

[기업수요맞춤형]

1일차 : 오후

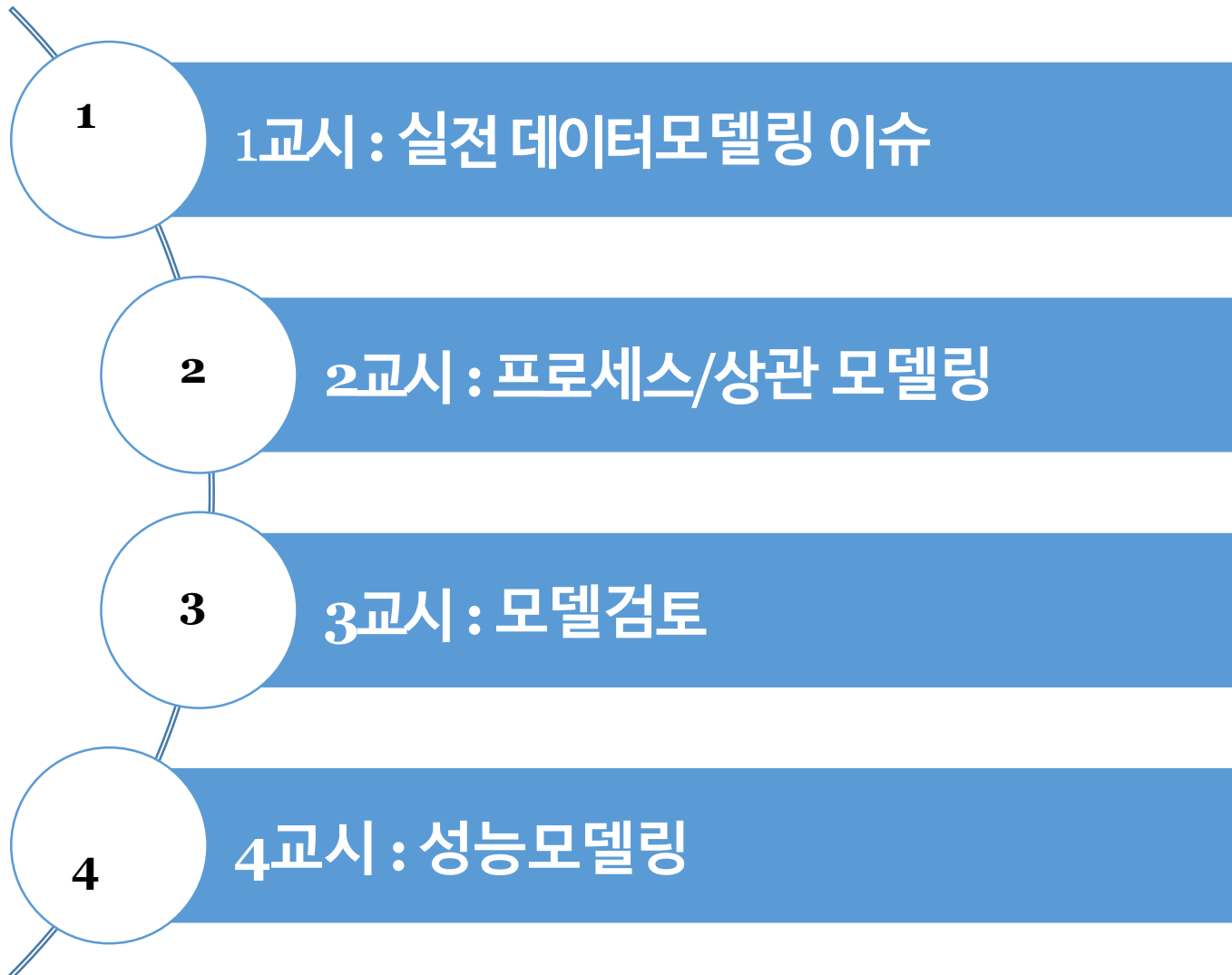
- ◆ 훈련기관명 : 대한상공회의소 서울기술교육센터
- ◆ 훈련과정명 : 실무 데이터베이스 모델링 사례연구(23년)
- ◆ 훈련기간 : 2023.05.24 ~ 2023.05.25

Copyright 2022. Daekyeong all rights reserved



대한상공회의소

서울기술교육센터 기업교육팀



『1과목』 Fundamental

5교시 :

실전 데이터모델링 이슈





1. 주요이슈

주요이슈

- M:N 관계 해소 방법
- 1:1 관계 해소 방법
- 엔티티타입 통합 방법
- 코드 엔티티타입 설계 방법
- 도미노 속성에 대한 모델링 방법
- 메시지 엔티티타입 설계 방법
- 이력 엔티티타입 설계 방법

1. 주요이슈

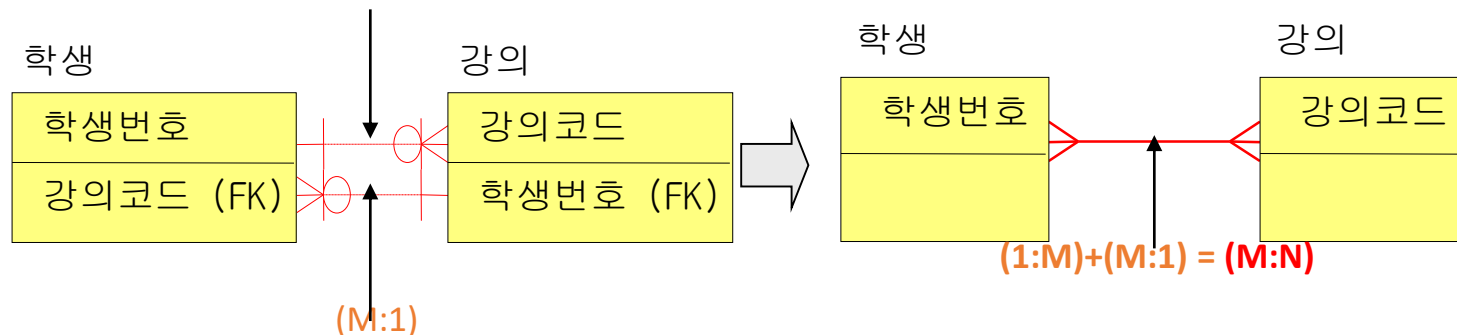
M:N관계 해소방법

- M:N 관계의 해소방법 – 관계 엔티타입 생성에 의한 통합
- M:N 관계의 해소방법 – 식별자에 의한 통합
- M:N 관계의 해소방법 – 속성에 의한 통합

M:N 관계의 생성

M:N관계 생성

한 명의 학생은 여러 개의 강의를
수강할 수 있다
(1:M)



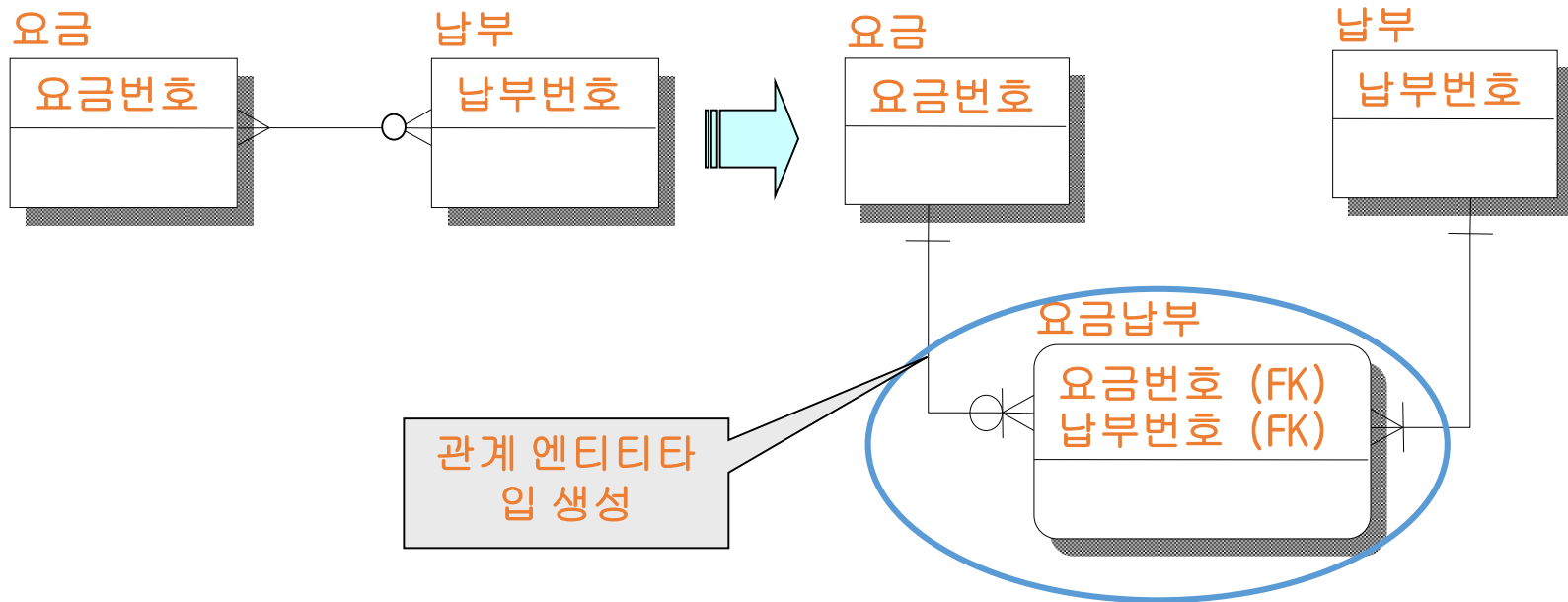
하나의 강의는 여러 명의 수강생이
있을 수 있다

한 명의 학생은 여러 개의 강의를 수강할 수 있고
하나의 강의는 여러 명의 수강생이 있을 수 있다

1. 주요이슈

M:N관계 해소방법

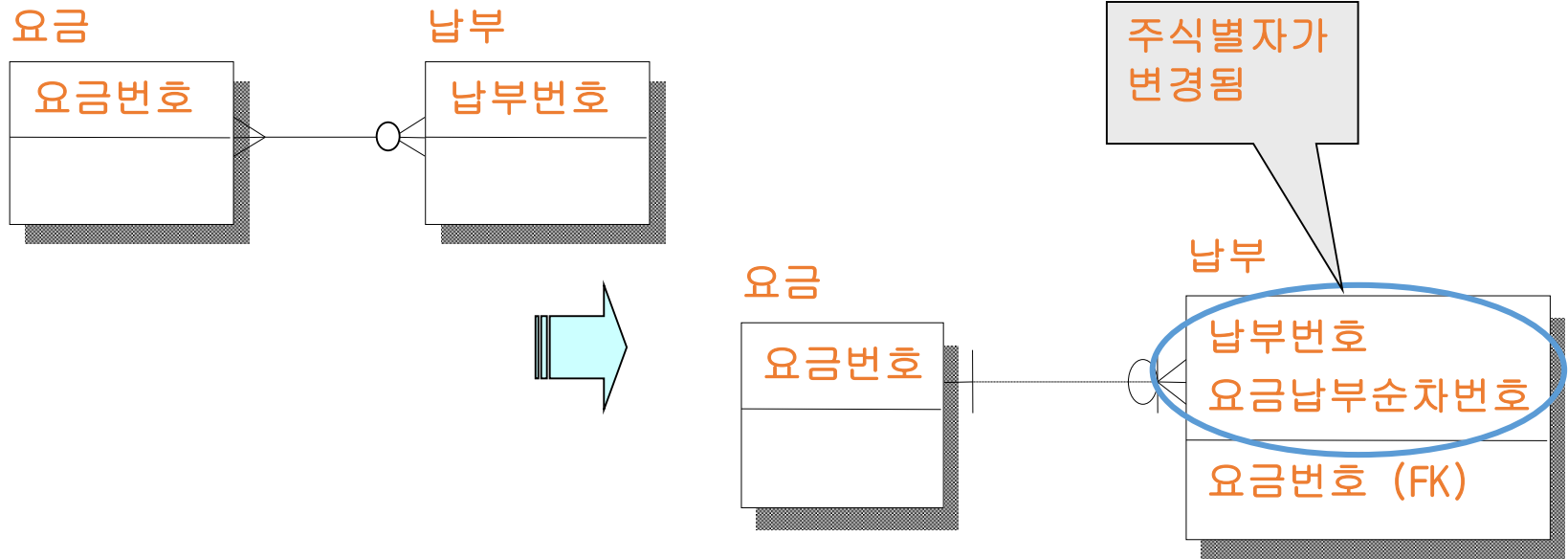
● M:N 관계의 해소방법 - 기본



1. 주요이슈

M:N관계 해소방법

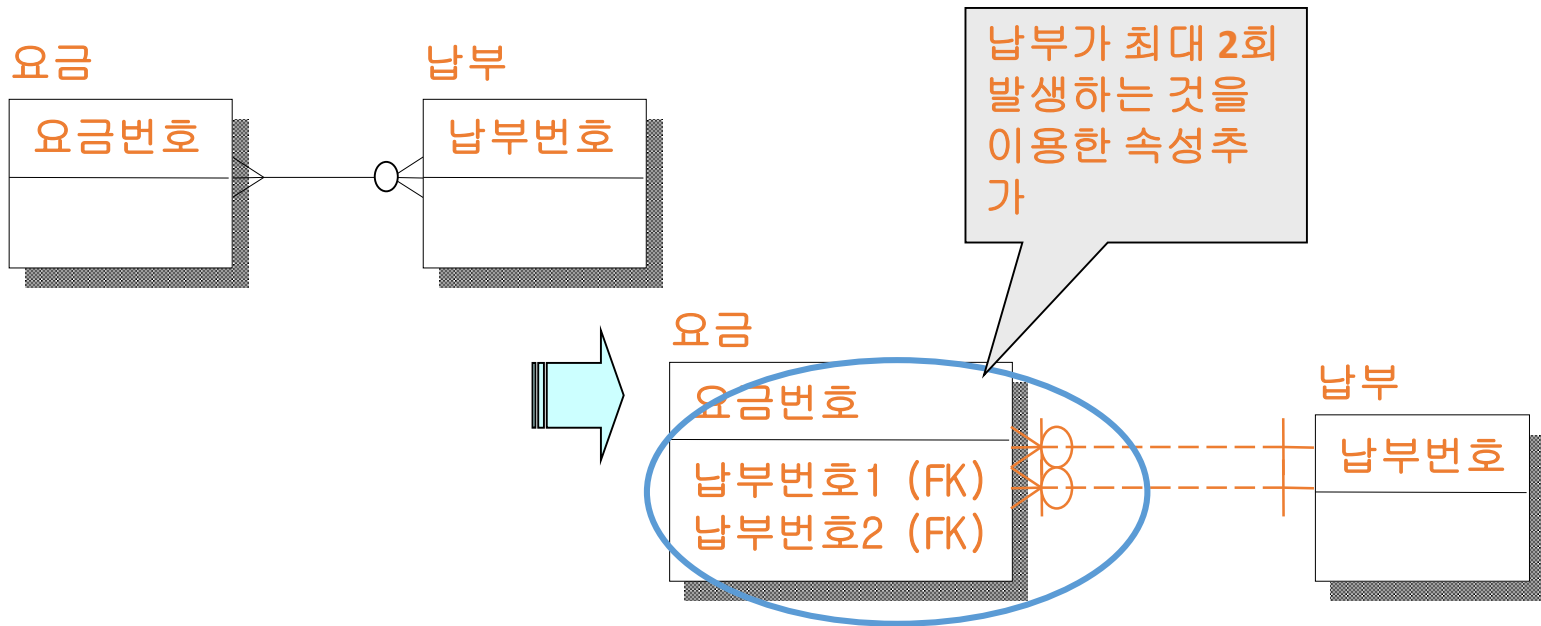
- M:N 관계의 해소방법 – 식별자에 의한 통합



1. 주요이슈

M:N관계 해소방법

- M:N 관계의 해소방법 – 속성에 의한 통합



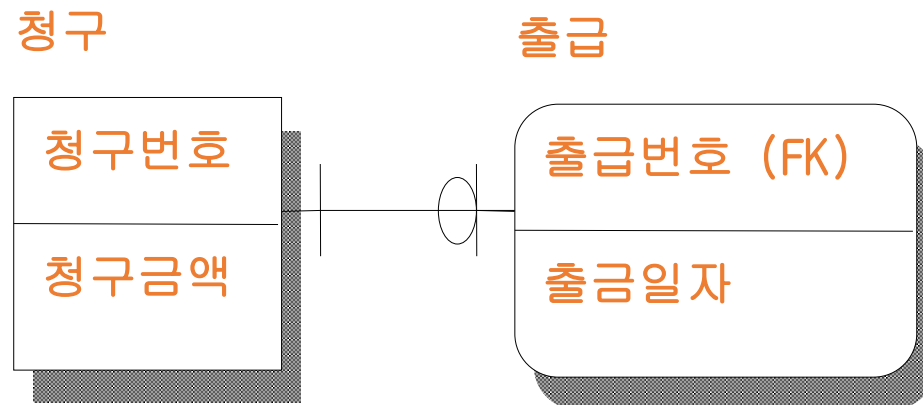
1. 주요이슈

1:1관계 해소방법

- 1:1 관계의 해소방법 – 개별 엔티티타입 유지
- 1:1 관계의 해소방법 – 하나의 엔티티타입통합(완전통합)
- 1:1 관계의 해소방법 – 하나의 엔티티타입통합(부분통합)
- 1:1 관계의 해소방법 – 엔티티타입통합(수퍼타입)

1:1관계 해소방법1

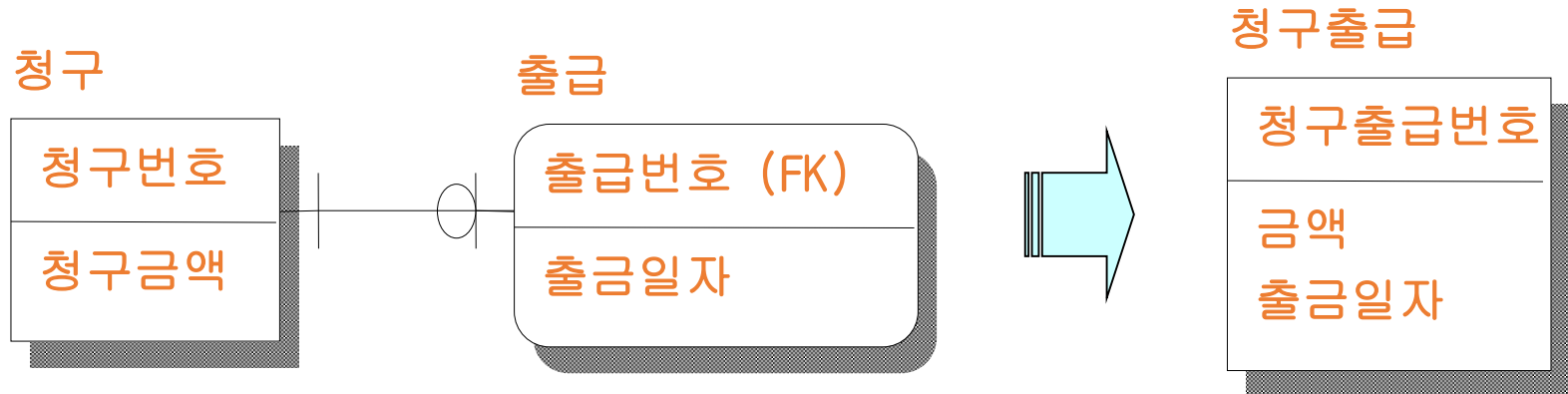
- 1:1 관계의 해소방법 – 개별 엔티티타입 유지



1. 주요이슈

1:1관계 해소방법2

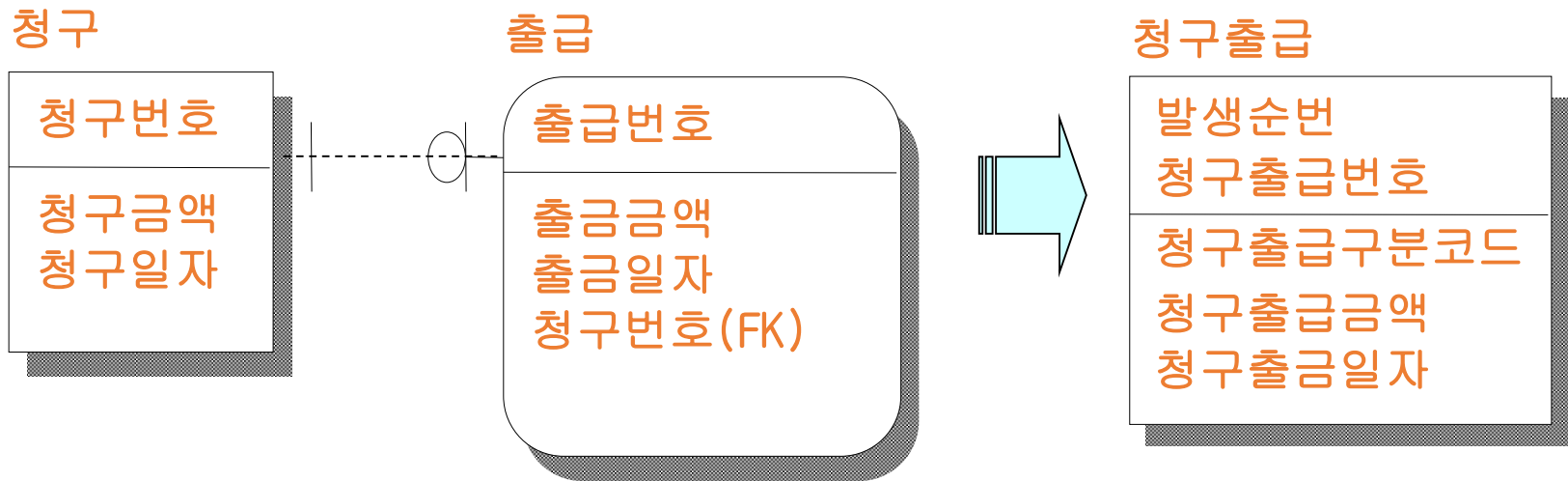
- 1:1 관계의 해소방법 – 하나의 엔티티타입통합(완전통합)



1. 주요이슈

1:1관계 해소방법3

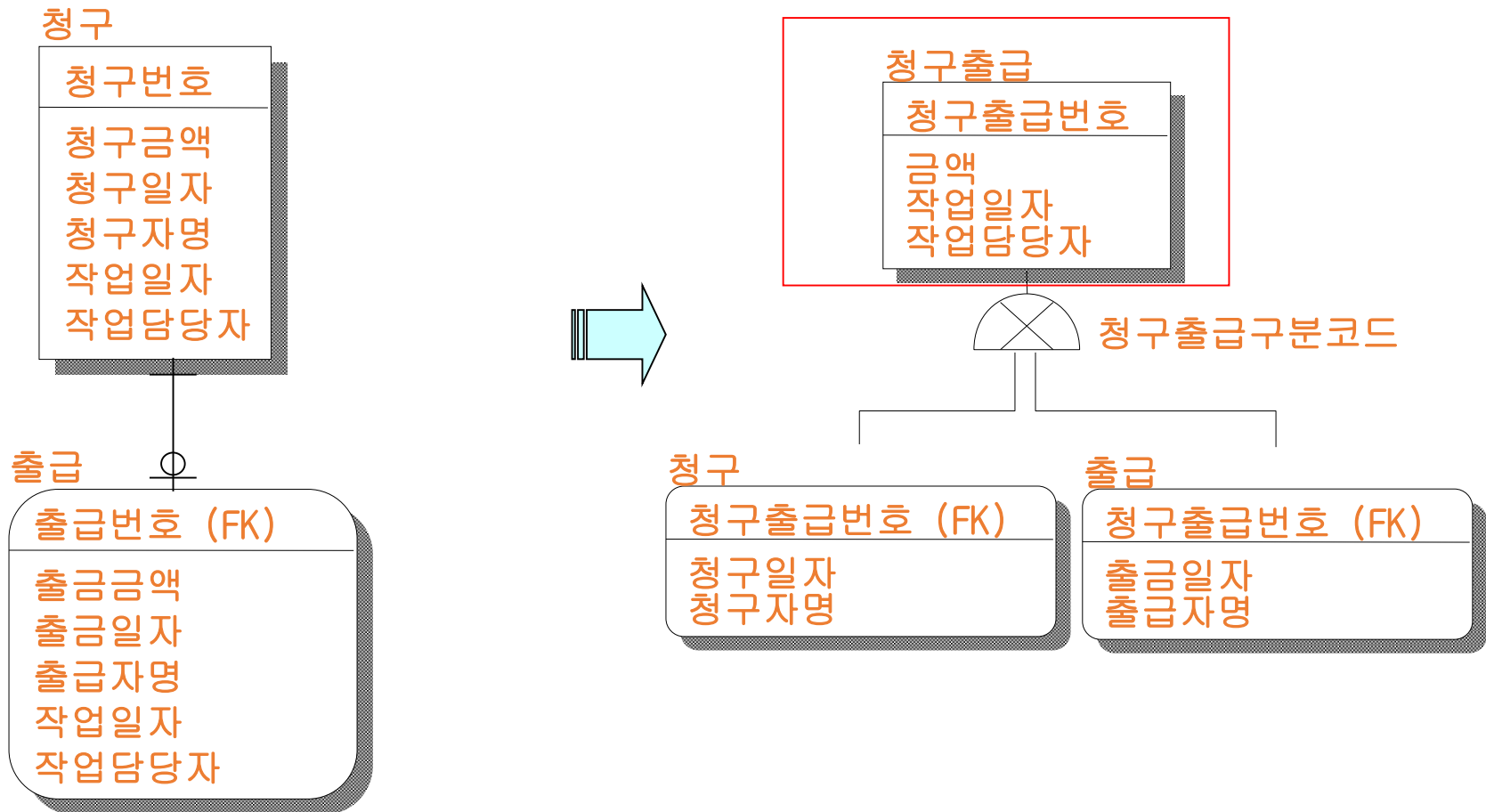
- 1:1 관계의 해소방법 – 하나의 엔티티타입통합(부분통합)



1. 주요이슈

1:1관계 해소방법4

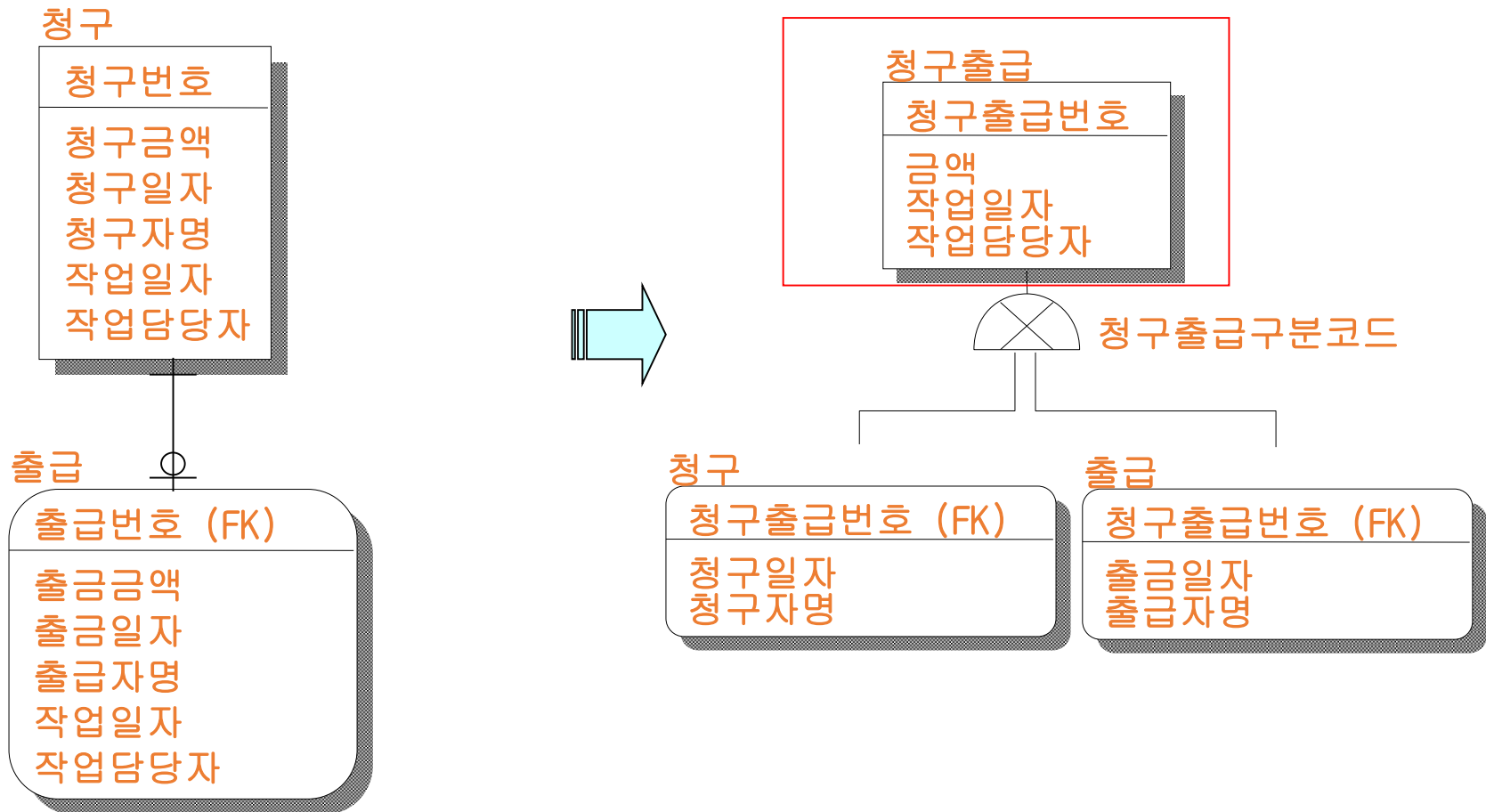
- 1:1 관계의 해소방법 – 엔티타입통합(수퍼타입)



1. 주요이슈

1:1관계 해소방법4

- 1:1 관계의 해소방법 – 엔티타입통합(수퍼타입)



2. 엔티티타입 통합

엔티티타입 통합

- 엔티티타입이 통합되면 업무처리 프로세스가 변형되거나 화면 사용자 인터페이스가 바뀌는가?





2. 엔티티타입 통합

엔티티타입 통합의 장단점

장점

- 종합적으로 정보를 조회하기에 용이함
- 불필요한 조인을 제거하여 성능을 향상시킨다.
- 비슷한 속성이 합해지므로 엔티티간 중복성이 제거됨
- 동일한 규칙이 적용되는 경우에 한번의 규칙만 적용하면 됨.
- ERD가 간결해진다.
- 물리적인 테이블의 관리 개수가 줄어든다.

단점

- 업무확장에 따른 데이터모델의 확정성이 유연하지 않다.
- 업무흐름을 데이터모델로만 가지고 정확하게 이해하기 어렵다.
- 많은 양의 데이터가 한군데 집약되므로 성능 저하현상도 있을 수 있다.(디스크 배치, 파티셔닝을 잘못할 경우)
- 속성에 제약조건을 걸지 못할 수 있다.
- SQL에 체크 해야 할 조건이 증가한다.

2. 엔티티타입 통합

엔티티타입 통합 진행절차



2. 엔티티타입 통합

엔티티타입 통합

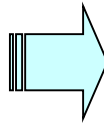
- PRIMARY KEY가 동일한 엔티티타입의 통합-완전통합

등록자

등록자주민등록번호
등록자명
우편번호
시도
시군구
읍면동
번지
휴대폰번호
전화번호
팩스번호

접수자

접수자주민등록번호
접수자명
접수일자



등록자

등록자주민등록번호
이름
우편번호
시도
시군구
읍면동
번지
휴대폰번호
전화번호
팩스번호
주소
접수일자

2. 엔티티타입 통합

엔티티타입 통합

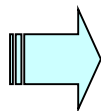
- PRIMARY KEY가 동일한 엔티티타입의 통합 - 추가속성

부동산소유자

주민등록번호
소유자명
우편번호
시도
시군구
읍면동
번지
전자메일
전화번호

부동산전세자

주민등록번호
전세자명
우편번호
시도
시군구
읍면동
번지
전자메일
전화번호



부동산관계자

주민등록번호
관계자구분
관계자명
우편번호
시도
시군구
읍면동
번지
전자메일
전화번호

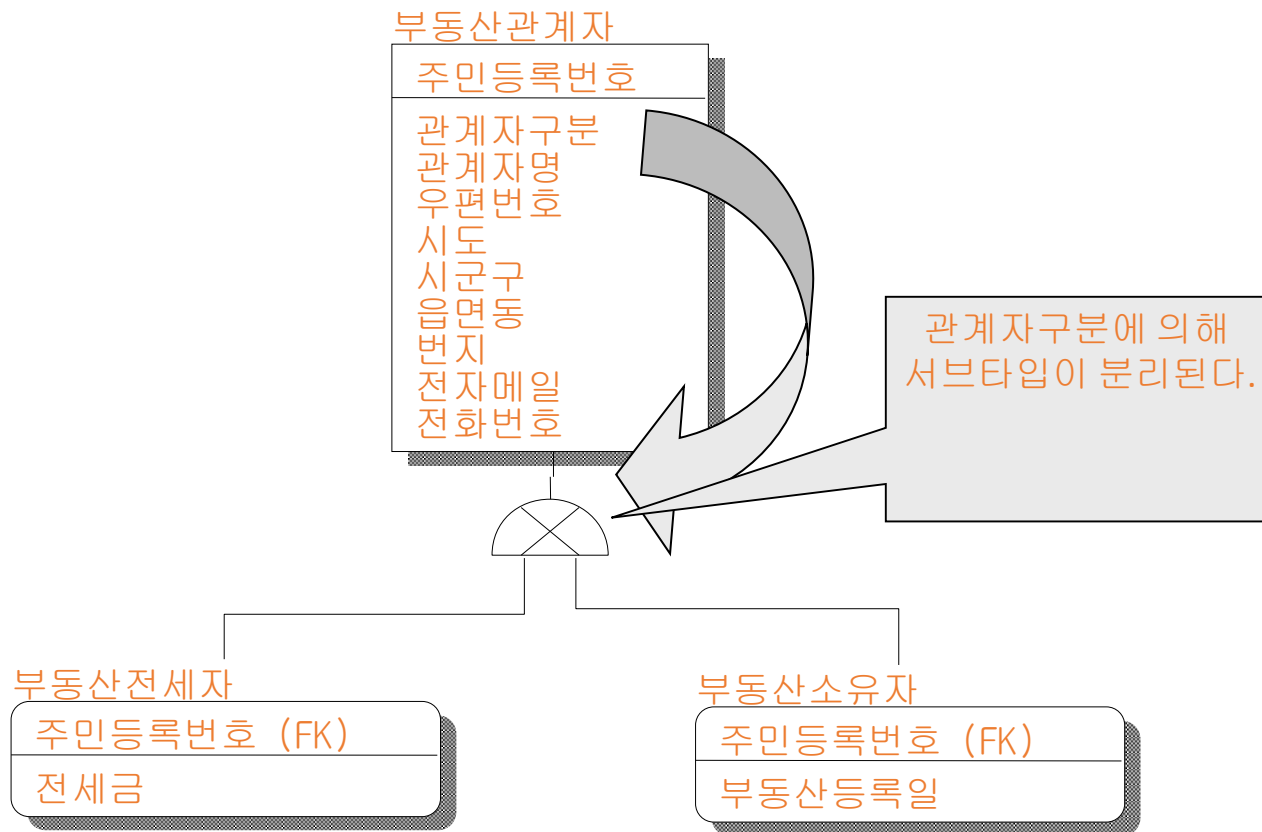
속성이 추가됨



2. 엔티티타입 통합

엔티티타입 통합

- 엔티티타입 통합 수퍼서브타입으로 통합

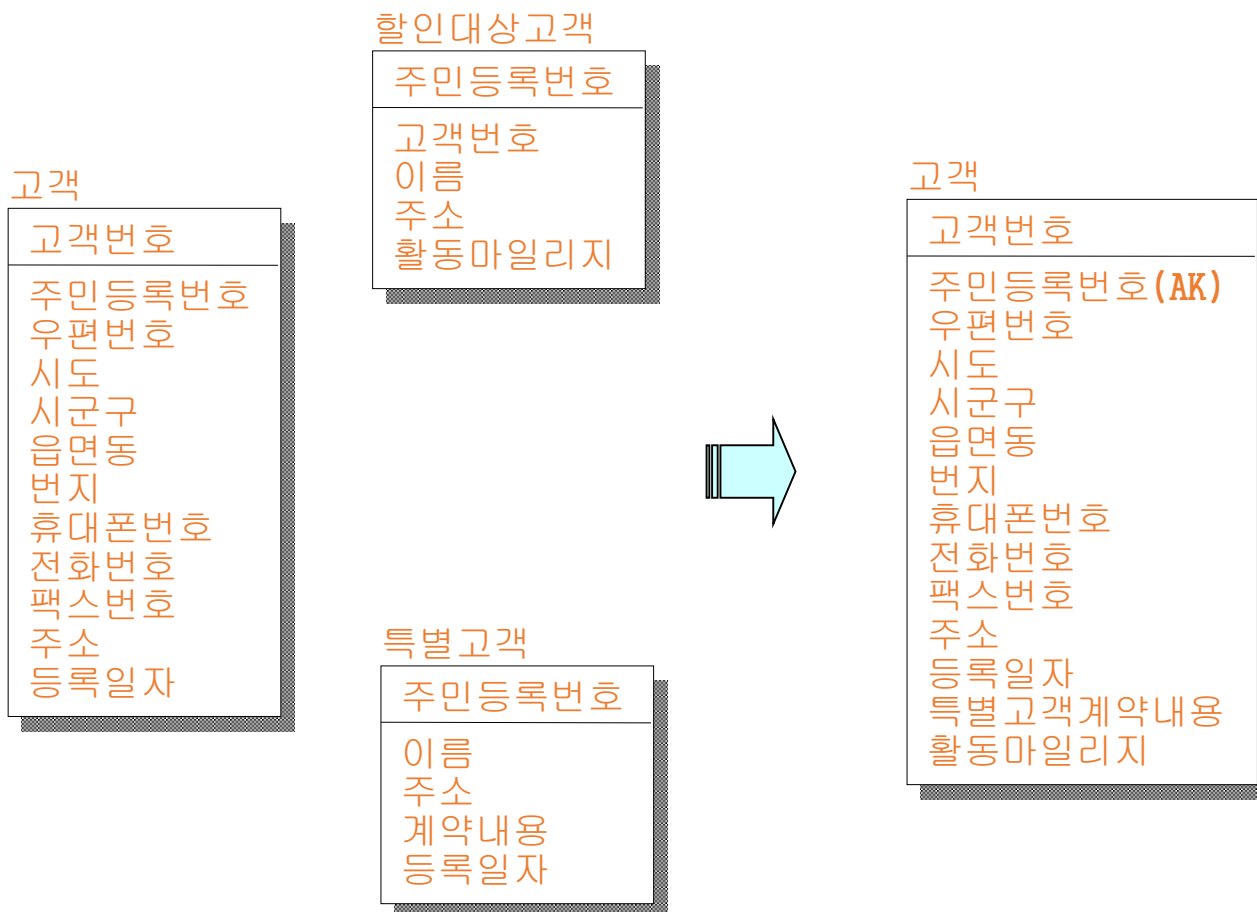




2. 엔티티타입 통합

엔티티타입 통합

- PRIMARY KEY가 비슷한 엔티티타입 통합



2. 엔티티타입 통합

엔티티타입 통합

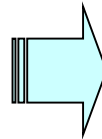
- PRIMARY KEY, 도메인, 속성이 비슷한 엔티티타입 통합

작업요청

작업요청번호
작업요청내용
작업요청일자
작업장소

작업완료

작업완료번호
작업완료내용
작업완료일자
작업장소
작업담당자



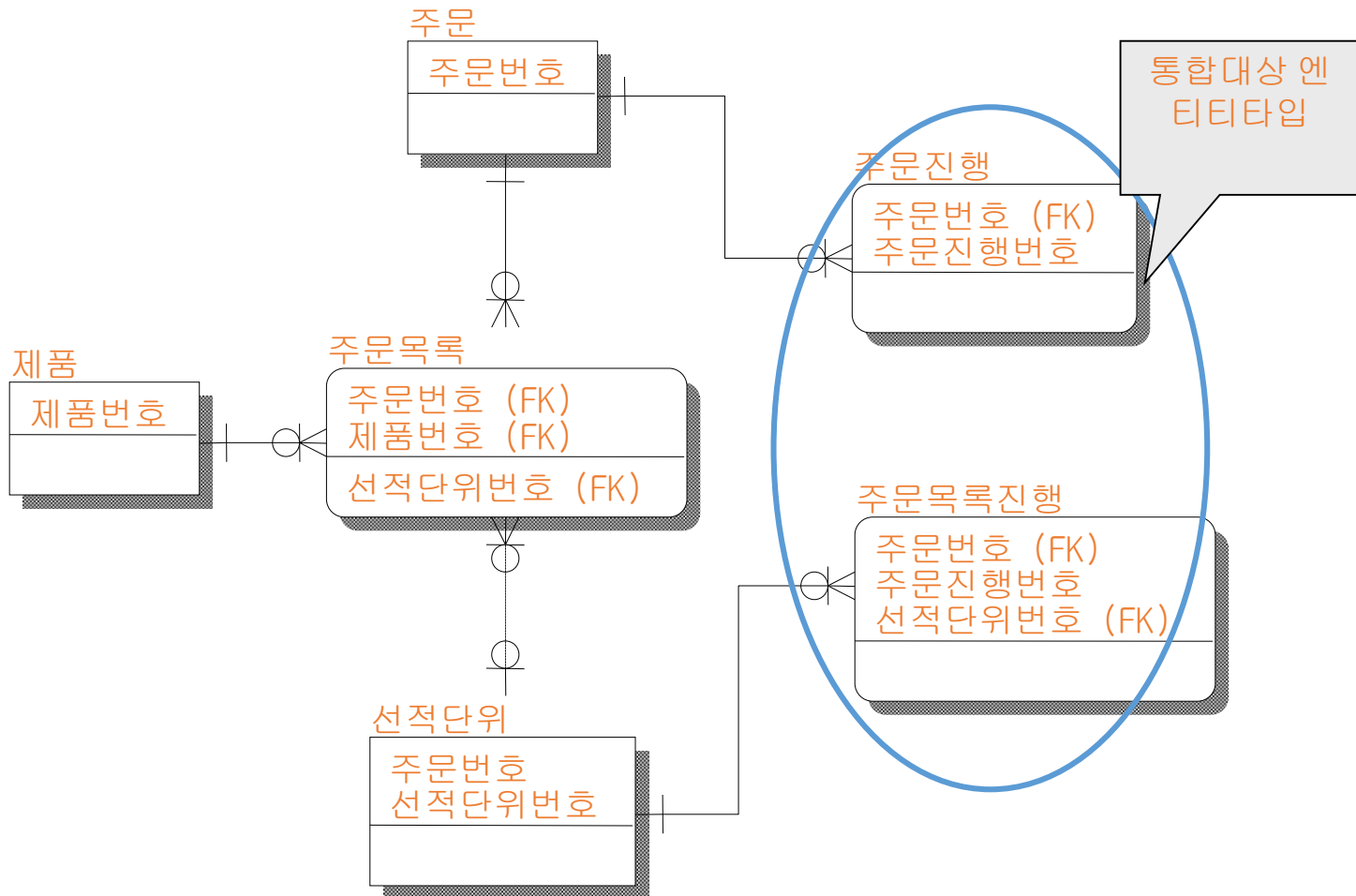
작업관리

작업번호
내용
일자
장소
작업담당자
관련작업번호 (FK)

2. 엔티티타입 통합

엔티티타입 통합

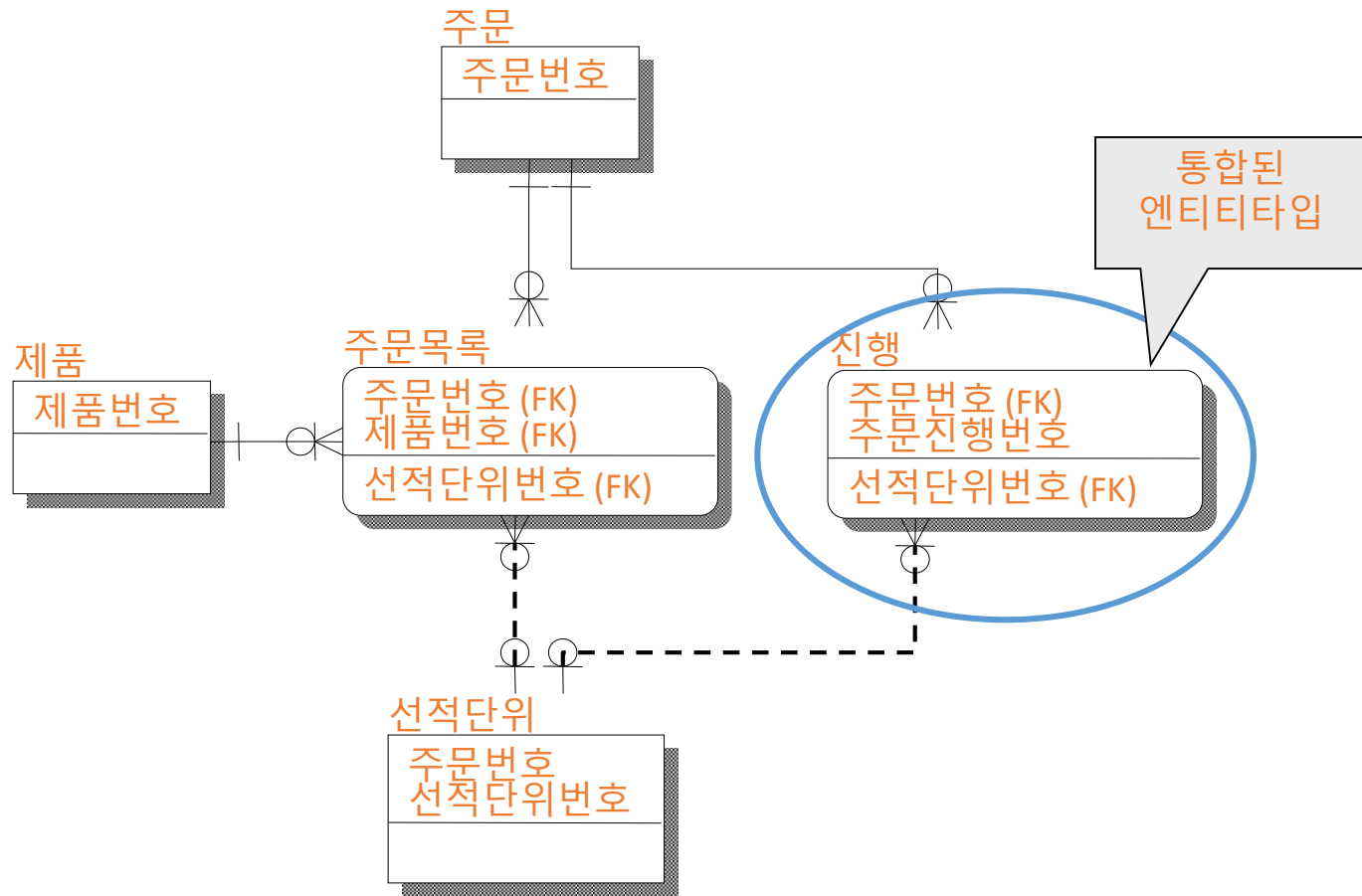
- 복합 PRIMARY KEY 구성 중 동일 속성에 의한 엔티티타입의 통합



2. 엔티티타입 통합

엔티티타입 통합

- 복합 PRIMARY KEY 구성 중 동일 속성에 의한 엔티티타입의 통합





2. 엔티티타입 통합

엔티티타입 통합 가이드

- 논리설계 단계에서는 가능한 모든 엔티티타입을 상세하게 표현
- 데이터베이스에 테이블로 설계하기 위한 물리설계 단계에서는 가능한 통합을 유도하여 표현
- 통합의 첫번째 기준은 트랜잭션이 통합하여 발생하는지 분리되어 발생하는지에 대한 사전조사 필요
- 데이터량/트랜잭션양이 많지 않은 경우 통합을 유도
- 데이터량/트랜잭션양이 많을 경우 트랜잭션의 유형에 따라 결정



3. 주요 사항에 대한 데이터모델링



주요 사항에 대한 데이터모델링 유형

- 코드 데이터모델링
- 도미노 속성 데이터모델링
- 메시지 데이터모델링
- 이력 데이터모델링

3. 주요 사항에 대한 데이터모델링

코드 데이터모델링

- 정보화시스템을 구현할 때 코드란 업무적으로 또는 정보화시스템에서 구분하기 용이하도록 간단하게 구분해놓은 단위를 말한다.

코드표현 방법 2가지

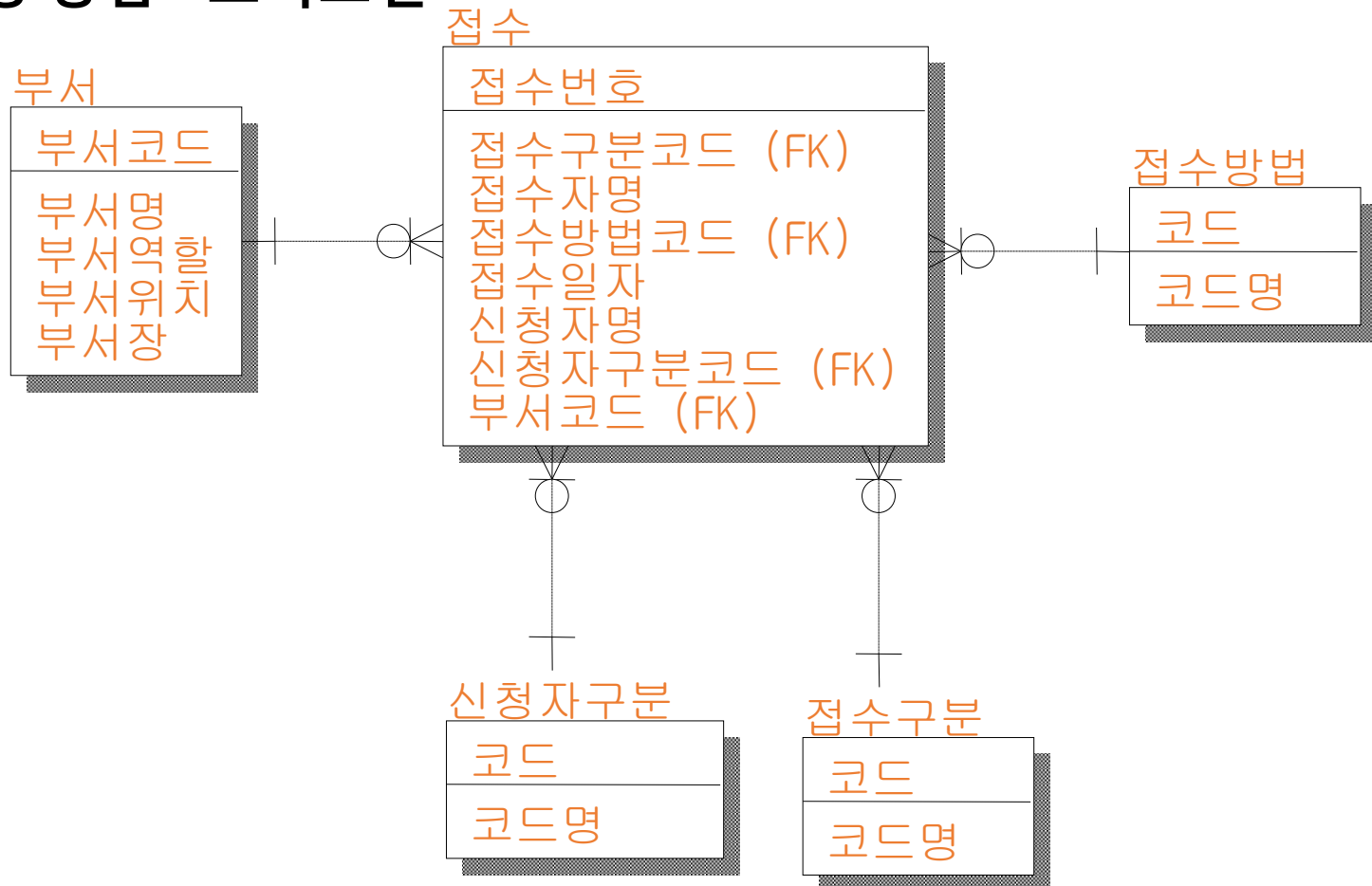
- 첫 번째로 코드에 대해 한 가지 코드의 값이 반복적으로 나타나는 경우가 있다
 - 예) 지불방법코드 {'현금', '계좌이체', '신용카드결제', '휴대폰결제'} [A001] {'01', '02', '03', '40'}
- 두 번째는 한 가지 코드에 대해 여러 개의 속성이 반복되어 나타나는 경향이 있는 코드값이 있을 수 있다.
 - 예) 영업부를 "01"로 인사부를 "02"로 총무부를 "03"으로 자재부 "04", 기술연구부를 "05"로 정의

부서코드	{ '01',	'02',	'03',	'04',	'05' }
부서장	{ '이춘식',	'김경재',	'황경빈',	'이현철',	'이동철' }
위치	{ '서울 여의도',	'서울 서초구',	'인천',	'인천',	'부산' }
역할	{ '영업',	'인사및급여',	'사무환경',	'자재구매',	'기술연구' }

3. 주요 사항에 대한 데이터모델링

코드 데이터모델링

- 코드에 대해 한 가지 코드의 값이 반복적으로 나타나는 경우 데이터 모델링 방법 - 초기모델



3. 주요 사항에 대한 데이터모델링

코드 데이터모델링

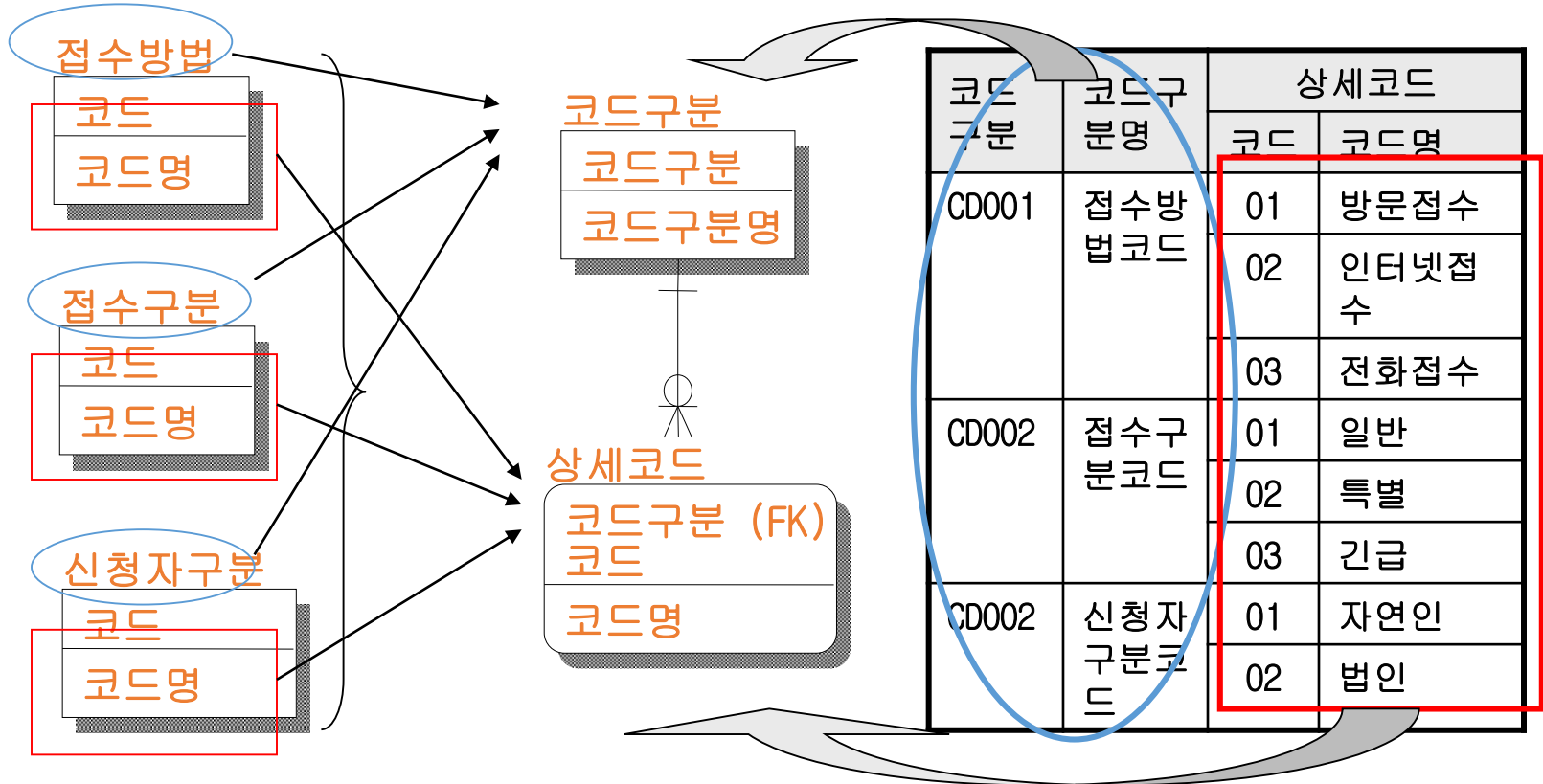
- 코드에 대해 한 가지 코드의 값이 반복적으로 나타나는 경우 데이터 모델링 방법 – 코드 값

분류번호	능력단위	상세코드	
		코드	코드명
CD001	접수방법코드	5	애플리케이션 인터페이스 구현하기
		01	방문접수
		02	인터넷접수
CD002	접수구분코드	03	전화접수
		01	일반
		02	특별
CD002	신청자구분코드	03	긴급
		01	자연인
		02	법인

3. 주요 사항에 대한 데이터모델링

코드 데이터모델링

- 코드에 대해 한 가지 코드의 값이 반복적으로 나타나는 경우 데이터 모델링 방법 - 코드 값 모델링

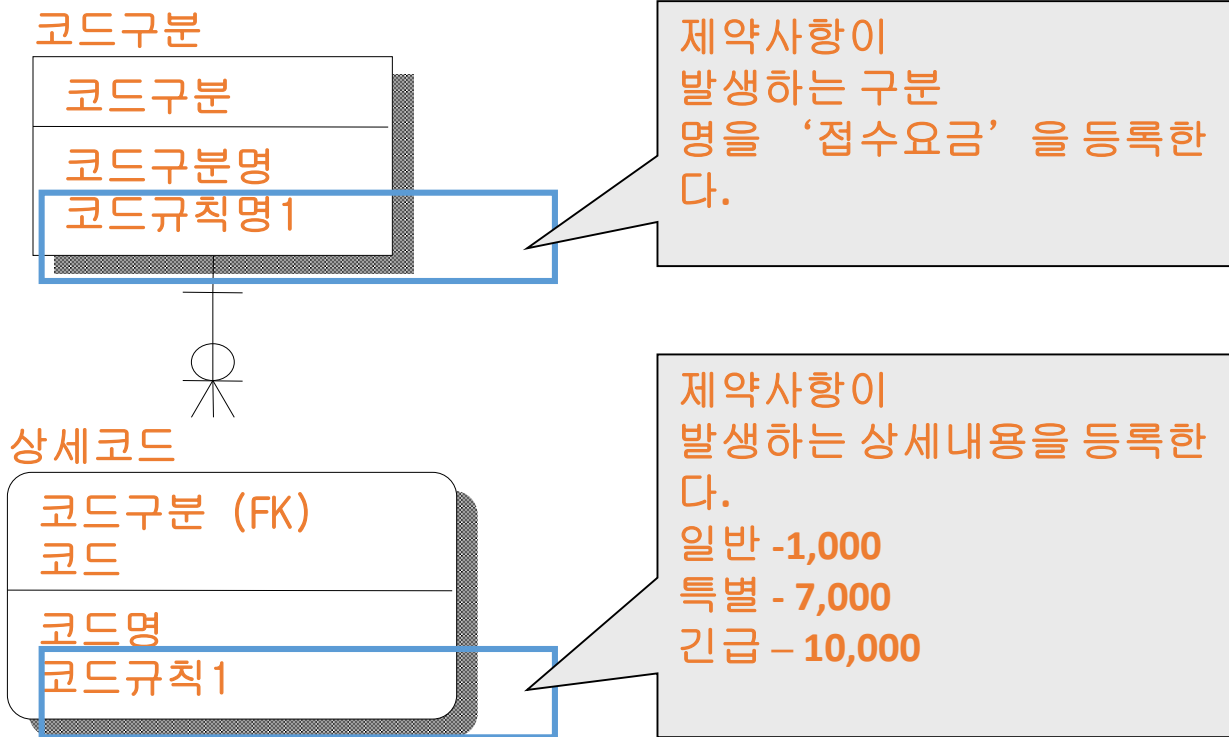


3. 주요 사항에 대한 데이터모델링

코드 데이터모델링

- 코드에 대해 한 가지 코드의 값이 반복적으로 나타나는 경우 데이터 모델링 방법 – 제약사항 추가

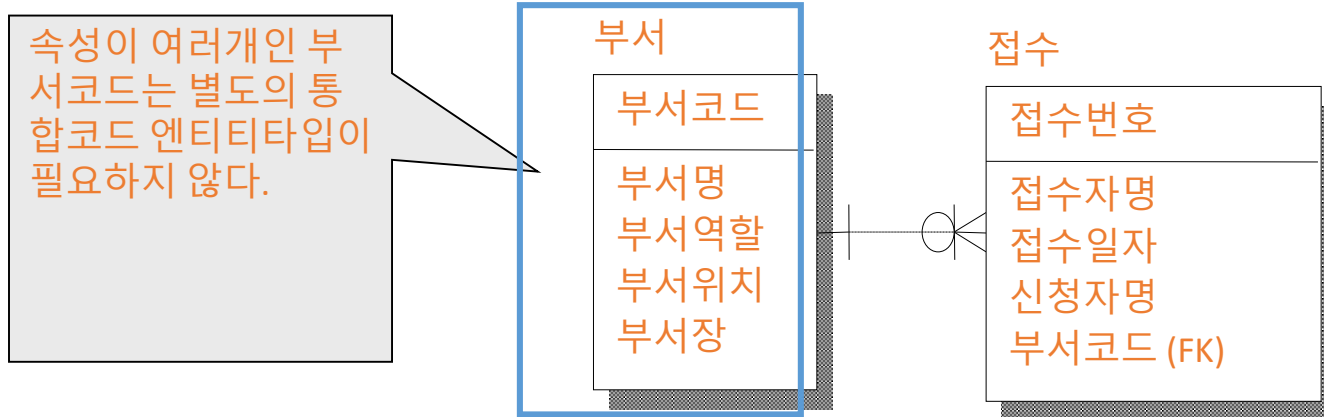
코드 엔티타입 설계방법



3. 주요 사항에 대한 데이터모델링

코드 데이터모델링

- 코드에 대해 여러 개의 코드의 값이 반복적으로 나타나는 경우 데이터모델링 방법



4. 코드값 처리 팁

코드에 대한 어플리케이션 설계 TIP

● 코드의 특징

- ① 코드값은 잘 변하지 않음
- ② 데이터의 크기가 크지 않음
- ③ 많은 엔티타입에서 참조하고 있음
- ④ 많은 테이블에서는 코드테이블을 코드 개수 만큼 다중 조인이 발생

● 코드값 변환을 잘못하는 대표적으로 세 가지 유형

- ① 첫 번째 모든 코드변환을 데이터베이스가 있는 위치에서 조인에 의해 변환- SQL구문이 복잡해짐, 가독성 저하
- ② 두 번째는 SQL문장내에서 DECODE를 사용하여 코드와 코드값을 매핑(MAPPING)- 코드값변환에 대한 별도의 작업(로직에서)은 필요하지 않지만 만약 코드가 변경되거나 추가라도 된다면 모든 SQL을 수정해야 하므로 코드가 변경될 때마다 관련 소스를 찾아 재 컴파일이 발생해야 함
- ③ 세 번째는 응용어플리케이션에서 코드와 코드값을 변환- 소스에 코드값을 기술하는 방법으로 역시 코드가 변경될 때마다. 모든 소스에 대해 수정과 재 컴파일- 발생해야 하는 치명적인 결함을 가지고 있음



4. 코드값 처리 팁

코드에 대한 어플리케이션 설계 TIP

- 코드값 처리를 위한 효율적인 방법
 - ① 첫 번째는 DBMS Function을 이용한 코드변환
 - ② 두 번째는 응용 어플리케이션의 메모리에 공통코드 캐쉬 후 변환을 사용하는 것이 효율적임

5. 도미노 속성

도미노 속성?

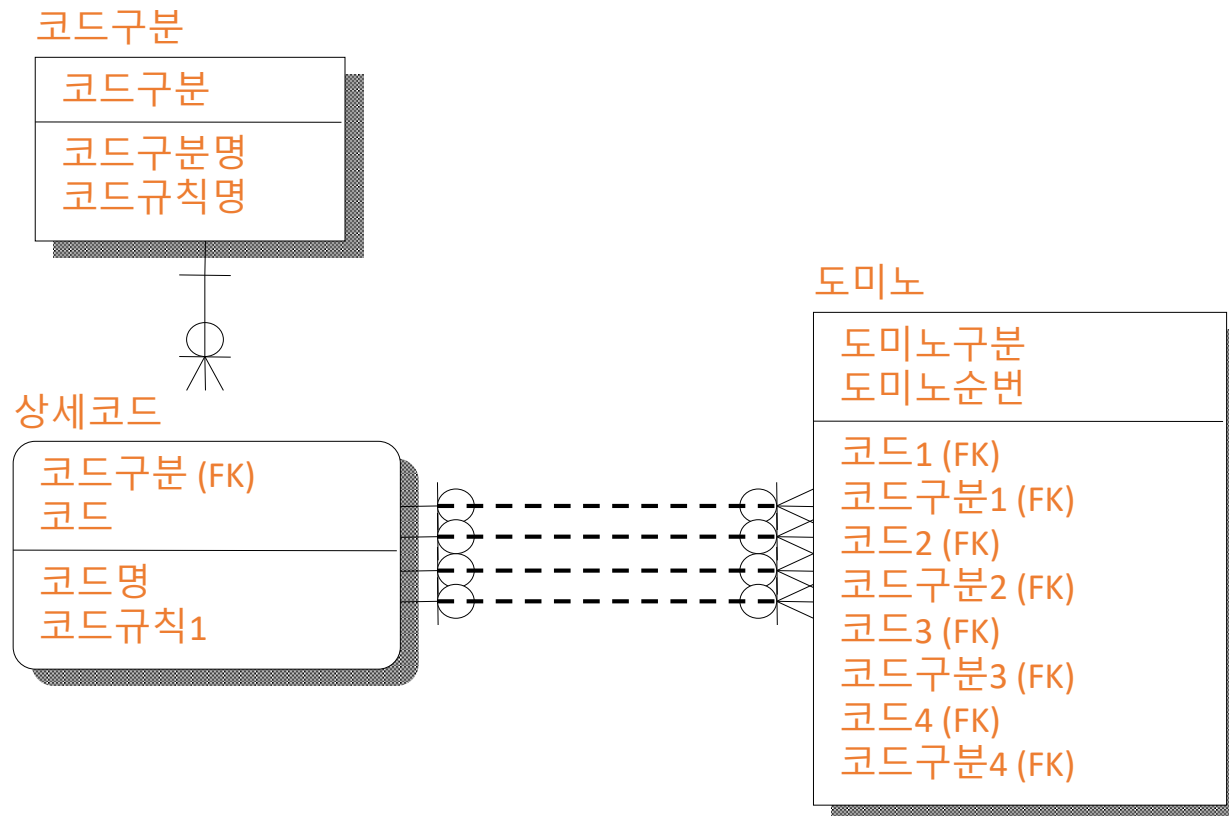
- 앞의 값에 규칙적인 제약이 연쇄적으로 발생하는 경우를 도미노속성이라 한다.

도미노 구분	순 번	도미노1		도미노2		도미노3		도미노4	
		코드 구분	코드	코드 구분	코드	코드 구분	코드	코드 구분	코드
DOM01	1	CD001	A	CD013	1001	CD201	01	CD723	OA
	2	CD001	A	CD013	1001	CD201	01	CD723	OA
	3	CD001	A	CD013	1001	CD201	02	CD723	CV
	4	CD001	A	CD013	1002	CD201	02	CD723	FS
	5	CD001	B	CD013	1001	CD201	01	CD723	OA
DOM02	1	CD002	01	CD783	AA	CD227	9	CD107	A33
	2	CD002	01	CD783	AA	CD227	4	CD107	B22
	3	CD002	02	CD783	AA	CD227	3	CD107	B22
	4	CD002	02	CD783	BB	CD227	3	CD107	B22
	5	CD002	02	CD783	AA	CD227	3	CD107	C55

5. 도미노 속성

도미노 속성 데이터모델링

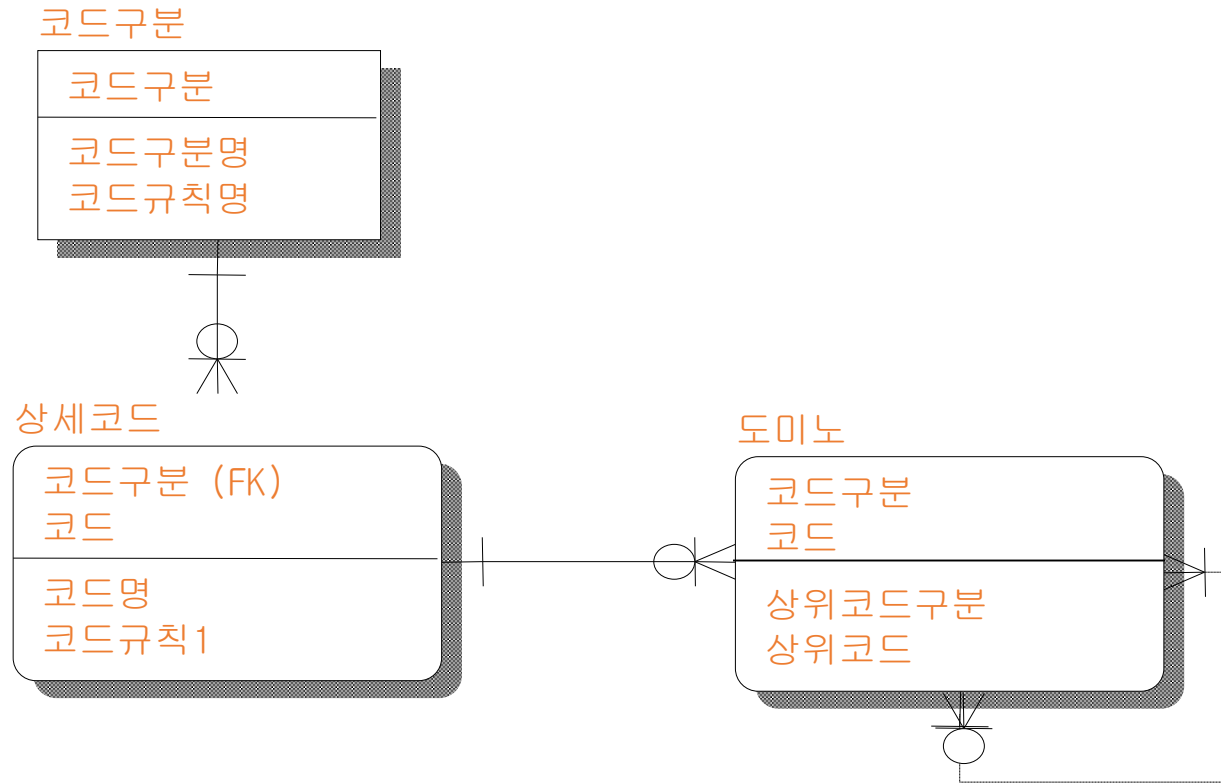
- 도미노속성 데이터모델 - 속성이용



5. 도미노 속성

도미노 속성 데이터모델링

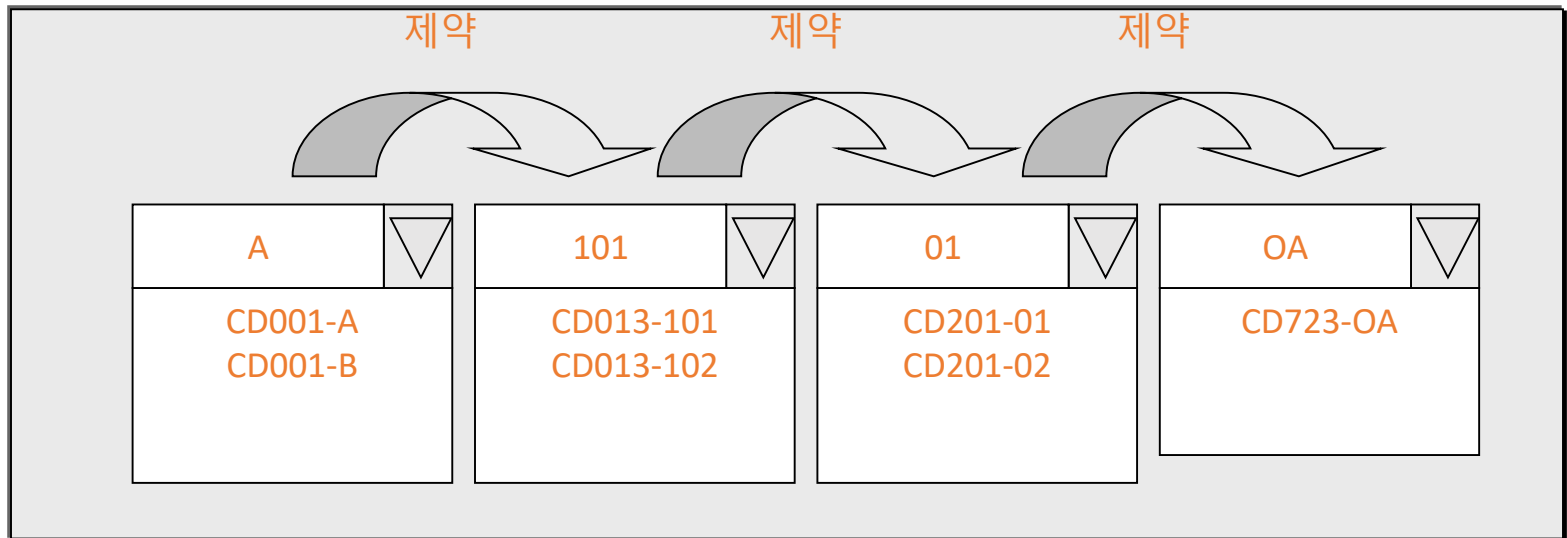
- 도미노속성 데이터모델 - BOM(BILL OF MATERIAL)이용



5. 도미노 속성

도미노 속성 활용

- 도미노속성과 USER INTERFACE





6. 메시지

메시지

- 메시지는 사용자가 시스템을 조작하면서 받는 정보시스템의 반응에 해당된다.

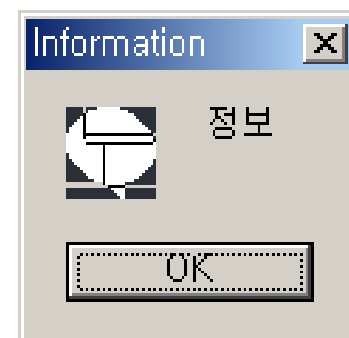
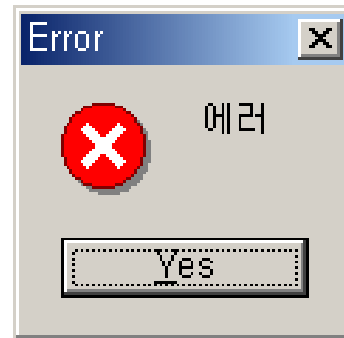
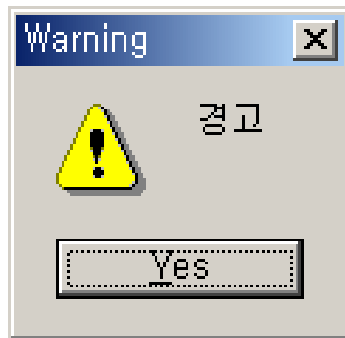
메시지처리 방법 세가지

1. 첫 번째 방법은 모든 개발소스에 메시지를 코딩하여 개발하는 방법
2. 두 번째 방법으로는 헤더파일이나 클라이언트 테스트파일에 메시지를 정의하고 어플리케이션에서 임포트하여 사용하는 방법
3. 세 번째는 데이터베이스에 저장하고 모든 개발자가 데이터베이스를 통해서 메시지의 내용을 가져와 보여주는 방법이 있을 수 있다.

6. 메시지

메시지종류

- 유형



저장되었습니다.

6. 메시지

메시지 조사

● 유형

메시지 번호	메시지 유형코드	메시지내용	응답 번호
E001	E	접수일자 입력오류입니다.	
E002	E	접수번호가 존재하지 않습니다.	A003
I001	I	금청구서가 발행되었습니다.	
I002	I	승인하였습니다.	
W001	W	청구일자가 주문일자보다 이전입니다.	
W002	W	선택된 고객이 없습니다.	
S001	S	ORACLE CONNECT 에러	A001
S002	S	인사시스템으로 데이터 전송 오류가 발생하였습니다.	A001
S003	S	인사시스템으로부터 데이터 수신오류가 발생하였습니다.	A001
A001	A	전산실로 문의하십시오.	
A002	A	작업을 계속하시겠습니까?	

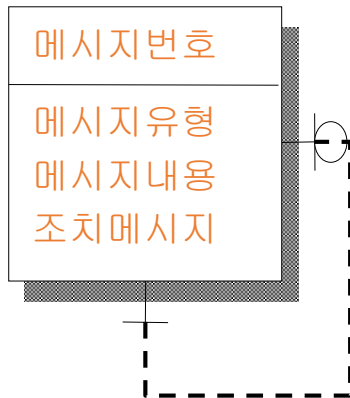
6. 메시지

메시지 모델링

● 방법

방법1. 하나의 엔티타입으로 설계

메시지

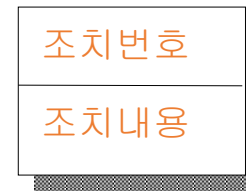


방법2. 응답을 분리하여 설계

메시지



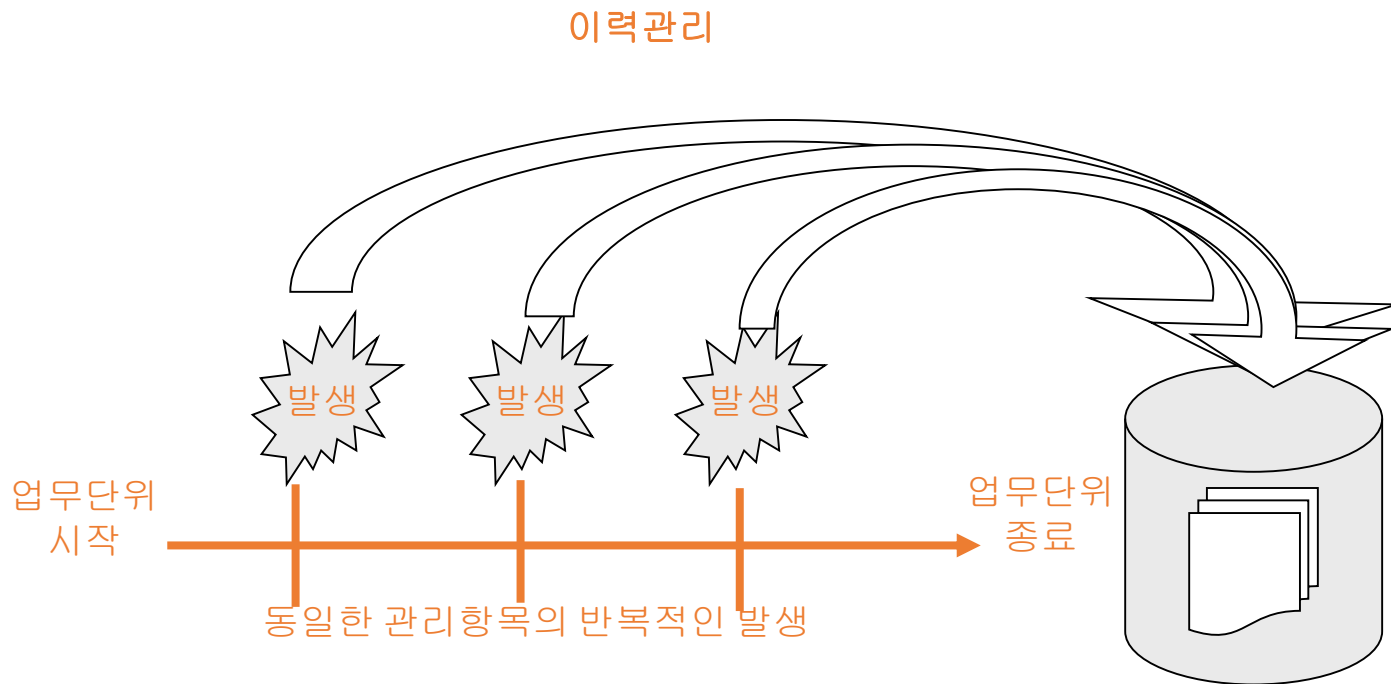
조치



7. 이력관리

이력관리

- 하나의 업무단위가 시간흐름에 따라 발생하는 과거와 현재 데이터를 지속적으로 유지하는 관리방법을 이력(HISTORY)관리라 한다.





7. 이력관리

이력의 세가지 유형

- 유형

변경이력

주문변경
계약변경

발생이력

급여계산
요금청구

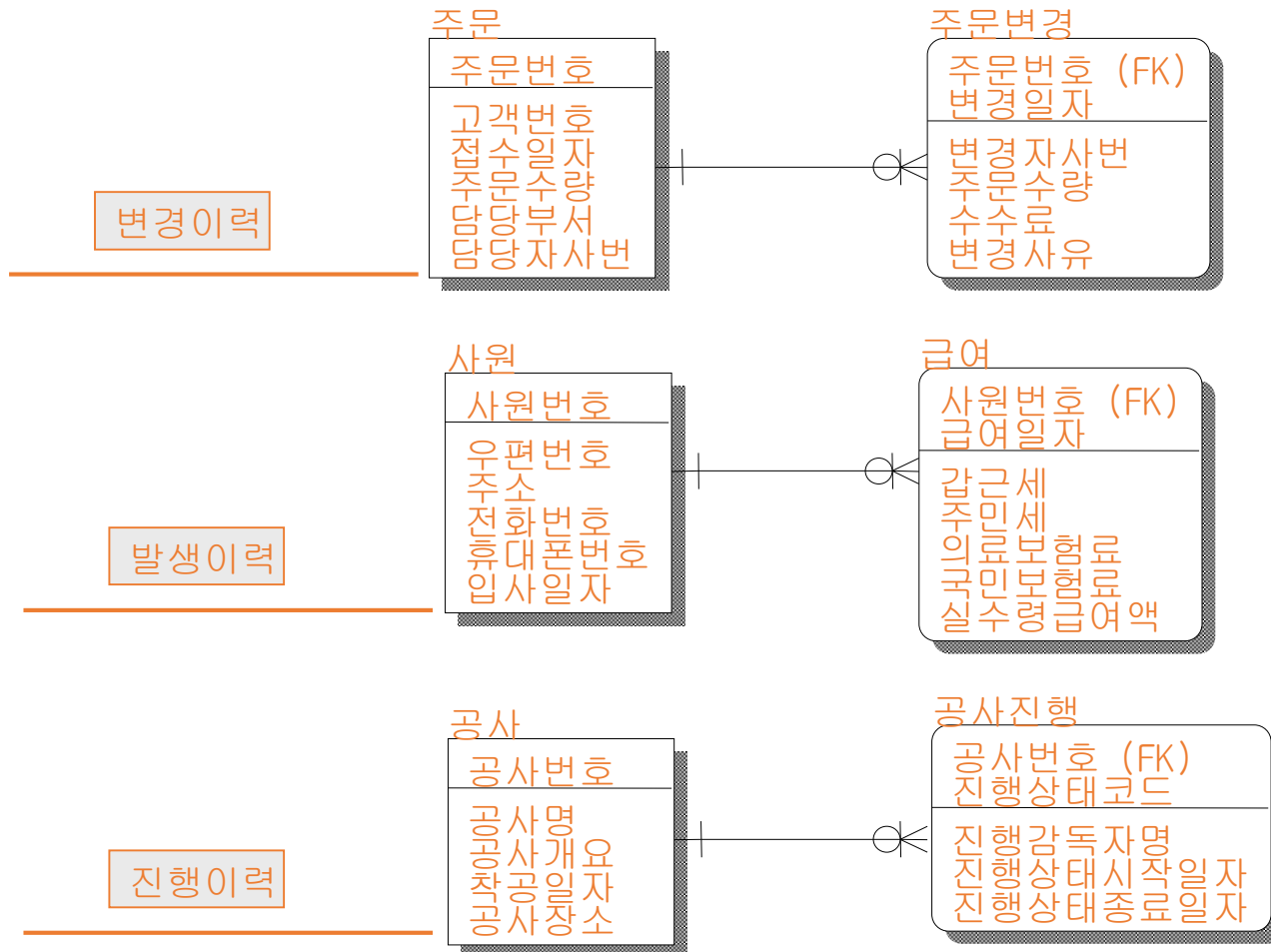
진행이력

공사진행
접수진행

7. 이력관리

이력의 세가지 유형

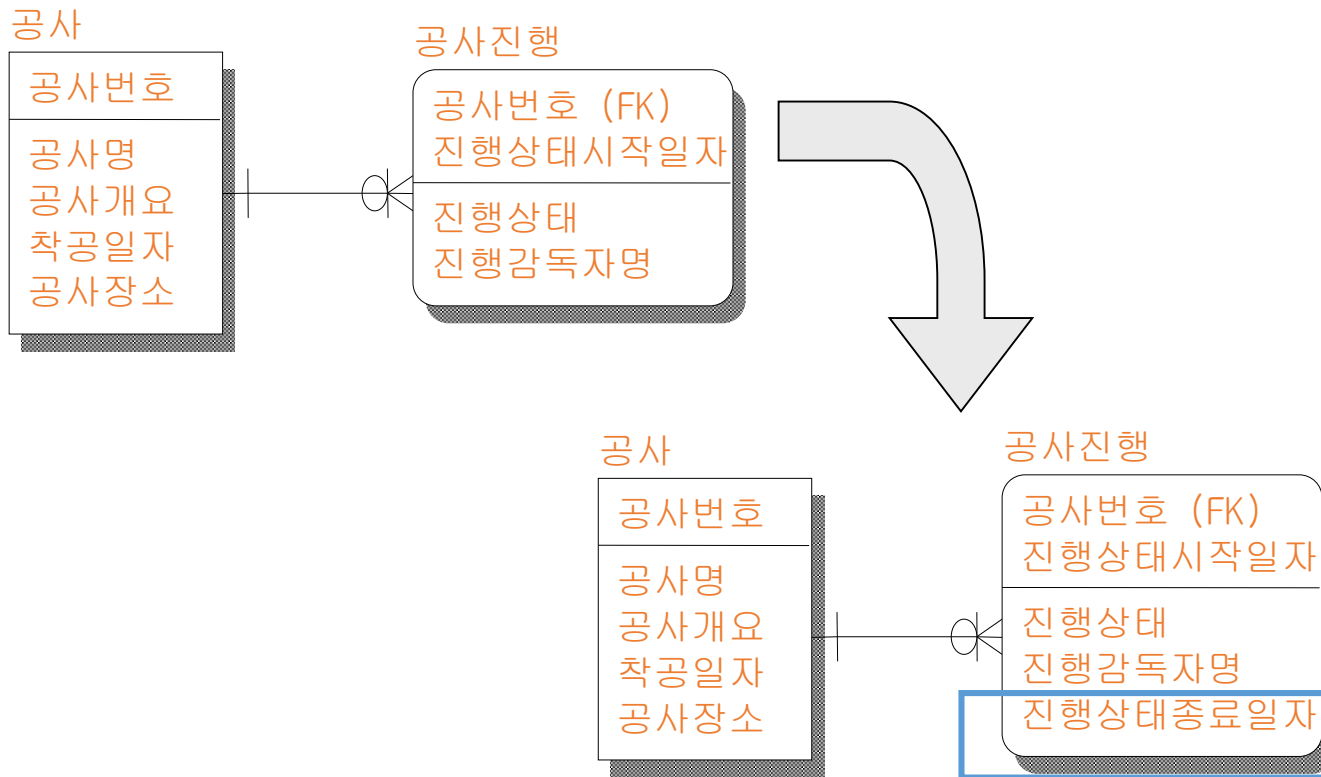
- 엔티티타입이 분리된 이력 데이터모델 유형



7. 이력관리

이력의 세가지 유형

- 엔티티타입이 분리된 이력 데이터모델 유형



이력관리-진행이력

● 진행상태 종료일자가 없는 경우

```
SELECT 진행시작1.공사번호, 진행시작1.진행상태
FROM 공사진행 진행시작1, (SELECT 공사번호, MAX(진행상태시작일자) 진행상태시작일자
                           FROM 공사진행
                           WHERE 진행상태시작일자 <= '2002년 7월 15일'
                           GROUP BY 공사번호) 진행시작2
WHERE 진행시작1.공사번호 = 진행시작2.공사번호
  AND 진행시작1.진행상태시작일자 = 진행시작2.진행상태시작일자
```

● 진행상태 종료일자가 있는 경우

```
SELECT 공사진행.공사번호, 공사진행.진행상태
FROM 공사진행
WHERE 공사진행.진행상태시작일자 <='2002년 7월 15일'
  AND 공사진행.진행상태종료일자 > '2002년 7월 15일'
```

7. 이력관리

이력관리-변경이력

- 사업부코드 '1001'에 대해서 가장 최근에 변경된 주문번호, 주문수량을 가져오는 SQL문장

변경이력 데이터조회

주문변경

주문번호
변경일자
변경자사번
주문수량
수수료
변경사유
사업부코드

```
SELECT 주문변경1. 주문번호, 주문변경1.주문수량
FROM 주문변경 주문변경1,
    (SELECT 주문번호, MAX(변경일자) 변경일자
     FROM 주문변경
     WHERE 사업부코드 = '1001'
     GROUP BY 주문번호) 주문변경2
WHERE 주문변경1.주문번호 = 주문변경2.주문번호
AND 주문변경1.변경일자 = 주문변경2.변경일자
```

7. 이력관리

이력관리-변경이력

- 사업부코드 '1001'에 대해서 가장 최근에 변경된 주문번호, 주문수량을 가져오는 SQL문장

변경이력 데이터조회

주문변경

주문번호
변경일자
변경자사번
주문수량
수수료
변경사
사업부코드
최신여부

```
SELECT 주문변경. 주문번호, 주문변경.주문수량  
FROM 주문변경  
WHERE 주문변경.사업부코드 = '1001'  
AND 주문변경.최신여부 = 'Y'
```


『1과목』 Fundamental

6교시 :

프로세스/상관 모델링



1. 프로세스 모델링

1. 프로세스 모델링이란?

- 정보화 시스템을 구축하기 위해, 어떤 일이 그 업무에서 행하여지고 있는지 또는 어떤 일을 그 업무에서 해야 하는지를 분석하는 방법이다.

프로세스

- 프로세스는 실행함으로써 입력과 출력이 수행되는 단위





2. 업무기능 분해

1. 업무기능 분해 정의

- 업무기능분해란 기업의 업무를 정보화 시스템이 구축될 수 있는 단위로 쪼개는 것이다.

2. 업무기능 분해 목적

- 첫번째 정보화 시스템을 구축하는 가장 최저 단위의 단위 프로세스를 도출하기 위함이다.
- 두번째 전체 업무의 체계를 프로세스를 중심으로 도식화하여 업무에 대한 기능을 파악하는 것이다.
- 단위프로세스는 실제 어플리케이션에서 로직이 구현된 단위이며 데이터베이스에도 영향을 미치는 가장 작은 단위의 업무프로세스이다.



2. 업무기능 분해

2. 업무기능과 업무프로세스

- 업무기능
 - 업무기능이란, 기업의 임무에 대해 한층 더 멀리 조명하여 하나로 표현하는 완벽한 단위업무기능의 그룹이다.
 - 대부분의 기업의 조직에서는 주요한 업무기능은 영업, 회계, 제조 등 5에서 10개의 업무기능으로 구분이 된다.
 - 업무기능에 대한 설명을 할 때는 업무기능이 무엇을 하는지 또 왜 하는지를 반드시 설명해야 한다.
 - 누가, 언제, 어디서 어떤 방법으로의 관한 내용은 서술하면 안된다.
- (예) 기업의 조직에서는 주요한 업무기능은 영업, 회계, 제조 등 5에서 10개의 업무기능으로 구분된다.



2. 업무기능 분해

2. 업무기능과 업무프로세스

- 업무프로세스
 - 프로세스란 입력과 출력이 존재하는 업무기능으로 프로세스를 실행함으로써 데이터의 입력과 출력이 수행된다.
 - 프로세스임을 판단하는 근거 첫번째는 해당 업무기능이 실행단위인지 관리단위인지를 판단한다.
 - 두 번째로는 업무기능에 입력과 출력의 행위가 이루어지는 지를 확인한다.
- (예) “접수하다”라는 하나의 프로세스를 실행하면 접수라는 행위에 의해 접수정보가 입력되어 접수엔티티타입에 접수정보가 생성되게 된다. “접수하다”라는 업무기능은 실행할 수 있는 단위이고 실행에 의해 접수정보가 받아들여(INPUT) 접수 엔티티타입에 출력 정보인 접수정보를 생성함으로써 즉 실행에 의해 입력과 출력이 수행되었으므로 프로세스가 되는 것이다.



2. 업무기능 분해

3. 업무기능분해의 소스

- 업무기능분해 소스(SOURCE)

 1. 해당업무의 구체화하여 기능분해

 2. 업무기능의 생명주기(LIFE CYCLE)을 분석하여 기능분해

 3. 엔티티의 상태를 체크하여 기능분해

 4. 업무이벤트에 따라 기능분해

 5. 데이터의 구조에 따라 기능분해



2. 업무기능 분해

1. 기능 분할

● 시스템과 기능

1. 시스템이란?

- 어떤 기능을 하기 위해 서로 어떤 작용을 주고받는 두 가지 이상의 부품으로 구성된 것
- 예) 망치 - 물체를 치는 기능을 위해 머리와 손잡이로 구성된 시스템
- 예) 문 - 열고 닫는 기능을 위해 몸체와 손잡이, 회전축, 지지대, 경첩 등으로 구성된 시스템

2. 기능(Function)이란?

- 기능이란 시스템이 하는 일, 즉 시스템의 ‘역할’을 뜻한다.

시스템	주요 기능
톱	물체를 자르는 역할
자동차	사람이나 물건을 이동시키는 역할
그릇	음식을 담는 역할
양산	빛을 차단시키는 역할



2. 업무기능 분해

1. 기능 분할

- 시스템에서 발생하는 모든 문제는 시스템 또는 부품의 문제가 아닌 기능의 문제
- 사람들이 구입하는 것 또한 제품 자체라기보다는 그 제품의 기능
- 시스템의 문제를 해결하기 위해서는 먼저 시스템을 기능으로 보는 연습 필요
- 기능 정의 방법
- 기능을 수행하는 주체인 도구(주어), 작용을 받는 대상(목적어), 작용 (동사)을 모두 포함해야 함

예) 자동차는 사람을 이동시키는 역할을 한다.

주어

목적어

동사



2. 업무기능 분해

1. 기능 분할

- 기능 정의 사례1
- 선풍기의 기능 정의

잘못된 기능 정의

선풍기는 사람을 시원하게 하는 역할을 한다.

주체인 선풍기가 대상인 사람을 어떻게 직접적으로 시원하게 변화시키는지 명확하지 않으므로 정확한 표현이 아님

올바른 기능 정의

선풍기는 주변 공기를 움직여서 공기의 온도를 낮추는 역할을 한다.

주체인 선풍기가 대상인 사람이 시원하게 느끼도록 하는데 어떤 역할을 하는지가 분명하게 표현됨



2. 업무기능 분해

1. 기능 분할

● 기능 정의 사례₂

시스템	잘못된 정의	올바른 정의
레이저 포인터	레이저 포인터는 발표 자료를 가리키는 역할을 한다.	레이저 포인터는 사람의 시선을 한 곳에 집중시키는 역할을 한다.
에어컨의 공기 필터	공기필터는 공기를 정화시키는 역할을 한다.	공기필터는 먼지를 잡는 역할을 한다.
진공 청소기	진공 청소기는 먼지를 제거하는 역할을 한다.	진공 청소기는 공기를 이동시키는 역할을 한다.(공기를 이동시키면 진공 상태가 되고, 그 결과 먼지가 다른 곳으로 이동하게 된다.)
칫솔	칫솔은 치아를 깨끗하게 하는 역할을 한다.	칫솔은 치아 표면이나 치아 사이의 이물질을 제거하는 역할을 한다.



2. 업무기능 분해

1. 기능 분할

- 기능 분석(Function Analysis)이란?
 - 시스템을 구성하는 부품, 환경 요소, 대상 간의 역할 관계를 분석하는 것
 - 즉, 시스템의 본질을 파악하기 위해 시스템 및 구성 부품의 진짜 기능이 무엇인지 분석하는 것
- 1. 부품 - 시스템을 구성하는 구성 요소, 하위 시스템(subsystem)
 - 예) 자동차의 부품 : 프레임, 엔진, 바퀴 등
- 2. 환경 요소 - 시스템에 포함되지는 않지만 시스템에 영향을 주는 모든 것
 - 상위 시스템(supersystem)
 - 예) 자동차의 환경 요소 : 도로, 연료, 매연 등
- 3. 대상 - 시스템을 이용해서 특성을 변화시키고 싶은 물체
 - 예) 자동차의 대상 : 사람, 물건 등



2. 업무기능 분해

1. 기능 분할

- 기능 분석(Function Analysis)이란?
 - 시스템을 구성하는 부품, 환경 요소, 대상 간의 역할 관계를 분석하는 것
 - 즉, 시스템의 본질을 파악하기 위해 시스템 및 구성 부품의 진짜 기능이 무엇인지 분석하는 것
- 1. 부품 - 시스템을 구성하는 구성 요소, 하위 시스템(subsystem)
 - 예) 자동차의 부품 : 프레임, 엔진, 바퀴 등
- 2. 환경 요소 - 시스템에 포함되지는 않지만 시스템에 영향을 주는 모든 것
 - 상위 시스템(supersystem)
 - 예) 자동차의 환경 요소 : 도로, 연료, 매연 등
- 3. 대상 - 시스템을 이용해서 특성을 변화시키고 싶은 물체
 - 예) 자동차의 대상 : 사람, 물건 등

2. 업무기능 분해

1. 기능 분할

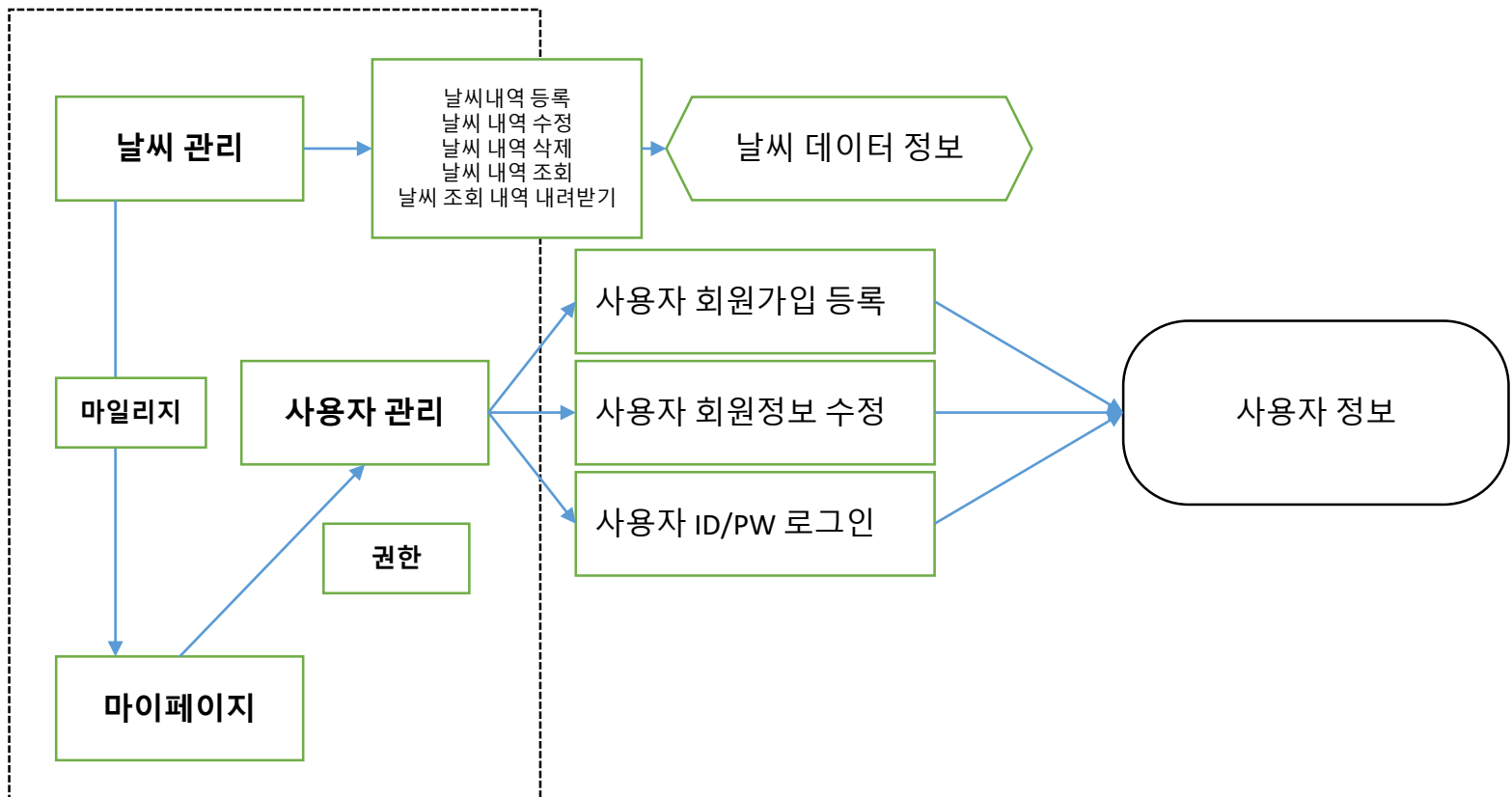
● 기능 분석도 작성 기호

구분	기호	
시스템 및 부품	직사각형	
대상(객체)	둥근 사각형	
환경 요소(상위 시스템)	육각형	
유용한 기능	실선 화살표	
유해한 기능	점선 화살표	
상세 기능 분석도의 시스템	점선 사각형	

2. 업무기능 분해

1. 기능 분할

● 기능 분석도 작성 기호



3. 업무기능분해의 소스

기능분해 - 업무구체화

● 사례

공사관리

아파트공사

도로공사

하수도공사

보험사고관리

자동차사고처리

화재사고처리

질병사고처리



3. 업무기능분해의 소스

기능분해 - 생명주기분석

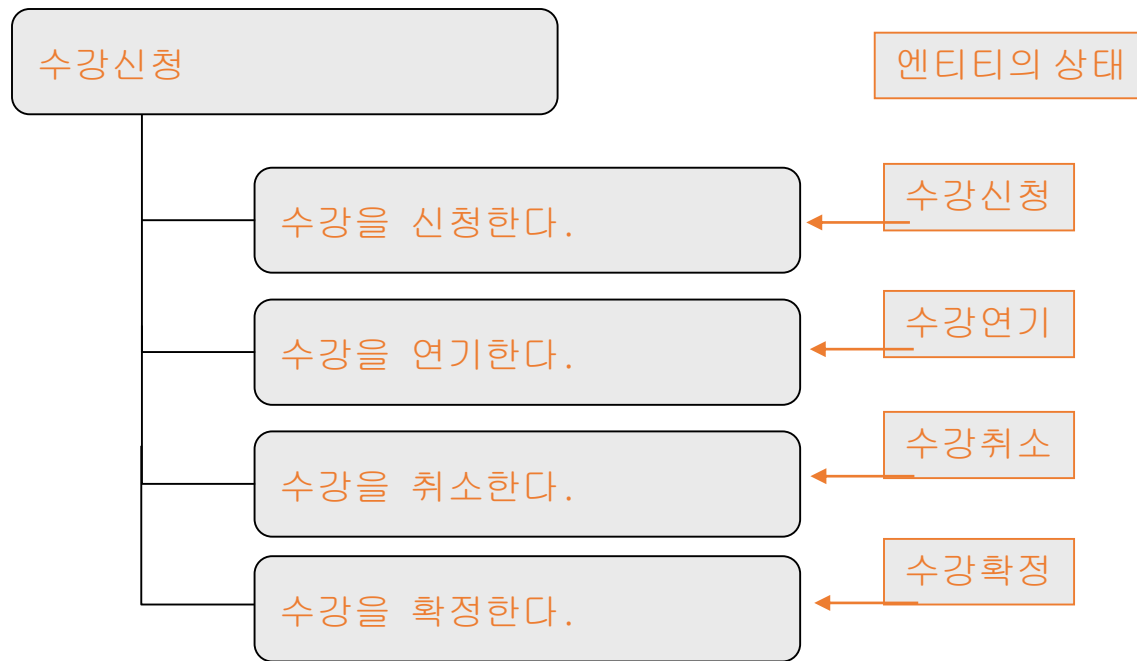
● 사례



3. 업무기능분해의 소스

기능분해 - 엔티티상태

● 사례

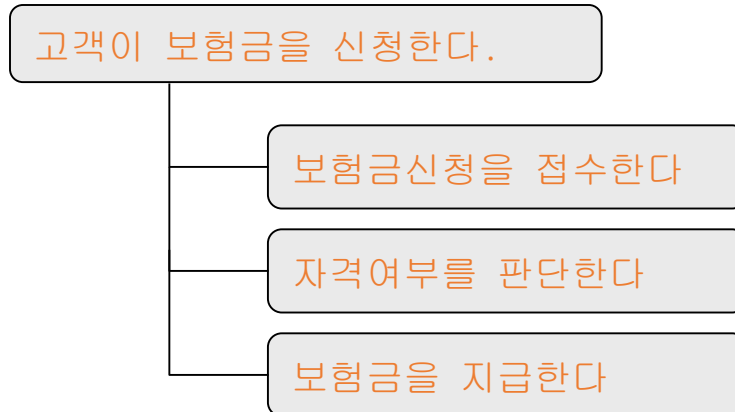




3. 업무기능분해의 소스

기능분해 - 업무이벤트

● 사례

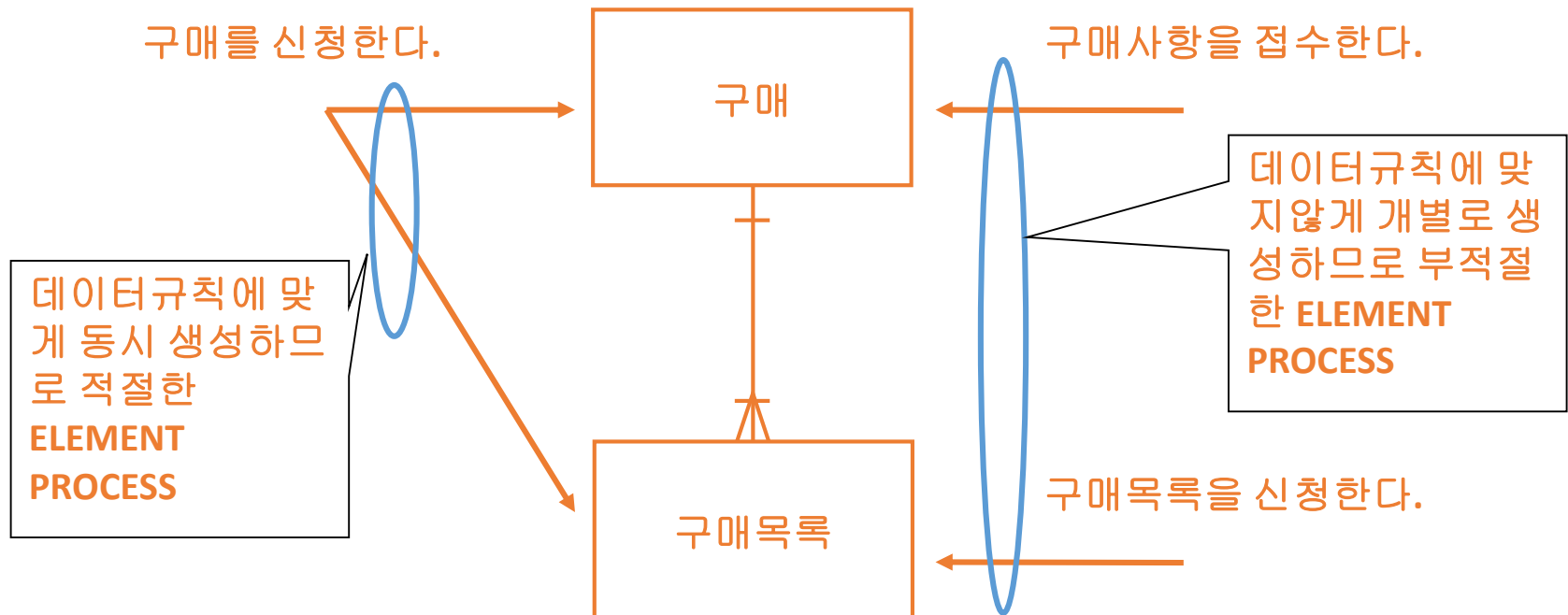


4. 트랜잭션

트랜잭션이란...

- 트랜잭션이란 논리적인 작업의 단위(**LOGICAL UNIT OF WORK**)이다.

업무데이터 무결성 보장



3. 단위 프로세스

단위 프로세스란?

- 단위프로세스 도출의 최저 단위는 트랜잭션을 보장하기 위한 단위이다.
- 단위프로세스는 한 시점에 한 사람이 한 지점에서 작업을 수행하는 단위이다.
- 단위 프로세스 규칙

규칙 1.

단위프로세스가 실행되면 완전하고 의미 있는 결과가 나와야 한다.

규칙 2.

형제(SIBLING) 프로세스는 비슷한 프로세스가 발생된다.

규칙 3.

단위 프로세스는 하나 이상의 응답이 존재한다.

규칙 4.

단위 프로세스는 반드시 데이터와 관련해서 수행된다.

3. 단위 프로세스

단위 프로세스란?

- 고객이 주문하는 프로세스에 대한 업무기능분해도 초기 모습



3. 단위 프로세스

단위 프로세스란?

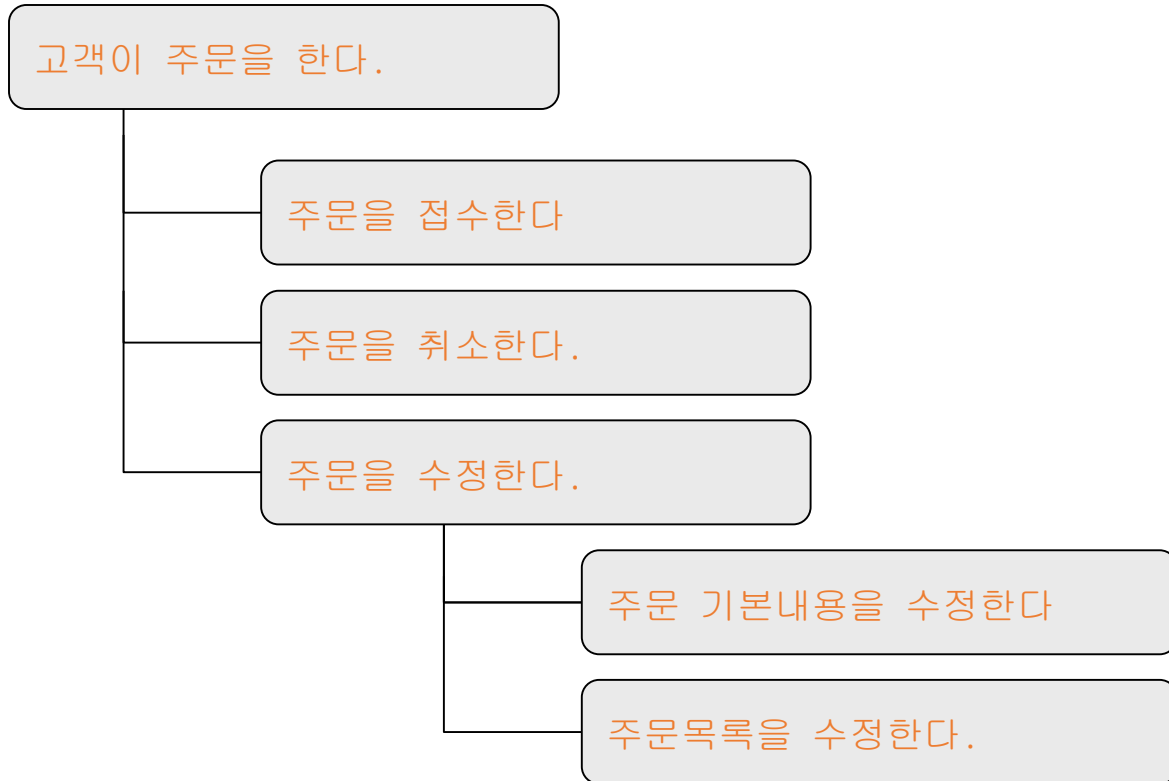
- 주문과 주문목록에 대한 부분 ERD



3. 단위 프로세스

정정된 업무기능분해도

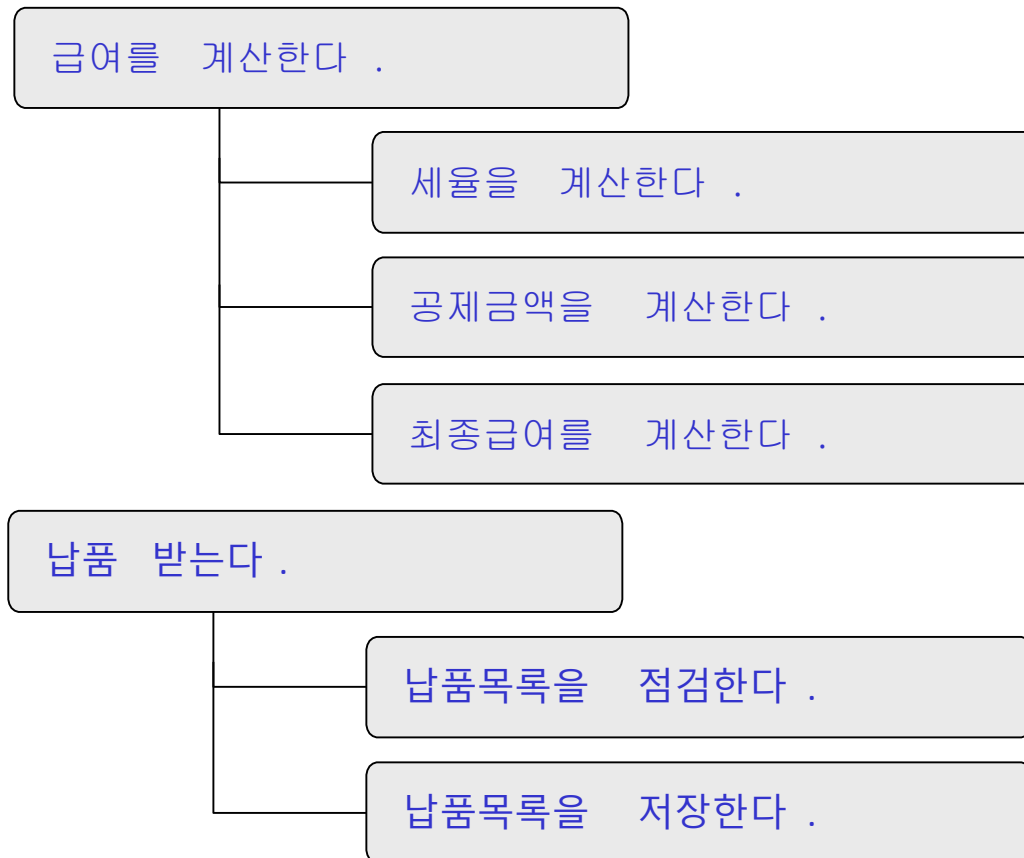
- 주문과 주문목록에 대한 부분



3. 단위 프로세스

정정된 업무기능분해도

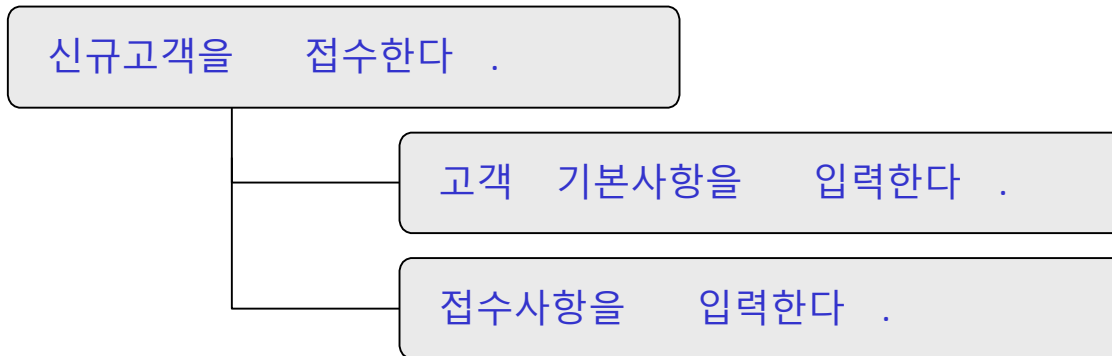
● ELEMENTAY PROCESS 식별



3. 단위 프로세스

정정된 업무기능분해도

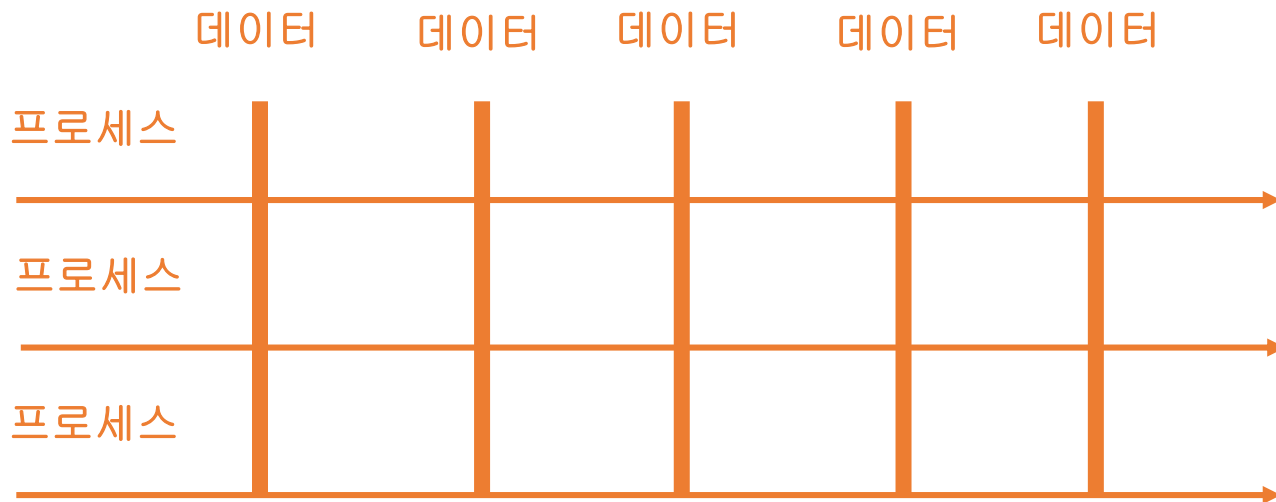
- ELEMENTAY PROCESS 식별



4. 상관모델링

상관모델링이란...

- 상관모델링이란 정보화시스템을 구축하기 위해 그 업무에 존재하는 무엇에 대해 무슨 일이 행해지고 있는지를 또는 무슨 일에 의해 무엇이 영향을 받는지를 분석하는 방법이다.



4. 상관모델링

CRUD 매트릭스

● CRUD MATRIX 상관모델링

엔티타입 ELEMENTARY PROCESS	고객	주문	주문목록	제품				
신규고객을 등록한다	C							
제품주문을 신청한다	R	C	C	R				
주문량을 변경한다		R	U					
주문을 취소한다		D	D					
제품을 등록한다				C				
고객정보를 조회한다.	R							

ELEMENT PROCESS가 엔티타입에 어떠한 일을 하는지 기술한다.

4. 상관모델링

CRUD 매트릭스

● CRUD 없는 엔티티타입

엔티티타입 ELEMENTARY PROCESS	고객	주문	사원	주문목록	제품			
신규고객을 등록한다	C							
제품주문을 신청한다	R	C		C				
주문량을 변경한다		R		U				
주문을 취소한다		D		D				
제품을 등록한다					C			
고객정보를 조회한다.	R							

엔티티타입에 발생하는
CRUD가 존재하지 않음

4. 상관모델링

CRUD 매트릭스

● CREATE 없는 엔티티타입

엔티티타입 ELEMENRRARY PROCESS	고객	주문	주문목록	제품					
신규고객을 등록한다	C								
제품주문을 신청한다	R	C	C	R					
주문량을 변경한다		R	U						
주문을 취소한다		D	D						
고객정보를 조회한다.	R								

엔티티타입에 READ는 존재하는데 데이터를 생성하는 ELEMENT PROCESS가 없다.

4. 상관모델링

CRUD 매트릭스

● READ 없는 엔티티타입

엔티티타입 ELEMENRRARY PROCESS	고객	주문	주문목록	제품				
신규고객을 등록한다	C							
제품주문을 신청한다	R	C	C	R				
주문량을 변경한다		R	U					
주문을 취소한다		D	D					
고객정보를 조회한다.	R							

주문목록은 생성하고 수정하고 삭제는 하지만 데이터가 읽혀지지 않는다.

4. 상관모델링

CRUD 매트릭스

표기가 안된 ELEMENTARY PROCESS

엔티티타입 \ ELEMENTARY PROCESS	고객	주문	주문목록	제품			
신규고객을 등록한다	C						
제품주문을 신청한다	R	C	C	R			
주문량을 변경한다		R	U				
주문을 취소한다		D	D				
제품을 등록한다.				C			
고객정보를 조회한다.	R						
납품업체정보를 등록한다.							
납품업체정보를 조회한다.							

ELEMENTARY PROCESS와 관련된 엔티티타입이 존재하지 않음

4. 상관모델링

CRUD 매트릭스

엔티티타입 중복생성

엔티티타입 ELEMENRARY PROCESS	고객	주문	주문목록	제품			
신규고객을 등록한다	C						
제품주문을 신청한다	R	C	C	C			
주문량을 변경한다		R	U				
주문을 취소한다		D	D				
제품을 등록한다.				C			
고객정보를 조회한다.	R						

제품 엔티티타입이 주문 신청할 때도 발생이 되고 제품등록할 때도 발생이 된다.

4. 상관모델링

CRUD 매트릭스

● 속성 CRUD MATRIX – 속성 대 ELEMENTARY PROCESS

엔티티타입 ELEMENTARY PROCESS	고 객								
	고객번호	고객주소	전화번호	팩스번호	등록일자	전자메일주소	자동이체은행번호	거래횟수	거래정지여부
신규고객등록하다	C	C	C	C	C	C	C		
주문신청하다	R	R	R	R		R	R	U	
고객정보를 변경한다	R	U	U	U	U	U	U		
거래를 중지한다.	R								U

『1과목』 Fundamental

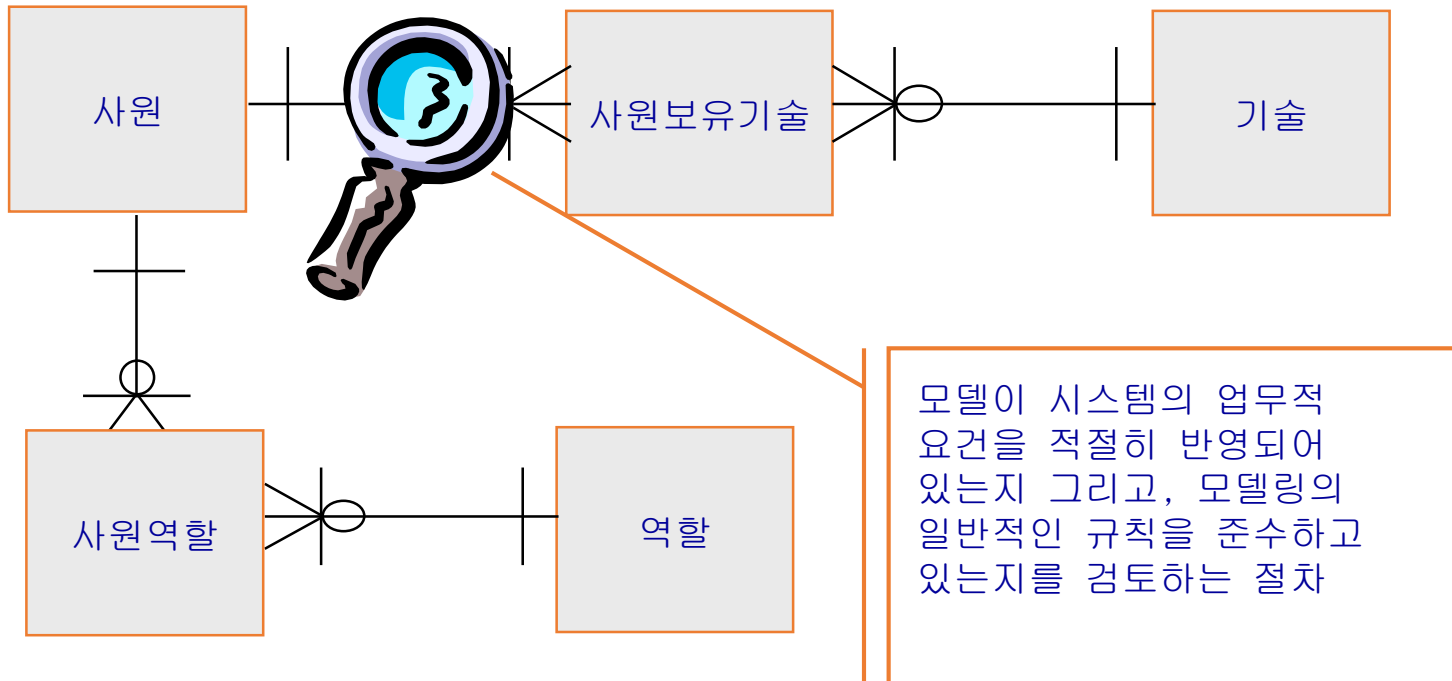
7교시 : 모델검토



1. 데이터모델 검토

1. 데이터모델 검토란?

- 모델이 시스템의 업무적 요건을 적절히 반영되어 있는지 그리고, 모델링의 일반적인 규칙을 준수하고 있는지를 검토하는 절차





1. 데이터모델 검토

모델검토수행자

1. 첫 번째는 모델링을 수행한 모델러가 모델검토 기준항목을 바탕으로 체크하는 방식
2. 두 번째는 시스템통합팀 이나 품질보증팀에서 하는 경우
3. 세 번째는 외부 감리인원을 초청하여 모델검토를 받는 경우.



2. 모델검토의 2가지 측면

업무적 측면의 검토

- 모델링이 시스템의 업무적 요건을 충분히 반영하고 있는지, 모델링의 관계가 해당 업무 프로세스에 잘 부합하는지를 검토한다. 주로 시스템 사용자와 모델러 간에 분석 단계 중 지속적인 검토가 이루어지며, 분석 단계 말에 프로젝트 내 품질보증팀 또는 시스템통합팀이나 제3의 독립된 조직의 주관 하에 검토를 실시하기도 한다. 업무적 요건 충족을 주로 검토하므로 요구사항 정의서, 업무기능 분해도, CRUD Matrix와 같은 분석 단계 산출물이 관련 자료로 활용된다.
- 검토 내역 사례)
 - 사용자 요구사항과 대비하여 미 도출된 엔티티타입 및 속성은 없는가?
 - 엔티티타입간의 관계는 업무절차와 잘 부합하는가?



2. 모델검토의 2가지 측면

모델규약 측면의 검토

- 데이터모델링이 갖추어야 할 모델링에 관한 일반적인 규약을 잘 준수하고 있는지를 검토한다. 분석 단계 말에 프로젝트 품질보증팀 또는 시스템통합팀이나 제3의 독립된 조직의 주관 하에 검토를 실시하며, 케이스툴(CASE TOOL)의 검증 기능이나 검토와 관련된 기타 툴이 활용되기도 한다.
- 검토 내역 사례)
 - 일반적인 검토 사례는 다음과 같은 부분이 있다.
 - 다른 타입/크기, CONSTRAINT를 가지는 중복된 속성은 존재하지 않는가?
 - 엔티티간의 관계가 M:N으로 정의된 것은 없는가?
 - 엔티티타입명, 속성명, 관계명에 대한 정의는 모델의 일반규약을 준수하고 있는가?



3. 데이터모델검토 4가지

데이터모델검토 4가지

1. 첫 번째는 엔티티타입 검토
2. 두 번째는 관계 검토
3. 세 번째는 속성 검토
4. 네 번째는 도메인/용어 검토



3. 데이터모델검토 4가지

1. 엔티티타입 검토

- 선정된 PRIMARY KEY가 업무적으로 발생하는 자료의 유일성을 보장하는가?
- 선정된 PRIMARY KEY는 효율적인 모습을 가지고 있는가?
- 자료의 발생유형이 유사한 엔티티는 통합되었는가?
- 독립된 엔티티타입이나 엔티티타입의 그룹은 존재하지 않는가?
- 병합 또는 분리되어야 할 엔티티타입은 존재하지 않는가?
- 추가적으로 도출되어야 하거나 불필요한 엔티티타입은 없는가?
- 엔티티타입이 주변 여러 엔티티타입의 공통 엔티티타입인 경우 자료원천이 어느 엔티티타입으로부터 인지 추적 가능한가?
- PRIMARY KEY의 순서는 시스템의 성능을 고려하여 적절한 순서로 정의되어 있는가?

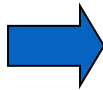
3. 데이터모델검토 4가지

1. 엔티티타입 검토

- 선정된 PRIMARY KEY가 업무적으로 발생하는 자료의 유일성을 보장하는가?
- 자료의 유일성을 보장할 수 없는 항목에 의한 Primary Key 선정
- 보험업무에서 초기 분석 당시 보험계약변경은 하루에 한 번 이상 발생하지 않는다고 예측했으나, 업무가 구체화되고 이전 데이터 분석이 이루어지면서 두 번 이상 발생하는 예가 발견되어 이력으로 관리

보험계약변경

계약번호
변경일자



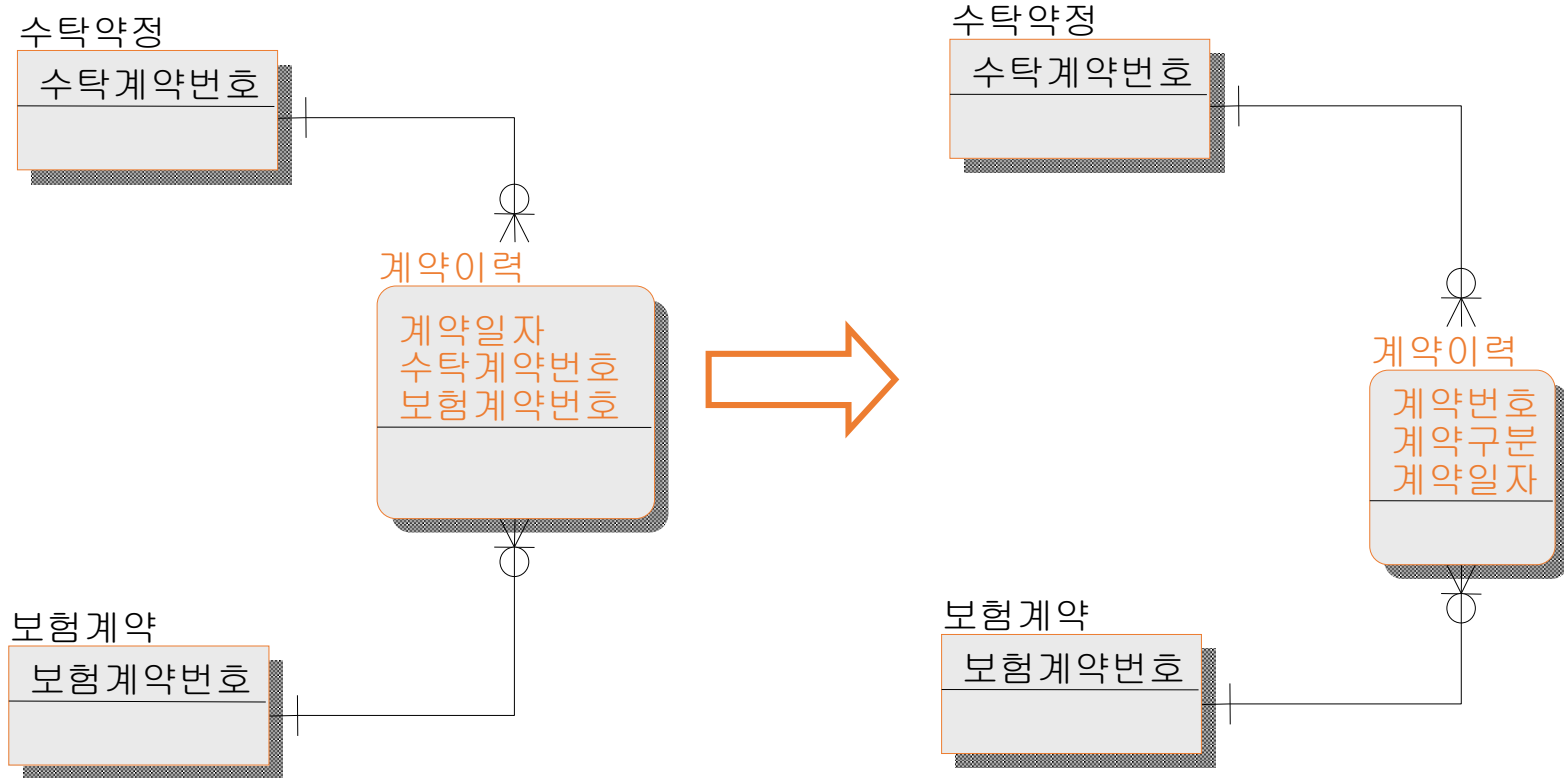
보험계약변경

계약번호
변경일자
변경순번

3. 데이터모델검토 4가지

1. 엔티타입 검토

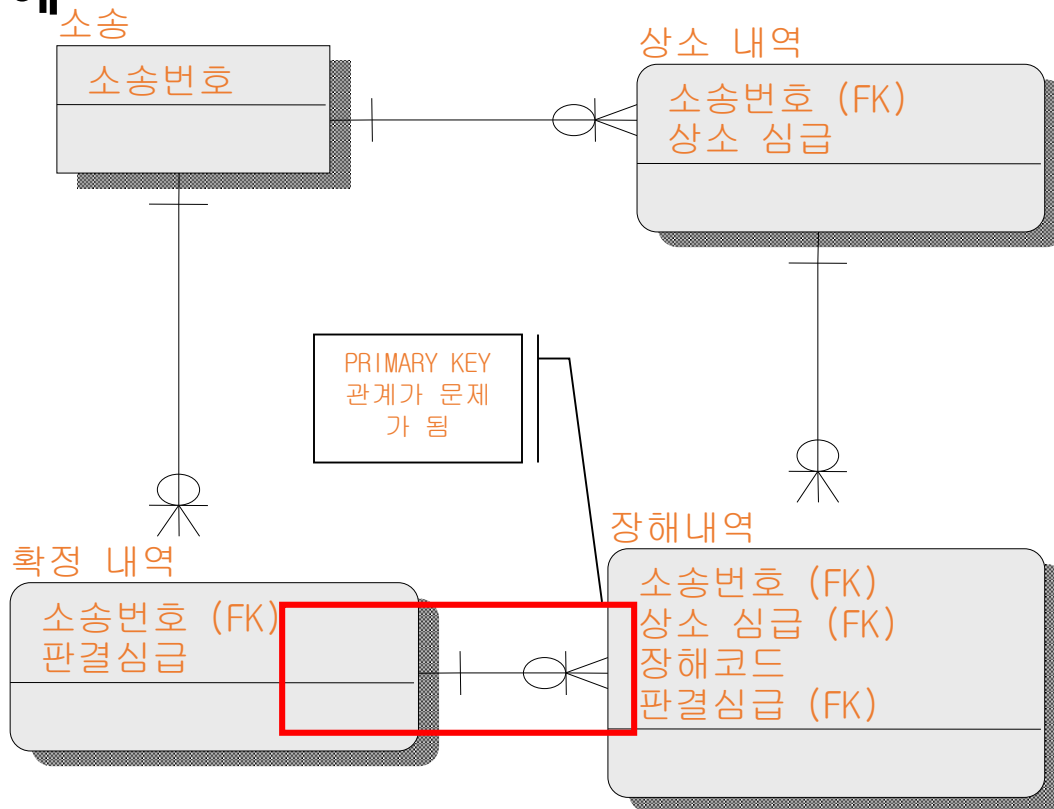
- 선정된 PRIMARY KEY가 업무적으로 발생하는 자료의 유일성을 보장하는가?
- 필요 이상의 항목에 의한 Primary Key 선정 – ARC관계 예



3. 데이터모델검토 4가지

1. 엔티타입 검토

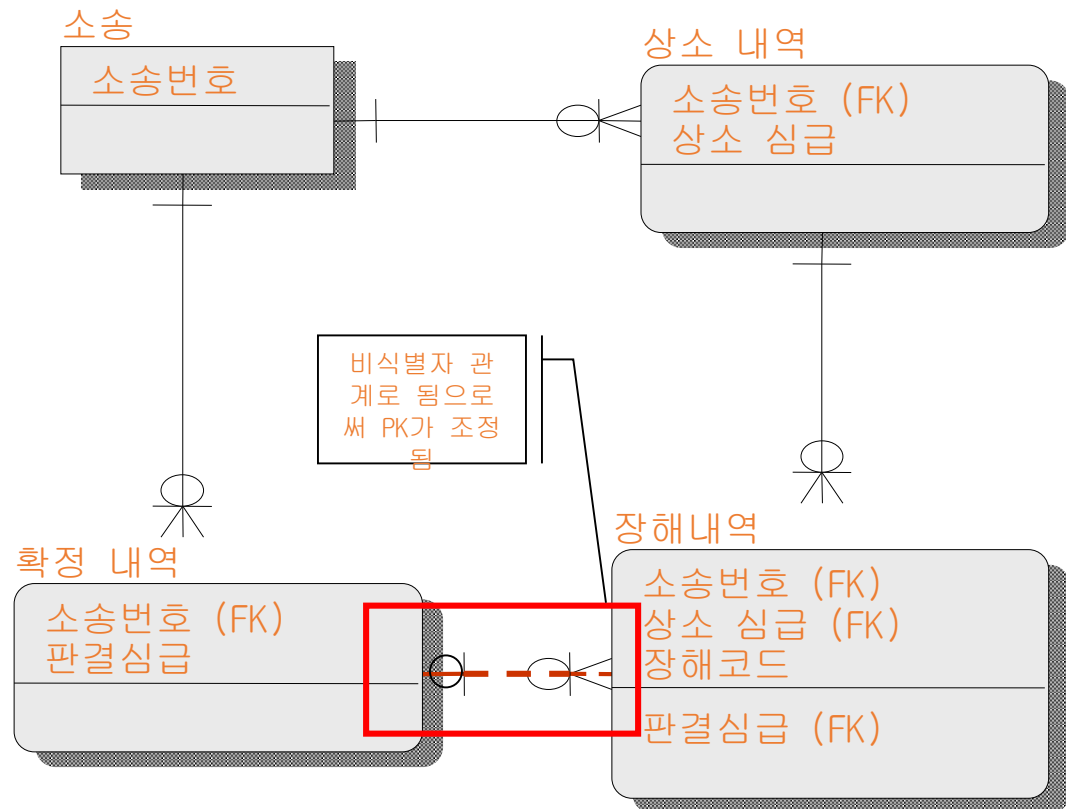
- 선정된 PRIMARY KEY가 업무적으로 발생하는 자료의 유일성을 보장하는가?
- 필요 이상의 항목에 의한 Primary Key 선정 – 완전종속과 부분종속의 문제 예



3. 데이터모델검토 4가지

1. 엔티타입 검토

- 선정된 PRIMARY KEY가 업무적으로 발생하는 자료의 유일성을 보장하는가?
- 필요 이상의 항목에 의한 Primary Key 선정 – 완전종속과 부분종속의 문제 해결 예



3. 데이터모델검토 4가지

1. 엔티티타입 검토

- 선정된 **PRIMARY KEY**는 효율적인 모습을 가지고 있는가?
- 선정된 속성은 해당 업무에 대해서 대표성 검증



사원

주민등록번호
(690815-XXXXXXX)

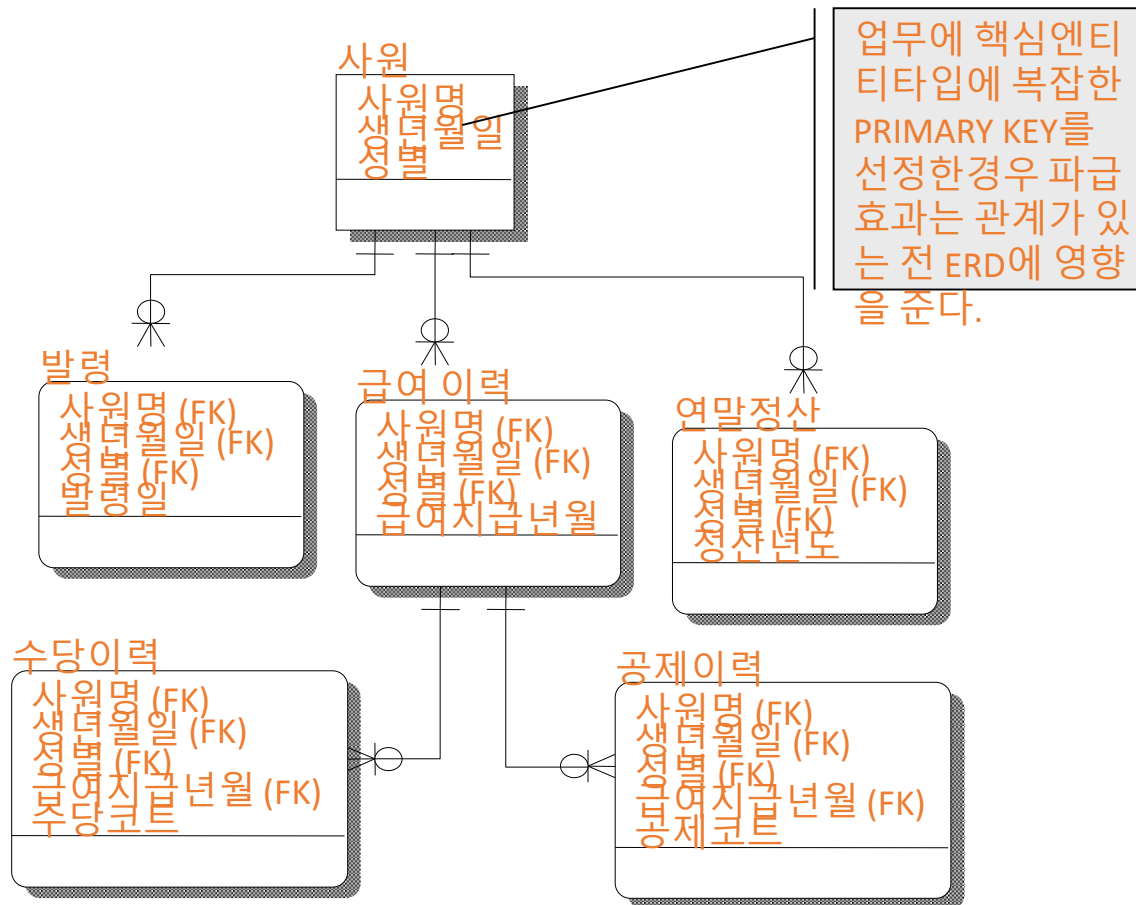
국민봉급자

사번
(45588)

3. 데이터모델검토 4가지

1. 엔티타입 검토

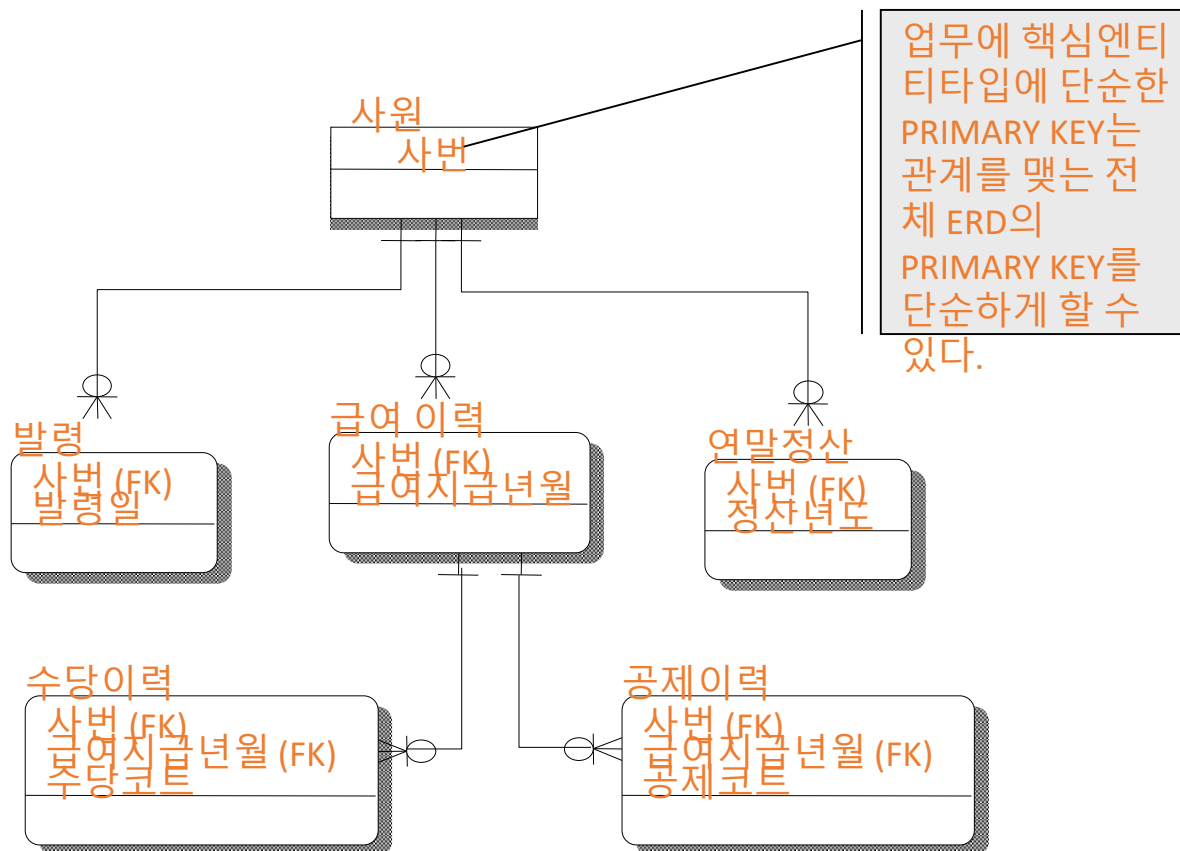
- 선정된 PRIMARY KEY는 효율적인 모습을 가지고 있는가?
- PRIMARY KEY가 복잡하게 선정된 경우



3. 데이터모델검토 4가지

1. 엔티티타입 검토

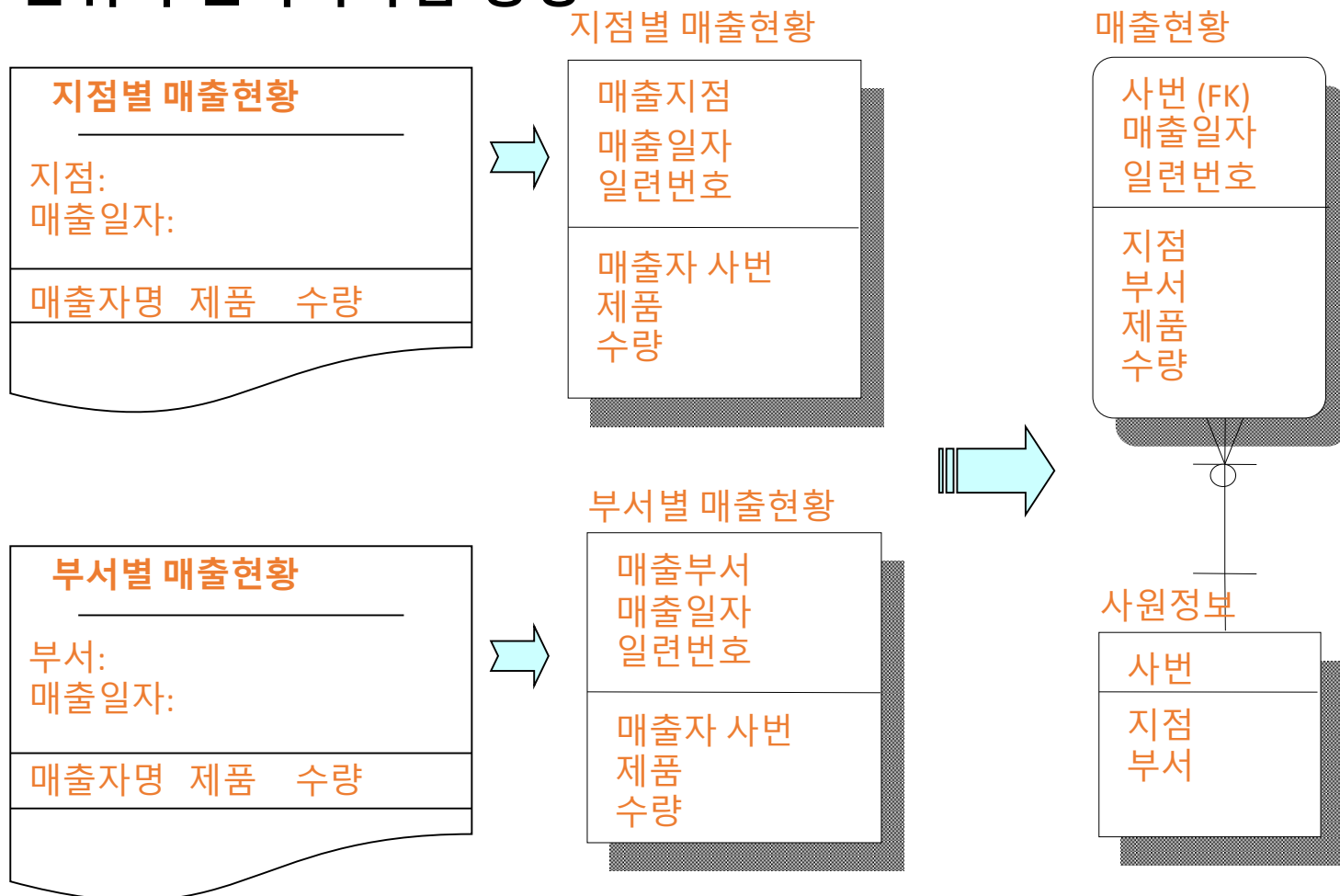
- 선정된 PRIMARY KEY는 효율적인 모습을 가지고 있는가?
- PRIMARY KEY가 복잡하게 선정된 경우 - 단순화



3. 데이터모델검토 4가지

1. 엔티티타입 검토

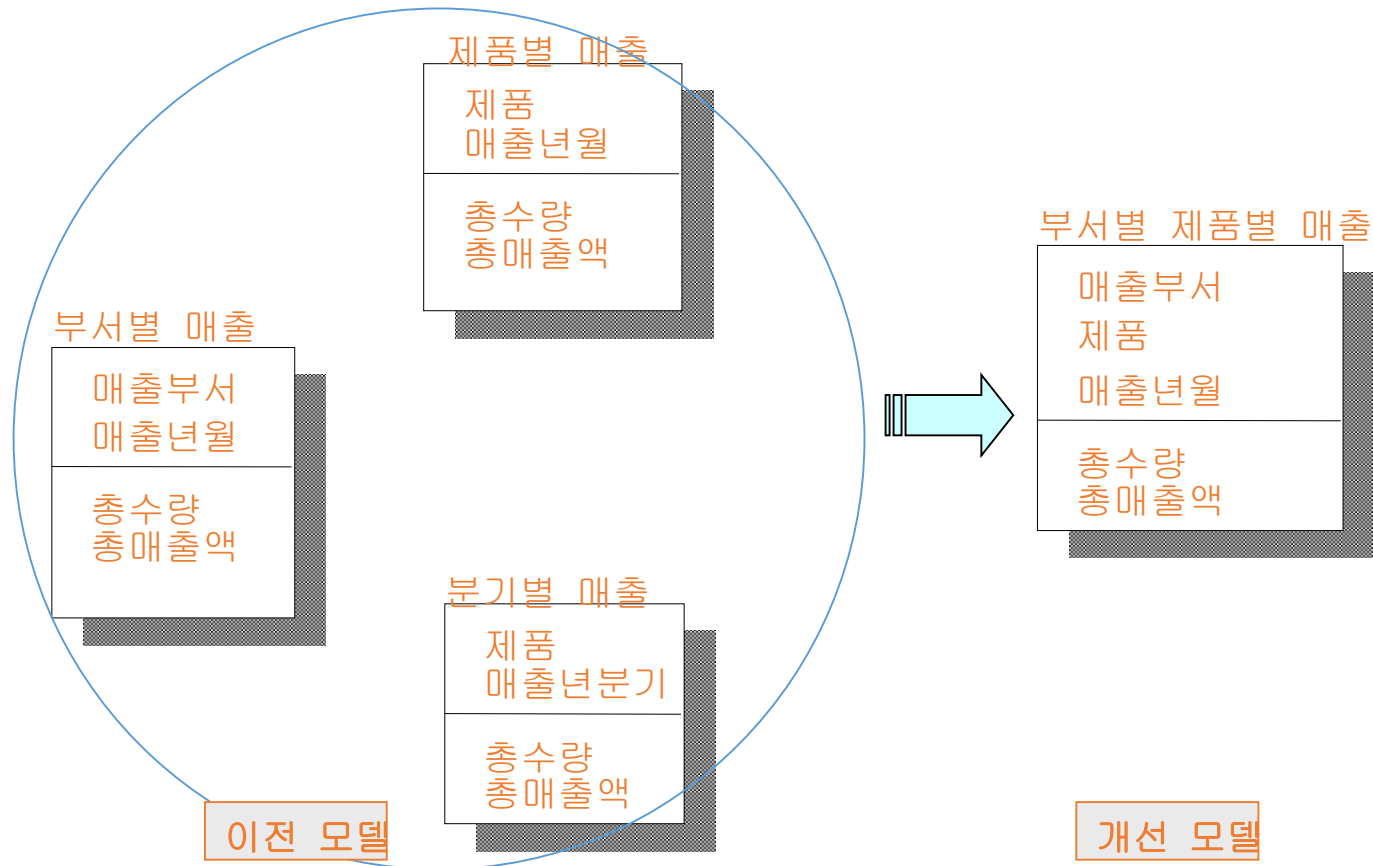
- 자료의 발생유형이 유사한 엔티티는 통합되었는가?
- 장표 단위의 엔티티타입 생성



3. 데이터모델검토 4가지

2. 엔티티타입

- 자료의 발생유형이 유사한 엔티티는 통합되었는가?
- 집약도가 큰 여러 개의 통계 엔티티타입



3. 데이터모델검토 4가지

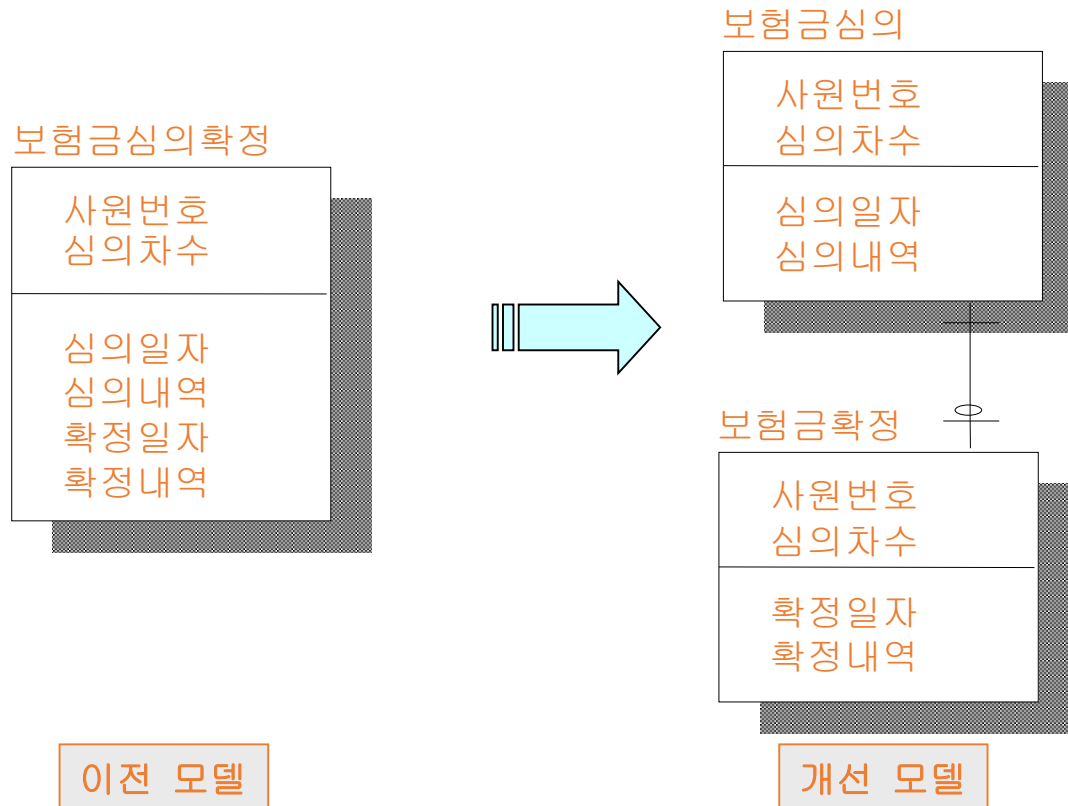
2. 엔티티타입

- 엔티티타입이나 엔티티타입의 그룹은 존재하지 않는가?
- 독립된 엔티티타입으로 존재하는 경우
 1. 첫째, 코드 엔티티는 관계를 가지나 코드 엔티티의 관계를 일일이 표현하면 ERD가 매우 복잡해지므로 일반적으로 표현하지 않는다.
 2. 둘째, 통계 엔티티타입은 검색 편의를 위한 단순한 조회용 엔티티로서 업무적 흐름을 가지지 않는다.
 3. 셋째, 시스템 관리 엔티티타입은 업무와 개별적으로 시스템을 통제하기 위한 엔티티타입이므로 경우에 따라서는 업무적 연계를 가지지 않을 수도 있다.

3. 데이터모델검토 4가지

2. 엔티티타입

- 병합 또는 분리되어야 할 엔티티타입은 존재하지 않는가?
- 연속적인 업무절차는 가지는 엔티티타입들의 병합으로 인해 발생 되는 문제¹

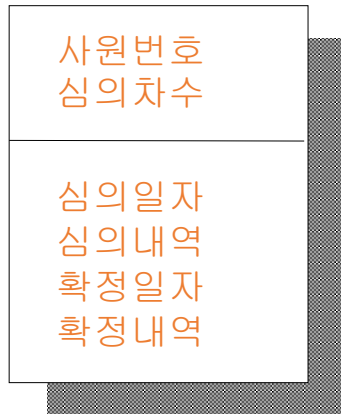


3. 데이터모델검토 4가지

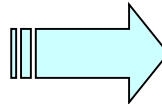
2. 엔티티타입

- 병합 또는 분리되어야 할 엔티티타입은 존재하지 않는가?
- 연속적인 업무절차는 가지는 엔티티타입들의 병합으로 인해 발생 되는 문제2

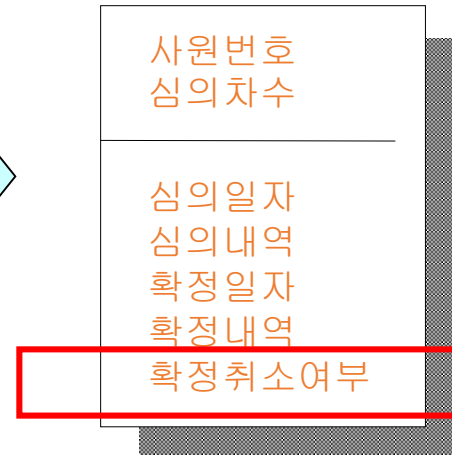
보험금심의확정



이전 모델



보험금심의확정



개선 모델

3. 데이터모델검토 4가지

2. 엔티티타입

- 병합 또는 분리되어야 할 엔티티타입은 존재하지 않는가?
- 대칭적인 업무를 가지는 엔티티타입의 분리

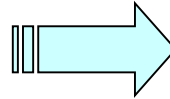
매출전표

전표번호
금액 발행인 발행일자

매입전표

전표번호
금액 발행인 발행일자

이전 모델



전표

전표번호
전표구분 금액 발행인 발행일자

개선 모델

3. 데이터모델검토 4가지

2. 엔티티타입

- 추가적으로 도출되어야 하거나 불필요한 엔티티타입은 없는가? 엔티티타입이 추가적으로 도출되는 경우
 1. 첫째, 도출된 엔티티타입 간에 관계를 분석하는 과정에서 M:N의 관계가 발생하여 이를 분해하면서 엔티티타입이 도출
 2. 둘째, 엔티티타입을 1차, 2차, 3차 정규화를 실시하는 과정
 3. 셋째, 단위 시스템간의 모델링을 통합하는 과정에서 시스템 간 연계 엔티티타입이 생성되지 않은 것이 발견되는 경우
 4. 넷째, 단위 시스템간의 모델링을 통합하는 과정에서 중복되거나 유사한 엔티티타입이 발견되어 제거 또는 통합하는 경우
 5. 다섯째, 통계 및 시스템 관리를 목적으로 엔티티타입을 추가 도출
 6. 여섯째, 초기에 도출된 엔티티가 추 후 분석과정에서 해당 업무영역에서 제외됨

3. 데이터모델검토 4가지

2. 엔티티타입

- 추가적으로 도출되어야 하거나 불필요한 엔티티타입은 없는가?
- 단위 시스템간에 중복된 엔티티타입 발생

시스템	엔티티 타입명	엔티티 타입명	엔티티 타입명	엔티티 타입명	엔티티 타입명	엔티티 타입명	엔티티 타입명
영업시스템	제품	거래처	전표	송장	의뢰	발주	지점
인사시스템	사원	부서	호봉	근태	급여	고과	전표
회계시스템	청구	거래처	계좌	물품	입금	통화	부서

 → 제품, 물품은 동일한 성격의 엔티티타입

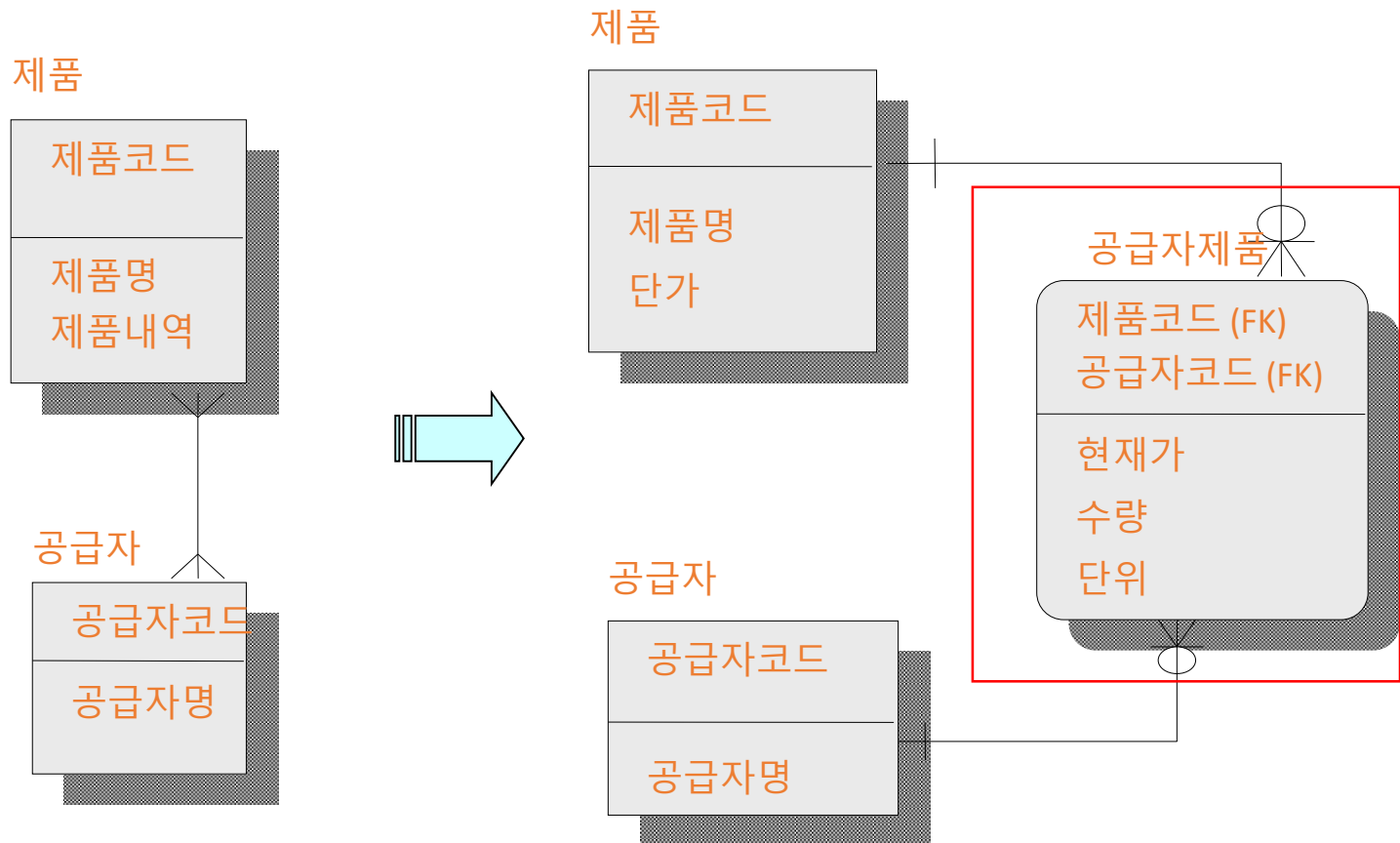
 → 거래처, 동일한 성격의 엔티티타입

 → 지점, 부서 동일한 성격의 엔티티타입

3. 데이터모델검토 4가지

2. 엔티타입

- 추가적으로 도출되어야 하거나 불필요한 엔티타입은 없는가?
- M:N 관계에 의한 엔티타입 생성



3. 데이터모델검토 4가지

2. 엔티티타입

- 추가적으로 도출되어야 하거나 불필요한 엔티티타입은 없는가?

1차 정규화 응용

일재고

물류센터코드
재고일자
월초재고수량
장기재고1개월수량
장기재고2개월수량
장기재고3개월수량
장기재고1개월주문수량
장기재고2개월주문수량
장기재고3개월주문수량
장기재고1개월금액
장기재고2개월금액
장기재고3개월금액
장기재고1개월주문금액
장기재고2개월주문금액
장기재고3개월주문금액

일재고

물류센터코드
재고일자
월초재고수량

일재고상세

물류센터코드 (FK)
재고일자 (FK)
재고기간
장기재고수량
장기재고주문수량
장기재고금액
장기재고주문금액

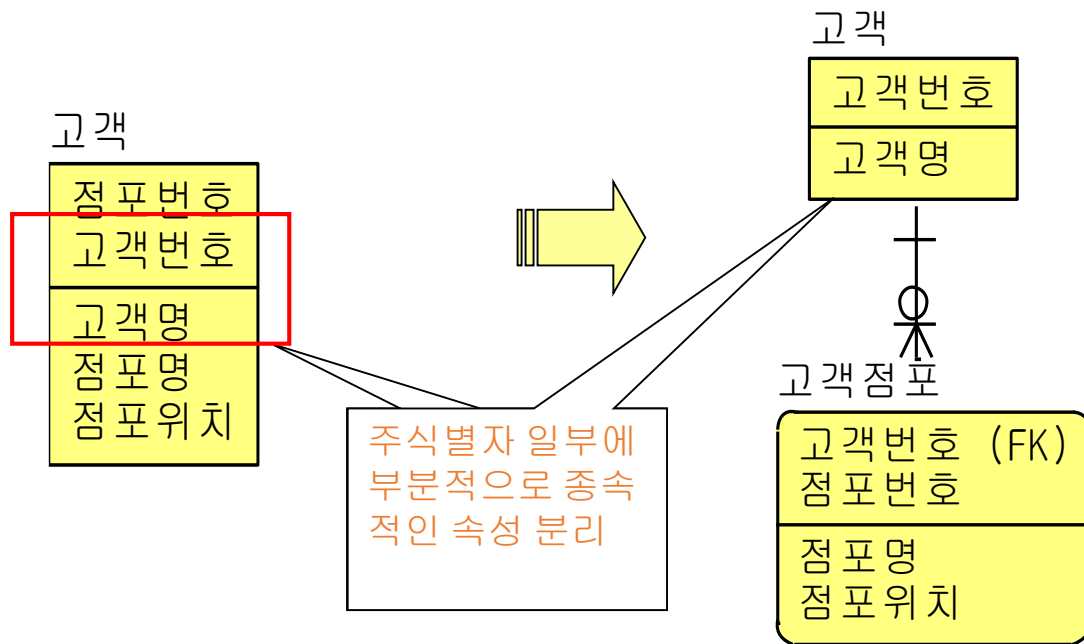
1차정규화가 안
되어 컬럼단위
로 중복된 모델
을 1차 정규화를
통해 분리함

3. 데이터모델검토 4가지

2. 엔티티타입

- 추가적으로 도출되어야 하거나 불필요한 엔티티타입은 없는가?

2차 정규화 응용

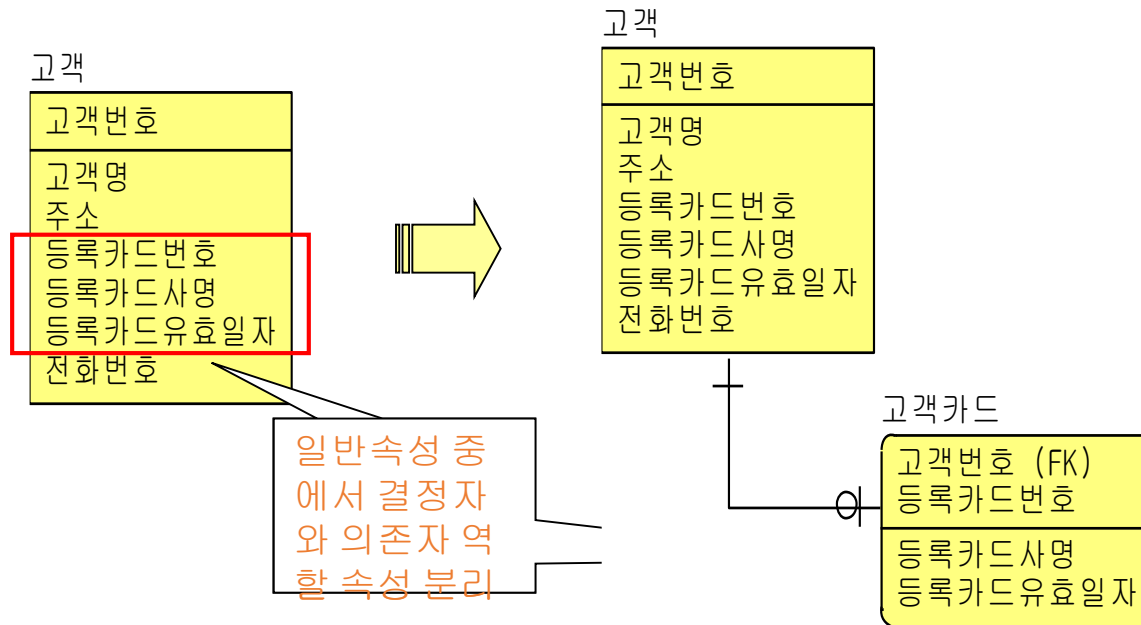


3. 데이터모델검토 4가지

2. 엔티티타입

- 추가적으로 도출되어야 하거나 불필요한 엔티티타입은 없는가?

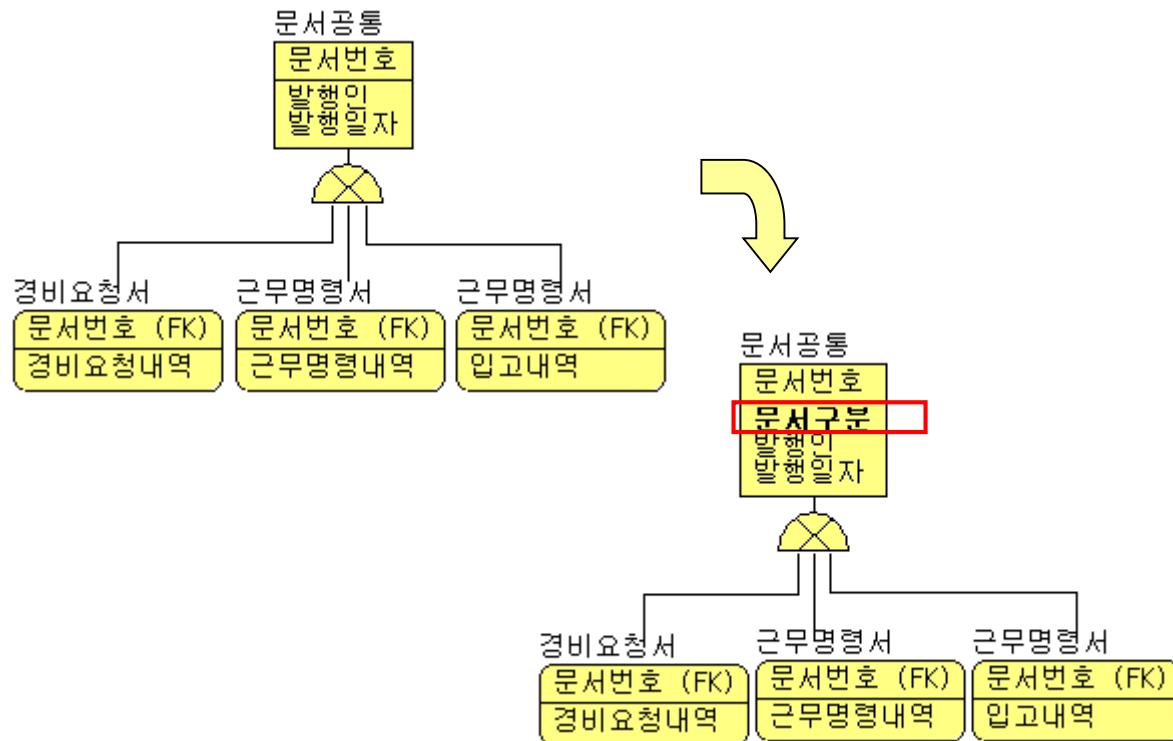
3차 정규화 응용



3. 데이터모델검토 4가지

2. 엔티티타입

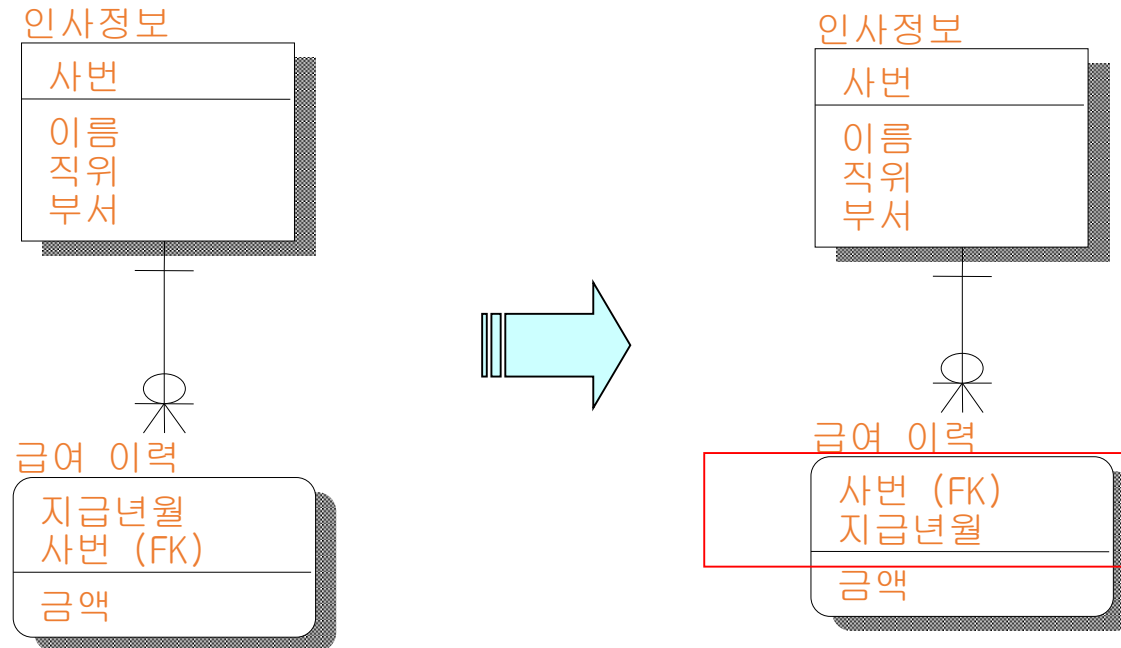
- 엔티티타입이 주변 여러 엔티티타입의 공통 엔티티타입인 경우 자료원천이 어느 엔티티타입으로 부터 인지 추적 가능한가?
- 슈퍼타입/서브타입 모델에서 자료 원천 구분 FLAG가 없는 문제 해결



3. 데이터모델검토 4가지

2. 엔티티타입

- PRIMARY KEY의 순서는 시스템의 성능을 고려하여 적절한 순서로 정의되어 있는가?
- 잘 사용되지 않는 속성이 PRIMARY KEY의 첫 번째 항목으로 선정되는 경우



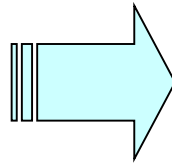
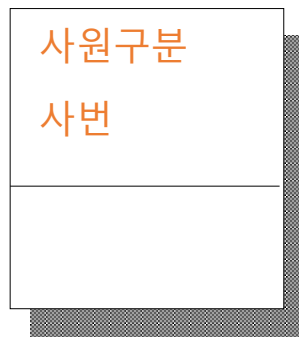
검색유형에 따라 PRIMARY KEY순서를 조정 하므로써
실제 SQL 실행시 성능저하를 예방할 수 있다.

3. 데이터모델검토 4가지

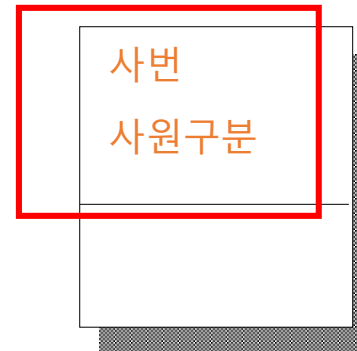
2. 엔티티타입

- PRIMARY KEY의 순서는 시스템의 성능을 고려하여 적절한 순서로 정의되어 있는가?
- 구분 FLAG와 같은 속성이 PRIMARY KEY의 첫 번째 항목으로 선정

사원정보



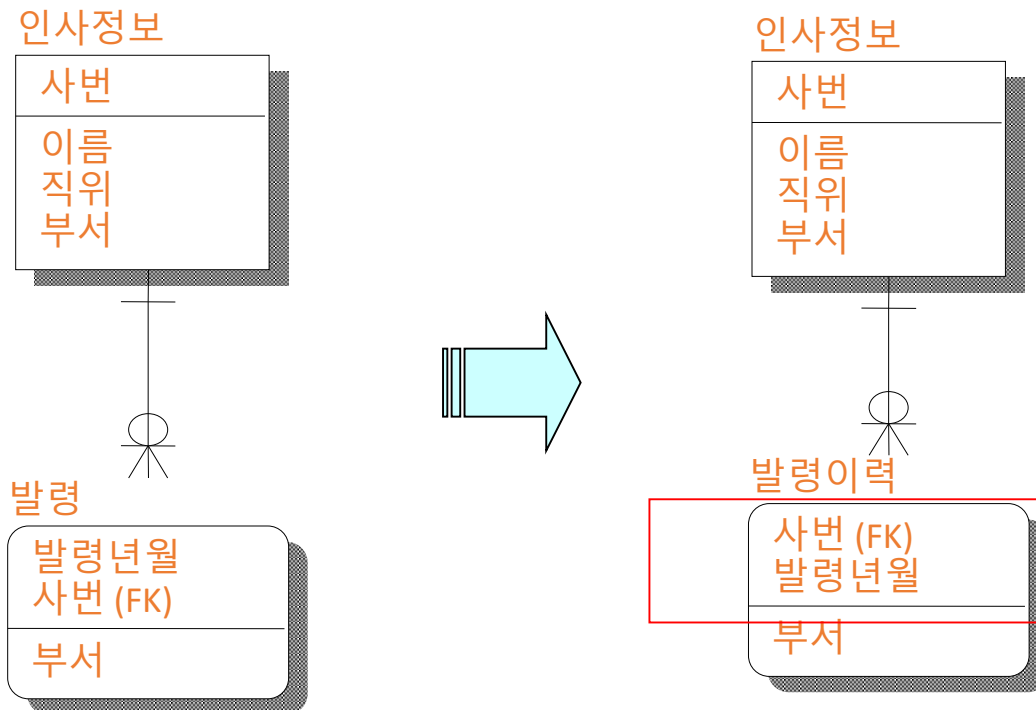
사원정보



3. 데이터모델검토 4가지

2. 엔티타입

- PRIMARY KEY의 순서는 시스템의 성능을 고려하여 적절한 순서로 정의되어 있는가?
- 날짜와 같이 주로 범위조회를 하는 속성이 PRIMARY KEY의 첫 번째 항목으로 선정되는 경우



날짜형식의 속성은 범위 조회하는 경우가 많으므로
PRIMARY KEY의 맨 첫번째에 있으면 인덱스를 이용하지 않을 수 있다.

3. 데이터모델검토 4가지

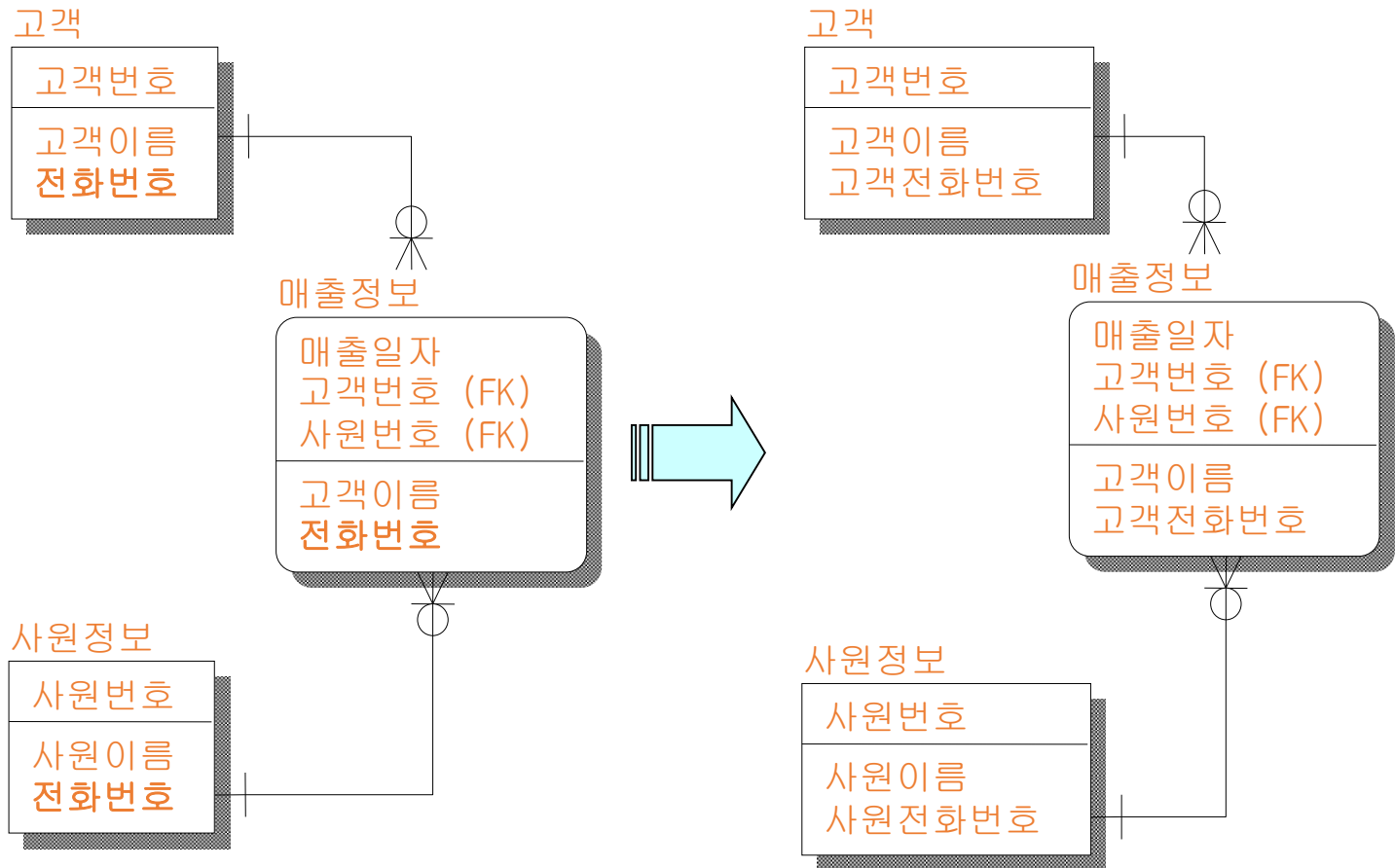
3. 속성 검토

- 반 정규화된 속성은 식별되는가?
- 반 정규화는 시스템 복잡도와 성능을 고려하여 적절하게 이루어졌는가?
- 동일명칭을 가지는 속성의 타입과 크기는 동일한가?
- 내부적인 속성을 가지고 있는 속성은 존재하지 않는가?
- 병합되어야 할 속성은 존재하지 않는가?
- 전후 레코드간 영향을 미칠 수 있는 속성은 없는가?
- 감사, 통계 등을 고려하여 속성이 정의되었는가?

3. 데이터모델검토 4가지

3. 속성 검토

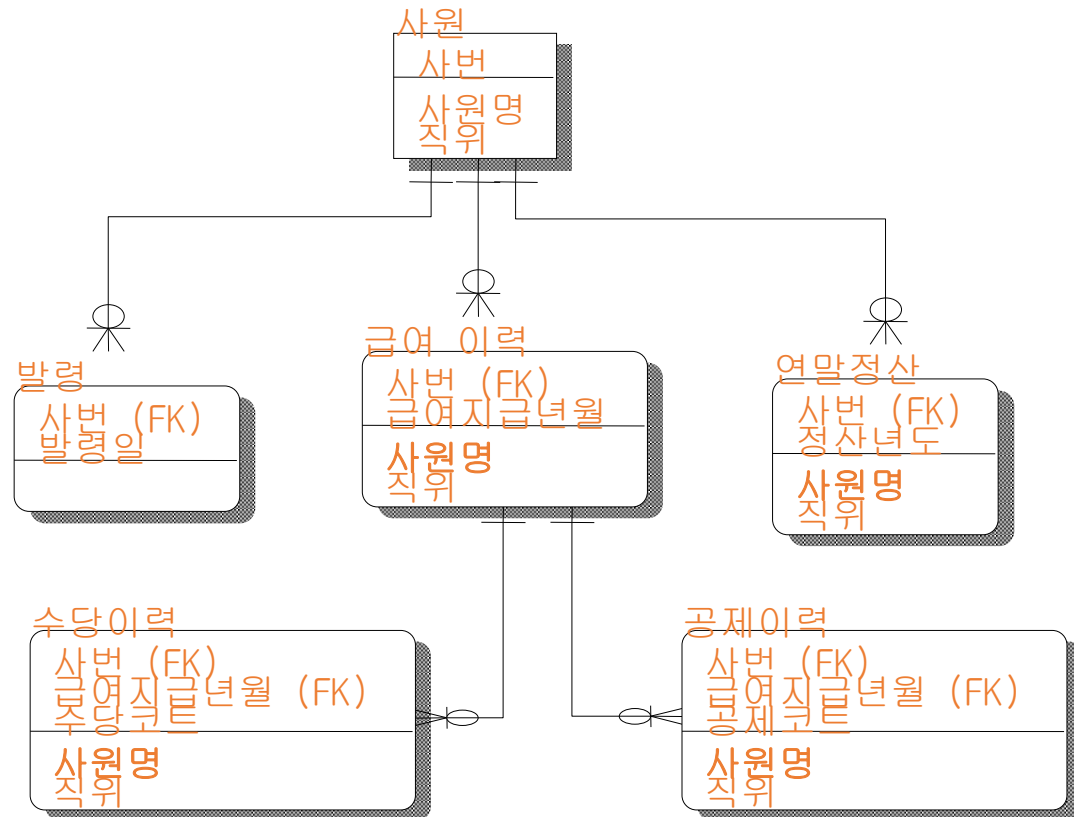
- 반 정규화된 속성은 식별되는가?
- 반 정규화된 속성이 실제로는 다른 속성이나 이름만 같은 속성이 공존함.



3. 데이터모델검토 4가지

3. 속성 검토

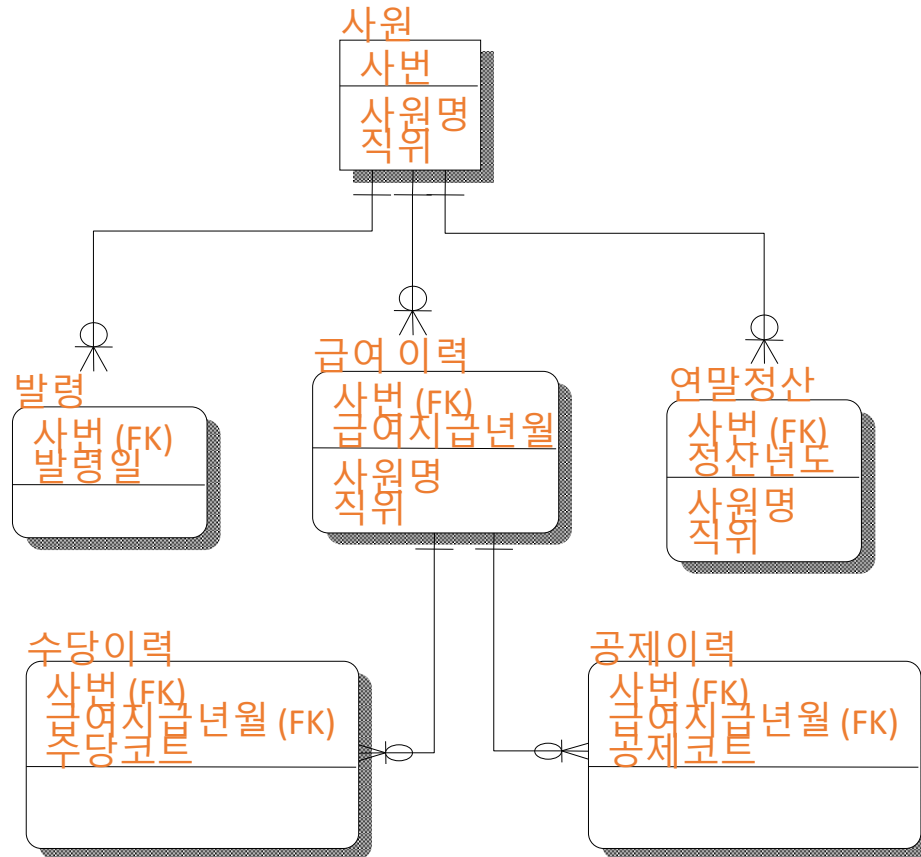
- 반 정규화는 시스템 복잡도와 성능을 고려하여 적절하게 이루어졌는가?
- 시스템 특성에 따르지 않은 과도한 반 정규화 - 개선이전



3. 데이터모델검토 4가지

3. 속성 검토

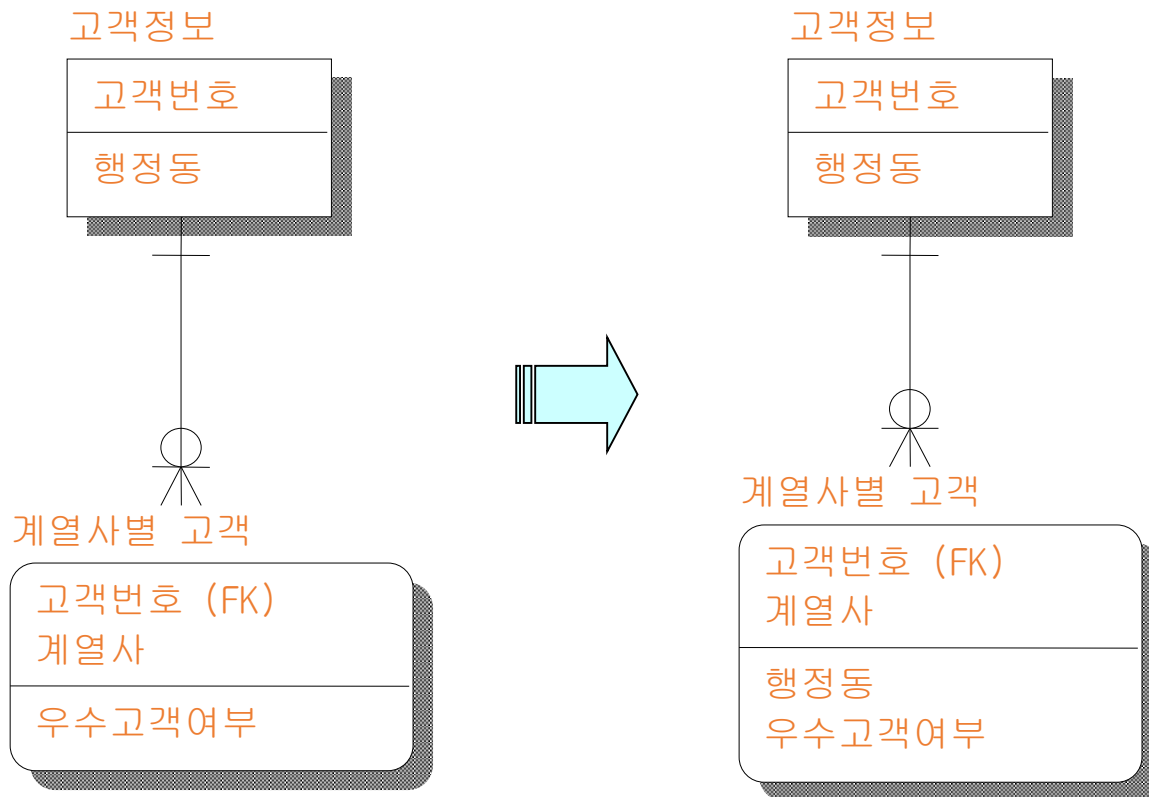
- 반 정규화는 시스템 복잡도와 성능을 고려하여 적절하게 이루어졌는가?
- 시스템 특성에 따르지 않은 과도한 반 정규화 - 개선이후



3. 데이터모델검토 4가지

3. 속성 검토

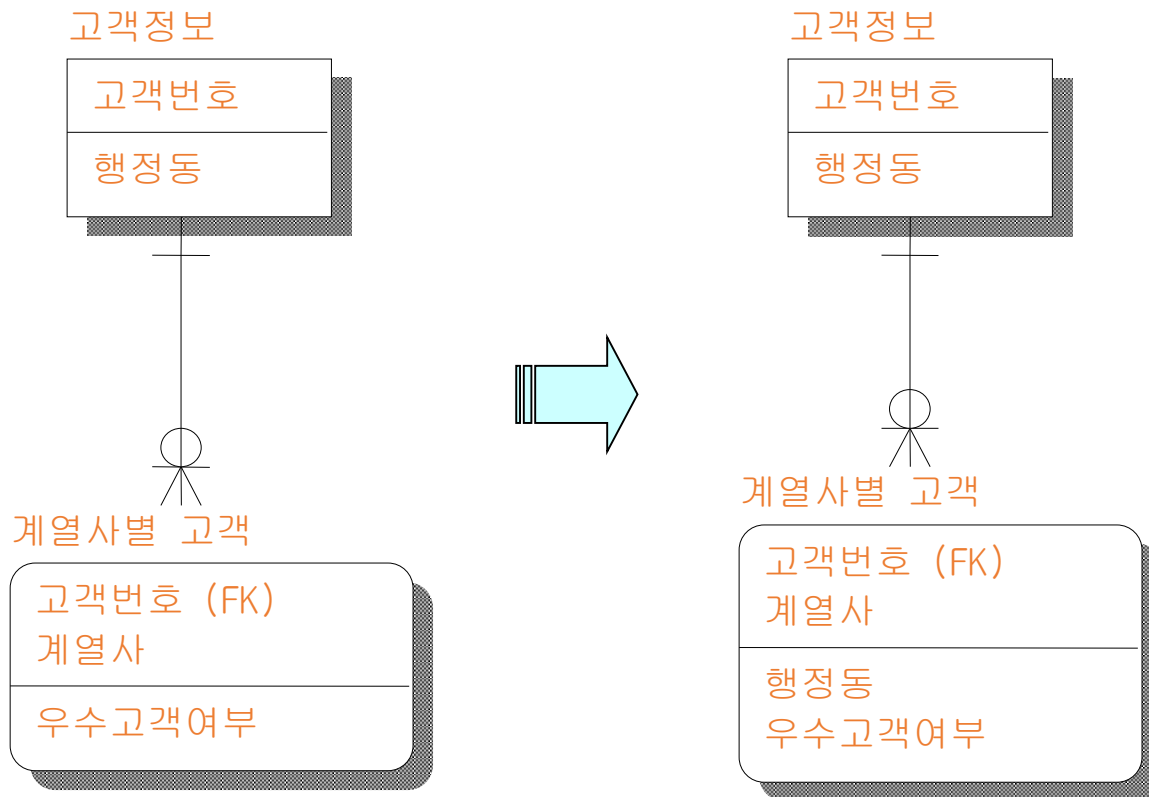
- 반 정규화는 시스템 복잡도와 성능을 고려하여 적절하게 이루어졌는가?
- 반 정규화를 하지 않으므로 인한 시스템 성능저하



3. 데이터모델검토 4가지

3. 속성 검토

- 반 정규화는 시스템 복잡도와 성능을 고려하여 적절하게 이루어졌는가?
- 반 정규화를 하지 않으므로 인한 시스템 성능저하



3. 데이터모델검토 4가지

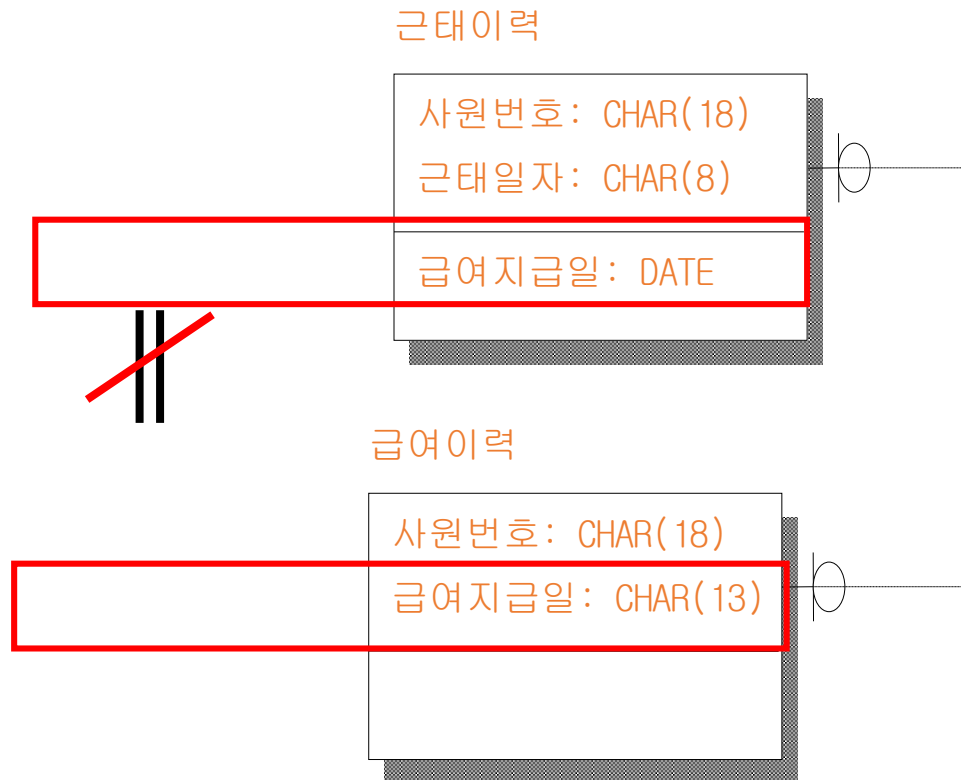
3. 속성 검토

- 동일명칭을 가지는 속성의 타입과 크기는 동일한가?
- TYPE 불일치
- 인사 시스템에서 사원정보의 사원명은 10 BYTE 이고 급여 이력의 사원명은 8 BYTE 이라고 하자. 이 때 김 미리네'라는 사원은 9 BYTE를 가진다. 따라서 사원정보에서 사원명을 읽어 급여이력이 복사하는 SQL(INSERT 또는 UPDATE)을 처리할 때 정보가 손실되게 되는 것이다.

3. 데이터모델검토 4가지

3. 속성 검토

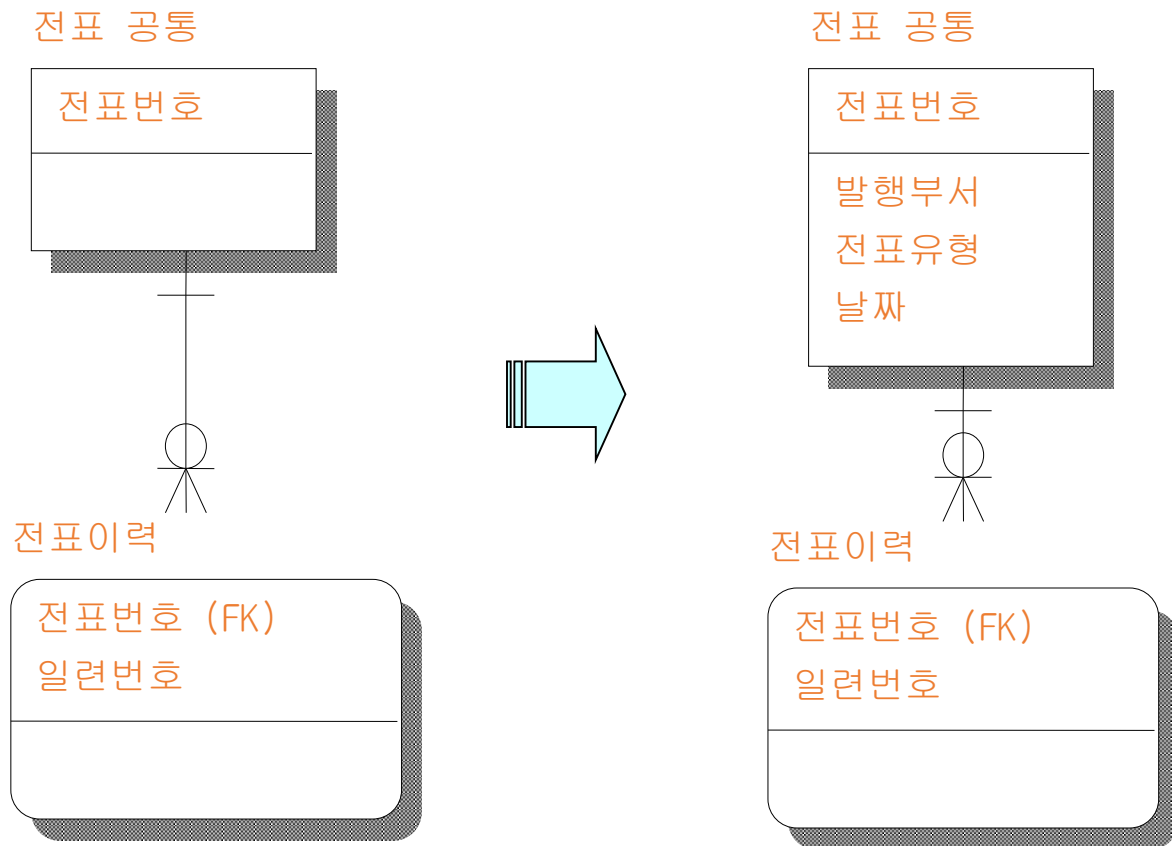
- 동일명칭을 가지는 속성의 타입과 크기는 동일한가?
- 컬럼 길이의 불일치



3. 데이터모델검토 4가지

3. 속성 검토

- 내부적인 속성을 가지고 있는 속성은 존재하지 않는가?
- 병합된 속성만 관리

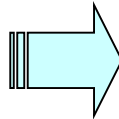
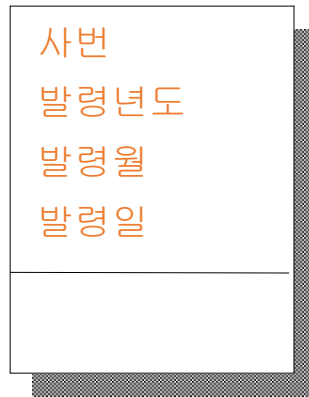


3. 데이터모델검토 4가지

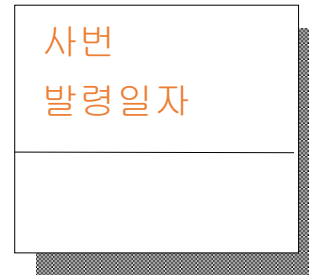
3. 속성 검토

- 병합되어야 할 속성은 존재하지 않는가?
- 분리된 속성관리

발령



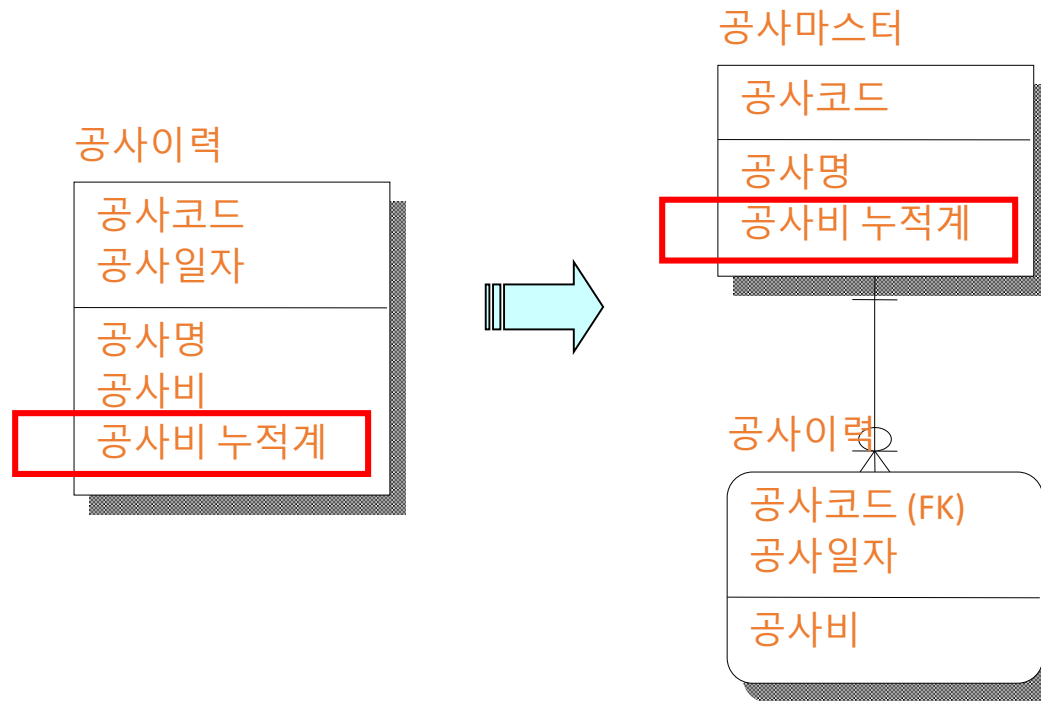
발령



3. 데이터모델검토 4가지

3. 속성 검토

- 전후 레코드간 영향을 미칠 수 있는 속성은 없는가?
- 전후 레코드간 영향을 주는 속성



3. 데이터모델검토 4가지

3. 속성 검토

- 감사, 통계 등을 고려하여 속성이 정의되었는가?
- 코드 화 할 수 있으나 텍스트로 정의된 속성
- 'ORACLE ADMIN'인 사원을 집계한다면 사번이 '11111'인 사원만 집계되어 그 값은 '1'로 나올 것이다.

사번	자격증취득일자	자격증명	자격증 번호
11111	1999.05.01	ORACLE ADMIN	A1220
11112	1999.10.10	오라클 Admin(ORACLE)	A2221
11113	1999.11.11	oracle admin	A1319
11114	2000.01.01	Oracle Administrator	A2142

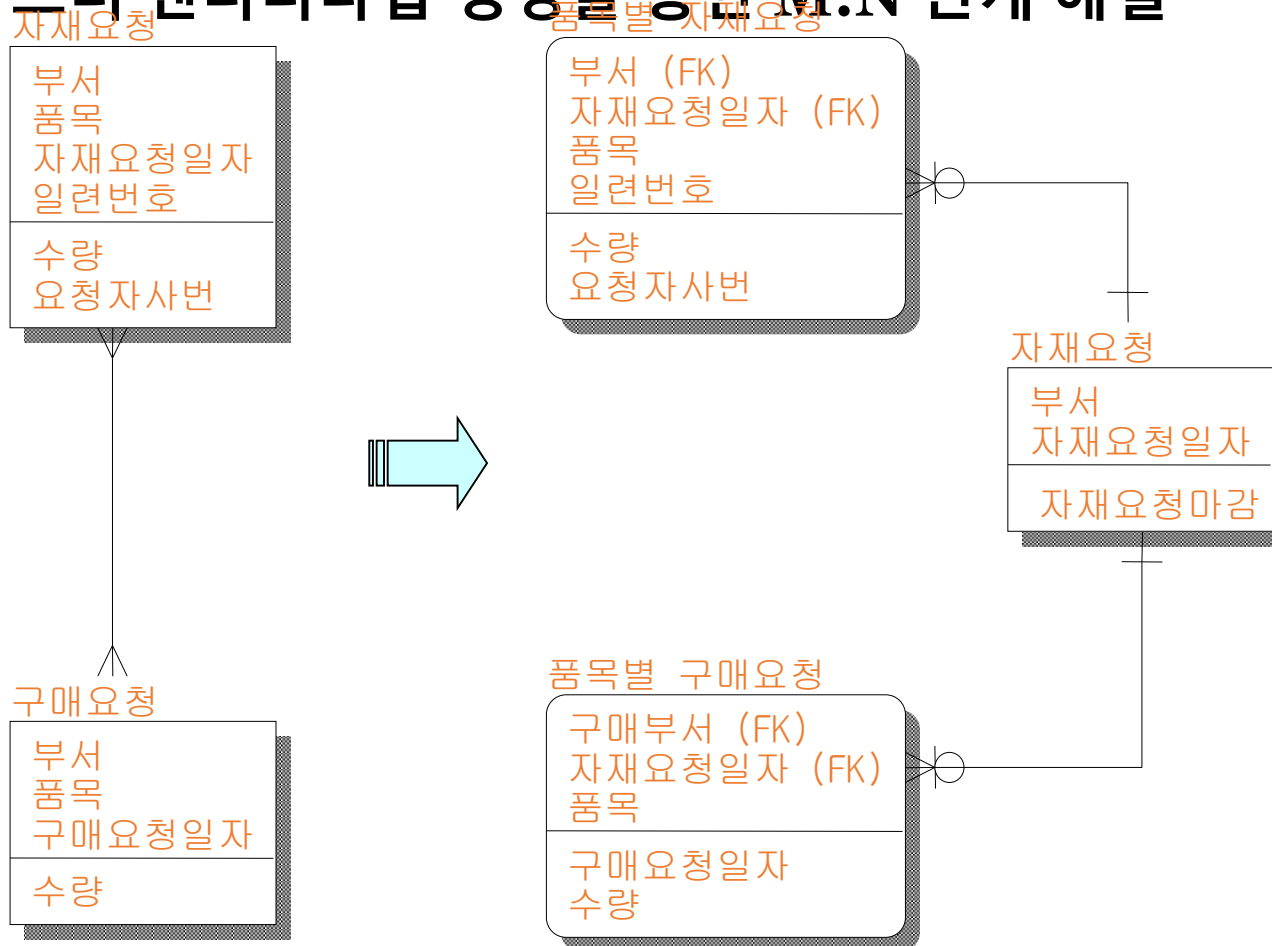
관계검토

- 엔티타입간의 관계가 M:N이 속성은 없는가?
- 엔티타입간의 관계는 업무적 흐름과 규약이 일치하는가?
- 업무적 흐름에 비추어 미 도출된 관계는 없는가?
- 관계에 대한 표현은 적절한 수준에서 이루어졌는가?

4. 관계검토

관계검토

- 엔티타입간의 관계가 M:N이 속성은 없는가?
- 상위 마스터 엔티타입 생성을 통한 M:N 관계 해결



4. 관계검토

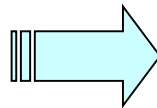
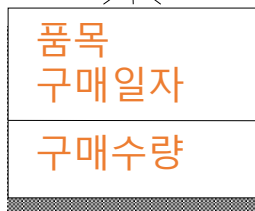
관계검토

- 엔티타입간의 관계가 M:N이 속성은 없는가?
- 두 엔티티 중 하나의 관계를 ALL OR NOTHING으로 하여 1:N의 관계를 정의하는 경우

구매요청



구매



구매요청



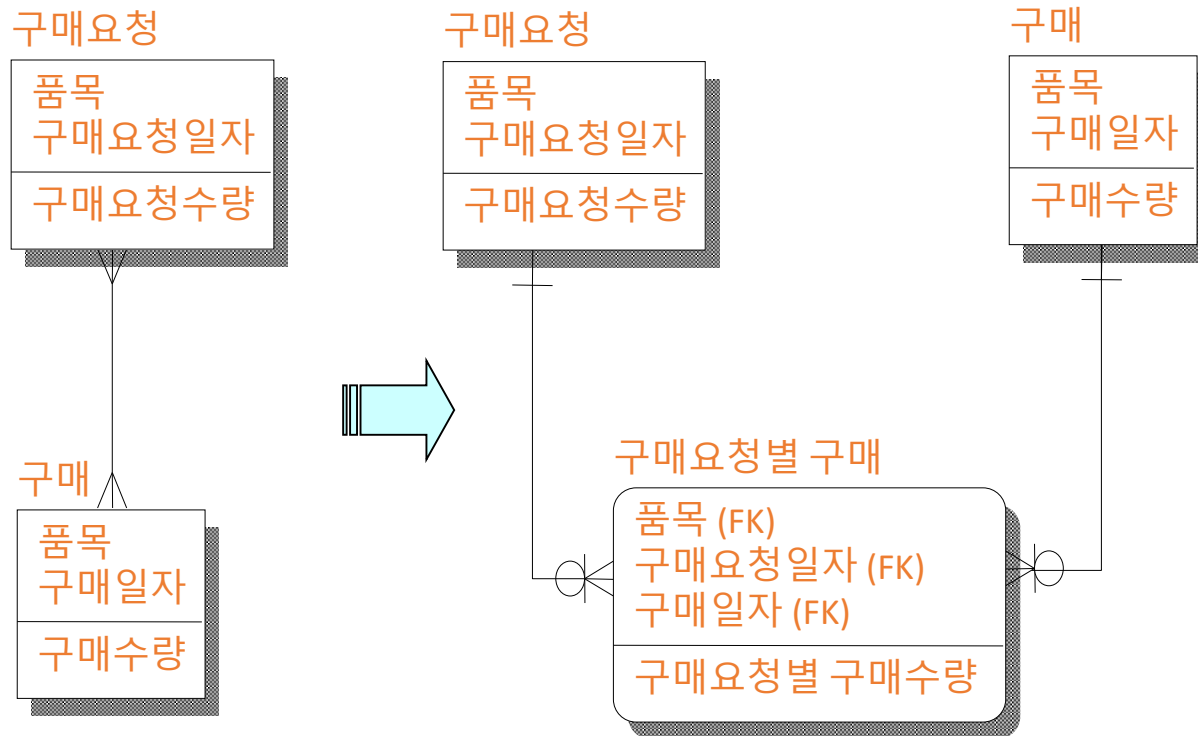
구매



4. 관계검토

관계검토

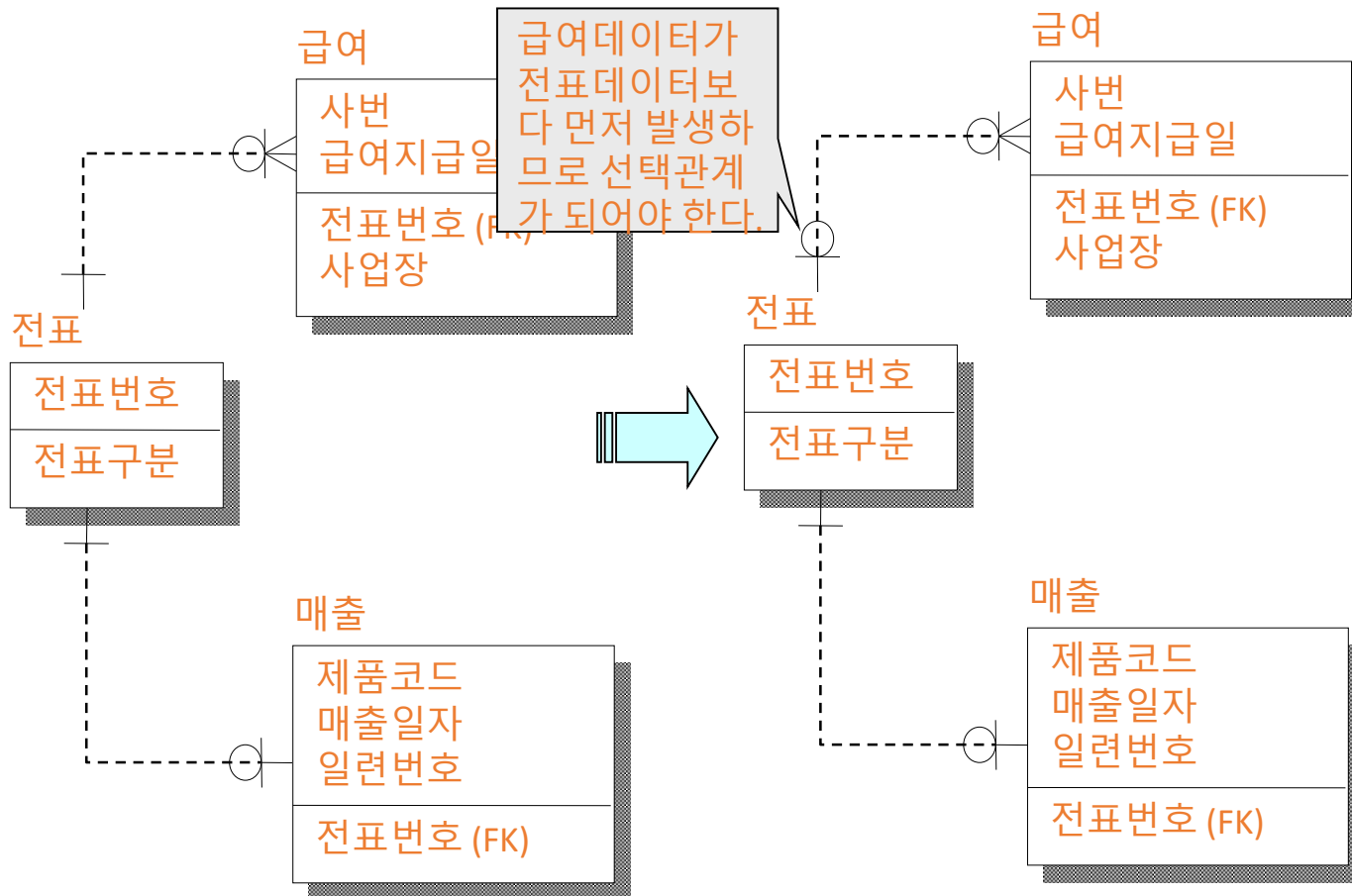
- 엔티타입간의 관계가 M:N이 속성은 없는가?
- 하위 엔티타입도출(관계엔티타입) 도출을 통한 해결



4. 관계검토

관계검토

- 엔티타입간의 관계는 업무적 흐름과 규약이 일치하는가?
- FK가 속성 생성시점이 자신의 레코드 생성시점과 다른 경우



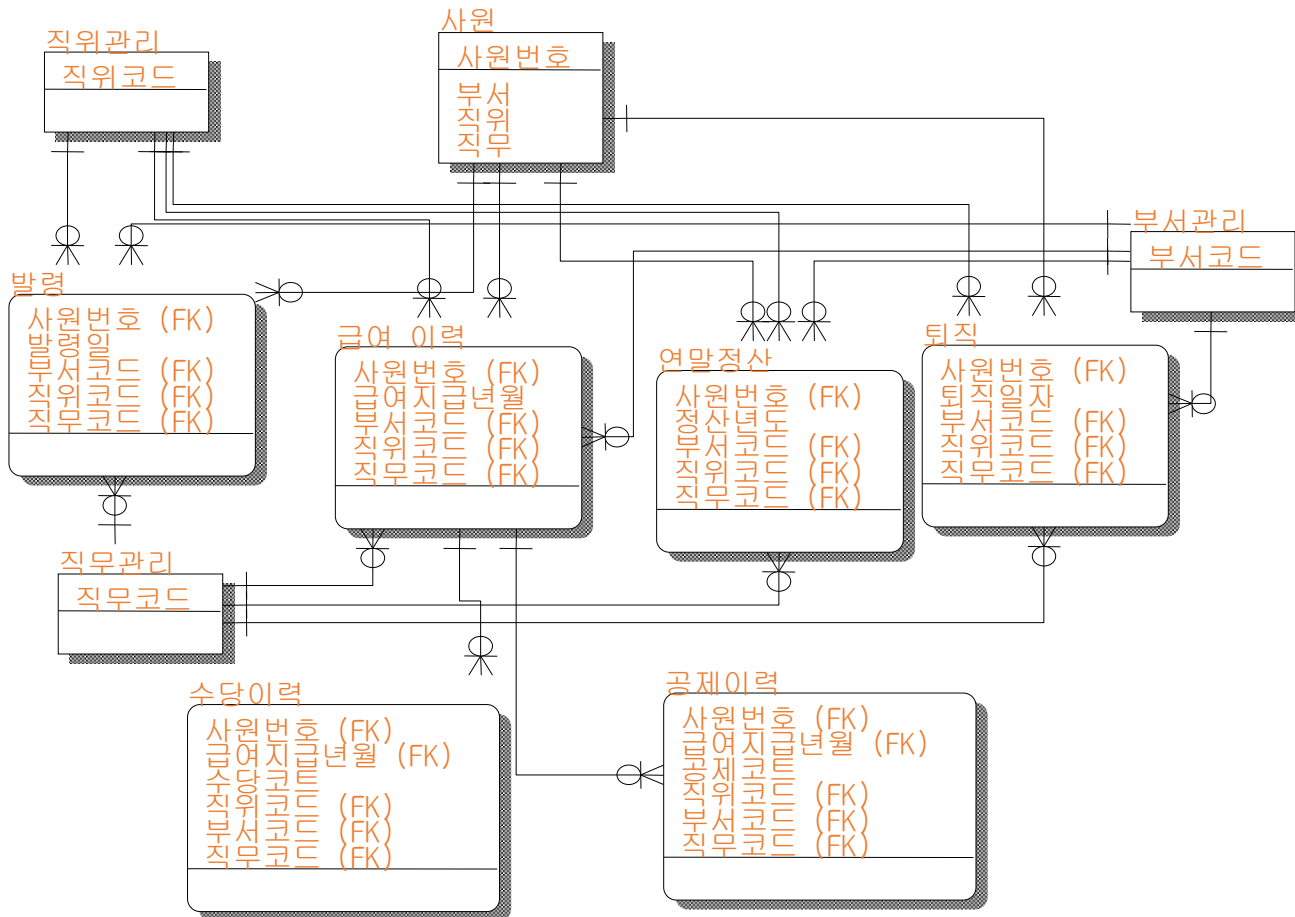
관계검토

- 업무적 흐름에 비추어 미 도출된 관계는 없는가?
- 단위 시스템담당자들의 업무협의 부족으로 인한 단위 시스템간 연계 엔티티타입간의 관계 미도출
 - 인사/급여 시스템의 급여정보가 전표 처리됨으로서 발생하는 회계시스템의 전표정보
 - 영업 시스템에서 매출을 통해 발생하는 회계시스템의 전표정보
 - 구매 시스템에서의 구매를 통해 발생하는 회계시스템의 전표 및 자산정보
 - 구매 시스템에서의 구매된 제품의 자재관리 시스템에서의 창고로의 입고 정보
 - 자재관리 시스템에서의 출고된 자재가 부서로 이동시 자산관리 시스템에서의 자산 이동 정보
 - 인사시스템의 직원 및 부서정보는 다른 모든 시스템에서 기본적으로 사용됨

4. 관계검토

관계검토

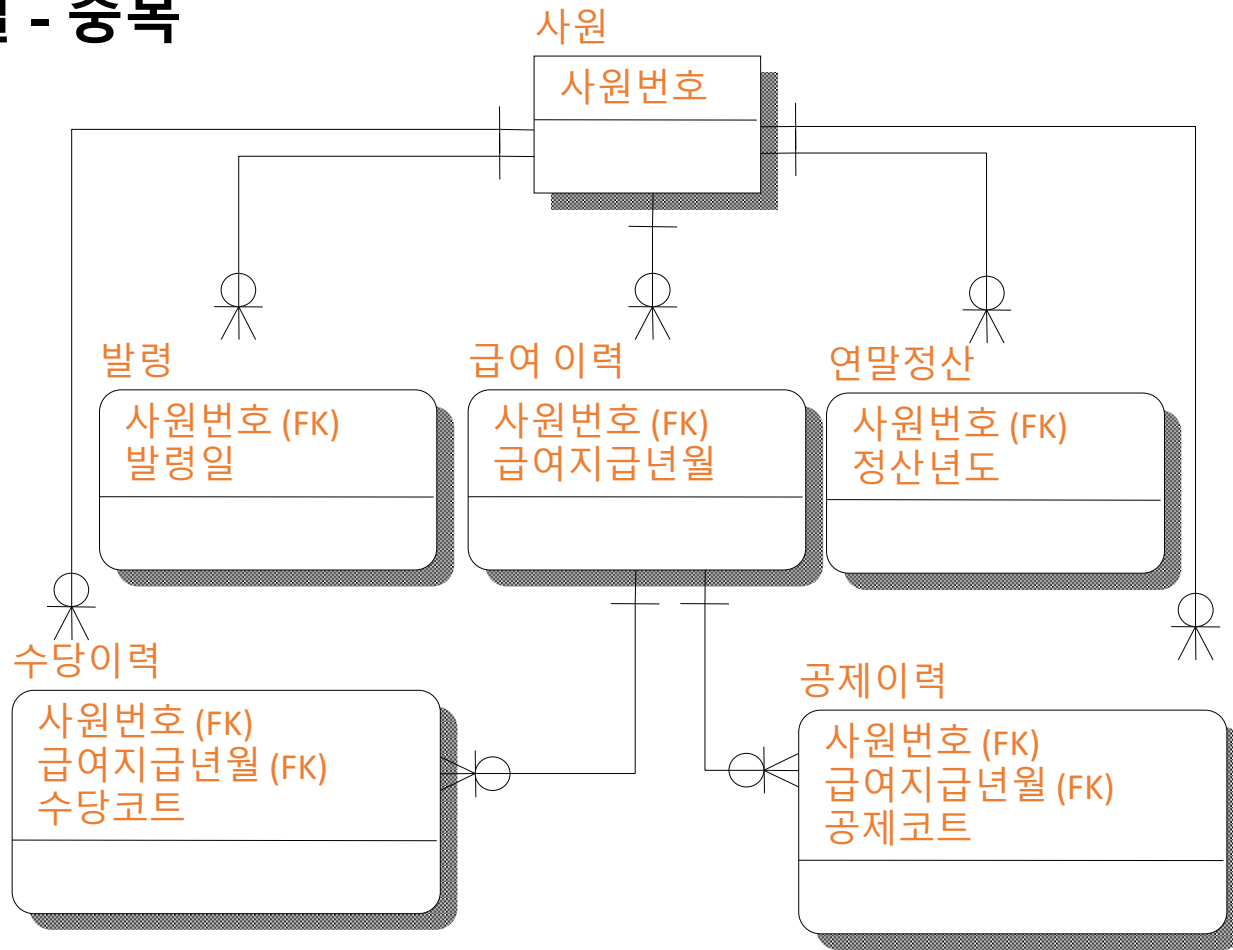
- 관계에 대한 표현은 적절한 수준에서 이루어졌는가?
- 코드성 엔티티타입의 복잡한 관계연결



4. 관계검토

관계검토

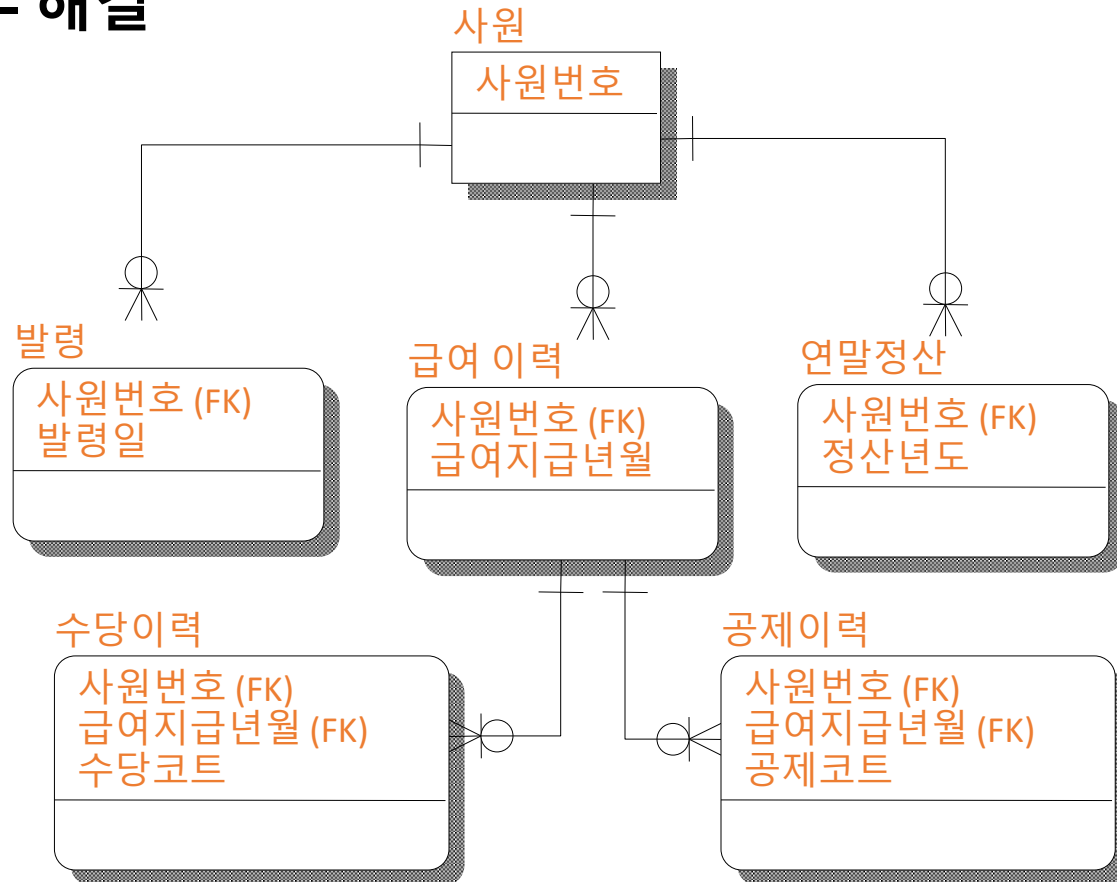
- 관계에 대한 표현은 적절한 수준에서 이루어졌는가?
- PRIMARY KEY를 상속받은 엔티티타입이 조상 엔티티타입과 관계 연결 - 중복



4. 관계검토

관계검토

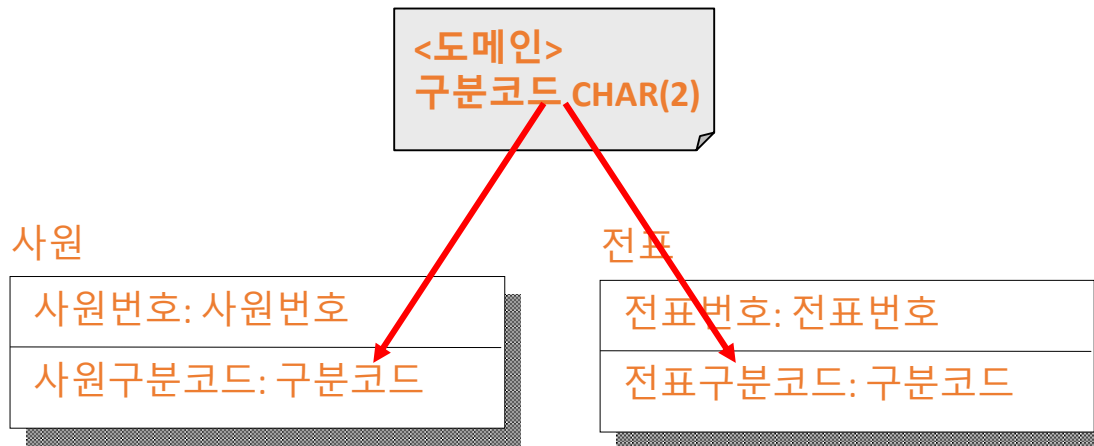
- 관계에 대한 표현은 적절한 수준에서 이루어졌는가?
- PRIMARY KEY를 상속받은 엔티티타입이 조상 엔티티타입과 관계 연결 - 해결



5. 도메인 검토

도메인 검토

- 도메인이 적절히 정의되어 관리되고 있는가?
- 도메인의 변경에 따라 속성이 변경되고 있는가?
- 도메인이 적절히 정의되어 관리되고 있는가?
- 잘못 선정된 도메인의 유형

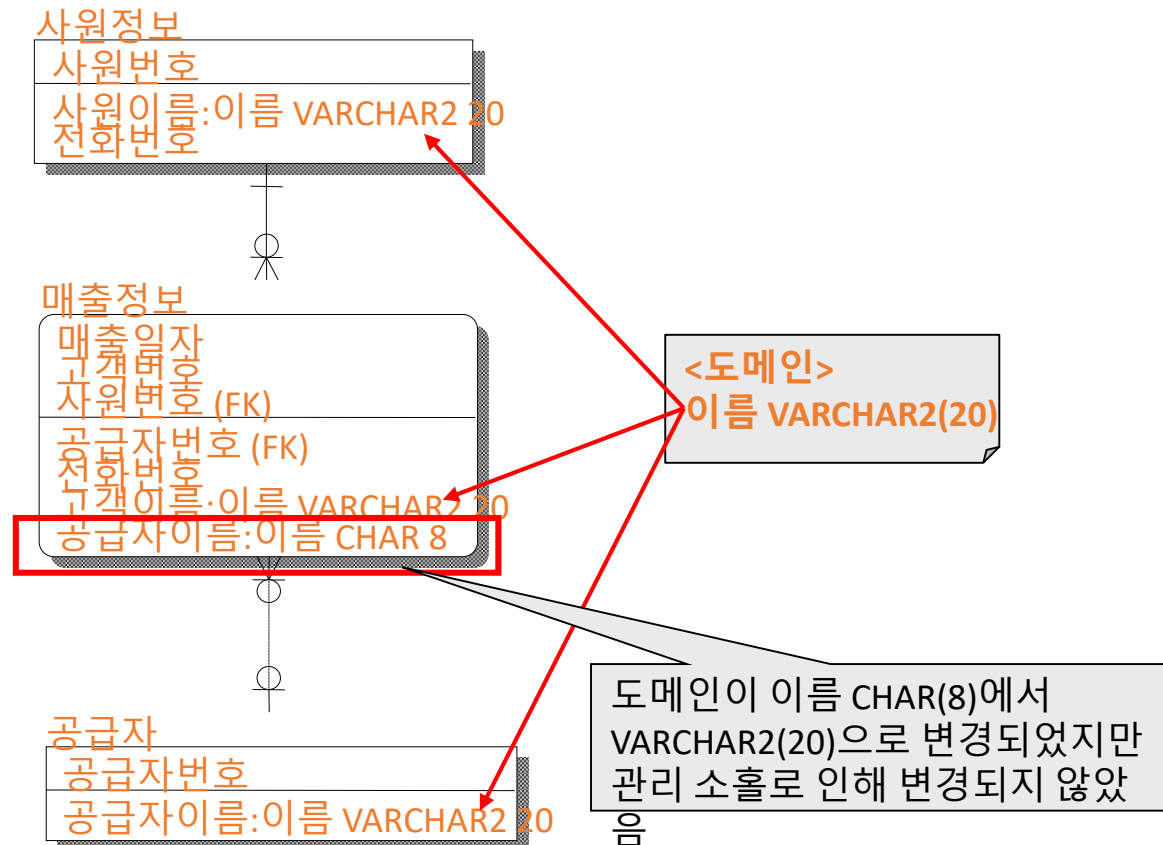


구분코드라는 동일한 도메인을 이용하도록 설계하였지만 전표구분코드가 CHAR(4)로 변경된다고 하여 사원구분코드도 CHAR(4)가 되어서는 안된다.

5. 도메인 검토

도메인 검토

- 도메인이 적절히 정의되어 관리되고 있는가?
- 관리가 소홀한 도메인 유형



『1과목』 Fundamental

8교시 :

성능모델링



학습목표

- ...할 수 있다.

눈높이 체크

- ...를 알고 계신가요?

1. 성능모델링

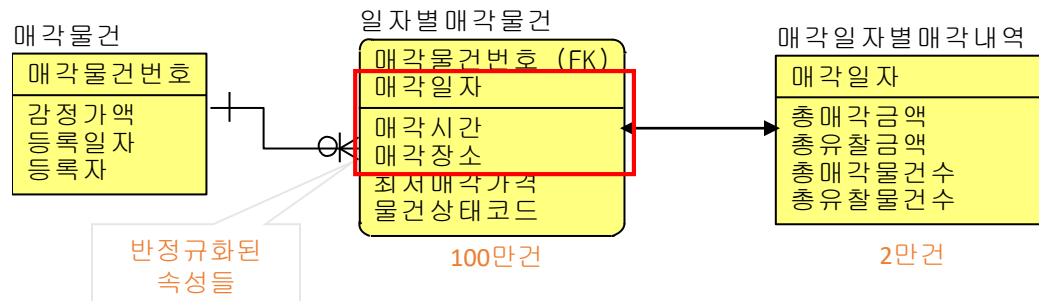
1. 성능모델링 이란?

- 성능 데이터모델링이란 데이터베이스 성능향상을 목적으로 설계단계의 데이터모델링 때부터 성능과 관련된 사항이 데이터모델링에 반영될 수 있도록 하는 것이다.

2. 정규화를 통한 성능향상

- 특정 매각장소에 대해 매각일자를 찾아 매각내역을 조회하려면 100만 건의 데이터를 읽어 매각일자를 DISTINCT하여 매각일자별매각내역과 조인이 된다.

성능이 저하된 반정규화 사례 - 엔티티타입 반정규화

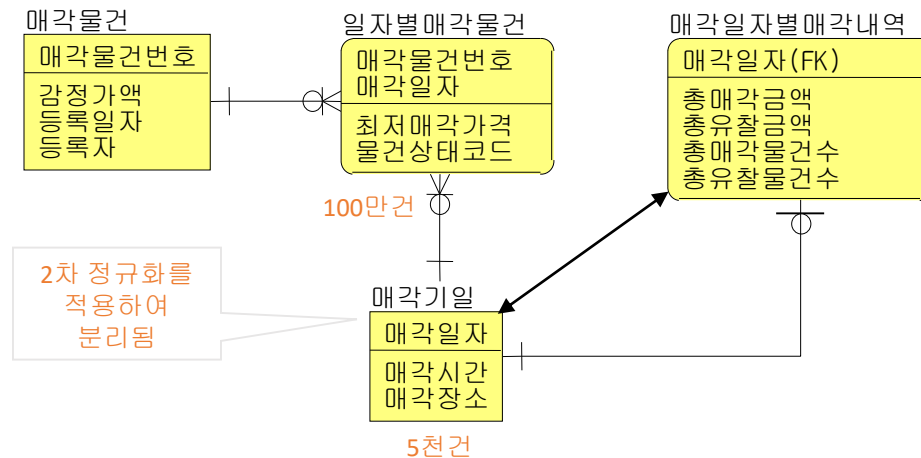


1. 성능모델링

2. 정규화를 통한 성능향상

- 특정 매각장소에 대해 매각일자를 찾아 매각내역을 조회하려면 500건의 매각기일과 매각일자별매각내역과 조인이 된다.

성능이 저하된 반정규화 사례 - 정규화를 통한 성능향상



1. 성능모델링

3. 반정규화를 통한 성능향상

● 성능이 저하된 반정규화 사례 – 속성이 중복된 경우

성능이 저하된 반정규화 사례 – 속성이 중복된 경우

모델

모델코드
모델명
제품류코드
물품가
출하가
A유형기능분류코드1
B유형기능분류코드2
C유형기능분류코드3
D유형기능분류코드4
E유형기능분류코드5
F유형기능분류코드6
G유형기능분류코드7
H유형기능분류코드8
I유형기능분류코드9
바코드
가로
세로
높이
모델구분
...

기존 인덱스 8개

유형기능분류코드단위로 데이터처리를 해야
하여 인덱스를 생성하려면 9개의 인덱스를
생성해야 함

```
CREATE INDEX INDEX_01 ON 모델( A유형기능분류코드1 );  
CREATE INDEX INDEX_02 ON 모델( A유형기능분류코드2 );  
CREATE INDEX INDEX_03 ON 모델( A유형기능분류코드3 );  
CREATE INDEX INDEX_04 ON 모델( A유형기능분류코드4 );  
CREATE INDEX INDEX_05 ON 모델( A유형기능분류코드5 );  
CREATE INDEX INDEX_06 ON 모델( A유형기능분류코드6 );  
CREATE INDEX INDEX_07 ON 모델( A유형기능분류코드7 );  
CREATE INDEX INDEX_08 ON 모델( A유형기능분류코드8 );  
CREATE INDEX INDEX_09 ON 모델( A유형기능분류코드9 );
```

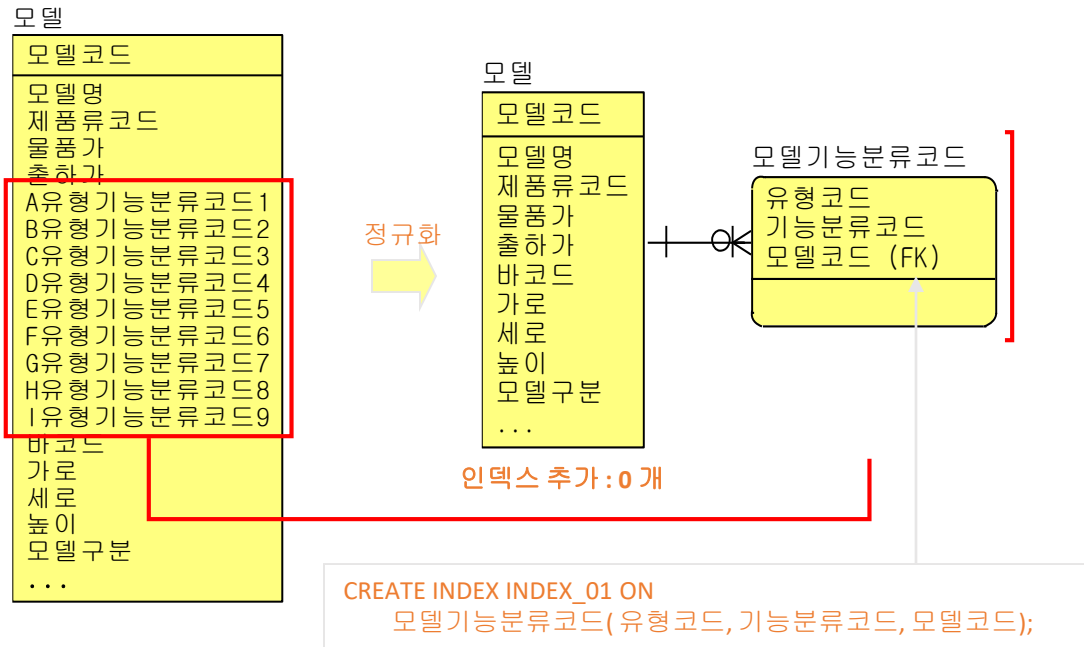
인덱스 추가 + 9 개 ???

1. 성능모델링

3. 반정규화를 통한 성능향상

● 성능이 저하된 반정규화 사례 – 속성이 중복된 경우

성능이 저하된 반정규화 사례 – 속성이 중복된 경우

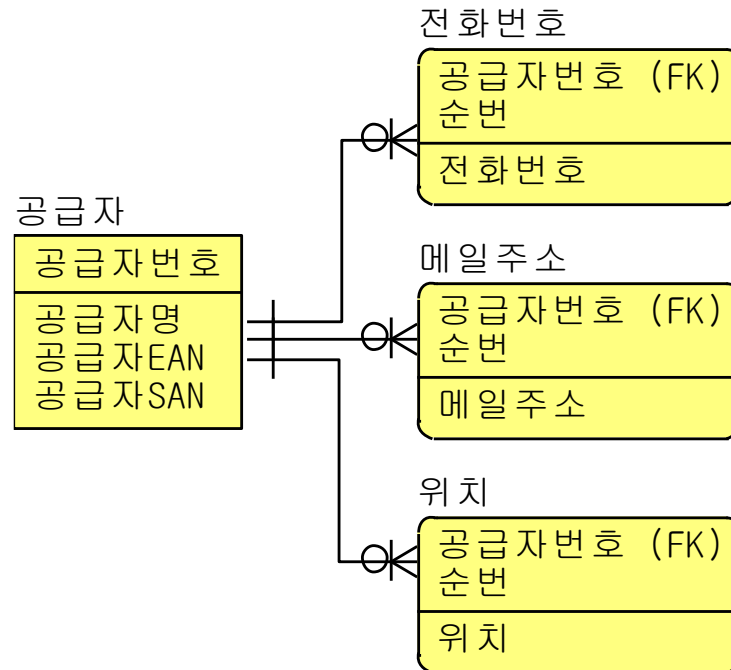


1. 성능모델링

3. 반정규화를 통한 성능향상

● 반정규화를 통한 성능향상 사례

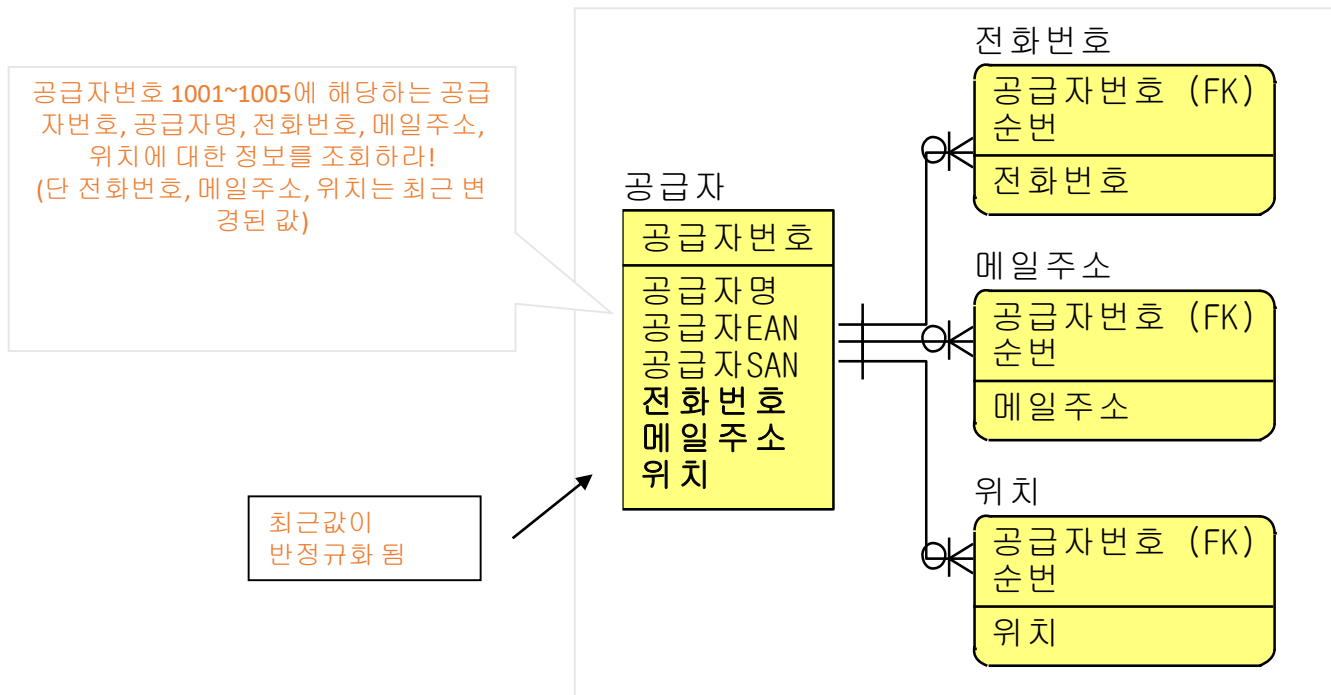
공급자번호 1001~1005에 해당하는 공급자번호, 공급자명, 전화번호, 메일주소, 위치에 대한 정보를 조회하라!
(단 전화번호, 메일주소, 위치는 최근 변경된 값)



1. 성능모델링

3. 반정규화를 통한 성능향상

● 반정규화를 통한 성능향상 사례

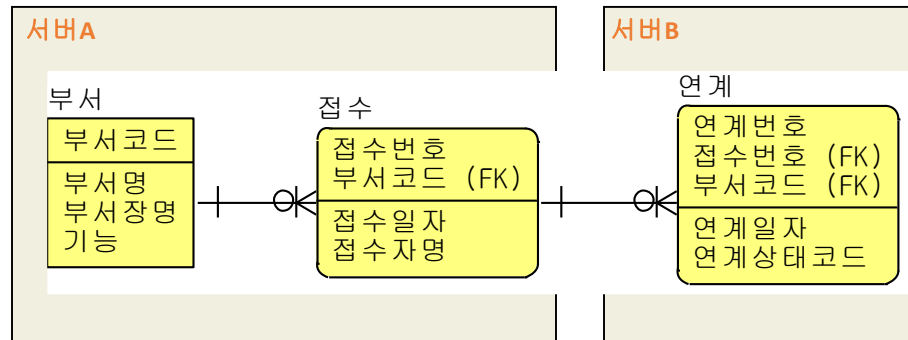


1. 성능모델링

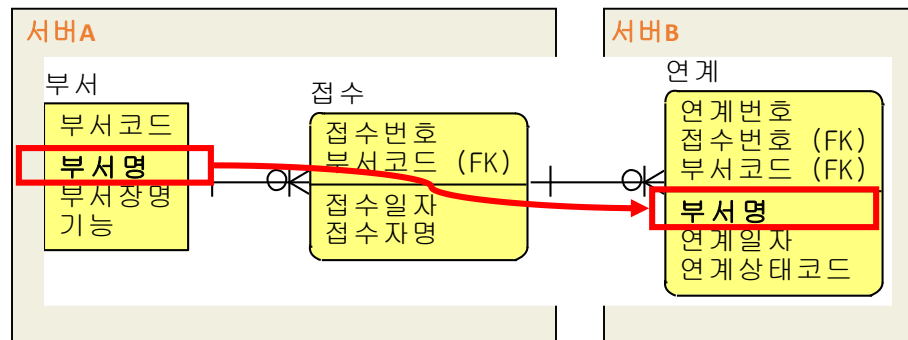
3. 반정규화를 통한 성능향상

● 반정규화를 통한 성능향상 사례

반정규화를 통한 성능향상 사례 - 다른 서버간



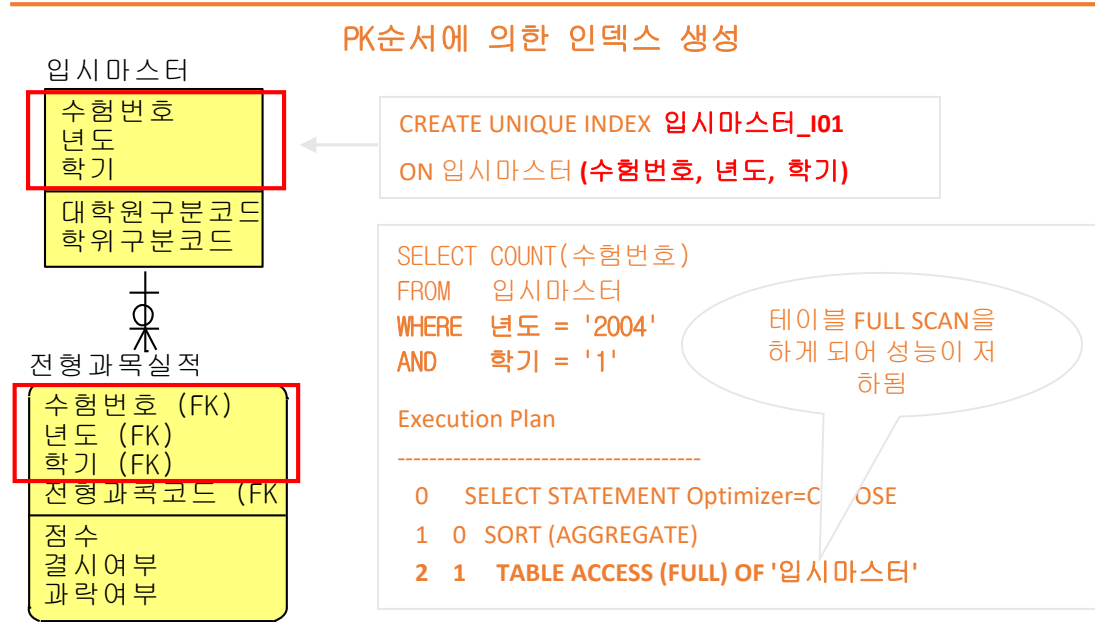
반정규화를 통한 성능향상 사례 - 다른 서버간



1. 성능모델링

4. PK순서 조정을 통한 성능향상

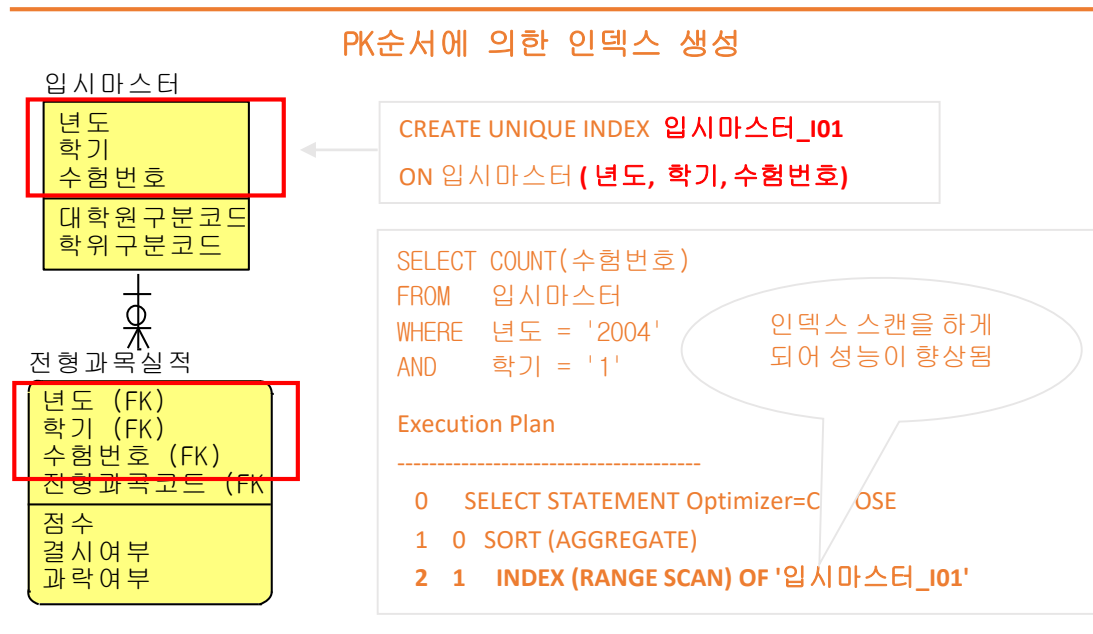
● PK순서에 의한 인덱스 생성



1. 성능모델링

4. PK순서 조정을 통한 성능향상

● PK순서에 의한 인덱스 생성



4. PK순서 조정을 통한 성능향상

● PK순서에 의한 인덱스 생성

현금출급기실적

거래일자
사무소코드
출급기번호
명세표번호

건수
금액

PK순서에 의한 인덱스 생성

```
SELECT 건수, 금액  
FROM 현금출급기실적  
WHERE 거래일자 BETWEEN '20040701' AND '20040702'  
AND 사무소코드 = '000368'
```

Execution Plan

```
0  SELECT STATEMENT Optimizer=CHOOSE  
1  0  SORT (AGGREGATE)  
2  1  INDEX (RANGE SCAN) OF ' 현금출급기실적_I01 ' (NON-UNIQUE) (Cost=1 Card=1)
```

인덱스를 이용하였으나 최적화된 인덱스 이용이 안되었음

4. PK순서 조정을 통한 성능향상

● PK순서에 의한 인덱스 생성

PK순서에 의한 인덱스 생성



** 넓은 범위조회 인한 성능저하

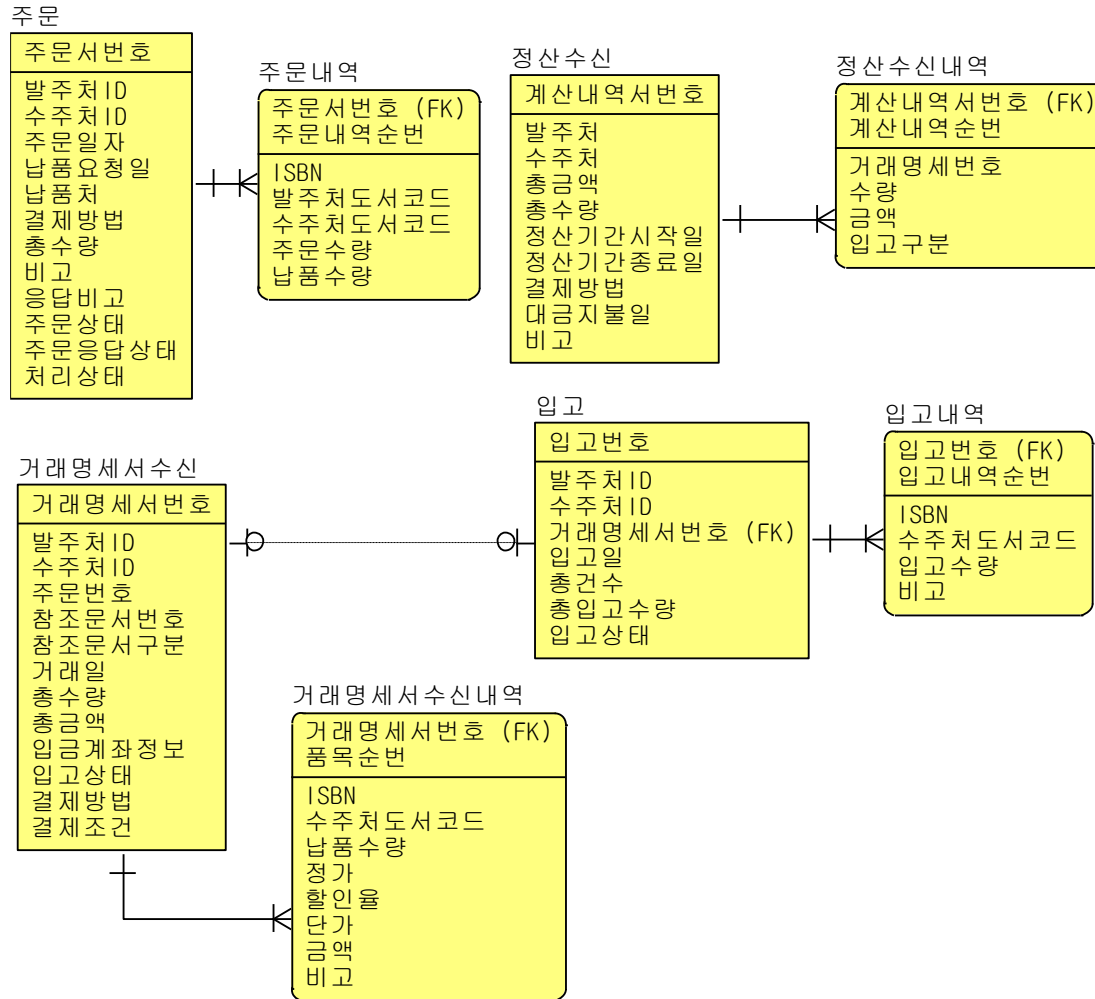


** 좁은 범위조회로 인한 성능향상

1. 성능모델링

5. 데이터모델 단순화를 통한 성능향상

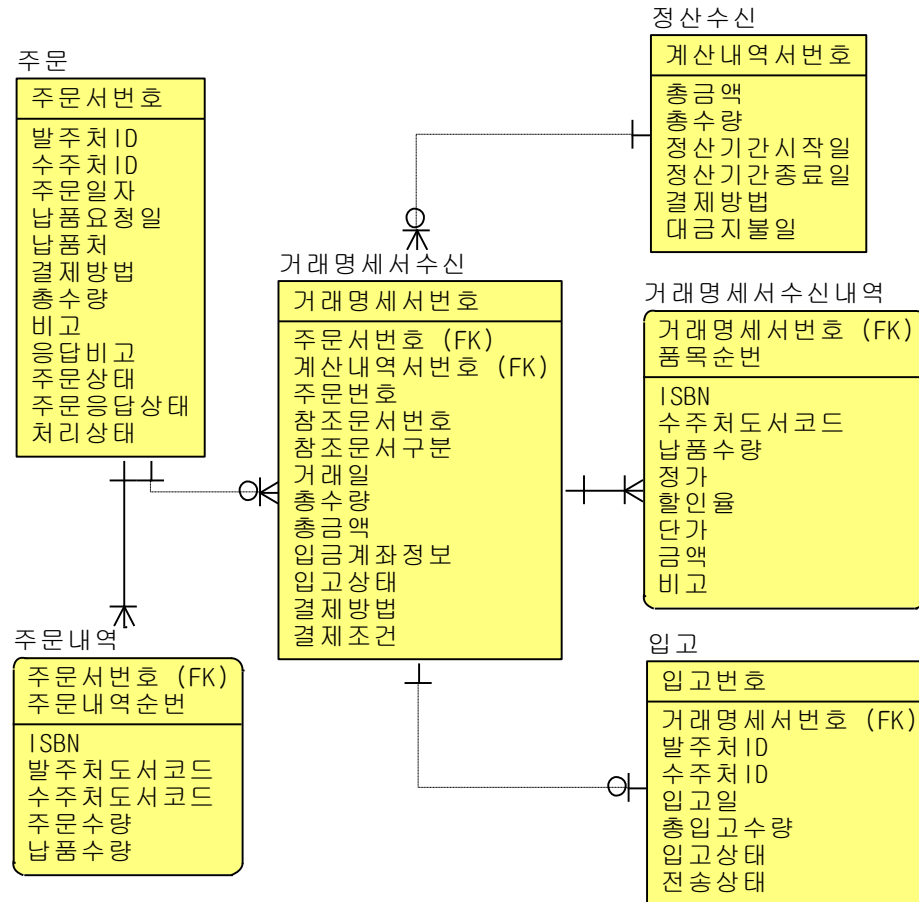
● 복잡한 데이터모델



1. 성능모델링

5. 데이터모델 단순화를 통한 성능향상

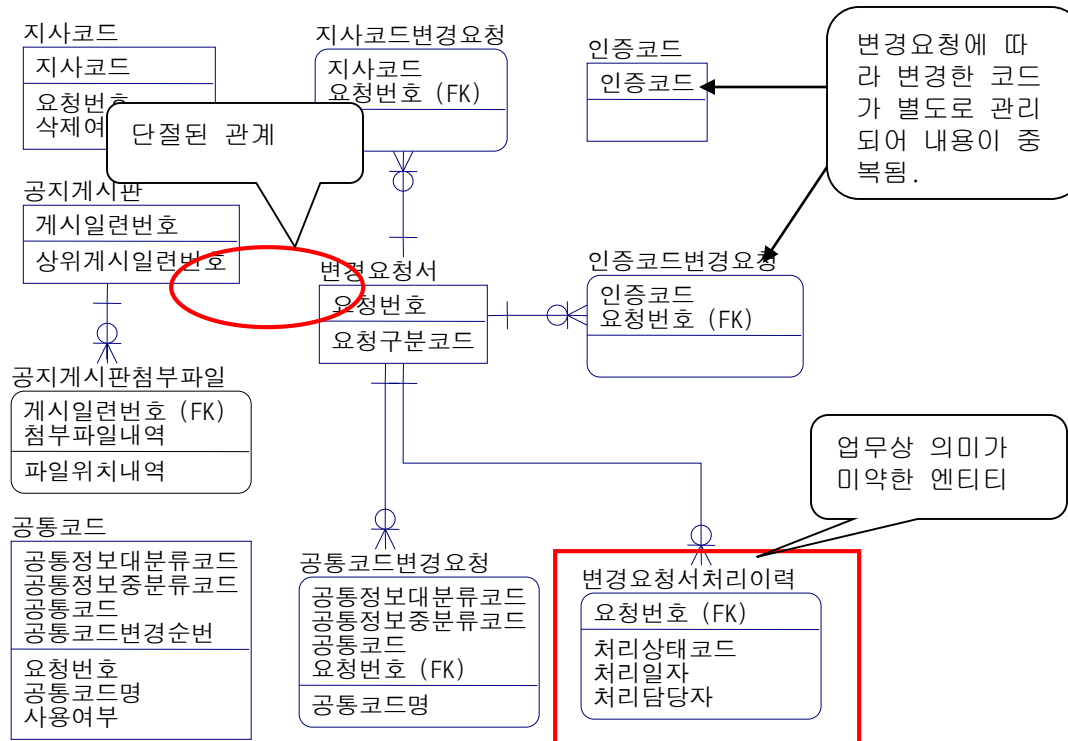
● 정리된 데이터모델



1. 성능모델링

5. 데이터모델 단순화를 통한 성능향상

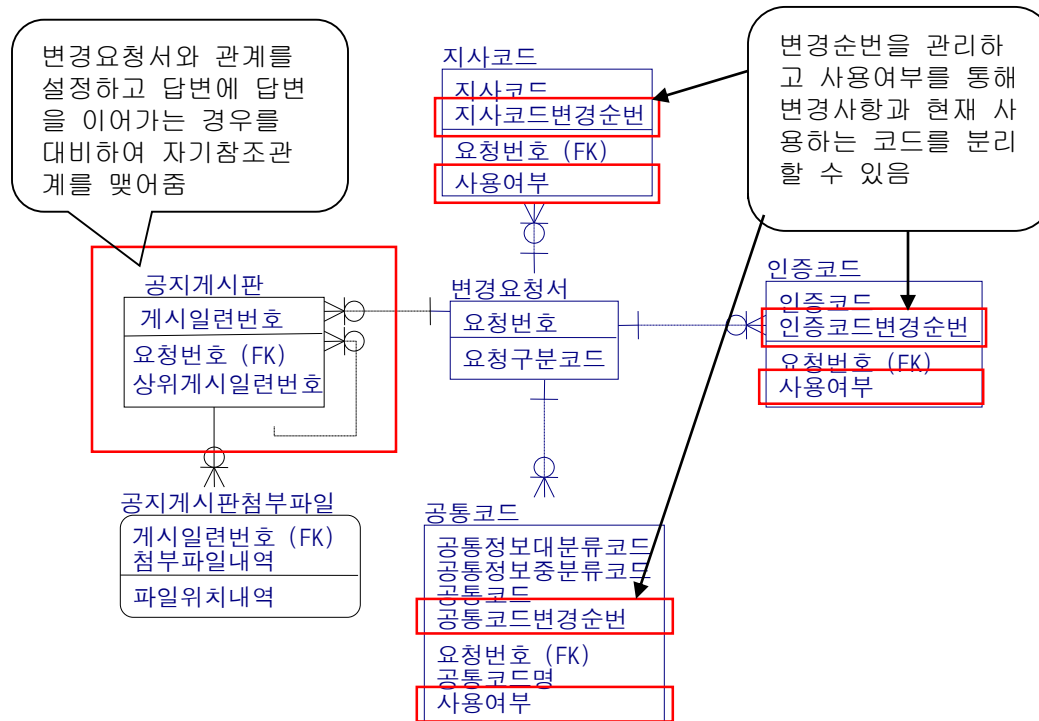
● 변경요청에 따른 코드관리 모델



1. 성능모델링

5. 데이터모델 단순화를 통한 성능향상

● 변경요청에 따른 코드관리 모델 - 가이드

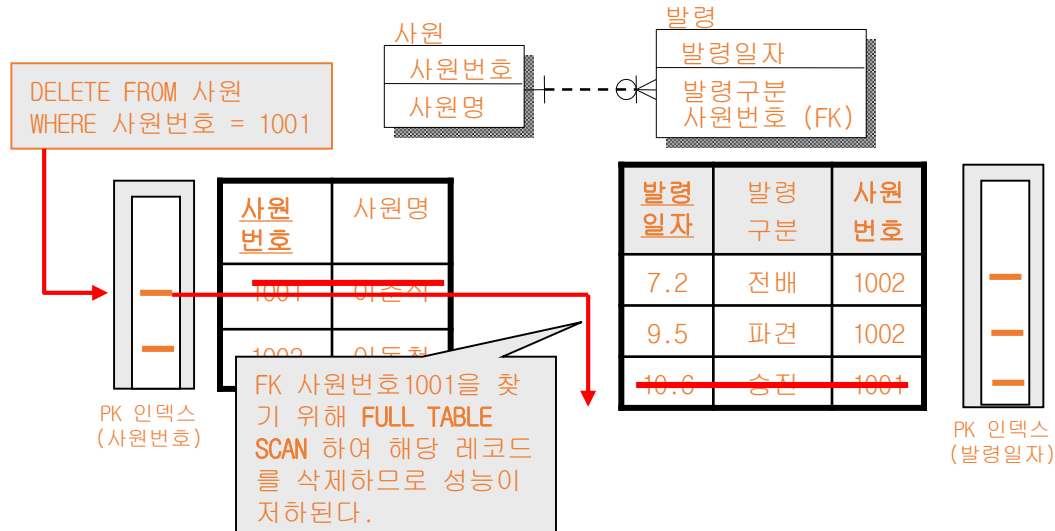


1. 성능모델링

6. FK 인덱스를 통한 성능향상

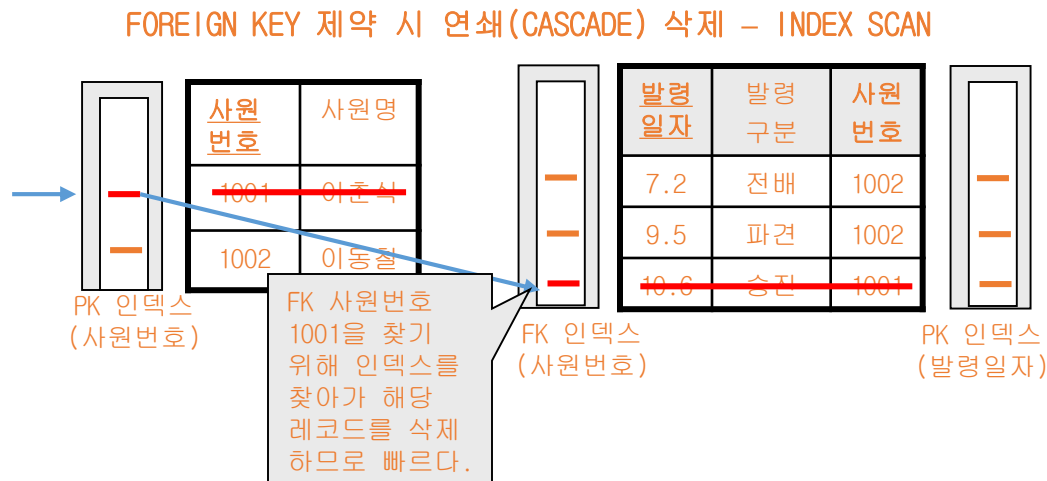
● FOREIGN KEY 제약 시 연쇄(CASCADE) 삭제 – FULL TABLE SCAN

FOREIGN KEY 제약 시 연쇄(CASCADE) 삭제 – FULL TABLE SCAN



6. FK 인덱스를 통한 성능향상

● FOREIGN KEY 제약 시 연쇄(CASCADE) 삭제 – INDEX SCAN

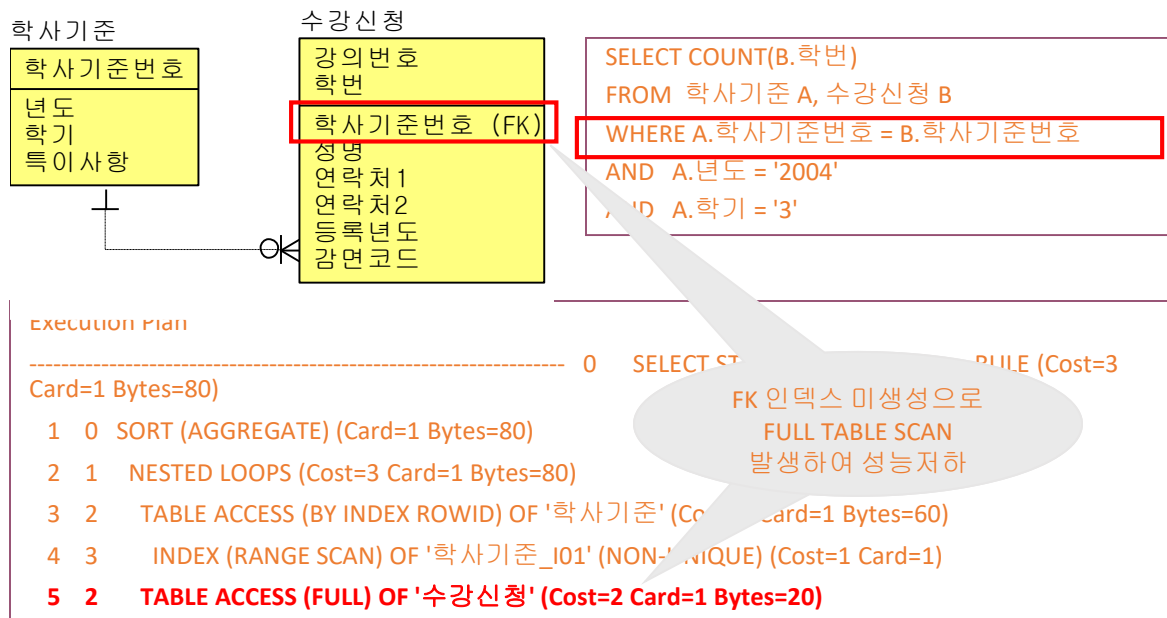


1. 성능모델링

6. FK 인덱스를 통한 성능향상

● FK 인덱스 미생성으로 인한 성능저하

FK 인덱스 미생성으로 인한 성능저하

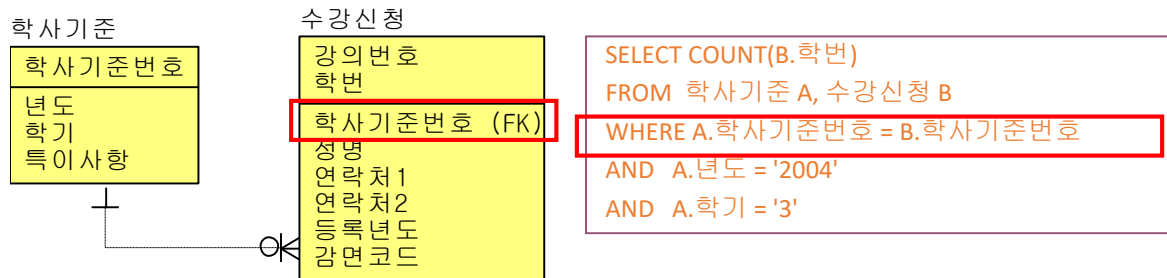


1. 성능모델링

6. FK 인덱스를 통한 성능향상

● FK 인덱스 생성으로 인한 성능향상

FK 인덱스 생성으로 인한 성능향상



인덱스 생성 : CREATE INDEX 수강신청_FK01 ON 수강신청(학사기준번호)

Execution Plan

```
0  SELECT STATEMENT Optimizer=RULE (Cost=2 Card=1)
1  0  SORT (AGGREGATE) (Card=1 Bytes=80)
2  1  NESTED LOOPS (Cost=2 Card=1 Bytes=80)
3  2  TABLE ACCESS (BY INDEX ROWID) OF '학사기준'
4  3  INDEX (RANGE SCAN) OF '학사기준_I01' (NON-UNIQUE)
5  2  INDEX (RANGE SCAN) OF '수강신청_FK01' (NON-UNIQUE)
```

FK 인덱스 생성으로
INDEX SCAN
발생하여 성능향상

7. 이력모델의 성능향상

● 발생이력/변경이력에서 최신여부에 컬럼 이 없을 경우

발생이력/변경이력에서 최신여부에 컬럼 이 없을 경우

접수통계

사업소코드
변경일자
접수물량
접수구분코드

```
SELECT A.사업소코드, SUM(A.접수물량)
FROM 접수통계 A,(SELECT 사업소코드, MAX(변경일자)
                  FROM 접수통계
                  WHERE 접수구분코드 = '01'
                  GROUP BY 사업소코드) B
WHERE A.사업소코드 = B.사업소코드
GROUP BY A.사업소코드
```

Execution Plan

```
0  SELECT STATEMENT Optimizer=CHOOSE
1  0  SORT (GROUP BY)
2  1   TABLE ACCESS (BY INDEX ROWID) OF '접수통계'
3  2    NESTED LOOPS
4  3     VIEW
5  4      SORT (GROUP BY)
6  5       TABLE ACCESS (BY INDEX ROWID) OF '접수통계'
7  6        INDEX (UNIQUE SCAN) OF '접수통계_I01' (UNIQUE)
8  3         INDEX (RANGE SCAN) OF 'PIAMT0010_PK' (UNIQUE)
```

인라인뷰의 그룹함수를
사용해 최근에 변경된
값을 가져오므로 성능
저하가 나타남

7. 이력모델의 성능향상

● 발생이력/변경이력에서 최신여부를 생성한 경우

발생이력/변경이력에서 최신여부를 생성한 경우

접수통계

사업소코드
변경일자
접수물량
접수구분코드
최신여부

```
SELECT 사업소코드, SUM(접수물량)
FROM 접수통계
WHERE 접수구분코드 = '01'
AND 최신여부 = 'Y'
GROUP BY 사업소코드
```

Execution Plan

```
0  SELECT STATEMENT Optimized
1  0  SORT (GROUP BY)
2  1  TABLE ACCESS (BY INDEX ROWID) OF '접수통계'
3  2  INDEX (RANGE SCAN) OF '접수통계_I01' (NON-UNIQUE)
```

WHERE절에서 바로 최신값을 판단하므로 성능이 향상됨

1. 성능모델링

7. 이력모델의 성능향상

● 진행이력에서 상태종료가 없는 경우

진행이력에서 상태종료가 없는 경우

기관정보

기관코드
적용일자
기관거래등급

```
SELECT A.기관코드, A.기관거래등급
FROM 기관정보 A,(SELECT 기관코드, MAX(적용일자)
                  FROM 기관정보
                  WHERE 적용일자 <= '20040701'
                  GROUP BY 기관코드) B
WHERE A.기관코드 = B.기관코드
AND A.적용일자 = B.적용일자
```

Execution Plan

```
0  SELECT STATEMENT Optimizer=CHOOSE
1  0  TABLE ACCESS (BY INDEX ROWID) OF '국가정보'
2  1  NESTED LOOPS
3  2  VIEW
4  3  SORT (GROUP BY)
5  4  TABLE ACCESS (FULL) OF '기관정보'
6  2  INDEX (RANGE SCAN) OF '국가정보_PK' (UNIQUE)
```

인라인뷰의 그룹함수를
사용해 값을 가져오므로
FULL TABLE SCAN이나 인
덱스 스캔의 범위가 넓어
져 성능이 저하됨

7. 이력모델의 성능향상

● 진행이력에서 상태종료가 있는 경우

진행이력에서 상태종료가 있는 경우

기관정보

기관코드
적용일자

적용종료일자
기관거래등급

```
SELECT A.기관코드, A.기관거래등급  
FROM 기관정보 A
```

```
WHERE 적용일자 <= '20040701'  
AND 적용종료일자 > '20040701'
```

Execution Plan

```
0  SELECT STATEMENT Optimizer=CHOOSE  
1  0  TABLE ACCESS (BY INDEX ROWID) OF '기관정보'  
2  1  INDEX (RANGE SCAN) OF '국가정보_I01' (UNIQUE)
```

WHERE절에서 바로
조건을 판단할 수 있
으므로 성능이 향상
됨

1. 성능모델링

8. 슈퍼타입/서브타입 모델의 성능향상

- 슈퍼타입/서브타입 개별 트랜잭션의 유형

The screenshot shows a web application interface with two main sections. The top section is titled '이철수 매수자로서 정보' (Information as Buyer Lee Cheol-su) and contains a form with the following fields:

매수보증금 :	23,000,000원	매수인구분 :	개인
입찰금액 :	230,000,000원	참여건수 :	7

The bottom section is titled '당사자공통정보조회화면' (Common Information Search for Parties) and contains a table with the following data:

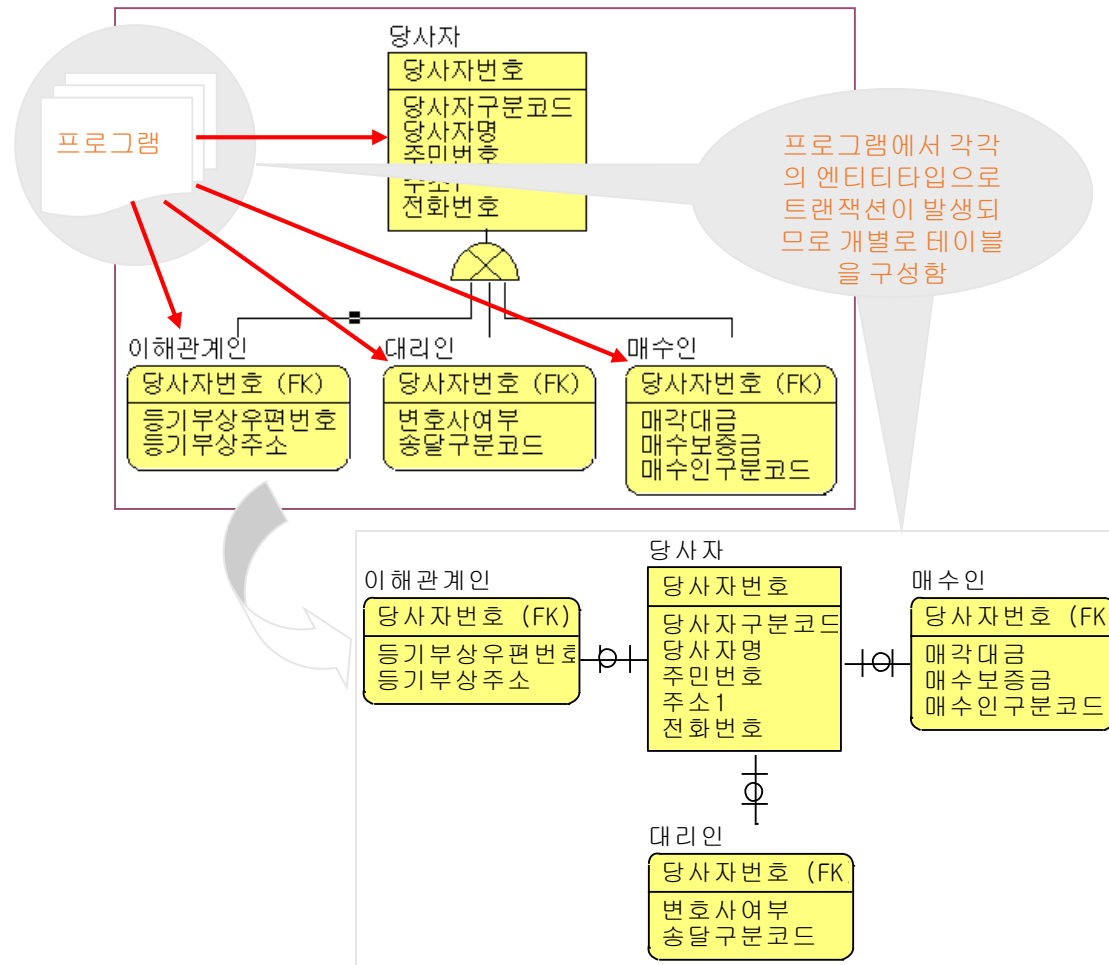
지역	사반리	조회
당사자구분 :	이해관계인	이름 : 이화수 [조회]
당사자구분 :	매수인	이름 : 이철수 [조회]
당사자구분 :	대리인	이름 : 이평수 [조회]

A red arrow points from the '이철수' (Lee Cheol-su) entry in the table to the '이철수 매수자로서 정보' form. A red arrow also points to the '조회' button next to the '이철수' entry, with the label '선택' (Select) next to it.

1. 성능모델링

8. 슈퍼타입/서브타입 모델의 성능향상

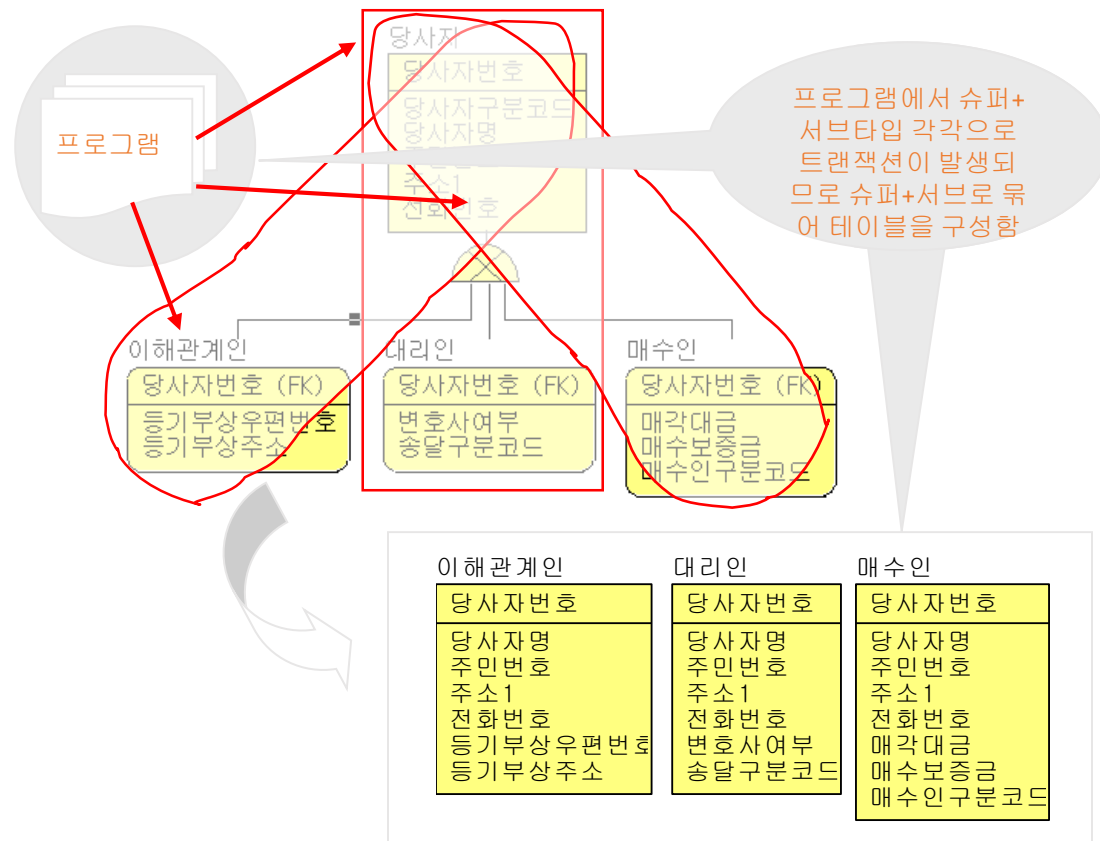
● 개별 엔티티타입으로 발생하는 트랜잭션



1. 성능모델링

8. 슈퍼타입/서브타입 모델의 성능향상

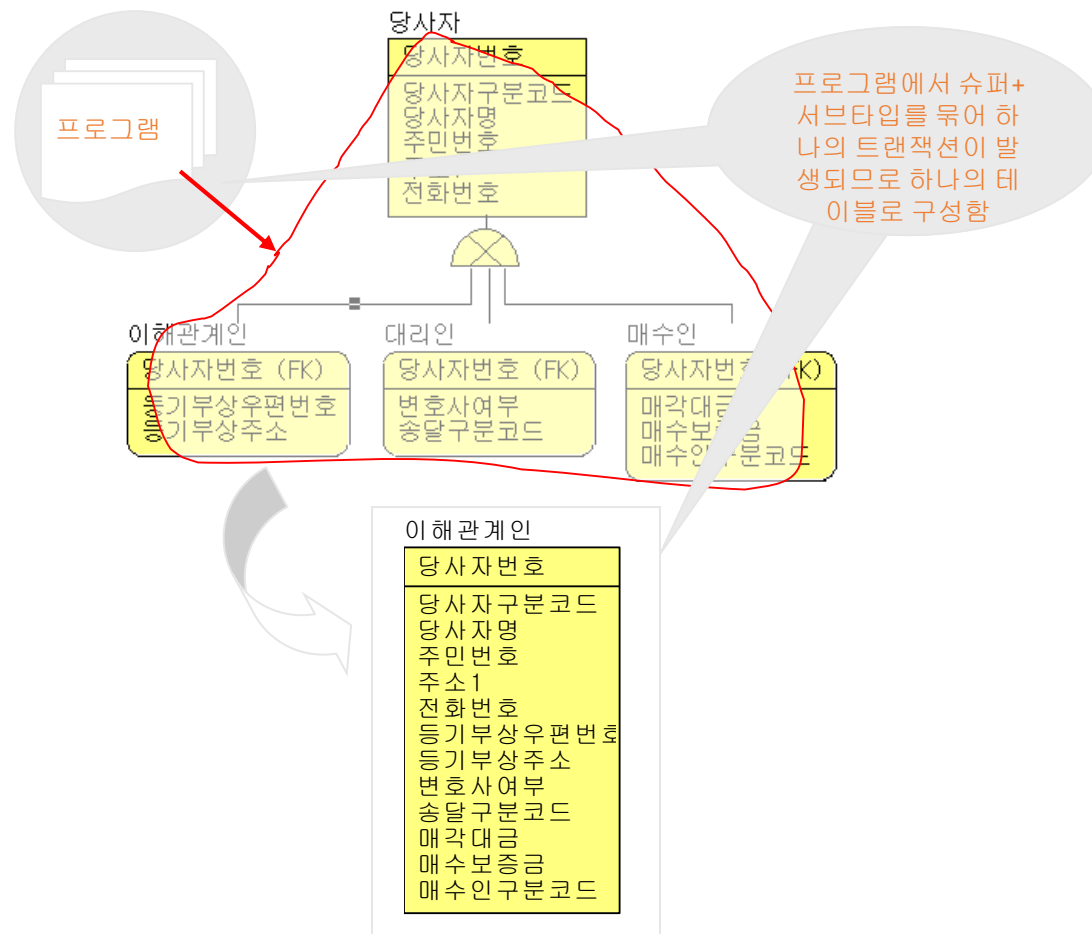
- 슈퍼+서브 각각의 엔티티타입으로 발생하는 트랜잭션



1. 성능모델링

8. 슈퍼타입/서브타입 모델의 성능향상

- 슈퍼+서브 전체엔티티타입으로 발생하는 트랜잭션



2. 채번테이블 성능

1. 성능저하

● 채번 테이블 사용으로 인한 성능저하

채번 테이블 사용으로 인한 성능저하

채번테이블

채번구분 기준일자
채번

트랜잭션LOG

로그번호
로그일자 로그시간 터미널번호 거래점번호

처리시간이 길수록 채번
테이블에 잠금현상으로
인해 다른 트랜잭션은 대
기상태에 있어 성능이 저
하됨

① 채번테이블 수정 : 채번테이블에 잠금현상(LOCK) 유발

UPDATE 채번테이블 SET 채번 = 채번 + 1 WHERE 채번구분 = '트랜잭션LOG'

② 대상테이블입력 : 채번테이블 조회하여 채번한 내용을 대상 테이블에 입력

INSERT INTO 트랜잭션LOG

VALUES((SELECT 채번 AS 로그번호

FROM 채번테이블 WHERE 채번구분 = '트랜잭션LOG'),

'20040701','13:00','A01','ED')

③ 트랜잭션 종료 : 잠금현상 해제

COMMIT;

잠금현상지속

2. 채번테이블 성능

1. 성능저하

● 채번 테이블 사용으로 인한 성능저하

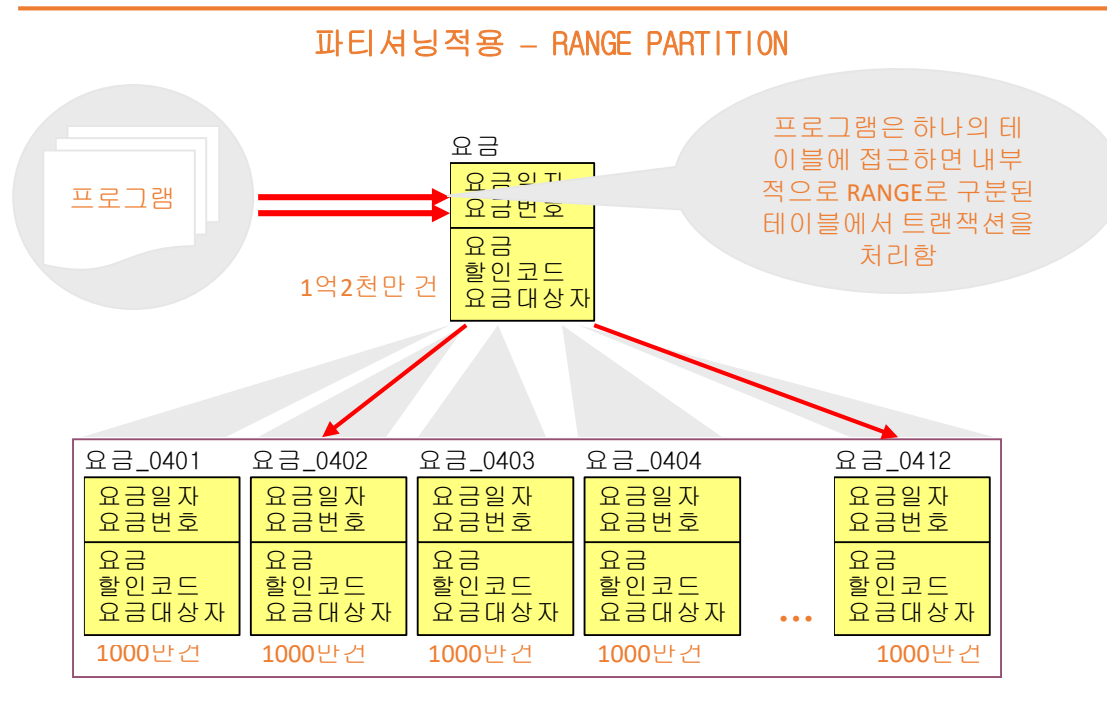
방법1) 채번 테이블	방법2) 테이블에 최대값 적용	방법3) 시퀀스오브젝트 이용
<div>채번테이블</div> <div>채번구분</div> <div>채번</div>	<pre>INSERT INTO 테이블 (일련번호, COL2, COL3...) SELECT DECODE(MAX(번호),NULL,1, MAX(번호)+1) 일련번호 FROM 트랜잭션 WHERE ROWNUM = 1;</pre>	<pre>create sequence 테이블_seq increment by 1 start with 1 maxvalue 1000000000 cache 30;</pre>
장점 - DUP에러 없음 단점 - LOCK유발 성능저하 관리항목증가	장점 - 관리항목 증가 없음 빠른 성능 단점 - 이론적 DUP에러가능	장점 - 빠른 성능, DUP에러 없음, LOCK없음 단점 - 관리항목 증가

테이블에 최대값을 이용하여 입력하는 방법2와 시퀀스 오브젝트를 생성하여
입력하는 방법3이 실전 프로젝트에서는 좋은 성능을 보여준다

3. 파티셔닝 적용

파티셔닝적용 – RANGE PARTITION

● 파티셔닝적용 – RANGE PARTITION

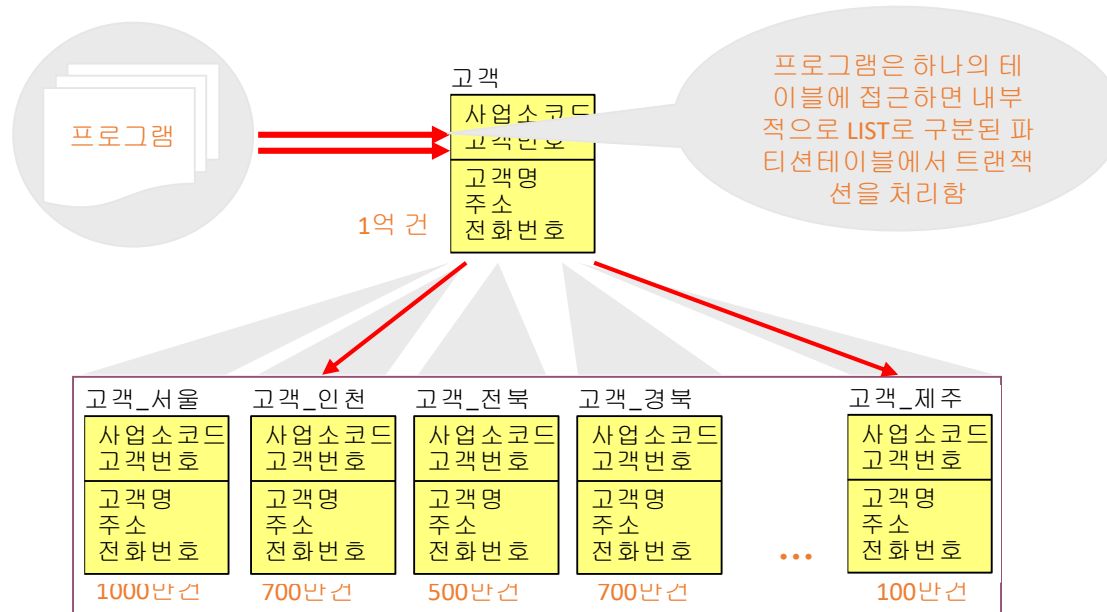


3. 파티셔닝 적용

파티셔닝적용 – RANGE PARTITION

● 파티셔닝적용 – RANGE PARTITION

파티셔닝적용 – LIST PARTITION



4. 컬럼분리

컬럼분리

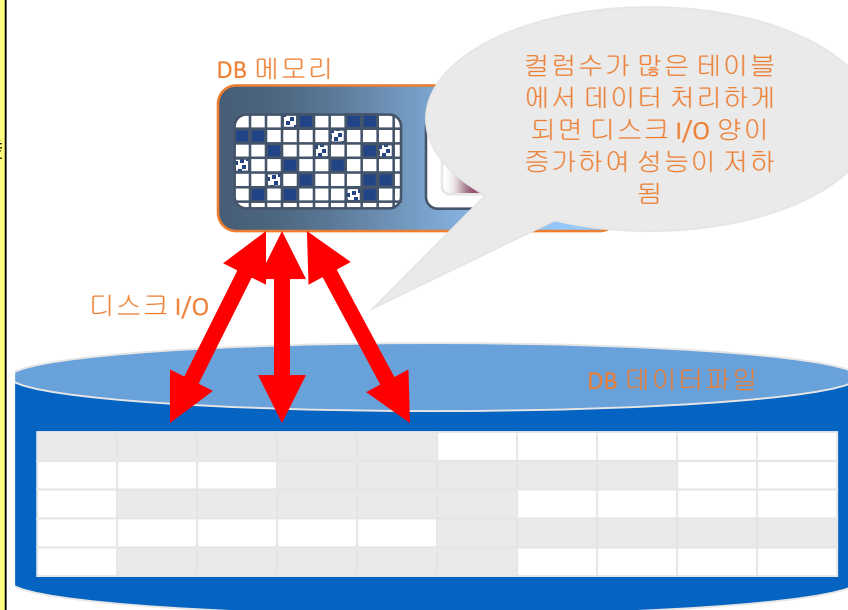
● 컬럼수가 많은 테이블

도서정보

도서번호
발행기관명
제품번호
출판사
국제표준번호
제품형태
수량
전자출판유형
전자출판포맷
전자출판소스
전자출판기술
회의의날자
회의의명
회의의장소
단체제
단체제
단체제
단체제
공고일
발행일
초판년
판권년
전체높이
전체넓이
...

컬럼수가 많은 테이블

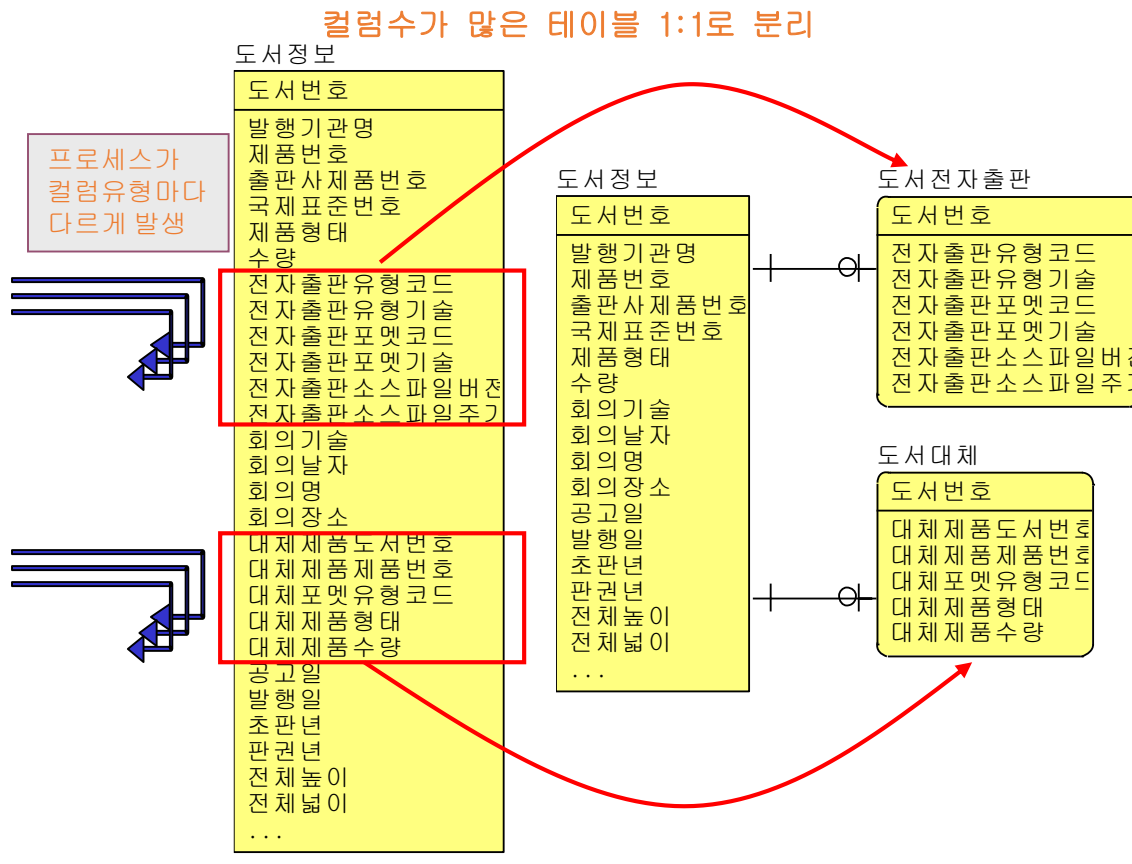
```
SELECT 발행기관명,수량,공고일,발행일
FROM 도서정보
WHERE 초판년='2002'
```





컬럼분리

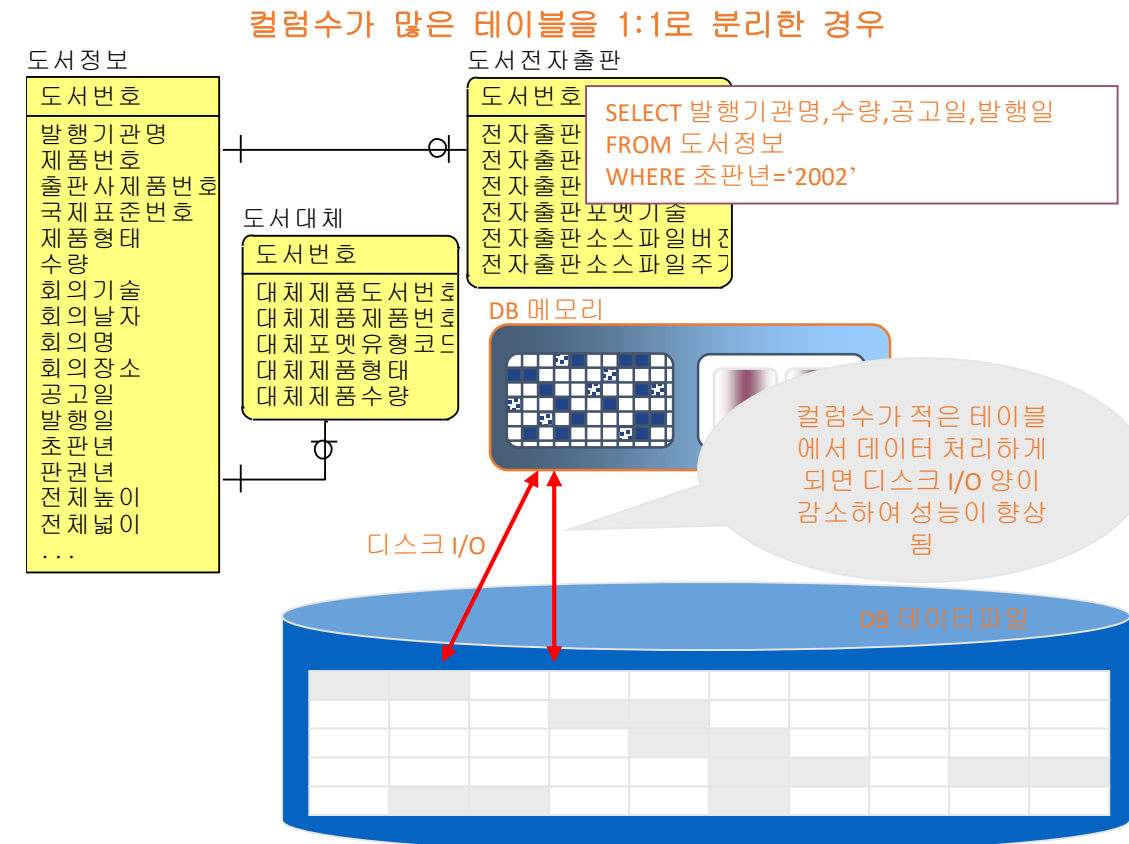
- 컬럼수가 많은 테이블 1:1로 분리



4. 컬럼분리

컬럼분리

- 컬럼수가 많은 테이블을 1:1로 분리한 경우



5. 적절한 데이터타입 사용

적절한 데이터타입 사용

● CHAR 타입 변형으로 성능저하

CHAR 타입 변형으로 성능저하

사용자

사용자 ID: CHAR(10)
사용자명: VARCHAR2(20)
전화번호: VARCHAR2(16)
주소: VARCHAR2(40)

인덱스: 사용자_I01(사용자ID)

```
SELECT 사용자명  
FROM 사용자  
WHERE RTRIM(사용자ID) = 'perfDB'
```

Execution Plan

```
0  SELECT STATEMENT Optimizer=CHOOSE  
1  0  TABLE ACCESS (FULL) OF '사용자'
```

인덱스가 있음에도 불구하고
인덱스컬럼 변형으로 인해
FULL TABLE SCAN 발생으로
성능저하

5. 적절한 데이터타입 사용

적절한 데이터타입 사용

● VARCHAR2 타입 이용으로 성능향상

VARCHAR2 타입 이용으로 성능향상

사용자

사용자ID: VARCHAR2(10)
사용자명: VARCHAR2(20)
전화번호: VARCHAR2(16)
주소: VARCHAR2(40)

```
SELECT 사용자명  
FROM 사용자  
WHERE 사용자ID = 'perfDB'
```

인덱스: 사용자_I01(사용자ID)

인덱스를 이용
하여 성능향상

Execution Plan

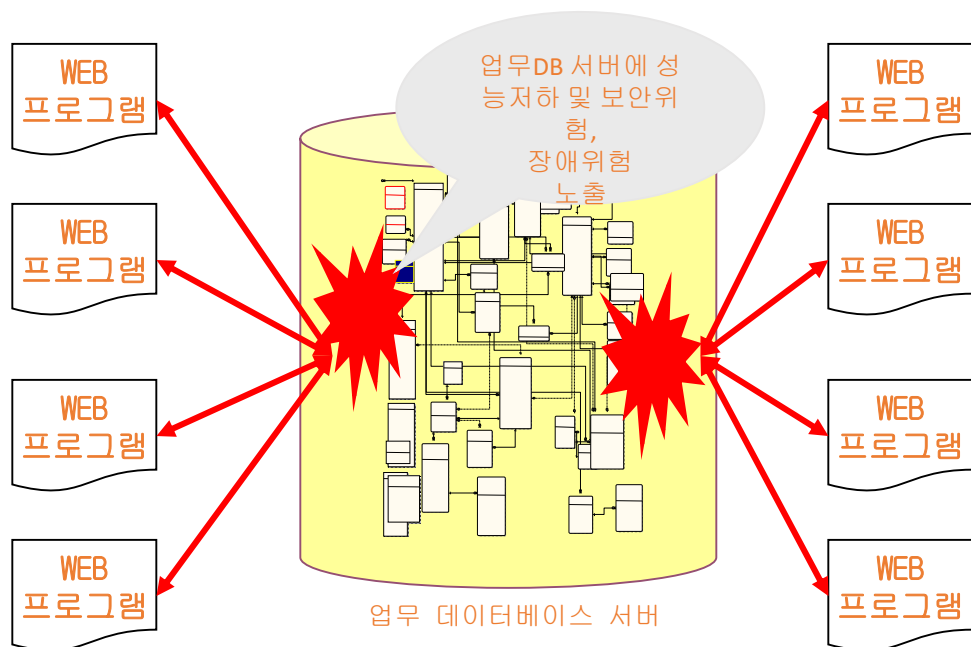
```
0  SELECT STATEMENT Optimizer=CHOOSE  
1  0  TABLE ACCESS (BY INDEX ROWID) OF '사용자'  
2  1  INDEX (UNIQUE SCAN) OF 'XPK사용자' (UNIQUE)
```

6. 분산모델 구현

분산모델 구현

- 업무 데이터베이스에 불특정 다수 트랜잭션 발생

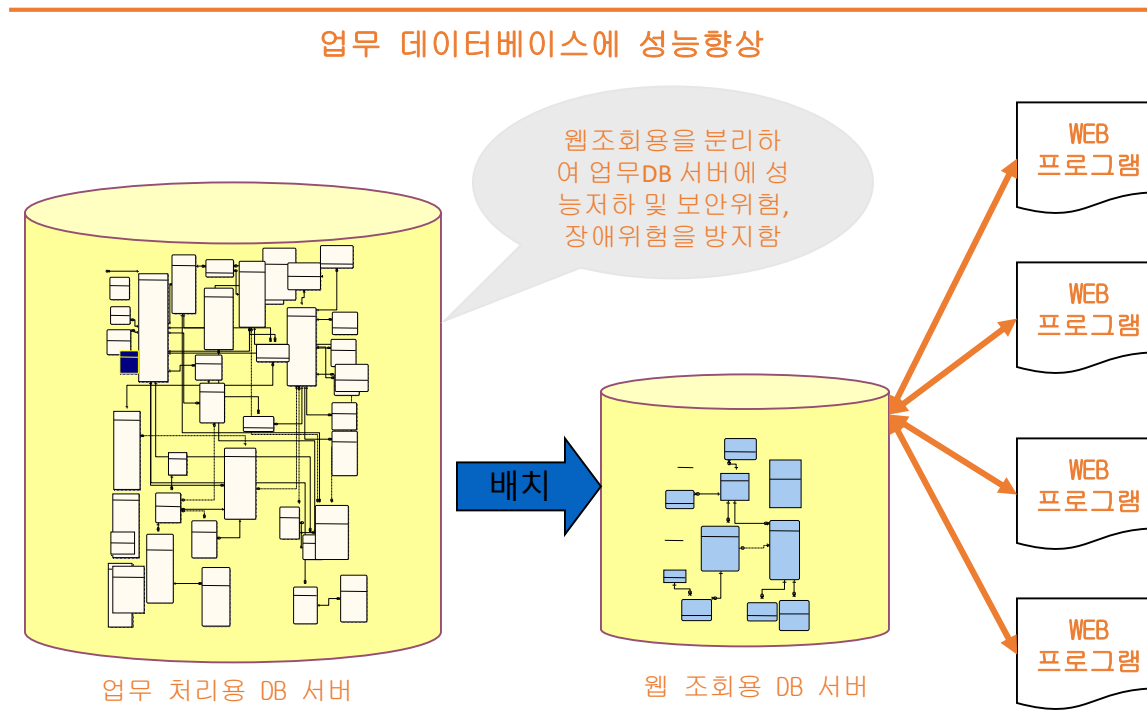
업무 데이터베이스에 불특정 다수 트랜잭션 발생



6. 분산모델 구현

분산모델 구현

- 업무 데이터베이스에 성능향상

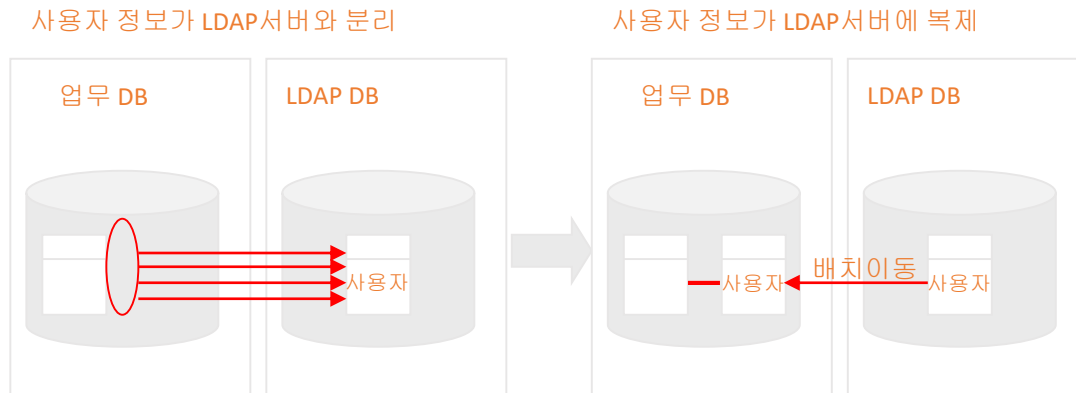


6. 분산모델 구현

분산모델 구현

- LDAP에서 사용자 정보 복제

LDAP에서 사용자 정보 복제



대량 데이터처리시 반복함수 호출로 성능저하

대량 데이터처리시 조인실행으로 성능향상



7. 도메인을 적용한 성능향상

도메인을 적용한 성능향상

- 데이터타입과 길이 불일치로 인한 성능저하

데이터타입과 길이 불일치로 인한 성능저하

장비

장비번호: VARCHAR2(8)
장비명: VARCHAR2(20)
장비도입일자: DATE
폐기여부: CHAR(1)

생산정보

라인번호: CHAR(18)
생산일련번호: NUMBER(6)
생산일자: CHAR(18)
장비번호: CHAR(10) (FK)

동일한 컬럼명이지만
데이터타입과 길이가
달라졌음



7. 도메인을 적용한 성능향상

도메인을 적용한 성능향상

- 데이터타입과 길이 불일치로 인한 성능저하

데이터타입과 길이 불일치로 인한 성능저하

```
SELECT B.라인번호, COUNT(B.생산일련번호)
FROM 장비 A, 생산정보 B
WHERE A.장비도입일자 = '20041201'
AND A.장비번호 = RTRIM(B.장비번호)
GROUP BY B.라인번호;
```

Execution Plan

```
0  SELECT STATEMENT Optimizer=RULE
1  0  SORT (GROUP BY)
2  1  NESTED LOOPS
3  2  TABLE ACCESS (FULL) OF '생산정보'
4  2  TABLE ACCESS (BY INDEX ROWID) OF '장비'
5  4  INDEX (UNIQUE SCAN) OF 'XPK장비' (UNIQUE)
```

인덱스 컬럼 변형으로
FULL TABLE SCAN이 발생
됨

Unit A

참고자료



문헌

1. <http://www.ncs.go.kr>
2. NELLDAL/JOHN LEWIS지음, 조영석/김대경/박찬영/송창근 역, 단계별로 배우는 컴퓨터과학, 홍릉과학출판사, 2018
3. 기타 서적 및 웹 사이트 자료 다수 참조

감사합니다.

❖ Mobile: 010-9591-1401

❖ E-mail: onlooker2zip@naver.com / dkkim@lklab.org