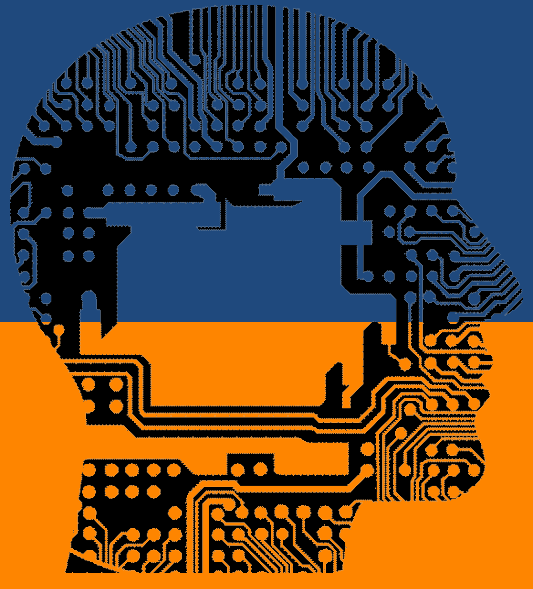
한국IT진흥부설 정보보호교육학원 아이섹

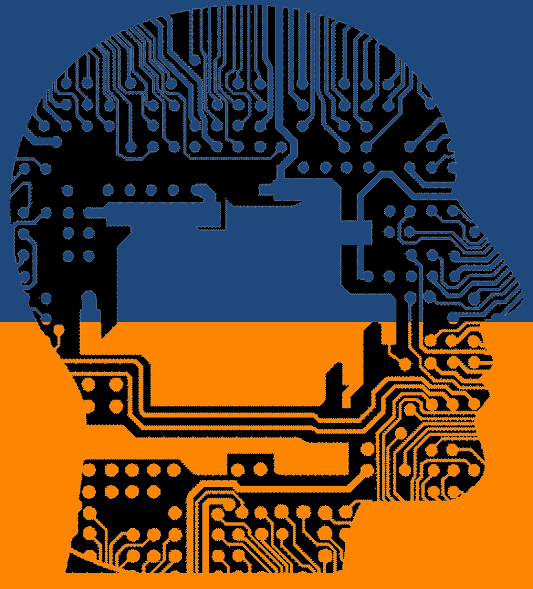


JAVA 웹 개발자 양성과정
DataBase - SQLD

1강 - 데이터 모델링의 이해

By SoonGu Hong

한국IT진흥부설 정보보호교육학원 아이섹



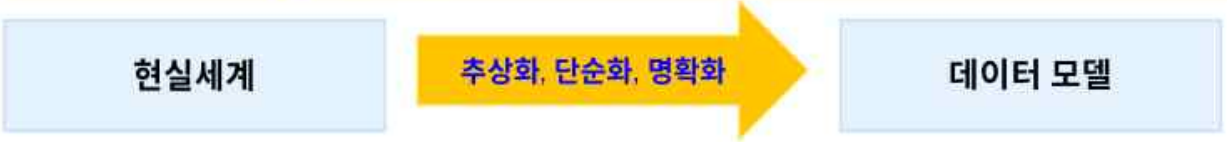
JAVA 웹 개발자 양성과정

DataBase

1. 데이터 모델의 이해

➢ 모델링의 정의

① 복잡한 "현실세계"를 단순화시켜 표현하는 것
② 사물 또는 사건에 관한 양상(Aspect)이나 관점(Perspective)을 연관된 사람이나 그룹을 위하여 명확하게 하는 것
③ 현실 세계의 추상화된 반영



❖복잡한 현실세계를 일정한 표기법에 의해 표현하는 일

➢ 모델링의 특징

① 추상화(모형화, 가설적)는 현실세계를 일정한 형식에 맞추어 표현을 한다는 의미
② 단순화는 복잡한 현실세계를 약속된 규약에 의해 제한된 표기법이나 언어로 표현
③ 명확화는 누구나 이해하기 쉽게 하기 위해 대상에 대한 애매모호함을 제거하고 정확(正確)하게 현상을 기술

❖모델링을 다시 정의하면 "현실세계를 추상화, 단순화, 명확화 하기 위해 일정한 표기법에 의해 표현하는 기법"으로 정리 할 수 있음

➢ 모델링의 세 가지 관점

➢ 데이터 모델링의 정의

① 정보 시스템을 구축하기 위한 **데이터 관점**의 업무 분석 기법
② 현실세계의 데이터에 대해 **약속된 표기법에 의해 표현**되는 과정
③ 데이터베이스를 구축하기 위한 **분석/설계**의 과정

➢ 데이터 모델이 제공하는 기능

① 시스템을 현재 또는 원하는 모습으로 **가시화**
② 시스템의 구조와 행동을 **명세화**
③ 시스템을 구축하는 **구조화된 틀**을 제공
④ 시스템을 구축하는 과정에서 결정한 것을 **문서화**
⑤ 다양한 영역에 집중하기 위해 다른 영역의 세부 사항은 숨기는 **다양한 관점**을 제공
⑥ 특정 목표에 따라 **구체화**된 상세 수준의 표현방법을 제공

➢ 데이터 모델링의 중요성 및 유의점

중요성	설명
파급효과(Leverage)	- 시스템 구축 작업 중에서 다른 어떤 설계 과정보다 데이터 설계가 중요함
복잡한 정보 요구사항의 **간결한 표현**(Conciseness)	- 데이터 모델은 구축할 시스템의 정보 요구사항과 한계를 가장 명확하고 간결하게 표현할 수 있는 도구
데이터 품질(Data Quality)	- 데이터의 중복, 비 유연성, 비 일관성이 발생할 수 있음

➢ 데이터 모델링의 3단계 진행

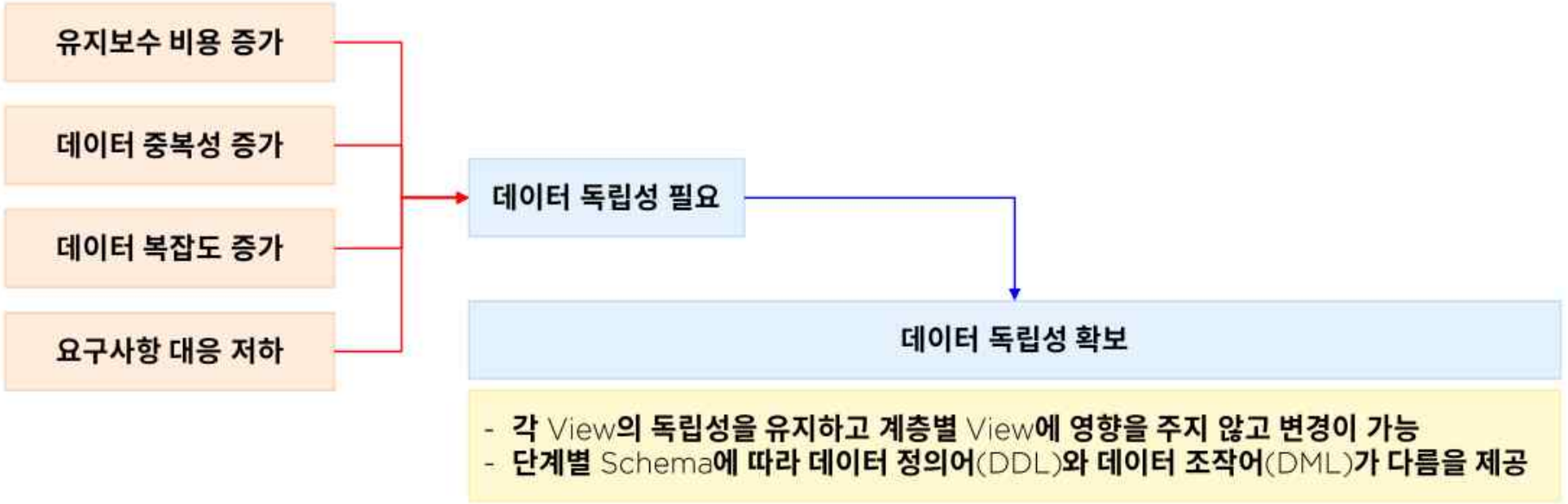
단계 명	설명
개념적 데이터 모델링	- 추상화 수준이 높고 업무 중심적이고 포괄적인 수준의 모델링 진행. 전사적 데이터 모델링, EA수립 시 많이 사용
논리적 데이터 모델링	- 시스템으로 구축하고자 하는 업무에 대해 Key, 속성, 관계 등을 정확하게 표현, 재 사용성이 높음
물리적 데이터 모델링	- 실제로 데이터베이스에 이식할 수 있도록 성능, 저장 등 물리적인 성격을 고려하여 설계

➢ 프로젝트 생명주기(Life Cycle)에서 데이터 모델링

① 프로젝트 생명 주기는 정보전략계획 -> 분석 -> 설계 -> 개발 -> 테스트 -> 전환/이행 단계가 있음
② 정보전략계획/분석 단계 : 개념적 데이터 모델링
③ 분석 단계 : 논리적 데이터 모델링
④ 설계 단계 : 물리적 데이터 모델링

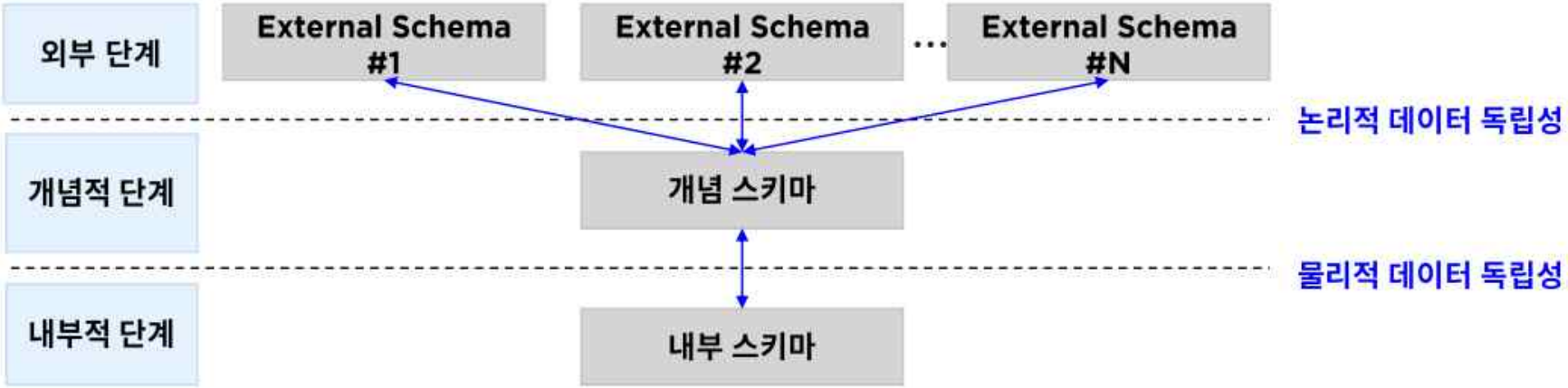
➢ 데이터독립성의 필요성

① 지속적으로 증가하는 유지보수 비용을 절감하고 데이터 복잡도를 낮추며 **중복된 데이터를 줄이기 위한 목적**이 있음
② 끊임없이 요구되는 사용자 요구사항에 대해 화면과 데이터베이스 간에 **서로 독립성을 유지하기 위한 목적**으로 데이터 독립성 개념이 출현



➢ 데이터베이스 3단계 구조

① ANSI/SPARC의 3단계 구성의 데이터독립성 모델은 외부 단계와 개념적 단계, 내부적 단계로 구성된 서로 간섭 되지 않는 모델을 제시하고 있다.



단계 명	설명	비고
외부 스키마 (External Schema)	- View 단계 여러 개의 사용자 관점으로 구성, 즉 개개 사용자 단계로서 개개 사용자가 보는 개인적 DB 스키마 - DB의 개개 사용자나 응용프로그래머가 접근하는 DB 정의	- 사용자 관점
개념 스키마 (Conceptual) Schema	- 개념 단계 하나의 개념적 스키마로 구성 모든 사용자 관점을 통합한 조직 전체의 DB를 기술하는 것 - 모든 응용시스템들이나 사용자들이 필요로 하는 데이터를 통합한 조직 전체의 DB를 기술한 것으로 DB에 저장되는 데이터와 그들간의 관계를 표현하는 스키마	- 통합 관점
내부 스키마 (Internal Schema)	- 내부 단계, 내부 스키마로 구성, DB가 물리적으로 저장된 형식 - 물리적 장치에서 데이터가 실제적으로 저장되는 방법을 표현하는 스키마	- 물리적 관점

➢ 데이터베이스 3단계 구조에서의 데이터 독립성 2가지

독립성	설명	비고
논리적 독립성	- 개념 스키마가 변경되어도 외부 스키마에는 영향을 미치지 않도록 지원하는 것 - 논리적 구조가 변경되어도 응용 프로그램에 영향 없음	- 사용자 특성에 맞는 변경 가능 - 통합 구조 변경 가능
물리적 독립성	- 내부 스키마가 변경되어도 외부/개념 스키마는 영향을 받지 않도록 지원하는 것 - 저장 장치의 구조 변경은 응용프로그램과 개념 스키마에 영향 없음	- 물리적 구조 영향 없이 개념 구조 변경 가능 - 개념 구조 영향 없이 물리적인 구조 변경 가능

➢ 데이터베이스 3단계 구조에서의 사상(매핑) 2가지

독립성	설명	비고
외부적/개념적 사상 (논리적 사상)	- 외부적 뷰와 개념적 뷰의 상호 호환성을 정의함	- 사용자가 접근하는 형식에 따라 다른 타입의 필드를 가질 수 있음 - 개념적 뷰의 필드 타입은 변화가 없음
개념적/내부적 사상 (물리적 사상)	- 개념적 뷰와 저장된 데이터베이스의 상호 관련성 정의	- 만약 저장된 데이터베이스 구조가 바뀐다면 개념적/내부적 사상이 바뀌어야 함 그래야 개념 스키마가 그대로 남아있게 됨

➢ 데이터 모델링의 세 가지 요소

① 업무가 관여하는 어떤 것(Things)
② 어떤 것이 가지는 성격(Attributes)
③ 업무가 관여하는 어떤 것 간의 관계(Relationships)

주문 — 사용자

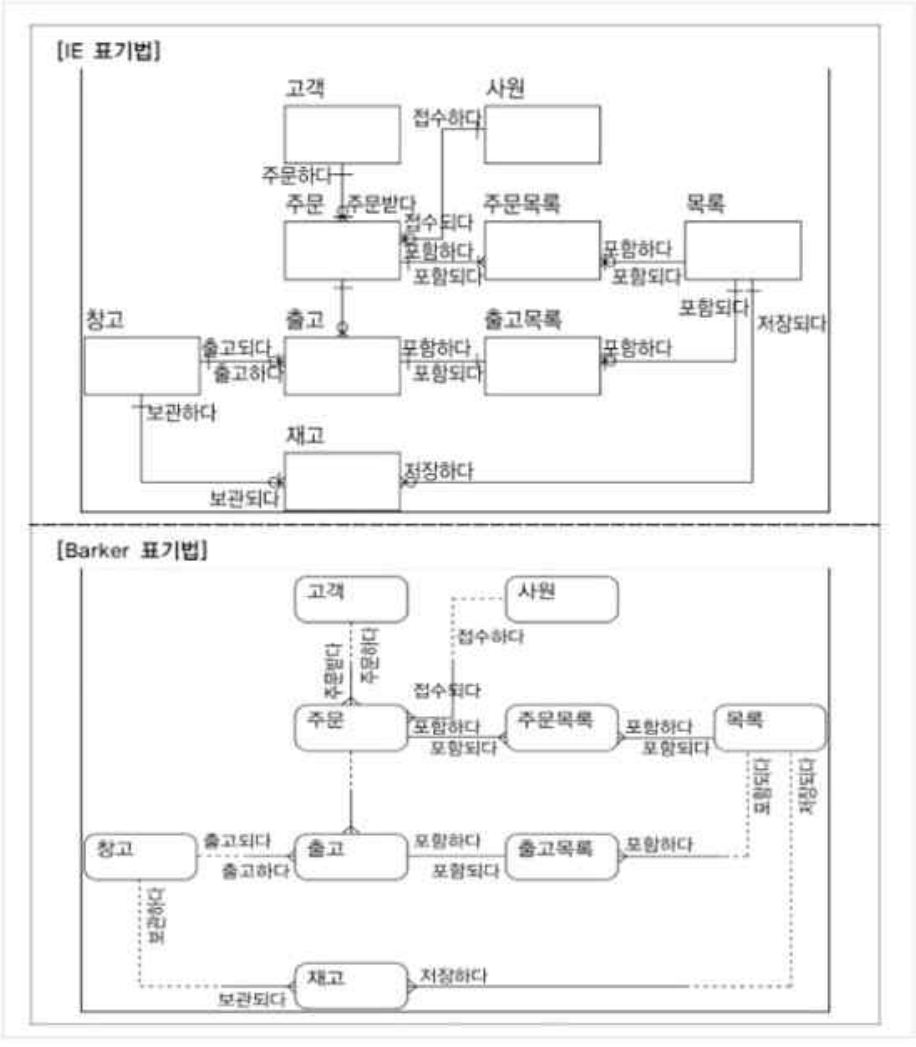
➢ 데이터 모델링 용어

개념	복수/집합 개념 타입/클래스	개별/단수 개념 어커런스/인스턴스
어떤 것 (Thing)	엔터티 타입(Entity Type)	엔터티(Entity)
	엔터티(Entity)	인스턴스(Instance) 어커런스(Occurrence)
어떤 것 간의 연관 (Association between Things)	관계(Relationship)	패어링(Pairing)
어떤 것의 성격 (Characteristic of a Thing)	속성(Attribute)	속성값(Attribute Value)

➢ 데이터 모델링 작업 순서

① 엔터티를 그린다.
② 엔터티를 적절하게 배치한다.
③ 엔터티간 관계를 설정한다.
④ 관계명을 기술한다.
⑤ 관계의 참여도를 기술한다.
⑥ 관계의 필수 여부를 기술한다.

읽어볼것



➢ 데이터 모델링의 이해관계자

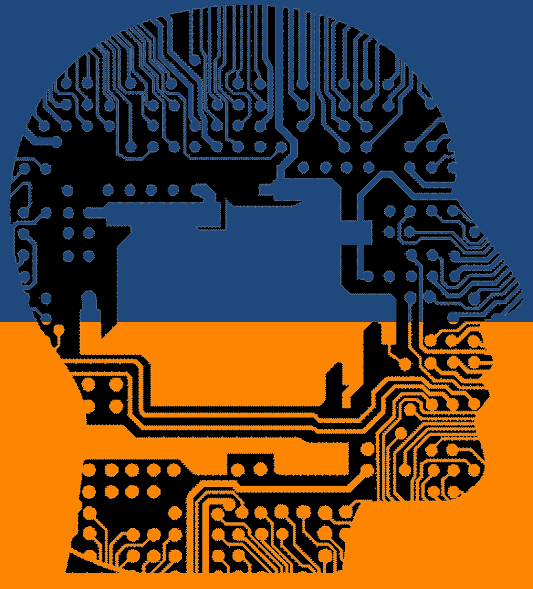
❖ 정보시스템을 구축하는 모든 사람(전문적으로 코딩 만하는 사람 포함)은 데이터 모델링도 전문적으로 할 수 있거나 적어도 완성된 모델을 정확하게 해석할 수 있어야 한다. 즉 프로젝트에 참여한 모든 IT기술자들은 데이터 모델링에 대해 정확하게 알고 있어야 한다는 것을 의미한다.

❖ IT기술에 종사하거나 전공하지 않았더라도 해당 업무에서 정보화를 추진하는 위치에 있는 사람도 데이터 모델링에 대한 개념 및 세부사항에 대해 어느 정도 지식을 가지고 있어야 한다.

➢ 좋은 데이터 모델의 요소

요소	설명
완전성	- 업무에 필요한 데이터가 모두 정의되어야 함
중복 배제	- 동일한 사실은 한번만 저장 해야함
업무 규칙	- 데이터 모델 분석만으로도 비즈니스 로직이 이해되어야 함
데이터 재사용	- 데이터 통합성과 독립성 고려해야함
의사소통	- 데이터 모델을 보고 이해 당사자들끼리 의사소통이 이루어져야 함
통합성	- 동일한 데이터는 유일하게 정의해서 다른 영역에서 참조해야 함

한국IT진흥부설
정보보호교육학원 아이섹



JAVA 웹 개발자 양성과정

DataBase

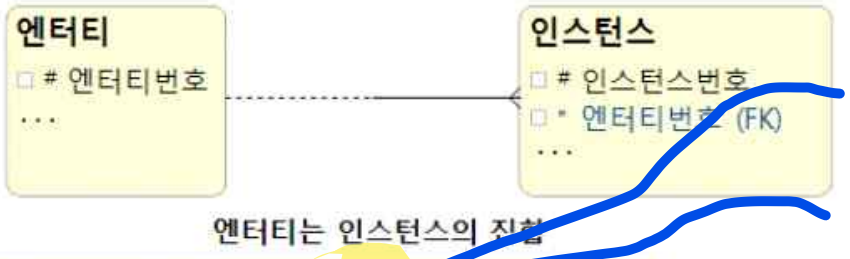
2. 엔터티

➢ 엔터티의 개념 → 테이블 개념

① 엔터티는 사람, 장소, 물건, 사건, 개념 등의 명사에 해당한다.
② 엔터티는 업무상 관리가 필요한 관심사에 해당한다.
③ 엔터티는 저장이 되기 위한 어떤 것(Thing)이다.

❖엔터티란 "업무에 필요하고 유용한 정보를 저장하고 관리하기 위한 집합적인 것(Thing)"으로 설명할 수 있다.

➢ 엔터티와 인스턴스



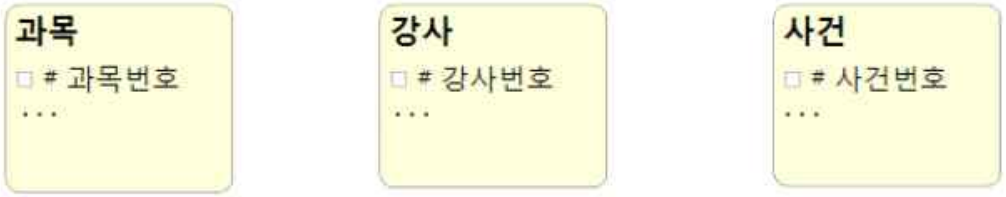
엔터티는 인스턴스의 집합

❖엔터티는 인스턴스의 집합이라고 할 수 있다.

엔터티	인스턴스
과목	수학
	영어
강사	이춘식
	조시형
사건	2010-001
	2010-002

- 엔터티 : 과목, 강사, 사건
- 과목 : 수학, 영어
- 강사 : 이춘식, 조시형
- 사건 : 2010-001, 2010-002

➢ 엔터티 표기법



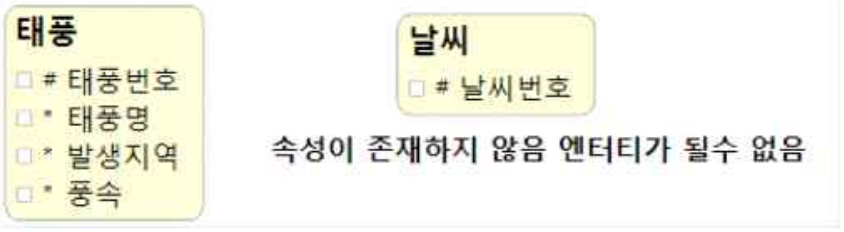
➢ 엔터티의 특징

① 반드시 해당 업무에서 필요하고 관리하고자 하는 정보이어야 한다.(예. 환자, 토익의 응시횟수, …)
② 유일한 식별자에 의해 식별이 가능해야 한다.
③ 영속적으로 존재하는 인스턴스의 집합 이어야 한다.('한개'가 아니라 '두 개 이상')
④ 엔터티는 업무 프로세스에 의해 이용되어야 한다.
⑤ 엔터티는 반드시 속성이 있어야 한다.
⑥ 엔터티는 다른 엔터티와 최소 한 개 이상의 관계가 있어야 한다.

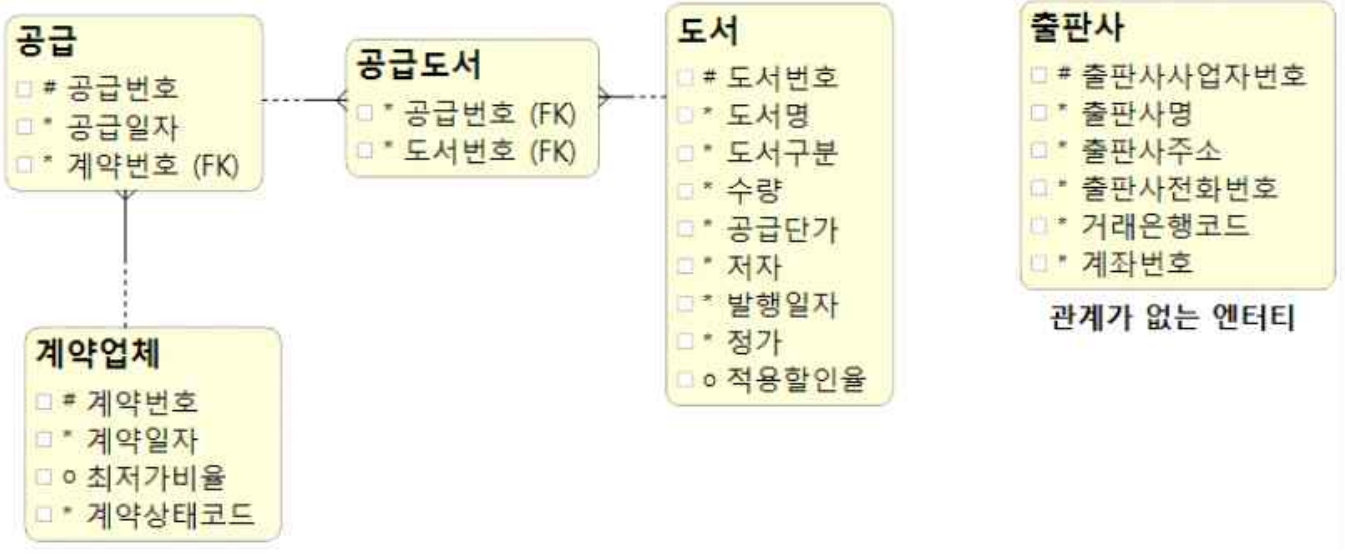
특징	설명
업무에서 필요로 하는 정보	- 반드시 시스템을 구축하고자 하는 업무에서 필요로 하고 관리하고자 하는 정보 여야 한다 - (EX. 환자라는 엔터티는 병원에서는 반드시 필요, 일반회사에서는 필요하지 않을 수 있음)
식별이 가능해야 함	- 인스턴스 각각을 구분하기 위해 유일한 식별자가 존재 해야함
인스턴스의 집합	- 엔터티의 특징 중 "한 개"가 아니라 "두 개 이상"이라는 집합개념은 매우 중요한 개념 - 하나의 엔터티는 여러 개의 인스턴스를 포함한다. (EX. 사원은 여러 명, 회사는 여러 개)
업무프로세스에 의해 이용	- 업무프로세스(Business Process)가 그 엔터티를 반드시 이용해야 한다 - 업무프로세스에 의해 CREATE, READ, UPDATE, DELETE 등이 발생하지 않는 고립된 엔터티의 경우는 엔터티를 제거하거나 아니면 누락된 프로세스가 존재하는지 살펴보고 해당 프로세스를 추가해야 함
속성을 포함	- 엔터티에는 반드시 속성(Attributes)이 포함되어야 한다 - 속성을 포함하지 않고 엔터티의 이름만 가지고 있는 경우는 관계가 생략되어 있거나 업무 분석이 미진하여 속성정보가 누락되는 경우에 해당
관계의 존재	- 엔터티는 다른 엔터티와 최소 한 개 이상의 관계가 존재해야 한다.

➢ 엔터티의 특징 - 계속

▪ **엔터티에는 반드시 속성(Attributes)이 포함되어야 한다.**



▪ **엔터티는 다른 엔터티와 최소 한 개 이상의 관계가 존재해야 한다.**



➢ 엔터티의 분류

유무(有無)형에 따른 분류		
유형	사원, 물품, 강사	- 물리적인 형태가 있고 안정적이며 지속적으로 활용되는 엔터티로 업무로부터 엔터티를 구분하기가 가장 용이하다
개념	조직, 보험상품	- 물리적인 형태는 존재하지 않고 관리해야 할 개념적 정보로 구분이 되는 엔터티
사건	주문, 청구, 미납	- 업무를 수행함에 따라 발생되는 엔터티로서 비교적 발생량이 많으며 각종 통계자료에 이용될 수 있다

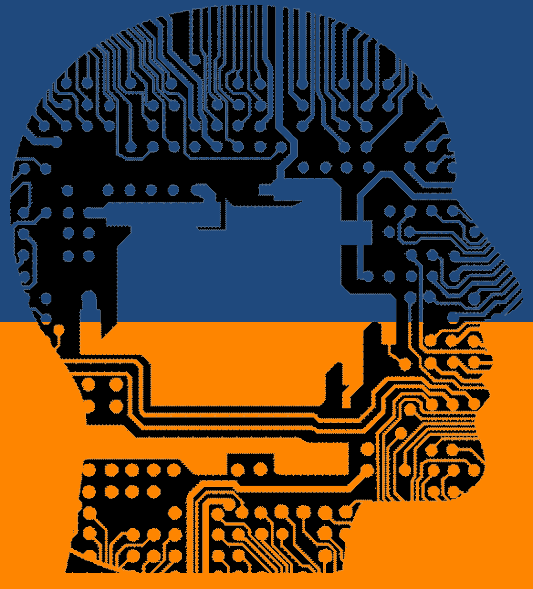
발생시점(發生時點)에 따른 분류		
기본	사원, 부서, 고객, 상품, 자재	- 업무에 원래 존재하는 정보로서 다른 엔터티와 관계에 의해 생성되지 않고 독립적으로 생성이 가능하고 자신은 타 엔터티의 부모의 역할을 하게 된다
중심	계약, 사고, 예금원장, 청구, 주문, 매출	- 본엔터티로부터 발생되고 그 업무에 있어서 중심적인 역할을 한다. - 데이터의 양이 많이 발생되고 다른 엔터티와의 관계를 통해 많은 행위엔터티를 생성
행위	주문목록, 사원변경이력	- 두 개 이상의 부모엔터티로부터 발생되고 자주 내용이 바뀌거나 데이터량이 증가 - 상세 설계단계나 프로세스와 상관모델링을 진행하면서 도출

➢ 엔터티의 명명

① 가능하면 **현업업무**에서 사용하는 용어를 사용한다.
② 가능하면 **약어를 사용하지 않는다.**
③ **단수 명사를** 사용한다.
④ 모든 엔터티에서 **유일하게** 이름이 부여되어야 한다.
⑤ **엔터티 생성 의미대로** 이름을 부여한다.

테이블 → tbl_emp

한국IT진흥부설
정보보호교육학원 아이섹



JAVA 웹 개발자 양성과정

DataBase

3. 속성

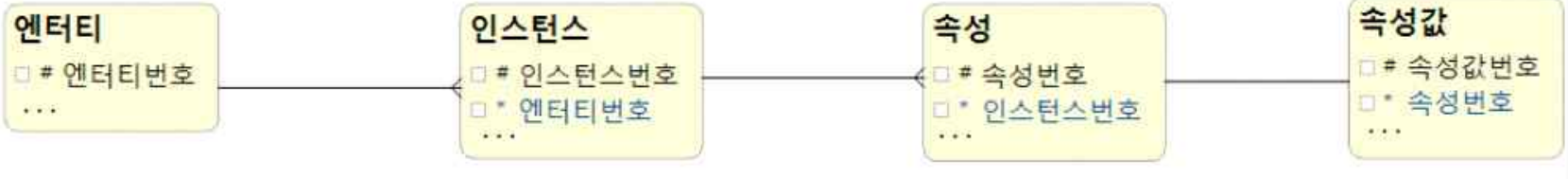
➢ 속성 (Attribute)의 개념

① 업무에서 필요로 한다.
② 의미상 더 이상 분리되지 않는다.
③ 엔터티를 설명하고 인스턴스의 구성요소가 된다.

❖ 속성은 업무에서 필요로 하는 인스턴스에서 관리하고자 하는 의미상 더 이상 분리 되지 않는 최소의 데이터 단위

➢ 엔터티, 인스턴스, 속성, 속성값의 관계

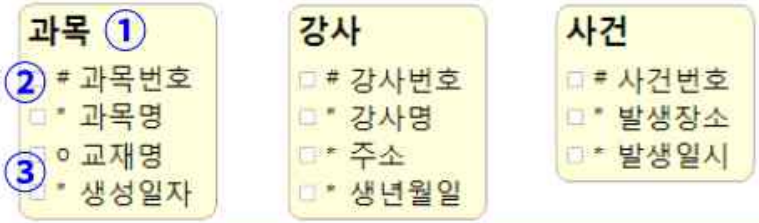
① 한 개의 엔터티는 두 개 이상의 인스턴스의 집합 이어야 한다.
② 한 개의 엔터티는 두 개 이상의 속성을 갖는다.
③ 한 개의 속성은 한 개의 속성값을 갖는다.



엔터티/인스턴스/속성/속성값의 관계

❖ 속성은 엔터티에 속한 엔터티에 대한 자세하고 구체적인 정보를 나타내며 각각의 속성은 구체적인 값을 갖게 된다.
❖ 이름, 주소, 생년월일과 같은 각각의 값을 대표하는 이름들을 속성이라 하고 홍길동, 서울시 강서구, 1967년 12월 31일과 같이 각각의 이름에 대한 구체적인 값을 속성 값(VALUE)이라고 한다.

➢ 속성의 표기법



① 속성명을 기재하고
② 해당 속성이 식별자인지 아닌지 표시하고
③ 해당 속성이 필수값(*)인지 선택값(0)인지 표시 한다.

❖ 속성의 표기법은 엔터티 내에 이름을 포함하여 표현하면 된다.

➢ 속성의 특징

① 엔터티와 마찬가지로 반드시 **해당 업무에서 필요하고 관리하고자 하는 정보** 이어야 한다.(ex. 강사의 교재이름)
② 정규화 이론에 근간하여 정해진 **주 식별자에 함수적 종속성을** 가져야 한다.
③ **하나의 속성에는 한 개의 값만을 가진다.** 하나의 속성에 여러 개의 값이 있는 다중 값일 경우 별도의 엔터티를 이용하여 분리 한다.

➢ 속성의 분류 – 특성에 따른 분류

① 속성은 업무분석을 통해 바로 정의한 속성을 **기본속성(Basic Attribute)**
② 원래 업무상 존재하지는 않지만 설계를 하면서 도출해내는 속성을 **설계속성(Designed Attribute)**
③ 다른 속성으로부터 계산이나 변형이 되어 생성되는 속성을 **파생속성(Derived Attribute)**이라고 한다.

상품

제품번호①	제품명 ①	제조일시 ①	제조원가 ①	제조사코드 ②
0000000001	LG그램12	2020-05-29 11:34:02	425,482	LGC
0000000002	갤럭시S2.4	2020-05-29 11:34:04	221,457	SSC
0000000003	애플윙	2020-05-29 11:34:07	441,454	APC

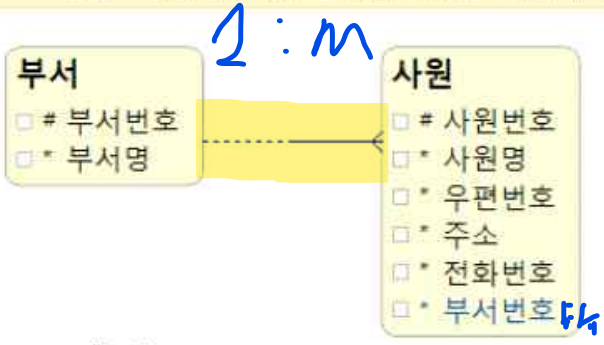
상품판매합계

제품번호 ①	판매일자 ①	판매금액 ③
0000000001	2020-06-05	55,425,482
0000000002	2020-06-05	15,221,457
0000000003	2020-06-05	87,441,454

① 기본속성
② 설계속성
③ 파생속성

➢ 속성의 분류 – 엔터티 구성방식에 따른 분류

① 엔터티를 식별할 수 있는 속성을 PK(Primary Key)속성, 다른 엔터티와의 관계에서 포함된 속성을 FK(Foreign Key)속성, 엔터티에 포함되어 있고 PK, FK에 포함되지 않은 속성을 일반속성이라 한다.



속성분류	속성명
PK속성	부서(부서번호) 사원(사원번호)
FK속성	사원(부서번호)
일반속성	부서(부서명) 사원(사원명, 우편번호, 주소, 전화번호)

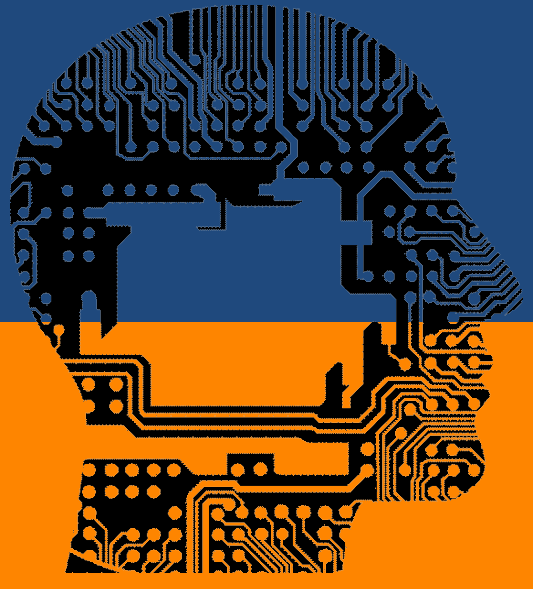
➢ 도메인(Domain)

① 각 속성은 가질 수 있는 값의 범위가 있는데 이를 그 속성의 도메인(Domain)이라 한다
② 학생이라는 엔터티가 있을 때 학점이라는 속성의 도메인은 0.0에서 4.0 사이의 실수 값이며 주소라는 속성은 길이가 20자리 이내인 문자열로 정의
③ 각 속성은 도메인 이외의 값을 갖지 못한다

➢ 속성의 명명

① 해당업무에서 사용하는 이름을 부여 한다.
② 서술 식 속성 명은 사용하지 않는다.
③ 약어 사용은 가급적 제한한다.
④ 전체 데이터모델에서 유일성 확보하는 것이 좋다.

한국IT진흥부설
정보보호교육학원 아이섹



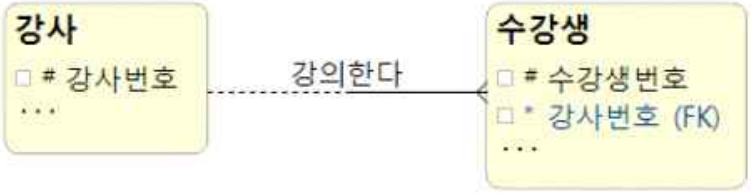
JAVA 웹 개발자 양성과정

DataBase

4. 관계

➢ 관계의 정의

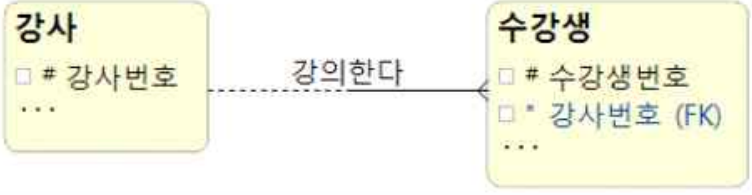
① 사전적으로 정의하면 **상호 연관성이** 있는 상태이다
② "엔터티의 인스턴스 사이의 논리적인 연관성으로서 **존재의 형태 로서나 행위로서 서로에게 연관성이 부여된 상태**" 라고 할 수 있다.

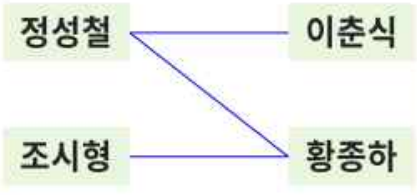


❖ 인스턴스 사이의 논리적인 연관성으로서 존재 또는 행위로서 서로에게 연관성이 부여된 상태

➢ 관계의 패어링

① 관계는 엔터티 안에 인스턴스가 개별적으로 관계를 가지는 것(패어링)이고 이것의 집합을 관계로 표현한다는 것이다.
② 개별 인스턴스가 각각 다른 종류의 관계를 가지고 있다면 두 엔터티 사이에 두 개 이상의 관계가 형성될 수 있다.
③ 각각의 엔터티의 인스턴스들은 자신이 관련된 인스턴스들과 관계의 어커런스로 참여하는 형태를 관계 패어링(Relationship Paring)이라 한다

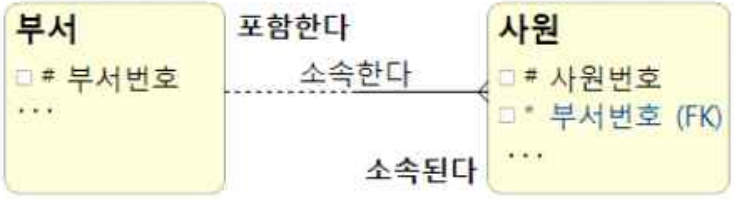




❖ 인스턴스 각각은 자신의 연관성을 가지고 있을 수 있음
❖ 이것을 집합하여 "강의한다"라는 관계를 도출

- 강사인 정성철은 이춘식과 황종하에게 강의를 하는 형태로 관계가 표현되어 있고
- 조시형은 황종하에게 강의를 하는 형태로 되어 있다.
- 이와 같이 엔터티내에 인스턴스와 인스턴스사이에 관계가 설정되어 있는 어커런스를 관계 패어링이라고 한다.

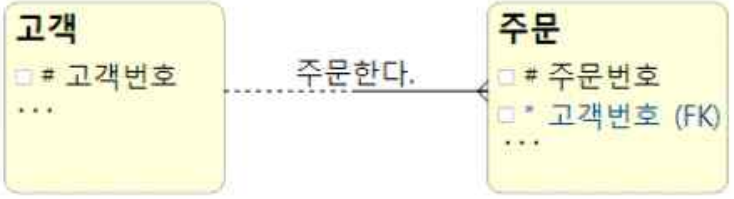
➢ 관계의 분류 – 존재의 의한 관계



❖ 사원은 부서에 항상 속해 있다.

- "소속된다'라는 의미는 행위에 따른 이벤트에 의해 발생되는 의미가 아니고 그냥 사원이 부서에 소속되어 있기 때문에 나타나는
- **즉 존재의 형태에 의해 관계가 형성**되어 있는 것이다.

➢ 관계의 분류 – 행위에 의한 관계



❖ 주문은 고객이 주문을 할 때 발생한다.

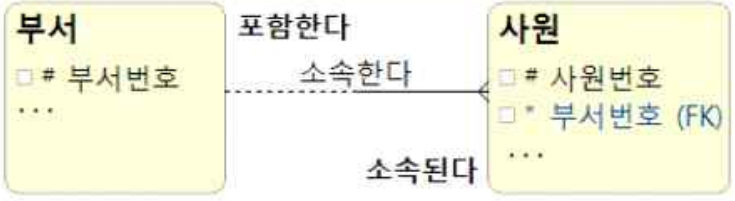
- 주문 엔터티의 주문번호는 고객이 '주문한다'라는 행위에 의해 발생되었기 때문에
- 두 엔터티 사이의 **관계는 행위에 의한 관계**가 되는 것이다

➢ 관계의 표기법

개념	설명
관계명(Membership)	관계의 이름
관계차수(Cardinality)	1:1, 1:M, M:N
관계선택사양(Optionality)	필수관계, 선택관계

➢ 관계의 표기법 - 관계명

① 엔터티가 관계에 참여하는 형태를 지칭한다.
② 각각의 관계는 두 개의 관계명을 가지고 있다.
③ 또한 각각의 관계명에 의해 두 가지의 관점으로 표현될 수 있다.



- 엔터티에서 관계가 시작되는 편을 관계시작점(The Beginning)이라고 부르고 받는 편을 관계끝점(The End)이라고 부른다.
- 관계 시작점과 끝점 모두 관계이름을 가져야 하며 참여자의 관점에 따라 관계이름이 능동적(Active)이거나 수동적(Passive)으로 명명된다.

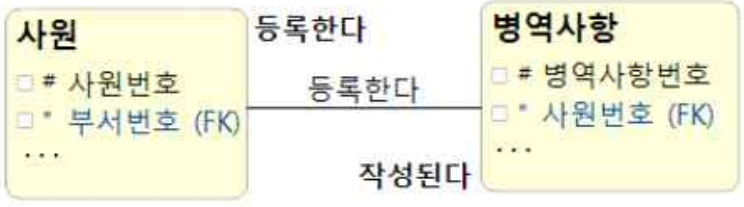
▪ 관계의 명명 규칙

- **애매한 동사를 피한다.** 예를 들면 '관계된다', '관련이 있다', '이다', '한다' 등은 구체적이지 않아 어떤 행위가 있는지 또는 두 참여자간 어떤 상태가 존재하는지 파악할 수 없다.
- **현재형으로 표현한다.** 예를 들면 '수강을 신청했다', '강의를 할 것이다'라는 식으로 표현해서는 안된다. '수강 신청한다', '강의를 한다'로 표현해야 한다.

➢ 관계의 표기법 - 관계차수

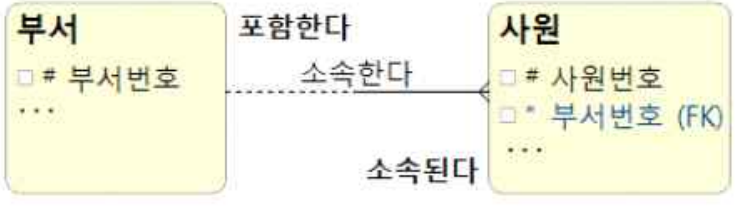
① 두 개의 엔터티 간 관계에서 참여자의 수를 표현하는 것을 **관계 차수(Cardinality)**라고 한다.
② 가장 일반적인 관계 차수 표현방법은 **1:M, 1:1, M:M**이다

▪ 관계 차수 (1:1)



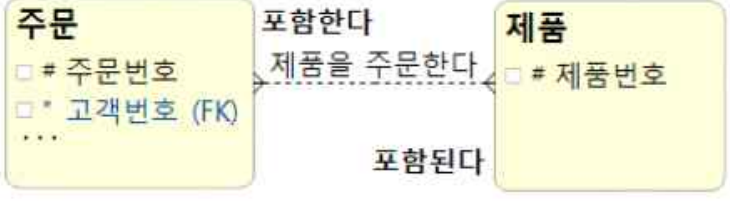
- 관계에 참여하는 각각의 엔터티는 관계를 맺는 다른 엔터티에 대해 단지 하나의 관계만을 가지고 있다.

▪ 관계 차수 (1:M)



- 한 명의 사원은 한 부서에 소속되고 한 부서에는 여러 사원을 포함한다.

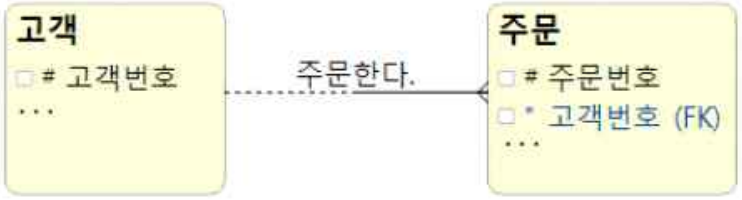
▪ 관계 차수 (M:M)



- 관계에 참여하는 각각의 엔터티는 관계를 맺는 다른 엔터티에 대해 하나나 그 이상의 수와 관계를 가지고 있다

➢ 관계의 표기법 - 관계선택사양(Optionality)

① "반드시 지하철의 문이 닫혀야만 지하철은 출발한다."
② 지하철출발과 지하철문닫힘은 **필수(Mandatory)적**으로 연결 관계가 있는 것이다. 이와 같은 것이 데이터 모델의 관계에서는 **필수참여관계(Mandatory)**가 된다.
③ 지하철의 출발을 알리는 안내방송은 지하철의 출발과 상관없이 방송해도 아무런 문제가 발생하지 않는다. 즉 안내방송시스템이 고장이 나도 지하철운행에는 별로 영향을 주지 않는다
④ 지하철의 출발과 지하철방송과는 정보로서 관련은 있지만 서로가 필수적인(Mandatory) 관계는 아닌 **선택적인 관계(Optional)**가 되는 것이다. 이와 같은 것이 데이터 모델 관계에서는 **선택참여관계(Optional)**가 된다



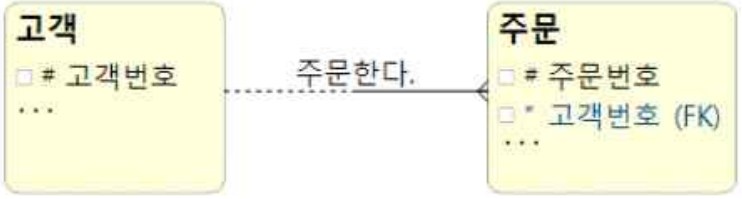
- 고객은 여러 개의 주문을 할 수 도 있고 한 개의 주문도 하지 않을 수 있다. (선택적 관계)
- 주문은 반드시 고객을 가진다. (필수적 관계)

➢ 관계 정의 시 체크 사항

① 두 개의 엔터티 사이에 관심있는 **연관 규칙**이 존재하는가?
② 두 개의 엔터티 사이에 **정보의 조합**이 발생되는가?
③ 업무기술서, 장표에 관계연결에 대한 **규칙이 서술**되어 있는가?
④ 업무기술서, 장표에 관계연결을 가능하게 하는 **동사(Verb)**가 있는가?

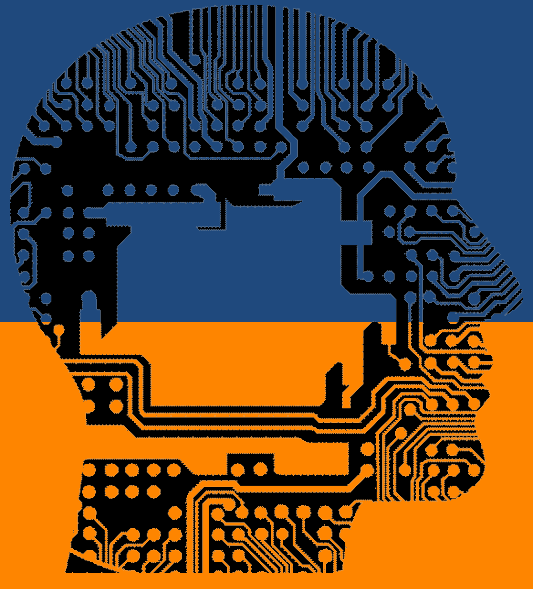
➢ 관계 읽기

① 기준(Source) 엔터티를 한 개(One) 또는 각(Each)으로 읽는다.
② 대상(Target) 엔터티의 관계 참여도 즉 개수(하나, 하나 이상)를 읽는다.
③ 관계선택사양과 관계 명을 읽는다.



각각의/하나의	기준 엔터티	관계 차수	대상엔터티	필수/선택	관계명
각각의	고객은	여러 개의	주문을	때때로	주문한다.
각각의	주문은	하나의	고객을	반드시	가진다.

한국IT진흥부설
정보보호교육학원 아이섹



JAVA 웹 개발자 양성과정

DataBase

5. 식별자

➢식별자(Identifiers) 개념

① 엔터티는 인스턴스들의 집합이라고 하였다. 여러 개의 집합체를 담고 있는 하나의 통에서 각각을 구분할 수 있는 논리적인 이름이 있어야 한다. 이 구분자를 식별자(Identifier)라고 한다.
② Entity의 각 Instance를 개별적으로 식별하기 위해 사용되는 Relationship 또는 Attribute들의 조합

고객(고객번호)	사원(사원번호)	주문(주문번호)	상품(상품번호)

❖ 식별자는 엔터티내에서 인스턴스들을 구분할 수 있는 구분자 이다.

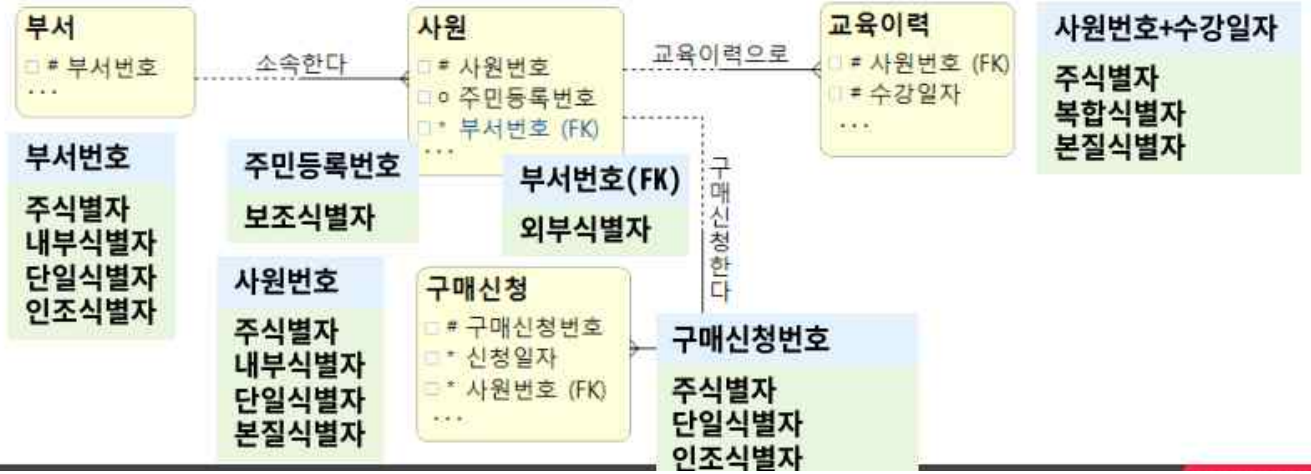
➢식별자의 특징

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

특징	내용	비고
유일성	- 주식별자에 의해 엔터티내에 모든 인스턴스들을 유일하게 구분함	예) 사원번호가 주식별자가 모든 직원들에 대해 개인별로 고유하게 부여됨
최소성	- 주식별자를 구성하는 속성의 수는 유일성을 만족하는 최소의 수가 되어야 함	예) 사원번호만으로도 고유한 구조인데 사원분류코드+사원번호로 식별자가 구성될 경우 부절한 주식별자 구조임
불변성	- 주식별자가 한 번 특정 엔터티에 지정되면 그 식별자의 값은 변하지 않아야 함	예) 사원번호의 값이 변한다는 의미는 이전기록이 말소되고 새로운 기록이 발생되는 개념임
존재성	- 주식별자가 지정되면 반드시 데이터 값이 존재 (Null 안됨)	예) 사원번호 없는 회사직원은 있을 수 없음

➢식별자 분류

분류	식별자	설명
대표성 여부	주식별자	- 엔터티 내에서 각 행을 구분할 수 있는 구분자이며, 타 엔터티와 참조관계를 연결할 수 있는 식별자 (ex. 사원번호, 고객번호)
	보조식별자	- 엔터티 내에서 각 행을 구분할 수 있는 구분자이나 대표성을 가지지 못해 참조관계 연결을 못함(ex. 주민등록번호)
스스로 생성여부	내부식별자	- 엔터티 내부에서 스스로 만들어지는 식별자(ex. 고객번호)
	외부식별자	- 타 엔터티와의 관계를 통해 타 엔터티로부터 받아오는 식별자(ex. 주문엔터티의 고객번호)
속성의 수	단일식별자	- 하나의 속성으로 구성된 식별자(ex. 고객엔터티의 고객번호)
	복합식별자	- 둘 이상의 속성으로 구성된 식별자(ex. 주문상세엔터티의 주문번호+상세순번)
대체여부	본질식별자	- 업무에 의해 만들어지는 식별자(ex. 고객번호)
	인조식별자	- 업무적으로 만들어지지는 않지만 원조식별자가 복잡한 구성을 가지고 있기 때문에 인위적으로 만든 식별자(ex. 주문엔터티의 주문번호(고객번호+주문번호+순번))



➢식별자 도출 기준

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

직원
- # 직원번호
- * 직원명
- * 주민등록번호
- o 우편번호
- o 주소
- * 전화번호
- o 이메일주소

- 주민등록번호도 식별자 후보가 될 수 있지만 **해당 업무에서 자주 사용하는 사원번호를 주식별자로 지정**함
- 직원명과 같은 이름은 **주식별자로 지정하지 않음**(동명이인이 없다고 해도 지정하면 안됨)
- 직원번호+주민등록번호로 복합 식별자로 하면 유일성을 보장하지만 **직원번호만으로도 충분하기 때문에 직원번호**로만 지정함

너무 많은 주식별자 속성

비효율_접수
- # 접수일자
- # 관할부서
- # 입력자사번
- # 접수방법코드
- # 신청인구분코드
- # 신청인주문번호
- # 신청횟수
- * 신청자명

...

```
SELECT 계약금
  FROM 접수
 WHERE 접수.접수일자 = '2010.07.15'
   AND 접수.관할부서 = '1001'
   AND 접수.입력자사번 = 'AB45588'
   AND 접수.접수방법코드 = 'E'
   AND 접수.신청인구분코드 = '01'
   AND 접수.신청인주민번호 = '7007171234567'
   AND 접수.신청횟수 = '1'
;
```

인조식별자를 통해 단순화한 주식별자 속성

효율_접수
- # 접수번호
- * 신청자명
- * 장소
- * 계약금
- * 접수일자
- * 관할부서
- * 입력자사번
- * 접수방법코드

...

```
SELECT 계약금
  FROM 접수
 WHERE 접수.접수번호 = '100120100715001'
 ;
```

접수번호 = 관할부서+접수일자+일련번호

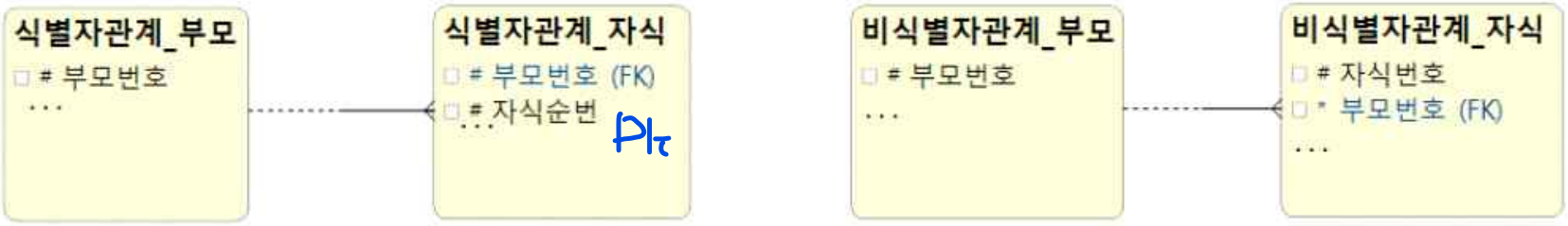
- 모델상에 표현하는 문장의 간편성뿐만 아니라 애플리케이션 구성에 있어서도 복잡한 소스구성을 피하기 위하여 과도한 복합키는 배제하도록 노력해야 한다.

➢식별자관계와 비식별자 관계의 결정

자주 나옴

① 외부식별자(Foreign Identifier)는 자기 자신의 엔터티에서 필요한 속성이 아니라 다른 엔터티와의 관계를 통해 자식 쪽에 엔터티에 생성되는 속성을 **외부식별자**라 하며 데이터베이스 생성 시에 **Foreign Key**역할을 한다.

② 자식엔터티에서 부모엔터티로부터 받은 외부식별자를 자신의 **주식별자로 이용할 것인지(식별자 관계)** 또는 **부모와 연결이 되는 속성으로서만 이용할 것인지를 결정(비식별자 관계)**해야 한다.



- 엔터티 사이 관계 유형은 **업무 특징, 자식엔터티의 주 식별자 구성, SQL전략**에 의해 결정된다.

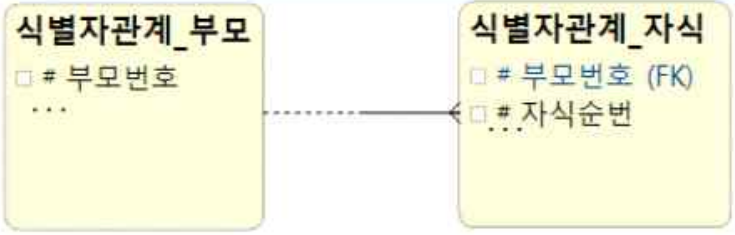
1 : M

부모 자식

예) 게시글 댓글

➢식별자 관계

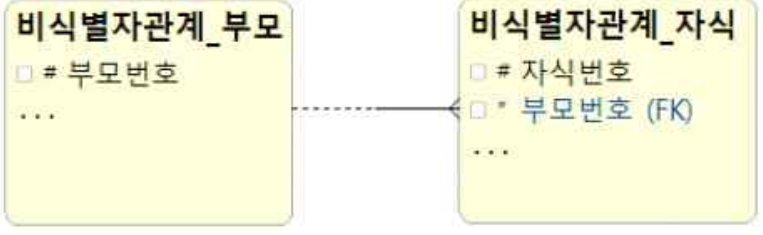
① 자식엔터티의 주식별자로 부모의 주식별자가 상속이 되는 경우를 식별자 관계(Identifying Relationship)라고 지칭한다.
② 부모로부터 받은 식별자를 자식엔터티의 주식별자로 이용하는 경우는 Null값이 오면 안되므로 반드시 부모엔터티가 생성되어야 자기 자신의 엔터티가 생성되는 경우이다.



- 외부 식별자의 주 식별자 역할
- 즉 부모의 식별자가 자식에게도 식별자 역할을 한다.

➢비식별자관계

① 부모엔터티로부터 속성을 받았지만 자식엔터티의 주식별자로 사용하지 않고 일반적인 속성으로만 사용하는 경우가 있다. 이와 같은 경우를 비식별자 관계(Non-Identifying Relationship)라고 하며 다음의 네 가지 경우에 비식별자 관계에 의한 외부속성을 생성한다.
② 자식엔터티에서 받은 속성이 반드시 필수가 아니어도 무방하기 때문에 부모 없는 자식이 생성될 수 있는 경우이다.
③ 자식엔터티에 주식별자로 사용하여도 되지만 자식엔터티에서 별도의 주식별자를 생성하는 것이 더 유리하다고 판단될 때 비식별자 관계에 의한 외부식별자로 표현한다.



- 외부 식별자는 FK로써의 역할을 함
- 즉 부모의 식별자는 자식에게 식별자 역할을 하지않고 FK로써의 역할을 한다.

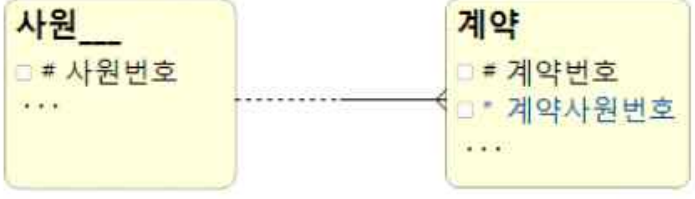
➢비식별자관계 - 계속

▪ 식별자 관계

▪ 비식별자 관계

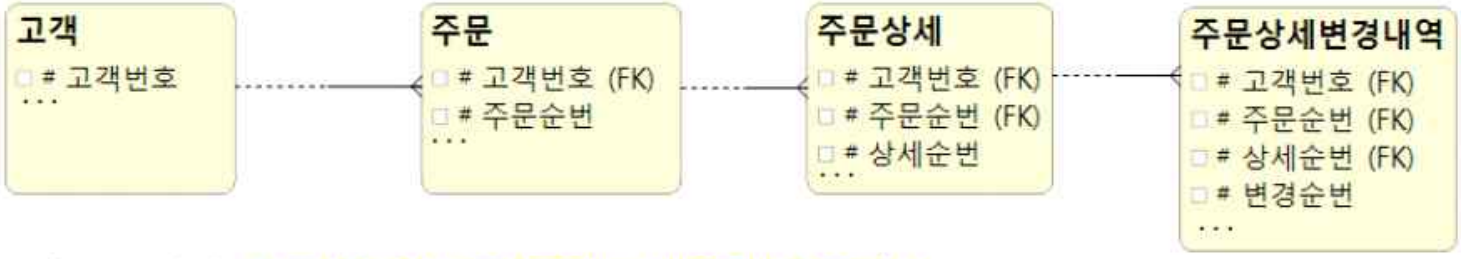
- 방문접수, 인터넷접수, 전화접수 엔터티가 접수 엔터티로 통합되면서 비 식별자 관계가 됨

▪ 자식 엔터티의 독립 주 식별자(계약번호)

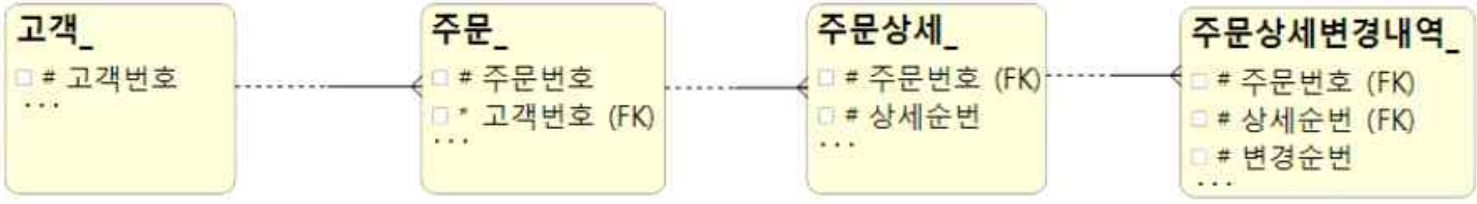


➢식별자 관계로만 설정할 경우의 문제점 자주나옴

- 식별자 관계 – 주문 엔터티가 고객 엔터티의 고객번호를 식별자로 받음



- 비식별자 관계 – 주문엔터티가 주문번호라는 식별자를 별도로 이용

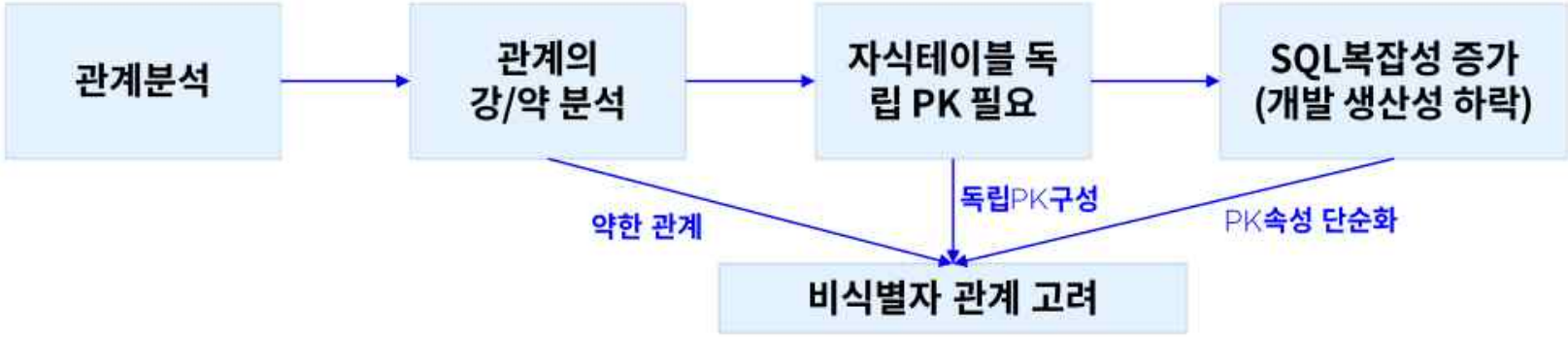


- 지속적으로 식별자 관계를 연결한 데이터 모델의 PK속성의 수는 데이터 모델의 흐름이 길어질수록 증가할 수 밖에 없는 구조를 가지게 된다.
- 개발자가 개발할 때 당연히 데이터 모델을 참조하면서 엔터티와 관계를 이용하여 개발해야 하는데 생성된 엔터티 스키마 정보만을 보고 개발하는 경우가 많다
- 조인에 참여하는 주식별자속성의 수가 많을 경우 정확하게 조인관계를 설정하지 않고 즉, 누락하여 개발하는 경우가 간혹 발견되기도 한다
- 정리하면 식별자 관계만으로 연결된 데이터 모델의 특징은 주식별자 속성이 지속적으로 증가할 수 밖에 없는 구조로서 개발의 복잡성과 오류가능성을 유발시킬 수 있는 요인이 될 수 있다는 사실을 기억해야 한다.

➢식별자관계와 비식별자관계

① 실제로 프로젝트를 전개할 때 식별자관계와 비식별자관계를 취사선택하여 연결하는 내공은 높은 수준의 데이터모델링의 기술이라 할 수 있다.

② 식별자관계에서 비식별자관계를 파악하는 기술이 필요한데 다음 흐름(Flow)에 따라 비식별자관계를 선정한다면 합리적으로 관계를 설정하는 모습이 될 수 있다. 기본적으로 식별자관계로 모든 관계가 연결되면서 다음 조건에 해당할 경우 비식별자관계로 조정하면 된다.



항목	식별자 관계	비 식별자 관계
목적	강한 연결 관계 표현	약한 연결 관계 표현
자식 주 식별자 영향	자식 주 식별자의 구성에 포함됨	자식 일반 속성에 포함됨
연결 고려사항	반드시 부모 엔터티 종속 자식 주식별자구성에 부모 주식별자포함 필요 상속받은 주식별자속성을 타 엔터티에 이전 필요	약한 종속관계 자식 주식별자구성을 독립적으로 구성 상속받은 주식별자속성을 타 엔터티에 차단 필요 부모 쪽의 관계 참여가 선택관계일수 있음

감사합니다
THANK YOU