Мамонов Антон 3ИСиП-19-1

База данных и роли

```
postgres=# CREATE DATABASE access_privileges;
CREATE DATABASE
postgres=# CREATE USER writer;
CREATE ROLE
postgres=# CREATE USER reader;
CREATE ROLE
```

Привилегии

```
postgres=# \c access_privileges
Вы подключены к базе данных "access_privileges" как пользователь "postgres
access_privileges=# REVOKE ALL ON SCHEMA public FROM public;
REVOKE
access_privileges=# GRANT ALL ON SCHEMA public TO writer;
GRANT
access_privileges=# GRANT USAGE ON SCHEMA public TO reader;
GRANT
```

Привилегии по умолчанию

```
access_privileges=# ALTER DEFAULT PRIVILEGES
access_privileges-# FOR ROLE writer
access_privileges-# GRANT SELECT ON TABLES TO reader;
ALTER DEFAULT PRIVILEGES
access_privileges=#
```

Пользователи

```
access_privileges=# CREATE ROLE w1 LOGIN IN ROLE writer;
CREATE ROLE
```

Конструкция IN ROLE сразу же добавляет новую роль в указанную. То есть такая команда эквивалентна: REATE ROLE w1 LOGIN; или GRANT writer TO w1;

```
access_privileges=# CREATE ROLE r1 LOGIN IN ROLE reader;
CREATE ROLE
```

Таблица

```
access_privileges=# \c - writer
Вы подключены к базе данных "access_privileges" как пользователь "writer".
access_privileges=> CREATE TABLE t(n integer);
CREATE TABLE
```

Проверка

Роль w1 может вставлять строки:

```
access_privileges=> \c - w1
Вы подключены к базе данных "access_privileges" как пользователь "w1".
access_privileges=> INSERT INTO t VALUES (42);
INSERT 0 1
```

Роль r1 может читать таблицу:

```
access_privileges=> \c - r1
Вы подключены к базе данных "access_privileges" как пользователь "r1".
access_privileges=> SELECT * FROM t;
 n
----
 42
(1 ёєЁюър)
```

Но не может изменить:

```
access_privileges=> UPDATE t SET n = n + 1;
ОШИБКА:  нет доступа к таблице t
```

Роль w1 может удалить таблицу:

```
access_privileges=> \c - w1
Вы подключены к базе данных "access_privileges" как пользователь "w1".
access_privileges=> drop table t;
DROP TABLE
```