4장 기본 설계

김동근
(kdgfox@gmail.com)

4.1 설계 과정

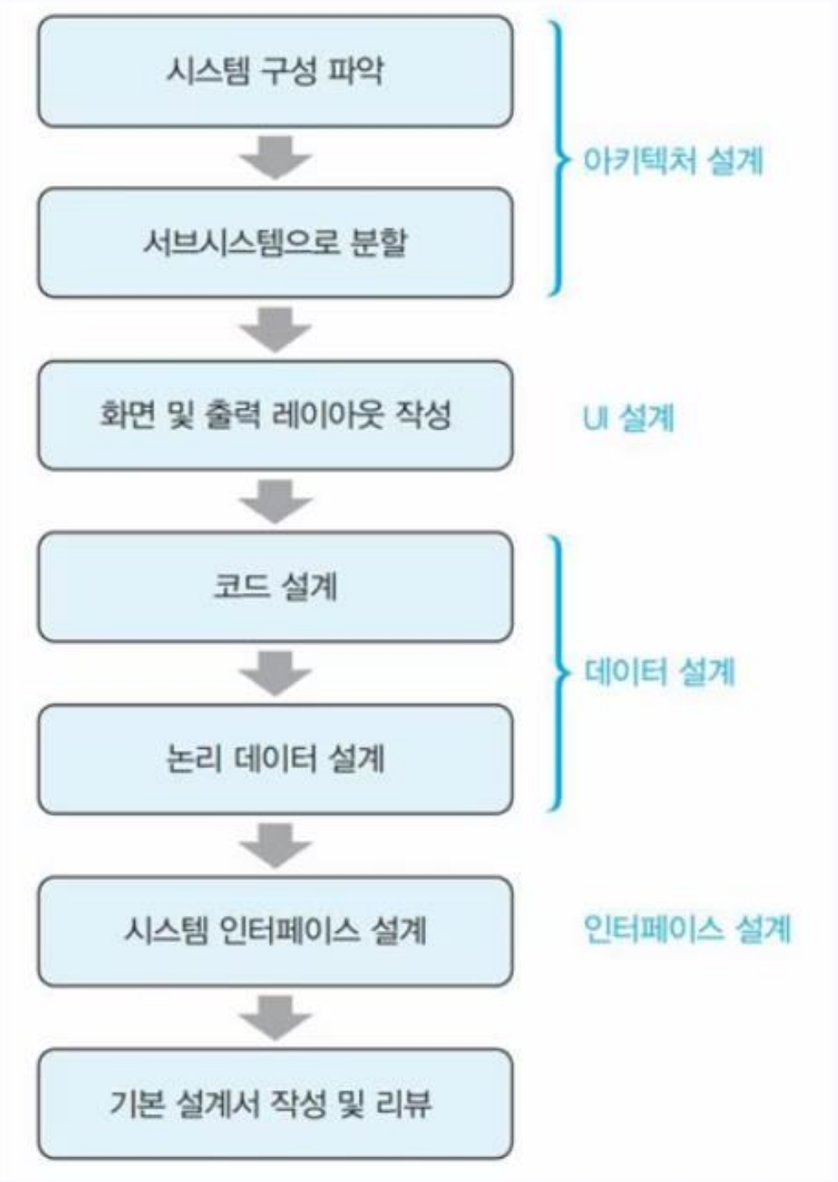
설계

- 문제를 해결하기 위한 SW의 컴포넌트를 구성해 나가는 것
- 기본 설계
- 응용 프로그램 개발에 기초가 되는 부분(뼈대)을 만드는 것

기본 설계 내용

- 아키텍처 설계 - 서브시스템으로 분할하기 위한 기능 목록, 인프라가 되는 하드웨어, 네트워크, SW 구성 요소 파악과 관계 정의.
- UI 설계 – 화면의 레이아웃과 화면의 관계 및 전환, 일괄 처리에 의한 인쇄물 의 레이아웃 설계
- 데이터 설계 – 코드 설계, 논리 데이터 설계
- 외부 인터페이스 설계 – 다른 시스템과의 연동 데이터 형식과 호출 방식

기본 설계 절차


시스템 구성 파악
서브시스템으로 분할
화면 및 출력 레이아웃 작성
코드 설계
논리 데이터 설계
시스템 인터페이스 설계
기본 설계서 작성 및 리뷰
아키텍처 설계
UI 설계
데이터 설계
인터페이스 설계

기본 설계서

요구 분석 명세의 내용을 정보 시스템의 단어로 대체한 것

어떻게 구현될지 결정한 것

시스템 개발의 기준

문서	누구를 위하여	어떤 내용
요구 분석 명세서	고객	정보 시스템으로 구현할 것
기본 설계	고객 개발자 협력 시스템	정보 시스템이 어떤 기능으로 만들어질 것 인지에 대해 기술
상세 설계	개발자	정보 시스템의 상세한 기능에 대한 것으로 바로 프로그래밍할 수 있는 자세한 설명

4.2 아키텍처 설계

시스템이나 소프트웨어의 전반적인 구조와 구성 요소를 정의하고 이들의 상호작용을 설계하는 과정

전략적인 시스템의 청사진

시스템 구성

하드웨어, 네트워크, SW 등 시스템 구조를 정의

응용 프로그램 구조를 정의

응용 프로그램 아키텍쳐

컴포넌트 구성과 연계 방법

기능 단위로 사용자 서비스, 인증 서비스, 결제 서비스 등

인터페이스

컴포넌트 간 통신 방식 (REST API, 메세지 큐 등)

데이타 흐름

데이타가 시스템 내에서 어떻게 이동하는지, 저장 구조

비기능 요건 고려

보안, 성능, 확장석, 유지 보수성

일반적인 아키텍쳐 유형

아키텍처 유형	설명
계층형 (Layered Architecture)	프레젠테이션, 비즈니스, 데이터 접근 등 계층으로 분리
클라이언트-서버	클라이언트가 요청 → 서버가 응답
마이크로서비스(MSA)	기능을 작고 독립된 서비스로 나누고 각각 배포
이벤트 기반(Event-driven)	이벤트 발생 → 처리 → 결과 전달
서버리스(Serverless)	인프라 관리 없이 함수 단위로 실행

4.3 UI 설계

Model(자료)

데이터와 비즈니스 로직을 관리

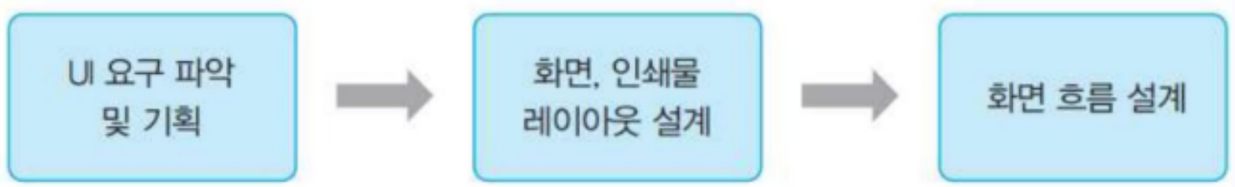
View(UI)

기능을 사용자에게 보여주는 외관

화면과 인쇄물

Control(처리)

사용자 입력을 처리하고 모델과 뷰를 연결



UI 요구 파악

- UI에 대한 요구와 범위를 분석
- UI/UX 전략을 세우기 위한 방법
- 퍼소나(Persona) - 경험이 동일한 사용자들을 묶은 후 가상의 인물로 정의
- 저니 맵(Journey Map) - 개별 퍼소나들이 제품 이용 흐름에 따라서 어떤 경험의 변화를 보이는지 시각화
- Elito Method - 발견된 주요 사실(key findings)를 가지고 그 의미와 사용자에게 제공할 가치, 구체적인 해결책을 같이 고민
- 어피니티 다이어그램(Affinity Diagram) - 인터뷰 등 필드 리서치 결과를 아래에서 위로 묶어 나가면서 개별 결과에서는 보이지 않던 가치를 찾음

화면 레이아웃 설계

와이어프레임

UI 기획 단계의 초기에 제작하는 것으로 페이지에 대한 개략적인 레이아웃이나 UI 요소등에 대한 뼈대를 설계하는 방법

각 페이지의 영역 구분 콘텐츠, 텍스트 배치 등을 화면 단위로 설계하며 화면 흐름도와 유사하다.

목업

디자인, 사용 방법 설명, 평가 등을 위해 와이어프레임보다 좀 더 실제 화면과 유사하게 만든 정적인 모형.

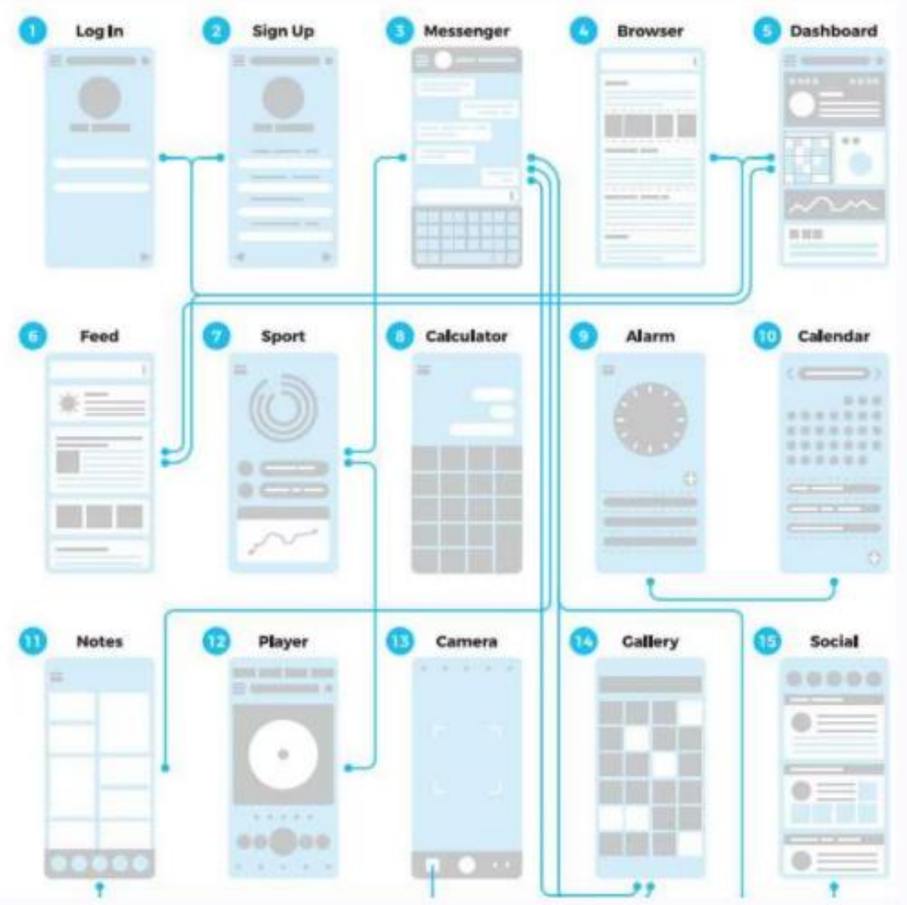
스토리보드

와이어프레임에 콘텐츠에 대한 설명, 페이지 간 이동 흐름 등을 추가한 문서

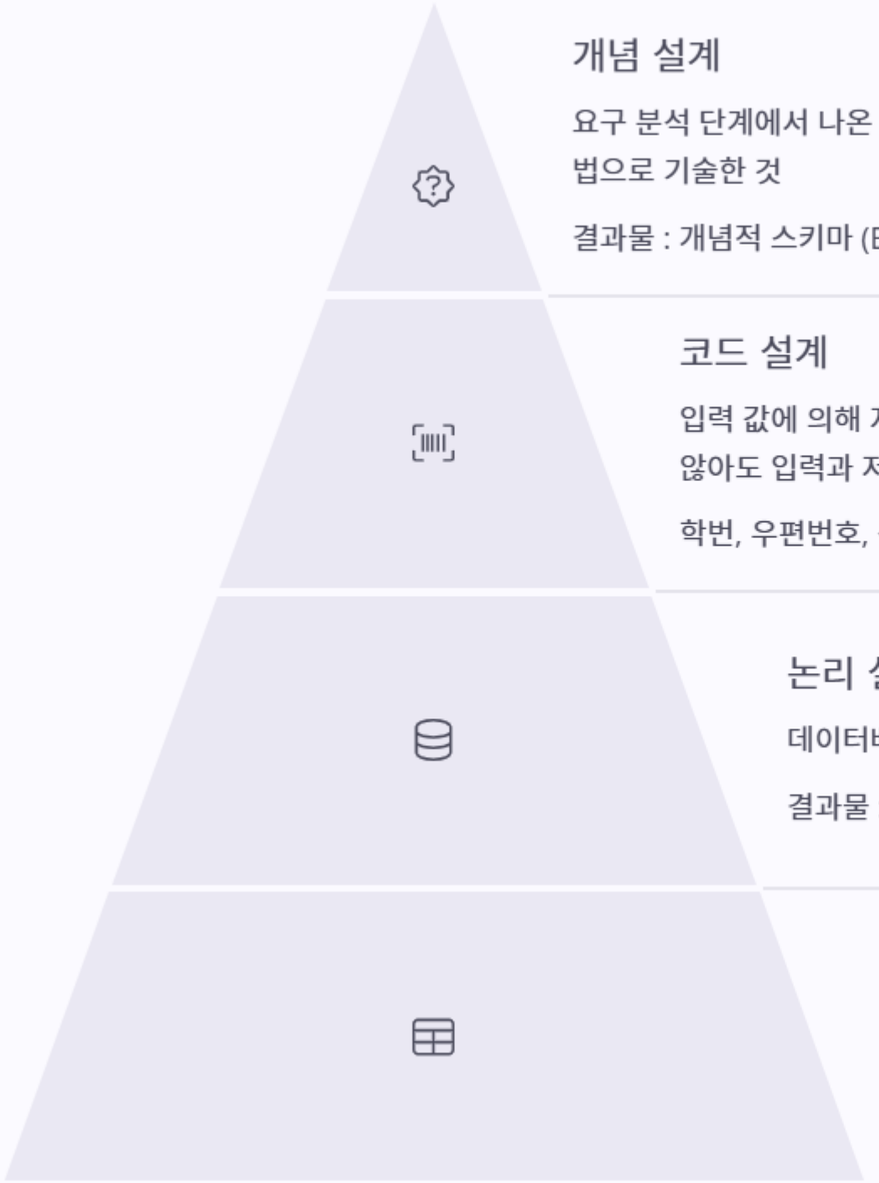
프로토타입

와이어 프레이이나 스토리 보드 등에 인터렉션을 적용한 것

예시



4.4 데이터 설계



개념 설계

요구 분석 단계에서 나온 결과(명세)를 ERD와 같은 DBMS에 독립적이고 고차원적인 표현기법으로 기술한 것

결과물 : 개념적 스키마 (ERD)

코드 설계

입력 값에 의해 자료 값에 편차가 생길 수 있는 정보나 반복되는 자료 값, 반복되지 않아도 입력과 저장의 효율을 위해 코드화

학번, 우편번호, 상품명, 주문 번호 등

논리 설계

데이터베이스의 논리적 구조를 정의, 정규화 과정을 통해 스키마를 도출

결과물 : 논리 스키마

물리 설계

목표 DBMS에 맞는 물리적 구조 설계(속성, 레코드 크기, 저장 공간의 효율화, 인덱스) 성능을 위해 역정규화

결과물 : 물리 스키마

개념설계(ER 모델)

ER(Entity Relataionship) 모델

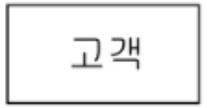
- 데이터 베이스 설계를 용이하게 하기위해 피터 첸(Peter chen)이 제안
- 개념적 설계를 위한 인기 있는 모델
- 구성요소

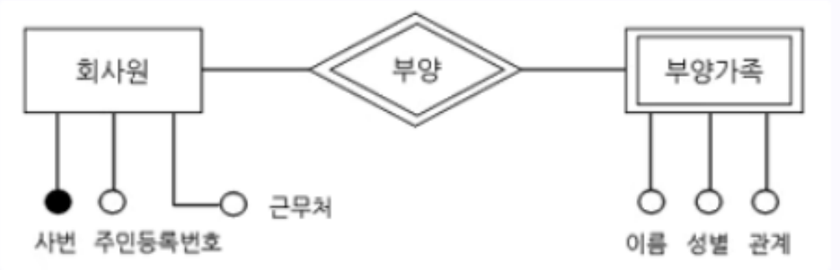
  개체(Entity), 속성(Attribute), 관계(relationship)

Entity

Entity란

- 사람, 사물, 개념, 사건과 같이 독립적으로 존재하면서 고유하게 식별이 가능한 실세계의 객체
- 다른 개체와 구별되는 이름을 가지고 있고, 각 객체만의 고유한 특성이나 상태, 즉 저장할 가치가 있는 중요 속성을 1개 이상 가지고 있는 대상
- 종류
  - 일반 엔티티　속성 중 키 속성이 있어 인스턴스를 구별할 수 있는 엔티티
  - 약 엔티티　　키 속성을 갖기 못하는 엔티티
- 표기 법 : 사각형으로 구성하면 사각형 안에 개체의 이름을 표기한다.





속성(Attribute)

- 개체나 관계가 가지고 있는 고유의 특성
- 더 이상 쪼갤 수 없는 정보의 단위
- 종류

  일반 속성, 키 속성, 다중 값 속성, 복합 속성, 유도 속성

- 표기법

속성 종류

1

일반 속성

Entity 또는 관계성의 성질을 나타내는 일반적인 단위

일반 속성은 원자 값을 의미한다.

◈ 예: 고객: 고객 아이디, 주소, 연락처, 적립금

2

유도 속성

▣ 속성들을 통해 산출되는 속성

◈ 예: 총점, 총 구매 금액

3

다중 값 속성

▣ 하나의 속성이 여러 개의 값을 가질 때를 의미한다.

▣ 이중의 작은 원으로 표시한다.

◈ 예: 취미

4

복합 속성

▣ 하나의 속성이 독립적인 여러 작은 부분들로 쪼개어 질 수 있는 경우를 의미한다.

▣ 복합 속성은 타원형으로 표시하고 구성은 일반 속성을 표시한다.

◈ 예: 주소(시, 구, 동) 카드 유효기간(년, 월), 전화번호(지역번호, 국번, 번호)

키 속성

엔티티의 대표 속성으로 개체 무결성을 유지 하는 주요 속성이다.

▣ 식별자(Identifier) 한 개체(Entity) 내에서 인스턴스를 구분할 수 있는 단일 속성 또는 속성 그룹

▣ 후보키(Candidate key) 개체 내에서 각각의 인스턴스를 구분할 수 있는 속성

▣ 기본키(Primary key) 후보키 중 인스턴스를 구분하는데 적합한 key (프로그램에서 자주 사용하거나, 데이터 양이 작은 key)

▣ 대체키(Alternate key) 후보키 중 기본 키로 선정되지 않은 key

▣ 복합키(Composite Key) 하나의 속성으로 기본키가 될 수 없는 경우 둘 이상의 속성을 묶어서 식별자로 정의

▣ 대리키(Surrogate kye) 식별자가 너무 길거나 여러 개의 속성으로 구성되어 있는 경우 인위적으로 추가

관계

- 두 Entity 간의 업무적인 연관성 또는 관련 사실
- 실체간 존재하는 상호강의 연관성으로 해당 실체와 관련된 업무가 수행되는 일련의 규칙으로 부터 정의됨.
- 관계성(Cardinality) : 1:1, 1:N, N:M
- 관계의 성질 : 필수(실선), 선택(점선)
- 표기법

관계를 설정하는 순서

1. 관계가 있는 두 실체를 실선(점선)으로 연결하고 관계를 부여
2. 관계 차수를 표현
3. 선택성을 표시

⊙ 예1) N:1관계 - 각 회사원은 한 부서에 소속되어 있다.

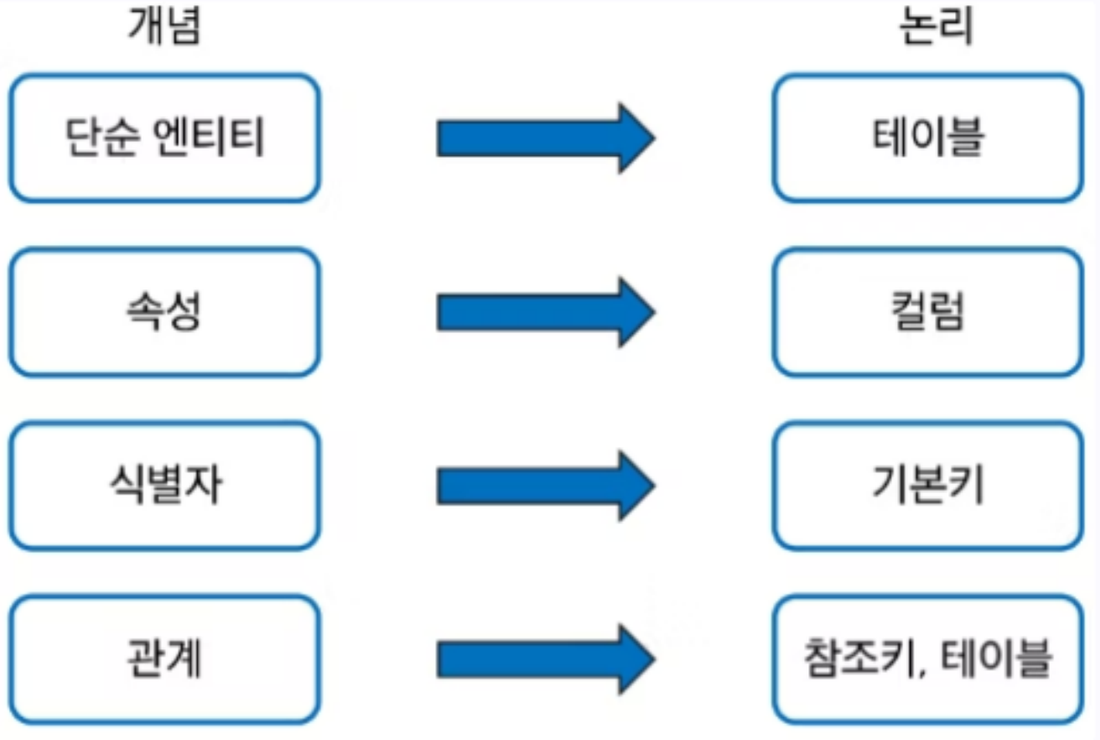
⊙ 예2) 1:1 관계 - 한 명의 고객은 꼭 한 대의 승용차만 보유한다.

⊙ 예3) N:M 관계 - 각 학생은 여러 개의 과목을 수강할 수 있고 반대로 한 과목에 여러 학생이 수강한다.

논리적 DB 모델링

- 개념적 데이타베이스 모델링 단계에서 정의된 ERD를 Mapping Rule을 적용하여 관계형 데이타베이스 이론에 입각한 스키마를 설계하는 단계와 이를 이용하여 필요한 경우 정규화 하는 단계로 구성한다.
- 기본키(Primary key)
  - 후보 키 중에서 선택한 주 키
  - 널(null)의 값을 가질 수 없다(not null)
  - 동일한 값이 중복해서 저장될 수 없다.(Unique)
- 참조키(Foreign Key)
  - 관계를 맺는 두 엔티티에서 서로 참조하는 릴레이션의 속성으로 지정되는 키

Mapping Rule



ER 모델로부터 테이블로 변환 원칙

- 1:N 관계의 경우 1쪽의 기본키를 N쪽으로 관계 속성으로 생성한다.
- 1:1 관계의 경우는 어느 쪽의 기본키를 다른 쪽에 관계 속성으로 생성해도 된다.
- N:M 관계의 경우는 양쪽의 기본키 둘을 합쳐 복합 키를 만들고 이 복합 키를 기본키로 하는 별도의 테이블로 생성한다.
- 다중 값 속성은 해당 엔티티의 기본키와 다중 값 속성을 합쳐서 복합키를 만들고 그 복합 키를 기본키로 하는 별도의 테이블을 생성
- 관계 속성은 1:N인 경우 N에 속성으로 1:1은 내용의 흐름을 보고 처리, N:M인 경우 관계 테이블의 속성으로 표시한다.

정규화(Normalization)

- 관계형 데이터베이스 설계에서 중복을 최소화하게 데이터를 구조화하는 프로세스

- 정규화 목적
  - 데이터 베이스의 변경 시 이상 현상 제거
  - 데이터 베이스 구조 확장 시 재 디자인 최소화
  - 사용자에게 데이터 모델을 더욱 의미 있게 작성하도록 함

중복 제거

관계형 데이터의 중복을 제거합니다.

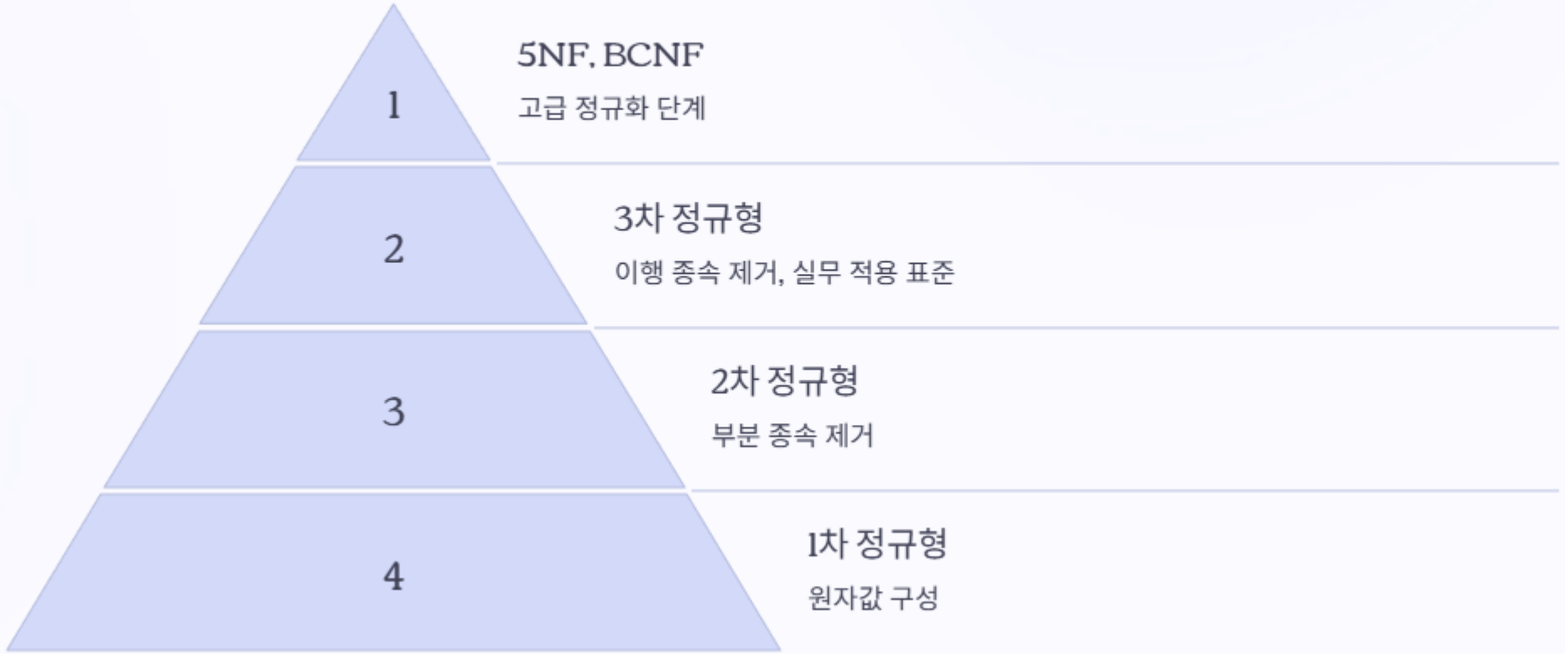
일관성 확보

데이터의 일관성을 확보합니다.

성능 최적화

입력, 수정, 삭제 성능을 최적화합니다.

정규화 단계 개요



https://en.wikipedia.org/wiki/Database_normalization

제 1 정규화

- 모든 속성은 반드시 하나의 값을 가져야 한다.

학번	이름	전화번호
001	홍길동	010-0000-0000, 010-1111-1111

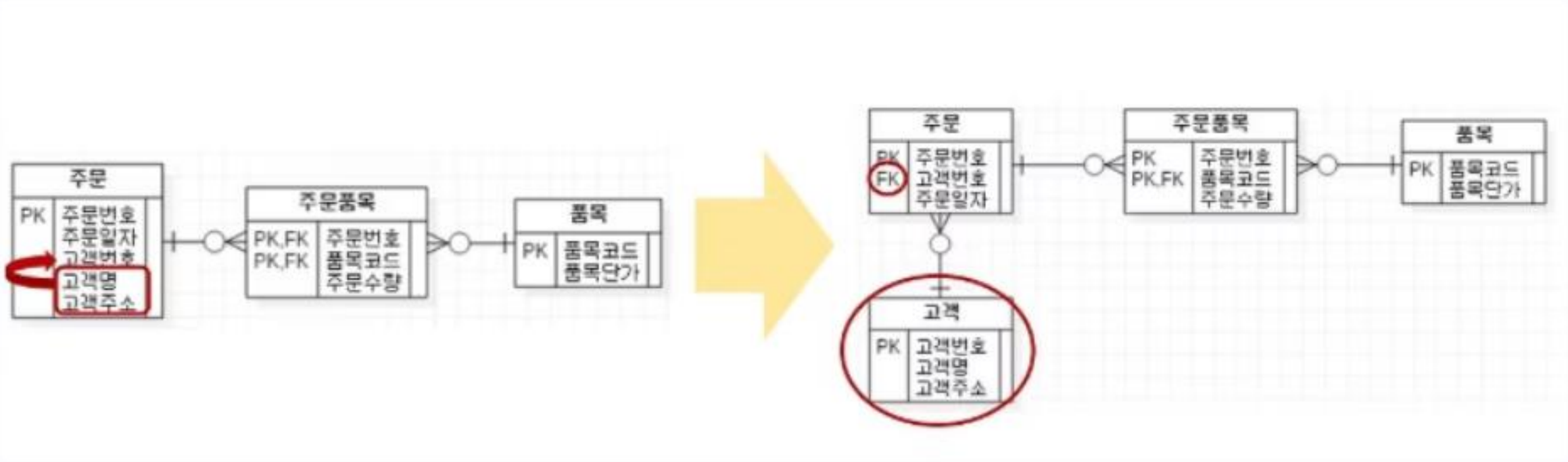
순번	학번	전화번호
1	001	010-0000-0000
2	001	010-1111-1111

제 2 정규화

- 모든 속성은 반드시 기본키에 종속되어야 한다.
- 부분 함수 종속을 제거한다. (기본키 일부에만 종속되는 경우 제거한다.)

제 3 정규화

- 이행 함수 종속 제거
- 기본키가 아닌 속성에 종속 관계 있는 속성을 제거하기



물리적 설계

물리적 설계

효율적이고 실현 가능한 최적화된 물리적 데이터베이스 구조를 개발하는 과정을 의미한다.

물리적 설계 관심사

물리적 설계 단계에서는 데이터를 물리적 장치에 어떻게 저장하느냐 하는 것에 관심을 갖게 된다.

주요 입력 자료

물리적 설계 단계의 주요 입력 자료는 논리적 설계에서 작성된 논리적 구조, 사용할 DBMS 및 운 영 체제의 특성과 사용자 운영 요구사항 이다.

물리적 설계의 출력

저장 레코드 위치 및 사용할 접근 방법의 명세이다. 물리적 설계의 주요 목 적은 성능과 운영 비용간의 최적의 균형을 유지하는 것이다.

물리적 설계 고려 요소

응답시간

데이터베이스 트랜잭션을 요청해서 응답을 얻기까지의 시간

메모리 효율성

데이터베이스 파일과 액세스 패스 구조에 대한 메모리의 총 공간

트랜잭션 산출물

일정 단위 시간 동안 처리되는 트랜잭션의 평균 수

DB 튜닝

인덱스 생성, 트리거 작성, 역정규화

튜닝의 기법 - 인덱스

인덱스 선정 기준

분포도가 좋은 칼럼은 단독적으로 생성하여 활용도 향상(15% 이내)

인덱스 생성 시 고려사항

지나치게 많은 인덱스는 Overhead를 발생시킴

인덱스 적용

작은 테이블에는 인덱스를 만들지 않음

인덱스 설정기술

중,대형 테이블에 인덱스를 설정하는 기술적 고려사항

인덱스 선정 기준: 자주 조합되어 사용되는 칼럼의 경우에는 결합 인덱스 생성 고려, 인덱스간의 역할 정의, 수정이 빈번히 일어나지 않는 칼럼을 인덱스로 사용, Access Path에 의해 인덱스 결정

인덱스 생성 시 고려사항: 인덱스가 새로 추가될 경우 기존 인덱스에 미치는 영향 고려, 분포도가 좋지 않은 칼럼을 인덱스로 하지 말아야 함, 분포도가 양호한 칼럼도 처리 범위에 따라 분포도가 나빠질 수 있음, 인덱스 사용 원칙을 준수하고 join 할 경우 인덱스 사용 주의해야 함

튜닝의 기법 - 역정규화

- 성능을 개선하기 위해 DB의 구조를 조정하는 행위
- 중복 컬럼을 허용하거나 테이블을 재정의하여 성능을 개선한다.

컬럼 중복 조정하기

- 중복 컬럼이란 기존 컬럼을 정확히 복사한 컬럼을 의미한다.
- 빈번하게 조회되거나 주요한 질의 처리시 테이블 조회를 용이하도록 중복을 고려
- 중복 컬럼 고려 대상
  - 여러 테이블의 접근을 줄이기 위해
  - 계산 등을 포함하는 빈번하거나 주요한 질의 처리 성능을 개선하기 위해

테이블 재정의

- 테이블을 수직적으로 분할

한 테이블의 일부 컬럼들만 빈번하게 조회할 경우 수직적으로 분할하고 1:1 관계로 변경한다.

- 테이블을 수평적으로 분할

테이블의 최근 데이타만 빈번하게 조회하는 경우 최근 3개, 6개월, 1년 단위의 데이타를 위한 테이블과 그외 테이블로 분리한다.

- 요약 테이블 생성

메인 화면이나 자주 조회하는 페이지에서 테이블 정보를 요약한 내용이 표시될 경우 요약 테이블을 작성해서 단순 쿼리 질의로 변경한다.

- 테이블 통합

1:1 관계의 테이블 중 자주 함께 참조되는 테이블인 경우