

★ SIST_Semi_ORACLE ★

프로젝트 3팀 [회의록]

- 프로젝트 기간: 2023년 11월 8일 오후 2:00 ~ 2023년 11월 17일 오후 6:00
- 조건: 『성적 처리 시스템』을 만들기위한 DB 설계(프로그램 개발전의 DB 단계까지)
 - ↳ 단계1) 현 상황에서 다룰 수 없는 요구사항 3가지 찾기
 - ↳ 단계2) 분석사항을 토대로 설정한 팀 정책 기반 ERD 그리기
 - ↳ 단계3) ERD 다이어그램 및 요구사항에 맞는 SQL문, PL/SQL문 작성

- 팀장: 채다선
- 팀원: 김경태, 김수환, 박가영, 박범구, 오수경

- 1. 각 파트를 나눠 테이블 작성


학생 관리							Untitled2
주민번호 및 자치리	Field	Domain	Type	NOT NULL	Default value	Comment	
학생 코드	Field2	Domain1	Type	NULL	Default value	Comment	
학생 이름	Field3	Domain	Type	NULL	Default value	Comment	
과명	Field4	Domain	Type	NULL	Default value	Comment	
수강과목	Field5	Domain	Type	NULL	Default value	Comment	
수강과목 총점	Field6	Domain	Type	NULL	Default value	Comment	
중도탈락여부	Field7	Domain	Type	NULL	NULL	Comment	

과목관리

Untitled

과목코드	Key	Domain	VARCHAR2	NOT NULL	Default value	Comment
과목명	Field	Domain	VARCHAR2	NULL	Default value	Comment
과목기간	Field2	Domain	VARCHAR2	NULL	Default value	Comment
교재명	Field3	Domain	VARCHAR2	NULL	Default value	Comment
과점명	Field4	Domain	VARCHAR2	NULL	Default value	Comment
교수님	Field5	Domain	VARCHAR2	NULL	Default value	Comment
강의차명	Field6	Domain	VARCHAR2	NULL	Default value	Comment

교수자 계정관리						Untitled3
교수자코드	Field	Domain	VARCHAR2	NOT NULL	Default value	교수자코드
교수자이름	Field2	Domain	VARCHAR2	NULL	Default value	Comment
배정학과명	Field3	Domain	VARCHAR2	NULL	Default value	Comment
과목기간	Field4	Domain	VARCHAR2	NULL	Default value	Comment
과제명	Field5	Domain	VARCHAR2	NULL	Default value	Comment
강의실	Field6	Domain	VARCHAR2	NULL	Default value	Comment
강의전행여부	Field7	Domain	VARCHAR2	NULL	Default value	Comment

과점 관리							Untitled5	
	과점코드	Key	Domain	NUMBER	NOT NULL	Default value	여러개의 과목으로 구성 가능	
	과점명	Field	Domain	VARCHAR2	NULL	Default value		
	과점기간	Field2	Domain	DATE	NULL	Default value		
	강의일	Field3	Domain	VARCHAR2	NULL	Default value		시작 연월일, 끝 연월일
	과목명	Field7	Domain	Type	NULL	Default value		Comment
	과점기간	Field4	Domain	Type	NULL	Default value		Comment
	교재명	Field5	Domain	Type	NULL	Default value		Comment
	교수자명	Field6	Domain	Type	NULL	Default value		Comment

성적 관리				Untitled6		
주민번호 뒷자리	Field	Domain	Type	NULL	Default value	Comment
성적 처리	Field2	Domain	Type	NULL	Default value	Comment
교육별 배경	Field3	Domain	Type	NULL	Default value	Comment
종도합격여부	Field4	Domain	Type	NULL	Default value	Comment

IDPW						Untitled
아이디	Field	Domain	Type	NULL	Default value	Comment
패스워드	Field2	Domain	Type	NULL	Default value	Comment

- 2. 테이블을 전체적인 관점으로 삭제할 부분과, 다른 테이블에서 가져올 부분을 연결



결정사항

1. 테이블의 컬럼을 작성완료
2. 앞으로 진행할 내용) 다음에는 테이블간 연결및 관계에 대해 다시 생각해보기
3. 마주한 문제) 필요한 컬럼을 가져와 테이블에 연결했는데, M:M의 관계를 가지게 됨.

M:M의 관계를 1:M의 관계로 바꾸기 위해 테이블을 더 세분화하고 중간테이블에 대해 고민중.

★3조 회의록(2)★ - 11월 9일 목요일

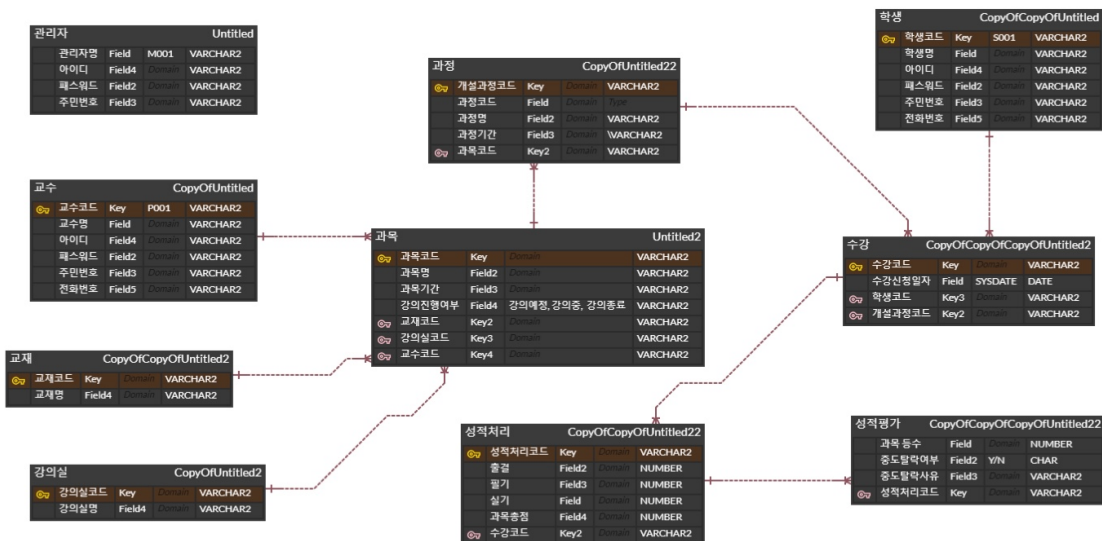
- 참석자: 채다선, 김경태, 김수환, 박가영, 박범구, 오수경
- 회의시간: 2023년 11월 9일 오전 9:00 ~ 2023년 11월 9일 오후 6:00

회의주제

- 요구분석서에 따라 테이블 컬럼 항목 재정리 & 컬럼별 타입 종류 설정
- 피드백 받은 내용으로 테이블에 반영

회의내용

- 1교시~ 3교시 진행내용
 - 요구분석서 기능 구현 세부 내용에 따라 각 테이블의 컬럼 항목 재정리 후 컬럼별 타입 종류 설정
 - 항목화 가능한 컬럼별로 테이블 분리 (과정 → 과정 개설 / 과정 상세
과목 → 과목 개설 / 과목 상세)
- 1차 피드백
 - 1차 ERD 다이어그램



- 피드백 상세 내용 :
- 확인 할 사항)

처음시작은 관리자/교수자/학생을 어떻게 구분할 것인지 논의가 진행되었어야 함.

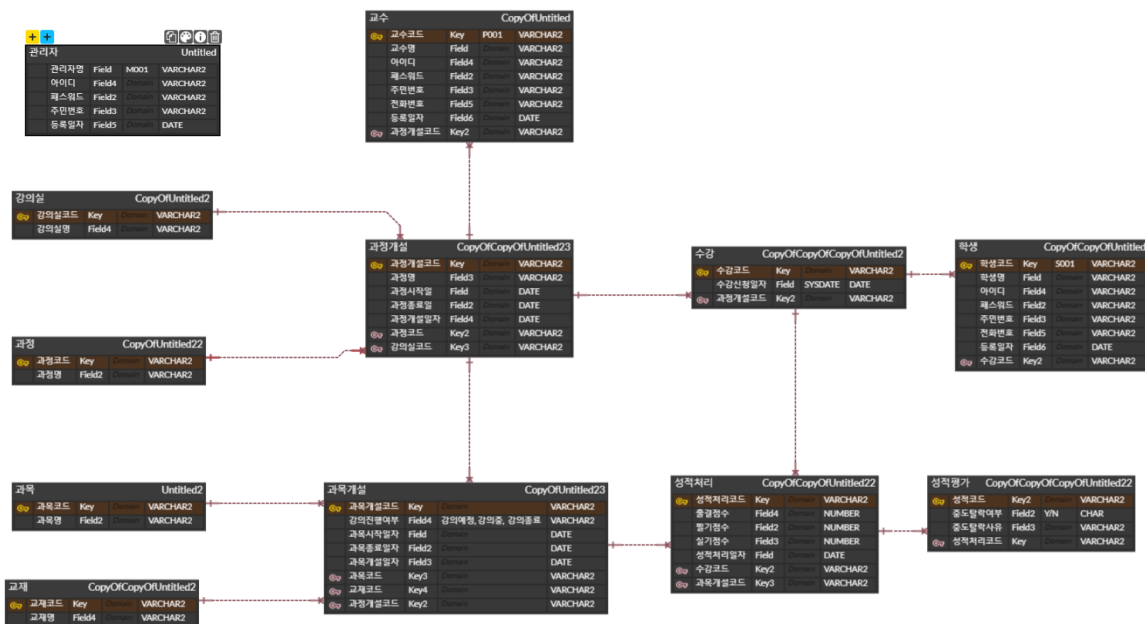
- 현재우리의 테이블 구조로는 테이블이 관리자/교수자/학생으로 구분되어 있기 때문에 → 화면에서 관리자/교수자/학생을 버튼등의 선택 이후로 파악됨

- 입력정보에서 주민번호 뒷자리를 기본으로 등록한다고 했지만, 본인이 맞는지 확인하거나 정보를 수정하기 위해서는 패스워드 컬럼이 독립적으로 있어야 원활한 진행이 가능함.
- 만약 없다면.. 주민번호 뒷자리를 입력하고 수정하는등 문제가 발생할 여지가 있음.

- 과정과 과목 $m:1 \rightarrow 1:m$ 의 관계로 수정되어야 함

★중요★ 피드백이나 논의된 부분이 하나의 컬럼이더라도 ERD 는 관계를 중요하게 보기 때문에 연관된 내용까지 확인 및 수정이 필요하기 때문에 모든 테이블을 다시 확인해야함

- 2차 ERD 다이어그램



- 수정 할 사항)

- 논의 할 사항)

- 1. 교수자는 과정 개설 후 등록가능.

과정이 개설되기 전에 교수는 어느 과정에도 등록 되어있을 수 없다.

- 2. 과목마다 교수자가 다르게 설정되는지 (한 과정에 다수의 교수자)

한 과정에 한 명의 교수자가 설정되는지

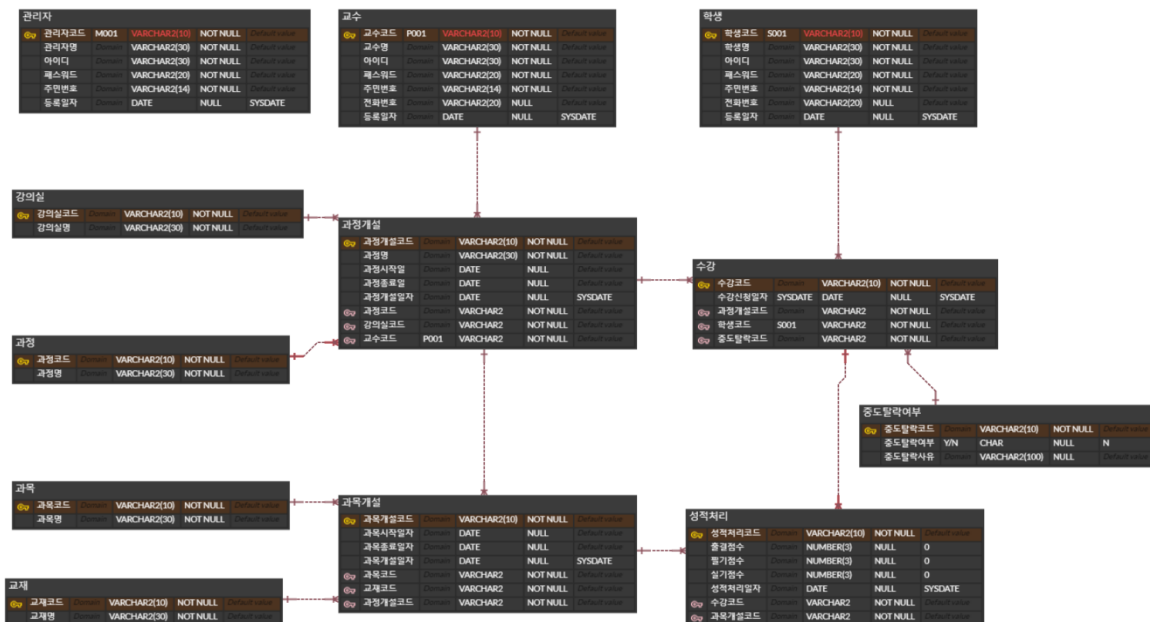
(한 과정에 한 명의 교수자, 즉 한 명의 교수자가 다수 과목 강의)

- 3. 과목개설 테이블의 강의진행여부 (강의예정 / 강의중 / 강의종료) 컬럼
하나의 속성은 하나의 속성값만을 가져야 하는 제 1 정규화에 위배되며
교육과정이 업데이트 될 때마다 강의 진행여부도 함께 업데이트 되어야 하는 문제 발생.
쿼리로 얻어낼 수 있는 결과(현재날짜로 얻어낼 수 있음)는 컬럼으로 만들어선 안 된다.
→ 무결성

- 잘한 사항)
과정개설 테이블과 과목개설 테이블이 참조할 수 있는 테이블 분리
강의실테이블 + 과정테이블 → 과정개설 테이블
교재테이블 + 과목테이블 → 과목개설 테이블

● 3차 피드백(3팀 두번째)

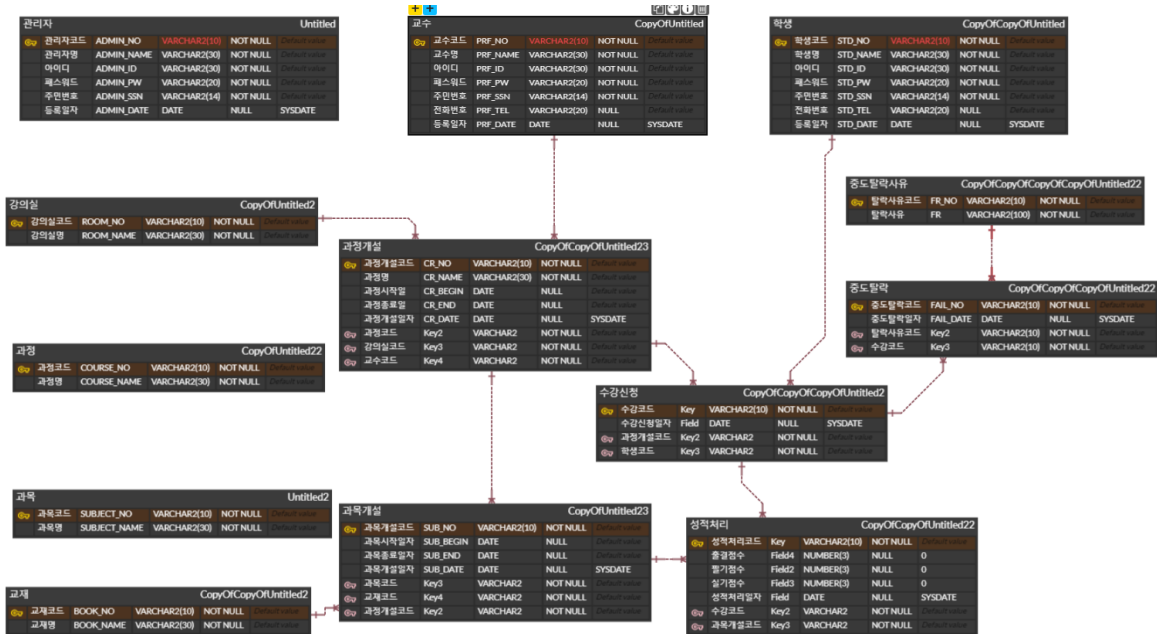
○ 3차 ERD 다이어그램



- 피드백 상세 내용 :
 - 잘한 사항) 12개의 테이블 각각 확인(관리자, 교수, 학생, 과정, 과목, 과정개설, 과목개설, 강의실...)
 - 논의 할 사항) 중도탈락여부의 위치 재확인 필요
 - 중도탈락여부가 수강에 포함되어있다면, 수많은 학생의 데이터에 중도탈락하지 않은 데이터까지 중도탈락에 데이터를 업데이트해야하는 메모리 낭비가 일어남.

● 4차 피드백(3팀 두번째)

○ 4차 ERD 다이어그램



○ 피드백 상세 내용 :

- 수정할 사항)
- 과정개설테이블의 과정명 삭제되어야함(과정테이블에 과정명이 이미 존재)
- 출결, 실기, 필기의 배점을 교수자가 직접 넣어야하므로, 해당 칼럼을 테이블에 포함시키는 방법에 대해 고민할 것.

결정사항

1. 요구분석서 기능 구현을 위한 테이블과 각 테이블 내 컬럼 설정 완료.
2. 테이블 간 연결관계와 컬럼의 데이터 타입, 변수명 설정 완료.

★3조 회의록(3)★ - 11월 10일 금요일

- 참석자: 채다선, 김경태, 김수환, 박가영, 박범구, 오수경
- 회의시간: 2023년 11월 10일 오전 9:00 ~ 2023년 11월 10일 오후 6:00

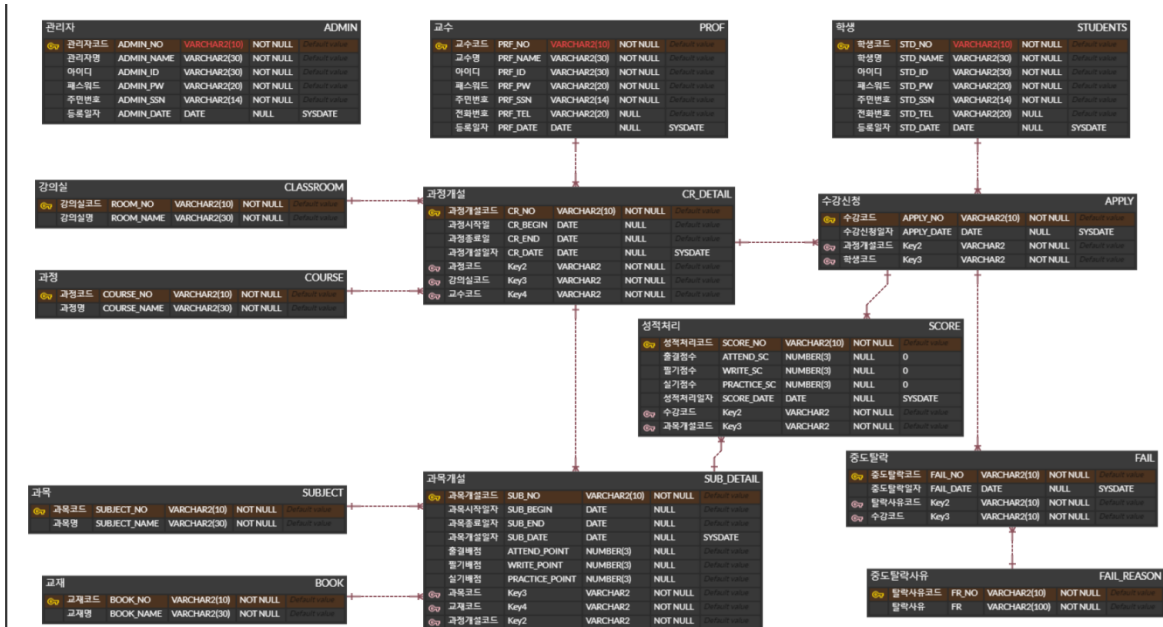
회의주제

- 확정된 ERD 다이어그램 기반 SQL DEVELOPER에 테이블, 기본키, 참조키 관계 생성
- 테스트 테이블 생성, 데이터 입력, 뷰 생성

회의내용

5차 피드백(3팀)

1차 ERD 다이어그램



피드백 상세 내용 : ERD 다이어그램 작성 완료, 다음 단계 진행

- 계정: TEAM3 / java006\$

- [3단계]DB 설계 시작!

- 논의 할 사항)

- 오라클 접속 팀 계정 생성

20231110_01_sys.sql

local_sys 접속



CREATE USER TEAM3
IDENTIFIED BY java006\$;
====>> User TEAM3이(가) 생성되었습니다.



GRANT CONNECT, RESOURCE, UNLIMITED
TABLESPACE TO TEAM3;
====>> Grant을(를) 성공했습니다.



ALTER USER TEAM3 DEFAULT TABLESPACE USERS;
====>> User TEAM3이(가) 변경되었습니다.



SELECT *

- 테이블 생성 역할 분배

- 1. 관리자 - (범구)
- 2. 교수 - (수환)
- 3. 학생 - (수경)
- 4. 강의실 - (다선)
- 5. 교재 - (다선)
- 6. 과목 - (다선)
- 7. 과정 - (수경)
- 8. 과정개설 - (가영)
- 9. 과목개설 - (경태)
- 10. 수강신청 - (범구)
- 11. 성적처리 - (수환)
- 12. 중도탈락사유 - (가영)
- 13. 중도탈락 - (경태)

- 6차 피드백(3팀)

- 팀장 요청받은 사항) 논리 ERD를 물리 ERD로 바뀌서 가져오기(한국어->영어)
- 요구분석서를 기반으로 돌아가는 쿼리 구성

■ 요구분석서의 요구사항 쿼리 작성

EX) 항목조회 -> SELECT 문, 입력 -> INSERT 문

- 최종 SQL 파일에

- 테이블 생성
- 제약조건
- 프로시저
- 함수
- 샘플데이터 등 ...

- 오후 진행 사항

- 각자 맡은 테이블에 테스트 데이터 입력 → 데이터 입력 성공
- [관리자] 출력 화면 VIEW 생성
 - 교수자 정보 (수경) : **VIEW_PROF_INFO**
 - 과정 정보 (경태): **VIEW_CR_INFO**
 - 과목 정보 (다선): **VIEW_SUB_INFO**
 - 학생 정보 (가영): **VIEW_STUDENT_INFO**
 - 성적 정보 출력_교수 (수환) : **VIEW_SCORE_INFO**
 - 성적 정보 출력_학생 (범구): **VIEW_STUDENTSC_INFO**

결정사항

1. ERD 다이어그램 작성 완료
2. [3단계] DB설계 시작에 따른 쿼리문 작성 필요(TABLE, VIEW 생성, 샘플데이터 작성 완료됨)

★3조 회의록(4)★ - 11월 13일 월요일

- 참석자: 채다선, 김경태, 김수환, 박가영, 박범구, 오수경
- 회의시간: 2023년 11월 13일 오전 9:00 ~ 2023년 11월 13일 오후 6:00

회의주제

- 기능 구현을 위한 프로시저, 트리거 생성
- 생성한 뷰, 프로시저, 트리거 테스트

회의내용

- 오전 진행 사항
 1. 관리자, 학생, 교수 로그인
 - 관리자 로그인 프로시저
`CREATE OR REPLACE PROCEDURE PRC_ADMIN_LOGIN`
 - 교수 로그인 프로시저
`CREATE OR REPLACE PROCEDURE PRC_PROF_LOGIN`
 - 학생 로그인 프로시저
`CREATE OR REPLACE PROCEDURE PRC_STUDENTS_LOGIN`
 2. 관리자는 입력된 교수자 정보에 대해 수정, 삭제할 수 있어야 함
 - 교수 코드 삭제 시 교수 코드를 참조하는 테이블도 삭제되는 트리거 생성
`CREATE OR REPLACE TRIGGER TRG_PRF_DELETE_01`
`CREATE OR REPLACE TRIGGER TRG_CRD_DELETE_01`
 3. 교수가 성적에 대한 수정 및 삭제를 할 수 있어야 함
 - 학생 코드 삭제 시 학생 코드를 참조하는 테이블도 삭제되는 트리거 생성
`CREATE OR REPLACE TRIGGER TRG_SBD_DELETE_01`
`CREATE OR REPLACE TRIGGER TRG_STU_DELETE_01`
`CREATE OR REPLACE TRIGGER TRG_APL_DELETE_01`
`CREATE OR REPLACE TRIGGER TRG_CLR_DELETE_01`
`CREATE OR REPLACE TRIGGER TRG_COS_DELETE_01`
`CREATE OR REPLACE TRIGGER TRG_FAR_DELETE_01`
`CREATE OR REPLACE TRIGGER TRG_BOK_DELETE_01`
 4. 관리자측 요구분석
 - 관리자 계정 관리 기능 구현
 - i) 관리자 로그인 : 관리자용 패스워드(주민번호 → 패스워드) 프로시저 생성
`CREATE OR REPLACE PROCEDURE PRC_ADMIN_PW`

- 교수자 계정 관리 기능 구현

i) 관리자는 여러 명의 교수자를 사전에 등록 가능

```
CREATE OR REPLACE PROCEDURE PRC_PROF_PW
```

ii) 교수자는 자신이 강의한 과목의 성적 처리 업무 수행 가능

```
CREATE OR REPLACE PROCEDURE PRC_SD_INSERT
```

```
CREATE OR REPLACE PROCEDURE PRC_SCORE_INFO
```

iii) 관리자는 모든 교수자의 정보를 출력 가능

```
CREATE OR REPLACE VIEW VIEW_PROF_INFO
```

5. 과목개설, 성적 처리 프로시저 생성

```
CREATE OR REPLACE PROCEDURE PRC_SD_INSERT
```

```
CREATE OR REPLACE PROCEDURE PRC_SCORE_INFO
```

6. 과정개설, 과목개설 프로시저 생성

```
CREATE OR REPLACE PROCEDURE PRC_CR_DETAIL_INSERT
```

```
CREATE OR REPLACE PROCEDURE PRC_SUB_DETAIL_INSERT
```

● 오후 진행 사항

- 1. 요구사항 내용 작성여부 재확인
- 2. 개설과정 프로시저: 조회조건 없을때 값 입력
 - 문제 상황) 개설 과정 프로시저에 데이터 입력 시 비교 대상 테이블의 데이터가 NULL 일 때 비교가 불가능
 - 해결 방안) 조회조건이 없는 각 경우에 따라 그룹함수(COUNT, MIN, MAX)를 이용해 조회조건이 없을 때를 분기처리 하여 코드 실행

```
CREATE OR REPLACE PROCEDURE PRC_CR_DETAIL_INSERT
( V_CR_NO IN CR_DETAIL.CR_NO%TYPE
... 매개변수 중략 ...)
IS
  V_PRFBEGIN CR_DETAIL.CR_BEGIN%TYPE;
  V_PRFEND CR_DETAIL.CR_END%TYPE;
  V_ROOMBEGIN CR_DETAIL.CR_BEGIN%TYPE ;
  V_ROOMEND CR_DETAIL.CR_END%TYPE;

  USER_DEFINE_ERROR1 EXCEPTION;
  USER_DEFINE_ERROR2 EXCEPTION;

  PRF_JUNG_CHECK NUMBER := 0;

BEGIN

  SELECT COUNT(*) INTO PRF_JUNG_CHECK
  FROM CR_DETAIL
  WHERE PRF_NO = V_PRF_NO;

  SELECT MIN(CR_BEGIN), MAX(CR_END) INTO V_PRFBEGIN, V_PRFEND
  FROM CR_DETAIL
  WHERE PRF_NO = V_PRF_NO;
```

```

SELECT MIN(CR_BEGIN), MAX(CR_END) INTO V_ROOMBEGIN, V_ROOMEND
FROM CR_DETAIL
WHERE ROOM_NO = V_ROOM_NO;

IF (PRF_JUNG_CHECK >=1)
THEN
    IF ((V_CR_BEGIN BETWEEN V_PRFBEGIN AND V_PRFEND) OR (V_CR_END BETWEEN V_PRFBEGIN AND
V_PRFEND))
    THEN RAISE USER_DEFINE_ERROR1;
    ELSIF ((V_CR_BEGIN BETWEEN V_ROOMBEGIN AND V_ROOMEND) OR (V_CR_END BETWEEN V_ROOMBEGIN
AND V_ROOMEND))
    THEN RAISE USER_DEFINE_ERROR2;
    ELSE
    INSERT INTO CR_DETAIL(CR_NO, CR_BEGIN, CR_END, COURSE_NO, ROOM_NO, PRF_NO)
VALUES(V_CR_NO, V_CR_BEGIN, V_CR_END, V_COURSE_NO, V_ROOM_NO, V_PRF_NO);
    END IF;
    ELSE
    INSERT INTO CR_DETAIL(CR_NO, CR_BEGIN, CR_END, COURSE_NO, ROOM_NO, PRF_NO)
VALUES(V_CR_NO, V_CR_BEGIN, V_CR_END, V_COURSE_NO, V_ROOM_NO, V_PRF_NO);
    END IF;

EXCEPTION
    WHEN USER_DEFINE_ERROR1 THEN
        RAISE_APPLICATION_ERROR(-20003, '과정기간을 다시 확인해주세요!');
        ROLLBACK;
    WHEN USER_DEFINE_ERROR2 THEN
        RAISE_APPLICATION_ERROR(-20004, '해당 교수 및 강의실에 대한 정보가 없습니다. ');
        ROLLBACK;
END;

```

결정사항

- 개설과정 프로시저 조회조건 없을 때 분기처리하여 실행완료!! (내일 코드 재확인필요)
- 요구사항에 따른 코드 작성은 완료되었으나, 샘플데이터, 프로시저 및 뷰 조회 내용 검토

★3조 회의록(5)★ - 11월 14일 화요일

- 참석자: 채다선, 김경태, 김수환, 박가영, 박범구, 오수경
- 회의시간: 2023년 11월 14일 오전 9:00 ~ 2023년 11월 14일 오후 6:00

회의주제

- 기능 구현을 위해 작성할 쿼리 종류 논의 (프로시저 / 트리거 / ...)
- 뷰 조회 테스트

회의내용

- **오전 진행 사항 : 작성한 프로시저 테스트 및 수정**
 - 1. 실행해야 하는 모든 쿼리문을 PL/SQL(프로시저, 트리거 등)로 만들어야 하는지
-> 결론: SQL 과 PL/SQL 구문의 사용 용례를 고려하여 작성해야 하므로
모든 구문을 PL/SQL로 작성하지는 않는다.
 - 회의 및 조사한 내용
SQL (Structured Query Language)과 PL/SQL (Procedural Language/SQL)은 데이터베이스와의 상호작용을 위해 사용되는 언어지만 각각의 목적과 사용 사례가 다르기 때문에 선택적으로 사용해야 한다.

	SQL (Structured Query Language)	PL/SQL (Procedural Language/SQL)
목적	데이터를 조회, 삽입, 갱신 및 삭제하는 등의 데이터베이스 관리 작업을 수행하는 데 사용	SQL에 프로시저, 함수, 트리거 등과 같은 절차적 프로그래밍 기능을 추가하여 더 복잡하고 유연한 데이터베이스 로직을 작성
사용 사례	테이블에서 데이터를 검색하거나 변경하는 간단한 쿼리 작업에 적합	데이터베이스에서 특정 작업을 수행하는 자체적인 프로시저 및 함수를 작성하거나, 트리거를 사용하여 특정 이벤트에 대한 반응을 정의할 때 사용
특징	선언적 언어로, 개별적인 데이터베이스 연산을 수행합니다. 주로 데이터의 검색, 추가, 갱신, 삭제와 같은 기본적인 작업에 사용	절차적 언어로, 변수, 조건문, 루프 등과 같은 프로그래밍의 특성을 SQL에 추가한 형태 (주로 데이터베이스 로직이나 비즈니스 룰을 구현할 때 사용)
선택 기준	SQL : 데이터 검색, 추가, 갱신 등의 기본적인 데이터 조작 작업 PL/SQL : 데이터 베이스 로직이나 특정 작업수행을 위한 프로시저, 함수 사용 등 복잡한 로직 작성 시 유용	

○ 2. 개설 과정 프로시저

- 문제 상황) 집계함수로 조회되는 구문이 여러 과목이 있는 경우 과목사이에 빈 공간을 인식못함
MAX / MIN만 인식하므로 중간 기간을 인식하지 못 해 과정개설이 가능해짐.
- 해결 방안) pl/sql에서 조건 비교 -> sql 상에서 조건 비교
기존에는 교수의 해당 과목의 비교를 sql로 집계함수를 통해 기간 최소값, 최대값 조회 후
pl/sql에서 비교하였으나, sql상에서 where 에서 기간을 확인후 가져올 수 있도록 수정

```

SELECT COUNT(*) INTO PRF_JUNG_CHECK
FROM CR_DETAIL
WHERE PRF_NO = V_PRF_NO
  AND (V_CR_BEGIN BETWEEN CR_BEGIN AND CR_END OR V_CR_END BETWEEN CR_BEGIN AND CR_END);

IF PRF_JUNG_CHECK > 0 THEN
  RAISE USER_DEFINE_ERROR1;
END IF;

SELECT COUNT(*) INTO ROOM_JUNG_CHECK
FROM CR_DETAIL
WHERE ROOM_NO = V_ROOM_NO
  AND (V_CR_BEGIN BETWEEN CR_BEGIN AND CR_END OR V_CR_END BETWEEN CR_BEGIN AND CR_END);

IF ROOM_JUNG_CHECK > 0 THEN
  RAISE USER_DEFINE_ERROR2;
END IF;

```

- 3. 성적 처리 프로시저 (배점) -> INSERT, UPDATE

- **문제 상황** 과목배점을 각 1점으로 최대값으로 처리할 경우, 성적 처리시 각 과목마다 최대 배점 확인 후 작성해야함
- **해결 방안** 성적 입력 시 배점 비율 (출석 20 /필기 40/ 실기40) 에 따라 자동으로 처리되도록

- 4. 중도탈락 / 중도탈락 사유 테이블 INSERT 테스트

- 오후 진행 사항

○ 1. 과목 개설 프로시저 중복 입력 처리 테스트

문제 상황) 중복되는 데이터 경우의 수 확인하여 입력가능한 데이터 조합 예상 후 입력 해보기

해결 방안) 조건문을 분기처리 될 수 있도록 수정

- 2. 생성한 뷰 데이터 테스트

- i) VIEW_STUDENT_INFO - 미해결
- ii) VIEW_SCORE_INFO - 성적 처리 프로시저 수정 필요

- 3. FK로 연결된 데이터 삭제 테스트(프로시저) 확인완료

결정사항

1. 개설과정, 개설과목 UPDATE 쿼리 작성하기
2. <관리자> 학생정보 출력 VIEW 수정
3. 수강신청 프로시저와 성적 처리 (과정명)
4. 성적 정보 삭제 확인
5. 작성한 VIEW 다 : 다 관계인 경우 쪼개기!

★3조 회의록(6)★ - 11월 15일 수요일

- 참석자: 채다선, 김경태, 김수환, 박가영, 박범구, 오수경
- 회의시간: 2023년 11월 15일 오전 9:00 ~ 2023년 11월 15일 오후 6:00

회의주제

- 생성한 트리거, 프로시저, 뷰 오류 수정
- 개설 과정, 개설 과목 업데이트 프로시저 생성

회의내용

- 오전 시간
 - 교수 데이터 삭제 시 다른 테이블의 교수 데이터도 함께 삭제되는 트리거 수정
`CREATE OR REPLACE TRIGGER TRG_PRF_DELETE_01`
→ 과정이 이미 개설된 교수인 경우 교수 정보가 삭제 되지 않도록
 - 개설 과정, 개설 과목 업데이트 프로시저 생성
`CREATE OR REPLACE PROCEDURE PRC_CR_DETAIL_UPDATE`
`CREATE OR REPLACE PROCEDURE PRC_SUB_DETAIL_UPDATE`
→ 주의사항 : 연결된 과정기간, 과목기간과 상충되지 않도록
 - 뷰 수정 (전반적으로 수정)
→ 주의사항 : inner join인 경우 교차되는 값만 나오기 때문에,
필요사항에 따라 left 또는 right 조인 진행
- 오후 시간
 - 특성에 맞게 페이지 분리(테이블생성, 프로시저, 트리거, 데이터, 시퀀스 등..)
 - 성적 업데이트 프로시저 수정
`CREATE OR REPLACE PROCEDURE PRC_SCORE_UPDATE`
→ 주의사항 : 성적 수정 시 과목 시작일과 비교하여 시작일 이후에만 성적 수정이 가능하도록 변경
 - 과목 종료일 이후 성적 확인가능하도록 뷰 수정
`CREATE OR REPLACE VIEW VIEW_STUDENTSC_INFO`
 - 데이터 시퀀스로 변경

결정사항

- 요구사항에 따라 코드 구성 1차 완료
- 진행예정) 데이터 입력 후 테스트 및 오류사항 수정

★3조 회의록(7)★ - 11월 16일 목요일

- 참석자: 채다선, 김경태, 김수환, 박가영, 박범구, 오수경
- 회의시간: 2023년 11월 16일 오전 9:00 ~ 2023년 11월 16일 오후 6:00

회의주제

- 데이터 입력 확인
- 추가로 필요한 프로시저 , 뷰
- 생성

회의내용

- 오전 시간
 - 성적처리 프로시저 (`PRC_SCORE_INSERT`) 수정
→ 중도탈락 시 성적이 입력되지 않게 하기 위해, 성적입력은 과목종료 후 입력가능 날짜 연산이 되는 WHERE 조건문 수정
 - 생성한 PL/SQL 문에 주석 달기, 불필요한 변수 삭제
- 오후 시간
 - 관리자의 이름, 비밀번호 업데이트 프로시저 (`PRC_ADMIN_UPDATE`)
 - 교수와 학생의 이름, 비밀번호, 전화번호 업데이트 프로시저
`PRC_PROF_PW`, `PRC_STUDENTS_UPDATE`
 - 모든 업데이트 구문에서 업데이트 하는 구문의 해당 번호가 있는지 존재여부 확인 프로시저
 - 샘플 데이터 내용을 최종 제출 파일(`team3_cds.sql`)로 통합 및 테스트
 - 현재 날짜 기준으로 성적처리 가능한 학생 정보를 출력하는 뷰 생성
`CREATE OR REPLACE VIEW POSSIBLE_VIEW`
 - 수강신청한 학생 정보 조회
`CREATE OR REPLACE VIEW APPLIED_VIEW`

결정사항

- 샘플 데이터 조건에 맞춰 테스트 및 수정 완료
- 최종 제출 SQL 파일 통합 완료