



3주차 - 데이터 테이블 다루기

문잡이 [21년차 게임 기획자] - 기획 강의 3 번째 - 데이터 테이블

https://www.youtube.com/live/wRajv0ARpjE?si=msjE_RqTD6dJD6oF

데이터 테이블

: 많은 데이터를 정리 및 관리하기 위한 테이블

e.g. 주인공 자신의 성장, 적, 스테이지(콘텐츠)를 이루는 테이블...

주인공 자신과 관련된 테이블

: 캐릭터, 장비 - 등급 및 강화 관련 내용 추가, 소모품, 스킬 - 성장 포함, 스탯 - 성장 포함, 레벨(+exp) - 성장 포함

적과 관련된 테이블

: 몬스터, 몬스터 스킬, 몬스터 사망 시 드랍, 몬스터 AI

스테이지와 관련된 테이블

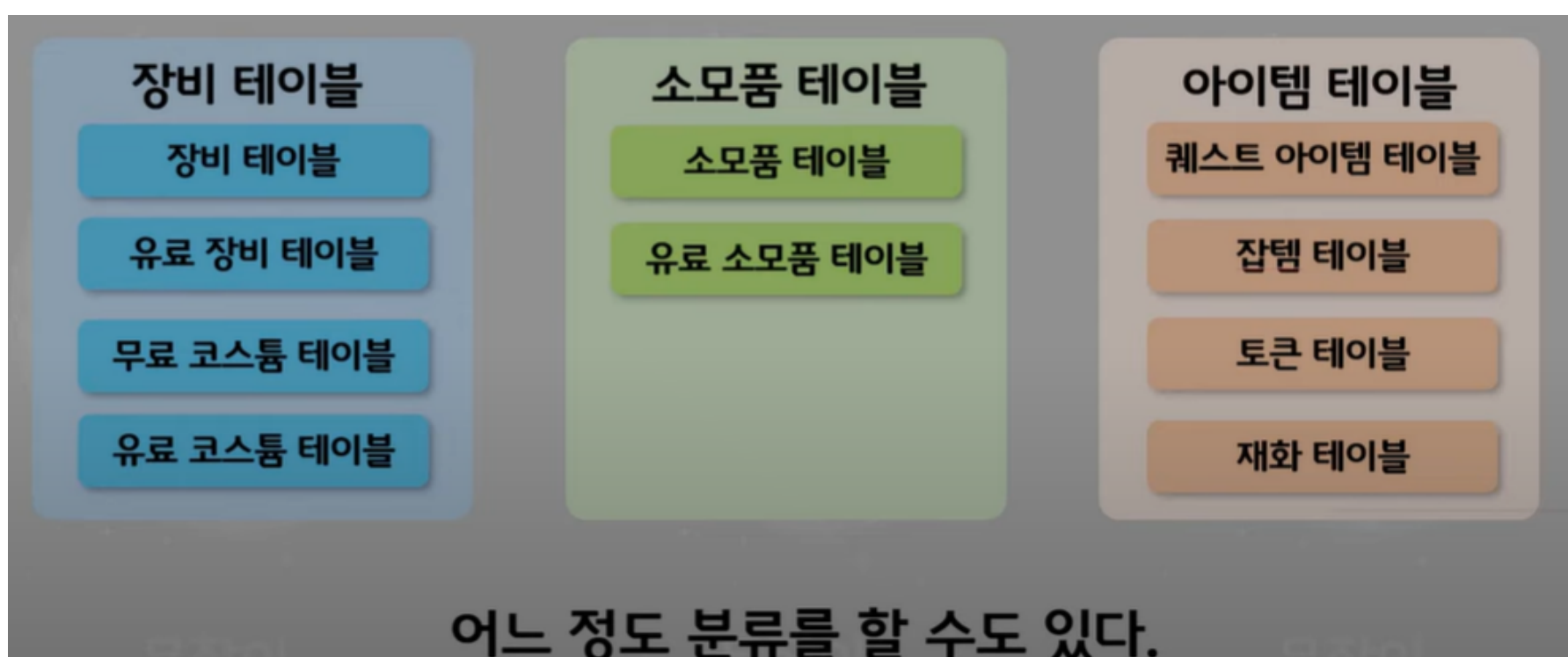
: 맵, NPC, 퀘스트, 콘텐츠 - 하나 or 콘텐츠 별로
+ NPC/몬스터/이벤트 스폰너 - 맵에 각각을 배치

⇒ 위 데이터는 보통 맵들을 통해 작업, 틀에서 나온 데이터를 사용 → 프로젝트 바이 프로젝트
결과물이 txt, json, 바이너리 등 다양한 형태로 나오고 그것을 엔진에서 사용

아이템 테이블 작업

: 아이템에는 무엇이 있을까? 장비, 소모품, 잡템... 끝?

⇒ 장비, 소모품, 잡템, 재화, 퀘스트 아이템, 각종 토큰, 유료 소모품, 유료 장비, 유/무료 코스튬



테이블을 잘게 쪼갰을 때?

장점

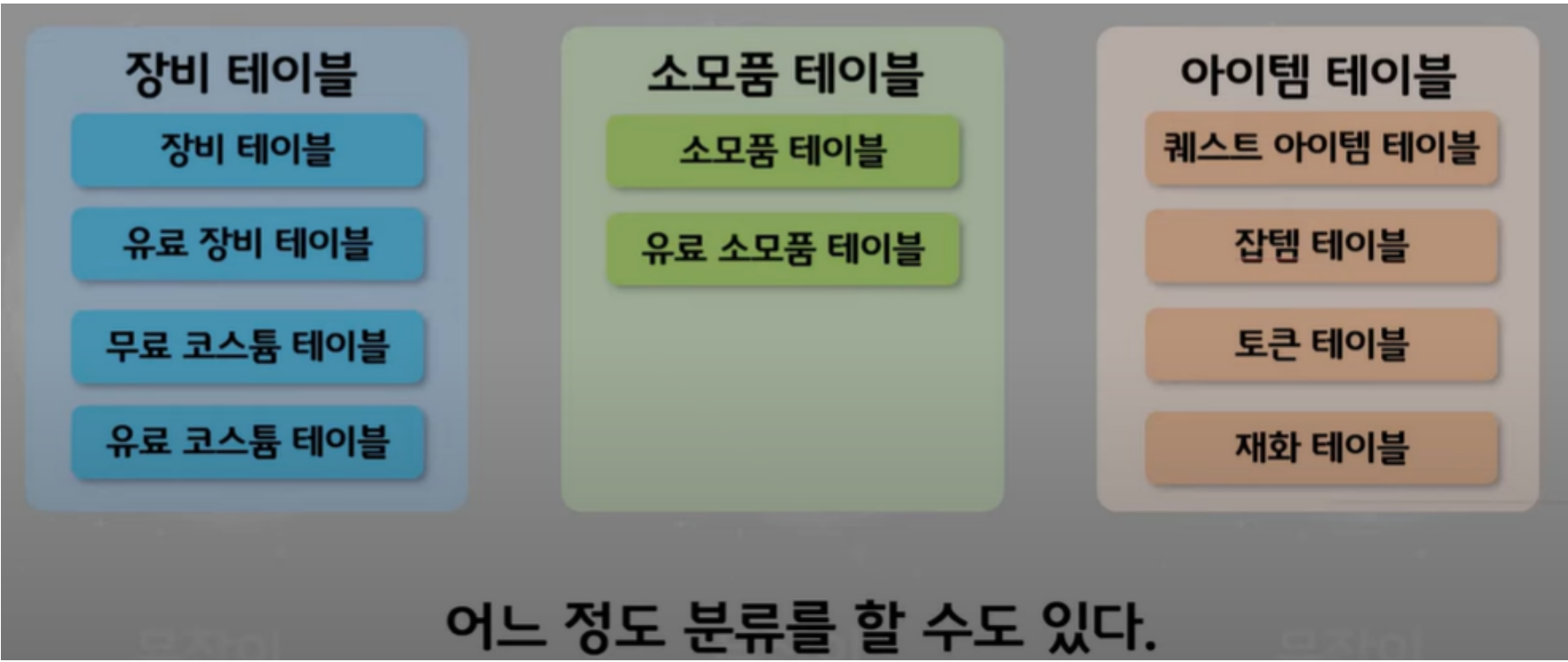
: 작업의 병목 현상 X - 테이블 1개인데 작업자가 여럿일 때 생기는 이슈 X

: 크리티컬한 버그 → 당장 해결이 어렵다면 해당 테이블만 돌리면 됨

단점

- : 데이터가 쌓일수록 다양한 테이블을 열어놓고 작업해야 함
- 하나의 요소 수정을 위해서 여러 개의 테이블을 열어놔야 함
- 보통 연관된 요소가 많아 테이블 하나로 끝나지 않는 경우가 많음
- : 밸런스 작업자들이 고생함
- 밸런스 작업자는 전체 테이블을 수정해야 하는 경우가 생김
- 무수하게 많은 테이블에 흩어져 있는 데이터들을 수정해야 할 수도 있음

아이템 테이블 요소



컬럼명(필드명)	컬럼명(영어)	설명	요소들	자료형
index	index	주소를 뜻한다.	아이템 테이블에서는 index로 사용한다. 상세 내용은 말로 푼다.	int
이름	name	아이템의 이름을 작성한다.	보통 글로벌을 염두에 두면, string 테이블을 따 로 만든다.	string
설명 (실제 사용 X)	description	실제 사용되는 내용이 아닌, 개발자가 확인하기 위한 용 도의 칼럼	개발자가 헛갈릴 만한 요소를 간략하게 작성하여 최소한의 방어용도로 사용한다.	string
아이템 설명	itemInfoTxt	아이템의 설명 내용을 작성한다.	글로벌을 염두에 두면, string 테이블의 key 값을 작성한다.	string
아이템 종류	itemType	아이템 테이블에 하나의 아이템 속성만 가지는 게 아니므 로, 아이템의 속성을 구분하기 위한 종류(type) 컬럼을 추가한다.	equip: 장비 consume: 소모품 currency: 재화 quest: 퀘스트용 아이템 token: 토큰 material: 재료 misc(miscellaneous): 잡템 costume: 코스튬	string
재화 타입	currencyType	재화의 타입을 뜻한다. 재화의 key 값을 정의하기 위한 요소.	cash: 현금 diamond: 유료 재화 ruby: 무료 재화 gold: 게임 내 재화	string
장비 부위	parts	장비일 경우 부위를 구분한다.	weapon: 무기 armor: 방어구 ringFirst: 반지 ringSecond: 반지	string
장비 등급	equipGrade	장비의 등급을 작성한다.	normal: 일반 magic: 마법 rare: 희귀 epic: 에픽 legend: 전설	string
장착 레벨 제한	equipLv	장착 레벨 제한을 작성한다.	장비를 장착할 수 있는 레벨 제한을 작성한다.	int
장착 직업 제한	equipClass	장착 직업을 작성한다.	장비를 장착할 수 있는 직업을 작성한다. ALL: 모두 가능 FIGHTER: 전사만 가능 MAGE: 마법사만 가능 HEALER: 힐러만 가능	string

유료 무료 구분	buyType	유료인지 무료인지를 구분한다.	Free: 무료 Paid: 유료	string
남은 시간	remainTime	시간 제한 아이템의 남은 시간을 작성한다.	남은 분을 입력한다. 하루 = 576분 일주일 = 4032분	int
목표 날짜	targetDATE	특정 날짜 제한인 경우 날짜와 시간을 작성한다.	yyyymmdd:hhmmss 형태로 입력한다. 20230903:240000	string
아이콘 파일	iconPath	아이콘의 경로와 파일 이름을 작성한다.	/asset/resource/UI/equip/icon/파일명.png	string
스킬 id	skillIndex	스킬의 인덱스를 연결한다.	스킬 테이블의 index를 작성한다. 소모품의 경우 스킬을 연결하여 사용한다.	int
소모품 그룹	consumeGroup	물약의 경우 그룹으로 묶어 쿨타임을 같이 적용한다.	그룹 id를 작성한다. 임의의 값으로 작성한다. 소모품을 사용하면 같은 그룹 id끼리는 쿨타임을 공유한다.	int
리소스 파일	resourcePath	리소스의 경로와 파일 이름을 작성한다.	/asset/resource/EQUIPMENT/파일명.prefab	string
구입 재화 종류	buyCurrencyType	구입 시 사용하는 재화의 종류를 작성한다.	currency 타입들의 enum 값을 작성한다. 보통 많이 쓰는 형태는 아래와 같다. diamond: 유료 다이아 ruby: 무료 다이아 gold: 게임 내 재화	string
구입 금액	buyCost	구입 금액을 작성한다.	구입 시의 금액을 작성한다.	int
판매 후 획득 재화 종류	sellCurrencyType	판매 시 획득하는 재화의 종류를 작성한다.	구입 시와 동일하다.	string
판매 금액	sellCost	판매 금액을 작성한다.	판매 시의 금액을 작성한다.	int
구입 가능 여부	useBuy	구입 가능 여부	TRUE: 구입 가능 FALSE: 구입 불가능	boolean
판매 가능 여부	useSell	판매 가능 여부	TRUE: 판매 가능 FALSE: 판매 불가능	boolean
거래 가능 여부	useTrade	거래소 등록 가능 여부 개인 거래가 있을 경우 따로 추가한다. isPrivateTrade 비슷한 형태	TRUE: 등록 가능 FALSE: 등록 불가능	boolean
파괴 가능 여부	useDestroy	파괴 가능 여부 특정 아이템의 경우 파괴만 가능하도록 만든다.	TRUE: 파괴 가능 FALSE: 파괴 불가능	boolean
스탯 id	statIndex	스탯의 인덱스를 작성한다.	스탯 테이블의 인덱스를 작성한다.	int
사용 레벨 제한	useLv	사용 레벨 제한	이 아이템을 사용할 수 있는 레벨 제한을 작성한다.	int

주소	이름	설명	아이템 설명	아이템 종류	재화 타입	장비 부위	장비 등급
index	name	description	itemInfoTxt	type	currencyType	parts	equipGrade
1	현금	실제 현금. 단, 빈 데이터로 보면 된다.		currency	cash		
2	유료 다이아	현금으로 구입하는 재화. 보통 푸른 다이아로 많이 사용 한다.		currency	daimond		
3	무료 다이아	이벤트 등으로 푸는 다이아.		currency	ruby		
4	골드	게임 내에서 거래되는 재화. 골드를 많이 사용 한다.		currency	gold		
1001	HP 물약(소)	HP 물약 가장 처음 것을 만든다.	HP 를 100 회복 시켜주는 빨간 물약	consume			
1010001	FIGHTERWEAPONNORMAL1	전사 직업의 첫 번째 일반 무기.	흔하게 볼 수 있는 평범한 검	equip		WEAPON	NORMAL
1010002	전사 무기					WEAPON	NORMAL
1011001	전사 무기	전사 직업의 첫 번째 마법 무기.				WEAPON	MAGIC
1012001	전사 무기	전사 직업의 첫 번째 회귀 무기.				WEAPON	RARE
1013001	전사 무기	전사 직업의 첫 번째 에픽 무기.				WEAPON	EPIC
1014001	전사 무기	전사 직업의 첫 번째 전설 무기.				WEAPON	LEGEND

// 실제 사용 예시

아이템의 index

: 세분화/규칙화 되어 있을수록 추후 최적화 작업에 용이함

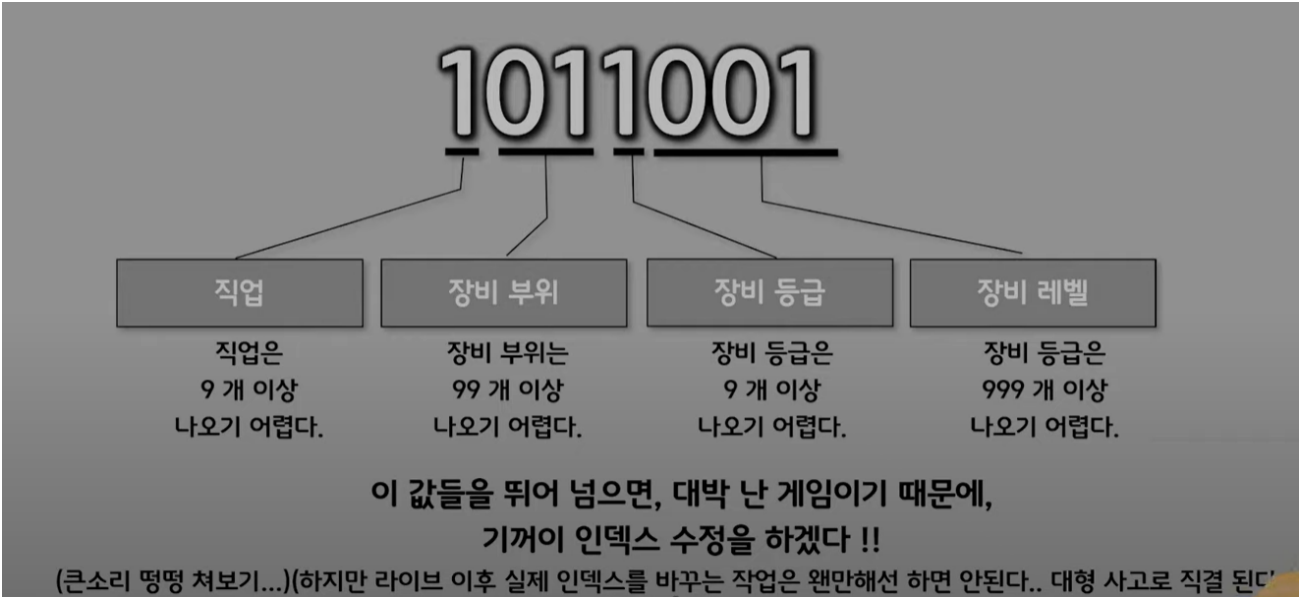
- 프로그램에서는 item 테이블 전체를 싹 돌면서 equip을 찾고,
- 찾은 equip을 싹 돌면서 거기서 warrior를 찾고,
- 그리고 warrior 안에서 또 찾을 게 있다면 그 안에서 찾는다.

→ 검색 = 이미 정해진 범위에서 찾기 X 다 돌면서 걸러내기 O

⇒ 미리 정의된 값들이 존재한다면 이 프로세스가 감소 & 서버에서의 작업 시간 down

⇒ index를 정하는 룰을 지키는 것이 좋음

item index range		
종류	시작값	끝 값
currecny	1	99
소모품 관련	1001	9999
퀘스트 관련	10001	99999
equip 관련 우측의 이미지 참조	1000001	9999999



// 예시

- + 최근에는 enum 값으로 작성하는 경우가 많아 예전처럼 index로 확인하는 경우가 줄고 있기는 함
- + **enum**: Enumeration의 약자. 열거, 목록이라는 뜻을 가짐. 요소, 멤버라 불리는 명명된 값의 집합을 이루는 자료형. JAVA에서는 final로 String, int 등을 나타내는 일정 개수의 기본 자료형의 값을 고정할 수 있다.