



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

НА ТЕМУ:

*«Сравнительный анализ данных об игроках и
результатах матчей НХЛ»*

Студент	<u>ИУ7-61Б</u>	_____	<u>М.В.Ефимова</u>
	(Группа)	(Подпись, дата)	(И.О.Фамилия)

Руководитель проекта	_____	<u>А.С. Григорьев</u>
	(Подпись, дата)	(И.О.Фамилия)

Москва, 2022 г.

РЕФЕРАТ

Курсовая работа представляет собой реализацию базы данных и приложения, содержащего информацию о статистике клубов Национальной Хоккейной Лиги и о статистике игроков, а также позволяющего пользователю отслеживать и сравнивать статистику клубов и хоккеистов. Готовое приложение позволяет создавать список избранных игроков и клубов и отслеживать его, а также отслеживать текущие шансы каждой команды на выход в плей-офф.

Приложение реализовано на языке программирования Python в среде разработки Visual Studio Code. Взаимодействие с приложением производится через консоль. В качестве СУБД был выбран PostgreSQL.

Ключевые слова: НХЛ, команды, хоккеисты, список избранного, Python, PostgreSQL.

Расчетно-пояснительная записка к курсовой работе содержит 56 страниц, 38 иллюстраций, 2 таблицы.

СОДЕРЖАНИЕ

РЕФЕРАТ.....	2
ВВЕДЕНИЕ	5
1 Аналитическая часть	6
1.1 Обзор существующих аналогов	6
1.1.1 NHL	6
1.1.2 Flashscore	7
1.1.3 EliteProspects	8
1.2 Формализация задачи.....	9
1.3 Формализация данных	10
1.4 Формализация ролей	11
1.5 Описание существующих СУБД.....	13
1.5.1 Классификация СУБД по модели данных	14
1.6 Выводы из аналитического раздела.....	17
2 Конструкторская часть	18
2.1 Проектирование базы данных	18
2.1.1 Таблицы	18
2.1.2 Целостность данных.....	21
2.1.3 Триггеры	23
2.2 Требования к программе.....	25
2.3 Выводы из конструкторского раздела.....	26
3 Технологическая часть	27
3.1 Средства реализации	27
3.2 Детали реализации.....	28
3.2.1 Создание таблиц	28
3.2.2 Используемые функции и триггеры	30
3.3 Взаимодействие с приложением	38

3.4 Выводы из технологического раздела	43
4 Экспериментальная часть	44
4.1 Технические характеристики	44
4.2 Анализ зависимости	44
4.2 Выводы из экспериментального раздела	53
ЗАКЛЮЧЕНИЕ	54
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	55
ПРИЛОЖЕНИЕ А.....	56

ВВЕДЕНИЕ

В настоящее время в Национальной Хоккейной Лиге более 600 действующих хоккеистов и 32 хоккейных клуба, которые ежедневно набирают очки в течение проходящих матчей, а также другую статистику. Эту статистику необходимо учитывать, а также было бы удобно не только следить за успехами любимых хоккеистов или клубов, но и иметь возможность сравнивать ее между собой.

Цель данной работы – создание базы данных для хранения данных о результативности игроков и о статистике команд НХЛ по результатам прошедших матчей, разработка приложения для регистрации и обновления статистики команд и игроков и построения статистических отчетов по введенным данным для выявления шансов каждой команды на выход в плей-офф в текущий момент времени.

Чтобы достигнуть поставленной цели, требуется решить следующие задачи:

- 1) формализовать задание, определить необходимый функционал;
- 2) провести анализ СУБД;
- 3) описать структуру базы данных, включая объекты, из которых она состоит;
- 4) спроектировать приложение для доступа к БД;
- 5) создать и заполнить БД;
- 6) разработать программное обеспечение, которое позволит пользователю получать, искать и сравнивать статистику игроков и команд Национальной Хоккейной Лиги;
- 7) исследовать зависимость шанса на выход в плей-офф команд от их статистики в регулярном чемпионате и количестве сыгранных матчей.

1 Аналитическая часть

В данном разделе будет проведен анализ существующих аналогов, схожих по функциональности с создаваемым приложением, проанализирована поставленная задача и рассмотрены различные способы ее реализации. Также будет проведена формализация данных и ролей, представлена диаграмма сущностей и диаграмма вариантов использования и проведен сравнительный анализ СУБД по модели данных.

1.1 Обзор существующих аналогов

В настоящий момент существует несколько сервисов для просмотра статистики хоккеистов и команд НХЛ, но у каждого сервиса есть недостатки. Большинство из сервисов предоставляют возможность только просмотра статистики, не давая возможности ее сравнить, но предоставляя возможность добавлять в избранное хоккеистов и команды, или только команды. Подробное описание нескольких существующих аналогов представлено в следующих разделах.

1.1.1 NHL

Данный сайт предоставляет удобный интерфейс для просмотра статистики хоккеистов как по всей лиге, так и отдельно по командам, а также статистики самих команд [1]. Он включает в себя возможность добавления команды в избранное, поиска игроков, перехода по клику на страницу игрока, содержащую только его статистику.

Однако на данном сайте отсутствует возможность добавлять игроков в избранное, а также сравнивать статистику игроков и команд между собой. Отсутствует и возможность просмотра шансов на выход команд в плей-офф.

Интерфейс сайта представлен на рисунке 1.1.

NHL

Scores

News

Video

Playoffs

Stats

Schedule

Players

App

Tickets

Fantasy

Shop

Teams

AR Cup

Stream Games

EN

BY SEASON

BY GAME

ALL-TIME

Legend

Print

SEASONS

2021-22

-

2021-22

SUM RESULTS

GAME TYPE

Playoffs

FRANCHISE

All Franchises

POSITION

All Skaters

REPORT

Summary

Search Players

Get Stats

More Filters

Clear Filters

Summary

Export

①	Player	Season	Team	S/C	Pos	GP	G	A	P	+/-	PIM	P/GP	EVG	EVP	PPG	PPP	SHG	SHP	OTG	GWG	S	S%	TOI/GP	FOW%
1	Connor McDavid	2021-22	EDM	L	C	16	10	23	33	+15	10	2.06	8	24	2	8	0	1	1	1	61	16.4	23:02	52.6
2	Leon Draisaitl	2021-22	EDM	L	C	16	7	25	32	+4	6	2.00	3	22	3	9	1	1	0	1	44	15.9	21:22	48.6
3	Cale Makar	2021-22	COL	R	D	17	7	19	26	+10	6	1.53	4	14	2	11	1	1	1	1	60	11.7	26:43	--
4	Nikita Kucherov	2021-22	TBL	L	R	20	7	19	26	+5	14	1.30	2	16	5	10	0	0	0	1	67	10.5	20:34	--
5	Mika Zibanejad	2021-22	NYR	R	C	20	10	14	24	0	4	1.20	4	12	6	11	0	1	0	1	63	15.9	20:51	51.0
6	Mikko Rantanen	2021-22	COL	L	R	17	5	19	24	+3	4	1.41	4	13	1	11	0	0	0	0	50	10.0	19:49	37.5
7	Adam Fox	2021-22	NYR	R	D	20	5	18	23	0	2	1.15	3	11	2	12	0	0	0	1	42	11.9	26:38	--
8	Gabriel Landeskog	2021-22	COL	L	L	17	11	10	21	+14	6	1.24	6	12	5	9	0	0	0	1	50	22.0	20:36	58.7

Рисунок 1.1 – Интерфейс сайта NHL

1.1.2 Flashscore

Данный сайт предоставляет возможность просмотра статистики команд, но статистику хоккеистов можно посмотреть только по одному [2]. В отличие от предыдущего сервиса, на данном сайте имеется возможность добавлять в избранное не только команды, но и отдельных игроков. Также присутствует возможность поиска команд и игроков.

Основным недостатком является отсутствующая возможность сравнить команды и игроков между собой, посмотреть статистику сразу всех игроков, а также увидеть шансы команд на выход в плей-офф.

Интерфейс сайта представлен на рисунке 1.2.

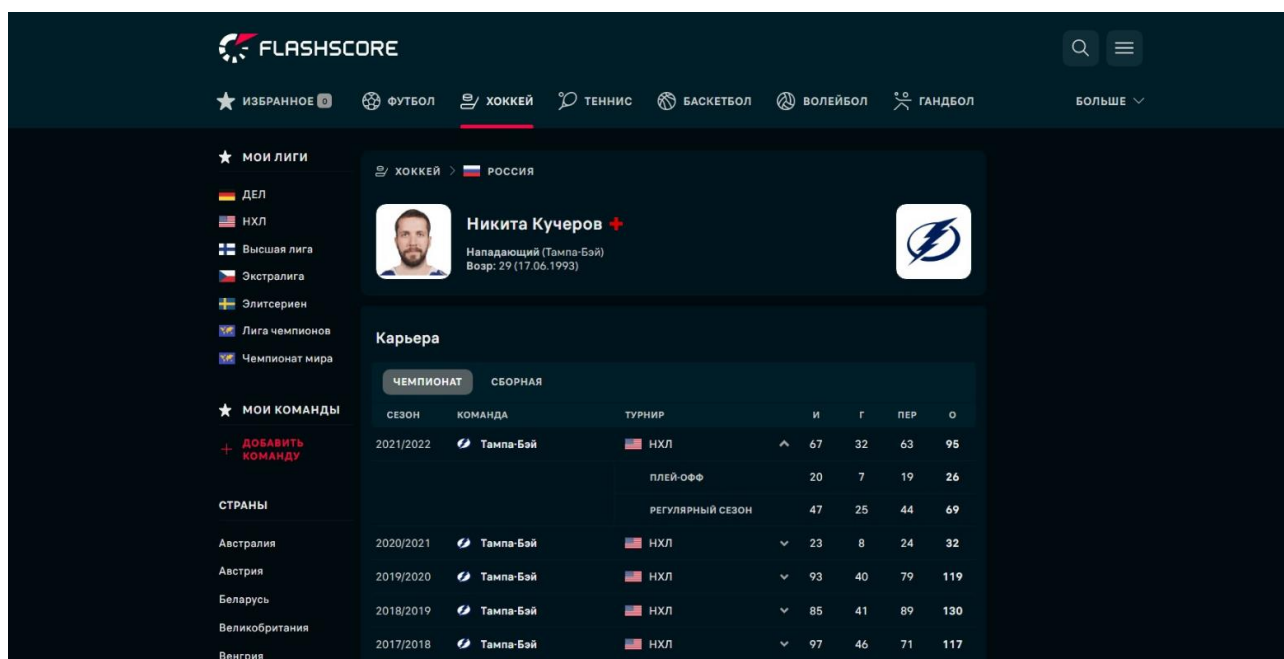


Рисунок 1.2 – Интерфейс сайта Flashscore

1.1.3 EliteProspects

Данный сайт имеет удобный интерфейс для просмотра информации и статистики об игроках и командах, предоставляет возможность поиска игроков и команд, просмотра как всех игроков лиги, так и отдельно по командам, также имеется возможность посмотреть информацию о персонале команд, прослушать произношение имен хоккеистов [3].

В отличие от предыдущих сервисов, на данном сайте возможность добавить команду или игрока в избранное отсутствует. Также нет возможности сравнить статистику игроков и команд, узнать шансы команд на выход в плей-офф.

Интерфейс сайта представлен на рисунке 1.3.

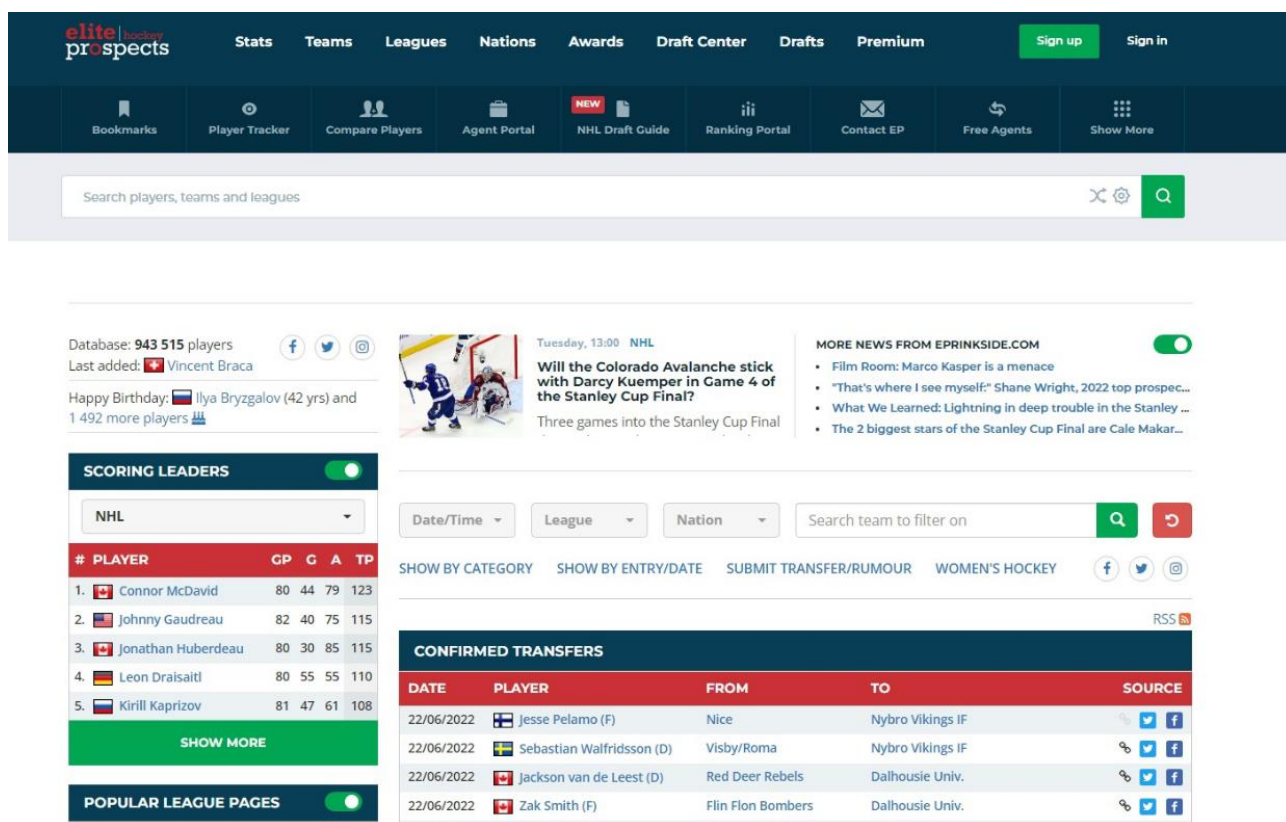


Рисунок 1.3 – Интерфейс сайта EliteProspects

1.2 Формализация задачи

В ходе выполнения курсовой работы необходимо разработать программу для отображения статистики игроков и команд НХЛ, хранящую информацию о личной статистике каждого игрока (голы, передачи, очки, плюс/минус и т.д.), статистике каждой команды (сыгранные матчи, очки, разница забитых и пропущенных шайб и т.д.). Также пользователь должен иметь возможность формировать персональный список любимых хоккеистов и любимых команд для дальнейшего быстрого просмотра информации о них, то есть добавлять игроков и команды в закладки. Также у пользователя должна иметься возможность просмотреть шансы команд на выход в плей-офф в текущий момент времени.

1.3 Формализация данных

База данных должна хранить информацию о:

- командах;
- хоккеистах;
- пользователях и их избранном списке хоккеистов и команд.

Сведения о каждой категории данных предоставлены в таблице 1.1.

Таблица 1.1 – категории и сведения о данных

Категория	Сведения
Команда	Id, название, страна, количество сыгранных матчей, количество побед в основное время и по буллитам/в овертайме, количество поражений, очки, разница шайб
Хоккеисты	Id, имя, фамилия, дата рождения, место рождения, представляемая страна, рост, вес, хват, игровой номер, команда, количество очков, количество голов, количество передач, количество сыгранных матчей, плюс/минус
Пользователь	Id, имя, адрес электронной почты, страна, год рождения, логин, пароль, права доступа
Личный список игроков	Id, id пользователя, id хоккеиста
Личный список команд	Id, id пользователя, id команды

На рисунке 1.4 приведена ER-диаграмма схемы сущностей.

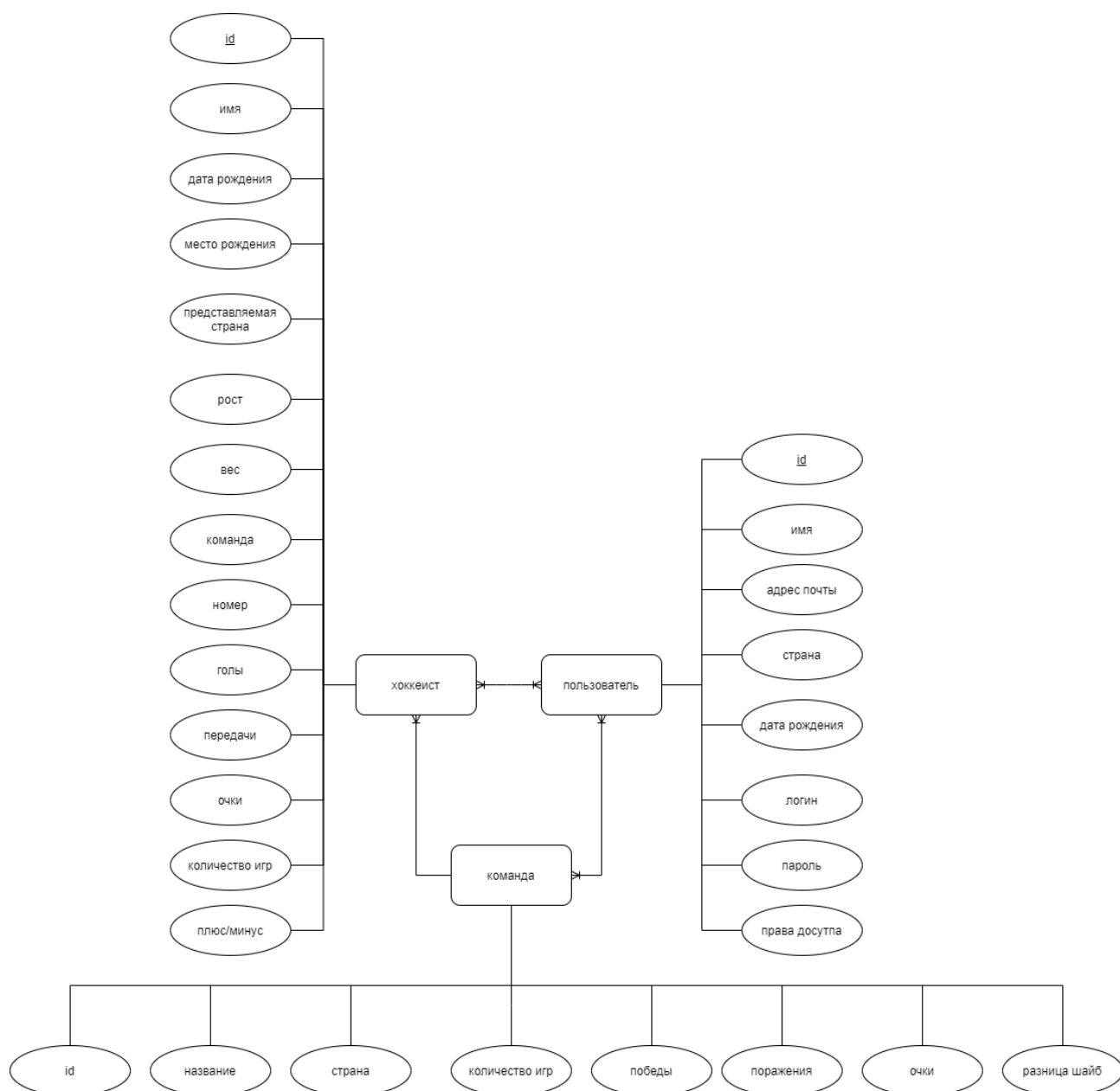


Рисунок 1.4 – ER-диаграмма сущностей нотации Чена

1.4 Формализация ролей

В данном курсовом проекте предусмотрено наличие двух типов пользователей: авторизованных и неавторизованных, так как для составления личного списка хоккеистов и команд необходима авторизация. Неавторизованные пользователи могут войти или зарегистрироваться. Авторизованные пользователи могут составлять личный список хоккеистов и команд и изменять его, а также пользоваться всеми возможностями приложения.

Для управления статистикой команд и хоккеистов, а также правами доступа пользователей предусмотрена роль администратора.

Функционал каждой роли представлен в таблице 1.2.

Таблица 1.2 – типы пользователей и их функционал

Тип пользователя	Функционал
Неавторизованный пользователь	Регистрация, авторизация
Авторизованный пользователь	Просмотр статистики игроков и команд, поиск игроков и команд, сравнение игроков и команд, составление и редактирования личного списка игроков и команд, просмотр шансов команд на выход в плей-офф.
Модератор	<p>Просмотр статистики игроков и команд, поиск игроков и команд, сравнение игроков и команд, составление и редактирования личного списка игроков и команд, просмотр шансов команд на выход в плей-офф.</p> <p>Добавление игроков и команд, удаление игроков и команд.</p>
Администратор	<p>Просмотр статистики игроков и команд, поиск игроков и команд, сравнение игроков и команд, составление и редактирования личного списка игроков и команд, просмотр шансов команд на выход в плей-офф.</p> <p>Добавление игроков и команд, удаление игроков и команд.</p> <p>Изменение прав доступа пользователей, удаление пользователей.</p>

На рисунке 1.5 приведена Use-Case-диаграмма сущностей.

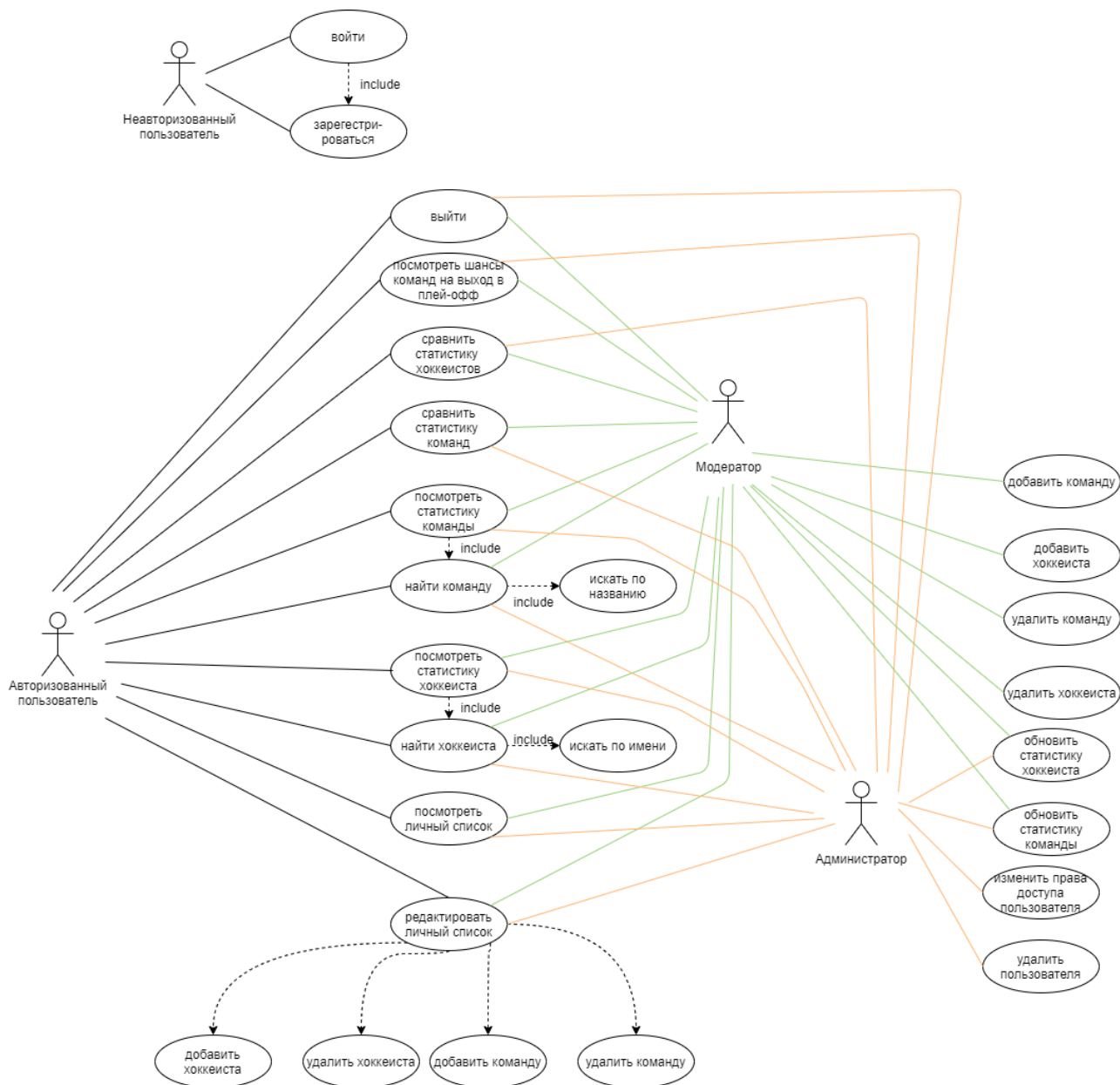


Рисунок 1.5 – Use-Case-диаграмма сущностей

1.5 Описание существующих СУБД

СУБД (система управления базами данных) — совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных [4].

Основными функциями СУБД являются:

- управление данными во внешней памяти;

- управление данными в оперативной памяти с использованием дискового кэша;
- журнализация изменений, резервное копирование и восстановление базы данных после сбоев;
- поддержка языков БД.

1.5.1 Классификация СУБД по модели данных

Модель данных — это абстрактное, самодостаточное, логическое определение объектов, операторов и прочих элементов, в совокупности составляющих абстрактную машину доступа к данным, с которой взаимодействует пользователь. Эти объекты позволяют моделировать структуру данных, а операторы — поведение данных [5].

Существует 4 основных типа моделей организации данных:

- иерархическая;
- сетевая;
- реляционная;
- функциональная.

В иерархической модели данных используется представление базы данных в виде древовидной структуры, состоящей из объектов разных уровней. Среди объектов существуют связи, любой объект может включать в себя несколько объектов более низкого уровня. Такие объекты находятся в отношении предка к потомку, при этом вероятна ситуация, когда объект-предок имеет несколько потомков, тогда как у объекта-потомка обязательно должен быть только один предок.

Структура иерархической модели данных представлена на рисунке 1.6.

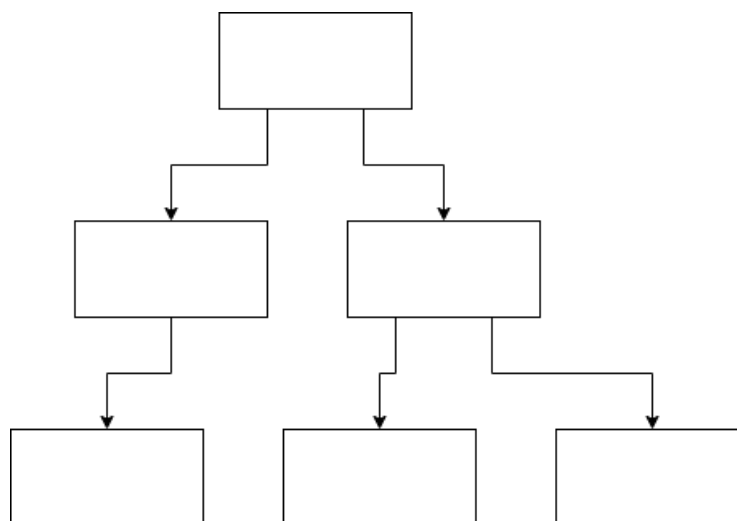


Рисунок 1.6 – структура иерархической модели данных

В сетевой модели данных, в отличие от иерархической, у потомка может быть любое число предков. Сетевая база данных состоит из набора экземпляров определенного типа записи и набора экземпляров определенного типа связей между этими записями [6].

Главным недостатком сетевой модели данных являются жесткость и высокая сложность схемы базы данных, построенной на основе этой модели. Так как логика процедуры выбора данных зависит от физической организации этих данных, то эта модель не является полностью независимой от приложения. Если придется изменить структуру данных, то нужно будет изменять и приложение.

Структура сетевой модели данных представлена на рисунке 1.7.

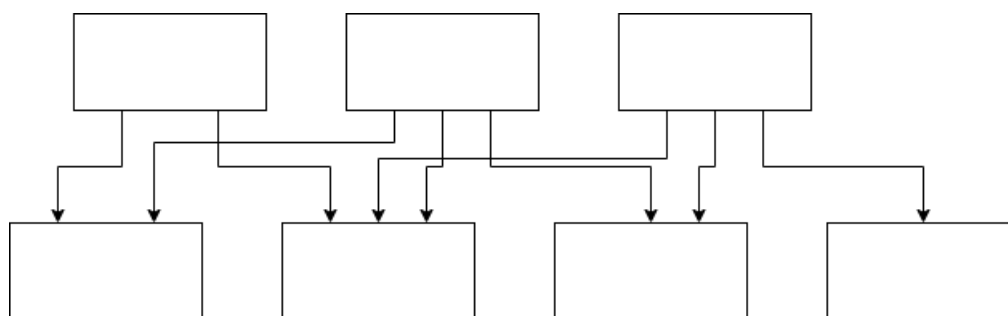


Рисунок 1.7 – структура сетевой модели данных

Реляционная модель данных является совокупностью данных и состоит из набора двумерных таблиц [7]. При табличной организации отсутствует иерархия элементов. Таблицы состоят из строк – записей и столбцов – полей. На пересечении строк и столбцов находятся конкретные значения. Для каждого поля определяется множество его значений. За счет возможности просмотра строк и столбцов в любом порядке достигается гибкость выбора подмножества элементов.

Реляционная модель является удобной и наиболее широко используемой формой представления данных.

Структура реляционной модели данных представлена на рисунке 1.8.



Рисунок 1.8 – структура реляционной модели данных

Наиболее популярными реляционными СУБД являются Oracle, Microsoft SQL Server и PostgreSQL.

Функциональные базы данных используются для решения аналитических задач, таких как финансовое моделирование и управление производительностью. Вместо того чтобы представлять объект записью с определенным содержанием или же кортежем в дереве, функциональная модель сообщает, какие функции (или операции) определены на этом объекте. Представление объекта — это дело реализации, и оно определяется на более низком уровне абстракции.

1.6 Выводы из аналитического раздела

В данном разделе был проведен обзор существующих аналогов, разобраны предоставляемые ими возможности и их недостатки, были формализованы задача, данные и роли. Также были представлены диаграммы компонентов и варианты использования. Были рассмотрены разные типы СУБД. В качестве используемой в данной работе была выбрана реляционная СУБД, так как она наиболее удобна в использовании при работе с таблицами и не имеет существенных недостатков, влияющих на сложность разработки программного обеспечения в данной работе.

2 Конструкторская часть

В данном разделе будет спроектированы база данных и приложение, а именно, будут описаны таблицы базы данных и их компоненты, выделены первичные ключи, рассмотрены требования к программе и каждому типу пользователей. Представлена диаграмма компонентов. Также будут описаны функционалы используемых в программе триггеров и представлены их.

2.1 Проектирование базы данных

2.1.1 Таблицы

База данных должна хранить рассмотренные в таблице 1.1 данные. В соответствии с этой таблицей можно выделить следующие таблицы:

- таблица команд Teams;
- таблица хоккеистов Players;
- таблица пользователей User;
- таблица хоккеистов, добавленных в избранное пользователями FavPlayers;
- таблица команд, добавленных в избранное пользователями FavTeams.

Таблица Teams должна хранить информацию о командах:

- id – уникальный идентификатор команды, PRIMARY KEY, INT;
- name – название команды, VARCHAR(50);
- country – страна команды, VARCHAR(50);
- games – количество сыгранных матчей, INT;
- wins – количество побед, INT;
- loses – количество поражений, INT;
- points – количество очков, INT;
- gdifference – разница забитых и пропущенных шайб, INT.

Таблица Players должна хранить информацию о хоккеистах:

- id – уникальный идентификатор игрока, PRIMARY KEY, INT;
- name – полное имя игрока, VARCHAR(50);
- number – игровой номер, INT;
- team_id – идентификатор команды, за которую играет хоккеист, внешний ключ (таблица Teams, поле id), FOREIGN KEY, INT;
- birth_year – год рождения, INT;
- birth_place – место рождения, VARCHAR(50);
- country – представляемая страна, VARCHAR(50);
- shoots – хват, VARCHAR(10);
- height – рост, INT;
- weight – вес, INT;
- points – количество очков, INT;
- goals – количество забитых голов, INT;
- assists – количество сделанных передач, INT;
- games – количество сыгранных матчей, INT;
- plusminus – полезность игрока (плюс/минус), INT.

Таблица Users должна хранить информацию о пользователях:

- id – уникальный идентификатор пользователя, PRIMARY KEY, INT;
- name – имя пользователя, VARCHAR(50);
- country – страна, VARCHAR(50);
- mail – почта пользователя, VARCHAR(50);
- birth_year – год рождения, INT;
- login – логин, VARCHAR(50) UNIQUE;
- password – пароль, VARCHAR(50);
- type – права доступа пользователя (0 – обычный пользователь, 1 – администратор, 2 – модератор), INT.

Таблица FavPlayers должна хранить информацию о хоккеистах, добавленных пользователями в личный список:

- id – уникальный идентификатор записи в личном списке, PRIMARY KEY, INT;
- usr_id – идентификатор пользователя, которому принадлежит запись в личном списке, внешний ключ (таблица Users, поле id), FOREIGN KEY, INT;
- player_id – идентификатор хоккеиста, которого добавили в личный список, внешний ключ (таблица Players, поле id), FOREIGN KEY, INT;

Таблица FavTeams должна хранить информацию о командах, добавленных пользователями в личный список:

- id – уникальный идентификатор записи в личном списке, PRIMARY KEY, INT;
- usr_id – идентификатор пользователя, которому принадлежит запись в личном списке, внешний ключ (таблица Users, поле id), FOREIGN KEY, INT;
- team_id – идентификатор команды, которую добавили в личный список, внешний ключ (таблица Teams, поле id), FOREIGN KEY, INT;

Диаграмма разрабатываемой базы данных представлена на рисунке 2.1.

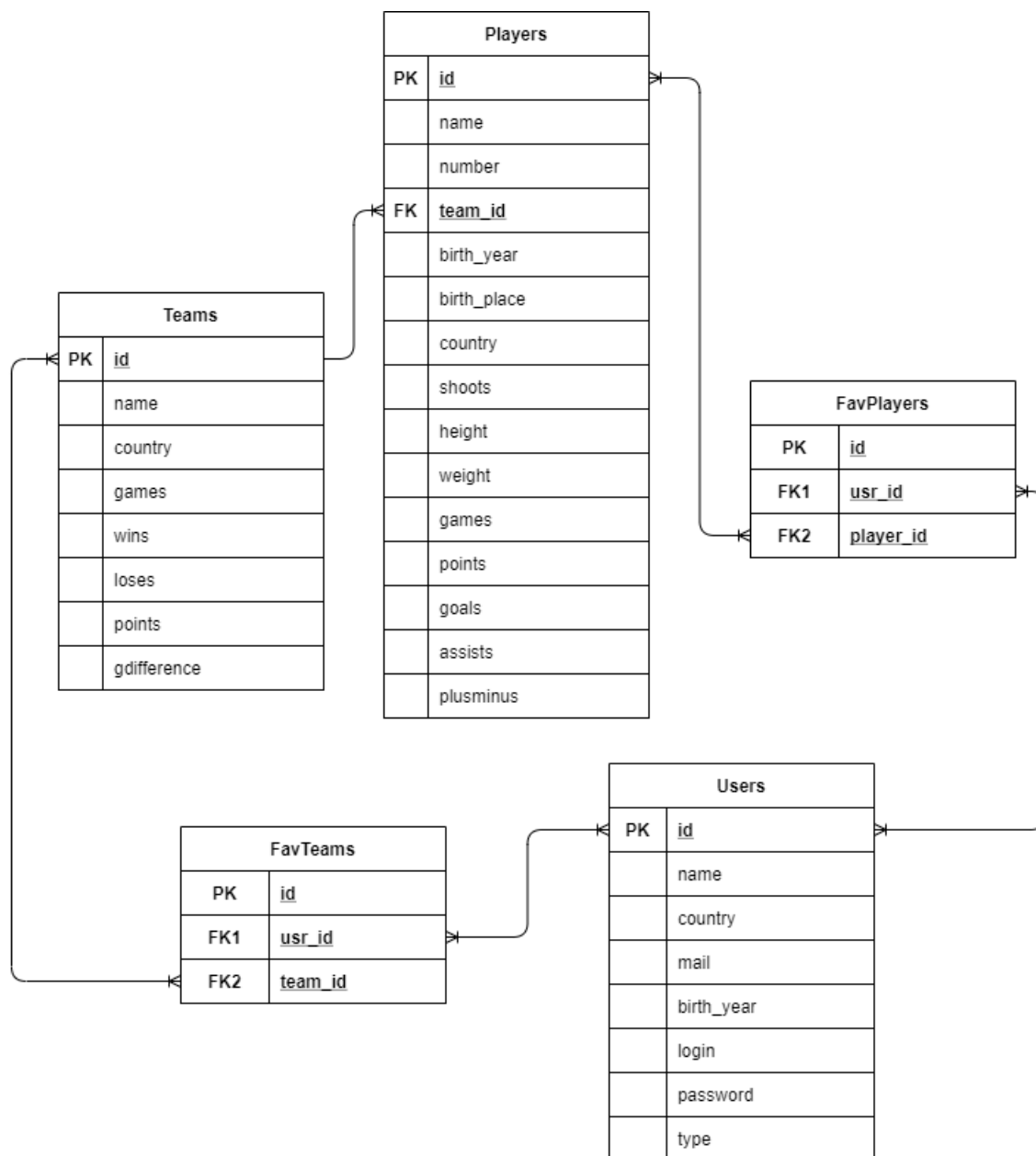


Рисунок 2.1 – ER-диаграмма

2.1.2 Целостность данных

Для обеспечения целостности таблиц необходимо, чтобы все строки в таблице имели уникальный идентификатор (первичный ключ).

Как было показано в предыдущем разделе, каждая таблица имеет первичный ключ:

1. у таблицы Teams – поле id;
2. у таблицы Players – поле id;
3. у таблицы Users – поле id;

4. у таблицы FavPlayers – поле id;
5. у таблицы FavTeams – поле id.

Для обеспечения ссылочной целостности необходимы внешние ключи, которые обеспечивают согласованное состояние между таблицей, на которую ссылаются, и таблицей, которая ссылается на другую.

Между таблицами Teams и Players существует связь "один ко многим", для ее формализации создается один внешний ключ: поле team_id таблицы Players ссылается на первичный ключ таблицы Teams (поле id).

Между таблицами Users и FavPlayers существует связь "многие ко многим", для ее формализации создаются два внешних ключа: поля usr_id и players_id таблицы FavPlayers ссылаются на первичные ключи таблиц Users и Players соответственно (поля id).

Между таблицами Users и FavTeams существует связь "многие ко многим", для ее формализации создаются два внешних ключа: поля usr_id и teams_id таблицы FavTeams ссылаются на первичные ключи таблиц Users и Teams соответственно (поля id).

В связи с тем, что при удалении строк в основных таблицах Teams, Players, Users могут иметься внешние ключи, которые ссылаются на данные строки, что означает, что строки не смогут быть удалены, пока не уничтожится связь, были созданы триггеры на удаление данных для каждой таблицы.

Для обеспечения целостности полей необходимо указать набор значений данных, которые являются допустимыми для поля, и определить, возможно ли использование значения NULL.

В таблице Teams для полей name, games, wins, loses, points, gdifference недопустимо значение NULL, поле country может принимать значение NULL.

В таблице Players для полей name, team_id, birth_year, height, weight, points, goals, assists, games, number, plusminus недопустимо значение NULL, поля birth_place, country, shoots могут принимать значение NULL.

В таблице Users для полей name, birth_year, mail, login, password, type недопустимо значение NULL, поле country может принимать значение NULL. Поле login также должно быть уникальным (в связи с тем, что авторизация пользователя осуществляется по его логину). Поле type может принимать только значения 0, 1, 2.

В таблицах FavPlayers и FavTeams для всех полей недопустимо значение NULL.

2.1.3 Триггеры

Триггер – это особая разновидность хранимой процедуры, которая автоматически выполняется при возникновении события на сервере базы данных, поэтому с их помощью можно успешно реализовать определенные действия.

Для данной работы было создано три триггера на удаление, каждый из которых удаляет записи, которые имеют связь с записями основных таблиц, так как для того, чтобы избежать ошибки, необходимо следить за корректным удалением данных. Это и осуществляют триггеры.

Триггер для таблицы Teams обеспечивает удаление из таблицы FavTeams всех записей, в которых участвует удаляемая команда.

Триггер для таблицы Players обеспечивает удаление из таблицы FavPlayers всех записей, в которых участвует удаляемый хоккеист.

Триггер для таблицы Users обеспечивает удаление всего списка избранного пользователя, то есть удаление из таблиц FavPlayers и FavTeams всех записей, которые добавил туда пользователь.

Схемы триггеров для основных таблиц приведены на рисунках 2.2-2.4.

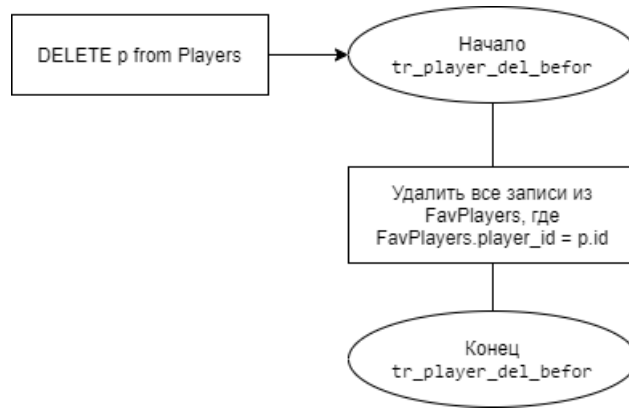


Рисунок 2.2 – схема триггера на удаление записей, связанных с хоккеистом, из другой таблицы

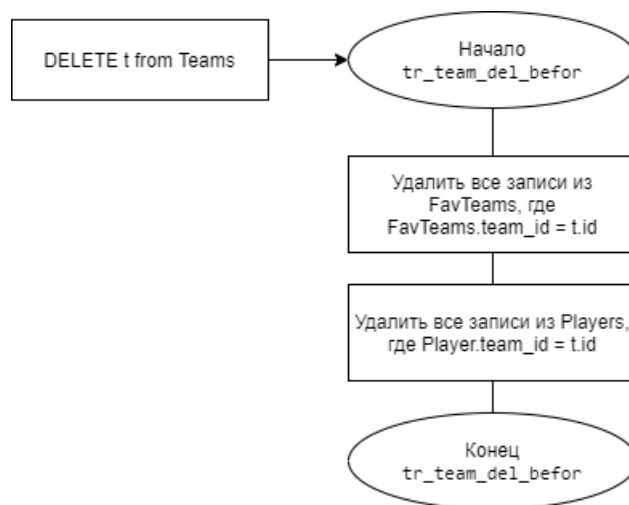


Рисунок 2.3 – схема триггера на удаление записей, связанных с командой, из других таблиц

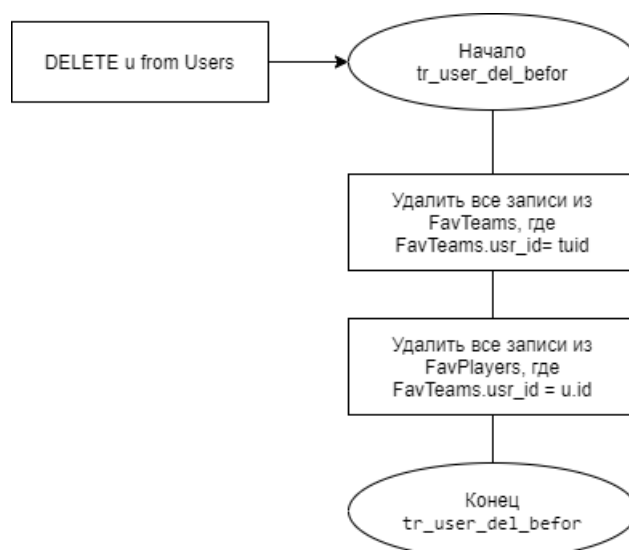


Рисунок 2.4 – схема триггера на удаление записей, связанных с пользователем,
из других таблиц

2.2 Требования к программе

Для успешной реализации поставленной задачи в соответствии с представленной ранее Use-Case диаграммой на рисунке 1.5 программа должна предоставлять следующие возможности для обычного пользователя:

- авторизация;
- регистрация;
- просмотр статистики всех игроков;
- просмотр статистики всех клубов;
- просмотр статистики игроков определенной команды;
- просмотр статистики определенной команды;
- добавление игрока в избранное;
- добавление команды в избранное;
- удаление игрока из избранного;
- удаление команды из избранного;
- просмотр списка избранного;
- поиск игрока по имени;
- поиск команды по названию;
- сравнение статистики игроков;
- сравнение статистики команд;
- просмотр текущих шансов команды на выход в плей-офф;
- выход из аккаунта.

Для модератора добавляются возможности:

- добавление нового игрока;
- добавление новой команды;
- удаление игрока;

- удаление команды;
- изменение статистики игроков;
- изменение статистики команд.

Для администратора добавляются возможности:

- изменение прав доступа пользователей;
- удаление пользователей.

2.3 Выводы из конструкторского раздела

В данном разделе была спроектирована база данных и приложение для доступа к ней. Были описаны таблицы базы данных и их компоненты. Представлена диаграмма компонентов и схемы триггеров. Описаны функционалы используемых триггеров. Также были формализованы требования к программе и описаны все функции, которые она должна выполнять, формализованы требования к каждому типу пользователей.

3 Технологическая часть

В данном разделе представлены архитектура приложения, средства разработки программного обеспечения, детали реализации и способы взаимодействия с программным продуктом.

3.1 Средства реализации

В данной работе в качестве СУБД был выбран PostgreSQL[8], так как:

- распространяется свободно;
- поддерживает сложные структуры и широкий спектр встроенных и определяемых пользователем типов данных;
- предоставляет обширный функционал, включая создание пользовательских функций, процедур, триггеров и возможности индексирования.

В данной работе в качестве языка программирования был выбран Python[9], так как:

- данный язык программирования позволяет обеспечить работу с PostgreSQL, а также имеет простой интерфейс подключения к базе данных;
- данный язык программирования обеспечивает быструю разработку программного продукта, он универсален, а написанный продукт можно легко расширять;
- данный язык программирования обладает полной документацией.

В данной работе в качестве среды разработки была выбрана «Visual Studio Code»[10] по следующим причинам:

- является бесплатной средой разработки;

- имеет большое количество различных расширений, облегчающих процесс разработки, также расширения просты в установке;
- имеет понятный и удобный в использовании интерфейс.

3.2 Детали реализации

3.2.1 Создание таблиц

На листинге 3.1 представлено создание таблицы Teams. Для всех полей за исключением поля country недопустимо значение NULL. Поле id является уникальным идентификатором и имеет ограничение PRIMARY KEY.

Листинг 3.1 – создание таблицы Teams

```
CREATE TABLE teams(  
    id SERIAL PRIMARY KEY NOT NULL,  
    name VARCHAR(50) NOT NULL,  
    country VARCHAR(50),  
    games INTEGER NOT NULL,  
    wins INTEGER NOT NULL,  
    loses INTEGER NOT NULL,  
    points INTEGER NOT NULL,  
    gdifference INTEGER NOT NULL  
);
```

На листинге 3.2 представлено создание таблицы Players. Для полей name, team_id, birth_year, height, weight, points, goals, assists, games, number, plusminus недопустимо значение NULL, поля birth_place, country, shoots могут принимать значение NULL. Поле id является уникальным идентификатором и имеет ограничение PRIMARY KEY. Поле team_id таблицы Players является внешним ключом и ссылается на первичный ключ таблицы Teams (поле id).

Листинг 3.2 – создание таблицы Players

```
CREATE TABLE players(  
    id SERIAL PRIMARY KEY NOT NULL,  
    name VARCHAR(50) NOT NULL,  
    team_id INTEGER NOT NULL,  
    birth_year INTEGER NOT NULL,  
    country VARCHAR(50),  
    birth_place VARCHAR(50),  
    height INTEGER NOT NULL,  
    weight INTEGER NOT NULL,  
    shoots VARCHAR(10),
```

```
points INTEGER NOT NULL,  
goals INTEGER NOT NULL,  
assists INTEGER NOT NULL,  
number INTEGER NOT NULL,  
games INTEGER NOT NULL,  
plusminus INTEGER NOT NULL,  
FOREIGN KEY (team_id) REFERENCES teams(id)  
);
```

На листинге 3.3 представлено создание таблицы Users. Для полей name, birth_year, mail, login, password, type недопустимо значение NULL, поле country может принимать значение NULL. Поле login также должно быть уникальным (в связи с тем, что авторизация пользователя осуществляется по его логину). Поле type может принимать только значения 0, 1, 2. Поле id является уникальным идентификатором и имеет ограничение PRIMARY KEY.

Листинг 3.3 – создание таблицы Users

```
CREATE TABLE users(  
    id SERIAL PRIMARY KEY NOT NULL,  
    name VARCHAR(50) NOT NULL,  
    country VARCHAR(50),  
    mail VARCHAR(80) NOT NULL,  
    birth_year INTEGER NOT NULL,  
    login VARCHAR(50) UNIQUE NOT NULL,  
    password VARCHAR(50) NOT NULL,  
    type INTEGER NOT NULL  
);
```

На листинге 3.4 представлено создание таблиц FavPlayers и FavTeams. В таблицах FavPlayers и FavTeams для всех полей недопустимо значение NULL. Поля id являюся уникальными идентификаторами для каждой из таблиц и имеют ограничение PRIMARY KEY.

Листинг 3.4 – создание таблиц FavPlayers и FavTeams

```
CREATE TABLE favplayers(  
    id SERIAL PRIMARY KEY NOT NULL,  
    usr_id INTEGER NOT NULL,  
    player_id INTEGER NOT NULL,  
    FOREIGN KEY (usr_id) REFERENCES users(id),  
    FOREIGN KEY (player_id) REFERENCES players(id)  
);
```

```
CREATE TABLE favteams(  
    id SERIAL PRIMARY KEY NOT NULL,  
    usr_id INTEGER NOT NULL,  
    team_id INTEGER NOT NULL,  
    FOREIGN KEY (usr_id) REFERENCES users(id),  
    FOREIGN KEY (team_id) REFERENCES teams(id)  
);
```

3.2.2 Используемые функции и триггеры

Для соответствия заявленным выше требованиям к функционалу приложения были написаны следующие функции:

- existing_log – проверка на существование логина;
- existing_mail – проверка на существование адреса электронной почты в системе;
- registration_us – регистрация нового пользователя;
- log_in – авторизация пользователя;
- print_all_players – печать статистики всех игроков лиги;
- print_all_teams – печать статистики всех клубов лиги;
- print_player_from_team – печать статистики игроков определенной команды;
- print_team – печать статистики определенной команды;
- add_player_to_list – добавить игрока в список избранных;
- add_team_to_list – добавить команду в избранное;
- del_player_from_list – удалить игрока из избранного;
- del_team_from_list – удалить команду из избранного;
- print_user_list – распечатать список избранного;
- find_player_by_name – поиск игрока по имени;
- find_team_by_name – поиск команды по названию;
- comparison_players – сравнение статистики игроков;
- comparison_teams – сравнение статистики команд;
- analyzing_teams – анализ шансов команд на выход в плей-офф;
- add_player_to_db – добавить игрока в базу данных;

- add_team_to_db – добавить команду в базу данных;
- delete_player_from_db – удалить игрока из базы данных;
- delete_team_from_db – удалить команду из базы данных;
- update_player_in_db – обновить статистику игроков в базе данных;
- update_team_in_db – обновить статистику команд в базе данных;
- delete_user_from_db – удалить пользователя из базы данных;
- change_user_type – изменить права доступа пользователя.

Далее на листингах 3.5-3.17 представлены коды ключевых функций данного программного продукта.

Листинг 3.5 – регистрация нового пользователя

```
def registration_us(name_us, country_us, mail_us, birth_us, log_us, pass_us,
type_us):
    insert_query = "('{name}', '{country}', '{mail}', '{birth_year}',
'{login}', '{password}', '{type}').format(
        name=name_us,
        country=country_us,
        mail=mail_us,
        birth_year=birth_us,
        login=log_us,
        password=pass_us,
        type=type_us)

    cursor.execute("INSERT INTO users (name, country, mail, birth_year, login,
password, type) VALUES " + insert_query)
    connection.commit()
```

Листинг 3.6 – добавление игрока в список избранного

```
def add_player_to_list(player_name, team_name, log_us):
    request = "select id from players where name = %s and team_id = (select id
from teams where name = %s)"
    cursor.execute(request, (player_name, team_name,))
    res = cursor.fetchone()

    request = "select id from users where login = %s"
    cursor.execute(request, (log_us,))
    res1 = cursor.fetchone()

    insert_query = "('{usr_id}', '{player_id}').format(usr_id = res1[0],
player_id = res[0])
    cursor.execute("INSERT INTO favplayers (usr_id, player_id) VALUES " +
insert_query)
```

```
connection.commit()
```

Листинг 3.7 – добавление команды в список избранного

```
def add_team_to_list(team_name, log_us):
    request = "select id from teams where name = %s"
    cursor.execute(request, (team_name,))
    res = cursor.fetchone()

    request = "select id from users where login = %s"
    cursor.execute(request, (log_us,))
    res1 = cursor.fetchone()

    insert_query = "('{usr_id}', '{team_id}').format(usr_id = res1[0], team_id
= res[0])
    cursor.execute("INSERT INTO favteams (usr_id, team_id) VALUES " +
insert_query)
    connection.commit()
```

Листинг 3.8 – удаление игрока из списка избранного

```
def del_player_from_list(player_name, team_name, log_us):
    request = "select id from players where name = %s and team_id = (select id
from teams where name = %s)"
    cursor.execute(request, (player_name, team_name,))
    res = cursor.fetchone()

    request = "select id from users where login = %s"
    cursor.execute(request, (log_us,))
    res1 = cursor.fetchone()

    request = "delete from favplayers where usr_id = %s and player_id = %s"
    cursor.execute(request, (res1[0], res[0],))
    connection.commit()
```

Листинг 3.9 – удаление команды из списка избранного

```
def del_team_from_list(team_name, log_us):
    request = "select id from teams where name = %s"
    cursor.execute(request, (team_name,))
    res = cursor.fetchone()

    request = "select id from users where login = %s"
    cursor.execute(request, (log_us,))
    res1 = cursor.fetchone()

    request = "delete from favteams where usr_id = %s and team_id = %s"
    cursor.execute(request, (res1[0], res[0],))
    connection.commit()
```


Листинг 3.10 – просмотр списка избранного

```
def print_user_list(log_us):
    request = "select player_id from favplayers where usr_id = (select id from
users where login = %s)"
    cursor.execute(request, (log_us,))
    res = cursor.fetchall()

    table_pl = PrettyTable()
    table_pl.field_names = ["Имя", "Игровой номер", "Год рождения", "Место
рождения", "Страна", "Рост", "Вес", "Хват", "Очки", "Голы", "Передачи",
"Количество игр", "+/-"]

    for row in res:
        request = "select * from players where id = %s"
        cursor.execute(request, (row[0],))
        res1 = cursor.fetchone()
        table_pl.add_row([res1[1], res1[3], res1[5], res1[4], res1[6], res1[7],
res1[8], res1[9], res1[10]])

    print("Избранные игроки: ")
    print(table_pl)

    request = "select team_id from favteams where usr_id = (select id from
users where login = %s)"
    cursor.execute(request, (log_us,))
    res2 = cursor.fetchall()

    table_tm = PrettyTable()
    table_tm.field_names = ["Название", "Страна", "Количество игр", "Победы",
"Поражения", "Очки", "Разница шайб"]

    for row in res2:
        request = "select * from teams where id = %s"
        cursor.execute(request, (row[0],))
        res3 = cursor.fetchone()
        table_tm.add_row([res3[1], res3[2], res3[3], res3[4], res3[5], res3[6],
res3[7]])

    print("Избранные команды: ")
    print(table_tm)
```

Листинг 3.11 – сравнение статистики игроков

```
def comparison_players(n):
    table_pl = PrettyTable()
    table_pl.field_names = ["Имя", "Игровой номер", "Год рождения", "Место
рождения", "Страна", "Рост", "Вес", "Хват", "Очки", "Голы", "Передачи",
"Количество игр", "+/-"]

    for i in range(n):
        player_name = input("Введите полное имя игрока: ")
        team_name = input("Введите полное название команды: ")
        request = "select * from players where name = %s and team_id = (select
```

```

id from teams where name = %s)"
    cursor.execute(request, (player_name, team_name,))
    res = cursor.fetchone()
    if res != []:
        table_pl.add_row([res[1], res[3], res[5], res[4], res[6], res[7],
res[8], res[9], res[10]])
    else:
        print("\nТаких игроков нет")
    print(table_pl)

```

Листинг 3.12 – сравнение статистики команд

```

def comparison_teams(n):
    table_tm = PrettyTable()
    table_tm.field_names = ["Название", "Страна", "Количество игр", "Победы",
"Поражения", "Очки", "Разница шайб"]

    for i in range(n):
        team_name = input("Введите полное название команды: ")
        request = "select * from teams where name = %s"
        cursor.execute(request, (team_name,))
        res = cursor.fetchone()
        if res != []:
            table_tm.add_row([res[1], res[2], res[3], res[4], res[5], res[6],
res[7]])
        else:
            print("\nКоманд с таким названием нет")
    print(table_tm)

```

Листинг 3.13 – анализ текущих шансов команд на выход в плей-офф

```

def analyzing_teams():
    request = "select * from teams"
    cursor.execute(request)
    res = cursor.fetchall()
    flag = True

    if res != []:
        table_tm = PrettyTable()
        table_tm.field_names = ["Название", "Шанс выйти в плей-офф (%)"]
        max_games = 0
        all_games = 86
        n = 0
        res.sort(key = lambda x: x[6], reverse=True)
        for row in res:
            n = n + 1
            if row[3] != all_games:
                flag = False
            if row[3] >= max_games:
                max_games = row[3]
            if n == 16:
                points_half_table = row[6]

```

```

        wins_half_table = row[4]
    if n == 17:
        points_half_table2 = row[6]
        games_half_table = row[3]

    max_points = 2 * max_games
    schance = 0.0
    n = 0

    for row in res:
        n = n + 1
        schance = row[6] / (max_points / 100)

        if n > 16:
            temp_points = (all_games - row[3]) * 2
            if (temp_points + row[6]) < points_half_table:
                schance = 0.0
            if (temp_points + row[6]) == points_half_table:
                if row[4] < wins_half_table:
                    schance = 0.0

        if n <= 16:
            temp_points = (all_games - games_half_table) * 2
            if (temp_points + points_half_table2) < row[6]:
                schance = 100.0

        if n <= 16 and flag:
            schance = 100.0
        if n > 16 and flag:
            schance = 0.0

        schance = float('{:.2f}'.format(schance))
        table_tm.add_row([row[1], schance])
    else:
        print("\nКоманд нет")
    print(table_tm)

```

Листинг 3.14 – добавление игрока

```

def add_player_to_db():
    player_name = input("Введите полное имя игрока: ")
    team = int(input("Введите id команды: "))
    birth_year1 = int(input("Введите год рождения: "))
    birth_place1 = input("Введите место рождения: ")
    country1 = input("Введите представляемую страну: ")
    shoot = input("Введите хват(R/L): ")
    height1 = int(input("Введите рост: "))
    weight1 = int(input("Введите вес: "))
    games1 = int(input("Введите количество сыгранных игр: "))
    points1 = int(input("Введите количество очков: "))
    goals1 = int(input("Введите количество голов: "))
    assists1 = int(input("Введите количество передач: "))
    number1 = int(input("Введите игровой номер: "))
    plusminus1 = int(input("Введите плюс/минус: "))

```

```

        insert_query += "('{name}', '{team_id}', '{birth_year}', '{country}',
'{birth_place}', {height}, '{weight}', '{shoots}', '{points}', '{goals}',
'{assists}', '{number}', '{games}', '{plusminus}')"
        .format(
            name=player_name,
            team_id=team,
            birth_year=birth_year1,
            country=country1,
            birth_place=birth_place1,
            height=height1,
            weight=weight1,
            shoots=shoot,
            points=points1,
            goals=goals1,
            assists=assists1,
            number=number1,
            games=games1,
            plusminus=plusminus1)

        cursor.execute("INSERT INTO players (name, team_id, birth_year, country,
birth_place, height, weight, shoots, points, goals, assists, number, games,
plusminus) VALUES " + insert_query)
        connection.commit()

```

Листинг 3.15 – удаление игрока

```

def delete_player_from_db():
    player_id = int(input("Введите id игрока: "))
    request = "delete from players where id = %s"
    cursor.execute(request, (player_id,))
    connection.commit()

```

Листинг 3.16 – удаление команды

```

def delete_team_from_db():
    team_id = int(input("Введите id команды: "))
    request = "delete from teams where id = %s"
    cursor.execute(request, (team_id,))
    connection.commit()

```

Листинг 3.17 – удаление пользователя

```

def delete_user_from_db():
    user_id = int(input("Введите id пользователя: "))
    request = "delete from users where id = %s"
    cursor.execute(request, (user_id,))
    connection.commit()

```

Для функций удаления игроков, команд и пользователей были написаны триггеры, представленные на листингах 3.18-3.20.

Листинг 3.17 – триггер на удаление записей, связанных с игроком, из другой таблицы

```
CREATE FUNCTION trigger_players_before_del () RETURNS trigger AS '  
BEGIN  
if (select count(*) from favplayers where favplayers.player_id = OLD.id)>0  
then delete from favplayers where favplayers.player_id = OLD.id;  
end if;  
return OLD;  
END;  
' LANGUAGE plpgsql;  
  
CREATE TRIGGER tr_player_del_befor  
BEFORE DELETE ON players FOR EACH ROW  
EXECUTE PROCEDURE trigger_players_before_del();
```

Листинг 3.18 – триггер на удаление записей, связанных с командой, из других таблиц

```
CREATE FUNCTION trigger_teams_before_del () RETURNS trigger AS '  
BEGIN  
if (select count(*) from favteams where favteams.team_id = OLD.id)>0  
then delete from favteams where favteams.team_id = OLD.id;  
end if;  
if (select count(*) from players where players.team_id = OLD.id)>0  
then delete from players where players.team_id = OLD.id;  
end if;  
return NULL;  
END;  
' LANGUAGE plpgsql;  
  
CREATE TRIGGER tr_team_del_befor  
BEFORE DELETE ON teams FOR EACH ROW  
EXECUTE PROCEDURE trigger_teams_before_del();
```

Листинг 3.18 – триггер на удаление записей, связанных с пользователем, из других таблиц

```
CREATE FUNCTION trigger_users_before_del () RETURNS trigger AS '  
BEGIN  
if (select count(*) from favplayers where favplayers.user_id = OLD.id)>0  
then delete from favteams where favplayers.user_id = OLD.id;  
end if;  
if (select count(*) from favteams where favteams.user_id = OLD.id)>0  
then delete from favteams where favteams.user_id = OLD.id;  
end if;  
return NULL;  
END;  
' LANGUAGE plpgsql;  
  
CREATE TRIGGER tr_user_del_befor
```

```
BEFORE DELETE ON users FOR EACH ROW  
EXECUTE PROCEDURE trigger_users_before_del();
```

3.3 Взаимодействие с приложением

В качестве программного продукта было создано консольное приложение.

Работа программы начинается с авторизации (рисунок 3.1). Пользователь должен ввести свои логин и пароль, чтобы выполнить вход. Если пользователь еще не имеет аккаунт, он может зарегистрироваться, выбрав пункт «Зарегистрироваться».

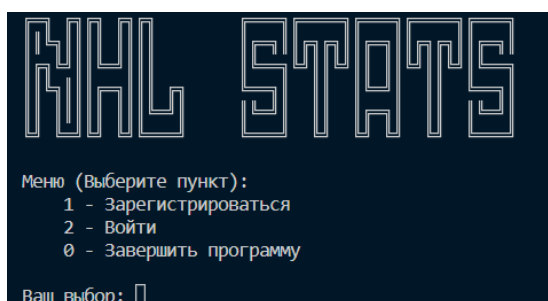


Рисунок 3.1 – авторизация пользователя

При выборе регистрации необходимо ввести все запрашиваемые данные, после чего пользователь будет зарегистрирован и произойдет вход в систему, что представлено на рисунке 3.2.

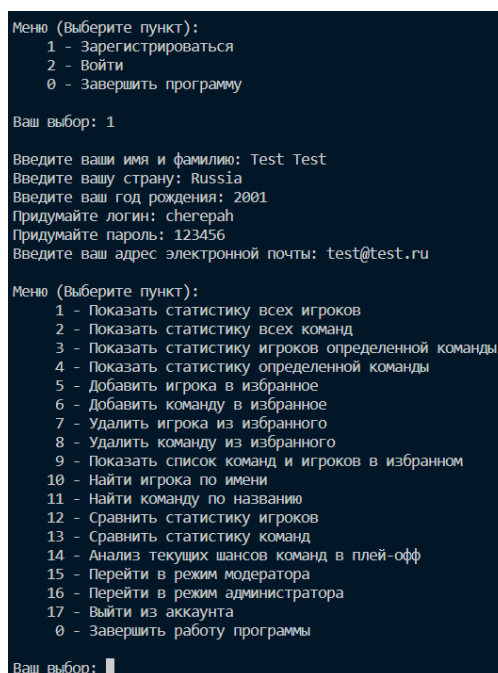


Рисунок 3.2 – успешная регистрация в системе

При выборе пункта «Войти» необходимо ввести корректные логин и пароль, после чего будет произведен вход в систему, что представлено на рисунке 3.3.

```

Меню (Выберите пункт):
1 - Зарегистрироваться
2 - Войти
0 - Завершить программу

Ваш выбор: 2

Введите логин: cherepah

Введите пароль: 123456

Меню (Выберите пункт):
1 - Показать статистику всех игроков
2 - Показать статистику всех команд
3 - Показать статистику игроков определенной команды
4 - Показать статистику определенной команды
5 - Добавить игрока в избранное
6 - Добавить команду в избранное
7 - Удалить игрока из избранного
8 - Удалить команду из избранного
9 - Показать список команд и игроков в избранном
10 - Найти игрока по имени
11 - Найти команду по названию
12 - Сравнить статистику игроков
13 - Сравнить статистику команд
14 - Анализ текущих шансов команд в плей-офф
15 - Перейти в режим модератора
16 - Перейти в режим администратора
17 - Выйти из аккаунта
0 - Завершить работу программы

Ваш выбор:

```

Рисунок 3.3 – успешный авторизация в системе

При успешном входе в систему появляется меню выбора действий, состоящее из 17 пунктов действия и пункта завершающего работу программы.

Ниже, на рисунках 3.4-3.14 представлены примеры работы нескольких пунктов данного меню.

Ваш выбор: 1

N	Имя	Игровой номер	Год рождения	Место рождения	Страна	Рост	Вес	Хват	Очки	Голы	Передачи	Количество игр	+/-
1	Kris Letang	45	2001	Latvia	Latvia	197	91	Left	13	3	10	27	0
2	Marc-Andre Fleury	80	1988	Slovakia	Sweden	173	62	Right	29	16	13	24	7
3	Jeff Carter	86	2000	Slovakia	Russia	197	98	Left	17	10	7	33	8
4	Mark Matheson	15	1984	Russia	Finland	187	89	Left	31	22	9	47	-3
5	Nicolas Aube-Kubel	82	1997	Slovakia	Czech Republic	203	61	Right	13	8	5	43	2
6	Nico Hischier	72	1988	Czech Republic	Norway	179	87	Left	33	22	11	24	7
7	Alex Doro	89	1997	Denmark	Denmark	195	92	Right	22	14	8	49	4
8	Drew O'Connor	81	1991	Norway	Finland	204	102	Right	27	20	7	30	4
9	Rasmus Dahlin	46	1993	Norway	Switzerland	193	90	Right	31	22	9	12	7
10	Nico Hischier	43	1999	Switzerland	Finland	188	63	Right	18	12	6	44	2
11	John Marino	28	1991	Canada	Latvia	207	112	Right	5	4	1	15	6
12	Alex Doro	33	2004	Norway	Russia	208	61	Right	15	6	9	10	8
13	Casey DeSmith	77	2001	Sweden	Norway	201	60	Left	17	2	15	15	3
14	Jake Guentzel	49	1990	Canada	Switzerland	192	82	Left	14	13	1	39	-2
15	Mackenzie Blackwood	72	1995	Sweden	Switzerland	208	109	Right	28	14	14	13	0
16	Sidney Crosby	5	1992	Czech Republic	Latvia	210	62	Right	16	9	7	27	0

Рисунок 3.4 – фрагмент таблицы со статистикой всех игроков

Ваш выбор: 2

N	Название	Страна	Количество игр	Победы	Поражения	Очки	Разница шайб
1	Anaheim Ducks	Canada	10	7	3	14	49
2	Arizona Coyotes	Canada	13	8	5	16	-45
3	Boston Bruins	USA	12	8	4	16	40
4	Buffalo Sabres	Canada	11	10	1	20	-14
5	Calgary Flames	Canada	9	8	1	16	22
6	Carolina Hurricanes	USA	11	8	3	16	-9
7	Chicago Blackhawks	USA	10	6	4	12	-25
8	Colorado Avalanche	Canada	14	10	4	20	-7
9	Columbus Blue Jackets	USA	12	7	5	14	-48
10	Dallas Stars	USA	12	9	3	18	12
11	Detroit Red Wings	Canada	6	5	1	10	18
12	Edmonton Oilers	USA	9	6	3	12	-27
13	Florida Panthers	Canada	13	8	5	16	12

Рисунок 3.5 – фрагмент таблицы со статистикой всех команд

Ваш выбор: 3
Введите полное название команды: Pittsburgh Penguins

Имя	Игровой номер	Год рождения	Место рождения	Страна	Рост	Вес	Хват	Очки	Голы	Передачи	Количество игр	+/-
Alex D'Orio	97	1994	Denmark	Sweden	186	72	Right	10	6	4	34	4
Andre Burakovsky	97	1989	Finland	Slovakia	183	112	Right	18	15	3	47	-2
John Marino	94	1996	Latvia	Russia	213	111	Left	20	16	4	26	8
Dowson Mercer	71	2004	USA	Latvia	195	81	Right	25	18	7	35	10
Mackenzie Blackwood	56	1988	Latvia	USA	198	98	Right	13	7	6	41	10
Evgeni Malkin	69	1981	Sweden	USA	198	108	Right	6	3	3	24	-2
Tristan Jarry	9	1990	Russia	Denmark	189	75	Right	18	15	3	14	3
Brian Dumoulin	25	1986	Canada	Latvia	171	70	Right	8	3	5	10	5
Alex D'Orio	93	1982	Norway	Sweden	172	102	Right	30	25	5	44	-4
Jake Guentzel	91	1994	Denmark	Sweden	191	108	Left	42	27	15	28	5
Rasmus Dahlin	35	2003	Finland	USA	207	66	Right	8	5	3	45	-4
Tristan Jarry	8	1982	Czech Republic	Switzerland	201	85	Left	9	9	0	47	6
Nicolas Aube-Kubel	3	1994	Czech Republic	Latvia	191	97	Left	28	27	1	28	-5
John Marino	34	1990	USA	Denmark	205	76	Left	25	22	3	37	6
Ben Meyers	67	1994	Finland	Canada	173	97	Right	28	27	1	10	7
Teddy Blueger	36	1984	Russia	Czech Republic	198	105	Left	21	14	7	25	0
Rasmus Dahlin	92	2004	Czech Republic	Latvia	218	67	Left	40	25	15	14	0
Nicolas Aube-Kubel	63	1987	Czech Republic	Norway	177	81	Left	20	11	9	28	4
Brian Boyle	70	2003	Sweden	Sweden	212	84	Left	29	28	1	40	3
Nathan Beaulieu	88	1983	Finland	Sweden	172	81	Left	38	27	11	27	4
Teddy Blueger	59	1985	Czech Republic	Latvia	198	111	Left	13	8	5	11	-1
Jason Zucker	79	2004	Slovakia	Canada	202	100	Right	25	18	7	50	0
Nathan Beaulieu	16	2003	Switzerland	Russia	197	110	Right	19	17	2	38	0

Рисунок 3.6 – фрагмент таблицы со статистикой всех игроков определенной команды

Ваш выбор: 5
Введите полное имя игрока: Alex D'Orio
Введите полное название его команды: Pittsburgh Penguins

Рисунок 3.7 – добавление игрока в список избранного

Ваш выбор: 6
Введите полное название команды: Pittsburgh Penguins

Рисунок 3.8 – добавление команды в список избранного

Ваш выбор: 9
Избранные игроки:

Имя	Игровой номер	Год рождения	Место рождения	Страна	Рост	Вес	Хват	Очки	Голы	Передачи	Количество игр	+/-
Alex Dario	97	1994	Denmark	Sweden	186	72	Right	10	6	4	34	4

Избранные команды:

Название	Страна	Количество игр	Победы	Поражения	Очки	Разница шайб
Pittsburgh Penguins	USA	10	7	3	14	-49

Рисунок 3.9 – просмотр списка избранного

Ваш выбор: 10
Введите имя или часть имени игрока: alex

Имя	Игровой номер	Год рождения	Место рождения	Страна	Рост	Вес	Хват	Очки	Голы	Передачи	Количество игр	+/-
Alex Dario	97	1994	Denmark	Sweden	186	72	Right	10	6	4	34	4
Alex Newhook	60	1983	Slovakia	Belarus	173	109	Right	27	14	13	38	8
Alex Dario	33	1993	Switzerland	Canada	202	76	Left	32	25	7	26	3
Alex Dario	18	1988	Czech Republic	Finland	191	85	Right	28	27	1	40	1
Alex Newhook	98	2000	Switzerland	Finland	186	87	Left	21	9	12	41	5
Alex Newhook	92	1984	Denmark	Latvia	173	112	Right	24	20	4	27	0
Alex Dario	77	1996	Russia	USA	176	90	Right	18	10	8	29	1
Alex Dario	5	1999	Russia	Czech Republic	211	64	Left	10	7	3	40	10
Alex Newhook	58	2003	Belarus	Slovakia	187	108	Left	14	5	9	31	4
Alex Dario	52	1982	USA	Norway	189	75	Left	15	11	4	27	9

Рисунок 3.10 – фрагмент таблицы результата поиска игрока по имени

Ваш выбор: 12
Введите количество игроков для сравнения: 2
Введите полное имя игрока: Brian Rust
Введите полное название команды: Anaheim Ducks
Введите полное имя игрока: Dowson Mercer
Введите полное название команды: Pittsburgh Penguins

Имя	Игровой номер	Год рождения	Место рождения	Страна	Рост	Вес	Хват	Очки	Голы	Передачи	Количество игр	+/-
Brian Rust	25	1990	Slovakia	Canada	220	87	Left	28	20	8	31	-4
Dowson Mercer	71	2004	USA	Latvia	195	81	Right	25	18	7	35	10

Рисунок 3.11 – сравнение игроков

Ваш выбор: 14

Название	Шанс выйти в плей-офф (%)
Buffalo Sabres	66.67
Colorado Avalanche	66.67
Los Angeles Kings	66.67
New York Rangers	66.67
Vancouver Canucks	66.67
Washington Capitals	66.67
Dallas Stars	60.0
Montreal Canadiens	60.0
Nashville Predators	60.0
New York Islanders	60.0
Seattle Kraken	60.0
Vegas Golden Knights	60.0
Arizona Coyotes	53.33
Boston Bruins	53.33
Calgary Flames	53.33
Carolina Hurricanes	53.33
Florida Panthers	53.33
Minnesota Wild	53.33
New Jersey Devils	53.33
Toronto Maple Leafs	53.33
Anaheim Ducks	46.67
Columbus Blue Jackets	46.67
Philadelphia Flyers	46.67
Pittsburgh Penguins	46.67
Winnipeg Jets	46.67
Chicago Blackhawks	40.0
Edmonton Oilers	40.0
Tampa Bay Lightning	40.0
Detroit Red Wings	33.33
Ottawa Senators	33.33
San Jose Sharks	33.33
St. Louis Blues	33.33

Рисунок 3.12 – пример результата анализа шансов команд на выход в плей-офф

Меню (Выберите пункт):

- 1 - Добавить игрока
- 2 - Добавить команду
- 3 - Удалить игрока
- 4 - Удалить команду
- 5 - Обновить статистику игрока
- 6 - Обновить статистику команды
- 7 - Вернуться в режим пользователя

Ваш выбор: █

Рисунок 3.13 – возможности модератора

Меню (Выберите пункт):

- 1 - Добавить игрока
- 2 - Добавить команду
- 3 - Удалить игрока
- 4 - Удалить команду
- 5 - Обновить статистику игрока
- 6 - Обновить статистику команды
- 7 - Удалить пользователя
- 8 - Изменить права доступа пользователя
- 9 - Вернуться в режим пользователя

Ваш выбор: █

Рисунок 3.14 – возможности администратора

3.4 Выводы из технологического раздела

В данном разделе были выбраны средства разработки, такие как СУБД – PostgreSQL, язык программирования – Python, среда разработки – Visual Studio Code. Были рассмотрены функции, которые должна выполнять программа и триггеры, используемые в программе. Также были предоставлены сведения об интерфейсе программы, процессе взаимодействия с приложением и примеры работы программы.

4 Экспериментальная часть

В данном разделе будет проведен анализ зависимости шансов на выход в плей-офф команд Национальной Хоккейной Лиги от количества проведенных ими матчей (то есть от того, насколько процентов регулярный сезон уже завершен).

4.1 Технические характеристики

Ниже приведены технические характеристики устройства, на котором было проведено тестирование ПО:

- операционная система: Windows 10 64-bit;
- оперативная память: 16 GB;
- процессор: 2.6 GHz 6-ядерный процессор Intel Core i7.

4.2 Анализ зависимости

Исходные данные представлены на рисунке 4.1. Для представленных данных регулярный сезон завершен на 15%, то есть команды сыграли в среднем 12 матчей из 86 возможных.

N	Название	Страна	Количество игр	Победы	Поражения	Очки	Разница шайб
1	Anaheim Ducks	Canada	10	7	3	14	49
2	Arizona Coyotes	Canada	13	8	5	16	-45
3	Boston Bruins	USA	12	8	4	16	40
4	Buffalo Sabres	Canada	11	10	1	20	-14
5	Calgary Flames	Canada	9	8	1	16	22
6	Carolina Hurricanes	USA	11	8	3	16	-9
7	Chicago Blackhawks	USA	10	6	4	12	-25
8	Colorado Avalanche	Canada	14	10	4	20	-7
9	Columbus Blue Jackets	USA	12	7	5	14	-48
10	Dallas Stars	USA	12	9	3	18	12
11	Detroit Red Wings	Canada	6	5	1	10	18
12	Edmonton Oilers	USA	9	6	3	12	-27
13	Florida Panthers	Canada	13	8	5	16	12
14	Los Angeles Kings	Canada	15	10	5	20	-27
15	Minnesota Wild	Canada	13	8	5	16	20
16	Montreal Canadiens	Canada	10	9	1	18	-14
17	Nashville Predators	Canada	11	9	2	18	-37
18	New Jersey Devils	Canada	9	8	1	16	30
19	New York Islanders	Canada	13	9	4	18	45
20	New York Rangers	USA	14	10	4	20	-25
21	Ottawa Senators	USA	9	5	4	10	-1
22	Philadelphia Flyers	USA	9	7	2	14	-19
23	Pittsburgh Penguins	USA	10	7	3	14	-49
24	San Jose Sharks	Canada	7	5	2	10	-1
25	Seattle Kraken	USA	10	9	1	18	37
26	St. Louis Blues	Canada	8	5	3	10	14
27	Tampa Bay Lightning	Canada	7	6	1	12	-26
28	Toronto Maple Leafs	USA	13	8	5	16	-46
29	Vancouver Canucks	USA	15	10	5	20	-12
30	Vegas Golden Knights	USA	10	9	1	18	44
31	Washington Capitals	USA	12	10	2	20	46
32	Winnipeg Jets	Canada	12	7	5	14	34

Рисунок 4.1 – исходные данные статистики команд

На рисунке 4.2 представлены текущие шансы всех команд на выход в плей-офф. При маленьком количестве матчей ненулевой шанс на выход в плей-офф есть у каждой команды, из-за чего невозможно определить какие команды действительно выйдут в плей-офф. Также шанс напрямую зависит от процента выигранных матчей.

Название	Шанс выйти в плей-офф (%)
Buffalo Sabres	66.67
Colorado Avalanche	66.67
Los Angeles Kings	66.67
New York Rangers	66.67
Vancouver Canucks	66.67
Washington Capitals	66.67
Dallas Stars	60.0
Montreal Canadiens	60.0
Nashville Predators	60.0
New York Islanders	60.0
Seattle Kraken	60.0
Vegas Golden Knights	60.0
Arizona Coyotes	53.33
Boston Bruins	53.33
Calgary Flames	53.33
Carolina Hurricanes	53.33
Florida Panthers	53.33
Minnesota Wild	53.33
New Jersey Devils	53.33
Toronto Maple Leafs	53.33
Anaheim Ducks	46.67
Columbus Blue Jackets	46.67
Philadelphia Flyers	46.67
Pittsburgh Penguins	46.67
Winnipeg Jets	46.67
Chicago Blackhawks	40.0
Edmonton Oilers	40.0
Tampa Bay Lightning	40.0
Detroit Red Wings	33.33
Ottawa Senators	33.33
San Jose Sharks	33.33
St. Louis Blues	33.33

Рисунок 4.2 – шанс выйти в плей-офф у команд при 15% завершенности регулярного чемпионата

Следующие данные представлены для регулярного сезона, завершившегося на 25%, то есть команды сыграли в среднем 21 матч. Статистика команд представлена на рисунке 4.3.

N	Название	Страна	Количество игр	Победы	Поражения	Очки	Разница шайб
1	Anaheim Ducks	Canada	20	13	7	26	49
2	Arizona Coyotes	Canada	22	10	12	20	-45
3	Boston Bruins	USA	20	14	6	28	40
4	Buffalo Sabres	Canada	19	15	4	30	-14
5	Calgary Flames	Canada	18	9	9	18	22
6	Carolina Hurricanes	USA	22	16	6	32	-9
7	Chicago Blackhawks	USA	23	18	5	36	-25
8	Colorado Avalanche	Canada	19	12	7	24	-7
9	Columbus Blue Jackets	USA	21	14	7	28	-48
10	Dallas Stars	USA	21	14	7	28	12
11	Detroit Red Wings	Canada	21	17	4	34	18
12	Edmonton Oilers	USA	22	16	6	32	-27
13	Florida Panthers	Canada	23	16	7	32	12
14	Los Angeles Kings	Canada	19	13	6	26	-27
15	Minnesota Wild	Canada	18	9	9	18	20
16	Montreal Canadiens	Canada	20	17	3	34	-14
17	Nashville Predators	Canada	21	15	6	30	-37
18	New Jersey Devils	Canada	22	13	9	26	30
19	New York Islanders	Canada	21	16	5	32	45
20	New York Rangers	USA	23	14	9	28	-25
21	Ottawa Senators	USA	22	15	7	30	-1
22	Philadelphia Flyers	USA	22	19	3	38	-19
23	Pittsburgh Penguins	USA	23	17	6	34	-49
24	San Jose Sharks	Canada	21	16	5	32	-1
25	Seattle Kraken	USA	20	16	4	32	37
26	St. Louis Blues	Canada	19	12	7	24	14
27	Tampa Bay Lightning	Canada	20	18	2	36	-26
28	Toronto Maple Leafs	USA	21	16	5	32	-46
29	Vancouver Canucks	USA	23	17	6	34	-12
30	Vegas Golden Knights	USA	20	13	7	26	44
31	Washington Capitals	USA	22	11	11	22	46
32	Winnipeg Jets	Canada	21	15	6	30	34

Рисунок 4.3 – данные о статистике команд для 25% завершенности регулярного сезона

На рисунке 4.4 представлены текущие шансы всех команд на выход в плей-офф при 25% завершенности сезона. Ненулевой шанс на выход в плей-офф все еще есть у каждой команды, из-за чего все также невозможно определить какие команды действительно выйдут в плей-офф, но уже сейчас вероятность выйти в плей-офф у команд с наибольшими шансами повышается.

Название	Шанс выйти в плей-офф (%)
Philadelphia Flyers	82.61
Chicago Blackhawks	78.26
Tampa Bay Lightning	78.26
Detroit Red Wings	73.91
Montreal Canadiens	73.91
Pittsburgh Penguins	73.91
Vancouver Canucks	73.91
Carolina Hurricanes	69.57
Edmonton Oilers	69.57
Florida Panthers	69.57
New York Islanders	69.57
San Jose Sharks	69.57
Seattle Kraken	69.57
Toronto Maple Leafs	69.57
Buffalo Sabres	65.22
Nashville Predators	65.22
Ottawa Senators	65.22
Winnipeg Jets	65.22
Boston Bruins	60.87
Columbus Blue Jackets	60.87
Dallas Stars	60.87
New York Rangers	60.87
Anaheim Ducks	56.52
Los Angeles Kings	56.52
New Jersey Devils	56.52
Vegas Golden Knights	56.52
Colorado Avalanche	52.17
St. Louis Blues	52.17
Washington Capitals	47.83
Arizona Coyotes	43.48
Calgary Flames	39.13
Minnesota Wild	39.13

Рисунок 4.4 – шанс выйти в плей-офф у команд при 25% завершенности регулярного чемпионата

Следующие данные представлены для регулярного сезона, завершившегося на 50%, то есть команды сыграли в среднем 43 матча. Статистика команд представлена на рисунке 4.5.

N	Название	Страна	Количество игр	Победы	Поражения	Очки	Разница шайб
1	Anaheim Ducks	Canada	43	25	18	50	49
2	Arizona Coyotes	Canada	42	21	21	42	-45
3	Boston Bruins	USA	44	30	14	60	40
4	Buffalo Sabres	Canada	45	30	15	60	-14
5	Calgary Flames	Canada	42	22	20	44	22
6	Carolina Hurricanes	USA	41	26	15	52	-9
7	Chicago Blackhawks	USA	42	29	13	58	-25
8	Colorado Avalanche	Canada	43	26	17	52	-7
9	Columbus Blue Jackets	USA	41	23	18	46	-48
10	Dallas Stars	USA	40	25	15	50	12
11	Detroit Red Wings	Canada	43	30	13	60	18
12	Edmonton Oilers	USA	44	32	12	64	-27
13	Florida Panthers	Canada	44	26	17	52	12
14	Los Angeles Kings	Canada	42	26	16	52	-27
15	Minnesota Wild	Canada	43	18	25	36	20
16	Montreal Canadiens	Canada	43	33	10	66	-14
17	Nashville Predators	Canada	43	26	17	52	-37
18	New Jersey Devils	Canada	43	27	16	54	30
19	New York Islanders	Canada	44	24	20	48	45
20	New York Rangers	USA	42	23	19	46	-25
21	Ottawa Senators	USA	41	22	19	44	-1
22	Philadelphia Flyers	USA	41	26	15	52	-19
23	Pittsburgh Penguins	USA	43	26	17	52	-49
24	San Jose Sharks	Canada	44	32	12	64	-1
25	Seattle Kraken	USA	44	33	11	66	37
26	St. Louis Blues	Canada	44	26	18	52	14
27	Tampa Bay Lightning	Canada	42	33	9	66	-26
28	Toronto Maple Leafs	USA	41	30	11	60	-46
29	Vancouver Canucks	USA	45	30	15	60	-12
30	Vegas Golden Knights	USA	43	25	18	50	44
31	Washington Capitals	USA	41	15	26	30	46
32	Winnipeg Jets	Canada	42	23	19	46	34

Рисунок 4.5 – данные о статистике команд для 50% завершенности регулярного сезона

На рисунке 4.6 представлены текущие шансы всех команд на выход в плей-офф при 50% завершенности сезона. Ненулевой шанс на выход в плей-офф все еще есть у каждой команды, из-за чего все также невозможно определить какие команды действительно выйдут в плей-офф, но уже сейчас вероятность выйти в плей-офф у команд с наибольшими шансами значительно повышается, так как количество оставшихся до конца сезона матчей уменьшается.

Название	Шанс выйти в плей-офф (%)
Montreal Canadiens	73.33
Seattle Kraken	73.33
Tampa Bay Lightning	73.33
Edmonton Oilers	71.11
San Jose Sharks	71.11
Boston Bruins	66.67
Buffalo Sabres	66.67
Detroit Red Wings	66.67
Toronto Maple Leafs	66.67
Vancouver Canucks	66.67
Chicago Blackhawks	64.44
New Jersey Devils	60.0
Carolina Hurricanes	57.78
Colorado Avalanche	57.78
Florida Panthers	57.78
Los Angeles Kings	57.78
Nashville Predators	57.78
Philadelphia Flyers	57.78
Pittsburgh Penguins	57.78
St. Louis Blues	57.78
Anaheim Ducks	55.56
Dallas Stars	55.56
Vegas Golden Knights	55.56
New York Islanders	53.33
Columbus Blue Jackets	51.11
New York Rangers	51.11
Winnipeg Jets	51.11
Calgary Flames	48.89
Ottawa Senators	48.89
Arizona Coyotes	46.67
Minnesota Wild	40.0
Washington Capitals	33.33

Рисунок 4.6 – шанс выйти в плей-офф у команд при 50% завершенности регулярного чемпионата

Следующие данные представлены для регулярного сезона, завершившегося на 75%, то есть команды сыграли в среднем 65 матчей. Статистика команд представлена на рисунке 4.7.

N	Название	Страна	Количество игр	Победы	Поражения	Очки	Разница шайб
1	Anaheim Ducks	Canada	64	36	28	72	49
2	Arizona Coyotes	Canada	65	34	31	68	-45
3	Boston Bruins	USA	62	38	24	76	40
4	Buffalo Sabres	Canada	65	35	30	70	-14
5	Calgary Flames	Canada	67	27	40	54	22
6	Carolina Hurricanes	USA	62	37	25	74	-9
7	Chicago Blackhawks	USA	64	35	29	70	-25
8	Colorado Avalanche	Canada	66	39	27	78	-7
9	Columbus Blue Jackets	USA	67	37	30	74	-48
10	Dallas Stars	USA	65	37	28	74	12
11	Detroit Red Wings	Canada	65	36	29	72	18
12	Edmonton Oilers	USA	64	40	24	80	-27
13	Florida Panthers	Canada	65	38	27	76	12
14	Los Angeles Kings	Canada	63	34	29	74	-27
15	Minnesota Wild	Canada	62	27	35	54	20
16	Montreal Canadiens	Canada	66	43	23	86	-14
17	Nashville Predators	Canada	67	36	31	72	-37
18	New Jersey Devils	Canada	66	37	29	74	30
19	New York Islanders	Canada	65	27	38	54	45
20	New York Rangers	USA	64	33	31	66	-25
21	Ottawa Senators	USA	65	33	32	66	-1
22	Philadelphia Flyers	USA	63	37	26	74	-19
23	Pittsburgh Penguins	USA	66	34	32	68	-49
24	San Jose Sharks	Canada	66	45	21	90	-1
25	Seattle Kraken	USA	65	41	24	82	37
26	St. Louis Blues	Canada	65	35	30	70	14
27	Tampa Bay Lightning	Canada	65	45	20	90	-26
28	Toronto Maple Leafs	USA	64	40	24	80	-46
29	Vancouver Canucks	USA	66	36	30	72	-12
30	Vegas Golden Knights	USA	67	37	30	74	44
31	Washington Capitals	USA	64	25	39	50	46
32	Winnipeg Jets	Canada	64	30	34	60	34

Рисунок 4.7 – данные о статистике команд для 75% завершенности регулярного сезона

На рисунке 4.8 представлены текущие шансы всех команд на выход в плей-офф при 75% завершенности сезона. Ненулевой шанс на выход в плей-офф все еще есть у каждой команды, из-за чего все также невозможно определить какие команды действительно выйдут в плей-офф, но уже сейчас вероятность выйти в плей-офф у команд с наименьшими шансами значительно уменьшается, так как количество оставшихся до конца сезона матчей слишком мало.

Название	Шанс выйти в плей-офф (%)
San Jose Sharks	67.16
Tampa Bay Lightning	67.16
Montreal Canadiens	64.18
Seattle Kraken	61.19
Edmonton Oilers	59.7
Toronto Maple Leafs	59.7
Colorado Avalanche	58.21
Boston Bruins	56.72
Florida Panthers	56.72
Carolina Hurricanes	55.22
Columbus Blue Jackets	55.22
Dallas Stars	55.22
Los Angeles Kings	55.22
New Jersey Devils	55.22
Philadelphia Flyers	55.22
Vegas Golden Knights	55.22
Anaheim Ducks	53.73
Detroit Red Wings	53.73
Nashville Predators	53.73
Vancouver Canucks	53.73
Buffalo Sabres	52.24
Chicago Blackhawks	52.24
St. Louis Blues	52.24
Arizona Coyotes	50.75
Pittsburgh Penguins	50.75
New York Rangers	49.25
Ottawa Senators	49.25
Winnipeg Jets	44.78
Calgary Flames	40.3
Minnesota Wild	40.3
New York Islanders	40.3
Washington Capitals	37.31

Рисунок 4.8 – шанс выйти в плей-офф у команд при 75% завершенности регулярного чемпионата

Следующие данные представлены для регулярного сезона, завершившегося на 90%, то есть команды сыграли в среднем 78 матчей. Статистика команд представлена на рисунке 4.9.

N	Название	Страна	Количество игр	Победы	Поражения	Очки	Разница шайб
1	Anaheim Ducks	Canada	77	44	33	88	49
2	Arizona Coyotes	Canada	78	42	36	84	-45
3	Boston Bruins	USA	78	48	30	96	40
4	Buffalo Sabres	Canada	79	43	36	86	-14
5	Calgary Flames	Canada	80	34	46	68	22
6	Carolina Hurricanes	USA	76	47	29	94	-9
7	Chicago Blackhawks	USA	76	43	33	86	-25
8	Colorado Avalanche	Canada	75	42	33	84	-7
9	Columbus Blue Jackets	USA	77	43	34	86	-48
10	Dallas Stars	USA	78	46	32	92	12
11	Detroit Red Wings	Canada	78	41	37	82	18
12	Edmonton Oilers	USA	77	48	29	96	-27
13	Florida Panthers	Canada	76	45	31	90	12
14	Los Angeles Kings	Canada	79	42	37	84	-27
15	Minnesota Wild	Canada	79	35	44	70	20
16	Montreal Canadiens	Canada	79	51	28	102	-14
17	Nashville Predators	Canada	77	38	39	76	-37
18	New Jersey Devils	Canada	78	43	35	86	30
19	New York Islanders	Canada	78	36	42	72	45
20	New York Rangers	USA	78	41	37	82	-25
21	Ottawa Senators	USA	77	38	39	76	-1
22	Philadelphia Flyers	USA	78	47	31	94	-19
23	Pittsburgh Penguins	USA	77	38	39	76	-49
24	San Jose Sharks	Canada	76	49	27	98	-1
25	Seattle Kraken	USA	79	50	29	100	37
26	St. Louis Blues	Canada	80	46	34	92	14
27	Tampa Bay Lightning	Canada	80	48	32	96	-26
28	Toronto Maple Leafs	USA	78	54	24	108	-46
29	Vancouver Canucks	USA	78	40	38	80	-12
30	Vegas Golden Knights	USA	77	42	35	84	44
31	Washington Capitals	USA	76	27	49	54	46
32	Winnipeg Jets	Canada	75	37	38	74	34

Рисунок 4.9 – данные о статистике команд для 90% завершенности регулярного сезона

На рисунке 4.10 представлены текущие шансы всех команд на выход в плей-офф при 90% завершенности сезона. Нулевой шанс на выход в плей-офф оказывается у команд, количество очков которых за оставшиеся 10% матчей не позволяет им пройти в 16 первых команд. Стопроцентный шанс выхода в плей-офф оказывается у команд, которые за оставшуюся часть сезона уже не смогут спуститься ниже 16-ой строчки по очкам.

Название	Шанс выйти в плей-офф (%)
Toronto Maple Leafs	100.0
Montreal Canadiens	63.75
Seattle Kraken	62.5
San Jose Sharks	61.25
Boston Bruins	60.0
Edmonton Oilers	60.0
Tampa Bay Lightning	60.0
Carolina Hurricanes	58.75
Philadelphia Flyers	58.75
Dallas Stars	57.5
St. Louis Blues	57.5
Florida Panthers	56.25
Anaheim Ducks	55.0
Buffalo Sabres	53.75
Chicago Blackhawks	53.75
Columbus Blue Jackets	53.75
New Jersey Devils	53.75
Vegas Golden Knights	52.5
Arizona Coyotes	52.5
Colorado Avalanche	52.5
Los Angeles Kings	52.5
Detroit Red Wings	51.25
New York Rangers	51.25
Vancouver Canucks	50.0
Nashville Predators	47.5
Ottawa Senators	47.5
Pittsburgh Penguins	47.5
Winnipeg Jets	46.25
New York Islanders	45.0
Minnesota Wild	0.0
Calgary Flames	0.0
Washington Capitals	0.0

Рисунок 4.10 – шанс выйти в плей-офф у команд при 90% завершенности регулярного чемпионата

Следующие данные представлены для регулярного сезона, завершившегося полностью, то есть все команды сыграли все 86 матчей. Статистика команд представлена на рисунке 4.11.

N	Название	Страна	Количество игр	Победы	Поражения	Очки	Разница шайб
1	Anaheim Ducks	Canada	86	51	35	102	49
2	Arizona Coyotes	Canada	86	46	40	92	-45
3	Boston Bruins	USA	86	48	38	96	40
4	Buffalo Sabres	Canada	86	46	40	92	-14
5	Calgary Flames	Canada	86	36	50	72	22
6	Carolina Hurricanes	USA	86	57	29	114	-9
7	Chicago Blackhawks	USA	86	46	40	92	-25
8	Colorado Avalanche	Canada	86	46	40	92	-7
9	Columbus Blue Jackets	USA	86	47	39	94	-48
10	Dallas Stars	USA	86	49	37	98	12
11	Detroit Red Wings	Canada	86	48	38	96	18
12	Edmonton Oilers	USA	86	53	33	106	-27
13	Florida Panthers	Canada	86	51	35	102	12
14	Los Angeles Kings	Canada	86	43	43	86	-27
15	Minnesota Wild	Canada	86	37	49	74	20
16	Montreal Canadiens	Canada	86	54	32	104	-14
17	Nashville Predators	Canada	86	41	45	82	-37
18	New Jersey Devils	Canada	86	46	40	92	30
19	New York Islanders	Canada	86	37	49	74	45
20	New York Rangers	USA	86	43	43	86	-25
21	Ottawa Senators	USA	86	42	44	84	-1
22	Philadelphia Flyers	USA	86	48	38	96	-19
23	Pittsburgh Penguins	USA	86	47	39	84	-49
24	San Jose Sharks	Canada	86	53	33	106	-1
25	Seattle Kraken	USA	86	52	34	104	37
26	St. Louis Blues	Canada	86	46	40	92	14
27	Tampa Bay Lightning	Canada	86	50	36	100	-26
28	Toronto Maple Leafs	USA	86	56	30	108	-46
29	Vancouver Canucks	USA	86	46	40	92	-12
30	Vegas Golden Knights	USA	86	44	42	88	44
31	Washington Capitals	USA	86	30	56	60	46
32	Winnipeg Jets	Canada	86	42	44	84	34

Рисунок 4.11 – данные о статистике команд для 100% завершенности
регулярного сезона

На рисунке 4.12 представлены шансы всех команд на выход в плей-офф при полностью завершенном сезоне. Нулевой шанс на выход в плей-офф оказался у команд, количество очков которых не позволило им пройти в 16 первых команд. Стопроцентный шанс выхода в плей-офф у команд, которые по очкам находятся первых 16-ти строчках. Таким образом, данная таблица показывает команды, которые вышли в плей-офф.

Название	Шанс выйти в плей-офф (%)
Carolina Hurricanes	100.0
Toronto Maple Leafs	100.0
Edmonton Oilers	100.0
San Jose Sharks	100.0
Montreal Canadiens	100.0
Seattle Kraken	100.0
Anaheim Ducks	100.0
Florida Panthers	100.0
Tampa Bay Lightning	100.0
Dallas Stars	100.0
Boston Bruins	100.0
Detroit Red Wings	100.0
Philadelphia Flyers	100.0
Columbus Blue Jackets	100.0
Arizona Coyotes	100.0
Buffalo Sabres	100.0
Chicago Blackhawks	0.0
Colorado Avalanche	0.0
New Jersey Devils	0.0
St. Louis Blues	0.0
Vancouver Canucks	0.0
Vegas Golden Knights	0.0
Los Angeles Kings	0.0
New York Rangers	0.0
Winnipeg Jets	0.0
Ottawa Senators	0.0
Pittsburgh Penguins	0.0
Nashville Predators	0.0
Minnesota Wild	0.0
New York Islanders	0.0
Calgary Flames	0.0
Washington Capitals	0.0

Рисунок 4.12 – шанс выйти в плей-офф у команд при 100% завершенности регулярного чемпионата

4.2 Выводы из экспериментального раздела

Проанализировав результаты, можно сделать вывод, что процент завершенности регулярного сезона сильно влияет на шанс выйти в плей-офф менее успешных команд и слабо влияет на шанс выйти в плей-офф у более успешных. Также с увеличением количества сыгранных матчей шансы у ряда команд уже становятся стопроцентами или же нулевыми еще до окончания сезона и становится возможным понять, кто действительно выйдет в плей-офф, а кто нет.

ЗАКЛЮЧЕНИЕ

Во время выполнения курсового проекта были рассмотрены существующие виды СУБД, описана структура базы данных и приложения. Были разработаны база данных и программное обеспечение, предоставляющее к ней интерфейс.

Программа реализована таким образом, что пользователь может получать информацию о статистике команд и хоккеистов Национальной Хоккейной Лиги, сравнивать эту статистику, осуществлять поиск хоккеистов и команд, создавать список избранных хоккеистов и команд, просматривать шансы команд на выход в плей-офф в текущий момент; модератор может добавлять и удалять хоккеистов и команды, а также обновлять их статистику; администратор может изменять права пользователей и удалять их.

В ходе выполнения поставленной задачи были изучены возможности языка Python, получен опыт работы с PostgreSQL и получены знания в области баз данных.

Цель работы достигнута, выполнены следующие задачи:

- 1) формализовано задание, определен необходимый функционал;
- 2) проведен анализ СУБД;
- 3) описана структуру базы данных, включая объекты, из которых она состоит;
- 4) спроектировано приложение для доступа к БД;
- 5) создана и заполнена БД;
- 6) разработано программное обеспечение, которое позволит пользователю получать, искать и сравнивать статистику игроков и команд Национальной Хоккейной Лиги;
- 7) исследована зависимость шанса на выход в плей-офф команд от их статистики в регулярном чемпионате и количестве сыгранных матчей.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. NHL. [Электронный ресурс]. Режим доступа: <https://www.nhl.com/> (дата обращения 21.06.2022)
2. Flashscore. [Электронный ресурс]. Режим доступа: <https://www.flashscore.ru.com/> (дата обращения 21.06.2022)
3. EliteProspects. [Электронный ресурс]. Режим доступа: <https://www.eliteprospects.com/> (дата обращения 21.06.2022)
4. ISO/IEC TR 10032:2003 Information technology — Reference model of data management
5. Дейт К. Дж. Введение в системы баз данных. — 8-е изд. — М.: «Вильямс», 2006.
6. Лазицкас Е.А., Загумённикова И.Н., Гилевский П.Г. Базы данных и системы управления базами данных. — 2-е изд. — Минск : Республиканский институт профессионального образования (РИПО), 2018. — 268 с.
7. Бородина А. И. Реляционная модель данных [Электронный ресурс]. Режим доступа: http://www.bseu.by/it/tohod/lekci2_3.htm (дата обращения 21.06.2022)
8. PostgreSQL: Документация. [Электронный ресурс]. Режим доступа: <https://postgrespro.ru/docs/postgresql/> (дата обращения: 22.06.2022)
9. Python: Документация. [Электронный ресурс]. Режим доступа: <https://docs.python.org/3/> (дата обращения: 22.06.2022)
10. Visual Studio Code: Документация. [Электронный ресурс]. Режим доступа: <https://code.visualstudio.com/docs> (дата обращения 22.06.2022)

ПРИЛОЖЕНИЕ А