

제1장 데이터모델링의 이해

제1절 데이터모델의 이해

1.모델링의 이해1

- 가. 모델링의 정의 : “다양한 현상을 표기법에 의해 표기하는 것”
- 나. ★ 특징 3가지: 추상화, 단순화, 명확화
- 다. 모델링의 3가지 관점: 데이터 관점, 프로세스 관점, 상관 관점

2.데이터모델의 기본개념이해

- 가. 데이터모델링의 정의: “정보시스템을 구축하기 위한 데이터관점의 업무분석기법”
- 나. 데이터모델이 제공하는 기능
 - : 시스템 가시화, 시스템 구조와 행동 명세화, 시스템구축의 구조화된 틀 제공
 - : 문서화, 세부 사항은 숨기는 다양한 관점 제공, 상세 수준의 표현방법 제공

3.데이터모델링의 중요성 및 유의점

- [중요성]
 - 가. 파급효과
 - 나. 복잡한 정보요구사항의 간결한 표현
 - 다. 데이터 품질
- ★
- [유의점]
 - 가. 중복 - 데이터베이스가 여러 장소에 같은 정보를 저장하지 않게 함
 - 나. 비유연성 - 데이터정의를 데이터 사용프로세스와 분리
 - 다. 비일관성 - ex)신용상태에 대한 갱신 없이 고객의 납부이력정보 갱신

4.데이터모델링의 3단계 진행

- 가. 개념적 데이터모델링 : 추상화 수준이 높고 업무 중심적이고 포괄적인 수준의 모델링 진행
- 나. 논리적 데이터모델링 : 시스템으로 구축하고자하는 업무에 대해 Key, 속성, 관계 등을 정확하게 표현, 높은 재사용
- 다. 물리적 데이터모델링 : 실제로 데이터베이스에 이식할 수 있도록 성능, 저장 등 물리적인 성격 고려 설계

5.프로젝트 생명주기에서 데이터모델링

- * 계획 단계-> 개념적모델링 // 분석단계 -> 논리적 모델링 // 설계단계 -> 물리적 모델링
- * 실재는: 분석단계에서 -> 개념적+논리적 모델링 // 설계단계 -> 물리적 모델링

6.데이터 모델링에서 데이터독립성의이해

- 가. 데이터 독립성의 필요성: 유지보수비용 증가, 데이터복잡도 증가, 데이터중복성 증가, 요구사항대응 저하
- 나. 데이터베이스 3단계구조: 외부단계 / 개념적단계 / 내부적단계
- 다. 데이터독립성 요소: 외부스키마 / 개념스키마 / 내부스키마
- 라. 두 영역의 데이터독립성: 논리적 독립성, 물리적 독립성
- 마. ★ 사상 ‘Mapping’ : “상호 독립적인 개념을 연결시켜주는 다리” / 외부적/개념적 사상 + 개념적/물리적 사상

7.데이터모델링의 중요한 세가지 개념

가. 데이터모델링의 세가지 요소

- 1) 업무가 관여하는 어떤 것(Things)
- 2) 어떤 것이 가지는 성격(Attributes)
- 3) 업무가 관여하는 어떤 것 간의 관계(Relationships)

나. 단수와 집합(복수)의 명명

개념	복수/집합개념 & 타입/클래스	개별/단수개념 & 어커런스/인스턴트
어떤것(Thing)	엔터티타입(Entity Type)	엔터티(Entity)
	엔터티(Entity)	인스턴스(Instance) / 어커런스(Occurrence)
어떤것간의 연관(Relationships)	관계(Relationship)	페어링(Pairing)
어떤것의 성격(Attributes)	속성(Attribute)	속성값(Attribute Value)

8.데이터모델링의 이해관계자

가. 이해관계자의 데이터 모델링 중요성 인식

나. 데이터 모델링의 이해관계자

9.데이터모델의 표기법인 ERD의 이해

가.데이터 모델 표기법: 엔터티를 사격형으로 표현, 관계를 마름모, 속성을 타원형으로 표현

나. ERD(Entity Relationship Diagram)표기법을 이용하여 모델링하는 방법

엔터티그리기->엔터티 배치->엔터티간의 관계설정->관계명 기술->관계의참여도 기술->관계의 필수여부기술

10. ★ 좋은 데이터모델의 요소

완전성 / 중복배제 / 업무규칙 / 데이터 재사용 / 의사소통 / 통합성

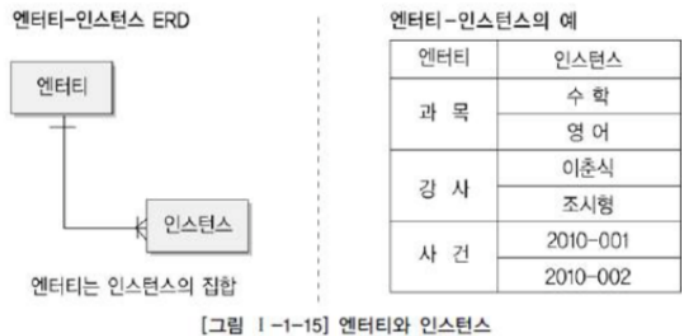
제2절 엔터티

1. 엔터티의 개념: “실체,객체”

2. 엔터티와 인스턴스에 대한 내용과 표기법 ——>

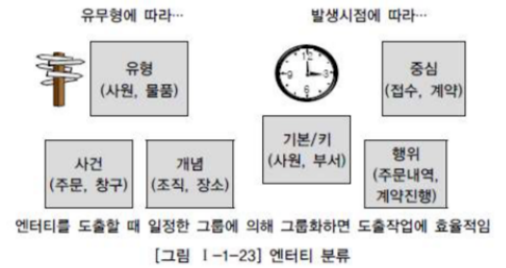
3. 엔터티의 특징

- 가. 업무에서 필요로 하는 정보
- 나. 식별자에 의해 식별이 가능 해야함
- 다. 인스턴스의 집합
- 라. 업무프로세스에 의해 이용
- 마. 속성을 포함
- 바. 관계의 존재



4. ★ 엔터티의 분류

- 가. 유무형에 따른 분류: 유형 엔터티, 개념 엔터티, 사건 엔터티
- 나. 발생시점에 따른 분류: 기본 엔터티, 중심 엔터티, 행위 엔터티
- 다. 엔터티 분류방법의 예 ----->



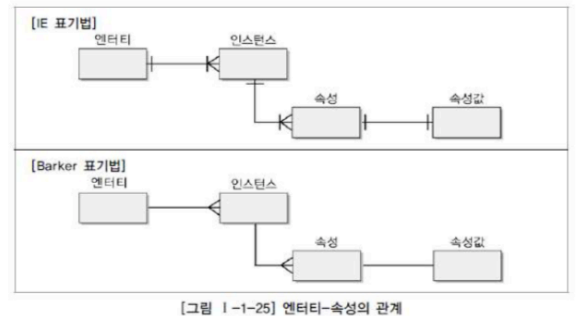
5. 엔터티의 명명: 업무목적에 따라 생성되는 자연스러운 이름을 부여

제3절 속성

1. 속성의 개념: “업무에서 필요로 하는 인스턴스로 관리하고자하는 의미상 더 이상 분리되지 않는 최소의 데이터 단위”

2. 엔터티, 인스턴스와 속성, 속성값에 대한 내용과 표기법

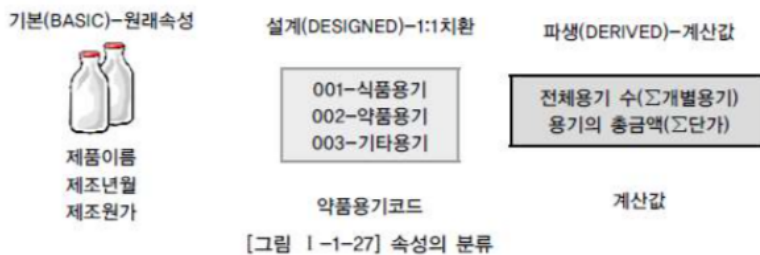
- 가. ★ 엔터티, 인스턴스, 속성, 속성값의관계
 - 한 개의 엔터티는 두개 이상의 인스턴스 집합이어야 한다.
 - 한 개의 엔터티는 두개 이상의 속성을 갖는다.
 - 한 개의 속성은 한 개의 속성값을 갖는다.
- 나. 속성의 표기법: IE표기법, Barker 표기법



3. 속성의 특징: 하나의 속성에는 한 개의 값. 하나의 속성에 여러 개의 값이 있는 다중값일 경우 별도의 엔터티를 이용하여 분리

4. 속성의 분류

- 가. ★ 속성의 특성에 따른 분류: 기본속성, 설계속성, 파생속성



나. 엔터티 구성방식에 따른 분류

- 엔터티를 식별할 수 있는 속성을 PK속성,
- 다른 엔터티와의 관계에서 포함된 속성을 FK속성,
- 엔터티에 포함되어 있고 PK,FK에 포함되지 않은 속성을 일반 속성이라 한다.
- 또한 의미를 쪼갤 수 있는지에 따라 단순형, 복합형으로 분류할 수 있다.
- 속성 하나에 한 개의 값 -> 단일값(Single Value) / 여러 개의 값 -> 다중값(Multi Value)

5. ★ 도메인: “각 속성이 가질 수 있는 값의 범위”

- > 엔터티 내에서 속성에 대한 데이터타입과 크기 그리고 제약사항을 지정하는 것

제4절 관계

1. 관계의 개념

가. 관계의 정의

“엔터티의 인스턴스 사이의 논리적인 연관성으로서 존재의 형태로거나 행위로서 서로에게 연관성이 부여된 상태”

나. 관계의 페어링: “페어링은 엔터티안에 인스턴스가 개별적으로 관계를 가지는 것이고, 이것의 집합을 관계로 표현”

다. 관계의 분류

- ERD: 존재에 의한 관계 / 행위에 의한 관계
- UML(Unified Modeling Language): 연관 관계 / 의존관계

2. 관계의 표기법

가. 관계명: 관계의 이름

나. 관계 차수: 1:1, 1:M, M:N

다. 관계선택사항: 필수관계, 선택관계

4. 관계의 정의 및 읽는 방법

가. 관계체크사항

나. 관계읽기



[그림 1-1-39] 관계의 읽는 방법

제5절 식별자

1. 식별자 개념

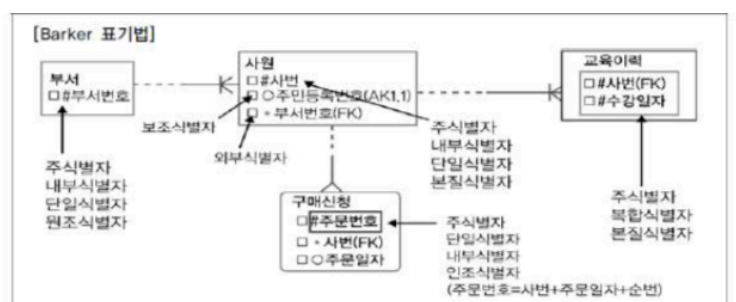
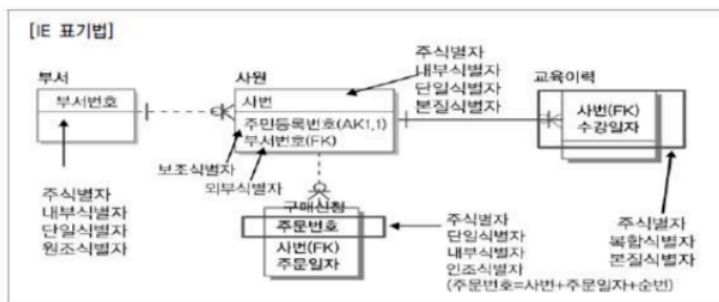
- 하나의 엔터티에 구성되어있는 여러 개의 속성 중에 엔터티를 대표할 수 있는 속성을 의미하며
- 하나의 엔터티는 반드시 하나의 유일한 식별자가 존재해야 한다.

2. ★ 식별자의 특징: 유일성, 최소성, 불변성, 존재성

3. ★ 식별자분류 및 표기법

가. 식별자 분류: 주식별자 보조식별자 / 내부식별자와 외부식별자 / 단일식별자와 복합식별자, 본질식별자와 인조식별자

나. 식별자 표기법

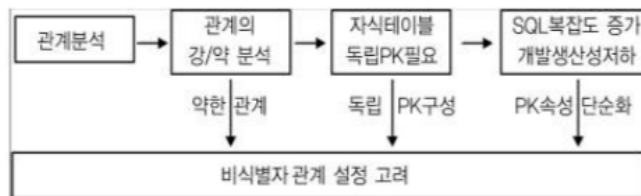


4. 주식별자 도출기준

- 가. 해당업무에서 자주 이용되는 속성을 주식별자로 지정하도록함
- 나. 명칭, 내역등과같이 이름으로 기술되는 것을 피함
- 다. 속성의 수가 많아지지 않도록함

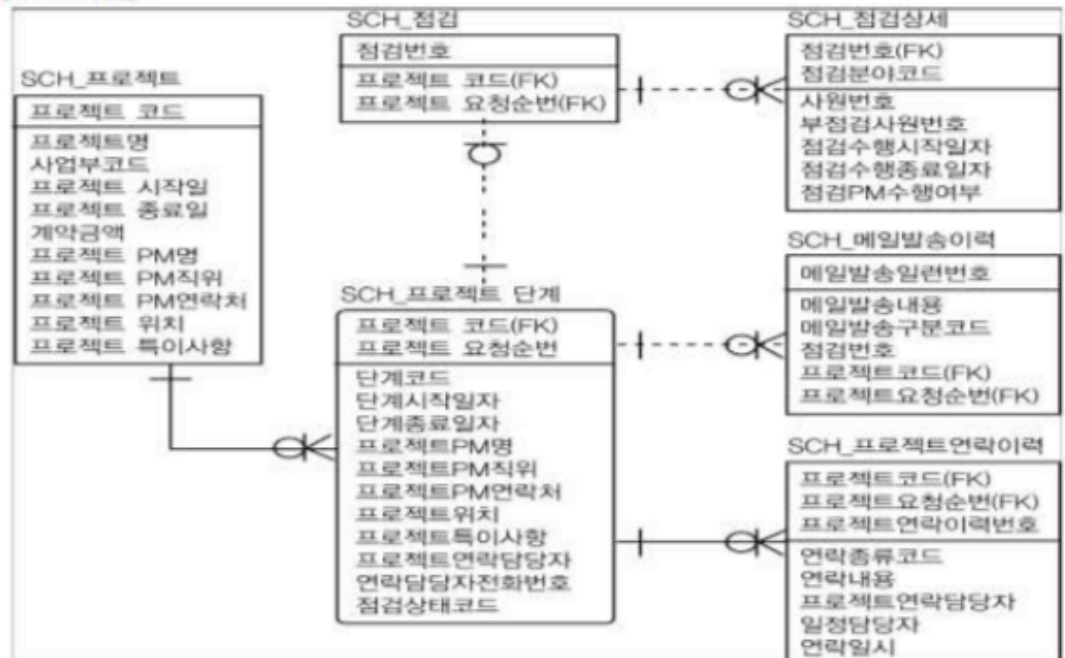
5. 식별자관계와 비식별자 관계에 따른 식별자

- 가. 식별자관계와 비식별자 관계의 결정: 외부식별자는 Foreign Key역할을 한다.
- 나. 식별자 관계: “자식엔터티의 주식별자로 부모의 주식별자가 상속이 되는 경우”
Null값이 오면 안되므로 반드시 부모엔터티가 생성되어야 자기자신의 엔터티생성
- 다. 비식별자 관계: “부모 엔터티로부터 속성으로 받았지만 자식엔터티의 주식별자로 사용하지않고 일반적인 속성으로만 사용하는 경우”
- 라. 식별자관계로만 설정할 경우의 문제점: 주식별자 속성이 지속적으로 증가, 복잡성과 오류 가능성 유발
- 마. 비식별자관계로만 설정한 경우의 문제점: 쓸데없이 부모엔터티까지 찾아가야 하는 경우가 발생한다
- 바. 식별자관계와 비식별자관계 모델링: 비식별자관계 선택프로세스, 식별자와 비식별자 관계비교,
식별자와 비식별자를 적용한 데이터 모델



[그림 1-1-53] 비식별자관계 설정 고려사항

[IE 표기법]



제2장 데이터 모델과 성능

제1절 성능데이터모델링의 개요

1. 성능데이터모델링의 정의

데이터모델링을 할 때 어떤 작업유형에따라 성능향상을 도모해야하는지
목표를 분명하게 해야 정확한 성능향상 모델링을 할 수 있다.

2. 성능 데이터모델링 수행 시점

분석/설계단계에서 데이터베이스 처리 성능을 향상시킬 수 있는 방법을 고려해야 한다

3. 성능 데이터모델링 고려사항

정규화..반정규화...

제2절 정규화와 성능

1. 정규화를 통한 성능향상전략

- ★“데이터에 대한 중복성을 제거하여준다”
- 일반적으로 정규화가 잘되어있으면 입력/수정/삭제의 성능이 향상되고 반정규화를 많이 하면 조회의 성능이 향상된다
- 중복속성에 대한 분리가 1차 정규화의 정의임. 로우단위의 대상, 칼럼 단위로 중복도 해당

2. 반정규화된 테이블의 성능저하사례1

3. 반정규화된 테이블의 성능저하사례2

4. 반정규화된 테이블의 성능저하사례3

5. 반정규화된 테이블의 성능저하사례4

6. ★ 함수적 종속성에 근거한 정규화 수행 필요

- 함수의 종속성(Functional Dependency)은 데이터들이 어떤 기준값에 의해 종속되는 현상을 지칭 하는 것이다.
- 결정자 예) 주민등록번호 // 종속자 예) 이름, 출생지,주소

제3절 반정규화와 성능

1. 반정규화를 통한 성능향상 전략

가. 반정규화의 정의

“정규화된 엔터티, 속성, 관계에 대해 시스템의 성능향상과 개발과 운영의
단순화를 위해 중복, 통합, 분리 등을 수행하는 데이터모델링 기법”
조인으로 인한 성능저하가 예상될 때 등

나. 반정규화의 적용방법: 반정규화대상 조사, 다른 방법 유도 검토, 반정규화 적용

2. 반정규화 기법

- 가. 테이블 반정규화: 테이블병합, 테이블분할, 테이블추가
- 나. 칼럼 반정규화: 중복칼럼 추가, 파생칼럼 추가, 이력테이블칼럼 추가, PK에 의한 칼럼 추가, 응용시스템 오작동을 위한 칼럼 추가
- 다. 관계 반정규화: 중복관계추가

3. 정규화가 잘 정의된 데이터모델에서 성능이 저하될 수 있는 경우

4. 정규화가 잘 정의된 데이터모델에서 성능이 저하된 경우

제4절 대량데이터에 따른 성능

1. 대량데이터 발생에 따른 테이블 분할 개요

- 로우 체이닝(Row Chaining)-“로우길이가 너무 길어서 데이터블록하나에 데이터가 모두 저장되지 않고 두개 이상 저장”
- 로우 마이그레이션(Row Migration)-“데이터블록에서 수정이 발생하면 수정된 데이터를 해당 데이터블록에서 저장하지 못하고 다른 블록의 빈 공간을 찾아 저장하는 방식”

2. 한 테이블에 많은 수의 칼럼을 가지고 있는 경우

3. 대량 데이터 저장 및 처리로 인한 성능

- 가. RANGE PARTITION 적용
- 나. LIST PARTITION 적용
- 다. HASH PARTITION 적용

4. 테이블에 대한 수평 분할/수직 분할의 절차

제5절 데이터베이스 구조와 성능

1. 슈퍼타입 / 서브타입 모델의 성능고려방법

가. 슈퍼/서브타입 데이터모델의 개요

“Extended ER모델”, 업무를 구성하는 데이터의 특징을 공통과 차이점의 특징을 고려하여 효과적으로 표현할 수 있기 때문에 자주 쓰임

(공통은 슈퍼타입으로 모델링하고, 공통으로부터 상속받아 다른 엔티티와 차이가 있는 속성은 별도의 서브엔티티 구분

나. 슈퍼/서브타입 데이터 모의 변환

다. 슈퍼/서브타입데이터모델의 변환 기술

라. 슈퍼/서브타입데이터모델의 변환타입비교

OneToOne Type / Plus Type / Single Type

2. ★ 인덱스 특성을 고려한 PK/FK 데이터베이스 성능 향상

- 일반적으로 프로젝트에서는 PK/FK 칼럼 순서의 중요성을 인지하지 못한 채로 데이터 모델링이 되어 있는 그 상태로 바로 DDL 을 생성함으로써 데이터베이스 데이터처리 성능에 문제를 유발하는 경우가 빈번하게 발생이 된다.

가. ★ PK/FK 칼럼 순서와 성능 개요

- 인덱스의 특징은 여러 개의 속성이 하나의 인덱스로 구성되어 있을 때 앞쪽에 위치한 속성의 값이 비교자로 있어야 인덱스가 좋은 효율을 나타낼 수 있다. 앞쪽에 위치한 속성값이 가급적 ‘=’아니면 최소한 범위 BETWEEN이 들어와야 됨

나. PK칼럼의 순서를 조정하지 않으면 성능이 저하 이유

- PK 의 순서를 인덱스 특징에 맞게 고려하지 않고 바로 그대로 생성하게 되면, 테이블에 접근하는 트랜잭션의 특징에 효율적이지 않은 인덱스가 생성되어 있으므로 인덱스의 범위를 넓게 이 용하거나 Full Scan 을 유발하게 되어 성능이 저하된다고 정리할 수 있다.

다. PK순서를 잘못 지정하여 성능이 저하된 경우-간단한 오류

라. PK순서를 잘못 지정하여 성능이 저하된 경우-복잡한 오류

3. 물리적인 테이블에 FK제약이 걸려있지 않을 경우 인덱스 미생성으로 성능저하

- 물리적인 테이블에 FK 제약 걸었을 때는 반드시 FK 인덱스를 생성하도록 하고 FK 제약이 걸리지 않았을 경우에는 FK 인덱스를 생성하는 것을 기본정책으로 하되 발생하는 트랜잭션에 의해 거의 활용되지 않았을 때에만 FK 인덱스를 지우는 방법으로 하는 것이 적절한 방법

제6절 분산데이터베이스와 성능 →

데이터베이스 분산 설계는 다음과 같은 경우에 적용하면 효과적이다.

- 성능이 중요한 사이트에 적용해야 한다.
- 공통코드, 기준정보, 마스터 데이터 등에 대해 분산환경을 구성하면 성능이 좋아진다.
- 실시간 동기화가 요구되지 않을 때 좋다. 거의 실시간(Near Real Time)의 업무적인 특징을 가지고 있을 때도 분산 환경을 구성할 수 있다
- 특정 서버에 부하가 집중이 될 때 부하를 분산할 때도 좋다.
- 백업 사이트(Disaster Recovery Site)를 구성할 때 간단하게 분산기능을 적용하여 구성할 수 있다.

1. 분산데이터베이스의 개요: “데이터베이스를 연결하는 빠른 네트워크 환경을 이용하여 데이터베이스를 여러 지역 여러 노드로 위치시켜 사용성/성능 등을 극대화시킨 데이터베이스”

2. 분산데이터베이스의 투명성: 분할, 위치, 지역 사상, 중복, 장애, 병행

3. 분산데이터베이스의 적용방법 및 장단점

- 가. 분산데이터베이스 적용방법
- 나. 분산데이터베이스 장단점

4.분산데이터베이스의 활용 방향성

5. 데이터베이스 분산구성의 가치

6. 분산데이터베이스의 적용기법

- 가. 테이블위치분산
- 나. 테이블분할분산 - 수평분할,수직분할
- 다. 테이블복제분산 - 부분복제,광역복제
- 라. 테이블요약분산 - 분석요약,통합요약

7. 분산데이터베이스를 적용하여 성능이 향상된 사례 ——>

[그림 1-2-55] 업무 특성에 따른 분산환경 구성

