**РЕФЕРАТ**

Работа содержит 38 страниц. В работе использованы 23 иллюстрация, 7 таблиц, 1 приложение, 10 источников литературы.

Перечень ключевых слов: концептуальное проектирование базы данных, логическое проектирование базы данных, физическое проектирование базы данных, ER-модель данных, реляционная модель данных, нормализация отношений, схема базы данных, SQLite, приложение базы данных, Python 3

Результатом выполнения данной работы является приложение для работы пользователей с базой данных «Киберспортивные соревнования». При проектировании базы данных использован ER-подход к проектированию реляционных баз данных. Интерфейс приложения содержит: форму для подачи заявок на участие, форма входа в личный кабинет для отслеживания прогресса обращения, панель администратора для модерации обращения, форма для оформления компании в качестве частного исполнителя заявленных пользователем услуг.

Приложение можно использовать в различных организациях, организующих киберспортивные соревнования и похожие мероприятия.

**ABSTRACT**

The work contains 38 pages. The work used 23 illustrations, 7 tables, 1 appendix, 10 literature sources.

List of keywords: conceptual database design, logical database design, physical database design, ER data model, relational data model, relationship normalization, database schema, SQLite, database application, Python 3

The result of this work is an application for users to work with the “E-Sports Competitions” database. When designing the database, the ER approach to the design of relational databases was used. The application interface contains: a form for submitting applications for participation,

a form for logging into your personal account to track the progress of your request, an admin panel for moderating your request, a form for registering a company as a private provider of the services requested by the user.

The application can be used in various organizations organizing eSports competitions and similar events.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc135227395)

[1 Проектирование базы данных 5](#_Toc135227396)

[1.1 Концептуальное проектирование базы данных 5](#_Toc135227397)

[1.1.1 Исследование предметной области (Неформальное описание предметной области) 5](#_Toc135227398)

[1.1.2 Бизнес-правила 7](#_Toc135227399)

[1.1.3 Построение ER-модели данных 8](#_Toc135227400)

[1.2 Нормализация ER-модели данных 12](#_Toc135227401)

[2 Создание базы данных 18](#_Toc135227402)

[2.1 Схема базы данных SQLite 18](#_Toc135227403)

[3. Разработка приложения для работы с базой данных 25](#_Toc135227404)

[ЗАКЛЮЧЕНИЕ 33](#_Toc135227405)

[CПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 34](#_Toc135227406)

[ПРИЛОЖЕНИЕ A 35](#_Toc135227407)

# ВВЕДЕНИЕ

Целью данной работы является проектирование базы данных организации киберспортивных соревнований и разработка приложения для компании, которая занимается им.

В основные функции компании по организации киберспортивных соревнований входит: проведение соревнований, регистрация команд и участников, организация места для проведения, сбор средств от спонсоров и выдача призового фонда победителям. Для выполнения всех этих функций компании необходима база данных с информацией и приложение, чтобы эту информацию получить и отредактировать.

Основными задачами при проектировании и разработке базы данных и приложения являются:

- исследование предметной области ;

- проектирование базы данных;

- Создание базы данных;

- заполнение базы данных информацией;

- разработка интерфейса;

- реализация приложения.

1. Проектирование базы данных
   1. Концептуальное проектирование базы данных
      1. Исследование предметной области (Неформальное описание предметной области)

**Сценарий проекта**

База данных разрабатывается для компании по проведению киберспортивных соревнований. Основная функция компании – проведение соревнований, направленных на создание спортивных состязаний между командами на основе компьютерных видеоигр. Соревнования имеют уникальные названия с определенным местом и временем проведения. Соревнования проходят в играх, на которые подали заявки команды, финансируемые их спонсором..

С данной информационной системой должны работать следующие группы пользователей:

1. Администраторы;
2. Клиенты.

При работе с системой администраторы должны иметь возможность решать следующие задачи:

1. Создавать соревнования и изменять их состояние;
2. Регистрировать команды на соревнования;
3. Смотреть и изменять информацию об игроках и их участии;
4. Назначать площадки для проведения соревнований;
5. Выбирать и назначать игры для команд.
6. Регистрировать взносы от спонсоров и назначать выигрыш после игры.

При работе с системой игроки должны иметь возможность решать следующие задачи:

1. Смотреть и изменять свою личную информацию;
2. Выбирать нужные им дисциплины и свой никнейм;
3. Смотреть за состоянием формирования соревнования.

Для решения описанных задач в БД должна храниться следующая информация.

Каждое соревнование характеризуется следующими параметрами:

1. Уникальный номер соревнования;
2. Название соревнования;
3. Описание соревнования;
4. Команды-участники;
5. Сумма призового фонда.

Каждая площадка характеризуется следующими параметрами:

1. Уникальное название площадки;
2. Адрес площадки;
3. Количество компьютеров на площадке;
4. Характеристики компьютеров на площадке;
5. Контактный номер управляющего площадки.

Каждая игра характеризуется следующими параметрами:

1. Уникальное название игры;
2. Ограничение на количество игроков;
3. Системные требования игры;
4. Жанр игры.

Каждый игрок характеризуется следующими параметрами:

1. Уникальный номер игрока;
2. ФИО игрока;
3. Никнейм игрока;
4. Игры в которых игрок желает принять участие.

Каждая команда характеризуется следующими параметрами:

1. Уникальное название команды;
2. Дата создания команды.

Каждый спонсор характеризуется следующими параметрами:

1. Уникальное название компании;
2. ФИО спонсора;
3. Контактный номер спонсора;
4. Сумма спонсирования.
   * 1. Бизнес-правила

На основе анализа неформального описания предметной области были сформулированы бизнес-правила, представленные в таблице 1.1.

Таблица 1.1 – Бизнес-правила

|  |  |
| --- | --- |
| Бизнес-правило | Ограничение |
| В каждом соревновании должно участвовать несколько команд. | Количество команд должно быть две и больше. |
| На площадке может быть несколько компьютеров. | Количество компьютеров на площадке должно быть два или больше. |
| Характеристики компьютеров на площадке могут различаться. | Характеристики компьютеров должны соответствовать или быть больше системных требований. |
| В команде может быть несколько игроков. | Количество игроков в команде должно быть один и больше. |

* + 1. Построение ER-модели данных

На основе анализа неформального описания предметной области были определены наборы объектов и связей.

**Определение объектов**

Потенциальные объекты и атрибуты, включая необязательные

* СОРЕВНОВАНИЕ

\* уникальный номер соревнования

\* название соревнования

\* описание соревнования

\* команда 1

\* команда 2

\* призовой фонд

* ПЛОЩАДКА

\* уникальное название площадки

\* адрес

\* количество компьютеров

\* характеристики компьютеров

\* контактный номер управляющего

* ИГРА

\* уникальное название игры

\* ограничение на количество игроков

\* системные требования

\* жанр игры

* ИГРОК

\* уникальный номер игрока

\* ФИО игрока

\* никнейм

\* игра 1

\* игра 2

* КОМАНДА

\* уникальное название команды

\* дата создания команды

* СПОНСОР

\* уникальное название компании

\* ФИО спонсора

\* контактный номер спонсора

\* сумма спонсирования

**Определение связей**

Слева направо: каждое соревнование (СОРЕВНОВАНИЕ) должно иметь одну площадку (ПЛОЩАДКА)

Справа налево: каждая площадка (ПЛОЩАДКА) может иметь одно или несколько соревнований (СОРЕВНОВАНИЕ)

Слева направо: в каждой площадке (ПЛОЩАДКА) должна иметь одну или несколько игр (ИГРА)

Справа налево: каждая игра (ИГРА) может иметь одну или больше площадок (ПЛОЩАДКА)

Слева направо: в каждой игре (ИГРА) может играть один или несколько игроков (ИГРОК)

Справа налево: каждый игрок (ИГРОК) должен выбрать одну или более игр (ИГРА)

Слева направо: каждый игрок (ИГРОК) должен входить в одну команду (КОМАНДА)

Справа налево: каждая команда (КОМАНДА) должна входить в одну или несколько соревнований (СОРЕВНОВАНИЕ)

Слева направо: каждое (СОРЕВНОВАНИЕ) должно иметь две и более команд (КОМАНДА)

Справа налево: каждое соревнование (СОРЕВНОВАНИЕ) может иметь одного или больше спонсоров (СПОНСОР)

**Определение кардинальности связей**

Каждое соревнование (СОРЕВНОВАНИЕ) должено иметь **одну** площадку (ПЛОЩАДКА)

Каждый игрок (ИГРОК) должен иметь **одну или несколько** игр (ИГРА)

Каждая команда (КОМАНДА) должна иметь **одного или нескольких** игроков (ИГРОК)

Каждое соревнование (СОРЕВНОВАНИЕ) должно иметь **две или несколько**  команд (КОМАНДА)

Каждый игрок (ИГРОК) должен иметь **одну** команду (КОМАНДА)

На рисунке 1.1 представлена ER- диаграмма, построенная средствами программы Lucid chart

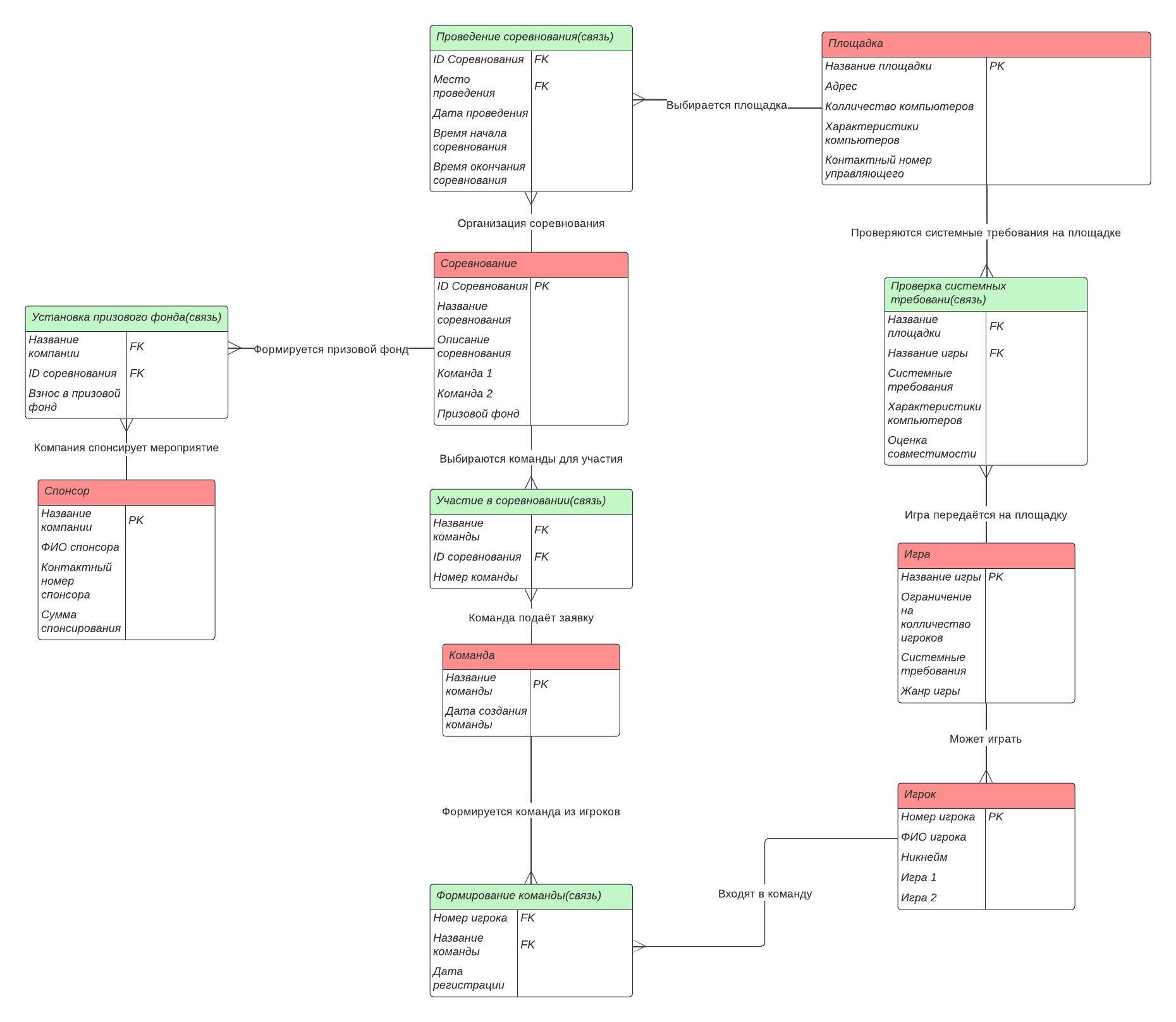


Рисунок 1.1 - ER- диаграмма

**Логическое проектирование базы данных**

* 1. Нормализация ER-модели данных

Все объекты построенной ER-модели данных должны находиться в третьей нормальной форме.

**Первая нормальная форма (1NF)**

Все атрибуты объекта СОРЕВНОВАНИЕ имеют только одно значение.

Все атрибуты объекта ПЛОЩАДКА имеют только одно значение.

Все атрибуты объекта ИГРА имеют только одно значение.

Все атрибуты объекта ИГРОК имеют только одно значение.

Все атрибуты объекта КОМАНДА имеют только одно значение.

Все атрибуты объекта СПОНСОР имеют только одно значение.

**Вторая нормальная форма (2NF)**

Атрибуты объекта СОРЕВНОВАНИЕ зависят от полного UID (уникальный номер соревнования).

Атрибуты объекта ПЛОЩАДКА зависят от полного UID (уникальное название площадки).

Атрибуты объекта ИГРА зависят от полного UID (уникальное название игры).

Атрибуты объекта ИГРОК зависят от полного UID (уникальный номер игрока).

Атрибуты объекта КОМАНДА зависят от полного UID (уникальное название команды).

Атрибуты объекта СПОНСОР зависят от полного UID (уникальное название компании).

**Третья нормальная форма (3NF)**

При проверке всех объектов не было выявлено независимых от UID атрибутов или атрибутов, зависимых от других атрибутов.

Названия таблиц и названия столбцов каждой таблицы приведены в соответствие с соглашениями об именах SQLite.

Таблица 1.2 – Игра

|  |  |  |  |
| --- | --- | --- | --- |
| Имя таблицы | | | |
| Game | | | |
| Key Type | Optionality | Column name | Data type |
| pk | \* | GameName | TEXT |
|  | \* | MaxPlayers | INTEGER |
|  | \* | SystemParams | TEXT |
|  | \* | GameGenre | TEXT |

Таблица 1.3 – Игрок

|  |  |  |  |
| --- | --- | --- | --- |
| Имя таблицы | | | |
| Player | | | |
| Key Type | Optionality | Column name | Data type |
| pk | \* | PlayerID | INTEGER |
|  | \* | PlayerFIO | TEXT |
|  | \* | Nickname | TEXT |
|  | \* | Game 1 | INTEGER |
|  |  | Game 2 | INTEGER |

Таблица 1.4 – Команда

|  |  |  |  |
| --- | --- | --- | --- |
| Имя таблицы | | | |
| Team | | | |
| Key Type | Optionality | Column name | Data type |
| pk | \* | TeamName | TEXT |
|  | \* | RegistrationDate | NUMERIC |

Таблица 1.5 – Спонсор

|  |  |  |  |
| --- | --- | --- | --- |
| Имя таблицы | | | |
| Sponsor | | | |
| Key Type | Optionality | Column name | Data type |
| pk | \* | CompanyName | TEXT |
|  | \* | SponsorFIO | TEXT |
|  | \* | SponsorNum | NUMERIC |
|  | \* | SponsorContribution | INTEGER |

Таблица 1.6 – Площадка

|  |  |  |  |
| --- | --- | --- | --- |
| Имя таблицы | | | |
| Platform | | | |
| Key Type | Optionality | Column name | Data type |
| pk | \* | PlatformName | TEXT |
|  | \* | Adress | TEXT |
|  | \* | CompNums | INTEGER |
|  | \* | CompParams | TEXT |
|  | \* | ContactNum | NUMERIC |

Таблица 1.7 – Соревнование

|  |  |  |  |
| --- | --- | --- | --- |
| Имя таблицы | | | |
| Competition | | | |
| Key Type | Optionality | Column name | Data type |
| pk | \* | ID | INTEGER |
|  | \* | CompName | TEXT |
|  | \* | CompDescr | TEXT |
|  | \* | Team 1 | TEXT |
|  | \* | Team 2 | TEXT |
|  | \* | MoneyForWin | INTEGER |

* 1. **Физическое проектирование базы данных**

Средствами программы DB Browser for SQLite на основе разработанной реляционной модели данных был сгенерирован скрипт на языке SQL для создания схемы реляционной базы данных

**Текст скрипта:**

CREATE TABLE "Competition" (

"ID" INTEGER NOT NULL UNIQUE,

"CompName" TEXT NOT NULL,

"CompDescr" TEXT NOT NULL,

"Team1" TEXT NOT NULL,

"Team2" TEXT NOT NULL,

"MoneyForWin" INTEGER,

PRIMARY KEY("ID")

) CREATE TABLE "Game" (

"GameName" TEXT NOT NULL UNIQUE,

"MaxPlayers" INTEGER NOT NULL,

"SystemParams" TEXT NOT NULL,

"GameGenre" TEXT NOT NULL,

PRIMARY KEY("GameName")

) CREATE TABLE "Player" (

"PlayerID" INTEGER NOT NULL UNIQUE,

"PlayerFIO" TEXT NOT NULL,

"Nickname" TEXT NOT NULL,

"Game1" INTEGER NOT NULL,

"Game2" INTEGER NOT NULL,

PRIMARY KEY("PlayerID")

) CREATE TABLE "Sponsor" (

"CompanyName" TEXT NOT NULL,

"SponsorFIO" TEXT NOT NULL,

"SponsorNum" NUMERIC,

"SponsorContribution" INTEGER NOT NULL,

PRIMARY KEY("CompanyName")

) CREATE TABLE "Team" (

"TeamName" TEXT NOT NULL UNIQUE,

"RegistrationDate" NUMERIC NOT NULL,

PRIMARY KEY("TeamName")

) CREATE TABLE "platform" (

"PlatformName" TEXT NOT NULL UNIQUE,

"adress" TEXT NOT NULL,

"CompNums" INTEGER NOT NULL,

"CompParams" TEXT,

"ContactNum" NUMERIC NOT NULL,

PRIMARY KEY("PlatformName")

)

1. Создание базы данных
   1. Схема базы данных SQLite

На рисунке 2.1 представлен перечень таблиц результат выполнения скрипта для создания схемы БД в DB Browser.

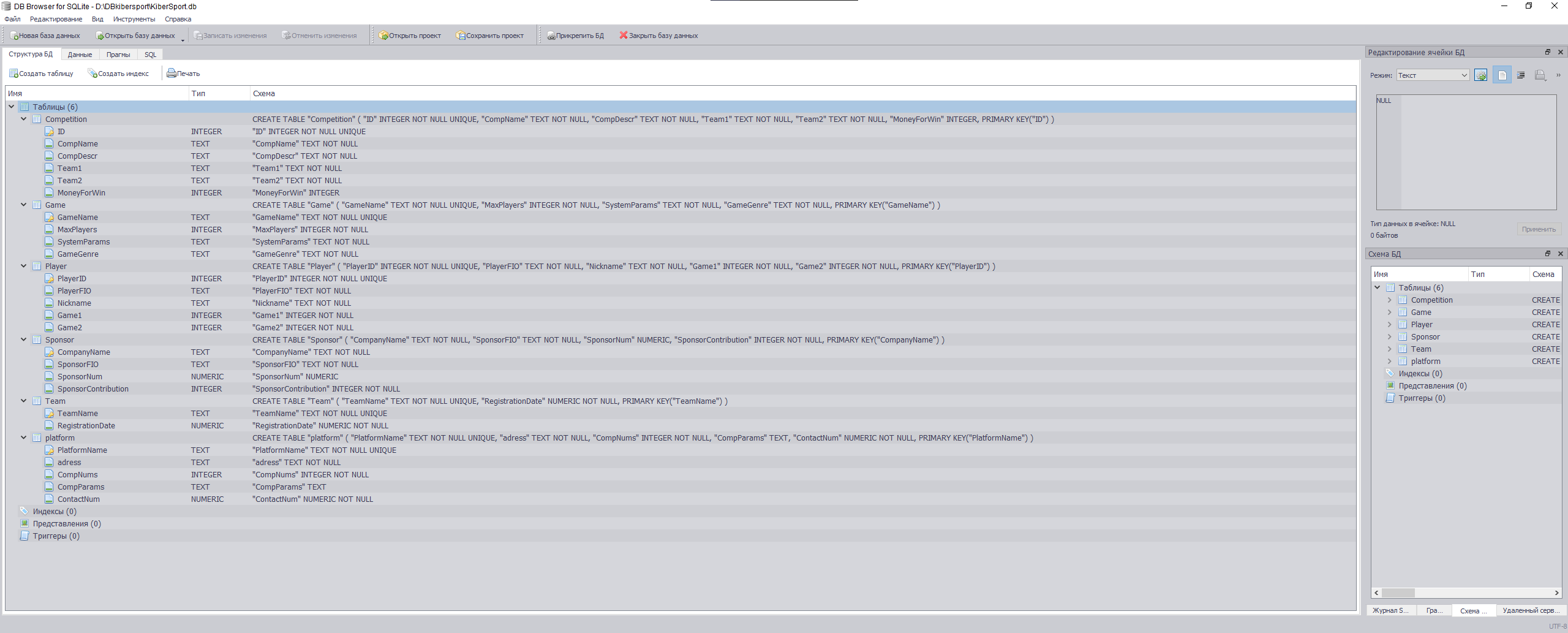


Рисунок 2.1 – Список таблиц базы данных

На рисунках 2.2 – 2.7 изображены схемы таблиц.

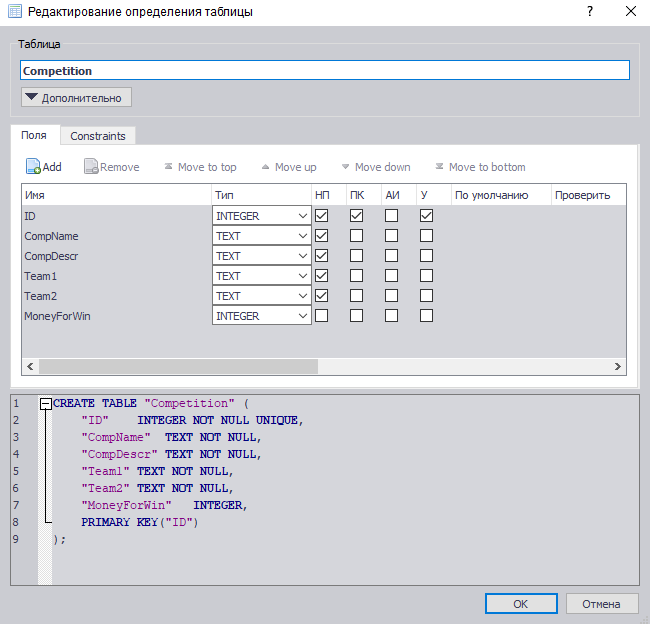


Рисунок 2.2 – Схема таблицы COMPETITION

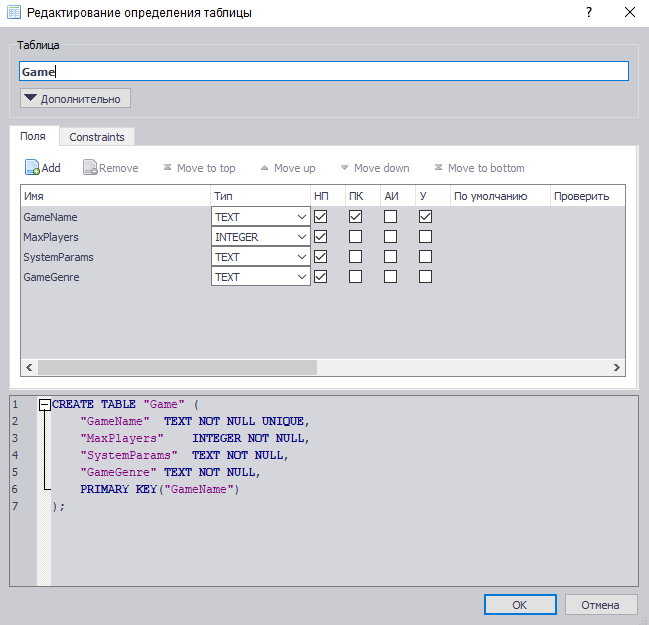


Рисунок2.3 – Схема таблицы GAME

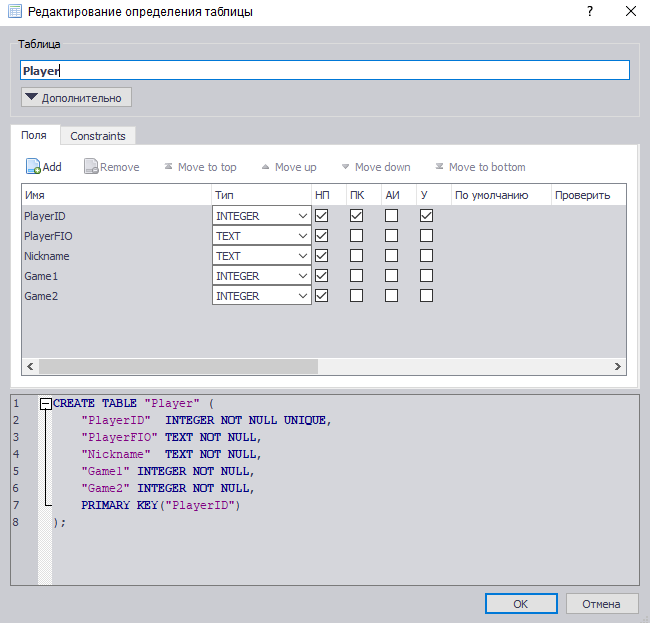


Рисунок2.4 – Схема таблицы PLAYER

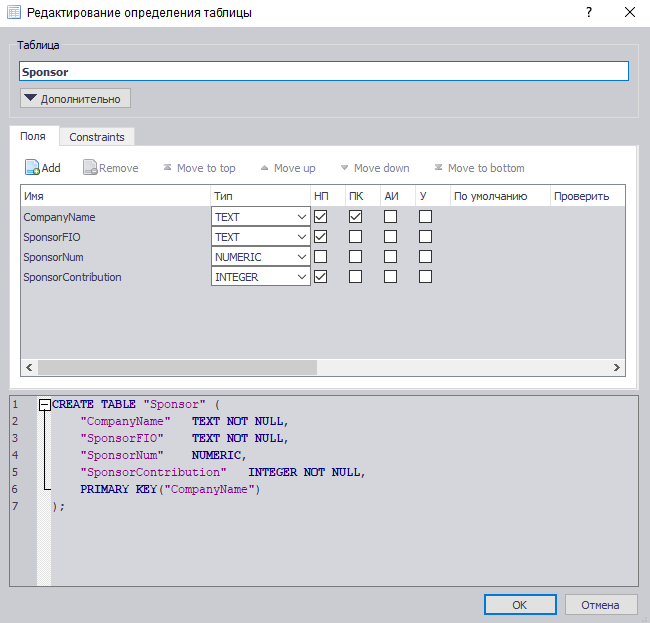


Рисунок 2.5– Схема таблицы SPONSOR

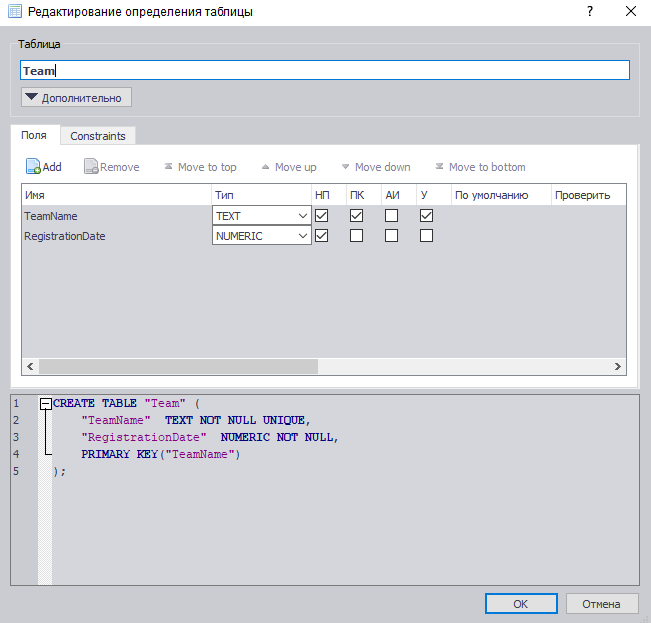


Рисунок 2.6 – Схема таблицы TEAM

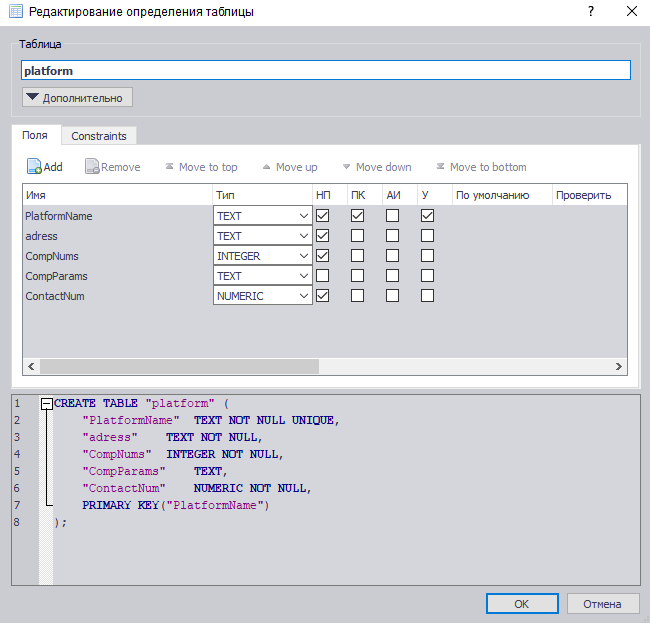


Рисунок 2.7 – Схема таблицы PLATFORM

**3. Разработка приложения для работы с базой данных**

Для работы пользователей с созданной базой данных было разработано приложение. Для разработки приложения использован язык Python. Интерфейс пользователя реализован средствами библиотеки TKinter.

Приложение представляет собой набор взаимосвязянных окон. Ссылочная целостность обеспечивается за счет того, что пользователь выбирает связанные обьекты из списков, которые формируются из данных о существующих в базе обьектах.

Многооконный интерфейс пользователя приложения показан на рис 3.1-3.15

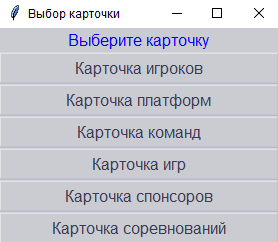


Рисунок 3.1 – Окно выбора

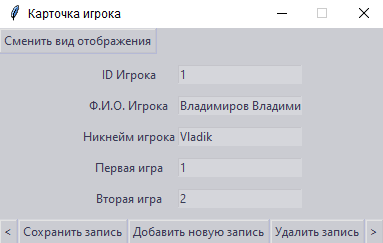


Рисунок 3.2 – Окно карточки игрока

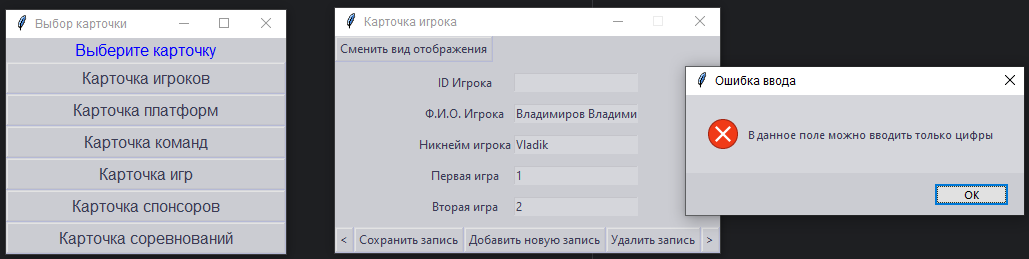


Рисунок 3.3 – Окно сообщения об ошибке на карточке игрока

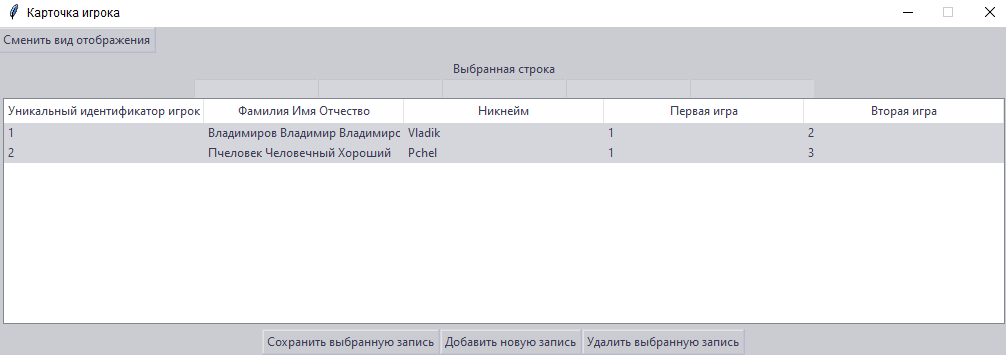


Рисунок 3.4 – Окно таблицы клиентов

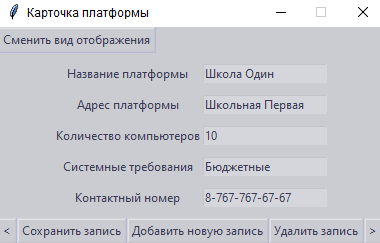


Рисунок 3.5 – Окно карточки платформ

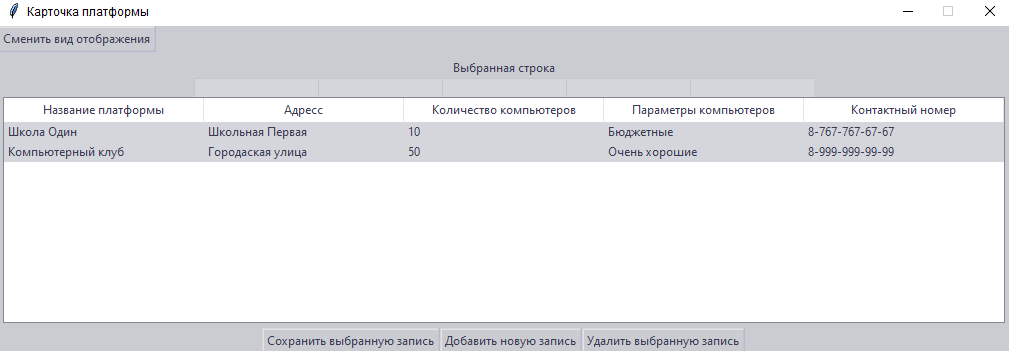


Рисунок 3.6 – Окно таблицы платформ

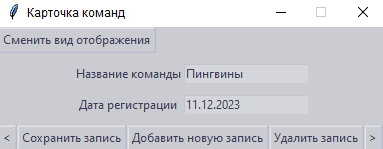


Рисунок 3.7 – Окно карточки команд

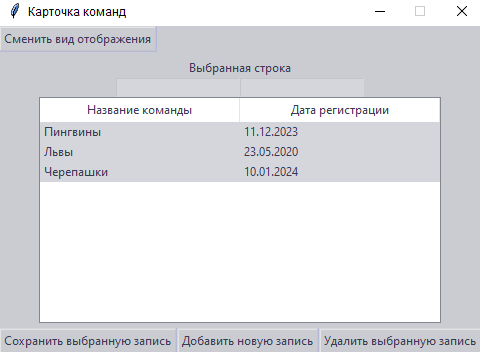


Рисунок 3.8 – Окно таблицы команд

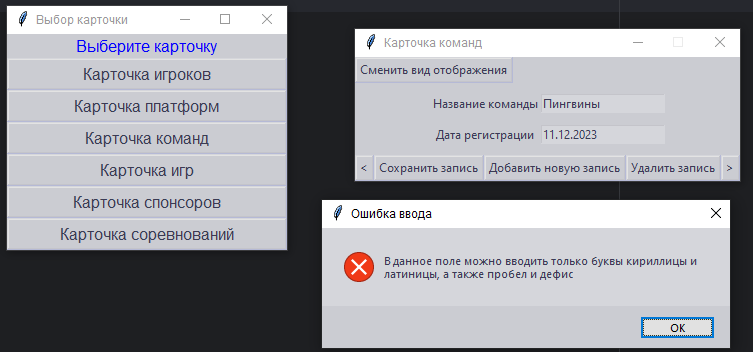


Рисунок 3.9 – Окно сообщения об ошибке на карточке команд

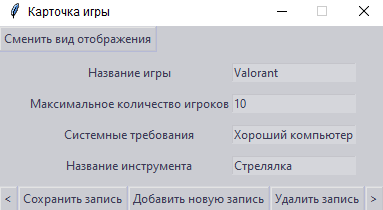


Рисунок 3.10 – Окно карточки игр

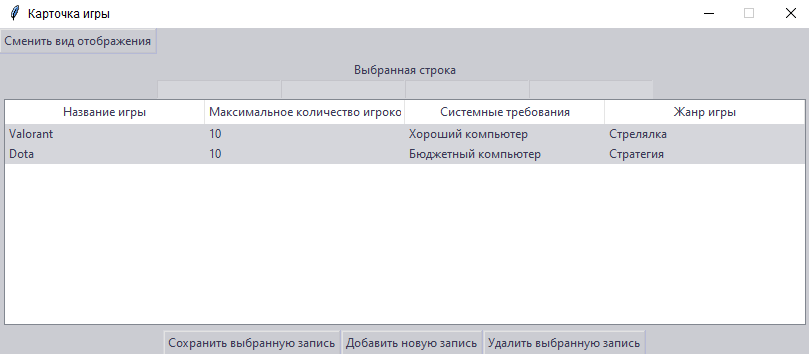


Рисунок 3.11 – Окно таблицы игр

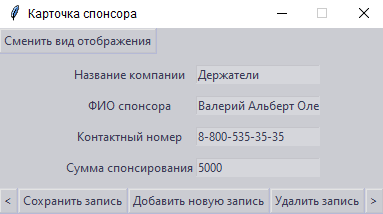


Рисунок 3.12 – Окно карточки спонсоров

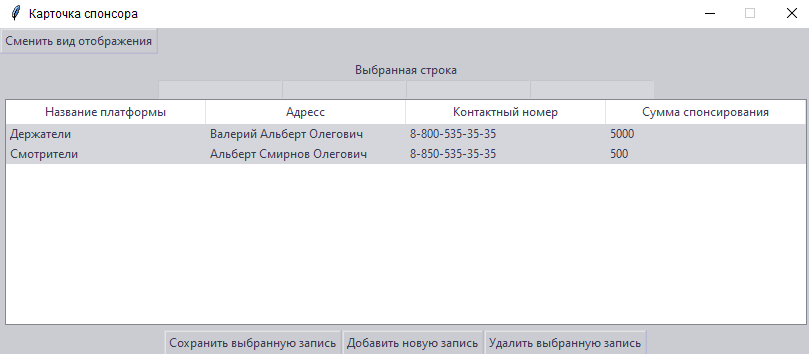


Рисунок 3.13 – Окно таблицы услуг

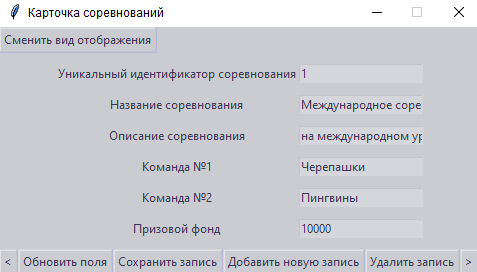
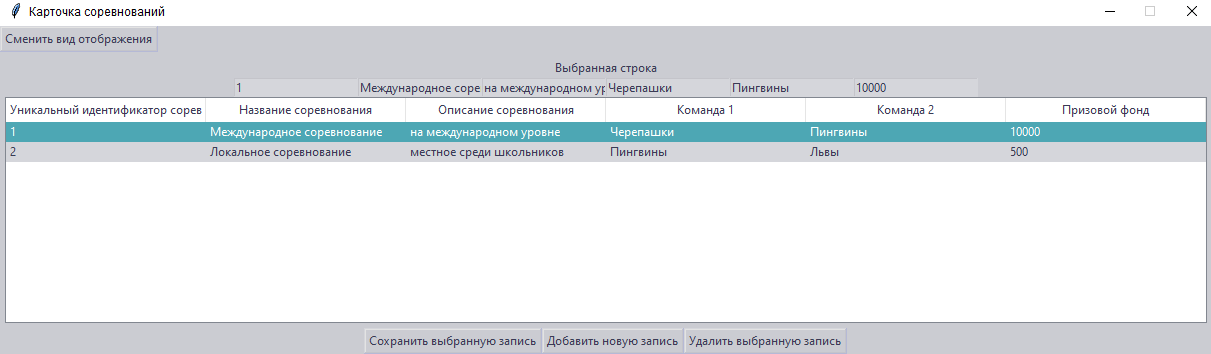


Рисунок 3.14 – Окно карточки заказов

Рисунок 3.15 – Окно таблицы заказов

# ЗАКЛЮЧЕНИЕ

Целью данной работы являлось проектирование базы данных заказов по ремонту квартир и разработка приложения для компании, которая занимается им.

При проектировании базы данных и разработке приложения были решены следующие задачи:

- исследование предметной области ;

- проектирование базы данных;

- Создание базы данных;

- заполнение базы данных информацией;

- разработка интерфейса;

- реализация приложения.

Результатом выполнения данной работы является приложение для работы пользователей с базой данных «Киберспортивные соревнования». При проектировании базы данных использован ER-подход к проектированию реляционных баз данных. Интерфейс приложения содержит: форму для подачи обращения, форма входа в личный кабинет для отслеживания прогресса обращения, панель администратора для модерации обращения, форма для оформления компании в качестве частного исполнителя заявленных пользователем услуг.

Все требования, объявленные в техническом задании, были полностью реализованы в данном программном продукте.

Все задачи, поставленные в начале проектирования проекта, были также решены.

Базу данных и приложение можно использовать в различных организациях, пользующихся исполнением и отслеживанием государственных услуг или их частным решением.

# CПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Малыхина, М.П. Базы данных: основы, проектирование, использование. [Текст] / М.П. Малыхина. - СПб.: BHV, 2007. - 528 c.
2. Мюллер, Р.Д. Проектирование баз данных и UML. [Текст] / Р.Д. Мюллер; Пер. с англ. Е.Н. Молодцова. - М.: Лори, 2013. - 420 c.
3. Гринченко, Н.Н. Проектирование баз данных. СУБД Microsoft Access: Учебное пособие для вузов. [Текст] / Н.Н. Гринченко и др. - М.: РиС, 2013. - 240 c.
4. Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. [Текст] / Т. Коннолли. - М.: Вильямс И.Д., 2017. - 1440 c.
5. Словарь данных и система управления базами данных [Электронный ресурс] // DataDictionaryandDatabaseManagementSystem, 2019. URL: https://studfile.net/preview/942920/page:17/ (Дата обращения: 28.10.2019).
6. Лукин, В.Н. Введение в проектирование баз данных. [Текст] / В.Н. Лукин. - М.: Вузовская книга, 2015. - 144 c.
7. A Newborn Business (Discover the Industry Behind Professional Gaming and Streaming) [Текст] / Zoltan Andrejkovics. - Kindle Edition, 2020. 138 с. : ил. - (A Newborn Business: Esports). ISBN 978-172-383-105-8
8. [Эрик Цтли](https://www.litres.ru/author/erik-ctli/), Новый спорт. История становления киберспортивной индустрии. [Текст] / Ridero, 2023.–270 с., ISBN 978-5-0050-3176-1
9. *Войков Л.М., Кузина Л.Б.* Методические рекомендации по проведению студенческих киберспортивных турниров (с учётом опыта международного киберспортивного фестиваля «Битва за науку» – 2023) / Л.М. Войков, Л.Б. Кузина // Электронный научно-публицистический журнал «Homo Cyberus». 2023. № 2 (15).  ISBN 978-617 - 594-635-0
10. Пол Сандерсон SQLite Forensics [Текст] / 315 c. Amazon 2018 ISBN 1980293074

# ПРИЛОЖЕНИЕ A

Исходный код приложения на Python:

from tkinter import \*  
import player as pl  
import platform as plt  
import sponsor as spns  
import team as tm  
import game as gm  
import competition as comp  
  
# def create():  
window = Tk()  
window.title("Выбор карточки")  
frame = Frame(  
 window  
)  
frame.grid(row=2)  
player = Button(  
 frame,  
 text="Карточка игроков",  
 font=25,  
 command=pl.create\_window,  
 height=1,  
 width=30  
)  
player.grid(row=1)  
team = Button(  
 frame,  
 text="Карточка команд",  
 font=25,  
 command=tm.create\_window,  
 height=1,  
 width=30  
)  
team.grid(row=3)  
game = Button(  
 frame,  
 text="Карточка игр",  
 font=25,  
 command=gm.create\_window,  
 height=1,  
 width=30  
)  
game.grid(row=4)  
sponsor = Button(  
 frame,  
 text="Карточка спонсоров",  
 font=25,  
 command=spns.create\_window,  
 height=1,  
 width=30  
)  
sponsor.grid(row=5)  
platform = Button(  
 frame,  
 text="Карточка платформ",  
 font=25,  
 command=plt.create\_window,  
 height=1,  
 width=30  
)  
platform.grid(row=2)  
competition = Button(  
 frame,  
 text="Карточка соревнований",  
 font=25,  
 command=comp.create\_window,  
 height=1,  
 width=30  
)  
competition.grid(row=6)  
title = Label(  
 window,  
 text="Выберите карточку",  
 foreground="Blue",  
 font=60  
)  
title.grid(row=1)  
mainloop()

import re

import tkinter.messagebox as t

global key

def digit(name, type):

if type == "1" and len(name) == 1:

result = re.match(r"^\d\*$", name) is not None

if not result:

t.showerror("Ошибка ввода", "В данное поле можно вводить только цифры")

return result

else:

return re.match(r"^.\*$", name) is not None

def word(name):

result = re.match(

r"[йцукенгшщзхъфывапролджэячсмитьбюёqwertyuiopasdfghjklzxcvbnmЙЦУКЕНГШЩЗХЪФЫВАПРОЛДЖЭЯЧСМИТЬБЮЁQWERTYUIOPASDFGHJKLZXCVBNM -]\*$",

name) is not None

if not result:

t.showerror("Ошибка ввода",

"В данное поле можно вводить только буквы кириллицы и латиницы, а также пробел и дефис")

return result

def phone(event):

change = event.char

if len(event.widget.get()) >= 19:

event.widget.delete(18, "end")

elif re.match(r"^[+]$", event.widget.get()) is not None:

event.widget.delete(0, "end")

elif re.match(r"^\d$", event.widget.get()) is not None:

event.widget.delete(0, "end")

event.widget.insert(0, "+" + change)

elif re.match(r"^[+]\d\d$", event.widget.get()) is not None:

event.widget.delete(2, "end")

event.widget.insert(2, " (" + change)

elif re.match(r"^[+]\d \(\d{4}$", event.widget.get()) is not None:

event.widget.delete(7, "end")

event.widget.insert(7, ") " + change)

elif re.match(r"^[+]\d \(\d{3}\) \d{4}$", event.widget.get()) is not None:

event.widget.delete(12, "end")

event.widget.insert(12, "-" + change)

elif re.match(r"^[+]\d \(\d{3}\) \d{3}-\d{3}$", event.widget.get()) is not None:

event.widget.delete(15, "end")

event.widget.insert(15, "-" + change)

elif re.match(r"^[+]\d \(\d{3}\) \d{3}-\d{2}-$", event.widget.get()) is not None:

event.widget.delete(15, "end")

elif re.match(r"^[+]\d \(\d{3}\) \d{3}-$", event.widget.get()) is not None:

event.widget.delete(12, "end")

elif re.match(r"^[+]\d \(\d{3}\) $", event.widget.get()) is not None:

event.widget.delete(7, "end")

elif re.match(r"^[+]\d \(\d{3}\)$", event.widget.get()) is not None:

event.widget.delete(7, "end")

elif re.match(r"^[+]\d \($", event.widget.get()) is not None:

event.widget.delete(2, "end")

elif re.match(r"^[+]\d $", event.widget.get()) is not None:

event.widget.delete(2, "end")

def date(event):

change = event.char

if len(event.widget.get()) >= 11:

event.widget.delete(10, "end")

elif re.match(r"^\d{5}$", event.widget.get()) is not None:

event.widget.delete(4, "end")

event.widget.insert(4, "-" + change)

elif re.match(r"^\d{4}-$", event.widget.get()) is not None:

event.widget.delete(4, "end")

elif re.match(r"^\d{4}-\d{3}$", event.widget.get()) is not None:

event.widget.delete(7, "end")

event.widget.insert(7, "-" + change)

elif re.match(r"^\d{4}-\d{2}-$", event.widget.get()) is not None:

event.widget.delete(7, "end")