Designs Project

By: Alfred Acosta

Table Of Contents

Executive Summary

Video games are becoming bigger and better as more time passes. Some of these game may have so much love and dedication put into them that the sheer amount of information overcomes the player and may throw them out of it. This holds true for Atlus' new long awaited game , Persona 5, which has been in the making for over 3 years. The purpose of this database design project is to give new players who are trying the series for the first time and veteran players so they can have some sort of guide since this game strictly involves time management a much easier time playing the game.

The pages that follow include the creation statements for the main, sub, and second entity tables along with some test data from the game itself included in it. Then it is followed by queries that is tested through the use of joins, views, stored procedures, and triggers so it can be proved useful to the many needs the player of the game might desire. Then, it shows the roles and powers of each person who has some sort of control over the database. Finally, we have the implementation notes, known problems, and future enhancements so the database can be improved to a much higher and more beneficial level.

ER Diagram
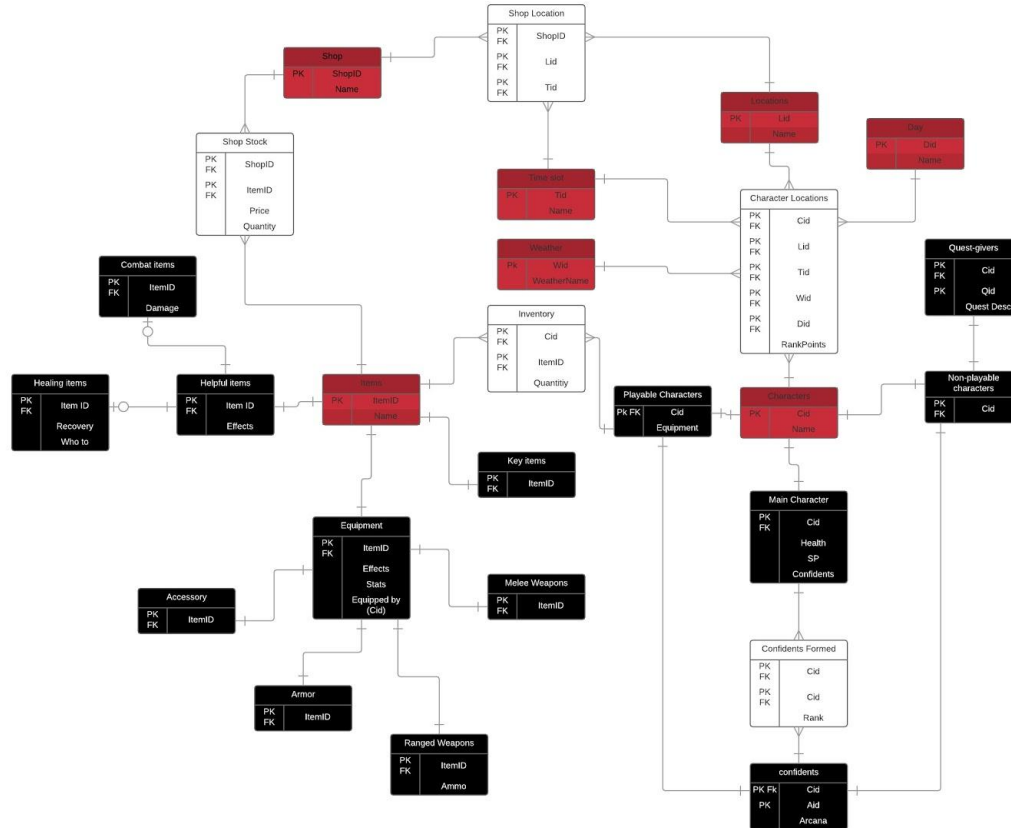
Characters

This table provides a list of all the characters in the game.

CREATE Table Characters (
   Cid varchar(4) Unique Not NULL,
   Name Text,
Primary key (Cid)
);

Dependencies: Cid -> Name

cid characte...	name text
c000	All
c001	Alan
c002	Ryuji
c003	Ann
c004	Morgana
c005	Yusuke
c006	Makoto
c007	Haru
c008	Akechi
c009	Futaba
c010	Mishama
c011	Kawakami
c012	Sojiro
c013	Sae
c014	Hifumi

Items

This table provides a list of all the items that are available to you in the game.

Create Table Items (
    ItemID varchar(4) Unique Not NULL,
    ItemName   Text,
Primary key (ItemID)
);

Functional Dependcies: ItemID -> ItemName

itemid characte...	itemname text
I001	Movie Ti...
I002	Adhesive...
I003	Protein
I004	Takemedic
I005	Takemed...
I006	Stun Gun
I007	Air Cann...
I008	Machete
I009	Spike Rod
I010	Heavy S...
I011	Mirage ...
I012	Imperial ...
I013	Iron Fist
I014	Cresent ...
I015	Quasar ...
I016	Eliminator
I017	Heavy s...

Shop

This table shows a list of all the shops you can visit in the game.

Create Table Shop(
    Sid   varchar(4) Unique Not NULL,
    ShopName   Text,
Primary key (Sid)
);

Functional Dependencie: Sid -> ShopName

sid characte...	shopname text
S001	Second-...
S002	Superma...
S003	Scarlet
S004	Bookstore
S005	Airsoft s...
S006	Big Bang...
S007	Diner
S008	Beef Bo...
S009	Conveni...
S010	Theater
S011	Flower S...
S012	Lottery S...
S013	Drug Sto...
S014	Rocinante
S015	Drink St...
S016	Station K...
S017	Accessor...
S018	Cosmeti...

Locations

This table provides a list of all the locations you can visit in the game

Create Table Locations (
    Lid   varchar(4) Unique Not NULL,
    LocationName   Text,
Primary key (Lid)
);

Functional Dependencies: Lid -> LocationName

lid characte...	locationn... text
L000	Yougen-J...
L001	Yougen-J...
L002	Yougen-J...
L003	Yougen-J...
L004	Yougen-J...
L005	Yougen-J...
L007	Shibuya:...
L008	Shibuya:...
L009	Shibuya:...
L010	Shibuya:...
L011	Shibuya:...
L012	Shibuya:...
L013	Aoyama-...
L014	Aoyama-...
L015	Aoyama-...
L016	Aoyama-...
L017	Aoyama-...
L018	Aoyama-...
L019	Aoyama-...

Time

This table provides a list of all the time slots you have access to in the game.

Create Table Time (
    Tid varchar(4) Unique Not NULL,
    TimeName Text,
Primary key (Tid)
);

Functional Dependencies: Tid-> TimeName

tid characte...	timename text
T000	Anytime
T001	Daytime
T002	Morning
T003	Lunchtime
T004	Afternoon
T005	Aftersch...
T006	Evening
T007	Summer...

Day

This table provides a list of the Days you have access to in a week.

Create Table Day (
    Did varchar(4) Unique Not NULL,
    DayName   Text,
Primary key (Did)
);

Functional DependeniesL Did -> DayName

did characte...	dayname text
D000	Anyday
D001	Monday
D002	Tuesday
D003	Wednesd...
D004	Thursday
D005	Friday
D006	Saturday
D007	Sunday

Weather

This table provides a list of all the possible weather situations a day can have.

Create Table Weather (
    Wid varchar(4) Unique Not NULL,
    WeatherName   Text,
Primary key (Wid)
);

Functional Dependcies: Wid -> WeatherName

wid characte...	weather... text
W000	Any
W001	Sunny
W002	Rain
W003	Heat Wave
W004	Torrentia...
W005	Pollen
W006	Snow

Playable Characters

This table lists all the possible allies you will have on your journey to rehabilitation.

Create Table Playable (
    Cid   varchar(4) Unique Not NULL references Characters(Cid),
    HP   integer,
    SP   integer,
Primary key (Cid)
);

Functional Dependencies: Cid -> HP, SP

cid characte...	hp integer	sp integer
c002	450	100
c003	200	250
c004	200	150
c005	450	150
c006	400	200
c007	50	500
c008	300	250
c009	300	300

MainCharacter

This table provides a detailed list of all the stats the main character can have.

Create Table MainCharacter (
    Cid   varchar(4) Unique Not NULL references Characters(Cid),
    Health int,
    SP   int,
    Money   int,
Primary key (Cid)
);

Functional Dependcies: Cid -> Health, Money
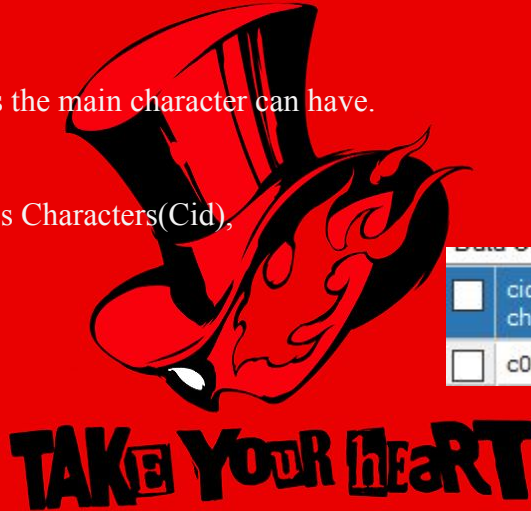
cid characte...	health integer	sp integer	money integer
c001	300	200	50000

Non Playable Characters

This table lists the minor NPCs that you may interact with on your journey.

Create Table NonPlayable (
   Cid varchar(4) Unique Not NULL references Characters(Cid),
Primary key (Cid)
);

cid characte...
c010
c011
c012
c013
c014

Quest Givers

This tables lists all the requests from others who need yout hrlp

Create Table QuestGivers (
   Cid varchar(4) Not NULL references NonPlayable(Cid),
   Qid varchar(4) Not NULL,
Primary key (Cid, Qid)
);

Functional Dependencies: Cid -> Qid

cid characte...	qid characte...
c010	q001
c010	q002
c011	q003
c014	q004

Confidents

This table lists all the bonds you may make that will help you on your journey.

Create Table Confidents (
    Cid varchar(4) Unique Not NULL references Characters(Cid),
    Aid varchar(4) Unique Not NULL,
    Arcana text,
Primary key (Cid)
);

Functional Dependencies Cid -> Aid, Aid -> Arcana

	cid characte...	aid characte...	arcana text
☐	c002	a007	Chariot
☐	c003	a006	Lovers
☐	c004	a001	Magician
☐	c005	a004	Emperor
☐	c006	a002	Priestess
☐	c007	a003	Empress
☐	c008	a008	Justice
☐	c009	a009	Hermit
☐	c010	a018	Moon
☐	c011	a014	Tempera...
☐	c012	a005	Heirophant
☐	c014	a017	Star

Key Items

This table provides a list of all the key items you can attain in the game.

Create Table KeyItems (
    ItemID varchar(4) Unique Not NULL references Items(Itemid),
Primary key (Itemid)
);

	itemid characte...
☐	I001

Equipment

This table provides a list of all the equipment you can get for you and your partners.

Create Table Equipment (
    ItemID   varchar(4) Unique Not NULL references Items(Itemid),
    Effects   text,
    Stats   integer,
    EquippedBy varchar(4) references Characters(Cid),
Primary key (Itemid, EquippedBy)
);

Functional Dependcies: ItemID -> Effects, Stats, EquippedBy

itemid characte...	effects text	stats integer	equippe... characte...
I008	Instant K...	204	c001
I009		132	c002
I010		200	c003
I011	Despair ...	204	c004
I012	Crit Cha...	204	c005
I013	Burn (s...	204	c006
I014	Reflect P...	204	c007
I015		204	c008
I016	Crit Cha...	156	c001
I017	Dizzy (s...	116	c002
I018		110	c003
I019		178	c004
I020	Burn (Lo...	178	c005
I021	Despair ...	134	c006
I022	Freeze (...	170	c007
I023	Confuse ...	234	c008
I024	Str +3	192	c001
I025	Resist Di...	180	c004
I026	SP +30	174	c003
I027		167	c005
I028	Resist Fir...	156	c006
I029	Resist C...	180	c008
I030	Reduce ...	0	c000
I031	Reduce ...	0	c000
I032	Evade N...	0	c000

Ranged Weapons

This table provides a list of all the guns you and your underaged friends can get from some sketchy guy in a back alley

Create Table RangedWeapons (
    ItemID varchar(4) Not NULL references Equipment(Itemid),
    Ammo   integer,
Primary key (Itemid)
)

Functional Dependcies: ItemID -> Ammo

itemid characte...	ammo integer
I016	16
I017	8
I018	12
I019	48
I020	32
I021	12
I022	6
I023	18

Melee Weapons

This table provides a list of all the dangerous weapons you and your friends somehow got a hold of from a certain sketchy someone.

Create Table MeleeWeapons (
    ItemID varchar(4) Not NULL references Equipment(Itemid),
Primary key (Itemid)
);

itemid characte...
I008
I009
I010
I011
I012
I013
I014
I015

¥ 7,430

Armor

This table provides a list of all the armor from the equipment table that you can wear to save you from near certain death.

Create Table Armor (
   ItemID varchar(4) Not NULL references Equipment(Itemid),
Primary key (Itemid)
);

itemid characte...
I024
I025
I026
I027
I028
I029

Accessory

This table lists all the accessories that your characters can equip to look pretty.

itemid	characte...
I030	
I031	
I032	

ItemID   varchar(4) Not NULL references Equipment(Itemid),
Primary key (Itemid)
);

Helpful Items

This table lists all the helpful items that will prove to be useful to you throughout the game.

```
Create Table HelpfulItems (
    ItemID  varchar(4) Not NULL references Items(Itemid),
    Effects text,
Primary key (Itemid)
);
```

Functional dependencies: ItemID -> Effects

itemid characte...	effects text
I002	Heal
I003	Heal
I004	Heal
I005	Heal
I006	Damage
I007	Damage

Combat Items

This table lists the items that can be used to attack or debuff enemies.

Create Table CombatItems (
    ItemID varchar(4) Not NULL references HelpfulItems(Itemid),
    Damage   integer,
Primary key (Itemid)
);

Functional dependencies: ItemID -> Damage

itemid characte...	damage integer
I006	50
I007	100

Healing Items

This table lists the items that can heal and buff your party when traversing the palaces.

	itemid characte...	recover integer	who text
☐	I002	20	one HP
☐	I003	30	one HP
☐	I004	100	one HP
☐	I005	100	all HP

Create Table HealingItems (
    ItemID varchar(4) Not NULL references HelpfulItems(Itemid),
    Recover integer,
    Who   text,
Primary key (Itemid)
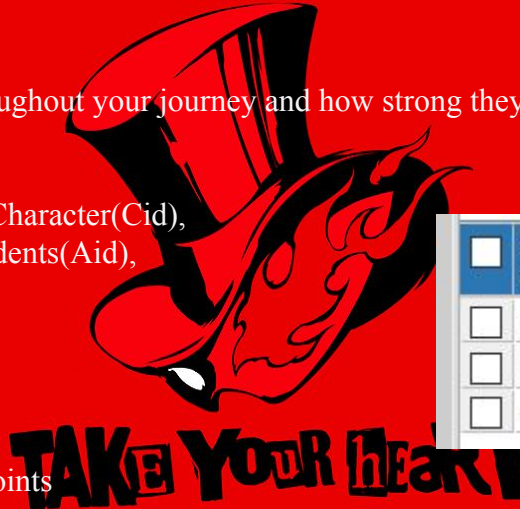);

Functional dependencies: ItemID -> Recover, Who

Bonds

This table lists the bonds you have formed throughout your journey and how strong they are.

Create Table Bonds (
    Cid varchar(4) Not NULL references MainCharacter(Cid),
    Aid varchar(4) Not NULL references Confidents(Aid),
    Rank   int,
    Points int,
Primary key (Cid, Aid)
);

Functional Dependencies: Cid, Aid -> Rank, Points

cid characte...	aid characte...	rank integer	points integer
c001	a007	3	5
c001	a006	2	3
c001	a009	4	6

Stock

This table lists how much the items in the stores cost and how much of the item they have in stock.

Create Table ShopStock (
    Sid   varchar(4) Not NULL references Shop(Sid),
    ItemID varchar(4) Not NULL references Items(ItemID),
    Price integer,
    Stock Integer CHECK (Stock <100),
Primary key (Sid, ItemID)
);

Functional Dependencies: Sid, ItemId -> Price, Stock
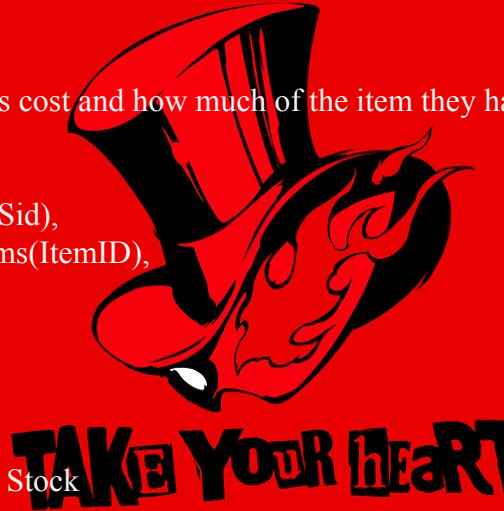
	sid characte...	itemid characte...	price integer	stock integer
☐	S001	I002	120	99
☐	S010	I001	1500	1
☐	S005	I008	30400	3
☐	S005	I018	9600	5
☐	S005	I025	32499	99
☐	S019	I030	50000	1

Shop Locations

This table lists the location of all the shops you can buy stuff from.

Create Table ShopLocation (
    Sid   varchar(4) Not NULL references Shop(Sid),
    Lid   varchar(4) Not NULL references Locations(Lid),
    Tid   varchar(4) Not NULL references Time(Tid),
Primary key (Sid, Lid, Tid)
);

Functional dependencies: Sid -> Lid, Tid

sid characte...	lid characte...	tid characte...
S001	L000	T000
S021	L010	T000
S022	L010	T005
S026	L022	T004
S028	L020	T006

Inventory

This table lists all the items you and your party are currently holding

Create Table Inventory (
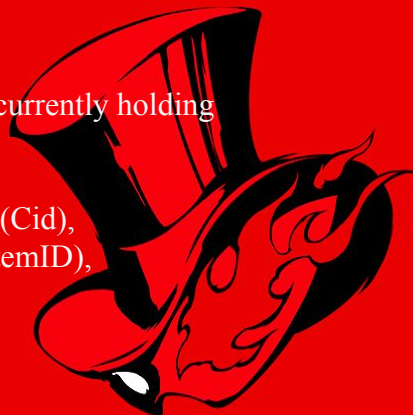   Cid varchar(4) Not NULL references Characters(Cid),
    ItemID varchar(4) Not NULL references Items(ItemID),
    Quantity Integer DEFAULT '0',
Primary key (Cid, ItemID)
);

Functional Dependencies: Cid, ItemID -> Quantity

cid characte...	itemid characte...	quantity integer
c001	I008	1
c004	I004	3
c003	I018	1

Confident Location

This table lists the location of all those you can form bonds with.

Create Table ConfidentLocation (
    Cid   varchar(4) Not NULL references Characters(Cid),
    Lid   varchar(4) Not NULL references Locations(Lid),
    Did varchar(4) Not NULL references Day(Did),
    Tid   varchar(4) Not NULL references Time(Tid),
    Wid varchar(4) Not NULL references Weather(Wid),
    RankPoints   integer,
Primary key (Cid, Lid, Did, Wid, Tid)
);

Functional Dependencies: Cid -> Lid, Did, Tid, Wid, Rank Points

cid character...	lid character...	did character...	tid character...	wid character...	rankpoints integer
c002	L011	D006	T005	W001	3
c005	L009	D005	T004	W001	1
c005	L009	D006	T005	W002	1
c006	L017	D003	T005	W002	2
c006	L020	D006	T005	W001	3
c012	L001	D000	T006	W000	3

REPORTS

This query shows how many side quests each character has that you can accept.

Select name, Count(Qid)
From Characters c,
Nonplayable n,
Questgivers q
Where c.cid = n.cid
AND   n.cid = q.cid
GROUP BY c.name;

name text	count bigint
Hifumi	1
Mishama	2
Kawakami	1

REPORTS

This query shows how much it costs to buy all the equipment for you or your teammates.

```
Select name, Sum(Price)
From Characters c,
Items i,
Equipment e,
ShopStock s
Where c.cid = e.Equippedby
AND   e.ItemID = s.ItemID
GROUP BY c.cid, c.name
ORDER BY c.cid;
```

name text	sum bigint
All	1600000
Alan	972800
Ann	307200
Morgana	1039968

Views

name text	itemname text	stats integer
Alan	Machete	204
Ryuji	Spike Rod	132
Ann	Heavy S...	200
Morgana	Mirage ...	204
Yusuke	Imperial ...	204
Makoto	Iron Fist	204
Haru	Cresent ...	204
Akechi	Quasar ...	204
Alan	Eliminator	156
Ryuji	Heavy s...	116
Ann	MP2 Prot...	110
Morgana	Northern...	178
Yusuke	Rebel Rifle	178
Makoto	Peacema...	134
Haru	Scorcher	170
Akechi	Moebius	234
Alan	Brave W...	192
Morgana	Morose ...	180

This view shows all the equippable items a character can equip and their stats.

Drop View if exists Equipped;

Create View Equipped AS
   Select c.name, i.Itemname, e.stats
       From Characters c, items i, equipment e
       Where e.EquippedBy = c.cid
       AND i.ItemID = e.ItemID;
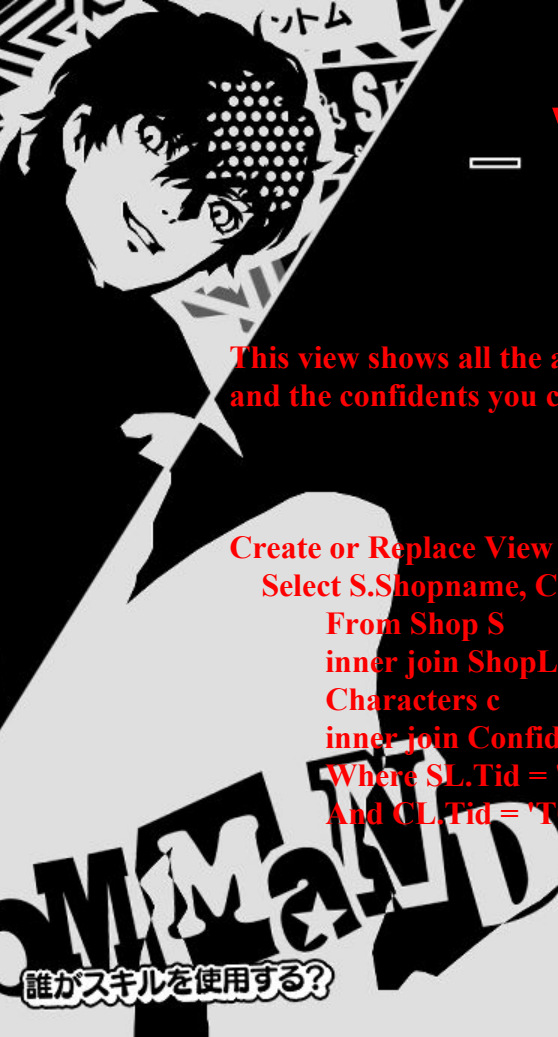
Views

This view shows all the activities, such as the shops you can visit and the confidents you can bond with, during the afternoon.

Create or Replace View Activities AS
  Select S.Shopname, C.name
    From Shop S
    inner join ShopLocation SL on SL.Sid = S.Sid,
    Characters c
    inner join ConfidentLocation CL on c.cid = CL.Cid
    Where SL.Tid = 'T004'
    And CL.Tid = 'T004';

shopname text	name text
Retro Ga...	Yusuke

Stored Procedures

This stored procedure shows all the available times a confident
Of your choice has so you can bond with them.

CREATE OR REPLACE FUNCTION FindingConfidentSpecified
 (Text, Refcursor) returns refcursor AS $$
DECLARE
    Confident TEXT := $1;
        result refcursor := $2;
BEGIN
open result for
Select C.name, L.LocationName, T.TimeName
From ConfidentLocation CL
        inner join Characters C on CL.cid = C.cid
    inner join Locations L on CL.Lid = L.Lid
        inner join Time T on CL.Tid = T.Tid
Where C.name = Confident;

return result;
END;
$$ LANGUAGE plpgsql;

Select FindingConfidentSpecified('Makoto', 'data');
fetch all from data;

name text	locationn... text	timename text
Makoto	Aoyama-...	Aftersch...
Makoto	Shinjuku...	Aftersch...

Stored Procedures

This stored procedure shows a sorted list of all the helpful Items that have the effect you desire.

```
CREATE OR REPLACE FUNCTION ItemType(Text)
RETURNS TABLE (ItemName Text)
AS $$
DECLARE
    item TEXT := $1;
BEGIN
RETURN QUERY
Select I.ItemName
From Items
I inner join HelpfulItems HI on I.Itemid = HI.Itemid
    Where HI.effects = item;
END;
$$ LANGUAGE plpgsql;

Select ItemType('Heal');
```

TRIGGERS

This Trigger makes it so you can heal all of your characters
Instead of just one

```
CREATE OR REPLACE FUNCTION HealALL(Heal) RETURNS TRIGGER AS $$
DECLARE
  Heal  int;
Begin
   If (Playable.HP < 500)
      THEN
      UPDATE PLAYABLE
      SET HP = HP + Heal;
end IF;
Return HP;
END;
$$
LANGUAGE plpgsql;
```

```
CREATE TRIGGER HealAll
     AFTER UPDATE on Playable
     FOR EACH ROW
   EXECUTE PROCEDURE HealALL(Heal);
```

name text	hp integer
Ryuji	450
Ann	200
Morgana	200
Yusuke	450
Makoto	400
Haru	50
Akechi	300
Futaba	300

37

ROLES

Rogue · 17 HP 216 SP 110

Ryuji · 17 HP 224 SP 89

Morgana · LV 16 HP 187 SP 109

Ann · 17 HP 196 SP 123

Yusuke · 17 HP 206 SP 93

This database has two roles: The first role
Belongs to the admin of the game, who
Can control all the data in the database.

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

The second role, which is the player,
Has the power to control only certain parts
of the game, like his main character, other
characters, etc.

CREATE ROLE Alan;
GRANT SELECT ON ALL TABLES IN
SCHEMA public TO Alan;

GRANT INSERT ON Inventory, Bonds,
MainCharacter, Playable to Alan;
GRANT UPDATE ON Inventory, Bonds,
MainCharacter, Playable to Alan;

Whose stats do you want to view?

□ Party ○ Back ✕ Stats

Implementation Notes

- This does not include how indepth the daily life system goes due to how much it interacts with other tables, such as the battle system.

- While all the data is real and accurate to the game, it does not have all the data due to me playing the game and not wanting to spoil the story by googling information. So the data came from me looking it up through my save file in the game.

Known Problems

- No way to separate stock from total price from the view table know as "Equipped by".

- Unknown way to equip one item to many people, such as a dress for all the girl characters except for one, besides creating more insert statements or having a separate CID for all playable characters.

- Did not include the percentages for how successful a status effect from a weapon can affect an enemy.

Future Enhancements

- **Add affinities and status ailments to the weapons so it can go in depth towards the battle system.**

- **Include a persona table, since it is part of the daily life and dungeon crawling life and is directly affected by the confidents.**

- **Finish the game so I can add more data.**