

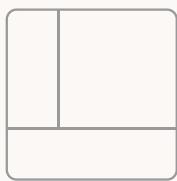
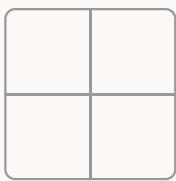
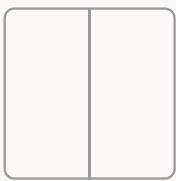
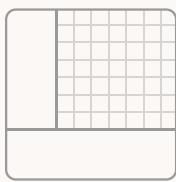
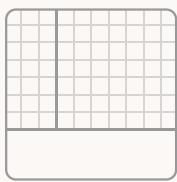
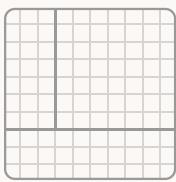
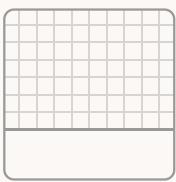
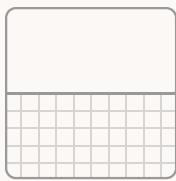
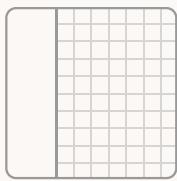
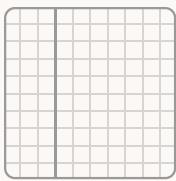
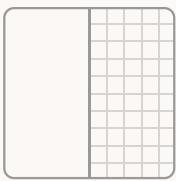
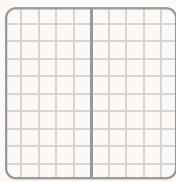
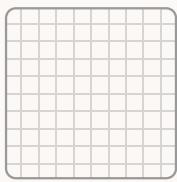
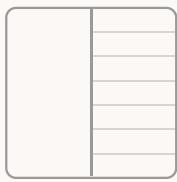
NOTEBOOK

SQLD

CONTENTS

| | | |
|-----|-------------------------------|----|
| 01. | 제 1 과목 – 데이터 모델링의 이해 | 1 |
| 02. | 제 1 과목 – 데이터 모델과 성능 | 2 |
| 03. | 제 2 과목 – SQL 기본 | 3 |
| 04. | 제 2 과목 – SQL 활용 | 4 |
| 05. | 제 2 과목 – 관리 구문 (SQL 최적화 기본원리) | 5 |
| 06. | [SQLP] 제 3 과목 – SQL 고급활용 및 튜닝 | 6 |
| 07. | | 7 |
| 08. | | 8 |
| 09. | | 9 |
| 10. | | 10 |
| 11. | | 11 |
| 12. | | 12 |

TEMPLATES



1
2
3
4
5
6
7
8
9
10
11
12

SECTION 1

데이터 모델링의 이해

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12

| <h2>데이터 모델링의 이해</h2> | | | | | | | |
|--|---|---------------------|--|-------------------------|---|-------------------------|---|
| <h3>데이터 모델링의 정의</h3> <ul style="list-style-type: none"> - 정보 시스템을 구축하기 위한 데이터 관점의 업무분석 기법. - 데이터 베이스를 구축하기 위한 분석/설계 과정. - 현실 세계의 데이터를 추상화, 단순화, 명확화하기 위해 일정한 표기법에 의해 표현하는 기법. | <h3>데이터 모델링의 유익점</h3> <ul style="list-style-type: none"> - 데이터의 품질 (데이터의 신뢰도, 정확성을 나타내는 최도)을 보장하기 위해 모델링 시 유의해야 할 점. | | | | | | |
| <h3>데이터 모델링의 특징</h3> | <p>① 추상화 (Abstraction) : 현실 세계를 일정한 형식으로 표현하는 것. 아이디어나 개념을 간략하게 표현하는 과정. → 일정한 형식에 맞추어 표현.</p> | | | | | | |
| <p>② 단순화 (Simplification) : 복잡한 현실세계를 정해진 표기법으로 단순하고 쉽게 표현하는 것. → 이해하기 쉽도록 저작된 표기법 / 언어로 표기.</p> | <p>③ 명확화/정확화 (Clarity) : 불분명함을 제거하고 명확하게 해석할 수 있도록 기술하는 것. → 모호함을 제거하여 정확하게 현실을 기술.</p> | | | | | | |
| <h3>데이터 모델링의 세 가지 관점</h3> | <p>〈데이터 모델링 시 유익점 . "중·비·유·일"〉</p> <table border="1"> <thead> <tr> <th>증복 (Duplication)</th> <th>같은 데이터가 여러 엔티티에 중복으로 저장되는 현상을 지양해야 한다.</th> </tr> </thead> <tbody> <tr> <td>비유연성 (Inflexibility)</td> <td>데이터 모델의 설계에 따라 애플리케이션의 소소한 변경에도 데이터 모델이 수시로 변경되어야 하는 상황이 생길 수 있다. 이런 상황은 시스템을 유지보수하는 데에 어려움을 가중시킬 수 있다.</td> </tr> <tr> <td>비일관성 (Inconsistency)</td> <td>데이터의 중복이 없는 경우에도 비일관성이 발생할 수 있다. 개발자가 다른 데이터와의 연관성을 고려하지 않고 일부 데이터만 변경할 수 있기 때문이다. 이런 위험을 예방하기 위해 데이터 모델링을 할 때 데이터 간의 연관 관계에 대해 명확하게 정의해야 한다.</td> </tr> </tbody> </table> | 증복 (Duplication) | 같은 데이터가 여러 엔티티에 중복으로 저장되는 현상을 지양해야 한다. | 비유연성 (Inflexibility) | 데이터 모델의 설계에 따라 애플리케이션의 소소한 변경에도 데이터 모델이 수시로 변경되어야 하는 상황이 생길 수 있다. 이런 상황은 시스템을 유지보수하는 데에 어려움을 가중시킬 수 있다. | 비일관성 (Inconsistency) | 데이터의 중복이 없는 경우에도 비일관성이 발생할 수 있다. 개발자가 다른 데이터와의 연관성을 고려하지 않고 일부 데이터만 변경할 수 있기 때문이다. 이런 위험을 예방하기 위해 데이터 모델링을 할 때 데이터 간의 연관 관계에 대해 명확하게 정의해야 한다. |
| 증복 (Duplication) | 같은 데이터가 여러 엔티티에 중복으로 저장되는 현상을 지양해야 한다. | | | | | | |
| 비유연성 (Inflexibility) | 데이터 모델의 설계에 따라 애플리케이션의 소소한 변경에도 데이터 모델이 수시로 변경되어야 하는 상황이 생길 수 있다. 이런 상황은 시스템을 유지보수하는 데에 어려움을 가중시킬 수 있다. | | | | | | |
| 비일관성 (Inconsistency) | 데이터의 중복이 없는 경우에도 비일관성이 발생할 수 있다. 개발자가 다른 데이터와의 연관성을 고려하지 않고 일부 데이터만 변경할 수 있기 때문이다. 이런 위험을 예방하기 위해 데이터 모델링을 할 때 데이터 간의 연관 관계에 대해 명확하게 정의해야 한다. | | | | | | |
| <p>① 데이터 관점 (What, Data) :</p> <p>데이터 우주의 모델링. 어떤 데이터들이 업무와 일어나는지, 데이터 간에 어떤 관계가 있는지에 관하여 모델링하는 방법.</p> | <p>1) 개념적 데이터 모델링 (Conceptual Data Modeling)</p> <ul style="list-style-type: none"> - 전사적 (회사 전체 차원의) 데이터 모델링 수행 시 행태지며, 추상화 레벨이 가장 높다. - 이 단계에서는 업무 중심적이고 포괄적 수준의 모델링이 진행된다. → 추상화 수준이 높고 업무 중심적 / 전체에 대해 포괄적인 수준의 모델링. | | | | | | |
| <p>② 프로세스 관점 (How, Process) :</p> <p>프로세스 우주의 모델링. 이 업무가 실제로 처리하고 있는 일의 무엇인지, 앞으로 처리해야 할 일의 무엇인지를 모델링하는 방법.</p> | <p>2) 논리적 데이터 모델링 (Logical Data Modeling)</p> <ul style="list-style-type: none"> - 재사용성이 가장 높은 모델링으로 DB 모델에 대한 키 (Keys), 속성 (Attributes), 관계 (Relationships) 등을 모두 표현하는 단계이다. → {키, 속성, 관계}를 표현 / 재사용성 높음 / 장구화를 노린다. | | | | | | |
| <p>③ 데이터와 프로세스의 상관 관점 (Data vs. Process, Interaction) :</p> <p>데이터와 프로세스의 관계를 위주로 한 모델링. 프로세스의 흐름에 따라 데이터가 어떤 영향을 받는지를 모델링하는 방법.</p> | <p>3) 물리적 데이터 모델링 (Physical Data Modeling)</p> <ul style="list-style-type: none"> - 실제 DB로 구현할 수 있도록 성능이나 가용성 등의 물리적 성격을 고려하여 모델을 표현하는 단계. → 실제 DB에 이식 / 물리적 성격 / 개념적 모델링보다 구체적. | | | | | | |

데이터 모델링의 이해

○ 데이터의 독립성

- 데이터 베이스에 대한 사용자들의 관점과 데이터가 실제로 표현되는 물리적 방식을 분리하여 독립성을 보장할 수 있다.



< 사용자 - 애플리케이션 - 데이터베이스 >

- 데이터의 독립성이 보장될 경우, 사용자 입장에서는 필요한 데이터만 조회할 수 있으면 되고, DB의 내부 구조를 굳이 알 필요 없다.
- DBA의 입장에서는 데이터의 독립성이 보장될 경우, 애플리케이션에 영향을 주지 않고 DB의 구조를 바꿀 수 있다.

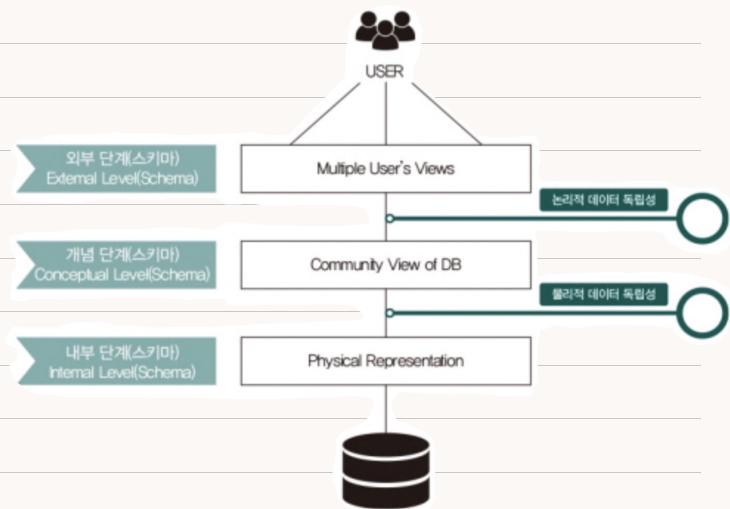
□ 독립성의 필요성

- 유저보수 비용 절감.
- 중복된 데이터 줄임.
- 데이터 복잡도 낮춤.
- 요구사항 대응률 높이기 용이.

American National Standards Institute -
Standards Planning And Requirements Committee

○ 데이터 모델링의 주요 3가지 개념

- 어떤 것 (Things) : 업무가 관여하는 사물 또는 개념.
- 성격 (Attributes) : 어떤 것 (Things)이 가지는 성격.
-



< DB 3단계 구조와 각 영역의 데이터 독립성 >

□ 데이터 베이스 3단계 구조

- 데이터베이스 관리시스템의 추상적인 설계표준인 ANSI - SPARC
- 아키텍처에서는 DB 구조를 3단계로 나눈다.

① 외부 스키마 (External Schema) :

여러 사용자의 관점. Multiple User's View 단계. 각 사용자가 보는 데이터베이스의 스키마를 정의.

② 개념 스키마 (Conceptual Schema) :

통합된 관점. Community View of DB 단계. 모든 사용자가 보는 데이터베이스의 스키마를 통합하여 전체 데이터베이스를 나타내는 것. DB에 저장되는 데이터들을 표현하고 데이터들 간의 관계를 나타냄.

③ 내부 스키마 (Internal Schema) :

물리적인 관점. Physical Representation 단계. 실질적인 데이터의 저장 구조나 커럼정의, 인덱스 등을 포함한 물리적 저장 구조를 나타냄.

□ 3단계 스키마 구조가 보장하는 독립성

① 논리적 독립성 :

개념 스키마가 변경되어도 외부스키마는 영향 받지 않는다.

② 물리적 독립성 :

내부 스키마가 변경되어도 외부/개념 스키마는 영향 받지 않는다.

데이터 모델링의 이해

○ ERD (Entity Relationship Diagram)

- 시스템에서 어떤 엔터티들이 존재하여 그들 간에 어떤 관계가 있는지를 나타내는 다이어그램.

□ ERD 표기방식

① Peter Chen: 주로 대학교재에서 사용하는 표기법. 실무에서 사용하는 경우는 드물.



② IDEFIX (Integration Definition for Information Modeling):

ERWin에서 사용되는 모델. 실무에서 사용하는 경우도 있음.



③ IE/Crow's Foot: '까마귀발 표기법'이라고도 부르며 가장 많이 사용됨.

ERWin, ERStudio에서 사용.



④ Min-Max / ISO: 각 엔터티의 참여도를 좀 더 상세하게 나타내는 표기법.



□ ERD 작성 순서

① 엔터티를 도출하고 그린다.



② 엔터티를 적절하게 배치한다.



③ 엔터티 간의 관계를 설정한다.



④ 관계명을 기입한다.



⑤ 관계의 참여도를 기입한다.



⑥ 관계의 필수/선택 여부를 기입한다.

⑤ UML: 주로 소프트웨어 공학에서 사용되는 모델.



● IE/Crow's Foot 표기법

| 기호 | 의미 |
|-------|--------------------------------------|
| □ | 엔터티 |
| ○ | 0개 |
| । | 1개 |
| < | 2개 이상 |
| — | 부모 엔터티의 속별자가 자식 엔터티의 주 속별자(속별자 관계) |
| ----- | 부모 엔터티의 속별자가 자식 엔터티의 일반 속별자(비속별자 관계) |

⑥ Case Method/Barker: Oracle에서 사용되는 모델. Crow's Foot과 비슷함.



데이터 모델링의 이해

○ 엔터티 (Entity)

- 저장이 되기 위한 어떤 것. '실체', '객체'.
- 사전적인 의미는 '독립체'이며, 데이터베이스에서 엔터티는 식별이 가능한 객체라는 의미를 갖는다.
- 업무에서 쓰이는 데이터를 용도별로 분류한 그룹.

□ 엔터티의 특징

- 업무에서 꼭 필요로 하는 정보이다.
- 식별자에 의해 식별이 가능해야 한다.
- 2개 이상의 인스턴스로 구성된 집합이다.
- 업무 프로세스에 의해 이용된다.
- 반드시 "속성"을 포함해야 한다.
- 다른 엔터티와 "관계"가 최소 1개 이상 존재한다.

인스턴스 (Instance) : 엔터티 안의 행 데이터.

ex) '학생' 엔터티 → '홍길동', '논개' 데이터 인스턴스.

② 중심 엔터티 : 기본 엔터티로부터 파생되고, 행위 엔터티를 생성.

업무에 있어서 중심적인 역할을 하며 데이터의 양이 많이 발생함.
예) 주문, 매출, 계약 등..

③ 행위 엔터티 : 2개 이상의 엔터티로부터 파생.

데이터가 자주 변경되거나 증가할 수 있음.
예) 주문 내역, 이벤트 응모 이력 등..

유/무형에 따른..

발생 시점에 따른..

유형
(서원, 상품...)

기본/기
(서원, 부서...)

사건
(주문, 칭구...)

개념
(책, 장소...)

중심
(주문, 계약...)

행위
(주문내역, 작업내역...)

엔터티 (Entity) ----- Table

인스턴스 (Instance) ----- Row

속성 (Attribute) ----- Column

□ 엔터티의 분류

- 유형 / 무형에 따른 분류

① 유형 엔터티 :

② 개념 엔터티 :

③ 사건 엔터티 :

- 발생 시점에 따른 분류.

① 기본 엔터티 : 업무에 원래 존재하는 정보.

독립적으로 생성되어 자식 엔터티를 가질 수 있음.

다른 엔터티로 주식별자 상속 X.

자신의 고유 주식별자를 가짐.

예) 상품, 회원, 서원, 부서 등..



SECTION 2

데이터 모델과 성능

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12

SECTION 3

SQL 기본

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12

SECTION 4

SQL 활용

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12

SECTION 5

관리 구문 (SQL 최적화 기본원리)

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12

SECTION 6

[SQLP]

SQL 고급 활용 및 튜토리얼

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12

SECTION 7

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12

SECTION 8

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12

SECTION 9

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12

SECTION 10

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12

SECTION 11

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12

SECTION 12

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12

TEMPLATES

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12

| | | |
|--|--|----|
| | | 1 |
| | | 2 |
| | | 3 |
| | | 4 |
| | | 5 |
| | | 6 |
| | | 7 |
| | | 8 |
| | | 9 |
| | | 10 |
| | | 11 |
| | | 12 |

