

System Analysis and Design#9

201823869 조성우



Distinguish between the two-tier, three tier, and n-tier client server architectures.

Tier Architecture는 클라이언트와 서버를 몇 개로 나누고 계층별 역할을 분담할 것인지를 기준으로 나뉜다.

2-Tier Architecture

Client/Server 모델은 일반적으로 사용자를 위한 인터페이스 프로그램이 사용자의 데스크탑에 위치하고, DBMS의 서비스가 서버로서 존재한다. 보통 서버는 많은 Client에게 서비스를 제공하기 위해서 보다 고사양의 machine에 설치된다.

즉, 유저의 인터페이스 환경과 Database 시스템 서버 환경과 같이 각각 두 부분으로 나누어지기 때문에 2-Tier라고 부른다.

1. 클라이언트 애플리케이션(클라이언트 계층)
2. 데이터베이스(데이터 계층)

클라이언트 시스템은 프리젠테이션 및 애플리케이션 계층을 모두 처리하고 서버 시스템은 데이터베이스 계층을 처리한다. 클라이언트 서버 애플리케이션이라고도 한다. 클라이언트와 서버 사이에 통신이 이루어진다. 클라이언트 시스템이 요청을 서버 시스템으로 보내고 서버 시스템이 요청을 처리하여 클라이언트 시스템으로 데이터를 다시 전송하는 방식.

3-Tier Architecture

3-Tier 구조는 2-Tier 구조의 제한을 극복하기 위해서 탄생한 구조로서 Multi-Tier 구조라고도 불린다. 3-Tier 구조에서는 유저의 인터페이스 환경과 Database 서버 환경 사이에 중간층(Middle_

Tier)이 추가된 구조이다.

3-Tier 구조는 2-Tier 구조보다 사용자가 많은 경우의 유연성을 향상시키기 위해 개발되었다.

Application을 분리하여 중간 계층인 Application Server에 두고 프레젠테이션 로직과 Database를 미들웨어로 연결하는 형태

1. 프레젠테이션 계층(클라이언트 계층)
2. 애플리케이션 계층(비즈니스 계층)
2. 데이터베이스 계층(데이터 계층)

클라이언트 시스템은 프레젠테이션 계층을 처리하고, 애플리케이션 서버는 애플리케이션 계층을 처리하며, 서버 시스템은 데이터베이스 계층을 처리한다.

N-Tier Architecture

계층은 N-Tier 애플리케이션이다.

N-Tier 애플리케이션 AKA 분산 애플리케이션.

3계층 아키텍처와 유사하지만 로직이 분산될 수 있도록 비즈니스 로직을 분산하기 위해 애플리케이션 서버의 수를 늘리고 개별 계층으로 표현한 방식이다.