Magic: The Gathering Database Documentation

John Kuroda
Akil Marshall
Israel Trusdell

March 27, 2020

Contents

1 Entity Relationship Diagram

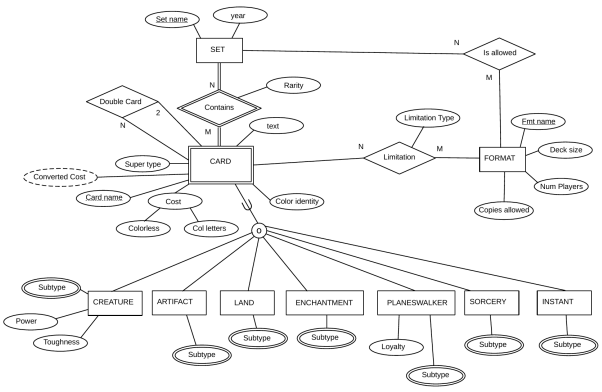

Figure 1: Entity relationship diagram.

2 Database Schema

3 Efforts of Normalization

We took our database to the third normal form.

3.1 First Normal Form

From the text book,

> First normal form (1NF)is now considered to be part of the formal definition of a relation in the basic (flat) relational model; historically, it was defined to disallow multivalued attributes, composite attributes, and their combinations. It states that the domain of an attribute must include only atomic (simple, indivisible) values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. Hence, 1NF disallows having a set of values, a tuple o values, or a combination of both as an attribute value for a single tuple. In other words, 1NF disallows relations within relations or relations as attribute values within tuples.

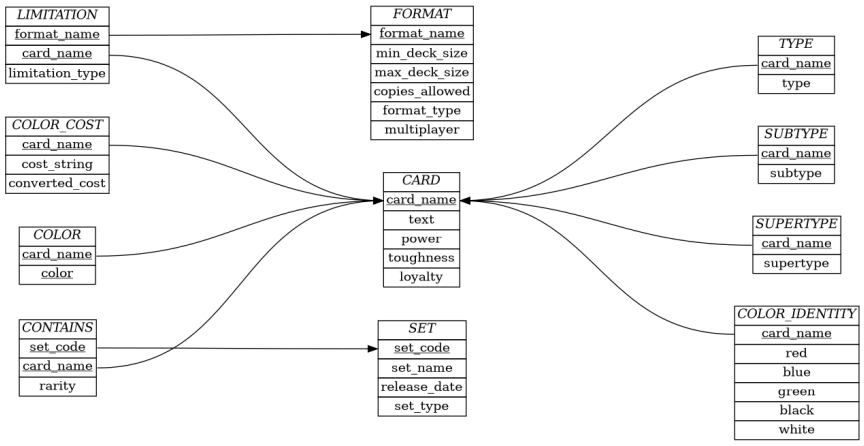

Figure 2: Database table schema

> The only attribute values permitted by 1NF are single atomic (or indivisible) values.

Our schema is in first normal form by design.

3.2 Second Normal Form

From the text book,

> **Definition.** A relation schema R is in 2NF if every non-prime attribute A in R is fully functionally dependent on the primary key of R.

We previously had included set_name in the primary key of CARD. This however violates the second normal form since the other attributes of CARD had no functional dependency on set_name. We then removed set_name from CARD to bring the schema to second normal form.

3.3 Third Normal Form

From the text book,

> **Definition.** According to Codd's original definition, a relation schema R is in 3NF if it satisfies 2NF and no non-prime attribute of R is transitively dependent on the primary key.

Previously we had included much of the color information in the CARD table however attributes like converted_cost were only transitively dependent on the primary key (card_name). To address this we created the tables COLOR,

COLOR_COST, and COLOR_IDENTITY, moving the appropriate attributes to the correct tables. With this each attribute depends directly on the primary key of it's table, thus our schema is in the third normal form.

4 Tables

4.1 CARD

A CARD represents a real, phyisical, entity in Magic: The Gathering (MTG). CARDs have attributes (descirbed below) that appear as printed images or text on the playing face of the CARD. These attributes give CARDs different playable characteristics in a game.

CARDs may or may not be grouped by these attributes. One example are all CARDs with they type creature, which represent all creature cards. Note that under these attributes, a CARD can be uniquely identified by its name, but a CARD may be reprinted in more than one SET. In other words, a CARD doesn't change between SETs and can be contained within one or more SETs.

4.1.1 Attributes

- card_name
  - **Description**: The name of the card.
  - **Data type**: String
  - **Domain**: Any valid card name.
  - **Is primary key**: Yes.
  - **Nullable**: No.
- text
  - **Description**: Everything in the text area of the card.
  - **Data type**: String
  - **Domain**: Any valid card text.
  - **Is primary key**: No.
  - **Nullable**: True
- power
  - **Description**: The card's power.
  - **Data type**: Integer
  - **Domain**: Any non-negative integer.
  - **Is primary key**: No.
  - **Nullable**: Yes.

- toughness
  - **Description**: The card's toughness.
  - **Data type**: Integer
  - **Domain**: Any non-negative integer.
  - **Is primary key**: No.
  - **Nullable**: Yes.
- loyalty
  - **Description**: The card's loyalty.
  - **Data type**: Integer
  - **Domain**: Any non-negative integer.
  - **Is primary key**: No.
  - **Nullable**: Yes.

4.2 SET

The MTG wiki had the following to say about what a set is.

> A set in Magic: The Gathering is a pool of cards released together and designed for the same play environment. Cards in a set can be obtained either randomly through booster packs, or in box sets that have a fixed selection of cards. An expansion symbol and, more recently, a three-character abbreviation is printed on each card to identify the set it belongs to.

4.2.1 Attributes

- set_code
  - **Description**: The alphanumeric code associated with a set.
  - **Data type**: String
  - **Domain**: Combinations of letters and digits.
  - **Is primary key**: Yes.
  - **Nullable**: No.
- set_name
  - **Description**: The name of the set.
  - **Data type**: String
  - **Domain**: Any valid set name.
  - **Is primary key**: No.

- **Nullable**: No

- release_date
  - **Description**: The date the set was released.
  - **Data type**: String
  - **Domain**: Any valid date.
  - **Is primary key**: No.
  - **Nullable**: No.

- set_type
  - **Description**: The type of set it is (core, expansion, etc).
  - **Data type**: String
  - **Domain**: Any valid set type.
  - **Is primary key**: No.
  - **Nullable**: No.

4.3 FORMAT

A SET represents a real, physical, collection of CARDs that are released together and designed for the same play environment. SETs are released throught the year and each have a name, code (three character abbreviation) and a set symbol which is not tracked.

4.3.1 Attributes

- format_name
  - **Description**: The name of the format.
  - **Data type**: String
  - **Domain**: Any valid format name.
  - **Is primary key**: Yes.
  - **Nullable**: No.

- min_deck_size
  - **Description**: The minimum number of cards allowed in a deck.
  - **Data type**: Integer
  - **Domain**: Any non-negative integer.
  - **Is primary key**: No.
  - **Nullable**: No.

- max_deck_size
  - **Description**: The maximum number of cards allowed in a deck.
  - **Data type**: Integer.
  - **Domain**: Any integer, negative integers are interpreted as infinity.
  - **Is primary key**: No.
  - **Nullable**: No.
- copies_allowed
  - **Description**: The maximum number of copies of a card allowed in a deck.
  - **Data type**: Integer
  - **Domain**: Any non-negative integer.
  - **Is primary key**: No.
  - **Nullable**: No.
- format_type
  - **Description**: The type of the format (constructed, draft, etc)
  - **Data type**: String
  - **Domain**: Valid magic format types.
  - **Is primary key**: No.
  - **Nullable**: No.
- multiplayer
  - **Description**: If the format can be played by more than 2 people.
  - **Data type**: Boolean
  - **Domain**: Any valid boolean.
  - **Is primary key**: No.
  - **Nullable**: No.

4.4 CONTAINS

This table is the implementation of the many-to-many relationship between CARD and SET. A card may be included in may sets and A set may contain many cards.

4.4.1 Attributes

- set_code
  - **Description**: A foreign key from SET.
  - **Data type**: String
  - **Domain**: Combinations of letters and digits.
  - **Is primary key**: Yes.
  - **Nullable**: No.
- card_name
  - **Description**: A foreign key from CARD.
  - **Data type**: String
  - **Domain**: Any valid card name.
  - **Is primary key**: Yes.
  - **Nullable**: No.
- rarity
  - **Description**: The rarity of the card (common, uncommon, etc).
  - **Data type**: String
  - **Domain**: Any valid magic card rarity.
  - **Is primary key**: No.
  - **Nullable**: No.

4.5 LIMITATION

This table is the implementation of the many-to-many relationship between FORMAT and CARD. A format may limit many cards and a card may be limited by many formats.

A LIMITATION represents a many-to-many relationship between CARD and FORMAT. The limitation or restriction of a CARD to a particular FORMAT is determined by the particular rules of the FORMAT. FORMATs can deem particular CARDs banned (not allowed in gameplay), restricted (only one copy of CARD is allowed in a deck), or illegal (CARD is not allowed).

4.5.1 Attributes

- format_name
  - **Description**: A foreign key from FORMAT.
  - **Data type**: String
  - **Domain**: Any valid format name.

  - **Is primary key**: Yes.
  - **Nullable**: No.

- card_name
  - **Description**: A foreign key from CARD.
  - **Data type**: String
  - **Domain**: Any valid card name.
  - **Is primary key**: Yes.
  - **Nullable**: No.

- limitation_type
  - **Description**: The way in which a card is limited (banned, restricted, etc).
  - **Data type**: String
  - **Domain**: Any valid limitation.
  - **Is primary key**: No.
  - **Nullable**: No.

4.6 COLOR

MTG wiki had the following to say about color.

> Color is a basic property of cards in Magic: The Gathering, forming the core of the game's mana system and overall strategy.

4.6.1 Attributes

- card_name
  - **Description**: A foreign key from CARD.
  - **Data type**: String
  - **Domain**: Any valid card name.
  - **Is primary key**: Yes.
  - **Nullable**: No.

- color
  - **Description**: The color a card is associated with, usually indicated by the physical color of the card.
  - **Data type**: String
  - **Domain**: Any valid magic card color.
  - **Is primary key**: No.
  - **Nullable**: No.

4.7 COLOR_COST

Due to the fact that converted_cost depends on cost_string which is not the primary key. This table solves that problem. COLOR_COST represents symobols on a CARD indicating the cost of casting the CARD. On a physical CARD, this can be a combination of intgers and symbols which represent color. Symbols are represented in this table with letters. R for red, U for blue, G for green, B for black, W for white. X represents casting costs with no required color.

4.7.1 Attributes

- card_name
  - **Description**: A foreign key from CARD.
  - **Data type**: String
  - **Domain**: Any valid card name.
  - **Is primary key**: Yes.
  - **Nullable**: No.
- cost_string
  - **Description**: An alphanumeric representation of a cards mana cost.
  - **Data type**: String
  - **Domain**: Strings over the alphabet $\sum = \{R, U, G, B, W, X, \phi\}$ where $\phi \in \mathbb{Z}_{>0}$ and each string that contains ϕ begins with ϕ.
  - **Is primary key**: No.
  - **Nullable**: Yes.
- converted_cost
  - **Description**: The sum over a cards mana cost. Each occurrence of

Table 1: How to sum a cost_string.

$\sum$	value
R	1
U	1
G	1
B	1
W	1
X	0
ϕ	ϕ

    a character in a cost_string is summed according to the above table.
  - **Data type**: Integer
  - **Domain**: Any non-negative integer.
  - **Is primary key**: No.
  - **Nullable**: No.

4.8 COLOR_IDENTITY

Each mana symbol that appears on a card is included within that cards color identity. Each card is associated with one or more colors.

4.8.1 Attributes

- card_name
  - **Description**: A foreign key from CARD.
  - **Data type**: String
  - **Domain**: Any valid card name.
  - **Is primary key**: Yes.
  - **Nullable**: No.
- red
  - **Description**: A flag to indicate the cards alignment with red.
  - **Data type**: Boolean
  - **Domain**: Any valid boolean.
  - **Is primary key**: No.
  - **Nullable**: No.
- blue
  - **Description**: A flag to indicate the cards alignment with blue.
  - **Data type**: Boolean
  - **Domain**: Any valid boolean.
  - **Is primary key**: No.
  - **Nullable**: No.
- green
  - **Description**: A flag to indicate the cards alignment with green.
  - **Data type**: Boolean
  - **Domain**: Any valid boolean.
  - **Is primary key**: No.
  - **Nullable**: No.
- white
  - **Description**: A flag to indicate the cards alignment with white.
  - **Data type**: Boolean
  - **Domain**: Any valid boolean.

  - **Is primary key**: No.
  - **Nullable**: No.

- black
  - **Description**: A flag to indicate the cards alignment with black.
  - **Data type**: Boolean
  - **Domain**: Any valid boolean.
  - **Is primary key**: No.
  - **Nullable**: No.

4.9 SUPERTYPE

Magic cards may have one or more supertypes, this table implements that one-to-many relationship.

4.9.1 Attributes

- card_name
  - **Description**: A foreign key from CARD.
  - **Data type**: String
  - **Domain**: Any valid card name.
  - **Is primary key**: Yes.
  - **Nullable**: No.

- supertype
  - **Description**: The supertype of the card (legendary, snow, etc).
  - **Data type**: String
  - **Domain**: Any valid magic card subtype.
  - **Is primary key**: No.
  - **Nullable**: No.

4.10 TYPE

Magic cards may have one or more types, this table implements that one-to-many relationship.

4.10.1 Attributes

- card_name
  - **Description**: A foreign key from CARD.
  - **Data type**: String
  - **Domain**: Any valid card name.
  - **Is primary key**: Yes.
  - **Nullable**: No.
- type
  - **Description**: The type of the card (creature, artifact, etc).
  - **Data type**: String
  - **Domain**: Any valid magic card type.
  - **Is primary key**: No.
  - **Nullable**: No.

4.11 SUBTYPE

Magic cards may have zero or more subtypes, this table implements that one-to-many relationship.

4.11.1 Attributes

- card_name
  - **Description**: A foreign key from CARD.
  - **Data type**: String
  - **Domain**: Any valid card name.
  - **Is primary key**: Yes.
  - **Nullable**: No.
- subtype
  - **Description**: The subtype of the card (equipment, curse, etc).
  - **Data type**: String
  - **Domain**: Any valid magic card subtype.
  - **Is primary key**: No.
  - **Nullable**: No.

5 Entity Constraints

In this section we describe the constraints on entities as far as insertion, deletion, and modification are concerned. However it must be understood that the users of this database will only be provided an interface that allows data to be read and not modified in any way. None the less we will have a discussion on how insertions, deletions, and modifications could be done with the proposed schema.

It should also be noted that because of the design of Magic: the Gathering some of the following operations could never occur.

5.1 Insertion

Card To add a new card entity you must ensure that the card_name is unique. However it is in the interest of the designer's of magic to ensure this is true thus you need not worry. In addition to the attributes in the CARD table each table that has CARD.card_name as a foreign key must also have one or more rows populated.

The following tables must be updated,

- LIMITATION
- COLOR_COST
- COLOR
- CONTAINS
- TYPE
- SUBTYPE
- SUPERTYPE
- COLOR_IDENTITY

Format To add a new format entity you must ensure that the format name is unique. However it is not very likely that a format will have the same name as a preexisting format. In addition to the attributes in the FORMAT table each table that has TABLE.format_name as a foreign key must also have one or more rows populated.

The following tables must be updated,

- LIMITATION

More specifically each card that is limited within the new format must be modified.

Set To add a new set entity you must ensure that the set name is unique. However the it is in the interest of the designers of magic that each set name is unique so this isn't a problem. In addition to the attributes within the SET table each table that has SET.set_name as a foreign key needs to be updated.

The following tables must be updated,

- CONTAINS

5.2 Deletion

Card Delete the entity within the CARD table and all tables that have the CARD.card_name as a foreign key.

The following tables must be checked for deletion,

- LIMITATION
- COLOR_COST
- COLOR
- CONTAINS
- TYPE
- SUBTYPE
- SUPERTYPE
- COLOR_IDENTITY

Format Delete the entity within the FORMAT table and all tables that have the FORMAT.format_name as a foreign key.

The following tables must be checked for deletion,

- LIMITATION

Set Delete the set entity within the SET table and all tables that have SET.set_name as a foreign key.

The following tables must be checked for deletion,

- CONTAINS

6 Domain Descriptions of Certain Attributes

In this section we either completely list the entire domain or provide examples and further description.

6.1 SET.set_type

- archenemy
- box
- commander
- core
- draft_innovation
- duel_deck
- expansion
- from_the_vault
- funny
- masterpiece
- masters
- memorabilia
- planechase
- premium_deck
- promo
- spellbook
- starter
- token
- treasure_chest

6.2 CONTAINS.rarity

- common
- rare
- uncommon
- mythic

6.3 COLOR.color

- black
- colorless
- red
- blue
- green
- white

6.4 LIMITATION.limitation_type

- none
- restricted
- banned

6.5 COLOR.color

- red
- blue
- black
- green
- white
- colorless

6.6 COLOR_COST.cost_string

We provide several examples of cost_strings.

Table 2: Cards and their cost_string.

Card Name	cost_string
Angrath, Captain of Chaos	2B/R/R
Conclave Naturalists	4G
Jace, Memory Adept	3UU
Ajani Vengeant	2RW
Chromanticore	WUBRG

6.7 SUPERTYPE.supertype

- Basic
- Legendary
- World
- Host
- Snow

6.8 TYPE.type

- artifact
- creature
- enchantment
- land
- planeswalker
- instant
- sorcery

6.9 SUBTYPE.subtype

- Clue
- Contraption
- Equipment
- Food
- Fortification
- Gold
- Key
- Treasure
- Vehicle
- Advisor
- Aetherborn
- Ally
- Angel
- Antelope
- Ape
- Archer
- Archon
- Army
- Artificer
- Assassin
- Assembly-Worker
- Atog
- Aurochs
- Avatar
- Azra
- Badger
- Barbarian
- Basilisk
- Bat
- Bear
- Beast
- Beeble
- Berserker
- Bird
- Blinkmoth
- Boar
- Bringer
- Brushwagg
- Camarid
- Camel
- Caribou
- Carrier
- Cat
- Centaur
- Cephalid
- Chimera
- Citizen
- Cleric
- Cockatrice
- Construct
- Coward
- Crab
- Crocodile
- Cyclops
- Dauthi
- Demigod
- Demon
- Deserter
- Devil
- Dinosaur
- Djinn
- Dragon
- Drake
- Dreadnought
- Drone
- Druid
- Dryad
- Dwarf
- Efreet

- Egg
- Elder
- Eldrazi
- Elemental
- Elephant
- Elf
- Elk
- Eye
- Faerie
- Ferret
- Fish
- Flagbearer
- Fox
- Frog
- Fungus
- Gargoyle
- Germ
- Giant
- Gnome
- Goat
- Goblin
- God
- Golem
- Gorgon
- Graveborn
- Gremlin
- Griffin
- Hag
- Harpy
- Hellion
- Hippo
- Hippogriff
- Homarid
- Homunculus
- Horror
- Horse
- Hound
- Human
- Hydra
- Hyena
- Illusion
- Imp
- Incarnation
- Insect
- Jackal
- Jellyfish
- Juggernaut
- Kavu
- Kirin
- Kithkin
- Knight
- Kobold
- Kor
- Kraken
- Lamia
- Lammasu
- Leech
- Leviathan
- Lhurgoyf
- Licid
- Lizard
- Manticore
- Masticore
- Mercenary
- Merfolk
- Metathran
- Minion
- Minotaur
- Mole
- Monger
- Mongoose
- Monk
- Monkey
- Moonfolk
- Mouse
- Mutant
- Myr
- Mystic
- Naga
- Nautilus
- Nephilim
- Nightmare
- Nightstalker
- Ninja
- Noble
- Noggle

- Nomad
- Nymph
- Octopus
- Ogre
- Ooze
- Orb
- Orc
- Orgg
- Ouphe
- Ox
- Oyster
- Pangolin
- Peasant
- Pegasus
- Pentavite
- Pest
- Phelddagrif
- Phoenix
- Pilot
- Pincher
- Pirate
- Plant
- Praetor
- Prism
- Processor
- Rabbit
- Rat
- Rebel
- Reflection
- Rhino
- Rigger
- Rogue
- Sable
- Salamander
- Samurai
- Sand
- Saproling
- Satyr
- Scarecrow
- Scion
- Scorpion
- Scout
- Sculpture
- Serf
- Serpent
- Servo
- Shade
- Shaman
- Shapeshifter
- Sheep
- Siren
- Skeleton
- Slith
- Sliver
- Slug
- Snake
- Soldier
- Soltari
- Spawn
- Specter
- Spellshaper
- Sphinx
- Spider
- Spike
- Spirit
- Splinter
- Sponge
- Squid
- Squirrel
- Starfish
- Surrakar
- Survivor
- Tentacle
- Tetravite
- Thalakos
- Thopter
- Thrull
- Treefolk
- Trilobite
- Triskelavite
- Troll
- Turtle
- Unicorn
- Vampire
- Vedalken
- Viashino

- Volver
- Wall
- Warlock
- Warrior
- Weird
- Werewolf
- Whale
- Wizard
- Wolf
- Wolverine
- Wombat
- Worm
- Wraith
- Wurm
- Yeti
- Zombie
- Zubera
- Aura
- Carouche
- Curse
- Saga
- Shrine
- Plains
- Island
- Swamp
- Mountain
- Forest
- Desert
- Gate
- Lair
- Locus
- Urza's
- Mine
- Power-Plant
- Tower
- Ajani
- Aminatou
- Angrath
- Arlinn
- Ashiok
- Bolas
- Chandra
- Dack
- Daretti
- Davriel
- Domri
- Dovin
- Elspeth
- Estrid
- Freyalise
- Garruk
- Gideon
- Huatli
- Jace
- Jaya
- Karn
- Kasmina
- Kaya
- Kiora
- Koth
- Liliana
- Nahiri
- Narset
- Nissa
- Nixilis
- Oko
- Ral
- Rowan
- Saheeli
- Samut
- Sarkhan
- Serra
- Sorin
- Tamiyo
- Teferi
- Teyo
- Tezzeret
- Tibalt
- Ugin
- Venser
- Vivien
- Vraska
- Will
- Windgrace
- Wrenn
- Xenagos
- Yanggu
- Yanling
- Adventure
- Arcane
- Trap
- Adventure
- Arcane