

03. 개념 모델 & 논리 모델 & 물리 모델

3.1 개념 모델(Conceptual Model)

- 주제 영역 모델, 비즈니스 모델
- 핵심 엔터티로 구성되는 최상위 수준 모델
- 주제 영역의 핵심적인 중요 엔터티와 주요 속성 도출
- 개념모델의 엔터티는 논리/물리 에서 그대로 구조로 존재
- 모델링 과정에서 변하지 않을 뼈대 생성
- 요구사항을 이해가능한 데이터로 간결하게 표현이 목표
- 구축 당사자 사이의 커뮤니케이션이 중요

- 1) 요구분석

- 데이터 관점(필요한 데이터, 데이터 구조 등)의 요구 분석

-상세 인터뷰 수행

2) 중요 엔터티 선별 : 핵심적인 엔터티만 간결하게 선별

3) 엔터티 정의: 데이터 성격에 맞도록 규명

4) 식별자 정의 : 엔터티와 식별자는 한 몸

5) 엔터티 통합 : 유사한 데이터 일반화 , 모델 오너십 결정

6) 엔터티 간 관계 도출

3.2 논리 모델(Logical Model)

- 개념 모델 상세화 작업
- 관계를 포함한 모든 데이터 요소가 도출
- 구조적으로 완성된 모델
- **삭제할 엔터티나 속성이 없는 모델**

- 1) 엔터티 정의

- 엔터티 상세화, 실체/행위/목적/기준 엔터티 도출

- 데이터 성격(정체성)에 맞는 도출

2) 관계 도출

3) 속성 도출 : 성능문제 추측은 무리

4) 주 식별자 확정:엔터티 관계 고려

5) 정규화

6) 이력관리:이력/내역 구분

7) 논리 모델 검증

- 현행엔터티 or 어플리케이션과 매핑 검증, 코드 맵핑

3.3 물리 모델(Physical Model)

- 물리 모델링에 DBMS 실행(구현)포함
- 인덱스,파티션, 테이블 타입, 뷰 등 포함
- **성능 최적화 목표:**
- 논리모델 틀 안에서 엔터티 합체/분해 변경 가능**
- 개발에 쓰이는 실전 모델로 커뮤니케이션의 도구는 아님
- 전체 엔터티 공통 속성은 구현 직전에 수행
- 인덱스 설계:데이터베이스 구현 이후 개발 중에서
- 인덱스 설계 중 주 식별자 순서 변경 가능

- 1) 서버 타입 모델의 변환 : 슈퍼-서브타입 모델 검토

2) 엔터티 합체와 분해: 1:1 관계 두 엔터티

3) 비정규화 :

- 테이블 중복 : 특정 성능 문제 해결을 위해서만 사용

4) PK(Primary key) 확정

- 핵심 상위 엔터티에 PK를 확정해 사용, 하위 파급효과 감소

5) 테이블 파티션 확정

- 관리 측면과 가용성, 데이터 백업 정책 고려

6) 데이터 저장 방법 확정

7) 인덱스 설계

8) 뷰 설계

9) 시스템 속성 추가

04. 정규화(Nomalization)

4.1 정규화란?

- 속성 간 부정확한 종속성을 제거, 더는 분해 될 수 없는 엔터티로
- 아노말리 현상을 최소화
- 반드시 정규형을 채택, 성능의 중요한 요건에만 비정규형 고려

4.2 정규화의 목적

- 안정성, 확장성
- 아노말리 발생방지 → 무결성 획득 → 데이터 품질 향상
- 안정성/신뢰도 상승
- 함수 종속으로 결정된 데이터 구조는 견고, 확장성 우수

4.3 아노말리(Anomaly)

- 데이터 이상현상, 중복데이터가 원인
- 업데이트 아노말리, 삭제 아노말리, 삽입 아노말리
- 중복데이터 :정합성을 제어할 수 없음

4.4 함수 종속(Functional Dependency)

- 데이터 종속성 : 함수 종속, 다가 종속, 조인 종속 등
- 함수종속:엔터티 내 대표(식별자)과 나머지 속성 연관 관계
- 업데이트 아노말리, 삭제 아노말리, 삽입 아노말리
- 중복데이터 :정합성을 제어할 수 없음
- 결정자(X) vs 종속자(Y) : Y= f(X)
- $X \rightarrow Y \rightarrow Z$: 이행종속(3정규형 관련)

4.5 정규형의 종류

1정규형

- 모든 속성은 반드시 하나의 값(단일값,원자값)을 가져야 함
- 다가속성 : 1:M 의 관계 되도록 새로운 릴레이션 생성
- 복합속성:의미에 따라 분리
- 인스턴스 대상 : 중첩 릴레이션

2정규형

- 후보 식별자 속성과 일반 속성 간의 종속성에 기반
- 부분 함수 종속으로 바라생한 중복데이터 제거

3정규형

- 이행적 종속성과 관련(기존 종속성에 신규 종속성 추론)
- 부분 함수 종속으로 발생한 중복데이터 제거

BC 정규형(Boyce-Codd Normal Form)

- 3정규형 보강 , 모든 결정자는 주 식별자 이어야 함

4정규형

- 다가 종속(MVD) 개념 기반
- 다가 종속이 발생한 릴레이션에서 새로운 엔터티 생성으로 제거

5정규형

- 조인 종속 : 무손실 조인, 비부가적 조인 분해

구분	제거 대상	특징
1정규화	다가 속성, 복합 속성, 반복 속성, 중첩 릴레이션 제거	속성이 추가되거나 일대다(1:M) 관계의 릴레이션이 추가되며 관계를 상속시킴
2정규화	부분 종속 제거	일대다(1:M) 관계의 릴레이션이 추가되며 관계를 상속받음
3정규화	이행 종속 제거	일대다(1:M) 관계의 릴레이션이 추가되며 관계를 상속받음
BC정규화	종속자가 키(Key)에 포함된 함수 종속 제거	모든 결정자는 키(Key)이어야 한다는 관점에서 3정규형과 동일
4정규화	다가 종속 제거	다가 속성의 개수만큼 일대다(1:M) 관계의 릴레이션이 추가됨
5정규화	조인 종속 제거	조인 종속이 존재하는 릴레이션이 사용하기 편함, 지나치게 이 상적인 정규형

4.6 정규형과 성능

- 정규화가 성능 저해한다는 건 오해
- 조회/인서트 성능
- 정규화 후 중복 최소화 인스턴스 크기 감소 한블럭 저장 증가
- 화면 구성 고려 (ex, 목록/상세 따라 비정규형 고려)
- 성능 이슈는 가장 빠른 단계에서 검토