

DB Object 스크립트(PL/SQL)

2조 박진혁, 문예빈, 최석용, 홍선민

작성일자: 25.08.06 - 25.08.07

목 차

A. 프로시저.....	3
1. 관리자 기능.....	3
2. 교사 기능.....	27
3. 교육생 기능.....	47
B. 뷰.....	60
1. 관리자 기능.....	60
2. 교사 기능.....	71
3. 교육생 기능.....	77

A. 프로시저

1. 관리자 기능

요구사항번호	B-3(관리자)	작성일	2025-08-06
요구사항명	특정 교사 정보 출력	작성자	문예빈
Procedure			
<pre>/* 특정 교사 선택 시 정보 출력*/ CREATE OR REPLACE PROCEDURE procTeacherSubjects(p_teacherNum IN NUMBER) IS BEGIN FOR rec IN (SELECT s.subjectName AS subject_name, TO_CHAR(os.openSubjectStartDay, 'YYYY-MM-DD') ' ~ ' TO_CHAR(os.openSubjectEndDay, 'YYYY-MM-DD') AS subject_period, c.curriculumName AS curriculum_name, TO_CHAR(oc.openCurriStartDate, 'YYYY-MM-DD') ' ~ ' TO_CHAR(oc.openCurriEndDate, 'YYYY-MM-DD') AS curriculum_period, LISTAGG(b.bookTitle, ', ') WITHIN GROUP (ORDER BY b.bookTitle) AS book_titles, oc.classroomNum '호실' AS classroom, CASE WHEN os.openSubjectEndDay < SYSDATE THEN '강의종료' WHEN os.openSubjectStartDay > SYSDATE THEN '강의예정' ELSE '강의중' END AS lecture_status,</pre>			

```

os.openSubjectStartDay AS start_sort -- 정렬용
FROM tblOpenSubject os
    INNER JOIN tblSubject s ON os.subjectNum = s.subjectNum
    INNER JOIN tblOpenSOC soc ON os.openSubjectNum = soc.openSubjectNum
    INNER JOIN tblOpenCurriculum oc ON soc.openCurriculumNum = oc.openCurriculumNum
    INNER JOIN tblCurriculum c ON oc.curriculumNum = c.curriculumNum
    INNER JOIN tblBook b ON os.openSubjectNum = b.openSubjectNum
WHERE oc.teacherNum = p_teacherNum
GROUP BY
    s.subjectName,
    os.openSubjectStartDay,
    os.openSubjectEndDay,
    c.curriculumName,
    oc.openCurriStartDate,
    oc.openCurriEndDate,
    oc.classroomNum,
    os.openSubjectEndDay,
    os.openSubjectStartDay
ORDER BY os.openSubjectStartDay
) LOOP
    DBMS_OUTPUT.PUT_LINE('과목명: ' || rec.subject_name);
    DBMS_OUTPUT.PUT_LINE('과목 기간: ' || rec.subject_period);
    DBMS_OUTPUT.PUT_LINE('과정명: ' || rec.curriculum_name);
    DBMS_OUTPUT.PUT_LINE('과정 기간: ' || rec.curriculum_period);
    DBMS_OUTPUT.PUT_LINE('교재명: ' || rec.book_titles);

```

```

DBMS_OUTPUT.PUT_LINE('강의실: ' || rec.classroom);
DBMS_OUTPUT.PUT_LINE('강의 진행 여부: ' || rec.lecture_status);
DBMS_OUTPUT.PUT_LINE('-----');
END LOOP;
END;

BEGIN
    procTeacherSubjects(4); -- 교사 번호
END;

```

```

/*특정 과목 입력 시 강의 가능한 교사들의 목록 출력*/
CREATE OR REPLACE PROCEDURE procTeacherWhoCan (
    p_subjectName IN VARCHAR2
)
IS
    v_teacherName tblTeacher.teacherName%TYPE;
    v_subjectName tblSubject.subjectName%TYPE;

    CURSOR v_cursor IS
        SELECT * FROM (
            SELECT DISTINCT t.teacherName AS teacherName, s.subjectName AS subjectName
            FROM tblPossibleSubject ps,
                 tblTeacher t,
                 tblSubject s
            WHERE ps.teacherNum = t.teacherNum
            AND ps.subjectNum = s.subjectNum
        );

```

```
        AND s.subjectName = p_subjectName
    );

BEGIN
    OPEN v_cursor;
LOOP
    FETCH v_cursor INTO v_teacherName, v_subjectName;
    EXIT WHEN v_cursor%NOTFOUND;

    DBMS_OUTPUT.PUT_LINE('강사명: ' || v_teacherName || ' | 과목명: ' || v_subjectName);
    DBMS_OUTPUT.PUT_LINE('-----');
END LOOP;
CLOSE v_cursor;
END;

BEGIN
    procTeacherWhoCan('Java'); -- 과목 이름
END;
```

요구사항번호	B-4(관리자)	작성일	2025-08-05
요구사항명	개설 과정 관리	작성자	문예빈
Procedure			
<pre> /* 특정 개설 과정이 수료한 경우 등록된 교육생 전체에 대해서 수료상태를 지정(중도탈락자 제외) */ CREATE OR REPLACE PROCEDURE procCompleteStudents (p_openCurriculumNum IN NUMBER) IS v_updated_rows NUMBER := 0; BEGIN UPDATE tblRegisterStudent SET completionCategoryNum = 3 WHERE openCurriculumNum = p_openCurriculumNum AND completionCategoryNum = 1; v_updated_rows := SQL%ROWCOUNT; IF v_updated_rows = 0 THEN DBMS_OUTPUT.PUT_LINE('⚠️ 수료 상태로 변경된 교육생이 없습니다.'); ELSE DBMS_OUTPUT.PUT_LINE('✅ 수료 상태로 변경된 교육생 수: ' v_updated_rows '명'); END IF; EXCEPTION </pre>			

```

WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('⚠ 오류 발생: ' || SQLERRM);
END;

BEGIN
    procCompleteStudents(1); -- 개설과정 번호
END;

/* 개설 교육 과정 번호를 입력하여 개설 과목 정보 및 교육생 정보 확인 */
CREATE OR REPLACE PROCEDURE procOpenCurriculum (p_openCurriculumNum IN NUMBER) IS
BEGIN
    FOR poc IN (
        SELECT
            s.subjectName AS subject_name,
            to_char(os.openSubjectStartDay, 'YYYY-MM-DD') || ' ~ ' ||
                to_char(os.openSubjectEndDay, 'YYYY-MM-DD') AS subject_period,
            listagg(b.bookTitle, ', ') WITHIN GROUP (ORDER BY b.bookTitle) AS book_title,
            t.teacherName AS teacher_name,
            st.studentName AS student_name,
            st.sResidentNum AS s_residentNum,
            st.sPhoneNum AS s_phoneNum,
            st.dateOfRegistration AS date_of_registration,
            cc.completionCategoryName AS completion_categoryName
        FROM tblOpenCurriculum oc
        INNER JOIN tblOpenSOC osoc ON oc.openCurriculumNum = osoc.OPENCURRICULUMNUM
        INNER JOIN tblOpenSubject os ON osoc.openSubjectNum = os.openSubjectNum
    )
    LOOP
        dbms_output.put_line(poc.subject_name || ' (' || poc.subject_period || ')');
        dbms_output.put_line('교과서 목록: ' || poc.book_title);
        dbms_output.put_line('교사 이름: ' || poc.teacher_name);
        dbms_output.put_line('학생 이름: ' || poc.student_name);
        dbms_output.put_line('주민번호: ' || poc.s_residentNum);
        dbms_output.put_line('전화번호: ' || poc.s_phoneNum);
        dbms_output.put_line('등록일: ' || poc.date_of_registration);
        dbms_output.put_line('성과 분류: ' || poc.completion_categoryName);
    END LOOP;
END;

```

```

        INNER JOIN tblSubject s ON os.subjectNum = s.subjectNum
        INNER JOIN tblBook b ON os.openSubjectNum = b.openSubjectNum
        INNER JOIN tblTeacher t ON oc.teacherNum = t.teacherNum
        INNER JOIN tblRegisterStudent rs ON oc.openCurriculumNum = rs.openCurriculumNum
        INNER JOIN tblStudent st ON rs.studentNum = st.studentNum
        INNER JOIN tblCompletionCategory cc ON rs.completionCategoryNum = cc.completionCategoryNum
    WHERE oc.openCurriculumNum = 1
    GROUP BY
        s.subjectName,
        os.openSubjectStartDay,
        os.openSubjectEndDay,
        t.teacherName,
        st.studentName,
        st.sResidentNum,
        st.sPhoneNum,
        st.dateOfRegistration,
        cc.completionCategoryName
    ORDER BY st.studentName
) LOOP
DBMS_OUTPUT.PUT_LINE('과목명: ' || poc.subject_name);
DBMS_OUTPUT.PUT_LINE('과목 기간: ' || poc.subject_period);
DBMS_OUTPUT.PUT_LINE('교재명: ' || poc.book_title);
DBMS_OUTPUT.PUT_LINE('교사명: ' || poc.teacher_name);
DBMS_OUTPUT.PUT_LINE('이름: ' || poc.student_name);
DBMS_OUTPUT.PUT_LINE('주민번호뒷자리: ' || poc.s_residentNum);

```

```

        DBMS_OUTPUT.PUT_LINE('전화번호: ' || poc.s_phoneNum);
        DBMS_OUTPUT.PUT_LINE('등록일: ' || poc.date_of_registration);
        DBMS_OUTPUT.PUT_LINE('수료상태: ' || poc.completion_categoryName);
        DBMS_OUTPUT.PUT_LINE('-----');
    END LOOP;
END;

BEGIN
    procOpenCurriculum(1);      -- 개설과정 번호
END;

```

요구사항번호	B-5(관리자)	작성일	2025-08-06
요구사항명	특정 개설 과정 선택 시 개설 과목 정보 출력	작성자	문예빈
Procedure			
<pre> CREATE OR REPLACE PROCEDURE procCurriculumInfo (p_openCurriculumNum IN NUMBER) IS v_curriculumName tblCurriculum.curriculumName%TYPE; v_subjectName tblSubject.subjectName%TYPE; v_subjectPeriod VARCHAR2(50); v_bookTitle VARCHAR2(1000); v_teacherName tblTeacher.teacherName%TYPE; </pre>			

```

CURSOR cur_curriculum IS
SELECT * FROM (
    SELECT
        c.curriculumName AS curriculumName,
        s.subjectName AS subjectName,
        TO_CHAR(os.openSubjectStartDay, 'YYYY-MM-DD') || ' ~ ' ||
            TO_CHAR(os.openSubjectEndDay, 'YYYY-MM-DD') AS subjectPeriod,
        LISTAGG(b.bookTitle, ', ') WITHIN GROUP (ORDER BY b.bookTitle) AS bookTitle,
        t.teacherName AS teacherName
    FROM tblOpenSubject os
        INNER JOIN tblSubject s ON os.subjectNum = s.subjectNum
        INNER JOIN tblOpenSOC osoc ON os.openSubjectNum = osoc.openSubjectNum
        INNER JOIN tblOpenCurriculum oc ON osoc.openCurriculumNum = oc.openCurriculumNum
        INNER JOIN tblCurriculum c ON oc.curriculumNum = c.curriculumNum
        INNER JOIN tblBook b ON os.openSubjectNum = b.openSubjectNum
        INNER JOIN tblTeacher t ON oc.teacherNum = t.teacherNum
    WHERE oc.openCurriculumNum = p_openCurriculumNum
    GROUP BY
        c.curriculumName,
        s.subjectName,
        os.openSubjectStartDay,
        os.openSubjectEndDay,
        t.teacherName
);

```

```

BEGIN
FOR rec IN cur_curriculum LOOP
    DBMS_OUTPUT.PUT_LINE('과정명: ' || rec.curriculumName);
    DBMS_OUTPUT.PUT_LINE('과목명: ' || rec.subjectName);
    DBMS_OUTPUT.PUT_LINE('과목기간: ' || rec.subjectPeriod);
    DBMS_OUTPUT.PUT_LINE('교재명: ' || rec.bookTitle);
    DBMS_OUTPUT.PUT_LINE('교사명: ' || rec.teacherName);
    DBMS_OUTPUT.PUT_LINE('-----');
END LOOP;
END;

BEGIN
    procCurriculumInfo(7); – 개설과정 번호
END;

```

요구사항번호	B-6(관리자)	작성일	2025-08-06
요구사항명	특정 교육생 선택시 개설 과정 정보와 수료 상태 출력	작성자	문예빈
Procedure			
<pre> CREATE OR REPLACE PROCEDURE procStudentCurriculum (p_studentNum IN NUMBER) IS v_curriculumName tblCurriculum.curriculumName%TYPE; v_period VARCHAR2(50); </pre>			

```

v_classroom VARCHAR2(20);
v_completion tblCompletionCategory.completionCategoryName%TYPE;

CURSOR cur_student IS
    SELECT
        c.curriculumName AS curriculumName,
        TO_CHAR(oc.openCurriStartDate, 'YYYY-MM-DD') || ' ~ ' ||
        TO_CHAR(oc.openCurriEndDate, 'YYYY-MM-DD') AS curriculumPeriod,
        oc.classroomNum || '호실' AS classroom,
        cc.completionCategoryName AS completionStatus
    FROM tblRegisterStudent rs
        INNER JOIN tblOpenCurriculum oc ON rs.openCurriculumNum = oc.openCurriculumNum
        INNER JOIN tblCurriculum c ON oc.curriculumNum = c.curriculumNum
        INNER JOIN tblCompletionCategory cc ON rs.completionCategoryNum = cc.completionCategoryNum
    WHERE rs.studentNum = p_studentNum;

BEGIN
    FOR rec IN cur_student LOOP
        DBMS_OUTPUT.PUT_LINE('과정명: ' || rec.curriculumName);
        DBMS_OUTPUT.PUT_LINE('과정기간: ' || rec.curriculumPeriod);
        DBMS_OUTPUT.PUT_LINE('강의실: ' || rec.classroom);
        DBMS_OUTPUT.PUT_LINE('수료상태: ' || rec.completionStatus);
        DBMS_OUTPUT.PUT_LINE('-----');
    END LOOP;
END;

```

```

BEGIN
    procStudentCurriculum(200); – 교육생 번호
END;

```

요구사항번호	B-7(관리자)	작성일	2025-08-06
요구사항명	특정 개설 과정을 선택 시 개설된 과목 정보를 출력 및 과목별 성적 등록 여부 출력	작성자	문예빈
Procedure			
<pre> CREATE OR REPLACE PROCEDURE procSubjectScoreStatus (p_openCurriculumNum IN NUMBER) IS v_count NUMBER := 0; CURSOR cur_subject IS SELECT s.subjectName AS subject_name, TO_CHAR(os.openSubjectStartDay, 'YYYY-MM-DD') ' ~ ' TO_CHAR(os.openSubjectEndDay, 'YYYY-MM-DD') AS subject_period, TO_CHAR(os.testDate, 'YYYY-MM-DD') AS test_date, CASE WHEN MAX(sc.attendanceScore) IS NULL THEN 'X' ELSE 'O' END AS attendanceScore; </pre>			

```

END AS attendance_registered,
CASE
    WHEN MAX(sc.writingScore) IS NULL THEN 'X'
    ELSE 'O'
END AS writing_registered,
CASE
    WHEN MAX(sc.practicalScore) IS NULL THEN 'X'
    ELSE 'O'
END AS practical_registered
FROM tblOpenSubject os
INNER JOIN tblSubject s ON os.subjectNum = s.subjectNum
INNER JOIN tblOpenSOC osoc ON os.openSubjectNum = osoc.openSubjectNum
INNER JOIN tblOpenCurriculum oc ON osoc.openCurriculumNum = oc.openCurriculumNum
LEFT OUTER JOIN tblScore sc ON oc.openCurriculumNum = sc.openCurriculumNum
WHERE oc.openCurriculumNum = p_openCurriculumNum
GROUP BY
    s.subjectName,
    os.openSubjectStartDay,
    os.openSubjectEndDay,
    os.testDate
ORDER BY os.openSubjectStartDay;

BEGIN
FOR rec IN cur_subject LOOP
    v_count := v_count + 1;

```

```

DBMS_OUTPUT.PUT_LINE('과목명: ' || rec.subject_name);
DBMS_OUTPUT.PUT_LINE('과목기간: ' || rec.subject_period);
DBMS_OUTPUT.PUT_LINE('시험날짜: ' || rec.test_date);
DBMS_OUTPUT.PUT_LINE('출결성적등록여부: ' || rec.attendance_registered);
DBMS_OUTPUT.PUT_LINE('필기성적등록여부: ' || rec.writing_registered);
DBMS_OUTPUT.PUT_LINE('실기성적등록여부: ' || rec.practical_registered);
DBMS_OUTPUT.PUT_LINE('-----');

END LOOP;

IF v_count = 0 THEN
    DBMS_OUTPUT.PUT_LINE('⚠️ 해당 개설 과정에 등록된 과목 정보가 없습니다.');
END IF;

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('❗️ 오류 발생: ' || SQLERRM);
END;

BEGIN
    procSubjectScoreStatus(3); -- 개설과정 번호
END;

```

요구사항번호	B-8(관리자)	작성일	2025-08-06
요구사항명	특정 개설 과정 선택시 기간별 출결 조회	작성자	문예빈
Procedure			
<pre> CREATE OR REPLACE PROCEDURE procAttendanceByDate (p_openCurriculumNum IN NUMBER, p_year IN NUMBER DEFAULT NULL, p_month IN NUMBER DEFAULT NULL, p_day IN NUMBER DEFAULT NULL) IS CURSOR cur_attendance IS SELECT st.studentName AS student_name, TO_CHAR(a.attendanceDate, 'YYYY-MM-DD') AS attendance_date, ac.attendanceCategoryName AS attendance_status FROM tblOpenCurriculum oc INNER JOIN tblRegisterStudent rs ON oc.openCurriculumNum = rs.openCurriculumNum INNER JOIN tblStudent st ON rs.studentNum = st.studentNum LEFT OUTER JOIN tblAttendance a ON rs.registerStudentNum = a.registerStudentNum INNER JOIN tblAttendanceCategory ac ON a.attendanceCategoryNum = ac.attendanceCategoryNum WHERE oc.openCurriculumNum = p_openCurriculumNum AND (p_year IS NULL OR EXTRACT(YEAR FROM a.attendanceDate) = p_year) AND (p_month IS NULL OR EXTRACT(MONTH FROM a.attendanceDate) = p_month) AND (p_day IS NULL OR EXTRACT(DAY FROM a.attendanceDate) = p_day) </pre>			

```

ORDER BY a.attendanceDate;

v_count NUMBER := 0;

BEGIN
FOR rec IN cur_attendance LOOP
    v_count := v_count + 1;
    DBMS_OUTPUT.PUT_LINE('이름: ' || rec.student_name);
    DBMS_OUTPUT.PUT_LINE('날짜: ' || rec.attendance_date);
    DBMS_OUTPUT.PUT_LINE('근태상황: ' || rec.attendance_status);
    DBMS_OUTPUT.PUT_LINE('-----');
END LOOP;

IF v_count = 0 THEN
    DBMS_OUTPUT.PUT_LINE('⚠️ 해당 조건에 맞는 출결 정보가 없습니다.');
END IF;

EXCEPTION
WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('❗️ 오류 발생: ' || SQLERRM);
END;

BEGIN
procAttendanceByDate(1, 2025, 8, 5); --(과정번호, 년, 월, 일) 년월일은 null허용
END;

```

요구사항번호	B-09(관리자)	작성일	25-08-06
요구사항명	사물함 배정	작성자	홍선민
Procedure			
<pre>--사물함 배정 프로시저 정의 CREATE OR REPLACE PROCEDURE procLockerAssign AS v_count NUMBER; BEGIN INSERT INTO TBLLOCKER (LOCKERNUM, CLASSROOMNUM, REGISTERSTUDENTNUM) SELECT LOCKERNUM_SEQ.NEXTVAL, O.CLASSROOMNUM, R.REGISTERSTUDENTNUM FROM TBLREGISTERSTUDENT R JOIN TBLOPENCURRICULUM O ON R.OPENCURRICULUMNUM = O.OPENCURRICULUMNUM WHERE R.COMPLETIONCATEGORYNUM = 1 AND NOT EXISTS (SELECT 1 FROM TBLLOCKER L WHERE L.REGISTERSTUDENTNUM = R.REGISTERSTUDENTNUM); v_count := SQL%ROWCOUNT;</pre>			

```

COMMIT;
DBMS_OUTPUT.PUT_LINE('[사물함 배정 결과]');
DBMS_OUTPUT.PUT_LINE(v_count || '명의 학생에게 사물함이 배정되었습니다.');
END;

```

--실행블럭

```

BEGIN
    procLockerAssign;
END;

```

요구사항번호	B-10(관리자)	작성일	25-08-06
요구사항명	개근 학생 포상	작성자	홍선민

Procedure

--개근 학생 전체 조회 프로시저 정의

```

CREATE OR REPLACE PROCEDURE procCheckPerfectAttendance IS
CURSOR cur IS
SELECT
    s.studentNum,
    s.studentName
FROM tblStudent s
INNER JOIN (
    SELECT registerStudentNum
    FROM tblAttendance
    GROUP BY registerStudentNum
)

```

```

HAVING MIN(attendanceCategoryNum) = 4
    AND MAX(attendanceCategoryNum) = 4
) a ON s.studentNum = a.registerStudentNum
ORDER BY s.studentNum;

v_studentNum tblStudent.studentNum%TYPE;
v_studentName tblStudent.studentName%TYPE;
BEGIN
DBMS_OUTPUT.PUT_LINE('[개근한 학생 목록]');
DBMS_OUTPUT.PUT_LINE('-----');

FOR rec IN cur LOOP
    v_studentNum := rec.studentNum;
    v_studentName := rec.studentName;

    DBMS_OUTPUT.PUT_LINE('학생 번호 : ' || v_studentNum);
    DBMS_OUTPUT.PUT_LINE('이름      : ' || v_studentName);
    DBMS_OUTPUT.PUT_LINE(""); -- 줄바꿈
END LOOP;
END;

```

--실행블럭

```

BEGIN
    procCheckPerfectAttendance;
END;

```

요구사항번호	B-11(관리자)	작성일	25-08-06
요구사항명	취업률 조회	작성자	홍선민
Procedure			
<pre>--과정별 취업률 조회 프로시저 정의 CREATE OR REPLACE PROCEDURE procEmployRateByCurr (p_openCurriculumNum IN TBLOPENCURRICULUM.OPENCURRICULUMNUM%TYPE) IS -- 결과 저장 변수 v_과정번호 TBLOPENCURRICULUM.OPENCURRICULUMNUM%TYPE; v_과정명 TBLCURRICULUM.CURRICULUMNAME%TYPE; v_취업률 NUMBER; BEGIN -- 단일 레코드 조회 SELECT oc.OPENCURRICULUMNUM, curr.CURRICULUMNAME, ROUND((COUNT(CASE WHEN js.JOBKINDCATEGORYNUM IN (1, 3) THEN 1 END) * 100.0) / COUNT(*), 2) INTO v_과정번호, v_과정명, v_취업률 FROM TBLREGISTERSTUDENT rs JOIN TBLOPENCURRICULUM oc</pre>			

```
ON rs.OPENCURRICULUMNUM = oc.OPENCURRICULUMNUM
JOIN TBLJOBSTATUS js
    ON rs.REGISTERSTUDENTNUM = js.REGISTERSTUDENTNUM
JOIN TBLCURRICULUM curr
    ON curr.CURRICULUMNUM = oc.CURRICULUMNUM
WHERE rs.COMPLETIONCATEGORYNUM = 3
    AND js.JOBKINDCATEGORYNUM IN (1, 2, 3)
    AND oc.OPENCURRICULUMNUM = p_openCurriculumNum
GROUP BY oc.OPENCURRICULUMNUM, curr.CURRICULUMNAME;
```

-- 출력

```
DBMS_OUTPUT.PUT_LINE('과정별 취업률 확인]');
DBMS_OUTPUT.PUT_LINE('과정번호: ' || v_과정번호);
DBMS_OUTPUT.PUT_LINE('과정명: ' || v_과정명);
DBMS_OUTPUT.PUT_LINE('취업률: ' || v_취업률 || '%');
```

EXCEPTION

WHEN NO_DATA_FOUND THEN

```
DBMS_OUTPUT.PUT_LINE('과정별 취업률 확인]');
DBMS_OUTPUT.PUT_LINE('해당 과정번호에 대한 수료생 데이터가 없습니다.');
```

END;

--실행블럭

BEGIN

--파라미터로 원하는 과정번호 입력

```
procEmployRateByCurr(1);
procEmployRateByCurr(7);
procEmployRateByCurr(8);
procEmployRateByCurr(9);
procEmployRateByCurr(10);

END;
```

--년도별 취업률 조회 프로시저 정의

```
CREATE OR REPLACE PROCEDURE procEmployRateByYear (
    p_year IN VARCHAR2
) IS
    v_total_count NUMBER := 0;
    v_employed_count NUMBER := 0;
    v_rate NUMBER := NULL;

BEGIN
    -- 전체 수료자 수 조회
    SELECT COUNT(*) INTO v_total_count
    FROM TBLREGISTERSTUDENT rs
    JOIN tblJobStatus js
        ON rs.registerStudentNum = js.registerStudentNum
    JOIN TBLOPENCURRICULUM oc
        ON rs.openCurriculumNum = oc.openCurriculumNum
    WHERE rs.completionCategoryNum = 3
        AND js.jobKindCategoryNum IN (1, 2, 3)
        AND TO_CHAR(oc.OPENCURRIENDDATE, 'YYYY') = p_year;
```

```

IF v_total_count = 0 THEN
    DBMS_OUTPUT.PUT_LINE('[과정별 취업률 확인]');
    DBMS_OUTPUT.PUT_LINE('해당 년도(' || p_year || ')에 대한 수료생 데이터가 없습니다.');
    RETURN;
END IF;

-- 취업자 + 퇴사자 수 조회
SELECT COUNT(*) INTO v_employed_count
FROM TBLREGISTERSTUDENT rs
JOIN tblJobStatus js
    ON rs.registerStudentNum = js.registerStudentNum
JOIN TBLOPENCURRICULUM oc
    ON rs.openCurriculumNum = oc.openCurriculumNum
WHERE rs.completionCategoryNum = 3
    AND js.jobKindCategoryNum IN (1, 3)
    AND TO_CHAR(oc.OPENCURRIENDDATE, 'YYYY') = p_year;

-- 취업률 계산
v_rate := ROUND((v_employed_count * 100.0) / v_total_count, 2);
DBMS_OUTPUT.PUT_LINE('[과정별 취업률 확인]');
DBMS_OUTPUT.PUT_LINE('년도: ' || p_year);
DBMS_OUTPUT.PUT_LINE('취업률: ' || v_rate || '%');
END;
/

```

--실행블럭

```
BEGIN  
    procEmployRateByYear('2024');  
    procEmployRateByYear('2025');  
END;  
/
```

2. 교사 기능

요구사항번호	C-02(교사)	작성일	25-08-06
요구사항명	강의 스케줄 조회	작성자	홍선민
Procedure			
<pre>--특정 과정에 등록한 학생의 상세 정보 확인하는 프로시저 정의문 --교육중인 과정에 등록한 교육생 정보(이름, 전화번호, 회원가입일, 수료상태) CREATE OR REPLACE PROCEDURE PROCSHOWRegStudentInfo(p_openCurriculumNum IN NUMBER) IS CURSOR c_student_info IS SELECT s.studentName AS 이름, s.sPhoneNum AS 전화번호, cc.COMPLETIONCATEGORYNAME AS 수료상태, TO_CHAR(s.dateOfRegistration, 'YYYY-MM-DD') AS 회원가입일 FROM tblStudent s JOIN tblRegisterStudent rs ON s.studentNum = rs.studentNum JOIN tblCompletionCategory cc ON rs.completionCategoryNum = cc.completionCategoryNum JOIN tblOpenCurriculum oc ON rs.openCurriculumNum = oc.openCurriculumNum WHERE rs.completionCategoryNum IN (1, 2) AND oc.openCurriEndDate >= SYSDATE AND rs.openCurriculumNum = p_openCurriculumNum; TYPE student_rec_type IS TABLE OF c_student_info%ROWTYPE;</pre>			

```

v_data student_rec_type;

v_curriculumName TBLCURRICULUM.curriculumName%TYPE;

BEGIN
    -- 학생 데이터 수집
    OPEN c_student_info;
    FETCH c_student_info BULK COLLECT INTO v_data;
    CLOSE c_student_info;

    DBMS_OUTPUT.PUT_LINE('[개설 과정별 등록 학생 확인]');

    IF v_data.COUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('과정번호 ' || p_openCurriculumNum || '에 대한 등록된 학생이 없습니다.');
    ELSE
        -- 과정명 조회
        SELECT c.curriculumName INTO v_curriculumName
        FROM tblOpenCurriculum oc
        JOIN tblCurriculum c ON oc.curriculumNum = c.curriculumNum
        WHERE oc.openCurriculumNum = p_openCurriculumNum;

        -- 헤더 출력
        DBMS_OUTPUT.PUT_LINE('-----');
        DBMS_OUTPUT.PUT_LINE('과정번호: ' || p_openCurriculumNum);
        DBMS_OUTPUT.PUT_LINE('과정명 : ' || v_curriculumName);
        DBMS_OUTPUT.PUT_LINE('-----');
    END IF;
END;

```

```

DBMS_OUTPUT.PUT_LINE(
    RPAD('이름', 10) || ' | ' ||
    RPAD('전화번호', 15) || ' | ' ||
    RPAD('수료상태', 12) || ' | ' ||
    '회원가입일'
);
DBMS_OUTPUT.PUT_LINE('-----');

-- 데이터 출력
FOR i IN 1..v_data.COUNT LOOP
    DBMS_OUTPUT.PUT_LINE(
        RPAD(v_data(i).이름, 10) || ' | ' ||
        RPAD(v_data(i).전화번호, 15) || ' | ' ||
        RPAD(v_data(i).수료상태, 12) || ' | ' ||
        v_data(i).회원가입일
    );
END LOOP;
END IF;
END;

--실행블럭
BEGIN
    PROCShowRegStudentInfo(1); -- 원하는 과정번호
    PROCShowRegStudentInfo(12);
END;
/

```

요구사항번호	C-04(교사)	작성일	25-08-06
요구사항명	성적 입출력	작성자	박진혁
Procedure			
<pre>/*특정 교육생 번호 선택 시 교육생 시험 점수 입력(출결, 필기, 실기 구분)*/ CREATE OR REPLACE PROCEDURE prc_insert_score (p_attendance_score IN NUMBER, -- 출결 점수 p_writing_score IN NUMBER, -- 필기 점수 p_practical_score IN NUMBER, -- 실기 점수 p_open_subject_num IN NUMBER, -- 과목 번호 (openSubjectNum) p_register_student_num IN NUMBER -- 수강생 번호 (registerStudentNum)) AS BEGIN INSERT INTO tblScore (ScoreNum, attendanceScore, writingScore, practicalScore, openSubjectNum, registerStudentNum) VALUES (tblScore_seq.NEXTVAL, p_attendance_score, p_writing_score,</pre>			

```

    p_practical_score,
    p_open_subject_num,
    p_register_student_num
);

COMMIT;
DBMS_OUTPUT.PUT_LINE('성적 입력이 완료되었습니다.');

EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('오류 발생: ' || SQLERRM);
END;
/

BEGIN
  prc_insert_score(
    p_attendance_score    => 80,    -- 출결점수
    p_writing_score        => 80,      -- 필기점수
    p_practical_score     => 80,      -- 실기점수
    p_open_subject_num    => 7,       -- 과목넘버
    p_register_student_num => 173    -- 수강생번호
  );
END;
/

```

```
/*교육생 시험 점수가 등록되어있을 경우 수정*/
```

```
CREATE OR REPLACE PROCEDURE prc_update_score (
    p_attendance_score    IN NUMBER, -- 출결 점수
    p_writing_score        IN NUMBER, -- 필기 점수
    p_practical_score     IN NUMBER, -- 실기 점수
    p_open_subject_num    IN NUMBER, -- 과목 번호 (openSubjectNum)
    p_register_student_num IN NUMBER  -- 수강생 번호 (registerStudentNum)
) AS
BEGIN
    UPDATE tblScore
    SET attendanceScore = p_attendance_score,
        writingScore   = p_writing_score,
        practicalScore = p_practical_score
    WHERE openSubjectNum = p_open_subject_num
        AND registerStudentNum = p_register_student_num;

    IF SQL%ROWCOUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('수정할 대상이 없습니다.');
    ELSE
        COMMIT;
        DBMS_OUTPUT.PUT_LINE('점수가 성공적으로 수정되었습니다.');
    END IF;

EXCEPTION
```

```

WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('오류 발생: ' || SQLERRM);
END;
/

BEGIN
    prc_update_score(
        p_attendance_score    => 20,
        p_writing_score       => 40,
        p_practical_score    => 40,
        p_open_subject_num   => 7,
        p_register_student_num => 173
    );
END;
/

/*특정 과정 선택 시 교육생 정보(교육생 이름, 전화번호, 수료 또는 중도포기) 및 성적 출력*/
CREATE OR REPLACE PROCEDURE prc_view_student_score (
    p_open_curriculum_num IN NUMBER
) AS
BEGIN
    DBMS_OUTPUT.PUT_LINE('▼ 수강생 성적 조회 결과 ▼');

    FOR rec IN (
        SELECT

```

```

oc.openCurriculumNum AS 과정번호,
ts.studentName AS 이름,
ts.sPhoneNum AS 전화번호,
CASE
    WHEN rs.completionCategoryNum = 3 THEN '수료'
    WHEN rs.completionCategoryNum = 2 THEN '중도포기'
END AS 수료상태,
(sc.attendanceScore * 0.2) AS 출결점수,
(sc.writingScore * 0.4) AS 필기점수,
(sc.practicalScore * 0.4) AS 실기점수
FROM tblRegisterStudent rs
INNER JOIN tblStudent ts ON rs.studentNum = ts.studentNum
INNER JOIN tblOpenCurriculum oc ON rs.openCurriculumNum = oc.openCurriculumNum
INNER JOIN tblOpenSOC soc ON oc.openCurriculumNum = soc.openCurriculumNum
INNER JOIN tblOpenSubject os ON soc.openSubjectNum = os.openSubjectNum
INNER JOIN tblSubject sb ON os.subjectNum = sb.subjectNum
INNER JOIN tblScore sc ON rs.registerStudentNum = sc.registerStudentNum
WHERE (rs.completionCategoryNum = 3 OR rs.completionCategoryNum = 2)
    AND oc.openCurriculumNum = p_open_curriculum_num
) LOOP
DBMS_OUTPUT.PUT_LINE(
'과정번호: ' || rec.과정번호 ||
'| 이름: ' || rec.이름 ||
'| 전화번호: ' || rec.전화번호 ||
'| 수료상태: ' || rec.수료상태 ||

```

```
' | 출결점수: ' || rec.출결점수 ||
' | 필기점수: ' || rec.필기점수 ||
' | 실기점수: ' || rec.실기점수
);
END LOOP;

EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('오류 발생: ' || SQLERRM);
END;
/

BEGIN
prc_view_student_score(p_open_curriculum_num => 7); --특정 과정번호 입력
END;
/
```

요구사항번호	C-05(교사)	작성일	2025-08-06
요구사항명	출결 관리 및 출결 조회	작성자	박진혁
Procedure			
<pre> /*자신이 강의한 과정에 한해 선택하는 경우 모든 교육생의 출결을 조회*/ CREATE OR REPLACE PROCEDURE prc_view_attendance (p_teacher_num IN NUMBER, -- 교사 번호 p_open_curriculum_num IN NUMBER -- 과정 번호) AS BEGIN DBMS_OUTPUT.PUT_LINE('▼ 출결 현황 조회 ▼'); FOR rec IN (SELECT tt.teacherName AS 교사이름, tc.curriculumName AS 과정명, ts.studentName AS 교육생이름, TO_CHAR(ad.attendanceDate, 'YYYY-MM-DD') AS 날짜, CASE ad.attendanceCategoryNum WHEN 1 THEN '결석' WHEN 2 THEN '조퇴' WHEN 3 THEN '지각' WHEN 4 THEN '출석' END AS 출결현황)</pre>			

```

FROM tblOpenCurriculum oc
    INNER JOIN tblTeacher tt ON oc.teacherNum = tt.teacherNum
    INNER JOIN tblCurriculum tc ON oc.curriculumNum = tc.curriculumNum
    INNER JOIN tblRegisterStudent rs ON oc.openCurriculumNum = rs.openCurriculumNum
    INNER JOIN tblStudent ts ON rs.studentNum = ts.studentNum
    INNER JOIN tblAttendance ad ON rs.registerStudentNum = ad.registerStudentNum
WHERE tt.teacherNum = p_teacher_num
    AND oc.openCurriculumNum = p_open_curriculum_num
) LOOP
    DBMS_OUTPUT.PUT_LINE(
        '날짜:' || rec.날짜 ||
        ' | 교사:' || rec.교사이름 ||
        ' | 과정명:' || rec.과정명 ||
        ' | 교육생:' || rec.교육생이름 ||
        ' | 출결:' || rec.출결현황
    );
END LOOP;

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('오류 발생: ' || SQLERRM);
END;
/
BEGIN

```

```
prc_view_attendance(  
    p_teacher_num => 1,  
    p_open_curriculum_num => 1  
);  
END;  
/  
/
```

요구사항번호	C-06(교사)	작성일	2025-08-06
요구사항명	수강 후기 조회	작성자	박진혁
Procedure			
<pre>/*수강 후기 조회*/ CREATE OR REPLACE PROCEDURE prc_show_teacher_reviews (p_teacher_num IN NUMBER) IS BEGIN DBMS_OUTPUT.PUT_LINE('▼ 종료된 과정의 후기 목록 ▼'); FOR rec IN (SELECT tc.curriculumName AS 과정명, TO_CHAR(oc.openCurriStartDate, 'YYYY-MM-DD') ' ~ ' TO_CHAR(oc.openCurriEndDate, 'YYYY-MM-DD') AS 과정기간, tr.reviewRating AS 평점, tr.reviewDetail AS 후기 FROM tblOpenCurriculum oc INNER JOIN tblTeacher tt ON oc.teacherNum = tt.teacherNum INNER JOIN tblReview tr ON oc.openCurriculumNum = tr.openCurriculumNum INNER JOIN tblCurriculum tc ON oc.curriculumNum = tc.curriculumNum WHERE oc.openCurriEndDate <= SYSDATE) LOOP DBMS_OUTPUT.PUT_LINE(rec.과정명 ' (' rec.과정기간 ')'); DBMS_OUTPUT.PUT_LINE('평점: ' rec.평점); DBMS_OUTPUT.PUT_LINE('후기: ' rec.후기); DBMS_OUTPUT.PUT_LINE('-----'); END LOOP; END;</pre>			

```
        AND tt.teacherNum = p_teacher_num
    ) LOOP
    DBMS_OUTPUT.PUT_LINE(
        '과정명: ' || rec.과정명 ||
        ' | 기간: ' || rec.과정기간 ||
        ' | 평점: ' || rec.평점 ||
        ' | 후기: ' || rec.후기
    );
END LOOP;

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('오류 발생: ' || SQLERRM);
END;
/

BEGIN
    prc_show_teacher_reviews(1); -- 1번 교사에 대한 후기 목록 조회
END;
/
```

요구사항번호	C-07(교사)	작성일	2025-08-06
요구사항명	Q&A 게시판 답변	작성자	박진혁
Procedure			
<pre> /*교육생들이 올린 질문*/ SELECT * FROM vw_QuestionAnswerDetails ORDER BY 질문번호 ASC; --View 테이블 조회 /*답변 등록*/ CREATE OR REPLACE PROCEDURE prc_insert_answer (p_question_num IN NUMBER, p_teacher_num IN NUMBER, p_answer_detail IN VARCHAR2) AS BEGIN INSERT INTO tblAnswer (answerNum, answerDetail, questionNum, teacherNum) VALUES (answerNum_seq.NEXTVAL, p_answer_detail, p_question_num, p_teacher_num); COMMIT; DBMS_OUTPUT.PUT_LINE('답변이 등록되었습니다.'); EXCEPTION WHEN OTHERS THEN DBMS_OUTPUT.PUT_LINE('답변 등록 실패: ' SQLERRM); END; </pre>			

```
/

/*답변 조회*/

CREATE OR REPLACE PROCEDURE prc_select_answer (
    p_teacher_num IN NUMBER,
    p_answer_num  IN NUMBER
) AS
    v_detail tblAnswer.answerDetail%TYPE;
BEGIN
    SELECT answerDetail
    INTO v_detail
    FROM tblAnswer
    WHERE teacherNum = p_teacher_num
        AND answerNum = p_answer_num;

    DBMS_OUTPUT.PUT_LINE('▼ 조회된 답변 내용 ▼');
    DBMS_OUTPUT.PUT_LINE(v_detail);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('해당 답변이 존재하지 않습니다.');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('오류 발생: ' || SQLERRM);
END;
/
```

```
/*답변 수정*/
```

```
CREATE OR REPLACE PROCEDURE prc_update_answer (
    p_teacher_num  IN NUMBER,
    p_question_num IN NUMBER,
    p_new_detail   IN VARCHAR2
) AS
BEGIN
    UPDATE tblAnswer
    SET answerDetail = p_new_detail
    WHERE teacherNum = p_teacher_num
        AND questionNum = p_question_num;

    IF SQL%ROWCOUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('수정 대상이 없습니다.');
    ELSE
        COMMIT;
        DBMS_OUTPUT.PUT_LINE('답변이 수정되었습니다.');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('수정 중 오류 발생: ' || SQLERRM);
END;
/
```

```
/*답변 삭제*/
```

```

CREATE OR REPLACE PROCEDURE prc_delete_answer (
    p_teacher_num IN NUMBER,
    p_question_num IN NUMBER
) AS
BEGIN
    DELETE FROM tblAnswer
    WHERE teacherNum = p_teacher_num
        AND questionNum = p_question_num;

    IF SQL%ROWCOUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('삭제 대상이 없습니다.');
    ELSE
        COMMIT;
        DBMS_OUTPUT.PUT_LINE('답변이 삭제되었습니다.');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('삭제 중 오류 발생: ' || SQLERRM);
END;
/
BEGIN
    prc_insert_answer(1, 1, '수업 중 답변 드립니다.'); --답변 등록 (질문번호, 교사번호, 내용)
    prc_select_answer(1, 1); --답변 조회(교사번호, 답변번호)

```

```

prc_update_answer(1, 1, '내용 수정됨'); --답변 수정(교사번호, 답변번호, 내용)
prc_delete_answer(1, 1); --답변 삭제(교사번호, 답변번호)
END;
/

```

요구사항번호	C-08(교사)	작성일	2025-08-06
요구사항명	취업률 조회	작성자	박진혁
Procedure			
<pre> /*취업률 조회 실행*/ BEGIN --파라미터로 원하는 과정번호 입력 procEmployRateByCurr(1); procEmployRateByCurr(7); procEmployRateByCurr(8); procEmployRateByCurr(9); procEmployRateByCurr(10); END; / </pre>			

3. 교육생 기능

요구사항번호	D-3(교육생)	작성일	2025-08-06
요구사항명	성적 조회 및 과목별 상세 정보	작성자	최석용
Procedure			
<pre>--인기 과정 조회 프로시저 CREATE OR REPLACE PROCEDURE procreviewrating IS BEGIN DBMS_OUTPUT.PUT_LINE('====='); DBMS_OUTPUT.PUT_LINE('과정명 평균 별점'); DBMS_OUTPUT.PUT_LINE('-----'); FOR rec IN (SELECT c.curriculumName AS 과정명, ROUND(AVG(r.reviewRating), 2) AS 평균별점 FROM tblReview r INNER JOIN tblOpenCurriculum oc ON r.openCurriculumNum = oc.openCurriculumNum INNER JOIN tblCurriculum c ON oc.curriculumNum = c.curriculumNum GROUP BY c.curriculumName ORDER BY</pre>			

```

        ROUND(AVG(r.reviewRating), 2) DESC
    )
LOOP
DBMS_OUTPUT.PUT_LINE(
    RPAD(rec.과정명, 30) || ' | ' || rec.평균별점
);
END LOOP;

DBMS_OUTPUT.PUT_LINE('=====');
END;
/

```

BEGIN
 procreviewrating;
END;
/

요구사항번호	D-3(교육생)	작성일	2025-08-06
요구사항명	성적 조회 및 과목별 상세 정보	작성자	최석용
Procedure			
--특정 학생의 성적 데이터 프로시저			
CREATE OR REPLACE PROCEDURE procscores (

```

p_registerstudentnum IN NUMBER
)
IS
v_found BOOLEAN := FALSE;
BEGIN
FOR rec IN (
  SELECT
    cul.CURRICULUMNUM AS 과정번호,
    cul.curriculumname AS 과정명,
    (sc.attendancescore * 0.2) AS 출석점수_20,
    (sc.writingscore * 0.4) AS 필기점수_40,
    (sc.practicalscore * 0.4) AS 실기점수_40,
    (sc.attendancescore * 0.2 + sc.writingscore * 0.4 + sc.practicalscore * 0.4) AS 총점_100
  FROM tblregisterstudent res
  INNER JOIN tblscore sc
    ON res.registerstudentnum = sc.registerstudentnum
  INNER JOIN tblopencurriculum ocul
    ON res.opencurriculumnum = ocul.opencurriculumnum
  INNER JOIN tblcurriculum cul
    ON ocul.curriculumnum = cul.curriculumnum
  WHERE res.registerstudentnum = p_registerstudentnum
)
LOOP
v_found := TRUE;

```

```

DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE('과정번호 : ' || rec.과정번호);
DBMS_OUTPUT.PUT_LINE('과정명 : ' || rec.과정명);
DBMS_OUTPUT.PUT_LINE('출석점수(20) : ' || TO_CHAR(rec.출석점수_20, 'FM99990.00'));
DBMS_OUTPUT.PUT_LINE('필기점수(40) : ' || TO_CHAR(rec.필기점수_40, 'FM99990.00'));
DBMS_OUTPUT.PUT_LINE('실기점수(40) : ' || TO_CHAR(rec.실기점수_40, 'FM99990.00'));
DBMS_OUTPUT.PUT_LINE('총점(100) : ' || TO_CHAR(rec.총점_100, 'FM99990.00'));
DBMS_OUTPUT.PUT_LINE('-----');

END LOOP;

IF NOT v_found THEN
    DBMS_OUTPUT.PUT_LINE('해당 학생의 점수 데이터가 없습니다.');
END IF;
END;

BEGIN
    procscores(171);
END;
/

```

--과목별 교재 정보 프로시저

```

CREATE OR REPLACE PROCEDURE procBookOpenSubject (
    -- 파라미터로 '개설 과목 번호'를 받습니다.
    p_open_subject_num IN tblOpenSubject.openSubjectNum%TYPE
)

```

```

)
IS
    v_found BOOLEAN := FALSE;
BEGIN
    FOR rec IN (
        SELECT
            sub.subjectname AS "과목명",
            b.booktitle AS "책제목",
            b.author AS "저자",
            b.publisher AS "출판사"
        FROM
            tblSubject sub
        INNER JOIN tblOpenSubject osub ON sub.subjectNum = osub.subjectNum
        INNER JOIN tblBook b ON osub.openSubjectNum = b.openSubjectNum
        -- 검색 기준을 subjectNum에서 openSubjectNum으로 변경했습니다.
        WHERE
            osub.openSubjectNum = p_open_subject_num
    )
    LOOP
        IF NOT v_found THEN
            DBMS_OUTPUT.PUT_LINE('===== [' || rec."과목명" || '] 과목 교재 정보 (개설번호: ' || p_open_subject_num || ') =====');
            v_found := TRUE;
        END IF;

        DBMS_OUTPUT.PUT_LINE('책제목 : ' || rec."책제목");
    )

```

```

DBMS_OUTPUT.PUT_LINE('저자 : ' || rec."저자");
DBMS_OUTPUT.PUT_LINE('출판사 : ' || rec."출판사");
DBMS_OUTPUT.PUT_LINE('-----');
END LOOP;

IF NOT v_found THEN
    DBMS_OUTPUT.PUT_LINE(p_open_subject_num || '번으로 개설된 과목의 교재 정보가 없습니다.');
END IF;
END;

BEGIN
    procBookOpenSubject(17); --java는 openSubjectNum이 17번이므로 17번 입력시 자바에 대한 책 2개
END;

```

--과목별 시험날짜와 시험문제 출력

```

CREATE OR REPLACE PROCEDURE probsubTestdate(
    p_subjectNum IN NUMBER
)
IS
BEGIN
    FOR rec IN (
        SELECT
            sub.subjectname AS subject_name,

```

```

        osub.testdate AS test_date,
        tque.examquestion AS exam_question
    FROM tblSubject sub
        INNER JOIN tblOpenSubject osub ON sub.subjectNum = osub.subjectNum
        INNER JOIN tblTestPaper tpaper ON osub.openSubjectNum = tpaper.openSubjectNum
        INNER JOIN tblTestService tserv ON tpaper.testPaperNum = tserv.testPaperNum
        INNER JOIN tblTestQuestion tque ON tserv.testQuestionNum = tque.testQuestionNum
    WHERE sub.subjectNum = p_subjectNum
    ORDER BY osub.testdate
)
LOOP
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE('과목명 : ' || rec.subject_name);
DBMS_OUTPUT.PUT_LINE('시험날짜 : ' || TO_CHAR(rec.test_date, 'YYYY-MM-DD'));
DBMS_OUTPUT.PUT_LINE('시험문제 : ' || rec.exam_question);
END LOOP;
END;

BEGIN
-- 예시: 17번 개설 과목의 시험 정보를 조회합니다.
procsubTestdate(2);
END;

```

요구사항번호	D-4(교육생)	작성일	2025-08-06
요구사항명	출결 관리 및 출결 조회	작성자	최석용
Procedure			
<pre>--특정 학생의 월별 출결 출력 프로시저 CREATE OR REPLACE PROCEDURE procmonth (p_registerStudentNum IN NUMBER) IS v_count_absent NUMBER := 0; v_count_early_leave NUMBER := 0; v_count_late NUMBER := 0; v_count_present NUMBER := 0; BEGIN DBMS_OUTPUT.PUT_LINE('== 학생번호: ' p_registerStudentNum '의 4월 출결 현황 =='); DBMS_OUTPUT.PUT_LINE('-----'); DBMS_OUTPUT.PUT_LINE(RPAD('날짜', 12) ' 출결상태'); DBMS_OUTPUT.PUT_LINE('-----'); FOR rec IN (SELECT ad.attendanceDate, CASE ad.attendanceCategoryNum WHEN 1 THEN '결석' WHEN 2 THEN '조퇴' END AS attendanceStatus FROM attendance ad WHERE ad.studentNum = p_registerStudentNum ORDER BY ad.attendanceDate) LOOP DBMS_OUTPUT.PUT_LINE(rec.attendanceDate ' ' rec.attendanceStatus); END LOOP; END;</pre>			

```

WHEN 3 THEN '지각'
WHEN 4 THEN '출석'
ELSE '기타'
END AS 출결현황
FROM tblattendance ad
WHERE TO_CHAR(ad.attendanceDate, 'MM') = '04'
AND ad.REGISTERSTUDENTNUM = p_registerStudentNum
ORDER BY ad.attendanceDate
)
LOOP
DBMS_OUTPUT.PUT_LINE(TO_CHAR(rec.attendanceDate, 'YYYY-MM-DD') || ' | ' || rec.출결현황);

-- 출결 상태별 카운트 집계
CASE rec.출결현황
WHEN '결석' THEN v_count_absent := v_count_absent + 1;
WHEN '조퇴' THEN v_count_early_leave := v_count_early_leave + 1;
WHEN '지각' THEN v_count_late := v_count_late + 1;
WHEN '출석' THEN v_count_present := v_count_present + 1;
END CASE;
END LOOP;

DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE('결석: ' || v_count_absent || '회, 조퇴: ' || v_count_early_leave || '회, 지각: ' || v_count_late || '회, 출석: ' || v_count_present || '회');
DBMS_OUTPUT.PUT_LINE('=====');
```

```
END;
```

```
BEGIN
```

```
    procmonth(11);
```

```
END;
```

--특정 학생의 일별 출결 출력 프로시저

```
CREATE OR REPLACE PROCEDURE procattendanceDate (
    p_registerStudentNum IN NUMBER,
    p_attendanceDate    IN DATE
)
IS
    v_row_count NUMBER := 0;
BEGIN
    DBMS_OUTPUT.PUT_LINE('== 학생번호 ' || p_registerStudentNum || '의 ' || TO_CHAR(p_attendanceDate, 'YYYY-MM-DD') || ' 출결 기록 ==');
    DBMS_OUTPUT.PUT_LINE('-----');

    FOR v_rec IN (
        SELECT
            stu.studentName AS 이름,
            CASE ad.attendanceCategoryNum
                WHEN 1 THEN '결석'
                WHEN 2 THEN '조퇴'
            END AS 출결상태
        FROM student stu
        JOIN attendance ad ON stu.studentNum = ad.studentNum
        WHERE stu.studentNum = p_registerStudentNum
        AND ad.attendanceDate = p_attendanceDate
    )
    LOOP
        DBMS_OUTPUT.PUT_LINE(v_rec.이름 || ' ' || v_rec.출결상태);
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('-----');
END;
```

```

        WHEN 3 THEN '지각'
        WHEN 4 THEN '출석'
    END AS 출결현황
FROM tblattendance ad
INNER JOIN TBLREGISTERSTUDENT rst ON ad.REGISTERSTUDENTNUM = rst.REGISTERSTUDENTNUM
INNER JOIN tblstudent stu ON stu.STUDENTNUM = rst.STUDENTNUM
WHERE rst.REGISTERSTUDENTNUM = p_registerStudentNum
    AND TRUNC(ad.attendanceDate) = TRUNC(p_attendanceDate)
)
LOOP
    v_row_count := v_row_count + 1;
    DBMS_OUTPUT.PUT_LINE('이름: ' || v_rec.이름 || ' | 상태: ' || v_rec.출결현황);
END LOOP;

IF v_row_count = 0 THEN
    DBMS_OUTPUT.PUT_LINE('>> 해당 날짜의 출결 기록이 없습니다.');
END IF;

DBMS_OUTPUT.PUT_LINE('-----');
END;

BEGIN
    proc_attendanceDate(11, TO_DATE('2025-04-25', 'YYYY-MM-DD'));
END;

```

요구사항번호	D-5(교육생)	작성일	2025-08-06
요구사항명	수강 완료 후 기능	작성자	최석용
Procedure			
<pre>--리뷰 남기기 프로시저 CREATE OR REPLACE PROCEDURE procreviewinsert (p_reviewRating IN NUMBER, p_reviewDetail IN VARCHAR2, p_openCurriculumNum IN NUMBER, p_registerStudentNum IN NUMBER) IS BEGIN -- openCurriculumNum이 7~12 사이인지 확인 IF p_openCurriculumNum BETWEEN 7 AND 12 THEN INSERT INTO tblReview (reviewNum, reviewRating, reviewDetail, openCurriculumNum, registerStudentNum) VALUES (reviewNum_seq.NEXTVAL, p_reviewRating, p_reviewDetail, p_openCurriculumNum, p_registerStudentNum); DBMS_OUTPUT.PUT_LINE('후기가 성공적으로 등록되었습니다.'); ELSE DBMS_OUTPUT.PUT_LINE('후기는 openCurriculumNum이 7~12번인 경우에만 등록할 수 있습니다.'); END IF; END;</pre>			

```
END IF;  
END;  
/  
  
BEGIN  
    procreviewinsert(5, '강사님이 끝까지 신경써주셔서 좋은 회사에 취업했어요. ', 8, 210);  
END;  
/
```

B. 뷰

1. 관리자 기능

요구사항번호	B-3(관리자)	작성일	2025-08-06
요구사항명	교사 계정 관리	작성자	문예빈
View			
<pre>/* 교사 전체 명단 출력 */ CREATE OR REPLACE VIEW vwTeacherInfo -- 뷰 정의문 AS SELECT t.teacherName AS 교사명, t.tResidentNum AS 주민번호뒷자리, t.tPhoneNum AS 전화번호, LISTAGG(s.subjectName, ', ') WITHIN GROUP (ORDER BY s.subjectName) AS 강의가능과목 FROM tblTeacher t INNER JOIN tblPossibleSubject p ON t.teacherNum = p.teacherNum INNER JOIN tblSubject s ON p.subjectNum = s.subjectNum GROUP BY t.teacherName, t.tResidentNum, t.tPhoneNum; SELECT * FROM vwTeacherInfo; -- 뷰 확인용 select문 /* 교사별 강의 가능 과목 출력 */</pre>			

-- 뷰 확인용 select문

SELECT

교사명, 강의가능과목

FROM vwTeacherInfo;

요구사항번호	B-4(관리자)	작성일	2025-08-06
요구사항명	개설 과정 관리	작성자	문예빈

View

CREATE OR REPLACE VIEW vwOpenCurriculum -- 뷰 정의문

AS

SELECT

c.curriculumName AS "과정명",
to_char(oc.openCurriStartDate, 'YYYY-MM-DD')
|| ' ~ ' || to_char(oc.openCurriEndDate, 'YYYY-MM-DD') AS "과정기간",
oc.classroomNum || '호실' AS 강의실명,

CASE

WHEN (

SELECT count(*) FROM tblSubjectOfCurriculum soc
WHERE soc.curriculumNum = oc.curriculumNum

) = (

SELECT count(*) FROM tblOpenSOC osoc
JOIN tblOpenSubject os ON osoc.openSubjectNum = os.openSubjectNum
WHERE osoc.openCurriculumNum = oc.openCurriculumNum
) THEN 'O'

```

    ELSE 'X'
END AS "개설과목등록여부",
(SELECT count(DISTINCT rs.studentNum)
FROM tblRegisterStudent rs
WHERE rs.openCurriculumNum = oc.openCurriculumNum
AND rs.completionCategoryNum = 1 ) || '명' AS "교육생 등록 인원"
FROM tblOpenCurriculum oc
JOIN tblCurriculum c ON oc.curriculumNum = c.curriculumNum
ORDER BY oc.openCurriculumNum;

SELECT * FROM vwOpenCurriculum; -- 뷰 확인용 select문

```

요구사항번호	B-5(관리자)	작성일	2025-08-06
요구사항명	개설 과목 관리	작성자	문예빈
View			
CREATE OR REPLACE VIEW vwOpenSubject --뷰 정의문			
AS			
SELECT			
c.curriculumName AS 과정명, to_char(oc.openCurriStartDate, 'YYYY-MM-DD') ' ~ ' to_char(oc.openCurriEndDate, 'YYYY-MM-DD') AS 과정기간, oc.classroomNum '호실' AS 강의실, s.subjectName AS 과목명,			

```

to_char(os.openSubjectStartDay, 'YYYY-MM-DD') || ' ~ ' ||
        to_char(os.openSubjectEndDay, 'YYYY-MM-DD') AS 과목기간,
        listagg(b.bookTitle, ', ') WITHIN GROUP (ORDER BY b.bookTitle) AS 교재명,
        t.teacherName AS 교사명
FROM tblOpenSubject os
        INNER JOIN tblOpenSOC osoc ON os.openSubjectNum = osoc.openSubjectNum
        INNER JOIN tblOpenCurriculum oc ON osoc.openCurriculumNum = oc.openCurriculumNum
        INNER JOIN tblCurriculum c ON oc.curriculumNum = c.curriculumNum
        INNER JOIN tblSubject s ON os.subjectNum = s.subjectNum
        INNER JOIN tblBook b ON os.openSubjectNum = b.openSubjectNum
        INNER JOIN tblTeacher t ON oc.teacherNum = t.teacherNum
GROUP BY
        c.curriculumName,
        oc.openCurriStartDate,
        oc.openCurriEndDate,
        oc.classroomNum,
        s.subjectName,
        os.openSubjectStartDay,
        os.openSubjectEndDay,
        t.teacherName
        ORDER BY oc.openCurriStartDate;

SELECT * FROM vwOpenSubject; -- 뷰 확인용 select문

```

요구사항번호	B-6(관리자)	작성일	2025-08-06
요구사항명	교육생 관리	작성자	문예빈

View

```
CREATE OR REPLACE VIEW vwStudent -- 뷰 정의문
```

AS

SELECT

```
s.studentName AS 이름,  
s.sResidentNum AS 주민번호뒷자리,  
s.sPhoneNum AS 전화번호,  
to_char(s.dateOfRegistration, 'YYYY-MM-DD') AS 등록일,
```

CASE

```
WHEN rs.studentNum IS NULL THEN '0'  
ELSE '1'
```

```
END || '회' AS 수강신청횟수,
```

```
jkc.jobKindCategoryName AS 취업여부
```

FROM tblStudent s

```
LEFT OUTER JOIN tblRegisterStudent rs ON s.studentNum = rs.studentNum  
LEFT OUTER JOIN tblJobStatus js ON rs.registerStudentNum = js.registerStudentNum  
LEFT OUTER JOIN tblJobKindCategory jkc ON js.jobKindCategoryNum = jkc.jobKindCategoryNum;
```

```
SELECT * FROM vwStudent; -- 뷰 확인용 select문
```

요구사항번호	B-7(관리자)	작성일	2025-08-06
요구사항명	시험 관리 및 성적 조회	작성자	문예빈

View

```
/* 개설 과정별 성적 정보 출력 */
CREATE OR REPLACE VIEW vwOpenSubjectScore -- 뷰 정의문
AS
SELECT
    c.curriculumName AS 개설과정명,
    to_char(oc.openCurriStartDate, 'YYYY-MM-DD') || ' ~ ' ||
        to_char(oc.openCurriEndDate, 'YYYY-MM-DD') AS 과정기간,
    oc.classroomNum || '호실' AS 강의실명,
    LISTAGG(s.subjectName, ', ') WITHIN GROUP (ORDER BY s.subjectName) AS 개설과목명,
    t.teacherName AS 교사명,
    LISTAGG(b.bookTitle, ', ') WITHIN GROUP (ORDER BY b.bookTitle) AS 교재명,
    st.studentName AS 교육생이름,
    st.sResidentNum AS 주민번호뒷자리,
    sc.writingScore AS 필기성적,
    sc.practicalScore AS 실기성적
FROM tblOpenCurriculum oc
    INNER JOIN tblCurriculum c ON oc.curriculumNum = c.curriculumNum
    INNER JOIN tblopenSOC osoc ON oc.openCurriculumNum = osoc.openCurriculumNum
    INNER JOIN tblopenSubject os ON osoc.openSubjectNum = os.openSubjectNum
    INNER JOIN tblSubject s ON os.subjectNum = s.subjectNum
    INNER JOIN tblteacher t ON oc.teacherNum = t.teacherNum
```

```
        INNER JOIN tblBook b ON os.openSubjectNum = b.openSubjectNum
        INNER JOIN tblRegisterStudent rs ON oc.openCurriculumNum = rs.openCurriculumNum
        INNER JOIN tblStudent st ON rs.studentNum = st.studentNum
        INNER JOIN tblScore sc ON rs.registerStudentNum = sc.registerStudentNum
    GROUP BY
        c.curriculumName,
        oc.openCurriStartDate,
        oc.openCurriEndDate,
        oc.classroomNum,
        t.teacherName,
        st.studentName,
        st.sResidentNum,
        sc.writingScore,
        sc.practicalScore
    ORDER BY oc.openCurriStartDate;
```

```
SELECT * FROM vwOpenSubjectScore; -- 뷰 확인용 select문
```

```
/* 개인별 성적 정보 출력 */
CREATE OR REPLACE VIEW vwStudentScore -- 뷰 정의문
AS
SELECT
    st.studentName AS 교육생이름,
    st.sResidentNum AS 주민번호뒷자리,
    c.curriculumName AS 개설과정명,
    to_char(oc.openCurriStartDate, 'YYYY-MM-DD') || ' ~ ' ||
    to_char(oc.openCurriEndDate, 'YYYY-MM-DD')
```

```
        to_char(oc.openCurriEndDate, 'YYYY-MM-DD') AS 과정기간,
        oc.classroomNum || '호실' AS 강의실명,
        LISTAGG(s.subjectName, ', ') WITHIN GROUP (ORDER BY s.subjectName) AS 개설과목명,
        t.teacherName AS 교사명,
        sc.attendanceScore AS 출결성적,
        sc.writingScore AS 필기성적,
        sc.practicalScore AS 실기성적
FROM tblOpenCurriculum oc
    INNER JOIN tblCurriculum c ON oc.curriculumNum = c.curriculumNum
    INNER JOIN tblopenSOC osoc ON oc.openCurriculumNum = osoc.openCurriculumNum
    INNER JOIN tblopenSubject os ON osoc.openSubjectNum = os.openSubjectNum
    INNER JOIN tblSubject s ON os.subjectNum = s.subjectNum
    INNER JOIN tblteacher t ON oc.teacherNum = t.teacherNum
    INNER JOIN tblRegisterStudent rs ON oc.openCurriculumNum = rs.openCurriculumNum
    INNER JOIN tblStudent st ON rs.studentNum = st.studentNum
    INNER JOIN tblScore sc ON rs.registerStudentNum = sc.registerStudentNum
GROUP BY
    st.studentName,
    st.sResidentNum,
    c.curriculumName,
    oc.openCurriStartDate,
    oc.openCurriEndDate,
    oc.classroomNum,
    t.teacherName,
    sc.attendanceScore,
```

```

sc.writingScore,
sc.practicalScore
ORDER BY oc.openCurriStartDate;

SELECT * FROM vwStudentScore; -- 뷰 확인용 select문

```

요구사항번호	B-10(관리자)	작성일	2025-08-06
요구사항명	개근 학생 포상	작성자	홍선민

View

--개근학생 확인 뷰 정의문

```

CREATE OR REPLACE VIEW vwCheckPerfectAttendance as
SELECT
    s.studentNum AS 학생번호, s.studentName AS 학생이름
FROM tblStudent s
INNER JOIN (
    SELECT registerStudentNum
    FROM tblAttendance
    GROUP BY registerStudentNum
    HAVING MIN(attendanceCategoryNum) = 4
        AND MAX(attendanceCategoryNum) = 4
) a ON s.studentNum = a.registerStudentNum
ORDER BY s.studentNum;

```

```

SELECT * FROM vwCheckPerfectAttendance; --뷰 확인용 select문

```

요구사항번호	B-11(관리자)	작성일	2025-08-06
요구사항명	취업률 조회	작성자	홍선민

View

--개설과정별 취업률 조회 뷰 정의

```

CREATE OR REPLACE VIEW vwEmployRateByCurr AS
SELECT
    oc.OPENCURRICULUMNUM AS 과정번호,
    curr.curriculumname AS 과정명,
    ROUND((COUNT(
        CASE
            WHEN js.jobKindCategoryNum = 1 THEN 1 --취업상태: 취업
            WHEN js.JOBKINDCATEGORYNUM = 3 THEN 1 --취업상태: 퇴사
        END) * 100.0) / COUNT(*), 2) AS "과정별 취업률"
FROM TBLREGISTERSTUDENT rs --수강생 테이블
    INNER JOIN TBLOPENCURRICULUM oc --개설과정 테이블
        ON rs.openCurriculumNum = oc.openCurriculumNum
    INNER JOIN tblJobStatus js --취업상태 테이블
        ON rs.registerStudentNum = js.registerStudentNum
    INNER JOIN TBLCURRICULUM curr
        ON curr.CURRICULUMNUM = oc.CURRICULUMNUM
WHERE rs.completionCategoryNum = 3
    AND js.JOBKINDCATEGORYNUM IN (1, 2, 3)
GROUP BY oc.OPENCURRICULUMNUM, curr.CURRICULUMNAME

```

```
ORDER BY oc.OPENCURRICULUMNUM;
```

--뷰 확인용 select문

```
SELECT * FROM vwEmployRateByCurr;
```

--수료년도별 취업률 조회 뷰 정의문

```
CREATE OR REPLACE VIEW vwEmployRateByYear AS
```

```
SELECT
```

```
    to_char(EXTRACT(YEAR FROM oc.OPENCURRIENDDATE)) AS 연도,
```

```
    ROUND((COUNT(
```

```
        CASE
```

```
            WHEN js.jobKindCategoryNum = 1 THEN 1 --취업상태: 취업
```

```
            WHEN js.JOBKINDCATEGORYNUM = 3 THEN 1 --취업상태: 퇴사
```

```
        END) * 100.0) / COUNT(*), 2) AS "연도별 취업률"
```

```
FROM TBLREGISTERSTUDENT rs
```

```
    INNER JOIN tblJobStatus js --취업상태 테이블
```

```
        ON rs.registerStudentNum = js.registerStudentNum
```

```
    INNER JOIN TBLOPENCURRICULUM oc --개설과정 테이블
```

```
        ON rs.openCurriculumNum = oc.openCurriculumNum
```

```
WHERE rs.completionCategoryNum = 3
```

```
    AND js.JOBKINDCATEGORYNUM IN (1, 2, 3)
```

```
GROUP BY EXTRACT(YEAR FROM oc.OPENCURRIENDDATE);
```

--뷰 확인용 select문

```
SELECT * FROM vwEmployRateByYear;
```

2. 교사 기능

요구사항번호	C-02(교사)	작성일	2025-08-06
요구사항명	강의 스케줄 조회	작성자	홍선민
View			
<p>--강의 스케줄 조회 메인 뷰 정의</p> <pre>CREATE OR REPLACE VIEW vwTeacherSchedule AS SELECT oc.openCurriculumNum AS 과정번호, tc.curriculumName AS 과정명, tt.teacherName AS 교사명, oc.openCurriStartDate AS 과정시작날짜, oc.openCurriEndDate AS 과정종료날짜, cr.classroomNum AS 강의실, ts.subjectName AS 과목명, os.openSubjectStartDay AS 과목시작날짜, os.openSubjectEndDay AS 과목종료날짜, tb.bookTitle AS 교재명, cr.classCapacity AS 강의실수용인원, rs.REGCOUNT AS 과정별등록인원, CASE WHEN os.openSubjectStartDay > SYSDATE THEN '강의예정' WHEN os.openSubjectEndDay < SYSDATE THEN '강의종료' ELSE '강의중' END AS 강의상태</pre>			

```
FROM tblOpenCurriculum oc
    INNER JOIN tblCurriculum tc ON oc.curriculumNum = tc.curriculumNum
    INNER JOIN tblClassroom cr ON oc.classroomNum = cr.classroomNum
    INNER JOIN tblOpenSOC soc ON oc.openCurriculumNum = soc.openCurriculumNum
    INNER JOIN tblOpenSubject os ON soc.openSubjectNum = os.openSubjectNum
    INNER JOIN tblSubject ts ON os.subjectNum = ts.subjectNum
    INNER JOIN tblBook tb ON os.openSubjectNum = tb.openSubjectNum
    INNER JOIN tblTeacher tt ON oc.teacherNum = tt.teacherNum
    LEFT OUTER JOIN (
        SELECT
            OPENCURRICULUMNUM,
            count(*) AS REGCOUNT
        FROM TBLREGISTERSTUDENT
        WHERE COMPLETIONCATEGORYNUM = 1
        GROUP BY OPENCURRICULUMNUM
        ORDER BY OPENCURRICULUMNUM ASC
    ) rs ON oc.OPENCURRICULUMNUM = rs.OPENCURRICULUMNUM;
```

--뷰 확인용 select문

```
SELECT * FROM vwTeacherSchedule;
```

--과정에 대한 정보만을 보여주는 뷰 정의문

```
CREATE OR REPLACE VIEW vwShowCurrInfo as
```

```
SELECT
    DISTINCT 과정번호,
    과정명,
```

```
교사명,  
과정시작날짜,  
과정종료날짜,  
과정별등록인원  
FROM VWTEACHERSCHEDULE  
ORDER BY 과정번호 ASC;
```

```
SELECT * FROM vwshowcurrInfo; --뷰 확인용 select문
```

--교육중인 과정에 등록한 교육생 정보(이름, 전화번호, 회원가입일, 수료상태) 확인 뷰 정의

```
CREATE OR REPLACE VIEW vwShowRegStudentInfo AS  
SELECT  
    rs.OPENCURRICULUMNUM          AS 과정번호,  
    cur.CURRICULUMNAME            AS 과정명,  
    s.studentName                  AS 이름,  
    s.sPhoneNum                   AS 전화번호,  
    cc.COMPLETIONCATEGORYNAME    AS 수료상태,  
    to_char(s.dateOfRegistration, 'yyyy-MM-dd') AS 회원가입일  
FROM tblStudent s  
LEFT OUTER JOIN tblRegisterStudent rs ON s.studentNum = rs.studentNum  
INNER JOIN TBLCOMPLETIONCATEGORY cc ON rs.COMPLETIONCATEGORYNUM = cc.COMPLETIONCATEGORYNUM  
INNER JOIN TBLOPENCURRICULUM oc ON rs.OPENCURRICULUMNUM = oc.OPENCURRICULUMNUM  
INNER JOIN TBLCURRICULUM cur ON oc.CURRICULUMNUM = cur.CURRICULUMNUM  
WHERE rs.COMPLETIONCATEGORYNUM IN (1, 2)  
    AND oc.OPENCURRIENDDATE >= sysdate;  
SELECT * FROM vwShowRegStudentInfo; --뷰 확인용 select문
```

요구사항번호	C-03(교사)	작성일	2025-08-06
요구사항명	배점 입출력	작성자	홍선민
View			
--시험문제별 배점 출력 뷰			
<pre>CREATE OR REPLACE VIEW vwShowTestQuestionInfo AS SELECT CASE WHEN testCategoryNum = 1 THEN '필기' WHEN testCategoryNum = 2 THEN '실기' END AS 문제유형, examQuestion AS 문제내용, score AS 배점 FROM tblTestQuestion; -- WHERE TESTQUESTIONNUM = 1; --문제 번호 입력 SELECT * FROM vwshowtestQuestionInfo; --뷰 확인용 select문</pre>			

요구사항번호	C-04(교사)	작성일	2025-08-06
요구사항명	성적 입출력	작성자	박진혁
View			
<pre>/*교육생 전체에 대한 성적 등록 여부 및 중도 포기 여부 뷰*/ CREATE OR REPLACE VIEW vw_student_dropout_status AS SELECT ts.studentName, CASE WHEN sc.registerStudentNum IS NULL THEN 'X' ELSE 'O' END AS 성적등록여부, CASE WHEN rs.completionCategoryNum = 2 THEN 'O' ELSE 'X' END AS 중도포기여부 FROM tblStudent ts FULL OUTER JOIN tblRegisterStudent rs ON ts.studentNum = rs.studentNum FULL OUTER JOIN tblScore sc ON rs.registerStudentNum = sc.registerStudentNum; SELECT * FROM vw_student_score_dropout_status; --뷰 확인용 select문</pre>			

요구사항번호	C-07(교사)	작성일	2025-08-06
요구사항명	Q&A 게시판 답변	작성자	박진혁

View

/*질문&답변 출력 뷰*/

```
CREATE OR REPLACE VIEW vwQuestionAnswerDetails AS
SELECT
    tc.curriculumName AS 과정명,
    tq.questionNum AS 질문번호,
    ts.studentName AS 학생이름,
    tq.questionTitle AS 제목,
    tq.questionDetail AS 질문내용,
    tt.teacherNum AS 교사번호,
    tt.teacherName AS 교사이름,
    ta.answerDetail AS 답변내용
FROM tblTeacher tt
    INNER JOIN tblOpenCurriculum oc ON tt.teacherNum = oc.teacherNum
    INNER JOIN tblRegisterStudent rs ON oc.openCurriculumNum = rs.openCurriculumNum
    INNER JOIN tblQuestion tq ON rs.registerStudentNum = tq.registerStudentNum
    INNER JOIN tblStudent ts ON rs.studentNum = ts.studentNum
    INNER JOIN tblCurriculum tc ON oc.curriculumNum = tc.curriculumNum
    FULL OUTER JOIN tblAnswer ta ON tq.questionNum = ta.questionNum;
```

SELECT * FROM vw_QuestionAnswerDetails ORDER BY 질문번호 ASC; --뷰 확인용 select문

3. 교육생 기능

요구사항번호	D-2(교육생)	작성일	2025-08-06
요구사항명	수강 신청 기능	작성자	최석용

View

--전체 과정 조회 view

```
CREATE OR REPLACE VIEW vwallcurr AS
SELECT CURRICULUMNAME AS 과정명
FROM TBLCURRICULUM;
```

--뷰 확인용 select문

```
SELECT * FROM vwallcurr;
```

--인기 과정 조회 view

```
CREATE OR REPLACE VIEW vwpopularcurr AS
SELECT
    c.curriculumName AS "과정명",
    ROUND(AVG(r.reviewRating), 2) AS "평균 별점"
FROM
    tblReview r
INNER JOIN
    tblOpenCurriculum oc ON r.openCurriculumNum = oc.openCurriculumNum
    INNER JOIN
        tblCurriculum c ON oc.curriculumNum = c.curriculumNum
GROUP BY
```

```
c.curriculumName  
ORDER BY  
"평균 별점" DESC;
```

--뷰 확인용 select문

```
SELECT * FROM vwpopularcurr;
```

요구사항번호	D-5(교육생)	작성일	2025-08-06
요구사항명	수강 후 기능	작성자	최석용

View

--전체 수강 후기 조회 view

```
CREATE OR REPLACE VIEW vwreviewall as  
SELECT  
stu.studentname AS 이름,  
r.REVIEWRATING AS 평점,  
r.REVIEWDETAIL AS 리뷰내용  
FROM tblReview r  
INNER JOIN TBLSTUDENT stu  
ON r.REGISTERSTUDENTNUM = stu.STUDENTNUM;
```

--뷰 확인용 select문

```
SELECT*FROM TBLREGISTERSTUDENT;
```

요구사항번호	D-6(교육생)	작성일	2025-08-06
요구사항명	Q&A 게시판	작성자	최석용

View

--학생들이 질문했던 내용과 답변 view

```
CREATE OR REPLACE VIEW vwqueall as
SELECT
    stu.studentname AS "이름",
    q.questiontitle AS "질문제목",
    q.questiondetail AS "질문내용",
    a.answerdetail AS "답변내용"
FROM
    TBLStudent stu
        INNER JOIN TBLREGISTERSTUDENT r
            ON stu.studentNum = r.studentNum
        INNER JOIN TBLQUESTION q
            ON r.registerStudentNum = q.registerStudentNum
        INNER JOIN TBLANSWER a
            ON q.questionNum = a.questionNum
```

--뷰 확인용 select문

```
SELECT * FROM vwqueall;
```