Национальный исследовательский ядерный университет «МИФИ»

(Московский Инженерно-Физический Институт)

Кафедра №42 «Криптология и кибербезопасность»

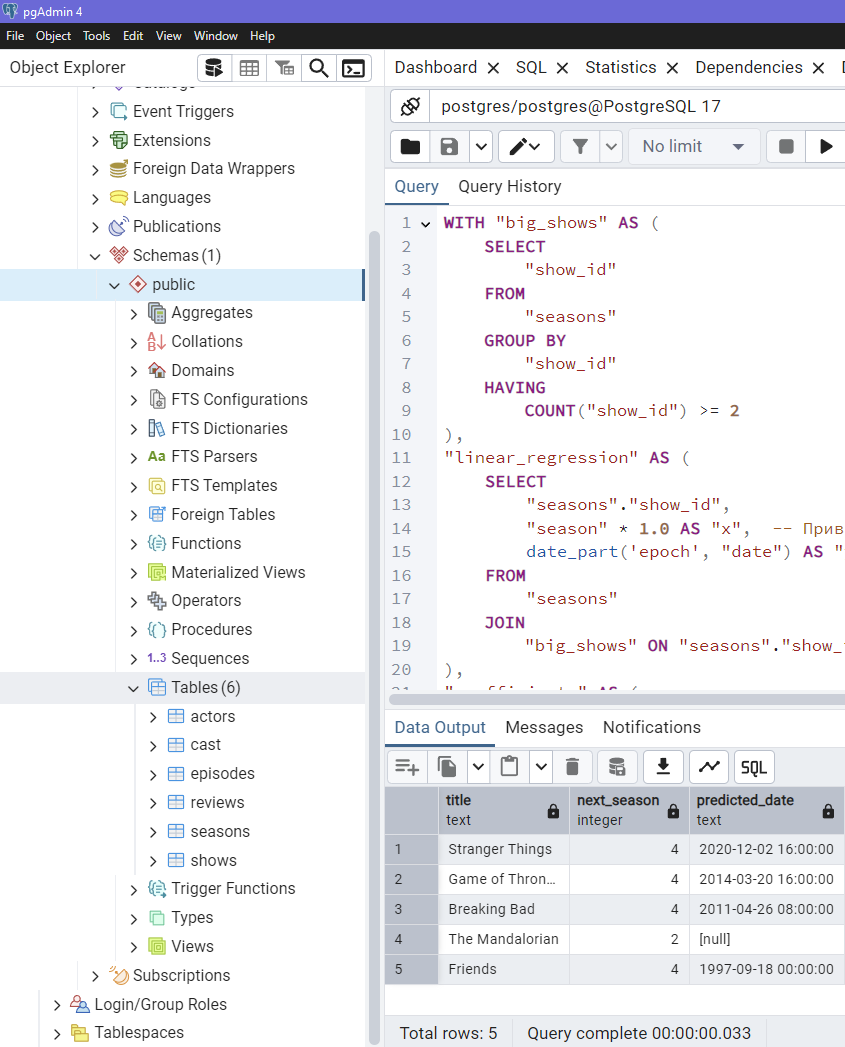
**Лабораторная работа №2-1**

**«**Пользователи. Роли. Привилегии**»**

Тимин Александр Б21-515 (2024г.)

# **1. Определить, в какой схеме находятся таблицы вашей базы данных. Следует ли изменить схему? Следует ли создать несколько отдельных схем для выбранной предметной области? Почему?**

В PostgreSQL, если явно не указана схема, все таблицы создаются в схеме public (это стандартное поведение). Так как при создании таблиц схема не была явно указана, все таблицы были созданы в public.



Здесь и далее будут приводиться скриншоты из pgAdmin 4.

В общем то можно создать несколько дополнительных схем (например, series, people, relationships):

* + **series** для информации о сериалах.
  + **people** для информации о людях (актеры, режиссеры и т.д.).
  + **relationships** для связей между людьми и сериалами (актеры и их роли).

Потенциально это облегчит администрирование, разграничение прав доступа и масштабирование проекта.

**Действия для переноса данных в схемы:**

1. Создать новые схемы:

CREATE SCHEMA series;

CREATE SCHEMA people;

CREATE SCHEMA relationships;

1. Переместить таблицы в соответствующие схемы:

ALTER TABLE shows SET SCHEMA series;

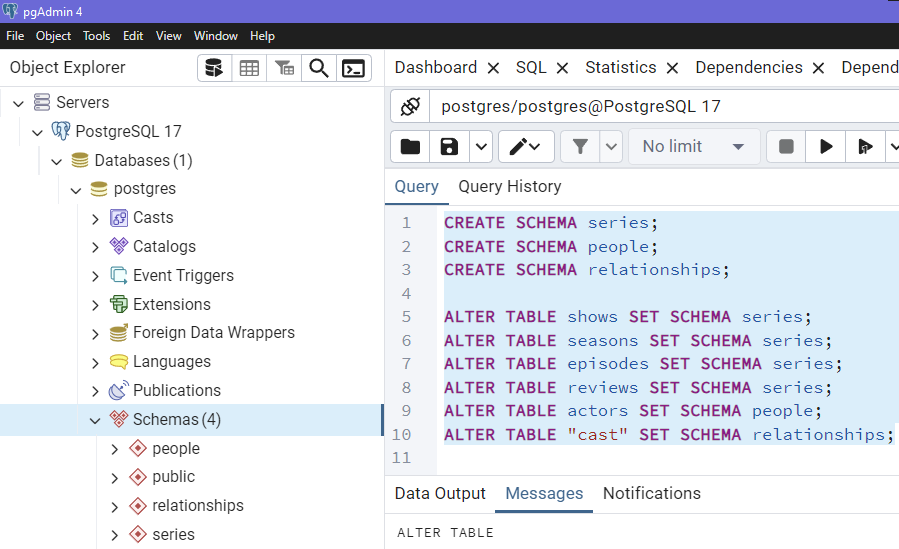
ALTER TABLE seasons SET SCHEMA series;

ALTER TABLE episodes SET SCHEMA series;

ALTER TABLE reviews SET SCHEMA series;

ALTER TABLE actors SET SCHEMA people;

ALTER TABLE “cast” SET SCHEMA relationships;



# **2. Определить, какие роли нужны для нормального функционирования вашей базы данных. Какие системные и объектные привилегии потребуются каждой роли? Понадобятся ли вложенные роли?**

Разделение пользователей на роли помогает реализовать принцип наименьших привилегий:

* **Администратор (admin)**: Полный контроль над базой данных. Эта роль нужна для настройки структуры базы, управления пользователями и привилегиями.
  + Системные привилегии: CREATEDB, CREATEROLE для управления БД и ролями.
  + Объектные привилегии: ALL PRIVILEGES на все таблицы и схемы.
* **Разработчик (developer)**: Разработчики должны иметь возможность изменять структуру таблиц, добавлять новые данные, но не управлять пользователями и базами данных.
  + Системные привилегии: только подключение (LOGIN).
  + Объектные привилегии: создание, чтение, обновление, но без удаления схем.
* **Читатель (reader)**: Для аналитиков или других сотрудников, которым нужен только доступ к данным.
  + Системные привилегии: только подключение (LOGIN).
  + Объектные привилегии: только чтение (SELECT) данных из таблиц.

Вложенные роли упрощают управление доступом. Например, если роль reader включена в роль developer, все разработчики автоматически получают права на чтение.

#### **3. Создать роли и выдать им необходимые объектные и системные привилегии.**

SQL-код для создания ролей и назначения привилегий:

1. Создание ролей:

CREATE ROLE admin WITH LOGIN PASSWORD 'admin' CREATEDB CREATEROLE;

CREATE ROLE developer WITH LOGIN PASSWORD 'dev';

CREATE ROLE reader WITH LOGIN PASSWORD 'reader';

1. Назначение вложенных ролей:

GRANT reader TO developer;

1. Назначение привилегий для схем:

GRANT ALL PRIVILEGES ON SCHEMA series TO admin;

GRANT ALL PRIVILEGES ON SCHEMA people TO admin;

GRANT ALL PRIVILEGES ON SCHEMA relationships TO admin;

GRANT CREATE, USAGE ON SCHEMA series TO developer;

GRANT CREATE, USAGE ON SCHEMA people TO developer;

GRANT CREATE, USAGE ON SCHEMA relationships TO developer;

GRANT USAGE ON SCHEMA series TO reader;

GRANT USAGE ON SCHEMA people TO reader;

GRANT USAGE ON SCHEMA relationships TO reader;

1. Назначение привилегий для таблиц:

GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA series TO admin;

GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA people TO admin;

GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA relationships TO admin;

GRANT SELECT, INSERT, UPDATE ON ALL TABLES IN SCHEMA series TO developer;

GRANT SELECT, INSERT, UPDATE ON ALL TABLES IN SCHEMA people TO developer;

GRANT SELECT, INSERT, UPDATE ON ALL TABLES IN SCHEMA relationships TO developer;

GRANT SELECT ON ALL TABLES IN SCHEMA series TO reader;

GRANT SELECT ON ALL TABLES IN SCHEMA people TO reader;

GRANT SELECT ON ALL TABLES IN SCHEMA relationships TO reader;

1. Настройка привилегий для будущих объектов:

ALTER DEFAULT PRIVILEGES IN SCHEMA series GRANT ALL PRIVILEGES ON TABLES TO admin;

ALTER DEFAULT PRIVILEGES IN SCHEMA people GRANT ALL PRIVILEGES ON TABLES TO admin;

ALTER DEFAULT PRIVILEGES IN SCHEMA relationships GRANT ALL PRIVILEGES ON TABLES TO admin;

ALTER DEFAULT PRIVILEGES IN SCHEMA series GRANT SELECT, INSERT, UPDATE ON TABLES TO developer;

ALTER DEFAULT PRIVILEGES IN SCHEMA people GRANT SELECT, INSERT, UPDATE ON TABLES TO developer;

ALTER DEFAULT PRIVILEGES IN SCHEMA relationships GRANT SELECT, INSERT, UPDATE ON TABLES TO developer;

ALTER DEFAULT PRIVILEGES IN SCHEMA series GRANT SELECT ON TABLES TO reader;

ALTER DEFAULT PRIVILEGES IN SCHEMA people GRANT SELECT ON TABLES TO reader;

ALTER DEFAULT PRIVILEGES IN SCHEMA relationships GRANT SELECT ON TABLES TO reader;

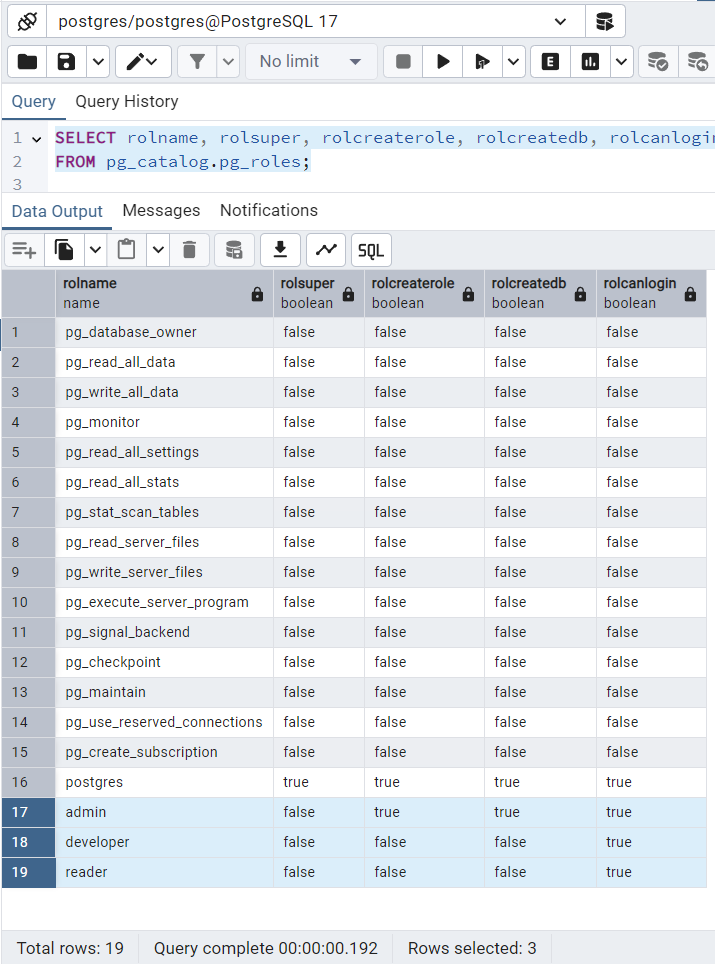
#### 

#### **4. Проверить по представлению системного каталога** pg\_catalog.pg\_roles**, что все нужные роли были созданы и обладают корректным набором привилегий.**

Для проверки ролей используем SQL-запрос:

SELECT rolname, rolsuper, rolcreaterole, rolcreatedb, rolcanlogin

FROM pg\_catalog.pg\_roles;



Запрос отобразил список ролей и их привилегии. Видно наличие созданных ролей: admin, developer и reader, а также, что флаг rolcanlogin установлен для всех ролей.

Проверка привилегий на схемы:

\dn+ series



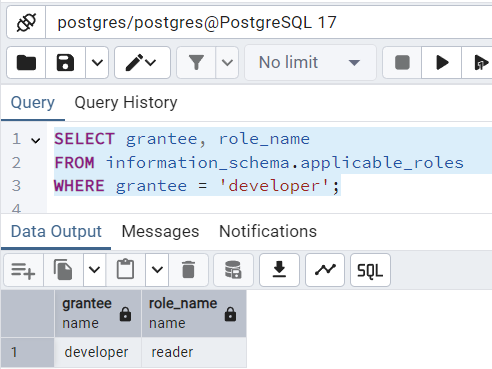
Из вывода команды видно, что ко всем схемам все роли имеют привилегию USAGE (доступ к объектам, не разрешает создание объектов), а также все кроме reader имеют привилегию CREATE (создание новых объектов).

Проверим наследование привилегий:

SELECT grantee, role\_name

FROM information\_schema.applicable\_roles

WHERE grantee = 'developer';



#### **Проверим привилегии по каждым таблицам для роли администратора:**

#### **SELECT grantee, privilege\_type, table\_schema, table\_name**

#### **FROM information\_schema.role\_table\_grants**

#### **WHERE grantee = 'admin';**

#### **Проверим привилегии по каждым таблицам для роли разработчика:**

#### **SELECT grantee, privilege\_type, table\_schema, table\_name**

#### **FROM information\_schema.role\_table\_grants**

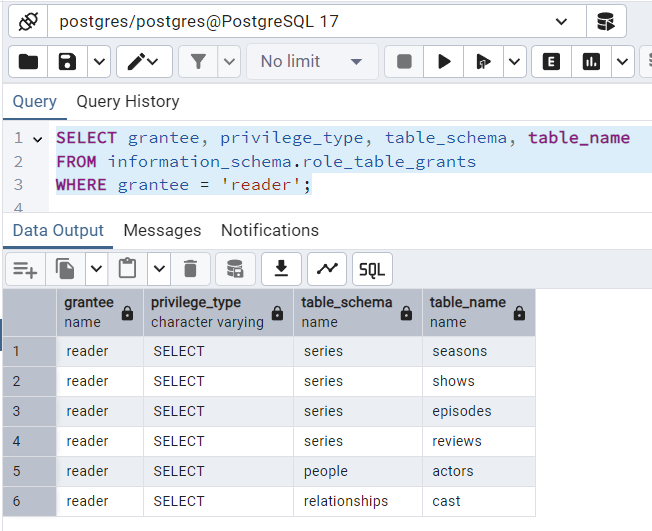
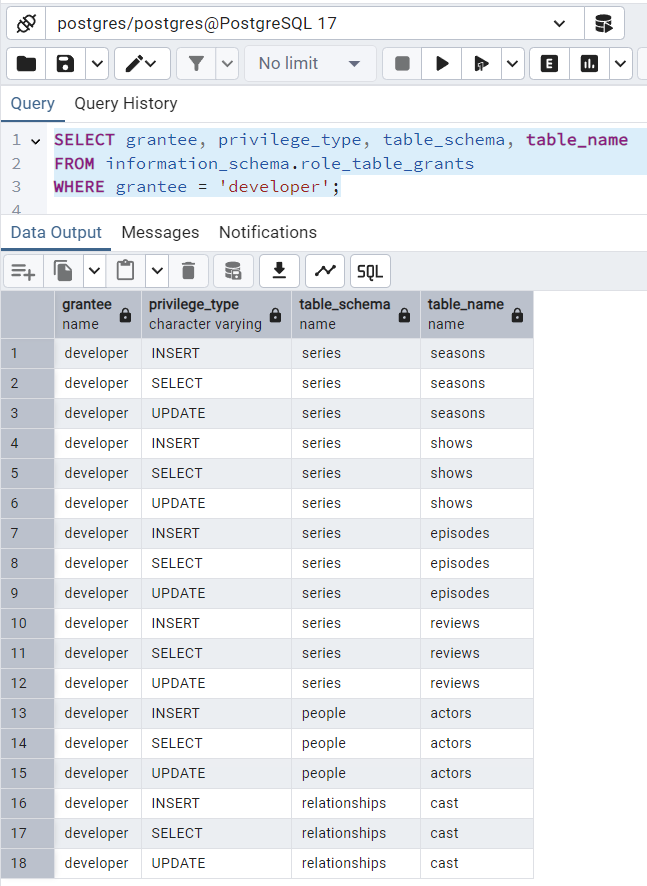
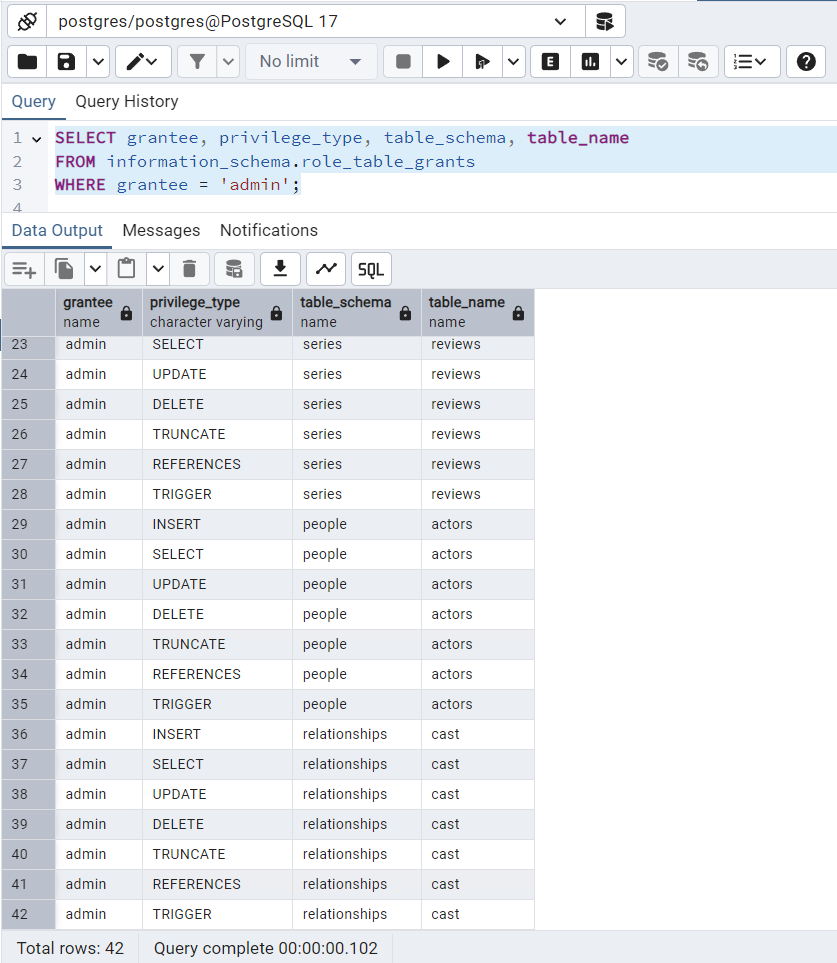
#### **WHERE grantee = 'developer';**

#### **Проверим привилегии по каждым таблицам для роли reader:**

#### **SELECT grantee, privilege\_type, table\_schema, table\_name**

#### **FROM information\_schema.role\_table\_grants**

#### **WHERE grantee = 'reader';**



#### **5. Попробовать подключиться от лица каждой роли и проверить доступ.**

Запросы:

1. От лица admin:

SELECT \* FROM series.shows;

INSERT INTO series.shows (title) VALUES ('New Show');

1. От лица developer:

SELECT \* FROM series.shows;

INSERT INTO series.shows (title) VALUES ('Dev Show');

1. От лица reader:

SELECT \* FROM series.shows;

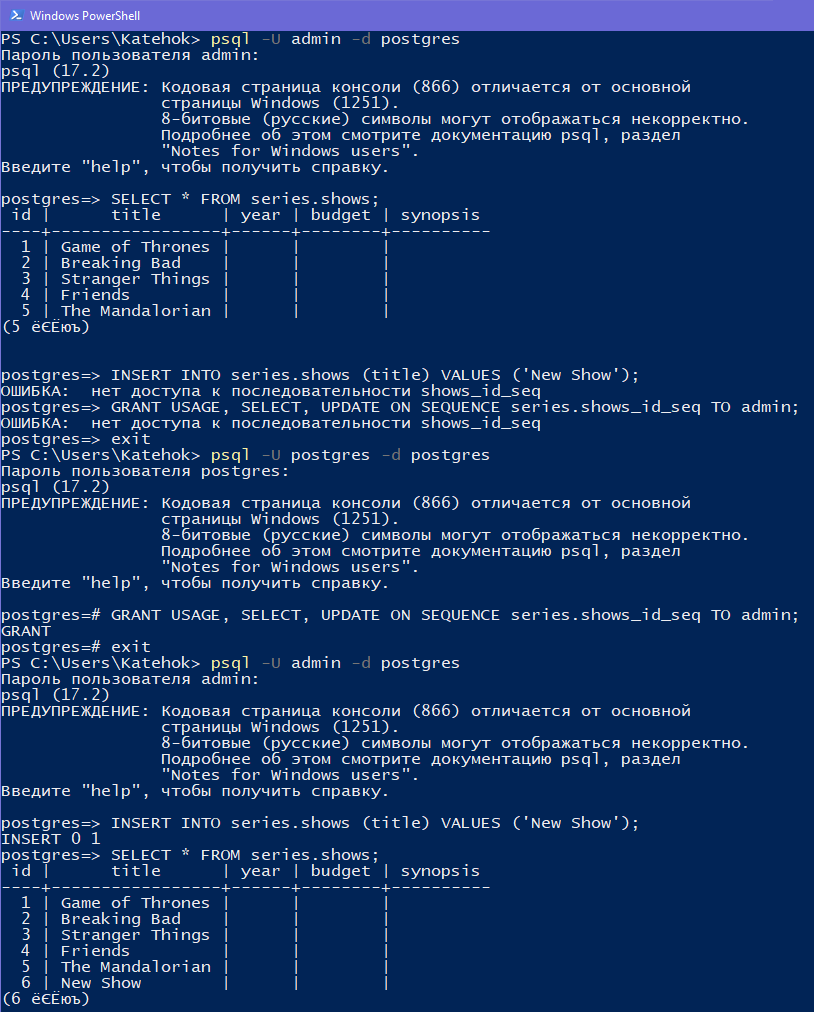
-- Проверка: ожидается ошибка.

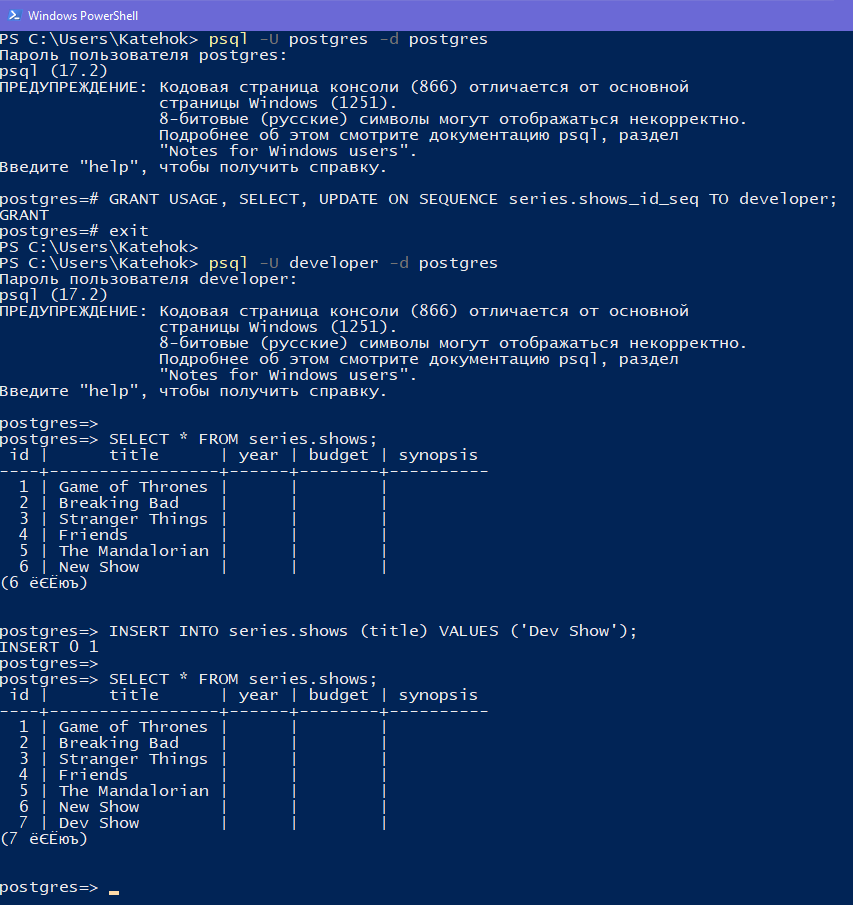
INSERT INTO series.shows (title) VALUES ('Test Show');

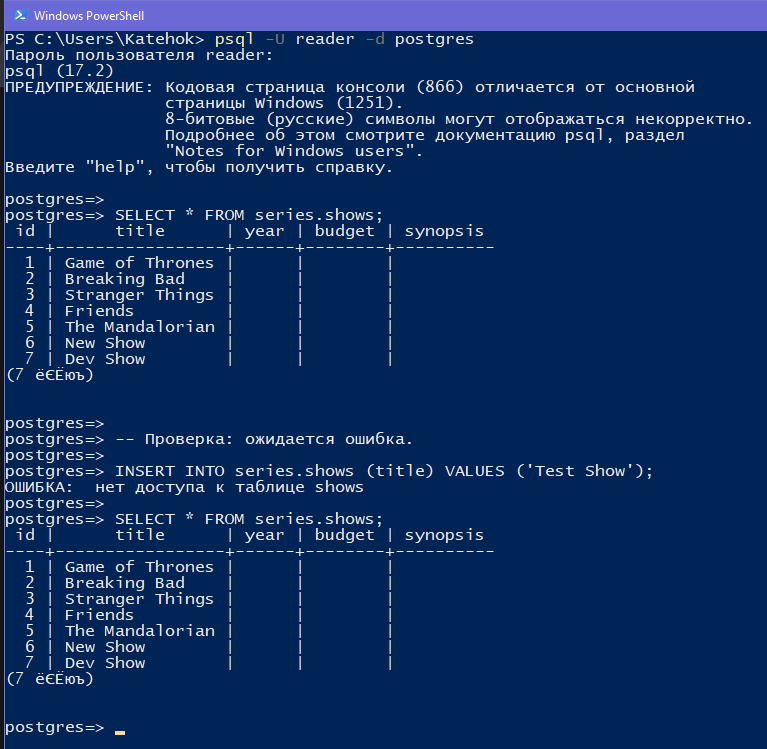
При попытке выполнить запрос вставки от лица администратора и разработчика высветилась ошибка, суть которой – отсутствие доступа к последовательности, с помощью которой работает автоинкремент индекса. Проблема решилась наделением соответствующих ролей необходимыми привилегиями:

GRANT USAGE, SELECT, UPDATE ON SEQUENCE series.shows\_id\_seq TO admin;

GRANT USAGE, SELECT, UPDATE ON SEQUENCE series.shows\_id\_seq TO developer;

****

****

****

**Заключение**

В ходе работы в соответствии с предметной областью была создана и заполнена база данных сериалов, к ней было выполнено несколько составных запросов. Также было проведено сравнение PostgreSQL и SQLite.

Приложение

* [директория последней лабы из репозитория предыдущего семестра](https://github.com/KATEHOK/DBS_labs-2024/tree/main/4);
* [ERD структуры таблиц](https://github.com/KATEHOK/DBS_labs-2024/blob/main/4/pics/ERD.png);
* [отчет (docx)](https://github.com/KATEHOK/DBS-labs-7/blob/main/1/report.docx);
* [отчет (pdf)](https://github.com/KATEHOK/DBS-labs-7/blob/main/1/report.pdf).