

[기업수요맞춤형]

2일차 : 오전

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

Copyright 2022. Daekyeong all rights reserved



- 1 1교시 : 데이터베이스 구축 준비
- 2 2교시 : 데이터베이스 구축 준비
- 3 3교시 : 데이터베이스 구축 준비
- 4 4교시 : 데이터베이스 구축 준비

『2과목』 Practice

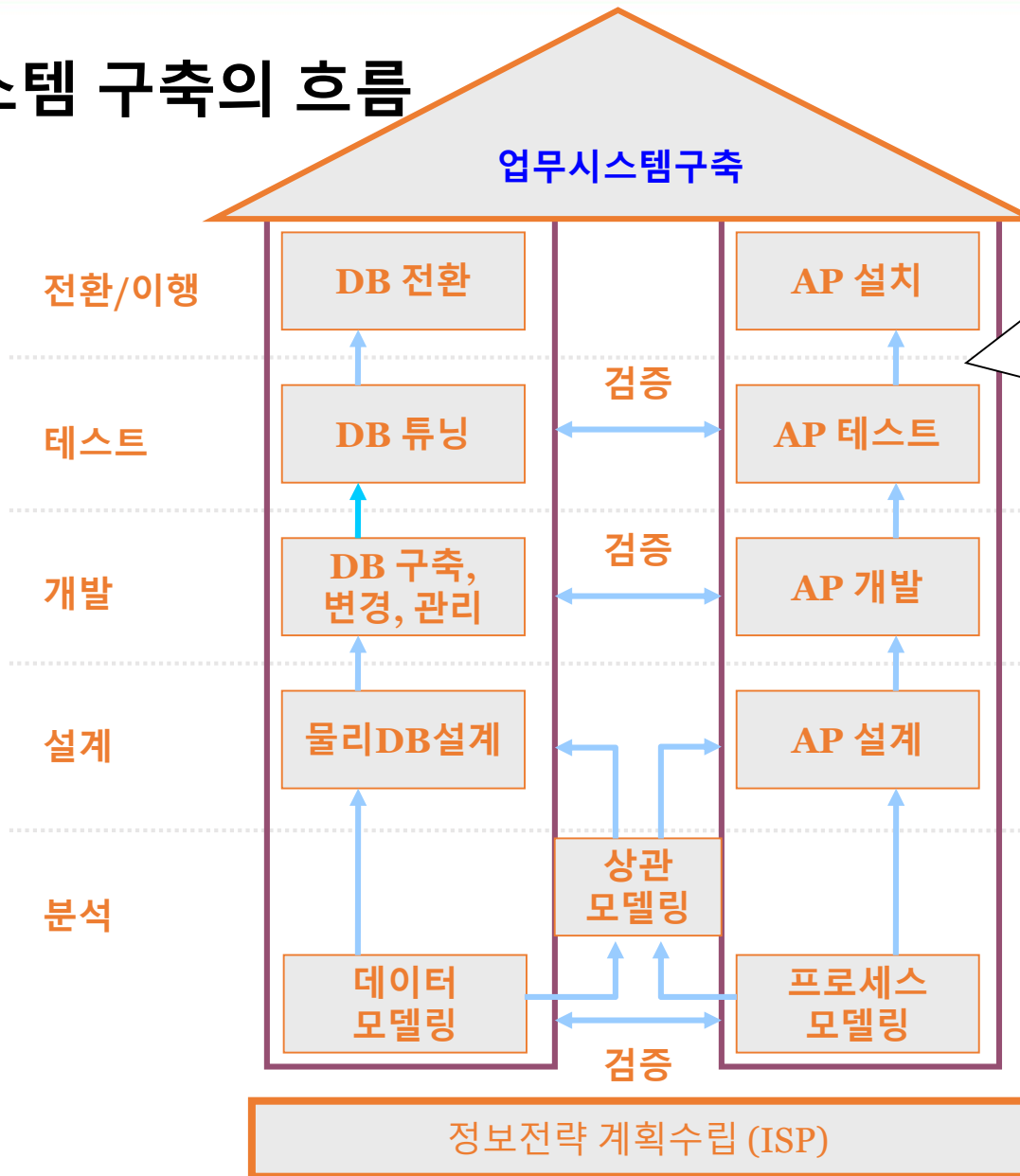
1-4교시 :

데이터베이스 구축 준비



1. 물리설계

업무시스템 구축의 흐름



데이터를 기반으로 세워나가는 기둥과 프로세스를 기반으로 세워나가는 두 개의 기둥이 있다.

1. 물리설계

물리설계의 세가지

1. 논리 모델인 ERD를 물리 모델링 ERD로(테이블 관계도)로 전환.
 2. 테이블 관계 이외의 데이터베이스와 관련된 사항, 즉 오브젝트, 접근방법, 트랜잭션 분석, 저장방법에 대해 설계한다.
 3. 분산 데이터베이스를 설계한다.
- ❖ 분산 데이터베이스의 경우 물리적인 데이터모델을 다 완성한 이후에 고려하는 경우도 있지만 논리적인 설계시점부터 고려하여 설계에 반영할 수 있다.



2. 관계형 테이블 전환

관계형 테이블 전환 개요

1. ERD 전환 > 관계형 테이블 전환





2. 관계형 테이블 전환

테이블관계도 변환 항목

1. 엔티티타입은 테이블로 전환한다.
2. 주식별자는 PRIMARY KEY로 변환한다.
3. 속성은 컬럼으로 변환한다.
4. 관계에 의한 외부식별자는 FOREIGN KEY로 변환한다.

엔티티타입전환 규칙

- 엔티티타입은 테이블로 전환한다.
- 독립엔티티타입은 독립테이블로 전환된다.
- 종속엔티티타입은 종속테이블로 전환된다.
- 슈퍼타입과 서브타입은 여러 형태의 테이블로 전환될 수 있다.

2. 관계형 테이블 전환

엔티티타입전환

- 엔티티타입은 테이블로 전환한다.
- 독립엔티티타입은 독립 테이블로...

사원

사원번호	이름	주소	전화번호
AF11002	홍길동	경북 포항시	1101)1234-5678
AF11017	이순신	전북 고창군	1111)0234-5678
AF11021	황진희	서울 은평구	1121)3234-5678

사원번호
이름
주소
전화번호

PRIMARY KEY

2. 관계형 테이블 전환

엔티티타입전환

- 엔티티타입은 테이블로 전환한다.
- 완전종속 엔티티타입은 완전종속 테이블로...

사원

사원번호
이름
주소
전화번호
전자메일



발령

사원번호 (FK)
발령시작일자
발령사유
발령종료일자

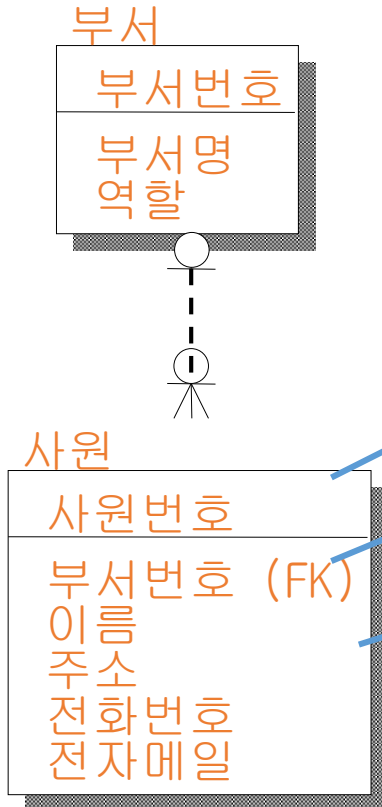
<u>사원번호(FK)</u>	<u>발령시작일자</u>	발령사유	발령종료일자
AF11002	2023.7.1	전배	2023.9.6
AF11002	2023.9.7	조직개편	2004.2.28
AF11002	2004.3.1	조직개편	
AF1017	2023.7.1	전배	



2. 관계형 테이블 전환

엔티타입전환

- 엔티타입은 테이블로 전환한다.
- 부분종속 엔티타입은 부분종속 테이블로...



<u>사원번호</u>	부서번호 (FK)	이름	주소	전화번호
AF11002	RD01	홍길동	경북 포항시	1101)1234-5678
AF11017		이순신	전북 고창군	1111)0234-5678
AF11021	T001	황진희	서울 은평구	1121)3234-5678



2. 관계형 테이블 전환

주식별자 전환

- 주식별자는 **PRIMARY KEY**로 변환한다.
- 1. 데이터모델의 주식별자는 **PRIMARY KEY**로 전환된다.
- 2. 참조 무결성제약을 유지시키는 역할을 한다.
- 3. 데이터를 유일하게 식별할 수 있게 한다.
- 4. **PRIMARY KEY**는 **NULL** 값을 갖지 않는다.
- 5. **PRIMARY KEY**는 변경되지 않으며 업무적으로 변경이 발생한다는 것은 테이블에서 삭제와 입력이 동시에 발생하는 것을 의미한다.
- 6. 가능하면 모든 테이블에 **PRIMARY KEY**를 정의한다.



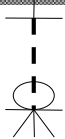
2. 관계형 테이블 전환

주식별자 전환

- 주식별자는 **PRIMARY KEY**로 변환한다.
- 데이터모델의 주식별자는 **PRIMARY KEY**로 전환된다.
- 주식별자로 선정된 속성은 모두 테이블에서 **PRIMARY KEY**가 된다. 참조 무결성 제약을 유지시키는 역할을 한다.

부서

부서번호
부서명
역할



사원

사원번호
부서번호 (FK)
이름
주소
전화번호
전자메일

부서번호	부서명	역할
RD01	기술연구부	R&D 업무
BB01	인사부	인사, 급여, 보상업무
T001	자재부	자재수급 업무

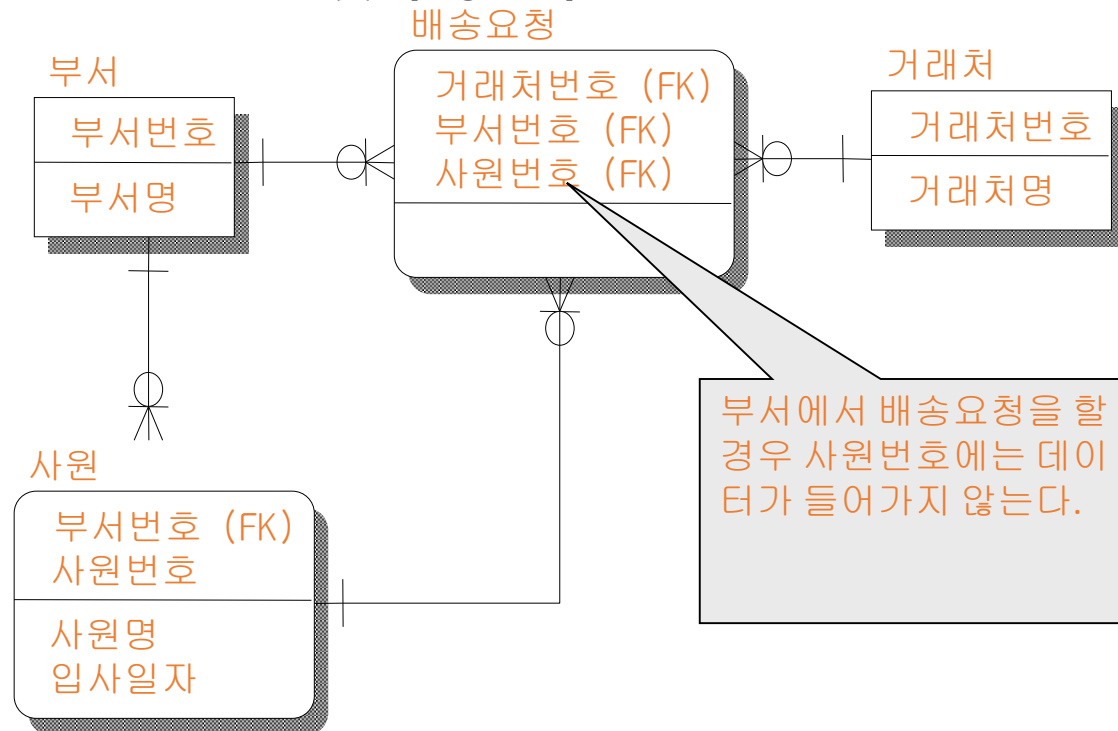
사원번호	부서번호 (FK)	이름	주소	전화번호
AF11002	RD01	홍길동	경북 포항시	1101)1234-5678
AF11017		이순신	전북 고창군	1111)0234-5678
AF11021	T001	황진희	서울 은평구	1121)3234-5678



2. 관계형 테이블 전환

주식별자 전환

- 데이터를 유일하게 식별할 수 있게 한다.
- 일반적으로는 테이블에 PRIMARY KEY를 지정하여 각각의 ROW를 유일하게 식별할 수 있도록 한다. 그러나 데이터베이스 운영하는데 있어 테이블간 참조무결성 제약 조건을 걸지 않는다면 그냥 PRIMARY KEY를 지정하지 않고 UNIQUE INDEX만 지정하여 테이블을 생성할 수도 있다.
- PRIMARY KEY는 NULL값을 갖지 않는다.





2. 관계형 테이블 전환

주식별자 전환

- PRIMARY KEY는 변경되지 않으며 업무적으로 변경이 발생한다는 것은 테이블에서 삭제와 입력이 동시에 발생하는 것을 의미한다.
- PRIMARY KEY의 변경은 UPDATE명령에 의해 변경에 된다고 하더라도 DELETE/INSERT가 동시에 되었다고 생각하도록 한다.
- **가능하면 모든 테이블에 PRIMARY KEY를 정의한다.**
- 테이블에 따라서는 PRIMARY KEY를 사용하지 않고 일반컬럼으로 이용하는 경우도 있고 단지 UNIQUE INDEX만 생성하여 사용하는 경우도 있다. 그러나 PRIMARY KEY를 정의하지 않으면 FOREIGN KEY 또한 생성할 수 없으므로 데이터베이스의 참조무결성제약을 걸지 못하는 경우가 발생한다. 가능하면 모든 테이블에 PRIMARY KEY를 생성하여 데이터의 참조무결성을 보장하도록 한다.

2. 관계형 테이블 전환

속성 전환

- 속성은 컬럼으로 변환한다.

사원

사원번호
부서번호 (FK)
이름
주소
전화번호
전자메일

<u>사원번호</u>	부서번호 (FK)	이름	주소	전화번호
AF11002	RD01	홍길동	경북 포항시	1101) 1234-5678
AF11017		이순신	전북 고창군	1111) 0234-5678
AF11021	T001	황진희	서울 은평구	1121) 3234-5678



2. 관계형 테이블 전환

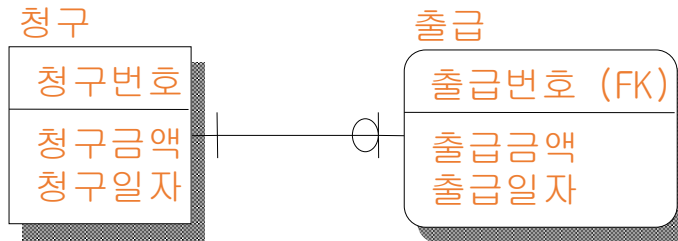
관계전환

- 관계에 의한 외부식별자는 FOREIGN KEY로 변환한다.
- 1. 1:1 주식별자관계 변환(IDENTIFYING RELATIONSHIP)
- 2. 1:1 비식별자관계(NON-IDENTIFYING RELATIONSHIP) 변환
- 3. 1:M 관계 변환
- 4. 자기참조관계(RECURSIVE RELATIONSHIP) 변환
- 5. 슈퍼타입/서브타입 관계 변환

2. 관계형 테이블 전환

관계전환

- 관계에 의한 외부식별자는 FOREIGN KEY로 변환한다.
- 1:1 주식별자관계 변환(IDENTIFYING RELATIONSHIP)



테이블명 : 청구

청구번호	청구금액	청구일자
1001	1000,000	2023-07-01

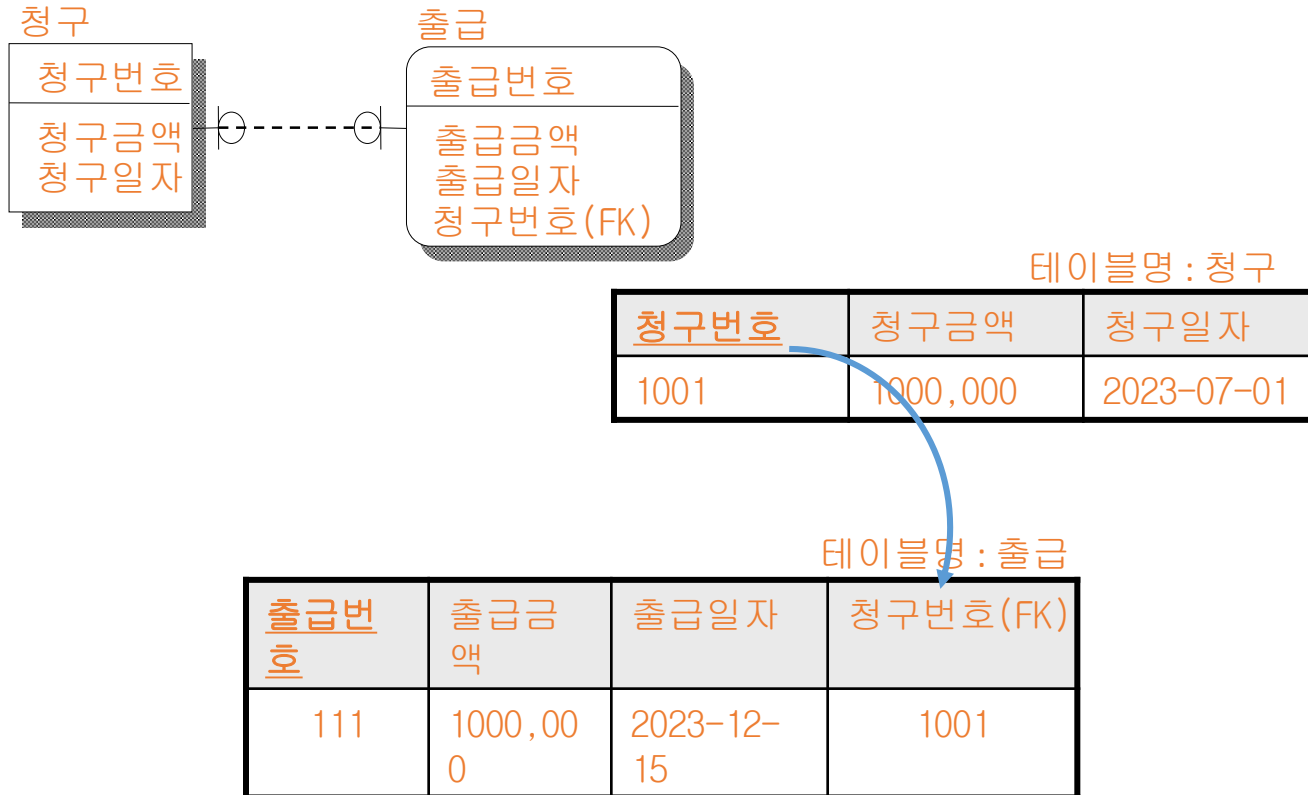
테이블명 : 출급

출급번호(FK)	출급금액	출급일자
1001	1000,000	2023-12-15

2. 관계형 테이블 전환

관계전환

- 관계에 의한 외부식별자는 FOREIGN KEY로 변환한다.
- 1:1 비식별자관계 변환(NON-IDENTIFYING RELATIONSHIP)

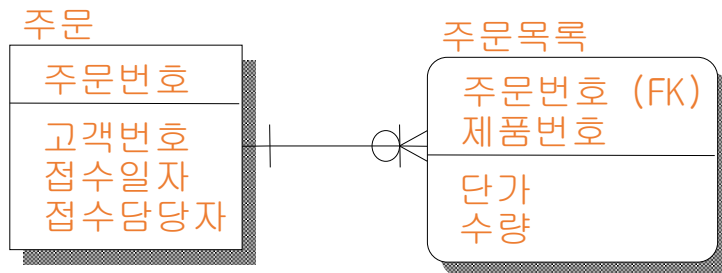




2. 관계형 테이블 전환

관계전환

- 관계에 의한 외부식별자는 **FOREIGN KEY**로 변환한다.
- 1:M 관계 변환



테이블명 : 주문

주문번호	고객번호	접수일자	접수담당자
1001	AB12345	2023-07-01	홍길동

테이블명 : 주문목록

주문번호(FK)	제품번호	단가	수량
1001	ER-001	23,000원	1,000

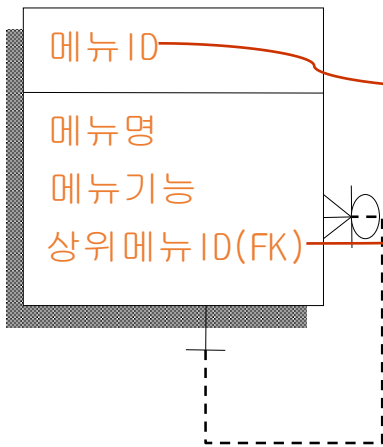


2. 관계형 테이블 전환

관계전환

- 관계에 의한 외부식별자는 FOREIGN KEY로 변환한다.
- 자기참조관계(RECURSIVE RELATIONSHIP) 변환

메뉴



테이블명 : 메뉴

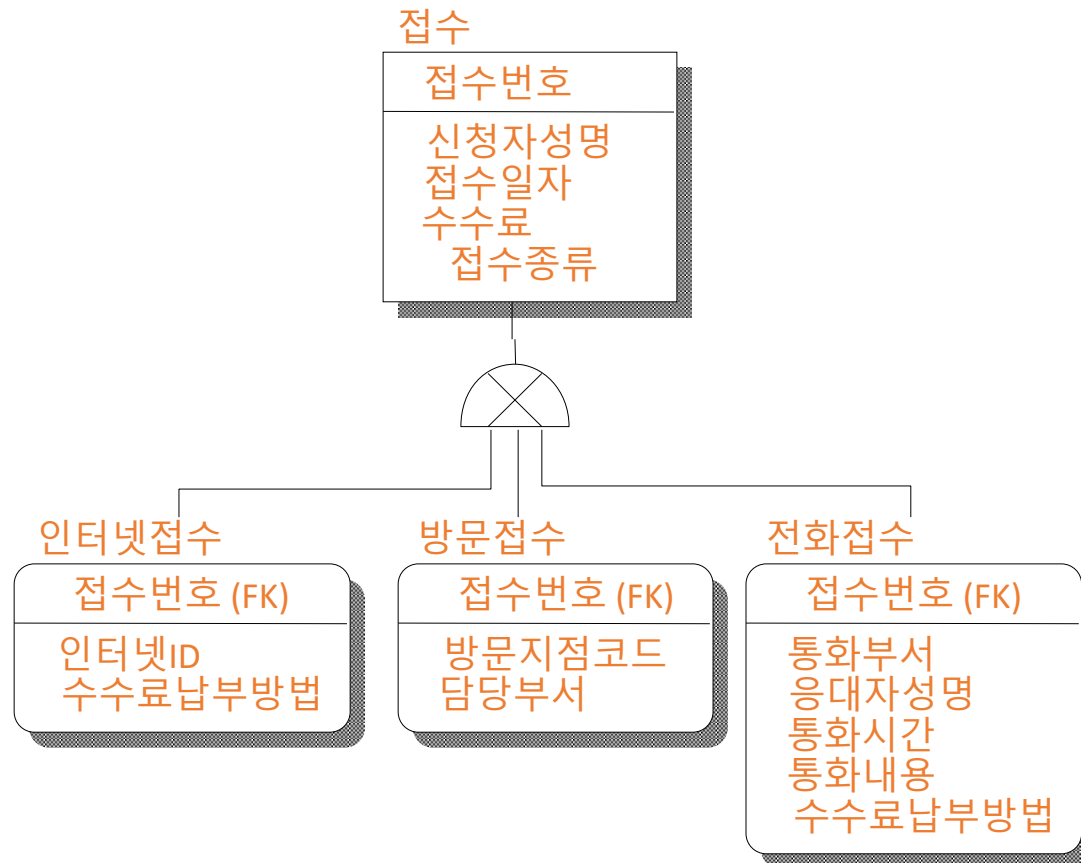
<u>메뉴 ID</u>	메뉴명	메뉴기능	상위메뉴 ID(FK)
A101	회원관리	회원관리주메뉴	
A102	회원가입	회원가입신청	A101
A103	회원정보수정	회원정보수정	A101
A104	탈퇴신청	회원탈퇴신청	A101



2. 관계형 테이블 전환

관계전환

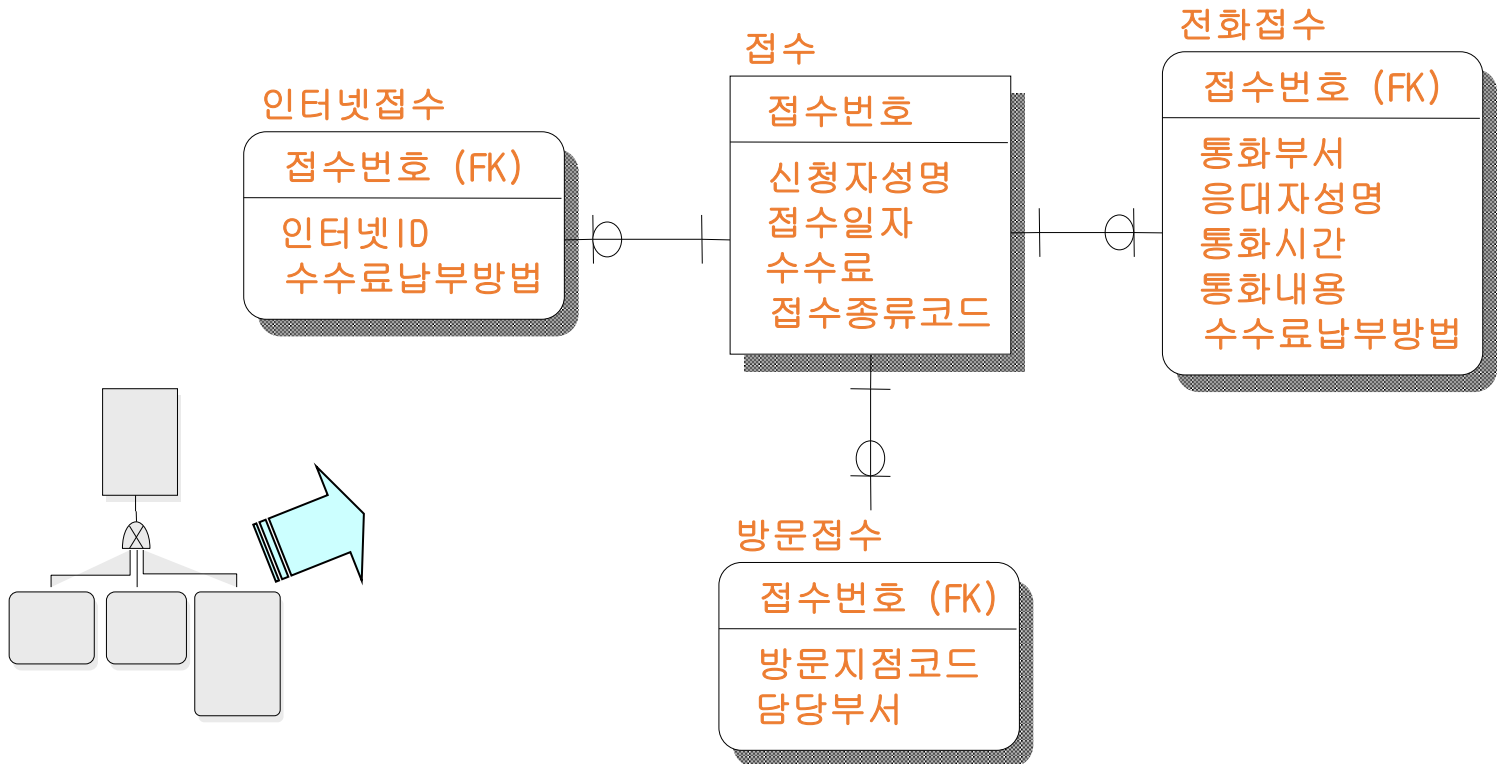
- 관계에 의한 외부식별자는 **FOREIGN KEY**로 변환한다.
- 슈퍼타입/서브타입 관계 변환



2. 관계형 테이블 전환

관계전환

- 관계에 의한 외부식별자는 FOREIGN KEY로 변환한다.
- 슈퍼타입/서브타입 관계 – 개별 테이블로 변환(1:1관계)



2. 관계형 테이블 전환

관계전환

- 관계에 의한 외부식별자는 FOREIGN KEY로 변환한다.
- 슈퍼타입/서브타입 관계 – 개별 테이블로 변환(1:1관계) - 테이블

접수번호	신청자성명	접수일자	수수료	접수종류
A101	황진희	2023-07-01	2,000원	인터넷
A102	이순신	2023-08-15	3,000원	방문
A103	홍길동	2023-12-15	2,000원	인터넷
A104	임꺽정	2023-12-20	2,500원	전화
A105	박문수	2003-01-01	2,500원	전화
A106	김 구	2003-03-02	3,000원	방문

테이블명 :인터넷접수

접수번호(FK)	인터넷ID	수수료납부방법
A101	DcLee	신용카드
A103	HCLEE	계좌이체

테이블명 :방문접수

접수번호(FK)	방문지점코드	담당부서
A102	100210	영업3부
A106	121000	영업1부

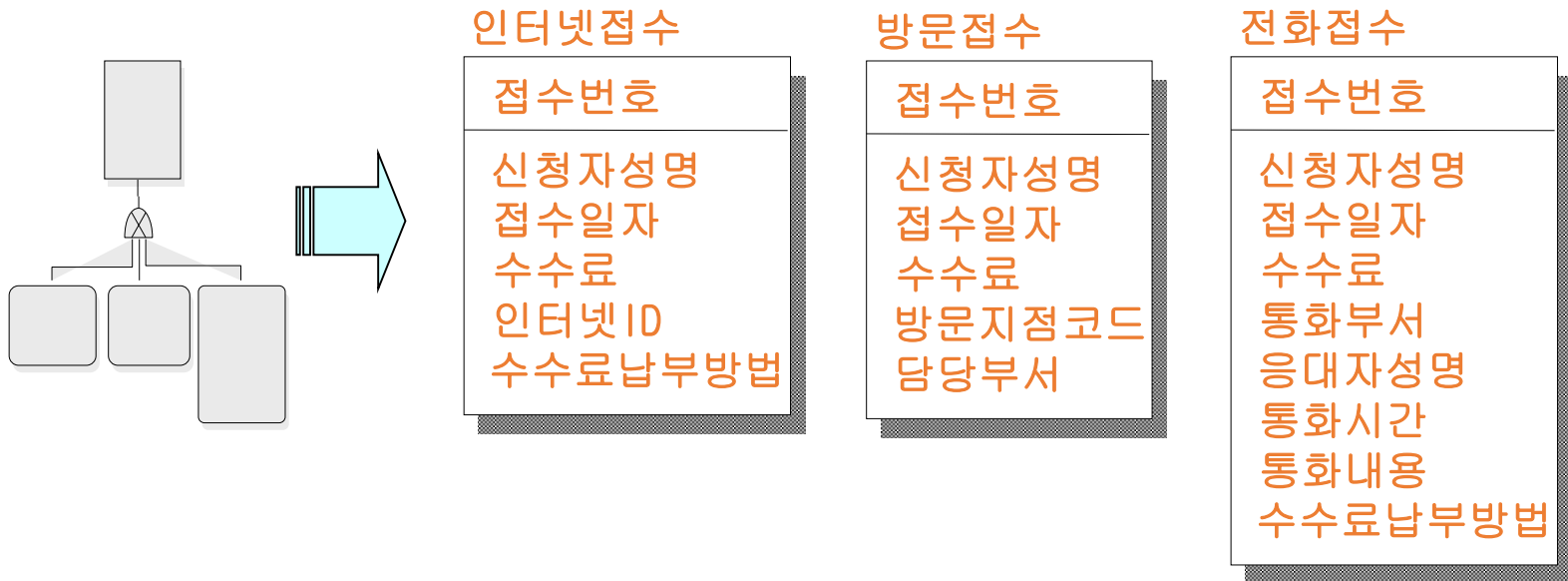
테이블명 :전화접수

접수번호(FK)	통화부서	응대자성명	통화시간	통화내용	수수료납부방법
A104	영업1부	홍길동	2분30초	접수신청	신용카드
A105	영업2부	한용운	1분12초	접수내용수정	신용카드

2. 관계형 테이블 전환

관계전환

- 관계에 의한 외부식별자는 FOREIGN KEY로 변환한다.
- 슈퍼타입/서브타입 관계 – 각 서브타입 테이블로 변환



2. 관계형 테이블 전환

관계전환

- 관계에 의한 외부식별자는 FOREIGN KEY로 변환한다.
- 슈퍼타입/서브타입 관계 – 각 서브타입 테이블로 변환 - 테이블

테이블명 : 인터넷접수

접수번호	신청자성명	접수일자	수수료	인터넷 ID	수수료납부방법
A101	황진희	2023-07-01	2,000원	DcLee	신용카드
A103	홍길동	2023-12-15	2,000원	HCLEE	계좌이체

테이블명 : 방문접수

접수번호	신청자성명	접수일자	수수료	방문지점코드	담당부서
A102	이순신	2023-08-15	3,000원	100210	영업3부
A106	김 구	2003-03-02	3,000원	121000	영업1부

테이블명 : 전화접수

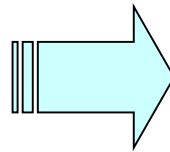
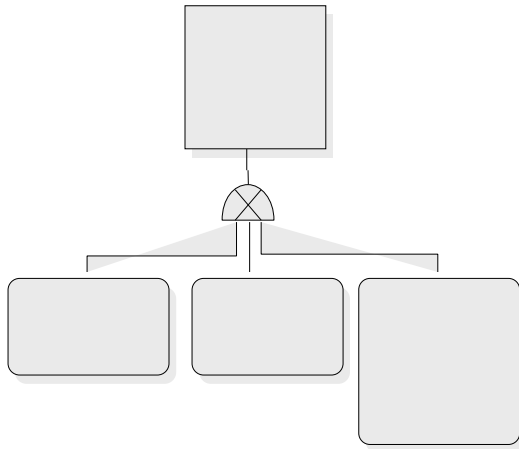
접수번호	신청자 성명	접수일자	수수료	통화 부서	응대자 성명	통화 시간	통화 내용	수수료 납부방법
A104	임꺽정	2023-12-20	2,500원	영업 1부	홍길동	2분30초	접수 신청	신용카드
A105	박문수	2003-01-01	2,500원	영업 2부	한용운	1분12초	접수 수정	신용카드



2. 관계형 테이블 전환

관계전환

- 관계에 의한 외부식별자는 FOREIGN KEY로 변환한다.
- 슈퍼타입/서브타입 관계 – 통합 테이블로 변환



접수

접수번호

신청자성명
접수일자
수수료
인터넷ID
방문지점코드
담당부서
통화부서
응대자성명
통화시간
통화내용
접수종류

2. 관계형 테이블 전환

관계전환

- 관계에 의한 외부식별자는 FOREIGN KEY로 변환한다.
- 슈퍼타입/서브타입 관계 – 통합 테이블로 변환 - 테이블

접수번호	신청자 성명	접수일자	수수료	인터넷 ID	수수료 납부방법	방문지점 코드
A101	황진희	2023-07-01	2,000원	DcLee		
A102	이순신	2023-08-15	3,000원		신용카드	100210
A103	홍길동	2023-12-15	2,000원	HCLEE		
A104	임객정	2023-12-20	2,500원			
A105	박문수	2003-01-01	2,500원			
A106	김 구	2003-03-02	3,000원		계좌이체	121000

테이블명 : 접수

담당부서	통화부서	응대자성명	통화시간	통화내용	접수종류
					인터넷
영업3부					방문
					인터넷
	영업1부	홍길동	2분30초	접수신청	전화
	영업2부	한용운	1분12초	접수수정	전화
영업1부					방문

3. 반정규화

반정규화?

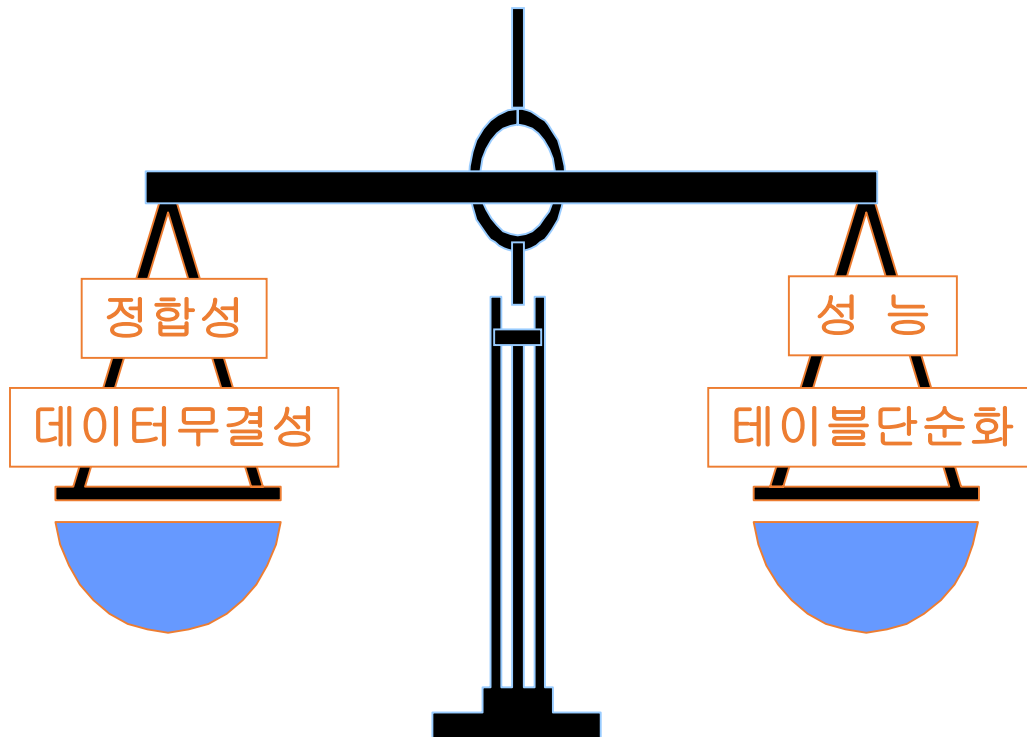
- 반정규화란 정규화된 엔티티타입, 속성, 관계에 대해 시스템의 성능 향상과 개발(DEVELOPMENT)과 운영(MAINTENANCE)의 단순화를 위해 데이터 모델을 통합하는 프로세스를 의미한다.



3. 반정규화

반정규화의 기준

- 반정규화를 하기 위해 가장 중요하게 검토해야 할 기준은 각각의 엔티티타입과 속성, 관계에 대해 데이터의 정합성, 데이터의 무결성을 가장 우선으로 가져가야 할지 아니면 데이터베이스구성의 단순화와 성능을 우선으로 가져가야 할지를 고려하는 것이다.



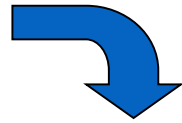
3. 반정규화

반정규화의 절차

● 반정규화의 절차

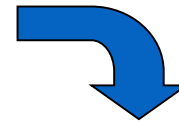
1. 반정규화 대상조사

- 범위처리빈도수 조사
- 대량의 범위 처리 조사
- 통계성 프로세스 조사
- 테이블 조인 개수



2. 다른 방법유도 검토

- 뷰(VIEW) 테이블
- 클러스터링 적용
- 인덱스의 조정
- 응용어플리케이션



3. 반정규화 적용

- 테이블 반정규화
- 속성의 반정규화
- 관계의 반정규화

테이블의 반정규화

● 테이블의 반정규화

1. 테이블 병합

- 1:1관계의 테이블병합
- 1:M관계의 테이블병합
- 수퍼타입 서브타입
테이블 병합

2. 테이블 분할

- 수직분할
- 수평분할

3. 테이블 추가

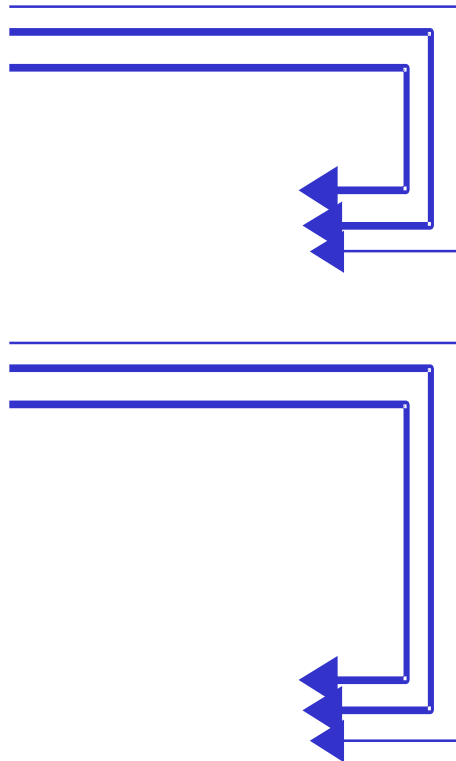
- 중복테이블 추가
- 통계테이블 추가
- 이력테이블 추가
- 부분테이블 추가

3. 반정규화

테이블의 반정규화

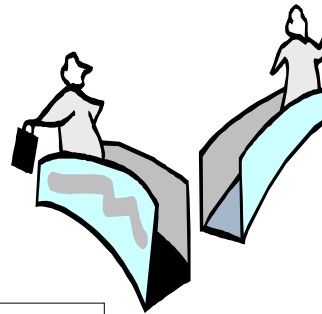
테이블의 반정규화-수직분할

PROCESS 가 특정
컬럼에 다르게 발생



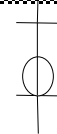
수납

수납번호
수납확인일자
수납확인담당자사번
수납확인반영일자
수납확인방법
수납필일자
납부방법
지점코드
청구번호
고객번호
회계처리일자
접수일자
정보작성점코드
수납필정정구분



수납

수납번호
수납필일자
납부방법
지점코드
청구번호
고객번호
회계처리일자
접수일자
정보작성점코드
수납필정정구분



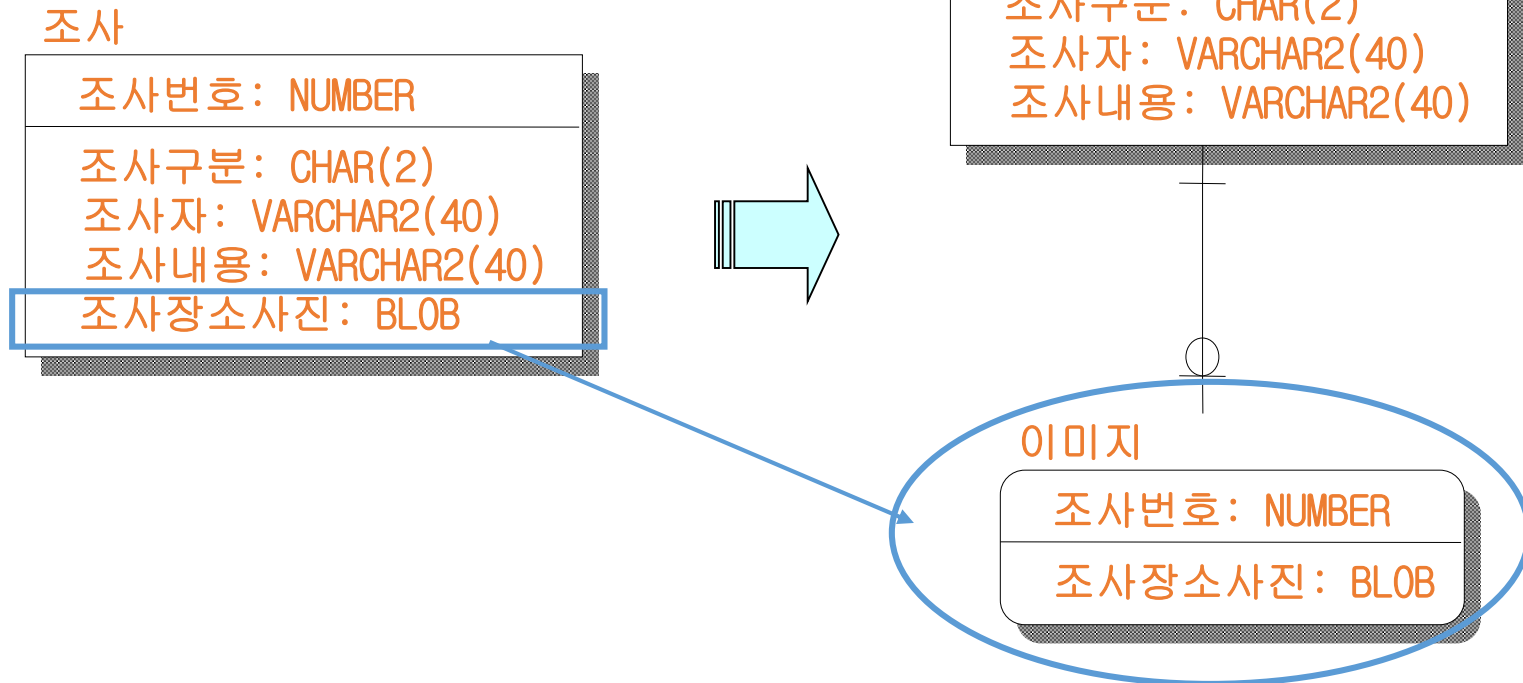
수납확인

수납번호 (FK)
수납확인일자
수납확인담당자사번
수납확인반영일자
수납확인방법

3. 반정규화

테이블의 반정규화

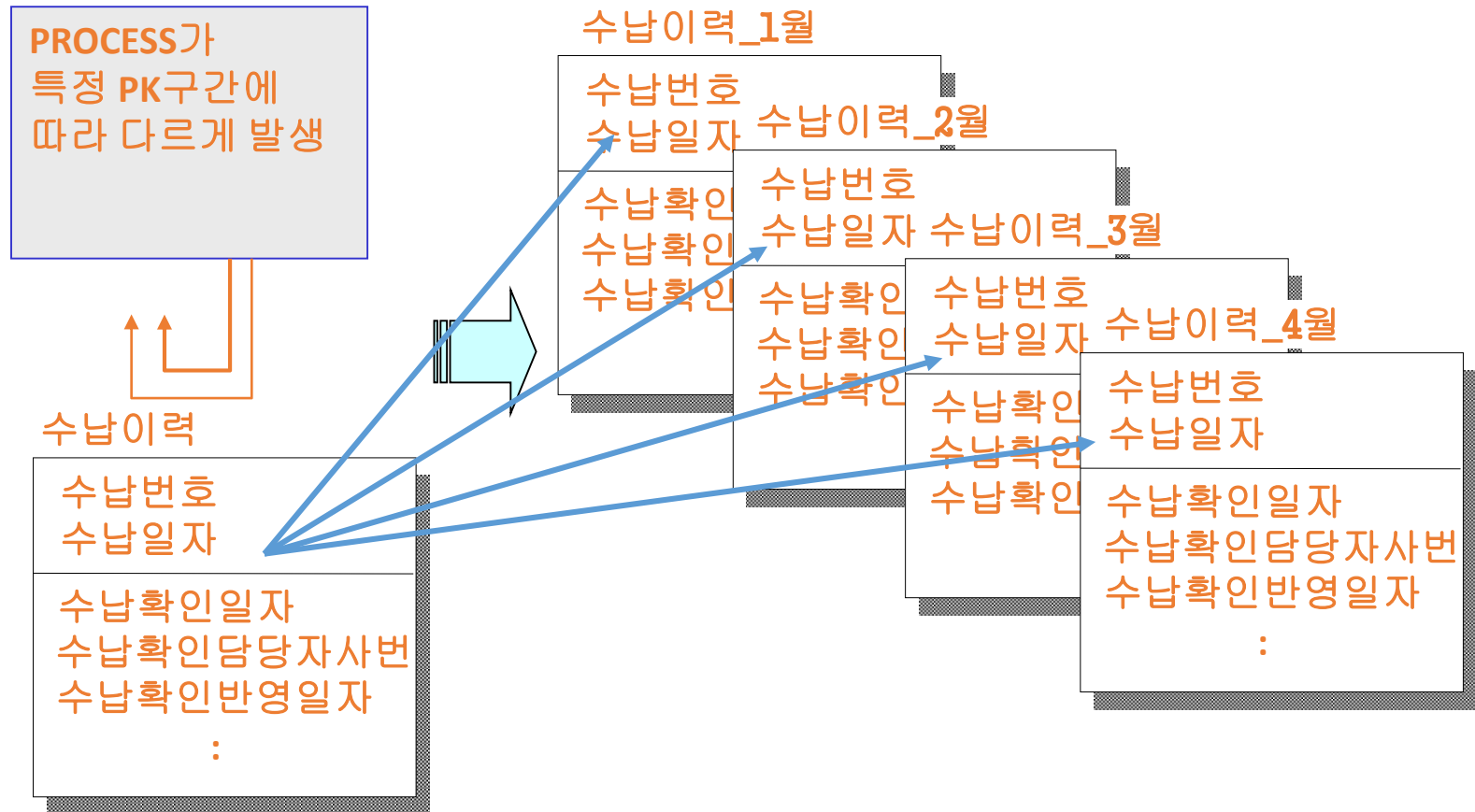
- 테이블의 반정규화-이미지
- CREATE TABLE 테이블_TEMP AS SELECT * FROM 테이블 -> LONG 데이터 유형은 사용할 수 없다.
- 이미지 타입을 위한 테이블 분리



3. 반정규화

테이블의 반정규화

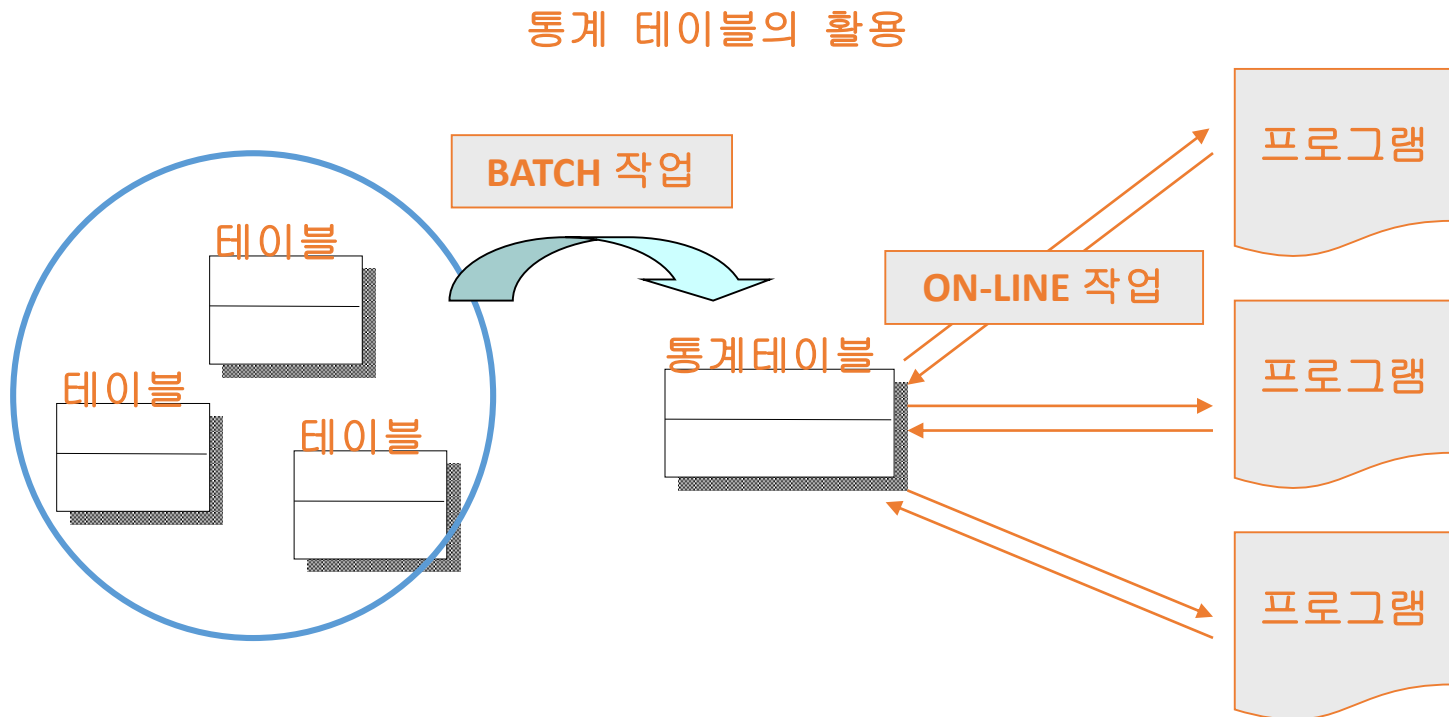
- 테이블의 반정규화-수평분할
- 프로세스가 구간별로 발생하는 엔티티타입 분리



3. 반정규화

테이블의 반정규화

- 테이블의 반정규화-통계



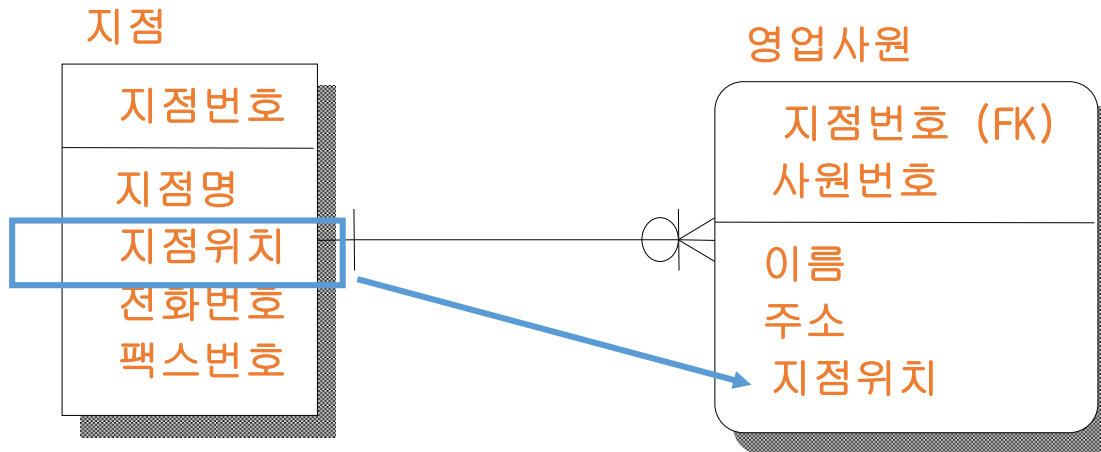
컬럼의 반정규화

- 중복 컬럼 추가
- 파생 컬럼 추가(컬럼(COLUMN), 로우(ROW)에 의한 파생 컬럼)
- 이력테이블 컬럼 추가
- PRIMARY KEY에 의한 컬럼 추가
- 응용시스템 오 작동 처리를 위한 컬럼 추가

3. 반정규화

컬럼의 반정규화

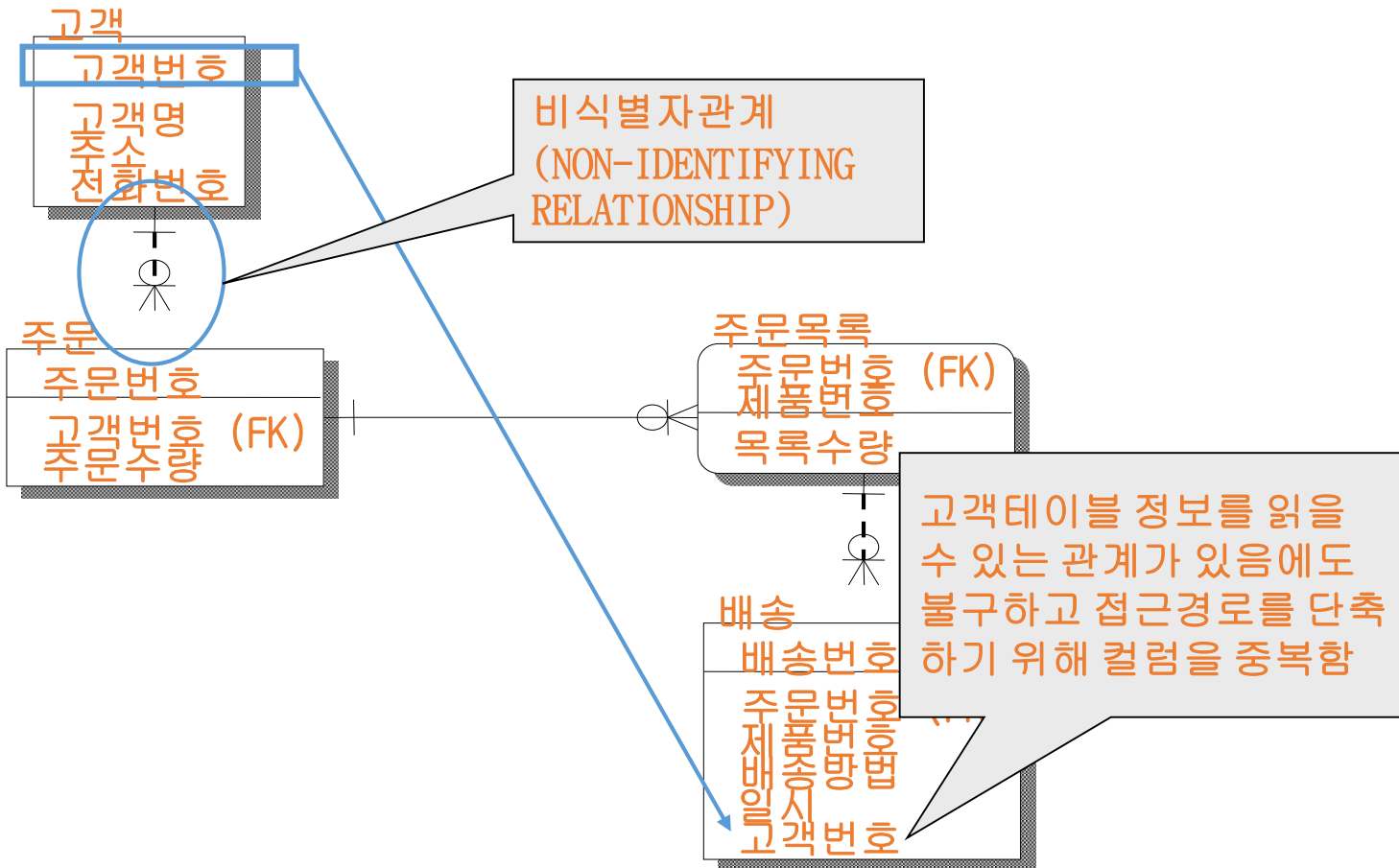
- 컬럼의 반정규화-자주사용하는 컬럼
- 컬럼 중복 – 자주 사용하는 컬럼 중복



3. 반정규화

컬럼의 반정규화

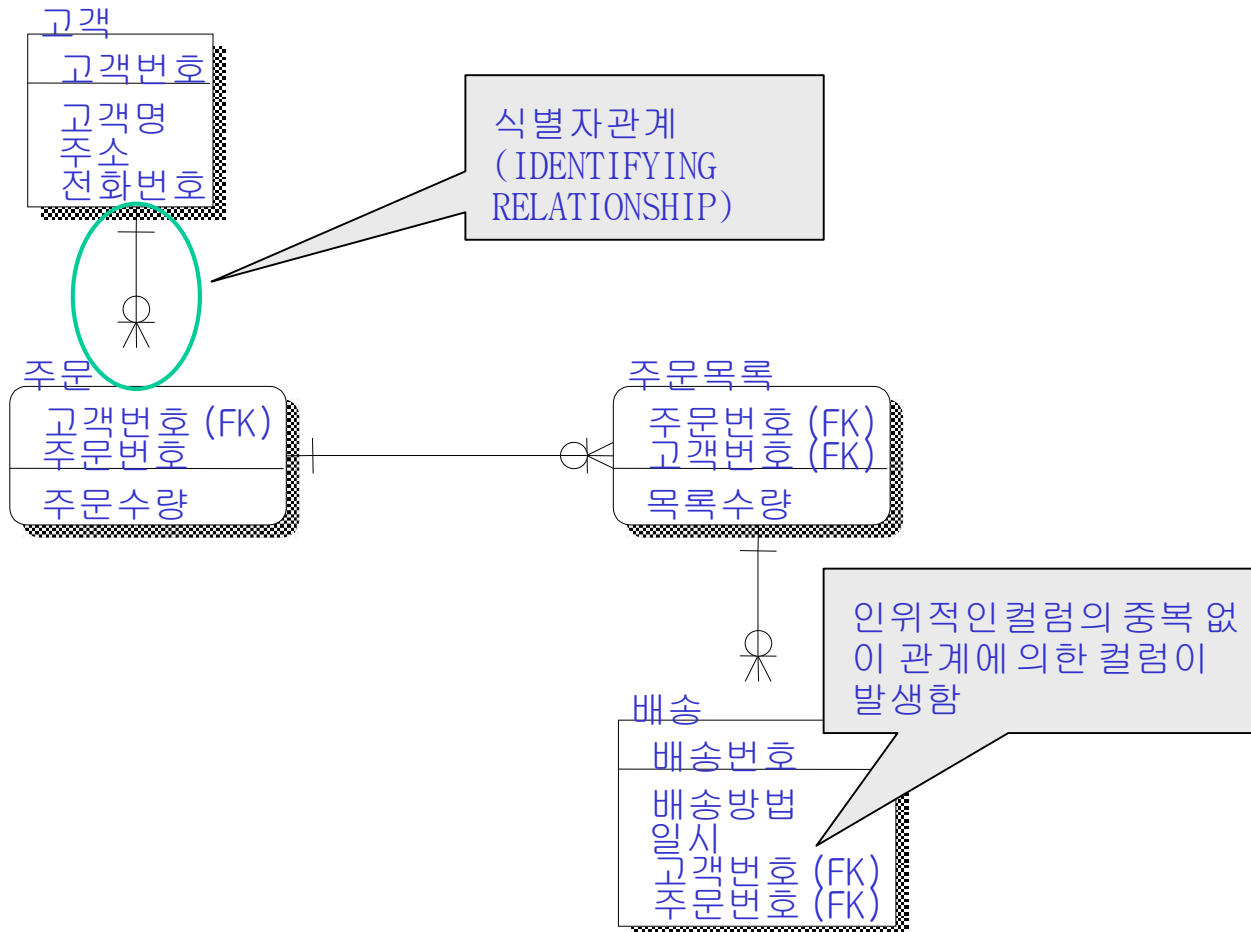
- 컬럼의 반정규화-경로단축
- 컬럼 중복 – 조회경로단축



3. 반정규화

컬럼의 반정규화

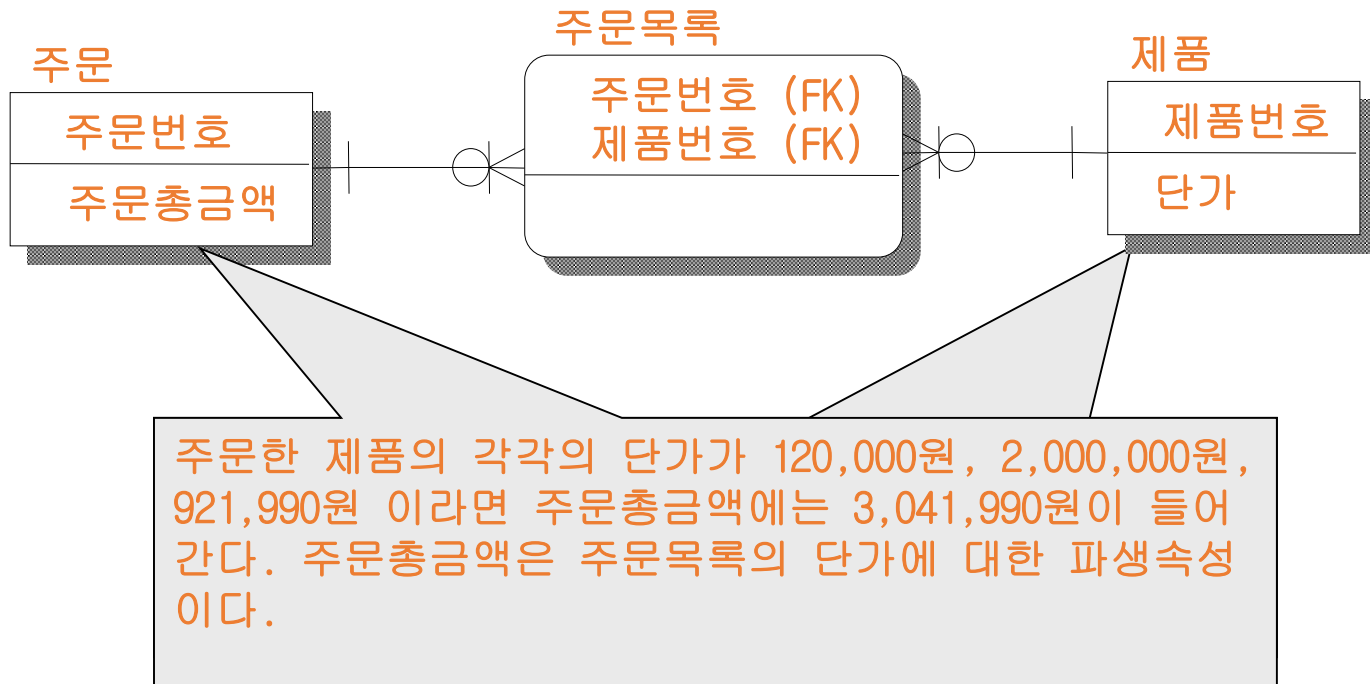
- 컬럼의 반정규화-경로단축
- 컬럼 중복 – 인위적인 컬럼 중복 없이 조희경로단축



3. 반정규화

컬럼의 반정규화

- 컬럼의 반정규화-파생컬럼
- 컬럼에 의한 파생컬럼(DERIVED COLUMN) 추가



3. 반정규화

컬럼의 반정규화

- SQL

```
SELECT A.주문번호, SUM(C.단가)
FROM 주문 A, 주문목록 B, 제품 C
WHERE A.주문번호 = '2023-01-001'
AND A.주문번호 = B.주문번호
AND B.제품번호 = C.제품번호
GROUP BY A.주문번호
```



```
SELECT 주문번호, 주문총금액
FROM 주문
WHERE A.주문번호 = '2023-01-001'
```

3. 반정규화

컬럼의 반정규화

- 컬럼의 반정규화-파생컬럼
- 로우(ROW)에 의한 파생컬럼(DERIVED COLUMN) 추가
- 이러한 속성의 반정규화가 될 수는 있지만 사용하지 않도록 함.

공사비

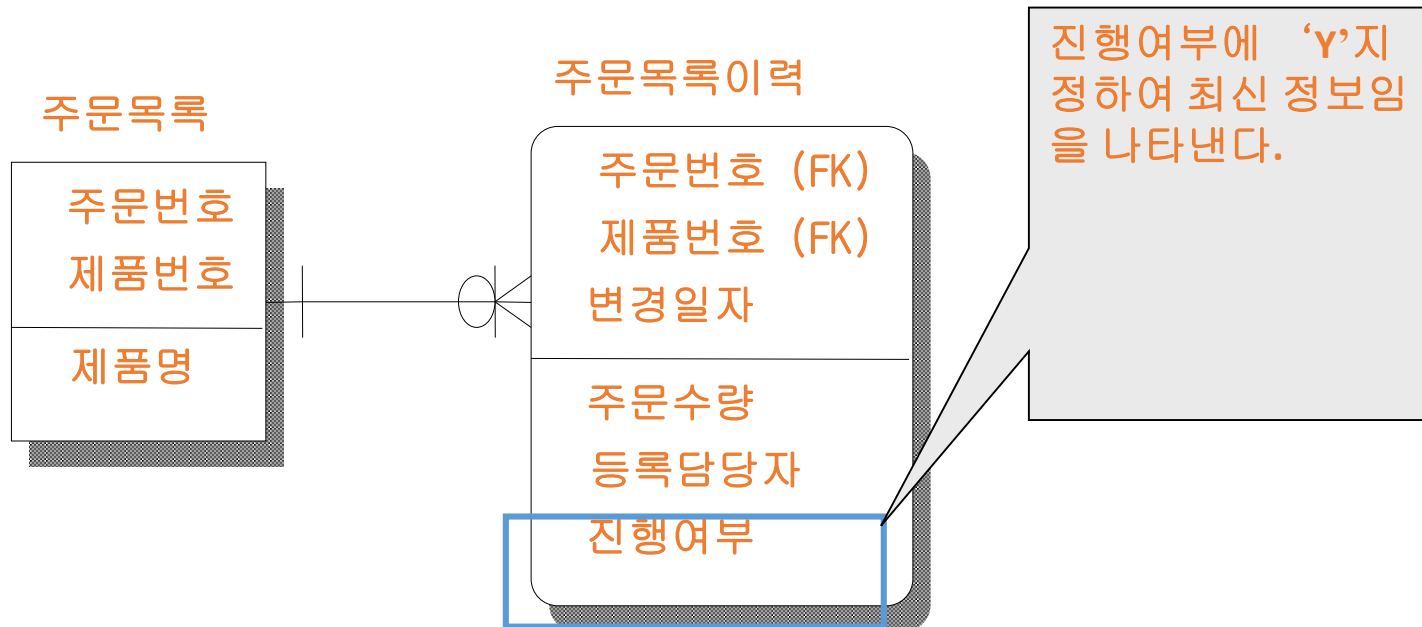
공사번호 공사일자
공사비 공사비누적

공사번호	공사일자	공사비	공사비누적
2023-001	2023년1월1일	7,500,000원	7,500,000원
2023-001	2023년3월5일	2,000,000원	9,500,000원
2023-001	2023년7월1일	9,721,239원	19,221,239원

3. 반정규화

컬럼의 반정규화

- 컬럼의 반정규화-최신컬럼
- 변경이력에 최신정보를 나타내는 컬럼 추가



3. 반정규화

컬럼의 반정규화

- 컬럼의 반정규화-최신컬럼
- SQL

```
SELECT A.주문번호, A.제품명, B.주문수량
FROM 주문목록 A, 주문목록이력 B
WHERE A.주문번호 BETWEEN '1001' AND '1100'
AND A.주문번호 = B.주문번호
AND A.제품번호 = B.제품번호
AND B.변경일자 = (SELECT MAX(C.변경일자)
                   FROM 주문목록이력 C
                   WHERE B.주문번호 = C.주문번호
                   AND B.제품번호 = C.제품번호)
```

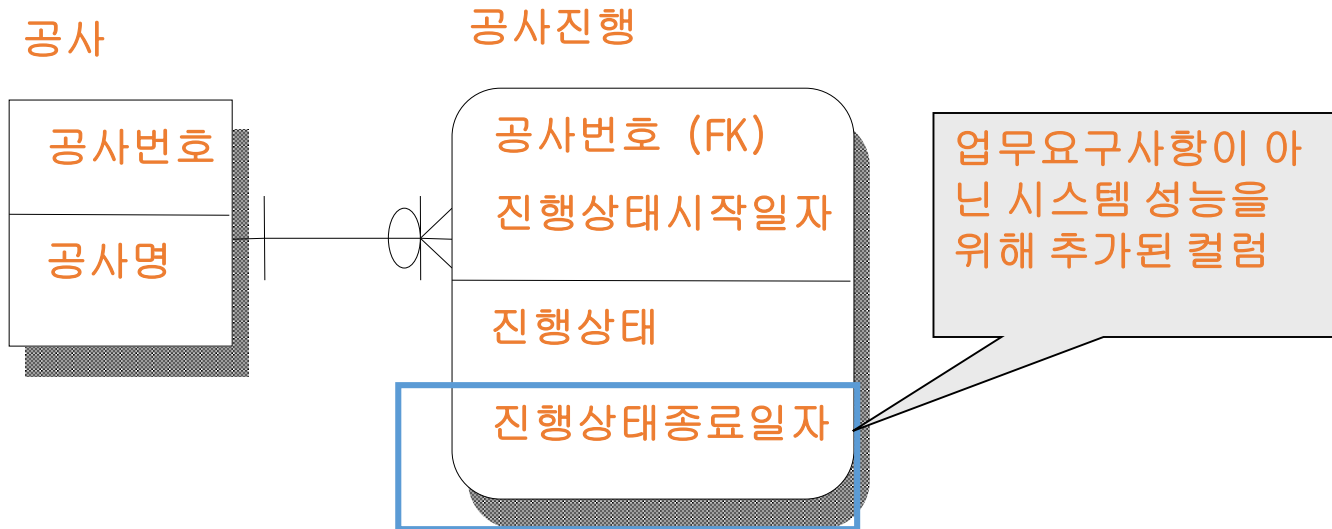


```
SELECT A.주문번호, A.제품명, B.주문수량
FROM 주문목록 A, 주문목록이력 B
WHERE A.주문번호 BETWEEN '1001' AND '1100'
AND A.주문번호 = B.주문번호
AND A.제품번호 = B.제품번호
AND B.진행여부 = 'Y'
```

3. 반정규화

컬럼의 반정규화

- 컬럼의 반정규화-진행이력컬럼
- 진행이력에 종료를 나타내는 컬럼 추가



3. 반정규화

컬럼의 반정규화

- 컬럼의 반정규화-진행이력컬럼
- SQL

```
SELECT A.공사번호, A.공사명, B.진행상태
FROM   공사 A, 공사진행 B,
       (SELECT 공사번호, MIN(진행상태시작일자) 진행상태종료일자
        FROM   공사진행
        WHERE  진행상태시작일자 > '2023년7월1일'
        GROUP BY 공사번호) C
WHERE  A.공사번호 = B.공사번호
AND    B.공사번호 = C.공사번호
AND    B.진행상태시작일자 = 진행상태종료일자
```

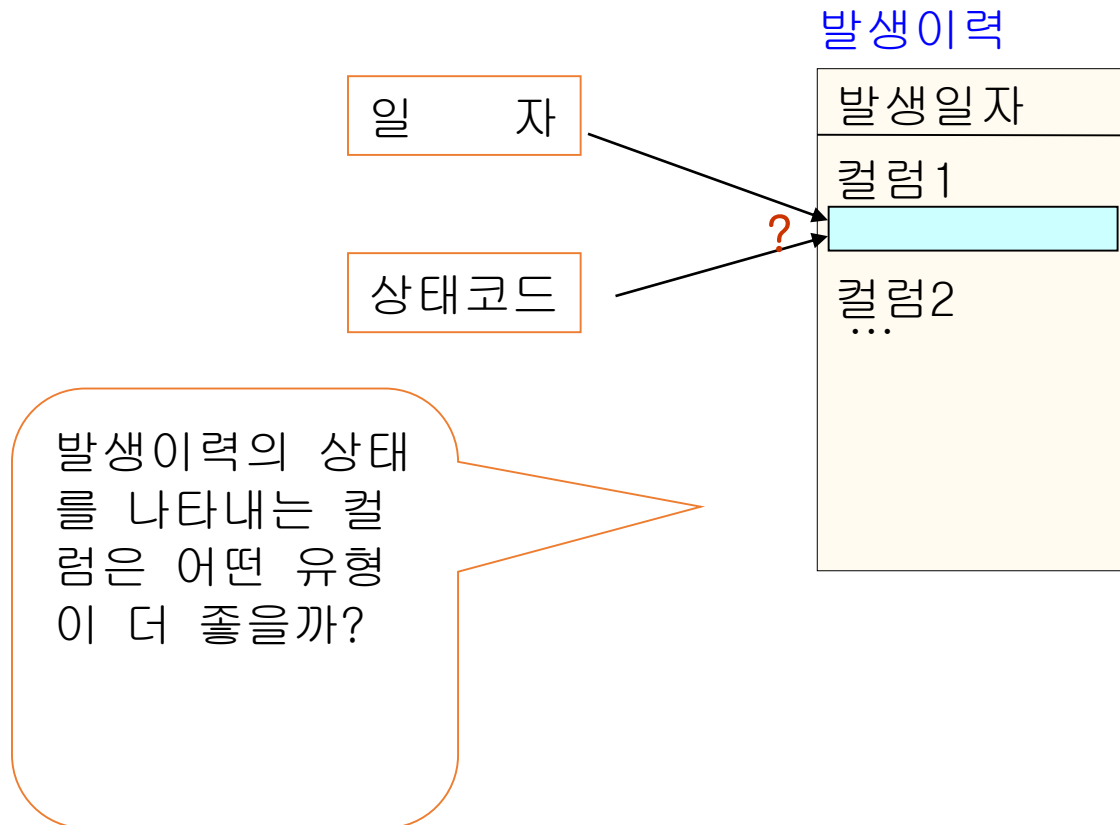


```
SELECT A.공사번호, A.공사명, B.진행상태
FROM   공사 A, 공사진행 B
WHERE  A.공사번호 = B.공사번호
AND    B.진행상태시작일자 <= '2023년7월1일'
AND    B.진행상태종료일자 > '2023년7월1일'
```

3. 반정규화

컬럼의 반정규화

- 컬럼의 반정규화-발생이력컬럼
- 발생이력에 컬럼 추가





3. 반정규화

컬럼의 반정규화

- 일자를 선택하는 경우
 - 시간을 나타내는 어떤 데이터에 대해 하나의 결과를 입력해야 하면서 기간도 파악할 수 있으면 더 좋은 경우에는 일자를 선택하도록 한다. 상태가 진행되었는지 미 진행되었는지는 일자 컬럼이 NULL상태거나 기본값상태인지(기본값을 지정하였을 경우) 아니면 결과가 입력된 값이었는지를 판단하여 데이터의 상태를 판단할 수 있다.
- 상태코드를 선택하는 경우
 - 시간을 나타내는 어떤 데이터에 대해 여러 개의 결과를 입력해야 하면서 기간은 의미가 없을 경우에는 상태코드를 선택하도록 한다. 상태코드의 장점은 해당 데이터에 대해 여러 개의 상황을 정의할 수 있다는 장점이 있지만 기간은 알 수 없는 단점이 있다.
- 일자와 상태코드 모두 선택해야 하는 경우
 - 시간을 나타내는 어떤 데이터에 대해 여러 개의 결과를 입력해야 하면서 기간은 의미가 있을 경우에는 상태코드와 일자에 대한 컬럼이 동시에 필요하게 된다.

3. 반정규화

컬럼의 반정규화

- 컬럼의 반정규화-발생이력컬럼에
- 발생이력에 컬럼 추가

예방정비

설비번호
정비일자
정비구분코드
정비계획일자
예방계획구분코드
예방정비부서코드
작업설명요약
예상시간
수행작업내용
정비종료일자

개량정비

설비번호
정비일자
정비구분코드
정비계획일자
개량계획구분코드
개량정비부서코드
교체대상여부
예상개량비용
예상시간
정비종료일자

고장정비

설비번호
정비일자
정비구분코드
고장형식코드
고장정비부서코드
정비등급
정비종료일자

발생이력의 상태를 나타내는 컬럼을 일자형태를 추가함으로써 데이터에 대한 상태(진행, 미진행)뿐만 아니라 기간도 파악할 수 있다.

3. 반정규화

컬럼의 반정규화

- 컬럼의 반정규화-복합PK
- 복합(COMPOSIT) PRIMARY KEY를 위한 컬럼 추가

```
SELECT COUNT(*)  
FROM 접수  
WHERE SUBSTR(접수.접수번호, 9, 4) = '1001'  
AND 접수상태 = '진행'
```

접수

접수번호
고객번호
진행상태

접수번호 구성

202307011001001

접수일자(YYYYMMDD) + 사업소번호(4) + 일련번호(3)

접수

접수번호
고객번호
진행상태
사업소번호

접수번호에 이미 사업소번호가 포함되어 있지만 개별적으로 처리할 일이 발생하였기 때문에 컬럼을 추가한다.

3. 반정규화

컬럼의 반정규화

- 컬럼의 반정규화-복합PK
 - 복합(COMPOSIT) PRIMARY KEY – 컬럼 추가

접수

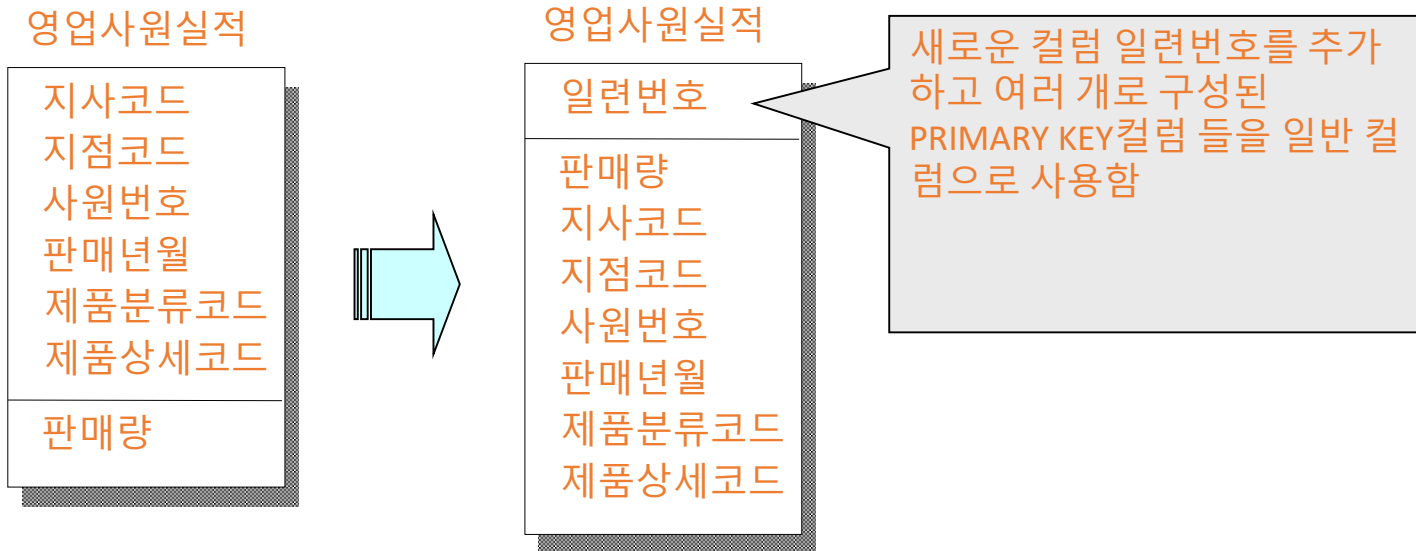
접수번호
고객번호
진행상태
사업소번호

접수번호에 이미 사업소번호가 포함되어 있지만 개별적으로 처리할 일이 발생하였기 때문에 컬럼을 추가한다.

3. 반정규화

컬럼의 반정규화

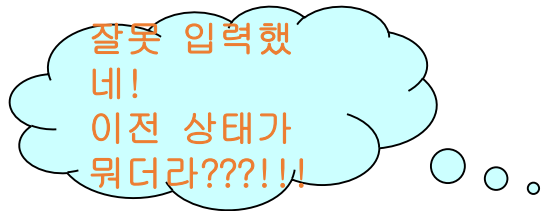
- 컬럼의 반정규화-다중컬럼PK
- 다중 컬럼으로 구성된 PRIMARY KEY - 컬럼추가



3. 반정규화

컬럼의 반정규화

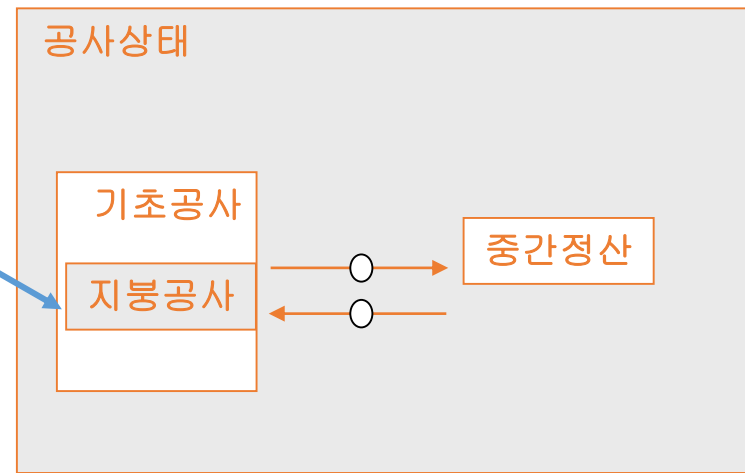
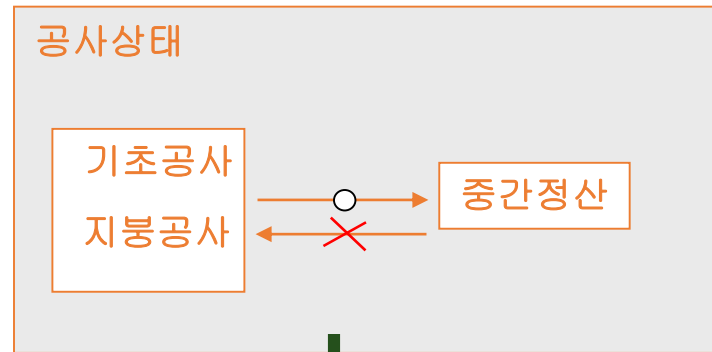
- 컬럼의 반정규화-오입력처리
- 오 입력 처리를 위한 컬럼 추가



공사

공사번호
공사명
착공일자
공사상태
이전공사상태

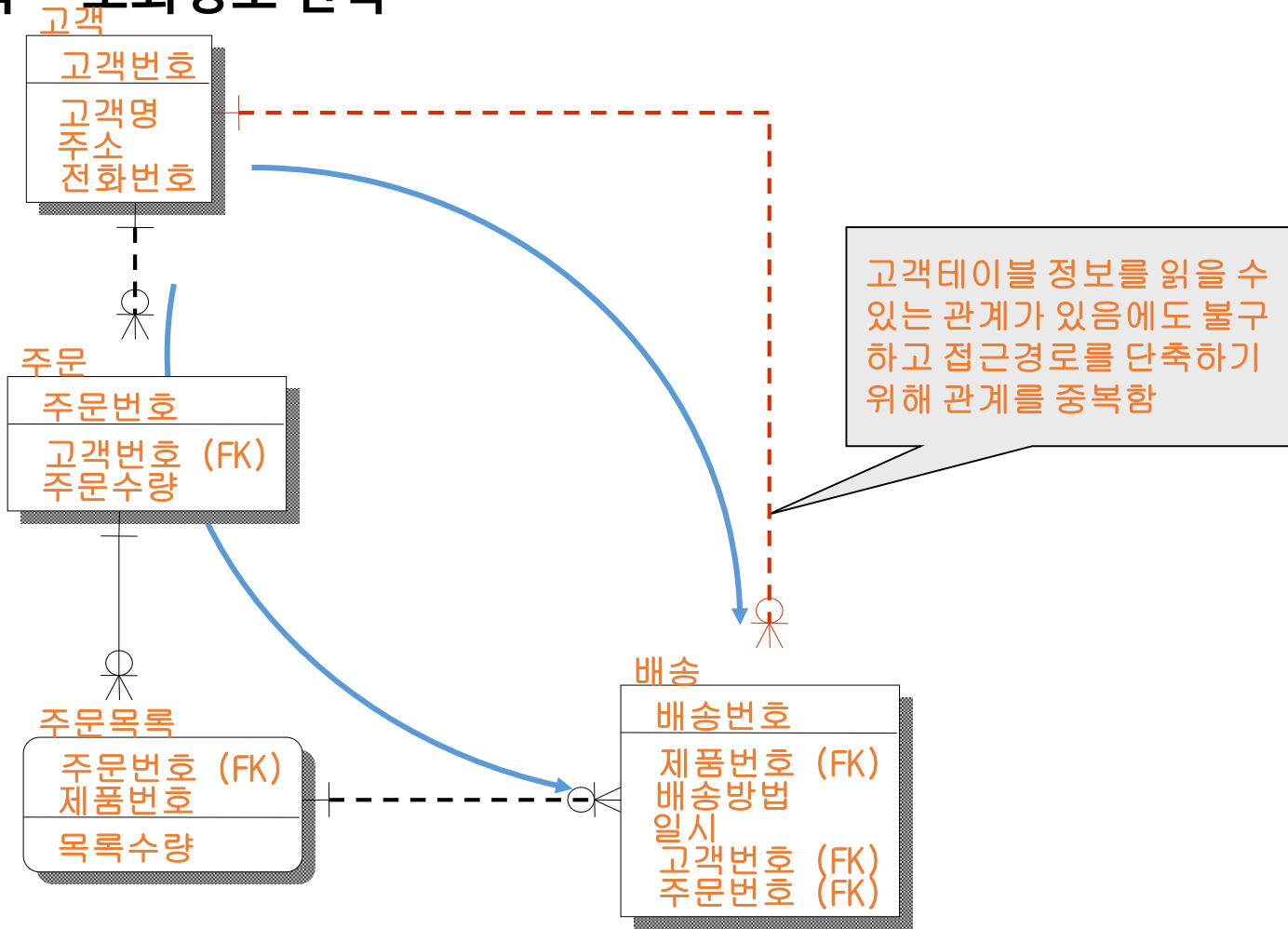
공사상태 값을 잘못입력하
여 다시 이전 상태로 돌아
가기 위해 이전공사상태 컬
럼을 추가함



3. 반정규화

컬럼의 반정규화

- 관계의 반정규화-조회경로단축
- 관계 중복 – 조회경로 단축



3. 반정규화

컬럼의 반정규화

- 관계의 반정규화-조회경로단축
- SQL

```
SELECT D.고객번호, D.고객명, A.일시  
FROM   배송 A, 주문목록 B, 주문 C, 고객 D  
WHERE  A.배송번호 = '20031001001'  
AND    배송.주문번호 = 주문목록.주문번호  
AND    배송.제품번호 = 주문목록.제품번호  
AND    주문.주문번호 = 배송.주문번호  
AND    고객.고객번호 = 주문.고객번호
```



```
SELECT B.고객번호, B.고객명, A.일시  
FROM   배송 A, 고객 B  
WHERE  A.배송번호 = '20031001001'  
AND    배송.고객번호 = 주문.고객번호
```



4. 오브젝트 설계

오브젝트 설계

- 무결성 제약의 정의
- 트랜잭션 분석
- 뷰 설계
- 인덱스 설계
- 용량설계
- 접근방법 설계
- 분산설계

4. 오브젝트 설계

무결성제약의 정의

- 무결성제약의 정의 위치





4. 오브젝트 설계

무결성의 종류

● 무결성의 종류

1. 입력 참조무결성

- 의존(DEPENDENT)
- 자동(AUTOMATIC)
- 기본(DEFAULT)
- 지정(CUSTOMIZED)
- NULL
- 미 지정

2. 수정 참조무결성

- 제한(RESTRICIT)
- 연쇄(CASCDE)

3. 삭제 참조무결성

- 제한(RESTRICIT)
- 연쇄(CASCDE)
- 기본(DEFAULT)
- 지정(CUSTOMIZED)
- NULL
- 미 지정

4. 오브젝트 설계

무결성의 종류

● 입력 무결성

- 입력 참조무결성을 유지하기 위해 6가지 기능이 존재할 수 있다.
- ① 의존(DEPENDENT) - 자신의 테이블에 데이터를 입력할 때, 참조하고 테이블의 PRIMARY KEY가 존재해야만 데이터입력이 가능하다.
- ② 자동(AUTOMATIC) - 자신의 테이블에 데이터를 입력할 때, 참조하고 있는 테이블의 PRIMARY KEY가 존재하지 않으면 PRIMARY KEY를 생성하고 자신의 테이블에 데이터를 생성한다.
- ③ 기본(DEFAULT) - 자신 테이블의 레코드를 입력할 때, 참조하는 테이블의 PRIMARY KEY를 기본값으로 바꾼 후 자신의 레코드를 입력한다.
- ④ 지정(CUSTOMIZED) - 사용자가 정의 해 놓은 일정한 조건을 만족한 이후에 자신의 레코드를 입력한다.
- ⑤ NULL - 자신 테이블의 레코드를 입력할 때 참조하는 테이블의 PRIMARY KEY가 없어도 그대로 입력한다. FOREIGN KEY 컬럼의 값은 NULL이 된다.
- ⑥ 미 지정 - 자신 테이블의 레코드를 입력할 때 참조하는 테이블의 PRIMARY KEY가 없어도 그대로 입력한다. FOREIGN KEY 컬럼의 값은 NULL이 된다.



4. 오브젝트 설계

무결성의 종류

- 수정 무결성

- 수정 참조무결성을 유지하기 위해 2가지 기능이 존재한다.
 - ① 제한(RESTRICT) - 자신 테이블의 PRIMARY KEY를 수정하려 하면 자신을 참조하고 있는 테이블의 FOREIGN KEY가 없어야 한다. 테이블의 FOREIGN KEY가 존재하면 자신 테이블의 PRIMARY KEY가 수정이 안된다.
 - ② 연쇄(Cascade) - 자신 테이블의 PRIMARY KEY를 수정하려 하면 참조되는 모든 테이블의 FOREIGN KEY를 수정하고 자신을 PRIMARY KEY를 수정한다.

4. 오브젝트 설계

무결성의 종류

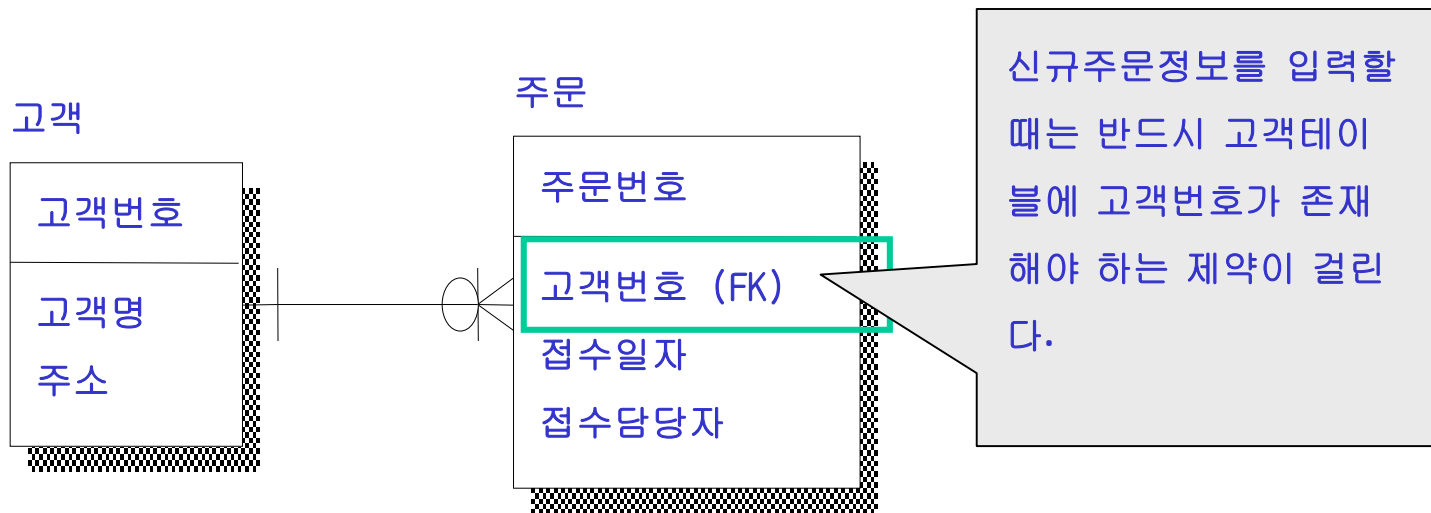
● 삭제 무결성

- 삭제 참조무결성을 유지하기 위해 6가지 기능이 존재할 수 있다.
- ① 제한(RESTRICT) - 자신 테이블의 레코드를 삭제하려 하면 자신을 참조하고 있는 테이블의 레코드가 없어야 한다. 있으면 삭제 안된다.
- ② 연쇄(CASCADE) - 자신 테이블의 레코드를 삭제하려 하면 참조되는 모든 테이블의 레코드를 삭제하고 자신을 삭제한다.
- ③ 기본(DEFAULT) - 자신 테이블의 레코드를 삭제할 때 참조되는 모든 테이블의 레코드들을 기본값으로 바꾼 후 자신의 레코드를 삭제한다.(비식별자관계(NON-IDENTIFYING RELATIONSHIP)에서 적용된다.)
- ④ 지정(CUSTOMIZED) - 사용자가 정의 해 놓은 일정한 조건을 만족한 이후에 자신의 레코드를 삭제한다.
- ⑤ NULL - 자신 테이블의 레코드를 삭제할 때 참조되는 모든 테이블의 레코드들을 NULL로 바꾼 후 자신의 레코드를 삭제한다.(비식별자관계(NON-IDENTIFYING RELATIONSHIP)에서 적용된다.)
- ⑥ 미 지정 - 자신 테이블의 레코드를 삭제해도 특별한 참조무결성 규칙을 적용하지 않는 경우이다. 조건 없이 삭제가 가능하다.

4. 오브젝트 설계

무결성의 종류

- 입력 무결성의 예



4. 오브젝트 설계

무결성의 종류

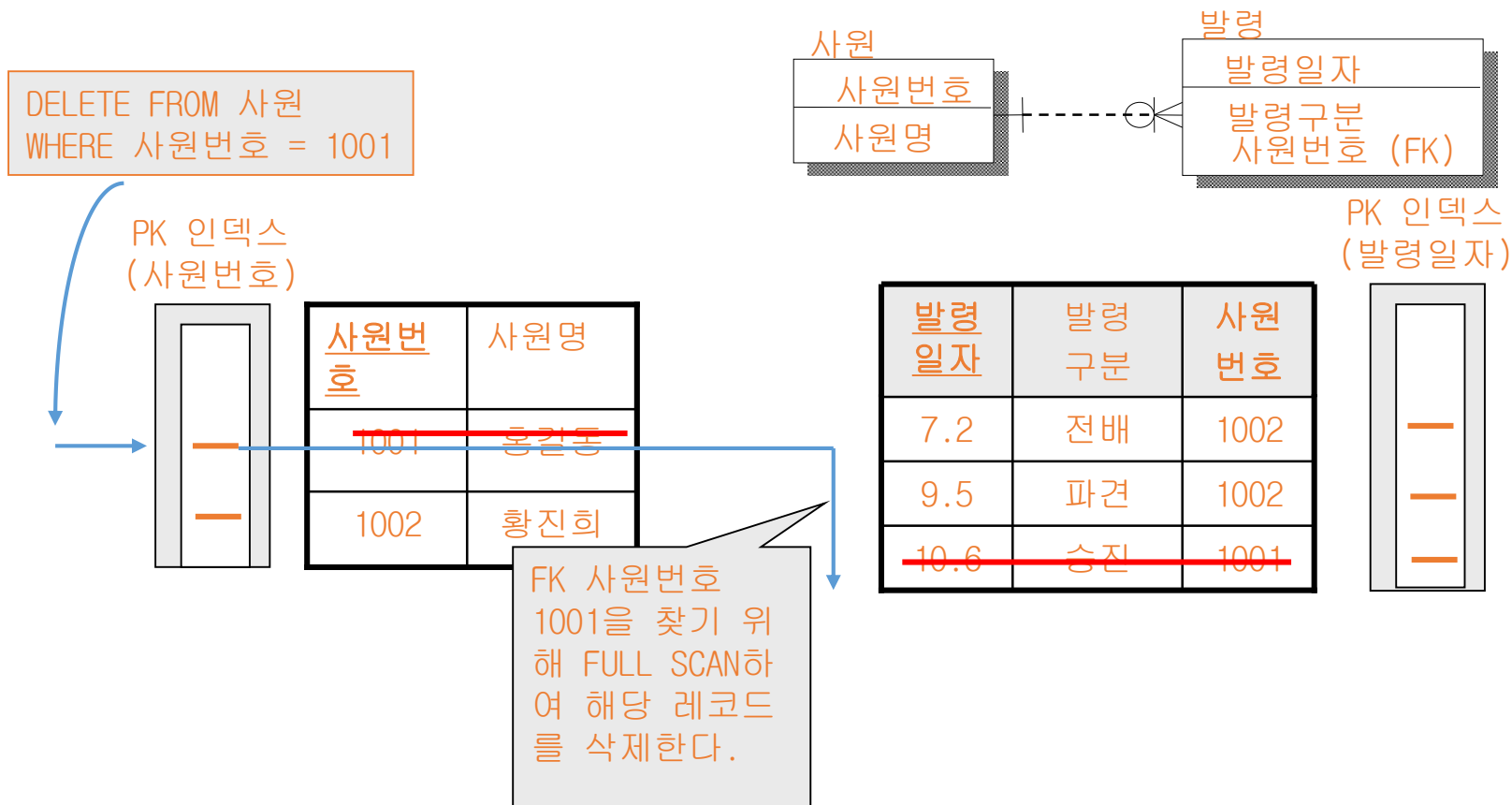
참조 무결성 설계의 예

부모 엔티티타입	자식 엔티티타입	관계 형식	NULL 허용	부모 입력	부모 수정	부모 삭제	자식 입력	자식 수정	자식 삭제	구현 방식
개량정비	정비부품	식별자관계	필수	제약없음	제한	제한	제한	제한	제약없음	DBMS
고장정비	정비부품	식별자관계	필수	제약없음	제한	제한	제한	제한	제약없음	DBMS
부품	설비	식별자관계	필수	제약없음	제한	제한	제한	제한	제약없음	DBMS
부품	부품	비식별자관계	NULL 허용	제약없음	제한	제한	제한	제한	제약없음	DBMS
부품	정비부품	식별자관계	필수	제약없음	제한	제한	제한	제한	제약없음	DBMS
설비	작업지시	식별자관계	필수	제약없음	제한	제한	제한	제한	제약없음	DBMS
설비	설비상세	식별자관계	필수	제약없음	제한	제한	제한	제한	제약없음	DBMS
설비	설비통계	식별자관계	필수	제약없음	제한	제한	제한	제한	제약없음	DBMS
설비	개량정비	식별자관계	필수	제약없음	제한	제한	제한	제한	제약없음	DBMS
설비	예방정비	식별자관계	필수	제약없음	제한	제한	제한	제한	제약없음	DBMS
설비	고장정비	식별자관계	필수	제약없음	제한	제한	제한	제한	제약없음	DBMS
예방정비	정비부품	식별자관계	필수	제약없음	제한	제한	제한	제한	제약없음	DBMS
자재	소요자재	식별자관계	필수	제약없음	제한	제한	제한	제한	제약없음	DBMS
작업설명서	작업수행	비식별자관계	NULL 허용	제약없음	제한	제한	제한	제한	제약없음	DBMS
작업수행	정비부품	비식별자관계	NULL 허용	제약없음	제한	제한	제한	제한	제약없음	DBMS
작업수행	작업자	식별자관계	필수	제약없음	제한	연쇄	제한	제한	제약없음	PROG
작업지시	작업수행	비식별자관계	NULL 허용	제약없음	제한	제한	제한	제한	제약없음	DBMS
작업지시	소요자재	식별자관계	필수	제약없음	제한	연쇄	제한	제한	제약없음	DBMS
작업지시	정비부품	비식별자관계	NULL 허용	제약없음	제한	제한	제한	제한	제약없음	DBMS
정비계획	예방정비	비식별자관계	NULL 허용	제약없음	제한	제한	제한	제한	제약없음	DBMS
정비계획	개량정비	비식별자관계	NULL 허용	제약없음	제한	제한	제한	제한	제약없음	DBMS

4. 오브젝트 설계

참조 무결성 설계 - FK

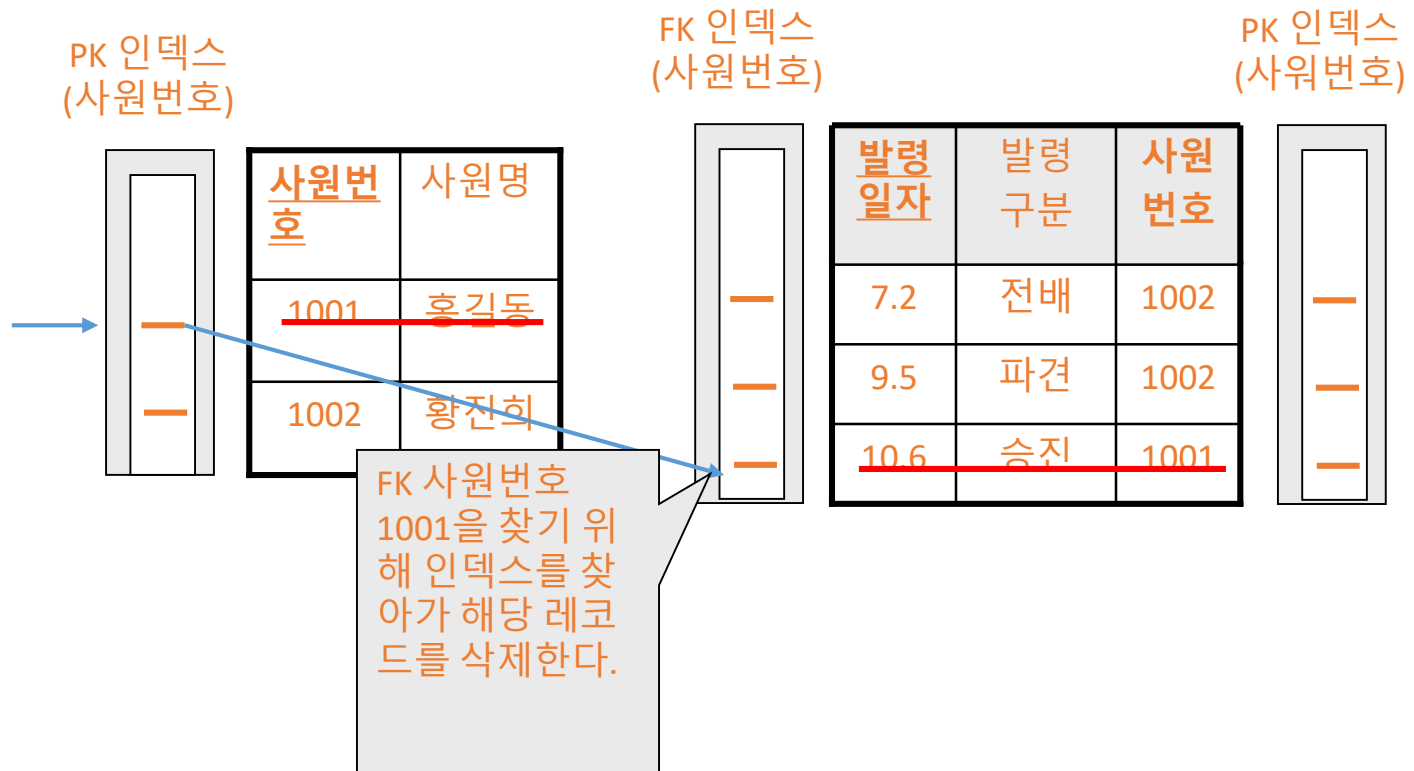
FOREIGN KEY 제약 시 연쇄(CASCADE) 삭제 - FULL SCAN



4. 오브젝트 설계

참조 무결성 설계 - FK

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



5. 트랜잭션 분석

트랜잭션 분석

- 트랜잭션 분석 위치



5. 트랜잭션 분석

트랜잭션?

- 업무처리의 단위(LOGICAL UNIT OF WORK)

업무처리의 단위(LOGICAL UNIT OF WORK)

트랜잭션 시작

트랜잭션 종료

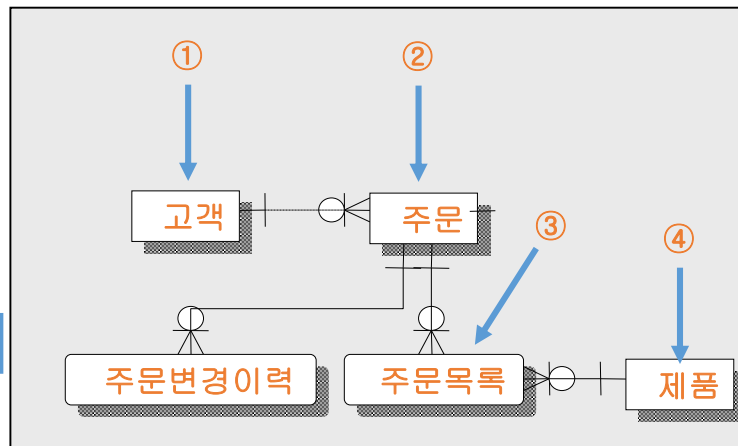


5. 트랜잭션 분석

트랜잭션 분석도

- 트랜잭션 분석도는 데이터베이스의 용량산정 및 디스크 구성전략의 근거자료로서 이용된다.

엔티티타입 ELEMENTARY PROCESS	고객	주문	주문목록	제품
신규고객을 등록한다	C			
제품주문을 신청한다	R	C	C	R
주문량을 변경한다		R	U	
주문을 취소한다		D	D	



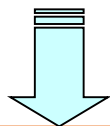
EP명	번호	테이블명	컬럼	CRUD	트랜잭션 당	트랜잭션 수	주 기
제품주문을 신청한다.	①	고객	고객번호, 고객명	R	1	200	일
	②	주문	주문번호, 주문일자, 고객번호	C	1	200	
	③	주문목록	주문번호, 제품번호, 단가	C	5	1,000	
	④	제품	제품번호, 제품명, 재고량	R	5	1,000	

5. 트랜잭션 분석

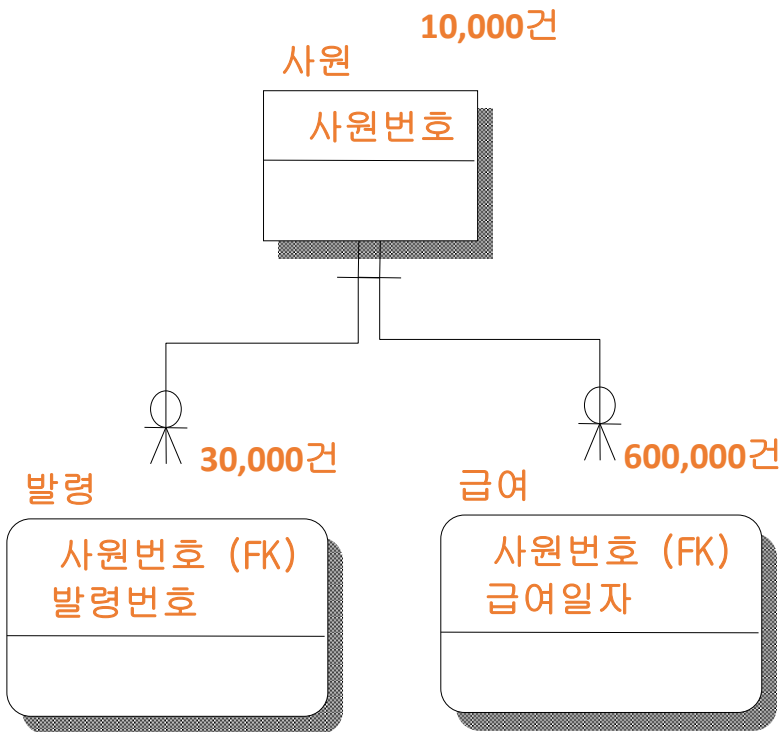
테이블 사용량 분석

- 용량산정 자료로 이용

사원 1명당
발령은 평균 3번 발생하고
급여는 상여가 포함되어 20번 발생한다.



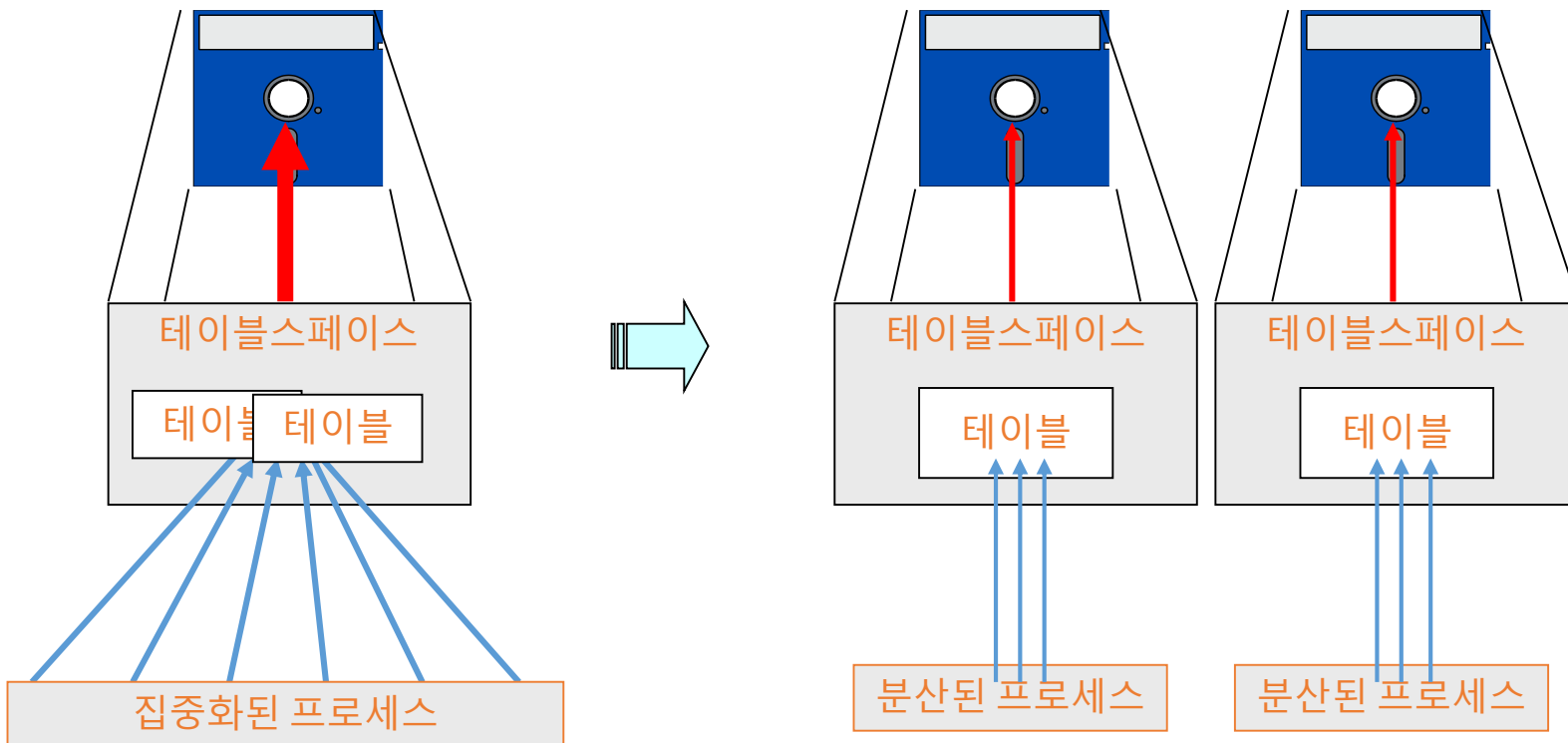
1년간
사원 10,000명당
발령은 평균 30,000번 발생하고
급여는 600,000번 발생한다.



5. 트랜잭션 분석

과도한 트랜잭션 분석

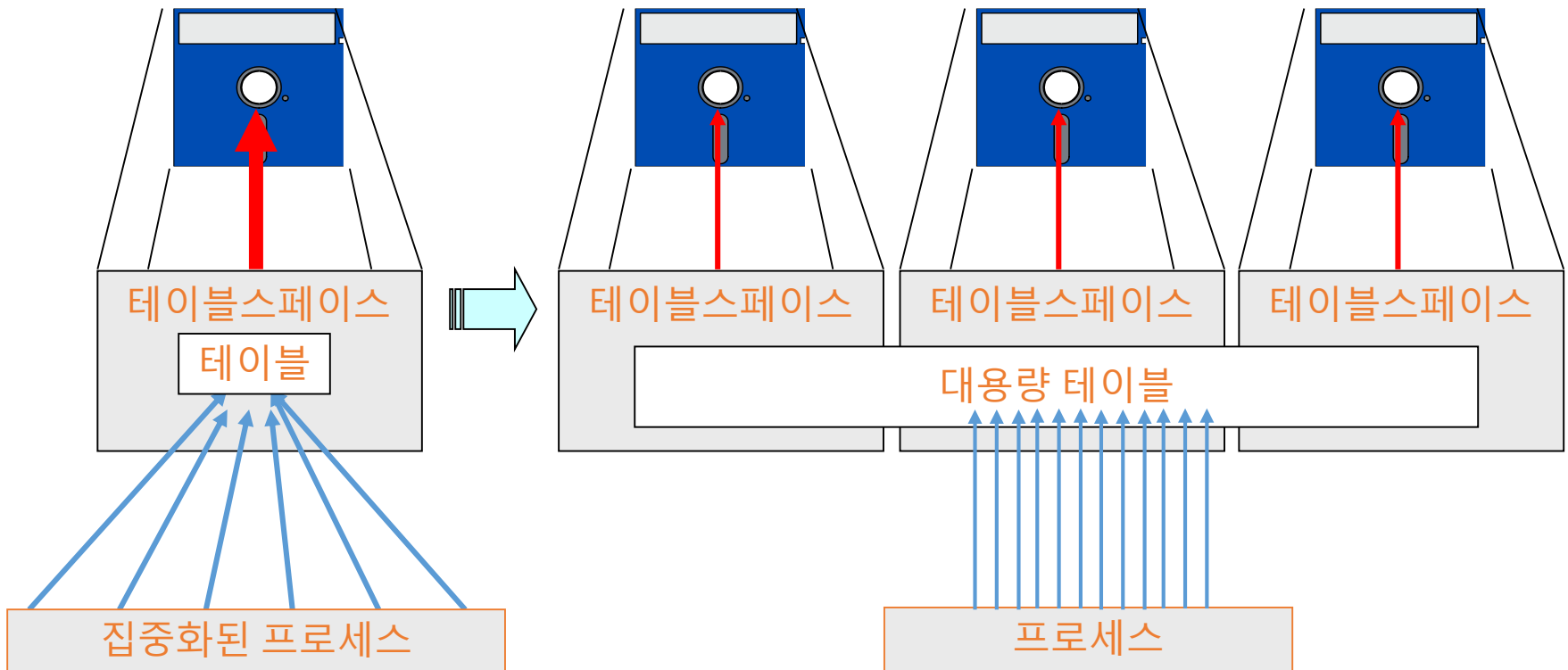
- 디스크 구성 자료로 이용
- 과도한 트랜잭션 디스크 집중에 의한 디스크 IO의 분산



5. 트랜잭션 분석

과도한 트랜잭션 분석

- 디스크 구성 자료로 이용
- 과도한 트랜잭션 디스크 집중에 의한 디스크 IO의 분산



5. 트랜잭션 분석

오라클트랜잭션 사용 예

- 오라클에서 사용하는 트랜잭션 제어 SQL문장의 예)
- **BEGIN TRANSACTION** 과 **END TRANSACTION** : 트랜잭션이 시작되는 것과 종료되는 것을 제어할 때
- **COMMIT** : 트랜잭션을 데이터베이스에 반영할 때
- **ROLLBACK** : 트랜잭션을 데이터베이스에 반영하지 않고 취소할 때
- **SAVEPOINT/ROLLBACK TO** : 트랜잭션 반영을 취소할 때 취소 지점을 명시하는 경우
- **SET TRANSACTION** : 읽기 전용으로 트랜잭션을 발생시킬 것인지 읽기/쓰기로 발생시킬 것인지를 지정
- **FOR UPDATE/LOCK TABLE** : 트랜잭션이 발생하는 동안 다른 사용자에게 의해 데이터가 변형되는 것을 방지.

6. 뷰 설계

뷰 설계

- 뷰 설계 위치





6. 뷰 설계

뷰의 정의

- 뷰(VIEW)란 데이터베이스 사용자들이 접근하도록 테이블을 이용하여 가상(VIRTUAL) 테이블을 구성하는 데이터베이스 오브젝트로서 별도의 저장공간을 가지지 않는다. (Materialized View는 저장공간을 가짐)
- 뷰는 실제 데이터 값이 저장되지는 않지만 실제 테이블에 대한 창이므로 뷰를 통해 테이블에 원하는 작업(SELECT, DELETE, UPDATE 등)을 할 수 있다.

뷰의 특징

- 복잡 테이블의 구조를 단순화 시킨다.
- 다양한 관점에서 데이터를 제시할 수 있다.
- 테이블 스키마 변경을 보호한다.
- 데이터의 보안을 유지한다.

6. 뷰 설계

뷰 설계

- 테이블 구조를 단순화 한다.

주문번호	신청자명	주문일자
A101	황진희	2023-07-01
A102	이순신	2023-08-15

주문번호(FK)	제품번호(FK)	단가
A101	BT92119	25,000원
A101	AI78119	50,000원
A102	BI33113	10,000원
A102	DE10110	30,000원



주문번호(FK)	신청자명	주문일자	주문번호(FK)	제품번호(FK)	단가
A101	황진희	2023-07-01	A101	BT92119	25,000원
A101	황진희	2023-07-01	A101	AI78119	50,000원
A102	이순신	2023-08-15	A102	BI33113	10,000원
A102	이순신	2023-08-15	A102	DE10110	30,000원



6. 뷰 설계

뷰 설계

- 뷰 생성 SQL

```
CREATE VIEW V_주문목록(주문번호, 신청자명, 주문일자, 제품번호, 단가) AS  
SELECT A.주문번호, A.신청자명, A.주문일자, B.제품번호, B.단가  
FROM 주문 A, 주문목록 B  
WHERE A.주문번호 = B.주문번호
```

6. 뷰 설계

뷰 설계

- 뷰 이용 SQL

```
SELECT A.주문번호, A.신청자명, A.주문일자, B.제품번호, B.단가  
FROM 주문 A, 주문목록 B  
WHERE A.주문번호 = B.주문번호  
AND B.주문번호 = 'A101'
```

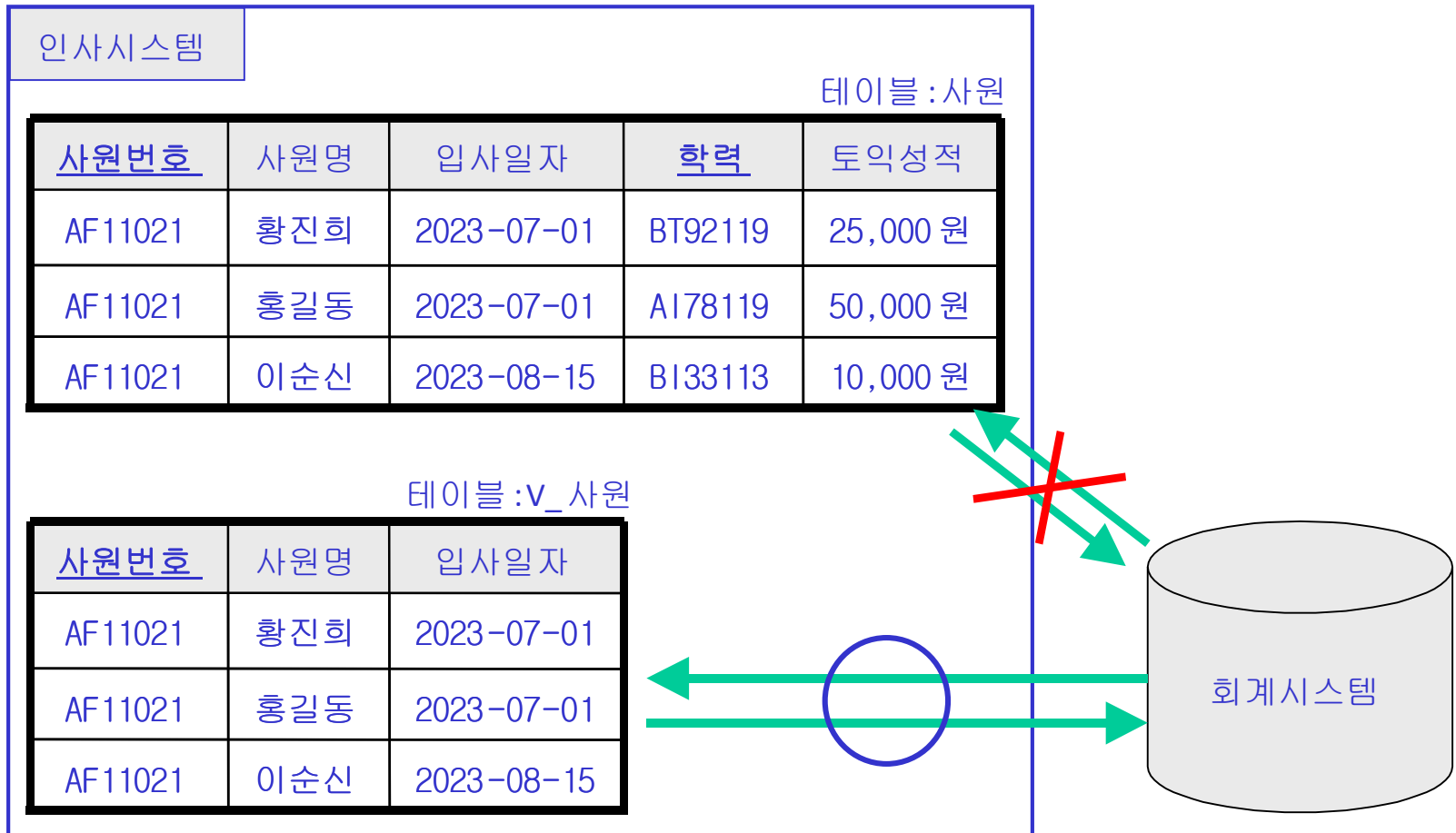


```
SELECT 주문번호, 신청자명, 주문일자, 제품번호, 단가  
FROM 주문목록  
WHERE 주문번호 = 'A101'
```

6. 뷰 설계

뷰 설계

- 다양한 관점에서 데이터를 제시할 수 있다.
- 데이터의 보안을 유지한다.



뷰 설계

- 논리적인 데이터의 독립성(LOGICAL DATA INDEPENDENCE)을 제공한다.
- 논리적인 데이터의 독립성이란 데이터베이스에 테이블의 구조가 바뀌어도 구조의 변화에 의해 뷰가 영향을 받지 않는다는 것이다. 업무적인 필요에 의해 뷰에 항목을 추가하거나 삭제할 수는 있지만 테이블의 스키마(SCHEME)가 변경되어 뷰가 바뀌지는 않는다.
- 논리적인 데이터 독립성에는 스키마가 추가되거나 삭제되는 형태가 있고 또한 컬럼의 위치가 바뀌는 형태와 같이 테이블 변형(RESTRUCTURING)이 발생하는 경우가 있다.
- 예) 앞의 사원과 V_사원의 예를 들어보자. 만약 사원의 정보에 급여 등급이라는 항목이 추가되었다고 하면 V_사원테이블은 다시 재구성해야 할까?

6. 뷰 설계

기타 뷰의 특징

- 데이터를 조회하는 기능(SELECT)에는 제한이 없다.
- 데이터를 입력, 수정, 삭제하는 기능에는 허용하지 않는 뷰가 존재한다.
- 뷰에는 인덱스, 클러스터링, 해쉬클러스터 등을 지정하는 것이 불가능하다.
- 테이블에 뷰를 이용하여 적절한 테이블 접근방법을 유도할 수 있다.

6. 뷰 설계

뷰 정의 방법

1. 뷰의 대상이 되는 테이블을 선정한다.
2. 뷰의 대상이 되는 컬럼을 선정한다.
3. 뷰 정의서를 작성한다.

뷰 정의서

뷰명	뷰설명	관련 테이블	컬럼명	데이터타입
V_EMP	회계시스템과 인터페이스	EMP	EMPNO	VARCHAR2(6)
			EMPNM	VARCHAR2(40)
			HIREDATE	DATE
V_ORDERITEM	주문과 주문목록을 함께 처리	ORDER	ORDERNO	VARCHAR2(6)
			ORDERNM	VARCHAR2(40)
			ORDERDATE	DATE
		ORDERITEM	ITEMNO	VARCHAR2(6)
			PRICE	NUMBER(10)

7. 인덱스 설계

인덱스 설계

- 인덱스 설계의 위치



7. 인덱스 설계

인덱스 설계 정의

- 인덱스(INDEX)는 테이블의 저장된 로우(ROW)를 식별 가능하도록 컬럼의 값과 그 값을 포함하고 있는 로우(ROW)의 논리적인 주소를 별도의 저장구조를 만들어 저장하는 것이다.

인덱스 설계 순서

1. 인덱스 대상선정

대상 테이블 선정
PK 인덱스 선정
FK 인덱스 선정
대상 컬럼 선정

2. 인덱스최적화

인덱스효율검토
인덱스데이터타입적용
인덱스정렬
클러스터링검토

3. 인덱스정의서작성

인덱스정의서작성

7. 인덱스 설계

인덱스 대상 선정

- 테이블내에서 자주 이용이 되며 분포도가 좋은 컬럼, 즉 평균 분포도가 10 ~ 15% 이내의 컬럼은 인덱스 지정 대상 컬럼이 된다. (이하의 컬럼)
- 컬럼이 테이블내에서 자주 사용이 된다는 의미는 SQL문장에서 조인에서 이용되거나, ORDER BY, GROUP BY, UNION, DISTINCT 에서 이용되는 컬럼이다.
- 여러 개의 컬럼이 항상 같이 이용되는 컬럼도 대상이 된다.

$$\text{분포도(\%)} = \frac{\text{데이터별 평균 로우 수}}{\text{테이블의 총 로우 수}} \times 100$$

인덱스 최적화

- **인덱스효율검토**
 - 인덱스 컬럼은 가능하면 수정이 자주 발생되지 않는 컬럼을 선정하도록 한다.
 - 평균분포도가 10 ~ 15%이내의 양호한 분포도를 가졌다고 할지라도 분포가 일정하지 하지 않고 기형적으로 불균형적이면 인덱스를 설정하지 않는 것이 낫다
 - 한 테이블에 인덱스의 개수가 5개를 초과한 경우에는 테이블의 입력, 수정, 삭제가 자주 발생하는지 검증하고 만약 자주 발생한다고 하면 꼭 필요한 인덱스를 남기고 삭제하도록 한다.
- **인덱스데이터타입적용**
 - 데이터모델링에서 도메인을 업무에 적절하게 지정했다고 하여도 적절한 인덱스 활용을 위해 데이터타입을 변형할 수도 있다.
 - 데이터길이가 변하는 컬럼은 반드시 가변길이타입(오라클의 경우 VARCHAR2() 형식)을 사용하도록 한다.
 - 데이터타입이 날짜인 경우는 보통 시,분,초 단위로 조회하지 않고 날짜 단위로 처리하므로 DATE형식보다는 VARCHAR2(8)형식을 이용한다.

7. 인덱스 설계

인덱스 최적화

● 인덱스정렬

- 인덱스는 유형에 따라 **UNIQUE INDEX, NON UNIQUE INDEX, CLUSTERED INDEX**가 존재하고 정렬상태에 따라 **순정렬 인덱스(ASC)**와 **역정렬 인덱스(DESC)**가 존재한다.
- 인덱스 중에 역정렬(DESC) 순서로 사용하는 인덱스를 검토한다.
- 여러 개의 컬럼이 이용되는 경우는 **결합인덱스**를 생성한다. 결합인덱스를 생성할 때는 인덱스 순서에 따라 성능을 많은 영향을 미친다. 결합인덱스에 앞쪽에 오는 컬럼이 범위를 줄일 수 있는 컬럼이 올 수 있도록 설계한다. 앞쪽에 위치할 수 있는 컬럼은 항상 사용이 되어야 하며 가능하면 '='로 비교되어야 하며 분포도가 좋은 컬럼이 되도록 선정한다.

인덱스 최적화

- 클러스터링 검토
 - 특정 컬럼에 대한 데이터를 읽는 성능은 향상해야 하지만 인덱스를 사용할 수 없는 경우는 클러스터링을 사용하는 것을 검토한다.
 - 인덱스는 평균분포도가 좁은 것이 좋지만 클러스터링 분포도가 넓은 것이 더 좋다.
 - 인덱스와 마찬가지로 클러스터링도 조회속도는 향상시키지만 입력, 수정, 삭제에 대한 성능은 저하시킨다.
 - 대량의 데이터를 처리하는 트랜잭션(OLTP)은 가능하면 클러스터링을 사용하지 않는 것이 더 낫다.

7. 인덱스 설계

인덱스 정의서 작성

● 인덱스 정의서 작성

엔티티 타입명	테이블 명	인덱스 명	컬럼 명	데이터 타입	인덱스 스페이스명	인덱스유형 (유일여부)	정렬	PK	FK
설비	TMS001	PK_TMS001	EQUIP_NO	VARCHAR2(6)	IS_EQUIP	Yes	ASC	PK	FK
		I_TMS001_01	INSTALL_DATE	DATE		No	ASC		
			USE_FLAG	CHAR(1)		No	ASC		
			EQUIP_NO	VARCHAR2(6)		No	ASC		
부품	TMS005	PK_TMS005	PART_NO	VARCHAR2(6)	IS_EQUIP	Yes	ASC	PK	
작업설명서	TMS015	PK_TMS015	WORK_EXP_DOCU_NO	VARCHAR2(6)	IS_EQUIP	Yes	ASC	PK	
정비부품	TMS021	PK_TMS021	EQUIP_NO	VARCHAR2(6)	IS_EQUIP	Yes	ASC	PK	FK
			REPAIR_DATE	DATE		Yes	ASC	PK	FK
			REPAIR_CLASS_CD	CHAR(1)		Yes	ASC	PK	FK
			EQUIP_NO	VARCHAR2(6)		Yes	ASC	PK	FK
		I_TMS021_01	EQUIP_NO	VARCHAR2(6)		No	ASC		FK
			WORK_INDI_NO	VARCHAR2(6)		No	ASC		FK
		I_TMS021_02	EQUIP_NO	VARCHAR2(6)		No	ASC		FK
			WORK_EXEC_NO	VARCHAR2(6)		No	ASC		FK
		I_TMS021_03	EQUIP_NO	VARCHAR2(6)		No	ASC		FK
정비계획	TMS023	PK_TMS023	REPAIR_CLASS_CD	CHAR(1)	IS_EQUIP	Yes	ASC	PK	
			REPAIR_PLAN_DATE	DATE		Yes	ASC	PK	
		I_TMS023_01	REPAIR_CHARGE_EMPNO	VARCHAR2(5)		No	ASC		

8. 용량 설계

용량 설계

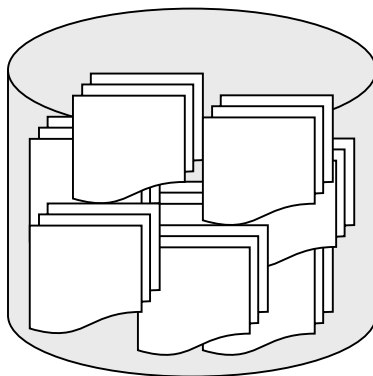
- 용량 설계의 위치



8. 용량 설계

용량 설계의 정의

- 물리 데이터베이스 설계에서 데이터가 저장되기 위한 공간을 정의하는 작업을 데이터베이스 용량설계라 한다.



테이블	월 발생 건수	테이블크 기
T0001	1,000	3G
T0002	100	10G
...

8. 용량 설계

데이터베이스 용량 산정 순서

- 데이터베이스 용량 산정 순서



8. 용량 설계

용량 설계

● 오브젝트 용량산정

위치	엔티티타입명	테이블명	ROW길이	보존기간	초기건수	발생건수	발생주기	년증가율
본사	사용자코드	CODE	44 BYTE	10년	2,500건	100건	년	5건
서울지사	사원	EMP	120 BYTE	10년	8,021건	80건	월	50건
서울지사	부서	DEPT	44 BYTE	10년	2,330건	10건	분기	10건
인천지사	제품	ITEM	260 BYTE	5년	3,211,874건	3,700건	일	200건

● 오브젝트 용량산정

분산위치	테이블명	테이블용량	테이블스페이스명	테이블스페이스 용량(40%확장고려)	데이터파일명
본사	TABLE1	30M	TS001	150M + 60M = 210M	DF001.DBF01
본사	TABLE2	20M			
본사	TABLE3	40M			
본사	TABLE4	60M			
본사	TABLE5	10M	TS002	615M + 246M = 861M	DF002.DBF01
본사	TABLE6	5M			DF002.DBF02
본사	TABLE7	100M			
본사	TABLE8	500M			



용량 설계

● 데이터파일 용량 산정

디스크	데이터파일 디렉토리	데이터 파일명	데이터파일 용량	테이블스페이스	테이블스페이스용량	비고
/DISK1	/DISK1/ORADATA/ORA8/DB1	DF001.DBF01	320M	TS001	210M	
				TS005	100M	
		DF002.DBF01	500M	TS002	861M	DF002.DBF01와 공유
/DISK2	/DISK2/ORADATA/ORA8/DB1	DF001.DBF02	110M	TS003	100M	
		DF002.DBF02	500M	TS002	861M	

● 디스크 용량 산정

디스크	데이터파일 디렉토리	디스크용량	사용된 디스크용량	디스크 사용비율	데이터 파일명	데이터파일 용량
/DISK1	/DISK1/ORADATA/ORA8/DB1	2G	820M	41%	DF001.DBF01	320M
					DF002.DBF01	500M
/DISK2	/DISK2/ORADATA/ORA8/DB1	2G	1,510M	75%	DF001.DBF02	500M
					DF002.DBF02	110M
					DF003.DBF01	900M



9. 접근방법 및 분산설계

접근방법 및 분산설계

- 접근방법 및 분산설계 위치





9. 접근방법 및 분산설계

접근방법 설계 정의

- 접근방법 설계를 한다는 것은 데이터베이스에 접근하여 데이터를 처리 할 때 어떤 방식을 통해 할 것인지를 설계하는 것이다.

접근방법의 구분

직접접근

- 스캔

간접접근

- B-TREE 인덱스
- 해싱
- 비트맵인덱스

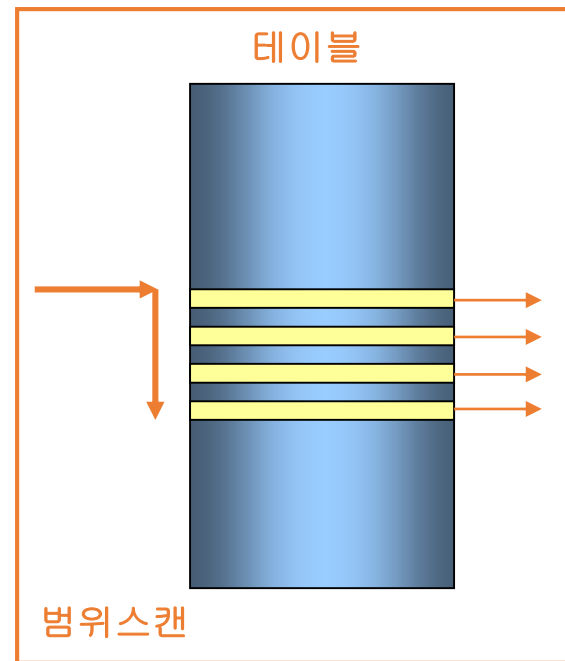
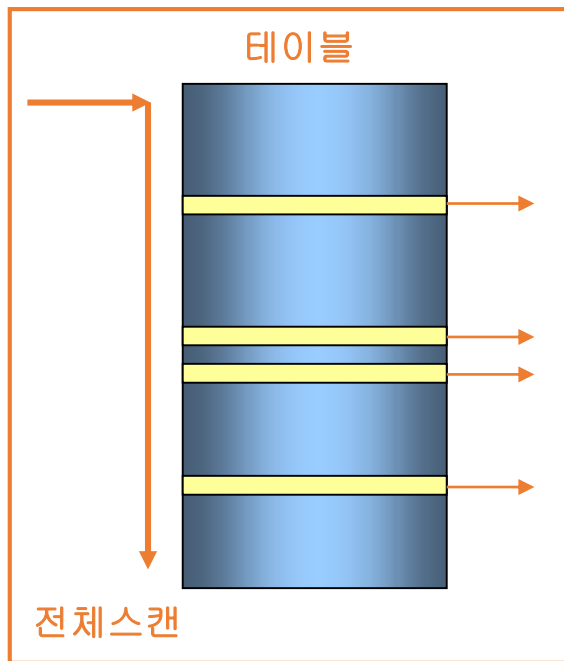
저장방법에 따른 접근

- 클러스터링

9. 접근방법 및 분산설계

스캔방식의 두 가지

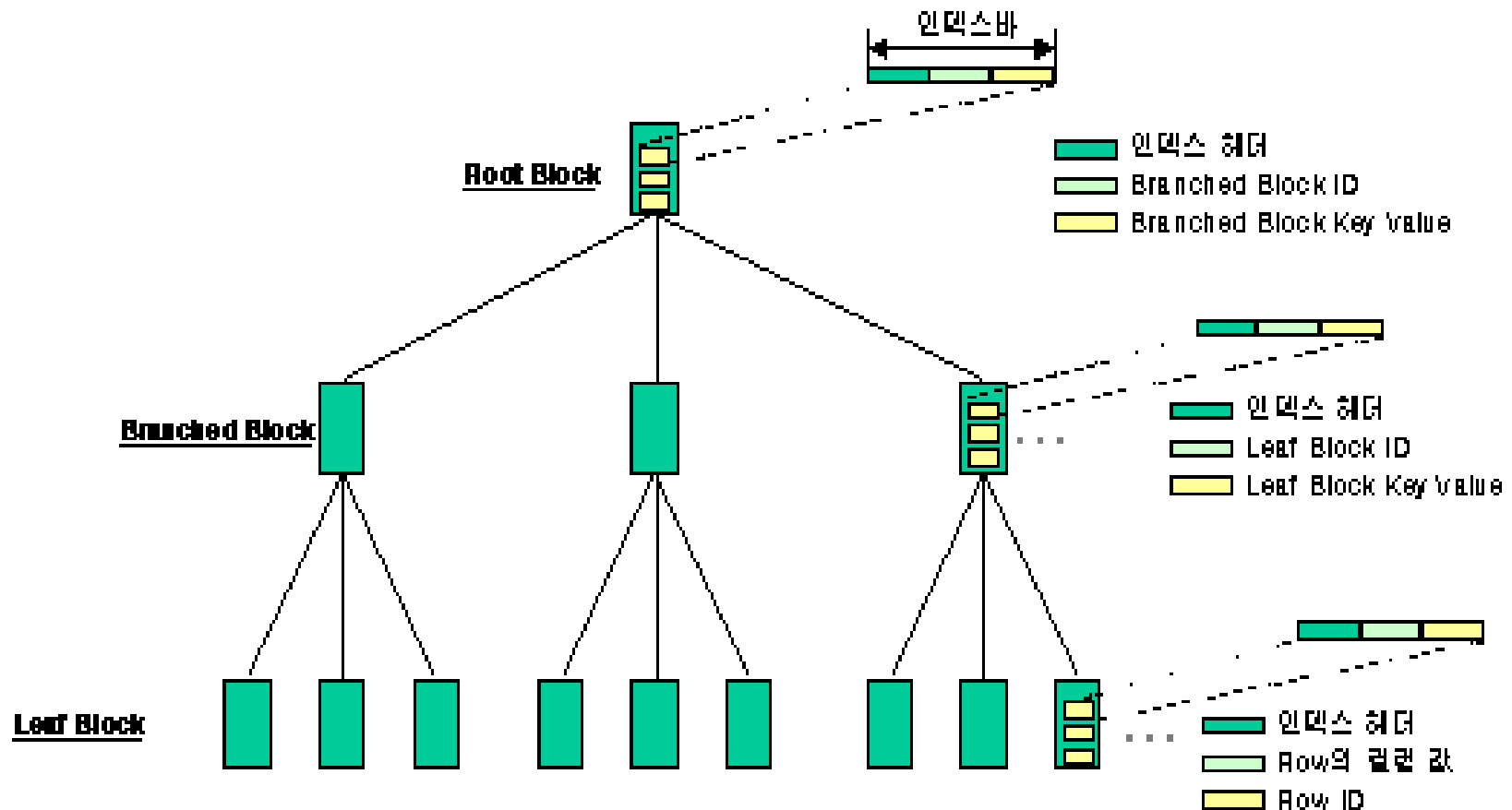
- 스캔방식의 두 가지



9. 접근방법 및 분산설계

B*Tree 인덱스의 구조

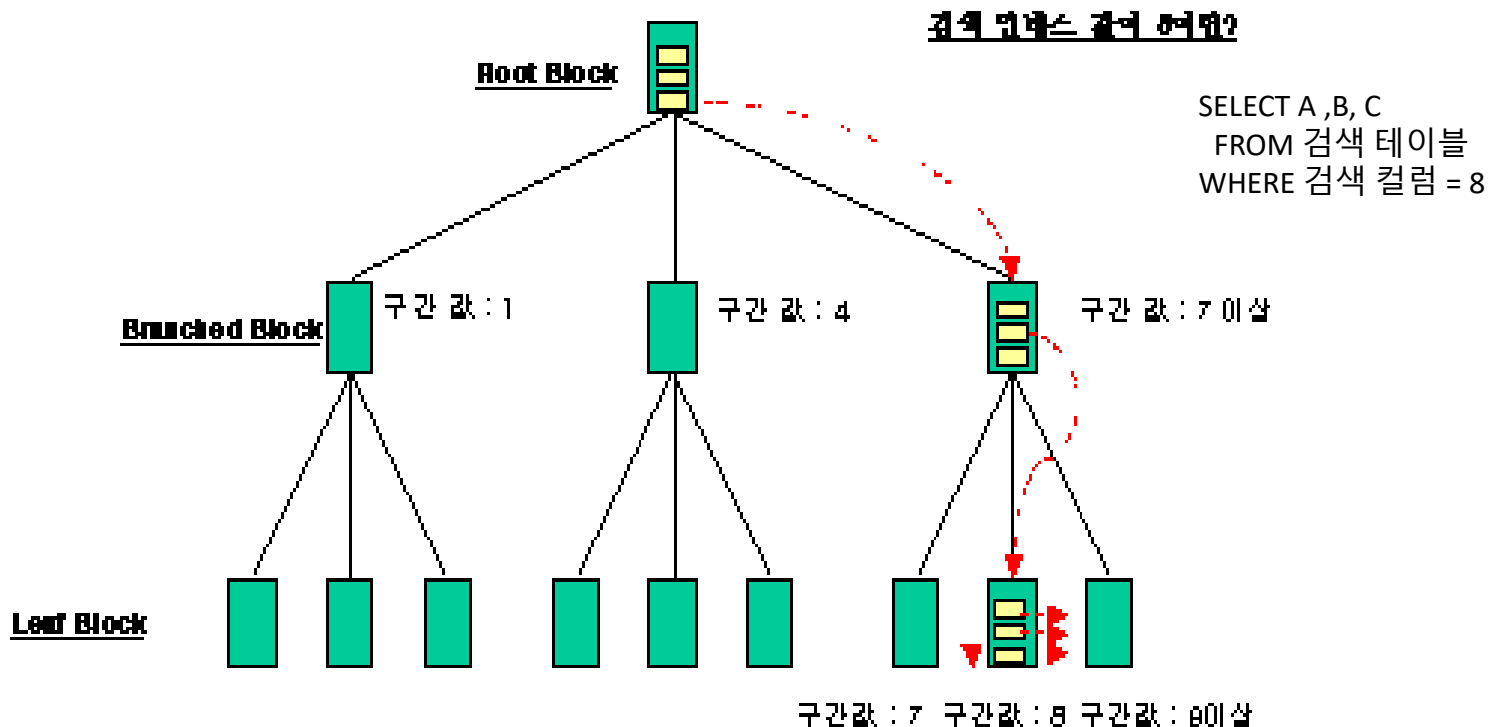
● B*Tree 인덱스의 구조



9. 접근방법 및 분산설계

B*Tree 인덱스의 구조

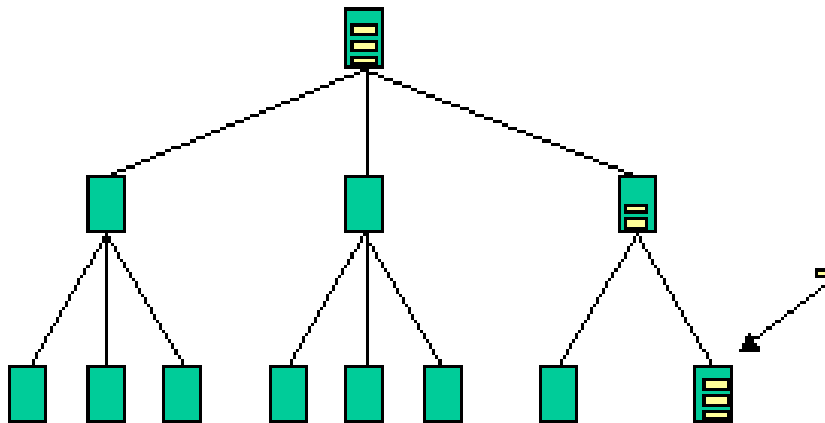
- B*Tree 인덱스의 검색 매카니즘



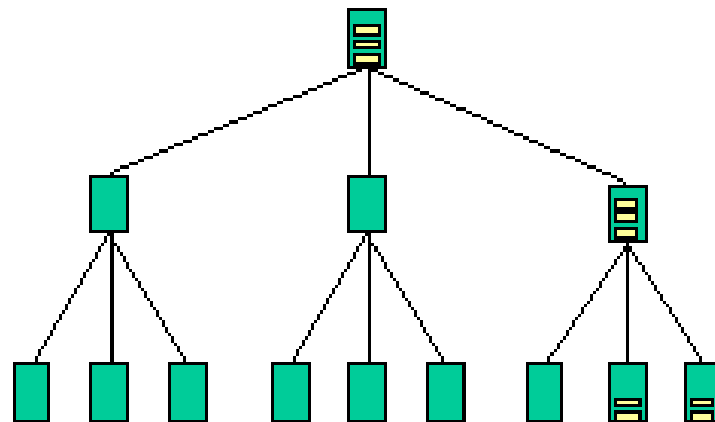
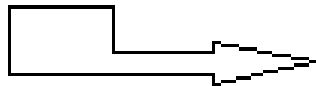
9. 접근방법 및 분산설계

B*Tree 인덱스의 구조

● B*Tree 인덱스의 입력, 수정, 삭제 매카니즘



새로운 인덱스 값 추가 전

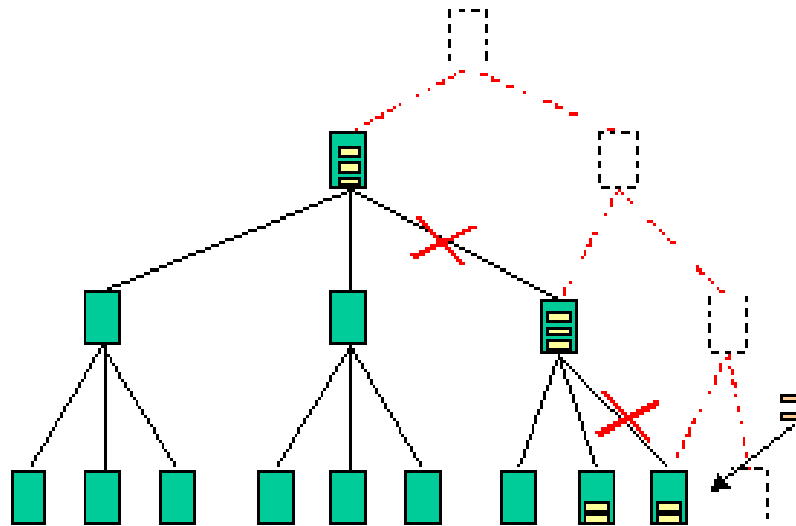


새로운 인덱스 값 추가 후

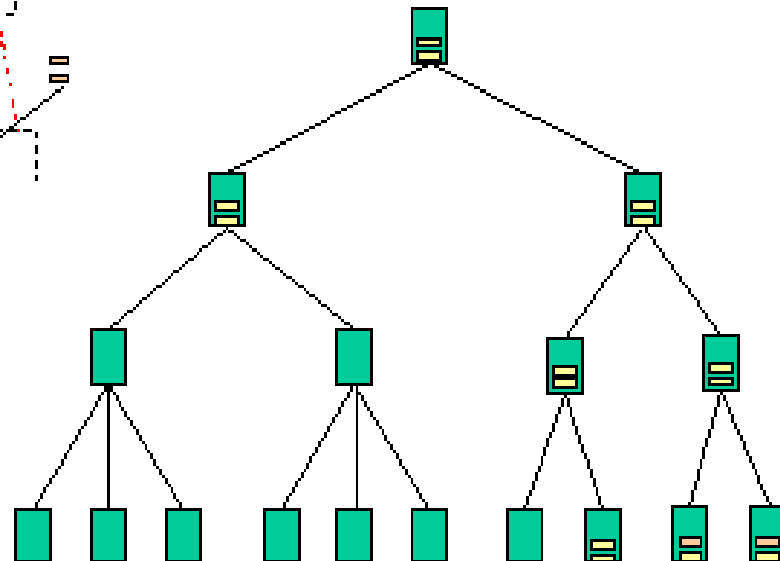
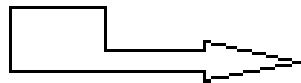
9. 접근방법 및 분산설계

B*Tree 인덱스의 구조

● B*Tree 인덱스의 입력, 수정, 삭제 매카니즘



새로운 인덱스 값 추가 전

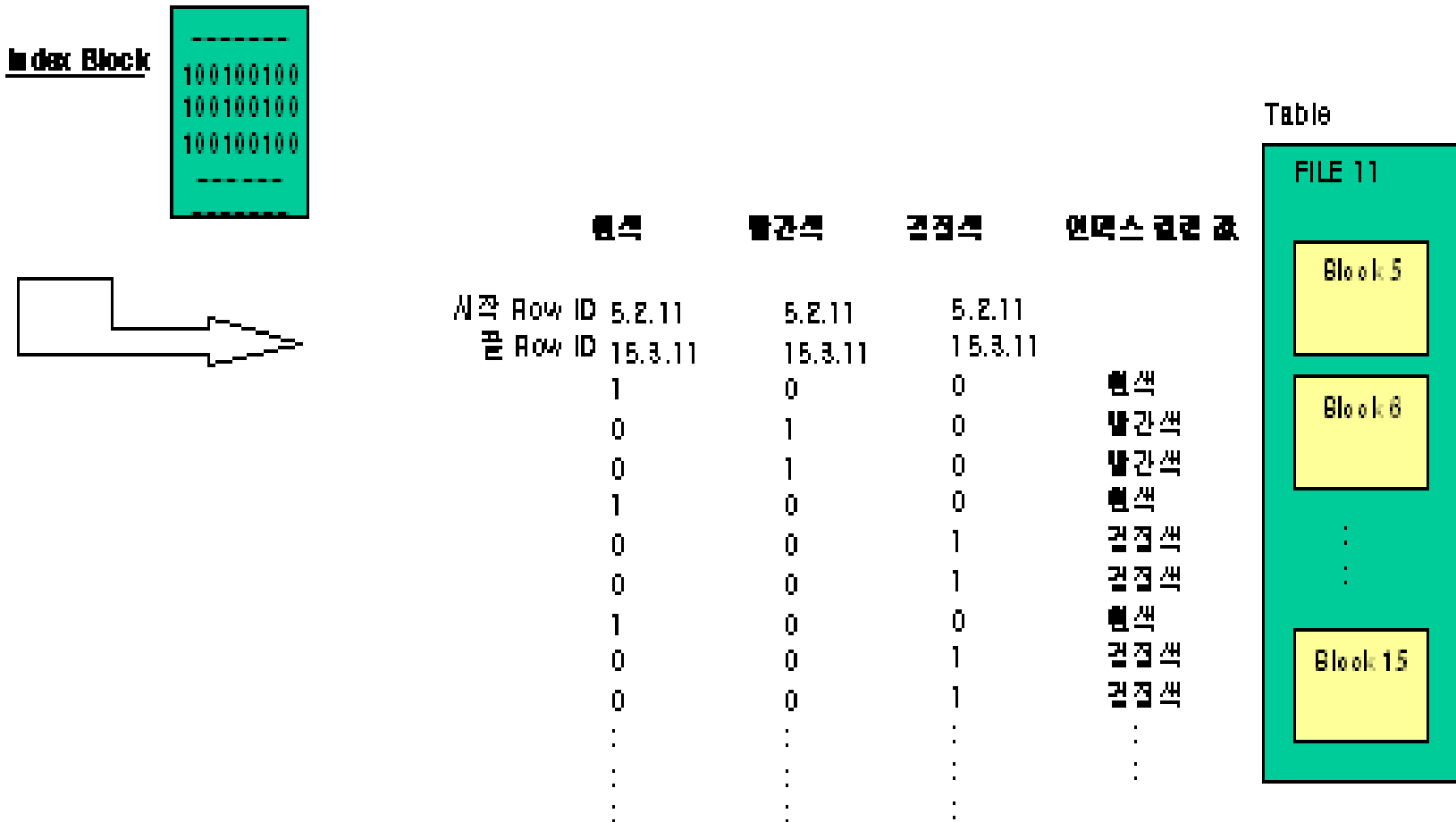


새로운 인덱스 값 추가 후

9. 접근방법 및 분산설계

BitMap 인덱스

● BitMap 인덱스의 매카니즘





9. 접근방법 및 분산설계

BitMap 인덱스

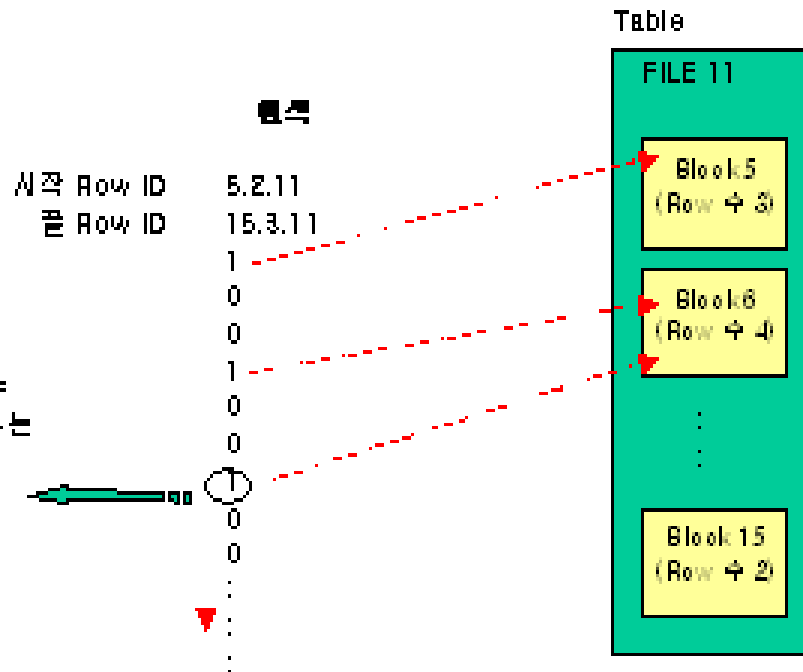
- BitMap 인덱스의 매카니즘

```
SELECT A, B, C
FROM 검색 대상 테이블
WHERE 인덱스 컬럼 = '흰색'
```

Row ID 계산법 = 시작 Row ID + 상대거리

예) 흰색 7번째의 Row ID는 이전 블록까지의 Row 수의 합인 3이다. 현재 블록에서의 Row 순서 4를 더하는 같은 7이 되므로 해당 값의 위치를 알 수 있다.

주의) 블록의 Row 수는 데이터 사전에서 관리하므로 해당 Row가 나올 때 까지 이전 블록의 모든 Row를 읽을 필요는 없다.

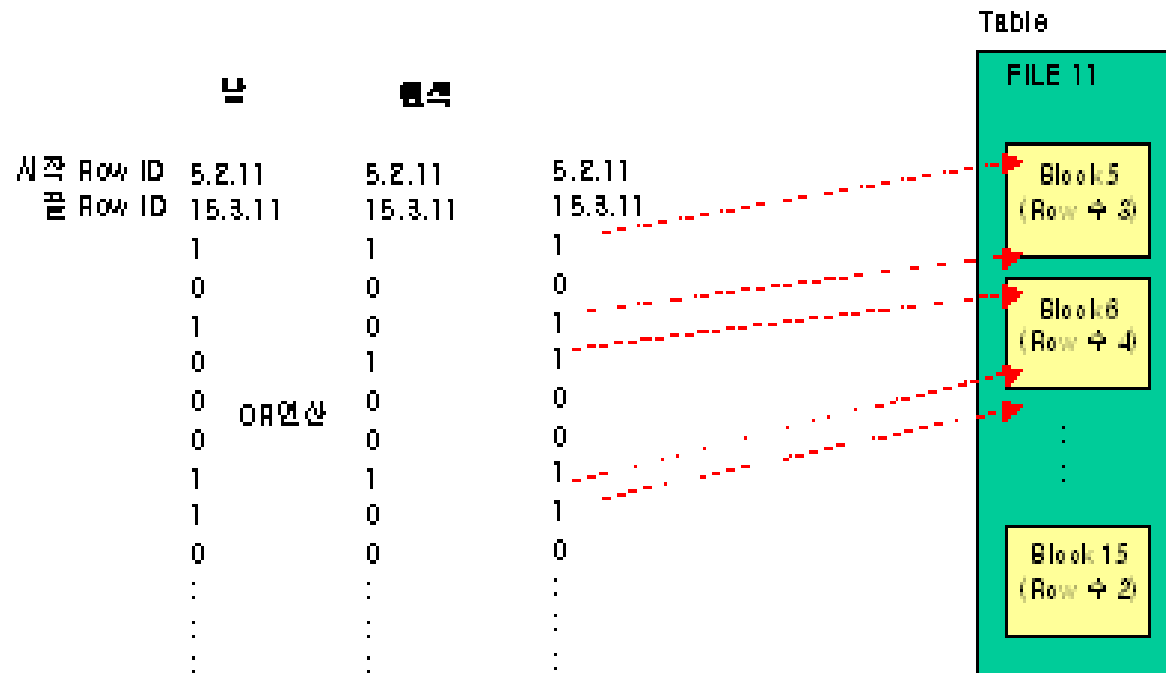


9. 접근방법 및 분산설계

BitMap 인덱스

- BitMap 인덱스의 매카니즘

```
SELECT A, B, C
FROM 검색 대상 테이블
WHERE 인덱스 컬럼1 = '흰색'
OR 인덱스 컬럼2 = '남'
```





9. 접근방법 및 분산설계

BitMap 인덱스

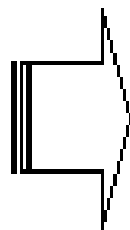
- BitMap 추가/수정/삭제 매카니즘

원색 중간색 권점색

·
·
·
1
0
0
0
1
0
0
1
0
0
·
·
·

·
·
·
0
1
1
0
0
0
0
0
0
·
·
·

·
·
·
0
0
0
1
1
0
1
·
·
·



100,000번
째 레코드에
권점색 정보
레코드 추가

원색 중간색 권점색

·
·
·
1
0
0
0
1
0
0
1
0
0
·
·
·

·
·
·
0
1
1
0
0
0
0
0
0
·
·
·

·
·
·
0
0
0
1
1
0
1
1
·
·
·

9. 접근방법 및 분산설계

B*Tree 와 BitMap 비교

- **인덱스의 검색 속도**
 - B*Tree 인덱스 : Tree 알고리즘에 의해 필요로 하는 인덱스 값만 읽을 수 있다. 그러나, 인덱스 크기가 BitMap 인덱스보다 매우 크다. 크기에서 백배 이상 차이가 날 수 있다.
 - BitMap 인덱스 : 항상 전체 인덱스를 읽어야 하나 인덱스 크기가 매우 작다.
 - 결론적으로 소량의 데이터를 읽을 때는 B*Tree 인덱스가 대량의 데이터를 읽을 때는 BitMap 인덱스가 유리하다.
- **인덱스의 변경 관리성**
 - B*Tree인덱스 : Tree알고리즘에 의해 손쉽게 인덱스 CUD가 가능하다.
 - BitMap 인덱스 : CUD는 인덱스 재생성과 별반 다를 게 없다.

9. 접근방법 및 분산설계

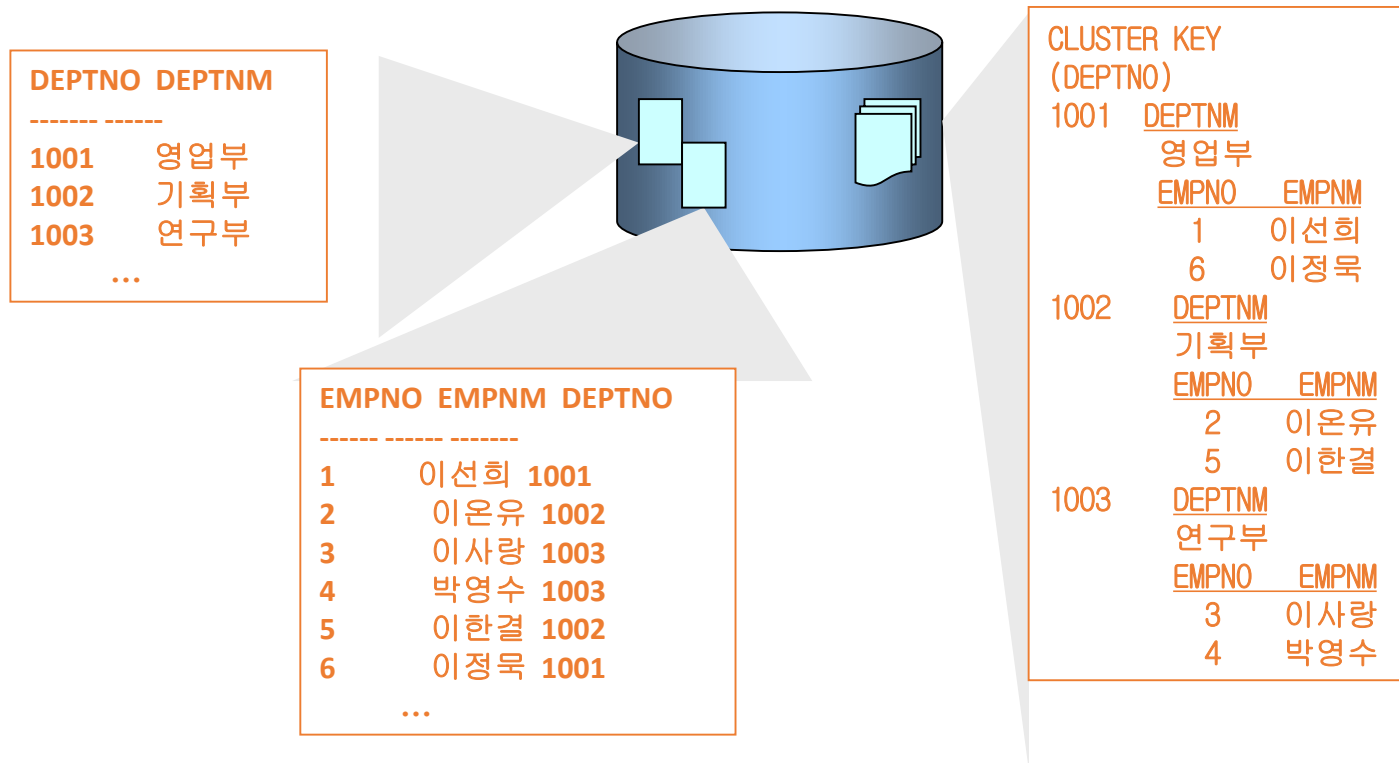
B*Tree 와 BitMap 비교

- 필요한 스토리지
 - B*Tree 인덱스 : 인덱스 트리를 관리하기 위해 많은 스토리지가 요구된다.
 - BitMap인덱스 : Bit연산으로 관리하므로 작은 스토리지가 요구된다.
- 연산 능력
 - B*Tree 인덱스 : And 연산에는 좋은 성능을 나타내나 or, != 등에서 성능에 취약하다.
 - BitMap 인덱스 : Bit 연산을 통해 빠르게 처리할 수 있다.
 - 결론적으로 OLTP는 주로 소량의 정보를 읽고, 또 빈번한 CUD가 발생하고, DSS는 대량의 정보를 읽고 CUD가 거의 전무하므로 OLTP는 B*Tree 가 DSS는 BitMap이 유용하게 사용된다.

9. 접근방법 및 분산설계

클러스터링

● 클러스터링



클러스터링이 안된 테이블

클러스터링이 된 테이블



9. 접근방법 및 분산설계

클러스터링

- 클러스터링 인덱스

클러스터드 인덱스

CLUSTER KEY	CLUSTER HEADER
1001	1
1002	10
1003	20

클러스터드 테이블

CLUSTER KEY (DEPTNO)		DEPTNM	
1	1001	영업부	
		EMPNO	EMPNM
		1	이선희
		6	이정묵
10	1002	기획부	
		EMPNO	EMPNM
		2	이온유
		5	이한결
...			

분산 설계

- 분산 설계 위치





10. 분산 설계

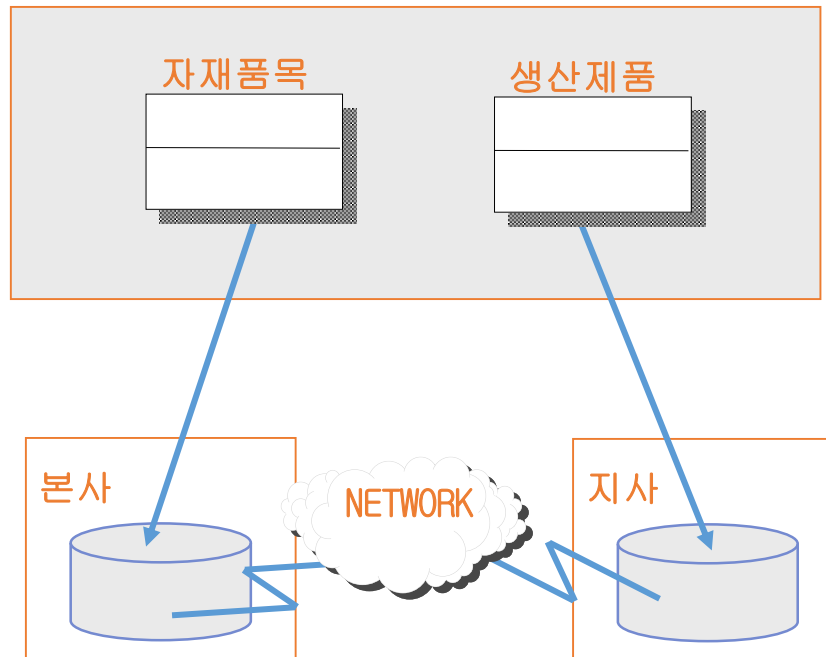
분산 설계 정의

- 한군데 데이터베이스를 집중시켜 관리하던 중앙 집중식 시스템 환경구축에서 빨라진 네트워크(NETWORK)를 이용하여 데이터베이스를 여러 서버나 지역에 분산시켜 놓고 하나의 데이터베이스처럼 사용할 수 있게 하는 설계가 분산 설계이다.

10. 분산 설계

테이블별 위치 분산

- 테이블별 위치 분산

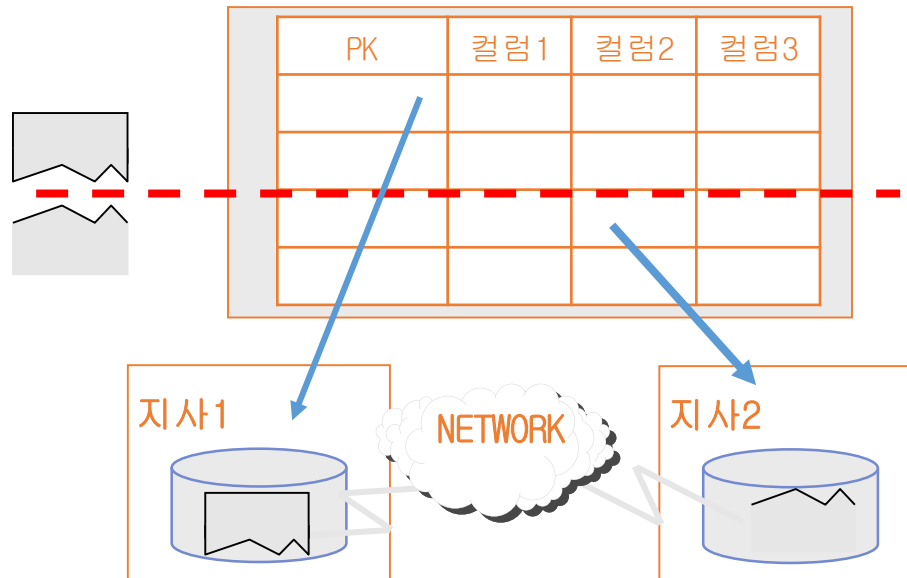


테이블 위치	자재품목	생산제품	협력회사	사원	부서
본사	●		●		●
지사		●		●	

10. 분산 설계

테이블별 위치 분산

- 수평 분할 분산 - 컬럼의 값을 기준으로 로우(ROW) 분리

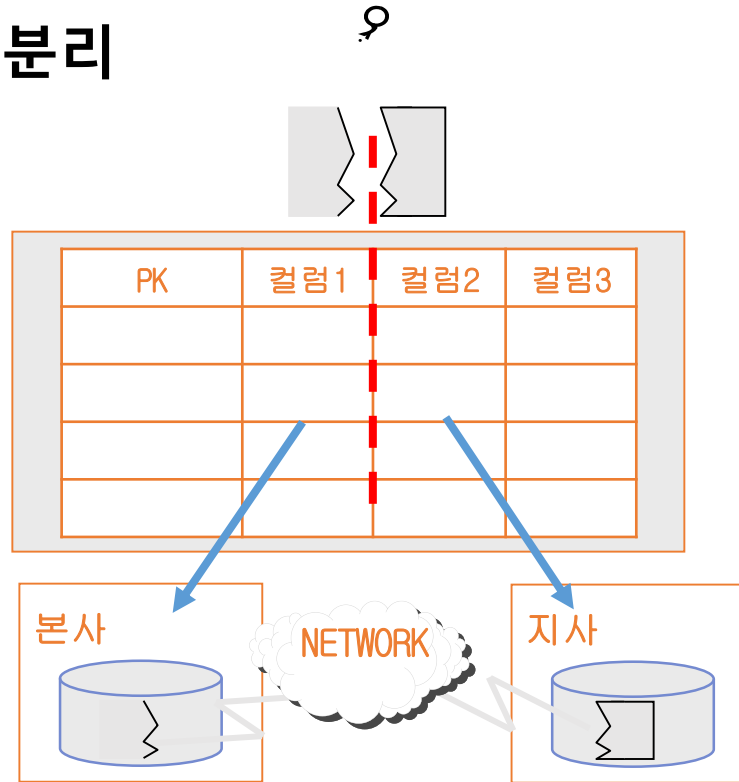


테이블 위치	고객	생산제품	협력회사	사원	부서
지사1	●	●	●	●	●
지사2	●	●	●	●	●

10. 분산 설계

테이블별 위치 분산

- 수직 분할 분산 - 컬럼을 분리

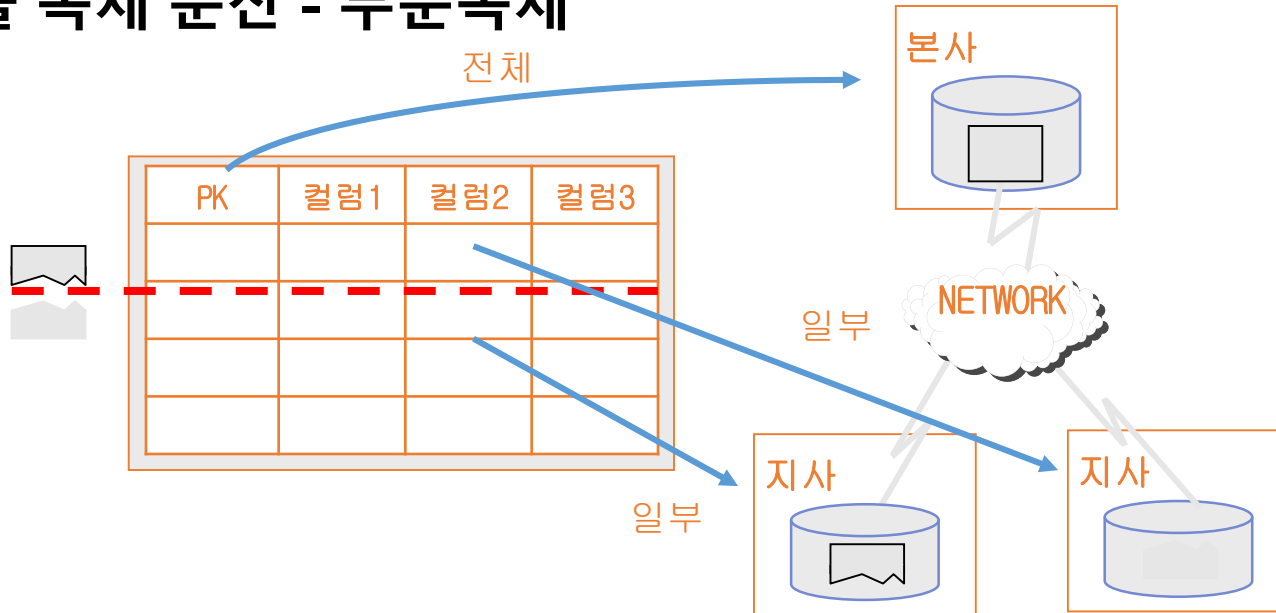


테이블 위치	제품	분할컬럼
본사	●	제품번호, 단가
지사	●	제품번호, 재고량

10. 분산 설계

테이블 복제 분산 - 부분복제

● 테이블 복제 분산 - 부분복제



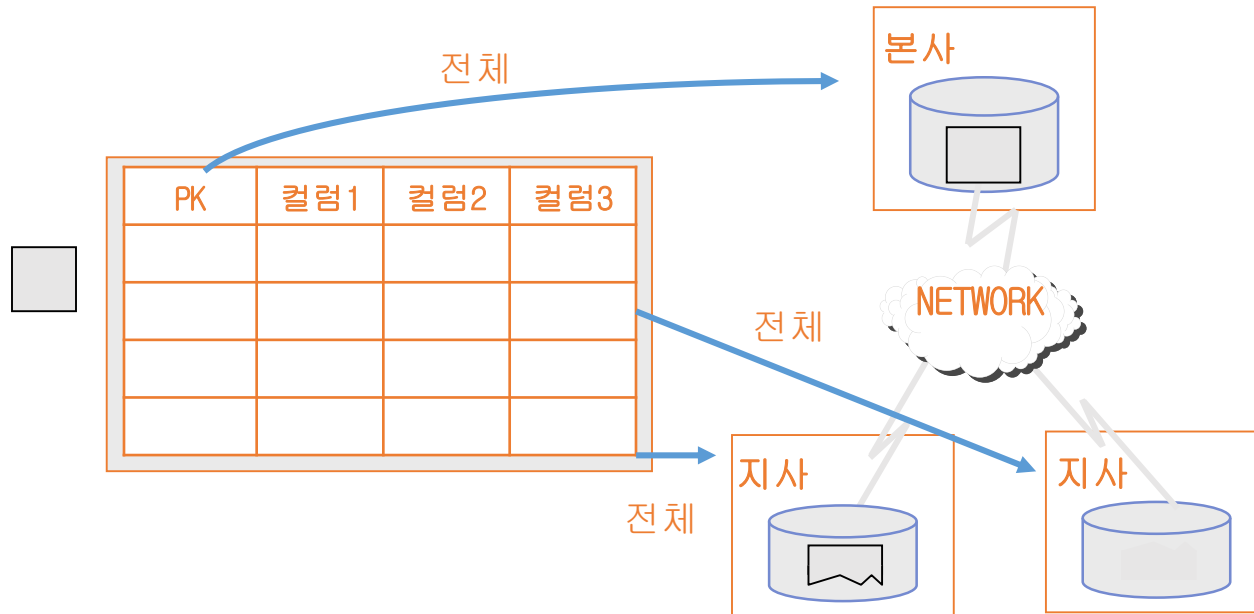
테이블 위치	고객
본사	●
지사1	○
지사2	○

본사에는 전국의 고객정보를 관리하고 지사 1의 고객테이블에서는 지사1에 속한 고객정보를 지사2의 고객테이블에서는 지사2에 속한 고객정보를 관리한다.

10. 분산 설계

테이블 복제 분산 - 광역복제

- 테이블 복제 분산 - 광역복제



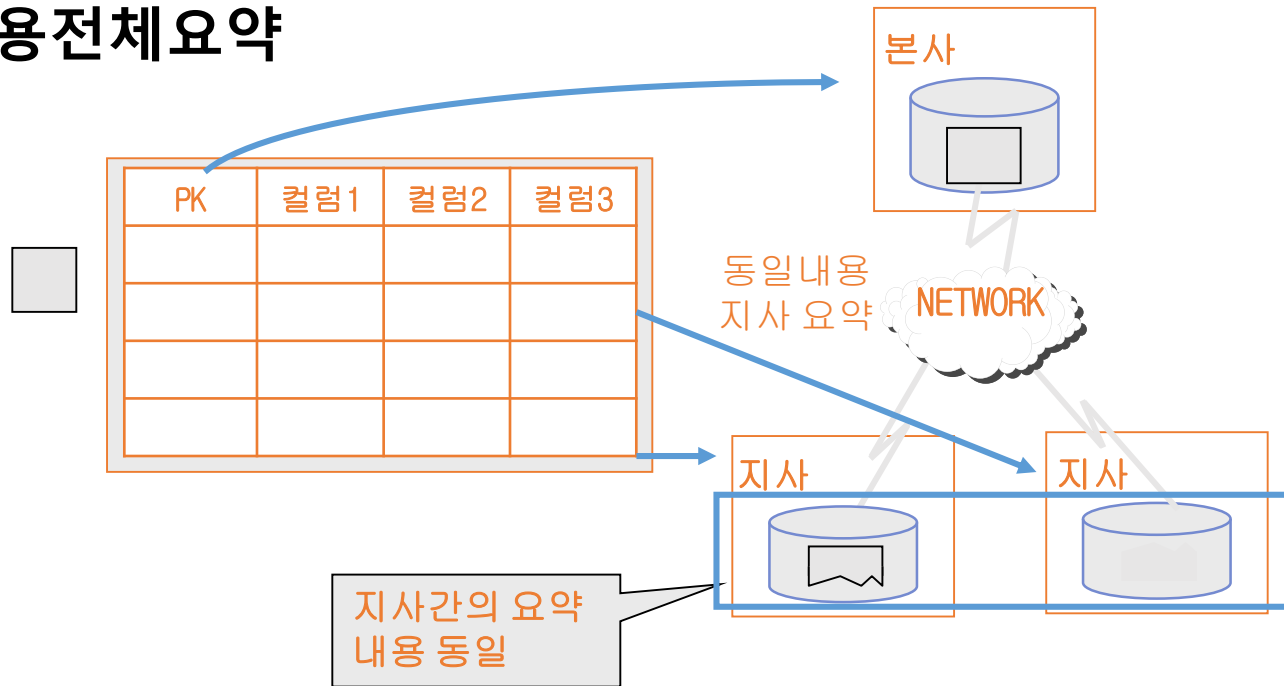
테이블 위치	코드
본사	●
지사1	●
지사2	●

본사, 지사1, 지사2 모두 동일한 양의 코드테이블의 데이터를 가지고 있다.

10. 분산 설계

테이블 요약 분산 - 분석요약

- 동일내용전체요약



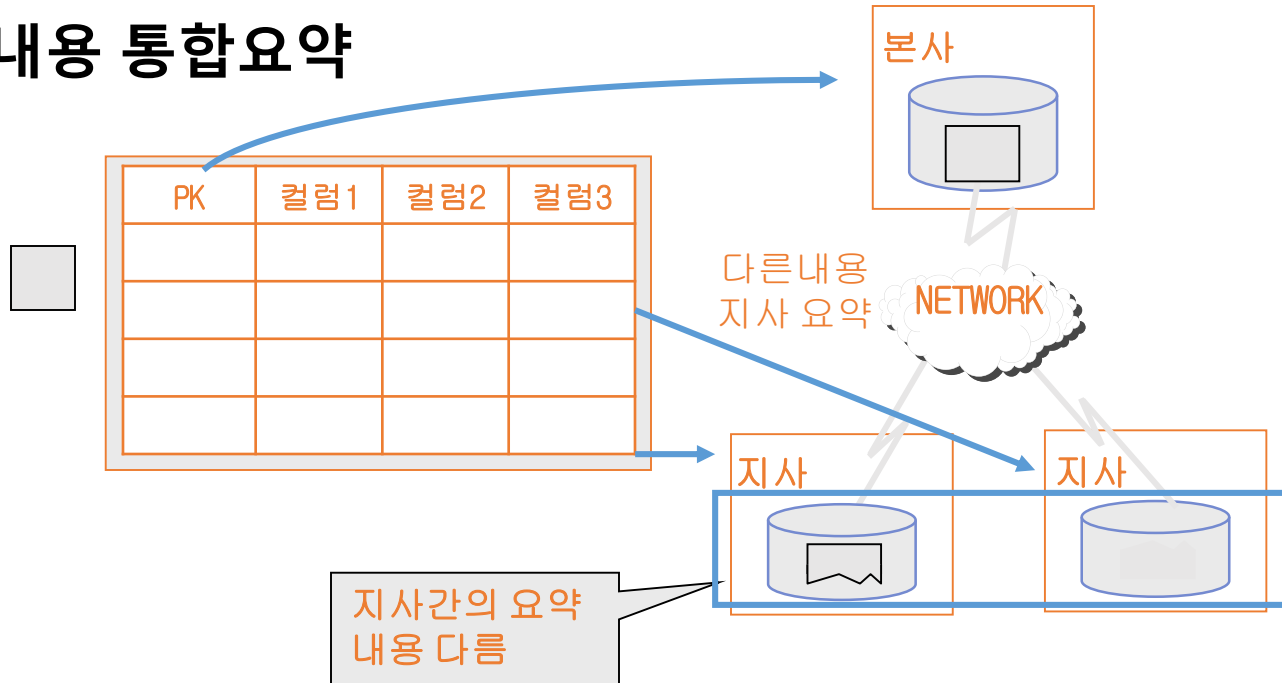
테이블 위치	판매실적
본사	●
지사1	●
지사2	●

지사1에서는 지사1의 판매실적이 있고 지사2에서는 지사2의 판매실적이 존재한다. 본사에서는 모든 지사의 판매실적을 통합한 실적 데이터가 생성된다.

10. 분산 설계

테이블 요약 분산 - 통합요약

- 동일내용 통합요약



테이블 위치	판매실적
본사	●
지사1	●
지사2	●

지사1과 지사2에 판매실적이 존재하지만 서로 다른 내용으로 존재한다. 본사에서는 모든 지사의 판매실적을 통합한 실적 데이터가 생성된다.

Unit A

참고자료



문헌

1. <http://www.ncs.go.kr>
2. NELDALE/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