**MTGData Base**

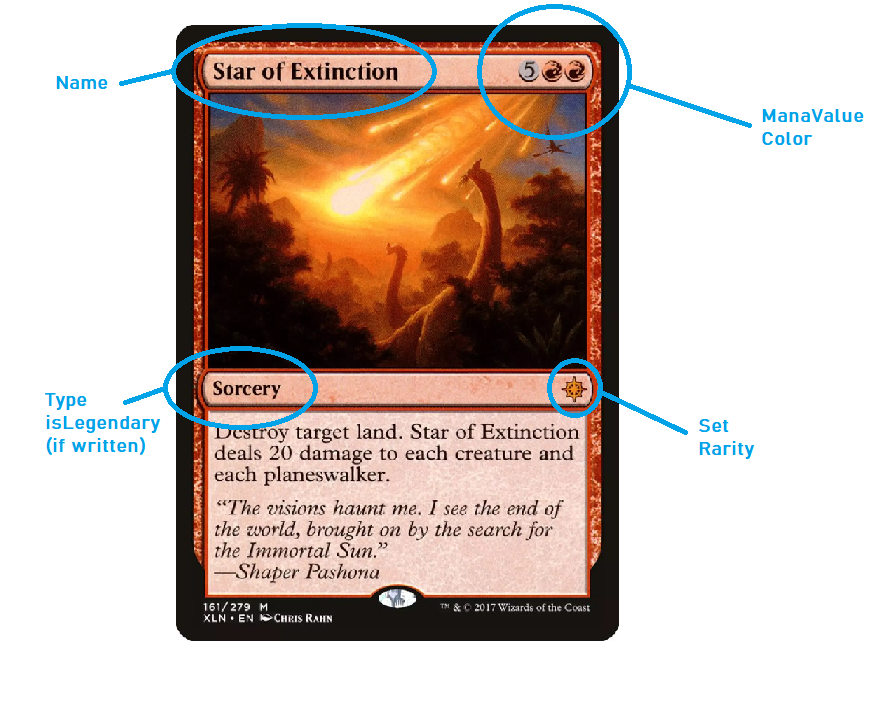
Итак, наш выбор предметной области для выполнения лабораторной работы был довольно специфическим. Мы любим настольные игры и нас вдохновил пример с базой для карточек из Гвинта. Мы решили сделать базу данных для карточек из известнейшей коллекционной карточной игры Magic the Gathering (сокращенно MTG).

Почитать подробнее можно здесь <https://ru.wikipedia.org/wiki/Magic:_The_Gathering>

Официальный сайт

<https://magic.wizards.com/ru>

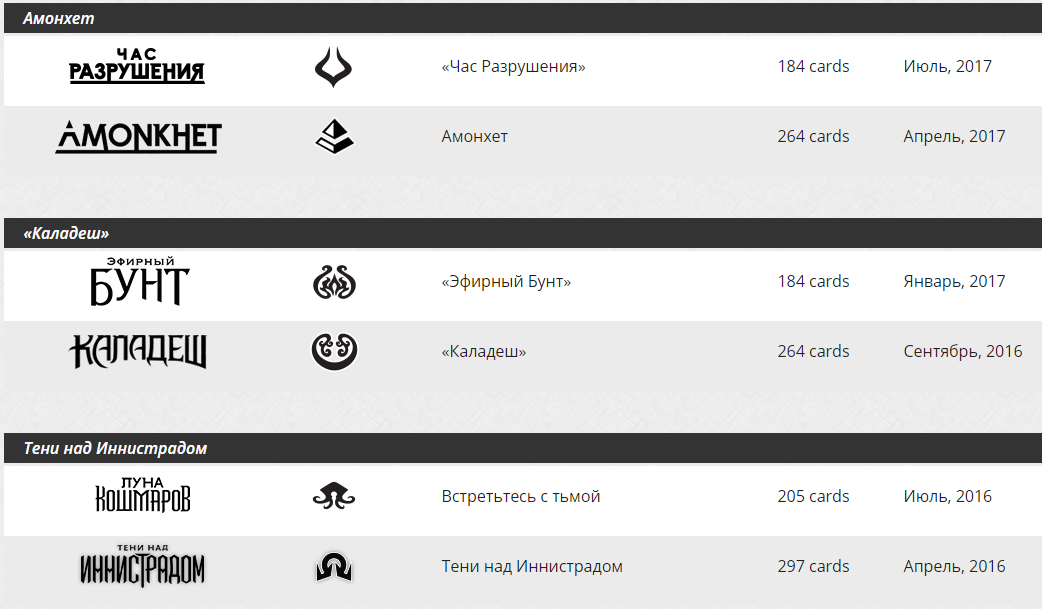
Каждая карточка MTG обладает определенным набором параметров, от ее названия, цветовой принадлежности до имени художника, нарисовавшего арт к ней. Мы решили использовать самые важные из них.



Параметры карточки:

1. Имя
2. Цвет (определяется значком в манастоимости карты)
3. Мановая ценность (определяется как цифра в сером кружке + количество значков цвета)
4. Тип (может быть существо, волшебство, мгновенное заклинание, чары, артефакт, мироходец)
5. Сет (или «выпуск», каждая карточка привязана к сету, значок которого изображен в ее левой стороне, у сета тоже есть свои параметры)
6. Редкость (определяется цветом значка сета)
7. Легендарность (обычно указывается после типа карты)

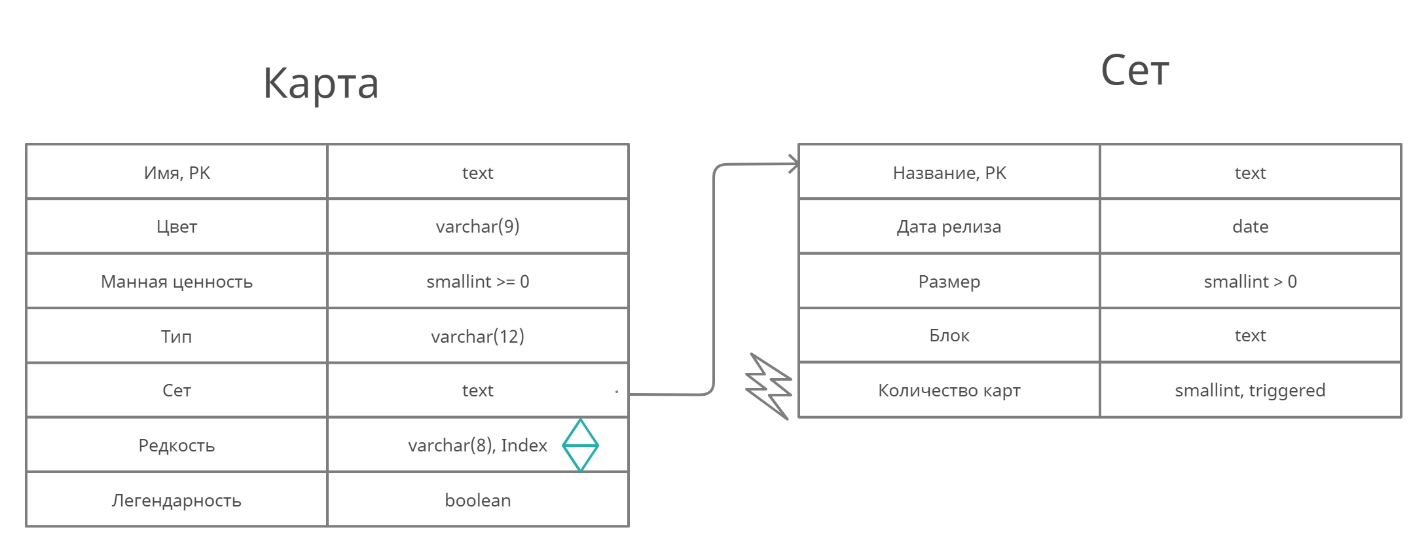
Информация о каждой карте лежит у нас в таблице «Cards»



Параметры сета:

1. Название
2. Дата релиза
3. Размер (общее количество уникальных карт выпущенных в этом сете)
4. Блок (сеты обьединяются в блоки, иногда по два, иногда по три, часто название блока эквивалентно названию сета)
5. Количество карт из данного сета в нашей базе (заполняется триггером)

Диаграмма нашей базы данных:



И вот сами запросы создания таблиц и бд:

-- Database: mtg

-- DROP DATABASE mtg;

CREATE DATABASE mtg

WITH

OWNER = "Pasha"

ENCODING = 'UTF8'

LC\_COLLATE = 'Russian\_Russia.1251'

LC\_CTYPE = 'Russian\_Russia.1251'

TABLESPACE = pg\_default

CONNECTION LIMIT = -1;

-- Table: public."Cards"

-- DROP TABLE public."Cards";

CREATE TABLE public."Cards"

(

"Name" text COLLATE pg\_catalog."default" NOT NULL,

"Color" character varying(9) COLLATE pg\_catalog."default" NOT NULL,

"ManaValue" smallint NOT NULL,

"Type" character varying(12) COLLATE pg\_catalog."default" NOT NULL,

"Set" text COLLATE pg\_catalog."default" NOT NULL,

"Rarity" character varying(8) COLLATE pg\_catalog."default" NOT NULL,

"isLegendary" boolean NOT NULL DEFAULT false,

CONSTRAINT "Cards\_pkey" PRIMARY KEY ("Name"),

CONSTRAINT "SetKey" FOREIGN KEY ("Set")

REFERENCES public."Sets" ("Name") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT "Cards\_ManaValue\_check" CHECK ("ManaValue" >= 0)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public."Cards"

OWNER to Pasha;

-- Index: cardsrarityindex

-- DROP INDEX public.cardsrarityindex;

CREATE INDEX cardsrarityindex

ON public."Cards" USING btree

("Rarity" COLLATE pg\_catalog."default" ASC NULLS LAST)

TABLESPACE pg\_default;

-- Trigger: dec\_cards\_count

-- DROP TRIGGER dec\_cards\_count ON public."Cards";

CREATE TRIGGER dec\_cards\_count

AFTER DELETE

ON public."Cards"

FOR EACH ROW

EXECUTE PROCEDURE public.dec\_cards\_count();

-- Trigger: inc\_cards\_count

-- DROP TRIGGER inc\_cards\_count ON public."Cards";

CREATE TRIGGER inc\_cards\_count

AFTER INSERT

ON public."Cards"

FOR EACH ROW

EXECUTE PROCEDURE public.inc\_cards\_count();

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-- Table: public."Sets"

-- DROP TABLE public."Sets";

CREATE TABLE public."Sets"

(

"Name" text COLLATE pg\_catalog."default" NOT NULL,

"ReleaseDate" date NOT NULL,

"Size" smallint NOT NULL,

"Block" text COLLATE pg\_catalog."default" NOT NULL,

"CountCards" smallint DEFAULT 0,

CONSTRAINT "Sets\_pkey" PRIMARY KEY ("Name"),

CONSTRAINT "Sets\_Size\_check" CHECK ("Size" > 0)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public."Sets"

OWNER to Pasha;

Обоснование нормальных форм:

1 Нф — строки таблицы

2 НФ — все атрибуты являются характеристиками карточки. Название карточки — уникальный ее идентификатор. Таким образом все атрибуты строго зависят от primary key.

3 НФ — данная зависимость не транзитивна так как мы не можем предсказать значение атрибута по значению другого не ключевого атрибута. (опять же они не связаны между собой, атрибуты карты и сета являются просто характеристиками и не зависят друг от друга) Например, по количеству карт в сете мы не можем определить его дату релиза и т.д.

Таким образом, наша бд находится в третьей нормальной форме.

Все подключения к БД осуществляем через выделенного пользователя, Pasha или Grisha соответственно. Вот пример роли:  
-- Role: "Pasha"

-- DROP ROLE "Pasha";

CREATE ROLE "Pasha" WITH

LOGIN

NOSUPERUSER

INHERIT

CREATEDB

CREATEROLE

REPLICATION

Соответственно, задаем все права, кроме суперпользователя, т.к. иначе в отделении от postgres нет смысла:)  
Реализовали некоторые функции вне задания, чтобы сделать GUI более интересным, такие как получение списка всех баз данных, получение списка таблиц у конкретной базы данных. На будущее есть перспективы доработать слабые места в приложении, когда появится свободное время