ELDEN RING DATABASE

By Raymond Tomo

TABLE OF CONTENTS

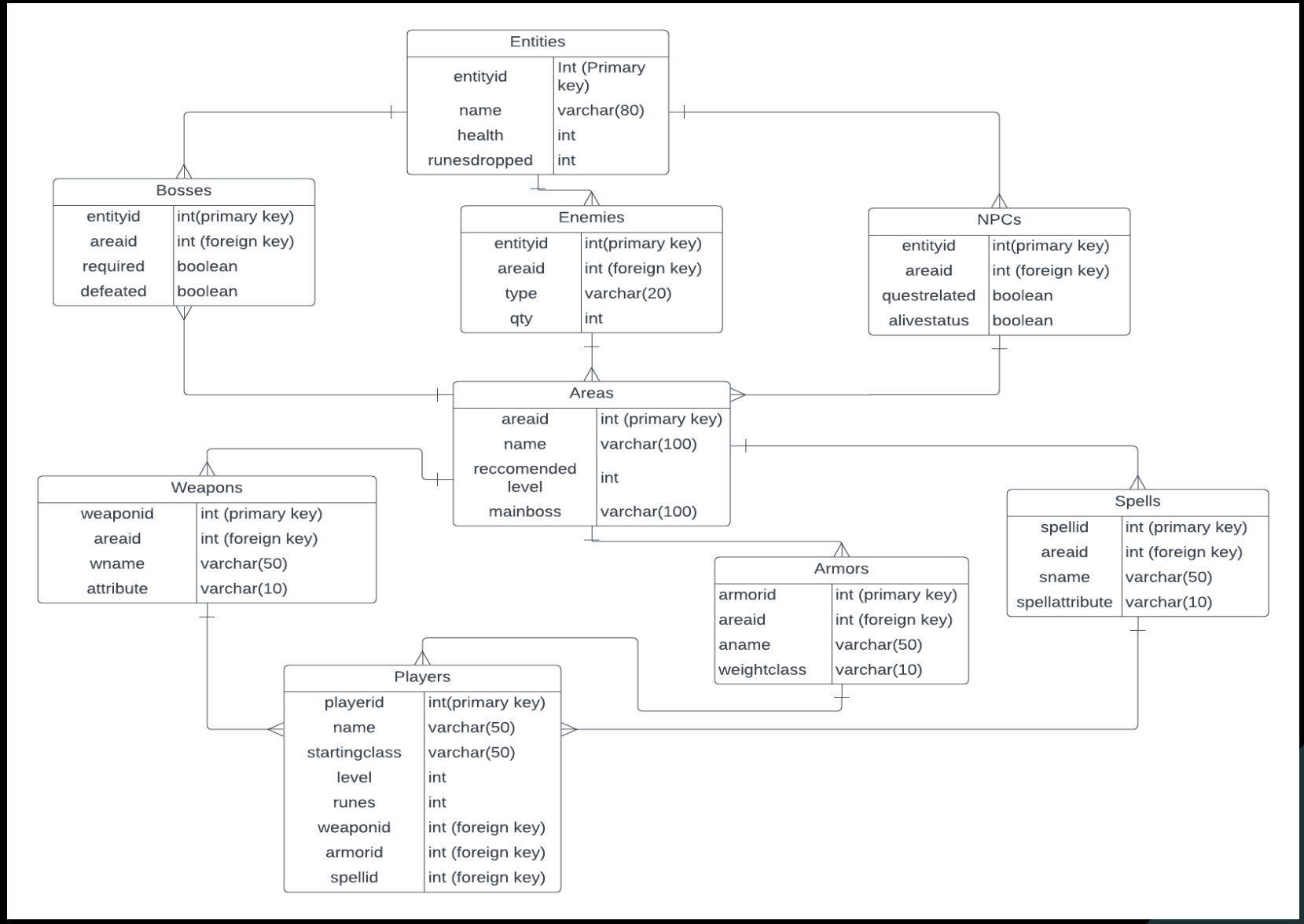
Introduction

In this document, I will outline the implementation for a database that exists within the game Elden Ring. One that will track the positions and items of enemies, items, NPCs, Players, and other entities to create a condensed but comprehensive view of the world. By doing this, it will highlight how some of the triggers and connections within the game would work, such as tying player progression to the number of essential/required bosses they have defeated, or the NPCs that are alive for certain conditions and endings of the game to be met.

"The fallen leaves tell a story. The great Elden Ring was shattered. In our home, across the fog, the Lands Between. Now, Queen Marika the Eternal is nowhere to be found, and in the Night of the Black Knives, Godwyn the Golden was the first to perish. Soon, Marika's offspring, demigods all, claimed the shards of the Elden Ring. The mad taint of their newfound strength triggered the Shattering. A war from which no lord arose. A war leading to abandonment by the Greater Will. Arise now, ye Tarnished. Ye dead, who yet live. The call of long-lost grace speaks to us all. Cross the fog, to the Lands Between. To stand before the Elden Ring. And become the Elden Lord."

Elden Ring Intro Cinematic

ER-Diagram


Entities
entityid
Int (Primary key)
name
varchar(80)
health
int
runesdropped
int
Bosses
entityid
int(primary key)
areaid
int (foreign key)
required
boolean
defeated
boolean
Enemies
entityid
int(primary key)
areaid
int (foreign key)
type
varchar(20)
qty
int
NPCs
entityid
int(primary key)
areaid
int (foreign key)
questrelated
boolean
alivestatus
boolean
Areas
areaid
int (primary key)
name
varchar(100)
reccomended level
int
mainboss
varchar(100)
Weapons
weaponid
int (primary key)
areaid
int (foreign key)
wname
varchar(50)
attribute
varchar(10)
Armors
armorid
int (primary key)
areaid
int (foreign key)
aname
varchar(50)
weightclass
varchar(10)
Spells
spellid
int (primary key)
areaid
int (foreign key)
sname
varchar(50)
spellattribute
varchar(10)
Players
playerid
int(primary key)
name
varchar(50)
startingclass
varchar(50)
level
int
runes
int
weaponid
int (foreign key)
armorid
int (foreign key)
spellid
int (foreign key)

TABLES

Entities Table

This table tracks all of the entities within the game, such as NPCS, enemies, and bosses, all of which are subclasses of this larger entity class. Health and rune value are provided, as that is a trait specific to all entities

```
Create table if not exists  Entities(
entityid int UNIQUE NOT NULL,
name varchar(100) NOT NULL,
health int NOT NULL,
runesdropped int check (runesdropped <= 500000),
Primary key (entityid)
);
```

Entities Sample

	entityid [PK] integer	name character varying (100)	health integer	runesdropped integer
1	1	godrickthegrafted	100000	50000
2	2	renallaqueenofthefullmoon	100000	10000
3	3	starscourgerahdan	200000	25000
4	4	rykardtheserpent	200000	25000
5	5	godfreyphantom	250000	30000
6	6	margitfellomen	2500	2500
7	7	morgottomenking	10000	50000
8	8	moghfellomen	10000	40000
9	9	moghlordofblood	250000	390000
10	10	maleniabladeofmiquella	250000	490000
11	11	malekithblackblade	300000	10000
12	12	godskinduo	300000	10000
13	13	godfreyfirsteldenlord	300000	490000
14	14	ragadonofthegoldenorder	400000	490000
15	15	eldenbeast	400000	500000
16	16	leonineknight	1000	10000
17	17	firegiant	50000	100000
18	18	nightscavalry	50000	10000
19	19	mimictear	50000	23000
20	20	vayre	500	0
21	21	fia	500	0
22	22	rannithewitch	1000	0
23	23	nomadicmerchant	500	0
24	24	dhunterofthedead	1000	0
25	25	patches	10000	0
26	26	loathsomedungeater	10000	0
27	27	goat	10000	0
28	28	rottenhound	10000	50
29	29	giant	10000	1000
30	30	knight	1000	2500
31	31	demihuman	100	220
32	32	dragon	100000	25000
33	33	oracleenvoy	10000	240
34	34	celebrant	1000	2500
35	35	omen	1050	1000

Areas Table

The area table is used to break up the regions of the map into a more manageable area, allowing the game to track where certain enemies and items have been placed within these locales.

```
Create table if not exists Areas(
areaid int UNIQUE NOT NULL,
name varchar(100) NOT NULL,
recommendedlevel int NOT NULL,
mainboss varchar(100) NOT NULL,
Primary key (areaid)
);
```

Areas Sample

	areaid [PK] integer	name character varying (100)	recommendedlevel integer	mainboss character varying (100)
1	1	limgrave	1	godrickthegrafted
2	2	weepingpeninsula	20	leonineknight
3	3	caelid	50	starscourgeradahn
4	4	lirunia	50	renallaqueenofthefullmoon
5	5	altusplain	50	none
6	6	leyndellroyalcapital	60	morgottomenking
7	7	mountaintopofgiants	80	firegiant
8	8	crumblingfaramazula	90	malekithblackblade
9	9	leyndellashencapital	100	eldenbeast
10	10	consecratedsnowfield	100	none
11	11	miquellashaligtree	100	maleniabladeofmiquella
12	12	moghwynpalace	100	moghlordofblood
13	13	noskellaancientcity	60	mimictear

Enemies Table

The basic bread and butter foes one might encounter in the open world, they are a marked decrease in difficulty from the true threats of the game, deserving to be a class of their own.

```
Create table if not exists Enemies(
entityid int NOT NULL,
areaid int NOT NULL,
type varchar(20) NOT NULL,
qty int NOT NULL,
Primary key (entityid),
Foreign Key (areaid)
References Areas(areaid)
);
```

Enemies Sample

	entityid [PK] integer	areaid integer	type character varying (20)	qty integer
1	27	1	animal	100
2	28	3	dog	20
3	29	1	humanoid	30
4	30	5	humaniod	50
5	31	2	humanoid	20
6	32	3	dragon	10
7	33	4	humanoid	20
8	34	12	humanoid	20
9	35	6	omen	20

Bosses Table

The key that ties this game together, bosses are crucial for both the progression of the game, as well as the state of the world as a result of bossses being defeated.

```
Create table if not exists Bosses(
entityid int NOT NULL,
areaid int NOT NULL,
required boolean NOT NULL,
defeated boolean NOT NULL,
Primary key (entityid),
Foreign Key (areaid)
References Areas(areaid)
);
```

Bosses Sample

	entityid [PK] integer	areaid integer	required boolean	defeated boolean
1	1	1	true	true
2	2	4	true	true
3	3	3	true	false
4	4	5	true	false
5	5	5	true	false
6	6	1	true	true
7	7	5	true	false
8	8	5	false	false
9	9	12	false	false
10	10	11	false	false
11	11	8	true	false
12	12	8	true	false
13	13	9	true	false
14	14	9	true	false
15	15	9	false	false
16	16	2	false	true
17	17	7	true	false
18	18	1	false	true
19	19	13	false	false

NPCs Table

Merchants, quest givers and other characters a player might encounter on their journey, their interactions and the fates of these characters are important to the ending the player ultimately receives.

```
Create table if not exists NPCs(
entityid int NOT NULL,
areaid int NOT NULL,
questrelated boolean NOT NULL,
alivestatus boolean NOT NULL,
Primary key (entityid),
Foreign Key (areaid)
References Areas(areaid)
);
```

NPCs- Sample

	entityid [PK] integer	areaid integer	questrelated boolean	alivestatus boolean
1	20	1	true	false
2	21	8	true	true
3	22	4	true	true
4	23	1	false	true
5	24	1	true	true
6	25	1	true	false
7	26	6	true	true

Weapons Table

The purpose of this table is to store the potential weapons the players might use and pick up over the course of the game, with their equipped weapon being tracked in the player section

```
Create table if not exists Weapons(
weaponid int UNIQUE NOT NULL,
wname varchar(50) NOT NULL,
attribute varchar(10) NOT NULL,
areaid int NOT NULL,
Primary key (weaponid),
Foreign Key (areaid)
References Areas(areaid)
);
```

Weapons Sample

	weaponid [PK] integer	wname character varying (50)	attribute character varying (10)	areaid integer
1	1	darkmoongreatsword	int	13
2	2	mohgwynssacredspear	arc	12
3	3	swordofnightandflame	dex	4
4	4	boltofgransax	str	6
5	5	riversofblood	dex	7

Armor Table

Just as crucial as weapons, armor allows a player to remain alive and is stored within it's own unique slot in the player's inventory

```
Create table if not exists Armor(
armorid int UNIQUE NOT NULL,
aname varchar(50) NOT NULL,
weightclass varchar(10) NOT NULL,
areaid int NOT NULL,
Primary key (armorid),
Foreign Key (areaid)
References Areas(areaid)
);
```

Armor Sample

	armorid [PK] integer	aname character varying (50)	weightclass character varying (10)	areaid integer
1	1	allknowingset	medium	9
2	2	blackknifeset	light	10
3	3	lusatsset	light	4
4	4	godskinapostleset	medium	3
5	5	radahnset	heavy	3

Spells Table

A useful complement to weapons, spells are priortized by players who go for builds revolving around sorceries, faith incantations, and other items.

```
Create table if not exists Spells(
spellid int UNIQUE NOT NULL,
sname varchar(50) NOT NULL,
spellattribute varchar(10) NOT NULL,
areaid int NOT NULL,
Primary key (spellid),
Foreign Key (areaid)
References Areas(areaid)
);
```

Spells- Sample

	spellid [PK] integer	sname character varying (50)	spellattribute character varying (10)	areaid integer
1	1	cometazure	int	5
2	2	burnoflame	faith	7
3	3	rannisdarkmoon	int	4
4	4	eldenstars	int	9
5	5	greyollsroar	faith	3

Players Table

This table contains all of the players currently saved to a local copy of Elden Ring, rather than an online database. This includes the standard values a player would encounter on their HUD

```
Create table if not exists Players(
playerid int UNIQUE NOT NULL,
name varchar(50) NOT NULL,
startingclass varchar(50) NOT NULL,
level int NOT NULL,
runes int NOT NULL,
weaponid int NOT NULL,
armorid int NOT NULL,
spellid int NOT NULL,
Primary key (playerid),
Foreign Key (weaponid)
References Weapons(weaponid),
Foreign Key (armorid)
References Armor(armorid),
Foreign Key (spellid)
References Spells(spellid)
);
```

Players - Sample

	playerid [PK] integer	name character varying (50)	startingclass character varying (50)	level integer	runes integer	weaponid integer	armorid integer	spellid integer
1	1	ironpineapple	knight	120	2000	3	5	5
2	2	alanthewizard	astrologer	60	0	1	3	5
3	3	timtheenchanter	prophet	200	500000	2	4	2


REPORTS

EntitiesWithTheMostRuneValue

As runes are one of the more important aspects of the game as they help the player level up and progress, having them as an accessible list would be important to the developers.

```
Select Entities.name, Entities.entityid,
    Entities.runesdropped
From Entities
Group by Entities.entityid
Order by runesdropped DESC;
```

Get Bosses With The Most Rune Value -Output

	name character varying (100)	entityid [PK] integer	runesdropped integer
1	eldenbeast	15	500000
2	ragadonofthegoldenorder	14	490000
3	maleniabladeofmiquella	10	490000
4	godfreyfirsteldenlord	13	490000
5	moghlordofblood	9	390000
6	firegiant	17	100000
7	godrickthegrafted	1	50000
8	morgottomenking	7	50000
9	moghfellomen	8	40000
10	godfreyphantom	5	30000
11	starscourgerahdan	3	25000
12	rykardtheserpent	4	25000
13	dragon	32	25000
14	mimictear	19	23000
15	nightscavalry	18	10000
16	renallaqueenofthefullmoon	2	10000
17	malekithblackblade	11	10000
18	godskinduo	12	10000
19	leonineknight	16	10000
20	knight	30	2500
21	celebrant	34	2500
22	margitfellomen	6	2500
23	omen	35	1000
23	omen	35	1000
24	giant	29	1000
25	oracleenvoy	33	240
26	demihuman	31	220
27	rottenhound	28	50
28	vayre	20	0
29	goat	27	0
30	loathsomedungeater	26	0
31	patches	25	0
32	dhunterofthedead	24	0
33	nomadicmerchant	23	0
34	rannithewitch	22	0
35	fia	21	0

MainStoryBossesCompletion

While many RPGs have their story progression tied to the quests completed, From Software often uses the bosses defeated as a metric of progression in their game. As such, this query and subquery will allow someone to track their progress through the game based on the bosses defeated.

```
Select DISTINCT Entities.name, Entities.entityid
From Entities
Inner Join Bosses
    On Bosses.entityid = Entities.entityid
    Where (Bosses.required = true)
    And Bosses.defeated=true;
```

MainStoryBossesCompletion- Output

	name character varying (100)	entityid [PK] integer
1	renallaqueenofthefullmoon	2
2	margitfellomen	6
3	godrickthegrafted	1


VIEWS

EnemyMasterList View

Because of the many foes the player might encounter, this tab focuses on distinguishing friend from foe by grouping the more dangerous enemies and bosses together, while excluding friendly NPCs.

```
Drop view EnemyMasterList;

Create or replace view EnemyMasterList as

Select DISTINCT Entities.name, Entities.entityid

From Entities, Bosses, Enemies

Where (Entities.entityid = Bosses.entityid) or (Entities.entityid =
    Enemies.entityid);


Select *

From EnemyMasterList;
```

EnemyMasterList View- Output

	name character varying (100)	entityid integer
1	celebrant	34
2	morgottomenking	7
3	ragadonofthegoldenorder	14
4	nightscavalry	18
5	margitfellomen	6
6	firegiant	17
7	maleniabladeofmiquella	10
8	starscourgerahdan	3
9	mimictear	19
10	godfreyfirsteldenlord	13
11	rykardtheserpent	4
12	eldenbeast	15
13	demihuman	31
14	knight	30
15	dragon	32
16	malekithblackblade	11
17	giant	29
18	oracleenvoy	33
19	renallaqueenofthefullmoon	2
20	goat	27
21	moghlordofblood	9
22	omen	35
23	godfreyphantom	5
24	leonineknight	16
25	godskinduo	12
26	moghfellomen	8
27	godrickthegrafted	1
28	rottenhound	28

STORED PROCEDURES

get_bosses_in_area Stored Procedure

This procedure retrieves all of the given bosses within an area, which allows the devs to see which boss is associated with each area.

```
Create or replace function get_bosses_in_area(int,REFCURSOR)
    returns refcursor as
$$
Declare
  selectedareaid  int =$1;
  resultset refcursor = $2;
Begin
  Open resultset for
    Select *
    from bosses
      Where selectedareaid = bosses.areaid;
  Return resultset;
End;
$$
Language plpgsql;
```

get_bosses_in_area Stored Procedure- Output

- select get_bosses_in_area(1, 'results');
- fetch all from results;

	entityid [PK] integer	areaid integer	required boolean	defeated boolean
1	1	1	true	true
2	6	1	true	true
3	18	1	false	true

get_bosses_cleared Stored Procedure

Tackling all bosses rather than story specific encounters, this procedure is a better overview of the player's progress in optional and required content within the world of the game.

```
Create or replace function
    get_bosses_cleared(boolean,
    REFCURSOR)

returns refcursor as

$$

Declare

  defeatedstatus boolean=$1;

  resultset refcursor  = $2;

Begin

  Open resultset for

    Select *

    from bosses

      Where defeatedstatus =
    bosses.defeated;

  Return resultset;

End;

$$

Language plpgsql;
```

get_bosses_cleared Stored Procedure

```
select get_bosses_cleared(true, 'results');
fetch all from results;
```

	entityid [PK] integer	areaid integer	required boolean	defeated boolean
1	1	1	true	true
2	2	4	true	true
3	6	1	true	true
4	16	2	false	true
5	18	1	false	true

TRIGGERS

past_level_cap Trigger

Due to Elden Ring including online multiplayer, there are a number of cheaters who attempt to modify their stats over what the game would allow. Thus, with any information being changed over a specific cap, it will trigger a flag within the database.

```
Create or replace function past_level_cap()
    Returns trigger as
    $$
    Begin
        Delete from Players
        Where level > 713;
    End;
    $$
    Language plpgsql;
Create trigger past_level_cap
after insert on players
For each row
Execute procedure past_level_cap();
```

past_level_cap Trigger

Insert into Players

Values(4,'CheatsMcgee','knight',800,
2000,3,5,5);

```
ERROR:  new row for relation "players" violates check constraint "players_level_check"
DETAIL:  Failing row contains (4, CheatsMcgee, knight, 800, 2000, 3, 5, 5).
SQL state: 23514
```


SECURITY

Player Role

The most basic of all roles, this simply allows a player to access and view their own records.

Create role playerrole;

Grant select on players to playerrole;

Patchreleaseteam Role

Able to update the visible values, these patch release team members are responsible for assisting with the work of bug fixing and other QA related concerns.

Create role patchreleaseteam;

Grant update on Entities,Areas,Bosses,NPCs,Enemies,Players,Weapons,Armor,Spells to patchreleaseteam;

Itembalancedev Role

While limited to only manipulating item information, Itembalancedevs can tweak and balance out overperforming or underperforming weapons within the game.

Create role itembalancedevrole;

Grant all on Weapons,Armor,Spells to itembalancedevrole;

Maindev Role

Being the head of the project, the main developer is given access to everything within the database.

Create role maindevrole;

Grant all on Entities,Areas,Bosses,NPCs,Enemies,Players,Weapons,Armor,Spells to maindevrole;

CLOSING REMARKS

Implementation Notes

Purpose: Because Game Design focuses on a different implementation of databases, most of the queries related to retrieving specific data efficiently, such as the bosses and enemies since Elden Ring is a boss focused game.

Test Data: All of the areas, NPCs, Items, bosses, and other items used are from Elden Ring, specifically the information avaliable on fextralife. Due to some data values such as health or runes being missing for some bosses and currently being mined, some estimates were used.

Known Issues

- Elden Ring is a new game with content still being published. As a result, a good deal of the data is subject to change and may grow over time to a much larger collection of bosses and foes.
- Weapons, spells, and items having their own ids, while helping to maintains BCNF, makes it more difficult to put them all together in a collective master list.
- Currently, there is no implementation for the covenants or multiplayer system that allows for invasions, and moderation functions that keeps cheaters from exploiting the system within the game.

Future Enhancements

- Potentially, I would run this with a static and dynamic database configuration. The static would store values that are not mutable, such as objects and entities are persistent regardless of progression. However, the player data would be kept in a dynamic database that allows for progression.
- Along with this, tying items under one flag may be more useful for the purposes of organization, though it could run the risk of being able to identify a weapon from armor, and attributes not carrying over properly.