



Disclaimer: Hearthstone Logo  
Property of Blizzard Entertainment

**By Thomas Famularo  
12/8/2016**

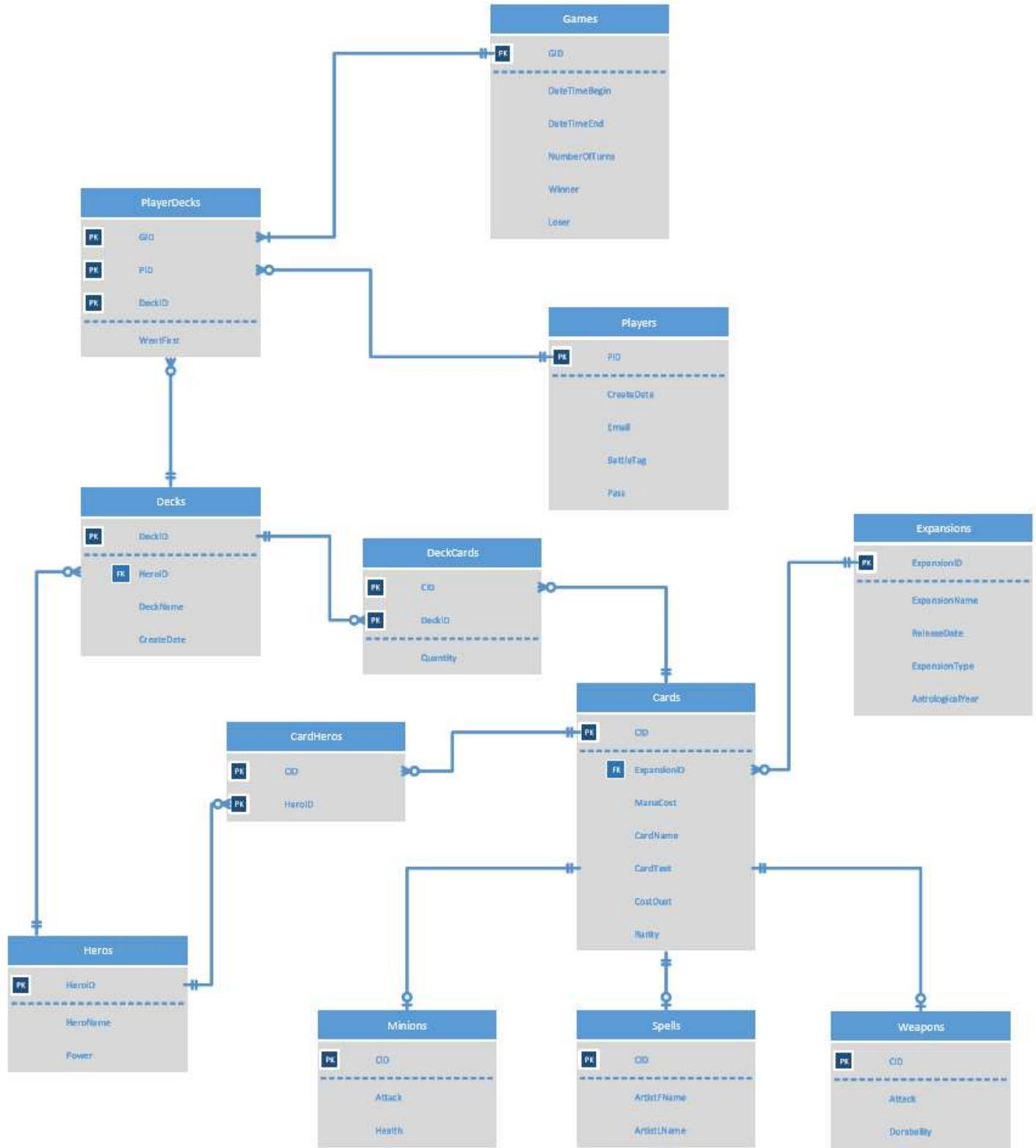
## Table of Contents

Executive Summary.....	3
Entity Relationship Diagram .....	4
Tables .....	5
Expansions .....	5
Cards .....	6
Minions .....	7
Spells .....	7
Weapons .....	9
Heros .....	10
CardHeros .....	11
Decks .....	12
DeckCards .....	13
Players .....	14
Games .....	15
PlayerDecks.....	16
Views.....	17
CardsInDeck2 .....	17
TotalDustCost.....	18
CountPlayerUseDeck.....	19
CardAndMinion .....	20
CardAndSpell.....	21
CardAndWeapon.....	22
Stored Procedures .....	23
InsertMinion.....	23
InsertSpell .....	24
InsertWeapon .....	25
InsertPlayer .....	26
CardsInDeck .....	27
Triggers.....	28
maxCardsInDeck.....	28
Security .....	29
Future Enhancements.....	31

## Executive Summery

HearhstoneDB has been crated as a new and innovate way to record Hearthstone Cards and Decks. One of the biggest problems with Hearthstone currently is the limit to how many decks the game will let you store. HearthstoneDB is designed to fix that, it will allow for as many as you would like to be stored in its system.

# Entity Relationship Diagram



# Tables

## Expansions

### Purpose

A table to keep track of the expansions and its associated names, release dates, types and Astrological Years.

### Create Statement

```
CREATE TABLE Expansions (
    ExpansionID      INT          NOT NULL,
    ExpansionName    CHAR(50)     NOT NULL,
    ReleaseDate      DATE        NOT NULL,
    ExpansionType    ExpansionType NOT NULL,
    AstrologicalYear AstrologicalYear ,
    PRIMARY KEY(ExpansionID)
);
```

### Functional Dependencies

Expansion ID → ExpansionName, ReleaseDate, ExpansionType, AstrologicalYear

### Sample Data

expansionid integer	expansionname character(50)	releasedate date	expansiontype expansiontype	astrologicalyear astrologicalyear
0	Classic	2014-03-11	Set	<NULL>
1	Goblins vs Gnomes	2014-12-08	Set	Horse
2	Blackrock Mountain	2015-04-02	Adventure	Sheep
3	The Grand Tournament	2015-08-24	Set	Sheep
4	The League of Explorers	2015-11-12	Adventure	Sheep
5	Whispers of the Old Gods	2016-04-26	Set	Monkey
6	One Night in Karazhan	2016-08-11	Adventure	Monkey
7	Mean Streets of Gadgetzan	2016-12-02	Set	Monkey

## Cards

### Purpose

A table to keep track of the cards and their associated expansions, mana costs, names, text, costs in dust, and rarities.

### Create Statement

```
CREATE TABLE Cards(
    CID          INT          NOT NULL,
    ExpansionID INT          NOT NULL REFERENCES Expansions(ExpansionID),
    ManaCost     INT          NOT NULL,
    CardName     CHAR(40)     NOT NULL,
    CardText     TEXT         ,
    CostDust    INT          NOT NULL,
    Rarity      Rarity       NOT NULL,
PRIMARY KEY(CID)
);
```

### Functional Dependencies

CID → ExpansionID, ManaCost, CardName, CardText, CostDust, Rarity

### Sample Data

cid integer	expansionid integer	manacost integer	cardname character(40)	cardtext text	costdust integer	rarity rarity
4	7	10	Kun the Forgotten King	Choose One - Gain 10 Armor; or Refresh your Mana Crystals. Battlecry: If your deck has no duplicates, summon all Demons from your hand.	1600	Legendary
5	7	9	Krul the Unshackled	Battlecry: All targets are chosen randomly.	1600	Legendary
6	7	9	Mayor Noggenfogger	Battlecry: Deal 3 damage to all other characters.	40	Common
7	7	7	Abyssal Enforcer	Battlecry: Give a random minion in your hand +5/+5.	1600	Legendary
8	7	7	Don Han Cho	Taunt Battlecry: Give adjacent minions Divine Shield.	100	Rare
9	7	7	Grimestreet Protector	Battlecry: If your deck has no duplicates, the next spell you cast this turn costs (0).	1600	Legendary
10	7	7	Inkmaster Solia	Battlecry: Summon a(1) {0} Jade Golem. Give it Taunt. @Battlecry: Summon a Jade Golem.	40	Common
11	7	7	Jade Chieftain	Taunt	40	Common
12	7	6	Ancient of Blossoms	Battlecry and Deathrattle: Summon a(1) {0} Jade Golem. @ Battlecry and Deathrattle: Sum	1600	Legendary
13	7	6	Aya Blackpaw	Battlecry: Summon a 6/6 Ogre.	400	Common
14	7	6	Big-Time Racketeer	Battlecry: Silence a minion with Deathrattle.	400	Epic
15	7	6	Defias Cleaner	Battlecry: If you control a minion with 6 or more Health, draw two cards.	400	Epic
16	7	6	Fight Promoter	Taunt Battlecry: Summon a(1) {0} Jade Golem.@{x}Taunt Battlecry: Summon a Jade Golem.	40	Common
17	7	6	Jade Behemoth	Costs (2) less for each Secret you've played this game.	100	Rare
18	7	6	Kabal Crystal Runner	At the end of your turn, add a random Demon to your hand.	400	Epic
19	7	6	Kabal Trafficker	Battlecry: If your opponent has 6 or more cards in hand, gain Charge.	400	Epic
20	7	6	Leatherclad Hogleader	Battlecry: If your weapon has at least 3 Attack, gain +4/+4.	400	Epic
21	7	6	Luckydo Buccaneer	Battlecry: Choose a friendly minion. Swap it with a minion in your deck.	1600	Legendary
22	7	6	Madam Goya	Whenever this attacks a minion and survives, draw a card.	400	Epic
23	7	6	Wind-up Burglebot	Taunt. Battlecry: Draw cards until you draw one that isn't a Dragon.	1600	Legendary
24	7	6	Wrathion	Taunt Whenever this minion deals damage, gain that much Armor.	100	Rare
25	7	5	Alley Armorsmith	Battlecry: Deal 5 damage to an enemy minion. Deathrattle: Deal 5 damage to your hero.	100	Rare
26	7	5	Bomb Squad			

## Minions

### Purpose

A table to keep track of the entity subtype of Minions and their associated attacks and health.

### Create Statement

```
CREATE TABLE Minions (
    CID           INT          NOT NULL,
    Attack        INT          NOT NULL,
    Health        INT          NOT NULL,
PRIMARY KEY(CID)
);
```

### Functional Dependencies

CID → Attack, Health

### Sample Data

cid integer	attack integer	health integer
4	7	7
5	7	9
6	5	4
7	6	6
8	5	6
9	6	6
10	5	5
11	5	5
12	3	8
13	5	3
14	1	1
15	5	7
16	4	4
17	3	6
18	5	5
19	6	6
20	6	6
21	5	5
22	4	3
23	5	5
24	4	5
25	2	7
26	2	2

# Spells

## Purpose

A table to keep track of the entity subtype of Spells and their associated Artists names.

## Create Statement

```
CREATE TABLE Spells(
    CID                      INT
    ArtistFName              CHAR(20)
    ArtistLName              CHAR(20)
PRIMARY KEY(CID)
);
```

## Functional Dependencies

CID → ArtistFName, ArtistLName

## Sample Data

cid integer	artistfname character(20)	artistlname character(20)
95	Evgeniy	Zaqumyenny
96	Arthur	Bozonnet
97	Charlene	Scanff
98	Arthur	Bozonnet
99	Zoltan	Boros
100	Mike	Saas
101	Tyler	West
102	Phil	Saunders
103	Raven	Mimura
104	Zoltan	Boros
105	Zolatan	Boros
106	Authur	Bozonnet
107	Daren	Bader
108	Konstantin	Turovec
109	Sean	McNally
110	Dan	Scott
111	Izzy	Hoover
112	Mark	Gibbons
113	Benjamin	Zhang
114	Kan	Lui
115	Matthew	Connor
116	Wayne	Reynolds
117	Matt	Dixon

# Weapons

## Purpose

A table to keep track of the entity subtype of Weapon and their associated attack and durability.

## Create Statement

```
CREATE TABLE Weapons (
    CID          INT          NOT NULL REFERENCES Cards(CID),
    Attack       INT          NOT NULL,
    Durability   INT          NOT NULL,
PRIMARY KEY(CID)
);
```

## Functional Dependencies

CID → Attack, Durability

## Sample Data

cid integer	attack integer	durability integer
1	2	4
2	2	3
3	2	2

## Heros

### Purpose

A table to keep track of Heroes and their associated name and power.

### Create Statement

```
CREATE TABLE Heros (
    HeroID          INT           NOT NULL,
    HeroName        CHAR(20)      NOT NULL,
    Power           TEXT          NOT NULL,
    PRIMARY KEY(HeroID)
);
```

### Functional Dependencies

HeroID → HeroName, Power

### Sample Data

heroid integer	heroname character(20)	power text
0	NONE	NONE
1	Druid	+1 Attack this turn. +1 Armor
2	Hunter	Deal 2 damage to the enemy hero.
3	Mage	Deal 1 damage.
4	Paladin	Summon a 1/1 Silver Hand Recruit.
5	Priest	Restore 2 Health.
6	Rogue	Equip a 1/2 Dagger.
7	Shaman	Summon a random Totem
8	Warlock	Draw a card and take 2 damage.
9	Warrior	Gain 2 Armor.

## CardHeros

### Purpose

A table to keep track of the relationship between cards and heroes.

### Create Statement

```
CREATE TABLE CardHeros(
    CID                      INT
    HeroID                   INT
PRIMARY KEY(CID, HeroID)
);
```

### Functional Dependencies

CID, HeroID →

### Sample Data

cid integer	heroid integer
1	2
2	9
3	7
4	1
5	8
6	0
7	8
8	4
8	2
8	9
9	4
10	3
11	7
12	0
13	1
13	6
13	7
14	0
15	0
16	0
17	1
18	3
19	8

## Decks

### Purpose

A table to keep track of Decks and their associated heros, names, and create dates.

### Create Statement

```
CREATE TABLE Decks (
    DeckID          INT           NOT NULL,
    HeroID          INT           NOT NULL REFERENCES Heros(HeroID),
    DeckName        CHAR(20)      NOT NULL,
    CreateDate      DATE          NOT NULL,
PRIMARY KEY(DeckID)
);
```

### Functional Dependencies

DeckID → HeroID, DeckName, CreateDate

### Sample Data

deckid integer	heroid integer	deckname character(20)	createdate date
1	2	Band of Beasts	2016-11-02
2	4	The Mean Market	2016-11-12
3	6	Gadgetzan Dockside	2016-11-22
4	8	Demons of Kabal	2016-12-02

## DeckCards

### Purpose

A table to keep track of the relationship between decks and cards.

### Create Statement

```
CREATE TABLE DeckCards (
    CID                      INT,
    DeckID                   INT,
    Quantity                 INT
PRIMARY KEY(CID, DeckID)
);
```

### Functional Dependencies

CID, DeckID → Quantity

### Sample Data

cid integer	deckid integer	quantity integer
1	1	2
8	1	2
46	1	2
69	1	2
80	1	2
81	1	2
86	1	2
110	1	2
118	1	2
119	1	2
6	1	2
74	1	2
78	1	2
33	1	2
121	1	2
8	2	2
9	2	2
32	2	2
63	2	2
82	2	2
90	2	2
91	2	2
107	2	2

# Players

## Purpose

A table to keep track of Players and their associated create dates, emails, battletags, and passwords.

## Create Statement

```
CREATE TABLE Players (
    PID           INT          NOT NULL,
    CreateDate   DATE         NOT NULL,
    Email         TEXT        NOT NULL,
    BattleTag    CHAR(25)    NOT NULL,
    Pass          CHAR(30)    NOT NULL,
PRIMARY KEY (PID)
);
```

## Functional Dependencies

PID → CreateDate, Email, BattleTag, Pass

## Sample Data

pid integer	createdate date	email text	battletag character(25)	pass character(30)
0	2014-03-11	HearthstoneSux@Blizzard.com	HSB#1100	JUSTKIDDING
1	2014-04-11	Oxbow@Drakes.ru	OxbowRU#1283	WhatTimeIsIt
2	2014-05-11	Idontcare@yahoo.com	Care#1235412	Idont
3	2014-06-11	What@isthis.kr	What#9269423	isthis
4	2014-07-11	JohnDefalcoIsTheWorst@mail.com	JohnSUX#1234	ihatehim
5	2014-08-11	Scott@gmail.com	Scott#000000	DragonBallz
6	2014-09-11	alan@labouseur.com	Alan#007	Idontknow
7	2014-10-11	alan@3NFconsulting.com	NF3#12343	18651894
8	2014-11-11	alan.labouseur@marist.edu	Marist#13132	489498658
9	2014-12-11	ThomasF152@yahoo.com	nPPredator#1100	Nicetry

## Games

### Purpose

A table to keep track of Games and their associated start times, end times, number of turns, winners and losers.

### Create Statement

```
CREATE TABLE Games (
    GID          INT          NOT NULL,
    DateTimeBegin TIMESTAMP    NOT NULL,
    DateTimeEnd   TIMESTAMP    NOT NULL,
    NumberOfTurns INT          NOT NULL,
    Winner        INT          NOT NULL REFERENCES Players(PID),
    Loser         INT          NOT NULL REFERENCES Players(PID),
PRIMARY KEY(GID)
);
```

### Functional Dependencies

GID → DateTimeBegin, DateTimeEnd, NumberOfTurns, Winner, Loser

### Sample Data

gid integer	datetimebegin timestamp without time zone	datetimeend timestamp without time zone	numberofturns integer	winner integer	loser integer
0	2016-12-07 00:00:00	2016-12-07 00:10:00	13	1	2
1	2016-12-07 01:00:00	2016-12-07 01:10:00	22	2	7
2	2016-12-07 02:00:00	2016-12-07 02:10:00	6	3	5
3	2016-12-07 03:00:00	2016-12-07 03:10:00	45	4	4
4	2016-12-07 04:00:00	2016-12-07 04:10:00	17	5	2
5	2016-12-07 05:00:00	2016-12-07 05:10:00	23	6	1
6	2016-12-07 06:00:00	2016-12-07 06:10:00	9	7	6
7	2016-12-07 07:00:00	2016-12-07 07:10:00	27	8	7
8	2016-12-07 08:00:00	2016-12-07 08:10:00	33	9	8
9	2016-12-07 09:00:00	2016-12-07 09:10:00	14	9	1
10	2016-12-07 10:00:00	2016-12-07 10:10:00	15	1	3
11	2016-12-07 11:00:00	2016-12-07 11:10:00	16	2	4
12	2016-12-07 12:00:00	2016-12-07 12:10:00	22	3	5
13	2016-12-07 13:00:00	2016-12-07 13:10:00	23	4	6
14	2016-12-07 14:00:00	2016-12-07 14:10:00	24	5	7
15	2016-12-07 15:00:00	2016-12-07 15:10:00	37	9	8
16	2016-12-07 16:00:00	2016-12-07 16:10:00	38	6	4
17	2016-12-07 17:00:00	2016-12-07 17:10:00	39	7	3
18	2016-12-07 18:00:00	2016-12-07 18:10:00	44	8	4
19	2016-12-07 19:00:00	2016-12-07 19:10:00	12	1	5
20	2016-12-07 20:00:00	2016-12-07 20:10:00	5	3	6
21	2016-12-07 21:00:00	2016-12-07 21:10:00	1	9	1
22	2016-12-07 22:00:00	2016-12-07 22:10:00	22	6	2

## PlayerDecks

### Purpose

A table to keep track of the relationship between players, decks and games.

### Create Statement

```
CREATE TABLE PlayerDecks(
    GID                      INT          NOT NULL REFERENCES GAMES(GID),
    PID                      INT          NOT NULL REFERENCES Players(PID),
    DeckID                   INT          NOT NULL REFERENCES Decks(DeckID),
    WentFirst                BOOLEAN     NOT NULL,
    PRIMARY KEY(GID, PID, DECKID)
);
```

### Functional Dependencies

GID, PID, DeckID → WentFirst

### Sample Data

gid integer	pid integer	deckid integer	wentfirst boolean
0	1	1	t
0	2	2	f
1	2	2	t
1	7	3	f
2	3	4	t
2	5	4	f
3	4	3	t
3	4	4	f
4	2	1	t
4	5	3	f
5	6	2	t
5	1	4	f
6	7	4	t
6	6	1	f
7	8	1	t
7	7	2	f
8	9	4	t
8	8	3	f
9	9	4	t
9	1	1	f
10	1	2	t
10	3	1	f
11	2	4	t

# Views

## CardsInDeck2

### Purpose

To be able to view the name of the card and how many there are of that card in the second deck.

### Create Statement

```
--The names and quantity of cards in a deck 2
CREATE OR REPLACE VIEW cardsInDeck2 AS
SELECT CardName, Quantity
    FROM DeckCards INNER JOIN Decks ON DeckCards.DeckID = Decks.DeckID
                  INNER JOIN Cards ON DeckCards.CID = Cards.CID
 WHERE DeckCards.DeckID = 2;
SELECT *
    FROM cardsInDeck2;
```

### Sample Data

cardname character(40)	quantity integer
Don Han Cho	2
Grimestreet Protector	2
Finja, the Flying Star	2
Grimestreet Enforcer	2
Grook Fu Master	1
Red Mana Wyrm	2
Second-Rate Bruiser	2
Backroom Bouncer	1
Daring Reporter	2
Grimestreet Pawnbroker	2
Grimestreet Outfitter	2
Meanstreet Marshal	2
Mistress of Mixtures	2
Weasel Tunneler	2
Small-Time Recruits	2
Smugglers Run	2

## TotalDustCost

### Purpose

To be able to view the total dust required to make all of the cards in each deck.

### Create Statement

```
--The total dust cost to create all decks
CREATE OR REPLACE VIEW totalDustCost AS
    SELECT DeckName, Sum(costDust)
        FROM DeckCards INNER JOIN Decks ON DeckCards.DeckID = Decks.DeckID
                    INNER JOIN Cards ON DeckCards.CID = Cards.CID
    GROUP BY (DeckName);
SELECT *
    FROM totalDustCost;
```

### Sample Data

deckname character(20)	sum bigint
Gadgetzan Dockside	6700
Demons of Kabal	17100
The Mean Market	5240
Band of Beasts	4960

## CountPlayerUseDeck

### Purpose

To be able to view how many times each player played each deck.

### Create Statement

```
--How many times each player played each deck
CREATE OR REPLACE VIEW countPlayerUseDeck AS
    SELECT Battletag, DeckName, Count(*)
        FROM PlayerDecks INNER JOIN Players ON PlayerDecks.PID = Players.PID
                                INNER JOIN Decks ON PlayerDecks.DeckID = Decks.DeckID
    GROUP BY (DeckName, Battletag)
    ORDER BY Battletag DESC;
    SELECT *
        FROM countPlayerUseDeck;
```

### Sample Data

battletag character(25)	deckname character(20)	count bigint
What#9269423	Band of Beasts	2
What#9269423	Demons of Kabal	2
What#9269423	The Mean Market	2
Scott#000000	Band of Beasts	2
Scott#000000	Demons of Kabal	1
Scott#000000	Gadgetzan Dockside	2
OxbowRU#1283	Band of Beasts	2
OxbowRU#1283	Demons of Kabal	2
OxbowRU#1283	The Mean Market	2
nPPredator#1100	Band of Beasts	2
nPPredator#1100	Demons of Kabal	3
NF3#12343	Band of Beasts	1
NF3#12343	Demons of Kabal	2
NF3#12343	Gadgetzan Dockside	1
NF3#12343	The Mean Market	1
Marist#13132	Band of Beasts	1
Marist#13132	Gadgetzan Dockside	2
Marist#13132	The Mean Market	1
JohnSUX#1234	Demons of Kabal	2
JohnSUX#1234	Gadgetzan Dockside	3
JohnSUX#1234	The Mean Market	1
Care#1235412	Band of Beasts	1
Care#1235412	Demons of Kabal	1

# CardAndMinion

## Purpose

To be able to view all the information about minion cards even though it is split between two tables.

## Create Statement

```
--View both the card and minion table
CREATE OR REPLACE VIEW cardAndMinion AS
    SELECT Cards.CID, ExpansionID, ManaCost, CardName, CardText, costdust, rarity, attack, health
        FROM Cards INNER JOIN Minions ON Cards.CID = Minions.CID
    ORDER BY Cards.CID DESC;
    SELECT *
        FROM cardAndMinion;
```

## Sample Data

cid integer	expansionid integer	manacost integer	cardname character(40)	cardtext text	costdust integer	rarity rarity	attack integer	health integer
94	7	1	Weasel Tunneler	Deathrattle: Shuffle this minion into your opponents deck.	300	Epic	1	1
93	7	1	Small-Time Buccaneer	Has +2 Attack While you have a weapon equipped.	100	Rare	1	2
92	7	1	Patches the Pirate	Charge After you play a Pirate, summon this minion from your deck.	1600	Legendary	1	1
91	7	1	Mistress of Mixtures	Deathrattle: Restore 4 Health to both players.	40	Common	2	2
90	7	1	Meanstreet Marshal	Deathrattle: If this minion has 2 or more Attack, draw a card.	500	Epic	1	2
89	7	1	Kabal Lackey	Battlecry: The next Secret you play this turn costs (0).	40	Common	2	1
88	7	1	Grimscale Chum	Battlecry: Give a random Murloc in your hand +1/+1.	60	Common	2	1
87	7	1	Alleycat	Battlecry: Summon a 1/1 Cat.	100	Rare	1	1
86	7	2	Iropic Seastrager	Battlecry: Give a random Beast in your hand +1/+1.	100	Rare	3	2
85	7	2	Mana Geedo	Whenever this minion is healed, summon a 2/2 Crystal.	500	Epic	2	3
84	7	2	Jade Swarm	Stealth Deathrattle: Summon a(1) (0) Jade Golem. \$Stealth Deathrattle: Summon a Jade Golem.	40	Common	1	1
83	7	2	Hobart Grapplehammer	Battlecry: Give all weapons in your hand and deck +1 Attack.	1600	Legendary	2	2
82	7	2	Grimestreet Outfitter	Battlecry: Give all minions in your hand +1/+1.	40	Common	1	1
81	7	2	Grimestreet Informant	Battlecry: Discover a Hunter, Paladin, or Warrior card.	300	Epic	1	1
80	7	2	Gadgetzan Socialite	Battlecry: Restore 2 Health.	40	Common	2	2
79	7	2	Gadgetzan Ferryman	Combo: Return a friendly minion to your hand.	100	Rare	2	3
78	7	2	Friendly Bartender	At the end of your turn, restore 1 Health to your hero.	60	Common	2	3
77	7	2	Dirty Rat	Taunt Battlecry: Your opponent summons a random minion from their hand.	400	Epic	2	6
76	7	2	Blowgill Sniper	Battlecry: Deal 1 damage.	40	Common	2	1
75	7	3	Wickerflame Burnbristle	Divine Shield. Taunt. Damage dealt by this minion also heals your hero.	2000	Legendary	2	2
74	7	3	Toxic Sewer Ooze	Battlecry: Remove 1 Durability from your opponents weapon.	40	Common	4	3
73	7	3	Shaky Zippunner	Deathrattle: Give a random minion in your hand +2/+2.	40	Common	3	3
72	7	3	Shaku, the Collector	Stealth. Whenever this attacks, add a random card to your hand (from your opponents class).	1600	Legendary	2	3

## CardAndSpell

### Purpose

To be able to view all the information about spell cards even though it is split between two tables.

### Create Statement

```
--View both the card and spell table
CREATE OR REPLACE VIEW cardAndSpell AS
    SELECT Cards.CID, ExpansionID, ManaCost, CardName, CardText, costdust, rarity, ArtistFName, ArtistLName
    FROM Cards INNER JOIN Spells ON Cards.CID = Spells.CID
    ORDER BY Cards.CID DESC;
SELECT *
    FROM cardAndSpell;|
```

### Sample Data

cid	expansionid	manacost	cardname	contexttext	costdust	rarity	artistfame	character(20)	artistname	character(20)
121	7	0	Counterfeit Coin	Gain 1 Mana Crystal this turn only.	100	Rare	Joe	Wilson		
120	7	1	Smugglers Run	Give all minions in your hand +1/+1.	40	Common	Alex	Olandelli		
119	7	1	Smugglers Crate	Give a random Beast in your hand +2/+2.	40	Common	Grace	Liu		
118	7	1	Potion of Madness	Gain control of an enemy minion with 2 or less Attack until end of turn.	40	Common	Arthur	Boronet		
117	7	1	Fins-Size Potion	Give all enemy minions -3 Attack this turn only.	100	Rare	Matt	Dixon		
116	7	1	Mark of the Lotus	Give your minions +1/+1.	40	Common	Wayne	Reynolds		
115	7	1	Jade Idol	Choose One - Summon a [1] [0] Jade Golem; or Shuffle 3 copies of this card into your deck; or Choose One - Summon a Jade Golem; or Shuffle 3 copies of this card into your deck.	120	Rare	Matthew	Connor		
114	7	1	Knife Guy	Discover a Taunt minion.	40	Common	Ken	Liu		
113	7	1	Random Reppers	Discover a card with Overload. Overload: (1)	400	Epic	Elamin	Zhang		
112	7	2	Stolen Goods	Give a random Taunt minion in your hand +3/+3.	100	Rare	Mark	Gibbons		
111	7	2	Jade Shuriken	Deal 2 damage. Combo: Summon a [0] Jade Golem. #Deal 2 damage. Combo: Summon a Jade Golem.	40	Common	Izzy	Hoores		
110	7	2	Hidden Cache	Secret: After your opponent plays a minion, give a random minion in your hand +2/+2.	100	Rare	Dan	Scott		
109	7	2	Devolve	Transform all enemy minions into random ones that cost (1) less.	100	Rare	Sean	McNally		
108	7	3	Volcanic Potion	Volcanic Potion	100	Rare	Konstantin	Turovec		
107	7	3	Small-Time Recruits	Draw three 1-Cost minions from your deck.	500	Epic	Dareen	Bader		
106	7	3	Potion of Polymorph	Secret: After your opponent plays a minion, transform it into a 1/1 Sheep.	120	Rare	Arthur	Boronet		
105	7	3	Plumed Power	Gain an empty Mana Crystal for each friendly minion.	450	Epic	Zolatan	Boros		
104	7	3	Sheep Skin	Summon a [1] [0] Jade Golem. #Gain an empty Mana Crystal. #Summon a Jade Golem. Gain an empty Mana Crystal.	40	Common	Zolatan	Boros		
103	7	3	Bloodfury Potion	Give a minion +3 Attack. If its a Demon, also give it +3 Health.	100	Rare	Raven	Murone		
102	7	4	Jade Lightning	Deal 4 damage. Summon a [0] Jade Golem. #Deal 4 damage. Summon a Jade Golem.	40	Common	Phil	Saunders		
101	7	4	Greater Healing Potion	Restore 12 Health to a friendly character.	120	Rare	Tyler	West		
100	7	4	Call in the Finishers	Call in the Finishers	60	Common	Mike	Saas		
99	7	4	Blastocrystal Potion	Destroy a minion and one of your Mana Crystals.	40	Common	Zolatan	Boros		

## CardAndWeapon

### Purpose

To be able to view all the information about weapon cards even though it is split between two tables.

### Create Statement

```
--View both the card and weapon table
CREATE OR REPLACE VIEW cardAndWeapon AS
    SELECT Cards.CID, ExpansionID, ManaCost, CardName, CardText, costdust, rarity, attack, durability
        FROM Cards INNER JOIN Weapons ON Cards.CID = Weapons.CID
    ORDER BY Cards.CID DESC;
SELECT *
    FROM cardAndWeapon;
```

### Sample Data

cid integer	expansionid integer	manacost integer	cardname character(40)	cardtext text	costdust integer	rarity rarity	attack integer	durability integer
3	7	3	Jade Claws	Battlecry: Summon a(1) {0} Jade Golem. Overload: (1){8}Battlecry: Summon a Jade Golem. Overload: (1)	100	Rare	2	2
2	7	4	Brass Knuckles	After your hero attacks, give a random minion in your hand +1/+1.	400	Epic	2	3
1	7	5	Piranha Launcher	After your hero attacks, summon a 1/1 Piranha.	400	Epic	2	4

# Stored Procedures

## InsertMinion

### Purpose

To be able to add a minion to the database without having to insert in to two tables, both the cards table and the minions tables.

### Create Statement

```
-- Stored Procedure to add a minion to the Database
CREATE OR REPLACE FUNCTION insertMinion(INT, INT, INT, CHAR(40), TEXT, INT, Rarity, INT, INT)
RETURNS void
AS
$$
DECLARE
    -- _ used to indicate a variable
    _CID           INT          := $1;
    _ExpansionID   INT          := $2;
    _ManaCost      INT          := $3;
    _CardName      CHAR(40)     := $4;
    _CardText      TEXT         := $5;
    _CostDust      INT          := $6;
    _Rarity        Rarity       := $7;
    _Attack         INT          := $8;
    _Health         INT          := $9;
BEGIN
    INSERT INTO Cards (CID, ExpansionID, ManaCost, CardName, CardText, CostDust, Rarity)
    VALUES (_CID, _ExpansionID, _ManaCost, _CardName, _CardText, _CostDust, _Rarity);
    INSERT INTO Minions (CID, Health, Attack)
    VALUES (_CID, _Health, _Attack);
END;
$$
LANGUAGE plpgsql;
```

## InsertSpell

### Purpose

To be able to add a spell to the database without having to insert in to two tables, both the cards table and the spells tables.

### Create Statement

```
--Stored Procedure to add a spell to the Database
CREATE OR REPLACE FUNCTION insertSpell(INT, INT, INT, CHAR(40), TEXT, INT, Rarity, CHAR(20), CHAR(20))
RETURNS void
AS
$$
DECLARE
-- _ used to indicate a variable
    _CID           INT          := $1;
    _ExpansionID   INT          := $2;
    _ManaCost      INT          := $3;
    _CardName      CHAR(40)     := $4;
    _CardText      TEXT         := $5;
    _CostDust      INT          := $6;
    _Rarity        Rarity       := $7;
    _ArtistFName   CHAR(20)     := $8;
    _ArtistLName   CHAR(20)     := $9;
BEGIN
    INSERT INTO Cards (CID, ExpansionID, ManaCost, CardName, CardText, CostDust, Rarity)
    VALUES (_CID, _ExpansionID, _ManaCost, _CardName, _CardText, _CostDust, _Rarity);
    INSERT INTO Spells (CID, ArtistFName, ArtistLName)
    VALUES (_CID, _ArtistFName, _ArtistLName);
END;
$$
LANGUAGE plpgsql;
```

## InsertWeapon

### Purpose

To be able to add a weapon to the database without having to insert in to two tables, both the cards table and the weapons tables.

### Create Statement

```
--Stored Procedure to add a weapon to the Database
CREATE OR REPLACE FUNCTION insertWeapon(INT, INT, INT, CHAR(40), TEXT, INT, Rarity, INT, INT)
RETURNS void
AS
$$
DECLARE
-- _ used to indicate a variable
    _CID          INT          := $1;
    _ExpansionID  INT          := $2;
    _ManaCost     INT          := $3;
    _CardName     CHAR(40)     := $4;
    _CardText     TEXT         := $5;
    _CostDust     INT          := $6;
    _Rarity       Rarity      := $7;
    _Attack       INT          := $8;
    _Durabality   INT          := $9;
BEGIN
    INSERT INTO Cards (CID, ExpansionID, ManaCost, CardName, CardText, CostDust, Rarity)
    VALUES (_CID, _ExpansionID, _ManaCost, _CardName, _CardText, _CostDust, _Rarity);
    INSERT INTO Weapons (CID, Attack, Durabality)
    VALUES (_CID, _Attack, _Durabality);
END;
$$
LANGUAGE plpgsql;
```

## InsertPlayer

### Purpose

A convenient way to insert a player into the database.

### Create Statement

```
-- Stored Procedure to add a player to the Database
CREATE OR REPLACE FUNCTION insertPlayer(INT, DATE, TEXT, CHAR(25), CHAR(30))
RETURNS void
AS
$$
DECLARE
-- _ used to indicate a variable
    _PID          INT            := $1;
    _CreateDate   DATE           := $2;
    _Emial        TEXT           := $3;
    _BattleTag    CHAR(25)       := $4;
    _Pass         CHAR(30)       := $5;
BEGIN
    INSERT INTO Players (PID, CreateDate, Emial, BattleTag, Pass)
    VALUES (_PID, _CreateDate, _Emial, _BattleTag, _Pass);
END;
$$
LANGUAGE plpgsql;
```

## CardsInDeck

### Purpose

To display the cards that are in any deck to be able to review the deck easier.

### Create Statement

```

CREATE OR REPLACE FUNCTION CardsInDeck(INT, refcursor)
RETURNS refcursor
AS
$$
DECLARE
    InputID          INT          := $1;
    resultset        refcursor   := $2;
BEGIN
    open resultset FOR
    SELECT CardName, Quantity
        FROM DeckCards INNER JOIN Decks ON DeckCards.DeckID = Decks.DeckID
                           INNER JOIN Cards ON DeckCards.CID = Cards.CID
    WHERE DeckCards.DeckID = InputID;
    RETURN resultset;
END;
$$
LANGUAGE plpgsql;

SELECT CardsInDeck(1, 'results');
FETCH ALL FROM results;

```

### Sample Data

cardname character(40)	quantity integer
Piranha Launcher	2
Don Han Cho	2
Dispatch Kodo	2
Rat Pack	2
Gadgetzan Socialite	2
Grimestreet Informant	2
Trogg Beastrager	2
Hidden Cache	2
Potion of Madness	2
Smugglers Crate	2
Mayor Noggenfogger	2
Toxic Sewer Ooze	2
Friendly Bartender	2
Grook Fu Master	2
Counterfeit Coin	2

# Triggers

## maxCardsInDeck

### Purpose

This exists to limit the number of cards that can be put in to any individual deck, this number cannot exceed 30.

### Create Statement

```
--Trigger to set the maximum number of cards in a deck to 30
CREATE OR REPLACE FUNCTION maxCardsInDeck()
RETURNS TRIGGER
AS
$$
DECLARE
    counter      INT;
BEGIN
    SELECT count(*) INTO counter
        FROM DeckCards
        WHERE deckID = NEW.deckID;
    IF (counter < 30)
        THEN
            --All good. Do nothing. There is no problem here.
        ELSE
            RAISE EXCEPTION 'A Deck cannot have more than 30 cards';
    END IF;
    RETURN new;
END;
$$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS maximumCards ON DeckCards;
CREATE TRIGGER maximumCards
    BEFORE INSERT
    ON DeckCards
    FOR EACH ROW
    EXECUTE PROCEDURE maxCardsInDeck();
```

# Security

The Database currently had 3 roles: Player, Developer, Admin

```
--Create Roles
CREATE ROLE Player;
CREATE ROLE Developer;
CREATE ROLE ADMIN;
```

Player has the ability to edit their own player information as well as their decks but nothing else

```
--Player
REVOKE ALL PRIVILEGES ON DeckCards FROM Player;
REVOKE ALL PRIVILEGES ON Heros FROM Player;
REVOKE ALL PRIVILEGES ON CardHeros FROM Player;
REVOKE ALL PRIVILEGES ON Cards FROM Player;
REVOKE ALL PRIVILEGES ON Games FROM Player;
REVOKE ALL PRIVILEGES ON Minions FROM Player;
REVOKE ALL PRIVILEGES ON Weapons FROM Player;
REVOKE ALL PRIVILEGES ON Spells FROM Player;
REVOKE ALL PRIVILEGES ON Expansions FROM Player;
GRANT SELECT, INSERT, UPDATE, DELETE ON Players TO Player;
GRANT SELECT, INSERT, UPDATE, DELETE ON PlayerDecks TO Player;
GRANT SELECT, INSERT, UPDATE, DELETE ON Decks TO Player;
GRANT SELECT, INSERT, UPDATE, DELETE ON DeckCards TO Player;
```

Developers have the ability to edit cards, heroes, expansions, games but not the things that players can edit.

```
--Developer
REVOKE ALL PRIVILEGES ON Players FROM Developer;
REVOKE ALL PRIVILEGES ON PlayerDecks FROM Developer;
REVOKE ALL PRIVILEGES ON Decks FROM Developer;
REVOKE ALL PRIVILEGES ON DeckCards FROM Developer;
GRANT SELECT, INSERT, UPDATE, DELETE ON DeckCards TO Developer;
GRANT SELECT, INSERT, UPDATE, DELETE ON Heros TO Developer;
GRANT SELECT, INSERT, UPDATE, DELETE ON Cards TO Developer;
GRANT SELECT, INSERT, UPDATE, DELETE ON Games TO Developer;
GRANT SELECT, INSERT, UPDATE, DELETE ON Minions TO Developer;
GRANT SELECT, INSERT, UPDATE, DELETE ON Weapons TO Developer;
GRANT SELECT, INSERT, UPDATE, DELETE ON Spells TO Developer;
GRANT SELECT, INSERT, UPDATE, DELETE ON Expansions TO Developer;
```

Admins have the ability to edit everything.

```
--Admin
GRANT SELECT, INSERT, UPDATE, DELETE ON DeckCards TO Developer;
GRANT SELECT, INSERT, UPDATE, DELETE ON Heros TO Developer;
GRANT SELECT, INSERT, UPDATE, DELETE ON Cards TO Developer;
GRANT SELECT, INSERT, UPDATE, DELETE ON Games TO Developer;
GRANT SELECT, INSERT, UPDATE, DELETE ON Minions TO Developer;
GRANT SELECT, INSERT, UPDATE, DELETE ON Weapons TO Developer;
GRANT SELECT, INSERT, UPDATE, DELETE ON Spells TO Developer;
GRANT SELECT, INSERT, UPDATE, DELETE ON Expansions TO Developer;
GRANT SELECT, INSERT, UPDATE, DELETE ON Players TO Player;
GRANT SELECT, INSERT, UPDATE, DELETE ON PlayerDecks TO Player;
GRANT SELECT, INSERT, UPDATE, DELETE ON Decks TO Player;
GRANT SELECT, INSERT, UPDATE, DELETE ON DeckCards TO Player;
```

## Future Enhancements

- Take into account the Rarity of a card to limit how many times a card can be input in to a deck
- Limit how many times a card can be into a deck
- Prevent a deck from being playable if the deck has less than 30 cards