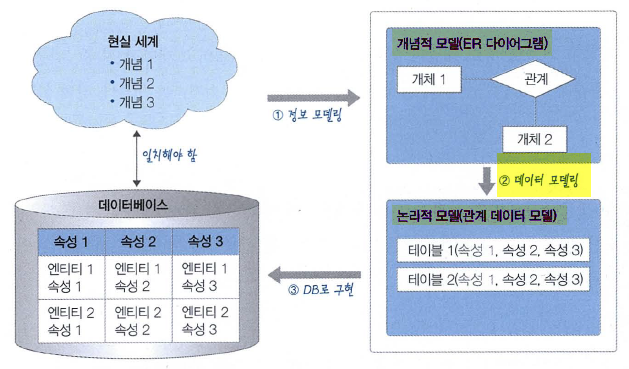
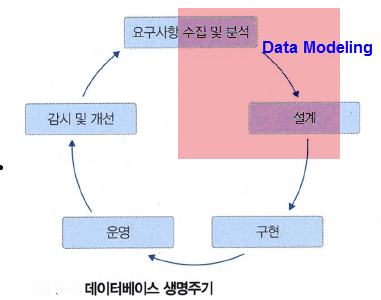
**Ch06 Data Modeling**

**I. Data Modeling**

**1. Data Modeling 개념**





**2. Data Modeling 과정**

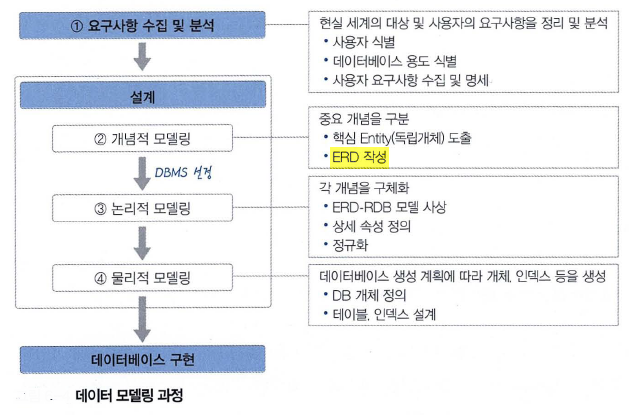
**(1) 요구사항 수집 및 분석**

①실제 문서를 수집하고 분석한다.

② 담당자와의 인터뷰나 설문조사를 통해 요구사항을 직접 수렴한다.

③ 비슷한 업무를 처리하는 기존의 데이터베이스를 분석한다.

④ 각 업무의 연관된 모든 부분을 살펴본다.



**(2) Conceptual Modeling (개념적 모델링)**

업무의 핵심적인 개념을 구분하고 전체적인 뼈대를 만드는 과정, 즉 개체(entity)를 추출하고 각 개체들간의

관계(relationship)를 정의하여 ER-Diagram을 만드는 과정이다.

설계자는 사용자의 요구사항을 분석하여 가장 핵심적인 개체와 개별 개체를 식별할수 있는 Primary Key 와

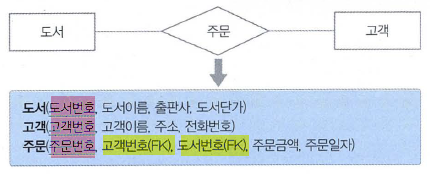
각 개체간의 관계를 정의하여 데이터베이스화 할수 있는 일반적인 개념으로 표현한다.



**(3) Logical Modeling (논리적)**

ER-Diargram을 이용하여 사용하고자 하는 DBMS에 맞게 Mapping하여 실제 DB로 구현하기 위한 모델을

만드는 과정이다. 즉, Relation Data Model 로 Mapping이 이루어진다.



① 개념적 모델링에서 추출하지 않았던 상세 속성들을 모두 추출한다.

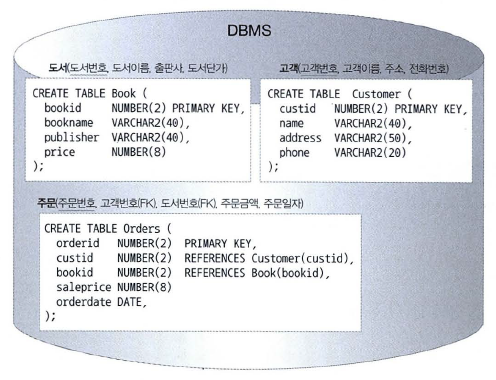
② 정규화를 수행한다.

여러 개체들이 데이터를 중복 저장하는 문제를 해결

③ 데이터의 표준화를 수행한다.

Data Dictionary(용어사전) 과 Domain(범위)를 정의하여 용어 정리 및 데이터 형태, 크기를 확정한다.

**(4) Physical Modeling (물리적)**



작성된 논리적 모델을 실제 저장장치에 저장하기 위한 물리적 구조를 정의하고 구현하는 과정이다.

① 자료의 입력에서부터 처리가 완료될때까지(Transaction)의 시간을 최소화 해야한다.

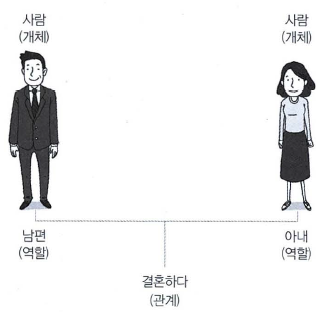
② 발생되는 Transaction을 분석하여 효율적으로 분산될수 있도록 해야한다.

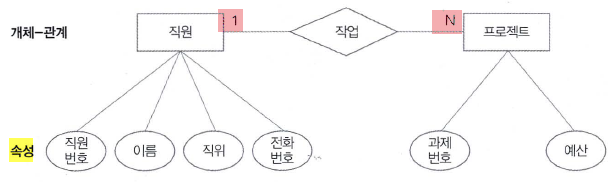
③ 데이터가 저장될 공간을 효율적으로 배치하고 검색(Index)을 효율적으로 할수 있도록 해야 한다.

**II. ER Model**

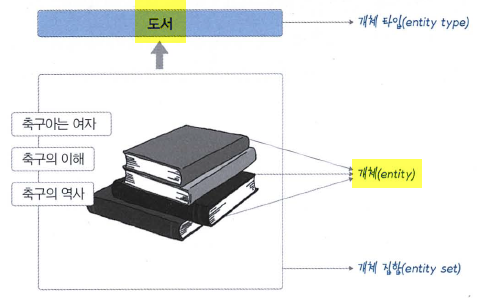
개념적 모델링에 사용

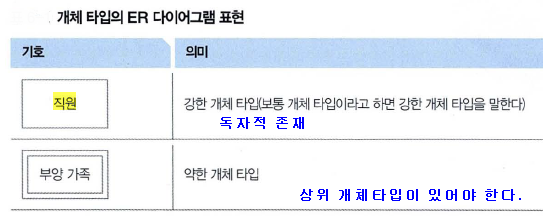
사물을 개체와 개체간의 관계(Relation)로 표현되며, 속성(attribute)에 의해서 식별된다.





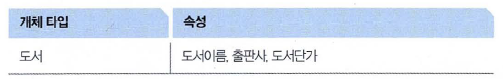
**1. Entity**

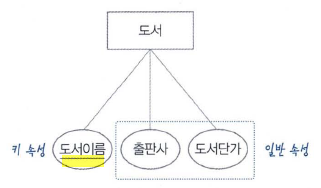
****



**2. Attribute (속성)**

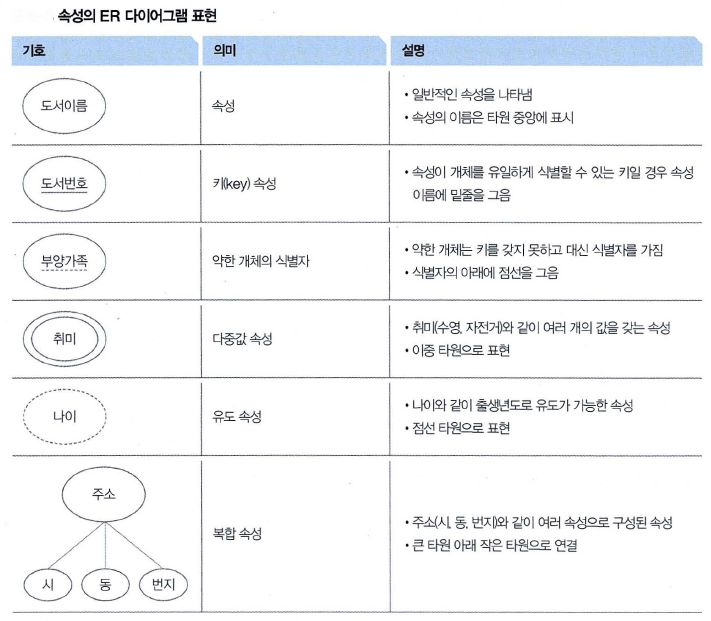
개체가 가진 성질.





[속성 유형]

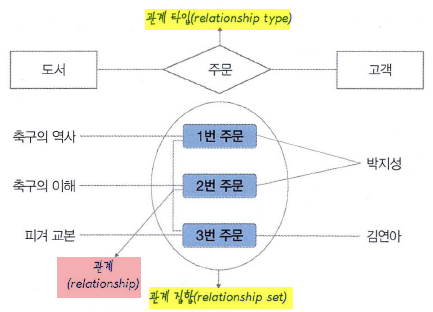
|  |  |
| --- | --- |
| 단순속성 & 복합속성 |  |
| 단일값 속성 & 다중값 속성 |  |
| 저장속성 & 유도 속성 |  |

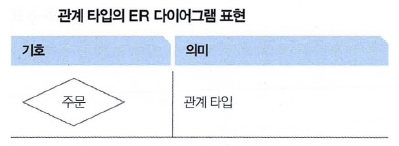


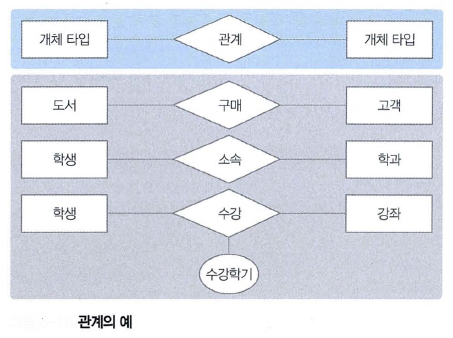
**3. Relation**

**(1) Relation 개념**

개체 사이의 연관성 표현





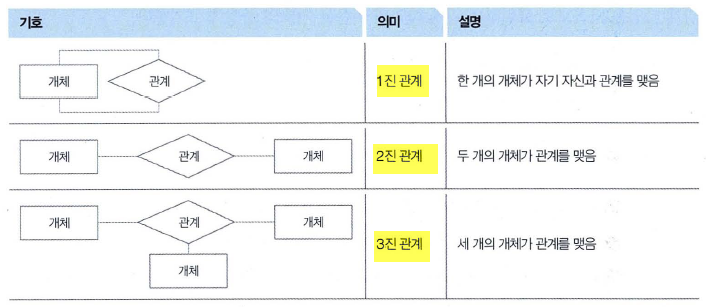


**(2) Relation Type의 유형**

각 개체타입이 관계를 맺은 형태, 즉 degree(차수) 와 cardinality(관계 대응수)에 따라 분류

1) degree(차수) 따른 유형

: 관계 집합에 참가하는 Entity Type의 수 (n진 관계)



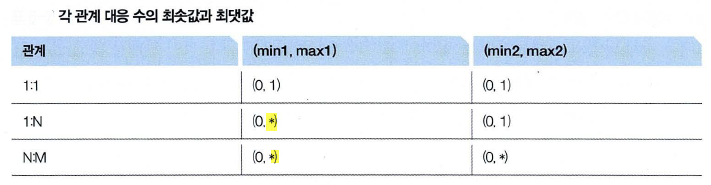
|  |  |
| --- | --- |
| 1진관계 (recursive relationship)  : 순환관계 |  |
| 2진관계 (binary relationship) |  |
| 3진관계 (ternay relationship) |  |

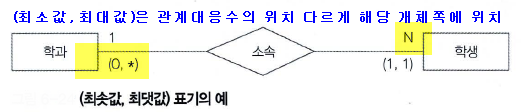
**2) Cardinality (관계 대응수)**

: 두 개체타입의 관계에 실제로 참여하는 개별 개체수





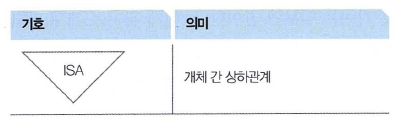


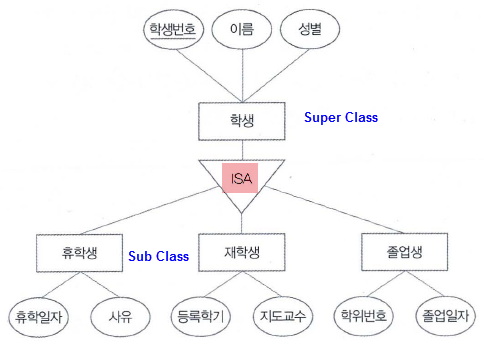


|  |  |
| --- | --- |
| 1 : 1 |  |
| 1 : N |  |
| N : M |  |

**(3) IS A 관계 (상하 관계)**

상위 개체 타입(Super Class)의 특성에 따라 하위 개체 타입(Sub Class)이 결정되는 형태 (is a)



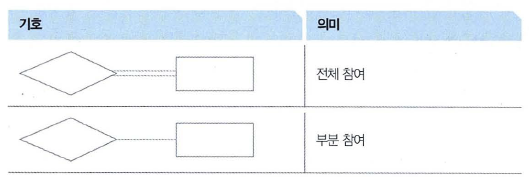


학생을 ISA 관계로 표현하면, 휴학생, 재학생, 졸업생을 모두 하나의 개체로 표현하거나 완전히 다른

세 개의 개체로 표현하는 것보다 모델링이 정교해진다.

(4) 참여 제약 조건

개체 집합내 모든 개체가 관계에 참여하는지 유무에 따라 전체참여 와 부분 참여로 구분됨





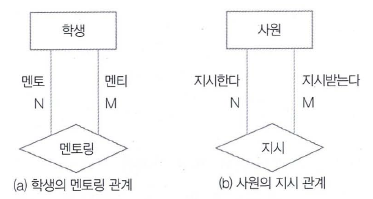
(5) 역할

각 개체들은 고유한 역할을 한다.



(6) 순환적 관계 (recursive relationship)

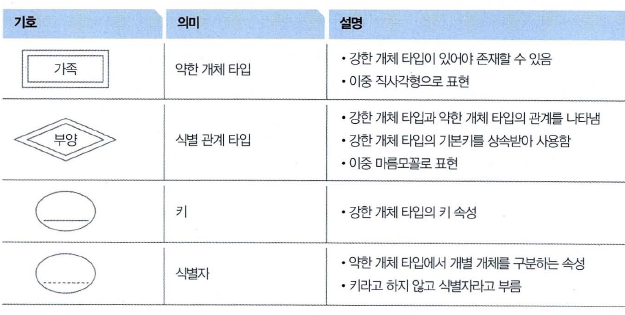
하나의 개체타입이 동일한 개체타입(자기 자신)과 순환적인 관계를 가지는 형태

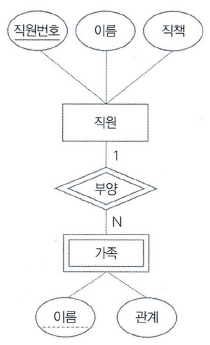


**4. 약한 개체 타입 과 식별자**

강한 개체 타입은 기본키를 가지지만, 약한 개체 타입은 자기 자신의 기본키만으로 식별이 어려우므로 상위의 강한 개체 타입의 기본키를 상속받아 사용한다.

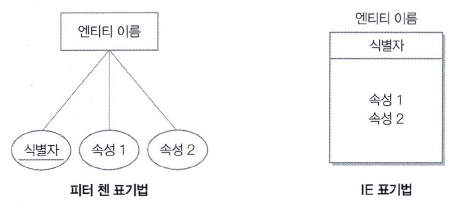
가족 개체타입의 '이름'과 같이 독립적인 키로는 존재할수 없지만, 상위 개체 타입의 키와 결합하여 약한 개체 타입의 개별 개체를 고유하게 식별하는 속성을 식별자 혹은 부분키(partial key)라고 한다.

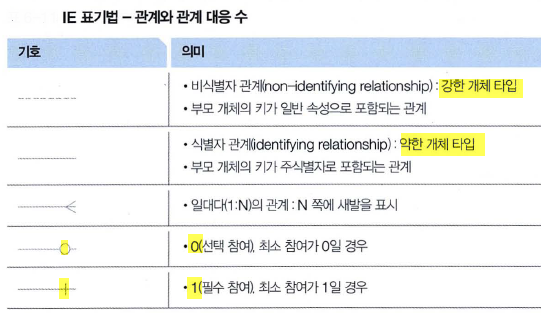


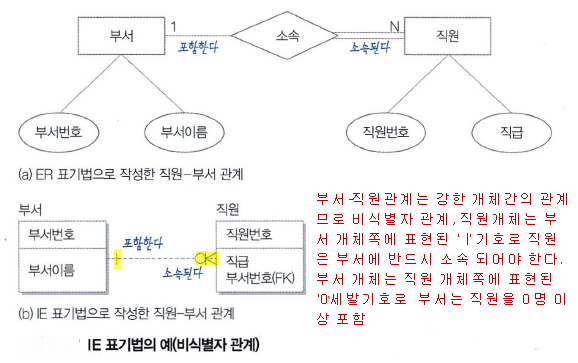


|  |
| --- |
| [식별관계 & 비식별관계]   * 관계 : 두 엔티티 간의 업무적인 연관성 * 식별 관계 : 부모 테이블의 기본키가 자식 테이블의 기본키 혹은 후보키 그룹의 구성원으로 전이되는 것 * 비식별 관계 : 부모 테이블의 기본키가 자식 테이블의 일반 컬럼으로 전이되는것 |

**5. IE 표기법 (Information Engineering Notation)**



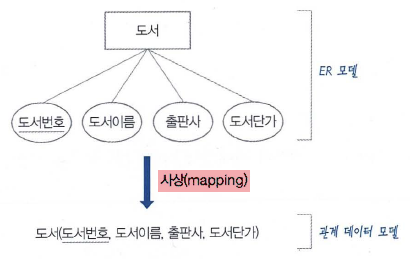


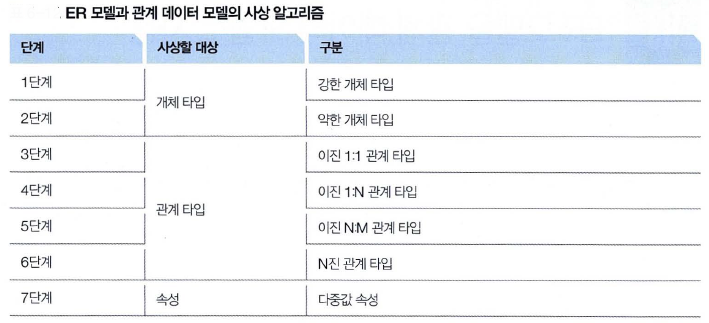




**III. ER 모델을 Relationship 모델로 Mapping**

|  |  |  |
| --- | --- | --- |
| **ER Model** | =>  Mapping | **Relation Model** |
| 개념적 모델 | 논리적 모델 |
| Entity Type, Relation Type, Attribute 표현 | Relation으로 DB 논리적 구조 표현 |
|  |  |





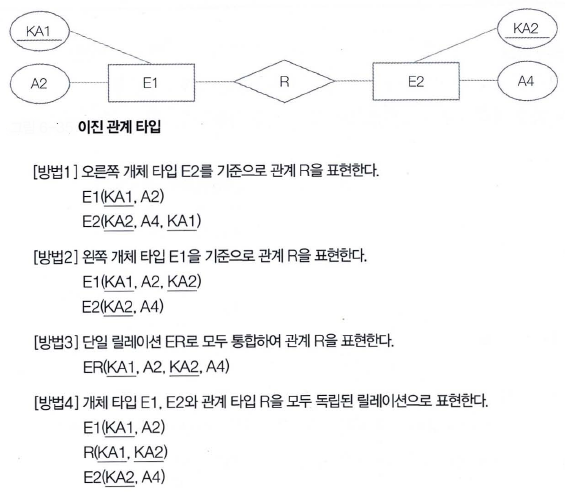
**1. Entity Type Mapping**

|  |  |
| --- | --- |
| 1단계:  개체타입E => Relation R  개체타입 속성=>R 속성  기본/외래키 => PK/FK |  |
| 2단계:  생성된 R은 자신의 키와 외래키를 사상하여 자신의 기본키를 구성. |

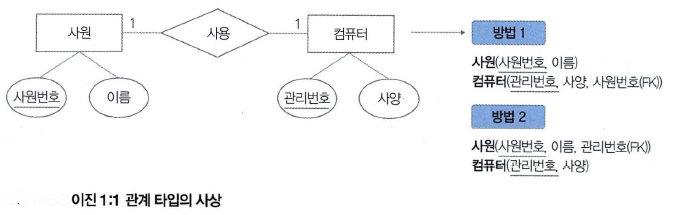
**2. Relation Type Mapping**

관계 타입은 각 관계타입이 맺고 있는 degree, cardinality에 따라 사상방식이 다르다.

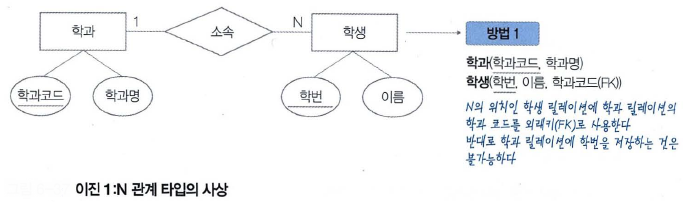
**※ Binary Relationship (이진관계 타입)**



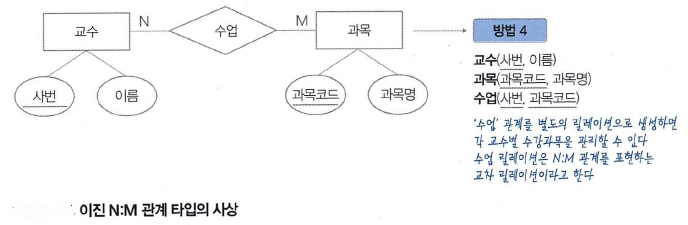
**(1) [3단계] 이진 1 : 1 관계 타입**



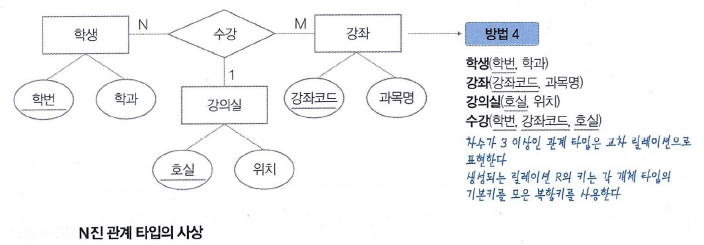
**(2) [4단계] 이진 1 : N 관계 타입**



**(3) [5단계] 이진 N : M 관계 타입**



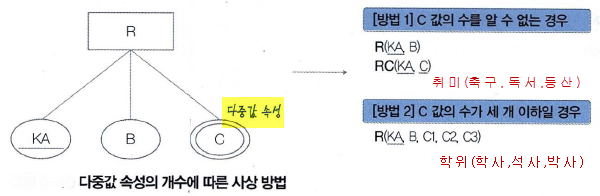
**(4) [6단계] N진 관계 타입**



**3. 다중값 속성의 사상**

다중값(하나의 속성에 여러 값을 갖는 속성)은 직접 사상할수 없으며 2가지 방법이 있다.

(예) 다중값 속성 : 취미가 (등산, 스키, 노래)



**(1) [7단계] 다중값 속성**

속성의 갯수를 알수 없는 경우 : 방법1

속성의 갯수가 제한적으로 정해지는 경우 : 방법2

