

**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ**  
**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ**  
**«СИНЕРГИЯ»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт:** |  | Информационных технологий |
|  |  | (наименование факультета/ института) |
| **Направление/специальность:** |  | Разработка, сопровождение и обеспечение безопасности информационных систем |
|  |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | Очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Лабораторный практикум №9:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **На тему:** |  | Администрирование информационных систем. Администрирование интранет-сервера IIS, | | |
|  | | |  | (наименование темы) |
|  |  | корпоративный сайт и виртуальный каталог. | | |
| **По дисциплине:** | |  | Администрирование информационных систем | |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Обучающийся****:** |  | |  | | --- | | Рамазанов Эмиль Рустамович | |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа:** |  | ВБИо-301рсоб |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель:** |  | Сибирев Иван Валерьевич |  |  |
|  |  | (ФИО) |  | (подпись) |

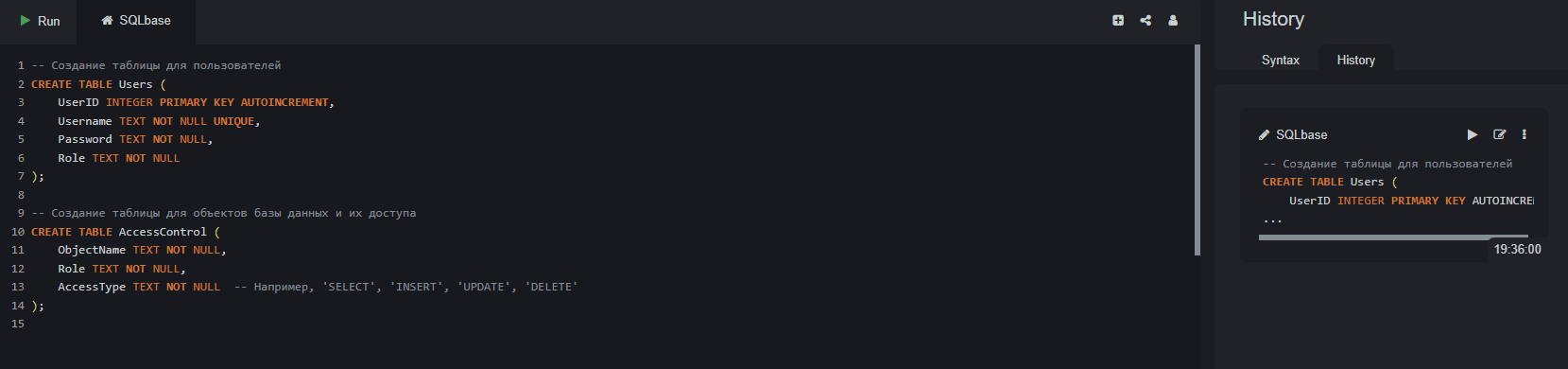
# **Москва 2024 г.**

Вместо создания ролей и пользователей, можно использовать простую структуру для управления доступом на уровне приложения.

Тем не менее, можно создать таблицы для хранения информации о пользователях и их доступах. Максимально универсальное и доступное решение, которое можно использовать распишу ниже.

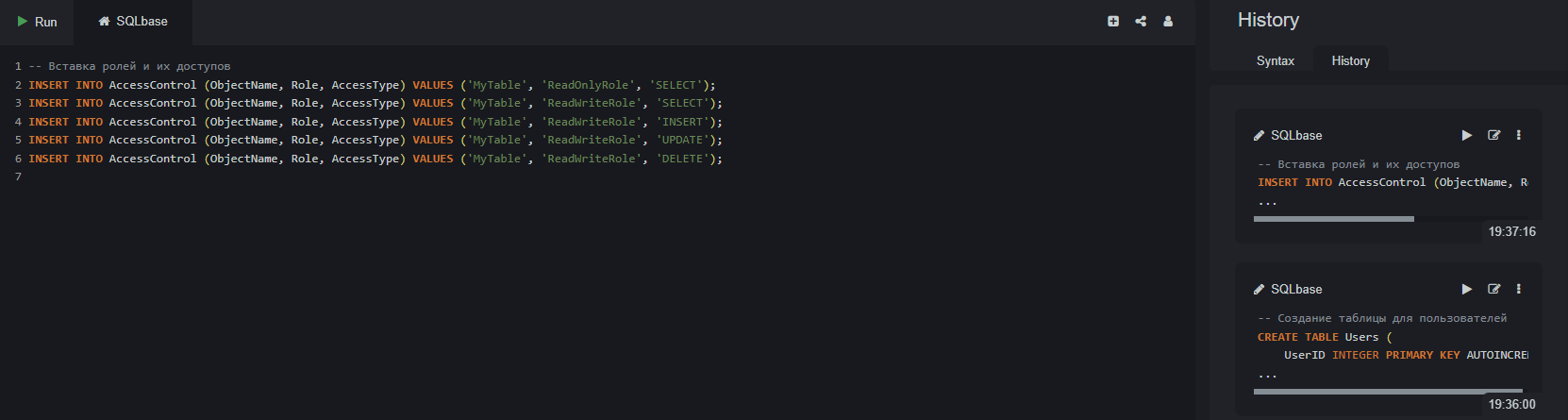
**1. Создание таблиц для хранения пользователей и их прав доступа**

|  |
| --- |
| -- Создание таблицы для пользователей  CREATE TABLE Users (  UserID INTEGER PRIMARY KEY AUTOINCREMENT,  Username TEXT NOT NULL UNIQUE,  Password TEXT NOT NULL,  Role TEXT NOT NULL  );  -- Создание таблицы для объектов базы данных и их доступа  CREATE TABLE AccessControl (  ObjectName TEXT NOT NULL,  Role TEXT NOT NULL,  AccessType TEXT NOT NULL -- Например, 'SELECT', 'INSERT', 'UPDATE', 'DELETE'  ); |



**2. Вставка данных о ролях и доступах**

|  |
| --- |
| -- Вставка ролей и их доступов  INSERT INTO AccessControl (ObjectName, Role, AccessType) VALUES ('MyTable', 'ReadOnlyRole', 'SELECT');  INSERT INTO AccessControl (ObjectName, Role, AccessType) VALUES ('MyTable', 'ReadWriteRole', 'SELECT');  INSERT INTO AccessControl (ObjectName, Role, AccessType) VALUES ('MyTable', 'ReadWriteRole', 'INSERT');  INSERT INTO AccessControl (ObjectName, Role, AccessType) VALUES ('MyTable', 'ReadWriteRole', 'UPDATE');  INSERT INTO AccessControl (ObjectName, Role, AccessType) VALUES ('MyTable', 'ReadWriteRole', 'DELETE'); |



**3. Определение объектов базы данных только для чтения (SELECT)**

|  |
| --- |
| -- Получение объектов, доступных для чтения  SELECT ObjectName  FROM AccessControl  WHERE AccessType = 'SELECT'; |



**4. Определение изменяемых объектов базы данных (INSERT, UPDATE, DELETE)**

|  |
| --- |
| -- Получение объектов, доступных для записи  SELECT ObjectName  FROM AccessControl  WHERE AccessType IN ('INSERT', 'UPDATE', 'DELETE'); |



**5. Создание пользователей и назначение их ролям**

|  |
| --- |
| -- Создание пользователей  INSERT INTO Users (Username, Password, Role) VALUES ('User 1', 'Password1', 'ReadOnlyRole');  INSERT INTO Users (Username, Password, Role) VALUES ('User 2', 'Password2', 'ReadWriteRole'); |

**Вывод:**

* В этом подходе мы вручную управляем доступом к данным через таблицы.
* В реальном приложении важно реализовать логику проверки прав доступа на уровне приложения. Лучше не использовать SQLite, потому что он не поддерживает управление доступом на уровне базы данных.
* Важно убедиться, что пароли хранятся безопасно, используя хеширование.