Министерство образования, науки и молодежной политики Нижегородской области

Государственное бюджетное профессиональное образовательное учреждение «Нижегородский Губернский колледж»

ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ

**ПМ. 01. Разработка модулей программного обеспечения для компьютерных систем**

**Специальность** **09.02.07 Информационные системы и программирование**

|  |  |
| --- | --- |
| Выполнил |  |
| Студент | Г.А.Куклев |
| подпись |  |
| Группа 41П курс 4 |  |
| Руководитель | Е.П.Голубева |
| подпись |  |
| Руководитель | В.В.Циркова |
| подпись |  |
| Руководитель | М.В.Полетаева |
| подпись |  |
|  |  |

Нижний Новгород 2025

**СОДЕРЖАНИЕ**

**[1. ОСНОВНЫЕ ПОНЯТИЯ 4](#_Toc192267624)**

[**2. РАЗРАБОТКА БАЗЫ ДАННЫХ 6**](#_Toc192267625)

[**3. РАЗРАБОТКА ПРИЛОЖЕНИЯ 8**](#_Toc192267626)

[**4. РАЗРАБОТКА БИБЛИОТЕКИ КЛАССОВ И ТЕСТИРОВАНИЕ 14**](#_Toc192267627)

[**ПРИЛОЖЕНИЕ А 17**](#_Toc192267628)

[**DataDictionary БД 17**](#_Toc192267629)

[**Руководство пользователя 19**](#_Toc192267630)

[**Ручное тестирование 23**](#_Toc192267631)

[**ПРИЛОЖЕНИЕ Б 30**](#_Toc192267632)

[**Листинг 1 - SQL-скрипт БД 30**](#_Toc192267633)

[**Листинг 2 - Код функций библиотек 86**](#_Toc192267634)

**ВВЕДЕНИЕ**

Учебная практика несет в себе цель закрепления и развития как теоретических, так и практических навыков, чтобы применять на производстве. В ходе учебной практики было закреплены и развиты навыки работы с Framework C# «Avalonia UI», с базой данных PostgreSQL, с Framework C# «EntityFramework».

Учебная практика является возможность получить навыки работы с Framework C# «Avalonia UI» и Framework C# «EntityFramework».

Целью учебной практики является освоение навыков работы с основными командами, функциями, программированием «Avalonia UI» и «EntityFramework»

Задачи учебной практики является получение практики в работе с инструментами «Avalonia UI» и «EntityFramework» в ходе выполнения задания по программированию, передаче данных, тестировании системы.

В ходе учебной практики будут получены все необходимые навыки и знания для решения задачи на производстве.

# **ОСНОВНЫЕ ПОНЯТИЯ**

* 1. **Фреймворк C# «Avalonia UI»**

Avalonia UI — это кроссплатформенный фреймворк для создания графических интерфейсов (GUI) на C#.

Avalonia UI позволяет разрабатывать десктопные приложения, работающие на Windows, macOS, Linux, а также встраиваемые системы (например, Raspberry Pi) и браузерах (через WebAssembly). Avalonia использует подход, аналогичный WPF (Windows Presentation Foundation), но с поддержкой множества платформ.

Особенности Avalonia UI:

1. Единая кодовая база для всех поддерживаемых ОС.
2. Использует декларативный язык разметки для описания UI, что упрощает создание сложных интерфейсов.
3. Интеграция с паттерном Model-View-ViewModel для разделения логики и представления.
4. Рендеринг через Skia, что обеспечивает высокую производительность.
5. Полная прозрачность и возможность кастомизации.
6. Гибкие стили, анимации и темы.
7. Совместимость с современными версиями .NET.
   1. **База данных PostgreSQL**

PostgreSQL — это объектно-реляционная СУБД с открытым исходным кодом, предназначенная для хранения, управления и обработки структурированных данных. Она подходит для проектов с высокими требованиями к надежности, масштабируемости и поддержке сложных запросов.

Особенности PostgreSQL:

1. Поддержка пользовательских типов данных, функций и расширений (например, PostGIS для геоданных).
2. Гарантии атомарности, согласованности, изоляции и долговечности транзакций.
3. Встроенная работа с полуструктурированными данными.
4. Поддержка горизонтального масштабирования.
5. Встроенные инструменты для текстовой аналитики.
6. Ролевая модель доступа, SSL-шифрование, аудит.
7. Оптимизатор запросов, индексы (B-tree, Hash, GIN, GiST).
   1. **Фреймворк C# «Entity Framework Core»**

Entity Framework (EF) Core — это ORM (Object-Relational Mapper) для C#, который упрощает взаимодействие с базами данных. Он позволяет работать с данными как с объектами C#, автоматизируя CRUD-операции и маппинг сущностей на таблицы БД.

Особенности Entity Framework (EF) Core:

1. Работает на .NET Core, .NET 5+ и поддерживает различные СУБД, включая PostgreSQL.
2. Написание запросов на языке C# вместо SQL.
3. Автоматическое управление схемой БД через код (например, Add-Migration и Update-Database).
4. Автоматическое отслеживание изменений в объектах для их синхронизации с БД.
5. Поддержка async/await для неблокирующих запросов.
6. Гибкая конфигурация через Fluent API или атрибуты.
7. Кэширование, оптимизация запросов, пакетные операции.

# **РАЗРАБОТКА БАЗЫ ДАННЫХ**

Для функционирования системы необходимо было создать базу данных по Excel файлам. Было решено создавать базу данных PostgreSQL, настраивать ее через СУБД «DBeaver». Исходя из данных Excel-файлов, было создано 15 таблиц.

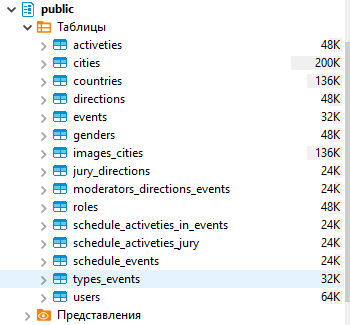


Рисунок 1 – Таблицы в DBeaver

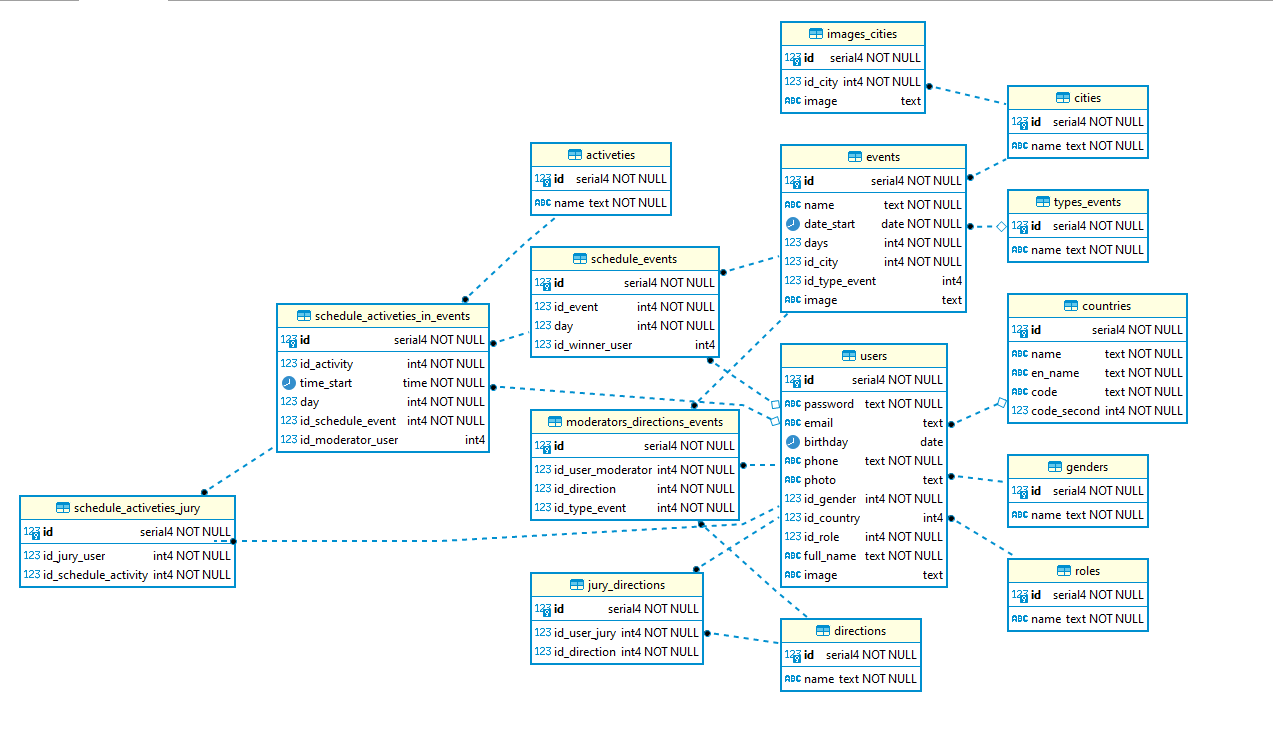


Рисунок 2 – ER-диаграмма БД

Также для упрощения БД был создан файл «DataDictionary.xlsx». Содержимое файла «[см. ПРИЛОЖЕНИЕ А. DataDictionary БД](#_DataDictionary_БД)»

Все таблицы были заполнены данными из представленных Excel-файлов. Полный sql-скрипт БД «[см. ПРИЛОЖЕНИЕ Б. Листинг 1 - SQL-скрипт БД](#_Листинг_1_-)»

# **РАЗРАБОТКА ПРИЛОЖЕНИЯ**

Перед непосредственной разработкой приложения была разработана диаграмма прецедентов, чтобы наглядно определить роли пользователей и их возможности работы

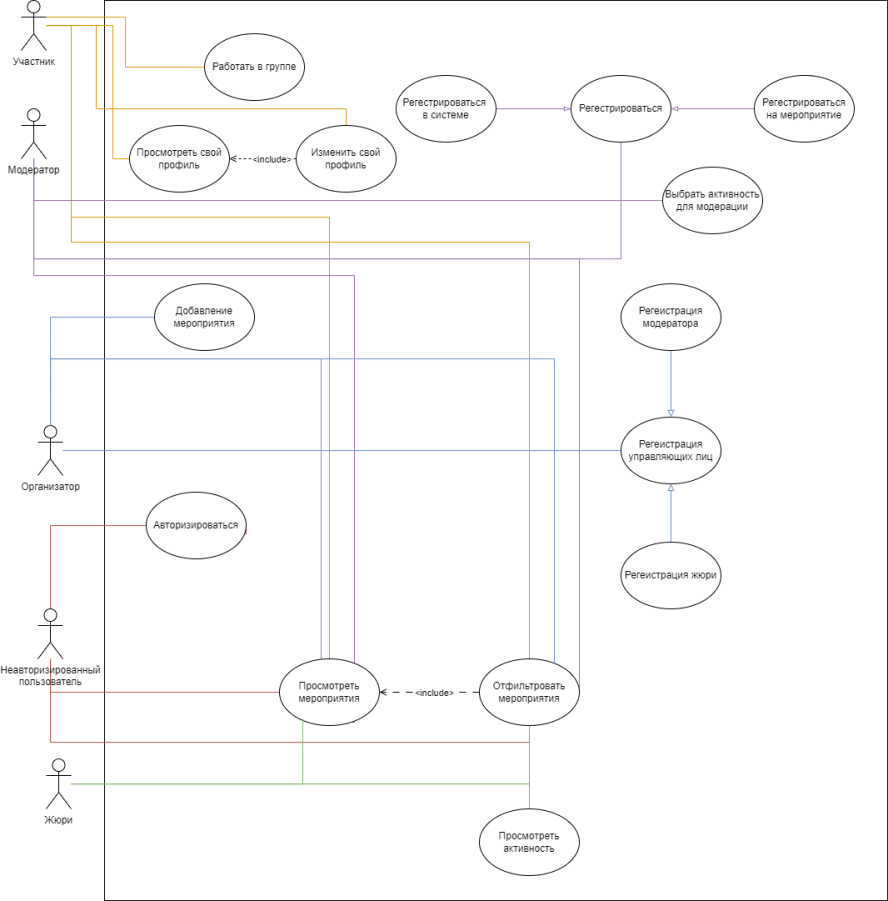


Рисунок 3 – Диаграмма прецедентов

После точного определения возможностей и ролей, был начат процесс создания desktop-приложения на Avalonia UI с использованием Entity Framework Core, чтобы подключаться к ранее разработанной базе данных.

Первым делом был разработан главный экран, на котором отображается:

1. Изображение мероприятия
2. Название мероприятия
3. Тип мероприятия
4. Дату мероприятия

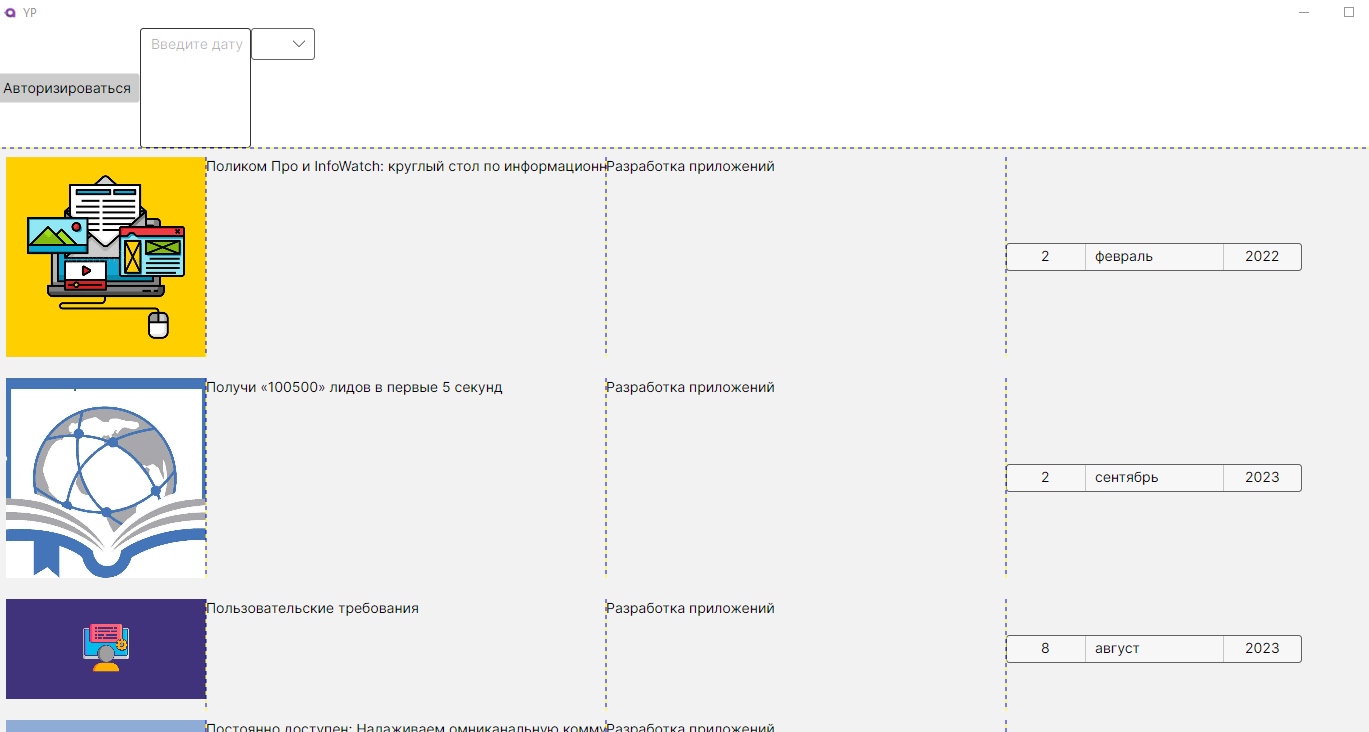


Рисунок 4. - Список мероприятий

На странице доступен функционал поиска мероприятий по дате и фильтрация по типу мероприятия

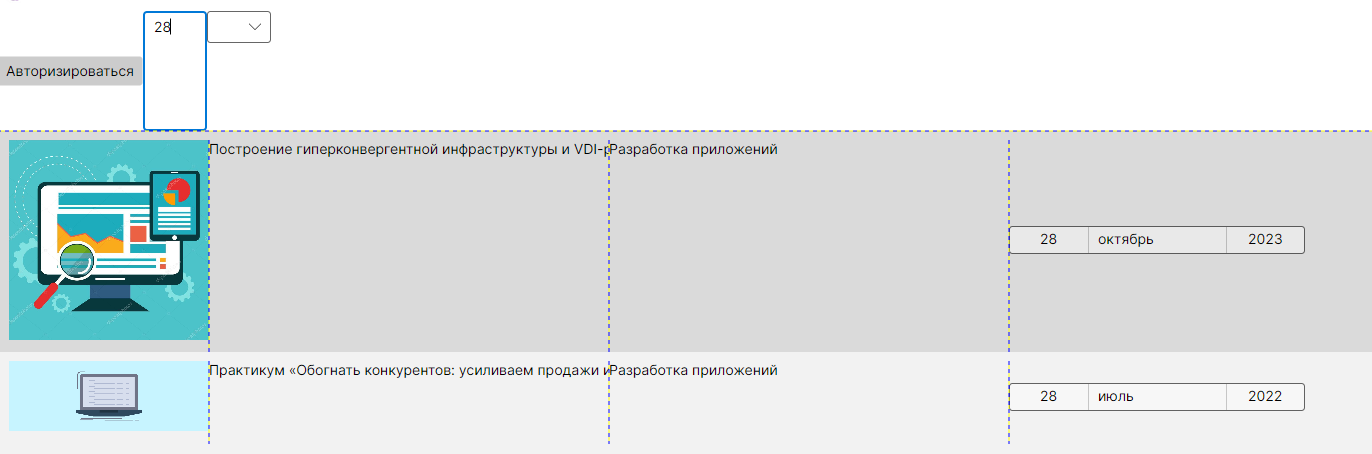


Рисунок 5. - Поиск по дате

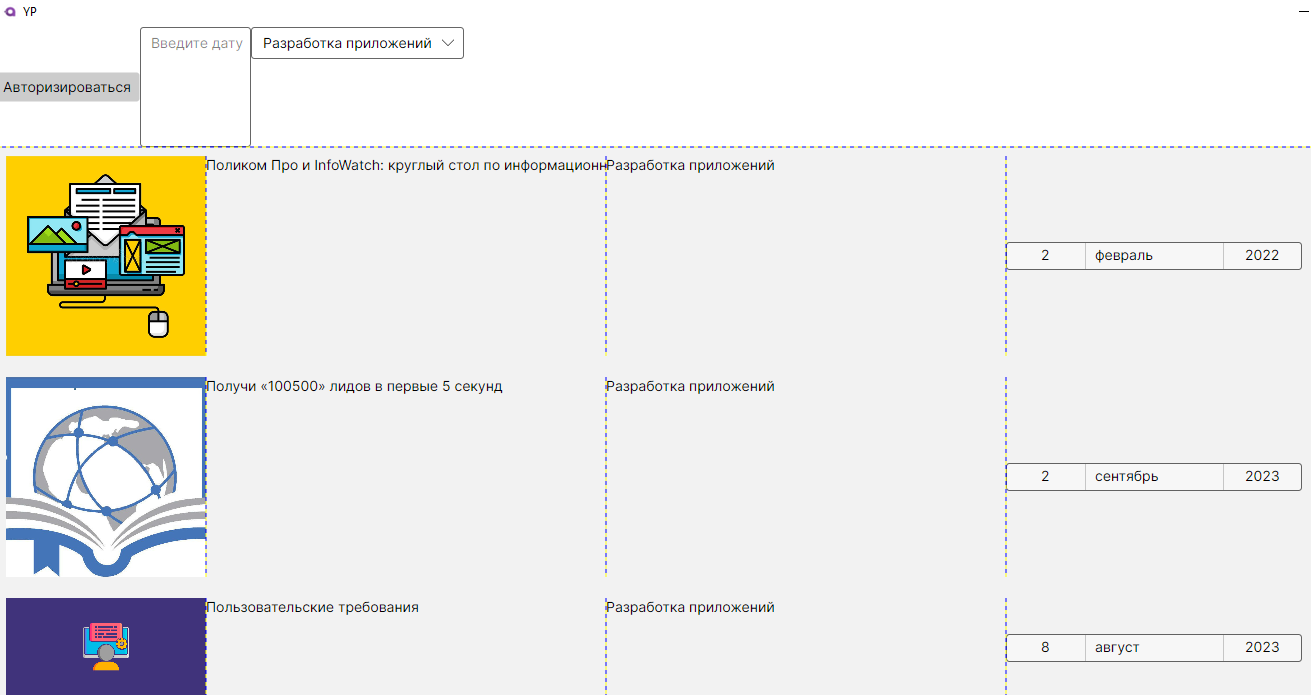


Рисунок 6 - Фильтрация по типу

Оба типа фильтрации работают параллельно.

Далее был разработан экран авторизации, содержит в себе следующие поля:

1. Айди пользователя
2. Пароль
3. Капча (Капча генерируется случайно после каждого открытия экрана авторизации)
4. Кнопка «Войти»

В поле «Айди пользователя» вводится уникальный идентификатор пользователя

В поле «Пароль» вводится пароль от учетной записи пользователя

В поле под капчей необходимо ввести комбинацию символов с изображения.

После 3 попыток неудачного входа система заблокируется и «Айди пользователя» должно быть только числом

После ввода корректных данных – будет произведен вход в систему (После первого входа система запомнит данные пользователя и сразу будет производить вход в систему)

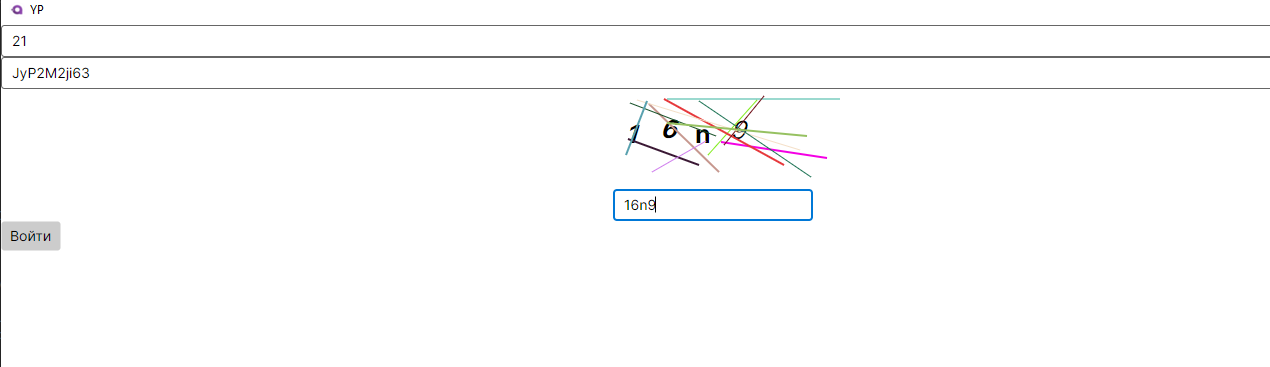


Рисунок 7 – Экран авторизации

Далее был разработан экран организатора, который имеет следующие элементы:

1. Изображение пользователя
2. Приветствие пользователя по ФИО
3. Кнопка «Зарегистрировать модератора/жюри»
4. Панели для работы организатора

Организатор может перейти к созданию нового жюри/модератора, нажав на кнопку «Зарегистрировать модератора/жюри».

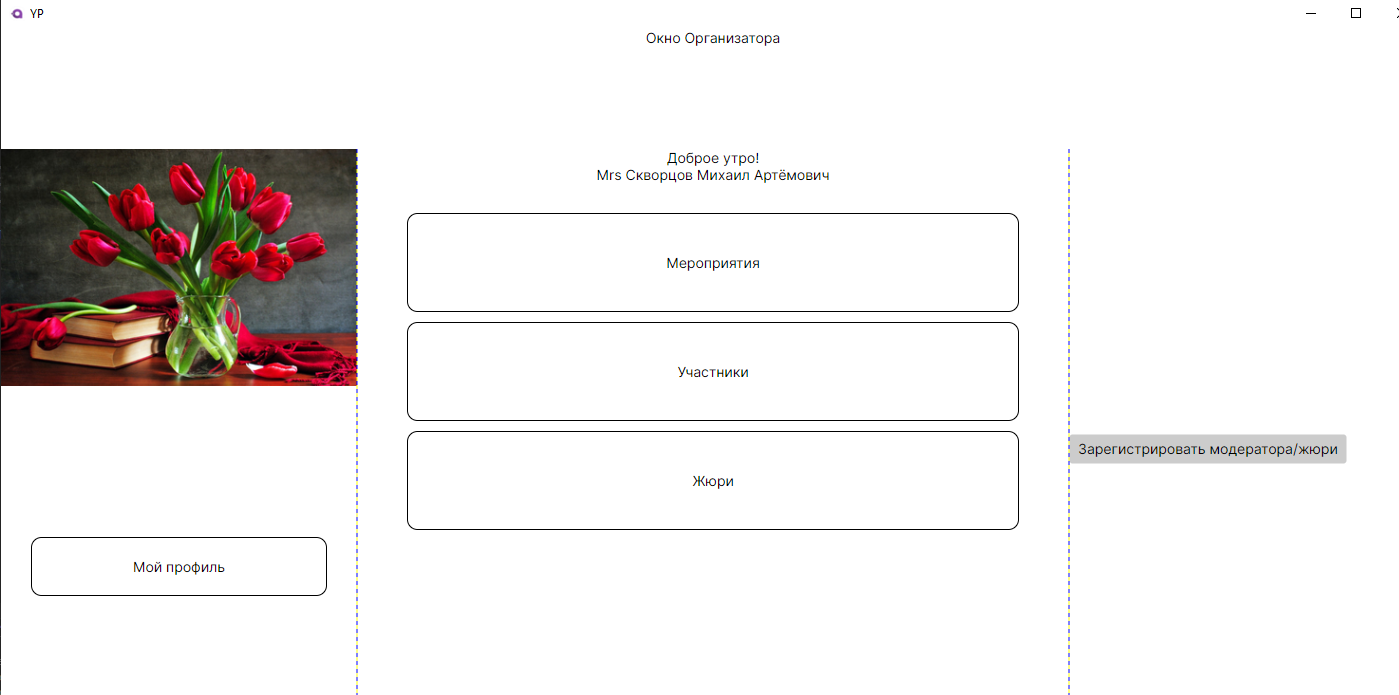


Рисунок 8 - Окно организатора

Далее был разработан экран создания жюри/модератора, на которой есть следующие элементы:

1. Поле «IdNumber» (не изменяется, программа генерирует автоматически)
2. Поле «ФИО»
3. Поле «Электронная почта»
4. Поле «Пол»
5. Поле «Роль»
6. Поле «Телефон»
7. Поле «Направление»
8. Поле «Мероприятие»
9. Кнопка «Загрузить изображение»
10. Поле «Пароль»
11. Поле «Повтор пароля»
12. Кнопка «Ок» (сохранить нового пользователя)
13. Кнопка «Отмена» (вернуться на экран пользователя)
14. Важно отметить, что:
15. При выборе роли «Жюри» можно выбрать только «Направление»
16. Пароли должны совпадать и содержать следующие параметры (Пароль не содержит не менее 6 символов; заглавные и строчные буквы; не менее одного спецсимвола; не менее одной цифры.)
17. Электронная почта должна соответствовать маске [test@gmail.com](mailto:test@gmail.com)
18. Пароли можно скрыть или отображать

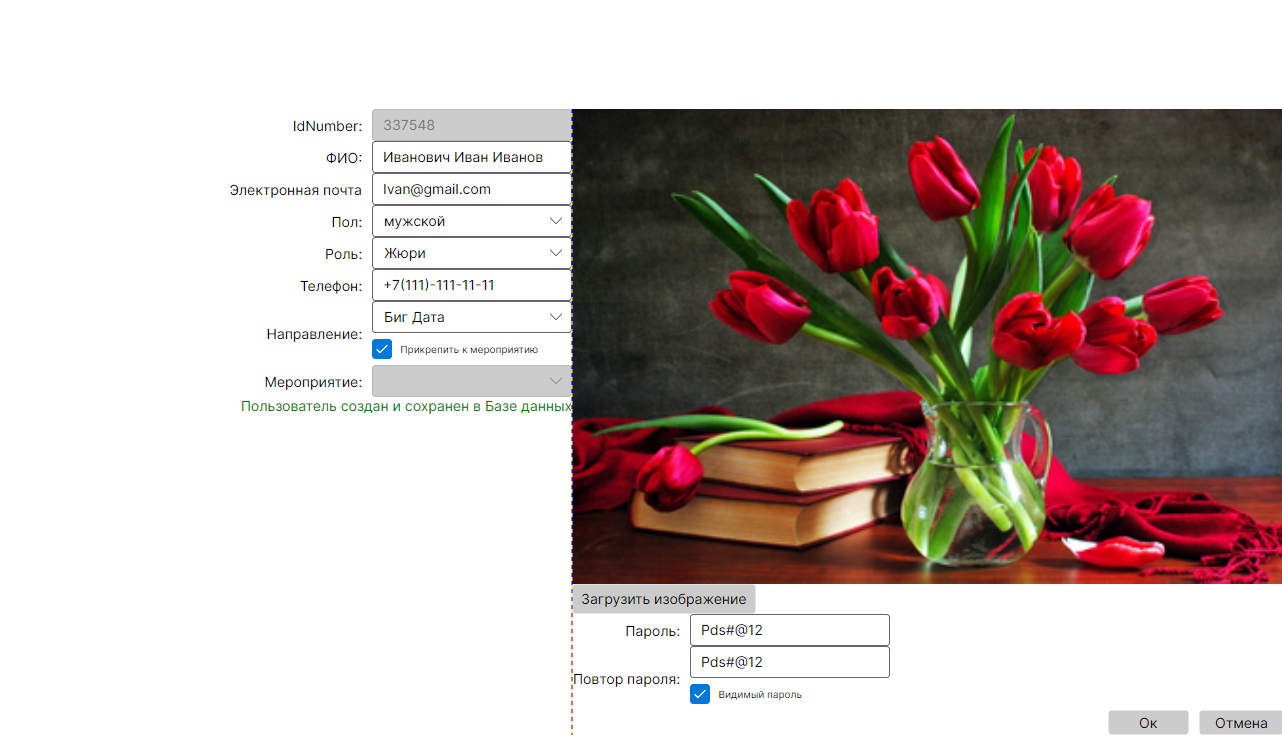


Рисунок 9 - Окно регистрации жюри/модератора

После создания программы было составлено руководство пользователя «см. ПРИЛОЖЕНИЕ А. Руководство пользователя»

Вся программа была реализована по паттерну MVVM, полный код всех View, Models, ViewModels можно просмотреть в репозитории по ссылке ( <https://github.com/CoolVery/Yp_Avalonia.git> ) в папке YP

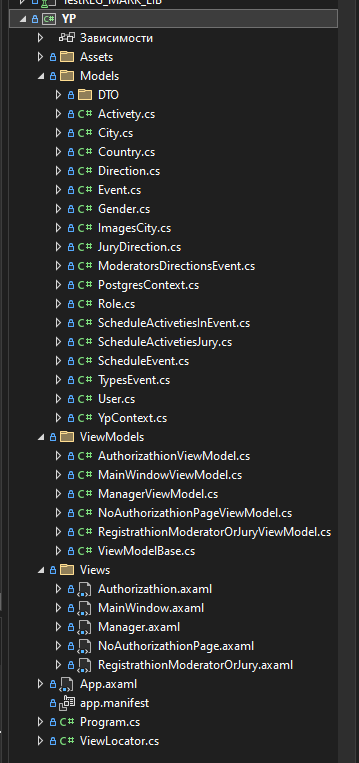


Рисунок 10 – Структура приложения в Visual Studio

# **РАЗРАБОТКА БИБЛИОТЕКИ КЛАССОВ И ТЕСТИРОВАНИЕ**

Было поставлено задание разработать библиотеку для проверки и генерации автомобильных знаков

Необходимо было создать библиотеку REG\_MARK\_LIB.dll или REG\_MARK\_LIB.jar, которая имеет следующие методы:

1. Boolean CheckMark(String mark) – данный метод проверяет переданный номерной знак в формате a999aa999 (латинскими буквами) и возвращает true или false в зависимости от правильности номерного знака. Метод должен учитывать также и существующие номера регионов.
2. String GetNextMarkAfter(String mark) – данный метод принимает номерной знак в формате a999aa999 (латинскими буквами) и выдает следующий номер в данной серии или создает следующую серию.
3. String GetNextMarkAfterInRange(String prevMark, String rangeStart, String rangeEnd) – данный метод принимает номерной знак в формате a999aa999 (латинскими буквами) и выдает следующий номер в данной данном промежутке номеров rangeStart до rangeEnd (включая обе границы). Если нет возможности выдать следующий номер, необходимо вернуть сообщение “out of stock”.
4. Integer GetCombinationsCountInRange(String mark1, String mark2) – данный метод принимает два номера в формате a999aa999 (латинскими буквами) и возвращает количество возможных номеров между ними (включая обе границы). Метод необходим, чтобы рассчитать оставшиеся свободные номера для региона.

Был создан проект «Библиотека классов» в которой были реализованы все функции, их код можно посмотреть в репозитории по ссылке ( <https://github.com/CoolVery/Yp_Avalonia.git> ) в папке REG\_MARK\_LIB и «[см. ПРИЛОЖЕНИЕ Б. Листинг 2. Код функций библиотек](#_Листинг_2_-)»

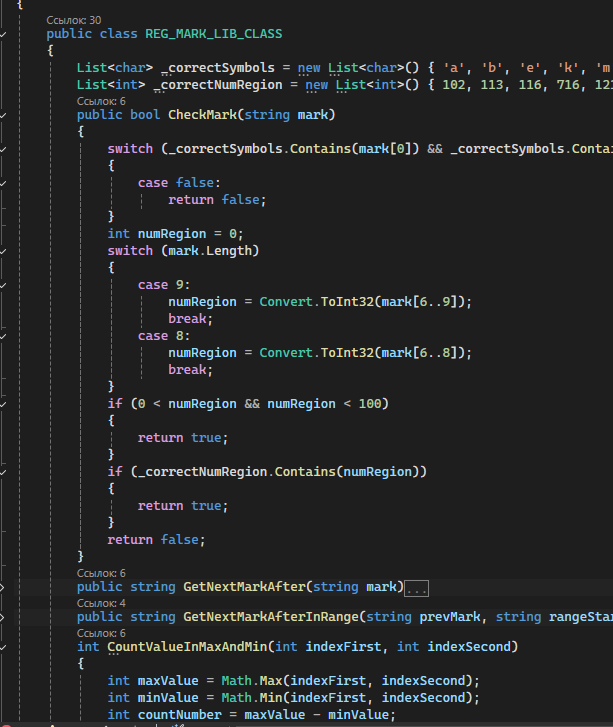


Рисунок 11 – Функции библиотеки

После разработки необходимо было протестировать функции библиотеки с помощью Unit-тестов. По итогу было разработано 10 простых и 5 сложных тестов. Код тестов можно просмотреть в репозитории по ссылке ( <https://github.com/CoolVery/Yp_Avalonia.git> ) в папке TestREG\_MARK\_LIB

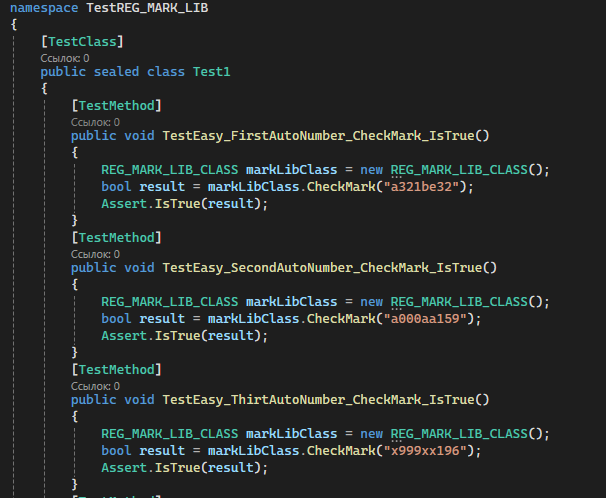


Рисунок 12 – Unit-тесты

После разработки desktop приложения необходимо было провести ручное тестирование механизма создания нового жюри/модератора. Было составлено 5 тест-кейсов с подробным описание, посмотреть можно «[см. ПРИЛОЖЕНИЕ А. Ручное тестирование](#_Ручное_тестирование)»

# **ПРИЛОЖЕНИЕ А**

# **DataDictionary БД**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблица activities | | | | |
| **Key** | **Field name** | **Data type / field size** | **REQUIRED?** | **Notes** |
| PK | id | serial4 | Y | Автоувеличение |
|  | name | text | Y |  |
|  |  |  |  |  |
| Таблица schedule\_activities\_in\_event | | | | |
| **Key** | **Field name** | **Data type / field size** | **REQUIRED?** | **Notes** |
| PK | id | serial4 | Y | Автоувеличение |
| FK | id\_activity | int4 | Y |  |
|  | time\_start | time | Y |  |
|  | day | int4 | Y |  |
| FK | id\_schedule\_event | int4 | Y |  |
| FK | id\_moderator\_user | int4 | Y |  |
|  |  |  |  |  |
| Таблица schedule\_activities\_jury | | | | |
| **Key** | **Field name** | **Data type / field size** | **REQUIRED?** | **Notes** |
| PK | id | int4 | Y | Автоувеличение |
| FK | id\_jury\_user | int4 | Y |  |
| FK | id\_schedule\_activity | int4 | Y |  |
|  |  |  |  |  |
| Таблица schedule\_events | | | | |
| **Key** | **Field name** | **Data type / field size** | **REQUIRED?** | **Notes** |
| PK | id | int4 | Y | Автоувеличение |
| FK | id\_event | int4 | Y |  |
|  | day | int4 | Y |  |
| FK | id\_winner\_user | int4 | Y |  |
|  |  |  |  |  |
| Таблица events | | | | |
| **Key** | **Field name** | **Data type / field size** | **REQUIRED?** | **Notes** |
| PK | id | serial4 | Y | Автоувеличение |
|  | name | text | Y |  |
|  | date\_start | date | Y |  |
|  | days | int4 | Y |  |
| FK | id\_city | int4 | Y |  |
| FK | id\_type\_event | int4 | Y |  |
|  |  |  |  |  |
| Таблица images\_cities | | | | |
| **Key** | **Field name** | **Data type / field size** | **REQUIRED?** | **Notes** |
| PK | id | serial4 | Y | Автоувеличение |
| FK | id\_city | int4 | Y |  |
|  | image | text | Y |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблица cities | | | | |
| **Key** | **Field name** | **Data type / field size** | **REQUIRED?** | **Notes** |
| PK | id | serial4 | Y | Автоувеличение |
|  | name | text | Y |  |
|  |  |  |  |  |
| Таблица types\_events | | | | |
| **Key** | **Field name** | **Data type / field size** | **REQUIRED?** | **Notes** |
| PK | id | serial4 | Y | Автоувеличение |
|  | name | text | Y |  |
|  |  |  |  |  |
| Таблица countries | | | | |
| **Key** | **Field name** | **Data type / field size** | **REQUIRED?** | **Notes** |
| PK | id | serial4 | Y | Автоувеличение |
|  | name | text | Y |  |
|  | en\_name | text | Y |  |
|  | code | text | Y |  |
|  | code\_second | int4 | Y |  |
|  |  |  |  |  |
| Таблица genders | | | | |
| **Key** | **Field name** | **Data type / field size** | **REQUIRED?** | **Notes** |
| PK | id | serial4 | Y | Автоувеличение |
|  | name | text | Y |  |
|  |  |  |  |  |
| Таблица roles | | | | |
| **Key** | **Field name** | **Data type / field size** | **REQUIRED?** | **Notes** |
| PK | id | serial4 | Y | Автоувеличение |
|  | name | text | Y |  |
|  |  |  |  |  |
| Таблица users | | | | |
| **Key** | **Field name** | **Data type / field size** | **REQUIRED?** | **Notes** |
| PK | id | serial4 | Y | Автоувеличение |
|  | password | text | Y |  |
|  | email | text | Y |  |
|  | birthday | date | Y |  |
|  | phone | text | Y |  |
|  | photo | text | Y |  |
| FK | id\_gender | int4 | Y |  |
| FK | id\_country | int4 | Y |  |
| FK | id\_role | int4 | Y |  |
|  | full\_name | text | Y |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблица moderators\_directions\_event | | | | |
| **Key** | **Field name** | **Data type / field size** | **REQUIRED?** | **Notes** |
| PK | id | serial4 | Y | Автоувеличение |
| FK | id\_user\_moderator | int4 | Y |  |
| FK | id\_direction | int4 | Y |  |
| FK | id\_type\_event | int4 | Y |  |
|  |  |  |  |  |
| Таблица jury\_directions | | | | |
| **Key** | **Field name** | **Data type / field size** | **REQUIRED?** | **Notes** |
| PK | id | serial4 | Y | Автоувеличение |
| FK | id\_user\_jury | int4 | Y |  |
| FK | id\_direction | int4 | Y |  |
|  |  |  |  |  |
| Таблица directions | | | | |
| **Key** | **Field name** | **Data type / field size** | **REQUIRED?** | **Notes** |
| PK | id | serial4 | Y | Автоувеличение |
|  | name | text | Y |  |

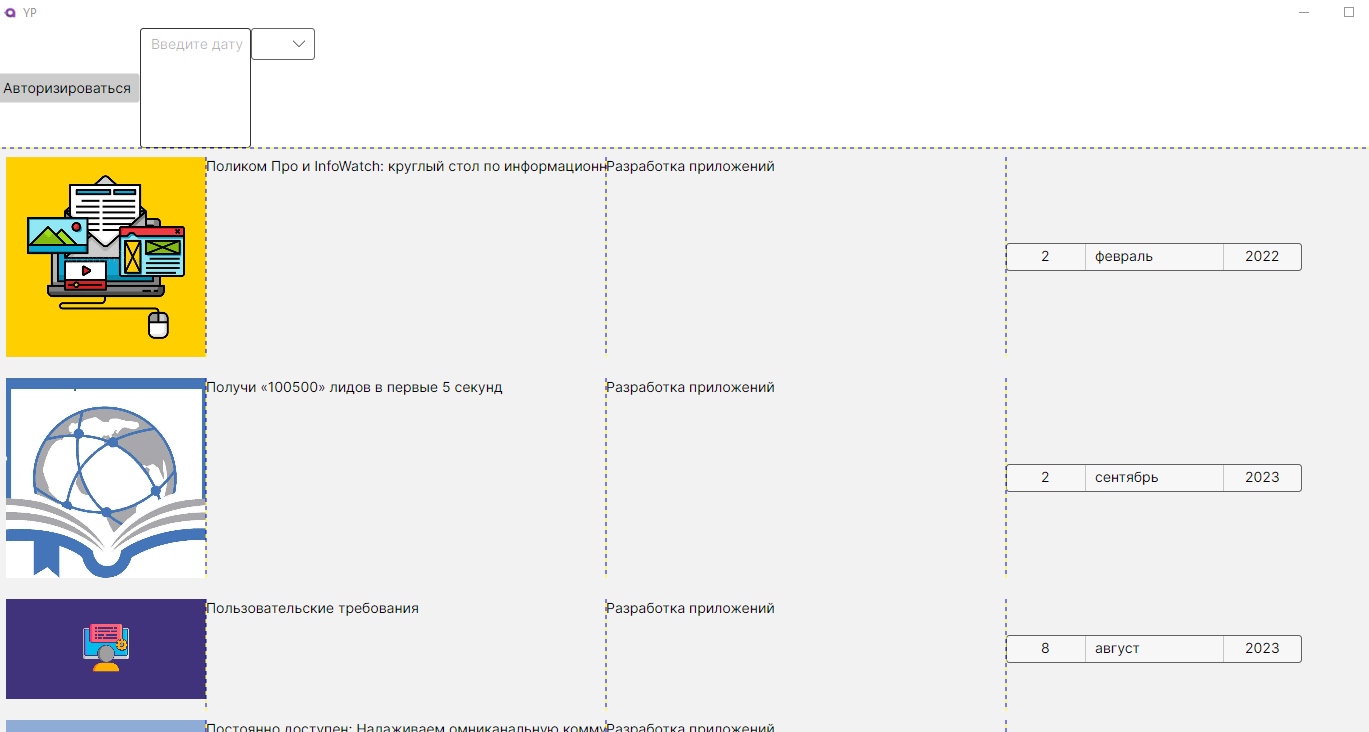
# **Руководство пользователя**

**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ ПРОГРАММЫ «YP»**

1. **Список мероприятий**

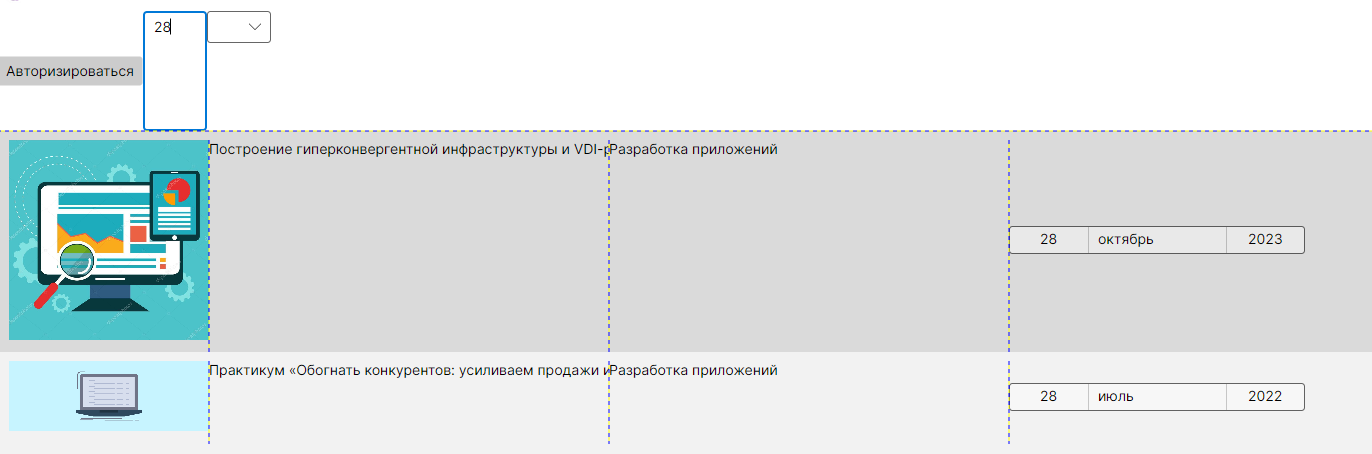
Программа после запуска открывает список мероприятий, показывая такие данные как:

* Изображение мероприятия
* Название мероприятия
* Тип мероприятия
* Дату мероприятия

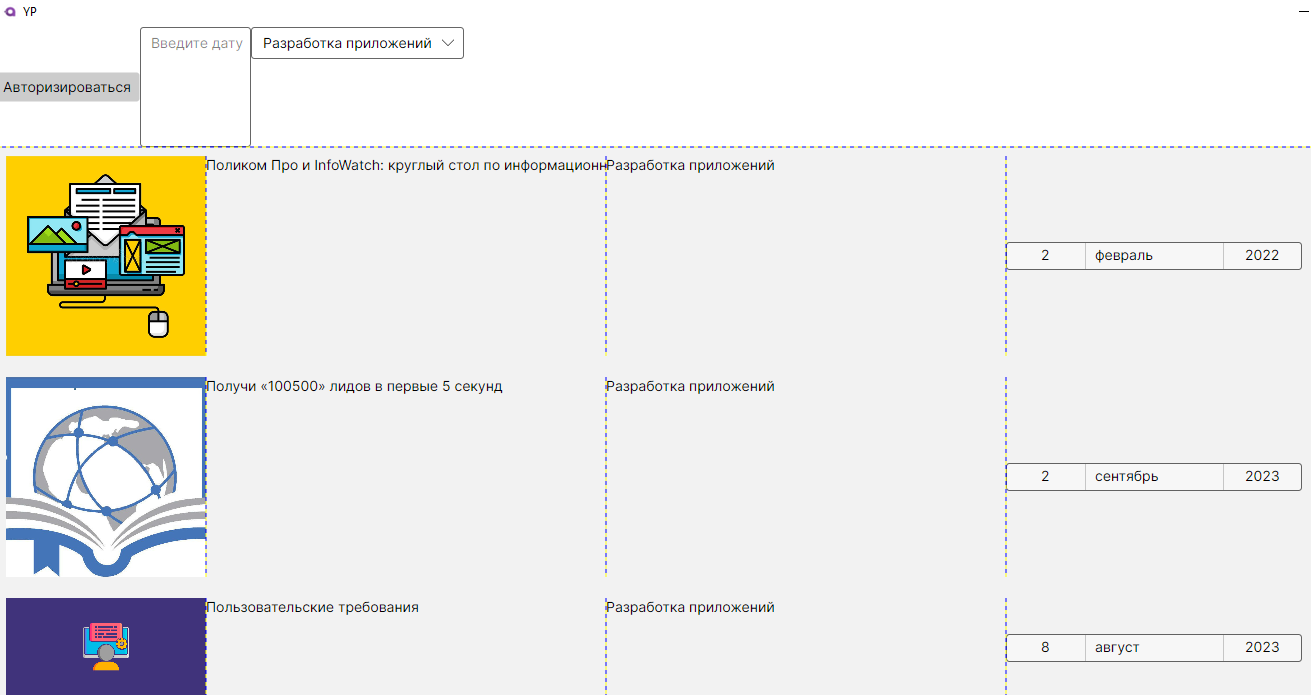


«Рисунок 1. Список мероприятий»

На странице доступен функционал поиска мероприятий по дате и фильтрация по типу мероприятия



«Рисунок 2. Поиск по дате»



«Рисунок 3. Фильтрация по типу»

Оба типа фильтрации работают параллельно.

Если Вы хотите войти в свою учетную запись, то нажмите на кнопку «Авторизироваться»

1. **Авторизация**

Окно авторизации содержит в себе следующие поля:

* Айди пользователя
* Пароль
* Капче
* Кнопка «Войти»

В поле «Айди пользователя» вводится уникальный идентификатор пользователя

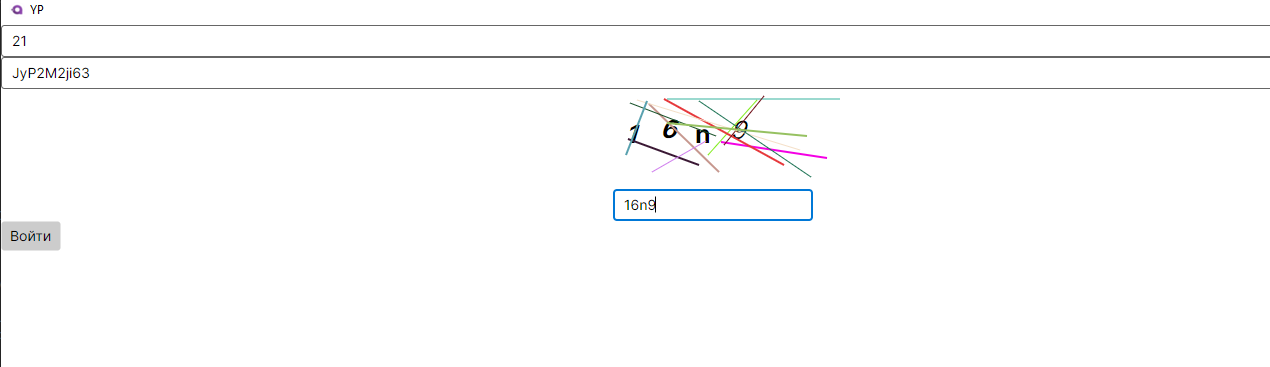
В поле «Пароль» вводится пароль от учетной записи пользователя

В поле под капчей необходимо ввести комбинацию символов с изображения.

Важно знать, что:

* После 3 попыток неудачного входа система заблокируется
* «Айди пользователя» должно быть только числом

После ввода корректных данных – будет произведен вход в систему (После первого входа система запомнит данные пользователя и сразу будет производить вход в систему)



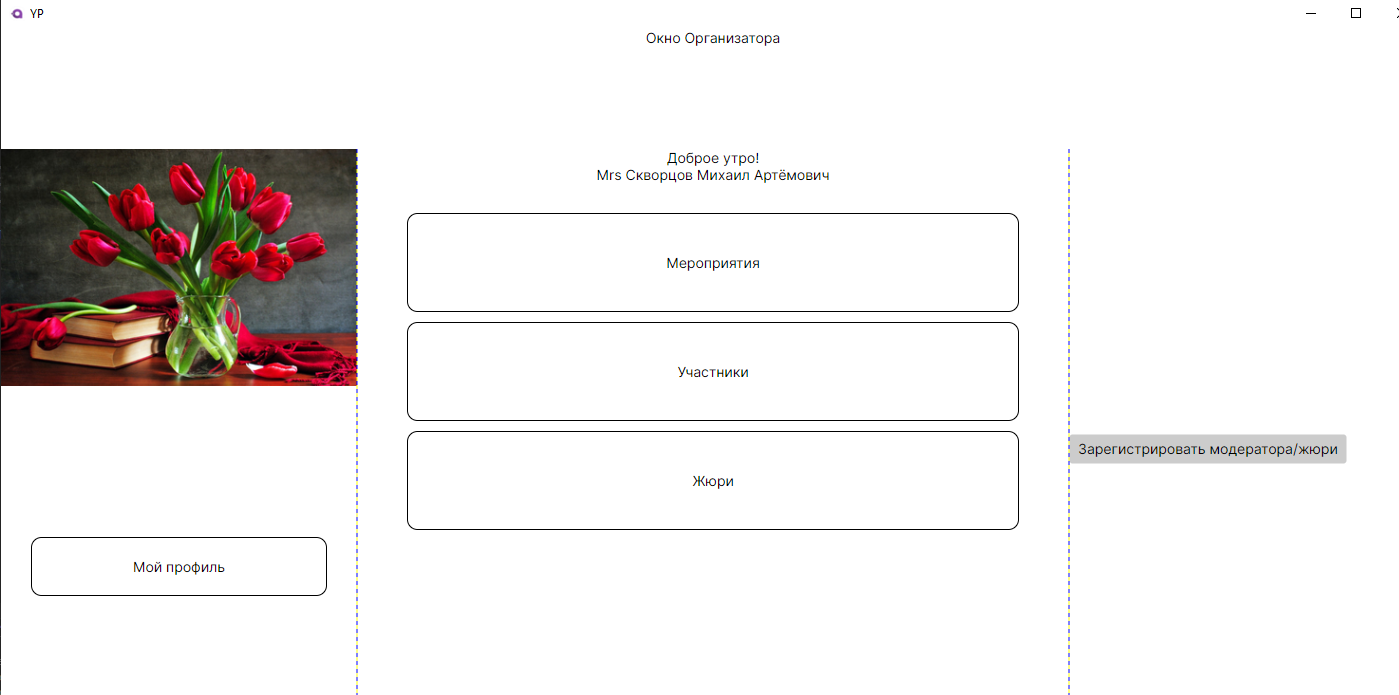
«Рисунок 4. Авторизация»

1. **Страница организатора**

На странице присутствует следующие элементы:

* Изображение пользователя
* Приветствие пользователя по ФИО
* Кнопка «Зарегистрировать модератора/жюри»
* Панели для работы организатора

Организатор может перейти к созданию нового жюри/модератора, нажав на кнопку «Зарегистрировать модератора/жюри».



«Рисунок 5. Окно организатора»

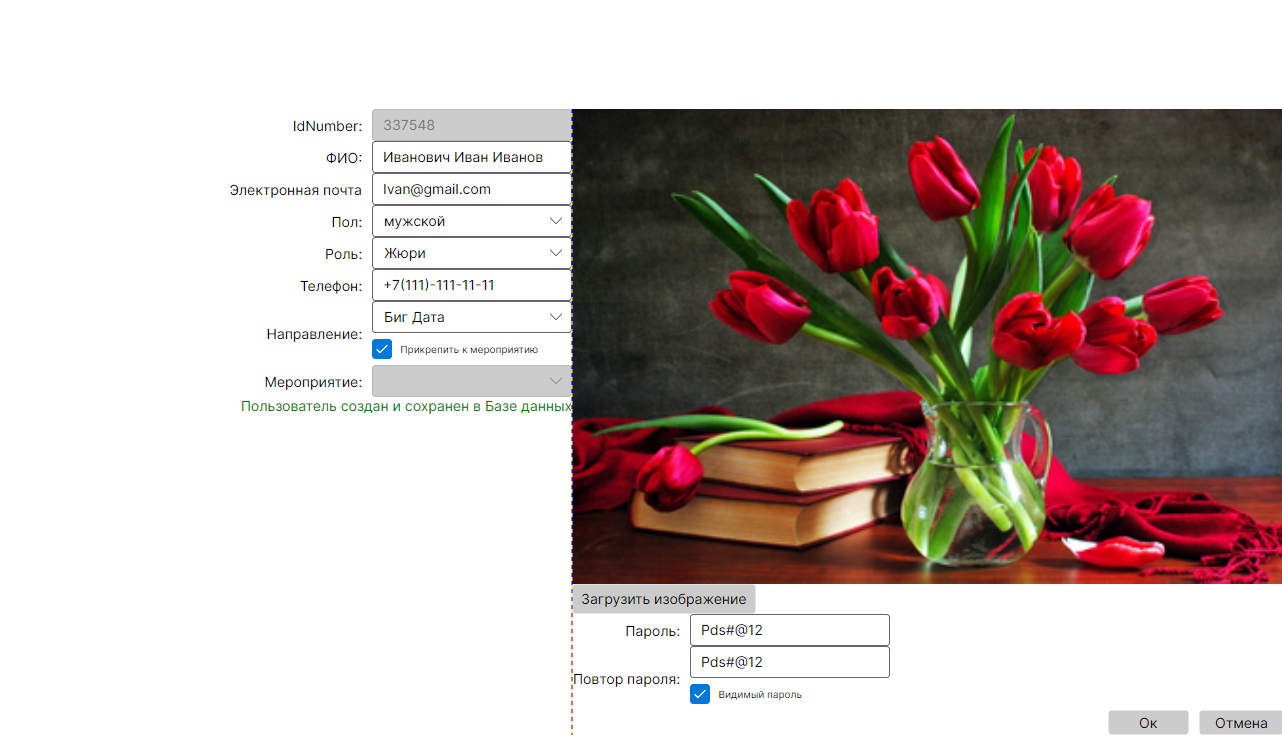
1. **Страница создания жюри/модератора**

На странице присутствует следующие элементы:

* Поле «IdNumber» (не изменяется, программа генерирует автоматически)
* Поле «ФИО»
* Поле «Электронная почта»
* Поле «Пол»
* Поле «Роль»
* Поле «Телефон»
* Поле «Направление»
* Поле «Мероприятие»
* Кнопка «Загрузить изображение»
* Поле «Пароль»
* Поле «Повтор пароля»
* Кнопка «Ок» (сохранить нового пользователя)
* Кнопка «Отмена» (вернуться на экран пользователя)

Важно отметить, что:

* При выборе роли «Жюри» можно выбрать только «Направление»
* Пароли должны совпадать и содержать следующие параметры (Пароль не содержит не менее 6 символов; заглавные и строчные буквы; не менее одного спецсимвола; не менее одной цифры.)
* Электронная почта должна соответствовать маске [test@gmail.com](mailto:test@gmail.com)
* Пароли можно скрыть или отображать



«Рисунок 6. Окно регистрации жюри/модератора»

# **Ручное тестирование**

**Аннотация теста**

|  |  |
| --- | --- |
| **Название проекта** | YP |
| **Рабочая версия** | 1.0.0 |
| **Имя тестирующего** | Куклев Георгий Александрович |
| **Дата(ы) теста** | 06.03.2025 |

**Расшифровка тестовых информационных полей:**

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| **Название проекта** | Название тестируемого проекта |
| **Рабочая версия** | Версия проекта/программного обеспечения (первый тест считается 1.0). |
| **Имя тестирующего** | Имя того, кто проводил тесты |
| **Дата(ы) теста** | Дата(ы) проведения тестов – это один или несколько дней. Если тесты проводились в более протяженный период времени, нужно отметить отдельную дату для каждого теста. |
| **Тестовый пример #** | Уникальный ID для каждого тестового примера. Следуйте некоторым конвенциям, чтобы указать типы тестов. Например,‘TC\_UI\_1′ означает‘user interface test case #1′ ( ТС\_ПИ\_1: тестовый случай пользовательского интерфейса#1) |
| **Приоритет тестирования**  *(Низкий/Средний/Высокий)* | Насколько важен каждый тест. Приоритет тестирования для бизнес-правил и функциональных тестовых случаев может быть средним или высоким, в то время как незначительные случаи пользовательского интерфейса могут иметь низкий приоритет. |
| **Заголовок/название теста** | Название тестового случая. Например, Подтвердите страницу авторизации с действительным именем пользователя и паролем. |
| **Краткое изложение теста** | Описание того, что должен достичь тест. |
| **Этапы теста** | Перечислите все этапы теста подробно. Запишите этапы теста в том порядке, в котором они должны быть реализованы. Предоставьте как можно больше подробностей и разъяснений. Пронумерованный список – хорошая идея. |
| **Тестовые данные** | Перечислите/опишите все тестовые данные, используемые для данного тестового случая. Так, фактические используемые входные данные можно отслеживать по результатам тестирования. Например, Имя пользователя и пароль для подтверждения входа. |
| **Ожидаемый результат** | Каким должен быть вывод системы после выполнения теста? Подробно опишите ожидаемый результат, включая все сообщения/ошибки, которые должны отображаться на экране. |
| **Фактический результат** | Каким должен быть фактический результат после выполнения теста? Опишите любое релевантное поведение системы после выполнения теста. |
| **Предварительное условие** | Любые предварительные условия, которые должны быть выполнены до выполнения теста. Перечислите все предварительные условия для выполнения этого тестового случая. |
| **Постусловие** | Каким должно быть состояние системы после выполнения теста? |
| **Статус**  *(Зачет/Незачет)* | Если фактический результат не соответствует ожидаемому результату, отметьте тест как неудачный. В ином случае обновление пройдено. |
| **Примечания/комментарии** | Используйте эту область для любых дополнительных заметок/комментариев/вопросов. Эта область предназначена для поддержки вышеуказанных полей (например, если есть некоторые особые условия, которые не могут быть описаны в любом из вышеуказанных полей, или если есть вопросы, связанные с ожидаемыми или фактическими результатами). |

Тестовый пример #1:

|  |  |
| --- | --- |
| **Тестовый пример #** | ТК 1 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Проверка создания нового жюри |
| **Краткое изложение теста** | Проверка создания нового пользователя с ролью «Жюри» |
| **Этапы теста** | 1. Нажать кнопку «Авторизоваться» 2. Ввести капчу, логин, пароль - JyP2M2ji63, айди пользователя - 21 3. Нажать кнопку «Зарегестрировать модератора/жюри» 4. Ввести все данные нового жюри (см. тестовые данные) 5. Нажать кнопку «Ок» |
| **Тестовые данные** | 1. Фио – Иванов Иван Иванович 2. Электронная почта - [test@gmail.com](mailto:test@gmail.com) 3. Пол – мужской 4. Роль – Жюри 5. Телефон - +7 (111)1111111 6. Направление – ИТ 7. Изображение – foto1.jpg 8. Пароль – Pasw@#21 9. Повтор пароля - Pasw@#21 |
| **Ожидаемый результат** | На экране появится сообщение «Пользователь создан и сохранен в Базе данных». В БД появится новый пользователь с роль. «Жюри» |
| **Фактический результат** | На экране появилось сообщение «Пользователь создан и сохранен в Базе данных». В БД появился новый пользователь с роль. «Жюри» |
| **Статус** | Зачет |
| **Предварительное условие** | Программа запущена |
| **Постусловие** | Программа продолжила работу |
| **Примечания/комментарии** | - |

Тестовый пример #2:

|  |  |
| --- | --- |
| **Тестовый пример #** | ТК 2 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Проверка создания модератора |
| **Краткое изложение теста** | Проверка создания нового пользователя с ролью «Модератор» |
| **Этапы теста** | 1. Нажать кнопку «Авторизоваться» 2. Ввести капчу, логин, пароль - JyP2M2ji63, айди пользователя - 21 3. Нажать кнопку «Зарегестрировать модератора/жюри» 4. Ввести все данные нового жюри (см. тестовые данные) 5. Нажать кнопку «Ок» |
| **Тестовые данные** | 1. Фио – Иванов Иван Иванович 2. Электронная почта - [test@gmail.com](mailto:test@gmail.com) 3. Пол – мужской 4. Роль – Жюри 5. Телефон - +7 (111)1111111 6. Направление – ИТ 7. Мероприятие – Пользовательские требования 8. Изображение – foto1.jpg 9. Пароль – Pasw@#21 10. Повтор пароля - Pasw@#21 |
| **Ожидаемый результат** | На экране появится сообщение «Пользователь создан и сохранен в Базе данных». В БД появится новый пользователь с роль. «Модератор» |
| **Фактический результат** | На экране появилось сообщение «Пользователь создан и сохранен в Базе данных». В БД появился новый пользователь с роль. «Модератор» |
| **Статус** | Зачет |
| **Предварительное условие** | Программа запущена |
| **Постусловие** | Программа продолжила работу |
| **Примечания/комментарии** | - |

Тестовый пример #3:

|  |  |
| --- | --- |
| Тестовый пример **#** | ТК 3 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Проверка почты |
| **Краткое изложение теста** | Проверка отмены создания нового пользователя, если почта не соответствует маске ввода |
| **Этапы теста** | 1. Нажать кнопку «Авторизоваться» 2. Ввести капчу, логин, пароль - JyP2M2ji63, айди пользователя - 21 3. Нажать кнопку «Зарегестрировать модератора/жюри» 4. Ввести все данные нового жюри (см. тестовые данные) 5. Нажать кнопку «Ок» |
| **Тестовые данные** | 1. Фио – Иванов Иван Иванович 2. Электронная почта - tttttt 3. Пол – мужской 4. Роль – Жюри 5. Телефон - +7 (111)1111111 6. Направление – ИТ 7. Мероприятие – Пользовательские требования 8. Изображение – foto1.jpg 9. Пароль – Pasw@#21 10. Повтор пароля - Pasw@#21 |
| **Ожидаемый результат** | Программа выведет сообщение «Электронная почта неправильная» и не создаст нового пользователя |
| **Фактический результат** | Программа вывела сообщение «Электронная почта неправильная» и не создала нового пользователя |
| **Статус** | Зачет |
| **Предварительное условие** | Программа запущена |
| **Постусловие** | Программа продолжила работу |
| **Примечания/комментарии** | - |

Тестовый пример #4:

|  |  |
| --- | --- |
| Тестовый пример **#** | ТК 4 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Проверка одинакового ввода паролей |
| **Краткое изложение теста** | Проверка на ввод одинакового «Пароль» и «Повтор пароля» |
| **Этапы теста** | 1. Нажать кнопку «Авторизоваться» 2. Ввести капчу, логин, пароль - JyP2M2ji63, айди пользователя - 21 3. Нажать кнопку «Зарегестрировать модератора/жюри» 4. Ввести все данные нового жюри (см. тестовые данные) 5. Нажать кнопку «Ок» |
| **Тестовые данные** | 1. Фио – Иванов Иван Иванович 2. Электронная почта - tttttt 3. Пол – мужской 4. Роль – Жюри 5. Телефон - +7 (111)1111111 6. Направление – ИТ 7. Мероприятие – Пользовательские требования 8. Изображение – foto1.jpg 9. Пароль – PaSW@#21 10. Повтор пароля - Pasw@#21 |
| **Ожидаемый результат** | Программа выведет сообщение «Пароли не совпадают» и не создаст нового пользователя |
| **Фактический результат** | Программа вывела сообщение «Пароли не совпадают» и не создала нового пользователя |
| **Статус** | Зачет |
| **Предварительное условие** | Программа запущена |
| **Постусловие** | Программа продолжила работу |
| **Примечания/комментарии** | - |

Test case #5:

|  |  |
| --- | --- |
| **Test Case #** | ТК 5 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Проверка требований к паролю |
| **Краткое изложение теста** | Проверка соответствие паролей к требованиям (Пароль не содержит не менее 6 символов;  • заглавные и строчные буквы;  • не менее одного спецсимвола;  • не менее одной цифры.) |
| **Этапы теста** | 1. Нажать кнопку «Авторизоваться» 2. Ввести капчу, логин, пароль - JyP2M2ji63, айди пользователя - 21 3. Нажать кнопку «Зарегестрировать модератора/жюри» 4. Ввести все данные нового жюри (см. тестовые данные) 5. Нажать кнопку «Ок» |
| **Тестовые данные** | 1. Фио – Иванов Иван Иванович 2. Электронная почта - tttttt 3. Пол – мужской 4. Роль – Жюри 5. Телефон - +7 (111)1111111 6. Направление – ИТ 7. Мероприятие – Пользовательские требования 8. Изображение – foto1.jpg 9. Пароль – 11 10. Повтор пароля - 11 |
| **Ожидаемый результат** | Программа выведет сообщение «Пароль не содержит не менее 6 символов;  • заглавные и строчные буквы;  • не менее одного спецсимвола;  • не менее одной цифры.» и не создаст нового пользователя |
| **Фактический результат** | Программа вывела сообщение «Пароль не содержит не менее 6 символов;  • заглавные и строчные буквы;  • не менее одного спецсимвола;  • не менее одной цифры.» и не создала нового пользователя |
| **Статус** | Зачет |
| **Предварительное условие** | Программа запущена |
| **Постусловие** | Программа продолжила работу |
| **Примечания/комментарии** | - |

# **ПРИЛОЖЕНИЕ Б**

# Листинг 1 - SQL-скрипт БД

|  |
| --- |
| -- public.activeties definition  -- Drop table  -- DROP TABLE public.activeties;  **CREATE** **TABLE** public.activeties (  id serial4 **NOT** **NULL**,  "name" **text** **NOT** **NULL**,  **CONSTRAINT** activeties\_name\_key **UNIQUE** (**name**),  **CONSTRAINT** activeties\_pkey **PRIMARY** **KEY** (id)  );  -- public.cities definition  -- Drop table  -- DROP TABLE public.cities;  **CREATE** **TABLE** public.cities (  id serial4 **NOT** **NULL**,  "name" **text** **NOT** **NULL**,  **CONSTRAINT** cities\_name\_key **UNIQUE** (**name**),  **CONSTRAINT** cities\_pkey **PRIMARY** **KEY** (id)  );  -- public.countries definition  -- Drop table  -- DROP TABLE public.countries;  **CREATE** **TABLE** public.countries (  id serial4 **NOT** **NULL**,  "name" **text** **NOT** **NULL**,  en\_name **text** **NOT** **NULL**,  code **text** **NOT** **NULL**,  code\_second **int4** **NOT** **NULL**,  **CONSTRAINT** countries\_code\_key **UNIQUE** (code),  **CONSTRAINT** countries\_en\_name\_key **UNIQUE** (en\_name),  **CONSTRAINT** countries\_name\_key **UNIQUE** (**name**),  **CONSTRAINT** countries\_pkey **PRIMARY** **KEY** (id)  );  -- public.directions definition  -- Drop table  -- DROP TABLE public.directions;  **CREATE** **TABLE** public.directions (  id serial4 **NOT** **NULL**,  "name" **text** **NOT** **NULL**,  **CONSTRAINT** directions\_name\_key **UNIQUE** (**name**),  **CONSTRAINT** directions\_pkey **PRIMARY** **KEY** (id)  );  -- public.genders definition  -- Drop table  -- DROP TABLE public.genders;  **CREATE** **TABLE** public.genders (  id serial4 **NOT** **NULL**,  "name" **text** **NOT** **NULL**,  **CONSTRAINT** genders\_name\_key **UNIQUE** (**name**),  **CONSTRAINT** genders\_pkey **PRIMARY** **KEY** (id)  );  -- public.roles definition  -- Drop table  -- DROP TABLE public.roles;  **CREATE** **TABLE** public.roles (  id serial4 **NOT** **NULL**,  "name" **text** **NOT** **NULL**,  **CONSTRAINT** roles\_name\_key **UNIQUE** (**name**),  **CONSTRAINT** roles\_pkey **PRIMARY** **KEY** (id)  );  -- public.types\_events definition  -- Drop table  -- DROP TABLE public.types\_events;  **CREATE** **TABLE** public.types\_events (  id serial4 **NOT** **NULL**,  "name" **text** **NOT** **NULL**,  **CONSTRAINT** types\_events\_pk **PRIMARY** **KEY** (id)  );  -- public.events definition  -- Drop table  -- DROP TABLE public.events;  **CREATE** **TABLE** public.events (  id serial4 **NOT** **NULL**,  "name" **text** **NOT** **NULL**,  date\_start **date** **NOT** **NULL**,  days **int4** **NOT** **NULL**,  id\_city **int4** **NOT** **NULL**,  id\_type\_event **int4** **NULL**,  image **text** **NULL**,  **CONSTRAINT** events\_pkey **PRIMARY** **KEY** (id),  **CONSTRAINT** events\_types\_events\_fk **FOREIGN** **KEY** (id\_type\_event) **REFERENCES** public.types\_events(id),  **CONSTRAINT** fk\_event\_id **FOREIGN** **KEY** (id\_city) **REFERENCES** public.cities(id) **ON** **DELETE** **SET** **NULL** **ON** **UPDATE** **SET** **NULL**  );  -- public.images\_cities definition  -- Drop table  -- DROP TABLE public.images\_cities;  **CREATE** **TABLE** public.images\_cities (  id serial4 **NOT** **NULL**,  id\_city **int4** **NOT** **NULL**,  image **text** **NULL**,  **CONSTRAINT** images\_cities\_pk **PRIMARY** **KEY** (id),  **CONSTRAINT** images\_cities\_cities\_fk **FOREIGN** **KEY** (id\_city) **REFERENCES** public.cities(id)  );  -- public.users definition  -- Drop table  -- DROP TABLE public.users;  **CREATE** **TABLE** public.users (  id serial4 **NOT** **NULL**,  "password" **text** **NOT** **NULL**,  email **text** **NULL**,  birthday **date** **NULL**,  phone **text** **NOT** **NULL**,  photo **text** **NULL**,  id\_gender **int4** **NOT** **NULL**,  id\_country **int4** **NULL**,  id\_role **int4** **NOT** **NULL**,  full\_name **text** **NOT** **NULL**,  image **text** **NULL**,  **CONSTRAINT** users\_pkey **PRIMARY** **KEY** (id),  **CONSTRAINT** fk\_country\_id **FOREIGN** **KEY** (id\_country) **REFERENCES** public.countries(id) **ON** **DELETE** **SET** **NULL** **ON** **UPDATE** **SET** **NULL**,  **CONSTRAINT** fk\_gender\_id **FOREIGN** **KEY** (id\_gender) **REFERENCES** public.genders(id) **ON** **DELETE** **SET** **NULL** **ON** **UPDATE** **SET** **NULL**,  **CONSTRAINT** fk\_roles\_id **FOREIGN** **KEY** (id\_role) **REFERENCES** public.roles(id) **ON** **DELETE** **SET** **NULL** **ON** **UPDATE** **SET** **NULL**  );  -- public.jury\_directions definition  -- Drop table  -- DROP TABLE public.jury\_directions;  **CREATE** **TABLE** public.jury\_directions (  id serial4 **NOT** **NULL**,  id\_user\_jury **int4** **NOT** **NULL**,  id\_direction **int4** **NOT** **NULL**,  **CONSTRAINT** jury\_directions\_pkey **PRIMARY** **KEY** (id),  **CONSTRAINT** fk\_direction\_id **FOREIGN** **KEY** (id\_direction) **REFERENCES** public.directions(id) **ON** **DELETE** **SET** **NULL** **ON** **UPDATE** **SET** **NULL**,  **CONSTRAINT** fk\_user\_id **FOREIGN** **KEY** (id\_user\_jury) **REFERENCES** public.users(id) **ON** **DELETE** **SET** **NULL** **ON** **UPDATE** **SET** **NULL**  );  -- public.moderators\_directions\_events definition  -- Drop table  -- DROP TABLE public.moderators\_directions\_events;  **CREATE** **TABLE** public.moderators\_directions\_events (  id serial4 **NOT** **NULL**,  id\_user\_moderator **int4** **NOT** **NULL**,  id\_direction **int4** **NOT** **NULL**,  id\_type\_event **int4** **NOT** **NULL**,  **CONSTRAINT** moderators\_directions\_events\_pkey **PRIMARY** **KEY** (id),  **CONSTRAINT** fk\_direction\_id **FOREIGN** **KEY** (id\_direction) **REFERENCES** public.directions(id) **ON** **DELETE** **SET** **NULL** **ON** **UPDATE** **SET** **NULL**,  **CONSTRAINT** fk\_event\_id **FOREIGN** **KEY** (id\_type\_event) **REFERENCES** public.events(id) **ON** **DELETE** **SET** **NULL** **ON** **UPDATE** **SET** **NULL**,  **CONSTRAINT** fk\_user\_id **FOREIGN** **KEY** (id\_user\_moderator) **REFERENCES** public.users(id) **ON** **DELETE** **SET** **NULL** **ON** **UPDATE** **SET** **NULL**  );  -- public.schedule\_events definition  -- Drop table  -- DROP TABLE public.schedule\_events;  **CREATE** **TABLE** public.schedule\_events (  id serial4 **NOT** **NULL**,  id\_event **int4** **NOT** **NULL**,  "day" **int4** **NOT** **NULL**,  id\_winner\_user **int4** **NULL**,  **CONSTRAINT** events\_activities\_pkey **PRIMARY** **KEY** (id),  **CONSTRAINT** schedule\_events\_events\_fk **FOREIGN** **KEY** (id\_event) **REFERENCES** public.events(id),  **CONSTRAINT** schedule\_events\_users\_fk **FOREIGN** **KEY** (id\_winner\_user) **REFERENCES** public.users(id)  );  -- public.schedule\_activeties\_in\_events definition  -- Drop table  -- DROP TABLE public.schedule\_activeties\_in\_events;  **CREATE** **TABLE** public.schedule\_activeties\_in\_events (  id serial4 **NOT** **NULL**,  id\_activity **int4** **NOT** **NULL**,  time\_start **time** **NOT** **NULL**,  "day" **int4** **NOT** **NULL**,  id\_schedule\_event **int4** **NOT** **NULL**,  id\_moderator\_user **int4** **NULL**,  **CONSTRAINT** schedule\_activeties\_in\_events\_pk **PRIMARY** **KEY** (id),  **CONSTRAINT** schedule\_activeties\_in\_events\_activeties\_fk **FOREIGN** **KEY** (id\_activity) **REFERENCES** public.activeties(id),  **CONSTRAINT** schedule\_activeties\_in\_events\_schedule\_events\_fk **FOREIGN** **KEY** (id\_schedule\_event) **REFERENCES** public.schedule\_events(id),  **CONSTRAINT** schedule\_activeties\_in\_events\_users\_fk **FOREIGN** **KEY** (id\_moderator\_user) **REFERENCES** public.users(id)  );  -- public.schedule\_activeties\_jury definition  -- Drop table  -- DROP TABLE public.schedule\_activeties\_jury;  **CREATE** **TABLE** public.schedule\_activeties\_jury (  id serial4 **NOT** **NULL**,  id\_jury\_user **int4** **NOT** **NULL**,  id\_schedule\_activity **int4** **NOT** **NULL**,  **CONSTRAINT** schedule\_activeties\_moderators\_jury\_pk **PRIMARY** **KEY** (id),  **CONSTRAINT** schedule\_activeties\_jury\_schedule\_activeties\_in\_events\_fk **FOREIGN** **KEY** (id\_schedule\_activity) **REFERENCES** public.schedule\_activeties\_in\_events(id),  **CONSTRAINT** schedule\_activeties\_jury\_users\_fk\_1 **FOREIGN** **KEY** (id\_jury\_user) **REFERENCES** public.users(id)  );  **INSERT** **INTO** public.activeties ("name") **VALUES**  ('ТОП 50 '),  ('Перспективные направления IT'),  ('Современные технологии'),  ('Что такое IOT?'),  ('Для чего это нужно?'),  ('Новые продукты и версии'),  ('Как составить план?'),  ('Что такое план?'),  ('Must Have 21 века'),  ('Управление знаниями');  **INSERT** **INTO** public.activeties ("name") **VALUES**  ('Коммуникативные неудачи'),  ('Дизайн-мышление'),  ('Технические собеседования'),  ('Исследование рынка'),  ('Подготовка специалистов'),  ('Секреты успеха'),  ('Способы поиска специалистов'),  ('Что нужно пользователям?'),  ('Осознанность личных целей'),  ('Soft skills');  **INSERT** **INTO** public.activeties ("name") **VALUES**  ('Развитие проактивности'),  ('Введение в IT-субкультуру'),  ('Основы ООП'),  ('Что нужно знать начинающему спецалисту'),  ('Идельный результат'),  ('Тайны и секреты'),  ('Какрой язык программирования выбрать?'),  ('Войти в ТОП'),  ('Секреты продвижения'),  ('Руководство проектами');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Абаза'),  ('Абакан'),  ('Абдулино'),  ('Абинск'),  ('Агидель'),  ('Агрыз'),  ('Адыгейск'),  ('Азнакаево'),  ('Азов'),  ('Ак-Довурак');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Аксай'),  ('Алагир'),  ('Алапаевск'),  ('Алатырь'),  ('Алдан'),  ('Алейск'),  ('Александров'),  ('Александровск'),  ('Александровск-Сахалинский'),  ('Алексеевка');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Алексин'),  ('Алзамай'),  ('Алупкане призн.'),  ('Алуштане призн.'),  ('Альметьевск'),  ('Амурск'),  ('Анадырь'),  ('Анапа'),  ('Ангарск'),  ('Андреаполь');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Анжеро-Судженск'),  ('Анива'),  ('Апатиты'),  ('Апрелевка'),  ('Апшеронск'),  ('Арамиль'),  ('Аргун'),  ('Ардатов'),  ('Ардон'),  ('Арзамас');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Аркадак'),  ('Армавир'),  ('Армянскне призн.'),  ('Арсеньев'),  ('Арск'),  ('Артём'),  ('Артёмовск'),  ('Артёмовский'),  ('Архангельск'),  ('Асбест');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Асино'),  ('Астрахань'),  ('Аткарск'),  ('Ахтубинск'),  ('Ачинск'),  ('Аша'),  ('Бабаево'),  ('Бабушкин'),  ('Бавлы'),  ('Багратионовск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Байкальск'),  ('Баймак'),  ('Бакал'),  ('Баксан'),  ('Балабаново'),  ('Балаково'),  ('Балахна'),  ('Балашиха'),  ('Балашов'),  ('Балей');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Балтийск'),  ('Барабинск'),  ('Барнаул'),  ('Барыш'),  ('Батайск'),  ('Бахчисарайне призн.'),  ('Бежецк'),  ('Белая Калитва'),  ('Белая Холуница'),  ('Белгород');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Белебей'),  ('Белёв'),  ('Белинский'),  ('Белово'),  ('Белогорск'),  ('Белогорскне призн.'),  ('Белозерск'),  ('Белокуриха'),  ('Беломорск'),  ('Белоозёрский');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Белорецк'),  ('Белореченск'),  ('Белоусово'),  ('Белоярский'),  ('Белый'),  ('Бердск'),  ('Березники'),  ('Берёзовский'),  ('Беслан'),  ('Бийск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Бикин'),  ('Билибино'),  ('Биробиджан'),  ('Бирск'),  ('Бирюсинск'),  ('Бирюч'),  ('Благовещенск'),  ('Благодарный'),  ('Бобров'),  ('Богданович');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Богородицк'),  ('Богородск'),  ('Боготол'),  ('Богучар'),  ('Бодайбо'),  ('Бокситогорск'),  ('Болгар'),  ('Бологое'),  ('Болотное'),  ('Болохово');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Болхов'),  ('Большой Камень'),  ('Бор'),  ('Борзя'),  ('Борисоглебск'),  ('Боровичи'),  ('Боровск'),  ('Бородино'),  ('Братск'),  ('Бронницы');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Брянск'),  ('Бугульма'),  ('Бугуруслан'),  ('Будённовск'),  ('Бузулук'),  ('Буинск'),  ('Буй'),  ('Буйнакск'),  ('Бутурлиновка'),  ('Валдай');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Валуйки'),  ('Велиж'),  ('Великие Луки'),  ('Великий Новгород'),  ('Великий Устюг'),  ('Вельск'),  ('Венёв'),  ('Верещагино'),  ('Верея'),  ('Верхнеуральск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Верхний Тагил'),  ('Верхний Уфалей'),  ('Верхняя Пышма'),  ('Верхняя Салда'),  ('Верхняя Тура'),  ('Верхотурье'),  ('Верхоянск'),  ('Весьегонск'),  ('Ветлуга'),  ('Видное');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Вилюйск'),  ('Вилючинск'),  ('Вихоревка'),  ('Вичуга'),  ('Владивосток'),  ('Владикавказ'),  ('Владимир'),  ('Волгоград'),  ('Волгодонск'),  ('Волгореченск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Волжск'),  ('Волжский'),  ('Вологда'),  ('Володарск'),  ('Волоколамск'),  ('Волосово'),  ('Волхов'),  ('Волчанск'),  ('Вольск'),  ('Воркута');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Воронеж'),  ('Ворсма'),  ('Воскресенск'),  ('Воткинск'),  ('Всеволожск'),  ('Вуктыл'),  ('Выборг'),  ('Выкса'),  ('Высоковск'),  ('Высоцк');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Вытегра'),  ('Вышний Волочёк'),  ('Вяземский'),  ('Вязники'),  ('Вязьма'),  ('Вятские Поляны'),  ('Гаврилов Посад'),  ('Гаврилов-Ям'),  ('Гагарин'),  ('Гаджиево');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Гай'),  ('Галич'),  ('Гатчина'),  ('Гвардейск'),  ('Гдов'),  ('Геленджик'),  ('Георгиевск'),  ('Глазов'),  ('Голицыно'),  ('Горбатов');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Горно-Алтайск'),  ('Горнозаводск'),  ('Горняк'),  ('Городец'),  ('Городище'),  ('Городовиковск'),  ('Гороховец'),  ('Горячий Ключ'),  ('Грайворон'),  ('Гремячинск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Грозный'),  ('Грязи'),  ('Грязовец'),  ('Губаха'),  ('Губкин'),  ('Губкинский'),  ('Гудермес'),  ('Гуково'),  ('Гулькевичи'),  ('Гурьевск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Гусев'),  ('Гусиноозёрск'),  ('Гусь-Хрустальный'),  ('Давлеканово'),  ('Дагестанские Огни'),  ('Далматово'),  ('Дальнегорск'),  ('Дальнереченск'),  ('Данилов'),  ('Данков');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Дегтярск'),  ('Дедовск'),  ('Демидов'),  ('Дербент'),  ('Десногорск'),  ('Джанкойне призн.'),  ('Дзержинск'),  ('Дзержинский'),  ('Дивногорск'),  ('Дигора');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Димитровград'),  ('Дмитриев'),  ('Дмитров'),  ('Дмитровск'),  ('Дно'),  ('Добрянка'),  ('Долгопрудный'),  ('Долинск'),  ('Домодедово'),  ('Донецк');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Донской'),  ('Дорогобуж'),  ('Дрезна'),  ('Дубна'),  ('Дубовка'),  ('Дудинка'),  ('Духовщина'),  ('Дюртюли'),  ('Дятьково'),  ('Евпаторияне призн.');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Егорьевск'),  ('Ейск'),  ('Екатеринбург'),  ('Елабуга'),  ('Елец'),  ('Елизово'),  ('Ельня'),  ('Еманжелинск'),  ('Емва'),  ('Енисейск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Ермолино'),  ('Ершов'),  ('Ессентуки'),  ('Ефремов'),  ('Железноводск'),  ('Железногорск'),  ('Железногорск-Илимский'),  ('Жердевка'),  ('Жигулёвск'),  ('Жиздра');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Жирновск'),  ('Жуков'),  ('Жуковка'),  ('Жуковский'),  ('Завитинск'),  ('Заводоуковск'),  ('Заволжск'),  ('Заволжье'),  ('Задонск'),  ('Заинск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Закаменск'),  ('Заозёрный'),  ('Заозёрск'),  ('Западная Двина'),  ('Заполярный'),  ('Зарайск'),  ('Заречный'),  ('Заринск'),  ('Звенигово'),  ('Звенигород');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Зверево'),  ('Зеленогорск'),  ('Зеленоградск'),  ('Зеленодольск'),  ('Зеленокумск'),  ('Зерноград'),  ('Зея'),  ('Зима'),  ('Златоуст'),  ('Злынка');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Змеиногорск'),  ('Знаменск'),  ('Зубцов'),  ('Зуевка'),  ('Ивангород'),  ('Иваново'),  ('Ивантеевка'),  ('Ивдель'),  ('Игарка'),  ('Ижевск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Избербаш'),  ('Изобильный'),  ('Иланский'),  ('Инза'),  ('Иннополис'),  ('Инсар'),  ('Инта'),  ('Ипатово'),  ('Ирбит'),  ('Иркутск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Исилькуль'),  ('Искитим'),  ('Истра'),  ('Ишим'),  ('Ишимбай'),  ('Йошкар-Ола'),  ('Кадников'),  ('Казань'),  ('Калач'),  ('Калач-на-Дону');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Калачинск'),  ('Калининград'),  ('Калининск'),  ('Калтан'),  ('Калуга'),  ('Калязин'),  ('Камбарка'),  ('Каменка'),  ('Каменногорск'),  ('Каменск-Уральский');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Каменск-Шахтинский'),  ('Камень-на-Оби'),  ('Камешково'),  ('Камызяк'),  ('Камышин'),  ('Камышлов'),  ('Канаш'),  ('Кандалакша'),  ('Канск'),  ('Карабаново');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Карабаш'),  ('Карабулак'),  ('Карасук'),  ('Карачаевск'),  ('Карачев'),  ('Каргат'),  ('Каргополь'),  ('Карпинск'),  ('Карталы'),  ('Касимов');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Касли'),  ('Каспийск'),  ('Катав-Ивановск'),  ('Катайск'),  ('Качканар'),  ('Кашин'),  ('Кашира'),  ('Кедровый'),  ('Кемерово'),  ('Кемь');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Керчьне призн.'),  ('Кизел'),  ('Кизилюрт'),  ('Кизляр'),  ('Кимовск'),  ('Кимры'),  ('Кингисепп'),  ('Кинель'),  ('Кинешма'),  ('Киреевск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Киренск'),  ('Киржач'),  ('Кириллов'),  ('Кириши'),  ('Киров'),  ('Кировград'),  ('Кирово-Чепецк'),  ('Кировск'),  ('Кирс'),  ('Кирсанов');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Киселёвск'),  ('Кисловодск'),  ('Клин'),  ('Клинцы'),  ('Княгинино'),  ('Ковдор'),  ('Ковров'),  ('Ковылкино'),  ('Когалым'),  ('Кодинск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Козельск'),  ('Козловка'),  ('Козьмодемьянск'),  ('Кола'),  ('Кологрив'),  ('Коломна'),  ('Колпашево'),  ('Кольчугино'),  ('Коммунар'),  ('Комсомольск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Комсомольск-на-Амуре'),  ('Конаково'),  ('Кондопога'),  ('Кондрово'),  ('Константиновск'),  ('Копейск'),  ('Кораблино'),  ('Кореновск'),  ('Коркино'),  ('Королёв');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Короча'),  ('Корсаков'),  ('Коряжма'),  ('Костерёво'),  ('Костомукша'),  ('Кострома'),  ('Котельники'),  ('Котельниково'),  ('Котельнич'),  ('Котлас');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Котово'),  ('Котовск'),  ('Кохма'),  ('Красавино'),  ('Красноармейск'),  ('Красновишерск'),  ('Красногорск'),  ('Краснодар'),  ('Краснозаводск'),  ('Краснознаменск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Краснокаменск'),  ('Краснокамск'),  ('Красноперекопскне призн.'),  ('Краснослободск'),  ('Краснотурьинск'),  ('Красноуральск'),  ('Красноуфимск'),  ('Красноярск'),  ('Красный Кут'),  ('Красный Сулин');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Красный Холм'),  ('Кремёнки'),  ('Кропоткин'),  ('Крымск'),  ('Кстово'),  ('Кубинка'),  ('Кувандык'),  ('Кувшиново'),  ('Кудрово'),  ('Кудымкар');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Кузнецк'),  ('Куйбышев'),  ('Кукмор'),  ('Кулебаки'),  ('Кумертау'),  ('Кунгур'),  ('Купино'),  ('Курган'),  ('Курганинск'),  ('Курильск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Курлово'),  ('Куровское'),  ('Курск'),  ('Куртамыш'),  ('Курчалой'),  ('Курчатов'),  ('Куса'),  ('Кушва'),  ('Кызыл'),  ('Кыштым');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Кяхта'),  ('Лабинск'),  ('Лабытнанги'),  ('Лагань'),  ('Ладушкин'),  ('Лаишево'),  ('Лакинск'),  ('Лангепас'),  ('Лахденпохья'),  ('Лебедянь');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Лениногорск'),  ('Ленинск'),  ('Ленинск-Кузнецкий'),  ('Ленск'),  ('Лермонтов'),  ('Лесной'),  ('Лесозаводск'),  ('Лесосибирск'),  ('Ливны'),  ('Ликино-Дулёво');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Липецк'),  ('Липки'),  ('Лиски'),  ('Лихославль'),  ('Лобня'),  ('Лодейное Поле'),  ('Лосино-Петровский'),  ('Луга'),  ('Луза'),  ('Лукоянов');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Луховицы'),  ('Лысково'),  ('Лысьва'),  ('Лыткарино'),  ('Льгов'),  ('Любань'),  ('Люберцы'),  ('Любим'),  ('Людиново'),  ('Лянтор');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Магадан'),  ('Магас'),  ('Магнитогорск'),  ('Майкоп'),  ('Майский'),  ('Макаров'),  ('Макарьев'),  ('Макушино'),  ('Малая Вишера'),  ('Малгобек');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Малмыж'),  ('Малоархангельск'),  ('Малоярославец'),  ('Мамадыш'),  ('Мамоново'),  ('Мантурово'),  ('Мариинск'),  ('Мариинский Посад'),  ('Маркс'),  ('Махачкала');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Мглин'),  ('Мегион'),  ('Медвежьегорск'),  ('Медногорск'),  ('Медынь'),  ('Межгорье'),  ('Междуреченск'),  ('Мезень'),  ('Меленки'),  ('Мелеуз');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Менделеевск'),  ('Мензелинск'),  ('Мещовск'),  ('Миасс'),  ('Микунь'),  ('Миллерово'),  ('Минеральные Воды'),  ('Минусинск'),  ('Миньяр'),  ('Мирный');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Михайлов'),  ('Михайловка'),  ('Михайловск'),  ('Мичуринск'),  ('Могоча'),  ('Можайск'),  ('Можга'),  ('Моздок'),  ('Мончегорск'),  ('Морозовск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Моршанск'),  ('Мосальск'),  ('Москва'),  ('Муравленко'),  ('Мураши'),  ('Мурино'),  ('Мурманск'),  ('Муром'),  ('Мценск'),  ('Мыски');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Мытищи'),  ('Мышкин'),  ('Набережные Челны'),  ('Навашино'),  ('Наволоки'),  ('Надым'),  ('Назарово'),  ('Назрань'),  ('Называевск'),  ('Нальчик');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Нариманов'),  ('Наро-Фоминск'),  ('Нарткала'),  ('Нарьян-Мар'),  ('Находка'),  ('Невель'),  ('Невельск'),  ('Невинномысск'),  ('Невьянск'),  ('Нелидово');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Неман'),  ('Нерехта'),  ('Нерчинск'),  ('Нерюнгри'),  ('Нестеров'),  ('Нефтегорск'),  ('Нефтекамск'),  ('Нефтекумск'),  ('Нефтеюганск'),  ('Нея');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Нижневартовск'),  ('Нижнекамск'),  ('Нижнеудинск'),  ('Нижние Серги'),  ('Нижний Ломов'),  ('Нижний Новгород'),  ('Нижний Тагил'),  ('Нижняя Салда'),  ('Нижняя Тура'),  ('Николаевск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Николаевск-на-Амуре'),  ('Никольск'),  ('Никольское'),  ('Новая Ладога'),  ('Новая Ляля'),  ('Новоалександровск'),  ('Новоалтайск'),  ('Новоаннинский'),  ('Нововоронеж'),  ('Новодвинск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Новозыбков'),  ('Новокубанск'),  ('Новокузнецк'),  ('Новокуйбышевск'),  ('Новомичуринск'),  ('Новомосковск'),  ('Новопавловск'),  ('Новоржев'),  ('Новороссийск'),  ('Новосибирск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Новосиль'),  ('Новосокольники'),  ('Новотроицк'),  ('Новоузенск'),  ('Новоульяновск'),  ('Новоуральск'),  ('Новохопёрск'),  ('Новочебоксарск'),  ('Новочеркасск'),  ('Новошахтинск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Новый Оскол'),  ('Новый Уренгой'),  ('Ногинск'),  ('Нолинск'),  ('Норильск'),  ('Ноябрьск'),  ('Нурлат'),  ('Нытва'),  ('Нюрба'),  ('Нягань');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Нязепетровск'),  ('Няндома'),  ('Облучье'),  ('Обнинск'),  ('Обоянь'),  ('Обь'),  ('Одинцово'),  ('Озёрск'),  ('Озёры'),  ('Октябрьск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Октябрьский'),  ('Окуловка'),  ('Олёкминск'),  ('Оленегорск'),  ('Олонец'),  ('Омск'),  ('Омутнинск'),  ('Онега'),  ('Опочка'),  ('Орёл');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Оренбург'),  ('Орехово-Зуево'),  ('Орлов'),  ('Орск'),  ('Оса'),  ('Осинники'),  ('Осташков'),  ('Остров'),  ('Островной'),  ('Острогожск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Отрадное'),  ('Отрадный'),  ('Оха'),  ('Оханск'),  ('Очёр'),  ('Павлово'),  ('Павловск'),  ('Павловский Посад'),  ('Палласовка'),  ('Партизанск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Певек'),  ('Пенза'),  ('Первомайск'),  ('Первоуральск'),  ('Перевоз'),  ('Пересвет'),  ('Переславль-Залесский'),  ('Пермь'),  ('Пестово'),  ('Петров Вал');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Петровск'),  ('Петровск-Забайкальский'),  ('Петрозаводск'),  ('Петропавловск-Камчатский'),  ('Петухово'),  ('Петушки'),  ('Печора'),  ('Печоры'),  ('Пикалёво'),  ('Пионерский');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Питкяранта'),  ('Плавск'),  ('Пласт'),  ('Плёс'),  ('Поворино'),  ('Подольск'),  ('Подпорожье'),  ('Покачи'),  ('Покров'),  ('Покровск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Полевской'),  ('Полесск'),  ('Полысаево'),  ('Полярные Зори'),  ('Полярный'),  ('Поронайск'),  ('Порхов'),  ('Похвистнево'),  ('Почеп'),  ('Починок');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Пошехонье'),  ('Правдинск'),  ('Приволжск'),  ('Приморск'),  ('Приморско-Ахтарск'),  ('Приозерск'),  ('Прокопьевск'),  ('Пролетарск'),  ('Протвино'),  ('Прохладный');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Псков'),  ('Пугачёв'),  ('Пудож'),  ('Пустошка'),  ('Пучеж'),  ('Пушкино'),  ('Пущино'),  ('Пыталово'),  ('Пыть-Ях'),  ('Пятигорск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Радужный'),  ('Райчихинск'),  ('Раменское'),  ('Рассказово'),  ('Ревда'),  ('Реж'),  ('Реутов'),  ('Ржев'),  ('Родники'),  ('Рославль');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Россошь'),  ('Ростов-на-Дону'),  ('Ростов'),  ('Рошаль'),  ('Ртищево'),  ('Рубцовск'),  ('Рудня'),  ('Руза'),  ('Рузаевка'),  ('Рыбинск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Рыбное'),  ('Рыльск'),  ('Ряжск'),  ('Рязань'),  ('Сакине призн.'),  ('Салават'),  ('Салаир'),  ('Салехард'),  ('Сальск'),  ('Самара');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Санкт-Петербург'),  ('Саранск'),  ('Сарапул'),  ('Саратов'),  ('Саров'),  ('Сасово'),  ('Сатка'),  ('Сафоново'),  ('Саяногорск'),  ('Саянск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Светлогорск'),  ('Светлоград'),  ('Светлый'),  ('Светогорск'),  ('Свирск'),  ('Свободный'),  ('Себеж'),  ('Севастопольне призн.'),  ('Северо-Курильск'),  ('Северобайкальск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Северодвинск'),  ('Североморск'),  ('Североуральск'),  ('Северск'),  ('Севск'),  ('Сегежа'),  ('Сельцо'),  ('Семёнов'),  ('Семикаракорск'),  ('Семилуки');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Сенгилей'),  ('Серафимович'),  ('Сергач'),  ('Сергиев Посад'),  ('Сердобск'),  ('Серов'),  ('Серпухов'),  ('Сертолово'),  ('Сибай'),  ('Сим');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Симферопольне призн.'),  ('Сковородино'),  ('Скопин'),  ('Славгород'),  ('Славск'),  ('Славянск-на-Кубани'),  ('Сланцы'),  ('Слободской'),  ('Слюдянка'),  ('Смоленск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Снежинск'),  ('Снежногорск'),  ('Собинка'),  ('Советск'),  ('Советская Гавань'),  ('Советский'),  ('Сокол'),  ('Солигалич'),  ('Соликамск'),  ('Солнечногорск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Соль-Илецк'),  ('Сольвычегодск'),  ('Сольцы'),  ('Сорочинск'),  ('Сорск'),  ('Сортавала'),  ('Сосенский'),  ('Сосновка'),  ('Сосновоборск'),  ('Сосновый Бор');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Сосногорск'),  ('Сочи'),  ('Спас-Деменск'),  ('Спас-Клепики'),  ('Спасск'),  ('Спасск-Дальний'),  ('Спасск-Рязанский'),  ('Среднеколымск'),  ('Среднеуральск'),  ('Сретенск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Ставрополь'),  ('Старая Купавна'),  ('Старая Русса'),  ('Старица'),  ('Стародуб'),  ('Старый Крымне призн.'),  ('Старый Оскол'),  ('Стерлитамак'),  ('Стрежевой'),  ('Строитель');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Струнино'),  ('Ступино'),  ('Суворов'),  ('Судакне призн.'),  ('Суджа'),  ('Судогда'),  ('Суздаль'),  ('Сунжа'),  ('Суоярви'),  ('Сураж');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Сургут'),  ('Суровикино'),  ('Сурск'),  ('Сусуман'),  ('Сухиничи'),  ('Сухой Лог'),  ('Сызрань'),  ('Сыктывкар'),  ('Сысерть'),  ('Сычёвка');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Сясьстрой'),  ('Тавда'),  ('Таганрог'),  ('Тайга'),  ('Тайшет'),  ('Талдом'),  ('Талица'),  ('Тамбов'),  ('Тара'),  ('Тарко-Сале');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Таруса'),  ('Татарск'),  ('Таштагол'),  ('Тверь'),  ('Теберда'),  ('Тейково'),  ('Темников'),  ('Темрюк'),  ('Терек'),  ('Тетюши');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Тимашёвск'),  ('Тихвин'),  ('Тихорецк'),  ('Тобольск'),  ('Тогучин'),  ('Тольятти'),  ('Томари'),  ('Томмот'),  ('Томск'),  ('Топки');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Торжок'),  ('Торопец'),  ('Тосно'),  ('Тотьма'),  ('Трёхгорный'),  ('Троицк'),  ('Трубчевск'),  ('Туапсе'),  ('Туймазы'),  ('Тула');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Тулун'),  ('Туран'),  ('Туринск'),  ('Тутаев'),  ('Тында'),  ('Тырныауз'),  ('Тюкалинск'),  ('Тюмень'),  ('Уварово'),  ('Углегорск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Углич'),  ('Удачный'),  ('Удомля'),  ('Ужур'),  ('Узловая'),  ('Улан-Удэ'),  ('Ульяновск'),  ('Унеча'),  ('Урай'),  ('Урень');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Уржум'),  ('Урус-Мартан'),  ('Урюпинск'),  ('Усинск'),  ('Усмань'),  ('Усолье-Сибирское'),  ('Усолье'),  ('Уссурийск'),  ('Усть-Джегута'),  ('Усть-Илимск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Усть-Катав'),  ('Усть-Кут'),  ('Усть-Лабинск'),  ('Устюжна'),  ('Уфа'),  ('Ухта'),  ('Учалы'),  ('Уяр'),  ('Фатеж'),  ('Феодосияне призн.');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Фокино'),  ('Фролово'),  ('Фрязино'),  ('Фурманов'),  ('Хабаровск'),  ('Хадыженск'),  ('Ханты-Мансийск'),  ('Харабали'),  ('Харовск'),  ('Хасавюрт');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Хвалынск'),  ('Хилок'),  ('Химки'),  ('Холм'),  ('Холмск'),  ('Хотьково'),  ('Цивильск'),  ('Цимлянск'),  ('Циолковский'),  ('Чадан');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Чайковский'),  ('Чапаевск'),  ('Чаплыгин'),  ('Чебаркуль'),  ('Чебоксары'),  ('Чегем'),  ('Чекалин'),  ('Челябинск'),  ('Чердынь'),  ('Черемхово');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Черепаново'),  ('Череповец'),  ('Черкесск'),  ('Чёрмоз'),  ('Черноголовка'),  ('Черногорск'),  ('Чернушка'),  ('Черняховск'),  ('Чехов'),  ('Чистополь');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Чита'),  ('Чкаловск'),  ('Чудово'),  ('Чулым'),  ('Чусовой'),  ('Чухлома'),  ('Шагонар'),  ('Шадринск'),  ('Шали'),  ('Шарыпово');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Шарья'),  ('Шатура'),  ('Шахты'),  ('Шахунья'),  ('Шацк'),  ('Шебекино'),  ('Шелехов'),  ('Шенкурск'),  ('Шилка'),  ('Шимановск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Шиханы'),  ('Шлиссельбург'),  ('Шумерля'),  ('Шумиха'),  ('Шуя'),  ('Щёкино'),  ('Щёлкиноне призн.'),  ('Щёлково'),  ('Щигры'),  ('Щучье');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Электрогорск'),  ('Электросталь'),  ('Электроугли'),  ('Элиста'),  ('Энгельс'),  ('Эртиль'),  ('Югорск'),  ('Южа'),  ('Южно-Сахалинск'),  ('Южно-Сухокумск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Южноуральск'),  ('Юрга'),  ('Юрьев-Польский'),  ('Юрьевец'),  ('Юрюзань'),  ('Юхнов'),  ('Ядрин'),  ('Якутск'),  ('Ялтане призн.'),  ('Ялуторовск');  **INSERT** **INTO** public.cities ("name") **VALUES**  ('Янаул'),  ('Яранск'),  ('Яровое'),  ('Ярославль'),  ('Ярцево'),  ('Ясногорск'),  ('Ясный'),  ('Яхрома');  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Абхазия','Abkhazia','AB',895),  ('Австралия','Australia','AU',36),  ('Австрия','Austria','AT',40),  ('Азербайджан','Azerbaijan','AZ',31),  ('Албания','Albania','AL',8),  ('Алжир','Algeria','DZ',12),  ('Американское Самоа','American Samoa','AS',16),  ('Ангилья','Anguilla','AI',660),  ('Ангола','Angola','AO',24),  ('Андорра','Andorra','AD',20);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Антарктида','Antarctica','AQ',10),  ('Антигуа и Барбуда','Antigua and Barbuda','AG',28),  ('Аргентина','Argentina','AR',32),  ('Армения','Armenia','AM',51),  ('Аруба','Aruba','AW',533),  ('Афганистан','Afghanistan','AF',4),  ('Багамы','Bahamas','BS',44),  ('Бангладеш','Bangladesh','BD',50),  ('Барбадос','Barbados','BB',52),  ('Бахрейн','Bahrain','BH',48);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Беларусь','Belarus','BY',112),  ('Белиз','Belize','BZ',84),  ('Бельгия','Belgium','BE',56),  ('Бенин','Benin','BJ',204),  ('Бермуды','Bermuda','BM',60),  ('Болгария','Bulgaria','BG',100),  ('Боливия, Многонациональное Государство','Bolivia, plurinational state of','BO',68),  ('Бонайре, Саба и Синт-Эстатиус','Bonaire, Sint Eustatius and Saba','BQ',535),  ('Босния и Герцеговина','Bosnia and Herzegovina','BA',70),  ('Ботсвана','Botswana','BW',72);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Бразилия','Brazil','BR',76),  ('Британская территория в Индийском океане','British Indian Ocean Territory','IO',86),  ('Бруней-Даруссалам','Brunei Darussalam','BN',96),  ('Буркина-Фасо','Burkina Faso','BF',854),  ('Бурунди','Burundi','BI',108),  ('Бутан','Bhutan','BT',64),  ('Вануату','Vanuatu','VU',548),  ('Венгрия','Hungary','HU',348),  ('Венесуэла Боливарианская Республика','Venezuela','VE',862),  ('Виргинские острова, Британские','Virgin Islands, British','VG',92);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Виргинские острова, США','Virgin Islands, U.S.','VI',850),  ('Вьетнам','Vietnam','VN',704),  ('Габон','Gabon','GA',266),  ('Гаити','Haiti','HT',332),  ('Гайана','Guyana','GY',328),  ('Гамбия','Gambia','GM',270),  ('Гана','Ghana','GH',288),  ('Гваделупа','Guadeloupe','GP',312),  ('Гватемала','Guatemala','GT',320),  ('Гвинея','Guinea','GN',324);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Гвинея-Бисау','Guinea-Bissau','GW',624),  ('Германия','Germany','DE',276),  ('Гернси','Guernsey','GG',831),  ('Гибралтар','Gibraltar','GI',292),  ('Гондурас','Honduras','HN',340),  ('Гонконг','Hong Kong','HK',344),  ('Гренада','Grenada','GD',308),  ('Гренландия','Greenland','GL',304),  ('Греция','Greece','GR',300),  ('Грузия','Georgia','GE',268);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Гуам','Guam','GU',316),  ('Дания','Denmark','DK',208),  ('Джерси','Jersey','JE',832),  ('Джибути','Djibouti','DJ',262),  ('Доминика','Dominica','DM',212),  ('Доминиканская Республика','Dominican Republic','DO',214),  ('Египет','Egypt','EG',818),  ('Замбия','Zambia','ZM',894),  ('Западная Сахара','Western Sahara','EH',732),  ('Зимбабве','Zimbabwe','ZW',716);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Израиль','Israel','IL',376),  ('Индия','India','IN',356),  ('Индонезия','Indonesia','ID',360),  ('Иордания','Jordan','JO',400),  ('Ирак','Iraq','IQ',368),  ('Иран, Исламская Республика','Iran, Islamic Republic of','IR',364),  ('Ирландия','Ireland','IE',372),  ('Исландия','Iceland','IS',352),  ('Испания','Spain','ES',724),  ('Италия','Italy','IT',380);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Йемен','Yemen','YE',887),  ('Кабо-Верде','Cape Verde','CV',132),  ('Казахстан','Kazakhstan','KZ',398),  ('Камбоджа','Cambodia','KH',116),  ('Камерун','Cameroon','CM',120),  ('Канада','Canada','CA',124),  ('Катар','Qatar','QA',634),  ('Кения','Kenya','KE',404),  ('Кипр','Cyprus','CY',196),  ('Киргизия','Kyrgyzstan','KG',417);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Кирибати','Kiribati','KI',296),  ('Китай','China','CN',156),  ('Кокосовые (Килинг) острова','Cocos (Keeling) Islands','CC',166),  ('Колумбия','Colombia','CO',170),  ('Коморы','Comoros','KM',174),  ('Конго','Congo','CG',178),  ('Конго, Демократическая Республика','Congo, Democratic Republic of the','CD',180),  ('Корея, Народно-Демократическая Республика','Korea, Democratic People''s republic of','KP',408),  ('Корея, Республика','Korea, Republic of','KR',410),  ('Коста-Рика','Costa Rica','CR',188);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Кот д''Ивуар','Cote d''Ivoire','CI',384),  ('Куба','Cuba','CU',192),  ('Кувейт','Kuwait','KW',414),  ('Кюрасао','Curaçao','CW',531),  ('Лаос','Lao People''s Democratic Republic','LA',418),  ('Латвия','Latvia','LV',428),  ('Лесото','Lesotho','LS',426),  ('Ливан','Lebanon','LB',422),  ('Ливийская Арабская Джамахирия','Libyan Arab Jamahiriya','LY',434),  ('Либерия','Liberia','LR',430);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Лихтенштейн','Liechtenstein','LI',438),  ('Литва','Lithuania','LT',440),  ('Люксембург','Luxembourg','LU',442),  ('Маврикий','Mauritius','MU',480),  ('Мавритания','Mauritania','MR',478),  ('Мадагаскар','Madagascar','MG',450),  ('Майотта','Mayotte','YT',175),  ('Макао','Macao','MO',446),  ('Малави','Malawi','MW',454),  ('Малайзия','Malaysia','MY',458);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Мали','Mali','ML',466),  ('Малые Тихоокеанские отдаленные острова Соединенных Штатов','United States Minor Outlying Islands','UM',581),  ('Мальдивы','Maldives','MV',462),  ('Мальта','Malta','MT',470),  ('Марокко','Morocco','MA',504),  ('Мартиника','Martinique','MQ',474),  ('Маршалловы острова','Marshall Islands','MH',584),  ('Мексика','Mexico','MX',484),  ('Микронезия, Федеративные Штаты','Micronesia, Federated States of','FM',583),  ('Мозамбик','Mozambique','MZ',508);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Молдова, Республика','Moldova','MD',498),  ('Монако','Monaco','MC',492),  ('Монголия','Mongolia','MN',496),  ('Монтсеррат','Montserrat','MS',500),  ('Мьянма','Myanmar','MM',104),  ('Намибия','Namibia','NA',516),  ('Науру','Nauru','NR',520),  ('Непал','Nepal','NP',524),  ('Нигер','Niger','NE',562),  ('Нигерия','Nigeria','NG',566);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Нидерланды','Netherlands','NL',528),  ('Никарагуа','Nicaragua','NI',558),  ('Ниуэ','Niue','NU',570),  ('Новая Зеландия','New Zealand','NZ',554),  ('Новая Каледония','New Caledonia','NC',540),  ('Норвегия','Norway','NO',578),  ('Объединенные Арабские Эмираты','United Arab Emirates','AE',784),  ('Оман','Oman','OM',512),  ('Остров Буве','Bouvet Island','BV',74),  ('Остров Мэн','Isle of Man','IM',833);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Остров Норфолк','Norfolk Island','NF',574),  ('Остров Рождества','Christmas Island','CX',162),  ('Остров Херд и острова Макдональд','Heard Island and McDonald Islands','HM',334),  ('Острова Кайман','Cayman Islands','KY',136),  ('Острова Кука','Cook Islands','CK',184),  ('Острова Теркс и Кайкос','Turks and Caicos Islands','TC',796),  ('Пакистан','Pakistan','PK',586),  ('Палау','Palau','PW',585),  ('Палестинская территория, оккупированная','Palestinian Territory, Occupied','PS',275),  ('Панама','Panama','PA',591);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Папский Престол (Государство — город Ватикан)','Holy See (Vatican City State)','VA',336),  ('Папуа-Новая Гвинея','Papua New Guinea','PG',598),  ('Парагвай','Paraguay','PY',600),  ('Перу','Peru','PE',604),  ('Питкерн','Pitcairn','PN',612),  ('Польша','Poland','PL',616),  ('Португалия','Portugal','PT',620),  ('Пуэрто-Рико','Puerto Rico','PR',630),  ('Республика Македония','Macedonia, The Former Yugoslav Republic Of','MK',807),  ('Реюньон','Reunion','RE',638);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Россия','Russian Federation','RU',643),  ('Руанда','Rwanda','RW',646),  ('Румыния','Romania','RO',642),  ('Самоа','Samoa','WS',882),  ('Сан-Марино','San Marino','SM',674),  ('Сан-Томе и Принсипи','Sao Tome and Principe','ST',678),  ('Саудовская Аравия','Saudi Arabia','SA',682),  ('Свазиленд','Swaziland','SZ',748),  ('Святая Елена, Остров вознесения, Тристан-да-Кунья','Saint Helena, Ascension And Tristan Da Cunha','SH',654),  ('Северные Марианские острова','Northern Mariana Islands','MP',580);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Сен-Бартельми','Saint Barthélemy','BL',652),  ('Сен-Мартен','Saint Martin (French Part)','MF',663),  ('Сенегал','Senegal','SN',686),  ('Сент-Винсент и Гренадины','Saint Vincent and the Grenadines','VC',670),  ('Сент-Люсия','Saint Lucia','LC',662),  ('Сент-Китс и Невис','Saint Kitts and Nevis','KN',659),  ('Сент-Пьер и Микелон','Saint Pierre and Miquelon','PM',666),  ('Сербия','Serbia','RS',688),  ('Сейшелы','Seychelles','SC',690),  ('Сингапур','Singapore','SG',702);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Синт-Мартен','Sint Maarten','SX',534),  ('Сирийская Арабская Республика','Syrian Arab Republic','SY',760),  ('Словакия','Slovakia','SK',703),  ('Словения','Slovenia','SI',705),  ('Соединенное Королевство','United Kingdom','GB',826),  ('Соединенные Штаты','United States','US',840),  ('Соломоновы острова','Solomon Islands','SB',90),  ('Сомали','Somalia','SO',706),  ('Судан','Sudan','SD',729),  ('Суринам','Suriname','SR',740);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Сьерра-Леоне','Sierra Leone','SL',694),  ('Таджикистан','Tajikistan','TJ',762),  ('Таиланд','Thailand','TH',764),  ('Тайвань (Китай)','Taiwan, Province of China','TW',158),  ('Танзания, Объединенная Республика','Tanzania, United Republic Of','TZ',834),  ('Тимор-Лесте','Timor-Leste','TL',626),  ('Того','Togo','TG',768),  ('Токелау','Tokelau','TK',772),  ('Тонга','Tonga','TO',776),  ('Тринидад и Тобаго','Trinidad and Tobago','TT',780);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Тувалу','Tuvalu','TV',798),  ('Тунис','Tunisia','TN',788),  ('Туркмения','Turkmenistan','TM',795),  ('Турция','Turkey','TR',792),  ('Уганда','Uganda','UG',800),  ('Узбекистан','Uzbekistan','UZ',860),  ('Украина','Ukraine','UA',804),  ('Уоллис и Футуна','Wallis and Futuna','WF',876),  ('Уругвай','Uruguay','UY',858),  ('Фарерские острова','Faroe Islands','FO',234);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Фиджи','Fiji','FJ',242),  ('Филиппины','Philippines','PH',608),  ('Финляндия','Finland','FI',246),  ('Фолклендские острова (Мальвинские)','Falkland Islands (Malvinas)','FK',238),  ('Франция','France','FR',250),  ('Французская Гвиана','French Guiana','GF',254),  ('Французская Полинезия','French Polynesia','PF',258),  ('Французские Южные территории','French Southern Territories','TF',260),  ('Хорватия','Croatia','HR',191),  ('Центрально-Африканская Республика','Central African Republic','CF',140);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Чад','Chad','TD',148),  ('Черногория','Montenegro','ME',499),  ('Чешская Республика','Czech Republic','CZ',203),  ('Чили','Chile','CL',152),  ('Швейцария','Switzerland','CH',756),  ('Швеция','Sweden','SE',752),  ('Шпицберген и Ян Майен','Svalbard and Jan Mayen','SJ',744),  ('Шри-Ланка','Sri Lanka','LK',144),  ('Эквадор','Ecuador','EC',218),  ('Экваториальная Гвинея','Equatorial Guinea','GQ',226);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Эландские острова','Åland Islands','AX',248),  ('Эль-Сальвадор','El Salvador','SV',222),  ('Эритрея','Eritrea','ER',232),  ('Эстония','Estonia','EE',233),  ('Эфиопия','Ethiopia','ET',231),  ('Южная Африка','South Africa','ZA',710),  ('Южная Джорджия и Южные Сандвичевы острова','South Georgia and the South Sandwich Islands','GS',239),  ('Южная Осетия','South Ossetia','OS',896),  ('Южный Судан','South Sudan','SS',728),  ('Ямайка','Jamaica','JM',388);  **INSERT** **INTO** public.countries ("name",en\_name,code,code\_second) **VALUES**  ('Япония','Japan','JP',392);  **INSERT** **INTO** public.directions ("name") **VALUES**  ('Информационная безопасность'),  ('Биг Дата'),  ('ИТ'),  ('Дизайн'),  ('Аналитика');  **INSERT** **INTO** public.events ("name",date\_start,days,id\_city,id\_type\_event,image) **VALUES**  ('Поликом Про и InfoWatch: круглый стол по информационной безопасности в Санкт-Петербурге ','2022-02-02',2,78,2,'9.png'),  ('Получи «100500» лидов в первые 5 секунд ','2023-09-02',3,78,2,'10.png'),  ('Пользовательские требования ','2023-08-08',2,67,2,'11.png'),  ('Постоянно доступен: Налаживаем омниканальную коммуникацию с клиентами ','2023-10-29',3,56,2,'12.png'),  ('Построение гиперконвергентной инфраструктуры и VDI-решения ','2023-10-28',3,45,2,'13.png'),  ('Практикум «Обогнать конкурентов: усиливаем продажи и создаем клиентский сервис» ','2022-07-28',2,78,2,'14.png'),  ('Практикум по Customer Development с экспертом ФРИИ: грамотные продажи за 4 часа ','2022-06-03',3,67,2,'15.png'),  ('Практический воркшоп по созданию договоренностей в Scrum-команде ','2022-05-16',1,7,2,'16.png'),  ('Первая встреча клуба «Leader stories» ','2022-03-15',1,34,2,'1.png'),  ('Первый в России JAVABOOTCAMP ','2022-10-25',3,56,2,'2.png');  **INSERT** **INTO** public.events ("name",date\_start,days,id\_city,id\_type\_event,image) **VALUES**  ('Встреча клуба Leader Stories «Зачем развивать сотрудников? Они же уйдут» ','2023-04-18',3,6,2,'3.png'),  ('Первый IoT-Forum в Санкт-Петербурге ','2023-05-30',3,8,2,'4.png'),  ('План проекта: практические советы, типичные ошибки ','2023-07-11',2,9,2,'5.png'),  ('Планирование проекта: что делать после того, как выяснили цель ','2022-03-20',1,70,2,'6.png'),  ('Поисковая оптимизация (SEO) ','2022-08-10',1,90,2,'7.png'),  ('Поисковая оптимизация. SEO оптимизация ','2022-08-15',2,80,2,'8.png'),  ('Практическое применение диаграммы потоков данных (DFD, Data Flow Diagram) ','2023-11-27',3,8,2,'17.png'),  ('Практическое применение диаграммы состояний (UML StateChart) ','2022-06-17',3,9,2,'18.png'),  ('Презентационная сессия «TTD: технологии для жизни 2022» ','2022-01-30',2,3,2,'19.png'),  ('Презентация курса «Методы, технологии, инструменты обучения персонала в технических, продуктовых и IT-компаниях» ','2023-10-01',3,5,2,'20.png');  **INSERT** **INTO** public.genders ("name") **VALUES**  ('мужской'),  ('женский');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (1,'image1.png'),  (2,'image2.png'),  (3,**NULL**),  (4,'image3.png'),  (5,'image4.png'),  (6,**NULL**),  (7,'image5.png'),  (8,'image6.png'),  (9,'image7.png'),  (10,'image8.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (11,**NULL**),  (12,'image9.png'),  (13,'image10.png'),  (14,**NULL**),  (15,'image11.png'),  (16,'image12.png'),  (17,**NULL**),  (18,'image13.png'),  (19,'image14.png'),  (20,**NULL**);  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (21,'image15.png'),  (22,**NULL**),  (23,'image16.png'),  (24,'image17.png'),  (25,'image18.png'),  (26,**NULL**),  (27,'image19.png'),  (28,'image20.png'),  (29,'image21.png'),  (30,'image22.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (31,'image23.png'),  (32,'image24.png'),  (33,'image25.png'),  (34,'image26.png'),  (35,'image27.png'),  (36,'image28.png'),  (37,'image29.png'),  (38,'image30.png'),  (39,'image31.png'),  (40,'image32.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (41,'image33.png'),  (42,'image34.png'),  (43,'image35.png'),  (44,'image36.png'),  (45,'image37.png'),  (46,'image38.png'),  (47,'image39.png'),  (48,'image40.png'),  (49,'image41.png'),  (50,'image42.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (51,'image43.png'),  (52,'image44.png'),  (53,'image45.png'),  (54,'image46.png'),  (55,'image47.png'),  (56,'image48.png'),  (57,'image49.png'),  (58,'image50.png'),  (59,'image51.png'),  (60,'image52.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (61,'image53.png'),  (62,'image54.png'),  (63,'image55.png'),  (64,'image56.png'),  (65,'image57.png'),  (66,'image58.png'),  (67,'image59.png'),  (68,'image60.png'),  (69,'image61.png'),  (70,'image62.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (71,'image63.png'),  (72,'image64.png'),  (73,'image65.png'),  (74,'image66.png'),  (75,'image67.png'),  (76,'image68.png'),  (77,'image69.png'),  (78,'image70.png'),  (79,'image71.png'),  (80,'image72.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (81,'image73.png'),  (82,'image74.png'),  (83,'image75.png'),  (84,'image76.png'),  (85,'image77.png'),  (86,'image78.png'),  (87,'image79.png'),  (88,'image80.png'),  (89,'image81.png'),  (90,'image82.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (91,'image83.png'),  (92,'image84.png'),  (93,'image85.png'),  (94,'image86.png'),  (95,'image87.png'),  (96,'image88.png'),  (97,'image89.png'),  (98,'image90.png'),  (99,'image91.png'),  (100,'image92.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (101,'image93.png'),  (102,'image94.png'),  (103,'image95.png'),  (104,'image96.png'),  (105,'image97.png'),  (106,'image98.png'),  (107,'image99.png'),  (108,'image100.png'),  (109,'image101.png'),  (110,'image102.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (111,'image103.png'),  (112,'image104.png'),  (113,'image105.png'),  (114,'image106.png'),  (115,'image107.png'),  (116,'image108.png'),  (117,'image109.png'),  (118,'image110.png'),  (119,'image111.png'),  (120,'image112.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (121,'image113.png'),  (122,'image114.png'),  (123,'image115.png'),  (124,'image116.png'),  (125,'image117.png'),  (126,'image118.png'),  (127,'image119.png'),  (128,'image120.png'),  (129,'image121.png'),  (130,'image122.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (131,'image123.png'),  (132,'image124.png'),  (133,'image125.png'),  (134,'image126.png'),  (135,'image127.png'),  (136,'image128.png'),  (137,'image129.png'),  (138,'image130.png'),  (139,'image131.png'),  (140,'image132.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (141,'image133.png'),  (142,'image134.png'),  (143,'image135.png'),  (144,'image136.png'),  (145,'image137.png'),  (146,'image138.png'),  (147,'image139.png'),  (148,'image140.png'),  (149,'image141.png'),  (150,'image142.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (151,'image143.png'),  (152,'image144.png'),  (153,'image145.png'),  (154,'image146.png'),  (155,'image147.png'),  (156,'image148.png'),  (157,'image149.png'),  (158,'image150.png'),  (159,'image151.png'),  (160,'image152.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (161,'image153.png'),  (162,'image154.png'),  (163,'image155.png'),  (164,'image156.png'),  (165,'image157.png'),  (166,'image158.png'),  (167,'image159.png'),  (168,'image160.png'),  (169,'image161.png'),  (170,'image162.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (171,'image163.png'),  (172,'image164.png'),  (173,'image165.png'),  (174,'image166.png'),  (175,'image167.png'),  (176,'image168.png'),  (177,'image169.png'),  (178,'image170.png'),  (179,'image171.png'),  (180,'image172.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (181,'image173.png'),  (182,**NULL**),  (183,'image175.png'),  (184,'image176.png'),  (185,'image177.png'),  (186,'image178.png'),  (187,'image179.png'),  (188,'image180.png'),  (189,'image181.png'),  (190,'image182.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (190,'image188.png'),  (190,'image180.png'),  (191,'image183.png'),  (192,'image184.png'),  (193,'image185.png'),  (194,'image186.png'),  (195,'image187.png'),  (196,'image188.png'),  (197,'image189.png'),  (198,'image190.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (199,'image191.png'),  (200,'image192.png'),  (201,'image193.png'),  (202,'image194.png'),  (203,**NULL**),  (204,'image196.png'),  (205,'image197.png'),  (206,'image198.png'),  (207,'image199.png'),  (208,'image200.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (209,'image201.png'),  (210,'image202.png'),  (211,'image203.png'),  (212,'image204.png'),  (213,'image205.png'),  (214,'image206.png'),  (215,'image207.png'),  (216,'image208.png'),  (217,'image209.png'),  (218,'image210.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (219,'image211.png'),  (220,'image212.png'),  (221,'image213.png'),  (222,'image214.png'),  (223,'image215.png'),  (224,'image216.png'),  (225,'image217.png'),  (226,**NULL**),  (227,'image219.png'),  (228,'image220.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (229,'image221.png'),  (230,'image222.png'),  (231,'image223.png'),  (232,'image224.png'),  (233,'image225.png'),  (234,'image226.png'),  (235,'image227.png'),  (236,'image228.png'),  (237,'image229.png'),  (238,'image230.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (239,'image231.png'),  (240,'image232.png'),  (241,'image233.png'),  (242,'image234.png'),  (243,'image235.png'),  (244,**NULL**),  (245,'image237.png'),  (246,'image238.png'),  (247,'image239.png'),  (248,'image240.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (249,'image241.png'),  (250,'image242.png'),  (251,'image243.png'),  (252,'image244.png'),  (253,'image245.png'),  (254,'image246.png'),  (255,'image247.png'),  (256,'image248.png'),  (257,'image249.png'),  (258,'image250.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (259,'image251.png'),  (260,'image252.png'),  (261,'image253.png'),  (262,**NULL**),  (263,'image255.png'),  (264,'image256.png'),  (265,'image257.png'),  (266,'image258.png'),  (267,'image259.png'),  (268,'image260.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (269,'image261.png'),  (270,'image262.png'),  (271,'image263.png'),  (272,'image264.png'),  (273,'image265.png'),  (274,'image266.png'),  (275,'image267.png'),  (276,'image268.png'),  (277,'image269.png'),  (278,'image270.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (279,'image271.png'),  (280,**NULL**),  (281,'image273.png'),  (282,'image274.png'),  (283,'image275.png'),  (284,'image276.png'),  (285,'image277.png'),  (286,'image278.png'),  (287,'image279.png'),  (288,'image280.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (289,'image281.png'),  (290,'image282.png'),  (291,'image283.png'),  (292,'image284.png'),  (293,'image285.png'),  (294,'image286.png'),  (295,**NULL**),  (296,'image288.png'),  (297,'image289.png'),  (298,'image290.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (299,'image291.png'),  (300,'image292.png'),  (301,'image293.png'),  (302,'image294.png'),  (303,'image295.png'),  (304,'image296.png'),  (305,**NULL**),  (306,**NULL**),  (307,'image299.png'),  (308,**NULL**);  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (309,'image301.png'),  (310,**NULL**),  (311,'image303.png'),  (312,'image304.png'),  (313,'image305.png'),  (314,'image306.png'),  (315,'image307.png'),  (316,'image308.png'),  (317,'image309.png'),  (318,'image310.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (319,'image311.png'),  (320,'image312.png'),  (321,'image313.png'),  (322,'image314.png'),  (323,'image315.png'),  (324,'image316.png'),  (325,'image317.png'),  (326,'image318.png'),  (327,'image319.png'),  (328,'image320.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (329,'image321.png'),  (330,'image322.png'),  (331,'image323.png'),  (332,'image324.png'),  (333,'image325.png'),  (334,'image326.png'),  (335,'image327.png'),  (336,'image328.png'),  (337,'image329.png'),  (338,'image330.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (339,'image331.png'),  (340,'image332.png'),  (341,'image333.png'),  (342,'image334.png'),  (343,'image335.png'),  (344,**NULL**),  (345,'image337.png'),  (346,'image338.png'),  (347,'image339.png'),  (348,'image340.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (349,'image341.png'),  (350,'image342.png'),  (351,'image343.png'),  (352,'image344.png'),  (353,'image345.png'),  (354,'image346.png'),  (355,'image347.png'),  (356,'image348.png'),  (357,'image349.png'),  (358,'image350.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (359,'image351.png'),  (360,'image352.png'),  (361,'image353.png'),  (362,'image354.png'),  (363,'image355.png'),  (364,'image356.png'),  (365,'image357.png'),  (366,'image358.png'),  (367,'image359.png'),  (368,'image360.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (369,'image361.png'),  (370,'image362.png'),  (371,'image363.png'),  (372,'image364.png'),  (373,'image365.png'),  (374,'image366.png'),  (375,'image367.png'),  (376,'image368.png'),  (377,'image369.png'),  (378,'image370.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (379,'image371.png'),  (380,'image372.png'),  (381,'image373.png'),  (382,'image374.png'),  (383,'image375.png'),  (384,'image376.png'),  (385,'image377.png'),  (386,'image378.png'),  (387,'image379.png'),  (388,'image380.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (389,'image381.png'),  (390,'image382.png'),  (391,'image383.png'),  (392,'image384.png'),  (393,'image385.png'),  (394,'image386.png'),  (395,'image387.png'),  (396,'image388.png'),  (397,'image389.png'),  (398,'image390.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (399,'image391.png'),  (400,'image392.png'),  (401,'image393.png'),  (402,'image394.png'),  (403,'image395.png'),  (404,'image396.png'),  (405,'image397.png'),  (406,'image398.png'),  (407,'image399.png'),  (408,'image400.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (409,'image401.png'),  (410,'image402.png'),  (411,'image403.png'),  (412,'image404.png'),  (413,'image405.png'),  (414,'image406.png'),  (415,'image407.png'),  (416,'image408.png'),  (417,'image409.png'),  (418,'image410.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (418,'image413.png'),  (418,'image428.png'),  (419,'image411.png'),  (420,'image412.png'),  (421,'image413.png'),  (422,'image414.png'),  (423,'image415.png'),  (424,'image416.png'),  (425,'image417.png'),  (426,'image418.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (427,'image419.png'),  (428,'image420.png'),  (429,'image421.png'),  (430,'image422.png'),  (431,'image423.png'),  (432,'image424.png'),  (433,'image425.png'),  (434,'image426.png'),  (435,'image427.png'),  (436,'image428.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (437,'image429.png'),  (438,'image430.png'),  (439,'image431.png'),  (440,'image432.png'),  (441,'image433.png'),  (442,'image434.png'),  (443,'image435.png'),  (444,'image436.png'),  (445,'image437.png'),  (446,'image438.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (447,'image439.png'),  (448,'image440.png'),  (449,'image441.png'),  (450,'image442.png'),  (451,'image443.png'),  (452,'image444.png'),  (453,'image445.png'),  (454,'image446.png'),  (455,'image447.png'),  (456,'image448.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (457,'image449.png'),  (458,'image450.png'),  (459,'image451.png'),  (460,'image452.png'),  (461,'image453.png'),  (462,'image454.png'),  (463,'image455.png'),  (464,'image456.png'),  (465,'image457.png'),  (466,'image458.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (467,'image459.png'),  (468,'image460.png'),  (469,'image461.png'),  (470,'image462.png'),  (471,'image463.png'),  (472,'image464.png'),  (473,'image465.png'),  (474,'image466.png'),  (475,'image467.png'),  (476,'image468.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (477,'image469.png'),  (478,'image470.png'),  (479,'image471.png'),  (480,'image472.png'),  (481,'image473.png'),  (482,'image474.png'),  (483,'image475.png'),  (484,'image476.png'),  (485,'image477.png'),  (486,'image478.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (487,'image479.png'),  (488,'image480.png'),  (489,'image481.png'),  (490,'image482.png'),  (491,'image483.png'),  (492,'image484.png'),  (493,'image485.png'),  (494,'image486.png'),  (495,'image487.png'),  (496,'image488.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (497,'image489.png'),  (498,'image490.png'),  (499,'image491.png'),  (500,'image492.png'),  (501,'image493.png'),  (502,'image494.png'),  (503,'image495.png'),  (504,'image496.png'),  (505,'image497.png'),  (506,'image498.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (507,'image499.png'),  (508,'image500.png'),  (509,'image501.png'),  (510,'image502.png'),  (511,'image503.png'),  (512,'image504.png'),  (513,'image505.png'),  (514,'image506.png'),  (515,'image507.png'),  (516,'image508.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (517,'image509.png'),  (518,'image510.png'),  (519,'image511.png'),  (520,'image512.png'),  (521,'image513.png'),  (522,'image514.png'),  (523,'image515.png'),  (524,'image516.png'),  (525,'image517.png'),  (526,'image518.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (527,'image519.png'),  (528,'image520.png'),  (529,'image521.png'),  (530,'image522.png'),  (531,'image523.png'),  (532,'image524.png'),  (533,'image525.png'),  (534,'image526.png'),  (535,'image527.png'),  (536,'image528.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (537,'image529.png'),  (538,'image530.png'),  (539,'image531.png'),  (540,'image532.png'),  (541,'image533.png'),  (542,'image534.png'),  (543,'image535.png'),  (544,'image536.png'),  (545,'image537.png'),  (546,'image538.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (547,'image539.png'),  (548,'image540.png'),  (549,'image541.png'),  (550,'image542.png'),  (551,'image543.png'),  (552,'image544.png'),  (553,'image545.png'),  (554,'image546.png'),  (555,'image547.png'),  (556,'image548.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (557,'image549.png'),  (558,'image550.png'),  (559,'image551.png'),  (560,'image552.png'),  (561,'image553.png'),  (562,'image554.png'),  (563,'image555.png'),  (564,'image556.png'),  (565,'image557.png'),  (566,'image558.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (567,'image559.png'),  (568,'image560.png'),  (569,'image561.png'),  (570,'image562.png'),  (571,'image563.png'),  (572,'image564.png'),  (573,'image565.png'),  (574,'image566.png'),  (575,'image567.png'),  (576,'image568.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (577,'image569.png'),  (578,'image570.png'),  (579,'image571.png'),  (580,'image572.png'),  (581,'image573.png'),  (582,'image574.png'),  (583,'image575.png'),  (584,'image576.png'),  (585,'image577.png'),  (586,'image578.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (587,'image579.png'),  (588,'image580.png'),  (589,'image581.png'),  (590,'image582.png'),  (591,'image583.png'),  (592,'image584.png'),  (593,'image585.png'),  (594,'image586.png'),  (595,'image587.png'),  (596,'image588.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (597,'image589.png'),  (598,'image590.png'),  (599,'image591.png'),  (600,'image592.png'),  (601,'image593.png'),  (602,'image594.png'),  (603,'image595.png'),  (604,'image596.png'),  (605,'image597.png'),  (606,'image598.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (607,'image599.png'),  (608,'image600.png'),  (609,'image601.png'),  (610,'image602.png'),  (611,'image603.png'),  (612,'image604.png'),  (613,'image605.png'),  (614,'image606.png'),  (615,'image607.png'),  (616,'image608.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (617,'image609.png'),  (618,'image610.png'),  (619,'image611.png'),  (620,'image612.png'),  (621,'image613.png'),  (622,'image614.png'),  (623,'image615.png'),  (624,**NULL**),  (625,'image617.png'),  (626,'image618.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (627,**NULL**),  (628,'image620.png'),  (629,**NULL**),  (630,'image622.png'),  (631,'image623.png'),  (632,'image624.png'),  (633,'image625.png'),  (634,'image626.png'),  (635,'image627.png'),  (636,'image628.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (637,**NULL**),  (638,'image630.png'),  (639,'image631.png'),  (640,'image632.png'),  (641,'image633.png'),  (642,'image634.png'),  (643,'image635.png'),  (644,'image636.png'),  (645,'image637.png'),  (646,'image638.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (647,'image639.png'),  (648,'image640.png'),  (649,'image641.png'),  (650,'image642.png'),  (651,'image643.png'),  (652,'image644.png'),  (653,'image645.png'),  (654,'image646.png'),  (655,'image647.png'),  (656,'image648.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (657,'image649.png'),  (658,'image650.png'),  (659,'image651.png'),  (660,'image652.png'),  (661,'image653.png'),  (662,'image654.png'),  (663,'image655.png'),  (664,'image656.png'),  (665,'image657.png'),  (666,'image658.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (667,'image659.png'),  (668,'image660.png'),  (669,'image661.png'),  (670,'image662.png'),  (671,'image663.png'),  (672,'image664.png'),  (673,'image665.png'),  (674,'image666.png'),  (675,'image667.png'),  (676,'image668.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (677,'image669.png'),  (678,'image670.png'),  (679,'image671.png'),  (680,'image672.png'),  (681,'image673.png'),  (682,'image674.png'),  (683,'image675.png'),  (684,'image676.png'),  (685,'image677.png'),  (686,'image678.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (687,'image679.png'),  (688,'image680.png'),  (689,'image681.png'),  (690,'image682.png'),  (691,'image683.png'),  (692,'image684.png'),  (693,'image685.png'),  (694,'image686.png'),  (695,'image687.png'),  (696,'image688.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (697,'image689.png'),  (698,'image690.png'),  (699,'image691.png'),  (700,'image692.png'),  (701,'image693.png'),  (702,'image694.png'),  (703,'image695.png'),  (704,'image696.png'),  (705,'image697.png'),  (706,'image698.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (707,'image699.png'),  (708,'image700.png'),  (709,'image701.png'),  (710,'image702.png'),  (711,'image703.png'),  (712,'image704.png'),  (713,'image705.png'),  (714,'image706.png'),  (715,'image707.png'),  (716,'image708.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (717,'image709.png'),  (718,'image710.png'),  (719,'image711.png'),  (720,'image712.png'),  (721,'image713.png'),  (722,'image714.png'),  (723,'image715.png'),  (724,'image716.png'),  (725,'image717.png'),  (726,'image718.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (727,'image719.png'),  (728,'image720.png'),  (729,'image721.png'),  (730,'image722.png'),  (731,'image723.png'),  (732,'image724.png'),  (733,'image725.png'),  (734,'image726.png'),  (735,'image727.png'),  (736,'image728.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (737,'image729.png'),  (738,'image730.png'),  (739,'image731.png'),  (740,'image732.png'),  (741,'image733.png'),  (742,'image734.png'),  (743,'image735.png'),  (744,'image736.png'),  (745,'image737.png'),  (746,'image738.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (747,'image739.png'),  (748,'image740.png'),  (749,'image741.png'),  (750,'image742.png'),  (751,'image743.png'),  (752,'image744.png'),  (753,'image745.png'),  (754,'image746.png'),  (755,'image747.png'),  (756,'image748.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (757,'image749.png'),  (758,'image750.png'),  (759,'image751.png'),  (760,'image752.png'),  (761,'image753.png'),  (762,'image754.png'),  (763,'image755.png'),  (764,'image756.png'),  (765,'image757.png'),  (766,'image758.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (767,'image759.png'),  (768,'image760.png'),  (769,'image761.png'),  (770,'image762.png'),  (771,'image763.png'),  (772,'image764.png'),  (773,'image765.png'),  (774,'image766.png'),  (775,'image767.png'),  (776,'image768.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (777,'image769.png'),  (778,'image770.png'),  (779,'image771.png'),  (780,'image772.png'),  (781,'image773.png'),  (782,'image774.png'),  (783,'image775.png'),  (784,'image776.png'),  (785,'image777.png'),  (786,'image778.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (787,'image779.png'),  (788,'image780.png'),  (789,'image781.png'),  (790,'image782.png'),  (791,'image783.png'),  (792,'image784.png'),  (793,'image785.png'),  (794,'image786.png'),  (795,'image787.png'),  (796,'image788.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (797,'image789.png'),  (798,'image790.png'),  (799,'image791.png'),  (800,'image792.png'),  (801,'image793.png'),  (802,'image794.png'),  (803,'image795.png'),  (804,'image796.png'),  (805,'image797.png'),  (806,'image798.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (807,'image799.png'),  (808,'image800.png'),  (809,'image801.png'),  (810,'image802.png'),  (811,'image803.png'),  (812,'image804.png'),  (813,'image805.png'),  (814,'image806.png'),  (815,'image807.png'),  (816,'image808.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (817,'image809.png'),  (818,'image810.png'),  (819,'image811.png'),  (820,'image812.png'),  (821,'image813.png'),  (822,'image814.png'),  (823,'image815.png'),  (824,'image816.png'),  (825,'image817.png'),  (826,'image818.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (827,'image819.png'),  (828,'image820.png'),  (829,'image821.png'),  (830,'image822.png'),  (831,'image823.png'),  (832,'image824.png'),  (833,'image825.png'),  (834,'image826.png'),  (835,'image827.png'),  (836,'image828.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (837,'image829.png'),  (838,'image830.png'),  (839,'image831.png'),  (840,'image832.png'),  (841,'image833.png'),  (842,'image834.png'),  (843,'image835.png'),  (844,'image836.png'),  (845,'image837.png'),  (846,'image838.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (847,'image839.png'),  (848,'image840.png'),  (849,'image841.png'),  (850,'image842.png'),  (851,'image843.png'),  (852,'image844.png'),  (853,'image845.png'),  (854,'image846.png'),  (855,'image847.png'),  (856,'image848.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (857,'image849.png'),  (858,'image850.png'),  (859,'image851.png'),  (860,'image852.png'),  (861,'image853.png'),  (862,'image854.png'),  (863,'image855.png'),  (864,'image856.png'),  (865,'image857.png'),  (866,'image858.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (867,'image859.png'),  (868,'image860.png'),  (869,'image861.png'),  (870,'image862.png'),  (871,'image863.png'),  (872,'image864.png'),  (873,'image865.png'),  (874,'image866.png'),  (875,'image867.png'),  (876,'image868.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (877,'image869.png'),  (878,'image870.png'),  (879,'image871.png'),  (880,'image872.png'),  (881,'image873.png'),  (882,'image874.png'),  (883,'image875.png'),  (884,'image876.png'),  (885,'image877.png'),  (886,'image878.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (887,'image879.png'),  (888,'image880.png'),  (889,'image881.png'),  (890,'image882.png'),  (891,'image883.png'),  (892,'image884.png'),  (893,'image885.png'),  (894,'image886.png'),  (895,'image887.png'),  (896,'image888.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (897,'image889.png'),  (898,'image890.png'),  (899,'image891.png'),  (900,'image892.png'),  (901,'image893.png'),  (902,'image894.png'),  (903,'image895.png'),  (904,'image896.png'),  (905,'image897.png'),  (906,'image898.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (907,'image899.png'),  (908,'image900.png'),  (909,'image901.png'),  (910,'image902.png'),  (911,'image903.png'),  (912,'image904.png'),  (913,'image905.png'),  (914,'image906.png'),  (915,'image907.png'),  (916,'image908.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (917,'image909.png'),  (918,'image910.png'),  (919,'image911.png'),  (920,'image912.png'),  (921,'image913.png'),  (922,'image914.png'),  (923,'image915.png'),  (924,'image916.png'),  (925,'image917.png'),  (926,'image918.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (927,'image919.png'),  (928,'image920.png'),  (929,'image921.png'),  (930,'image922.png'),  (931,'image923.png'),  (932,'image924.png'),  (933,'image925.png'),  (934,'image926.png'),  (935,'image927.png'),  (936,'image928.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (937,'image929.png'),  (938,'image930.png'),  (939,'image931.png'),  (940,'image932.png'),  (941,'image933.png'),  (942,'image934.png'),  (943,'image935.png'),  (944,'image936.png'),  (945,'image937.png'),  (946,'image938.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (947,'image939.png'),  (948,'image940.png'),  (949,'image941.png'),  (950,'image942.png'),  (951,'image943.png'),  (952,'image944.png'),  (953,'image945.png'),  (954,'image946.png'),  (955,'image947.png'),  (956,'image948.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (957,'image949.png'),  (958,'image950.png'),  (959,'image951.png'),  (960,'image952.png'),  (961,'image953.png'),  (962,'image954.png'),  (963,'image955.png'),  (964,'image956.png'),  (965,'image957.png'),  (966,'image958.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (967,'image959.png'),  (968,'image960.png'),  (969,'image961.png'),  (970,'image962.png'),  (971,'image963.png'),  (972,'image964.png'),  (973,'image965.png'),  (974,'image966.png'),  (975,'image967.png'),  (976,'image968.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (977,'image969.png'),  (978,'image970.png'),  (979,'image971.png'),  (980,'image972.png'),  (981,'image973.png'),  (982,'image974.png'),  (983,'image975.png'),  (984,'image976.png'),  (985,'image977.png'),  (986,'image978.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (987,'image979.png'),  (988,'image980.png'),  (989,'image981.png'),  (990,'image982.png'),  (991,'image983.png'),  (992,'image984.png'),  (993,'image985.png'),  (994,'image986.png'),  (995,'image987.png'),  (996,'image988.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (997,'image989.png'),  (998,'image990.png'),  (999,'image991.png'),  (1000,'image992.png'),  (1001,'image993.png'),  (1002,'image994.png'),  (1003,'image995.png'),  (1004,'image996.png'),  (1005,'image997.png'),  (1006,'image998.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (1007,'image999.png'),  (1008,'image1000.png'),  (1009,'image1001.png'),  (1010,'image1002.png'),  (1011,'image1003.png'),  (1012,'image1004.png'),  (1013,'image1005.png'),  (1014,'image1006.png'),  (1015,'image1007.png'),  (1016,'image1008.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (1017,'image1009.png'),  (1018,'image1010.png'),  (1019,'image1011.png'),  (1020,'image1012.png'),  (1021,'image1013.png'),  (1022,'image1014.png'),  (1023,'image1015.png'),  (1024,'image1016.png'),  (1025,'image1017.png'),  (1026,'image1018.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (1027,'image1019.png'),  (1028,'image1020.png'),  (1029,'image1021.png'),  (1030,'image1022.png'),  (1031,'image1023.png'),  (1032,'image1024.png'),  (1033,'image1025.png'),  (1034,'image1026.png'),  (1035,'image1027.png'),  (1036,'image1028.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (1037,'image1029.png'),  (1038,'image1030.png'),  (1039,'image1031.png'),  (1040,'image1032.png'),  (1041,'image1033.png'),  (1042,'image1034.png'),  (1043,'image1035.png'),  (1044,'image1036.png'),  (1045,'image1037.png'),  (1046,'image1038.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (1047,'image1039.png'),  (1048,'image1040.png'),  (1049,'image1041.png'),  (1050,'image1042.png'),  (1051,'image1043.png'),  (1052,'image1044.png'),  (1053,'image1045.png'),  (1054,'image1046.png'),  (1055,'image1047.png'),  (1056,'image1048.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (1057,'image1049.png'),  (1058,'image1050.png'),  (1059,'image1051.png'),  (1060,'image1052.png'),  (1061,'image1053.png'),  (1062,'image1054.png'),  (1063,'image1055.png'),  (1064,'image1056.png'),  (1065,'image1057.png'),  (1066,'image1058.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (1067,'image1059.png'),  (1068,'image1060.png'),  (1069,'image1061.png'),  (1070,'image1062.png'),  (1071,'image1063.png'),  (1072,'image1064.png'),  (1073,'image1065.png'),  (1074,'image1066.png'),  (1075,'image1067.png'),  (1076,'image1068.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (1077,'image1069.png'),  (1078,'image1070.png'),  (1079,'image1071.png'),  (1080,'image1072.png'),  (1081,'image1073.png'),  (1082,'image1074.png'),  (1083,'image1075.png'),  (1084,'image1076.png'),  (1085,'image1077.png'),  (1086,'image1078.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (1087,'image1079.png'),  (1088,'image1080.png'),  (1089,'image1081.png'),  (1090,'image1082.png'),  (1091,'image1083.png'),  (1092,'image1084.png'),  (1093,'image1085.png'),  (1094,'image1086.png'),  (1095,'image1087.png'),  (1096,'image1088.png');  **INSERT** **INTO** public.images\_cities (id\_city,image) **VALUES**  (1097,'image1089.png'),  (1098,'image1090.png');  **INSERT** **INTO** public.jury\_directions (id\_user\_jury,id\_direction) **VALUES**  (31,1),  (32,1),  (33,2),  (34,2),  (35,3),  (36,1),  (37,1),  (38,4),  (39,2),  (40,4);  **INSERT** **INTO** public.jury\_directions (id\_user\_jury,id\_direction) **VALUES**  (92866829,4),  (337548,2);  **INSERT** **INTO** public.moderators\_directions\_events (id\_user\_moderator,id\_direction,id\_type\_event) **VALUES**  (73640936,5,2);  **INSERT** **INTO** public.roles ("name") **VALUES**  ('Участник'),  ('Модератор'),  ('Организатор'),  ('Жюри');  **INSERT** **INTO** public.schedule\_activeties\_in\_events (id\_activity,time\_start,"day",id\_schedule\_event,id\_moderator\_user) **VALUES**  (1,'09:00:00',1,1,41),  (2,'10:45:00',2,1,42),  (3,'12:30:00',2,1,43),  (4,'09:00:00',1,2,44),  (5,'10:45:00',2,2,45),  (6,'12:30:00',2,2,46),  (7,'09:00:00',1,3,47),  (8,'10:45:00',2,3,48),  (9,'12:30:00',2,3,49),  (10,'09:00:00',1,4,50);  **INSERT** **INTO** public.schedule\_activeties\_in\_events (id\_activity,time\_start,"day",id\_schedule\_event,id\_moderator\_user) **VALUES**  (11,'10:45:00',1,4,51),  (12,'12:30:00',2,4,52),  (13,'09:00:00',1,5,53),  (14,'10:45:00',2,5,54),  (15,'12:30:00',3,5,55),  (16,'09:00:00',1,6,56),  (17,'10:45:00',1,6,57),  (18,'12:30:00',1,6,58),  (19,'09:00:00',1,7,59),  (20,'10:45:00',1,7,60);  **INSERT** **INTO** public.schedule\_activeties\_in\_events (id\_activity,time\_start,"day",id\_schedule\_event,id\_moderator\_user) **VALUES**  (21,'12:30:00',2,7,61),  (22,'09:00:00',1,8,42),  (23,'10:45:00',1,8,43),  (24,'16:00:00',1,8,44),  (25,'09:00:00',1,9,48),  (26,'10:45:00',2,9,49),  (27,'12:30:00',2,9,50),  (28,'09:00:00',1,10,52),  (29,'10:45:00',1,10,53),  (30,'12:30:00',2,10,54);  **INSERT** **INTO** public.schedule\_activeties\_jury (id\_jury\_user,id\_schedule\_activity) **VALUES**  (31,1),  (37,1),  (38,1),  (32,1),  (32,1),  (38,2),  (39,2),  (33,2),  (38,2),  (33,2);  **INSERT** **INTO** public.schedule\_activeties\_jury (id\_jury\_user,id\_schedule\_activity) **VALUES**  (39,3),  (40,3),  (34,3),  (39,3),  (35,3),  (31,4),  (37,4),  (38,4),  (32,4),  (36,4);  **INSERT** **INTO** public.schedule\_activeties\_jury (id\_jury\_user,id\_schedule\_activity) **VALUES**  (32,5),  (38,5),  (39,5),  (33,5),  (37,5),  (33,6),  (39,6),  (40,6),  (34,6),  (35,6);  **INSERT** **INTO** public.schedule\_activeties\_jury (id\_jury\_user,id\_schedule\_activity) **VALUES**  (32,7),  (31,7),  (37,7),  (38,7),  (36,7),  (33,8),  (32,8),  (38,8),  (39,8),  (37,8);  **INSERT** **INTO** public.schedule\_activeties\_jury (id\_jury\_user,id\_schedule\_activity) **VALUES**  (34,9),  (33,9),  (39,9),  (40,9),  (32,9),  (35,10),  (38,10),  (31,10),  (37,10),  (33,10);  **INSERT** **INTO** public.schedule\_activeties\_jury (id\_jury\_user,id\_schedule\_activity) **VALUES**  (36,11),  (39,11),  (32,11),  (38,11),  (34,11),  (37,12),  (40,12),  (33,12),  (39,12),  (35,12);  **INSERT** **INTO** public.schedule\_activeties\_jury (id\_jury\_user,id\_schedule\_activity) **VALUES**  (38,13),  (32,13),  (37,13),  (31,13),  (36,13),  (39,14),  (33,14),  (38,14),  (32,14),  (37,14);  **INSERT** **INTO** public.schedule\_activeties\_jury (id\_jury\_user,id\_schedule\_activity) **VALUES**  (40,15),  (34,15),  (39,15),  (33,15),  (38,15),  (35,16),  (37,16),  (34,16),  (32,16),  (39,16);  **INSERT** **INTO** public.schedule\_activeties\_jury (id\_jury\_user,id\_schedule\_activity) **VALUES**  (36,17),  (38,17),  (35,17),  (33,17),  (40,17),  (37,18),  (39,18),  (36,18),  (34,18),  (35,18);  **INSERT** **INTO** public.schedule\_activeties\_jury (id\_jury\_user,id\_schedule\_activity) **VALUES**  (37,19),  (34,19),  (32,19),  (38,19),  (36,19),  (38,20),  (35,20),  (33,20),  (39,20),  (37,20);  **INSERT** **INTO** public.schedule\_activeties\_jury (id\_jury\_user,id\_schedule\_activity) **VALUES**  (39,21),  (36,21),  (34,21),  (37,21),  (34,22),  (38,22),  (35,22),  (32,22),  (38,22),  (35,23);  **INSERT** **INTO** public.schedule\_activeties\_jury (id\_jury\_user,id\_schedule\_activity) **VALUES**  (39,23),  (36,23),  (33,23),  (39,23),  (36,24),  (40,24),  (37,24),  (34,24),  (34,24),  (37,25);  **INSERT** **INTO** public.schedule\_activeties\_jury (id\_jury\_user,id\_schedule\_activity) **VALUES**  (32,25),  (38,25),  (38,25),  (35,25),  (38,26),  (33,26),  (39,26),  (39,26),  (36,27),  (39,27);  **INSERT** **INTO** public.schedule\_activeties\_jury (id\_jury\_user,id\_schedule\_activity) **VALUES**  (34,27),  (40,27),  (37,27),  (34,28),  (32,28),  (35,28),  (38,28),  (38,28),  (35,29),  (33,29);  **INSERT** **INTO** public.schedule\_activeties\_jury (id\_jury\_user,id\_schedule\_activity) **VALUES**  (36,29),  (39,29),  (39,30),  (36,30),  (34,30),  (37,30),  (40,30);  **INSERT** **INTO** public.schedule\_events (id\_event,"day",id\_winner\_user) **VALUES**  (1,2,1),  (4,2,3),  (5,3,5),  (6,2,7),  (7,3,8),  (11,2,10),  (14,2,9),  (16,1,13),  (18,3,15),  (20,3,17);  **INSERT** **INTO** public.types\_events ("name") **VALUES**  ('IT в бизнесе'),  ('Разработка приложений'),  ('IT-инфраструктура'),  ('Стартапы'),  ('Информационная безопасность');  **INSERT** **INTO** public.users ("password",email,birthday,phone,photo,id\_gender,id\_country,id\_role,full\_name,image) **VALUES**  ('TImO7k','m73dobuzn6jc@gmail.com','1980-02-11','7(821)611-43-32','foto13.jpg',1,71,1,'Трофимов Любомир Русланович','foto2.jpg'),  ('PtwSEA','9mtkjdsrhp2c@yahoo.com','1989-01-14','7(821)779-78-75','foto14.jpg',1,91,1,'Симонов Остап Федотович','foto3.jpg'),  ('EzwyiP','wxey0gmcjuz2@mail.com','1994-07-30','7(821)819-98-39','foto15.jpg',1,81,1,'Захаров Арнольд Германнович','foto4.jpg'),  ('pEVpH3','lqih53fw9ryx@tutanota.com','1973-01-15','7(821)177-06-74','foto16.jpg',1,18,1,'Фокин Клемент Игнатьевич','foto5.jpg'),  ('eEJ83V','p473g2jlcmhi@outlook.com','1981-06-24','7(821)856-28-93','foto17.jpg',1,80,1,'Захаров Аверкий Альбертович','foto6.jpg'),  ('0Utzck','t1dg96ikvorc@tutanota.com','1977-06-26','7(821)623-59-80','foto18.jpg',1,45,1,'Егоров Афанасий Тарасович','foto7.jpg'),  ('HLAFQB','ah91upwo7xfl@mail.com','1972-12-04','7(821)007-84-60','foto19.jpg',1,66,1,'Ермаков Клемент Проклович','foto8.jpg'),  ('nrBfHG','fskhuw2oxgev@outlook.com','1996-05-11','7(821)376-94-02','foto20.jpg',1,87,1,'Лазарев Марк Юлианович','foto9.jpg'),  ('xM5QB5','6opkrq9n87d1@outlook.com','1989-08-06','7(821)685-48-91','foto21.jpg',1,45,1,'Петров Геннадий Даниилович','foto10.jpg'),  ('v5MxkG','epflsy9iobdx@tutanota.com','1989-04-26','7(821)149-13-09','foto22.jpg',1,86,1,'Субботин Мартин Пантелеймонович','foto11.jpg');  **INSERT** **INTO** public.users ("password",email,birthday,phone,photo,id\_gender,id\_country,id\_role,full\_name,image) **VALUES**  ('8MkKg6','zy8tom2bxnuk@mail.com','1995-11-09','7(821)162-52-21','foto23.jpg',1,45,1,'Рожков Демьян Эдуардович','foto12.jpg'),  ('GYlXDR','i4ex5whc7dqk@gmail.com','1987-01-29','7(821)749-53-09','foto25.jpg',1,44,1,'Петухов Алан Пётрович','foto14.jpg'),  ('10PHSE','qsnyrwodje0k@outlook.com','1993-12-08','7(821)496-70-50','foto26.jpg',1,20,1,'Бобров Марк Юрьевич','foto15.jpg'),  ('uNRgsg','tacixb04vh5g@gmail.com','1999-05-18','7(821)137-45-80','foto27.jpg',1,73,1,'Александров Варлаам Робертович','foto16.jpg'),  ('9YNtvb','96o7iwszty5n@tutanota.com','1975-11-30','7(821)579-45-88','foto28.jpg',1,81,1,'Максимов Егор Дамирович','foto17.jpg'),  ('4rSPmK','yn6pfe4kasbo@outlook.com','1996-11-15','7(821)277-58-70','foto29.jpg',1,13,1,'Комиссаров Антон Протасьевич','foto18.jpg'),  ('STSVII','vny26gsmxu9k@tutanota.com','1975-12-30','7(821)760-71-19','foto30.jpg',1,61,1,'Мамонтов Прохор Созонович','foto19.jpg'),  ('DsEhqN','mwlny4zqtc65@outlook.com','1988-07-15','7(821)478-05-95','foto31.jpg',1,22,1,'Баранов Венедикт Ефимович','foto20.jpg'),  ('JyP2M2ji63','3swfq0n756y1@mail.com','1989-01-05','7(329)246-20-13','foto1.jpg',1,1,3,'Скворцов Михаил Артёмович','foto21.jpg'),  ('8fb3RJT8c6','wukm6dacf7v0@gmail.com','1980-09-01','7(226)494-15-47','foto2.jpg',2,87,3,'Романова Анастасия Тимофеевна','foto22.jpg');  **INSERT** **INTO** public.users ("password",email,birthday,phone,photo,id\_gender,id\_country,id\_role,full\_name,image) **VALUES**  ('2T2xCYef86','5ku2w7nqzvot@gmail.com','1981-03-02','7(725)164-24-56','foto3.jpg',1,10,3,'Белкин Дмитрий Александрович','foto23.jpg'),  ('nG7xGVi892','8s54jayek2mt@outlook.com','1981-09-12','7(388)191-50-31','foto4.jpg',2,63,3,'Лаврова Анастасия Маратовна','foto24.jpg'),  ('Ka74TC2r7m','dlm7wocnzhtp@mail.com','1983-07-14','7(581)960-76-73','foto5.jpg',2,69,3,'Громова Виктория Матвеевна','foto25.jpg'),  ('9nM7A9Mtv5','70rkvgtfjswm@gmail.com','1995-11-10','7(466)705-98-66','foto7.jpg',1,59,3,'Прохоров Максим Серафимович','foto27.jpg'),  ('d6UAu83Sd8','vy3ajxkge8p7@mail.com','1984-11-20','7(640)704-75-62','foto8.jpg',1,28,3,'Мешков Даниил Николаевич','foto28.jpg'),  ('JS4K8pr54u','tkgduj6na1hm@gmail.com','1989-01-14','7(500)841-51-43','foto9.jpg',2,57,3,'Рябова Алиса Викторовна','foto29.jpg'),  ('42xF46LVkh','uw846cn27x9k@outlook.com','1981-12-21','7(053)696-95-19','foto10.jpg',1,63,3,'Соловьев Демид Артёмович','foto30.jpg'),  ('ugWkzE','oconnell.steve@feest.com','1978-02-03','7(186)808-50-25','foto11.jpg',1,82,4,'Одинцов Дмитрий Лаврентьевич','foto31.jpg'),  ('TsCX7X','olson.shanny@gmail.com','1972-09-15','7(215)064-59-70','foto12.jpg',2,43,4,'Белова Инга Прокловна','foto32.jpg'),  ('R2buBG','whirthe@beer.info','1950-02-21','7(810)322-94-05','foto13.jpg',2,36,4,'Соловьёва Аюна Станиславовна','foto33.jpg');  **INSERT** **INTO** public.users ("password",email,birthday,phone,photo,id\_gender,id\_country,id\_role,full\_name,image) **VALUES**  ('6Kuk9B','rerdman@gmail.com','1950-07-20','7(963)659-08-16','foto14.jpg',1,75,4,'Зайцев Иван Артемович','foto34.jpg'),  ('XQZbTX','pkutch@hotmail.com','1950-08-19','7(138)268-54-96','foto15.jpg',2,89,4,'Некрасова Лаура Богдановна','foto35.jpg'),  ('AA6JS6','trace.lindgren@beahan.com','1963-03-06','7(276)229-95-45','foto16.jpg',1,24,4,'Брагин Осип Владиславович','foto36.jpg'),  ('j7WWL5','jadon85@gmail.com','1973-03-07','7(346)523-76-14','foto17.jpg',1,64,4,'Игнатьев Мирослав Тарасович','foto37.jpg'),  ('2QCbSj','mathilde77@yahoo.com','1952-09-11','7(742)194-06-10','foto18.jpg',2,75,4,'Матвеева Вера Митрофановна','foto38.jpg'),  ('SqJHTL','cleveland.hamill@gmail.com','1994-09-26','7(097)858-38-14','foto20.jpg',1,8,4,'Фомичёв Варлаам Дмитрьевич','foto40.jpg'),  ('4VUmGj4t36','i8brhz2gyx4j@gmail.com','1974-01-22','7(416)870-22-00','foto9.jpg',1,73,2,'Покровский Роман Дмитриевич','foto41.jpg'),  ('b9zAs99XD3','dic8qah9zbot@mail.com','1957-06-12','7(278)138-47-92','foto10.jpg',1,47,2,'Чернышев Ярослав Андреевич','foto42.jpg'),  ('n6cx69AM4T','o3gmly907wcn@mail.com','1960-05-29','7(093)787-84-57','foto8.jpg',2,21,2,'Суворова Ева Алексеевна','foto43.jpg'),  ('P8c83u5fUR','qz901omryb7j@mail.com','1996-08-11','7(699)704-90-18','foto11.jpg',2,96,2,'Лосева Аделина Георгиевна','foto44.jpg');  **INSERT** **INTO** public.users ("password",email,birthday,phone,photo,id\_gender,id\_country,id\_role,full\_name,image) **VALUES**  ('f7S5zxH58B','m93nsaih4kl8@yahoo.com','1968-09-23','7(312)920-22-96','foto12.jpg',1,79,2,'Покровский Марк Максимович','foto45.jpg'),  ('c2vGi532VN','qgb9ea8wnl50@mail.com','1989-01-20','7(714)693-32-92','foto13.jpg',2,72,2,'Акимова София Александровна','foto46.jpg'),  ('8jZ5LL2a2n','rzsj6wqd42vt@yahoo.com','1998-05-01','7(605)246-65-83','foto14.jpg',1,63,2,'Майоров Александр Михайлович','foto47.jpg'),  ('D44jjR45gH','wg9h3ixavl25@tutanota.com','1982-05-24','7(037)333-08-17','foto16.jpg',2,72,2,'Прохорова Анна Фёдоровна','foto48.jpg'),  ('2c73vxTLP9','2nib6c5vl18k@outlook.com','1969-03-28','7(594)615-77-80','foto17.jpg',1,64,2,'Соловьев Иван Дмитриевич','foto49.jpg'),  ('nOEBrK','4sqplurb5eki@gmail.com','1991-02-08','7(821)277-59-90','foto12.jpg',1,68,1,'Корнилов Владимир Степанович','foto1.jpg'),  ('COZkSO','kd64ino5fcx3@outlook.com','1982-10-17','7(821)976-48-88','foto24.jpg',1,93,1,'Емельянов Анатолий Авксентьевич','foto13.jpg'),  ('fk9BHN2g96','ekibvrcm71a6@gmail.com','1980-08-06','7(959)240-88-47','foto6.jpg',1,78,3,'Карпов Алексей Артёмович','foto26.jpg'),  ('kFwax6','graham.robb@boyer.com','1980-09-24','7(089)418-02-33','foto19.jpg',2,75,4,'Пестова Ева Альбертовна','foto39.jpg'),  ('69S6fMzeM2','wbs3znh5uxfr@tutanota.com','1980-10-30','7(596)301-43-97','foto18.jpg',2,48,2,'Лебедева Виктория Марковна','foto50.jpg');  **INSERT** **INTO** public.users ("password",email,birthday,phone,photo,id\_gender,id\_country,id\_role,full\_name,image) **VALUES**  ('j7TMm92s9X','yl9hxt5dzajv@tutanota.com','1998-02-08','7(181)588-68-39','foto19.jpg',2,46,2,'Шишкина Светлана Александровна','foto51.jpg'),  ('XU5b8N42th','4rqfa385p9gz@mail.com','1978-01-27','7(442)333-89-12','foto20.jpg',2,76,2,'Юдина Татьяна Максимовна','foto52.jpg'),  ('6R82k4nFnX','ox7k5w6l04mi@gmail.com','1953-05-31','7(377)251-56-96','foto21.jpg',1,24,2,'Рябинин Григорий Матвеевич','foto53.jpg'),  ('eC445cE8Yd','a8071jvd2m6z@tutanota.com','1975-12-24','7(575)142-09-75','foto22.jpg',2,79,2,'Тарасова Валерия Егоровна','foto54.jpg'),  ('8T69Vef8Er','qxel0og2ps4t@gmail.com','1976-05-12','7(979)385-40-57','foto23.jpg',1,6,2,'Алексеев Михаил Глебович','foto55.jpg'),  ('34nYAv56Cs','ixut6e0cnd84@gmail.com','1973-05-09','7(901)002-16-80','foto24.jpg',2,72,2,'Иванова Виктория Павловна','foto56.jpg'),  ('ru6x7PT2V4','es06joah3pfu@gmail.com','1962-08-26','7(272)350-20-30','foto25.jpg',2,98,2,'Богданова Олеся Евгеньевна','foto57.jpg'),  ('a7pXZ78a2J','0kmd26jfi859@mail.com','1988-11-23','7(812)026-33-34','foto26.jpg',1,39,2,'Иванов Фёдор Тимофеевич','foto58.jpg'),  ('8uTv4L8Cg4','mn7bawsorg51@outlook.com','1962-05-05','7(550)677-95-09','foto27.jpg',1,13,2,'Кузнецов Семён Владиславович','foto59.jpg'),  ('6VM3r9jmS5','1os2bmtpg6nv@mail.com','1989-01-13','7(596)221-45-84','foto28.jpg',2,20,2,'Шульгина Елизавета Денисовна','foto60.jpg');  **INSERT** **INTO** public.users ("password",email,birthday,phone,photo,id\_gender,id\_country,id\_role,full\_name,image) **VALUES**  ('c6m4L2ZD9j','vbpjslo28w6d@mail.com','1987-04-25','7(588)448-48-41','foto29.jpg',2,7,2,'Петрова Василиса Георгиевна','foto61.jpg'),  ('1',**NULL**,'-infinity','+7(111)-111-11-11',**NULL**,1,**NULL**,4,'Иванов Иван Иванович','foto18.jpg'),  ('333',**NULL**,**NULL**,'+7(333)-334-23-14',**NULL**,1,**NULL**,4,'Михайлов Михаил Михайлович','foto44.jpg'),  ('444',**NULL**,**NULL**,'+7(323)-232-33-23',**NULL**,1,**NULL**,2,'Никилоаев Николай Николаевич','foto57.jpg'),  ('Pds#@12','Ivan@gmail.com',**NULL**,'+7(111)-111-11-11',**NULL**,1,**NULL**,4,'Иванович Иван Иванов','foto1.jpg'); |

# Листинг 2 - Код функций библиотек

|  |
| --- |
| namespace REG\_MARK\_LIB  {  public class REG\_MARK\_LIB\_CLASS  {  List<char> \_correctSymbols = new List<char>() { 'a', 'b', 'e', 'k', 'm', 'h', 'o', 'p', 'c', 't', 'y', 'x'};  List<int> \_correctNumRegion = new List<int>() { 102, 113, 116, 716, 121, 123, 124, 125, 134, 150, 190, 750, 790, 152, 154, 159, 161, 163, 164, 196, 173, 174, 177, 197, 199, 777, 799, 186 };  public bool CheckMark(string mark)  {  switch (\_correctSymbols.Contains(mark[0]) && \_correctSymbols.Contains(mark[4]) && \_correctSymbols.Contains(mark[5]))  {  case false:  return false;  }  int numRegion = 0;  switch (mark.Length)  {  case 9:  numRegion = Convert.ToInt32(mark[6..9]);  break;  case 8:  numRegion = Convert.ToInt32(mark[6..8]);  break;  }  if (0 < numRegion && numRegion < 100)  {  return true;  }  if (\_correctNumRegion.Contains(numRegion))  {  return true;  }  return false;  }  public string GetNextMarkAfter(string mark)  {  string resultNum = string.Copy(mark);  int numberMark = Convert.ToInt32(mark[1..4]);  if (numberMark != 999)  {  string numberMarkStr = (++numberMark).ToString();  switch (numberMarkStr.Length)  {  case 1:  numberMarkStr = String.Concat("00", numberMarkStr);  break;  case 2:  numberMarkStr = String.Concat("0", numberMarkStr);  break;  }  resultNum = resultNum.Remove(1, 1).Insert(1, numberMarkStr[0].ToString());  resultNum = resultNum.Remove(2, 1).Insert(2, numberMarkStr[1].ToString());  resultNum = resultNum.Remove(3, 1).Insert(3, numberMarkStr[2].ToString());  return resultNum;  }  else  {  resultNum = resultNum.Remove(1, 1).Insert(1, "0");  resultNum = resultNum.Remove(2, 1).Insert(2, "0");  resultNum = resultNum.Remove(3, 1).Insert(3, "0");  int countEndSymbol = resultNum.Count(x => x == 'x');  if (countEndSymbol == 3)  {  resultNum = resultNum.Remove(0, 1).Insert(0, "a");  resultNum = resultNum.Remove(4, 1).Insert(4, "a");  resultNum = resultNum.Remove(5, 1).Insert(5, "a");  return resultNum;  }  else if (countEndSymbol == 2)  {  int indexFirstLastSymbol = resultNum.IndexOf('x');  int indexSecondLastSymbol = resultNum.LastIndexOf('x');  if (indexFirstLastSymbol == 0 && indexSecondLastSymbol == 4)  {  int indexInList = \_correctSymbols.IndexOf(resultNum[5]);  resultNum = resultNum.Remove(5, 1).Insert(5, \_correctSymbols[indexInList++].ToString());  return resultNum;  }  else if (indexFirstLastSymbol == 0 && indexSecondLastSymbol == 5)  {  int indexInList = \_correctSymbols.IndexOf(resultNum[4]);  resultNum = resultNum.Remove(4, 1).Insert(4, \_correctSymbols[indexInList++].ToString());  return resultNum;  }  else  {  int indexInList = \_correctSymbols.IndexOf(resultNum[0]);  resultNum = resultNum.Remove(0, 1).Insert(0, \_correctSymbols[indexInList++].ToString());  return resultNum;  }  }  else  {  switch(resultNum.IndexOf('x'))  {  case 0:  int indexInList = \_correctSymbols.IndexOf(resultNum[5]);  resultNum = resultNum.Remove(5, 1).Insert(5, \_correctSymbols[indexInList++].ToString());  return resultNum;  case 4:  indexInList = \_correctSymbols.IndexOf(resultNum[5]);  resultNum = resultNum.Remove(5, 1).Insert(5, \_correctSymbols[indexInList++].ToString());  return resultNum;  case 5:  indexInList = \_correctSymbols.IndexOf(resultNum[4]);  resultNum = resultNum.Remove(4, 1).Insert(4, \_correctSymbols[indexInList++].ToString());  resultNum = resultNum.Remove(5, 1).Insert(5, "a");  return resultNum;  }  }  }  return resultNum;  }  public string GetNextMarkAfterInRange(string prevMark, string rangeStart, string rangeEnd)  {  int indexFirstSymbolRangeStart = \_correctSymbols.IndexOf(rangeStart[0]);  int indexFirstSymbolRangeEnd = \_correctSymbols.IndexOf(rangeEnd[0]);  int indexFirstSymbolPrevMark = \_correctSymbols.IndexOf(prevMark[0]);  if (indexFirstSymbolRangeStart < indexFirstSymbolPrevMark && indexFirstSymbolPrevMark < indexFirstSymbolRangeEnd)  {  return GetNextMarkAfter(prevMark);  }  else if (indexFirstSymbolRangeStart == indexFirstSymbolRangeEnd && indexFirstSymbolRangeStart == indexFirstSymbolPrevMark)  {  int indexSecondSymbolRangeStart = \_correctSymbols.IndexOf(rangeStart[4]);  int indexSecondSymbolRangeEnd = \_correctSymbols.IndexOf(rangeEnd[4]);  int indexSecondSymbolPrevMark = \_correctSymbols.IndexOf(prevMark[4]);  if (indexSecondSymbolRangeStart < indexSecondSymbolPrevMark && indexSecondSymbolPrevMark < indexSecondSymbolRangeEnd)  {  return GetNextMarkAfter(prevMark);  }  else if (indexSecondSymbolRangeStart == indexSecondSymbolRangeEnd && indexSecondSymbolRangeStart == indexSecondSymbolPrevMark)  {  int indexThirdSymbolRangeStart = \_correctSymbols.IndexOf(rangeStart[5]);  int indexThirdSymbolRangeEnd = \_correctSymbols.IndexOf(rangeEnd[5]);  int indexThirdSymbolPrevMark = \_correctSymbols.IndexOf(prevMark[5]);  if (indexThirdSymbolRangeStart < indexThirdSymbolPrevMark && indexThirdSymbolPrevMark < indexThirdSymbolRangeEnd)  {  return GetNextMarkAfter(prevMark);  }  else if (indexThirdSymbolRangeStart == indexThirdSymbolRangeEnd && indexThirdSymbolRangeStart == indexThirdSymbolPrevMark)  {  int numberRangeStart = Convert.ToInt32(rangeStart[1..3]);  int numberRangeEnd = Convert.ToInt32(rangeEnd[1..3]);  int numberPevMark = Convert.ToInt32(prevMark[1..3]);  if (numberRangeStart < numberPevMark && numberPevMark < numberRangeEnd)  {  return GetNextMarkAfter(prevMark);  }  else return "out of stock";  }  else  {  return "out of stock";  }  }  else  {  return "out of stock";  }  }  else  {  return "out of stock";  }  }  int CountValueInMaxAndMin(int indexFirst, int indexSecond)  {  int maxValue = Math.Max(indexFirst, indexSecond);  int minValue = Math.Min(indexFirst, indexSecond);  int countNumber = maxValue - minValue;  return countNumber;  }  public int GetCombinationsCountInRange(string mark1, string mark2)  {  int indexFirstSymbolMarkOne = \_correctSymbols.IndexOf(mark1[0]);  int indexFirstSymbolMarkTwo = \_correctSymbols.IndexOf(mark2[0]);  switch (indexFirstSymbolMarkOne == indexFirstSymbolMarkTwo)  {  case true:  int numberMarkOne = Convert.ToInt32(mark1[1..4]);  int numberMarkTwo = Convert.ToInt32(mark2[1..4]);  int countNumber = CountValueInMaxAndMin(numberMarkOne, numberMarkTwo);  int countValueSecondSymbolInMark = CountValueInMaxAndMin(\_correctSymbols.IndexOf(mark1[4]), \_correctSymbols.IndexOf(mark2[4])) + 1;  int countValueThirdSymbolInMark = CountValueInMaxAndMin(\_correctSymbols.IndexOf(mark1[5]), \_correctSymbols.IndexOf(mark2[5])) + 1;  int result = countNumber \* countValueSecondSymbolInMark \* countValueThirdSymbolInMark;  return result;  case false:  numberMarkOne = Convert.ToInt32(mark1[1..4]);  numberMarkTwo = Convert.ToInt32(mark2[1..4]);  countNumber = 999 - Math.Min(numberMarkOne, numberMarkTwo) + Math.Max(numberMarkOne, numberMarkTwo);  int countValueFirstSymbolInMark = CountValueInMaxAndMin(\_correctSymbols.IndexOf(mark1[0]), \_correctSymbols.IndexOf(mark2[0])) + 1;  countValueSecondSymbolInMark = CountValueInMaxAndMin(\_correctSymbols.IndexOf(mark1[4]), \_correctSymbols.IndexOf(mark2[4])) + 1;  countValueThirdSymbolInMark = CountValueInMaxAndMin(\_correctSymbols.IndexOf(mark1[5]), \_correctSymbols.IndexOf(mark2[5])) + 1;  result = countNumber \* countValueSecondSymbolInMark \* countValueThirdSymbolInMark \* countValueFirstSymbolInMark;  return result;  }  }  }  } |