

1

## 파일 시스템

# 01 파일 시스템

## 1 1960~70년대의 파일시스템

- 각 응용프로그램은 개별적으로 자신의 데이터를 파일로 관리하고 유지
- 개발자는 각 프로그램에서 해당 파일을 열고, 읽고, 갱신함





※ 출처 : 실무에 바로 활용하는 소프트웨어공학, 김희영저, 21세기사

# 01 파일 시스템

## 2 파일시스템의 문제점

### 데이터의 중복성

-  파일시스템은 프로그램마다 데이터 종속성 등으로 인해서 공유가 안 되는 경우가 많아서 프로그램마다 같은 정보를 중복해서 저장하는 경우가 많음
-  이는 저장공간의 낭비를 초래하기도 하지만, 데이터를 관리하는 측면에서 같은 정보를 여러 곳에서 보관하면 데이터를 수정할 경우 모든 데이터를 수정해야 하는 문제가 발생함

# 01 파일 시스템

## 2 파일시스템의 문제점



### 데이터의 중복성

- 🔍 데이터의 종속성은 프로그램의 구조가 데이터의 구조에 영향을 받는 것을 의미함
- 🔍 데이터의 구조가 프로그램의 데이터 저장방식을 결정하고 반대로 프로그램의 데이터 저장방식에 따라 데이터의 저장방식이 바뀔 수 있음
- 🔍 데이터의 종속성 때문에 데이터의 구조가 변경되면 프로그램까지 같이 바뀌는 비용이 들기 때문에 프로그램 개발과 유지보수가 어려워짐

# 01 파일 시스템

## 2 파일시스템의 문제점

### 데이터의 불일치

-  중복된 데이터가 서로 일치하지 않는 경우를 데이터 불일치라고 함
-  데이터 중복성의 중복성과 같이 여러 곳에서 같은 정보가 저장되기 때문에 모두 수정되지 않는 경우, 어떤 것이 올바른 것인지 알 수 없음

## 2 관계형 데이터베이스

## 02 관계형 데이터베이스

### 1 관계형 데이터베이스의 탄생

- 🔍 1970년대 코드(E.F. Codd)에 의해 제안
- 🔍 1977년 래리 엘리슨이 코드의 아이디어를 활용하여 “오라클”출시
- 🔍 이후 SQL이 표준화되고, 관계형 데이터베이스의 성능 향상으로 각광을 받게 됨

## 02 관계형 데이터베이스

### 1 관계형 데이터베이스의 탄생

 대표적인 관계형 데이터베이스 제품

- 오라클
- DB2
- SQL 서버
- MySQL
- Sybase



## 02 관계형 데이터베이스

### 2 관계형 데이터베이스의 구성

- 엔터티 : 흔히 테이블이라고 함
- 애트리뷰트 : 흔히 칼럼이라고 함
- 인스턴스 : 데이터가 입력된 내역

학번	성명	전화번호	주소

테이블의 예

학번	성명	전화번호	주소

학생 테이블의 주소 칼럼

※ 출처 : 실무에 바로 활용하는 소프트웨어공학, 김희영저, 21세기사

## 02 관계형 데이터베이스

### 2 관계형 데이터베이스의 구성



#### 스키마

: 데이터 구조와 제약조건을 정의한 것

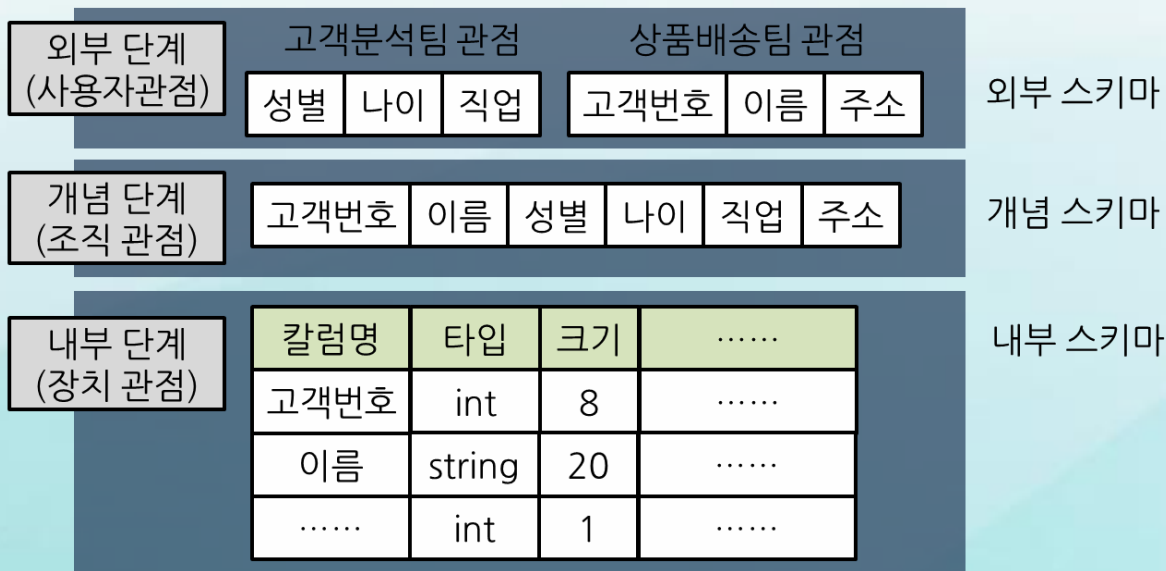


#### 3단계 구조

- 외부 스키마 : 개별 사용자의 관점에서 바라본 데이터베이스 표현
- 개념 스키마 : 조직 전체의 관점에서 데이터베이스를 이해하고 표현한 것
- 내부 스키마 : 저장 장치에 저장하기 위한 관점에서 이해하고 표현

## 02 관계형 데이터베이스

### 2 관계형 데이터베이스의 구성



※ 출처 : 실무에 바로 활용하는 소프트웨어공학, 김희영저, 21세기사

## 2 3단계 데이터베이스 설계

## 03 3단계 데이터베이스 설계

### 1 3단계 데이터 베이스 설계의 절차






현실 세계를 개념적으로 설계하여 논리적 설계와 물리적 설계로 데이터베이스를 상세화해 나가는 것이 일반적 과정

개념적 설계	현실세계를 데이터 주제위주로 추상화
논리적 설계	개념적 데이터구조를 기반으로 구체화
물리적 설계	컴퓨터 내부의 물리적 장치에 DB구현

## 03 3단계 데이터베이스 설계

### 1 3단계 데이터 베이스 설계의 절차



#### 개념적 설계

-  요구사항 분석의 결과를 기초 입력자료로 활용하여 설계를 시작
-  데이터 요소와 요소간의 관계를 표현
-  개념 ERD로 도식화하면서 설계를 진행함

## 03 3단계 데이터베이스 설계

### 1 3단계 데이터 베이스 설계의 절차




#### 논리적 설계

-  개념적 설계 단계에서 생성한 개념적 데이터베이스 구조를 기반으로 논리적 구조를 설계
-  ERD를 테이블 스키마로 변환

## 03 3단계 데이터베이스 설계

### 1 3단계 데이터 베이스 설계의 절차

#### 물리적 설계

-  논리적 설계의 결과를 기반으로 저장장치에 DB가 구현될 수 있도록 함
-  저장장치에 실제로 데이터를 저장하기 위하여 내부 저장장치와 접근경로 등을 설계
-  저장장치에 적합한 인덱스 구조와 탐색기법 등을 정의

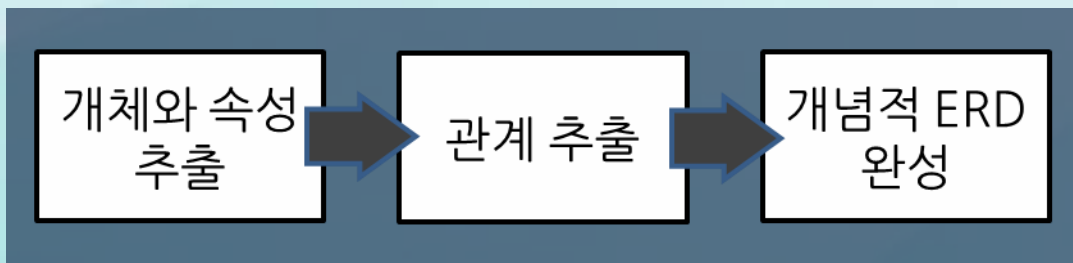


## 03 3단계 데이터베이스 설계

### 2 개념적 설계




분석단계에서 진행된 요구사항의 분석결과를 기반으로 현실 세계에서 중요한 데이터 요소인 개체, 즉, 엔터티를 추출한 후 개체간의 관계를 결정하여 ERD로 표현하는 것



## 03 3단계 데이터베이스 설계

### 2 개념적 설계

#### 개체와 속성의 추출

 DFD에서 추출방법


- 데이터 저장소와 Data Dictionary에서 개체와 속성을 추출
- 개체는 테이블로 전환, 속성은 칼럼으로 전환

개체	속성
과목	과목번호, 과목명, 담당교수, 강의실
학적	학번, 성명, 수강신청과목, 과목교수

## 03 3단계 데이터베이스 설계

### 2 개념적 설계

#### 개체와 속성의 추출

 클래스 다이어그램에서 추출방법

- 엔터티 클래스에서 추출

개체	속성
사용자	아이디, 비밀번호, 이름, 주소, 생년월일, 등록일, 전화번호

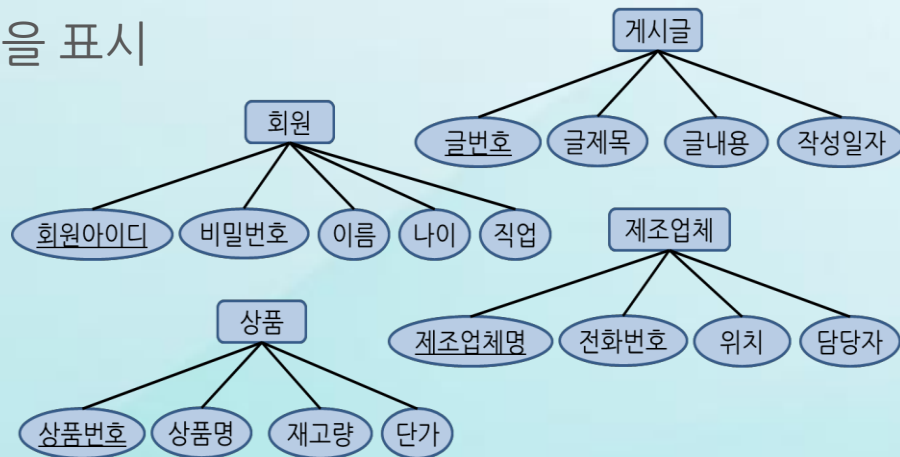
## 03 3단계 데이터베이스 설계

### 2 개념적 설계

#### 개체와 속성의 추출

🔍 요구사항 분석과정에서 개체와 속성을 추출

- 회원아이디의 언더바 : 기본키임을 표시



※ 출처 : 실무에 바로 활용하는 소프트웨어공학, 김희영저, 21세기사

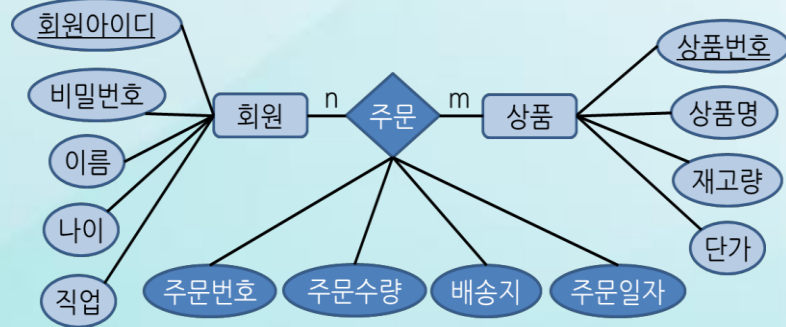
## 03 3단계 데이터베이스 설계

### 2 개념적 설계

#### 관계의 추출

🔍 관계는 의미 있는 연관성

- 마름모꼴의 주문 : 관계를 표시
- 카디널리티 : 관계를 맺고 있는 두 개체에  
서 한 개체의 인스턴스가 상대방 개체의  
인스턴스와 어떠한 수적인 관계를 갖는가  
를 의미



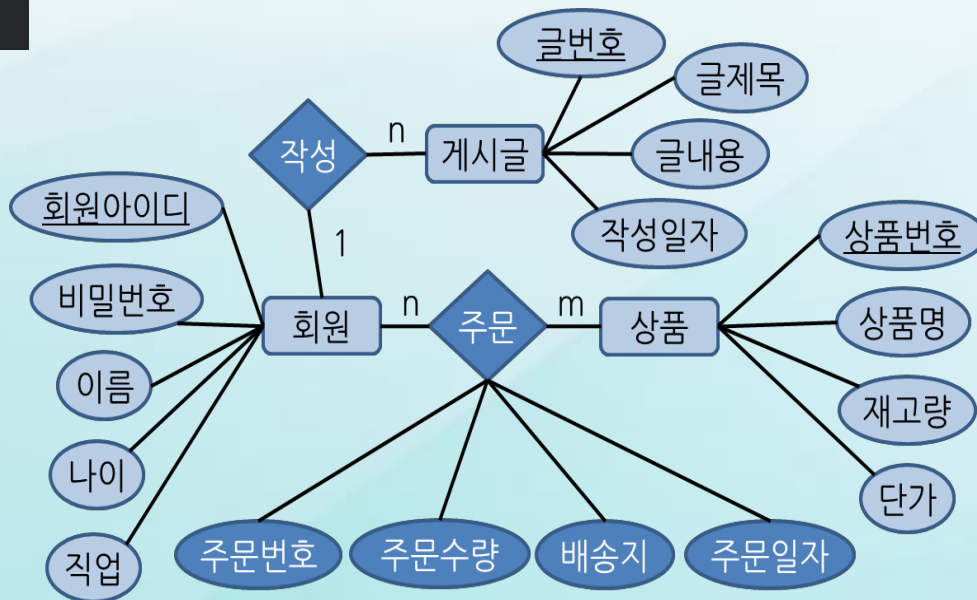
※ 출처 : 실무에 바로 활용하는 소프트웨어공학, 김희영저, 21세기사

## 03 3단계 데이터베이스 설계

### 2 개념적 설계

#### 관계의 추출

🔍 개념적 ERD는 전문가가 설계하는 것이 필요



※ 출처 : 실무에 바로 활용하는 소프트웨어공학, 김희영저, 21세기사


## 03 3단계 데이터베이스 설계

### 3 논리적 설계

- 🔍 DBMS가 처리할 수 있는 데이터베이스의 논리적 구조를 설계하는 것
- 🔍 내부 스키마로 전환되는 작업을 통해 구체화함
- 🔍 개념적 설계와의 차이
  - 다중 값 속성이나 복합속성의 표현을 허용하지 않음

## 03 3단계 데이터베이스 설계

### 3 논리적 설계


 개념적 ERD에서 논리적 ERD로의 변환규칙

- 모든 개체는 테이블로 변환
- (다:다)의 관계는 테이블로 변환
- (1:다)의 관계는 외래키를 추가
- 다중값 속성은 별도의 테이블을 추가



## 03 3단계 데이터베이스 설계


### 3 논리적 설계

 모든 개체는 테이블로 변환

- 개체이름은 그대로 테이블 이름으로 활용
- 개체가 가진 속성은 테이블의 속성이 됨
- 개체가 가진 키 속성은 테이블의 기본키(Primary key)로 변환

## 03 3단계 데이터베이스 설계

### 3 논리적 설계

 모든 개체는 테이블로 변환

- 회원 테이블

회원아이디	비밀번호	이름	나이	직업

- 상품테이블

상품번호	상품명	재고량	단가

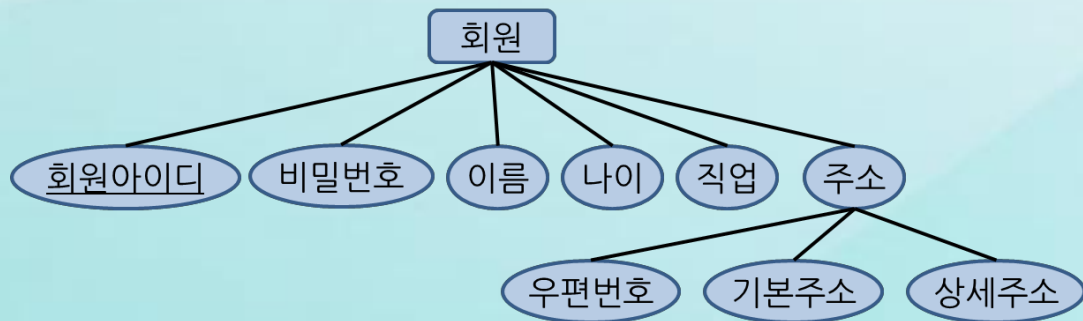
## 03 3단계 데이터베이스 설계

### 3 논리적 설계

모든 개체는 테이블로 변환

- 복합속성의 처리


: 단순속성으로 분리하여 별도의 칼럼,  
즉, 속성으로 정의



※ 출처 : 실무에 바로 활용하는 소프트웨어공학, 김희영저, 21세기사

## 03 3단계 데이터베이스 설계

### 3 논리적 설계

 모든 개체는 테이블로 변환

- 개선된 회원 테이블

회원 아이디	비밀 번호	이름	나이	직업	우편 번호	기본 주소	상세 주소

## 03 3단계 데이터베이스 설계

### 3 논리적 설계

🔍 (다:다)의 관계는 테이블로 변환

- 주문은 (다:다)의 관계  
: 별도의 테이블 추가

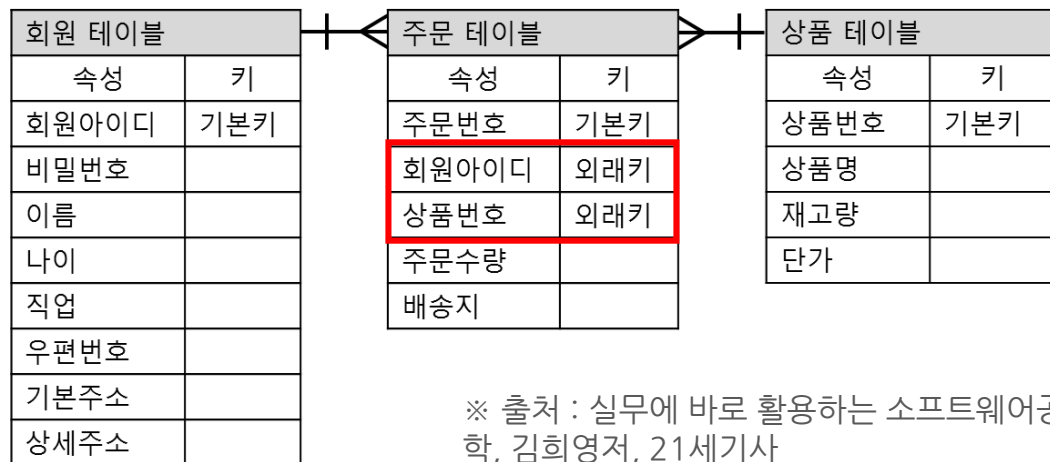
주문번호	회원아이디	상품번호	주문수량	배송지	주문일자

## 03 3단계 데이터베이스 설계

### 3 논리적 설계

🔍 (다:다)의 관계는 테이블로 변환

- 외래키를 통해 관계를 유지



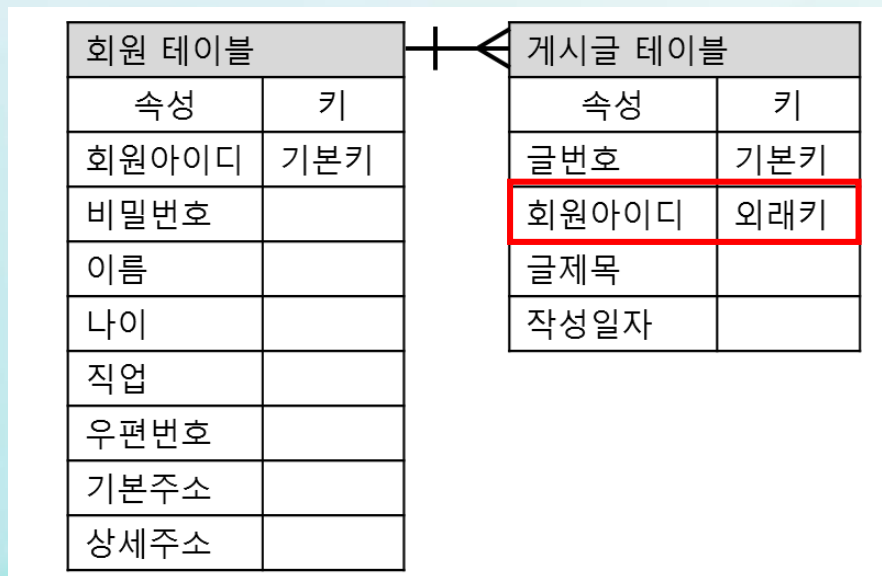
## 03 3단계 데이터베이스 설계

### 3 논리적 설계

🔍 (1:다)의 관계는 외래키를 추가

- 게시글 테이블에서 누가 글을 작성하였는지 알기 위해서는 회원아이디를 외래키로 추가 필요

※ 출처 : 실무에 바로 활용하는 소프트웨어공학, 김희영저, 21세기사



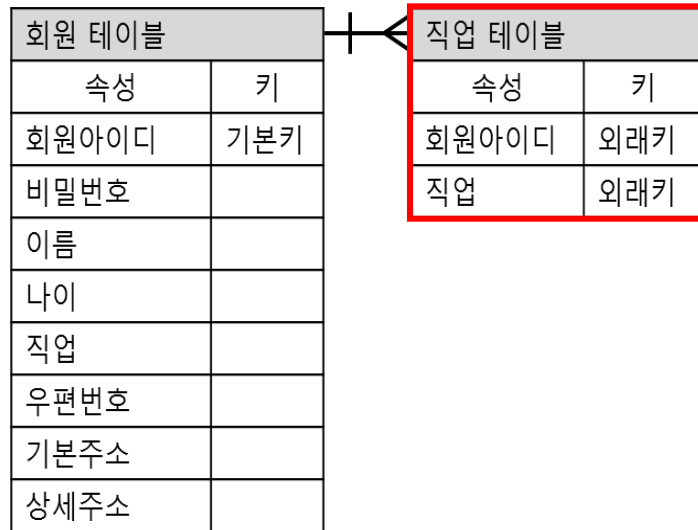
## 03 3단계 데이터베이스 설계

### 3 논리적 설계

다중값 속성은 별도의 테이블 추가

- 회원 테이블에서 직업테이블 분리

※ 출처 : 실무에 바로 활용하는 소프트웨어공학, 김희영저, 21세기사





## 03 3단계 데이터베이스 설계

### 4 물리적 설계

- 🔍 특정 DBMS제품에 맞는 설계를 의미
- 🔍 이미 선정된 하드웨어를 고려하여 필요한 인덱스의 구조나 내부 저장장치 등에 대한 설계
- 🔍 물리적 설계 내역
  - 저장공간 설계
  - 무결성 설계와 인덱스 설계
  - 데이터베이스 구축

## 03 3단계 데이터베이스 설계

### 4 물리적 설계

#### 저장공간 설계

- 데이터베이스의 초기 사이즈, 증가 사이즈, 트랜잭션 관련 옵션, 최대 사이즈와 자동증가 등을 고려
- 데이터 증가 예상 건수, 주기, 저장 데이터의 총 길이 등을 고려

 테이블 저장 용량을 분석

## 03 3단계 데이터베이스 설계

### 4 물리적 설계

#### 저장공간 설계

- 테이블 인덱스에 대한 크기를 기준으로 오브젝트별 용량을 산정
- 테이블 저장공간 별 용량의 합계로 전체 테이블 용량을 산정
- 테이블 스페이스에 따른 디스크 용량과 I/O 분산설계를 통해

 데스크 용량을 산정

## 03 3단계 데이터베이스 설계

### 4 물리적 설계



무결성 설계와 인덱스 설계

- 데이터 무결성의 점검

- 실체 무결성, 영역 무결성, 참조 무결성, 사용자 정의 무결성의 순서로 점검

## 03 3단계 데이터베이스 설계

### 4 물리적 설계

#### 무결성 설계와 인덱스 설계

- 인덱스는 테이블에 접근하는 모든 경로를 수집하여 결정
  - 반복 수행되는 접근경로
  - 분포도가 양호한 칼럼
  - 조회 조건에 사용되는 칼럼
  - 자주 결합되어 사용되는 칼럼 등
- 인덱스 후보목록 활용

## 03 3단계 데이터베이스 설계

### 4 물리적 설계

🔍 물리적 설계의 데이터베이스 구축

- “CREAT” 명령어를 통해 데이터베이스 구축
- 물리적 설계는 특정 DBMS제품에 의존적이므로 해당 제품을 판매하는 기업에 교육을 문의하는 것이 바람직함

