Федеральное агентство связи

Сибирский Государственный Университет Телекоммуникаций и

Информатики СибГУТИ

Кафедра прикладной математики и кибернетики

Визуальное программирование и человеко-машинное

взаимодействие.

РГР. Часть 1.

Исследование предметной области и создание ER диаграммы.

Вариант 8

Выполнил(а): студент 2 курса,

группы ИП-014 Заескова Вера Вячеславовна

Проверил: ст. преподаватель Милешко Антон Владимирович

Новосибирск, 2022 г.

**РГР. Часть 1**

**Задание на РГР**

Создать ПО для отображения и обработки статистических данных для определённого вида спорта. ПО должно включать 2 основных окна: окно отображающее таблицы БД со статистической информацией и результаты запросов к БД, переключение таблиц ирезультатов должно быть реализовано через вкладки; и окно для менеджера запросов к БД.

Первое окно должно давать возможность просматривать и изменять все таблицы БД, а также просматривать результаты запросов к БД. Должна иметься возможность удалить вкладки с результатами запросов, но не вкладки с таблицами. Также должна иметься возможность перейти к окну менеджера запросов.

Окно менеджера запросов должно предоставлять интерфейс для создания, сохранения, удаления, редактирования запросов. Созданные запросы должны отображаться в виде спискас названиями запросов, в который можно добавлять новые запросы, удалять, просматривать существующие. Для создания и редактирования запросов должен предоставляться визуальный интерфейс, а не язык запросов. Редактор запросов должен поддерживать операции выборки, соединения, группирования, подзапросы (в качестве подзапроса используются ранее сохранённые запросы).

Ход работы:

1. Исследование предметной области и создание ER диаграммы.
2. Перевод ER диаграммы в реляционную модель, создание и заполнение БД.
3. Проработка визуального интерфейса приложения
4. Создание диаграммы классов приложения
5. Реализация основного окна приложения
6. Реализация менеджера запросов
7. Тестирование и отладка

Лошадиные скачки UK flat ([www.attheraces.com/results](http://www.attheraces.com/results/)/ [https://www.flatstats.co.uk](https://www.flatstats.co.uk/) https://[www.horseracing.net/)](http://www.horseracing.net/))

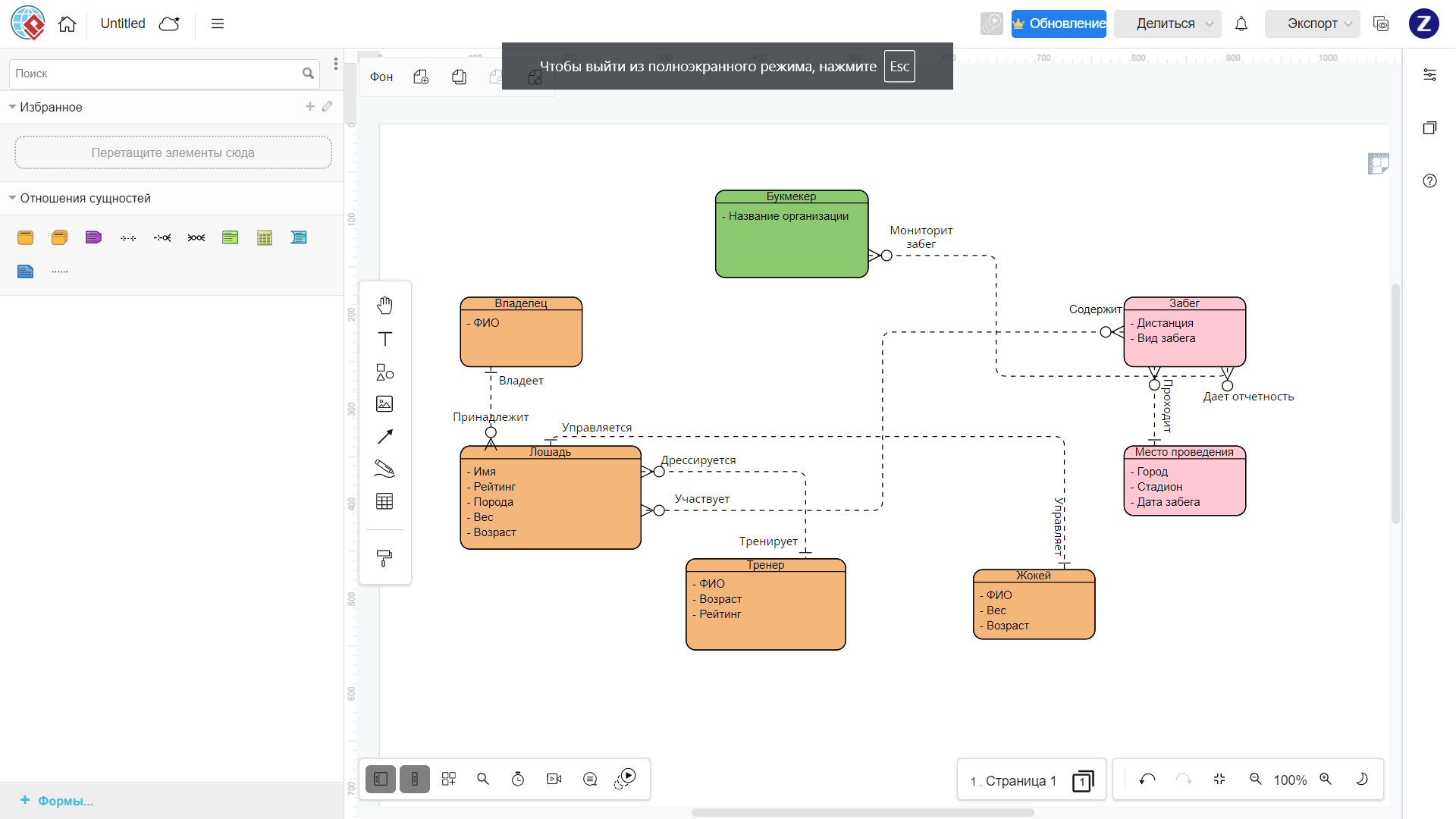
**Предметная область**

**Тема: Лошадиные скачки (гладкие)**

1. Лошадиные скачки
   1. Владелец
      1. ФИО
   2. Тренер
      1. ФИО
      2. Возраст
      3. Рейтинг
   3. Лошадь
      1. Имя
      2. Рейтинг
      3. Порода
      4. Вес
      5. Возраст
   4. Жокей
      1. ФИО
      2. Вес
      3. Возраст
   5. Место проведения
      1. Город
      2. Стадион
      3. Дата забега
   6. Забег
      1. Дистанция
      2. Вид забега
   7. Букмекер
      1. Название букмекерской организации

# ER-диаграмма

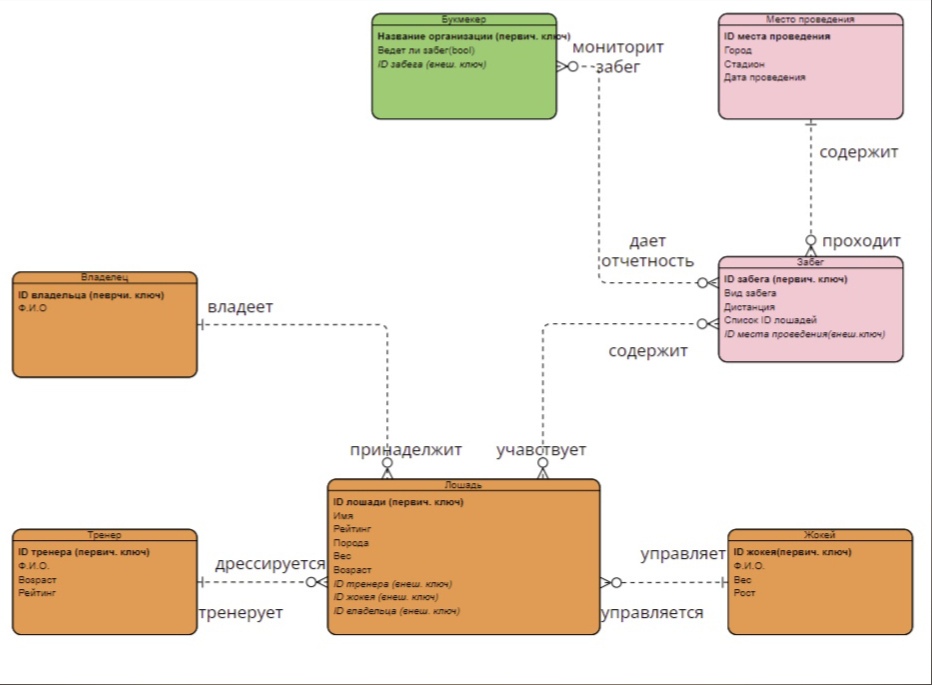
**Сущности**: владелец, лошадь, тренер, жокей, место проведение, забег, букмекер.



**РГР. Часть 2. ER-модель**

Дана ER-модель, задача преобразовать ее в реляционную модель.

**ER-модель:**



**Распишем каждое отношение как сущность:**

Сущности: Лошадь, Тренер, Жокей, Владелец, Забег, Место проведения, Букмекер

**Распишем каждую сущность как отношение:**

**1. Лошадь:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID лошади** | **Имя** | **Рейтинг** | **Порода** | **Вес** | **Возраст** | **ID тренера** | **ID жокея** | **ID владельца** |

**2. Тренер:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID тренера** | **Ф.И.О.** | **Возраст** | **Рейтинг** |

**3. Жокей:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID жокея** | **Ф.И.О.** | **Вес** | **Рост** |

**4. Владелец:**

|  |  |
| --- | --- |
| **ID владельца** | **Ф.И.О.** |

**5. Забег:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID забега** | **Вид забега** | **Дистанция** | **Список ID лошадей** | **ID места проведения** |

**6. Место проведения:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID места проведения** | **Город** | **Стадион** | **Дата проведения** |

**7. Букмекер:**

|  |  |  |
| --- | --- | --- |
| **Название организации** | **Ведет ли забег** | **ID забега** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID лошади** | **Имя** | **Рейтинг** | **Порода** | **Вес** | **Возраст** | **ID тренера** | **ID жокея** | **ID владельца** |

М

1

М

М

М

|  |  |  |  |
| --- | --- | --- | --- |
| **ID тренера** | **Ф.И.О.** | **Возраст** | **Рейтинг** |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID жокея** | **Ф.И.О.** | **Вес** | **Рост** |

1

1

|  |  |
| --- | --- |
| **ID владельца** | **Ф.И.О.** |

М

1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID забега** | **Вид забега** | **Дистанция** | **Список ID лошадей** | **ID места проведения** |

1

М

|  |  |  |  |
| --- | --- | --- | --- |
| **ID места проведения** | **Город** | **Стадион** | **Дата проведения** |

М

|  |  |  |
| --- | --- | --- |
| **Название организации** | **Ведет ли забег** | **ID забега** |

**РГР. Часть 3. Проработка визуального интерфейса приложения**

Главное окно приложения содержит в себе основное меню:

File:

* Save (сохранение БД);
* Load (загрузка БД);
* Exit (выход из приложения).

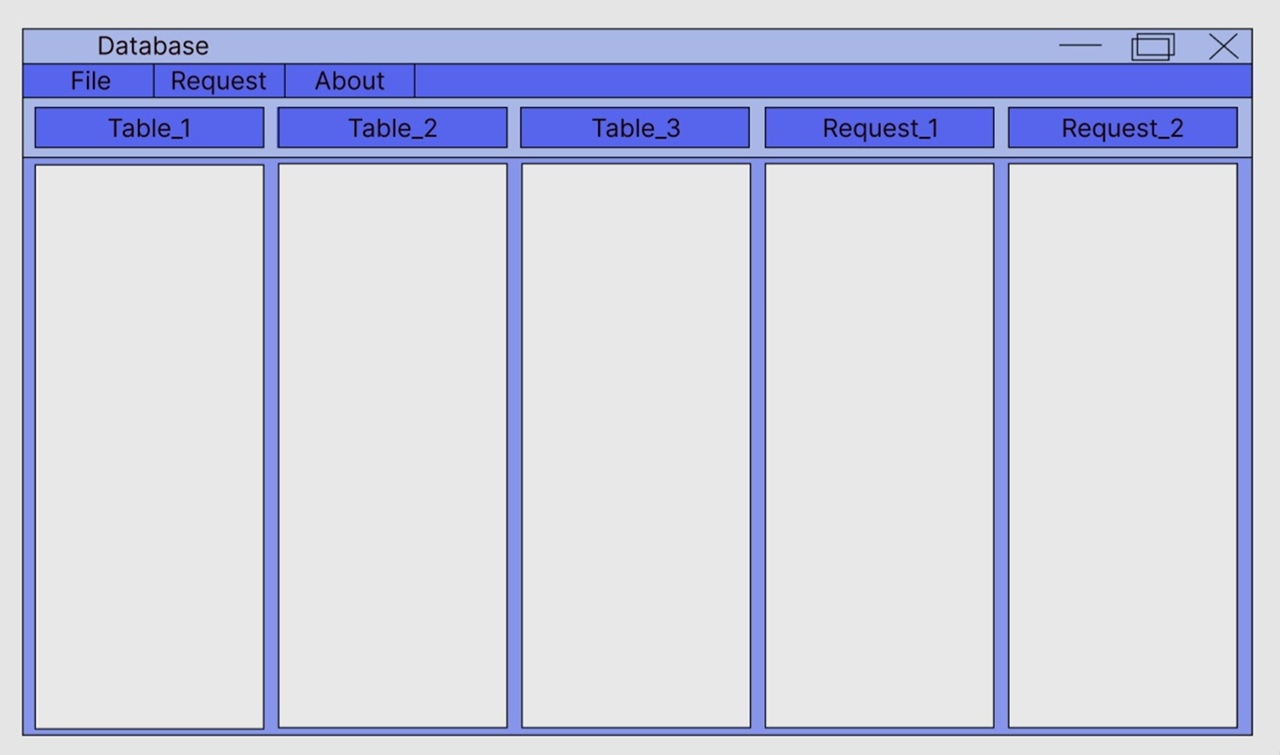
About: о приложении.

Request: открывает окно со списком запросов.

Ниже под меню находятся вкладки с таблицами и результатами запросов.

При смене вкладок меняется содержимое, отображаемое в таблице ниже.

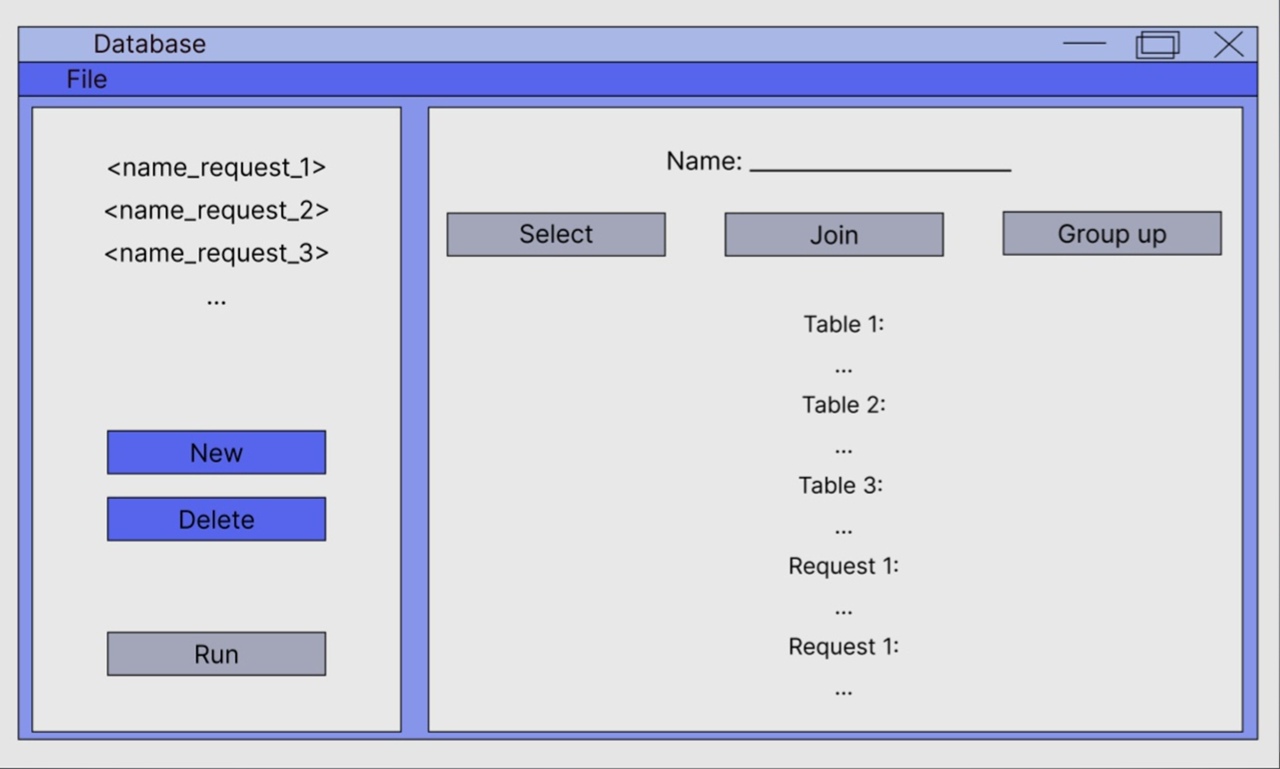
Главное меню:



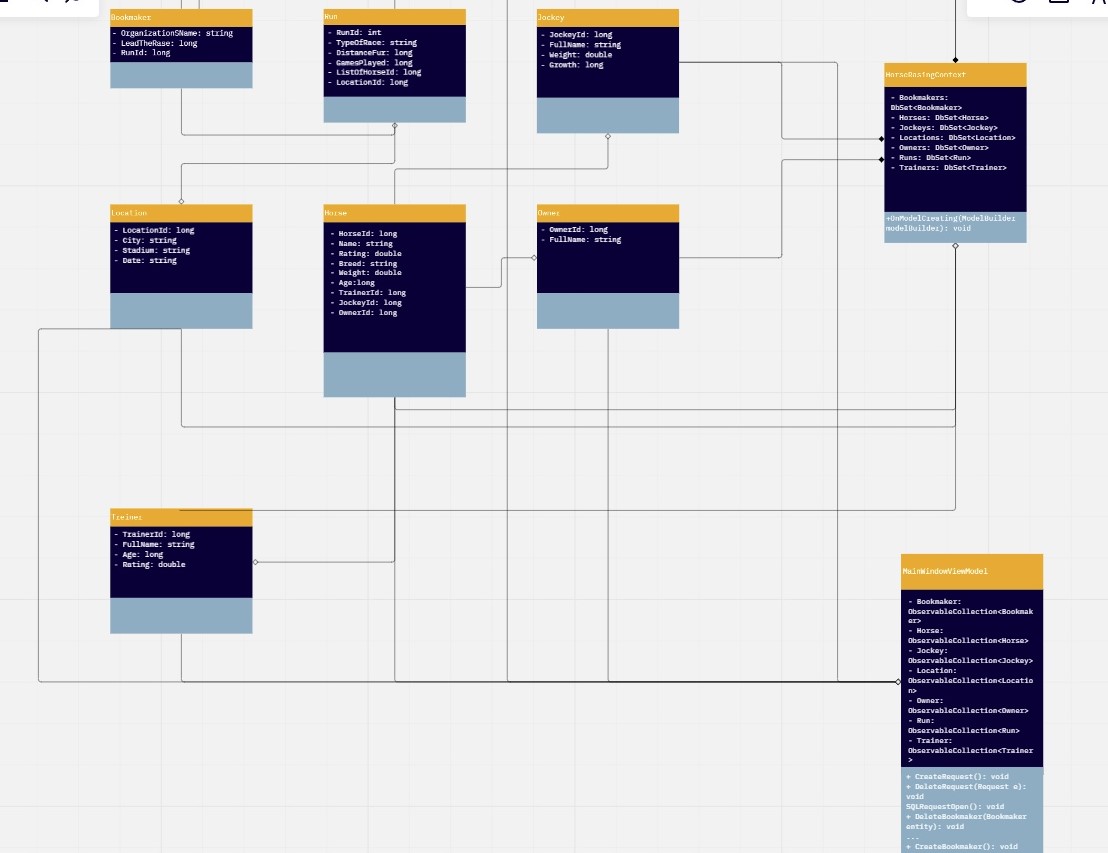
В списке запросов можно создать новый или удалить запрос.

Также можно запустить выполнение запроса. Справа находится конструктор запроса.

Список запросов:



**РГР. Часть 4. Диаграмма классов.**

****

**Класc HorseRidingContext** - контекстуальный класс, который позволяет работать с базой данных через модели. Каждый DbSet представляет собой сущности в БД

public virtual DbSet<Bookmaker> Bookmakers - представление сущности отношения Bookmaker;

public virtual DbSet<Horse> Horses- представление сущности отношения Horse;

public virtual DbSet<Jockey> Jockeys- представление сущности отношения Jockey;

public virtual DbSet<Location> Locations- представление сущности отношения Location;

public virtual DbSet<Owner> Owners - представление сущности отношения Owner;

public virtual DbSet<Run> Runs - представление сущности отношения Run;

public virtual DbSet<Trainer> Trainers- представление сущности отношения Trainer;

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder) - метод считывания базы данных, из которой будет собираться информация.

protected override void OnModelCreating(ModelBuilder modelBuilder) - метод, который точно пересоздает каждое отношение нашей базы данных как объект, с которым можно взаимодействовать в коде.

**Классы представления нашей базы данных:**

**Класс Bookmaker**

public string OrganizationSName { get; set; } = null!;

public long? LeadTheRase { get; set; }

public long? RunId { get; set; }

**Класс Horse**

public long HorseId { get; set; }

public string? Name { get; set; }

public double? Rating { get; set; }

public string? Breed { get; set; }

public double? Weight { get; set; }

public long? Age { get; set; }

public long? TrainerId { get; set; }

public long? JockeyId { get; set; }

public long? OwnerId { get; set; }

**Класс Jockey**

public long JockeyId { get; set; }

public string? FullName { get; set; }

public double? Weight { get; set; }

public long? Growth { get; set; }

**Класс Location**

public long LocationId { get; set; }

public string? City { get; set; }

public string? Stadium { get; set; }

public string? Date { get; set; }

**Класс Owner**

public long OwnerId { get; set; }

public string? FullName { get; set; }

**Класс Run**

public long RunId { get; set; }

public string? TypeOfRace { get; set; }

public long? DistanceFur { get; set; }

public long? ListOfHorseId { get; set; }

public long? LocationId { get; set; }

**Класс Trainer**

public long TrainerId { get; set; }

public string? FullName { get; set; }

public long? Age { get; set; }

public double? Rating { get; set; }

**public class MainWindowViewModel : ViewModelBase -** класс, в котором представлены все основные данные и методы для работы со слоем View, каждая ObservableCollection – наблюдаемая коллекция, содержимое которых будет отображаться в окне программы, также они непосредственно связаны с моделями из DatabaseContext.

public ObservableCollection< Bookmaker > Bookmaker;

public ObservableCollection< Horse > Horse;

public ObservableCollection< Jockey > Jockey;

public ObservableCollection< Location > Location;

public ObservableCollection< Owner > Owner;

public ObservableCollection< Run > Run;

public void CreateRequest();

public void DeleteRequest(Request e);

public void SQLRequestOpen();

public void SQLRequestRun();

**MainWindow -** отрисовка главного окна.

**DataBaseView -** отрисовка всех отношений в главном окне и методов взаимодействия с ними.

**SQLRequestView** - отрисовка запросов к бд.

**РГР. Часть 5. Основное окно приложения.**

В данной части нашей работы было необходимо разработать основное окно нашего приложения. Я выполнила это задание используя диаграмму классов и получила следующий результат:

