

업무 쿼리문(ANSI-SQL)

2조 박진혁, 문예빈, 최석용, 홍선민

작성일자: 25.08.04 - 25.08.05

목 차

분류	페이지
관리자 기능	3
교사 기능	27
교육생 기능	39

B. 관리자

요구사항번호	B-2(관리자)	작성일	2025-08-04
요구사항명	기초 정보 관리	작성자	문예빈
DML			
<pre>/* 과정명 관리 */ --1.입력 INSERT INTO tblCurriculum(curriculumNum, curriculumName) VALUES (seqCurriculum.NEXTVAL, '실무 중심 풀스택 SW 개발자 양성과정'); --2. 출력 SELECT DISTINCT curriculumName AS 과정명 FROM tblCurriculum; --3. 수정 UPDATE tblCurriculum SET curriculumName = 'AI 융합 웹서비스 개발자 양성과정' WHERE curriculumNum = 13; --4. 삭제 DELETE FROM tblCurriculum WHERE curriculumNum = 13; /* 과목명 관리 */ --1.입력</pre>			

```
INSERT INTO tblSubject(subjectNum, subjectName, progressPeriod)
VALUES ((SELECT max(subjectNum) FROM tblSubject) + 1, '유니티', 15);
```

--2. 출력

```
SELECT
    subjectName AS 과목명
FROM tblSubject;
```

--3. 수정

```
UPDATE tblSubject SET subjectName = 'Elastic Search'
WHERE subjectNum = 21;
```

--4. 삭제

```
DELETE FROM tblSubject
WHERE subjectNum = 21;
```

/* 강의실명(정원포함) 관리 */

--1.입력

```
INSERT INTO tblClassroom(classroomNum, classCapacity)
VALUES ((SELECT max(classroomNum) FROM tblClassroom) + 1, 30);
```

--2. 출력

```
SELECT
    classroomNum AS 강의실명,
    classCapacity AS 강의실정원
FROM tblClassroom;
```

--3. 수정(강의실정원)

```
UPDATE tblclassroom SET classCapacity = 26
  WHERE classroomNum = 7;
```

--4. 삭제

```
DELETE FROM tblClassroom
  WHERE classroomNum = 7;
```

/* 교재 관리 */

--1.입력

```
INSERT INTO tblBook(bookNum, bookTitle, author, publisher, openSubjectNum)
  VALUES((SELECT max(bookNum) FROM tblBook) + 1, '모두의 파이썬', '이승천', '길벗(주)', 2);
```

--2. 출력

```
SELECT
  bookTitle AS 제목,
  author AS 저자,
  publisher AS 출판사
FROM tblBook;
```

--3-1. 교재명 수정

```
UPDATE tblBook SET bookTitle = '모두의 파이썬 개정2판'
  WHERE bookNum = 42;
```

--3-2. 저자 수정

```
UPDATE tblBook SET author = '이승찬'
  WHERE bookNum = 42;
```

--3-3. 출판사 수정

```
UPDATE tblBook SET publisher = '길벗'
WHERE bookNum = 42;
```

--4. 삭제

```
DELETE FROM tblBook
WHERE bookNum = 42;
```

요구사항번호	B-3(관리자)	작성일	2025-08-04
요구사항명	교사 계정 관리	작성자	문예빈
DML			
/*교사 기본 정보*/			
--1. 입력			
INSERT INTO tblTeacher (teacherNum, teacherName, teacherID, tResidentNum, tPhoneNum)			
VALUES (tblTeacher_seq.nextval, '문예빈', 'moon123', '4894561', '010-7894-4561');			
-- 2. 출력(교사 전체 명단)			
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;
```

```
-- 3-1. 이름 수정
```

```
UPDATE tblTeacher SET teacherName = '김예빈'
```

```
WHERE teacherNum = 11;
```

```
-- 3-2. 주민번호 수정
```

```
UPDATE tblTeacher SET tResidentNum = '4876543'
```

```
WHERE teacherNum = 11;
```

```
-- 3-3. 연락처 수정
```

```
UPDATE tblTeacher SET tPhoneNum = '010-7891-2345'
```

```
WHERE teacherNum = 11;
```

```
-- 4. 삭제
```

```
DELETE tblTeacher WHERE teacherNum = 11;
```

```
/*교사 강의 가능 과목 정보*/
```

```
-- 1. 입력
```

```
INSERT INTO tblPossibleSubject(possiblesubjectNum, teacherNum, subjectNum)
```

```

VALUES(SEQ_PS.NEXTVAL, 11, (SELECT subjectNum FROM tblSubject WHERE subjectName = 'Java'));

-- 2. 출력
SELECT
    t.teacherName AS 교사명,
    listagg(s.subjectName, ', ') WITHIN GROUP (ORDER BY s.subjectName) AS 과목명
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;

-- 3. 수정
UPDATE tblPossibleSubject SET
    subjectNum = (SELECT subjectNum FROM tblSubject WHERE subjectName = 'SQL')
    WHERE teacherNum = 11
    AND subjectNum = (SELECT subjectNum FROM tblSubject WHERE subjectName = 'Java');

-- 4. 삭제
DELETE tblPossibleSubject
    WHERE possibleSubjectNum = 78;

/* 특정 교사 정보 출력*/
SELECT
    s.subjectName AS "배정된 개설 과목명",
    to_char(os.openSubjectStartDay, 'YYYY-MM-DD')

```



```

        || ' ~ ' || to_char(os.openSubjectEndDay, 'YYYY-MM-DD') AS "개설 과목 기간",
        c.curriculumName AS "과정명",
        to_char(oc.openCurriStartDate, 'YYYY-MM-DD')
        || ' ~ ' || to_char(oc.openCurriEndDate, 'YYYY-MM-DD') AS "개설 과정 기간",
        listagg(b.bookTitle, ', ') WITHIN GROUP (ORDER BY b.bookTitle) AS "교재명",
        oc.classroomNum || '호실' AS "강의실",
        CASE
            WHEN os.openSubjectEndDay < sysdate THEN '강의종료'
            WHEN os.openSubjectStartDate > sysdate THEN '강의예정'
            ELSE '강의중'
        END AS "강의 진행여부"
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 = 1
GROUP BY
    s.subjectName,
    os.openSubjectStartDate,
    os.openSubjectEndDay,
    c.curriculumName,
    oc.openCurriStartDate,
    oc.openCurriEndDate,
    oc.classroomNum,
    os.openSubjectEndDay,

```

```
os.openSubjectStartDay
ORDER BY openSubjectStartDay;
```

요구사항번호	B-4(관리자)	작성일	2025-08-04
요구사항명	개설 과정 관리	작성자	문예빈

DML

```
/*개설 과정 정보 > 과정명, 과정 기간(시작 년월일, 끝 년월일), 강의실 정보, 개설 과목 등록 여부, 교육생 등록인원 */
-- 1. 입력
INSERT INTO tblOpenCurriculum(openCurriculumNum, openCurriStartDate, openCurriEndDate, teacherNum, classroomNum, curriculumNum)
VALUES(SEQ_OC.NEXTVAL, to_date('2025-08-04', 'YYYY-MM-DD'), to_date('2026-01-30', 'YYYY-MM-DD'), 7, 1, 14);

-- 2. 출력
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;

```

-- 3-1. 과정 기간 수정

```

UPDATE tblOpenCurriculum SET openCurriStartDate = TO_DATE('2025-08-04', 'YYYY-MM-DD'),
                             openCurriEndDate = TO_DATE('2026-01-30', 'YYYY-MM-DD')
    WHERE openCurriculumNum = 6;

```

-- 3-2. 강의실 정보 수정

```

UPDATE tblOpenCurriculum SET classroomNum = 1
    WHERE openCurriculumNum = 6;

```

-- 3-3. 담당 교사 수정

```

UPDATE tblOpenCurriculum SET teacherNum = 6 WHERE openCurriculumNum = 1;

```

-- 4.1 강의실 삭제

```
DELETE tblClassroom
```

```
WHERE classroomNum = 1;
```

– 4.2 개설 과정 삭제

```
DELETE tblOpenCurriculum
```

```
WHERE openCurriculumNum = 1;
```

```
/* 특정 개설 과정이 수료한 경우 등록된 교육생 전체에 대해서 수료상태를 지정(중도탈락자 제외) */
```

```
UPDATE tblRegisterStudent SET completionCategoryNum = 3
```

```
WHERE openCurriculumNum = 1 AND completionCategoryNum = 1;
```

```
/* 특정 개설 과정 정보 과목명, 과목기간(시작 년월일, 끝년월일), 교재명, 교사명) 및 등록된 교육생 정보(교육생 이름, 주민번호 뒷자리, 등록일, 수료 및  
중도 탈락) 조회 */
```

```
SELECT
```

```
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,
```

```
st.studentName AS 이름,
```

```
st.sResidentNum AS 주민번호뒷자리,
```

```
st.sPhoneNum AS 전화번호,
```

```
to_char(st.dateOfRegistration, 'YYYY-MM-DD') AS 등록일,
```

```
cc.completionCategoryName AS 수료상태
```

```
FROM tblOpenCurriculum oc
```

```
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 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 = 7
GROUP BY
    s.subjectName,
    os.openSubjectStartDay,
    os.openSubjectEndDay,
    t.teacherName,
    st.studentName,
    st.sResidentNum,
    st.sPhoneNum,
    st.dateOfRegistration,
    cc.completionCategoryName
    ORDER BY st.studentName;
```

요구사항번호	B-5(관리자)	작성일	2025-08-05
요구사항명	개설 과목 관리	작성자	문예빈
DML			
<pre> /*개설 과목 정보*/ -- 1. 입력 INSERT INTO tblOpenSubject(openSubjectNum, openSubjectStartDay, openSubjectEndDay, testDate, SubjectNum) VALUES (SEQ_OS.NEXTVAL, TO_DATE('2025-08-06', 'YYYY-MM-DD'), TO_DATE('2025-09-19', 'YYYY-MM-DD'), TO_DATE('2025-09-19', 'YYYY-MM-DD'), 20); -- 2. 개설 과정 정보(과정명, 과정기간(시작 년월일, 끝 년월일), 강의실)와 과목명, 과목기간(시작 년월일, 끝 년월일), 교재명, 교사명 출력 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 </pre>			

```

        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;

-- 3-1. 과목 기간 수정
UPDATE tblOpenSubject SET openSubjectStartDay = TO_DATE('2025-08-05', 'YYYY-MM-DD'),
                        openSubjectEndDay = TO_DATE('2025-08-20', 'YYYY-MM-DD')
WHERE openSubjectNum = 1;

-- 3-2. 교재 연관 과목 수정
UPDATE tblBook SET openSubjectNum = 10 WHERE bookNum = 1;

-- 4-1. 개설과목 정보 삭제
DELETE tblOpenSubject WHERE openSubjectNum = 1;

-- 4-2. 과목별 교재 정보 삭제
DELETE tblBook WHERE openSubjectNum = 1;

```

```

/* 특정 개설 과정 선택 시 개설 과목 정보 출력*/
SELECT
    c.curriculumName AS 과정명,
    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 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 = 7
GROUP BY
    c.curriculumName,
    s.subjectName,
    os.openSubjectStartDay,
    os.openSubjectEndDay,
    t.teacherName;

```


요구사항번호	B-6(관리자)	작성일	2025-08-05
요구사항명	교육생 관리	작성자	문예빈
DML			
<pre> /*교육생 정보*/ -- 1. 교육생 등록 INSERT INTO tblStudent(studentNum, studentName, studentID, sResidentNum, sPhoneNum, dateOfRegistration) VALUES (tblStudent_seq.nextval, '문예빈', 'moon123', '1234567', '010-1111-2222', sysdate); -- 2. 교육생 정보 출력 : 교육생 이름, 주민번호 뒷자리, 전화번호, 등록일, 수강(신청) 횟수, 취업여부를 출력 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; </pre>			

/*특정 교육생 선택시 교육생이 수강 신청한 또는 수강중인 수강했던 개설 과정 정보(과정명, 과정기간(시작 년월일, 끝 년월일), 강의실, 수료 및 중도 탈락의 여부 출력*/

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,
cc.completionCategoryName AS 수료상태

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 = 200;

요구사항번호	B-7(관리자)	작성일	2025-08-05
요구사항명	시험 관리 및 성적 조회	작성자	문예빈
DML			
/*특정 개설 과정을 선택 > 개설된 과목 정보를 출력하고, 개설 과목 별로 성적 등록 여부를 조회*/			
SELECT			
s.subjectName AS 과목명,			
to_char(os.openSubjectStartDay, 'YYYY-MM-DD') ' ~ '			
to_char(os.openSubjectEndDay, 'YYYY-MM-DD') AS 과목기간,			
to_char(os.testDate, 'YYYY-MM-DD') AS 시험날짜,			

```

CASE
    WHEN max(sc.attendanceScore) IS NULL THEN 'X'
    ELSE 'O'
END AS 출결성적등록여부,
CASE
    WHEN max(sc.writingScore) IS NULL THEN 'X'
    ELSE 'O'
END AS 필기성적등록여부,
CASE
    WHEN max(sc.practicalScore) IS NULL THEN 'X'
    ELSE 'O'
END AS 실기성적등록여부
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 = 4
GROUP BY
    s.subjectName,
    os.openSubjectStartDay,
    os.openSubjectEndDay,
    os.testDate
    ORDER BY os.openSubjectStartDay;

```

/*개설 과정별 성적 정보 출력 > 개설 과정, 과정기간, 강의실명, 개설 과목명, 교사명, 교재명, /교육생 이름, 주민번호 뒷자리, 필기, 실기*/

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;

```

/*개인별 성적 정보 출력 > 교육생 이름, 주민번호 뒷자리, 개설과정명, 과정기간, 강의실명/, 개설과목명, 과목기간, 교사명, 출결, 필기, 실기*/

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') 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;
```

요구사항번호	B-8(관리자)	작성일	2025-08-05
요구사항명	출결 관리 및 출결 조회	작성자	문예빈
DML			
/*특정 개설 과정을 선택 시 모든 교육생의 출결을 조회 + 근태 상황 / 출결 현황을 기간별(년, 월, 일) 조회*/			
SELECT			
st.studentName AS 이름,			
to_char(a.attendanceDate, 'YYYY-MM-DD') AS 날짜,			

```

        ac.attendanceCategoryName AS 근태상황
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 = 7
    AND (:year IS NULL OR EXTRACT(YEAR FROM a.attendanceDate) = :year)
    AND (:month IS NULL OR EXTRACT(MONTH FROM a.attendanceDate) = :month)
    AND (:day IS NULL OR EXTRACT(DAY FROM a.attendanceDate) = :day)
    ORDER BY a.attendanceDate;

```

요구사항번호	B-9(관리자)	작성일	2025-08-05
요구사항명	사물함 배정	작성자	홍선민
DML			
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
);

```

요구사항번호	B-10(관리자)	작성일	2025-08-05
요구사항명	개근 학생 포상	작성자	홍선민
DML			
<pre> 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; </pre>			

요구사항번호	B-11(관리자)	작성일	2025-08-05
요구사항명	취업률 조회	작성자	박진혁, 홍선민
DML			
--개설과정별 취업률 조회 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; --수료년도별 취업률 조회 SELECT			

```

to_char(EXTRACT(YEAR FROM oc.OPENCURRIENDDATE)) AS 년도,
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 "년도별 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);

SELECT * FROM vwEmployRateByYear;

```

C. 교사

요구사항번호	C-02(교사)	작성일	2025-08-04~2025-08-06
요구사항명	강의 스케줄 조회	작성자	홍선민
DML			
<pre> 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 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 </pre>			

```

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;

```

```

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;
```

요구사항번호	C-03(교사)	작성일	2025-08-04
요구사항명	배점 입출력	작성자	박진혁
DML			
<pre>--시험문제 및 배점 입력 INSERT INTO tblTestQuestion (testQuestionNum, examQuestion, score, testCategoryNum) VALUES (tblTestQuestion_seq.NEXTVAL, '문제내용', (배점점수 입력: number), (필기: 1 / 실기: 2)); --시험문제별 배점 출력 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</pre>			

요구사항번호	C-04(교사)	작성일	2025-08-04
요구사항명	성적 입출력	작성자	박진혁
DML			
<pre>--교사가 과정을 마친 과정 목록 출력 SELECT oc.openCurriculumNum AS 과정번호, tc.curriculumName AS 과정명, TO_CHAR(oc.openCurriStartDate, 'YYYY-MM-DD') ' ~ ' TO_CHAR(oc.openCurriEndDate, 'YYYY-MM-DD') AS 과정기간 FROM tblTeacher tt INNER JOIN tblOpenCurriculum oc ON tt.teacherNum = oc.teacherNum INNER JOIN tblCurriculum tc ON oc.curriculumNum = tc.curriculumNum WHERE oc.openCurriEndDate <= SYSDATE -- 종료된 과정만 출력 AND tt.teacherNum = 1; -- 교사번호 입력 --특정 과정번호 선택 시 교육생 정보 출력 SELECT oc.openCurriculumNum AS 과정번호, tc.curriculumName AS 과정명, rs.registerStudentNum AS 수강생번호, ts.studentName AS 이름 FROM tblTeacher tt INNER JOIN tblOpenCurriculum oc ON tt.teacherNum = oc.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</pre>			

```

WHERE oc.openCurriEndDate <= SYSDATE -- 종료된 과정만 출력
      AND tt.teacherNum = 1 -- 교사번호 입력
--      AND oc.openCurriculumNum = 1; --특정 과정번호 선택

--특정 교육생 정보 선택 시 교육생 시험 점수 입력(출결, 필기, 실기 구분)
INSERT INTO tblScore (ScoreNum, attendanceScore, writingScore, practicalScore, openSubjectNum, registerStudentNum)
      VALUES (tblScore_seq.NEXTVAL, 출결점수(NUMBER), 필기점수(NUMBER), 실기점수(NUMBER), 과정번호(NUMBER), 수강생번호(NUMBER));

--교육생 시험 점수가 등록되어있을 경우 수정
UPDATE tblScore
SET attendanceScore = 20,
      writingScore = 40,
      practicalScore = 40
WHERE openSubjectNum = 7
      AND registerStudentNum = 173;

--과정 목록 출력 시 과정번호, 과정명, 과정기간(시작 년월일, 끝 년월일), 강의실, 과목명, 과목기간(시작 년월일, 끝 년월일), 교재명, 출결, 필기, 실기 배점,
성적 등록 여부 등 출력
SELECT
      DISTINCT oc.openCurriculumNum AS 과정번호,
      tc.curriculumName AS 과정명,
      TO_CHAR(oc.openCurriStartDate, 'YYYY-MM-DD') || ' ~ ' || TO_CHAR(oc.openCurriEndDate, 'YYYY-MM-DD') AS 과정기간,
      oc.classroomNum AS 강의실,
      sb.subjectName AS 과목목록,
      TO_CHAR(os.openSubjectStartDay, 'YYYY-MM-DD') || ' ~ ' || TO_CHAR(os.openSubjectEndDay, 'YYYY-MM-DD') AS 과목기간,
      tb.bookTitle AS 교재목록,
      (100 * 0.2) AS "출결 배점",

```

```

(100 * 0.4) AS "필기 배점",
(100 * 0.4) AS "실기 배점"
FROM tblOpenCurriculum oc
    INNER JOIN tblRegisterStudent rs ON oc.openCurriculumNum = rs.openCurriculumNum
    INNER JOIN tblStudent ts ON rs.studentNum = ts.studentNum
    INNER JOIN tblCurriculum tc ON oc.curriculumNum = tc.curriculumNum
    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 tblBook tb ON os.openSubjectNum = tb.openSubjectNum
    INNER JOIN tblScore sc ON oc.openCurriculumNum = sc.openCurriculumNum
--WHERE oc.openCurriculumNum = 1; --특정 과목 선택
GROUP BY
    oc.openCurriculumNum, tc.curriculumName, oc.openCurriStartDate, oc.openCurriEndDate,
    oc.classroomNum, sb.subjectName, os.openSubjectStartDay, os.openSubjectEndDay, tb.bookTitle
ORDER BY 과정기간 ASC;

--특정 과정 선택 시 교육생 정보(교육생 이름, 전화번호, 수료 또는 중도포기) 및 성적 출력
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 -- 수료 조건
    OR rs.completionCategoryNum = 2; -- 중도포기 조건
--    AND oc.openCurriculumNum = 1; --특정 과정 선택

--교육생 전체에 대한 성적 등록 여부 및 중도 포기 여부
SELECT
    ts.studentName,
    CASE
        WHEN sc.registerStudentNum IS NULL THEN 'X'
        ELSE 'O'
    END 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;

요구사항번호	C-05(교사)	작성일	2025-08-04
요구사항명	출결 관리 및 출결 조회	작성자	박진혁
DML			
<pre>-- 자신이 강의한 과정에 한해 선택하는 경우 모든 교육생의 출결을 조회 SELECT tt.teacherName AS "교사 이름", tc.curriculumName AS 과정명, ts.studentName AS "교육생 이름", 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 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 teacherNum = 1; --교사번호 입력</pre>			

```
--      AND   oc.openCurriculumNum = 1; --과정번호 입력
```

요구사항번호	C-06(교사)	작성일	2025-08-04
요구사항명	수강 후기 조회	작성자	박진혁

DML

```
-- 수강 후기 및 평점 조회
```

```
SELECT
```

```
    tc.curriculumName,
```

```
    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 -- 종료된 과정만 출력
```

```
--      AND tt.teacherNum = 1; -- 교사번호 입력
```

요구사항번호	C-07(교사)	작성일	2025-08-04
요구사항명	Q&A 게시판 답변	작성자	박진혁
DML			
<pre> --교육생들이 올린 질문 --View 테이블 조회 SELECT * FROM vw_QuestionAnswerDetails ORDER BY 질문번호 ASC; --답변 등록 INSERT INTO tblAnswer (answerNum, answerDetail, questionNum, teacherNum) VALUES (answerNum_seq.nextVal, '답변내용 작성', 질문번호(NUMBER), 교사번호(NUMBER)); --답변 조회 SELECT questionNum AS 질문번호, answerNum AS 답변번호, answerDetail AS 답변내용 FROM tblAnswer WHERE questionNum = 1; --질문번호 --답변 수정 UPDATE tblAnswer SET answerDetail = '수정된 내용 작성' WHERE answerNum = 1; --답변번호 COMMIT; </pre>			

```
--답변 삭제
DELETE FROM tblAnswer
WHERE answerNum = 1; --답변번호
```

요구사항번호	C-08(교사)	작성일	2025-08-04
요구사항명	취업률 조회	작성자	박진혁
DML			
<pre>--취업률 조회 SELECT TO_CHAR(oc.openCurriEndDate, 'YYYY') AS 연도, rs.openCurriculumNum AS 과정번호, ROUND((COUNT(CASE WHEN js.jobKindCategoryNum = 1 THEN 1 END) * 100.0) / COUNT(*), 2) AS 취업율 FROM tblRegisterStudent rs INNER JOIN tblJobStatus js ON rs.registerStudentNum = js.registerStudentNum INNER JOIN tblOpenCurriculum oc ON rs.openCurriculumNum = oc.openCurriculumNum WHERE rs.completionCategoryNum = 3 AND js.jobKindCategoryNum IN (1, 2, 3) AND EXTRACT(YEAR FROM oc.openCurriEndDate) = '2025' --연도 입력 GROUP BY rs.openCurriculumNum, oc.openCurriEndDate HAVING rs.openCurriculumNum = 7; --개설과정번호 입력</pre>			

--

D. 교육생

요구사항번호	D-2 (교육생)	작성일	2025-08-04
요구사항명	수강 신청 및 인기 과정 조회	작성자	최석용
DML			
<p>1. 수강 신청 가능한 전체 과정을 조회</p> <pre>SELECT CURRICULUMNAME FROM TBLCURRICULUM;</pre> <p>2. 평균 평점이 높은 인기 과정 조회</p> <pre>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;</pre> <p>3. 교육생은 원하는 과정을 수강 신청</p> <pre>UPDATE TBLREGISTERSTUDENT SET COMPLETIONCATEGORYNUM = 1, OPENCURRICULUMNUM = 6 WHERE REGISTERSTUDENTNUM = 384;</pre>			

요구사항번호	D-3 (교육생)	작성일	2025-08-04
요구사항명	성적 조회	작성자	최석용
DML			
<p>1. 교육생이 로그인 후 교육생 개인정보와 교육생이 수강한 과정명, 과정기간(시작 년월일, 끝 년월일), 강의실, 사물함번호가 출력됨 > 개인정보는 이름, 아이디, 핸드폰 번호를 출력한다.</p> <pre> SELECT stu.studentname AS "이름", stu.studentid AS "아이디", stu.sphonenum AS "핸드폰 번호", cul.curriculumname AS "과정명", ocul.OPENCURRISTARTDATE AS "과정시작일", ocul.OPENCURRIENDDATE AS "과정종료일", cls.classroomnum AS "강의실번호", loc.lockernum AS 사물함번호 FROM tblstudent stu INNER JOIN tblregisterstudent rs ON stu.studentnum = rs.studentnum INNER JOIN tblopencurriculum ocul ON rs.opencurriculumnum = ocul.opencurriculumnum INNER JOIN tblcurriculum cul ON ocul.curriculumnum = cul.curriculumnum INNER JOIN tblclassroom cls ON ocul.classroomnum = cls.classroomnum INNER JOIN tbllocker loc ON rs.REGISTERSTUDENTNUM = loc.lockernum </pre>			


```
WHERE stu.STUDENTNUM = '2';
```

2. 성적 정보는 과정별로 목록 형태로 출력

> 과정번호, 과목명, 배점 정보(출결, 필기, 실기 배점) 성적 정보(출결, 필기, 실기 점수), 총점 출력

```
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 = 12;
```

2. 과목 별 교재 출력

```
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;
```

3. 과목별 시험 일자

> 과목명, 시험일자, 시험문제

```
SELECT
    sub.subjectname AS "과목명",
    osub.testdate AS "시험날짜",
    tque.examquestion AS "시험문제"
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
ORDER BY
    sub.subjectname,
    osub.testdate;
```

요구사항번호	D-4 (교육생)	작성일	2025-08-05
요구사항명	출결 관리 및 출결 조회	작성자	최석용
DML			
<pre> --1. 전체 출결 통계 SELECT stu.studentName AS 이름, ad.attendancedate as "전체 출결", CASE ad.attendanceCategoryNum WHEN 1 THEN '결석' WHEN 2 THEN '조퇴' 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.REGISTERSTUDENTNUM where ad.REGISTERSTUDENTNUM = 11 ORDER BY ad.attendancedate asc; --2. 월별 조회 select stu.studentName AS 이름, ad.attendancedate as "4월", CASE ad.attendanceCategoryNum </pre>			

```

        WHEN 1 THEN '결석'
        WHEN 2 THEN '조퇴'
        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.REGISTERSTUDENTNUM
    WHERE TO_CHAR(ad.attendancedate, 'MM') = '04' and ad.REGISTERSTUDENTNUM = 11;

```

--3. 일별 조회

```

select
    stu.studentName AS 이름,
    ad.attendancedate as "날짜",
    CASE ad.attendanceCategoryNum
        WHEN 1 THEN '결석'
        WHEN 2 THEN '조퇴'
        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.REGISTERSTUDENTNUM
    WHERE TRUNC(ad.attendancedate) = TO_DATE('2025-04-25', 'YYYY-MM-DD') and ad.REGISTERSTUDENTNUM = 11;

```

요구사항번호	D-5(교육생)	작성일	2025-08-05
요구사항명	수강 완료	작성자	최석용
DML			
1. 수강 후기 INSERT INTO tblReview (reviewNum, reviewRating, reviewDetail, openCurriculumNum, registerStudentNum) VALUES (reviewNum_seq.nextVal, 5, '수료 후에도 신경써서 자소서 등 취업에 필요한 내용을 봐주셔서 너무 좋았습니다.', 12, 325); 2. 수강 후기 조회 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 WHERE stu.STUDENTNUM = 174; 3. 취업 상황 입력 INSERT INTO tblJobstatus (jobStatusNum, city, company, jobKindCategoryNum, registerStudentNum) VALUES (jobStatusNum_seq.nextVal, '경기', '카카오', 3, 335);			

요구사항번호	D-6(교육생)	작성일	2025-08-05
요구사항명	Q&A 게시판 질문	작성자	최석용
DML			
1. 교육생 질문 작성 INSERT INTO tblQuestion (questionNum, questionTitle, questionDetail, registerStudentNum) VALUES (questionNum_seq.nextVal, '면접 대비해서 CS 공부를 더 하고 싶어요.', '기술면접 CS 질문 대비해서 볼만한 책이나 자료가 있을까요?', 13); - 답변확인 SELECT stu.studentname AS 이름, q.questiontitle 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 WHERE stu.STUDENTNUM = 13; - 학생들이 질문한 전체 질문 목록 출력 SELECT stu.studentname AS 이름, q.QUESTIONTITLE AS 제목,			

```
q.questiondetail AS 질문내용
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;
```