**1과목 데이터 모델링의 이해**

**1-1 데이터 모델링의 이해**

**1-1-1 데이터 모델링의 이해**

모델링의 특징

1. 현실세계를 일정한 형식에 맞추어 표현하는 추상화의 의미를 가질 수 있음
2. 복잡한 현실을 제한된 언어나 표기법을 통해 이해하기 쉽게 하는 단순화의 의미를 가지고 있음
3. 애매모함을 배제하고 누구나 이해가 가능하도록 정확하게 현상을 기술하는 정확화의 의미를 가짐
4. 시스템 구현만을 위해 진행하지 않음, 시스템 구현을 포함한 업무분석 및 업무형상화를 하는 목적 있음

데이터 모델링이 필요한 이유

1. 업무정보를 구성하는 기초가 되는 정보들에 대해 일정한 표기법에 의해 표현
2. 분석된 모델을 가지고 데이터베이스를 생성하여 개발 및 데이터관리에 사용하기 위한 것
3. 데이터모델링 자체로서 업무의 흐름을 설명하고 분석하는 부분에 의미를 가짐

데이터 모델링 할 때 유의점

1. 중복성 : 여러 장소의 데이터베이스에 같은 정보를 저장하지 않도록 하여 중복성을 최소
2. 비유연성 : 데이터의 정의를 데이터의 사용 프로세스와 분리하여 유연성을 높인다
3. 비일관성 : 데이터간의 상호 연관관계를 명확하게 정의하여 일관성 있게 데이터가 유지되도록 한다

데이터 모델링 개념

1. 개념적 데이터 모델링 :
   1. 추상화 수준이 높고 업무중심적이고 포괄적인 수준의 모델링을 진행
   2. ~~전사적 데이터 모델링~~
   3. ~~EA수립시 많이 사용~~
2. 논리적 데이터 모델링 :
   1. 시스템으로 구축하고자 하는 업무에 대해 Key, 속성, 관계 등을 정확하게 표현
   2. ~~재사용성 높음~~
3. 물리적 데이터 모델링 :
   1. 실제로 데이터베이스에 이식할 수 있도록 성능, 저장 등 물리적인 성격을 고려하여 설계

데이터베이스 스키마 구조 3단계

1. ~~외부스키마 :~~
2. 개념스키마 :
   1. 모든 사용자 관점을 통합한 조직 전체 관점의 통합적 표현
   2. 모든 응용시스템이나 사용자들이 필요로 하는 데이터를 통합한 조직 전체의 DB를 기술한 것으로 DB에 저장되는 데이터와 그들간의 관계를 표현하는 스키마
3. ~~내부스키마 :~~

ERD 설명

1. 1976년 피터첸에 의해 E-R모델 표기법이 만들어짐
2. 엔터티를 어디에 배치하는가에 대한 문제는 필수사항이 아니다

ERD 작성 순서

1. 엔터티를 그린다
2. 엔터티를 적절하게 배치
3. 엔터티간 관계를 설정
4. 관계명을 기술
5. 관계의 참여도를 기술
6. 관계의 필수여부를 기술

**1-1-2 엔터티**

엔터티의 특징

1. 속성이 없는 엔터티는 있을 수 없다.(엔터티는 반드시 속성을 가져야 한다)
2. 엔터티는 다른 엔터티와 관계가 있을 수 밖에 없다.(단, 통계성 코드성 엔터티의 경우 생략 가능)
3. 데이터로서 존재하지만 업무에서 필요로 하지 않으면 해당 업무의 엔터티로 성립될 수 없다.
4. ~~반드시 해당 업무에서 필요하고 관리하고자 하는 정보이어야 한다.~~
5. 유일한 식별자에 의해 식별이 가능해야 한다
6. ~~영속적으로 존재하는 (2개 이상의)인스턴스의 집합이어야 한다~~
7. 엔터티는 업무 프로세스에 의해 이용되어야 한다
8. 엔터티는 반드시 속성이 있어야 한다
9. ~~엔터티는 다른 엔터티와 최소 한 개 이상의 관계가 있어야 한다.~~

기본 엔터티(키엔터티)

1. 다른 엔터티로부터 주식별자를 상속받지 않고 자신의 고유한 주식별자를 가진다

엔터티의 이름을 부여하는 방법

1. 현업의 업무 용어를 사용하여 업무상의 의미를 분명하게 한다
2. 모든 엔터티에서 유일한 이름을 부여
3. 엔터티가 생성되는 의미대로 자연스럽게 부여
4. ~~가능하면 약어를 사용하지 않는다~~
5. ~~단수명사를 사용~~

**1-1-3 속성**

속성(ATTRIBUTE)

1. 업무에서 필요로 하는 인스턴스에 관리하고자 하는 의미상 더 이상 분리되지 않는 최소의 데이터 단위

속성의 특징

1. 엔터티에 대한 자세하고 구체적인 정보를 나타낸다
2. 하나의 엔터티는 두 개 이상의 속성을 갖는다
3. 속성도 집합이다
4. 하나의 인스턴스에서 각각의 속성은 한 개의 속성값을 가져야 한다

속성의 특성에 따른 분류

1. 기본속성 : ~~기본적인 베이스~~
2. 설계속성 : ~~계산된 속성~~
3. 파생속성 : ~~정의된 속성~~

도메인

1. 가질 수 있는 값의 범위
2. 엔터티 내에서 속성에 대한 데이터타입과 크기, 제약사항을 지정

속성의 명칭 부여

1. 약어사용은 가급적 제한
2. 서술식 속성명은 사용하지 않는다
3. 해당업무에서 사용하는 이름을 부여
4. 전체 데이터모델에서 유일성 확보하는 것이 좋다

**1-1-4 관계**

데이터모델링의 관계 설명

1. 존재적 관계와 행위에 의한 관계를 구분하는 표기법이 없다
2. 존재에 의한 관계와 행위에 대한 관계로 구분될 수 있으나 ERD에서는 관계를 연결할 때, 존재와 행위를 구분하지 않고 단일화된 표기법을 사용
3. UML에서는 연관관계와 의존관계에 대해 다른 표기법을 가지고 표현한다
4. UML에는 클래스다이어그램 관계 중 연관관계와 의존관계가 있고 이것은 실선과 점선의 표기법으로 다르게 표현
5. 관계는 존재적 관계와 행위에 의한 관계로 나누어볼 수 있다.
6. 부서와 사원 엔터티 간의 ‘소속’관계는 존재적 관계의 사례
7. 주문과 배성 엔터티 간의 ‘배송근거’관계는 행위에 의한 관계의 사례
8. 관계 표기법은 관계명, 관계차수, 선택성 3가지 개념으로 표현

관계 표기법

1. 관계명
2. 관계차수 : 1:1, 1:M, M:N
3. 관계선택사항

관계 읽기

1. 두 개의 엔터티 사이에 관심있는 연관규칙이 존재?
2. 두 개의 엔터티 사이에 정보의 조합이 발생?
3. 업무기술서, 장표에 관계연결에 대한 규칙이 서술?
4. 업무기술서, 장표에 관계연결을 가능하게 하는 동사가 있는가?

**1-1-5 식별자**

식별자의 종류

1. ~~엔터티 내에서 대표성을 가지는가에 따라 (주, 보조)~~
2. ~~엔터티 내에서 스스로 생성되었는지 여부에 따라 (내부, 외부)~~
3. ~~단일 속성으로 식별되는가에 따라 (단일, 복합)~~
4. ~~업무적으로 의미가 있던 식별자 속성을 대체하여 새롭게 만든 식별자를 구분 (본질, 인조)~~

주식별자를 지정할 때 고려사항

1. 주식별자에 의해 엔터티 내의 모든 인스턴스들이 유일하게 구분
2. 주식별자를 구성하는 속성의 수는 유일성을 만족하는 최소의 수가 되어야 한다
3. 지정된 주식별자의 값은 자주 변하지 않는 것이어야 한다
4. 주식별자가 지정이 되면 반드시 값이 들어와야 한다.

주식별자의 특징

1. 유일성 : 주식별자에 의해 엔터티내에 모든 인스턴스들을 유일하게 구분
2. 최소성 : 주식별자를 구성하는 속성의 수는 유일성을 만족하는 최소의 수가 되어야 함
3. 불변성 : 주식별자가 한 번 특정 엔터티에 지정되면 그 식별자의 값은 변하지 않아야 함
4. 존재성 : 주식별자가 지정되면 반드시 데이터 값이 존재 (Null 안됨)

주식별자를 도출하기 위한 기준

1. 해당 업무에서 자주 이용되는 속성
2. ~~명칭, 내영 등과 같이 이름으로 기술되는 것들은 가능하면 지정하지 않는다.~~
3. 복합으로 주식별자로 구성할 경우 너무 많은 속성이 포함되지 않도록 한다

비식별자관계를 선택하는 기준

1. 관계의 강약을 분석하여 상호간의 연관성이 약할 경우 비식별관계를 고려
2. 자식테이블에서 독립적인 primary Key의 구조를 가지기 원할 때 비식별자관계를 고려
3. 모든 관계가 식별자 관계로 연결되면 SQL Where절에서 비교하는 항목이 증가되어 조인에 참여한 테이블에 따라 SQL문장이 길어져 SQL문의 복잡성이 증가되는 것을 방지하기 위해 비식별자관계를 고려

비식별자관계로 연결하는 것을 고려

1. 부모엔터티에 참조값이 없어도 자식엔터티의 인스턴스가 생성될 수 있다.
2. 여러 개의 엔터티를 하나로 통합하면서 각각의 엔터티가 갖고 있던 여러 개의 개별 관계가 통합
3. 자식쪽 엔터티의 주식별자를 부모엔터티와는 별도로 생성하는 것이 더 유리하다고 판단하는 경우
4. ~~부모엔터티의 인스턴스가 자식과 관계를 가지고 있지만 자식만 남겨두고 먼저 소멸될 경우~~

**1-2 데이터 모델과 성능**

**1-2-1 성능 데이터 모델링의 개요**

성능 데이터모델링

1. 데이터의 증가가 빠를수록 성능저하에 따른 성능개선비용은 증가한다
2. 데이터모델은 성능을 튜닝 하면서 변경이 될 수 있는 특징이 있다
3. 분석/설계 단계에서 성능을 고려한 데이터모델링을 수행할 경우 성능저하에 따른 Rework 비용을 최소화 할 수 있는 기회를 가지게 된다

데이터 모델링 수행 절차

1. 데이터모델링을 할 때 정규화를 정확하게 수행
2. 데이터베이스 용량산정을 수행
3. 데이터베이스에 발생되는 트랜잭션의 유형을 파악
4. 용량과 트랜잭션의 유형에 따라 반정규화를 수행
5. 이력모델의 조정, PK/FK조정, 슈퍼타입/서브타입 조정 등을 수행

성능 데이터 모델링 할 때 고려사항

1. 용량산정 : 전체적인 데이터베이스에 발생되는 트랜잭션의 유형과 양을 분석하는 자료가 되므로 성능데이터 모델링을 할 때 중요한 작업이 될 수 있다.
2. 물리적인 데이터 모델링 : PK/FK의 칼럼의 순서조정, FK인덱스 생성 등은 성능 향상을 위한 데이터 모델링 작업에 중요한 요소가 된다
3. 이력데이터 : 시간에 따라 반복적으로 발생이 되기 때문에 대량 데이터일 가능성이 높아 특별히 성능을 고려하여 칼럼 등을 추가하도록 설계해야 한다.

**1-2-2 정규화와 성능**

25p : 35번 ~ 26p : 39번

**1-2-3 반정규화와 성능**

반정규화를 고려할 때 판단요소

1. 반정규화 정보에 대한 재현의 적시성으로 판단

디스크 I/O를 줄이기 위한 반정규화 기법

1. 테이블추가 – 부분테이블추가

|  |  |  |
| --- | --- | --- |
| 테이블의 반정규화 | | |
| 기법 분류 | 반정규화 기법 |  |
| 테이블병합 | 1:1 관계 테이블 병합 |  |
| 1:M 관계 테이블 병합 |  |
| 슈퍼/서브타입 테이블 병합 |  |
| 테이블분할 | 수직분할 |  |
| 수평분할 |  |
| 테이블추가 | 중복테이블 추가 |  |
| 통계테이블 추가 |  |
| 이력테이블 추가 |  |
| 부분테이블 추가 |  |
| 칼럼의 반정규화 | | |
|  | 중복칼럼 추가 | 조인감소를 위해 여러 테이블에 동일한 칼럼을 갖는다 |
|  | 파생칼럼 추가 | 조회 성능을 우수하게 하기 위해 미리 계산된 칼럼을 갖는다 |
|  | 이력테이블 칼럼추가 | 최신값을 처리하는 이력의 특성을 고려하여 기능성 칼럼 추가 |
|  | PK에 의한 칼럼 추가 |  |
|  | 응용시스템 오작동을 위한 칼럼 추가 |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**1-2-4 대량 데이터에 따른 성능**

28p : 43번 ~ 29p : 44번

**1-2-5 데이터베이스 구조와 성능**

칼럼수가 많은 테이블에 대한 설명

1. 로우체이닝이 발생할 정도로 한 테이블에 많은 칼럼들이 존재할 경우 조회성능저하가 발생할 수 있다.
2. 트랜잭션이 접근하는 칼럼유형을 분석하여 1:1로 테이블을 분리하면 디스크 I/O가 줄어들어 조회 성능을 향상 시킬 수 있다

데이터 액세스 성능을 향상시키기 위한 방법

1. 하나의 테이블에 많은 양의 데이터가 저장되면 인게스를 추가하고 테이블을 몇 개로 쪼개도 성능이 저하되는 경우가 있다.
2. 이때 논리적으로 하나의 테이블이지만 물리적으로는 여러 개의 테이블로 분리하여 데이터 액세스 성능도 향상시키고, 데이터 관리방법도 개선할 수 있도록 테이블에 적용하는 기법을 파티셔닝이라고 한다

31p : 47번 ~ 34p : 51번

논리데이터모델 > 물리적인 테이블 형식으로 변환

1. 트랜잭션 : 항상 전체를 대상으로 일괄 처리하는데 테이블은 서브타입 별로 개별 유지하는 것으로 변환하면 Union 연산에 의해 성능이 저하 될 수 있다.
2. 트랜잭션 : 항상 서브타입 개별로 처리하는데 테이블은 하나로 통합하여 변환하면 불필요하게 많은 양의 데이터가 집적되어 있어 성능이 저하될 수 있다.
3. 트랜잭션 : 항상 슈퍼+서브 타입을 함께 처리하는데 개별로 유지 하면 조인에 의해 성능이 저하될 수 있다.

**1-2-6 분산 데이터베이스와 성능**

분산 데이터베이스의 분산설계를 하여 효율성을 증대

1. 공통코드, 기준정보 등 마스터 데이터는 분산데이터베이스에 복제분산을 적용
2. 거의 실시간 업무적인 특성을 가지고 있을 때 분산데이터베이스를 사용하여 구성할 수 있다.
3. 백업 사이트를 구성할 때 간단하게 분산기능을 적용하여 구성할 수 있다.