EarthBoundb

By Philip Kirwin

December 8, 2016

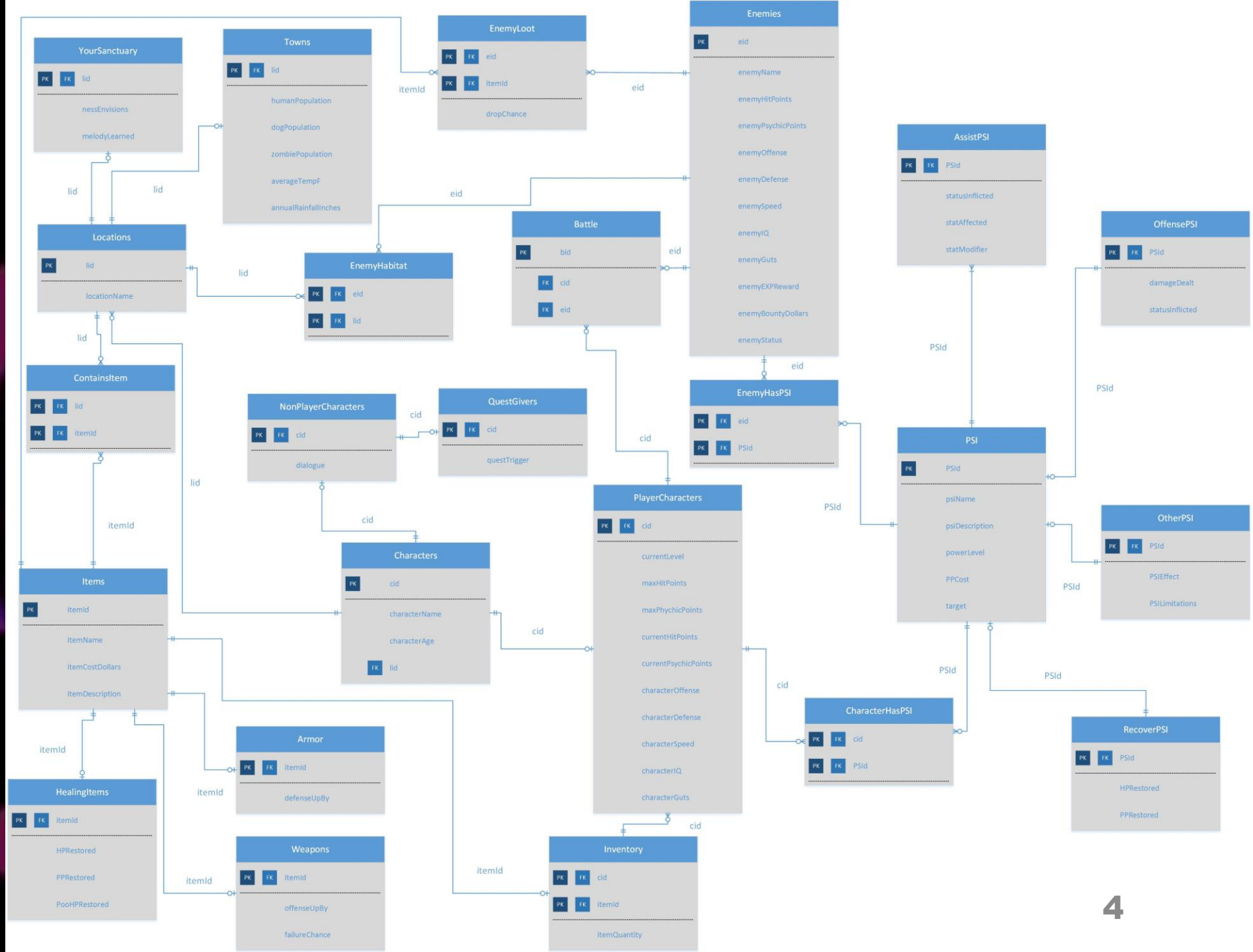
Table of Contents

Executive Summary

As is the case with most classic RPG video games, EarthBound suffers from containing a large volume of data in the form of vast arrays of characters, enemies, locations, items, and powers at the player's disposal. Naturally, new players may find the game frustrating at first, at least in part due to the sheer volume of data that is placed before them even from the game's introduction that can be difficult to remember and keep track of. EarthBoundb sets out to solve this problem as it is built to manage a majority of the core information that new EarthBound players can easily lose track of.

The following pages first demonstrate the core design of the database via an E/R Diagram. Next, each table in the database will be described in depth with sample data included. Following that is a demonstration of some of the views, stored procedures, triggers, and report queries that will be implemented along with the database for the sake of improving one's ability to access specific data. Then, the roles included with the database will be described along with the privileges held by each. Finally, implementation notes will be provided along with known issues and future enhancements as to maximize the utility of this database.

E/R Diagram


YourSanctuary
PK FK lid
nessEnvisions
melodyLearned
Towns
PK FK lid
humanPopulation
dogPopulation
zombiePopulation
averageTempF
annualRainfallInches
EnemyLoot
PK FK eid
PK FK itemId
dropChance
Enemies
PK eid
enemyName
enemyHitPoints
enemyPsychicPoints
enemyOffense
enemyDefense
enemySpeed
enemyIQ
enemyGuts
enemyEXPReward
enemyBountyDollars
enemyStatus
AssistPSI
PK FK PSId
statusInflicted
statAffected
statModifier
OffensePSI
PK FK PSId
damageDealt
statusInflicted
Locations
PK lid
locationName
EnemyHabitat
PK FK eid
PK FK lid
Battle
PK bid
FK cid
FK eid
ContainsItem
PK FK lid
PK FK itemId
NonPlayerCharacters
PK FK cid
dialogue
QuestGivers
PK FK cid
questTrigger
EnemyHasPSI
PK FK eid
PK FK PSId
PSI
PK PSId
psiName
psiDescription
powerLevel
PPCost
target
OtherPSI
PK FK PSId
PSIEffect
PSILimitations
PlayerCharacters
PK FK cid
currentLevel
maxHitPoints
maxPhychicPoints
currentHitPoints
currentPsychicPoints
characterOffense
characterDefense
characterSpeed
characterIQ
characterGuts
Characters
PK cid
characterName
characterAge
FK lid
Items
PK itemId
itemName
itemCostDollars
itemDescription
CharacterHasPSI
PK FK cid
PK FK PSId
RecoverPSI
PK FK PSId
HPRestored
PPRestored
Armor
PK FK itemId
defenseUpBy
HealingItems
PK FK itemId
HPRestored
PPRestored
PooHPRestored
Weapons
PK FK itemId
offenseUpBy
failureChance
Inventory
PK FK cid
PK FK itemId
itemQuantity
lid
itemId
eid
cid
PSId

Locations

This table is made to keep track of the various locations that exist in EarthBound.

```
CREATE TABLE Locations(
    lid INT NOT NULL UNIQUE,
    locationName TEXT NOT NULL,
    PRIMARY KEY(lid)
);
```

Dependencies: lid → locationName

Data Output | Explain | Messages | Histo

	lid integer	locationname text
1	1	Onett
2	2	Twoson
3	3	Threed
4	4	Fourside
5	5	Giant Step
6	6	Lilliput Steps
7	7	Milky Well
8	8	Winters
9	9	Dalaam
10	10	Grapefruit Falls
11	11	Peaceful Rest Valley

Your Sanctuary

This table acts as a subgroup to Locations that focuses on the Locations where Ness receives melodies.

```
CREATE TABLE YourSanctuary(
    lid INT NOT NULL UNIQUE REFERENCES Locations(lid),
    nessEnvisions TEXT NOT NULL,
    melodyLearned INT NOT NULL UNIQUE,
    PRIMARY KEY(lid)
);
```

Dependencies: lid → nessEnvisions, melodyLearned

Data Output | Explain | Messages | History

	lid integer	nessenvisions text	melodylearned integer
1	5	A small, cute puppy	1
2	6	A baby in a red cap	2
3	7	His mother, telling him to be a thoughtful, strong boy	3

Towns

This table organizes general information about the many towns one finds within the game.

```
CREATE TABLE Towns(
    lid INT NOT NULL UNIQUE REFERENCES Locations(lid),
    humanPopulation INT,
    dogPopulation INT,
    zombiePopulation INT,
    averageTempF INT,
    annualRainfallInches INT,
    PRIMARY KEY(lid)
);
```

Data Output | Explain | Messages | History

	lid integer	humanpopulation integer	dogpopulation integer	zombiepopulation integer	averagetempf integer	annualrainfallinches integer
1	1	3500	2	0	72	10
2	2	3711	1	0	72	12
3	3	1500	2	2000	62	56
4	4	313003	0	0	67	32
5	8	506	0	0	1	20
6	9	1033	0	0	70	0

Dependencies: lid → humanPopulation, dogPopulation, zombiePopulation, averageTempF, averageRainfallInches

Items

Data Output | Explain | Messages | History

	itemid integer	itemname text	itemcostdollars integer	itemdescription text
1	1	Cracked Bat	18	Ness can equip this weapon.
2	2	Fry Pan	56	Paula can equip this weapon.
3	3	Laser Gun	0	Jeff can equip this weapon.
4	4	Copper Bracelet	349	Must be equipped on your arm. It increses your Defense.
5	5	Cookie	7	When eaten, you revover about 6 HP.
6	6	Hamburger	14	When eaten, you recover about 50 HP. 100% beef.
7	7	Hard Hat	298	Should be worn when searching for buried treasure, or while you are at the construction site.
8	8	Magic Tart	480	Replenishes 20 PP.
9	9	Picnic Lunch	24	When eaten, you recover about 80 HP. There is even a slice of your favorite cake!

This table helps to organize items that players may find or purchase.

```
CREATE TABLE Items(
    itemId INT NOT NULL UNIQUE,
    itemName TEXT NOT NULL,
    itemCostDollars INT,
    itemDescription TEXT NOT NULL,
    PRIMARY KEY(itemId)
);
```

Dependencies: itemId → itemName, itemCostDollars, itemDescription

Healing Items

This table focuses on items whose purpose is to replenish health or Psychic Points.

```
CREATE TABLE HealingItems(
    itemId INT NOT NULL UNIQUE REFERENCES Items(itemId),
    HPRestored INT,
    PPRestored INT,
    PooHPRestored INT,
    PRIMARY KEY(itemId)
);
```

Dependencies: itemId → HPRestored, PPRestored, PooHPRestored

Data Output | Explain | Messages | History

	itemid integer	hprestored integer	pprestored integer	poohprestored integer
1	5	6	0	6
2	6	48	0	6
3	7	0	20	0
4	9	84	0	6

Weapons

This table includes items that help improve a character's offensive abilities.

```
CREATE TABLE Weapons(
    itemId INT NOT NULL UNIQUE REFERENCES Items(itemId),
    offenseUpBy INT NOT NULL,
    errorRate DECIMAL (8, 7),
    PRIMARY KEY(itemId)
);
```

Dependencies: itemId → offenseUpBy, errorRate

Data Output | Explain | Messages | Histor

	itemid integer	offenseupby integer	errorrate numeric(8,7)
1	1	4	0.0625000
2	2	10	0.0625000
3	3	48	0.0000000

Armor

This table focuses on items that allow a character to suffer less damage when attacked.

```
CREATE TABLE Armor(
    itemId INT NOT NULL UNIQUE REFERENCES Items(itemId),
    defenseUpBy INT NOT NULL,
    PRIMARY KEY(itemId)
);
```

Dependencies: itemId → defenseUpBy

Data Output | Explain | Messa

	itemid integer	defenseupby integer
1	4	10
2	7	15

Contains Item

This table shows where items can be found by acting as a link between Items and Locations.

```
CREATE TABLE ContainsItem(
    lid INT NOT NULL REFERENCES Locations(lid),
    itemId INT NOT NULL REFERENCES Items(itemId),
    PRIMARY KEY(lid,itemId)
);
```

Dependencies: lid, itemId →

Data Output | Explain

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

Characters

This table provides data about the multitude of characters one may interact with.

```
CREATE TABLE Characters(
    cid INT NOT NULL UNIQUE,
    characterName TEXT NOT NULL,
    characterAge INT,
    lid INT NOT NULL REFERENCES Locations(lid),
    PRIMARY KEY(cid)
);
```

Dependencies: cid → characterName, characterAge, lid

Data Output | Explain | Messages | History

	cid integer	charactername text	characterage integer	lid integer
1	1	Ness	13	1
2	2	Paula Polestar	13	2
3	3	Jeff Andonuts	13	8
4	4	Poo	14	9
5	5	Pokey Minch	13	1
6	6	Apple Kid	13	2
7	7	Geldegarde Monotoli	50	4
8	8	The Camera Man	60	1

Non-Player Characters

This table focuses on characters that the player can interact with, but not directly control.

```
CREATE TABLE NonPlayerCharacters(
    cid INT NOT NULL UNIQUE REFERENCES Characters(cid),
    dialogue TEXT NOT NULL,
    PRIMARY KEY(cid)
);
```

Dependencies: cid → dialogue

Data Output | Explain | Messages | History

	cid integer	dialogue text
1	5	Spankety, spankety, spankety!
2	6	Yeah, everything is...*KABOOOOM!* Uh, Ivegotsomeproblemsheregottago, bye!
3	7	Look at my skinny arms, thin body, and gray hair... Ive become so weak since I lost the Mani Mani Statue.
4	8	Look at the camera... Ready... Say, "fuzzy pickles."

Quest Givers

This table focuses on NPCs who are capable of giving the player a quest or completing one.

```
CREATE TABLE QuestGivers(
    cid INT NOT NULL UNIQUE REFERENCES NonPlayerCharacters(cid),
    questTrigger TEXT NOT NULL,
    PRIMARY KEY(cid)
);
```

Dependencies: cid → questTrigger

Data Output | Explain | Messages | History

	cid integer	questtrigger text
1	6	Pay $100
2	7	Reach the top of the Monotoli Building

Player Characters

This table focuses on characters that the player is in direct control of.

```
CREATE TABLE PlayerCharacters(
    cid INT NOT NULL UNIQUE REFERENCES Characters(cid),
    currentLevel INT NOT NULL,
    maxHitPoints INT NOT NULL,
    maxPsychicPoints INT NOT NULL,
    currentHitPoints INT NOT NULL,
    currentPsychicPoints INT NOT NULL,
    characterOffense INT NOT NULL,
    characterDefense INT NOT NULL,
    characterSpeed INT NOT NULL,
    characterIQ INT NOT NULL,
    characterGuts INT NOT NULL,
    PRIMARY KEY(cid)
);
```

Data Output | Explain | Messages | History

	cid integer	currentlevel integer	maxhitpoints integer	maxpsychicpoints integer	currenthitpoints integer	currentpsychicpoints integer	characteroffense integer	characterdefense integer	characterspeed integer	characteriq integer	characterguts integer
1	1	35	277	95	277	95	100	65	20	19	25
2	3	31	156	0	156	0	80	66	22	29	17
3	4	30	175	112	175	112	94	48	27	23	19
4	2	32	137	125	137	125	70	74	32	25	17

Dependencies: cid→currentLevel, maxHitPoints, maxPsychicPoints, currentHitPoints, currentPsychicPoints, characterOffense, characterDefense, characterSpeed, characterIQ, characterGuts

Inventory

This table provides a link between Items and Player Characters, allowing them to interact.

```
CREATE TABLE Inventory(
    cid INT NOT NULL REFERENCES PlayerCharacters(cid),
    itemId INT NOT NULL REFERENCES Items(itemId),
    itemQuantity INT NOT NULL,
    PRIMARY KEY(cid, itemId)
);
```

Dependencies: cid, itemId → itemQuantity

Data Output | Explain | Messages

	cid integer	itemid integer	itemquantity integer
1	1	1	1
2	2	2	1
3	3	3	1
4	1	4	1
5	4	8	3
6	3	6	4
7	2	5	2
8	2	7	1
9	1	7	1

Enemies

This table organizes the adversaries that the player will face throughout the game.

```
CREATE TABLE Enemies(
    eid INT NOT NULL UNIQUE,
    enemyName TEXT NOT NULL,
    enemyHitPoints INT NOT NULL,
    enemyPsychicPoints INT NOT NULL,
    enemyOffense INT NOT NULL,
    enemyDefense INT NOT NULL,
    enemySpeed INT NOT NULL,
    enemyIQ INT NOT NULL,
    enemyGuts INT NOT NULL,
    enemyEXPReward INT NOT NULL,
    enemyBountyDollars INT NOT NULL,
    enemyStatus TEXT NOT NULL,
    PRIMARY KEY(eid)
);
```

Data Output | Explain | Messages | History

	eid integer	enemyname text	enemyhitpoints integer	enemypsychicpoints integer	enemyoffense integer	enemydefense integer	enemyspeed integer	enemyiq integer	enemyguts integer	enemyexpreward integer	enemybountydollars integer	enemystatus text
1	2	New Age Retro Hippie	87	0	19	14	5	0	10	160	23	Common
2	3	Urban Zombie	171	0	31	24	10	0	15	700	58	Common
3	4	Titanic Ant	235	102	19	23	6	20	9	685	150	Boss
4	6	Ranboob	232	42	41	63	20	8	1	2486	158	Common
5	1	Spiteful Crow	0	0	5	3	77	0	0	3	5	Common
6	5	Master Belch	650	0	50	88	16	0	20	12509	664	Boss

Dependencies: eid → enemyName, enemyHitPoints, enemyPsychicPoints, enemyOffense, enemyDefense, enemySpeed, enemyIQ, enemyGuts, enemyEXPReward, enemyBountyDollars, enemyStatus

Enemy Habitat

This table links Enemies to Locations by indicating where enemies may be found.

```
CREATE TABLE EnemyHabitat(
    eid INT NOT NULL REFERENCES Enemies(eid),
    lid INT NOT NULL REFERENCES Locations(lid),
    PRIMARY KEY(eid, lid)
);
```

Dependencies: eid, lid →

Data Output | Explain

	eid integer	lid integer
1	1	1
2	2	2
3	3	3
4	4	5
5	5	10
6	6	7

Enemy Loot

This table links Enemies to Items and thus allows enemies to drop loot when defeated.

```
CREATE TABLE EnemyLoot(
    eid INT NOT NULL REFERENCES Enemies(eid),
    itemId INT NOT NULL REFERENCES Items(itemId),
    dropChance DECIMAL(8,7) NOT NULL,
    PRIMARY KEY(eid, itemId)
);
```

Dependencies: eid, itemId → dropChance

Data Output | Explain | Messages

	eid integer	itemid integer	dropchance numeric(8,7)
1	1	5	1.0000000
2	3	6	0.0312500
3	6	9	0.0156250

Battle

This table links Enemies to Player Characters by storing data relevant to battles between them.

```
CREATE TABLE Battle(
    bid INT NOT NULL UNIQUE,
    cid INT NOT NULL REFERENCES PlayerCharacters(cid),
    eid INT NOT NULL REFERENCES Enemies(eid),
    victory BOOLEAN,
    PRIMARY KEY(bid)
);
```

Dependencies: bid → cid, eid, victory

Data Output | Explain | Messages | Hist

	bid integer	cid integer	eid integer	victory boolean
1	1	1	4	
2	2	2	1	
3	3	1	5	
4	4	2	5	
5	5	3	5	
6	6	4	6	

PSI

This table organizes the many psychic abilities seen throughout the game.

```
CREATE TABLE PSI(
    PSId INT NOT NULL UNIQUE,
    psiName TEXT NOT NULL,
    psiDescription TEXT NOT NULL,
    powerLevel TEXT NOT NULL,
    PPCost INT NOT NULL,
    target TEXT NOT NULL,
    PRIMARY KEY(PSId)
);
```

Data Output | Explain | Messages | History

	psid integer	psiname text	psidescription text	powerlevel text	ppcost integer	target text
1	1	Lifeup	Restores 300 HP to one person.	beta	8	one ally
2	2	PK Fire	Fire bursts from the fingers and a row of enemies take about 240 points of damage each.	gamma	20	one row of foes
3	3	Brainshock	Makes one enemy feels strange.	alpha	10	one enemy
4	4	Teleport	Allows you to immediately return to a place where you have already been. You need a good running approach for this to work.	alpha	2	all allies
5	5	PK Rockin	A deadly PSI attack which only Ness can use. It is a psychokinetic wave generated by concentration that deals each enemy about 640	omega	98	all enemies
6	6	PSI Magnet	Grabs 2-8 points of PP from one enemy and adds it to your own.	alpha	0	one enemy
7	7	Offense Up	Increase one persons Offense for the duration of the current battle.	alpha	10	one ally
8	8	Healing	In addition to the effects of Healing α, this cures poisonings, nausea, feeling strange and uncontrollable crying.	beta	8	one ally
9	9	PK Starstorm	The method of "shaking off the stars" which Poo learned in his training. It deals about 720 points of damage to each enemy.	omega	42	all enemies
10	10	Lifeup	Restores 400 HP to everyone.	omega	24	all allies
11	11	Shield	Protect one person with the shield of light. It reduces the damage caused by enemy attacks by 50%.	alpha	6	one ally
12	12	PK Freeze	Causes a very cold wind to swirl around one enemy, inflicting about 360 points of damage. May freeze the enemy completely.	beta	9	one enemy

Dependencies: PSId → psiName, psiDescription, powerLevel, PPCost, target

Offense PSI

This table focuses on PSI whose primary purpose is for combat.

```
CREATE TABLE OffensePSI(
    PSId INT NOT NULL UNIQUE REFERENCES PSI(PSId),
    damageDealt INT NOT NULL,
    statusInflicted TEXT,
    PRIMARY KEY(PSId)
);
```

Dependencies: PSId → damageDealt, statusInflicted

Output pane

Data Output | Explain | Messages | History

	psid integer	damagedealt integer	statusinflicted text
1	2	240	none
2	5	640	none
3	9	720	none
4	12	360	frozen

Recover PSI

This table focuses on PSI whose primary purpose is to heal allies.

```
CREATE TABLE RecoverPSI(
    PSId INT NOT NULL UNIQUE REFERENCES PSI(PSId),
    HPRestored INT,
    PPRestored INT,
    PRIMARY KEY(PSId)
);
```

Dependencies: PSId → HPRestored, PPRestored

Data Output | Explain | Messages | His

	psid integer	hprestored integer	pprestored integer
1	1	300	0
2	6	0	5
3	10	400	0

Assist PSI

This table focuses on PSI whose primary purpose is to support allies in battle.

```
CREATE TABLE AssistPSI(
    PSId INT NOT NULL UNIQUE REFERENCES PSI(PSId),
    statusInflicted TEXT,
    statAffected TEXT,
    statModifier INT,
    PRIMARY KEY(PSId)
);
```

Dependencies: PSId → statusInflicted, statAffected, statModifier

Data Output | Explain | Messages | History

	psid integer	statusinflicted text	stataffected text	statmodifier integer
1	3	strangeness	none	0
2	7	none	offense	15
3	11	shield	none	0

Other PSI

This table focuses on PSI whose primary purpose is to be fulfilled outside of combat.

```
CREATE TABLE OtherPSI(
    PSId INT NOT NULL UNIQUE REFERENCES PSI(PSId),
    PSIEffect TEXT,
    PSILimitations TEXT,
    PRIMARY KEY(PSId)
);
```

Dependencies: → PSId, PSIEffect, PSILimitations

Data Output | Explain | Messages | History

	psid integer	psieffect text	psilimitations text
1	4	teleport to previously visited location	must run in a straight line without any collisions for a somewhat long distance or it fails

Character Has PSI

This table provides a link between Player Characters and PSI based on what a character may use.

```
CREATE TABLE CharacterHasPSI(
    cid INT NOT NULL REFERENCES PlayerCharacters(cid),
    PSId INT NOT NULL REFERENCES PSI(PSId),
    PRIMARY KEY(cid, PSId)
);
```

Dependencies: cid, PSId →

Data Output | Explain

	cid integer	psid integer
1	1	1
2	4	1
3	2	2
4	4	2
5	4	3
6	1	4
7	4	4
8	1	5
9	2	6
10	4	6
11	2	7
12	1	8
13	4	8
14	4	9
15	1	10
16	1	11
17	4	11
18	2	12
19	4	12

Enemy Has PSI

This table provides a link between Enemies and PSI based on what they may use.

```
CREATE TABLE EnemyHasPSI(
    eid INT NOT NULL REFERENCES Enemies(eid),
    PSId INT NOT NULL REFERENCES PSI(PSId),
    PRIMARY KEY(eid, PSId)
);
```

Dependencies: eid, PSId →

Data Output | Explain

	eid integer	psid integer
1	4	6
2	6	11

Character and PSI

This view provides the user with a list of characters, the PSI they can learn, and other relevant data.

```
CREATE VIEW CharacterAndPSI AS
    SELECT characterName, maxPsychicPoints, psiName, powerLevel
    FROM PlayerCharacters, Characters, CharacterHasPSI, PSI
    WHERE PlayerCharacters.cid = Characters.cid
      AND PlayerCharacters.cid = CharacterHasPSI.cid
      AND PSI.PSId = CharacterHasPSI.PSId;
```

Data Output | Explain | Messages | History

	charactername text	maxpsychicpoints integer	psiname text	powerlevel text
1	Ness	95	Lifeup	beta
2	Poo	112	Lifeup	beta
3	Paula Polestar	125	PK Fire	gamma
4	Poo	112	PK Fire	gamma
5	Poo	112	Brainshock	alpha
6	Ness	95	Teleport	alpha
7	Poo	112	Teleport	alpha
8	Ness	95	PK Rockin	omega
9	Paula Polestar	125	PSI Magnet	alpha
10	Poo	112	PSI Magnet	alpha
11	Paula Polestar	125	Offense Up	alpha
12	Ness	95	Healing	beta
13	Poo	112	Healing	beta
14	Poo	112	PK Starstorm	omega
15	Ness	95	Lifeup	omega
16	Ness	95	Shield	alpha
17	Poo	112	Shield	alpha
18	Paula Polestar	125	PK Freeze	beta
19	Poo	112	PK Freeze	beta

Enemy and PSI

In a similar fashion as CharacterAndPSI, EnemyAndPSI created a list of Enemies that can learn PSI, the actual moves they can learn, and other related information.

```
CREATE VIEW EnemyAndPSI AS
    SELECT enemyName, enemyPsychicPoints, psiName, powerLevel
    FROM Enemies, EnemyHasPSI, PSI
    WHERE Enemies.eid = EnemyHasPSI.eid
      AND PSI.PSId = EnemyHasPSI.PSId;
```

Data Output | Explain | Messages | History

	enemyname text	enemypsychicpoints integer	psiname text	powerlevel text
1	Titanic Ant	102	PSI Magnet	alpha
2	Ranboob	42	Shield	alpha

Area Boss

This view returns Enemies that have the “Boss” status as well as the area that they guard and information about defeating them.

```
CREATE VIEW AreaBoss AS
    SELECT locationName, enemyName, enemyHitPoints, enemyEXPReward,
enemyBountyDollars
    FROM Locations, Enemies, EnemyHabitat
    WHERE Locations.lid = EnemyHabitat.lid
      AND Enemies.eid = EnemyHabitat.eid
      AND Enemies.enemyStatus = 'Boss';
```

Data Output | Explain | Messages | History

	locationname text	enemyname text	enemyhitpoints integer	enemyexpreward integer	enemybountydollars integer
1	Giant Step	Titanic Ant	235	685	150
2	Grapefruit Falls	Master Belch	650	12509	664

Find Person

Given a character name, this procedure returns the name of that character's original location.

```
CREATE OR REPLACE FUNCTION findPerson(TEXT)
RETURNS TABLE(characterName TEXT, locationName TEXT) AS
$$
DECLARE
    soughtPerson  TEXT := $1;
BEGIN
    RETURN QUERY
    SELECT Characters.characterName, Locations.locationName
    FROM Characters, Locations
    WHERE Locations.lid = Characters.lid
      AND Characters.characterName = soughtPerson;
END;
$$ LANGUAGE plpgsql;
```

Data Output | Explain | Messages | Hi

	charactername text	locationname text
1	Ness	Onett

Data Output | Explain | Messages | H

	charactername text	locationname text
1	Apple Kid	Twoson

Get Character PSI

Given a character name, this procedure returns the PSI (power level included) that can be learned by said character, if any.

```
CREATE OR REPLACE FUNCTION getCharacterPSI(TEXT)
RETURNS TABLE(characterName TEXT, psiName TEXT, powerLevel TEXT) AS
$$
DECLARE
    searchCharacter  TEXT := $1;
BEGIN
    RETURN QUERY
    SELECT Characters.characterName, PSI.psiName, PSI.powerLevel
    FROM Characters, PSI, CharacterHasPSI
    WHERE Characters.cid = CharacterHasPSI.cid
      AND PSI.PSId = CharacterHasPSI.PSId
      AND Characters.characterName = searchCharacter;
END;
$$ LANGUAGE plpgsql;
```

Data Output | Explain | Messages | History

	charactername text	psiname text	powerlevel text
1	Ness	Lifeup	beta
2	Ness	Teleport	alpha
3	Ness	PK Rockin	omega
4	Ness	Healing	beta
5	Ness	Lifeup	omega
6	Ness	Shield	alpha

Data Output | Explain | Messages | History

	charactername text	psiname text	powerlevel text
1	Paula Polestar	PK Fire	gamma
2	Paula Polestar	PSI Magnet	alpha
3	Paula Polestar	Offense Up	alpha
4	Paula Polestar	PK Freeze	beta

Battle Victory

This trigger sets the victory flag in the Battle table to true if an enemy in said battle has its HP fall to zero.

```
CREATE OR REPLACE FUNCTION
battleVictory() RETURNS TRIGGER AS
$$
BEGIN
    IF new.enemyHitPoints <= 0
    THEN
        UPDATE Battle
        SET victory = true
        WHERE Battle.eid =
new.eid;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER battleVictory
AFTER UPDATE on Enemies
FOR EACH ROW
EXECUTE PROCEDURE battleVictory();
```

Data Output | Explain | Messages | History

	eid integer	enemyname text	enemyhitpoints integer	enemyp integer
1	2	New Age Retro Hippie	87	
2	3	Urban Zombie	171	
3	4	Titanic Ant	235	
4	6	Ranboob	232	
5	1	Spiteful Crow	24	
6	5	Master Belch	0	

Data Output | Explain | Messages | Histo

	bid integer	cid integer	eid integer	victory boolean
1	1	1	4	
2	6	4	6	
3	2	2	1	
4	3	1	5	t
5	5	3	5	t
6	4	2	5	t

Battle Defeat

This trigger sets the victory flag in the Battle table to false if a Player Character in said battle has its HP fall to zero.

```
CREATE OR REPLACE FUNCTION battleDefeat()
RETURNS TRIGGER AS
$$
BEGIN
    IF new.currentHitPoints <= 0
    THEN
        UPDATE Battle
        SET victory = false
        WHERE Battle.cid = new.cid;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER battleDefeat
AFTER UPDATE on PlayerCharacters
FOR EACH ROW
EXECUTE PROCEDURE battleDefeat();
```

Data Output | Explain | Messages | History

	cid integer	currentlevel integer	maxhitpoints integer	maxpsychicpoints integer	currenthitpoints integer	currentpsyc integer
1	1	35	277	95	277	
2	3	31	156	0	156	
3	4	30	175	112	175	
4	2	32	137	125	0	

Data Output | Explain | Messages | Hist

	bid integer	cid integer	eid integer	victory boolean
1	1	1	4	
2	6	4	6	
3	3	1	5	
4	5	3	5	
5	2	2	1	f
6	4	2	5	f

Reports

This query shows the current inventory size of any given Playable Character.

```
SELECT characterName, SUM(itemQuantity) AS InventorySize
FROM PlayerCharacters INNER JOIN Characters
                          ON PlayerCharacters.cid = Characters.cid
                      INNER JOIN Inventory
                          ON PlayerCharacters.cid = Inventory.cid
GROUP BY characterName;
```

Data Output | Explain | Messages | His

	charactername text	inventorysize bigint
1	Jeff Andonuts	5
2	Poo	3
3	Paula Polestar	4
4	Ness	3

Reports

This query shows a list of PSI that can only be learned by one Playable Character.

```
SELECT psiName, powerLevel, COUNT(characterName) AS UserCount
FROM Characters, PlayerCharacters, CharacterHasPSI, PSI
WHERE Characters.cid = PlayerCharacters.cid
  AND PlayerCharacters.cid = CharacterHasPSI.cid
  AND CharacterHasPSI.PSId = PSI.PSId
GROUP BY psiName, PowerLevel
HAVING COUNT(characterName) = 1;
```

Output pane

Data Output | Explain | Messages | History

	psiname text	powerlevel text	usercount bigint
1	Offense Up	alpha	1
2	PK Rockin	omega	1
3	Brainshock	alpha	1
4	PK Starstorm	omega	1
5	Lifeup	omega	1

Roles

This database, in its current state, has two roles: admin and player.

Admin role: Has administrative permissions across the entirety of the database.

```
CREATE ROLE admin;
GRANT ALL ON ALL TABLES IN SCHEMA public TO admin;
```

Player role: Has permission to manipulate parts of the database, such as Inventory, Battle, or PlayerCharacters, as they play the game.

```
CREATE ROLE player;
REVOKE ALL ON ALL TABLES IN SCHEMA public FROM player;
GRANT SELECT ON ALL TABLES IN SCHEMA public TO player;
GRANT INSERT ON Inventory, Battle TO Player;
GRANT UPDATE ON Inventory, Battle, PlayerCharacters TO Player;
```

Implementation Notes

- It should be noted that the dropChance field in the enemyLoot table is a probability ranging from zero to one, rather than a percentage.
- The OtherPSI table refers exclusively to PSI abilities that have no effect inside of battle.
- It should be noted that Poo responds differently to most food items than other characters, and thus requires his own HPRestored field in the HealingItem table.
- Accuracy of a weapon is an attribute of the weapon itself (errorRate) in EarthBound, rather than being a character attribute.

Known Issues

- There is currently no method by which to attribute a status affliction to a Player Character of Enemy. As such, references to status afflictions has simply left as part of the description of corresponding PSI.
- There is currently no way to calculate damage, and therefore no way for this system to simulate battles beyond listing participants.
- Calculations for the stats of Player Characters currently does not take equipment (weapons or armor) into account.

Future Enhancements

- Support can be added in the future for additional game mechanics, including but not limited to statis ailments, damage calculations, and item use and consumption.
- Support can be made for data from other games in this series to be included in this database.