

# 응용 SW 기초 활용 기술 part 2

# 데이터베이스의 개요



온라인평생교육원

# ᠍ 학습목표



- 데이터베이스의 이해
- 데이터베이스의 종류



- 데이터베이스, 데이터베이스 관리 시스템, 데이터 시스템을 설명할 수 있다.
- 관계형 데이터베이스와 NoSQL에 대해 설명하고, 응용 프로그램 개발에 적합한 데이터베이스를 선택할 수 있다.

# 🛐 데이터베이스의 이해

# 1 데이터베이스

데이터(Data)

현실 세계에서 수집한 값이나 수치

#### 질적 데이터

종교, 전화번호 등과 같이 수치로 나타낼 수 없는 데이터

#### 양적 데이터

온도, 실적 등과 같이 수치로 나타낼 수 있는 데이터

정보(Information)

필요에 따라 데이터를 의미 있게 사용할 수 있도록 처리한 결과물

#### 데이터와 정보의 예

#### 데이터

설악산의 높이를 측정한 값

정보

설악산을 소개하기 위해 여행 안내 책자에 실린 설악산의 높이

가공되지 않은 상태

데이터를 처리하여 얻은 결과물

수집된 데이터를 효율적으로 저장하는 것이 바로 데이터베이스

# 🔞 데이터베이스의 이해

# 1 데이터베이스

데이터베이스를 이용하지 않을 경우, 특정 대상에 대한 데이터를 수집하고 관리



#### 다양한 응용프로그램 사용

응용프로그램에서 수집한 데이터 입력 및 저장



파일 생성

#### 데이터 관리

해당 응용프로그램을 이용하여 생성한 파일에 접근하여 추가 수집된 데이터를 저장하거나 수정 및 삭제

#### 응용프로그램 별로 데이터 파일을 관리

예 에엘 파일 이용 ➡엑셀 프로그램 사용 워드 파일 이용 ➡워드 프로그램 사용



#53

#### 데이터 교환이 어려움

예

학생 데이터를 저장하고 관리할 때 각각 다른 응용 프로그램을 사용할 때

엑셀 파일로 관리

워드 파일로 관리

기본 데이터

학사 데이터

# 🖅 데이터베이스의 이해

# 1 데이터베이스

하나의 응용프로그램을 이용하여 데이터를 관리



동일한 데이터가 중복되어 저장

동일한 데이터를 수정할 때 하나의 데이터 파일만 수정할 경우 데이터 파일마다 데이터가 달라지는 문제가 발생하여, 일관성 유지가 어려움

> 데이터를 한 곳에 모아 관리한다면, 데이터를 효율적으로 처리할 수 있음

# ᠍ 데이터베이스의 이해

# 1 데이터베이스

#### 데이터베이스

여러 사람들이 공유하고 사용할 목적으로 통합하여 저장한 운영 데이터의 집합

공유 데이터 (Shared Data)	여러 사용자가 필요한 정보를 공동으로 이용할 수 있음
통합 데이터 (Integrated Data)	데이터의 중복을 최소화하여 최적화시킴
저장 데이터 (Stored Data)	여러 사용자가 응용 프로그램을 통해 이용할 수 있도록 컴퓨터가 접근 가능한 매체에 저장함
운영 데이터 (Operational Data)	특정 조직을 운영하는 데 필요하고 지속적으로 유지해야 함



### 🛐 데이터베이스의 이해

### 1 데이터베이스

#### 실시간 접근성

▲ 사용자의 데이터 요구에 대개 몇 초 이내 실시간으로 처리 및 응답이 가능해야 함

#### 동시 공유

■ 동시에 여러 사용자가 다른 데이터 및 같은 데이터를 사용함

### 데이터베이스 의 특징

#### 지속적인 변화

- 현실 세계는 항상 변화하기 때 문에 지속적으로 변화해야 함
- 데이터베이스의 데이터를 동적 으로 삽입・수정・삭제하여 현실 세계를 반영한 정확한 데이터를 유지해야 함

#### 내용에 의한 참조

■ 데이터가 저장된 주소가 아닌데이터의 내용으로 원하는데이터를 참조할 수 있어야 함

#### 데이터베이스의 특징 예

#### 은행에서 계좌 및 고객의 데이터를 통합하여 저장한 데이터베이스의 경우

입금



출금

컴퓨터 및 스마트폰을 이 용하여 <mark>실시간</mark> 확인 가능 고객

- 웹 데이터베이스의 이해
- 2 데이터베이스 관리 시스템

#### 데이터베이스 관리 시스템(DBMS)

- 다수의 사용자들이 데이터베이스 내의 데이터에 접근하여 관리할 수 있도록 해주는 소프트웨어
- DBMS(Database Management System)라고 약칭하여 부름

#### 사용자가 응용 프로그램을 통해 데이터베이스를 쉽게 사용할 수 있도록 해줌

#### DBMS의 주요기능

정의 기능	데이터베이스의 구조를 정의하거나 수정
조작 기능	사용자가 데이터를 처리할 수 있도록 데이터의 검색 · 삽입 · 수정 · 삭제를 수행
제어 기능	여러 사용자가 데이터를 공유해도 정확하고 안전하게 유지

- ᠍ 데이터베이스의 이해
- 2 데이터베이스 관리 시스템

#### 제어 기능을 제공하기 위해 유의할 점

- 1 동시에 실행되는 여러 작업이 올바르게 이루어지도록 해야 함<mark>(동시성 제어)</mark>
- 2 사용 권한이 있는 사용자와 응용 프로그램에게만 데이터 접근을 허용해야 함<mark>(보안)</mark>
- 사용 권한이 있는 사용자나 응용 프로그램에 의해 데이터의 검색 · 삽입 · 수정 · 삭제가 이루어진 후에도 데이터가 정확하고 일관되게 유지되어야 함
- 4 장애가 발생해도 회복이 가능할 수 있도록 해야 함

#### DBMS의 장점

1 데이터의 중복 최소화

5 데이터를 표준화하여 관리

2 데이터 공유

6 데이터 독립성

3 데이터 보안 향상

7 응용 프로그램을 쉽게 개발하고 관리

4 데이터의 무결성 유지

- ☑ 데이터베이스의 이해
- 2 데이터베이스 관리 시스템

#### DBMS의 단점

1 구축이 복잡함

- 3 데이터 파괴에 대한 회복이 어려움
- 장애 발생 시, 데이터 회복을 위해 데이터베이스를 미리 백업해야 함
- 2 많은 비용이 듦
- 4 장애 발생 시 전체 시스템에 영향

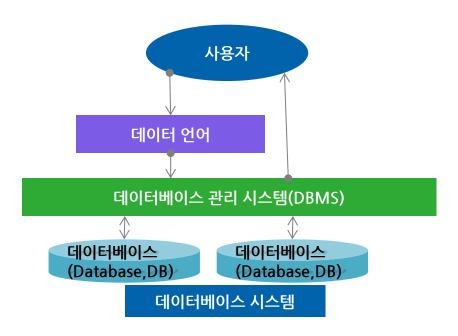
이러한 단점에도 불구하고, 데이터베이스 관리 시스템을 이용함으로써 얻는 이점이 훨씬 많기 때문에 데이터베이스 관리 시스템을 도입하는 경우가 많음

### 🛐 데이터베이스의 이해

### 3 데이터베이스 시스템

데이터베이스 시스템

데이터베이스에 데이터를 <mark>저장</mark>하고, 저장된 데이터를 <mark>관리</mark>하고 <mark>조직</mark>하여 필요한 <mark>정보를 생성</mark>해주는 전체 시스템



### ᠍ 데이터베이스의 종류

# 1 데이터베이스 관리 시스템(DBMS)의 발전

1세대 DBMS

- 1960년~1970년대
- 네트워크 DBMS: 데이터베이스를 노드와 간선을 이용한 그래프 형태로 구성
- 계층 DBMS : 데이터베이스를 트리 형태로 구성

2세대 DBMS

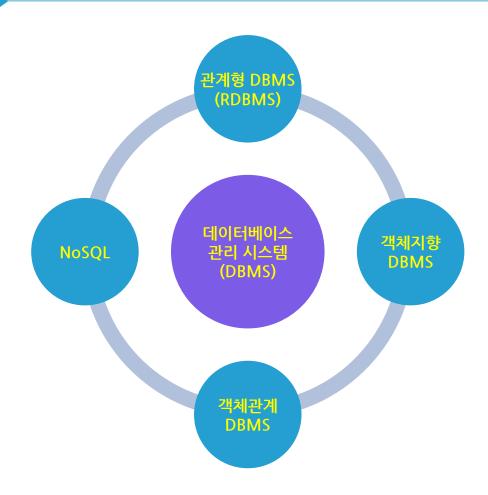
- 1970년대 이후~지금까지 사용
- 관계형 DBMS

3세대 DBMS

- 1980년대 후반에 등장한 객체지향 DBMS
- 1990년대 후반에 등장한 객체관계 DBMS

4세대 DBMS

- 2000년대 이후
- NoSQL



### ᠍ 데이터베이스의 종류

### 2 관계형 데이터베이스

관계형 데이터베이스

IBM 산호세 연구소의 Edgar F. Codd가 제안한 <mark>관계형 모델</mark>을 기반으로 하는 데이터베이스

데이터베이스를 테이블 형태로 구성하여 <mark>이해하기 쉬움</mark> 관계형 데이터베이스 관리 시스템을 RDBMS(Relational Database Management System)이라고 함 대부분의 RDBMS에서는 SQL(Structured Query Language) 언어를 사용하여 데이터 조작을 쉽게 할 수 있을

→ 관계형 데이터베이스 주요 용어

릴레이션(Relation)

데이터베이스에 데이터를 저장하는 기본 단위 속성(Attribute) 또는 필드(Field)

릴레이션(Relation)의 열

레코드(Record) 또는 튜플(Tuple)

릴레이션(Relation)의 행

+ 렐리이션의 예 학생 릴레이션 (테이블) 이름 학부 20153301 홍길동 전기전자공학부 20153302 성춘향 기계공학부

속성(열)

### 🔞 데이터베이스의 종류

### 2 관계형 데이터베이스

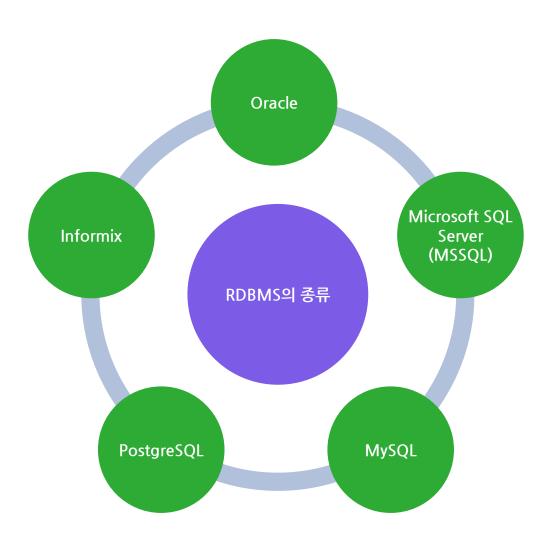
DB 엔진

인터넷 키워드 및 관련 업종 구직·구인 횟수 등 다양한 항목을 자체 수치화해 점수로 만들어 매달 순위를 발표함



관계형 데이터베이스 내의 데이터에 접근하여 관리할 수 있도록 도와주는 관계형 데이터베이스 관리 시스템

약 122개의 제품이 있음



### 🔞 데이터베이스의 종류

### 3-NoSQL

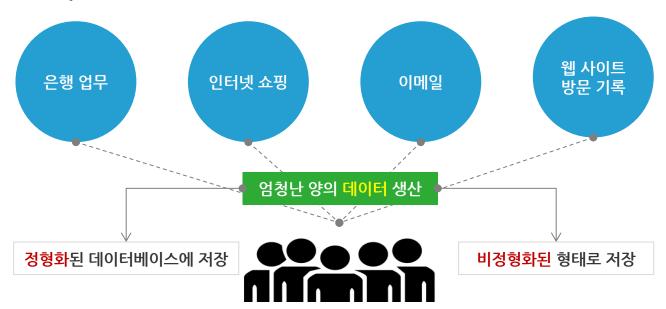
NoSQL

- 관계형 데이터베이스에서 벗어난 방식으로데이터의 저장 및 검색을 위한 기능을 제공하는 데이터베이스
- 비관계형 데이터베이스

NoSQL의 개념이 처음 도입되는 시기에는 비관계형 데이터베이스 사용을 강조하기 위해 Non SQL, Non relational로 불림

#### 현재는 SQL 계열 쿼리 언어를 사용할 수 있다는 사실을 강조하면서 Not only SQL로 불림

#### + NoSQL의 등장배경



### ᠍ 데이터베이스의 종류

# 3 NoSQL

+ NoSQL의 등장배경

빅데이터(Big Data)

저장된 <mark>대용량의 데이터</mark> 자체 혹은 대용량의 데이터로부터 <mark>가치를</mark> 추출하고 결과를 분석하는 기술

기존 RDBMS를 이용하여 대용량의 데이터를 저장하고 관리하는 데 한계를 맞이하게 됨



#### 웹 기반 서비스

저장 데이터 형태가 <mark>지속적으로 변화하고,</mark> 사용자의 데이터 요구가 일관적이지 않고 다양해짐



유연한 형태의 데이터베이스가 필요

#### NoSQL의 사용

- ☑ 많은 데이터 처리를 위한 빅 데이터 분야
- 실시간 웹 애플리케이션 분야

### 🛐 데이터베이스의 종류

# 3 NoSQL

#### NoSQL의 종류

- 1 컬럼(Column) 모델
  - ☑ 데이터의 행이 아니라 데이터의 열을 읽고 쓰는 데 최적화
  - ☑ Cassandra, Hbase 등

온라인 경매로 유명한 eBay에서 사용

- 2 문서(Document) 모델
  - ▼ 주로 JSON 또는 XML 형식의 문서와 같은 반정형(Semi-Structured) 데이터를 저장하도록 설계
  - 문서별로 다른 스키마를 적용
  - MongoDB, Couchbase 등

소셜 네트워크 Facebook에서 사용

- 3 그래프(Graph) 모델
  - 노드(Node)와 간선(Edge)을 사용
  - AllegroGraph, Virtuoso 등
- 4 키-값(Key-value) 모델
  - 키(Key)와 값(Value)이 쌍으로 데이터를 저장 및 관리
  - Redis, Dynamo 등

# ☑ 요점정리

#### 데이터베이스의 이해

- + 데이터(Data)와 정보(Information)
  - 데이터: 현실 세계에서 수집한 값이나 사실
  - 정보: 데이터를 의사 결정에 활용하기 위해 처리한 것
- + 데이터베이스(Database)
  - 여러 사람들이 공유하고 사용할 목적으로 통합하여 저장한 운영 데이터의 집합
- + 데이터베이스 관리 시스템(DBMS)
  - 다수의 사용자들이 데이터베이스 내의 데이터에 접근하여 관리할 수 있도록 해주는 소프트웨어
  - 데이터베이스 시스템은 데이터베이스, 데이터베이스 관리 시스템, 데이터 언어, 사용자로 구성됨

#### 데이터베이스의 종류

- + 관계형 데이터베이스
  - 관계형 데이터베이스는 데이터베이스를 테이블 형태로 구성하여 이해하기 쉬움
  - 관계형 데이터베이스 시스템(RDBMS): Oracle, MSSQL, MySQL 등
- + NoSQL
  - Not Only SQL이라고도 불리는 비관계형 데이터베이스
  - NoSQL 데이터베이스는 데이터 모델에 따라 컬럼 형식 데이터베이스, 문서 데이터베이스, 키-값 데이터베이스, 그래프 데이터베이스 등으로 분류될 수 있으며 종류가 매우 다양함

# ☑ 요점정리

# POINT MANUAL

### + NoSQL의 종류

종류	설명
컬럼(Column) 모델	■ 데이터의 행이 아니라 데이터의 열을 읽고 쓰는 데 최적화
	■ Cassandra, Hbase 등
	■ 주로 JSON 또는 XML 형식의 문서와 같은 반정형 (Semi-Structured)데이터를 저장하도록 설계
문서(Document) 모델	■ 문서별로 다른 스키마를 적용
	■ MongoDB, Couchbase 등
21.71/Va.v.al.va\ [7.6]	■ 키(Key)와 값(Value)이 쌍으로 데이터를 저장 및 관리
키-값(Key-value) 모델	■ Redis, Dynamo 등
	■ 노드(Node)와 간선(Edge)을 사용
그래프(Graph) 모델	■ AllegroGraph, Virtuoso 등