1장 데이터베이스와 SQL

YBIGTA Engineering Team 14기 이용하

INDEX

1. 데이터베이스

데이터베이스

데이터(Data)

: 컴퓨터 안에 기록되어 있는 숫자

데이터베이스(Database)

- 넓은 의미 : 컴퓨터 안에 기록된 모든 것. 데이터의 집합

- 일반적으로 통용되는 의미 :
  특정 데이터를 확인하고 싶을 때 간단하게 찾아낼 수 있도록 정리된 형태의 데이터

- 데이터베이스 내의 데이터는 영구적으로 보존되어야 함.
  따라서 하드디스크나 플래시메모리(SSD) 등 비휘발성 저장장치에 저장

DB와 DBMS

DB(Database)

: 저장장치 내에 정리되어 저장된 데이터의 집합

DBMS(Database Management System)

: DB 관리 시스템. DB를 효율적으로 관리하는 소프트웨어

DBMS가 필요한 이유

1 생산성

- 시스템 개발 과정에서의 생산성 향상 도모
- 데이터 검색, 추가, 삭제, 갱신과 같은 기본 기능 제공
- 시스템 구축 과정의 비용 측면에서 효율적

2 기능성

- DB를 다루는 기능을 많이 제공함
- DB 관리 기능을 유저가 확장할 수도 있어 유연한 시스템 개발 가능

3 신뢰성

- 대규모 DB는 많은 요청에 대응하기 위해 하드웨어를 여러 대로 구성해 신뢰성을 높이는 동시에 성능 향상을 꾀하기도 함
- 일부 DBMS는 컴퓨터 여러 대를 두고, SW를 통해 확장성(Scalability)과 부하 분산(Load balancing)을 구현
  => '클러스터 구성' 또는 '스케일 아웃'
- 데이터베이스의 데이터를 다른 저장장치로 내보내거나(export), 데이터베이스 안에 데이터를 집어넣는(import) 기능을 통해 데이터베이스를 간단하게 백업할 수 있음

SQL

SQL

- DBMS와의 대화에 필요한 것
- DB 중에서도 관계형 데이터베이스 관리 시스템(RDBMS : Relational Database Management System) 조작할 때 사용
- 현재 ISO 등에 의해 표준화가 진행되어 C언어나 Java 등과 마찬가지로 표준 언어 (※ 표준 언어란 곧 생산성을 향상시킬 수 있다는 의미)

SQL 명령의 종류

1 DML(데이터 조작)

- Data Manipulation Language
- 데이터베이스에 새롭게 데이터를 추가하거나 삭제하거나 내용을 갱신하는 등, 데이터를 조작할 때 사용. SQL의 가장 기본이 되는 명령 셋(set)

2 DDL(데이터 정의)

- Data Definition Language
- 데이터베이스는 '데이터베이스 객체(object)'라는 데이터 그릇을 이용하여 데이터를 관리하는데, 이 같은 객체를 만들거나 삭제하는 명령어

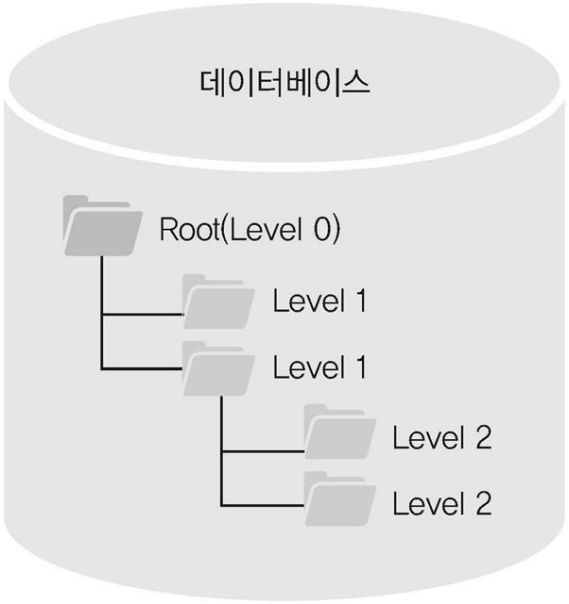
3 DCL(DB 제어)

- Data Control Language
- 데이터를 제어하는 명령어. 트랜잭션을 제어하는 명령과 데이터 접근권한을 제어하는 명령 포함

2. 다양한 데이터베이스

데이터 저장 방법에 따라 분류 / 오래된 순서로 정리

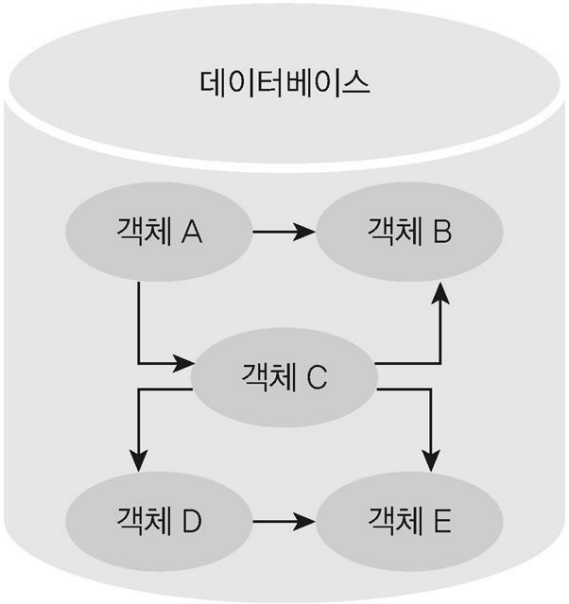
계층형 데이터베이스



: 역사가 오래된 DBMS로, 폴더와 파일 등의 계층 구조로 데이터를 저장하는 방식
ex) 하드디스크, DVD 파일시스템

- 현재 DBMS로서 채택되는 경우는 많지 않음

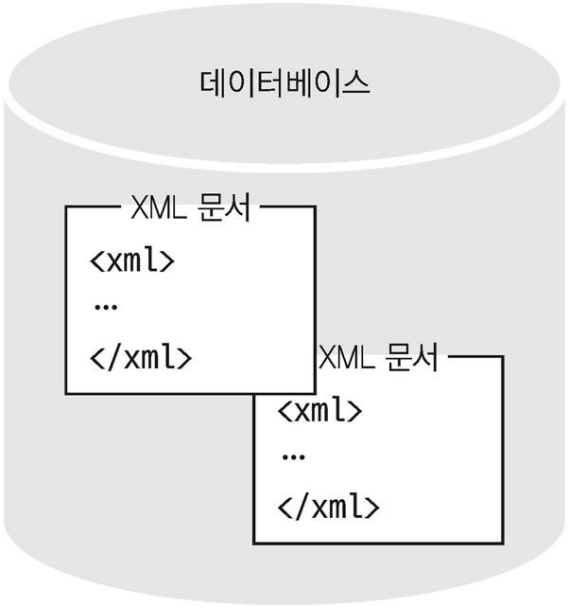
객체지향 데이터베이스



: 객체(object) 그대로를 DB의 데이터로 저장하는 것

- Java와 C++같은 객체지향 언어의 객체지향 프로그래밍에 적합한 DB

XML 데이터베이스



: XML 형식으로 기록된 데이터를 저장하는 DB

- XML : 태그를 이용해 마크업 문서를 작성할 수 있게 정의한 자료 형식
- SQL 명령 대신 XQuery라는 전용 명령어를 사용해 데이터 검색

키-밸류 스토어(KVS)

: 키와 그에 대응하는 값(value)이라는 단순한 형태의 데이터를 저장하는 DB

- NoSQL(Not Only SQL) 이라는 슬로건으로부터 생겨난 DB로,
  열 지향 데이터베이스라고도 불림

RDBMS 사용 시스템

- main frame(대형 범용기기)부터 소형 워크스테이션까지 널리 쓰임
- 휴대전화에도 RDMS가 내장

  ex) 구글이 개발한 모바일 OS 안드로이드에는 ‘SQLite’라는 작은 하드웨어용으로 경량화한,
  임베디드 시스템에 자주 쓰이는 RDMS가 표준으로 기본 내장

데이터베이스 제품

시중에 판매되거나 오픈소스로 자유롭게 다운로드할 수 있는 유명한 제품 / 오래된 순서로 나열

1 Oracle

- 오라클에서 개발한 RDBMS
- 현재 가장 많이 쓰이는 RDBMS 중 하나로, RDBMS의 사실상 표준이라고 해도 문제없을 정도로 유명

2 DB2

- IBM에서 개발한 RDBMS
- Oracle이 유닉스 워크스테이션 중심이었던 것과 달리, DB2는 발표된 이래 한동안 IBM 컴퓨터에서만 구동. 이후 유닉스, 윈도우 등의 플랫폼에서도 구동 가능하게 되었지만 시장 점유율 확대할 수 없었음

3 SQL Server

- 마이크로소프트가 개발한 DBMS
- 윈도우 플랫폼에서만 동작

데이터베이스 제품

4 PostgreSQL

- 오픈소스 커뮤니티가 개발한 RDBMS
- 실험적인 기능이 포함되어 있거나 독특한 구조를 가지기도 함

5 MySQL

- 오픈소스 커뮤니티가 개발한 RDBMS
- 첫 발매 당시 경량 데이터베이스라는 점을 강조해 필요한 최소한의 기능만을 갖추었으나, 현재 기능 확장

6 SQLite

- 오픈소스 커뮤니티가 개발한 RDBMS
- 임베디드 시스템에 자주 쓰이는 작은 RDBMS

SQL의 방언과 표준화

각 DB 제품별로 기능 확장이 이루어지는 과정에서

‘비슷한 조작을 실행하더라도 서로 다른 명령어가 필요한’ 상황 발생

특정 DB 제품에만 통용되는 고유 방언 탄생

SQL의 방언과 표준화

- 키워드 생략

  ex) DELETE (FROM) : 일부 데이터베이스 제품에서 FROM 생략 가능

- 외부결합

  ex) Oracle : (+) / SQL Server : *=

방언 대신 표준 SQL을 사용하는 편이 좋다!

3. 데이터베이스 서버

클라이언트/서버 모델

: 사용자 조작에 따라 요청을 전달하는 '클라이언트'와 해당 요청을 받아 처리하는 '서버'로 소프트웨어를 나누고, 복수의 컴퓨터 상에서 하나의 모델을 구현하는 시스템

클라이언트/서버 모델

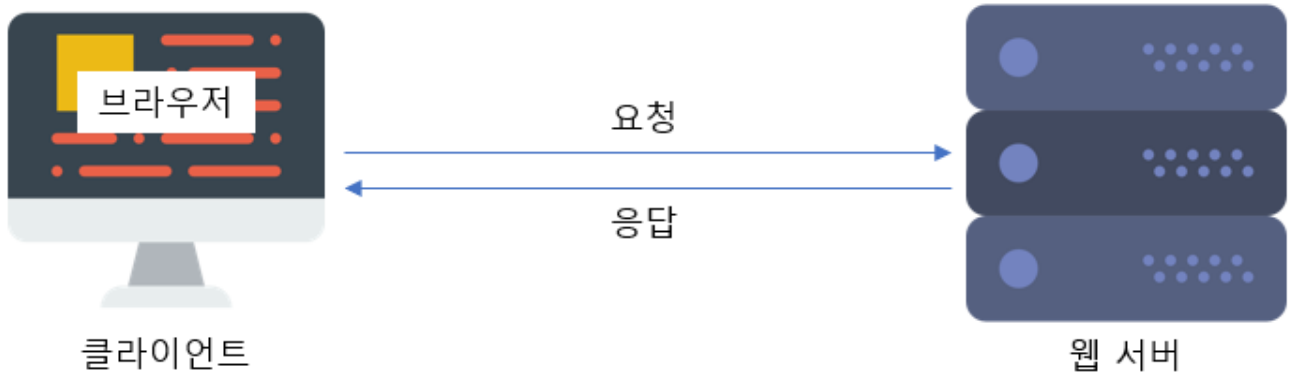
RDBMS는 복수와 클라이언트가 보내오는 요청에 응답할 수 있도록 클라이언트/서버 모델로 동작

- 일반적인 RDBMS는 네트워크 상에 하나의 서버를 두고 독점해 사용
- 한 대의 PC에 클라이언트와 서버 모두를 실행시켜 운용 가능

웹 시스템에서의 클라이언트/서버

웹 시스템 : 브라우저와 웹 서버로 구성되는 클라이언트/서버 모델의 시스템

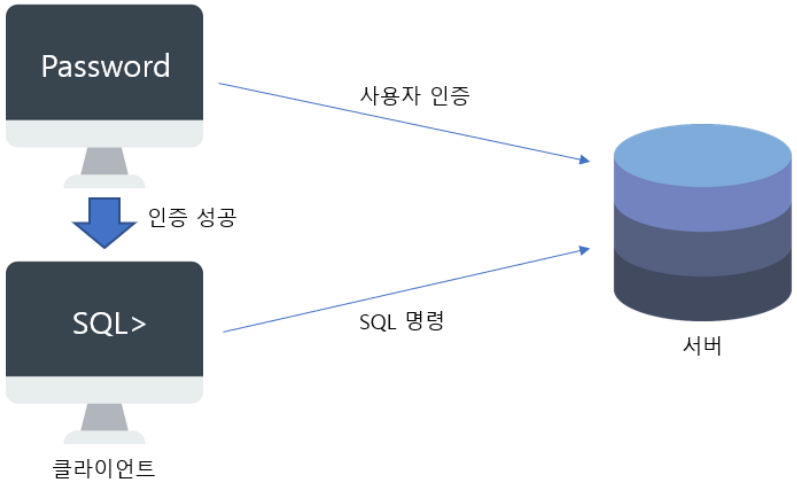
- 클라이언트 기능을 하는 브라우저가 사용자가 지정한 URL과 연결된 웹 서버에 요청(request)을 보내면, 웹 서버에서는 그에 맞는 데이터를 네트워크를 통해 전송 후 응답(response)을 클라이언트로 반환
- 실제 웹에서는 요청과 응답이 되풀이되면서 웹 페이지가 표시됨



RDBMS의 클라이언트/서버

RDBMS도 웹 시스템과 마찬가지로 클라이언트/서버 모델로 시스템 구성

- 하지만 단순히 요청과 응답을 되풀이하는 것은 아님



- 웹 시스템에는 없었던 사용자 인증 필요.
- 사용자 별로 데이터베이스 접근을 제한할 수 있기 때문에 사용자 ID와 비밀번호를 이용한 사용자 인증 필요

웹 애플리케이션의 구조

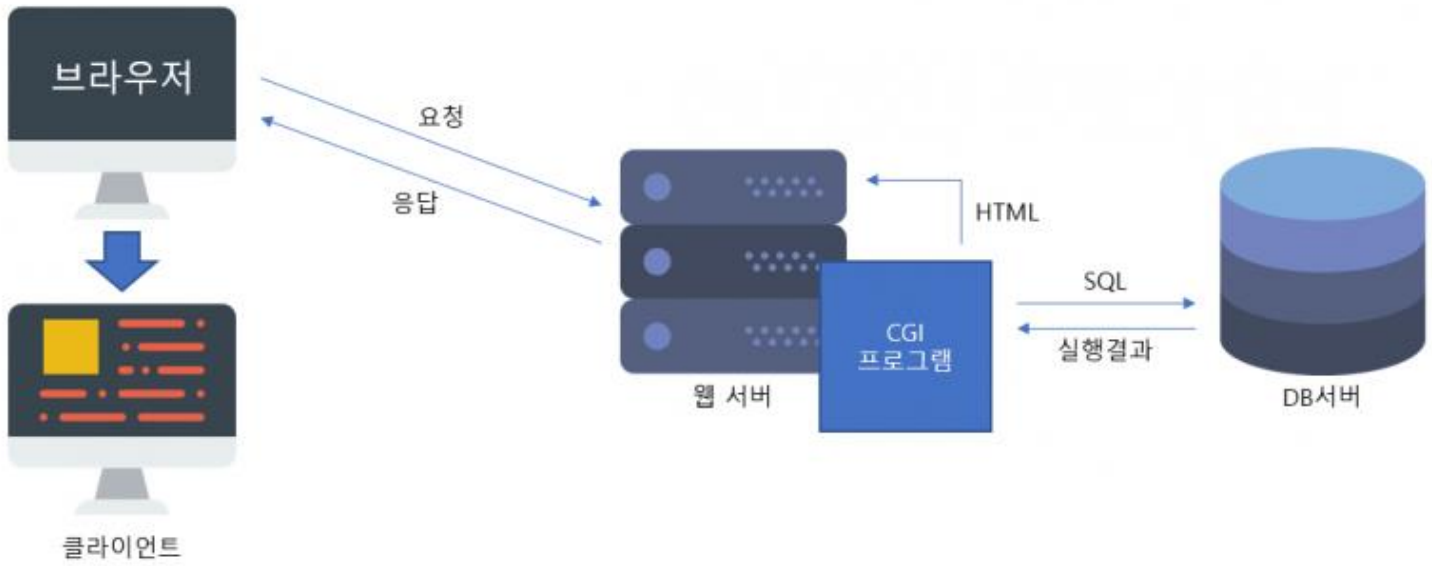
웹 애플리케이션 : 일반적으로 '웹 서버 + 데이터베이스 서버'

- 웹 사이트가 정적인 HTML만으로 구성되어 있다면 웹 서버만으로 시스템 구축이 가능하지만,
  웹 애플리케이션 정도의 규모이면 DB 필요
- 웹 서버에서 동적으로 HTML을 생성하려면 제어용 프로그램 필요.
  웹 서버에 CGI라 불리는 동적 콘텐츠를 위한 확장 방식을 이용해 프로그램과 웹 서버 간을 연동, 통신하여 처리

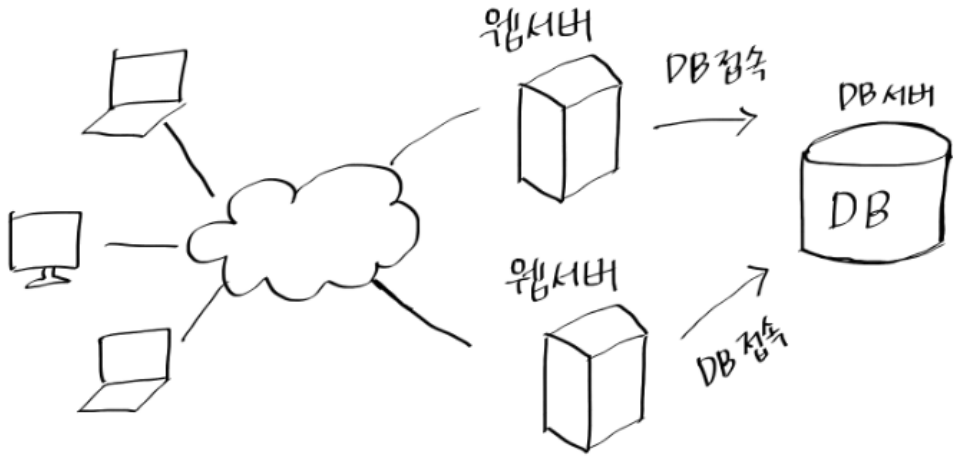
웹 애플리케이션의 구조

CGI : Common Gateway Interface

- 서버와 응용프로그램 사이에 데이터를 주고 받기 위한 표준화된 방법
- 웹 서버와 외부 프로그램 사이에서 정보를 주고 받는 방법이나 규약



웹 애플리케이션의 구조



- 실제로 DB에 접속하는 것은 PHP나 루비 등의 프로그래밍 언어로 만들어진 CGI 프로그램. DB 서버를 사용하기 위해서는 먼저 DB 서버와의 접속이 성립되어야 하는데, 이때 웹 서버의 CGI 프로그램이 DB의 클라이언트가 됨
- 클라이언트와 서버가 네트워크로 연결되어 있다면 서로 다른 머신에 두어도 무방. 웹 서버와 DB 서버를 서로 다른 머신에 두면 처리가 분산되어 시스템 전체 성능 향상

MySQL 클라이언트/서버

DB 서버 : MySQL 서비스

클라이언트 : mysql 커맨드

- MySQL 패키지를 설치하면 서버와 클라이언트 모두 사용 가능
- PC 한 대로도 클라이언트와 서버 모두 실행 가능하지만 네트워크 기능은 필요함.
  클라이언트에서 서버에 접속할 필요가 있는데 이때 네트워크를 경유해 PC의 서버로 되돌아오는 형태로 접속. 이를 '루프 백 접속'이라 함

Thank You

♡ 엔지니어링 팀 많은 관심 가져주세요! ♡