



데이터 모델링

4장. 물리 데이터 모델링

1절. 물리 데이터 모델링 이해

2절. 물리 요소 조사 및 분석

3절. 논리-물리 데이터 모델 변환

4절. 반정규화

1-1 물리 데이터 모델링 정의

- ❖ 물리 데이터 모델이란 논리적 모델을 특정 데이터베이스로 설계함으로써 생성된, 데이터를 저장할 수 있는 물리적인 스키마를 말한다.
 - ✓ 논리 데이터 모델을 사용하고자 하는 각 DBMS의 특성을 고려하여 데이터 베이스 저장 구조(물리 데이터 모델)로 변환하는 것이다.
- ❖ 물리 데이터 모델링과 데이터베이스 디자인
 - ✓ 물리 데이터 모델링은 데이터의 구조에 관련된 것들을 물리적인 모습까지 설계하는 것
 - ✓ 데이터베이스 디자인은 이러한 물리적인 모델(설계도면)을 DBMS 관점의 오브젝트로 생성하는 최적의 설계(디자인)를 하는 것(예 : 저장공간의 효율적 사용 계획, 오브젝트 파티셔닝 설계, 최적의 인덱스 설계)

1-2 물리 데이터 모델 의의

- ❖ 물리적 데이터 모델링은 관계 데이터 모델링(RDM, Relation Data Modeling)이라고도 하며, 논리적 데이터 모델을 각각의 관계형 데이터베이스 관리 시스템의 특성, 기능, 성능 등을 고려하여 데이터베이스의 물리적인 구조(Schema)를 작성해가는 과정이다.
- ❖ 물리적 데이터 모델링 단계는 논리 데이터 모델에서 도출된 내용 변환을 포함하여 데이터의 저장공간, 데이터의 분산, 데이터 저장 방법 등을 고려한다. 이 과정에서 데이터베이스 운용 성능(Performance)에 결정적인 영향을 끼친다.

2-1 시스템 구축 관련 명명 규칙

- ❖ 사내의 시스템 구축과 관련된 명명 규칙을 파악하여 물리 데이터 모델의 각 요소의 내용에 이를 적용해야 한다

2-2 하드웨어 자원 파악

- ❖ **CPU**
 - ✓ 중앙처리 장치의 성능과 집중적인 부하가 발생하는 시간 등을 파악
- ❖ **MEMORY**
 - ✓ 전체 메모리의 규모 및 시스템이 사용하는 메모리 영역을 포함하여 사용 가능한 메모리 영역을 파악
- ❖ **DISK**
 - ✓ 전체 디스크의 크기, 분할된 형태, 현재 디스크 활용률 등을 파악하고 사용 가능한 공간을 확인
- ❖ **I/O Controller**
 - ✓ 현재 입/출력 컨트롤러의 성능 및 적절하게 운용되고 있는가를 파악
- ❖ **Network**
 - ✓ 현재 처리 가능한 속도, 집중적인 부하가 발생하는 시간대, 동시 접속 최대 가용 사이트 수를 파악

2-3 운영체제 및 DBMS 버전 파악

- ❖ 데이터베이스 운영 환경과 관련된 운영체제의 관련 요소를 파악하고 적절하게 관리되고 있는가를 파악한다. 특히 인스턴스 관리 기법 등에 대해서 파악하고 분석한다.

2-4 DBMS 파라미터 정보 파악

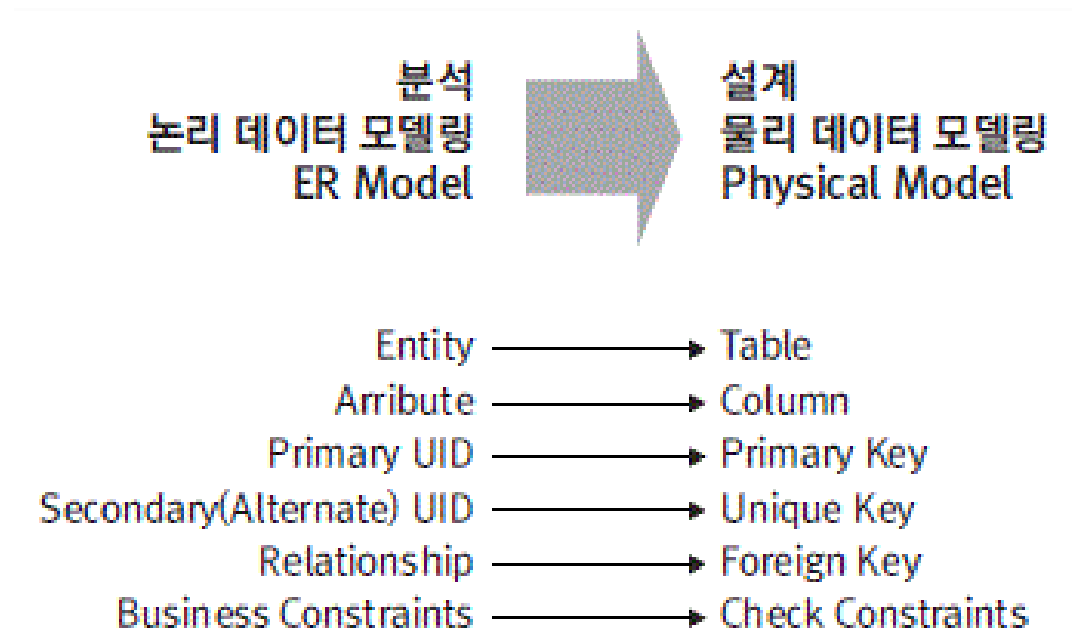
- ❖ DBMS 환경 적용 단계에서 가장 중요하게 고려해야 하는 단계이며, DBMS 파라미터는 데이터베이스 관리 시스템별로 많은 차이가 있으며 관리하는 방법도 서로 다르다.
- ❖ 데이터 저장 공간 관리 기법, 메모리 관리 기법, 데이터 쿼리에서 활용하는 옵티마이저 (Optimizer)의 운영 방법 등도 중요한 고려사항이 된다.

2-5 데이터베이스 운영과 관련된 관리 요소 파악

- ❖ 사용자 관리 기법 및 정책
- ❖ 백업/복구 기법 및 정책
- ❖ 보안 관리 정책

1 단계	엔티티-테이블 변환 <ul style="list-style-type: none">- 단위 엔티티를 테이블로 변환- 서브타입 변환
2 단계	속성-칼럼 변환 <ul style="list-style-type: none">- 일반 속성 변환- Primary UID 기본키(Primary Key) 변환- Primary UID(관계의 UID Bar) 기본키(Primary Key) 변환
3 단계	관계 변환 <ul style="list-style-type: none">- 1:M 관계 변환- 1:1 관계 변환- 1:M 순환 관계 변환
4 단계	배타적 관계 변환
5 단계	데이터 타입 선택

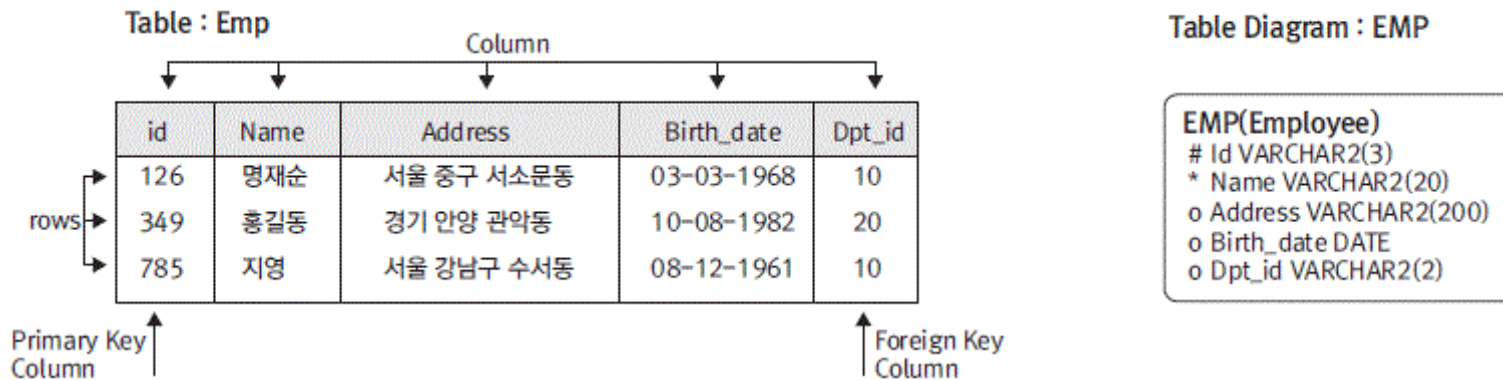
3-1 논리-물리 데이터 모델 변환(Transformation) 용어



3-2 엔티티-테이블 변환

❖ 단위 엔티티를 테이블로 변환

테이블은 데이터를 저장하기 위해서 생성된 데이터베이스에서의 가장 기본적인 오브젝트



✓ 테이블(Table)

테이블은 칼럼(Column)과 로우(Row)를 가진다. 칼럼은 지정된 유형의 데이터 값을 저장하는 데 사용된다.

✓ 로우(Rows)

테이블의 한 로우에 대응. 튜플, 인스턴스, 어커런스라고도 한다.

✓ 칼럼(Columns)

각 사원 개개인의 관리 항목에 대한 Value를 저장한다.

✓ 기본키(Primary keys)

하나의 칼럼 혹은 몇 개의 칼럼 조합으로 테이블 내에 동일한 값을 갖는 튜플이 존재하지 않도록 한다.

✓ 외래키(Foreign keys)

외부 데이터 집합과의 관계를 구현한 구조이다.

3-3 속성-칼럼 변환

❖ 일반 속성 변환

- ✓ 엔티티에 있는 각 속성들에 대한 칼럼명을 사례 데이터 표의 칼럼명 란에 기록한다
- ✓ 칼럼의 명칭은 속성의 명칭과 반드시 일치할 필요는 없으나 프로그래머와 사용자의 혼동을 피하기 위해 가능한 표준화된 약어를 사용한다
- ✓ 표준화된 약어의 사용은 SQL 해독 시간을 감소시킨다.
- ✓ SQL의 예약어(reserved word)의 사용을 피한다.
- ✓ 가급적 칼럼 명칭은 짧은 것이 좋으며, 짧은 명칭은 개발자의 생산성에 긍정적인 영향을 준다.
- ✓ 필수 입력 속성은 Nulls/Unique 란에 NN을 표시한다.
- ✓ 실제 테이블에 대한 설계를 검증하기 위한 목적으로 가능하다면 표본 데이터를 입력시킨다.

3-3 속성-칼럼 변환

❖ Primary UID 기본키(Primary Key) 변환

- ✓ 논리 데이터 모델에서의 Primary UID는 물리 데이터 모델에서는 기본키로 생성된다.
- ✓ 실제 DDL에서는 기본키 제약 조건의 형태로 오브젝트가 생성된다.

1) 변환 절차

- 사례 데이터 표의 키 형태란에 엔티티의 Primary UID에 속하는 모든 속성에 PK를 표시
- PK로 표시된 모든 칼럼은 Nulls/Unique란에 반드시 NN,U로 표시되어야 함
- 여러 개의 칼럼으로 UID가 구성되어 있는 경우는 각각의 칼럼에 NN,U1을 표시
- 또 다른 Unique Key(Secondary UID)가 있다면 U2로 표시

2) 변환 예

칼럼명	EMPNO	ENAME	JOB
키 형태	PK		
Nulls/ Unique	NN,U	NN	
건본 데이터	1354	홍길동	부장
	4562	박경수	사원
	2763	김정구	
	2874	신동파	차장
	6538	정명훈	

3-3 속성-칼럼 변환

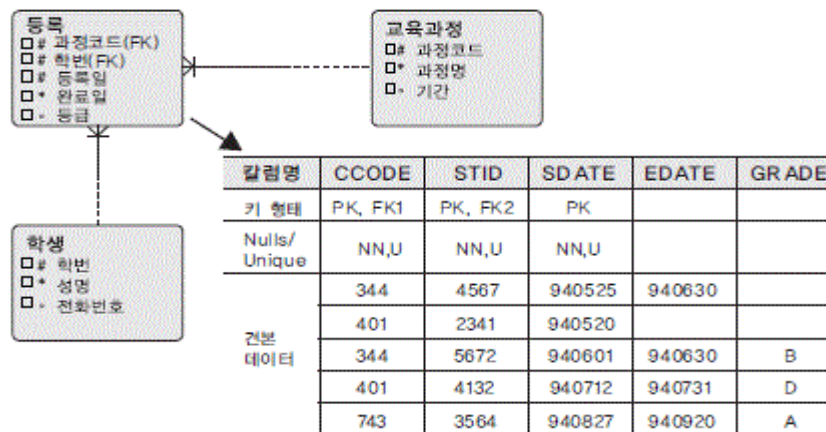
❖ Primary UID(관계의 UID Bar) 기본키(Primary Key) 변환

✓ 다른 집합(엔티티)으로부터의 관계에 의해서 생성되는 UID 속성(관계 속성)도 존재 함

1) 변환 절차

- PK의 일부분으로 표시
 - Nulls/Unique 란에 각각 NN,U1을 표시
 - 키 형태란에 PK, FK를 표시
 - 여러 UID BAR가 있는 경우는 (PK, FK1), (PK, FK2), ...
 - 여러 칼럼으로 구성된 경우 PK,FK1을 각각 표시
- 추가된 FK 칼럼에 표본 데이터를 추가

2) 변환 예



3-4 관계 변환

❖ 1:M 관계 변환

- ✓ 논리 데이터 모델에서 존재하는 관계 중에서 가장 많은 형태의 관계다. M쪽 관계의 형태에 따라서 관계 칼럼의 선택사항이 결정되게 된다.

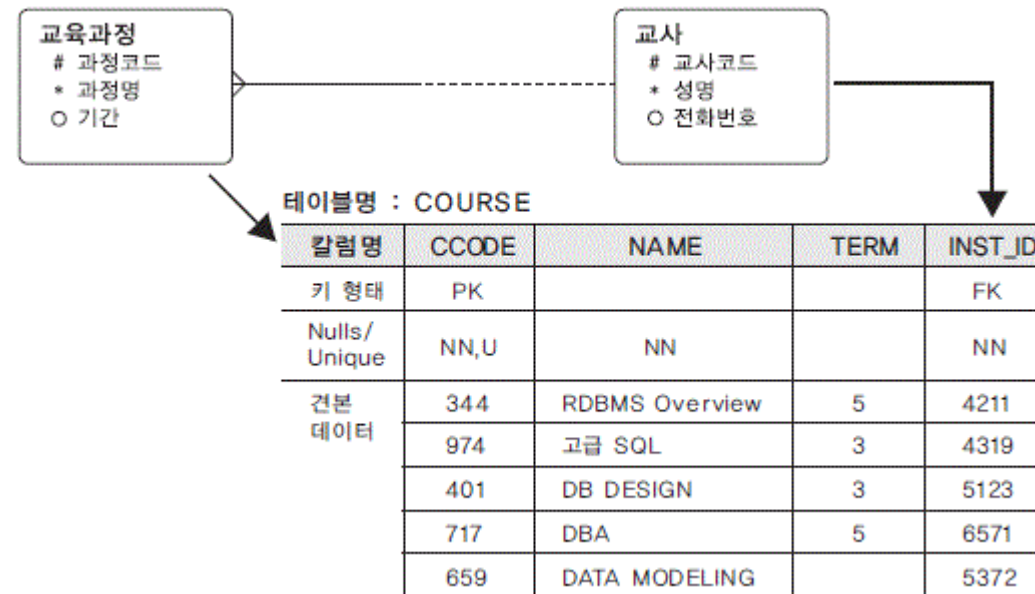
1) 변환 절차

- 1(One) 에 있는 PK를 M(Many)의 FK로 변환
 - FK의 명칭 결정
 - 키 형태 란에 FK 표시
 - Nulls/Unique 란에 NN 표시(Must Be 관계시)
 - 필수 관계가 아닌 경우에는 NN를 체크하지 않는다.
- 표본 데이터 추가
- UID BAR 가 있는 경우는 전단계에서 실시

3-4 관계 변환

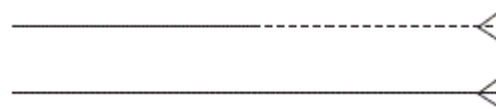
❖ 1:M 관계 변환

2) 변환 예



3) 1:M 관계에서 1 쪽이 Mandatory 관계일 때의 변환시 주의사항

- 자식 쪽의 레코드(Row)가 반드시 하나 이상은 되어야만 부모 쪽의 레코드(Row)를 생성할 수 있다.
- 자식 쪽의 레코드(Row)를 삭제할 경우에는 전체를 다 삭제할 수는 없고 반드시 하나 이상의 자식 레코드 (Row)를 남겨두어야 한다. 또는 자식, 부모 레코드(Row)를 동시에 삭제해야 한다.



3-4 관계 변환

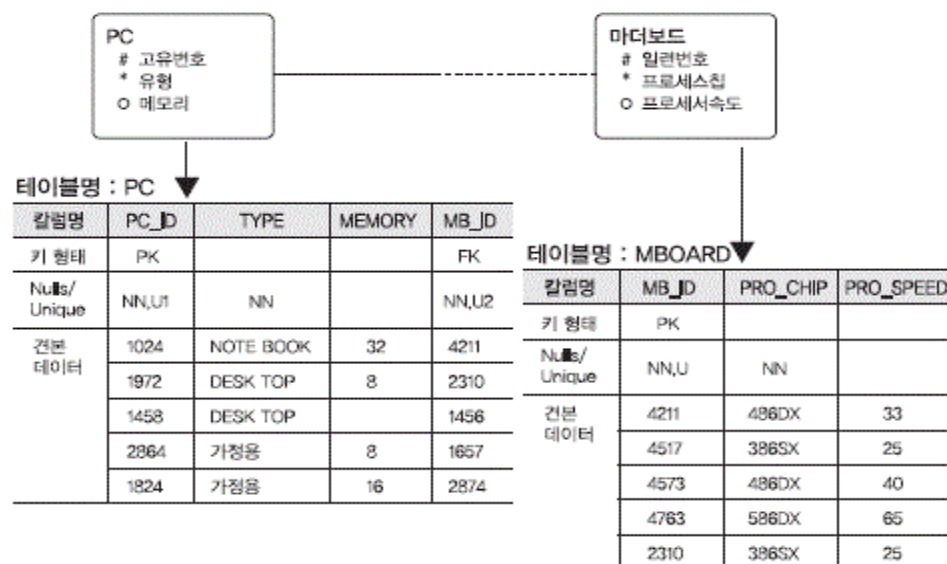
❖ 1:1 관계 변환

- ✓ 논리 모델에서는 자주 발생하지는 않는 관계
- ✓ 물리 모델로 변환하는 과정은 Optionality(기수성)에 따라서 다른 방법으로 적용

1) 변환 절차

- 한쪽이 Optional이고 다른 한쪽이 Mandatory라면 Mandatory쪽의 테이블에 외래키가 생성 된다.
- NN 표시를 한다.

2) 변환 예

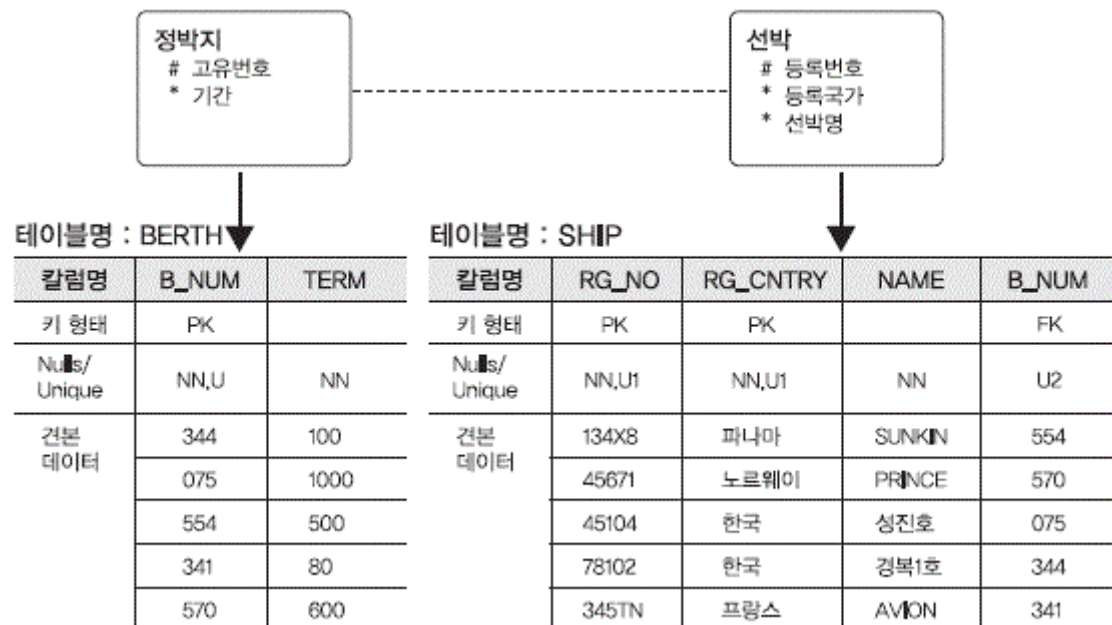


3-4 관계 변환

❖ 1:1 관계 변환

3) 변환 시 주의사항

- 양쪽 다 Optional인 경우에는 보다 빈번하게 사용되는 테이블이 외래키를 가지는 것이 유리
- 양쪽다 Mandatory라면 변환시에 어떤 테이블에 외래키를 생성할 것인지를 선택해야 한다.
- 1:1 관계에 의해서 생긴 모든 외래키 부분은 Unique Key가 필수적이다.



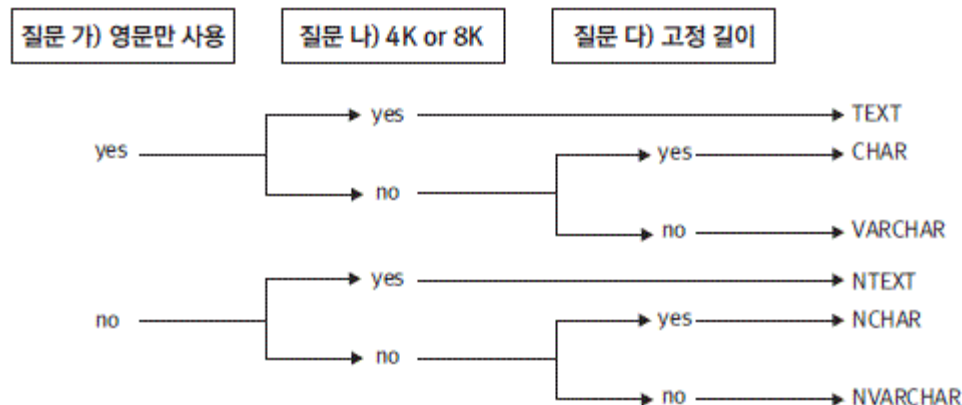
3-5 데이터 타입 선택

❖ 개념

- ✓ 논리 데이터 모델에서 정의된 논리적인 데이터 타입(정보 타입 : Information Type)을 물리적인 DBMS의 특성과 성능 등을 고려하여 최적의 데이터 타입을 선택

❖ 문자 타입 (Character Data Types)

- ✓ 세부 문자 타입 선택을 위한 기준
 - 영문만 사용되는가?
 - 4000자 혹은 8000자 이상의 문자열이 포함되는가?
 - 입력되는 값의 길이가 일정한가?
- ✓ 문자 형식 데이터 타입 설정 예



3-5 데이터 타입 선택

❖ 숫자 타입 (Numeric Types)

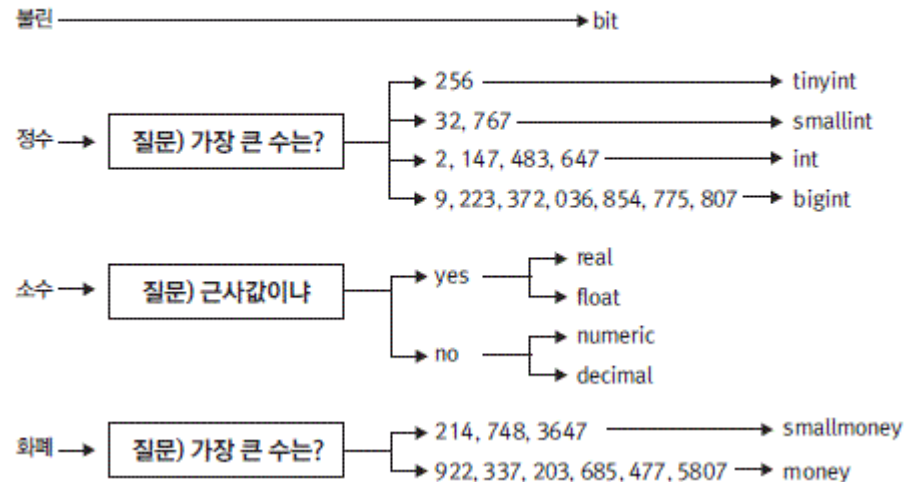
✓ 세부 숫자 타입 선택을 위한 기준

- 정말 숫자 데이터인지 판단한다.
- 세부 숫자 타입 결정

✓ 숫자 형식 데이터 타입 설정 예

- 불린(Boolean) : 참(TRUE) 혹은 거짓(False)을 저장하는 경우에 선택한다.
- 정수(Integer) : 소수점 이하를 처리하지 않는 경우에 선택한다.
- 소수(Decimal) : 소수점 이하를 처리하는 경우에 선택한다.
- 화폐(Money) : 금액을 저장하기 위한 경우에 선택한다.

질문) 세부 숫자 타입



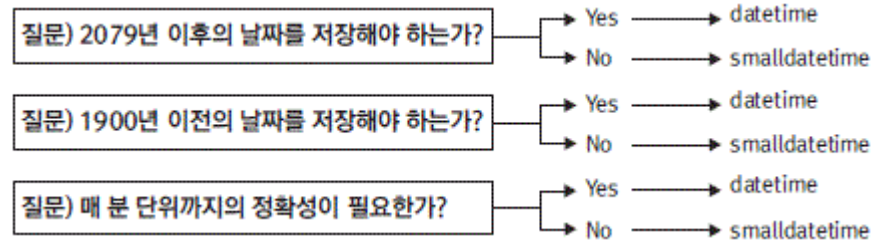
3-5 데이터 타입 선택

❖ 날짜 타입 (Datetime Types)

✓ 세부 날짜 타입 선택을 위한 기준

- 날짜 타입 또는 문자 타입으로 할 것인지는 이미 논리 데이터 모델에서 결정됨
- 일반적인 시간까지를 저장할 것이냐, 아니면 정밀한 시간을 저장할 것이냐에 따라 결정함

✓ 세부 날짜 타입 지정 예



3-6 데이터 표준 적용

❖ 개념

- ✓ 전사적으로 미리 생성된 데이터 표준을 적용하여 논리 데이터 모델링 과정에서 정의된 엔티티, 속성, 관계들을 물리 데이터 모델로 변환함
- ✓ 예) 표준 단어, 표준 용어, 표준 도메인

❖ 데이터 표준 적용 대상

- ✓ 데이터베이스(Database)
- ✓ 스토리지 그룹(Storage Group)
- ✓ 테이블스페이스(Tablespace)
- ✓ 테이블(Table)
- ✓ 칼럼(Column)
- ✓ 인덱스(Index)
- ✓ 뷰(View)

3-6 데이터 표준 적용

❖ 칼럼(Column) 명명규칙

- ✓ 고객 내부에서 통용되는 약어가 있다면 그대로 표현한다.
- ✓ 가급적 4자리 이내로 정의하며, 한글발음을 영역하지 말고 영문의미의 약자 형태로 표현한다.
- ✓ 가급적 영어 자음위주로 구성하며, 필요한 경우에는 모음도 사용할 수 있다.
 - 예) 상세 Detail → DTL, 제품 Product → PRD, 합계 Total → TOT
- ✓ 자음이 중복되는 경우 중복 자음 중 하나만 사용한다
- ✓ 영문의미가 3개 이상의 단어로 구성된 경우는 각 단어의 첫 자를 취한다
 - 예) 행동평점시스템 Behavior Scoring System → BSS
- ✓ 단어영문명의 중복이 발생한 경우 중복 제거 작성 우선 순위는 아래와 같다.
 - 중복제거 1순위: 영문의미를 다른 영문 단어로 변경하여 정의한다.
 - 중복제거 2순위: 영문의미가 전달될 수 있는 모음을 포함하여 정의한다.
 - 중복제거 3순위: 단어영문명의 길이를 4자로 정의한다.
- ✓ 접사(접두사,접미사)가 포함된 표준단어의 경우에는 공백이나 특수문자의 사용을 배제하여, 인식하기 쉬운 글자로 구성한다.
 - 예) Unit Price(단가) → UPRC(O), UN_PR(X)
- ✓ 숫자가 포함된 경우는 단어영문명+숫자 형태로 구성한다.
 - 예) 1주 → W1, 1회차 → SEQ1

4-1 반정규화(Denormalization) 개요

- ❖ 정규화 작업이 완료되면 데이터 모델은 데이터의 중복을 최소화하고 데이터의 일관성과 정확성, 안정성을 보장하는 데이터 구조가 완성됨
- ❖ 정규화된 데이터 모델은 시스템의 성능 향상, 개발 과정의 편의성, 운영의 단순화를 위해 정규화의 원칙들에 위배되는 행위를 의도적으로 수행하게 되는데, 이를 반정규화 과정이라 함
 - ✓ 논리 데이터 모델이 확정된 후, 물리 데이터 모델링 과정에서 여러 가지 효과를 얻고자 하는 것
 - ✓ 논리 데이터 모델을 정의하는 과정에서 반정규화부터 고려하게 되면 데이터 모델이 복잡해지고, 오류 가능성도 커질 수 있음
- ❖ 반정규화는 테이블 관점, 칼럼 관점으로 수행할 수 있고, 추가적인 분할이나 데이터 중복 등이 주된 방법
 - ✓ 장점 : 성능과 관리효율을 증대시킬 수 있음
 - ✓ 단점 : 데이터의 일관성 및 정합성을 해칠 위험을 내포하고 있고, 이를 유지하는 비용이 발생하여 지나치면 성능에도 악영향을 미칠 수 있음
- ❖ 데이터 모델의 일관성과 무결성을 우선으로 할지, 데이터베이스의 성능과 단순화에 우선 순위를 둘 것인지를 적절하게 조정하는 것이 중요
 - ✓ 모델러 또는 설계자가 단독으로 결정하기 보다는 관련자들과 협의를 통해 도출
 - ✓ 모델러 또는 설계자의 다양한 경험이 필요

4-1 반정규화(Denormalization) 개요

❖ 반정규화 절차

1. 반정규화 대상 조사

- 범위 처리 빈도수
- 대량의 범위 처리
- 통계성 프로세스
- 테이블 조인 개수

2. 다른 방법 유도 검토

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

3. 반정규화 적용

- 테이블의 반정규화
- 속성의 반정규화
- 관계의 반정규화

3-1. 테이블의 반정규화

- 테이블 분할
(수직/수평 분할)
- 테이블 추가
(중복, 통계, 이력, 부분)

3-2. 속성의 반정규화

- 중복 컬럼 추가
- 추출(Derived) 컬럼 추가
- 이력 테이블 컬럼 추가
- Primary Key에 의한 컬럼 추가
- 응용 시스템 오작동 처리를 위한 컬럼 추가

3-3. 관계의 반정규화

- 중복관계 추가

4-2 테이블 분할

❖ 수평 분할(Horizontal Partitioning)

✓ 개념

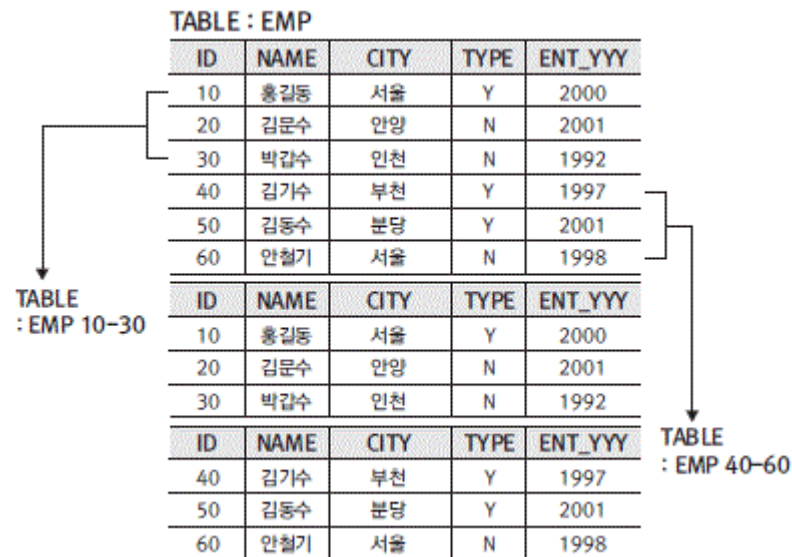
- 레코드 기준으로 테이블을 분할하는 것
- 하나의 트랜잭션이 분할된 여러 테이블을 동시에 처리하는 경우가 없을 것

✓ 사용 의의

- 하나의 테이블에 데이터가 너무 많이 있고, 레코드 중에서 특정한 덩어리의 범위만을 주로 액세스 하는 경우에 사용함
- 분할된 각 테이블은 서로 다른 디스크에 위치시켜 물리적인 디스크의 효율성을 극대화할 수 있음
- 수평 분할은 DBMS 차원에서 제공하는 방법으로 범위 분할, 해쉬 분할, 복합 분할 등의 기법이 사용

✓ 주의 사항

- UNION을 통한 처리나 여러 개의 SQL을 이용해야 하므로 수행속도가 늦어짐
- 뷰를 활용할 수 있으나 여러 테이블을 합친 뷰는 조회만 가능



4-2 테이블 분할

❖ 수직 분할 (Vertical Partitioning)

✓ 개념

- 하나의 테이블이 가지는 칼럼의 개수가 많아지기 때문에 테이블을 분할하는 것

✓ 수직 분할이 일어나는 이유

- 조회 위주의 칼럼과 갱신 위주의 칼럼으로 나뉘는 경우
- 특별히 자주 조회되는 칼럼이 있는 경우
- 특정 칼럼 크기가 아주 큰 경우
- 특정 칼럼에 보안을 적용해야 하는 경우

✓ 주의 사항

- 기본키는 모든 테이블에 넣고 일반 컬럼들은 각각 테이블에 하나씩만 존재시킬 것
- 각 테이블에는 모든 로우가 존재하도록 할 것(Outer Join 방지)
- 수량, 금액 등 Group by가 되는 컬럼들은 사용빈도가 많은 쪽 테이블에 위치시키는 것이 유리
- 분할된 테이블 간의 조인이 많이 발생하는지 검증할 것

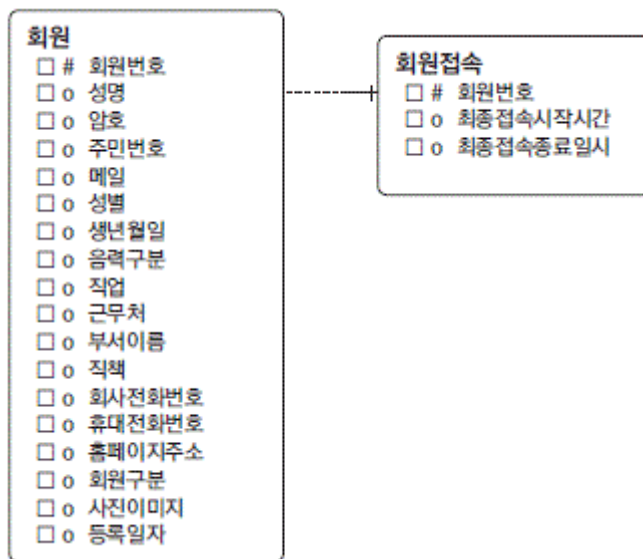
회원

- ☐ # 회원번호
- ☐ o 성명
- ☐ o 암호
- ☐ o 주민번호
- ☐ o 메일
- ☐ o 성별
- ☐ o 생년월일
- ☐ o 음력구분
- ☐ o 직업
- ☐ o 근무처
- ☐ o 부서이름
- ☐ o 직책
- ☐ o 회사전화번호
- ☐ o 휴대전화번호
- ☐ o 홈페이지주소
- ☐ o 최종접속시작시간
- ☐ o 최종접속종료일시
- ☐ o 회원구분
- ☐ o 사진이미지
- ☐ o 등록일자

4-2 테이블 분할

❖ 수직 분할 (Vertical Partitioning) - 갱신 위주의 칼럼 수직 분할

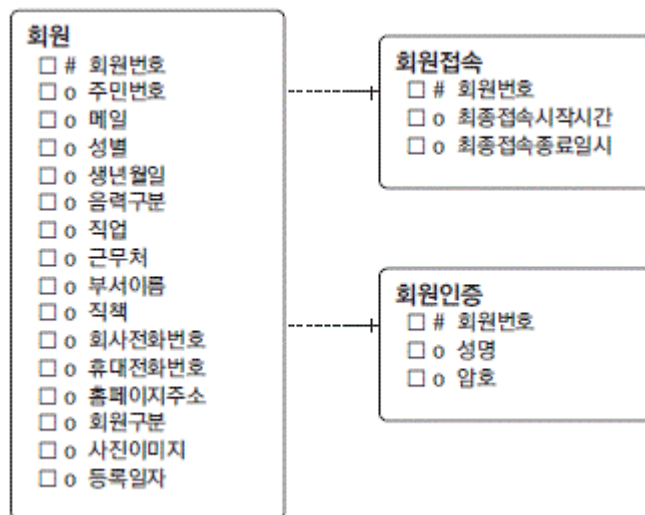
- ✓ 데이터 갱신 시 업데이트하려는 레코드에 잠금(Locking)을 수행하기 때문이며, 변경작업이 끝날 때까지 다른 프로세스가 이 데이터의 값을 변경하지 못하도록 금지하게 된다. 즉 몇 개의 칼럼에 대한 작업이 나머지 조회 위주의 칼럼 이용을 방해하기 때문에 이러한 갱신 위주의 칼럼들을 수직 분할하여 사용의 효율성을 증가시킨다.
- ✓ DBMS 버전별 적용 검토
 - DBMS에 따라 이러한 특성을 갖지 않는 경우도 있음



4-2 테이블 분할

❖ 수직 분할 (Vertical Partitioning) - 자주 조회되는 칼럼 분할

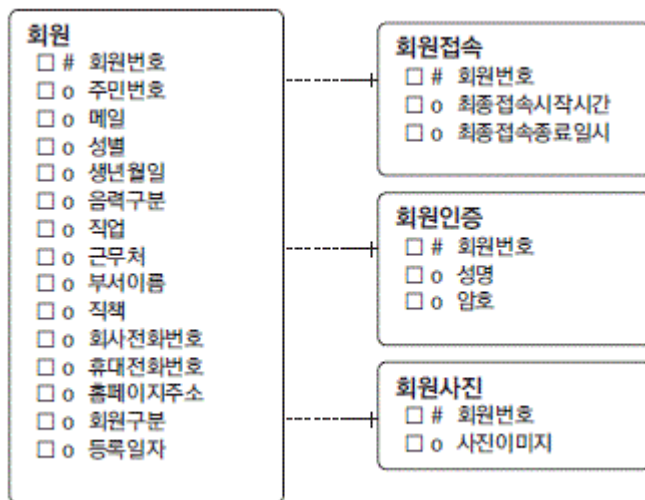
- ✓ 자주 조회되는 칼럼들을 분리해서 별도의 테이블로 관리하면 조회 쿼리의 작업 성능을 향상시킬 수 있다. 즉, 칼럼 수가 아주 많은 테이블에서 일부 칼럼들을 주로 사용한다면 이들을 별도로 분리함으로써 물리적인 I/O의 양을 줄여서 데이터 액세스 성능을 향상시킬 수 있다.
- ✓ 물리적인 DBMS 매커니즘 고려
 - 칼럼 수가 적은 테이블을 자주 읽는다면 초기 데이터를 메모리로 적재하는 비용이 절약되고, 또한 메모리상에 상대적으로 오래 머무를 수 있기 때문에 데이터의 재사용성을 높여주는 효과를 가져올 수 있음



4-2 테이블 분할

❖ 수직 분할 (Vertical Partitioning) - 특정 칼럼의 크기가 아주 큰 경우 분할

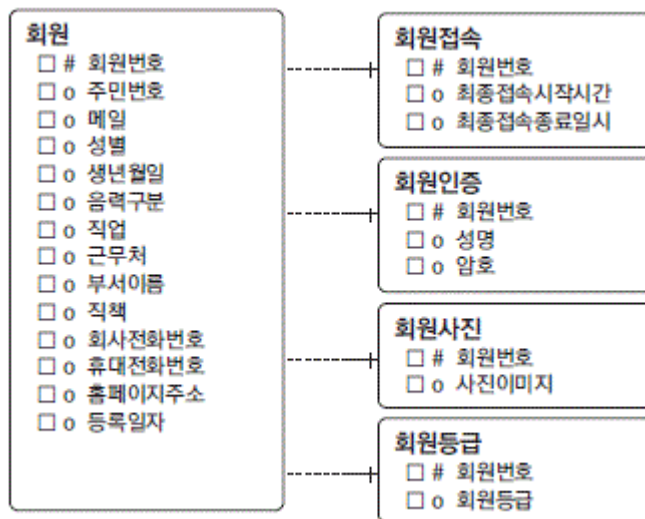
- ✓ 테이블의 칼럼에 텍스트 및 이미지와 같은 LOB(Large Objects) 데이터가 포함될 때 성능이 저하될 가능성이 있고, 백업과 복원 같은 관리나 프로그래밍 같은 개발 부분에서 성능 저하 요인으로 작용할 수 있기 때문에 이러한 데이터 형식들을 분리할 수 있다.
- 게시판 본문과 헤더(작성자, 작성일, 글제목 등) 정보가 한 레코드에 있게 되면 게시글 목록을 보기 위해 불필요한 본문 데이터까지 액세스해야 해서 I/O 비효율이 발생한다.
- LOB 데이터 형식을 지원하는 방법은 DBMS마다 다소 차이가 있지만, 이들이 I/O에 미치는 영향은 매우 크기 때문에 이들을 분리하는 방법을 사용하게 된다.



4-2 테이블 분할

❖ 수직 분할 (Vertical Partitioning) - 특정 칼럼에 보안을 적용해야 하는 경우의 분할

- ✓ 많은 DBMS들이 칼럼에 대한 권한 제어기능을 제공하지 않기 때문에, 보안 적용이 필요한 칼럼을 별도의 테이블로 만들어 권한을 제어하기 위한 목적으로 수직 분할을 할 수 있다.



4-2 테이블 분할

❖ 테이블 분할의 장단점

✓ 장점

- 스캔 처리 시 액세스 량이 감소
- 테이블 관리가 용이(Backup, Recovery, 재생성 등)
- 테이블의 Lock, Contention 감소
- 보안 유지가 용이

✓ 단점

- 테이블 구분 없이 전체적인 데이터를 처리해야 하는 경우 UNION이 발생
- 처리속도가 나빠지는 경우가 많다.
- 트랜잭션 처리 시 여러 테이블을 처리하는 경우가 증가
- 복잡한 처리의 SQL 통합이 어려워 진다.
- 부분범위 처리가 불가능해 질 수 있다.
- 여러 테이블을 합친 VIEW는 조회만 가능하다.
- 기본키의 유일성 유지관리가 어렵다.

4-3 중복 테이블 생성

❖ 개념

- ✓ 많은 양의 정보들을 자주 Group By, Sum 등과 같은 집계 함수를 이용해서 실시간으로 통계 정보들을 추출하고 있는데, 이를 위해서 특정 통계 테이블을 두거나 중복 테이블을 추가할 수 있다.

❖ 중복 테이블 생성 판단 근거

- ✓ 정규화에 충실하면 종속성, 활용성은 향상되나 수행 속도 증가가 발생하는 경우
- ✓ 많은 범위를 자주 처리해야 하는 경우
- ✓ 특정 범위의 데이터만 자주 처리되는 경우
- ✓ 처리 범위를 줄이지 않고는 수행 속도를 개선할 수 없는 경우
- ✓ 요약 자료만 주로 요구되는 경우
- ✓ 추가된 테이블의 처리를 위한 오버헤드를 고려하여 결정
- ✓ 인덱스의 조정이나 부분 범위 처리로 유도, 클러스터링을 이용하여 해결할 수 있는지를 철저히 검토한 후 결정

❖ 중복 테이블 유형

- ✓ 집계용 테이블, 이력관리용 테이블, INDEXING 테이블, Statement 테이블, 부분통합 테이블

4-4 중복 칼럼 생성

❖ 개념

- ✓ 데이터 정합성의 유지를 위하여 논리 데이터 모델링 과정에서 정규화를 통하여 중복 칼럼을 최대한 제거하는 작업을 수행한다.
- ✓ 그런데 물리 데이터 모델링 과정에서 이러한 정규화를 여기면서 다시 데이터의 중복(중복 칼럼 생성)을 수행하곤 한다.

❖ 중복 칼럼 생성을 고려하는 경우

- ✓ 빈번하게 조인을 일으키는 칼럼
- ✓ 조인의 범위가 다량인 경우를 온라인화해야 하는 경우(속도가 중요)
- ✓ 액세스의 조건으로 자주 사용되는 칼럼
- ✓ 자주 사용되는 액세스 조건이 다른 테이블에 분산되어 있어 상세한 조건 부여에도 불구하고 액세스 범위를 줄이지 못하는 경우
- ✓ 접근 경로의 단축을 위해서 사용
- ✓ 연산된 결과를 주로 사용하는 경우
- ✓ 여러 칼럼으로 판단된 값이 검색의 조건으로 사용되는 경우

4-4 중복 칼럼 생성

❖ 중복 칼럼 생성 방법

- ✓ 복사된 칼럼의 도메인은 원본 칼럼과 동일하게 해야 함
- ✓ 상위 레벨의 테이블에 집계된 칼럼을 추가(M:1 관계)
- ✓ 하위 레벨의 테이블로 중복 칼럼을 복사(M:1 관계)
- ✓ 여러 개의 로우로 구성되는 값을 하나의 로우에 나열(칼럼으로 관리)
- ✓ 기본키의 칼럼이 길거나 여러 개의 칼럼으로 구성되어 있는 경우 인위적인 기본키를 추가

❖ 중복 칼럼 생성 시 유의사항

- ✓ 다중 테이블 클러스터링으로 해결할 수 있는지 검토
- ✓ SQL GROUP 함수 이용하여 처리할 수 있는지 검토
- ✓ 저장 공간의 지나친 낭비를 고려하여 적절한 대비책을 마련
- ✓ 반복 칼럼은 특별한 경우를 제외하고는 절대 사용할 필요가 없고, 있다면 `sum(decode..)` 용법과 같은 SQL 기법 등을 활용하여 이러한 부분을 피할 수 있도록 함
- ✓ 경우에 따라 상대 테이블의 ROWID를 복사하는 경우가 효과적일 때도 있음
- ✓ 데이터의 일관성 보장에 유의해야 한다. 성능을 향상시키기 위해서 데이터의 일관성을 그르치는 일이 일어나서는 안됨
- ✓ 칼럼의 중복이 지나치게 심하면 데이터 처리의 오버헤드가 발생하게 됨
- ✓ 사용자나 프로그램은 반드시 원본 칼럼만 수정하는 것이 바람직

4-4 중복 칼럼 생성

❖ 중복 칼럼 생성 시 주의 사항

- ✓ 일반적으로 수행 속도를 우려해서 지나치게 많은 중복 칼럼을 생성하고 있음
- ✓ 클러스터링, 결합 인덱스, 적절한 SQL을 이용하면 특별한 경우를 제외하고는 거의 해결 가능
- ✓ 중복 칼럼을 이용하면 손쉽게 액세스 효율을 개선할 수 있으나 지나친 중복화는 반드시 반대급부가 따르게 됨
- ✓ JOIN, SUB-QUERY 액세스 경로의 최적화 방안을 보다 철저히 강구

❖ 예제] 인터넷 쇼핑몰의 업무처리규정을 토대로 데이터 모델을 작성하시오.

인터넷 쇼핑몰에서는 다양한 가전제품을 판매하고 있다.

회원인 경우에는 포인트가 적립되고, 관심 분야를 체크하면 최신 가전제품들에 대한 신기술 정보를 모바일로 제공해 준다. 회원으로 가입하기 위해서는 회원ID, 비밀번호, 이름, 생년월일, 전화번호, 핸드폰번호, 이메일주소, 주소를 입력해야 한다. 회원으로 등록하지 않아도 주문은 가능하다.

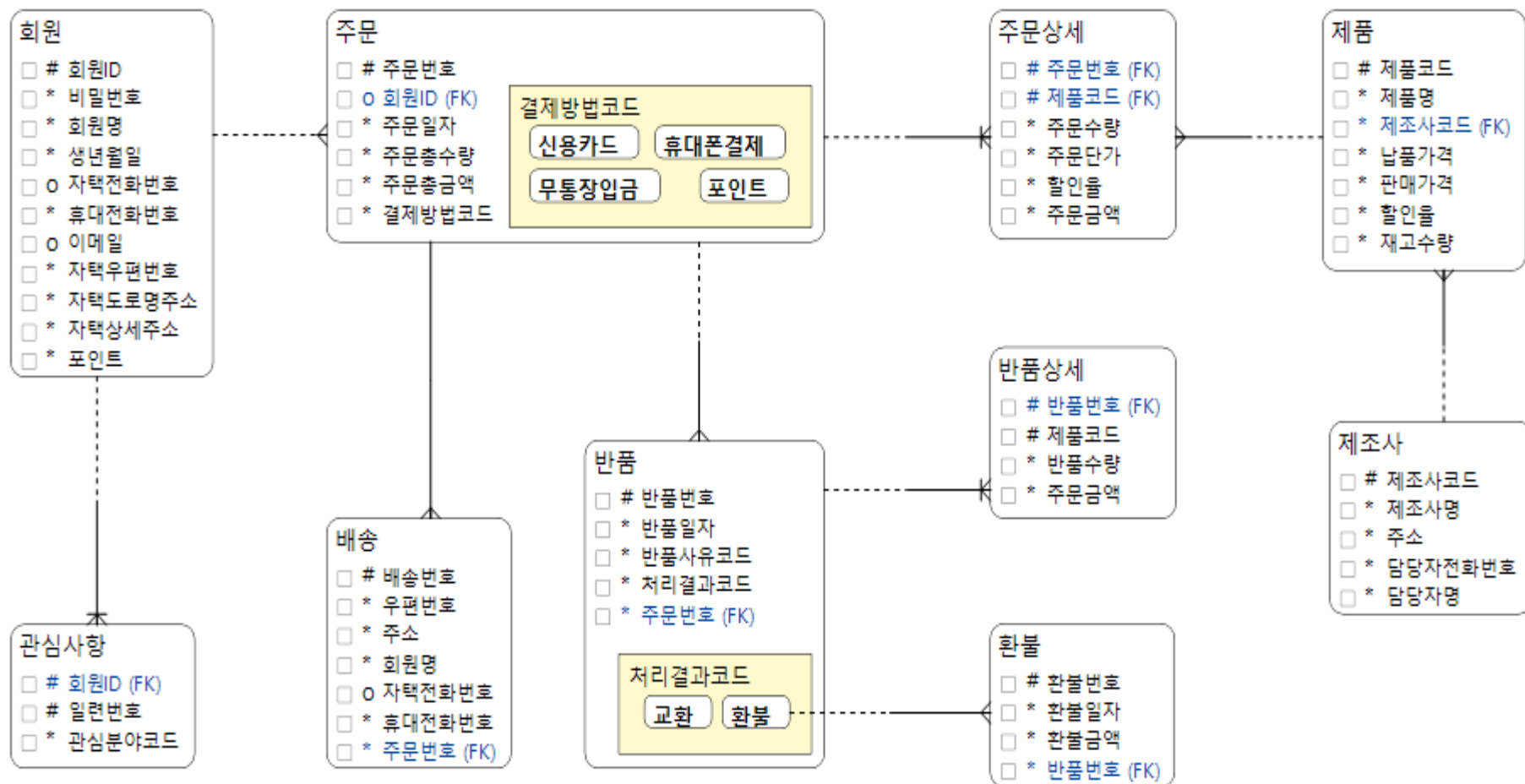
가전제품을 관리하기 위해 제품명, 제조사, 납품가, 판매가, 할인율, 재고수량 등을 관리하고 있다. 이 쇼핑몰에서는 가전제품을 공급해 주는 제조사의 정보를 기록하는데, 제조사명, 전화번호, 담당자명, 주소 등을 관리한다.

회원이 가전제품을 주문할 때에는 일자, 수량, 단가, 할인율 등을 관리되고, 결제방법은 신용카드, 휴대폰 결제, 무통장 입출금, 포인트 등의 네가지 방법이 있다. 배송하기 위해서는 회원인 경우에는 회원 정보에서 주소 정보를 참조하지만, 비회원인 경우에는 우편번호, 주소, 이름, 전화번호, 핸드폰 번호 등을 입력해야 한다.

배송 후에는 반품이나 환불이 발생할 수 있다. 반품할 때에는 일자, 수량, 반품사유, 처리결과 등이 기록되고, 처리결과는 교환과 환불이 있다. 환불할 경우에는 일자, 금액에 대한 정보가 관리되어야 한다.

물리 데이터 모델링 - 실습

실습1. 개념/논리 데이터 모델링



물리 데이터 모델링 - 실습

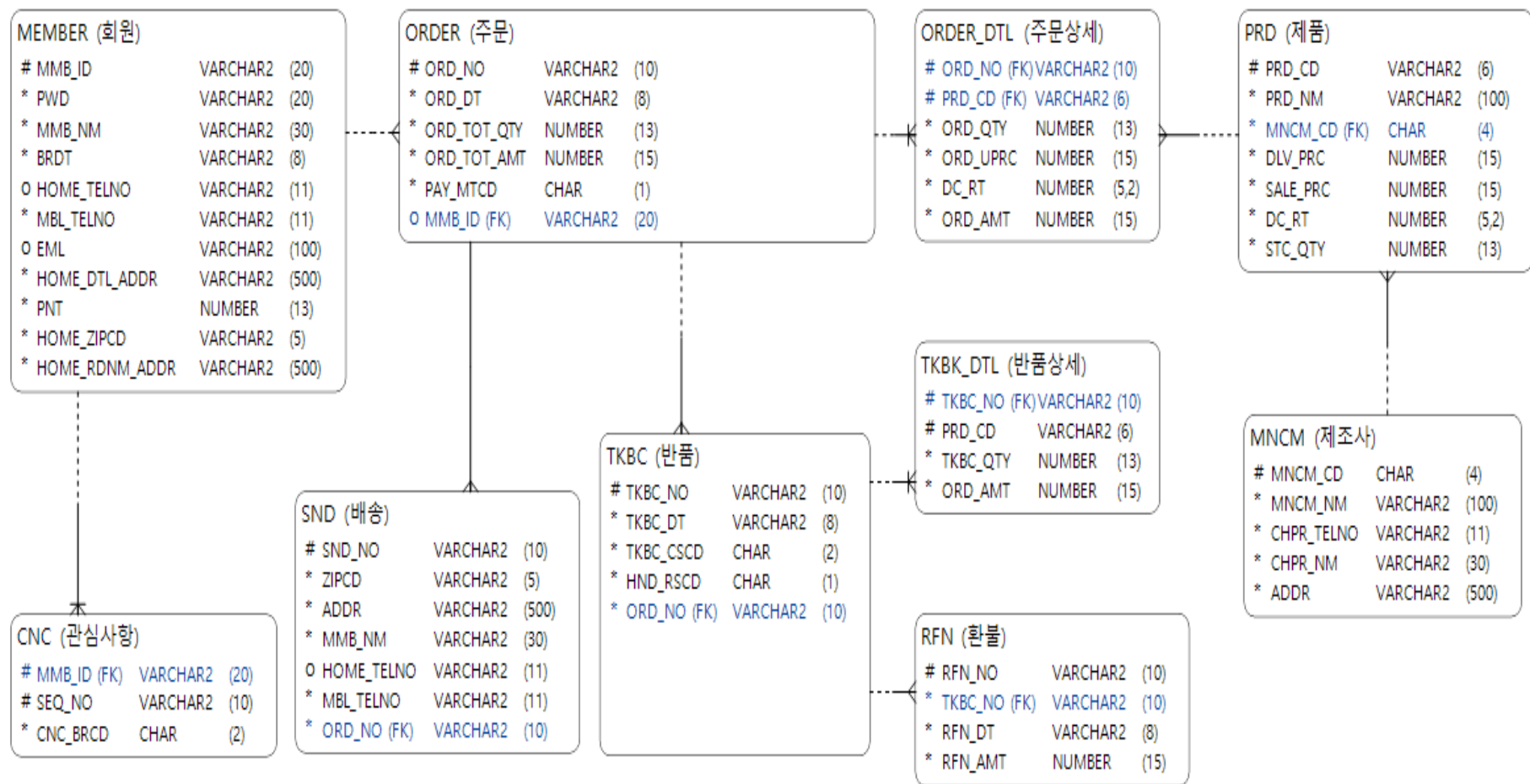
실습2. 데이터 표준화

1	논리명(단어명)	물리명(영문 약어명)	영문출처명(영문명)	한자	엔터티 분류어	속성 분류어	동의어	설명(정의)							
2	ID	ID	Identification		N	Y		신분증							
3	가격	PRC	Price		N	Y		물건이 지니고 있는 가치를 돈으로 나타낸 것							
4	결과코드	RSCD	Result Code		N	Y		어떤 원인으로 결말이 생김							
5	결제	PAY	Payment	1	대표 도메인명	도메인명	도메인유형	도메인그룹	논리 데이터타입	데이터길이	소수점	최소값	최대값	포맷	설명
6	관심	CNC	Concern	2	번호	반품번호VC10	번호		VARCHAR	10					
7	금액	AMT	Amount	3	번호	배송번호VC10	번호		VARCHAR	10					
8	납품	DLV	Delivery	4	번호	비밀번호VC20	번호		VARCHAR	20					
9	단가	UPRC	Unit Price	5	번호	우편번호VC5	번호		VARCHAR	5					
10	담당자	CHPR	Charge Person	6	번호	일련번호VC10	번호	1	용어명	용어설명	도메인명	논리 데이터타입	길이	소수점	
11	도로명	RDNM	Road Name	7	번호	전화번호VC11	번호	2	결제방법코드	결제방법코드	결제코드CH1				
12	명	NM	Name	8	번호	주문번호VC10	번호	3	관심분야코드	관심분야코드	관심코드CH2				
13	반품	TKBC	Take Back	9	번호	카드번호VC20	번호	4	납품가격	납품가격	단가NM15				
14	방법코드	MTCD	Method Code	10	번호	환불번호VC10	번호	5	담당자명	담당자명	이름VC30				
15	배송	SND	Send	11	금액	금액NM15	일반	6	담당자전화번호	담당자전화번호	전화번호VC11				
16	번호	NO	Number	12	금액	단가NM15	일반	7	반품번호	반품번호	반품번호VC10				
17	분야코드	BRCD	Branch Code	13	내용	주소VC500	일반	8	반품사유코드	반품사유코드	사유코드CH2				
18	비밀번호	PWD	Password	14	명	이름VC30	일반	9	반품수량	반품수량	수량NM13				
19	사유코드	CSCD	Cause Code	15	명	제조사명VC100	일반	10	배송번호	배송번호	배송번호VC10				
20	사항	MTR	MATTER	16	명	제품명VC100	일반	11	비밀번호	비밀번호	비밀번호VC20				
21	상세	DTL	Detail	17	비율	비율NM5,2	일반	12	생년월일	생년월일	일자VC8				
22	생년월일	BRDT	Birth Date	18	수량	수량NM13	일반	13	우편번호	우편번호	우편번호VC5				
23	수량	QTY	Quantity	19	수량	포인트NM13	일반	14	이메일	이메일	이메일VC100				
24	우편번호	ZIPCD	Zip Code	20	식별자	IDVC20	일반	15	일련번호	일련번호	일련번호VC10				
25	율	RT	Rate	21	식별자	이메일VC100	일반	16	자택도로명주소	자택도로명주소	주소VC500				
				22	일시	일자VC8	일반	17	자택상세주소	자택상세주소	주소VC500				
								18	자택우편번호	자택우편번호	우편번호VC5				
								19	자택전화번호	자택전화번호	전화번호VC11				
								20	제조사명	제조사명	제조사명VC100				
								21	제조사코드	제조사코드	제조사코드CH4				
								22	제품명	제품명	제품명VC100				

35

물리 데이터 모델링 - 실습

실습3. 물리 데이터 모델링



물리 데이터 모델링 - 실습

실습4. 스크립트 생성

```
CREATE TABLE MEMBER
(
  MMB_ID          VARCHAR2(20) NOT NULL,
  PWD             VARCHAR2(20) NOT NULL,
  MMB_NM          VARCHAR2(30) NOT NULL,
  BRDT           VARCHAR2(8) NOT NULL,
  HOME_TELNO      VARCHAR2(11),
  MBL_TELNO       VARCHAR2(11) NOT NULL,
  EML             VARCHAR2(100),
  HOME_DTL_ADDR   VARCHAR2(500) NOT NULL,
  PNT             NUMBER(13) NOT NULL,
  HOME_ZIPCD      VARCHAR2(5) NOT NULL,
  HOME_RDNM_ADDR  VARCHAR2(500) NOT NULL
);

COMMENT ON COLUMN MEMBER.MMB_ID IS '회원ID';
COMMENT ON COLUMN MEMBER.PWD IS '비밀번호';
COMMENT ON COLUMN MEMBER.MMB_NM IS '회원명';
COMMENT ON COLUMN MEMBER.BRDT IS '생년월일';
COMMENT ON COLUMN MEMBER.HOME_TELNO IS '주택전화번호';
COMMENT ON COLUMN MEMBER.MBL_TELNO IS '휴대전화번호';
COMMENT ON COLUMN MEMBER.EML IS '이메일';
COMMENT ON COLUMN MEMBER.HOME_DTL_ADDR IS '주택상세주소';
COMMENT ON COLUMN MEMBER.PNT IS '포인트';
COMMENT ON COLUMN MEMBER.HOME_ZIPCD IS '주택우편번호';
COMMENT ON COLUMN MEMBER.HOME_RDNM_ADDR IS '주택도로명주소';
COMMENT ON TABLE MEMBER IS '회원';
ALTER TABLE MEMBER
  ADD CONSTRAINT MEMBER_PK PRIMARY KEY ( MMB_ID );
```