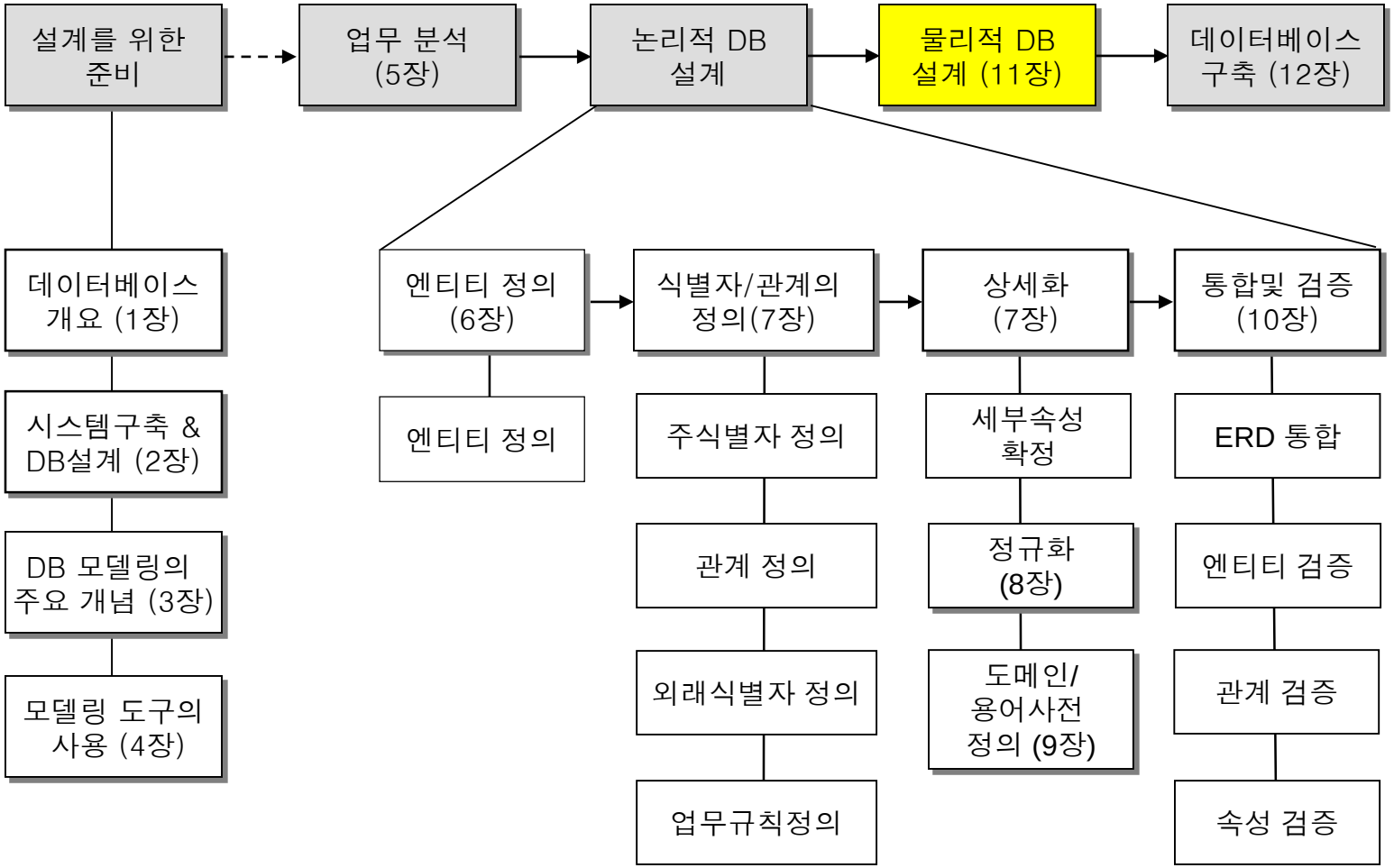
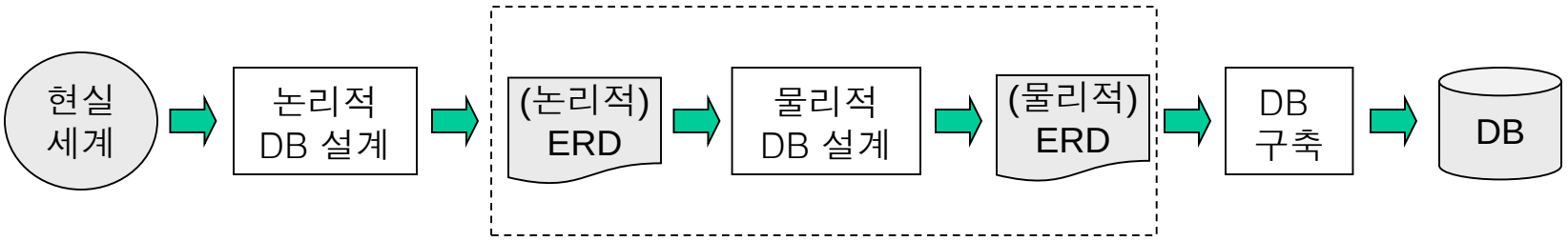
11장. 논리적 설계를 물리적 설계로 전환하기

- ❑ 물리적 설계의 개요
- ❑ 테이블, 컬럼, 키로의 전환
- ❑ 반정규화
- ❑ 뷰의 설계
- ❑ 인덱스의 설계
- ❑ 테이블 기술서


설계를 위한 준비
업무 분석 (5장)
논리적 DB 설계
물리적 DB 설계 (11장)
데이터베이스 구축 (12장)
데이터베이스 개요 (1장)
시스템구축 & DB설계 (2장)
DB 모델링의 주요 개념 (3장)
모델링 도구의 사용 (4장)
엔티티 정의 (6장)
엔티티 정의
식별자/관계의 정의(7장)
주식별자 정의
관계 정의
외래식별자 정의
업무규칙정의
상세화 (7장)
세부속성 확정
정규화 (8장)
도메인/ 용어사전 정의 (9장)
통합및 검증 (10장)
ERD 통합
엔티티 검증
관계 검증
속성 검증

11.1 개요

- 물리적 데이터베이스 설계
  - 엔티티, 관계, 속성 등의 요소가 테이블, 컬럼, 키 등으로 변환
  - 성능을 고려하여 반정규화 과정 시행
  - 인덱스와 뷰를 설계



<그림 11.1> 데이터베이스 설계 과정

11.1 개요

❑ 논리적 DB 설계와 물리적 DB 설계의 차이점

논리적 DB 설계 (데이터 모델링)	물리적 DB 설계
DBMS 의 종류나 제품에 상관 없이 진행 (ERD 는 어떤 데이터베이스를 사용해도 적용 가능)	특정 DBMS 를 전제로 진행 (적용 DBMS의 특성을 고려함)
엔티티 (entity)	테이블 (table)
속성 (attribute)	컬럼 (column)
주식별자 (primary identifier)	기본키 (primary key)
외래별자 (foreign identifier)	외래키 (foreign key)
-	뷰 (view)
-	인덱스 (index)

<그림 11.2> 논리적 DB 설계와 물리적 DB 설계의 비교

11.2 테이블, 컬럼, 키로의 변환

❑ 논리적 DB 설계→ 물리적 DB 설계

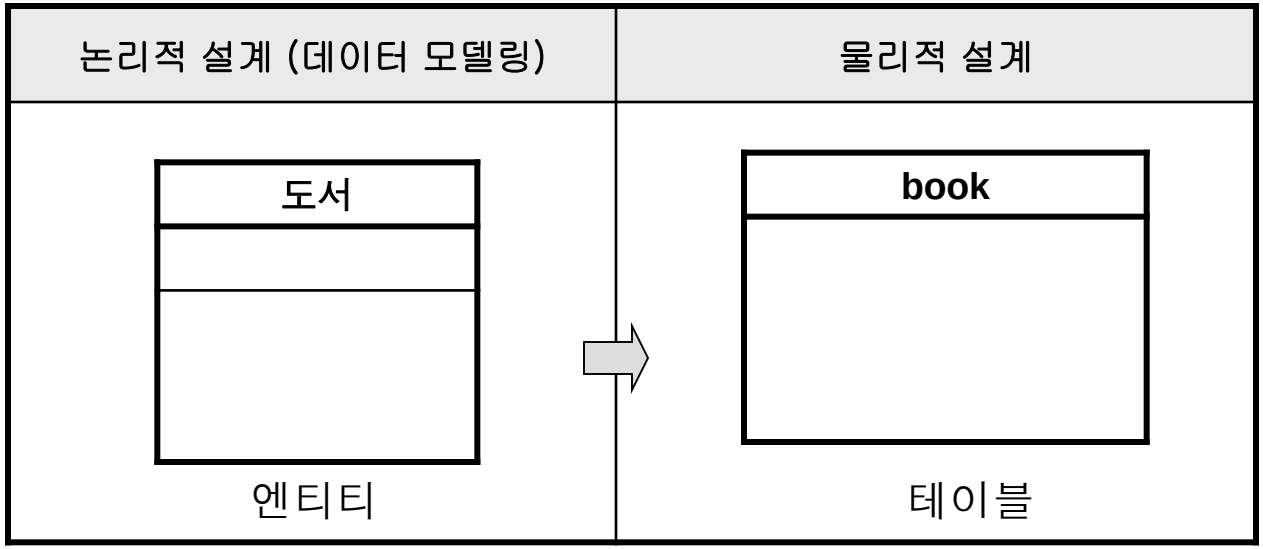
- 물리적 데이터베이스의 기본적인 내용은 논리적 설계의 산출물인 ERD의 요소들을 관계형 데이터베이스의 요소들로 전환하는 것

논리적 설계 (데이터 모델링)	물리적 설계	데이터베이스
엔티티 (entity)	테이블 (table)	테이블
속성 (attribute)	컬럼 (column)	컬럼
주식별자 (primary identifier)	기본키 (primary key)	기본키
외래식별자 (foreign identifier)	외래키 (foreign key)	외래키
관계(relationship)	관계(relationship)	
관계의 카디낼러티	관계의 카디낼러티	-
관계의 참여도	관계의 참여도	-

<그림 11.3> ERD 요소들의 전환

11.2 테이블, 컬럼, 키로의 변환

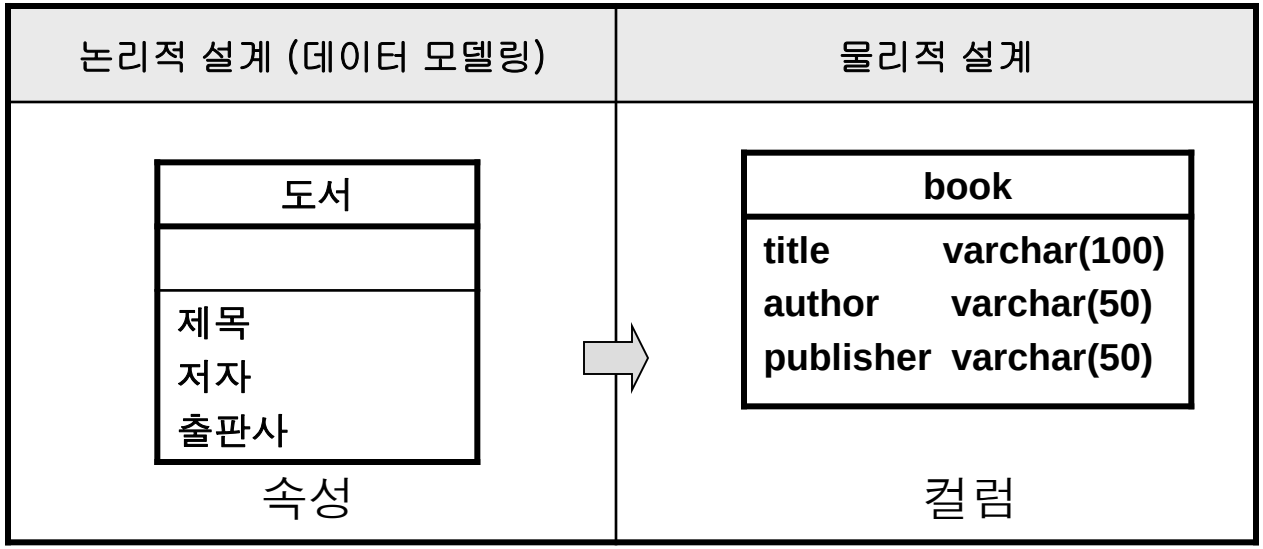
- 엔티티 → 테이블
  - 엔티티는 그대로 테이블로 변환됨
  - 한글 엔티티명 → 영문 테이블명



<그림 11.4> 엔티티 → 테이블 전환

11.2 테이블, 컬럼, 키로의 변환

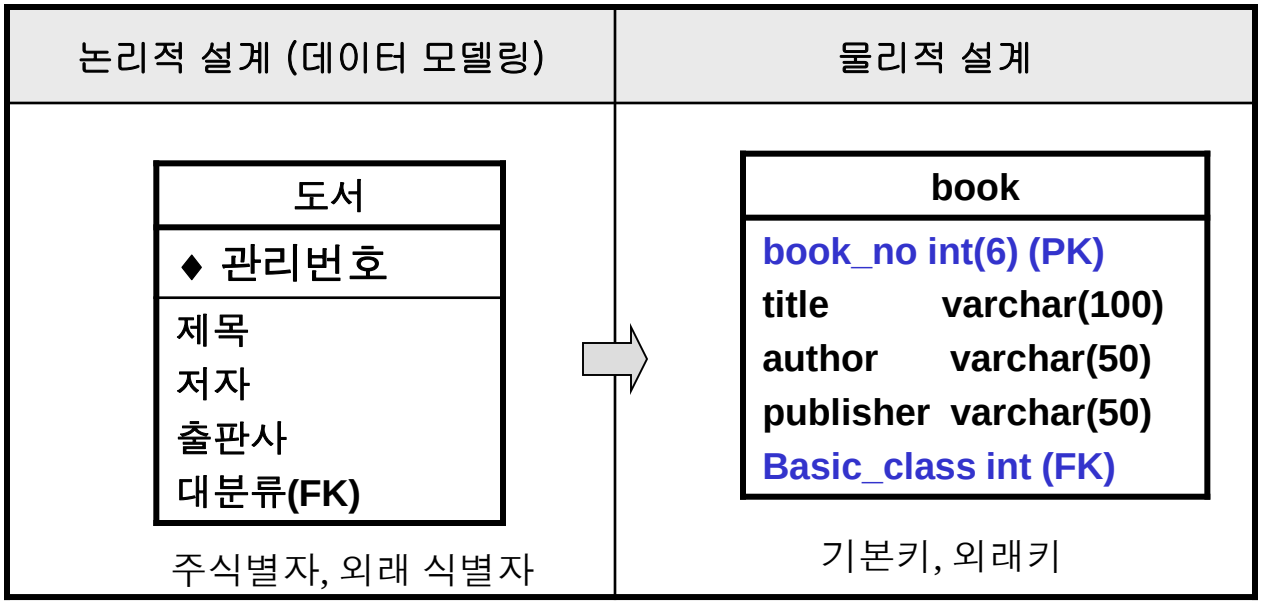
- 속성 → 컬럼
  - 한글 속성명 → 영문 컬럼명 (용어사전 이용)
  - 데이터 타입의 지정



<그림 11.5> 속성 → 컬럼 전환

11.2 테이블, 컬럼, 키로의 변환

❑ 주식별자 → 기본키, 외래식별자 → 외래키

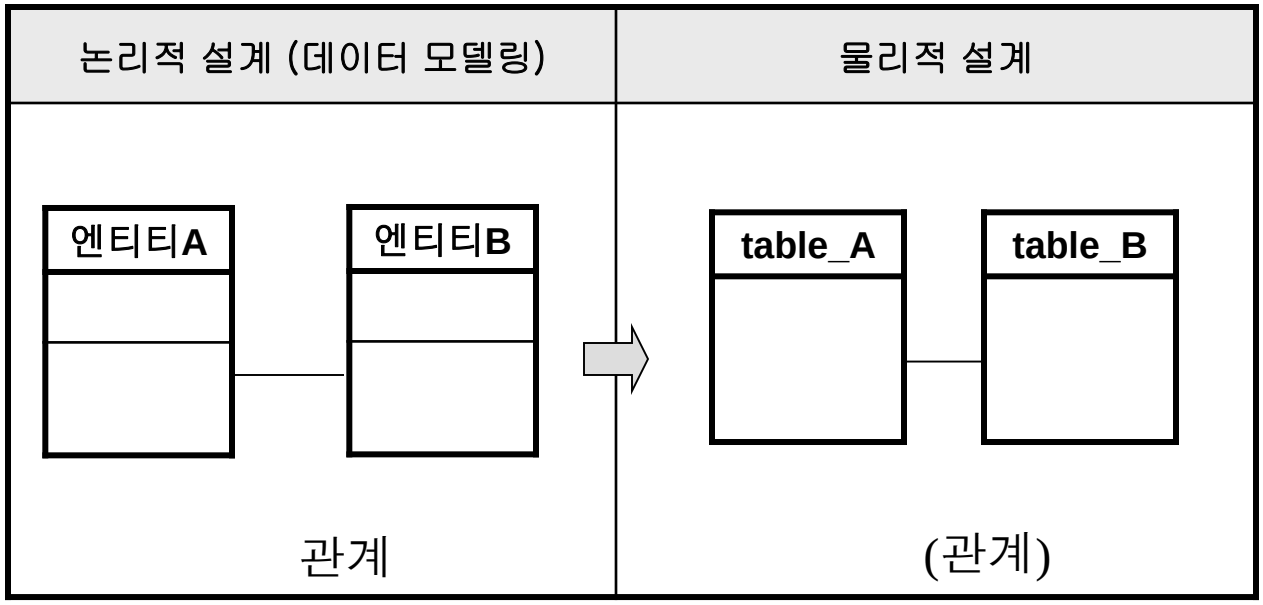


<그림 11.6> 주식별자, 외래 식별자 → 기본키, 외래키 전환

11.2 테이블, 컬럼, 키로의 변환

❑ 관계의 전환

- 논리적 설계에서 엔티티간의 관계(relationship)는 물리적 설계에서도 그대로 유지

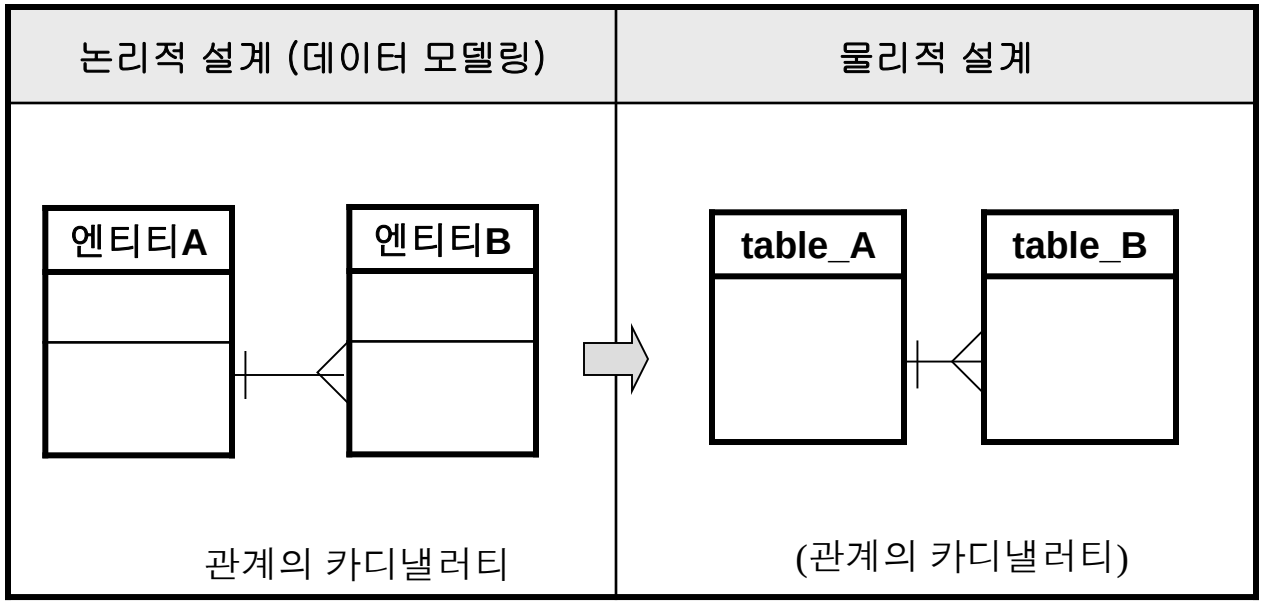


<그림 11.7> 관계의 전환

11.2 테이블, 컬럼, 키로의 변환

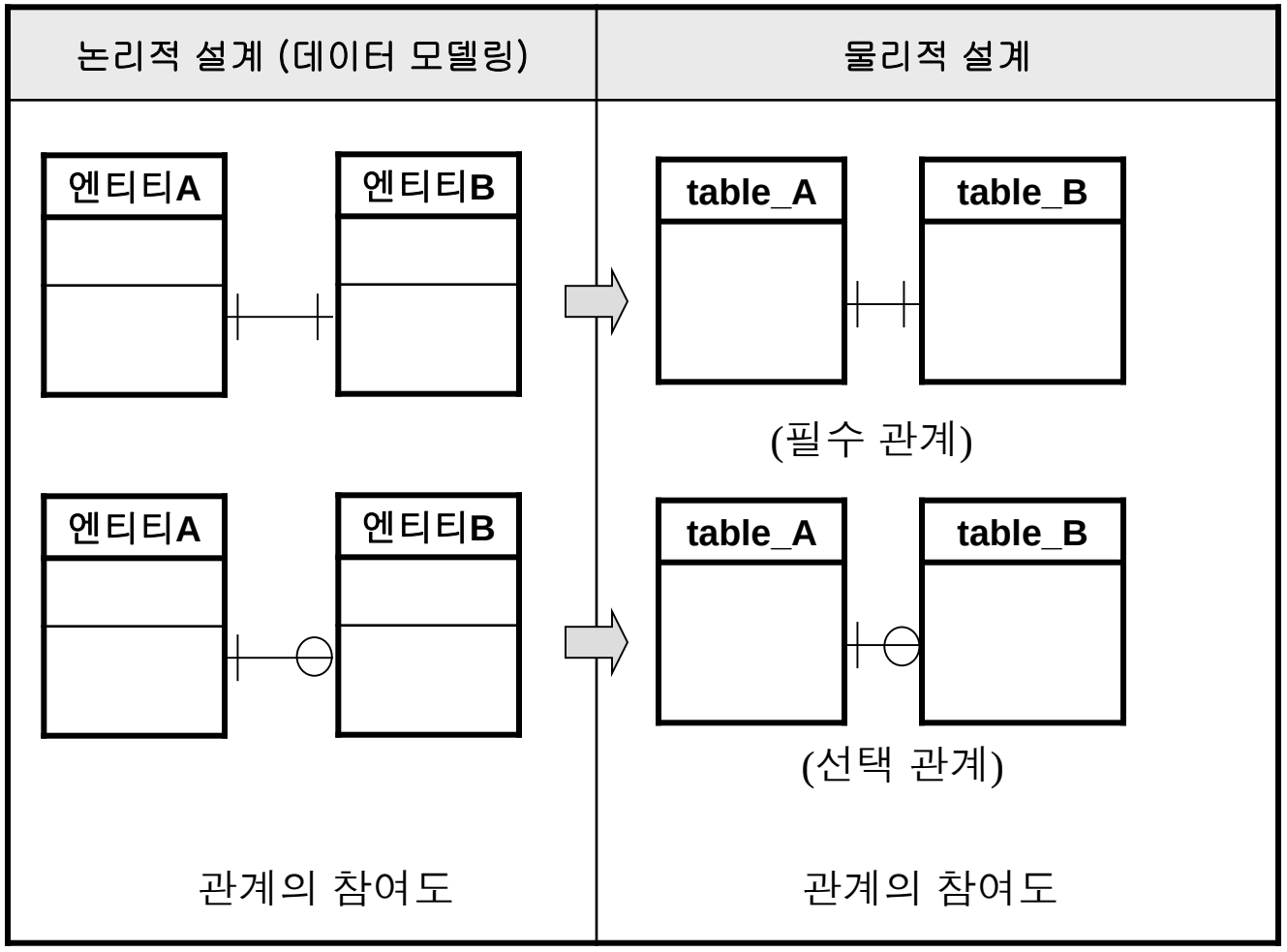
❑ 관계의 전환

- 카디낼러티, 선택/필수도 그대로 적용됨



<그림 11.8> 관계의 카디낼러티 전환

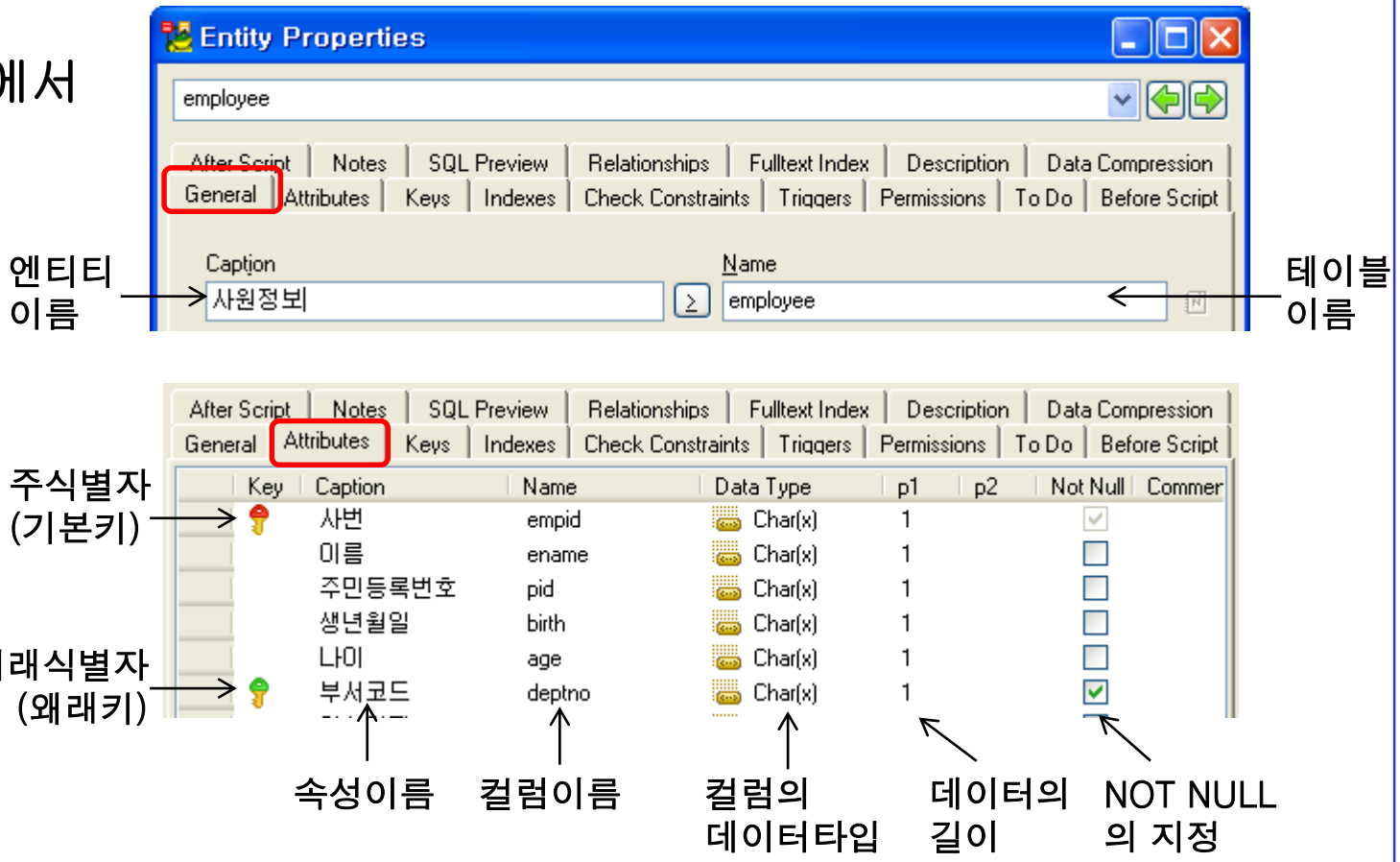
11.2 테이블, 컬럼, 키로의 변환



<그림 11.9> 관계의 참여도 전환

11.2 테이블, 컬럼, 키로의 변환

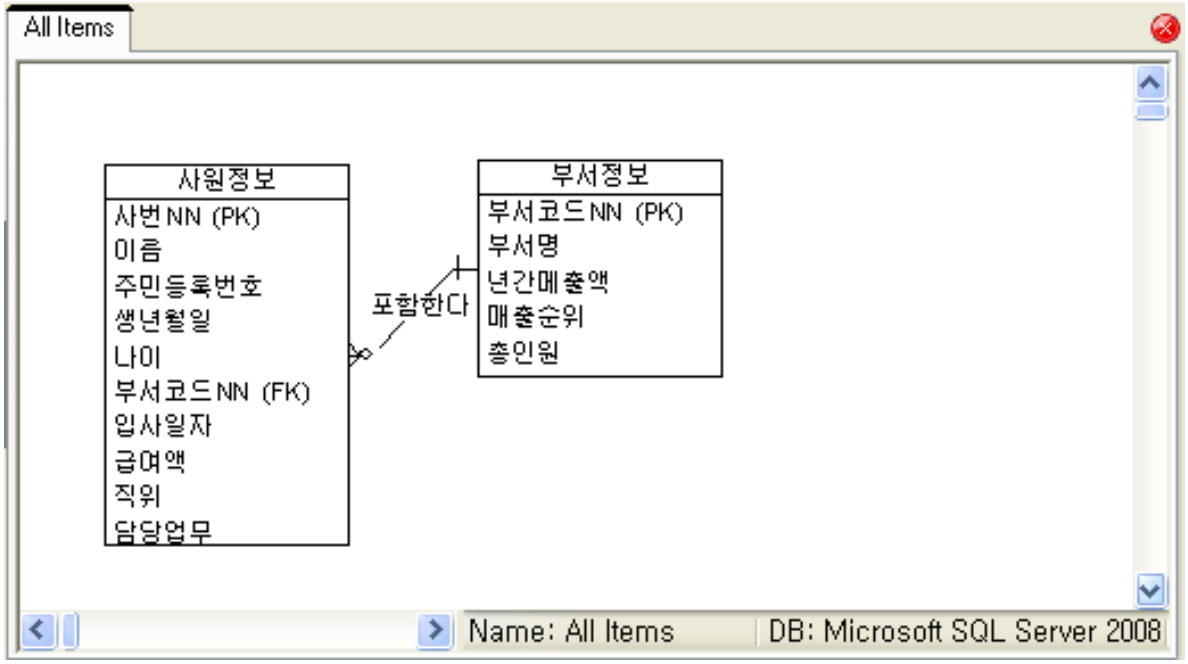
❑ 모델링 도구에서 물리적 DB 설계



<그림 11.10> 엔티티와 테이블의 편집

11.2 테이블, 컬럼, 키로의 변환

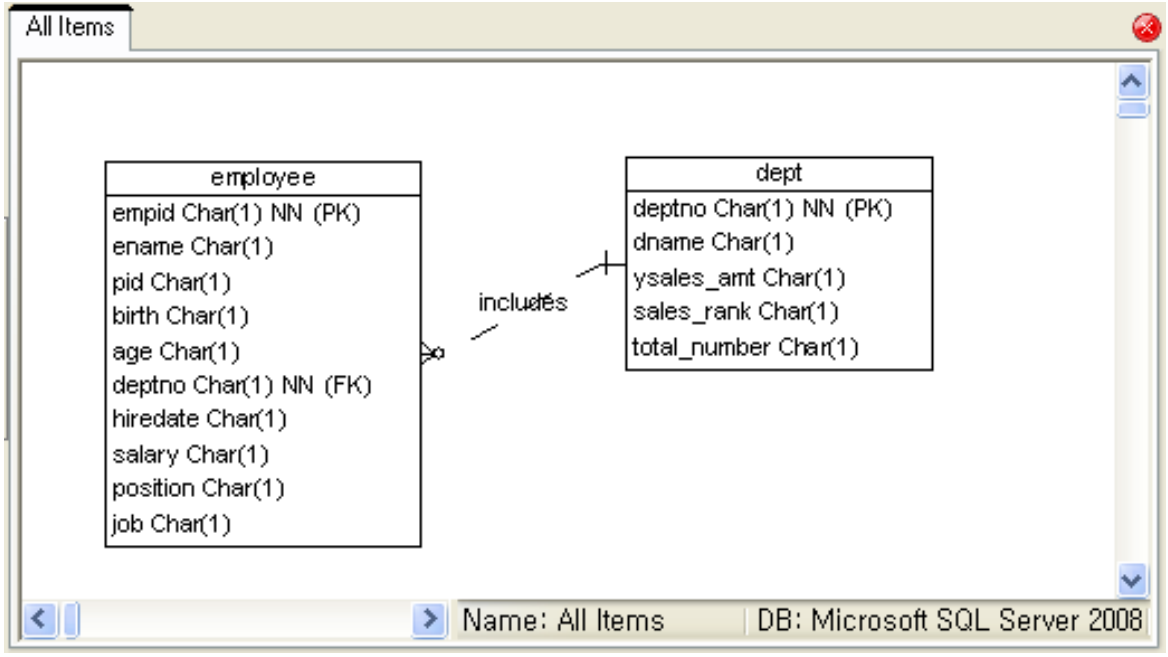
- 모델링 도구에서 물리적 DB 설계
  - 논리적 ERD 보기
    - 메인 메뉴의 [view]에서 [logical view] 선택을 해제



<논리적 ERD>

11.2 테이블, 컬럼, 키로의 변환

- 모델링 도구에서 물리적 DB 설계
  - 물리적 ERD 보기
    - 메인 메뉴의 [view]에서 [Physical view]를 선택



<물리적 ERD>

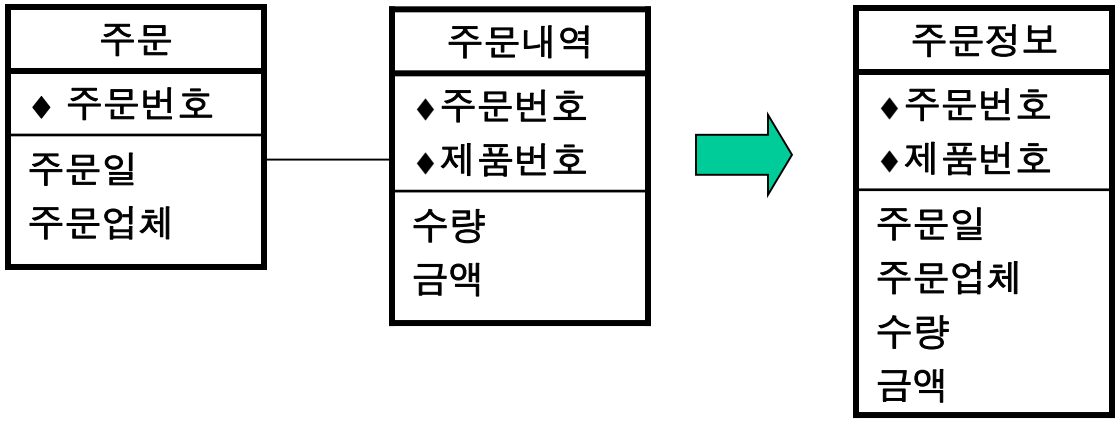
11.3 반정규화

- 반정규화(de-normalization)란
  - 정규화의 반대 개념
  - 어느 정도의 중복은 감수하고 데이터베이스의 성능(특별히 검색 속도)을 향상시키고자 ERD를 전규화 이전 상태로 변경하는 과정
  - 정규화 과정은 엔티티들을 분리하는 형태로 진행이 된다면 반정규화 과정은 엔티티들을 통합해 가는 형태로 진행
  - 반정규화를 하는 경우 데이터 무결성에 문제가 있을 수 있으므로 대비책 필요

  - 반정규화를 하지 않고도 성능을 향상시킬 수 있다면 그것이 더 나은 대안이다

11.3 반정규화

❑ 엔티티 통합/분할에 의한 반정규화
- 두 엔티티를 조인하는데 걸리는 시간을 절약하려는 목적
- 항상 혹은 대부분 조인에 의한 검색을 하고, 검색이 빈번히 이루어지는 두개의 엔티티를 대상으로 한다



<그림 11.13> 두 엔티티의 통합에 의한 반정규화

11.3 반정규화

❑ 엔티티 통합/분할에 의한 반정규화

(반정규화 이전)

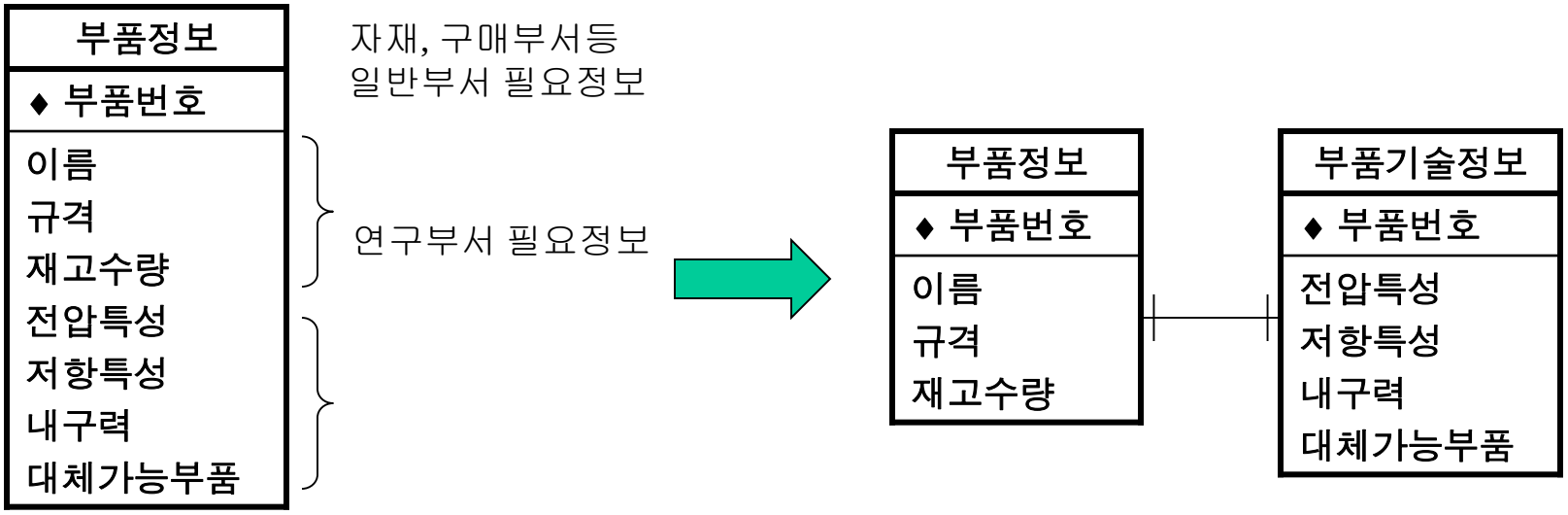
```
SELECT 주문번호, 주문일, 주문업체, 제품번호, 수량, 금액
FROM   주문, 주문내역
WHERE  주문.주문번호 = 주문내역.주문번호 ;
```

(반정규화 이후)

```
SELECT 주문번호, 주문일, 주문업체, 제품번호, 수량, 금액
FROM   주문정보;
```

11.3 반정규화

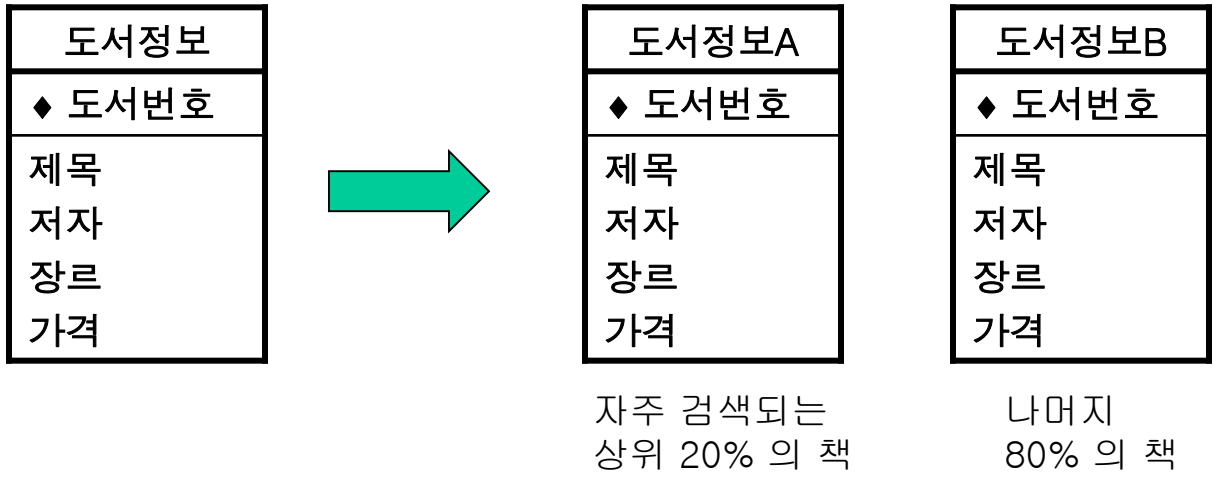
- 엔티티 통합/분할에 의한 반정규화
  - 엔티티의 튜플수 및 속성의 수가 매우 많고, 엔티티의 속성들이 그룹화되어 각 그룹이 특정 부서 혹은 응용 프로그램에 의해서만 사용될 때
  - 엔티티의 데이터 크기 감소, 검색의 분산



<그림 11.14> 엔티티 수직분할에 의한 반정규화

11.3 반정규화

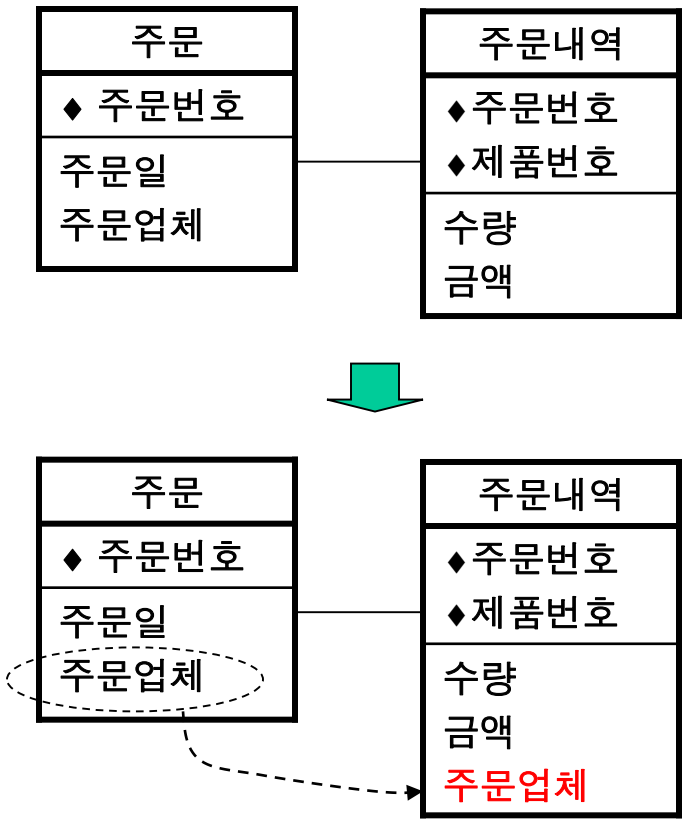
- 엔티티 통합/분할에 의한 반정규화
  - 튜플의 검색 빈도가 다르다는 점을 이용
  - 엔티티의 데이터 크기 감소 효과



<그림 11.15> 엔티티 수평분할에 의한 반정규화

11.3 반정규화

- 속성의 중복에 의한 반정규화
  - 조인하여 가져다 사용하는 속성의 수가 적을 때는 엔티티의 통합은 비효율적임
  - 죠인에 의해서 가져오는 속성을 중복해서 저장하는 편이 더 합리적

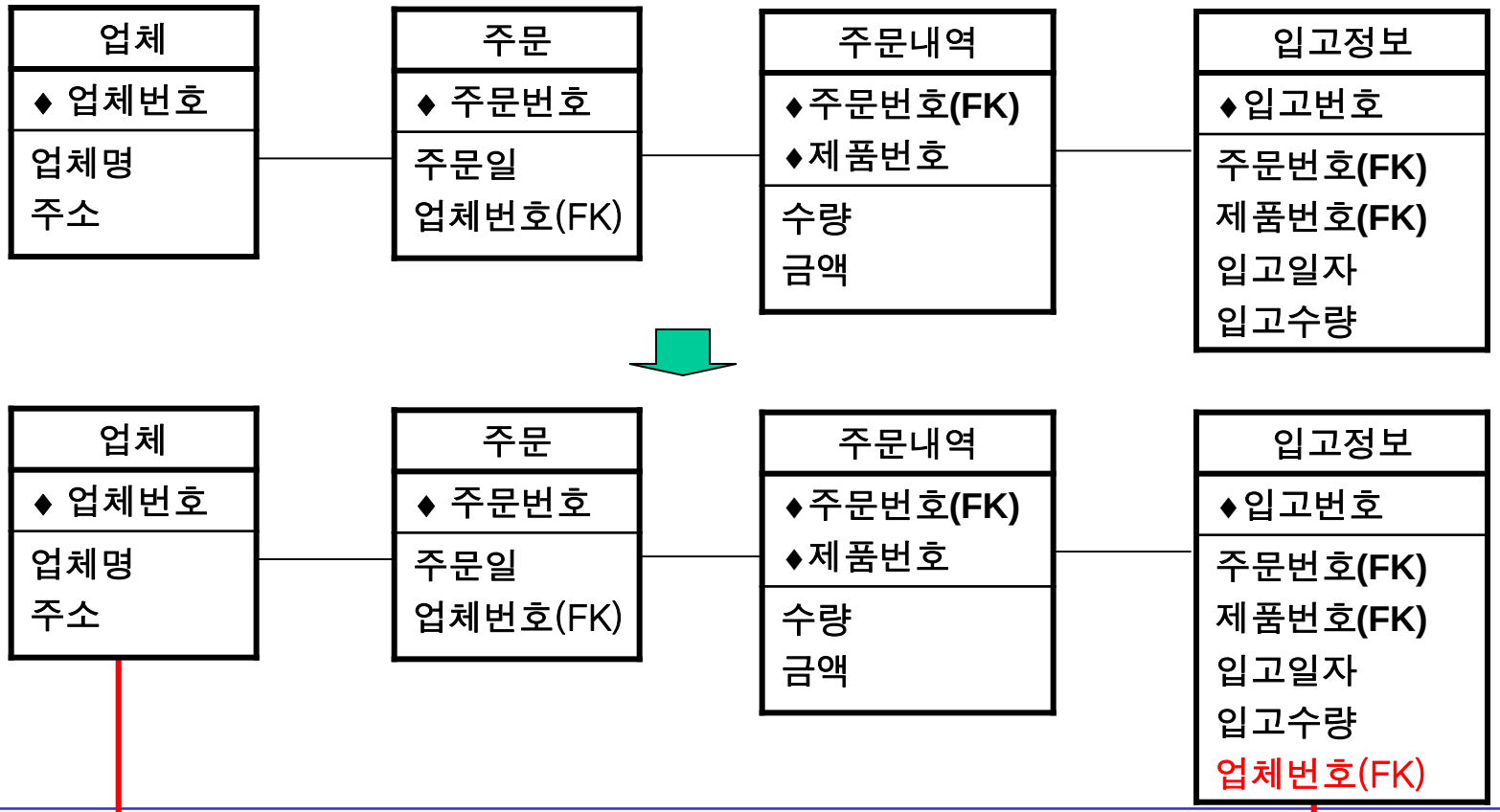


<그림 11.16> 속성의 중복에 의한 반정규화

11.3 반정규화

여러 엔티티를 조인해야 원하는 정보를 얻을 수 있는 경우에 적용

❑ 관계에 대한 반정규화



11.4 뷰(view)의 설계

❑ 뷰 설계 정의서의 예

- 뷰의 정의는 데이테베이스의 시스템 카탈로그에 저장
  - 뷰가 어떻게 정의되었는지를 알아 보려면 DBMS 에 따라 매우 번거로운 작업
- 뷰 정의서를 작성해 보관하는 것이 나중에 뷰를 관리하는데 도움이 된다.

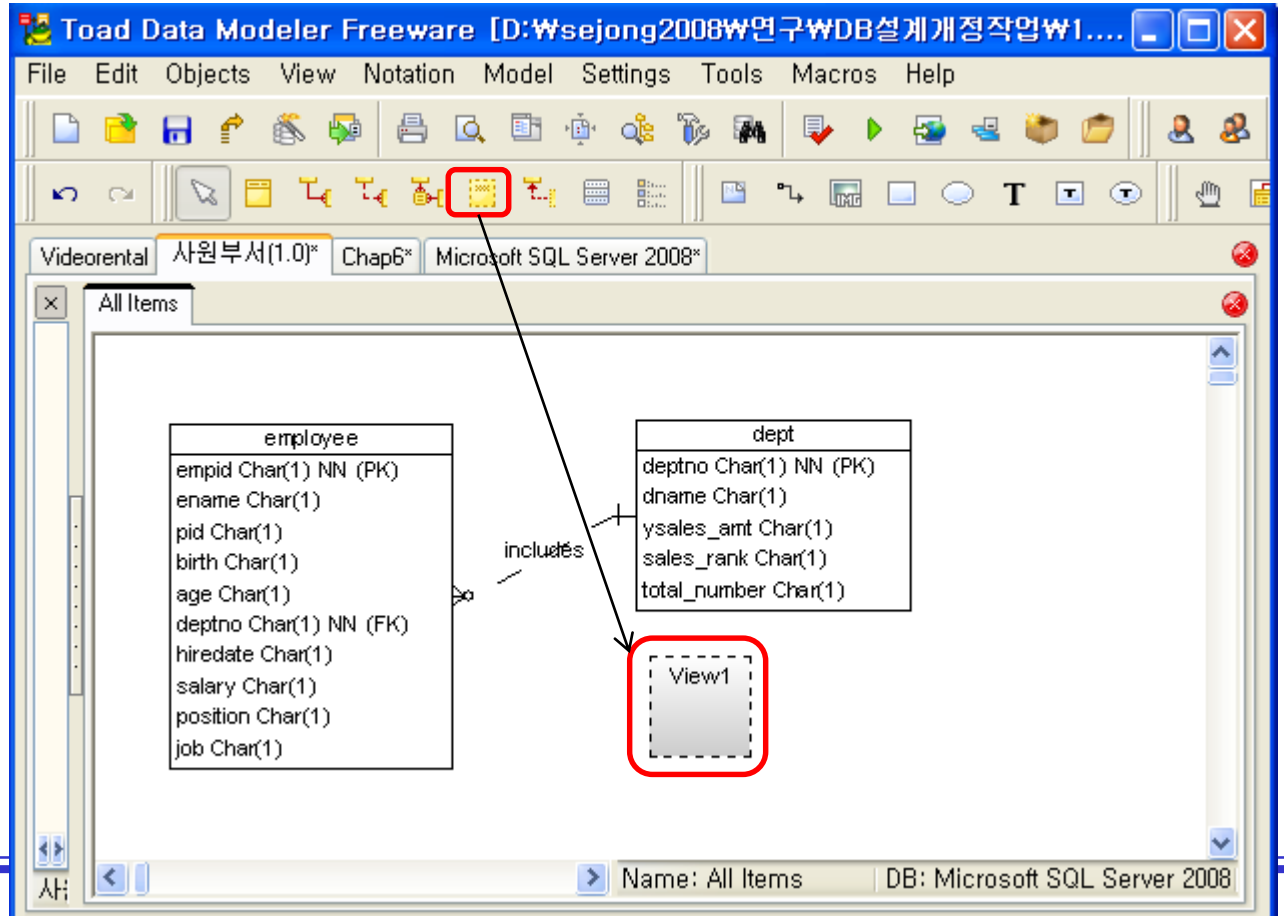
뷰명	뷰 설명	관련테이블	SQL
High_salary_emp	회계시스템과 인터페이스	emp	SELECT empno, ename, hiredate FROM emp WHERE sal > 3500

<그림 11.19> 뷰 정의서의 예

11.4 뷰(view)의 설계

❑ 모델링 도구에서 뷰의 입력

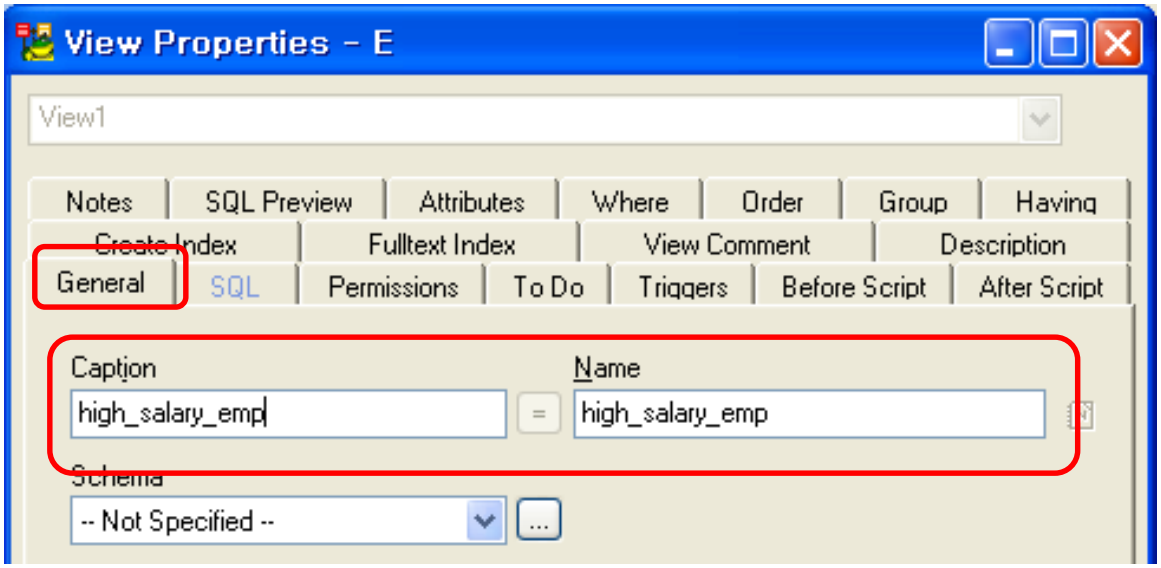
(1)



11.4 뷰(view)의 설계

❑ 모델링 도구에서 뷰의 입력

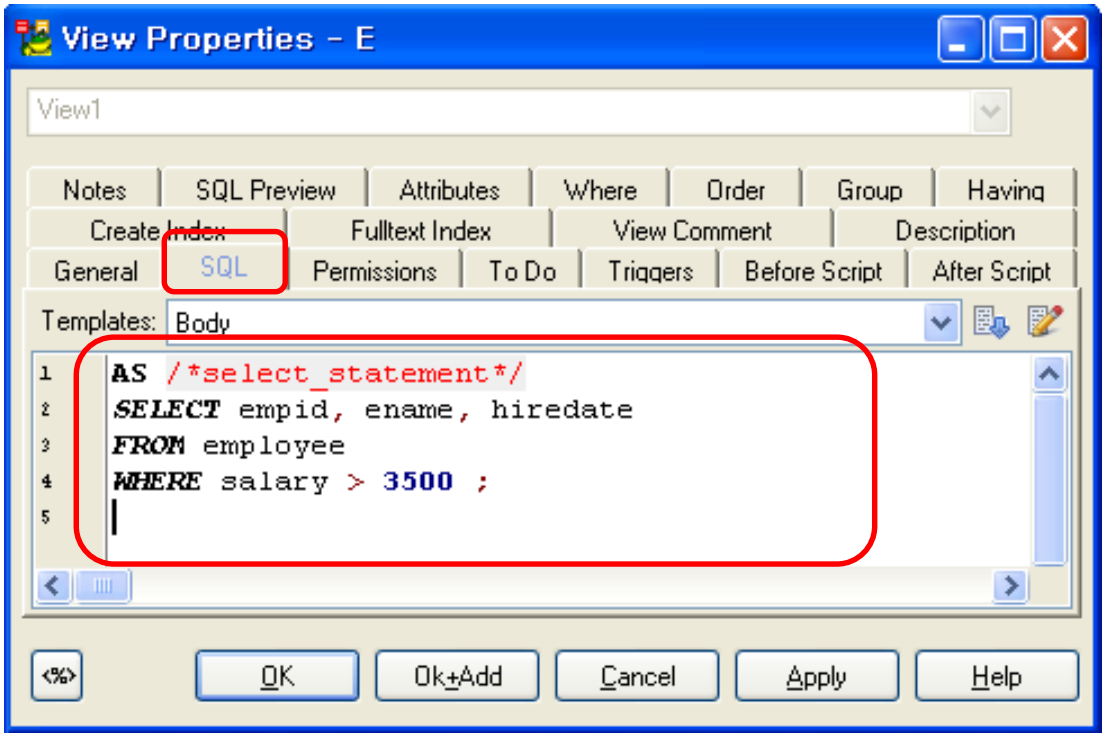
(2)



11.4 뷰(view)의 설계

❑ 모델링 도구에서 뷰의 입력

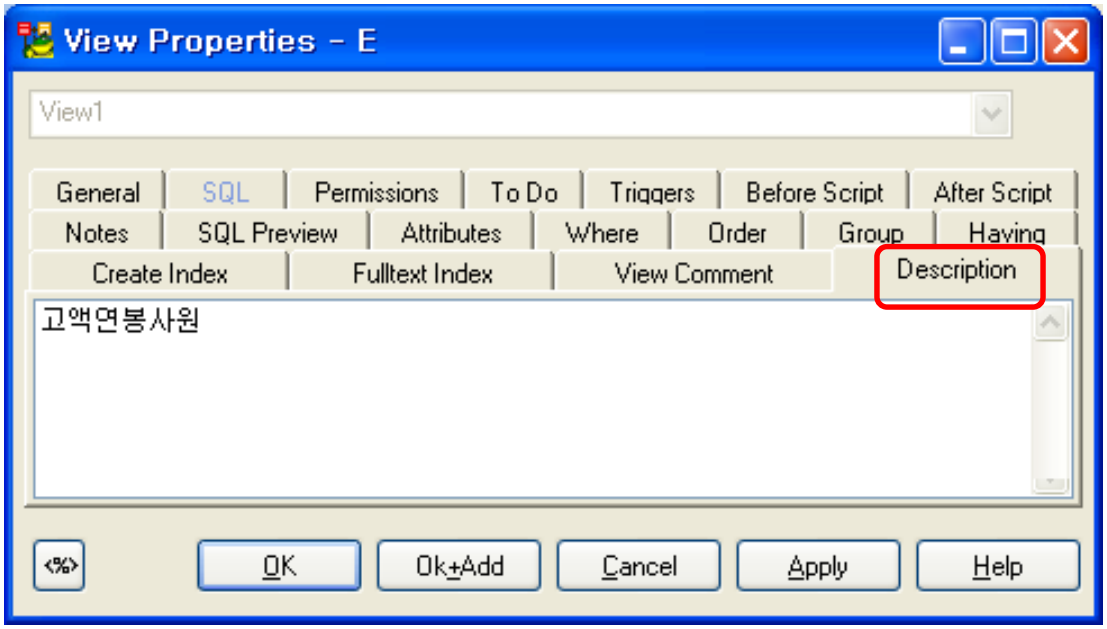
(3)



11.4 뷰(view)의 설계

❑ 모델링 도구에서 뷰의 입력

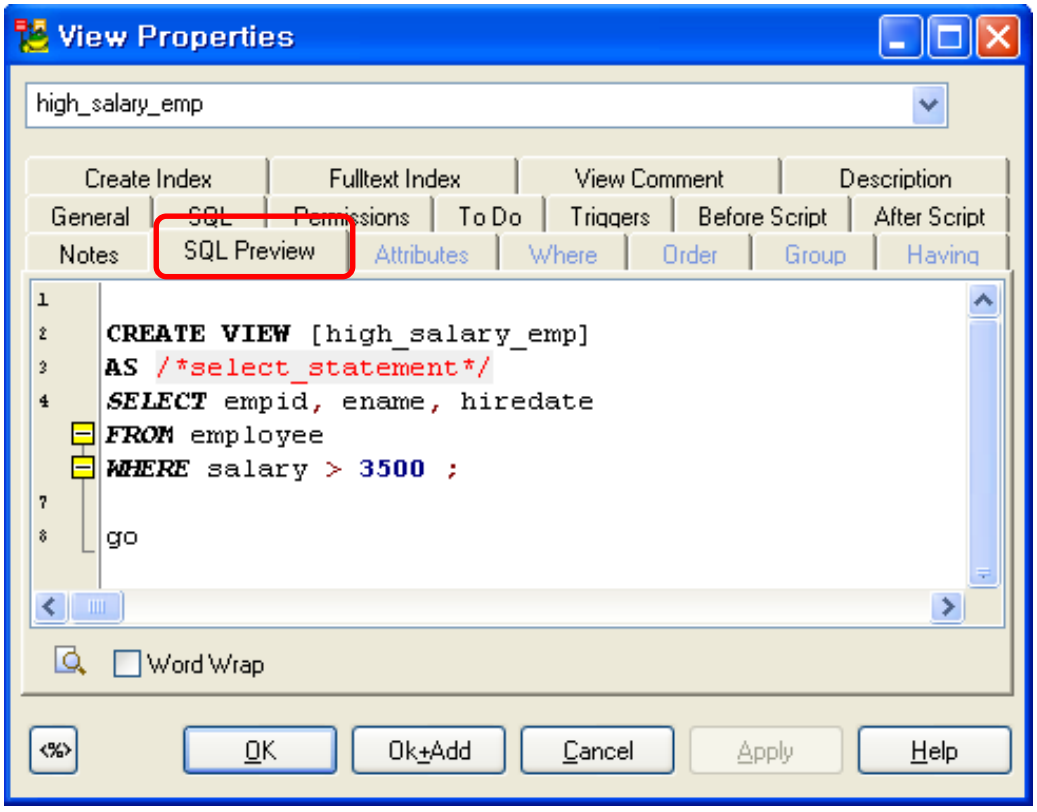
(4)



11.4 뷰(view)의 설계

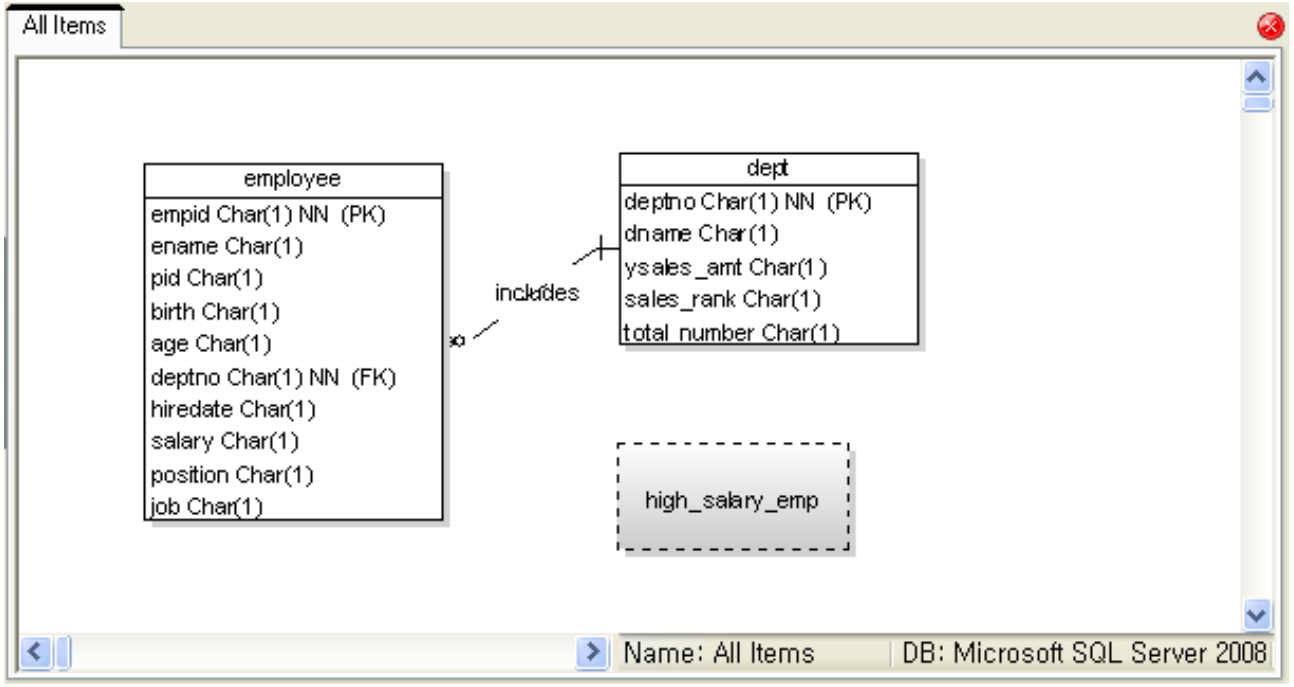
❑ 모델링 도구에서 뷰의 입력

(5)



11.5 인덱스의 설계

❑ 모델링 도구에서 뷰의 입력

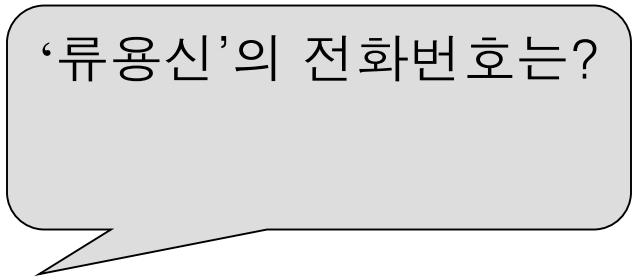


<생성된 뷰>

11.5 인덱스의 설계

- 인덱스(index)
  - 인덱스(index)는 테이블에 대한 검색 속도를 향상시킬 수 있는 확실한 수단
  - 책 뒤에 붙어 있는 색인과 비슷한 역할

고객번호	이름	성별	전화번호	취미
98001	김철수	M	111-2323	등산
98002	홍길동	M	731-4325	낚시
98003	김영희	F	456-1763	등산
98004	박순섭	F	345-4352	여행
98005	강고인	M	633-2156	낚시
98006	류용신	F	354-2323	여행
….	….	…	…	…



<그림 11.20> 고객정보의 예 (10만명 저장)

11.5 인덱스의 설계

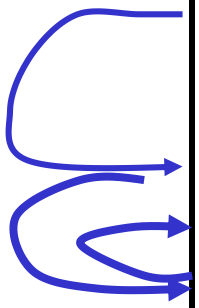
➢ 순차적 접근

- 순차 접근에 의한 검색은 튜플수가 많아지면 시간이 매우 오래 걸리기 때문에 데이터베이스와 같이 대량의 데이터에 대한 검색 방법으로는 적당하지 않다

고객번호	이름	성별	전화번호	취미
98001	김철수	M	111-2323	등산
98002	홍길동	M	731-4325	낚시
98003	김영희	F	456-1763	등산
98004	박순섭	F	345-4352	여행
98005	강고인	M	633-2156	낚시
98006	류용신	F	354-2323	여행
….	….	…	…	…

11.5 인덱스의 설계

- 이진 검색
  - 데이터가 정렬되어 있다면 적은 비교횟수로 원하는 튜플을 찾을 수 있다
  - 그러나 10만개의 튜플에 대해 정렬 상태를 유지하는 것은 매우 많은 비용이 든다

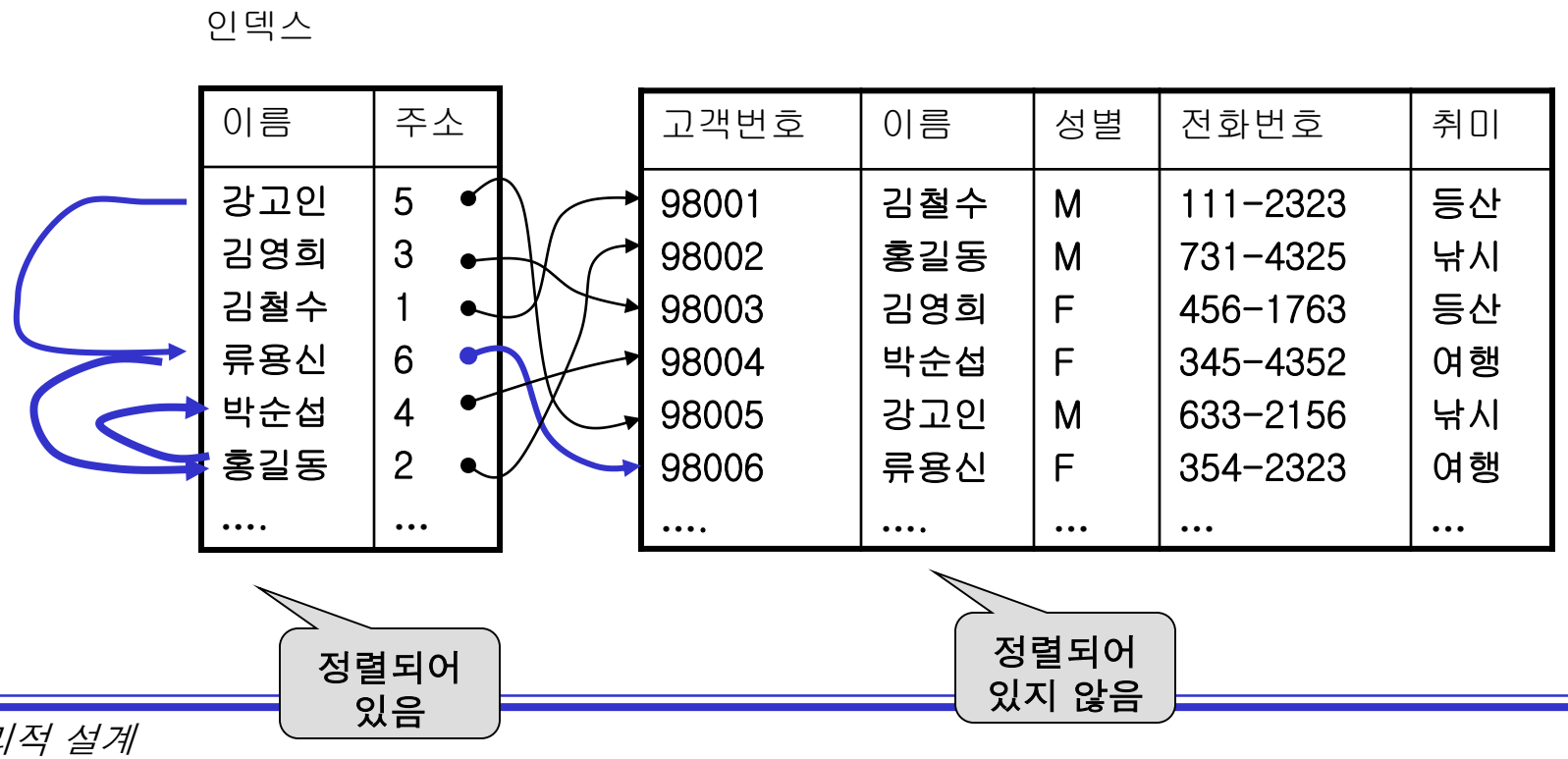


고객번호	이름	성별	전화번호	취미
98005	강고인	M	633-2156	낚시
98003	김영희	F	456-1763	등산
98001	김철수	M	111-2323	등산
98006	류용신	F	354-2323	여행
98004	박순섭	F	345-4352	여행
98002	홍길동	M	731-4325	낚시
....	....	...	...	...

11.5 인덱스의 설계

➢ 인덱스로 검색

- 실제 데이터는 정렬되어 있지 않지만 마치 정렬되어 있는 것과 같은 효과를 얻을 수 있다



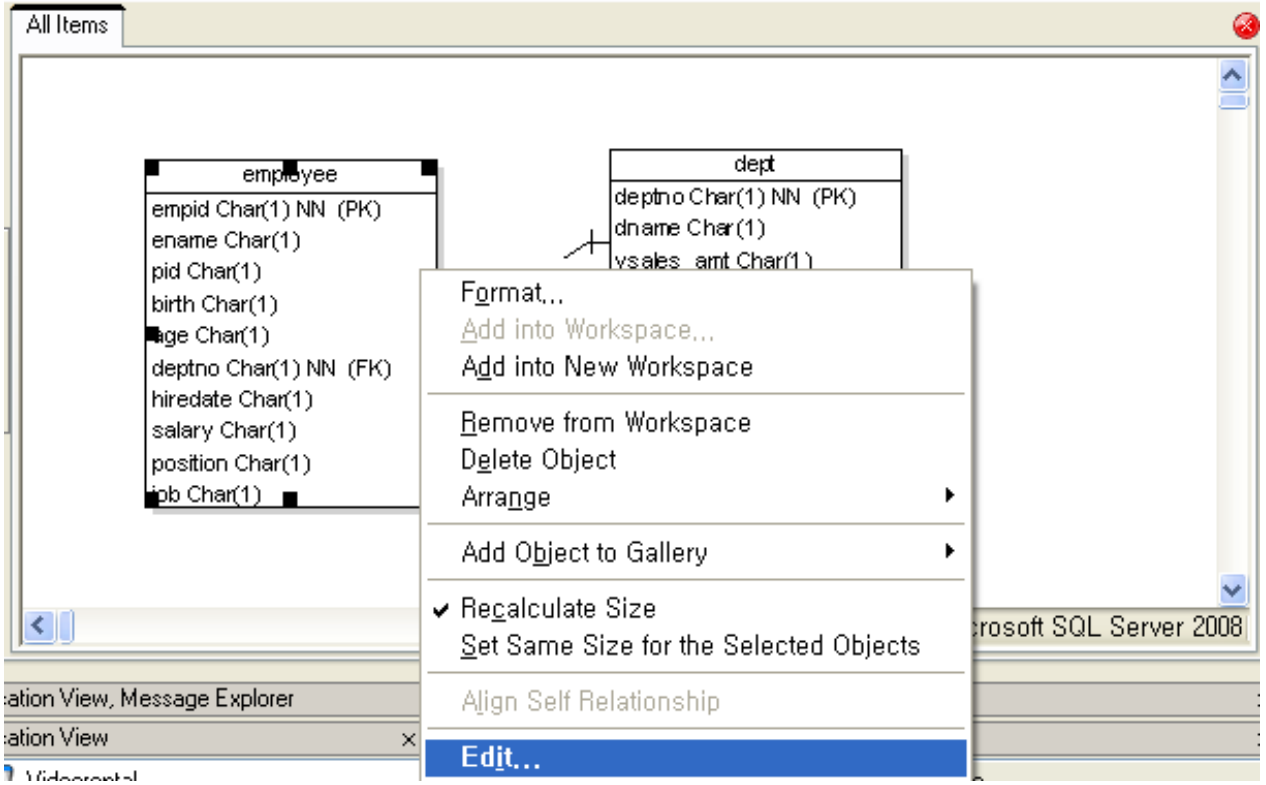
11.5 인덱스의 설계

- 인덱스에 대한 검토
  - 인덱스의 수가 많으면 인덱스를 재정렬하는데 많은 시간이 소모되므로 DBMS의 성능을 저하 시킨다.
  - 그러므로 인덱스는 꼭 필요한 컬럼에 대해서만 지정을 해야 한다
  - 모든 경우에 대해 인덱스가 성능을 발휘하는 것은 아니다
    - 인덱스로 지정하는 컬럼은 SQL의 WHERE 절에서 비교 대상이 되는 컬럼 또는 JOIN에 사용되는 컬럼이어야 한다.
    - 튜플의 수가 적으면 (예:200~300개) 인덱스를 지정하여도 별 효과가 없다.
    - 인덱스로 지정한 컬럼에 의해 검색했을 때 검색 결과가 전체 튜플의 10~15% 미만일 때 인덱스의 효과가 있다.

11.5 인덱스의 설계

❑ 모델링 도구에서 인덱스의 입력

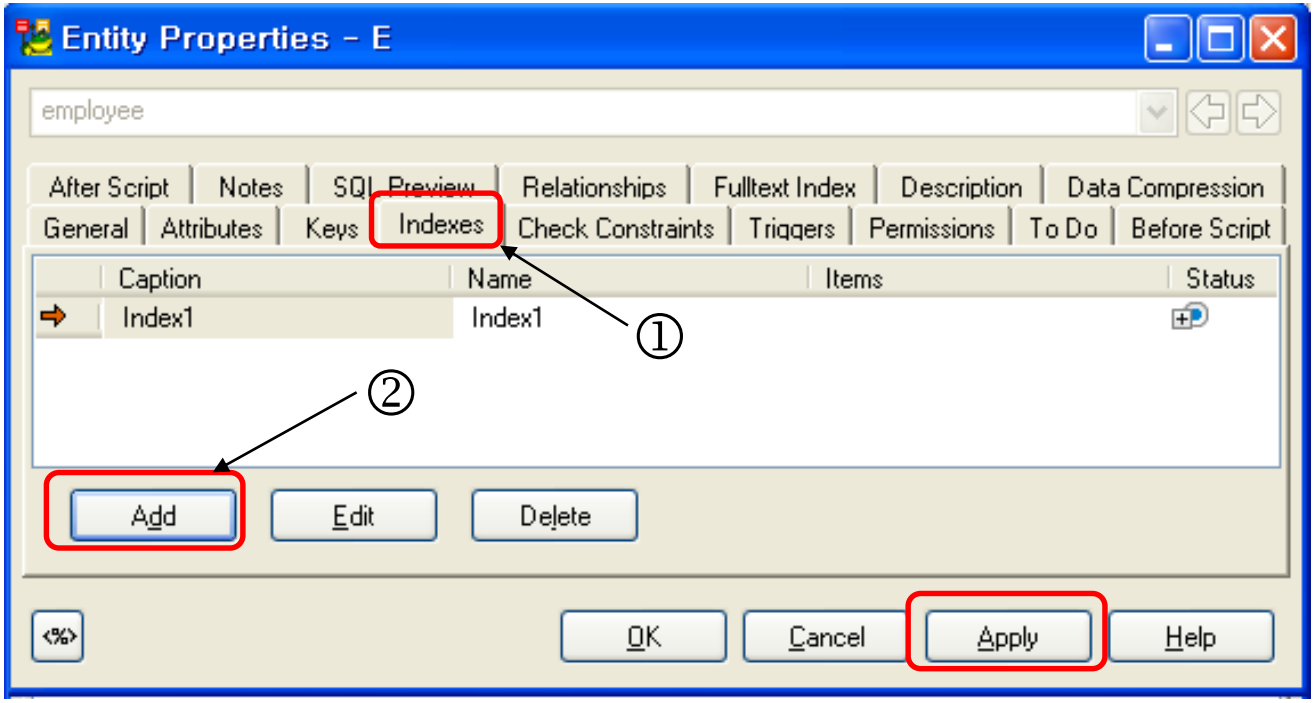
(1)



11.5 인덱스의 설계

❑ 모델링 도구에서 인덱스의 입력

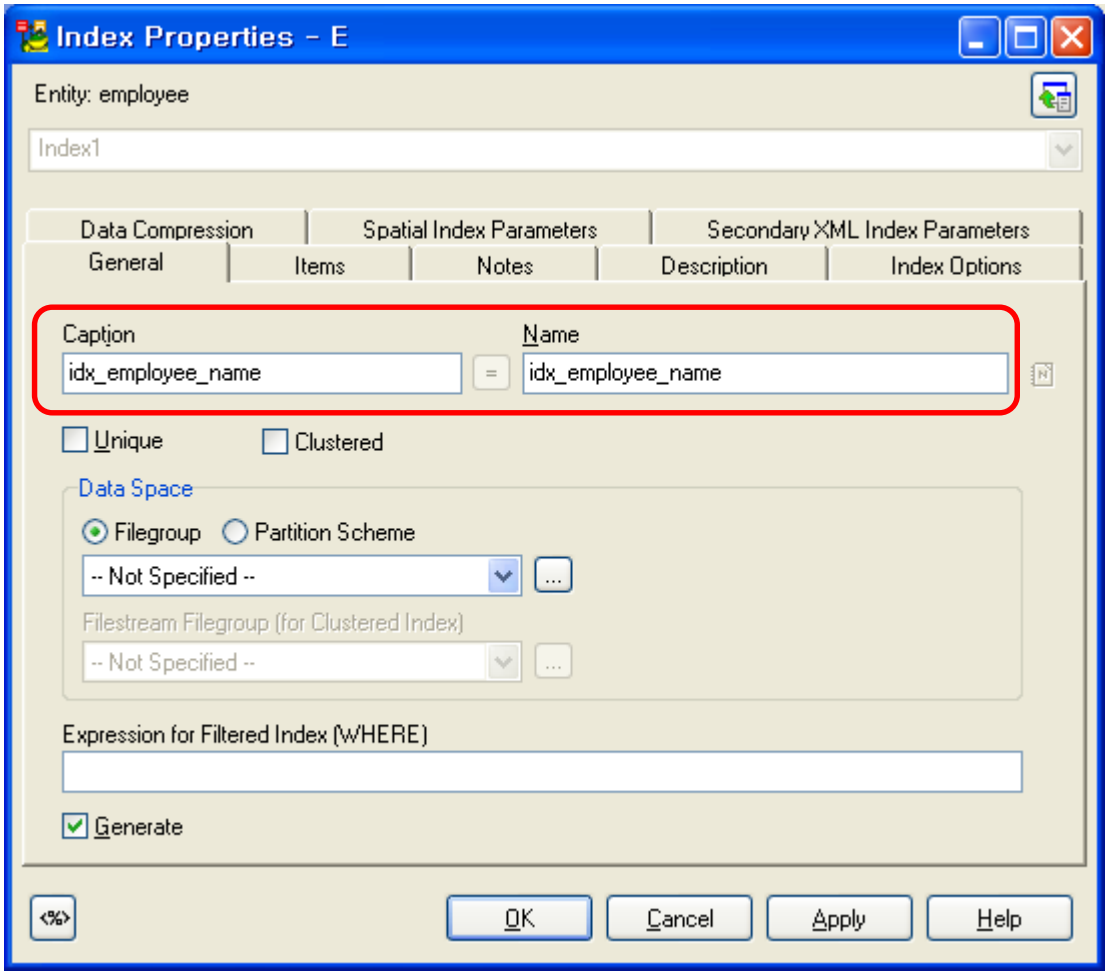
(2)



11.5 인덱스의 설계

❑ 모델링 도구에서 인덱스의 입력

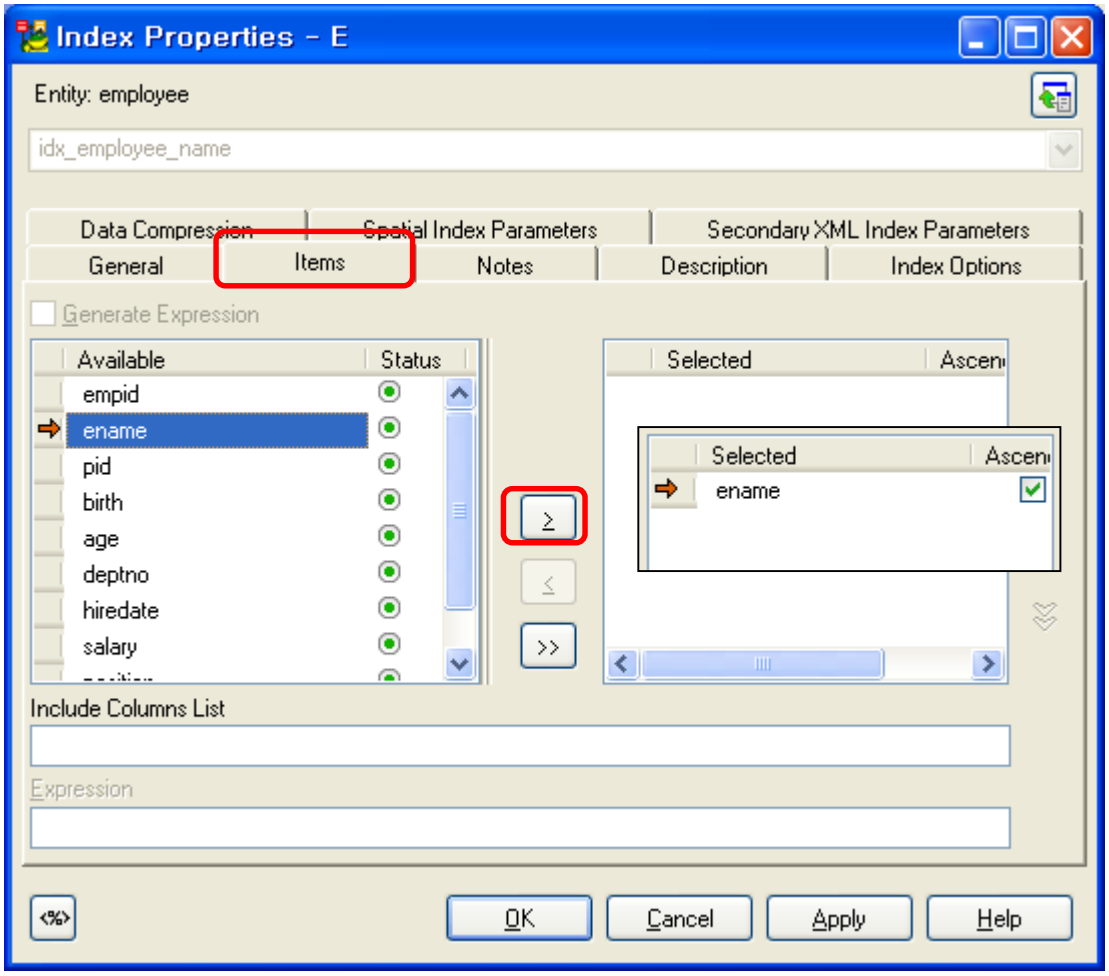
(3)



11.5 인덱스의 설계

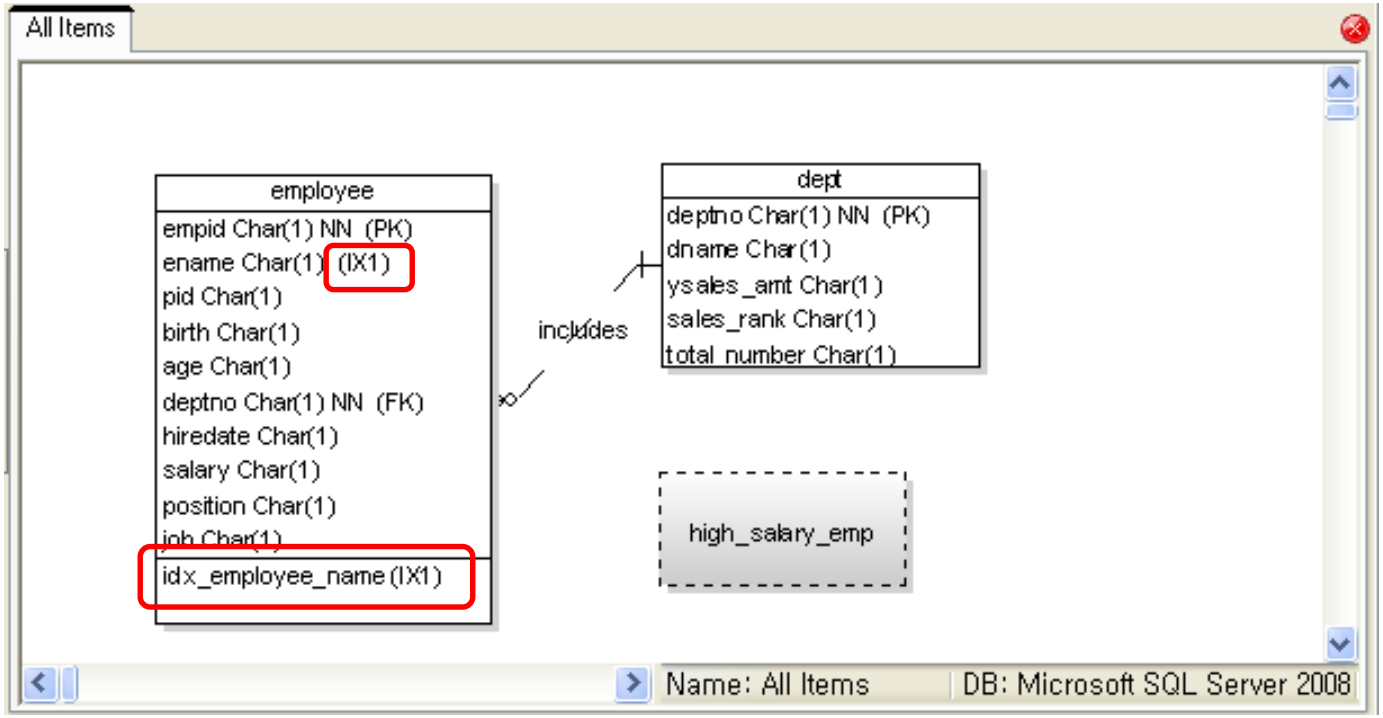
❑ 모델링 도구에서 인덱스의 입력

(4)



11.5 인덱스의 설계

❑ 모델링 도구에서 인덱스의 입력



<생성된 인덱스>

11.6 테이블 기술서의 작성

- 테이블 기술서
  - 개별 테이블에 대한 보다 자세한 문서화 수단
  - 모델링 도구에 테이블에 대한 정보가 저장되어 있지만 여러 가지 이유로 테이블 하나하나에 대한 출력된 문서를 필요로 한다

  - 테이블 기술서에는 하나의 테이블에 대한 모든 정보가 상세히 기술되어 있어서 데이터베이스를 기반으로 응용 프로그램을 작성하는 개발자나 유지보수 담당자에게 매우 유용하다.

Name	Orders	Table 기술서	작성일	2004. 11. 23	page
System	컴퓨터부품관리		작성자	한 소 연	/
Description	주문 정보를 가지고 있는 테이블				

NO	Column name	Data Type	NN	KY	Defult	Descirption
1	order_no	integer	✔	(PK)		주문 일련번호
2	supplier_sup_no	integer	✔	(FK)		공급회사의 일련번호
3	send_date	date				주문제품을 받는날
4	total_money	integer				주문된 제품의 총 금액
5	order_date	date				주문한 날짜
6	end_date	date				납품 완료일
7	status	char				상태정보
8						
9						
10						
11						

비고

* status : U-변경, C-취소, X-납품완료
* 인덱스지정 : supplier_sup_no (desc)
* FK(supplier_sup_no) → prod_company(supplier_no)

(NN : not null. 선택시 null을 허용하지 않음

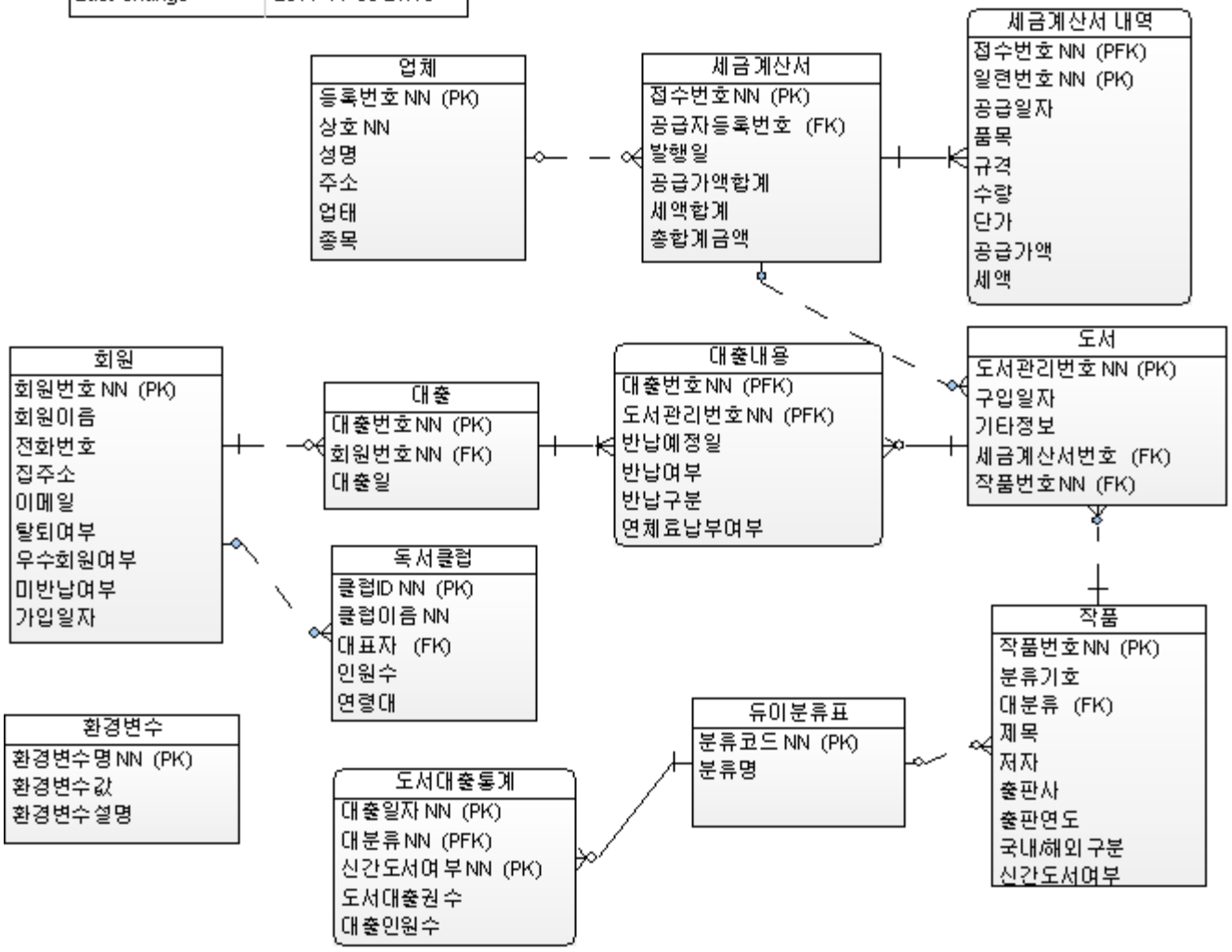
KY: key. 기본키는 PK, 외래키는 FK, 기본키이면서 외래키인 경우는 PFK)

<그림 11.22> 테이블 기술서의 예

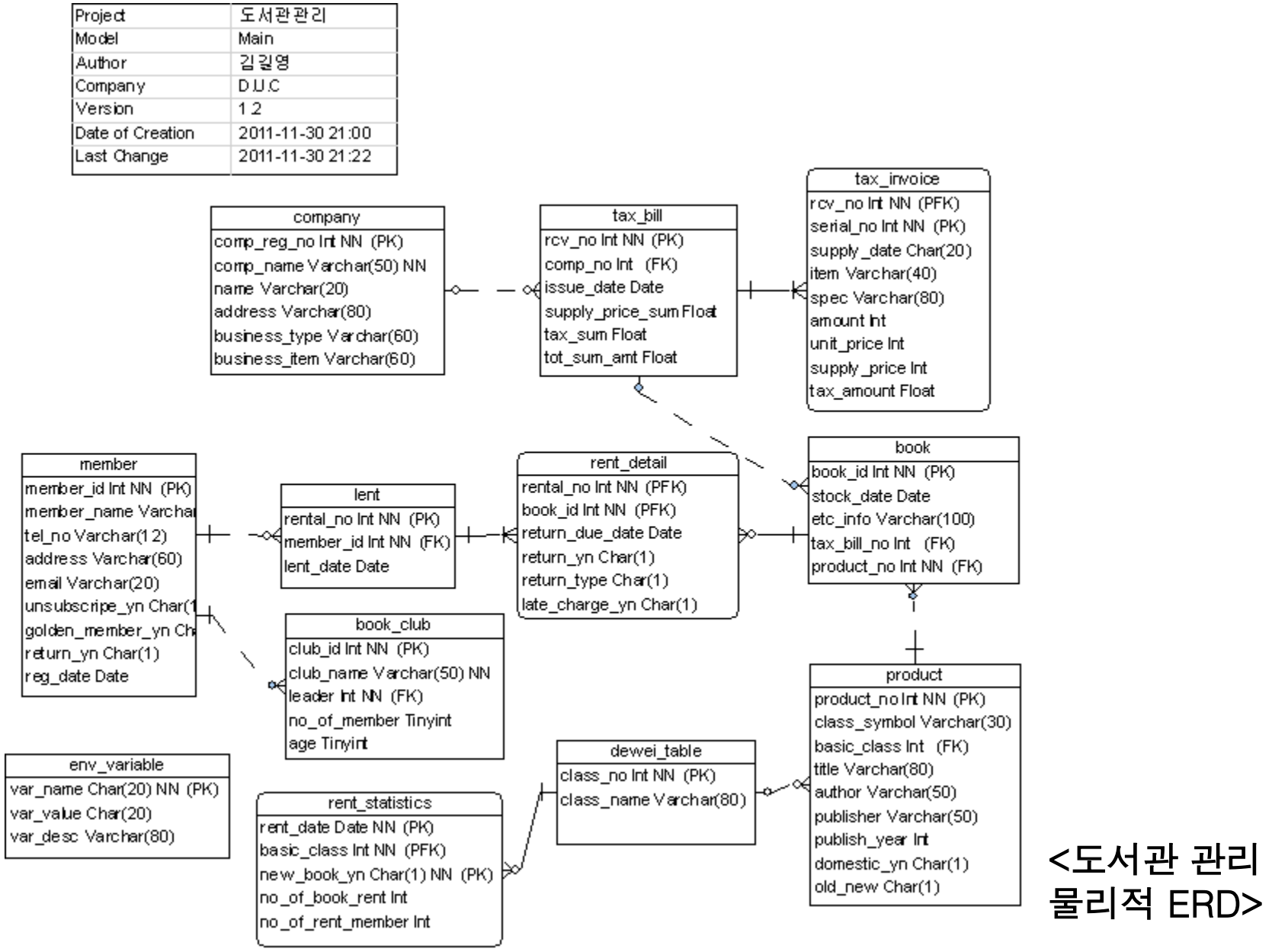
11.7 물리적 설계의 예

- 도서관 관리
  - *다음 slide 참조*

Project	도서관 관리
Model	Main
Author	김길영
Company	D.U.C
Version	1.2
Date of Creation	2011-11-30 21:00
Last Change	2011-11-30 21:15



<도서관 관리
논리적 ERD>



<도서관 관리 물리적 ERD>