**DB Object 스크립트(PL-SQL)**

| 요구사항번호 | **B-09** 물품 대여 | 작성일 | 2024-08-20 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생에게 물품 대여해주는 기능 제공 | 작성자 | 임호균 |
| DML | | | |
| 1. 관리자는 대여 물품 목록, 대여 기록을 등록,수정,삭제,조회 할 수 있다. 2. 물품 대여를 등록 , 조회시 항목은 다음과 같다.  * 대여 물품 목록 * 물품 목록 * 총 개수 * 여분 개수  1. 대여  * 대여한 학생 * 대여일 * 반납일 * 비고   **-- 대여 내역 등록**  CREATE OR REPLACE PROCEDURE rental\_insert (  pseq IN tblRental\_Student.seq%TYPE,  psstudent IN tblRental\_Student.student\_seq%TYPE,  prental IN tblRental\_Student.rental\_seq%TYPE,  prent IN tblRental\_Student.rent\_date%TYPE,  preturn IN tblRental\_Student.return\_date%TYPE,  pnote IN tblRental\_Student.note%TYPE  )  IS  BEGIN  INSERT INTO tblRental\_Student(  seq,  student\_seq,  rental\_seq,  rent\_date,  return\_date,  note  ) VALUES (  pseq,  psstudent,  prental,  prent,  preturn,  pnote  );  COMMIT;  DBMS\_OUTPUT.PUT\_LINE('등록완료');    END rental\_insert;  /  **-- 대여 내역 등록 호출**  BEGIN  rental\_insert (RENTAL\_STUDENT\_SEQ.NEXTVAL, 1000, 5, '2023-12-24', '2023-12-25', '마음이 따뜻해지네요.');  END;  /  **-- 대여 내역 수정**  CREATE OR REPLACE PROCEDURE rental\_mod (  pseq IN tblRental\_Student.seq%TYPE,  pstudent IN tblRental\_Student.student\_seq%TYPE,  prental IN tblRental\_Student.rental\_seq%TYPE,  prent IN tblRental\_Student.rent\_date%TYPE,  preturn IN tblRental\_Student.return\_date%TYPE  )  IS  BEGIN  UPDATE tblRental\_Student  SET  student\_seq = pstudent,  rental\_seq = prental,  rent\_date = prent,  return\_date = preturn  WHERE seq = pseq;    DBMS\_OUTPUT.PUT\_LINE('수정완료');  COMMIT;  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('수정실패');  COMMIT;  END rental\_mod;  /  **-- 대여 내역 수정 호출**  BEGIN  rental\_mod (  pseq => 114,  pstudent => 900,  prental => 10,  prent => '2024-05-05',  preturn => '2024-05-07'  );  END;  /  **-- 대여 내역 삭제**  CREATE OR REPLACE PROCEDURE rental\_delete (  pseq IN tblRental\_Student.seq%TYPE  )  IS  BEGIN  DELETE FROM tblRental\_Student  WHERE seq = pseq;    DBMS\_OUTPUT.PUT\_LINE('삭제 완료');  COMMIT;  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('삭제 실패');  ROLLBACK;  END rental\_delete;  /  **-- 대여 내역 삭제 호출**  BEGIN  rental\_delete (  pseq => 114  );  END;  /  -**- 대여 내역 조회**  CREATE OR REPLACE PROCEDURE rental\_view  IS  BEGIN  FOR v IN (  SELECT  B.ITEM AS 대여물품,  SI.NAME AS 대여한학생,  B.RENT\_DATE AS 대여일,  B.RETURN\_DATE AS 반납일,  B.TOTAL\_COUNT AS 총갯수,  B.CURRENT\_COUNT AS 여분갯수,  B.NOTE AS 비고  FROM (SELECT  A.STUDENT\_SEQ,  A.RENT\_DATE,  A.RETURN\_DATE,  A.NOTE,  A.ITEM,  A.TOTAL\_COUNT,  A.CURRENT\_COUNT,  S.STUDENTINFO\_SEQ  FROM ( SELECT  RS.STUDENT\_SEQ,  RS.RENT\_DATE,  RS.RETURN\_DATE,  RS.NOTE,  R.TOTAL\_COUNT,  R.CURRENT\_COUNT,  R.ITEM  FROM tblRental\_Student RS  INNER JOIN tblRental R  ON RS.RENTAL\_SEQ = R.SEQ) A  INNER JOIN tblStudent S  ON A.STUDENT\_SEQ = S.SEQ) B  INNER JOIN tblStudentInfo SI  ON B.STUDENTINFO\_SEQ = SI.SEQ  )  LOOP  DBMS\_OUTPUT.PUT\_LINE('대여물품: '||v.대여물품);  DBMS\_OUTPUT.PUT\_LINE('대여한학생: '||v.대여한학생);  DBMS\_OUTPUT.PUT\_LINE('대여일: '||v.대여일);  DBMS\_OUTPUT.PUT\_LINE('반납일: '||v.반납일);  DBMS\_OUTPUT.PUT\_LINE('총갯수: '||v.총갯수||'개');  DBMS\_OUTPUT.PUT\_LINE('여분갯수: '||v.여분갯수||'개');  DBMS\_OUTPUT.PUT\_LINE('비고: '||v.비고);  DBMS\_OUTPUT.PUT\_LINE('==============================대여 내역===============================');  END LOOP;    END rental\_view;  /  **-- 대여 내역 조회 호출**  BEGIN  rental\_view;  END;  / | | | |

| 요구사항번호 | **B-10** 분실물 센터 | 작성일 | 2024-08-20 |
| --- | --- | --- | --- |
| 요구사항명 | 분실물을 관리하는 기능 제공 | 작성자 | 임호균 |
| DML | | | |
| 1. 관리자는 학원에서 나온 분실물을 등록,수정,삭제,조회가 가능하다. 2. 분실물 등록, 조회 가능한 항목은 다음과 같다.  * 분실물 이름 * 분실물에 대한 설명 * 습득 날짜 * 반환 날짜 * 습득 위치   **-- 분실물 정보 등록**  CREATE OR REPLACE PROCEDURE lost\_item\_insert(  pseq IN tblLostItem.seq%TYPE,  pstudent\_seq IN tblLostItem.student\_seq%TYPE,  pitem IN tblLostItem.item%TYPE,  pcomment IN tblLostItem.item\_comment%TYPE,  plocation IN tblLostItem.get\_location%TYPE,  pget IN tblLostItem.get\_date%TYPE,  preturn IN tblLostItem.return\_date%TYPE  )  IS  BEGIN  INSERT INTO tblLostItem (  seq,  student\_seq,  item,  item\_comment,  get\_location,  get\_date,  return\_date  ) VALUES (  pseq,  pstudent\_seq,  pitem,  pcomment,  plocation,  pget,  preturn  );    COMMIT;  DBMS\_OUTPUT.PUT\_LINE('등록 완료');    END lost\_item\_insert;  /  **-- 분실물 정보 등록 호출**  BEGIN  lost\_item\_insert (LOSTITEM\_SEQ.NEXTVAL,  500,  '지갑',  '테스트용 지갑',  '강의실5',  TO\_DATE('2024-01-01', 'YYYY-MM-DD'),  TO\_DATE('2024-01-02', 'YYYY-MM-DD')  );  END;  /  **-- 분실물 정보 수정**  CREATE OR REPLACE PROCEDURE lost\_item\_mod(  pseq IN tblLostItem.seq%TYPE,  pstudent\_seq IN tblLostItem.student\_seq%TYPE,  pitem IN tblLostItem.item%TYPE,  pcomment IN tblLostItem.item\_comment%TYPE,  plocation IN tblLostItem.get\_location%TYPE,  pget IN tblLostItem.get\_date%TYPE,  preturn IN tblLostItem.return\_date%TYPE  )  IS  BEGIN  UPDATE tblLostItem  SET student\_seq = pstudent\_seq,  item = pitem,  item\_comment = pcomment,  get\_location = plocation ,  get\_date = pget,  return\_date = preturn  WHERE seq = pseq;    DBMS\_OUTPUT.PUT\_LINE('수정 완료');  COMMIT;  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('수정 실패');  ROLLBACK;    END lost\_item\_mod;  /  **-- 분실물 정보 수정 호출**  BEGIN  lost\_item\_mod (  pseq => 16,  pstudent\_seq => 500,  pitem => '지갑 -> 핸드폰',  pcomment => '아이폰 울트라 16',  plocation => '엘리베이터 (그만잃어버리기)',  pget => TO\_DATE('2023-12-24', 'YYYY-MM-DD'),  preturn => TO\_DATE('2024-01-02', 'YYYY-MM-DD')  );  END;  /  **-- 분실물 정보 삭제**  CREATE OR REPLACE PROCEDURE lost\_item\_delete (  pseq IN tblLostItem.seq%TYPE  )  IS  BEGIN  DELETE FROM tblLostItem  WHERE seq = pseq;    DBMS\_OUTPUT.PUT\_LINE('삭제 완료');  COMMIT;  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('삭제 실패');  ROLLBACK;  END lost\_item\_delete;  /  **-- 분실물 정보 삭제 호출**  BEGIN  lost\_item\_delete (  pseq => 16  );  END;  /  **-- 분실물 정보 조회**  CREATE OR REPLACE PROCEDURE lost\_item\_view  IS  BEGIN  FOR c IN (  SELECT  A.NAME AS "이름",  LI.ITEM AS "분실물",  LI.ITEM\_COMMENT AS "설명",  LI.GET\_LOCATION AS "습득처",  LI.GET\_DATE AS "습득일자",  LI.RETURN\_DATE AS "반환일자"  FROM tblLostitem LI  INNER JOIN  (SELECT \* FROM tblStudent S  INNER JOIN tblStudentInfo SI  ON S.STUDENTINFO\_SEQ = SI.SEQ) A  ON LI.STUDENT\_SEQ = A.STUDENTINFO\_SEQ  )  LOOP  DBMS\_OUTPUT.PUT\_LINE('이름: '|| c.이름);  DBMS\_OUTPUT.PUT\_LINE('분실물: '|| c.분실물);  DBMS\_OUTPUT.PUT\_LINE('설명: '|| c.설명);  DBMS\_OUTPUT.PUT\_LINE('습득처: '|| c.습득처);  DBMS\_OUTPUT.PUT\_LINE('습득일자: '|| c.습득일자);  DBMS\_OUTPUT.PUT\_LINE('반환일자: '|| c.반환일자);  DBMS\_OUTPUT.PUT\_LINE('------------------------------------------------');  END LOOP;  END lost\_item\_view;  /  **-- 분실물 정보 조회 호출**  BEGIN  lost\_item\_view;  END;  / | | | |

| 요구사항번호 | **B-11** 자격증 취득 관리 | 작성일 | 2024-08-20 |
| --- | --- | --- | --- |
| 요구사항명 | 자격증을 관리하는 기능 제공 | 작성자 | 임호균 |
| DML | | | |
| 1. 관리자는 과정 전, 중으로 자격증 취득 한 학생의 정보를  등록, 수정, 삭제, 조회 할 수 있다. 2. 출력 되는 자격증 정보는 아래와 같다.  * 학생 번호 * 자격증 명 * 자격증 취득일   **--자격증 등록**  CREATE OR REPLACE PROCEDURE license\_insert (  pseq IN NUMBER,  pstudentinfo\_seq IN NUMBER,  plicenselist\_seq IN NUMBER,  pget\_date IN DATE  )  IS  BEGIN  INSERT INTO tblLicense (  seq,  studentinfo\_seq,  licenselist\_seq,  get\_date  )  VALUES (  pseq,  pstudentinfo\_seq,  plicenselist\_seq,  pget\_date  );    COMMIT;  DBMS\_OUTPUT.PUT\_LINE('자격증 등록이 완료되었습니다.');  END;  /  -**-자격증 등록 호출**  BEGIN  license\_insert (1008, 50, 2, DATE '2024-01-15');  END;  /  **--자격증 등록 수정**  CREATE OR REPLACE PROCEDURE license\_mod (  plicenselist IN tblLicense.licenselist\_seq%TYPE,  pget\_date IN tblLicense.get\_date%TYPE,  pseq IN tblLicense.seq%TYPE  )  IS  BEGIN  UPDATE tblLicense  SET licenselist\_seq = plicenselist,  get\_date = pget\_date  WHERE seq = pseq;    DBMS\_OUTPUT.PUT\_LINE('수정 완료');  COMMIT;    EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('수정 실패. 다시 입력하세요');  ROLLBACK;  END license\_mod;  /  **--자격증 등록 수정 호출**  BEGIN  license\_mod (  pseq => 1009,  plicenselist => 1,  pget\_date => DATE '2024-08-19'  );  END;  /  **--자격증 등록 삭제**  CREATE OR REPLACE PROCEDURE license\_delete (  pseq IN tblLicense.seq%TYPE  )  IS  BEGIN  DELETE FROM tblLicense  WHERE seq = pseq;    DBMS\_OUTPUT.PUT\_LINE('삭제완료');  COMMIT;    EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('삭제실패');  ROLLBACK;  END license\_delete;  /  **--자격증 등록 삭제 호출**  BEGIN  license\_delete (  pseq => 1009  );  END;  /  **--자격증 조회**  CREATE OR REPLACE PROCEDURE license\_view\_info AS  BEGIN  FOR c IN (  SELECT  s.name AS "이름",  s.seq AS "학생번호",  ll.name\_license AS "자격증명",  l.get\_date AS "취득일자"  FROM tblLicense l  JOIN tblLicenseList ll  ON l.licenselist\_seq = ll.seq  JOIN tblStudentinfo s  ON l.studentinfo\_seq = s.seq  ) LOOP    dbms\_output.put\_line('이름: '|| c.이름);  dbms\_output.put\_line('학생번호: '|| c.학생번호);  dbms\_output.put\_line('자격증명: '|| c.자격증명);  dbms\_output.put\_line('취득일자: '|| c.취득일자);  dbms\_output.put\_line('-----------------------------');  END LOOP;  END;  /  **--자격증 조회 호출**  EXECUTE license\_view\_info; | | | |

| 요구사항번호 | **B-13** 교사 평가 | 작성일 | 2024-08-20 |
| --- | --- | --- | --- |
| 요구사항명 | 교사 평가 관리 | 작성자 | 임호균 |
| DML | | | |
| 1. 수강 중인 교육생 또는 수료 후 교육생이 교사 평가를 진행하고 관리자는 평가 결과를 등록, 수정, 삭제, 조회 할 수 있다. 2. 등록, 조회 할 수 있는 항목으로는 다음이 있다.  * 교사 번호 * 강의 만족도 ( 문항 + 응답 ) * 강의 건의 사항   **--교사평가 등록**  CREATE OR REPLACE PROCEDURE teahcer\_rating\_insert (  pseq IN NUMBER,  pteacher\_seq IN NUMBER,  pstudent\_seq IN NUMBER,  pq1 IN NUMBER,  pq2 IN NUMBER,  pq3 IN NUMBER,  pq4 IN NUMBER,  pq5 IN NUMBER,  psuggestion IN VARCHAR2  )  IS  BEGIN  INSERT INTO tblTeacherRating (  seq,  teacher\_seq,  student\_seq,  q1,  q2,  q3,  q4,  q5,  suggestion  )  VALUES (  pseq,  pteacher\_seq,  pstudent\_seq,  pq1,  pq2,  pq3,  pq4,  pq5,  psuggestion  );    COMMIT;  DBMS\_OUTPUT.PUT\_LINE('평가 등록 완료.');  END teahcer\_rating\_insert;  /  **--교사평가 등록 호출**  BEGIN  teahcer\_rating\_insert (teacherrating\_seq.NEXTVAL, 1, 1, 5, 5, 5, 5, 5, '이해가 잘가요');  END;  /  **--교사평가 수정**  CREATE OR REPLACE PROCEDURE teacher\_rating\_mod (  pseq IN tblTeacherRating.seq%TYPE,  pteacher\_seq IN tblTeacherRating.teacher\_seq%TYPE,  psuggestion IN tblTeacherRating.suggestion%TYPE  )  IS  BEGIN  UPDATE tblTeacherRating  SET teacher\_seq = pteacher\_seq,  suggestion = psuggestion  WHERE seq = pseq;  DBMS\_OUTPUT.PUT\_LINE('수정완료.');  COMMIT;    EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('수정실패.');  ROLLBACK;  END teacher\_rating\_mod;  /  **--교사평가 수정 호출**  BEGIN  teacher\_rating\_mod (  pseq => 3001,  pteacher\_seq => 4,  psuggestion => '친절하게 잘 설명해주십니다.'  );  END;  /  -**-교사평가 삭제**  CREATE OR REPLACE PROCEDURE teacher\_rating\_delete (  pseq IN tblTeacherRating.seq%TYPE  )  IS  BEGIN  DELETE FROM tblTeacherRating  WHERE seq = pseq;    DBMS\_OUTPUT.PUT\_LINE('삭제완료.');  COMMIT;    EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('삭제실패.');  ROLLBACK;  END teacher\_rating\_delete;  /  **--교사평가 삭제 호출**  BEGIN  teacher\_rating\_delete (  pseq => 3001  );  END;  /  **--교사평가 조회**  CREATE OR REPLACE PROCEDURE teahcer\_rating\_view AS  BEGIN  FOR c IN (  SELECT  t.seq AS "교사번호",  t.name AS "교사명",  tr.q1 AS "Q1",  tr.q2 AS "Q2",  tr.q3 AS "Q3",  tr.q4 AS "Q4",  tr.q5 AS "Q5",  tr.suggestion AS "의견"  FROM tblTeacherRating tr  INNER JOIN tblTeacher t  ON tr.teacher\_seq = t.seq  )  LOOP  DBMS\_OUTPUT.PUT\_LINE('교사번호 : '||c.교사번호);  DBMS\_OUTPUT.PUT\_LINE('교사명 : '||c.교사명);  DBMS\_OUTPUT.PUT\_LINE('Q1: '||c.Q1);  DBMS\_OUTPUT.PUT\_LINE('Q2: '||c.Q2);  DBMS\_OUTPUT.PUT\_LINE('Q3: '||c.Q3);  DBMS\_OUTPUT.PUT\_LINE('Q4: '||c.Q4);  DBMS\_OUTPUT.PUT\_LINE('Q5: '||c.Q5);  DBMS\_OUTPUT.PUT\_LINE('의견: '||c.의견);  DBMS\_OUTPUT.PUT\_LINE('----------------------------');  END LOOP;  END teahcer\_rating\_view;  /  **--교사평가 조회 호출**  execute teahcer\_rating\_view; | | | |

| 요구사항번호 | **B-14** 과정 평가 | 작성일 | 2024-08-20 |
| --- | --- | --- | --- |
| 요구사항명 | 교육 과정 수료 후 과정평가 관리 기능을 제공한다. | 작성자 | 임호균 |
| DML | | | |
| 1. 수강 중인 교육생 또는 수료 후 교육생이 과정 평가를 진행하고  관리자는 평가 결과를 등록, 수정, 삭제, 조회 할 수 있다. 2. 등록, 조회 할 수 있는 항목은 다음과 같다.  * 과정명 * 과정 만족도 * 과정 건의사항   **--과정 평가 등록**  CREATE OR REPLACE PROCEDURE class\_rating\_insert (  pseq IN tblClassRating.seq%TYPE,  popenclass\_seq IN tblClassRating.openclass\_seq%TYPE,  pstudent\_seq IN tblClassRating.student\_seq%TYPE,  pq1 IN tblClassRating.q1%TYPE,  pq2 IN tblClassRating.q2%TYPE,  pq3 IN tblClassRating.q3%TYPE,  pq4 IN tblClassRating.q4%TYPE,  pq5 IN tblClassRating.q5%TYPE,  psuggestion IN tblClassRating.suggestion%TYPE  )  AS  BEGIN  INSERT INTO tblClassRating (  seq,  openclass\_seq,  student\_seq,  q1,  q2,  q3,  q4,  q5,  suggestion  ) VALUES (  pseq,  popenclass\_seq,  pstudent\_seq,  pq1,  pq2,  pq3,  pq4,  pq5,  psuggestion  );  COMMIT;  DBMS\_OUTPUT.PUT\_LINE('평가 등록 완료.');  END class\_rating\_insert;  /  **--과정 평가 등록 호출**  BEGIN  class\_rating\_insert (seqClassRating.NEXTVAL, 1, 100, 5, 5, 5, 5, 5, '만족스럽군요..');  END;  /  -**-과정평가 수정**  CREATE OR REPLACE PROCEDURE class\_rating\_mod (  pseq IN tblClassRating.seq%TYPE,  popenclass\_seq IN tblClassRating.openclass\_seq%TYPE,  psuggestion IN tblClassRating.suggestion%TYPE  )  IS  BEGIN  UPDATE tblClassRating  SET openclass\_seq = popenclass\_seq,  suggestion = psuggestion  WHERE seq = pseq;  DBMS\_OUTPUT.PUT\_LINE('수정완료.');  COMMIT;    EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('수정실패.');  ROLLBACK;  END class\_rating\_mod;  /  **--과정평가 수정 호출**  BEGIN  class\_rating\_mod (  pseq => 1009,  popenclass\_seq => 20,  psuggestion => '기간이 짧아서 아쉽군요..'  );  END;  /  -**-과정평가 삭제**  CREATE OR REPLACE PROCEDURE class\_rating\_delete (  pseq IN tblClassRating.seq%TYPE  )  IS  BEGIN  DELETE FROM tblClassRating  WHERE seq = pseq;    DBMS\_OUTPUT.PUT\_LINE('삭제완료.');  COMMIT;    EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('삭제실패.');  ROLLBACK;  END class\_rating\_delete;  /  **--과정평가 삭제 호출**  BEGIN  class\_rating\_delete (  pseq => 1009  );  END;  /  **--과정 평가 조회**  CREATE OR REPLACE PROCEDURE class\_rating\_view  AS  BEGIN  FOR c IN (  SELECT  c.name AS "과정명",  cr.q1 AS "Q1",  cr.q2 AS "Q2",  cr.q3 AS "Q3",  cr.q4 AS "Q4",  cr.q5 AS "Q5",  cr.suggestion AS "의견"  FROM tblOpenClass oc  JOIN tblClass c  ON oc.class\_seq = c.seq  LEFT JOIN tblClassRating cr  ON oc.seq = cr.openclass\_seq  ) LOOP  DBMS\_OUTPUT.PUT\_LINE('과정명: '||c.과정명);  DBMS\_OUTPUT.PUT\_LINE('Q1: '||c.Q1);  DBMS\_OUTPUT.PUT\_LINE('Q2: '||c.Q2);  DBMS\_OUTPUT.PUT\_LINE('Q3: '||c.Q3);  DBMS\_OUTPUT.PUT\_LINE('Q4: '||c.Q4);  DBMS\_OUTPUT.PUT\_LINE('Q5: '||c.Q5);  DBMS\_OUTPUT.PUT\_LINE('의견: '||c.의견);  DBMS\_OUTPUT.PUT\_LINE('----------------------------------------------');  END LOOP;  END class\_rating\_view;  /  **--과정 평가 조회 호출**  BEGIN  class\_rating\_view;  END;  / | | | |

| 요구사항번호 | **B-15** 공지 게시판 게시글 작성 | 작성일 | 2024-08-20 |
| --- | --- | --- | --- |
| 요구사항명 | 게시판을 관리하는 기능을 제공한다 | 작성자 | 임호균 |
| DML | | | |
| 1. 관리자는 게시판 기능을 이용해 게시글을 등록, 수정, 삭제, 조회 할 수 있고 게시판과 각 게시판별 분류는 아래와 같다.  * 공지사항 게시판 * 취업 자료 게시판 * 취업 공고 * 취업 자료 * 취업 후기 * 학습 자료 게시판 * 수업 내용 자료 * 수업 테스트 자료 * 기타   **--게시판 등록(공지)**  CREATE OR REPLACE PROCEDURE newsbox\_insert (  pseq IN tblNewsbox.seq%TYPE,  ptitle IN tblNewsbox.title%TYPE,  pcontent IN tblNewsbox.content%TYPE,  pcreation IN tblNewsbox.creation\_date%TYPE  )  IS  BEGIN  INSERT INTO tblNewsBox (  seq,  title,  content,  creation\_date  ) VALUES (  pseq,  ptitle,  pcontent,  pcreation  );    COMMIT;  DBMS\_OUTPUT.PUT\_LINE('등록완료.');  END newsbox\_insert;  /  **--게시글(종합) 등록 호출**  BEGIN  newsbox\_insert(NEWSBOX\_SEQ.NEXTVAL, '2024년 하절기 휴무 안내', '안녕하세요. 여름맞이 어쩌고 저쩌고 ...입니다..', sysdate);  END;  /  **--게시글(종합) 수정**  CREATE OR REPLACE PROCEDURE newsbox\_mod (  pseq IN tblNewsbox.seq%TYPE,  ptitle IN tblNewsbox.title%TYPE,  pcontent IN tblNewsbox.content%TYPE,  pcreation IN tblNewsbox.creation\_date%TYPE  )  IS  BEGIN  UPDATE tblNewsBox  SET title = ptitle,  content = pcontent,  creation\_date = pcreation  WHERE seq = pseq;    DBMS\_OUTPUT.PUT\_LINE('수정완료.');  COMMIT;  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('수정실패.');  ROLLBACK;    END newsbox\_mod;  /  **--게시글(종합) 수정 호출**  BEGIN  newsbox\_mod (  pseq => 13,  ptitle => '테스트',  pcontent => '여름은 덥다.',  pcreation => sysdate  );  END;  /  **--게시글(종합) 삭제**  CREATE OR REPLACE PROCEDURE newsbox\_delete (  pseq IN tblNewsbox.seq%TYPE  )  IS  BEGIN  DELETE FROM tblNewsbox  WHERE seq = pseq;    DBMS\_OUTPUT.PUT\_LINE('삭제완료.');  COMMIT;    EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('삭제실패.');  ROLLBACK;  END newsbox\_delete;  /  **– 게시글(종합) 삭제 호출**  BEGIN  newsbox\_delete (  pseq => 12  );  END;  /  **--게시판 조회(공지)**  CREATE OR REPLACE PROCEDURE newsbox\_view  IS  BEGIN  FOR c IN (  SELECT \* FROM tblNewsBox  )  LOOP  DBMS\_OUTPUT.PUT\_LINE ('번호: '||c.seq);  DBMS\_OUTPUT.PUT\_LINE ('제목: '||c.title);  DBMS\_OUTPUT.PUT\_LINE ('내용: '||c.content);  DBMS\_OUTPUT.PUT\_LINE ('등록일자: '||c.creation\_date);  DBMS\_OUTPUT.PUT\_LINE('-----------------------------------------------');  END LOOP;  END newsbox\_view;  /  **--조회(공지) 호출**  BEGIN  newsbox\_view;  END;  /  **--게시글(취업) 조회**  CREATE OR REPLACE PROCEDURE jobbox\_view  IS  BEGIN  FOR c IN (  SELECT \* FROM tbljobbox  )  LOOP  DBMS\_OUTPUT.PUT\_LINE ('번호: '||c.seq);  DBMS\_OUTPUT.PUT\_LINE ('제목: '||c.title);  DBMS\_OUTPUT.PUT\_LINE ('내용: '||c.content);  DBMS\_OUTPUT.PUT\_LINE ('등록일자: '||c.creation\_date);  DBMS\_OUTPUT.PUT\_LINE ('좋아요수: '||c.like\_count);  DBMS\_OUTPUT.PUT\_LINE ('싫어요수: '||c.dislike\_count);  DBMS\_OUTPUT.PUT\_LINE('-----------------------------------------------');  END LOOP;  END jobbox\_view;  /  **--게시글(공지) 조회 호출**  BEGIN  jobbox\_view;  END;  /  **--게시글(취업) 등록**  CREATE OR REPLACE PROCEDURE jobbox\_insert (  pseq IN tblJobBox.seq%TYPE,  ptitle IN tblJobBox.title%TYPE,  pcontent IN tblJobBox.content%TYPE,  pcreation IN tblJobBox.creation\_date%TYPE,  plike IN tblJobBox.like\_count%TYPE,  pdislike IN tblJobBox.dislike\_count%TYPE  )  IS  BEGIN  INSERT INTO tblJobBox (  seq,  title,  content,  creation\_date,  like\_count,  dislike\_count  ) VALUES (  pseq,  ptitle,  pcontent,  pcreation,  plike,  pdislike  );  COMMIT;  DBMS\_OUTPUT.PUT\_LINE('등록완료');    END jobbox\_insert;  /  **--게시글(취업) 등록 호출**  BEGIN  jobbox\_insert (JOBBOX\_SEQ.NEXTVAL, '테스트', '테스트입니다~~~', SYSDATE, 500, 1);  END;  /  **--게시글(취업) 수정**  CREATE OR REPLACE PROCEDURE jobbox\_mod (  pseq IN tblJobBox.seq%TYPE,  ptitle IN tblJobBox.title%TYPE,  pcontent IN tblJobBox.content%TYPE,  pcreation IN tblJobBox.creation\_date%TYPE  )  IS  BEGIN  UPDATE tblJobBox  SET tblJobBox.title = ptitle,  tblJobBox.content = pcontent,  tblJobBox.creation\_date = pcreation  WHERE seq = pseq;    DBMS\_OUTPUT.PUT\_LINE('수정완료.');  COMMIT;    EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('수정실패.');  ROLLBACK;    END jobbox\_mod;  /  BEGIN  jobbox\_mod (  pseq => 6,  ptitle => '수정된 제목 테스트',  pcontent => '수정된 내용',  pcreation => sysdate  );  END;  /  **--게시글(취업) 삭제**  CREATE OR REPLACE PROCEDURE jobbox\_delete (  pseq IN tblJobBox.seq%TYPE  )  IS  BEGIN  DELETE FROM tblJobBox  WHERE seq = pseq;    DBMS\_OUTPUT.PUT\_LINE('삭제완료.');  COMMIT;  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('삭제실패.');  ROLLBACK;  END jobbox\_delete;  /  **--게시글(취업) 삭제 호출**  BEGIN  jobbox\_delete (  pseq => 6  );  END;  /  =======================================================================  **--게시글 (자료) 조회**  CREATE OR REPLACE PROCEDURE tipbox\_view  IS  BEGIN  FOR c IN (  SELECT \* FROM tblTipBox  )  LOOP  DBMS\_OUTPUT.PUT\_LINE ('번호: '||c.seq);  DBMS\_OUTPUT.PUT\_LINE ('제목: '||c.title);  DBMS\_OUTPUT.PUT\_LINE ('내용: '||c.content);  DBMS\_OUTPUT.PUT\_LINE ('등록일자: '||c.creation\_date);  DBMS\_OUTPUT.PUT\_LINE ('좋아요수: '||c.like\_count);  DBMS\_OUTPUT.PUT\_LINE ('싫어요수: '||c.dislike\_count);  DBMS\_OUTPUT.PUT\_LINE('-----------------------------------------------');  END LOOP;  END tipbox\_view;  /  **--게시글(자료) 조회 호출**  BEGIN  tipbox\_view;  END;  /  **--게시글(자료) 등록**  CREATE OR REPLACE PROCEDURE tipbox\_insert (  pseq IN tblTipBox.seq%TYPE,  ptitle IN tblTipBox.title%TYPE,  pcontent IN tblTipBox.content%TYPE,  pcreation IN tblTipBox.creation\_date%TYPE,  plike IN tblTipBox.like\_count%TYPE,  pdislike IN tblTipBox.dislike\_count%TYPE  )  IS  BEGIN  INSERT INTO tblTipBox (  seq,  title,  content,  creation\_date,  like\_count,  dislike\_count  ) VALUES (  pseq,  ptitle,  pcontent,  pcreation,  plike,  pdislike  );  COMMIT;  DBMS\_OUTPUT.PUT\_LINE('등록완료');    END tipbox\_insert;  /  **--게시글(자료) 등록 호출**  BEGIN  tipbox\_insert (25, '테스트자료', '테스트 자료 입니다~~~', SYSDATE, 1000, 5);  END;  /  **--게시글(취업) 수정**  CREATE OR REPLACE PROCEDURE tipbox\_mod (  pseq IN tblTipBox.seq%TYPE,  ptitle IN tblTipBox.title%TYPE,  pcontent IN tblTipBox.content%TYPE,  pcreation IN tblTipBox.creation\_date%TYPE  )  IS  BEGIN  UPDATE tblTipBox  SET tblTipBox.title = ptitle,  tblTipBox.content = pcontent,  tblTipBox.creation\_date = pcreation  WHERE seq = pseq;    DBMS\_OUTPUT.PUT\_LINE('수정완료.');  COMMIT;    EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('수정실패.');  ROLLBACK;    END tipbox\_mod;  /  **--게시글(자료) 수정 호출**  BEGIN  tipbox\_mod (  pseq => 42,  ptitle => '수정된 제목 테스트',  pcontent => '수정된 내용',  pcreation => sysdate  );  END;  /  **--게시글(취업) 삭제**  CREATE OR REPLACE PROCEDURE tipbox\_delete (  pseq IN tblTipBox.seq%TYPE  )  IS  BEGIN  DELETE FROM tblTipBox  WHERE seq = pseq;    DBMS\_OUTPUT.PUT\_LINE('삭제완료.');  COMMIT;  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('삭제실패.');  ROLLBACK;  END tipbox\_delete;  /  **--게시글(취업) 삭제 호출**  BEGIN  tipbox\_delete (  pseq => 24  );  END;  / | | | |

| 요구사항번호 | **B-16** 스터디 그룹 | 작성일 | 2024-08-20 |
| --- | --- | --- | --- |
| 요구사항명 | 스터디 그룹 관리, 멘토 교사 배정 기능을 제공한다. | 작성자 | 임호균 |
| DML | | | |
| 1. 교육생의 스터디 그룹을 등록, 수정, 삭제, 조회 할 수 있다. 2. 스터디 그룹의 속성은 다음과 같다 3. 구성 교육생 4. 그룹 구성 일자 5. 그룹 구성 결심   **--스터디 그룹 등록**  CREATE OR REPLACE PROCEDURE studygroup\_insert (  pseq IN tblStudyGroup.seq%TYPE,  pteacher\_seq IN tblStudyGroup.teacher\_seq%TYPE,  pname IN tblStudyGroup.name%TYPE,  pcreation IN tblStudyGroup.creation\_date%TYPE,  pgoal IN tblStudyGroup.goal%TYPE  )  IS  BEGIN  INSERT INTO tblStudyGroup (  seq,  teacher\_seq,  name,  creation\_date,  goal  ) VALUES (  pseq,  pteacher\_seq,  pname,  pcreation,  pgoal  );    COMMIT;  DBMS\_OUTPUT.PUT\_LINE('등록 완료.');    END studygroup\_insert;  /  **--스터디 그룹 등록 호출**  BEGIN  studygroup\_insert (studygroup\_seq.NEXTVAL, NULL, '그룹102', '2024-08-12','재미있게 가보자');  END;  /  **--스터디 그룹 수정**  CREATE OR REPLACE PROCEDURE studygroup\_mod (  pseq IN tblStudyGroup.seq%TYPE,  pteacher\_seq IN tblStudyGroup.teacher\_seq%TYPE,  pname IN tblStudyGroup.name%TYPE,  pgoal IN tblStudyGroup.goal%TYPE  )  IS  BEGIN  UPDATE tblStudyGroup  SET teacher\_seq = pteacher\_seq,  name = pname,  goal = pgoal  WHERE seq = pseq;    DBMS\_OUTPUT.PUT\_LINE('수정 완료!');  COMMIT;  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('수정 실패');  ROLLBACK;    END studygroup\_mod;  /  **--스터디 그룹 수정 호출**  BEGIN  studygroup\_mod (  pseq => 121,  pteacher\_seq => 5,  pname => '열공단',  pgoal => '불사르자'  );  END;  /  **--스터디 그룹 삭제**  CREATE OR REPLACE PROCEDURE studygroup\_delete (  pseq IN tblStudyGroup.seq%TYPE  )  IS  BEGIN  DELETE FROM tblStudyGroup  WHERE seq = pseq;    DBMS\_OUTPUT.PUT\_LINE('삭제완료');  COMMIT;    EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('삭제실패');  ROLLBACK;  END studygroup\_delete;  /  **--스터디 그룹 삭제 호출**  BEGIN  studygroup\_delete (  pseq => 121  );  END;  /  **--스터디 그룹 등록**  CREATE OR REPLACE PROCEDURE studygroup\_insert (  pseq IN tblStudyGroup.seq%TYPE,  pteacher\_seq IN tblStudyGroup.teacher\_seq%TYPE,  pname IN tblStudyGroup.name%TYPE,  pcreation IN tblStudyGroup.creation\_date%TYPE,  pgoal IN tblStudyGroup.goal%TYPE  )  IS  BEGIN  INSERT INTO tblStudyGroup (  seq,  teacher\_seq,  name,  creation\_date,  goal  ) VALUES (  pseq,  pteacher\_seq,  pname,  pcreation,  pgoal  );    COMMIT;  DBMS\_OUTPUT.PUT\_LINE('등록 완료.');    END studygroup\_insert;  /  **--스터디 그룹 등록 호출**  BEGIN  studygroup\_insert (studygroup\_seq.NEXTVAL, NULL, '그룹102', '2024-08-12','재미있게 가보자');  END;  / | | | |

| 요구사항번호 | **B-17** 스터디 룸 관리 기능 | 작성일 | 2024-08-20 |
| --- | --- | --- | --- |
| 요구사항명 | 스터디룸 예약에 관한 관리 기능을 제공한다. | 작성자 | 임호균 |
| DML | | | |
| 1. 스터디 룸 사용 내역을 등록, 수정, 삭제, 조회 할 수 있다. 2. 스터디 룸 사용 내역에 교육생의 이름, 과정명을 조회한다. 3. 스터디 룸의 정보는 강의실과 동일하며 이는 아래와 같다.    1. 스터디 룸 이름    2. 스터디 룸 최대 수용 인원 4. 관리자는 스터디 룸 예약 정보를 관리한다. 출력되는 예약 정보는 아래와 같다.    1. 강의실 번호 (스터디룸 번호)    2. 대여 신청일    3. 대여 신청 시각    4. 스터디룸 이용 날짜    5. 이용 시작 시각    6. 이용 종료 시각    7. 사용 인원   **– 스터디룸 등록**  CREATE OR REPLACE PROCEDURE studyroom\_insert (  pseq IN tblBookingInfo.seq%TYPE,  pclassroom IN tblBookingInfo.classroom\_seq%TYPE,  pconfirm IN tblBookingInfo.confirm\_date%TYPE,  preserve IN tblBookingInfo.reserve\_date%TYPE,  pstart IN tblBookingInfo.start\_time%TYPE,  pend IN tblBookingInfo.end\_time%TYPE  )  IS  BEGIN  INSERT INTO tblBookingInfo (  seq,  classroom\_seq,  confirm\_date,  reserve\_date,  start\_time,  end\_time  ) VALUES (  pseq,  pclassroom,  pconfirm,  preserve,  pstart,  pend  );    COMMIT;  DBMS\_OUTPUT.PUT\_LINE('등록 완료.');    END studyroom\_insert;  /  **--스터디룸 등록 호출**  BEGIN  studyroom\_insert (  BOOKINGINFO\_SEQ.NEXTVAL,  5,  SYSDATE,  '2024-08-30',  TO\_TIMESTAMP('2024-08-05 18:56:45', 'YYYY-MM-DD HH24:MI:SS'),  TO\_TIMESTAMP('2024-08-05 19:56:33', 'YYYY-MM-DD HH24:MI:SS'));  END;  /  **--스터디룸 수정**  CREATE OR REPLACE PROCEDURE studyroom\_mod (  pseq IN tblBookingInfo.seq%TYPE,  pclassroom IN tblBookingInfo.classroom\_seq%TYPE,  pconfirm IN tblBookingInfo.confirm\_date%TYPE,  preserve IN tblBookingInfo.reserve\_date%TYPE,  pstart IN tblBookingInfo.start\_time%TYPE,  pend IN tblBookingInfo.end\_time%TYPE  )  IS  BEGIN  UPDATE tblBookingInfo  SET  classroom\_seq = pclassroom,  confirm\_date = pconfirm,  reserve\_date = preserve,  start\_time = pstart,  end\_time = pend  WHERE seq = pseq;    DBMS\_OUTPUT.PUT\_LINE('수정완료.');  COMMIT;    EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('수정실패.');  ROLLBACK;  END studyroom\_mod;  /  **--스터디룸 수정 호출**  BEGIN  studyroom\_mod (  pseq => 143,  pclassroom => 1,  pconfirm => sysdate,  preserve => '2024-12-25',  pstart => '2024-01-01',  pend =>'2024-01-01'  );  END;  /  **--스터디룸 삭제**  CREATE OR REPLACE PROCEDURE studyroom\_delete (  pseq IN tblBookingInfo.seq%TYPE  )  IS  BEGIN  DELETE FROM tblBookingInfo  WHERE seq = pseq;    DBMS\_OUTPUT.PUT\_LINE('삭제완료.');  COMMIT;  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('삭제실패.');  ROLLBACK;  END studyroom\_delete;  /  **--스터디룸 삭제 호출**  BEGIN  studyroom\_delete (  pseq => 143  );  END;  /  **--스터디룸 조회**  CREATE OR REPLACE PROCEDURE studyroom\_view  AS  BEGIN  FOR c IN (  SELECT  DISTINCT(SI.name) AS 학생명,  CS.name AS 과정명,  CR.name AS 스터디룸,  CR.capacity AS 수용인원,  BI.reserve\_date AS 예약날짜,  BI.start\_time AS 이용시각,  BI.end\_time AS 종료시각    FROM tblStudent\_BookingInfo SBI  INNER JOIN tblBookingInfo BI  ON SBI.bookinginfo\_seq = BI.seq  INNER JOIN tblClassroom CR  ON BI.classroom\_seq = CR.seq  INNER JOIN tblOpenClass OC  ON CR.seq = OC.classroom\_seq  INNER JOIN tblStudent S  ON OC.seq = S.openclass\_seq  INNER JOIN tblStudentInfo SI  ON S.studentinfo\_seq = SI.seq  INNER JOIN tblClass CS  ON OC.class\_seq = CS.seq  WHERE ROWNUM <= 1000 --1000개만 출력  )  LOOP  DBMS\_OUTPUT.PUT\_LINE('===============학생 정보 및 스터디룸 이용 내역================');  DBMS\_OUTPUT.PUT\_LINE('학생명: '||c.학생명);  DBMS\_OUTPUT.PUT\_LINE('과정명: '||c.과정명);  DBMS\_OUTPUT.PUT\_LINE('스터디룸: '||c.스터디룸);  DBMS\_OUTPUT.PUT\_LINE('수용인원: '||c.수용인원);  DBMS\_OUTPUT.PUT\_LINE('예약날짜: '||c.예약날짜);  DBMS\_OUTPUT.PUT\_LINE('이용시각: '||c.이용시각);  DBMS\_OUTPUT.PUT\_LINE('종료시각: '||c.종료시각);  DBMS\_OUTPUT.PUT\_LINE('');  END LOOP;    END studyroom\_view;  /  **--스터디룸 조회 호출**  BEGIN  studyroom\_view;  END;  / | | | |