**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (ДГТУ)

Факультет «Информатика и вычислительная техника»

Кафедра «Кибербезопасность информационных систем»

**Лабораторная работа № 6**

на тему «разграничение доступа в СУБД PostgreSQL»

|  |
| --- |
| Выполнил: студент группы ВКБ43 |
| Ковалев Данил Петрович |
| (Фамилия, имя, отчество) |
| Проверил: |
|  |
| (Фамилия, имя, отчество) |

Оглавление

[Цель 3](#_Toc179639729)

# **Цель:** получить практические навыки в разграничении доступа при работе с базами данных на примере СУБД PostgreSQL.

**Ход работы:**

**Задание 1**. Создать несколько пользователей, продемонстрировать назначение им различных привилегий (на создание объектов БД, на их просмотр и изменение).

Для создания пользователей я использовал скрипт, который представлен ниже на рисунке 1.

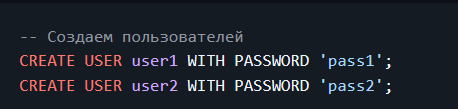


Рисунок 1 - создание пользователей user1 и user2

Предоставим теперь права для пользователя user1. Здесь мы добавляем возможность создавать объекты БД, выдаю доступ к schema, выдаю привелегии на чтение и запись всех таблиц в схеме public. Код представлен ниже на рисунке 2.

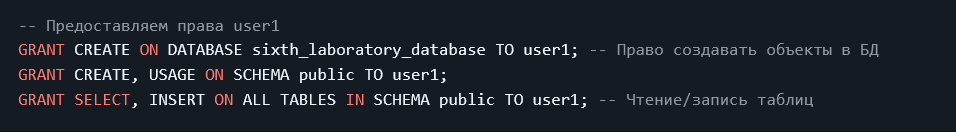


Рисунок 2 - Выдача прав на взаимодействие с БД для user1

Предоставим теперь права для пользователя user2. Здесь мы добавляем возможность права только на подключение к базе данных, доступ к чтению данных из схемы public и выполнять операцию SELECT на все таблицы базы данных. Код представлен на рисунке 3.

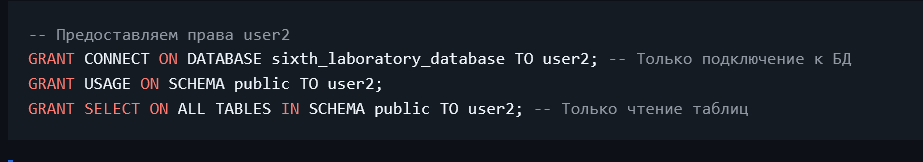


Рисунок 3 - Выдача прав на взаимодействие с БД для user2

**Задание 2**. Создать от имени одного из пользователей таблицы, реализовать на их базе представление и предоставить права на просмотр и изменение данных в представлении другому пользователю. Объяснить результаты.

Создадим теперь таблицы от имени пользователя user1, используя команды, которые представлены на рисунке 4.

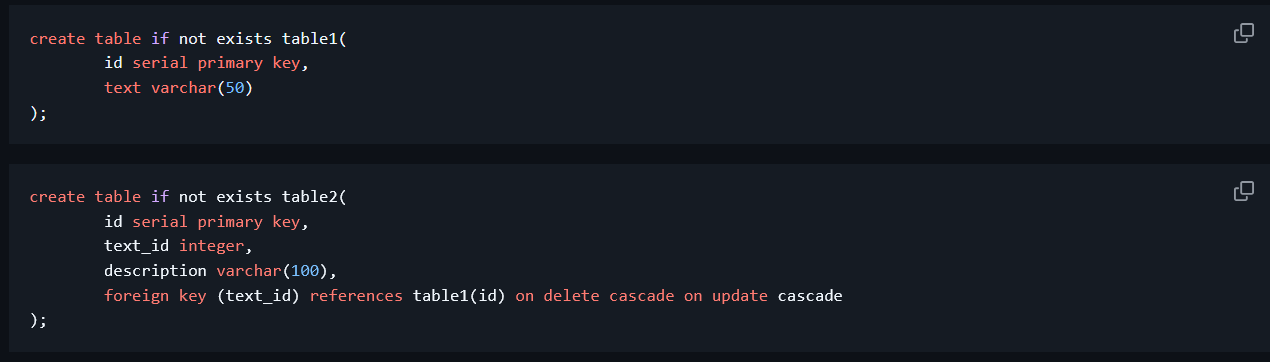


Рисунок 4 – создание таблиц от лица пользователя user1

Заполним table1 данными, которые представлены на рисунке 5.



Рисунок 5 – заполнение таблицы table1

Проверим, что данные успешно добавлены в таблицу table1. Результат представлен на рисунке 6.

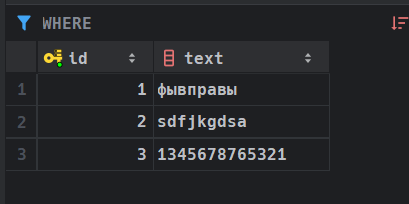


Рисунок 6 – успешность сохранения данных

Теперь заполним table2 данными, которые представлены на рисунке 7.

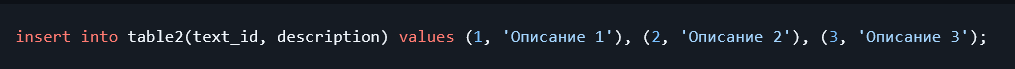


Рисунок 7 – заполнение table2

Проверим, что все данные сохранились, и они есть в базе данных. Результат представлен на рисунке 8.

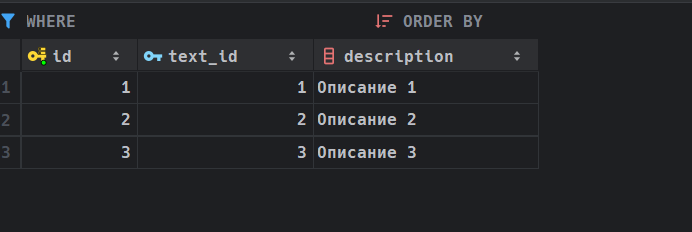


Рисунок 8 – успешность сохранения данных

Создадим теперь представление с названием “View”, которое будет просто отображать колонки “text”, “description”, соединяя таблицы через INNER JOIN. Код для создания представления представлен на рисунке 9.

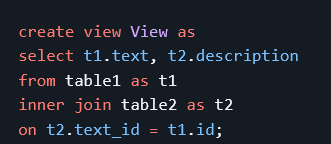


Рисунок 9 – создание представления

Попробуем просмотреть данные, которые получили из-под данного View, в результате вышло то, что представлено на рисунке 10.

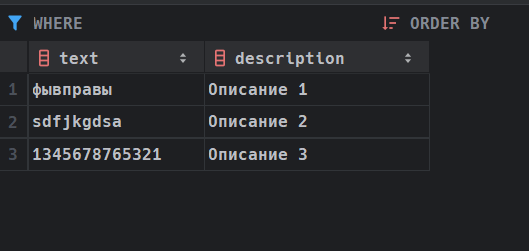


Рисунок 10 – просмотр содержимого View

Выдадим теперь доступ на выбор данных из представления для пользователя user2. Код представлен на рисунке 11.

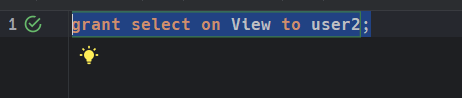


Рисунок 11 – выдача прав на выбор из представления для user2

Теперь попробуем проверить, что из-под пользователя “user2” у нас удачно все отображается в View. Результат представлен на рисунке 12.

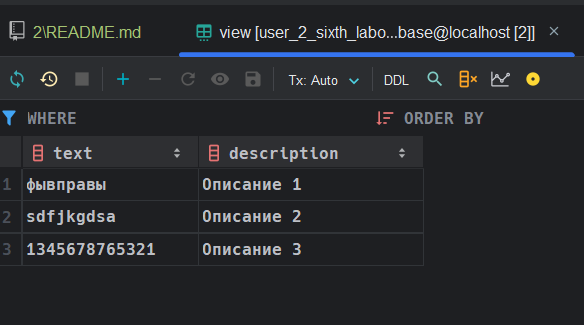


Рисунок 12 – проверка, что SELECT \* FROM View работает для user2

При попытке просмотра данных таблицы “table1” возникает ошибка, что говорит нам о том, что все корректно работает. Вывод ошибки представлен на рисунке 13.

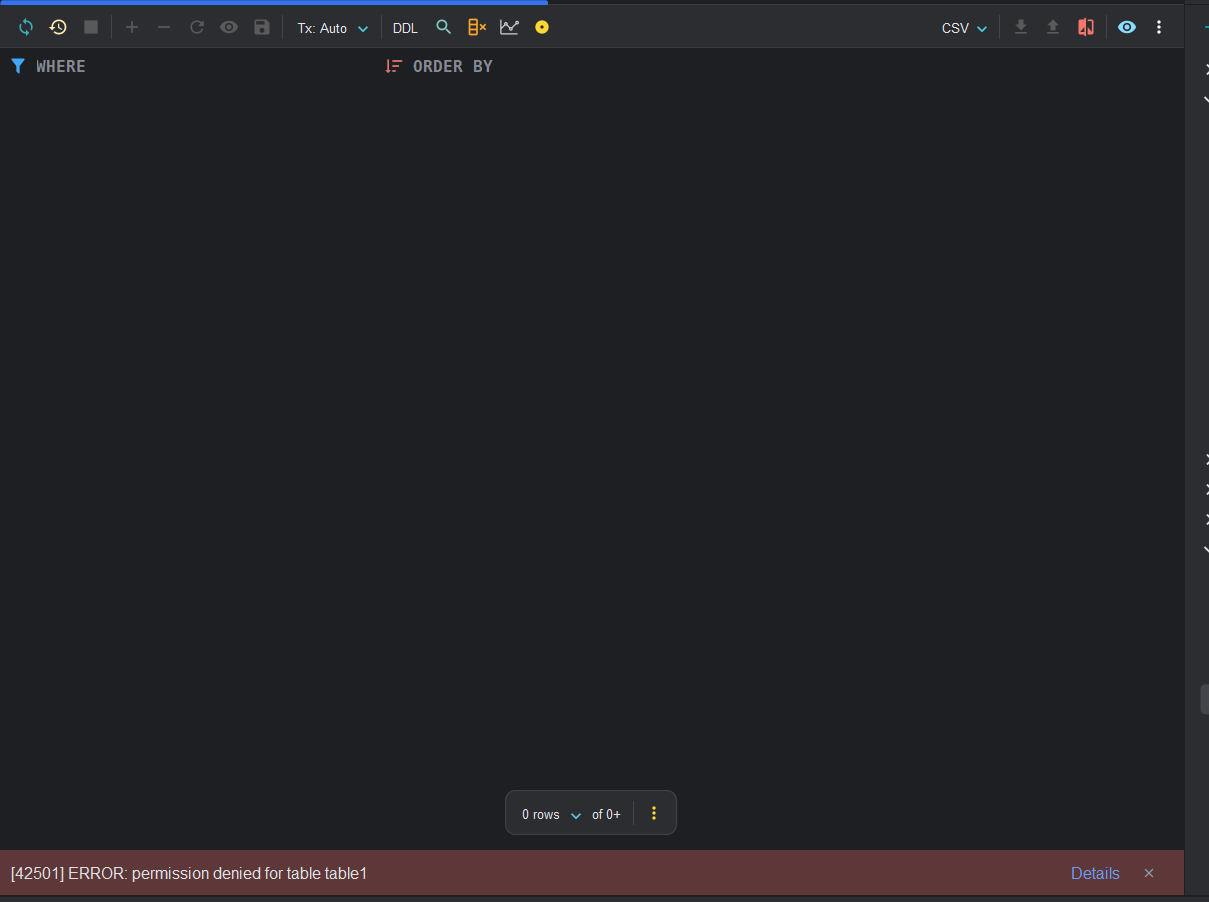


Рисунок 13 – ошибка при просмотре “table1” из-под пользователя user2

**Задание 3.** Создать от имени администратора (или пользователя, обладающего достаточными привилегиями) несколько таблиц в БД. Предоставить различным ролям различные привилегии (на выборку и модификацию) к созданным таблицам и продемонстрировать механизм разрешения использования ролей в рамках сеанса конкретного пользователя.

Для выполнения данного задания нужно войти из-под пользователя “postgres”. Находясь под данной учетной записью, был использован такой код, который представлен ниже на рисунке 14. Механизм разделения прав представлен в задании 4.

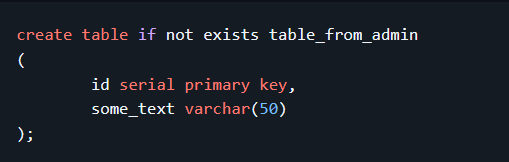


Рисунок 14 – создания таблицы для админа

**Задание 4.** Предоставить различным ролям различные привилегии (на выборку и модификацию) к созданным таблицам и продемонстрировать механизм разрешения использования ролей в рамках сеанса конкретного пользователя.

Для выполнения задания были выданы права на чтение с таблицы для пользователя “user1”, а для пользователя “user2” были выданы права только на вставку данных. Код представлен ниже на рисунке 15.

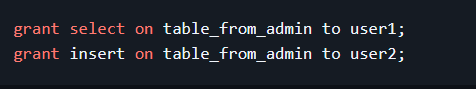


Рисунок 15 – код для выдачи прав “table\_from\_admin”

Для проверки работоспособности попробуем вставить данные в таблицу, находясь за пользователя в “user2”. Код представлен ниже на рисунке 16.

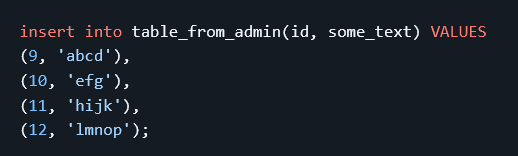


Рисунок 16 – код для вставки данных в таблицу

Видим, что изменения успешно применились, на ошибку не обращайте внимания, случайно было вставлено 2 раза по ошибке автора лабораторной работы. Результат выполнения кода представлен на рисунке 17.

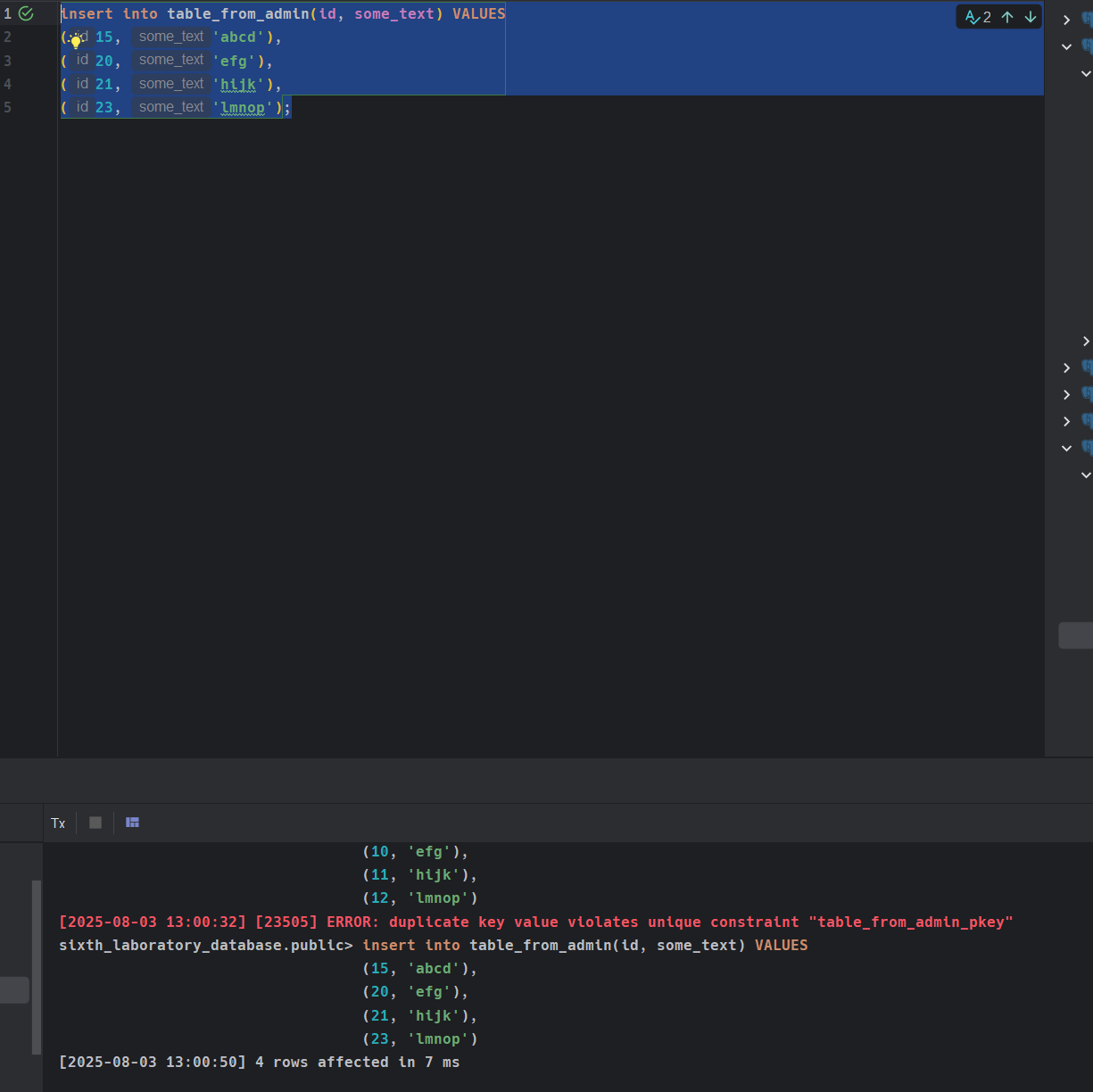


Рисунок 17 – результат выполнения запроса

Сделаем попытку просмотра данных и убедимся, что для пользователя “user2” ничего не видно в таблице. Результат ошибки представлен на рисунке 18.

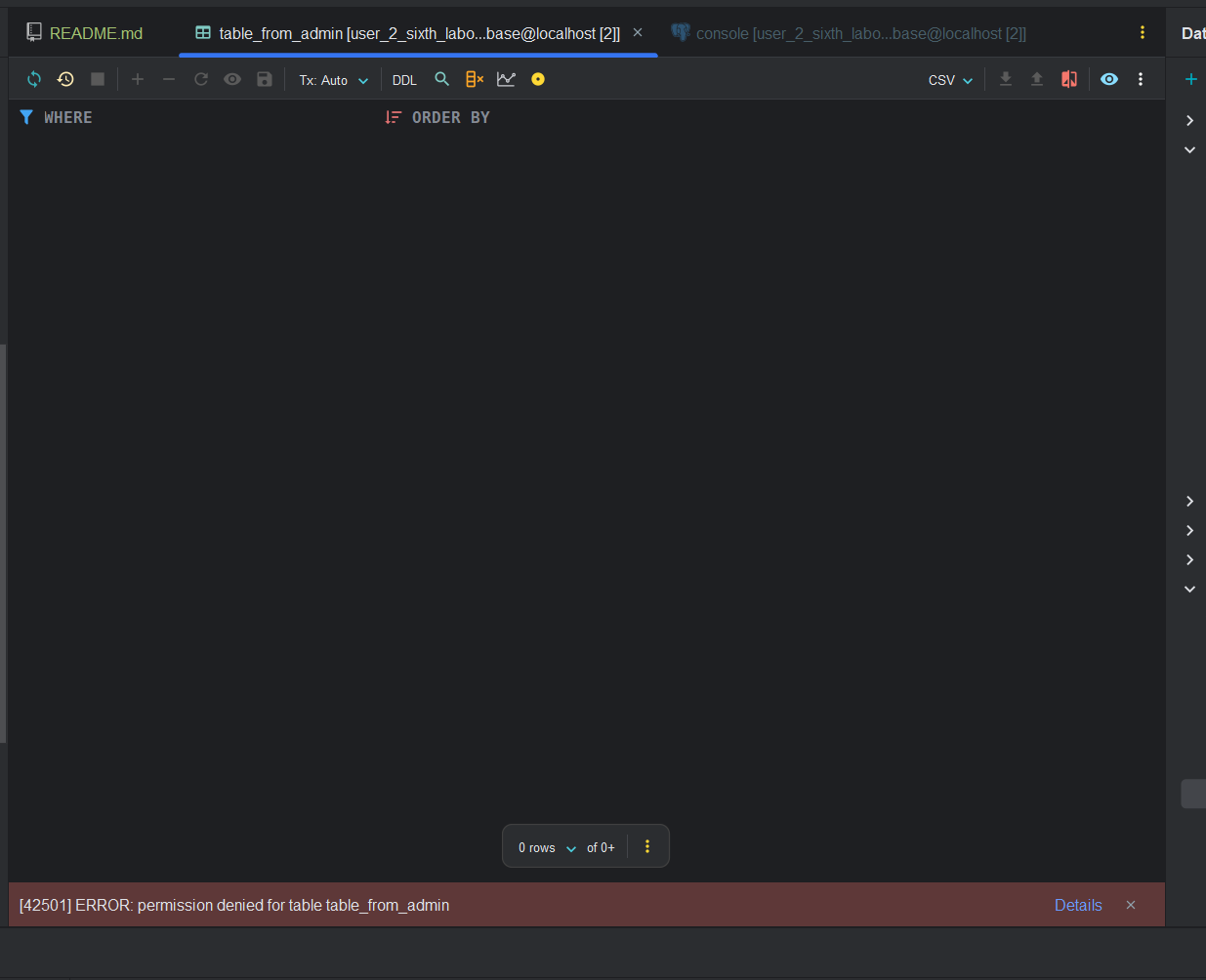


Рисунок 18 – ошибка при чтении данных из-под пользователя “user2”

Сделаем попытку просмотра данных из-под пользователя user1. На рисунке 19, что все данные видны.

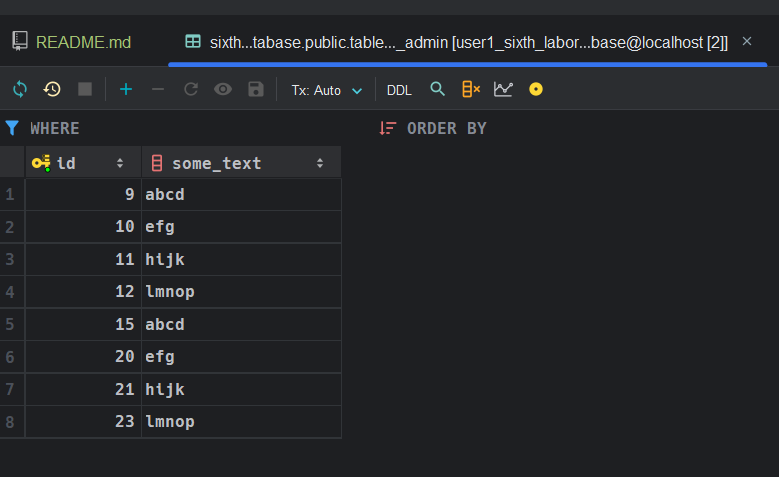


Рисунок 19 - Успешная выборка данных из таблицы от имени пользователя user1

Теперь будет произведена попытка вставки данных в таблицу “table\_for\_admins”. Ожидаемый результат, что будет возмущена ошибка “permission denied”, что говорит о корректности настройки распределения прав. Все представлено на рисунке 20.

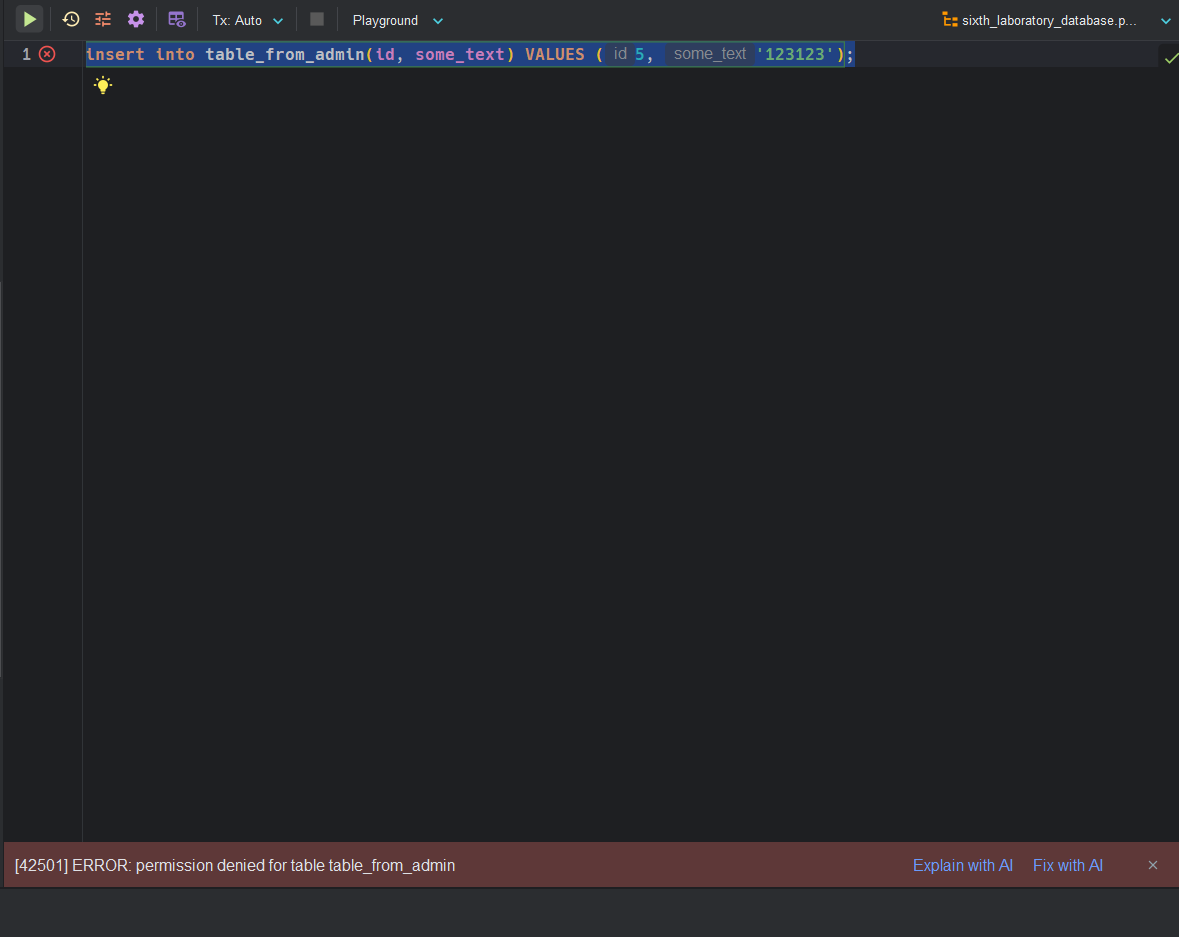


Рисунок 20 – попытка вставки данных в “table\_for\_admins” от имени пользователя user1