

## 1. 데이터베이스(DataBase)의 정의

조직 내에서 여러 응용시스템과 다수의 사용자들이 공동으로 사용하기 위하여 중복을 배제하고 통합된 환경으로 저장한 데이터의 집합

## 2. 데이터베이스의 특징

- 실시간 접근성
  - ➡ 사용자나 프로그램의 데이터 요구에 실시간으로 처리 · 응답함.
- 계속적인 변화
  - ➡ 삽입 · 삭제 · 갱신을 통해 데이터의 내용이 계속적으로 변화함
- 동시 공유
  - ➡ 목적이 다른 사용자나 프로그램이 동시에 동일한 데이터를 이용 가능함.
- 내용에 의한 참조
  - ➡ 데이터의 탐색 시 주소나 저장위치가 아닌 내용에 의해 참조가 가능함.

## 3. 데이터베이스의 장 · 단점

장점	단점
<ul style="list-style-type: none"> <li>● 데이터 중복의 최소화</li> <li>● 데이터의 독립성 유지</li> <li>● 데이터의 공유</li> <li>● 데이터의 보안성 유지</li> <li>● 데이터의 무결성 유지</li> <li>● 데이터의 일관성 유지</li> </ul>	<ul style="list-style-type: none"> <li>● Overhead가 발생</li> <li>● 운영 비용의 증가</li> <li>● 오류발생시 복구가 어려움</li> </ul>

## 4. 데이터베이스 관리 시스템(DBMS : DataBase Management System)

- ➡ 사용자나 응용 프로그램이 데이터베이스를 공동이용할 수 있도록 관리 · 운영하는 소프트웨어 집단
- ➡ 대표적인 DBMS : Access, dBaseIII+, FoxPro

## 5. 스키마(Schema)

- ❶ 데이터베이스 내의 데이터들의 논리적 구조 및 관계를 기술한 것
- ❷ 데이터베이스 내의 개체, 속성, 관계와 이들간의 제약조건이 표현됨.
- ❸ 3단계 스키마 구조로 이루어짐.
  - 외부 스키마(External Schema)
    - ➔ 응용 프로그래머나 사용자 관점에서의 데이터베이스의 논리적 구조
    - ➔ 서브 스키마 또는 View 라고도 하며, 여러 개의 외부 스키마가 존재할 수 있음.
  - 내부 스키마(Internal Schema)
    - ➔ 물리적 저장장치 관점에서 본 데이터베이스의 구조
    - ➔ 물리적 스키마로 불리우며, 데이터베이스에 대한 접근방법, 물리적 기록매체를 이용하는 정보를 정의함.
  - 개념 스키마(Conceptual Schema)
    - ➔ 기관이나 조직체의 입장에서 본 데이터베이스 전체의 논리적 구조
    - ➔ 논리적 스키마로 불리우며, 데이터베이스에 표현되는 정보의 논리적 구조를 정의함.

## 6. 데이터베이스 언어(DataBase Language)

- ❶ 데이터 정의어(DDL : Data Definition Language)
  - ➔ 데이터베이스의 생성에 사용되는 언어
  - ➔ 데이터의 구조, 처리방식 등을 정의
- ❷ 데이터 조작어(DML : Data Manipulation Language)
  - ➔ 데이터의 삽입 · 삭제 · 수정 · 검색 등 데이터의 실제적인 조작에 사용되는 언어
  - ➔ 응용프로그램과 DBMS 간의 Interface를 제공하는 역할
- ❸ 데이터 제어어(DCL : Data Control Language)
  - ➔ 데이터베이스를 공유하기 위한 데이터 제어를 정의하고 기술하는 언어
  - ➔ 데이터 보안, 데이터 무결성, 데이터 복구에 관한 정의

## 7. 데이터베이스 사용자(DataBase User)

### ● 일반 사용자(End User)

- ➡ 단말기에서 DML이나 질의어를 이용하여 데이터베이스에 접근하는 사용자

### ● 응용 프로그래머(Application Programmer)

- ➡ DML을 삽입하여 만든 응용 프로그램을 통해 데이터베이스에 접근하는 사용자

### ● 데이터베이스 관리자(DBA : DataBase Administrator)

- ➡ DDL이나 DCL을 이용하여 데이터베이스를 정의·제어하고 시스템 전반에 관한 관리·운동을 하는 사람 또는 집단
- ➡ 데이터베이스 관리자의 임무
  - 데이터베이스 시스템의 총괄
  - 데이터의 복구, 보안, 무결성에 대한 관리
  - 데이터베이스 운영에 관한 책임

## 8. 데이터베이스의 논리적 모델

### ● 계층적 데이터베이스(Hierarchical DataBase)

- ➡ 데이터의 상호관계를 계층적 Tree구조로 표현하는 데이터베이스 모델
- ➡ 데이터 상호간에 상하(부모-자식) 관계가 존재
- ➡ 1:N(1:다) 관계를 표현하기에 용이한 데이터베이스 모델

### ● 네트워크 데이터베이스(Network DataBase)

- ➡ 데이터의 상호관계를 Graph구조로 표현하는 데이터베이스 모델
- ➡ 데이터 상호간에는 오너-멤버 관계가 존재
- ➡ N:M(다:다) 관계를 표현하기에 용이한 데이터베이스 모델

### ● 관계형 데이터베이스(Relational DataBase)

- ➡ 데이터의 상호관계를 행과 열로 구성된 Table 구조로 표현하는 데이터베이스 모델
- ➡ 데이터의 추가·삭제·수정·검색 작업이 용이
- ➡ 가장 널리 이용되고 있는 데이터베이스 모델

### TIP

#### 데이터베이스의 개념적 구성

- ➡ 개체
  - : 데이터베이스에 저장될 실세계의 객체나 대상
- ➡ 속성
  - : 개체가 가지는 특성
- ➡ 관계
  - : 개체와 개체간의 연관성 1:1, 1:N, N:M 관계의 형태가 있음.

## 9. 관계형 데이터베이스의 구조

### 테이블(Table)

- 특정 항목에 대한 데이터의 집합으로 데이터베이스의 기본 구성 요소로 릴레이션(Relation)이라고도 함.
- 테이블은 행과 열로 구성되며, 일반적으로 열을 필드(Field), 행을 레코드(Record)라고 함.

### 속성(Attribute)

- 테이블을 구성하는 열, 필드(Field)를 의미하며, 데이터베이스를 구성하는 최소 단위라 할 수 있음.
- 차수(Degree) : 한 Table(=Relation) 내에 포함된 속성(Attribute)의 갯수

### 튜플(Tuple)

- 테이블을 구성하는 행, 레코드(Record)를 의미
- 카디널리티(Cardinality) : 한 Table(=Relation) 내에 포함된 튜플(Tuple)의 갯수

## 10. 키(Key)의 종류

### 후보키(Candidate Key)

- Tuple을 유일하게 식별할 수 있는 특성을 가진 속성이나 속성의 집합
- 유일성과 최소성을 모두 만족함.

### 기본키(Primary Key)

- 후보키 중에서 각 Tuple을 식별하기 위해 선택한 키
- Null값을 가질 수 없음

### 대체키(Alternative Key)

- 후보키 중 기본키를 제외한 나머지 후보키를

### 슈퍼키(Super Key)

- 유일성만 있고 최소성이 없는 속성들의 집합

### 외래키(Foreign Key)

- 다른 테이블을 참조하기 위해 사용되는 속성
- 두 테이블 사이의 관계설정에 이용되는 키

### TIP

#### 유일성(Uniqueness)

릴레이션에 있는 모든 튜플에 대해 모두 다르고 유일한 것

#### 최소성(Minimality)

모든 튜플을 유일하게 식별하는 데 꼭 필요한 속성들만으로 구성되는 것

## 11. 데이터베이스의 구축 순서

- ① 데이터베이스 목적 설정
- ② 데이터베이스에 필요한 테이블 정의
- ③ 테이블에 필요한 필드 정의
- ④ 테이블간의 관계 설정
- ⑤ 데이터 입력

MEMO