /  -- D-1  -- 성적조회  CREATE OR REPLACE PROCEDURE procTraineeInfo(  p\_name IN vwtrainees.t\_name%TYPE  ) IS  CURSOR c\_info IS  SELECT DISTINCT vt.t\_name, vt.t\_id, vt.t\_ssn, vt.t\_tel, vc.s\_name, vc.osl\_startdate, vc.osl\_enddate  FROM vwtrainees vt  INNER JOIN vwgrades vg ON vt.seq\_opencurriculum = vg.seq\_opencurriculum  INNER JOIN vwCurriculum vc ON vc.seq\_opensubjectlist = vg.seq\_opensubjectlist  WHERE vt.t\_name = p\_name  ORDER BY vt.t\_name, vt.t\_id, vt.t\_ssn, vt.t\_tel, vc.s\_name, vc.osl\_startdate, vc.osl\_enddate;  v\_info c\_info%ROWTYPE;  BEGIN  OPEN c\_info;  LOOP  FETCH c\_info INTO v\_info;  EXIT WHEN c\_info%NOTFOUND;  DBMS\_OUTPUT.PUT\_LINE('교육생 이름: ' || v\_info.t\_name || ', 아이디: ' || v\_info.t\_id || ', 주민등록번호: ' || v\_info.t\_ssn || ', 전화번호: ' || v\_info.t\_tel || ', 과목 이름: ' || v\_info.s\_name || ', 시작 날짜: ' || v\_info.osl\_startdate || ', 종료 날짜: ' || v\_info.osl\_enddate);  END LOOP;  CLOSE c\_info;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('해당 이름을 가진 교육생의 정보를 찾을 수 없습니다.');  WHEN OTHERS THEN  ROLLBACK;  RAISE;  END procTraineeInfo;  /  BEGIN  procTraineeInfo('모백양');  END;  / | | | |

| 요구사항번호 | D-3(교육생) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 교육 과정 평가 (수료 학생만 가능) | 작성자 | 장원준 |
| DML | | | |
| CREATE OR REPLACE PROCEDURE procAddCurriculumEvaluation(  p\_seq\_curriculum IN tblCurriculumEvaluation.seq\_opencurriculum%TYPE,  p\_seq\_opencurriculum IN tblCurriculumEvaluation.seq\_opencurriculum%TYPE,  p\_seq\_trainee IN tblCurriculumEvaluation.seq\_traineelist%TYPE,  p\_score IN tblCurriculumEvaluation.grade%TYPE,  p\_comment IN tblCurriculumEvaluation.content%TYPE  ) IS  CURSOR c\_trainee IS  SELECT s.name, tl.status FROM tbltraineelist tl  INNER JOIN tbltrainees s ON s.seq\_trainee = tl.seq\_trainee;  v\_trainee c\_trainee%ROWTYPE;  BEGIN  OPEN c\_trainee;  LOOP  FETCH c\_trainee INTO v\_trainee;  EXIT WHEN c\_trainee%NOTFOUND;  IF v\_trainee.status = '수료' THEN  INSERT INTO tblCurriculumEvaluation VALUES (p\_seq\_curriculum, p\_seq\_opencurriculum, p\_seq\_trainee, p\_score, p\_comment);  DBMS\_OUTPUT.PUT\_LINE('교육생 이름: ' || v\_trainee.name || ', 상태: ' || v\_trainee.status || ', 교육 평가 정보가 추가되었습니다.');  END IF;  END LOOP;  CLOSE c\_trainee;  EXCEPTION  WHEN OTHERS THEN  ROLLBACK;  DBMS\_OUTPUT.PUT\_LINE('교육 평가 정보 추가에 실패하였습니다.');  RAISE;  END procAddCurriculumEvaluation;  /  BEGIN  procAddCurriculumEvaluation(1, 1, 1, 5, '설명을 자세하게 해주신다.');  END;  / | | | |

| 요구사항번호 | D-4(교육생) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 교사 추천 도서 조회 | 작성자 | 장원준 |
| DML | | | |
| -- 교사 추천 도서 조회  create or replace procedure procTextbookInfo(  pSeq\_trainee in varchar2  )  is  cursor vcursor is  select  distinct osl.seq\_opencurriculum 교육과정명,  tea.name 교사명,  rb.grade 별점,  b.name "책 제목",  b.publisher "출판사명"  from tblrecommendtextbook rb  inner join tblopensubjectlist osl  on rb.seq\_teacher = osl.seq\_teacher  inner join tbltraineelist tl  on tl.seq\_opencurriculum = osl.seq\_opencurriculum  inner join tblteacher tea  on tea.seq\_teacher = osl.seq\_teacher  inner join tbltextbook b  on b.seq\_textbook = rb.seq\_textbook  where osl.seq\_teacher = rb.seq\_teacher  and tl.seq\_trainee = pSeq\_trainee;  vrecord vcursor%rowtype;  begin  open vcursor;  loop  fetch vcursor into vrecord;  exit when vcursor%notfound;    dbms\_output.put\_line('교육과정명: ' || vrecord.교육과정명 || ', 교사명: ' || vrecord.교사명 || ', 별점: ' || vrecord.별점 || ', 책 제목: ' || vrecord."책 제목" || ', 출판사명: ' || vrecord."출판사명");  dbms\_output.put\_line('----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  end loop;  close vcursor;  exception  when no\_data\_found then  dbms\_output.put\_line('교과서 정보를 찾을 수 없습니다.');  when others then  rollback;  raise;  end procTextbookInfo;  /  begin  procTextbookInfo(1);  end;  / | | | |

| 요구사항번호 | C-4(교사) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 출결 현황 기간별로 구분 후 조회 | 작성자 | 정혜정 |
| DML | | | |
| create or replace procedure procTraineeAttendanceOut (  pSeq\_teacher in number,  pSeq\_openSubjectList in number,  pStartDate in date,  pEndDate in date  )  is  cursor vcursor is  select  vc.seq\_openCurriculum "개설 교육과정 번호",  asl.seq\_subject "과목 번호",  tl.seq\_trainee "교육생 번호",  a.day 날짜,  to\_char(a.inTime, 'HH24:MI:SS') "출근 시간",  to\_char(a.outTime, 'HH24:MI:SS') "퇴근 시간",  tas.situation "상태"  from vwCurriculum vc  inner join tblAvailableSubjectList asl  on asl.seq\_subject = vc.seq\_subject  inner join tblTraineeList tl  on tl.seq\_openCurriculum = vc.seq\_openCurriculum  inner join tblAttendance a  on a.seq\_traineeList = tl.seq\_traineeList  inner join tblAttendanceStatus tas  on tas.seq\_attendanceStatus = a.seq\_attendanceStatus  where a.inTime between pStartDate and pEndDate  and vc.seq\_teacher = pSeq\_teacher  and vc.seq\_openSubjectList = pSeq\_openSubjectList  order by tl.seq\_trainee;  vrecord vcursor%rowtype;    begin  -- 정보 출력  open vcursor;  loop  fetch vcursor into vrecord;  exit when vcursor%notfound;    -- 정보 출력  dbms\_output.put\_line('개설 교육과정 번호: ' || vrecord."개설 교육과정 번호");  dbms\_output.put\_line('과목 번호: ' || vrecord."과목 번호");  dbms\_output.put\_line('교육생 번호: ' || vrecord."교육생 번호");  dbms\_output.put\_line('날짜: ' || vrecord.날짜);  dbms\_output.put\_line('출근 시간: ' || vrecord."출근 시간");  dbms\_output.put\_line('퇴근 시간: ' || vrecord."퇴근 시간");  dbms\_output.put\_line('상태: ' || vrecord.상태);  dbms\_output.put\_line('-----------------------------------------------------------------------------------');  end loop;    close vcursor;  exception  when no\_data\_found then  dbms\_output.put\_line('해당 정보가 없습니다.');  when others then  dbms\_output.put\_line('오류 발생: ' || SQLERRM);  end procTraineeAttendanceOut;  /  begin  procTraineeAttendanceOut (1, 1, to\_date('2023-09-04', 'YYYY-MM-DD'), to\_date('2023-10-04', 'YYYY-MM-DD'));  end;  /  -- 4.1.2. 특정(특정 과정, 특정 인원) 출결 현황을 조회할 수 있어야 한다.  -- 특정 과정  create or replace procedure procSpecificCurriculumAttendanceOut (  pSeq\_teacher in number,  pSeq\_openSubjectList in number,  pSeq\_openCurriculum in number  )  is  cursor vcursor is  select  vc.seq\_openCurriculum "개설 교육과정 번호",  asl.seq\_subject "과목 번호",  tl.seq\_trainee "교육생 번호",  a.day 날짜,  to\_char(a.inTime, 'HH24:MI:SS') "출근 시간",  to\_char(a.outTime, 'HH24:MI:SS') "퇴근 시간"  from vwCurriculum vc  inner join tblAvailableSubjectList asl  on asl.seq\_subject = vc.seq\_subject  inner join tblTraineeList tl  on tl.seq\_openCurriculum = vc.seq\_openCurriculum  inner join tblAttendance a  on a.seq\_traineeList = tl.seq\_traineeList  where vc.seq\_openCurriculum = pSeq\_openCurriculum  and vc.seq\_teacher = pSeq\_teacher  and vc.seq\_openSubjectList = pSeq\_openSubjectList  order by tl.seq\_trainee;  vrecord vcursor%rowtype;    begin  -- 정보 출력  open vcursor;  loop  fetch vcursor into vrecord;  exit when vcursor%notfound;    -- 정보 출력  dbms\_output.put\_line('개설 교육과정 번호: ' || vrecord."개설 교육과정 번호");  dbms\_output.put\_line('과목 번호: ' || vrecord."과목 번호");  dbms\_output.put\_line('교육생 번호: ' || vrecord."교육생 번호");  dbms\_output.put\_line('날짜: ' || vrecord.날짜);  dbms\_output.put\_line('출근 시간: ' || vrecord."출근 시간");  dbms\_output.put\_line('퇴근 시간: ' || vrecord."퇴근 시간");  dbms\_output.put\_line('-----------------------------------------------------------------------------------');  end loop;    close vcursor;  exception  when no\_data\_found then  dbms\_output.put\_line('해당 정보가 없습니다.');  when others then  dbms\_output.put\_line('오류 발생: ' || SQLERRM);  end procSpecificCurriculumAttendanceOut;  /  begin  procSpecificCurriculumAttendanceOut (1, 1, 7);  end;  /  -- 특정 인원  create or replace procedure procSpecificTraineeAttendanceOut (  pSeq\_teacher in number,  pSeq\_openSubjectList in number,  pSeq\_trainee in number  )  is  cursor vcursor is  select  vc.seq\_openCurriculum "개설 교육과정 번호",  asl.seq\_subject "과목 번호",  tl.seq\_trainee "교육생 번호",  a.day 날짜,  to\_char(a.inTime, 'HH24:MI:SS') "출근 시간",  to\_char(a.outTime, 'HH24:MI:SS') "퇴근 시간"  from vwCurriculum vc  inner join tblAvailableSubjectList asl  on asl.seq\_subject = vc.seq\_subject  inner join tblTraineeList tl  on tl.seq\_openCurriculum = vc.seq\_openCurriculum  inner join tblAttendance a  on a.seq\_traineeList = tl.seq\_traineeList  where vc.seq\_openSubjectList = pSeq\_openSubjectList  and vc.seq\_teacher = pSeq\_teacher  and tl.seq\_trainee = pSeq\_trainee  order by tl.seq\_trainee;  vrecord vcursor%rowtype;    begin  -- 정보 출력  open vcursor;  loop  fetch vcursor into vrecord;  exit when vcursor%notfound;    -- 정보 출력  dbms\_output.put\_line('개설 교육과정 번호: ' || vrecord."개설 교육과정 번호");  dbms\_output.put\_line('과목 번호: ' || vrecord."과목 번호");  dbms\_output.put\_line('교육생 번호: ' || vrecord."교육생 번호");  dbms\_output.put\_line('날짜: ' || vrecord.날짜);  dbms\_output.put\_line('출근 시간: ' || vrecord."출근 시간");  dbms\_output.put\_line('퇴근 시간: ' || vrecord."퇴근 시간");  dbms\_output.put\_line('-----------------------------------------------------------------------------------');  end loop;    close vcursor;  exception  when no\_data\_found then  dbms\_output.put\_line('해당 정보가 없습니다.');  when others then  dbms\_output.put\_line('오류 발생: ' || SQLERRM);  end procSpecificTraineeAttendanceOut;  /  begin  procSpecificTraineeAttendanceOut (1, 1, 1);  end;  / | | | |

| 요구사항번호 | B-3 (관리자) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 강의진행여부 자동 변경 | 작성자 |  |
| PROCEDURE | | | |
| create or replace procedure procUpdateCurriProgress  is  vSeq number;  vStartDate DATE;  vEndDate DATE;    cursor vcursor is  select seq\_opencurriculum, startdate, enddate  from tblopencurriculum;  begin  open vcursor;  loop  fetch vcursor into vSeq, vStartdate, vEnddate;  exit when vcursor%notfound;  -- 개강일이 현재보다 미래일 때  if vstartDate > sysdate  then update tblopencurriculum set seq\_curriculumprogress = 1 where seq\_opencurriculum = vseq;    -- 현재가 개강일과 종료일 사이일 때  ELSIF vstartDate < sysdate and venddate > sysdate  then update tblopencurriculum set seq\_curriculumprogress = 2 where seq\_opencurriculum = vseq;    -- 현재가 종료일보다 미래일 때  ELSIF venddate < sysdate  then update tblopencurriculum set seq\_curriculumprogress = 3 where seq\_opencurriculum = vseq;  END IF;  end loop;  close vcursor;  end procUpdateCurriProgress;  /  begin  procUpdateCurriProgress();  end;  / | | | |

| 요구사항번호 | B-4 (관리자) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 강의 종료 시 교육생을 수료 상태로 변경하고 수료 날짜를 자동 업데이트 | 작성자 |  |
| PROCEDURE | | | |
| -- 테스트를 위한 null 만들기  update tbltraineelist  set status = null,  day = null  where seq\_traineelist between 1 and 20  and seq\_traineelist != 10;  rollback;  -- 과정 종료된 목록 불러오기  select tl.seq\_traineeList, oc.endDate, tl.status, tl.day  from tblTraineeList tl  left outer join tblOpenCurriculum oc  on tl.seq\_openCurriculum = oc.seq\_openCurriculum  where oc.endDate < sysdate;  -- 전체  select tl.seq\_traineeList, oc.seq\_openCurriculum, oc.endDate, tl.status, tl.day  from tblTraineeList tl  left outer join tblOpenCurriculum oc  on tl.seq\_openCurriculum = oc.seq\_openCurriculum;  create or replace procedure procUpdateCurriCompletionDate  is  vSeq\_traineeList tblTraineeList.seq\_traineeList%type;  vEndDate tblOpenCurriculum.endDate%type;  vStatus tblTraineeList.status%type;    cursor vcursor is  select tl.seq\_traineeList, oc.endDate, tl.status  from tblTraineeList tl  left outer join tblOpenCurriculum oc  on tl.seq\_openCurriculum = oc.seq\_openCurriculum  where oc.endDate < sysdate; --과정이 종료된 경우    begin  open vcursor;  loop  fetch vcursor into vSeq\_traineeList, vEndDate, vStatus;  exit when vcursor%notfound;    -- 날짜가 비어있는 경우  if vStatus is null then  update tblTraineeList  set status = '수료',  day = vEndDate  where seq\_traineeList = vSeq\_traineeList;  end if;    end loop;  close vcursor;  end procUpdateCurriCompletionDate;  /  --프로시저 실행  begin  procUpdateCurriCompletionDate;  end;  / | | | |

| 요구사항번호 | B-8 (관리자) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 출결 테이블에 결석생 자동 추가 | 작성자 |  |
| PROCEDURE | | | |
| create or replace procedure procUpdateAbsent  is  vHoliday date;  vtrue number;  vStart date;  vEnd date;  vStatus varchar2(30);  vSeq number;  vcnt number;  vcnt1 number;    cursor vcursor1 is  select holiday  from tblPublicHoliday;    cursor vcursor2 is  select oc.startdate, oc.enddate, tl.status, tl.seq\_traineelist  from tblOpenCurriculum oc  inner join tblTraineeList tl  on oc.seq\_opencurriculum = tl.seq\_opencurriculum;  begin  vtrue := 0;  open vcursor1;  loop  fetch vcursor1 into vHoliday;  exit when vcursor1%notfound;  -- 오늘이 휴일이나 주말이면 vtrue가 0 이상  if vholiday = sysdate or to\_char(sysdate, 'd') = '1' or to\_char(sysdate, 'd') = '7'  then vtrue := vtrue + 1;  end if;  end loop;  close vcursor1;    -- 오늘이 평일이면 vtrue = 0  if vtrue = 0 then  open vcursor2;  loop  fetch vcursor2 into vStart, vEnd, vStatus, vSeq;  exit when vcursor2%notfound;  select count(\*) into vcnt from tblAttendance  where to\_char(day, 'yyyy-mm-dd') = to\_char(sysdate, 'yyyy-mm-dd') and seq\_traineelist = vseq;  -- 오늘 아무기록이 없고 수강중일 때 결석 입력  if (sysdate between vstart and vend) and vcnt = 0 and vstatus is null  then insert into tblAttendance values ((select max(seq\_attendance) from tblAttendance) + 1, vseq, 7, to\_date(sysdate, 'yyyy-mm-dd'), null, null);  end if;  end loop;  close vcursor2;  end if;    end procUpdateAbsent;  /  begin  procUpdateAbsent();  end;  / | | | |

| 요구사항번호 | B-8 (관리자) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 출결 상황을 자동 업데이트 | 작성자 |  |
| PROCEDURE | | | |
| create or replace procedure procUpdateaAttendanceStatus  is  vSeq\_attendance tblAttendance.seq\_attendance%type;  vInTime tblAttendance.inTime%type;  vOutTime tblAttendance.outTime%type;  vSeq\_attendanceStatus tblAttendance.seq\_attendanceStatus%type;  vCurrentTime varchar2(10); -- 현재 시간을 담아둘 변수    cursor vcursor is  select  seq\_attendance,  inTime,  outTime,  seq\_attendanceStatus  from tblAttendance;      begin  -- 현재 시간의 시간 부분을 가져오기  vCurrentTime := to\_char(sysdate, 'HH24:MI');  open vcursor;    loop  fetch vcursor into vSeq\_attendance, vInTime, vOutTime, vSeq\_attendanceStatus;  exit when vcursor%notfound;    -- seq\_attendanceStatus가 null이면 값을 넣음  if vSeq\_attendanceStatus is null then  -- 9시 전에 오고, 17시 50분 이후 퇴실 한 경우  if to\_char(vInTime, 'HH24:MI') < '09:00' and to\_char(vOutTime, 'HH24:MI') >= '17:50' then  update tblAttendance  set seq\_attendanceStatus = 1 -- 정상  where seq\_attendance = vSeq\_attendance;  -- inTime이 null이고 현재 시간이 9시 이후인 경우 or 9시 이후에 온 경우  elsif vInTime is null and vCurrentTime >= '09:00' or to\_char(vInTime, 'HH24:MI') >= '09:00' then  update tblAttendance  set seq\_attendanceStatus = 2 -- 지각  where seq\_attendance = vSeq\_attendance;  -- outTime이 null이고 현재 시간이 18시 이후인 경우 or 그 전에 간 경우  elsif vOutTime is null and vCurrentTime >= '18:00' or to\_char(vOutTime, 'HH24:MI') < '17:50' then  update tblAttendance  set seq\_attendanceStatus = 3 -- 조퇴  where seq\_attendance = vSeq\_attendance;  end if;  end if;    end loop;  close vcursor;  end;  /  begin  procUpdateaAttendanceStatus;  end;  / | | | |

| 요구사항번호 | B-4 (관리자) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 과목 기간들이 과정 기간보다 작거나 같도록 제한하는 트리거 | 작성자 |  |
| TRIGGER | | | |
| CREATE OR REPLACE TRIGGER trgCheckPeriodSum  BEFORE INSERT OR UPDATE ON tblSubjectList  FOR EACH ROW  DECLARE  v\_seq\_curri number;  v\_period\_total NUMBER;  v\_period\_now NUMBER;  v\_period NUMBER;  BEGIN  v\_seq\_curri := :new.seq\_curriculum;  -- 해당 과정의 총 period를 가져옵니다.  SELECT period INTO v\_period\_total  from tblcurriculum c  inner join tblcourseperiod cp  on c.seq\_courseperiod = cp.seq\_courseperiod  where c.seq\_curriculum = v\_seq\_curri;    --현재 period 합을 가져옵니다.  SELECT SUM(period) into v\_period\_now  from tblSubjectList sl  inner join tblSubject s  on sl.seq\_subject = s.seq\_subject  where sl.seq\_curriculum = v\_seq\_curri;    --추가한 과목의 기간을 가져옵니다.  select period into v\_period from tblsubject where seq\_subject = :NEW.seq\_subject;    -- 새로운 코스 기간이나 업데이트된 코스 기간의 period 합이 tblsubject의 period 합과 같은지 확인합니다.  IF v\_period\_total < v\_period\_now+v\_period THEN  RAISE\_APPLICATION\_ERROR(-20001, '과정 총 기간을 확인해주세요.');  END IF;  END;  / | | | |

| 요구사항번호 | C-3 (교사) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 필기, 실기, 출석 최대 점수를 제한하는 트리거 | 작성자 |  |
| TRIGGER | | | |
| CREATE OR REPLACE TRIGGER trgCheckPoints  BEFORE INSERT ON tblTestInfo  FOR EACH ROW  DECLARE  v\_total\_points NUMBER;  BEGIN  -- writtenpoints, practicalpoints, attendancepoints의 합을 계산합니다.  v\_total\_points := :NEW.writtenpoints + :NEW.practicalpoints + :NEW.attendancepoints;  IF :NEW.writtenpoints >= 40 THEN  RAISE\_APPLICATION\_ERROR(-20002, '최대 40까지 입력 가능합니다.');  END IF;  IF :NEW.practicalpoints >= 40 THEN  RAISE\_APPLICATION\_ERROR(-20003, '최대 40까지 입력 가능합니다.');  END IF;  IF :NEW.attendancepoints >= 30 THEN  RAISE\_APPLICATION\_ERROR(-20004, '최대 30까지 입력 가능합니다.');  END IF;  -- writtenpoints, practicalpoints, attendancepoints의 합이 100을 초과하는 경우 오류를 발생시킵니다.  IF v\_total\_points > 100 THEN  RAISE\_APPLICATION\_ERROR(-20005, '총점이 100을 초과할 수 없습니다.');  END IF;  END;  / | | | |

| 요구사항번호 | C-2 (교사) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 성적이 100점이 넘지 않게 제한하는 트리거 | 작성자 |  |
| TRIGGER | | | |
| CREATE OR REPLACE TRIGGER trgTraineeCheckPoints  BEFORE INSERT ON tblgrades  FOR EACH ROW  DECLARE  v\_total\_grade NUMBER;  v\_seq\_testinfo number;  v\_writtenpoints number;  v\_practicalpoints number;  v\_attendancepoints number;  BEGIN  -- writtengrade, practicalgrade, attendancegrade의 합을 계산합니다.  v\_total\_grade := :NEW.writtengrade + :NEW.practicalgrade + :NEW.attendancegrade;  v\_seq\_testinfo := :new.seq\_testinfo;    select writtenpoints, practicalpoints, v\_attendancepoints into v\_writtenpoints, v\_practicalpoints, v\_attendancepoints  from tbltestinfo where seq\_testinfo = v\_seq\_testinfo;  IF :NEW.writtengrade > v\_writtenpoints THEN  RAISE\_APPLICATION\_ERROR(-20006, '최대 배점까지만 입력 가능합니다.');  END IF;    IF :NEW.practicalgrade > v\_practicalpoints THEN  RAISE\_APPLICATION\_ERROR(-20007, '최대 배점까지만 입력 가능합니다.');  END IF;    IF :NEW.attendancegrade > v\_attendancepoints THEN  RAISE\_APPLICATION\_ERROR(-20008, '최대 배점까지만 입력 가능합니다.');  END IF;  -- writtengrade, practicalgrade, attendancegrade의 합이 100을 초과하는 경우 오류를 발생시킵니다.  IF v\_total\_grade > 100 THEN  RAISE\_APPLICATION\_ERROR(-20009, '총점이 100을 초과할 수 없습니다.');  END IF;  END;  / | | | |

| 요구사항번호 | D-3 (교육생) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생이 수료했을 때만 평가를 할 수 있게 제한하는 트리거 | 작성자 |  |
| TRIGGER | | | |
| CREATE OR REPLACE TRIGGER trgAllowEvaluation  BEFORE INSERT ON tblcurriculumevaluation  FOR EACH ROW  DECLARE  v\_status VARCHAR2(20);  BEGIN  -- tbltraineelist에서 해당 trainee의 status를 가져옵니다.  SELECT status INTO v\_status  FROM tbltraineelist  WHERE seq\_traineelist = :NEW.seq\_traineelist;  -- status가 '수료'인 경우에만 데이터를 삽입할 수 있습니다.  IF v\_status != '수료' or v\_status is null THEN  RAISE\_APPLICATION\_ERROR(-20010, '수료 상태일 때만 평가를 입력할 수 있습니다.');  END IF;  END;  -- 5번  -- 성적은 종료 후 입력이 되니까 그 전에 입력 하지 못하도록 하는 트리거  CREATE OR REPLACE TRIGGER trgCheckEnddate  BEFORE INSERT ON tblGrades  FOR EACH ROW  DECLARE  v\_enddate DATE;  BEGIN  -- tblOpensubjectList에서 해당 과목의 종료일(enddate)을 가져옵니다.  SELECT osl.enddate INTO v\_enddate  from tblTestInfo ti  inner join tblOpenSubjectList osl  on ti.seq\_opensubjectlist = osl.seq\_opensubjectlist  where ti.seq\_testinfo = :NEW.seq\_testInfo;    -- 종료일(enddate) 이전에 성적 입력 시도시 차단  IF v\_enddate > SYSDATE THEN  RAISE\_APPLICATION\_ERROR(-20011, '과목 종료일 이후에만 시험 정보를 삽입할 수 있습니다.');  END IF;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  RAISE\_APPLICATION\_ERROR(-20012, '해당 과목 ID가 tblOpensubjectList에 존재하지 않습니다.');  END;  / | | | |

| 요구사항번호 | C-2 (교사) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 과목 종료일을 가져오고 종료일 이전에 성적 입력 시 제한하는 트리거 | 작성자 |  |
| TRIGGER | | | |
| CREATE OR REPLACE TRIGGER trgCheckEnddate  BEFORE INSERT ON tblGrades  FOR EACH ROW  DECLARE  v\_enddate DATE;  BEGIN  -- tblOpensubjectList에서 해당 과목의 종료일(enddate)을 가져옵니다.  SELECT osl.enddate INTO v\_enddate  from tblTestInfo ti  inner join tblOpenSubjectList osl  on ti.seq\_opensubjectlist = osl.seq\_opensubjectlist  where ti.seq\_testinfo = :NEW.seq\_testInfo;    -- 종료일(enddate) 이전에 성적 입력 시도시 차단  IF v\_enddate > SYSDATE THEN  RAISE\_APPLICATION\_ERROR(-20011, '과목 종료일 이후에만 시험 정보를 삽입할 수 있습니다.');  END IF;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  RAISE\_APPLICATION\_ERROR(-20012, '해당 과목 ID가 tblOpensubjectList에 존재하지 않습니다.');  END;  / | | | |

| 요구사항번호 | C-3 (교사) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 중도 탈락한 학생에 대한 성적 입력을 제한하는 트리거 | 작성자 |  |
| TRIGGER | | | |
| CREATE OR REPLACE TRIGGER trgCheckDropout  BEFORE INSERT OR UPDATE ON tblGrades  FOR EACH ROW  DECLARE  v\_status VARCHAR2(20);  v\_day DATE;  v\_writtendate date;  v\_practicaldate date;  BEGIN  -- tbltraineelist에서 해당 trainee의 status와 day 값을 가져옵니다.  SELECT status, day INTO v\_status, v\_day  FROM tbltraineelist  WHERE seq\_traineelist = :new.seq\_traineelist;    --writtendate와 practicaldate 값을 가져와서 비교합니다.  select writtendate, practicaldate into v\_writtendate, v\_practicaldate from tblTestInfo where seq\_testinfo = :new.seq\_testinfo;    -- status가 '중도탈락'이고, 시험 날짜가 중도탈락 날짜 이후면 성적을 삽입 불가  IF v\_status = '중도탈락' and (v\_writtendate > v\_day OR v\_practicaldate > v\_day)  THEN RAISE\_APPLICATION\_ERROR(-20013, '중도탈락 상태일 때는 해당 날짜의 성적을 삽입할 수 없습니다.');  END IF;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  RAISE\_APPLICATION\_ERROR(-20014, '해당 trainee\_id가 tbltraineelist에 존재하지 않습니다.');  END;  / | | | |

| 요구사항번호 | B-13 (관리자) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 출석 인정 상태로 변했을 때 출결 상태의 번호가 1로 변하는 트리거 | 작성자 | 정원혁 |
| TRIGGER | | | |
| create or replace trigger trgUpdateAttendance  after update of admitattendance on tblAttendancePapers  for each row  begin  if :new.admitattendance = '출석 인정' then  update tblAttendance a  set a.seq\_attendanceStatus = 1  where a.seq\_traineeList = :new.seq\_traineeList and a.day = :new.day;  end if;  end;  / | | | |

| 요구사항번호 | B-13 (관리자) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 교육과정 기간 수료 중에 면접 스케줄을 잡을 수 없게하는 트리거 | 작성자 | 장원준 |
| TRIGGER | | | |
| CREATE OR REPLACE TRIGGER trgInterviewScheduleInsert  BEFORE INSERT ON tblinterviewschedule  FOR EACH ROW  DECLARE  v\_curriculum\_progress\_seq NUMBER;  v\_trainee\_exists NUMBER;  BEGIN  -- Check if seq\_trainee exists in tbltraineelist table  SELECT COUNT(\*) INTO v\_trainee\_exists  FROM tbltraineelist  WHERE seq\_trainee = :NEW.seq\_trainee;    -- If seq\_trainee does not exist in tbltraineelist table, allow insert  IF v\_trainee\_exists = 0 THEN  RETURN;6    END IF;  -- Check if seq\_curriculumprogress is not null in tblopencurriculum table  SELECT seq\_curriculumprogress  INTO v\_curriculum\_progress\_seq  FROM tblopencurriculum  WHERE seq\_curriculumprogress = (SELECT seq\_opencurriculum FROM tbltraineelist WHERE seq\_trainee = :NEW.seq\_trainee);    IF v\_curriculum\_progress\_seq IS NOT NULL THEN  -- Raise an exception to prevent the insert  RAISE\_APPLICATION\_ERROR(-20015, 'seq\_curriculumprogress 컬럼이 NULL이 아닌 경우에는 tblinterviewschedule 테이블에 insert 할 수 없습니다.');  END IF;  END;  /  / | | | |

| 요구사항번호 | B-13 (관리자) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 합격자만 교육생 목록으로 insert 할 수 있게 제한하는 트리거 | 작성자 | 장원준 |
| TRIGGER | | | |
| CREATE OR REPLACE TRIGGER trgPerInsertTraineelist  BEFORE INSERT ON tbltraineelist  FOR EACH ROW  DECLARE  v\_status tblinterviewResults.Status%TYPE;  v\_seq\_schedule tblinterviewschedule.seq\_schedule%TYPE;  BEGIN  -- tblinterviewschedule 테이블에서 해당 seq\_trainee의 seq\_schedule 값을 가져옵니다.  SELECT seq\_schedule INTO v\_seq\_schedule  FROM tblinterviewschedule  WHERE seq\_trainee = :NEW.seq\_trainee;  -- 가져온 seq\_schedule 값이 tblinterviewResults 테이블에 있는지 확인합니다.  SELECT Status INTO v\_status  FROM tblinterviewResults  WHERE seq\_schedule = v\_seq\_schedule;  -- 가져온 seq\_schedule 값이 없거나, 가져온 Status 값이 '불합격'인 경우에만 삽입을 막습니다.  IF v\_status = '불합격' OR v\_seq\_schedule IS NULL THEN  RAISE\_APPLICATION\_ERROR(-20016, '불합격 또는 아직 면접을 보지 않았습니다.');  END IF;  END;  / | | | |

| 요구사항번호 | B-13 (관리자) | 작성일 | 2023-03-08 |
| --- | --- | --- | --- |
| 요구사항명 | 수료자만 취업 현황 테이블에 insert 할수 있게하는 트리거 | 작성자 | 장원준 |
| TRIGGER | | | |
| CREATE OR REPLACE TRIGGER trgEmploymentStatus  BEFORE INSERT ON tblEmploymentStatus  FOR EACH ROW  DECLARE  v\_status VARCHAR2(20);  BEGIN  -- tbltraineelist에서 해당 trainee의 status를 가져옵니다.  SELECT status INTO v\_status  FROM tbltraineelist  WHERE seq\_traineelist = :NEW.seq\_traineelist;  -- status가 '수료'인 경우에만 데이터를 삽입할 수 있습니다.  IF v\_status != '수료' or v\_status is null THEN    RAISE\_APPLICATION\_ERROR(-20017, '수료 상태일 때만 평가를 입력할 수 있습니다.');  END IF;  END;  / | | | |