**데이터 쿼리문(ANSI-SQL)**

| 요구사항번호 | A-01 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 주변 맛집 추천 | 작성자 | 김지우 |
| DML | | | |
| -- 관리자, 교사, 교육생 모두 가능 > 로그인  (예시: 교육생)  select  s.studentSeq as "교육생번호",  a.name as "이름",  a.tel as "전화번호"  from tblApplicant a  inner join tblStudent s  on a.applicantSeq = s.studentSeq  where ssn = 1066962; -- 주민번호 뒷자리 입력 시 로그인  -- 가게명 주소, 거리, 추천수, 메뉴 정보(음식 및 가격) 출력  select  r.name as "가게명",  r.address as "주소",  r.time as "거리",  r.likeCount as "추천수",  m.name as "메뉴",  m.price as "가격"  from tblRestaurant r  inner join tblMenu m  on r.restaurantSeq = m.menuSeq;    -- 새로운 맛집 및 메뉴 추가 기능  insert into tblRestaurant(restaurantSeq, name, address, time, likeCount) values (81, '죠샌드위치', '서울 강남구 논현로87길 41', 2, 0);  insert into tblMenu(menuSeq, restaurantSeq, name, price) values (81, 81, '샌드위치', 7000);  select \* from tblRestaurant;  select \* from tblMenu  -- 특정 맛집 추천 > 추천수 증가  update tblRestaurant set  likeCount = 20  where restaurantSeq = 81;  -- 해당 맛집 가격 및 메뉴 정보 수정 기능  update tblMenu set  name = '곡물 샌드위치',  price = 7500  where menuSeq = 81;  --가게 및 메뉴 삭제  delete from tblRestaurant where restaurantSeq = 81;  delete from tblMenu where menuSeq = 81; | | | |

| 요구사항번호 | B-01 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 기초 정보 관리 | 작성자 | 김지우 |
| DML | | | |
| --로그인 > 주민번호 뒷자리 7개 입력 > 번호 및 아이디 출력  select  adminSeq as "관리자 번호",  id as "관리자 id"  from tblAdmin  where pw = 1904993;  -- 과정 추가  select \* from tblCourse;  insert into tblCourse(courseSeq, courseTitle) values (21, '실무 프로젝트 기반의 응용 SW 개발자 양성과정');  -- 과정명 수정  update tblCourse set  courseTitle = '자바, 스프링 기반의 응용 SW 개발자 양성과정'  where courseSeq = 21;  -- 과정 삭제  delete from tblCourse where courseSeq = 21;  -- 과목 추가  select \* from tblSubject;  insert into tblSubject(subjectSeq, subjectTitle) values (34, '자료구조');  -- 과목명 수정  update tblSubject set  subjectTitle = '안드로이드 어플 앱'  where subjectSeq = 34;  -- 과목 삭제  delete from tblSubject where subjectSeq = 34;  -- 교재 추가  select \* from tblBook;  insert into tblBook(bookSeq, BookTitle, publisher) values (52, '세상에서 가장 쉬운 코딩책', '길벗');  -- 교재 수정  update tblBook set  bookTitle = '세상에서 가장 보기 쉬운 코딩책'  where bookSeq = 52;  -- 교재 삭제  delete from tblBook where bookSeq = 52;  -- 강의실 추가  select \* from tblClassroom;  insert into tblClassroom(classroomSeq, classroomTitle, capacity) values (7, '강의실7', '30');  -- 강의실 정원 수 수정  update tblClassroom set  capacity = '26'  where classroomSeq = 7;  -- 강의실 삭제  delete from tblClassroom where classroomSeq = 7; | | | |

| 요구사항번호 | B-02 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 교사 계정 관리 | 작성자 | 백정석 |
| DML | | | |
| 1. 교사 전체 출력   SELECT  t.name as 교사명,  t.ssn as "주민번호 뒷자리",  t.tel as 전화번호,  s.subjecttitle as 강의강의과목  FROM tblteacher t  INNER JOIN tblteachersubject ts ON t.teacherseq = ts.teacherseq  INNER JOIN tblsubject s ON ts.subjectseq = s.subjectseq;   1. 특정 교사 선택 출력   SELECT  s.subjecttitle AS "배정된 개설 과목",  os.startdate AS "과목 시작 일",  os.enddate AS "과목 종료 일",  c.coursetitle AS "과정명",  oc.startdate AS "과정 시작 일",  oc.enddate AS "과정 종료 일",  b.booktitle AS "교재명",  cr.classroomtitle AS "강의실",  CASE  WHEN oc.enddate <= sysdate THEN '강의종료'  WHEN oc.startdate <= sysdate THEN '강의중'  WHEN oc.startdate >= sysdate THEN '예정'  END AS "강의진행여부"  FROM tblopensubject os  INNER JOIN tblteacher t ON os.teacherseq = t.teacherseq  INNER JOIN tblsubject s ON os.subjectseq = s.subjectseq  INNER JOIN tblopencourse oc ON os.opencourseseq = oc.opencourseseq  INNER JOIN tblcourse c ON oc.courseseq = c.courseseq  INNER JOIN tblbook b ON os.bookseq = b.bookseq  INNER JOIN tblclassroom cr ON oc.classroomseq = cr.classroomseq  WHERE t.name = '이지은'; | | | |

| 요구사항번호 | B-03 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 개설 과정 관리 | 작성자 | 이재성 |
| DML | | | |
| **1. 과정명 추가**  INSERT INTO tblCourse values (21, <과정명>);    **2. 개설 과정 조회**  SELECT \* from tblOpenCourse;    **3. 개설 과정 등록**  INSERT INTO TBLOPENCOURSE (OPENCOURSESEQ,CLASSROOMSEQ,COURSESEQ,STARTDATE,ENDDATE,CAPACITY)  VALUES (<과정시퀀스> , <강의실시퀀스>, <과정시퀀스> , <개설과정시작일>, <개설과정종료일>, <인원>);    **4. 개설 과정 정보 출력**  select  distinct  c.coursetitle as "과정명",  oc.startdate as "과정시작일",  oc.enddate as "과정종료일",  cr.classroomtitle as "강의실명",  case  when os.subjectseq is not null then '등록'  end as "개설과목등록여부",  oc.capacity as "교육생 등록 인원"  from tblOpenCourse oc  inner join tblCourse c on oc.courseseq = c.courseseq  inner join tblStudent s on oc.opencourseseq = s.opencourseseq  inner join tblClassroom cr on cr.classroomseq = oc.classroomseq  inner join tblOpenSubject os on os.opencourseseq = oc.opencourseseq  order by oc.startdate;    **5. 특정 개설 과정 선택 시 등록된 개설 과목 정보 확인**  select  s.subjecttitle as "과목명",  oc.startdate as "과목시작일",  oc.enddate as "과목종료일",  b.booktitle as "교재명",  t.name as "교사명",  a.name as "교육생 이름",  a.ssn as "주민번호 뒷자리",  a.tel as "전화번호",  st.registerdate as "등록일",  case  when st.completeorfail = 1 then '수료'  when st.completeorfail = 0 then '중도탈락'  when st.completeorfail is null then '수강중'  end as "수료 및 중도탈락"  from tblOpencourse oc  inner join tblOpenSubject os on os.opencourseseq = oc.opencourseseq  inner join tblSubject s on os.subjectseq = s.subjectseq  inner join tblBook b on os.bookseq = b.bookseq  inner join tblteacher t on os.teacherseq = t.teacherseq  inner join tblStudent st on oc.opencourseseq = st.opencourseseq  inner join tblApplicant a on st.applicantseq = a.applicantseq  where opencourseseq = <개설과정번호>;    **6. 개설 과정에 등록된 교육생 확인**  select  oc.opencourseseq as "과정번호",  a.name as "이름",  a.ssn as "주민번호 뒷자리",  a.tel as "전화번호",  s.registerdate "등록일",  case  when s.completeorfail = 1 then '수료'  when s.completeorfail = 0 then '중도탈락'  when s.completeorfail is null then '수강중'  end as "수료 및 중도탈락"  from tblOpenCourse oc  inner join tblStudent s on oc.opencourseseq = s.opencourseseq  inner join tblApplicant a on s.applicantseq = a.applicantseq  where oc.opencourseseq = <개설과정번호>  order by a.name;  **7. 특정 개설 과정 수료 후 수료 날짜 지정**  update tblStudent  set completeorfaildate = <수료날짜>  where opencourseseq = <개설과정번호> and completeorfail is null;    **8. 개설 과정 삭제**  Delete tblOpenCourse where opencourseseq = <개설과정번호>; | | | |

| 요구사항번호 | B-04 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 개설 과목 과정 | 작성자 | 이재성 |
| DML | | | |
| **1. 개설 과목 입력**  INSERT INTO TBLOPENSUBJECT (OPENSUBJECTSEQ,SUBJECTSEQ,BOOKSEQ,TEACHERSEQ,POINTSSEQ,OPENCOURSESEQ,STARTDATE,ENDDATE)  VALUES (<개설과목번호>, <과목번호>, <교재번호>, <교사번호>, <배점번호>, <개설과정번호>, <개설과목시작일>, <개설과목종료일>);    **2. 개설 과목 수정**  UPDATE tblOpenSubject SET <수정할 내용> WHERE opensubjectseq = <개설과목번호>;    **3. 개설 과목 삭제**  DELETE tblOpenSuject WHERE opensubjectseq = <개설과목번호>;    **4. 개설 과목 목록 조회**  SELECT \* FROM tblOpenSubject;    **5. 교사 강의 가능 목록 출력**  select  rownum,  s.subjecttitle as "과목명",  t.name "교사명"  from tblSubject s  inner join tblTeacherSubject ts on s.subjectseq = ts.subjectseq  inner join tblTeacher t on ts.teacherseq = t.teacherseq  where subjecttitle = <과목명>;    **6. 특정 과정 선택 시 과정에 등록된 과목 출력**  select  os.opensubjectseq,  c.coursetitle as "과정명",  oc.startdate || '~' || oc.enddate as "과정진행날짜",  s.subjecttitle as "과목명",  os.startdate || '~' || os.enddate as "과목진행날짜",  b.booktitle as "교재명",  t.name as "교사명"  from tblOpenSubject os  inner join tblSubject s on os.subjectseq = s.subjectseq  inner join tblBook b on os.bookseq = b.bookseq  inner join tblTeacher t on os.teacherseq = t.teacherseq  inner join tblOpenCourse oc on oc.opencourseseq = os.opencourseseq  inner join tblCourse c on c.courseseq = oc.courseseq  where os.opencourseseq = <개설과정번호>  order by opensubjectseq; | | | |

| 요구사항번호 | B-05 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 관리 | 작성자 | 백정석 |
| DML | | | |
| 1. 교육생 전체 출력   SELECT  a.name AS "이름",  a.ssn AS "주민번호",  a.tel AS "전화번호",  s.registerdate AS "등록일",  (  SELECT COUNT(\*)  FROM tblstudent s2  WHERE s2.applicantseq = s.applicantseq  ) AS "수강횟수"  FROM tblapplicant a  INNER JOIN tblstudent s ON a.applicantseq = s.applicantseq  INNER JOIN tblOpenCourse oc ON oc.openCourseSeq = s.openCourseSeq;   1. 특정 교육생 출력   SELECT  c.coursetitle AS 과정명,  oc.startdate AS "과정 시작 일",  oc.enddate AS "과정 종료 일",  cr.classroomtitle AS "강의실",  CASE  WHEN s.completeOrFail = 0 THEN '중도탈락'  WHEN s.completeOrFail = 1 THEN '수료'  ELSE '수강중'  END AS "수료여부"  FROM tblstudent s  INNER JOIN tblopencourse oc ON s.opencourseseq = oc.opencourseseq  INNER JOIN tblcourse c ON oc.courseseq = c.courseseq  INNER JOIN tblclassroom cr ON oc.classroomseq = cr.classroomseq  WHERE s.studentseq = '1'; | | | |

| 요구사항번호 | B-06 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 시험 관리 및 성적 조회 | 작성자 | 이재성 |
| DML | | | |
| **1. 특정 개설 과정 선택 시 등록된 개설 과정 출력**  select  s.subjecttitle as "과목명",  os.startdate as "과목시작일",  os.enddate as "과목종료일",  c.coursetitle as "개설과정명",  cr.classroomtitle as "강의실명",  t.name as "교사명",  b.booktitle as "교재명"  from tblOpenCourse oc  inner join tblOpenSubject os on oc.opencourseseq = os.opencourseseq  left outer join tblexamregister e on os.opensubjectseq = e.opensubjectseq  left outer join (select opensubjectseq from tblScore group by opensubjectseq) oi on os.opensubjectseq = oi.opensubjectseq  inner join tblSubject s on s.subjectseq = os.subjectseq  inner join tblCourse c on c.courseseq = oc.courseseq  inner join tblTeacher t on t.teacherseq = os.teacherseq  inner join tblBook b on b.bookseq = os.bookseq  inner join tblClassroom cr on cr.classroomseq = oc.classroomseq  where oc.opencourseseq = <개설과정번호>    **2. 특정 개설 과정 선택 시 성적 등록 여부 및 시험 문제 파일 등록 여부 출력**  select  oc.opencourseseq,  s.subjectseq as "과목번호",  s.subjecttitle as "과목명",  case  when oi.opensubjectseq is not null then 'Y'  else 'N'  end as "성적등록여부",  case  when writtendate is not null then 'Y'  else 'N'  end as "필기시험등록여부",  case  when practicialdate is not null then 'Y'  else 'N'  end as "실기시험등록여부"  from tblOpenSubject os  inner join tblOpenCourse oc on os.opencourseseq = oc.opencourseseq  inner join tblSubject s on os.subjectseq = s.subjectseq  inner join tblExamRegister e on os.opensubjectseq = e.opensubjectseq  left outer join (select opensubjectseq from tblScore group by opensubjectseq) oi on os.opensubjectseq = oi.opensubjectseq  where oc.opencourseseq = <개설과정번호>  order by oc.opencourseseq;    **3. 과목 선택 후 수강하는 교육생들 성적 출력**  select  rownum,  c.coursetitle as "개설과정명",  oc.startdate || '~' || oc.enddate as "개설과정기간",  cr.classroomtitle as "강의실명",  sj.subjecttitle as "개설 과목명",  t.name as "교사명",  b.booktitle as "교재명",  a.name as "교육생명",  a.ssn as "주민번호 뒷자리",  s.writtenscore as "필기점수",  s.practicalscore as "실기점수"  from tblOpenSubject os  inner join tblScore s on os.opensubjectseq = s.opensubjectseq  inner join tblOpenCourse oc on oc.opencourseseq = os.opencourseseq  inner join tblCourse c on oc.courseseq = c.courseseq  inner join tblClassroom cr on oc.classroomseq = cr.classroomseq  inner join tblTeacher t on os.teacherseq = t.teacherseq  inner join tblBook b on os.bookseq = b.bookseq  inner join tblStudent st on s.studentseq = st.studentseq  inner join tblApplicant a on st.applicantseq = a.applicantseq  inner join tblSubject sj on os.subjectseq = sj.subjectseq  where os.opensubjectseq = <개설과목번호>;  **5. 개인별 수강 과목 성적 출력**  select  st.studentseq,  a.name as "교육생 이름",  a.ssn as "주민번호 뒷자리",  c.coursetitle as "개설과정명",  oc.startdate || '~' || oc.enddate as "개설과정기간",  cr.classroomtitle as "강의실명",  sj.subjecttitle as "개설과목명",  os.startdate || '~' || os.enddate as "개설과목기간",  t.name as "교사명",  s.attendancescore as "출석점수",  s.writtenscore as "필기점수",  s.practicalscore as "실기점수"  from tblOpenSubject os  inner join tblScore s on os.opensubjectseq = s.opensubjectseq  inner join tblStudent st on s.studentseq = st.studentseq  inner join tblApplicant a on a.applicantseq = st.applicantseq  inner join tblSubject sj on os.subjectseq = sj.subjectseq  inner join tblOpenCourse oc on os.opencourseseq = oc.opencourseseq  inner join tblClassRoom cr on oc.classroomseq = cr.classroomseq  inner join tblTeacher t on os.teacherseq = t.teacherseq  inner join tblCourse c on oc.courseseq = c.courseseq  where st.studentseq = <학생번호>; | | | |

| 요구사항번호 | B-07 (관리자) | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 출결 관리 및 출결 조회 | 작성자 | 남건욱 |
| DML | | | |
| **- 관리자가 특정 과정에 모든 교육생 출결 확인**  # 프로시저로 처리  **- 특정 인원의 출결 전체 조회**  select  to\_char(checkout,'yyyy-mm-dd') as "날짜",  case when checkin is null then  '안옴'  else  to\_char(checkin,'yyyy-mm-dd hh24:mi')  end  as "입실시간",    case when checkin is null then  '안옴'  else  to\_char(checkout,'yyyy-mm-dd hh24:mi')  end  as "퇴실시간",  case when record is null and checkin is not null and to\_char(checkin,'hh24:mi') < '09:01' then '정상'  when to\_char(checkin,'hh24:mi') > '09:00' then '지각'  when checkin is null and checkout is null then record  when to\_char(checkout,'hh24:mi') < '14:00' then '조퇴'  else record end as "분류"  from (select \* from tblattendance where studentseq = 89) order by 날짜; -- 학생의 seq 입력  - **특정 인원의 출결 현황 조회 > 달 조회**  select  to\_char(checkout,'yyyy-mm-dd') as "날짜",  case when checkin is null then  '안옴'  else  to\_char(checkin,'yyyy-mm-dd hh24:mi')  end  as "입실시간",    case when checkin is null then  '안옴'  else  to\_char(checkout,'yyyy-mm-dd hh24:mi')  end  as "퇴실시간",  case when record is null and checkin is not null and to\_char(checkin,'hh24:mi') < '09:01' then '정상'  when to\_char(checkin,'hh24:mi') > '09:00' then '지각'  when checkin is null and checkout is null then record  when to\_char(checkout,'hh24:mi') < '14:00' then '조퇴'  else record end as "분류"  from (select \* from tblattendance where studentseq = 83) where checkout >= '2022-08-01' and checkout < '2022-09-01 ' order by 날짜;  **- 특정 인원의 출결 현황 조회 > 일 조회**  select to\_char(checkout,'yyyy-mm-dd') as "날짜", to\_char(checkin,'yyyy-mm-dd hh24:mi') as "입실시간", to\_char(checkout,'yyyy-mm-dd hh24:mi') as "퇴실시간",  case when record is null and checkin is not null and to\_char(checkin,'hh24:mi') < '09:01' then '정상'  when to\_char(checkin,'hh24:mi') > '09:00' then '지각'  when checkin is null and checkout is null then record  when to\_char(checkout,'hh24:mi') < '14:00' then '조퇴'  else record end as "분류"  from tblattendance where studentseq = 260 and to\_char(checkin,'yyyy-mm-dd') = '2023-04-04'; -- 학생의 시퀀스 번호, 요일입력 | | | |

| 요구사항번호 | B-08 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 선발과정 | 작성자 | 이재성 |
| DML | | | |
| **1. 지원한 전체 사람들의 정보 출력**  select  a.name as "이름",  a.ssn as "주민번호",  a.tel as "전화번호",  r.recruitdate as "면접날짜",  r.score as "면접점수",  case  when pass = 'pass' then '합격'  else '탈락'  end as "합격여부"  from tblRecruit r  inner join tblApplicant a  on a.applicantseq = r.applicantseq  order by r.recruitdate;    **2. 합격한 지원자들의 인적사항, 면접 점수, 면접 날짜 출력**  select  a.name as "이름",  a.ssn as "주민번호",  a.tel as "전화번호",  r.score as "면접점수",  r.recruitdate as "면접날짜",  r.interviewer as "면접관"  from tblRecruit r  inner join tblApplicant a on a.applicantseq = r.applicantseq  order by r.recruitdate; | | | |

| 요구사항번호 | B-09 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 세미나 초청 | 작성자 | 백정석 |
| DML | | | |
| 1. 세미나 정보 출력   SELECT  s.teacher AS 강사명,  s.title AS 강의명,  s.seminardate AS 강연날짜,  c.name AS 기업명,  c.companysize AS 기업규모  FROM tblseminar s  INNER JOIN tblcompany c ON s.companyseq = c.companyseq; | | | |

| 요구사항번호 | B-10 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 취업결과관리 (추가요구사항( | 작성자 | 오소엽 |
| DML | | | |
| -- 관리자  -- 취업결과 작성  insert into tblJobResult(jobResultSeq, studentSeq, hireDate, job, company, companySize, satisfaction)  values( (select max(jobResultSeq) + 1 from tbljobResult), **<교육생seq>**, ‘**<취업날짜>**’, ‘**<직무>**’, ‘**<기업명>’**, ‘**<기업규모>**’, **<취업후만족도>**);  -- 취업결과 전체 조회 (vwJobResult로 조회할 수도 있다.)  select  stdn.studentSeq as 교육생번호,  jr.hireDate as 입사일,  jr.job as 직무,  jr.company as 기업명,  jr.companySize as 기업규모,  jr.satisfaction || '점' as 만족도  from tblStudent stdn  inner join tbljobResult jr  on stdn.studentSeq = jr.studentSeq;    -- 기간별 취업결과 조회 (연도별, 특정 기간별)  -- 연도별  select  jr.hireDate as 입사일,  stdn.studentSeq as 교육생번호,  a.name as 이름,  jr.job as 직무,  jr.company as 기업명,  jr.companySize as 기업규모,  jr.satisfaction || '점' as 만족도,  c.courseTitle as 수강과정  from tblStudent stdn  inner join tbljobResult jr  on stdn.studentSeq = jr.studentSeq  inner join tblApplicant a  on a.applicantSeq = stdn.applicantSeq  inner join tblOpenCourse openC  on stdn.openCourseSeq = openC.openCourseSeq  inner join tblCourse c  on openC.courseSeq = c.courseSeq  where substr(to\_char(hireDate, 'yyyy-mm-dd'),1,4) = ‘**<검색할 연도>**’;    -- 특정 기간별  select  stdn.studentSeq as 교육생번호,  a.name as 이름,  jr.hireDate as 입사일,  jr.job as 직무,  jr.company as 기업명,  jr.companySize as 기업규모,  jr.satisfaction || '점' as 만족도,  c.courseTitle as 수강과정  from tblStudent stdn  inner join tbljobResult jr  on stdn.studentSeq = jr.studentSeq  inner join tblApplicant a  on a.applicantSeq = stdn.applicantSeq  inner join tblOpenCourse openC  on stdn.openCourseSeq = openC.openCourseSeq  inner join tblCourse c  on openC.courseSeq = c.courseSeq;  where hireDate between (**‘<신청날짜 or 스터디날짜>'**) and (**‘<신청날짜 or 스터디날짜>’**)  order by jr.hireDate, stdn.studentSeq;  -- 직무별 취업결과 조회  select  stdn.studentSeq as 교육생번호,  a.name as 이름,  jr.hireDate as 입사일,  jr.job as 직무,  jr.company as 기업명,  jr.companySize as 기업규모,  jr.satisfaction || '점' as 만족도,  c.courseTitle as 수강과정  from tblStudent stdn  inner join tbljobResult jr  on stdn.studentSeq = jr.studentSeq  inner join tblApplicant a  on a.applicantSeq = stdn.applicantSeq  inner join tblOpenCourse openC  on stdn.openCourseSeq = openC.openCourseSeq  inner join tblCourse c  on openC.courseSeq = c.courseSeq;  where jr.companySize in (‘**<검색할 기업규모>**’)  order by jr.hireDate, jr.company;  -- 기업별 취업결과 조회  -- 기업(삼성전자, 삼성SDS, 라인, 넷마블게임즈, SK이노베이션, LG전자, SK텔레콤, 더존비즈온, KT, 카카오, SK하이닉스, 네이버, 쿠팡)  select  stdn.studentSeq as 교육생번호,  a.name as 이름,  jr.hireDate as 입사일,  jr.job as 직무,  jr.company as 기업명,  jr.companySize as 기업규모,  jr.satisfaction || '점' as 만족도,  c.courseTitle as 수강과정  from tblStudent stdn  inner join tbljobResult jr  on stdn.studentSeq = jr.studentSeq  inner join tblApplicant a  on a.applicantSeq = stdn.applicantSeq  inner join tblOpenCourse openC  on stdn.openCourseSeq = openC.openCourseSeq  inner join tblCourse c  on openC.courseSeq = c.courseSeq  where jr.company like '%**<검색할 기업명>**%'  order by jr.hireDate;    -- 만족도별 취업결과 조회  select  stdn.studentSeq as 교육생번호,  a.name as 이름,  jr.hireDate as 입사일,  jr.job as 직무,  jr.company as 기업명,  jr.companySize as 기업규모,  jr.satisfaction || '점' as 만족도,  c.courseTitle as 수강과정  from tblStudent stdn  inner join tbljobResult jr  on stdn.studentSeq = jr.studentSeq  inner join tblApplicant a  on a.applicantSeq = stdn.applicantSeq  inner join tblOpenCourse openC  on stdn.openCourseSeq = openC.openCourseSeq  inner join tblCourse c  on openC.courseSeq = c.courseSeq  where jr.satisfaction >= **<검색할 만족도 범위>**  order by jr.satisfaction desc; | | | |

| 요구사항번호 | C-01 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 강의스케줄 조회 | 작성자 | 백정석 |
| DML | | | |
| 1. 특정 교사 스케줄 출력   SELECT  os.subjectseq AS "과목번호",  c.coursetitle AS "과정명",  oc.startdate AS "과정 시작 일",  oc.enddate AS "과정 종료 일",  cr.classroomtitle AS "강의실",  os.startdate AS "과목 시작 일",  os.enddate AS "과목 종료 일",  b.booktitle AS "교재명",  (  SELECT COUNT(\*)  FROM tblstudent t1  WHERE t1.opencourseseq = oc.opencourseseq  ) AS "등록 인원"  FROM tblteacher t  INNER JOIN tblopensubject os ON t.teacherseq = os.teacherseq  INNER JOIN tblopencourse oc ON os.opencourseseq = oc.opencourseseq  INNER JOIN tblcourse c ON oc.courseseq = c.courseseq  INNER JOIN tblclassroom cr ON oc.classroomseq = cr.classroomseq  INNER JOIN tblbook b ON b.bookseq = os.bookseq  WHERE t.teacherseq = 1  ORDER BY os.subjectseq;   1. 교육중인 과정 교육생 출력   SELECT DISTINCT  s.studentseq AS "교육생 번호",  a.name AS "교육생 이름",  a.tel AS "전화번호",  s.registerdate AS "등록일",  CASE  WHEN s.completeOrFail = 0 THEN '중도탈락'  WHEN s.completeOrFail = 1 THEN '수료'  ELSE '수강중'  END AS "수료여부"  FROM tblapplicant a  INNER JOIN tblstudent s ON a.applicantseq = s.applicantseq  INNER JOIN tblopencourse oc ON s.opencourseseq = oc.opencourseseq  INNER JOIN tblopensubject os ON oc.opencourseseq = os.opencourseseq  INNER JOIN tblteacher t ON os.teacherseq = t.teacherseq  WHERE t.teacherseq = 5  AND oc.startdate <= sysdate  AND oc.enddate >= sysdate; | | | |

| 요구사항번호 | C-02 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 배점 입출력 | 작성자 | 백정석 |
| DML | | | |
| 1. 배점정보 출력   SELECT  s.subjectseq AS "과목번호",  c.coursetitle AS "과정명",  oc.startdate AS "과정 시작 일",  oc.enddate AS "과정 종료 일",  cr.classroomtitle AS "강의실",  s.subjecttitle AS "과목명",  os.startdate AS "과목 시작 일",  os.enddate AS "과목 종료 일",  b.booktitle AS "교재명",  p.attendance AS "출결배점",  p.written AS "필기배점",  p.practical AS "실기배점"  FROM tblpoints p  INNER JOIN tblopensubject os ON p.pointsseq = os.pointsseq  INNER JOIN tblsubject s ON os.subjectseq = s.subjectseq  INNER JOIN tblopencourse oc ON os.opencourseseq = oc.opencourseseq  INNER JOIN tblcourse c ON oc.courseseq = c.courseseq  INNER JOIN tblclassroom cr ON oc.classroomseq = cr.classroomseq  INNER JOIN tblbook b ON os.bookseq = b.bookseq;   1. 특정과목 배점 출력   SELECT  p.attendance AS "출결배점",  p.written AS "필기배점",  p.practical AS "실기배점",  er.writtendate AS "필기 시험날짜",  er.practicaldate AS "실기 시험날짜"  FROM tblopensubject os  INNER JOIN tblsubject s ON os.subjectseq = s.subjectseq  LEFT OUTER JOIN tblpoints p ON os.pointsseq = p.pointsseq  INNER JOIN tblexamregister er ON os.opensubjectseq = er.opensubjectseq  WHERE s.subjectseq = 1; | | | |

| 요구사항번호 | C-03 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 성적 입출력 | 작성자 | 남건욱 |
| DML | | | |
| **- 교사가 자신이 강의하는 과목을 조회**  select  t.opensubjectseq as 개설과목번호,  ( select subjecttitle from tblsubject where subjectseq =t .subjectseq) as 과목명,  t.startdate as 시작기간,  t.enddate as 끝기간,  case  when t.startdate < sysdate and t.enddate > sysdate then '진행중'  when t.enddate < sysdate then '강의종료'  else '강의예정'  end as 강의여부  from tblopensubject t where t.teacherseq = 4; -- 교사 시퀀스 입력, 강의종료, 강의중인, 강의예정인 과목 모두 출력  **-- 채점을 과목을 선택 > 개설과목번호를 시퀀스를 기억 > [ opensubjectseq ]**  **-- 오프라인으로 시험문제를 채점 ~**  **-- 시험 점수 레코드 추가 ( PL/SQL에 추가 )**  insert into tblscore values((select nvl( max(attendanceseq), 0) + 1 from tblscore),  “개설과목번호”, “원하는 교육생의 시퀀스 번호”, null, null, null);  **- 출석 점수 입력 ( PL/SQL에 추가 )**  update tblscore set attendancescore = “점수” where studentseq = “원하는 교육생의 시퀀스 번호” and opensubjectseq **= “성적 입력원하는 과목 시퀀스”;**  **- 필기 점수 입력 ( PL/SQL에 추가 )**  update tblscore set writtenscore= “점수” where studentseq = “원하는 교육생의 시퀀스 번호” and opensubjectseq **= “성적 입력원하는 과목 시퀀스”;**  **- 실기 점수 입력 ( PL/SQL에 추가 )**  update tblscore set practicalscore= “점수” where studentseq = **“원하는 교육생의 시퀀스 번호”** and opensubjectseq **= “성적 입력원하는 과목 시퀀스”;**  - 과목 목록, 과목번호, 과정명, 과정기간, 강의실, 과목명, 과목기간, 교재멸  **- 과목 목록 출력 시 과목번호, 과정명, 과정기간(시작 년월일, 끝 년월일) 강의실 과목명, 과목기간(시작년월일, 끝년월일), 교재명, 출결, 필기, 실기배점, 성적등록여부 출력 ( PL/SQL에 추가 )**  - # 프로시저 사용 ( 이 때, 한 교육 생의 전체 출결 정보 확인 SQL을 이용)  **- 특정 과목번호로 선택시 교육생 정보 및 성적이 출결, 필기, 실기 점수로 구분되어서 출력되어야 한다. 단 중도 탈락자를 고려하여 출력**  select  (select name from tblapplicant a inner join tblstudent s on a.applicantseq = s.applicantseq where s.studentseq=sos.studentseq) as "이름",  (select tel from tblapplicant a inner join tblstudent s on a.applicantseq = s.applicantseq where s.studentseq=sos.studentseq) as "전화번호",  sos.studentseq as "학생번호",  sos.opensubjectseq as "과목번호",  case  when sos.completeorfail = 0 then '중도포기자'  when sos.completeorfail = 1 then '수료자'  else '교육수강중'  end as "중도포기여부",  case  when sos.completeorfail = 0 then to\_char(sos.completeorfaildate,'yyyy-mm-dd')  else '해당사항없음'  end as "중도포기날짜",  case  when  sos.completeorfail is null or sos.completeorfail = 1 then  sc.attendancescore  end as "출결점수",  case  when sos.completeorfail is null or sos.completeorfail = 1 then  sc.writtenscore end  as "필기점수",  case  when sos.completeorfail is null or sos.completeorfail = 1 then  sc.practicalscore end as "실기점수"  from (select  \*  from tblstudent s inner join tblopensubject os on s.opencourseseq = os.opencourseseq and os.opensubjectseq= 50) sos inner join tblscore sc on sos.studentseq=sc.studentseq and sc.opensubjectseq = 50; -- 개설과목 시퀀스를 입력 | | | |

| 요구사항번호 | C-04 (교사) | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 출결 관리 및 출결 조회 | 작성자 | 남건욱 |
| DML | | | |
| **- 교사가 강의하는 과정 조회**  select  oc.opencourseseq as "과정번호",  (select coursetitle from tblcourse where tblcourse.courseseq=oc.courseseq ) as "강의과정",  oc.startdate as "시간기간",  oc.enddate as "끝기간",  case  when oc.startdate < sysdate and oc.enddate > sysdate then '진행중'  when oc.enddate < sysdate then '강의종료'  else '강의예정'  end as 강의여부 -- 선생의 seq를 입력  from (select distinct opencourseseq from tblopensubject where teacherseq = 4) toc inner join tblopencourse oc on toc.opencourseseq = oc.opencourseseq ; -- 선생이 진행하는 과정 조회  **- 교사가 강의하는 과정에 모든 교육생 출결 확인**  # 프로시저로 처리  **- 특정인원 출결 조회**  select  to\_char(checkout,'yyyy-mm-dd') as "날짜",  case when checkin is null then  '안옴'  else  to\_char(checkin,'yyyy-mm-dd hh24:mi')  end  as "입실시간",    case when checkin is null then  '안옴'  else  to\_char(checkout,'yyyy-mm-dd hh24:mi')  end  as "퇴실시간",  case when record is null and checkin is not null and to\_char(checkin,'hh24:mi') < '09:01' then '정상'  when to\_char(checkin,'hh24:mi') > '09:00' then '지각'  when checkin is null and checkout is null then record  when to\_char(checkout,'hh24:mi') < '14:00' then '조퇴'  else record end as "분류"  from (select \* from tblattendance where studentseq = 89) order by 날짜;  - **특정 인원의 출결 현황 조회 > 달 조회**  select  to\_char(checkout,'yyyy-mm-dd') as "날짜",  case when checkin is null then  '안옴'  else  to\_char(checkin,'yyyy-mm-dd hh24:mi')  end  as "입실시간",    case when checkin is null then  '안옴'  else  to\_char(checkout,'yyyy-mm-dd hh24:mi')  end  as "퇴실시간",  case when record is null and checkin is not null and to\_char(checkin,'hh24:mi') < '09:01' then '정상'  when to\_char(checkin,'hh24:mi') > '09:00' then '지각'  when checkin is null and checkout is null then record  when to\_char(checkout,'hh24:mi') < '14:00' then '조퇴'  else record end as "분류"  from (select \* from tblattendance where studentseq = 83) where checkout >= '2022-08-01' and checkout < '2022-09-01 ' order by 날짜; -- 달입력  **- 특정 인원의 출결 현황 조회 > 일 조회**  select to\_char(checkout,'yyyy-mm-dd') as "날짜", to\_char(checkin,'yyyy-mm-dd hh24:mi') as "입실시간", to\_char(checkout,'yyyy-mm-dd hh24:mi') as "퇴실시간",  case when record is null and checkin is not null and to\_char(checkin,'hh24:mi') < '09:01' then '정상'  when to\_char(checkin,'hh24:mi') > '09:00' then '지각'  when checkin is null and checkout is null then record  when to\_char(checkout,'hh24:mi') < '14:00' then '조퇴'  else record end as "분류"  from tblattendance where studentseq = 260 and to\_char(checkin,'yyyy-mm-dd') = '2023-04-04'; -- 요일입력 | | | |

| 요구사항번호 | D-01 | 작성일 | 2023-04-04 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 계정 | 작성자 | 남건욱 |
| DML | | | |
| 1. **교육생 로그인**   **- 교육생이 로그인 시 교육생 개인정보와 과정기간, 강의실 정보가 출력된다.**  select a.name "이름", a.tel "전화번호", c.coursetitle "과정이름", o.startdate "시간기간", o.enddate "종료기간" from tblStudent s  inner join tblApplicant a  on a.applicantseq = s.applicantseq and a.name = '이순옥' and a.ssn = '1066962' – 검색하려는 교육생의 이름과 주민번호  inner join tblopencourse o  on o.opencourseseq = s.opencourseseq  inner join tblcourse c  on c.courseseq = o.courseseq;   1. **성적 정보 출력**   **- 성적정보 출력 : 개설과목번호, 과목명, 시작년월일, 끝 년월일, 출결배점, 필기배점, 실기배점, 출석점수, 필기점수, 실기점수, 필기시험날짜, 실기시험날짜**    select  'SUB' || sc.opensubjectseq as 개설과목번호,  (select subjecttitle from tblsubject where subjectseq = ( select subjectseq from tblopensubject os where os.opensubjectseq = sc.opensubjectseq )) as 과목명,  (select startdate from tblopensubject where opensubjectseq = sc.opensubjectseq ) as 시작년월일,  (select startdate from tblopensubject where opensubjectseq = sc.opensubjectseq ) as 끝년월일,  (select b.booktitle from tblopensubject os inner join tblbook b on os.bookseq = b.bookseq where os.opensubjectseq = sc.opensubjectseq) as 교재명,  (select t.name from tblteacher t inner join tblopensubject os on t.teacherseq = os.teacherseq where os.opensubjectseq = sc.opensubjectseq) as 교사명,  -- 출결 배점  (select p.attendance from tblopensubject os inner join tblpoints p on os.pointsseq = p.pointsseq where os.opensubjectseq = sc.opensubjectseq) as 출결배점,  -- 필기 배점  (select p.written from tblopensubject os inner join tblpoints p on os.pointsseq = p.pointsseq where os.opensubjectseq = sc.opensubjectseq) as 필기배점,  -- 실기배점  (select p.practical from tblopensubject os inner join tblpoints p on os.pointsseq = p.pointsseq where os.opensubjectseq = sc.opensubjectseq) as 실기배점,  sc.attendancescore as 출석점수,  sc.writtenscore as 필기점수,  sc.practicalscore as 실기점수,  (select writtendate from tblExamRegister e where e.opensubjectseq = sc.opensubjectseq) as 필기시험날짜,  (select practicaldate from tblExamRegister e where e.opensubjectseq = sc.opensubjectseq) as 실기시험날짜  from (select \* from tblscore where studentseq = (select s.studentseq from tblapplicant a inner join tblstudent s on a.applicantseq = s.applicantseq and a.name='이순옥' and a.ssn='1066962' )) sc order by 시작년월일; -- 이름, 주민번호로 찾음   1. **출결 관리 및 출결 조회**   **- 매일 근태 관리를 기록할 수 있어야 한다. (**출근 1회, 퇴근 1회**)**  - 새로운 근태 기록을 추가 > 병가를 기록  insert into tblattendance values( (select nvl( max(attendanceseq), 0) + 1 from tblattendance),  ( select studentseq from tblstudent where applicantseq = (select applicantseq from tblapplicant where name ='홍민석' and ssn = '1541919')) ,  '병가', null , to\_date( to\_char(sysdate, 'yyyy-mm-dd')||'17:50', 'yyyy-mm-dd hh24 mi' )); -- default로 enddate에는 값을 넣음  - 새로운 근태 기록을 추가 > 정상출석인 경우  insert into tblattendance values( (select nvl( max(attendanceseq), 0) + 1 from tblattendance),  ( select studentseq from tblstudent where applicantseq = (select applicantseq from tblapplicant where name ='홍민석' and ssn = '1541919')) ,  null, to\_date( to\_char(vdate, 'yyyy-mm-dd')||'09:00', 'yyyy-mm-dd hh24 mi'), to\_date( to\_char(vdate, 'yyyy-mm-dd')||'17:50', 'yyyy-mm-dd hh24 mi'));  **- 본인의 출결 현황 조회 > 전체 출결 현황 조회**  select  to\_char(checkout,'yyyy-mm-dd') as "날짜",  case when checkin is null then  '안옴'  else  to\_char(checkin,'yyyy-mm-dd hh24:mi')  end  as "입실시간",    case when checkin is null then  '안옴'  else  to\_char(checkout,'yyyy-mm-dd hh24:mi')  end  as "퇴실시간",  case when record is null and checkin is not null and to\_char(checkin,'hh24:mi') < '09:01' then '정상'  when to\_char(checkin,'hh24:mi') > '09:00' then '지각'  when checkin is null and checkout is null then record  when to\_char(checkout,'hh24:mi') < '14:00' then '조퇴'  else record end as "분류"  from (select \* from tblattendance where studentseq = 89) order by 날짜;  - **본인의 출결 현황 조회 > 달 조회**  select  to\_char(checkout,'yyyy-mm-dd') as "날짜",  case when checkin is null then  '안옴'  else  to\_char(checkin,'yyyy-mm-dd hh24:mi')  end  as "입실시간",    case when checkin is null then  '안옴'  else  to\_char(checkout,'yyyy-mm-dd hh24:mi')  end  as "퇴실시간",  case when record is null and checkin is not null and to\_char(checkin,'hh24:mi') < '09:01' then '정상'  when to\_char(checkin,'hh24:mi') > '09:00' then '지각'  when checkin is null and checkout is null then record  when to\_char(checkout,'hh24:mi') < '14:00' then '조퇴'  else record end as "분류"  from (select \* from tblattendance where studentseq = 83) where checkout >= '2022-08-01' and checkout < '2022-09-01 ' order by 날짜;  **- 본인의 출결 현황 조회 > 일 조회**  select to\_char(checkout,'yyyy-mm-dd') as "날짜", to\_char(checkin,'yyyy-mm-dd hh24:mi') as "입실시간", to\_char(checkout,'yyyy-mm-dd hh24:mi') as "퇴실시간",  case when record is null and checkin is not null and to\_char(checkin,'hh24:mi') < '09:01' then '정상'  when to\_char(checkin,'hh24:mi') > '09:00' then '지각'  when checkin is null and checkout is null then record  when to\_char(checkout,'hh24:mi') < '14:00' then '조퇴'  else record end as "분류"  from tblattendance where studentseq = 260 and to\_char(checkin,'yyyy-mm-dd') = '2023-04-04'; | | | |

| 요구사항번호 | D-03 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 스터디 신청 및 조회 (추가요구사항) | 작성자 | 오소엽 |
| DML | | | |
| -- 스터디 신청 및 관리 (교육생, 관리자)  -- 교육생  -- 스터디 신청(프로시저 procInsertStudy 실행하여 데이터 추가할 수도 있다.)  insert into tblStudy(studySeq, studentSeq, studyTitle, registerDate, studyDate, capacity)  values ( (select max(study.studySeq)+1 from tblStudy study), **<교육생seq>**, ‘**<스터디 주제>**’, ‘**<신청날짜>**’, ‘**<스터디날짜>**’, **<정원>**);  -- 관리자  -- 스터디 목록 전체 조회 (=> vwStudy로 조회할 수도 있다.)  select  study.studySeq as 스터디번호,  stdn.studentSeq as 교육생번호,  a.name as 이름,  opc.courseSeq as 수강과정,  study.studyTitle as 스터디내용,  study.registerDate as 신청날짜,  study.studyDate as 스터디날짜,  study.capacity || '명' as 정원  from tblStudent stdn  inner join tblStudy study  on stdn.studentSeq = study.studentSeq  inner join tblApplicant a  on stdn.applicantSeq = a.applicantSeq  inner join tblOpenCourse opc  on stdn.openCourseSeq = opc.openCourseSeq  order by study.studySeq, study.registerDate, study.studyDate;  -- 특정 교육생의 스터디 내역 조회  select  study.studySeq as 스터디번호,  stdn.studentSeq as 교육생번호,  a.name as 이름,  opc.courseSeq as 수강과정,  study.studyTitle as 스터디내용,  study.registerDate as 신청날짜,  study.studyDate as 스터디날짜,  study.capacity || '명' as 정원  from tblStudent stdn  inner join tblStudy study  on stdn.studentSeq = study.studentSeq  inner join tblApplicant a  on stdn.applicantSeq = a.applicantSeq  inner join tblOpenCourse opc  on stdn.openCourseSeq = opc.openCourseSeq  where stdn.studentSeq = **<교육생seq>**;  -- 기간별 스터디 내역 조회  select  study.studySeq as 스터디번호,  stdn.studentSeq as 교육생번호,  a.name as 이름,  opc.courseSeq as 수강과정,  study.studyTitle as 스터디내용,  study.registerDate as 신청날짜,  study.studyDate as 스터디날짜,  study.capacity || '명' as 정원  from tblStudent stdn  inner join tblStudy study  on stdn.studentSeq = study.studentSeq  inner join tblApplicant a  on stdn.applicantSeq = a.applicantSeq  inner join tblOpenCourse opc  on stdn.openCourseSeq = opc.openCourseSeq  where study.registerDate between '**<신청날짜 or 스터디날짜>**' and '**<신청날짜 or 스터디날짜>**'  order by study.studySeq, study.registerDate, study.studyDate;  -- 스터디 주제별 조회  select  study.studySeq as 스터디번호,  stdn.studentSeq as 교육생번호,  a.name as 이름,  opc.courseSeq as 수강과정,  study.studyTitle as 스터디내용,  study.registerDate as 신청날짜,  study.studyDate as 스터디날짜,  study.capacity || '명' as 정원  from tblStudent stdn  inner join tblStudy study  on stdn.studentSeq = study.studentSeq  inner join tblApplicant a  on stdn.applicantSeq = a.applicantSeq  inner join tblOpenCourse opc  on stdn.openCourseSeq = opc.openCourseSeq  where study.studyTitle like '%**<검색할 스터디주제>**%'  order by study.studySeq, study.registerDate, study.studyDate; | | | |

| 요구사항번호 | D-04 | 작성일 | 2023-04-05 |
| --- | --- | --- | --- |
| 요구사항명 | 취업히스토리 작성 및 조회 (추가요구사항) | 작성자 | 오소엽 |
| DML | | | |
| -- 취업 히스토리 등록  insert into tblJobHistory(jobHistorySeq, studentSeq, job, companyName, companySize, review, pass, applyDate)  values ((select max(jobHistorySeq)+1 from tblJobHistory), **<학생번호>**, ‘**<직무>’**, ‘**<기업명>**’, ‘**<기업규모>**’, ‘**<지원후기>**’, ‘**<합격불합격>**’, ‘**<지원날짜>**’);  -- 취업 히스토리 수정  update tblJobHistory set <변경할 컬럼> = <변경할 내용> where <변경할 조건>;  -- 수료생의 취업히스토리 전체 목록 조회 (vwJobHistory로 조회할 수도 있다.)  select  stdn.studentSeq as studentSeq,  substr(a.name,1,1) || lpad('\*', length(a.name)-2, '\*') ||  substr(a.name, length(a.name), 1) as name,  jh.applyDate as applyDate,  case  when jr.hireDate is null then '미취업'  when jr.hireDate is not null then to\_char(jr.hireDate)  end as hireDate,  jh.companyName as company,  jh.job as job,  jh.companySize as companySize,  jh.review as review,  jh.pass as pass,  case  when jr.satisfaction is not null then jr.satisfaction || '점'  when jr.satisfaction is null then '-'  end as satisfaction,  spec.education as education,  spec.major as major,  case  when cf.certificateName is not null then cf.certificateName  when cf.certificateName is null then '없음'  end as certificate,  c.courseTitle as courseTitle,  case  when  stdn.completeOrFail is null and sysdate > opc.startDate  and sysdate < stdn.completeOrFailDate then '수강중'  when stdn.completeOrFail = 1  and sysdate > opc.endDate then stdn.completeOrFailDate ||' (수료)'  when stdn.completeOrFail = 0 then stdn.completeOrFailDate || ' (중도탈락)'  when stdn.completeOrFail is null and sysdate < opc.startDate then '개강전'  end as completeOrFail  from tblStudent stdn  inner join tblSpec spec  on stdn.studentSeq = spec.studentSeq  inner join tblApplicant a  on a.applicantSeq = stdn.applicantSeq  inner join tblOpenCourse opc  on stdn.openCourseSeq = opc.openCourseSeq  inner join tblCourse c  on opc.courseSeq = c.courseSeq  inner join tblJobHistory jh  on stdn.studentSeq = jh.studentSeq  left outer join tblJobResult jr  on stdn.studentSeq = jr.studentSeq  left outer join tblCerificate cf  on spec.specSeq = cf.specSeq  order by applyDate, hireDate, studentSeq, completeOrFailDate;  -- 특정 교육생의 취업 히스토리 조회  select  stdn.studentSeq as studentSeq,  substr(a.name,1,1) || lpad('\*', length(a.name)-2, '\*') ||  substr(a.name, length(a.name), 1) as name,  jh.applyDate as applyDate,  case  when jr.hireDate is null then '미취업'  when jr.hireDate is not null then to\_char(jr.hireDate)  end as hireDate,  jh.companyName as company,  jh.job as job,  jh.companySize as companySize,  jh.review as review,  jh.pass as pass,  case  when jr.satisfaction is not null then jr.satisfaction || '점'  when jr.satisfaction is null then '-'  end as satisfaction,  spec.education as education,  spec.major as major,  case  when cf.certificateName is not null then cf.certificateName  when cf.certificateName is null then '없음'  end as certificate,  c.courseTitle as courseTitle,  case  when  stdn.completeOrFail is null and sysdate > opc.startDate  and sysdate < stdn.completeOrFailDate then '수강중'  when stdn.completeOrFail = 1  and sysdate > opc.endDate then stdn.completeOrFailDate ||' (수료)'  when stdn.completeOrFail = 0 then stdn.completeOrFailDate || ' (중도탈락)'  when stdn.completeOrFail is null and sysdate < opc.startDate then '개강전'  end as completeOrFail  from tblStudent stdn  inner join tblSpec spec  on stdn.studentSeq = spec.studentSeq  inner join tblApplicant a  on a.applicantSeq = stdn.applicantSeq  inner join tblOpenCourse opc  on stdn.openCourseSeq = opc.openCourseSeq  inner join tblCourse c  on opc.courseSeq = c.courseSeq  inner join tblJobHistory jh  on stdn.studentSeq = jh.studentSeq  left outer join tblJobResult jr  on stdn.studentSeq = jr.studentSeq  left outer join tblCerificate cf  on spec.specSeq = cf.specSeq  where stdn.studentSeq between **<교육생seq>** and **<교육생seq>**  order by studentSeq;  -- 특정 기간 내 취업 히스토리 조회  select  stdn.studentSeq as studentSeq,  substr(a.name,1,1) || lpad('\*', length(a.name)-2, '\*') ||  substr(a.name, length(a.name), 1) as name,  jh.applyDate as applyDate,  case  when jr.hireDate is null then '미취업'  when jr.hireDate is not null then to\_char(jr.hireDate)  end as hireDate,  jh.companyName as company,  jh.job as job,  jh.companySize as companySize,  jh.review as review,  jh.pass as pass,  case  when jr.satisfaction is not null then jr.satisfaction || '점'  when jr.satisfaction is null then '-'  end as satisfaction,  spec.education as education,  spec.major as major,  case  when cf.certificateName is not null then cf.certificateName  when cf.certificateName is null then '없음'  end as certificate,  c.courseTitle as courseTitle,  case  when  stdn.completeOrFail is null and sysdate > opc.startDate  and sysdate < stdn.completeOrFailDate then '수강중'  when stdn.completeOrFail = 1  and sysdate > opc.endDate then stdn.completeOrFailDate ||' (수료)'  when stdn.completeOrFail = 0 then stdn.completeOrFailDate || ' (중도탈락)'  when stdn.completeOrFail is null and sysdate < opc.startDate then '개강전'  end as completeOrFail  from tblStudent stdn  inner join tblSpec spec  on stdn.studentSeq = spec.studentSeq  inner join tblApplicant a  on a.applicantSeq = stdn.applicantSeq  inner join tblOpenCourse opc  on stdn.openCourseSeq = opc.openCourseSeq  inner join tblCourse c  on opc.courseSeq = c.courseSeq  inner join tblJobHistory jh  on stdn.studentSeq = jh.studentSeq  left outer join tblJobResult jr  on stdn.studentSeq = jr.studentSeq  left outer join tblCerificate cf  on spec.specSeq = cf.specSeq  where jh.applyDate between '**<검색날짜>**' and '**<검색날짜>**'  order by applyDate, hireDate, studentSeq, completeOrFailDate;  -- 특정 기업에 지원한 내역 조회  select  stdn.studentSeq as studentSeq,  substr(a.name,1,1) || lpad('\*', length(a.name)-2, '\*') ||  substr(a.name, length(a.name), 1) as name,  jh.applyDate as applyDate,  case  when jr.hireDate is null then '미취업'  when jr.hireDate is not null then to\_char(jr.hireDate)  end as hireDate,  jh.companyName as company,  jh.job as job,  jh.companySize as companySize,  jh.review as review,  jh.pass as pass,  case  when jr.satisfaction is not null then jr.satisfaction || '점'  when jr.satisfaction is null then '-'  end as satisfaction,  spec.education as education,  spec.major as major,  case  when cf.certificateName is not null then cf.certificateName  when cf.certificateName is null then '없음'  end as certificate,  c.courseTitle as courseTitle,  case  when  stdn.completeOrFail is null and sysdate > opc.startDate  and sysdate < stdn.completeOrFailDate then '수강중'  when stdn.completeOrFail = 1  and sysdate > opc.endDate then stdn.completeOrFailDate ||' (수료)'  when stdn.completeOrFail = 0 then stdn.completeOrFailDate || ' (중도탈락)'  when stdn.completeOrFail is null and sysdate < opc.startDate then '개강전'  end as completeOrFail  from tblStudent stdn  inner join tblSpec spec  on stdn.studentSeq = spec.studentSeq  inner join tblApplicant a  on a.applicantSeq = stdn.applicantSeq  inner join tblOpenCourse opc  on stdn.openCourseSeq = opc.openCourseSeq  inner join tblCourse c  on opc.courseSeq = c.courseSeq  inner join tblJobHistory jh  on stdn.studentSeq = jh.studentSeq  left outer join tblJobResult jr  on stdn.studentSeq = jr.studentSeq  left outer join tblCerificate cf  on spec.specSeq = cf.specSeq  where jh.companyName in like ‘%**<검색할 기업명>**%’  order by applyDate, hireDate, studentSeq, completeOrFailDate;  -- 특정 직무에 지원한 내역 조회  select  stdn.studentSeq as studentSeq,  substr(a.name,1,1) || lpad('\*', length(a.name)-2, '\*') ||  substr(a.name, length(a.name), 1) as name,  jh.applyDate as applyDate,  case  when jr.hireDate is null then '미취업'  when jr.hireDate is not null then to\_char(jr.hireDate)  end as hireDate,  jh.companyName as company,  jh.job as job,  jh.companySize as companySize,  jh.review as review,  jh.pass as pass,  case  when jr.satisfaction is not null then jr.satisfaction || '점'  when jr.satisfaction is null then '-'  end as satisfaction,  spec.education as education,  spec.major as major,  case  when cf.certificateNAme is not null then cf.certificateNAme  when cf.certificateNAme is null then '없음'  end as certificate,  c.courseTitle as courseTitle,  case  when  stdn.completeOrFail is null and sysdate > opc.startDate  and sysdate < stdn.completeOrFailDate then '수강중'  when stdn.completeOrFail = 1  and sysdate > opc.endDate then stdn.completeOrFailDate ||' (수료)'  when stdn.completeOrFail = 0 then stdn.completeOrFailDate || ' (중도탈락)'  when stdn.completeOrFail is null and sysdate < opc.startDate then '개강전'  end as completeOrFail  from tblStudent stdn  inner join tblSpec spec  on stdn.studentSeq = spec.studentSeq  inner join tblApplicant a  on a.applicantSeq = stdn.applicantSeq  inner join tblOpenCourse opc  on stdn.openCourseSeq = opc.openCourseSeq  inner join tblCourse c  on opc.courseSeq = c.courseSeq  inner join tblJobHistory jh  on stdn.studentSeq = jh.studentSeq  left outer join tblJobResult jr  on stdn.studentSeq = jr.studentSeq  left outer join tblCerificate cf  on spec.specSeq = cf.specSeq  where jh.job like '%**<검색할 직무>**%'  order by applyDate, hireDate, studentSeq, completeOrFailDate;  -- 불합격 히스토리 조회  select  stdn.studentSeq as studentSeq,  substr(a.name,1,1) || lpad('\*', length(a.name)-2, '\*') ||  substr(a.name, length(a.name), 1) as name,  jh.applyDate as applyDate,  case  when jr.hireDate is null then '미취업'  when jr.hireDate is not null then to\_char(jr.hireDate)  end as hireDate,  jh.companyName as company,  jh.job as job,  jh.companySize as companySize,  jh.review as review,  jh.pass as pass,  case  when jr.satisfaction is not null then jr.satisfaction || '점'  when jr.satisfaction is null then '-'  end as satisfaction,  spec.education as education,  spec.major as major,  case  when cf.certificateNAme is not null then cf.certificateNAme  when cf.certificateNAme is null then '없음'  end as certificate,  c.courseTitle as courseTitle,  case  when  stdn.completeOrFail is null and sysdate > opc.startDate  and sysdate < stdn.completeOrFailDate then '수강중'  when stdn.completeOrFail = 1  and sysdate > opc.endDate then stdn.completeOrFailDate ||' (수료)'  when stdn.completeOrFail = 0 then stdn.completeOrFailDate || ' (중도탈락)'  when stdn.completeOrFail is null and sysdate < opc.startDate then '개강전'  end as completeOrFail  from tblStudent stdn  inner join tblSpec spec  on stdn.studentSeq = spec.studentSeq  inner join tblApplicant a  on a.applicantSeq = stdn.applicantSeq  inner join tblOpenCourse opc  on stdn.openCourseSeq = opc.openCourseSeq  inner join tblCourse c  on opc.courseSeq = c.courseSeq  inner join tblJobHistory jh  on stdn.studentSeq = jh.studentSeq  left outer join tblJobResult jr  on stdn.studentSeq = jr.studentSeq  left outer join tblCerificate cf  on spec.specSeq = cf.specSeq  where jh.pass = 'fail'  order by applyDate, hireDate, studentSeq, completeOrFailDate;  ----------------------------------------------------------------------------  -- [취업 히스토리 관련 통계]  -- 직무별 지원자와 지원자리뷰 현황  select  job.job as 직무,  job.review as 지원자리뷰,  count(\*) as 리뷰수  from (  select  stdn.studentSeq as studentSeq,  substr(a.name,1,1) || lpad('\*', length(a.name)-2, '\*') ||  substr(a.name, length(a.name), 1) as name,  jh.applyDate as applyDate,  case  when jr.hireDate is null then '미취업'  when jr.hireDate is not null then to\_char(jr.hireDate)  end as hireDate,  jh.companyName as company,  jh.job as job,  jh.companySize as companySize,  jh.review as review,  case  when jr.satisfaction is not null then jr.satisfaction || '점'  when jr.satisfaction is null then '-'  end as satisfaction,  spec.education as education,  spec.major as major,  case  when cf.certificateNAme is not null then cf.certificateNAme  when cf.certificateNAme is null then '없음'  end as certificate,  c.courseTitle as courseTitle,  case  when  stdn.completeOrFail is null and sysdate > opc.startDate  and sysdate < stdn.completeOrFailDate then '수강중'  when stdn.completeOrFail = 1  and sysdate > opc.endDate then stdn.completeOrFailDate ||' (수료)'  when stdn.completeOrFail = 0 then stdn.completeOrFailDate || ' (중도탈락)'  when stdn.completeOrFail is null and sysdate < opc.startDate then '개강전'  end as completeOrFail  from tblStudent stdn  inner join tblSpec spec  on stdn.studentSeq = spec.studentSeq  inner join tblApplicant a  on a.applicantSeq = stdn.applicantSeq  inner join tblOpenCourse opc  on stdn.openCourseSeq = opc.openCourseSeq  inner join tblCourse c  on opc.courseSeq = c.courseSeq  inner join tblJobHistory jh  on stdn.studentSeq = jh.studentSeq  left outer join tblJobResult jr  on stdn.studentSeq = jr.studentSeq  left outer join tblCerificate cf  on spec.specSeq = cf.specSeq  order by applyDate, hireDate, studentSeq, completeOrFailDate) job  where job.job like **'%<검색할 직무>%'**  group by job.job, job.review  order by job.job, job.review desc;  -- 기업별 지원자 리뷰 통계 조회  select  job.company as 기업명,  job.review as 지원자리뷰,  count(\*) as 리뷰수  from (  select  stdn.studentSeq as studentSeq,  substr(a.name,1,1) || lpad('\*', length(a.name)-2, '\*') ||  substr(a.name, length(a.name), 1) as name,  jh.applyDate as applyDate,  case  when jr.hireDate is null then '미취업'  when jr.hireDate is not null then to\_char(jr.hireDate)  end as hireDate,  jh.companyName as company,  jh.job as job,  jh.companySize as companySize,  jh.review as review,  case  when jr.satisfaction is not null then jr.satisfaction || '점'  when jr.satisfaction is null then '-'  end as satisfaction,  spec.education as education,  spec.major as major,  case  when cf.certificateNAme is not null then cf.certificateNAme  when cf.certificateNAme is null then '없음'  end as certificate,  c.courseTitle as courseTitle,  case  when  stdn.completeOrFail is null and sysdate > opc.startDate  and sysdate < stdn.completeOrFailDate then '수강중'  when stdn.completeOrFail = 1  and sysdate > opc.endDate then stdn.completeOrFailDate ||' (수료)'  when stdn.completeOrFail = 0 then stdn.completeOrFailDate || ' (중도탈락)'  when stdn.completeOrFail is null and sysdate < opc.startDate then '개강전'  end as completeOrFail  from tblStudent stdn  inner join tblSpec spec  on stdn.studentSeq = spec.studentSeq  inner join tblApplicant a  on a.applicantSeq = stdn.applicantSeq  inner join tblOpenCourse opc  on stdn.openCourseSeq = opc.openCourseSeq  inner join tblCourse c  on opc.courseSeq = c.courseSeq  inner join tblJobHistory jh  on stdn.studentSeq = jh.studentSeq  left outer join tblJobResult jr  on stdn.studentSeq = jr.studentSeq  left outer join tblCerificate cf  on spec.specSeq = cf.specSeq  order by applyDate, hireDate, studentSeq, completeOrFailDate) job  where job.company like **'%<검색할 기업명>%'**  group by job.company, job.review  order by job.company, job.review desc;  -- 기업규모별 지원자리뷰 통계 조회  select  job.companySize as 기업규모,  job.review as 지원자리뷰,  count(\*) as 리뷰수  from (  select  stdn.studentSeq as studentSeq,  substr(a.name,1,1) || lpad('\*', length(a.name)-2, '\*') ||  substr(a.name, length(a.name), 1) as name,  jh.applyDate as applyDate,  case  when jr.hireDate is null then '미취업'  when jr.hireDate is not null then to\_char(jr.hireDate)  end as hireDate,  jh.companyName as company,  jh.job as job,  jh.companySize as companySize,  jh.review as review,  case  when jr.satisfaction is not null then jr.satisfaction || '점'  when jr.satisfaction is null then '-'  end as satisfaction,  spec.education as education,  spec.major as major,  case  when cf.certificateNAme is not null then cf.certificateNAme  when cf.certificateNAme is null then '없음'  end as certificate,  c.courseTitle as courseTitle,  case  when  stdn.completeOrFail is null and sysdate > opc.startDate  and sysdate < stdn.completeOrFailDate then '수강중'  when stdn.completeOrFail = 1  and sysdate > opc.endDate then stdn.completeOrFailDate ||' (수료)'  when stdn.completeOrFail = 0 then stdn.completeOrFailDate || ' (중도탈락)'  when stdn.completeOrFail is null and sysdate < opc.startDate then '개강전'  end as completeOrFail  from tblStudent stdn  inner join tblSpec spec  on stdn.studentSeq = spec.studentSeq  inner join tblApplicant a  on a.applicantSeq = stdn.applicantSeq  inner join tblOpenCourse opc  on stdn.openCourseSeq = opc.openCourseSeq  inner join tblCourse c  on opc.courseSeq = c.courseSeq  inner join tblJobHistory jh  on stdn.studentSeq = jh.studentSeq  left outer join tblJobResult jr  on stdn.studentSeq = jr.studentSeq  left outer join tblCerificate cf  on spec.specSeq = cf.specSeq  order by applyDate, hireDate, studentSeq, completeOrFailDate) job  where job.companySize in **('<중소, 중견, 스타트업, 대기업 중 원하는 항목 검색>')**  group by job.companySize, job.review  order by job.companySize, job.review desc; | | | |