

데이터 모델링

김정연

010-6214-6127

jykipool@naver.com

데이터 모델링

데이터 모델링이란 주어진 업무로부터 업무에 필요한 정보를 도출하고 이를 약속된 표기법으로 표현하여 이를 시스템으로 구축하는 과정이라 할 수 있다.

모델

작품을 만들기 전에 만든 물건, 또는 완성된 작품의 대표적인 보기

데이터
모델

현실세계의 정보나 데이터를 시스템으로 구축하기 위해 추상화 하여 체계적으로 표현한 것

데이터
모델링

복잡한 현실 세계에 존재하는 데이터를 단순화 시켜 표현해 컴퓨터 세계의 데이터베이스로 옮기는 변환 과정

“데이터 모델은 정보시스템을 구축하고 업무 정책을 설계하거나 또는 기업 전략을 바꾸기 위한 청사진이며 계획이다.” (Zachman, 1987)

모델링에 사용되는 3가지 원리



분류화 Classification

- 관계 있는 개체들을 그룹화
- 시스템에 요구되는 데이터의 특징을 규명할 때, 각각의 엔티티를 정의하는 것보다 엔티티 타입으로 묶어 정의



추상화 Abstraction

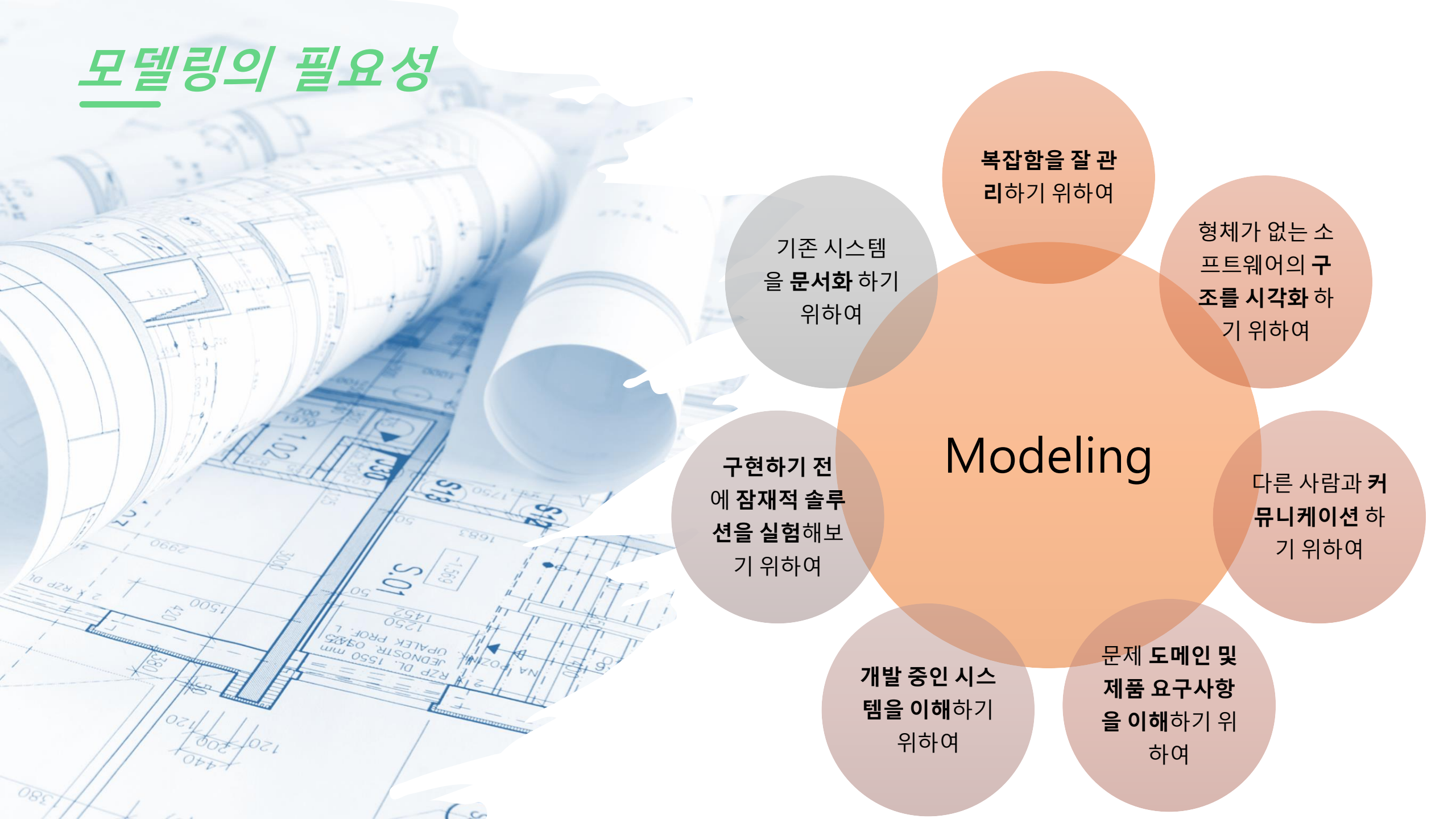
- 자세함을 무시하며 필요한 부분만을 추출하는 활동
- 현실에서 출발하되 불필요한 부분을 제거하고 사물의 본질을 드러나게 하는 과정



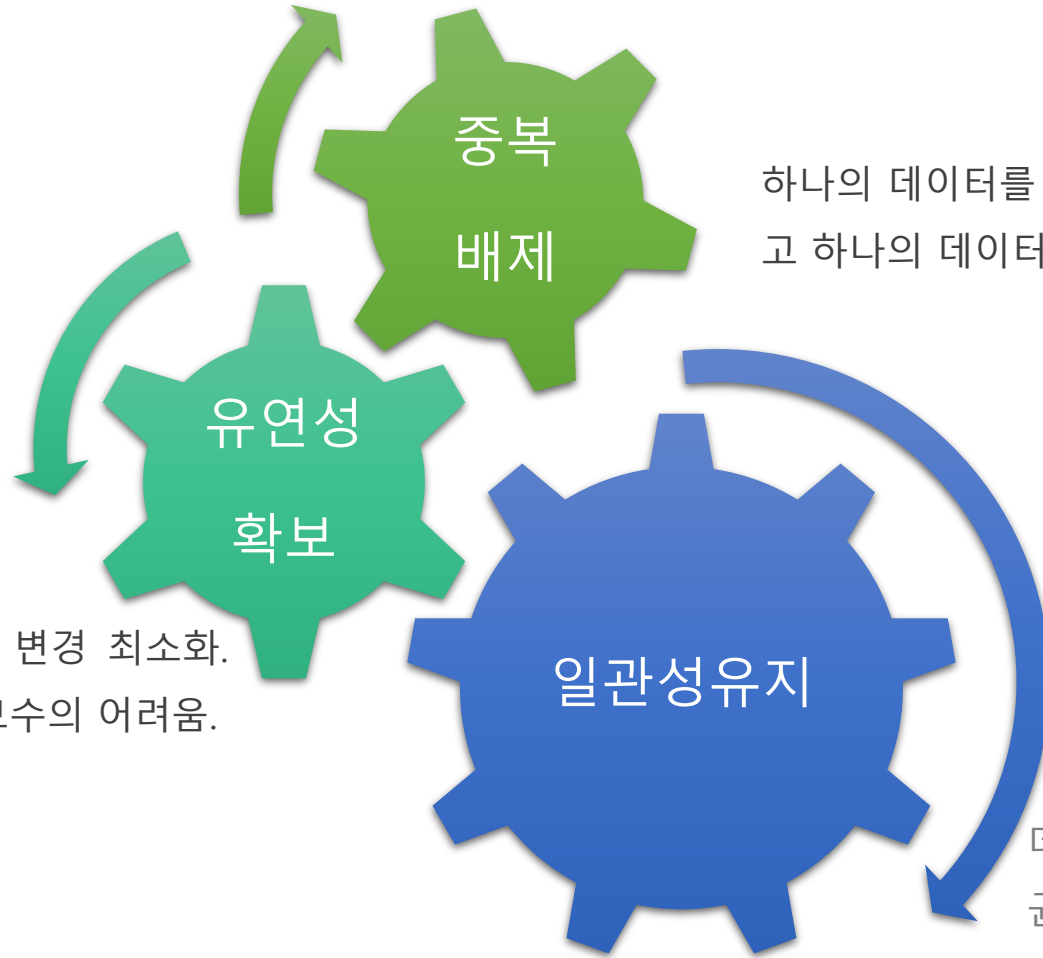
일반화 Generalization

- 개체 사이의 유사성이 존재할 때 이 유사성을 모아 하나의 새로운 개체 타입을 정의하는 것
- 상위 개체의 정보가 하위 개체에 그대로 상속
- 재사용을 위한 의미적 종속관계

모델링의 필요성



데이터모델링 시 유의사항

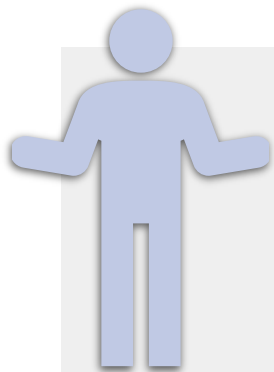


하나의 데이터를 여러 장소에서 관리하는 것을 지양하고 하나의 데이터는 하나의 장소에서 관리.

업무변화에 따른 데이터 모델의 변경 최소화.
데이터 모델을 변경한다면 유지보수의 어려움.

데이터가 다양한 응용프로그램 간에 이동될 때 정보를 균일하게 유지하는 것.

Entity Relationship 모델 구성요소



엔티티(Entity)

- 현실세계에 실제로 존재(학생, 교수, 고객)하거나 개념적(조직, 강의)인 것.
- 엔티티는 업무를 구현하는데 필요한 정보이고 관리해야 할 주체, 대상, 행위 등의 집합적인 것.



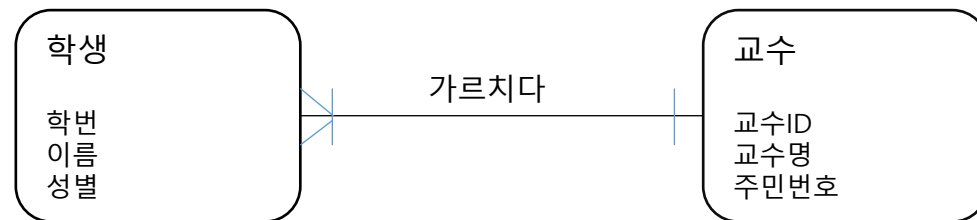
관계(Relationship)

- 엔티티와 엔티티 간의 존재하는 업무규칙을 정의.
- 관계수(cardinality), 선택성(Optionality), 식별성(Identifier Inheritance) 등을 정의

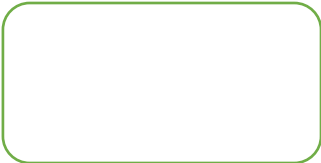
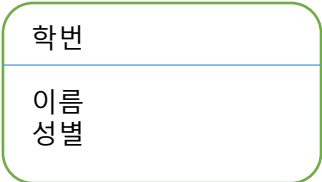


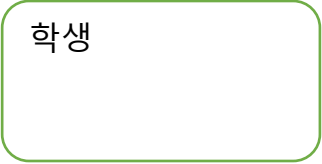
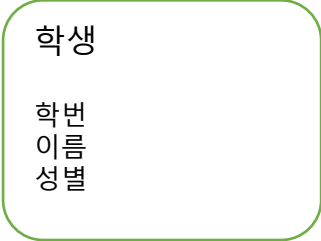



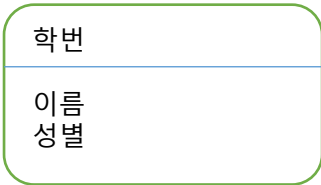




속성(Attribute)

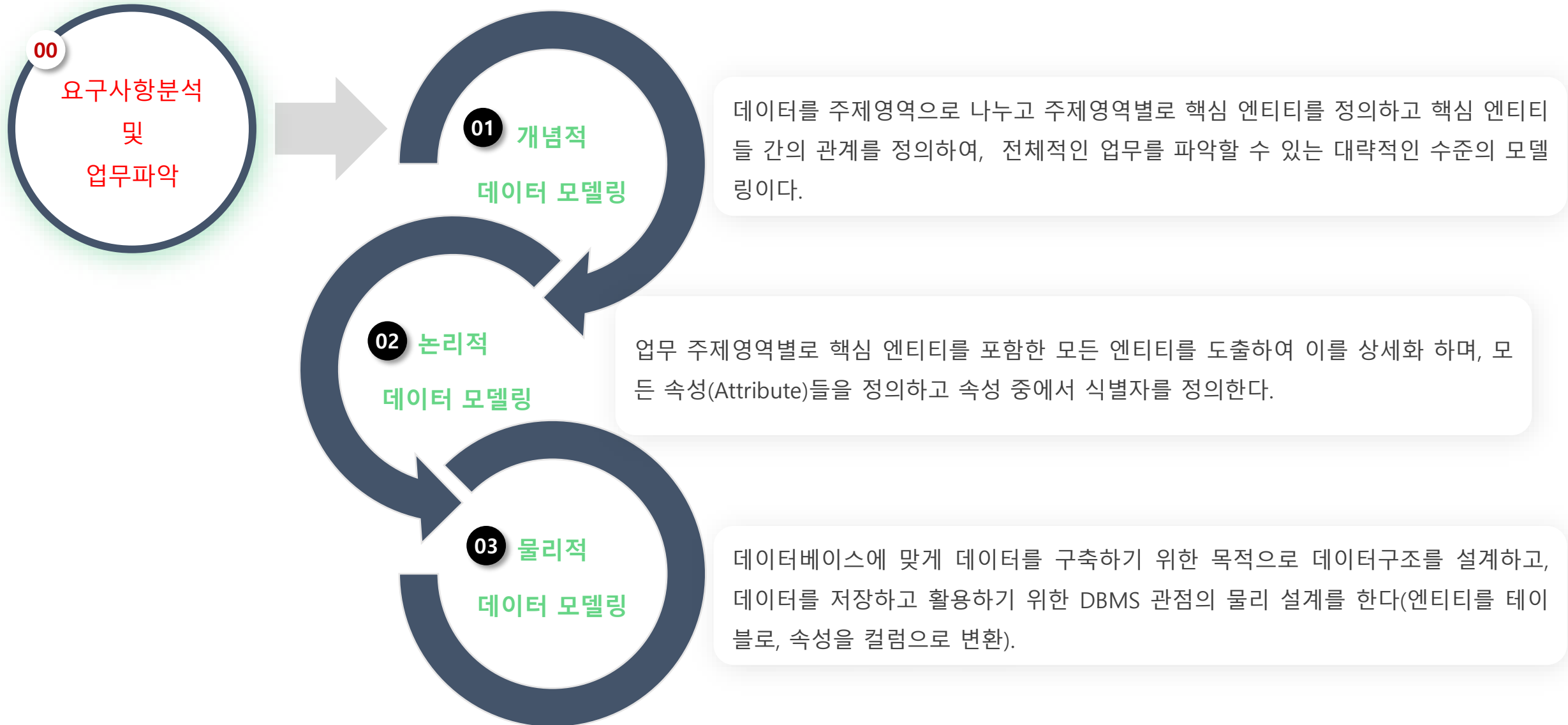
- 데이터를 표현하는 가장 작은 단위.
- 엔티티 특성이나 상태를 정의
- 하나의 엔티티는 두 개 이상의 속성을 가짐.



데이터모델링 표기법(Entity Relationship 모델)

표기법	엔티티(Entity)	속성(Attribute)	관계(Relationship)
Information Engineering	<p>학생</p> 	<p>학생</p> 	<p>  1:1  1:M </p>
Barker	<p>학생</p> 	<p>학생</p> 	<p>  1:1  1:M </p>
IDEF1X (Integration Definition for Information Modeling)	<p>학생</p> 	<p>학생</p> 	<p>  1:1  1:M </p>

데이터 모델링 과정



0. 요구사항관리

데이터를 비롯하여 관련 애플리케이션 및 시스템 전반에 걸친 사용자의 요구를 수집하고 분류하여 반영하는 작업으로, 사용자의 정보 요구 사항을 종합적으로 검토, 확인하여 요건에 적합하도록 시스템을 개선, 반영함으로써 사용자의 만족도를 높이고 고품질의 서비스를 가능하게 한다.

요구사항 정의서	작성	검토	승인
	김성환		
	7/26	/	/

□ 기본사항

프로젝트명			
작성자	김성환	작성일	2007-07-25

□ 기능적 요구사항 (Functional Requirement)

업무	요구사항	ID	비고
신규기능 추가	주요한 메뉴 카테고리 사용	R-0001	
	합업창을 관리 할 수 있는 기능 추가	R-0002	
	이벤트 관리기능 추가	R-0003	
	설문조사 이벤트 개발	R-0004	
	서식 중 관리자가 상위에 표시 할 수 있는 기능 추가	R-0005	
	민원업무 표시, 글씨 정렬, 명칭변경	R-0006	
	민원신청 내역 조회 기능 확대기능 추가	R-0007	
	메인에서 민원서비스 조회 기능 추가	R-0008	
	건강보험서식 변경에 따른 수정, 국민연금 조속종액 신규 추가	R-0009	
	정보마당 관리기능 추가	R-0010	
	차량설치가 안될 때 수동설치 기능 추가	R-0011	
	RSS 기능 추가	R-0013	
	사업장관리번호 일련화 관련 방안	R-0014	
	전체 페이지 스크랩 기능 추가	R-0016	
	사업자등록 번호 국제청 G4C 연계 기능추가	R-0017	
	가입자 등록 내역, 사업장 가입 내역 출력	R-0018	
	서식창기가 어려우세요 기능 추가	R-0019	
	전자민원마리해보기 기능 추가	R-0020	
	발의 회원 표시	R-0021	
	관리자페이지 글자사항 추가	R-0022	
	관리자페이지 오류사항 정보공유 기능추가	R-0023	
	정보보호 SSL 기능추가	R-0024	
	사업장의 정보 변경 시 사업장 내역변경 신고 페이지로 이동	R-0025	
	사업장 경우 1년이상 활동 없을 시 자동 탈퇴 기능추가	R-0026	
	인증서 오류 시 Yesign.or.kr 로 이동 기능 추가	R-0027	
	인증서 등록/추가 부분 삭제 기능 추가	R-0028	
	인증서 오류 시 수동설치 페이지로 이동	R-0029	
	여러명 업로드 시 엑셀 업로드 기능 추가	R-0030	

요구사항 명세서			
작성한		검토	승인
7/09	/	/	/

■ 기능적 요구사항

업무 명		신규기능추가		
요구사항 ID		R-0001	요구사항 명	주요한 메뉴 카테고리 사용
개요		주요한 메뉴들을 모든페이지에서 볼 수 있도록 화면에 사용함		
요구사항 내역	상세설명	<ul style="list-style-type: none"> 전자민원 공인인증서 전자민원마리해보기 민원현황조회는 중요한 부분으로 모든 페이지에서 확인 할 수 있었으면 한다. 		
	유형	기능		
	중요도	중	난이도	중
	출처	관련부서 포털서비스팀		

업무 명		신규기능추가		
요구사항 ID		R-0002	요구사항 명	합업창을 관리 할 수 있는 기능 추가
개요		합업창 관리 기능 추가		
요구사항 내역	상세설명	기존에 합업창 관리가 없어서 일일이 작업 하였다.		
	유형	기능		
	중요도	하	난이도	하
	출처	관련부서 포털서비스팀		

업무 명		신규기능추가		
요구사항 ID		R-0003	요구사항 명	이벤트 관리기능 추가
개요		이벤트의 이벤트를 관리 할 수 있도록 화면에 추가함		
요구사항 내역	상세설명	이벤트의 이벤트를 관리 할 수 있도록 관리자에 기능 추가		
	유형	기능		
	중요도	하	난이도	하
	출처	관련부서 포털서비스팀		

요구사항 정의 필수요소

1. 요구사항 출처 및 위치
2. 요구사항 ID
3. 요구사항 타이틀 및 설명
4. 기능, 비기능 요구사항
5. 수용여부

0. 요구사항관리 : 요구사항 수집 방법



0. 요구사항관리 : 요구사항분류

외부 인터페이스 요건

- 외부 기관이나 외부 시스템과의 사이에서 발생하는 모든 입/출력에 대한 요건으로 외부 기관이 추가 변경되거나 제도 및 기준 등이 변화하여 인터페이스 형식이 바뀌었을 경우.
- 중복성 배제, 표준 준수

기능 개선 요건

- 애플리케이션에서 입력 발생하는 출력에 대한 요건으로 기능 개선 요건에는 입력에 대한 유효 체크, 정확한 처리 순서, 비정상 상태에 대한 반응(오버플로우, 통신 장비, 에러 처리), 매개변수의 기능, 입출력 관계, 입출력 순서, 입력을 출력으로 변환하는 공식 등이 포함되어야 함.
- 불가변성, 범용성 준수

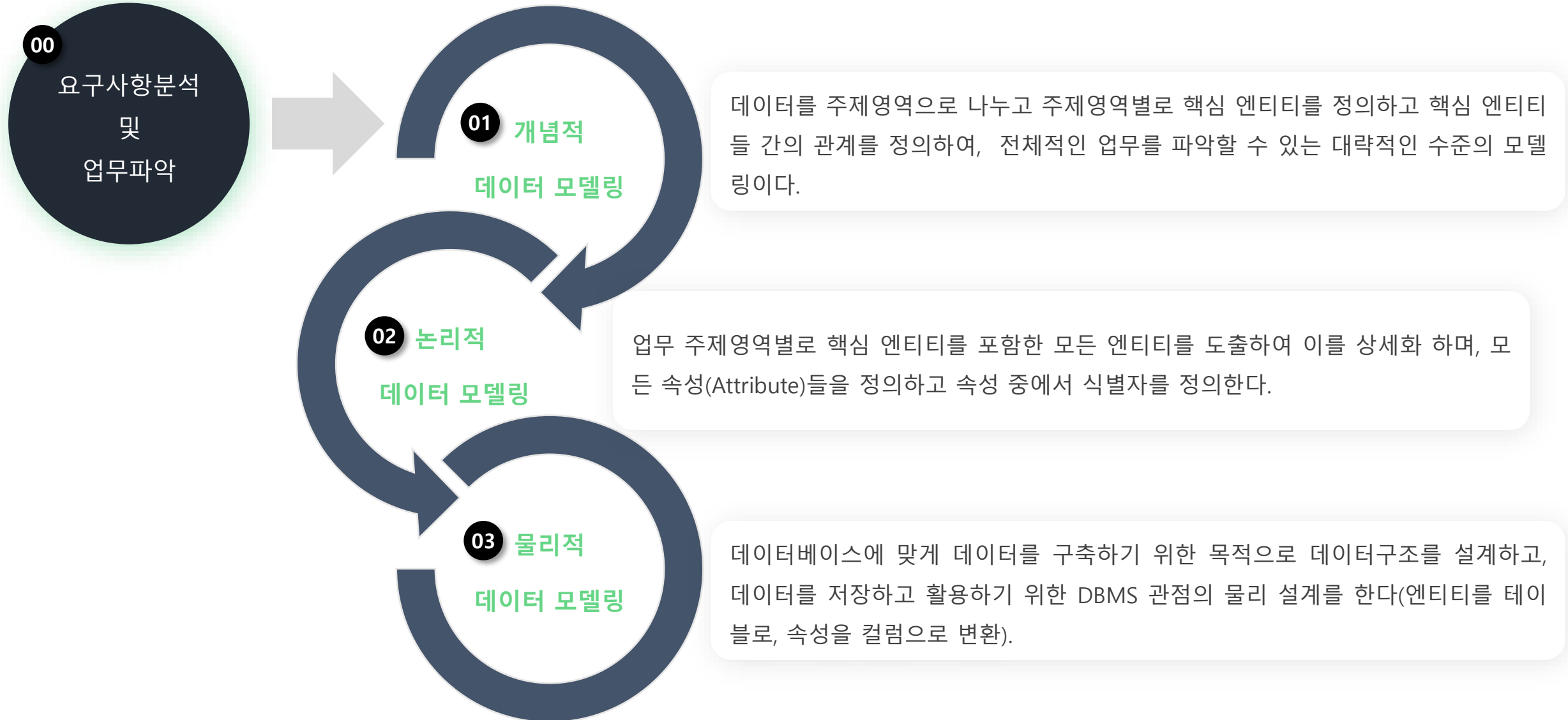
성능 개선 요건

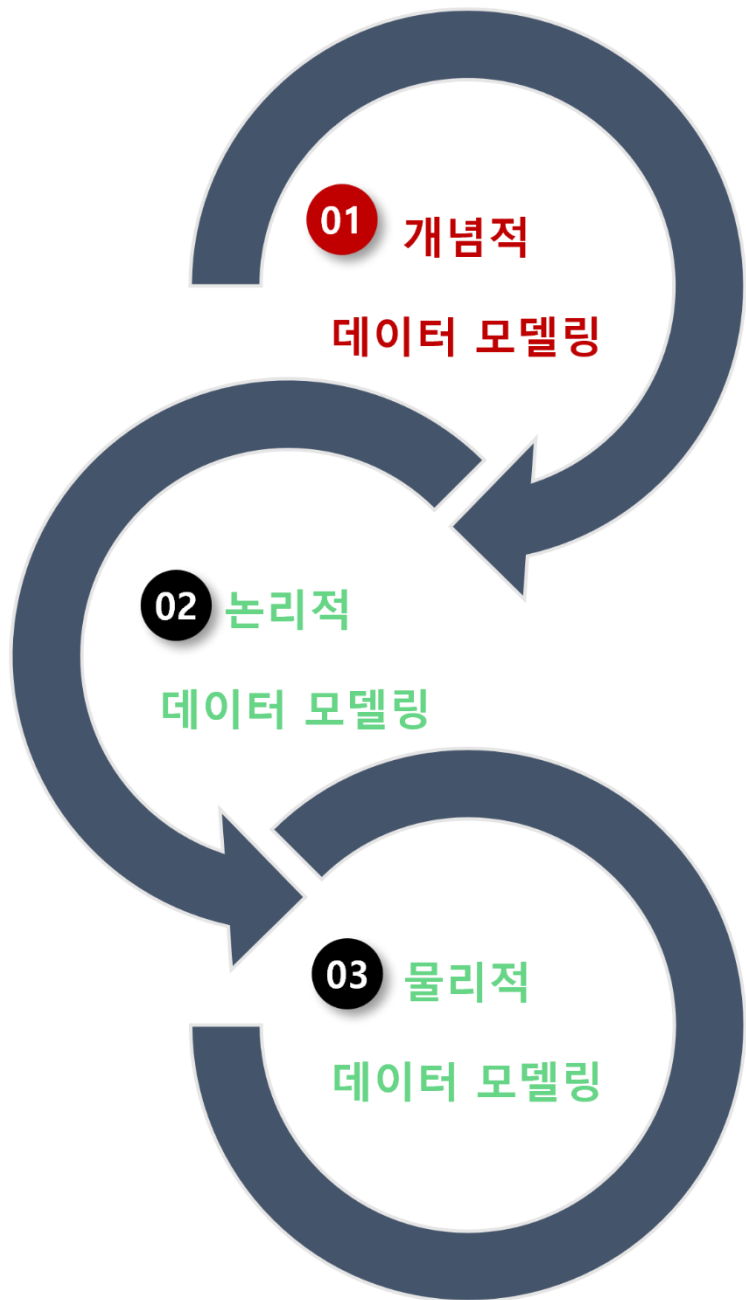
- 해당 기관의 사용자가 필요로 하는 성능 개선 사항으로 정적인 수치 요구 사항으로, 지원하는 터미널 수, 지원하는 동시 사용자 수, 처리하는 정보의 양과 종류 등이 있으며 동적인 수치 요구 사항은 일정한 기간 내에 처리하는 트랜잭션이나 작업의 수 등. 각 기관의 서비스 특성을 고려하여 정적, 동적 기준을 마련하고 해당 기준에 맞게 서비스되고 있는지를 모니터링 작업을 통해 항시적으로 관리해야 함.
- 실현 가능성, 측정 가능성 판단

보안 개선 요건

- 중요 데이터에 대한 훼손, 변조, 도난, 유출에 대한 물리적 접근 통제(제한 구역, 동제 구역 등) 및 사용 통제(인증, 암호화, 방화벽 등)에 대한 요건. 보안 개선 요건에 대한 관리를 위해서는 먼저 보안 관리가 필요한 정보에 대한 등급을 설정하고, 해당 등급별로 접근 가능한 이용자의 등급을 관리해야 함. 또한 접근 방식에 있어서의 접근 통제 기준 및 사용 통제 기준이 제시되어야 한다. 해당 기준에 따라 모니터 작업을 통해 안정적인 서비스가 제공될 수 있도록 관리.
- 보안 개선 요건이 향후에 재변경되지 않도록 근본적인 개선 방안을 요청해야 하며, 실현 가능성 여부를 판단해야 함

데이터 모델링 과정



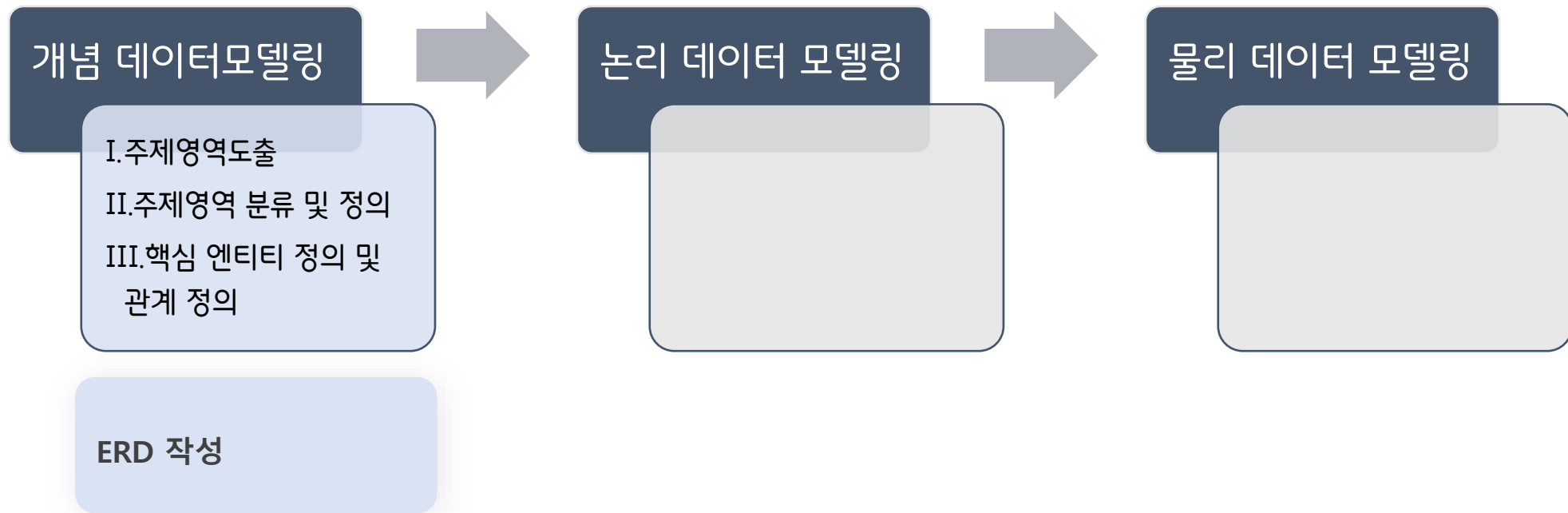


데이터 모델링 과정

1. 개념적 데이터 모델링



1. 개념적 데이터 모델링



1. 개념적 데이터 모델링

I. 주제영역 도출

주제영역이란

- 기업이 비전이나 목표를 달성하기 위하여 사람들이 관심을 기울이는 업무에서 발생하는 데이터의 묶음(집합)
- 데이터 Divide & Conquer 방식에 유리(Top-Down)
- 주제영역 하위에는 소속된 주제영역 혹은 엔티티 타입이 존재
- 주제영역은 항상 관련된 업무기능(Function)이 존재

주제영역 도출

- 업무에서 사용하는 용어 위주로 도출
- 업무지침서의 목차
- 기업의 조직 및 팀 구성
- 현행시스템
- 인터넷 시스템의 상위 메뉴
- 업무 기능 이름
- 상향식 : 엔티티를 그룹핑
- 하향식 : 주제영역으로 부터 엔티티 도출

1. 개념적 데이터 모델링

II. 주제영역 분류 및 정의

주제영역 분류

- 데이터 관점
- 업무관점

주제영역 정의

- 주제영역을 설명하고 주제영역에 해당되는 데이터 범위나 내용을 명확하게 정의
- 중복 배제, 동일한 기준 적용

Who
주체

What(How)
대상/자원

Where
장소

Biz. Event(When/Why포함)
약정/계약

Additional Biz. Event, Status
행위/이벤트/상태

핵심 데이터 주제영역

중요 데이터 주제영역

출처: 비투엔

1. 개념적 데이터 모델링

III. 핵심 엔티티 정의 및 관계 정의

핵심 엔티티 식별

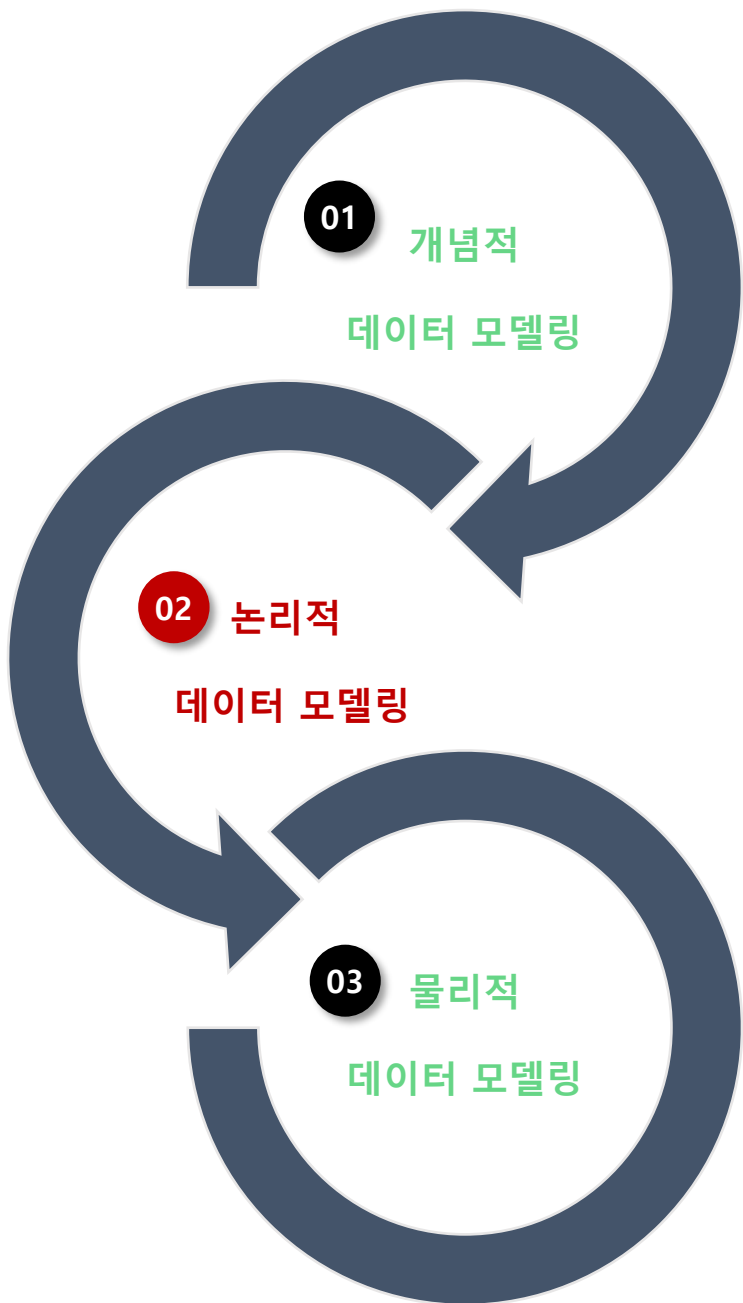
- 최하위 주제 영역별 핵심 엔티티를 최소 1개 이상 도출
- 만약 주제 영역에 엔티티가 없다면 주제영역을 과도하게 세분화한 것임
- 핵심 엔티티는 업무의 주체, 대상, 자원, 장소 등에 해당되는 엔티티
- 가급적 주제 영역별로 핵심 엔티티 수가 크게 차이 나지 않도록 하는 것이 좋음

관계 정의

- Relationship : 엔티티와 엔티티 간의 관계
- 개념적 모델의 경우 엔티티 간의 관계의 유무와 관계 명 정도를 정의하여 전체적인 업무 관점에서 데이터의 연관성을 파악할 수 있도록 하는 것이 일반적임

속성 및 식별자 정의

- 속성을 모두 도출할 필요는 없지만 식별자 및 주요 속성은 정의하는 것이 좋음
- 식별자는 엔티티 개념을 가장 잘 표현할 수 있는 속성으로 구성

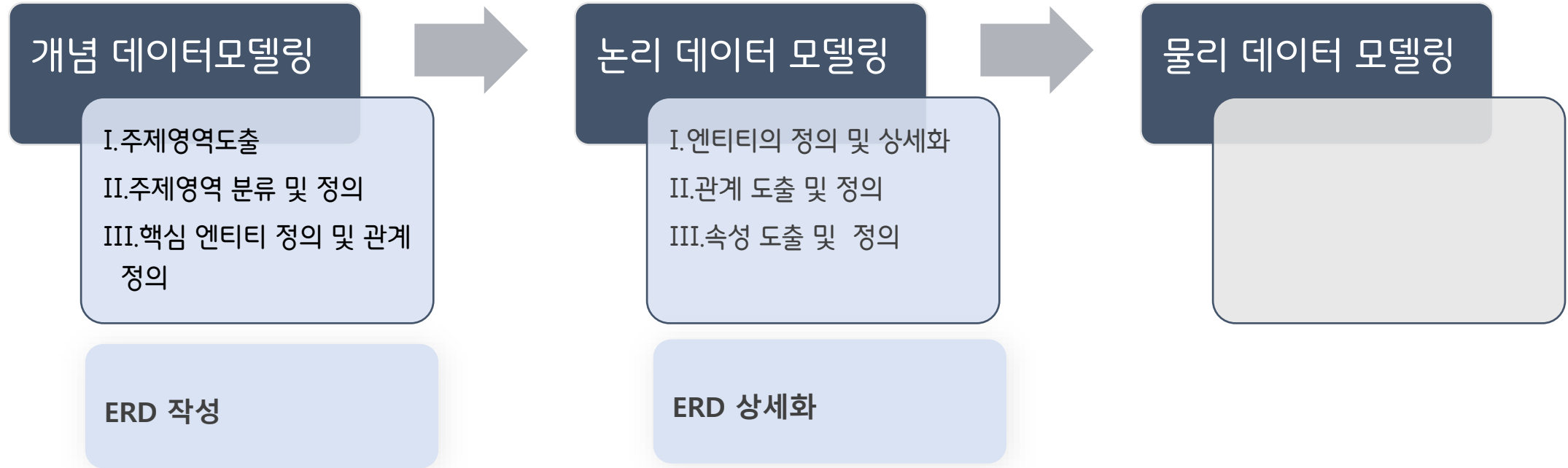


데이터 모델링 과정

2. 논리적 데이터 모델링



2. 논리적 데이터 모델링



2. 논리적 데이터 모델링

I. 엔티티 정의 및 상세화

핵심 엔티티

(Key Entity,

기준정보)

유형 및 분류(각종 코드 정보)/업무규칙 및 지식 (업무 규정, 지식 비즈니스 이벤트 발생 조건)

업무 주체 및 대상 (비즈니스 이벤트의 주체 또는 대상)/장소(비즈니스 이벤트가 발생하는 장소)

독립적으로 생성가능, 부모엔티티(독립적으로 식별자를 가짐)

예) 사원, 부서, 고객, 상품

중요 엔티티

(Main Entity,

업무기본정보)

업무주체와 업무 대상 간의 거래/업무 행위

업무 핵심 기능에서 반복적으로 발생하는 데이터

핵심 엔티티 간의 엔티티 또는 핵심 엔티티로부터 발생

상위 엔티티의 주식별자를 상속받지 않고 독립적인 식별자를 가짐

예) 계약, 카드정보

행위 엔티티

(Action Entity,

업무상세)

업무행위에 대한 상세/내역

업무결과/상태

부모엔티티로 부터 발생

예)카드사용실적, 공사내역

2. 논리적 데이터 모델링

II. 엔티티 정의 및 상세화

엔티티 도출 및 식별

- 관심대상이 되는 데이터를 분석하여 엔티티를 구체화 하는 과정
- 현행 데이터 모델 참조
- 현행 데이터베이스 리버스
- 요구사항, 업무, 현업 담당자 및 데이터베이스 관리자와의 인터뷰 등을 통해 To-Be 모델을 고려하여 작성 (데이터 표준화 필요)

엔티티 명명 및 정의

- 전체 주제영역 내에서 유일
- 최적화된 단수형 명사
- 수식어를 사용
- 교차 엔티티는 양쪽 엔티티명을 조합하여 사용
- 엔티티는 데이터 집합을 규정하고 데이터 발생 규칙 및 업무규칙 등을 기술하여 정의
- 독립적인 엔티티의 경우엔 식별자 중심으로 정의
- 종속적인 엔티티의 경우에는 엔티티 간의 관계 성격, 데이터 발생 기준 등을 정의

엔티티 통합

- 일반화를 이용하여 구조적 측면에서 하나의 엔티티로 설계 또는 여러 시스템이나 업무 개별적으로 관리하던 데이터를 통합관리
- 분류체계, 식별체계, 속성체계를 일원화

2. 논리적 데이터 모델링

II. 관계 도출 및 정의

관계 도출

- 주체 : 핵심 엔티티와 중요 엔티티 간의 관계에서 업무 행위의 주체를 나타냄
- 대상 : 핵심 엔티티와 중요 엔티티 혹은 행위 엔티티 간의 관계에서 업무 행위의 대상을 나타냄
- 상세 : 중요 엔티티와 행위 엔티티 간의 관계에 해당
- 인과 : 중요 엔티티와 행위 엔티티, 중요 엔티티와 행위 엔티티 간의 관계에서 업무행위의 원인이나 근거를 나타냄
- 역할/자격 : 핵심 엔티티와 다른 엔티티 간의 관계에서 업무 행위에 대해 특정역할을 나타냄
- 참조 : 핵심 엔티티와 다른 엔티티 간의 관계에서 엔티티 고유 정보에 부속한 정보들을 참조하는 것을 나타냄

관계 정의

- 엔티티 간의 관계수(Cardinality) 및 선택성, 식별자 상속을 정의
- 1:1 관계 : (필수-필수 관계), (필수-선택 관계), (선택-선택 관계)
- 1:M 관계 : (필수-필수 관계), (필수-선택 관계), (선택-선택 관계)
- M:N 관계

관계명 정의

- 관계명은 일반적으로 하나의 엔티티에서 다른 엔티티를 바라보는 입장에서 명명

2. 논리적 데이터 모델링

III. 속성 도출 및 정의

속성의 도출

- 업무 요건을 만족할 수 있는 모든 속성을 도출하고 속성명, 속성정의, 필수여부, 데이터 타입 등을 정의한다.
- **핵심 엔티티의 속성**: 식별자 및 명칭, 특성/특징, 접촉정보, 위치정보, 약관/정책, 관계 속성 등
- **중요 엔티티 및 행위 엔티티의 속성**: 거래 주체나 대상 엔티티의 식별자에 해당하는 참조식별자, 계약일자, 계약 금액 등의 거래정보 등의 업무처리정보를 포함, 코드 성격이 데이터

속성 명 부여 및 속성의 정의

- 속성명을 정의할 때에는 엔티티의 명을 앞에 붙여 속성을 설명할 수 있는 명확한 명칭을 부여
- 표준단어들의 조합으로 구성
- 속성= 수식어+주제어+...+ 도메인
- 엔티티 명과는 다른 명칭을 부여하고 도메인 명으로 만도 부여 가능
- 속성을 정의할 때는 속성의 설명 및 발생규칙 등을 정의
- 엔티티를 통합할 때는 인조식별자 사용가능
- 도메인 및 데이터 타입 정의: 데이터 값의 범위 제한
- Optionality/Mandatory 지정 : 속성 데이터 입력 여부 정의
- Default 값 지정 : Null일 경우 기본값을 지정 가능

2. 논리적 데이터 모델링

데이터의 무결성 데이터의 무결성은 데이터의 정확성, 일관성, 유효성이 유지되는 것을 의미

개체 무결성

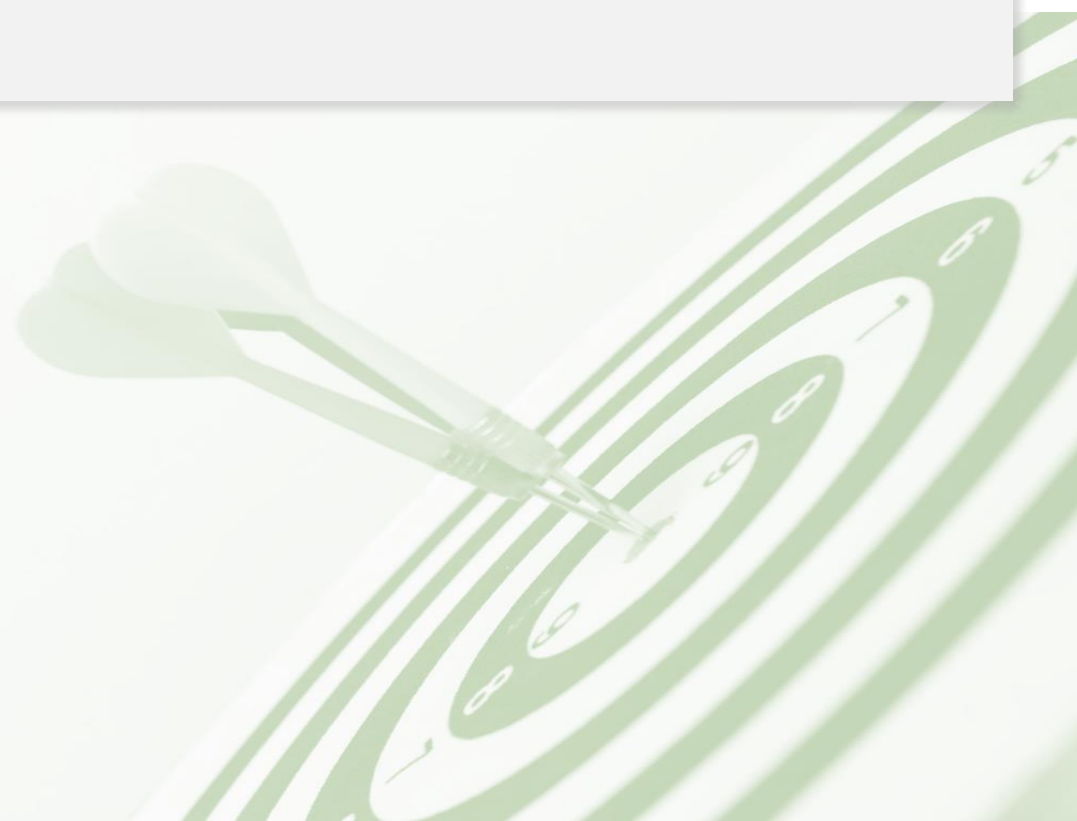
- 모든 엔티티(테이블)는 기본 키 (primary key)로 선택된 속성 (column)를 가져야 한다. 기본 키로 선택된 속성은 고유한 값을 가져야 하며, 빈 값은 허용하지 않는다.

관계 무결성

- 관계형 데이터베이스 모델에서 참조 무결성은 참조 관계에 있는 두 테이블의 데이터가 항상 일관된 값을 갖도록 유지되는 것을 말한다.

도메인 무결성

- 도메인 무결성은 엔티티(테이블)에 존재하는 속성의 무결성을 보장하기 위한 것으로 속성의 타입, NULL값의 허용 등에 대한 사항을 정의하고, 올바른 데이터가 입력 되었는지를 확인하는 것이다. DBMS의 기본값 설정, NOT NULL 옵션 등의 제약 사항으로 도메인 무결성을 보장할 수 있다.



2. 논리적 데이터 모델링

Key의 종류와 기능

1. 후보키 (Candidate Key)

릴레이션을 구성하는 속성들 중에서 튜플을 유일하게 식별할 수 있는 속성들(유일성과 최소성을 만족)의 부분집합 하나 이상의 후보키가 존재하며, 기본키가 될 수 있는 키들.

ex) <학생> 릴레이션에서 '학번'이나 '주민번호'는 기본키로 사용할 수 있는 후보키임.

2. 기본키 (Primary Key)

후보키 중에서 선택한 주키(Main Key)로 null 불가

한 릴레이션에서 특정 튜플을 유일하게 구별할 수 있는 속성으로 동일한 값이 중복으로 저장될 수 없음.

ex) <학생> 릴레이션에는 '학번'이나 '주민번호'가 기본키

3. 대체키 (Alternate Key)

후보키가 둘 이상일 때 기본키를 제외한 나머지 후보키로, 보조키라고도 함.

ex) <학생> 릴레이션에서 '학번'을 기본키로 정의하면 '주민번호'는 대체키가 됨.

4. 슈퍼키 (Super Key)

슈퍼키는 한 릴레이션 내에 있는 속성들의 집합으로 구성된 키로서 릴레이션을 구성하는 모든 튜플 중 슈퍼키로 구성된 속성의 집합과 동일한 값은 나타내지 않음(유일성은 만족시키나, 최소성은 만족시키지 못함).

ex) <학생> 릴레이션에서는 '학번', '주민번호', '학번'+ '주민번호', '학번'+ '주민번호'+ '성명' 등으로 슈퍼키를 구성할 수 있음.

5. 외래키 (Foreign Key)

관계(Relation)를 맺고 있는 릴레이션 R1, R2에서 릴레이션 R1이 참조하고 있는 릴레이션 R2의 기본키와 같은 R1 릴레이션의 속성

외래키는 참조되는 릴레이션의 기본키와 대응되어 릴레이션 간에 참조 관계를 표현하는데 중요한 도구로 사용되며, 외래키로 지정되면 참조 테이블의 기본키에 없는 값은 입력할 수 없음.

정규화(Normalization)

I. 정규화의 의미와 이상현상

정규화의 의미

정규화는 논리적 데이터 모델을 일관성이 있고 중복을 제거하여 보다 안정성을 갖는 바람직한 자료구조로 만들기 위해 여러 단계를 거친다. 그 단계는 제 1차 정규형에서부터 제 5차 정규형과 BCNF(Boyce-Codd Normal Form)까지로 구성되어 있다.

정규화의 필요성

- 중복의 제거로 저장 공간의 최소화
- 종속성 삭제로 일관성 및 무결성 보장
- 자료의 삽입, 갱신 및 삭제에 따른 이상현상(Anomaly) 제거
- 데이터 신규 발생시 DB 재구성의 필요성을 감소(유연한 구조)
- 연관관계 이용한 관리 및 이해 편리

이상현상

잘 만들어진 데이터 모델이라고 해도 엔터티에 데이터를 삽입, 수정, 삭제 할 때 오류가 발생할 개연성을 가지고 있다. 이러한 것들을 통칭해서 변경 이상(Modification Anomaly)이라고 한다. 변경 이상이 발생하는 엔터티를 그대로 운용하게 되면 데이터의 일관성, 무결성을 해칠 가능성이 있다. 정규화 과정을 통해서 변경 이상의 엔터티를 정규화된 엔터티로 변환하게 된다.

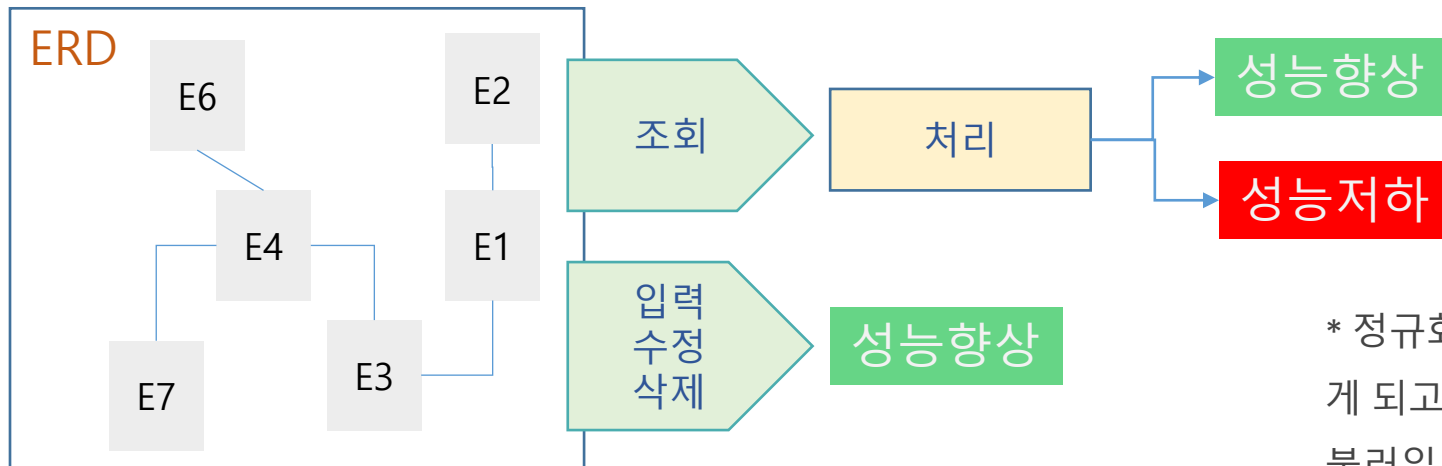
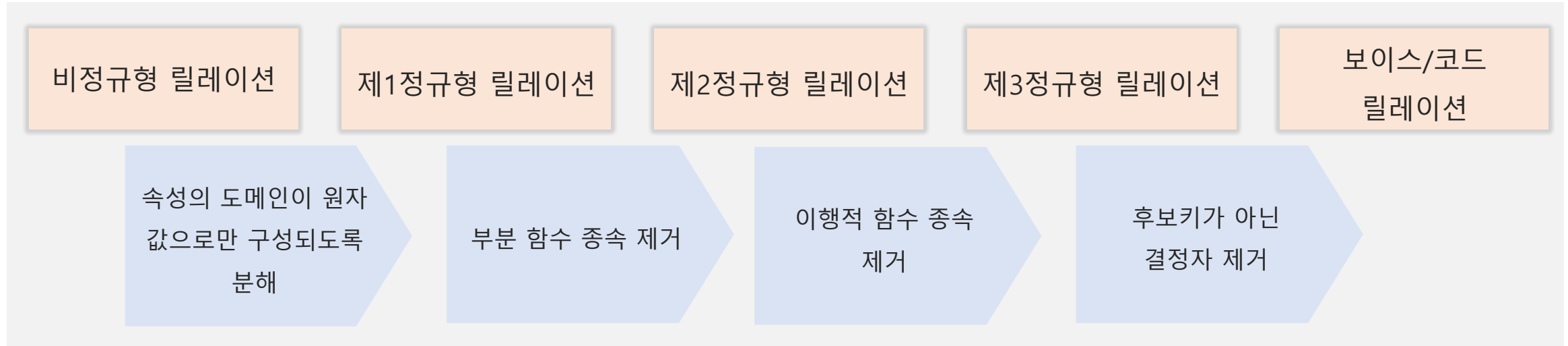
입력 이상 (Insertion Anomaly) : 별도의 사실이 발생하기 전까지 원하는 데이터를 삽입할 수 없음. 어떤 데이터를 삽입하려고 할 때 불필요하게 원하지 않는 데이터도 함께 삽입되게 된다.

삭제 이상(Deletion Anomaly) : 일부 정보를 삭제함으로써 유지되어야 할 정보까지도 삭제되는 연대 삭제가 발생한다.

수정 이상(Update Anomaly): 일부 속성 값을 갱신 함으로써 원하지 않는 정보의 이상 현상(무결성 파괴, 정보의 모순성)이 발생 한다

정규화(Normalization)

정규화 절차



* 정규화로 인해 테이블(엔티티)가 분해되다 보면 많은 **조인**을 하게 되고 결국 CPU와 메모리를 많이 사용하여 **처리속도의 지연**을 불러일으킨다. 이를 해결하기 위하여 결국 **반정규화**를 시행하기도 한다.

정규화(Normalization)

정규화 필요성 및 원칙

정규화의 필요성

중복의 제거로 저장 공간의 최소화

종속성 삭제로 일관성 및 무결성 보장

자료의 삽입, 갱신 및 삭제에 따른
이상현상(Anomaly) 제거

데이터 신규 발생시 DB 재구성의
필요성 감소(유연한 구조)

연관관계 이용한 관리 및 이해 편리

정규화의 원칙



정보의 무손실

분해된 정보는 Relation에 의해 합해졌을 때 원래의 정보가 손실되지 않도록 분해되어야 하며, 분해되기 전보다 이상적인 구조여야 함.



데이터의 중복성 감소

중복으로 인한 이상현상(입력이상, 삭제이상, 수정이상)을 제거해야 함.



분리의 원칙

하나의 독립된 관계성은 분리된 관계성으로 표현해야 함.

정규화(Normalization)

II. 제1차 정규화

제 1차 정규화

- 모든 속성은 **반드시 하나의 값을** 가져야 한다. 즉, 반복되는 형태가 있으면 안 되고 **동일한 형식**이어야 한다.
- 레코드들은 서로 식별 가능해야 합니다(반드시 식별자 선언).
- 각 속성들은 유일한 이름을 가져야 한다.
- **속성의 원자성 확보**(모든 속성 값은 더는 분해할 수 없는 하나의 값만 가지며, 다중 값을 가질 수 없다)

학생_이름	수강_수업_이름
홍길동	국어, 수학, 영어
성춘향	영어, 과학
이몽룡	사회, 체육

속성의 원자성은 엔티티의 1개 속성에는 1개 정보만 들어가야 한다.

예를 들어 아래 테이블과 같이 '수강_수업_이름'이라는 컬럼에 '국어, 영어, 수학' 이렇게 3개 정보가 동시에 들어가 있으면 이는 속성의 원자성을 위배하는 것이다.

이와 같은 경우 '수강_수업_목록'이라는 별도 엔티티를 만들어 속성의 원자성 문제를 해결할 수 있다.

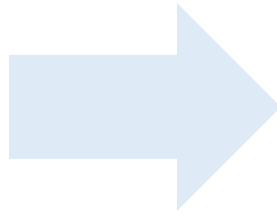
정규화(Normalization)

제1차 정규화

강의번호	강의명	내용	시작일	종료일
LEC01	A	데이터 모델링	2017-04-20	2017-05-20
LEC02	A	데이터 모델링	2017-06-01	2017-07-01

교육과정별강의

- ☐ # 교육과정ID
- ☐ ○ 강의명
- ☐ ○ 강의내용
- ☐ # 강의ID
- ☐ ○ 내용
- ☐ ○ 시작일
- ☐ ○ 종료일
- ☐ ○ 모집인원
- ☐ ○ 총강의시간
- ☐ ○ 일강의시간
- ☐ ○ 강의료
- ☐ ○ 정부지원여부
- ☐ # 강사명
- ☐ ○ 강사주소
- ☐ ○ 강사전화번호
- ☐ ○ 강사등급
- ☐ ○ 강사등급코드



교육과정

- ☐ # 교육과정번호
- ☐ ○ 강의명
- ☐ ○ 강의내용

개설하다

강의

- ☐ # 강의번호
- ☐ * 강의명
- ☐ ○ 내용
- ☐ ○ 종료일
- ☐ ○ 시작일
- ☐ ○ 모집인원
- ☐ ○ 총강의시간
- ☐ ○ 일강의시간
- ☐ ○ 강의료
- ☐ ○ 정부지원유무
- ☐ # 교육과정번호_1
- ☐ ○ 강사명

정규화(Normalization)

II. 제2차 정규화

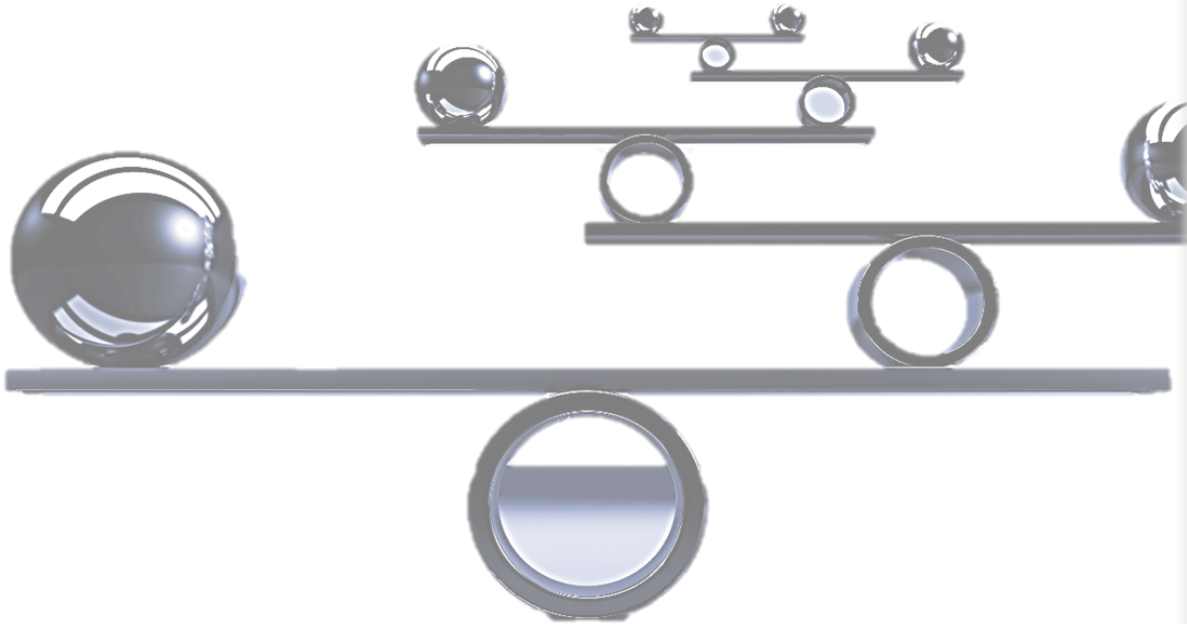
제 2차 정규화

- 식별자가 아닌 모든 속성들은 식별자 전체 속성에 완전 종속되어야 한다.
- 기본키가 아닌 모든 속성들이 기본키에 종속적이어야 2차 정규형을 만족할 수 있다.
- 부분 함수적 종속적을 완전 함수적 종속적으로 바꿔주는 것이다.

함수 종속성은 수학에서의 함수와 같이 두 필드의 집합이 many-to-one 관계로 사상되는 것을 말한다. 즉, 함수와 같이 어떠한 값을 통해 종속 관계에 있는 다른 값을 유일하게 결정할 수 있다는 것이다.

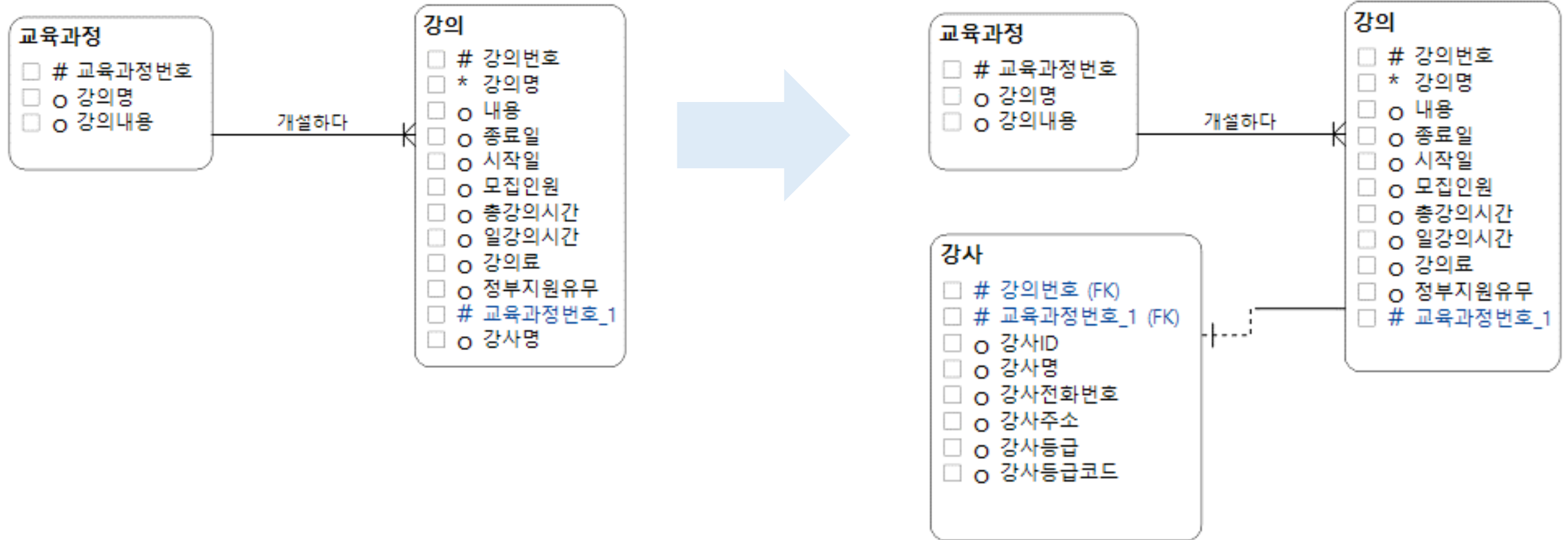
어떤 테이블 R에 존재하는 필드들의 부분집합을 각각 X와 Y라고 할 때, X의 한 값이 Y에 속한 오직 하나의 값에만 사상될 경우에 "Y는 X에 함수 종속 (Y is functionally dependent on X)"이라고 하며, $X \rightarrow Y$ 라고 표기한다.

예를 들어, 테이블에 [생일]과 [나이]라는 필드가 존재할 경우에 [나이] 필드는 [생일] 필드에 함수 종속이다. 즉, 생일을 알고 있다면, 나이에 대한 필드를 참조하지 않거나, 아예 필드를 유지하지 않아도 나이를 결정할 수 있다. 모델링 단계에서 함수 종속적인 속성을 찾아 중복성을 최소화한다.



정규화(Normalization)

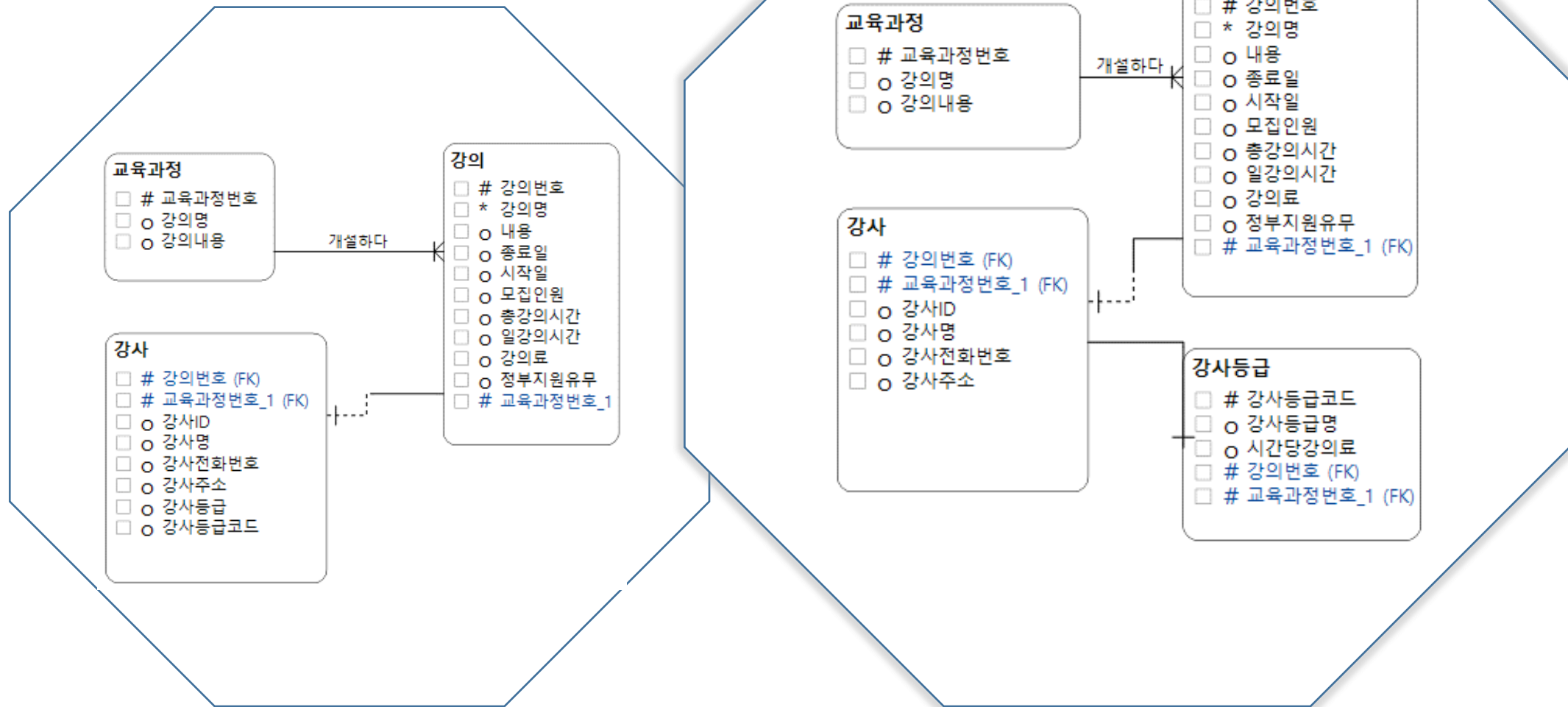
제2차 정규화



정규화(Normalization)

제3차 정규화

제 2 정규화를 충족하면서 기본키가 아닌 속성 간에는 서로 종속이 될 수 없고, 속성 간의 종속성을 배제시킨다. 즉, **이행 함수적 종속을 제거**한다.



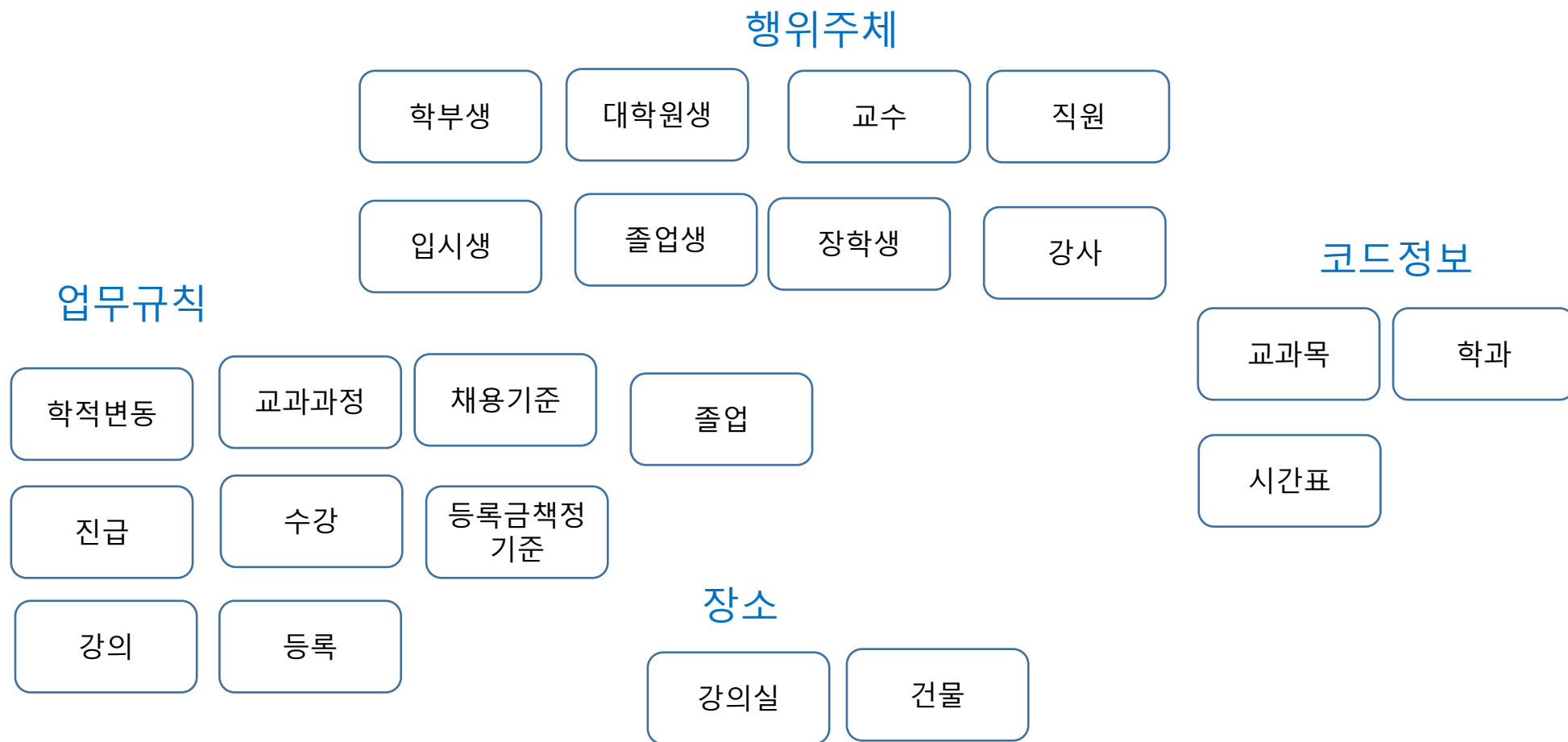
대학학사업무 예시

대	중	소	주요업무
학사	1.입시관리	원서접수	
		입시성적	
		입학사정	
		합격자관리	
	2. 학적관리	입학처리	
		학적기본	
		학적변동	휴학, 복학등의 학생신분 변동관리
	3. 교과관리	수업환경	강의실관리
		교과목	교과목마스터관리 대체과목관리 선수과목관리
		교과과정	교과과정관리 강의교수관리
		수업시간표	
		강의계획서	
		강사료관리	
	4. 수강관리	수강배정	강의실 배정
		수강신청	
	5.성적관리	성적입력	
		성적처리	

대	중	소	주요업무
학사	6. 장학관리	장학생관리	
		장학기준관리	
	7.등록관리	등록기준관리	등록금책정 기준관리
		등록대상자관리	등록금책정에 따른 대상자관리
		등록처리	은행으로부터 수신된 등록금 납부처리
	8.졸업관리	환불처리	휴학, 자퇴 등의 등록금 환불처리
		졸업대상사관리	
		졸업사정관리	졸업적격여부 판단관리
행정	1.교원인사	졸업확정관리	졸업사정에 따른 확정관리 (학위번호 부여)
		교원채용관리	
		교원인사기본	
		교원임용관리	교수임용관리
	2.직원인사	시간강사	
		직원채용관리	
		직원인사기본	
		근무성적평정	
		직원임용관리	직원 인사발령, 승진/승급 관리
		직원교육훈련	

대학학사업무 예시

엔티티 후보 도출



대학학사업무 예시

엔티티 도출-식별자 정의

학부생

☐ # 학번

교수

☐ # 교수번호

강사

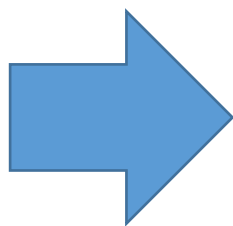
☐ ○ 강사번호

대학원생

☐ # 학번

직원

☐ # 직원번호



학생

☐ # 학번

☐ * 학생구분코드

학생구분코드

학부생

대학원생

교직원

☐ # 교직원번호

☐ ○ 교직원구분코드

교직원구분코드

교수

강사

직원

대학학사업무 예시

속성정의

엔티티명	주요속성	
학적	<ul style="list-style-type: none">신상정보(성명, 주민번호, 주소, 전화번호)학생신분정보(단과대, 학과, 학년)최종학적상태코드(재학/졸업)	
학적변동	<ul style="list-style-type: none">학적상태코드	
진급	<ul style="list-style-type: none">학년	
교직원	<ul style="list-style-type: none">신상정보(성명, 주민번호, 주소, 전화번호)임용일자퇴직일자직급	
교과목	<ul style="list-style-type: none">교과목명개설일자폐강일자주개설학과대상학년학점	
교과과정	<ul style="list-style-type: none">개설학과개설학년폐강여부폐강사유인정시수	

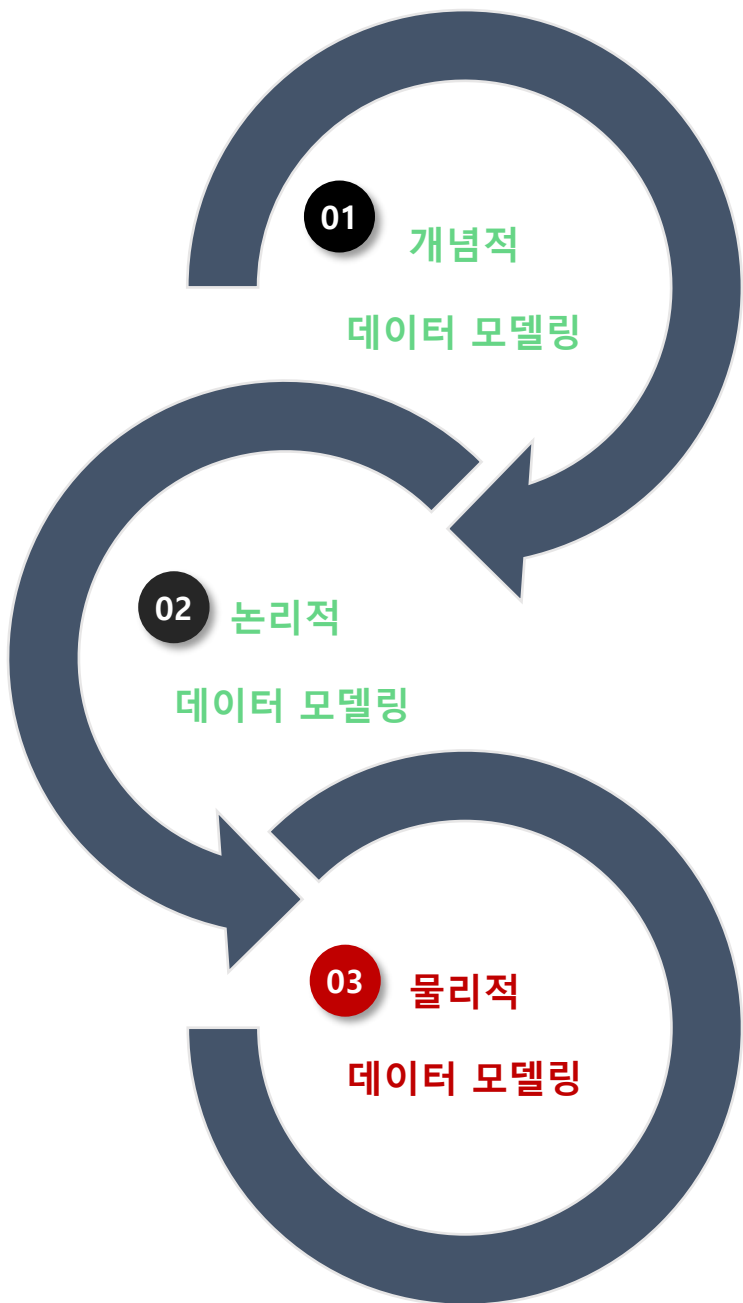
엔티티명	주요속성	
수강성적	<ul style="list-style-type: none">수강신청일자수강여부성적	
강의교수	<ul style="list-style-type: none">강의시작일자강의종료일자강의시수	
등록	<ul style="list-style-type: none">등록금수납일자	
장학코드	<ul style="list-style-type: none">장학명재원구분코드지급기준평점평균	
장학대상자	<ul style="list-style-type: none">장학금장학금지급일자	
졸업	<ul style="list-style-type: none">졸업판정여부졸업불가사유코드학위번호졸업논문제목	
강의실	<ul style="list-style-type: none">강의실명수용인원	
건물	<ul style="list-style-type: none">건물명	

대학학사업무 예시

속성정의

엔티티명	주요속성	
학적	<ul style="list-style-type: none">신상정보(성명, 주민번호, 주소, 전화번호)학생신분정보(단과대, 학과, 학년)최종학적상태코드(재학/졸업)	
학적변동	<ul style="list-style-type: none">학적상태코드	
진급	<ul style="list-style-type: none">학년	
교직원	<ul style="list-style-type: none">신상정보(성명, 주민번호, 주소, 전화번호)임용일자퇴직일자직급	
교과목	<ul style="list-style-type: none">교과목명개설일자폐강일자주개설학과대상학년학점	
교과과정	<ul style="list-style-type: none">개설학과개설학년폐강여부폐강사유인정시수	

엔티티명	주요속성	
수강성적	<ul style="list-style-type: none">수강신청일자수강여부성적	
강의교수	<ul style="list-style-type: none">강의시작일자강의종료일자강의시수	
등록	<ul style="list-style-type: none">등록금수납일자	
장학코드	<ul style="list-style-type: none">장학명재원구분코드지급기준평점평균	
장학대상자	<ul style="list-style-type: none">장학금장학금지급일자	
졸업	<ul style="list-style-type: none">졸업판정여부졸업불가사유코드학위번호졸업논문제목	
강의실	<ul style="list-style-type: none">강의실명수용인원	
건물	<ul style="list-style-type: none">건물명	

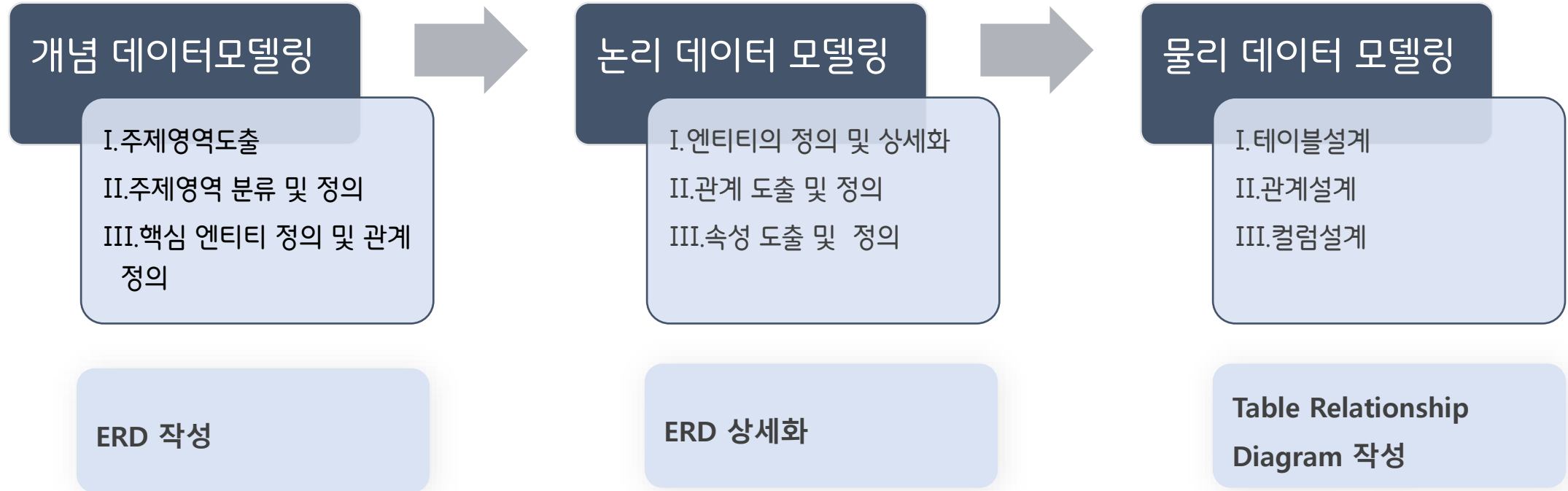


데이터 모델링 과정

3. 물리적 데이터 모델링

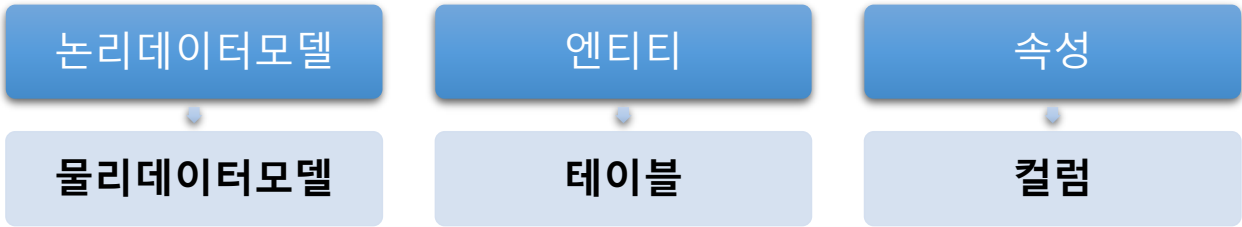


3. 물리적 데이터 모델링



3. 물리적 데이터 모델링

I. 테이블 설계



Column

Row



학번	학생명	학년	성별	학과코드
20210001	홍길동	2	M	100
20220001	이몽룡	1	M	100
20220002	성춘향	1	F	100

논리데이터모델을 물리데이터모델로 전환 시 고려사항

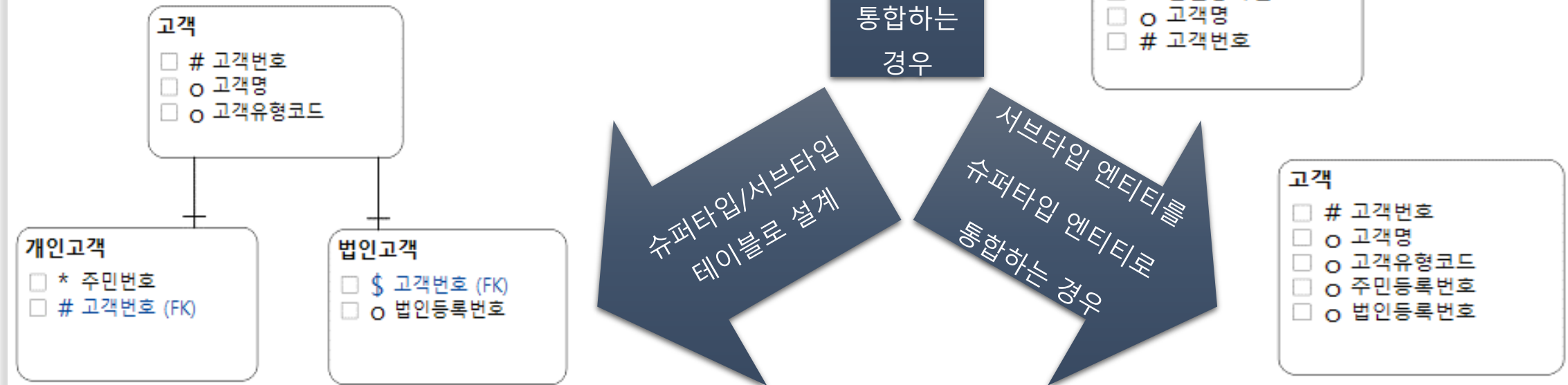
- ▶ Super/Sub 관계의 엔티티를 몇 개의 테이블로 만들 것인가
- ▶ 배타적 관계 엔티티의 외부키를 몇 개로 할 것인가
- ▶ 성능 향상을 위해 테이블을 추가 또는 통합할 것인가
- ▶ 통계 작업을 위해 집계 테이블 등의 임시 테이블은 몇 개로 할 것인가
- ▶ 테이블 컬럼을 다른 테이블에 중복할 것인가, 중복 한다면 어떤 애플리케이션이 관련되어 있는가, 인덱스의 생성이 필요한가
- ▶ 분산환경에서 테이블을 중복할 것인가
- ▶ 데이터가 분산 환경에서 이동 시 문제를 어떻게 해결할 것인가

3. 물리적 데이터 모델링

I. 테이블 설계

테이블

- 테이블 명명 규칙에 따라 부여 : 주제 영역 명과 엔티티 명 조합
- 슈퍼/서브타입 엔티티의 경우 공통 속성이 많을 경우 하나의 테이블로 설계, 슈퍼타입 엔티티에서 관리하는 속성이 적거나 응용프로그램에서 슈퍼타입 엔티티를 사용하는 횟수가 적을 때는 서브타입 쪽으로 슈퍼타입 엔티티를 통합하여 설계



3. 물리적 데이터 모델링

I. 테이블 설계 : 반정규화

- **반정규화란** 정규화된 데이터 모델을 통합, 중복, 분리하는 과정으로, 의도적으로 정규화 원칙을 위배하는 행위
- 반정규화는 의도적으로 중복을 만들어 검색(Read) 성능은 향상되지만, 입력(Create), 수정(Update), 삭제>Delete)의 성능은 낮아짐.
- 반정규화를 위해서는 사전에 데이터의 일관성과 무결성을 우선으로 할지, 데이터베이스의 성능과 단순화를 우선으로 할지를 결정해야 함.
- 반정규화 방법: 테이블 통합, 테이블 분할, 중복 테이블 추가, 중복 속성 추가, 관계 중복 등.

반정규화 대상조사



- ☐ 범위처리 빈도수 조사
- ☐ 대량 범위 처리 조사
- ☐ 통계성 프로세스 조사
- ☐ 테이블 조인 개수

다른 방법 유도 검토



- ☐ 뷰 테이블
- ☐ 클러스터링 적용
- ☐ 인덱스의 조정
- ☐ 응용애플리케이션

반정규화 적용



- ☐ 테이블의 반정규화
- ☐ 속성의 반정규화
- ☐ 관계의 반정규화

3. 물리적 데이터 모델링

1. 테이블 설계 : 테이블 반정규화

기법분류	기법	내용
테이블통합	1:1관계	1:1 관계 테이블을 하나의 테이블로 통합
	1:M	1:M 관계 테이블을 하나의 테이블로 통합
	슈퍼/서브타입	슈퍼/서브 타입의 테이블을 하나의 테이블로 통합
테이블분할	수직분할	칼럼단위의 테이블을 디스크 I/O를 분산처리하기 위해 테이블을 1:1로 분리하여 성능향상
	수평분할	로우단위로 집중 발생하는 트랜잭션을 분석하여 디스크 I/O및 데이터 접근의 효율성을 높여 성능을 향상하기 위해 로우단위로 테이블을 쪼갬(Relation없음)
테이블추가	중복테이블추가	다른 업무이거나 서버가 다른 경우 동일한 테이블 구조를 중복하여 원격조인을 제거하여 성능향상
	통계테이블추가	SUM, AVG 등을 미리 수행하여 계산해 둠으로써 조회 시 성능을 향상
	이력테이블추가	이력테이블 중에서 마스터 테이블에 존재하는 레코드를 중복하여 이력테이블에 존재하는 방법
	부분테이블추가	하나의 테이블의 전체 칼럼 중에서 집중화 되는 컬럼들이 있을 때, 성능을 향상시키기 위해 별도의 반정규화된 테이블 생성

3. 물리적 데이터 모델링

I. 테이블 설계 : 테이블 통합

테이블 통합

- 테이블 통합은 두 개의 테이블이 조인되는 경우가 많아 하나의 테이블로 합쳐 사용하는 것이 성능 향상에 도움이 될 경우 수행한다.
- 두 개의 테이블에서 발생하는 프로세스가 동일하게 자주 처리되는 경우, 두 개의 테이블을 이용하여 항상 조회를 수행하는 경우 테이블 통합을 고려해 본다.
- 테이블 통합의 종류에는 1:1 관계 테이블 통합, 1:N 관계 테이블 통합, 슈퍼타입/서브타입 테이블 통합이 있다.
- **테이블 통합 시 고려 사항**
 - ✓ 데이터 검색은 간편하지만 레코드 증가로 인해 처리량이 증가
 - ✓ 테이블 통합으로 인해 입력, 수정, 삭제 규칙이 복잡해질 수 있음
 - ✓ Not Null, Default, Check 등의 제약조건을 설계하기 어려움



- I. **1:1관계의 경우**, 병합했을 때 이행적 종속이 발생하더라도, 자주 변경(Update)되는 정보가 아니며 같이 조회(Read)되는 경우가 많다면 통합한다.
- II. **1:M 관계** 일지라도 항상 조인이 발생하는 경우 1쪽 데이터의 중복을 무릅쓰고 병합할 수 있다
- III. **슈퍼타입과 서브타입 관계로** 연결되어 있을 경우 필요에 따라 병합 할 수 있다.

3. 물리적 데이터 모델링

1. 테이블 설계 : 테이블 분할/중복테이블 추가

수평적 분할

테이블의 수평적 분할(Horizontal Partitioning)

- 천만 개의 데이터를 백만 개 데이터 10개로 자름.
- 수평 분할(horizontal partitioning)은 하나의 테이블의 각 행을 다른 테이블에 분산시키는 것.
- 예) 고객테이블을 성별에 따라 '남녀'로 나누어 CustomerMen과 CustomerWomen 두 개의 테이블로 분할. 테이블은 2개로 분할되지만, 모든 고객을 나타내기 위해 양자를 결합한 뷰를 생성

중복 테이블 추가

중복테이블의 추가

통계관리,이력관리 등 데이터의 효율적인 이용을 위하여 부가적인 정보가 필요한 경우 별도의 테이블로 만들 수 있다



수직적 분할

테이블의 수직적 분할(Vertical Partitioning)

- 트랜잭션이 집중 발생하는 속성들을 따로 뽑아서 분할.
- 테이블의 일부 열을 빼내는 형태로 분할한다. 관계의 정규화는 본질적으로 수직 분할에 관련된 과정이다. 수직 분할에는 여러 저장 장치를 사용할 수도 있다. 예를 들어 별로 사용되지 않는 열이나 매우 자릿수가 많은 열을 다른 장치에 두는 등의 조치를 생각할 수 있다. 분할된 테이블들을 포함하는 뷰를 생성하면 원래의 경우보다 성능이 저하되지만, 사용 빈도가 높은 데이터에만 액세스할 경우 성능이 향상된다.
- 예) 최근 1개월 뉴스만 모아 서비스

3. 물리적 데이터 모델링

1. 테이블 설계 : 중복테이블 추가



중복 테이블 추가란?

많은 양의 정보들을 자주 Group By, Sum 등과 같은 집계 함수를 이용해서 실시간으로 통계 정보들을 계산해 낼 수 있다. 하지만 대부분 이러한 계산의 유형은 매우 많은 양의 데이터가 대상이 되고, 하나의 테이블이 아닌 여러 개의 테이블에서 필요한 데이터를 추출하는 경우가 대부분이다. 이를 위해서 특정 통계 테이블을 두거나 중복 테이블을 추가할 수 있다.

중복테이블 추가 판단?

- ① 정규화에 충실하면 종속성, 활용성은 향상되나 수행 속도 증가가 발생하는 경우 고려한다.
- ② 많은 범위를 자주 처리해야 하는 경우에 고려한다.
- ③ 특정 범위의 데이터만 자주 처리되는 경우에 고려한다.
- ④ 처리 범위를 줄이지 않고는 수행 속도를 개선할 수 없는 경우에 고려한다.
- ⑤ 요약 자료만 주로 요구되는 경우에 고려한다.
- ⑥ 추가된 테이블의 처리를 위한 오버헤드를 고려하여 결정한다.
- ⑦ 인덱스의 조정이나 부분 범위 처리로 유도, 클러스터링을 이용하여 해결할 수 있는지를 철저히 검토한 후 결정한다.

3. 물리적 데이터 모델링

II. 관계 설계

기본적인 관계(1:1 또는 1:M)

- 부모 테이블의 PK를 자식 테이블에서 PK로 상속 받거나 일반 컬럼으로 상속

상호배타적인 관계(Exclusive-OR)

- 일반적으로 컬럼을 통합하여 설계
- 자식 테이블에서 상호 배타적 관계를 설계하는 방법은 참조 컬럼을 한 컬럼으로 통합하여 설계하는 방법과 개별 컬럼으로 분리하는 방법
- 부모테이블의 경우엔 테이블을 통합하거나 PK 구조를 통일하는 방법 사용.

재귀적 관계

- DBMS에서 순환관계를 처리하기 어렵다면 PK를 계층구조를 가지는 코드 형태로 설계

관계 Optionality 확정

- 논리모델링 단계에서 정확하지 않거나 모호한 Optionality를 검토하여 확정
- 부모테이블과 자식테이블 Mandatory인 경우엔 테이블에 데이터가 입력되는 시점이 동시에 입력되는 경우임.
- 자식테이블이 Optional인 경우엔 부모 테이블의 PK에 해당하는 데이터가 자식 테이블에 존재하지 않아도 되며, 자식테이블의 데이터가 입력될 때 부모 테이블에 해당데이터가 이미 존재해야 함.
- 부모, 자식 모두 Optional 인 경우는 부모 테이블이 존재하지 않는 상태에서 자식 테이블의 데이터가 발생하는 경우

3. 물리적 데이터 모델링

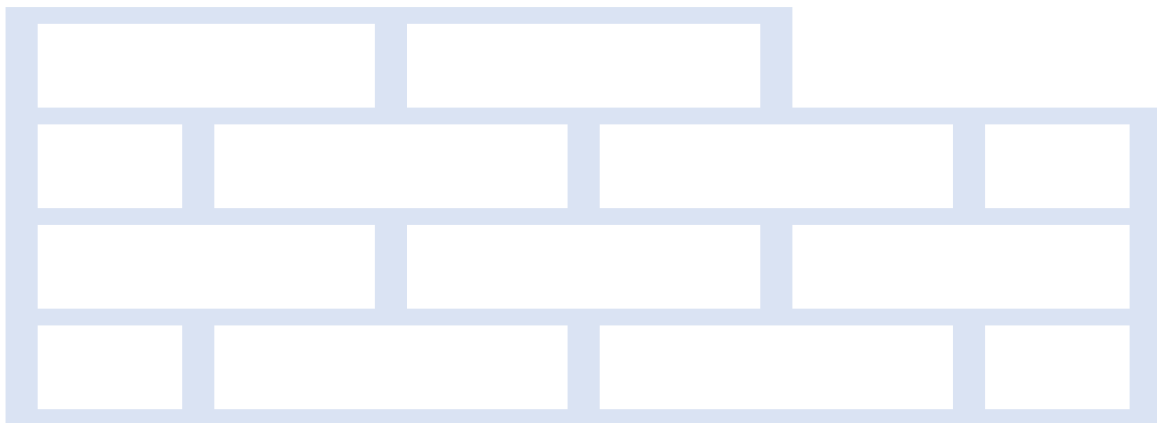
III. 컬럼 설계

속성-컬럼 변환

- 컬럼변환은 표준화를 적용하여 컬럼명으로 변환, 데이터타입과 길이, PK조정
- 대부분의 논리모델 속성을 테이블 컬럼으로 1:1 변환

컬럼의 반정규화

- 중복 컬럼 추가는 조인해서 데이터를 처리할 때 데이터를 조회하는 경로를 단축하기 위해 자주 사용하는 속성을 하나 더 추가하는 것이다. 중복 속성을 추가하면 데이터의 무결성 확보가 어렵고, 디스크 공간이 추가로 필요함.



3. 물리적 데이터 모델링

III. 컬럼 설계 : 컬럼 반정규화

컬럼반정규화기법	내용
중복칼럼추가	조인을 감소시켜 성능을 향상시키기 위해 컬럼을 추가
파생컬럼추가 (Derived Column)	트랜잭션이 처리되는 시점에 계산에 의해 발생하는 성능저하를 예방하기 위해 미리 값을 계산하여 컬럼에 보관
이력테이블 칼럼추가	대량의 이력데이터를 처리할 때 불특정 일자를 조회할 때 나타날 수 있는 성능저하를 방지하기 위해 기능성 칼럼(최근값여부, 시작/종료일자 등) 추가
PK에 의한 컬럼추가	복합의미를 갖는 PK를 단일 속성으로 구성하였을 경우 발생할 수 있는 성능 저하를 예방하기 위해 PK안에 데이터를 쪼개어 일반 컬럼으로 추가.
응용시스템 오작동을 위한 컬럼 추가	업무적 의미는 없지만 사용자가 데이터처리를 잘못하여 원래 값으로 복구하고자 할 때 이전 데이터를 임시적으로 중복보관할 수 있도록 컬럼 추가.

3. 물리적 데이터 모델링

III. 컬럼 설계 : 컬럼 반정규화

중복 컬럼 생성 조건

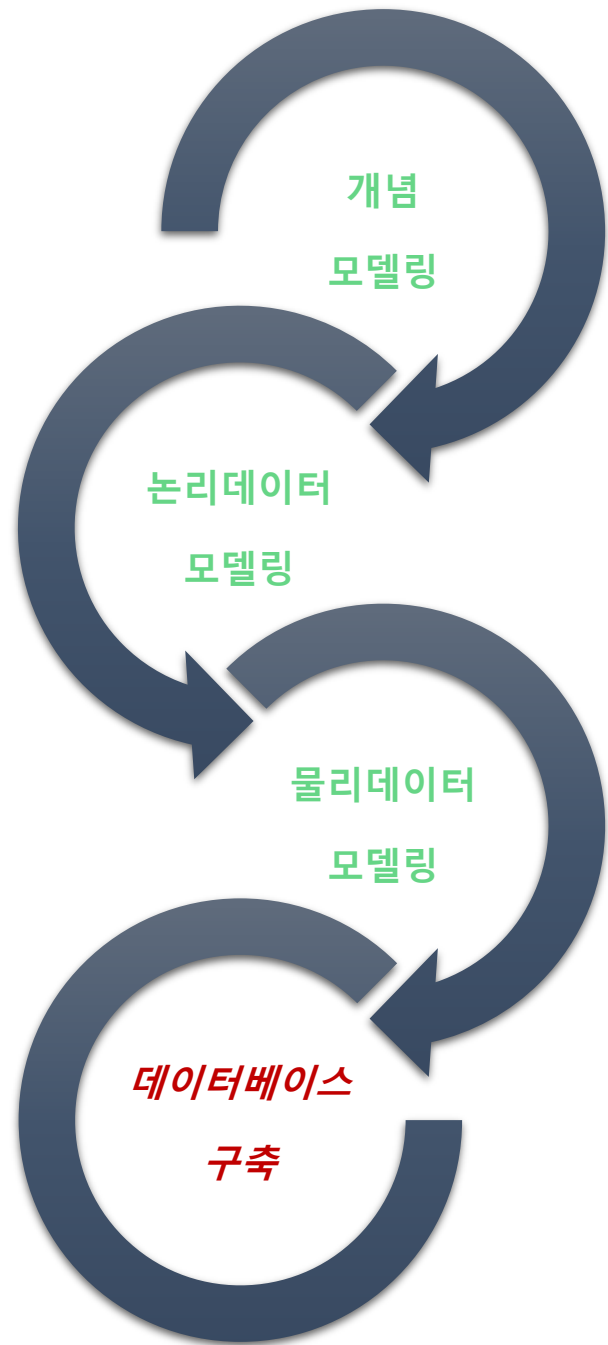
- 빈번하게 조인을 일으키는 칼럼에 대해서 고려
- 조인의 범위가 다량인 경우를 온라인화해야 하는 경우처럼 속도가 중요한 칼럼에 대해서는 중복 칼럼을 고려.
- 액세스의 조건으로 자주 사용되는 칼럼에 대해서 고려.
- 자주 사용되는 액세스 조건이 다른 테이블에 분산되어 있어 상세한 조건 부여에도 불구하고 액세스 범위를 줄이지 못하는 경우에 자주 사용되는 조건들을 하나의 테이블로 모아서 조건의 변별성 을 극대화함.
- 복사된 칼럼의 도메인은 원본 칼럼과 동일하게 해야 함(데이터의 일관성유지)
- 접근 경로의 단축을 위해서 부모 테이블의 칼럼을 자식 테이블에 중복가능
- 상위 레벨의 테이블에 집계된 칼럼을 추가(M:1 관계) : 집계 컬럼 추가
- 연산된 결과를 주로 사용하는 경우에도 미리 연산을 하여 중복 칼럼 생성
- 기본키의 칼럼이 길거나 여러 개의 칼럼으로 구성되어 있는 경우 인위적인 기본키 추가

3. 물리적 데이터 모델링

III. 컬럼 설계 : 컬럼 반정규화

중복 컬럼 생성시 유의사항

- ✓ 다중 테이블 클러스터링으로 해결할 수 있는지 검토한다.
- ✓ SQL GROUP 함수 이용하여 처리할 수 있는지 검토한다.
- ✓ 저장 공간의 지나친 낭비를 고려하여 적절한 대비책을 마련해야 한다.
- ✓ 반복 칼럼은 특별한 경우를 제외하고는 절대 사용할 필요가 없고, 있다면 sum(decode..) 용법과 같은 SQL 기법 등을 활용하여 이러한 부분을 피할 수 있도록 한다.
- ✓ 경우에 따라 상대 테이블의 ROWID를 복사하는 경우가 효과적일 때도 있다.
- ✓ 데이터의 일관성 보장에 유의해야 한다. 성능을 향상시키기 위해서 데이터의 일관성을 그르치는 일이 일어나서는 안된다.
- ✓ 칼럼의 중복이 지나치게 심하면 데이터 처리의 오버헤드가 발생하게 된다.
- ✓ 사용자나 프로그램은 반드시 원본 칼럼만 수정하는 것이 바람직하다.
- ✓ 일반적으로 수행 속도를 우려해서 지나치게 많은 중복 칼럼을 생성하고 있는 것이 현실이다. 가능하 면 중복 칼럼을 적게 가져가는 것이 바람직하다.
- ✓ 클러스터링, 결합 인덱스, 적절한 SQL을 이용하면 특별한 경우를 제외하고는 거의 해결 가능하기 때문에 이 부분을 먼저 적극적으로 고려해 보는 것이 바람직하다.
- ✓ 중복 칼럼을 이용하면 손쉽게 액세스 효율을 개선할 수 있으나 지나친 중복화는 반드시 데이터 일관성 오류 발생의 개연성 증가 및 데이터 처리 오버헤드 증가라는 반대급부가 있다는 것을 염두 에 두고 수행해야 한다.
- ✓ JOIN, SUB-QUERY 액세스 경로의 최적화 방안을 보다 철저히 강구해야 한다.



데이터 모델링 과정

4. 데이터베이스 구축



3. 데이터베이스 구축

데이터베이스

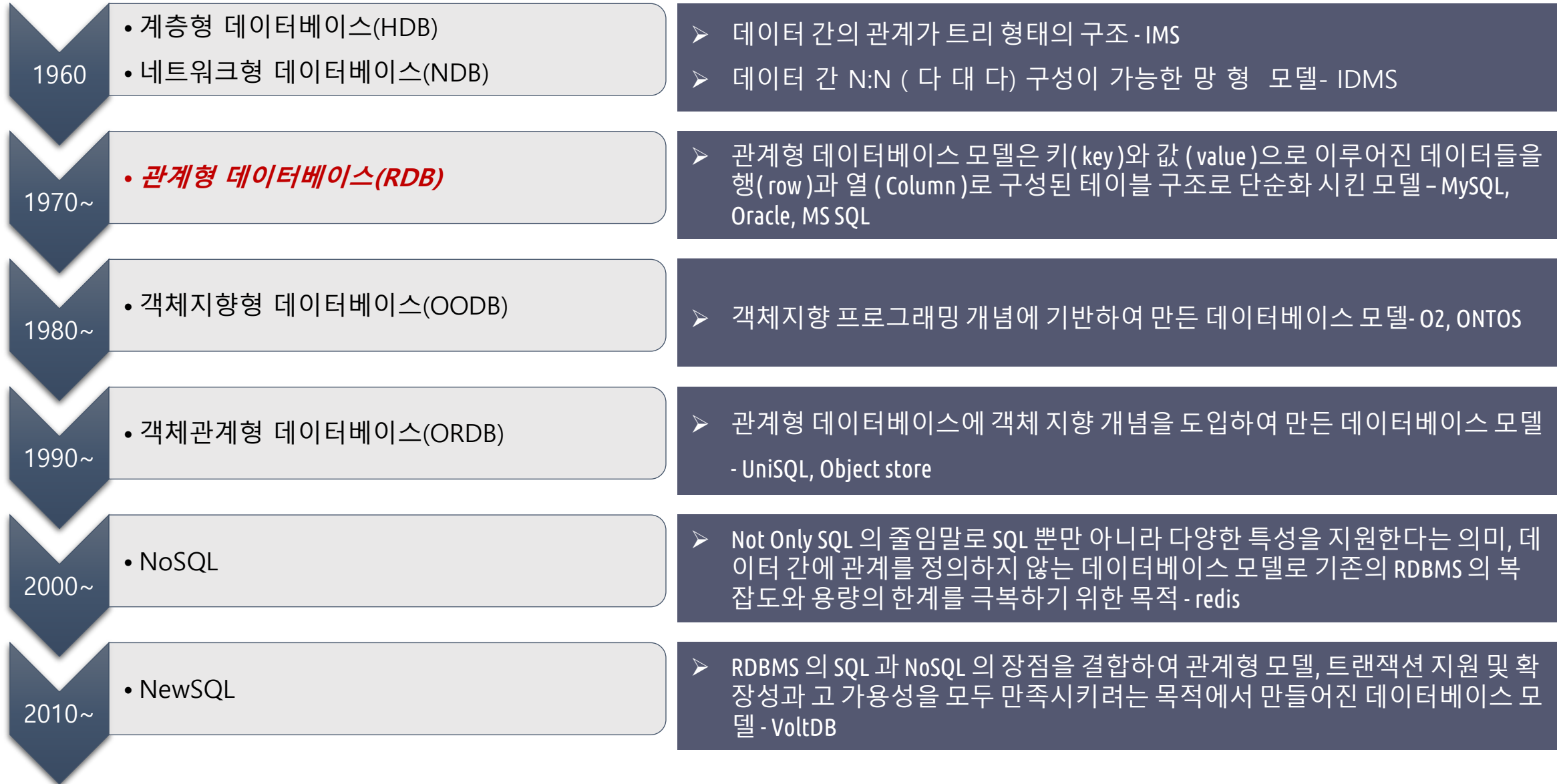
- 데이터베이스란 여러 사람들이 공유하고 사용할 목적으로 통합 관리되는 데이터들의 모임
- 등산할 때 기반이 되는 기지를 베이스캠프라 하듯이 데이터베이스라는 용어도 1950년대 미국에서 데이터의 기지라는 뜻에서 데이터베이스라는 용어를 처음 사용

DBMS

- 다수의 사용자들이 데이터베이스 내의 데이터를 접근할 수 있도록 해주는 소프트웨어들
- DBMS 특징
 - ✓ 데이터의 독립성
 - ✓ 데이터의 무결성
 - ✓ 데이터의 보안성
 - ✓ 데이터의 일관성
 - ✓ 데이터 중복의 최소화

3. 데이터베이스 구축

Database 종류



3. 데이터베이스 구축

SQL

- 특징
 - ✓ 트랜잭션(전부 아니면 무): 전체 트랜잭션이 하나의 단위로 기록. 실패 시 전체 롤백
 - ✓ 정규화: DB설계 시 중복을 최소화해서 구조화 하는 프로세스
- 장점
 - ✓ 데이터의 성능이 일반적으로 좋아 정렬, 탐색, 분류가 빠름
 - ✓ 신뢰성이 높아 데이터의 무결성을 보장
 - ✓ 정규화에 따른 갱신 비용을 최소화
- 단점
 - ✓ 기존에 작성된 스키마를 수정하기 어려움
 - ✓ 데이터베이스의 부하를 분석하기 어려움
 - ✓ 빅데이터를 처리하는데 매우 비효율적임.

NoSQL

- 특징
 - ✓ 거대한 Map으로서 key-value 형식을 지원함.
 - ✓ 관계형 DB와 달리 PK, FK JOIN 등 관계를 정의하지 않음.
 - ✓ 스키마에 대한 정의가 없다.
- 장점
 - ✓ 대용량 데이터 처리를 하는데 효율적임.
 - ✓ 읽기 작업보다 쓰기 작업이 더 빠르고 관계형 데이터베이스에 비해 쓰기와 읽기 성능이 빠름.
 - ✓ 데이터 모델링이 유연함.
 - ✓ 뛰어난 확장성으로 검색에 유리함.
 - ✓ 최적화된 키 값 저장 기법을 사용하여 응답속도나 처리효율 등에서 성능이 뛰어남.
 - ✓ 복잡한 데이터 구조를 표현할 수 있음.
- 단점
 - ✓ 쿼리 처리시 데이터를 파싱 후 연산을 해야 해서 큰 크기의 document를 다룰 때는 성능이 저하됨

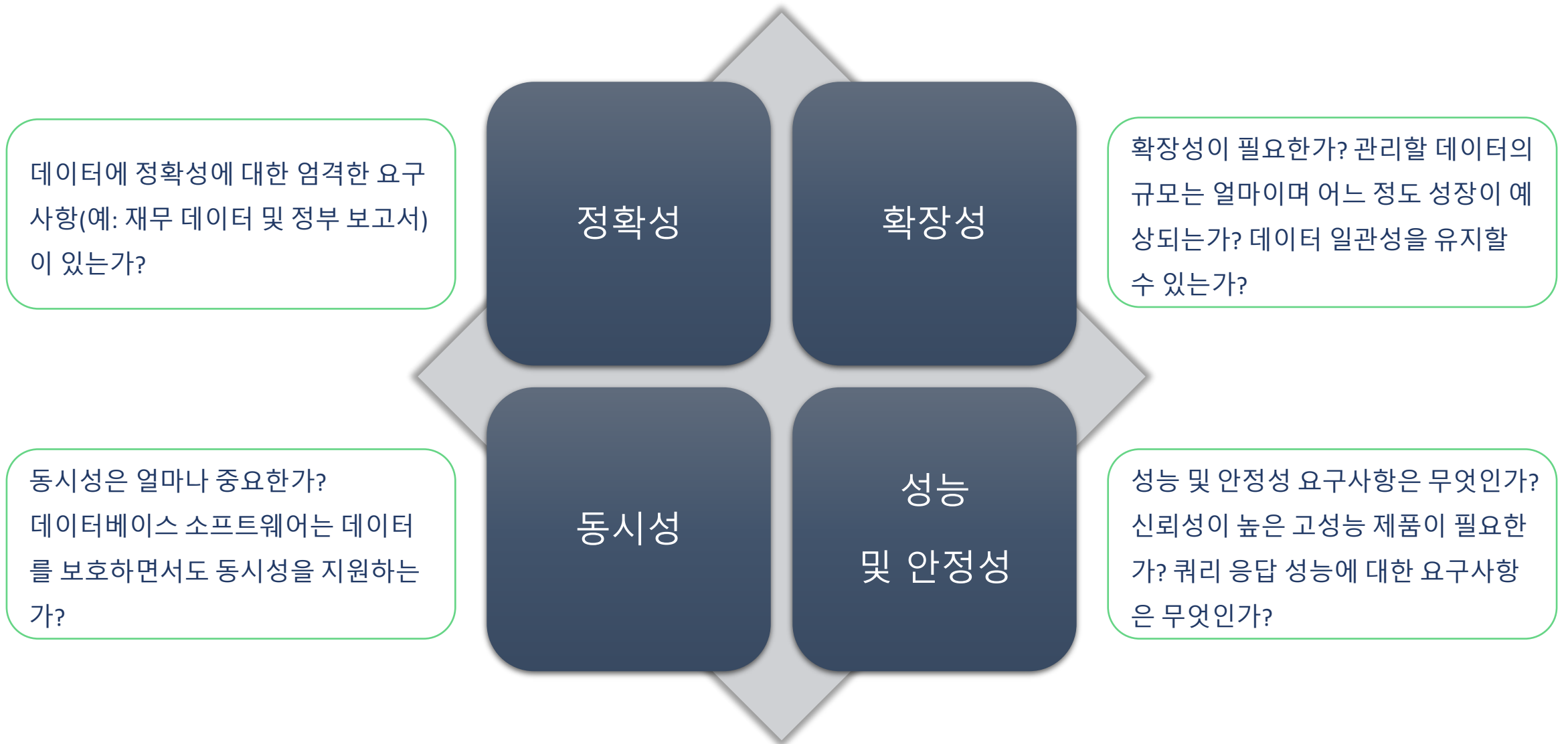
3. 데이터베이스 구축

Relational Database

	원자성(Atomicity)	완전한 데이터베이스 트랜잭션을 구성하는 모든 요소를 정의
	일관성(Consistency)	트랜잭션 후 데이터 포인트를 올바른 상태로 유지하기 위한 규칙을 정의
	고립성(Isolation)	혼란을 피하기 위해 트랜잭션이 커밋될 때까지 트랜잭션의 효과가 다른 사람에게 보이지 않게 함
	내구성(Durability)	트랜잭션이 커밋되면 데이터 변경 사항이 영구적이 되도록 함

3. 데이터베이스 구축

관계형 데이터베이스를 선택 시 고려사항





END

- 김정연
- jkimpool@naver.com
- 010-6214-6127

개념적 데이터모델링

CONTENTS A

컨텐츠에 대한 내용을 적어요

Enjoy your stylish business and campus life with
BIZCAM

CONTENTS A

컨텐츠에 대한 내용을 적어요

Enjoy your stylish business and campus life
with BIZCAM

CONTENTS A

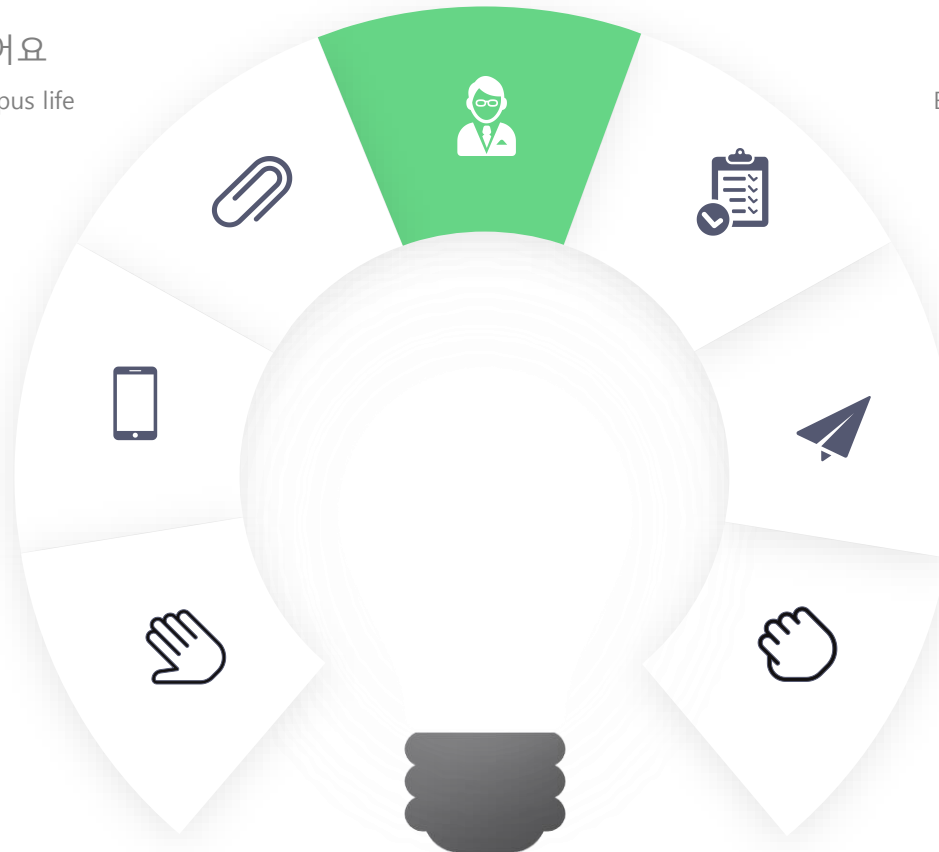
컨텐츠에 대한 내용을 적어요

Enjoy your stylish business and campus life
with BIZCAM

CONTENTS A

컨텐츠에 대한 내용을 적어요

Enjoy your stylish business and campus life
with BIZCAM



CONTENTS A

컨텐츠에 대한 내용을 적어요

Enjoy your stylish business and campus life
with BIZCAM

CONTENTS A

컨텐츠에 대한 내용을 적어요

Enjoy your stylish business and campus life
with BIZCAM

CONTENTS A

컨텐츠에 대한 내용을 적어요

Enjoy your stylish business and campus life
with BIZCAM

PPT PRESENTATION

Enjoy your stylish business and campus life with BIZCAM

