**DB Object 스크립트(PL-SQL)**

**1조**

| **<프로시저>** |
| --- |

| 요구사항번호 | **B-01(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 기초 정보 관리 기능 | 작성자 | 이도훈 |
| DML | | | |
| /--1.과정명 관리(과정명 등록, 수정, 삭제, 조회)  --입력  CREATE OR REPLACE PROCEDURE procAddCurriculum (  p\_curriculum\_name VARCHAR2  )  IS  BEGIN  BEGIN  INSERT INTO tblCurriculum (curriculum\_num, curriculum)  VALUES (curriculum\_seq.NEXTVAL, p\_curriculum\_name);  DBMS\_OUTPUT.PUT\_LINE('과정이 추가되었습니다.');  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('오류 발생: ' || SQLERRM);  END;  END procAddCurriculum;  --수정  CREATE OR REPLACE PROCEDURE procUpdatecurriculum (  p\_curriculum\_num NUMBER,  p\_new\_curriculum\_name VARCHAR2  )  IS  BEGIN  UPDATE tblCurriculum  SET curriculum = p\_new\_curriculum\_name  WHERE curriculum\_num = p\_curriculum\_num;  END procUpdatecurriculum;  --삭제  CREATE OR REPLACE PROCEDURE procDeleteCurriculum (  p\_curriculum\_num NUMBER  )  IS  BEGIN  DELETE FROM tblCurriculum  WHERE curriculum\_num = p\_curriculum\_num;  END procDeleteCurriculum;  --출력  CREATE OR REPLACE PROCEDURE procInfoCurriculum  IS  BEGIN  FOR cur\_record IN (SELECT curriculum\_num AS 과정번호, curriculum AS 과정명 FROM tblCurriculum)  LOOP  DBMS\_OUTPUT.PUT\_LINE('과정번호: ' || cur\_record.과정번호 || ', 과정명: ' || cur\_record.과정명);  END LOOP;  END procInfoCurriculum;  -–3. 강의실명 관리(강의실명을 등록, 수정, 삭제, 조회)  --등록  CREATE OR REPLACE PROCEDURE procAddClassroom (  p\_classroom\_name VARCHAR2,  p\_total\_people NUMBER  )  IS  BEGIN  INSERT INTO tblclassroom (classroom\_num, classroom, total\_people)  VALUES (classroom\_seq.NEXTVAL, p\_classroom\_name, p\_total\_people);  END procAddClassroom;  --수정  CREATE OR REPLACE PROCEDURE procUpdateClassroom (  p\_classroom\_num NUMBER,  p\_new\_classroom\_name VARCHAR2  )  IS  BEGIN  UPDATE tblClassroom  SET classroom = p\_new\_classroom\_name  WHERE classroom\_num = p\_classroom\_num;  END procUpdateClassroom; -- 강의실번호, 강의실명  CREATE OR REPLACE PROCEDURE procUpdateClassroom (  p\_classroom\_name VARCHAR2,  p\_new\_total\_people NUMBER  )  IS  BEGIN  UPDATE tblClassroom  SET total\_people = p\_new\_total\_people  WHERE classroom = p\_classroom\_name;  END procUpdateClassroom; -- 강의실명, 정원  --삭제  CREATE OR REPLACE PROCEDURE procDeleteClassroom (  p\_classroom\_num NUMBER  )  IS  BEGIN  DELETE FROM tblClassroom  WHERE classroom\_num = p\_classroom\_num;  END procDeleteClassroom;  --조회  CREATE OR REPLACE PROCEDURE procInfoClassroom  IS  BEGIN  FOR classroom\_record IN (SELECT classroom\_num AS 강의실번호, classroom AS 강의실명, total\_people AS 정원 FROM tblClassroom)  LOOP  DBMS\_OUTPUT.PUT\_LINE('강의실번호: ' || classroom\_record.강의실번호 || ', 강의실명: ' || classroom\_record.강의실명 || ', 정원: ' || classroom\_record.정원);  END LOOP;  END procInfoClassroom;  --교재 관리  --1.입력  CREATE OR REPLACE PROCEDURE procAddBook (  p\_book\_name VARCHAR2,  p\_publisher VARCHAR2  )  IS  BEGIN  INSERT INTO tblBook (book\_num, book, publisher)  VALUES (book\_seq.nextVal, p\_book\_name, p\_publisher);    DBMS\_OUTPUT.PUT\_LINE('교재명: ' || p\_book\_name);  DBMS\_OUTPUT.PUT\_LINE('출판사: ' || p\_publisher);  END procAddBook;  --2. 출력  CREATE OR REPLACE PROCEDURE procInfoBook  IS  BEGIN  FOR book\_record IN (SELECT book\_num AS 교재번호, book AS 제목, publisher AS 출판사 FROM tblBook)  LOOP  DBMS\_OUTPUT.PUT\_LINE('교재번호: ' || book\_record.교재번호 || ', 제목: ' || book\_record.제목 || ', 출판사: ' || book\_record.출판사);  END LOOP;  END procInfoBook;    --3-1. 교재명 수정  CREATE OR REPLACE PROCEDURE procUpdateBook (  p\_book\_num NUMBER,  p\_new\_title VARCHAR2  )  IS  BEGIN  UPDATE tblBook  SET book = p\_new\_title  WHERE book\_num = p\_book\_num;  END procUpdateBook;    --3-1. 출판사 수정  CREATE OR REPLACE PROCEDURE procUpdateBook (  p\_book\_num NUMBER,  p\_new\_publisher VARCHAR2  )  IS  BEGIN  UPDATE tblBook  SET bookPublisher = p\_new\_publisher  WHERE book\_num = p\_book\_num;  END procUpdateBook;  --4. 삭제  CREATE OR REPLACE PROCEDURE procDeleteBook (  p\_book\_num NUMBER  )  IS  BEGIN  DELETE FROM tblBook  WHERE book\_num = p\_book\_num;  END procDeleteBook; | | | |

| 요구사항번호 | **B-02(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 교사 계정 관리 | 작성자 | 황주원 |
| DML | | | |
| --1. 교사 정보 등록(이름, 주민번호 뒷자리, 전화번호, 상태(대기, 강의)를 입력하여 교사정보를 등록)  create or replace procedure procprofessorInsert(  p\_seq in tblprofessor.PROFESSOR\_NUM%type,  p\_name in tblprofessor.NAME%type,  p\_pw in tblprofessor.pw%type,  p\_state IN tblprofessor.STATE%TYPE,  p\_tel in tblprofessor.tel%type)  is  begin  insert into TBLPROFESSOR values(p\_seq, p\_name, p\_pw, p\_state, p\_tel);  end procprofessorInsert;  BEGIN  procprofessorInsert(11,'황주원', '111111','강의','010-1111-1111');  END;  --2. 교사 정보 수정  CREATE OR REPLACE PROCEDURE procprofessorUpdate(  p\_professor\_num IN tblProfessor.PROFESSOR\_NUM%TYPE,  p\_name IN tblProfessor.NAME%TYPE,  p\_pw IN tblProfessor.PW%TYPE,  p\_state IN tblProfessor.STATE%TYPE,  p\_tel IN tblProfessor.TEL%TYPE  )  IS  BEGIN  UPDATE tblProfessor  SET  name = p\_name,  pw = p\_pw,  state = p\_state,  tel = p\_tel  WHERE professor\_num = p\_professor\_num;    END procprofessorUpdate;  BEGIN  procprofessorUpdate(  p\_professor\_num => 1,  p\_name => '이길동',  p\_pw => '1111111',  p\_state => '대기',  p\_tel => '010-1111-1111'  );  END;  --3. 교사 정보 삭제  CREATE OR REPLACE PROCEDURE procprofessorDelete(  p\_professor\_num IN tblProfessor.PROFESSOR\_NUM%TYPE  )  IS  BEGIN  UPDATE tblProfessor  SET NAME = '-',  PW = '-',  STATE = '-',  TEL = '-'  WHERE PROFESSOR\_NUM = p\_professor\_num;  END procprofessorDelete;  BEGIN  procprofessorDelete(1);  END;  --4. 교사 정보 조회(이름, 주민번호 뒷자리, 전화번호, 강의 가능 과목)을 출력한다.  CREATE OR REPLACE PROCEDURE procprofessorRead(p\_cursor OUT SYS\_REFCURSOR)  IS  BEGIN  OPEN p\_cursor FOR  SELECT  p.NAME AS "이름",  p.PW AS "주민번호뒷자리",  p.TEL AS "전화번호",  s.SUBJECT AS "강의가능과목"  FROM tblProfessor p  INNER JOIN tblAbleSubject a  ON p.professor\_num = a.PROFESSOR\_NUM  INNER JOIN tblsubject s  ON a.SUBJECT\_NUM = s.SUBJECT\_NUM;  END procprofessorRead;  – 출력 선언  DECLARE  p\_cursor SYS\_REFCURSOR;  v\_name VARCHAR2(100);  v\_pw VARCHAR2(100);  v\_tel VARCHAR2(100);  v\_subject VARCHAR2(100);  BEGIN  procprofessorRead(p\_cursor);  LOOP  FETCH p\_cursor INTO v\_name, v\_pw, v\_tel, v\_subject;  EXIT WHEN p\_cursor%NOTFOUND;  dbms\_output.put\_line('이름: ' || v\_name);  dbms\_output.put\_line('주민번호뒷자리: ' || v\_pw);  dbms\_output.put\_line('전화번호: ' || v\_tel);  dbms\_output.put\_line('강의가능과목: ' || v\_subject);  dbms\_output.put\_line('--------------------');  END LOOP;  CLOSE p\_cursor;  END;  /\*교사 강의 가능 과목 정보\*/  -- 1. 교사 강의 가능 과목 입력  CREATE OR REPLACE PROCEDURE procablesubjectInsert  (  p\_professor\_name IN tblProfessor.NAME%TYPE,  p\_subject\_name IN tblSubject.SUBJECT%TYPE  )  IS  BEGIN  INSERT INTO tblAbleSubject  VALUES  (  able\_subject\_seq.nextval,  (SELECT professor\_num FROM tblProfessor WHERE name = p\_professor\_name),  (SELECT subject\_num FROM tblSubject WHERE subject = p\_subject\_name)  );    END procablesubjectInsert;  BEGIN  procablesubjectInsert('강서윤', '자바');  END;  -- 2. 교사 강의 가능 과목 출력  CREATE OR REPLACE PROCEDURE procablesubjectRead(p\_cursor OUT SYS\_REFCURSOR)  IS  BEGIN  OPEN p\_cursor FOR  SELECT  p.NAME AS "교사명",  s.SUBJECT AS "과목명"  FROM tblSubject s  INNER JOIN tblablesubject a  ON s.SUBJECT\_NUM = a.SUBJECT\_NUM  INNER JOIN tblprofessor p  ON a.PROFESSOR\_NUM = p.PROFESSOR\_NUM;  END procablesubjectRead;  – 출력선언  DECLARE  p\_cursor SYS\_REFCURSOR;  v\_teacher\_name VARCHAR2(100);  v\_subject\_name VARCHAR2(100);  BEGIN  procablesubjectRead(p\_cursor);  LOOP  FETCH p\_cursor INTO v\_teacher\_name, v\_subject\_name;  EXIT WHEN p\_cursor%NOTFOUND;  dbms\_output.put\_line('교사명: ' || v\_teacher\_name);  dbms\_output.put\_line('과목명: ' || v\_subject\_name);  dbms\_output.put\_line('--------------------');  END LOOP;  CLOSE p\_cursor;  END;  -- 3. 교사 강의 가능 과목 수정  CREATE OR REPLACE PROCEDURE procablesubjectUpdate(  p\_able\_subject\_num IN tblAbleSubject.ABLE\_SUBJECT\_NUM%TYPE,  p\_professor\_num IN tblAbleSubject.PROFESSOR\_NUM%TYPE,  p\_subject\_num IN tblAbleSubject.SUBJECT\_NUM%TYPE)  IS  BEGIN  UPDATE tblAbleSubject a  SET  a.PROFESSOR\_NUM = p\_professor\_num,  a.SUBJECT\_NUM = p\_subject\_num  WHERE a.ABLE\_SUBJECT\_NUM = p\_able\_subject\_num;    END procablesubjectUpdate;  BEGIN  procablesubjectUpdate(1,2,2);  END;  /\* 전체 교사 명단 출력\*/  CREATE OR REPLACE PROCEDURE procallprofessor(p\_cursor OUT SYS\_REFCURSOR)  IS  BEGIN  OPEN p\_cursor FOR  SELECT  p.NAME AS "이름",  p.PW AS "주민번호뒷자리",  p.TEL AS "전화번호",  s.SUBJECT AS "과목명"  FROM tblprofessor p  INNER JOIN tblablesubject a  ON p.PROFESSOR\_NUM = a.PROFESSOR\_NUM  INNER JOIN tblSubject s  ON s.SUBJECT\_NUM = a.SUBJECT\_NUM;  END procallprofessor;  DECLARE  p\_cursor SYS\_REFCURSOR;  v\_name VARCHAR2(100);  v\_pw VARCHAR2(100);  v\_tel VARCHAR2(100);  v\_subject VARCHAR2(100);  BEGIN  procallprofessor(p\_cursor);  LOOP  FETCH p\_cursor INTO v\_name, v\_pw, v\_tel, v\_subject;  EXIT WHEN p\_cursor%NOTFOUND;  dbms\_output.put\_line('이름: ' || v\_name);  dbms\_output.put\_line('주민번호뒷자리: ' || v\_pw);  dbms\_output.put\_line('전화번호: ' || v\_tel);  dbms\_output.put\_line('과목명: ' || v\_subject);  dbms\_output.put\_line('--------------------');  END LOOP;  CLOSE p\_cursor;  END;  /\* 특정 교사 정보 출력\*/  CREATE OR REPLACE PROCEDURE procspecificprofessor  (  p\_professor\_num IN tblProfessor.PROFESSOR\_NUM%TYPE,  p\_cursor OUT SYS\_REFCURSOR  )  IS  BEGIN  OPEN p\_cursor FOR  SELECT  p.NAME AS 교사명,  s.SUBJECT AS "과목명",  o.START\_DATE AS "과목 시작일",  o.END\_DATE AS "과목 종료일",  b.BOOK AS "교재명",  c.CLASSROOM AS "강의실",  CASE  WHEN o.END\_DATE < SYSDATE THEN '강의종료'  WHEN o.START\_DATE > SYSDATE THEN '강의예정'  ELSE '강의중'  END AS "강의 여부"  FROM tblopensubject o  INNER JOIN tblablesubject a ON a.ABLE\_SUBJECT\_NUM = o.ABLE\_SUBJECT\_NUM  INNER JOIN tblsubject s ON s.SUBJECT\_NUM = a.SUBJECT\_NUM  INNER JOIN tblBook b ON o.BOOK\_NUM = b.BOOK\_NUM  INNER JOIN tblOpenCurriculum oc ON o.OPEN\_CURRICULUM\_NUM = oc.OPEN\_CURRICULUM\_NUM  INNER JOIN tblClassroom c ON c.CLASSROOM\_NUM = oc.CLASSROOM\_NUM  INNER JOIN tblProfessor p ON a.PROFESSOR\_NUM = p.PROFESSOR\_NUM  WHERE p.PROFESSOR\_NUM = p\_professor\_num;  END procspecificprofessor;  – 출력 구문  DECLARE  p\_cursor SYS\_REFCURSOR;  v\_professor\_name varchar2(200);  v\_subject\_name VARCHAR2(200);  v\_start\_date DATE;  v\_end\_date DATE;  v\_book\_name VARCHAR2(200);  v\_classroom\_name VARCHAR2(200);  v\_lecture\_status VARCHAR2(200);  BEGIN  procspecificprofessor(1, p\_cursor);  LOOP  FETCH p\_cursor INTO v\_professor\_name,v\_subject\_name, v\_start\_date, v\_end\_date, v\_book\_name, v\_classroom\_name, v\_lecture\_status;  EXIT WHEN p\_cursor%NOTFOUND;  dbms\_output.put\_line('교사명: ' || v\_professor\_name);  dbms\_output.put\_line('과목명: ' || v\_subject\_name);  dbms\_output.put\_line('과목 시작일: ' || TO\_CHAR(v\_start\_date, 'YYYY-MM-DD'));  dbms\_output.put\_line('과목 종료일: ' || TO\_CHAR(v\_end\_date, 'YYYY-MM-DD'));  dbms\_output.put\_line('교재명: ' || v\_book\_name);  dbms\_output.put\_line('강의실: ' || v\_classroom\_name);  dbms\_output.put\_line('강의 여부: ' || v\_lecture\_status);  dbms\_output.put\_line('--------------------');  END LOOP;  CLOSE p\_cursor;  END; | | | |

| 요구사항번호 | **B-03(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 개설 과정 관리 | 작성자 | 이정은 |
| DML | | | |
| --1. 과정 등록하기(과정번호, 과정명) / (개설과정번호, 과정번호, 강의실번호, 과정시작일, 과정종료일, 과정기간)  create or replace procedure procAddCurriculum(  pcurriculum varchar2,  presult out number --확인을 위한 반환 변수  )  is  begin  insert into tblcurriculum VALUES (curriculum\_seq.nextVal, pcurriculum);  presult := 1; --성공  exception  when others then  presult := 0; --실패  dbms\_output.put\_line('fail');  end procAddCurriculum;  declare  begin  procAddCurriculum('자바 프로그래밍 언어 심화과정');  dbms\_output.put\_line('insert success');  end;  //  create or replace procedure procAddCurriculum(  pcurriculum\_num number,  pclassroom number,  pstart\_date date,  pend\_date date,  pperiod number,  presult out number --확인을 위한 반환 변수  )  is  begin  INSERT INTO tblopencurriculum VALUES (open\_curriculum\_seq.nextVAl, pcurriculum\_num, pclassroom, pstart\_date, pend\_date,pperiod);  presult := 1; --성공  exception  when others then  presult := 0; --실패  dbms\_output.put\_line('fail');  end procAddCurriculum;  DECLARE  vcurriculum\_num number := 61;  vclassroom NUMBER := 1;  vstart\_date DATE := TO\_DATE('2023-11-30', 'yyyy-mm-dd');  vend\_date DATE := TO\_DATE('2024-05-27', 'yyyy-mm-dd');  vperiod NUMBER := 6;  vresult NUMBER;  BEGIN  procAddCurriculum(vcurriculum\_num, vclassroom, vstart\_date, vend\_date, vperiod, vresult);    -- 결과 확인  IF vresult = 1 THEN  dbms\_output.put\_line('Insert success');  ELSE  dbms\_output.put\_line('Insert failed');  END IF;  END;  --2. 수정하기  create or replace procedure procUpdateCurriculum(  pcurriculum varchar2,  presult out number --확인을 위한 반환 변수  )  is  begin  update tblcurriculum set curriculum = pcurriculum where curriculum\_num=61;  presult := 1;  exception  when others then  presult := 0; --실패  end procUpdateCurriculum;  declare  vresult number;  begin  procUpdateCurriculum('벡엔드 기초과정', vresult);  dbms\_output.put\_line(vresult);  end;  --3. 삭제하기  create or replace procedure procDelCurriculum(  popen\_curriculum\_num number,  presult out number --확인을 위한 반환 변수  )  is  begin  delete from tblopencurriculum where open\_curriculum\_num=popen\_curriculum\_num;  presult := 1; --성공  exception  when others then  presult := 0; --실패  end procDelCurriculum;  DECLARE  vresult NUMBER;  BEGIN  procDelCurriculum(23, vresult);  dbms\_output.put\_line(vresult);  END;  --4. 개설 과정 선택시, 개설과목정보(과목명, 과목시작일, 과목종료일, 교재명, 교사명)을 출력  CREATE OR REPLACE PROCEDURE procSelCurriculum(  p\_open\_curriculum\_num NUMBER,  presult OUT NUMBER  )  IS  v\_curriculum\_name tblcurriculum.curriculum%TYPE;  v\_subject\_name tblsubject.subject%TYPE;  v\_start\_date tblopensubject.start\_date%TYPE;  v\_end\_date tblopensubject.end\_date%TYPE;  v\_book\_name tblbook.book%TYPE;  v\_professor\_name tblprofessor.name%TYPE;  CURSOR vcursor IS  SELECT  c.curriculum AS 개설과정명,  s.subject AS 과목명,  os.start\_date AS 과목시작일,  os.end\_date AS 과목종료일,  b.book AS 교재명,  p.name AS 교수명  FROM tblopensubject os  INNER JOIN tblbook b ON b.book\_num = os.book\_num  INNER JOIN tblablesubject ab ON os.able\_subject\_num = ab.able\_subject\_num  INNER JOIN tblsubject s ON s.subject\_num = ab.subject\_num  INNER JOIN tblprofessor p ON p.professor\_num = ab.professor\_num  INNER JOIN tblopencurriculum oc ON oc.open\_curriculum\_num = os.open\_curriculum\_num  INNER JOIN tblcurriculum c ON c.curriculum\_num = oc.curriculum\_num  WHERE oc.open\_curriculum\_num = p\_open\_curriculum\_num;  BEGIN  OPEN vcursor;  LOOP  FETCH vcursor INTO v\_curriculum\_name, v\_subject\_name, v\_start\_date, v\_end\_date, v\_book\_name, v\_professor\_name;  exit when vcursor%notfound;    DBMS\_OUTPUT.PUT\_LINE('개설과정명 : ' || v\_curriculum\_name || ', 과목명 : ' || v\_subject\_name || ', 과목시작일 : ' || v\_start\_date  || ', 과목종료일 : ' || v\_end\_date || ', 교재명 : ' || v\_book\_name || ', 교사명 : ' || v\_professor\_name);  DBMS\_OUTPUT.PUT\_LINE('--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  END LOOP;  CLOSE vcursor;  presult := 1;  EXCEPTION  WHEN OTHERS THEN  presult := 0; --실패  END procSelCurriculum;  /  DECLARE  vresult number;  v\_open\_curriculum\_num NUMBER := 4; -- 원하는 개설 과정 번호 지정  BEGIN  procSelCurriculum(v\_open\_curriculum\_num, vresult);  --dbms\_output.put\_line(vresult);  END;  / | | | |

| 요구사항번호 | **B-04(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 개설 과목 관리 | 작성자 | 안대명 |
| DML | | | |
| -- 개설과목 추가  CREATE OR REPLACE PROCEDURE procAddOpenSubject(  p\_open\_subject\_num NUMBER,  p\_open\_curriculum\_num NUMBER,  p\_able\_subject\_num NUMBER,  p\_start\_date DATE,  p\_end\_date DATE,  p\_sub\_progress\_state VARCHAR2,  p\_result OUT NUMBER  )  IS  BEGIN  INSERT INTO tblOpenSubject (open\_subject\_num, open\_curriculum\_num, able\_subject\_num, start\_date, end\_date, sub\_progress\_state)  VALUES (p\_open\_subject\_num, p\_open\_curriculum\_num, p\_able\_subject\_num, p\_start\_date, p\_end\_date, p\_sub\_progress\_state);    p\_result := 1; -- 성공  EXCEPTION  WHEN OTHERS THEN  p\_result := 0; -- 실패  END procAddOpenSubject;  -- 개설과목 수정  CREATE OR REPLACE PROCEDURE procUpdateOpenSubject(  p\_open\_subject\_num NUMBER,  p\_sub\_progress\_state VARCHAR2,  p\_result OUT NUMBER  )  IS  BEGIN  UPDATE tblOpenSubject SET sub\_progress\_state = p\_sub\_progress\_state WHERE open\_subject\_num = p\_open\_subject\_num;    p\_result := 1; -- 성공  EXCEPTION  WHEN OTHERS THEN  p\_result := 0; -- 실패  END procUpdateOpenSubject;  -- 개설과목 삭제  CREATE OR REPLACE PROCEDURE procDeleteOpenSubject(  p\_open\_subject\_num NUMBER,  p\_result OUT NUMBER  )  IS  BEGIN  DELETE FROM tblOpenSubject WHERE open\_subject\_num = p\_open\_subject\_num;    p\_result := 1; -- 성공  EXCEPTION  WHEN OTHERS THEN  p\_result := 0; -- 실패  END procDeleteOpenSubject;  -- 개설과정 선택 시 개설과목 정보 출력  CREATE OR REPLACE PROCEDURE procSelCurriculum(  p\_open\_curriculum\_num NUMBER,  p\_result OUT NUMBER  )  IS  BEGIN  FOR rec IN (  SELECT  OC.curriculum\_num AS 과정번호,  C.curriculum AS 과정명,  OC.start\_date AS 과정시작일,  OC.end\_date AS 과정종료일,  OC.classroom\_num AS 강의실번호,  OS.open\_subject\_num AS 개설과목번호,  OS.able\_subject\_num AS 강의가능과목번호,  S.subject AS 과목명,  OS.start\_date AS 과목시작일,  OS.end\_date AS 과목종료일,  OS.book\_num AS 교재번호,  B.book AS 교재명,  OS.sub\_progress\_state AS 강의진행여부  FROM tblOpenCurriculum OC  INNER JOIN tblCurriculum C ON OC.curriculum\_num = C.curriculum\_num  INNER JOIN tblOpenSubject OS ON OC.open\_curriculum\_num = OS.open\_curriculum\_num  INNER JOIN tblAbleSubject ASB ON OS.able\_subject\_num = ASB.able\_subject\_num  INNER JOIN tblSubject S ON ASB.subject\_num = S.subject\_num  LEFT JOIN tblBook B ON OS.book\_num = B.book\_num  WHERE OC.open\_curriculum\_num = p\_open\_curriculum\_num  ) LOOP  DBMS\_OUTPUT.PUT\_LINE('과정번호: ' || rec.과정번호);  DBMS\_OUTPUT.PUT\_LINE('과정명: ' || rec.과정명);  DBMS\_OUTPUT.PUT\_LINE('과정시작일: ' || TO\_CHAR(rec.과정시작일, 'YYYY-MM-DD'));  DBMS\_OUTPUT.PUT\_LINE('과정종료일: ' || TO\_CHAR(rec.과정종료일, 'YYYY-MM-DD'));  DBMS\_OUTPUT.PUT\_LINE('강의실번호: ' || rec.강의실번호);  DBMS\_OUTPUT.PUT\_LINE('개설과목번호: ' || rec.개설과목번호);  DBMS\_OUTPUT.PUT\_LINE('강의가능과목번호: ' || rec.강의가능과목번호);  DBMS\_OUTPUT.PUT\_LINE('과목명: ' || rec.과목명);  DBMS\_OUTPUT.PUT\_LINE('과목시작일: ' || TO\_CHAR(rec.과목시작일, 'YYYY-MM-DD'));  DBMS\_OUTPUT.PUT\_LINE('과목종료일: ' || TO\_CHAR(rec.과목종료일, 'YYYY-MM-DD'));  DBMS\_OUTPUT.PUT\_LINE('교재번호: ' || rec.교재번호);  DBMS\_OUTPUT.PUT\_LINE('교재명: ' || rec.교재명);  DBMS\_OUTPUT.PUT\_LINE('강의진행여부: ' || rec.강의진행여부);  DBMS\_OUTPUT.PUT\_LINE('----------------------------------------');  END LOOP;    p\_result := 1; -- 성공  EXCEPTION  WHEN OTHERS THEN  p\_result := 0; -- 실패  END procSelCurriculum; | | | |

| 요구사항번호 | **B- 05(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 관리 | 작성자 | 이도훈 |
| DML | | | |
| --출력  --교육생기본정보  CREATE OR REPLACE PROCEDURE procBasicInfoStudent  IS  BEGIN  FOR student\_record IN (SELECT name AS 교육생이름, pw AS 주민번호뒷자리, tel AS 전화번호 FROM tblstudent)  LOOP  DBMS\_OUTPUT.PUT\_LINE('교육생이름: ' || student\_record.교육생이름 || ', 주민번호뒷자리: ' || student\_record.주민번호뒷자리 || ', 전화번호: ' || student\_record.전화번호);  END LOOP;  END procBasicInfoStudent;  --교육생정보  CREATE OR REPLACE PROCEDURE procStudentInfo  IS  BEGIN  FOR curriculum\_record IN (  SELECT  s.name AS 교육생이름,  s.pw AS 주민번호뒷자리,  s.tel AS 전화번호,  r.register\_date AS 등록일,  c.curriculum AS 과정명  FROM tblstudent s  INNER JOIN tblregister r ON s.student\_num = r.register\_num  INNER JOIN tblOpenCurriculum o ON o.open\_curriculum\_num = r.open\_curriculum\_num  INNER JOIN tblCurriculum c ON c.curriculum\_num = o.curriculum\_num  )  LOOP  DBMS\_OUTPUT.PUT\_LINE('교육생이름: ' || curriculum\_record.교육생이름 || ', 주민번호뒷자리: ' || curriculum\_record.주민번호뒷자리 || ', 전화번호: ' || curriculum\_record.전화번호 || ', 등록일: ' || curriculum\_record.등록일 || ', 과정명: ' || curriculum\_record.과정명);  END LOOP;  END procStudentInfo;  --특정 교육생 선택  특정 교육생 선택시 교육생이 수강 신청한 또는 수강중인 수강했던 개설 과정 정보(과정명, 과정기간(시작 년월일, 끝 년월일), 강의실, 수료 및 중도 탈락의 여부, 수료 및 중도탈락 날짜를 출력  CREATE OR REPLACE PROCEDURE procOptionStudent (  p\_student\_num NUMBER  )  IS  BEGIN  BEGIN  FOR student\_record IN (  SELECT  s.name AS 교육생명,  cur.curriculum AS 과정명,  TO\_CHAR(c.start\_date, 'YYYY-MM-DD') || ' ~ ' || TO\_CHAR(c.end\_date, 'YYYY-MM-DD') AS 과정기간,  cla.classroom AS 강의실,  NVL2(co.complete\_date, 'O', 'X') AS 수료,  NVL2(d.drop\_date, 'O', 'X') AS 중도탈락,  co.complete\_date AS 수료날짜,  d.register\_num AS 중도탈락날짜  FROM tblopencurriculum c  INNER JOIN tblRegister r ON c.open\_curriculum\_num = r.open\_curriculum\_num  INNER JOIN tblStudent s ON s.student\_num = r.student\_num  LEFT OUTER JOIN tblcomplete co ON r.register\_num = co.register\_num  LEFT OUTER JOIN tblDrop d ON d.register\_num = r.register\_num  INNER JOIN tblCurriculum cur ON c.curriculum\_num = cur.curriculum\_num  INNER JOIN tblclassroom cla ON cla.classroom\_num = c.classroom\_num  WHERE s.student\_num = p\_student\_num  )  LOOP  DBMS\_OUTPUT.PUT\_LINE(  '교육생명: ' || student\_record.교육생명 || ', ' ||  '과정명: ' || student\_record.과정명 || ', ' ||  '과정기간: ' || student\_record.과정기간 || ', ' ||  '강의실: ' || student\_record.강의실 || ', ' ||  '수료: ' || student\_record.수료 || ', ' ||  '중도탈락: ' || student\_record.중도탈락 || ', ' ||  '수료날짜: ' || TO\_CHAR(student\_record.수료날짜, 'YYYY-MM-DD HH24:MI:SS') || ', ' ||  '중도탈락날짜: ' || TO\_CHAR(student\_record.중도탈락날짜, 'YYYY-MM-DD HH24:MI:SS')  );  dbms\_output.put\_line('-----------------------------------------------------------------------------------------------------------------------------------------------------------------------');  END LOOP;  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('오류 발생: ' || SQLERRM);  END;  END procOptionStudent;  --특정 교육생 정보 검색  특정 교육생 정보를 쉽게 확인하기 위한 검색 기능을 사용할 수 있어야 한다.  CREATE OR REPLACE PROCEDURE procSearchStudent(  p\_student\_num NUMBER  )  IS  BEGIN  FOR student\_info IN (  SELECT  s.name AS 교육생이름,  s.tel AS 전화번호,  curr.curriculum AS 과정명,  TO\_CHAR(c.start\_date, 'YYYY-MM-DD') || ' ~ ' || TO\_CHAR(c.end\_date, 'YYYY-MM-DD') AS 과정기간,  subj.subject AS 과목명,  TO\_CHAR(su.start\_date, 'YYYY-MM-DD') || ' ~ ' || TO\_CHAR(su.end\_date, 'YYYY-MM-DD') AS 과목기간  FROM tblstudent s  INNER JOIN tblRegister r ON s.student\_num = r.student\_num  INNER JOIN tblOpenCurriculum c ON c.open\_curriculum\_num = r.open\_curriculum\_num  INNER JOIN tblOpenSubject su ON su.open\_curriculum\_num = c.open\_curriculum\_num  INNER JOIN tblAbleSubject able ON able.able\_subject\_num = su.able\_subject\_num  INNER JOIN tblSubject subj ON subj.subject\_num = able.subject\_num  INNER JOIN tblCurriculum curr ON curr.curriculum\_num = c.curriculum\_num  WHERE s.student\_num = p\_student\_num  )  LOOP  DBMS\_OUTPUT.PUT\_LINE('교육생 이름: ' || student\_info.교육생이름 || ', ' ||  '전화번호: ' || student\_info.전화번호 || ', ' ||  '과정명: ' || student\_info.과정명 || ', ' ||  '과정기간: ' || student\_info.과정기간 || ', ' ||  '과목명: ' || student\_info.과목명 || ', ' ||  '과목기간: ' || student\_info.과목기간);  END LOOP;  dbms\_output.put\_line('-----------------------------------------------------------------------------------------------------------------------------------------------------------------------');  END procSearchStudent;  --교육생에 대한 수료 및 중도 탈락 처리를 할 수 있어야 한다. 수료 또는 중도탈락 날짜를 입력할 수 있어야 한다.  CREATE OR REPLACE PROCEDURE procOptionStudent (  p\_student\_num NUMBER  )  IS  BEGIN  DECLARE  student\_found BOOLEAN := FALSE;  BEGIN  FOR student\_record IN (  SELECT  s.name AS 교육생명,  cur.curriculum AS 과정명,  TO\_CHAR(c.start\_date, 'YYYY-MM-DD') || ' ~ ' || TO\_CHAR(c.end\_date, 'YYYY-MM-DD') AS 과정기간,  cla.classroom AS 강의실,  NVL2(co.complete\_date, 'O', 'X') AS 수료,  NVL2(d.drop\_date, 'O', 'X') AS 중도탈락,  co.complete\_date AS 수료날짜,  d.register\_num AS 중도탈락날짜  FROM tblopencurriculum c  INNER JOIN tblRegister r ON c.open\_curriculum\_num = r.open\_curriculum\_num  INNER JOIN tblStudent s ON s.student\_num = r.student\_num  LEFT OUTER JOIN tblcomplete co ON r.register\_num = co.register\_num  LEFT OUTER JOIN tblDrop d ON d.register\_num = r.register\_num  INNER JOIN tblCurriculum cur ON c.curriculum\_num = cur.curriculum\_num  INNER JOIN tblclassroom cla ON cla.classroom\_num = c.classroom\_num  WHERE s.student\_num = p\_student\_num  )  LOOP  student\_found := TRUE;  DBMS\_OUTPUT.PUT\_LINE(  '교육생명: ' || student\_record.교육생명 || ', ' ||  '과정명: ' || student\_record.과정명 || ', ' ||  '과정기간: ' || student\_record.과정기간 || ', ' ||  '강의실: ' || student\_record.강의실 || ', ' ||  '수료: ' || student\_record.수료 || ', ' ||  '중도탈락: ' || student\_record.중도탈락 || ', ' ||  '수료날짜: ' || TO\_CHAR(student\_record.수료날짜, 'YYYY-MM-DD HH24:MI:SS') || ', ' ||  '중도탈락날짜: ' || TO\_CHAR(student\_record.중도탈락날짜, 'YYYY-MM-DD HH24:MI:SS')  );  DBMS\_OUTPUT.PUT\_LINE('-----------------------------------------------------------------------------------------------------------------------------------------------------------------------');  END LOOP;  IF NOT student\_found THEN  DBMS\_OUTPUT.PUT\_LINE('해당 교육생 정보를 찾을 수 없습니다.');  END IF;  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('오류 발생: ' || SQLERRM);  END;  END procOptionStudent;  --삭제  CREATE OR REPLACE PROCEDURE procDeleteStudent(  p\_student\_num NUMBER  )  IS  BEGIN  UPDATE tblstudent  SET name = '-',  pw = '-',  tel = '-'  WHERE student\_num = p\_student\_num;  END procDeleteStudent; | | | |

| 요구사항번호 | **B-06(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 시험 관리 및 성적 조회 | 작성자 | 황주원 |
| DML | | | |
| – 1. 시험 관리 및 성적 조회 입력  CREATE OR REPLACE PROCEDURE proctestInsert(  p\_OPEN\_SUBJECT\_NUM IN tblTest.OPEN\_SUBJECT\_NUM%TYPE,  p\_TEST\_FILE IN tblTest.TEST\_FILE%TYPE,  p\_TEST\_DATE IN tblTest.TEST\_DATE%TYPE,  p\_ATT\_RATIO IN tblTest.ATT\_RATIO%TYPE,  p\_WRITE\_RATIO IN tblTest.WRITE\_RATIO%TYPE,  p\_PRACTICAL\_RATIO IN tblTest.PRACTICAL\_RATIO%TYPE,  p\_SCORE\_REGISTRATION\_STATE IN tblTest.SCORE\_REGISTRATION\_STATE%TYPE)  IS  BEGIN  INSERT INTO tblTest  VALUES (test\_seq.nextval, p\_OPEN\_SUBJECT\_NUM, p\_TEST\_FILE, p\_TEST\_DATE, p\_ATT\_RATIO, p\_WRITE\_RATIO, p\_PRACTICAL\_RATIO, p\_SCORE\_REGISTRATION\_STATE);    END proctestInsert;  BEGIN  proctestInsert(29,'test29',sysdate, 20,20,20,'부');  END;  – 2. 시험 관리 및 성적 조회 출력  – a.특정 개설 과정을 선택 > 개설된 과목 정보를 출력하고, 개설 과목 별로 성적 등록 여부를 조회  CREATE OR REPLACE PROCEDURE procspecificcurriculumRead(  p\_curriculum\_num IN tblCurriculum.CURRICULUM\_NUM%TYPE,  p\_cursor OUT SYS\_REFCURSOR)  IS  BEGIN  OPEN p\_cursor FOR  SELECT DISTINCT  s.subject AS "과목명",  os.start\_date AS "시작날짜",  os.end\_date AS "종료날짜",  os.sub\_progress\_state AS "과목진행상태",  p.name AS "교사명",  CASE  WHEN t.score\_registration\_state = '여' THEN '등록'  WHEN t.score\_registration\_state = '부' THEN '미등록'  END AS "필기성적등록여부"  FROM tblCurriculum cr  INNER JOIN tblOpenCurriculum oc ON cr.curriculum\_num = oc.curriculum\_num  INNER JOIN tblOpenSubject os ON os.open\_curriculum\_num = oc.open\_curriculum\_num  INNER JOIN tblTest t ON os.open\_subject\_num = t.open\_subject\_num  INNER JOIN tblAbleSubject ables ON os.able\_subject\_num = ables.able\_subject\_num  INNER JOIN tblSubject s ON s.subject\_num = ables.subject\_num  INNER JOIN tblProfessor p ON p.professor\_num = ables.professor\_num  WHERE cr.curriculum\_num = p\_curriculum\_num;  END procspecificcurriculumRead;  DECLARE  p\_cursor SYS\_REFCURSOR;  v\_subject\_name VARCHAR2(100);  v\_start\_date DATE;  v\_end\_date DATE;  v\_sub\_progress\_state VARCHAR2(100);  v\_professor\_name VARCHAR2(100);  v\_score\_registration\_state VARCHAR2(100);  BEGIN  procspecificcurriculumRead(1, p\_cursor); -- 여기에 원하는 curriculum\_num 입력  LOOP  FETCH p\_cursor INTO v\_subject\_name, v\_start\_date, v\_end\_date, v\_sub\_progress\_state, v\_professor\_name, v\_score\_registration\_state;  EXIT WHEN p\_cursor%NOTFOUND;  dbms\_output.put\_line('과목명: ' || v\_subject\_name);  dbms\_output.put\_line('시작날짜: ' || TO\_CHAR(v\_start\_date, 'YYYY-MM-DD'));  dbms\_output.put\_line('종료날짜: ' || TO\_CHAR(v\_end\_date, 'YYYY-MM-DD'));  dbms\_output.put\_line('과목진행상태: ' || v\_sub\_progress\_state);  dbms\_output.put\_line('교사명: ' || v\_professor\_name);  dbms\_output.put\_line('필기성적등록여부: ' || v\_score\_registration\_state);  dbms\_output.put\_line('--------------------');  END LOOP;  CLOSE p\_cursor;  END;  – b. 개설 과목명 입력시 결과 출력  CREATE OR REPLACE PROCEDURE procsubcurriculumRead(  p\_subject\_name IN tblSubject.SUBJECT%TYPE,  p\_cursor OUT SYS\_REFCURSOR)  IS  BEGIN  OPEN p\_cursor FOR  SELECT DISTINCT  cr.curriculum AS "과정명",  s.subject AS "과목명",  oc.start\_date AS "시작날짜",  oc.end\_date AS "종료날짜",  c.classroom AS "강의실명",  c.total\_people AS "정원"  FROM tblCurriculum cr  INNER JOIN tblOpenCurriculum oc ON cr.curriculum\_num = oc.curriculum\_num  INNER JOIN tblOpenSubject os ON os.open\_curriculum\_num = oc.open\_curriculum\_num  INNER JOIN tblAbleSubject ables ON os.able\_subject\_num = ables.able\_subject\_num  INNER JOIN tblSubject s ON s.subject\_num = ables.subject\_num  INNER JOIN tblProfessor p ON p.professor\_num = ables.professor\_num  INNER JOIN tblClassroom c ON oc.classroom\_num = c.classroom\_num  WHERE s.subject = p\_subject\_name;  END procsubcurriculumRead;  DECLARE  p\_cursor SYS\_REFCURSOR;  v\_curriculum\_name VARCHAR2(200);  v\_subject\_name VARCHAR2(200);  v\_start\_date DATE;  v\_end\_date DATE;  v\_classroom\_name VARCHAR2(200);  v\_total\_people NUMBER;  BEGIN  procsubcurriculumRead('자바', p\_cursor); -- 여기에 원하는 과목명을 전달합니다.  LOOP  FETCH p\_cursor INTO v\_curriculum\_name, v\_subject\_name, v\_start\_date, v\_end\_date, v\_classroom\_name, v\_total\_people;  EXIT WHEN p\_cursor%NOTFOUND;  dbms\_output.put\_line('과정명: ' || v\_curriculum\_name);  dbms\_output.put\_line('과목명: ' || v\_subject\_name);  dbms\_output.put\_line('시작날짜: ' || TO\_CHAR(v\_start\_date, 'YYYY-MM-DD'));  dbms\_output.put\_line('종료날짜: ' || TO\_CHAR(v\_end\_date, 'YYYY-MM-DD'));  dbms\_output.put\_line('강의실명: ' || v\_classroom\_name);  dbms\_output.put\_line('정원: ' || v\_total\_people);  dbms\_output.put\_line('--------------------');  END LOOP;  CLOSE p\_cursor;  END;  – c.교육생 개인별 성적 출력시 교육생 이름, 주민번호 뒷자리, 개설 과정명, 개설 과정기간, 강의실명 등을 출력하고, 교육생 개인이 수강한 모든 개설 과목에 대한 성적 정보(개설 과목명, 개설 과목 기간, 교사명, 출력, 필기, 실기)를 같이 출력한다.  CREATE OR REPLACE PROCEDURE procstudentscoreRead  (  p\_student\_num IN NUMBER  )  IS  CURSOR StudentCourseInfoCursor IS  SELECT  s.name AS 교육생이름,  cur.curriculum AS 개설과정명,  ocur.start\_date AS 개설과정시작일,  ocur.end\_date AS 개설과정종료일,  r.classroom AS 강의실명,  sub.subject AS 개설과목명,  osub.start\_date AS 개설과목시작일,  osub.end\_date AS 개설과목종료일,  sc.att\_score AS 출결점수,  sc.write\_score AS 필기점수,  sc.practical\_score AS 실기점수  FROM tblOpenSubject osub  INNER JOIN tblOpenCurriculum ocur  ON osub.open\_curriculum\_num = ocur.open\_curriculum\_num  INNER JOIN tblCurriculum cur  ON ocur.curriculum\_num = cur.curriculum\_num  INNER JOIN tblClassroom r  ON ocur.classroom\_num = r.classroom\_num  INNER JOIN tblCourse c  ON osub.open\_subject\_num = c.open\_subject\_num  INNER JOIN tblRegister r  ON c.register\_num = r.register\_num  INNER JOIN tblStudent s  ON r.student\_num = s.student\_num  LEFT OUTER JOIN tblScore sc  ON c.course\_num = sc.course\_num  INNER JOIN tblAbleSubject asub  ON osub.able\_subject\_num = asub.able\_subject\_num  INNER JOIN tblSubject sub  ON asub.subject\_num = sub.subject\_num  WHERE s.student\_num = p\_student\_num;  BEGIN    FOR rec IN StudentCourseInfoCursor  LOOP  DBMS\_OUTPUT.PUT\_LINE('=====================================');  DBMS\_OUTPUT.PUT\_LINE('교육생이름: ' || rec.교육생이름);  DBMS\_OUTPUT.PUT\_LINE('개설과정명: ' || rec.개설과정명);  DBMS\_OUTPUT.PUT\_LINE('개설과정시작일: ' || TO\_CHAR(rec.개설과정시작일, 'YYYY-MM-DD'));  DBMS\_OUTPUT.PUT\_LINE('개설과정종료일: ' || TO\_CHAR(rec.개설과정종료일, 'YYYY-MM-DD'));  DBMS\_OUTPUT.PUT\_LINE('강의실명: ' || rec.강의실명);  DBMS\_OUTPUT.PUT\_LINE('개설과목명: ' || rec.개설과목명);  DBMS\_OUTPUT.PUT\_LINE('개설과목시작일: ' || TO\_CHAR(rec.개설과목시작일, 'YYYY-MM-DD'));  DBMS\_OUTPUT.PUT\_LINE('개설과목종료일: ' || TO\_CHAR(rec.개설과목종료일, 'YYYY-MM-DD'));  DBMS\_OUTPUT.PUT\_LINE('출결점수: ' || rec.출결점수);  DBMS\_OUTPUT.PUT\_LINE('필기점수: ' || rec.필기점수);  DBMS\_OUTPUT.PUT\_LINE('실기점수: ' || rec.실기점수);  END LOOP;  CLOSE StudentCourseInfoCursor;    END procstudentscoreRead;  BEGIN  procstudentscoreRead(83); -- 조회할 학생번호 입력  END;  – d. 과목별 출력시 개설 과정명, 개설 과정기간, 강의실명, 개설 과목명, 교사명, 교재명 등을 출력하고, 해당 개설 과목을 수강한 모든 교육생들의 성적 정보(교육생 이름, 필기, 실기)를 같이 출력한다.  CREATE OR REPLACE PROCEDURE procsubscoreRead(p\_subject\_name VARCHAR2)  IS  CURSOR StudentCourseInfoCursor IS  SELECT  s.name AS 교육생이름,  cur.curriculum AS 개설과정명,  ocur.start\_date AS 개설과정시작일,  ocur.end\_date AS 개설과정종료일,  r.classroom AS 강의실명,  sub.subject AS 개설과목명,  osub.start\_date AS 개설과목시작일,  osub.end\_date AS 개설과목종료일,  sc.att\_score AS 출결점수,  sc.write\_score AS 필기점수,  sc.practical\_score AS 실기점수  FROM tblOpenSubject osub  INNER JOIN tblOpenCurriculum ocur  ON osub.open\_curriculum\_num = ocur.open\_curriculum\_num  INNER JOIN tblCurriculum cur  ON ocur.curriculum\_num = cur.curriculum\_num  INNER JOIN tblClassroom r  ON ocur.classroom\_num = r.classroom\_num  INNER JOIN tblCourse c  ON osub.open\_subject\_num = c.open\_subject\_num  INNER JOIN tblRegister r  ON c.register\_num = r.register\_num  INNER JOIN tblStudent s  ON r.student\_num = s.student\_num  LEFT OUTER JOIN tblScore sc  ON c.course\_num = sc.course\_num  INNER JOIN tblAbleSubject asub  ON osub.able\_subject\_num = asub.able\_subject\_num  INNER JOIN tblSubject sub  ON asub.subject\_num = sub.subject\_num  WHERE sub.subject = p\_subject\_name;  BEGIN  FOR rec IN StudentCourseInfoCursor  LOOP  DBMS\_OUTPUT.PUT\_LINE('=====================================');  DBMS\_OUTPUT.PUT\_LINE('교육생이름: ' || rec.교육생이름);  DBMS\_OUTPUT.PUT\_LINE('개설과정명: ' || rec.개설과정명);  DBMS\_OUTPUT.PUT\_LINE('개설과정시작일: ' || TO\_CHAR(rec.개설과정시작일, 'YYYY-MM-DD'));  DBMS\_OUTPUT.PUT\_LINE('개설과정종료일: ' || TO\_CHAR(rec.개설과정종료일, 'YYYY-MM-DD'));  DBMS\_OUTPUT.PUT\_LINE('강의실명: ' || rec.강의실명);  DBMS\_OUTPUT.PUT\_LINE('개설과목명: ' || rec.개설과목명);  DBMS\_OUTPUT.PUT\_LINE('개설과목시작일: ' || TO\_CHAR(rec.개설과목시작일, 'YYYY-MM-DD'));  DBMS\_OUTPUT.PUT\_LINE('개설과목종료일: ' || TO\_CHAR(rec.개설과목종료일, 'YYYY-MM-DD'));  DBMS\_OUTPUT.PUT\_LINE('출결점수: ' || rec.출결점수);  DBMS\_OUTPUT.PUT\_LINE('필기점수: ' || rec.필기점수);  DBMS\_OUTPUT.PUT\_LINE('실기점수: ' || rec.실기점수);  END LOOP;  END procsubscoreRead;  BEGIN  procsubscoreRead('자바');  END;  – 3. 시험 관리 및 성적 조회 수정  CREATE OR REPLACE PROCEDURE proctestUpdate(  p\_test\_num IN tbltest.TEST\_NUM%TYPE,  p\_open\_subject\_num IN tbltest.OPEN\_SUBJECT\_NUM%TYPE,  p\_test\_file IN tbltest.TEST\_FILE%TYPE,  p\_att\_ratio IN tbltest.ATT\_RATIO%TYPE,  p\_write\_ratio IN tbltest.WRITE\_RATIO%TYPE,  p\_practical\_ratio IN tbltest.PRACTICAL\_RATIO%TYPE,  p\_score\_registration\_state IN tbltest.SCORE\_REGISTRATION\_STATE%TYPE)  IS  BEGIN  UPDATE tbltest t  SET  t.open\_subject\_num = p\_open\_subject\_num,  t.test\_file = p\_test\_file,  t.test\_date = sysdate,  t.att\_ratio = p\_att\_ratio,  t.write\_ratio = p\_write\_ratio,  t.practical\_ratio = p\_practical\_ratio,  t.score\_registration\_state = p\_score\_registration\_state  WHERE t.test\_num = p\_test\_num;    END proctestUpdate;  BEGIN  proctestUpdate(1, 2, 'test2', 10, 10, 10, '부'); --맨 앞 컬럼에 수정하고싶은 test\_num 입력  END; | | | |

| 요구사항번호 | **B-07(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 출결 관리 및 출결 조회 | 작성자 | 이정은 |
| DML | | | |
| --1. 개설 과정 선택 시 모든 교육생의 출결을 조회할 수 있다.  create or replace procedure procSelAtt(  p\_open\_curriculum\_num NUMBER,  presult OUT NUMBER  )  is  v\_open\_curriculum\_num tblcurriculum.curriculum%type;  v\_name tblstudent.name%type;  v\_att\_date tblattendance.att\_date%type;  v\_att\_state tblattendance.att\_state%type;    cursor vcursor is  SELECT  oc.open\_curriculum\_num AS 개설과정,  s.name AS 교육생,  a.att\_date AS 날짜,  a.att\_state AS 근태현황  FROM tblstudent s INNER JOIN tblregister r ON s.student\_num=r.student\_num  INNER JOIN tblattendance a ON r.register\_num=a.register\_num  INNER JOIN tblopencurriculum oc ON oc.open\_curriculum\_num = r.open\_curriculum\_num  WHERE oc.open\_curriculum\_num=4  GROUP BY a.att\_date, a.att\_state, s.name, s.student\_num, oc.open\_curriculum\_num  ORDER BY a.att\_date, s.student\_num;  begin  open vcursor;  loop  FETCH vcursor INTO v\_open\_curriculum\_num, v\_name, v\_att\_date, v\_att\_state;  exit when vcursor%notfound;    DBMS\_OUTPUT.PUT\_LINE('개설과정명 : ' || v\_open\_curriculum\_num || ', 교육생 이름 : ' || v\_name || ', 날짜 : ' || v\_att\_date || ', 근태상황 : ' || v\_att\_state);  presult := 1;  end loop;  close vcursor;  EXCEPTION  WHEN OTHERS THEN  presult := 0;  end procSelAtt;  /  DECLARE  vresult number;  v\_open\_curriculum\_num NUMBER := 4; -- 원하는 개설 과정 번호 지정  BEGIN  procSelAtt(v\_open\_curriculum\_num, vresult);  --dbms\_output.put\_line(vresult);  END;  /  --2. 특정 사람의 전체 출결 현황 조회할 수 있다.  CREATE OR REPLACE PROCEDURE procSelAtt\_personal(  p\_student\_num NUMBER,  presult OUT NUMBER  )  IS  BEGIN  FOR attendance\_record IN (  SELECT  oc.open\_curriculum\_num AS 개설과정,  s.name AS 교육생,  COUNT(CASE WHEN a.att\_state = '정상' THEN 1 END) AS 출석,  COUNT(CASE WHEN a.att\_state = '지각' THEN 1 END) AS 지각,  COUNT(CASE WHEN a.att\_state = '조퇴' THEN 1 END) AS 조퇴,  COUNT(CASE WHEN a.att\_state = '외출' THEN 1 END) AS 외출,  COUNT(CASE WHEN a.att\_state = '병가' THEN 1 END) AS 병가,  COUNT(CASE WHEN a.att\_state = '기타' THEN 1 END) AS 기타  FROM tblstudent s  INNER JOIN tblregister r ON s.student\_num = r.student\_num  INNER JOIN tblattendance a ON r.register\_num = a.register\_num  INNER JOIN tblopencurriculum oc ON oc.open\_curriculum\_num = r.open\_curriculum\_num  WHERE s.student\_num = p\_student\_num  GROUP BY oc.open\_curriculum\_num, s.name  ) LOOP  DBMS\_OUTPUT.PUT\_LINE('개설과정: ' || attendance\_record.개설과정);  DBMS\_OUTPUT.PUT\_LINE('교육생: ' || attendance\_record.교육생);  DBMS\_OUTPUT.PUT\_LINE('출석: ' || attendance\_record.출석);  DBMS\_OUTPUT.PUT\_LINE('지각: ' || attendance\_record.지각);  DBMS\_OUTPUT.PUT\_LINE('조퇴: ' || attendance\_record.조퇴);  DBMS\_OUTPUT.PUT\_LINE('외출: ' || attendance\_record.외출);  DBMS\_OUTPUT.PUT\_LINE('병가: ' || attendance\_record.병가);  DBMS\_OUTPUT.PUT\_LINE('기타: ' || attendance\_record.기타);  END LOOP;  presult := 1;  EXCEPTION  WHEN OTHERS THEN  presult := 0; --실패  END procSelAtt\_personal;  /  DECLARE  vresult number;  v\_student\_num NUMBER := 83; -- 원하는 학생 번호 지정  BEGIN  procSelAtt\_personal(v\_student\_num, vresult);  --dbms\_output.put\_line(vresult);  END;  / | | | |

| 요구사항번호 | **B-08(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 수강과정 관리 | 작성자 | 황주원 |
| DML | | | |
| – 1. 수강과정 입력  CREATE OR REPLACE PROCEDURE procregisterInsert(  p\_register\_num IN tblregister.REGISTER\_NUM%TYPE,  p\_student\_num IN tblregister.STUDENT\_NUM%TYPE,  p\_open\_curriculum\_num IN tblregister.OPEN\_CURRICULUM\_NUM%TYPE,  p\_register\_date IN tblregister.REGISTER\_DATE%TYPE)  IS  BEGIN  INSERT INTO tblregister (register\_num, student\_num, open\_curriculum\_num, register\_date)  VALUES (p\_register\_num, p\_student\_num, p\_open\_curriculum\_num, p\_register\_date);    END procregisterInsert;  BEGIN  procregisterInsert(register\_seq.nextval, 250, 1, sysdate); -- 적절한 값을 전달합니다.  END;  –2. 수강과정 출력  CREATE OR REPLACE PROCEDURE procregisterRead  (  p\_cursor OUT SYS\_REFCURSOR  )  IS  BEGIN  OPEN p\_cursor FOR  SELECT  r.REGISTER\_NUM AS "수강과정번호",  s.NAME AS "교육생이름",  c.CURRICULUM AS "과정명",  r.REGISTER\_DATE AS "등록일"  FROM tblregister r  INNER JOIN tblOpenCurriculum oc ON oc.OPEN\_CURRICULUM\_NUM = r.OPEN\_CURRICULUM\_NUM  INNER JOIN tblCurriculum c ON c.CURRICULUM\_NUM = oc.CURRICULUM\_NUM  INNER JOIN tblStudent s ON s.STUDENT\_NUM = r.STUDENT\_NUM;  END procregisterRead;  DECLARE  p\_cursor SYS\_REFCURSOR;  v\_register\_num NUMBER;  v\_student\_name VARCHAR2(200);  v\_curriculum\_name VARCHAR2(200);  v\_register\_date DATE;  BEGIN  procregisterRead(p\_cursor);  LOOP  FETCH p\_cursor INTO v\_register\_num, v\_student\_name, v\_curriculum\_name, v\_register\_date;  EXIT WHEN p\_cursor%NOTFOUND;  dbms\_output.put\_line('수강과정번호: ' || v\_register\_num);  dbms\_output.put\_line('교육생이름: ' || v\_student\_name);  dbms\_output.put\_line('과정명: ' || v\_curriculum\_name);  dbms\_output.put\_line('등록일: ' || TO\_CHAR(v\_register\_date, 'YYYY-MM-DD'));  dbms\_output.put\_line('--------------------');  END LOOP;  CLOSE p\_cursor;  END;  –3. 수강과정 수정  CREATE OR REPLACE PROCEDURE procregisterUpdate(  p\_register\_num IN tblregister.REGISTER\_NUM%TYPE,  p\_student\_num IN tblregister.STUDENT\_NUM%TYPE,  p\_open\_curriculum\_num IN tblregister.OPEN\_CURRICULUM\_NUM%TYPE,  p\_register\_date IN tblregister.REGISTER\_DATE%TYPE)  IS  BEGIN  UPDATE tblregister r  SET  r.STUDENT\_NUM = p\_student\_num,  r.OPEN\_CURRICULUM\_NUM = p\_open\_curriculum\_num,  r.REGISTER\_DATE = p\_register\_date  WHERE r.REGISTER\_NUM = p\_register\_num;    END procregisterUpdate;  BEGIN  procregisterUpdate(1, 2, 2, sysdate); -- 수정할 register\_num을 처음 값으로 입력합니다  END; | | | |

| 요구사항번호 | **B- 09(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 중도탈락내역 및 수료내역 관리 | 작성자 | 황주원 |
| DML | | | |
| – 1. 수료내역 등록  CREATE OR REPLACE PROCEDURE proccompleteInsert(  p\_register\_num IN tblComplete.REGISTER\_NUM%TYPE,  p\_complete\_date IN tblComplete.COMPLETE\_DATE%TYPE)  IS  BEGIN  INSERT INTO tblComplete (register\_num, complete\_date)  VALUES (p\_register\_num, p\_complete\_date);    END proccompleteInsert;  BEGIN  proccompleteInsert(83, sysdate); -- 적절한 값을 전달합니다.  END;  – 2. 수료내역 출력  CREATE OR REPLACE PROCEDURE proccompleteRead(  p\_cursor OUT SYS\_REFCURSOR)  IS  BEGIN  OPEN p\_cursor FOR  SELECT  r.REGISTER\_NUM AS "수강과정번호",  s.NAME AS "수강생이름",  c.CURRICULUM AS "과정명",  co.COMPLETE\_DATE AS "수료일"  FROM tblRegister r  INNER JOIN tblStudent s ON s.STUDENT\_NUM = r.STUDENT\_NUM  INNER JOIN tblOpenCurriculum oc ON oc.OPEN\_CURRICULUM\_NUM = r.OPEN\_CURRICULUM\_NUM  INNER JOIN tblCurriculum c ON c.CURRICULUM\_NUM = oc.CURRICULUM\_NUM  INNER JOIN tblComplete co ON co.REGISTER\_NUM = r.REGISTER\_NUM;  END proccompleteRead;  DECLARE  p\_cursor SYS\_REFCURSOR;  v\_register\_num NUMBER;  v\_student\_name VARCHAR2(200);  v\_curriculum\_name VARCHAR2(200);  v\_complete\_date DATE;  BEGIN  proccompleteRead(p\_cursor);  LOOP  FETCH p\_cursor INTO v\_register\_num, v\_student\_name, v\_curriculum\_name, v\_complete\_date;  EXIT WHEN p\_cursor%NOTFOUND;  dbms\_output.put\_line('수강과정번호: ' || v\_register\_num);  dbms\_output.put\_line('수강생이름: ' || v\_student\_name);  dbms\_output.put\_line('과정명: ' || v\_curriculum\_name);  dbms\_output.put\_line('수료일: ' || TO\_CHAR(v\_complete\_date, 'YYYY-MM-DD'));  dbms\_output.put\_line('--------------------');  END LOOP;  CLOSE p\_cursor;  END;  –3. 수료내역 수정  CREATE OR REPLACE PROCEDURE proccompleteUpdate(  p\_register\_num IN tblComplete.REGISTER\_NUM%TYPE,  p\_new\_register\_num IN tblComplete.REGISTER\_NUM%TYPE,  p\_new\_complete\_date IN tblComplete.COMPLETE\_DATE%TYPE)  IS  BEGIN    UPDATE tblComplete c  SET c.REGISTER\_NUM = p\_new\_register\_num  WHERE c.REGISTER\_NUM = p\_register\_num;  UPDATE tblComplete c  SET c.COMPLETE\_DATE = p\_new\_complete\_date  WHERE c.REGISTER\_NUM = p\_new\_register\_num;  END proccompleteUpdate;  BEGIN  proccompleteUpdate(1, 83, sysdate); -- 수정할 수강과정번호를 첫번째 값에 입력, 그 후 수강과정번호와 수료일 수정값입력  END;  – 4. 중도탈락내역 등록  CREATE OR REPLACE PROCEDURE procdropInsert(  p\_register\_num IN tbldrop.register\_num%TYPE,  p\_drop\_date IN tbldrop.drop\_date%TYPE)  IS  BEGIN  INSERT INTO tbldrop (register\_num, drop\_date)  VALUES (p\_register\_num, p\_drop\_date);    END procdropInsert;  BEGIN  procdropInsert(83, sysdate);  END;  – 5. 중도탈락내역 출력  CREATE OR REPLACE PROCEDURE procdropRead  IS  CURSOR DropInfoCursor IS  SELECT  c.curriculum AS 과정명,  s.name AS 교육생이름,  d.drop\_date AS 중도탈락일  FROM tblregister r  INNER JOIN tblstudent s ON s.student\_num = r.student\_num  INNER JOIN tbldrop d ON r.register\_num = d.register\_num  INNER JOIN tblopencurriculum oc ON r.OPEN\_CURRICULUM\_NUM = oc.OPEN\_CURRICULUM\_NUM  INNER JOIN tblcurriculum c ON c.curriculum\_num = oc.curriculum\_num;  BEGIN  FOR rec IN DropInfoCursor  LOOP  DBMS\_OUTPUT.PUT\_LINE('과정명: ' || rec.과정명);  DBMS\_OUTPUT.PUT\_LINE('교육생이름: ' || rec.교육생이름);  DBMS\_OUTPUT.PUT\_LINE('중도탈락일: ' || rec.중도탈락일);  DBMS\_OUTPUT.PUT\_LINE('=====================================');  END LOOP;  CLOSE DropInfoCursor;  END procdropRead;  BEGIN  procdropRead;  END;  – 6. 중도탈락내역 수정  CREATE OR REPLACE PROCEDURE procdropUpdate(  p\_register\_num IN tbldrop.register\_num%TYPE,  p\_new\_register\_num IN tbldrop.register\_num%TYPE,  p\_new\_drop\_date IN tbldrop.drop\_date%TYPE)  IS  BEGIN    UPDATE tbldrop d  SET d.REGISTER\_NUM = p\_new\_register\_num  WHERE d.REGISTER\_NUM = p\_register\_num;  UPDATE tbldrop d  SET d.DROP\_DATE = p\_new\_drop\_date  WHERE d.REGISTER\_NUM = p\_new\_register\_num;  END procdropUpdate;  BEGIN  procdropUpdate(7, 83, sysdate);  END; | | | |

| 요구사항번호 | **B-10(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 수강 후기 관리 | 작성자 | 안대명 |
| DML | | | |
| -- 1. 수강후기 추가 프로시저:  CREATE OR REPLACE PROCEDURE procAddReview(  p\_review\_num NUMBER,  p\_register\_num NUMBER,  p\_review\_text VARCHAR2,  p\_review\_date DATE,  p\_result OUT NUMBER  )  IS  BEGIN  INSERT INTO tblReview (review\_num, register\_num, review, review\_date)  VALUES (p\_review\_num, p\_register\_num, p\_review\_text, p\_review\_date);    p\_result := 1; -- 성공  EXCEPTION  WHEN OTHERS THEN  p\_result := 0; -- 실패  dbms\_output.put\_line('fail');  END procAddReview;  -- 2. 수강후기 수정 프로시저:  CREATE OR REPLACE PROCEDURE procUpdateReview(  p\_review\_num NUMBER,  p\_review\_text VARCHAR2,  p\_result OUT NUMBER  )  IS  BEGIN  UPDATE tblReview SET review = p\_review\_text WHERE review\_num = p\_review\_num;    p\_result := 1; -- 성공  EXCEPTION  WHEN OTHERS THEN  p\_result := 0; -- 실패  dbms\_output.put\_line('fail');  END procUpdateReview;  -- 3. 수강후기 삭제 프로시저:  CREATE OR REPLACE PROCEDURE procDeleteReview(  p\_review\_num NUMBER,  p\_result OUT NUMBER  )  IS  BEGIN  DELETE FROM tblReview WHERE review\_num = p\_review\_num;    p\_result := 1; -- 성공  EXCEPTION  WHEN OTHERS THEN  p\_result := 0; -- 실패  dbms\_output.put\_line('fail');  END procDeleteReview;  -- 수강후기 테이블 전체 정보 출력  BEGIN  FOR review\_row IN (SELECT \* FROM tblReview) LOOP  dbms\_output.put\_line('Review Number: ' || review\_row.review\_num);  dbms\_output.put\_line('Register Number: ' || review\_row.register\_num);  dbms\_output.put\_line('Review Text: ' || review\_row.review);  dbms\_output.put\_line('Review Date: ' || TO\_CHAR(review\_row.review\_date, 'YYYY-MM-DD'));  dbms\_output.put\_line('-------------------------');  END LOOP;  END; | | | |

| 요구사항번호 | **B- 11(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 취업 현황 관리 | 작성자 | 이도훈 |
| DML | | | |
| --입력  CREATE OR REPLACE PROCEDURE procAddEmployment(  p\_employment\_state VARCHAR2,  p\_field VARCHAR2,  p\_insurance\_state VARCHAR2  )  IS  BEGIN  INSERT INTO tblEmployment(register\_num, employment\_state, field, insurance\_state)  VALUES (employment\_seq.nextval, p\_employment\_state, p\_field, p\_insurance\_state);  END procAddEmployment;  --출력  -- 수강생이 수료 후, 취업관련 자료를 불러올 때엔 수강과정번호, 취업여부, 취업분야, 고용보험가입여부를 불러오도록 한다.  CREATE OR REPLACE PROCEDURE procInfoEmployment  IS  BEGIN  FOR employment\_record IN (  SELECT  stu.name AS 교육생명,  emp.register\_num AS 수강과정번호,  emp.employment\_state AS 취업여부,  emp.field AS 취업분야,  emp.insurance\_state AS 고용보험가입여부  FROM tblregister reg  INNER JOIN tblstudent stu ON stu.student\_num = reg.student\_num  LEFT OUTER JOIN tblcomplete com ON com.register\_num = reg.register\_num  INNER JOIN tblemployment emp ON emp.register\_num = com.register\_num  )  LOOP  DBMS\_OUTPUT.PUT\_LINE('교육생명: ' || employment\_record.교육생명 || ', ' ||  '수강과정번호: ' || employment\_record.수강과정번호 || ', ' ||  '취업여부: ' || employment\_record.취업여부 || ', ' ||  '취업분야: ' || employment\_record.취업분야 || ', ' ||  '고용보험가입여부: ' || employment\_record.고용보험가입여부);  END LOOP;  END procInfoEmployment; | | | |

| 요구사항번호 | **B-12(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 관리자 출퇴근내역 | 작성자 | 황주원 |
| DML | | | |
| – 1. 등록  CREATE OR REPLACE PROCEDURE procadminattInsert(  p\_admin\_attendance\_num IN tblAdminAttendance.attendance\_num%TYPE,  p\_admin\_num IN tblAdminAttendance.admin\_num%TYPE,  p\_attendance\_date IN tblAdminAttendance.att\_date%TYPE,  p\_attendance\_state IN tblAdminAttendance.att\_state%TYPE)  IS  BEGIN  INSERT INTO tblAdminAttendance (attendance\_num, admin\_num, att\_date, att\_state)  VALUES (p\_admin\_attendance\_num, p\_admin\_num, p\_attendance\_date, p\_attendance\_state);    END procadminattInsert;  BEGIN  procadminattInsert(admin\_att\_seq.nextval, 1, sysdate, '병가');  END;  – 2. 출력  CREATE OR REPLACE PROCEDURE procadminattRead  IS  CURSOR AdminAttendanceCursor IS  SELECT  aa.ATTENDANCE\_NUM AS 출퇴근번호,  a.NAME AS 관리자이름,  aa.ATT\_DATE AS 날짜,  aa.ATT\_STATE AS 근태상황  FROM tblAdminAttendance aa  LEFT OUTER JOIN tblAdmin a ON a.ADMIN\_NUM = aa.ADMIN\_NUM;  BEGIN    FOR rec IN AdminAttendanceCursor  LOOP  DBMS\_OUTPUT.PUT\_LINE('=====================================');  DBMS\_OUTPUT.PUT\_LINE('출퇴근번호: ' || rec.출퇴근번호);  DBMS\_OUTPUT.PUT\_LINE('관리자이름: ' || rec.관리자이름);  DBMS\_OUTPUT.PUT\_LINE('날짜: ' || TO\_CHAR(rec.날짜, 'YYYY-MM-DD HH24:MI:SS'));  DBMS\_OUTPUT.PUT\_LINE('근태상황: ' || rec.근태상황);  END LOOP;  CLOSE AdminAttendanceCursor;    END procadminattRead;  BEGIN  procadminattRead;  END; | | | |

| 요구사항번호 | **B-13(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 지원자 면접 관리 | 작성자 | 안대명 |
| DML | | | |
| –1. 지원자 면접 추가 프로시저:  CREATE OR REPLACE PROCEDURE procAddApplicantInterview(  p\_applicant\_num NUMBER,  p\_admin\_num NUMBER,  p\_interview\_result VARCHAR2,  p\_interview\_date DATE,  p\_result OUT NUMBER  )  IS  BEGIN  -- 먼저 지원자를 추가  INSERT INTO tblApplicant (applicant\_num, name, tel)  VALUES (p\_applicant\_num, '이정재', '010-8238-9688');    -- 추가된 지원자에 대한 면접결과 추가  INSERT INTO tblInterview (applicant\_num, admin\_num, interview\_num, interview\_date, interview\_result)  VALUES (p\_applicant\_num, p\_admin\_num, interview\_seq.nextVal, p\_interview\_date, p\_interview\_result);    p\_result := 1; -- 성공  EXCEPTION  WHEN OTHERS THEN  p\_result := 0; -- 실패  dbms\_output.put\_line('fail');  END procAddApplicantInterview;  –2. 지원자 면접 수정 프로시저:  CREATE OR REPLACE PROCEDURE procUpdateApplicantInterview(  p\_interview\_num NUMBER,  p\_interview\_result VARCHAR2,  p\_result OUT NUMBER  )  IS  BEGIN  UPDATE tblInterview SET interview\_result = p\_interview\_result WHERE interview\_num = p\_interview\_num;    p\_result := 1; -- 성공  EXCEPTION  WHEN OTHERS THEN  p\_result := 0; -- 실패  dbms\_output.put\_line('fail');  END procUpdateApplicantInterview;  3. 지원자 면접 삭제 프로시저:  CREATE OR REPLACE PROCEDURE procDeleteApplicantInterview(  p\_interview\_num NUMBER,  p\_result OUT NUMBER  )  IS  BEGIN  DELETE FROM tblInterview WHERE interview\_num = p\_interview\_num;    p\_result := 1; -- 성공  EXCEPTION  WHEN OTHERS THEN  p\_result := 0; -- 실패  dbms\_output.put\_line('fail');  END procDeleteApplicantInterview;  –4. 지원자 면접 추가, 수정, 삭제 실행 코드:  DECLARE  v\_applicant\_num NUMBER := 31; -- 지원자 번호  v\_admin\_num NUMBER := 31; -- 관리자 번호  v\_interview\_result VARCHAR2(50) := '합격'; -- 면접 결과  v\_interview\_date DATE := TO\_DATE('2023-09-05', 'YYYY-MM-DD'); -- 면접 일자  v\_add\_result NUMBER;  BEGIN  -- 지원자 면접 추가 호출  procAddApplicantInterview(v\_applicant\_num, v\_admin\_num, v\_interview\_result, v\_interview\_date, v\_add\_result);    -- 결과 확인  IF v\_add\_result = 1 THEN  dbms\_output.put\_line('Insert success');  ELSE  dbms\_output.put\_line('Insert failed');  END IF;  END;  DECLARE  v\_interview\_num\_to\_update NUMBER := 2; -- 수정할 면접 번호  v\_updated\_interview\_result VARCHAR2(50) := '합격 (수정)'; -- 수정할 면접 결과  v\_update\_result NUMBER;  BEGIN  -- 지원자 면접 수정 호출  procUpdateApplicantInterview(v\_interview\_num\_to\_update, v\_updated\_interview\_result, v\_update\_result);    -- 결과 확인  IF v\_update\_result = 1 THEN  dbms\_output.put\_line('Update success');  ELSE  dbms\_output.put\_line('Update failed');  END IF;  END;  DECLARE  v\_interview\_num\_to\_delete NUMBER := 2; -- 삭제할 면접 번호  v\_delete\_result NUMBER;  BEGIN  -- 지원자 면접 삭제 호출  procDeleteApplicantInterview(v\_interview\_num\_to\_delete, v\_delete\_result);    -- 결과 확인  IF v\_delete\_result = 1 THEN  dbms\_output.put\_line('Delete success');  ELSE  dbms\_output.put\_line('Delete failed');  END IF;  END; | | | |

| 요구사항번호 | **B-14(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 공지사항 | 작성자 | 이정은 |
| DML | | | |
| --1. 공지사항 추가하기  create or replace procedure procAddNotice(  p\_open\_curriculum\_num number,  p\_admin\_num number,  pnotice varchar2,  pnotice\_date date,  pviews number,  presult out number --확인을 위한 반환 변수  )  is  begin  insert into tblnotice VALUES (notice\_seq.nextVal, p\_open\_curriculum\_num, p\_admin\_num, pnotice, pnotice\_date, pviews);  presult := 1; --성공  exception  when others then  presult := 0; --실패  end procAddNotice;  declare  vresult number;  begin  procAddNotice(4, 5, '안녕하세요. 오늘은 서프라이즈로 쉬겠습니다.', sysdate, 350, vresult);  dbms\_output.put\_line(vresult);  end;  --2. 공지사항 수정하기  create or replace procedure procUpdateNotice(  pnotice varchar2,  presult out number --확인을 위한 반환 변수  )  is  begin  update tblnotice set notice = pnotice where notice\_num=41;  presult := 1;  exception  when others then  presult := 0; --실패  end procUpdateNotice;  declare  vresult number;  begin  procUpdateNotice('안녕하세요. 오늘은 엘리베이터 문제로 계단으로 오셔야 합니다.', vresult);  dbms\_output.put\_line(vresult);  end;  --3. 공지사항 삭제하기  create or replace procedure procDelNotice(  pnotice\_num number,  presult out number --확인을 위한 반환 변수  )  is  begin  delete from tblnotice where notice\_num=pnotice\_num;  presult := 1; --성공  exception  when others then  presult := 0; --실패  end procDelNotice;  DECLARE  vresult NUMBER;  BEGIN  procDelNotice(41, vresult);  dbms\_output.put\_line(vresult);  END;  --4. 모든 과정에 대한 공지사항을 출력한다. 출력하는 정보는 개설과정번호, 과정명, 강의실번호, 공지사항내용, 공지일, 조회수를 출력한다.  create or replace procedure procSelNotice(  presult out number  )  is  v\_open\_curriculum\_num tblcurriculum.curriculum\_num%type;  v\_curriculum tblcurriculum.curriculum%type;  v\_classroom\_num tblopencurriculum.classroom\_num%type;  v\_notice tblnotice.notice%type;  v\_notice\_date tblnotice.notice\_date%type;    cursor vcursor is  SELECT  oc.open\_curriculum\_num AS 개설과정번호,  c.curriculum AS 과정명,  oc.classroom\_num AS 강의실번호,  n.notice AS 공지사항내용,  n.notice\_date AS 공지일  FROM tblnotice n INNER JOIN tblopencurriculum oc ON oc.open\_curriculum\_num=n.open\_curriculum\_num  INNER JOIN tblcurriculum c ON c.curriculum\_num=oc.curriculum\_num;  begin  open vcursor;  loop  FETCH vcursor INTO v\_open\_curriculum\_num, v\_curriculum, v\_classroom\_num, v\_notice, v\_notice\_date;  exit when vcursor%notfound;    DBMS\_OUTPUT.PUT\_LINE('개설과정명번호 : ' || v\_open\_curriculum\_num);  DBMS\_OUTPUT.PUT\_LINE('과정명 : ' || v\_curriculum);  DBMS\_OUTPUT.PUT\_LINE('강의실번호 : ' || v\_classroom\_num);  DBMS\_OUTPUT.PUT\_LINE('공지사항내용 : ' || v\_notice);  DBMS\_OUTPUT.PUT\_LINE('공지일 : ' || v\_notice\_date);  DBMS\_OUTPUT.PUT\_LINE('----------------------------------------------------------------------');  presult := 1;  end loop;  close vcursor;  exception  when others then  presult:=0; --실패  end procSelNotice;  /  declare  vresult number;  begin  procSelNotice(vresult);  DBMS\_OUTPUT.PUT\_LINE(vresult);  end; | | | |

| 요구사항번호 | **C-01(교사)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 강의 스케줄 조회 | 작성자 | 차수민 |
| DML | | | |
| – 1. 강의 스케줄 정보 출력 > 과정번호, 과정명, 과정기간(시작 년월일 / 끝 년월일),  과목번호, 과목명, 과목기간(시작 년월일 / 끝 년월일), 강의실명, 교재명, 교육생 등록 인원  CREATE OR REPLACE PROCEDURE procSchedule(  p\_professor\_num NUMBER,  p\_subject\_progress\_state varchar2  )  IS  BEGIN  DBMS\_OUTPUT.PUT\_LINE('===============================[강의 스케줄]==============================  =');  IF p\_subject\_progress\_state = '강의 예정' THEN  FOR schedule IN (SELECT DISTINCT s.개설과정번호, s.과정명, s.과정시작일, s.과정종료일,  s.개설과목번호, s.과목명, s.과목시작일, s.과목종료일, s.강의실명, s.교재명,  (SELECT COUNT(\*)  FROM vwSchedule v  WHERE v.개설과정번호 = s.개설과정번호 AND v.개설과목번호 = s.개설과목번호) AS 등록인원  FROM vwSchedule s  WHERE 교사번호 = p\_professor\_num AND  sysdate < 과목시작일  ORDER BY 개설과정번호, 개설과목번호)  LOOP  DBMS\_OUTPUT.PUT\_LINE('과정번호: ' || schedule.개설과정번호);  DBMS\_OUTPUT.PUT\_LINE('과정명: ' || schedule.과정명);  DBMS\_OUTPUT.PUT\_LINE('과정시작일: ' || schedule.과정시작일);  DBMS\_OUTPUT.PUT\_LINE('과정종료일: ' || schedule.과정종료일);  DBMS\_OUTPUT.PUT\_LINE('과목번호: ' || schedule.개설과목번호);  DBMS\_OUTPUT.PUT\_LINE('과목명: ' || schedule.과목명);  DBMS\_OUTPUT.PUT\_LINE('과목시작일: ' || schedule.과목시작일);  DBMS\_OUTPUT.PUT\_LINE('과목종료일: ' || schedule.과목종료일);  DBMS\_OUTPUT.PUT\_LINE('강의실명: ' || schedule.강의실명);  DBMS\_OUTPUT.PUT\_LINE('교재명: ' || schedule.교재명);  DBMS\_OUTPUT.PUT\_LINE('등록인원: ' || schedule.등록인원);  DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------------------------');  END LOOP;  ELSIF p\_subject\_progress\_state = '강의 중' THEN  FOR schedule IN (SELECT DISTINCT s.개설과정번호, s.과정명, s.과정시작일, s.과정종료일,  s.개설과목번호, s.과목명, s.과목시작일, s.과목종료일, s.강의실명, s.교재명,  (SELECT COUNT(\*)  FROM vwSchedule v  WHERE v.개설과정번호 = s.개설과정번호 AND v.개설과목번호 = s.개설과목번호) AS 등록인원  FROM vwSchedule s  WHERE 교사번호 = p\_professor\_num AND  (sysdate > 과목시작일 AND sysdate < 과목종료일)  ORDER BY 개설과정번호, 개설과목번호)  LOOP  DBMS\_OUTPUT.PUT\_LINE('과정번호: ' || schedule.개설과정번호);  DBMS\_OUTPUT.PUT\_LINE('과정명: ' || schedule.과정명);  DBMS\_OUTPUT.PUT\_LINE('과정시작일: ' || schedule.과정시작일);  DBMS\_OUTPUT.PUT\_LINE('과정종료일: ' || schedule.과정종료일);  DBMS\_OUTPUT.PUT\_LINE('과목번호: ' || schedule.개설과목번호);  DBMS\_OUTPUT.PUT\_LINE('과목명: ' || schedule.과목명);  DBMS\_OUTPUT.PUT\_LINE('과목시작일: ' || schedule.과목시작일);  DBMS\_OUTPUT.PUT\_LINE('과목종료일: ' || schedule.과목종료일);  DBMS\_OUTPUT.PUT\_LINE('강의실명: ' || schedule.강의실명);  DBMS\_OUTPUT.PUT\_LINE('교재명: ' || schedule.교재명);  DBMS\_OUTPUT.PUT\_LINE('등록인원: ' || schedule.등록인원);  DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------------------------');  END LOOP;  ELSIF p\_subject\_progress\_state = '강의 종료' THEN  FOR schedule IN (SELECT DISTINCT s.개설과정번호, s.과정명, s.과정시작일, s.과정종료일,  s.개설과목번호, s.과목명, s.과목시작일, s.과목종료일, s.강의실명, s.교재명,  (SELECT COUNT(\*)  FROM vwSchedule v  WHERE v.개설과정번호 = s.개설과정번호 AND v.개설과목번호 = s.개설과목번호) AS 등록인원  FROM vwSchedule s  WHERE 교사번호 = p\_professor\_num AND  sysdate > 과목종료일  ORDER BY 개설과정번호, 개설과목번호)  LOOP  DBMS\_OUTPUT.PUT\_LINE('과정번호: ' || schedule.개설과정번호);  DBMS\_OUTPUT.PUT\_LINE('과정명: ' || schedule.과정명);  DBMS\_OUTPUT.PUT\_LINE('과정시작일: ' || schedule.과정시작일);  DBMS\_OUTPUT.PUT\_LINE('과정종료일: ' || schedule.과정종료일);  DBMS\_OUTPUT.PUT\_LINE('과목번호: ' || schedule.개설과목번호);  DBMS\_OUTPUT.PUT\_LINE('과목명: ' || schedule.과목명);  DBMS\_OUTPUT.PUT\_LINE('과목시작일: ' || schedule.과목시작일);  DBMS\_OUTPUT.PUT\_LINE('과목종료일: ' || schedule.과목종료일);  DBMS\_OUTPUT.PUT\_LINE('강의실명: ' || schedule.강의실명);  DBMS\_OUTPUT.PUT\_LINE('교재명: ' || schedule.교재명);  DBMS\_OUTPUT.PUT\_LINE('등록인원: ' || schedule.등록인원);  DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------------------------');  END LOOP;  END IF;  DBMS\_OUTPUT.PUT\_LINE('조회가 완료되었습니다.');  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('조회에 실패하였습니다.');  END procSchedule;  BEGIN  procSchedule(1, '강의 중');  END;  – 2. 교육생 정보 출력 > 교육생 번호, 교육생 이름, 전화번호, 등록일, 과목명, 수료 또는 탈락  CREATE OR REPLACE PROCEDURE procCurStudent(  p\_curriculum\_num NUMBER  )  IS  BEGIN  DBMS\_OUTPUT.PUT\_LINE('===============================[교육생 정보]==============================  =');  FOR cirStudent IN (SELECT 교육생번호, 이름, 전화번호, 등록일, 수료여부  FROM vwStudent  WHERE 개설과정번호 = p\_curriculum\_num  ORDER BY 교육생번호)  LOOP  DBMS\_OUTPUT.PUT\_LINE('이름: ' || cirStudent.이름);  DBMS\_OUTPUT.PUT\_LINE('전화번호: ' || cirStudent.전화번호);  DBMS\_OUTPUT.PUT\_LINE('등록일: ' || cirStudent.등록일);  DBMS\_OUTPUT.PUT\_LINE('수료여부: ' || cirStudent.수료여부);  DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------------------------');  END LOOP;  DBMS\_OUTPUT.PUT\_LINE('조회가 완료되었습니다.');  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('조회에 실패하였습니다.');  END procCurStudent;  BEGIN  procCurStudent(1);  END; | | | |

| 요구사항번호 | **C-02(교사)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 배점 입출력 | 작성자 | 차수민 |
| DML | | | |
| – 1. 과목 목록 출력 > 과정번호, 과정명, 과정기간(시작 년월일 / 끝 년월일),  과목번호, 과목명, 과목기간(시작 년월일 / 끝 년월일), 강의실명, 교재명,  출결 배점, 필기 배점, 실기 배점, 성적등록여부  CREATE OR REPLACE PROCEDURE procSubject(  p\_professor\_num NUMBER  )  IS  BEGIN  DBMS\_OUTPUT.PUT\_LINE('================================[과목 목록]==============================  ==');  FOR subject IN (SELECT DISTINCT 개설과정번호, 과정명, 과정시작일, 과정종료일,  개설과목번호, 과목명, 과목시작일, 과목종료일, 강의실명, 교재명,  t.test\_num AS 시험번호, t.att\_ratio AS 출결배점, t.write\_ratio AS 필기배점, t.practical\_ratio AS 실기배점,  t.score\_registration\_state AS 성적등록여부  FROM vwSchedule s  INNER JOIN tblTest t  ON s.개설과목번호 = t.open\_subject\_num  WHERE s.교사번호 = p\_professor\_num AND  sysdate > s.과목종료일  ORDER BY s.개설과정번호, s.개설과목번호)  LOOP  DBMS\_OUTPUT.PUT\_LINE('과정번호: ' || subject.개설과정번호);  DBMS\_OUTPUT.PUT\_LINE('과정명: ' || subject.과정명);  DBMS\_OUTPUT.PUT\_LINE('과정시작일: ' || subject.과정시작일);  DBMS\_OUTPUT.PUT\_LINE('과정종료일: ' || subject.과정종료일);  DBMS\_OUTPUT.PUT\_LINE('과목번호: ' || subject.개설과목번호);  DBMS\_OUTPUT.PUT\_LINE('과목명: ' || subject.과목명);  DBMS\_OUTPUT.PUT\_LINE('과목시작일: ' || subject.과목시작일);  DBMS\_OUTPUT.PUT\_LINE('과목종료일: ' || subject.과목종료일);  DBMS\_OUTPUT.PUT\_LINE('강의실명: ' || subject.강의실명);  DBMS\_OUTPUT.PUT\_LINE('교재명: ' || subject.교재명);  DBMS\_OUTPUT.PUT\_LINE('시험번호: ' || subject.시험번호);  IF subject.출결배점 IS NULL THEN  DBMS\_OUTPUT.PUT\_LINE('출결배점: ' || 'NULL');  ELSE  DBMS\_OUTPUT.PUT\_LINE('출결배점: ' || subject.출결배점);  END IF;  IF subject.필기배점 IS NULL THEN  DBMS\_OUTPUT.PUT\_LINE('필기배점: ' || 'NULL');  ELSE  DBMS\_OUTPUT.PUT\_LINE('필기배점: ' || subject.필기배점);  END IF;  IF subject.실기배점 IS NULL THEN  DBMS\_OUTPUT.PUT\_LINE('실기배점: ' || 'NULL');  ELSE  DBMS\_OUTPUT.PUT\_LINE('실기배점: ' || subject.실기배점);  END IF;  IF subject.성적등록여부 = '여' THEN  DBMS\_OUTPUT.PUT\_LINE('성적등록여부: ' || '입력');  ELSE  DBMS\_OUTPUT.PUT\_LINE('성적등록여부: ' || '미입력');  END IF;  DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------------------------');  END LOOP;  DBMS\_OUTPUT.PUT\_LINE('조회가 완료되었습니다.');  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('조회에 실패하였습니다.');  END procSubject;  BEGIN  procSubject(1);  END;  – 2. 배점 입력 > 출결 배점, 필기 배점, 실기 배점, 시험 날짜, 시험 문제  CREATE OR REPLACE PROCEDURE procPoints(  p\_test\_num NUMBER,  p\_test\_file varchar2,  p\_test\_date varchar2,  p\_attendance\_ratio NUMBER,  p\_written\_ratio NUMBER,  p\_practical\_ratio NUMBER,  p\_score\_registration\_state varchar2  )  IS  BEGIN  DBMS\_OUTPUT.PUT\_LINE('================================[배점 입력]==============================  ==');  UPDATE tblTest  SET test\_file = p\_test\_file, test\_date = to\_date(p\_test\_date, 'yyyy-mm-dd'), att\_ratio = p\_attendance\_ratio,  write\_ratio = p\_written\_ratio, practical\_ratio = p\_practical\_ratio, score\_registration\_state = p\_score\_registration\_state  WHERE test\_num = p\_test\_num;  DBMS\_OUTPUT.PUT\_LINE('입력되었습니다.');  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('입력에 실패하였습니다.');  END procPoints;  BEGIN  procStudent(2, 'SQL\_Test.docx', '2023-09-16', 20, 30, 50, '여');  END; | | | |

| 요구사항번호 | **C-03(교사)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 성적 입출력 | 작성자 | 차수민 |
| DML | | | |
| – 1. 과목 목록 출력 > 과정번호, 과정명, 과정기간(시작 년월일 / 끝 년월일),  과목번호, 과목명, 과목기간(시작 년월일 / 끝 년월일), 강의실명, 교재명,  출결 배점, 필기 배점, 실기 배점, 성적등록여부  BEGIN  procSubject(1);  END;  – 2. 교육생 정보 출력 > 교육생 번호, 교육생 이름, 전화번호, 등록일, 수료여부, 시험번호, 수강번호, 출결점수, 필기점수, 실기점수  CREATE OR REPLACE PROCEDURE procSubStudent(  p\_subject\_num NUMBER  )  IS  BEGIN  DBMS\_OUTPUT.PUT\_LINE('===============================[교육생 정보]=============================  ==');  FOR subStudent IN (SELECT 교육생번호, 이름, 전화번호, 등록일, 수료여부, 시험번호, 수강번호, 출결점수, 필기점수, 실기점수  FROM vwScore  WHERE 개설과목번호 = p\_subject\_num  ORDER BY 교육생번호)  LOOP  DBMS\_OUTPUT.PUT\_LINE('교육생번호: ' || subStudent.교육생번호);  DBMS\_OUTPUT.PUT\_LINE('이름: ' || subStudent.이름);  DBMS\_OUTPUT.PUT\_LINE('전화번호: ' || subStudent.전화번호);  DBMS\_OUTPUT.PUT\_LINE('등록일: ' || subStudent.등록일);  DBMS\_OUTPUT.PUT\_LINE('수료여부: ' || subStudent.수료여부);  DBMS\_OUTPUT.PUT\_LINE('시험번호: ' || subStudent.시험번호);  DBMS\_OUTPUT.PUT\_LINE('수강번호: ' || subStudent.수강번호);  IF subStudent.출결점수 IS NULL THEN  DBMS\_OUTPUT.PUT\_LINE('출결점수: ' || '미입력');  ELSE  DBMS\_OUTPUT.PUT\_LINE('출결점수: ' || subStudent.출결점수);  END IF;  IF subStudent.필기점수 IS NULL THEN  DBMS\_OUTPUT.PUT\_LINE('필기점수: ' || '미입력');  ELSE  DBMS\_OUTPUT.PUT\_LINE('필기점수: ' || subStudent.필기점수);  END IF;  IF subStudent.실기점수 IS NULL THEN  DBMS\_OUTPUT.PUT\_LINE('실기점수: ' || '미입력');  ELSE  DBMS\_OUTPUT.PUT\_LINE('실기점수: ' || subStudent.실기점수);  END IF;  DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------------------------');  END LOOP;  DBMS\_OUTPUT.PUT\_LINE('조회가 완료되었습니다.');  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('조회에 실패하였습니다.');  END procSubStudent;  BEGIN  procSubStudent(2);  END;  – 3. 성적 입력 > 시험번호, 수강번호, 출결점수, 필기점수, 실기점수  CREATE OR REPLACE PROCEDURE procScore(  p\_test\_num NUMBER,  p\_course\_num NUMBER,  p\_attendance\_score NUMBER,  p\_written\_score NUMBER,  p\_pratical\_score NUMBER  )  IS  BEGIN  DBMS\_OUTPUT.PUT\_LINE('================================[성적 입력]==============================  ==');  INSERT INTO tblScore VALUES (score\_seq.nextval, p\_test\_num, p\_course\_num, p\_attendance\_score, p\_written\_score, p\_pratical\_score);  DBMS\_OUTPUT.PUT\_LINE('입력되었습니다.');  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('입력에 실패하였습니다.');  END procScore;  BEGIN  procScore(2, 2, 70, 85, 60);  END; | | | |

| 요구사항번호 | **C-04(교사)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 출결 조회 | 작성자 | 차수민 |
| DML | | | |
| – 1. 강의 스케줄 정보 출력 > 과정번호, 과정명, 과정기간(시작 년월일 / 끝 년월일),  과목번호, 과목명, 과목기간(시작 년월일 / 끝 년월일), 강의실명, 교재명, 교육생 등록 인원  BEGIN  procSchedule(1, '강의 중');  END;  – 2. 교육생 출결 출력 > 교육생 번호, 교육생 이름, 출결일, 근태상황  CREATE OR REPLACE PROCEDURE procStudAttendance(  p\_professor\_num NUMBER,  p\_curriculum\_num NUMBER,  p\_start varchar2,  p\_end varchar2  )  IS  BEGIN  DBMS\_OUTPUT.PUT\_LINE('===============================[교육생 출결]=============================  ==');  FOR studAttendance IN (SELECT DISTINCT s.student\_num AS 교육생번호, s.name AS 이름,  a.att\_date AS 출결일, a.att\_state AS 근태상황  FROM tblStudent s  INNER JOIN tblRegister r  ON s.student\_num = r.student\_num  INNER JOIN tblAttendance a  ON r.register\_num = a.register\_num  INNER JOIN tblOpenCurriculum ocur  ON ocur.open\_curriculum\_num = r.open\_curriculum\_num  INNER JOIN tblOpenSubject osub  ON ocur.open\_curriculum\_num = osub.open\_curriculum\_num  INNER JOIN tblAbleSubject asub  ON osub.able\_subject\_num = asub.able\_subject\_num  INNER JOIN tblProfessor p  ON asub.professor\_num = p.professor\_num  WHERE p.professor\_num = p\_professor\_num AND  ocur.open\_curriculum\_num = p\_curriculum\_num AND  (a.att\_date >= to\_date(p\_start, 'yyyy-mm-dd') AND a.att\_date <= to\_date(p\_end, 'yyyy-mm-dd'))  ORDER BY 출결일, 교육생번호)  LOOP  DBMS\_OUTPUT.PUT\_LINE('교육생번호: ' || studAttendance.교육생번호);  DBMS\_OUTPUT.PUT\_LINE('이름: ' || studAttendance.이름);  DBMS\_OUTPUT.PUT\_LINE('출결일: ' || studAttendance.출결일);  DBMS\_OUTPUT.PUT\_LINE('근태상황: ' || studAttendance.근태상황);  DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------------------------');  END LOOP;  DBMS\_OUTPUT.PUT\_LINE('조회가 완료되었습니다.');  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('조회에 실패하였습니다.');  END procStudAttendance;  BEGIN  procStudAttendance(1, 4, '2023-09-01', '2023-09-01');  END;  – 3. 교사 출결 출력 > 교사 번호, 교사 이름, 출결일, 근태상황  CREATE OR REPLACE PROCEDURE procProfAttendance(  p\_proffesor\_num NUMBER,  p\_start varchar2,  p\_end varchar2  )  IS  BEGIN  DBMS\_OUTPUT.PUT\_LINE('=================================[교사 출결]=============================  ==');  FOR profAttendance IN (SELECT p.professor\_num AS 교사번호, p.name AS 이름, pa.att\_date AS 출결일, pa.att\_state AS 근태상황  FROM tblProfessorAttendance pa  INNER JOIN tblProfessor p  ON pa.professor\_num = p.professor\_num  WHERE p.professor\_num = p\_proffesor\_num AND  (pa.att\_date >= to\_date(p\_start, 'yyyy-mm-dd') AND pa.att\_date <= to\_date(p\_end, 'yyyy-mm-dd'))  ORDER BY 출결일)  LOOP  DBMS\_OUTPUT.PUT\_LINE('교사번호: ' || profAttendance.교사번호);  DBMS\_OUTPUT.PUT\_LINE('이름: ' || profAttendance.이름);  DBMS\_OUTPUT.PUT\_LINE('출결일: ' || profAttendance.출결일);  DBMS\_OUTPUT.PUT\_LINE('근태상황: ' || profAttendance.근태상황);  DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------------------------');  END LOOP;  DBMS\_OUTPUT.PUT\_LINE('조회가 완료되었습니다.');  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('조회에 실패하였습니다.');  END procProfAttendance;  BEGIN  procProfAttendance(1, '2023-09-01', '2023-09-13');  END; | | | |

| 요구사항번호 | **C-05(교사)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 강의가능과목 조회 | 작성자 | 차수민 |
| DML | | | |
| – 1. 강의가능과목 출력 > 과목번호, 과목명, 교사 번호, 교사 이름  CREATE OR REPLACE PROCEDURE procAbleSubject(  p\_subject\_num NUMBER  )  IS  BEGIN  DBMS\_OUTPUT.PUT\_LINE('============================[강의가능과목 목록]============================');  FOR ableSubject IN (SELECT sub.subject\_num AS 과목번호, sub.subject AS 과목명, p.professor\_num AS 교사번호, p.name AS 교사명  FROM tblAbleSubject asub  INNER JOIN tblSubject sub  ON asub.subject\_num = sub.subject\_num  INNER JOIN tblProfessor p  ON asub.professor\_num = p.professor\_num  WHERE sub.subject\_num = p\_subject\_num  ORDER BY 교사번호)  LOOP  DBMS\_OUTPUT.PUT\_LINE('과목번호: ' || ableSubject.과목번호);  DBMS\_OUTPUT.PUT\_LINE('과목명: ' || ableSubject.과목명);  DBMS\_OUTPUT.PUT\_LINE('교사번호: ' || ableSubject.교사번호);  DBMS\_OUTPUT.PUT\_LINE('교사명: ' || ableSubject.교사명);  DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------------------------');  END LOOP;  DBMS\_OUTPUT.PUT\_LINE('조회가 완료되었습니다.');  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('조회에 실패하였습니다.');  END procAbleSubject;  BEGIN  procAbleSubject(1);  END; | | | |

| 요구사항번호 | **C-06(교사)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 질문 및 답변 입출력 | 작성자 | 차수민 |
| DML | | | |
| – 1. 질문 및 답변 출력 > 질문번호, 교육생 번호, 교육생 이름, 질문, 질문일, 답변, 답변일  CREATE OR REPLACE PROCEDURE procQna(  p\_professor\_num NUMBER  )  IS  BEGIN  DBMS\_OUTPUT.PUT\_LINE('===============================[질문 및 답변]============================  ==');  FOR qna IN (SELECT 질문번호, 교육생번호, 교육생, 질문, 질문일, 답변, 답변일  FROM vwQna  WHERE 교사번호 = p\_professor\_num)  LOOP  DBMS\_OUTPUT.PUT\_LINE('질문번호: ' || qna.질문번호);  DBMS\_OUTPUT.PUT\_LINE('교육생번호: ' || qna.교육생번호);  DBMS\_OUTPUT.PUT\_LINE('교육생: ' || qna.교육생);  DBMS\_OUTPUT.PUT\_LINE('질문: ' || qna.질문);  IF qna.답변 IS NULL THEN  DBMS\_OUTPUT.PUT\_LINE('답변: ' || '무응답');  DBMS\_OUTPUT.PUT\_LINE('답변일: ' || '해당없음');  ELSE  DBMS\_OUTPUT.PUT\_LINE('답변: ' || qna.답변);  DBMS\_OUTPUT.PUT\_LINE('답변일: ' || qna.답변일);  END IF;  DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------------------------');  END LOOP;  DBMS\_OUTPUT.PUT\_LINE('조회가 완료되었습니다.');  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('조회에 실패하였습니다.');  END procQna;  BEGIN  procQna(2);  END;  – 2. 답변 입력 > 답변 번호, 질문 번호, 답변, 답변일  CREATE OR REPLACE PROCEDURE procAnswer(  p\_question\_num NUMBER,  p\_answer varchar2  )  IS  BEGIN  DBMS\_OUTPUT.PUT\_LINE('================================[답변 입력]==============================  ==');  INSERT INTO tblAnswer VALUES (answer\_seq.nextval, p\_question\_num, p\_answer, sysdate);  DBMS\_OUTPUT.PUT\_LINE('입력되었습니다.');  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('입력에 실패하였습니다.');  END procAnswer;  BEGIN  procAnswer(121, '답변 완료');  END;  – 3. 답변 수정 > 답변 번호, 질문 번호, 답변, 답변일  CREATE OR REPLACE PROCEDURE procReAnswer(  p\_question\_num NUMBER,  p\_answer varchar2  )  IS  BEGIN  DBMS\_OUTPUT.PUT\_LINE('================================[답변 수정]==============================  ==');  UPDATE tblAnswer SET answer = p\_answer WHERE question\_num = p\_question\_num;  DBMS\_OUTPUT.PUT\_LINE('수정되었습니다.');  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('수정에 실패하였습니다.');  END procReAnswer;  BEGIN  procReAnswer(121, '답변 수정 완료');  END; | | | |

| 요구사항번호 | **D-01(교육생)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 개인 정보 조회 | 작성자 | 차수민 |
| DML | | | |
| CREATE OR REPLACE PROCEDURE procStudentInfo(  p\_student\_num NUMBER  )  IS  BEGIN  DBMS\_OUTPUT.PUT\_LINE('===============================[교육생 정보]=============================  ==');  FOR student IN (SELECT ocur.open\_curriculum\_num AS 과정번호, cur.curriculum AS 과정명, ocur.start\_date AS 과정시작일,  ocur.end\_date AS 과정종료일, c.classroom AS 강의실명, s.student\_num AS 교육생번호, s.name AS 이름,  s.tel AS 연락처  FROM tblStudent s  INNER JOIN tblRegister r  ON s.student\_num = r.student\_num  INNER JOIN tblOpenCurriculum ocur  ON r.open\_curriculum\_num = ocur.open\_curriculum\_num  INNER JOIN tblCurriculum cur  ON ocur.open\_curriculum\_num = cur.curriculum\_num  INNER JOIN tblClassroom c  ON ocur.classroom\_num = c.classroom\_num  WHERE s.student\_num = p\_student\_num)  LOOP  DBMS\_OUTPUT.PUT\_LINE('과정번호: ' || student.과정번호);  DBMS\_OUTPUT.PUT\_LINE('과정명: ' || student.과정명);  DBMS\_OUTPUT.PUT\_LINE('과정시작일: ' || student.과정시작일);  DBMS\_OUTPUT.PUT\_LINE('과정종료일: ' || student.과정종료일);  DBMS\_OUTPUT.PUT\_LINE('교육생번호: ' || student.교육생번호);  DBMS\_OUTPUT.PUT\_LINE('이름: ' || student.이름);  DBMS\_OUTPUT.PUT\_LINE('연락처: ' || student.연락처);  DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------------------------');  END LOOP;  DBMS\_OUTPUT.PUT\_LINE('조회가 완료되었습니다.');  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('조회에 실패하였습니다.');  END procStudentInfo;  BEGIN  procStudentInfo(83);  END; | | | |

| 요구사항번호 | **D-02(교육생)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 성적 조회 | 작성자 | 차수민 |
| DML | | | |
| – 1. 성적 출력 > 과목번호, 과목명, 과목기간(시작 년월일 / 끝 년월일), 교재명, 교사명,  출결 배점, 필기 배점, 실기 배점, 출결 점수, 필기 점수, 실기 점수, 시험일, 시험문제  CREATE OR REPLACE PROCEDURE procStudScore(  p\_student\_num NUMBER  )  IS  BEGIN  DBMS\_OUTPUT.PUT\_LINE('===============================[교육생 성적]=============================  ==');  FOR score IN (SELECT 과목번호, 과목명, 과목시작일, 과목종료일, 교재명, 교사명, 출결배점, 필기배점, 실기배점,  출결점수, 필기점수, 실기점수, 시험일, 시험문제  FROM vwStudScore  WHERE 교육생번호 = p\_student\_num  ORDER BY 과목번호)  LOOP  DBMS\_OUTPUT.PUT\_LINE('과목번호: ' || score.과목번호);  DBMS\_OUTPUT.PUT\_LINE('과목명: ' || score.과목명);  DBMS\_OUTPUT.PUT\_LINE('과목시작일: ' || score.과목시작일);  DBMS\_OUTPUT.PUT\_LINE('과목종료일: ' || score.과목종료일);  DBMS\_OUTPUT.PUT\_LINE('교재명: ' || score.교재명);  DBMS\_OUTPUT.PUT\_LINE('교사명: ' || score.교사명);  IF score.출결배점 IS NULL THEN  DBMS\_OUTPUT.PUT\_LINE('출결배점: ' || '미입력');  ELSE  DBMS\_OUTPUT.PUT\_LINE('출결배점: ' || score.출결배점);  END IF;  IF score.필기배점 IS NULL THEN  DBMS\_OUTPUT.PUT\_LINE('필기배점: ' || '미입력');  ELSE  DBMS\_OUTPUT.PUT\_LINE('필기배점: ' || score.필기배점);  END IF;  IF score.실기배점 IS NULL THEN  DBMS\_OUTPUT.PUT\_LINE('실기배점: ' || '미입력');  ELSE  DBMS\_OUTPUT.PUT\_LINE('실기배점: ' || score.실기배점);  END IF;  IF score.출결점수 IS NULL THEN  DBMS\_OUTPUT.PUT\_LINE('출결점수: ' || 'NULL');  ELSE  DBMS\_OUTPUT.PUT\_LINE('출결점수: ' || score.출결점수);  END IF;  IF score.필기점수 IS NULL THEN  DBMS\_OUTPUT.PUT\_LINE('필기점수: ' || 'NULL');  ELSE  DBMS\_OUTPUT.PUT\_LINE('필기점수: ' || score.필기점수);  END IF;  IF score.실기점수 IS NULL THEN  DBMS\_OUTPUT.PUT\_LINE('실기점수: ' || 'NULL');  ELSE  DBMS\_OUTPUT.PUT\_LINE('실기점수: ' || score.실기점수);  END IF;  DBMS\_OUTPUT.PUT\_LINE('시험일: ' || score.시험일);  DBMS\_OUTPUT.PUT\_LINE('시험문제: ' || score.시험문제);  DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------------------------');  END LOOP;  DBMS\_OUTPUT.PUT\_LINE('조회가 완료되었습니다.');  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('조회에 실패하였습니다.');  END procStudScore;  BEGIN  procStudScore(83);  END; | | | |

| 요구사항번호 | **D-03(교육생)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 출결 관리 및 출결 조회 | 작성자 | 차수민 |
| DML | | | |
| – 1. 출결 출력 > 교육생 번호, 이름, 출결일, 근태상황  CREATE OR REPLACE PROCEDURE procAttendance(  p\_student\_num NUMBER,  p\_start varchar2,  p\_end varchar2  )  IS  BEGIN  DBMS\_OUTPUT.PUT\_LINE('===============================[교육생 출결]=============================  ==');  FOR attendance IN (SELECT DISTINCT s.student\_num AS 교육생번호, s.name AS 이름, a.att\_date AS 출결일, a.att\_state AS 근태상황  FROM tblStudent s  INNER JOIN tblRegister r  ON s.student\_num = r.student\_num  INNER JOIN tblAttendance a  ON r.register\_num = a.register\_num  WHERE s.student\_num = p\_student\_num AND  (a.att\_date >= to\_date(p\_start, 'yyyy-mm-dd') AND a.att\_date <= to\_date(p\_end, 'yyyy-mm-dd'))  ORDER BY 출결일)  LOOP  DBMS\_OUTPUT.PUT\_LINE('교육생번호: ' || attendance.교육생번호);  DBMS\_OUTPUT.PUT\_LINE('이름: ' || attendance.이름);  DBMS\_OUTPUT.PUT\_LINE('출결일: ' || attendance.출결일);  DBMS\_OUTPUT.PUT\_LINE('근태상황: ' || attendance.근태상황);  DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------------------------');  END LOOP;  DBMS\_OUTPUT.PUT\_LINE('조회가 완료되었습니다.');  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('조회에 실패하였습니다.');  END procAttendance;  BEGIN  procAttendance(83, '2023-09-01', '2023-09-15');  END; | | | |

| 요구사항번호 | **D-04(교육생)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 추천도서 조회 | 작성자 | 김혜민 |
| DML | | | |
| CREATE OR REPLACE PROCEDURE procBestBookInfo(  p\_professor\_num NUMBER  )  IS  BEGIN  FOR book\_record IN (  SELECT  tbb.bestboon\_num AS "추천도서번호",  tbb.subject\_num AS "과목번호",  tbb.book AS "도서명",  tbb.public AS "출판사",  tpf.name AS "추천교사명"  FROM tblBestBook tbb  INNER JOIN tblSubject ts ON tbb.subject\_num = ts.subject\_num  INNER JOIN tblAbleSubject tas ON tas.subject\_num = ts.subject\_num  INNER JOIN tblProfessor tpf ON tpf.professor\_num = tas.professor\_num  WHERE tas.professor\_num = p\_professor\_num  )  LOOP  DBMS\_OUTPUT.PUT\_LINE('추천도서번호: ' || book\_record.추천도서번호 || ', ' ||  '과목번호: ' || book\_record.과목번호 || ', ' ||  '도서명: ' || book\_record.도서명 || ', ' ||  '출판사: ' || book\_record.출판사 || ', ' ||  '추천교사명: ' || book\_record.추천교사명);  END LOOP;  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('오류 발생: ' || SQLERRM);  END procBestBookInfo; | | | |

| 요구사항번호 | **D-05(교육생)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 상담일지 조회 | 작성자 | 김혜민 |
| DML | | | |
| --입력  CREATE OR REPLACE PROCEDURE procAddCounsel(  p\_register\_num NUMBER,  p\_counsel\_text VARCHAR2  )  IS  BEGIN  INSERT INTO tblCounsel(counsel\_num, register\_num, counsel, counsel\_date)  VALUES (counsel\_seq.nextval, p\_register\_num, p\_counsel\_text, SYSDATE);  DBMS\_OUTPUT.PUT\_LINE('상담 정보가 성공적으로 추가되었습니다.');  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('상담 정보 추가 중 오류 발생: ' || SQLERRM);  END procAddCounsel;    --출력  CREATE OR REPLACE PROCEDURE procCounselInfo(  p\_student\_num NUMBER  )  IS  BEGIN  FOR counsel\_record IN (  SELECT  tbc.counsel AS 상담번호,  std.name AS 교육생이름,  tbc.register\_num AS 수강과정번호,  tbc.counsel AS 상담내용,  tbc.counsel\_date AS 상담일,  cul.curriculum AS 과정명,  toc.start\_date AS 과정시작일,  toc.end\_date AS 과정종료일,  cla.classroom\_num AS 강의실번호  FROM tblStudent std  INNER JOIN tblRegister tbr ON std.student\_num = tbr.student\_num  INNER JOIN tblOpenCurriculum toc ON toc.open\_curriculum\_num = tbr.open\_curriculum\_num  INNER JOIN tblCurriculum cul ON cul.curriculum\_num = toc.curriculum\_num  INNER JOIN tblCounsel tbc ON tbr.register\_num = tbc.register\_num  INNER JOIN tblClassroom cla ON cla.classroom\_num = toc.classroom\_num  WHERE std.student\_num = p\_student\_num  )  LOOP  DBMS\_OUTPUT.PUT\_LINE('상담번호: ' || counsel\_record.상담번호 || ', ' ||  '교육생이름: ' || counsel\_record.교육생이름 || ', ' ||  '수강과정번호: ' || counsel\_record.수강과정번호 || ', ' ||  '상담내용: ' || counsel\_record.상담내용 || ', ' ||  '상담일: ' || TO\_CHAR(counsel\_record.상담일, 'YYYY-MM-DD') || ', ' ||  '과정명: ' || counsel\_record.과정명 || ', ' ||  '과정시작일: ' || TO\_CHAR(counsel\_record.과정시작일, 'YYYY-MM-DD') || ', ' ||  '과정종료일: ' || TO\_CHAR(counsel\_record.과정종료일, 'YYYY-MM-DD') || ', ' ||  '강의실번호: ' || counsel\_record.강의실번호);  END LOOP;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('해당 교육생에 대한 상담 정보가 없습니다.');  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('오류 발생: ' || SQLERRM);  END procCounselInfo; | | | |

| 요구사항번호 | **D-06(교육생)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 질문 조회 | 작성자 | 이도훈 |
| DML | | | |
| --입력  CREATE OR REPLACE PROCEDURE procAddQuestion(  p\_course\_num NUMBER,  p\_question\_text VARCHAR2  )  IS  BEGIN  BEGIN  INSERT INTO tblquestion(question\_num, course\_num, question, question\_date)  VALUES (question\_seq.nextval, p\_course\_num, p\_question\_text, SYSDATE);  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('오류 발생');  END;  END procAddQuestion;  --출력  --질문 기본 정보  CREATE OR REPLACE PROCEDURE procBasicInfoQuestion  IS  BEGIN  FOR question\_record IN (  SELECT  question\_num AS 질문번호,  course\_num AS 수강과목번호,  question AS 질문내용,  question\_date AS 질문일  FROM tblquestion  )  LOOP  BEGIN  DBMS\_OUTPUT.PUT\_LINE('질문번호: ' || question\_record.질문번호 || ', ' ||  '수강과목번호: ' || question\_record.수강과목번호 || ', ' ||  '질문내용: ' || question\_record.질문내용 || ', ' ||  '질문일: ' || TO\_CHAR(question\_record.질문일, 'YYYY-MM-DD HH24:MI:SS'));  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('오류 발생: ' || SQLERRM);  END;  END LOOP;  END procBasicInfoQuestion;  --질문 정보  CREATE OR REPLACE PROCEDURE procInfoQuestion  IS  BEGIN  FOR question\_record IN (  SELECT  s.name AS 교육생명,  cla.classroom AS 강의실,  cu.curriculum AS 과정명,  q.question AS 질문내용,  q.question\_date AS 질문일  FROM tblquestion q  INNER JOIN tblcourse c ON c.course\_num = q.course\_num  INNER JOIN tblregister r ON r.register\_num = c.course\_num  INNER JOIN tblstudent s ON s.student\_num = r.student\_num  INNER JOIN tblopencurriculum o ON o.open\_curriculum\_num = r.open\_curriculum\_num  INNER JOIN tblcurriculum cu ON cu.curriculum\_num = o.curriculum\_num  INNER JOIN tblclassroom cla ON cla.classroom\_num = o.classroom\_num  )  LOOP  BEGIN  DBMS\_OUTPUT.PUT\_LINE('교육생명: ' || question\_record.교육생명 || ', ' ||  '강의실: ' || question\_record.강의실 || ', ' ||  '과정명: ' || question\_record.과정명 || ', ' ||  '질문내용: ' || question\_record.질문내용 || ', ' ||  '질문일: ' || TO\_CHAR(question\_record.질문일, 'YYYY-MM-DD HH24:MI:SS'));  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('오류 발생: ' || SQLERRM);  END;  END LOOP;  END procInfoQuestion;  --삭제  CREATE OR REPLACE PROCEDURE procDeleteQuestion(  p\_question\_num NUMBER  )  IS  BEGIN  BEGIN  UPDATE tblquestion  SET question = '-'  WHERE question\_num = p\_question\_num;  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('오류 발생: ' || SQLERRM);  END;  END procDeleteQuestion; | | | |

| 요구사항번호 | **D-07(교육생)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 개설과정 조회 | 작성자 | 김혜민 |
| DML | | | |
| --1. 개설과정과 관련된 정보를 불러올 시, 과정번호,과정명, 과정기간, 강의실 번호 등의 정보를 불러온다.  create or replace procedure procSelReview(  pcurriculum\_num number,  presult out number  )  is  v\_curriculum\_num tblOpenCurriculum.curriculum\_num%type;  v\_curriculum tblCurriculum.curriculum%type;  v\_period tblOpenCurriculum.period%type;  v\_start\_date tblOpenCurriculum.start\_date%type;  v\_end\_date tblOpenCurriculum.end\_date%type;  v\_classroom tblClassroom.classroom%type;    cursor vcursor is  select  toc.curriculum\_num AS 과정번호,  cu.curriculum AS 과정명,  toc.period AS 과정기간,  toc.start\_date AS 과정시작일,  toc.end\_date AS 과정종료일,  tc.classroom AS 강의실명  from tblOpenCurriculum toc  inner join tblCurriculum cu on toc.curriculum\_num = cu.curriculum\_num  inner join tblClassroom tc on toc.classroom\_num = tc.classroom\_num;  begin  open vcursor;  loop  FETCH vcursor INTO v\_curriculum\_num, v\_curriculum, v\_period, v\_start\_date, v\_end\_date, v\_classroom;  exit when vcursor%notfound;    DBMS\_OUTPUT.PUT\_LINE('과정번호 : ' || v\_curriculum\_num || ', 과정명 : ' || v\_curriculum || ', 과정기간 : ' || v\_period || ', 과정시작일 : ' || v\_start\_date || ', 과정종료일 : ' || v\_end\_date || ', 강의실명 : ' || v\_classroom);  presult := 1;  end loop;  close vcursor;  exception  WHEN OTHERS THEN  presult := 0;  end procSelReview;  /  DECLARE  vresult number;  v\_curriculum\_num NUMBER := 4; -- 원하는 개설 과정 번호 지정  BEGIN  procSelReview(v\_curriculum\_num, vresult);  --dbms\_output.put\_line(vresult);  END;  / | | | |

| 요구사항번호 | **D-08(교육생)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 수업계획서 조회 | 작성자 | 차수민 |
| DML | | | |
| CREATE OR REPLACE PROCEDURE procPlan(  p\_student\_num NUMBER  )  IS  BEGIN  DBMS\_OUTPUT.PUT\_LINE('===============================[수업 계획서]=============================  ==');  FOR plan IN (SELECT p.plan\_num AS 계획서번호, ocur.open\_curriculum\_num AS 과정번호, p.plan\_week AS 주차,  p.plan\_content AS 내용  FROM tblPlan p  INNER JOIN tblOpenCurriculum ocur  ON p.open\_curriculum\_num = ocur.open\_curriculum\_num  INNER JOIN tblRegister r  ON ocur.open\_curriculum\_num = r.open\_curriculum\_num  INNER JOIN tblStudent s  ON r.student\_num = s.student\_num  WHERE s.student\_num = p\_student\_num)  LOOP  DBMS\_OUTPUT.PUT\_LINE('계획서번호: ' || plan.계획서번호);  DBMS\_OUTPUT.PUT\_LINE('과정번호: ' || plan.과정번호);  DBMS\_OUTPUT.PUT\_LINE('주차: ' || plan.주차);  DBMS\_OUTPUT.PUT\_LINE('내용: ' || plan.내용);  DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------------------------------');  END LOOP;  DBMS\_OUTPUT.PUT\_LINE('조회가 완료되었습니다.');  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('조회에 실패하였습니다.');  END procPlan;  BEGIN  procPlan(83);  END; | | | |

| 요구사항번호 | **D-09(교육생)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 수강내역 | 작성자 | 김혜민 |
| DML | | | |
| - 수강중인 강좌의 수강번호, 강좌명, 수강기간 등을 출력하여 도시한다.  - 수강번호 및 수강과정번호도 표기한다.  CREATE OR REPLACE PROCEDURE procStudentCurriculumInfo(  p\_student\_num NUMBER  )  IS  BEGIN  FOR curriculum\_record IN (  SELECT  s.student\_num AS 교육생번호,  s.name AS 교육생명,  tr.register\_num AS 수강과정번호,  toc.curriculum\_num AS 과정번호,  toc.period AS 과정기간,  cu.curriculum AS 과정명  FROM tblStudent s  INNER JOIN tblRegister tr ON s.student\_num = tr.student\_num  INNER JOIN tblOpenCurriculum toc ON tr.open\_curriculum\_num = toc.open\_curriculum\_num  INNER JOIN tblCurriculum cu ON toc.curriculum\_num = cu.curriculum\_num  WHERE s.student\_num = p\_student\_num  )  LOOP  DBMS\_OUTPUT.PUT\_LINE('교육생번호: ' || curriculum\_record.교육생번호 || ', ' ||  '교육생명: ' || curriculum\_record.교육생명 || ', ' ||  '수강과정번호: ' || curriculum\_record.수강과정번호 || ', ' ||  '과정번호: ' || curriculum\_record.과정번호 || ', ' ||  '과정기간: ' || curriculum\_record.과정기간 || ', ' ||  '과정명: ' || curriculum\_record.과정명);  END LOOP;  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('오류 발생: ' || SQLERRM);  END procStudentCurriculumInfo; | | | |

| 요구사항번호 | **D-10(교육생)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 과제 조회 | 작성자 | 이도훈 |
| DML | | | |
| CREATE OR REPLACE PROCEDURE procInfoTask  IS  BEGIN  BEGIN  FOR task\_record IN (  SELECT  s.name AS 교육생이름,  cur.curriculum AS 과정명,  su.subject AS 과목명,  t.task\_file AS 과제파일,  t.task\_date AS 제출일  FROM tbltask t  INNER JOIN tblcourse c ON c.course\_num = t.course\_num  INNER JOIN tblregister r ON r.register\_num = c.register\_num  INNER JOIN tblstudent s ON s.student\_num = r.student\_num  INNER JOIN tblOpenCurriculum cu ON cu.open\_curriculum\_num = r.open\_curriculum\_num  INNER JOIN tblCurriculum cur ON cur.curriculum\_num = cu.curriculum\_num  INNER JOIN tblOpenSubject o ON o.open\_subject\_num = c.open\_subject\_num  INNER JOIN tblAbleSubject a ON a.able\_subject\_num = o.able\_subject\_num  INNER JOIN tblsubject su ON su.subject\_num = a.subject\_num  )  LOOP  DBMS\_OUTPUT.PUT\_LINE(  '교육생이름: ' || task\_record.교육생이름 || ', ' ||  '과정명: ' || task\_record.과정명 || ', ' ||  '과목명: ' || task\_record.과목명 || ', ' ||  '과제파일: ' || task\_record.과제파일 || ', ' ||  '제출일: ' || TO\_CHAR(task\_record.제출일, 'YYYY-MM-DD HH24:MI:SS')  );  END LOOP;  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('오류 발생: ' || SQLERRM);  END;  END procInfoTask; | | | |

| 요구사항번호 | **D-11(교육생)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 프로젝트 조회 | 작성자 | 이도훈 |
| DML | | | |
| CREATE OR REPLACE PROCEDURE procGetProjectInfo  IS  BEGIN  BEGIN  FOR project\_record IN (  SELECT  s.name AS 교육생이름,  cur.curriculum AS 과정명,  su.subject AS 과목명,  j.project\_file AS 프로젝트파일,  j.project\_date AS 제출일  FROM tblproject j  INNER JOIN tblcourse c ON c.course\_num = j.course\_num  INNER JOIN tblregister r ON r.register\_num = c.register\_num  INNER JOIN tblstudent s ON s.student\_num = r.student\_num  INNER JOIN tblOpenCurriculum cu ON cu.open\_curriculum\_num = r.open\_curriculum\_num  INNER JOIN tblCurriculum cur ON cur.curriculum\_num = cu.curriculum\_num  INNER JOIN tblOpenSubject o ON o.open\_subject\_num = c.open\_subject\_num  INNER JOIN tblAbleSubject a ON a.able\_subject\_num = o.able\_subject\_num  INNER JOIN tblsubject su ON su.subject\_num = a.subject\_num  )  LOOP  DBMS\_OUTPUT.PUT\_LINE(  '교육생이름: ' || project\_record.교육생이름 || ', ' ||  '과정명: ' || project\_record.과정명 || ', ' ||  '과목명: ' || project\_record.과목명 || ', ' ||  '프로젝트파일: ' || project\_record.프로젝트파일 || ', ' ||  '제출일: ' || TO\_CHAR(project\_record.제출일, 'YYYY-MM-DD HH24:MI:SS')  );  END LOOP;  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('오류 발생: ' || SQLERRM);  END;  END procGetProjectInfo; | | | |

| 요구사항번호 | **D-12(교육생)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 수강후기 등록 및 조회 | 작성자 | 이정은 |
| DML | | | |
| --1. 수강후기 수정기능  create or replace procedure procUpdateReview(  preview varchar2,  presult out number --확인을 위한 반환 변수  )  is  begin  update tblreview set review = preview where review\_num=1;  presult := 1;  exception  when others then  presult := 0; --실패  end procUpdateReview;  declare  vresult number;  begin  procUpdateReview('진짜진짜 완전 강추!! 실력 단번에 늘 수 있어요.', vresult);  dbms\_output.put\_line(vresult);  end;  --2. 수강후기 삭제기능  삭제 전에 임의로 아무거나 추가하기 -> insert into tblreview values(0, 1, '굿잡', sysdate);  create or replace procedure procDelReview(  preview\_num number,  presult out number --확인을 위한 반환 변수  )  is  begin  delete from tblreview where review\_num=preview\_num;  presult := 1; --성공  exception  when others then  presult := 0; --실패  end procDelReview;  DECLARE  vresult NUMBER;  BEGIN  procDelReview(0, vresult);  dbms\_output.put\_line(vresult);  END;  --3. 특정 수강과정 별 수강후기(수강과정, 후기내용, 작성일, 작성자)를 조회한다.  create or replace procedure procSelReview(  pregister\_num number,  presult out number  )  is  v\_register\_num tblReview.register\_num%type;  v\_curriculum tblCurriculum.curriculum%type;  v\_review tblReview.review%type;  v\_review\_date tblReview.review\_date%type;  v\_name tblStudent.name%type;    cursor vcursor is  select  tr.register\_num as 수강과정번호,  c.curriculum as 수강과정명,  tr.review as 후기내용,  tr.review\_date as 작성일,  ts.name as 작성자  from tblReview tr  inner join tblComplete tcp on tr.register\_num = tcp.register\_num  inner join tblRegister tbr on tcp.register\_num = tbr.register\_num  inner join tblStudent ts on tbr.student\_num = ts.student\_num  inner join tblOpenCurriculum toc on tbr.open\_curriculum\_num = toc.open\_curriculum\_num  inner join tblCurriculum c on c.curriculum\_num = toc.curriculum\_num;  begin  open vcursor;  loop  FETCH vcursor INTO v\_register\_num, v\_curriculum, v\_review, v\_review\_date, v\_name;  exit when vcursor%notfound;    DBMS\_OUTPUT.PUT\_LINE('수강과정번호 : ' || v\_register\_num || ', 수강과정명 : ' || v\_curriculum || ', 후기내용 : ' || v\_review || ', 작성일 : ' || v\_review\_date || ', 작성자' || v\_name);  presult := 1;  end loop;  close vcursor;  exception  WHEN OTHERS THEN  presult := 0;  end procSelReview;  /  DECLARE  vresult number;  v\_register\_num NUMBER := 13; -- 원하는 개설 과정 번호 지정  BEGIN  procSelReview(v\_register\_num, vresult);  --dbms\_output.put\_line(vresult);  END;  / | | | |

| 요구사항번호 | **D-13(교육생)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 취업현황 조회 | 작성자 | 김혜민 |
| DML | | | |
| CREATE OR REPLACE PROCEDURE procStudentEmploymentInfo  IS  BEGIN  FOR employment\_record IN (  SELECT  st.student\_num AS 학생번호,  st.name AS 학생명,  tep.register\_num AS 수강과정번호,  tbc.curriculum AS 과정명,  tep.employment\_state AS 취업여부,  tep.field AS 취업분야,  tep.insurance\_state AS 고용보험가입여부  FROM tblStudent st  INNER JOIN tblRegister tr ON tr.student\_num = st.student\_num  INNER JOIN tblComplete tc ON tc.register\_num = tr.register\_num  INNER JOIN tblEmployment tep ON tep.register\_num = tc.register\_num  INNER JOIN tblOpenCurriculum toc ON tr.open\_curriculum\_num = toc.open\_curriculum\_num  INNER JOIN tblCurriculum tbc ON tbc.curriculum\_num = toc.curriculum\_num  )  LOOP  DBMS\_OUTPUT.PUT\_LINE('학생번호: ' || employment\_record.학생번호 || ', ' ||  '학생명: ' || employment\_record.학생명 || ', ' ||  '수강과정번호: ' || employment\_record.수강과정번호 || ', ' ||  '과정명: ' || employment\_record.과정명 || ', ' ||  '취업여부: ' || employment\_record.취업여부 || ', ' ||  '취업분야: ' || employment\_record.취업분야 || ', ' ||  '고용보험가입여부: ' || employment\_record.고용보험가입여부);  END LOOP;  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('오류 발생: ' || SQLERRM);  END procStudentEmploymentInfo; | | | |

| **<뷰>** |
| --- |

| 요구사항번호 | **B- 05(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 관리 | 작성자 | 이도훈 |
| DML | | | |
| CREATE VIEW vwStudentInfo AS  SELECT  s.name AS 교육생이름,  s.pw AS 주민번호뒷자리,  s.tel AS 전화번호,  r.register\_date AS 등록일,  c.curriculum AS 과정명  FROM tblstudent s  INNER JOIN tblregister r  ON s.student\_num = r.register\_num  INNER JOIN tblOpenCurriculum o  ON o.open\_curriculum\_num = r.open\_curriculum\_num  INNER JOIN tblCurriculum c  ON c.curriculum\_num = o.curriculum\_num; | | | |

| 요구사항번호 | **B-08(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 수강과정 관리 | 작성자 | 황주원 |
| DML | | | |
| CREATE OR REPLACE VIEW vwregisterRead AS  SELECT  r.REGISTER\_NUM AS 수강과정번호,  s.NAME AS 교육생이름,  c.CURRICULUM AS 과정명,  r.REGISTER\_DATE AS 등록일  FROM tblregister r  INNER JOIN tblOpenCurriculum oc ON oc.OPEN\_CURRICULUM\_NUM = r.OPEN\_CURRICULUM\_NUM  INNER JOIN tblCurriculum c ON c.CURRICULUM\_NUM = oc.CURRICULUM\_NUM  INNER JOIN tblStudent s ON s.STUDENT\_NUM = r.STUDENT\_NUM ; | | | |

| 요구사항번호 | **B- 09(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 중도탈락내역 및 수료내역 관리 | 작성자 | 황주원 |
| DML | | | |
| CREATE OR REPLACE VIEW vwcompleteRead AS  SELECT  r.REGISTER\_NUM AS 수강과정번호,  s.NAME AS 수강생이름,  c.CURRICULUM AS 과정명,  co.COMPLETE\_DATE AS 수료일  FROM tblRegister r  INNER JOIN tblstudent s ON s.STUDENT\_NUM = r.STUDENT\_NUM  INNER JOIN tblOpenCurriculum oc ON oc.OPEN\_CURRICULUM\_NUM = r.OPEN\_CURRICULUM\_NUM  INNER JOIN tblCurriculum c ON c.CURRICULUM\_NUM = oc.CURRICULUM\_NUM  INNER JOIN tblComplete co ON co.REGISTER\_NUM = r.REGISTER\_NUM;  CREATE OR REPLACE VIEW vwdropRead AS  select  c.curriculum as 과정명,  s.name as 교육생이름,  d.drop\_date as 중도탈락일  from tblregister r inner join tblstudent s on s.student\_num=r.student\_num  inner join tbldrop d on r.register\_num=d.register\_num  inner join tblopencurriculum oc on r.OPEN\_CURRICULUM\_NUM=oc.OPEN\_CURRICULUM\_NUM  inner join tblcurriculum c on c.curriculum\_num=oc.curriculum\_num; | | | |

| 요구사항번호 | **B- 11(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 취업현황 관리 | 작성자 | 이도훈 |
| DML | | | |
| CREATE VIEW vwEmploymentInfo AS  SELECT  stu.name AS 교육생명,  emp.register\_num AS 수강과정번호,  emp.employment\_state AS 취업여부,  emp.field AS 취업분야,  emp.insurance\_state AS 고용보험가입여부  FROM tblregister reg  INNER JOIN tblstudent stu  ON stu.student\_num = reg.student\_num  LEFT OUTER JOIN tblcomplete com  ON com.register\_num = reg.register\_num  INNER JOIN tblemployment emp  ON emp.register\_num = com.register\_num; | | | |

| 요구사항번호 | **D-02(교육생)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 성적 조회 | 작성자 | 차수민 |
| DML | | | |
| CREATE OR REPLACE VIEW vwStudScore AS  SELECT s.student\_num AS 교육생번호, t.open\_subject\_num AS 과목번호, sub.subject AS 과목명, osub.start\_date AS 과목시작일,  osub.end\_date AS 과목종료일, b.book AS 교재명, p.name AS 교사명, t.att\_ratio AS 출결배점, t.write\_ratio AS 필기배점,  t.practical\_ratio AS 실기배점, sc.att\_score AS 출결점수, sc.write\_score AS 필기점수, sc.practical\_score AS 실기점수,  t.test\_date AS 시험일, t.test\_file AS 시험문제  FROM tblOpenSubject osub  INNER JOIN tblCourse c  ON osub.open\_subject\_num = c.open\_subject\_num  INNER JOIN tblRegister r  ON c.register\_num = r.register\_num  INNER JOIN tblStudent s  ON r.student\_num = s.student\_num  LEFT OUTER JOIN tblComplete c  ON r.register\_num = c.register\_num  LEFT OUTER JOIN tblDrop d  ON r.register\_num = d.register\_num  LEFT OUTER JOIN tblScore sc  ON c.course\_num = sc.course\_num  LEFT OUTER JOIN tblTest t  ON osub.open\_subject\_num = t.open\_subject\_num  INNER JOIN tblBook b  ON osub.book\_num = b.book\_num  INNER JOIN tblAbleSubject asub  ON osub.able\_subject\_num = asub.able\_subject\_num  INNER JOIN tblProfessor p  ON asub.professor\_num = p.professor\_num  INNER JOIN tblSubject sub  ON asub.subject\_num = sub.subject\_num  INNER JOIN tblStudent s  ON r.student\_num = s.student\_num; | | | |

| 요구사항번호 | **D-08(교육생)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 질문 | 작성자 | 이도훈 |
| DML | | | |
| CREATE VIEW vwQuestionInfo AS  SELECT  s.name AS 교육생명,  cla.classroom AS 강의실,  cu.curriculum AS 과정명,  q.question AS 질문내용,  q.question\_date AS 질문일  FROM tblquestion q  INNER JOIN tblcourse c  ON c.course\_num = q.course\_num  INNER JOIN tblregister r  ON r.register\_num = c.register\_num  INNER JOIN tblstudent s  ON s.student\_num = r.student\_num  INNER JOIN tblopencurriculum o  ON o.open\_curriculum\_num = r.open\_curriculum\_num  INNER JOIN tblcurriculum cu  ON cu.curriculum\_num = o.curriculum\_num  INNER JOIN tblclassroom cla  ON cla.classroom\_num = o.classroom\_num; | | | |

| 요구사항번호 | **D-10(교육생)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 과제 조회 | 작성자 | 이도훈 |
| DML | | | |
| CREATE VIEW vwtaskinfo AS  SELECT  s.name AS 교육생이름,  cur.curriculum AS 과정명,  su.subject AS 과목명,  t.task\_file AS 과제파일,  t.task\_date AS 제출일  from tbltask t  INNER JOIN tblcourse c  ON c.course\_num = t.course\_num  INNER JOIN tblregister r  ON r.register\_num = c.register\_num  INNER JOIN tblstudent s  ON s.student\_num = r.student\_num  INNER JOIN tblOpenCurriculum cu  ON cu.open\_curriculum\_num = r.open\_curriculum\_num  INNER JOIN tblCurriculum cur  ON cur.curriculum\_num = cu.curriculum\_num  INNER JOIN tblOpenSubject o  ON o.open\_subject\_num = c.open\_subject\_num  INNER JOIN tblAbleSubject a  ON a.able\_subject\_num = o.able\_subject\_num  INNER JOIN tblsubject su  ON su.subject\_num = a.subject\_num; | | | |

| 요구사항번호 | **D-11(교육생)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 프로젝트 조회 | 작성자 | 이도훈 |
| DML | | | |
| CREATE VIEW vwProjectInfo AS  SELECT  s.name AS 교육생이름,  cur.curriculum AS 과정명,  su.subject AS 과목명,  j.project\_file AS 프로젝트파일,  j.project\_date AS 제출일  from tblproject j  INNER JOIN tblcourse c  ON c.course\_num = j.course\_num  INNER JOIN tblregister r  ON r.register\_num = c.register\_num  INNER JOIN tblstudent s  ON s.student\_num = r.student\_num  INNER JOIN tblOpenCurriculum cu  ON cu.open\_curriculum\_num = r.open\_curriculum\_num  INNER JOIN tblCurriculum cur  ON cur.curriculum\_num = cu.curriculum\_num  INNER JOIN tblOpenSubject o  ON o.open\_subject\_num = c.open\_subject\_num  INNER JOIN tblAbleSubject a  ON a.able\_subject\_num = o.able\_subject\_num  INNER JOIN tblsubject su  ON su.subject\_num = a.subject\_num; | | | |

| 요구사항번호 | **C-01(교사)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 강의 스케줄 조회 | 작성자 | 차수민 |
| DML | | | |
| CREATE OR REPLACE VIEW vwSchedule AS  SELECT ocur.open\_curriculum\_num AS 개설과정번호,  cur.curriculum AS 과정명,  ocur.start\_date AS 과정시작일,  ocur.end\_date AS 과정종료일,  osub.open\_subject\_num AS 개설과목번호,  sub.subject AS 과목명,  osub.start\_date AS 과목시작일,  osub.end\_date AS 과목종료일,  r.classroom AS 강의실명,  b.book AS 교재명,  p.professor\_num AS 교사번호  FROM tblOpenCurriculum ocur  INNER JOIN tblOpenSubject osub  ON ocur.open\_curriculum\_num = osub.open\_curriculum\_num  INNER JOIN tblClassroom r  ON ocur.classroom\_num = r.classroom\_num  INNER JOIN tblCurriculum cur  ON ocur.curriculum\_num = cur.curriculum\_num  INNER JOIN tblAbleSubject asub  ON osub.able\_subject\_num = asub.able\_subject\_num  INNER JOIN tblProfessor p  ON asub.professor\_num = p.professor\_num  INNER JOIN tblSubject sub  ON asub.subject\_num = sub.subject\_num  INNER JOIN tblBook b  ON osub.book\_num = b.book\_num  INNER JOIN tblCourse c  ON osub.open\_subject\_num = c.open\_subject\_num  ORDER BY 개설과정번호, 개설과목번호;  CREATE OR REPLACE VIEW vwStudent AS  SELECT ocur.open\_curriculum\_num AS 개설과정번호, s.student\_num AS 교육생번호, s.name AS 이름, s.tel AS 전화번호, r.register\_date AS 등록일,  CASE  WHEN c.complete\_date IS NOT NULL THEN '수료'  WHEN d.drop\_date IS NOT NULL THEN '중도탈락'  ELSE '해당없음'  END AS 수료여부  FROM tblOpenCurriculum ocur  INNER JOIN tblRegister r  ON ocur.open\_curriculum\_num = r.open\_curriculum\_num  INNER JOIN tblStudent s  ON r.student\_num = s.student\_num  LEFT OUTER JOIN tblComplete c  ON r.register\_num = c.register\_num  LEFT OUTER JOIN tblDrop d  ON r.register\_num = d.register\_num  ORDER BY 개설과정번호; | | | |

| 요구사항번호 | **C-03(교사)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 성적 입출력 | 작성자 | 차수민 |
| DML | | | |
| CREATE OR REPLACE VIEW vwScore AS  SELECT t.open\_subject\_num AS 개설과목번호, s.student\_num AS 교육생번호, s.name AS 이름, s.tel AS 전화번호, r.register\_date AS 등록일,  CASE  WHEN c.complete\_date IS NOT NULL THEN '수료'  WHEN d.drop\_date IS NOT NULL THEN '중도탈락'  ELSE '해당없음'  END AS 수료여부,  c.course\_num AS 수강번호,  t.test\_num AS 시험번호,  sc.att\_score AS 출결점수,  sc.write\_score AS 필기점수,  sc.practical\_score AS 실기점수  FROM tblOpenSubject osub  INNER JOIN tblCourse c  ON osub.open\_subject\_num = c.open\_subject\_num  INNER JOIN tblRegister r  ON c.register\_num = r.register\_num  INNER JOIN tblStudent s  ON r.student\_num = s.student\_num  LEFT OUTER JOIN tblComplete c  ON r.register\_num = c.register\_num  LEFT OUTER JOIN tblDrop d  ON r.register\_num = d.register\_num  LEFT OUTER JOIN tblScore sc  ON c.course\_num = sc.course\_num  LEFT OUTER JOIN tblTest t  ON osub.open\_subject\_num = t.open\_subject\_num; | | | |

| 요구사항번호 | **C-06(교사)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 질문 및 답변 입출력 | 작성자 | 차수민 |
| DML | | | |
| CREATE OR REPLACE VIEW vwQna AS  SELECT s.student\_num AS 교육생번호, s.name AS 교육생,  q.question\_num AS 질문번호, q.question AS 질문, q.question\_date AS 질문일,  a.answer\_num AS 답변번호, a.answer AS 답변, a.answer\_date AS 답변일,  p.professor\_num AS 교사번호, p.name AS 교사  FROM tblQuestion q  LEFT OUTER JOIN tblAnswer a  ON q.question\_num = a.question\_num  INNER JOIN tblCourse c  ON q.course\_num = c.course\_num  INNER JOIN tblOpenSubject osub  ON c.open\_subject\_num = osub.open\_subject\_num  INNER JOIN tblAbleSubject asub  ON osub.able\_subject\_num = asub.able\_subject\_num  INNER JOIN tblProfessor p  ON asub.professor\_num = p.professor\_num  INNER JOIN tblRegister r  ON c.register\_num = r.register\_num  INNER JOIN tblStudent s  ON r.student\_num = s.student\_num; | | | |

| 요구사항번호 | **B-04(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 개설 과목 관리 | 작성자 | 안대명 |
| DML | | | |
| -- CREATE VIEW 문을 사용하여 뷰 생성  CREATE OR REPLACE VIEW vwOpenCurriculumInfo AS  SELECT  OC.curriculum\_num AS 과정번호,  C.curriculum AS 과정명,  OC.start\_date AS 과정시작일,  OC.end\_date AS 과정종료일,  OC.classroom\_num AS 강의실번호,  OS.open\_subject\_num AS 개설과목번호,  OS.able\_subject\_num AS 강의가능과목번호,  S.subject AS 과목명,  OS.start\_date AS 과목시작일,  OS.end\_date AS 과목종료일,  OS.book\_num AS 교재번호,  B.book AS 교재명,  OS.sub\_progress\_state AS 강의진행여부  FROM tblOpenCurriculum OC  INNER JOIN tblCurriculum C ON OC.curriculum\_num = C.curriculum\_num  INNER JOIN tblOpenSubject OS ON OC.open\_curriculum\_num = OS.open\_curriculum\_num  INNER JOIN tblAbleSubject ASB ON OS.able\_subject\_num = ASB.able\_subject\_num  INNER JOIN tblSubject S ON ASB.subject\_num = S.subject\_num  LEFT JOIN tblBook B ON OS.book\_num = B.book\_num;  -- 생성한 뷰를 사용하여 데이터 조회  SELECT \* FROM vwOpenCurriculumInfo WHERE 과정번호 = 1; | | | |

| 요구사항번호 | **B-10(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 수강후기 관리 | 작성자 | 안대명 |
| DML | | | |
| -- CREATE VIEW 문을 사용하여 뷰 생성  CREATE OR REPLACE VIEW vwReviewInfo AS  SELECT  R.review\_num AS Review\_Number,  R.register\_num AS Register\_Number,  S.name AS Student\_Name,  R.review AS Review\_Text,  R.review\_date AS Review\_Date  FROM tblReview R  INNER JOIN tblRegister RG ON R.register\_num = RG.register\_num  INNER JOIN tblStudent S ON RG.student\_num = S.student\_num;  -- 생성한 뷰를 사용하여 데이터 조회  SELECT \* FROM vwReviewInfo; | | | |

| 요구사항번호 | **B-13(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 지원자 면접 | 작성자 | 안대명 |
| DML | | | |
| -- CREATE VIEW 문을 사용하여 뷰 생성  CREATE OR REPLACE VIEW vwApplicantInterview AS  SELECT  A.applicant\_num AS Applicant\_Number,  A.name AS Applicant\_Name,  A.tel AS Applicant\_Tel,  I.interview\_num AS Interview\_Number,  I.interview\_date AS Interview\_Date,  I.interview\_result AS Interview\_Result,  AD.name AS Admin\_Name,  AD.pw AS Admin\_Password,  AD.tel AS Admin\_Tel  FROM tblApplicant A  INNER JOIN tblInterview I ON A.applicant\_num = I.applicant\_num  INNER JOIN tblAdmin AD ON I.admin\_num = AD.admin\_num;  -- 생성한 뷰를 사용하여 데이터 조회  SELECT \* FROM vwApplicantInterview; | | | |

| 요구사항번호 | **B-03(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 개설 과정 관리 | 작성자 | 이정은 |
| DML | | | |
| -- CREATE VIEW 문을 사용하여 뷰 생성  CREATE OR REPLACE VIEW vwOpenCurriculum AS  SELECT  c.curriculum AS 개설과정명,  s.subject AS 과목명,  os.start\_date AS 과목시작일,  os.end\_date AS 과목종료일,  b.book AS 교재명,  p.name AS 교수명  FROM tblopensubject os INNER JOIN tblbook b ON b.book\_num=os.book\_num  INNER JOIN tblablesubject ab ON os.able\_subject\_num=ab.able\_subject\_num  INNER JOIN tblsubject s ON s.subject\_num=ab.subject\_num  INNER JOIN tblprofessor p ON p.professor\_num = ab.professor\_num  INNER JOIN tblopencurriculum oc ON oc.open\_curriculum\_num = os.open\_curriculum\_num  INNER JOIN tblcurriculum c ON c.curriculum\_num=oc.curriculum\_num; | | | |

| 요구사항번호 | **D-01(교육생)** | 작성일 | 2023-09-15 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 개인정보 | 작성자 | 김혜민 |
| DML | | | |
| -- CREATE VIEW 문을 사용하여 뷰 생성  CREATE OR REPLACE VIEW vwStudentInfo  select  std.name as "이름",  std.pw as "주민번호",  std.tel as "전화번호",  cul.curriculum as "과정명",  toc.start\_date as "과정시작일",  toc.end\_date as "과정종료일",  tcr.classroom\_num as "강의실번호"  from tblStudent std  inner join tblRegister tbr  on std.student\_num = tbr.student\_num  inner join tblOpenCurriculum toc  on toc.open\_curriculum\_num = tbr.open\_curriculum\_num  inner join tblCurriculum cul  on cul.curriculum\_num = toc.curriculum\_num  inner join tblClassroom tcr  on tcr.classroom\_num = toc.classroom\_num  -- 생성한 뷰를 사용하여 데이터 조회  SELECT \* FROM vwStudentInfo; | | | |

| 요구사항번호 | **B-07(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 출결 관리 및 출결 조회 | 작성자 | 이정은 |
| DML | | | |
| -- CREATE VIEW 문을 사용하여 뷰 생성  CREATE OR REPLACE VIEW vwAtt AS  SELECT  oc.open\_curriculum\_num AS 개설과정,  s.name AS 교육생,  a.att\_date AS 날짜,  a.att\_state AS 근태현황  FROM tblstudent s INNER JOIN tblregister r ON s.student\_num=r.student\_num  INNER JOIN tblattendance a ON r.register\_num=a.register\_num  INNER JOIN tblopencurriculum oc ON oc.open\_curriculum\_num = r.open\_curriculum\_num  WHERE oc.open\_curriculum\_num=4  GROUP BY a.att\_date, a.att\_state, s.name, s.student\_num, oc.open\_curriculum\_num  ORDER BY s.student\_num, a.att\_date; | | | |

| 요구사항번호 | **B-14(관리자)** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 공지사항 | 작성자 | 이정은 |
| DML | | | |
| -- CREATE VIEW 문을 사용하여 뷰 생성  CREATE OR REPLACE VIEW vwNotice AS  SELECT  oc.open\_curriculum\_num AS 개설과정번호,  c.curriculum AS 과정명,  oc.classroom\_num AS 강의실번호,  n.notice AS 공지사항내용,  n.notice\_date AS 공지일,  n.views AS 조회수  FROM tblnotice n INNER JOIN tblopencurriculum oc ON oc.open\_curriculum\_num=n.open\_curriculum\_num  INNER JOIN tblcurriculum c ON c.curriculum\_num=oc.curriculum\_num; | | | |

| 요구사항번호 | **D-02(교육생)** | 작성일 | 2023-09-15 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 성적 조회 | 작성자 | 김혜민 |
| DML | | | |
| -- CREATE VIEW 문을 사용하여 뷰 생성  CREATE OR REPLACE VIEW vwScoreInfo AS  select  distinct  std.student\_num as "학생번호",  tbs.subject as "과목명",  tos.start\_date as "과목시작일",  tos.end\_date as "과목종료일",  tb.book as "교재명",  tpf.name as "교사명",  tt.att\_ratio as "출결배점",  tt.write\_ratio as "필기배점",  tt.practical\_ratio as "실기배점",  ts.att\_score as "출결점수",  ts.write\_score as "필기점수",  ts.practical\_score as "실기점수",  tt.test\_date as "시험날짜",  tt.test\_file as "시험문제"  from tblstudent std  inner join tblRegister tr  on std.student\_num = tr.student\_num  inner join tblCourse tc  on tr.register\_num = tc.register\_num  inner join tblScore ts  on ts.course\_num = tc.course\_num  inner join tblTest tt  on ts.test\_num = tt.test\_num  inner join tblOpenSubject tos  on tt.open\_subject\_num = tos.open\_subject\_num  inner join tblAbleSubject tas  on tas.able\_subject\_num = tos.able\_subject\_num  inner join tblSubject tbs  on tbs.subject\_num = tas.subject\_num  inner join tblProfessor tpf  on tas.professor\_num = tpf.professor\_num  inner join tblBook tb  on tos.book\_num = tb.book\_num  where std.student\_num = 224 --로그인한 학생의 번호 입력  order by "학생번호" asc; | | | |

| 요구사항번호 | **D-03(교육생)** | 작성일 | 2023-09-15 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 출결 관리/조회 | 작성자 | 김혜민 |
| DML | | | |
| -- CREATE VIEW 문을 사용하여 뷰 생성  CREATE OR REPLACE VIEW StudentAttendaceInfo AS  select  ta.att\_date as "날짜",  ta.register\_num "수강과정번호",  tc.curriculum as "과정명",  ts.student\_num as "교육생번호",  ts.name as "교육생이름",  ta.att\_state as "근태상황"  from tblAttendance ta  inner join tblRegister tr  on ta.register\_num = tr.register\_num  inner join tblStudent ts  on tr.student\_num= ts.student\_num  inner join tblOpenCurriculum toc  on tr.open\_curriculum\_num = toc.open\_curriculum\_num  inner join tblCurriculum tc  on toc.curriculum\_num = tc.curriculum\_num; | | | |

| 요구사항번호 | **D-04(교육생)** | 작성일 | 2023-04-07 |
| --- | --- | --- | --- |
| 요구사항명 | 수강후기 조회기능 | 작성자 | 김혜민 |
| DML | | | |
| -- CREATE VIEW 문을 사용하여 뷰 생성  CREATE OR REPLACE VIEW StudentReview AS  select  tr.register\_num as 수강과정번호,  tbc.curriculum as 수강과정명,  tr.review as 후기내용,  tr.review\_date as 작성일,  ts.name as 작성자  from tblReview tr  inner join tblComplete tcp  on tr.register\_num = tcp.register\_num  inner join tblRegister tbr  on tcp.register\_num = tbr.register\_num  inner join tblStudent ts  on tbr.student\_num = ts.student\_num  inner join tblOpenCurriculum toc  on tbr.open\_curriculum\_num = toc.open\_curriculum\_num  inner join tblCurriculum tbc  on tbc.curriculum\_num = toc.curriculum\_num; | | | |

| 요구사항번호 | **D-05(교육생)** | 작성일 | 2023-09-15 |
| --- | --- | --- | --- |
| 요구사항명 | 취업현황 조회 | 작성자 | 김혜민 |
| DML | | | |
| -- CREATE VIEW 문을 사용하여 뷰 생성  CREATE OR REPLACE VIEW JobStateInfo AS  select  st.student\_num as 학생번호,  st.name as 학생명,  tep.register\_num as 수강과정번호,  tbc.curriculum as 과정명,  tep.employment\_state as 취업여부,  tep.field as 취업분야,  tep.insurance\_state as 고용보험가입여부  from tblStudent st  inner join tblRegister tr  on tr.student\_num = st.student\_num  inner join tblComplete tc  on tc.register\_num = tr.register\_num  inner join tblEmployment tep  on tep.register\_num = tc.register\_num  inner join tblOpenCurriculum toc  on tr.open\_curriculum\_num = toc.open\_curriculum\_num  inner join tblCurriculum tbc  on tbc.curriculum\_num = toc.curriculum\_num ; | | | |

| 요구사항번호 | **D-06(교육생)** | 작성일 | 2023-09-15 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 추천도서 | 작성자 | 김혜민 |
| DML | | | |
| -- CREATE VIEW 문을 사용하여 뷰 생성  CREATE OR REPLACE VIEW RecommendBook AS  select  tbb.bestboon\_num as 추천도서번호,  tbb.subject\_num as 과목번호,  tbb.book as 도서명,  tbb.public as 출판사,  tpf.name as 추천교사명  from tblBestBook tbb  inner join tblSubject ts  on tbb.subject\_num = ts.subject\_num  inner join tblAbleSubject tas  on tas.subject\_num = ts.subject\_num  inner join tblProfessor tpf  on tpf.professor\_\_num = tas.professor\_num  where tas.professor\_num = 1; – 교사번호로 책 검색 | | | |

| 요구사항번호 | **D-07(교육생)** | 작성일 | 2023-09-15 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 상담 일지 | 작성자 | 김혜민 |
| DML | | | |
| -- CREATE VIEW 문을 사용하여 뷰 생성  CREATE OR REPLACE VIEW StudentCounselInfo AS  select  tbc.counsel as 상담번호,  std.name as 교육생이름,  tbc.register\_num as 수강과정번호,  tbc.counsel AS 상담내용,  tbc.counsel\_date as 상담일,  cul.curriculum as 과정명,  toc.start\_date as 과정시작일,  toc.end\_date as 과정종료일,  cla.classroom\_num as 강의실번호  from tblStudent std  inner join tblRegister tbr  on std.student\_num = tbr.student\_num  inner join tblOpenCurriculum toc  on toc.open\_curriculum\_num = tbr.open\_curriculum\_num  inner join tblCurriculum cul  on cul.curriculum\_num = toc.curriculum\_num  inner join tblCounsel tbc  on tbr.register\_num = tbc.register\_num  INNER JOIN tblClassroom cla  ON cla.classroom\_num = toc.classroom\_num; | | | |

| 요구사항번호 | **D-09(교육생)** | 작성일 | 2023-09-15 |
| --- | --- | --- | --- |
| 요구사항명 | 개설과정 조회 | 작성자 | 김혜민 |
| DML | | | |
| -- CREATE VIEW 문을 사용하여 뷰 생성  CREATE OR REPLACE VIEW StudentProcessInfo AS  select  toc.curriculum\_num AS 과정번호,  cu.curriculum AS 과정명,  toc.period AS 과정기간,  toc.start\_date AS 과정시작일,  toc.end\_date AS 과정종료일,  tc.classroom AS 강의실명  from tblOpenCurriculum toc  inner join tblCurriculum cu  on toc.curriculum\_num = cu.curriculum\_num  inner join tblClassroom tc  – 개설과정 선택 시 과목정보 및 교육생 정보 출력  -- 개설과정 선택 시 CREATE VIEW 문을 사용하여 뷰 생성  CREATE OR REPLACE VIEW StudentProcessSelect AS  select  s.student\_num AS 교육생번호,  s.name AS 이름,  s.pw AS 주민번호 뒷자리,  s.tel AS 전화번호,  toc.curriculum\_num AS 과정번호,  cu.curriculum\_num AS 과정명,  b.book AS 교재명,  b.publisher AS 출판사,  ts.subject\_num 과목명,  tos.start\_date AS 과목시작일,  tos.end\_date AS 과목종료일,  ts.common\_sub\_state AS 공통과목여부,  pf.name AS 교사명  from tblOpenSubject tos  inner join tboOpenCurriculum toc  on toc.open\_curriculum\_num = tos.open\_curriculum\_num  inner join tblBook b  on b.book\_num = tos.book\_num  inner join tblAbleSubject tas  on tas.able\_subject\_num = tos.able\_subject\_num  inner join tblSubject ts  ts.subject\_num = tas.subject\_num  tblProfessor pf  pf.professor\_num = tas.professor\_num  inner join tblCurriculum cu  toc.curriculum\_num = cu.curriculum\_num  inner join tblRegister tr  tr.open\_curriculum\_num = toc.curriculum\_num  inner join tblStudent s  s.student\_num = tr.student\_num; | | | |

| 요구사항번호 | **D-10(교육생)** | 작성일 | 2023-09-15 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 수업계획서 | 작성자 | 김혜민 |
| DML | | | |
| -- CREATE VIEW 문을 사용하여 뷰 생성  CREATE OR REPLACE VIEW CoursePlanInfo AS  select  p.plan\_num AS 계획서번호,  p.open\_curriculum\_num AS 개설과정번호,  cu.curriculum AS 과정명,  toc.start\_date AS 과정시작일,  toc.end\_date AS 과정종료일,  toc.period AS 과정기간,  p.plan\_week AS 주차,  p.plan\_content AS 수업내용  from tblOpenCurriculum toc  inner join tblPlan p  on p.open\_curriculum\_num = toc.open\_curriculum\_num  inner join tblCuriculum cu  on cu.curriculum\_num = toc.curriculum\_num  where toc.open\_curriculum\_num = 10; – 개설된 강좌번호로 수업계획서 조회 | | | |

| 요구사항번호 | **D-11(교육생)** | 작성일 | 2023-09-15 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 수강내역 | 작성자 | 김혜민 |
| DML | | | |
| -- CREATE VIEW 문을 사용하여 뷰 생성  CREATE OR REPLACE VIEW StudentRegisterInfo AS  SELECT  s.student\_num AS 교육생번호,  s.name AS 교육생명,  tr.register\_num AS 수강과정번호,  toc.curriculum\_num AS 과정번호,  toc.period AS 과정기간,  cu.curriculum AS 과정명,  FROM tblStudent s  INNER JOIN tblRegister tr  ON s.student\_num = tr.student\_num  INNER JOIN tblOpenCurriculum  ON tr.open\_curriculum\_num = toc.open\_curriculum\_num  INNER JOIN tblCurriculum cu  ON toc.curriculum\_num = cu.curriculum\_num; | | | |

| **<트리거>** |
| --- |

| 요구사항번호 | **Trigger-01** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 관리자 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogAdmin(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_admin\_seq;  CREATE OR REPLACE TRIGGER trgLogAdmin  AFTER  INSERT OR UPDATE OR DELETE  ON tblAdmin  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogAdmin VALUES (log\_admin\_seq.nextval, v\_message, DEFAULT);  END trgLogAdmin; | | | |

| 요구사항번호 | **Trigger-02** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 관리자출퇴근내역 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogAdminAttendance(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_admin\_attendance\_seq;  CREATE OR REPLACE TRIGGER trgLogAdminAttendance  AFTER  INSERT OR UPDATE OR DELETE  ON tblAdminAttendance  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogAdminAttendance VALUES (log\_admin\_attendance\_seq.nextval, v\_message, DEFAULT);  END trgLogAdminAttendance; | | | |

| 요구사항번호 | **Trigger-03** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 면접 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogInterview(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_interview\_seq;  CREATE OR REPLACE TRIGGER trgLogInterview  AFTER  INSERT OR UPDATE OR DELETE  ON tblInterview  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogInterview VALUES (log\_interview\_seq.nextval, v\_message, DEFAULT);  END trgLogInterview; | | | |

| 요구사항번호 | **Trigger-04** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 지원자 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogApplicant(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_applicant\_seq;  CREATE OR REPLACE TRIGGER trgLogApplicant  AFTER  INSERT OR UPDATE OR DELETE  ON tblApplicant  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogApplicant VALUES (log\_applicant\_seq.nextval, v\_message, DEFAULT);  END trgLogApplicant; | | | |

| 요구사항번호 | **Trigger-05** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 공지사항 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogNotice(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_notice\_seq;  CREATE OR REPLACE TRIGGER trgLogNotice  AFTER  INSERT OR UPDATE OR DELETE  ON tblNotice  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogNotice VALUES (log\_notice\_seq.nextval, v\_message, DEFAULT);  END trgLogNotice; | | | |

| 요구사항번호 | **Trigger-06** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 과정 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogCurriculum(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_curriculum\_seq;  CREATE OR REPLACE TRIGGER trgLogCurriculum  AFTER  INSERT OR UPDATE OR DELETE  ON tblCurriculum  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogCurriculum VALUES (log\_curriculum\_seq.nextval, v\_message, DEFAULT);  END trgLogCurriculum; | | | |

| 요구사항번호 | **Trigger-07** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 강의실 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogClassroom(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_classroom\_seq;  CREATE OR REPLACE TRIGGER trgLogClassroom  AFTER  INSERT OR UPDATE OR DELETE  ON tblClassroom  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogClassroom VALUES (log\_classroom\_seq.nextval, v\_message, DEFAULT);  END trgLogClassroom; | | | |

| 요구사항번호 | **Trigger-08** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 개설과정 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogOpenCurriculum(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_open\_curriculum\_seq;  CREATE OR REPLACE TRIGGER trgLogOpenCurriculum  AFTER  INSERT OR UPDATE OR DELETE  ON tblOpenCurriculum  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogOpenCurriculum VALUES (log\_open\_curriculum\_seq.nextval, v\_message, DEFAULT);  END trgLogOpenCurriculum; | | | |

| 요구사항번호 | **Trigger-09** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 수업계획서 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogPlan(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_plan\_seq;  CREATE OR REPLACE TRIGGER trgLogPlan  AFTER  INSERT OR UPDATE OR DELETE  ON tblPlan  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogPlan VALUES (log\_plan\_seq.nextval, v\_message, DEFAULT);  END trgLogPlan; | | | |

| 요구사항번호 | **Trigger-10** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 교사 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogProfessor(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_professor\_seq;  CREATE OR REPLACE TRIGGER trgLogProfessor  AFTER  INSERT OR UPDATE OR DELETE  ON tblProfessor  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogProfessor VALUES (log\_professor\_seq.nextval, v\_message, DEFAULT);  END trgLogProfessor; | | | |

| 요구사항번호 | **Trigger-11** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 교사출퇴근내역 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogProfessorAttendance(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_professor\_attendance\_seq;  CREATE OR REPLACE TRIGGER trgLogProfessorAttendance  AFTER  INSERT OR UPDATE OR DELETE  ON tblProfessorAttendance  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogProfessorAttendance VALUES (log\_professor\_attendance\_seq.nextval, v\_message, DEFAULT);  END trgLogProfessorAttendance; | | | |

| 요구사항번호 | **Trigger-12** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 과제 | 작성자 | 이도훈 |
| DML | | | |
| CREATE TABLE tblLogTask (  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT null  );  CREATE SEQUENCE seqLogTask;  create or replace trigger trgTask  after  insert or update or delete  on tblTask  declare  vmessage varchar2(1000);  begin  --dbms\_output.put\_line(to\_char(sysdate, 'hh24:mi:ss') || ' 트리거가 실행되었습니다.');  IF inserting THEN  dbms\_output.put\_line('새로운 항목이 추가되었습니다.');  vmessage := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  dbms\_output.put\_line('기존 항목이 수정되었습니다.');  vmessage := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  dbms\_output.put\_line('기존 항목이 삭제되었습니다.');  vmessage := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogTask VALUES (seqLogTask.nextVal, vmessage, default);  end trgTask; | | | |

| 요구사항번호 | **Trigger-13** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 수강과목 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogCourse(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_course\_seq;  CREATE OR REPLACE TRIGGER trgLogCourse  AFTER  INSERT OR UPDATE OR DELETE  ON tblCourse  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogCourse VALUES (log\_course\_seq.nextval, v\_message, DEFAULT);  END trgLogCourse; | | | |

| 요구사항번호 | **Trigger-14** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 질문내역 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogQuestion(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_question\_seq;  CREATE OR REPLACE TRIGGER trgLogQuestion  AFTER  INSERT OR UPDATE OR DELETE  ON tblQuestion  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogQuestion VALUES (log\_question\_seq.nextval, v\_message, DEFAULT);  END trgLogQuestion; | | | |

| 요구사항번호 | **Trigger-15** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 답변내역 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogAnswer(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_answer\_seq;  CREATE OR REPLACE TRIGGER trgLogAnswer  AFTER  INSERT OR UPDATE OR DELETE  ON tblAnswer  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogQuestion VALUES (log\_answer\_seq.nextval, v\_message, DEFAULT);  END trgLogAnswer; | | | |

| 요구사항번호 | **Trigger-16** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 프로젝트 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogProject(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_project\_seq;  CREATE OR REPLACE TRIGGER trgLogProject  AFTER  INSERT OR UPDATE OR DELETE  ON tblProject  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogQuestion VALUES (log\_project\_seq.nextval, v\_message, DEFAULT);  END trgLogProject; | | | |

| 요구사항번호 | **Trigger-17** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 성적 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogScore(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_score\_seq;  CREATE OR REPLACE TRIGGER trgLogScore  AFTER  INSERT OR UPDATE OR DELETE  ON tblScore  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogScore VALUES (log\_score\_seq.nextval, v\_message, DEFAULT);  END trgLogScore; | | | |

| 요구사항번호 | **Trigger-18** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 시험 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogTest(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_test\_seq;  CREATE OR REPLACE TRIGGER trgLogTest  AFTER  INSERT OR UPDATE OR DELETE  ON tblTest  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogTest VALUES (log\_test\_seq.nextval, v\_message, DEFAULT);  END trgLogTest; | | | |

| 요구사항번호 | **Trigger-19** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 교재 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogBook(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_book\_seq;  CREATE OR REPLACE TRIGGER trgLogBook  AFTER  INSERT OR UPDATE OR DELETE  ON tblBook  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogBook VALUES (log\_book\_seq.nextval, v\_message, DEFAULT);  END trgLogBook; | | | |

| 요구사항번호 | **Trigger-20** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 추천도서 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogBestBook(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_best\_book\_seq;  CREATE OR REPLACE TRIGGER trgLogBestBook  AFTER  INSERT OR UPDATE OR DELETE  ON tblBestBook  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogBestBook VALUES (log\_best\_book\_seq.nextval, v\_message, DEFAULT);  END trgLogBestBook; | | | |

| 요구사항번호 | **Trigger-21** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 과목 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogSubject(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_subject\_seq;  CREATE OR REPLACE TRIGGER trgLogSubject  AFTER  INSERT OR UPDATE OR DELETE  ON tblSubject  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogSubject VALUES (log\_subject\_seq.nextval, v\_message, DEFAULT);  END trgLogSubject; | | | |

| 요구사항번호 | **Trigger-22** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 수강후기 | 작성자 | 이정은 |
| DML | | | |
| CREATE TABLE tblLogreview(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_review\_seq;  CREATE OR REPLACE TRIGGER trgLogreview  AFTER  INSERT OR UPDATE OR DELETE  ON tblCounsel  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogreview VALUES (log\_review\_seq.nextval, v\_message, DEFAULT);  END trgLogreview; | | | |

| 요구사항번호 | **Trigger-23** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 상담 | 작성자 | 이정은 |
| DML | | | |
| CREATE TABLE tblLogCounsel(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_counsel\_seq;  CREATE OR REPLACE TRIGGER trgLogCounsel  AFTER  INSERT OR UPDATE OR DELETE  ON tblCounsel  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogCounsel VALUES (log\_counsel\_seq.nextval, v\_message, DEFAULT);  END trgLogCounsel; | | | |

| 요구사항번호 | **Trigger-24** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 수료내역 | 작성자 | 이정은 |
| DML | | | |
| CREATE TABLE tblLogcomplete(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_complete\_seq;  CREATE OR REPLACE TRIGGER trgLogcomplete  AFTER  INSERT OR UPDATE OR DELETE  ON tblcomplete  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogcomplete VALUES (log\_complete\_seq.nextval, v\_message, DEFAULT);  END trgLogcomplete; | | | |

| 요구사항번호 | **Trigger-25** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 취업 | 작성자 | 이정은 |
| DML | | | |
| CREATE TABLE tblLogEmployment(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_Employment\_seq;  CREATE OR REPLACE TRIGGER trgLogEmployment  AFTER  INSERT OR UPDATE OR DELETE  ON tblEmployment  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogEmployment VALUES (log\_Employment\_seq.nextval, v\_message, DEFAULT);  END trgLogEmployment; | | | |

| 요구사항번호 | **Trigger-26** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 중도탈락내역 | 작성자 | 이정은 |
| DML | | | |
| CREATE TABLE tblLogdrop(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_drop\_seq;  CREATE OR REPLACE TRIGGER trgLogdrop  AFTER  INSERT OR UPDATE OR DELETE  ON tbldrop  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogdrop VALUES (log\_drop\_seq.nextval, v\_message, DEFAULT);  END trgLogdrop; | | | |

| 요구사항번호 | **Trigger-27** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 교육생 | 작성자 | 이정은 |
| DML | | | |
| CREATE TABLE tblLogstudent(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_student\_seq;  CREATE OR REPLACE TRIGGER trgLogstudent  AFTER  INSERT OR UPDATE OR DELETE  ON tblstudent  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogstudent VALUES (log\_student\_seq.nextval, v\_message, DEFAULT);  END trgLogstudent; | | | |

| 요구사항번호 | **Trigger-28** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 수강과정 | 작성자 | 이정은 |
| DML | | | |
| CREATE TABLE tblLogregister(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_register\_seq;  CREATE OR REPLACE TRIGGER trgLogregister  AFTER  INSERT OR UPDATE OR DELETE  ON tblregister  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogregister VALUES (log\_register\_seq.nextval, v\_message, DEFAULT);  END trgLogregister; | | | |

| 요구사항번호 | **Trigger-29** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 출결 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogAttendance(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_attendance\_seq;  CREATE OR REPLACE TRIGGER trgLogAttendance  AFTER  INSERT OR UPDATE OR DELETE  ON tblAttendance  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogAttendance VALUES (log\_attendance\_seq.nextval, v\_message, DEFAULT);  END trgLogAttendance; | | | |

| 요구사항번호 | **Trigger-30** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 개설과목 | 작성자 | 차수민 |
| DML | | | |
| CREATE TABLE tblLogOpenSubject(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_open\_subject\_seq;  CREATE OR REPLACE TRIGGER trgLogOpenSubject  AFTER  INSERT OR UPDATE OR DELETE  ON tblOpenSubject  DECLARE  v\_message varchar2(1000);  BEGIN  IF inserting THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF updating THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF deleting THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogOpenSubject VALUES (log\_open\_subject\_seq.nextval, v\_message, DEFAULT);  END trgLogAOpenSubject; | | | |

| 요구사항번호 | **Trigger-31** | 작성일 | 2023-09-16 |
| --- | --- | --- | --- |
| 요구사항명 | 강의가능과목 | 작성자 | 안대명 |
| DML | | | |
| CREATE TABLE tblLogAbleSubject(  seq NUMBER PRIMARY KEY,  message varchar2(1000) NOT NULL,  regdate DATE DEFAULT sysdate NOT NULL  );  CREATE SEQUENCE log\_ablesubject\_seq;  CREATE OR REPLACE TRIGGER trgLogAbleSubject  AFTER INSERT OR UPDATE OR DELETE  ON tblAbleSubject  FOR EACH ROW  DECLARE  v\_message VARCHAR2(1000);  BEGIN  IF INSERTING THEN  v\_message := '새로운 항목이 추가되었습니다.';  ELSIF UPDATING THEN  v\_message := '기존 항목이 수정되었습니다.';  ELSIF DELETING THEN  v\_message := '기존 항목이 삭제되었습니다.';  END IF;  INSERT INTO tblLogOpenSubject (seq, message, regdate)  VALUES (log\_open\_subject\_seq.nextval, v\_message, SYSDATE);  END trgLogAbleSubject; | | | |