# 멀티유저 온라인 게임을 위한 데이터베이스 설계에 관한 연구

김종수*, 김태석,
동의대학교 소프트웨어공학과
e-mail:seatree@deu.ac.kr

# A Study on a Database Design for the Multi-User Online Game

Jong-Soo Kim*, Tai-Suk Kim
Dept of Software Engineering, Dong-Eui University

요 약

멀티유저 온라인 게임과 관련된 서버 측 설계에서 효율적인 네트워크 구성에 대해서 살펴보고, 현재 인기를 끌고 있는 멀티유저 온라인 게임의 사례 분석을 통하여 사용자에게 흥미를 끌기 위해 구성될 수 있는 다양한 엔티티(Entity)를 추출하여 정규형 단계를 거침으로서, 멀티유저 온라인 게임의 데이터베이스 설계 구현에 효율적으로 적용될 수 있는 설계를 제안한다.

## 1. 서론

MMORPG(Massive Multi-player Online Role Playing Game)와 같은 게임에서는 사용자의 흥미를 유발 시키는 클라이언트 측 구현이 중요한 요소이지만, 게임 운영자 측면에서 볼 때는 클라이언트에게 안전하고 지속적인 서비스를 해 줄 수 있는 게임 서버의 제작이 더 중요한 요소라고 볼 수 있다[1].

일반적으로 네트워크 게임은 분산된 작업이 가능하게하고, 기존에 개발된 소프트웨어의 재사용성을 높이기 위해 다계층(multi-tier) 구조로 제작되는데, 이 경우 각각의 목적에 맞는 서버 구성을 위해 하드웨어나 소프트웨어적으로 서버가 분리되는 경우가 많으며, 데이터베이스 서버의 경우는 새로운 계층으로 분리하는 것이 거의 일반적이다. 다계층으로 설계되는 애플리케이션의 한 종류인 온라인 게임 분야 중에서도 MMORPG와 같은 게임 서버의 구조는 그 성격이 폐쇄적이므로, 자료의 공유가 어렵다.

본 논문에서는 현재 일반에게 널리 서비스 되고 있는 게임을 바탕으로 서버 측 구조 중에서 데이터베이스와 관련된 부분이 어떻게 구현되었는지 유추해 보고, 대규모 멀티유저 온라인 게임에서 효율적으로 적용될 수 있는 데이터베이스 설계 방법을 제안한다.

## 2. 멀티유저 온라인 게임 DB 설계의 요구사항

흥미로운 게임을 제작하기 위해 게임 서버의 설계 기술은 매우 중요하다. 게임 서버를 구성하기 위해서는 하드웨어적으로 구성되는 네트워크 구조와 클라이언트에서 발생한 중요한 데이터를 저장하는 데이터베이스 서버의 구성과 관련된 기술 그리고 효율적으로 데이터를 저장할 수 있는 설계에 대한 기술이 필요하다.

게임 서버의 설계와 관련하여 현재 많이 이용되고 있는 네트워크 구성 기술은 P2P(Peer to Peer) 구조, C/S(Client/Server)구조, 분산 서버 구조가 있으며, 이것을 조합한 하이브리드(Hybrid)형 서버 구조가 있다.

P2P 방식은 실시간 전략 시뮬레이션 게임이나 액션 게임의 경우에 많이 응용되고 있는 방법으로 게임 플레이어의 컴퓨터가 직접적으로 네트워크 상에서 상대편을 찾고, 상대편이 발견된 다음에는 다

른 컴퓨터의 개입 없이 두 사람의 컴퓨터 간에 서로 메시지를 주고받는 방식을 사용한다.

C/S 방식은 여러 대의 클라이언트가 서버와 자료를 주고받는 방식으로 대부분의 게임 서버 구성에서 부분적으로나 전체적으로 채택되는 방식이다. C/S 방식의 게임 서버 구성에서 모든 클라이언트는 공유해야 할 정보를 서버에 보내며 서버는 이러한 입력을 기반으로 게임의 모든 진행 결과를 전체 클라이언트에게 보낸다.

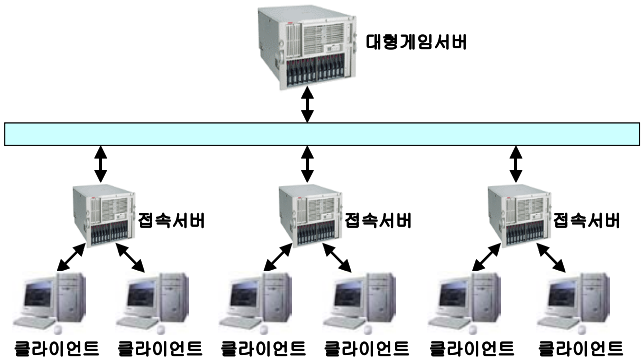


그림 1 분산 서버 형식의 게임 서버 구성

분산 서버 구조는 대규모 네트워크 게임을 위해서 적합한 구조인데, 그림 1과 같은 구성을 가진다. 분산 서버 구조에서 게임 클라이언트는 여러 개로 묶어진 서버군 중 한 개의 서버 군에 접속하여 게임을 진행하게 되는데, 이렇게 함으로써 서버의 작업량을 여러 서버로 나눌 수 있으며, 부하 분산의 효과가 있다.

부하를 분산하는 방식에 크게 2가지 방식이 있는데 게임 플레이어의 서버 접속을 여러 대의 접속 서버가 담당하고, 메인 서버가 이를 제어하는 방식과 게임을 여러 개의 맵으로 나누고 게임 플레이어가 어느 맵에 있는가에 따라 해당 맵을 담당하는 서버에 접속하는 방식이 있다.

네트워크 게임 서버의 데이터베이스 구성과 관련해서 어떠한 데이터베이스 시스템 구조를 채택하는가 하는 것도 중요한 문제인데, 데이터베이스 서버의 분리는 다른 서버의 일부 작업들은 데이터베이스 시스템과 같이 수행하고, 다른 작업들은 게임 서버와 클라이언트가 담당하게 함으로서 작업을 분담시킬 수 있고, 부하를 분산할 수 있다는 장점이 있다.

여러 개의 분산 서버로 구성되는 네트워크 게임 서버의 데이터베이스 구성에 분산 데이터베이스나 병렬 데이터베이스 시스템[2]을 활용할 수도 있지만, 분산 데이터베이스 시스템이나 병렬 데이터베이스 시스템은 아직 발전단계에 있고, 참고할 만한 시스템을 찾기 어려운 실정이다.

## 3. 사례 연구를 통한 데이터베이스 설계

데이터베이스 설계의 반복 작업을 최소화하고 효율적인 게임 서버의 분석과 설계를 위해 기존에 인기를 끌고 있는 사이트를 기초로 데이터베이스 설계를 유추하는 방식은 매우 유용하다. 본 논문에서 제안하고 있는 효율적인 데이터베이스 구성과 설계에 있어서도 요즘에 학생들과 일반인들에게 크게 인기를 얻고 있는 네트워크 게임의 I/O 설계를 바탕으로 멀티유저 온라인 게임에서 효율적으로 적용될 수 있는 데이터베이스 설계를 분석하였다.

### 3.1 네트워크 게임 DB 설계 분석

그림 2는 현재 학생들과 일반인들에게 많은 인기를 얻고 있는 네트워크 게임으로 Wizet사에서 제작하여 Nexon사에서 서비스하고 있는 메이플스토리(Maple Story)[3]라는 게임으로 DirectX를 이용하여 만들어 졌다.

그림 2 Nexon이 서비스 하고 있는 MapleStory[3]

게임은 횡 스크롤 방식을 채택하였고, 분산 서버 구조의 형태를 가진다는 것을 유추할 수 있으며. 서버의 부하를 분산하기 위해 여러 개의 맵으로 구성된 게임형태를 가지고 있다. 수익 모델에 있어서는 부분 유료화를 채택하고 있으며, 국제화에 성공하여 현재 일본, 중국, 기타 영어권 나라에 성공적으로 서비스 하고 있다.

### 3.2 기존 게임의 I/O분석을 통한 DB설계 유추

관계형 데이터베이스 설계의 목표는 불필요한 데이터의 중복을 없애고, 정보 검색을 쉽게 할 수 있는 관계 스키마 집합을 생성하는 것이다. 일반적으로 데이터베이스의 설계는 애플리케이션 구현 전에 이루어지며, 설계 전단계의 분석 단계로부터 정의된 문제 영역에서 추출된 엔티티(Entity)와 이들 사이의 관계가 정규형 단계를 거치면서 설계 목표를 달성할 수 있고, 이것을 바탕으로 I/O가 설계되거나, 설계된 I/O를 바탕으로 데이터베이스 설계의 정확성이 다시 검토된다. 그러나 본 논문에서는 일반적인 시스템 설계 과정과 다르게 이미 네트워크 게임으로 성공적으로 운영되고 있는 게임 시스템의 몇 가지 GUI를 바탕으로 효율적인 데이터베이스 설계를 유추해 보려고 한다.


그림 3 item 저장소[3]


그림 4 장비 저장소[3]

게임 내에서 캐릭터는 다양한 아이템을 가질 수 있는데, 그림 3과 4는 캐릭터가 습득한 아이템과 현재 캐릭터가 착용하고 있는 아이템에 대한 정보를 보여주는 창이다. 각각의 GUI는 다른 창으로 구성되어 있지만, 캐릭터가 가지는 아이템이라는 공통점이 있다.

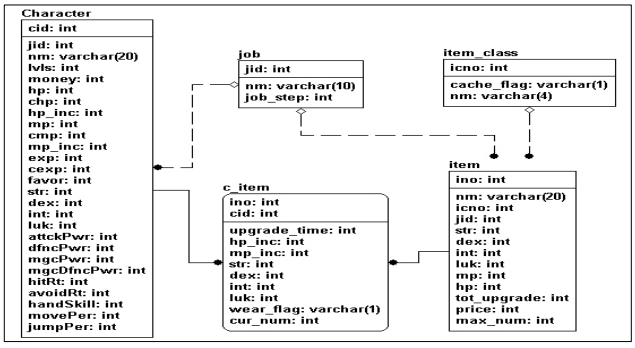


그림 5 캐릭터의 아이템과 장비를 위한 DB 설계

그림 3의 item 저장소 GUI에서 캐릭터가 가질 수 있는 아이템은 장비, 소비, 설치, 기타, 펫(Pet)으로 구분되는 것을 볼 수 있다. 그리고 현재 캐릭터가 착용하고 있는 아이템은 그림 4에 나타나 있다. 두개의 GUI를 바탕으로 데이터베이스 설계를 유추해 보면 그림 5와 같다.

ER-다이어그램에서 item 엔티티는 다른 엔티티와 종속 관계없이 별도로 데이터를 관리할 수 있는 독립적인 엔티티가 된다. item 엔티티는 직업별로 가질 수 있는 아이템의 분류가 필요한데, 이것을 위해서 Character 엔티티에서와 같이 item 엔티티도 job 엔티티의 기본 키를 외래 키로 가질 수 있다. 또한 아이템은 그림 3의 GUI에서 보듯이 아이템의 종류는 장비, 소비, 설치, 기타, 펫(Pet)이 있는데, 이러한 아이템의 분류는 item_class 엔티티가 담당한다.

그림 6은 현재 캐릭터가 사용하고 있는 기술(Skill)에 대한 정보를 보여주는 창으로써, 게임 내에서 사용자 캐릭터는 자신의 직업, 경험치 그리고 레벨에 따라 특정한 기술을 선택하여 사용할 수 있다. 게임 플레이어는 캐릭터의 기술을 사용함으로써, 경험치를 빨리 획득한다든지, 캐릭터의 상태를 보다 나은 상태로 만들 수 있다. 특정한 캐릭터가 사용할 수 있는 기술도 데이터베이스로 관리되어야 한다. 그림 7은 게임 내에서 특정 캐릭터와 접촉하기 위한 친구 등록과 2명 이상의 캐릭터가 동시에 게임을 진행하는 파티(Party) 시스템의 정보를 보여 주는 창이다.


그림 6 skill Inventory[3]


그림 7 User List[3]

그림 6을 분석해 보면, 캐릭터의 직업별로 전직을 할 수 있는 직업이 정해져 있고, 한 개의 캐릭터는 1차, 2차, 3차, 4차 전직에 따라 사용할 수 있는 기술이 늘어남을 알 수 있다.

그림 7을 분석해 보면, 한 개의 캐릭터는 여러 명의 캐릭터를 친구로 등록할 수 있으며, 게임이나 퀘스트의 수행 시, 몇몇 캐릭터와 파티 시스템을 사용하여 게임을 진행할 수 있다. 친구 등록과 파티 시스템은 게임 내에서 커뮤니티를 형성하게 해주는 중요한 역할을 한다. 그러므로 이에 대한 정보도 데이터베이스로 관리되어야 한다. 그림 6과 7의 GUI를 바탕으로 데이터베이스 설계를 유추해 보면 그림 8과 같다.

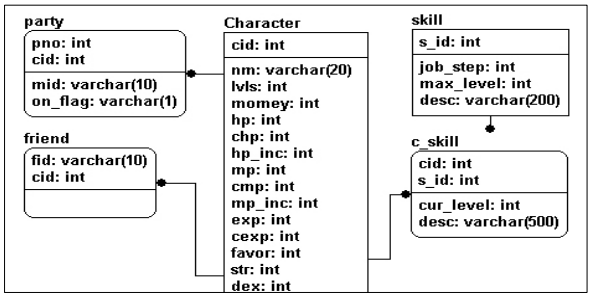


그림 8 친구, 파티, 기술 관리를 위한 DB 설계

그림 8의 ERD에서 전체 기술에 대한 정보는 skill 엔티티에서 관리하고, 캐릭터가 사용할 수 있는 기술에 대한 정보는 c_skill 엔티티가 담당하게 된다. 그리고 친구 등록과 파티 시스템은 하나의 GUI에 구현되었지만 별개의 엔티티를 가진다. 친구 등록은 friend 엔티티가 파티 시스템 관리는 party 엔티티가 담당한다. 파티 시스템의 구현에서 파티에 참가하는 캐릭터들에게 게임에서 얻은 전체 경험치와 사이버머니를 어떻게 분배할 것인가 하는 것도 중요한 문제인데, 이것은 일반적으로 실시간 게임 진행을 맡은 게임 서버가 담당하므로 이것과 관련된 정보를 데이터베이스로 관리할 필요는 없다.

### 3.3 설계 유추를 통한 DB 설계

기존의 게임을 바탕으로 분석된 데이터베이스 설계에서 나타난 엔티티를 바탕으로 한 멀티유저 온라인 게임의 데이터베이스는 그림 9와 같이 설계되었다.

추가적으로 데이터베이스 설계 시에 고려되어야 할 사항은 기존에 구축되어 있는 타 시스템과의 인터페이스 부분이다. 게임을 위한 데이터베이스 서버와 기존의 회사업무 또는 인터넷 쇼핑몰 시스템은 통합시스템으로 한번에 구축되면 더 좋은 시스템이 구축될 수 있지만, 비용이나 시간측면에서 그렇게 하지 못하는 경우가 많다. 게임을 위한 데이터베이스의 설계에서도 데이터의 효율적인 이용이라는 측면에서 다른 시스템과의 인터페이스를 충분히 고려한 설계가 이루어져야 한다.

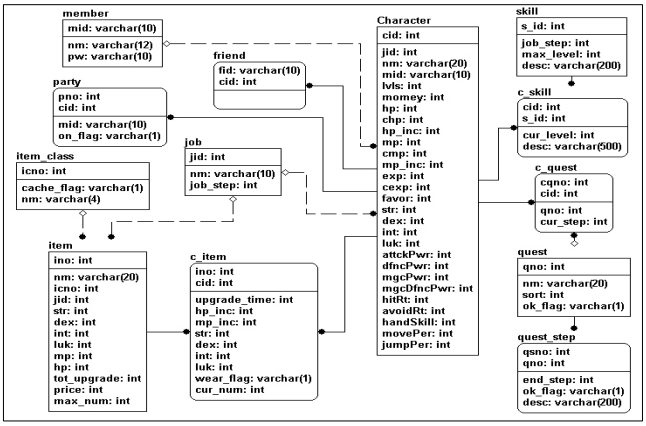


그림 9 네트워크 게임을 위한 DB 설계

### 4. 결론

본 논문에서 DirectX API를 기반으로 제작된 멀티유저 온라인 게임의 I/O를 바탕으로 서버 측 구현에 필요한 데이터베이스 설계를 유추하였고, 게임을 위한 데이터베이스 서버 구현에 효율적으로 적용될 수 있는 설계를 제안하였다.

대규모 멀티유저 온라인 게임을 위한 데이터베이스 설계에 게임 참여자들의 흥미를 유발하기 위한 다양한 엔티티를 유추해서 설계하였고 정규화를 통해서 중복된 데이터를 최소화하기 위한 설계의 예를 보였다.

향후에는 멀티유저 온라인 게임을 위한 데이터베이스와 관련된 API의 구현에 효율적으로 적용될 수 있는 디자인 패턴에 대해서 연구할 예정이다.

### 참고문헌

[1] 김종수, “웹을 기반으로 한 JAVA 네트워크 게임 시스템의 설계와 구현”, 석사 논문, 부산외국어대학교. pp. 1-11, 2003.

[2] Abraham Silberschatz, Henry F. Korth and S. Sudarshan, “데이터베이스 시스템, 4판”, pp. 703-704, 2003.

[3] http://www.maplestory.com