**1조 데이터 쿼리문(ANSI-SQL)**

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-01 | 작성일 | 2022-05-18 |
| 요구사항명 | 관리자 계정 | 작성자 | 김성태 |
| DML | | | |
| 계정으로 로그인 후 관리자 기본정보 간단하게 출력  select  admin.seq,  admin.name as "이름",  admin.tel as "연락처"  from admin  where ssn = 1068450; --로그인 한 관리자의 주민번호 뒷자리(패스워드) | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-02 | 작성일 | 2022-05-18 |
| 요구사항명 | 기초정보 관리기능 | 작성자 | 김성태 |
| DML | | | |
| --과정 명 관리  select \* from curriculum;  insert into curriculum values (21, '이클립스 완전정복');  update curriculum set name = '오라클 완전정복' where seq = 21;  delete from curriculum where seq = 21;  --과목 명 관리  select \* from subject;  insert into subject values (31, 'HTML');  update subject set subject\_name = CSS where seq = 31;  delete from subject where seq = 31;  --강의실 명 관리  select \* from lectureroom;  insert into lectureroom values (7, '7강의실', 25);  update lectureroom set room\_name = '상담실' where seq = 7;  delete from lectureroom where seq = 7;  --교재 명 관리  select \* from book;  insert into book values (41, 'HTML5 웹 프로그래밍 입문', '윤인성', '한빛아카데미', '2019-07-20');  update book set publisher = '정문출판' where seq = 41;  delete from book where seq = 41; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-03 | 작성일 | 2022-05-18 |
| 요구사항명 | 교사계정 관리기능 | 작성자 | 김성태 |
| DML | | | |
| --교사정보 등록  insert into teacher values (11, '홍길동', 1010101, '010-1111-1111');  insert into teaching\_subject values (11, 1);  --교사정보 수정  update teacher set name = '김세종' where name = '홍길동';  --교사정보 삭제  delete from teaching\_subject where teacher\_seq = 11;  delete from teacher where name = '김세종';  --교사정보 조회  select  t.name as "이름",  t.ssn as "주민번호",  t.tel as "연락처",  s.subject\_name as "강의가능 과목"  from teacher t  inner join teaching\_subject ts  on t.seq = ts.teacher\_seq  inner join subject s  on ts.subject\_seq = s.seq  order by t.name asc;  --교사의 강의과목에 대한 정보 조회  select  t.name as "이름",  ts.subject\_seq as "과목번호",  s.subject\_name as "과목명",  os.begin\_date as "과목 시작",  os.end\_date as "과목 종료",  c.name as "과정명",  oc.begin\_date as "과정 시작",  oc.end\_date as "과정 종료",  b.subject as "교재명",  l.room\_name as "강의실",    case  when oc.begin\_date > sysdate then '강의 예정'  when oc.end\_date < sysdate then '강의 종료'  else '강의 중'  end as "진행 상태"  from teacher t  inner join teaching\_subject ts  on t.seq = ts.teacher\_seq  inner join subject s  on ts.subject\_seq = s.seq  inner join open\_subject os  on t.seq = os.teacher\_seq  inner join open\_curs oc  on os.open\_curs\_seq = oc.seq  inner join curriculum c  on oc.curs\_seq = c.seq  inner join subject\_book sb  on s.seq = sb.subject\_seq  inner join book b  on sb.book\_seq = b.seq  inner join lectureroom l  on oc.room\_seq = l.seq  where t.name = '풍종진'; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-04 | 작성일 | 2022-05-19 |
| 요구사항명 | 개설과정 관리기능 | 작성자 | 임종운 |
| DML | | | |
| **<>안에 CRUD할 대상의 정보를 입력**  **1. 과정명 추가**  INSERT INTO curriculum(seq, name) VALUES(curriculum\_seq.nextval, **<과정명>**);  **2. 과정명 및 강의실 정보 조회**  SELECT \* FROM curriculum; --과정명 조회  SELECT \* FROM lectureroom; -- 강의실 조회  **3. 과정 정보 입력 – 과정명 및 강의실명은 테이블에서 선택하여 입력**  INSERT INTO open\_curs(seq, curs\_seq, room\_seq, begin\_date, end\_date, current\_term, student\_limit, student\_num)              VALUES(**<과정 일련번호>, <과정명>, <강의실명>, <과정 시작일자>, <과정 종료일자>, <과정 개월입력>, <학생 정원>, <학생  수>**);  **4. 개설 과정 정보 출력시 개설 과정명, 개설 과정기간, 강의실명, 개설 과목 등록 여부, 교육생 등록 인원 출력**  SELECT      oc.seq AS 개설과정번호,      cc.name AS 과정명,      oc.begin\_date AS 시작일자,      oc.end\_date AS 종료일자,      l.room\_name AS 강의실명,      oc.student\_num AS 교육생등록인원,      --이게 사실 필요가 없는 컬럼인것같음      p.등록인원  FROM open\_curs oc      INNER JOIN curriculum cc ON oc.curs\_seq = cc.seq      INNER JOIN lectureroom l ON oc.room\_seq = l.seq      INNER JOIN (SELECT          open\_curs\_seq, count(member\_seq) AS 등록인원      FROM enrollment      GROUP BY open\_curs\_seq) p ON oc.seq = p.open\_curs\_seq  ORDER BY oc.seq;  **5. 특정 개설 과정 선택시 개설 과정에 등록된 개설 과목 정보 확인**  SELECT      s.subject\_name, os.begin\_date, os.end\_date, b.subject ,t.name  FROM (SELECT seq  FROM open\_curs oc WHERE seq = **<확인할 개설 과정번호>**) sr      INNER JOIN open\_subject os ON sr.seq = os.open\_curs\_seq      INNER JOIN subject s ON os.subject\_seq = s.seq      INNER JOIN subject\_book sb ON s.seq = sb.subject\_seq      INNER JOIN book b ON sb.book\_seq = b.seq      INNER JOIN teacher t ON os.teacher\_seq = t.seq;  **6. 과정에 등록된 교육생 정보 확인**  SELECT      m.name, m.ssn, m.tel, e.enroll\_date, s.drop\_out  FROM (SELECT \*      FROM open\_curs oc      WHERE seq = **<과정번호>**) rs      INNER JOIN enrollment e ON rs.seq = e.open\_curs\_seq      INNER JOIN member m ON e.member\_seq = m.seq      INNER JOIN student s ON m.seq = s.member\_seq;  **7. 특정 개설과정 수료시, 교육생 전체에 대해서 수료 날짜를 지정할 수 있어야 한다. 단 중도 탈락자는 제외한다.**  UPDATE student s SET s.drop\_out = '수료', s.drop\_out\_date = **<수료 날짜 지정>**  WHERE      EXISTS(          SELECT \*          FROM member m          WHERE s.member\_seq = m.seq          AND s.open\_curs\_seq = 12            --수료 완료 과정      )      AND      (s.drop\_out IS NULL      OR      s.drop\_out<>'중도탈락');  **8. 개설 과정 삭제**  **CASCADE** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-05 | 작성일 | 2022-05-19 |
| 요구사항명 | 개설과목 관리기능 | 작성자 | 임종운 |
| DML | | | |
| **<>안에 CRUD할 대상의 정보를 입력**  **1. 특정 개설 과정 선택 시 개설 과목 정보 출력**  SELECT      os.seq,      c.name AS 개설과정,      oc.begin\_date||'~'||oc.end\_date AS 과정진행일자,      s.subject\_name AS 과목명,      os.begin\_date||'~'||os.end\_date AS 과목기간,      b.subject AS 교재명,      t.name AS 담당교사  FROM open\_curs oc      INNER JOIN curriculum c ON oc.curs\_seq = c.seq      INNER JOIN open\_subject os ON oc.seq = os.open\_curs\_seq      INNER JOIN teacher t ON os.teacher\_seq = t.seq      INNER JOIN subject s ON os.subject\_seq = s.seq      INNER JOIN subject\_book sb ON s.seq = sb.subject\_seq      INNER JOIN book b ON sb.book\_seq = b.seq  WHERE oc.seq = **<개설과정 일련번호>**; --특정 개설과정 선택  **2. 개설과목 입력**  **2.1 과목명 목록**  SELECT \* FROM subject;  **2.2 과목 강의가능 강사 검색**  SELECT t.seq, t.name, t.ssn, t.tel, s.subject\_name  FROM teacher t      INNER JOIN teaching\_subject ts ON t.seq = ts.teacher\_seq      INNER JOIN subject s ON s.seq = ts.subject\_seq  WHERE s.subject\_name LIKE '**%<과목명 입력>%**';  **2.3 개설과목 입력**  INSERT INTO open\_subject(seq, subject\_seq, teacher\_seq, open\_curs\_seq, begin\_date, end\_date)              VALUES(open\_subject\_seq.nextval, <과목 일련번호>, <교수 일련번호>, <과정 일련번호>, <과정 시작기간>, <과정종료기간>);  **3. 개설과목 수정**  UPDATE open\_subject SET <수정할 대상> WHERE seq = <수정할 과목번호>;  **4. 개설과목 삭제**  **CASCADE** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-06 | 작성일 | 2022-05-19 |
| 요구사항명 | 교육생 관리 기능 | 작성자 | 임종운 |
| DML | | | |
| **<>안에 CRUD할 대상의 정보를 입력**  **1. 교육생 정보 등록**  INSERT INTO member(seq, name, ssn, tel) VALUES(member\_seq.nextVal, **<교육생이름>, <교육생주민번호뒷자리>, <교육생전화번호>**);  **2. 교육생 정보 출력**  SELECT      m.seq, m.name, m.ssn, m.tel,      s.regdate ,rs.수강신청횟수  FROM student s      INNER JOIN member m ON s.member\_seq = m.seq      INNER JOIN (SELECT          en.member\_seq,          count(en.seq) AS 수강신청횟수      FROM enrollment en          INNER JOIN member m ON en.member\_seq = m.seq      GROUP BY en.member\_seq) rs ON rs.member\_seq = m.seq  WHERE m.name LIKE('%%');    --교육생 검색기능 -> 이름 아직 추가는 미정  **3. 특정 교육생 선택 시 교육생이 신청한 과정명, 과정기간, 강의실, 수료여부 및 날짜 출력**  SELECT      cc.name AS 과정명,      oc.begin\_date||'~'||oc.end\_date AS 과정기간,      lr.room\_name AS 강의실,      CASE          WHEN              s.drop\_out IS NOT NULL          THEN s.drop\_out          WHEN              s.drop\_out IS NULL AND              TO\_DATE(s.regdate, 'yyyy-mm-dd')              BETWEEN TO\_DATE(oc.begin\_date, 'yyyy-mm-dd') AND TO\_DATE(oc.end\_date, 'yyyy-mm-dd')          THEN '수강 중'          WHEN              s.drop\_out IS NULL AND              TO\_DATE(s.regdate, 'yyyy-mm-dd') < TO\_DATE(oc.begin\_date, 'yyyy-mm-dd')          THEN '수강 예정'      END AS 수강상태,      CASE          WHEN              s.drop\_out\_date IS NOT NULL          THEN TO\_CHAR(s.drop\_out\_date)          WHEN              s.drop\_out\_date IS NULL          THEN '미등록'      END AS 종료날짜  FROM student s      INNER JOIN open\_curs oc ON s.open\_curs\_seq = oc.seq      INNER JOIN lectureroom lr ON oc.room\_seq = lr.seq      INNER JOIN curriculum cc ON oc.curs\_seq = cc.seq  WHERE s.member\_seq = <교육생일련번호>;  **4. 교육생 수료정보 수정**  UPDATE student SET drop\_out = **<변경할 상태>**, drop\_out\_date = **<상태를 변경한날짜>**  WHERE seq = (SELECT m.seq FROM member m WHERE m.seq = **<수정할 교육생 일련번호>)**;  **5. 강의 종료된 과정 목록 -> 선택과정의 교육생 정보 확인(취업 현황)**  **5.1강의 종료 과정목록**  SELECT      oc.seq, c.name  FROM open\_curs oc      INNER JOIN curriculum c ON oc.curs\_seq = c.seq  WHERE end\_date < sysdate;  **5.2선택 과정 교육생 정보확인**  SELECT      m.name, m.tel, jc.company, jc.hire\_date  FROM student s      INNER JOIN job\_current jc ON s.seq = jc.student\_seq      INNER JOIN member m ON s.member\_seq = m.seq  WHERE open\_curs\_seq = **<과정목록 일련번호>**;  **6. 교육생 정보 삭제**  **CASCADE** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-07 | 작성일 | 2022-05-18 |
| 요구사항명 | 시험 관리 및 성적 조회 기능 | 작성자 | 강지윤 |
| DML | | | |
| ▶ where절 <> 위치에는 <>안에 있는 내용에 해당하는 데이터 입력  **–1.특정 개설 과정을 선택하면 개설 과목 정보 출력 (개설 과목명, 과목시작일, 과목종료일, 개설과정명, 강의실명, 교사명, 교재명)**  --where절에서 원하는 개설과정명을 입력하면 됨.  select      distinct subject\_name as 개설과목,      open\_subject.begin\_date as 시작일,      open\_subject.end\_date as 종료일,     curriculum.name as 개설과정명,     lectureroom.room\_name as 강의실명,     teacher.name as 교사명,     book.subject as 교재명  from open\_curs      inner join open\_subject on open\_subject.open\_curs\_seq = open\_curs.seq      inner join curriculum on open\_curs.curs\_seq = curriculum.seq      inner join subject on subject.seq = open\_subject.subject\_seq      inner join lectureroom on open\_curs.room\_seq = lectureroom.seq      inner join teacher on open\_subject.teacher\_seq = teacher.seq      inner join subject\_book on subject\_book.subject\_seq = subject.seq      inner join book on subject\_book.book\_seq = book.seq      where curriculum.name = <개설과정명>      order by curriculum.name;  **—2.개설 과목별(수강종료, 수강중인 개설과목만)로 시험 문제 파일 등록 여부(필기,실기),  성적 등록 여부(필기,실기,출결)확인**  --where절에서 원하는 개설과목명을 입력하면 됨.  select       distinct subject.subject\_name as 과목명,       open\_subject.seq as 개설과목번호,       open\_subject.subject\_seq as 과목번호,       open\_curs.seq as 개설과정번호,       curriculum.name as 개설과정명,      (case when r.written\_test\_question = '시험문제등록안함' then 'N' else 'Y' end) as "필기시험문제등록여부",       (case when s.skill\_test\_question = '시험문제등록안함' then 'N' else 'Y' end) as "실기시험문제등록여부",       (case when subject\_score.written\_score is not null then 'Y' else 'N' end) as "필기성적등록여부",       (case when  subject\_score.skill\_score is not null then 'Y' else 'N' end) as "실기성적등록여부",       (case when  subject\_score.attendance\_score is not null then 'Y' else 'N' end) as "출결성적등록여부"  from open\_subject      inner join subject\_score on subject\_score.open\_subject\_seq =  open\_subject.seq      inner join subject on subject.seq = open\_subject.subject\_seq      inner join open\_curs on open\_curs.seq = open\_subject.open\_curs\_seq      inner join curriculum on curriculum.seq = open\_curs.curs\_seq      inner join written\_test on written\_test.open\_subject\_seq = open\_subject.seq      inner join skill\_test on skill\_test.open\_subject\_seq = open\_subject.seq      inner join written\_test\_question r on r.written\_test\_seq = written\_test.seq      inner join skill\_test\_question s on s.skill\_test\_seq = skill\_test.seq      where subject.subject\_name = <개설과목명>      order by 개설과정번호;  **--3. 개설 과목별 성적 출력시 1)개설과목별정보 + 2)개설과목별 수강생의 성적정보 출력**  --where절에 원하는 개설과목명을 입력  --너무 길어져서 2개로 나눔.  **--1)개설 과목별 정보 출력(개설 과정명, 과목시작일, 과목종료일, 과정번호, 과정명, 과정기간, 강의실명, 교사명, 교재명)**  select      distinct subject\_name as 과목명,      open\_subject.begin\_date as 과목시작일,      open\_subject.end\_date as 과목종료일,      open\_curs.seq as 과정번호,     curriculum.name as 과정명,     open\_curs.current\_term || '개월' as 과정기간,     lectureroom.room\_name as 강의실명,     teacher.name as 교사명,     book.subject as 교재명  from open\_curs      inner join open\_subject on open\_subject.open\_curs\_seq = open\_curs.seq      inner join curriculum on open\_curs.curs\_seq = curriculum.seq      inner join subject on subject.seq = open\_subject.subject\_seq      inner join lectureroom on open\_curs.room\_seq = lectureroom.seq      inner join teacher on open\_subject.teacher\_seq = teacher.seq      inner join subject\_book on subject\_book.subject\_seq = subject.seq      inner join book on subject\_book.book\_seq = book.seq      where subject\_name = <개설과목명>      order by 과정번호;  **--2)개설 과목별 수강생의 성적 정보(교육생 이름, 주민번호 뒷자리, 필기점수, 실기점수,출결점수)**  select          s.seq as "교육생 번호",          m.name as "교육생 이름",          m.ssn as "주민번호 뒷자리",          subject.subject\_name as "개설과목명",          ss.written\_score as 필기점수,          ss.skill\_score as 실기점수,          ss.attendance\_score as 출결점수  from student s          inner join member m on m.seq = s.member\_seq          inner join subject\_score ss on ss.student\_seq = s.seq          inner join open\_subject os on os.seq = ss.open\_subject\_seq          inner join subject on subject.seq = os.subject\_seq          where subject.subject\_name = <개설과목명>;  **--4.교육생 개인별 출력시 교육생정보+교육생이 수강한 모든 개설과목에 대한 성적 정보**  --where절에 원하는 교육생 이름을 입력  --(교육생 이름, 주민번호 뒷자리, 개설과정명, 개설과정시작, 개설과정종료, 강의실, 개설 과목명, 교사명, 개설과목시작, 개설과목종료날짜, 필기, 실기, 출결)  select           distinct subject.subject\_name as "개설과목명",          s.seq as "교육생 번호",          m.name as "교육생 이름",          m.ssn as "주민번호 뒷자리",           curriculum.name as 과정명,           open\_curs.begin\_date as "과정시작날짜",           open\_curs.end\_date as "과정종료날짜",           lectureroom.room\_name as 강의실명,          teacher.name as 교사명,          os.begin\_date as "과목시작날짜",          os.end\_date as "과목종료날짜",          ss.written\_score as 필기점수,          ss.skill\_score as 실기점수,          ss.attendance\_score as 출결점수  from student s          inner join member m on m.seq = s.member\_seq          inner join subject\_score ss on ss.student\_seq = s.seq          inner join open\_subject os on os.seq = ss.open\_subject\_seq          inner join subject on subject.seq = os.subject\_seq          inner join open\_curs on open\_curs.seq = os.open\_curs\_seq          inner join curriculum on open\_curs.curs\_seq = curriculum.seq       inner join lectureroom on open\_curs.room\_seq = lectureroom.seq          inner join teacher on os.teacher\_seq = teacher.seq          inner join subject\_book on subject\_book.subject\_seq = subject.seq          inner join book on subject\_book.book\_seq = book.seq          where m.name = <교육생 이름>         order by "교육생 번호"; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-08 | 작성일 | 2022-05-18 |
| 요구사항명 | 출결 관리 및 출결 조회 | 작성자 | 고민지 |
| DML | | | |
| **1. 전체 교육생의 일자별 출결상황 조회**  create or replace view attendance\_info  as  select      a.student\_seq as "학생번호",      a.seq as "출결번호",      a.attendance\_date as "날짜",      case          when min(d.checkin) is not null then '출석'          when to\_char(a.attendance\_date, 'd') = 7 or to\_char(a.attendance\_date, 'd') = 1 then '(주말)'          when (select sick\_or\_etc from sick\_or\_etc where attendance\_seq = a.seq) = '병가' then '병가'          when (select sick\_or\_etc from sick\_or\_etc where attendance\_seq = a.seq) = '기타' then '기타'        when (select count(\*) from holiday where holiday\_date = a.attendance\_date) > 0 then '(공휴일)'          else '결석'      end as "출석",      case          when to\_date(min(d.checkin), 'hh24:mi') >= to\_date('09:10', 'hh24:mi') then '지각'          else '.'      end as "지각",      case          when to\_date(max(d.checkout), 'hh24:mi') < to\_date('17:50', 'hh24:mi') then '조퇴'          else '.'      end as "조퇴",      case          when (select count(\*) from attendance\_detail where attendance\_seq = a.seq ) > 1 then '외출'          else '.'      end as "외출"  from attendance a      left outer join attendance\_detail d          on d.attendance\_seq = a.seq              inner join student s                  on s.seq = a.student\_seq                      where a.attendance\_date < sysdate                          group by a.seq, a.attendance\_date, a.student\_seq; | | | |
| **2. 특정개설과정의 모든 교육생 출결상황 조회**  select      "학생번호",      count(decode("출석", '출석', 1)) as "출석일수",      count(decode("지각조퇴", '출석..', 1)) as "정상일수",      count(decode("지각조퇴", '출석지각.', 1)) as "지각일수",      count(decode("지각조퇴", '출석.조퇴', 1)) as "조퇴일수",      count(decode("지각조퇴", '출석지각조퇴', 1)) as "지각 및 조퇴일수",      count(decode("지각조퇴", '외출', 1)) as "외출일수",      count(decode("지각조퇴", '결석', 1)) as "결석일수",      count(decode("출석", '병가', 1)) as "병가일수",      count(decode("출석", '기타', 1)) as "기타일수"  from      (select i.\*, 출석||지각||조퇴 as "지각조퇴" from attendance\_info i          where "학생번호" in (select seq from student where open\_curs\_seq = <선택한 특정개설과정번호>)              and "날짜" between <조회하는 시작년월일> and <조회하는 끝년월일>)              group by "학생번호";      **3. 특정교육생의 출결현황 조회**  select \* from attendance\_info where "학생번호" = <학생번호>; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-09 | 작성일 | 2022-05-18 |
| 요구사항명 | 취업 공고 관리 및 조회 | 작성자 | 강지윤 |
| DML | | | |
| ▶ where절 < > 위치에는 <>안에 있는 내용에 해당하는 데이터 입력  **--1. 전체 취업 공고조회(회사명, 지역, 분야, 계약형태, 모집인원, 모집시작날짜, 모집종료날짜)**  select          a.company as 회사명,          city.city as 지역,          job\_field.field\_name as 분야,          contract\_type.contract\_type as 계약형태,          a.people as 모집인원,          a.begin\_date as 모집시작날짜,          a.end\_date as 모집종료날짜  from job\_announce a          inner join city on city.seq = a.city\_seq          inner join job\_field on job\_field.seq = a.field\_seq          inner join contract\_type on contract\_type.seq = a.contract\_seq          order by 모집시작날짜;  **--2. 기간별 조회(모집종료날짜가 'yyyy-mm-dd'부터 'yyyy-mm-dd' 사이인 기간 조회)**  --where절에 원하는 기간 입력  select          a.company as 회사명,          city.city as 지역,          job\_field.field\_name as 분야,          contract\_type.contract\_type as 계약형태,          a.people as 모집인원,          a.begin\_date as 모집시작날짜,          a.end\_date as 모집종료날짜  from job\_announce a          inner join city on city.seq = a.city\_seq          inner join job\_field on job\_field.seq = a.field\_seq          inner join contract\_type on contract\_type.seq = a.contract\_seq          where a.end\_date between <기간시작날짜> and <기간종료날짜>          order by 모집종료날짜;    **--3.분야별 (분야명으로 검색 ex. 웹개발) 조회**  --where절에 원하는 분야명 입력  select          a.company as 회사명,          city.city as 지역,          job\_field.field\_name as 분야,          contract\_type.contract\_type as 계약형태,          a.people as 모집인원,          a.begin\_date as 모집시작날짜,          a.end\_date as 모집종료날짜  from job\_announce a          inner join city on city.seq = a.city\_seq          inner join job\_field on job\_field.seq = a.field\_seq          inner join contract\_type on contract\_type.seq = a.contract\_seq          where job\_field.field\_name = <분야명>          order by 모집시작날짜;  **--4. 지역별(지역명으로 검색 - ex. 서울) 조회**  --where절에 원하는 지역명 입력  select          a.company as 회사명,          city.city as 지역,          job\_field.field\_name as 분야,          contract\_type.contract\_type as 계약형태,          a.people as 모집인원,          a.begin\_date as 모집시작날짜,          a.end\_date as 모집종료날짜  from job\_announce a          inner join city on city.seq = a.city\_seq          inner join job\_field on job\_field.seq = a.field\_seq          inner join contract\_type on contract\_type.seq = a.contract\_seq          where city.city = <지역명>          order by 모집시작날짜;  **--5. 취업공고 등록**  --등록할 데이터 입력  insert into job\_announce values ((select max(seq) + 1 from job\_announce), 회사명, 지역번호, 분야번호, 계약형태번호, 모집인원수, 모집시작날짜, 모집종료날짜);  **--6. 취업공고 수정**  --where절에 수정할 취업공고번호를 입력하고, 수정할 데이터 입력  update job\_announce set company = 회사명 where seq = 수정할취업공고번호;  update job\_announce set city\_seq = 지역번호 where seq = 수정할취업공고번호;  update job\_announce set field\_seq = 분야번호 where seq = 수정할취업공고번호;  update job\_announce set constract\_seq = 계약형태번호 where seq = 수정할취업공고번호;  update job\_announce set people = 모집인원 where seq = 수정할취업공고번호;  update job\_announce set begin\_date = 모집시작날짜 where seq = 수정할취업공고번호;  update job\_announce set end\_date = 모집종료날짜 where seq = 수정할취업공고번호;  **--7. 취업공고 삭제**  delete from job\_announce where seq = 삭제할취업공고번호; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | B-10 | 작성일 | 2022-05-18 |
| 요구사항명 | 교육생 상담일지 관리 및 조회 | 작성자 | 강지윤 |
| DML | | | |
| ▶where절 < > 위치에는 <>안에 있는 내용에 해당하는 데이터 입력  **--1. 전체 상담일지 조회 (교육생 이름, 교사이름, 상담주제, 상담날짜)**  select      member.name as 교육생이름, teacher.name as 교사이름, counsel\_topic.counsel\_topic as 상담주제, counsel\_date as 상담날짜  from counsel      inner join counsel\_topic on counsel\_topic.seq = counsel\_topic\_seq      inner join student on student.seq = student\_seq      inner join member on member.seq = student.member\_seq      inner join teacher on teacher.seq = counsel.teacher\_seq;  **--2. 상담일지 기간별 조회**  --where절에 원하는 기간 입력  select      member.name as 교육생이름, teacher.name as 교사이름, counsel\_topic.counsel\_topic as 상담주제, counsel\_date as 상담날짜  from counsel      inner join counsel\_topic on counsel\_topic.seq = counsel\_topic\_seq      inner join student on student.seq = student\_seq      inner join member on member.seq = student.member\_seq      inner join teacher on teacher.seq = counsel.teacher\_seq      where counsel\_date between  <기간시작날짜> and <기간종료날짜>; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-01 | 작성일 | 2022-05-18 |
| 요구사항명 | 교사 계정 | 작성자 | 고민지 |
| DML | | | |
| **1. 로그인한 교사의 개인정보 조회**  select      t.seq,      t.name as "이름",      t.tel as "연락처"  from teacher t      where ssn = <로그인 한 교사의 패스워드>;  **1-2 개인정보 수정하기**  update teacher set (name, tel) = (<이름>,<연락처>) where ssn = <로그인 한 교사의 패스워드>; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-02 | 작성일 | 2022-05-18 |
| 요구사항명 | 강의 스케줄 조회 | 작성자 | 고민지 |
| DML | | | |
| **1. 로그인한 교사의 강의정보를 강의예정, 강의 중, 강의 종료로 구분하여 조회**  select      s.seq as "과목 번호",      case          when s.begin\_date > sysdate then '강의 예정'          when s.end\_date < sysdate then '강의 종료'          else '강의 중'      end as "진행 상태",     (select r.name from curriculum r where r.seq = c.curs\_seq) as "과정명",      c.begin\_date as "과정 시작일",      c.end\_date as "과정 종료일",      (select l.room\_name from lectureroom l where l.seq = c.room\_seq) as "강의실",      (select j.subject\_name from subject j where j.seq = s.subject\_seq) as "과목명",      s.begin\_date as "과목 시작일",      s.end\_date as "과목 종료일",      (select subject from book where seq = b.book\_seq) as 교재명,      subject\_student\_num(s.seq) as "교육생 등록인원"    from open\_subject s      inner join open\_curs c          on s.open\_curs\_seq = c.seq              right outer join subject\_book b                   on b.subject\_seq = s.subject\_seq      where teacher\_seq = <로그인한 교사의 번호>; | | | |
| **2. 로그인한 교사의 특정 강의과목의 수강생 정보 조회**  select      m.name as "이름",      m.tel as "전화번호",      t.regdate as "등록일",      case          when t.drop\_out = '중도탈락' and t.drop\_out\_date < s.end\_date then '중도탈락'          when s.begin\_date >= sysdate then '수강예정'          when (s.begin\_date < sysdate) and (sysdate < s.end\_date) then '수강중'          when s.end\_date <= sysdate then '수료'      end as "수강 상태"  from open\_subject s      inner join open\_curs c          on s.open\_curs\_seq = c.seq              inner join student t                  on t.open\_curs\_seq = c.seq                      inner join member m                          on t.member\_seq = m.seq  where s.seq = <특정 개설과목번호>; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-03 | 작성일 | 2022-05-18 |
| 요구사항명 | 평가 관리 및 조회 | 작성자 | 고민지 |
| DML | | | |
| **1. 로그인한 교사의 강의종료과목 관련 정보를 조회**  select      s.seq as "과목 번호",     (select r.name from curriculum r where r.seq = c.curs\_seq) as "과정명",      c.begin\_date as "과정 시작일",      c.end\_date as "과정 종료일",      (select l.room\_name from lectureroom l where l.seq = c.room\_seq) as "강의실",      (select j.subject\_name from subject j where j.seq = s.subject\_seq) as "과목명",      s.begin\_date as "과목 시작일",      s.end\_date as "과목 종료일",      (select subject from book where seq = b.book\_seq) as 교재명,      p.attend\_point as "출석 배점",      p.written\_point as "필기 배점",      p.skill\_point as "실기 배점"  from open\_subject s      inner join open\_curs c          on s.open\_curs\_seq = c.seq              inner join point p                  on p.open\_subject\_seq = s.seq                      right outer join subject\_book b                          on b.subject\_seq = s.subject\_seq                              where s.end\_date < sysdate                                    and s.teacher\_seq = <로그인한 교사의 번호>  **1-2. 배점정보 입력/ 수정 / 삭제**  --입력  create sequence point\_seq start with (select seq + 1 from (select seq from point order by seq desc) where rownum =1);  insert into point values (point\_seq.nextVal, <선택한 개설과목번호>, <출석배점> , <필기배점>,  <실기배점>);  -- 수정  update point set (attend\_point, written\_point, skill\_point) =  (<출석배점> , <필기배점>,  <실기배점>) where open\_subject\_seq = <선택한 개설과목번호>;  -- 삭제  delete from point where open\_subject\_seq = <선택한 개설과목번호>; | | | |
| **2. 특정 강의 종료 과목의 시험정보 조회**  select      "분류",      seq as "시험번호",      written\_name as "시험명",      written\_date as "시험일자"  from (select '필기' as "분류", w.\* from written\_test w          UNION          select '실기' as "분류" , s.\* from skill\_test s)      where open\_subject\_seq = <선택한 개설과목번호>;          order by "분류" desc, "시험일자";  **2-2 시험정보 입력 / 수정 / 삭제   > 실기인지 필기인지 구분하여 사용! \*\*\***  -- 입력  create sequence written\_test\_seq start with (select seq + 1 from (select seq from written\_test order by seq desc) where rownum =1);  insert into written\_test values(written\_test\_seq.nextVal, <선택한 개설과목번호>, <시험이름>, <시험날짜>)  create sequence skill\_test\_seq start with (select seq + 1 from (select seq from skill\_test order by seq desc) where rownum =1);  insert into skill\_test values(skill\_test\_seq.nextVal, <선택한 개설과목번호>, <시험이름>, <시험날짜>)  -- 수정  update written\_test set (written\_name, written\_date) = (<시험이름>, <시험날짜>) where seq = <시험번호>;  update skill\_test set (skill\_name, skill\_date) = (<시험이름>, <시험날짜>) where seq = <시험번호>;    -- 삭제  delete from written\_test where seq = <시험번호>;  delete from skill\_test where seq = <시험번호>;  **2-3 특정 강의 종료과목의 시험문제 조회**  select      question\_seq as "문제번호",      written\_test\_question as "문제"  from written\_test\_question  where written\_test\_seq = <필기시험번호>;  select      question\_seq as "문제번호",      skill\_test\_question as "문제"  from skill\_test\_question  where skill\_test\_question = <실기시험번호>;  **2-4 시험문제 입력/ 수정/ 삭제**  -- 입력  create sequence written\_question\_seq start with (select seq + 1 from (select seq from written\_test\_question order by seq desc) where rownum =1);  insert into written\_test\_question values (written\_question\_seq.nextVal, <선택한 실기시험번호>, <문제번호>, <문제>);  create sequence skill\_question\_seq start with (select seq + 1 from (select seq from skill\_test\_question order by seq desc) where rownum =1);  insert into skill\_test\_question values (skill\_question\_seq.nextVal, <선택한 실기시험번호>, <문제번호>, <문제>);    -- 수정  update written\_test\_question set (question\_seq, written\_test\_question) = (<문제번호>, <문제>)      where seq = <선택한 필기시험문제번호>;  update skill\_test\_question set (question\_seq, skill\_test\_question) = (<문제번호>, <문제>)      where seq = <선택한 실기시험문제번호>;  -- 삭제  delete from written\_test\_question where seq = <선택한 필기시험문제번호>;  delete from skill\_test\_question where seq = <선택한 실기시험문제번호>; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-04 | 작성일 | 2022-05-18 |
| 요구사항명 | 성적 관리 및 조회 | 작성자 | 권민채 |
| DML | | | |
| 1. 과목 목록 출력  select  open\_subject.seq as 과목번호,  curriculum.name as 과정명,    subject.subject\_name as 개설과목,    open\_subject.begin\_date as "과목 시작일",    open\_subject.end\_date as "과목 종료일",    lectureroom.room\_name as 강의실명,    teacher.name as 교사명  from open\_curs      inner join open\_subject on open\_subject.open\_curs\_seq = open\_curs.seq      inner join curriculum on open\_curs.curs\_seq = curriculum.seq      inner join subject on subject.seq = open\_subject.subject\_seq      inner join lectureroom on open\_curs.room\_seq = lectureroom.seq      inner join teacher on open\_subject.teacher\_seq = teacher.seq      order by open\_subject.seq;  2. 특정 과목번호의 교육생 정보 및 성적 조회  select  open\_subject.seq as 과목번호,  subject.subject\_name as 과목명,  member.name as 이름,  member.tel as 전화번호,  student.drop\_out as "중도탈락 여부",  subject\_score.written\_score as 필기점수,  subject\_score.skill\_score as 실기점수,  subject\_score.attendance\_score as 출결점수  from student  inner join member on member.seq = student.member\_seq  inner join subject\_score on subject\_score.student\_seq = student.seq  inner join open\_subject on open\_subject.seq = subject\_score.open\_subject\_seq  inner join subject on subject.seq = open\_subject.subject\_seq  where open\_subject.seq = <과목번호>; | | | |
| 3. 특정 교육생 번호와 과목명 입력시 시험 점수를 입력   select  open\_subject.seq as 과목번호,  subject.subject\_name as 과목명,  member.name as 이름,  student.seq as 교육생번호,  student.drop\_out as "중도탈락 여부",  subject\_score.written\_score as 필기점수,  subject\_score.skill\_score as 실기점수,  subject\_score.attendance\_score as 출결점수  from student  inner join member on member.seq = student.member\_seq  inner join subject\_score on subject\_score.student\_seq = student.seq  inner join open\_subject on open\_subject.seq = subject\_score.open\_subject\_seq  inner join subject on subject.seq = open\_subject.subject\_seq  where subject\_score.written\_score is null;  update subject\_score  set  subject\_score.written\_score = <필기점수>,  subject\_score.skill\_score = <실기점수>,  subject\_score.attendance\_score = <출결점수>,  where subject\_score.open\_subject\_seq =<과목번호> and subject\_score.student\_seq = <교육생번호>; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-05 | 작성일 | 2022-05-18 |
| 요구사항명 | 출결 관리 및 출결 조회 | 작성자 | 권민채 |
| DML | | | |
| 1.전체 출결현황 조회  select      a.student\_seq as "학생번호",      a.seq as "출결번호",      a.attendance\_date as "날짜",      case          when min(d.checkin) is not null then '출석'          when to\_char(a.attendance\_date, 'd') = 7 or to\_char(a.attendance\_date, 'd') = 1 then '(주말)'          when (select sick\_or\_etc from sick\_or\_etc where attendance\_seq = a.seq) = '병가' then '병가'          when (select sick\_or\_etc from sick\_or\_etc where attendance\_seq = a.seq) = '기타' then '기타'        when (select count(\*) from holiday where holiday\_date = a.attendance\_date) > 0 then '(공휴일)'          else '결석'      end as "출석",      case          when to\_date(min(d.checkin), 'hh24:mi') >= to\_date('09:10', 'hh24:mi') then '지각'          else '.'      end as "지각",      case          when to\_date(max(d.checkout), 'hh24:mi') < to\_date('17:50', 'hh24:mi') then '조퇴'          else '.'      end as "조퇴",      case          when (select count(\*) from attendance\_detail where attendance\_seq = a.seq ) > 1 then '외출'          else '.'      end as "외출"  from attendance a      left outer join attendance\_detail d          on d.attendance\_seq = a.seq              inner join student s                  on s.seq = a.student\_seq                      where a.attendance\_date < sysdate                          group by a.seq, a.attendance\_date, a.student\_seq;  2.특정개설과정의 모든 교육생 출결상황 조회  select      "학생번호",      count(decode("출석", '출석', 1)) as "출석일수",      count(decode("지각조퇴", '출석..', 1)) as "정상일수",      count(decode("지각조퇴", '출석지각.', 1)) as "지각일수",      count(decode("지각조퇴", '출석.조퇴', 1)) as "조퇴일수",      count(decode("지각조퇴", '출석지각조퇴', 1)) as "지각 및 조퇴일수",      count(decode("지각조퇴", '외출', 1)) as "외출일수",      count(decode("지각조퇴", '결석', 1)) as "결석일수",      count(decode("출석", '병가', 1)) as "병가일수",      count(decode("출석", '기타', 1)) as "기타일수"  from      (select i.\*, 출석||지각||조퇴 as "지각조퇴" from attendance\_info i          where "학생번호" in (select seq from student where open\_curs\_seq = <선택한 특정개설과정번호>)              and "날짜" between <조회하는 시작년월일> and <조회하는 끝년월일>)              group by "학생번호"; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-06 | 작성일 | 2022-05-18 |
| 요구사항명 | 상담일지 작성 및 조회 | 작성자 | 권민채 |
| DML | | | |
| 1. 상담일지 조회  select      member.name as 교육생이름,      counsel\_topic.counsel\_topic as 상담주제,      counsel\_date as 상담날짜,      curriculum.name as 수강과정  from counsel      inner join counsel\_topic on counsel\_topic.seq = counsel\_topic\_seq      inner join student on student.seq = student\_seq      inner join member on member.seq = student.member\_seq      inner join teacher on teacher.seq = counsel.teacher\_seq    inner join open\_curs on open\_curs.seq = student.open\_curs\_seq    inner join curriculum on curriculum.seq = open\_curs.curs\_seq   where teacher.name = <로그인한 교사 이름>    order by counsel\_date; | | | |
| 2. 특정 교육생의 상담현황 조회  select      member.name as 교육생이름,      counsel\_topic.counsel\_topic as 상담주제,      counsel\_date as 상담날짜,      curriculum.name as 수강과정  from counsel      inner join counsel\_topic on counsel\_topic.seq = counsel\_topic\_seq      inner join student on student.seq = student\_seq      inner join member on member.seq = student.member\_seq      inner join teacher on teacher.seq = counsel.teacher\_seq    inner join open\_curs on open\_curs.seq = student.open\_curs\_seq    inner join curriculum on curriculum.seq = open\_curs.curs\_seq   where member.name = <교육생 이름>    order by counsel\_date; | | | |
| 3. 특정 과정별 상담현황 조회  select      member.name as 교육생이름,      counsel\_topic.counsel\_topic as 상담주제,      counsel\_date as 상담날짜,      curriculum.name as 수강과정  from counsel      inner join counsel\_topic on counsel\_topic.seq = counsel\_topic\_seq      inner join student on student.seq = student\_seq      inner join member on member.seq = student.member\_seq      inner join teacher on teacher.seq = counsel.teacher\_seq    inner join open\_curs on open\_curs.seq = student.open\_curs\_seq    inner join curriculum on curriculum.seq = open\_curs.curs\_seq   where curriculum.name = <과정이름>   order by counsel\_date; | | | |
| 4.상담일지 등록  insert into counsel values (counsel\_seq.nextval, <교육생번호>, <상담주제번호>, <상담일자>);  5.상담일지 수정  update counsel  set (student\_seq, counsel\_topic\_seq, counsel\_date) = (<교육생번호>,<상담주제번호>,<상담일자>)  where counsel\_seq = <상담일지 번호>;    6.상담일지 삭제  delete from counsel where counsel\_seq = <상담일지 번호>; | | | |
| 3. 지역별 취업공고 조회  select  city.city as "지역",  job\_announce.company as "회사명",  job\_field.field\_name as "분야",  contract\_type.contract\_type as "직위",  job\_announce.people as "모집인원",  job\_announce.begin\_date as "모집 시작일",  job\_announce.end\_date as "모집 종료일"  from job\_announce  inner join city on job\_announce.city\_seq = city.seq  inner join job\_field on job\_announce.field\_seq= job\_field.seq  inner join contract\_type on job\_announce.contract\_seq = contract\_type.seq  where city.city = <지역명>; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-07 | 작성일 | 2022-05-17 |
| 요구사항명 | 과제 등록 및 제출 현황 조회 | 작성자 | 주혜원 |
| DML | | | |
| --과제조회  select  st.seq as 교육생번호,  m.name as 교육생이름,  s.seq as 과제제출번호,  s.task\_seq as 과제번호,  s.summit\_date as 제출날짜,  t.open\_curs\_seq as 과정,  t.open\_subject\_seq 과목,  t.task\_description as  "과제설명"  from task\_list t      inner join summit s          on t.seq = s.task\_seq              inner join student st                  on  st.seq = s.student\_seq                      inner join member m                          on m.seq = st.member\_seq                                 order by st.seq;  --과제 등록 ->프로시저 →완료      insert into task\_list(seq,open\_curs\_seq,open\_subject\_seq,task\_description,summit limit) values ((select max(seq) + 1 from task\_list),12,<과목번호>, <추가과제>,<제출기한>); | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | D-01 | 작성일 | 2022-05-17 |
| 요구사항명 | 교육생 계정 | 작성자 | 주혜원 |
| DML | | | |
| 계정으로 로그인 후교육생 기본정보 간단하게 출력  s.seq as "교육생번호",  m.name as "이름",  m.ssn as  "주민등록번호",  c.name as"과정이름",  s.regdate as "등록일",  l.room\_name as "강의실명"  from student  s      inner join member m          on m.seq = s.member\_seq              inner join open\_curs   o                  on o.seq = s.open\_curs\_seq                      inner join curriculum c                          on c.seq = o.curs\_seq                              inner join lectureroom l                                  on l.seq = o.room\_seq; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 명 | D-02 | 작성일 | 2022-05-17 |
| System | 성적 조회 | 작성자 | 주혜원 |
| Query | | | |
| --제목 출력 ->포로시저O 완료  --교육생 계정으로 로그인 후 본인의 개인정보와 본인이 수강한 과정명, 과정기간(시작 년월일, 끝 년월일), 강의실  select  s.seq as "교육생번호",  m.name as "이름",  m.tel as "전화번호",  c.name as"과정이름",  o.current\_term as "과정기간",  o.begin\_date  as "과정시작일",  o.end\_date as "과정종료일",  l.room\_name as "강의실"  from student  s      inner join member m          on m.seq = s.member\_seq              inner join open\_curs   o                  on o.seq = s.open\_curs\_seq                        inner join curriculum c                           on c.seq = o.curs\_seq                              inner join lectureroom l                                  on l.seq = o.room\_seq                                      where m.ssn = <4256385>;    --학생번호,과목번호, 과목명, 과목기간, 교재명, 교사명, 과목별 배점 정보(출결,필기,실기 배점), 과목별 성적정보(출결,필기,실기 점수), 과목별 시험 날짜  select  -> 프로시저 ㅇ 완료  s.seq "학생 번호",  os.subject\_seq as "과목번호",  m.ssn as”주민번호”,  sb.subject\_name as "과목이름",  os.end\_date - os.begin\_date as "과목기간",  b.subject as "교재이름",  t.name as "교사명",  p.attend\_point as "출석배점",  p.written\_point as "필기배점",  p.skill\_point as "실기배점",  (select written\_score from subject\_score where student\_seq = s.seq and  open\_subject\_seq = os.seq) as "필기점수",  (select skill\_score from subject\_score where student\_seq = s.seq and  open\_subject\_seq = os.seq) as "실기점수",  (select attendance\_score from subject\_score where student\_seq = s.seq and  open\_subject\_seq = os.seq) as "출결점수",  st.skill\_date as "실기시험날짜"  from student s      inner join open\_curs o          on o.seq = s.open\_curs\_seq              inner join open\_subject os                  on o.seq = os.open\_curs\_seq                      inner join subject sb                          on sb.seq = os.subject\_seq                              inner join subject\_book sbb                                  on sb.seq= sbb.book\_seq                                      inner join book b                                          on b.seq= sbb.book\_seq                                              inner join teacher t                                                  on t.seq = os.teacher\_seq                                                          left outer join point p                                                             on os.seq = p.open\_subject\_seq                                                                 inner join skill\_test st                                                                      on os.seq = st.open\_subject\_seq                                                                          inner join member m                                                                               on m.seq = s.member\_seq                                                                                  where m.ssn =<2660833>                                                                                    order by s.seq;  --과목 컬럼은 따로 빼서 만듦(너무 커져서)  --필기시험 ->프로시저 O 완료  select      wt.open\_subject\_seq as "과목번호",      s.subject\_name as "과목이름",      os.end\_date - os.begin\_date as "과목기간",      wt.seq as "필기시험번호",      wt.written\_name as "필기시험 이름",      q.written\_test\_question as "필기시험문제"  from written\_test wt          inner join written\_test\_question q              on wt.seq = q.written\_test\_seq                  inner join open\_subject os                      on os.seq = wt.open\_subject\_seq                          inner join subject s                             on  s.seq = os.subject\_seq                                  where  wt.open\_subject\_seq= <51>; --보고싶은 과목 번호 입력    --실기시험 ->프로시저 ->완료  select      st.open\_subject\_seq as "과목번호",      s.subject\_name as "과목이름",      os.end\_date - os.begin\_date as "과목기간",      st.seq as "실기시험번호",      st.skill\_name as "실기시험 이름",      q.skill\_test\_question as "실기시험문제"  from skill\_test st          inner join skill\_test\_question q              on st.seq = q.skill\_test\_seq                  inner join open\_subject os                      on os.seq = st.open\_subject\_seq                          inner join subject s                             on  s.seq = os.subject\_seq                                   where  st.open\_subject\_seq= <51>; --보고싶은 과목 번호 입력 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 명 | D-03 | 작성일 | 2022-05-17 |
| System | 출결 관리 및 출결 조회 | 작성자 | 주혜원 |
| Query | | | |
| --근태,기록하기      insert into attendance\_detail (seq, attendance\_seq,checkin,checkout) values ((select max(seq) + 1 from attendance.detial),     (select max(attendance\_seq) + 1 from attendance.detial), <입실시간>,<퇴실시간>);  --출결조회하기  select      "학생번호",      count(decode("출석", '출석', 1)) as "출석일수",      count(decode("지각조퇴", '출석..', 1)) as "정상일수",      count(decode("지각조퇴", '출석지각.', 1)) as "지각일수",      count(decode("지각조퇴", '출석.조퇴', 1)) as "조퇴일수",      count(decode("지각조퇴", '출석지각조퇴', 1)) as "지각 및 조퇴일수",      count(decode("지각조퇴", '외출', 1)) as "외출일수",      count(decode("지각조퇴", '결석', 1)) as "결석일수",      count(decode("출석", '병가', 1)) as "병가일수",      count(decode("출석", '기타', 1)) as "기타일수"  from (select i.\*, 출석||지각||조퇴 as "지각조퇴" from attendance\_info i   where <3> in (select seq from student where open\_curs\_seq = <11>)  and "날짜" between <'2022-03-03'> and <'2022-05-05'>)              group by "학생번호"; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 명 | D-04 | 작성일 | 2022-05-17 |
| System | 상담일지조회 | 작성자 | 주혜원 |
| Query | | | |
| select  -> 프로시저  m.name as 이름,  c.counsel\_date as 상담날짜,  t.counsel\_topic as 상담주제,  t.name  as 교사명  from counsel c      inner join counsel\_topic t          on t.seq = c.counsel\_topic\_seq              inner join student s                  on s.seq = c.student\_seq                      inner join member m                          on m.seq = s.member\_seq                              inner join teacher t                                  on t.seq = c.teacher\_seq                                      where m.ssn = <2660833>; | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | D-05 | 작성일 | 2022-05-18 |
| 요구사항명 | 과제 조회 및 제출 | 작성자 | 김성태 |
| DML | | | |
| --과목 별 과제 확인  select  c.name as "과정 명",  s.subject\_name as "과목 명",  t.task\_description as "과제설명",  t.summit\_limit as "제출기한"  from task\_list t  inner join open\_curs oc  on t.open\_curs\_seq = oc.seq  inner join curriculum c  on oc.curs\_seq = c.seq  inner join open\_subject os  on t.open\_subject\_seq = os.seq  inner join subject s  on os.subject\_seq = s.seq;  --자신이 제출한 과제 목록 조회  select  s.seq,  t.task\_description as “과제설명”,  m.name as “제출자”,  s.summit\_date as “제출날짜”,  t.summit\_limit as “제출기한”    from summit s  inner join task\_list t  on s.task\_seq = t.seq  inner join student stu  on s.student\_seq = stu.seq  inner join member m  on stu.member\_seq = m.seq  where m.ssn = <2108289>; -- 로그인 한 교육생의 주민번호 뒷자리(패스워드) | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | D-06 | 작성일 | 2022-05-19 |
| 요구사항명 | 스터디 조회기능 | 작성자 | 임종운 |
| DML | | | |
| **<>안에 CRUD할 대상의 정보를 입력**  **1. 스터디 조회**  SELECT      st.topic, sr.study\_name, sr.study\_description, m.name, sr.email, m.tel, sr.begin\_date, sr.end\_date, sr.completement  FROM study\_recruit sr      INNER JOIN study\_topic st ON sr.study\_topic\_seq = st.seq      INNER JOIN student s ON s.seq = sr.student\_seq      INNER JOIN member m ON s.member\_seq = m.seq  WHERE      sr.completement = **<모두보기(‘o’), 모집중 공고만보기(‘x’)>**  **2. 스터디 모집 공고 등록**  **2.1 등록하는 사람 이름 확인**  SELECT      s.seq, m.name  FROM student s      INNER JOIN member m ON s.member\_seq = m.seq;  **2.2 스터디 토픽조회**  SELECT      \*  FROM study\_topic  WHERE topic LIKE '%%';  **2.3 스터디 등록**  INSERT INTO study\_recruit(seq, student\_seq, study\_topic\_seq, study\_name, study\_description, begin\_date, end\_date, recruit\_limit, email, completement)              VALUES ( study\_recruit\_seq.nextval, <등록하는 사람의 일련번호>, <스터디 주제 일련번호>, <'스터디 이름'>, <'스터디 설명'>, <시작일자>, <종료일자>, <모집정원>, <이메일>, <마감여부>);  **2.4 스터디 모집상태 변경**  UPDATE study\_recruit SET completement = **<모집상태(‘o’, ‘x’)>** WHERE seq = **<스터디 일련번호>**;  **2.4 스터디 모집 삭제**  **CASCADE** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | D-07 | 작성일 | 2022-05-18 |
| 요구사항명 | 취업공고 조회 | 작성자 | 권민채 |
| DML | | | |
| 1. 분야별 취업 공고 조회  select  city.city as "지역",  job\_announce.company as "회사명",  job\_field.field\_name as "분야",  contract\_type.contract\_type as "직위",  job\_announce.people as "모집인원",  job\_announce.begin\_date as "모집 시작일",  job\_announce.end\_date as "모집 종료일"  from job\_announce  inner join city on job\_announce.city\_seq = city.seq  inner join job\_field on job\_announce.field\_seq= job\_field.seq  inner join contract\_type on job\_announce.contract\_seq = contract\_type.seq  where contract\_type.contract\_type = <직위>; | | | |
| 2. 기간별 취업공고 조회  select  city.city as "지역",  job\_announce.company as "회사명",  job\_field.field\_name as "분야",  contract\_type.contract\_type as "직위",  job\_announce.people as "모집인원",  job\_announce.begin\_date as "모집 시작일",  job\_announce.end\_date as "모집 종료일"  from job\_announce  inner join city on job\_announce.city\_seq = city.seq  inner join job\_field on job\_announce.field\_seq= job\_field.seq  inner join contract\_type on job\_announce.contract\_seq = contract\_type.seq  where job\_announce.end\_date between  <모집시작일>  and <모집종료일>; | | | |