**오브젝트 스크립트(PLSQL)**

**교사**

2조

한가람, 김원중, 이승진, 이지온, 이지원

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-1. 강의 스케줄 조회 | 작성일 | 2025-02-05 |
| 요구사항명 | 강의 스케줄 조회 | 작성자 | 한가람 |
| DML | | | |
| --1. 강의 예정  create or replace procedure procGetPrearrangedClass(  pTeacherId in number  )  is  cursor vcursor  is  select os.openedsubjectid as 개설과목번호  , c.coursename as 과정명  , to\_char(oc.openedstartdate, 'yyyymmdd') || ' ~ ' || to\_char(oc.openedcompletiondate, 'yyyymmdd') as 과정기간  , crm.classroomname as 강의실명  , s.subjectname as 과목명  , to\_char(os.subjectStartDate, 'yyyymmdd') || ' ~ ' || to\_char(os.subjectEndDate, 'yyyymmdd') as 과목기간  , tb.textbookname as 교재명  , ( select count(\*) from tblstudent where studentCourseId = oc.openedCourseId) as 교육생수  from tblOpenedCourse oc --개설과정  inner join tblCourse c --강의과정  on oc.courseid = c.courseid  inner join tblOpenedSubject os --개설과목  on os.openedcourseid = oc.openedcourseid  inner join tblSubject s --과목  on s.subjectid = os.subjectid  inner join tblClassroom crm --강의실  on crm.classroomid = oc.classroomid  inner join tbltextbook tb --교재  on tb.subjectId = s.subjectId  where oc.teacherId = pTeacherId --교사  and sysdate < os.subjectStartDate -- 과목기간  order by oc.openedcourseid, os.openedsubjectid  ;  vopenedsubjectid tblOpenedSubject.openedsubjectid%type;  vcoursename tblCourse.coursename%type;  vopenedcourseperiod varchar(30);  vclassroomname tblClassroom.classroomname%type;  vsubjectname tblSubject.subjectname%type;  vopenedsubjectperiod varchar(30);  vtextbookname tbltextbook.textbookname%type;  vstudentcnt number;  begin  open vcursor;  dbms\_output.put\_line('======================================================================================================================================================================================================================');  dbms\_output.put\_line('|| 개설과목번호 || ' || '과정명 || ' || '과정기간 || ' || '강의실명 || ' || '과목명 || ' || '과목기간 || ' || '교재명 || ' || '수강인원 ||');  dbms\_output.put\_line('======================================================================================================================================================================================================================');  loop  fetch vcursor into vopenedsubjectid, vcoursename, vopenedcourseperiod, vclassroomname, vsubjectname, vopenedsubjectperiod, vtextbookname, vstudentcnt;  dbms\_output.put\_line('|| ' || vopenedsubjectid || ' || ' || vcoursename || ' || ' || vopenedcourseperiod || ' || ' || vclassroomname || ' || ' || vsubjectname || ' || ' || vopenedsubjectperiod || ' || ' || vtextbookname || ' || ' || vstudentcnt || ' || ');  exit when vcursor%notfound;  end loop;  dbms\_output.put\_line('======================================================================================================================================================================================================================');  close vcursor;  end;  /  --2. 강의 중  create or replace procedure procGetOnClass(  pTeacherId in number  )  is  cursor vcursor  is  select os.openedsubjectid as 개설과목번호  , c.coursename as 과정명  , to\_char(oc.openedstartdate, 'yyyymmdd') || ' ~ ' || to\_char(oc.openedcompletiondate, 'yyyymmdd') as 과정기간  , crm.classroomname as 강의실명  , s.subjectname as 과목명  , to\_char(os.subjectStartDate, 'yyyymmdd') || ' ~ ' || to\_char(os.subjectEndDate, 'yyyymmdd') as 과목기간  , tb.textbookname as 교재명  , ( select count(\*) from tblstudent where studentCourseId = oc.openedCourseId) as 교육생수  from tblOpenedCourse oc --개설과정  inner join tblCourse c --강의과정  on oc.courseid = c.courseid  inner join tblOpenedSubject os --개설과목  on os.openedcourseid = oc.openedcourseid  inner join tblSubject s --과목  on s.subjectid = os.subjectid  inner join tblClassroom crm --강의실  on crm.classroomid = oc.classroomid  inner join tbltextbook tb --교재  on tb.subjectId = s.subjectId  where oc.teacherId = pTeacherId --교사  and sysdate between oc.openedstartdate and oc.openedcompletiondate --과정기간  and sysdate between os.subjectStartDate and os.subjectEndDate -- 과목기간  order by oc.openedcourseid, os.openedsubjectid  ;  vopenedsubjectid tblOpenedSubject.openedsubjectid%type;  vcoursename tblCourse.coursename%type;  vopenedcourseperiod varchar(30);  vclassroomname tblClassroom.classroomname%type;  vsubjectname tblSubject.subjectname%type;  vopenedsubjectperiod varchar(30);  vtextbookname tbltextbook.textbookname%type;  vstudentcnt number;  begin  open vcursor;  dbms\_output.put\_line('=================================================================================================');  dbms\_output.put\_line('|| 개설과목번호 || 과정명 || 과정기간 || 강의실명 || 과목명 || 과목기간 || 교재명 || 수강인원 || ');  dbms\_output.put\_line('=================================================================================================');  loop  fetch vcursor into vopenedsubjectid, vcoursename, vopenedcourseperiod, vclassroomname, vsubjectname, vopenedsubjectperiod, vtextbookname, vstudentcnt;  dbms\_output.put\_line('|| ' || vopenedsubjectid || ' || ' || vcoursename || ' || ' || vopenedcourseperiod || ' || ' || vclassroomname || ' || ' || vsubjectname || ' || ' || vopenedsubjectperiod || ' || ' || vtextbookname || ' || ' || vstudentcnt || ' || ');  exit when vcursor%notfound;  end loop;  dbms\_output.put\_line('=================================================================================================');  close vcursor;  end;  /  --3. 강의 종료  create or replace procedure procGetEndClass(  pTeacherId in number  )  is  cursor vcursor  is  select os.openedsubjectid as 개설과목번호  , c.coursename as 과정명  , to\_char(oc.openedstartdate, 'yyyymmdd') || ' ~ ' || to\_char(oc.openedcompletiondate, 'yyyymmdd') as 과정기간  , crm.classroomname as 강의실명  , s.subjectname as 과목명  , to\_char(os.subjectStartDate, 'yyyymmdd') || ' ~ ' || to\_char(os.subjectEndDate, 'yyyymmdd') as 과목기간  , tb.textbookname as 교재명  , ( select count(\*) from tblstudent where studentCourseId = oc.openedCourseId) as 교육생수  from tblOpenedCourse oc --개설과정  inner join tblCourse c --강의과정  on oc.courseid = c.courseid  inner join tblOpenedSubject os --개설과목  on os.openedcourseid = oc.openedcourseid  inner join tblSubject s --과목  on s.subjectid = os.subjectid  inner join tblClassroom crm --강의실  on crm.classroomid = oc.classroomid  inner join tbltextbook tb --교재  on tb.subjectId = s.subjectId  where oc.teacherId = pTeacherId --교사  and sysdate > os.subjectEndDate -- 과목기간  order by oc.openedcourseid, os.openedsubjectid  ;  vopenedsubjectid tblOpenedSubject.openedsubjectid%type;  vcoursename tblCourse.coursename%type;  vopenedcourseperiod varchar(30);  vclassroomname tblClassroom.classroomname%type;  vsubjectname tblSubject.subjectname%type;  vopenedsubjectperiod varchar(30);  vtextbookname tbltextbook.textbookname%type;  vstudentcnt number;  begin  open vcursor;  dbms\_output.put\_line('==================================================================================================');  dbms\_output.put\_line(' || 개설과목번호 || 과정명 || 과정기간 || 강의실명 || 과목명 || 과목기간 || 교재명 || 수강인원 || ');  dbms\_output.put\_line('==================================================================================================');  loop  fetch vcursor into vopenedsubjectid, vcoursename, vopenedcourseperiod, vclassroomname, vsubjectname, vopenedsubjectperiod, vtextbookname, vstudentcnt;  dbms\_output.put\_line(' || ' || vopenedsubjectid || ' || ' || vcoursename || ' || ' || vopenedcourseperiod || ' || ' || vclassroomname || ' || ' || vsubjectname || ' || ' || vopenedsubjectperiod || ' || ' || vtextbookname || ' || ' || vstudentcnt || ' ||');  exit when vcursor%notfound;  end loop;  dbms\_output.put\_line('==================================================================================================');  close vcursor;  end;  /  --4. 교육생 정보 조회  create or replace procedure procGetStudent(  pTeacherId in number  )  is  cursor vcursor  is  select s.studentName as 교육생이름  , s.studentPhoneNumber as 전화번호  , to\_char(s.studentRegistrationDate, 'yyyy-mm-dd') as 등록일  , studentCompletionStatus as 수강상태  from tblStudent s  where s.studentCourseId in (  select openedCourseId  from tblOpenedCourse oc --개설과정  where teacherId = pTeacherId --교사ID  and sysdate between oc.openedstartdate and oc.openedcompletiondate --현재 수강 중인 대상  )  ;  vstudentName tblStudent.studentName%type;  vstudentPhoneNumber tblStudent.studentPhoneNumber%type;  vstudentRegistrationDate varchar2(20);  vstudentCompletionStatus tblStudent.studentCompletionStatus%type;  begin  open vcursor;  dbms\_output.put\_line('==================================================' );  dbms\_output.put\_line('|| 교육생이름 || 전화번호 || 등록일 || 수강상태 ||' );  dbms\_output.put\_line('==================================================' );  loop  fetch vcursor into vstudentName, vstudentPhoneNumber, vstudentRegistrationDate, vstudentCompletionStatus;  dbms\_output.put\_line('|| ' || vstudentName ||' || ' || vstudentPhoneNumber || ' || ' || vstudentRegistrationDate || ' || ' || vstudentCompletionStatus|| ' ||' );  exit when vcursor%notfound;  end loop;  dbms\_output.put\_line('==================================================' );  close vcursor;  end;  / | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-2.배점입출력 | 작성일 | 2025-02-05 |
| 요구사항명 | 배점입출력 | 작성자 | 한가람 |
| DML | | | |
| --1. 과목 목록 출력  create or replace procedure procPrintSubjectListforScoreWeight(  pTeacherId in number  )  is  cursor vcursor  is  select os.openedsubjectid as 개설과목번호  , c.coursename as 과정명  , to\_char(oc.openedstartdate, 'yyyymmdd') || ' ~ ' || to\_char(oc.openedcompletiondate, 'yyyymmdd') as 과정기간  , crm.classroomname as 강의실명  , s.subjectname as 과목명  , to\_char(os.subjectStartDate, 'yyyymmdd') || ' ~ ' || to\_char(os.subjectEndDate, 'yyyymmdd') as 과목기간  , tb.textbookname as 교재명  , os.subjectattendancescoreweight as 출결배점  , os.subjectwrittenscoreweight as 필기배점  , os.subjectpracticalscoreweight as 실기배점  from tblOpenedCourse oc --개설과정  inner join tblCourse c --강의과정  on oc.courseid = c.courseid  inner join tblOpenedSubject os --개설과목  on os.openedcourseid = oc.openedcourseid  inner join tblSubject s --과목  on s.subjectid = os.subjectid  inner join tblClassroom crm --강의실  on crm.classroomid = oc.classroomid  inner join tbltextbook tb --교재  on tb.subjectId = s.subjectId  where oc.teacherId = pTeacherId --교사  and sysdate between oc.openedstartdate and oc.openedcompletiondate --과정기간  and sysdate > os.subjectEndDate -- 과목기간 , 교사는 자신이 강의를 마친 과목의 목록  order by oc.openedcourseid, os.openedsubjectid  ;  vopenedsubjectid tblOpenedSubject.openedsubjectid%type;  vcoursename tblCourse.coursename%type;  vopenedcourseperiod varchar2(20);  vclassroomname tblClassroom.classroomname%type;  vsubjectname tblSubject.subjectname%type;  vopenedsubjectperiod varchar2(20);  vtextbookname tbltextbook.textbookname%type;  vsubjectattendancescoreweight tblOpenedSubject.subjectattendancescoreweight%type;  vsubjectwrittenscoreweight tblOpenedSubject.subjectwrittenscoreweight%type;  vsubjectpracticalscoreweight tblOpenedSubject.subjectpracticalscoreweight%type;  begin  open vcursor;  dbms\_output.put\_line('=========================================================================================================================' );  dbms\_output.put\_line('|| 개설과목번호 || 과정명 || 과정기간 || 강의실명 || 과목명 || 과목기간 || 교재명 || 출결배점 || 필기배점 || 실기배점 ||' );  dbms\_output.put\_line('=========================================================================================================================' );  loop  fetch vcursor into vopenedsubjectid, vcoursename, vopenedcourseperiod, vclassroomname, vsubjectname, vopenedsubjectperiod, vtextbookname, vsubjectattendancescoreweight, vsubjectwrittenscoreweight, vsubjectpracticalscoreweight;  dbms\_output.put\_line('|| ' || vopenedsubjectid || ' || ' || vcoursename || ' || '|| vopenedcourseperiod || ' || ' || vclassroomname || ' || ' || vsubjectname || ' || ' || vopenedsubjectperiod || ' || ' || vtextbookname || ' || ' || vsubjectattendancescoreweight || ' || ' || vsubjectwrittenscoreweight || ' || ' || vsubjectpracticalscoreweight );  exit when vcursor%notfound;  end loop;  dbms\_output.put\_line('=========================================================================================================================' );  close vcursor;  end;  /  --2. 출결 배점 등록 및 수정  create or replace procedure procInputAttendanceScoreWeight(  popenedsubjectid in number, --개설과목번호  psubjectattendancescoreweight in number --출결배점  )  is  vsubjectAttendanceScoreWeight number; --출결배점  leak\_score\_weight exception;  begin  if psubjectattendancescoreweight < 20 then  raise leak\_score\_weight;  end if;  update tblOpenedSubject set subjectattendancescoreweight = psubjectattendancescoreweight where openedsubjectid = popenedsubjectid;  commit;  select subjectAttendanceScoreWeight into vsubjectAttendanceScoreWeight  from tblOpenedSubject  where openedsubjectid = popenedsubjectid;  dbms\_output.put\_line('개설과목\_ID : ' || popenedsubjectid );  dbms\_output.put\_line('출결배점 : ' || vsubjectAttendanceScoreWeight );  exception  when leak\_score\_weight then  dbms\_output.put\_line('최소 20점 이상의 배점을 입력해주세요.' );  rollback;  end;  /  --3. 출결 배점 삭제  create or replace procedure procDeleteAttendanceScoreWeight(  popenedsubjectid in number --개설과목번호  )  is  vsubjectAttendanceScoreWeight number; --출결배점  begin  update tblOpenedSubject set subjectattendancescoreweight = null where openedsubjectid = popenedsubjectid;  commit;  select subjectAttendanceScoreWeight into vsubjectAttendanceScoreWeight  from tblOpenedSubject  where openedsubjectid = popenedsubjectid;  dbms\_output.put\_line('개설과목\_ID : ' || popenedsubjectid );  dbms\_output.put\_line('출결배점 : ' || vsubjectAttendanceScoreWeight );  end;  /  --4. 출결 배점 조회  create or replace procedure procGetAttendanceScoreWeight(  popenedsubjectid in number  )  is  cursor vcursor  is  select openedsubjectid as 개설과목번호  , subjectattendancescoreweight as 출결배점  from tblOpenedSubject  where openedsubjectid = popenedsubjectid  ;  vopenedsubjectid tblOpenedSubject.openedsubjectid%type;  vsubjectattendancescoreweight tblOpenedSubject.subjectattendancescoreweight%type;  begin  open vcursor;  loop  fetch vcursor into vopenedsubjectid, vsubjectattendancescoreweight;  exit when vcursor%notfound;  dbms\_output.put\_line('개설과목번호 : ' || vopenedsubjectid );  dbms\_output.put\_line('출결배점 : ' || vsubjectattendancescoreweight );  end loop;  close vcursor;  end;  /  --5. 필기 배점 등록 및 수정  create or replace procedure procInputWrittenScoreWeight(  popenedsubjectid in number, --개설과목번호  psubjectwrittenscoreweight in number --필기배점  )  is  vsubjectwrittenscoreweight number; --필기배점  begin  update tblOpenedSubject set subjectwrittenscoreweight = psubjectwrittenscoreweight where openedsubjectid = popenedsubjectid;  commit;  select subjectwrittenscoreweight into vsubjectwrittenscoreweight  from tblOpenedSubject  where openedsubjectid = popenedsubjectid;  dbms\_output.put\_line('개설과목\_ID : ' || popenedsubjectid );  dbms\_output.put\_line('필기배점 : ' || vsubjectwrittenscoreweight );  end;  /  --6. 필기 배점 삭제  create or replace procedure procDeletwrittenScoreWeight(  popenedsubjectid in number --개설과목번호  )  is  vsubjectwrittenscoreweight number; --필기배점  begin  update tblOpenedSubject set subjectwrittenscoreweight = null where openedsubjectid = popenedsubjectid;  commit;  select subjectwrittenscoreweight into vsubjectwrittenscoreweight  from tblOpenedSubject  where openedsubjectid = popenedsubjectid;  dbms\_output.put\_line('개설과목\_ID : ' || popenedsubjectid );  dbms\_output.put\_line('출결배점 : ' || vsubjectwrittenscoreweight );  end;  /  --7. 필기 배점 조회  create or replace procedure procGetwrittenScoreWeight(  popenedsubjectid in number  )  is  cursor vcursor  is  select openedsubjectid as 개설과목번호  , subjectwrittenscoreweight as 필기배점  from tblOpenedSubject  where openedsubjectid = popenedsubjectid  ;  vopenedsubjectid tblOpenedSubject.openedsubjectid%type;  vsubjectwrittenscoreweight tblOpenedSubject.subjectwrittenscoreweight%type;  begin  open vcursor;  loop  fetch vcursor into vopenedsubjectid, vsubjectwrittenscoreweight;  exit when vcursor%notfound;  dbms\_output.put\_line('개설과목번호 : ' || vopenedsubjectid );  dbms\_output.put\_line('필기배점 : ' || vsubjectwrittenscoreweight );  end loop;  close vcursor;  end;  /  --8. 실기 배점 등록 및 수정  create or replace procedure procInputpracticalScoreWeight(  popenedsubjectid in number, --개설과목번호  psubjectpracticalscoreweight in number --실기배점  )  is  vsubjectpracticalscoreweight number; --실기배점  begin  update tblOpenedSubject set subjectpracticalscoreweight = psubjectpracticalscoreweight where openedsubjectid = popenedsubjectid;  commit;  select subjectpracticalscoreweight into vsubjectpracticalscoreweight  from tblOpenedSubject  where openedsubjectid = popenedsubjectid;  dbms\_output.put\_line('개설과목\_ID : ' || popenedsubjectid );  dbms\_output.put\_line('실기배점 : ' || vsubjectpracticalscoreweight );  end;  /  --9. 실기 배점 삭제  create or replace procedure procDeletpracticalScoreWeight(  popenedsubjectid in number --개설과목번호  )  is  vsubjectpracticalscoreweight number; --실기배점  begin  update tblOpenedSubject set subjectpracticalscoreweight = null where openedsubjectid = popenedsubjectid;  commit;  select subjectpracticalscoreweight into vsubjectpracticalscoreweight  from tblOpenedSubject  where openedsubjectid = popenedsubjectid;  dbms\_output.put\_line('개설과목\_ID : ' || popenedsubjectid );  dbms\_output.put\_line('실기배점 : ' || vsubjectpracticalscoreweight );  end;  /  --10. 실기 배점 조회  select openedsubjectid as 개설과목번호  , subjectpracticalscoreweight as 실기배점  from tblOpenedSubject  where openedsubjectid = 38;  create or replace procedure procGetpracticalScoreWeight(  popenedsubjectid in number  )  is  cursor vcursor  is  select openedsubjectid as 개설과목번호  , subjectpracticalscoreweight as 실기배점  from tblOpenedSubject  where openedsubjectid = popenedsubjectid  ;  vopenedsubjectid tblOpenedSubject.openedsubjectid%type;  vsubjectpracticalscoreweight tblOpenedSubject.subjectpracticalscoreweight%type;  begin  open vcursor;  loop  fetch vcursor into vopenedsubjectid, vsubjectpracticalscoreweight;  exit when vcursor%notfound;  dbms\_output.put\_line('개설과목번호 : ' || vopenedsubjectid );  dbms\_output.put\_line('실기배점 : ' || vsubjectpracticalscoreweight );  end loop;  close vcursor;  end;  /  --11. 모든 배점 등록 및 수정  create or replace procedure procInputAllScoreWeight(  popenedsubjectid in number, --개설과목번호  psubjectattendancescoreweight in number, --출결배점  psubjectwrittenscoreweight in number, --필기배점  psubjectpracticalscoreweight in number --실기배점  )  is  vsubjectAttendanceScoreWeight number;  vsubjectwrittenscoreweight number;  vsubjectpracticalscoreweight number;  leak\_score\_weight exception;  invalid\_total\_score\_weight exception;  begin  if psubjectattendancescoreweight < 20 then --출결 배점  raise leak\_score\_weight;  end if;  if psubjectattendancescoreweight + psubjectwrittenscoreweight + psubjectpracticalscoreweight != 100 then  raise invalid\_total\_score\_weight;  end if;  update tblOpenedSubject  set subjectattendancescoreweight = psubjectattendancescoreweight  , subjectwrittenscoreweight = psubjectwrittenscoreweight  , subjectpracticalscoreweight = psubjectpracticalscoreweight  where openedsubjectid = popenedsubjectid;  commit;  select subjectAttendanceScoreWeight, subjectwrittenscoreweight, subjectpracticalscoreweight into vsubjectAttendanceScoreWeight, vsubjectwrittenscoreweight, vsubjectpracticalscoreweight  from tblOpenedSubject  where openedsubjectid = popenedsubjectid;  dbms\_output.put\_line('개설과목\_ID : ' || popenedsubjectid );  dbms\_output.put\_line('출결배점 : ' || vsubjectAttendanceScoreWeight );  dbms\_output.put\_line('필기배점 : ' || vsubjectwrittenscoreweight );  dbms\_output.put\_line('실기배점 : ' || vsubjectpracticalscoreweight );  exception  when leak\_score\_weight then  dbms\_output.put\_line('최소 20점 이상의 배점을 입력해주세요.' );  when invalid\_total\_score\_weight then  dbms\_output.put\_line('모든 배점의 합이 100이 되도록 입력해주세요.' );  end;  /  --11. 필기 시험 문제 등록  create or replace procedure procinputWittenExam(  popenedsubjectid in number, --개설과목번호  pwrittenExamDate in date, --시험일  pwrittenExamQuestions in varchar2 --문제  )  is  vwrittenExamId tblWrittenExam.writtenExamId%type;  vopenedSubjectId tblWrittenExam.openedSubjectId%type;  vwrittenExamDate tblWrittenExam.writtenExamDate%type;  vwrittenExamQuestions tblWrittenExam.writtenExamQuestions%type;  begin  INSERT INTO tblWrittenExam(writtenExamId, openedSubjectId, writtenExamDate, writtenExamQuestions)  VALUES ( (select max(writtenExamId) + 1 from tblWrittenExam) ,popenedsubjectid , pwrittenExamDate, pwrittenExamQuestions);  commit;  select max(writtenExamId) into vwrittenExamId  from tblWrittenExam;  select writtenExamId, openedSubjectId, writtenExamDate, writtenExamQuestions into vwrittenExamId, vopenedSubjectId, vwrittenExamDate, vwrittenExamQuestions  from tblWrittenExam  where writtenExamId = vwrittenExamId;  dbms\_output.put\_line('필기시험\_ID : ' || vwrittenExamId );  dbms\_output.put\_line('개설과목\_ID : ' || vopenedSubjectId );  dbms\_output.put\_line('시험날짜 : ' || vwrittenExamDate );  dbms\_output.put\_line('시험문제 : ' || vwrittenExamQuestions );  end;  /  --12. 필기 시험 문제 삭제  create or replace procedure procDeleteWittenExam(  pwrittenExamId number  )  is  vwrittenExamId tblWrittenExam.writtenExamId%type;  vopenedSubjectId tblWrittenExam.openedSubjectId%type;  vwrittenExamDate tblWrittenExam.writtenExamDate%type;  vwrittenExamQuestions tblWrittenExam.writtenExamQuestions%type;  begin  delete from tblWrittenExam  where writtenExamId = pwrittenExamId;  commit;  dbms\_output.put\_line('문제가 삭제되었습니다.' );  end;  /  --13. 필기 시험 문제 수정  create or replace procedure procUpdateWittenExam(  pwrittenExamId in number,  pwrittenExamDate in date,  pwrittenExamQuestions in varchar2  )  is  vwrittenExamId tblWrittenExam.writtenExamId%type;  vopenedSubjectId tblWrittenExam.openedSubjectId%type;  vwrittenExamDate tblWrittenExam.writtenExamDate%type;  vwrittenExamQuestions tblWrittenExam.writtenExamQuestions%type;  begin  update tblWrittenExam  set writtenExamDate = pwrittenExamDate  , writtenExamQuestions = pwrittenExamQuestions  where writtenExamId = pwrittenExamId;  commit;  select writtenExamId, openedSubjectId, writtenExamDate, writtenExamQuestions into vwrittenExamId, vopenedSubjectId, vwrittenExamDate, vwrittenExamQuestions  from tblWrittenExam  where writtenExamId = pwrittenExamId;  dbms\_output.put\_line('문제가 수정되었습니다.' );  dbms\_output.put\_line('필기시험\_ID : ' || vwrittenExamId );  dbms\_output.put\_line('개설과목\_ID : ' || vopenedSubjectId );  dbms\_output.put\_line('시험날짜 : ' || vwrittenExamDate );  dbms\_output.put\_line('시험문제 : ' || vwrittenExamQuestions );  end;  /  --14. 필기 시험 문제 조회  create or replace procedure procGetWittenExam(  popenedSubjectId in number  )  is  vwrittenExamId tblWrittenExam.writtenExamId%type;  vopenedSubjectId tblWrittenExam.openedSubjectId%type;  vwrittenExamDate tblWrittenExam.writtenExamDate%type;  vwrittenExamQuestions tblWrittenExam.writtenExamQuestions%type;  cursor vcursor  is  select writtenExamId as 필기시험\_ID  , openedSubjectId as 개설과목\_ID  , writtenExamDate as 시험날짜  , writtenExamQuestions as 시험문제  from tblWrittenExam  where openedSubjectId = popenedSubjectId;  begin  open vcursor;  loop  fetch vcursor into vwrittenExamId, vopenedSubjectId, vwrittenExamDate, vwrittenExamQuestions;  exit when vcursor%notfound;  dbms\_output.put\_line('필기시험\_ID : ' || vwrittenExamId );  dbms\_output.put\_line('개설과목\_ID : ' || vopenedSubjectId );  dbms\_output.put\_line('시험날짜 : ' || vwrittenExamDate );  dbms\_output.put\_line('시험문제 : ' || vwrittenExamQuestions );  end loop;  close vcursor;  end;  /  --15. 실기 시험 문제 등록  create or replace procedure procinputPracticalExam(  popenedsubjectid in number, --개설과목번호  ppracticalExamDate in date, --시험일  ppracticalExamQuestions in varchar2 --문제  )  is  vpracticalExamId tblPracticalExam.practicalExamId%type;  vopenedSubjectId tblPracticalExam.openedSubjectId%type;  vpracticalExamDate tblPracticalExam.practicalExamDate%type;  vpracticalExamQuestions tblPracticalExam.practicalExamQuestions%type;  begin  INSERT INTO tblPracticalExam(practicalExamId, openedSubjectId, practicalExamDate, practicalExamQuestions)  VALUES ( (select max(practicalExamId) + 1 from tblPracticalExam), popenedsubjectid , ppracticalExamDate, ppracticalExamQuestions);  commit;  select max(practicalExamId) into vpracticalExamId  from tblPracticalExam;  select practicalExamId, openedSubjectId, practicalExamDate, practicalExamQuestions into vpracticalExamId, vopenedSubjectId, vpracticalExamDate, vpracticalExamQuestions  from tblPracticalExam  where practicalExamId = vpracticalExamId;  dbms\_output.put\_line('실기시험\_ID : ' || vpracticalExamId );  dbms\_output.put\_line('개설과목\_ID : ' || vopenedSubjectId );  dbms\_output.put\_line('시험날짜 : ' || vpracticalExamDate );  dbms\_output.put\_line('시험문제 : ' || vpracticalExamQuestions );  end;  /  --16. 실기 시험 문제 삭제  create or replace procedure procDeletePracticalExam(  ppracticalExamId number  )  is  vpracticalExamId tblPracticalExam.practicalExamId%type;  vopenedSubjectId tblPracticalExam.openedSubjectId%type;  vpracticalExamDate tblPracticalExam.practicalExamDate%type;  vpracticalExamQuestions tblPracticalExam.practicalExamQuestions%type;  begin  delete from tblPracticalExam  where practicalExamId = ppracticalExamId;  commit;  dbms\_output.put\_line('문제가 삭제되었습니다.' );  end;  /  --17. 실기 시험 문제 수정  create or replace procedure procUpdatePracticalExam(  pPracticalExamId in number,  pPracticalExamDate in date,  pPracticalExamQuestions in varchar2  )  is  vpracticalExamId tblPracticalExam.practicalExamId%type;  vopenedSubjectId tblPracticalExam.openedSubjectId%type;  vpracticalExamDate tblPracticalExam.practicalExamDate%type;  vpracticalExamQuestions tblPracticalExam.practicalExamQuestions%type;  begin  update tblPracticalExam  set practicalExamDate = pPracticalExamDate  , practicalExamQuestions = pPracticalExamQuestions  where practicalExamId = pPracticalExamId;  commit;  select practicalExamId, openedSubjectId, practicalExamDate, practicalExamQuestions into vpracticalExamId, vopenedSubjectId, vpracticalExamDate, vpracticalExamQuestions  from tblPracticalExam  where practicalExamId = pPracticalExamId;  dbms\_output.put\_line('실기시험\_ID : ' || vpracticalExamId );  dbms\_output.put\_line('개설과목\_ID : ' || vopenedSubjectId );  dbms\_output.put\_line('시험날짜 : ' || vpracticalExamDate );  dbms\_output.put\_line('시험문제 : ' || vpracticalExamQuestions );  end;  /  --18. 실기 시험 문제 조회  create or replace procedure procGetPracticalExam(  popenedSubjectId in number  )  is  vpracticalExamId tblPracticalExam.practicalExamId%type;  vopenedSubjectId tblPracticalExam.openedSubjectId%type;  vpracticalExamDate tblPracticalExam.practicalExamDate%type;  vpracticalExamQuestions tblPracticalExam.practicalExamQuestions%type;  cursor vcursor  is  select practicalExamId as 실기시험\_ID  , openedSubjectId as 개설과목\_ID  , practicalExamDate as 시험날짜  , practicalExamQuestions as 시험문제  from tblPracticalExam  where openedSubjectId = popenedSubjectId;  begin  open vcursor;  loop  fetch vcursor into vpracticalExamId, vopenedSubjectId, vpracticalExamDate, vpracticalExamQuestions;  exit when vcursor%notfound;  dbms\_output.put\_line('실기시험\_ID : ' || vpracticalExamId );  dbms\_output.put\_line('개설과목\_ID : ' || vopenedSubjectId );  dbms\_output.put\_line('시험날짜 : ' || vpracticalExamDate );  dbms\_output.put\_line('시험문제 : ' || vpracticalExamQuestions );  end loop;  close vcursor;  end;  / | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-3.성적입출력 | 작성일 | 2025-02-05 |
| 요구사항명 | 성적입출력 | 작성자 | 한가람 |
| DML | | | |
| --1. 과목 목록 출력  create or replace procedure procPrintSubjectListforManageScore(  pTeacherId in number  )  is  cursor vcursor  is  select os.openedsubjectid as 개설과목번호  , c.coursename as 과정명  , to\_char(oc.openedstartdate, 'yyyymmdd') as 과정기간\_start  , to\_char(oc.openedcompletiondate, 'yyyymmdd') as 과정기간\_end  , crm.classroomname as 강의실명  , s.subjectname as 과목명  , to\_char(os.subjectStartDate, 'yyyymmdd') as 과목기간\_start  , to\_char(os.subjectEndDate, 'yyyymmdd') as 과목기간\_end  , tb.textbookname as 교재명  , os.subjectattendancescoreweight as 출결배점  , os.subjectwrittenscoreweight as 필기배점  , os.subjectpracticalscoreweight as 실기배점  , case when os.openedsubjectid in (  --성적 등록이 안된 과목  select distinct openedsubjectId  from (  --개인 성적 정보  select g.gradeId  , os.openedsubjectId  , s.studentId  , s.studentCourseId  , s.studentName  , s.studentCompletionStatus  , gradeWrittenScore  , gradePracticalScore  , gradeAttendanceScore  from tblStudent s  inner join tblOpenedCourse oc  on s.studentCourseId = oc.openedCourseId  inner join tblOpenedSubject os  on os.openedCourseId = oc.openedCourseId  left outer join tblGrade g  on g.openedsubjectId = os.openedsubjectId  and g.studentId = s.studentId  ) x  where ( x.gradeWrittenScore is null  or x.gradePracticalScore is null  or x.gradeAttendanceScore is null )  and x.studentCompletionStatus != '중도포기'  ) then 'X'  else 'O'  end as 성적등록여부  from tblOpenedCourse oc --개설과정  inner join tblCourse c --강의과정  on oc.courseid = c.courseid  inner join tblOpenedSubject os --개설과목  on os.openedcourseid = oc.openedcourseid  inner join tblSubject s --과목  on s.subjectid = os.subjectid  inner join tblClassroom crm --강의실  on crm.classroomid = oc.classroomid  inner join tbltextbook tb --교재  on tb.subjectId = s.subjectId  where oc.teacherId = pTeacherId --교사  and sysdate between oc.openedstartdate and oc.openedcompletiondate --과정기간  and sysdate > os.subjectEndDate -- 과목기간 , 교사는 자신이 강의를 마친 과목의 목록  order by oc.openedcourseid, os.openedsubjectid  ;  v\_openedsubjectid tblOpenedSubject.openedsubjectid%type;  v\_coursename tblCourse.coursename%type;  v\_openedstartdate tblOpenedCourse.openedstartdate%type;  v\_openedcompletiondate tblOpenedCourse.openedcompletiondate%type;  v\_classroomname tblClassroom.classroomname%type;  v\_subjectname tblSubject.subjectname%type;  v\_subjectStartDate tblOpenedSubject.subjectStartDate%type;  v\_subjectEndDate tblOpenedSubject.subjectEndDate%type;  v\_textbookname tbltextbook.textbookname%type;  v\_subjectattendancescoreweight tblOpenedSubject.subjectattendancescoreweight%type;  v\_subjectwrittenscoreweight tblOpenedSubject.subjectwrittenscoreweight%type;  v\_subjectpracticalscoreweight tblOpenedSubject.subjectpracticalscoreweight%type;  v\_openedsubjectYn varchar(2);  begin  open vcursor;  dbms\_output.put\_line('========================================================================================================================' );  dbms\_output.put\_line('|| 개설과목번호 || 과정명 || 과정기간 || 강의실명 || 과목명 || 과목기간 || 교재명 || 출결배점 || 필기배점 || 실기배점 ||' );  dbms\_output.put\_line('========================================================================================================================' );  loop  fetch vcursor into v\_openedsubjectid, v\_coursename, v\_openedstartdate, v\_openedcompletiondate, v\_classroomname, v\_subjectname  , v\_subjectStartDate, v\_subjectEndDate, v\_textbookname, v\_subjectattendancescoreweight, v\_subjectwrittenscoreweight  , v\_subjectpracticalscoreweight, v\_openedsubjectYn;  exit when vcursor%notfound;  dbms\_output.put\_line(v\_openedsubjectid || ' || ' || v\_coursename || ' || '|| to\_char(v\_openedstartdate, 'yyyymmdd') || ' ~ ' || to\_char(v\_openedcompletiondate , 'yyyymmdd')  || ' || ' || v\_classroomname || ' || ' || v\_subjectname || ' || ' || to\_char(v\_subjectStartDate, 'yyyymmdd') || ' ~ ' || to\_char(v\_subjectEndDate , 'yyyymmdd')  || ' || ' || v\_textbookname || ' || ' || v\_subjectattendancescoreweight || ' || ' || v\_subjectwrittenscoreweight  || ' || ' || v\_subjectpracticalscoreweight || ' || ' || v\_openedsubjectYn );  end loop;  dbms\_output.put\_line('========================================================================================================================' );  close vcursor;  end;  /  --2. 교육생 정보 출력  create or replace procedure procPrintStudentInfo(  popenedsubjectId in number  )  is  cursor vcursor  is  select s.studentId  , s.studentName as 교육생이름  , s.studentPhoneNumber as 전화번호  , s.studentRegistrationDate as 등록일  , s.studentCompletionStatus as 수강상태  , g.gradeWrittenScore as 필기점수  , g.gradePracticalScore as 실기점수  , g.gradeAttendanceScore as 출결점수  from tblStudent s  inner join tblOpenedCourse oc  on s.studentCourseId = oc.openedCourseId  inner join tblOpenedSubject os  on os.openedCourseId = oc.openedCourseId  left outer join tblGrade g  on g.openedsubjectId = os.openedsubjectId  and g.studentId = s.studentId  where os.openedsubjectId = 38  ;  v\_studentId tblStudent.studentId%type;  v\_studentName tblStudent.studentName%type;  v\_studentPhoneNumber tblStudent.studentPhoneNumber%type;  v\_studentRegistrationDate tblStudent.studentRegistrationDate%type;  v\_studentCompletionStatus tblStudent.studentCompletionStatus%type;  v\_gradeWrittenScore tblGrade.gradeWrittenScore%type;  v\_gradePracticalScore tblGrade.gradePracticalScore%type;  v\_gradeAttendanceScore tblGrade.gradeAttendanceScore%type;  begin  open vcursor;  dbms\_output.put\_line('===================================================================================================' );  dbms\_output.put\_line('|| 교육생\_ID || 교육생이름 || 전화번호 || 등록일 || 수강상태 || 필기점수 || 실기점수 || 출결점수 ||' );  dbms\_output.put\_line('===================================================================================================' );  loop  fetch vcursor into v\_studentId, v\_studentName, v\_studentPhoneNumber, v\_studentRegistrationDate, v\_studentCompletionStatus, v\_gradeWrittenScore, v\_gradePracticalScore, v\_gradeAttendanceScore;  exit when vcursor%notfound;  dbms\_output.put\_line(v\_studentId || ' || ' || v\_studentName || ' || ' || v\_studentPhoneNumber || ' || ' || v\_studentRegistrationDate || ' || ' || v\_studentCompletionStatus || ' || ' || v\_gradeWrittenScore || ' || ' || v\_gradePracticalScore || ' || ' || v\_gradeAttendanceScore );  end loop;  dbms\_output.put\_line('===================================================================================================' );  close vcursor;  end;  /  --3. 필기점수 입력  create or replace procedure procInputgradeWrittenScore(  popenedsubjectId in number,  pstudentId in number,  pGradeWrittenScore in number  )  is  vCnt number := 0;  v\_gradeId tblGrade.gradeId%type;  v\_subjectName tblSubject.subjectName%type;  v\_studentName tblStudent.studentName%type;  v\_gradeWrittenScore tblGrade.gradeWrittenScore%type;  already\_input exception;  begin  select count(\*) into vCnt  from tblGrade  where openedsubjectId = popenedsubjectId  and studentId = pstudentId;  if vCnt > 0 then  raise already\_input;  end if;  INSERT INTO tblGrade(gradeId, openedsubjectId, studentId, gradeWrittenScore)  VALUES (( select max(gradeId) + 1 from tblGrade ), popenedsubjectId, pstudentId, pGradeWrittenScore);  commit;  select max(gradeId) into v\_gradeId  from tblGrade  ;  select g.gradeId, s.subjectName, st.studentName, gradeWrittenScore into v\_gradeId, v\_subjectName, v\_studentName, v\_gradeWrittenScore  from tblGrade g  inner join tblopenedSubject os  on os.openedsubjectId = g.openedsubjectId  inner join tblStudent st  on st.studentId = g.studentId  inner join tblSubject s  on s.subjectId = os.subjectId  where gradeId = v\_gradeId;  dbms\_output.put\_line('=================================================');  dbms\_output.put\_line('|| 성적\_ID || 과목명 || 교육생이름 || 필기점수 ||');  dbms\_output.put\_line('=================================================');  dbms\_output.put\_line(v\_gradeId || ' || ' || v\_subjectName || ' || ' || v\_studentName || ' || ' || v\_gradeWrittenScore);  dbms\_output.put\_line('=================================================');  exception  when already\_input then  dbms\_output.put\_line('이미 성적이 등록되었습니다.');  when others then  dbms\_output.put\_line('잘못된 데이터가 입력되었습니다.');  rollback;  end;  /  --4. 실기점수 입력  create or replace procedure procInputgradePracticalScore(  popenedsubjectId in number,  pstudentId in number,  pGradePracticalScore in number  )  is  vCnt number := 0;  vIsNull number;  v\_gradeId tblGrade.gradeId%type;  v\_subjectName tblSubject.subjectName%type;  v\_studentName tblStudent.studentName%type;  v\_gradePracticalScore tblGrade.gradePracticalScore%type;  not\_input\_witten\_score exception;  already\_input exception;  begin  select count(\*) into vCnt  from tblGrade  where openedsubjectId = popenedsubjectId  and studentId = pstudentId;  if vCnt < 1 then  raise not\_input\_witten\_score;  else  select case when gradePracticalScore is null then 0 --성적이 없으면 0  else 1  end isNullYn  into vIsNull  from tblGrade  where openedsubjectId = popenedsubjectId  and studentId = pstudentId;  end if;  if vIsNull = 1 then  raise already\_input;  end if;  update tblGrade set gradePracticalScore = pGradePracticalScore  where openedsubjectId = popenedsubjectId --개설과목명  and studentId = pstudentId --교육생번호  ;  commit;  select max(gradeId) into v\_gradeId  from tblGrade  ;  select g.gradeId, s.subjectName, st.studentName, gradePracticalScore into v\_gradeId, v\_subjectName, v\_studentName, v\_gradePracticalScore  from tblGrade g  inner join tblopenedSubject os  on os.openedsubjectId = g.openedsubjectId  inner join tblStudent st  on st.studentId = g.studentId  inner join tblSubject s  on s.subjectId = os.subjectId  where gradeId = v\_gradeId;  dbms\_output.put\_line('=================================================');  dbms\_output.put\_line('|| 성적\_ID || 과목명 || 교육생이름 || 실기점수 ||');  dbms\_output.put\_line('=================================================');  dbms\_output.put\_line(v\_gradeId || ' || ' || v\_subjectName || ' || ' || v\_studentName || ' || ' || v\_gradePracticalScore);  dbms\_output.put\_line('=================================================');  exception  when not\_input\_witten\_score then  dbms\_output.put\_line('필기점수를 입력하지 않았습니다. 먼저 필기점수를 입력해주세요.');  when already\_input then  dbms\_output.put\_line('이미 성적이 등록되었습니다.');  when others then  dbms\_output.put\_line('잘못된 데이터가 입력되었습니다.');  rollback;  end;  / | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-4.출결조회 | 작성일 | 2025-02-05 |
| 요구사항명 | 출결조회 | 작성자 | 한가람 |
| DML | | | |
| --1. 기간별(년) 출결 조회  create or replace procedure procGetAttendanceByYear(  pteacherId in number,  pYYYY in number  )  is  cursor vcursor  is  --기간별(년) 출결 조회  select c.courseName as 과정명  , s.studentName as 교육생이름  , to\_char(at.attendancedate, 'yyyy-mm-dd') as 출석일  , at.attendancestatus  from tblAttendance at  inner join tblStudent s  on at.studentId = s.studentId  inner join tblOpenedCourse oc  on oc.openedCourseId = s.studentCourseId  inner join tblCourse c  on c.courseId = oc.courseId  where oc.teacherId = pteacherId --교사 자신의 ID  and at.attendancedate between to\_date( pYYYY ||'0101000000','yyyymmddhh24miss') and to\_date(pYYYY || '1231235959','yyyymmddhh24miss') --기간  order by c.courseName, to\_char(at.attendancedate, 'yyyy-mm-dd'), s.studentName ;  v\_courseName tblCourse.courseName%type;  v\_studentName tblStudent.studentName%type;  v\_attendancedate varchar2(10);  v\_attendancestatus tblAttendance.attendancestatus%type;  begin  open vcursor;  dbms\_output.put\_line('================================================');  dbms\_output.put\_line('|| 과정명 || 교육생이름 || 출석일 || 출결상태 ||');  dbms\_output.put\_line('================================================');  loop  fetch vcursor into v\_courseName, v\_studentName, v\_attendancedate, v\_attendancestatus;  exit when vcursor%notfound;  dbms\_output.put\_line(v\_courseName || ' || ' || v\_studentName || ' || ' || v\_attendancedate || ' || ' || v\_attendancestatus);  end loop;  dbms\_output.put\_line('================================================');  close vcursor;  exception  when others then  dbms\_output.put\_line('정확한 값을 입력해주세요.');  end;  /  --2. 기간별(월) 출결 조회  create or replace procedure procGetAttendanceByMonth(  pteacherId in number,  pYYYYMM in number  )  is  cursor vcursor  is  --기간별(월) 출결 조회  select c.courseName as 과정명  , s.studentName as 교육생이름  , to\_char(at.attendancedate, 'yyyy-mm-dd') as 출석일  , at.attendancestatus  from tblAttendance at  inner join tblStudent s  on at.studentId = s.studentId  inner join tblOpenedCourse oc  on oc.openedCourseId = s.studentCourseId  inner join tblCourse c  on c.courseId = oc.courseId  where oc.teacherId = pteacherId --교사 자신의 ID  and at.attendancedate between to\_date(pYYYYMM || '01', 'yyyymmdd') and add\_months(to\_date(pYYYYMM || '01', 'yyyymmdd'), 1) - 1 + 1/24\*23 + 1/24/60\*59 + 1/24/60/60\*59 --기간  order by c.courseName, to\_char(at.attendancedate, 'yyyy-mm-dd'), s.studentName ;  v\_courseName tblCourse.courseName%type;  v\_studentName tblStudent.studentName%type;  v\_attendancedate varchar2(10);  v\_attendancestatus tblAttendance.attendancestatus%type;  begin  open vcursor;  dbms\_output.put\_line('================================================');  dbms\_output.put\_line('|| 과정명 || 교육생이름 || 출석일 || 출결상태 ||');  dbms\_output.put\_line('================================================');  loop  fetch vcursor into v\_courseName, v\_studentName, v\_attendancedate, v\_attendancestatus;  exit when vcursor%notfound;  dbms\_output.put\_line(v\_courseName || ' || ' || v\_studentName || ' || ' || v\_attendancedate || ' || ' || v\_attendancestatus);  end loop;  close vcursor;  dbms\_output.put\_line('================================================');  exception  when others then  dbms\_output.put\_line('정확한 값을 입력해주세요.');  end;  /  --3. 기간별(일) 출결 조회  create or replace procedure procGetAttendanceByDate(  pteacherId in number,  pYYYYMMDD in number  )  is  cursor vcursor  is  --기간별(일) 출결 조회  select c.courseName as 과정명  , s.studentName as 교육생이름  , to\_char(at.attendancedate, 'yyyy-mm-dd') as 출석일  , at.attendancestatus  from tblAttendance at  inner join tblStudent s  on at.studentId = s.studentId  inner join tblOpenedCourse oc  on oc.openedCourseId = s.studentCourseId  inner join tblCourse c  on c.courseId = oc.courseId  where oc.teacherId = pteacherId --교사 자신의 ID  and at.attendancedate between to\_date(pYYYYMMDD || '000000', 'yyyymmddHH24miss') and to\_date(pYYYYMMDD || '235959', 'yyyymmddHH24miss') --기간  order by c.courseName, to\_char(at.attendancedate, 'yyyy-mm-dd'), s.studentName ;  v\_courseName tblCourse.courseName%type;  v\_studentName tblStudent.studentName%type;  v\_attendancedate varchar2(10);  v\_attendancestatus tblAttendance.attendancestatus%type;  begin  open vcursor;  dbms\_output.put\_line('================================================');  dbms\_output.put\_line('|| 과정명 || 교육생이름 || 출석일 || 출결상태 ||');  dbms\_output.put\_line('================================================');  loop  fetch vcursor into v\_courseName, v\_studentName, v\_attendancedate, v\_attendancestatus;  exit when vcursor%notfound;  dbms\_output.put\_line(v\_courseName || ' || ' || v\_studentName || ' || ' || v\_attendancedate || ' || ' || v\_attendancestatus);  end loop;  close vcursor;  dbms\_output.put\_line('================================================');  exception  when others then  dbms\_output.put\_line('정확한 값을 입력해주세요.');  end;  /  --4. 과정별 출결 조회  create or replace procedure procGetAttendanceByCourse(  pteacherId in number,  popenedCourseId in number  )  is  cursor vcursor  is  select c.courseName as 과정명  , s.studentName as 교육생이름  , to\_char(at.attendancedate, 'yyyy-mm-dd') as 출석일  , at.attendancestatus  from tblAttendance at  inner join tblStudent s  on at.studentId = s.studentId  inner join tblOpenedCourse oc  on oc.openedCourseId = s.studentCourseId  inner join tblCourse c  on c.courseId = oc.courseId  where oc.teacherId = pteacherId --교사 자신의 ID  and oc.openedCourseId = popenedCourseId --개설과정 ID  order by c.courseName, to\_char(at.attendancedate, 'yyyy-mm-dd'), s.studentName ;  v\_courseName tblCourse.courseName%type;  v\_studentName tblStudent.studentName%type;  v\_attendancedate varchar2(10);  v\_attendancestatus tblAttendance.attendancestatus%type;  begin  open vcursor;  dbms\_output.put\_line('================================================');  dbms\_output.put\_line('|| 과정명 || 교육생이름 || 출석일 || 출결상태 ||');  dbms\_output.put\_line('================================================');  loop  fetch vcursor into v\_courseName, v\_studentName, v\_attendancedate, v\_attendancestatus;  exit when vcursor%notfound;  dbms\_output.put\_line(v\_courseName || ' || ' || v\_studentName || ' || ' || v\_attendancedate || ' || ' || v\_attendancestatus);  end loop;  close vcursor;  dbms\_output.put\_line('================================================');  exception  when others then  dbms\_output.put\_line('정확한 값을 입력해주세요.');  end;  /  --5. 교육생별 출결 조회  create or replace procedure procGetAttendanceByStudent(  pteacherId in number,  pstudentId in number  )  is  cursor vcursor  is  select c.courseName as 과정명  , s.studentName as 교육생이름  , to\_char(at.attendancedate, 'yyyy-mm-dd') as 출석일  , at.attendancestatus  from tblAttendance at  inner join tblStudent s  on at.studentId = s.studentId  inner join tblOpenedCourse oc  on oc.openedCourseId = s.studentCourseId  inner join tblCourse c  on c.courseId = oc.courseId  where oc.teacherId = pteacherId --교사 자신의 ID  and s.studentId = pstudentId --개설과정 ID  order by c.courseName, to\_char(at.attendancedate, 'yyyy-mm-dd'), s.studentName ;  v\_courseName tblCourse.courseName%type;  v\_studentName tblStudent.studentName%type;  v\_attendancedate varchar2(10);  v\_attendancestatus tblAttendance.attendancestatus%type;  begin  open vcursor;  dbms\_output.put\_line('================================================');  dbms\_output.put\_line('|| 과정명 || 교육생이름 || 출석일 || 출결상태 ||');  dbms\_output.put\_line('================================================');  loop  fetch vcursor into v\_courseName, v\_studentName, v\_attendancedate, v\_attendancestatus;  exit when vcursor%notfound;  dbms\_output.put\_line(v\_courseName || ' || ' || v\_studentName || ' || ' || v\_attendancedate || ' || ' || v\_attendancestatus);  end loop;  close vcursor;  dbms\_output.put\_line('================================================');  exception  when others then  dbms\_output.put\_line('정확한 값을 입력해주세요.');  end;  / | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-5.학생 전공 유무 조회 | 작성일 | 2025-02-05 |
| 요구사항명 | 학생 전공 유무 조회 | 작성자 | 한가람 |
| DML | | | |
| create or replace procedure procGetStudentMajorYn(  pteacherId in number  )  is  cursor vcursor  is  select c.courseName as 과정명  , to\_char(oc.openedstartdate, 'yyyymmdd') || ' ~ ' || to\_char(oc.openedcompletiondate, 'yyyymmdd') as 과정기간  , s.studentName as 학생명  , s.studentMajor as 전공유무  from tblstudent s  inner join tblOpenedCourse oc  on s.studentCourseId = oc.openedCourseId  inner join tblCourse c  on c.courseId = oc.courseId  where oc.teacherId = pteacherId --본인의 교사\_ID  ;  vcourseName tblCourse.courseName%type;  vopenedCoursePeriod varchar2(30);  vstudentName tblstudent.studentName%type;  vstudentMajor tblstudent.studentMajor%type;  begin  open vcursor;  dbms\_output.put\_line('==============================================');  dbms\_output.put\_line('|| 과정명 || 과정기간 || 학생명 || 전공유무 ||');  dbms\_output.put\_line('==============================================');  loop  fetch vcursor into vcourseName, vopenedCoursePeriod, vstudentName, vstudentMajor;  exit when vcursor%notfound;  dbms\_output.put\_line(vcourseName || ' || ' || vopenedCoursePeriod || ' || ' || vstudentName || ' || ' || vstudentMajor);  end loop;  close vcursor;  dbms\_output.put\_line('==============================================');  end;  / | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-6.자료실 관리 | 작성일 | 2025-02-05 |
| 요구사항명 | 자료실 관리 | 작성자 | 한가람 |
| DML | | | |
| --1. 자료실 글 목록 출력  create or replace procedure procPrintResourceList  is  cursor vcursor  is  select rownum as 일련번호  , resourceRoomId as 글ID  , resourceTitle as 글제목  , resourceCreationDate as 작성일  , teacherName as 작성자  from (  select rr.resourceRoomId  , rr.resourceTitle  , rr.resourceCreationDate  , t.teacherName  from tblResourceRoom rr  inner join tblTeacher t  on rr.teacherId = t.teacherId  order by rr.resourceCreationDate desc  ) xx  ;  v\_rnum number;  v\_resourceRoomId tblResourceRoom.resourceRoomId%type;  v\_resourceTitle tblResourceRoom.resourceTitle%type;  v\_resourceCreationDate tblResourceRoom.resourceCreationDate%type;  v\_teacherName tblTeacher.teacherName%type;  begin  open vcursor;  dbms\_output.put\_line('==============================================');  dbms\_output.put\_line('일련번호 || 글ID || 글제목 || 작성일 || 작성자');  dbms\_output.put\_line('==============================================');  loop  fetch vcursor into v\_rnum, v\_resourceRoomId, v\_resourceTitle, v\_resourceCreationDate, v\_teacherName;  exit when vcursor%notfound;  dbms\_output.put\_line(v\_rnum || ' || ' || v\_resourceRoomId || ' || ' || v\_resourceTitle || ' || ' || v\_resourceCreationDate || ' || ' || v\_teacherName);  end loop;  dbms\_output.put\_line('==============================================');  close vcursor;  end;  /  --2. 자료실 글 작성  create or replace procedure procWriteResource(  pteacherId number,  presourceTitle varchar2,  presourceContent varchar2  )  is  v\_resourceRoomId tblResourceRoom.resourceRoomId%type;  v\_teacherId tblTeacher.teacherId%type;  v\_teacherName tblTeacher.teacherName%type;  v\_resourceCreationDate tblResourceRoom.resourceCreationDate%type;  v\_resourceTitle tblResourceRoom.resourceTitle%type;  v\_resourceContent tblResourceRoom.resourceContent%type;  begin  INSERT INTO tblResourceRoom (resourceRoomId, teacherId, resourceTitle, resourceContent, resourceCreationDate)  VALUES ( ( select max(resourceRoomId) + 1 from tblResourceRoom ), pteacherId, presourceTitle, presourceContent, sysdate);  commit;  select max(resourceRoomId) into v\_resourceRoomId  from tblResourceRoom;  select rr.resourceRoomId  , t.teacherId  , t.teacherName  , rr.resourceCreationDate  , rr.resourceTitle  , rr.resourceContent  into v\_resourceRoomId, v\_teacherId, v\_teacherName, v\_resourceCreationDate, v\_resourceTitle, v\_resourceContent  from tblResourceRoom rr  inner join tblTeacher t  on rr.teacherId = t.teacherId  where rr.resourceRoomId = v\_resourceRoomId  ;  dbms\_output.put\_line('==================================================================');  dbms\_output.put\_line('|| 자료실\_ID || 작성자\_ID || 작성자 || 작성일자 || 제목 || 내용 ||');  dbms\_output.put\_line('==================================================================');  dbms\_output.put\_line(v\_resourceRoomId || ' || ' || v\_teacherId || ' || ' || v\_teacherName || ' || ' || v\_resourceCreationDate || ' || ' || v\_resourceTitle || ' || ' || v\_resourceContent);  dbms\_output.put\_line('==================================================================');  exception  when others then  dbms\_output.put\_line('잘못된 입력입니다.');  rollback;  end;  /  --3. 자료실 글 수정  create or replace procedure procUpdateResource(  pteacherId number,  presourceRoomId number,  presourceTitle varchar2,  presourceContent varchar2  )  is  resource\_not\_mine exception;  v\_resourceRoomId tblResourceRoom.resourceRoomId%type;  v\_teacherId tblTeacher.teacherId%type;  v\_teacherName tblTeacher.teacherName%type;  v\_resourceCreationDate tblResourceRoom.resourceCreationDate%type;  v\_resourceTitle tblResourceRoom.resourceTitle%type;  v\_resourceContent tblResourceRoom.resourceContent%type;  begin  select teacherId into v\_teacherId  from tblResourceRoom where resourceRoomId = presourceRoomId;  if pteacherId != v\_teacherId then  raise resource\_not\_mine;  end if;  update tblResourceRoom  set resourceTitle = presourceTitle  , resourceContent = presourceContent  where teacherId = pteacherId --자신이 작성한 글을 삭제하기 위해 자신의 교사\_ID를 넣는다.  and resourceRoomId = presourceRoomId  ;  commit;  dbms\_output.put\_line('작성하신 글이 수정되었습니다.');  select rr.resourceRoomId  , t.teacherId  , t.teacherName  , rr.resourceCreationDate  , rr.resourceTitle  , rr.resourceContent  into v\_resourceRoomId, v\_teacherId, v\_teacherName, v\_resourceCreationDate, v\_resourceTitle, v\_resourceContent  from tblResourceRoom rr  inner join tblTeacher t  on rr.teacherId = t.teacherId  where rr.resourceRoomId = presourceRoomId  ;  dbms\_output.put\_line('==================================================================');  dbms\_output.put\_line('|| 자료실\_ID || 작성자\_ID || 작성자 || 작성일자 || 제목 || 내용 ||');  dbms\_output.put\_line('==================================================================');  dbms\_output.put\_line(v\_resourceRoomId || chr(9) || v\_teacherId || chr(9) || v\_teacherName || chr(9) || v\_resourceCreationDate || chr(9) || v\_resourceTitle || chr(9) || v\_resourceContent);  dbms\_output.put\_line('==================================================================');  exception  when resource\_not\_mine then  dbms\_output.put\_line('자신의 글만 수정할 수 있습니다.');  when others then  dbms\_output.put\_line('수정 중 오류가 발생하였습니다.');  rollback;  end;  /  --4. 자료실 글 삭제  create or replace procedure procDeleteResource(  pteacherId number,  presourceRoomId number  )  is  resource\_not\_mine exception;  v\_resourceRoomId tblResourceRoom.resourceRoomId%type;  v\_teacherId tblTeacher.teacherId%type;  v\_teacherName tblTeacher.teacherName%type;  v\_resourceCreationDate tblResourceRoom.resourceCreationDate%type;  v\_resourceTitle tblResourceRoom.resourceTitle%type;  v\_resourceContent tblResourceRoom.resourceContent%type;  begin  select teacherId into v\_teacherId  from tblResourceRoom where resourceRoomId = presourceRoomId;  if pteacherId != v\_teacherId then  raise resource\_not\_mine;  end if;  delete from tblResourceRoom  where teacherId = pteacherId  and resourceRoomId = presourceRoomId  ;  commit ;  dbms\_output.put\_line('작성하신 글이 삭제되었습니다.');  exception  when resource\_not\_mine then  dbms\_output.put\_line('자신의 글만 수정할 수 있습니다.');  when NO\_DATA\_FOUND then  dbms\_output.put\_line('데이터가 없습니다..');  when others then  dbms\_output.put\_line('수정 중 오류가 발생하였습니다.');  rollback;  end;  /  --5. 자료실 글 조회  create or replace procedure procGetResource(  presourceRoomId number  )  is  v\_resourceRoomId tblResourceRoom.resourceRoomId%type;  v\_teacherId tblResourceRoom.teacherId%type;  v\_teacherName tblTeacher.teacherName%type;  v\_resourceCreationDate tblResourceRoom.resourceCreationDate%type;  v\_resourceTitle tblResourceRoom.resourceTitle%type;  v\_resourceContent tblResourceRoom.resourceContent%type;  begin  select resourceRoomId as 자료실\_ID  , teacherId as 작성자ID  , teacherName as 작성자  , resourceCreationDate as 작성일  , resourceTitle as 글제목  , resourceContent as 글내용  into v\_resourceRoomId, v\_teacherId, v\_teacherName, v\_resourceCreationDate, v\_resourceTitle, v\_resourceContent  from (  select rr.resourceTitle  , rr.resourceCreationDate  , rr.teacherId  , t.teacherName  , rr.resourceContent  , rr.resourceRoomId  from tblResourceRoom rr  inner join tblTeacher t  on rr.teacherId = t.teacherId  ) xx  where resourceRoomId = presourceRoomId; --선택한 글의 ID  dbms\_output.put\_line('==================================================================');  dbms\_output.put\_line('|| 자료실\_ID || 작성자\_ID || 작성자 || 작성일자 || 제목 || 내용 ||');  dbms\_output.put\_line('==================================================================');  dbms\_output.put\_line(v\_resourceRoomId || ' || ' || v\_teacherId || ' || ' || v\_teacherName || ' || ' || v\_resourceCreationDate || ' || ' || v\_resourceTitle || ' || ' || v\_resourceContent);  dbms\_output.put\_line('==================================================================');  exception  when NO\_DATA\_FOUND then  dbms\_output.put\_line('존재하지 않는 글입니다.');  when others then  dbms\_output.put\_line('조회 중 오류가 발생하였습니다.');  end;  / | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-7.경력사항 작성 | 작성일 | 2025-02-05 |
| 요구사항명 | 경력사항 작성 | 작성자 | 한가람 |
| DML | | | |
| --1. 경력사항 작성  create or replace procedure procWriteTeacherExperience(  pteacherId number,  pteacherExperienceCertification varchar2  )  is  v\_TeacherExperienceId tblTeacherExperience.TeacherExperienceId%type;  v\_teacherId tblTeacherExperience.teacherId%type;  v\_teacherName tblTeacher.teacherName%type;  v\_teacherExperienceCertification tblTeacherExperience.teacherExperienceCertification%type;  cursor vcursor  is  select te.TeacherExperienceId  , te.teacherId  , t.teacherName  , te.teacherExperienceCertification  from tblTeacherExperience te  inner join tblTeacher t  on te.teacherId = t.teacherId  where te.teacherId = pteacherId  ;  begin  insert into tblTeacherExperience(TeacherExperienceId, teacherExperienceCertification, teacherId)  values ( ( select max(TeacherExperienceId) + 1 from tblTeacherExperience ), pteacherExperienceCertification, pteacherId);  commit;  dbms\_output.put\_line('경력사항이 작성되었습니다.');  select max(TeacherExperienceId) into v\_TeacherExperienceId  from tblTeacherExperience;  dbms\_output.put\_line('=======================================================');  dbms\_output.put\_line('|| 경력사항\_ID || 교사\_ID || 이름 || 경력사항/자격증 ||');  dbms\_output.put\_line('=======================================================');  open vcursor;  loop  fetch vcursor into v\_TeacherExperienceId, v\_teacherId, v\_teacherName, v\_teacherExperienceCertification;  exit when vcursor%notfound;  dbms\_output.put\_line(v\_TeacherExperienceId || ' || ' || v\_teacherId || ' || ' || v\_teacherName || ' || ' || v\_teacherExperienceCertification);  end loop;  close vcursor;  dbms\_output.put\_line('=======================================================');  exception  when others then  dbms\_output.put\_line('작성 중 오류가 발생하였습니다.');  rollback;  end;  /  --2. 경력사항 수정  create or replace procedure procUpdateTeacherExperience(  pteacherId number,  pTeacherExperienceId number,  pteacherExperienceCertification varchar2  )  is  resource\_not\_mine exception;  v\_TeacherExperienceId tblTeacherExperience.TeacherExperienceId%type;  v\_teacherId tblTeacherExperience.teacherId%type;  v\_teacherName tblTeacher.teacherName%type;  v\_teacherExperienceCertification tblTeacherExperience.teacherExperienceCertification%type;  begin  select teacherId into v\_teacherId  from tblTeacherExperience where TeacherExperienceId = pTeacherExperienceId;  if pteacherId != v\_teacherId then  raise resource\_not\_mine;  end if;  update tblTeacherExperience  set teacherExperienceCertification = pteacherExperienceCertification  where teacherId = pteacherId --자신의 교사\_ID  and TeacherExperienceId = pTeacherExperienceId --수정 대상 경력사항\_ID  ;  commit;  dbms\_output.put\_line('작성하신 글이 수정되었습니다.');  select te.TeacherExperienceId  , te.teacherId  , t.teacherName  , te.teacherExperienceCertification  into v\_TeacherExperienceId, v\_teacherId, v\_teacherName, v\_teacherExperienceCertification  from tblTeacherExperience te  inner join tblTeacher t  on te.teacherId = t.teacherId  where TeacherExperienceId = pTeacherExperienceId  ;  dbms\_output.put\_line('=======================================================');  dbms\_output.put\_line('|| 경력사항\_ID || 교사\_ID || 이름 || 경력사항/자격증 ||');  dbms\_output.put\_line('=======================================================');  dbms\_output.put\_line(v\_TeacherExperienceId || ' || ' || v\_teacherId || ' || ' || v\_teacherName || ' || ' || v\_teacherExperienceCertification);  dbms\_output.put\_line('=======================================================');  exception  when resource\_not\_mine then  dbms\_output.put\_line('자신의 글만 수정할 수 있습니다.');  when others then  dbms\_output.put\_line('수정 중 오류가 발생하였습니다.');  rollback;  end;  /  --3. 경력사항 삭제  create or replace procedure procDeleteTeacherExperience(  pteacherId number,  pTeacherExperienceId number  )  is  resource\_not\_mine exception;  v\_teacherId tblTeacherExperience.teacherId%type;  begin  select teacherId into v\_teacherId  from tblTeacherExperience where TeacherExperienceId = pTeacherExperienceId;  if pteacherId != v\_teacherId then  raise resource\_not\_mine;  end if;  delete from tblTeacherExperience  where teacherId = pteacherId --자신의 교사\_ID  and TeacherExperienceId = pTeacherExperienceId; --삭제 대상 경력사항\_ID  commit;  dbms\_output.put\_line('작성하신 경력사항이 삭제되었습니다.');  exception  when resource\_not\_mine then  dbms\_output.put\_line('자신의 글만 수정할 수 있습니다.');  when NO\_DATA\_FOUND then  dbms\_output.put\_line('데이터가 없습니다.');  when others then  dbms\_output.put\_line('수정 중 오류가 발생하였습니다.');  rollback;  end;  /  --4. 경력사항 조회  create or replace procedure procGetTeacherExperience(  pteacherId number  )  is  cursor vcursor  is  select te.teacherExperienceId as 교사경력사항\_ID  , te.teacherId as 교사ID  , t.teacherName as 교사이름  , te.teacherExperienceCertification as "경력사항/자격증"  from tblTeacherExperience te  inner join tblteacher t  on te.teacherid = t.teacherid  where te.teacherId = pteacherId  ;  v\_teacherExperienceId tblTeacherExperience.teacherExperienceId%type;  v\_teacherId tblTeacherExperience.teacherId%type;  v\_teacherName tblteacher.teacherName%type;  v\_teacherExperienceCertification tblTeacherExperience.teacherExperienceCertification%type;  begin  open vcursor;  dbms\_output.put\_line('=======================================================');  dbms\_output.put\_line('|| 교사경력사항\_ID || 교사ID || 교사이름 || 경력사항 ||');  dbms\_output.put\_line('=======================================================');  loop  fetch vcursor into v\_teacherExperienceId, v\_teacherId, v\_teacherName, v\_teacherExperienceCertification;  exit when vcursor%notfound;  dbms\_output.put\_line(v\_teacherExperienceId || ' || ' || v\_teacherId || ' || ' || v\_teacherName || ' || ' || v\_teacherExperienceCertification);  end loop;  close vcursor;  dbms\_output.put\_line('=======================================================');  end;  / | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-8.출결 관리 | 작성일 | 2025-02-05 |
| 요구사항명 | 출결 관리 | 작성자 | 이승진 |
| DML | | | |
| -- 1. 강서연 교사가 강의하는 과정과 듣는 교육생, 출결을 조회하시오.  SELECT DISTINCT  C.openedcourseid AS 개설과정번호,  co.coursename AS 과정명,  T.teachername AS 교사명,  S.studentname AS 교육생,  A.studentid AS 교육생id  FROM  tblopenedcourse C  INNER JOIN  tblteacher T ON T.teacherid = C.teacherid  INNER JOIN  tblcourse co ON C.courseid = co.courseid  INNER JOIN  tblstudent S ON C.openedcourseid = S.studentcourseid  INNER JOIN  tblattendance A ON S.studentid = A.studentid  WHERE  T.teachername = '강서연';  -- 2. 3번 과정을 듣고 있는 모든교육생의 출석일수와 출결상태를 조회하시오.  SELECT  S.studentname AS 교육생,  EXTRACT(YEAR FROM A.attendancedate) AS 년도,  EXTRACT(MONTH FROM A.attendancedate) AS 월,  EXTRACT(DAY FROM A.attendancedate) AS 일,  A.attendancestatus AS 출결상태,  coursename AS 과정명  FROM  tblstudent S  INNER JOIN  tblopenedcourse O ON S.studentCourseId = O.Openedcourseid  INNER JOIN  tblattendance A ON s.studentId = A.studentId  INNER JOIN  tblcourse C ON O.courseid = c.courseid  WHERE  c.courseid = 3 AND  A.attendancedate BETWEEN '2024-01-01' AND '2025-12-31'  ORDER BY  A.attendancedate, c.courseid, s.studentName;  -- 3. 5번 과정을 듣고 있는 교육생들의 출결 현황을 조회(남, 여 구분)하시오.  SELECT  name AS 교육생,  courseName AS 과정명,  SUM(nomal) AS 정상,  SUM(ji) AS 지각,  SUM(gs) AS 결석  FROM (  SELECT  s.studentName AS name,  c.courseName AS courseName,  COUNT(CASE WHEN a.attendanceStatus = '정상' THEN 1 END) AS nomal,  COUNT(CASE WHEN a.attendanceStatus = '지각' THEN 1 END) AS ji,  COUNT(CASE WHEN a.attendanceStatus = '결석' THEN 1 END) AS gs,  CASE  WHEN SUBSTR(s.studentResidentNumberBack, 1, 1) IN ('1', '3') THEN '남자'  WHEN SUBSTR(s.studentResidentNumberBack, 1, 1) IN ('2', '4') THEN '여자'  ELSE '기타'  END AS 성별  FROM  tblStudent s  INNER JOIN  tblOpenedCourse o ON s.studentCourseId = o.openedCourseId  INNER JOIN  tblCourse c ON o.courseId = c.courseId  INNER JOIN  tblAttendance a ON s.studentId = a.studentId  WHERE  o.courseId = 5 -- 과정 변경  AND SUBSTR(s.studentResidentNumberBack, 1, 1) IN (1,3) -- 성별 변경  GROUP BY  s.studentName, c.courseName, a.attendanceStatus,  CASE  WHEN SUBSTR(s.studentResidentNumberBack, 1, 1) IN ('1', '3') THEN '남자'  WHEN SUBSTR(s.studentResidentNumberBack, 1, 1) IN ('2', '4') THEN '여자'  ELSE '기타'  END  ORDER BY  s.studentName  )  GROUP BY name, courseName  ORDER BY courseName, name;  -- 4. 학생들의 근태 상황을 수강하는 과정 별로 조회하고, 출석률을 오름차순으로 정렬하시오.  SELECT  name AS 교육생,  courseName AS 과정명,  SUM(nomal) AS 정상,  SUM(ji) AS 지각,  SUM(leaveEarly) AS 조퇴,  SUM(outing) AS 외출,  SUM(sick) AS 병가,  SUM(etc) AS 기타,  ROUND((SUM(nomal) \* 100.0) / NULLIF(SUM(nomal) + SUM(ji) + SUM(leaveEarly) + SUM(outing) + SUM(sick) + SUM(etc), 0), 2) AS 출석률  FROM (  SELECT  s.studentName AS name,  c.courseName AS courseName,  COUNT(CASE WHEN a.attendanceStatus = '정상' THEN 1 END) AS nomal,  COUNT(CASE WHEN a.attendanceStatus = '지각' THEN 1 END) AS ji,  COUNT(CASE WHEN a.attendanceStatus = '조퇴' THEN 1 END) AS leaveEarly,  COUNT(CASE WHEN a.attendanceStatus = '외출' THEN 1 END) AS outing,  COUNT(CASE WHEN a.attendanceStatus = '병가' THEN 1 END) AS sick,  COUNT(CASE WHEN a.attendanceStatus = '기타' THEN 1 END) AS etc  FROM  tblStudent s  INNER JOIN  tblOpenedCourse o ON s.studentCourseId = o.openedCourseId  INNER JOIN  tblCourse c ON o.courseId = c.courseId  INNER JOIN  tblAttendance a ON s.studentId = a.studentId  WHERE  o.courseId = 2 -- 과정 변경  GROUP BY  s.studentName, c.courseName  ORDER BY  s.studentName  )  GROUP BY name, courseName  ORDER BY courseName, 출석률 DESC; | | | |