**데이터 쿼리문(ANSI-SQL)**

| 요구사항번호 | B04 | 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,  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 | B05 | 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 | B06 | 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 | D06 | 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** | | | |