

DB프로그래밍

14주차

담당교수: 김희숙
(jasmin11@hanmail.net)

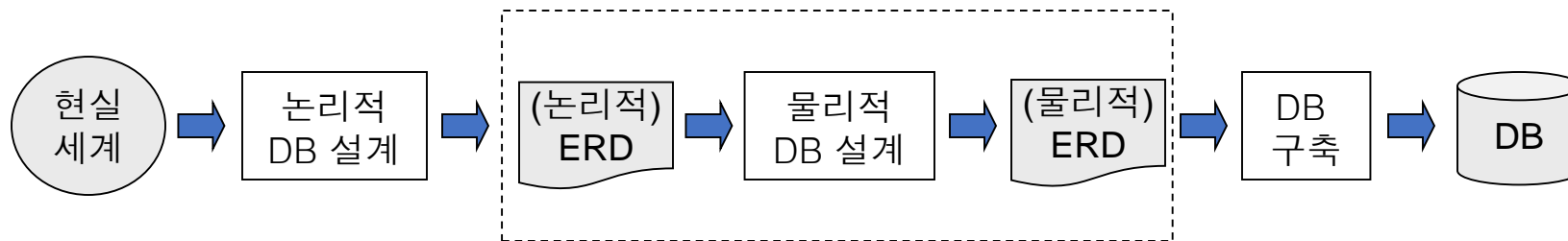
물리적 설계

14주차 01-01

담당교수: 김희숙
(jasmin11@hanmail.net)

11.1 개요

- 물리적 데이터베이스 설계
 - 엔티티, 관계, 속성 등의 요소 → 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

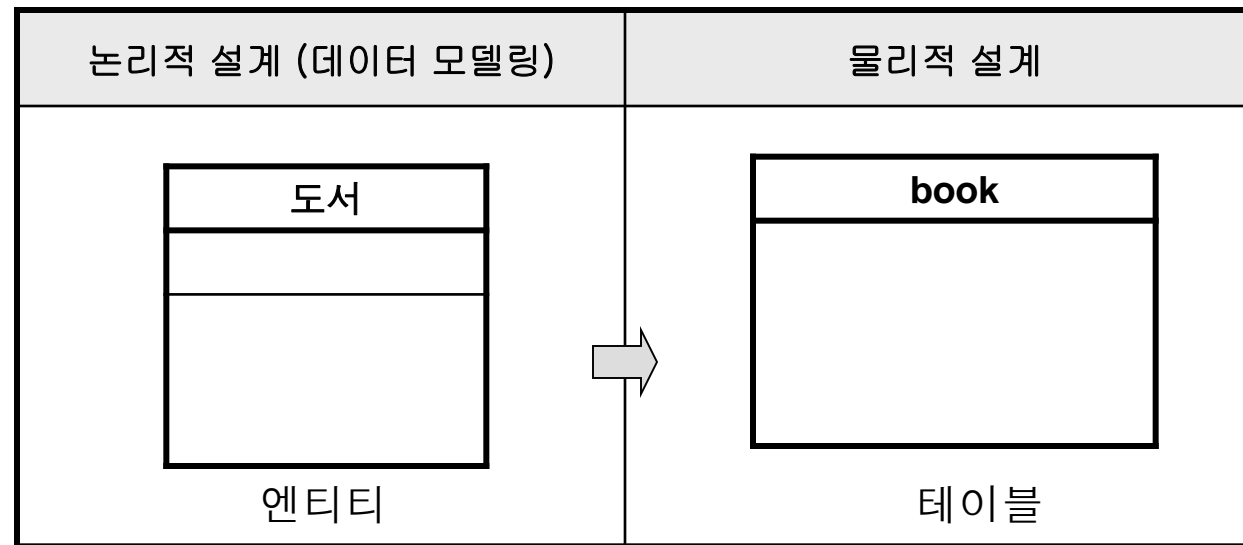
- 논리적 DB 설계 → 물리적 DB 설계
 - 물리적 데이터베이스의 기본적인 내용은 논리적 설계의 산출물인 ERD의 요소들을 관계형 데이터베이스의 요소들로 전환하는 것

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 카디널리티	→ 관계의 카디널리티	→ -
관계의 참여도	→ 관계의 참여도	→ -

<그림 11.3> ERD 요소들의 전환

11.2 테이블, 컬럼, 키로의 변환

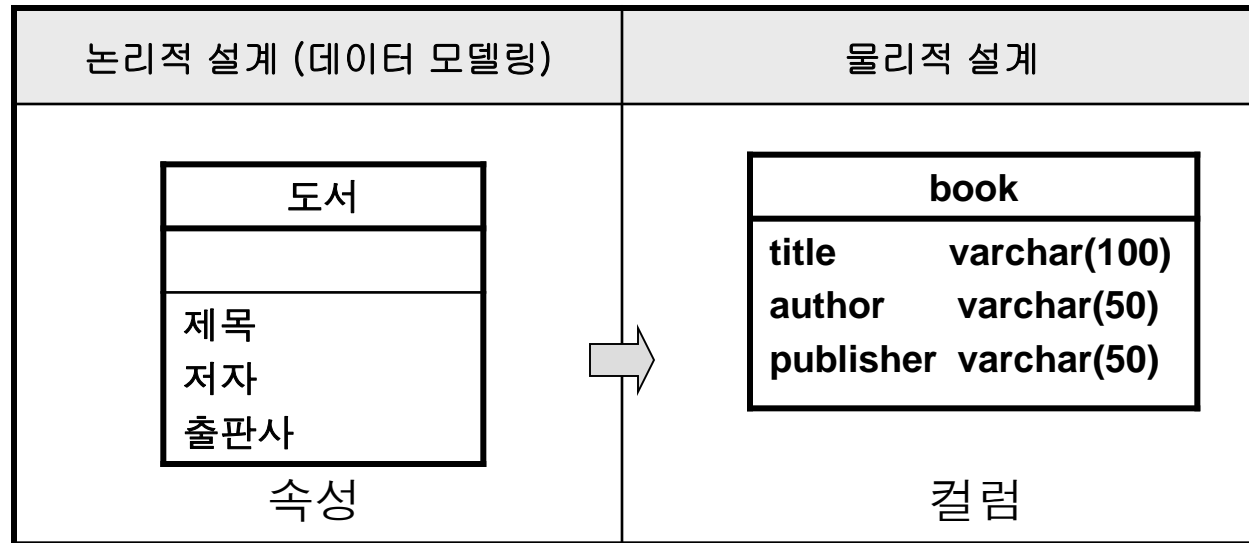
- 엔티티 → 테이블
 - 엔티티는 그대로 테이블로 변환됨
 - 한글 엔티티명 → 영문 테이블명



<그림 11.4> 엔티티 → 테이블 전환

11.2 테이블, 컬럼, 키로의 변환

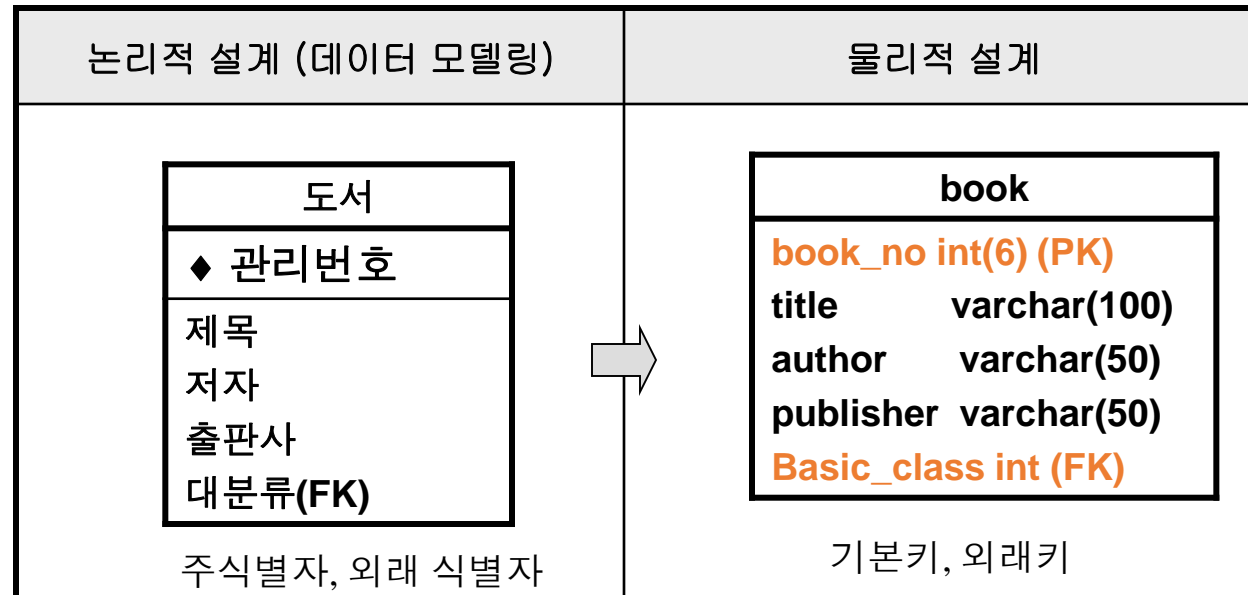
- 속성 → 컬럼
 - 한글 속성명 → 영문 컬럼명 (용어사전 이용)
 - 데이터 타입의 지정



<그림 11.5> 속성 → 컬럼 전환

11.2 테이블, 컬럼, 키로의 변환

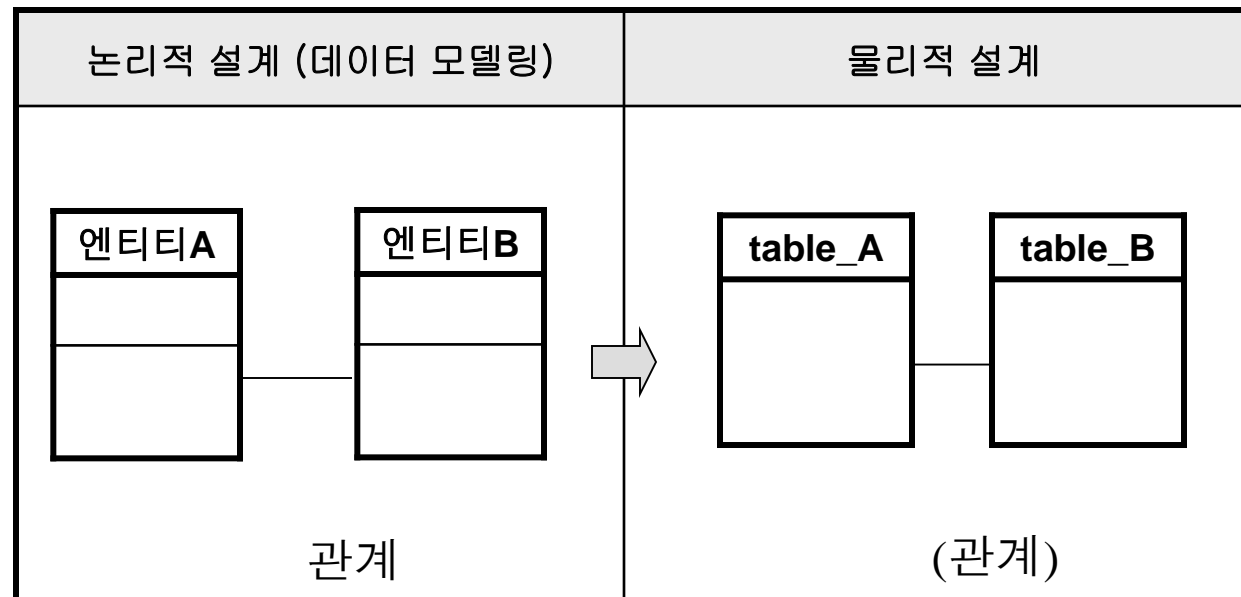
- 주식별자 → 기본키, 외래식별자 → 외래키



<그림 11.6> 주식별자, 외래 식별자 → 기본키, 외래키 전환

11.2 테이블, 컬럼, 키로의 변환

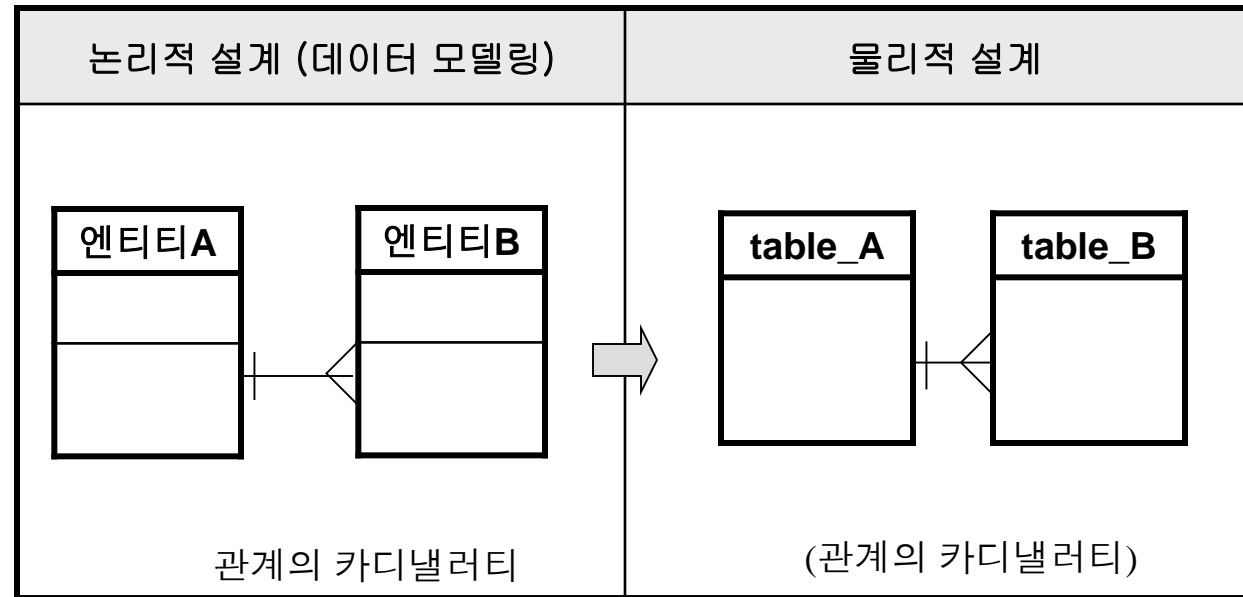
- 관계의 전환
 - 논리적 설계에서 엔티티간의 관계(relationship)는 물리적 설계에서도 그대로 유지



<그림 11.7> 관계의 전환

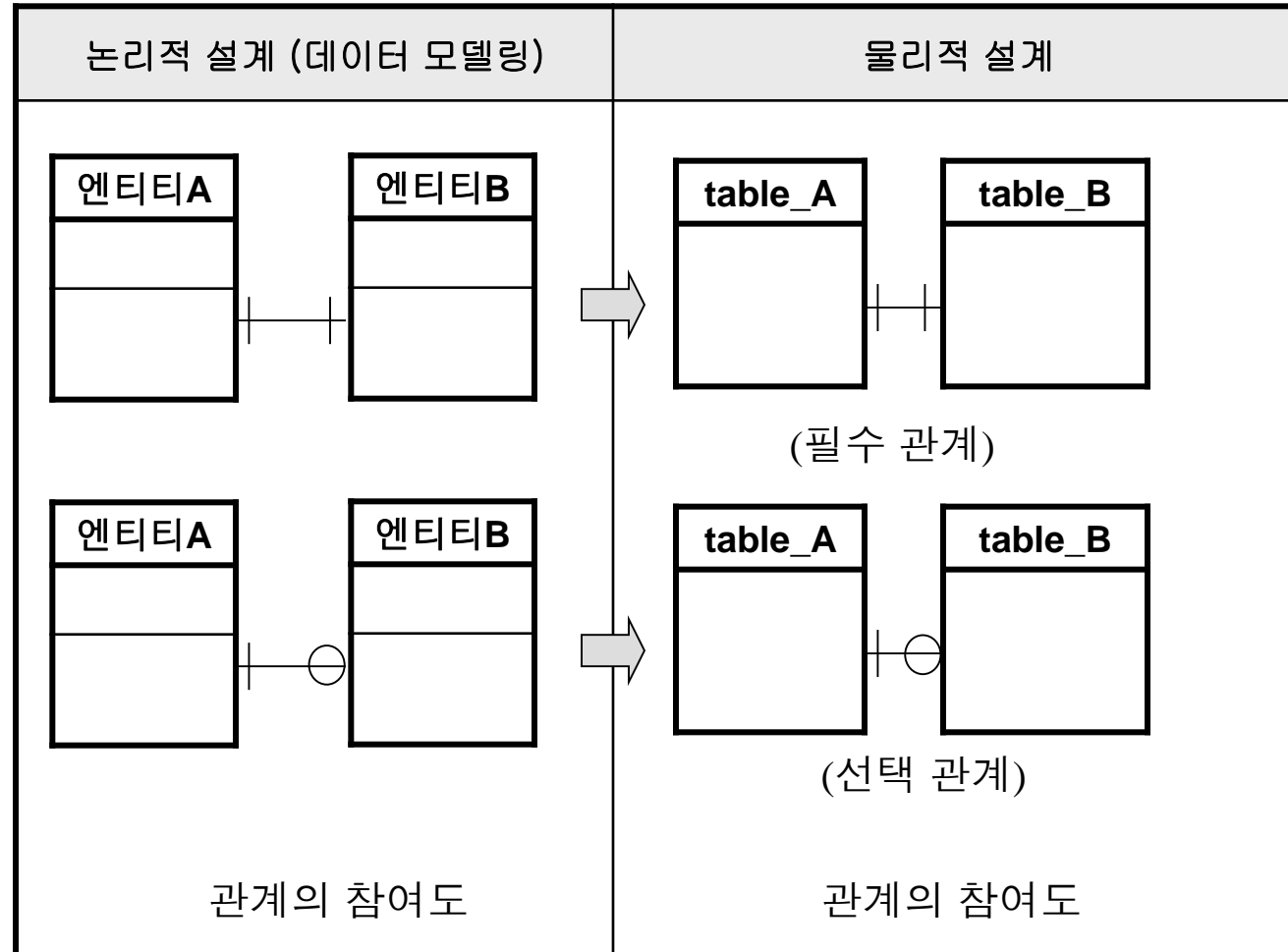
11.2 테이블, 컬럼, 키로의 변환

- 관계의 전환
 - 카디널리티, 선택/필수도 그대로 적용됨



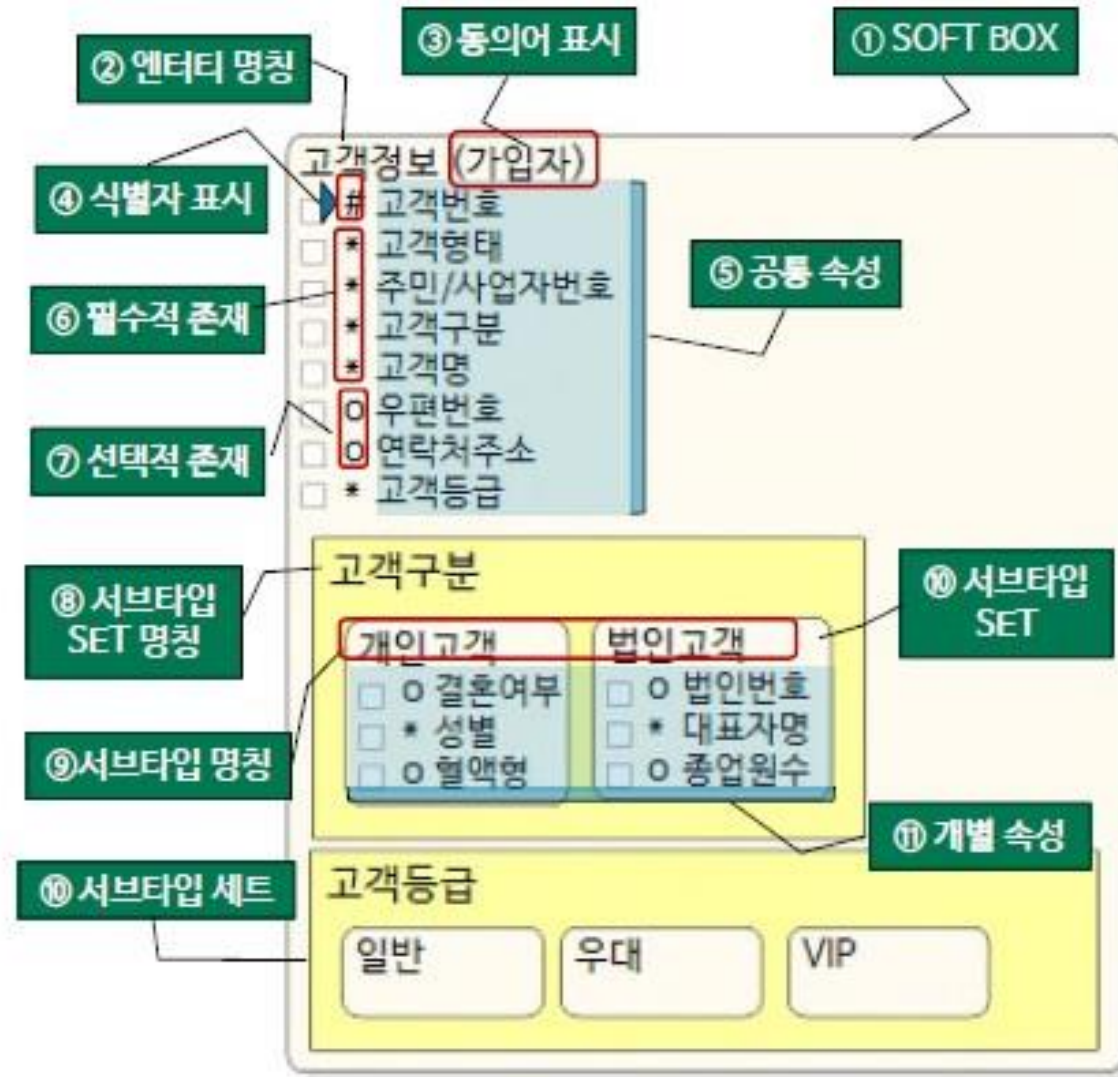
<그림 11.8> 관계의 카디널리티 전환

11.2 테이블, 컬럼, 키로의 변환

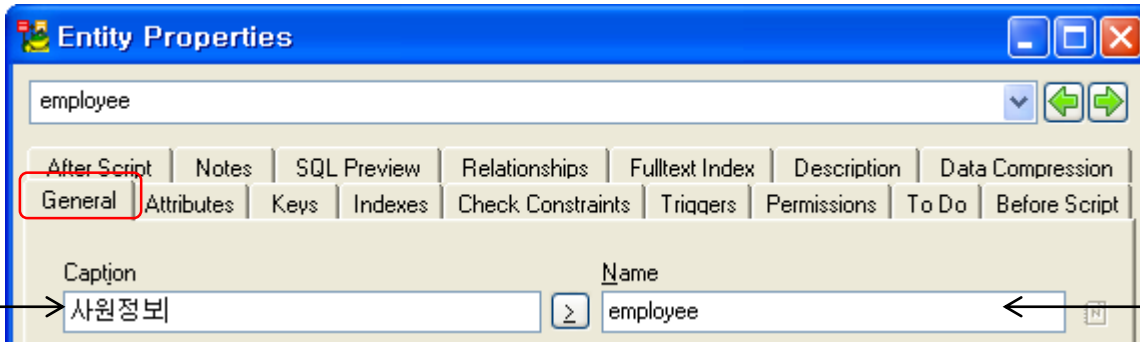


<그림 11.9> 관계의 참여도 전환

[참고] 모델링 툴: DA#

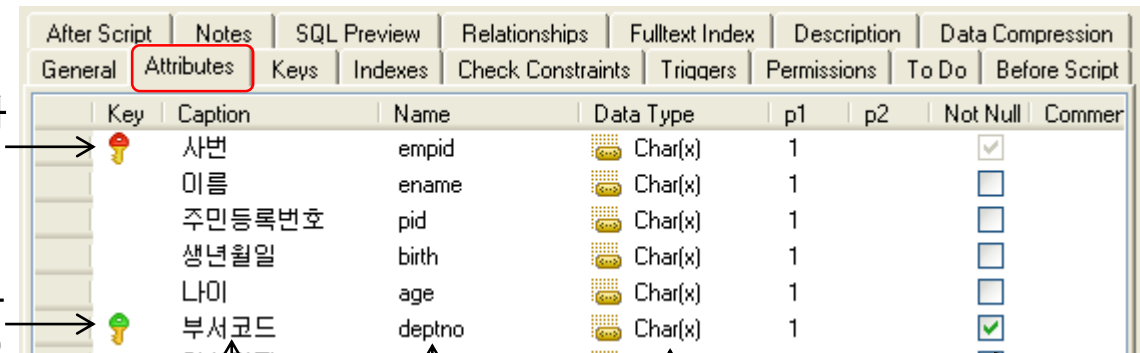


[참고] 모델링 툴: Case Studio



Entity Properties dialog box showing the General tab. The entity name is 'employee' and the table name is 'employee'.

엔티티 이름 → 사원정보 ← 테이블 이름

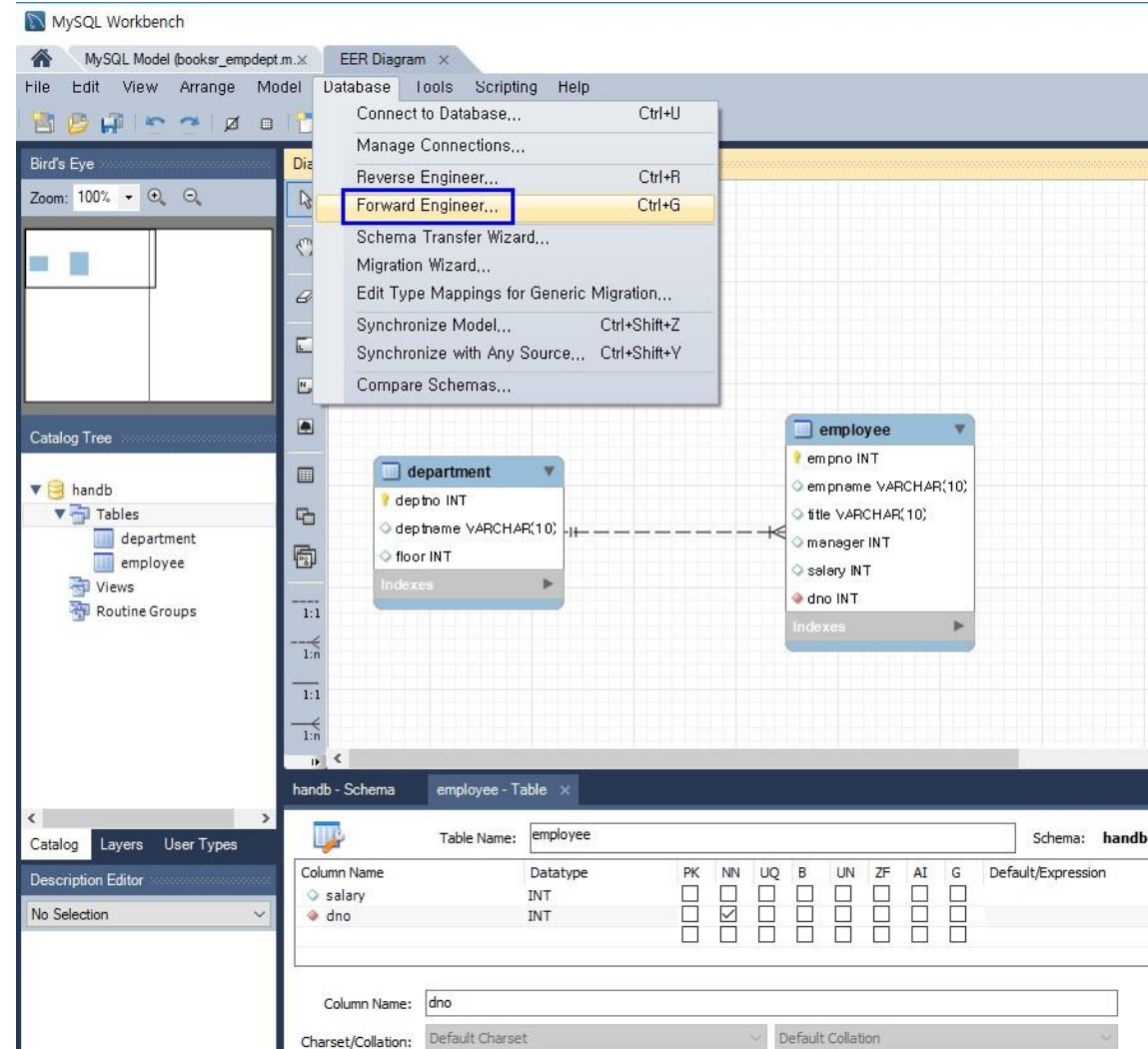
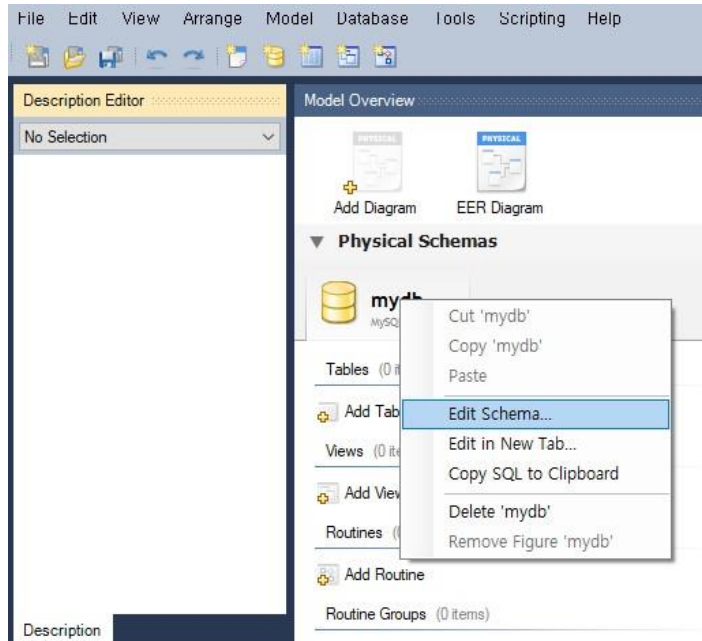


Entity Properties dialog box showing the Attributes tab. The table lists attributes with their properties.

Key	Caption	Name	Data Type	p1	p2	Not Null	Commer
주식별자 (기본키)	사번	empid	Char(x)	1		<input checked="" type="checkbox"/>	
	이름	ename	Char(x)	1		<input type="checkbox"/>	
	주민등록번호	pid	Char(x)	1		<input type="checkbox"/>	
	생년월일	birth	Char(x)	1		<input type="checkbox"/>	
	나이	age	Char(x)	1		<input type="checkbox"/>	
외래식별자 (왜래키)	부서코드	deptno	Char(x)	1		<input checked="" type="checkbox"/>	

속성 이름 컬럼 이름 컬럼의 데이터 타입 데이터의 길이 NOT NULL의 지정

[참고] 모델링 툴: MySQL(Diagram)



[요약] 뷰 설계 정의서

❖ 뷰 설계 정의서:

- 뷰 정의는 데이터베이스 시스템 카탈로그에 저장
- 뷰가 어떻게 정의되었는지를 알아 보려면 DBMS에 따라 번거로운 작업이 될 수 있다
- 뷰 정의서를 작성해 보관하는 것이 나중에 뷰를 관리하는데 도움 된다

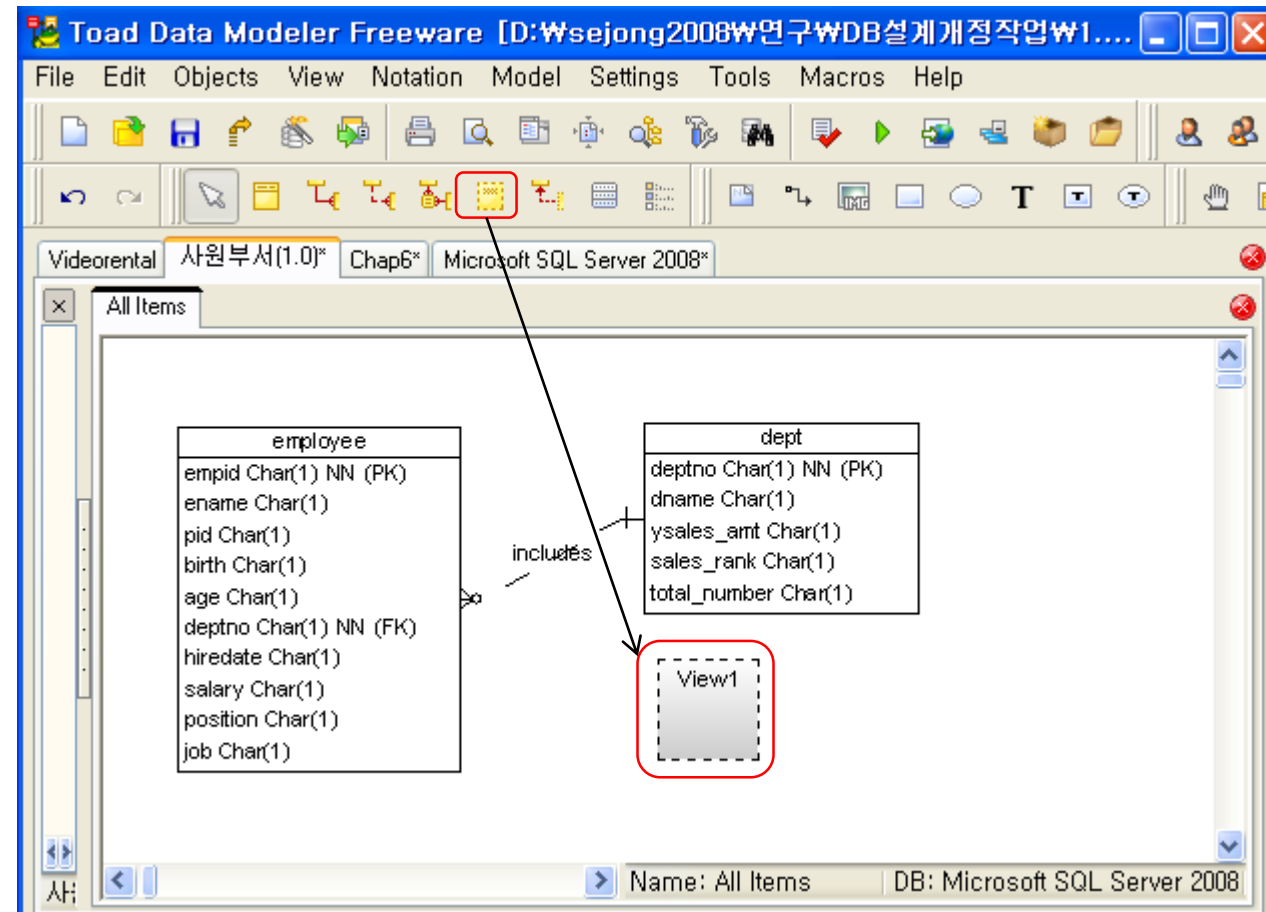
뷰 설계 정의서

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

11.4 뷰(view)의 설계

- 모델링 도구에서 뷰의 입력

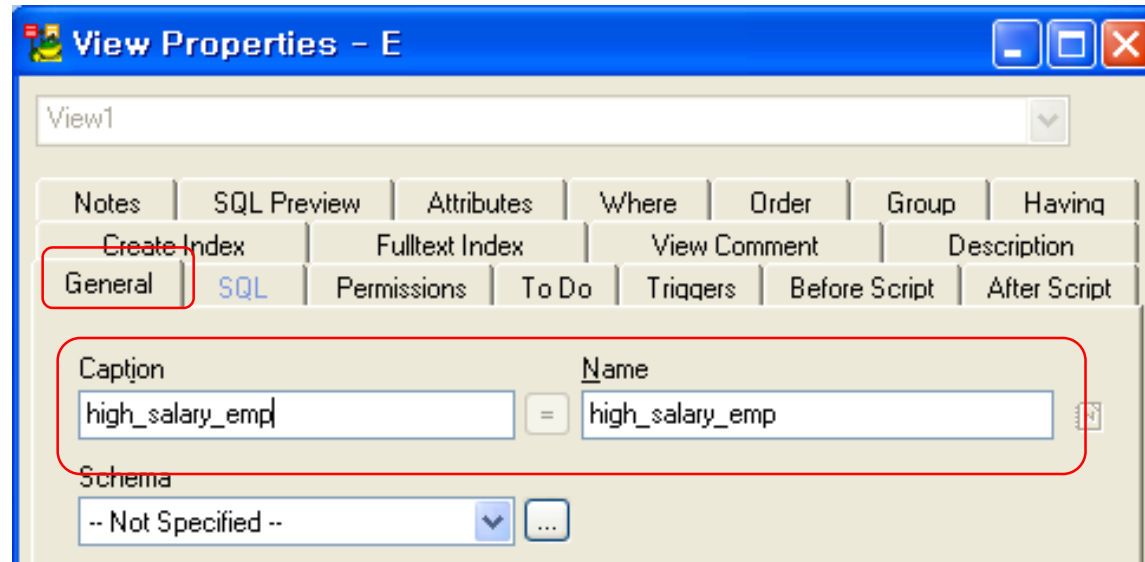
(1)



11.4 뷰(view)의 설계

- 모델링 도구에서 뷰의 입력

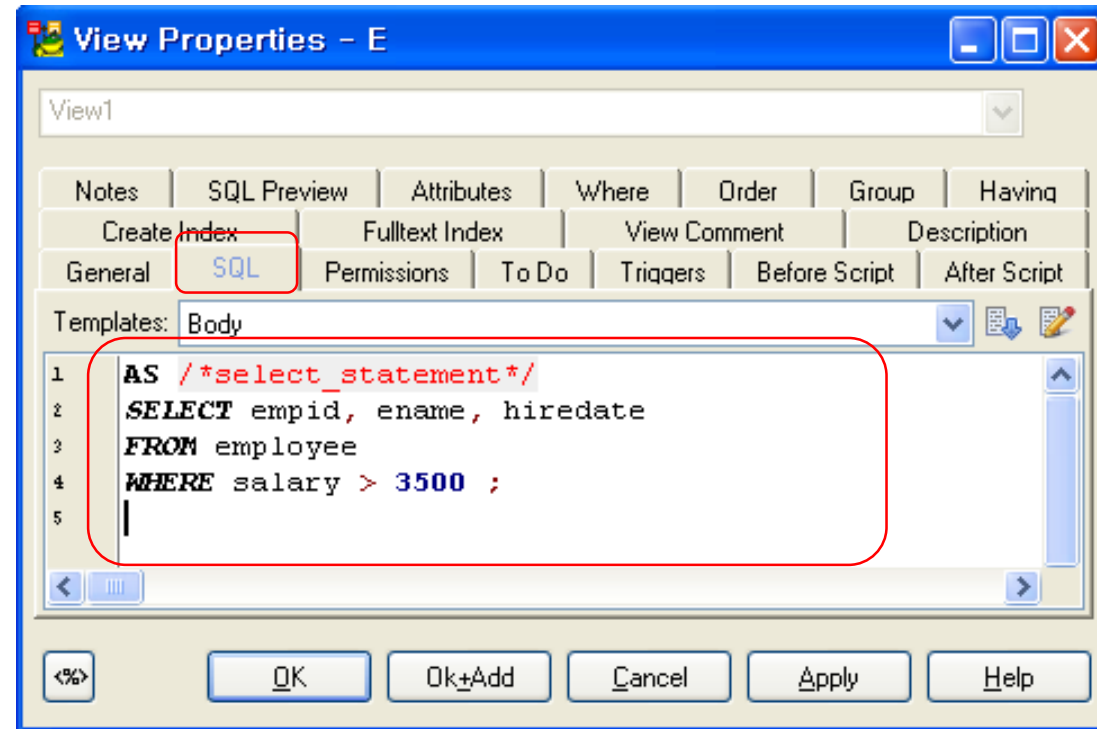
(2)



11.4 뷰(view)의 설계

- 모델링 도구에서 뷰의 입력

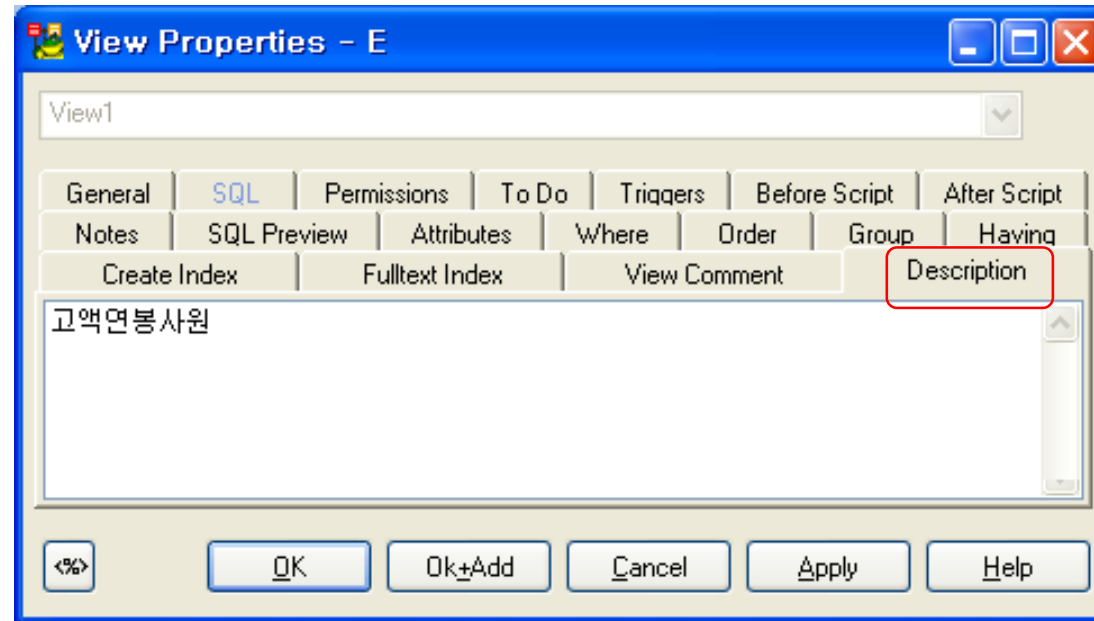
(3)



11.4 뷰(view)의 설계

- 모델링 도구에서 뷰의 입력

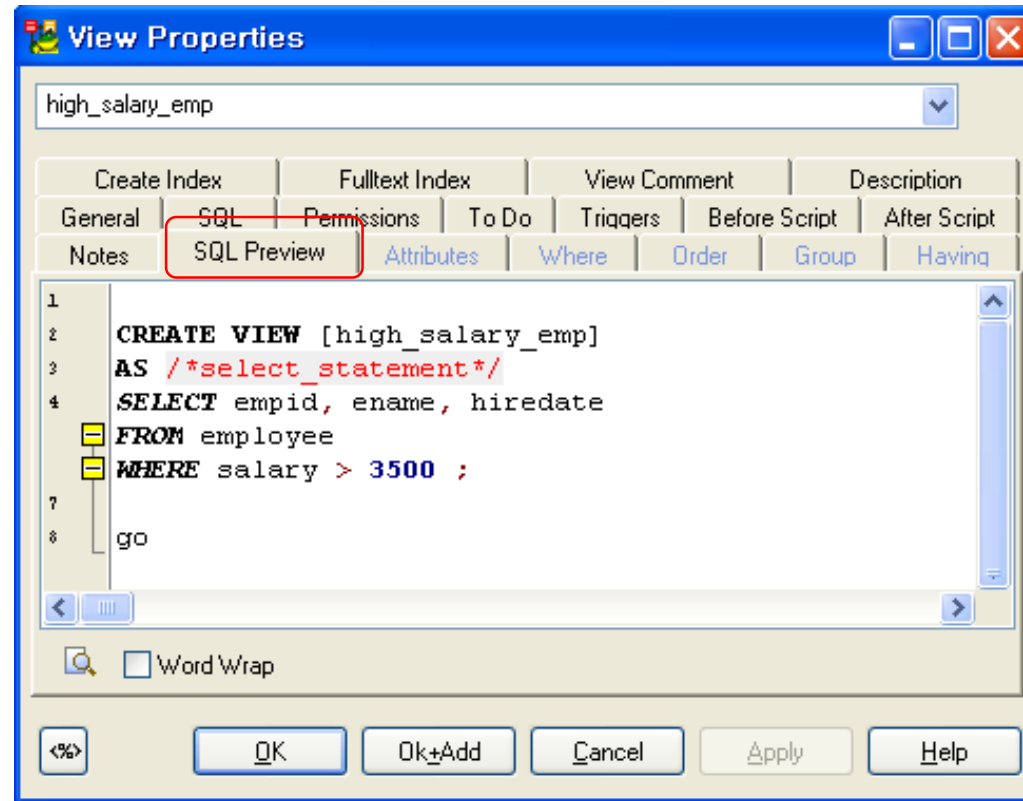
(4)



11.4 뷰(view)의 설계

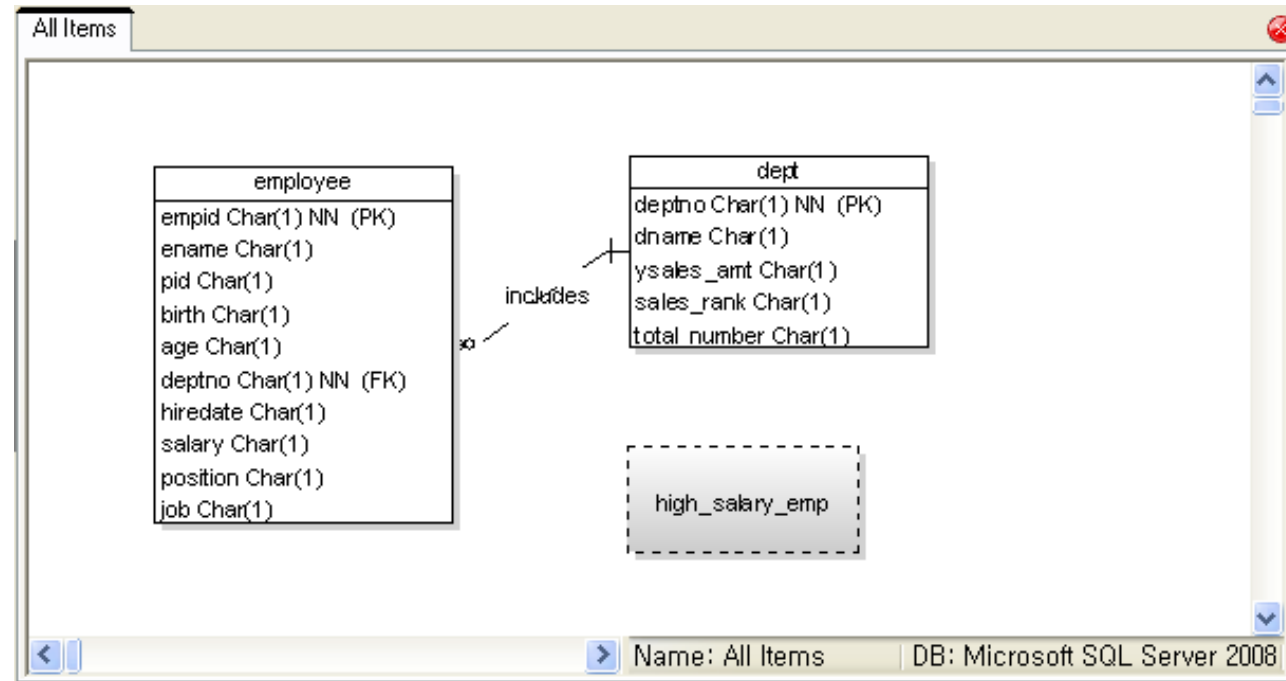
- 모델링 도구에서 뷰의 입력

(5)



11.4 뷰(view)의 설계

- 모델링 도구에서 뷰의 입력



<생성된 뷰>

[요약] 도메인 정의서

❖ 도메인:

- 엔티티 속성들이 가질 수 있는 값들의 집합
- 테이블 컬럼에 대한 데이터 타입

❖ 도메인 정의 방법

- 1) 각 엔티티 속성을 모아서 목록을 만든다
- 2) 속성의 이름에 여러 단어를 포함한 경우 이를 분리한다
- 3) 속성에 포함된 마지막 단어를 가지고 정렬한다
- 4) 도메인을 정한다
- 5) 도메인만 모아 중복을 제거하고 정렬한 뒤 데이터 타입과 길이를 정한다

❖ 도메인 지정 방법

- 1) 직접 데이터 타입과 길이 지정
- 2) 도메인을 먼저 정의한 뒤 각 컬럼에 도메인 지정

❖ 도메인 사용 이유

동일한 성격의 데이터를 저장하는 컬럼들 사이에서 데이터 타입이나 길이가 서로 다르게 지정되는 것을 막기 위한 것

요구사항 명세서

요구 사항 명세서

1. 각각의 코스는 한 과목으로 구성되며, 한 명의 강사가 강의한다.
2. 수강생은 하나 이상의 코스를 수강 신청할 수 있다.
3. 코스의 정원을 초과한 수강 신청은 실패한다.
4. 강사는 다음과 같이 등급을 자동으로 적용한다.
 - 코스 운영 40회 이상 : 1등급
 - 코스 운영 30회 이상 : 2등급
 - 코스 운영 20회 이상 : 3등급
 - 코스 운영 10회 이상 : 4등급
 - 코스 운영 0회 이상 : 5등급(초기값)
5. 시간당 강사료는 다음과 같이 적용한다.
 - 1등급 : 시간당 5만원
 - 2등급 : 시간당 4만원
 - 3등급 : 시간당 3만원
 - 4등급 : 시간당 2만원
 - 5등급 : 시간당 1만원
6. 강사료는 다음과 같이 자동으로 계산한다.
$$(\text{시간당 강사료}) \times 8\text{시간/일} \times (\text{강의 일수})$$
7. 수강생에게는 다음과 같이 포인트를 자동으로 적용한다.
 - 1) 처음에 0포인트에서 시작함
 - 2) 1회 수강 신청(완료)시 1포인트 추가
 - 3) 최고 50포인트를 넘지 않도록 함
8. 수강 신청시 할인율은 다음과 같이 자동으로 적용한다.
 - 1) 해당 수강생의 현재(수강 신청 직전)의 포인트를 그대로 적용한다.
 - 2) 할인율은 수강료 할인율(%)이 됨.
9. (수강 신청) 등록 금액은 다음과 같이 자동으로 계산한다.
$$(\text{해당 코스의 수강료}) \times (100 - \text{할인율}) / 100$$

요구사항 명세서

실체/관계 분석

[범례]

코스 : 실체

강의한다 : 관계 (동사)

1. 각각의 코스는 한 과목으로 구성되며, 한 명의 강사가 강의한다.
2. 수강생은 하나 이상의 코스를 수강 신청할 수 있다.
3. 코스의 정원을 초과한 수강 신청은 실패한다.
4. 강사는 다음과 같이 등급을 자동으로 적용한다.
 - 코스 운영 40회 이상 : 1등급
 - 코스 운영 30회 이상 : 2등급
 - 코스 운영 20회 이상 : 3등급
 - 코스 운영 10회 이상 : 4등급
 - 코스 운영 0회 이상 : 5등급(초기값)

도메인 정의서

속성 분석

이름 영문화

실체	속성	특기 사항
과목	과목 ID	PK
	과목명	
강사	강사 ID	PK
	강사 이름	
	주민번호	YYMMDD-#####
	등급	1~5
	강사 전화	
수강생	수강생 ID	PK
	수강생 이름	
	포인트	0~50
	수강생 전화	
	주소	
코스	코스 ID	PK
	과목 ID	FK
	강사 ID	FK
	정원	
	교재명	
	시작 일자	
	종료 일자	
	수강료	
	강사료	
수강 신청	수강 신청 ID	PK
	수강생 ID	FK
	코스 ID	FK
	할인율	0~50
	등록 일자	
	등록 금액	

실체(테이블) 이름	속성 이름	열 이름	식별 번호
과목 (subject)	과목 ID	subject_id	1
	과목명	subj_name	2
강사 (instructor)	강사 ID	instructor_id	3
	강사 이름	inst_name	4
	주민번호	jumin	5
	등급	grade	6
	강사 전화	inst_phone	7
수강생 (student)	수강생 ID	student_id	8
	수강생 이름	stud_name	9
	포인트	point	10
	수강생 전화	stud_phone	11
	주소	address	12
코스 (course)	코스 ID	course_id	13
	과목 ID	subject_id	14
	강사 ID	instructor_id	15
	정원	capacity	16
	교재명	textbook	17
	시작 일자	start_date	18
	종료 일자	end_date	19
	수강료	sugang_fee	20
	강사료	lecture_fee	21
수강 신청 (register)	수강 신청 ID	register_id	22
	수강생 ID	student_id	23
	코스 ID	course_id	24
	할인율	discount_rate	25
	등록 일자	reg_date	26
	등록 금액	reg_price	27

[예제] 모델링 산출물: 학원관리

도메인 정의서

1. 속성 이름 정렬 2 속성 이름 분류

구분	속성 이름군	설명	도메인 이름
이름	강사 이름, 수강생 이름	짧은 이름	DM_SNAME
	과목명, 교재명	긴 이름	DM_LNAME
	주소	주소	DM_ADDR
날짜	등록 일자	날짜(생략 가능)	DM_DATE_N
	시작 일자, 종료 일자	날짜(생략 불가)	DM_DATE_NN
	등급	등급	DM_GRADE
코드	강사 전화, 수강생 전화	전화	DM_PHONE
	주민번호	주민번호	DM_JUMIN
	과목 ID	2바이트(smallint) IDENTITY 주 키	DM_SID
	강사 ID, 수강생 ID, 코스 ID, 수강 신청 ID	4바이트(int) IDENTITY 주 키	DM_IID
	과목 ID	2바이트(smallint) 외래 키	DM_SFK
	강사 ID, 수강생 ID, 코스 ID	4바이트(int) 외래 키	DM_IFK
	포인트	포인트	DM_POINT
수량	할인율	할인율	DM_DISCNT
	정원	정원	DM_CAPA
금액	강사료, 등록 금액	금액(생략 가능)	DM_PRICE_N
	수강료	금액(생략 불가)	DM_PRICE_NN

(출처: "데이터베이스 관리와 실습", 손광수, 한빛아카데미, 2013.07)  동양미래대학교

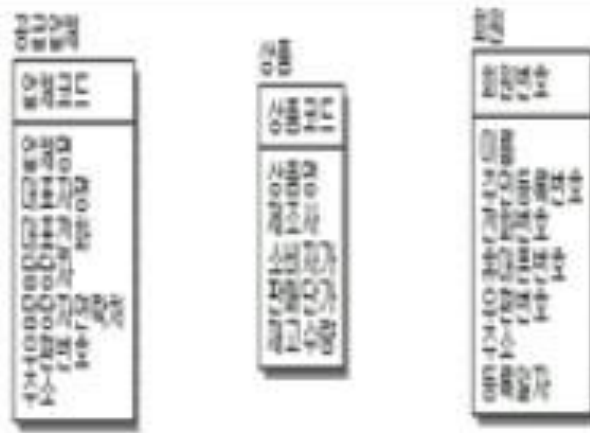
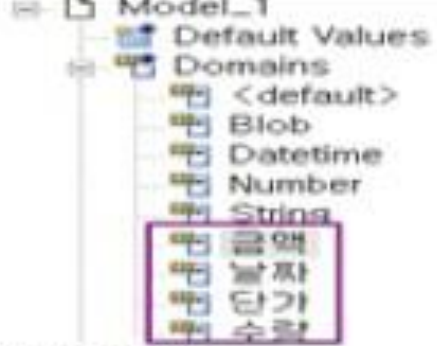
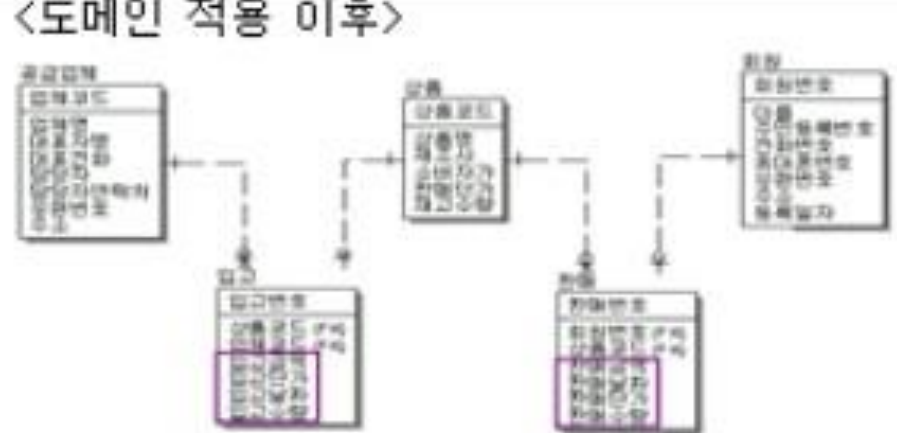
3. 도메인 정의

구분	도메인 이름	데이터 형	크기	N/NN	ID	기본값	유효성 검증(CHECK, Rules)	속성/열 (식별 번호)
이름	DM_SNAME	varchar	10	NN				4, 9
	DM_LNAME	varchar	50	NN				2, 17
	DM_ADDR	varchar	80	N				12
날짜	DM_DATE_N	char	6	N			@date LIKE '[0-9][0-9][1-9][1-9]' OR @date LIKE '[0-9][0-9][1-9][1-2][0-9]' OR @date LIKE '[0-9][0-9][1-9][3][0-1]' OR @date LIKE '[0-9][0-9][1][0-2][1-9]' OR @date LIKE '[0-9][0-9][1][0-2][1-2][0-9]' OR @date LIKE '[0-9][0-9][1][0-2][3][0-1]'	26
	DM_DATE_NN	char	6	NN				18, 19
코드	DM_GRADE	tinyint		NN	5		grade >= 1 AND grade <= 5	6
	DM_PHONE	varchar	14	N				7, 11
	DM_JUMIN	char	14	NN			jumin LIKE '[0-9][0-9]0[1-9]0[1-9]-[1-2][0-9][0-9][0-9][0-9][0-9]' OR jumin LIKE '[0-9][0-9]0[1-9][1-2][0-9]-[1-2][0-9][0-9][0-9][0-9][0-9]' OR jumin LIKE '[0-9][0-9]0[1-9]3[0-1]-[1-2][0-9][0-9][0-9][0-9][0-9]' OR jumin LIKE '[0-9][0-9]1[0-2]0[1-9]-[1-2][0-9][0-9][0-9][0-9][0-9]' OR jumin LIKE '[0-9][0-9]1[0-2][1-2][0-9]-[1-2][0-9][0-9][0-9][0-9][0-9]' OR jumin LIKE '[0-9][0-9]1[0-2]3[0-1]-[1-2][0-9][0-9][0-9][0-9][0-9]'	5
	DM_SID	smallint		(NN)	Y			1
	DM_IID	int		(NN)	Y			3, 8, 13, 22
	DM_SFK	smallint		NN				14
	DM_IFK	int		NN				15, 23, 24
	DM_POINT	tinyint		NN	0		point >= 0 AND point <= 50	10
	DM_DISCNT	tinyint		N			discount_rate >= 0 AND discount_rate <= 50	25
	DM_CAPA	tinyint		NN				16
금액	DM_PRICE_N	int		N				21, 27
	DM_PRICE_NN	int		NN	0			20

[실습 1-1] 도메인 제약조건 check : (er73-도메인-개인신상-00)

<p>개인신상</p> <p>일련번호: int</p> <p>이름: varchar(8)</p> <p>주민등록번호: char(14)</p> <p>연락처: char(14)</p> <p>혈액형: char(2)</p> <p>신장: int</p>	<p><도메인></p> <ol style="list-style-type: none"> 1. 일련번호 : identify (1,1) 2. 주민등록번호 : 주민번호 check (User-Defined) 주민등록번호 LIKE '[0-9][0-9][0-9][0-9][0-9][0-9]-[0-9][0-9][0-9][0-9][0-9][0-9][0-9]' 3. 혈액형 : 혈액형 check (Valid Values List: A, B, O, AB) 4. 신장 : 신장 check (Min/Max: MIN 0)
--	---

[실습 1-2] 도메인 (날짜, 단가, 수량, 금액): (er73-도메인-상품회원-00)

	 <p>[properties %EntityName%AttDomain]</p>	<p><도메인 적용 이후></p> 
--	---	---

[요약] 테이블 기술서

❖ 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 기술서

하나의 테이블에 대한 모든 정보가 상세히 기술 데이터베이스를 기반으로 응용 프로그램을 작성하는 개발자나 유지보수 담당자에게 매우 유용

❖ 도메인 지정 방법

- 1) 직접 데이터 타입과 길이 지정
- 2) 도메인을 먼저 정의한 뒤 각 컬럼에 도메인 지정

❖ 도메인 사용 이유

동일한 성격의 데이터를 저장하는 컬럼들 사이에서 데이터 타입이나 길이가 서로 다르게 지정되는 것을 막기 위한 것

[예제] 모델링 산출물: 도서관리

테이블 기술서

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	Default	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)



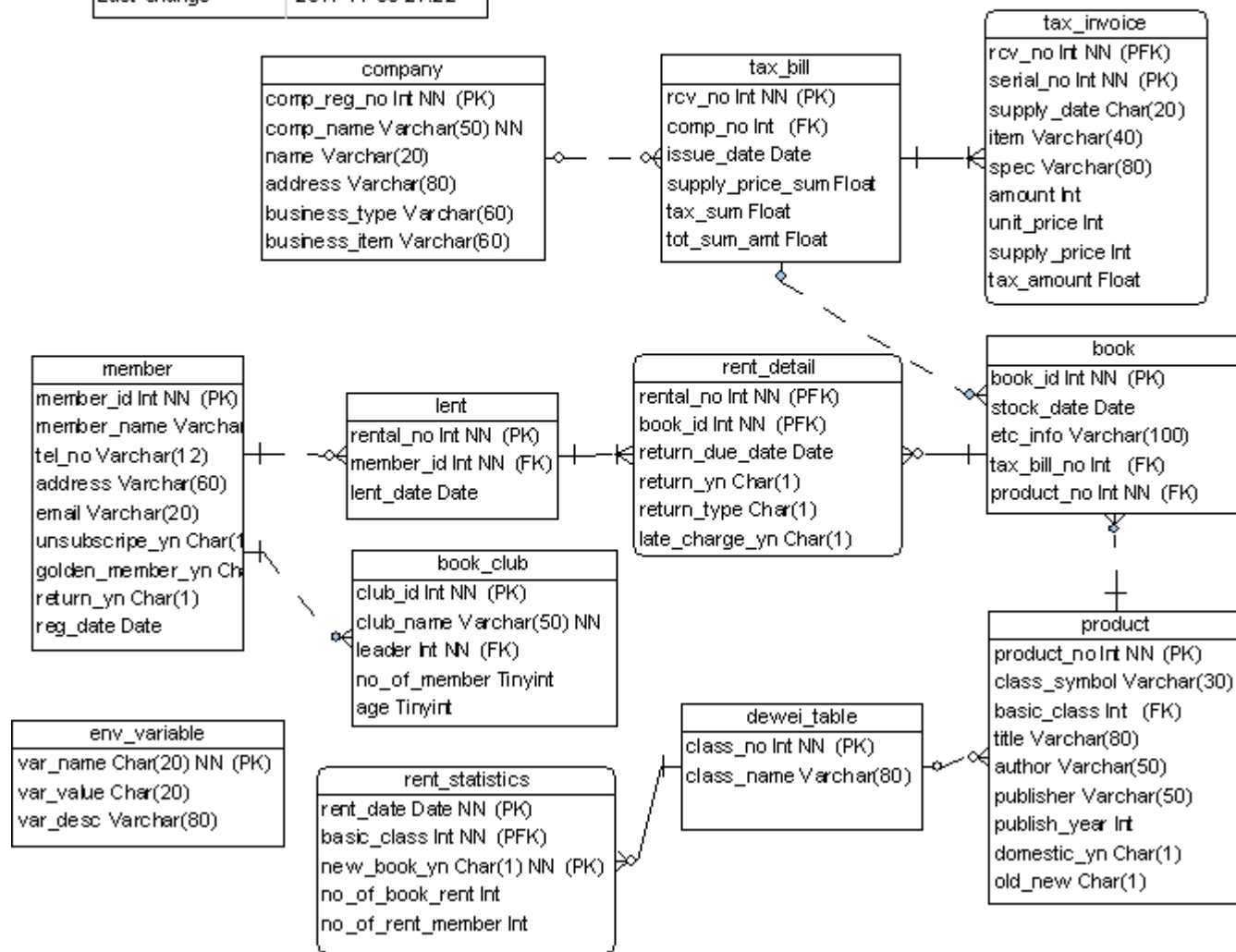
Project	도서관 관리
Model	Main
Author	김길영
Company	D.U.C
Version	1.2
Date of Creation	2011-11-30 21:00
Last Change	2011-11-30 21:15



[예제] 모델링 산출물: 도서관리

물리적 설계: 최종 ERD

Project	도서관관리
Model	Main
Author	김길영
Company	D.J.C
Version	1.2
Date of Creation	2011-11-30 21:00
Last Change	2011-11-30 21:22



인덱스

14주차 01-02

담당교수: 김희숙
(jasmin11@hanmail.net)

[요약] 인덱스

❖ 인덱스 사용 이유

- ❖ 파일 내 레코드 위치를 빨리 찾기 위해 사용
- ❖ 인덱스가 없는 경우 순차검색

-인덱스 사용 예 이름 정렬

도서관의 책 정렬

사전의 단어 정렬

전화번호부의 이름 정렬

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
....

‘류용신’의 전화번호는?

<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



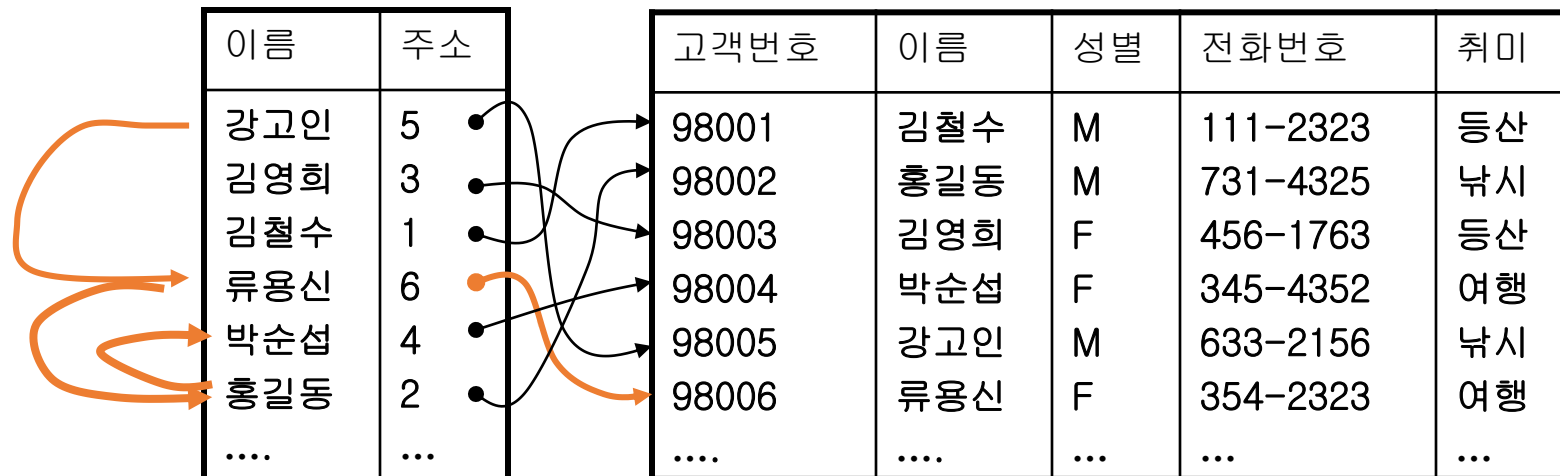
고객번호	이름	성별	전화번호	취미
98005	강고인	M	633-2156	낚시
98003	김영희	F	456-1763	등산
98001	김철수	M	111-2323	등산
98006	류용신	F	354-2323	여행
98004	박순섭	F	345-4352	여행
98002	홍길동	M	731-4325	낚시
....

11.5 인덱스의 설계

➤ 인덱스로 검색

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

인덱스



정렬되어
있음

정렬되어
있지 않음

11.5 인덱스의 설계

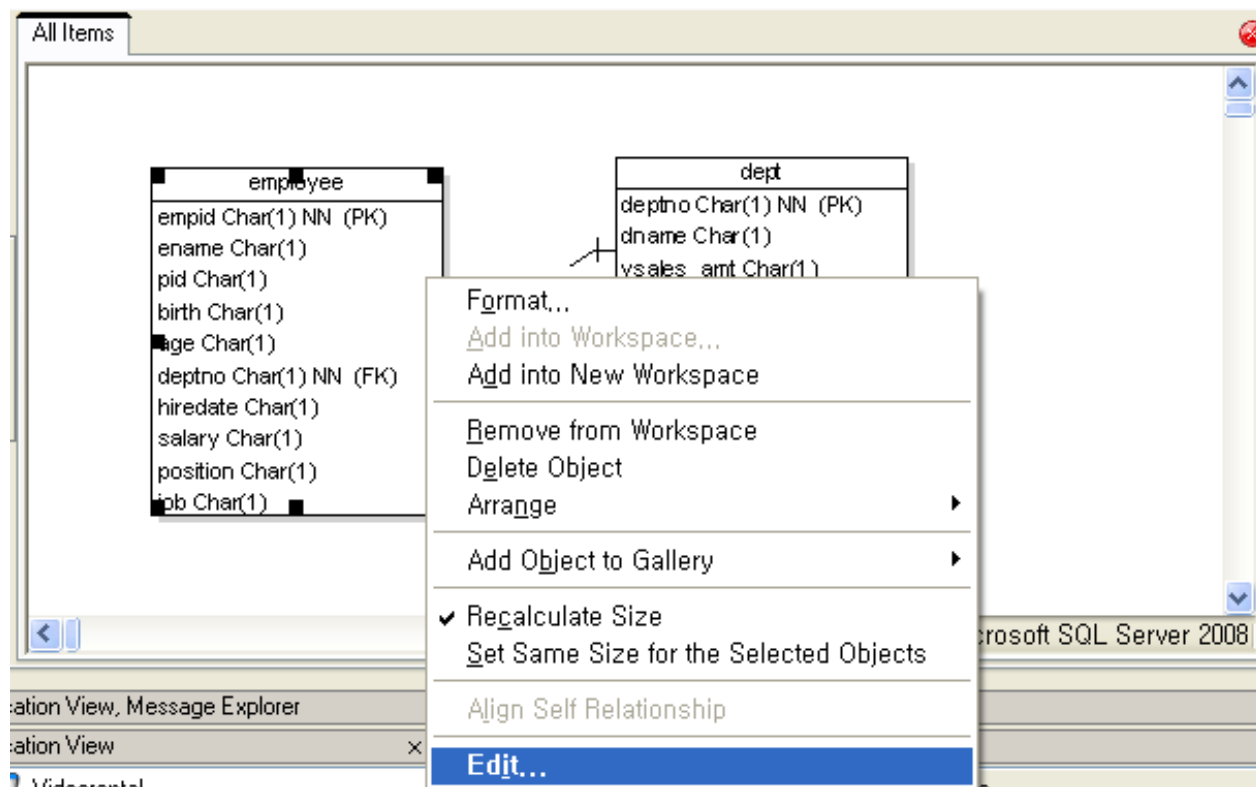


- 인덱스에 대한 검토
 - 인덱스의 수가 많으면 인덱스를 재정렬하는데 많은 시간이 소모되므로 DBMS의 성능을 저하 시킨다.
 - 그러므로 인덱스는 꼭 필요한 컬럼에 대해서만 지정을 해야 한다
 - 모든 경우에 대해 인덱스가 성능을 발휘하는 것은 아니다
 - 인덱스로 지정하는 컬럼은 SQL의 WHERE 절에서 비교 대상이 되는 컬럼 또는 JOIN에 사용되는 컬럼이어야 한다.
 - 튜플의 수가 적으면 (예:200~300개) 인덱스를 지정하여도 별 효과가 없다.
 - 인덱스로 지정한 컬럼에 의해 검색했을 때 검색 결과가 전체 튜플의 10~15%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

11.5 인덱스의 설계

- 모델링 도구에서 인덱스의 입력

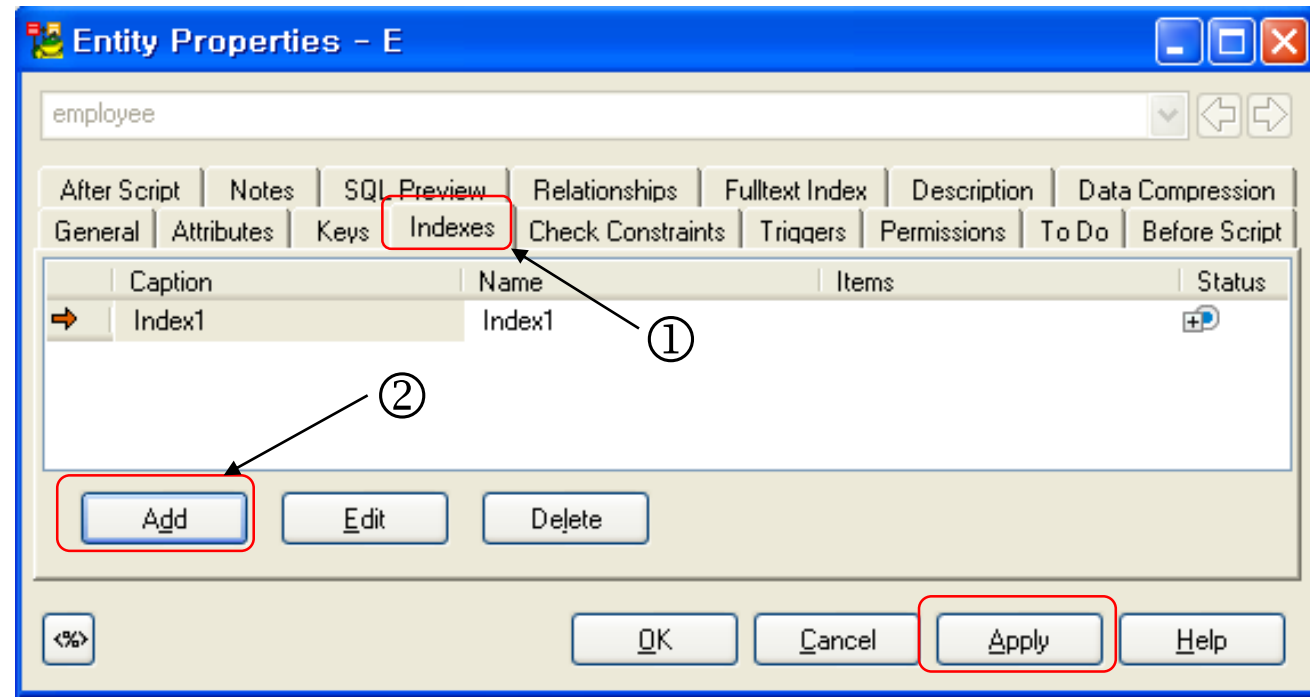
(1)



11.5 인덱스의 설계

- 모델링 도구에서 인덱스의 입력

(2)



11.5 인덱스의 설계

- 모델링 도구에서
인덱스의 입력

(3)

Index Properties - E

Entity: employee

Index1

Data Compression | Spatial Index Parameters | Secondary XML Index Parameters

General | Items | Notes | Description | Index Options

Caption: idx_employee_name = Name: idx_employee_name

☐ Unique ☒ Clustered

Data Space

☒ Filegroup ☐ Partition Scheme

-- Not Specified --

Filestream Filegroup (for Clustered Index)

-- Not Specified --

Expression for Filtered Index (WHERE)

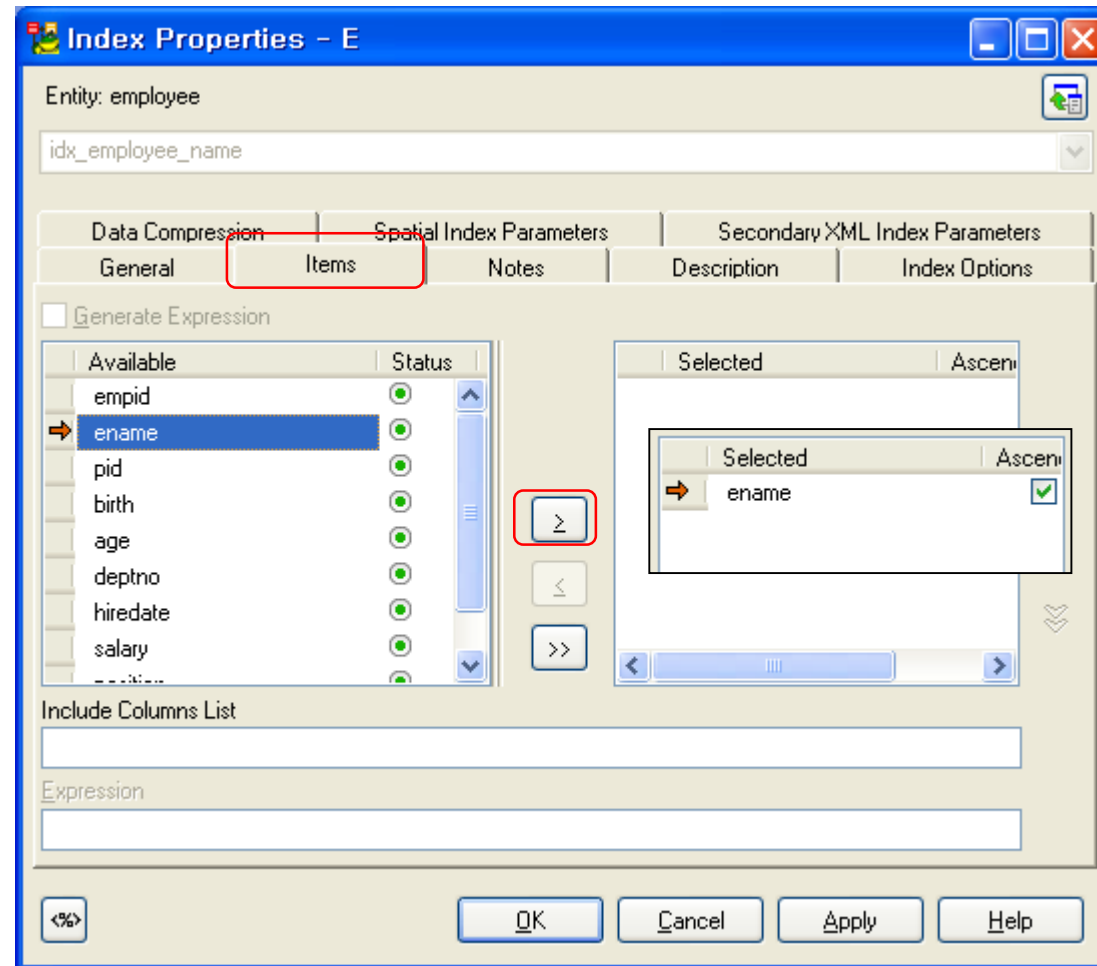
☒ Generate

<%> OK Cancel Apply Help

11.5 인덱스의 설계

- 모델링 도구에서
인덱스의 입력

(4)



역정규화

14주차 02-01

담당교수: 김희숙
(jasmin11@hanmail.net)

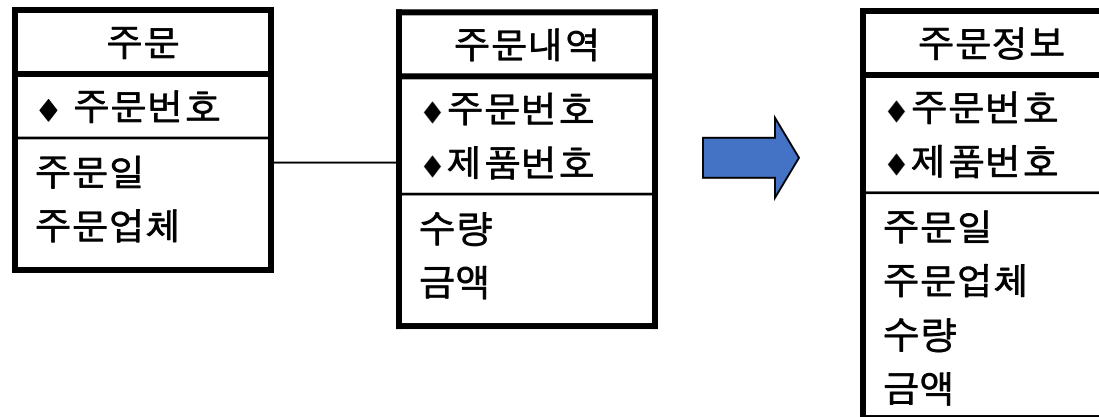
11.3 반정규화



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

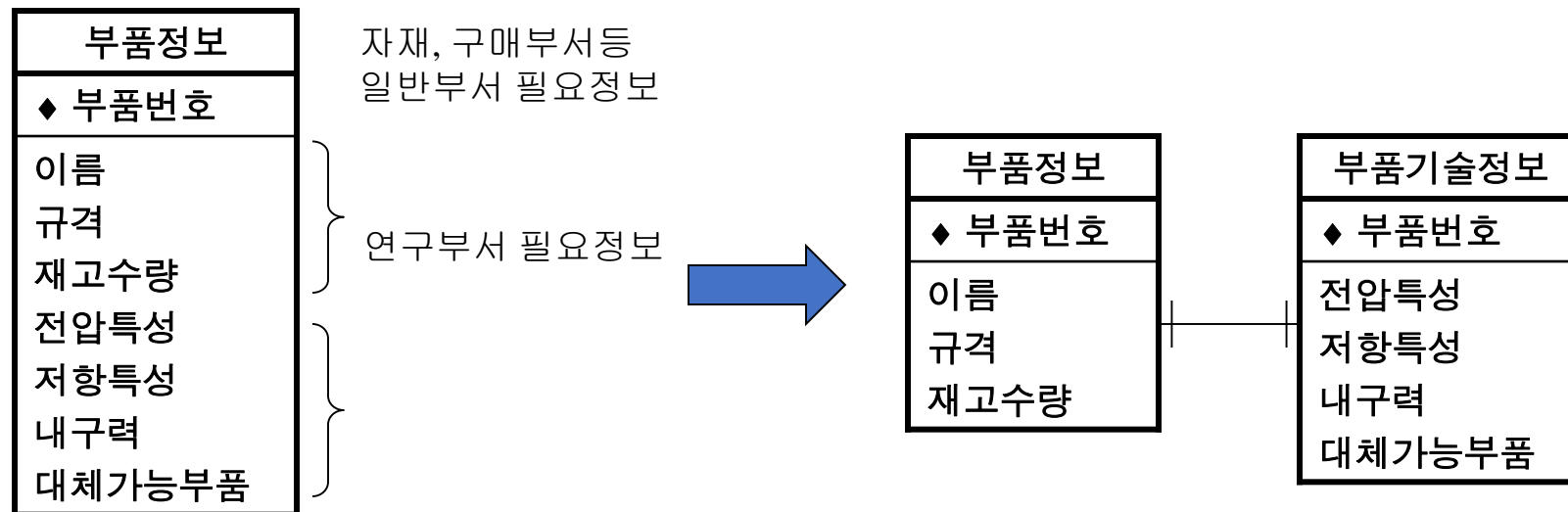
```
SELECT 주문번호, 주문일, 주문업체, 제품번호, 수량, 금액  
FROM 주문, 주문내역  
WHERE 주문.주문번호 = 주문내역.주문번호 ;
```

(반정규화 이후)

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

11.3 반정규화

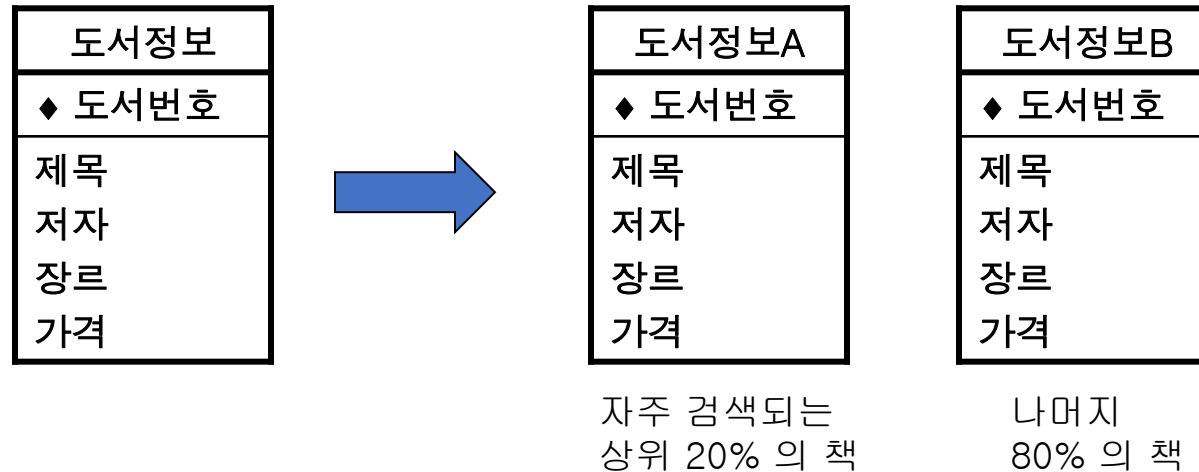
- 엔티티 통합/분할에 의한 반정규화
 - 엔티티의 튜플수 및 속성의 수가 매우 많고, 엔티티의 속성들이 그룹화되어 각 그룹이 특정 부서 혹은 응용 프로그램에 의해서만 사용될 때
 - 엔티티의 데이터 크기 감소, 검색의 분산



<그림 11.14> 엔티티 수직분할에 의한 반정규화

11.3 반정규화

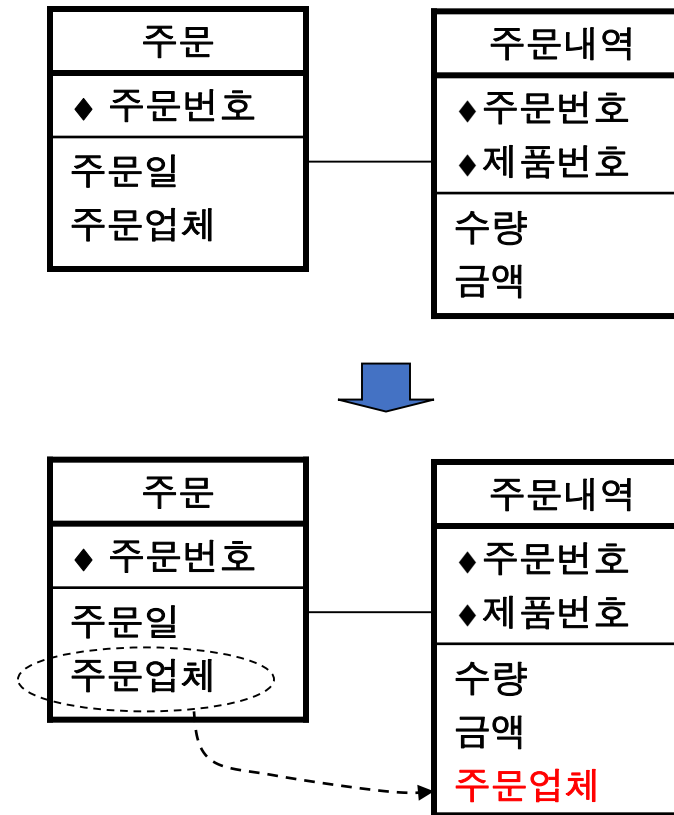
- 엔티티 통합/분할에 의한 반정규화
 - 튜플의 검색 빈도가 다르다는 점을 이용
 - 엔티티의 데이터 크기 감소 효과



<그림 11.15> 엔티티 수평분할에 의한 반정규화

11.3 반정규화

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

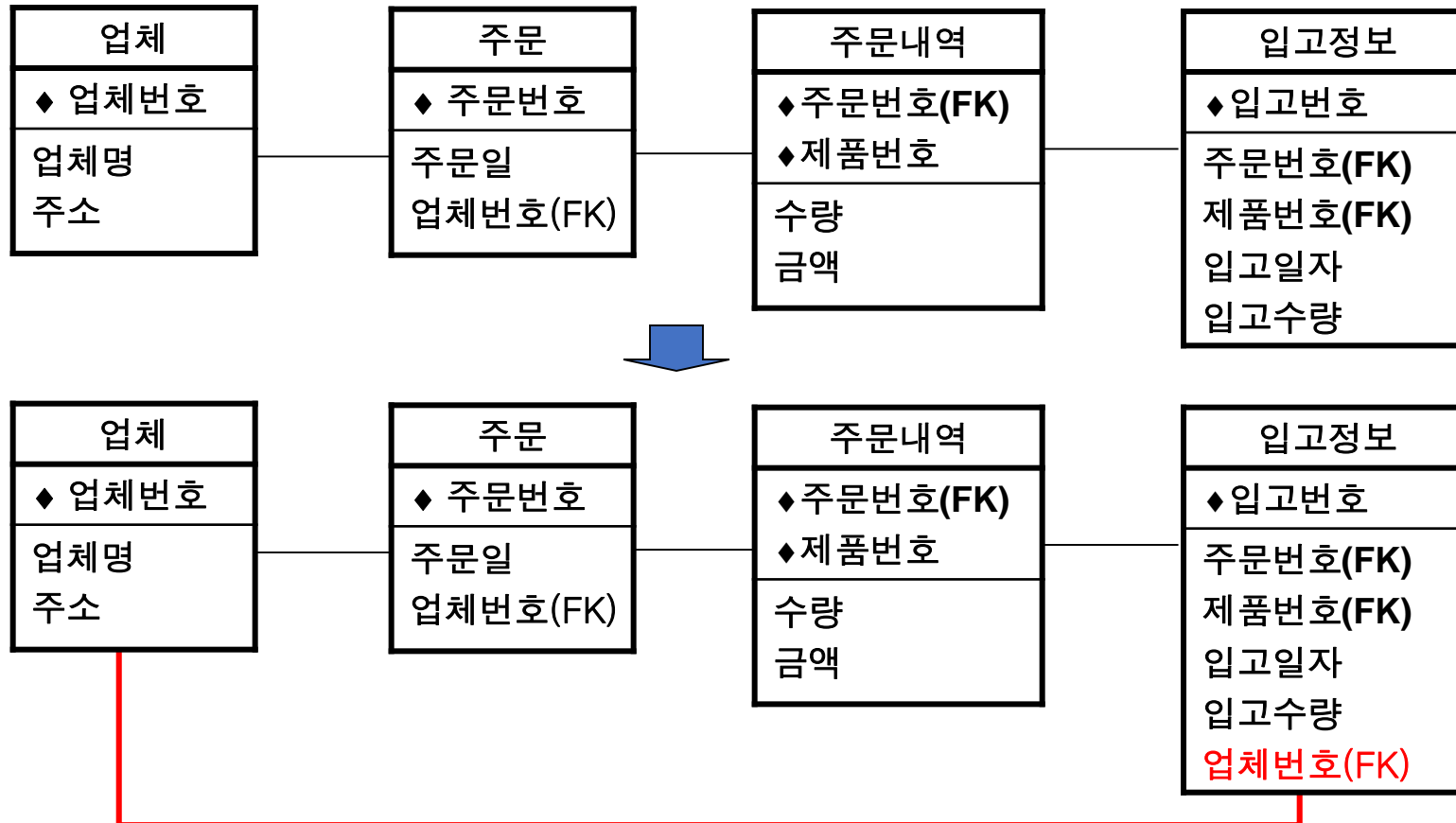


<그림 11.16> 속성의 중복에 의한 반정규화

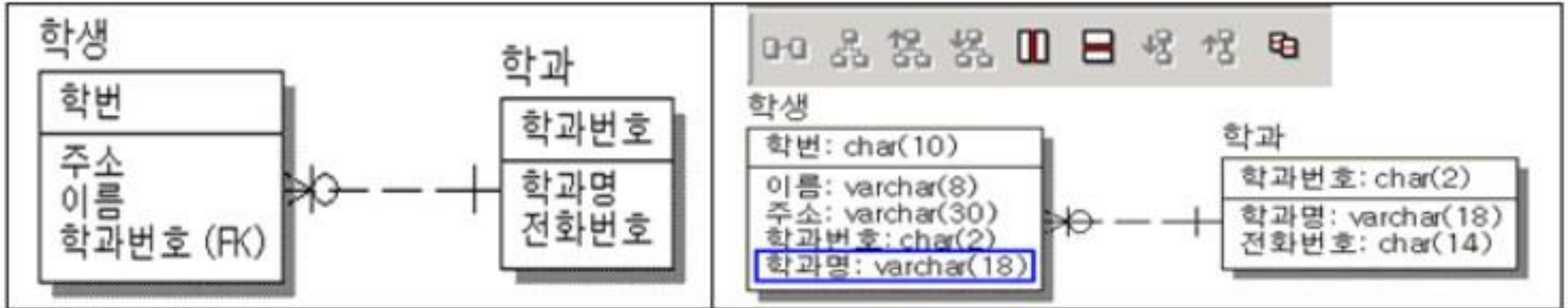
11.3 반정규화

• 관계에 대한 반정규화

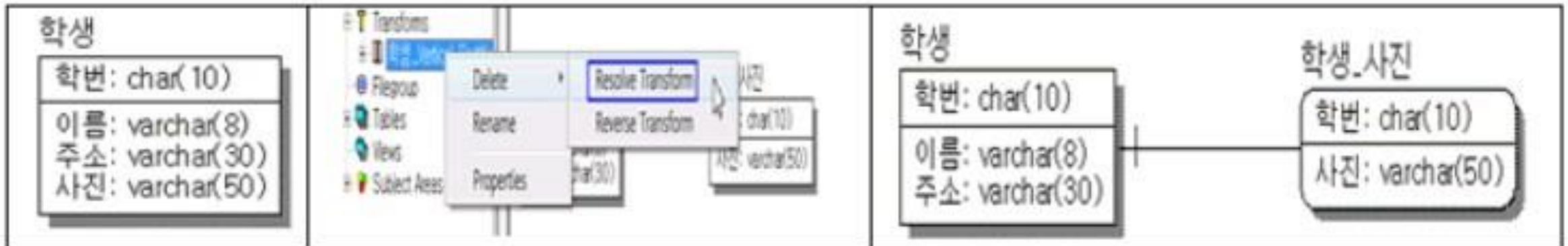
여러 엔티티를 조인해야 원하는 정보를 얻을 수 있는 경우에 적용



[실습 2-1] 컬럼 역정규화 : (er73-역정규화 1-학생학과-00)



[실습 2-2] 역정규화(수직분할): (er73-역정규화 2-학생사진-00)

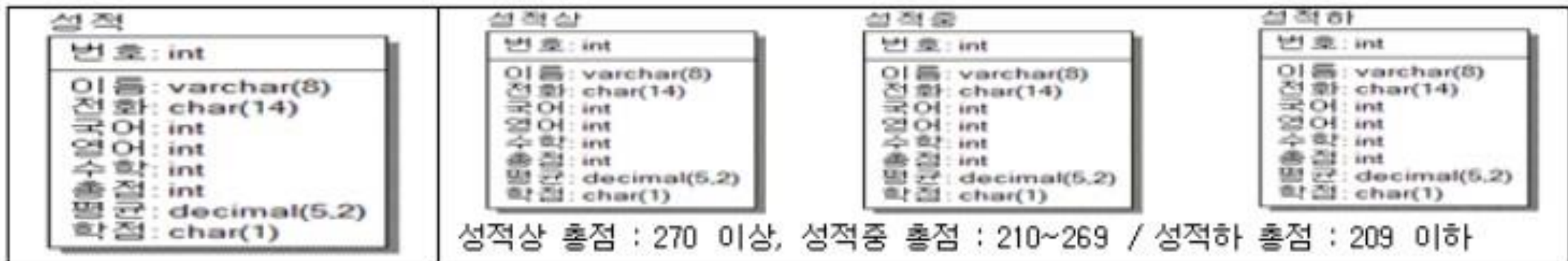


[실습] (역정규화)

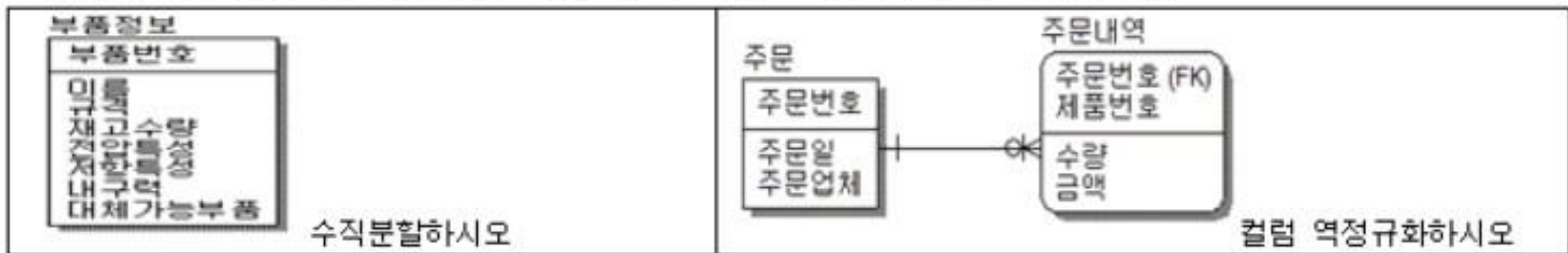
[실습 2-3] 테이블 통합(롤업/롤다운) : (er73-역정규화 2-학생사진-00)



[실습 2-4] 역정규화(수평분할) : (er73-역정규화 3-성적-00)



[Quiz] (오세종 p304~305) (er73-역정규화 4-부품-00, er73-역정규화 4-주문-00)



트랜잭션

14주차 02-02

담당교수: 김희숙
(jasmin11@hanmail.net)



[요약] 트랜잭션 특성(ACID)

❖ 트랜잭션

- 논리적인 작업 단위를 구성하는 연산 들의 집합
- 실행 중 멈추거나 중단되지 않는 최소 작업 단위
- 데이터베이스 응용 프로그램은 트랜잭션의 집합

-트랜잭션

Commit 작업성공

Rollback 작업철회

checkpoint 저장점



[요약] 트랜잭션 특성(ACID)

❖ 트랜잭션 4가지 특성

1) 원자성 Atomicity

트랜잭션에서 정의된 연산들은 모두 성공적으로 실행되었는지
아니면 전혀 실행되지 않은 상태로 남아 있어야 한다(All or Nothing)

2) 일관성 Consistency

트랜잭션이 실행 되기 전의 데이터베이스 내용이 잘못 되어 있지 않다면
트랜잭션이 실행된 이후에도 데이터베이스의 내용에 잘못이 있으면 안된다

3) 고립성 Isolation

트랜잭션이 실행되는 도중에 다른 트랜잭션의 영향을 받아
잘못된 결과를 만들어서는 안된다

4) 지속성 Durability

트랜잭션이 성공적으로 수행되면 그 트랜잭션이 갱신한 데이터베이스의 내용은
영구적으로 저장된다

-트랜잭션 4가지 특성(ACID)

1) 원자성 Atomicity

2) 일관성 Consistency

3) 고립성 Isolation

4) 지속성 Durability



관계대수

14주차 02-03

담당교수: 김희숙
(jasmin11@hanmail.net)

[요약] 관계대수

- ❖ 비절차적 언어(non procedural language)
 - 수행절차를 기술하지 않고 사용자가 원하는 결과만을 명시
- ❖ 절차적 언어(procedural language)
 - 원하는 결과를 얻기 위해 수행되어야 할 일련의 절차를 명시

관계해석
관계대수

[요약] 관계대수

❖ 관계대수

- 수학 수식구조와 거의 동일, 관계대수의 실행결과는 새로운 테이블이 된다

1. 실렉션(selection) : 선택연산

2. 프로젝션(projection) : 추출연산

3. 합집합(union), 교집합(intersect), 차집합(difference)

4. 카티션 프로덕트(product)

5. 조인

- 2개 이상의 테이블로 부터 필요한 열을 선택하여 하나의 테이블로 논리적인 결합하는 것
- 내부조인(세타조인, 동등조인, 자연조인)
- 외부조인(왼쪽외부조인, 오른쪽외부조인, 완전외부조인)

❖ 조인

내부조인

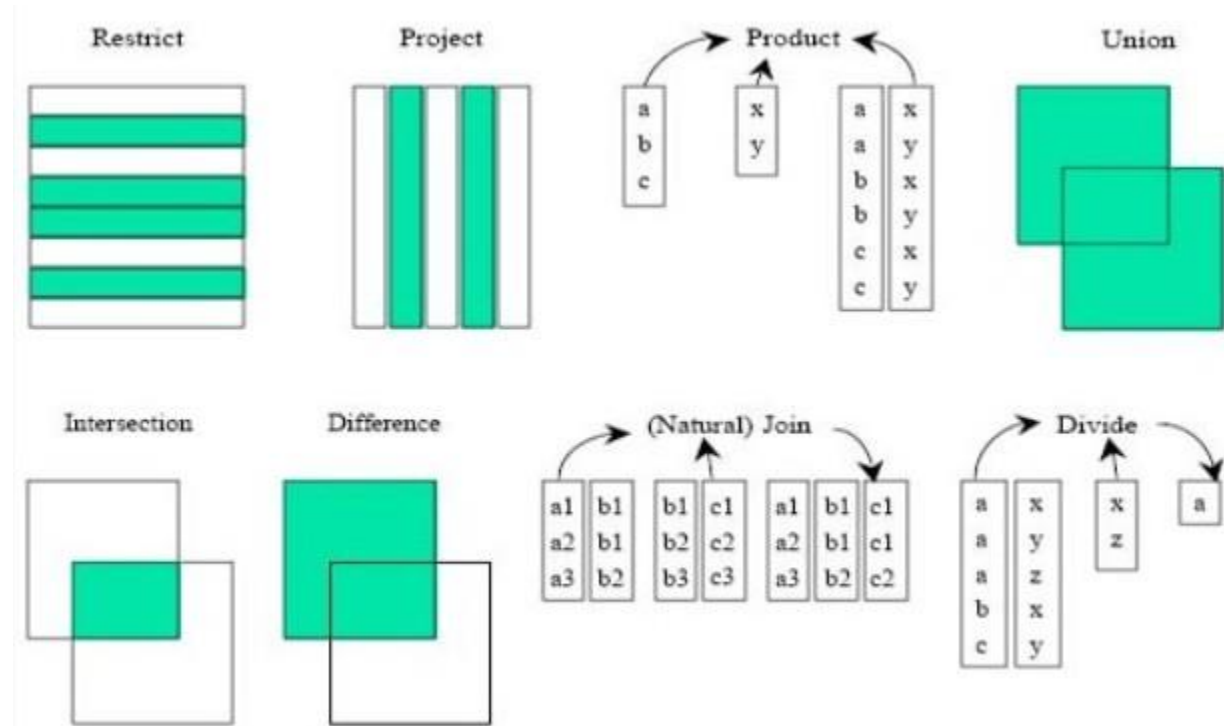
(theta join, equi join, natural join)

외부조인

(left outer join, right outer join,
full outer join)



[요약] 관계대수



질의21) 교수 이름과 소속된 학과 이름을 검색하라

질의 21 $\pi_{\text{professor.name, department.dept_name}}(\sigma_{\text{professor.dept_id} = \text{department.dept_id}}(\text{professor} \times \text{department}))$

질의24) 교수 이름과 소속된 학과 이름을 검색하라

질의 24 $\pi_{\text{professor.name, department.dept_name}}(\text{professor} \bowtie_{\text{professor.dept_id} = \text{department.dept_id}} \text{department})$

꿈을 가지고
행복하게 살아가시기 바랍니다

감사합니다