

chap 1 정리하기

1. 데이터 모델링의 이해

1. 데이터 모델링의 이해

1. 데이터 모델의 이해

- 모델링
 - 모델링의 정의: 모델을 만들어가는 일 자체 → 현실세계를 추상화, 단순화, 명확화 하기 위해 일정한 표기법에 의해 표현하는 기법
 - 모델링의 특징: 추상화, 단순화, 명확화
 - 모델링의 관점: 데이터관점(what), 상관관점(vs), 프로세스관점(how)
- 데이터 모델링
 - 데이터 모델링: 정보시스템을 구축하기 위해 해당 업무에 어떤 데이터가 존재하는 지 또는 업무가 필요로 하는 정보는 무엇인지를 분석하는 방법
 - 데이터 모델의 기능: 가시화, 명세화, 구조화, 문서화, 관점 제공, 표현방법 제공
 - 데이터 모델링의 중요성: 파급효과, 간결한 표현, 데이터 품질(중복, 비유연성, 비일관성)
 - 데이터 모델링의 진행: 개념적 → 논리적 → 물리적 (추상 → 구체)
 - 데이터독립성 ↔ 데이터종속성 ; 구조, 독립성, 사상(mapping)
 - 데이터 모델링의 중요한...
 - 데이터 모델링의 세 가지 요소: thing, attribute, relationship
 - 단수와 집합의 명명
 - ERD: 데이터 모델 표기법
 - 좋은 데이터 모델: 완전성, 중복배제, 업무규칙, 데이터재사용, 의사소통, 통합성

2. 엔터티

- 엔터티: 업무에 필요하고 유용한 정보를 저장하고 관리하기 위한 집합적인 것
- 특징: 업무에서 필요로 하는 정보, 식별가능, 인스턴스의 집합, 업무프로세스에 의해 이용, 속성 포함, 관계의 존재

- 분류: 유무형, 발생시점(기본, 중심, 행위)
- 명명: 현업용어, 약어, 단수명사, 유일성, 생성의미

3. 속성

- 속성: 업무에서 필요로 하는 인스턴스로 관리하고자 하는 의미상 더 이상 분리되지 않는 **최소**의 데이터 단위. 엔터티를 설명하고 인스턴스의 구성요소가 됨.
- 관계
 - 한 개의 엔터티는 두 개 이상의 인스턴스의 집합이어야 함
 - 한 개의 엔터티는 두 개 이상의 속성을 가짐
 - 한 개의 속성은 한 개의 속성값을 가짐
- 특징
 - 업무에서 필요하고 관리하고자 하는 정보여야 함
 - 정규화 이론에 근간하여 정해진 주식별자에 함수적 종속성을 가져야 함
 - 하나의 속성에는 한 개의 값만을 가짐. 다중값일 경우 별도의 엔터티를 이용하여 분리
- 분류
 - 속성: 기본속성, 설계속성, 파생속성
 - 구성방식: PK, FK, 일반
- 도메인: 속성이 가질 수 있는 값의 범위
- 명명: 현업용어, 서술x, 약어x, 유일성

4. 관계

- 정의: 엔터티의 인스턴스 사이의 논리적인 연관성으로서 존재의 형태로서나 행위로서 서로에게 연관성이 부여된 상태
- 페어링: 엔터티 안에 인스턴스가 개별적으로 관계를 가지는 것
- 분류: 연관관계, 의존관계
- 표기법: 관계명, **관계차수(1:1, 1:M, M:N)**, 관계선택사양
- 방법
 - 관계 체크사항: 연관규칙, 정보의 조합, 규칙 서술, 동사
 - 관계 읽기: 기준 엔터티, 대상 엔터티, 관계명

5. 식별자

- 식별자: 하나의 엔터티에 구성되어 있는 여러 개의 속성 중에 엔터티를 대표할 수 있는 속성을 의미하며 하나의 엔터티는 반드시 하나의 유일한 식별자가 존재해야 함.

식별자는 논리 데이터 모델링 단계에서 사용, 키는 데이터베이스 테이블에 접근을 위한 매개체로서 물리 데이터 모델링 단계에서 사용

- 특징
 - 주식별자: 유일하게 구분, 최소의 수, 자주 변하지 않음, 반드시 값 존재
 - 대체식별자: 주식별자와 같음
 - 외부식별자: 참조무결성 제약조건에 따른 특징을 가지고 있음
- 분류
 - 대표성: 주식별자, 보조식별자
 - 스스로 생성: 내부식별자, 외부식별자
 - 단일 속성: 단일식별자, 복합식별자
 - 대체여부: 본질식별자, 인조식별자
- 도출기준
 - 자주 이용되는 속성을 주식별자로 지정
 - 이름으로 기술되는 것들은 가능하면 주식별자로 지정하지 않음
 - 복합 주식별자로 구성할 경우 너무 많은 속성이 포함되지 않도록

2. 데이터 모델과 성능

1. 성능 데이터 모델링

- 정의
 - 데이터 모델 구조에 의해 성능 저하
 - 데이터가 대용량이 됨 → 성능 저하
 - 인덱스 특성을 고려하지 않고 생성함 → 성능 저하
 - 위와 같은 경우에서 성능을 향상시키는 것
- 수행 시점: 분석/설계 단계
- 순서 및 고려: 정규화, DB 용량 산정, 트랜잭션 유형 파악(유형에 따라 반정규화 필요), 여러 조정 수행, 성능 관점에서 데이터 모델 검증

2. 정규화와 성능

- 정규화: 데이터를 결정하는 결정자에 의해 함수적 종속을 가지고 있는 일반속성을 의존자로 하여 입력/수정/삭제/이상을 제거
-

3. 반정규화와 성능

- 반정규화: 정규화된 엔터티, 속성, 관계에 대해 시스템의 성능 향상, 단순화를 위해 중복, 통합, 분리 등을 수행
- 적용 방법: 반정규화 대상 조사 → 타 방법 유도 검토 → 반정규화 적용
- 기법: 테이블 반정규화, 칼럼 반정규화, 관계 반정규화

4. 대량 데이터에 따른 성능

- 테이블 분할!
- 성능 저하의 원인
 - 하나의 테이블에 데이터 대량 집중
 - 하나의 테이블에 여러 칼럼 존재
 - 대량의 데이터가 처리되는 테이블 → 인덱스 적절하게 구성
 - 대량의 데이터가 하나의 테이블에 존재
 - 칼럼이 많아지는 경우
- 처리방법
 - Range Partition: 날짜 숫자값으로 분리 가능한 경우. 데이터 삭제가 쉬움. RANGE로 구분
 - List Partition: 대용량 데이터를 특정값에 따라 분리, 저장. 삭제가 어려움. LIST로 구분
 - Hash Partition: 지정된 HASH 조건에 따라 해쉬 알고리즘을 적용. 데이터 확인이 어렵고 삭제 불가
- 수평/수직 분할 절차
 - 데이터 모델링 완성 → DB 용량 산정(칼럼 수 고려) → 트랜잭션 처리 패턴 분석 → 칼럼 단위 처리인지 로우 단위 처리인지 분석
 - 칼럼수가 많으면 1:1로 분리 가능한지 검토.
 - 칼럼수는 적는데 데이터량이 많다면 파티셔닝 고려

5. 데이터베이스 구조와 성능

- 슈퍼/서브타입 모델의 성능 고려 방법
 - 슈퍼타입: 공통부분, 서브타입: 차이가 있는 속성.
 - 논리적데이터 모델에서 이용, 분석 단계에서 많이 쓰임
 - 물리적 데이터 모델로 설계 시 문제점이 나타남
- 슈퍼/서브타입 데이터 모델의 변환
 - 성능저하의 원인
 - 트랜잭션은 항상 일괄 처리 vs 테이블 개별 유지 → union 연산 시 저하
 - 트랜잭션은 항상 서브타입을 개별 처리 vs 테이블 하나로 통합됨 → 불필요하게 많은 양의 데이터 집약
 - 트랜잭션은 항상 슈퍼+서브 타입을 공통으로 처리하는데 개별로 유지되어 있거나 하나의 테이블로 집약되어 성능이 저하되는 경우
 - 변환 기준: 데이터의 양과 해당 테이블에 발생하는 트랜잭션의 유형
 - 소량의 데이터: 데이터 처리의 유연성 고려 → 가급적 1:1
 - 데이터가 대량: 3가지의 변화 방법
 - 1 to 1: 개별 테이블 유지
 - plus: 슈퍼+서브 테이블
 - single: 하나의 테이블
- PK/FK와 성능
 - 인덱스 설계 중요
 - PK/FK 설계 중요
 - PK 순서 중요
 - FK 인덱스 중요

6. 분산 데이터베이스와 성능

- 개요: DB를 연결하는 빠른 네트워크 환경을 이용하여 여러 지역 여러 노드로 위치시켜 사용성/성능 등을 극대화. 분산되어 있는 DB를 하나의 가상 시스템으로 사용
- 투명성
 - 분할 투명성, 위치 투명성, 지역사상 투명성, 중복 투명성, 장애 투명성, 병행 투명성

- 장점: 시스템 용량 확장, 신뢰성, 가용성, 효율성, 융통성, 빠름, 요구 수용 증대
- 단점: 개발 비용, 오류, 처리 비용, 복잡, 불규칙, 통제, 데이터 무결성 위협
- 기법
 - 테이블 위치 분산
 - 테이블 분할 분산
 - 테이블 복제 분산 (프로젝트에 자주 쓰임)
 - 테이블 요약 분산
- 효과적 적용
 - 성능이 중요한 사이트에 적용
 - 백업 구성 가능
 - 실시간 동기화 요구되지 않을 때 적용
 -

▼ 참고

[SQLD][과목 1] 제 2 장 : 데이터모델과 성능

제 1절 : 성능데이터 모델링의 개요 1. 성능 데이터 모델링의 정의
데이터 모델 구조에 의해 성능 저하 데이터가 대용량이 됨으로 인
해 불가피하게 성능 저하 인덱스 특성을 충분히 고려하지 않고 인

🔗 <https://ysyblog.tistory.com/63>



[SQLD 핵심 요약] 데이터 모델과 성능

*성능 데이터모델링 -데이터베이스 성능 향상이 목적으로 설계단
계의 데이터 모델링 때부터 성능과 관련된 사항이 데이터 모델링
에 반영될 수 있도록 하는것 -데이터의 증가가 빠를수록 성능저하

🔗 <https://javagirl.tistory.com/47?category=853450>

