Dark Souls Boss Database

Designed by: Mike Mucci

TABLE OF CONTENTS

Executive Summery……………………………………………………………3

Entity-Relationship Diagram…………………………………………………...4

Tables: Create Statements, F(n) Dependencies, Sample data………………...5

Location Table…………………………………………………………………5

Races Table……………………………………………………………………..6

Damage Type Table……………………………………………………………7

Bosses Table……………………………………………………………………8

Weaknesses/Resistances Tables………………………………………………...9

Items Table……………………………………………………………………10

Offensive Items Table…………………………………………………………11

Defensive Items Table…………………………………………………….…...12

Boss Drops Table……………………………………………………….……...13

Views: Definitions and Sample output…………………………………….…...14

Important Boss Information View………………………………………….……14

Boss Item Drop Information View………………………………………….……15

Offensive/Defensive Item Information View…………………………...……...16

Stored Procedures: Definitions and Sample output…………………………….…17

AddItem Procedure……………………………………………………………..…17

Security: Users and Privileges……………………………………………….…19

Administrator/Player Users and Privileges….…………………………………....19

Implementation Notes/Known Problems/Future Enhancements…………….20

Executive summery

Overview

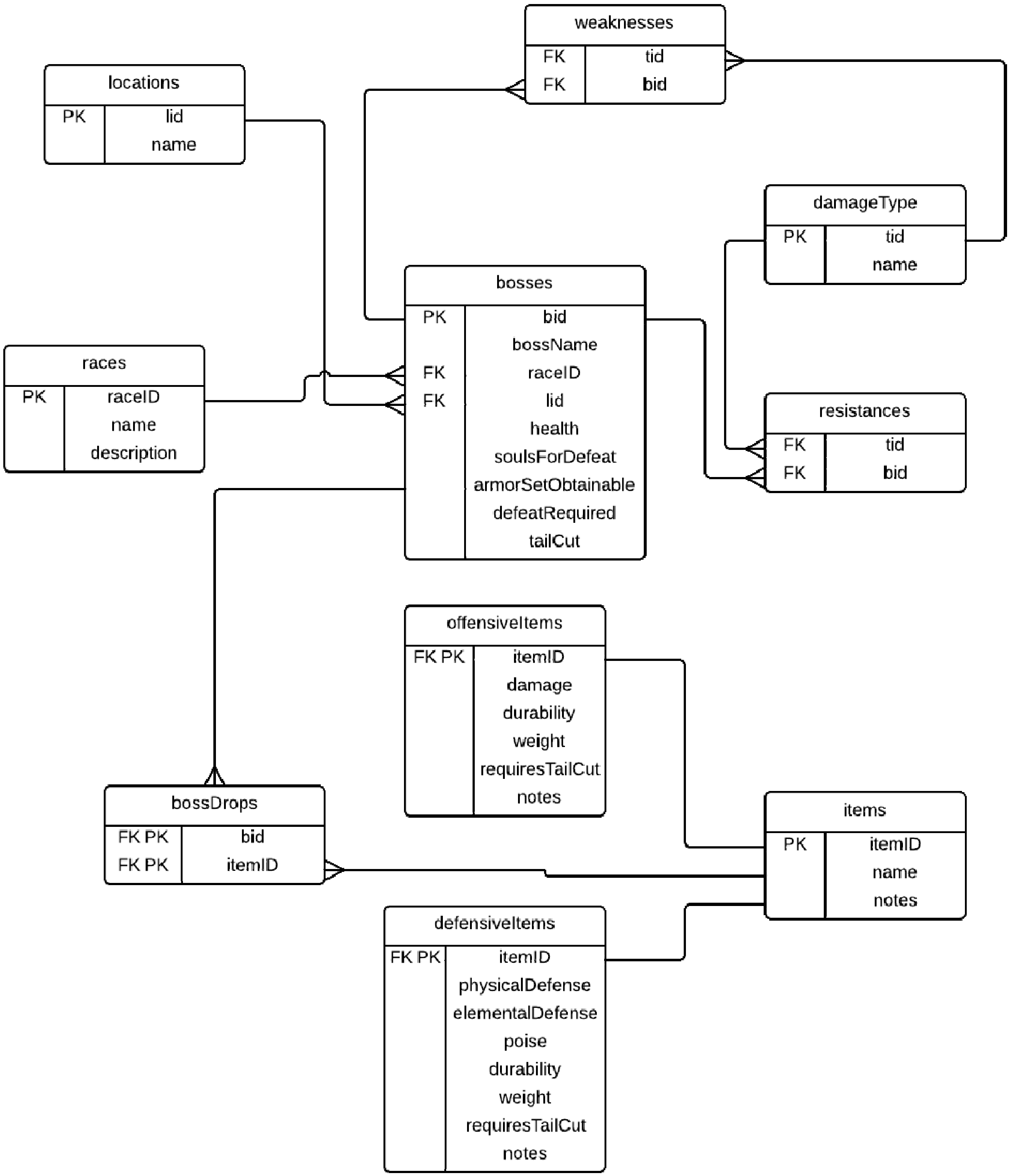
Dark Souls is an action RPG (role-playing game) which takes place in a bleak fictional land called Lordran, which is filled with death a myriad of undead who roam the earth. Anything that breaths has a soul, of which has a certain value. Those undead who do not obtain souls, lose their minds and are called hollow. Undeads are collected and sent to the Undead Asylum, inevitably becoming hollow. However, there is a legend that a chosen undead will leave the Asylum and make way to Lordran to decide the fate of the world.

Objectives

The player takes the role as the Chosen Undead and throughout the story, the player has to defeat a multitude of bosses. The majority of them are very hard and the only way to overcome them is with strategy. The information contained in this database is to help those reading it, whether they be players or anyone else interested in the game, to know as much as possible about the main bosses in the game.

Note: At the end of the game's first play through (NewGame), the player can replay the game but at a higher difficulty, called NewGame+. The differences between NG and NG+ are inconsistent, therefore this database only reflects the data gathered from NG.

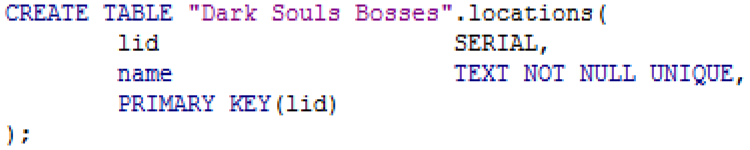
ENTITY-RELATIONSHIP DIAGRAM



Tables

Locations table

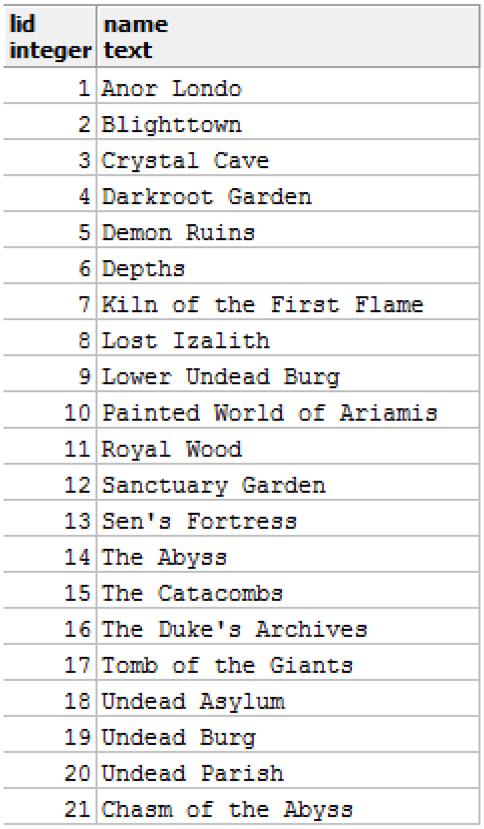
The locations table contains all the locations in the game where a boss can be located. An auto incrementing integer, serial, is used to easily keep the primary key unique.



```
CREATE TABLE "Dark Souls Bosses".locations(
        lid                     SERIAL,
        name                    TEXT NOT NULL UNIQUE,
        PRIMARY KEY(lid)
);
```

Functional dependency: lid→name

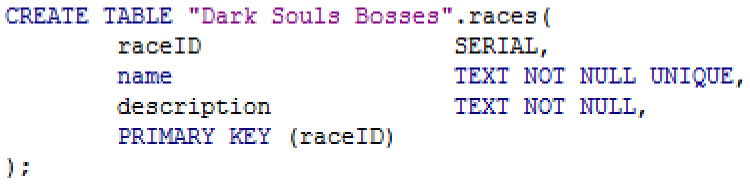
Sample Data:



lid integer	name text
1	Anor Londo
2	Blighttown
3	Crystal Cave
4	Darkroot Garden
5	Demon Ruins
6	Depths
7	Kiln of the First Flame
8	Lost Izalith
9	Lower Undead Burg
10	Painted World of Ariamis
11	Royal Wood
12	Sanctuary Garden
13	Sen's Fortress
14	The Abyss
15	The Catacombs
16	The Duke's Archives
17	Tomb of the Giants
18	Undead Asylum
19	Undead Burg
20	Undead Parish
21	Chasm of the Abyss

RACES TABLE

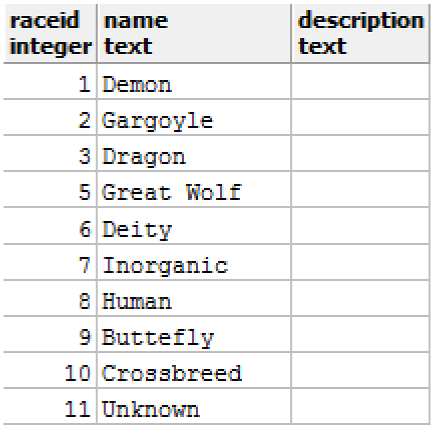
The races table contains all the races that a boss can be.



```
CREATE TABLE "Dark Souls Bosses".races(
        raceID                  SERIAL,
        name                    TEXT NOT NULL UNIQUE,
        description             TEXT NOT NULL,
        PRIMARY KEY (raceID)
);
```

Functional Dependency: raceID→ name, description

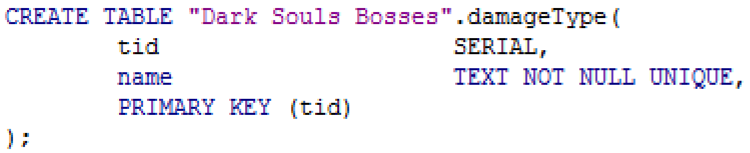
<u>Sample Data:</u>



raceid integer	name text	description text
1	Demon	
2	Gargoyle	
3	Dragon	
5	Great Wolf	
6	Deity	
7	Inorganic	
8	Human	
9	Buttefly	
10	Crossbreed	
11	Unknown	

Damage Type table

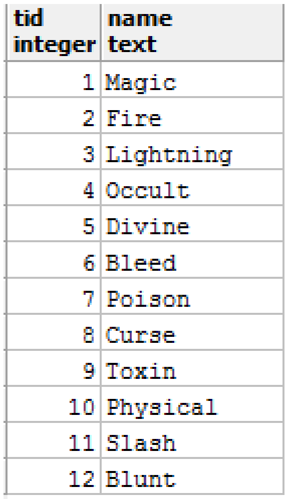
The damage type table contains the elemental, physical, and status damage types that an attack can do.



```
CREATE TABLE "Dark Souls Bosses".damageType(
        tid                         SERIAL,
        name                        TEXT NOT NULL UNIQUE,
        PRIMARY KEY (tid)
);
```

Functional Dependency: tid→name.

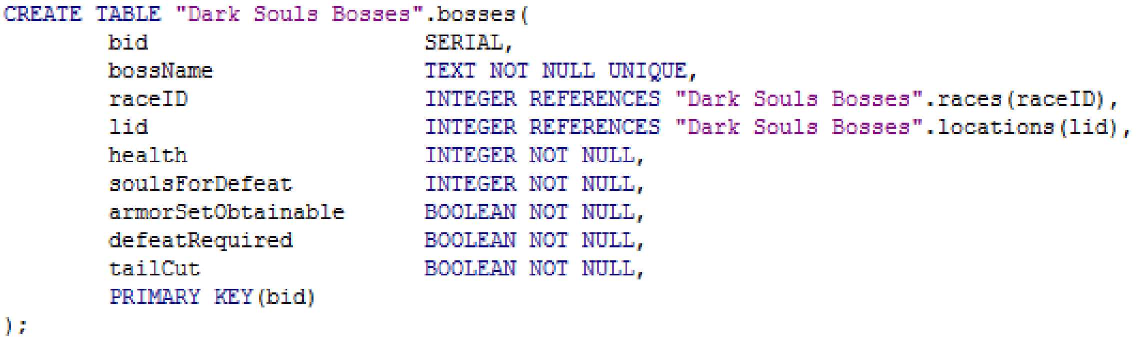
Sample Data:



tid integer	name text
1	Magic
2	Fire
3	Lightning
4	Occult
5	Divine
6	Bleed
7	Poison
8	Curse
9	Toxin
10	Physical
11	Slash
12	Blunt

Bosses table

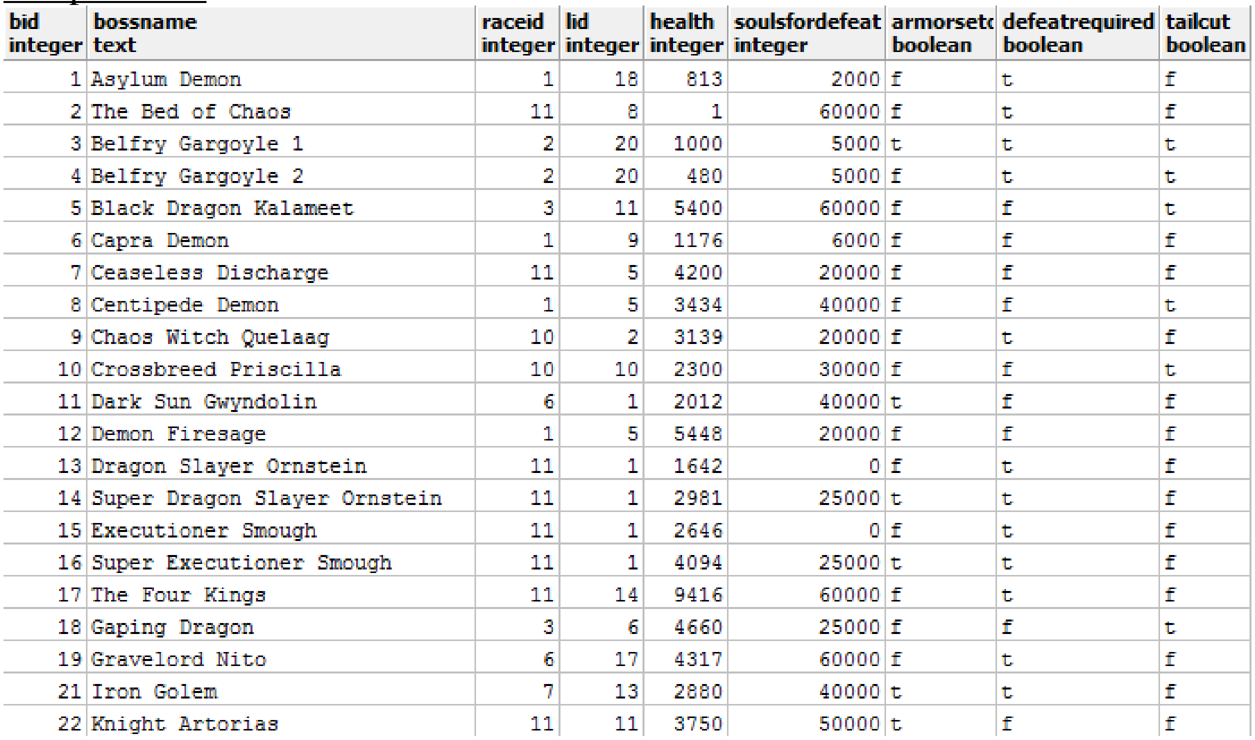
The bosses table contains all the main bosses in the game. Some bosses reappear in the game as common enemies, but as lesser versions of their stronger counterpart; they are weaker and have less health and are thereby excluded.



```
CREATE TABLE "Dark Souls Bosses".bosses(
        bid                     SERIAL,
        bossName                TEXT NOT NULL UNIQUE,
        raceID                  INTEGER REFERENCES "Dark Souls Bosses".races(raceID),
        lid                     INTEGER REFERENCES "Dark Souls Bosses".locations(lid),
        health                  INTEGER NOT NULL,
        soulsForDefeat          INTEGER NOT NULL,
        armorSetObtainable      BOOLEAN NOT NULL,
        defeatRequired          BOOLEAN NOT NULL,
        tailCut                 BOOLEAN NOT NULL,
        PRIMARY KEY(bid)
);
```

Functional Dependency: bid→bossName, raceID, lid, health, soulsForDefeat, armorSetObtainable, defeatRequired, tailCut

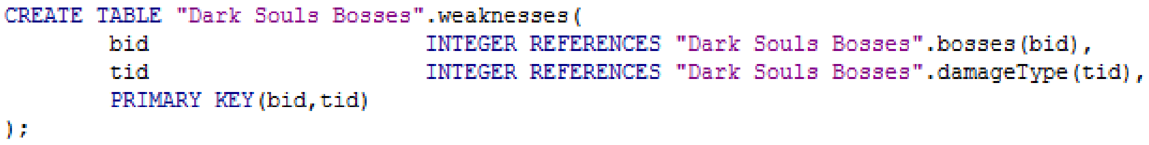
Sample Data:



bid integer	bossname text	raceid integer	lid integer	health integer	soulsfordefeat integer	armorsetc boolean	defeatrequired boolean	tailcut boolean
1	Asylum Demon	1	18	813	2000	f	t	f
2	The Bed of Chaos	11	8	1	60000	f	t	f
3	Belfry Gargoyle 1	2	20	1000	5000	t	t	t
4	Belfry Gargoyle 2	2	20	480	5000	f	t	t
5	Black Dragon Kalameet	3	11	5400	60000	f	f	t
6	Capra Demon	1	9	1176	6000	f	f	f
7	Ceaseless Discharge	11	5	4200	20000	f	f	f
8	Centipede Demon	1	5	3434	40000	f	f	t
9	Chaos Witch Quelaag	10	2	3139	20000	f	t	f
10	Crossbreed Priscilla	10	10	2300	30000	f	f	t
11	Dark Sun Gwyndolin	6	1	2012	40000	t	f	f
12	Demon Firesage	1	5	5448	20000	f	f	f
13	Dragon Slayer Ornstein	11	1	1642	0	f	t	f
14	Super Dragon Slayer Ornstein	11	1	2981	25000	t	t	f
15	Executioner Smough	11	1	2646	0	f	t	f
16	Super Executioner Smough	11	1	4094	25000	t	t	f
17	The Four Kings	11	14	9416	60000	f	t	f
18	Gaping Dragon	3	6	4660	25000	f	f	t
19	Gravelord Nito	6	17	4317	60000	f	t	f
21	Iron Golem	7	13	2880	40000	t	t	f
22	Knight Artorias	11	11	3750	50000	t	f	f

WEAKNESSES TABLE

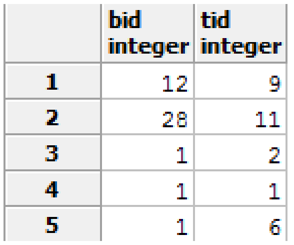
This table contains the damage types/elemental weaknesses of the bosses matched by the boss' ID.



```
CREATE TABLE "Dark Souls Bosses".weaknesses(
        bid                             INTEGER REFERENCES "Dark Souls Bosses".bosses(bid),
        tid                             INTEGER REFERENCES "Dark Souls Bosses".damageType(tid),
        PRIMARY KEY(bid,tid)
);
```

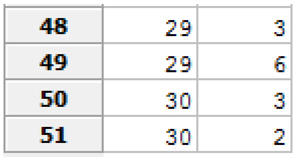
Functional Dependency: bid, tid→

Sample Data:



	bid	tid
	integer	integer
1	12	9
2	28	11
3	1	2
4	1	1
5	1	6

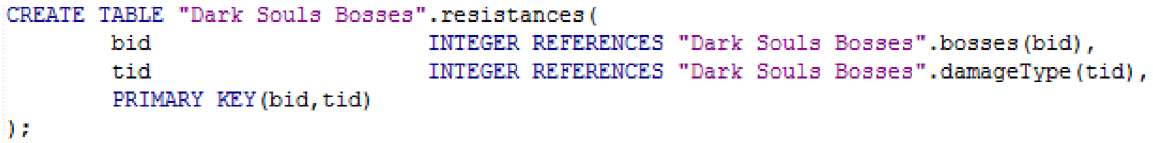
…



48	29	3
49	29	6
50	30	3
51	30	2

RESISTANCES TABLE

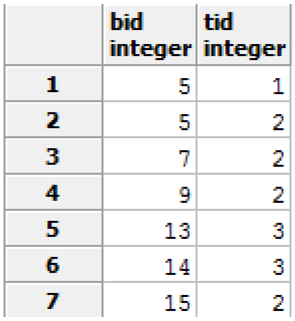
This table contains the damage/elemental resistances of the bosses, again matched by the boss' ID.



```
CREATE TABLE "Dark Souls Bosses".resistances(
        bid                             INTEGER REFERENCES "Dark Souls Bosses".bosses(bid),
        tid                             INTEGER REFERENCES "Dark Souls Bosses".damageType(tid),
        PRIMARY KEY(bid,tid)
);
```

Functional Dependency: bid, tid→

Sample Data:

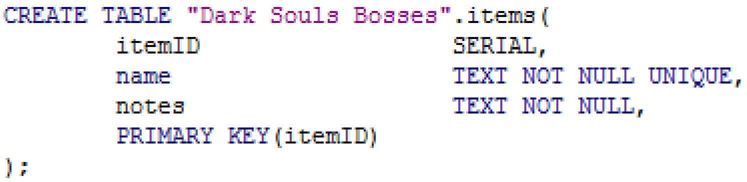


	bid	tid
	integer	integer
1	5	1
2	5	2
3	7	2
4	9	2
5	13	3
6	14	3
7	15	2

…

ITEMS TABLE

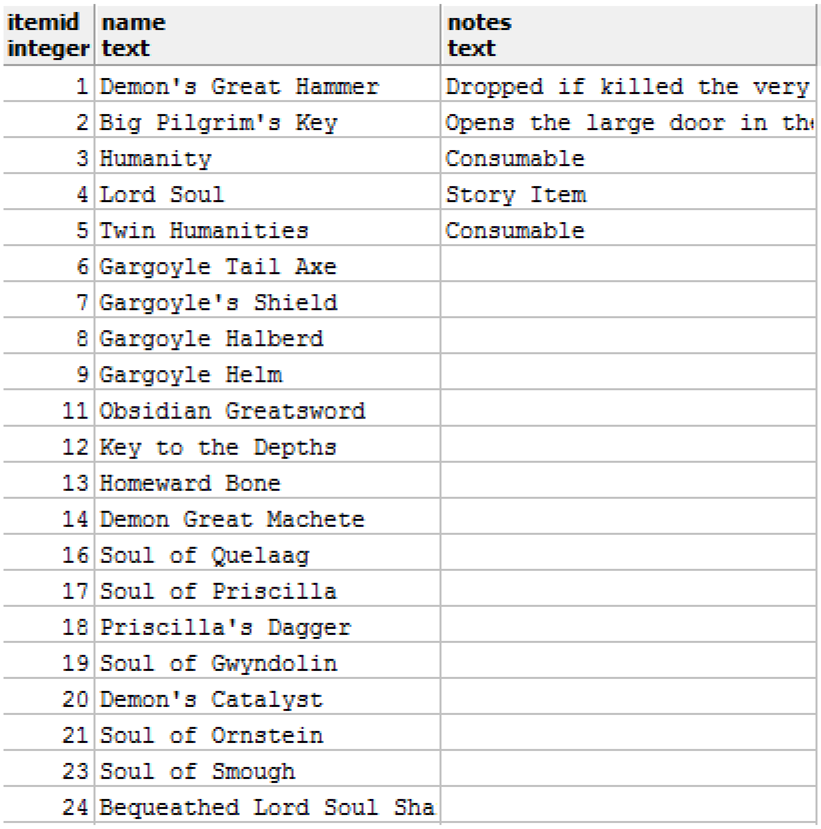
This table contains the all the items that the bosses can drop upon death.



```
CREATE TABLE "Dark Souls Bosses".items(
        itemID                  SERIAL,
        name                    TEXT NOT NULL UNIQUE,
        notes                   TEXT NOT NULL,
        PRIMARY KEY(itemID)
);
```

Functional Dependency: itemID→name, notes

<u>Sample Data:</u>

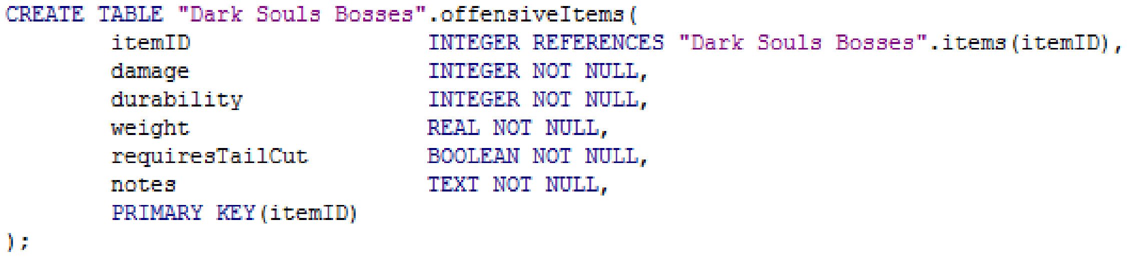


itemid integer	name text	notes text
1	Demon's Great Hammer	Dropped if killed the very
2	Big Pilgrim's Key	Opens the large door in th
3	Humanity	Consumable
4	Lord Soul	Story Item
5	Twin Humanities	Consumable
6	Gargoyle Tail Axe	
7	Gargoyle's Shield	
8	Gargoyle Halberd	
9	Gargoyle Helm	
11	Obsidian Greatsword	
12	Key to the Depths	
13	Homeward Bone	
14	Demon Great Machete	
16	Soul of Quelaag	
17	Soul of Priscilla	
18	Priscilla's Dagger	
19	Soul of Gwyndolin	
20	Demon's Catalyst	
21	Soul of Ornstein	
23	Soul of Smough	
24	Bequeathed Lord Soul Sha	

…

Offensive items table

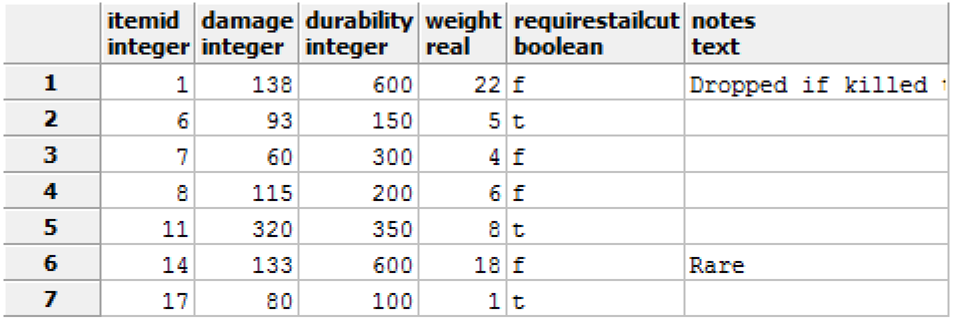
This table contains the information about the offensive items. Offensive items have been categorized as items that can do damage. The ID in this table matches the global itemID in the items table.



```
CREATE TABLE "Dark Souls Bosses".offensiveItems(
        itemID                  INTEGER REFERENCES "Dark Souls Bosses".items(itemID),
        damage                  INTEGER NOT NULL,
        durability              INTEGER NOT NULL,
        weight                  REAL NOT NULL,
        requiresTailCut         BOOLEAN NOT NULL,
        notes                   TEXT NOT NULL,
        PRIMARY KEY(itemID)
);
```

Functional Dependency: itemID→damage, durability, weight, requiresTailCut, notes

Sample Data:

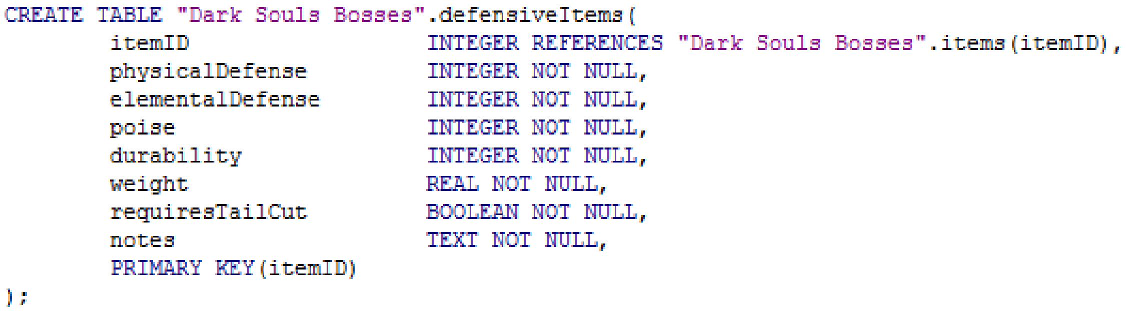


	itemid integer	damage integer	durability integer	weight real	requirestailcut boolean	notes text
1	1	138	600	22	f	Dropped if killed
2	6	93	150	5	t	
3	7	60	300	4	f	
4	8	115	200	6	f	
5	11	320	350	8	t	
6	14	133	600	18	f	Rare
7	17	80	100	1	t	

…

Defensive Items table

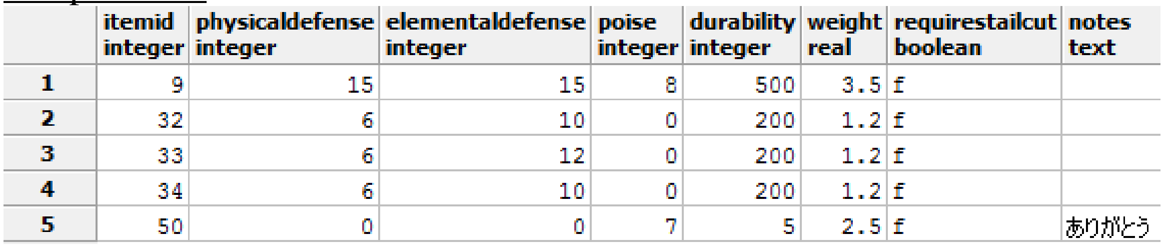
This table contains the information about the defensive items. Defensive items have been categorized as those that protect against damage.



```
CREATE TABLE "Dark Souls Bosses".defensiveItems(
        itemID                  INTEGER REFERENCES "Dark Souls Bosses".items(itemID),
        physicalDefense         INTEGER NOT NULL,
        elementalDefense        INTEGER NOT NULL,
        poise                   INTEGER NOT NULL,
        durability              INTEGER NOT NULL,
        weight                  REAL NOT NULL,
        requiresTailCut         BOOLEAN NOT NULL,
        notes                   TEXT NOT NULL,
        PRIMARY KEY(itemID)
);
```

Functional Dependency: itemID→physicalDefense, elementalDefense, poise, durability, weight, requiresTailCut, notes

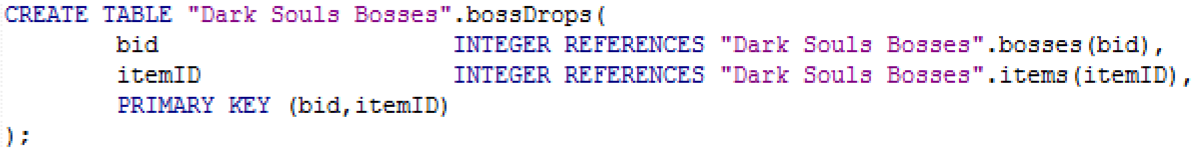
Sample Data:



	itemid integer	physicaldefense integer	elementaldefense integer	poise integer	durability integer	weight real	requirestailcut boolean	notes text
1	9	15	15	8	500	3.5	f	
2	32	6	10	0	200	1.2	f	
3	33	6	12	0	200	1.2	f	
4	34	6	10	0	200	1.2	f	
5	50	0	0	7	5	2.5	f	ありがとう

Boss Drops table

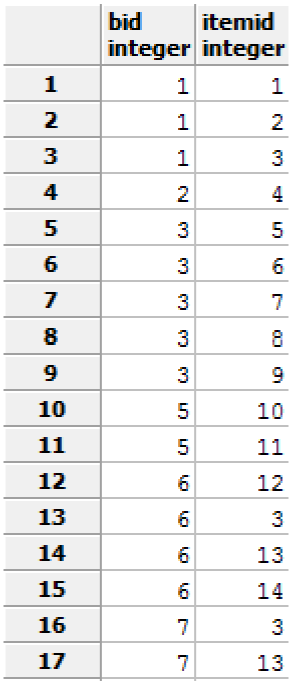
This table matches the bosses to what items they drop upon defeat.



```
CREATE TABLE "Dark Souls Bosses".bossDrops(
        bid                     INTEGER REFERENCES "Dark Souls Bosses".bosses(bid),
        itemID                  INTEGER REFERENCES "Dark Souls Bosses".items(itemID),
        PRIMARY KEY (bid,itemID)
);
```

Functional Dependency: bid,itemID→

Sample Data:

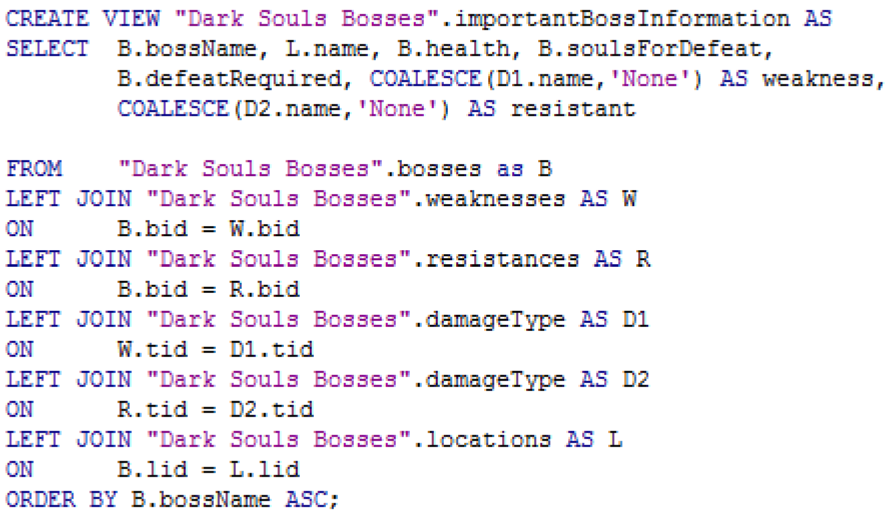


	bid integer	itemid integer
1	1	1
2	1	2
3	1	3
4	2	4
5	3	5
6	3	6
7	3	7
8	3	8
9	3	9
10	5	10
11	5	11
12	6	12
13	6	3
14	6	13
15	6	14
16	7	3
17	7	13

VIEWS

IMPORTANT BOSS INFORMATION VIEW

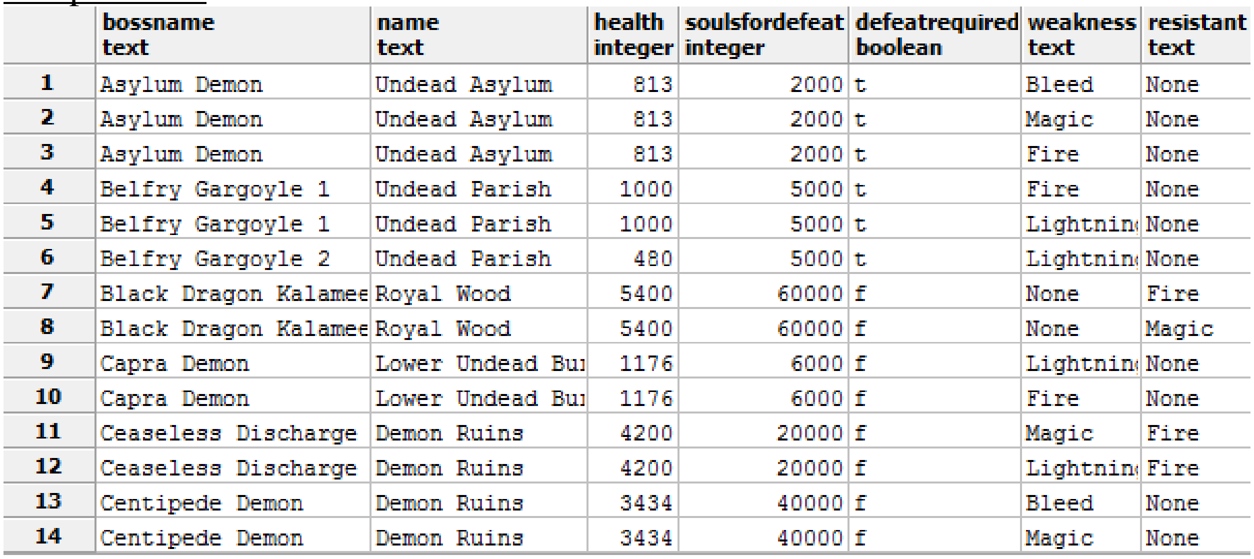
This view allows one to see all the pertinent information about every boss.



```
CREATE VIEW "Dark Souls Bosses".importantBossInformation AS
SELECT  B.bossName, L.name, B.health, B.soulsForDefeat,
        B.defeatRequired, COALESCE(D1.name,'None') AS weakness,
        COALESCE(D2.name,'None') AS resistant

FROM    "Dark Souls Bosses".bosses as B
LEFT JOIN "Dark Souls Bosses".weaknesses AS W
ON      B.bid = W.bid
LEFT JOIN "Dark Souls Bosses".resistances AS R
ON      B.bid = R.bid
LEFT JOIN "Dark Souls Bosses".damageType AS D1
ON      W.tid = D1.tid
LEFT JOIN "Dark Souls Bosses".damageType AS D2
ON      R.tid = D2.tid
LEFT JOIN "Dark Souls Bosses".locations AS L
ON      B.lid = L.lid
ORDER BY B.bossName ASC;
```

Sample Data:

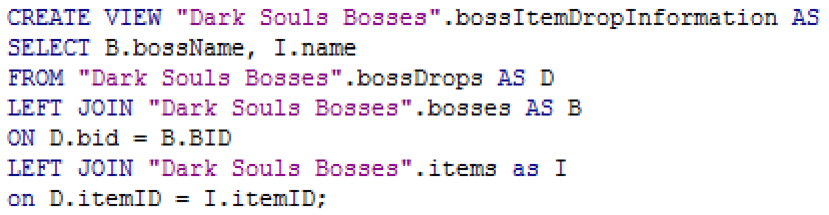


	bossname	name	health	soulsfordefeat	defeatrequired	weakness	resistant
	text	text	integer	integer	boolean	text	text
1	Asylum Demon	Undead Asylum	813	2000	t	Bleed	None
2	Asylum Demon	Undead Asylum	813	2000	t	Magic	None
3	Asylum Demon	Undead Asylum	813	2000	t	Fire	None
4	Belfry Gargoyle 1	Undead Parish	1000	5000	t	Fire	None
5	Belfry Gargoyle 1	Undead Parish	1000	5000	t	Lightnin	None
6	Belfry Gargoyle 2	Undead Parish	480	5000	t	Lightnin	None
7	Black Dragon Kalamee	Royal Wood	5400	60000	f	None	Fire
8	Black Dragon Kalamee	Royal Wood	5400	60000	f	None	Magic
9	Capra Demon	Lower Undead Bu	1176	6000	f	Lightnin	None
10	Capra Demon	Lower Undead Bu	1176	6000	f	Fire	None
11	Ceaseless Discharge	Demon Ruins	4200	20000	f	Magic	Fire
12	Ceaseless Discharge	Demon Ruins	4200	20000	f	Lightnin	Fire
13	Centipede Demon	Demon Ruins	3434	40000	f	Bleed	None
14	Centipede Demon	Demon Ruins	3434	40000	f	Magic	None

"Duplicates" are present because certain bosses have multiple weaknesses/resistances.

Boss item Drop information view

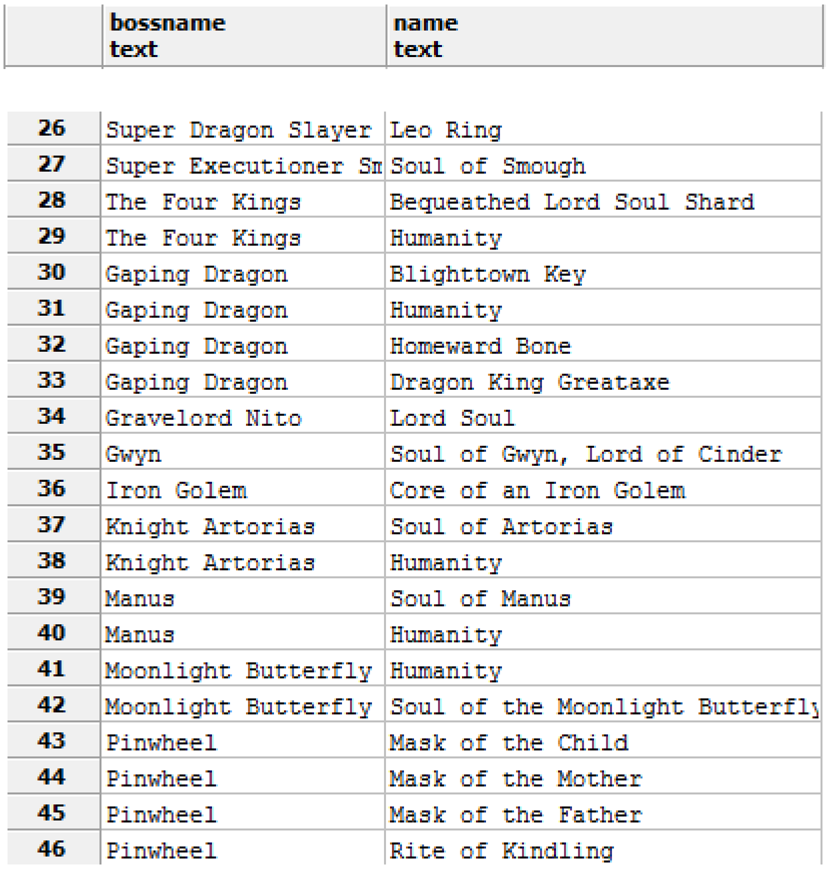
This view shows what items each boss drops. This is different from the table because this result shows the item's name, for easy readability and no need for manual cross-referencing.



```
CREATE VIEW "Dark Souls Bosses".bossItemDropInformation AS
SELECT B.bossName, I.name
FROM "Dark Souls Bosses".bossDrops AS D
LEFT JOIN "Dark Souls Bosses".bosses AS B
ON D.bid = B.BID
LEFT JOIN "Dark Souls Bosses".items as I
on D.itemID = I.itemID;
```

Sample Data:

…



	bossname text	name text
26	Super Dragon Slayer	Leo Ring
27	Super Executioner Sm	Soul of Smough
28	The Four Kings	Bequeathed Lord Soul Shard
29	The Four Kings	Humanity
30	Gaping Dragon	Blighttown Key
31	Gaping Dragon	Humanity
32	Gaping Dragon	Homeward Bone
33	Gaping Dragon	Dragon King Greataxe
34	Gravelord Nito	Lord Soul
35	Gwyn	Soul of Gwyn, Lord of Cinder
36	Iron Golem	Core of an Iron Golem
37	Knight Artorias	Soul of Artorias
38	Knight Artorias	Humanity
39	Manus	Soul of Manus
40	Manus	Humanity
41	Moonlight Butterfly	Humanity
42	Moonlight Butterfly	Soul of the Moonlight Butterfly
43	Pinwheel	Mask of the Child
44	Pinwheel	Mask of the Mother
45	Pinwheel	Mask of the Father
46	Pinwheel	Rite of Kindling

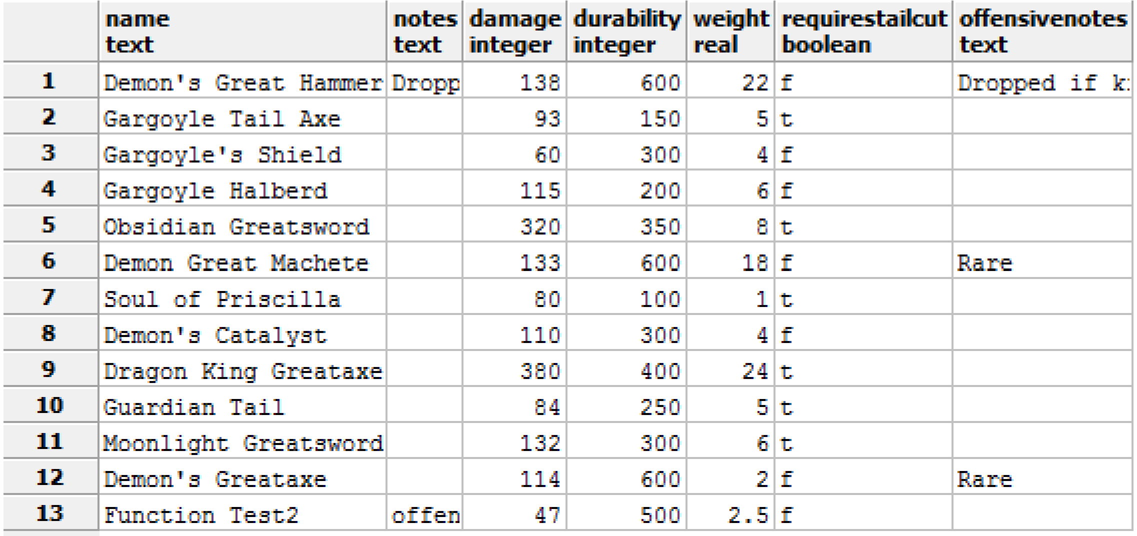
…

Offensive item information view

This view shows the offensive items but with their names' substituted in for the ID's for easy readability.

```
CREATE VIEW "Dark Souls Bosses".offensiveItemInformation AS
SELECT I.name,I.notes,O.damage,O.durability,O.weight,O.requiresTailCut,O.notes as offensiveNotes
FROM "Dark Souls Bosses".items AS I
JOIN "Dark Souls Bosses".offensiveItems AS O
ON O.itemID = I.itemID;
```

Sample Data:



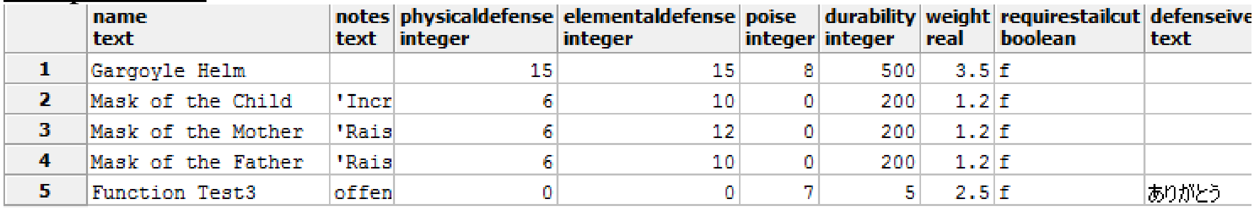
	name text	notes text	damage integer	durability integer	weight real	requirestailcut boolean	offensivenotes text
1	Demon's Great Hammer	Dropp	138	600	22	f	Dropped if k
2	Gargoyle Tail Axe		93	150	5	t	
3	Gargoyle's Shield		60	300	4	f	
4	Gargoyle Halberd		115	200	6	f	
5	Obsidian Greatsword		320	350	8	t	
6	Demon Great Machete		133	600	18	f	Rare
7	Soul of Priscilla		80	100	1	t	
8	Demon's Catalyst		110	300	4	f	
9	Dragon King Greataxe		380	400	24	t	
10	Guardian Tail		84	250	5	t	
11	Moonlight Greatsword		132	300	6	t	
12	Demon's Greataxe		114	600	2	f	Rare
13	Function Test2	offen	47	500	2.5	f	

Defensive item information view

This view shows the defensive items with their ID's replaced with their respective names.

```
CREATE VIEW "Dark Souls Bosses".defensiveItemInformation AS
SELECT I.name,I.notes,D.physicalDefense,D.elementalDefense,D.poise,D.durability,D.weight,
D.requiresTailCut,D.notes as defenseiveNotes
FROM "Dark Souls Bosses".items AS I
JOIN "Dark Souls Bosses".defensiveItems AS D
ON D.itemID = I.itemID;
```

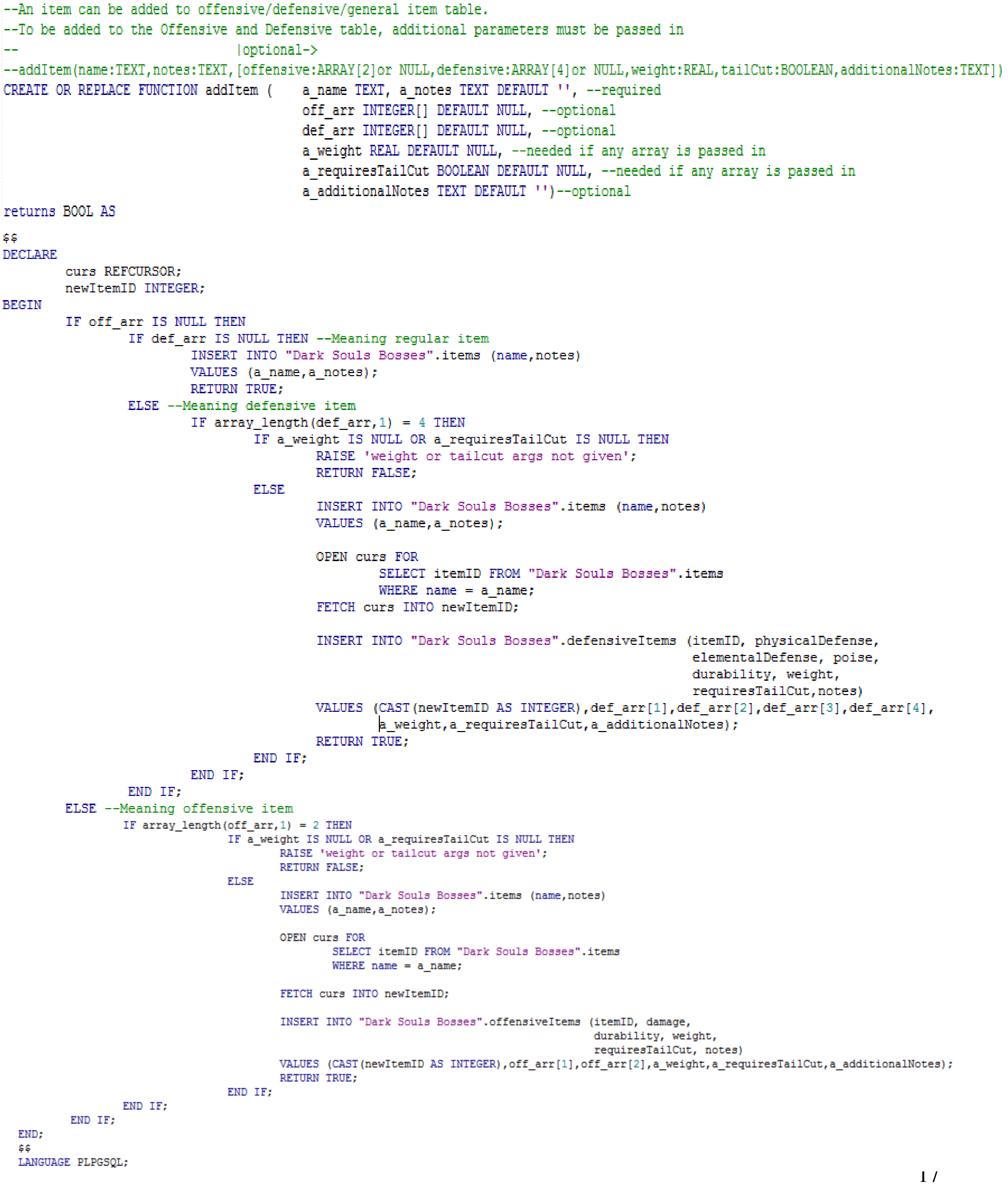
Sample Data:



	name text	notes text	physicaldefense integer	elementaldefense integer	poise integer	durability integer	weight real	requirestailcut boolean	defenseive text
1	Gargoyle Helm		15	15	8	500	3.5	f	
2	Mask of the Child	'Incr	6	10	0	200	1.2	f	
3	Mask of the Mother	'Rais	6	12	0	200	1.2	f	
4	Mask of the Father	'Rais	6	10	0	200	1.2	f	
5	Function Test3	offen	0	0	7	5	2.5	f	ありがとう

Stored Procedures

Add Item Function (explained on next page)



```
--An item can be added to offensive/defensive/general item table.
--To be added to the Offensive and Defensive table, additional parameters must be passed in
--                             |optional->
--addItem(name:TEXT,notes:TEXT,[offensive:ARRAY[2]or NULL,defensive:ARRAY[4]or NULL,weight:REAL,tailCut:BOOLEAN,additionalNotes:TEXT])
CREATE OR REPLACE FUNCTION addItem (    a_name TEXT, a_notes TEXT DEFAULT '', --required
                                        off_arr INTEGER[] DEFAULT NULL, --optional
                                        def_arr INTEGER[] DEFAULT NULL, --optional
                                        a_weight REAL DEFAULT NULL, --needed if any array is passed in
                                        a_requiresTailCut BOOLEAN DEFAULT NULL, --needed if any array is passed in
                                        a_additionalNotes TEXT DEFAULT '')--optional
returns BOOL AS
$$
DECLARE
        curs REFCURSOR;
        newItemID INTEGER;
BEGIN
        IF off_arr IS NULL THEN
                IF def_arr IS NULL THEN --Meaning regular item
                        INSERT INTO "Dark Souls Bosses".items (name,notes)
                        VALUES (a_name,a_notes);
                        RETURN TRUE;
                ELSE --Meaning defensive item
                        IF array_length(def_arr,1) = 4 THEN
                                IF a_weight IS NULL OR a_requiresTailCut IS NULL THEN
                                        RAISE 'weight or tailcut args not given';
                                        RETURN FALSE;
                                ELSE
                                        INSERT INTO "Dark Souls Bosses".items (name,notes)
                                        VALUES (a_name,a_notes);

                                        OPEN curs FOR
                                                SELECT itemID FROM "Dark Souls Bosses".items
                                                WHERE name = a_name;
                                        FETCH curs INTO newItemID;

                                        INSERT INTO "Dark Souls Bosses".defensiveItems (itemID, physicalDefense,
                                                                                        elementalDefense, poise,
                                                                                        durability, weight,
                                                                                        requiresTailCut,notes)
                                        VALUES (CAST(newItemID AS INTEGER),def_arr[1],def_arr[2],def_arr[3],def_arr[4],
                                                a_weight,a_requiresTailCut,a_additionalNotes);
                                        RETURN TRUE;
                                END IF;
                        END IF;
                END IF;
        ELSE --Meaning offensive item
                IF array_length(off_arr,1) = 2 THEN
                                IF a_weight IS NULL OR a_requiresTailCut IS NULL THEN
                                        RAISE 'weight or tailcut args not given';
                                        RETURN FALSE;
                                ELSE
                                        INSERT INTO "Dark Souls Bosses".items (name,notes)
                                        VALUES (a_name,a_notes);

                                        OPEN curs FOR
                                                SELECT itemID FROM "Dark Souls Bosses".items
                                                WHERE name = a_name;

                                        FETCH curs INTO newItemID;

                                        INSERT INTO "Dark Souls Bosses".offensiveItems (itemID, damage,
                                                                                        durability, weight,
                                                                                        requiresTailCut, notes)
                                        VALUES (CAST(newItemID AS INTEGER),off_arr[1],off_arr[2],a_weight,a_requiresTailCut,a_additionalNotes);
                                        RETURN TRUE;
                                END IF;
                END IF;
        END IF;
END;
$$
LANGUAGE PLPGSQL;
```

This function allows an admin to add an item while additionally adding data to the offensive or defensive if necessary. If one want to just add an item, then the first two parameters are the only ones that need to be supplied. If offensive or defensive data needs to be added, then the rest of the parameters need to be supplied. Here is the syntax:

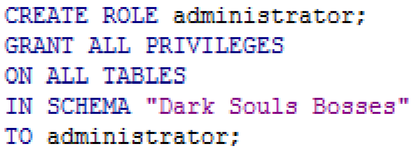
```
addItem(name:TEXT,notes:TEXT,[offensive:ARRAY(2)||NULL,
defensive:ARRAY(4)||NULL, weight:REAL, tailCut:BOOLEAN,
additionalNotes:TEXT])
```

Everything within the [ ] is optional, however, if one parameter is passed into one of the optional fields, then the rest are mandatory.

SECURITY

For this database, there would be two types of users:

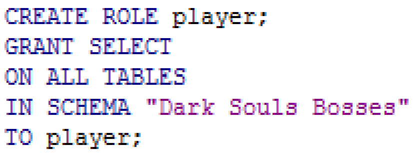
Admin:



```
CREATE ROLE administrator;
GRANT ALL PRIVILEGES
ON ALL TABLES
IN SCHEMA "Dark Souls Bosses"
TO administrator;
```

Admin will have the ability to do anything with the database, naturally as the title 'admin' would suggest.

Player (general user):



```
CREATE ROLE player;
GRANT SELECT
ON ALL TABLES
IN SCHEMA "Dark Souls Bosses"
TO player;
```

The player is just able to view the data; no internal manipulation.

Implementation Notes

Implementation went pretty well actually. Along the way, when I was gathering the data, I realized I needed to get more specific with the Items information. Originally I had one items table for every item, but then I realized that offensive and defensive items, in addition to regular consumable items, have different important information attached to them. So the fact that I divide it up into offensive and defensive, I can display more information and improve the usefulness of the database.

I also realized when I was gathering the data on weaknesses and resistances that there's more to these attributes than the normal elements I originally had, like fire and electricity. I expanded the table to just damage types so it would be able to expand and hold different types of damaging effects and not be so restrictive to only elements.

Known Problems

Uhh, it's not implemented in the real world yet. All pretentiousness aside, the only problem I see is that in the defensiveItems table, physicalDefense and elementalDefense represent the average type of damage/defense. There are four different types of physical attacks and three types of elemental attacks. So it's not very specific in that respect.

Also, the addItem procedure doesn't check if both offensive and defensive arrays are passed in. It just adds the item with the offensive stats.

Future Enhancements

I would fix the offensive and defensive tables to include the various types of physical and elemental damage to accurately reflect the item's attributes. Also I would add to the addItem procedure to check if both arrays are supplied and then print out an appropriate response of, "Please insert only one array".

I would also add more notes to the items and bosses to make the database more complete.