

Государственное бюджетное профессиональное образовательное учреждение
«Нижегородский радиотехнический колледж»

МДК 11.01 Технология разработки и защиты баз данных.

ОТЧЁТ

по практической работе.
Тема «**Роли и атрибуты**»

Выполнил:
обучающийся группы 2ИСиП-19-1
Мамонов Антон

Проверил:
Преподаватель
Гутянская Е.М.

Нижний Новгород
Г.

Задание 1. Создайте роль creator без права входа в систему, но с правом создания баз данных и ролей.

create role creator createdb createrole;

```
postgres=# create role creator createdb createrole;  
CREATE ROLE
```

Задание 2. Создайте пользователя weak с правом входа в систему.

```
postgres=# create role weak login;  
CREATE ROLE
```

Задание 3. Убедитесь, что weak не может создать базу данных

```
postgres=# \c - weak;  
Вы подключены к базе данных "postgres" как пользователь "weak".  
postgres=> create database test_database;  
ОШИБКА: нет прав на создание базы данных
```

Задание 4. Включите пользователя weak в группу creator

```
postgres=# grant creator to weak;  
GRANT ROLE
```

Задание 5. Создайте новую базу данных под пользователем weak.

Имя роли	Список ролей		Член ролей
	Атрибуты		
alice	Создаёт роли		{}
creator	Создаёт роли, Создаёт БД, Вход запрещён		{}
dmask			{}
postgres	Суперпользователь, Создаёт роли, Создаёт БД, Репликация, Пропускать RLS		{}
weak			{creator}

Создадим роль.

```
postgres=# create role alice login createrole
postgres=# ;
CREATE ROLE
```

alice имеет возможность подключения и создания других ролей. проверим это:

```
privileges_database=# \c - alice;
Вы подключены к базе данных "privileges_database" как пользователь "alice".
```

Создать bob.

```
privileges_database=> create role bob login;
CREATE ROLE
```

```
postgres=> \c - bob
Вы подключены к базе данных "postgres" как пользователь "bob".
postgres=> create role charlie login;
ОШИБКА: нет прав для создания роли
```

bob не сможет создать другую роль:

```
postgres=> \c - bob
Вы подключены к базе данных "postgres" как пользователь "bob".
postgres=> create role charlie login;
ОШИБКА: нет прав для создания роли
```

Просмотр имеющихся ролей:

Имя роли		Список ролей	Член ролей
		Атрибуты	
alice		Создаёт роли	{}
bob			{}
creator		Создаёт роли, Создаёт БД, Вход запрещён	{}
dmosk			{}
postgres		Суперпользователь, Создаёт роли, Создаёт БД, Репликация, Пропускать RLS	{}

Существующие роли можно изменять.

```
postgres=> alter role bob nologin;
ALTER ROLE
```

```
postgres=> alter role alice nocreaterole;
ALTER ROLE
```

Теперь bob не сможет подключиться ((

```
postgres=> \c - bob
ВАЖНО: для роли "bob" вход запрещён
Сохранено предыдущее подключение
```

alice теперь не может ни создавать новые роли, ни изменять атрибуты существующих.

```
postgres=> alter role bob login;  
ОШИБКА: нет доступа
```

```
postgres=> \c - postgres  
Пароль пользователя postgres:  
Вы подключены к базе данных "postgres" как пользователь "postgres".  
postgres=# grant postgres to alice;  
GRANT ROLE  
postgres=# \du
```

Имя роли	Список ролей	Атрибуты	Член ролей
alice			{postgres}

Все команды теперь попадают в журнал сообщений

```
postgres=# alter role alice set log_min_duration_statement = 0;  
ALTER ROLE
```

Ограничим alice действия конкретной базой данных.

```
postgres=# alter role alice reset log_min_duration_statement;  
ALTER ROLE  
postgres=# alter role alice in database postgres set log_min_duration_statement  
= 0;  
ALTER ROLE
```

```
postgres=# \c - alice  
Вы подключены к базе данных "postgres" как пользователь "alice".  
postgres=> set role postgres;  
SET  
postgres=# alter role bob login;  
ALTER ROLE
```

```
postgres=# select session_user, current_user;  
session_user | current_user  
-----+-----  
alice        | postgres  
(1 строка)
```

```
postgres=# reset role;  
RESET
```

Владение объектами

```
postgres=> create table test(id integer);  
CREATE TABLE  
postgres=> \dt test;  
Список отношений  
Схема | Имя | Тип | Владелец  
-----+-----+-----+-----  
public | test | таблица | alice  
(1 строка)
```

Удаление ролей

```
postgres=> \c - postgres
Пароль пользователя postgres:
Вы подключены к базе данных "postgres" как пользователь "postgres".
postgres=# drop role alice;
ОШИБКА: роль "alice" нельзя удалить, так как есть зависящие от неё объекты
ПОДРОБНОСТИ: владелец объекта таблица test
```

```
postgres=# reassign owned by alice to bob;
REASSIGN OWNED
postgres=# \dt test;
          Список отношений
 Схема | Имя | Тип | Владелец
-----+-----+-----+-----
 public | test | таблица | bob
(1 строка)

postgres=# drop role alice;
DROP ROLE
```

```
postgres=# drop owned by bob;
DROP OWNED
postgres=# drop role bob;
DROP ROLE
```