**데이터베이스(Database, DB)란?**

여러 사람이 공유할 목적으로 체계화해 통합, 관리하는 데이터의 집합. 스프레드 시트와 기능은 거의 유사하지만 컴퓨터 언어로 제어가 가능하며 앱이나 웹을 통해 공유가 가능하여 전세계 누구나 데이터베이스에 접근하고 편집이 가능하다는 장점이 있음.

**데이터베이스의 특징**

1. 실시간 접근성(Real-Time Accessibility): 실시간 처리에 의한 응답이 가능해야 한다.

2. 계속적인 변화(Continuous Evolution): 새로운 데이터의 삽입(Insert), 삭제(Delete), 갱신(Update)로 항상 최신의 데이터를 유지한다.

3. 동시 공용(Concurrent Sharing): 다수의 사용자가 동시에 같은 내용의 데이터를 이용할 수 있어야 한다.

4. 내용에 의한 참조(Content Reference): 데이터베이스에 있는 데이터를 참조할 때 사용자의 요구에 따른 데이터 내용으로  데이터를 찾는다.

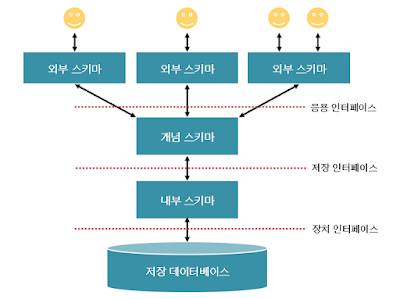
**데이터베이스(Database, DB) 용어**

* 관계형 데이터베이스: 여러개의 테이블이 특정 관계로 이루어져있는 구조를 가진 데이터 베이스.
* 엔티티(Entity): 사람, 장소, 사물, 사건 등과 같이 독립적으로 존재하면서 고유하게 식별이 가능한 실세계의 객체. (ex) 과목 코드가 F035 인 자료구조 등.
* 엔티티 집합(Entity Set): 동일한 속성을 가진 엔티티들의 집합. 엔티티 집합에 속한 요소들이 여러 엔티티 집합에 속할 수도 있다.

**스키마(Schema)**

데이터 베이스의 구조를 전반적으로 기술한 것. 구체적으로 데이터베이스를 구성하는 데이터 개체(Entity), 속성(Attribute), 관계(Relationship) 등을 정의 한 것을 말한다. 사용자의 관점에 따라 외부 스키마, 개념 스키마, 내부 스키마로 구분한다. DBMS는 외부 스키마에 명세된 사용자의 요구를 개념 스키마 형태로 변환하고, 이를 다시 내부 스키마 형태로 반환한다.

* 외부 스키마(사용자 뷰): 사용자의 입장에서 정의한 데이터 베이스의 논리적 구조. 데이터들을 어떤 형식, 구조, 화면을 통해 사용자에게 보여줄 것인가에 대한 명세를 말하며 하나의 데이터베이스에는 여러개의 외부 스키마가 있을 수 있다. 일반 사용자는 SQL을 이용하여 DB를 쉽게 사용할 수 있다. 응용 프로그래머는 C, 자바 등의 언어를 사용하여 DB에 접근한다.
* 개념 스키마(전체적인 뷰): 데이터베이스의 전체적인 논리적 구조. 모든 이용자가 필요로 하는 데이터를 총합한 조직 전체의 데이터 베이스로 하나만 존재한다. 개체 간의 관계와 제약조건, 데이터 베이스의 접근 권한, 보안 등에 관한 명세를 나타낸다. 데이터 베이스 관리자에 의해서 구성된다.
* 내부 스키마: 물리적 저장장치의 입장에서 본 데이터베이스 구조. 실제로 데이터베이스에 저장될 레코드의 물리적인 구조, 저장 데이터 항목의 표현 방법, 내부 레코드의 물리적 순서 등을 나타낸다.



**표(Table) 관련 용어**



* Relation(= Table): 관계형 데이터 베이스에서 정보를 구분하여 저장하는 기본 단위.
* Tuple(=Record): 테이블에서 행을 의미. 튜플은 릴레이션에서 같은 값을 가질 수 없다. 튜플의 수는 카디날리티(Cardinality)라고 한다.
* Attribute( =Field): 테이블에서 열을 의미. 같은 말로는 칼럼이라고도 하며 어트리뷰트의 수는 디그리(Degree)라고도 한다.
* 식별자(Identifier): 여러개의 집합체를 담고있는 관계형 데이터 베이스에서 각각의 구분할 수 있는 논리적인 개념
  + 특성
    - 유일성: 하나의 릴레이션에서 모든 행은 서로 다른 키 값을 가져야 한다.
    - 최소성: 꼭 필요한 최소한의 속성들로만 키를 구성해야 한다.

**Key**

데이터베이스에서 조건에 만족하는 튜플을 찾거나 순서대로 정렬할 때 튜플들을 서로 구문할 수 있는 기준이 되는 어트리뷰트.

* 후보키(Cardidate Key)
  + 릴레이션을 구성하는 속성들 중에서 튜플을 유일하게 식별하기 위해 사용하는 속성들의 부분집합.
  + 기본 키로 사용할 수 있는 속성들을 의미.
  + 모든 릴레이션에는 반드시 하나 이상의 후보키가 존재한다.
  + 유일성과 최소성을 만족시켜야 한다.
* 기본키(Primary Key)
  + 후보키 중에서 선택한 Main Key이다.
  + 한 릴레이션에서 특정 튜플을 유일하게 구별할 수 있는 속성.
  + null 값을 가질 수 없다.
  + 동일한 값이 중복되어 저장될 수 없다.
* 슈퍼키(Super Key)
  + 한 릴레이션 내에 있는 속성들의 집합으로 구성된 키로서 릴레이션을 구성하는 모든 튜플들 중 슈퍼키로 구성된 속성의 집합과 동일한 값을 나타나지 않는다.
  + 릴레이션을 구성하는 모든 튜플에 대해 유일성을 만족시키지만, 최소성을 만족시키지 못한다.
  + (ex) 학번 + 주민번호를 사용하여 슈퍼키를 만들면 유일성은 만족하지만, 학번이나 주민번호 하나만 가지고도 다른 튜플들을 구분할 수 있으므로 최소성은 만족시키지 못한다.
* 외래키(Foregin Key)
  + 어떤 릴레이션간의 기본키를 참조하는 속성. 테이블들 간의 관계를 나타내기 위해서 사용된다.
  + 다른 릴레이션의 기본 키를 그대로 참조하는 속성의 집합을 의미.
  + 외래키가 되는 속성과 기본키가 되는 속성의 이름은 달라도 되지만, 외래키의 속성의 도메인과 참조되는 기본키 속성의 도메인은 반드시 같아야 한다. 도메인이 같아여 연관성 있는 투플을 찾기 위한 비교 연산이 가능하기 때문이다.
  + 외래키가 없으면 2013년 1월1일에 제품을 주문한 고객의 이름을 검색할 수 없지만, 외래키가 있어서 이 날짜에 주문한 고객이 김현준임을 쉽게 알 수 있다.
  + 외래키는 기본키를 참조하지만 기본키가 아니기 때문에 null값을 가질 수 있고, 서로다른 투플이 같은 값을 가질 수 있다.



**관계 데이터 모델의 제약조건**

* 기본키를 구성하는 모든 속성은 null 값을 가질 수 없다.
* 외래키는 참조할 수 없는 값을 가질 수 없다.

**데이터베이스 언어 종류**

* DML, Data Multipulation Language (데이터 조작어): 저장된 데이터를 실질적으로 처리하는데 사용하는 언어. 데이터 베이스의 생성 및 변경, 제거
* DDL, Data Definition Langauge(데이터 정의어): 데이터 베이스를 정의하는 언어. 데이터베이스 안의 값들을 변경, 수정, 입력.
* DCL, Data Control Language(데이터 제의어): 데이터베이스에 접근하거나 객체에 권한을 주는 역할. 데이터 베이스의 접속 권한 등을 수정.