

# 데이터베이스

## 강의 노트

제 1 회차  
데이터베이스 개요

## 데이터베이스의 기본 개념

1. 데이터와 정보의 차이점
2. 데이터베이스의 정의
3. 데이터베이스의 특징
4. 데이터베이스의 출현 배경

## 1. 데이터와 정보의 차이점

### 1) 데이터베이스의 정의

*데이터(Data)란?*

데이터란 현실 세계에서 단순한 관찰이나 측정을 통해서 수집된 **사실(Fact)** 또는 **값(Value)**이 어떤 **기준에 의해 정리되어 있는 것**

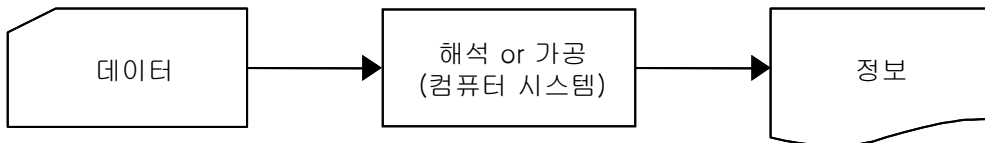
*정보(Information)란?*

어떤 상황에서 적절한 **의사 결정을 할 수 있도록 지원하는 지식**

컴퓨터 시스템과 같은 처리기를 통한 데이터의 유효한 해석이나 데이터 상호간의 관계



정보는 데이터가 어떤 목적에 의해 해석되거나 가공된 형태임



[데이터와 정보의 차이점]

## 2. 데이터베이스의 정의

### 1) 데이터베이스의 정의

*데이터베이스(Database)란?*

어느 한 조직의 다양한 응용 프로그램들이 공동으로 사용하는 데이터들을 통합하여 저장한 운영 데이터의 집합

사람들이 필요로 하는 데이터를 모아둔 것

데이터베이스(Database)



DB



여기서 잠깐!

#### ‘데이터베이스’ 및 ‘데이터베이스 시스템’이란 용어의 기원

1) 1963년 6월 미국 SDC(System Development Corporation)사가 개최한 제1차 심포지엄의 제목인 “컴퓨터 중심의 데이터베이스 개발과 관리(Development and Management of a Computer-centered Data Base)”에서 **데이터베이스란 용어가 공식적으로 처음 사용**된 것으로 알려져 있음

- 이때의 데이터베이스는 저장장치에 저장된 파일을 의미하며, 현재의 데이터베이스와는 차이가 있음
- 이때 데이터베이스의 영문 표기가 ‘Data Base’라는 두 단어로 표기되었다가, 점차 한 단어인 ‘Database’로 통일되었음

2) 1965년 9월 제2차 SDC 심포지엄의 제목인 “컴퓨터 중심의 데이터베이스 시스템(Computer-centered Data Base Systems)”에서 **데이터베이스 시스템이란 용어가 공식적으로 처음 사용**된 것으로 알려져 있음

## 2. 데이터베이스의 정의

### 2) 데이터베이스의 정의에 함축된 개념

#### 공용 데이터(Shared Data)

- 한 조직의 여러 응용 프로그램이 공동으로 사용하는 것
- 즉, 여러 사용자가 서로 다른 목적으로 공유함

#### 통합된 데이터(Integrated Data)

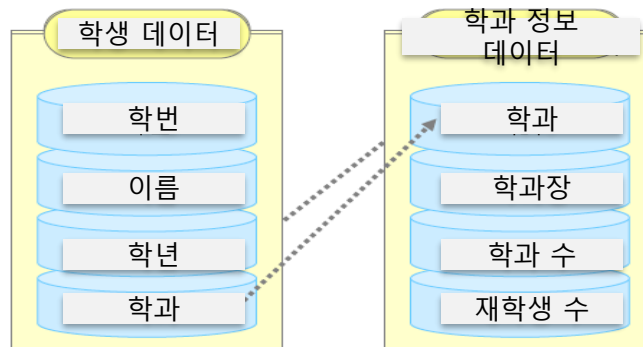
- 여러 부서에서 사용하는 데이터를 한 곳에 모아서 공동 관리하는 것 (기억장소 절약)
- 원칙적으로 동일한 데이터의 중복을 허용하지 않지만, 검색의 효율성을 위해서 **최소한의 중복(Minimal Redundancy)**을 허용하여 통합함



### 최소한의 중복(Minimal Redundancy)

- DB는 원칙적으로 데이터의 중복된 저장을 허용하지 않지만, 보다 효율적인 처리를 위해서 최소한의 중복이 부득이한 경우가 있음

예)



- ✓ 학생 데이터에 '학과'가 포함되어 있고, 학과 정보 데이터에도 '학과'가 포함되어 있으면 '학과'라는 데이터가 중복되었다고 생각할 수 있음
- ✓ 하지만, 이런 중복은 추후 어떤 학생이 속한 학과의 학과장이나 재학생 수를 검색하기 위한 부득이한 중복이므로, 이런 중복을 최소한의 중복이라고 함

- 최소한의 중복은 미리 파악해서 관리할 수 있으므로, 통제된 중복(Controlled Redundancy)이라고도 함

## 2. 데이터베이스의 정의

### 저장된 데이터(Stored Data)

- 컴퓨터가 접근할 수 있는 디스크와 같은 저장 매체에 저장된 것

### 운영 데이터(Operational Data)

- 조직의 운영에 기본적으로 반드시 필요한 데이터를 저장하는 것
- 조직의 고유한 기능을 수행하는데 필수적인 데이터를 저장하는 것
- 일시적으로 필요한 임시 데이터나 단순한 입출력 데이터는 운영 데이터에 해당하지 않음

### 3. 데이터베이스의 특징

#### 동시 공유(Concurrent Sharing)

- 여러 응용 프로그램이나 사용자들이 서로 다른 목적으로 데이터를 **동시에 사용**할 수 있음

#### 지속적인 변화(Continuous Evolution)

- DB에 저장된 데이터는 고정된 것이 아니며, **삽입, 삭제, 갱신** 등을 통해서 **지속적으로 변화**함으로써 현재의 정확한 데이터를 유지해야 함

#### 실시간 접근성(Real-Time Accessibility)

- 컴퓨터가 접근할 수 있는 기록 매체에 저장되어 관리되므로, **언제든지 필요한 시점에 바로 접근** 가능해야 함
- 즉, 어떤 질의에 대해 실시간에 바로 응답해야 함

#### 내용에 의한 참조(Content Reference)

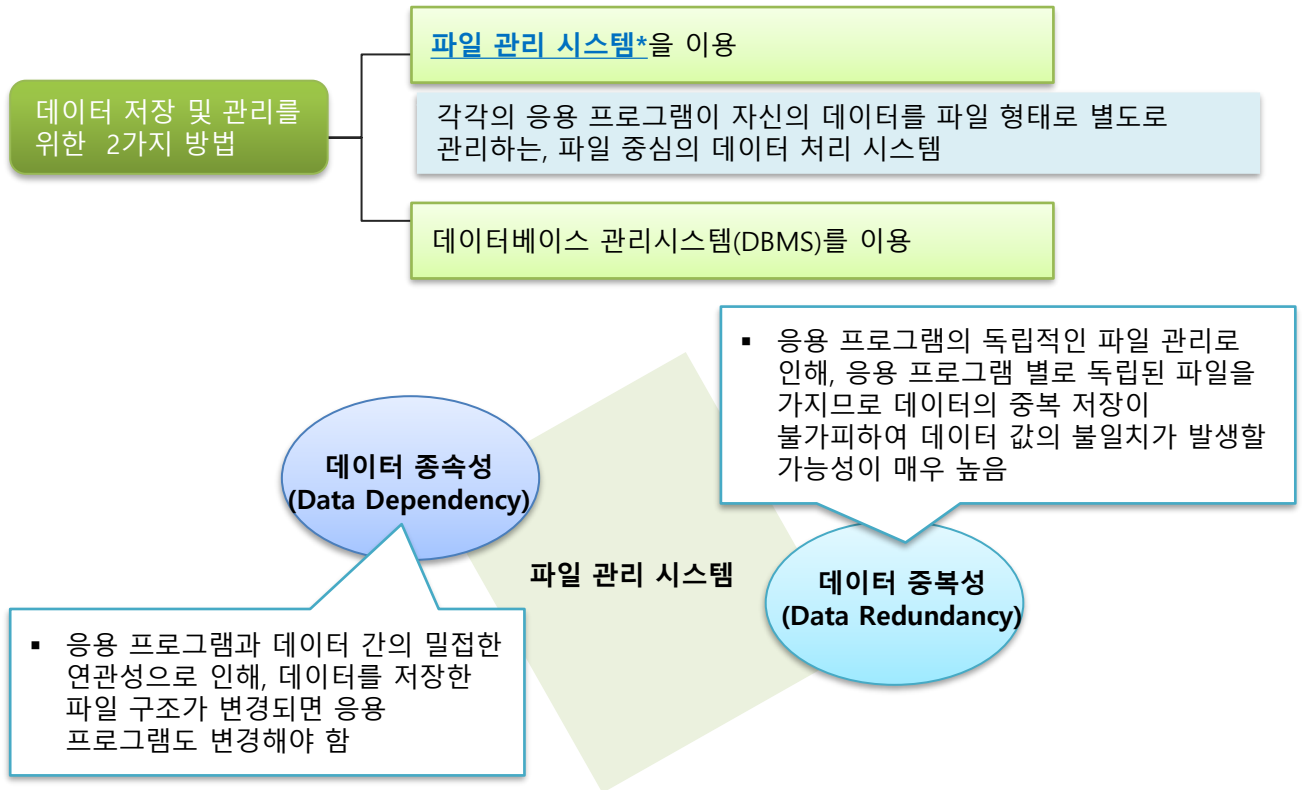
- 데이터가 저장된 주소나 위치가 아닌, 데이터의 **내용 즉, 값(Value)에 의해서 참조**됨
- 즉, 사용자가 원하는 데이터의 조건을 명시하면, 조건을 만족하는 레코드가 어디에 위치하든 접근 가능함

#### 데이터베이스 관리 시스템(Database Management System: DBMS)에 의한 관리

- DB의 구축 및 관리를 위해서 전용 소프트웨어인 **DBMS**가 필요함

## 4. 데이터베이스의 출현 배경

### 1) 파일 관리 시스템의 특징



### 2) 파일 처리 시스템의 문제점

데이터의 중복 저장으로 인한 비효율성

**데이터 일관성(Consistency)** 유지의 어려움

**데이터 무결성(Integrity)** 유지의 어려움

데이터 공유의 어려움



#### 4. 데이터베이스의 출현 배경



##### 데이터의 중복 저장으로 인한 비효율성

여러 파일에 동일한 데이터가 중복될 가능성이 높으므로, 저장공간의 낭비, 유지·보수의 어려움, 데이터 보안의 어려움 등이 유발됨

예)

‘회원정보 관리 프로그램’의 데이터 파일과 ‘회원등급 관리 프로그램’의 데이터 파일에 회원번호와 이름, 휴대폰 번호가 중복되어 있으므로, 회원 수가 많을수록 기억공간의 낭비가 심화됨



## 4. 데이터베이스의 출현 배경



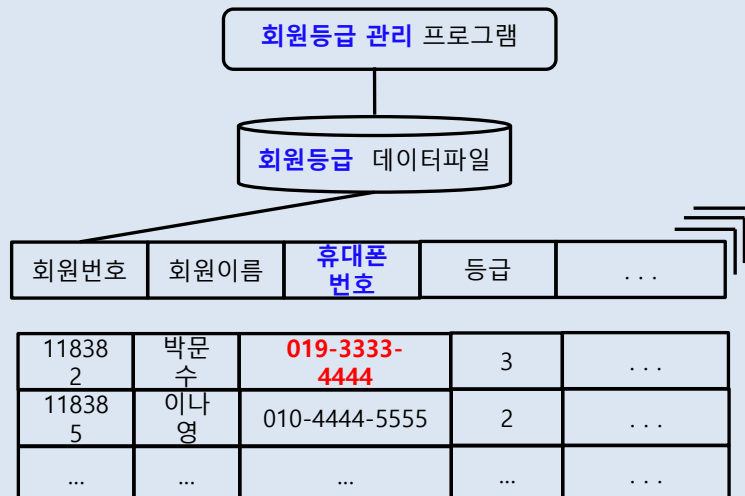
### 데이터 일관성(Consistency) 유지의 어려움

동일한 데이터가 여러 파일에 흩어져 있으므로, 시간이 지남에 따라 동일한 데이터가 서로 다른 값을 가질 가능성이 커짐

예)

어떤 회원의 '휴대폰번호'가 두 개의 다른 데이터 파일에 저장되어 있는데, 회원정보 관리 프로그램을 통해서만 휴대폰번호를 변경한 경우, 양쪽 파일의 데이터 값이 서로 다르므로 데이터의 일관성이 유지되지 않음

→ 회원 등급이 변경되어 회원정보 관리 프로그램에서 휴대폰으로 문자를 보내면, 변경 전의 휴대폰 번호로 문자를 보내게 되므로 문자가 제대로 전달되지 않음



## 4. 데이터베이스의 출현 배경



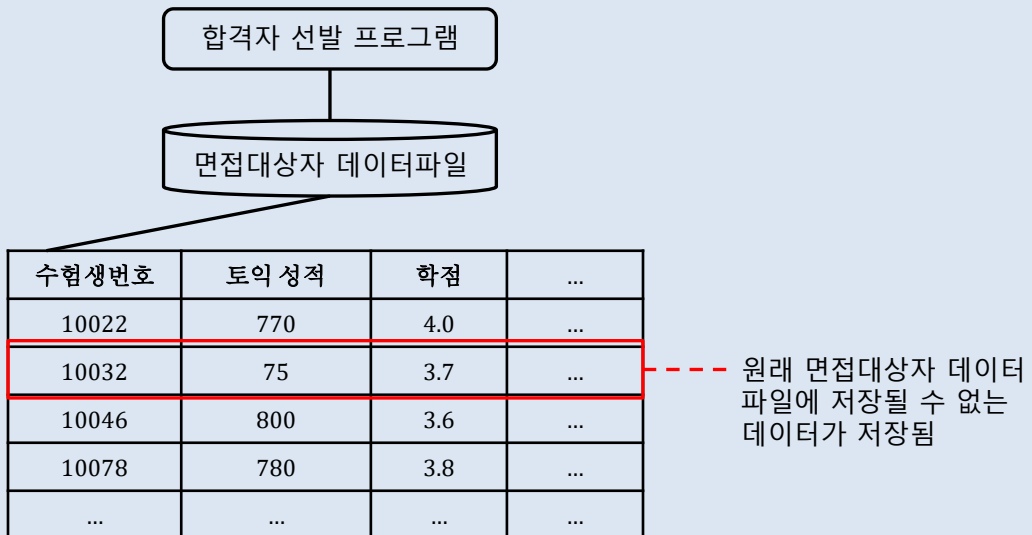
### 데이터 무결성(Integrity) 유지의 어려움

어떤 데이터가 반드시 만족해야 하는 무결성 제약조건을 일일이 프로그램에서 처리해야 하므로, 프로그램에서 빠뜨린 경우 무결성을 유지하기 어려움

예)

어떤 회사에서 서류 전형을 통해서 면접 대상자를 선정했는데, 이때 토익 성적이 700점 이상인 지원자만 선정했음. 그런데 면접 대상자 데이터를 입력하면서 실수로 한 지원자의 토익 점수를 75점(원래는 750점)으로 잘못 입력했음

→ 토익 성적 700점 이상인 수험생 정보만 저장되어야 한다는 무결성 제약조건을 위반하게 됨



#### 무결성 유지 방법

##### 파일 시스템에서 무결성 유지 방법

파일 시스템에서 무결성 제약조건을 유지하려면 각각의 무결성 제약조건을 일일이 응용 프로그램에서 처리해야 함

예) 면접 대상자 데이터 파일에 데이터를 저장하기 전에 응용 프로그램에서 IF문 등을 사용해서 토익 성적이 700점 이상인지를 먼저 확인한 다음, 700점 이상인 경우에만 데이터 파일에 저장되도록 처리해야 함

##### 데이터베이스에서 무결성 유지 방법

예) 데이터베이스를 정의할 때 무결성 제약조건(토익성적  $\geq 700$ )을 함께 정의하면, DB가 갱신될 때마다 DBMS가 자동으로 제약조건의 만족 여부를 검사하므로 응용 프로그램에서 별도로 처리할 필요가 없음

## 4. 데이터베이스의 출현 배경

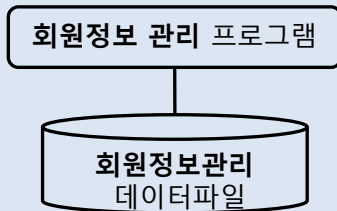


### 데이터 공유의 어려움

데이터의 구조가 응용 프로그램마다 다르므로 데이터 파일이 동일한 데이터를 포함하고 있어도 데이터 공유가 어려움

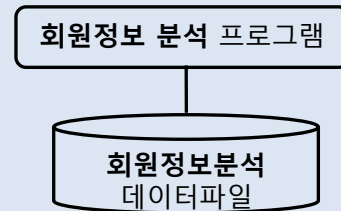
예)

회원정보 관리 프로그램과 회원정보 분석 프로그램은 동일한 데이터를 사용하지만, 회원이름과 직업의 **필드** 크기를 다르게 정의했기 때문에 데이터를 공유하기 어려움



[회원정보관리 데이터파일의 레코드 구조]

필드	구조
회원번호	숫자(8 Bytes)
회원이름	문자(8 Bytes)
휴대폰번호	문자(13 Bytes)
주소	문자(30 Bytes)
직업	문자(15 Bytes)
취미	문자(20 Bytes)

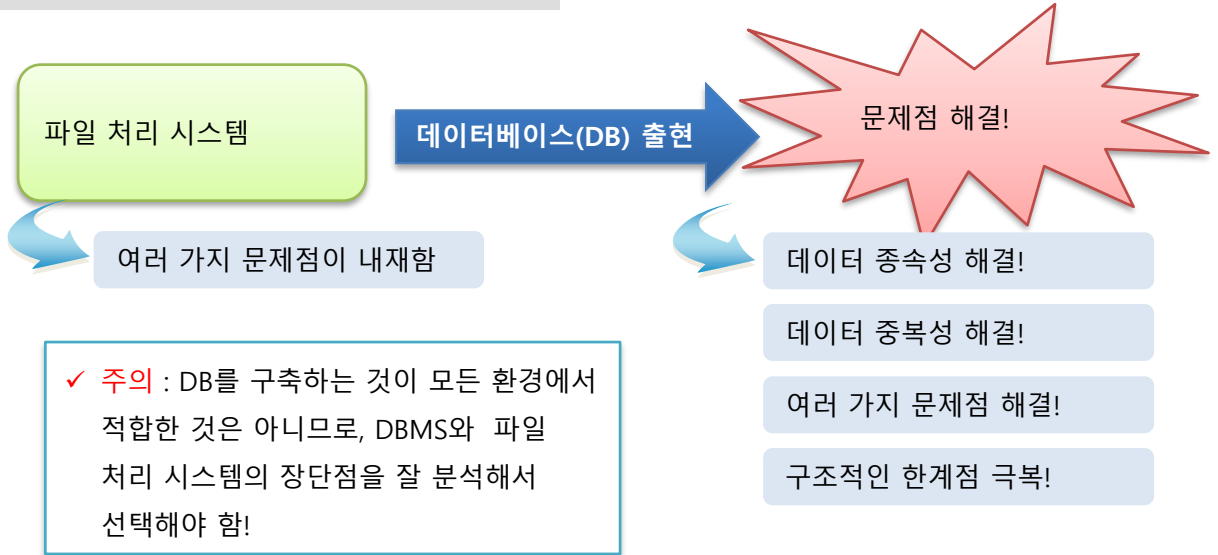


[회원정보분석 데이터파일의 레코드 구조]

필드	구조
회원번호	숫자(8 Bytes)
회원이름	문자(10 Bytes)
휴대폰번호	문자(13 Bytes)
주소	문자(30 Bytes)
직업	문자(20 Bytes)
취미	문자(20 Bytes)

## 4. 데이터베이스의 출현 배경

### 3) DB 출현 배경과 파일 처리 시스템의 연관성



### 데이터베이스의 필요성

1. DB 구축의 필요성
2. DB의 장단점
3. DB의 개념적 구성 요소
4. DB의 저장 구조

## 1. DB 구축의 필요성

### 1) 파일 관리 시스템의 특징

#### 기술적 측면에서의 필요성

데이터의 중복으로 인한 데이터의 일관성 유지가 어려움

데이터 접근에 대한 관리 및 통제가 필요함

데이터 파일과 프로그램간의 종속성으로 인해 비효율적임

데이터가 여기 저기 흩어져 있으므로 보안 유지가 어려움

#### 조직적 측면에서의 필요성

조직의 자산으로서 데이터를 관리할 필요가 있음

데이터의 공유가 필요함

데이터의 일관성 및 무결성 유지가 필요함

데이터의 최신성과 및 유용성을 유지할 필요가 있음

## 2. DB의 장단점

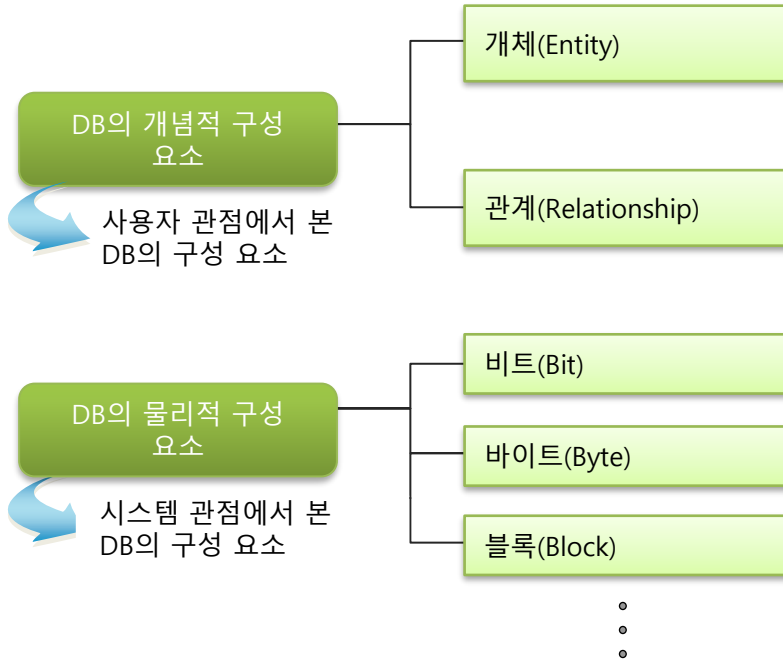
### 1) DB의 장단점

DB의 장점	DB의 단점
<ul style="list-style-type: none"> <li>• 데이터의 공유가 가능함</li> <li>• 데이터에 대한 접근을 통제할 수 있음</li> <li>• 중복 데이터가 적어짐</li> <li>• 효율적인 유지 보수가 가능함</li> <li>• 데이터의 일관성 유지가 가능함</li> <li>• 생산성이 향상됨</li> </ul>	<ul style="list-style-type: none"> <li>• 초기 개발 비용이 많이 듦</li> <li>• DB 관리를 전담할 전문 인력(DB 관리자)이 필요함</li> <li>• 적시에, 또한 지속적인 유지보수가 수반되지 않으면 무용지물이 될 수 있음</li> </ul>



### 3. DB의 개념적 구성 요소

#### 1) DB의 개념적 구성 요소



#### 2) 개체(Entity)

##### 개체

##### 정의

DB가 저장하는 **유형, 무형**의 정보 대상으로, **존재하면서 서로 구별될 수 있는 요소**

##### 특징

- 파일 시스템에서 '**레코드(Record)**'에 대응함
- 단독으로 존재 가능하고, 정보로서의 역할이 가능함
- **하나 이상의 속성(Attribute)**으로 구성됨

예) 개체 - 학생  
속성 - 학번, 이름, 학과 등

##### 종류

- 유형의 개체 : 사람, 집 등 물리적으로 존재하는 개체
- 무형의 개체 : 인사, 급여, 교과목 등 개념적으로 존재하는 개체

### 3. DB의 개념적 구성 요소

#### 2) 개체(Entity)

##### 속성

##### 정의

개체의 특성을 나타내는 요소로, **이름을 가진 정보의 가장 작은 논리적인 단위**

##### 특징

- 파일 시스템에서 '데이터 항목(Data Item)' 혹은 '**필드(Field)**'에 대응함
- 단독으로 존재할 때는 대개 무의미함

#### 3) 관계(Relationship)

##### 정의

일반적으로 **개체들 간의 의미 있는 연결 또는 연관성을 의미**하는 요소

##### 특징

- 관계도 **하나의 개체로 간주**될 수 있음
- 속성 관계와 개체 관계로 세분할 수 있으며, 그냥 '관계'라고 하면 '개체 관계'를 의미함

##### 유형

- **일 대 일(1 : 1)** : 한 개 개체가 한 개 개체와 연관되어 있는 유형
- **일 대 다(1 : n)** : 한 개 개체가 여러 개체와 연관성이 있는 유형
- **다 대 다(m : n)** : 여러 개체가 여러 개체와 연관성이 있는 유형

## 4. DB의 저장 구조

### 1) DB의 저장 구조

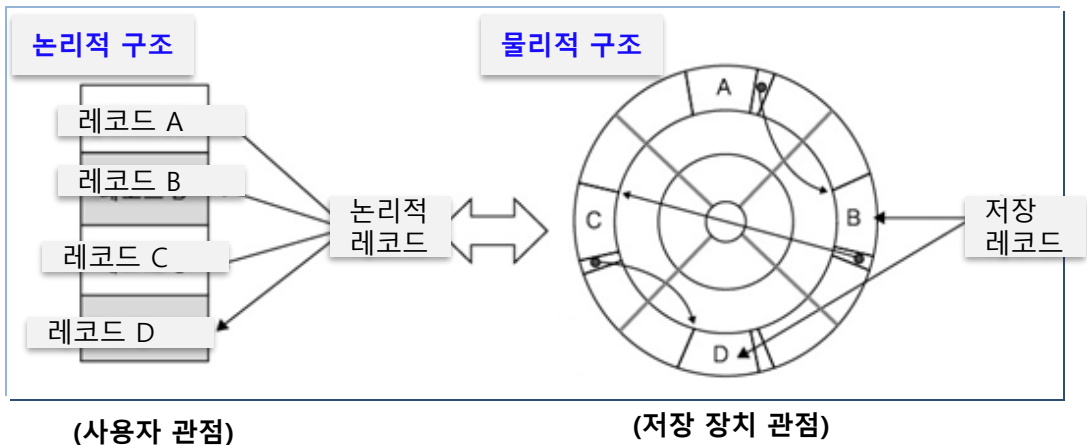
#### 물리적 저장 장치 위에 저장된 DB의 저장 구조

##### 논리적 구조

- ✓ 물리적 저장 장치 위에 저장된 DB의 저장 구조를 **사용자 관점**에서 본 것으로, 사용자가 생각하는 데이터의 구조를 표현한 것

##### 물리적 구조

- ✓ 물리적 저장 장치 위에 저장된 DB의 저장 구조를 **시스템(저장 장치) 관점**에서 본 것으로, 디스크와 같은 저장 장치에 저장되는 데이터의 실제 구조를 표현한 것



[DB의 논리적 구조와 물리적 구조]