

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

Выполнил: студент 2 курса, группы ИП-014

Малышев Владимир Александрович

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 - 15

15. Бейсбол MLB (<https://www.baseball-reference.com/teams/>)

# Предметная область

Тема: Бейсбол (MLB)

## 1. Бейсбол;

### 1.1 MLB(Major League Baseball);

### 1.2 Активные бейсбольные команды (Active Franchises);

#### 1.2.1 Состав команды;

#### 1.2.2 Топ-10 игроков за все время;

#### 1.2.3 Характеристики игроков команды;

#### 1.2.4 История турниров команды;

##### 1.3.4.1 Год проведения;

##### 1.3.4.2 Лига;

##### 1.3.4.3 Статистика команды за все матчи(G,W,L,Ties,W-L%, etc.);

#### 1.2.5 Показатели игроков в матче;

### 1.3 Неактивные бейсбольный команды (Inactive Franchises);

#### 1.3.1 Состав команды;

#### 1.3.2 Топ-10 игроков за все время;

#### 1.3.3 Характеристики игроков команды;

#### 1.3.4 История турниров команды;

##### 1.3.4.1 Год проведения;

##### 1.3.4.2 Лига;

##### 1.3.4.3 Статистика команды за все матчи(G,W,L,Ties,W-L%, etc.);

#### 1.3.5 Показатели игроков в матче;

### 1.4 Команды национальной ассоциации (National Association Franchises)

#### 1.4.1 Состав команды;

#### 1.4.2 Топ-10 игроков за все время;

#### 1.4.3 Характеристики игроков команды;

#### 1.4.4 История турниров команды;

##### 1.3.4.1 Год проведения;

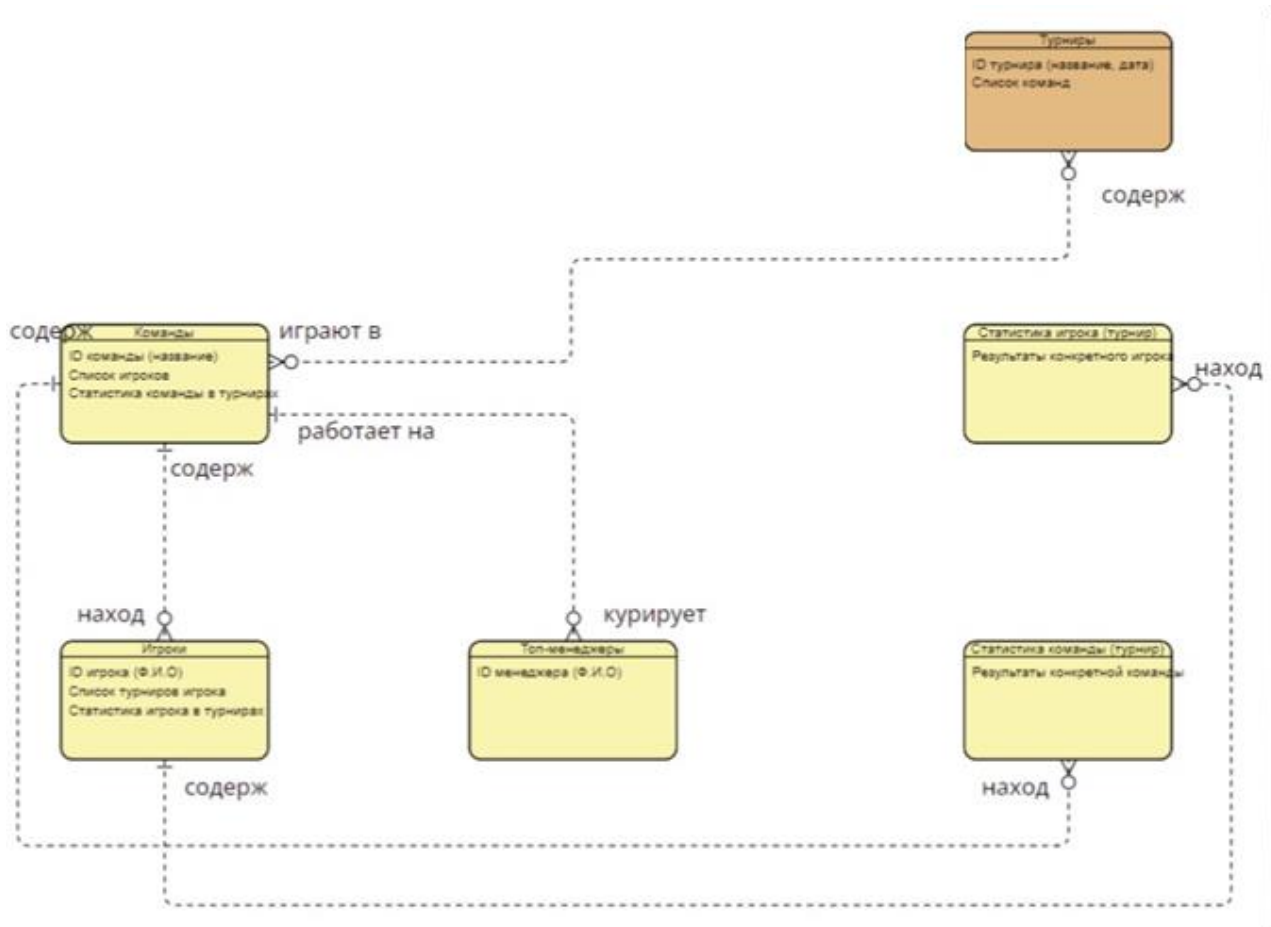
##### 1.3.4.2 Лига;

1.3.4.3 Статистика команды за все матчи(G,W,L,Ties,W-L%, etc.);

1.4.5 Показатели игроков в матче;

## ER-диаграмма

**Сущности:** команды, игроки, топ-менеджеры, турниры, статистика игрока за турнир, статистика команды за турнир.



## Часть 2

### Создание реляционной модели

#### 1. Players

1.1. id (PK)

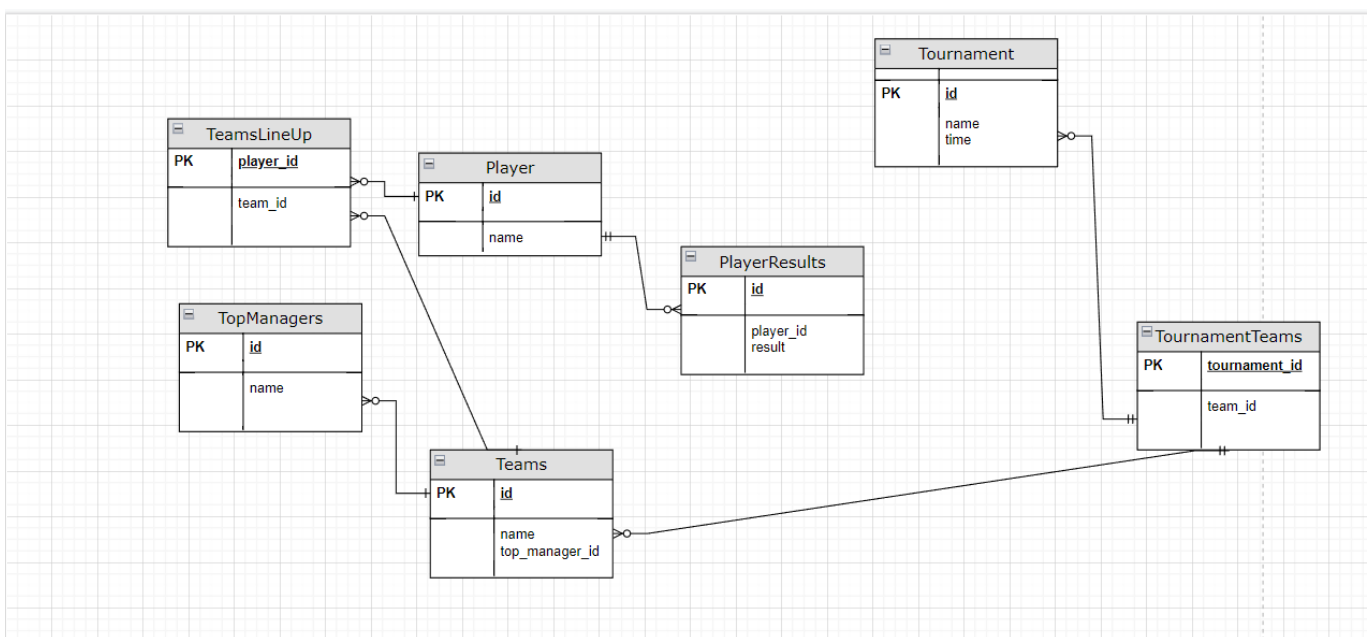
1.2. name (string)

#### 2. PlayersResults

2.1. id (PK)

2.2. player\_id (FK)

- 2.3. result (integer)
- 3. TopManagers
  - 3.1. id (PK)
  - 3.2. name (string)
- 4. Teams
  - 4.1. id (PK)
  - 4.2. name (string)
  - 4.3. top\_manager\_id (FK)
- 5. TeamsLineUp
  - 5.1. player\_id (PK, FK)
  - 5.2. team\_id (FK)
- 6. Tournament
  - 6.1. id (PK)
  - 6.2. name (string)
  - 6.3. time (datetime)
- 7. TournamentTeams
  - 7.1. tournament\_id (FK, UNIQUE)
  - 7.2. team\_id (FK, UNIQUE)



## Часть 3

# Проработка визуального интерфейса

## Окно инструментов:

**File** - кнопка, которая открывает вложенное окно, содержащее:

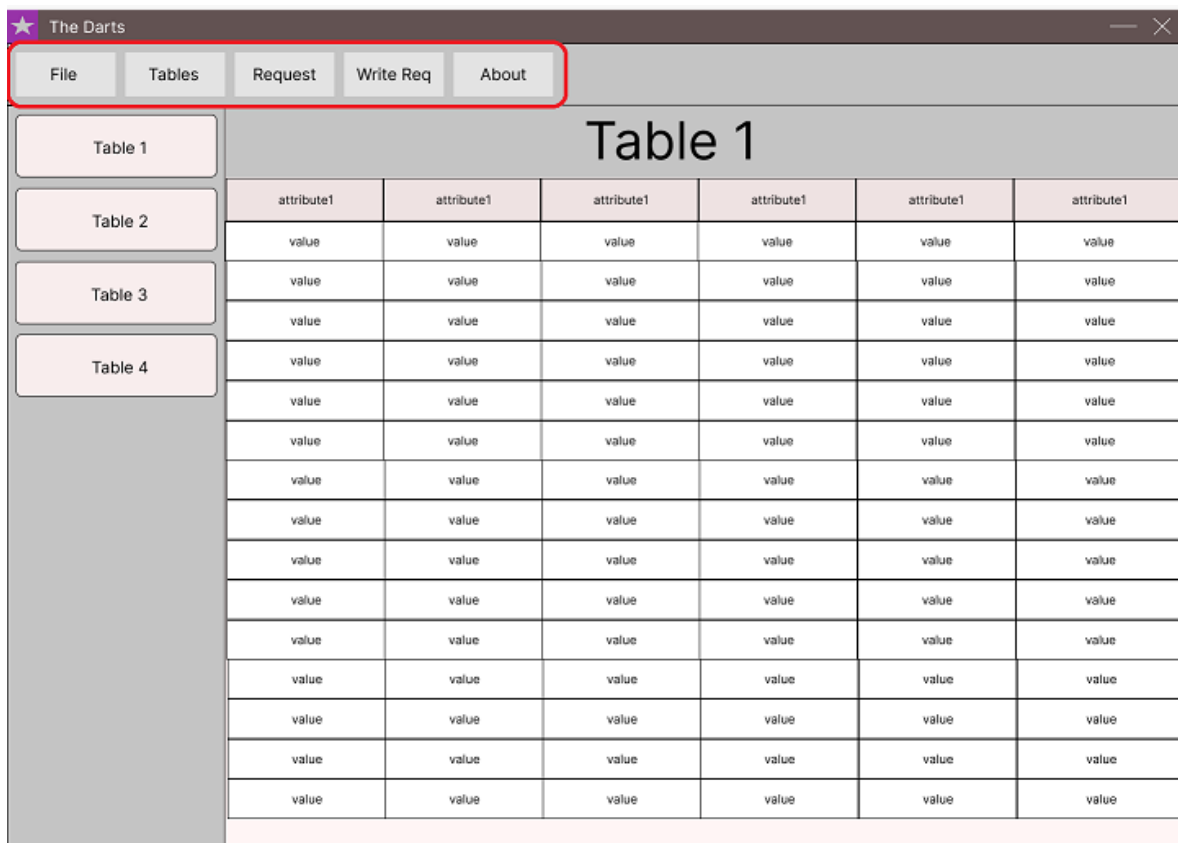
**Save** - кнопка, которая перезаписывает исходный файл базы данных.

**Load** - кнопка, которая загружает указанный файл в приложение и выводит все отношения, содержащиеся на данный момент в базе данных.

**Tables** - открытая по умолчанию вкладка, отображающая все отношения базы данных. Если пользователь находится во Write request, то при нажатии на кнопку Tables, пользователь будет возвращаться обратно ко всем отношениям базы данных.

**Write request** - кнопка, которая отвечает за переход к написанным нами SQL-запросам, в правой части приложения за место кнопок Table 1, 2, 3, ..., n, появятся Request 1, 2, 3, ..., n, За место атрибутов таблицы, будут выведены атрибуты запроса.

**About** - кнопка, которая открывает вложенное окно с карт. Описанием кнопок и информацией о создателе приложения.



## Панель отображения таблиц:

В этой панели отображаются все отношения базы данных, при переходе в на панели инструментов во вкладку Request, все названия отношения помещаются на название запросов.

[illegible]

## Панель отображения атрибутов:

Панель, в которой находятся все атрибуты каждого отношения. В каждом столбце атрибута, будет содержаться информация описанная в атрибуте. Столбцы можно будет скорллить как в лево, так и вправо с помощью скролл- полей.

★ The Darts

FileTablesRequestWrite ReqAbout

Table 1

Table 2

Table 3

Table 4

Table 1

attribute1	attribute1	attribute1	attribute1	attribute1	attribute1
value	value	value	value	value	value
value	value	value	value	value	value
value	value	value	value	value	value
value	value	value	value	value	value
value	value	value	value	value	value
value	value	value	value	value	value
value	value	value	value	value	value
value	value	value	value	value	value
value	value	value	value	value	value
value	value	value	value	value	value
value	value	value	value	value	value
value	value	value	value	value	value
value	value	value	value	value	value
value	value	value	value	value	value
value	value	value	value	value	value

## Окно написания запросов содержит:



## Панель инструментов:

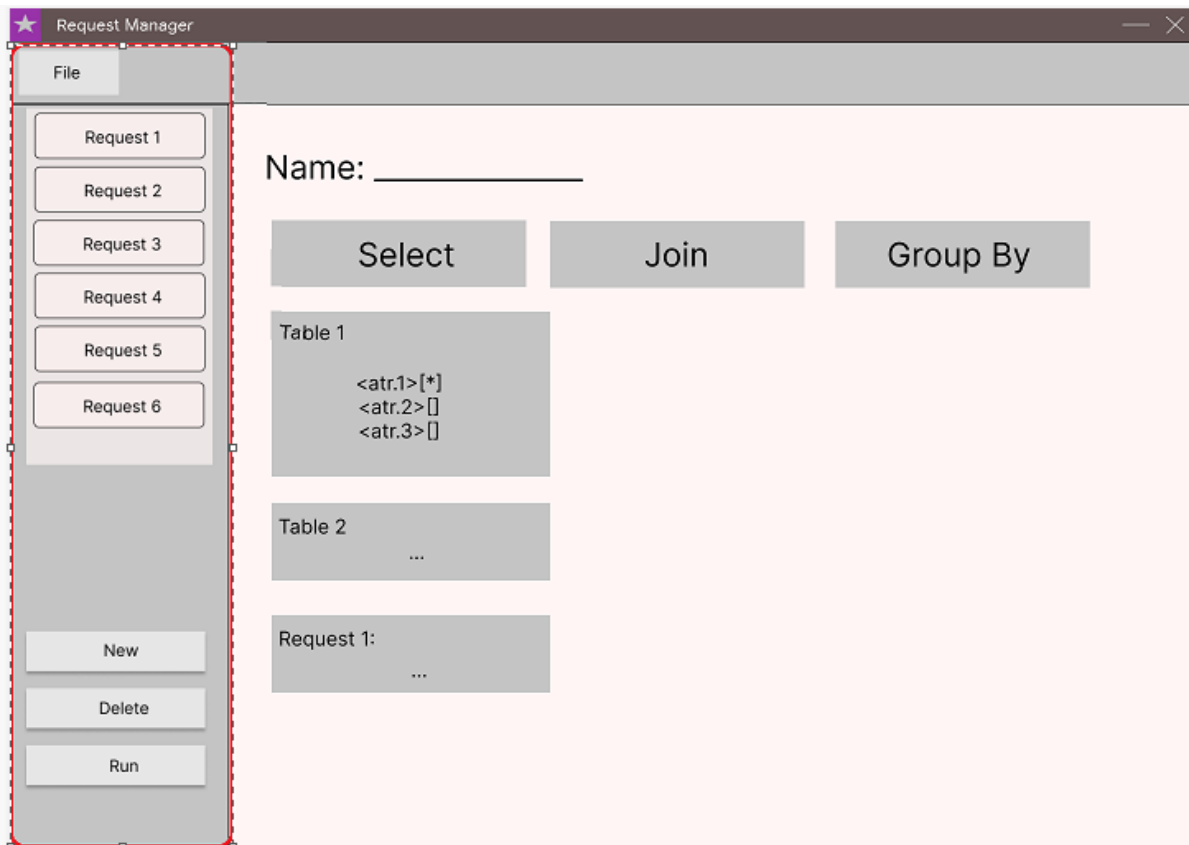
**File** - некликабельное поле, отвечающее за панель взаимодействия с файлом.

**Список кнопок с запросами** - набор кнопок, который отвечает за переключение между запросами. Отображает название запроса и то, что в нем содержится.

**New** - кнопка, отвечающая за создание нового запроса.

**Delete** - кнопка, отвечающая за удаление уже существующего запроса.

**Run** - кнопка, отвечающая за запуск запроса и формирование отношения.



## Окно написания запроса:

Поле Name отвечает за название запроса.

**Select** - кнопка, формирующая запрос аналогичный запросу Select вSQL.

**Join** - кнопка, формирующая запрос аналогичный запросу Join вSQL.

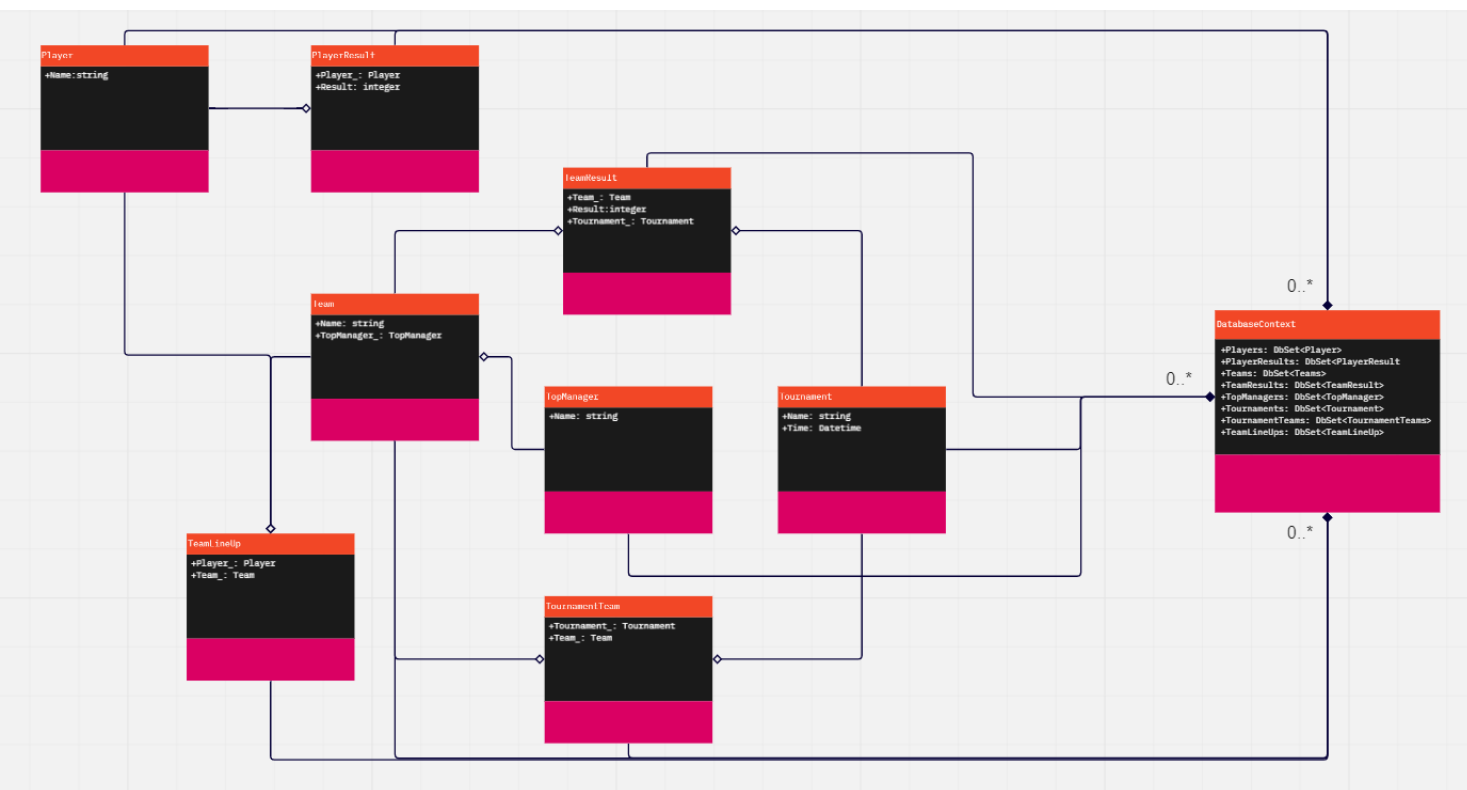
**Group by** - кнопка, формирующая запрос аналогичный запросу Group by в SQL.

Представленные ниже поля будут отвечать за введения атрибутов для взаимодействия в запросе. А также из каких таблиц их брать. Поля с Request будут отвечать за подзапросы.

The screenshot shows a window titled "Request Manager" with a menu bar containing "File". On the left side, there is a vertical list of buttons labeled "Request 1" through "Request 6", and below them, "New", "Delete", and "Run" buttons. The main area of the window is a light pink form with a red border. At the top of the form is a label "Name:" followed by a text input field. Below this are three buttons: "Select", "Join", and "Group By". Under the "Select" button, there are three input fields labeled "Table 1", "Table 2", and "Request 1:". The "Table 1" field contains the text "<atr.1>[\*]", "<atr.2>[]", and "<atr.3>[]". The "Table 2" field contains "...". The "Request 1:" field contains "...".

## Часть 4

# Создание диаграммы классов



## Классы:

1. Player
  - 1.1. public string Name
2. PlayerResult
  - 2.1. public Player Player\_
  - 2.2. public int Result
3. Team
  - 3.1. public string Name
  - 3.2. public TopManager TopManager\_
4. TopManager
  - 4.1. public string Name
5. TeamResult
  - 5.1. public Team Team\_
  - 5.2. public int Result
  - 5.3. public Tournament Tournament\_
6. Tournament
  - 6.1. public string Name
  - 6.2. public DateTime Time
7. TournamentTeam
  - 7.1. public Tournament Tournament\_
  - 7.2. public Team Team\_

## 8. TeamLineUp

8.1. public Player Player\_

8.2. public Team Team\_

## 9. DatabaseContext

9.1. public DbSet<Player> Players

9.2. public DbSet<PlayerResult> PlayerResults

9.3. public DbSet<Team> Teams

9.4. public DbSet<TopManager> TopManagers

9.5. public DbSet<TeamResult> TeamResults

9.6. public DbSet<Tournament> Tournaments

9.7. public DbSet<TournamentTeam> TournamentTeams

9.8. public DbSet<TeamLineUp> TeamLineUps