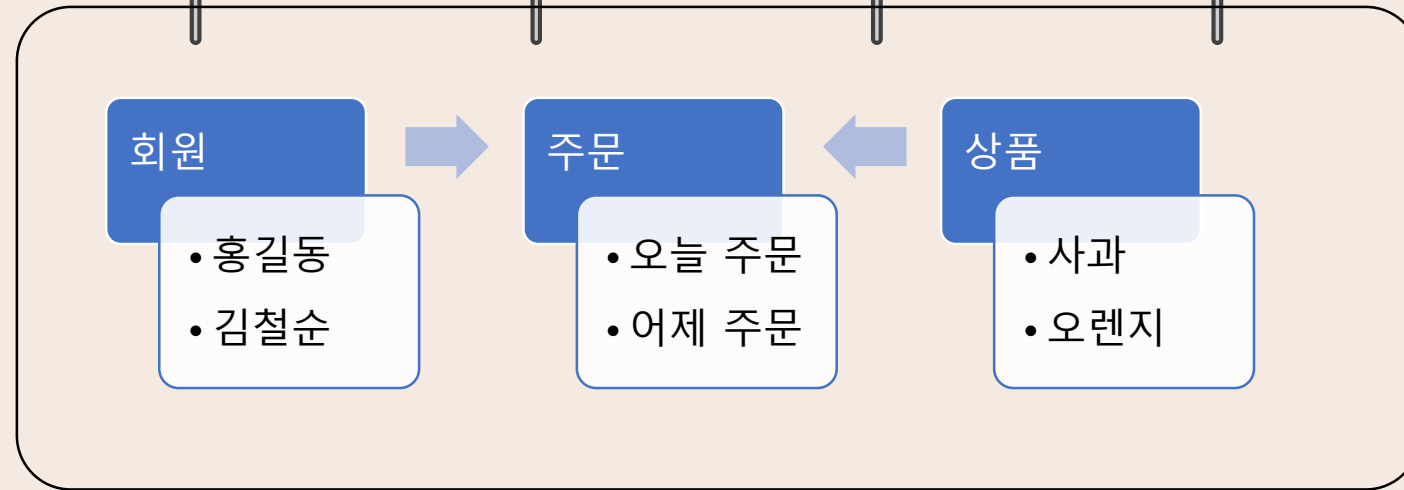


Data Modeling

데이터 모델링 이론



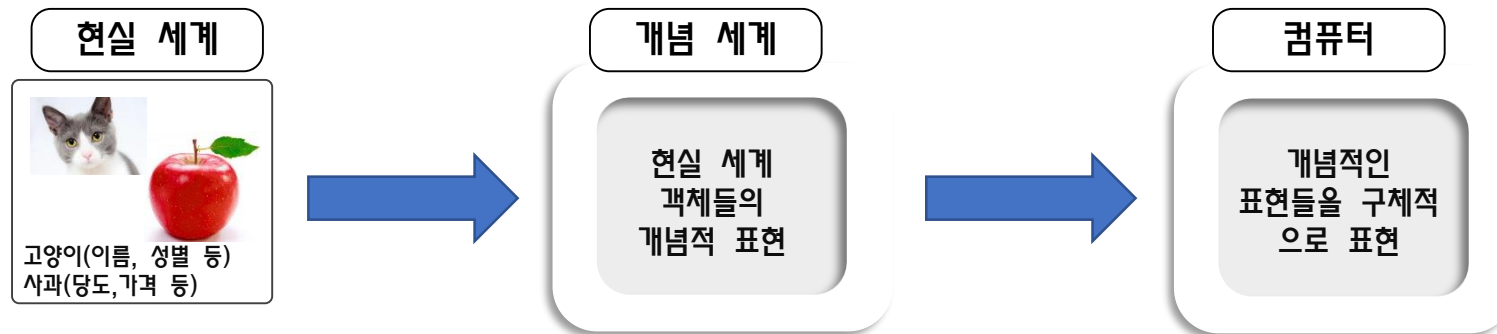
≡ 데이터 모델링이란?

≡ 모델링의 정의

업무에서 사용하는 데이터들을 Database에 기록/관리/활용하기 위하여 구조화하고, 형상화 시키는 일련의 과정을 의미합니다.

인간의 모든 활동이 모델링의 대상이 될 수 있습니다.

현실 세계 → 문서 → 설계 → 구현 → DataBase화



사용자의 **요구 사항**을 분석한 다음 이를 **문서화**하는 일련의 과정입니다.

업무 분석/설계 단계에서 **적절한 표기법(ERD)으로 표현**하는 방법입니다.

데이터들의 사용에 대하여 구조화/조직화하는 단계입니다.

실체(Entity)와 관계(Relation)에 의하여 문서를 표현하는 방식입니다.

≡ 데이터 모델링이란?

≡ data modeling

업무에서 사용하는 데이터들을 Database에 기록/관리/활용하기 위하여 구조화하고, 형상화 시키는 일련의 과정을 의미합니다.

모델링은 업무를 파악하여 개념들을 정리/분류하여 Entity, Attribute, Relation으로 형상화합니다.

정보 모델링

유형 또는 무형의 개체들을 사람이 이해할 수 있는 추상적 개념의 정보 구조로 표현하는 것
추상적 개념이란 속성들의 집합인 개체 타입으로 표현하는 것을 의미합니다.
다음 그림은 회원, 장바구니, 상품 등을 객체 타입으로 표현한 것입니다.

member(회원)
member_id(PK) name email password address role

cart(장바구니)
cart_id(PK) member_id(FK)

item(상품)
item_id(PK) item_nm price stock_number item_detail item_sell_status

데이터 모델링

개념 세계의 정보 구조를 논리적 구조(DBMS가 이해 가능한 구조)로 변환시키는 과정입니다.
특정 DBMS와 무관하게 독립적으로 적용됩니다.

데이터 구조화

테이블 구조 설계에 해당합니다.
논리적 구조를 물리적 구조로 변환 시키는 일입니다.

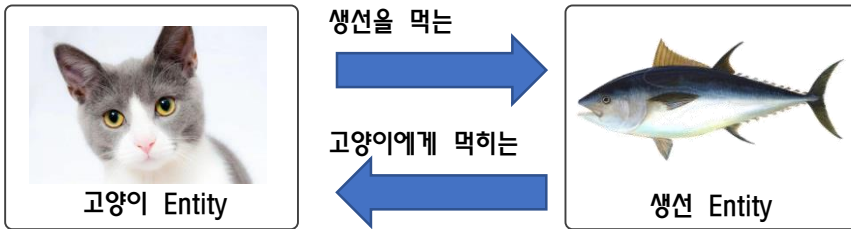
≡ 데이터 모델링이란?

≡ 데이터 모델

데이터 모델은 개념적 데이터 모델과 논리적 데이터 모델이 있습니다.

개념적 데이터 모델

개체의 **타입**과 **속성** 및 개체들간의 **관계**를 설정하여 모델을 표현하는 방법입니다.
일반적으로 개체-관계 모델(Entity-Relation Model)이라고 합니다.



논리적 데이터 모델

데이터 베이스 관리 시스템(DBMS)에 따라 계층형(Hierarchical) 데이터 모델, 네트워크형(Network) 데이터 모델, 관계형(Relation) 데이터 모델 등을 사용하여 표현합니다.

아이디	이름
① 계층형 데이터 모델	트리 구조를 이용한 데이터 모델
② 망(Network)형 데이터 모델	그래프 형태를 기반으로 데이터 표현 계층형 모델의 보완
③ 관계형 데이터 모델	개체와 개체간의 관계를 2차원 테이블(Relation)으로 표현하는 데이터 모델
④ 객체 지향 데이터 모델	객체 지향의 개념을 도입 주요 핵심 개념 : 클래스/객체/추상화/상속/캡슐화

≡ 관계형 데이터 모델 이론

≡ 관계형 데이터 모델의 구조

현재 업계에서는 관계형 DBMS가 가장 많이 사용되고 있습니다.

관계형 데이터 모델

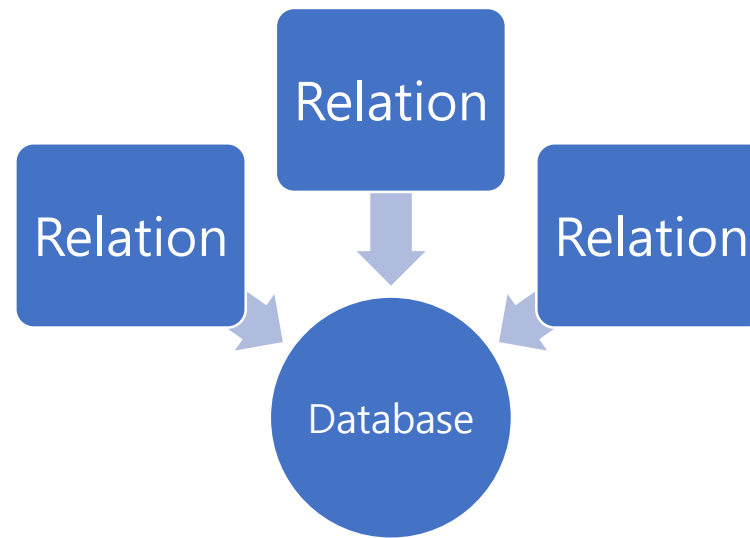
파일 시스템의 단점을 극복하고자, 1970년 E.F.Codd 박사가 발표하였습니다.

수학적 이론을 바탕으로 하여 데이터들을 2차원 형태의 테이블 형태로 표현하는 기법입니다.

즉, 테이블 형식을 이용하여 데이터를 정의하고 설명한 모델입니다.

아이디	이름	주소	성별	입사일자
kim9	김구	용산	남자	70/08/25
soon	유관순	마포	여자	80/07/17
lee	이순신	영등포	남자	85/08/15
maria	조마리아	금천	여자	90/12/25

Relation(테이블)



≡ 관계형 데이터 모델 이론

≡ 관계형 데이터 모델의 구조

Relation(테이블)에 대하여 좀 더 상세히 살펴 보겠습니다.

여러 개의 Relation이 모이면 하나의 Database가 됩니다.

Relation 예시

사원 Relation

속성(attribute) →							
schema →	아이디	이름	주민번호	주소	성별	이메일	입사일자
instance →	kim9	김구	700811-1710789	용산	남자	aaa@naver.com	70/08/25
	soon	유관순	800815-2026456	마포	여자	bbb@naver.com	80/07/17
	lee	이순신	900717-1745678	강남	남자	ccc@daum.net	85/08/15
	maria	조마리아	000811-4665432	금천	여자	ddd@daum.net	90/12/25

← tuple

용어 정리

기본 키 ↑

- 릴레이션 : 1개의 개체 집합체(예시 : 사원, 부서, 상품, 주문 등등)
- 데이터베이스 = 릴레이션(테이블)의 집합
- 행(Row) : Tuple(튜플)이라고 하며, 각 튜플은 1개의 인스턴스가 됩니다.
- 속성(Attribute) : 각 열(Column)에 부여된 이름
- 차수(Degree) : 속성(열)의 개수를 의미합니다.
- 도메인(Domain) : 속성 값이 취할 수 있는 값의 집합입니다.
 - (ex) 성별 속성의 도메인은 '남/여' 중에 한 개입니다.
- 기본키(primary key) : 튜플을 구분 짓는 고유 식별자 성격의 속성(Attribute)을 의미합니다.

모델링과 설계 매핑

항목	설명
테이블	Entity(엔티티)
Row(행)	Instance(인스턴스)
Column(열)	Attribute(속성)
Foreign Key	Relation(관계)

관계형 데이터 모델 이론

관계형 데이터 모델의 제약

관계형 모델의 키(key)에 대하여 살펴 봅니다.

키(Key)의 정의

데이터베이스 내에서 하나 이상의 속성(열, 컬럼)들로 구성하여, 릴레이션의 튜플(행, 레코드)을 식별할 수 있는 식별자를 의미합니다.

식별자는 릴레이션간의 관계를 올바르게 유지하기 위하여 반드시 필요한 개념입니다.

키는 데이터베이스의 무결성(integrity)을 유지하고 데이터를 효율적으로 관리하기 위해 필수적인 역할을 합니다.

키가 가져야 할 주요 특징은 다음과 같습니다.

항목	설명
유일성 (Uniqueness)	키 값은 각 레코드마다 고유해야 합니다. 즉, 동일한 키 값이 두 개 이상의 레코드에서 나타나서는 안 됩니다. 이를 통해 각 레코드를 식별하고 찾을 수 있습니다.
최소성 (Minimality)	키 값은 최소한의 속성(attribute)으로 구성되어야 합니다. 필요한 최소한의 정보로 레코드를 식별할 수 있어야 합니다. 불필요한 속성을 포함하면 데이터의 중복과 복잡성이 증가할 수 있습니다.
불변성 (Immutability)	키 값은 변경되지 않아야 합니다. 키 값이 변경되면 해당 레코드의 식별성이 손실되어 데이터의 일관성을 해치게 됩니다.
영구성 (Permanence)	키 값은 레코드의 생명 주기 동안 계속 유지되어야 합니다. 레코드가 삭제되더라도 키 값은 다른 레코드에서 재사용되지 않아야 합니다.

유일성과 최소성

예를 들어서 id와 email을 합치면 유일성(Uniqueness)을 보장 받을 수 있습니다.

하지만 최소성(Minimality) 관점에서 id만 있으면 데이터를 식별하는 데 문제가 없습니다.

≡ 관계형 데이터 모델 이론

≡ 관계형 데이터 모델의 제약

관계형 모델의 키(key)에 대하여 살펴 봅니다.

항목	설명
슈퍼키 (Super Key)	튜플을 고유하게 식별할 수 있는 속성들의 집합을 의미합니다. 사원 릴레이션에서 {아이디}, {주민 번호}, {이메일}, {아이디, 주민 번호}, {아이디, 이메일}, {이메일, 주민번호}, {아이디, 주민번호, 이메일} 등은 모두 슈퍼키에 해당합니다.
후보키 (Candidate Key)	슈퍼 키중에서 유일성 과 최소성 을 만족하는 속성 집합을 의미합니다. 사원 릴레이션에서 {아이디}와 {주민 번호} 및 {이메일} 등은 모두 후보키가 될 수 있습니다. 슈퍼 키는 모든 후보키를 포함하는 관계입니다.
기본키 (primary key)	여러 개의 후보 키 중에서 하나를 선택하여 사용합니다. 이전에는 사원의 아이디를 많이 사용하였으나, 최근에는 아이디를 사용하지 않고 이메일을 많이 사용합니다.
대체키 (Alternate Key)	후보키 중에서 기본키로 선정되고 남은 키들을 의미합니다. 기본키가 {아이디}이라면, {주민 번호}와 {이메일}은 대체키가 됩니다.
외래키 (Foreign Key)	특정 릴레이션의 기본 키를 참조하고 있는 다른 릴레이션의 컬럼을 의미합니다.

슈퍼키

{아이디, 주민번호} {아이디, 이메일}
{이메일, 주민번호} {아이디, 주민번호, 이메일}

후보키

기본키

{아이디}

대체키

{이메일}
{주민번호}

키(key)의 종류

관계형 데이터 모델 이론

제약(制約) 조건

데이터에 대한 정확성을 유지하기 위하여 원하지 않는 데이터가 입력/수정되는 것을 방지하기 위한 제한 조건입니다.

무결성(integrity_진실성)이란 데이터가 정확하다는 것을 의미합니다.

제약은 단어 그대로 특정 컬럼에 특정 데이터들이 추가 또는 수정 되지 못하도록 제한을 두는 것을 의미합니다.

제약 조건은 **키 제약 조건**과 **무결성 제약 조건**으로 나눌 수 있습니다.

제약 조건의 목적

- 사용자로 하여금 잘못된 데이터를 입력하지 못하도록 하기 위함
- 원치 않는 데이터를 수정하지 못하도록 사전에 테이블을 제한 사항을 두기 위함

키 제약 조건의 예시

사원 테이블

아이디	이름	주소	급여
an	안중근	석대문	800
kim9	김구	강남	600
yoon	윤봉길	용산	500

사원 테이블의 '아이디'처럼 튜플을 유일하게 식별할 수 있는 키는 다른 튜플의 키와 중복된 값이 존재해서는 안됩니다.

kim9 → primary key를 위반하고 있습니다. 동일한 아이디는 입력 불가능합니다.

← -200
급여는 음수 값이 들어오면 안됩니다.

관계형 데이터 모델 이론

제약(制約) 조건

데이터에 대한 정확성을 유지하기 위하여 원하지 않는 데이터가 입력/수정되는 것을 방지하기 위한 제한 조건입니다.

실체 무결성

각 인스턴스를 유일하게 식별할 수 있는 속성이나 속성 그룹을 가지고 있어야 합니다.
기본 키를 구성하는 아이디는 null 값이 되어서는 안되며,
반드시, unique 한 값을 가지고 있어야 합니다.

아이디	이름	주소
kim9	김구	용산
	유관순	마포

영역 무결성

속성들의 데이터 타입, 길이, 유효한 값이 일관되게 유지되어야 합니다.
국내 거주자들을 대상으로 하는 테이블에서 '뉴욕' 거주자는
존재해서는 안되며, 입사 일자에 '68'일이라는 데이터는
존재해서는 안됩니다.

아이디	이름	주소	입사일자
kim9	김구	용산	70/08/25
soon	유관순	뉴욕	80/07/17
lee	이순신	강남	85/08/68

참조 무결성

데이터 모델에서 정의된 실체 간의 관계 조건을 유지하는 것입니다.
게시물 릴레이션의 외래 키(작성자)는 부모 릴레이션의
기본 키(아이디)에 존재하는 값 이외의 값을 가질 수
없습니다. 작성자 park은 존재하지 않는 회원입니다.

아이디	이름	주소
kim9	김구	용산
soon	유관순	마포

글번호	작성자	글제목
1	kim9	java
2	soon	python
3	park	database

≡ 데이터 모델링이란?

≡ 데이터 모델링 과정

데이터베이스의 생명 주기 중에서 데이터 모델링이라 함은 ①요구사항 수집 및 분석 및 ②설계 과정을 말합니다.

요구 사항 수
집 및 분석



설계



구현

요구 사항 수집 및 분석

현실 세계의 대상 및 사용자의 요구 사항들을 수집하고 이를 체계적으로 정리하여 분석을 하는 단계입니다
사용자를 식별하고, 데이터 베이스에 대한 용도를 식별해야 합니다
수집된 데이터를 이용하여 명세를 작성해야 합니다

설계

설계는 다음과 같이 개념적, 논리적, 물리적 모델링 단계로 구분됩니다.
개념적 모델링은 중요한 개념들을 분리하여 핵심적인 Entity를 도출해내는 과정입니다.
논리적 모델링은 각각의 개념을 구체화 시키는 과정으로, 속성들에 대한 상세 정의 및 정규화를 수행하는 과정입니다.
물리적 모델링은 데이터 베이스 객체 생성에 따라 객체(테이블, 시퀀스 등등), 인덱스 등을 정의하는 단계입니다.

구현

명시한 요구 사항 및 설계를 이용하여 데이터 베이스를 구현하는 단계입니다.

≡ 데이터 모델링이란?

≡ 개념적 모델링(conceptual modeling)

요구 사항 분석의 내용을 토대로 업무의 핵심 개념들을 분류하고 전체적인 뼈대를 만드는 과정

주요 처리 내용

- 무슨 정보가 필요한가? → 개체(entity) 추출하기
- 개체, 속성, 관계로 현실 세계를 표현하는 기법
- 산출물로 ER 다이어그램(ERD, Entity Relationship Diagram)이 있습니다.

개념적 모델링의 필요성

- 고객의 업무 **요구 사항**을 정확히 **파악**하기 위함
- 사전 실수의 **방지** 및 오해의 **지양**
- 시스템의 **문서화**
- **설계**(물리적 DB 설계)에 대한 적절한 **기반 제공**
- 관계자들의 원활한 **의사 전달**
- 새로운 모델의 생성 및 변경시 **비용 절감**

표기 방법



ERD 사용 예시



☰ 데이터 모델링이란?

☰ 논리적 모델링(logical modeling)

ERD를 토대로 DBMS에 맞게 사상(mapping)시켜 실제 데이터베이스로 구현하기 위한 모델을 만드는 과정

논리적 모델링 처리 과정

- 실체(Entity)와 속성(Attribute)의 구조적 설계
- 개념적 모델링에서 추출하지 않았던 상세 속성들을 모두 추출함
- 정규화를 수행함
- 데이터 표준화 수행(용어 정리, 데이터 타입/크기 정의)



Entity와 Attribute

항목	설명
고객	고객 아이디, 고객 이름, 성별, 나이, 주소, 전화 번호
주문	송장 번호, 고객 번호(FK), 도서 번호(FK), 주문 일자, 주문 금액
상품	상품 번호, 상품 이름, 제조사 이름, 품목 단가

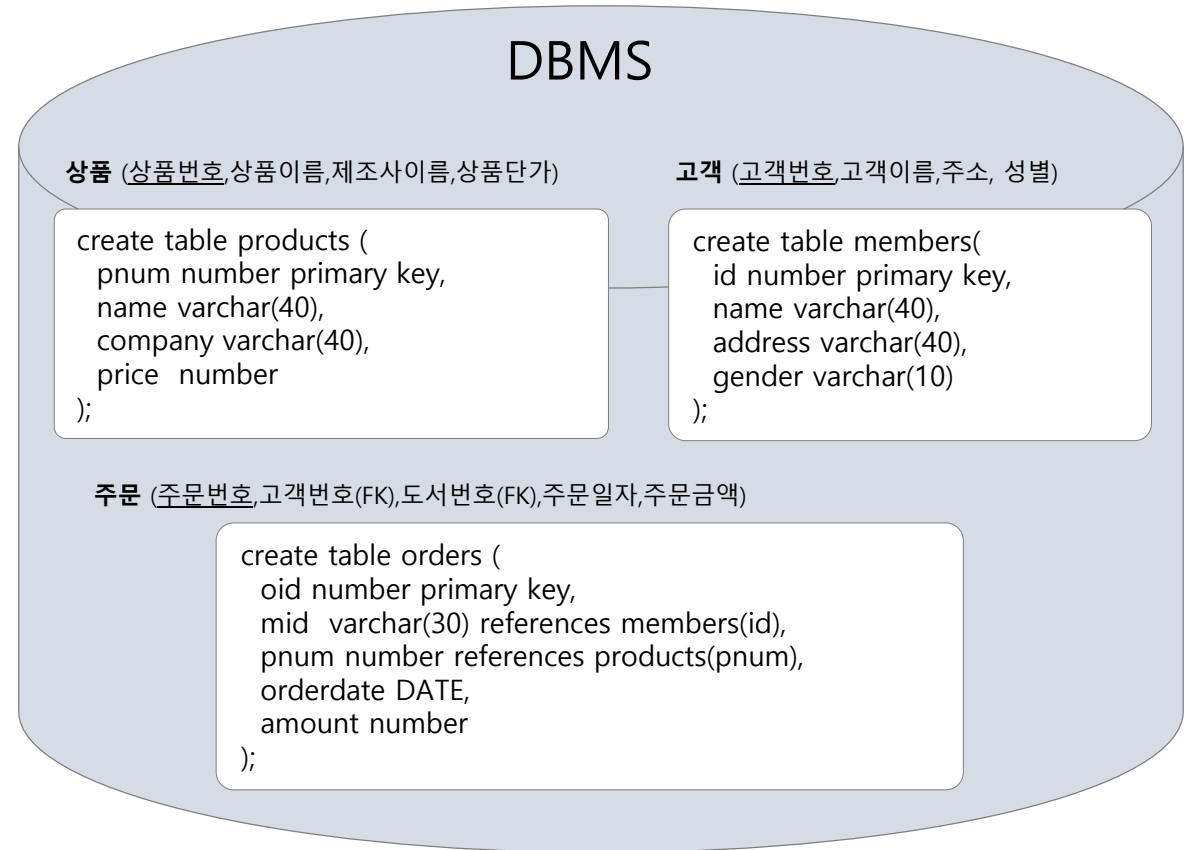
≡ 데이터 모델링이란?

≡ 물리적 모델링(physical modeling)

작성된 논리적 모델을 저장 장치에 저장하기 위한 물리적 구조를 정의하고 구현하는 과정

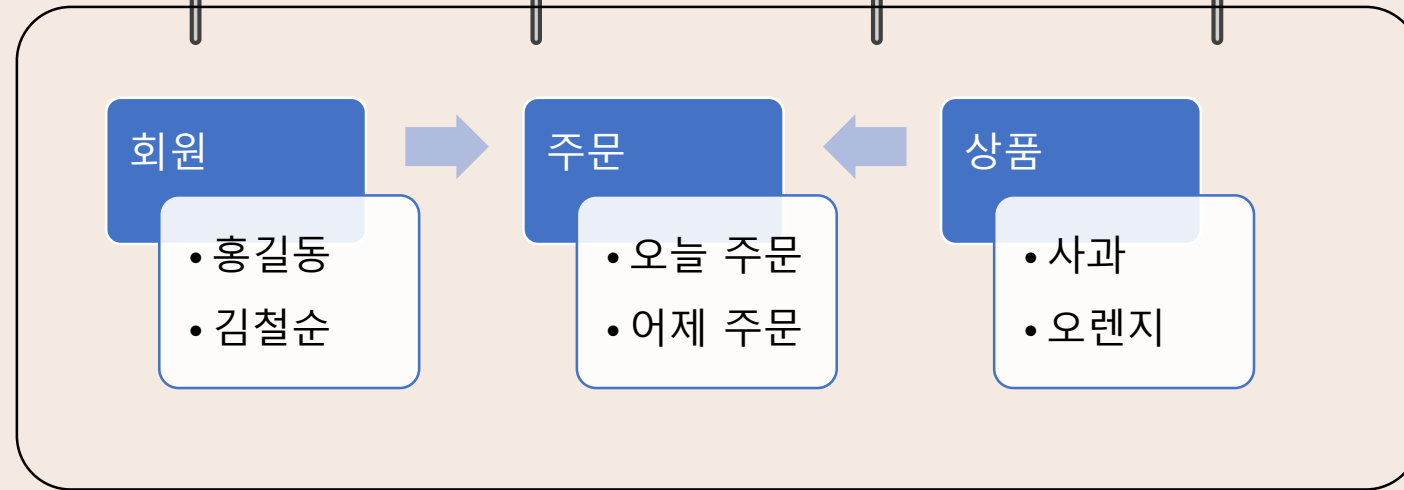
물리적 모델링 고려 사항

- 개발 DBMS 선정(mysql, Oracle)
- 컬럼의 데이터 타입과 Size 정의
- 데이터 사용량 분석
 - 어느 칼럼을 많이 사용하는가?
- 사용자들의 업무 프로세스 선택
- 역정규화
- 인덱스/데이터 분산 정의
 - 파티션 테이블 등등
- DB 객체 생성(뷰/인덱스/시퀀스 등등)
- DB 생성



Data Modeling

Entity와 Attribute



ER 구성 모델 요소

ER 모델의 구성 요소

엔터티(Entity), 속성(Attribute), 식별자(Identifier), 관계(Relationship)

항목	설명
엔터티(Entity)	업무가 관여하는 어떠한 것(Things)
속성(Attribute)	어떤 것의 성격(Attribute)
식별자(Identifier)	Entity 내의 행(instance)을 구분할 수 있는 속성
관계(Relationship)	업무가 관여하는 어떠한 것들간의 관계(Relationship)

사용 예시

남편 '현빈'은 키가 180cm에 성격은 친절하고 아내 '손예진'은 키가 165cm에 세심하며 활발한 성격입니다.

항목	설명
엔터티(Entity)	현빈, 손예진
속성(Attribute)	남자/여자, 키, 세심, 친절, 활발함
관계(Relationship)	서로 사랑하는 사이



이미지 출처) 중앙 일보

ER 구성 모델 요소

엔터티(Entity)

데이터 모델링 과정에서 식별이 되어야 하는 대상을 의미하고, 차후 Table으로 변경됩니다.

데이터로 **관리**를 수행해야 하는 중요 **대상**입니다.

하나의 어떠한 그룹으로 일반적으로 **명사**를 사용합니다.

영속적으로 존재해야 하는 어떠한 단위입니다.

물리적 모델링 과정에서 테이블로 만들어집니다.

상호 배타성, 식별성(주식별자)을 가지고 있습니다.

Instance라고 하는 객체들의 집합적 개념입니다.

ER 구성 모델 요소

엔터티 설계시 고려 사항

엔터티는 다음과 같은 사항들을 잘 고려하여 구성해야 합니다.

고유한 이름(Unique)을 사용해야 합니다.

예약어를 사용하지 말아야 합니다.

동의어 인식이 가능해야 합니다.

동음 이의어를 사용하지 말아야 합니다.

ER 구성 모델 요소

엔터티(Entity) 표시하기

엔터티는 **모서리**가 **등근 상자**로 표현합니다.

이름은 단수형으로 작성합니다.

크기나 위치는 큰 의미가 없습니다.

물리적 모델링 과정을 마치고 설계 중에는 테이블로 만들어집니다.

사용 예시

사원

직업

주문

교통 수단

연예인

ER 구성 모델 요소

엔터티와 instance

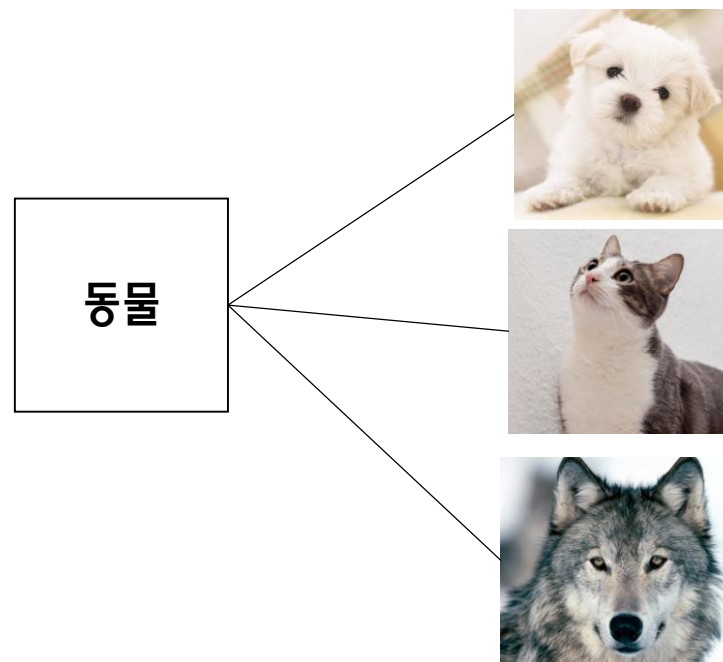
인스턴스는 Entity가 가질 수 있는 **구성 멤버**를 말합니다.

Entity	Instance
사람	손흥민, 푸틴, 이순신
동물	강아지, 고양이, 늑대
가수	오마이걸, 신화, 우주 소녀, 트와이스
운동 선수	이강인, 류현진, 황선우
운동 경기	축구, 야구, 농구

객체 지향과의 관계

특정 클래스로부터 생성이 되는 객체와 동일한 의미입니다.

실제 객체 변수를 다른 용어로 instance라고 부르기도 합니다.



이미지 출처) phoneky, freepik, 뉴스 핑귌

ER 구성 모델 요소

연습 문제

문제) 다음 물음에 대하여 대답해 보세요.

3개 이상의 엔티티 예시를 들어 보세요.

그리고, 각각 엔티티에 대하여 인스턴스를 예를 2개 이상 들어 보세요.

사용 예시

Entity	Instance
국가	한국, 미국, 영국, 중국, 일본
과일	사과, 바나나, 오렌지
의류	티셔츠, 블라우스, 청바지
커피	아메리카노, 카페 라떼, 에스프레소

ER 구성 모델 요소

속성(Attribute)

Entity를 설명하는 **정보/속성(Property)**을 의미합니다.

관리해야 하는 어떠한 특징으로 더 이상 분리할 수 없는 최소의 단위가 됩니다.

정보의 특정 부분(설명/분석/한정/분류)에 해당합니다.

속성은 데이터의 유형(문자열/숫자/날짜 등등)을 가집니다.

물리적 모델링 단계에서 컬럼(column)으로 변환이 됩니다.

엔터티의 단일 값(Single Value) 정보를 가집니다.

ER 구성 모델 요소

속성 설계시 고려 사항

속성은 다음과 같은 사항들을 잘 고려하여 구성해야 합니다.

가능한 한 작은 단위로 나누어야 합니다.

- 제품 코드 'PL123'는 PL(제품)과 123(Release)으로 분리합니다.

중복성(Redundancy)을 제거해야 합니다.

- 파생(derived data) 데이터는 업무상으로 보존은 하되, ERD에서 표현 하지 않도록 합니다.
- 예를 들어서 '연봉'은 '급여'로부터 구할 수 있는 값이므로, 파생 데이터입니다.

선택성을 고려해야 합니다.

- 필수 사항은 *로 표시하고, 옵션 사항은 o로 표시합니다.

Full Naming 기법을 사용하여 명확/상세하게 설명하도록 합니다.

- 수량(x) → 제품_수량(o) / 생산_수량(o)
- 날짜(x) → 시작_일자(o) / 종료_일자(o)

ER 구성 모델 요소

속성 예시

인스턴스는 Entity가 가질 수 있는 **구성 멤버**를 말합니다.

항목	설명
사람	이름, 혈액형, 이상형, 성격
동물	종(조류/파충류 등등), 발의 수, 식성(육식/채식)
가수	(그룹) 이름, 인기곡, 소속사
운동 선수	이름, 경기 종목, 올림픽 참가 여부
운동 경기	이름, 종류(구기/육상 등등), 경기 인원

ER 구성 모델 요소

속성의 선택성

필수 사항은 *로 표시하고, 옵션 사항은 o로 표시합니다.

특정 엔터티(Entity)나 관계(Relationship)의 속성이 얼마나 필수적인지 혹은 선택적인지를 나타내는 개념입니다.

속성의 선택성은 데이터베이스 설계에서 중요한 역할을 하며, 엔터티와 관계 간의 관계를 정의하고 데이터의 무결성을 유지하는 데 도움이 됩니다.

항목	설명
필수 속성	모든 인스턴스에서 반드시 값이 존재해야 함을 나타냅니다. 데이터베이스에 레코드를 추가할 때 반드시 값을 지정해야 하는 속성이 필수적인 속성입니다. *(mandatory) 기호로 표시합니다.
선택 속성	모든 인스턴스에서 값이 있을 수도 있고 없을 수도 있는 속성입니다. NULL 값을 가질 수 있으며, 데이터베이스에 레코드를 추가할 때 값을 지정하지 않아도 됩니다. o(optional) 기호로 표시합니다.
반선택(Conditional/Partial)	특정 조건이 충족될 때만 값이 존재하는 속성입니다. 예를 들어, 결혼 상태가 "기혼"일 때만 결혼 날짜 속성이 존재하는 경우입니다.

사용 예시

회원 정보에서 생일과 이메일은 반드시 입력될 필요가 없습니다.

따라서 선택적인 속성이 되고, 나머지 항목들은 필수 속성이 됩니다.

사번	이름	주소	생일	이메일
kim	김윤선	마포 상수동	1970/08/15	bluesky@naver.com
song	송지숙	노원 상계동		redsun@daun.net
jang	장윤석	강남 신사동	1990/12/25	

* 사번
* 이름
* 주소
o 생일
o 이메일

ER 구성 모델 요소

연습 문제

문제) 다음 물음에 대하여 답해 보세요.

다음 문제를 읽고, Entity 및 Attribute를 도출해 보세요.

문제 내용

나는 관리 기법으로 운영되는 강사 지도 강좌를 제공하는 교육 기관의 관리자입니다.
우리는 많은 강좌를 가르치고 있으며 각 강좌는 code, 강좌명, 수업료를 가지고 있습니다.
"Python 기초"와 "R 프로그래밍"은 우리의 인기 강좌 중의 하나입니다.
강좌의 수업 일수는 1~4 일간으로 다양합니다.
우리는 각 강사의 이름과 전화 번호를 관리하며, 강좌를 개설하고 강사를 배정합니다.
각 강사들은 여러 개의 강좌를 가르칠 수 있으며, 특히 '김철수'와 '박영희' 강사는 우리의 최고 강사들입니다.
각 강좌는 단 한명의 강사에 의해 진행되며 어떤 강좌는 아직 강사가 배정되지 않을 수도 있습니다.
한 학생이 동시에 여러 강좌를 수강할 수 있으며 많은 학생이 그렇게 합니다.
예를 들어 ABC 전자 회사의 홍길동은 우리가 제공한 모든 강좌를 수강했습니다.
우리는 각 학생의 성명과 전화 번호를 관리하고자 하며, 때로는 학생과 강사가 그들의 전화 번호를 알려 주지 않을 때도 있습니다.

강사
이름
전화
번호

강좌
코드
강좌
명

강좌
명
수업
료

연습 문제

문제) 다음 물음에 대하여 대답해 보세요.

다음 문제를 읽고, Entity 및 Attribute를 도출해 보세요.

문제 내용

나는 비디오 가게의 사장이며, 이 곳에서 관리할 테이프를 수백개 이상 보유하고 있습니다.

각 비디오 테이프는 테이프 번호를 가지고 있으며, 각 영화마다 제목과 종류(예를 들어 코미디, 공포, 드라마, 액션, 전쟁 등)를 알 필요가 있습니다.

우리는 각 영화당 많은 테이프를 보유하고 있으며, 영화마다 특정 번호를 부여하고 각 테이프가 어떤 영화를 포함하고 있는지 관리합니다.

테이프는 Beta 혹은 VHS 방식일 수 있습니다.

우리는 각 영화를 위해 적어도 한개 이상의 테이프를 보유하고 있으며, 각 테이프는 항상 한가지의 영화를 담고 있습니다.

보유한 테이프의 길이는 매우 길어 복수개의 테이프로 된 영화는 하나도 없다. 우리는 특정 배우가 출연한 영화를 자주 찾습니다.

‘안성기’나 ‘강수연’은 항상 인기가 좋다. 그래서 우리는 각 영화마다 주연배우를 알 필요가 있으며 본명 및 생년월일까지도 알고자 합니다.

우리는 보유하고 있는 영화의 주연들에 대한 정보만 관리하고 싶습니다.

우리는 많은 고객을 보유하고 있으며 신용 고객 클럽에 가입한 회원들에게만 테이프를 대여합니다.

이 클럽에 가입하기 위하여 고객은 좋은 신용을 가져야 하며 각 회원들의 성명, 주소,전화 번호,회원 번호를 관리하고자 합니다.

우리는 고객이 현재 어떤 테이프를 빌려 갔는지를 관리하고자 하며 고객은 한번에 여러 개의 테이프를 빌려갈 지도 모릅니다.

또한, 우리는 현재의 대여 정보만 관리하지 과거의 이력 정보는 관리하지 않습니다.

영화 제목
종류
테이프 번호

배우
성명
주소
전화 번호
회원 번호

고객
성명
주소
전화 번호
회원 번호

테이프
종류
영화 제목
테이프 번호

엔터티의 확장

엔터티는 슈퍼 타입 엔터티나 서브 타입의 엔터티로 확장이 가능합니다.

고객 관리 시스템

우리 매장의 고객은 내국인과 외국인으로 분류하고 있습니다.

내국인은 고객 번호 및 이름, 주민 번호, 생일, 주소 정보를 가지고 관리합니다.

외국인은 고객 번호 및 이름 이외에, 외국인 등록 번호, 국적, 국내 거주 기간을 사용하여 관리를 하고 있습니다.

고객 현황

내국인 고객

- 고객 번호
- 이름
- 주민 번호
- 생일
- 주소

외국인 고객

- 고객 번호
- 이름
- 외국인 등록 번호
- 국적
- 거주 기간

Entity 확장

고객(SuperType)

- 고객 번호
- 이름

내국인(SubType)

- 주민 번호
- 생일
- 주소

외국인(SubType)

- 외국인 등록 번호
- 국적
- 거주 기간

ER 구성 모델 요소

엔터티의 확장

엔터티는 슈퍼 타입 엔터티나 서브 타입의 엔터티로 확장이 가능합니다.

항목	설명
SuperType	Entity 내의 공통적인 속성을 따로 끄집어 내어 놓은 것입니다. 고객은 내국인과 외국인을 모두 의미합니다.
SubType	Entity 내의 차별적인 속성을 의미합니다. 특화된 속성만 별도로 관리합니다. 주민 번호는 내국인은 보유하고 있고, 외국인 등록 번호는 존재하지 않습니다.

슈퍼 타입의 특징

모든 속성들은 서브 타입으로 상속(Inheritance)되는 공통 속성입니다.

서브 타입의 특징

슈퍼 타입의 모든 속성을 상속 받습니다.
슈퍼 타입의 모든 관계를 상속 받습니다.
서브 타입 그 자체도 Entity이므로, 속성/관계/업무 기능을 별도로 가질 수 있습니다.
서브 타입은 자체 서브 타입을 가질 수도 있습니다.
일반적으로 2~3 Level 정도가 적당합니다.

항목	설명
일반화(Generalization)	두 개 이상의 하위 수준의 엔터티를 결합시켜 한 개의 상위 수준의 엔터티로 통합하는 기법 상향식 접근 방식
특수화(Specialization)	한 개의 상위 수준의 엔터티를 두 개 이상의 하위 수준의 엔터티로 분해하는 기법 하향식 접근 방식

엔터티의 확장

문제) 서브 타입 예시를 추가해 주세요.

사원 관리 시스템

우리 회사의 직원들은 크게 기술직 직원과 관리직 직원으로 분류하고 있습니다.

모든 직원들은 이름, 전화 번호, 주민 번호 및 주소 정보를 가지고 관리됩니다.

기술직 직원은 경력에 따른 기술 등급 및 수당 정보를 관리해야 합니다.

관리직 직원들은 관리해야 할 직원들의 수가 정해져 있고, 관리에 따른 수당을 별도로 받고 있습니다.

고객 현황

기술직 직원

- 이름
- 전화 번호
- 우편 번호
- 주소
- 기술 등급
- 기술 수당

관리직 직원

- 이름
- 전화 번호
- 우편 번호
- 주소
- 관리직원수
- 관리 수당

Entity 확장

직원(SuperType)

- 이름
- 전화 번호
- 우편 번호
- 주소
- 직원 구분

기술직(SubType)

- 기술 등급
- 기술 수당

관리직(SubType)

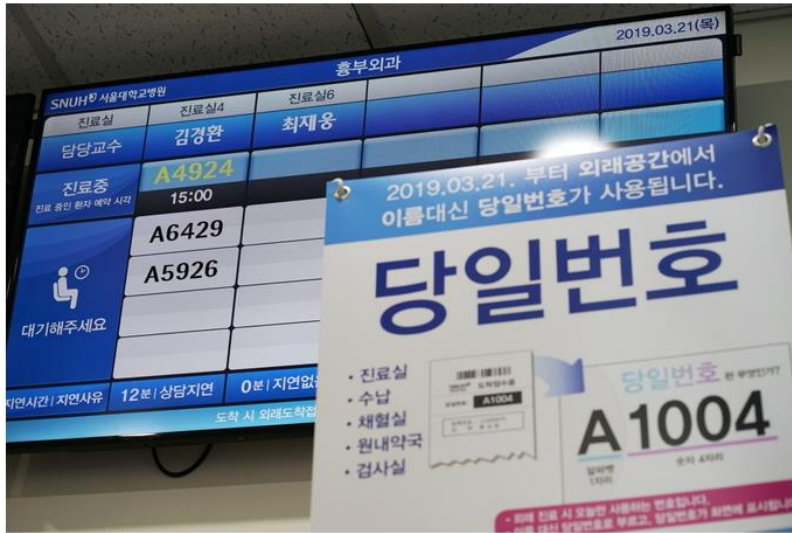
- 관리직원수
- 관리 수당

식별자(Identifier)

인스턴스를 개별적으로 식별할 수 있는 속성을 의미합니다.

데이터 모델링에서 식별자(Identifier)는 엔터티 내에서 각 개체를 고유하게 식별할 수 있는 속성을 의미합니다.
각 엔터티는 최소한 하나 이상의 식별자를 가져야 합니다.

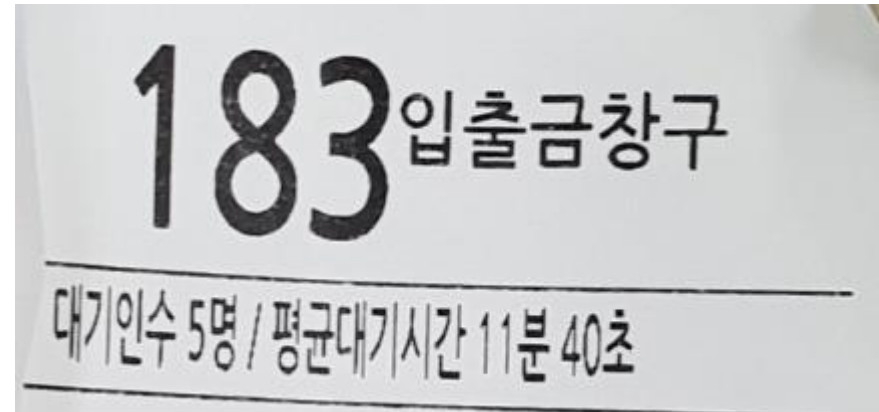
식별자 예시



▲ 외래대기모니터에 환자 개인의 당일번호가 표시된다.

서울 대병원 환자 이름 호명하지 않는 진료 서비스 시작
진료 당일 이름 대신 본인의 고유 번호를 부여 받아 하루 동안 사용
출처) 메디포뉴스

(<http://www.medifonews.com/news/article.html?no=144863>)



은행 번호 표

ER 구성 모델 요소

식별자의 특징

식별자는 유일성, 최소성, 불변성, 존재성의 특징을 가지고 있습니다.

유일성(Uniqueness)

- 직원들의 직원 번호, 회원들의 아이디, 은행의 번호 대기표

최소성(Minimum)

- 유일성을 만족시키는 최소한의 속성으로만 구성되어야 합니다.
- 회원 구분을 위하여 [아이디+이름] 보다는 [아이디]만으로 구분하도록 합니다.

불변성(Stability)

- 포털 사이트에서 회원 가입 완료 후 id는 변경이 불가능합니다.

존재성(Mandatory)

- 모든 직원들을 구분하려면 '직원 번호'는 반드시 입력이 되어야 합니다.

식별자(Identifier)

다음은 식별자 예시입니다.

데이터 모델링에서 식별자는 각 엔티티를 구분하고 고유하게 식별할 수 있는 속성으로 사용됩니다.
각 식별자는 엔티티의 특성과 요구 사항에 맞게 선택되어야 합니다.

고객 정보 엔티티(Customer)

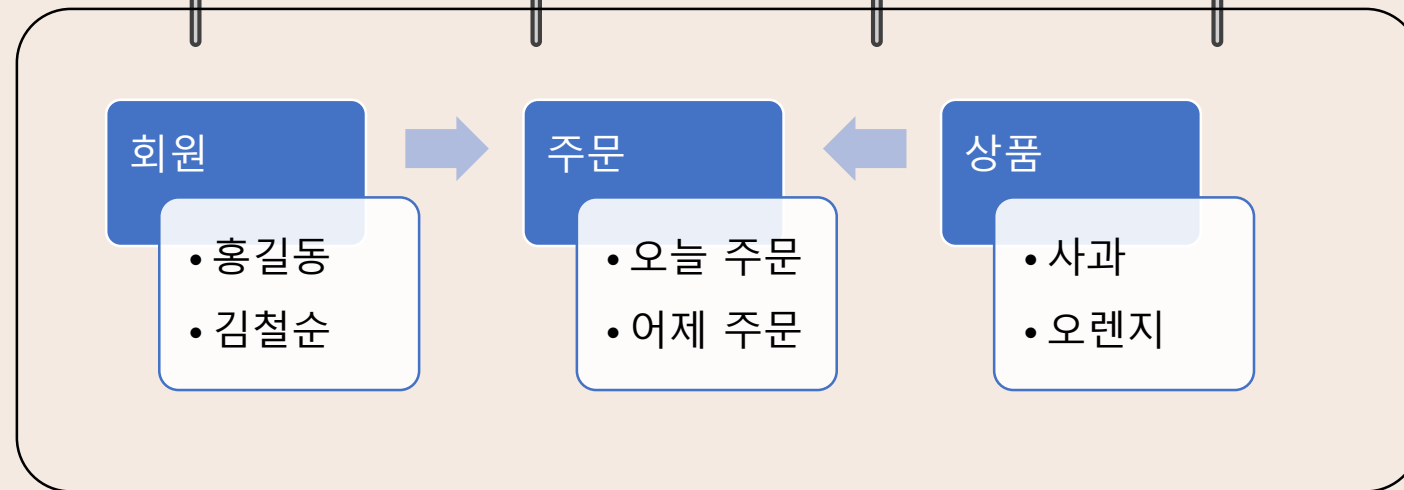
식별자 예시	설명
고객 ID(Customer ID)	각 고객을 식별하는 고유한 값으로 숫자나 문자열로 표현될 수 있습니다.
주민 등록 번호(Social Security Number)	국가에 따라 고유한 고객 식별 번호로 사용될 수 있습니다.
이메일 주소(Email Address)	고객의 이메일 주소를 식별자로 사용할 수 있습니다.
회원 번호(Membership Number)	고객이 가입한 회원제 프로그램에서 제공되는 고유 번호입니다.

제품 정보 엔티티(Product)

식별자 예시	설명
제품 코드(Product Code)	제품을 고유하게 식별하는 코드로, 주로 숫자나 문자열 조합으로 표현됩니다.
바코드(Barcode)	상품의 바코드를 사용하여 제품을 고유하게 식별할 수 있습니다.
시리얼 번호(Serial Number)	각 제품마다 고유한 시리얼 번호를 사용하여 식별할 수 있습니다.

Data Modeling

Relationship



ER 구성 모델 요소

관계(Relationship)

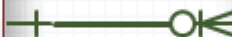
Entity와 Entity 간에 존재하는 업무 규칙을 정의하고, Entity 간에 어떠한 관계가 이루어지는 지 표현해야 합니다.

Entity들이 상호 연관되는 양상을 의미합니다.

관계는 항상 엔터티 2개 사이에 존재(2개의 관점)하는 개념입니다.

관계는 Entity 양쪽에서 이름이 만들어 집니다.

회원			
🔑	아이디	VARCHAR(30)	id
●	이름	VARCHAR(255)	name
●	주소	VARCHAR(255)	address



게시물			
👤	작성자	VARCHAR(30)	writer
●	글제목	VARCHAR(50)	subject
●	글내용	VARCHAR(255)	content
●	작성일자	DATE	regdate

ER 구성 모델 요소

관계 예시

관계는 2개의 Entity 사이에서 상호 발생하는 개념입니다.

항목	설명
사람 ↔ 직업	[사람]은 직업을 가집니다. 특정 [직업]은 특정 [사람]에 의하여 할당이 될 수 있습니다.
제품 ↔ 종류	[제품]은 여러 부류의 [종류]로 분류가 됩니다. [종류]는 특정 [제품]에 대한 분류 입니다.
사원 ↔ 부서	[사원]은 특정 [부서]에 속해 있습니다. [부서]에는 몇 명의 [사원]을 포함 해야 합니다.
손님 ↔ 예약	[손님]이 [예약]을 합니다. [예약]은 [손님]에 의해 발생이 됩니다.

ER 구성 모델 요소

관계의 구성 요소

관계는 차수, 선택성, 식별성, 관계 이름 등으로 구성이 됩니다.

항목	설명
차수	어떤 엔터티가 다른 엔터티와의 관계에서 상호 몇 개의 인스턴스가 대응이 되었는 지를 표시하는 것입니다.
선택성(optionality)	엔터티 인스턴스가 상대 엔터티 인스턴스에 반드시 존재해야 하는 지를 표시하는 것입니다. '선택(optional)' 속성은 점선으로 표시하며, '0개 이상 xxx를 가질수 있습니다.'라고 표현합니다. '필수(mandatory)' 속성은 직선으로 표시하며, '반드시 xxx해야 합니다.'라고 표현합니다.
식별자 상속	엔터티의 식별자를 다른 엔터티에서 상속을 받을 때, 식별자로 상속 받을지, 일반 속성으로 상속 받을지를 표시하는 것입니다.
관계 이름	관계의 의미나 이름을 표시하는 데, 의미를 명확하게 한다는 의미에서 일반적으로 양쪽 엔터티에 모두 기술합니다.

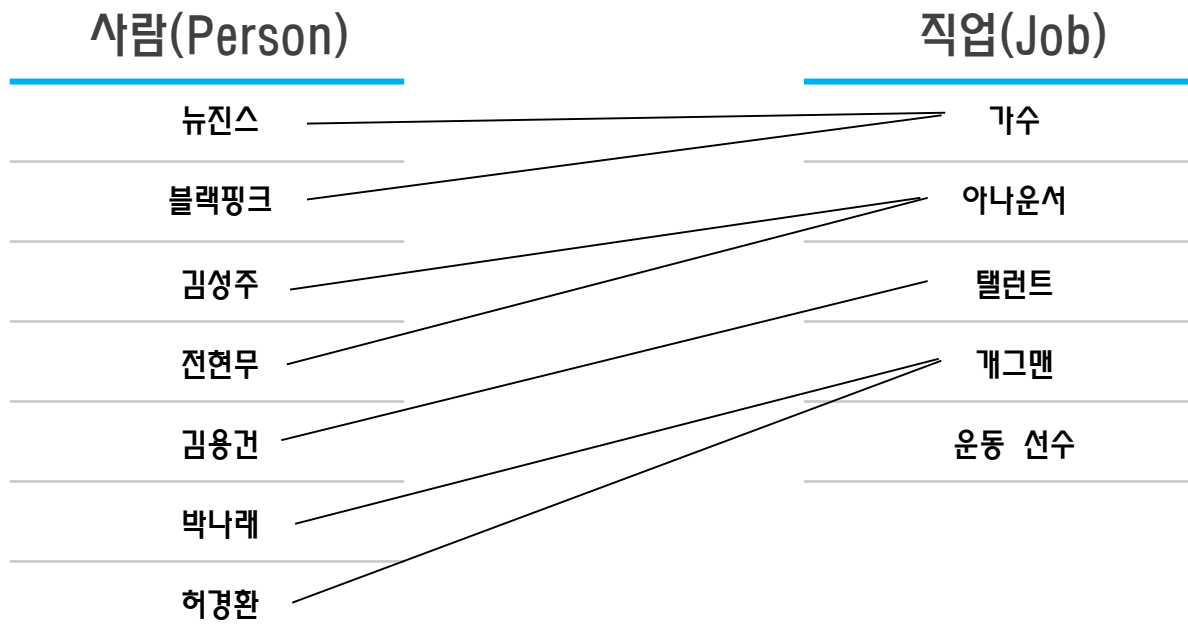
ER 구성 모델 요소

관계도 그리기

다음 그림을 보고 관계도를 그려 보세요.

모든 사람은 반드시 1개의 직업을 가져야 한다고 가정하겠습니다.

가장 먼저 구성 요소들간의 연결 선을 그려 봅시다.



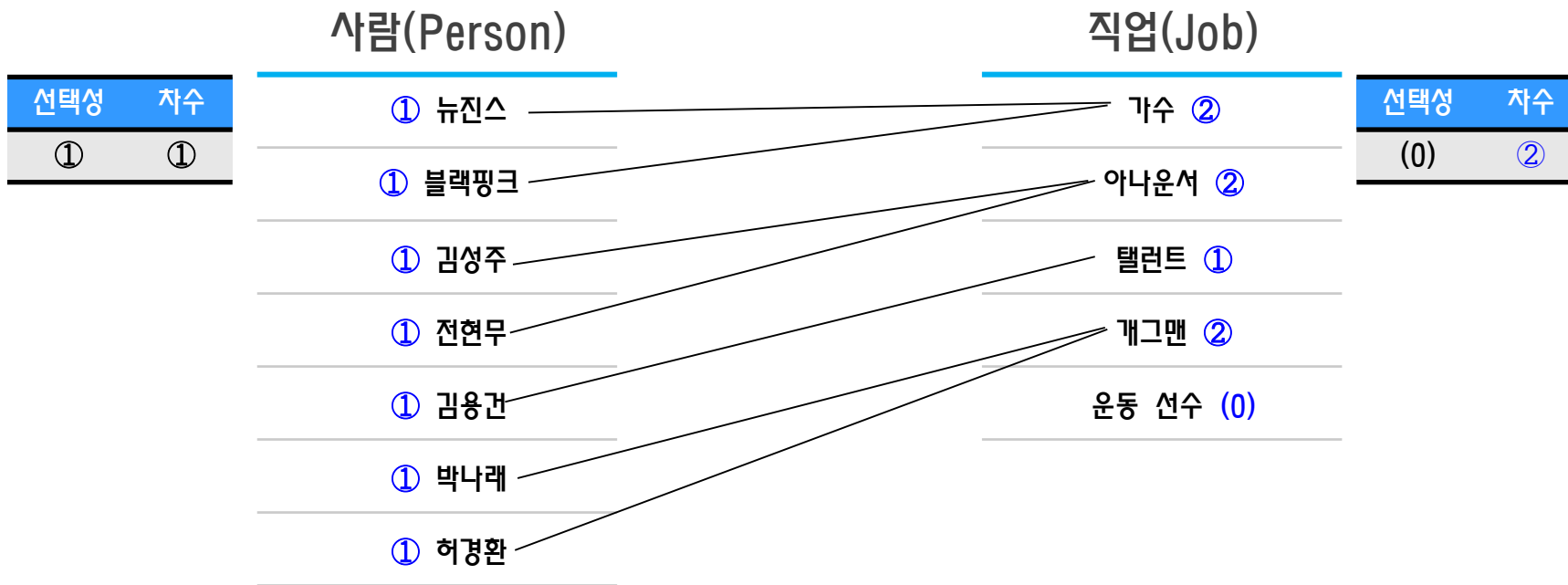
ER 구성 모델 요소

관계도 그리기

다음 그림을 보고 관계도를 그려 보세요.

연결선의 수를 이용하여 선택성과 차수를 표시해 봅니다.

'선택성'은 최소 값, '차수'는 최대 값으로 표시 가능합니다.



ER 구성 모델 요소

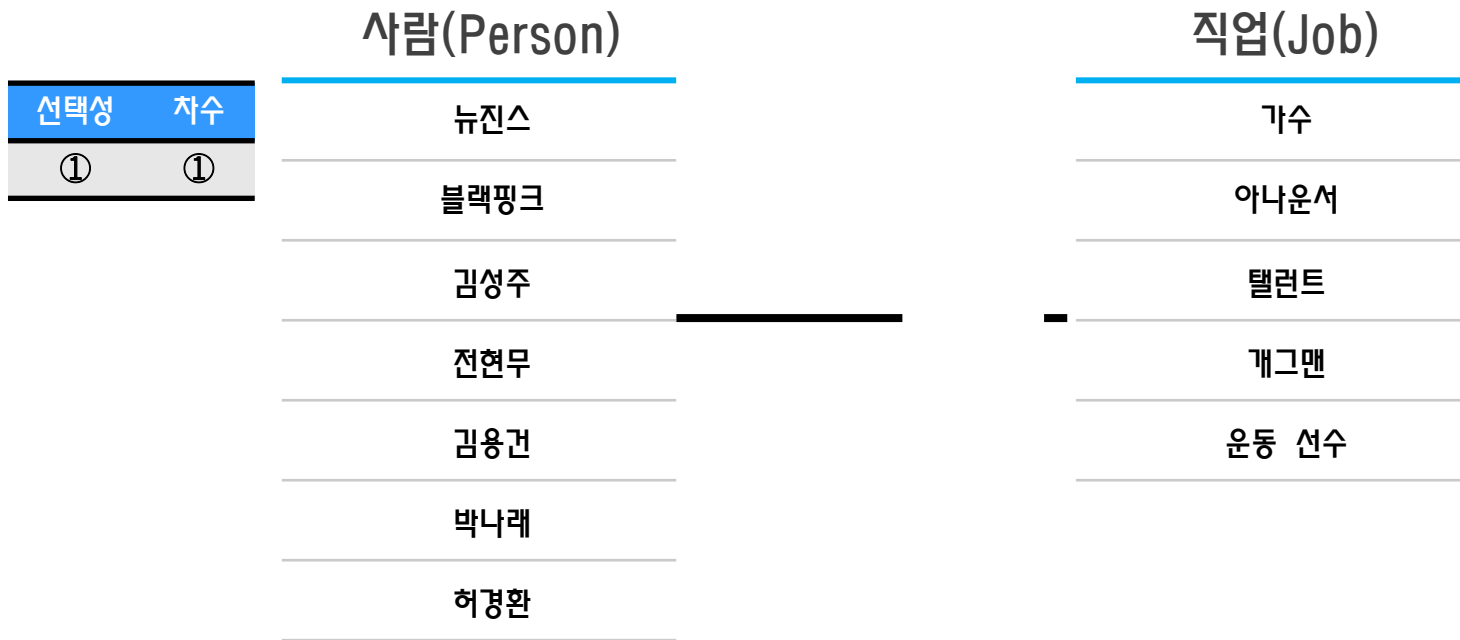
관계도 그리기

다음 그림을 보고 관계도를 그려 보세요.

‘사람(Person)’ Entity에 대한 관계를 그려 봅니다.

선택성이 ①이라는 것은 필수(mandatory) 속성이라는 의미로, 자신 Entity Person의 영역에 실선을 길게 그려 줍니다.

차수는 ①이고, 맞은 편 Entity Job 쪽에 짧은 선 하나를 그려 줍니다.



ER 구성 모델 요소

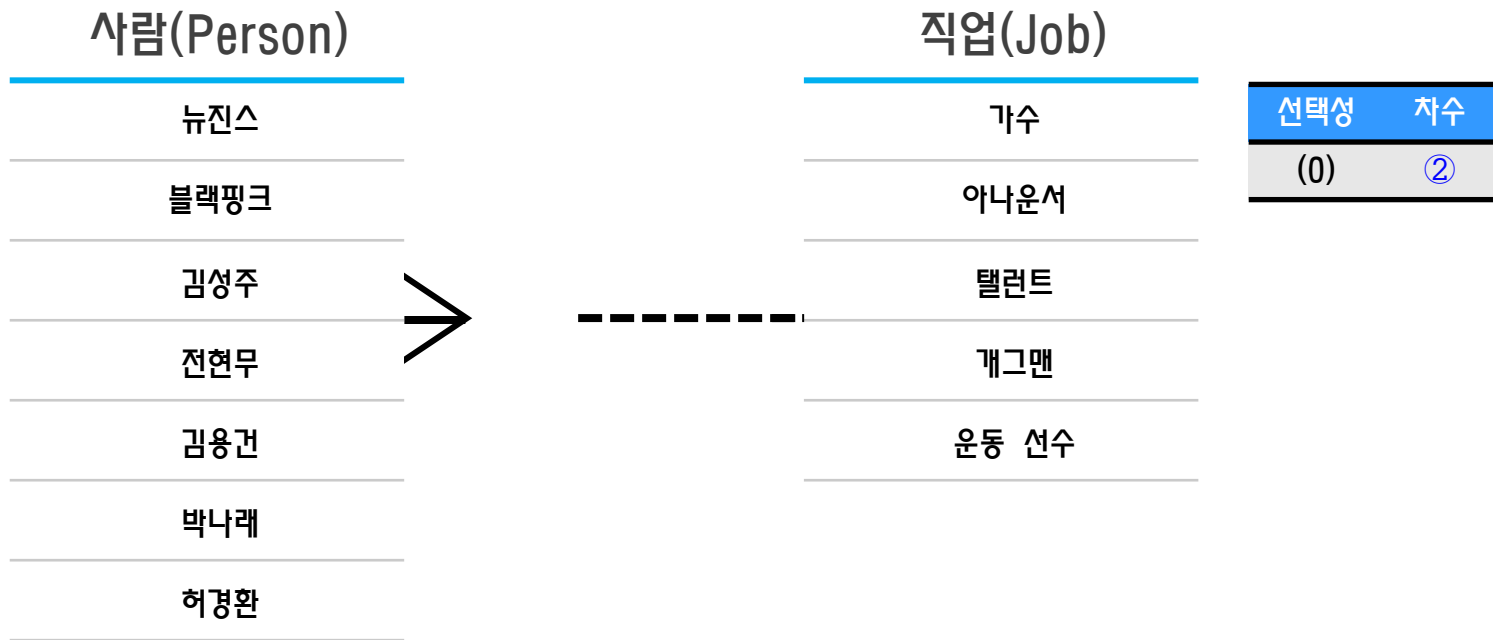
관계도 그리기

다음 그림을 보고 관계도를 그려 보세요.

‘직업(Job)’ Entity에 대한 관계를 그려 봅니다.

선택성이 (0)이라는 것은 optional 속성이라는 의미로, , 자신 Entity Job의 영역에 점선을 길게 그려 줍니다.

차수는 ②이고, 맞은 편 Entity Person에 crow's foot(까마귀 발) 형태로 그려 줍니다.

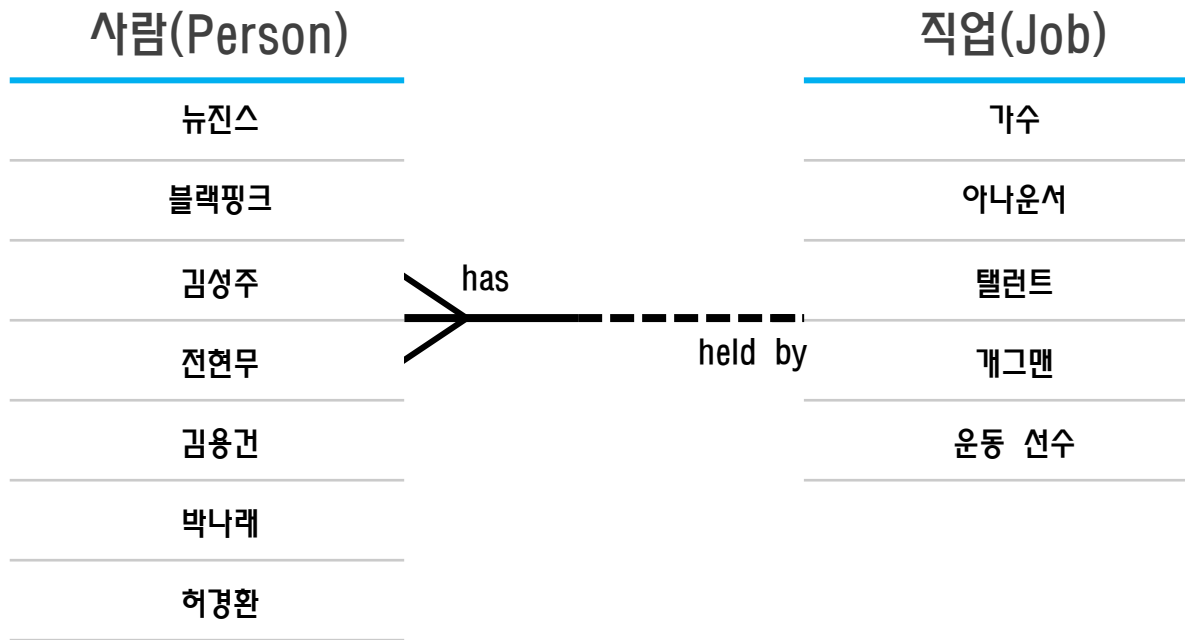


ER 구성 모델 요소

관계도 그리기

다음 그림을 보고 관계도를 그려 보세요.

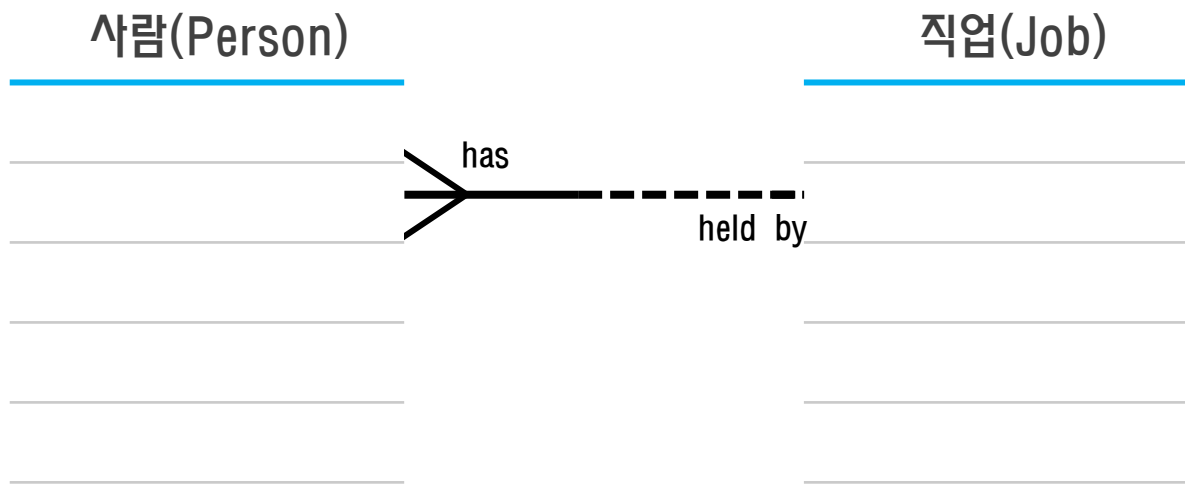
두 Entity 간의 관계도를 동시에 그리고, 관계 이름을 표시합니다.



ER 구성 모델 요소

관계 읽기 01

다음 관계도를 보고 올바른 관계 문장을 모두 골라 보세요.

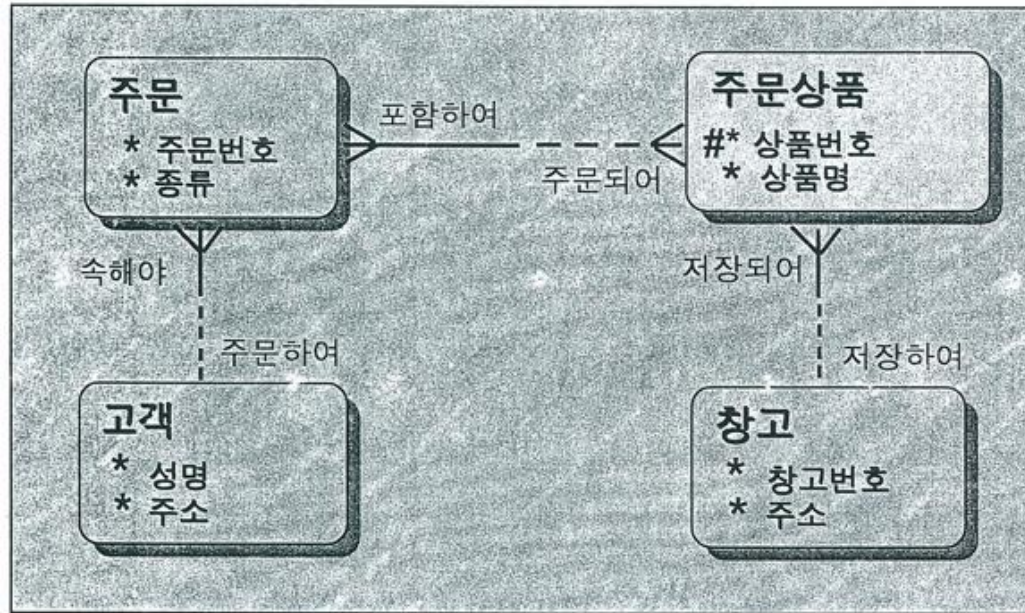


- A) [사람]은 단 하나의 [직업]을 가져야 합니다.
- B) [사람]은 하나 이상의 [직업]을 가질수 있습니다.
- C) [직업]은 단 하나의 [사람]을 반드시 포함해야 합니다.
- D) [직업]은 0명 이상의 [사람]들을 포함할 수 있습니다.

ER 구성 모델 요소

관계 읽기 02

다음 그림을 보고 관계 문장을 만들어 보세요.



- ◎ 각각의 주문은 () 주문 상품을 ()
- ◎ 각각의 주문 상품은 () 주문에 의하여 ()
- ◎ 각각의 주문은 () 고객에 ()
- ◎ 각각의 고객은 () 주문을 ()
- ◎ 각각의 주문 상품은 () 창고에 ()
- ◎ 각각의 창고는 () 주문 상품을 ()

ER 구성 모델 요소

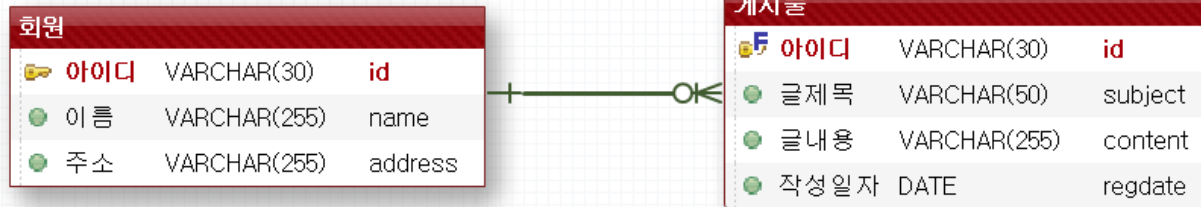
식별/비식별 관계

식별과 비식별 관계 여부는 상속된 속성의 위치에 따라 결정이 됩니다.

항목	설명
식별 관계	상위 엔터티의 식별자 전체가 하위의 식별자에 포함 이 되는 관계
비식별 관계	상위 엔터티의 식별자 전체가 하위 엔터티의 일반 속성에 포함 이 되는 관계

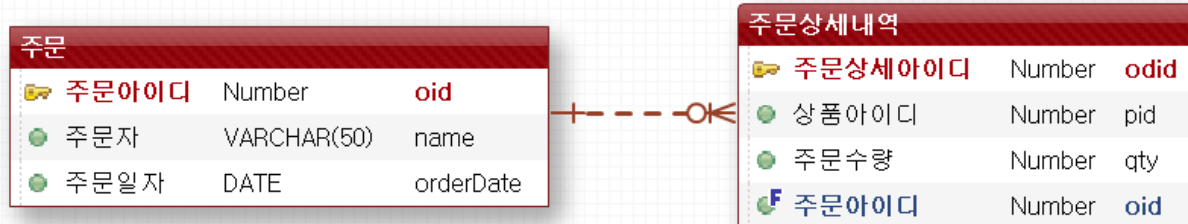
식별 관계

회원 Entity의 식별자인 id 속성이 게시물 Entity의 식별자에 포함이 되어 있습니다.



비식별 관계

주문 Entity의 식별자인 oid 속성이 주문 상세 내역 Entity의 일반 속성에 포함이 되어 있습니다.



≡ 관계형 데이터 모델 이론

≡ 관계 유형

관계 유형은 한 개의 엔터티가 다른 엔터티와 대응하는 인스턴스의 수를 의미합니다.

대응 되는 수에 따라서, 일대일, 일대다, 다대다의 유형이 존재합니다.

관계의 종류	표현 기법	관계 예시	설명
일대일	1:1	회원(일) ↔ 장바구니(일)	회원은 하나의 장바구니를 소유할 수 있습니다.
일대다	1:m	장바구니(일) ↔ 상품(다) 회원(일) ↔ 주문(다)	1개의 장바구니에는 여러 개의 상품을 담을 수 있습니다. 1명의 회원은 여러 개의 주문이 가능합니다.
다대다	m:n	회원(다) ↔ 상품(다)	

≡ 관계형 데이터 모델 이론

≡ 관계 유형(일대일)

특정 엔터티의 인스턴스 1개가 다른 엔터티의 인스턴스 1개와 관계 유형을 맺는 경우입니다.

일반적으로 일대일 유형은 흔히 발생하지 않은 케이스입니다.

특정 회원이 쇼핑을 하는 경우에 장바구니를 하나 생성해야 합니다.

관계의 종류	표현 기법	관계 예시	설명
일대일	1:1	회원(일) ↔ 장바구니(일)	회원은 하나의 장바구니를 소유할 수 있습니다.

case a)

—————

case b)

—————

case c)

- - - - -

관계형 데이터 모델 이론

관계 유형(일대다)

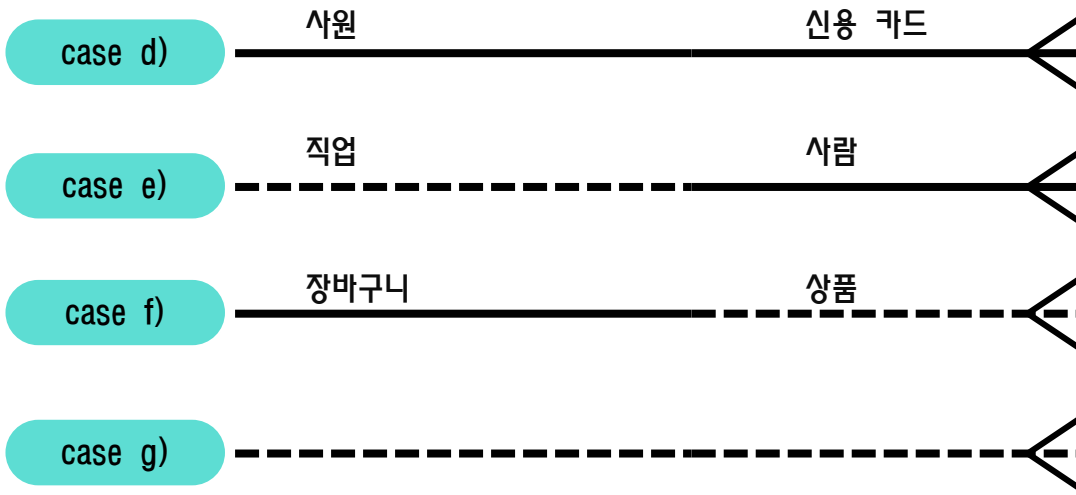
특정 엔터티의 인스턴스 1개가 다른 엔터티의 인스턴스 여러 개와 관계 유형을 맺는 경우입니다.

가장 일반적으로 자주 발생하는 패턴입니다.

일대다에서 '일'에 해당하는 엔터티를 Parent 또는 Master 엔터티라고 합니다.

'다'에 해당하는 엔터티를 Child 또는 Detail 엔터티라고 합니다.

관계의 종류	표현 기법	관계 예시	설명
일대다	1:m	장바구니(일) ↔ 상품(다) 회원(일) ↔ 주문(다)	1개의 장바구니에는 여러 개의 상품을 담을 수 있습니다. 1명의 회원은 여러 개의 주문이 가능합니다.



case d) 두 엔터티는 필수 엔터티로써 상대 엔터티가 존재하지 않으면 존재할 수 없는 엔터티

case e) 가장 일반적인 관계 유형으로써, '일'쪽의 엔터티는 독립적으로 존재하고, '다'쪽의 엔터티는 상대 엔터티의 맥락에서만 존재합니다.

case f) 별도의 슬라이드 참조

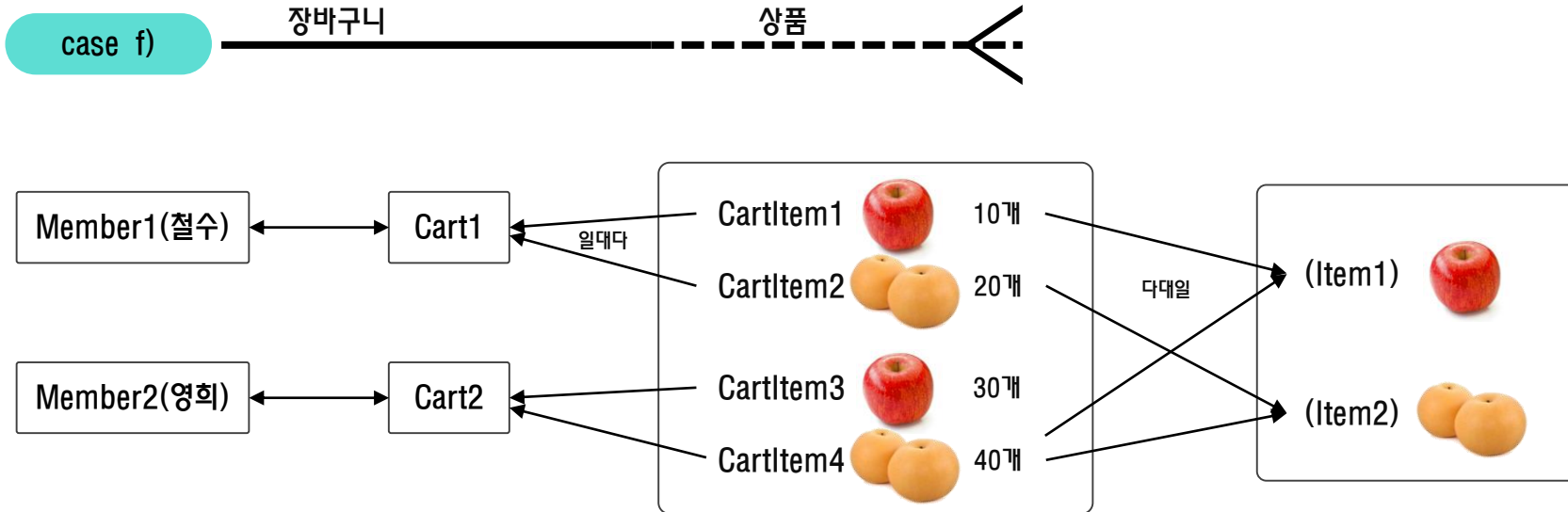
case g) 자주 발생하지 않는 케이스입니다.

≡ 관계형 데이터 모델 이론

≡ 관계 유형(일대다)

특정 엔터티의 인스턴스 1개가 다른 엔터티의 인스턴스 여러 개와 관계 유형을 맺는 경우입니다.

고객의 장바구니에는 여러 개의 상품들을 담을 수 있습니다.
모든 상품들이 장바구니에 담길 필요는 없고, 고객은 필요한 품목만 골라서 담을 수 있습니다.



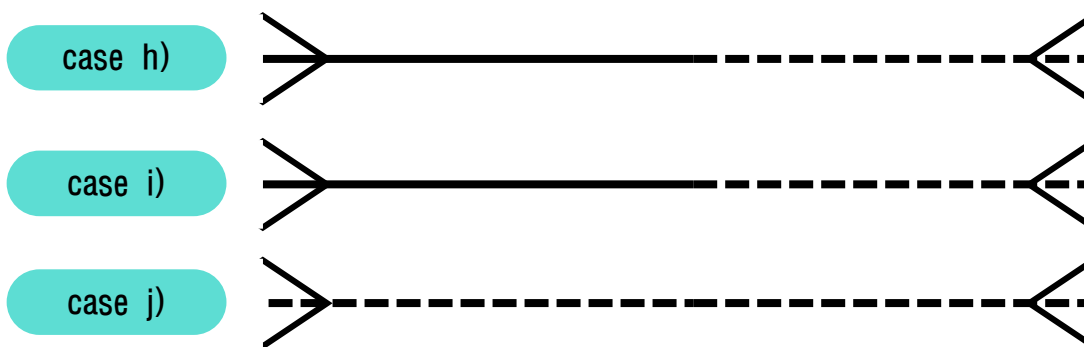
≡ 관계형 데이터 모델 이론

≡ 관계 유형(다대다)

특정 엔터티의 인스턴스 여러 개가 다른 엔터티의 인스턴스 여러 개와 관계 유형을 맺는 경우입니다.

회원이 상품 구매를 할 때, 여러 개를 동시에 구매할 수 있습니다.
특정 상품은 여러 명의 회원들에게 구매될 수 있습니다.

관계의 종류	표현 기법	관계 예시	설명
다대다	m:n	회원(다) ↔ 상품(다)	



case h) 일반적인 환경에서 혼하지 않는 경우입니다.
case i), case j)는 모델의 초기 버전에서 일반적으로 많이 발생합니다.
반드시, 교차(중간) 엔터티를 도입하여 일대다 관계로 변형하여 처리하도록 합니다.

ER 구성 모델 요소

연습 문제

문제) 다음 문제에 대하여 ER 모델(엔터티와 Relationship)을 만들어 보세요.

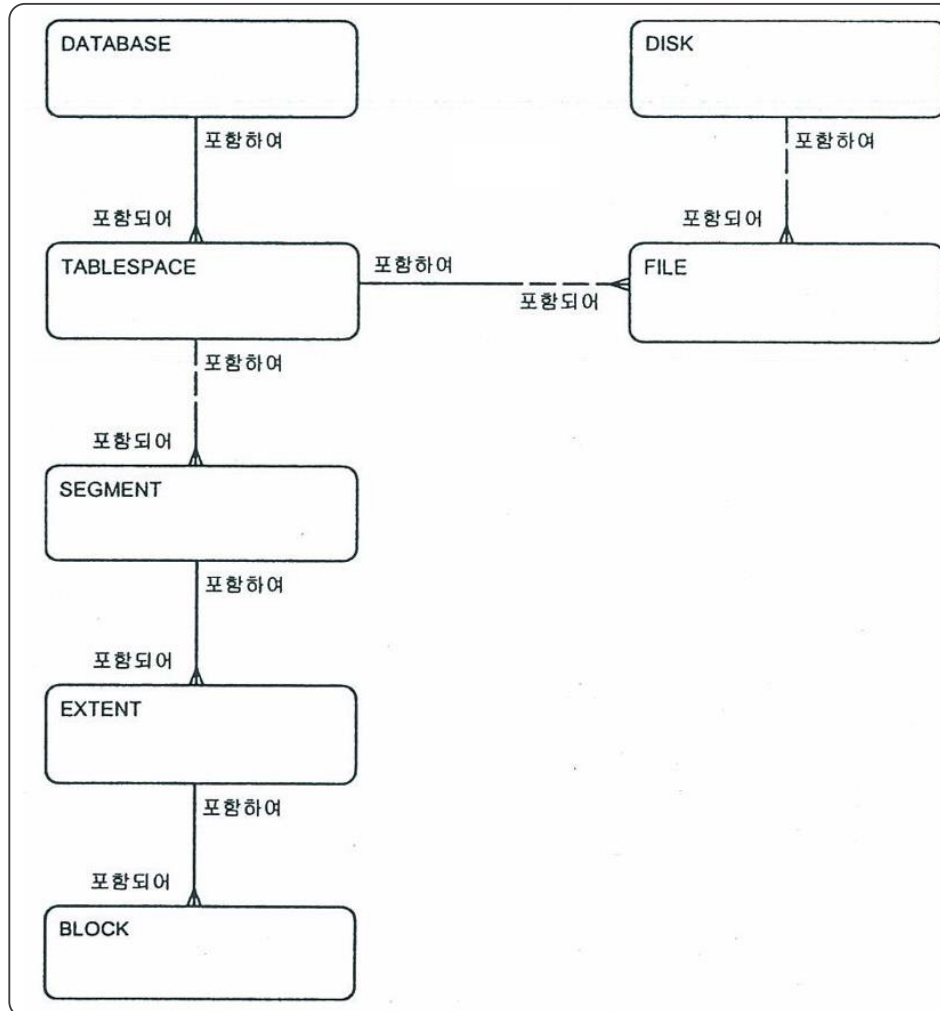
요구 사항

- ◎ 각 사원은 반드시 하나의 부서에 속해야 합니다.
- ◎ 각 부서는 0명 이상의 사원들로 구성이 될 수 있습니다.
- ◎ 각 사원은 0개 이상의 업무에 할당될 수 있습니다.
- ◎ 각 업무는 0명 이상의 사원들에 의하여 수행될 수 있습니다.

ER 구성 모델 요소

연습 문제

문제) 다음 문제에 대하여 ER 모델(엔터티와 Relationship)을 보고 문장으로 표현해 보세요.



연습 문제

문제) 다음 문제에 대하여 ER 모델(엔터티와 Relationship)을 만들어 보세요.

각 엔터티들 사이의 관계의 존재를 확인하기 위해서 Relationship Matrix를 사용하면 좋습니다.

문제 내용

나는 관리 기법으로 운영되는 강사 지도 강좌를 제공하는 교육 기관의 관리자입니다.
우리는 많은 강좌를 가르치고 있으며 각 강좌는 code, 강좌명, 수업료를 가지고 있습니다.
"Python 기초"와 "R 프로그래밍"은 우리의 인기 강좌 중의 하나입니다.
강좌의 수업 일수는 1~4 일간으로 다양합니다.
우리는 각 강사의 이름과 전화 번호를 관리하며, 강좌를 개설하고 강사를 배정합니다.
각 강사들은 여러 개의 강좌를 가르칠 수 있으며,
특히 '김철수'와 '박영희' 강사는 우리의 최고 강사들입니다.
각 강좌는 단 한명의 강사에 의해 진행되며 어떤 강좌는 아직 강사가 배정되지 않을 수도 있습니다.
한 학생이 동시에 여러 강좌를 수강할 수 있으며 많은 학생이 그렇게 합니다.
예를 들어 ABC 전자의 홍길동은 우리가 제공한 모든 강좌를 수강했습니다.
우리는 각 학생의 성명과 전화 번호를 관리하고자 하며,
때로는 학생과 강사가 그들의 전화 번호를 알려 주지 않을 때도 있습니다.

Relationship Matrix

matrix 만들기
상관 관계가 없으면 dashline으로 표시합니다.
세로에 있는 개체(entity)를 주어로 해석하면 됩니다.

	강사	학생	강좌
강사	-		
학생		-	
강좌			-

ER 구성 모델 요소

m:n 관계 분석

M:N(다대다) 관계는 교차 엔터티(Intersection Entity)를 이용하여 해결합니다.

m:n 관계를 가진 엔터티를 살펴 보겠습니다.

m:n 관계는 한 개체가 다른 여러 개체와 관련성을 갖고, 반대로 여러 개체도 다른 개체와 관련성을 갖는 상황을 의미합니다.

이를 '다대다' 관계라고도 합니다.

구현 절차

step 01

새로운 교차 엔터티를 생성합니다.
교차 엔터티는 m:n을 m:1로 만들어 주는 역할을 합니다.

step 02

두개의 m:1 관계를 생성하고 선택 가능성을 유도합니다.

step 03

m:n 관계를 제거하도록 합니다.

풀어 봅시다.

고객 홍길동이 셔츠 2벌과 바지 3벌을 구매하는 예시를 살펴 봅시다.

다음 페이지에 계속...

ER 구성 모델 요소

m:n 관계 분석 01

M:N(다대다) 관계는 교차 엔터티(Intersection Entity)를 이용하여 해결합니다.

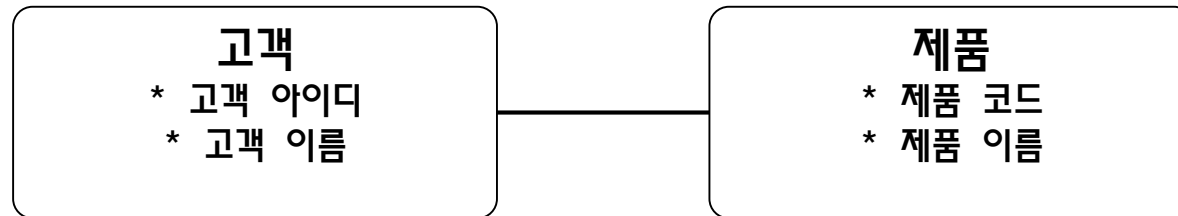
제반 사항

AB 마트의 고객 '홍길동'은 최소 1개 이상의 제품을 구매할 수 있습니다.

진열대에 놓여 있는 '셔츠' 및 '바지' 등은 여러 명의 고객들이 에 구매될 수 있습니다.

고객 홍길동이 셔츠 2벌과 바지 3벌을 구매한다고 가정해 봅시다.

→ 셔츠 2벌/바지 3벌은 어느 엔터티에 저장되어야 할까요?



다음 페이지에 계속...

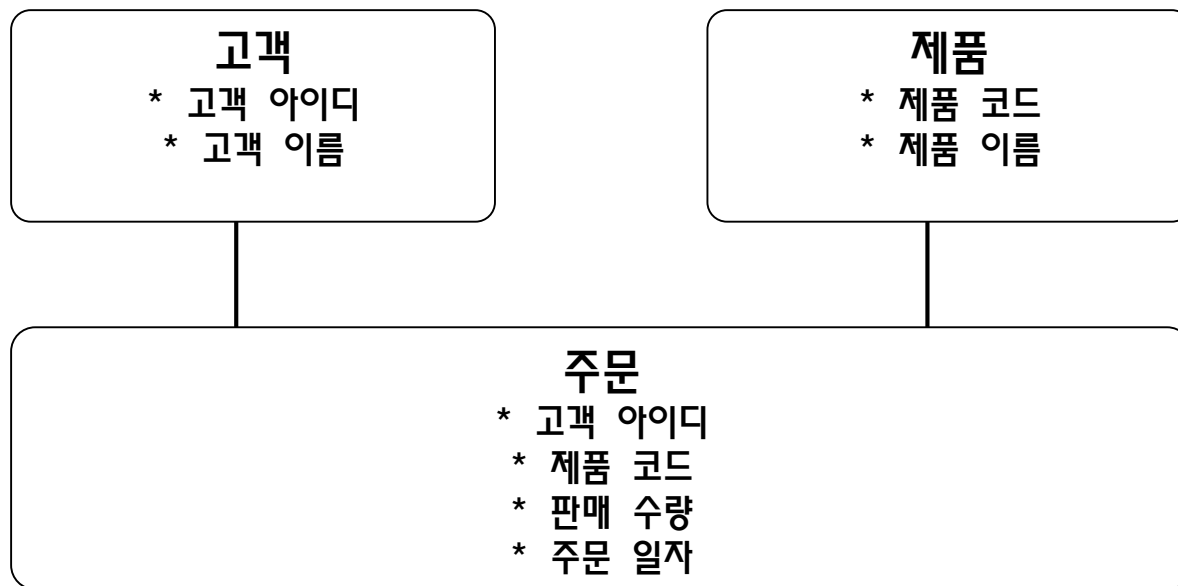
m:n 관계 분석 02

M:N(다대다) 관계는 교차 엔터티(Intersection Entity)를 이용하여 해결합니다.

제반 사항

주문에 대한 청구서 문제

- 단일 청구가 되어 합니다.(셔츠 2벌/바지 3벌)
- [주문]과 [제품]의 새로운 m:n 관계가 성립됩니다.



다음 페이지에 계속...

m:n 관계 분석 03

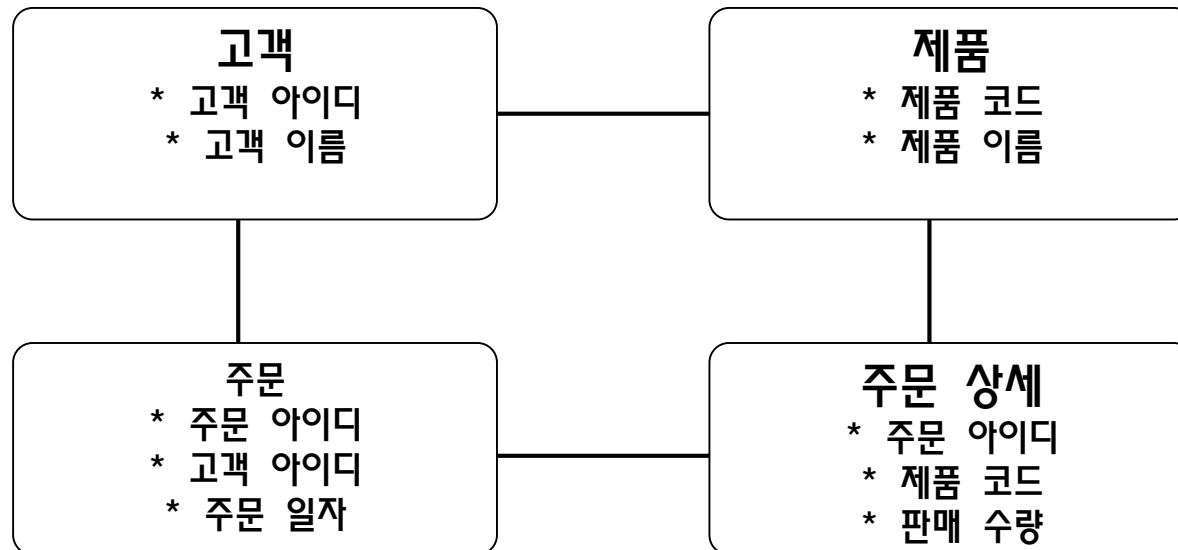
M:N(다대다) 관계는 교차 엔터티(Intersection Entity)를 이용하여 해결합니다.

제반 사항

새로운 엔터티 : 주문 상세

→ [주문]과 [제품]의 m:n 관계를 다시 해석함.

→ [주문 상세]라는 엔터티를 도출합니다.



m:n 관계 분석 04

M:N(다대다) 관계는 교차 엔터티(Intersection Entity)를 이용하여 해결합니다.

제반 사항

새로운 엔터티 : 주문 상세

→ [주문]과 [제품]의 m:n 관계를 다시 해석함.

→ [주문 상세]라는 엔터티를 도출합니다.

고객 테이블

고객 아이디	이름
hong	홍길동
kim9	김구

주문 테이블

주문 아이디	고객 아이디	주문 일자
1	hong	2022/03/01
2	kim9	2022/07/17

제품 테이블

제품 코드	제품 이름
10	바지
20	셔츠

주문 상세 테이블

주문 아이디	제품 코드	판매 수량
1	10	3
1	20	2
2	10	1
2	20	4

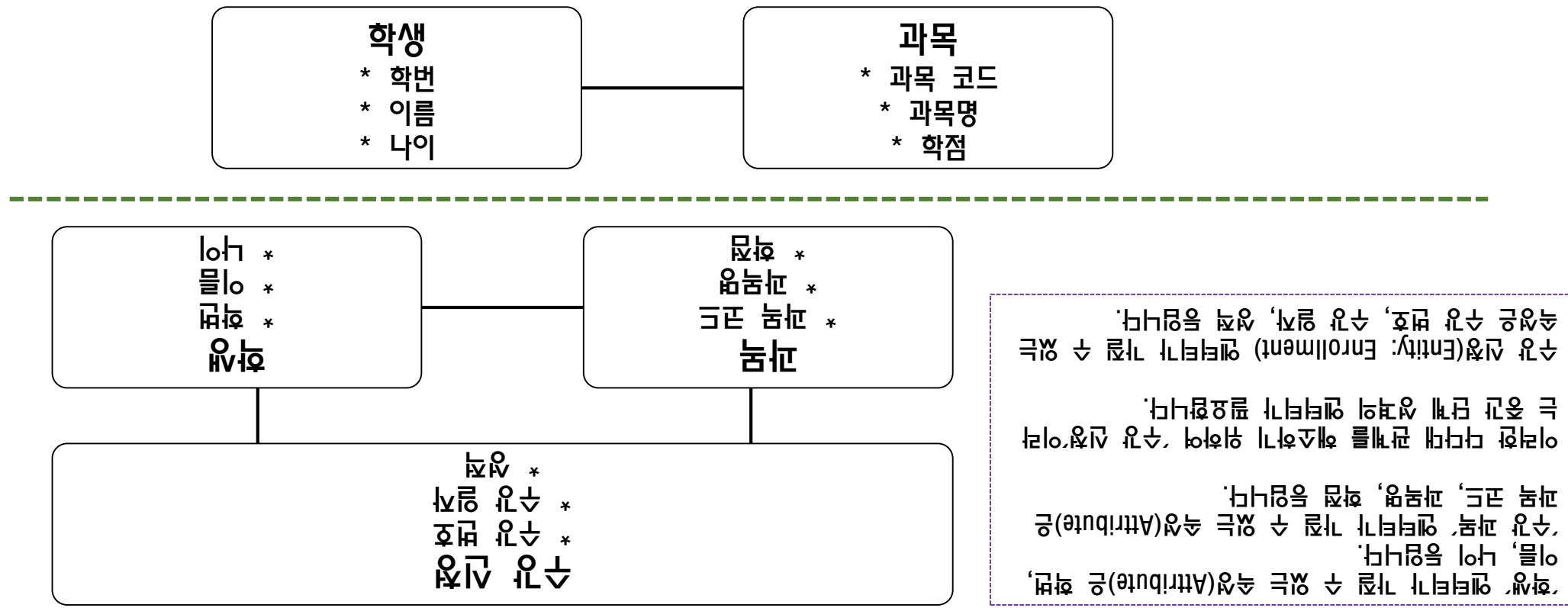
연습 문제

문제05) 다음 m:n 엔터티에 대한 관계를 분해해 보세요.

학생(Student) 1명은 여러 과목(Subject)을 수강할 수 있습니다.

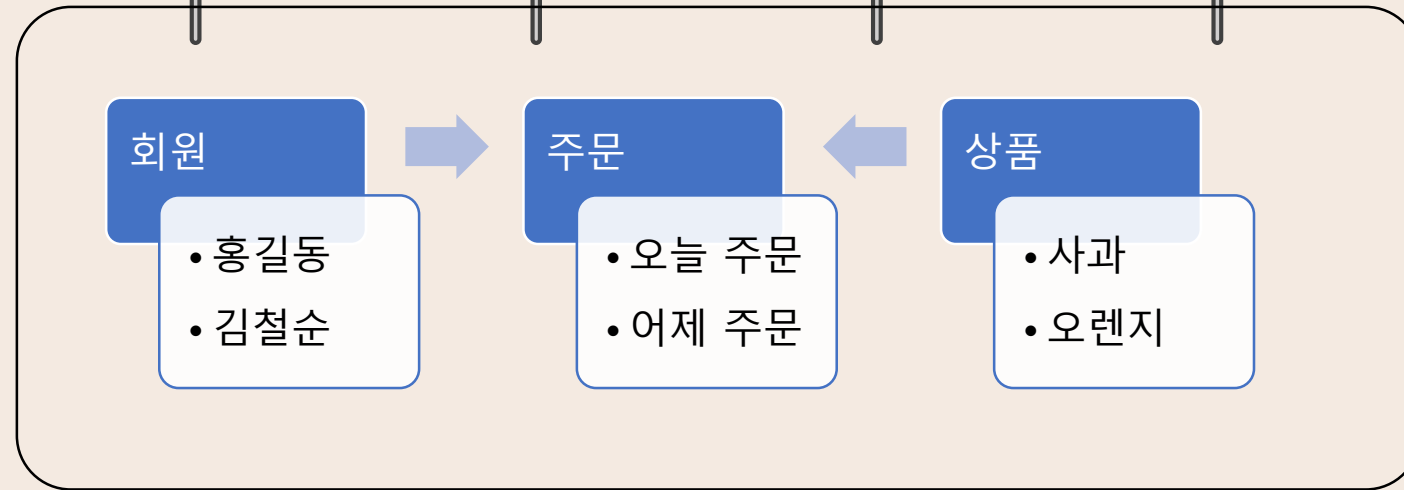
하나의 과목은 여러 학생들이 수강할 수 있습니다.

이러한 학생과 수강 과목 간의 다대대 관계를 분해해 보세요.



Data Modeling

정규화(Normalization)



정규화(Normalization)

테이블에 입력/수정/삭제시 이상 현상을 최소화하기 위하여 좀 더 작은 단위의 테이블로 분해하는 과정입니다.

데이터 이상 현상

- 데이터 집합에서 예상하지 못한, 비정상적인 패턴이나 동작을 나타내는 현상을 의미합니다.
- 이상 현상은 데이터의 오류, 이상치, 누락, 부정확성 등을 포함할 수 있습니다.

정규화를 하는 이유

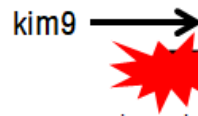
- 데이터를 입력/수정/삭제 등의 작업이 발생하는 **이상 현상을 막을 수** 있습니다.
- 독립된 Entity로 정의함으로써 **높은 응집력**과 **낮은 결함도**를 유지할 수 있습니다.
- 중복된 데이터를 최소화함으로써 **저장 공간을 최소화**할 수 있습니다.
- 좀더 작은 단위의 Entity로 분해함으로써 **재활용성**이 높아집니다.

연관 지식

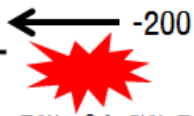
- 함수의 종속성

사원 테이블

아이디	이름	주소	급여
an	안중근	서대문	800
kim9	김구	강남	600
yoon	윤봉길	용산	500



primary key를 위반하고 있습니다.
동일한 아이디는 입력 불가능합니다.



급여는 음수 값이 들어오면 안됩니다.

정규화

데이터를 수학적인 방법에 의하여 구조화시켜 데이터를 재배포 혹은 분해하는 기술을 의미합니다.

항목	설명
정의	<p>중복 되는 데이터를 나누어 여러 개의 테이블에 분산 저장하는 기법입니다.</p> <p>실세계의 데이터를 구조화/체계적으로 재배포/분해하는 방법입니다.</p> <p>데이터 모델을 좀더 개선시키고자 하는 절차에 관련된 이론에 입각합니다.</p>
필요 충분 조건	<p>데이터 분해시 데이터의 손실은 없어야 합니다.(무손실 분해)</p> <p>분해된 테이블들은 조인 연산을 통하여 원래 테이블의 모든 내용을 표현할 수 있어야 합니다.</p>
목적	<p>데이터의 일관성을 보장 받기 위함입니다.</p> <p>중복된 데이터의 최소화로 안정성을 확보하기 위함입니다.</p> <p>다양한 비즈니스 요구 사항에 대한 유연한 대처 능력을 키우기 위함입니다.</p>
기대 효과	<p>중복의 배제로 인한 메모리 적재율이 낮아 집니다.</p> <p>물리적 디스크로의 액세스 빈도 수가 낮아 집니다. ↓.(메모리 활용률 ↑)</p> <p>일관성/정합성 유지에 유리합니다.</p> <p>업무의 생산성에 유리합니다.</p>

함수의 종속성

특정 속성의 모든 값에 대하여 다른 속성의 값이 유일하게 결정이 되게 하는 함수의 속성을 의미합니다.

정규화(normalization)란 중복된 데이터가 최소화될 수 있도록 데이터베이스의 구조를 결정짓는 일을 의미합니다.
이러한 데이터베이스의 정규화 과정에서 함수 종속성이라는 개념은 매우 중요합니다.

사원 Relation

아이디	이름	주민번호	성별	생년월일	나이
kim9	김구	700811-1710789	남자	70/08/25	53
soon	유관순	800815-2026456	여자	80/07/17	43
lee	이순신	900717-1745678	남자	85/08/15	33
maria	조마리아	000811-4665432	여자	90/12/25	23

주민 등록 번호

성별

'주민 등록 번호'로 '성별'을 파악할 수 있습니다.

생년월일

나이

'생년월일'을 알면 '나이'를 가늠할 수 있습니다.

X

Y

'X'를 알면 'Y'를 알 수 있습니다.

어떤 속성 X의 값을 알면 다른 속성 Y의 값이 유일하게 정해지는 의존 관계를 '속성 Y는 속성 X에 종속합니다.(dependent)' 또는 '속성 X는 속성 Y를 결정합니다.(determine)'라고 합니다. 이때 기준 값인 X를 **결정자(Determinant)**라고 하고, 종속이 되는 값 Y를 **종속자(Dependent)**라고 부릅니다.

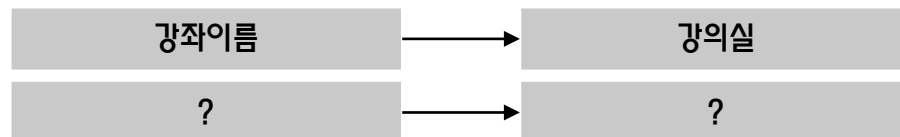
'주민 등록 번호' → '성별'과 같이 **왼쪽 속성의 모든 값에 대하여 오른쪽 속성의 값이 유일하게 결정이 될때 '함수적으로 종속한다'**라고 하고, 이러한 성질을 '함수 종속성'이라고 부릅니다.

함수의 종속성

문제) '학생 수강 성적표'에서 함수의 종속성 관계를 가진 속성들을 찾아 보세요.

학생 수강 성적표

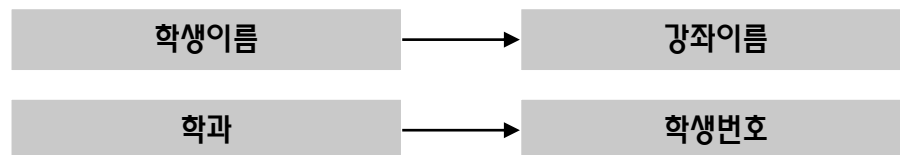
학생번호	이름	주소	학과	사무실	강좌이름	강의실	성적
501	박지성	영국 맨체스터	컴퓨터과	공학관101	스포츠경영학	체육관 103	3.5
401	김연아	대한민국 서울	체육학과	체육관101	데이터베이스	공학관 110	4.0
402	장미란	대한민국 강원도	체육학과	체육관101	데이터베이스	공학관 110	3.5
502	주신수	미국 클리블랜드	컴퓨터과	공학관101	자료 구조	공학관 111	4.0
501	박지성	영국 맨체스터	컴퓨터과	공학관101	자료 구조	공학관 111	3.5



'강의실'은 '강좌 이름'만 알면 알아낼 수 있습니다.

강좌이름 ← 강의실
 성적 ← 강좌이름
 이름 ← 강좌이름

'학생 수강 성적표'에서 다음 항목은 종속성 관계를 가지고 있는 지 파악해 보세요.



학과 ← 학생이름
 성적 ← 학과

함수의 종속성

이행적 함수 종속은 릴레이션 $A \rightarrow B$ 이고, $B \rightarrow C$ 이면 $A \rightarrow C$ 인 관계가 성립하는 관계를 의미합니다.

고객(Customer) Relation

아이디	이벤트 번호	당첨 여부	등급	할인율
bluesky	A	Y	vip	15%
bluesky	E	N	vip	15%
bluesky	G	Y	vip	15%
redfox	B	N	vvip	20%
redfox	E	Y	vvip	20%
yellowcoffee	C	Y	silver	5%
yellowcoffee	F	Y	silver	5%
jangten	D	N	gold	10%

이행적 함수 종속

‘아이디’를 알면 해당 고객의 ‘등급’을 확인할 수 있고, ‘아이디’를 알면 해당 고객의 ‘할인율’도 파악이 됩니다.
그리고, ‘등급’을 알면 해당 고객의 ‘할인율’도 파악이 됩니다.
즉, 이러한 관계를 **이행적 함수 종속** 관계라고 합니다.



데이터의 이상 현상

이상(Anomaly) 현상은 정규화를 함으로써 해결할 수 있습니다.

정규화의 목적은 반복적인 데이터에 대하여 여러 개의 테이블로 분해하여 효율적으로 데이터를 관리하기 위함입니다.

정규화가 이루어지지 않으면 데이터 이상 현상(Anomaly)이 발생하게 되고, 이 현상은 현실 세계의 실제 데이터와 데이터베이스에 저장된 값이 불일치하는 문제를 야기할 수 있습니다.

데이터 이상은 삽입 이상, 업데이트 이상, 삭제 이상이라는 세 가지 유형이 존재합니다.

이상 현상	이름
삽입 이상	데이터 삽입시 필요 없는 속성까지 입력해야 하는 현상을 말합니다.
갱신 이상	데이터 갱신시 필요하지 않는 데이터까지 갱신이 되는 현상을 말합니다. 일반적으로 동일한 데이터에 대하여 2건 이상의 업데이트가 일어나는 경우에 많이 발생합니다.
삭제 이상	데이터 삭제시 보존이 되어야 하는 데이터까지 삭제가 되는 현상을 말합니다.

정규화 예시 01

데이터의 이상 현상

정규화를 수행하기 전에 이상 증상을 파악해야 합니다.

다음 학생들에 대한 정보를 이용하여 이상 증상을 파악해 보도록 하겠습니다.

학생(Student) Relation

동일한 학생인데 데이터 갱신시 일부 데이터만 변경이 되는 문제가 발생합니다.

학번	이름	주민 등록 번호	학과	대학	과목 코드	성적
9021756	신동진	701223-1111111	기계 공학	공대	ABC01	B
9021756	신동진	701223-2222222	기계 공학	공대	ABC02	A
9323789	유남균	801125-1026222	컴퓨터 공학	공대	COM01	C
9323789	유남균	801125-1026222	컴퓨터 공학	공대	COM02	B
9431156	이가희	190717-2026456	가정 의학	의대	MED38	A

학번 '9431156'인 학생이 탈퇴시 학과까지 삭제가 되는 문제가 발생하고 있습니다.

통계학

자연대

신설 학과인 '통계학'과를 입력하고자 할 때 학생 이름도 반드시 입력해야 하는 문제가 발생하고 있습니다.

이상 현상

이상 현상	이름
삽입 이상	'통계학'이라는 신설 학과를 등록하는 경우, 학생 정보와 과목에 대한 정보를 입력해야 하는 문제가 있습니다.
갱신 이상	'신동진' 학생의 주민 등록 번호 변경이 되었는데, 데이터 중복으로 인하여 한 개의 행이 갱신되지 않는 문제가 있습니다. 일반적으로 갱신 이상은 동일한 데이터에 대하여 2건 이상의 업데이트가 일어나는 경우에 많이 발생합니다.
삭제 이상	퇴학된 학생 '이가희'가 삭제될 때, 대학 학과 정보가 동시에 삭제가 되는 문제가 있습니다.

정규화 예시 01

데이터의 이상 현상

이상 현상이 있는 릴레이션을 분해하여 이를 없애 주는 절차를 정규화(Normalization)라고 합니다.

다음 단계를 사용하여 이상 현상을 해결해 보도록 하겠습니다.

항목	이름
제 1 정규형	모든 속성은 원자 값(Atomic Value)/단일 값을 가져야 합니다. 복수의 속성 집단은 별도의 엔터티로 분리하도록 하세요. 즉, 중복되는 행은 별도로 분리해야 합니다.
제 2 정규형	속성은 엔터티의 전체 고유한 식별자에 의하여 종속이 되어야 합니다. (Composite Key) 주식별자에 종속적이지 않는 속성은 분리하도록 합니다.
제 3 정규형	비-UID 속성은 다른 비-UID 속성에 종속될 수 없습니다. 속성에 종속적인 속성의 분리(이행적 함수 종속 제거)
보이스코드 정규화	다수의 주식별자 분리
제 4 정규형	다가 종속(Multi-Value Dependency) 속성 분리
제 5 정규형	결합 종속(Join Dependency)일 경우는 두 개 이상의 N개로 분리

정규화 예시 01

정규화(제 1정규형)

중복되는 행은 별도로 분리해야 합니다.

공통 정보		학생 정보			학과 정보	
학번	이름	주민 등록 번호	학과	대학	과목 코드	성적
9021756	신동진	701223-1111111	기계 공학	공대	ABC01	B
9021756	신동진	701223-1111111	기계 공학	공대	ABC02	A
9323789	유남균	801125-1026222	컴퓨터 공학	공대	COM01	C
9323789	유남균	801125-1026222	컴퓨터 공학	공대	COM02	B
9431156	이가희	190717-2026456	가정 의학	의대	MED38	A

이상 현상 분석

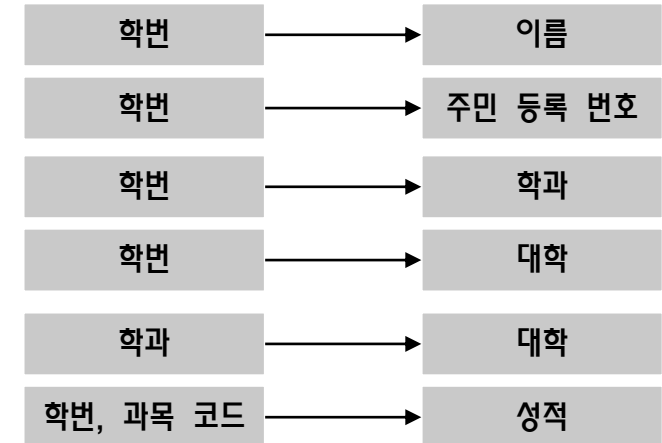
학생의 [성적]을 알려면 반드시 [학번]과 [과목 코드] 정보를 알고 있어야 하는 문제가 있습니다.

따라서, 기본 키는 {학번, 과목 코드}가 되어야 합니다.

모든 칼럼들은 크게 '학생' 정보와 '과목' 정보로 분류되므로, '학생'과 '수강 과목' 테이블로 분리가 되어야 합니다.

이상 현상들이 발생할 가능성이 존재합니다.

함수의 종속성



정규화 예시 01

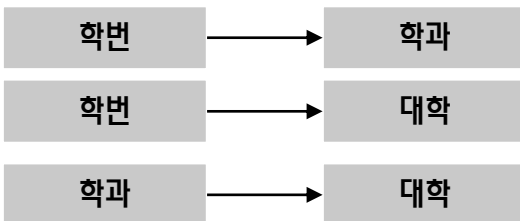
정규화(제 2정규형)

기본 키가 아닌 모든 속성들은 기본 키에 완전히 종속적이어야 합니다.

학생 Table

학번	이름	주민 등록 번호	학과	대학
9021756	신동진	701223-1111111	기계 공학	공대
9323789	유남균	801125-1026222	컴퓨터 공학	공대
9323789	유남균	801125-1026222	컴퓨터 공학	공대
9431156	이가희	190717-2026456	가정 의학	의대

함수의 종속성



이행적 함수 종속



수강 과목 Table

학번	과목 코드	성적
9021756	ABC01	B
9021756	ABC02	A
9323789	COM01	C
9323789	COM02	B
9431156	MED38	A

학번, 과목 코드 → 성적

이상 현상 분석

수강 과목 테이블에서 기본 키는 {학번, 과목 코드}이므로 성적은 기본 키에 함수적 종속성을 완전히 충족하므로 문제가 없습니다. 하지만 학생 테이블을 살펴 보면 [대학] 칼럼은 [학번] 칼럼에 완전 종속적임과 동시에 [학과] 칼럼에 대해서도 완전 종속적입니다. 이러한 현상을 "이행적 함수 종속"이라고 하는데, 이것 또한 이상 현상을 발생시킬 수 있습니다. 이행적 함수 종속일때 $X \rightarrow Y$ 이고, $Y \rightarrow Z$ 이면 $X \rightarrow Z$ 인 관계가 성립합니다.

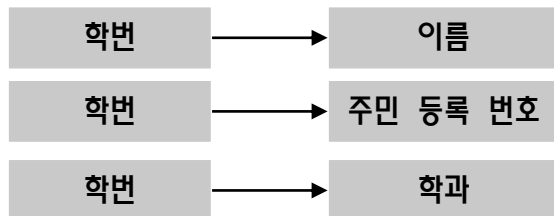
정규화 예시 01

정규화(제 3정규형)

기본 키가 아닌 모든 속성들은 기본 키에 이행적 함수 종속이 아니어야 합니다.

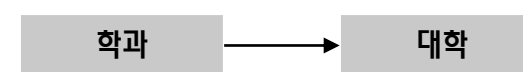
학생 Table

학번	이름	주민 등록 번호	학과
9021756	신동진	701223-1111111	기계 공학
9323789	유남균	801125-1026222	컴퓨터 공학
9431156	이가희	190717-2026456	가정 의학



학과 Table

학과	대학
기계 공학	공대
컴퓨터 공학	공대
가정 의학	의대



이상 현상 분석

학생 테이블에 존재하던 "이행적 함수 종속" 관계를 제거하여 학생 테이블과 학과 테이블로 분리했습니다.

학생 테이블은 다른 모든 속성들이 기본 키인 학번에 완전히 함수적으로 종속이 되고 있음을 확인할 수 있습니다.

정규화 예시 02

데이터의 이상 현상

다음은 고객들의 주문 현황에 대하여 이상 증상을 파악해 보도록 하겠습니다.

주문 Relation

주문 번호	고객명	제품 번호	제품 설명	수량	고객 주소	주문 날짜
1001	신동진	P01	청바지	10	마포 공덕	2022/03/01
1001	신동진	P02	T셔츠	20	마포 공덕	2022/03/01
1002	유남균	P01	청바지	30	용산 용문	2022/07/17
1002	유남균	P03	블라우스	40	용산 용문	2022/07/17
1003	신동진	P03	블라우스	50	마포 공덕	2022/08/15
1003	신동진	P04	선글라스	60	마포 공덕	2022/08/15

이상 현상

이상 현상	이름
삽입 이상	고객이 신규 주문을 하고자 하는 경우, 고객의 주소지를 다시 등록해야 하는 문제가 있습니다.
갱신 이상	고객 '신동진'이 이사를 하는 경우, 고객의 주소를 4군데를 갱신해야 하는 문제가 있습니다.
삭제 이상	고객 '유남균'이 탈퇴를 하고자 할 때, 기존 주문 정보가 동시에 삭제가 되는 문제가 있습니다.

정규화 예시 02

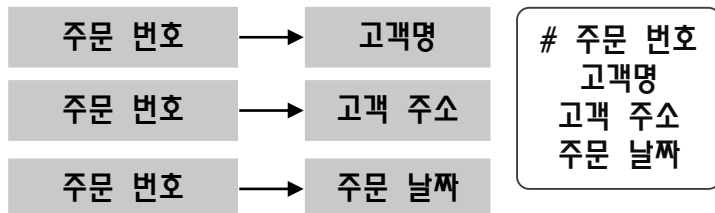
정규화

고객들의 주문 현황에 대하여 정규화를 수행해 보겠습니다.

주문 Table

주문 번호	고객명	고객 주소	주문 날짜
1001	신동진	마포 공덕	2022/03/01
1002	유남균	용산 용문	2022/07/17
1003	신동진	마포 공덕	2022/08/15

함수의 종속성



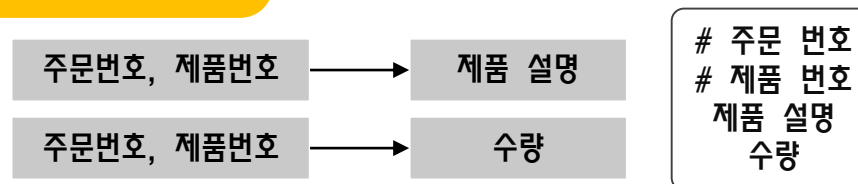
이상 현상 분석

주문 테이블에서 속성 '고객명', '고객 주소', '주문 날짜'는 '주문 번호'에 종속적이고, 주문 상세 테이블에서 속성 '제품 설명'과 '수량'은 {주문 번호, '제품 번호'}에 종속적입니다. 주문 테이블에서 '고객명' 데이터에 중복된 행이 보이고 있고, 주문 상세 테이블 역시 '제품 번호' 속성에 중복된 행이 보이고 있습니다. 이것은 이상 현상을 일으킬 수 있습니다.

주문 상세 Table

주문 번호	제품 번호	제품 설명	수량
1001	P01	청바지	10
1001	P02	T셔츠	20
1002	P01	청바지	30
1002	P03	블라우스	40
1003	P03	블라우스	50
1003	P04	선글라스	60

함수의 종속성



정규화 예시 02

정규화

고객들의 주문 현황에 대하여 정규화를 수행해 보겠습니다.

주문 Table

주문 번호	고객명	고객 주소	주문 날짜
1001	신동진	마포 공덕	2022/03/01
1002	유남균	용산 용문	2022/07/17
1003	신동진	마포 공덕	2022/08/15

주문 번호
고객명
고객 주소
주문 날짜

이상 현상 분석

주문 상세 테이블의 '제품 번호', '제품 설명' 속성을 별도로 제품 테이블로 분리해 냈습니다.

주문 상세 Table

주문 번호	제품 번호	수량
1001	P01	10
1001	P02	20
1002	P01	30
1002	P03	40
1003	P03	50
1003	P04	60

주문번호, 제품번호 → 제품 설명

주문 번호
제품 번호
수량

제품 Table

제품 번호	제품 설명
P01	청바지
P02	T셔츠
P03	블라우스
P04	선글라스

제품 번호 → 제품 설명

제품 번호
제품 설명

정규화 예시 02

정규화

고객들의 주문 현황에 대하여 정규화를 수행해 보겠습니다.

고객 Table

고객명	고객 주소
신동진	마포 공덕
유남균	용산 용문

고객명
고객 주소

주문 Table

주문 번호	고객명	주문 날짜
1001	신동진	2022/03/01
1002	유남균	2022/07/17
1003	신동진	2022/08/15

주문 번호
고객명
주문 날짜

주문 상세 Table

주문 번호	제품 번호	수량
1001	P01	10
1001	P02	20
1002	P01	30
1002	P03	40
1003	P03	50
1003	P04	60

주문 번호
제품 번호
수량

제품 Table

제품 번호	제품 설명
P01	청바지
P02	T셔츠
P03	블라우스
P04	선글라스

제품 번호
제품 설명

이상 현상 분석

주문 테이블의 '고객명', '고객 주소' 속성을 별도로 고객 테이블로 분리해 냈습니다.

정규화 예시 02

정규화

정규화 결과를 이용한 ERD입니다.

