

3

3주차: 데이터테이블

데이터 테이블

많은 데이터를 정리 / 관리하기 위한 테이블

플레이어의 성장과 관련된 테이블부터, 적과 관련된 테이블, 스테이지(콘텐츠)를 이루는 테이블 등이 존재

테이블 예시

플레이어의 성장과 관련된 테이블 예시

- 캐릭터 테이블
 - 간단한 데이터 존재 (이름, 직업 구분 등)
- 장비 테이블
 - 등급 및 강화 관련 내용 추가 (장비 강화 테이블, 장비 스탯 테이블 등 따로 만드는 경우도 많음)
- 소모품 테이블
 - 물약 등에 대한 정보
- 스킬 테이블 (성장 포함)
- 스탯 테이블 (성장 포함)
 - 캐릭터 뿐만 아니라 장비의 스탯 테이블도 존재할 수 있음
 - 스탯 종류 테이블, 캐릭터 성장에 따른 스탯 변화 테이블 등 다양한 테이블이 있다.
- 레벨 테이블 (+exp) (성장 포함)

적과 관련된 테이블 예시

- 몬스터 테이블
- 몬스터 스킬 테이블
- 몬스터 사망시 드랍 테이블
- 몬스터 AI 테이블
 - 몬스터의 행동 패턴이라고 이해하면 된다.

스테이지와 관련된 테이블 예시

- 맵 테이블
 - A맵에서 B맵으로 넘어갈 때 시작 위치를 어디로 설정할 지
 - 각 맵에 대한 성격 (마을, 사냥터, 던전 등)
- NPC 테이블
 - NPC는 기능(상점, 수리 등)이 있는 NPC와 없는 NPC로 구분할 수 있음
- 퀘스트 테이블
- 콘텐츠 테이블
 - 콘텐츠 전체를 하나의 테이블로 만들 수도, 각각의 콘텐츠 별로 만들 수도 있다.

스테이지에 들어가지만 테이블과 살짝 성격이 다른 것들

- NPC 스폰너
 - NPC를 맵에 배치한다.
- 몬스터 스폰너
 - 몬스터를 맵에 배치한다.
- 이벤트 트리거
 - 이벤트를 맵에 배치한다.

아이템 테이블 작업해보기

아이템에 해당하는 것들

재화, 잡템, 소모품(유료/무료), 퀘스트 아이템, 성장 관련 재료, 각종 토큰, 장비(유료/무료), 코스튬(유료/무료)

아이템 테이블 분류하기

방법1. 세분화된 개별적인 테이블로 만들 수 있다.

잡템 테이블	장비 테이블	퀘스트 아이템 테이블	유료 소모품 테이블	재화 테이블
소모품 테이블	토큰 테이블	유료 장비 테이블	유료 코스튬 테이블	무료 코스튬 테이블

방법2. 어느 정도 분류를 한 뒤 하나의 테이블로 묶을 수 있다.

장비 테이블
장비 테이블
유료 장비 테이블
무료 코스튬 테이블
유료 코스튬 테이블

아이템 테이블
퀘스트 아이템 테이블
잡템 테이블
토큰 테이블
재화 테이블

소모품 테이블
소모품 테이블
유료 소모품 테이블

방법3. 하나의 테이블로 만들 수도 있다. → 강의에서는 이 방법으로 진행

		아이템 테이블		
잡템 테이블	장비 테이블	퀘스트 아이템 테이블	유료 소모품 테이블	재화 테이블
소모품 테이블	토큰 테이블	유료 장비 테이블	유료 코스튬 테이블	무료 코스튬 테이블

테이블을 잘게 쪼갠 때의 장점

1. 작업의 병목 현상이 생기지 않는다.

테이블은 1개인데, 작업자가 여러 명이라 생기는 이슈 방지 가능

2. 문제가 생긴 테이블 하나만 되돌리면 된다.

크리티컬한 버그가 생겨 해결이 힘든 상황에서 해당 테이블만 되돌리면 된다.

테이블을 잘게 쪼갠 때의 단점

1. 데이터가 쌓일 수록 어떠한 요소를 추가하기 위해서 다양한 테이블을 열어놓고 작업해야 한다.

보통 연관되어 있는 요소들이 많기 때문에 테이블 하나로 끝나지 않는 경우가 많음.

2. 밸런스 작업을 할 때, 전체 테이블을 수정해야 하는 경우가 생긴다.

아이템 테이블에 들어가야 하는 항목들

index, 이름, 아이템 종류, 아이콘 파일, 스킬 id, 리소스 파일, 구입 금액, 판매 금액, 구입가능 여부, 판매가능 여부, 거래가능 여부, 스탯id, 장착 레벨 제한, 장착 직업 제한 등

(장비 부위가 빠진 상태)

아이템 테이블 예시

컬럼명(필드명)	컬럼명(영어)	설명	요소들	자료형
index	index ↗	주소를 뜻한다	아이템 테이블에서는 index 로 사용 한다. 상세 내용은 말로 본다.	int
이름	name	아이템의 이름을 작성 한다.	보통 글로발을 염두에 두면, string 테이블을 따로 만든다.	string
설명(실제 사용 X)	description	실제 사용되는 내용이 아닌, 개발자가 확인하기 위한 용도의 컬럼	개발자가 헷갈릴 만한 요소를 간략하게 작성하여 최소한의 방어용으로 사용 한다.	string
아이템 설명	itemInfoTxt	아이템의 설명 내용을 작성 한다.	글로발을 염두에 두면, string 테이블의 키 값을 작성 한다.	string
아이템 종류	type	아이템 테이블에 하나의 아이템 속성만 가지는 것이 아니므로, 아이템의 속성을 구분하기 위한 종류(type) 컬럼을 추가 한다.	equip : 장비 consume : 소모품 currency : 재화 quest : 퀘스트용 아이템 token : 토큰 material : 재료 misc(miscellaneous) : 잡템 costume : 코스튬	string
재화 타입	currencyType	재화의 타입을 뜻 한다. 재화의 key 값을 정의하기 위한 요소라고 보면 된다.	cash : 현금 diamond : 유료 재화 ruby : 무료 재화 gold : 게임 내 재화	string

장비 등급	equipGrade	장비의 등급을 작성 한다.	normal : 일반 magic : 마법 rare : 희귀 epic : 에픽 legend : 전설	string
장착 레벨 제한	equipLv	장착 레벨 제한을 작성 한다.	장비를 장착할 수 있는 레벨 제한을 작성 한다.	int
유료 무료 구분	buyType ↗	유료인지 무료인지를 구분 한다.	Free : 무료 Paid : 유료	string
남은 시간	remainTime	시간 제한 아이템의 남은 시간을 작성 한다.	남은 분을 입력 한다. 하루는 = 576 분 일주일 = 4032 분	int
목표 날짜	targetDATE	특정 날짜 제한인 경우 날짜와 시간을 작성 한다.	yyyymmdd:hhmmss 형태로 입력 한다. 20230903:240000	string
아이콘 파일	iconPath	아이콘의 경로와 파일 이름을 작성 한다.	/asset/resource/UI/equip/icon/파일명.png	string
스킬 id	skillIndex	스킬의 인덱스를 연결 한다.	스킬 테이블의 index 를 작성 한다. 소모품의 경우 스킬을 연결하여 사용 한다.	int
소모품 그룹	consumeGroup	물약의 경우 그룹으로 묶어 쿨타임을 같이 적용 한다.	그룹 id 를 작성 한다. 임의의 값으로 작성 한다. 소모품을 사용하면 같은 그룹 id 끼리는 쿨타임을 공유 한다.	int
리소스 파일	resourcePath	리소스의 경로와 파일 이름을 작성 한다.	/asset/resource/EQUIPMENT/파일명.prefab	

구입 재화 종류	buyCurrencyType	구입 시 사용하는 재화의 종류를 작성 한다.	currency 타입들의 enum 값을 작성 한다. 보통 많이 쓰는 형태는 아래와 같다. diamond : 유료 다이아. ruby : 무료 다이아. gold : 게임 내 재화	string
구입 금액	buyCost	구입 금액을 작성 한다.	구입 시의 금액을 작성 한다.	int
판매 후 획득 재화 종류	sellCurrencyType	판매 시 획득하는 재화의 종류를 작성 한다.	구입 시와 동일 하다.	string
판매 금액	sellCost	판매 금액을 작성 한다.	판매 시의 금액을 작성 한다.	int
구입 가능 여부	useBuy ↗	구입 가능 여부	TRUE : 구입 가능 FALSE : 불가능	string
판매 가능 여부	useSell	판매 가능 여부	TRUE : 판매 가능 FALSE : 불가능	string
거래 가능 여부	useTrade	거래소 등록 가능 여부 개인 거래가 있을 경우 따로 추가 한다. isPrivateTrade 비슷 한 형태	TRUE : 등록 가능 FALSE : 불가능	string
파괴 가능 여부	useDestroy	파괴 가능 여부 특정 아이템의 경우 파괴만 가능하도록 만든다.	TRUE : 파괴 가능 FALSE : 파괴 불가능	string
스탯 id	statIndex	스탯의 인덱스를 작성 한다.	스탯 테이블의 인덱스를 작성 한다.	int
사용 레벨 제한	useLv	사용 레벨 제한	이 아이템을 사용할 수 있는 레벨 제한을 작성 한다.	int
장착 직업 제한	equipClass	장착 직업을 작성 한다.	장비를 장착할 수 있는 직업을 작성 한다. ALL : 모두 가능 FIGHTER : 전사만 가능 MAGE : 마법사만 가능 HEALER : 힐러만 가능	

넘버링과 Index, ID

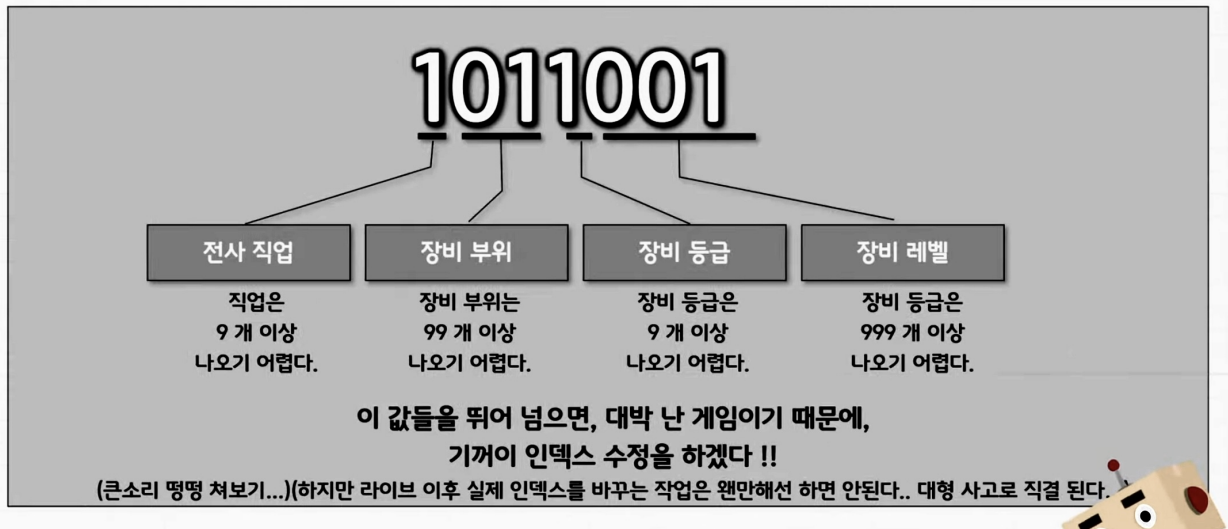
넘버링: 단순히 순차적으로 번호를 부여하는 것, 의미를 갖지 않는다 ≠ Index, ID

Index: 효율적인 테이블 탐색을 위해 의미가 있는 숫자를 사용한다. 한번 정해지면 쉽게 변경되지 않는 것에 Index를 부여하는 경우가 많다. (ex. 스킬, 장비, 재화, 몬스터, NPC 등)

아이템의 index 는 여러가지 의미로 폭 넓게 사용 된다. mmo rpg 프로젝트를 진행 하다 보면, 다양한 의미의 최적화 작업을 위해서 index 가 구분이 되어 있는 것이 좋다. 이는 세분화 되어 있고 규칙화 되어 있을 수록 추후 서버에서 다양한 업무를 수행 함에 있어서 최적의 속도를 뽑아낼 수 있다. (물론 이 부분 역시 프로젝트 마다 다르다.)	
짧은 경험을 바탕으로 소개하자면, 일반적으로 사람이 생각하기에 item 테이블이 있고, 이 테이블 안에 equip 이라는 type 이 존재하고, 여기에 warrior 라는 직업 구분이 있다고 생각하면, 그 안에 있는 것들은 몇 개 안되니까~ 라고 생각 할 수 있다. 그런데 프로그램에선, item 테이블 전체 를 썩 돌면서 먼저 equip 들을 찾는다. 찾은 equip 을 썩 다 돌면서 거기서 warrior 를 찾는다. 그리고 warrior 안에서 또 찾을게 있다면 그 안에서 찾는다. 이와 같이 서버 단에서 무언가를 검색 (특히 거래소)(혹은 무언가 인게임내에서 검색)(혹은 CS 운영툴 혹은 로그 조회 등) 할 경우, 찾는 방식이 이미 정해진 범위 안에서 찾는게 아니라 한 번 돌아서 걸러내고, 또 돌아서 걸러내고, 또 돌아서 걸러내고 하는 방식 이다. 이럴 때, 미리 정의된 값들이 존재한다면, 이 부분이 확 줄어 든다. 이렇게 하기 위해서 데이터 작업자(보통 기획자) 와 서버가 같이 정의한 값들이 존재 한다면, 서버 단에서 작업을 하는 시간이 매우 줄어 든다. 데이터가 많은 mmo rpg 일수록 이런 방법이 최적화에서 많이 사용되기 때문에 index 를 정하는 룰을 지키는게 좋다. 또한, 기획자 입장에서 앞의 몇 자리 정도만 보고서도 그게 무엇인지 알 수 있는 것도 좋다. 이 부분은 역시나 데이터 양이 많아지면, index 를 여기저기 가져다 쓰는 경우가 많이 생기는데, 이럴 때 좋은 방법 중에 하나 이다. 물론 최근에는 enum 값으로 작성하는 경우가 많기 때문에 예전처럼 index 로 확인 하는 경우가 줄고 있긴 하다.	

예시를 들어 보여드리자면,

item index range		
종류	시작값	끝 값
currecny	1	99
equip 관련 우측의 이미지 참조	1000001	9999999
소모품 관련		
퀘스트 관련		



아이템 테이블 채우기

빈란을 어떻게 해석할 지는 프로그래머와 미리 약속되어야한다.

항목 한글 이름	항목 영어 이름	레코드 값 예시
주소	index	1011001
이름	name	전사 무기
설명	description	전사 직업의 첫번째 일반 무기
아이템 설명	itemInfoTxt	흔하게 볼 수 있는 평범한 검이다.
아이템 종류	itemType	equip
재화 타입	currencyType	gold
장비 부위	parts	WEAPON
장비 등급	equipGrade	NORMAL
장착 레벨 제한	equipLv	1
장착 직업 제한	equipClass	WARRIOR
유료 무료 구분	buyType	FREE
남은 시간	remainTime	
목표 날짜	targetDate	
아이콘 파일 위치	iconPath	/asset/resource/UI/equip/icon/WarriorWeapon_Normal_Lv1.png
리소스 파일 위치	resourcePath	/asset/resource/equip/weapon/WarriorWeapon_Normal_Lv1.prefab
스킬 id	skillIndex	
....