МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

**«Челябинский государственный университет»**

**(ФГБОУ ВО «ЧелГУ»)**

**Пояснительная записка**

**к курсовой работе**

по дисциплине «Базы и хранилища данных»

Разработка базы данных для предметной области «Скачки»

Челябинск, 2020 год

Лист задания

1. **Провести анализ предметной области по следующему описанию:**

 В информационной системе клуба требуется автоматизация для любителей скачек.

В ИС должна быть информация о лошадях, их владельцах, жокеях (наездниках). Также в клубе время от времени проводятся состязания, необходимо сформировать базу, для хранения информации по каждому проводимому состязанию.

**Перечень входных (первичных) документов**

В качестве первичных документов для решения данной задачи используются:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ФИО владельца | Возраст | Телефон | Дата рождения | Кличка лошади | пол лошади |
| Ажуров  Сергей  Пробкович | 25 | 9275674357 | 23.05.1985 | “стрела” | муж |
| Финник  Борис Владимирович | 43 | 9467586473 | 27.08.1992 | “Jack” | жен |
| Гребенщиков  Павел  Афстафьевич | 19 | 9467365749 | 18.10.1995 | “нео” | муж |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ФИО жокея | возраст | телефон | рейтинг | название состязания | дата проведения | место проведения | занятое место |
| Архипелаг  Сергей  Финляндьевич | 25 | 9275674357 | 8.56 | “приди первым” | 24.07.2015 | ипподром “быстрай” | II |
| Осипов  Анжел  Борисович | 28 | 9275374657 | 9.14 | “в никуда” | 29.09.2016 | ипподром “рыхта” | I |

**Ограничения предметной области**:

* Жокеи (наездники) могут менять лошадь на каждых скачках, то есть в одном соревновании он на одной лошади, а в другом - на другой;
* Все даты хранятся в базе стандартно (DD.MM.YYYY);
* Телефон хранится в 10-ти значном формате;
* Рейтинг жокеев представляет собой 10-бальную шкалу с десятичными дробями (пример: 7.8);
* Рейтинг жокея считается хорошим, если он выше 7 баллов;
* В атрибуте **“кличка”** информация хранится в кавычках (пример: “Jack”);
* Один владелец может иметь несколько лошадей;
* Каждое место в скачках (I,II или III) может занять только один наездник;
* В атрибуте **“пол”** информация хранится в виде “Муж”, “Жен”;
* Скачки проходят каждый месяц на разных ипподромах города.

1. **Выполнить концептуальное (инфологическое) проектирование.**
2. **Выполнить даталогическое проектирование для реляционной базы данных.**
3. **Выбрать СУБД (обосновав выбор) и выполнить физическое проектирование.**
4. **Создать базу данных в выбранной СУБД с учетом ограничений предметной области.**
5. **Реализовать следующие отчеты (запросы):**
6. Вывести всю информацию о владельцах, имеющих более 1 лошади.
7. Вывести всю информацию о жокеях, которых можно выставить на скачки, у которых показатель рейтинга выше заданного пользователем значения.
8. Необходимо знать какие лошади и где заняли первые и вторые места за период, заданный пользователем.
9. Вывести тройку лучших жокеев за последний месяц на ипподроме по выбору пользователя
10. **Выбрать язык программирования и разработать приложение для работы с БД (формы ввода/редактирования данных и отчеты).**
11. **Оформить пояснительную записку**

**Содержание**

[Введение 4](#_Toc53957269)

[Анализ предметной области 5](#_Toc53957270)

[Концептуальное (инфологическое) проектирование 7](#_Toc53957271)

[Даталогическое проектирование для реляционной базы данных. Логическая модель 9](#_Toc53957272)

[Выбор СУБД и физическое проектирование 10](#_Toc53957273)

[Описание приложения для работы с БД 13](#_Toc53957274)

[Заключение 18](#_Toc53957275)

[Список использованной литературы 19](#_Toc53957276)

[Приложение 1 Скрипт для создания базы данных 20](#_Toc53957277)

[Приложение 2 SQL-скрипты запросов для отчетов 24](#_Toc53957278)

[Приложение 3 Исходный код программы 28](#_Toc53957279)

## Введение

Целью данной курсовой работы является применение на практике теоретических знаний, которые были получены в курсе «Базы и хранилища данных».

В ходе данной курсовой работы нужно проанализировать исходные данные, разработать базу данных «Скачки», где будет храниться информация о результатах соревнований.

Нужно изучить особенности проведения скачек, определить основные требования

Разработать концептуальную, логическую и модель базы данных.

Создать схему БД и реализовать базу данных в выбранной СУБД, также обосновать, какие ресурсы и почему были использованы

Нужно разработать интерфейс для работы с базой данных.

## Анализ предметной области

Имеется клуб, который ежемесячно проводит соревнования на нескольких ипподромах города. Нужно автоматизировать информационную систему клуба.

Необходимо регистрировать соревнования, записывать данные жокеев, лошадей, владельцев, дату и место и результаты заездов.

Необходимо реализовать доступ для управления, редактирования и просмотра информации в базе данных

У владельца может быть несколько лошадей

Жокеи могут менять лошадь каждое соревнование.

У жокеев имеется рейтинг, который представляет 10-бальную шкалу. Рейтинг жокея считается хорошим, если он выше 7 баллов.

В результате соревнования жокеям присваиваются места, занятые в заезде.

Места в соревновании уникальные.

В итоге должна быть разработана программа, с помощью которой можно делать выборки в базе данных по соревнованиям и участникам соревнований.

После проведения соревнований пользователь может зайти и посмотреть результат соревнований.

Деятельность базы данных должны поддерживать администратор и регистратор.

Администратор отвечает за работоспособность базы данных и ведение каталога. Регистратор вносит и редактирует информацию. Также существует роль пользователь, который будет просматривать информацию о соревнованиях и отчеты.

## Концептуальное (инфологическое) проектирование

Сущность – реальный или абстрактный объект предметной области, информация о котором должна храниться в базе данных

В данной предметной области можно выделить следующие сущности и атрибуты:

Лошадь (horse): кличка, возраст, владелец

Жокей (jockey): имя, адрес, возраст, рейтинг

Владелец (owner): имя, адрес, дата рождения, телефон

Ипподром (racetrack): название, адрес, телефон,

Заезд (race): название, дата, время, ипподром, лошади, жокеи, занятое место, результат – время заезда

Жокей с помощью лошади участвует в заезде.

Заезд проходит на ипподроме

Владелец владеет лошадью

В результате каждого заезда каждому участнику (жокею) присваивается место согласно времени, которое было показано в заезде.

Лошадь зависит от владельца

Заезд зависит от: жокея, лошади, ипподрома

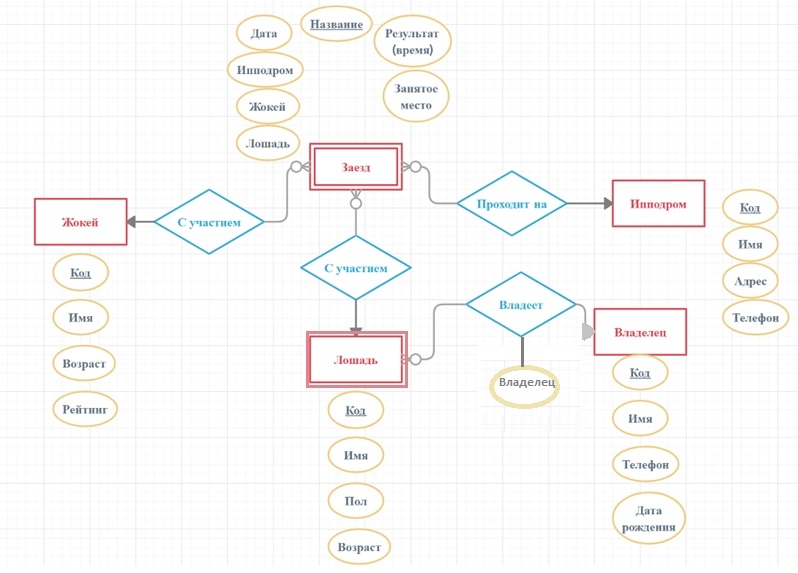
Заезд проходит на 1 ипподроме.

В 1 заезде участвует несколько жокеев.

В 1 заезде участвует несколько лошадей

У 1 владельца может быть несколько лошадей.

Рисунок 1 – Концептуальная модель



## Даталогическое проектирование для реляционной базы данных. Логическая модель

При переходе от концептуальной модели к логической нужно провести нормализацию базы данных, разбив таблицу с соревнованиями на Заезд и Результат заезда, чтобы избавиться от избыточности в сущности Заезд.

Каждый результат заезда будет храниться отдельно от основной информации заезда и будет связан с таблицей заезд с помощью вторичного ключа – код заезда.

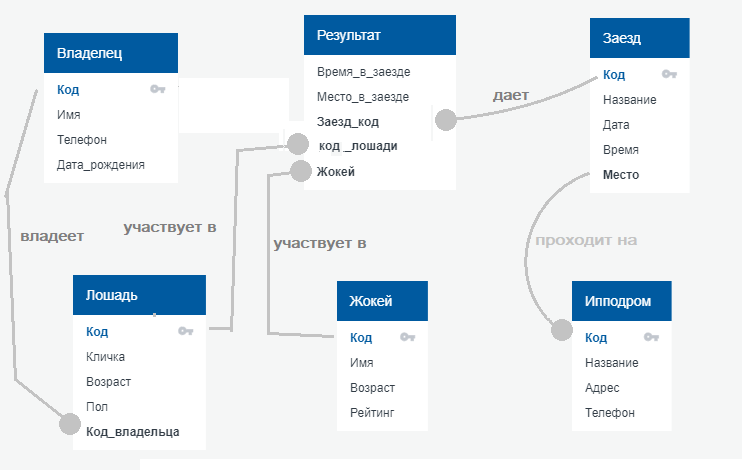
В результате логическая модель должна содержать 6 таблиц: владелец, лошадь, жокей, ипподром, заезд, результат. У каждой таблицы кроме результата заезда имеется код – первичный ключ.

Таблица лошадь содержит вторичный ключ, в котором хранится код владельца.

Таблица заезд содержит вторичный ключ, в котором содержится код ипподрома.

Таблица Результат содержит 3 вторичных ключа, где хранится код заезда, код лошади и код жокея.

Рисунок 2 – логическая модель



## Выбор СУБД и физическое проектирование

В физической модели сущность становится таблицей.

Все сущности из логической модели перейдут в таблицы.

В деноромализации базы данных нет необходимости.

Схема базы данных включает в себя описания содержания, структуры и ограничений [целостности](https://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BB%D0%BE%D1%81%D1%82%D0%BD%D0%BE%D1%81%D1%82%D1%8C_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), используемые для создания и поддержки [базы данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)

База данных будет называться “HorseRaceDb”, схема базы данных - dbo. Схема содержит 6 таблиц.

База данных будет иметь 6 таблиц с полями и условиями:

Таблица 1 Владелец (Owner)

|  |  |  |
| --- | --- | --- |
| Id | Int | PK |
| Name | varchar(50) |  |
| BirthDate | Date |  |
| Telephone | Numeric(10,0) |  |

Таблица 2 Жокей (Jockey)

|  |  |  |
| --- | --- | --- |
| Id | Int | PK |
| Name | Varchar(50) |  |
| BirthDate | Date |  |
| Rating | Decimal(18,2) |  |
| Telephone | Numeric(10,0) |  |

Таблица 3 Ипподром (RaceTrack)

|  |  |  |
| --- | --- | --- |
| Id | Int | PK |
| Name | Varchar(40) |  |
| Address | Varchar(40) |  |
| Telephone | Numeric(10,0) |  |

Таблица 4 Лошадь (Horse)

|  |  |  |
| --- | --- | --- |
| Id | Int | PR |
| Name | Varchar(20) |  |
| Gender | Varchar(5) |  |
| Age | Int |  |
| OwnerId | int | FK |

На таблицу создали 2 ограничения:

Gender принимает только 2 значения «Муж» и «Жен»

([Gender] like '!"Муж!"' escape '!' OR [Gender] like '!"Жен!"' escape '!' )

И имя лошади должно храниться в кавычках

([Name] like '!"%!"' escape '!' )

Таблица 5 Соревнование (Race)

|  |  |  |
| --- | --- | --- |
| Id | Int | PK |
| Name | Varchar(50) |  |
| Date | Date |  |
| Time | Time(7) |  |
| RaceTrackId | int | FK |

Таблица 6 Результат соревнования (RaceReult)

|  |  |  |
| --- | --- | --- |
| ResultPlace | int |  |
| ResultTime | Time(7) |  |
| RaceId | Int | FK |
| HorseId | Int | FK |
| JockeyId | int | FK |

В качестве СУБД было решено использовать Microsoft SQL Server.

Данная СУБД является распространенной, у нее удобный графический интерфейс.

Она позволяет работать с большим объемом данных.

В ней можно регулировать и отслеживать производительность.

Определение индекса:

**Индекс** – объект базы данных для ускорения поиска в ней и повышения производительности. Индекс в таблице может быть только один и должен содержать уникальные значения.

Индексы желательно использовать для таблиц с большим объемом информации, по которым часто будет производителься поиск.

В данной БД индексом будут столбцы id, так как по ним будет чаще всего поиск и нужно, чтобы значение в него добавлялось автоматически

Также индексом будет пара столбцов в таблице соревнования с id соревнования и id жокея, так как эти данные должны быть уникальны и по ним часто будет вестись поиск

Параметры:

Тестовая база данных размещена на локальном сервере, продуктивная будет размещена на отдельном сервере

**Параметры продуктивного сервера:**

Доступность: круглосуточно.

Количество пользователей (нагрузка): примерно 30 в день. В дни проведения соревнований нагрузка будет больше.

Операционная система: Windows Server 2012

Название: HorseRaceDb

Оперативная память: 16 Гб

Размер диска: 500 Гб

Процессор: i5 7500

Домен: HorseRase74.ru

## Описание приложения для работы с БД

Приложение написано на языке С# с использованием интерфейса WPF, среда разработки Visial Studio 2019.

В качестве языка разработки был выбран С#, поскольку он наиболее удобный для работы с СУБД Windows Server 2019 и отладке программы. А WPF-интерфейс многофункционален, гибок и прост в использовании.

Приложение состоит из окна с вкладками.

В каждой вкладке с 1 по 6 можно вывести информацию, добавить и удалить поля.

**Подробнее о вкладках:**

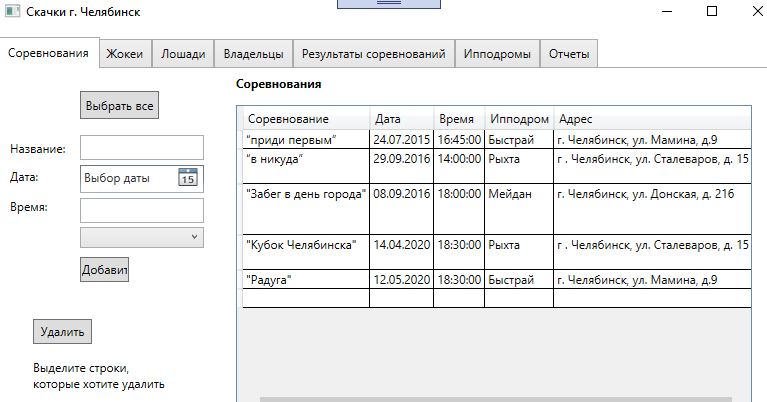
Рисунок 3 – Вкладка 1 Соревнования

Рисунок 4 – Вкладка 1 Соревнования, выбор ипподрома

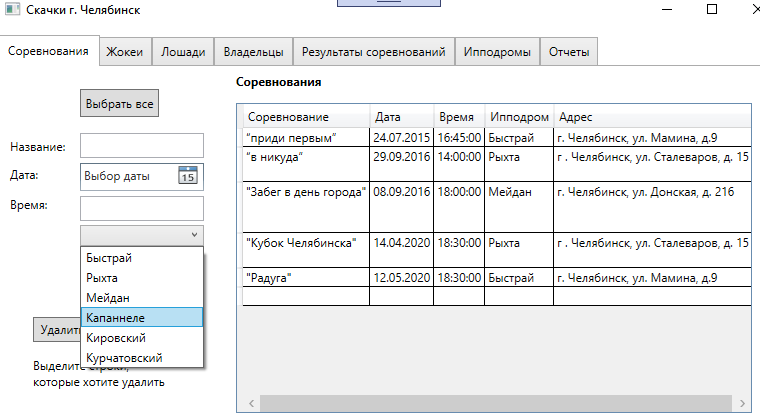


Рисунок 5 – Вкладка 1 Соревнования, выбор строк в DataGrid для удаления

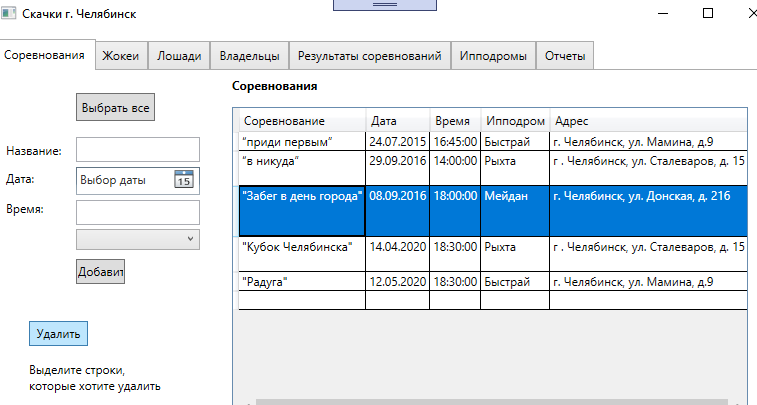


Рисунок 6 - Вкладка 2 Жокеи

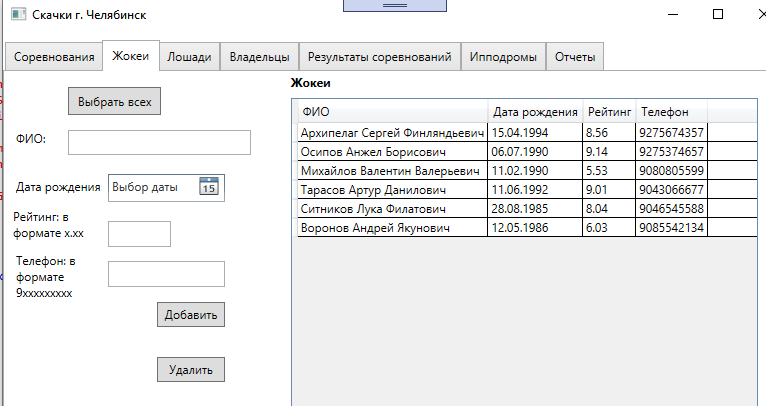


Рисунок 7 - Вкладка 3 Лошади

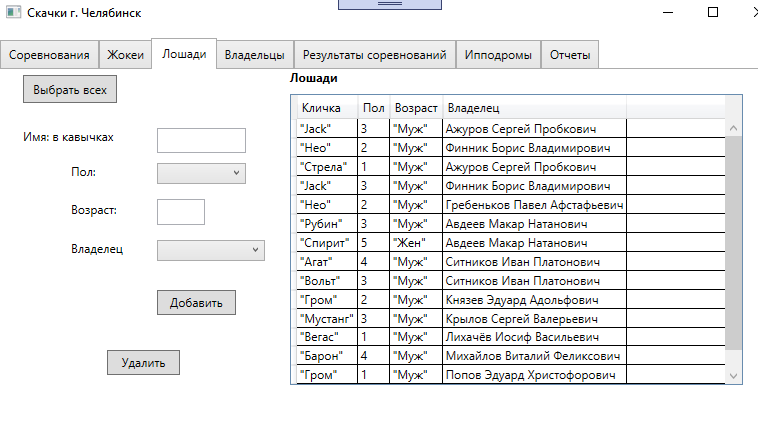


Рисунок 8 - Вкладка 4 Владельцы

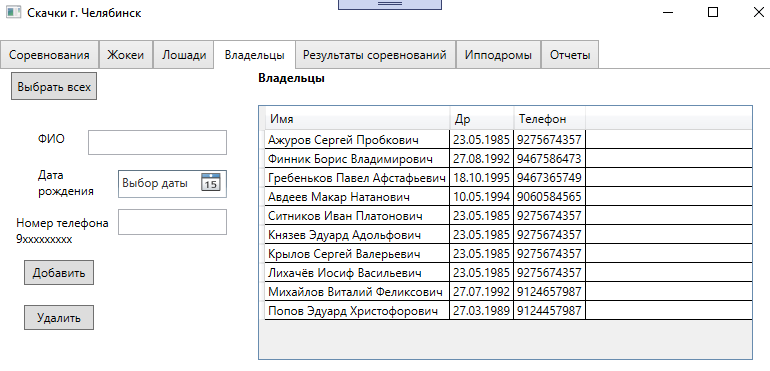


Рисунок 9 - Вкладка 5 Результат соревнований



Рисунок 10 - Вкладка 6 Ипподромы

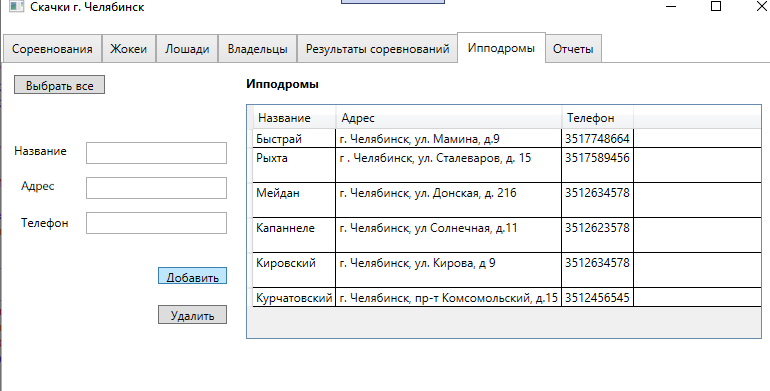
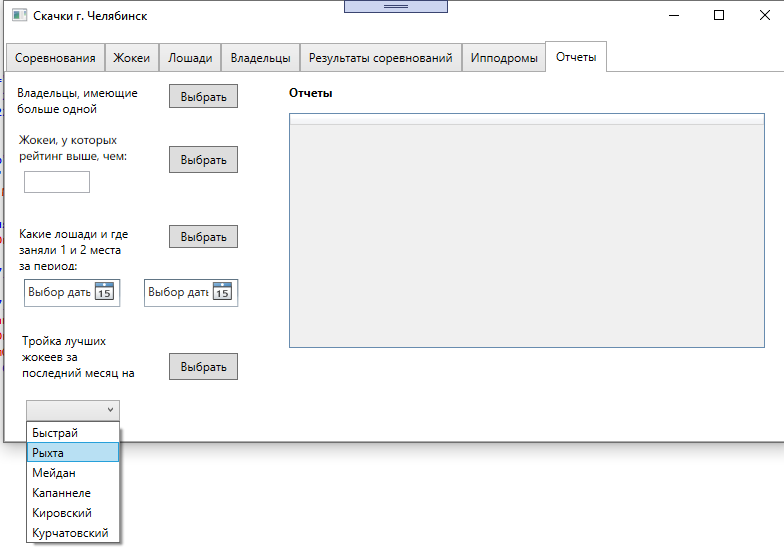


Рисунок 11 - Вкладка 7 Отчеты



## Заключение

В ходе данной курсовой работы мы разработали свою базу данных для клуба, который занимается скачками.

Мы проанализировали предметную область, создали концептуальную, логическую и физическую модели базы данных.

Для выбора подходящей базы мы ознакомились с основными СУБД, их свойствами, положительными и отрицательными характеристиками и выбрали наиболее подходящую.

Создали базу данных с использованием СУБД SQL Sever и заполнили ее тестовыми данными. Написали приложение, которое подключается к базе данных и выполняет запросы.

## Список использованной литературы

1. М.П. Малыхина. Базы данных: основы, проектирование, использование**.** 2-е изд. СПб: БХВ-Петербург, 2007.
2. В.В. Кирилов, Г.Ю. Громов. Введение в реляционные базы данных. – СПб.:БХВ-Петербург, 2009г. – 464с.
3. «Профессиональное руководство по SQL Server. Структура и реализация» Кен Хендерсон
4. Metanit Руководство по MS SQL Server 2017 <https://metanit.com/sql/sqlserver/>
5. Справочник по Transact-SQL (компонент Database Engine)

<https://docs.microsoft.com/ru-ru/sql/t-sql/language-reference?view=sql-server-ver15>

1. 10 причин перейти на Microsoft SQL Server 2019

<https://softline.ru/about/blog/10-prichin-pereyti-na-microsoft-sql-server-2019>

1. «Transact-SQL» Михаил Фленов

### Приложение 1 Скрипт для создания базы данных

CREATE DATABASE HorseRaceDb

GO

Use HorseRaceDb

CREATE TABLE Owner

(

Id INT PRIMARY KEY IDENTITY,

Name varchar(50),

BirthDate date,

Telephone numeric(10,0)

)

CREATE TABLE Horse

(

Id INT PRIMARY KEY IDENTITY,

Name NVARCHAR(20) NOT NULL,

Gender NVARCHAR(5),

Age INT,

OwnerId INT NOT NULL

FOREIGN KEY (OwnerId) REFERENCES Owner(Id),

CHECK([Name] like '!"%!"' escape '!'),

CHECK ([Gender] like '!"Муж!"' escape '!' OR [Gender] like '!"Жен!"' escape '!' )

CREATE TABLE Jockey

(

Id INT PRIMARY KEY IDENTITY,

Name varchar(50) NOT NULL,

BithDate date,

Rating decimal(18,2),

Telephone numeric(10,0)

)

CREATE TABLE RaceTrack

(

Id INT PRIMARY KEY IDENTITY,

Name varchar(50) NOT NULL,

Address varchar(50),

Telephone numeric(10,0)

)

CREATE TABLE Race

(

Id INT PRIMARY KEY IDENTITY,

Name varchar(50),

Date date NOT NULL,

Time time(7) NOT NULL,

RaceTrackId INT NOT NULL,

FOREIGN KEY (RaceTrackId) REFERENCES RaceTrack(Id)

)

CREATE TABLE RaceResult

(

ResultPlace INT UNIQUE,

ResultTime time(7),

RaceId INT NOT NULL,

HorseId INT NOT NULL,

JockeyId INT NOT NULL,

FOREIGN KEY (RaceId) REFERENCES Race(Id),

FOREIGN KEY (HorseId) REFERENCES Horse(Id),

FOREIGN KEY (JockeyId) REFERENCES Jockey(Id)

)

CREATE INDEX Id\_RaceResult

ON RaceResult (RaceId, JockeyId);

**Скрипт для заполнения базы данных:**

INSERT INTO [HorseRaceDb].[dbo].[Owner] (Name, BirthDate, Telephone) values ('Ажуров Сергей Пробкович', '23.05.1985', 9275674357)

INSERT INTO [HorseRaceDb].[dbo].[Owner] (Name, BirthDate, Telephone) values ('Финник Борис Владимирович', '27.08.1992', 9467586473)

INSERT INTO [HorseRaceDb].[dbo].[Owner] (Name, BirthDate, Telephone) values ('Гребеньков Павел Афстафьевич', '18.10.1995', 9467365749)

INSERT INTO [HorseRaceDb].[dbo].[Owner] (Name, BirthDate, Telephone) values ('Авдеев Макар Натанович', '10.05.1994', 9060584565)

INSERT INTO [HorseRaceDb].[dbo].[Owner] (Name, BirthDate, Telephone) values ('Ситников Иван Платонович', '23.05.1985', 9275674357)

INSERT INTO [HorseRaceDb].[dbo].[Owner] (Name, BirthDate, Telephone) values ('Князев Эдуард Адольфович', '23.05.1985', 9275674357)

INSERT INTO [HorseRaceDb].[dbo].[Owner] (Name, BirthDate, Telephone) values ('Крылов Сергей Валерьевич', '23.05.1985', 9275674357)

INSERT INTO [HorseRaceDb].[dbo].[Owner] (Name, BirthDate, Telephone) values ('Лихачёв Иосиф Васильевич', '23.05.1985', 9275674357)

INSERT INTO [HorseRaceDb].[dbo].[Owner] (Name, BirthDate, Telephone) values ('Михайлов Виталий Феликсович', '27.07.1992', 9124657987)

INSERT INTO [HorseRaceDb].[dbo].[Owner] (Name, BirthDate, Telephone) values ('Попов Эдуард Христофорович', '27.03.1989', 9124457987)

INSERT INTO [HorseRaceDb].[dbo].[Jockey] (Name, BirthDate, Rating, Telephone) values ('Архипелаг Сергей Финляндьевич', '15.04.1994', 8.56, 9275674357)

INSERT INTO [HorseRaceDb].[dbo].[Jockey] (Name, BirthDate, Rating, Telephone) values ('Осипов Анжел Борисович', '06.07.1990', 9.14, 9275374657)

INSERT INTO [HorseRaceDb].[dbo].[Jockey] (Name, BirthDate, Rating, Telephone) values ('Михайлов Валентин Валерьевич', '11.02.1990', 5.53, 9080805599)

INSERT INTO [HorseRaceDb].[dbo].[Jockey] (Name, BirthDate, Rating, Telephone) values ('Тарасов Артур Данилович', '11.06.1992', 9.01, 9043066677)

INSERT INTO [HorseRaceDb].[dbo].[Jockey] (Name, BirthDate, Rating, Telephone) values ('Ситников Лука Филатович', '28.08.1985', 8.04, 9046545588)

INSERT INTO [HorseRaceDb].[dbo].[Jockey] (Name, BirthDate, Rating, Telephone) values ('Воронов Андрей Якунович', '12.05.1986', 6.03, 9085542134)

INSERT INTO [HorseRaceDb].[dbo].[Horse] (Name, Gender, Age, OwnerId) values ('"Стрела"', '"Муж"', 1, 2)

INSERT INTO [HorseRaceDb].[dbo].[Horse] (Name, Gender, Age, OwnerId) values ('"Jack"', '"Муж"', 3, 3)

INSERT INTO [HorseRaceDb].[dbo].[Horse] (Name, Gender, Age, OwnerId) values ('"Нео"', '"Муж"', 2, 5)

INSERT INTO [HorseRaceDb].[dbo].[Horse] (Name, Gender, Age, OwnerId) values ('"Рубин"', '"Муж"', 3, 6)

INSERT INTO [HorseRaceDb].[dbo].[Horse] (Name, Gender, Age, OwnerId) values ('"Спирит"', '"Жен"', 5, 6)

INSERT INTO [HorseRaceDb].[dbo].[Horse] (Name, Gender, Age, OwnerId) values ('"Агат"', '"Муж"', 4, 7)

INSERT INTO [HorseRaceDb].[dbo].[Horse] (Name, Gender, Age, OwnerId) values ('"Вольт"', '"Муж"', 3, 7)

INSERT INTO [HorseRaceDb].[dbo].[Horse] (Name, Gender, Age, OwnerId) values ('"Гром"', '"Муж"', 2, 8)

INSERT INTO [HorseRaceDb].[dbo].[Horse] (Name, Gender, Age, OwnerId) values ('"Мустанг"', '"Муж"', 3, 9)

INSERT INTO [HorseRaceDb].[dbo].[Horse] (Name, Gender, Age, OwnerId) values ('"Вегас"', '"Муж"', 1, 10)

INSERT INTO [HorseRaceDb].[dbo].[Horse] (Name, Gender, Age, OwnerId) values ('"Барон"', '"Муж"', 4,11)

INSERT INTO [HorseRaceDb].[dbo].[Horse] (Name, Gender, Age, OwnerId) values ('"Гром"', '"Муж"', 1, 12)

зд

INSERT INTO [HorseRaceDb].[dbo].[RaceTrack] (Name, Address, Telephone) values ('Быстрай','г. Челябинск, ул. Мамина, д.9',3517748664)

INSERT INTO [HorseRaceDb].[dbo].[RaceTrack] (Name, Address, Telephone) values ('Быстрай','г. Челябинск, ул. Мамина, д.9',3517748664)

INSERT INTO [HorseRaceDb].[dbo].[RaceTrack] (Name, Address, Telephone) values ('Быстрай','г. Челябинск, ул. Мамина, д.9',3517748664)

INSERT INTO [HorseRaceDb].[dbo].[RaceTrack] (Name, Address, Telephone) values ('Быстрай','г. Челябинск, ул. Мамина, д.9',3517748664)

INSERT INTO [HorseRaceDb].[dbo].[RaceTrack] (Name, Address, Telephone) values ('Быстрай','г. Челябинск, ул. Мамина, д.9',3517748664)

INSERT INTO [HorseRaceDb].[dbo].[RaceResult] (ResultPlace, ResultTime, HorseId, JockeyId, RaceId) values (2, '0:04:15', 30, 2, 1 )

INSERT INTO [HorseRaceDb].[dbo].[RaceResult] (ResultPlace, ResultTime, HorseId, JockeyId, RaceId) values (1, '0:03:05', 31, 3, 1 )

INSERT INTO [HorseRaceDb].[dbo].[RaceResult] (ResultPlace, ResultTime, HorseId, JockeyId, RaceId) values (3, '0:04:15', 41, 4, 1 )

INSERT INTO [HorseRaceDb].[dbo].[RaceResult] (ResultPlace, ResultTime, HorseId, JockeyId, RaceId) values (4, '0:03:05', 42, 5, 1 )

INSERT INTO [HorseRaceDb].[dbo].[RaceResult] (ResultPlace, ResultTime, HorseId, JockeyId, RaceId) values (5, '0:04:15', 43, 5, 1 )

INSERT INTO [HorseRaceDb].[dbo].[RaceResult] (ResultPlace, ResultTime, HorseId, JockeyId, RaceId) values (6, '0:03:05', 44, 7, 1 )

INSERT INTO [HorseRaceDb].[dbo].[RaceResult] (ResultPlace, ResultTime, HorseId, JockeyId, RaceId) values (1, '0:02:15', 30, 2, 4 )

INSERT INTO [HorseRaceDb].[dbo].[RaceResult] (ResultPlace, ResultTime, HorseId, JockeyId, RaceId) values (2, '0:04:03', 42, 3, 4 )

INSERT INTO [HorseRaceDb].[dbo].[RaceResult] (ResultPlace, ResultTime, HorseId, JockeyId, RaceId) values (3, '0:05:14', 43, 4, 4 )

INSERT INTO [HorseRaceDb].[dbo].[RaceResult] (ResultPlace, ResultTime, HorseId, JockeyId, RaceId) values (4, '0:05:20', 44, 5, 4 )

INSERT INTO [HorseRaceDb].[dbo].[RaceResult] (ResultPlace, ResultTime, HorseId, JockeyId, RaceId) values (5, '0:05:25', 45, 6, 4 )

INSERT INTO [HorseRaceDb].[dbo].[RaceResult] (ResultPlace, ResultTime, HorseId, JockeyId, RaceId) values (6, '0:05:30', 51, 7, 4 )

INSERT INTO [HorseRaceDb].[dbo].[Race] (Name, Date, Time, RaceTrackId) values ('"приди первым"','24.07.2015', '16:45:00', 1)

INSERT INTO [HorseRaceDb].[dbo].[Race] (Name, Date, Time, RaceTrackId) values ('"в никуда"','24.07.2015', '16:45:00', 2)

INSERT INTO [HorseRaceDb].[dbo].[Race] (Name, Date, Time, RaceTrackId) values ('"Забег в день города"','24.07.2015', '16:45:00', 3)

INSERT INTO [HorseRaceDb].[dbo].[Race] (Name, Date, Time, RaceTrackId) values ('"Кубок Челябинска"','24.07.2015', '16:45:00', 2)

INSERT INTO [HorseRaceDb].[dbo].[Race] (Name, Date, Time, RaceTrackId) values ('"Радуга"','24.07.2015', '16:45:00', 1)

### Приложение 2 SQL-скрипты запросов для отчетов

***Запрос 1***

Владельцы, имеющие более 1 лошади

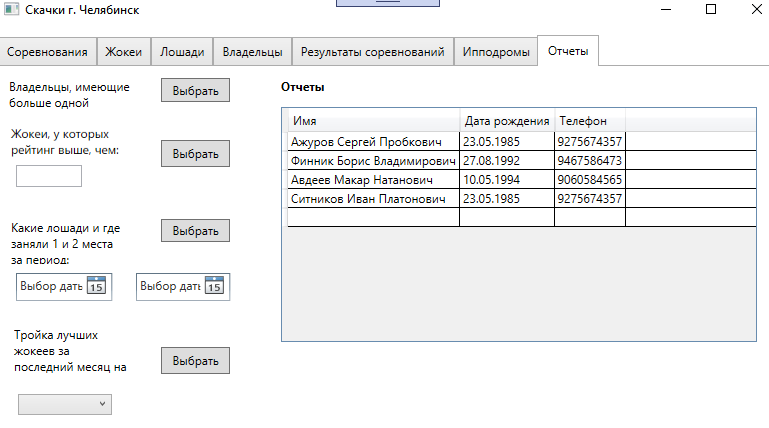
SELECT Owner.Name, Owner.BirthDate, Owner.Telephone

from HorseRaceDb.dbo.Owner

WHERE EXISTS (SELECT OwnerId, COUNT(OwnerId) FROM HorseRaceDb.dbo.Horse WHERE Owner.Id=Horse.OwnerId

GROUP BY OwnerId HAVING COUNT(OwnerId)>1)

Рисунок 12 – Реализация запроса 1 в приложении

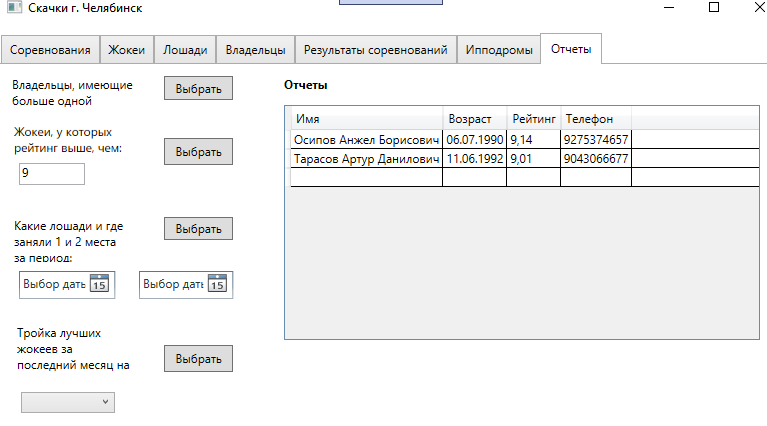


***Запрос 2:***

Вывести информацию о жокеях, которых можно выставить на скачки, у которых показатель рейтинга выше заданного пользователем значения

SELECT Name, BirthDate, Rating, Telephone FROM HorseRaceDb.dbo.Jockey WHERE Rating > raiting;

Рисунок 13 – Реализация запроса 2 в приложении



***Запрос 3:***

Необходимо знать, какие лошади и где заняли первые и вторые места за период, заданный пользователем.

SELECT Horse.Name, RaceTrack.Name, Race.Date, RaceResult.ResultPlace

FROM HorseRaceDb.dbo.Race, HorseRaceDb.dbo.RaceResult, HorseRaceDb.dbo.Horse, HorseRaceDb.dbo.RaceTrack

WHERE Race.Id = RaceResult.RaceId

AND Horse.Id = RaceResult.HorseId

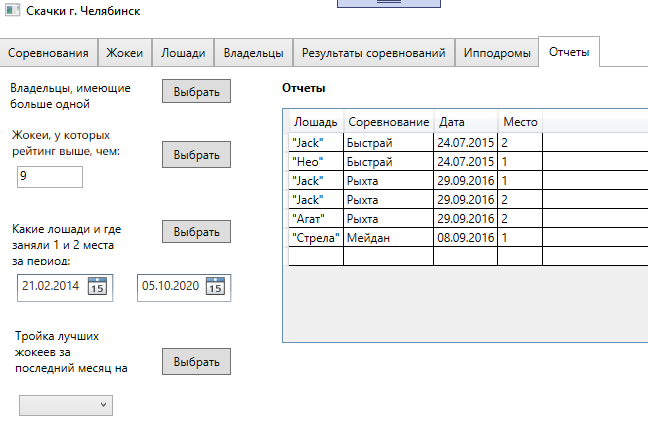
AND RaceTrack.Id = Race.RaceTrackId

AND Date > SelectedDate1

AND Date < SelectedDate2

AND RaceResult.ResultPlace IN(1,2)

Рисунок 14 – Реализация запроса 3 в приложении



***Запрос 4:***

Вывести тройку лучших жокеев за последний месяц на ипподроме по выбору пользователя

SELECT TOP 3 Jockey.Name, Jockey.Rating, Race.Name, Race.Date

FROM HorseRaceDb.dbo.Race, HorseRaceDb.dbo.Jockey, HorseRaceDb.dbo.RaceResult

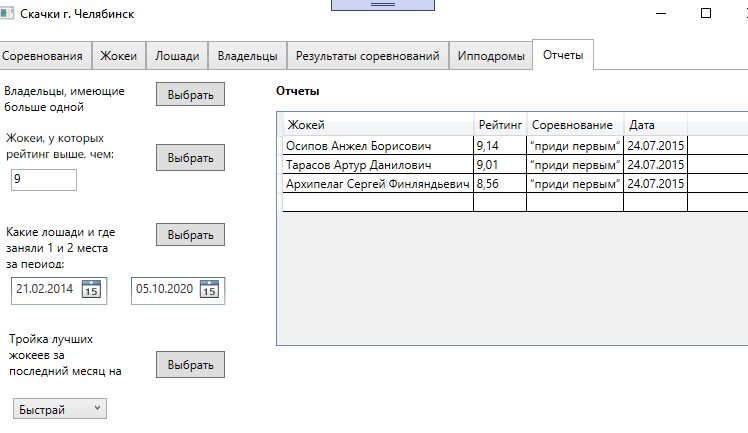
WHERE RaceResult.RaceId=Race.Id AND

RaceResult.JockeyId=Jockey.Id AND

Race.RaceTrackId = SelectedValue

ORDER BY date desc, Rating DESC

Рисунок 15 – Реализация запроса 4 в приложении



### Приложение 3 Исходный код программы

***Класс для подключения к базе данных:***

namespace HorseDb

{

class DbWork

{

public SqlConnection Connect()

{

SqlConnectionStringBuilder builder = new SqlConnectionStringBuilder();

builder.DataSource = @"USER\MSSQLSERVER1";

builder.IntegratedSecurity = true;

return new SqlConnection(builder.ConnectionString);

}

}

}

***Классы для описания сущностей:***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Controls;

namespace HorseDb

{

//Заезд

class Race

{

private int v1;

private string v2;

private string v3;

public int Id { get; set; }

public string Name { get; set; }

public string Data { get; set; }

public string Time { get; set; }

public int TrackId { get; set; }

public Race(int Id, string Name, string Data, string Time, int TrackId)

{

this.Id = Id;

this.Name = Name;

this.Data = Data;

this.Time = Time;

this.TrackId = TrackId;

}

public Race(int v1, string v2, string v3)

{

this.v1 = v1;

this.v2 = v2;

this.v3 = v3;

}

}

//Ипподром

class RaceTrack

{

public int Id { get; set; }

public string Name { get; set; }

public string Adress { get; set; }

public string Telephone { get; set; }

public RaceTrack(int Id, string Name, string Adress, string Telephone)

{

this.Id = Id;

this.Name = Name;

this.Adress = Adress;

this.Telephone = Telephone;

}

}

//Результат заезда

class RaceResult

{

public string Race { get; set; }

public string Horse { get; set; }

public string Jockey { get; set; }

public TimeSpan ResultTime { get; set; }

public int ResultPlace { get; set; }

public RaceResult(string Race, string Horse, string Jockey, TimeSpan ResultTime, int ResultPlace)

{

this.Race = Race;

this.Horse = Horse;

this.Jockey = Jockey;

this.ResultTime = ResultTime;

this.ResultPlace = ResultPlace;

}

}

//Лошадь

public class Horse

{

public int Id { get; set; }

public string Name { get; set; }

public int Age { get; set; }

public string Gender { get; set; }

public string Owner { get; set; }

public Horse(int Id, string Name, string Gender, int Age, string Owner)

{

this.Id = Id;

this.Name = Name;

this.Gender = Gender;

this.Age = Age;

this.Owner = Owner;

}

}

//Жокей

class Jockey

{

public int Id { get; set; }

public string Name { get; set; }

public string BirthDate { get; set; }

public double Rating { get; set; }

public string Telephone { get; set; }

public Jockey(int Id, string Name, string BirthDate, double Rating, string Telephone)

{

this.Id = Id;

this.Name = Name;

this.BirthDate = BirthDate;

this.Rating = Rating;

this.Telephone = Telephone;

}

}

//Владелец

public class Owner

{

public int Id { get; set; }

public string Name { get; set; }

public string BirthDate { get; set; }

public string Telephone { get; set; }

public Owner(int Id, string Name, string BirthDate, string Telephone)

{

this.Id = Id;

this.Name = Name;

this.BirthDate = BirthDate;

this.Telephone = Telephone;

}

}

}

***Класс для работы в WPF***

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace HorseDb

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

//Выбрать все заезды

private void Tb1SelectRaces(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

string sql = "SELECT Race.Id, Race.Name, Race.Date, Race.Time, RaceTrack.Name, RaceTrack.Address FROM HorseRaceDb.dbo.Race, HorseRaceDb.dbo.RaceTrack WHERE Race.RaceTrackId=RaceTrack.Id";

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

DataTable dt = new DataTable();

dt.Columns.Add("Id");

dt.Columns.Add("Соревнование");

dt.Columns.Add("Дата");

dt.Columns.Add("Время");

dt.Columns.Add("Ипподром");

dt.Columns.Add("Адрес");

while (reader.Read())

{

dt.Rows.Add(

reader.GetInt32(0),

reader.GetValue(1).ToString(),

Convert.ToDateTime(reader.GetValue(2)).ToShortDateString(),

reader.GetTimeSpan(3).ToString(),

reader.GetValue(4).ToString(),

reader.GetValue(5).ToString()

);

}

dgRaceTrack.ItemsSource = dt.DefaultView;

dgRaceTrack.Columns[0].Visibility = Visibility.Hidden;

reader.Close();

connection.Close();

}

//Добавить звезд

private void InsertRaceTb1(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

int n = 0;

string s = RaceTrackList.SelectedItem.ToString();

string sql = "SELECT Id FROM HorseRaceDb.dbo.RaceTrack WHERE Name='" + s + "'";

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

n = Convert.ToInt32(reader.GetValue(0));

}

reader.Close();

sql = "INSERT INTO HorseRaceDb.dbo.Race (Name, Date, Time, RaceTrackId) VALUES ('" + Tb1Name.Text + "','" + dp1.SelectedDate.Value.ToShortDateString() + "','" + Tb1Time.Text + "'," + n + ")";

command = new SqlCommand(sql, connection);

command.ExecuteNonQuery();

connection.Close();

MessageBox.Show("Добавлено соревнование: " + Tb1Name.Text);

}

//Открытие списка с ипподромами

void OnDropDownOpened(object sender, EventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT Id, Name From HorseRaceDb.dbo.RaceTrack", connection);

var table = new DataTable();

da.Fill(table);

RaceTrackList.ItemsSource = table.DefaultView;

}

//Удалить заезд

private void Tb1Delete(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

List<DataRowView> str = new List<DataRowView>();

foreach (var r in dgRaceTrack.SelectedItems)

str.Add((DataRowView)r);

SqlCommand command;

string sql;

foreach (var r in str)

{

sql = "DELETE FROM HorseRaceDb.dbo.Race WHERE Id=" + r.Row["Id"];

command = new SqlCommand(sql, connection);

command.ExecuteNonQuery();

}

connection.Close();

MessageBox.Show("Удалено");

}

//Вкладка ЖОКЕИ

//Выбрать всех жокеев

private void ButtonSelectJockeysTb2(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

String sql = "SELECT \* FROM HorseRaceDb.dbo.Jockey";

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

var jockeys = new List<Jockey>();

while (reader.Read())

{

jockeys.Add(new Jockey(

Convert.ToInt32(reader.GetValue(0)),

reader.GetString(1),

Convert.ToDateTime(reader.GetValue(2)).ToShortDateString(),

Convert.ToDouble(reader.GetValue(3)),

reader.GetValue(4).ToString()

));

}

dg2.ItemsSource = jockeys;

dg2.Columns[1].Header = "ФИО";

dg2.Columns[2].Header = "Дата рождения";

dg2.Columns[3].Header = "Рейтинг";

dg2.Columns[4].Header = "Телефон";

dg2.Columns[0].Visibility = Visibility.Hidden;

reader.Close();

connection.Close();

}

//Добавить жокея

private void btnInsertJockey(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

string sql = "INSERT INTO HorseRaceDb.dbo.Jockey (Name, BirthDate, Rating, Telephone) VALUES ('" + Tb2Fio.Text + "','" + dp2.SelectedDate.Value.ToShortDateString() + "','" + Tb2Rating.Text + "','" + Tb2Telephone.Text + "')";

SqlCommand command = new SqlCommand(sql, connection);

command.ExecuteNonQuery();

connection.Close();

MessageBox.Show("Добавлен жокей: " + Tb2Fio.Text);

}

//Удалить жокея

private void Tb2Delete(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

List<Jockey> str = new List<Jockey>();

foreach (var r in dg2.SelectedItems)

str.Add((Jockey)r);

SqlCommand command;

string sql;

foreach (var r in str)

{

sql = "DELETE FROM HorseRaceDb.dbo.Jockey WHERE Id=" + r.Id;

command = new SqlCommand(sql, connection);

command.ExecuteNonQuery();

}

connection.Close();

MessageBox.Show("Удалено");

}

//Выбрать лошадей

private void SelectTb3(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

string sql = "SELECT Horse.Id, Horse.Name, Horse.Gender, Horse.Age, Owner.Name FROM HorseRaceDb.dbo.Horse, HorseRaceDb.dbo.Owner WHERE Horse.OwnerId=Owner.Id";

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

List<Horse> horses = new List<Horse>();

while (reader.Read())

{

horses.Add(new Horse(

Convert.ToInt32(reader.GetValue(0)),

reader.GetValue(1).ToString(),

reader.GetValue(2).ToString(),

Convert.ToInt32(reader.GetValue(3)),

reader.GetValue(4).ToString()

));

}

connection.Close();

dg3.ItemsSource = horses;

dg3.Columns[1].Header = "Кличка";

dg3.Columns[2].Header = "Пол";

dg3.Columns[3].Header = "Возраст";

dg3.Columns[4].Header = "Владелец";

dg3.Columns[0].Visibility = Visibility.Hidden;

reader.Close();

connection.Close();

}

//Открыть список владельцев

void cbOwnersOpened(object sender, EventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

SqlDataAdapter da = new SqlDataAdapter("Select Id, Name From HorseRaceDb.dbo.Owner", connection);

var table = new DataTable();

da.Fill(table);

cbOwners.ItemsSource = table.DefaultView;

}

//Выбор пола лошади

void cbGenderOpened(object sender, EventArgs e)

{

cbGender.Items.Clear();

cbGender.Items.Add("\"Муж\"");

cbGender.Items.Add("\"Жен\"");

}

//Добавить лошадь

private void InsertTb3(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

string sql = "INSERT INTO HorseRaceDb.dbo.Horse (Name, Gender, Age, OwnerId) VALUES ('" + Tb31.Text + "','" + cbGender.SelectedItem.ToString()+ "','" + Tb33.Text + "','" + cbOwners.SelectedValue + "')";

SqlCommand command = new SqlCommand(sql, connection);

command.ExecuteNonQuery();

connection.Close();

MessageBox.Show("Добавлена лошадь: " + Tb31.Text);

}

//Удалить лошадь

private void Tb3Delete(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

List<Horse> str = new List<Horse>();

foreach (var r in dg3.SelectedItems)

str.Add((Horse)r);

SqlCommand command;

string sql;

foreach (var r in str)

{

sql = "DELETE FROM HorseRaceDb.dbo.Horse WHERE Id=" + r.Id;

command = new SqlCommand(sql, connection);

command.ExecuteNonQuery();

}

connection.Close();

MessageBox.Show("Удалена лошадь");

}

//Выбрать всех владельцев

private void Tb4Select(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

string sql = "SELECT \* FROM HorseRaceDb.dbo.Owner";

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

List<Owner> owners = new List<Owner>();

while (reader.Read())

{

owners.Add(new Owner(

Convert.ToInt32(reader.GetValue(0)),

reader.GetValue(1).ToString(),

Convert.ToDateTime(reader.GetValue(2)).ToShortDateString(),

reader.GetValue(3).ToString()

));

}

dg4.ItemsSource = owners;

dg4.Columns[1].Header = "Имя";

dg4.Columns[2].Header = "Др";

dg4.Columns[3].Header = "Телефон";

dg4.Columns[0].Visibility = Visibility.Hidden;

connection.Close();

reader.Close();

connection.Close();

}

//Добавить владельца

private void Tb4Insert(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

string sql = "INSERT INTO HorseRaceDb.dbo.Owner (Name, BirthDate, Telephone) VALUES ('" + Tb41.Text + "','" + dp4.SelectedDate.Value.ToShortDateString()+ "'," + Tb43.Text + ")";

SqlCommand command = new SqlCommand(sql, connection);

command.ExecuteNonQuery();

connection.Close();

MessageBox.Show("Добавлен владелец: " + Tb41.Text);

}

//Удалить владельца

private void Tb4Delete(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

List<Owner> str = new List<Owner>();

foreach (var r in dg4.SelectedItems)

str.Add((Owner)r);

SqlCommand command;

string sql;

foreach (var r in str)

{

sql = "DELETE FROM HorseRaceDb.dbo.Owner WHERE Id=" + r.Id;

command = new SqlCommand(sql, connection);

command.ExecuteNonQuery();

}

connection.Close();

MessageBox.Show("Удален владелец");

}

//Вкладка результат соревнования

private void Tb5Select(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

string sql = "SELECT Race.Name, Horse.Name, Jockey.Name, ResultTime, ResultPlace " +

"FROM HorseRaceDb.dbo.RaceResult, HorseRaceDb.dbo.Race, HorseRaceDb.dbo.Horse, HorseRaceDb.dbo.Jockey " +

"WHERE RaceResult.RaceId=Race.Id AND RaceResult.HorseId = Horse.Id AND JockeyId = Jockey.Id";

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

List<RaceResult> results = new List<RaceResult>();

while (reader.Read())

{

results.Add(new RaceResult(

reader.GetValue(0).ToString(),

reader.GetValue(1).ToString(),

reader.GetValue(2).ToString(),

reader.GetTimeSpan(3),

reader.GetInt32(4)

));

}

dg5.ItemsSource = results;

dg5.Columns[0].Header = "Соревнование";

dg5.Columns[1].Header = "Лошадь";

dg5.Columns[2].Header = "Жокей";

dg5.Columns[3].Header = "Время";

dg5.Columns[4].Header = "Место";

connection.Close();

reader.Close();

connection.Close();

}

//Добавить результат соревнования

private void Tb5Insert(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

string sql = "INSERT INTO HorseRaceDb.dbo.RaceResult (ResultPlace, ResultTime, RaceId, HorseId, JockeyId) VALUES ('" + Tb5Place.Text + "','" + Tb5Time.Text + "'," + cbTb5Race.SelectedValue + ","+ cbTb5Horse.SelectedValue + "," + cbTb5Jockey.SelectedValue + ")";

SqlCommand command = new SqlCommand(sql, connection);

command.ExecuteNonQuery();

connection.Close();

MessageBox.Show("Добавлен результат соревнования: " + Tb41.Text);

}

//Открыть список жокеев

private void cb5OpenedJockey(object sender, EventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

SqlDataAdapter da = new SqlDataAdapter("Select Id, Name From HorseRaceDb.dbo.Jockey", connection);

var table = new DataTable();

da.Fill(table);

cbTb5Jockey.ItemsSource = table.DefaultView;

}

//Открыть список лошадей

private void cb5OpenedHorse(object sender, EventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT Id, Name FROM HorseRaceDb.dbo.Horse", connection);

var table = new DataTable();

da.Fill(table);

cbTb5Horse.ItemsSource = table.DefaultView;

}

//Открыть список заездов

private void cb5OpenedRace(object sender, EventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

SqlDataAdapter da = new SqlDataAdapter("SELECT Id, Name FROM HorseRaceDb.dbo.Race", connection);

var table = new DataTable();

da.Fill(table);

cbTb5Race.ItemsSource = table.DefaultView;

}

//удалить результат

private void Tb5Delete(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

List<RaceResult> str = new List<RaceResult>();

foreach (var r in dg5.SelectedItems)

str.Add((RaceResult)r);

SqlCommand command;

string sql;

foreach (var r in str)

{

sql = "DELETE FROM HorseRaceDb.dbo.RaceResult " +

"WHERE RaceId = (Select Id from HorseRaceDb.dbo.Race Where Name = '" + r.Race + "') AND " +

"JockeyId = (Select Id from HorseRaceDb.dbo.Jockey Where Name = '" + r.Jockey + "') AND " +

"HorseId=(Select Id from HorseRaceDb.dbo.Horse Where Name = '" + r.Horse + "')";

command = new SqlCommand(sql, connection);

command.ExecuteNonQuery();

}

connection.Close();

MessageBox.Show("Удален результат");

}

//Выбрать все ипподромы

private void Tb6Select(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

string sql = "SELECT \* FROM HorseRaceDb.dbo.RaceTrack";

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

List<RaceTrack> raceTracks = new List<RaceTrack>();

while (reader.Read())

{

raceTracks.Add(new RaceTrack(

reader.GetInt32(0),

reader.GetValue(1).ToString(),

reader.GetValue(2).ToString(),

reader.GetValue(3).ToString()

));

}

dg6.ItemsSource = raceTracks;

dg6.Columns[1].Header = "Название";

dg6.Columns[2].Header = "Адрес";

dg6.Columns[3].Header = "Телефон";

dg6.Columns[0].Visibility = Visibility.Hidden;

connection.Close();

reader.Close();

connection.Close();

}

//Добавить ипподром

private void Tb6Insert(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

string sql = "INSERT INTO HorseRaceDb.dbo.RaceTrack (Name, Address, Telephone) VALUES ('" + Tb6Name.Text + "','" + Tb6Address.Text + "'," + Tb6Telephone.Text + ")";

SqlCommand command = new SqlCommand(sql, connection);

command.ExecuteNonQuery();

connection.Close();

MessageBox.Show("Добавлен ипподром: " + Tb6Name.Text);

}

//Удалить ипподром

private void Tb6Delete(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

List<RaceTrack> str = new List<RaceTrack>();

foreach (var r in dg6.SelectedItems)

str.Add((RaceTrack)r);

SqlCommand command;

string sql;

foreach (var r in str)

{

sql = "DELETE FROM HorseRaceDb.dbo.RaceTrack WHERE Id=" + r.Id;

command = new SqlCommand(sql, connection);

command.ExecuteNonQuery();

}

connection.Close();

MessageBox.Show("Ипподром удален");

}

//Отчет 1

private void Tb7\_1(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

String sql = "Select Owner.Name, Owner.BirthDate, Owner.Telephone from HorseRaceDb.dbo.Owner where EXISTS (SELECT OwnerId, COUNT(OwnerId) FROM HorseRaceDb.dbo.Horse where Owner.Id=Horse.OwnerId group by OwnerId having COUNT(OwnerId)>1)";

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

DataTable dt = new DataTable();

dt.Columns.Add("Имя");

dt.Columns.Add("Дата рождения");

dt.Columns.Add("Телефон");

while (reader.Read())

{

dt.Rows.Add(

reader.GetValue(0).ToString(),

Convert.ToDateTime(reader.GetValue(1)).ToShortDateString(),

reader.GetValue(2).ToString()

);

}

connection.Close();

dg7.ItemsSource = dt.DefaultView;

}

//Отчет 2

private void Tb7\_2(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

string raiting = Tb7Raiting.Text;

String sql = "select Name, BirthDate, Rating, Telephone from HorseRaceDb.dbo.Jockey where Rating>" + raiting;

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

DataTable dt = new DataTable();

dt.Columns.Add("Имя");

dt.Columns.Add("Возраст");

dt.Columns.Add("Рейтинг");

dt.Columns.Add("Телефон");

while (reader.Read())

{

dt.Rows.Add(

reader.GetValue(0).ToString(),

Convert.ToDateTime(reader.GetValue(1)).ToShortDateString(),

reader.GetValue(2).ToString(),

reader.GetValue(3).ToString()

);

}

connection.Close();

dg7.ItemsSource = dt.DefaultView;

}

//Отчет 3

private void Tb7\_3(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

string sql = "Select Horse.Name, RaceTrack.Name, Race.Date, RaceResult.ResultPlace " +

"from HorseRaceDb.dbo.Race, HorseRaceDb.dbo.RaceResult, HorseRaceDb.dbo.Horse, HorseRaceDb.dbo.RaceTrack " +

"Where Race.Id = RaceResult.RaceId " +

"and Horse.Id = RaceResult.HorseId " +

"and RaceTrack.Id = Race.RaceTrackId " +

"and Date>'" + dp7\_1.SelectedDate + "' AND Date<'" + dp7\_2.SelectedDate +

"' and RaceResult.ResultPlace IN(1,2)";

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

DataTable dt = new DataTable();

dt.Columns.Add("Лошадь");

dt.Columns.Add("Соревнование");

dt.Columns.Add("Дата");

dt.Columns.Add("Место");

while (reader.Read())

{

dt.Rows.Add(

reader.GetValue(0).ToString(),

reader.GetValue(1).ToString(),

Convert.ToDateTime(reader.GetValue(2)).ToShortDateString(),

reader.GetValue(3).ToString()

);

}

connection.Close();

dg7.ItemsSource = dt.DefaultView;

}

//Отчет 4

private void Tb7\_4(object sender, RoutedEventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

string sql = "SELECT TOP 3 Jockey.Name, Jockey.Rating, Race.Name, Race.Date " +

"FROM HorseRaceDb.dbo.Race, HorseRaceDb.dbo.Jockey, HorseRaceDb.dbo.RaceResult " +

"WHERE RaceResult.RaceId=Race.Id AND " +

"RaceResult.JockeyId=Jockey.Id AND " +

"Race.RaceTrackId="+ cbTb7.SelectedValue +

" ORDER BY date desc, Rating desc";

SqlCommand command = new SqlCommand(sql, connection);

SqlDataReader reader = command.ExecuteReader();

DataTable dt = new DataTable();

dt.Columns.Add("Жокей");

dt.Columns.Add("Рейтинг");

dt.Columns.Add("Соревнование");

dt.Columns.Add("Дата");

while (reader.Read())

{

dt.Rows.Add(

reader.GetValue(0).ToString(),

reader.GetValue(1).ToString(),

reader.GetValue(2).ToString(),

Convert.ToDateTime(reader.GetValue(3)).ToShortDateString()

);

}

connection.Close();

dg7.ItemsSource = dt.DefaultView;

}

//Выбрать все ипподромы для отчета

void cb7OpenedReport1(object sender, EventArgs e)

{

var connection = new DbWork().Connect();

connection.Open();

SqlDataAdapter da = new SqlDataAdapter("Select Id, Name From HorseRaceDb.dbo.RaceTrack", connection);

var table = new DataTable();

da.Fill(table);

cbTb7.ItemsSource = table.DefaultView;

}

}

}

***Интерфейс WPF***

<Window x:Class="HorseDb.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:HorseDb"

mc:Ignorable="d"

Title="Скачки г. Челябинск" Height="450" Width="800">

<Grid>

<TabControl x:Name="Races" Margin="0,10,0,0" RenderTransformOrigin="0.676,0.428">

<TabItem x:Name="TbRaces">

<TabItem.Header>

<StackPanel >

<TextBlock Margin="3">Соревнования</TextBlock>

</StackPanel>

</TabItem.Header>

<TabItem.Content>

<Grid Margin="0,0,26,20" RenderTransformOrigin="1.039,0.706" Height="365" VerticalAlignment="Bottom">

<Grid.ColumnDefinitions>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Button x:Name="btnTb1Select" Content="Выбрать все" Margin="80,41,593,296" Click="Tb1SelectRaces"/>

<Button x:Name="buttonInsertRace" Content="Добавить" Click="InsertRaceTb1" Margin="80,207,623,133"/>

<Button x:Name="btTb1Del" Content="Удалить" Margin="33,269,660,71" Click="Tb1Delete" RenderTransformOrigin="0.364,0.429"/>

<TextBlock Margin="10,90,680,250" Text="Название:"/>

<TextBlock Margin="10,118,680,222"><Run Text="Дата"/><Run Text=":"/></TextBlock>

<TextBlock Margin="10,148,680,192"><Run Text="Время"/><Run Text=":"/></TextBlock>

<TextBlock Margin="33,309,573,10" TextWrapping="Wrap"><Run Text="Выделите строки, "/><Run Text="которые хотите удалить"/></TextBlock>

<TextBlock Margin="236,25,429,315" Text="Соревнования" FontWeight="Bold"/>

<TextBox x:Name="Tb1Name" Margin="80,85,548,255" TextWrapping="Wrap" Text=""/>

<DatePicker Name="dp1" SelectedDateFormat="Short" Margin="80,115,548,222" />

<TextBox x:Name="Tb1Time" Margin="80,148,548,192" Text=""/>

<ComboBox Name="RaceTrackList" Height="21" VerticalAlignment="Top" Margin="80,177,548,0" DisplayMemberPath="Name" SelectedValuePath="Id" DropDownOpened="OnDropDownOpened" />

<!--SelectionChanged="phonesList\_SelectionChanged" -->

<DataGrid x:Name="dgRaceTrack" Margin="236,55,0,0"/>

</Grid>

</TabItem.Content>

</TabItem>

<TabItem x:Name="TbJockeys">

<TabItem.Header>

<StackPanel >

<TextBlock Margin="3">Жокеи</TextBlock>

</StackPanel>

</TabItem.Header>

<TabItem.Content>

<Grid Margin="0,0,26,20" RenderTransformOrigin="1.039,0.706" Height="365" VerticalAlignment="Bottom">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="177\*"/>

<ColumnDefinition Width="40\*"/>

<ColumnDefinition Width="543\*"/>

</Grid.ColumnDefinitions>

<TextBlock Margin="70,21,426,325" Text="Жокеи" Grid.Column="2" FontWeight="Bold"/>

<Button Content="Выбрать всех" Margin="62,34,20,303" Click="ButtonSelectJockeysTb2" RenderTransformOrigin="0.724,0.5"/>

<Button Content="Добавить" Margin="151,249,533,91" Click="btnInsertJockey" Grid.ColumnSpan="3" />

<Button Content="Удалить" Margin="151,304,533,36" Click="Tb2Delete" RenderTransformOrigin="0.364,0.429" Grid.ColumnSpan="3"/>

<TextBox x:Name="Tb2Fio" TextWrapping="Wrap" Text="" Margin="62,77,507,263" Grid.ColumnSpan="3"/>

<DatePicker Name="dp2" SelectedDateFormat="Short" Margin="102,121,533,216" Grid.ColumnSpan="3" />

<TextBox x:Name="Tb2Rating" TextWrapping="Wrap" Text="" Margin="102,168,10,171"/>

<TextBox x:Name="Tb2Telephone" TextWrapping="Wrap" Text="" Margin="102,208,533,131" Grid.ColumnSpan="3"/>

<TextBlock Margin="10,77,120,263" Text="ФИО:"/>

<TextBlock Margin="10,125,80,215" Text="Дата рождения:"/>

<TextBlock Margin="7,155,80,171" TextWrapping="Wrap" Text="Рейтинг: в формате х.хх"/>

<TextBlock Margin="10,199,90,111" TextWrapping="Wrap" Text="Телефон: в формате 9ххххххххх"/>

<DataGrid x:Name="dg2" Grid.Column="2" Margin="70,45,0,10" />

</Grid>

</TabItem.Content>

</TabItem>

<TabItem x:Name="TbHorses">

<TabItem.Header>

<StackPanel >

<TextBlock Margin="3">Лошади</TextBlock>

</StackPanel>

</TabItem.Header>

<TabItem.Content>

<Grid Margin="0,0,26,20" RenderTransformOrigin="1.039,0.706" Height="365" VerticalAlignment="Bottom">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="127\*"/>

<ColumnDefinition Width="633\*"/>

</Grid.ColumnDefinitions>

<TextBlock Margin="163,18,408,322" Text="Лошади" Grid.Column="1" FontWeight="Bold"/>

<TextBox x:Name="Tb31" TextWrapping="Wrap" Margin="30,77,507,263" Grid.Column="1"/>

<TextBox x:Name="Tb33" TextWrapping="Wrap" Margin="30,148,548,191" Grid.Column="1"/>

<TextBlock Margin="10,77,120,263" Text="ФИО:"/>

<TextBlock Margin="22,77,10,263"><Run Text="Имя:"/><Run Text=" в кавычках"/></TextBlock>

<TextBlock Margin="70,112,10,227" TextWrapping="Wrap" Text="Пол:"/>

<TextBlock Margin="70,150,623,191" TextWrapping="Wrap" Text="Возраст:" RenderTransformOrigin="2.34,0.792" Grid.ColumnSpan="2"/>

<ComboBox Name="cbOwners" Height="21" VerticalAlignment="Top" Margin="30,189,488,0" DropDownOpened="cbOwnersOpened" Grid.Column="1" DisplayMemberPath="Name" SelectedValuePath="Id" />

<ComboBox Name="cbGender" Height="21" VerticalAlignment="Top" Margin="30,112,507,0" DropDownOpened="cbGenderOpened" Grid.Column="1"/>

<Button Content="Выбрать всех" Margin="22,24,10,313" Click="SelectTb3" RenderTransformOrigin="0.737,0.893"/>

<Button Content="Добавить" Margin="30,239,517,101" Click="InsertTb3" Grid.Column="1"/>

<Button x:Name="btnTb3Delete" Content="Удалить" Margin="106,299,573,41" RenderTransformOrigin="0.364,0.429" Grid.ColumnSpan="2" Click="Tb3Delete"/>

<DataGrid x:Name="dg3" Grid.Column="1" Margin="163,43,10,31" />

<TextBlock Margin="70,189,623,152" TextWrapping="Wrap" Text="Владелец" RenderTransformOrigin="0.667,0.625" Grid.ColumnSpan="2"/>

</Grid>

</TabItem.Content>

</TabItem>

<TabItem x:Name="TbOwners" Height="28" VerticalAlignment="Bottom">

<TabItem.Header>

<StackPanel Orientation="Horizontal">

<TextBlock Margin="3">Владельцы</TextBlock>

</StackPanel>

</TabItem.Header>

<TabItem.Content>

<Grid Margin="0,0,26,20" RenderTransformOrigin="1.039,0.706" Height="365" VerticalAlignment="Bottom">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="291\*"/>

<ColumnDefinition Width="89\*"/>

</Grid.ColumnDefinitions>

<TextBlock Margin="257,18,245,322" Text="Владельцы" FontWeight="Bold" RenderTransformOrigin="0.778,0.44"/>

<Button x:Name="btnTb4Select" Content="Выбрать всех" Margin="10,21,480,316" Click="Tb4Select" RenderTransformOrigin="0.627,0.571"/>

<Button x:Name="btnTb4Insert" Content="Добавить" Margin="23,209,483,131" Click="Tb4Insert"/>

<Button x:Name="btn4Delete" Content="Удалить" Margin="23,254,483,86" RenderTransformOrigin="0.364,0.429" Click="Tb4Delete"/>

<DataGrid x:Name="dg4" Margin="257,54,0,56" Grid.ColumnSpan="2" />

<TextBlock Margin="37,79,510,261" Text="ФИО" />

<TextBlock Margin="37,115,483,212" ><Run Text="Дата "/><LineBreak/><Run Text="рождения"/></TextBlock>

<TextBlock Margin="15,163,465,156" TextWrapping="Wrap" Text="Номер телефона 9ххххххххх" />

<TextBox x:Name="Tb41" TextWrapping="Wrap" Margin="87,79,350,261"/>

<TextBox x:Name="Tb43" TextWrapping="Wrap" Margin="117,158,350,181"/>

<DatePicker Name="dp4" SelectedDateFormat="Short" Margin="117,119,350,218" />

</Grid>

</TabItem.Content>

</TabItem>

<TabItem x:Name="RaceResults">

<TabItem.Header>

<StackPanel >

<TextBlock Margin="3">Результаты соревнований</TextBlock>

</StackPanel>

</TabItem.Header>

<TabItem.Content>

<Grid Margin="0,0,26,20" RenderTransformOrigin="1.039,0.706" Height="365" VerticalAlignment="Bottom">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="127\*"/>

<ColumnDefinition Width="125\*"/>

<ColumnDefinition Width="508\*"/>

</Grid.ColumnDefinitions>

<TextBlock Margin="10,16,431,324" Text="Результат" Grid.Column="2" FontWeight="Bold"/>

<Button x:Name="btnTb5Select" Content="Выбрать все" Margin="22,24,10,313" RenderTransformOrigin="0.724,0.5" Click="Tb5Select"/>

<Button x:Name="btnTb5Insert1" Content="Добавить" Margin="104,229,53,108" RenderTransformOrigin="0.724,0.5" Click="Tb5Insert" Grid.ColumnSpan="2"/>

<ComboBox Name="cbTb5Jockey" Height="21" VerticalAlignment="Top" Margin="104,163,10,0" DisplayMemberPath="Name" SelectedValuePath="Id" DropDownOpened="cb5OpenedJockey" Grid.ColumnSpan="2" />

<ComboBox Name="cbTb5Horse" Height="21" VerticalAlignment="Top" Margin="104,189,10,0" DisplayMemberPath="Name" SelectedValuePath="Id" DropDownOpened="cb5OpenedHorse" Grid.ColumnSpan="2" />

<ComboBox Name="cbTb5Race" Height="21" VerticalAlignment="Top" Margin="104,137,10,0" DisplayMemberPath="Name" SelectedValuePath="Id" DropDownOpened="cb5OpenedRace" Grid.ColumnSpan="2" />

<TextBlock TextWrapping="Wrap" Margin="15,77,23,263" Text="Занятое место" />

<TextBlock Margin="59,107,28,233" Text="Время" RenderTransformOrigin="0.725,0.44" />

<TextBox x:Name="Tb5Place" TextWrapping="Wrap" Margin="104,77,10,263" Grid.ColumnSpan="2"/>

<DataGrid x:Name="dg5" Grid.Column="2" Margin="10,43,10,31" />

<TextBox x:Name="Tb5Time" TextWrapping="Wrap" Margin="104,107,10,233" Grid.ColumnSpan="2"/>

<TextBlock Margin="15,138,23,202" Text="Соревнование" RenderTransformOrigin="0.725,0.44" />

<TextBlock Margin="59,167,28,173" Text="Жокей" RenderTransformOrigin="0.45,0.48" />

<TextBlock Margin="55,192,28,148" Text="Лошадь" RenderTransformOrigin="0.886,0.16" />

<Button x:Name="btnTb5Delete" Content="Удалить" Margin="92,296,65,41" RenderTransformOrigin="0.724,0.5" Click="Tb5Delete" Grid.ColumnSpan="2"/>

</Grid>

</TabItem.Content>

</TabItem>

<TabItem x:Name="TbRaceTracks">

<TabItem.Header>

<StackPanel Orientation="Horizontal">

<TextBlock Margin="3">Ипподромы</TextBlock>

</StackPanel>

</TabItem.Header>

<TabItem.Content>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="181\*"/>

<ColumnDefinition Width="212\*"/>

</Grid.ColumnDefinitions>

<TextBlock Margin="242,10,418,330" Text="Ипподромы" FontWeight="Bold" Grid.ColumnSpan="2"/>

<Button Content="Выбрать все" Margin="10,10,257,336" Click="Tb6Select"/>

<Button Content="Добавить" Margin="154,202,135,146" Click="Tb6Insert"/>

<Button Content="Удалить" Margin="154,240,135,106" Click="Tb6Delete"/>

<TextBlock Margin="17,149,285,196" Text="Телефон"/>

<TextBlock Margin="17,112,285,231" Text="Адрес" TextOptions.TextHintingMode="Animated"/>

<TextBlock Margin="10,77,285,266" Text="Название"/>

<TextBox x:Name="Tb6Telephone" TextWrapping="Wrap" Margin="82,147,135,196"/>

<TextBox x:Name="Tb6Name" TextWrapping="Wrap" Margin="82,77,135,266"/>

<TextBox x:Name="Tb6Address" TextWrapping="Wrap" Margin="82,112,135,231"/>

<DataGrid x:Name="dg6" Margin="242,39,20,91" Grid.ColumnSpan="2" />

</Grid>

</TabItem.Content>

</TabItem>

<TabItem >

<TabItem.Header>

<StackPanel Orientation="Horizontal">

<TextBlock Margin="3">Отчеты</TextBlock>

</StackPanel>

</TabItem.Header>

<TabItem.Content>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="181\*"/>

<ColumnDefinition Width="212\*"/>

</Grid.ColumnDefinitions>

<TextBlock Margin="282,10,20,339" Text="Отчеты" FontWeight="Bold"/>

<Button Content="Выбрать" Margin="162,10,127,331" Click="Tb7\_1"/>

<Button Content="Выбрать" Margin="162,72,127,266" Click="Tb7\_2"/>

<TextBlock Margin="12,151,230,169" TextWrapping="Wrap" Text="Какие лошади и где заняли 1 и 2 места за период:"/>

<TextBlock Margin="12,57,230,276" TextWrapping="Wrap" Text="Жокеи, у которых рейтинг выше, чем:" TextOptions.TextHintingMode="Animated"/>

<TextBlock Margin="10,10,225,321" Text="Владельцы, имеющие больше одной лошади" TextWrapping="Wrap"/>

<TextBlock Margin="15,258,225,59" TextWrapping="Wrap" Text="Тройка лучших жокеев за последний месяц на ипподроме:"/>

<TextBox x:Name="Tb7Raiting" TextWrapping="Wrap" Margin="17,97,275,246"/>

<DatePicker Name="dp7\_1" SelectedDateFormat="Short" Margin="17,205,245,132" />

<DatePicker Name="dp7\_2" SelectedDateFormat="Short" Margin="137,205,127,132" />

<DataGrid x:Name="dg7" Margin="282,39,20,91" Grid.ColumnSpan="2" />

<Button Content="Выбрать" Margin="162,151,127,191" Click="Tb7\_3" RenderTransformOrigin="0.658,0.593"/>

<Button Content="Выбрать" Margin="162,279,127,59" Click="Tb7\_4" RenderTransformOrigin="0.452,0.63"/>

<ComboBox Name="cbTb7" Height="21" VerticalAlignment="Top" Margin="19,326,245,0" DisplayMemberPath="Name" SelectedValuePath="Id" DropDownOpened="cb7OpenedReport1" />

</Grid>

</TabItem.Content>

</TabItem>

</TabControl>

</Grid>

</Window>