

COMIC BOOK CHARACTER GUIDE

Database Systems

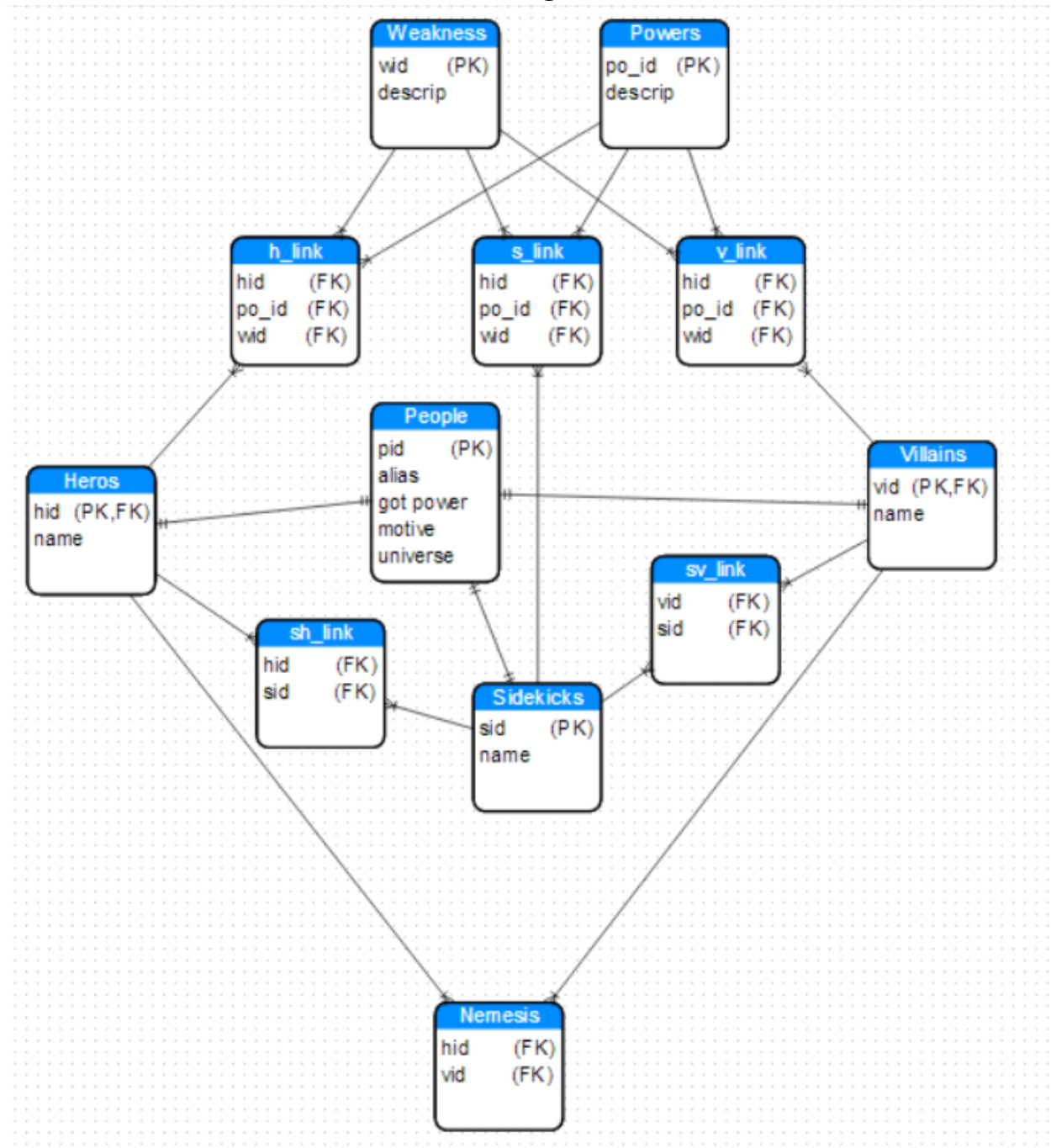
Ryan Plunkett



Table of Contents

Cover Page.....	1
Table of Contents.....	2
ER Diagram.....	3
Executive Summary.....	4
Overview.....	4
Tables.....	5-15
People.....	5
Heroes.....	6
Sidekicks.....	7
Villains.....	8
Powers.....	9
Weakness.....	10
H_link.....	11
S_link.....	12
V_link.....	13
Nemesis.....	14
Sh_link.....	15
Sv_link.....	16
Functional Dependencies.....	17
Report.....	18
View	19, 20
Stored Procedure.....	21
Trigger.....	22
Security.....	23
Known Problems.....	24
Future Enhancements.....	24

ER Diagram



Executive Summary

This document outlines the Database I created for the relationship between heroes and villains in the DC and Marvel universes. The goal of the Database is to create a simple and easy to understand list for each relationship. This would be especially helpful for people who are new to the genre. The ER Diagram gives an outline of this list, followed by a description of each entity including create statements to build tables as well as real data to fill the tables. There are also examples of reports, views, stored procedures, and triggers.

Overview

The database is broken up into two different groups. The first group contained all the people and the categories they belong to; either heroes, villains, or sidekicks as well as which heroes' villains and sidekicks work with each other and which are nemeses. The second group identifies which people have which powers and weaknesses

People

The people table contains all the information in every person in the database. These people may be extended to be either a hero, a villain or a sidekick. The table also the persons real life alias, a description of how they got their power, their motivation, and what universe they belong to.

```
CREATE TABLE people (  
    pid serial NOT NULL,  
    alias VARCHAR(50),  
    got_power VARCHAR(100),  
    motivation VARCHAR(100),  
    universe VARCHAR(20),  
    CHECK (universe = 'Marvel' OR universe = 'DC'),  
    PRIMARY KEY (pid)  
);
```

	pid integer	alias character varying(50)	got_power character varying(100)	motivation character varying(100)	universe character varying(20)
1	1	Bruce Wane	trained for years w	parents were murder	DC
2	2	Clark Kent	is an alien from an	Home planet was des	DC
3	3	Jimmy Olsen	trained for years w	parents were murder	DC
4	4	Dick Grayson	trained By Batman	parents were murder	DC
5	5	Peter Parker	bit by a radioactiv	uncle was murdered	Marvel
6	6	Bruce Banner	absorbed massive am	his anger	Marvel
7	7	James Howlett	born a mutant	self interest	Marvel
8	8	Anna Marie	born a mutant	equality for mutant	Marvel
9	9	Red Hood	fell into a vat of	to kill batman	DC
10	10	Alexander Luthor	is rich	to kill superman	DC
11	11	Norman Osborn	ingested a syrum	to kill spiderman	Marvel
12	12	Samuel Sterns	exposed to gamma ra	take over the world	Marvel
13	13	Andy Maguire	sience experiment f	help people in need	Marvel
14	14	Victor Creed	born a mutant	get what he wants	Marvel
15	15	Dr. Harleen Quinze	falls in love with	to help the joker	Marvel
16	16	Kyle Gibney	born a mutant	loves to fight	Marvel

Heroes

The heroes table contains the information about a given person that is classified as a hero by either DC or Marvel comics. The heroes table is connected to the peoples table through hid which references pid in the peoples table. The table is also connected to h_link and sh_link. As well as the nemesis table.

```
CREATE TABLE heroes(  
  name VARCHAR(50),  
  hid INT REFERENCES people (pid),  
  PRIMARY KEY (hid)  
);
```

	name character varying(50)	hid integer
1	Batman	1
2	Superman	2
3	Spiderman	5
4	The Hulk	6
5	Wolverine	7

Sidekicks

The sidekicks table contains information on every person in that peoples that is also a sidekick. The sidekicks table is connected to the peoples table through sid which references pid in the peoples table. It is also connected to heroes and villains through sh_id and sv_id respectively, as well as s_link

```
CREATE TABLE sidekicks (  
    name VARCHAR(50) ,  
    sid INT REFERENCES people (pid),  
    PRIMARY KEY (sid)  
);
```

	name character varying(50)	sid integer
1	Robin	4
2	Flamebird	3
3	Alpha	13
4	Rogue	8
5	Harley Quinn	15
6	Wildchild	16

Villains

The villains table contains information on every person in that peoples that is also a villain. The villains table is connected to the peoples table through vid which references pid in the peoples table. It is also connected to heroes and sidekicks through sh_id and sv_id respectively, as well as v_link and nemesis.

```
CREATE TABLE villains (  
  name VARCHAR(50),  
  vid INT REFERENCES people (pid),  
  PRIMARY KEY (vid)  
);
```

	name character varying(50)	vid integer
1	The Joker	9
2	Lex Luthor	10
3	Green Goblin	11
4	The Leader	12
5	Sabretooth	14

Powers

The powers table contains a list of all possible powers that any subclass of people may have called descrip. It has its own primary key, po_id, which unlike the others does not reference pid. Powers is linked to heroes, sidekicks, and villains through their respective linking tables.

```
CREATE TABLE powers (  
    po_id serial NOT NULL,  
    name VARCHAR(100),  
    PRIMARY KEY (po_id)  
);
```

	po_id integer	name character varying(100)
1	1	super strength
2	2	flight
3	3	spider abilities
4	4	healing
5	5	claws
6	6	super intellect
7	7	money
8	8	acrobatics
9	9	death touch
10	10	kenetic enegy

Weakness

Like the powers table, the weakness table contains a list of all possible powers that any subclass of people may have called descrip. It has its own primary key, wid, which also like powers does not reference pid. Weakness is linked to heroes, sidekicks, and villains through their respective linking tables.

```
CREATE TABLE weakness (  
    wid serial NOT NULL,  
    name VARCHAR(100),  
    PRIMARY KEY (wid)  
);
```

	wid integer	name character varying(100)
1	1	kryptonite
2	2	magnets
3	3	morality
4	4	responsibility
5	5	anger
6	6	physical harm
7	7	insanity
8	8	pride

H_Link

H_link is one of the associate or linking tables in my database. Meaning it takes its information from the other tables in the list and compiles them into a coherent table. H_Link gets its data from powers, weakness, and heroes. Meaning its table will show all the powers and weaknesses for each of the heroes in the heroes table.

```
CREATE TABLE h_link(  
    hid INT REFERENCES heroes (hid),  
    po_id INT REFERENCES powers (po_id),  
    wid INT REFERENCES weakness (wid)  
);
```

	hid integer	po_id integer	wid integer
1	1	8	3
2	2	1	1
3	5	3	4
4	6	1	5
5	7	4	2

S_Link

S_link is another one of the associate or linking tables in my database. It hold the same fuction as h_link does only it works with sidekicks instead if heroes.

```
CREATE TABLE s_link(  
    sid INT REFERENCES sidekicks (sid),  
    po_id INT REFERENCES powers (po_id),  
    wid INT REFERENCES weakness (wid)  
);
```

	sid integer	po_id integer	wid integer
1	4	8	6
2	3	8	6
3	13	10	6
4	8	9	6
5	15	8	6
6	16	4	7

V_Link

V_link is the second linking table that is similar to h_link. Like its predecessors it holds the same function as h_link and s_link do, only it works with villains instead of heroes or sidekicks.

```
CREATE TABLE v_link(  
    vid INT REFERENCES villains (vid),  
    po_id INT REFERENCES powers (po_id),  
    wid INT REFERENCES weakness (wid)  
);
```

	vid integer	po_id integer	wid integer
1	9	6	7
2	10	7	8
3	11	8	7
4	12	6	6
5	14	5	5

Nemesis

The Nemesis table is also a linking table only instead of linking to powers or weakness, nemesis links heroes and villains through their respective id's. This will allow the user to see which heroes are enemies with which villains.

```
CREATE TABLE nemesis(  
    hid INT REFERENCES heroes (hid),  
    vid INT REFERENCES villains (vid)  
);
```

	hid integer	vid integer
1	1	9
2	2	10
3	5	11
4	6	12
5	7	14

Sh_Link

Sh_link is an associate table which stores hero with his or her respective sidekick.

Allowing users to see the different crime fighting teams.

```
CREATE TABLE sh_link(  
    hid INT REFERENCES heroes(hid),  
    sid INT REFERENCES sidekicks(sid)  
);
```

	hid integer	sid integer
1	1	4
2	2	3
3	5	13
4	7	8

Sv Link

Sv_link is an associate table which is similar to sh_link, except it stores villains with his or her respective sidekick. Allowing users to see the different crime committing gangs.

```
CREATE TABLE sv_link(  
    vid INT REFERENCES villains (vid),  
    sid INT REFERENCES sidekicks (sid)  
);
```

	vid integer	sid integer
1	9	15
2	14	16

Functional Dependencies

People

Pid -> alias, got_power, motivation, universe,

Heroes

Hid -> name

Sidekicks

Sid -> name

Villains

Vid -> name

Powers

Po_id -> descrip

Weakness

Wid -> descrip

H_link

NONE

S_link

NONE

V_link

NONE

Nemesis

NONE

Sh_link

NONE

Sv_link

NONE

Reports

Every person's alias from the peoples table as well as their corresponding power

```
Select distinct people.alias, powers.name
From people, powers, s_link, v_link, h_link
Where people.pid = h_link.hid
      and powers.po_id = h_link.po_id
or people.pid = s_link.sid
      and powers.po_id = s_link.po_id
or people.pid = v_link.vid
      and powers.po_id = v_link.po_id
Order by alias
```

	alias character varying(50)	name character varying(100)
1	Alexander Luthor	money
2	Andy Maguire	kenetic energy
3	Anna Marie	death touch
4	Bruce Banner	super strength
5	Bruce Wane	acrobatics
6	Clark Kent	super strength
7	Dick Grayson	acrobatics
8	Dr. Harleen Quinze	acrobatics
9	James Howlett	healing
10	Jimmy Olsen	acrobatics
11	Kyle Gibney	healing
12	Norman Osborn	acrobatics
13	Peter Parker	spider abilities
14	Red Hood	super intellect
15	Samuel Sterns	super intellect
16	Victor Creed	claws

Views

All in one view

```
select distinct heroes.name, people.alias, people.universe,  
                people.got_power, people.motivation, powers.name  
from heroes, people, powers, h_link  
where heroes.hid = people.pid  
        and people.pid = h_link.hid  
        and powers.po_id = h_link.po_id
```

union all

```
select sidekicks.name, people.alias, people.universe,  
        people.got_power, people.motivation, powers.name  
from sidekicks, people, powers, s_link  
where sidekicks.sid = people.pid  
        and people.pid = s_link.sid  
        and powers.po_id = s_link.po_id
```

union all

```
select villains.name, people.alias, people.universe,  
        people.got_power, people.motivation, powers.name  
from villains, people, powers, v_link  
where villains.vid = people.pid  
        and people.pid = v_link.vid  
        and powers.po_id = v_link.po_id
```

***Output on next page**

	name character varying(50)	alias character varying(50)	universe character varying(20)	got_power character varying(100)	motivation character varying(100)	name ie character varying(100)
1	Superman	Clark Kent	DC	is an alien from an	Home planet was des	super strength
2	The Hulk	Bruce Banner	Marvel	absorbed massive am	his anger	super strength
3	Wolverine	James Howlett	Marvel	born a mutant	self interest	healing
4	Batman	Bruce Wane	DC	trained for years w	parents were murder	acrobatics
5	Spiderman	Peter Parker	Marvel	bit by a radioactive	uncle was murdered	spider abilities
6	Robin	Dick Grayson	DC	trained By Batman	parents were murder	acrobatics
7	Flamebird	Jimmy Olsen	DC	trained for years w	parents were murder	acrobatics
8	Alpha	Andy Maguire	Marvel	science experiment f	help people in need	kinetic energy
9	Rogue	Anna Marie	Marvel	born a mutant	equality for mutant	death touch
10	Harley Quinn	Dr. Harleen Quinze	Marvel	falls in love with	to help the joker	acrobatics
11	Wildchild	Kyle Gibney	Marvel	born a mutant	loves to fight	healing
12	The Joker	Red Hood	DC	fell into a vat of	to kill batman	super intellect
13	Lex Luthor	Alexander Luthor	DC	is rich	to kill superman	money
14	Green Goblin	Norman Osborn	Marvel	ingested a syrum	to kill spiderman	acrobatics
15	The Leader	Samuel Sterns	Marvel	exposed to gamma ra	take over the world	super intellect
16	Sabretooth	Victor Creed	Marvel	born a mutant	get what he wants	claws

Stored Procedures

Get power given name

This stored procedure returns the power of any hero, villain or sidekick that is entered.

```
CREATE OR REPLACE FUNCTION personPower(who TEXT)
RETURNS TABLE (power TEXT) AS
$$
BEGIN
    select powers.name
      FROM powers, heroes, villains, sidekicks, h_link, s_link, v_link
     WHERE powers.po_id = h_link.po_id
           AND h_link.hid = heroes.hid
           AND heroes.name= who
     OR powers.po_id = v_link.po_id
           AND v_link.vid = villains.vid
           AND villains.name= who
     OR powers.po_id = s_link.po_id
           AND s_link.sid = sidekicks.sid
           AND sidekicks.name= who;
END;
$$ LANGUAGE PLPGSQL;

select personPower('Batman')
```

Triggers

CREATE OR REPLACE FUNCTION delHero()

RETURNS trigger AS

\$\$

BEGIN

DELETE FROM heroes

WHERE NOT EXISTS(select people.pid

from people

);

RETURN NULL;

END;

\$\$LANGUAGE plpgsql;

CREATE TRIGGER delHeroTrigger

AFTER DELETE ON people

FOR EACH ROW

EXECUTE PROCEDURE delHero();

Security

Since we don't want everyone and their mother to have the ability to add and delete whatever they want on a whim we will need some form of security to restrict who is able to do what. This is why there will be two kinds of users, admins and users. Admins will have full access to the database while users only have the select feature. And because admins is a privileged user its best to give them a basic password to begin with and can be altered at a later date.

CREATE ROLE admin

GRANT SELECT, INSERT, UPDATE, ALTER

ON ALL TABLES IN SCHEMA public

TO admin

password = 'admin'

CREATE ROLE user

GRANT SELECT

ON ALL TABLES IN SCHEMA public

TO user

Known Problems

- There are no known issues at this time.

Future Enhancements

- The biggest and most obvious future enhancement of this database would be to add more data. The DC Marvel universes are full of hundreds of heroes and villains that all deserve a place in this database in order to make this database the one stop for all the information you may need on anyone in that universe
- More entities can be added to further strengthen the relationship between all this data. For example a table can be added for all the non-hero/villain/sidekick characters who play a very important role in the story.