

Федеральное агентство связи  
Сибирский Государственный Университет Телекоммуникаций и  
Информатики  
СибГУТИ  
Кафедра прикладной математики и кибернетики  
РГР по дисциплине Визуальное программирование и человеко-машинное  
взаимодействие. Ч. 1

Выполнил: студент 2 курса, группы ИП-014

Рыльский Григорий Максимович

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 и результатов 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 списка с 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

## Вариант задания - 22

22. Теннис (<https://datahub.io/sports-data/atp-world-tour-tennis-data>  
<https://www.atptour.com/en/stats/> <https://www.ultimatetennisstatistics.com/>)

### Предметная область

Тема: Теннис

#### 1. Теннис;

1.1 Men's World Tennis Tour;

1.2 Место проведения;

1.3 Сезон;

1.3.2 Общая статистика сезона;

#### 1.4 Турнир

1.4.1 Дата проведения турнира;

1.4.2 Четверти турнира (1/16, 1/8, 1/4, etc.)

1.4.3 Статистика турнира;

1.4.4 Статистика игроков турнира;

1.4.5 Финальный результат (финал - победитель);

#### 1.5 Матчи

1.5.1 ID матча (по четверти, турниру и спортсменам);

1.5.2 Список игроков (2 игрока);

1.5.3 Покрытие;

1.5.4 Результат игры;

#### 1.6. Игрок

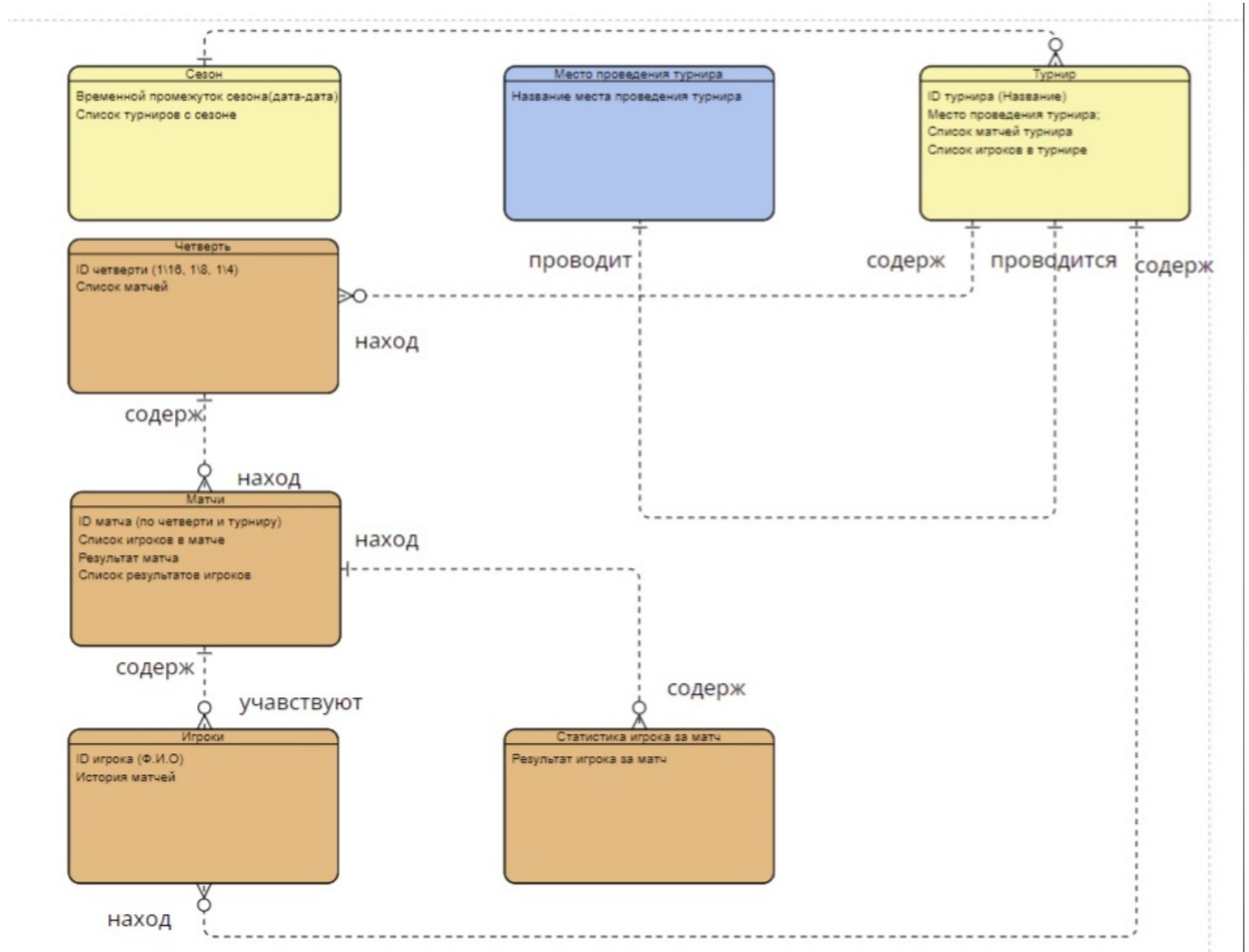
1.6.1 Ф.И.О. участника;

1.6.2 Статистика игрока (Current: ATP SR, ITF SR, Highest: ATP SR, ITF SR, Professional win-loss records, etc.);

1.6.3 Статистика матчей\турниров;

# ER-диаграмма

**Сущности:** Сезон, место проведения, турнир, четверть, матчи, игроки, статистика игрока за матч.



## **2.Реляционная модель:**

### **1. Matches**

- 1.1 ID(Integer) – первичный ключ
- 1.2 winner – внешний ключ
- 1.3 player1 – внешний ключ
- 1.4 player2 – внешний ключ
- 1.5 quarter – внешний ключ

### **2.Player**

- 1.1 ID(Integer) – первичный ключ
- 1.2 name(String)
- 1.3 country(String)
- 1.4 rank(Integer)

### **3.PlayerStats**

- 1.1 ID(Integer) – первичный ключ
- 1.2 aces(Integer)
- 1.3 doubleFaults(Integer)
- 1.4 score(Integer)
- 1.5 match – внешний ключ
- 1.6 player – внешний ключ

### **4.Quarter**

- 1.1 ID(Integer) – первичный ключ
- 1.2 qtrType(Integer)
- 1.3 tourney – внешний ключ

### **5.Season**

- 1.1 ID (Integer) – первичный ключ
- 1.2 StartDate(Time)

1.3 EndDate(Time)

1.4 name(String)

## 6. Tourney

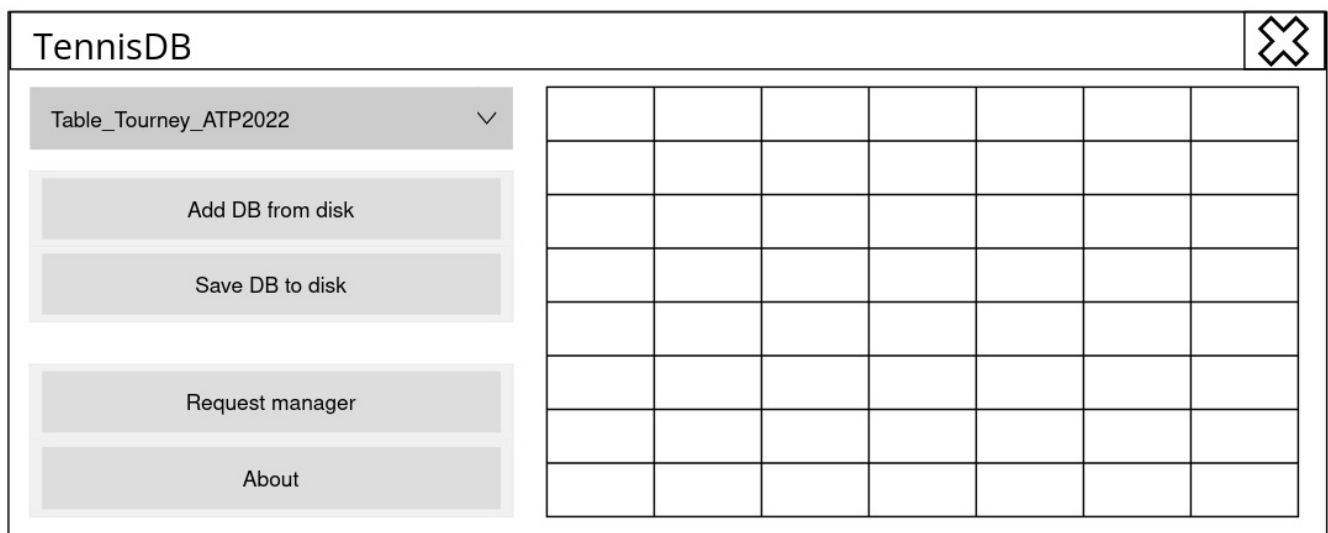
1.1 ID – первичный ключ

1.2 place(String)

1.3 name(String)

1.4 season – Внешний ключ

## 3.Проработка визуального интерфейса приложения:



Table\_Tourney\_ATP2022 – выбор нужного турнира.

Add DB from disk – открыть базу данных.

Save DB to disk – сохранить базу данных.

Requests manager – ссылка на менеджер запросов.

TennisDB: Request manager ✕

Request: Find ATP 2022 Winner ▼

SelectJoinGroupFind ATP 2022 Winner

Execute selected request

Add new request

Remove selected request

Table1: Table\_Tourney\_ATP2022▼

Column: Players▼

Subrequest: None▼

Condition: Max score in matches▼


Execute selected request – выполнять выбранные запросы

Add new request – добавить новые запросы

Select - кнопка, формирующая запрос аналогичный запросу Select вSQL.

Join - кнопка, формирующая запрос аналогичный запросу Join вSQL.

Group - кнопка, формирующая запрос аналогичный запросу Group

#### 4. Создание диаграммы классов:

```

classDiagram
    class Player {
        +name : string
        +country : string
        +rank : integer
    }
    class PlayerStats {
        +ases : integer
        +doubleFaults : integer
        +score : integer
        +match : Matches
        +player : Player
    }
    class Matches {
        +winner : Player
        +player1 : Player
        +player2 : Player
        +quarter : Quarter
    }
    class Quarter {
        +qtrType : integer
        +tourney : Tourney
    }
    class Tourney {
        +name : string
        +place : string
        +season : Season
    }
    class Season {
        +startDate : time
        +endDate : time
        +name : string
    }
    class DataBaseContext {
        +Matches : DbSet<Matches>
        +Player : DbSet<Player>
        +PlayerStarts : DbSet<PlayerStarts>
        +Quarter : DbSet<Quarter>
        +Season : DbSet<Season>
        +Tourney : DbSet<Tourney>
        #OnModelCreating(ModelBuilder)
    }
    class MainWindowViewModel : ViewModelBase {
        +Matches : ObservableCollection<Matches>
        +Player : ObservableCollection<Player>
        +PlayerStarts : ObservableCollection<PlayerStarts>
        +Quarter : ObservableCollection<Quarter>
        +Season : ObservableCollection<Season>
        +Tourney : ObservableCollection<Tourney>
        +CreateRequests() : void
        +DeleteRequests() : void
    }

    Player --> PlayerStats
    Player --> Matches
    Player --> Quarter
    Player --> Tourney
    Player --> Season
    Player --> DataBaseContext
    PlayerStats --> Matches
    Matches --> Quarter
    Quarter --> Tourney
    Tourney --> Season
    DataBaseContext --> MainWindowViewModel
  
```

The diagram illustrates the relationships between various classes in the TennisDB application. The **Player** class is central, with associations to **PlayerStats**, **Matches**, **Quarter**, **Tourney**, **Season**, and **DataBaseContext**. **PlayerStats** is associated with **Matches**. **Matches** is associated with **Quarter**, which is associated with **Tourney**, which is associated with **Season**. The **DataBaseContext** class manages collections of all these entities and is associated with the **MainWindowViewModel**, which provides the user interface for interacting with the data.

## Классы:

### 1.Season

1.1 public time StartDate

1.2 public time EndDate

1.3 public string name

### 2.Tourney

2.1 public string name

2.2 public string place

2.3 public Season season

### 3.Quarter

3.1 public int qtrType

3.2 public Tourney tourney

### 4.Player

4.1 public string name

4.2 public string country

4.3 public int rank

### 5.Matches

5.1 public Player winner

5.2 public Player player1

5.3 public Player player2

5.4 public Quarter quarter

### 6.PlayerStarts

6.1 public int aces

6.2 public int doubleFaults

6.3 public int score

6.4 public Matches match



7. DatabaseContext – класс, который позволяет работать с базой данных через модели. Каждый DbSet представляет собой сущности в БД

7.1. public DbSet Players

7.2. public DbSet Matches

7.3. public DbSet PlayerStats

7.4. public DbSet Quarters

7.5. public DbSet Seasons

7.6. public DbSet Tourneys

7.7. protected void OnModelCreating(ModelBuilder modelBuilder)

8. MainWindowViewModel – здесь представлены все основные данные и методы для работы с View, каждая ObservableCollection – наблюдаемая коллекция, содержимое которых будет отображаться в окне программы, также они непосредственно связаны с моделями из DatabaseContext

8.1. public ObservableCollection Players

8.2. public ObservableCollection Matches

8.3. public ObservableCollection PlayerStats

8.4. public ObservableCollection Quarters

8.5. public ObservableCollection Seasons

8.6. public ObservableCollection Tourneys

8.7. public void SQLRequestOpen()

8.8. public void SQLRequestRun()

## 5. Реализация основного окна приложения:

Rylskiy\_Tennis

Match

Player

PlayerStat

Quarter

Season

Tourney

Id	Name	Country	Rank	Delete
1	John Isner	USA	27	Delete
2	Cameron Norrie	GBR	11	Delete

New Record

Create new request