**오브젝트 스크립트(PLSQL)**

**교육생**

2조

한가람, 김원중, 이승진, 이지온, 이지원

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-0.로그인 | 작성일 | 2025-02-07 |
| 요구사항명 | 교육생 로그인 | 작성자 | 김원중 |
| DML | | | |
| -- C-0\_교육생\_로그인  CREATE OR REPLACE PROCEDURE procLoginStudent(  p\_studentId IN NUMBER,  p\_password IN VARCHAR2  ) AS  v\_count NUMBER;  -- 로그인 시 출력할 데이터를 담을 변수들  vstudentRegistrationDate tblStudent.studentRegistrationDate%type;  vstudentName tblStudent.studentName%type;  vstudentEmail tblStudent.studentEmail%type;  vstudentPhoneNumber tblStudent.studentPhoneNumber%type;  vcourseName tblCourse.courseName%type;  vopenedStartDate tblOpenedCourse.openedStartDate%type;  vopenedCompletionDate tblOpenedCourse.openedCompletionDate%type;  vclassroomName tblClassroom.classroomName%type;  BEGIN  -- 해당 교육생 존재하는지 확인  SELECT COUNT(\*) INTO v\_count FROM tblStudent WHERE studentId = p\_studentId AND studentResidentNumberBack = p\_password;  -- 로그인 결과 처리  IF v\_count > 0 THEN  DBMS\_OUTPUT.PUT\_LINE('로그인 성공');  DBMS\_OUTPUT.PUT\_LINE(' ');  DBMS\_OUTPUT.PUT\_LINE(' ');  select  tblStudent.studentRegistrationDate,  tblStudent.studentName,  tblStudent.studentEmail,  tblStudent.studentPhoneNumber,  tblCourse.courseName,  tblOpenedCourse.openedStartDate,  tblOpenedCourse.openedCompletionDate,  tblClassroom.classroomName  into  vstudentRegistrationDate, vstudentName, vstudentEmail, vstudentPhoneNumber, vcourseName, vopenedStartDate, vopenedCompletionDate, vclassroomName  from tblStudent  join tblOpenedCourse on tblStudent.studentCourseId = tblOpenedCourse.openedCourseId  join tblCourse on tblCourse.courseId = tblOpenedCourse.courseId  join tblClassroom on tblClassroom.classroomId = tblOpenedCourse.classroomId  where tblStudent.studentId = p\_studentId;  DBMS\_OUTPUT.PUT\_LINE('----------------------------------------------------------------------');  DBMS\_OUTPUT.PUT\_LINE(' ' || vcourseName);  DBMS\_OUTPUT.PUT\_LINE('----------------------------------------------------------------------');  DBMS\_OUTPUT.PUT\_LINE('강의실 과정시작일 과정종료일');  DBMS\_OUTPUT.PUT\_LINE(vclassroomName ||  ' ' ||  TO\_CHAR(vopenedStartDate, 'YYYY/MM/DD') ||  ' ' ||  TO\_CHAR(vopenedCompletionDate, 'YYYY/MM/DD'));  DBMS\_OUTPUT.PUT\_LINE('----------------------------------------------------------------------');  DBMS\_OUTPUT.PUT\_LINE('교육생 전화번호 이메일');  DBMS\_OUTPUT.PUT\_LINE(vstudentName ||  ' ' ||  vstudentPhoneNumber ||  ' ' ||  vstudentEmail);  DBMS\_OUTPUT.PUT\_LINE('');  DBMS\_OUTPUT.PUT\_LINE('등록일 ' || TO\_CHAR(vstudentRegistrationDate, 'YYYY/MM/DD'));  ELSE  DBMS\_OUTPUT.PUT\_LINE('로그인 실패');  END IF;  END procLoginStudent;  /  BEGIN -- C-0\_교육생\_로그인\_호출  procLoginStudent(300, '4028467');  END;  / | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-1.성적 조회 | 작성일 | 2025-02-07 |
| 요구사항명 | 교육생 성적 조회 | 작성자 | 김원중 |
| DML | | | |
| -- C-1\_교육생\_성적\_조회  CREATE OR REPLACE PROCEDURE procGetStudentScore(  p\_studentId IN NUMBER  )  AS  vwrittenExamQuestions tblWrittenExam.writtenExamQuestions%type;  vpracticalExamQuestions tblPracticalExam.practicalExamQuestions%type;  BEGIN  FOR rec IN (  SELECT  tblSubject.subjectId,  tblSubject.subjectName,  tblOpenedSubject.subjectStartDate,  tblOpenedSubject.subjectEndDate,  tblTextbook.textbookName,  tblTeacher.teacherName,  tblOpenedSubject.subjectWrittenScoreWeight,  tblOpenedSubject.subjectPracticalScoreWeight,  tblOpenedSubject.subjectAttendanceScoreWeight,  tblGrade.gradeWrittenScore,  tblGrade.gradePracticalScore,  tblGrade.gradeAttendanceScore,  tblWrittenExam.writtenExamDate,  tblWrittenExam.writtenExamQuestions,  tblPracticalExam.practicalExamDate,  tblPracticalExam.practicalExamQuestions  FROM tblStudent  JOIN tblOpenedCourse ON tblStudent.studentCourseId = tblOpenedCourse.openedCourseId  JOIN tblOpenedSubject ON tblOpenedCourse.openedCourseId = tblOpenedSubject.openedCourseId  JOIN tblTeacher ON tblOpenedCourse.teacherId = tblTeacher.teacherId  JOIN tblSubject ON tblSubject.subjectId = tblOpenedSubject.subjectId  JOIN tblTextbook ON tblSubject.subjectId = tblTextbook.subjectId  JOIN tblWrittenExam ON tblWrittenExam.openedSubjectId = tblOpenedSubject.openedSubjectId  JOIN tblPracticalExam ON tblPracticalExam.openedSubjectId = tblOpenedSubject.openedSubjectId  JOIN tblGrade ON tblGrade.studentId = tblStudent.studentId and tblGrade.openedSubjectId = tblOpenedSubject.openedSubjectId  WHERE tblStudent.studentId = p\_studentId  ORDER BY tblOpenedSubject.subjectid  ) LOOP  IF rec.writtenExamDate < SYSDATE THEN  vwrittenExamQuestions := rec.writtenExamQuestions;  ELSE  vwrittenExamQuestions := '미시행';  END IF;  IF rec.practicalExamDate < SYSDATE THEN  vpracticalExamQuestions := rec.practicalExamQuestions;  ELSE  vpracticalExamQuestions := '미시행';  END IF;  DBMS\_OUTPUT.PUT\_LINE('');  DBMS\_OUTPUT.PUT\_LINE('===============================================================================');  -- DBMS\_OUTPUT.PUT\_LINE('-------------------------------------------------------------------------------');  DBMS\_OUTPUT.PUT\_LINE(RPAD(rec.subjectId, 2,' ') || '| ' || RPAD(rec.subjectName,45,' ') || ' |' ||  TO\_CHAR(rec.subjectStartDate, 'YYYY/MM/DD') || ' ~ ' ||  TO\_CHAR(rec.subjectEndDate, 'YYYY/MM/DD')  );  DBMS\_OUTPUT.PUT\_LINE('-------------------------------------------------------------------------------');  DBMS\_OUTPUT.PUT\_LINE('강사: ' || rec.teacherName || ' ');  DBMS\_OUTPUT.PUT\_LINE('교재: ' || rec.textbookName);  DBMS\_OUTPUT.PUT\_LINE('-------------------------------------------------------------------------------');  DBMS\_OUTPUT.PUT\_LINE(' 필기 실기 출결');  DBMS\_OUTPUT.PUT\_LINE('반영 비율 ' ||  rec.subjectWrittenScoreWeight || '%' || ' ' ||  rec.subjectPracticalScoreWeight || '%' || ' ' ||  rec.subjectAttendanceScoreWeight || '%'  );  DBMS\_OUTPUT.PUT\_LINE('성적 ' ||  RPAD(NVL(TO\_CHAR(rec.gradeWrittenScore), '없음'), 4, ' ') || ' ' ||  RPAD(NVL(TO\_CHAR(rec.gradePracticalScore), '없음'), 4, ' ') || ' ' ||  RPAD(NVL(TO\_CHAR(rec.gradeAttendanceScore), '없음'), 4, ' ')  );  DBMS\_OUTPUT.PUT\_LINE('-------------------------------------------------------------------------------');  DBMS\_OUTPUT.PUT\_LINE(' 시험일 문제' );  DBMS\_OUTPUT.PUT\_LINE('필기 ' ||  TO\_CHAR(rec.writtenExamDate, 'YYYY/MM/DD') || ' ' ||  vwrittenExamQuestions);  DBMS\_OUTPUT.PUT\_LINE('실기 ' ||  TO\_CHAR(rec.practicalExamDate, 'YYYY/MM/DD') || ' ' ||  vpracticalExamQuestions);  DBMS\_OUTPUT.PUT\_LINE('===============================================================================');  END LOOP;  END procGetStudentScore;  /  BEGIN -- C-1\_교육생\_성적\_조회\_호출  procGetStudentScore(256);  END;  / | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-2.출결 관리 및 출결 조회 | 작성일 | 2025-02-07 |
| 요구사항명 | 교육생 출결 관리 및 출결 조회 | 작성자 | 김원중 |
| DML | | | |
| -- C-2-1\_교육생\_출근\_기록  CREATE OR REPLACE PROCEDURE procStudentCheckin(  p\_studentId IN NUMBER  )  AS  vcnt NUMBER;  BEGIN  SELECT COUNT(\*) INTO vcnt FROM tblAttendance  WHERE TO\_CHAR(attendanceDate, 'YYYY/MM/DD') = TO\_CHAR(SYSDATE, 'YYYY/MM/DD') AND  studentId = p\_studentId;  IF vcnt = 0 THEN  INSERT INTO tblAttendance VALUES  ((SELECT MAX(attendanceId) FROM tblAttendance) + 1, TO\_CHAR(SYSDATE, 'YYYY/MM/DD'), SYSDATE, NULL, NULL, p\_studentId);  -- 출력  DBMS\_OUTPUT.PUT\_LINE('입실 완료');  DBMS\_OUTPUT.PUT\_LINE('현재 시간 : ' || SYSDATE);  ELSE  DBMS\_OUTPUT.PUT\_LINE('오늘은 이미 입실 등록을 하였습니다.');  END IF;  END procStudentCheckin;  /  BEGIN -- C-2-1\_교육생\_출근\_기록\_호출  procStudentCheckin(300);  END;  /  -- C-2-1\_교육생\_출결상태\_트리거  CREATE OR REPLACE TRIGGER trgAutoAttendanceStatus  BEFORE INSERT  ON tblAttendance  FOR EACH ROW  BEGIN  IF :NEW.attendanceCheckInTime < TRUNC(:NEW.attendanceCheckInTime) + INTERVAL '9' HOUR + INTERVAL '10' MINUTE THEN  :NEW.attendanceStatus := '정상';  ELSE  :NEW.attendanceStatus := '지각';  END IF;  END trgAutoAttendanceStatus;  /  ALTER TRIGGER trgAutoAttendanceStatus ENABLE;  ALTER TRIGGER trgAutoAttendanceStatus DISABLE;  -- C-2-1\_교육생\_퇴근\_기록  CREATE OR REPLACE PROCEDURE procStudentCheckout(  p\_studentId IN NUMBER  )  AS  vcnt NUMBER;  BEGIN  SELECT COUNT(\*) INTO vcnt FROM tblAttendance  WHERE TO\_CHAR(attendanceDate, 'YYYY/MM/DD') = TO\_CHAR(SYSDATE, 'YYYY/MM/DD') AND  studentId = p\_studentId;  IF vcnt = 1 THEN  SELECT COUNT(attendanceCheckOutTime) INTO vcnt FROM tblAttendance  WHERE TO\_CHAR(attendanceDate, 'YYYY/MM/DD') = TO\_CHAR(SYSDATE, 'YYYY/MM/DD') AND  studentId = p\_studentId;  IF vcnt = 0 THEN  UPDATE tblAttendance SET attendanceCheckOutTime = SYSDATE  WHERE TO\_CHAR(attendanceDate, 'YYYY/MM/DD') = TO\_CHAR(SYSDATE, 'YYYY/MM/DD') AND studentId = p\_studentId;  -- 출력  DBMS\_OUTPUT.PUT\_LINE('퇴실 완료');  DBMS\_OUTPUT.PUT\_LINE('현재 시간 : ' || SYSDATE);  ELSE  DBMS\_OUTPUT.PUT\_LINE('오늘은 이미 퇴실 등록을 하였습니다.');  END IF;  ELSE  DBMS\_OUTPUT.PUT\_LINE('오늘의 입실 기록이 없습니다.');  DBMS\_OUTPUT.PUT\_LINE('매니저님에게 문의하세요.');  END IF;  END procStudentCheckout;  /  BEGIN -- C-2-1\_교육생\_퇴근\_기록\_호출  procStudentCheckout(300);  END;  /  -- C-2-2\_교육생\_출결\_조회(일)  CREATE OR REPLACE PROCEDURE procViewStudentAttendanceDay(  p\_studentId IN NUMBER  )  AS  vattendanceDate VARCHAR2(30);  vattendanceCheckInTime VARCHAR2(30);  vattendanceCheckOutTime VARCHAR2(30);  vattendanceStatus tblAttendance.attendanceStatus%type;  BEGIN  SELECT  TO\_CHAR(attendanceDate, 'YYYY/MM/DD'),  TO\_CHAR(attendanceCheckInTime, 'HH24:MI:SS'),  TO\_CHAR(attendanceCheckOutTime, 'HH24:MI:SS'),  attendanceStatus  INTO  vattendanceDate, vattendanceCheckInTime, vattendanceCheckOutTime, vattendanceStatus  FROM tblAttendance WHERE studentId = p\_studentId AND TO\_CHAR(attendanceDate, 'YY/MM/DD') = TO\_CHAR(SYSDATE, 'YY/MM/DD');  DBMS\_OUTPUT.PUT\_LINE('날짜 ' || '입실 시간 ' || '퇴실 시간 ' || '출결 상태');  DBMS\_OUTPUT.PUT\_LINE(vattendanceDate || ' ' ||  vattendanceCheckInTime || ' ' ||  vattendanceCheckOutTime || ' ' ||  vattendanceStatus );  END procViewStudentAttendanceDay;  /  BEGIN -- C-2-2\_교육생\_출결\_조회(일)\_호출  procViewStudentAttendanceDay(300);  END;  /  -- C-2-2\_교육생\_출결\_조회(월)  CREATE OR REPLACE PROCEDURE procViewStudentAttendanceMonth(  p\_studentId IN NUMBER,  p\_month IN VARCHAR2  )  AS  BEGIN  DBMS\_OUTPUT.PUT\_LINE('날짜 ' || '입실 시간 ' || '퇴실 시간 ' || '출결 상태');  FOR rec IN (  SELECT  TO\_CHAR(attendanceDate, 'YYYY/MM/DD') AS day,  TO\_CHAR(attendanceCheckInTime, 'HH24:MI:SS') AS CheckinTime,  TO\_CHAR(attendanceCheckOutTime, 'HH24:MI:SS') AS CheckouTime,  attendanceStatus  FROM tblAttendance WHERE studentId = p\_studentId AND TO\_CHAR(attendanceDate, 'YYYY/MM') = TRIM(p\_month)  ORDER BY day ASC  ) LOOP  IF rec.CheckinTime IS NULL THEN rec.CheckinTime := ' ';  END IF;  IF rec.CheckouTime IS NULL THEN rec.CheckouTime := ' ';  END IF;  DBMS\_OUTPUT.PUT\_LINE(rec.day || ' ' ||  rec.CheckinTime || ' ' ||  rec.CheckouTime || ' ' ||  rec.attendanceStatus );  END LOOP;  END procViewStudentAttendanceMonth;  /  BEGIN -- C-2-2\_교육생\_출결\_조회(월)\_호출  procViewStudentAttendanceMonth(300, '2025/01');  END;  /  -- C-2-2\_교육생\_출결\_조회(전체)  CREATE OR REPLACE PROCEDURE procViewStudentAttendanceAll(  p\_studentId IN NUMBER  )  AS  BEGIN  DBMS\_OUTPUT.PUT\_LINE('날짜 ' || '입실 시간 ' || '퇴실 시간 ' || '출결 상태');  FOR rec IN (  SELECT  TO\_CHAR(attendanceDate, 'YYYY/MM/DD') AS day,  TO\_CHAR(attendanceCheckInTime, 'HH24:MI:SS') AS CheckinTime,  TO\_CHAR(attendanceCheckOutTime, 'HH24:MI:SS') AS CheckouTime,  attendanceStatus  FROM tblAttendance WHERE studentId = p\_studentId  ORDER BY day ASC  ) LOOP  IF rec.CheckinTime IS NULL THEN rec.CheckinTime := ' ';  END IF;  IF rec.CheckouTime IS NULL THEN rec.CheckouTime := ' ';  END IF;  DBMS\_OUTPUT.PUT\_LINE(rec.day || ' ' ||  rec.CheckinTime || ' ' ||  rec.CheckouTime || ' ' ||  rec.attendanceStatus );  END LOOP;  END procViewStudentAttendanceAll;  /  BEGIN -- C-2-2\_교육생\_출결\_조회(전체)\_호출  procViewStudentAttendanceAll(100);  END;  / | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-3.교사 경력사항 조회 | 작성일 | 2025-02-07 |
| 요구사항명 | 교육생 교사 경력사항 조회 | 작성자 | 김원중 |
| DML | | | |
| -- C-3\_교육생\_교사\_경력사항\_조회  CREATE OR REPLACE PROCEDURE procViewTeacherExperience(  p\_teacherId IN NUMBER  )  AS  vteacherName tblTeacher.teacherName%type;  BEGIN  SELECT  teacherName INTO vteacherName  FROM tblTeacher  WHERE teacherId = p\_teacherId;  DBMS\_OUTPUT.PUT\_LINE('강사명 :' || vteacherName);  DBMS\_OUTPUT.PUT\_LINE('--------------------------------------------------------------');  FOR rec IN (  SELECT  teacherExperienceCertification  FROM tblTeacherExperience  JOIN tblTeacher ON tblTeacher.teacherId = tblTeacherExperience.teacherId  WHERE tblTeacherExperience.teacherId = p\_teacherId  ) LOOP  -- 출력  DBMS\_OUTPUT.PUT\_LINE('경력사항 :' || rec.teacherExperienceCertification);  END LOOP;  DBMS\_OUTPUT.PUT\_LINE('--------------------------------------------------------------');  END procViewTeacherExperience;  /  BEGIN -- C-3\_교육생\_교사\_경력사항\_조회\_호출  procViewTeacherExperience(6);  END;  / | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-4.자율게시판 이용 | 작성일 | 2025-02-07 |
| 요구사항명 | 교육생 자율게시판 이용 | 작성자 | 김원중 |
| DML | | | |
| -- C-4\_교육생\_자유게시판\_작성  CREATE OR REPLACE PROCEDURE procFreeBoardCreate( -- 제목, 내용, 학생번호 변수  p\_freeBoardTitle VARCHAR2,  p\_freeBoardContent VARCHAR2,  p\_studentId NUMBER  )  AS  BEGIN  INSERT INTO tblFreeBoard VALUES  ((SELECT MAX(freeBoardId) FROM tblFreeBoard) + 1 ,  p\_freeBoardTitle,  p\_freeBoardContent,  SYSDATE,  p\_studentId);  END procFreeBoardCreate;  /  BEGIN -- C-4\_교육생\_자유게시판\_작성\_호출  procFreeBoardCreate('제목', '내용', 1);  END;  /  -- C-4\_교육생\_자유게시판\_조회(목록)  CREATE OR REPLACE PROCEDURE procFreeBoardReview(  ppage NUMBER  )  AS  BEGIN  DBMS\_OUTPUT.PUT\_LINE('글번호 |' || '제목 |' || '작성시간 |' || '작성자|');  DBMS\_OUTPUT.PUT\_LINE('------------------------------------------------------------------------');  FOR rec IN (  SELECT \* FROM (  SELECT 글번호, 제목, 작성시간, 작성자, ROWNUM as rnum FROM  (SELECT  tblFreeBoard.freeBoardId AS 글번호,  tblFreeBoard.freeBoardTitle AS 제목,  tblFreeBoard.freeBoardCreationDate AS 작성시간,  tblStudent.studentName AS 작성자  FROM tblFreeBoard  JOIN tblStudent ON tblStudent.studentId = tblFreeBoard.studentId  ORDER BY tblFreeBoard.freeBoardId desc))  WHERE rnum BETWEEN 1 + ((ppage - 1) \* 20) AND 20 + ((ppage - 1) \* 20)  ) LOOP  -- 출력  DBMS\_OUTPUT.PUT\_LINE(rec.글번호 || ' ' || '|' ||  RPAD(rec.제목, 30, ' ') || ' ' || '|' ||  rec.작성시간 || ' ' || '|' ||  rec.작성자 || '|' );  END LOOP;  END procFreeBoardReview;  /  BEGIN -- C-4\_교육생\_자유게시판\_조회(목록)\_호출  procFreeBoardReview(1);  END;  /  -- C-4\_교육생\_자유게시판\_조회(글)  CREATE OR REPLACE PROCEDURE procFreeBoardReviewContext(  pfreeBoardId NUMBER  )  AS  vrow tblFreeBoard%ROWTYPE;  vname VARCHAR2(100);  vcnt NUMBER := 0;  vlength NUMBER;  BEGIN  SELECT \* INTO vrow FROM tblFreeBoard WHERE freeBoardId = pfreeBoardId;  SELECT studentName INTO vname FROM tblStudent WHERE studentId = vrow.studentId;  DBMS\_OUTPUT.PUT\_LINE(vrow.freeBoardTitle);  DBMS\_OUTPUT.PUT\_LINE('------------------------------------------------------------------------------------------------');  DBMS\_OUTPUT.PUT\_LINE(vname || ' ' || vrow.freeBoardCreationDate);  DBMS\_OUTPUT.PUT\_LINE('------------------------------------------------------------------------------------------------');  vlength := LENGTH(vrow.freeBoardContent);    LOOP  DBMS\_OUTPUT.PUT\_LINE(SUBSTR(vrow.freeBoardContent, 1 + (50 \* vcnt), 50));  vcnt := vcnt + 1;  EXIT WHEN (vcnt \* 50) >= vlength;  END LOOP;    END procFreeBoardReviewContext;  /  BEGIN -- C-4\_교육생\_자유게시판\_조회(글)\_호출  procFreeBoardReviewContext(70);  END;  /  -- C-4\_교육생\_자유게시판\_수정  CREATE OR REPLACE PROCEDURE procFreeBoardUpdate( -- 글번호, 제목, 내용  p\_freeBoardId NUMBER,  p\_freeBoardTitle VARCHAR2,  p\_freeBoardContent VARCHAR2  )  AS  BEGIN  UPDATE tblFreeBoard  SET freeBoardTitle = p\_freeBoardTitle,  freeBoardContent = p\_freeBoardContent  WHERE freeBoardId = p\_freeBoardId;  END procFreeBoardUpdate;  /  BEGIN -- C-4\_교육생\_자유게시판\_수정\_호출  procViewStudentAttendanceAll('수정된 제목', '수정된 내용', 101);  END;  /  -- C-4\_교육생\_자유게시판\_삭제  CREATE OR REPLACE PROCEDURE procFreeBoardDelete( -- 글번호 변수  p\_freeBoardId NUMBER  )  AS  BEGIN  DELETE FROM tblFreeBoard WHERE freeBoardId = p\_freeBoardId;  END procFreeBoardDelete;  /  BEGIN -- C-4\_교육생\_자유게시판\_삭제\_호출  procFreeBoardDelete(101);  END;  / | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-5. 국민취업제도 | 작성일 | 2025-02-07 |
| 요구사항명 | 교육생 국민취업제도 | 작성자 | 김원중 |
| DML | | | |
| -- C-5\_교육생\_국민취업제도\_등록  CREATE OR REPLACE PROCEDURE procNationalRetrievalInsert( -- 학생 번호, 참여여부, 지원유형, 날짜, 승인상태 변수  pstudentId NUMBER,  pnationalRetrievalParticipation VARCHAR2,  pnationalRetrievalSupport NUMBER,  pntnlRtrvlplctnDt DATE,  pntnlRtrvlprvlSts VARCHAR2  )  IS  BEGIN  INSERT INTO tblNationalRetrieval VALUES(  pstudentId,  pnationalRetrievalParticipation,  pnationalRetrievalSupport,  pntnlRtrvlplctnDt,  pntnlRtrvlprvlSts);  END procNationalRetrievalInsert;  /  BEGIN -- C-5\_교육생\_국민취업제도\_등록\_호출  procNationalRetrievalInsert(131, 'X', NULL, NULL, NULL);  END;  /  -- C-5\_교육생\_국민취업제도\_수정  CREATE OR REPLACE PROCEDURE procNationalRetrievalUpdate( -- 학생 번호, 참여여부, 지원유형, 날짜, 승인상태 변수  pstudentId NUMBER,  pnationalRetrievalParticipation VARCHAR2,  pnationalRetrievalSupport NUMBER,  pntnlRtrvlplctnDt DATE,  pntnlRtrvlprvlSts VARCHAR2  )  IS  BEGIN  UPDATE tblNationalRetrieval SET  nationalRetrievalParticipation = pnationalRetrievalParticipation,  nationalRetrievalSupport = pnationalRetrievalSupport,  ntnlRtrvlplctnDt = pntnlRtrvlplctnDt,  ntnlRtrvlprvlSts = pntnlRtrvlprvlSts  WHERE studentId = pstudentId;  END procNationalRetrievalUpdate;  /  BEGIN -- C-5\_교육생\_국민취업제도\_수정\_호출  procNationalRetrievalUpdate(131, 'O', 1, '2025/02/06', '승인완료');  END;  / | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항번호 | C-6.사후처리 입력 | 작성일 | 2025-02-07 |
| 요구사항명 | 교육생 사후처리 입력 | 작성자 | 김원중 |
| DML | | | |
| -- C-6\_교육생\_사후처리\_취업활동등록  CREATE OR REPLACE PROCEDURE procStudentHistoryInsert(  pstudentId NUMBER,  pdate DATE  )  IS  vcnt NUMBER;  BEGIN  -- 수료완료일 때 가능  SELECT COUNT(\*) INTO vcnt FROM tblStudent  WHERE studentCompletionStatus = '수료완료' AND  studentId = pstudentId;  IF vcnt = 1 THEN  UPDATE tblStudentHistory SET historyInsuranceEnrollmentDate = pdate  WHERE studentId = pstudentId;  ELSE  DBMS\_OUTPUT.PUT\_LINE('수료완료인 교육생만 작성 가능');  END IF;  END procStudentHistoryInsert;  /  BEGIN -- C-6\_교육생\_사후처리\_취업활동등록\_호출  procStudentHistoryInsert(1, '2025/02/01');  END;  /  -- C-6\_교육생\_사후처리\_취업활동조회  CREATE OR REPLACE PROCEDURE procStudentHistoryReview( -- 학생번호  pstudentId NUMBER  )  IS  vstudentName tblStudent.studentName%type;  vDate VARCHAR2(30);  BEGIN  SELECT  TO\_CHAR(historyInsuranceEnrollmentDate, 'YYYY/MM/DD'), studentName  INTO vDate, vstudentName  FROM tblStudentHistory  JOIN tblStudent ON tblStudent.studentId = tblStudentHistory.studentId  WHERE tblStudentHistory.studentId = pstudentId;  DBMS\_OUTPUT.PUT\_LINE(vstudentName || ' 교육생');  DBMS\_OUTPUT.PUT\_LINE('입사일 : ' || vDate);  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('기록이 없습니다.');  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('기록이 없습니다.');  END procStudentHistoryReview;  /  BEGIN -- C-6\_교육생\_사후처리\_취업활동조회\_호출  procStudentHistoryReview(1); -- 학생번호  END;  /  -- C-6\_교육생\_사후처리\_취업활동수정  CREATE OR REPLACE PROCEDURE procStudentHistoryUpdate(  pstudentId NUMBER,  pdate DATE  )  IS  vcnt NUMBER;  BEGIN  -- 제약사항-과정을 수강 중이면 기록할 수 없다.  -- 수료완료일 때 가능  SELECT COUNT(\*) INTO vcnt FROM tblStudent  WHERE studentCompletionStatus = '수료완료' AND  studentId = pstudentId;  IF vcnt = 1 THEN  UPDATE tblStudentHistory SET hstrnsrncTrmntnDt = pdate  WHERE studentId = pstudentId;  ELSE  DBMS\_OUTPUT.PUT\_LINE('수료완료인 교육생만 작성 가능');  END IF;  END procStudentHistoryUpdate;  /  BEGIN -- C-6\_교육생\_사후처리\_취업활동수정\_호출  procStudentHistoryInsert(1, '2025/02/01');  END;  / | | | |