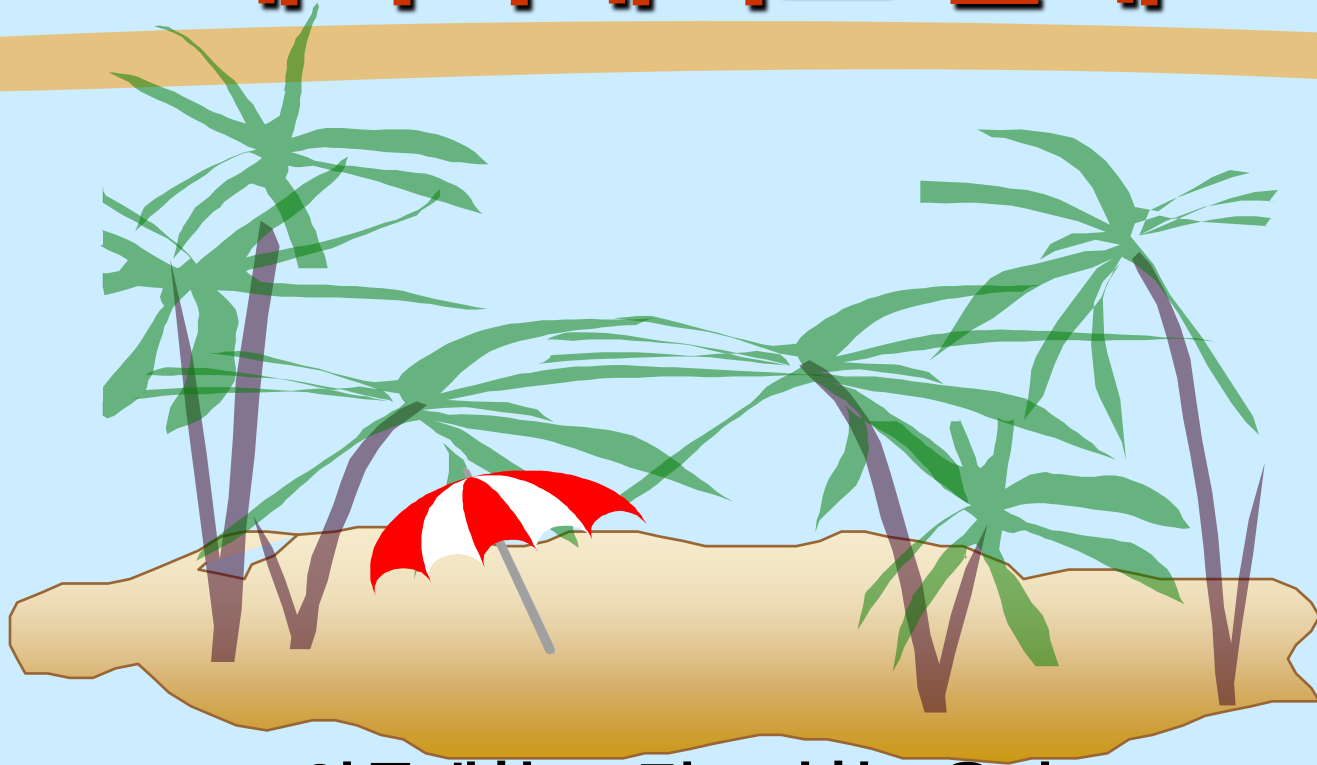


# 데이터베이스 설계



안동대학교 정보과학교육과

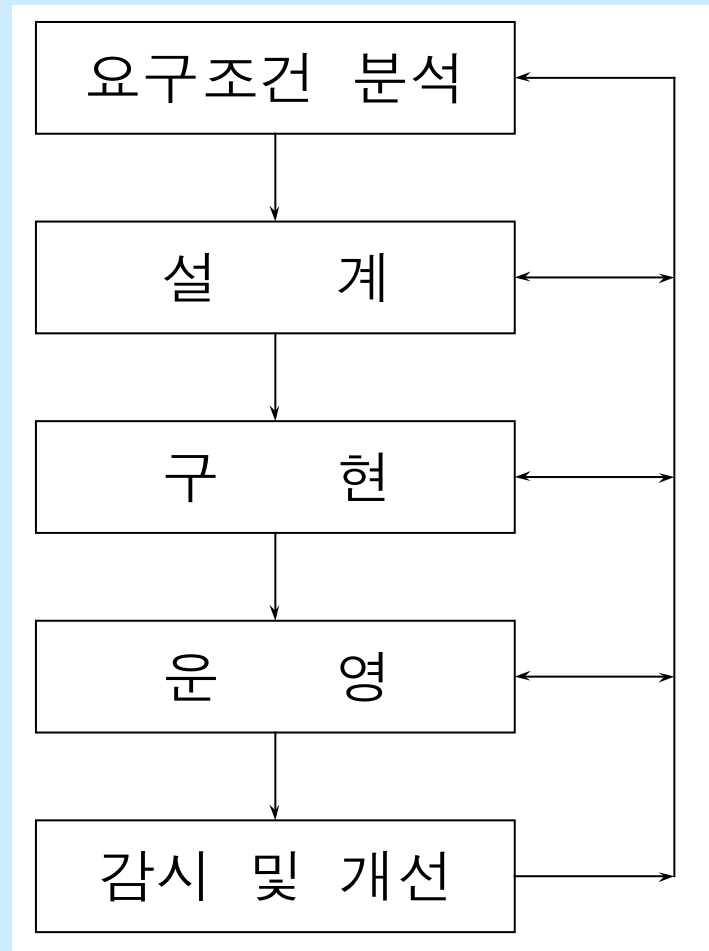
# 배울 내용



- ❑ 데이터베이스 설계 환경
- ❑ 요구 조건 분석
- ❑ 개념적 설계
- ❑ 논리적 설계
- ❑ 물리적 설계
- ❑ 데이터베이스 구현

# 데이터베이스 설계 환경

## ❑ 데이터베이스 생명 주기 (Database life Cycle)



# 데이터베이스 설계 단계

## 단계별 주요 작업 내용



# 데이터베이스 설계 전략

## ❑ 데이터 중심(data-driven) DB 설계

### ❑ DB 내용과 구조 설계

## ❑ 처리 중심(processing-driven) DB 설계

### ❑ 데이터의 처리와 응용 소프트웨어(트랜잭션) 설계

⇒ 병행적으로 추진

# 데이터베이스 설계 고려사항

- ❑ 무결성(Integrity) - 제약 조건
- ❑ 일관성(Consistency) - 응답, 출력
- ❑ 회복(Recovery) - 장애 복구
- ❑ 보안(Security)- 불법 접근
- ❑ 효율성(Efficiency) - 응답시간, 저장 공간, 처리도
- ❑ 데이터베이스 확장(Growth) - 응용과 데이터의 확대

# 요구 조건 분석 (Requirements analysis)

- 잠정적인 사용자 식별
- 사용자가 의도하는 DB의 용도 파악
- 공식적 요구조건 명세 정의

# 요구조건 내용

- 정적 정보 구조 요소:
  - 개체, 애트리뷰트, 관계성, 제약 조건
- 동적 DB 처리 요구조건 :
  - 트랜잭션 유형, 실행 빈도
- 범 기관적 제약조건 :
  - 경영 목표, 정책, 규정, ...



# 요구조건 분석 과정

- 정보의 내용과 처리 요구조건의 수집
  - 방법 : 서면 조사, 설문지, 인터뷰를 통해 문서화
  - 내용 : 업무, 데이터, 처리형태
- 범기관적 경영 목표와 제약조건을 식별
  - 장래 정보 전략
- 공식적 요구조건 명세(requirement specification)의 작성
  - 데이터, 트랜잭션, 작업-데이터 관계, 제약 조건
- 요구조건 명세의 검토 및 확인
  - 잠정적 확정

# 작업-데이터(task-data)와의 관계 예

작업 데이터	학생명부	성적표	과목표	교수명부	
학번	○	○			
학생이름	○	○			
주소	○			○	
학과	○	○	○	○	
과목이름		○	○		
과목번호		○	○		
성적		○			
학점		○	○		
교수이름			○	○	
전공				○	
직급				○	
전화번호	○			○	

# 요구조건 명세 작성 기법

## □ 다이어그램 방식

- HIPO, SADT, DFD

## □ 컴퓨터 이용 기법

- PSL/PSA(Problem Statement Language/Problem Statement Analyzer)

# 개념적 설계 (Conceptual design)

## □ 개념적(추상적) 스키마 모델링

- 데이터의 조직과 표현에 초점

- 데이터 중심 설계

## □ 트랜잭션 모델링

- 응용을 위한 데이터 처리에 초점

- 처리 중심 설계

# 개념적 스키마 모델링

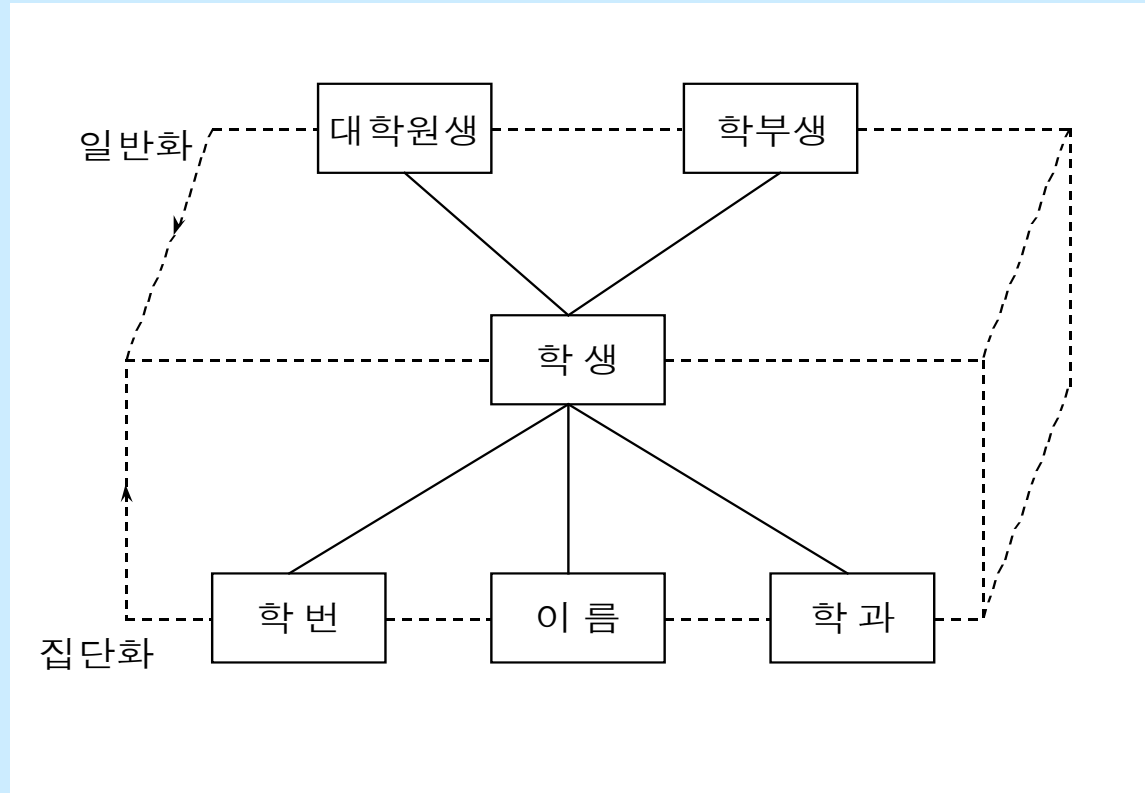
- ❑ 개념적 모델링 (Conceptual modeling)
  - ❑ 특정 DBMS에 독립적이고 고차원적인 표현
  - ❑ 개념적 구조 즉 개념적 스키마(conceptual schema)를 생성
  - ❑ 개념적 데이터 모델로 개체, 속성, 관계성을 선정 및 표현
  - ❑ E-R 다이어그램으로 표현
    - 개념적 데이터 모델
    - 고급 데이터 모델

# 개념적 데이터 모델

- ❑ Conceptual data model
- ❑ 고급 데이터 모델 (High-level data model)
- ❑ 특성
  - ❑ 표현력 : 개체 타입, 관계성, 제약 조건
  - ❑ 단순성 : 이해와 사용이 단순
  - ❑ 최소성 : 작은 수의 기본 개념만 사용
  - ❑ 다이어그램식 표현 : 시각적이고 종합적
  - ❑ 공식성 : 공식적 명세를 위해 모호하지 않고 정확
- ❑ DBMS에 독립적

# 개념적 스키마 모델링의 기본 원리

- 추상화(abstraction) : 개념(concept)화
  - 집단화(aggregation)
  - 일반화(generalization)



# 개념적 스키마 설계 방법

- 뷰 통합 방법(view integration approach)
- 애트리뷰트 합성 방법 (attribute synthesis approach)



# 뷰 통합 방법 (View integration)

- 하향식 방법(top-down approach)이라고도 함
  - 요구조건 명세로부터 먼저 몇 개의 부문별 뷰(View)를 식별하고 모델링
    - 개체, 키, 관계성, 애트리뷰트
- 부문별 뷰들을 통합해서 하나의 전체적 개념 스키마 작성
  - 동일성 통합(identity integration) : 같은 요소나 동의어들을 통합
  - 집단화(aggregation) : 개체 원소들을 그룹핑
  - 일반화(generalization) : 개체들의 공통 성질을 기초로 대 분류로 확장
  - 상호 모순 해결
    - 모순 : 이름, 타입, 도메인, 제약조건, 키 애트리뷰트, 집단화

# 애트리뷰트 합성방법 (Attribute Synthesis)

- ❑ 상향식 방법(bottom-up)이라고도 함
  - ❑ 애트리뷰트 리스트에서 출발
- ❑ 작업-데이터(task-data) 관계에 기초
  - ❑ 애트리뷰트들을 식별, 분류
    - ❑ 유일성 여부에 따라 구분
  - ❑ 개체 정의
    - ❑ 키 애트리뷰트, 설명 애트리뷰트
  - ❑ 관계성 정의
    - ❑ 개체간, 개체-애트리뷰트, 애트리뷰트 간 관계
    - ❑ 그래프 표기법으로 표현(ERD)
  - ❑ 개념적 구조(ERD)를 분석, 확인
    - ❑ Cardinality, 종속 정보, 누락 정보

# 트랜잭션 모델링 (Transaction modeling)

- 응용을 위한 트랜잭션을 명세
  - 주요 트랜잭션을 식별하고 기능적 특성을 DB 설계 초기에 명세
  - 스키마에 트랜잭션이 필요로 하는 정보가 모두 포함되고 있는지 확인
  - 트랜잭션의 상대적 중요성, 예상 실행 빈도수 파악
    - 물리적 DB 설계의 자료
- 입출력과 기능적 행태 명세
  - 입력 데이터, 출력 데이터, 내부 제어 흐름
- 트랜잭션 유형
  - 검색, 갱신, 혼합(검색, 갱신)

# 논리적 설계 (Logical design)

- 개념적 스키마로부터 목표 DBMS가 처리할 수 있는 논리적 스키마(logical schema) 생성
  - 요구 조건 명세 만족
  - 무결성, 일관성 제약 조건 만족

# DBMS의 선정 - 비용 중심

- ❑ 기능 : 기본적인 기능과 추가적인 기능
- ❑ 소프트웨어 구입 비용 : options
- ❑ 유지 비용 : version up
- ❑ 하드웨어 구입 비용
  - ❑ 부수적인 메모리, 주변 장치
- ❑ DB 생성과 변환 비용
  - ❑ 구형에서 신형 시스템
  - ❑ from scratch
- ❑ 인건비 : 새로운 직책과 조직
- ❑ 교육 훈련 비용
- ❑ 운영 비용

# 논리적 설계 단계

## □ 논리적 데이터 모델로 변환

- 개념적 스키마를 목표 DBMS에 맞는 논리적 데이터 모델로 변환

➡ 데이터 모델링

□ Relational, Hierarchical, Network, O-O, O-R

- 결과 : 목표 DBMS의 DDL로 기술된 스키마 (물리적 설계 매개변수는 제외)

## □ 트랜잭션 인터페이스 설계

- 전체적 트랜잭션 골격 및 인터페이스 정의

- 데이터 접근 방법 및 인터페이스를 절차적으로 명세

## □ 스키마의 평가 및 정제

- 정량적 정보와 성능 평가 기준에 따라 평가

□ 정량적 정보 : 데이터의 양, 처리빈도수, 처리 작업량

□ 성능 평가 기준 : 논리적 레코드 접근, 데이터 전송량, DB 크기

# 관계 데이터 모델로의 변환 예

□ 개체 타입 : 개체 릴레이션(entity relation)

□ 관계 타입

□ 연관된 개체 타입의 키 애트리뷰트들을 포함하는 관계 릴레이션 (relationship relation)

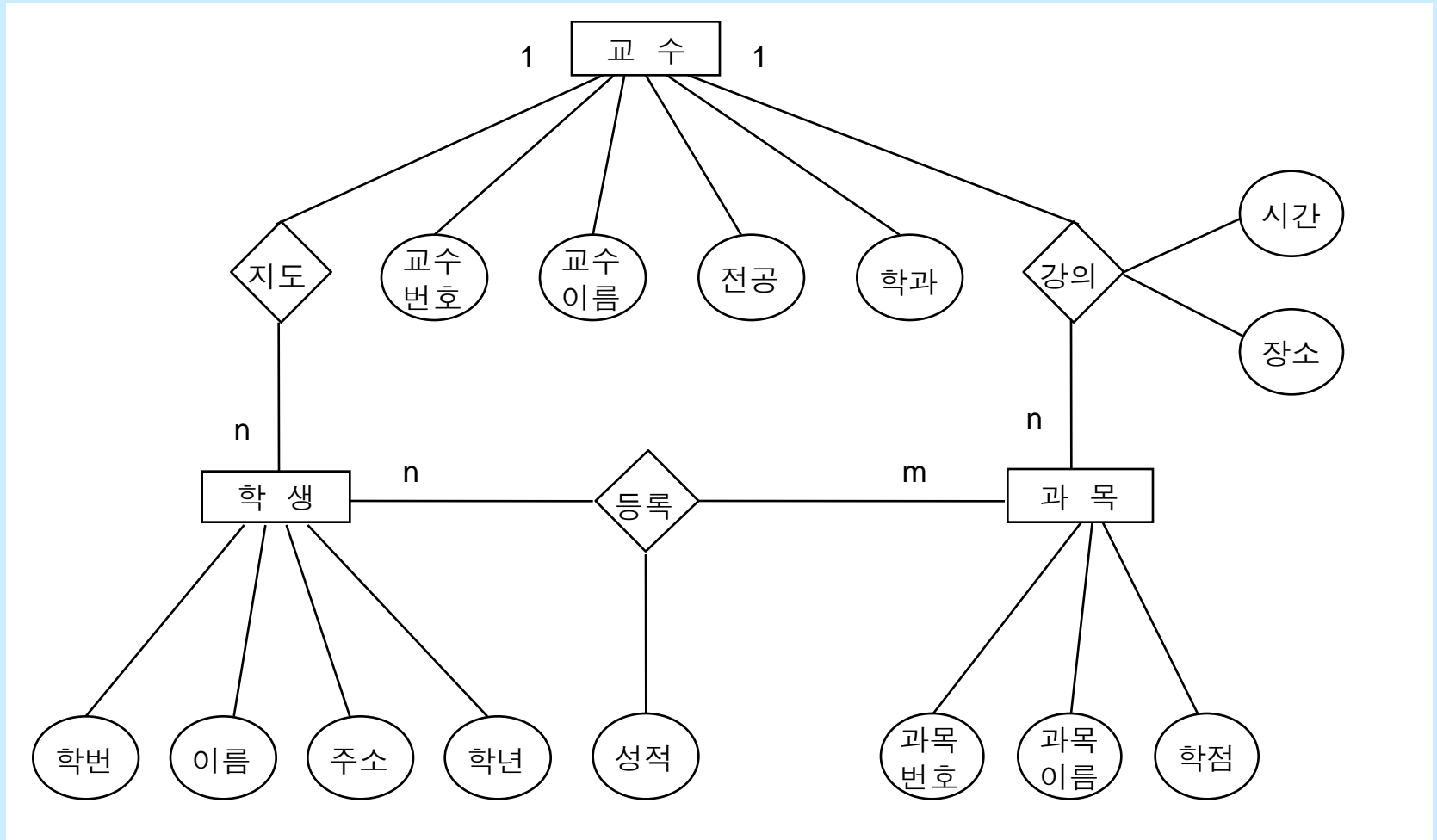
□ 두 릴레이션에 공통 애트리뷰트(키)를 포함시켜 관계성을 표현 : 묵시적 표현

➡ 설계 선택

□ 데이터의 중복

□ 효율적 처리

# 학사 관계 데이터베이스 E-R 다이어그램





# 학사 관계 데이터베이스 스키마 다이어그램

## □ 독립된 릴레이션으로 관계를 표현

교수

교수번호	교수이름	전공	학과
------	------	----	----

지도

교수번호	학번
------	----

학생

학번	이름	주소	학년
----	----	----	----

강의

교수번호	과목번호	시간	장소
------	------	----	----

과목

과목번호	과목이름	학점
------	------	----

등록

학번	과목번호	성적
----	------	----

개체 릴레이션

관계 릴레이션

# 교수와 학생간의 지도 관계의 표현

## □ 공통 애트리뷰트로 관계를 표현

교수

교수번호	교수이름	전공	학과
------	------	----	----

학생

학번	이름	주소	학년	교수번호
----	----	----	----	------

(a)

교수

교수번호	교수이름	전공	학과	학번
------	------	----	----	----

학생

학번	이름	주소	학년
----	----	----	----

(b)

# 물리적 설계

- 논리적 스키마로부터 효율적인 내부적 스키마 설계
- DBMS가 지원하는 기법을 선택하여 이용
  - 저장 구조
  - 접근 경로

# 저장 레코드의 양식 설계

## □ 고려 사항

- 데이터 타입
- 데이터 값의 분포
- 사용될 응용
- 접근 빈도

## □ 데이터 표현 및 압축 양식도 설계

# 레코드 집중의 분석 및 설계

- 집중 저장 : 물리적 순차성 지원
- 레코드 크기와 물리적 저장장치의 특성에 의존
- 순차 처리 : 큰 블록
- 임의 접근 처리 : 작은 블록

# 접근 방법 설계(1)

- 접근경로 : 저장레코드의 접근을 위한 절차
  - 저장 구조 : (인덱스를 통한) 접근 방법과 저장 레코드를 정의
  - 탐색 기법 : 응용에 적절한 접근 경로 정의
  
- 기본 및 보조 접근 경로의 설계
  - 기본 접근 경로 : 기본키를 이용한 기본 인덱스 이용
    - 초기 레코드 적재, 레코드의 물리적 위치, 기본키에 의한 검색
    - 주요 응용의 효율적 처리
  - 보조 접근 경로 : 보조키에 의한 인덱스 이용

# 접근 방법 설계(2)

## □ 물리적 설계의 옵션 선택 시 고려사항

- 응답 시간
- 저장 공간의 효율화
- 트랜잭션 처리도(throughput)

## □ 분석을 위한 감시 유틸리티 이용

- 시스템 카탈로그나 데이터 사전에 성능 통계를 수집,저장
- 설계 tune-up

# 데이터베이스 구현

- ❑ 목표 DBMS의 DDL로 기술된 명령문(스키마)의 컴파일 및 실행
  - ❑ DB 스키마, 공백 DB화일 생성
- ❑ 데이터 적재
- ❑ 트랜잭션 구현



# DB 설계 과정 요약

