Лабораторная работа №2-1: «Пользователи. Роли. Привилегии»

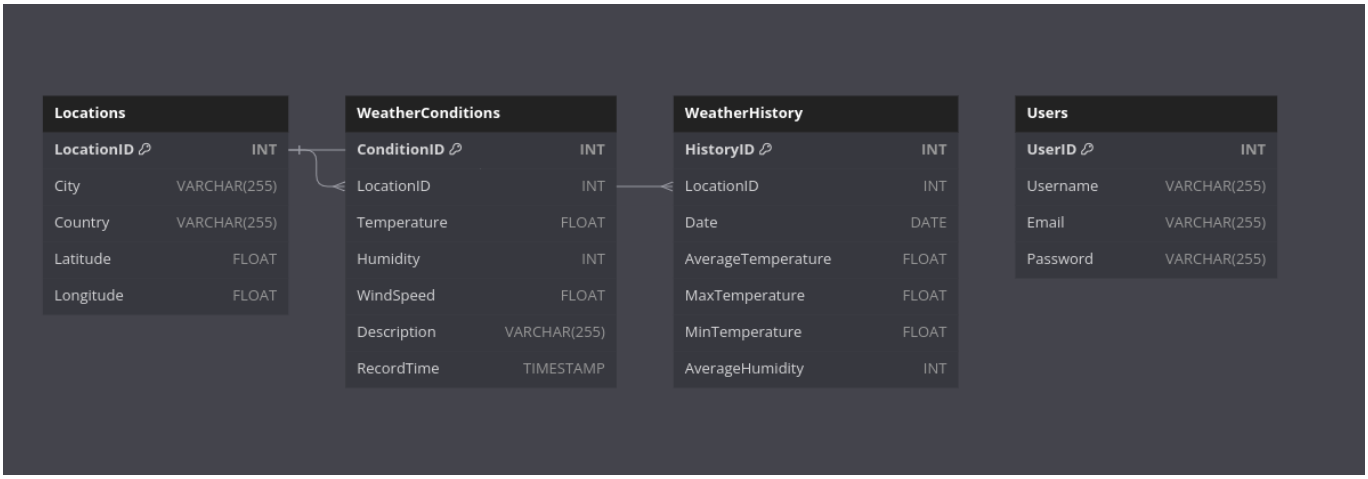
Выполнил студент группы Б20-505, Ландаков Павел

Структура базы данных

За основу была взята структура базы данных метереологического приложения. Все таблицы можно создать при помощи следующего SQL запроса:

```
CREATE TABLE Locations (
    LocationID INT PRIMARY KEY,
    City VARCHAR(255),
    Country VARCHAR(255),
    Latitude FLOAT,
    Longitude FLOAT
);

CREATE TABLE WeatherConditions (
    ConditionID INT PRIMARY KEY,
    LocationID INT,
    Temperature FLOAT,
    Humidity INT,
    WindSpeed FLOAT,
    Description VARCHAR(255),
    RecordTime TIMESTAMP,
    FOREIGN KEY (LocationID) REFERENCES Locations(LocationID)
);

CREATE TABLE WeatherHistory (
    HistoryID INT PRIMARY KEY,
    LocationID INT,
    Date DATE,
    AverageTemperature FLOAT,
    MaxTemperature FLOAT,
    MinTemperature FLOAT,
    AverageHumidity INT,
    FOREIGN KEY (LocationID) REFERENCES Locations(LocationID)
);


CREATE TABLE Users (
    UserID INT PRIMARY KEY,
    Username VARCHAR(255),
    Email VARCHAR(255),
    Password VARCHAR(255)
);
```

Даграмма классов:



Определить, в какой схеме находятся таблицы Вашей базы данных

Определить к какой схеме относятся таблицы можно введя `\d`:

```
lndk@localhost:lab2> \d
+--------+-------------------+-------+-------+
| Schema | Name              | Type  | Owner |
|--------+-------------------+-------+-------|
| public | locations         | table | lndk  |
| public | users             | table | lndk  |
| public | weatherconditions | table | lndk  |
| public | weatherhistory    | table | lndk  |
+--------+-------------------+-------+-------+
```

Или при помощи подобного SQL запроса:

```
SELECT table_name, table_schema
FROM information_schema.tables
WHERE table_schema NOT IN ('information_schema', 'pg_catalog')
ORDER BY table_schema, table_name;
```

```
+-------------------+--------------+
| table_name        | table_schema |
|-------------------+--------------|
| locations         | public       |
| users             | public       |
| weatherconditions | public       |
| weatherhistory    | public       |
+-------------------+--------------+
```

Следует ли изменить схему? Следует ли создать несколько отдельных схем для выбранной предметной области? Почему?

Нахождение таблиц в схеме `public` не является чем-то незаконным. Но в соответствии с лабораторной работой для упрощения организации и разделения данных (в перспективе) можно создать несколько отдельных схем и разнести таблицы по ним. Также разделение на схемы облегчит управление доступом и повысит безопасность, потому что можно будет настроить доступ ролей к определенным схемам.

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

Могут потребоваться следующие роли:

- **DBA (админ)**: имеет полный доступ ко всей базе;
- **Разработчик**: CREATE, ALTER, DROP на схемах и объектах, которые они разрабатывают;
- **Аналитик данных**: SELECT на необходимых таблицах;
- **Оператор ввода данных**: INSERT, UPDATE на определенных таблицах;
- **Пользователь приложения**: SELECT, UPDATE на ограниченном наборе данных, связанных с пользовательским интерфейсом.

Базово вложенные роли не являются необходимыми, но если к примеру потребуется роль "Модератор", то его права могут быть наследованы от *Пользователя приложения* и дополнительно расширены до возможностей редактирования других пользователей.

Создать роли и выдать им необходимые объектные и системные привилегии

SQL запрос:

```
CREATE ROLE dba WITH LOGIN PASSWORD 'strong_password_dba' SUPERUSER
CREATEDB CREATEROLE;

CREATE ROLE developer WITH LOGIN PASSWORD 'strong_password_dev';
GRANT CREATE ON DATABASE lab2 TO developer;

CREATE ROLE data_analyst WITH LOGIN PASSWORD 'strong_password_analyst';
GRANT SELECT ON ALL TABLES IN SCHEMA public TO data_analyst;

CREATE ROLE data_entry WITH LOGIN PASSWORD 'strong_password_entry';
GRANT INSERT, UPDATE ON TABLE WeatherConditions, WeatherHistory TO
data_entry;

CREATE ROLE app_user WITH LOGIN PASSWORD 'strong_password_user';
GRANT SELECT ON TABLE Locations TO app_user;
GRANT SELECT, UPDATE ON TABLE Users TO app_user;
```

Проверить по представлению системного каталога pg catalog.pg roles, что все нужные роли были созданы и обладают корректным

набором привилегий.

SQL запрос:

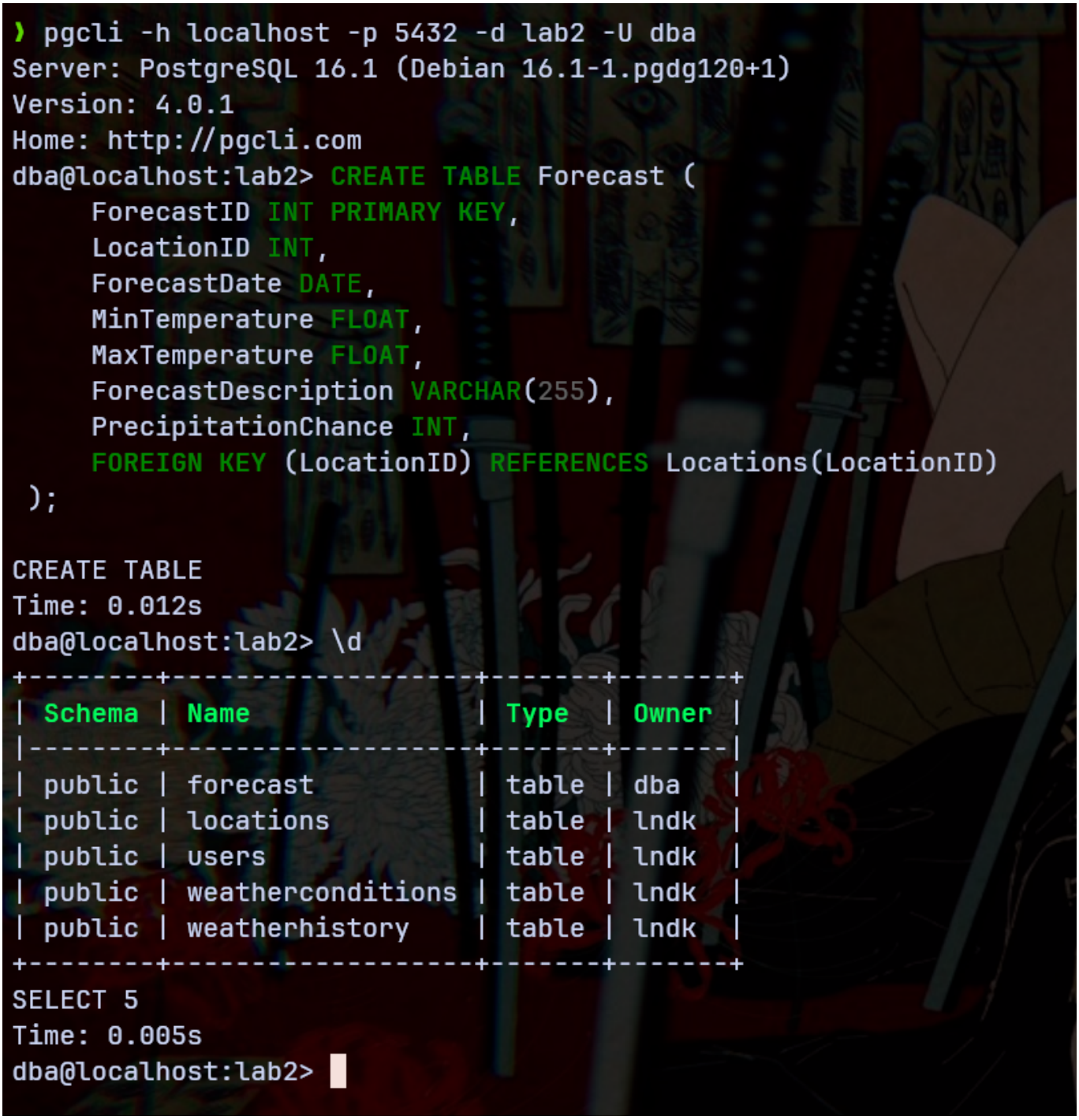
```
SELECT rolname, rolsuper, rolcreaterole, rolcreatedb, rolcanlogin,
rolconnlimit, rolpassword
FROM pg_catalog.pg_roles
WHERE rolname NOT LIKE 'pg_%';
```

Результат:

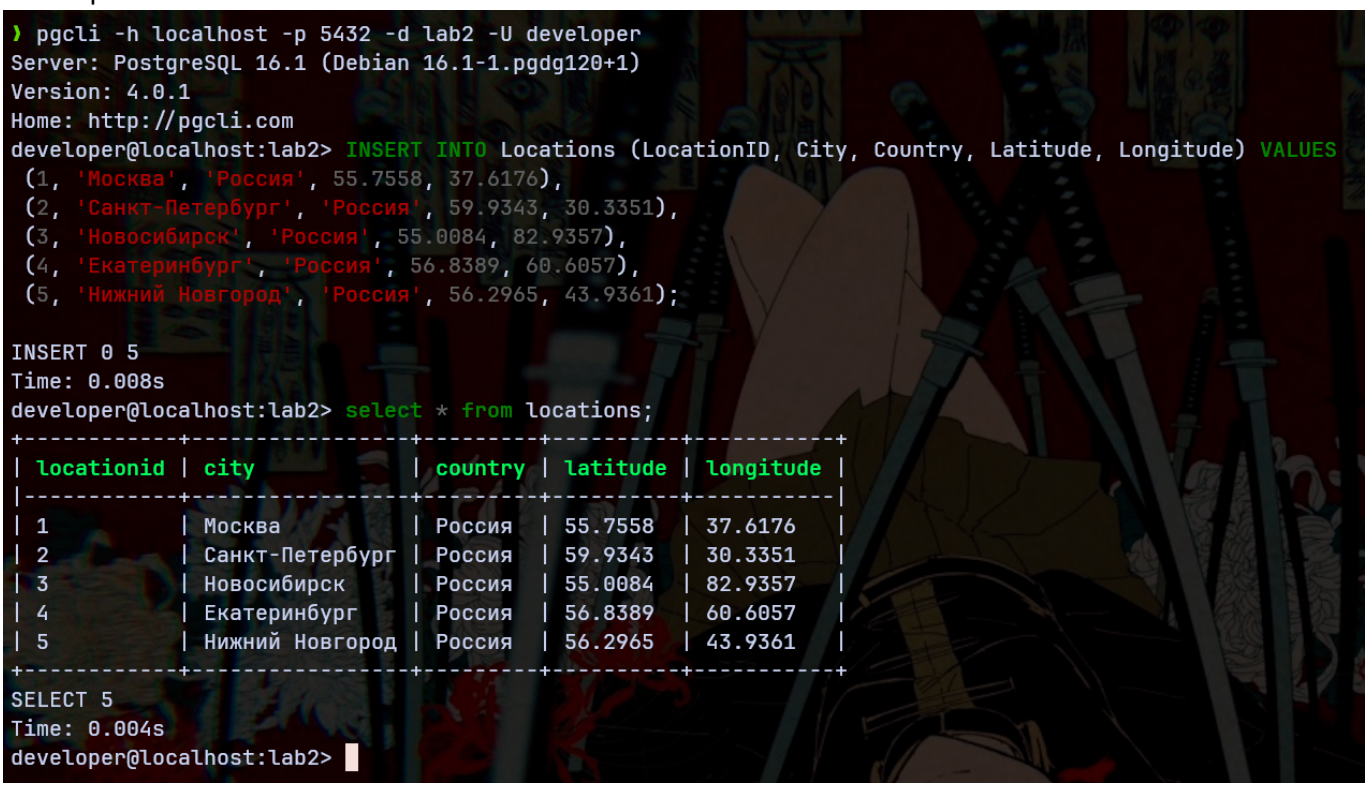
```
+--------------+----------+---------------+-------------+-------------+----
----------+-------------+
| rolname      | rolsuper | rolcreaterole | rolcreatedb | rolcanlogin |
rolconnlimit | rolpassword |
|--------------+----------+---------------+-------------+-------------+----
----------+-------------|
| lndk         | True     | True          | True        | True        | -1
| ********    |
| dba          | True     | True          | True        | True        | -1
| ********    |
| developer    | False    | False         | False       | True        | -1
| ********    |
| data_analyst | False    | False         | False       | True        | -1
| ********    |
| data_entry   | False    | False         | False       | True        | -1
| ********    |
| app_user     | False    | False         | False       | True        | -1
| ********    |
+--------------+----------+---------------+-------------+-------------+----
----------+-------------+
```

Попробовать подключиться от лица каждой роли (из тех, которым разрешено подключение ксерверу БД). Убедиться, что роль имеет доступ к разрешённым данным и не имеет доступа ко всем остальным

DBA:

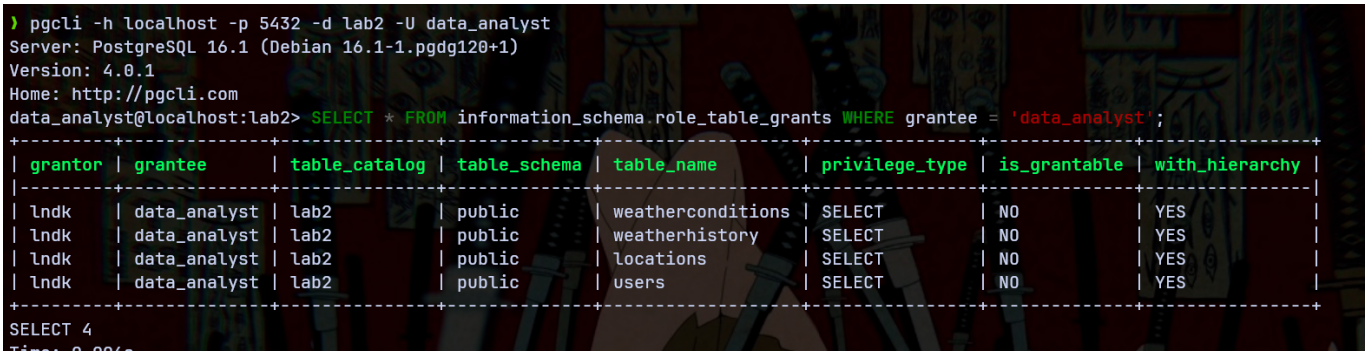

❯ pgcli -h localhost -p 5432 -d lab2 -U dba
Server: PostgreSQL 16.1 (Debian 16.1-1.pgdg120+1)
Version: 4.0.1
Home: http://pgcli.com
dba@localhost:lab2> CREATE TABLE Forecast (
ForecastID INT PRIMARY KEY,
LocationID INT,
ForecastDate DATE,
MinTemperature FLOAT,
MaxTemperature FLOAT,
ForecastDescription VARCHAR(255),
PrecipitationChance INT,
FOREIGN KEY (LocationID) REFERENCES Locations(LocationID)
);
CREATE TABLE
Time: 0.012s
dba@localhost:lab2> \d
+--------+--------------------+-------+-------+
| Schema | Name | Type | Owner |
|--------+--------------------+-------+-------|
public	forecast	table	dba
public	locations	table	lndk
public	users	table	lndk
public	weatherconditions	table	lndk
public	weatherhistory	table	lndk
+--------+--------------------+-------+-------+
SELECT 5
Time: 0.005s
dba@localhost:lab2>

Developer:



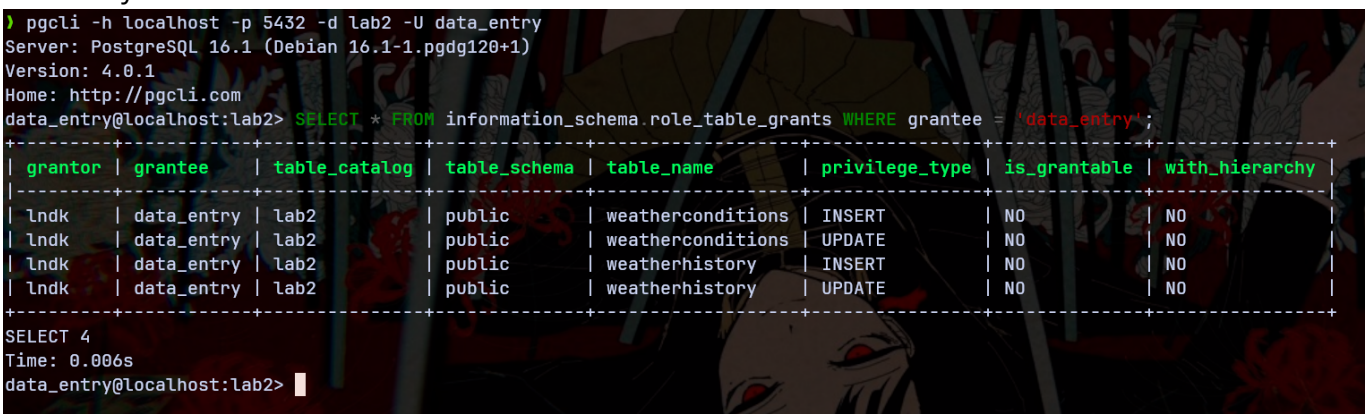
```
❯ pgcli -h localhost -p 5432 -d lab2 -U developer
Server: PostgreSQL 16.1 (Debian 16.1-1.pgdg120+1)
Version: 4.0.1
Home: http://pgcli.com
developer@localhost:lab2> INSERT INTO Locations (LocationID, City, Country, Latitude, Longitude) VALUES
 (1, 'Москва', 'Россия', 55.7558, 37.6176),
 (2, 'Санкт-Петербург', 'Россия', 59.9343, 30.3351),
 (3, 'Новосибирск', 'Россия', 55.0084, 82.9357),
 (4, 'Екатеринбург', 'Россия', 56.8389, 60.6057),
 (5, 'Нижний Новгород', 'Россия', 56.2965, 43.9361);

INSERT 0 5
Time: 0.008s
developer@localhost:lab2> select * from locations;
+------------+-----------------+---------+----------+-----------+
| locationid | city            | country | latitude | longitude |
|------------+-----------------+---------+----------+-----------|
| 1          | Москва          | Россия  | 55.7558  | 37.6176   |
| 2          | Санкт-Петербург | Россия  | 59.9343  | 30.3351   |
| 3          | Новосибирск     | Россия  | 55.0084  | 82.9357   |
| 4          | Екатеринбург    | Россия  | 56.8389  | 60.6057   |
| 5          | Нижний Новгород | Россия  | 56.2965  | 43.9361   |
+------------+-----------------+---------+----------+-----------+
SELECT 5
Time: 0.004s
developer@localhost:lab2>
```

Data analyst:



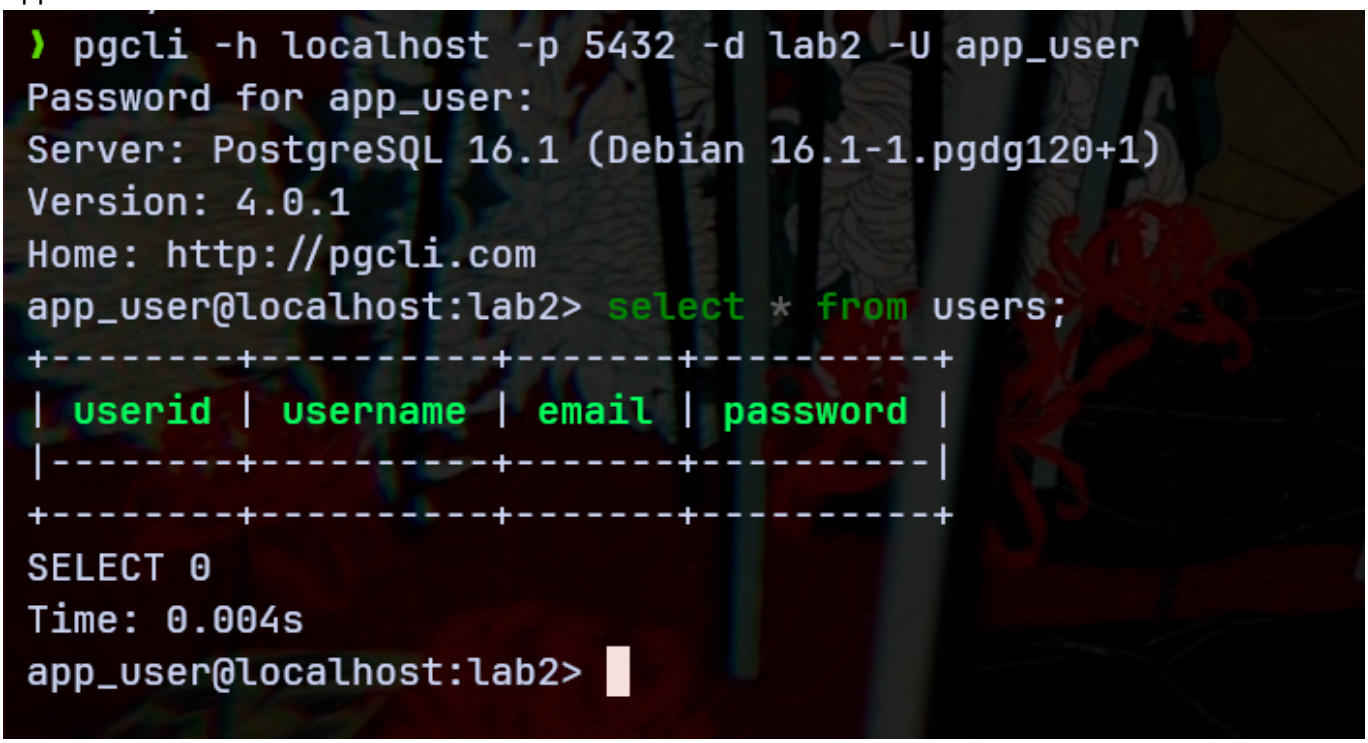
```
❯ pgcli -h localhost -p 5432 -d lab2 -U data_analyst
Server: PostgreSQL 16.1 (Debian 16.1-1.pgdg120+1)
Version: 4.0.1
Home: http://pgcli.com
data_analyst@localhost:lab2> SELECT * FROM information_schema.role_table_grants WHERE grantee = 'data_analyst';
+---------+--------------+---------------+--------------+------------------+----------------+--------------+----------------+
| grantor | grantee      | table_catalog | table_schema | table_name       | privilege_type | is_grantable | with_hierarchy |
|---------+--------------+---------------+--------------+------------------+----------------+--------------+----------------|
| lndk    | data_analyst | lab2          | public       | weatherconditions | SELECT         | NO           | YES            |
| lndk    | data_analyst | lab2          | public       | weatherhistory    | SELECT         | NO           | YES            |
| lndk    | data_analyst | lab2          | public       | locations         | SELECT         | NO           | YES            |
| lndk    | data_analyst | lab2          | public       | users             | SELECT         | NO           | YES            |
+---------+--------------+---------------+--------------+------------------+----------------+--------------+----------------+
SELECT 4
```

Data entry:



```
❯ pgcli -h localhost -p 5432 -d lab2 -U data_entry
Server: PostgreSQL 16.1 (Debian 16.1-1.pgdg120+1)
Version: 4.0.1
Home: http://pgcli.com
data_entry@localhost:lab2> SELECT * FROM information_schema.role_table_grants WHERE grantee = 'data_entry';
+---------+------------+---------------+--------------+------------------+----------------+--------------+----------------+
| grantor | grantee    | table_catalog | table_schema | table_name       | privilege_type | is_grantable | with_hierarchy |
|---------+------------+---------------+--------------+------------------+----------------+--------------+----------------|
| lndk    | data_entry | lab2          | public       | weatherconditions | INSERT         | NO           | NO             |
| lndk    | data_entry | lab2          | public       | weatherconditions | UPDATE         | NO           | NO             |
| lndk    | data_entry | lab2          | public       | weatherhistory    | INSERT         | NO           | NO             |
| lndk    | data_entry | lab2          | public       | weatherhistory    | UPDATE         | NO           | NO             |
+---------+------------+---------------+--------------+------------------+----------------+--------------+----------------+
SELECT 4
Time: 0.006s
data_entry@localhost:lab2>
```

App user:



Заключение:

В результате данной лабораторной работы было произведено ознакомление со схемами и ролями в PostgreSQL. Для тестовой базы данных метереологического приложения были созданы несколько ролей, которым были выданы необходимые привилегии для работы с определенными таблицами.