

**ФГАОУ ВО «МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ»**

**Лабораторная работа №5**

Управление доступом в базах данных

По дисциплине:

Базы данных

Выполнил студент 1-го курса группы 243-323

Онищенко А. А.

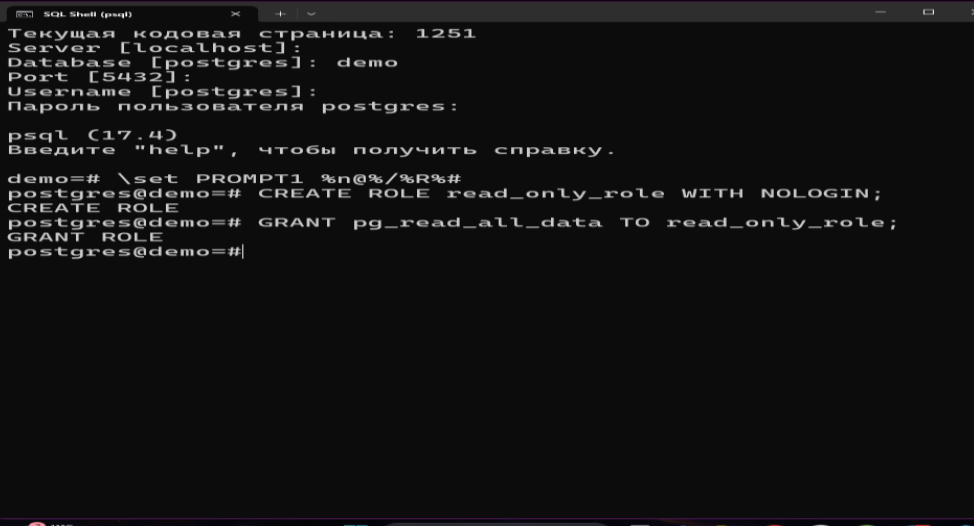
Проверил

\_\_\_\_\_ Красникова И.Н.

**Москва, 2024**

## 1. Упражнение 5.1

Создайте роль для доступа на чтение к демонстрационной базе данных без права создания сеансов работы с сервером БД.



The screenshot shows a Windows desktop environment. A terminal window titled "SQL Shell (psql)" is open, displaying the following text:

```

Текущая кодовая страница: 1251
Server [localhost]:
Database [postgres]: demo
Port [5432]:
Username [postgres]:
Пароль пользователя postgres:

psql (17.4)
Введите "help", чтобы получить справку.

demo=# \set PROMPT1 %n@%/%R%#
postgres@demo=# CREATE ROLE read_only_role WITH NOLOGIN;
CREATE ROLE
postgres@demo=# GRANT pg_read_all_data TO read_only_role;
GRANT ROLE
postgres@demo=#
  
```

At the bottom of the screen, the Windows taskbar is visible, showing the Start button, a search bar labeled "Поиск", and several application icons including File Explorer, Microsoft Edge, and various utility programs.

## 2. Упражнение 5.2

Создайте пользователя сервера БД и предоставьте ему привилегию использования роли, созданной в предыдущем упражнении. Проверьте, что этот пользователь может выполнять любые запросы на выборку из таблиц демонстрационной базы данных, но не может их обновлять.

The screenshot displays two terminal windows side-by-side.

**Left Window (SQL Shell [pgdg]):**

```
postgres@demo=#
postgres@demo=#
postgres@demo=#
postgres@demo=#
postgres@demo=#
postgres@demo=#
postgres@demo=#
postgres@demo=#
postgres@demo=#
postgres@demo=# CREATE ROLE analyst WITH LOGIN PASSWORD 'test';
CREATE ROLE
postgres@demo=# GRANT read_only_role TO analyst;
GRANT ROLE
postgres@demo=#
postgres@demo=#
postgres@demo=#
postgres@demo=#
postgres@demo=#
postgres@demo=#
postgres@demo=#
postgres@demo=#
postgres@demo=#
postgres@demo=#
```

**Right Window (SQL Shell [pgdg]):**

```
Текущая каталожная страница: 1251
Server [localhost]:
Database [postgres]: demo
Port [5432]:
Username [postgres]: analyst
Пароль пользователя analyst:

psql (17.4)
Введите "help", чтобы получить справку.

demo=> \set PROMPT1 %A/%R%
analyst@demo=> SELECT * FROM bookings.aircrafts;
Текущая каталожная страница: 1251
?column?
-----
      0
(1 строка)

analyst@demo=> SELECT * FROM bookings.aircrafts;
Текущая каталожная страница: 1251
aircraft_code | model          | range
-----|-----|-----
773           | Boeing 777-300 | 11180
763           | Boeing 767-300 | 7980
SU9           | Sukhoi SuperJet-100 | 3880
320           | Airbus A320-200 | 5780
321           | Airbus A321-200 | 5680
319           | Airbus A319-100 | 6780
733           | Boeing 737-300  | 4280
CN1           | Cessna 280 Caravan | 1280
CR2           | Bombardier CRJ-200 | 2780
(9 строк)
```

```
SQL Shell (psql)
analyst@demo=> UPDATE aircrafts
demo-> SET range = range + 100
demo-> WHERE aircraft_code = CN1;
ОШИБКА: отношение "aircrafts" не существует
СТРОКА 1: UPDATE aircrafts

analyst@demo=> SELECT * FROM bookings.aircrafts;
Текущая кодовая страница: 1251
aircraft_code | model | range
-----
773 | Boeing 777-300 | 11100
763 | Boeing 767-300 | 7900
SU9 | Sukhoi SuperJet-100 | 3000
320 | Airbus A320-200 | 5700
321 | Airbus A321-200 | 5600
319 | Airbus A319-100 | 6700
733 | Boeing 737-300 | 4200
CN1 | Cessna 208 Caravan | 1200
CR2 | Bombardier CRJ-200 | 2700
(9 строк)

analyst@demo=> DROP bookings.aircrafts;
ОШИБКА: ошибка синтаксиса (примерное положение: "bookings")
СТРОКА 1: DROP bookings.aircrafts;

analyst@demo=> DROP TABLE bookings.aircrafts;
ОШИБКА: нужно быть владельцем таблицы aircrafts
analyst@demo=>
```

### 3. Упражнение 5.3

Заберите у пользователя привилегию, выданную в предыдущем упражнении. Убедитесь, что этот пользователь не сможет выбирать данные из таблиц демобазы.

```
SQL Shell (psql)
Текущая кодовая страница: 1251
Server [localhost]:
Database [postgres]: demo
Port [5432]:
Username [postgres]:
Пароль пользователя postgres:

psql (17.4)
Введите "help", чтобы получить справку.

demo=# \set %n@%/%R%#
неправильное имя переменной: "%n@%/%R%#"
demo=# \set PROMPT1 %n@%/%R%#
postgres@demo=# REVOKE read_only_role FROM analyst;
REVOKE ROLE
postgres@demo=#
```

```
SQL Shell (psql)
Текущая кодовая страница: 1251
Server [localhost]:
Database [postgres]: demo
Port [5432]:
Username [postgres]: analyst
Пароль пользователя analyst:

psql (17.4)
Введите "help", чтобы получить справку.

demo=> \set PROMPT1 %n@%/%R%#
analyst@demo=> SELECT 0;
Текущая кодовая страница: 1251
?column?
-----
0
(1 строка)

analyst@demo=>SELECT * FROM bookings.aircrafts;
ОШИБКА: нет доступа к схеме bookings
СТРОКА 1: SELECT * FROM bookings.aircrafts;

analyst@demo=>SELECT * FROM bookings.tickets;
ОШИБКА: нет доступа к схеме bookings
СТРОКА 1: SELECT * FROM bookings.tickets;

analyst@demo=>
```

## 4. Упражнение 5.4

Постройте пример, показывающий, что для доступа к таблицам схемы необходимо также предоставить право использования (USAGE) этой схемы.

В упражнении 5.1 привилегия была создана при помощи встроенной роли `pg_read_all_data`, которая предоставляет права как на `SELECT` так и на `USAGE`, которая предоставляет доступ к схемам. Воспользуемся ролью `analyst` из прошлых упражнений, чтобы продемонстрировать необходимость `USAGE`:

```
SQL Shell (psql)
Текущая кодовая страница: 1251
Server [localhost]:
Database [postgres]: demo
Port [5432]:
Username [postgres]:
Пароль пользователя postgres:

psql (17.4)
Введите "help", чтобы получить справку.

demo=# \set PROMPT1 %n@%/%R%#
postgres@demo=# GRANT SELECT TO analyst;
ОШИБКА: роль "select" не существует
postgres@demo=# GRANT SELECT ON ALL TABLES TO analyst;
ОШИБКА: ошибка синтаксиса (примерное положение: "TO")
СТРОКА 1: GRANT SELECT ON ALL TABLES TO analyst;
               ^
postgres@demo=# GRANT SELECT ON ALL TABLES IN SCHEMA bookings TO analyst;
GRANT
postgres@demo=#

SQL Shell (psql)
Текущая кодовая страница: 1251
Server [localhost]:
Database [postgres]: demo
Port [5432]:
Username [postgres]: analyst
Пароль пользователя analyst:

psql (17.4)
Введите "help", чтобы получить справку.

demo=> \set PROMPT1 %n@%/%R%#
analyst@demo=> SELECT * FROM bookings.aircrafts;
ОШИБКА: нет доступа к схеме bookings
СТРОКА 1: SELECT * FROM bookings.aircrafts;
               ^
analyst@demo=>
```

# Ответы на контрольные вопросы

## 1. Что такое принципал?

Принципал (principal) в контексте систем управления базами данных (СУБД) — это объект, которому могут быть предоставлены разрешения. Принципалы могут быть индивидуальными (например, учетные записи пользователей) или группами (например, роли). Они представляют собой субъекты безопасности, которые могут выполнять действия в базе данных.

## 2. Что такое объект?

Объект в СУБД — это элемент, с которым можно взаимодействовать. Это может быть таблица, представление, индекс, процедура или другой элемент базы данных. Объекты имеют свои свойства и могут быть защищены с помощью привилегий, которые определяют, какие действия могут выполняться над ними.

## 3. Что такое действие?

Действие — это операция, которую принципал может выполнять над объектом. В SQL действия могут включать операции, такие как `SELECT`, `INSERT`, `UPDATE`, `DELETE`, и другие. Каждое действие требует соответствующих привилегий для его выполнения.

## 4. Как назначаются роли? Чем они отличаются?

Роли назначаются с помощью команды `GRANT`, которая позволяет пользователю или другому принципалу получить доступ к определенным привилегиям. Роли могут быть предопределенными (например, администраторы, пользователи) или пользовательскими, созданными для конкретных нужд. Основное отличие ролей заключается в том, что они могут группировать привилегии, что упрощает управление доступом, позволяя назначать наборы привилегий нескольким пользователям одновременно.

## 5. Какие операторы SQL отвечают за создание, модификации и удаление ролей?

Операторы SQL, отвечающие за создание, модификацию и удаление ролей, включают:

- Создание роли: `CREATE ROLE`
- Изменение роли: `ALTER ROLE`
- Удаление роли: `DROP ROLE`

## 6. Какие операторы SQL отвечают за предоставление привилегий?

Операторы SQL, отвечающие за предоставление привилегий, включают:

- Предоставление привилегий: `GRANT`
- Отмена привилегий: `REVOKE`