**데이터 베이스(DATABASE)란?**

자료를 많이! 모은것!  
 하지만 자료라고 해서 아무런 연관 없는 자료를 그냥 모으면 그건 유용하지 않다.  
 데이터 베이스에 있는 자료는 서로 연관성이 있는 자료들이어야 한다.  
게임이라면 플레이어 데이터 플레이어 아이템 유저 패스워드 같은 식으로 데이터를 연관시켜 모아놓은 것을 의미한다.

**데이터 베이스 관리시스템(DBMS : DATABASE MANAGEMENT SYSTEM)**

데이터 베이스와 더불어 많이 쓰이는 용어로 데이터 베이스 관리 시스템이라는 용어가 있다. 데이터베이스와 데이터 베이스 관리 시스템은 다른 개념을 가리킨다.  
 DB는 말그대로 데이터의 집합 DBMS는 그것을 편하게 관리하기 위한 시스템이다.

**응용(또는 사용자) 프로그램**

그런데 각 기관에서 데이터베이스를 관리하려면 데이터베이스 관리 시스템에 있는 프로그램을 활용하여, 그 프로그램등에 맞춰서 또다른 관리 프로그램을 만들어야 한다.  
 게임을 생각해보면 유저 패스워드가 있고 그 데이터를 넣고 뺄 수 있는 DBMS가 있다고 해서 유저패스워드를 통한 로그인 시스템까지 지원해주는 것은 아니다.  
 그것은 또 우리가 만들어야 한다. 이를 사용자 프로그램이라고 한다.

**데이터 베이스 시스템이 파일관리보다 좋은 점.**

**1) 자료의 중복(data redundancy)와 자료의 불일치(data inconsistency)**

파일 처리 접근 방식을 쓰면, 꼭 같은 정보가 여러군데 되불이 될 수 있다. 보기를 들어 게임에서 아이템데이터가 변경되었다고 치자. 그런데 유저 아이템데이터는 유저의 세이브파일에도 들어 있고 게임의 로직 아이템파일에도 들어있으며, 각 NPC데이터 파일에도 들어있다. 그럼 결국 세가지 파일을 다 바꿔야 할것이다. 만약 하나라도 이 데이터가 바뀌게 된다면 세 개의 파일을 모두 바꿔줘야 한다. 혹은 그러다 어떤 파일을 바꾸지 못하는 경우가 생기는데 이런 경우 자료 불일치라고 한다. 또한 파일로 관리할 시 이렇게 세 개의 파일이 모두 아이템 데이터를 가지는 경우가 생기는데 이는 같은 자료를 중복적으로 가지게 되는 경우이다.

**2) 자료처리의 불편함.**

파일 처리 접근 방식은 보통 고급언어에서 응용프로그램을 짜야 하는데 이는 고급언어에 대한 훈련이 필요하다. 그런데 데이터베이스는 거진 공통적인 명령을 사용하고 고급 프로그래밍 언어보다 데이터를 처리하는데 필수적인 기능만을 모아놓았다. 프로그래밍언어로 파일 안의 데이터를 처리하기 위한 명령어는 복잡하거나 상황이나 언어 플랫폼에 따라서 달라지는 반면 데이터베이스는 훨씬 편하게 사용이 가능하다.

**4) 자료의 무결성(data integrity)**

자료의 무결성이란 자료에 잘못이 없어야 한다는 것을 말하는데, 자료의 불일치를 포함하여 여러가지 경우에 자료의 무결성이 깨질 수 있다. 책에 따라서 자료의 불일치와 자료의 무결성에 풀이가 다를수 있는데 일반적으로 자료의 불일치는 자료의 무결성을 깨게 된다. 하지만 자료의 불일치가 아니더라도 자료의 무결성을 깨는 경우가 많다.

**5) 원자성(atomicity)**

트랜잭션 개념

**6) 동시접근 문제(concurrent acces anomalies)**

어떤 파일을 동시에 2개의 게임 쓰레드에서 수정하려고 한다면 파일 처리 방식에서는 자료의 값이 잘못 될수가 있다. 하지만 데이터 베이스에서는 동시 접근 제어(concurrency control) 기능이 있어서, 동시접근 하면 순서를 지켜서 처리하도록 보장한다.

**7) 보안문제(security problems)**

파일처리 접근 방식에서는 자료에 대한 보안에 어려움이 있다 수많은 에디터를 통해서 파일을 에디팅하는 모습을 볼 수 있을 것이다. 하지만 서버와 DB를 통한 관리는 자료를 모두 한군데에 저장하고 사용자의 권한등을 자세히 정함으로서 자료의 보안을 쉽게 유지할 수 있다.

**데이터 모델링의 특징**

추상화: 현실세계를 일정한 공통적인 특징을 뽑아 간략하게 표현  
단순화: 현실세계를 약속된 규약이나 제한된 표기법과 언어로 표현  
명확화: 누구나 이해하기 쉽게 애매모호함을 제거하고 정확하게 현상을 기술

**데이터 모델링의 관점**

데이터 : 비지니스 프로세스에서 사용되는 데이터

프로세스 : 비지니스 프로세스에서 수행되는 작업

데이터와 프로세스 : 데이터와 프로세스간의 관계.

로그인 프로세스를 위해서 회원 데이터가 필요하다.

회원정보수정 프로세스가 일어나면 회원 데이터가 변경된다.

**데이터 모델링의 단계**

**개념적 모델링** : 기업의 비지니스 프로세스를 분석하고 기업 전체에 대한   
데이터 모델링 수행 업무 관점 모델링, Entity와 속성을 도출 개념적 ERD 작성

**논리적 모델링** : 식별자를 도출하고 필요한 모든 관계(릴레이션) 정의  
 정규화를 수행하여 데이터 모델의 독립성 확보

**물리적 모델링** : 데이터베이스를 실제로 구축

**데이터 독립성 :** 데이터베이스의 구조 변화로 인한 영향을 프로그램에 미치지 않도록 하는 것DBMS의 궁극적인 목표로 응용프로그램이 데이터에 종속되지 않는 것을 의미한다

독립성을 확보하는 방법 : 3층스키마, 정규화

**3층 스키마 구조(피피티 보여줌)**

**offset** : 상대주소: 오프셋(offset)은 일반적으로 동일 오브젝트 안에서 오브젝트 처음부터 주어진 요소나 지점까지의 변위차를 나타내는 [정수형](https://ko.wikipedia.org/wiki/%EC%A0%95%EC%88%98%ED%98%95)이다.

이를테면, 문자 A의 배열이 abcdef를 포함한다면 'c' 문자는 A 시작점에서 2의 오프셋을 지닌다고 할 수 있다.

[어셈블리어](https://ko.wikipedia.org/wiki/%EC%96%B4%EC%85%88%EB%B8%94%EB%A6%AC%EC%96%B4)와 같은 저급 프로그래밍 언어에서 오프셋은 **상대 주소**(relative address)로 부른다.

**스키마란 ?** : **데이터베이스 스키마**(database schema)는 [데이터베이스](http://ko.wikipedia.org/wiki/%EB%8D%B0%EC%9D%B4%ED%84%B0%EB%B2%A0%EC%9D%B4%EC%8A%A4)에서 [자료](http://ko.wikipedia.org/wiki/%EC%9E%90%EB%A3%8C)의 구조, 자료의 표현 방법, 자료 간의 관계를 정의한 것을 말하는 [전산학](http://ko.wikipedia.org/wiki/%EC%A0%84%EC%82%B0%ED%95%99) 용어

**외부스키마** : 사용자 관점 업무상 관련이 있는 데이터   
 관련 데이터베이스의 VIEW를 표시(업무에 따라 볼 수 있는 데이터가 다름)  
 응용 프로그램이 접근하는 데이터베이스를 정의한다.

**개념 스키마** : 설계자 관점, 사용자 전체 집단의 데이터베이스 구조  
 전체 데이터베이스 내의 규칙과 구조를 표현   
 EX) 데이터 접근 권환, 엔터티 관계, 보안 등등

**내부 스키마** : 개발자 관점, 데이터베이스의 물리적 저장구조.

논리적 독립성 : 개념스키마가 변경되어도 외부스키마 즉 어플리케이션에는 영향이 없다. VIEW 예시

물리적 독립성 : 내부스키마가 변경되어도 개념스키마나 외부스키마에는 영향이 없다. INDEX 예시 : 검색속도를 빠르게 하기 위해 인덱스를 생성한다고 해서 개념스키마에 변화가 생기는 것은 아니다.

**엔터티(Entitiy) :** Thomas Bruce : 정보가 저장될 수 있는 장소, 사람, 사건, 개념, 물건 등이다.

엔터티의 특징 : 엔터티는 반드시 속성을 가지고 있다.  
 두 개 이상의 인스턴스가 있어야 한다.  
 엔터티는 인스턴스를 식별할 수 있는 식별자가 있다.  
 엔터티는 다른 엔터티와 최소 한 개 이상 관계가 있어야 한다.  
 엔터티는 연관이 있는 데이터들의 집합이다.

**엔터티 분류**

1) 유/무형

유형엔터티 : 물리적 형태가 존재하는 것

ex) 고객, 강사, 사원

무형엔터티 : 물리적 형태가 존재하지 않는, 개념적으로 사용되는 것  
 ex) 거래소 종목, 코스닥 종목 등등

사건엔터티 : 비지니스 프로세스 과정에서 생성되는 엔터티

ex) 환불건, 수수료 등등

2) 발생 시점

기본엔터티 : 독립적으로 생성되는 것.

ex) 고객, 상품, 부서

중심엔터티 : 기본엔터티로 부터 발생되고 행위엔터티를 생성하는 것

ex)계좌, 고객이 계좌를 개설한다. 주문, 고객이 상품을 주문한다.

행위엔터티 : 중심엔터티에 의해 생성되는 것,

ex) 주문 이력, 채결 이력 등등

속성 : 엔터티의 항목

속성은 더 이상 분리되지 않는 단위이다. 인스턴스의 구성요소이다.(인스턴스는 속성들의 값의 집합이다.)

**함수적 종속** : X -> Y : Y는 X에 함수적 종속이다.   
**A이면 B이고 동시에 A이면 C일 수 없지만, B이면 반드시 A인 것은 아니다.**

완전함수적종속 : 종속자가 기본키에만 종속되는 경우. 기본키가 여러 개의 속성으로 이루어져 있다면 여러개의 속성들에 종속되는 경우

부분함수적종속 : 종속자가 기본키가 아닌 다른 속성에 종속되거나 기본키를 구성하는 여러 속성 중 일부 속성에 종속되는 경우

이행적 함수 종속X→Y, Y→Z :   
X를 통해 Y를 알 수 있고, Y를 통해 Z를 알 수 있는 경우

속성의 종류

단일속성 : 하나의 의미로 구성된 것으로 회원ID, 이름 등이다.

복합속성 : 여러 개의 의미가 있는 속성. 주소(도,시,군 등으로 분해가능)

다중값속성 : 하나의 속성이 여러개의 값을 가지는 경우. 분해해야 한다.

기본속성 : 비지니스 프로세스에서 도출되는 본래의 속성

설계속성 : 데이터모델링 과정에서 발생되는 속성(유일한 값을 부여한다)

ex) 상품코드, 지점코드

파생속성 : 다른 속성에 의해 만들어지는 속성

ex) 평균, 합계 등등

**관계**

존재관계와 행위관계로 분리된다.

존재관계 : 두 엔터티가 존재여부(속함)의 관계에 있는 것.

ex) 고객-관리점, 학생-반, 상품-카테고리

행위관계 : 두 엔터티간의 행위가 있는것

ex) 고객 – 계좌  
 고객은 계좌를 만들지, 계좌에 속하지 않는다.

**관계차수**

1:1관계

완전 1:1관계 : 엔터티에 행이 하나 있을 때 다른 엔터티에 관계맺는 행이 반드시 하나 존재하는 관계

선택적 1:1관계 : 엔터티에 행이 하나 있을 때 다른 엔터티에 관개맺는 행이 하나이거나 없을 수도 있다.

1:N관계 :

엔터티에 행이 하나 있을 때 다른 엔터티에 관개 맺는 행이 여러   
있는 관계. 고객-아이디 (넥슨 아이디 3개까지 가능)

M:N관계 :

두개의 엔터티가 서로 여러개의 관계를 가지고 있는 경우  
 학생은 여러 과목을 수강하고 한 과목은 여러 학생이 듣는다.  
 JOIN시 카텐시아곱이 발생함으로 이 경우 1:N로 해소해 줘야 한다.

필수적 관계 : 반드시 하나는 존재해야 하는 관계 |로 표시

선택적 관계 : 없을 수도 있는 관계 O로 표시

**식별관계** : 부모테이블의 키를 자식테이블에서도 키로 사용하는 경우

부모테이블의 키를 통해 자식테이블을 특정할 수 있다. 실선으로 표현

**비식별관계** : 부모테이블의 키를 자식테이블이 일반 컬럼으로 사용하는 경우 : 부모테이블의 키를 통해 자식테이블을 특정할 수 없다.   
점선으로 표현

**식별자 : 유일성을 만족하는 속성(중복안됨)**

1.슈퍼키 : 유일성은 만족하지만 최소성은 만족안함

2.후보키 : 유일성과 최소성 만족

3.기본키 : 후보키들 중에서 엔터티를 대표하는 키

4.대체키 : 후보키 중에서 기본키 빼고 나머지들.

**식별자 종류**

**1. 대표성 여부**

주식별자 : 기본키  
보조식별자 : 대체키

**2.생성여부**내부식별자 : 엔터티 내부에서 생성되는 식별자  
외부식별자 : 관계에 의해 만들어지는 식별자.

**3. 속성의 수**단일식별자 : 속성이 하나  
복합식별자 : 두개 이상의 속성

**4. 대체 여부**

본질식별자 : 비지니스프로세스 상에서 만들어지는 식별자

인조식별자 : 인위적으로 만들어내는 식별자(상품코드)

**정규화**

데이터의 중복을 제거하고 데이터 모델의 독립성을 확보하기 위한 방법  
정규화를 통해서 이상현상(Anomaly)를 해결할 수 있다.  
제 1정규화 부터 제 5정규화까지 있지만 실질적으로는 제 3정규화만 수행한다.

정규화 절차

제 1정규화 : 속성의 원자성을 확보한다.  
 기본키를 설정한다.

제 2정규화 : 기본키가 2개 이상의 속성으로 이루어진 경우 부분함수종속성을 제거한다.

제 3정규화 : 기본키를 제외한 컬럼 간의 종속성을 제거한다.   
즉 이행함수종속성을 제거한다.

BCNF : 후보키가 기본키를 종속시키면 분해한다.

제 4정규화 : 여러 칼럼들이 하나의 컬럼을 종속시키는 경우 분해하여 다중값 종속을 제거한다.

제 5정규화 : 조인에 의해서 종속성이 발생되는 경우 분해한다.