* 1. Concept
		1. 클라이언트 계정의 생성
			1. 각 클라이언트 계정들은 고유한 식별 번호를 가진다.

: 이 식별 번호는 계정의 이름과는 별도로 구분하는 데이터이다.

* + - 1. 클라이언트 계정의 식별 번호는 8바이트 부호 없는 정수형을 사용한다.

: 값 범위는 0 ~ 약 1,844경이다. (단, 값 0은 무효한 값이다.))

 ※ 계정 이름도 겹치지 않아야 하는 조건이 있으므로, 식별자로 사용할 수 있다.

 그러나, 컴퓨터는 문자열보다 숫자 값이 훨씬 다루기 편하고 연산이 빠르기 때문에, 각 계정마다 고유한 숫자를 할당하는 것이다.

* + - 1. 클라이언트 계정의 식별 번호는 한 군데의 데이터베이스에서만 발급을 담당한다.

: 생성한 계정의 식별 번호는 여러 군데에서 보관할 수도 있으나, 발급 자체는 반드시 한 군데에서만 담당한다.

 ※ 여기서의 의미는, 계정을 고유하게 구분해야 할 필요가 있는 서버 지역 별로 한 군데에서만 담당해야 한다는 의미이다.

 서버 지역은 전 세계에 하나만 있는 경우도 있지만, 몇 개로 쪼개어져 있을 수도 있다. 이 때는 쪼개어진 각각의 서버 지역마다 계정 발급 담당이 하나씩 존재한다. 물론, 이 경우에는 서버 지역끼리는 기본적으로 완전히 분리해서 게임을 서비스한다고 가정하고 있다.

* + - 1. 한 번 생성한 클라이언트 계정의 식별 번호는 **도중에 다른 값으로 변경하거나, 다른 계정 식별을 위한 식별 번호로써 재사용하지 않는다.**
			2. 인증 서버 및 데이터베이스를 제외한 다른 서버 모듈들과 데이터베이스에서는, 각 플레이어를 식별하는 데, 이 클라이언트 계정의 식별 번호를 이용한다.

: 즉, 인증 데이터베이스는 로그인 한 사용자의 계정 정보를 통해서 클라이언트의 식별 번호를 가져올 수 있어야 한다.

* + - 1. 각 사용자의 고유 식별 번호는, **그 사용자의 실제 게임 데이터가 저장되어 있는 게임 DB를 식별할 수 있는 정보를 포함**한다.

: 게임 DB는 접근 빈도 및 사용 성능상의 문제로 인해 분산할 가능성이 매우 높다.

 그런 상황에서 이런 기능이 없다면, 사용자의 실제 게임 데이터를 찾으려고 할 때마다 인증 DB를 뒤져야 하는 상황이 될 수 있다. 이는 인증 DB를 성능 병목 지점이 되게 만들 수 있으므로, 인증 번호 그 자체에 게임 DB 번호를 포함할 수 있는 수단을 주는 게 좋다.



<사용자 번호 디자인 방식>



<같은 게임 서버에 접속하더라도, 데이터에 접근하는 게임 DB는 다를 수 있다.>