

데이터베이스 모델링



데이터베이스

- 데이터 베이스란?

- ➡ 저장할 수 있는 공간

- 데이터베이스 관리 시스템(DBMS)이란?

- ➡ 저장할 수 있는 공간을 효율적으로 정의하고 필요에 맞게 저장하고 검색할 수 있는 관리 시스템 의미



데이터베이스 개념

- 논리적으로 연관된 하나 이상의 자료 모음
- 여러 개의 자료 파일을 조직적으로 통합하고 중복 배제 및 구조화한 자료의 집합체



데이터베이스 개념

- 통합 데이터
 - ➡ 데이터 통합으로 인해 중복 최소화의 효과
- 저장된 데이터
 - ➡ 컴퓨터가 접근, 처리 가능한 데이터
- 운영 데이터
 - ➡ 특정한 조직의 운영에 중요한 데이터
- 공용 데이터
 - ➡ 특정 목적으로 모인 조직내의 모든 사용자가 공통적으로 사용 가능한 데이터 집합



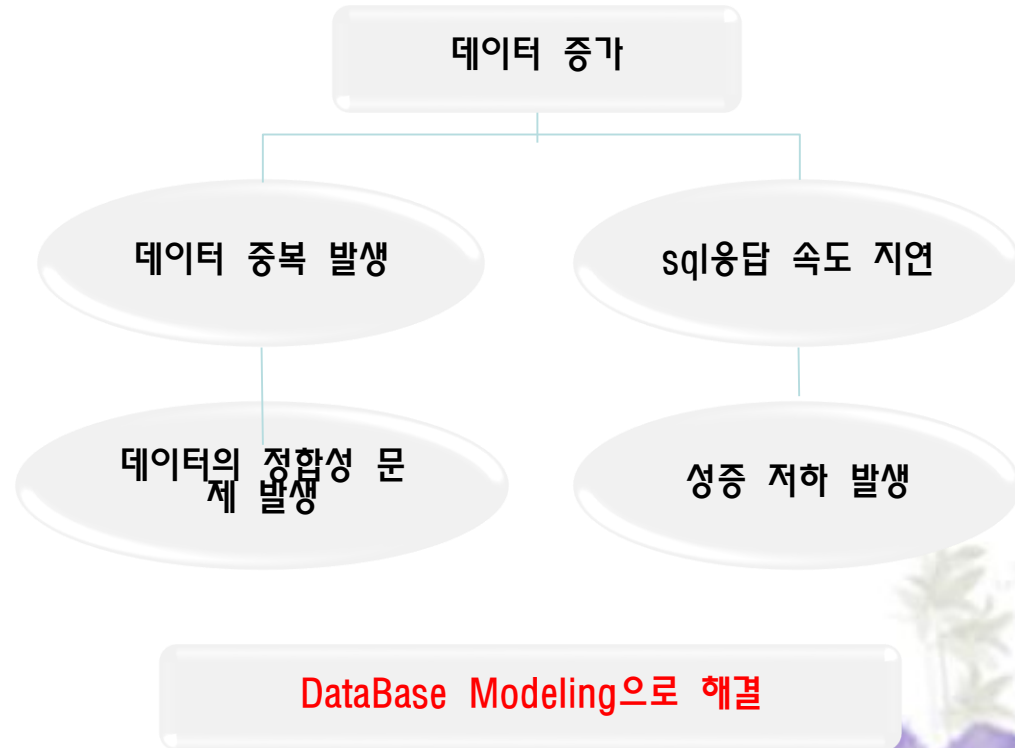
데이터 증가에 따른 문제

1. 데이터 중복 발생

- 데이터 정합성 문제
- Application에서 해결

2. SQL응답 속도 지연

- 성능 저하 발생
- sql 튜닝으로 처리 가능



- 근본적인 문제 해결을 하려면 Data Modeling 으로 해결 해야

데이터베이스시스템(DBMS)

- 사용자의 모든 요구를 수행할 수 있도록 설계되어 데이터베이스 생성, 정의, 유지 및 사용자가 원하는 모든 정보 조작할 수 있는 기능과 최적의 상태로 데이터를 유지할 수 있는 여러 가지 제어 기능을 보유한 소프트웨어



데이터베이스 시스템의 장점

데이터 중복 최소화

데이터 공통 사용

데이터 일관성 유지

데이터 무결성 유지

데이터 보안 보장

데이터 표준화 가능

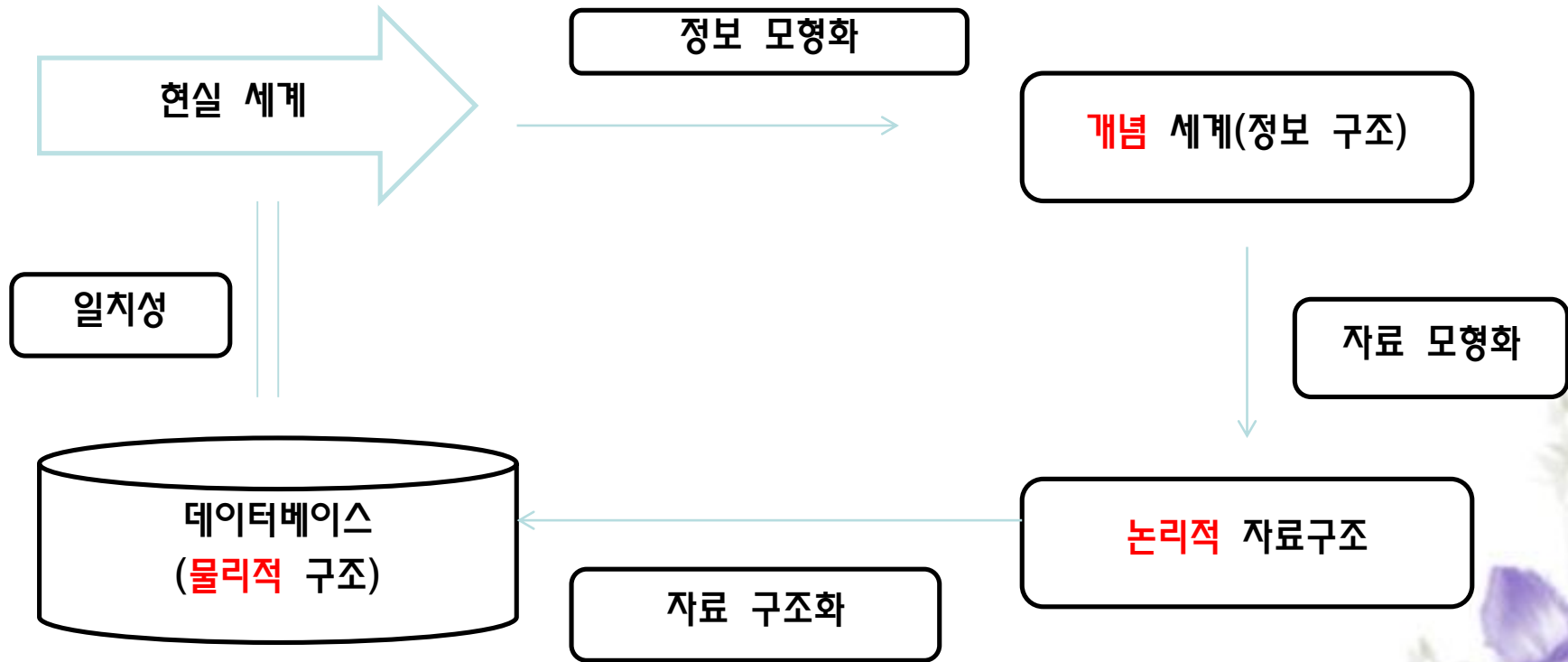


데이터베이스 설계

- 현실 세계를 정보 모형화(Information Modeling)을 기반으로 컴퓨터가 알고 있는 정보 구조로 개념 세계를 구성하고 다시 모형화를 통해 논리적인 자료 구조화 한 후 자료구조화(Data Structuring)을 통해 물리적인 구조인 데이터베이스를 만듦



데이터베이스화 과정



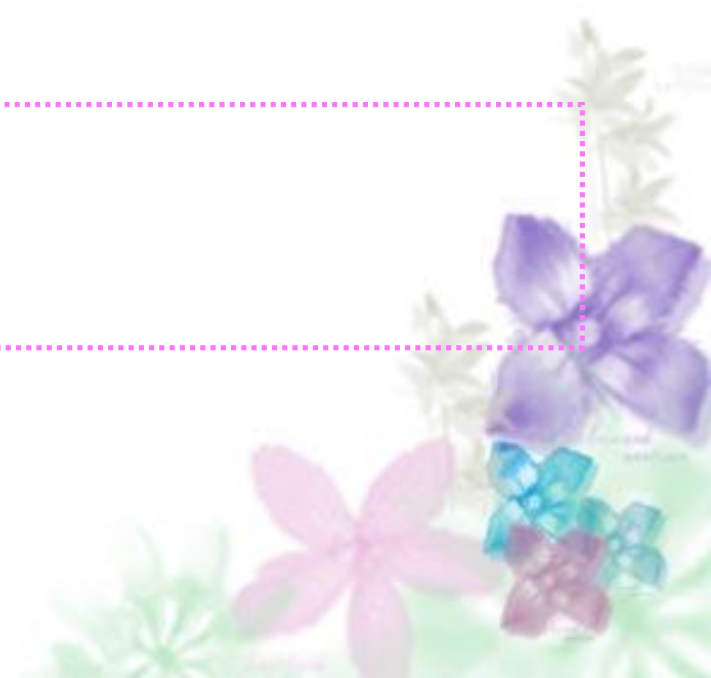
데이터베이스 모델링

* 데이터베이스 모델링 소개

: 데이터 모델링이란? 현실 세계의 업무적인 프로세서를 물리적으로 데이터베이스화 하기 위한 과정으로 **실체와 관계를 중심으로** 체계적으로 표현하고 문서화하는 기법이며 정보 시스템의 중심을 데이터의 관점에서 접근하는 데이터 중심의 분석 방법이다.

정의

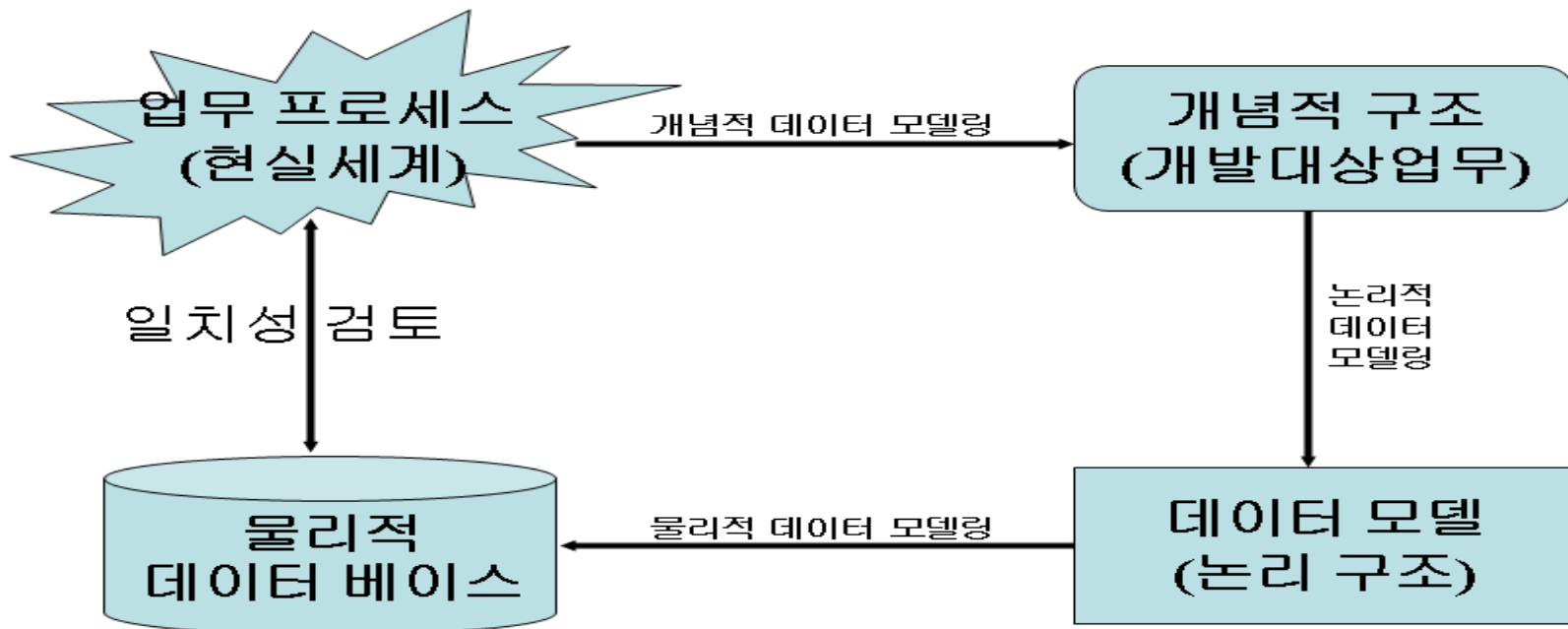
현실세계의 업무적인 프로세스를 물리적으로 데이터베이스화 시키기 위한 과정.



데이터베이스 모델링

과정

- ① 개념적 데이터베이스 모델링
- ② 논리적 데이터베이스 모델링
- ③ 물리적 데이터베이스 모델링



데이터베이스 모델링

모델링 단계		내 용
개념적 데이터 모델링		<p>목적 : 업무를 일반화 시킨다.</p> <p>결과물 : 실체-관계 모델(Entity-Relationship Model)을 통한 E-R Diagram 생성</p>
논리적 데이터 모델링	관계형 데이터베이스 이론 적용	<p>1. 메핑룰 (Mapping Rule) : 개념적 데이터 모델링 단계에서 얻어진 개체-관계 모델을 관계형 데이터베이스 이론에 맞게 변환시키는 작업. 결과물 : 관계 스키마</p> <p>2. 완벽한 정규화 수행 : 정규화를 수행하는 목적은 보다 효율적으로 데이터를 저장할 수 있는 구조를 만들기 위함이다.</p>
물리적 데이터 모델링		<p>1. 개발DBMS를 선정하고 컬럼의 데이터 타입과 사이즈를 정의.</p> <p>2. 데이터 사용량을 분석과 사용자들이 데이터소스에 액세스 할 때의 구체적인 프로세스를 분석.</p> <p>3. 역정규화 수행 : 역정규화는 데이터베이스가 보다 효율적으로 동작하도록 하기 위해 정규화에 위배되는 행위를 일부러 하는 것을 말한다.</p> <p>4. 인덱스, 뷰, 스토어드 프로시저, 함수, 트리거, 제약조건 등 데이터베이스 내의 개체들 정의</p> <p>5. 물리적인 데이터베이스 생성</p>

데이터베이스 모델링

프로젝트 진행과정

계획 - 주제선정

분석 - 업무에 대한 정확한 이해 필요.

설계 - 데이터베이스 모델링 (개념적 - 논리적 - 물리적)

구현 - application 개발단계

테스트 - 디버깅

유지보수



일반적으로 프로젝트 기간이 1년이라고 했을 때 분석, 설계 단계가 6개월 정도 소요된다.

핵심용어

- Data Modeling

- ➡ 시스템에 필요한 Data를 중심으로 Entity, Attribute, Relationship 및 여러 요소를 도출하여 검증하는 과정

- ERD

- ➡ Entity Relationship Diagram으로 Data Modeling 후의 산출물을 말하며, Entity, Attribute 및 Relationship으로 구성됨



데이터베이스 모델링의 필요성 및 주요 요소

- Data Modeling의 목적 – 데이터의 정합성(수정,삭제,삽입 시 데이터일치)과 성능에 있음
- 성능과 데이터 정합성 – 모든 시스템에서 가장 중요한 요소
- 최적화된 Data Modeling을 통해 데이터 정합성과 성능 보장 가능한 시스템 구축을 위해서라도 필요



데이터베이스 모델링의 필요성 및 주요 요소

추세

- ▶ 고객사와 모델러, 프로젝트 리더들의 생각
- ▶ 데이터 양의 증가
- ▶ Data Modeling과 SQL Tuning에 대한 많은 관심 고조

모델링 주체

- ▶ 개발자, DBA, 업무 분석가/현업
- ▶ Data Modeling은 Modeler의 role을 가지고 있는 한 사람만이 주체로 하는것이 아님.
- ▶ 데이터와 업무에 관련이 있는 담당자들이 함께 참여해야 성공
- ▶ 관련된 사람들이 함께 참여해야만 Data 정합성 및 성능을 고려할 수 있음.

데이터베이스 모델링 순서 정리

1. 요구사항 분석

- ➡ 현업의 요구사항에 대한 내용을 수집 분석하여 개념 설계에 필요한 정보를 정리함.

2. 개념 설계

- ➡ Entity, Relationship 및 Attribute를 도출함.

3. Logical(논리) 설계

- ➡ 정규화(Normalization) 및 Entity통합을 수행함

4. Phisical(물리) 설계

- ➡ 데이터베이스로 구현하기 위해 Entity는 테이블로 Attribute는 컬럼으로 구현 함

.



1. 개념적 데이터베이스 모델링


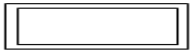


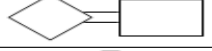






개념적 데이터 모델링

- 현재의 업무프로세스를 일반화 시킨다.
- 사각형 / 원형 / 다이아몬드 / 실선 등을 이용하여 ER-Diagram를 만든다.
- 업무적인 관점에서 접근 하고 분석하는 단계
- 관계형 데이터베이스 관점에서 하는 것은 아니다.

개념적 데이터 모델링

1.1 실체-관계 모델(Entity-Relationship Model)

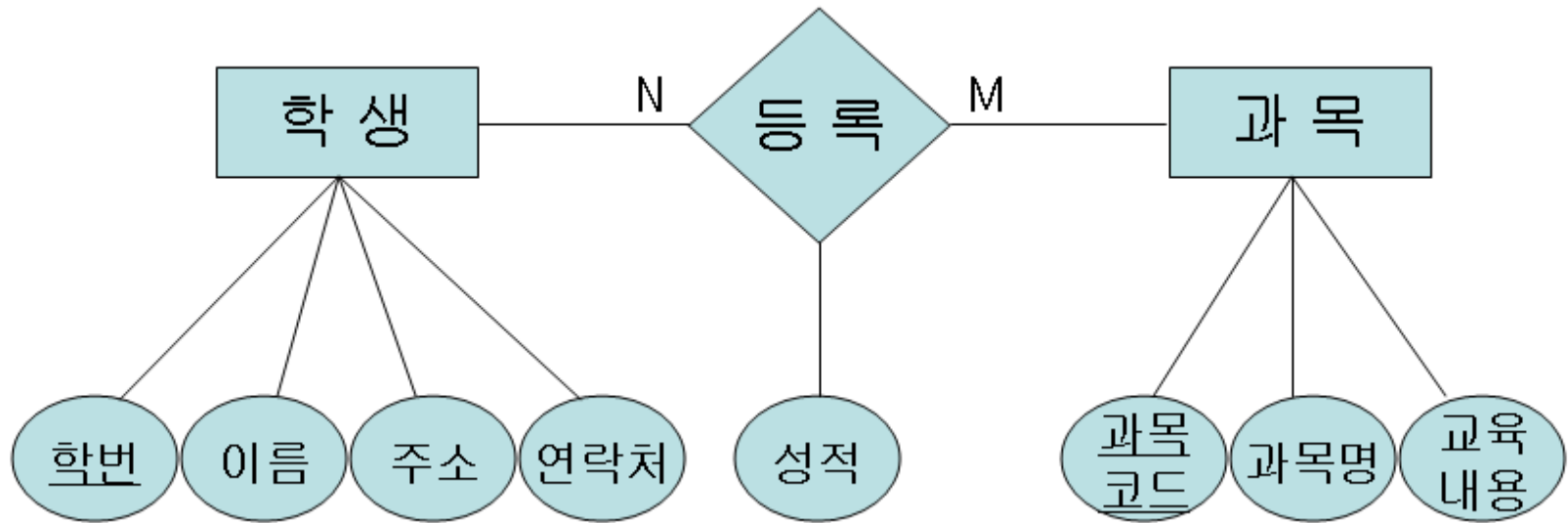
: 실체-관계 모델은 1976년 P. Chen이 제안한 것으로써 실체-관계 모델의 장점은 표현력이 풍부하고 직관적이어서 일반적으로 사람의 관점과 유사한 모델을 제공한다는 점이다.

기 호	의 미
	개 체 타입
	약한 개 체 타입
	관 계 타입
	식별 관계 타입
	전체 참여 개체 타입
	속성
	식별자 속성
	부분키 속성
	다중 값 속성
	복합 속성
	유도 속성

개념적 데이터 모델링

1.1 실체-관계 모델(Entity-Relationship Model)

: 실체-관계 모델을 통해서 표현된 결과물을 E-R Diagram이라고 한다.



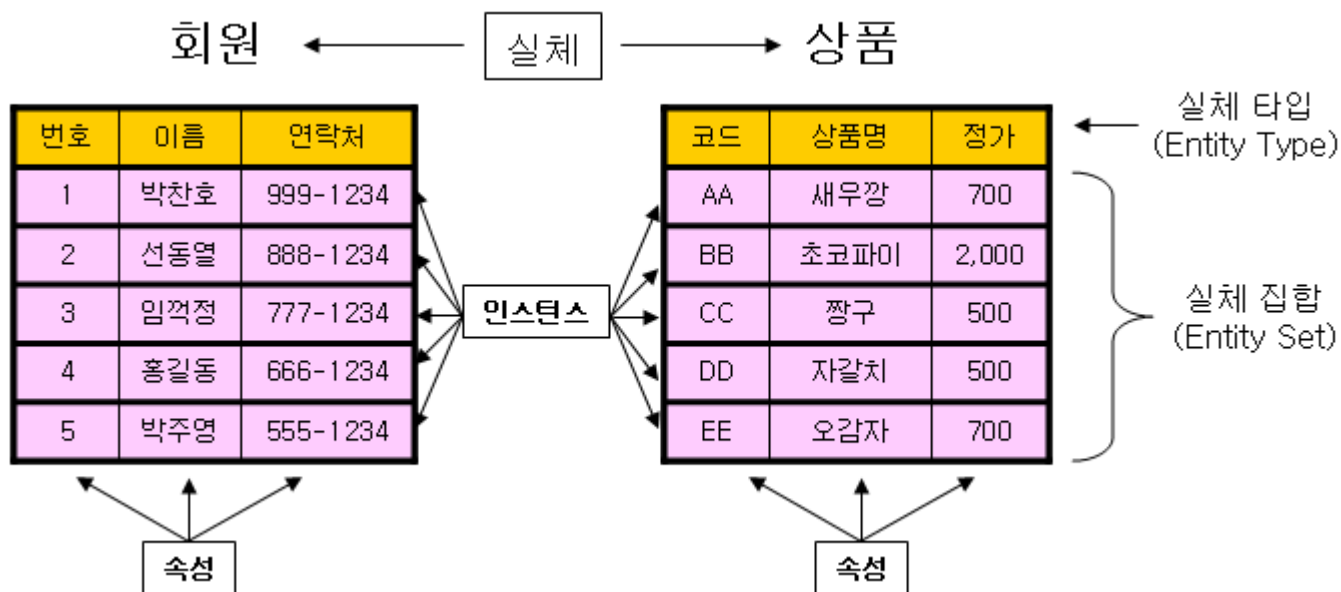
개념적 데이터 모델링

2.1 실체(Entity) 정의

: 현실 세계에서 다른 모든 것들과 구분되는 유형, 무형의 것을 실체라고 정의하며 업무 수행을 위해서 알아야 될 대상이 되는 유형, 무형의 것을 실체로 정의한다.

즉, 데이터로 관리되어야 하는 항목을 실체로 정의한다.

인스턴스(레코드)라 불리는 개별적인 객체들의 집합으로 반드시 주 식별자(PK)가 존재해야 한다.



개념적 데이터 모델링

2.2 실체(Entity) 파악 요령

- 관련 분야에 대한 지식 필요.
- 상식/논리/관찰력 필요.
- 문서를 중심으로 파악한다.
- 분석된 내용을 토대로 **명사 위주로 파악한다.**
- 담당자와의 인터뷰를 통해서 파악한다.

개념적 데이터 모델링

2) 엔티티 정의서 작성 - 엔티티 작성의 예

엔티티명	엔티티 설명	동의어/유의어	관련속성	비고
도서	인터넷을 통해 판매하고자 하는 책의 정보	책	도서번호 도서명	
회원	인터넷을 통해 등록한 회원의 정보	일반회원	주민번호 주소 전화번호 전자메일 핸드폰번호 결제방법 구매포인트	
주문	도서를 구매하기 위해 회원이 입력한 배송지, 결제방법에 관한 정보	주문서 주문내역	주문번호 주문일자 배송지주소 배송지전화번호 배송방법 결제방법	
주문목록	회원이 주문한 도서목록에 대한 수량 및 가격	구매도서목록	수량 단가	24

2.3 속성(Attribute) 정의

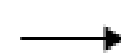
: 속성(Attribute)이란 정보의 요소로써 관리되는 항목으로 실체의 성질, 분류, 수량, 상태, 특성 등을 나타내는 세부 항목이다.

- * 한 실체에 정의되는 속성의 숫자는 10개 내외로 정의하자.
- * 속성의 명칭을 정의할 때에는 가독성이 높은 용어를 사용하자.
- * 속성을 정의할 때에 세분화해서 정의하도록 하자..
- * 식별자 업무규칙은 한 실체내의 인스턴스를 유일하게 구분 할 수 있어야 하며 모든 실체는 반드시 하나 이상의 식별자를 보유하여야 하며 복수개의 식별자도 보유 할 수 있다.

개념적 데이터 모델링



상품
상품명
제조사
정가
판매가
제조국가
...



실체(Entity)

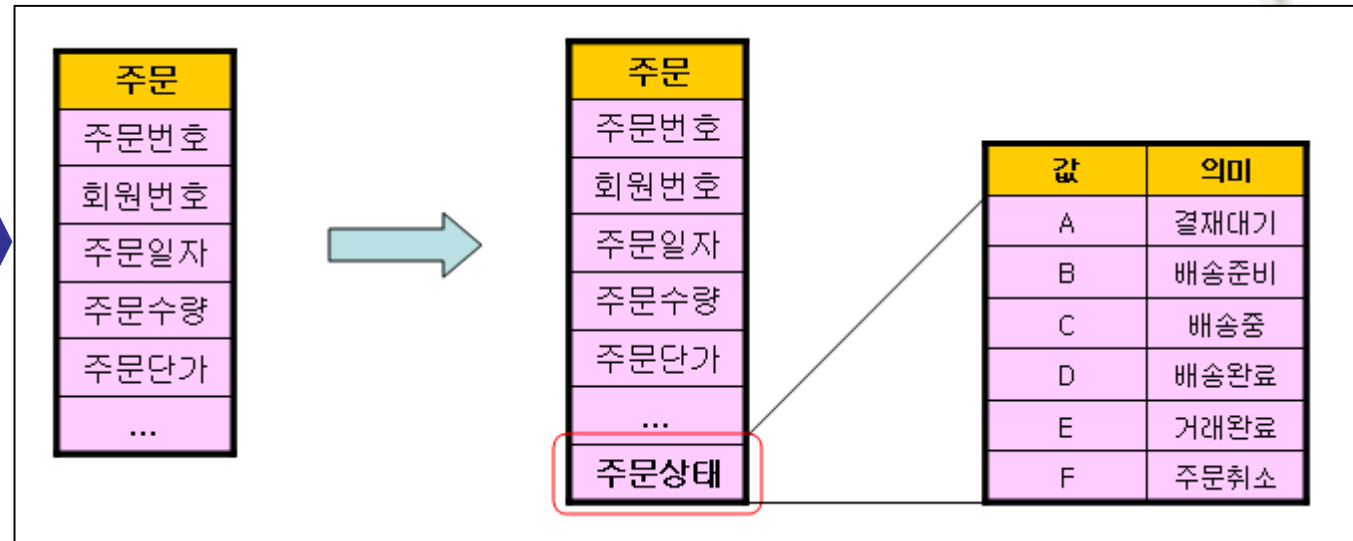
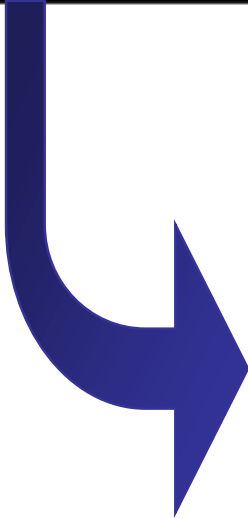


속성(Attribute)

개념적 데이터 모델링

2.4 속성(Attribute)의 유형

속성의 유형	의 미
기본 속성	해당 실체가 원래 가지고 있는 속성
추출 속성	기본 속성의 가공 처리를 통해서 얻을 수 있는 속성
설계 속성	시스템의 효율성을 위해 임의로 추가되는 속성



2.5 식별자(Identifier) 정의

: 식별자란 하나의 실체 내에서 각각의 인스턴스를 유일(Unique)하게 구분해 낼 수 있는 속성 또는 속성 그룹을 말하며, 하나의 실체는 하나 이상의 식별자를 반드시 보유하고 있어야만 한다.

2.6 식별자(Identifier)의 유형

1. 후보 식별자(Candidate Identifier)
2. 주 식별자(Primary Identifier)
3. 대체 식별자(Alternate Identifier)
4. 복합 식별자(Composite Identifier)
5. 대리 식별자(Surrogate Identifier)
6. 외래 식별자(Foreign Identifier)



개념적 데이터 모델링

1. 후보 식별자(Candidate Identifier)

- 실체 내에서 각각의 인스턴스를 유일하게 구분 할 수 있는 속성으로 기본키가 될 수 있는 후보 속성을 뜻한다.

예를 들어 사원 테이블에서 사원번호, 주민번호, 이름, 주소...등의 속성이 있다면 여기서 사원번호와 주민번호가 후보 식별자가 될 수 있다.



개념적 데이터 모델링

어떤 컬럼에 기본키가 정의되면 not null, no Duplicate속성이 적용되며 유니크한 클러스터드 인덱스가 만들어진다.

2. 주 식별자(Primary Identifier)

- 실체에서 각 인스턴스를 유일하게 식별하는 데 가장 적합한 키로 후보식별자 들 중에서 선택한 하나의 속성이나 속성의 그룹을 말한다.

- 후보키 중에서 기본키를 선정 할 때 고려사항

* 해당 실체를 대표 할 수 있을 것

* 업무적으로 활용도가 높을 것

* 길이가 짧은 것(조인 같은 것 할 때 사이즈 작은 컬럼을 비교하는 것이 속도 높다.)

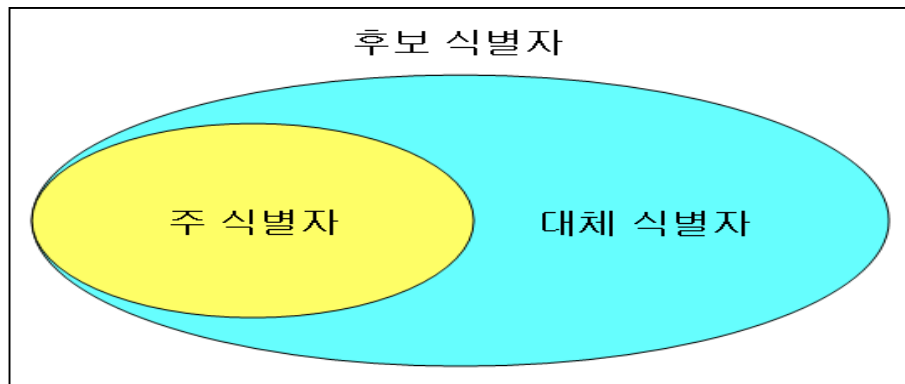
ex) 사원테이블의 후보키들 중 주민번호보다는 사원번호가 기본키에 적당.

개념적 데이터 모델링

3. 대체 식별자(Alternate Identifier)

- 후보키 중에서 기본키로 선정되지 않은 속성을 뜻함.
- 사원테이블에서 주민번호 컬럼에 이에 해당되며 인덱스를 설정할 때 고려 대상이 되기도 한다. 즉, 실체 인스턴스를 접근하는 데 자주 사용되는 Index로 활용하고자 할 때 대체키로 선정하면 된다.

* 후보키 / 기본키 / 대체키의 관계



개념적 데이터 모델링

4. 복합 식별자(Composite Identifier)

- 하나의 속성으로 기본키가 될 수 없는 경우 둘 이상의 컬럼을 묶어서 식별자로 정의하는 것.
- 이때 복합키 중에서 주로 조회의 조건으로 사용되는 컬럼을 먼저 정의 하는 것이 성능향상에 도움이 된다.

예) 급여내역

급여지급년월	사번	지급일자	급여액
201303	1	20130331	1500000
201303	2	20130331	1000000
201304	1	20130430	1620000
201304	2	20130430	1230000

개념적 데이터 모델링

4. 복합 식별자(Composite Identifier)

- 위의 급여내역 테이블에서 “누가 언제 얼마의 급여를 받았는가?”에 대한 정보를 관리 해야 한다면...

이때,

급여지급년월 기본키? - 중복

사번 기본키? - 중복

위 두 속성을 합쳐서 하나의 기본키로 만드는 것 복합키!

복합키로 만들 때 어떠한 속성을 먼저 작성?

- 실제 작업자 들이 어떠한 속성을 기준으로 보다 많은 조회를 하느냐에 따라 순서를 결정하는 것이 중요함.
- 조회프로세스가 비슷하다면 데이터가 입력되는 순서에 따라서 차례대로 저장 될 수 있는 속성으로 순서를 결정한다.

개념적 데이터 모델링

5. 대리 식별자(Surrogate Identifier)

- 식별자가 너무 길거나 여러 개의 속성으로 구성되어 있는 경우 인위적으로 추가한 식별자로 인공키 라고도 부른다.
- 식별자가 여러 속성으로 구성된 경우는 데이터를 수정하거나 검색을 하는 데 수행 속도가 떨어 질 수 있다.
- 대리키는 역정규화 작업의 한 가지 방법론이다.

(**역정규화란** 정규화 이론에 입각해 모델링을 하다 보면 데이터를 검색하는 속도가 떨어 질 수 있으므로 작업의 수행속도 향상을 위해서 정규화 이론에 위배되는 설계를 임의로 하는 것을 말함.)



개념적 데이터 모델링

예) 급여내역 테이블

급여지급년월	사번	지급일자	급여액
201303	1	20130331	1500000
201303	2	20130331	1000000
201304	1	20130430	1620000
201304	2	20130430	1230000

급여내역 실체의 식별자는 급여지급년월 + 사번을 합쳐 복합키로 구성되어 있음. 이때, 급여내역 실체가 자식 실체를 가지고 있다면 자식 실체에 복합키가 FK로 전이되어 질 것 이며 이러한 경우 연관된 정보를 보기 위해 조인의 조건을 여러 번 작성 하게 되므로 속도가 느려진다. 이러한 경우 대리키를(일련번호) 를 두어 조인의 속도를 향상 시킬 수 있다)

개념적 데이터 모델링

예) 급여내역 테이블

일련번호	급여지급년월	사번	지급일자	급여액
1	201303	1	20130331	1500000
2	201303	2	20130331	1000000
3	201304	1	20130430	1620000
4	201304	2	20130430	1230000

* 위와 같이 일련번호를 PK로 만들어 줌으로써 자식 엔티티로 전이되는 Foreign key가 단일 해지기 때문에 조인 속도를 향상시킬 수 있다.



개념적 데이터 모델링

6. 외래 식별자(Foreign Identifier)

- 부모 엔티티의 기본키를 자식 엔티티 에서 참조하는 것.
- 참조되는 부모의 속성값 이외의 값은 자식 엔티티의 속성값이 될 수 없음.
- 기본적으로 null 허용 / 중복 가능 함.



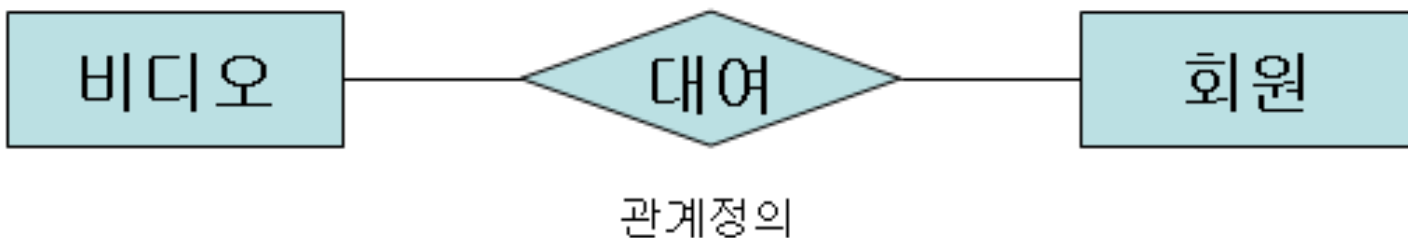
개념적 데이터 모델링

2.7 관계(Relationship) 정의

: 관계란 두 실체 사이에 인스턴스들, 즉 실체 집합 사이에 대응되는 연관성을 정의하는 것으로 쉽게 표현해서 **두 실체 간의 업무적인 연관성**이라고 정의할 수 있으며, 이러한 관계를 정의할 때 동사구로 관계를 정의한다.

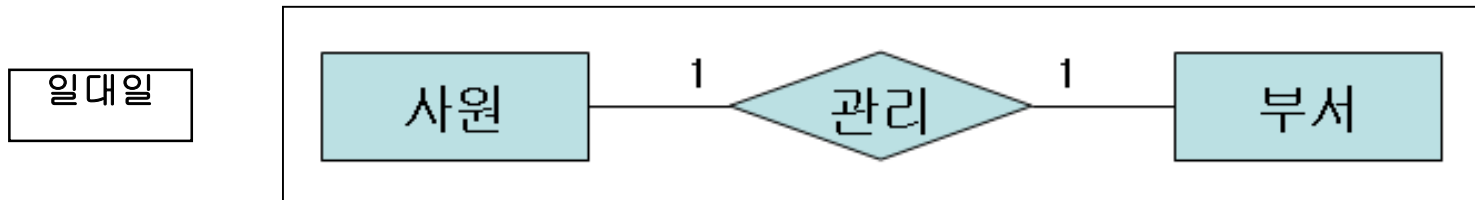
* 관계 정의 방법

1. 관계가 있는 **두 실체를 실선으로 연결**하고, **관계를 부여**한다.
2. **관계 차수**를 표현한다.
3. **선택성**을 표시한다.



개념적 데이터 모델링

2.8 차수성 정의



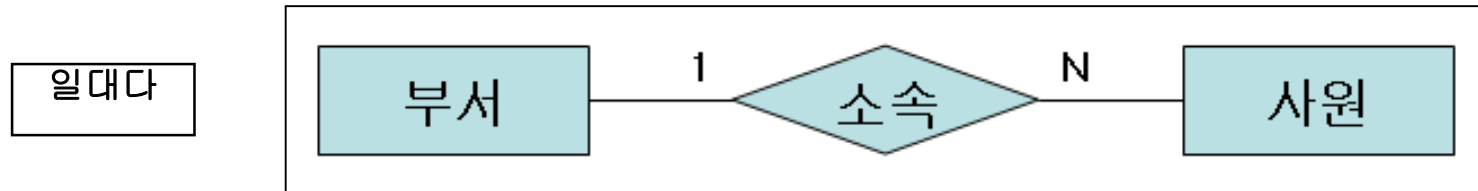
예) 각각의 부서를 관리하는 부서장에 대한 관계를 나타내고 있다.

좌 -> 우 : 직원 중 에서 어떤 부서를 관리하는 부서장이 존재하는데 한 명의 부서 밖에는 관리 할 수 없다.

우 -> 좌 : 한 부서에는 한 명의 관리 사원이 존재한다.

개념적 데이터 모델링

2.8 차수성 정의



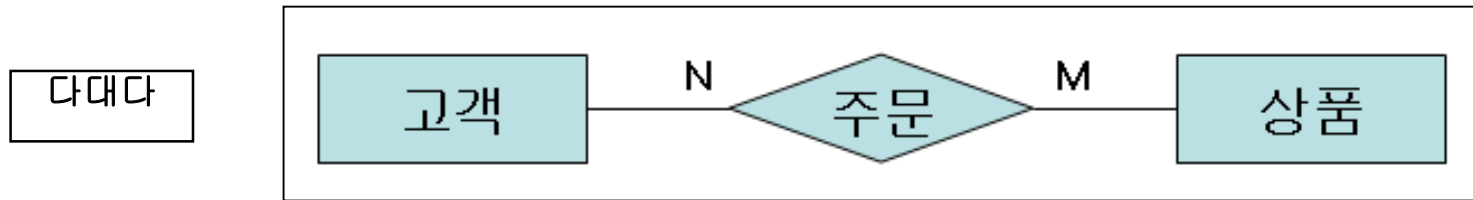
예) 부서와 사원 사이에는 소속이라는 관계가 성립한다.

좌 -> 우 : 하나의 부서에는 한 명 이상의 사원이 배치되어진다.

우 -> 좌 : 한 명의 사원은 단 하나의 부서에 소속되어 진다.

개념적 데이터 모델링


2.8 차수성 정의



예) 고객과 상품 실체간에는 주문이라는 관계가 성립한다.

좌 -> 우 : 한 명의 고객은 하나 이상의 상품을 주문 할 수 있다.

우 -> 좌 : 한 상품은 한 명 이상의 고객에 의해 주문되어 질 수 있다.

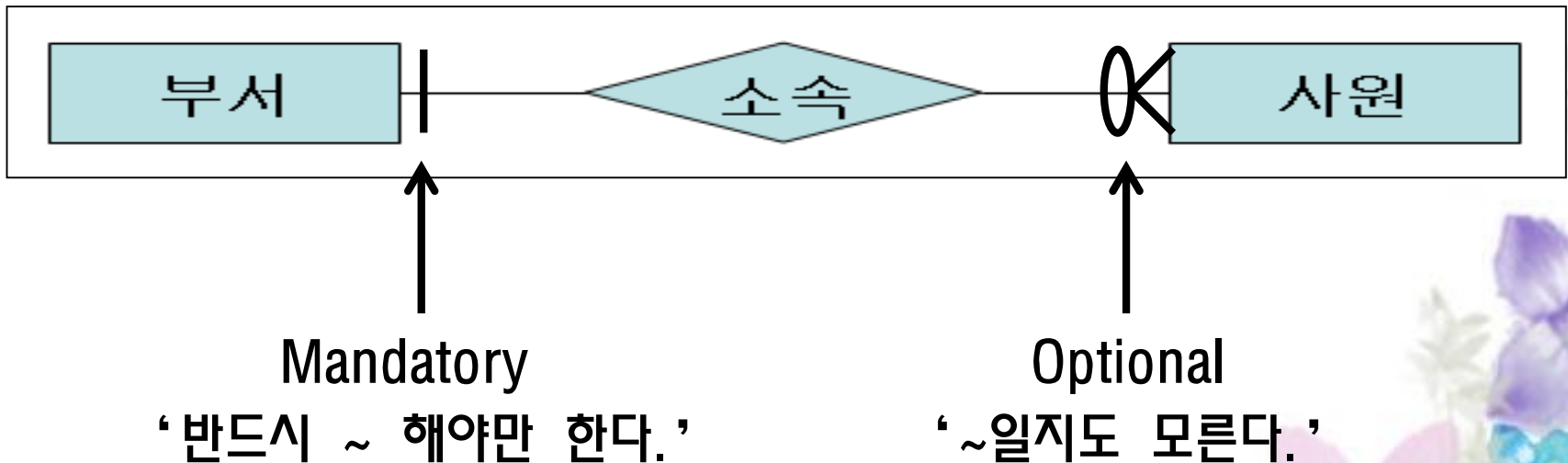
 M:N의 관계는 이렇게 이론상으로 존재하지만 실제 데이터베이스에서는 1:1 과 1:N관계만 존재한다. 이는 논리적 데이터베이스 모델링단계를 거치면서 1:N의 관계로 해결되어 진다.

개념적 데이터 모델링

2.9 선택성

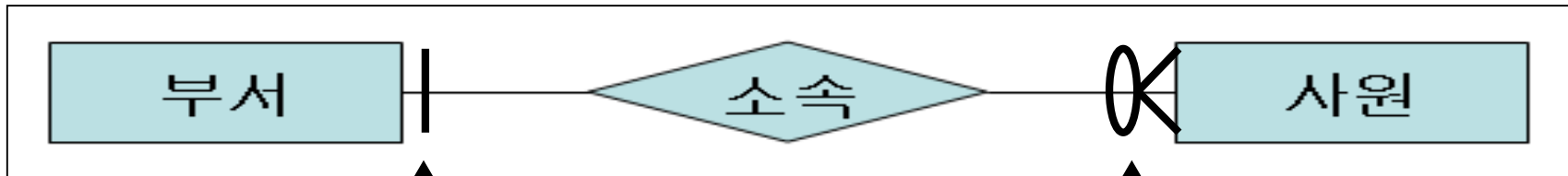
- 두 실체간의 관계가 설정 되었을 때 항상 두 실체의 모든 인스턴스 간에 관계가 존재해야 하는지 아니면 모든 인스턴스에 대해 존재 할 필요가 없는지를 나타내는 부분.

예) 부서와 사원 간의 소속 관계 Diagram



개념적 데이터 모델링

예) 부서와 사원 간의 소속 관계 Diagram



Mandatory

‘반드시 ~ 해야만 한다.’

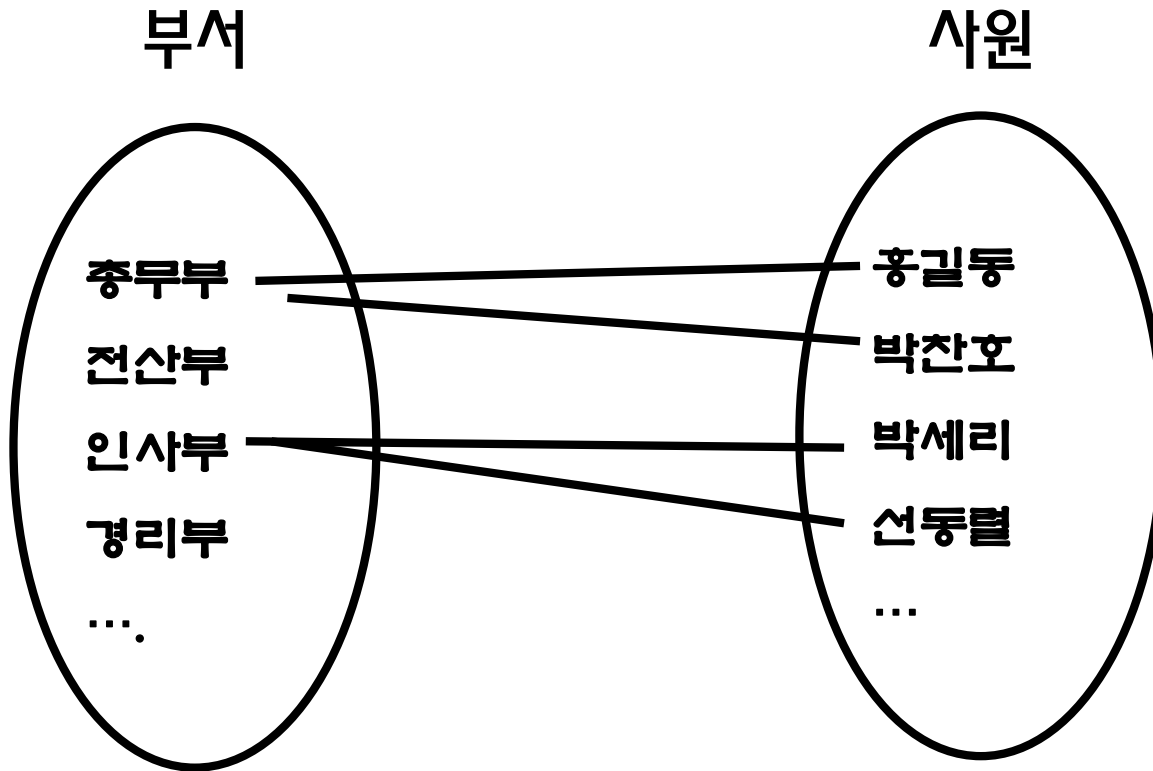
Optional

‘~일지도 모른다.’

좌 -> 우 : 각 부서는 하나 이상의 사원을 배치 받을지도 모른다.

우 -> 좌 : 사원은 하나의 부서에 반드시 소속되어야 한다.

개념적 데이터 모델링



좌 -> 우 : 각 부서는 하나 이상의 사원을 배치 받을지도 모른다.

우 -> 좌 : 사원은 하나의 부서에 반드시 소속되어야 한다.

개념적 데이터 모델링

부서코드	부서명
100	총무부
200	전산부
300	인사부
400	경리부

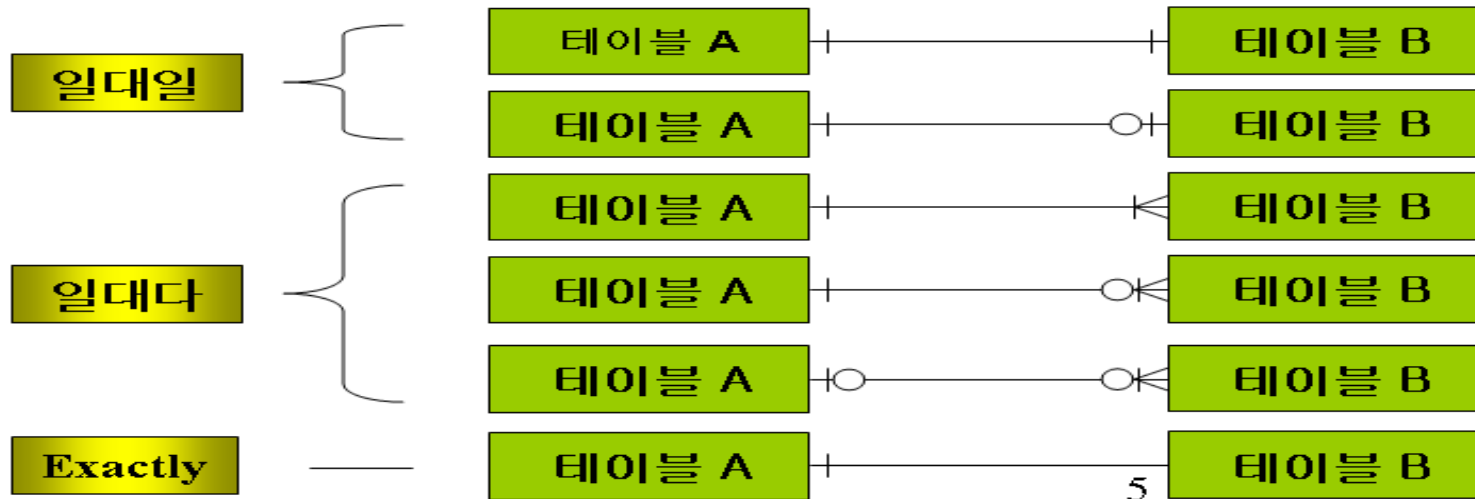
사원번호	사원이름	부서코드
1	홍길동	100
2	박찬호	100
3	박세리	300
4	선동렬	300



개념적 데이터 모델링

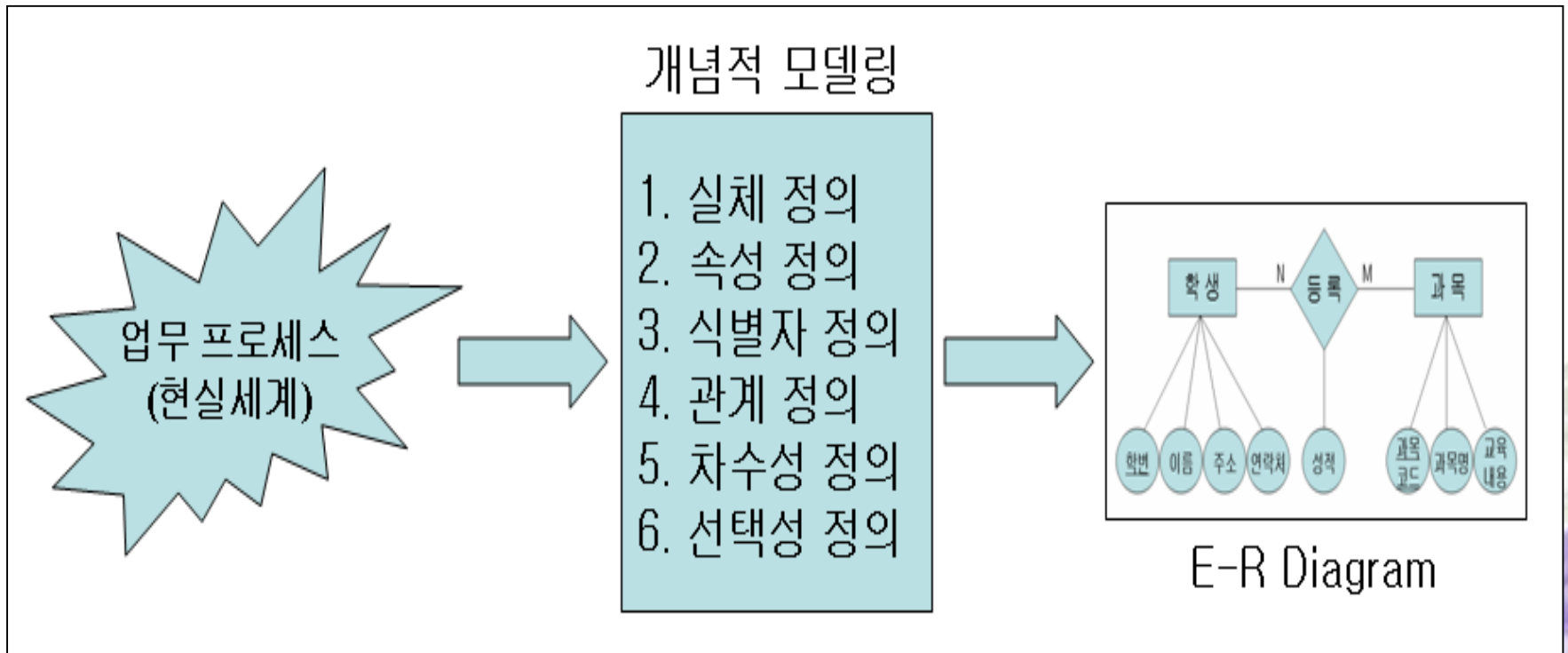
2.10 관계의 표현 방식

: 정보공학(Information Engineering)표기방식



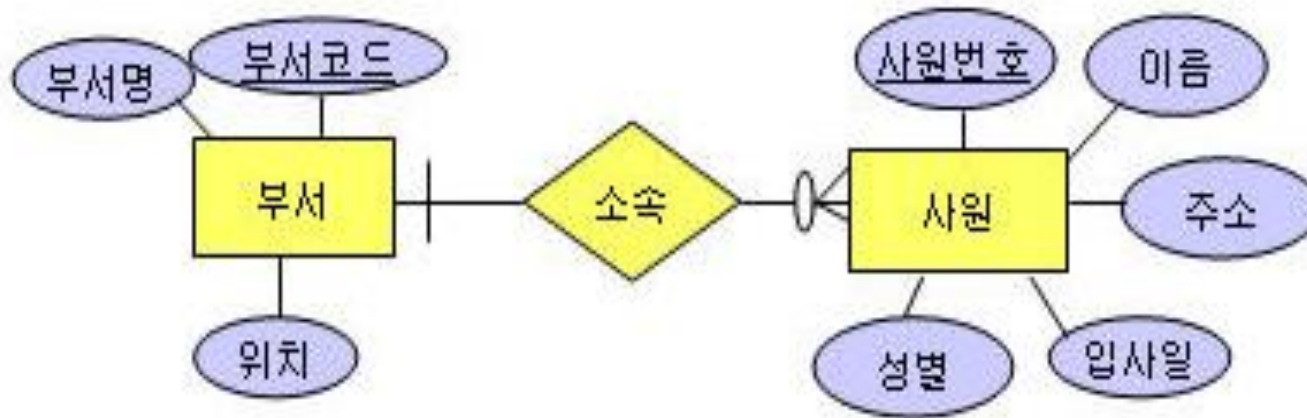
개념적 데이터 모델링

* 개념적 데이터 모델링 정리



개념적 데이터 모델링

예)



1단계 > 업무 분석 : 각 사원의 소속 부서에 대한 정의를 내리고자 한다.

2단계 > 실체 도출 : 부서, 사원

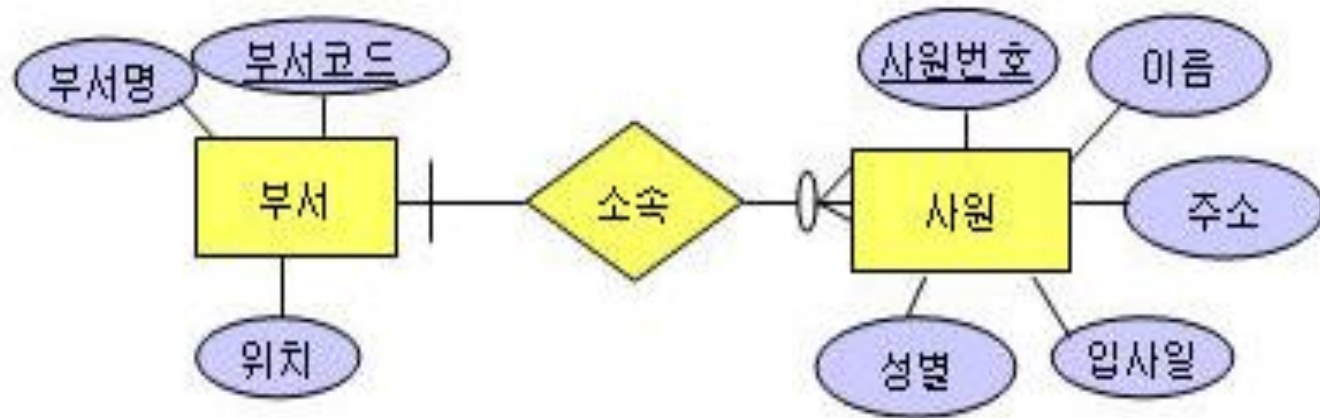
3단계 > 속성 도출 : 사원 - 사원번호, 이름, 주소, 성별, 입사일

부서 - 부서번호, 부서명, 위치

4단계 > 속성중 식별자 도출 : 사원 - 사원번호

부서 - 부서번호

개념적 데이터 모델링



5단계 > 관계 설정 :

부서에는 사원이 배치되어진다.
사원은 부서에 소속되어진다.

6단계 > 차수 관계 설정

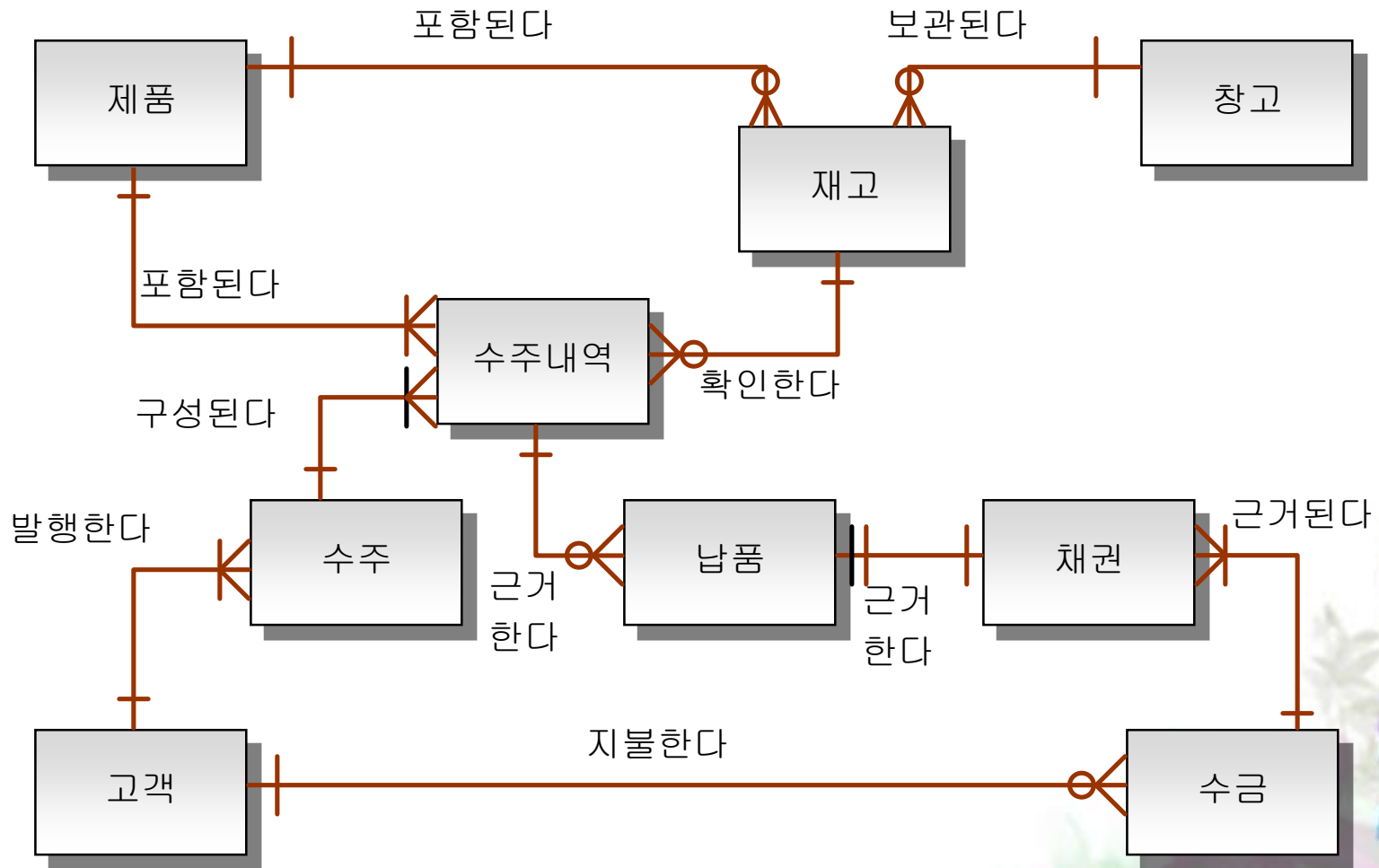
각 부서에는 하나 이상의 사원이 배치되어질 지도 모른다.
각 사원은 단 하나의 부서에 소속되어진다.
=> 일:다(1:N) 관계가 성립

7단계 > 관계 식별 여부 파악

각 부서에는 하나 이상의 사원이 배치되어질 지도 모른다.
각 사원은 반드시 하나의 부서에 소속되어진다.

개념적 데이터 모델링

엔티티 관계도의 예



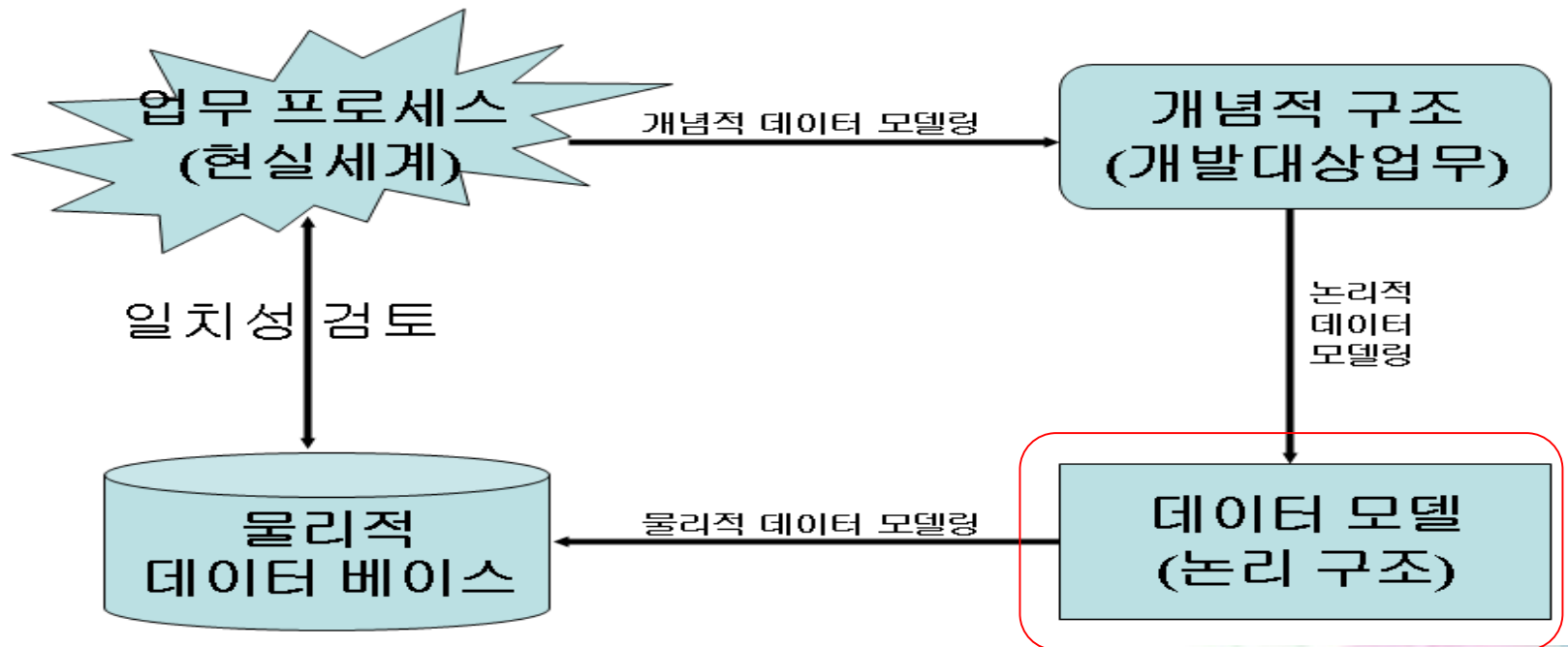
2. 논리적 데이터베이스 모델링

논리적 데이터 모델링

1. 논리적 데이터베이스 모델링에 대한 이해

1.1 논리적 데이터베이스 모델링이란?

: 논리적 데이터베이스 모델링 단계부터는 관계형 데이터베이스 이론이 적용되는 단계로써, 실제 DBMS를 고려하지 않은 상태로 데이터베이스 내의 스키마를 정의하는 단계이다.



논리적 데이터 모델링

1.2 관계형 모델의 개념

- 관계란 업무적인 연관성, 관계는 두 테이블 사이에서 존재하며 한 개는 반드시 부모, 한 개는 자식, 부모의 pk가 자식의 fk로 전이된다.
- 조인의 조건대상이 된다.
- 주체관계를 파악해서 부모와 자식을 선별
- 부모테이블의 데이터가 먼저 정의되어야 한다.



논리적 데이터 모델링

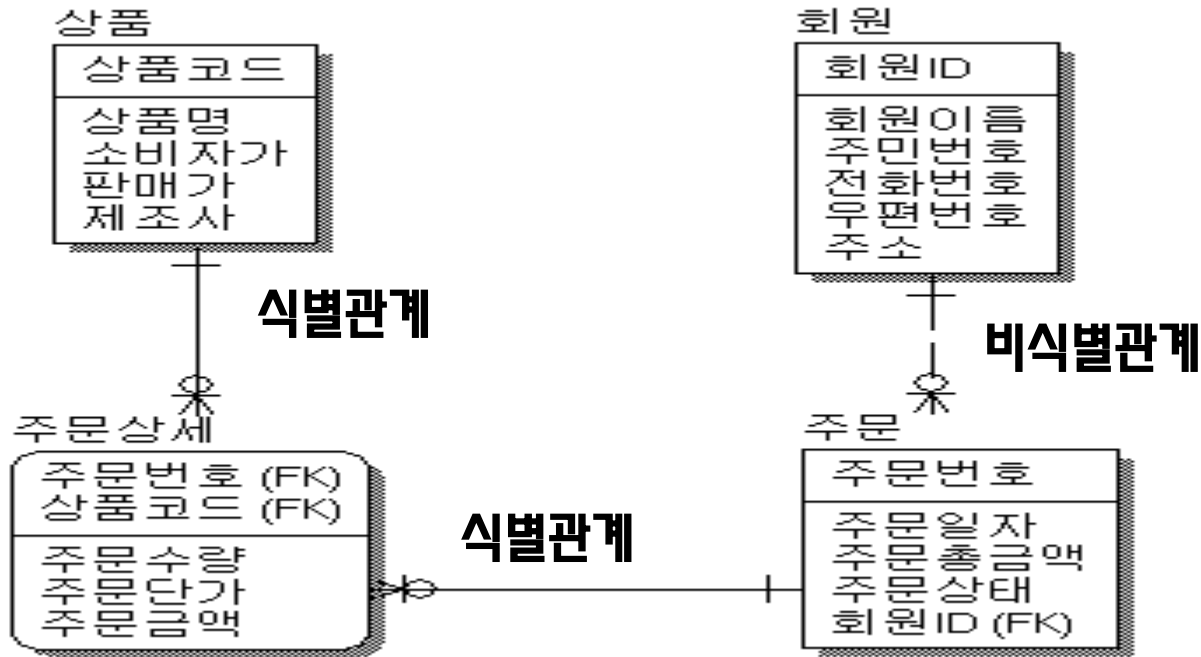
1.3 관계유형

- ① 식별관계 - 부모테이블의 기본키가 자식 테이블의 **기본키** 혹은 **기본키그룹**의 구성원으로 전이된다.
- ② 비식별관계 - 부모테이블의 기본키가 자식테이블의 **일반속성 그룹**의 구성원 으로 전이된다.



논리적 데이터 모델링

예)



회원테이블과 주문 테이블의 관계는 주문테이블의 회원ID(포린키)컬럼이 일반 속성 그룹의 구성원으로 전이 되었기 때문에 **비식별관계** 이다.

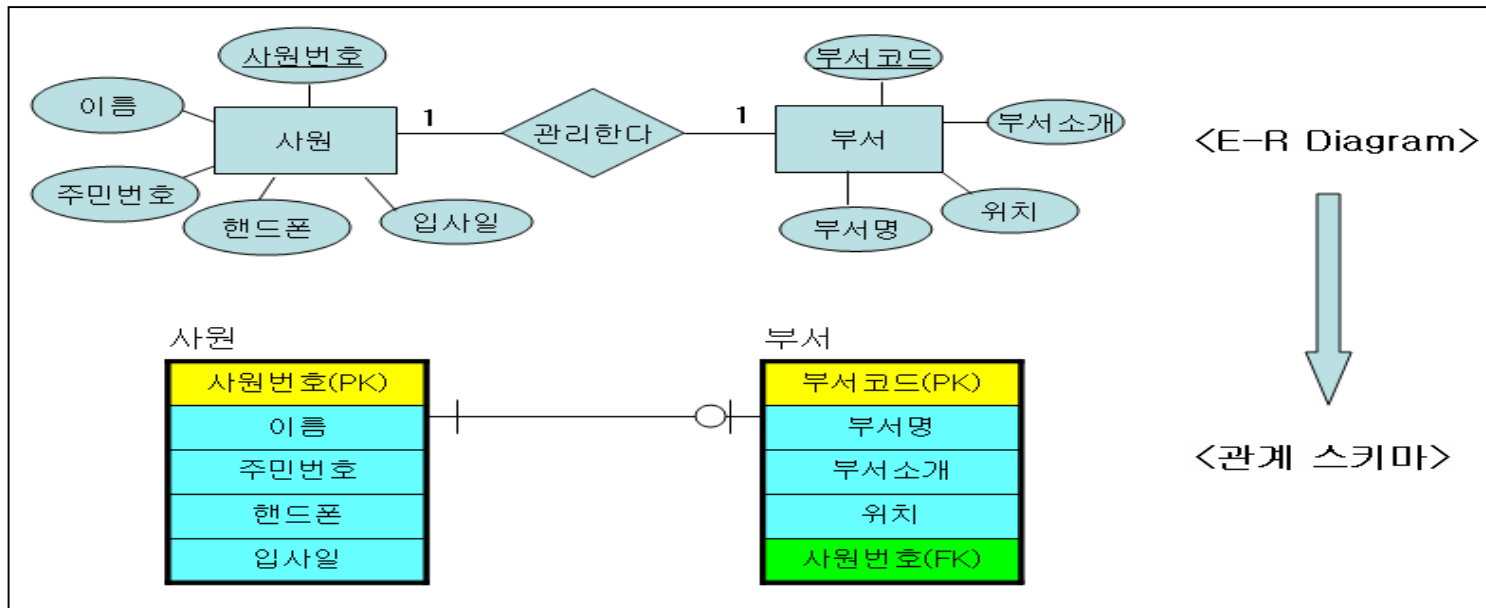
반면, 주문상세테이블과 상품테이블의 기본키는 주문상세(자식)테이블의 기본키 그룹의 구성원으로 전이되었기 때문에 **식별관계**이다.

논리적 데이터 모델링

2. 매핑룰(Mapping Rule)

- 4단계) 관계를 포린키로 대응시키기 위해서는 관계 차수에 따라 세가지 경우로 나누어 적용시켜야 한다.

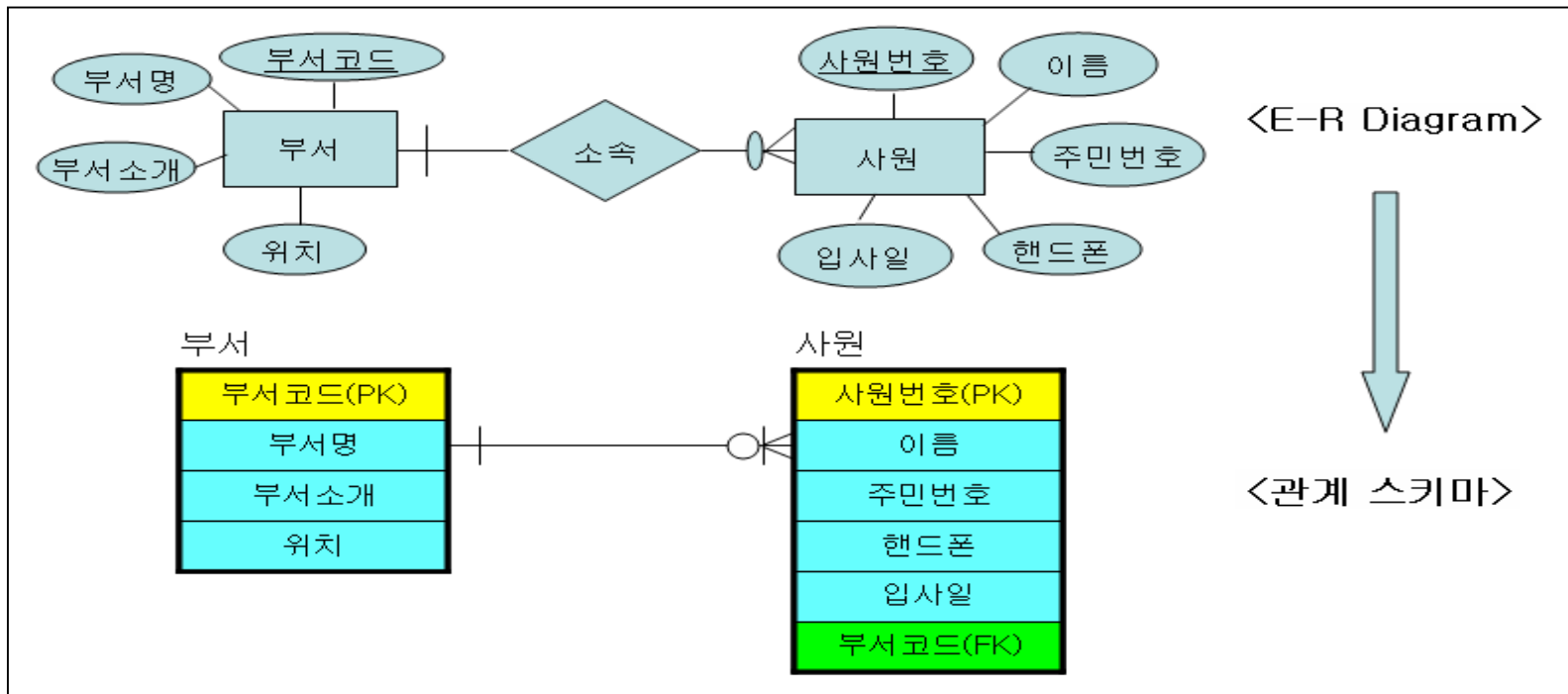
① **일 대 일 관계** - 왼쪽 엔티티의 기본키를 오른쪽 엔티티로 전이 시키거나 또는 오른쪽 엔티티의 기본키를 왼쪽 엔티티로 전이 시킴으로써 두 엔티티 간의 관계를 해결 할 수 있다.



논리적 데이터 모델링

2. 메핑룰(Mapping Rule)

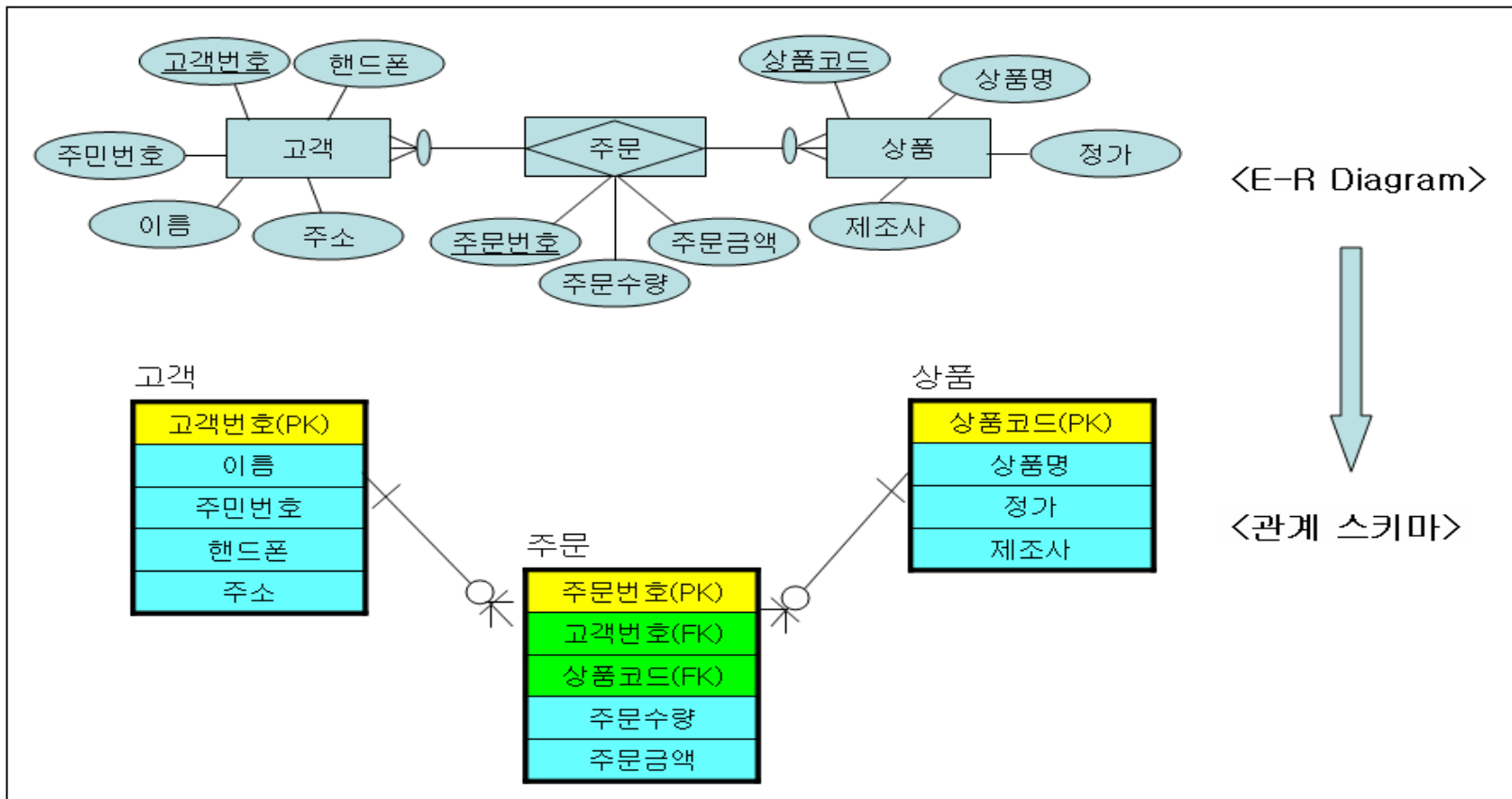
② 일 대 다 관계 - 일(1) 쪽의 기본키를 다(N)쪽의 관계스키마에 포린키로 전이시키는 것이 자료의 중복성을 해결 할 수 있는 효율적인 방법이다.



논리적 데이터 모델링

2. 메핑룰(Mapping Rule)

③ 다 대 다 관계 - 교차 엔티티 즉, 두 엔티티 간의 관계정보를 표현 할 수 있는 새로운 관계스키마를 도출.



논리적 데이터 모델링

2. 메핑룰(Mapping Rule)

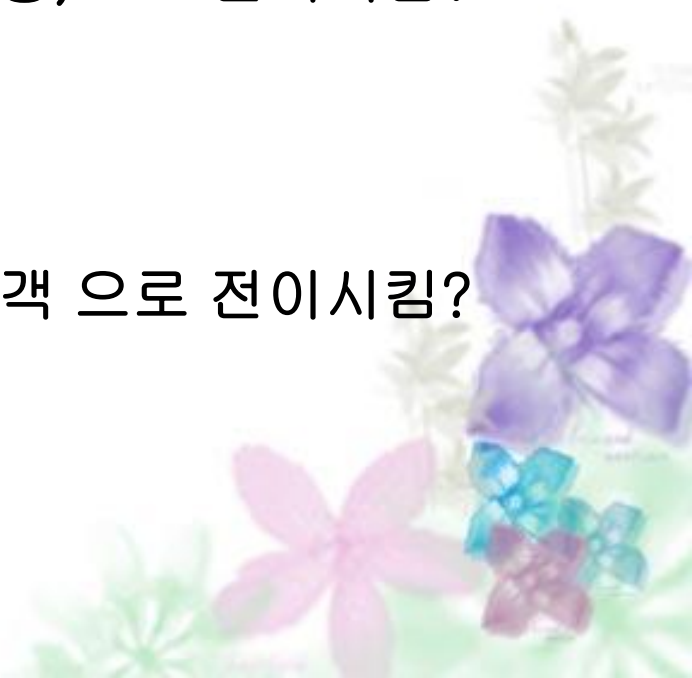
③ 다 대 다 관계 - 예

‘한 고객은 한 개 이상의 차량을 구입할 수 있고, 한가지 차량은 여러 고객에게 팔 릴 수 있다.’ 고객 과 차량은 N : M의 관계

* 논리적데이터 모델링 단계에서 어떻게 관계 스키마 정의?

- 좌 -> 우 : 고객의 기본키를 상품(차량)으로 전이시킴?

- 우 -> 좌 : 상품(차량)의 기본키를 고객 으로 전이시킴?



논리적 데이터 모델링

테이블 명 : 고객

컬럼명	고객ID	고객이름	고객주소	전화번호
키형태	PK			
견본데이터	HONG	홍길동	서울	222-2222
테이블 명 : 상품	PARK	박찬호	대전	333-3333

컬럼명	차량번호	차량이름	색상	배기량	고객ID
키형태	PK				FK
견본데이터	1	소나타	파랑	1000	HONG
	1	소나타	파랑	1000	PARK
	2	티코	노랑	2000	PSR
	3	그렌저	보라	2100	HONG



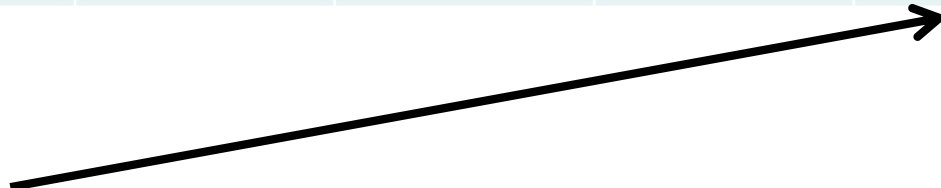
논리적 데이터 모델링

테이블 명 : 고객

컬럼명	고객ID	고객이름	고객주소	전화번호	차량번호
키형태	PK				FK
견본데이터	HONG	홍길동	서울	222-2222	1
	HONG	홍길동	서울	222-222	3
	PARK	박찬호	대전	333-3333	1
	PSR	박세리	대구	444-444	2

테이블 명 : 상품

컬럼명	차량번호	차량이름	색상	배기량
키형태	PK			
견본데이터	1	소나타	파랑	1000
	2	티코	노랑	2000
	3	그렌저	보라	2100



논리적 데이터 모델링

테이블 명 : 고객

컬럼명	고객ID	고객이름	고객주소	전화번호
키형태	PK			
견본데이터	HONG	홍길동	서울	222-2222
	PARK	박찬호	대전	333-3333
	PSR	박세리	대구	444-444

테이블 명 : 상품

컬럼명	차량번호	차량이름	색상	배기량
키형태	PK			
견본데이터	1	소나타	파랑	1000
	2	티코	노랑	2000
	3	그랜저	보라	2100

테이블 명 : 계약

컬럼명	고객ID	차량번호	계약일	계약금
키형태	PK,FK	PK,FK		
견본데이터	HONG	1	20130301	1000
	PARK	1	20130402	2000
	PSR	2	20130401	1600
	HONG	3	20130601	1500

논리적 데이터 모델링

정규화

◆ 정규화 정의

- 1972년 E.F. CODD 박사에 의해 제안된 이론으로 실 세계에서 발생하는 데이터를 수학적 방법에 의해 구조화시켜 체계적으로 관리할 수 있도록 한 이론.

◆ 정규화 목적

- 정보의 중복을 최소화
- 정보모형의 단순화
- 정보 공유도 증대
- 정보의 일관성 확보
- 정보 품질 증대




논리적 데이터 모델링

정규화

◆ 정규화 필요성

- 엔티티를 구성하는 속성간의 중복을 제거하여 데이터베이스를 최적화
- 속성간의 함수종속성에 의해 발생하는 이상현상을 제거

 정규화를 통해 Data 감소 및 중복된 Data 제거는 가능하지만 조인의 증가로 성능 저하 발생

이상현상	내 용
입력이상	데이터 입력 시 필요 없는 속성까지 입력해야 하는 현상
수정이상	데이터 수정 시 원하지 않는 데이터까지 수정되는 현상
삭제이상	데이터 삭제 시 필요한 데이터까지 삭제되는 현상

논리적 데이터 모델링

3. 정규화(Normalization)

3.1 제 1 정규형

: 반복되는 속성이나 그룹의 속성을 제거하고, 새로운 실체를 추가한 후에 기존의 실체와 일대다의 관계를 형성한다.

3.2 제 2 정규형

: 복합키(Composit Primary Key)로 구성된 경우 해당 테이블 안의 컬럼들은 복합키 전체에 의존적이어야 한다.

만일 복합키 일부에 의존적인 컬럼이 존재한다면 이를 제거해야 한다.

3.3 제 3 정규형

: 한 테이블 안의 기본 키가 아닌 컬럼들은 기본키(Primary Key)에 의존해야 한다.
만일 기본키가 아닌 컬럼에 종속되는 속성이 존재한다면 이를 제거해야 한다.

* **정규화(Normalization)**란 속성이 제 위치에 제대로 위치하게끔 정의하는 과정을 말하며 정규형(Normal Form)이란 정규화를 수행하는데 있어서의 규칙을 말한다.

논리적 데이터 모델링

1차 정규화

- 반복되는 속성이나 그룹의 속성을 제거하고, 새로운 엔티티를 추가한 후에 기존의 엔티티와 일대다의 관계를 형성

예) 하나의 제품에 대해 여러 개의 주문서가 접수된 내용

주문목록

제품번호	제품명	재고 수량	주문 번호	수출 여부	고객 번호	사업자 번호	우선 순위	주문 수량
1001	모니터	1990	AB345	X	4520	398201	1	150
			AD347	Y	2341		3	600
1007	마우스	9702	CA210	X	3280	200212	8	1200
			AB345	X	4520	398201	1	300
			CB230	X	2341	563892	3	690
1201	스피커	2108	CB231	Y	8320		2	80

주문목록

제품번호
제품명
재고수량
주문번호
수출여부
고객번호
사업자번호
우선순위
주문수량

논리적 데이터 모델링

1차 정규화

제품에 관련된 정보로부터 반복되어 생성되는 주문 관련 정보를 분리함으로써 1차 정규화를 수행

제품

제품번호	제품명	재고수량
1001	모니터	1990
1007	마우스	9702
1201	스피커	2108

제품

제품번호
제품명
재고수량

주문목록

제품번호(FK)
주문번호
수출여부
고객번호
사업자번호
우선순위
주문수량

주문목록

제품번호	주문번호	수출여부	고객번호	사업자번호	우선순위	주문수량
1001	AB345	X	4520	398201	1	150
1001	AD347	Y	2341		3	600
1007	CA210	X	3280	200212	8	1200
1007	AB345	X	4520	398201	1	300
1007	CB230	X	2341	563892	3	690
1201	CB231	Y	8320		2	80

논리적 데이터 모델링

1차 정규화 예)

만약, 홍길동 사람이 자격증을 3개 갖고 있다면 홍길동 정보를 입력하기 위하여 3개의 레코드가 필요하다.

테이블 명 : 회원

컬럼 명	회원번호	이름	주소	핸드폰번호	주민번호	자격증	취득일
키 형태	PK						
건본 데 이 터	1	홍길동	서울시 면 목 8동	(111)111-1111	111111-111111	운전면허1종	19980203
	1	홍길동	서울시 면 목 8동	(111)111-1111	111111-111111	MCSE	19990603
	1	홍길동	서울시 면 목 8동	(111)111-1111	111111-111111	정보처리기사	20000502



논리적 데이터 모델링

1차 정규화 예)

반복되어지는 그룹의 속성을 제거한 후 새로운 테이블을 생성하고 기존의 테이블과 1 : N의 관계를 형성한다.

테이블 명 : 회원

컬럼 명	회원번호	이름	주소	핸드 폰 번호	주민번호
키 형태	PK				
견본 테 이 터	1	홍길동	서울시 면목 8동	(111)111-1111	111111-111111

테이블 명 : 자격증

컬럼 명	자격증ID	회원번호	자격증	취득일
키 형태	PK	FK		
견본 테 이 터	1	1	운전면허 1종	19980203
	2	1	MCSE	19990603
	3	1	정보처리 기사	20000502

논리적 데이터 모델링

1차 정규화 예)

만약, 학원에서 진행 중인 과정들의 데이터를 과정이라는 테이블로 관리한다고 했을 때, 각 과정에 대해 소요되는 교재를 과정 테이블에서 컬럼으로 관리한다고 한다면 아래와 같이 많은 null이 들어간다.

테이블 명 : 과정

과정 코드	과정 명	교육내용	교육 기간	강의 시간	수강료	교재 1	교재 2	교재 3	교재 4	교재 5
PK										
A0001	웹프로그래밍	웹 구현에 필요한 기본 구현 로직과 프	5개월	8시간	1800000	SQL Sever	Htmi활용	ASP기초	JSP활용	실무 구축
A0002	웹마스터	웹 사이트를 구축할 수 있	4개월	4시간	700000	Windows 2000	SQL Server	NULL	NULL	NULL
A0003	ASP프로그래밍	ASP를 활용한 웹 사이트 구	3개월	8시간	1400000	Htmi기초	ASP활용	NULL	NULL	NULL
J0001	JAVA프로그래밍	Java프로그래머가 되기 위	9개월	8시간	2000000	SQL Server	JAVAR기초	JSP	NULL	NULL

논리적 데이터 모델링

1차 정규화 예)

교재 컬럼처럼 반복되는 속성을 갖는 컬럼의 경우 따로 분리해서 기존의 테이블과의 1:N의 관계를 형성한다.

테이블 명 : 과정

컬럼 명	과정코드	과정명	교육내용	교육기간	강의 시간	수강료
키 형태	PK					
건본 데 이 터	A0001	웹 프로그래밍	웹 구현에 필 요한 기본 현 로직과 프	5개월	8시간	1800000
	A0002	웹마스터	웹 사이트를 구축할 수 있	4개월	4시간	700000
	A0003	ASP프로그래밍	ASP를 활용한 웹 사이트 구	3개월	8시간	1400000
	J0001	JAVA프로그래밍	Java프로그래 머가 되기 위	9개월	8시간	2000000

테이블 명 : 교재명

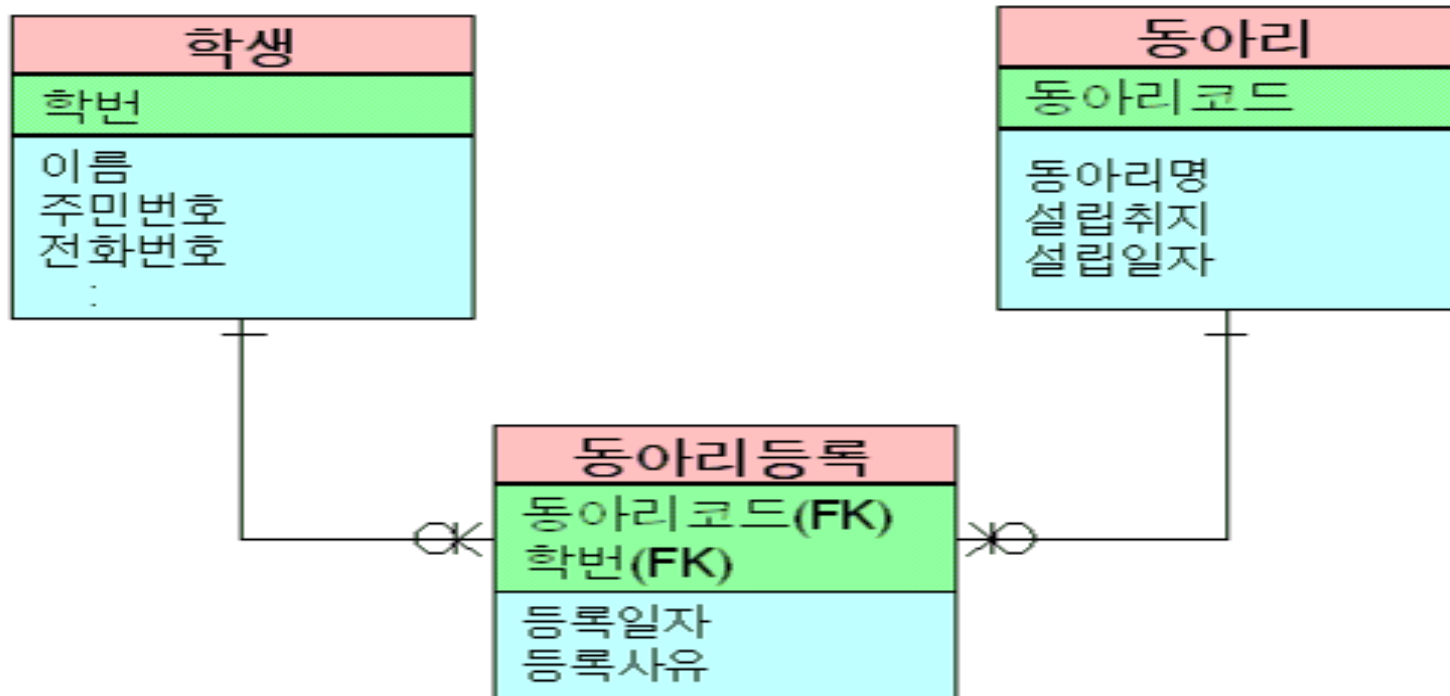
컬럼 명	교재번호	교재명	과정코드
키 형태	PK		FK
건본 데 이 터	1	SQL Sever	A0001
	2	Htm1활용	A0001
	3	ASP기초	A0001
	4	JSP활용	A0001

	9	JAVAR기초	J0001
	10	JSP	J0001

논리적 데이터 모델링

◆ 2차 정규화

복합키(composit PK)로 구성된 경우 해당 테이블 안의 모든컬럼 들은 복합키 전체에 의존적이어야 한다. 만일, 복합키 일부에 의존적인 컬럼이 존재한다면 제거 해야 한다. 복합키가 아닌 경우 제 2 정규화의 대상이 되지 않는다.



논리적 데이터 모델링

2차 정규화

동아리등록 테이블

동아리코드	학번	등록일자	등록사유
A	1	20130301	놀이
A	2	20130301	공부
B	3	20130310	놀이
C	1	20130405	친목

의존적이다 => 등록일자는 어떤 학생이 어떤 동아리에 등록한 날짜를 알려준다.
등록사유는 어떤 학생이 어떤 동아리에 어떤 이유로 가입했다.
그러므로 일자,사유는 복합키에 의존적이다.

* 의존적이지 않는 것은 만약, 동아리에 가입할 때 전화번호, 핸드폰, 주소 등의 속성이 있다면 이에 대한 정보는 학번에만 의존적이므로 복합키에 의존적인 게 아니기에 2차 정규화 할 때 해당 테이블 안에 비의존적인 애들은 제거한다. 이러한 것들은 학생 테이블에 넣는다.

논리적 데이터 모델링

◆ 3차 정규화

- 한 엔티티 안의 모든 주식별자(PK)가 아닌 속성들은 주식별자(PK)에 의존해야 한다.

테이블 명 : 주문

컬럼명	주문ID	상품ID	회원ID	회원명	전화번호	회원등급	수량	단가
키형태	PK							

위 주문 테이블에서 키가 아닌 모든 컬럼은 기본키인 주문ID에 종속적인가?

- 회원명, 전화번호, 회원등급은 회원ID에 종속적이다. 위와 같이 설계되면 회원이 주문할 때마다 동일한 회원에 대한 회원명, 전화번호, 회원등급이 데이터가 중복되게 된다. 또한, 회원의 정보가 수정되면 주문테이블에서도 회원의 정보를 수정해야 한다. 제 3정규화는 기본키에 의존하지 않고, 키가 아닌 일반 컬럼에 의존하는 컬럼을 제거 시켜 원래의 제자리로 위치 시키는 것이다.

논리적 데이터 모델링

◆ 3차 정규화

예)

테이블 명 : 사원

컬럼명	사원번호	이름	주민번호	전화번호	핸드폰번호	우편번호	주소	입사일
키형태	PK							

만약, 주소를 시, 구, 읍면동, 나머지 주소 필드로 구성했다면 이 모든 속성이 사원번호에 의존적이라고 할 수 있나?

- 시, 구, 읍면동은 우편번호에 의존적이다.
- 그러므로 시, 구, 읍면동은 테이블에서 제거해야 한다.
- 우편번호 테이블을 만들어 우편번호(pk), 시, 구, 읍면동, 나머지 주소등을 일반속성으로 만든다.
- 사원테이블에는 우편번호(fk), 나머지 주소등 속성 필요.

논리적 데이터 모델링

◆ 정규화 정리

-정규화란 데이터의 중복을 최소화 하고 데이터 무결성을 극대화 하기 위한 것이다.

- 정규화를 하는 이유는 중복을 방지, 효율적 으로 데이터 저장 ,속성들을 본래의 테이블에 정확하게 위치하게 위함.
- 가장 작은 논리적인 단위로 구분하는 것을 관계형데이터베이스 이다.
- 보다 더 작은 여러 개의 테이블로 나누어 지게 된다.
- 부모와 자식들간 관계가 만들어지고 서로 조인 해서 검색한다.
- 데이터를 수정할 때는 많은 시스템에 부담을 줄 수 있다

그러므로 오버헤드 때문에 시스템이 안정적이지 않다면 정규화는 의미가 없을 수 있으므로 정규화를 안 하는 경우도 있다. 그래서 물리적 데이터베이스에서는 역정규화를 통해 정규화를 없앤다.

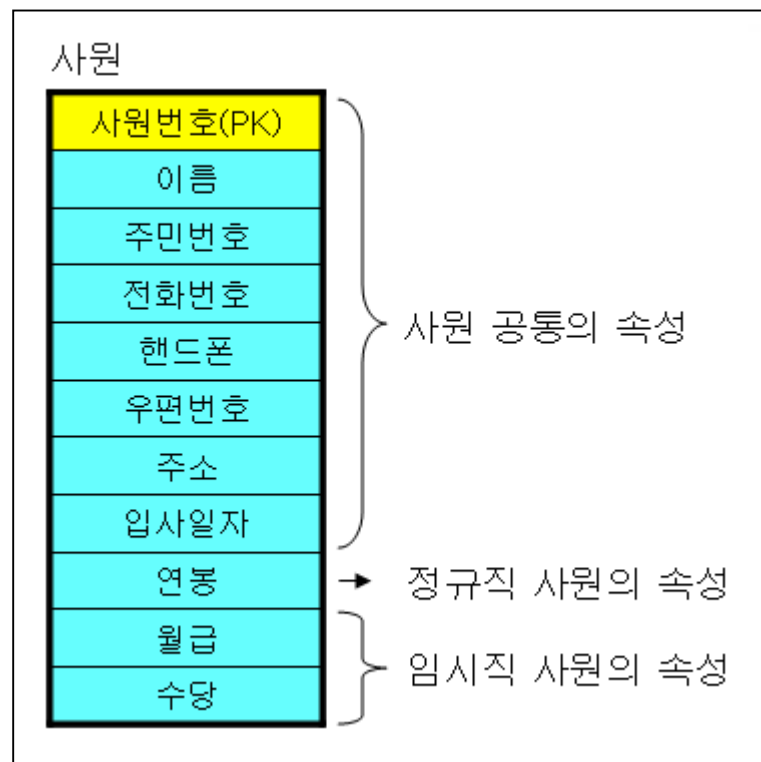
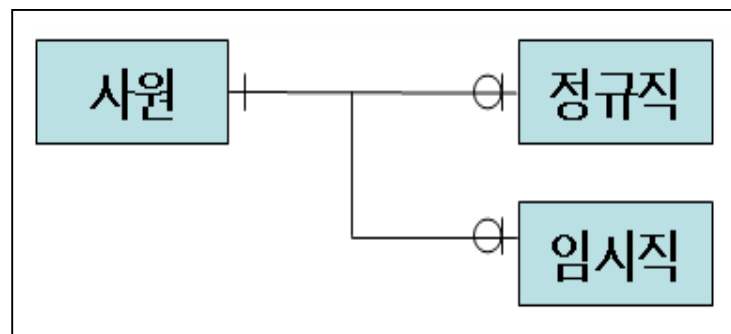
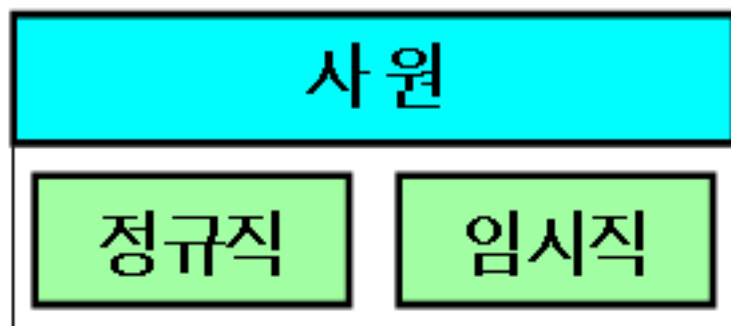
- 정규화는 1 - 2 -3 차순 으로 가는 것이 아니라 정규화 방식에 따라 정규화 1 ,2 , 3을 적용하는 것이다.
- 1차정규화, 3차정규화 의미가 없으면서 2차 정규화만 할 수 도 있다.

논리적 데이터 모델링

4. 특수한 경우의 엔티티 모델링

4.1 수퍼 타입과 서브 타입

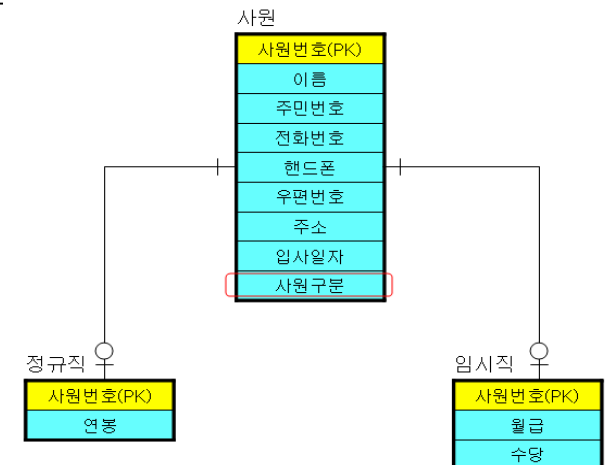
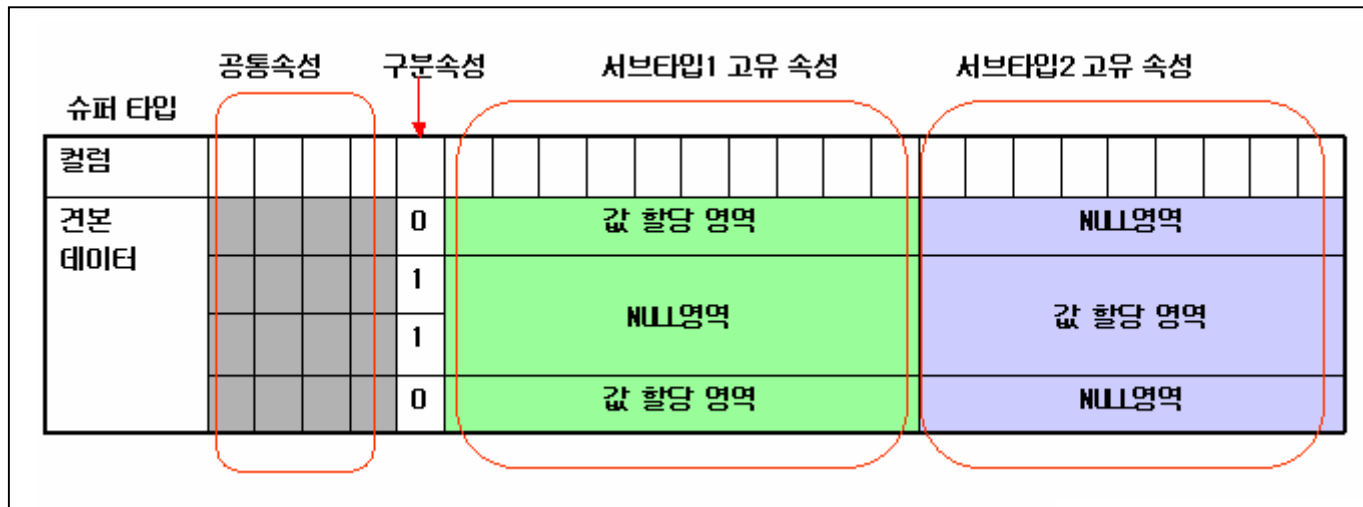
: 하나의 실체는 두 개 이상의 실체로 분할될 수 있으며, 이들의 공통 분모를 모아서 수퍼 타입으로 정의하고 나머지 배타적인 속성들을 모아서 서브타입으로 정의한다.



논리적 데이터 모델링

4. 특수한 경우의 엔티티 모델링

4.1 수퍼 타입과 서브 타입



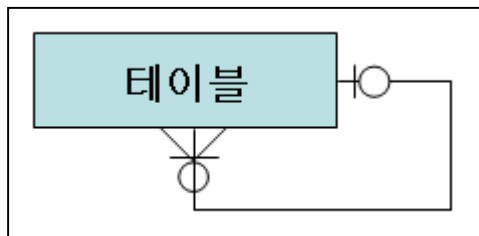
논리적 데이터 모델링

4. 특수한 경우의 엔티티 모델링

4.2 재귀적 관계

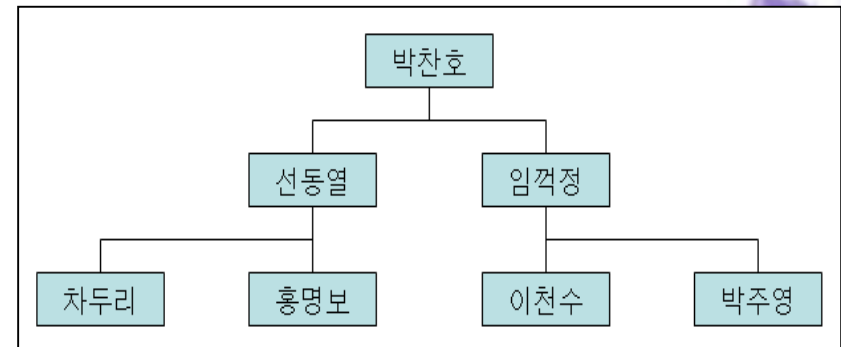
: 자기 자신과 관계를 맺음으로 해서 자기 자신의 기본키가 자기 자신에게 포린키로 전이되는 관계이다.

* 계층적인 구조를 정의할 때 재귀적 관계를 사용한다.



직원

직원번호(PK)	이름	주민번호	관리사번(FK)
1	박찬호	690001-1234567	NULL
2	선동열	700001-1234567	1
3	임꺽정	650001-1234567	1
4	차두리	710001-1234567	2
5	홍명보	760001-1234567	2
6	이천수	700001-1234567	3
7	박주영	730001-1234567	3



논리적 데이터 모델링

4. 특수한 경우의 엔티티 모델링

4.2 재귀적 관계

예 테이블명 : 코드분류

컬럼명	코드	코드명	코드구분
키 형태	FK		FK
기본 데이터	01	컴퓨터	NULL
	02	경영/경제	NULL
	NULL
	0101	운영체제	01
	0102	프로그래밍	01
	0103	데이터베이스	01

	010101	Windows	0101
	010201	Visual Basic	0102
	010301	SQL Server	0103

최상위분류

01 컴퓨터
02 경영 경제

2단계분류

0101 컴퓨터의 운영체제
0102 컴퓨터의 프로그래밍

3단계분류

010101 컴퓨터의 운영체제 windows
010102 컴퓨터의 운영체제 sqlserver

하위단계 분류

논리적 데이터 모델링

4. 특수한 경우의 엔티티 모델링

4.2 재귀적 관계

예) 계층형 게시판

게시판

컬럼 명	번호	제목	내용	작성자	E-mail	상위레코드번호
키 형태	PK					

번호 (PK)	제목	내용	작성자	이메일	상위레코 드번호(FK)
1					Null
2					1
3					1
4					null

3. 물리적 데이터베이스 모델링

물리적 데이터 모델링

1. 물리적 데이터베이스 모델링에 대한 이해

1.1 물리적 데이터 모델링

논리적 데이터 모델링	물리적 데이터 모델링
특정 DBMS 제품을 고려하지 않고 관계형 데이터베이스 이론에 맞게 설계	특정 DBMS 제품의 특성에 맞는 최적화된 데이터베이스 설계

1.2 물리적 데이터 모델링 작업 단계

- 1) 개발 DBMS를 선정하고 컬럼의 데이터 타입과 사이즈를 정의한다.
- 2) 데이터 사용량 분석과 사용자들이 데이터 소스에 액세스 할 때의 구체적인 프로세스 분석.
- 3) 역정규화 수행
- 4) 데이터베이스 내의 개체들 정의(인덱스, 뷰, 저장프로시저...)
- 5) 물리적인 데이터베이스 생성

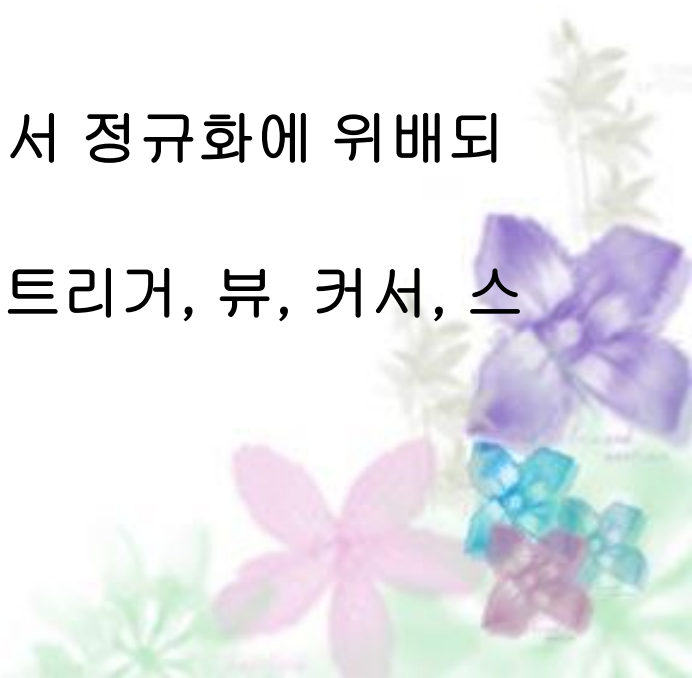
물리적 데이터 모델링

2. 역정규화(De-Normalization) = 반정규화

- 역정규화는 정규화에 위배되는 것.
- 논리적 설계에서 테이블이 가장 작은 논리적 단위로 분할 되어진다. 테이블이 나뉘지면 부모와 자식에 있어 참조하는 상황이 발생. 검색시 조인이 필요하고 이러한 경우 insert ,delete , update 할 때 시스템의 부하가 많이 발생한다.
그러므로 정규화된 스키마는 시스템의 부하를 유발한다.

역정규화란 시스템의 퍼포먼스 향상을 위해서 정규화에 위배되어지는 행위를 말한다.

역정규화 이후에 인덱스설정, 테이블 분할, 트리거, 뷰, 커서, 스토어프로시저 이러한 것을 생각한다.



물리적 데이터 모델링

2. 역정규화(De-Normalization)

2.1 역정규화 고려사항

- 정확한 업무이해 필수.
- 업무에 따른 데이터 사용량(입력, 수정, 삭제, 조회)과 업무 프로세스 분석.
- 데이터 사용량 분석 - 사용량이 많은 테이블을 기준으로 우선적으로 역정규화 고려



물리적 데이터 모델링

2. 역정규화(De-Normalization)

2.2 역정규화 의 유형

- 데이터 중복(컬럼의 역정규화)
- 파생 컬럼의 생성
- 테이블 분리
- 요약테이블 생성
- 테이블 통합



물리적 데이터 모델링

2. 역정규화(De-Normalization)

2.2.1 컬럼 역정규화

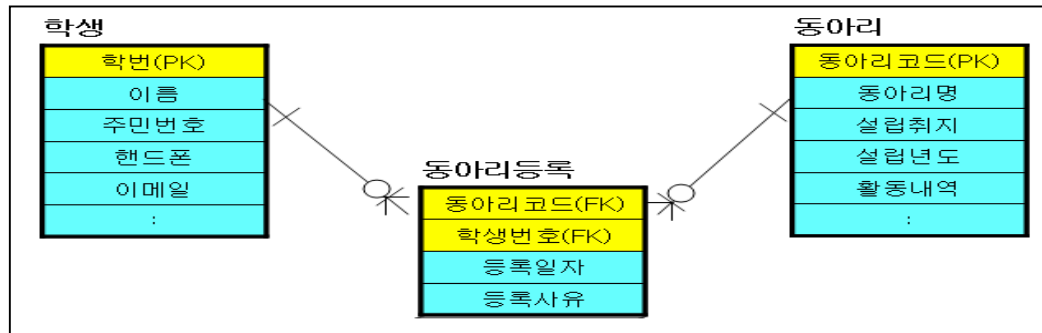
- 조인 프로세스를 줄이기 위해 조인을 해서 다른 테이블에 있는 컬럼의 데이터를 조회하는 경우 **아예 해당 컬럼을 중복 함으로 해서 기본적으로 조회 할 경우 조인을 수행하지 않도록 하기 위한 방법이다.**
- 조회가 많이 이루어지는 테이블의 경우 데이터 중복을 통해 성능 향상을 기대 할 수 있으며 교차 실체(행위실체)의 경우 데이터 중복 역정규화의 대상이 된다.



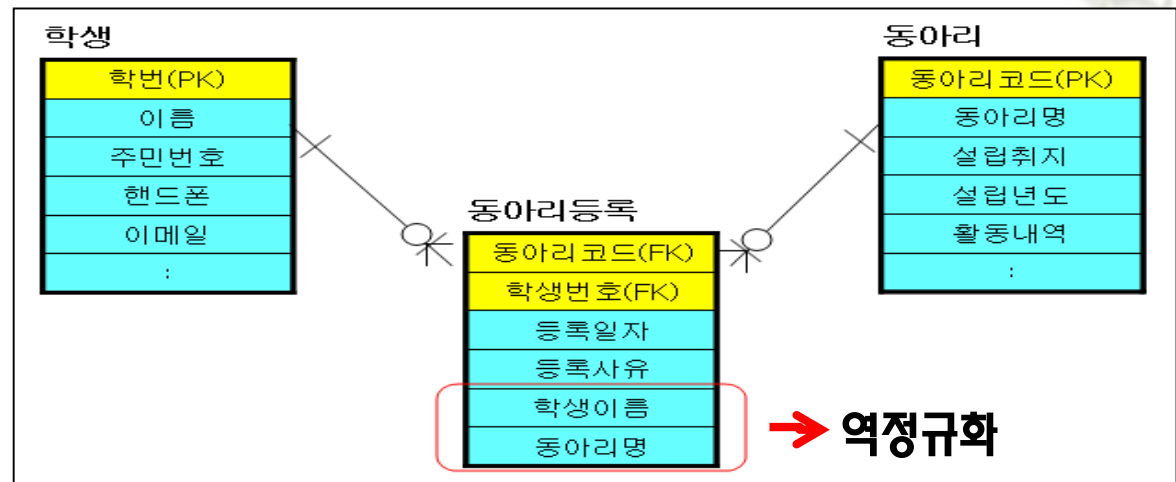
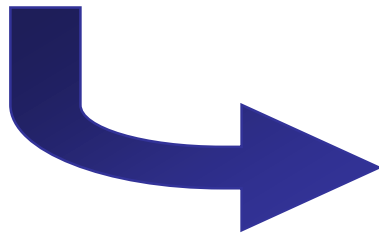
물리적 데이터 모델링

2. 역정규화(De-Normalization)

2.2.1 컬럼 역정규화



예) 동아리등록 검색 시 학생이름과 동아리명을 함께 검색하기 위해 조인을 수행 해야하는데 이러한 조회의 건수가 많다고 가정했을 때 컬럼의 역정규화!



물리적 데이터 모델링

2. 역정규화(De-Normalization)

2.2.2 파생 컬럼의 생성

- 계산을 통해서 얻어질 수 있는 결과 값을 테이블의 컬럼으로 만들어서 값으로 저장하게 하면 조회 할 때 마다 연산을 통해 결과값을 얻지 않아도 되기 때문에 조회의 성능을 향상 시킬 수 있다.

예) 판매테이블의 판매금액

$$\text{판매금액} = \text{판매수량} + \text{판매단가}$$

예) 성적 테이블의 총점과 평균

* 미리 만들어두면 조회 할때 연산하지 않아도 되기에 효과적이다.

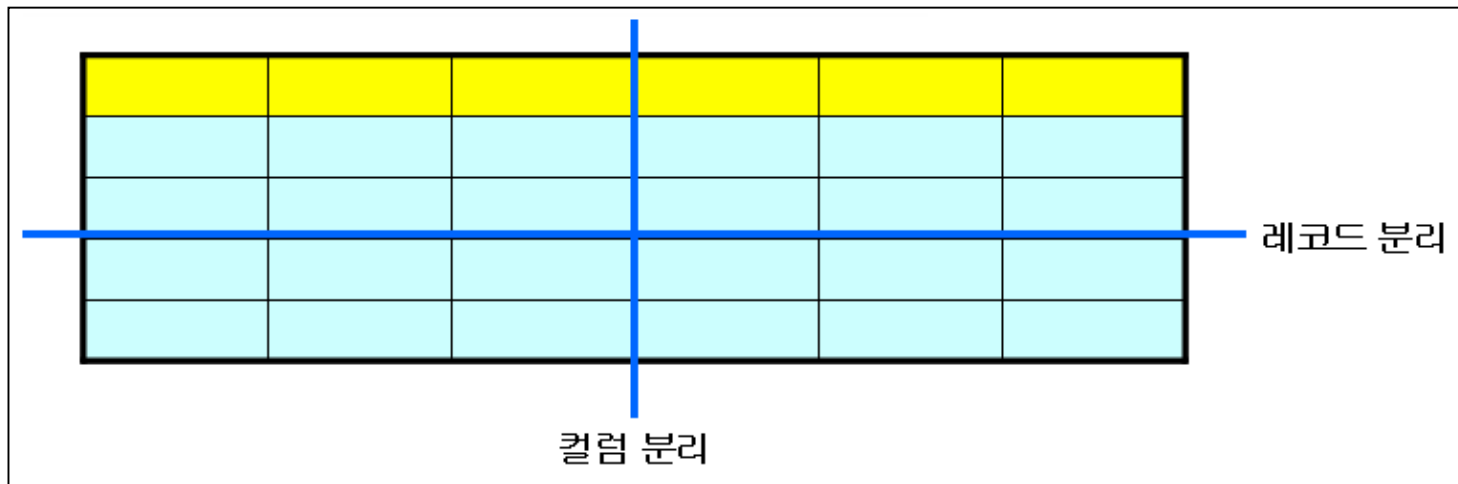


물리적 데이터 모델링

2. 역정규화(De-Normalization)

2.2.3 테이블 분리

- 레코드를 기준으로 분리
- 컬럼을 기준으로 분리



물리적 데이터 모델링

2. 역정규화(De-Normalization)

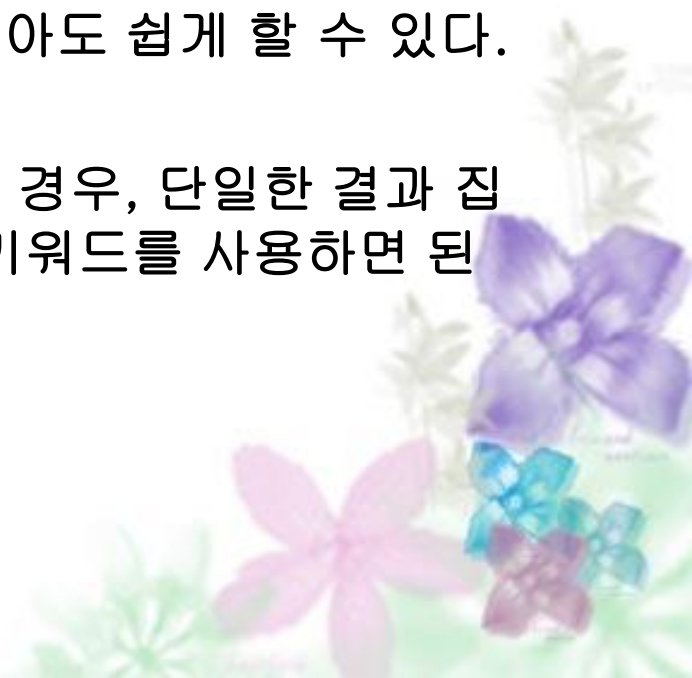
2.2.3 테이블 분리

1) 레코드 분리

- 만약, 성적테이블에서 공부 잘하는 학생, 중간학생, 못하는 학생을 기준으로 조회가 많이 이루어진다고 한다면 총점, 평균 학점 필드를 기준으로 검색이 많아진다.

이러한 경우, 성적테이블을 그 기준으로 성적상, 중, 하 테이블로 분리한다. 이렇게 하면 where 조건을 걸지 않아도 쉽게 할 수 있다.

- 이렇게 레코드를 기준으로 데이터를 관리 할 경우, 단일한 결과 집합을 구하기 위해서는 Union, Union All이란 키워드를 사용하면 된다.



물리적 데이터 모델링

2. 역정규화(De-Normalization)

2.2.3 테이블 분리

2) 컬럼 분리

예) 사원테이블 - 사원(업무적으로 자주 활용하는 것)과 부가정보(업무적으로 자주 사용 안 하는 것)로 분리

한 테이블에 컬럼이 너무 많음, 정규화에 위배되지는 않지만 컬럼의 개수는 10개내 에로 정의하는 것이 좋다 이유는 하나의 레코드사이즈를 줄여서 한 페이지에 저장 될 수 있는 레코드를 더 많이 저장할 수 있는 것이 성능에 좋다.

사원과 부가정보는 1 : 1 관계이다.



물리적 데이터 모델링

2. 역정규화(De-Normalization)

2.2.3 테이블 분리

2) 컬럼분리 예)

사원

사원번호(PK)
사원ID
비밀번호
이름
주민번호
집전화번호
핸드폰번호
사내전화번호
이메일
우편번호
주소
생년월일
취미
특기
입사일자
시력좌
시력우
키
몸무게
혈액형



사원

사원번호(PK)
사원ID
비밀번호
이름
주민번호
핸드폰번호
사내전화번호
이메일
입사일자

부가정보

사원번호(PK)
집전화번호
우편번호
주소
생년월일
취미
특기
시력좌
시력우
키
몸무게
혈액형



물리적 데이터 모델링

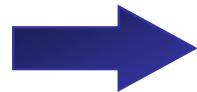
2. 역정규화(De-Normalization)

2.2.3 테이블 분리

2) 컬럼분리 예)

게시판

번호(PK)
리플번호
순서
제목
내용
글쓴이
작성일시
조회수
추천수
이메일
IP주소
비밀번호
첨부파일
다운로드수
총리플수



게시판

번호(PK)
리플번호
순서
제목
글쓴이
작성일시
조회수
추천수
총리플수

게시내용

번호(PK)
내용
이메일
IP주소
비밀번호
첨부파일
다운로드수



물리적 데이터 모델링

2. 역정규화(De-Normalization)

2.2.4 요약테이블 생성

- 판매테이블과 같이 비교적 많은 데이터가 저장되는 테이블에서 요약된 정보를 얻기 위하여 Group By 와 Sum등을 이용하여 가공된 결과를 얻는 질의를 자주 실행하게 되는 경우 해당 질의를 수행하기 위해서 많은 프로세스가 발생하게 되는데 이러한 조회의 프로세스를 줄이기 위해 요약된 정보만을 저장하는 테이블을 만든 것.



물리적 데이터 모델링

2. 역정규화(De-Normalization)

2.2.4 요약테이블 생성

판매

판매번호(PK)
회원번호
상품코드
매장번호
판매일자
판매수량
판매단가
판매금액



판매통계

판매일자(PK)
매장번호(PK)
상품코드(PK)
일일판매수량
일일판매금액

위 판매 테이블의 조회의 유형은 일일 판매내역, 일일 매출액을 판매일자별로 조회하고 또한 주별/월별/분기별/연도별 매출액도 위 판매 테이블을 근거로 조회한다고 했을 때 평균 하루에 판매되는 건수가 10,000건 이상이라면 질의를 수행하는 데 있어 시스템 부하가 클 수 밖에 없다. 이러한 경우 요약테이블을 만들어서 관리 한다면 좀더 적은 양의 데이터를 조회해서 원하는 결과를 얻을 수 있음.

물리적 데이터 모델링

2. 역정규화(De-Normalization)

2.2.5 테이블 통합

- 테이블을 나누고 보니 데이터를 입력, 수정, 삭제 할 때 마다 부모테이블과 자식테이블의 데이터를 참조해야 하고 또한 연관된 정보를 보기 위해 조인을 걸어서 조회를 해야 만 한다.

그래서 이러한 프로세스가 시스템의 성능에 많은 영향을 끼친다면 분할된 테이블을 원래의 상태로 합치는 것도 성능향상을 위해 필요함.



물리적 데이터 모델링

2. 역정규화(De-Normalization)

2.2.5 테이블 통합 예)

개인신상

일련번호(PK)	이름	주민번호	연락처
1	박찬호	690001-1234567	011-0001-0001
2	선동열	700001-1234567	017-0001-0002
3	임꺽정	650001-1234567	018-0001-0003
4	차두리	710001-1234567	019-0001-0004
5	홍명보	760001-1234567	010-0001-0005
6	이천수	700001-1234567	010-0002-0006
7	박주영	730001-1234567	010-0003-0007
8	이승엽	660001-1234567	010-0004-0008

자동차

일련번호(FK)	차종	차명	년식	구입일
5	SUV	산타페	2006년	20060708
2	승용차	에쿠스	2005년	20050708
1	승용차	소나타	2003년	20030305
3	트럭	타이탄	2002년	20020507

고객관리

일련번호	이름	주민번호	연락처	차종	차명	년식	구입일
1	박찬호	690001-1234567	011-0001-0001	승용차	소나타	2003년	20030305
2	선동열	700001-1234567	017-0001-0002	승용차	에쿠스	2005년	20050708
3	임꺽정	650001-1234567	018-0001-0003	트럭	타이탄	2002년	20020507
4	차두리	710001-1234567	019-0001-0004	NULL	NULL	NULL	NULL
5	홍명보	760001-1234567	010-0001-0005	SUV	산타페	2006년	20060708
6	이천수	700001-1234567	010-0002-0006	NULL	NULL	NULL	NULL
7	박주영	730001-1234567	010-0003-0007	NULL	NULL	NULL	NULL
8	이승엽	660001-1234567	010-0004-0008	NULL	NULL	NULL	NULL

물리적 데이터 모델링

역정규화(De-Normalization) 정리

- 시스템의 성능향상을 위해서 긍정적으로 도움이 될 수 있는 요인이 된다면 역정규화의 유형이 된다. 즉, 합당한 이유가 된다.
- 정규화를 해놓고 데이터 사용량을 분석한 다음에 성능향상을 위해 역정규화를 한다.



ER-Win 사용 방법

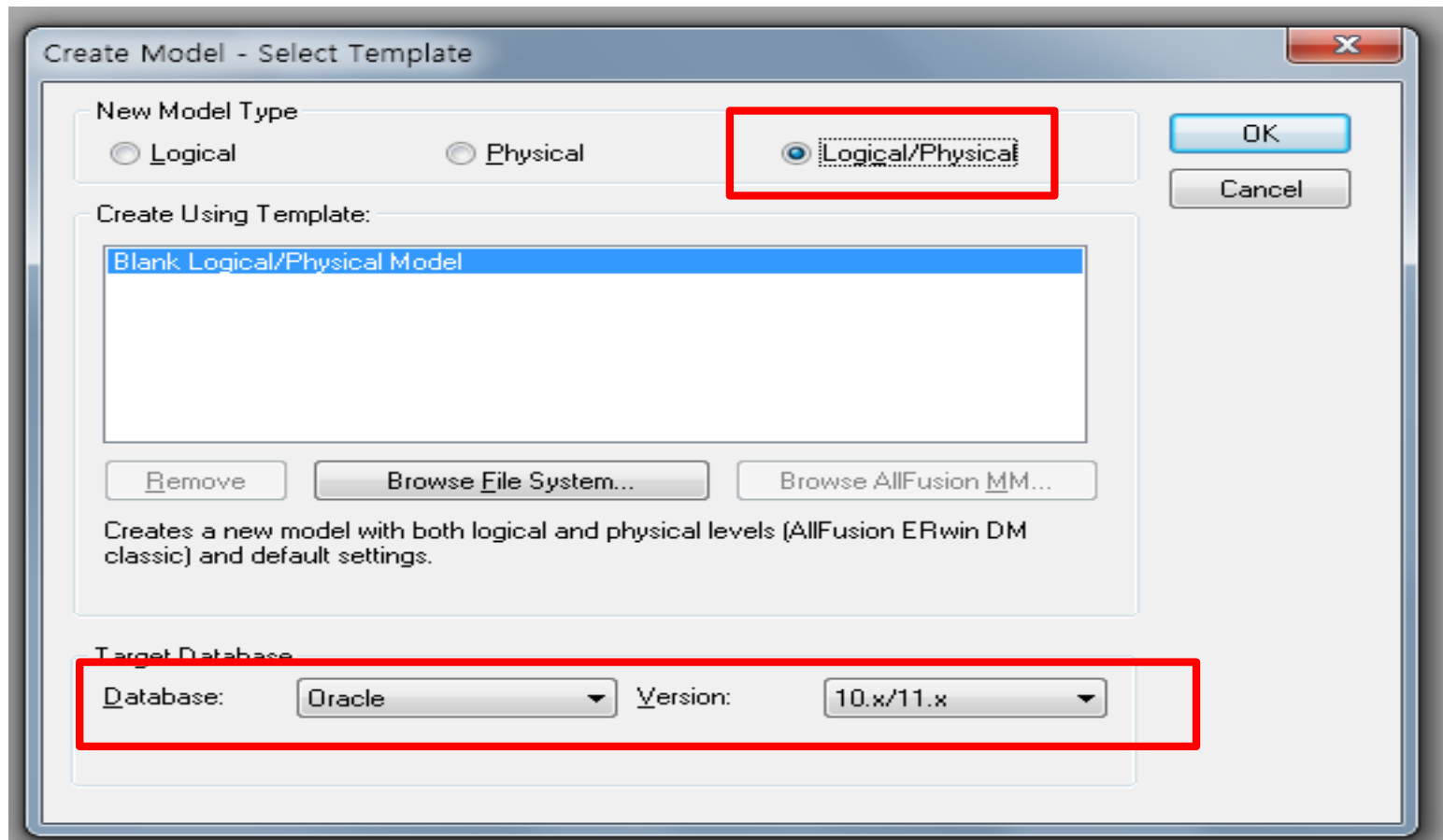
ER-Win

ER-Win을 데이터베이스 모델링 CASE(Computer Added Soft Engineering) Tool이라고 한다.

* ERwin.exe 을 설치한다.



ER-Win 실행



ER-Win 표기 방식

① IE(Information Engineering) 표기방식

- 정보공학 표기 방식으로 우리가 일반적으로 모델링을 할 때 가장 많이 사용하는 유형임

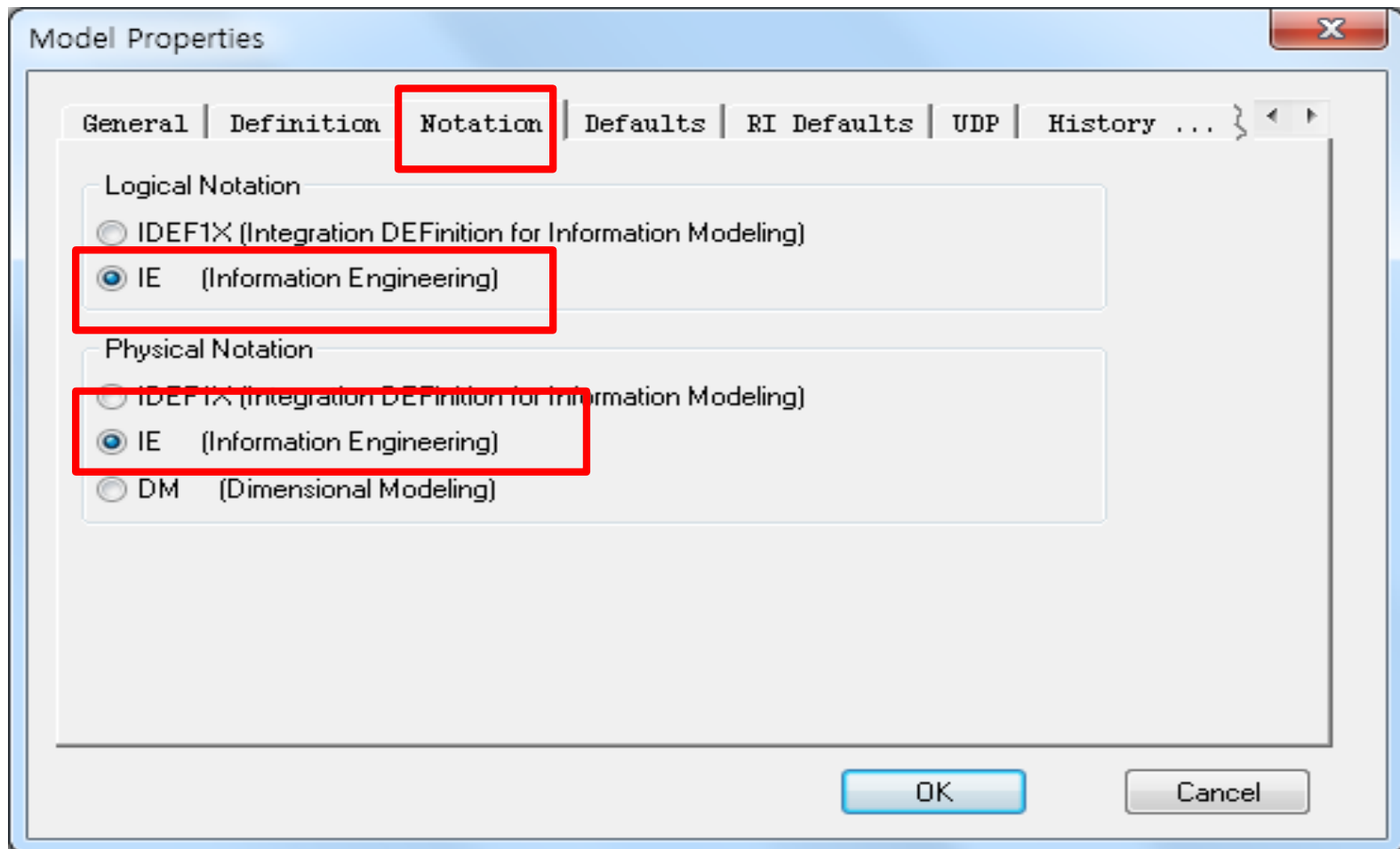
② Ideflx(Integration DEFinition for Information Modeling) 표기방식

- 미 국방성에서 프로젝트 표준안으로 개발한 표기 방식



ER-Win 표기 방식

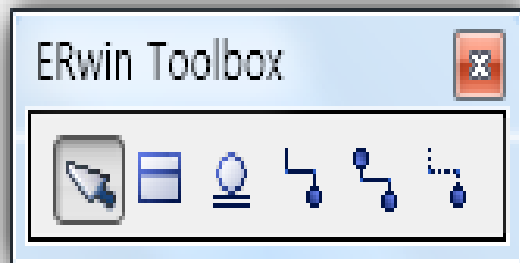
메뉴에서 Model -> Model Properties 선택



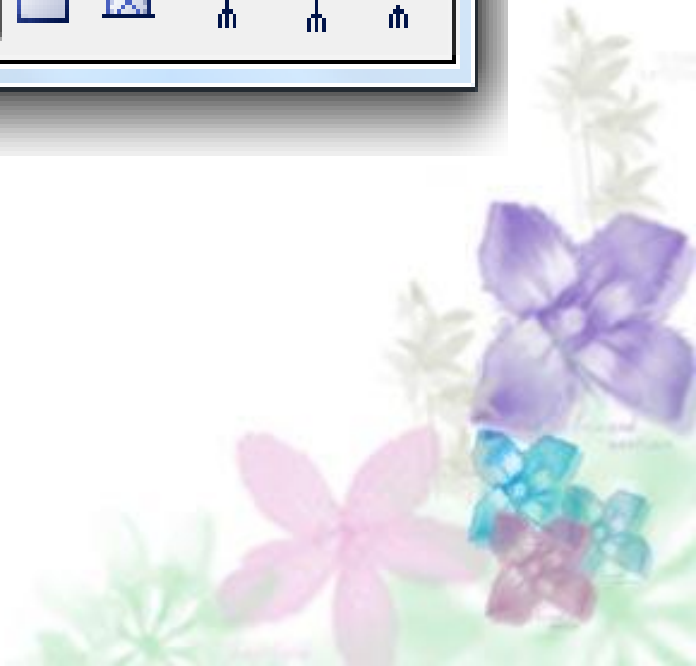
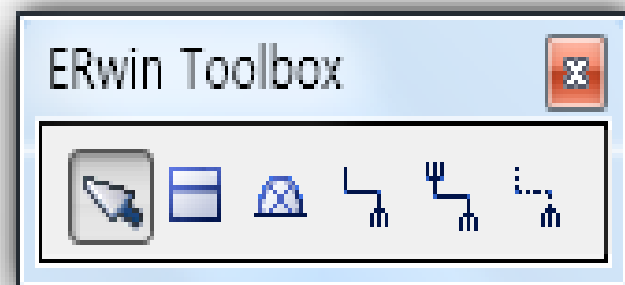
ER-Win 표기 방식

표기 방식을 변경하면

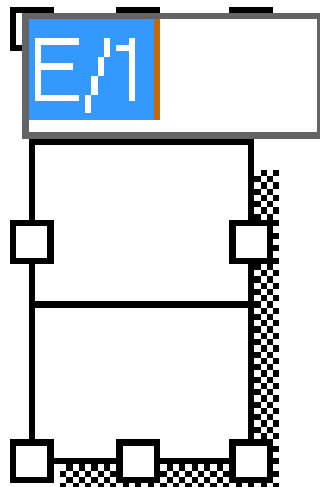
Ideflx 표기 방식



IE 표기 방식



엔티티(Entity) 생성



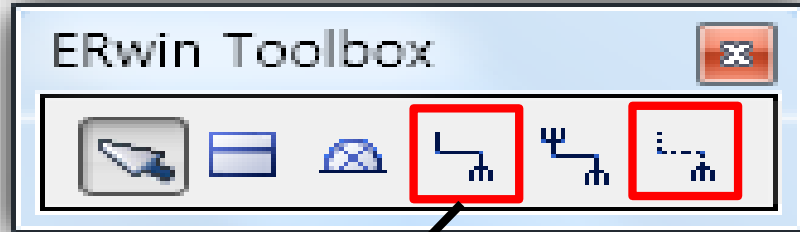
→ 엔티티 명

→ 기본 키 영역

→ 일반 속성 영역

세 영역을 이동하기 위해서 엔티티를 선택 한 후 탭 키를 이용하여 영역을 이동할 수 있으며 기본키와 일반 속성에 속성을 추가하고자 할 때 엔터를 친다.

식별관계 / 비식별관계



Identifying Relationship
(식별관계)

Non Identifying Relationship
(비 식별관계)

다 대 다 관계 정의

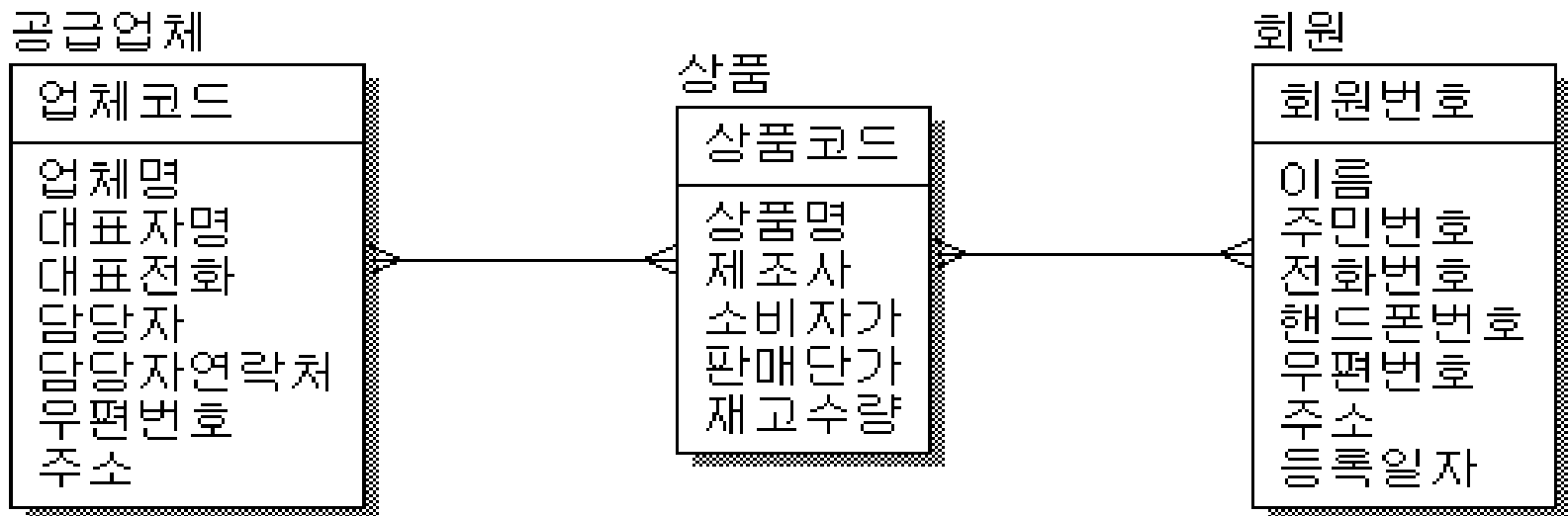
다 대 다 관계는 논리적으로 존재 할 수 있지만 물리적으로 존재 할 수 없다.



Many-to-Many Relationship
(다 대 다 관계)

다 대 다 관계 정의

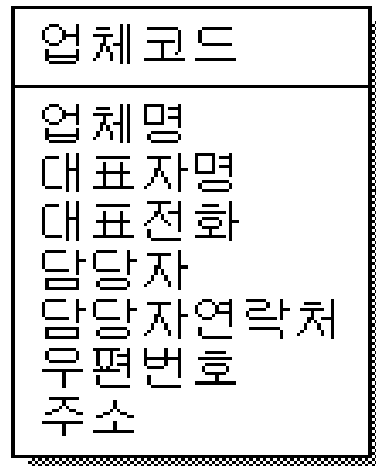
하나의 공급 업체는 여러 개의 상품을 납품 할 수 있고 하나의 상품은 여러 공급업체에서 납품을 받을 수 있기 때문에 **공급업체와 상품 엔티티간은 다 대 다 관계이며**, 하나의 상품은 여러 회원에게 판매 할 수 있고 한 명의 회원은 여러 개의 상품을 구매 할 수 있으므로 이 역시 다 대 다 관계이다.



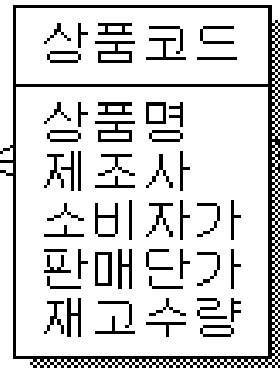
다 대 다 관계 정의

논리적 모델링에서도 다대다 관계를 풀어서 교차실체(행위실체)를 정의해야 한다.

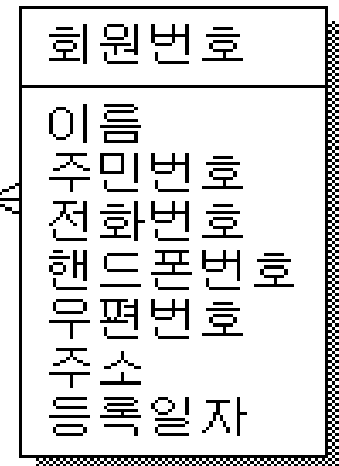
공급업체



상품



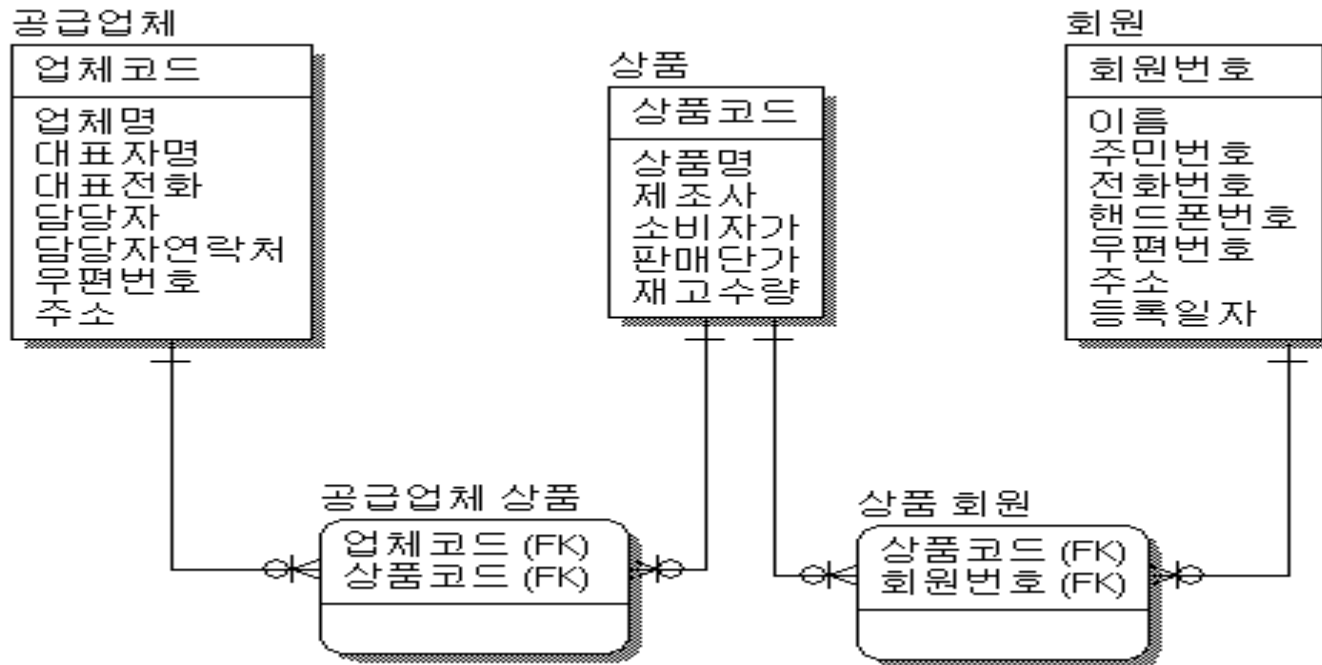
회원



ERwin Transform Toolbar



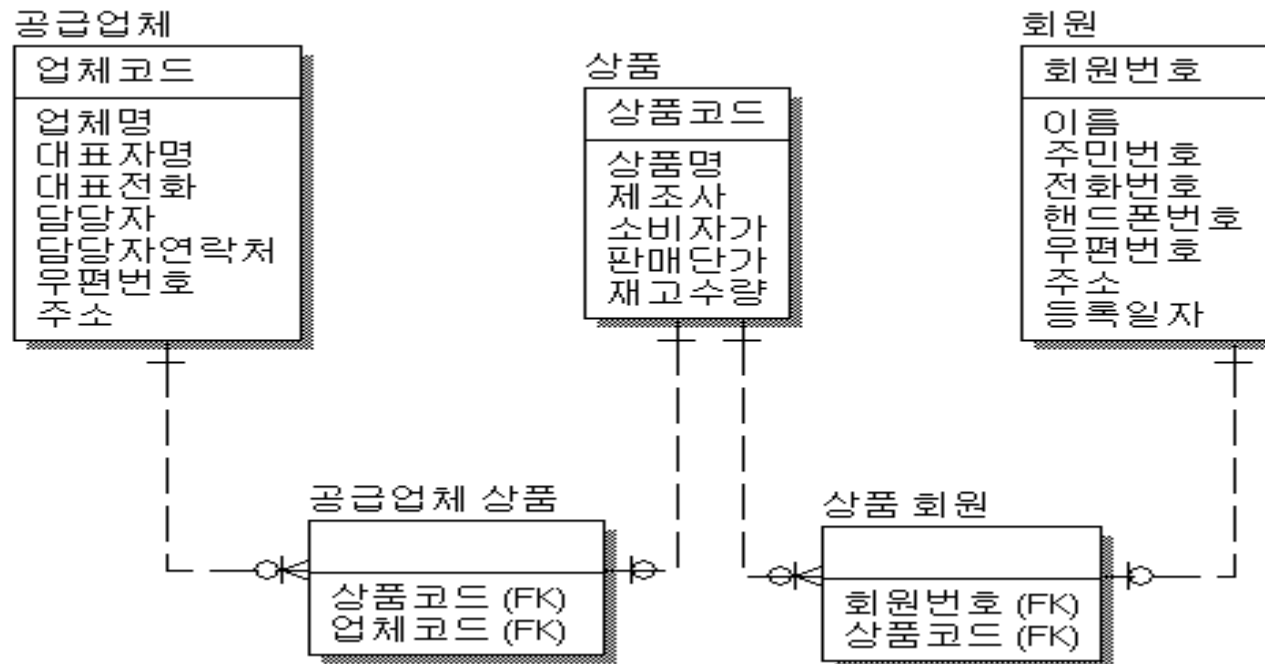
다 대 다 관계 정의



위 그림처럼 입고(공급업체상품) 엔티티와 판매(상품회원) 엔티티에서 정의된 것 처럼 두개의 컬럼을 이용해서 각각의 레코드를 식별 할 수 없다. 때문에 입고엔티티는 입고번호를, 판매 엔티티에는 판매번호를 기본키로 정의해서 사용한다(이를 대리키라고 한다)

다 대 다 관계 정의

우선, 모든 관계선을 비 식별자로 변경한다.



다 대 다 관계 정의

기본키와 필요한 일반속성을 추가 해 본다.

공급업체

업체 코드
업체명
대표자명
대표전화
담당자명
담당자연락처
우편번호
주소

상품

상품 코드
상품명
제조사
소비자가
판매단가
재고수량

회원

회원 번호
이름
주민번호
전화번호
핸드폰번호
주소
생년월일

공급업체 상품

입고 번호
업체 코드 (FK)
상품 코드 (FK)
입고일자
입고수량
입고단가
입고금액

상품 회원

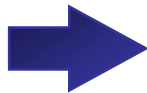
판매 번호
상품 코드 (FK)
회원 번호 (FK)
판매일자
판매수량
판매단가
판매금액

재귀적 관계 정의

재귀적 관계란 자기 자신과 관계를 맺는 것.
반드시 비 식별 관계로 정의해야 한다.

사원

사원번호
이름
주민번호
전화번호
핸드폰번호
우편번호
주소



사원

사원번호
이름
주민번호
전화번호
핸드폰번호
우편번호
주소

Relationships

Relationship: R/1 (사원 to 사원)

Name: R/1

New... Delete

General | Definition | **Rolename** | RI Actions | UDP

Rolename Info

Migrated Key: XPK.사원

Migrated Attribute: Rolename:

사원번호

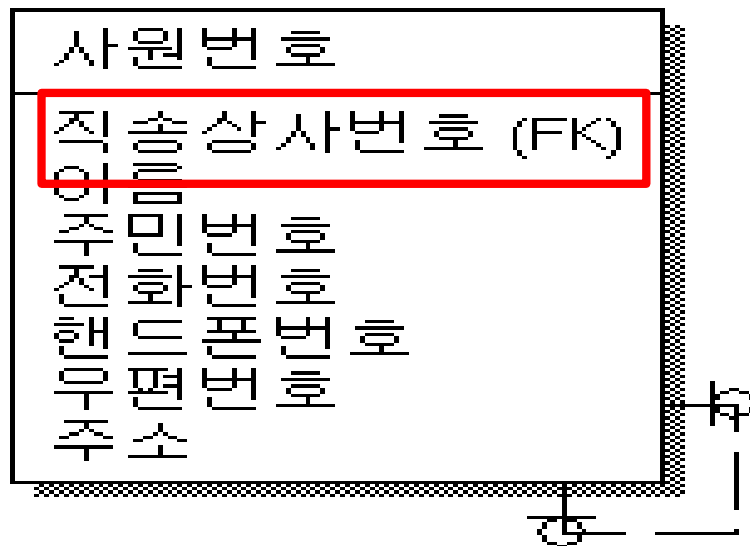
Rolename: 직송상사번호

Migrated Att: 사원번호

OK Cancel

재귀적 관계 정의

사원

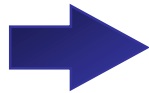


슈퍼타입과 서브타입

하나의 엔티티에서 배타적인 관계를 파악하고 이를 각각의 실체로 정의하고자 했을 때 각각의 실체에 공통되는 속성들을 모아 놓은 엔티티를 슈퍼타입으로 정의하고 개별적인 속성들을 따로 분리해서 각각의 서브타입으로 정의 할 수 있다.

사원

사번
이름
주민번호
전화번호
사원구분
연봉
월급수당



사원

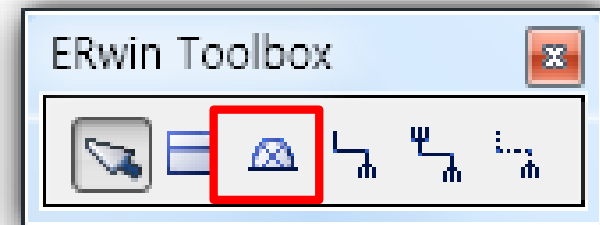
사번
이름
주민번호
전화번호
사원구분

정규직

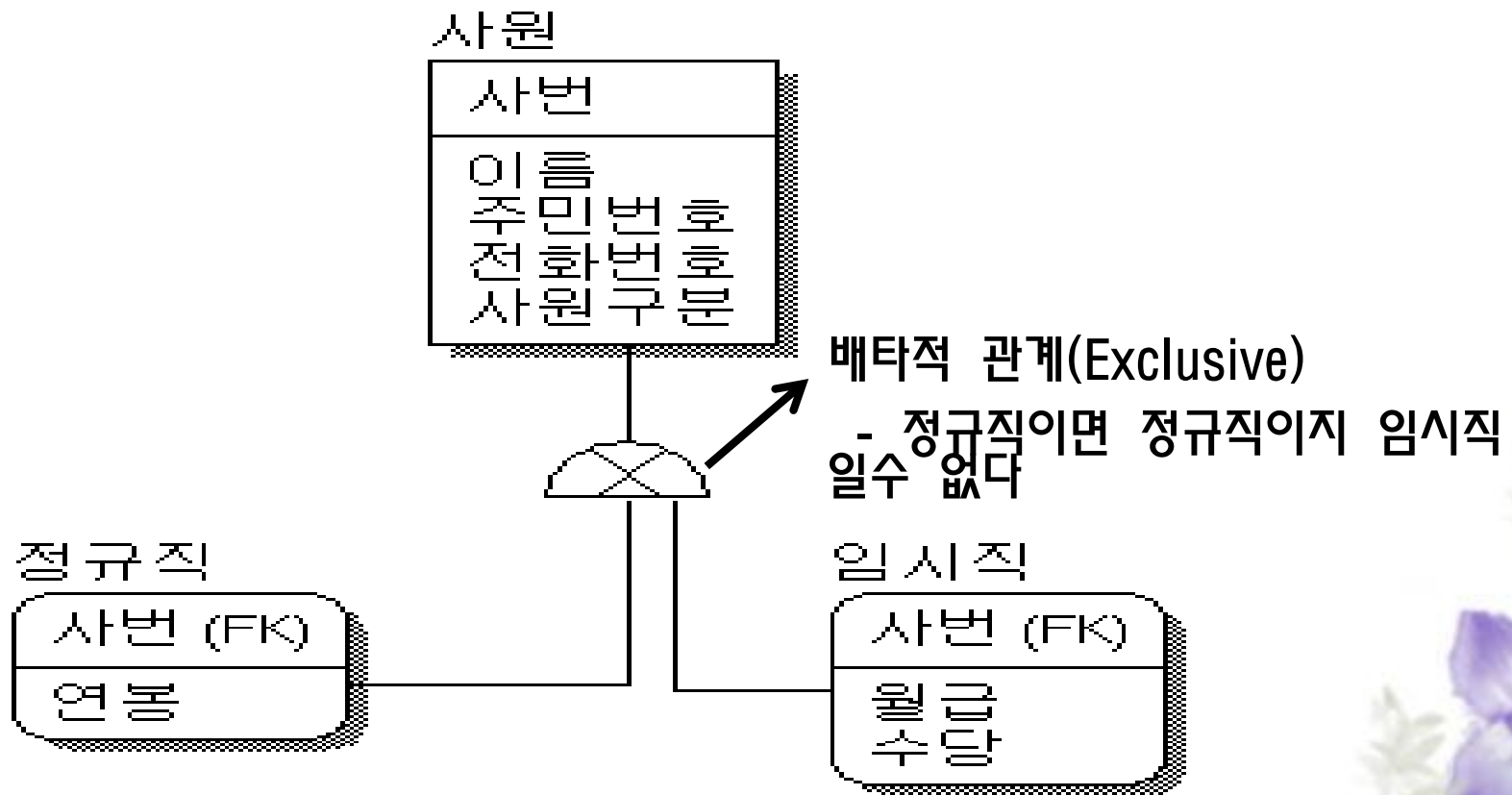
연봉

임시직

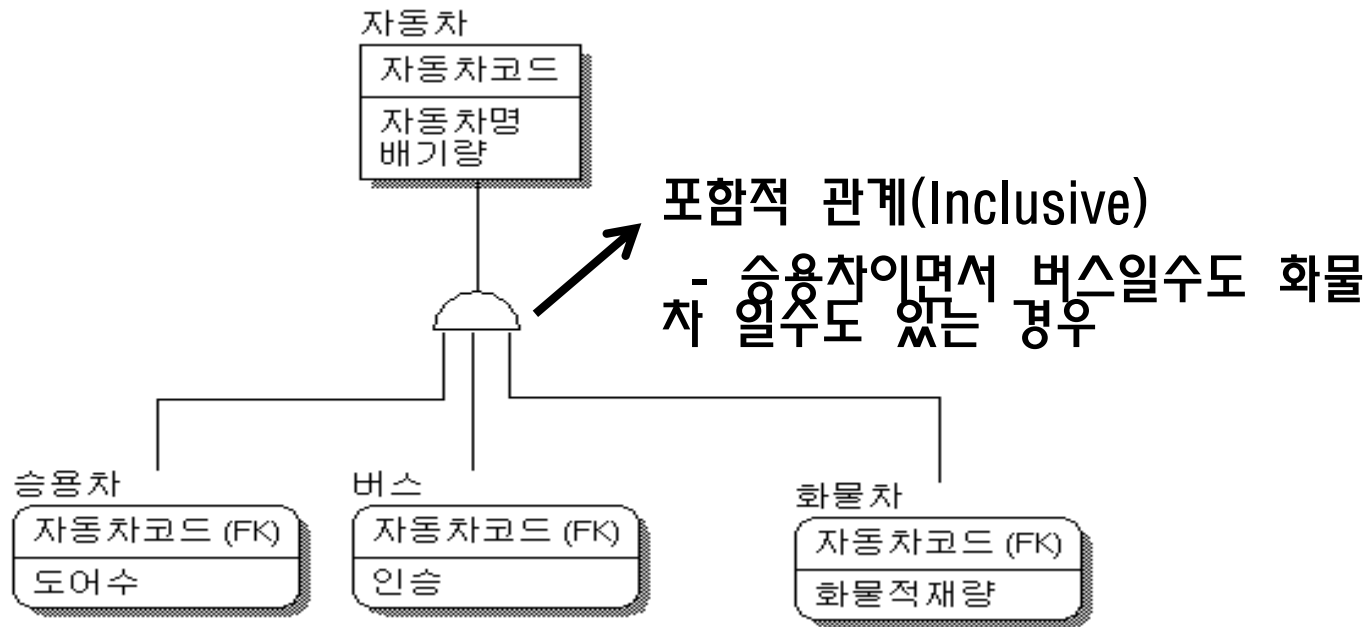
월급수당



슈퍼타입과 서브타입



슈퍼타입과 서브타입

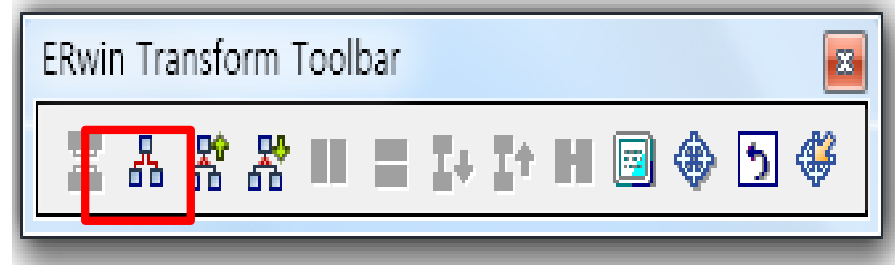
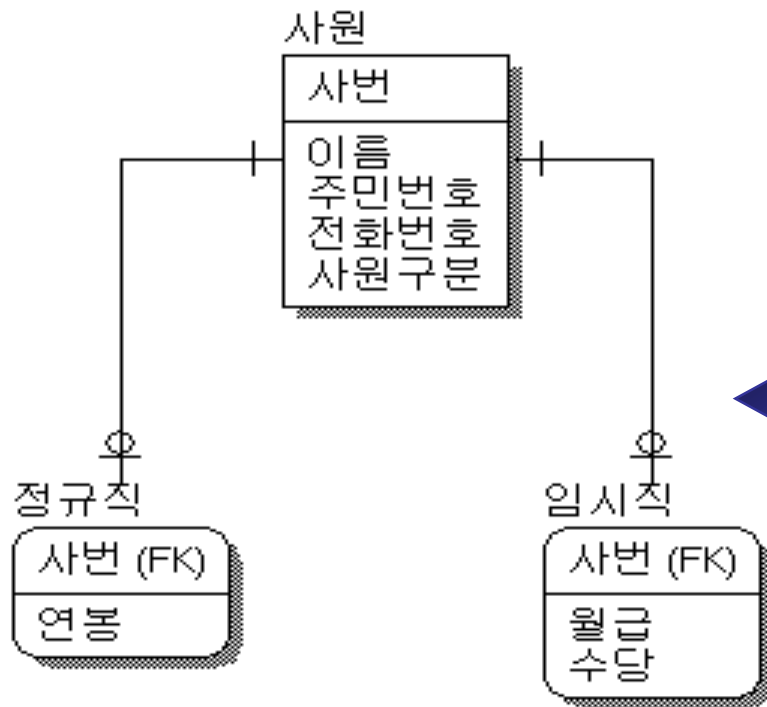


일반적으로 슈퍼타입과 서브타입의 관계는 Exclusive한 관계이다.

슈퍼타입과 서브타입

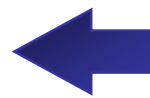
슈퍼타입과 서브타입과의 관계는 식별 관계인가 아니면 비 식별 관계인가?

- 슈퍼타입 엔티티의 기본키가 서브타입 엔티티의 기본키로 전이되었기에 식별관계이다 슈퍼타입과 서브타입의 관계를 식별관계로 전환해서 표현 할 수도 있다.



SuperType-Subtype Identity

식별관계로 전환



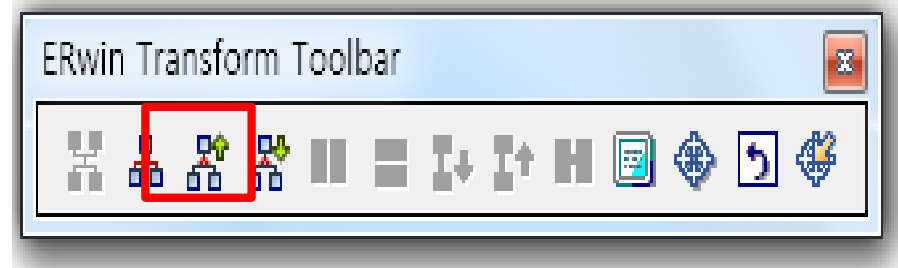
슈퍼타입과 서브타입

앞에서 처럼 테이블을 정의하면 결국 직원들의 정보가 세 개의 엔티티로 나뉘어 저장되기 때문에 프로그램 관리가 복잡해 질 수 있다. 이러한 경우

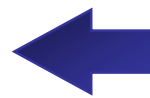
SuperType-Subtype Rollup 또는 SuperType-Subtype Rolldown 한다.

직원

사번
이름
주민번호
전화번호
사원구분
연봉
월급
수당



SuperType-Subtype Rollup



슈퍼타입으로 통합

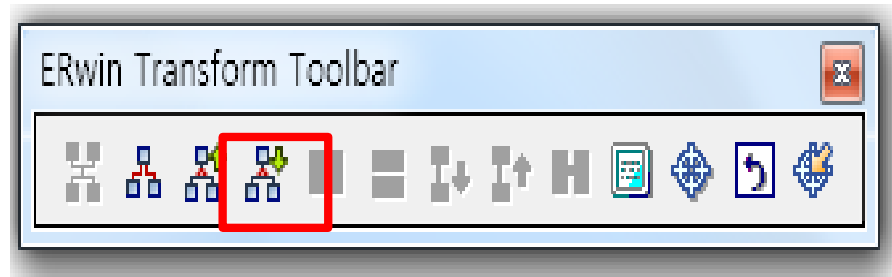
슈퍼타입과 서브타입

정규직

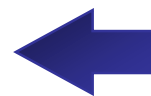
사번
연봉
이름
주민번호
전화번호
사원구분

임시직

사번
월급
수당
이름
주민번호
전화번호
사원구분



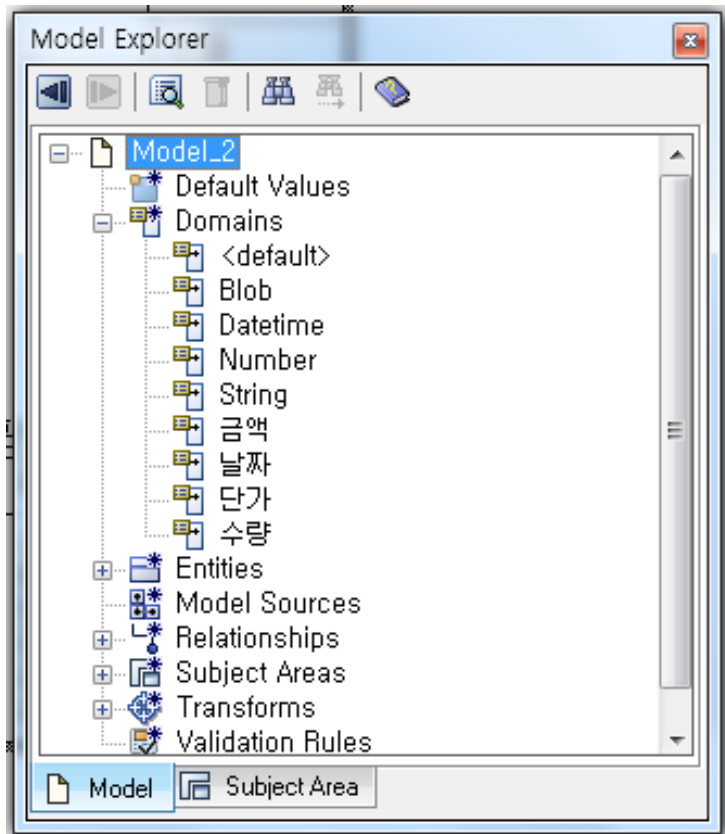
SuperType-Subtype Rolldown



서브타입으로 분리

도메인 설정

여러 엔티티에서 공통적으로 적용되는 속성이 존재하면 그것을 하나의 개체로 만들어 여러 엔티티에 적용시키는 것. – 사용자 정의 데이터타입

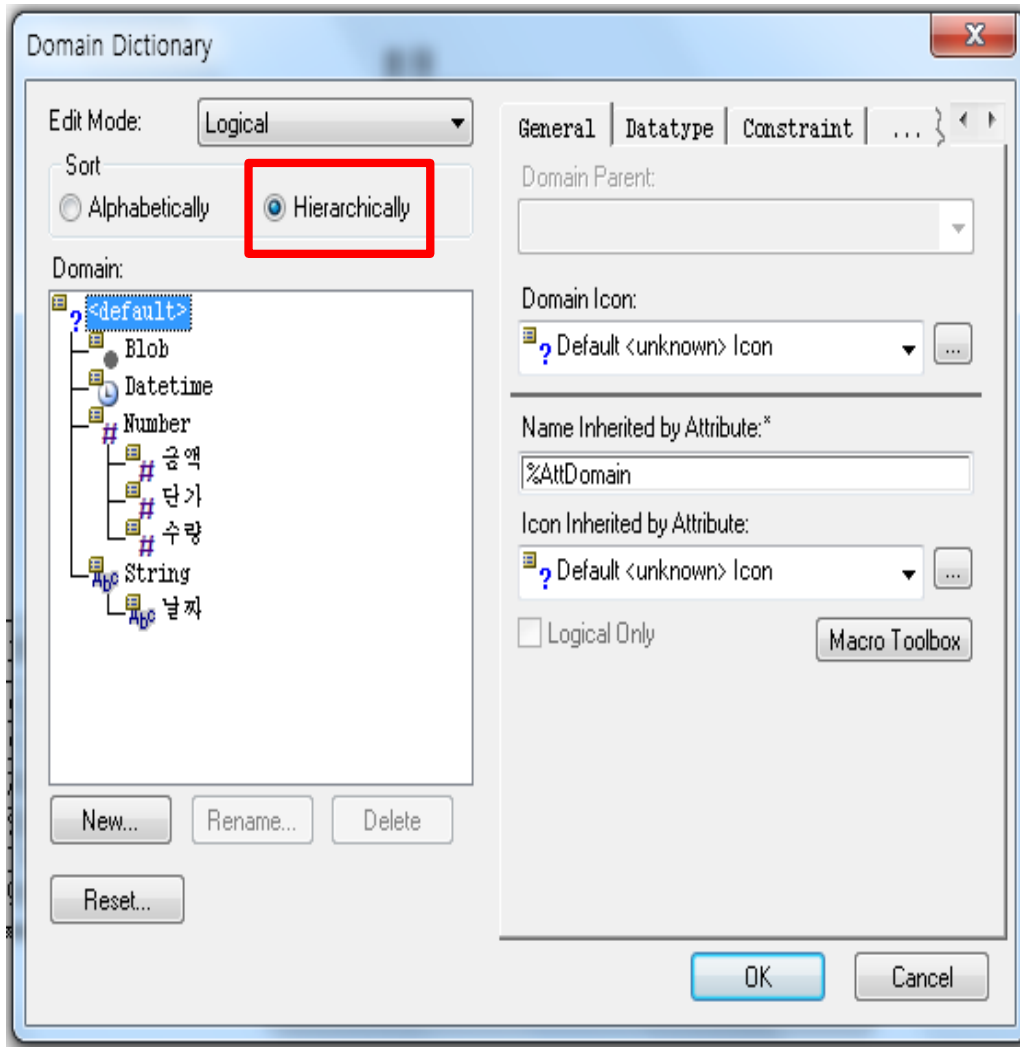


View -> Model Explorer ->

Domains -> 원하는Type선택 ->

오른쪽 버튼 New선택 ->속성명입력

도메인 설정



Domains ->
default 오른쪽 버튼 ->
properties 클릭

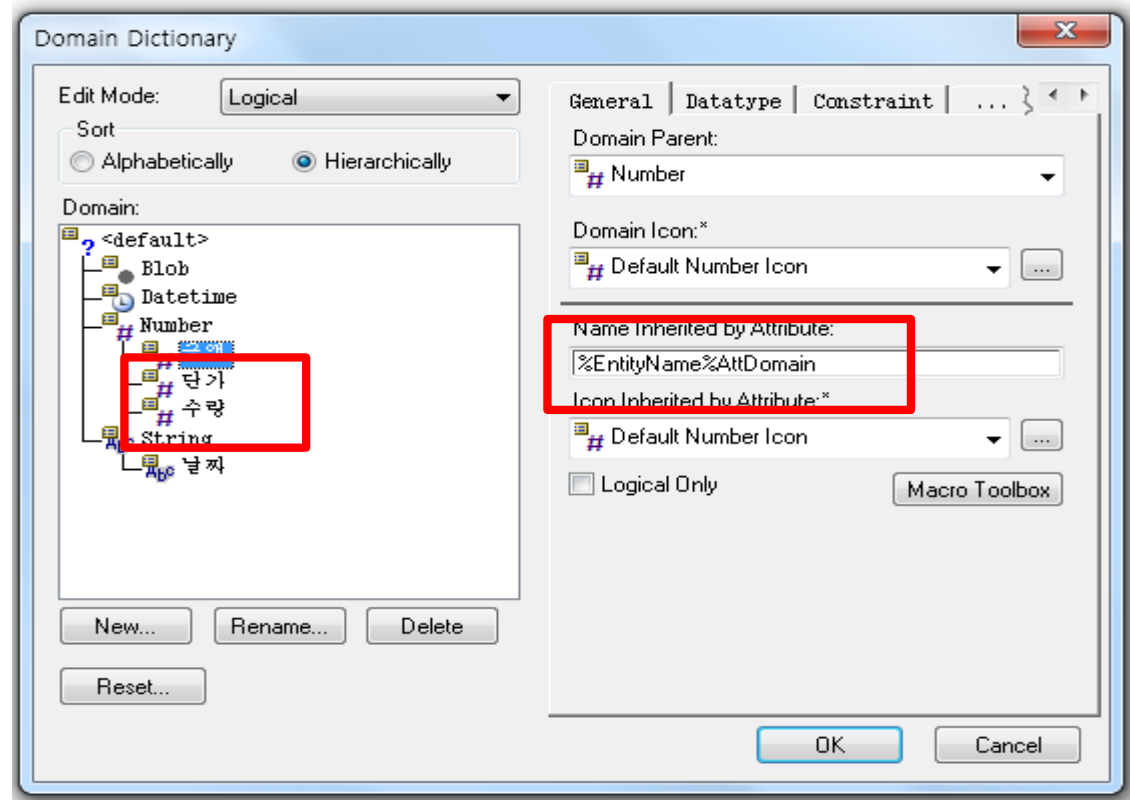
도메인 설정

속성들의 가독성을 높이기 위해 속성명 앞에 엔티티명을 추가 할 수 있다.

Domains ->

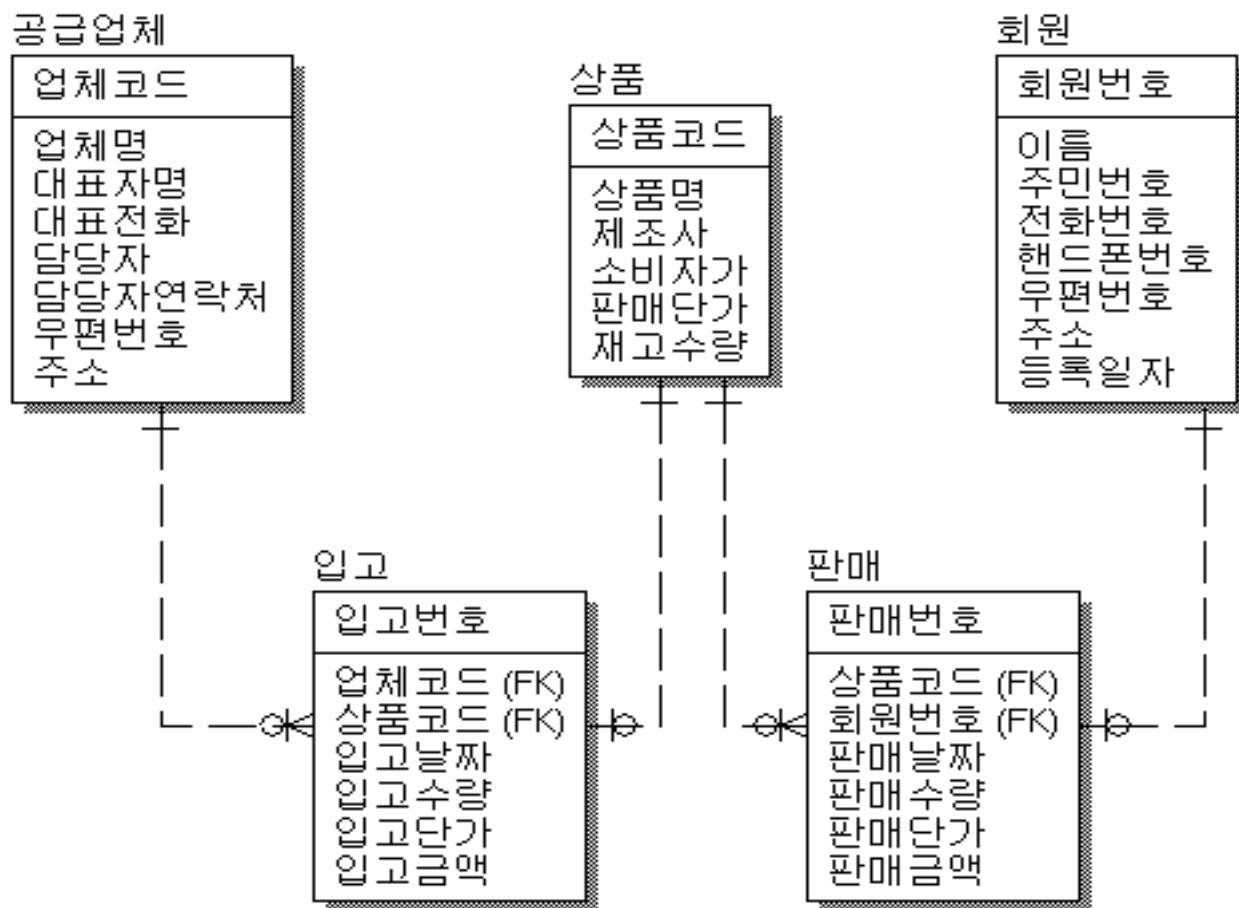
default 오른쪽 버튼 ->

properties 클릭



%EntityName%AttDomain

도메인 설정

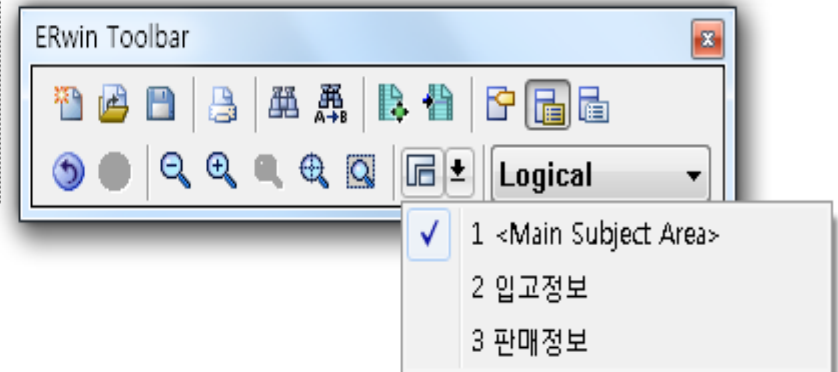
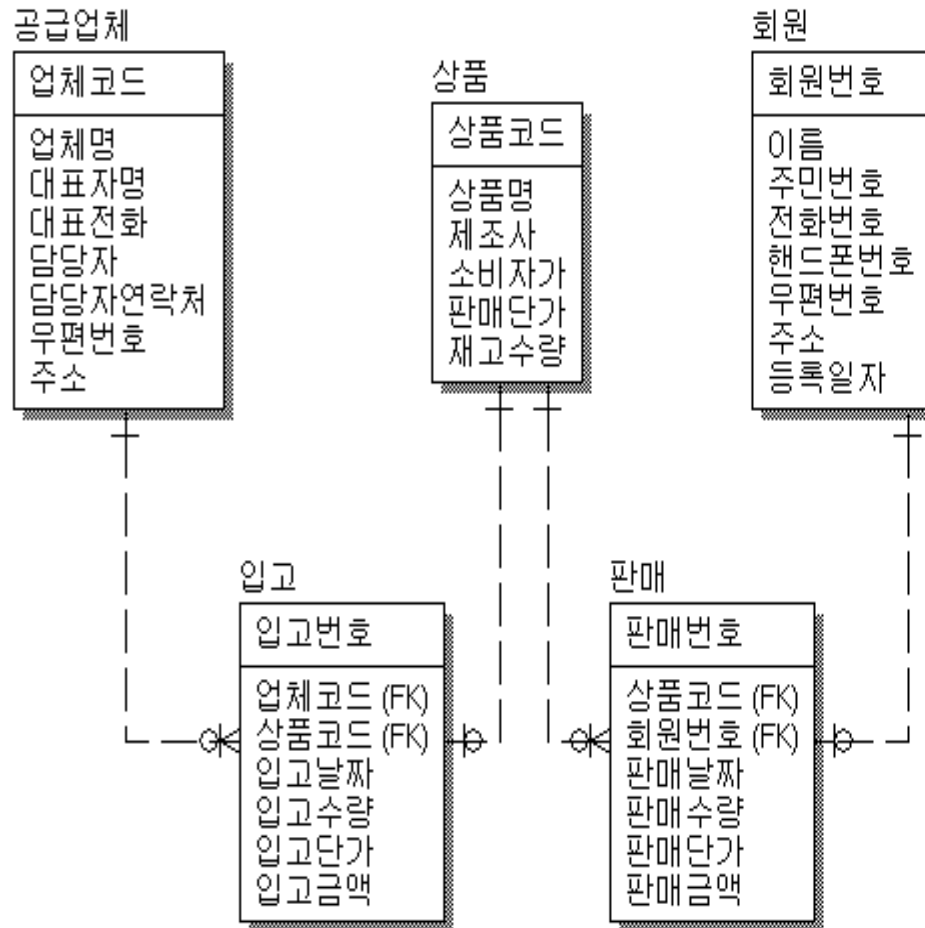


Subject Area

업무분석을 해서 엔티티를 추출하게 되면 많은 엔티티 들이 정의되어 한 화면에 모두 보면서 관리하기가 힘들다. 이러한 경우 업무적으로 관련 있거나 혹은 개발자가 보고자 하는 엔티티들 만을 가지고 새로운 화면을 구성 할 수 있다.(주제별 영역)



Subject Area

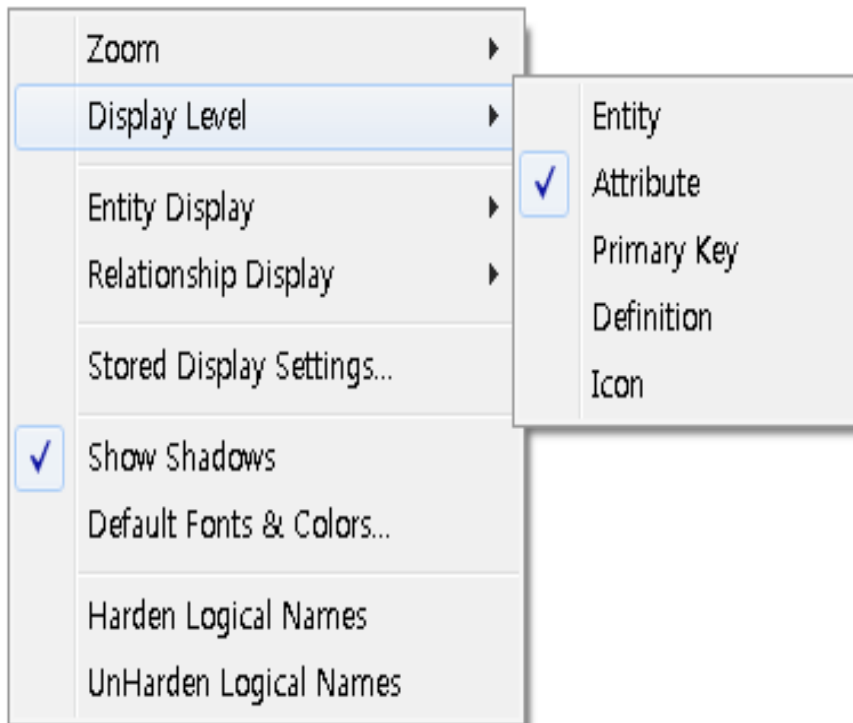


Subject Area연습2.erwin

Stored Display

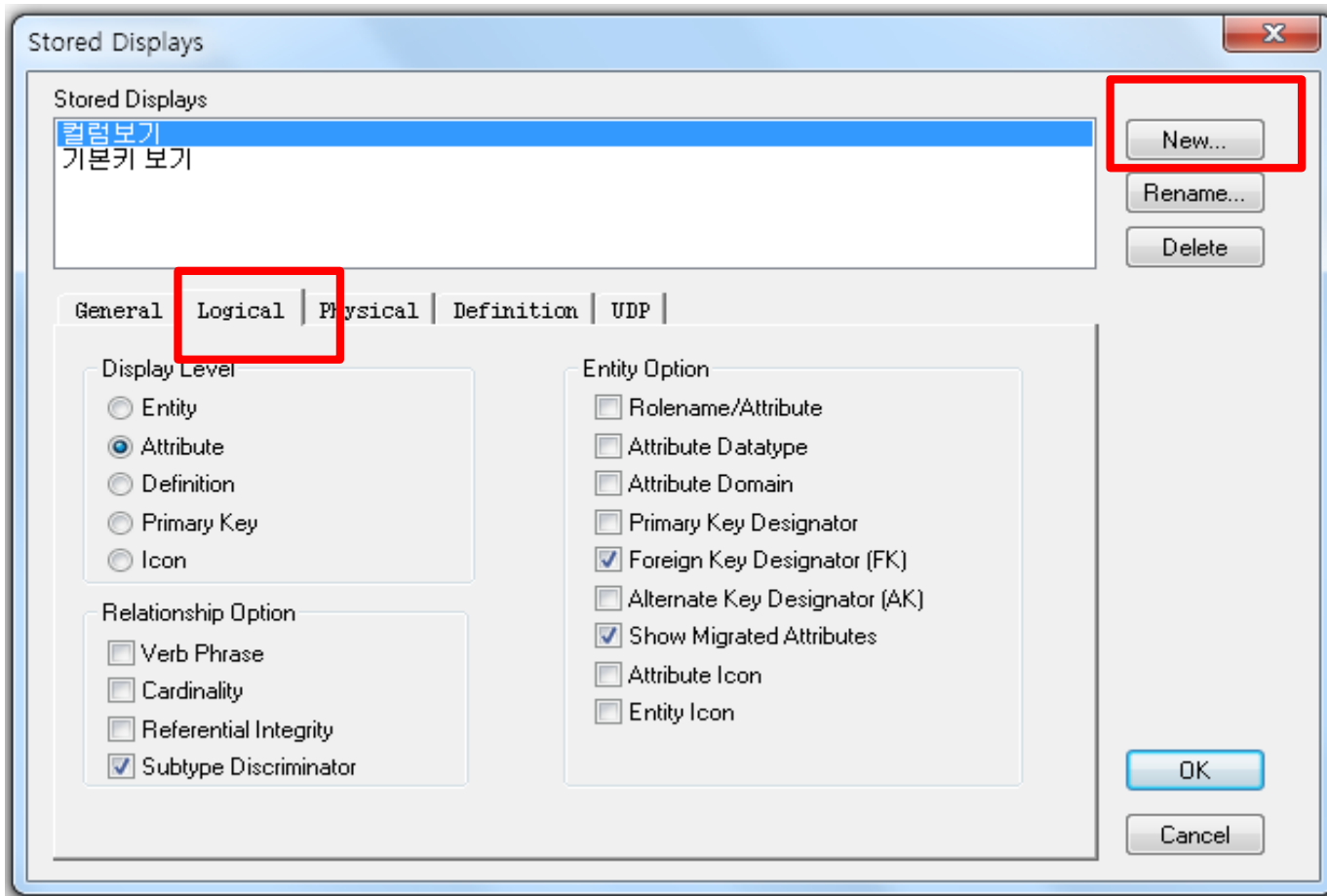
보는 형식을 저장해 놓고 데이터베이스 스키마를 관리자가 좀더 편리하게 볼 수 있도록 하는 기능.

* 바탕화면에서 오른쪽 버튼 클릭



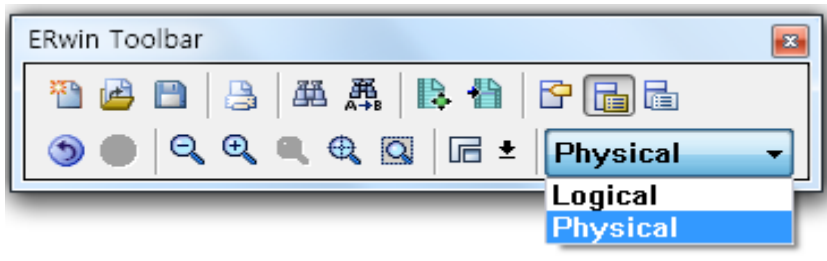
Stored Display

보기 형식을 저장하기 위해서는 다이어그램 바탕화면에서 오른쪽버튼을 누르면 팝업메뉴에서 Stored display Settings.. 메뉴를 선택한다.

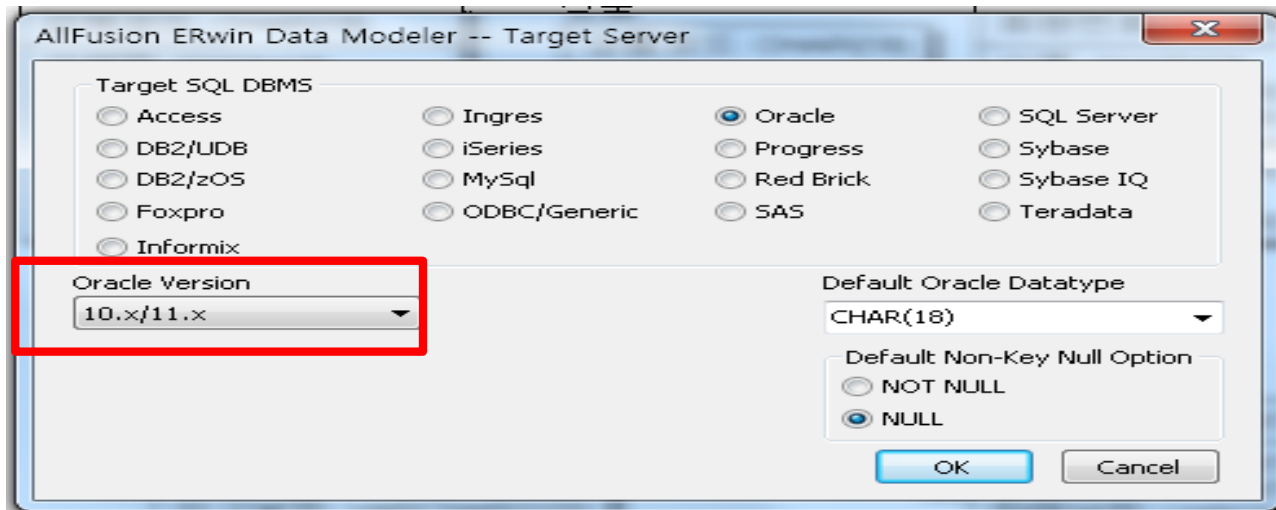


물리적(Physical) 모델링

물리적 데이터베이스로 전환하기 위해서 Erwin Toolbar의 Physical선택



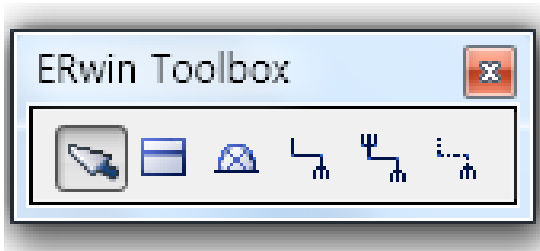
메뉴가 Physical에 관련된 것으로 변경되면 DataBase -> Choose Database를 선택하여 DBMS를 선정한다.



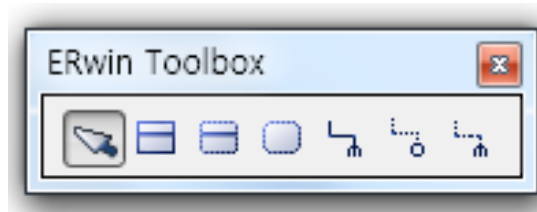
물리적(Physical) 모델링

논리적 모델링에서 물리적 모델링으로 전환 할 경우 ER-Win의 용어가 엔티티(Entity)에서 테이블(table)로 속성(Attribute) 또한 컬럼(Column)으로 바뀌게 된다.

DataBase메뉴는 Physical 모델링 단계에서만 나타난다.



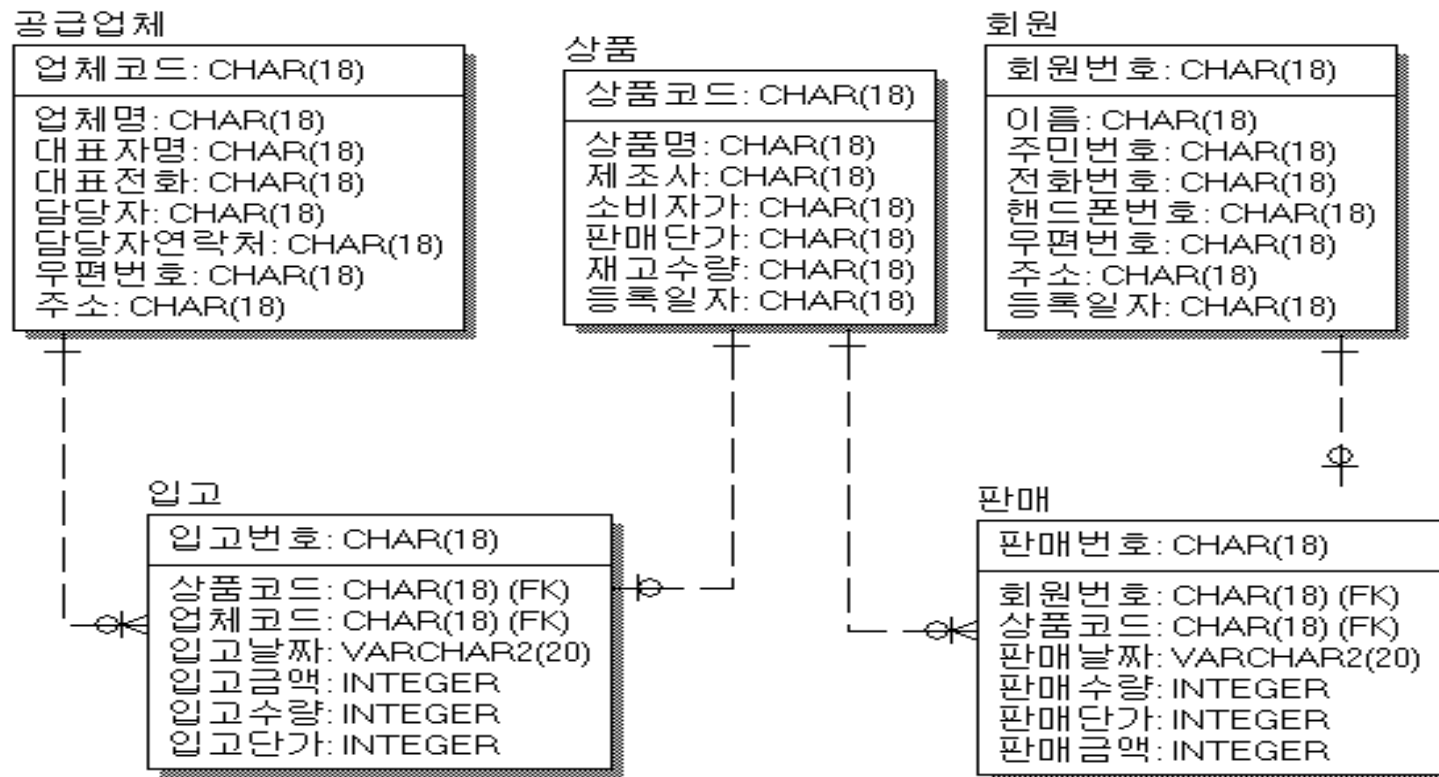
Logical 상태



Physical 상태

Column Data Type과 Size

컬럼의 dataType과 size를 정의하려면 테이블을 선택 한 뒤 오른쪽 버튼을 누르고 팝업메뉴에서 Columns 메뉴를 선택한다.



Check , default

Check제약조건은 테이블을 만들거나 수정 하면서 컬럼에 직접 정의하는 제약조건으로 데이터가 컬럼에 입력 될 수 있는 경우의 수를 제한 함으로써 잘못된 데이터가 입력 될 수 없도록 하기 위해 사용한다.

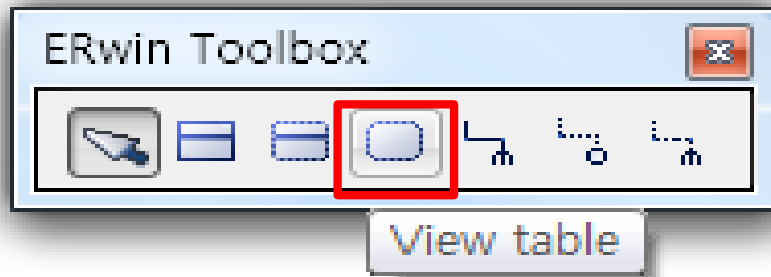
성적

번호: NUMBER(1)
이름: CHAR(18)
전화: CHAR(18)
국어: NUMBER(3)
영어: NUMBER(3)
수학: NUMBER(3)
총점: NUMBER(3)
평균: NUMBER(5,2)
학점: CHAR(1)

제약조건	적용컬럼	내용
Check	전화	전화번호입력 양식 정의 Ex) (3자리)-3자리-4자리
Check	학점	학점에 허용될 값 지정(A,B,C,D,F)
Check	국어, 영어, 수학	0 ~100사이의 값의 입력범위를 각 컬럼에 적용
default	국어, 영어, 수학	값을 입력하지 않는 경우에 0값 입력

뷰 정의

ER-Win에서 뷰를 만들려면 물리적 모델링단계에서 ER-Win Toolbox에 있는 View table을 선택해서 만든다.



```
CREATE VIEW 판매View
```

```
AS
```

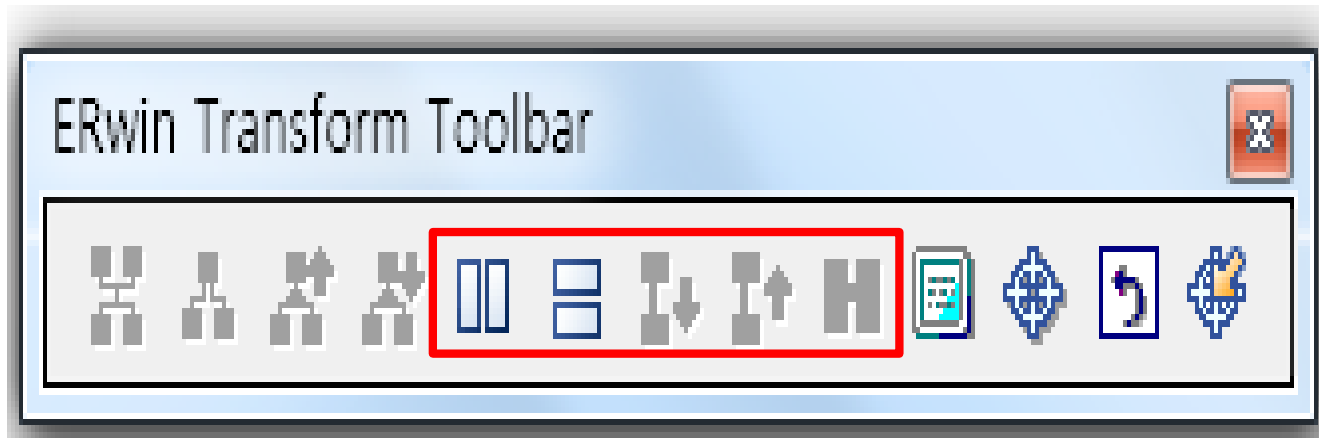
```
SELECT 판매.판매번호,회원.이름,상품.상품명,판매.판매날짜,  
       판매.판매수량,판매.판매단가,판매.판매금액
```

```
FROM 판매,회원,상품
```

```
WHERE 판매.상품코드 = 상품.상품코드 and 판매.회원번호 = 회원.회원번호;
```

역 정규화(Denormalization) 정의 방법

역정규화란 물리적 데이터베이스 모델링 단계에서 퍼포먼스 향상을 위해 정규화에 위배되는 설계를 하는 것이다.(슬라이드 75 참조)



역 정규화(Denormalization) 정의 방법

사원

사원번호: CHAR(18)
이름: CHAR(18)
주민번호: CHAR(18)
전화번호: CHAR(18)
핸드폰번호: CHAR(18)
e_mail: CHAR(18)
우편번호: CHAR(18)
주소: CHAR(18)
입사일: CHAR(18)
취미: CHAR(18)
종교: CHAR(18)
시력좌: CHAR(18)
시력우: CHAR(18)
결혼유무: CHAR(18)
키: CHAR(18)
몸무게: CHAR(18)
혈액형: CHAR(18)

- 이 테이블은 한 테이블에 컬럼이 너무 많다.
- 한테이블에 컬럼의 개수는 10개 이내로 하는 것이 좋다.
- 컬럼이 자주 사용되는 것과 자주 사용되지 않는 컬럼을 기준으로 Vertical Patition(컬럼분할) 역정규화를 한다.



역 정규화(Denormalization) 정의 방법

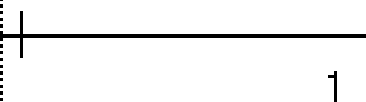
- Vertical Partition 한 후 식별관계와 차수를 지정해준다.

사원

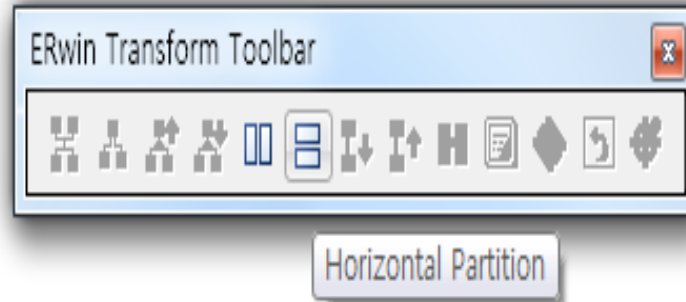
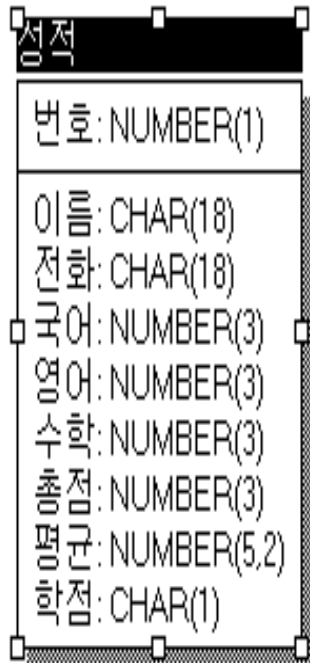
사원번호: CHAR(18)
이름: CHAR(18)
주민번호: CHAR(18)
전화번호: CHAR(18)
핸드폰번호: CHAR(18)
e_mail: CHAR(18)
우편번호: CHAR(18)
주소: CHAR(18)
입사일: CHAR(18)

부가정보

사원번호: CHAR(18) (FK)
취미: CHAR(18)
종교: CHAR(18)
시력좌: CHAR(18)
시력우: CHAR(18)
결혼유무: CHAR(18)
키: CHAR(18)
몸무게: CHAR(18)
혈액형: CHAR(18)



역 정규화(Denormalization) 정의 방법



- 성적을 기준으로 성적상, 성적중, 성적하 테이블로 분리한다.
- Check제약조건 추가
- 성적상 — 총점 ≥ 250
- 성적중 — 총점 between 180 and 249
- 성적하 — 총점 < 180

역 정규화(Denormalization) 정의 방법

성적상

번호: NUMBER(1)
이름: CHAR(18)
전화: CHAR(18)
국어: NUMBER(3)
영어: NUMBER(3)
수학: NUMBER(3)
총점: NUMBER(3)
평균: NUMBER(5,2)
학점: CHAR(1)

성적중

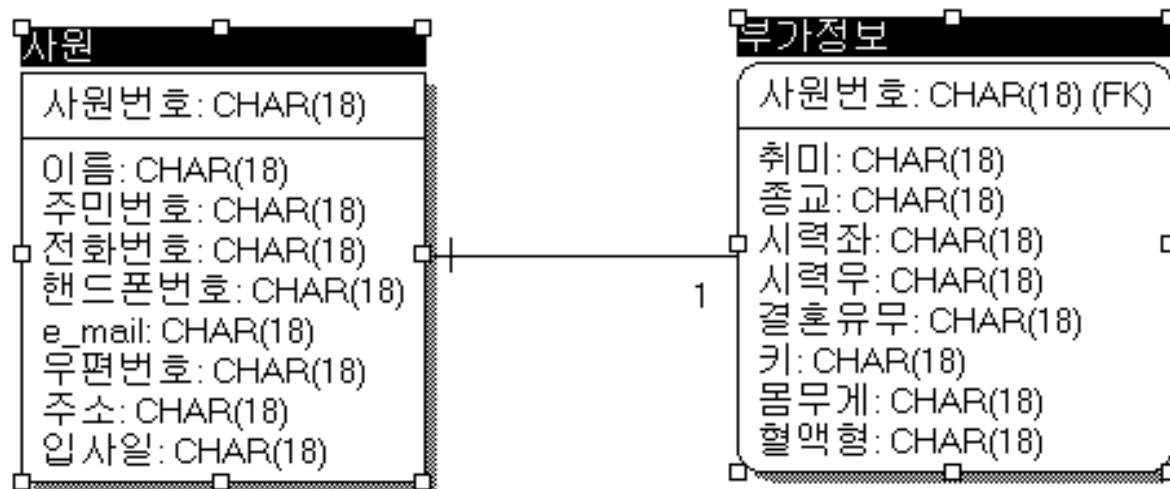
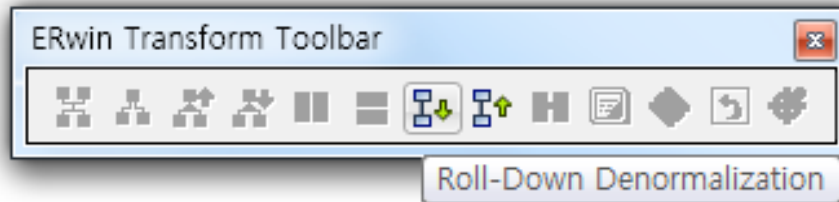
번호: NUMBER(1)
이름: CHAR(18)
전화: CHAR(18)
국어: NUMBER(3)
영어: NUMBER(3)
수학: NUMBER(3)
총점: NUMBER(3)
평균: NUMBER(5,2)
학점: CHAR(1)

성적하

번호: NUMBER(1)
이름: CHAR(18)
전화: CHAR(18)
국어: NUMBER(3)
영어: NUMBER(3)
수학: NUMBER(3)
총점: NUMBER(3)
평균: NUMBER(5,2)
학점: CHAR(1)

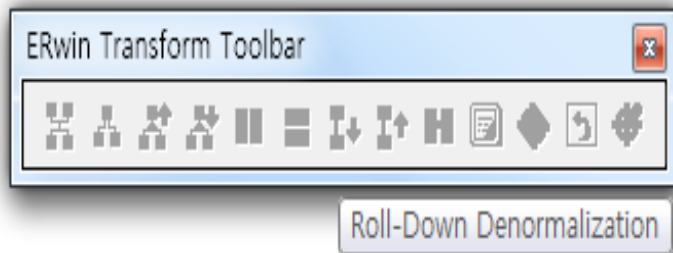


역 정규화(Denormalization) 정의 방법



- 테이블을 분리하고 보니 연관된 정보를 보기 위해 조인을 수행 해야 하고 데이터를 입력,수정,삭제할 때 관련된 해당테이블에 데이터가 존재하는지 참조해야 하는 번거로움이 있다. 그래서 테이블을 합치겠다.

역 정규화(Denormalization) 정의 방법

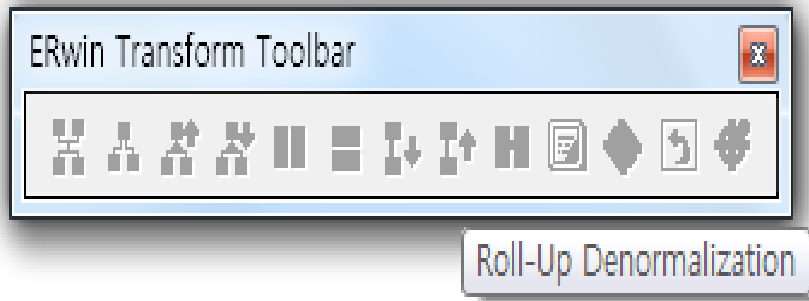


- Roll-Dow Denormalization
은 자식 테이블을 기준으로 합
치겠다.

부가정보

사원번호: CHAR(18)
취미: CHAR(18)
종교: CHAR(18)
시력좌: CHAR(18)
시력우: CHAR(18)
결혼유무: CHAR(18)
키: CHAR(18)
몸무게: CHAR(18)
혈액형: CHAR(18)
이름: CHAR(18)
주민번호: CHAR(18)
전화번호: CHAR(18)
우편번호: CHAR(18)
핸드폰번호: CHAR(18)
e_mail: CHAR(18)
주소: CHAR(18)
입사일: CHAR(18)

역 정규화(Denormalization) 정의 방법



- Roll-Up Denormalization
은 부모 테이블을 기준으로
합치겠다.

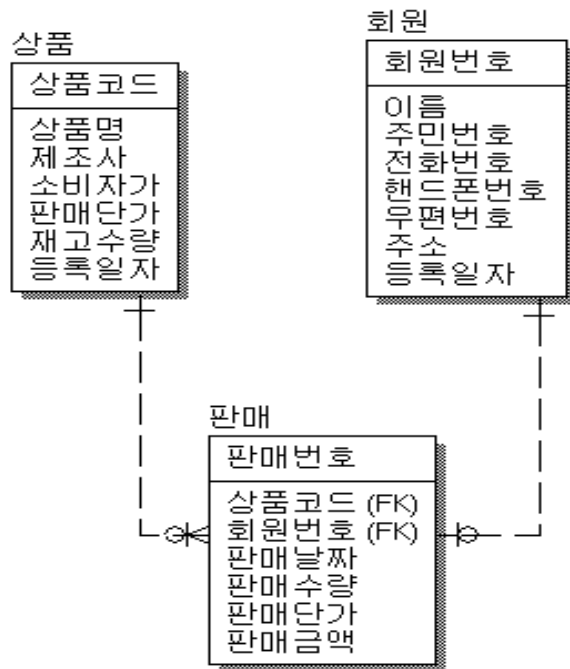
사원

사원번호: CHAR(18)
이름: CHAR(18)
주민번호: CHAR(18)
전화번호: CHAR(18)
우편번호: CHAR(18)
핸드폰번호: CHAR(18)
e_mail: CHAR(18)
주소: CHAR(18)
입사일: CHAR(18)
취미: CHAR(18)
종교: CHAR(18)
시력좌: CHAR(18)
시력우: CHAR(18)
결혼유무: CHAR(18)
키: CHAR(18)
몸무게: CHAR(18)
혈액형: CHAR(18)

역 정규화(Denormalization) 정의 방법

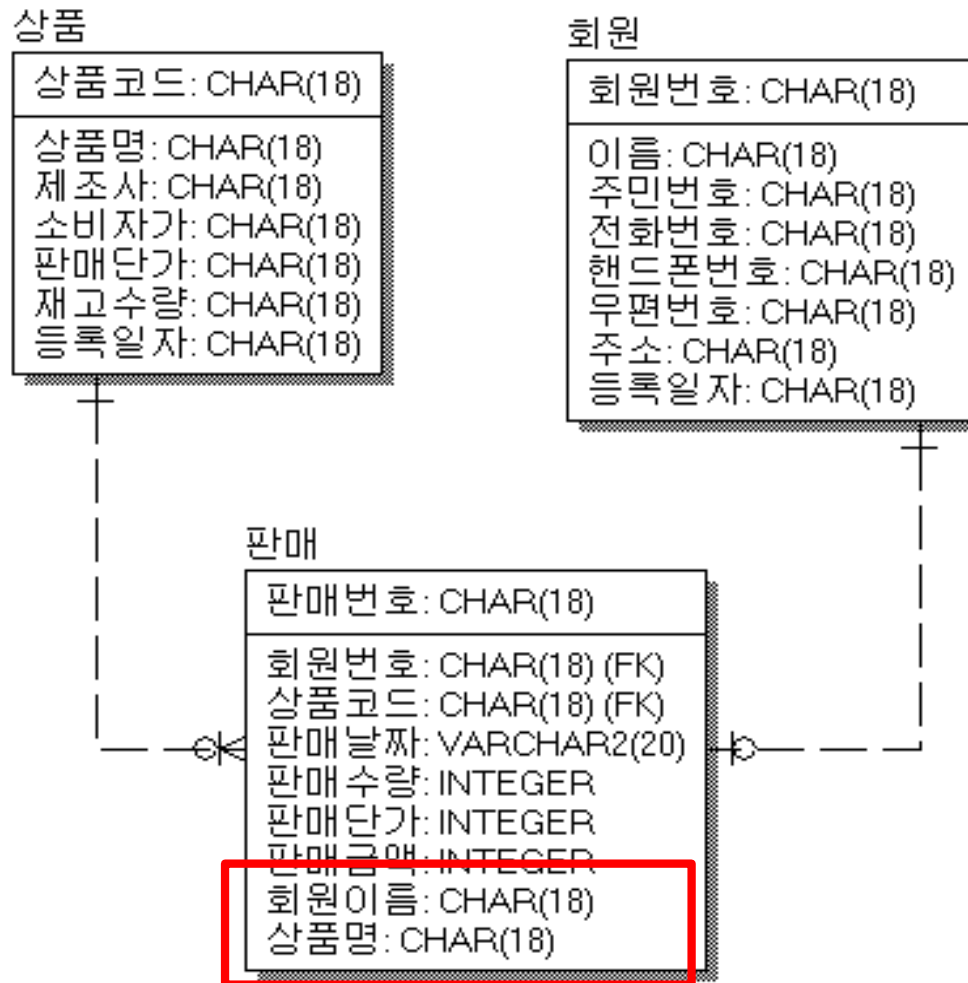


- Column Denormalization은 주로 조인의 속도를 향상시키기 위해서 조회하는 테이블에 컬럼을 중복하는 것이다.



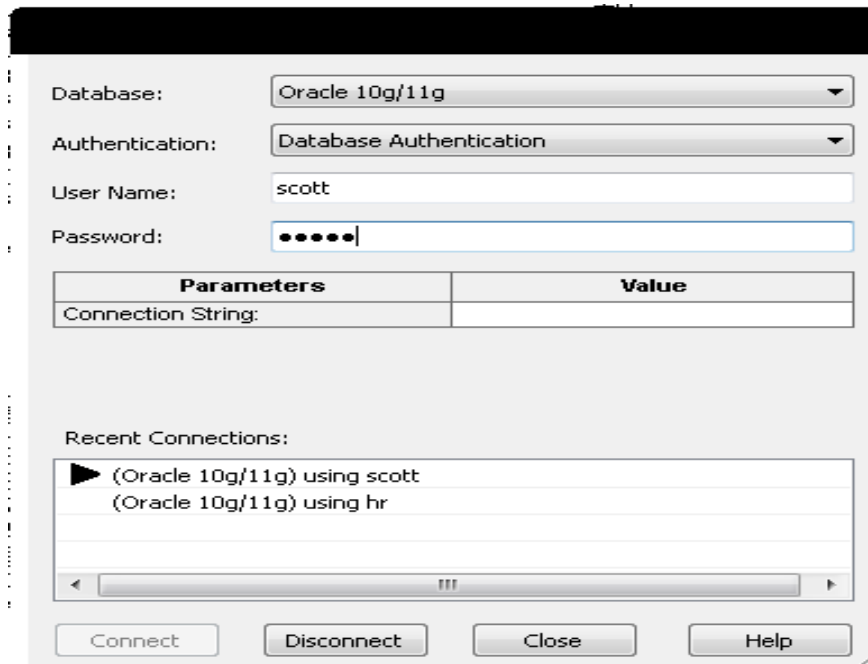
- 판매테이블을 조회할 때 회원이름과 상품명을 함께 조회하기 위해 조인을 수행해야 한다. 이러한 경우 조회의 성능 향상을 위해 판매테이블에 회원이름과 상품명을 중복하게 한다.

역 정규화(Denormalization) 정의 방법



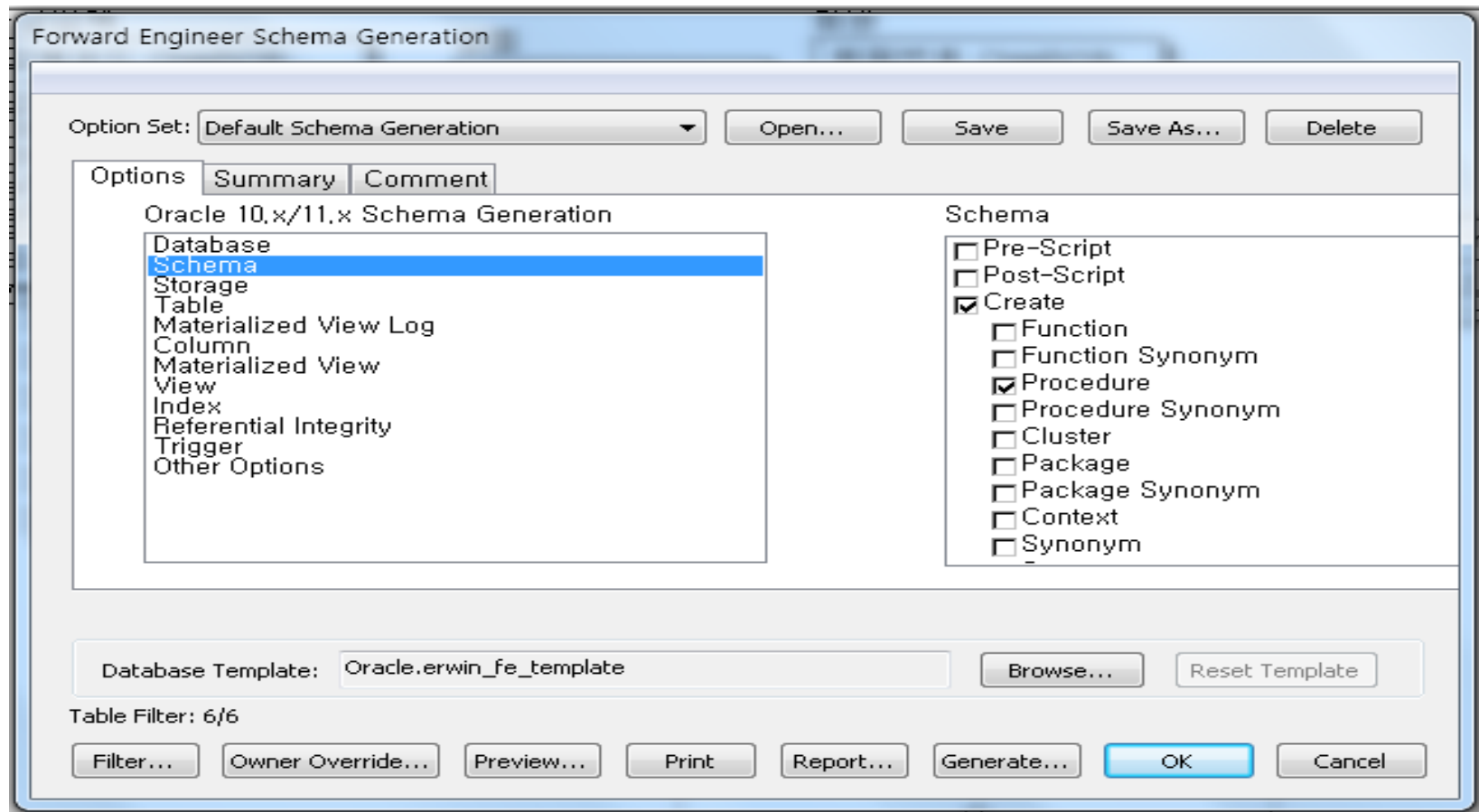
데이터베이스 스키마 생성

- 물리적 데이터 모델링 단계를 거치면서 데이터베이스의 스키마를 모두 정의했다면 이렇게 정의된 내용을 실제 데이터베이스 객체로 만들어 질 수 있도록 데이터베이스 스키마를 생성한다.
- 데이터베이스에 연결하기 위해서는 ER-win의 물리적 모델링단계로 전환한 후 Database메뉴 -> DataBase Connection메뉴를 선택한다.



데이터베이스 스키마 생성

- 정상적으로 연결이 되었다면 데이터베이스 스키마를 생성한다.
- Tools메뉴 -> Forward Engineer -> Schema generation 메뉴를 선택한다.



감

사

함

니

다

!