이번엔 밸런스데이터와 세이브데이터를 분리해 SQL문이 너무 복잡해지지 않도록 한다

1. 빌딩

Table Building

|  |  |  |
| --- | --- | --- |
| 데이터타입 | 이름 | 설명 |
| Integer primary key | Building\_id | 건물의 id |
| Text | Name | 건물의 이름 |
| Number(3) | Image | 건물 이미지 파일 이름 |
| Text(30) | Description | 건물에 대한 설명 |
| Number(2) | Human | 건물에 필요한 사람 수 |
| integer | button\_image\_id | 이 빌딩이 빌드버튼일 때 나타나는 그림 |

Table ExtendedBuilding

|  |  |  |
| --- | --- | --- |
| 데이터타입 | 이름 | 설명 |
| Integer primary key | Building\_id | 건물의 id |
| Integer | Terrain\_id | 지역 id |

Table BuildingFunction

|  |  |  |
| --- | --- | --- |
| 데이터타입 | 이름 | 설명 |
| Integer primary key | Function\_id | 기능의 id |
| Integer | Building\_id | 건물의 id |
| Integer | Resource\_id | 증가 자원 id |
| Integer | Value | 자원 증가량 |
| Integer | Add\_value | 이득 지역을 통한 추가량 |
| Number(1) | Type | 해당 기능의 종류  1. 패시브 : 매턴 반영  2. 건물 생성시 반영  3. 다음턴  4. 일반기능 : 사용자가 버튼터치  5. 월드맵  6. 예약 |
| Number(1) | Lack\_type | 자원 증가량이 마이너스이다. 부족시 어떤 처리를 해야할까  1. 불가능  2. 경고 |
| Text | Name | 기능의 이름 |
| integer | button\_image\_id | 해당 기능이 버튼으로 나타날 때 어떤 이미지를 사용하는가 |

무기 업그레이드는 무기 레벨을 리소스에 두면 됨

Table BuildingView

|  |  |  |
| --- | --- | --- |
| 데이터타입 | 이름 | 설명 |
| Integer primary key | view\_id | 리소스 표시뷰의 id |
| Integer | Building\_id | 건물의 id |
| Integer | Resource\_id | 보여줄 리소스 |

2. 자원

Table Resource

일부데이터는 매번 수정이 필요하다. 뷰를 써도 좋을 것 같긴 한데, 어떻게 사용하면 좋을까?

|  |  |  |
| --- | --- | --- |
| 데이터타입 | 이름 | 설명 |
| Integer primary key | resource\_id | 자원의 id |
| Text | Name | 자원의 이름 |
| Number(2) | Type | 자원의 타입  1. 소모성  2. 식량  3. 비소모성  4. 인구 |
| Number(3) | Image | 이미지 파일 이름 |

Table Maintain

한 자원을 유지하기 위해 소모되는 또 다른 자원

|  |  |  |
| --- | --- | --- |
| 데이터타입 | 이름 | 설명 |
| Integer primary key | maintain\_id | 유지 항목 id |
| Integer | resource\_id | 유지해야하는 자원 |
| Integer | Consume \_id | 감소해야할 자원 |
| Integer | Consume\_number | 소모할 자원 수 |

사람/병사

빵/고기

Table Terrain

|  |  |  |
| --- | --- | --- |
| 데이터타입 | 이름 | 설명 |
| Integer primary key | terrain\_id | 지역 id |
| Text | Name | 지역 이름 |
| Number(3) | Image | 이미지 파일 이름 |

Table UnitKind

|  |  |  |
| --- | --- | --- |
| 데이터타입 | 이름 | 설명 |
| Integer primary key | Unitkind\_id | 유닛 종류 id |
| Number(1) default 2 | Type | 1 : 성주  2 : 일반사람(소속건물에 좌우됨)  3 : 병사 1렙  4 : 병사 2렙  5 : 거지 |
| Integer | Building\_id | 소속된 건물 |
| Number(3) | Image | 이미지 파일 이름 |
| Text(30) | Description | 유닛에 대한 설명이다 |

3. 맵

Table Tile

이에 대해서는 실제 입력은 OX같은 문자열 나열을 통해 타일을 정한 뒤 이를 디비로 임포트하는 클래스를 만들어 해결하는 것이 좋아보임

|  |  |  |
| --- | --- | --- |
| 데이터타입 | 이름 | 설명 |
| Number(2) | x | 타일의 x |
| Number(2) | y | 타일의 y |
| Integer | Terrain\_id | 지역 id |
| Integer | Map\_id | 맵 id |

Table Map

|  |  |  |
| --- | --- | --- |
| 데이터타입 | 이름 | 설명 |
| Number(3) | x | 맵의 x |
| Number(3) | y | 맵의 y |
| Integer | Preview\_terrain\_id | 대표격 타일 id |
| Integer | Preview\_building\_id | 대표격 빌딩 id |
| Integer primary key | map\_id | 맵 id |

4. 캠페인

Table Campaign

|  |  |  |
| --- | --- | --- |
| 데이터타입 | 이름 | 설명 |
| integer primary key | campaign\_id | 캠페인의 아이디를 말한다 |
| text | title\_name | 타이틀 이름 |

Table CampaignMission

|  |  |  |
| --- | --- | --- |
| 데이터타입 | 이름 | 설명 |
| integer | campaign\_id | 캠페인의 아이디 |
| integer | mission\_id | 미션 아이디 |
| text | name | 캠페인의 이름 |
| number(1) | event | 1 : resource  2 : building  3 : worldmap  4 : war |
| integer | building\_id | 만족해야하는 빌딩의 id |
| integer | resource\_id | 만족해야하는 자원의 id |
| integer | number | 만족해야하는 최소 개수 |
| number(1) | condition | 1 : 이상  2 : 이하 |
| text(100) | description | 해당 미션에 대한 설명 |

Table CampaignMessage

|  |  |  |
| --- | --- | --- |
| 데이터타입 | 이름 | 설명 |
| integer | campaign\_id | 캠페인의 아이디 |
| integer | chapter\_id | 미션이 나오기전에 나오는 스토리 데이터이며,  미션최대값+1인 경우 미션 완료 후 나오는  메시지를 의미한다 |
| integer | message\_id | 이 순서대로 메시지가 나와야 한다는 의미이다 |
| text | actor | 말을 하고 있는 사람 |
| text(100) | message | 메시지. 말. |

5. 기타

Table ImageName

|  |  |  |
| --- | --- | --- |
| 데이터타입 | 이름 | 설명 |
| integer | image\_id | 이미지 id. |
| integer | type | 어느 테이블에서 쓰는 이미지인가 |
| text | name | 이미지의 설명 이름. 일종의 태그 |
| text | value | 이미지 파일 이름 |