[데이터 모델링]

데이터 모델링 : 현실세계 → 데이터베이스 (추상화)

업무프로세스 → 데이터모델링 표기법 → 규칙정의

⇒ 추상화 (간략), 단순화 (쉬운이해), 명확성 (의미)

① 개념적 모델링 : 중요한 부분, 개념적ERD작성 - 전사적

② 논리적 모델링 : 식별자 도출, 관계정의, 정규화 (독립성) ↳ 재사용성

③ 물리적 모델링 : 실제 DB구축, 테이블 / 함수 / 인덱스

모델링관점

데이터 → 구조분석, 정적분석

프로세스 : 작업 → 시나리오 분석, 도메인 분석, 동적분석

비지니스프로세스

관계 → CRUD 분석

*작성절차

[ERD (Entity Relationship Diagram)]

데이터모델링의 표준, 엔터티 간 관계정의 모델링 방법 (↳ 이해쉽게, 복잡하지 않게)

- 중요엔터티 : 왼쪽상단
- 관계참여도 : 엔터티 간 참여하는 관계수
- 관계필수여부

- 데이터 모델링 고려사항

① 독립성 (정규화) ② 고객요구사항 표현

③ 데이터 품질 확보

(데이터 표준정의, 표준준수율 관리)

3-Level Schema [3층 스키마] : DB의 독립성 확보 방법 (3단계 계층으로 분리해서 뷰(view))

사용자, 설계자, 개발자 → (관점) → 데이터베이스 → (기술) → ANSI 표준 (미국표준협회) (관계정의)

데이터복잡도↑, 중복제거, 요구사항변화 대응력↑, 관리 / 유지보수 비용↓

- 독립성 < 논리적 : 개념스키마 (영향) ✗→ 외부스키마
  물리적 : 내부스키마 ✗→ 개념스키마

3층 논리적 물리적

- 외부 : 사용자 관점, 응용프로그램이 접근하는 DB
- 개념 : 설계자 관점, 통합데이터베이스 구조
- 내부 : 개발자 관점, 물리적 저장구조

[엔터티 (Entity)] : 업무에서 관리해야하는 데이터 집합

변별가능한 객체, 저장 / 관리 되어야하는 데이터

저장할수있는 장소 / 사람 / 사건 / 개념 / 물건 ...

[엔터티 특징]

- 식별자 (유일)
- 인스턴스 집합 (2개이상) ↳ 릴레이션 / 테이블이 가질수있는 값 (행의수)
- 속성
- 관계 (최소 한개이상)
- 업무 (관리되어야 하는 집합)

[엔터티 종류]

1) 유형 / 무형

① 유형엔터티 (지속적으로 사용, 업무에서 도출)

② 개념엔터티 (물리적 형태X, 개념적)

③ 사건엔터티 (비지니스 프로세스 실행에서 생성)

2) 발생시점 기준

① 기본엔터티 (키엔터티, 독립적으로 생성)

② 중심엔터티 (기본 - 행위 중간)

③ 행위엔터티 (2개이상의 엔터티로 부터 발생)

[속성] : 엔터티가 가지는 항목

더이상 분해 / 분리되지 않는 단위, 인스턴스의 구성요소

- 특징 : 업무에서 관리되는 정보
  하나의 값만 가짐
  주식별자에 함수적으로 종속

[속성 종류]

1) 분해여부

① 단일속성 (하나의 의미로 구성)

② 복합속성 (여러개 의미) ex. 주소 : 시/구/동 ...

③ 다중값속성 (속성에 여러개값 가질수있음) 엔터티로 분해

2) 특성

① 기본속성 (본래의 속성)

② 설계속성 (모델링과정에서 발생, 유일한 값 부여)

③ 파생속성 (다른 속성에 의해 생성)

* 도메인 (Domain) : 속성이 가질수있는 범위

[관계 (Relationship)] : 엔터티 간의 관련성

① 존재관계 : 엔터티 간 상태 (존재여부)
고객 (회원가입) → 소속 → 관리점 (지점명, 담당자)

② 행위관계 : 엔터티 간 행위
계좌 (개설) → 발주 → 주문 (주문이력)

[관계차수 (Cardinality)]

두개의 엔터티 간 참여하는 수

① 1대1 관계 (표현)

- 완전 1대1 : 반드시 하나존재 (= 필수적관계 '|')
- 선택적 1대1 : 하나이거나없거나 (= 선택적관계 'O')

② 1대N 관계

엔터티 행에 하나 — 다른 엔터티 값이 여러개

③ M대N 관계

두엔터티가 서로 여러개의 관계

관계형 데이터베이스에서 M대N 관계의 조인은 카테시안곱 발생 → 추가적 엔터티 도출 → 1대N, N대1로 해소해야함

[식별관계와 비식별관계]

① 식별관계 : 강한개체가 다른엔터티와 관계를 가질때 다른 엔터티에게 기본키를 공유하는 것 (실선)

강한 개체 (Strong Entity) ⟷ 약한개체 (Week Entity)

독립적으로 존재할수있는 개체 / 종속된 개체

- 강한 개체의 기본값이 변경되면 식별관계에 있는 (기본키를 공유받은) 엔터티의 값 (약한개체) 도 변경됨

② 비식별관계 : 강한 개체의 기본키를 다른 엔터티의 기본키가 아닌 일반 컬럼으로 관계를 가지는 것 (점선)

[엔터티 식별자 (Entity Identifier)]

엔터티를 대표할수 있는 유일성을 만족하는 속성

[주식별자 (기본키, Primary Key)]

최소성, 대표성, 유일성 (인스턴스를 유일하게 식별), 불변성 (자주 변경되지 X)

[키의 종류]

후보키 (Candidate key) : 유일성 & 최소성 만족

기본키 (Primary key) : 후보키 중 엔터티를 대표

슈퍼키 (Super key) : 유일성 만족O, 최소성 만족 X

대체키 (Alternate key) : 여러 후보키 중 기본키를 설정하고 남은 키

외래키 (Foreign key) : 하나 혹은 다수 다른 테이블의 기본 필드를 가리킴. 참조무결성 (Reference Integrity) 확인을 위해 사용 (허용된 값만 DB에 저장하기 위해)

[식별자의 종류]

1) 대표성

주식별자 : 유일성 & 최소성 만족. 엔터티를 대표. 다른 엔터티와 참조 관계로 연결 가능.

보조식별자 : 유일성 & 최소성 만족. 대표성 만족 X

2) 생성여부

내부식별자 : 엔터티 내부에서 스스로 생성.

외부식별자 : 다른 엔터티와의 관계로 인해 생성.

3) 속성의수

단일식별자 : 하나의 속성으로 구성

복합식별자 : 두개이상의 속성으로 구성

4) 대체여부

본질식별자 : 비지니스 프로세스에서 생성

인조식별자 : 인위적으로 생성 (후보식별자 중 주식별자로 선정할 것이 없거나 주식별자가 너무 많은 컬럼으로 되어있는 경우사용)

[데이터 모델과 성능]

정규화 : 데이터의 일관성, 데이터 중복 최소화, 데이터 유연성 최대화를 위한 방법. 데이터를 분해하는 과정

- 중복제거 → 모델의 독립성 확보 (불필요한 데이터의 입력 = 이상현상 Anomaly 방지)
- 비지니스의 변화 발생 → 모델의 변경 최소화가능
- 제1~5 정규화가 있지만, 실질적으론 제3까지만 수행

[정규화 절차]

1) 제1정규화

- 속성 (Attribute) 의 원자성을 확보
- 기본키 (Primary key) 설정 : (Y는 X에 X → Y 함수적으로 종속) 정규화는 함수적 종속성을 근거로 함

2) 제2정규화

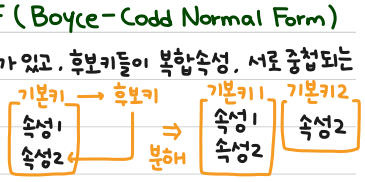
- 기본키가 2개 이상의 속성으로 이루어진 경우에만 발생 (하나인 경우 생략)
- 부분함수 종속성을 제거 → 테이블을 분해

3) 제3정규화

- 제1정규화, 제2정규화를 수행한 다음
- 기본키를 제외한 컬럼간에 종속성 제거
  ↳ 이행함수 종속성

4) BCNF (Boyce-Codd Normal Form)

- 복수의 후보키가 있고, 후보키들이 복합속성, 서로 중첩되는 경우의 작업.



5) 제4정규화

- 여러 컬럼들이 하나의 컬럼을 종속시키는 경우 분해하여 다중값 종속성을 제거

6) 제5정규화

- 조인에 의해서 종속성이 발생되는 경우 분해

[정규화의 문제점]

데이터 조회 (select) 시에 조인 (join) 을 유발

→ CPU와 메모리를 많이 사용

⇒ 문제점 해결 : 인덱스, 옵티마이저

⇒ 성능저하 해결을 위해 반정규화 사용 (반정규화는 데이터를 중복시키기 때문에 또 다른 문제점을 발생)

⇒ 입출력 데이터의 양을 줄여서 성능을 향상시킬 수 있음

[반정규화 (De-Normalization)]

데이터의 중복을 허용하고 조인을 줄이는 데이터베이스 성능향상 방법 ⟶ 조회 (select) 속도를 향상하지만, 데이터유연성은 낮아짐

- 반정규화를 수행하는 경우

정규화에 충실하면 종속성, 활용성은 향상되지만 수행속도가 느려지는 경우

다량의 범위를 자주 처리해야하는 경우

특정 범위의 데이터만 자주 처리하는 경우

요약/ 집계 정보가 자주 요구되는 경우

[반정규화 절차]

1) 대상조사 및 검토 : 데이터처리 범위, 통계성 등을 확인해서 반정규화 대상을 조사

2) 다른 방법 검토 : 클러스터링, 뷰, 인덱스 튜닝, 응용 프로그램, 파티션 등을 검토
↳ 인덱스 정보를 저장시 물리적으로 정렬해서 저장하는 방법

3) 반정규화 수행 : 테이블, 속성, 관계 등을 반정규화

[반정규화 기법]

① 계산된 컬럼 추가 : 배치프로그램으로 미리 계산한 결과를 특정 컬럼에 추가

② 테이블 수직분할 : 하나의 테이블을 두개이상의 테이블로 분할 (컬럼을 분할)

③ 테이블 수평분할 : 하나의 테이블에 있는 값을 기준으로 테이블을 분할

* 파티션 (Partition) 기법 : 논리적으로는 하나의 테이블이지만 여러개의 데이터 파일에 분산되어서 저장

Range Partition : 데이터 값의 범위를 기준

List Partition : 특정한 값을 지정

Hash Partition : 해시 함수를 적용

Composite Partition : 범위와 해시를 복합적으로 사용

장점 데이터 조회시에 엑세스 (Access) 범위가 줄어들어 성능이 향상됨

- 데이터가 분할되어 있어서 I/O (Input Output) 의 성능이 향상됨
- 각 파티션을 독립적으로 백업 및 복구 가능

④ 테이블 병합 : 1대1 관계의 테이블을 하나의 테이블로 병합해서 성능을 향상. 1대N관계의 테이블을 병합하여 성능을 향상 (많은 양의 데이터 중복이 발생)

슈퍼타입과 서브타입관계가 발생하면 테이블을 통합하여 성능을 향상

* Super type (부모) - Sub type (자식)
  고객엔터티 - 개인고객, 법인고객

배타적 관계 : 고객이 개인이거나 법인인 경우

포괄적 관계 : 고객이 개인일수도 법인일수도 있는 경우

* 슈퍼타입 및 서브타입 변환 방법

OneToOne Type : 슈퍼타입과 서브타입을 개별 테이블로 도출. 테이블 수가 많아서 조인 발생. 관리어려움.

Plus Type : 슈퍼타입과 서브타입 테이블로 도출. 조인 발생 관리어려움.

Single Type : 슈퍼타입과 서브타입을 하나의 테이블로 도출. 조인성능 좋고 관리가 편함. 입출력 성능이 나쁨.

[분산 데이터베이스]

물리적으로 떨어진 데이터베이스에 네트워크로 연결

→ 단일 데이터베이스 이미지를 보여줌

→ 분산된 작업처리를 수행 (투명성을 제공)

⟷ 중앙집중형 데이터베이스

(시스템 구축시 한대의 물리적 시스템에 데이터베이스 관리 시스템을 설치)

→ 여러명의 사용자가 접속하여 사용.

[분산 데이터베이스의 투명성 종류]

① 분할투명성 : 고객은 하나의 논리적 릴레이션이 여러 단편으로 분할되어 각 단편의 사본이 여러 시스템에 저장되어있음을 인식할 필요없음.

② 위치투명성 : 고객이 사용하려는 데이터의 저장장소를 명시할 필요없음. 어느 위치에 있더라도 동일한 명령을 사용하여 접근할수 있어야 함.

③ 지역사상투명성 : 지역 DBMS와 물적 데이터베이스 사이의 사상이 보장됨에 따라 각 지역시스템 이름과 무관한 이름 사용가능

④ 중복 투명성 : 데이터베이스 객체가 여러 시스템에 중복되어 존재함에도 고객과는 무관하게 데이터의 일관성이 유지.

⑤ 장애투명성 : 분산되어 있는 각 지역의 시스템이나 통신망에 이상이 발생해도, 데이터의 무결성 보장

⑥ 병행투명성 : 여러 고객의 응용프로그램이 동시에 분산 데이터 베이스에 대한 트랜잭션을 수행하는 경우에도 결과에 이상이 없음.

[분산데이터베이스 설계방식]

① 상향식 설계방식

지역스키마 작성 후 전역스키마를 작성

(지역별로 DB구축한 후 전역으로 통합)

② 하향식 설계방식

전역스키마 작성 후 해당 지역사상스키마 작성

(기업전체 전사 데이터모델수렴 → 전역스키마 생성 → 각 지역별 지역스키마 생성 → DB구축)

- 여러 종류의 데이터베이스 관리 시스템이 있으면 이기종 데이터베이스 관리시스템으로 연동

→ 데이터베이스 미들웨어(ODBC, JDBC)를 사용

* 분산데이터베이스 장점과 단점

장점 데이터베이스의 신뢰성과 가용성이 높음

빠른 응답이 가능 (∵ 병렬처리 수행)

시스템 용량확장이 쉬움

단점 관리와 통제가 어려움 (분리되어있어서)

보안관리 어려움

데이터 무결성 관리가 어려움, 설계가 복잡