

쌍용교육센터 관리 시스템

DB 프로젝트 최종 요약본

2조 박진혁, 최석용, 문예빈, 홍선민

프로젝트 개요

주제	쌍용교육센터 관리 시스템 데이터베이스 구축
개발 기간	2025.07.28 - 2025.08.08
개발 환경	Windows 11, Google Drive, Oracle Cloud Infrastructure - Autonomous Database(DB), DBeaver(개발 툴), Draw.io, ERDCloud
사용 언어	ANSI-SQL, PL/SQL

기능 요약

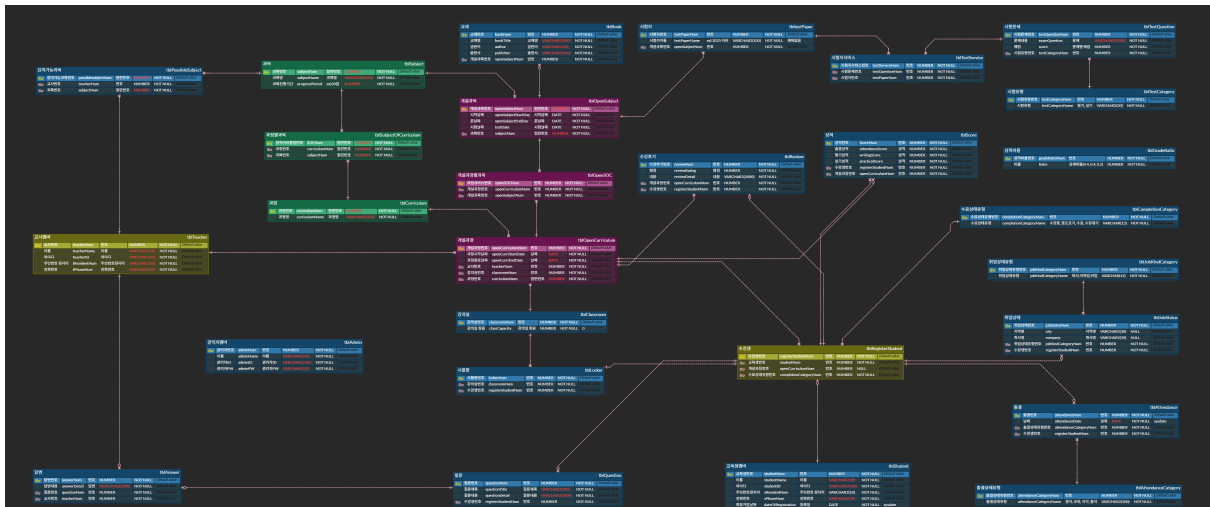
- 관리자 기능
 - 기본 기능: 기초 정보 관리, 교사 계정 관리, 개설 과정 관리, 개설 과목 관리, 교육생 관리, 시험관리 및 성적조회, 출결관리 및 출결조회
 - 추가 기능: 사물함 배정, 개근 포상, 수료 후 기능 > 취업률 조회
- 교사 기능
 - 기본 기능: 강의스케줄 조회, 배점 입출력, 성적 입출력, 출결관리 및 출결조회
 - 추가 기능: 수강 후기 조회, Q&A 게시판 답변
- 교육생 기능
 - 기본 기능: 성적 조회, 출결관리 및 출결조회
 - 추가 기능: 수강 완료(취업 상태 입력, 재취업 지원 신청, 수강 후기 작성), 사물함 조회, Q&A 게시판 질문, 수강 신청 > 인기 과정 조회

협업 방식

- 기능 설계, 데이터베이스 구조 설계, 요구사항 분석 등 프로젝트 전체 구조를 커뮤니케이션을 통해 의사 결정 및 공동으로 작업하여 전원 숙지
- 구글 드라이브(Google Drive)상 정의서 문서 공유를 통해 담당 파트 동시 작업, 작업 완료 후 디스코드 화면 공유를 통한 코드 공동 점검 및 리뷰 → 오류 발견 시 해결법 공동 탐색
- ERDCloud, draw.io 공유가 가능한 툴을 이용하여 담당 파트 동시 작업
- Oracle Cloud 서버를 활용하여 장소에 구애받지 않고 협업 가능한 환경 구축

데이터 구조(ERD)

- 총 28개 테이블



구현 예시

- ANSI-SQL

```
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 tbOpenCurriculum oc
    INNER JOIN tbTeacher tt ON oc.teacherNum = tt.teacherNum
    INNER JOIN tbCurriculum tc ON oc.curriculumNum = tc.curriculumNum
    INNER JOIN tbRegisterStudent rs ON oc.openCurriculumNum = rs.openCurriculumNum
    INNER JOIN tbStudent ts ON rs.studentNum = ts.studentNum
    INNER JOIN tbAttendance ad ON rs.registerStudentNum = ad.registerStudentNum
```

- 프로시저 - 과정별 취업률 조회

```
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
    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.');
```

- 뷰 - 인기 과정 조회

```

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;
```

프로젝트 후기

1. 데이터베이스 설계 과정

- 원문 형태의 요구사항을 데이터 구조로 정리하는 과정이 도전적이었으며, 모호한 부분을 명확히 하는 것이 중요하다는 점을 깨닫게 되었다.
- 요구사항 분석이 명확하지 않으면 전체 데이터 구조가 무너질 수 있다는 점을 인식할 수 있었고, 요구사항은 이해했지만 이를 테이블 구조로 구현하는 과정이 예상보다 어려웠다.
- 테이블 구조와 관련해 여러 고민(테이블 수, 카테고리 테이블 통합 등)을 하였고 이 과정을 거쳐 초기 설계와 최종 결과물 사이에 많은 변화가 생겼다. 이 과정에서 많은 공부가 되었다. 또한 초기 ERD 설계의 중요성을 실감하였다.
- 정적 정보와 동적 정보의 개념을 이해하고 구별해야 한다는 사실을 알게 되었다.
- 정규화로 인해 테이블과 조인이 많아진 것에 대한 의문이 있었는데, 어떤 방식이 더 좋은 방식인지는 고민이 필요할 것 같다.

2. SQL 작성 및 구현

- JOIN을 적극 활용하고 INNER와 OUTER JOIN을 상황에 맞게 사용할 수 있었다.
- ERD 설계를 바탕으로 쿼리를 작성하면서 JOIN에 대한 이해도가 높아졌다.
- 외래키 설정이 쿼리 작성 시 고려할 점을 많게 만든다는 점을 경험하였다.
- 프로시저 문법에 대한 이해도가 아직 부족하여 ANSI-SQL을 먼저 작성한 후 PL/SQL로 변환했다.
- PL/SQL에 어려움을 느껴 사용을 무의식적으로 회피했던 점이 아쉬웠으며, 더 많은 공부가 필요하다는 것을 느꼈다.

3. 협업 및 프로젝트 관리

- 오라클 클라우드 데이터베이스를 처음 사용하면서 트랜잭션(커밋, 롤백)의 중요성을 경험할 수 있었다.
- 다양한 툴을 활용한 협업으로 작업의 유연성과 확장성이 극대화되었다.
- 팀원들의 의견을 경청하고 수용하여 의견을 빠르게 통합할 수 있었다.
- 업무 분담과 상호 협력으로 기한 내에 여유롭게 작업 완료할 수 있었다.
- 팀원들 간의 의사소통과 존중이 좋은 결과를 만든다는 점을 재확인했다.

4. 주요 학습 포인트

- 데이터베이스 프로젝트에서 단순히 테이블 생성과 데이터 삽입이 중요하다기보다는 체계적인 설계와 구현이 중요하다는 점을 느꼈다.
- 데이터 최적화가 데이터베이스 기술의 핵심임을 느꼈으며, 데이터베이스 성능 관련 최적화 부분이 부족했다는 아쉬움을 느꼈다.

- 수업에서 이해가 부족했던 부분들을 프로젝트를 통해 배울 수 있었다.
- 전반적으로 ERD 설계의 중요성, 요구사항 분석의 정확성, 협업의 가치에 대해 공통적으로 배우고 느낄 수 있었다. 또한 SQL 구현 과정에서 팀원들 각자의 도전과 성장이 있었으며, 이론적 지식을 실제 프로젝트에 적용하는 과정에서 더 깊은 이해를 얻게 되었다.