



Codd's Hearthstone Game

Database Design Project

Cristian Hernandez

April 20, 2016

Contents

I.	Executive Summary	2
II.	Entity Relationship Diagram	3
III.	Tables	4
IV.	Security	13
V.	Views	14
VI.	Stored Procedures	15
VII.	Implementations	16
VIII.	Known Problems	17
IX.	Future Enhancements	18

Executive Summary

Overview

Hearthstone is an online card game that was created by Blizzard Entertainment on March 11, 2014. The game consists of two players facing each other with decks that consist of 30 cards each and the objective is to use minions and spells to take their opponent's health to 0. The players choose a "class hero" and build a deck around it. Each hero has a unique "hero power" that does something in game which allows a large variety of decks built for different class heroes.

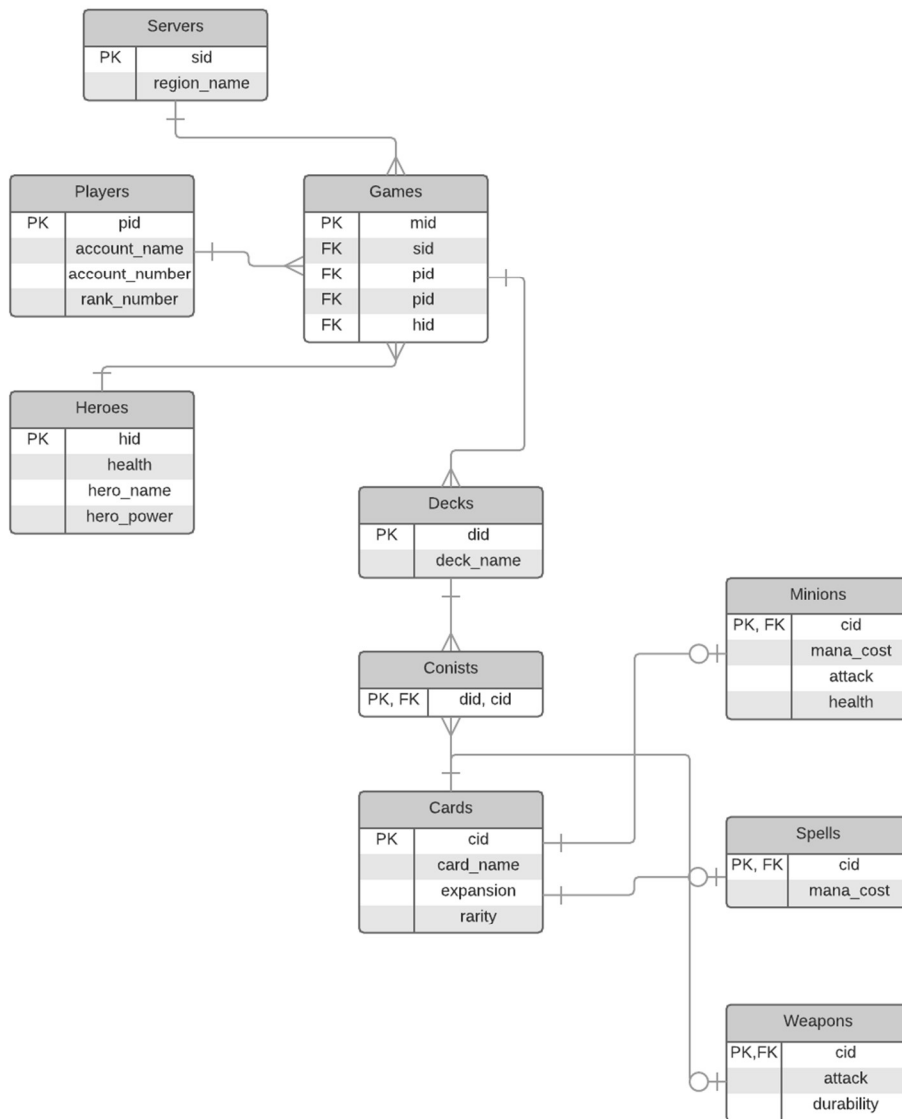
Hearthstone has become a huge success where the game has been played all over the world, tournaments have been held worldwide, in college or anywhere else. The game creates expansions which add new cards into the game to keep people engaged into the game and to keep their creative minds going into making new decks into the game.

Objectives

The purpose of this document is to provide an outline of a database system in which the game is run through players, decks, heroes, and the types of cards in the game. The database provides information on the cards stored in the game through specific details such as the health, attack, mana cost of minion, or if it is a spell, the cost of the spell and what it does.

This document will demonstrate a basic overview of the database, providing the tables of certain aspects of the table, views, reports, stored procedures, triggers, and security. This design was created and tested on PostgreSQL 9.4

Entity Relationship Diagram



Tables

Players

Purpose: The purpose of the Players table is to provide the information of each player in the game by showing the players account name, account number and their rank in the game so far.

Statement

```
CREATE TABLE Players (  
  pid CHAR(3) Not null,  
  account_name TEXT,  
  account_number NUMERIC(4),  
  rank_number INTEGER,  
  PRIMARY KEY (pid)  
);
```

Functional Dependencies

pid → account_name, account_number, rank_number

Sample Data

	Data Output	Explain	Messages	History
	pid character(3)	account_name text	account_number numeric(4,0)	rank_number integer
1	p01	PabloTheGreat	3003	1
2	p02	Chez	1996	2
3	p03	Crazygirl	1008	16
4	p04	DarenP	1234	25

Heroes

Purpose: The purpose of the Heroes table is to show the playable characters in the game, also showing the name of the character, the hero power used in the game, and the health of each hero.

Statement

```
CREATE TABLE Heroes (  
  hid CHAR(2) NOT NULL,  
  hero_name TEXT,  
  hero_power TEXT,  
  health numeric(2),  
  PRIMARY KEY (hid)  
);
```

Functional Dependencies

hid → hero_name, hero_power, health

Sample Data

	hid character(2)	hero_name text	hero_power text	health numeric(2,0)
1	h1	Mage	Fireblast	30
2	h2	Warrior	Armor Up!	30
3	h3	Shaman	Totemic Call	30
4	h4	Rogue	Dagger Mastery	30
5	h5	Hunter	Steady Shot	30
6	h6	Druid	Shapeshift	30
7	h7	Paladin	Reinforce	30
8	h8	Warlock	Lifetap	30
9	h9	Priest	Lesser Heal	30

Cards

Purpose: The purpose of the Cards table is to show a sample of cards of the game. The table shows the name of the card, how rare the card is, the type of card it is (i.e. minion, spell, or weapon) and the expansion of the card.

Statement

```
CREATE TABLE Cards (  
  cid CHAR(4) NOT NULL,  
  card_name TEXT,  
  card_type TEXT,  
  rarity TEXT,  
  expansion TEXT,  
  PRIMARY KEY (cid)  
);
```

Functional Dependencies

cid → card_name, card_type, rarity, expansion

Sample Data

Output pane

	cid character(4)	card_name text	card_type text	rarity text	expansion text
1	c001	Arcane Blast	Spell	Epic	The Grand Tournament
2	c002	Arcane Missiles x2	Spell	Common	Classic
3	c003	Mana Wurm x2	Minion	Common	Classic
4	c004	Flamecannon	Spell	Common	Goblins and Gnomes
5	c005	Frostbolt x2	Spell	Common	Classic
6	c006	Unstable Portal x2	Spell	Rare	Goblins and Gnomes
7	c007	Arcane Intellect x2	Spell	Common	Classic
8	c008	Effigy x2	Spell	Common	The Grand Tournament
9	c009	Flamewaker x2	Minion	Rare	Blackrock Mountain
10	c010	Mirror Entity	Spell	Common	Classic
11	c011	Fireball x2	Spell	Common	Classic
12	c012	Ethereal Conjurer x2	Minion	Common	League of Explorers
13	c013	Mad Scientist x2	Minion	Common	Curse of Naxxramas
14	c014	Piloted Shredder x2	Minion	Common	Goblins and Gnomes
15	c015	Sludge Belcher	Minion	Rare	Curse of Naxxramas
16	c016	Dr. Boom	Minion	Legendary	Goblins and Gnomes

Minions

Purpose: The purpose of the minions table is to show the cards that are minions and it shows their mana cost, attack, and health.

Statement

```
CREATE TABLE Minions (  
  cid CHAR(4) NOT NULL REFERENCES cards(cid),  
  manacost INTEGER,  
  attack INTEGER,  
  health INTEGER,  
  PRIMARY KEY(cid)  
);
```

Functional Dependencies

cid → manacost, attack, health

Sample Data

	Data Output	Explain	Messages	History
	cid character(4)	manacost integer	attack integer	health integer
1	c003	1	1	3
2	c009	3	2	4
3	c012	5	6	3
4	c013	2	2	2
5	c014	4	4	3
6	c015	5	3	5
7	c016	7	7	7
8	c017	8	8	8
9	c020	2	1	4
10	c028	6	5	5
11	c029	8	4	10
12	c030	2	0	7
13	c031	3	1	3
14	c032	3	4	2
15	c033	5	5	4
16	c034	6	6	3

Spells

Purpose: The spells table is used to show the spell cards in the game that provides the cost of the card and its id corresponding to the card.

Statement

```
CREATE TABLE Spells (  
  cid CHAR(4) NOT NULL REFERENCES cards(cid),  
  manacost INTEGER,  
  PRIMARY KEY(cid)  
);
```

Functional Dependencies

$cid \rightarrow manacost$

Sample Data

Output pane		
Data Output Explain Message		
	cid character(4)	manacost integer
1	c001	1
2	c002	1
3	c004	2
4	c005	2
5	c006	2
6	c007	3
7	c008	3
8	c010	3
9	c011	4
10	c018	1
11	c019	1
12	c022	2
13	c023	2
14	c024	3
15	c025	3
16	c027	5

Weapons

Purpose: The purpose for weapons is to show the weapon ids in the game including, their mana cost, the attack on the weapon, and the durability on the weapon (Amount of times you can use the weapon).

Statement

```
CREATE TABLE Weapons (  
  cid CHAR(4) NOT NULL REFERENCES cards(cid),  
  manacost INTEGER,  
  attack INTEGER,  
  durability INTEGER,  
  PRIMARY KEY(cid)  
);
```

Functional Dependencies

cid → manacost, attack, durability

Sample Data

	Data Output	Explain	Messages	History
	cid character(4)	manacost integer	attack integer	durability integer
1	c021	2	3	2
2	c026	4	4	2
3	c047	5	2	8

Decks

Purpose: The purpose of the Decks table is to provide the deck name of the deck that is used by the player.

Statement

```
CREATE TABLE Decks (  
  did CHAR(2) NOT NULL,  
  deck_name TEXT,  
  PRIMARY KEY (did)  
);
```

Functional Dependencies

did \rightarrow deck_name

Sample Data

	did character(2)	deck_name text
1	m1	Tempo Mage
2	w2	Control Warrior
3	s3	Aggro Shaman
4	r4	Raptor Rogue

Servers

Purpose: The purpose for the Servers table is to provide the servers in different regions of the game.

Functional Dependencies

sid \rightarrow region_name

Sample Data

	sid character(3)	region_name text
1	s01	Americas
2	s02	Europe
3	s03	Asia

Games

Purpose: The purpose of the Games table is to provide an online match set up with the server, two players, and the heroes that they are using to play each other with.

Statement

```
CREATE TABLE Games (  
  mid CHAR(4) NOT NULL,  
  sid char(3) NOT NULL REFERENCES servers(sid),  
  pid_1 char(3) NOT NULL REFERENCES players(pid),  
  pid_2 char(3) NOT NULL REFERENCES players(pid),  
  hid_1 char(2) NOT NULL REFERENCES heroes(hid),  
  hid_2 char(2) NOT NULL REFERENCES heroes(hid),  
  PRIMARY KEY (mid)  
);
```

Functional Dependencies

$\text{mid} \rightarrow \text{sid}, \text{pid}_1, \text{pid}_2, \text{hid}_1, \text{hid}_2$

Sample Data

	mid character(4)	sid character(3)	pid_1 character(3)	pid_2 character(3)	hid_1 character(2)	hid_2 character(2)
1	m001	s01	p01	p02	h1	h2
2	m002	s02	p03	p04	h3	h4
3	m003	s03	p01	p03	h1	h3
4	m004	s01	p02	p04	h2	h4
5	m005	s02	p01	p04	h1	h4
6	m006	s03	p02	p03	h2	h3

Consists

Purpose: The purpose of the consists table is to show the deck and the cards that are corresponded to each deck.

Statement

```
CREATE TABLE Consists (  
  did CHAR(2) NOT NULL REFERENCES decks(did),  
  cid char(4) NOT NULL REFERENCES cards(cid),  
  PRIMARY KEY(did, cid)  
);
```

Functional Dependencies

(did, cid) →

Sample Data

	did character(2)	cid character(4)
1	m1	c001
2	m1	c002
3	m1	c003
4	m1	c004
5	m1	c005
6	m1	c006
7	m1	c007
8	m1	c008
9	m1	c009
10	m1	c010
11	m1	c011
12	m1	c012
13	m1	c013
14	m1	c014
15	m1	c015
16	m1	c016

Security

There are only two primary users in this database: players and servers. Both users have certain privileges throughout the game.

Players

Players would be able to have access to their cards, games played, and decks because they would need access to these aspects in order to play the game.

Grant Update on decks To players;
Grant Delete on decks To Players;
Revoke Delete on Games to Players;
Revoke Insert on Cards to Players;

Server

Servers would need access to everything else in the game such as the games, cards, players, and heroes in order to update information, add new cards, update them and so on to keep the game running properly.

Grant Insert on players to servers;
Grant Insert on cards to servers;
Grant update on games to servers;
Grant Insert on heroes to servers;
Grant delete on cards to servers;
Grant update on cards to servers;

Views

CreatedDeck

Purpose: The purpose of the Created Deck is to provide the deck names and card names of each card in the deck created.

Statement

```
Create View CreatedDeck As
select deck_name, card_name
FROM decks INNER JOIN consists ON decks.did = consists.did
        INNER JOIN cards ON cards.cid = consists.cid
```

Sample Data

	deck_name text	card_name text
1	Tempo Mage	Arcane Blast
2	Tempo Mage	Arcane Missiles x2
3	Tempo Mage	Mana Wurm x2
4	Tempo Mage	Flamecannon
5	Tempo Mage	Frostbolt x2
6	Tempo Mage	Unstable Portal x2
7	Tempo Mage	Arcane Intellect x2
8	Tempo Mage	Effigy x2
9	Tempo Mage	Flamewaker x2
10	Tempo Mage	Mirror Entity
11	Tempo Mage	Fireball x2
12	Tempo Mage	Ethereal Conjurer x2
13	Tempo Mage	Mad Scientist x2
14	Tempo Mage	Piloted Shredder x2
15	Tempo Mage	Sludge Belcher
16	Tempo Mage	Dr. Boom

Stored Procedures

Getting cards by their rarity.

Purpose: The procedure helps finding cards by how rare they are because when sorting through them, you can find out how many cards are by how rare they are.

Statement

```
create or replace function get_card_by_mana_rare(text, REFCURSOR) returns refcursor as
$$
declare
    Rare text    := $1;
    card_name REFCURSOR := $2;
begin
    open card_name for
        select cards.card_name
        from   cards
        where  cards.rarity = Rare;
    return card_name;
end;
$$
language plpgsql;

select get_card_by_mana_rare('Legendary', 'results');
Fetch all from results;
```

Sample data

	card_name text
1	Dr. Boom
2	Ragnaros the Firelord
3	Grommash Hellscream
4	Harrison Jones
5	Justicar Trueheart
6	Sylvanas Windrunner
7	Baron Geddon
8	Sir Finley Mrrgglton
9	Loatheb

Implementations

A couple of implementations that can be allowed for later on in the game are new heroes. There should be a system when new heroes are created, their cards should have no problem corresponding with the hero and fill factors to insert the cards without any issue of sorting the cards later on.

In addition, the game would create new deck slots and it was cause corruption in the database there are only amount of spaces available in the tables. Fill factors may be used to leave space later on so the new slots will be available for use.

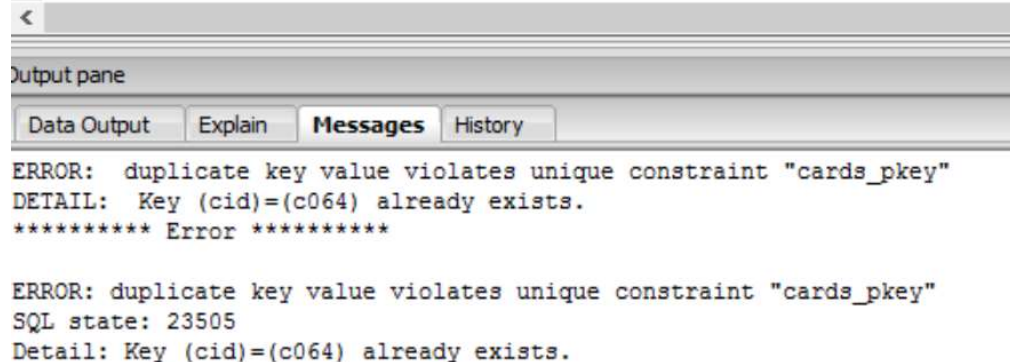
Known Problems

In the game of Hearthstone decks are allowed to use the same card twice in their deck according to how rare it is. Every card but legendary cards are allowed to be used twice in each deck. Normalization causes a huge problem when it comes to trying to organize data in the normal forms.

Two copies of the same card serve the same purpose, has no difference beside the turn you use it in game. One problem that occurs is that there can be confusion with the data is that since it's the same card it is going to use the same card ID.

```
INSERT INTO Cards(cid, card_name, rarity, card_type, expansion)
VALUES('c064', 'Shielded Minibot', 'Common', 'Minion', 'Classic');

INSERT INTO Cards(cid, card_name, rarity, card_type, expansion)
VALUES('c064', 'Shielded Minibot', 'Common', 'Minion', 'Classic');
```



The screenshot shows a database client interface. At the top, there is a text area containing two SQL INSERT statements. The first statement is: `INSERT INTO Cards(cid, card_name, rarity, card_type, expansion) VALUES('c064', 'Shielded Minibot', 'Common', 'Minion', 'Classic');`. The second statement is identical. Below the text area is a tabbed interface with four tabs: 'Data Output', 'Explain', 'Messages', and 'History'. The 'Messages' tab is selected. The output pane displays the following error message:
ERROR: duplicate key value violates unique constraint "cards_pkey"
DETAIL: Key (cid)=(c064) already exists.
***** Error *****

ERROR: duplicate key value violates unique constraint "cards_pkey"
SQL state: 23505
Detail: Key (cid)=(c064) already exists.

As shown above, you wouldn't be able to have the same card with the same ID, it wouldn't work in that case.

Future Enhancements

In the future some of these things could be helpful for later on:

- A way of identifying the same card twice since they are allowed in decks
- In the game, golden cards working as the same as regular cards
- Not getting the same card a third time or more in the game