CH01 과제 20190937 이혜린

연습문제 05) 파일시스템과 DBMS의 장단점을 비교하여 설명하시오.

파일 시스템은 데이터 정의를 응용 프로그램에 하고 데이터 저장을 파일 시스템에 합니다. 따라서 데이터 정의와 프로그램의 독립성을 유지할 수 없습니다. 또한 데이터를 파일 단위로 저장하므로 데이터 중복이 발생할 수 있고 데이터 일관성이 결여됩니다. 따라서 프로그램 개발 생산성이 낮습니다. 하지만 응용 프로그램이 파일에 직접 접근하기 때문에 운영체제가 지원하고 별도의 소프트웨어 설치가 필요없기 때문에 CPU/주기억장치 사용량이 적다는 장점을 가지고 있습니다.

DBMS는 데이터 정의를 DBMS에 하고 데이터 저장을 데이터베이스에 합니다. 따라서 데이터 정의와 프로그램의 독립성이 유지가 가능합니다. DBMS를 이용하여 데이터를 공유하기 때문에 데이터 중복의 가능성이 낮고, 중복 제거로 데이터의 일관성이 최대한 유지됩니다. 짧은 시간에 큰 프로그램을 개발할 수 있고, 이외에도 데이터 복구, 보안, 동시성 제어, 데이터 관리 기능 등을 수행할 수 있습니다. 또한 데이터 무결성 유지, 데이터 표준 준수에 용이합니다. 하지만 별도의 소프트웨어를 설치하는 것이기 때문에 CPU/주기억장치 사용량이 많다는 단점도 가지고 있습니다.

연습문제 07) 다음 데이터베이스 사용자들의 역할을 설명하시오.

일반 사용자, 응용 프로그래머, SQL 사용자, DBA

일반사용자는 최종 사용자(end user)로, 실행 화면이나 프로그래머가 개발한 응용프로그램을 이용해 데이터베이스에 접근하는 대다수 사용자입니다. 프로그래밍 능력, SQL 언어, DBMS 지식, 데이터 구성 모두 알 필요가 없습니다.

응용 프로그래머는 일반 사용자가 사용할 수 있도록 프로그램을 만드는 사람입니다. 자바, C, JSP 등 프로그래밍 언어와 SQL을 사용하여 일반 사용자를 위한 사용자 인터페이스와 데이터를 관리하는 응용 로직을 개발합니다. 프로그래밍 능력, SQL 언어 모두 높은 수준을 가져야하고 DBMS 지식과 데이터 구성 모두 알 필요가 있습니다.

SQL 사용자는 SQL을 사용하여 업무를 처리하는 IT 부서의 담당자입니다. 응용 프로그램으로 구현되어 있지 않은 업무를 SQL을 사용하여 처리합니다. 프로그래밍 능력도 어느정도 가지고 있는게 좋고, SQL 언어를 높은 수준으로 알고 있어야 하며 DBMS 지식과 데이터 구성 모두 알 필요가 있습니다.

SQL 사용자에서 조금 더 확장된 것이 데이터베이스 관리자(DBA)라고 볼 수 있습니다. DBA는 DBMS의 슈퍼 사용자(super user)로, 데이터베이스 운영 조직의 데이터베이스 시스템을 총괄하는 사람이며, 데이터 설계, 구현, 유지보수의 전 과정을 담당합니다. 프로그래밍 지식도 알고 있는 것이 좋고, SQL언어, DBMS 지식, 데이터 구성 모두 높은 수준으로 알아야 할 필요가 있습니다.

DBA 주요 역할

- 데이터베이스 시스템 구성 요소 선정
- 데이터베이스 구조 정의
- 물리적 저장 구조와 접근 방법 결정
- 무결성 유지를 위한 제약 조건 정의
- 보안 및 접근 권한 정책 결정
- 백업 및 회복 기법 정의
- 데이터베이스 시스템 관리
- 시스템 성능 감시 및 성능 분석
- 데이터베이스 재구성
- 데이터베이스 관련 의견 조정과 분쟁 해결

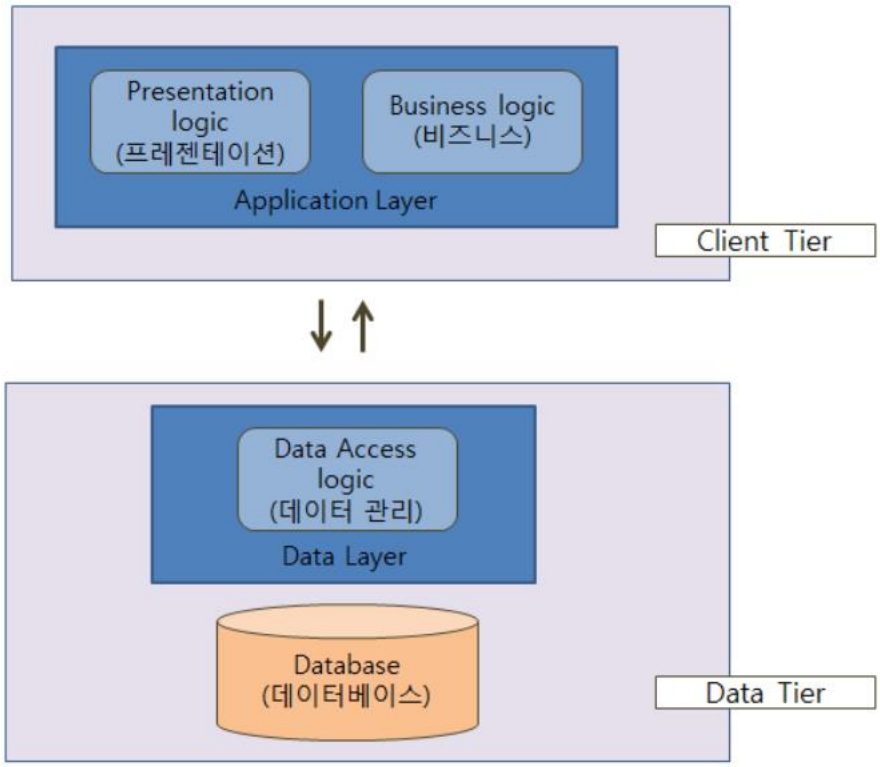
심화문제 11) 클라이언트/서버 구조를 설명하고, 2-tier, 3-tier 개념을 설명하시오. (인터넷 참고)

클라이언트(client)는 서비스 요구자, 서버(server)는 서비스 제공자의 형태로 네트워킹하며 CPU, 하드디스크, 주변기기 등의 자원을 공유하는 분산 처리하는 방식입니다. 클라이언트 쪽에서 요구하면 서버에서는 그것에 대한 과정을 통해 응답하는데 프로세스를 의뢰하는 장치나 컴퓨터 혹은 프로그램을 '클라이언트'라' 하고, 의뢰받은 프로세스를 '실행', 응답하는 쪽을 '서버'라고 합니다. 개방형을 지향해서 상호 독립적으로 작업을 하여도 정보가 고립되지 않으며, 다른 기종들 간의 자유로운 통합이 가능합니다. 이러한 시스템에서 클라이언트는 대개 개인용 컴퓨터들로 자유롭게 컴퓨터 구성을 선택해 관리하고 자신의 필요를 충족할 수 있습니다. 반대로 서버는 엄격한 규칙에 따라 구성해야만 각종 자원을 공유할 수 있으며 높은 수준의 자료 보전성과 신뢰성을 가지게 되어 클라이언트들의 요구에 응답할 수 있습니다. 대부분의 업무용 프로그램은 클라이언트/서버 구조를 적용하며, 인터넷을 예로 들면 웹 브라우저는 인터넷상의 어딘가에 위치한 웹 서버에게 웹 페이지나 파일의 전송을 요구하는 클라이언트 프로그램입니다.

계층(Tier) : 컴포넌트들의 물리적인 분리를 뜻합니다.

층(Layer) : 컴포넌트들의 논리적인 분리를 뜻합니다.

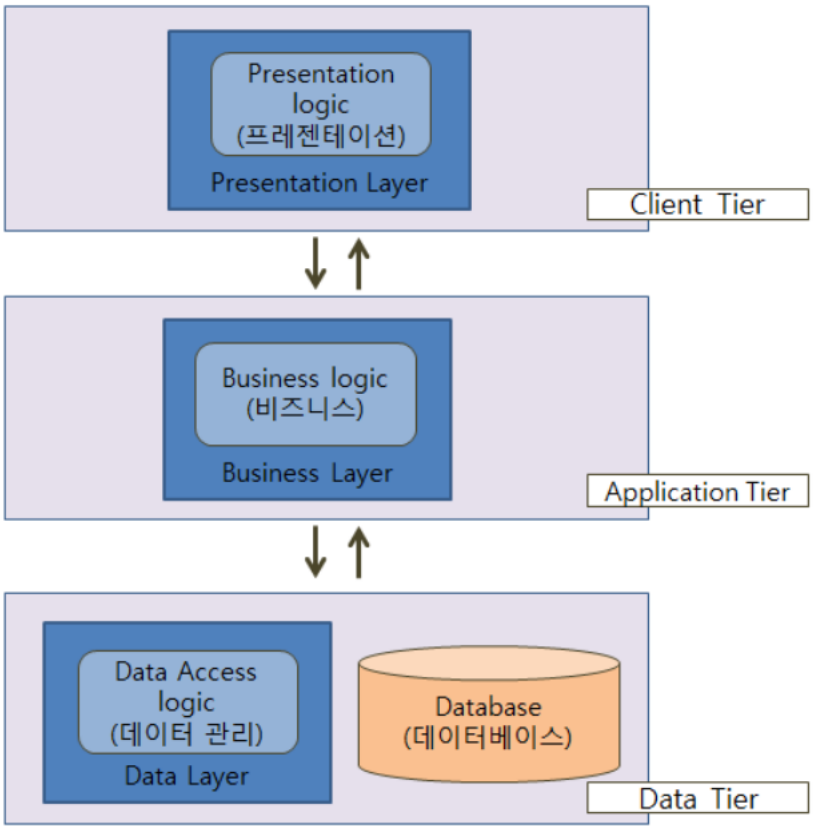
2-TIER



2계층 구조(2 Tier Architecture)란 유저 시스템 인터페이스 환경과 데이터베이스 시스템 서버 환경으로 두 개의 물리적 컴퓨터로 나뉘어진 것을 일컫습니다.

클라이언트가 직접 서버의 DB에 접속하여 자원을 활용할 수 있게 됩니다. 중요한 비즈니스 로직들은 서버에서 돌지만 간단한 연산문제나 간단한 처리는 터미널 형태에서 작업을 하는 것입니다. 이 결과 서버의 부담이 많이 줄어들게 되었습니다. 하지만 비즈니스 로직의 업그레이드시 클라이언트 각각을 업그레이드 해야한다는 문제가 발생하게 되었습니다. 옛날에는 하나만 비즈니스 로직을 변경하면 모두 일괄적으로 변경이 되었는데 이제 그 비즈니스 로직(중요한 것은 서버에 있겠지만)이 바뀌면 모든 클라이언트를 업그레이드 해야하는 비용문제가 발생할 수 있습니다.

3-TIER



3계층 구조(3 Tier Architecture)란 프레젠테이션 로직 (클라이언트, 사용자 인터페이스), 비즈니스 로직, 데이터베이스 로직을 각각 다른 플랫폼 상에서 구현한 것입니다. 3계층 구조에서 각 계층은 물리적으로도 독립적이며 각 계층의 변경이 다른 계층에 의존하지 않습니다.

1. 프레젠테이션(클라이언트) 계층

프레젠테이션 계층은 응용 프로그램의 최상위에 위치하고 있는데 이는 서로 다른 층에 있는 데이터 등과 커뮤니케이션을 합니다.

- 사용자 인터페이스를 지원합니다. (인터넷 브라우저의 정적인 데이터를 제공합니다.)

- 이 계층은 GUI, 또는front-end도 불립니다.

- 비즈니스 로직이나 데이터 관리코드를 포함해서는 안됩니다.

- 주로 웹서버를 뜻합니다(물리적 : WEB서버)

ex) HTML, javascript, CSS, image

2. 애플리케이션 계층

이 계층은 비즈니스 로직 계층 또는 트랜잭션 계층이라고도 하는데, 비즈니스 로직은 워크스테이션으로부터의 클라이언트 요청에 대해 마치 서버처럼 행동합니다. 차례로 어떤 데이터가 필요한지를 결정하고, 메인프레임 컴퓨터 상에 위치하고 있을 세 번째 계층의 프로그램에 대해서는 마치 클라이언트처럼 행동합니다.

- 정보처리의 규칙을 가지고 있습니다.(동적인 데이터를 제공합니다)

- middleware 또는 back-end로 불립니다.

- 프레젠테이션코드나 데이터관리 코드를 포함해서는 안됩니다.

-주로 어플리케이션 서버를 뜻합니다(물리적 : WAS서버)

ex) Java EE, ASP.NET, PHP

3. 데이터 계층

데이터 계층은 데이터베이스와 그것에 액세스해서 읽거나 쓰는 것을 관리하는 프로그램을 포함합니다. 애플리케이션의 조직은 이것보다 더욱 복잡해질 수 있지만, 3계층 관점은 대규모 프로그램에서 일부분에 관해 생각하기에 편리한 방법입니다.

-데이터베이스를 주로 뜻합니다.

-DB 또는 File System를 접근 및 관리합니다.

-back-end라고도 불립니다.

-주로 DB서버를 뜻합니다(물리적 : DB서버)

ex) MySQL DB, Oracle DB

3계층구조를 사용하면 각 계층별로 웹 디자이너, 소프트웨어 엔지니어, DB 관리자가 역할분담을 하여 일을 효율적으로 할 수 있습니다. 회사 규모 및 사용자의 증가에 따라서 1,2,3계층 구조를 고려해야 합니다. 비용은 3계층으로 갈수록 많이 듭니다.