Dark Souls Relational Database

Designed and Documented by: Bradley Lamitie

Made for Database Management using material taught by Alan Labouseur

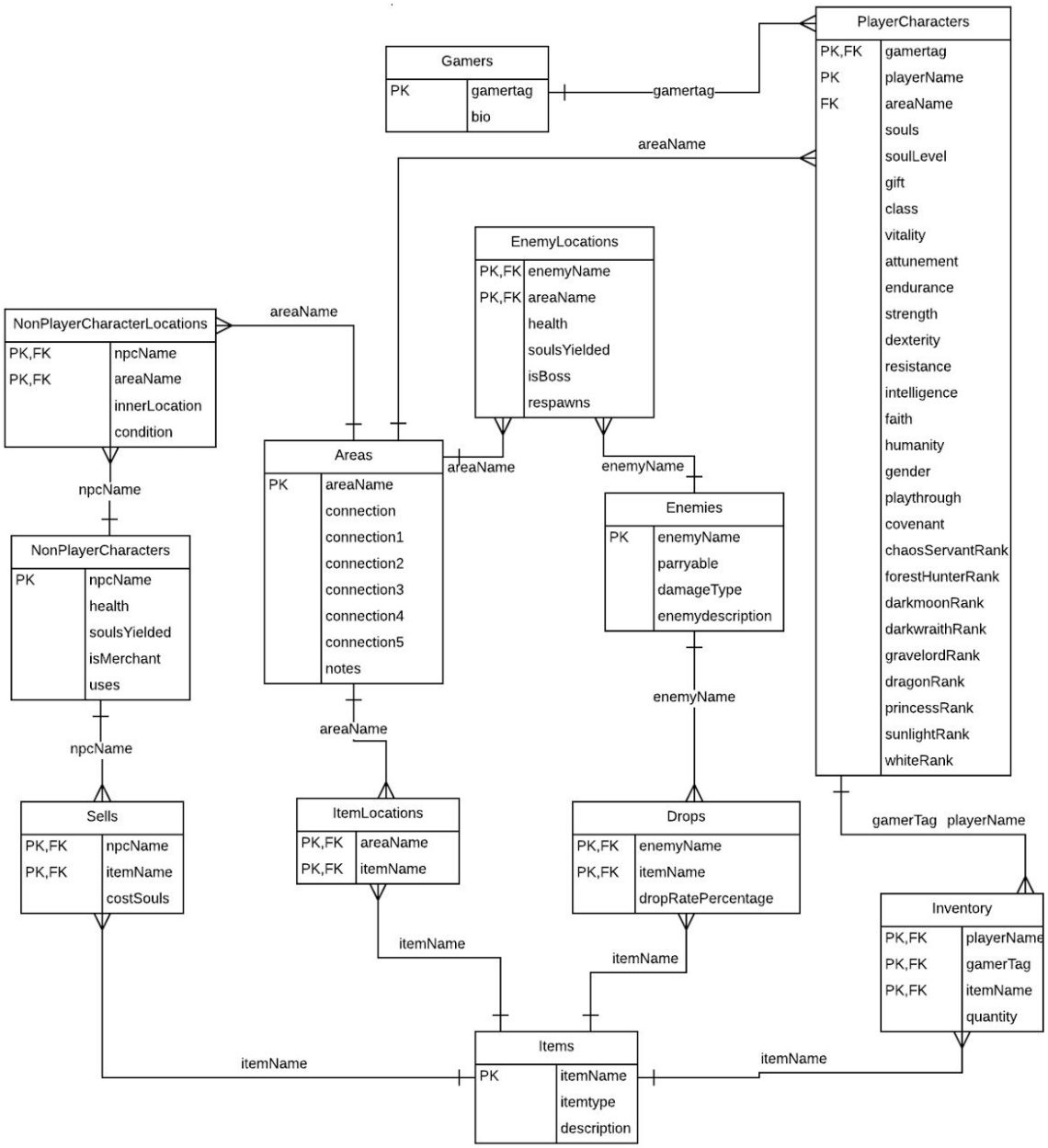
Executive Summary

This is the documentation of the Dark Souls database guide. This guide can be used to find locations of specific items, Non Player Characters, and Enemies. This guide is primarily targeting gamers, people interested in Dark Souls or other games, and game designers.

First, a walkthrough of the Entity Relationship diagram will be explained, followed by an explanation of all views and stored procedures. Next, Security clearance of the database is explained and finally a summary of all future enhancements and known problems.

Entity Relationship Diagram

Database Design Project
Dark Souls Database



Tables

Items table

This table shows an item ,it's type and description

Items	
PK	itemName
	itemtype
	description

itemName - the name of the item.
itemType - the type of item (consumable, upgrade material, armor, weapon, sorcery, pyromancy, miracle, or key).
description - a brief description of the item's uses.

Create statement:

```
CREATE TABLE Items
(
   itemName          TEXT        NOT NULL UNIQUE PRIMARY KEY,
   itemType          TEXT        NOT NULL,
   description       TEXT        NOT NULL
);
```

Functional Dependencies:

itemName --> itemType,description

Sample Data:

Binoculars	Key	Key Item used to view far away
Heal	Miracle	Heals a small amount of health
Great Heal Excerpt	Miracle	Heals a large amount of HP.
Seek Guidance	Miracle	Show More online hints.
Force	Miracle	This quickly-acting miracle inflicts no damage, but propels foes back and defends against arrows.
Homeward	Miracle	Return to last used bonfire, similar to the Homeward Bone item.
Talisman	Talisman	Used for casting miracles with MagAdjust of 118.
Thorolund Talisman	Talisman	Used for casting miracles with fixed MagAdjust of 165.
Soul Arrow	Sorcery	Shoots a fast projectile at target that does Magic damage
Great Soul Arrow	Sorcery	Stronger Soul Arrow
Heavy Soul Arrow	Sorcery	Slower Soul Arrow, but higher damage than Great
Great Heavy Soul Arrow	Sorcery	Stronger Heavy Soul Arrow
Magic Weapon	Sorcery	Must be used with catalyst in left hand, adds Magic damage to your right hand weapon
Magic Shield	Sorcery	Must be used with catalyst in right hand, temporarily increases the stability of your shield

Areas table

This table shows all connecting areas to a single area

Areas	
PK	areaName
	connection
	connection1
	connection2
	connection3
	connection4
	connection5
	notes

areaName - the name of the area
connection - the name of an adjacent area
connection1 - the name of an adjacent area
connection2 - the name of an adjacent area
connection3 - the name of an adjacent area
connection4 - the name of an adjacent area
connection5 - the name of an adjacent area
notes - any important information the user should know about the area

Create statement:

```
CREATE TABLE Areas
(
        areaName        TEXT        NOT NULL UNIQUE PRIMARY KEY,
        connection      TEXT        NOT NULL,
        connection1     TEXT        DEFAULT(' '),
        connection2     TEXT        DEFAULT(' '),
        connection3     TEXT        DEFAULT(' '),
        connection4     TEXT        DEFAULT(' '),
        connection5     TEXT        DEFAULT(' '),
        notes           TEXT
);
```

Functional Dependencies:

areaName --> connection, connection1, connection2, connection3, connection4, connection5, notes

Sample Data:

areaname text	connection text	connection1 text	connection2 text	connection3 text	connection4 text	connection5 text	notes text
Northern Undead Asylum	Firelink Shrine						Tutorial Level
Firelink Shrine	Undead Burg	The Catacombs	New Londo Ruins	Northern Undea...	Undead Parish	Firelink Altar	First Hub. Anastacia of Astora in cage below the bonfire. Petrus of Thorolund of Way of White covena
Undead Burg	Firelink Shrine	Undead Parish	Darkroot Basin	Depths			NPC Solaire of Astora located on the bridge. Bonfire and Broken altar for Warrior of Sunlight under He
Undead Parish	Undead Burg	Darkroot Garden	Firelink Shrine	Sens Fortress			Location of the first Bell of Awakening. Oswald of Carim is located at the entrance of the Bell Tower.
The Depths	Blighttown	Undead Burg					None
Blighttown	Valley of the D...	The Depths	The Great Hollow	Queelags Domain			Quelana of Izalith advanced pyromancy vendor. Shiva of the East Forest Keeper covenant vendor.
Quelaags Domain	Blighttown	Ash Lake					Location of the second Bell of Awakening. Quelaags Sister and Eingyi for Chaos Servant covenant.
The Great Hollow	Blighttown	Ash Lake					Hidden (and optional area) with no possibility to summon or getting summoned. Contains the most C
Ash Lake	The Great Holl...						Hidden (and optional area) with no possibility to summon or getting summoned. Is also a dead end,
Sens Fortress	Undead Parish	Anor Londo					None
Anor Londo	Sens Fortress	The Dukes Archi...	Painted World of...				None
Painted World of Ariamis	Anor Londo						Optional, requires Peculiar Doll in order to enter the painting in Anor Londo
Kiln of the First Flame	Firelink Altar						Final area of the game.
The Dukes Archives	Anor Londo	Crystal Cave					None

Gamers table

This table shows all gamers

Gamers	
PK	gamertag
	bio

gamertag - the user's Xbox gamertag username
bio - the players own biography

Create Statement:

```
CREATE TABLE Gamers
(
    gamertag        TEXT        NOT NULL UNIQUE PRIMARY KEY,
    bio             TEXT
);
```

Functional Dependencies:
gamertag --> bio

Sample Data:

gamertag text	bio text
Cosmic0blivion	I Like food
WhiteKitten87	Meow
Fallout878	Fallout Boy!
AcexMerk	Hi, Im Eddie
PapaChopz	MasterShake Yo

Enemies Table

This table shows all the enemies

Enemies	
PK	enemyName
	parryable
	damageType
	enemydescription

enemyName - the name of the enemy

parryable - determines if the enemy's attacks can be parried.

damageType - the type of damage the enemy deals

enemydescription - a brief description about the enemy

Create Statement:

```
CREATE TABLE Enemies
(
   enemyName         TEXT          NOT NULL PRIMARY KEY,
   parryable         BOOLEAN       NOT NULL,
   damageType        TEXT          NOT NULL,
   enemyDescription  TEXT          NOT NULL
);
```

Functional Dependencies:

enemyName --> parryable, damageType, enemydescription

Sample Data:

enemyname text	parryable boolean	damagetype text	enemydescription text
Armored Tusk	false	physical	A large, heavily armored boar.
Asylum Demon	false	physical	A lesser demon who guards the Undead Asylum, preventing the prisoners wit...
Basilisk	false	curse	These frog-like creatures jump around and spit gas that can result in a curse ...
Butcher	false	physical	These cannibal butchers carry two weapons, a massive butcher blade and a ...
Channeler	false	physical/sorcery	The Channelers are powerful servants of Seath the Scaleless.
CragSpider	false	physical/fire	A Hollow transformed into a spider creature that is able to breathe fire.
Crystal Golem	false	physical	A golem created from Seaths experimentation with the Primordial Crystal.
Dragon Slayer Ornstein	false	physical/lightning	Ornstein is the captain of the Four Knights of Gwyn, and presumably, the lea...
Executioner Smough	false	physical	Smough is the royal executioner of Anor Londo. He longs to be ranked with t...
Gwyn, Lord of Cinder	true	physical/Fire	Gwyn is one of the gods that defeated the Everlasting Dragons long ago.
Havel the Rock	true	physical	Havel, the Rock was a bishop of the Way of White.
Hollow	true	physical	Hollows are undead who have lost their sanity.
Hydra	false	physical/sorcery	A giant aquatic serpent with many heads.
Moonlight Butterfly	false	Sorcery	Large magical butterfly created by Seath the Scaleless from magic.
Slime	false	physical	Slimey creatures that often drop from the ceiling.
Titanite Demon	false	physical/lightning	The Prowling Demons, also known as the Titanite Demons, are demons born ...

NonPlayerCharacters table

This table shows all NonPlayerCharacters

NonPlayerCharacters	
PK	npcName
	health
	soulsYielded
	isMerchant
	uses

npcName - the name of the Non player Character
health - how much health an NPC has
soulsYielded - the amount of souls gained after killing the NPC
isMerchant - determines if the npc is a merchant or not
uses - what types of help the NPC offers

Create Statement:

```
CREATE TABLE NonPlayerCharacters
(
   npcName          TEXT        NOT NULL UNIQUE PRIMARY KEY,
   health           INT         NOT NULL,
   soulsYielded     INT         NOT NULL,
   isMerchant       BOOLEAN     NOT NULL,
   uses             TEXT        NOT NULL
);
```

Functional Dependencies:
npcName --> health, soulsYielded, isMerchant, uses

npcname text	health integer	soulsyielded integer	ismercha... boolean	uses text
Oscar, knight of Astora	793	1000	true	gives you keys and an estus flask.
Snuggly the Crow	0	0	false	trades items for other items
Crestfallen Warrior	793	1000	false	gives some conversation
Petrus of Thorolund	594	1000	true	sells miracles
The Crow	0	0	false	transports the player back to Northern Undead Asylum
Anastacia of Astora	0	0	false	Levels up your Estus Flask
Griggs of Vinheim	659	1000	true	sells sorcery
Laurentius	719	1000	true	sells and levels up pyromancy
Big Hat Logan	719	1000	true	sells sorceries

PlayerCharacters table

This table shows all the players

PlayerCharacters		
PK,FK	gamertag	gamertag - the gamertag linked to the playerCharacter
PK	playerName	playerName - the name of the character the user creates
FK	areaName	areaName - the name of the location the playerCharacter is located
	souls	souls - how many souls the user has accumulated
	soulLevel	soulLevel - the player's level
	gift	gift - what gift the player picked when creating the character
	class	class - whatever class the player chose when creating the character
	vitality	Vitality - how many points the user has put into vitality
	attunement	Attunement - how many points the user has put into attunement
	endurance	Endurance - how many points the user has put into endurance
	strength	Strength - how many points the user has put into strength
	dexterity	Dexterity - how many points the user has put into dexterity
	resistance	Resistance - how many points the user has put into resistance
	intelligence	Intelligence - how many points the user has put into intelligence
	faith	Faith - how many points the user has put into faith
	humanity	Humanity - how many humanity the player has consumed
	gender	Gender - the gender of the player character
	playthrough	Playthrough - what playthrough the player is on
	covenant	Covenant - the covenant the player belongs to
	chaosServantRank	chaosServantRank - the player's rank in the chaos Servant covenant
	forestHunterRank	forestHunterRank - the player's rank in the Forest Hunter covenant
	darkmoonRank	darkmoonRank - the players rank in the Darkmoon covenant
	darkwraithRank	darkwraithRank - the players rank in the Darkwraiths covenant
	gravelordRank	gravelordRank - the players rank in the Gravelord Servant covenant
	dragonRank	dragonRank - the players rank in the Path of the Dragon covenant
	princessRank	princessRank - the players rank in the Princess Guard covenant
	sunlightRank	sunlightRank - the players rank in the Sunlight Warrior covenant
	whiteRank	whiteRank - the players rank in the Way of the White covenant

Create Statement:

```
CREATE TABLE PlayerCharacters
(
   Gamertag             TEXT    NOT NULL references Gamers(gamertag),
   playerName           TEXT    NOT NULL UNIQUE,
   areaName             TEXT    NOT NULL references Areas(areaName),
   souls                INT     NOT NULL,
   soulLevel            INT     NOT NULL,
   gift                 TEXT    NOT NULL,
   class                TEXT    NOT NULL,
   vitality             INT     NOT NULL,
   attunement           INT     NOT NULL,
   endurance            INT     NOT NULL,
   strength             INT     NOT NULL,
   dexterity            INT     NOT NULL,
   resistance           INT     NOT NULL,
   intelligence         INT     NOT NULL,
   faith                INT     NOT NULL,
   humanity             INT     NOT NULL,
   gender               CHAR    NOT NULL,
   playthrough          INT     NOT NULL,
   covenant             TEXT,
   chaosServantRank     INT     NOT NULL,
   forestHunterRank     INT     NOT NULL,
   darkmoonRank         INT     NOT NULL,
   darkwraithRank       INT     NOT NULL,
   gravelordRank        INT     NOT NULL,
   dragonRank           INT     NOT NULL,
   princessRank         INT     NOT NULL,
   sunlightRank         INT     NOT NULL,
   whiteRank            INT     NOT NULL,
   PRIMARY KEY(gamertag, playerName)
);
```

Functional Dependencies:

(gamertag,playerName)--> areaName, souls, soulLevel, gift, class, vitality, attunement, endurance, strength, dexterity, resistance, intelligence, faith, humanity, gender, playthrough, covenant , chaosServantRank, forestHunterRank, darkmoonRank, darkwraithRank, gravelordRank, dragonRank, princessRank, sunlightRank, whiteRank

Sample Data:

gamertag text	playername text	areaname text	souls integer	soullevel integer	gift text	class text
Cosmic0blivion	The Wall	Ash Lake	42	25825	Binoculars	Deprived
WhiteKitten87	Allen	Undead Parish	36	254856	Divine Blessing	Cleric
Cosmic0blivion	Steve	Northern Und...	23	12345	Black Firebomb	Wanderer
PapaChopz	Destroyer	The Depths	1	0	Master Key	Pyroman...
AcexMerk	Gretchen	Sens Fortress	356	1254	Old Witchs Ri...	Sorcerer

vitality integer	attunement integer	endurance integer	strength integer	dexterity integer	resistance integer	intelligence integer	faith integer	humanity integer	gender character	playthr... integer
12	15	18	16	24	85	56	8	6	M	1
3	26	89	75	11	10	5	26	5	M	2
13	29	17	24	24	24	24	3	1	M	2
1	1	12	52	3	15	12	20	16	M	1
15	48	20	11	11	11	14	15	16	M	3

covenant text	chaosse... integer	foresthu... integer	darkmoo... integer	darkwrai... integer	gravelor... integer	dragonr... integer	princess... integer	sunlightr... integer	whiterank integer
Way of the White	0	0	0	0	0	0	0	0	0
DarkWraith	1	2	1	1	0	3	2	4	1
Path of the Drag...	1	1	1	1	1	2	1	1	0
DarkMoon	3	3	3	3	3	3	3	3	3
Princess Guard	1	1	1	1	1	1	1	1	2

Inventory table

This table shows what items each player has in their inventory

Inventory		
PK,FK	playerName	playerName - the name of the character the user creates
PK,FK	gamerTag	Gamertag - the gamertag linked to the player character
PK,FK	itemName	itemName - the name of the item
	quantity	Quantity - how much the player has of the item specified

Create Statement:

```
CREATE TABLE Inventory
(
   playerName        TEXT        NOT NULL references PlayerCharacters(playerName),
   gamertag          TEXT        NOT NULL references Gamers(gamertag),
   itemName          TEXT        NOT NULL references Items(itemName),
   Quantity          INT         NOT NULL,
   PRIMARY KEY (gamertag, itemName, playerName)
);
```

Functional Dependencies:

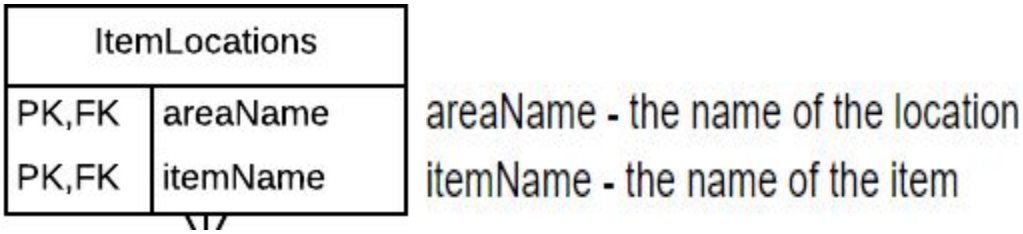
(gamertag, itemName, playerName)--> quantity

Sample Data:

playerna... text	gamertag text	itemname text	quantity integer
The Wall	CosmicOblivion	Crystal Soul Spear	1
The Wall	CosmicOblivion	Heal	10
The Wall	CosmicOblivion	Iron Flesh	1
The Wall	CosmicOblivion	Sorcerers Catalyst	1
The Wall	CosmicOblivion	Fire Orb	7
The Wall	CosmicOblivion	Great Heavy Soul Arrow	1
The Wall	CosmicOblivion	Bellowing Dragoncrest Ring	4
The Wall	CosmicOblivion	Lingering Dragoncrest Ring	1
The Wall	CosmicOblivion	Homing Crystal Soulmass	1
The Wall	CosmicOblivion	Talisman	3
The Wall	CosmicOblivion	Crystal Magic Weapon	1
Steve	CosmicOblivion	Great Heavy Soul Arrow	1
Steve	CosmicOblivion	Fire Orb	1
Steve	CosmicOblivion	Heal	1
Steve	CosmicOblivion	Bellowing Dragoncrest Ring	2
Steve	CosmicOblivion	Talisman	1
Steve	CosmicOblivion	Sorcerers Catalyst	1

ItemLocations table

This table shows the location and extra information about a specific enemy type



Create Statement:

```
CREATE TABLE ItemLocations
(
   areaName          TEXT          NOT NULL references Areas(areaName),
   itemName          TEXT          NOT NULL references Items(itemName),
   PRIMARY KEY (itemName, areaName)
);
```

No functional dependencies.

Sample Data:

areaname text	itemname text
Northern Undead Asylum	Soul of a Lost Undead
Northern Undead Asylum	Rusted Iron Ring
Firelink Shrine	Soul of a Lost Undead
Firelink Shrine	Firebomb
Firelink Shrine	Large Soul of a Lost Undead
Firelink Shrine	Homeward Bone
Firelink Shrine	Binoculars
Undead Burg	Throwing Knife
Undead Burg	Soul of a Nameless Soldier
Undead Burg	Soul of a Lost Undead
Undead Burg	Gold Pine Resin
Undead Burg	Black Firebomb

Drops table

This table shows the enemies and the items they drop.

Drops	
PK,FK	enemyName
PK,FK	itemName
	dropRatePercentage

enemyName - the name of the enemy
itemName - the item that can drop from the enemy
dropRatePercentage - the probability the item will drop.

Create Statement:

```
CREATE TABLE Drops
(
   enemyName              TEXT          NOT NULL references Enemies(enemyName),
   itemName               TEXT          NOT NULL references Items(itemName),
   dropRatePercentage      INT          NOT NULL,
   PRIMARY KEY (itemName, enemyName)
);
```

Functional Dependencies:
(itemName,enemyName) --> dropRatePercentage

Sample Data:

enemyname text	itemname text	droprate... integer
Havel the Rock	Havels Ring	100
Titanite Demon	Demon Titanite	100
Titanite Demon	Titanite Catch Pole	20
Armored Tusk	Fang Boar Helm	25
Channeler	Channelers Trident	10
Butcher	Sack	100
Slime	Green Titanite Shard	2
Slime	Large Titanite Shard	2
Basilisk	Eye of Death	6
Crystal Golem	Blue Titanite Chunk	5
Hydra	Dusk Crown Ring	100
Hydra	Dragon Scale	100
Asylum Demon	Demons Great Hamm...	100
Asylum Demon	Big Pilgrims Key	100
Asylum Demon	Humanity	100

Sells table

This table shows the items each npc sells and for how many souls

Sells	
PK,FK	npcName
PK,FK	itemName
	costSouls

npcName - the name of the non player character selling the items
itemName - the name of the item being sold
costSouls - the amount the npc is selling the item for in souls

Create Statement:

```
CREATE TABLE Sells
(
   npcName          TEXT        NOT NULL references NonPlayerCharacters(npcName),
   itemName         TEXT        NOT NULL references Items(itemName),
   costSouls        INT         NOT NULL,
   PRIMARY KEY (npcName, itemName)
);
```

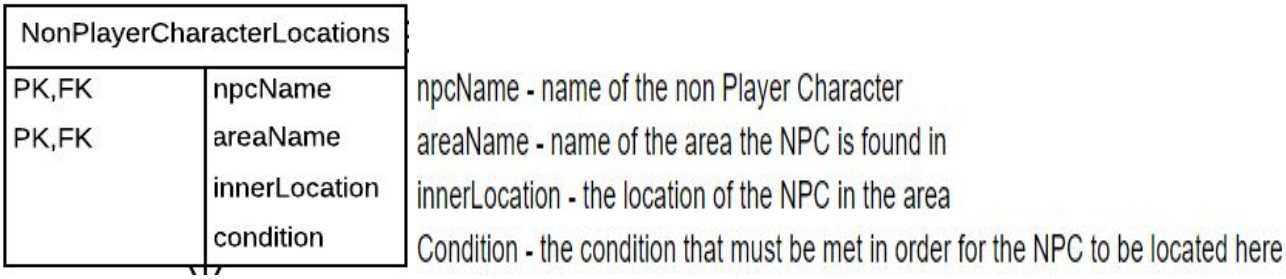
Functional Dependencies:
(npcName,itemName)--> costSouls

Sample Data:

npcname text	itemname text	costsouls integer
Petrus of Thorolund	Heal	4000
Petrus of Thorolund	Great Heal Excerpt	10000
Petrus of Thorolund	Seek Guidance	2000
Petrus of Thorolund	Force	4000
Petrus of Thorolund	Homeward	8000
Petrus of Thorolund	Talisman	1000
Petrus of Thorolund	Thorolund Talisman	5000
Griggs of Vinheim	Soul Arrow	1000
Griggs of Vinheim	Great Soul Arrow	6000
Griggs of Vinheim	Heavy Soul Arrow	2000
Griggs of Vinheim	Great Heavy Soul Arrow	8000
Griggs of Vinheim	Magic Weapon	3000
Griggs of Vinheim	Magic Shield	3000
Griggs of Vinheim	Aural Decoy	1000

NonPlayerCharacterLocations table

This table shows the location and extra information about NonPlayerCharacters



Create Statement:

```
CREATE TABLE NonPlayerCharacterLocations
(
   npcName          TEXT          NOT NULL references NonPlayerCharacters(npcName),
   areaName         TEXT          NOT NULL references Areas(areaName),
   innerLocation    TEXT          NOT NULL,
   condition        TEXT          NOT NULL,
   PRIMARY KEY (npcName, areaName, innerLocation)
);
```

Functional Dependencies:
(npcName, areaName) --> innerLocation, condition

Sample Data:

npcname text	areaname text	innerlocation text	condition text
Oscar, knight of Astora	Northern Undead Asylum	behind the wall the cannonball destroys	none
Snuggly the Crow	Northern Undead Asylum	after beating the first boss, further up the hill	none
Crestfallen Warrior	Firelink Shrine	sits against a wall near the bonfire.	none
Crestfallen Warrior	Anor Londo	attacks the player heading towards New Lond Ruins	Exhaust dialog and ring both bells
Petrus of Thorolund	Firelink Shrine	can be found near the elevator to Undead Parish	none
The Crow	Firelink Shrine	can be found after jumping off the elevator to Undead Parish	none
Anastacia of Astora	Firelink Shrine	Down the staircase on the left	none

EnemyLocations table

This table shows the location and extra information about a specific enemy type

enemyName - name of the enemy
areaName - name of the area where the enemy is located
Health - how much health the enemy has
soulsYielded - how many souls the player earns by killing the enemy
isBoss - determines if the enemy is a boss or not
Respawns - determines if the enemy respawns or not

Create Statement:

```
CREATE TABLE EnemyLocations
(
   enemyName        TEXT        NOT NULL references Enemies(enemyName),
   areaName         TEXT        NOT NULL references Areas(areaName),
   health           INT         NOT NULL,
   soulsYielded     INT         NOT NULL,
   isBoss           BOOLEAN     NOT NULL,
   respawns         BOOLEAN     NOT NULL,
   PRIMARY KEY (enemyName, areaName)
);
```

Functional Dependencies:

(enemyName, areaName) --> health, soulsYielded, isBoss, respawns

Sample Data:

enemyname text	areaname text	health integer	soulsyiel... integer	isboss boolean	respawns boolean
Hollow	Northern Undead Asylum	69	20	false	true
Hollow	Undead Parish	63	20	false	true
Hollow	Undead Burg	53	20	false	true
Hollow	The Depths	74	50	false	true
Hollow	Painted World of Ariamis	132	300	false	true
Havel the Rock	Undead Burg	1034	3000	false	false
Titanite Demon	Undead Parish	1506	2000	false	false
Titanite Demon	Sens Fortress	2510	3000	false	false
Titanite Demon	Anor Londo	2635	5000	false	false
Armored Tusk	Undead Parish	37	750	false	false

Views

EnemyLocationInfo view

Finds all Enemies and their locations

```
CREATE OR REPLACE VIEW EnemyLocationInfo
AS
SELECT DISTINCT  e.*
FROM Areas a INNER JOIN EnemyLocations e ON a.areaName= e.areaName
ORDER BY e.areaName;
Select * from EnemyLocationInfo;
```

Sample Data:

enemyname text	areaname text	health integer	soulsyiel... integer	isboss boolean	respawns boolean
Dragon Slayer Ornstein	Anor Londo	5626	50000	true	false
Executioner Smough	Anor Londo	5736	50000	true	false
Titanite Demon	Anor Londo	2635	5000	false	false
Basilisk	Ash Lake	193	400	false	true
Hydra	Ash Lake	3863	10000	false	false
CragSpider	Blighttown	167	100	false	true
Gwyn, Lord of Cinder	Kiln of the First Flame	4185	70000	true	false
Asylum Demon	Northern Undead As...	813	2000	true	false
Hollow	Northern Undead As...	69	20	false	true
Hollow	Painted World of Ari...	132	300	false	true
Titanite Demon	Sens Fortress	2510	3000	false	false

ClaimableItems view

Finds all Items lying on the ground in every location

```
CREATE OR REPLACE VIEW ClaimableItems
AS
SELECT DISTINCT a.areaName, itemName
FROM Areas a INNER JOIN ItemLocations il ON a.areaName=il.areaName
ORDER BY a.areaName;

Select * from ClaimableItems;
```

Sample Data:

areaname text	itemname text
Anor Londo	Demon Titanite
Anor Londo	Soul of a Hero
Anor Londo	Titanite Chunk
Ash Lake	Dragon Scale
Blighttown	Fire Keepers Soul
Blighttown	Green Titanite Shard
Blighttown	Humanity
Blighttown	Large Soul of a Nameless Soldier
Blighttown	Large Soul of a Proud Knight
Blighttown	Large Titanite Shard
Blighttown	Soul of a Proud Knight
Blighttown	Twin Humanities

Reports

BossLocationInfo report

Finds all Bosses and their locations

```
SELECT DISTINCT  e.areaName AS "BossLocation", enemyName AS "BossName"
FROM Areas a INNER JOIN EnemyLocations e ON a.areaName= e.areaName
WHERE isBoss = true
ORDER BY e.areaName;
```

Sample Data:

BossLocation text	BossName text
Anor Londo	Executioner Smough
Anor Londo	Dragon Slayer Ornstein
Kiln of the First Flame	Gwyn, Lord of Cinder
Northern Undead Asylum	Asylum Demon
The Dukes Archives	Moonlight Butterfly

Rare Item Info Report

Finds all Items with less than or equal to a 5% droprate, the enemies that drop them and the enemies locations.

```
SELECT DISTINCT  d.itemName AS "Item Name", el.areaName AS "Enemy Location",
e.enemyName AS "Enemy Name", dropRatePercentage AS "Drop Rate"
FROM Enemies e INNER JOIN Drops d ON d.enemyName= e.enemyName
                        INNER JOIN EnemyLocations el ON e.enemyName= el.enemyName
WHERE d.dropRatePercentage <= 5
ORDER BY el.areaName;
```

Sample Data:

Item Name text	Enemy Location text	Enemy Name text	Drop Rate integer
Green Titanite Shard	The Depths	Slime	2
Large Titanite Shard	The Depths	Slime	2
Blue Titanite Chunk	The Dukes Archives	Crystal Golem	5

Stored Procedures

AdjacentAreastoMe(TEXT,TEXT, REFCURSOR)

finds Areas adjacent to a player specified by a gamertag and playerName

```
CREATE OR REPLACE FUNCTION AdjacentAreastoMe(TEXT,TEXT, REFCURSOR)
RETURNS refcursor AS
$$
DECLARE
  playersName TEXT        := $1;
  gamerstag   TEXT        := $2;
  resultset   REFCURSOR := $3;
BEGIN
  OPEN resultset FOR
    SELECT connection,connection1,connection2,connection3,connection4,connection5
    FROM   PlayerCharacters p INNER JOIN Areas a ON p.areaName = a.areaName
     WHERE  p.playerName = playersName AND p.gamertag = gamerstag;
  RETURN resultset;
END;
$$
LANGUAGE plpgsql;

SELECT AdjacentAreastoMe('The Wall','Cosmic0blivion', 'results');
FETCH ALL FROM results;
```

Sample Data:

connection text	connection1 text	connection2 text	connection3 text	connection4 text	connection5 text
Firelink Shrine					

AdjacentAreas

finds the areas adjacent to the specified Area

```
CREATE OR REPLACE FUNCTION AdjacentAreas(TEXT, REFCURSOR) RETURNS refcursor
AS
$$
DECLARE
  specifiedArea    TEXT      := $1;
  resultset                  REFCURSOR := $2;
BEGIN
  OPEN resultset FOR
    SELECT connection,connection1,connection2,connection3,connection4,connection5
    FROM    Areas
    WHERE  areaName = specifiedArea;
  RETURN resultset;
END;
$$
LANGUAGE plpgsql;

SELECT AdjacentAreas('Firelink Shrine', 'results1');
FETCH ALL FROM results1;
```

Sample Data:

connection text	connection1 text	connection2 text	connection3 text	connection4 text	connection5 text
Undead Burg	The Catacombs	New Londo Ruins	Northern Undead Asylum	Undead Parish	Firelink Altar

AffordableItems

finds and lists all items for sale that cost less than the souls the player has

```
CREATE OR REPLACE FUNCTION AffordableItems(TEXT, TEXT, REFCURSOR) RETURNS
refcursor AS
$$
DECLARE
  specifiedGamertag          TEXT     := $1;
  specifiedPlayerName        TEXT     := $2;
  resultset                           REFCURSOR := $3;
BEGIN
  OPEN resultset FOR
    SELECT *
    FROM   Sells s INNER JOIN Items i ON i.itemName = s.itemName
    WHERE  s.costSouls<(SELECT souls
                FROM PlayerCharacters
                WHERE gamertag = specifiedGamertag AND playerName =
                  specifiedPlayerName
                );
  RETURN resultset;
END;
$$
LANGUAGE plpgsql;

SELECT AffordableItems('Cosmic0blivion', 'Steve', 'results2');
FETCH ALL FROM results2;
```

Sample Data:

npcname text	itemname text	costsouls integer	itemname text	itemtype text	description text
Petrus of Thorolund	Heal	4000	Heal	Miracle	Heals a small amount of health
Petrus of Thorolund	Great Heal Excerpt	10000	Great Heal Excerpt	Miracle	Heals a large amount of HP.
Petrus of Thorolund	Seek Guidance	2000	Seek Guidance	Miracle	Show More online hints.
Petrus of Thorolund	Force	4000	Force	Miracle	This quickly-acting miracle inflicts no damage, but propels foes back a...
Petrus of Thorolund	Homeward	8000	Homeward	Miracle	Return to last used bonfire, similar to the Homeward Bone item.
Petrus of Thorolund	Talisman	1000	Talisman	Talisman	Used for casting miracles with MagAdjust of 118.
Petrus of Thorolund	Thorolund Talisman	5000	Thorolund Talisman	Talisman	Used for casting miracles with fixed MagAdjust of 165.
Griggs of Vinheim	Soul Arrow	1000	Soul Arrow	Sorcery	Shoots a fast projectile at target that does Magic damage
Griggs of Vinheim	Great Soul Arrow	6000	Great Soul Arrow	Sorcery	Stronger Soul Arrow
Griggs of Vinheim	Heavy Soul Arrow	2000	Heavy Soul Arrow	Sorcery	Slower Soul Arrow, but higher damage than Great
Griggs of Vinheim	Great Heavy Soul Arrow	8000	Great Heavy Soul Arrow	Sorcery	Stronger Heavy Soul Arrow
Griggs of Vinheim	Magic Weapon	3000	Magic Weapon	Sorcery	Must be used with catalyst in left hand, adds Magic damage to your ri...
Griggs of Vinheim	Magic Shield	3000	Magic Shield	Sorcery	Must be used with catalyst in right hand, temporarily increases the st...
Griggs of Vinheim	Aural Decoy	1000	Aural Decoy	Sorcery	Throws a decoy that lures enemies away by creating a sound

Triggers

emp_Item()

checks to see if any of the data being entered is null and tries to alert the user

```
CREATE OR REPLACE FUNCTION emp_Item() RETURNS trigger AS $emp_Item$
   BEGIN
      -- Check that itemName and itemType are given
      IF NEW.itemName IS NULL THEN
         RAISE EXCEPTION 'itemName cannot be null';
      END IF;
      IF NEW.itemType IS NULL THEN
         RAISE EXCEPTION '% cannot have null type', NEW.itemType;
      END IF;


      RETURN NEW;
   END;
$emp_Item$ LANGUAGE plpgsql;

CREATE TRIGGER emp_Item BEFORE INSERT OR UPDATE ON Items
   FOR EACH ROW EXECUTE PROCEDURE emp_Item();

-- This is the query that i used to test to make sure the Trigger worked.
   INSERT INTO Items
   VALUES('Potato',NULL, NULL);
```

Output:

```
ERROR:  <NULL> cannot have null type
CONTEXT:  PL/pgSQL function emp_item() line 8 at RAISE
********** Error **********

ERROR: <NULL> cannot have null type
SQL state: P0001
Context: PL/pgSQL function emp_item() line 8 at RAISE
```

Security

Supervisor Privileges

Restart all roles for supervisor. and regrant them privileges:

```
REVOKE ALL ON ALL TABLES IN SCHEMA public FROM supervisor;
DROP ROLE supervisor;
CREATE ROLE supervisor;
GRANT SELECT, INSERT, UPDATE, DELETE
ON ALL TABLES IN SCHEMA public
TO supervisor;
```

Public Privileges

Grants Read only privileges to public

```
GRANT SELECT ON ALL TABLES IN SCHEMA public TO PUBLIC;
```

Implementation Notes

The implementation could've went better. My original E/R diagram was a bit of a tangled mess. So this implementation is much easier, but it comes with a steep cost. A lot of detail was lost when I had to change my E/R diagram which set me behind a little bit, considering all my stored procedures were counting on the data that i cut out.

Known Problems

As mentioned above, the lack of depth with the data is a big problem, mostly because there's a lot of things that got left out.

Future Enhancements

Some future enhancements I have been considering is adding specific data and enter all the rest of it. Another enhancement I've been thinking of implementing is a web page linked to the database that would allow users to use it as a sort of walkthrough for their game.