

Лекция 6.

Importing and Exporting Data

Importing and Exporting Data

Цель:

- Использовать **psql** в командной строке для эффективного взаимодействия с базой данных.
- Использовать команду **COPY** для загрузки данных в базу данных.
- Использовать **Excel** для обработки данных.
- Упростить свой код с помощью **SQLAlchemy** в **Python**.

The COPY Command

Postgres COPY эффективно передает данные из базы данных в файл. или из файла в базу данных.
Например, проведем следующие манипуляции:

```
admin@pgadmin-VM:~$ sudo -i -u postgres
[sudo] пароль для admin:
postgres@pgadmin-VM:~$ psql
psql (14.7 (Ubuntu 14.7-0ubuntu0.22.04.1))
Type "help" for help.
```

```
postgres=# \c sqllda2
You are now connected to database "sqllda2" as user "postgres".
sqllda2=# COPY (SELECT * FROM customers LIMIT 5) TO STDOUT WITH CSV HEADER;
```

The COPY Command

```
customer_id,title,first_name,last_name,suffix,email,gender,ip_address,phone,street_address,city,state,postal_code,latitude,longitude,date_added
1,,Arlena,Riveles,,ariveles8@stumbleupon.com,F,98.36.172.246,,,,,,2017-04-23 00:00:00
2,Dr,Ode,Stovin,,ostovin1@npr.org,M,16.97.59.186,314-534-4361,2573 Fordem Parkway,Saint Louis,MO,63116,38.5814,-90.2625,2014-10-02 00:00:00
3,,Braden,Jordan,,bjordan2@geocities.com,M,192.86.248.59,,5651 Kennedy Park,Pensacola,FL,32590,30.6143,-87.2758,2018-10-27 00:00:00
4,,Jessica,Nussen,,jnussen3@salon.com,F,159.165.138.166,615-824-2506,224 Village Circle,Nashville,TN,37215,36.0986,-86.8219,2017-09-03 00:00:00
5,,Lonnie,Rembaud,,lrembaud4@discovery.com,F,18.131.58.65,786-499-3431,38 Lindbergh Way,Miami,FL,33124,25.5584,-80.4582,2014-03-06 00:00:00
```

Этот оператор возвращает пять строк из таблицы, где каждая запись находится на новой строке, а каждое значение разделено запятой в типичном формате файла **.csv**. Заголовок также включен.

The COPY Command

COPY (SELECT * FROM customers LIMIT 5) TO STDOUT WITH CSV HEADER;

COPY — это команда, используемая для передачи данных в формат файла.

(SELECT * FROM customers LIMIT 5) — это запрос, который хотим скопировать.

TO STDOUT указывает, что результаты следует распечатать, а не сохранять в файл на жестком диске.

«**Standard Out**» — это общий термин для отображения вывода в среде терминала командной строки.

WITH — это необязательное ключевое слово, используемое для разделения параметров, которые будем использовать при передаче из базы данных в файл.

CSV указывает, что будем использовать формат файла **CSV**.

Можем указать **BINARY** или вообще исключить это и получить вывод в текстовом формате.

HEADER указывает, что хотим распечатать заголовок.

<https://www.postgresql.org/docs/current/sql-copy.html>

The COPY Command

Хотя опция **STDOUT** полезна, однако чаще нам нужно сохранить данные в файл.

Команда **COPY** предлагает функциональные возможности для этого, но данные сохраняются локально на сервере **Postgres**. Необходимо указать полный путь к файлу (относительные пути к файлам не допускаются).

```
COPY (SELECT * FROM customers LIMIT 5) TO '/home/postgres/my_file.csv' WITH CSV HEADER;
```

Copying Data with psql

одним из первых клиентов **Postgres** на самом деле была программа командной строки под названием **psql**.

Этот интерфейс используется до сих пор, а **psql** предлагает отличные возможности для запуска скриптов **Postgres** и взаимодействия с локальной вычислительной средой.

Это позволяет удаленно вызывать команду **COPY** с помощью специфичной для **psql** инструкции **\copy**, которая вызывает **COPY**.

Чтобы запустить **psql**, вы можете запустить следующую команду в Терминале:

```
psql -h my_host -p 5432 -d my_database -U my_username
```

psql. Выгрузка данных из таблиц во внешнее хранилище(csv)

Например:

```
CREATE TEMP VIEW customers_sample AS (  
SELECT *  
FROM customers  
LIMIT 5  
);
```

```
\copy customers_sample TO 'my_file2.csv' WITH CSV header;
```

```
DROP VIEW customers_sample;
```


psql. Выгрузка данных из таблиц во внешнее хранилище(csv)



```
admin@pgadmin-VM:~$ sudo -i -u postgres
```

```
postgres@pgadmin-VM:~$ psql
```

```
psql (14.11 (Ubuntu 14.11-0ubuntu0.22.04.1))Type "help" for help.
```

```
postgres=# \l
```

```
postgres=# \c sqlda2
```

```
You are now connected to database "sqlda2" as user "postgres".
```

```
sqlda2=# SHOW data_directory;
```

```
data_directory ----- /var/lib/postgresql/14/main(1 row)
```

```
sqlda2=# CREATE TEMP VIEW customers_sample AS (SELECT *FROM customers LIMIT 5);
```

```
sqlda2=# \dv
```

```
List of relations Schema | Name | Type | Owner -----+-----+-----+-----
-- pg_temp_12 | customers_sample | view | postgres(1 row)
```

```
sqlda2=# sqlda2=# \copy (SELECT *FROM customers LIMIT 5) TO
'/var/lib/postgresql/my_file3.csv' WITH CSV header;
```

```
sqlda2=# DROP VIEW customers_sample;
```

Настройка COPY и \copy

Варианты настройки команд **COPY** и **\copy**:

- **FORMAT format_name** может использоваться для указания формата. Возможные варианты для **format_name**: **csv**, текст или двоичный файл. В качестве альтернативы вы можете просто указать CSV или BINARY без ключевого слова FORMAT или вообще не указывать формат и позволить выходным данным по умолчанию быть в формате текстового файла.
- **DELIMITER 'delimiter_character'** можно использовать для указания символа-разделителя для CSV-файлов или текстовых файлов (например, ',' для CSV-файлов или '|' для файлов, разделенных вертикальной чертой).
- **NULL 'null_string'** может использоваться для указания того, как должны быть представлены нулевые значения (например, ' ', если пробелы представляют нулевые значения, или 'NULL', если именно так должны быть представлены отсутствующие значения в данных).
- **HEADER** указывает, что должен быть выведен заголовок.

Настройка COPY и \copy

Варианты настройки команд **COPY** и **\copy**:

- **QUOTE** 'quote_character' можно использовать для указания того, как поля со специальными символами (например, запятая в текстовом значении в файле CSV) могут заключаться в кавычки, чтобы они игнорировались при **COPY**.
- **ESCAPE** 'escape_character' определяет символ, который можно использовать для экранирования следующего символа.
- **ENCODING** 'encoding_name' позволяет указать кодировку, что особенно полезно, когда вы имеете дело с иностранными языками, содержащими специальные символы или пользовательский ввод.

psql. Выгрузка данных в таблицу из внешнего хранилища(csv)

```
\copy customers FROM 'my_file.csv' CSV HEADER DELIMITER ',';
```

Варианты настройки команд COPY и \copy:

- **QUOTE** 'quote_character' можно использовать для указания того, как поля со специальными символами (например, запятая в текстовом значении в файле CSV) могут заключаться в кавычки, чтобы они игнорировались при COPY.
- **ESCAPE** 'escape_character' определяет символ, который можно использовать для экранирования следующего символа.
- **ENCODING** 'encoding_name' позволяет указать кодировку, что особенно полезно, когда вы имеете дело с иностранными языками, содержащими специальные символы или пользовательский ввод.

Вариант 1. Выгрузка данных в таблицу из внешнего хранилища(csv)

1. Создать таблицу в Postgresql

https://github.com/BosenkoTM/SQL-for-Begginer-Data-Analytics/blob/main/Datasets/top_cities.csv

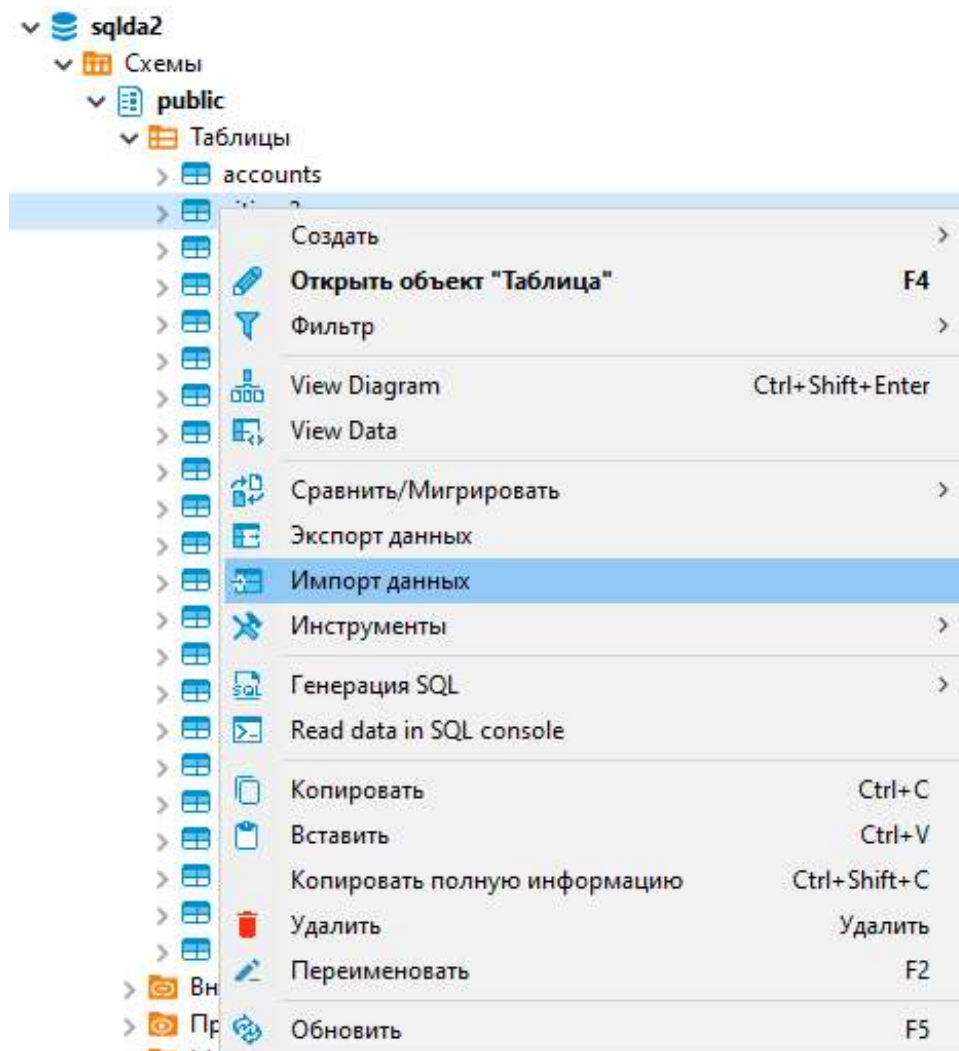
```
CREATE TABLE public.cities_2 (  
  cities_id int4 primary key,  
  city varchar NOT NULL,  
  number_of_customers int4 NOT NULL  
);
```

cities_2	
123 cities_id	int4 NOT NULL
ABC city	varchar NOT NULL
123 number_of_customers	int4 NOT NULL

city	number_of_customers
Washington	1447
Houston	904
New York City	731
El Paso	713
Dallas	607
Atlanta	571
Sacramento	506
Los Angeles	466
San Antonio	426
Miami	426

Вариант 1. Выгрузка данных в таблицу из внешнего хранилища(csv)

2. Загрузить данные в таблицу Postgresql



Вариант 2. Выгрузка данных в таблицу из внешнего хранилища(csv). sqlalchemy

2. Загрузить данные в таблицу PostgreSQL

https://colab.research.google.com/drive/1HLrOzlobYiHIISh_7PCjWfI7QmDPt_dM?usp=sharing

```
1 !pip install sqlalchemy
```

```
1 from google.colab import files
2
3 uploaded = files.upload()
```

```
1 import pandas as pd
2 from sqlalchemy import create_engine
```

```
1 df = pd.read_csv('top_cities.csv')
2 df.columns = [c.lower() for c in df.columns]
```

```
1 df
```

```
1 engine = create_engine('postgresql://login:pass@ip:5432/db')
2 engine.connect()
3 connection = engine.connect()
4 print(engine)
```

```
1 df.to_sql('df', con=engine, if_exists='replace',
2 | | | | index=False)
```

```
1 connection = engine.connect()
2 connection.close()
```

Упражнение 1. Экспорт данных в файл для дальнейшей обработки в Excel



Сохранить файл, содержащий города с наибольшим количеством клиентов компании. Этот анализ поможет исполнительному комитету компании решить, где они могут открыть следующий дилерский центр.

Упражнение 1. Экспорт данных в файл для дальнейшей обработки в Excel



Решение

1. Откройте инструмент командной строки, чтобы выполнить это упражнение, например **cmd** для Windows или **Terminal** для Mac.
2. В интерфейсе командной строки подключитесь к базе данных с помощью команды **psql**.
3. Скопируйте таблицу клиентов из вашей базы данных в локальный файл в формате .csv.

Упражнение 1. Экспорт данных в файл для дальнейшей обработки в Excel

Решение

```
CREATE TEMP VIEW top_cities AS (  
  SELECT city,  
    count(1) AS number_of_customers  
  FROM customers  
  WHERE city IS NOT NULL  
  GROUP BY 1  
  ORDER BY 2 DESC  
  LIMIT 10  
);
```

```
\copy top_cities TO 'top_cities.csv' WITH CSV HEADER DELIMITER ','
```

```
DROP VIEW top_cities;
```

Упражнение 1. Экспорт данных в файл для дальнейшей обработки в Excel

Решение

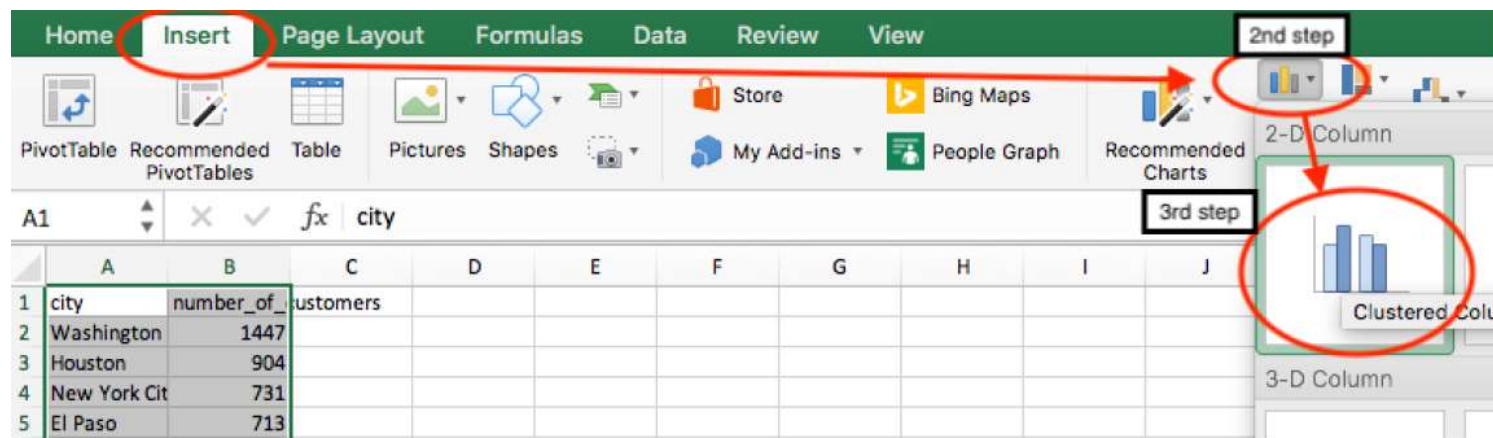
С помощью Microsoft Excel или текстового редактора откройте файл **top_cities.csv**:

A	B	C
city	number_of_customers	
Washington	1447	
Houston	904	
New York Cit	731	
El Paso	713	
Dallas	607	
Atlanta	571	
Sacramento	506	
Los Angeles	466	
San Antonio	426	
Miami	426	

Упражнение 1. Экспорт данных в файл для дальнейшей обработки в Excel

Решение

Затем в верхнем меню выберите «Вставка», а затем щелкните значок гистограммы (), чтобы создать двухмерную гистограмму:



Упражнение 1. Экспорт данных в файл для дальнейшей обработки в Excel

Решение

Затем в верхнем меню выберите «Вставка», а затем щелкните значок гистограммы (), чтобы создать двухмерную гистограмму:

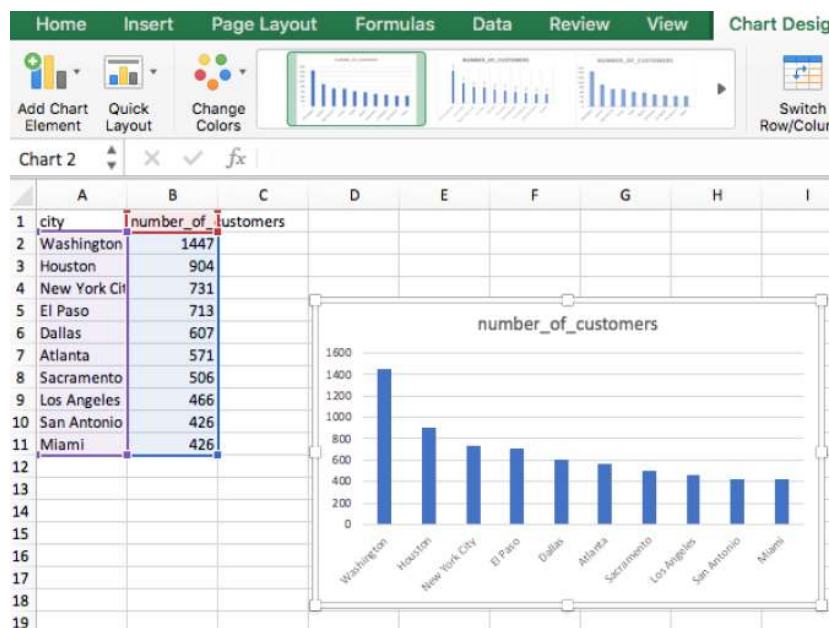


Figure 6.8: Final output from our visualization