



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

ФАКУЛЬТЕТ

ИУ «ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ»

КАФЕДРА

ИУ7 «ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

КУРСОВАЯ РАБОТА

НА ТЕМУ:

Разработка базы данных соревновательной игры

Студент

ИУ7-61Б

(группа)

(подпись, дата)

(И.О. Фамилия)

Руководитель курсового
проекта

(подпись, дата)

Волкова Л.Л.

(И.О. Фамилия)

Консультант

(подпись, дата)

(И.О. Фамилия)

2025 г.

РЕФЕРАТ

Расчетно-пояснительная записка 18 с., 5 рис., 3 источников, 1 прил.

KEYWORD1 KEYWORD2 KEYWORD3.

Объектами исследования стали TODO

Цель работы — TODO

В процессе работы составлены TODO

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	7
1 Аналитическая часть	8
1.1 Формализация соревновательной игры	8
1.2 Формализация данных	8
1.3 Категории пользователя	9
1.4 Выбор модели данных	10
2 Конструкторская часть	11
2.1 Проектируемая база данных	11
2.2 Функциональная модель	12
2.3 Ролевая модель	13
3 Технологическая часть	14
3.1 Тестирование	14
4 Исследовательская часть	15
ЗАКЛЮЧЕНИЕ	16
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	17
ПРИЛОЖЕНИЕ А	18

ВВЕДЕНИЕ

Целью курсовой работы является разработка базы данных соревновательной игры.

Для достижения поставленной цели необходимо решить следующие задачи:

- formalизовать соревновательную игру, описать предметную область
- провести анализ существующих решений
- формализовать сущности базы данных
- спроектировать архитектуру базы данных и ограничения целостности

1 Аналитическая часть

Были выбраны соревновательные и обучающие игры. Составлены следующие критерии:

- Наличие временных соревнований
- Внутриигровые вознаграждения
- Образовательная направленность
-

Сравнение представлено в таблице 1.1.

1.1 Формализация соревновательной игры

Соревновательная игра состоит из основной части, оцениваемая количеством очков (целым числом), и заключительной, где очки игроков сравниваются между собой.

Соревнование характеризуется сроками проведения (начало и конец), информацией о выдаваемых наградах, а также параметрами игры. Выдаваемые награды характеризуются собственно наградой, а также критерием выдачи исходя из таблицы лидеров. Возможны следующие критерии:

- место в таблице лидеров (от минимального до максимального);
- перцентиль в таблице лидеров (от минимальной до максимальной доли участников).

Следует предусмотреть, что при развитии приложения возможно добавление как новых видов критериев, так и новых форматов параметров игры.

В основной части игрок зарабатывает очки, решая математические примеры на скорость, указанные в параметрах соревнования. Результат, как и время подачи результата сохраняется в таблицу лидеров соответствующего соревнования.

Заключительная часть наступает по истечению срока соревнования. Составляется таблица лидеров, сортируя игроков сначала по убыванию игрового счёта, потом по возрастанию времени подачи последнего результата. Награды выдаются игрокам, исходя из таблицы лидеров.

В рамках поставленной цели требуется разработать базу данных, содержащую информацию о соревнованиях, об игроках, о их участии в соревнованиях, а также информацию о выданных наградах и выдаваемых наградах соревнования.

1.2 Формализация данных

Разрабатываемая база данных должна содержать информацию об игроках, профилей игроков, соревнований, результатах игры, наград. Информация об игровом уровне хранится как Сущности базы данных представлены в таблице 1.2

Таблица 1.1 — Сравнение существующих решений

1	2	3	
---	---	---	--

Таблица 1.2 — Описание сущностей базы данных

Сущность	Данные
Аккаунт	Логин, почта, права доступа, пароль
Профиль	Имя, описание, изображение
Соревнование	Имя, описания, даты начала и конца, описание игрового уровня, выданы ли
Тип награды	Имя, описание, редкость, изображение, внутриигровое представление
Награда	Дата выдачи. Ссылается на тип награды, соревнование и игрока
Награда соревнования	Критерий выдачи. Ссылается на тип награды и соревнование

Диаграмма сущностей базы данных представлена на рисунке 1.1

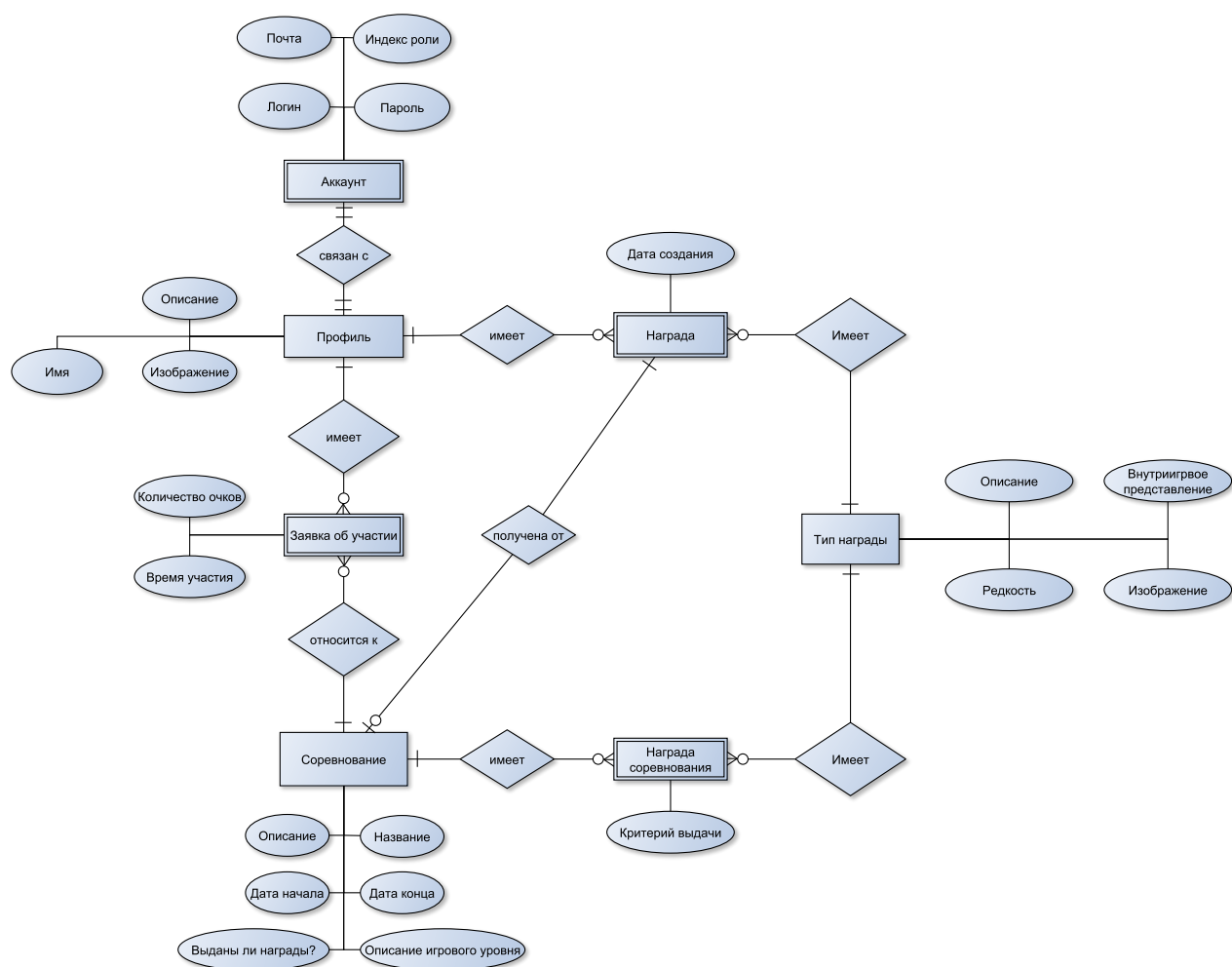


Рисунок 1.1 — Диаграмма сущностей БД в нотации Чена

1.3 Категории пользователя

В рамках задачи было выделено три категории пользователя:

- гость,
- игрок,
- администратор.

Все категории пользователя имеют возможность просматривать профили игроков, таб-

лицы лидеров соревнования, а также собственно соревнований (данные о сроках проведения и наградах)

Гость имеет возможность авторизации и создания аккаунта. При авторизации гость может стать игроком или администратором.

Игрок может просматривать свои награды, редактировать свой профиль, а также участвовать в соревнованиях.

Администратор может:

- назначать и отзывать награды игрока;
- создавать и редактировать соревнования;
- отзывать нежелательные результаты, удаляя их;
- создавать и редактировать типы наград.

Игрок и администратор имеют возможность выйти из аккаунта, в последствии чего устанавливается категория пользователя "гость".

Диаграмма пользования БД представлена на рисунке 1.2.

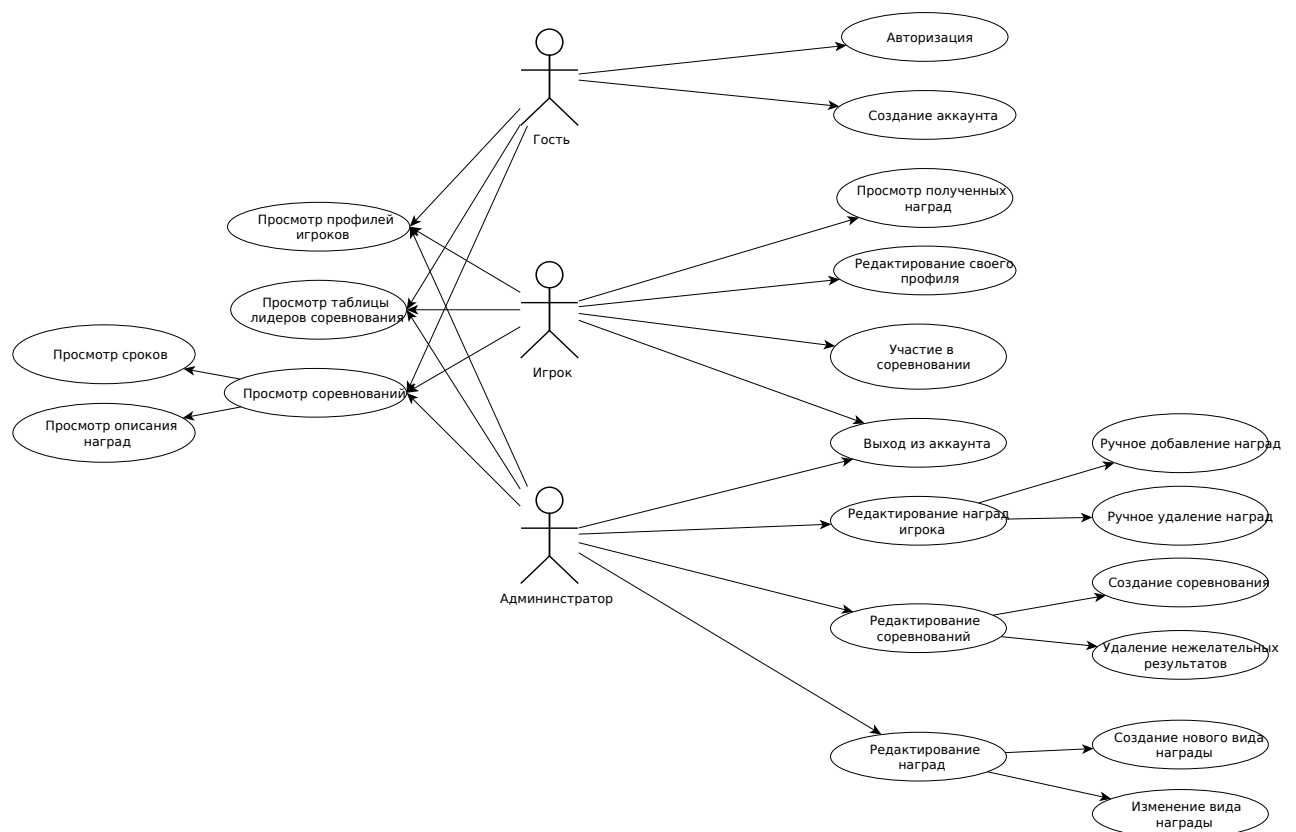


Рисунок 1.2 — Диаграмма пользования БД

1.4 Выбор модели данных

Вывод

TODO

2 Конструкторская часть

2.1 Проектируемая база данных

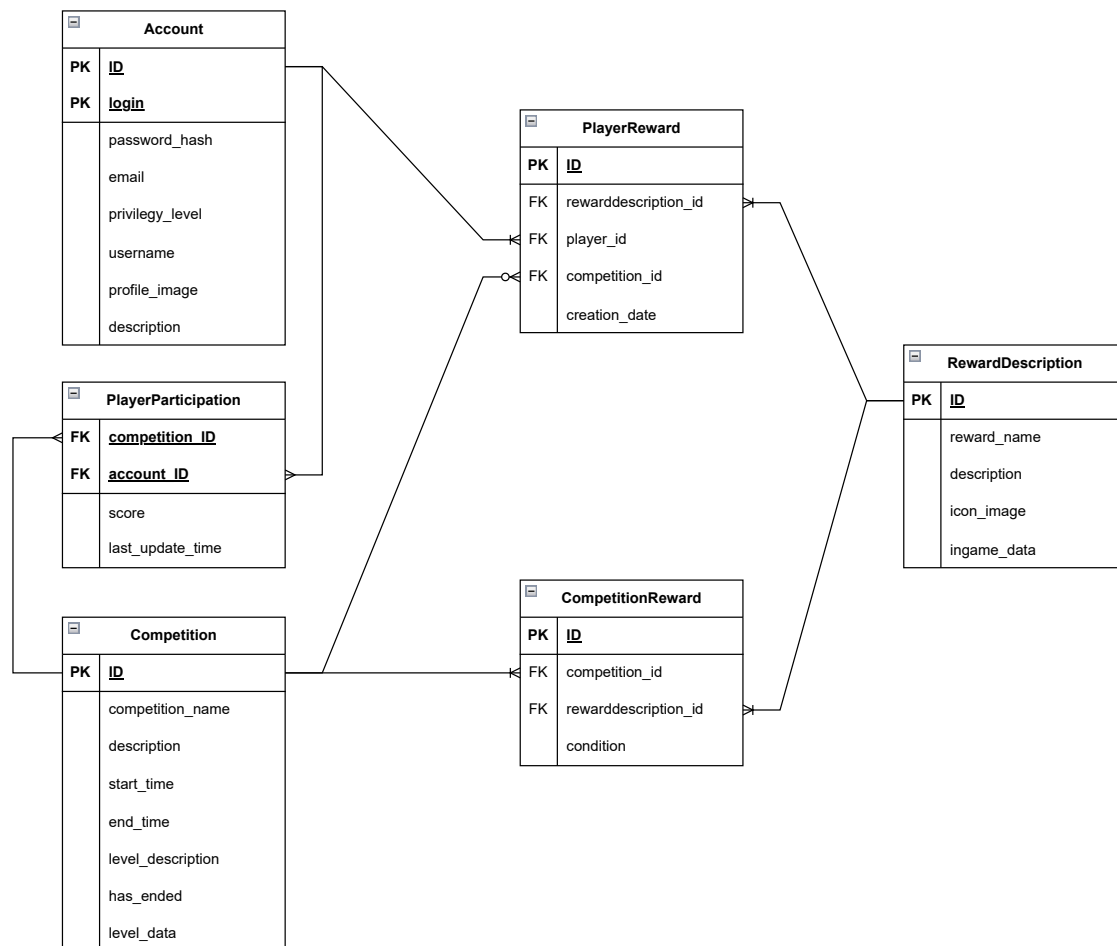


Рисунок 2.1 — Схема проектируемой базы данных

2.2 Функциональная модель

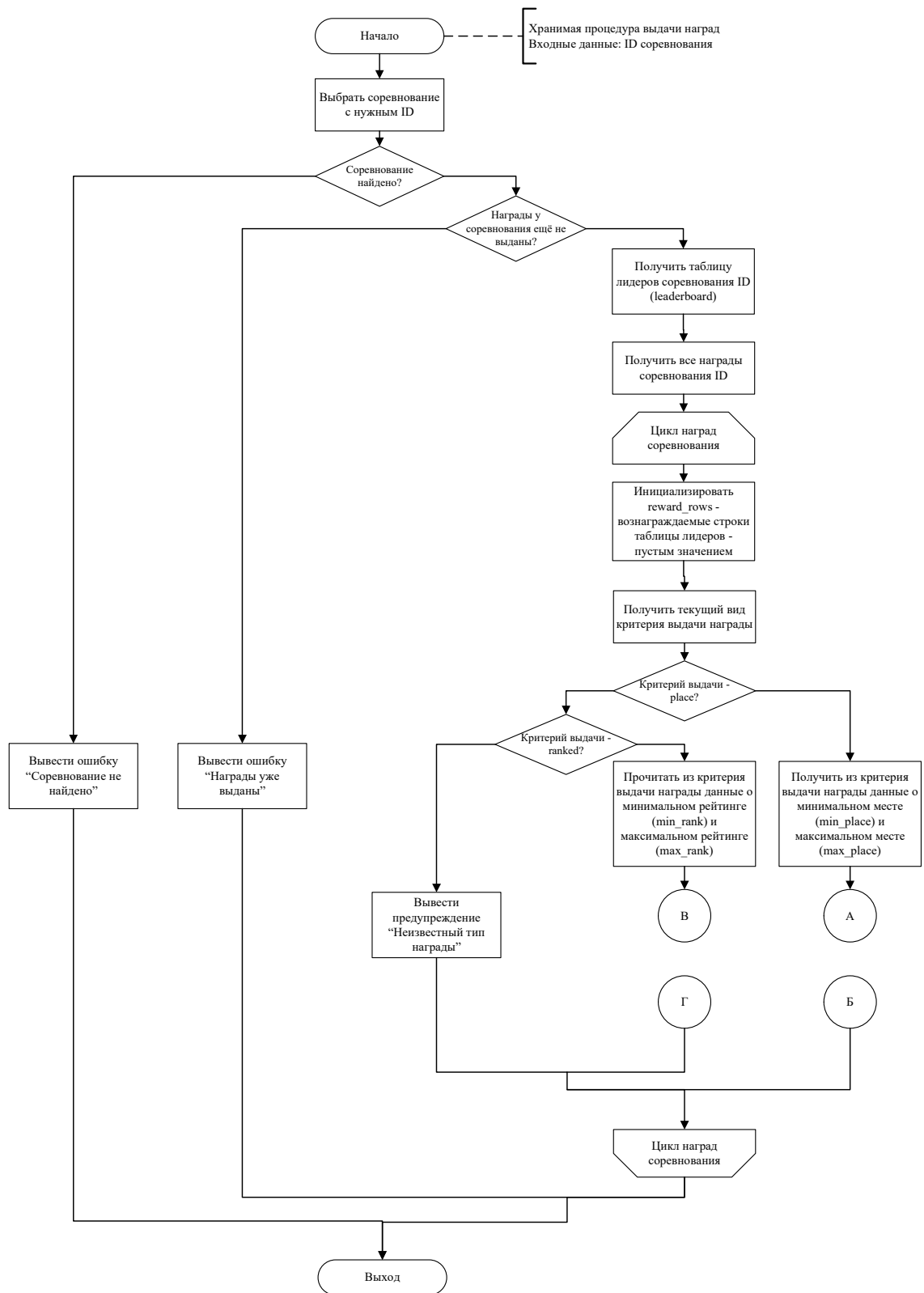


Рисунок 2.2 — Хранимая процедура БД

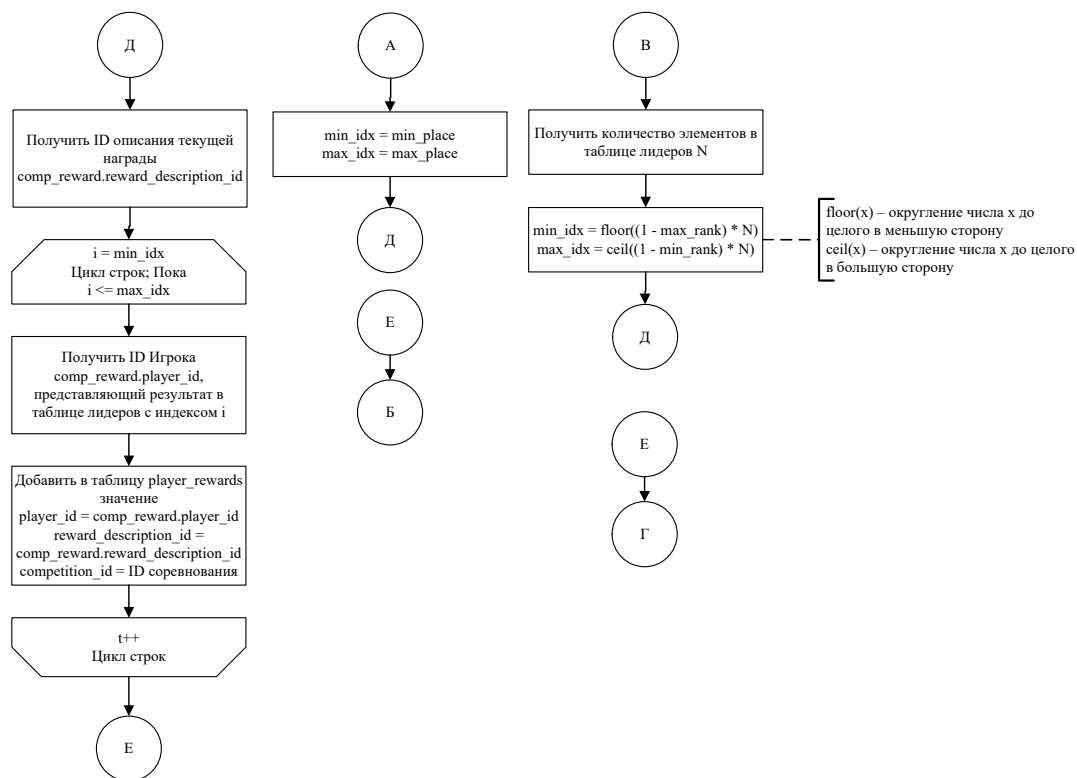


Рисунок 2.3 — Хранимая процедура БД

2.3 Ролевая модель

В рамках задачи было выделено три категории пользователя:

- гость,
- игрок,
- администратор,
- демон выдачи наград.

Вывод

TODO

3 Технологическая часть

В качестве СУБД была выбрана реляционная система управления базами данных PostgreSQL, поскольку в ней представлены все необходимые возможности, в частности создание хранимых процедур.

Для реализации приложения был выбран язык C# версии 11.0 вместе с фреймворком ASP.NET Core, поскольку в нём представлены все необходимые возможности для создания программного интерфейса взаимодействия с базой данных.

Для интерфейса доступа к СУБД был выбран Entity Framework Core, поскольку он предоставляет возможность взаимодействия с объектами базы данных как с классами C#.

3.1 Тестирование

Для тестирования приложения были составлены модульные тесты, проверяющие различные слои:

—

Вывод

TODO

4 Исследовательская часть

Цель исследования - сравнить зависимость времени выдачи наград от количества вознаграждаемых игроков в двух случаях: при помощи хранимой процедуры и при помощи обычных запросов.

Описание способов выдачи наград:

- хранимая процедура — вызов хранимой процедуры (один запрос на СУБД);
- обычные запросы — реализация логики добавления наград с помощью EF Core (множество запросов в СУБД).

Замеры времени нуждаются в нескольких вариантах изначальных входных данных, что требует создание независимых вариантов базы данных, инициализируемые различными параметрами. Для создания независимых образов PostgreSQL использовалась библиотека Testcontainers.Po

Исследование проводилось в два этапа: подготовка данных и замеры времени.

На каждую подготовку данных выделялся отдельный образ PostgreSQL, которым был инициализирован сценарием (?). Программа заполняла СУБД нужными данными, после чего делала снимок базы данных при помощи `pg_dump` [1] в файл. Для заполнения базы данных случайными данными использовалась библиотека `Vogus` [2].

Если снимок данных подготавливался для N выдаваемых наград, генерировалось:

- 1 соревнование;
- $N/2$ участников и записей в таблицы лидеров;
- 5 критериев наград, каждый из которых вознаграждает $\frac{N}{5}$ участников.

Критерии выдачи награды генерировались с равной вероятностью двух возможных видов: по диапазону доли в таблице лидеров, по диапазону мест.

На каждый замер времени выделялся отдельный образ PostgreSQL, который был инициализирован снимком базы данных, полученным на этапе подготовке данных. После этого, выполнялись замеры времени работы или хранимой процедуры, или обычных запросов. Для измерения реального времени работы использовался класс `Stopwatch` [3]

Вывод

TODO

ЗАКЛЮЧЕНИЕ

Были выполнены следующие задачи:

— DONE1

— DONE2

Исследование DONE

Цель DONE

Поставленные задачи выполнены. Цель работы достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация команды `pg_dumpPostgreSQL`. — Режим доступа: <https://www.postgresql.org/docs/current/app-pgdump.html> (дата обращения: 05.05.2025).
2. Описание библиотеки Bogus [Электронный ресурс]. — Режим доступа: <https://www.nuget.org/packages/bogus> (дата обращения: 05.05.2025).
3. Документация класса `System.Diagnostics.Stopwatch` [Электронный ресурс]. — Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/api/system.diagnostics.stopwatch?view=net-8.0> (дата обращения: 05.05.2025).

ПРИЛОЖЕНИЕ А

Презентация состоит из 16 слайдов