**데이터 모델링의 이해**

**요약 정리**

**1. 모델링의 이해**

- 모델 : 일정한 표기법에 의해 표현해 놓은 모형

- 모델링 : 모델을 만들어 가는 일. 현실세계를 추상화, 단순화, 명확화 하기 위해 일정한 표기법에 의해 표기하는 기법

- 모델링의 특징

(1) 추상화 : 현실세계를 일정한 형식에 맞추어 표현

(2) 단순화 : 복잡한 현실세계를 일정한 규약에 의해 제한된 표기법이나 언어로 쉽게 표현

(3) 명확화 : 누구나 이해하기 쉽게 대상에 대한 모호함을 제거하고 정확하게 기술

- 모델링의 관점

(1) 데이터 관점 : 업무가 어떤 데이터와 관련이 있고 데이터 간의 관계는 무엇인지에 대해 모델

링(Data, What)

(2) 프로세스 관점 : 업무가 실제하고 있는 일은 무엇인지 또는 무엇을 해야 하는 지 모델링

(Process, How)

(3)데이터와 프로세스의 상관 관점 : 업무가 처리하는 일의 방법에 따라 데이터는 어떻게 영향을

받는 지 모델링(Interaction)

**2. 데이터 모델**

- 데이터 모델 : 정보시스템을 구축하기 위해 해당 업무에 어떤 데이터가 존재하는지 또는 업무가 필요로 하는 정보는 무엇인지 분석하는 방법.

- 데이터 모델링 : 정보시스템을 구축하기 위한 데이터 관점의 업무 분석 기법.

현실세계와 데이터에 대해 약속된 표기법으로 표현하는 과정

데이터베이스를 구축하기 위한 분석과 설계의 과정

- 데이터 모델의 기능

(1) 시스템을 현재 또는 원하는 모습으로 가시화

(2) 시스템의 구조와 행동을 명세화

(3) 시스템을 구축하는 구조화된 틀 제공

(4) 시스템을 구축하는 과정에서 결정한 것을 문서화

(5) 다양한 영역에 집중하기 위해 다른 영역의 세부 사항을 숨기는 다양한 관점 제공

(6) 특정 목표에 따라 구체화된 상세 수준의 표현방법 제공

**3. 데이터 모델링의 중요성/유의점**

- 중요성

(1) 파급효과(Leverage) : 데이터 모델이 변경되면 프로젝트에 큰 차질이 생기기 때문

(2) 간결한 표현(Conciseness) : 데이터 모델은 구축할 시스템 정보 요구사항과 한계를 가장 명확

하고 간결하게 표현할 수 있는 도구다. 요구사항 파악하기 가장 좋은 방법은 각각 페이지별 기능적인 요구를 파악하기 보다 데이터 모델을 리뷰하는 것이다.

(3) 데이터 품질(Data Quality) : 데이터 품질이 낮으면 치뤄야 할 비용이 커짐

- 유의점

(1) 중복(Duplication)

(2) 비유연성(Inflexibility) : 데이터의 정의를 데이터의 사용 프로세스와 분리

(3) 비일관성(Inconsistency)

**4. 데이터 모델링 3단계**

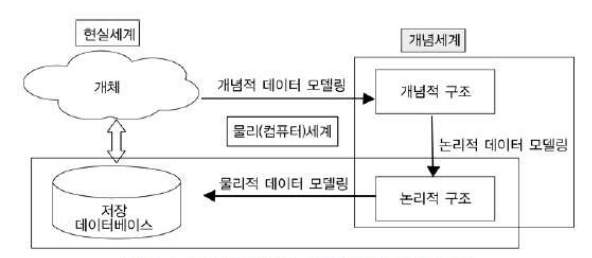
(1) 개념적 데이터 모델링 : 추상화 수준이 높고 업무중심적. 포괄적인 수준의 모델링 진행.

엔터티-관계 다이어그램 생성. 사용자와 시스템 개발자가 데이터 요구 사항 발견하게 지원함.

(2) 논리적 데이터 모델링 : 시스템으로 구축하고자 하는 업무에 key, 속성, 관계 등을 정확히 표현.

재사용성이 높음. 비즈니스 데이터에 존재하는 사실들을 기록. 정규화 진행.

(3) 물리적 데이터 모델링 : DB에 이식 가능하게 성능, 저장 등 물리적인 성격을 고려하여 설계. 구체적.



**5. 데이터 독립성**

- 필요성

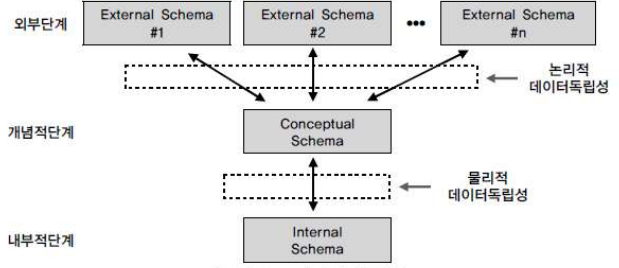
(1) 유지보수 비용을 낮추기 위해

(2) 데이터 복잡도를 낮추기 위해

(3) 데이터 중복성을 낮추기 위해

(4) 요구사항에 대해 화면과 데이터베이스간 독립성을 유지하기 위해

- 데이터 독립성과 DB 구조



- 데이터베이스 3단계 구조

(1) 외부 단계

(2) 개념적 단계

(3) 내부적 단계

- 데이터 독립성 요소

(1) 외부스키마(External Schema) : 각각 사용자가 보는 개인적인 DB 스키마. 사용자 관점이며

접근하는 특성에 따른 스키마가 구성됨

(2) 개념스키마(Conceptual Schema) : 모든 사용자 관점을 통합한 조직전체의 DB 기술이며 통합

관점. DB에 저장되는 데이터와 관계를 표현

(3) 내부스키마(Internal Schema) : DB가 물리적으로 저장되는 형식. 데이터가 실제로 저장되는

방법 표현. 물리적 저장구조

- 영역별 데이터 독립성

(1) 논리적 데이터 독립성 : 개념 스키마가 변경되어도 외부 스키마에 영향을 미치지 않음.

논리적 구조가 변경되어도 응용 프로그램에 영향 없음.

통합 구조 변경 가능. 사용자 특성에 맞는 변경 O

(2) 물리적 데이터 독립성 : 내부 스키마가 변경되어도 외부/개념 스키마는 영향 X

저장장치의 구조 변경은 응용 프로그램과 개념스키마에 영향 X

물리적 구조 영향 없이 개념구조 변경 O

개념 구조 영향 없이 물리적 구조 변경 O

- 매핑(mapping, 사상)

독립적인 개념을 연결시켜주는 다리

(1) 외부적/개념적 매핑(논리적 매핑) : 외부 뷰와 개념적 뷰의 상호 관련성 정의

(2) 개념적/내부적 매핑(물리적 매핑) : 개념적 뷰와 DB의 상호관련성 정의

**6. DB 모델링의 3가지 요소**

(1) Things(어떤 것) : 엔터티

(2) Attributes(속성)

(3) Relationships(관계)

**7. ERD**

- 정의 : 데이터 모델에 대한 표기법. Entity Relationships Diagram

- 작업 순서 : 엔터티를 그림(사각형으로) – 엔터티 배치(좌측상단~우측하단) – 엔터티간 관계 설정 – 관계명 기술 – 관계 참여도 기술(관계차수 Cardinality) – 관계의 필수 여부 기술

**8. 좋은 데이터 모델의 요소**

(1) 완전성(Completeness)

(2) 중복배제(Non-Redundancy)

(3) 업무규칙(Business Rules)

(4) 데이터 재사용(Data Reusability)

(5) 의사소통(Communication)

(6) 통합성(Integration)

**9. 엔터티(Entity)**

- 정의 : 사람, 물건, 개념 등의 명사. 업무상 관리가 필요한 관심사. 저장이 되기 위한 어떤 것(Thing)

업무에 필요하고 유용한 정보를 저장하고 관리하기 위한 집합적인 것(=table)

- 인스턴스 : 테이블의 행(row). 인스턴스의 집합 = 엔터티. 즉 엔터티 하나의 값. 저장된 데이터 내용의 전체 집합

- 특징

(1) 반드시 업무에 필요하고 관리하고자 하는 정보

(2) 유일한 식별자에 의해 식별이 가능

(3) 영속적으로 존재하는 인스턴스의 집합

(4) 업무 프로세스에 의해 사용됨

(5) 속성이 있음 : 예외적으로 Associative Entity는 주식별자 속성만 있어도 괜찮음

(6) 다른 엔터티와 최소 한 개 이상의 관계가 있음

- 분류

유뮤(有無)형에 따른 분류

(1) 유형엔터티(Tangible Entity) : 물리적인 형태 있음

(2) 개념엔터티(Conceptual Entity) : 관리해야 할 개념적 정보로 구분됨

(3) 사건엔터티(Event Entity) : 업무를 수행함에 따라 발생되는 엔터티. 통계자료에 사용

발생시점에 따른 분류

(1) 기본엔터티 : 업무에 원래 존재하던 정보. 타 엔터티와의 관계에서 생성 X 독립적으로 생성

되며 다른 엔터티의 부모 역할을 함

(2) 중심엔터티 : 기본엔터티로부터 발생. 업무에 있어서 중심적인 역할을 함. 행위엔터티 생성.

(3) 행위엔터티 : 두 개 이상의 부모 엔터티로부터 발생. 자주 내용과 데이터량이 변경.

- 명명(命名)

(1) 업무에서 사용하는 용어 사용

(2) 약어 사용을 피한다

(3) 단수명사 사용

(4) 유일한 이름 부여

(5) 엔터티 생성 의미대로 이름 부여

**10. 속성(Attribute)**

- 정의 : 인스턴스로 관리하고자 하는 의미상 분리되지 않는 최소의 데이터 단위. 엔터티를 설명하고 인스턴스의 구성요소.

- 특징

(1) 해당 업무에서 필요함

(2) 정해진 주식별자에 함수적 종속성을 갖음

(3) 하나의 속성에는 단 하나의 속성 값만 갖음. 하나의 속성에 여러 개의 값이 있으면 엔터티를

이용하여 분리한다

- 분류

특성에 따라

(1) 기본속성 : 업무로부터 추출한 모든 속성.

(2) 설계속성 : 업무상 필요한 데이터 이외에 모델링, 업무 규칙화를 위해 새로 만들거나 변형된

속성. 일련번호 같은 단일한 식별자를 부여하기 위해 모델상에서 새로 정의하는 속성

(3) 파생속성 : 다른 속성에 영향을 받아 발생하는 속성. 일반적으로는 계산된 값. 데이터 정합성

을 유지하기 위해 가급적 적게 정의하는 것이 좋음

엔터티 구성 방식에 따라

(1) PK(Primary Key) : 엔터티를 식별하는 속성

(2) FK(Foreign Key) : 다른 엔터티와 관계에서 포함된 속성

(3) 일반속성

세부 의미를 쪼갤 수 있는지

(1) 복합속성(Composite Attribute)

(2) 단순속성(Simple Attribute)

- 도메인 : 각 속성이 가질 수 있는 값의 범위. 엔터티 내 데이터타입, 크기, 제약 사항 지정

- 명명

(1) 해당업무에서 사용하는 이름 부여

(2) 서술식으로 된 이름 X

(3) 약어 X

(4) 전체 데이터모델에서 유일성 확보 필요

**11. 관계**

- 정의 : 엔터티의 인스턴스 사이의 논리적인 연관성. 존재의 형태나 행위로서 서로에게 연관성이 부여됨

- 페어링 : 각각 엔터티의 인스턴스들은 자신과 관련된 인스턴스들과 관계를 맺는 것

- 분류

UML(Unified Modeling Language) : 연관관계(Association)와 의존관계(Dependency)가 있다. 실선과 점선 표기법으로 다르게 표현됨. 연관관계는 이용하는 관계, 의존관계는 상대방 행위에 의해 관계 형성

반면 ERD는 존재적 관계와 행위에 의한 관계를 구분하지 않음

- 표기법

(1) 관계명(Membership)

(2) 관계차수(Cardinality) : 1:1, 1:M, N:M 처럼 참여자의 수 표현

(3) 관계선택사양(Optionality) : 필수 관계, 선택 관계

- 관계 정의 필수 요소

(1) 두 개의 엔터티 사이에 관심있는 연관규칙이 존재해야 함

(2) 두 개의 엔터티 사이에 정보의 조합이 발생함

(3) 관계 연결에 대한 규칙이 서술됨

(4) 관계 연결을 가능하게 하는 동사가 포함됨

**12. 식별자**

- 정의 및 키(key)와 비교 : 하나의 엔터티에 구성되어 있는 속성 중 엔터티를 대표할 수 있는 속성.

식별자는 논리 데이터 모델링에서 사용되며 키(key)는 DB 테이블 접근을

위한 매개체로 물리적 데이터 모델링 단계에서 사용됨

- 특징

(1) 유일성 : 주식별자에 의해 엔터티 내 모든 인스턴스들이 유일하게 구분됨

(2) 최소성 : 주식별자를 구성하는 속성의 수는 유일성을 만족하는 최소의 수

(3) 불변성 : 주식별자의 값은 자주 변하지 않음

(4) 존재성 : 주식별자가 지정되면 반드시 값이 들어와야 함(NULL X)

- 분류

(1) 대표성 여부 – 주식별자/보조식별자

(2) 스스로 생성 여부 – 내부식별자/외부식별자

(3) 속성의 수 – 단일식별자/복합식별자

(4) 대체 여부 – 본질식별자/인조식별자

- 주식별자 도출기준

(1) 해당 업무에서 자주 이용되는 속성

(2) 명칭, 내역 등과 같이 이름으로 기술되는 것은 가급적 피한다

(3) 복합으로 주식별자로 구성할 경우 너무 많은 속성이 포함되지 않도록 한다

- 식별자 관계

부모로부터 받은 FK를 자식엔터티의 주식별자로 사용하고 그것만으로 주식별자로 사용하면 부모 자식간의 관계는 1:1이 된다. 또 다른 부모에게 받은 속성이나 스스로 가진 속성을 주식별자로 사용하면 1:M이다.

- 비식별자 관계(Non-Identifying Relationship)

부모로부터 속성을 받았지만 자식엔터티의 주식별자로 사용하지 않는 경우를 말한다.

(1) 자식엔터티가 받은 속성이 반드시 필수가 아니라 부모 없는 자식이 생성될 경우

(2) 엔터티별로 데이터의 생명주기를 다르게 관리할 경우

(3) 여러 개의 엔터티가 하나의 엔터티로 통합되어 표현되었는데 각각의 엔터티가 별도의 관계를

가질 때

(4) 자식쪽 엔터티의 주식별자를 부모엔터티와는 별도로 생성하는 것이 더 유리할 때

- 비식별자 관계로만 설정할 경우 문제점

자식엔터티에서 데이터를 처리할 때 부모엔터티까지 찾아가는 경우, SQL 구문에서 많은 조인이 필요하여 성능 저하을 유발할 수 있다.

- 비식별자 선택 고려 사항 프로세스

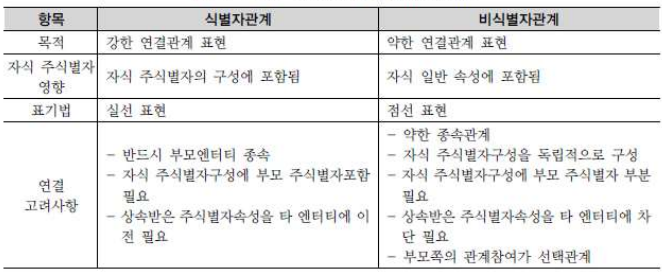
(1) 관계 분석

(2) 관계의 강/약 분석 – 약한 관계면 비식별자 설정 고려

(3) 자식테이블 독립 PK의 필요성 여부 – 부모와 독립된 PK가 필요하면 비식별자 설정 고려

(4) SQL - 복잡도 증가 및 개발생산성 저하 – PK 속성 단순화를 통해 비식별자 설정 고려

- 식별자와 비식별자 관계 비교



13. 추가 정리 – 엔터티, 인스턴스, 속성, 속성값

엔터티에는 두 개 이상의 인스턴스가 있다

엔터티에는 고유의 성격을 표현하는 속성이 두 개 이상 있다

속성은 한 개의 속성값만 갖는다(Single-Valued Attribute)

**데이터 모델링의 이해**

**빈칸채우기**

**1. 모델링의 이해**

- ( ) : 일정한 표기법에 의해 표현해 놓은 모형

- ( ) : 모델을 만들어 가는 일. 현실세계를 추상화, 단순화, 명확화 하기 위해 일정한 표기법에 의해 표기하는 기법

- 모델링의 특징

(1) ( ) : 현실세계를 일정한 형식에 맞추어 표현

(2) ( ) : 복잡한 현실세계를 일정한 규약에 의해 제한된 표기법이나 언어로 쉽게 표현

(3) ( ) : 누구나 이해하기 쉽게 대상에 대한 모호함을 제거하고 정확하게 기술

- 모델링의 관점

(1) ( ) : 업무가 어떤 데이터와 관련이 있고 데이터 간의 관계는 무엇인지에 대해 모델

링(Data, What)

(2) ( ) : 업무가 실제하고 있는 일은 무엇인지 또는 무엇을 해야 하는 지 모델링

(Process, How)

(3)데이터와 프로세스의 상관 관점 : 업무가 처리하는 일의 방법에 따라 데이터는 어떻게 영향을

받는 지 모델링(Interaction)

**2. 데이터 모델**

- ( ) : 정보시스템을 구축하기 위해 해당 업무에 어떤 데이터가 존재하는지 또는 업무가 필요로 하는 정보는 무엇인지 분석하는 방법.

- 데이터 모델링 : 정보시스템을 구축하기 위한 ( ) 관점의 업무 분석 기법.

현실세계와 데이터에 대해 ( )으로 표현하는 과정

( )를 구축하기 위한 분석과 설계의 과정

- 데이터 모델의 기능

(1) 시스템을 현재 또는 원하는 모습으로 ( )

(2) 시스템의 구조와 행동을 ( )

(3) 시스템을 구축하는 구조화된 ( ) 제공

(4) 시스템을 구축하는 과정에서 결정한 것을 ( )

(5) 다양한 영역에 집중하기 위해 다른 영역의 세부 사항을 숨기는 다양한 관점 제공

(6) 특정 목표에 따라 구체화된 상세 수준의 표현방법 제공

**3. 데이터 모델링의 중요성/유의점**

- 중요성

(1) ( ) : 데이터 모델이 변경되면 프로젝트에 큰 차질이 생기기 때문

(2) ( ) : 데이터 모델은 구축할 시스템 정보 요구사항과 한계를 가장 명확

하고 간결하게 표현할 수 있는 도구다. 요구사항 파악하기 가장 좋은 방법은 각각 페이지별 기능적인 요구를 파악하기 보다 데이터 모델을 리뷰하는 것

(3) ( ) : 데이터 품질이 낮으면 치뤄야 할 비용이 커짐

- 유의점

(1) ( )

(2) ( ) : 데이터의 정의를 데이터의 사용 프로세스와 분리

(3) ( )

**4. 데이터 모델링 3단계**

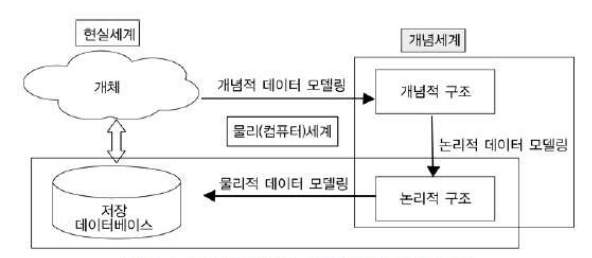
(1) 개념적 데이터 모델링 : ( ) 수준이 높고 업무중심적. 포괄적인 수준의 모델링 진행.

엔터티-관계 다이어그램 생성. 사용자와 시스템 개발자가 데이터 요구 사항 발견하게 지원함.

(2) 논리적 데이터 모델링 : 시스템으로 구축하고자 하는 업무에 key, 속성, 관계 등을 정확히 표현.

( )이 높음. 비즈니스 데이터에 존재하는 사실들을 기록. ( ) 진행.

(3) 물리적 데이터 모델링 : DB에 이식 가능하게 성능, 저장 등 물리적인 성격을 고려하여 설계. 구체적.



**5. 데이터 독립성**

- 필요성

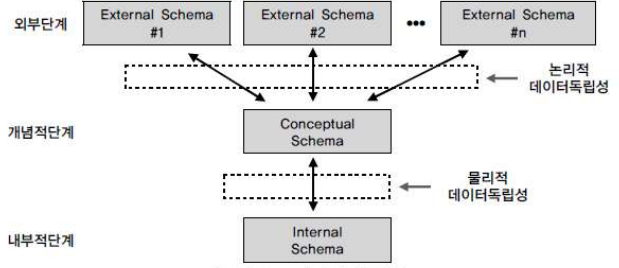
(1) ( )을 낮추기 위해

(2) ( )를 낮추기 위해

(3) ( )을 낮추기 위해

(4) 요구사항에 대해 화면과 데이터베이스간 ( )을 유지하기 위해

- 데이터 독립성과 DB 구조



- 데이터베이스 3단계 구조

(1) ( )

(2) ( )

(3) ( )

- 데이터 독립성 요소

(1) ( ) : 각각 사용자가 보는 개인적인 DB 스키마. 사용자 관점이며 접근하는 특성에 따른 스키마가 구성됨

(2) ( ) : 모든 사용자 관점을 통합한 조직전체의 DB 기술이며 통합 관점. DB에 저장되는 데이터와 관계를 표현

(3) ( ) : DB가 물리적으로 저장되는 형식. 데이터가 실제로 저장되는 방법 표현. 물리적 저장구조

- 영역별 데이터 독립성

(1) ( ) 데이터 독립성 : 개념 스키마가 변경되어도 외부 스키마에 영향을 미치지 않음.

논리적 구조가 변경되어도 응용 프로그램에 영향 없음.

통합 구조 변경 가능. 사용자 특성에 맞는 변경 O

(2) ( ) 데이터 독립성 : 내부 스키마가 변경되어도 외부/개념 스키마는 영향 X

저장장치의 구조 변경은 응용 프로그램과 개념스키마에 영향 X

물리적 구조 영향 없이 개념구조 변경 O

개념 구조 영향 없이 물리적 구조 변경 O

- 매핑(mapping, 사상)

독립적인 개념을 연결시켜주는 다리

(1) 외부적/개념적 매핑( 매핑) : 외부 뷰와 개념적 뷰의 상호 관련성 정의

(2) 개념적/내부적 매핑( 매핑) : 개념적 뷰와 DB의 상호관련성 정의

**6. DB 모델링의 3가지 요소**

(1) ( )

(2) ( )

(3) ( )

**7. ERD**

- 정의 : 데이터 모델에 대한 표기법. Entity Relationships Diagram

- 작업 순서 : ( )를 그림(사각형으로) – 엔터티 배치(좌측상단~우측하단) – 엔터티간 ( )– 관계명 기술 – 관계 참여도 기술(관계차수 Cardinality) – 관계의 필수 여부 기술

**8. 좋은 데이터 모델의 요소**

(1) ( )

(2) ( )

(3) ( )

(4) ( )

(5) 의사소통(Communication)

(6) 통합성(Integration)

**9. 엔터티(Entity)**

- 정의 : 사람, 물건, 개념 등의 명사. 업무상 관리가 필요한 관심사. 저장이 되기 위한 어떤 것(Thing)

업무에 필요하고 유용한 정보를 저장하고 관리하기 위한 집합적인 것(=table)

- ( ) : 테이블의 행(row). ( )의 집합 = 엔터티. 즉 엔터티 하나의 값. 저장된 데이터 내용의 전체 집합

- 특징

(1) 반드시 업무에 필요하고 관리하고자 하는 정보

(2) ( )에 의해 식별이 가능

(3) 영속적으로 존재하는 인스턴스의 집합

(4) 업무 프로세스에 의해 사용됨

(5) ( )이 있음 : 예외적으로 Associative Entity는 주식별자 속성만 있어도 괜찮음

(6) 다른 엔터티와 최소 한 개 이상의 ( )가 있음

- 분류

유뮤(有無)형에 따른 분류

(1) ( ) : 물리적인 형태 있음

(2) ( ) : 관리해야 할 개념적 정보로 구분됨

(3) ( ) : 업무를 수행함에 따라 발생되는 엔터티. 통계자료에 사용

발생시점에 따른 분류

(1) ( ) : 업무에 원래 존재하던 정보. 타 엔터티와의 관계에서 생성 X 독립적으로 생성

되며 다른 엔터티의 부모 역할을 함

(2) ( ) : 기본엔터티로부터 발생. 업무에 있어서 중심적인 역할을 함. 행위엔터티 생성.

(3) ( ) : 두 개 이상의 부모 엔터티로부터 발생. 자주 내용과 데이터량이 변경.

- 명명(命名)

(1) 업무에서 사용하는 용어 사용

(2) 약어 사용을 피한다

(3) 단수명사 사용

(4) 유일한 이름 부여

(5) 엔터티 생성 의미대로 이름 부여

**10. 속성(Attribute)**

- 정의 : 인스턴스로 관리하고자 하는 의미상 분리되지 않는 ( ) 데이터 단위. ( )를 설명하고 인스턴스의 구성요소.

- 특징

(1) 해당 업무에서 필요함

(2) 정해진 ( )에 함수적 종속성을 갖음

(3) 하나의 속성에는 단 하나의 ( )만 갖음. 하나의 속성에 여러 개의 값이 있으면 엔터티를

이용하여 분리한다

- 분류

특성에 따라

(1) ( ) : 업무로부터 추출한 모든 속성.

(2) ( ) : 업무상 필요한 데이터 이외에 모델링, 업무 규칙화를 위해 새로 만들거나 변형

된 속성. 일련번호 같은 단일한 식별자를 부여하기 위해 모델상에서 새로 정의하는 속성

(3) ( ) : 다른 속성에 영향을 받아 발생하는 속성. 일반적으로는 계산된 값. 데이터

정합성을 유지하기 위해 가급적 적게 정의하는 것이 좋음

엔터티 구성 방식에 따라

(1) ( ) : 엔터티를 식별하는 속성

(2) ( ) : 다른 엔터티와 관계에서 포함된 속성

(3) 일반속성

세부 의미를 쪼갤 수 있는지

(1) 복합속성(Composite Attribute)

(2) 단순속성(Simple Attribute)

- ( ) : 각 속성이 가질 수 있는 값의 범위. 엔터티 내 데이터타입, 크기, 제약 사항 지정

- 명명

(1) 해당업무에서 사용하는 이름 부여

(2) ( )으로 된 이름 X

(3) 약어 X

(4) 전체 데이터모델에서 유일성 확보 필요

**11. 관계**

- 정의 : 엔터티의 인스턴스 사이의 논리적인 ( ) 존재의 형태나 행위로서 서로에게 연관성이 부여됨

- 페어링 : 각각 엔터티의 인스턴스들은 자신과 관련된 인스턴스들과 관계를 맺는 것

- 분류

UML(Unified Modeling Language) : 연관관계(Association)와 의존관계(Dependency)가 있다. 실선과 점선 표기법으로 ( ) 표현됨. 연관관계는 이용하는 관계, 의존관계는 상대방 행위에 의해 관계 형성

반면 ERD는 존재적 관계와 행위에 의한 관계를 구분( )

- 표기법

(1) ( )

(2) ( ) : 1:1, 1:M, N:M 처럼 참여자의 수 표현

(3) ( ) : 필수 관계, 선택 관계

- 관계 정의 필수 요소

(1) 두 개의 엔터티 사이에 관심있는 ( )이 존재해야 함

(2) 두 개의 엔터티 사이에 ( )이 발생함

(3) 관계 연결에 대한 ( )이 서술됨

(4) 관계 연결을 가능하게 하는 ( )가 포함됨

**12. 식별자**

- 정의 및 키(key)와 비교 : 하나의 엔터티에 구성되어 있는 속성 중 엔터티를 대표할 수 있는 속성.

식별자는 ( )에서 사용되며 키(key)는 DB 테이블 접근을

위한 매개체로 물리적 데이터 모델링 단계에서 사용됨

- 특징

(1) 유일성 : 주식별자에 의해 엔터티 내 모든 인스턴스들이 유일하게 구분됨

(2) 최소성 : 주식별자를 구성하는 속성의 수는 유일성을 만족하는 최소의 수

(3) 불변성 : 주식별자의 값은 자주 ( )

(4) 존재성 : 주식별자가 지정되면 반드시 ( )이 들어와야 함( X)

- 분류

(1) 대표성 여부 – 주식별자/보조식별자

(2) 스스로 생성 여부 – 내부식별자/외부식별자

(3) 속성의 수 – 단일식별자/복합식별자

(4) 대체 여부 – 본질식별자/인조식별자

- 주식별자 도출기준

(1) 해당 업무에서 자주 이용되는 속성

(2) 명칭, 내역 등과 같이 ( )으로 기술되는 것은 가급적 피한다

(3) 복합으로 주식별자로 구성할 경우 ( ) 속성이 포함되지 않도록 한다

- 식별자 관계

부모로부터 받은 FK를 자식엔터티의 주식별자로 사용하고 그것만으로 주식별자로 사용하면 부모 자식간의 관계는 1:1이 된다. 또 다른 부모에게 받은 속성이나 스스로 가진 속성을 주식별자로 사용하면 1:M이다.

- 비식별자 관계(Non-Identifying Relationship)

부모로부터 속성을 받았지만 자식엔터티의 주식별자로 사용하지 않는 경우를 말한다.

(1) 자식엔터티가 받은 속성이 반드시 필수가 아니라 ( )이 생성될 경우

(2) 엔터티별로 데이터의 ( )를 다르게 관리할 경우

(3) 여러 개의 엔터티가 하나의 엔터티로 통합되어 표현되었는데 각각의 엔터티가 ( )를

가질 때

(4) 자식쪽 엔터티의 ( )를 부모엔터티와는 별도로 생성하는 것이 더 유리할 때

- 비식별자 관계로만 설정할 경우 문제점

자식엔터티에서 데이터를 처리할 때 부모엔터티까지 찾아가는 경우, SQL 구문에서 많은 조인이 필요하여 성능 저하을 유발할 수 있다.

- 비식별자 선택 고려 사항 프로세스

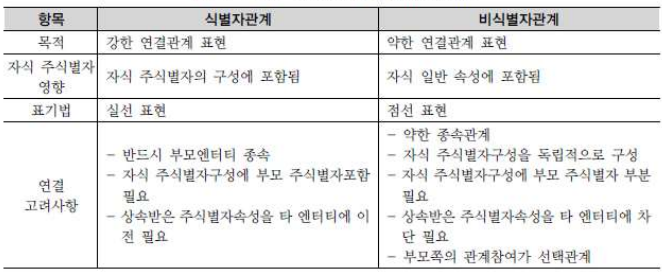
(1) 관계 분석

(2) 관계의 강/약 분석 – ( )면 비식별자 설정 고려

(3) 자식테이블 독립 PK의 필요성 여부 – 부모와 ( )가 필요하면 비식별자 설정 고려

(4) SQL - 복잡도 증가 및 개발생산성 저하 – PK 속성 단순화를 통해 비식별자 설정 고려

- 식별자와 비식별자 관계 비교



13. 추가 정리 – 엔터티, 인스턴스, 속성, 속성값

엔터티에는 두 개 이상의 ( )가 있다

엔터티에는 고유의 성격을 표현하는 ( )이 두 개 이상 있다

속성은 한 개의 ( )만 갖는다(Single-Valued Attribute)