**쿼리문 정의서(ANSI-SQL)**

**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  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 = 1 --교사  and sysdate < os.subjectStartDate -- 과목기간  order by oc.openedcourseid, os.openedsubjectid  ;  --2. 강의 중  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 = 1 --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  ;  --3. 강의 종료  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 = 1 --교사  and sysdate > os.subjectEndDate -- 과목기간  order by oc.openedcourseid, os.openedsubjectid  --4. 교육생 정보 조회  select s.studentName as 교육생이름  , s.studentPhoneNumber as 전화번호  , s.studentRegistrationDate as 등록일  , studentCompletionStatus as 수강상태  from tblStudent s  where s.studentCourseId in (  select openedCourseId  from tblOpenedCourse oc --개설과정  where teacherId = 1 --교사ID  and sysdate between oc.openedstartdate and oc.openedcompletiondate --현재 수강 중인 대상  ); | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-2. 배점입출력 | 작성일 | 2025-02-05 |
| 요구사항명 | 배점 입출력 | 작성자 | 한가람 |
| DML | | | |
| --1. 과목 목록 출력  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 = 1 --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  ;  --2. 출결 배점 등록 및 수정  update tblOpenedSubject set subjectattendancescoreweight = 20 where openedsubjectid = 38;  --3. 출결 배점 삭제  update tblOpenedSubject set subjectattendancescoreweight = null where openedsubjectid = 38;  --4. 출결 배점 조회  select openedsubjectid as 개설과목번호  , subjectattendancescoreweight as 출결배점  from tblOpenedSubject  where openedsubjectid = 38;    --5. 필기 배점 등록 및 수정  update tblOpenedSubject set subjectwrittenscoreweight = 20 where openedsubjectid = 38;  --6. 필기 배점 삭제  update tblOpenedSubject set subjectwrittenscoreweight = null where openedsubjectid = 38;  --7. 필기 배점 조회  select openedsubjectid as 개설과목번호  , subjectwrittenscoreweight as 필기배점  from tblOpenedSubject  where openedsubjectid = 38;  --8. 실기 배점 등록 및 수정  update tblOpenedSubject set subjectpracticalscoreweight = 20 where openedsubjectid = 38;  --9. 실기 배점 삭제  update tblOpenedSubject set subjectpracticalscoreweight = null where openedsubjectid = 38;  --10. 실기 배점 조회  select openedsubjectid as 개설과목번호  , subjectpracticalscoreweight as 실기배점  from tblOpenedSubject  where openedsubjectid = 38;    --11. 필기 시험 문제 등록  INSERT INTO tblWrittenExam(writtenExamId, openedSubjectId, writtenExamDate, writtenExamQuestions)  VALUES ( (select max(writtenExamId) + 1 from tblWrittenExam) ,38 , '2024-11-19', '파이썬 라이브러리 중 행렬이나 대규모 다차원 배열을 쉽게 처리할 수 있는 것은?');  --12. 필기 시험 문제 삭제  delete from tblWrittenExam  where writtenExamId = 78;    --13. 필기 시험 문제 수정  update tblWrittenExam  set writtenExamDate = '2024-11-19'  , writtenExamQuestions = '파이썬 라이브러리 중 행렬이나 대규모 다차원 배열을 쉽게 처리할 수 있는 것은 무엇인가요?'  where writtenExamId = 78;  --14. 필기 시험 문제 조회  select writtenExamId as 필기시험\_ID  , openedSubjectId as 개설과목\_ID  , writtenExamDate as 시험날짜  , writtenExamQuestions as 시험문제  from tblWrittenExam  where openedSubjectId = 38;  --15. 실기 시험 문제 등록  INSERT INTO tblPracticalExam(practicalExamId, openedSubjectId, practicalExamDate, practicalExamQuestions)  VALUES ( (select max(practicalExamId) + 1 from tblPracticalExam) ,38 , '2024-12-02', '파이썬을 언어를 이용해 다음의 연산을 계산하시오.');  --16. 실기 시험 문제 삭제  delete from tblPracticalExam  where practicalExamId = 78;    --17. 실기 시험 문제 수정  update tblPracticalExam  set practicalExamDate = '2024-12-02'  , practicalExamQuestions = '다음 빈칸에 들어갈 키워드를 입력하시오.'  where practicalExamId = 78;    --18. 실기 시험 문제 조회  select practicalExamId as 필기시험\_ID  , openedSubjectId as 개설과목\_ID  , practicalExamDate as 시험날짜  , practicalExamQuestions as 시험문제  from tblPracticalExam  where openedSubjectId = 38; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-3. 성적입출력 | 작성일 | 2025-02-05 |
| 요구사항명 | 성적 입출력 | 작성자 | 한가람 |
| DML | | | |
| --1. 과목 목록 출력  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  , 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 = 1 --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  ;  --2. 교육생 정보 출력  select 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  , s.studentId  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  ;  --3. 필기점수 입력  --INSERT INTO tblStudent (studentId, studentRegistrationDate, studentMajor, studentName, studentEmail, studentPhoneNumber, studentResidentNumberBack, studentCompletionStatus, studentCourseId)  -- VALUES (337, '2024-04-08','O', '테스트', 'test1111@kakao.com', '010-0505-0123', '1340124', '수강중', 7);  INSERT INTO tblGrade(gradeId, openedsubjectId, studentId, gradeWrittenScore)  VALUES (( select max(gradeId) + 1 from tblGrade ), 38, 337, 80);    --4. 실기점수 입력  update tblGrade set gradePracticalScore = 92  where openedsubjectId = 38 --개설과목명  and studentId = 337 --교육생번호; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-4. 학생 전공 유무 조회 | 작성일 | 2025-02-05 |
| 요구사항명 | 학생 전공 유무 조회 | 작성자 | 한가람 |
| DML | | | |
| 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 = 1 --본인의 교사\_ID  ; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-5. 자료실 관리 | 작성일 | 2025-02-05 |
| 요구사항명 | 학생 전공 유무 조회 | 작성자 | 한가람 |
| DML | | | |
| --1. 자료실 글 목록 출력  select rownum as 일련번호  , resourceTitle as 글제목  , resourceCreationDate as 작성일  , teacherName as 작성자  from (  select rr.resourceTitle  , rr.resourceCreationDate  , t.teacherName  from tblResourceRoom rr  inner join tblTeacher t  on rr.teacherId = t.teacherId  order by rr.resourceCreationDate desc  ) xx  ;  --2. 자료실 글 작성  INSERT INTO tblResourceRoom (resourceRoomId, teacherId, resourceTitle, resourceContent, resourceCreationDate)  VALUES ( ( select max(resourceRoomId) + 1 from tblResourceRoom ), 1, '(도큐먼트, Ruby) Ruby Programming Language Documentation', 'https://docs.ruby-lang.org/en/', sysdate);  --3. 자료실 글 수정  update tblResourceRoom  set resourceTitle = '(도큐먼트, Pandas) pandas documentation'  , resourceContent = 'https://pandas.pydata.org/docs/'  where teacherId = 1 --자신이 작성한 글을 삭제하기 위해 자신의 교사\_ID를 넣는다.  and resourceRoomId = 66  --4. 자료실 글 삭제  delete from tblResourceRoom  where teacherId = 1  and resourceRoomId = 66  ;  --5. 자료실 글 조회  select resourceRoomId as 자료실\_ID  , teacherName as 작성자  , resourceCreationDate as 작성일  , resourceTitle as 글제목  , resourceContent as 글내용  from (  select rr.resourceTitle  , rr.resourceCreationDate  , t.teacherName  , rr.resourceContent  , rr.resourceRoomId  from tblResourceRoom rr  inner join tblTeacher t  on rr.teacherId = t.teacherId  ) xx  where resourceRoomId = 66; --선택한 글의 ID | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-6. 경력사항 작성 | 작성일 | 2025-02-05 |
| 요구사항명 | 경력사항 작성 | 작성자 | 한가람 |
| DML | | | |
| --1. 경력사항 작성  insert into tblTeacherExperience(TeacherExperienceId, teacherExperienceCertification, teacherId)  values ( ( select max(TeacherExperienceId) + 1 from tblTeacherExperience ), '정보처리기사', 1)  --2. 경력사항 수정  update tblTeacherExperience  set teacherExperienceCertification = '정보처리기술사'  where teacherId = 1 --자신의 교사\_ID  and TeacherExperienceId = 70 --수정 대상 경력사항\_ID  ;  --3. 경력사항 삭제  delete from tblTeacherExperience  where teacherId = 1 --자신의 교사\_ID  and TeacherExperienceId = 70 --삭제 대상 경력사항\_ID  ;  --4. 경력사항 조회  select te.teacherExperienceId as 교사경력사항\_ID  , te.teacherExperienceCertification as "경력사항/자격증"  , t.teacherName as 교사이름  from tblTeacherExperience te  inner join tblteacher t  on te.teacherid = t.teacherid  where te.teacherId = 1; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-7. 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; | | | |