Kingdom Hearts

Database

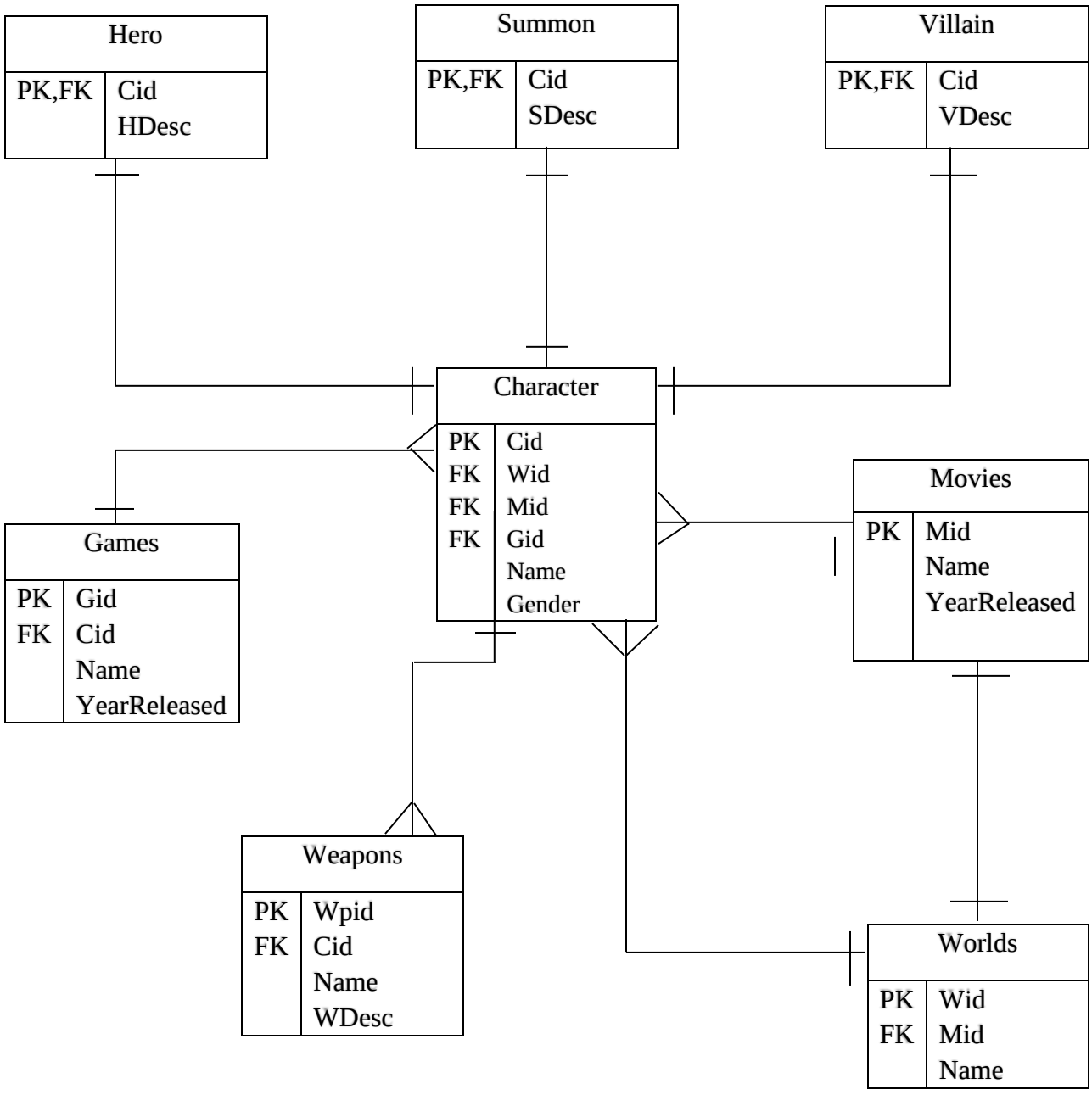
By Megan Crane

Table of contents

Executive Summary

This database is designed to store the characters in the well-known video game, Kingdom Hearts. The database stores information such as the character's name, world location, weapon, and tendency towards good or evil. The database also stores the information regarding the original movies and games characters were taken from as well as the years those movies and games were released. The goal of this database is to efficiently store the given data while also making it possible to view the connections between characters, movies, and games. This database was created with the flexibility in mind that new characters and worlds could be easily added as further Kingdom Hearts games are developed.

ER Diagram


Hero
PK,FK Cid
HDesc
Summon
PK,FK Cid
SDesc
Villain
PK,FK Cid
VDesc
Character
PK Cid
FK Wid
FK Mid
FK Gid
Name
Gender
Games
PK Gid
FK Cid
Name
YearReleased
Movies
PK Mid
Name
YearReleased
Weapons
PK Wpid
FK Cid
Name
WDesc
Worlds
PK Wid
FK Mid
Name

Create Statements

The purpose of this section is to show the create statements of the various tables as well as their functional dependencies and sample data.

Character Table

The Character Table consists of the fields Name and Gender. These attributes set a base for each character used in the game.

Functional Dependencies

Character: Cid $\longrightarrow$ Name, Gender

SQL Create Statement

```
drop table if exists Character;


create table Character (
Cid char(4) not null,
Name text not null,
Gender text not null,
Wid char(4) references Worlds(Wid),
Mid char(4) references Movies(Mid),
Gid char(4)  references Games(Gid),
primary key (cid)
);
```

Sample Data

Data Output | Explain | Messages | History

	cid character(4)	name text	gender text	wid character(4)	mid character(4)	gid character(4)
1	0001	Alice	Female	0001	0004	
2	0002	Ariel	Female	0004	0006	
3	0003	Beast	Male		0007	
4	0004	Cloud Strife	Male			0002
5	0005	Donald Duck	Male			
6	0006	Hades	Male			
7	0007	Sephiroth	Male			0002
8	0008	Sora	Male			0001
9	0009	Merlin	Male		0002	
10	0010	Minnie Mouse	Female		0001	
11	0011	Kairi	Female			0001
12	0012	Tarzan	Male	0002	0005	
13	0013	Mushu	Male			
14	0014	Genie	Male	0003	0008	

Hero Table

The Hero Table consists of the field name HDesc. This gives a short description of who the character is.

Functional Dependencies

Hero: Cid $\longrightarrow$ HDesc

SQL Create Statement

drop table if exists Hero;

create table Hero (

HDesc text not null,

Cid char(4) not null references Character(Cid),

primary key(Cid)

);

Sample Data

Data Output | Explain | Messages | History

	hdesc text	cid character(4)
1	Female Protagonist, she is a Princess of Heart who falls down a rabbit hole.	0001
2	Female Protagonist, a mermaid curious about the world above the sea.	0002
3	Male Protagonist, cursed to be a monster until he finds true love.	0003
4	Male Protagonist, he is recognized by his trademark enormouse buster sword.	0004
5	Male Protagonist, a duck with speech problems, temper issues, and strings of bad luck.	0005
6	Male Protagonist, he is the wielder of the keyblade.	0008
7	Male Protagonist, he is a wise old wizard.	0009
8	Female Protagonist, she is the wife of Mickey.	0010
9	Female Protagonist, she is a Princess of Heart and one of Soras best friends.	0011
10	Male Protagonist, he is a jungle man raised by apes.	0012

Villain Table

The Villain Table consists of the field VDesc. This gives a short description of the character in question.

Functional Dependencies

Villain: Cid $\longrightarrow$ VDesc

SQL Create Statement

drop table if exists Villain;

create table Villain (

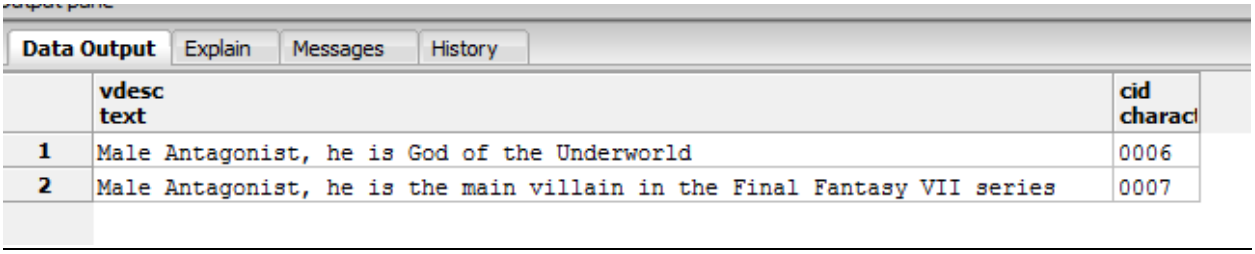
VDesc text not null,

Cid char(4) not null references Character(Cid),

primary key (Cid)

);

Sample Data



Data Output | Explain | Messages | History

	vdesc text	cid charac
1	Male Antagonist, he is God of the Underworld	0006
2	Male Antagonist, he is the main villain in the Final Fantasy VII series	0007

Summon Table

The Summon Table has the field SDesc. This gives a short description of the character being looked at.

Functional Dependencies

Summon: Cid $\longrightarrow$ SDesc

SQL Create Statement

drop table if exists Summon;

create table Summon (

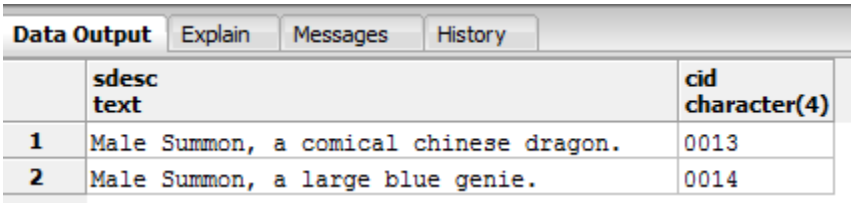
SDesc text not null,

Cid char(4) not null references Character(Cid),

primary key (cid)

);

Sample Data



Data Output | Explain | Messages | History

	sdesc text	cid character(4)
1	Male Summon, a comical chinese dragon.	0013
2	Male Summon, a large blue genie.	0014

Movies Table

The Movies Table has the fields Name and YearReleased. These help to keep track of the distinct movies involved in the game.

Functional Dependencies

Movies: Mid ⟶ Name, YearReleased

SQL Create Statement

delete if exists table Movies

create table Movies (

Mid char(4) not null,

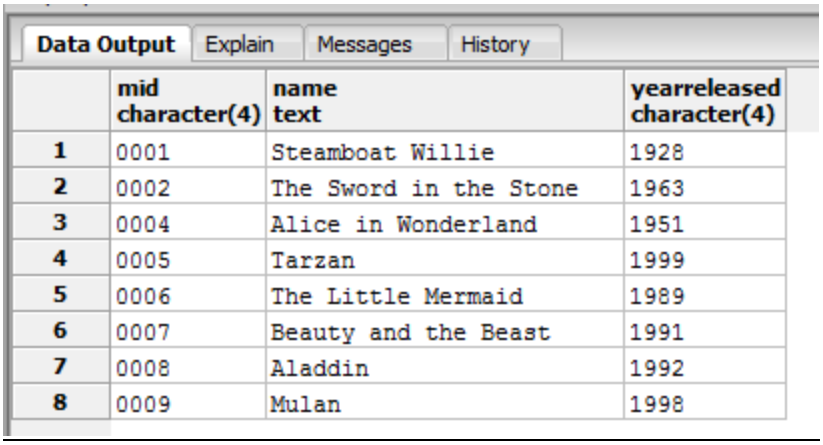
Name text not null,

YearReleased char(4) not null,

primary key (Mid)

);

Sample Data



Data Output | Explain | Messages | History

	mid character(4)	name text	yearreleased character(4)
1	0001	Steamboat Willie	1928
2	0002	The Sword in the Stone	1963
3	0004	Alice in Wonderland	1951
4	0005	Tarzan	1999
5	0006	The Little Mermaid	1989
6	0007	Beauty and the Beast	1991
7	0008	Aladdin	1992
8	0009	Mulan	1998

Games Table

The Games Table includes the field values Name and YearReleased. These values keep track of the distinct games references in the Kingdom Hearts game.

Functional Dependencies

Games: Gid ⟶ Name, YearReleased

SQL Create Statement

drop if exists table Games;

create table Games (

Gid char(4) not null,

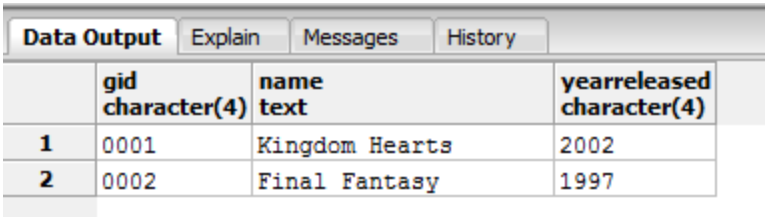
Name text not null,

YearReleased char(4),

primary key (gid)

);

Sample Data



Data Output | Explain | Messages | History

	gid character(4)	name text	yearreleased character(4)
1	0001	Kingdom Hearts	2002
2	0002	Final Fantasy	1997

Worlds Table

The Worlds Table has the field value Name. This distinguishes one game world from another.

Functional Dependencies

Worlds: Wid $\longrightarrow$ Name

SQL Create Statement

drop table if exists Worlds;

create table Worlds (

Wid char(4) not null,

Name text not null,

Mid char(4) not null references Movies(Mid),

primary key (wid)

);

Sample Data

Data Output | Explain | Messages | History

	wid character(4)	name text	mid character(4)
1	0001	Wonderland	0004
2	0002	Deep Jungle	0005
3	0003	Agrabah	0008
4	0004	Atlantica	0006

Weapon Table

The Weapon Table has the field values Name and Description. These two values distinguish the many types of weapons available in the game.

Functional Dependencies

Weapon: WPid ⟶ Name, Description

SQL Create Statement

```
drop table if exists Weapon;


create table Weapon (
WPid char(4) not null,
Name text not null,
Description text not null,
Cid char(4) references Character(Cid),
primary key (WPid)
);
```

Sample Data

Data Output | Explain | Messages | History

	wpid character(4)	name text	description text	cid character(4)
1	0001	Kingdom Key	The key chain attached draws out the Keyblades true form and power.	0008
2	0002	Ultima Weapon	The ultimate Keyblade. Raises max MP by 2 and possesses mx power and attributes.	0008
3	0003	Mages Staff	A staff that heightens magic power.	0005
4	0004	Save the Queen	A staff of immense magical and physical power. Raises max MP by 2.	0005

This view is built up of a villain's name, description, and world location. The view Villain_Location is beneficial because it will show the updated locations of any villain if they jump to a different world. A view is helpful in this situation because views show current up-to-date data.

```
create view Villain_Location AS

        select    c.name,
                  w.name,
                  VDesc
        from      character c,
                  villain v,
                  worlds w
        where     c.cid = v.cid
                  and c.wid = w.wid
        order by c.name asc
```

These two reports are useful for anyone who wants to know which weapons each hero can use as well as the statistics of the weapons in order to choose the best weapon for your situation, as well as where the various characters used in the game originated from in case a player wants to watch the movie they came from or play the game.

This Report shows the weapons a character can use and their statistics

```
select c.name,
       c.cid,
       HDesc,
       WPid,
       w.name,
       Description
from character c,
    hero h,
    weapon w
where     w.cid = c.cid
        and h.cid = c.cid
order by c.cid asc
```

This Report shows the character names descriptions and original games or movies

```
select  c.name,
           c.cid,
           c.mid,
```

```
        c.gid,
        m.name,
        g.name
from character c
full outer join movies m
on m.mid = c.mid
full outer join games g
on g.gid = c.gid;
```

Stored Procedures

This is a Stored Procedure to find out which world each hero is located in. It gives the character's name, their description, and the world's name by referencing the Character Table, the Hero Table and the Worlds Table.

```
create or replace function HeroWorld(int, refcursor) returns refcursor as $$
declare
        world_num int := $1;
        resultset refcursor := $4;
begin
        open resultset for
                select c.name, w.wid, w.name, h.hdesc
                from character c,
                        weapons w,
                        hero h
                where world_num in (c.wid)
                and c.cid = h.cid;
        return resultset;
end;
$$
language plpgsql;
select HeroWorld(0001, 'results');
fetch all from results;
```

This is a Stored Procedure that references to the Trigger that checks if a value of null was placed in the movie name field. If it was, the value is changed to 'Unknown Movie Title'.

```
create function MovieTitle()
return trigger as $$
begin
	if (name = null) then
		update movies set name = 'Unkown Movie Title' where name = null;
	end if;
end
$$language plpgsql;
```

This Trigger checks the values entered into the Movies Table. The Trigger calls the Stored Procedure 'MovieTitle' and runs the Stored Procedure to see if any null values were entered in the name field and if they were, it swaps the null value for the assigned one.

```
Create Trigger movie_title_trigger
    After insert or update
    On Movies
    For each row
    Execute Procedure MovieTitle();
```

Security

Create User KHAdmin with Password 'Alpaca';

Revoke all on Character from KHAdmin;

Revoke all on Hero from KHAdmin;

Revoke all on Villain from KHAdmin;

Revoke all on Summon from KHAdmin;

Revoke all on Worlds from KHAdmin;

Revoke all on Weapon from KHAdmin;

Revoke all on Movies from KHAdmin;

Revoke all on Games from KHAdmin;

Grant insert, update, delete, select on Character to KHAdmin;

Grant insert, update, delete, select on Hero to KHAdmin;

Grant insert, update, delete, select on Villain to KHAdmin;

Grant insert, update, delete, select on Summon to KHAdmin;

Grant insert, update, delete, select on Worlds to KHAdmin;

Grant insert, update, delete, select on Weapon to KHAdmin;

Grant insert, update, delete, select on Movies to KHAdmin;

Grant insert, update, delete, select on Games KHAdmin;

Create User KHUser with Password 'Alpaca'

```
Revoke all on Character from KHUser;

Revoke all on Hero from KHUser;

Revoke all on Villain from KHUser;

Revoke all on Summon from KHUser;

Revoke all on Worlds from KHUser;

Revoke all on Weapon from KHUser;

Revoke all on Movies from KHUser;

Revoke all on Games from KHUser;


Grant select on Character to KHUser;

Grant select on Hero to KHUser;

Grant select on Villain to KHUser;

Grant select on Summon to KHUser;

Grant select on Worlds to KHUser;

Grant select on Weapon to KHUser;

Grant select on Movies to KHUser;

Grant select on Games to KHUser;
```

Implementation, Known Problems, & Future Enhancements

Implementation

This database should be easy to implement because all of the create statements have been developed already. As long as the user uses those exact statements, there should be minimal issues. There is always the flexibility of creating more views based off of the specific information that the user wants.

Known Problems

The only issue that stands out at this point is that all of the table ID numbers have a character limit of 4. This means that as more Kingdom Hearts games are created and characters and other various data is input, the database runs the possibility of running out of available space. The current space allowed is more than enough for the database currently, however that will become an issue as the database grows.

Future Enhancements

This database focuses a lot on individual character information in the Kingdom Hearts game. In the future , however, it could be expanded to include more information on special attack forms as well as power-ups and collaborative attack maneuvers and magic spells.