

**МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА – Российский технологический университет»**

|  |
| --- |
| **РТУ МИРЭА** |
|  |
| **Институт кибербезопасности и цифровых технологий (ИКБ)** |
|  |
| КБ-2 «Информационно-аналитические системы кибербезопасности» |

**ОТЧЕТ О ВЫПОЛНЕНИИ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ №2**

# **В РАМКАХ ДИСЦИПЛИНЫ «ТЕХНОЛОГИИ ХРАНЕНИЯ В СИСТЕМАХ КИБЕРБЕЗОПАСНОСТИ»**

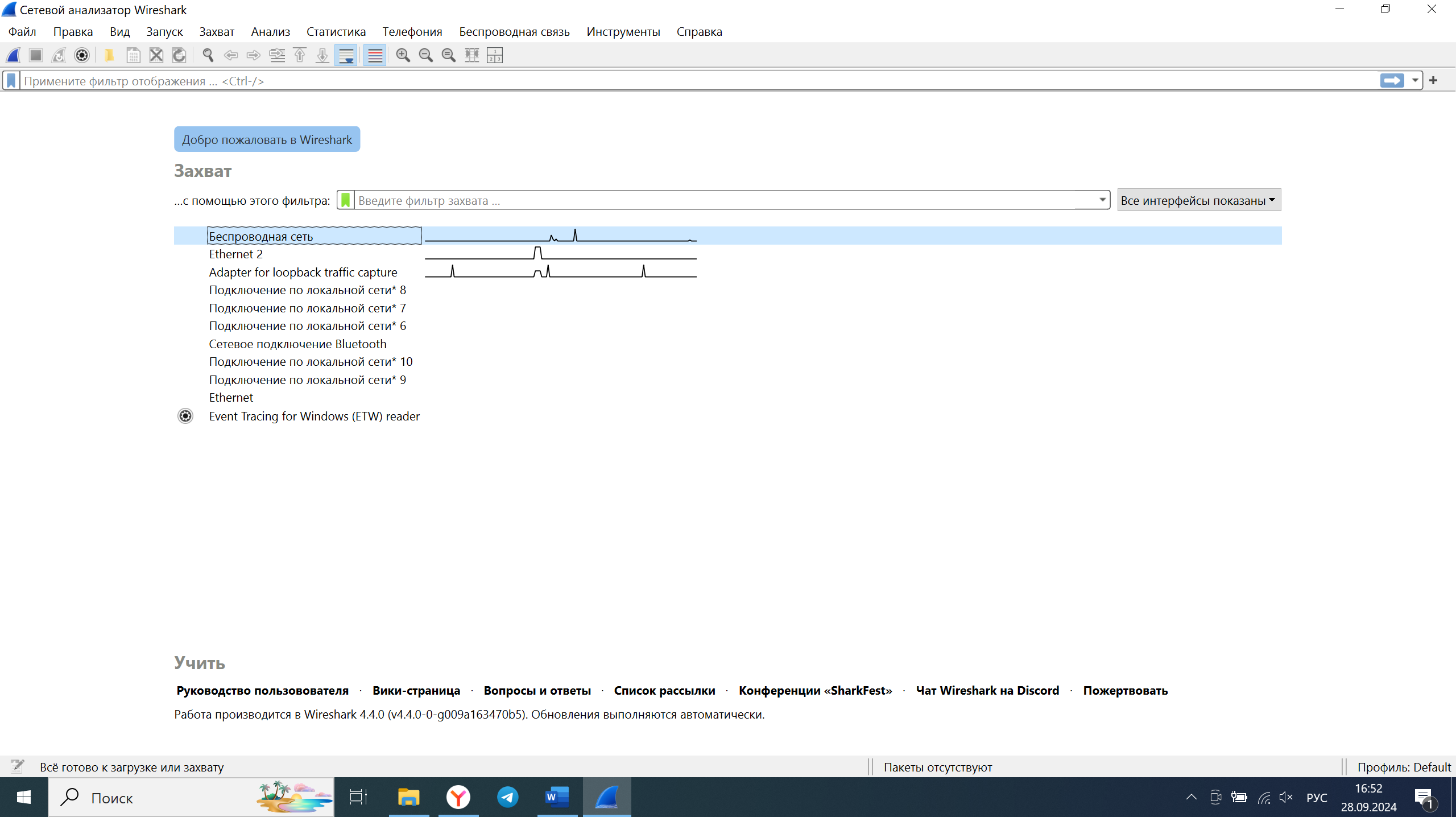
Выполнил:

Студент 3-ого курса

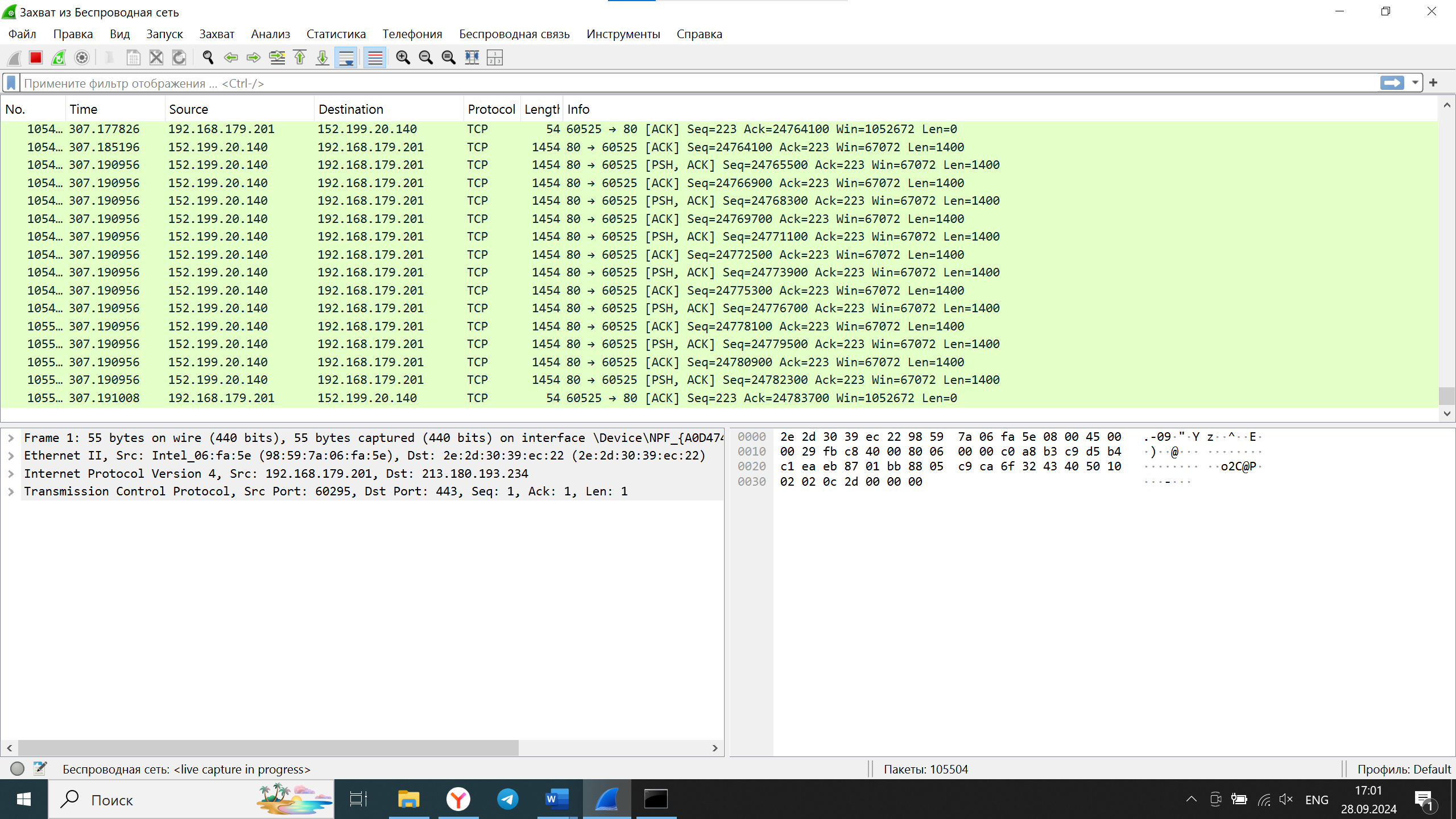
Учебной группы БИСО-02-22

Зубарев В.С.

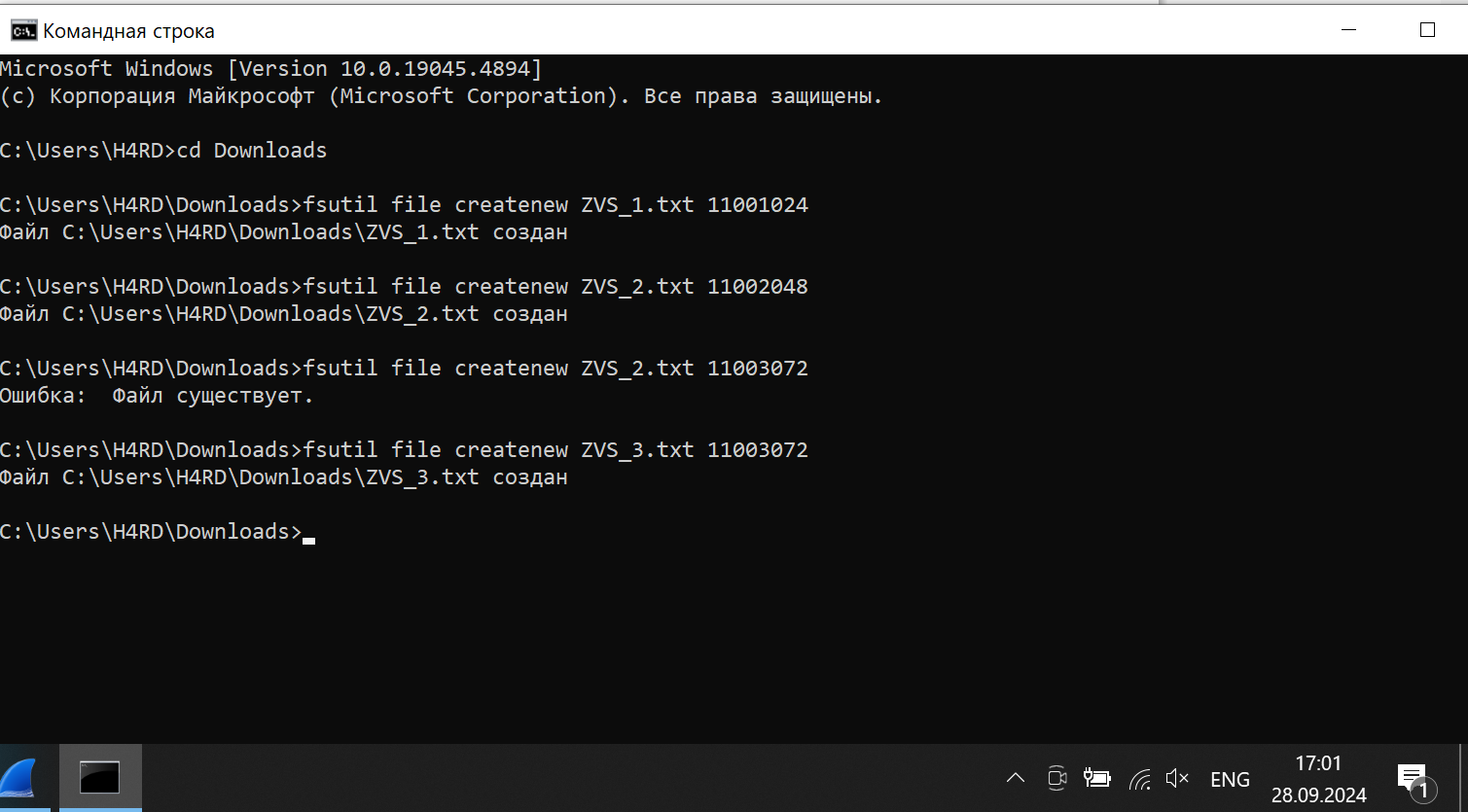
1. Установите и запустите ПО «Wireshark»



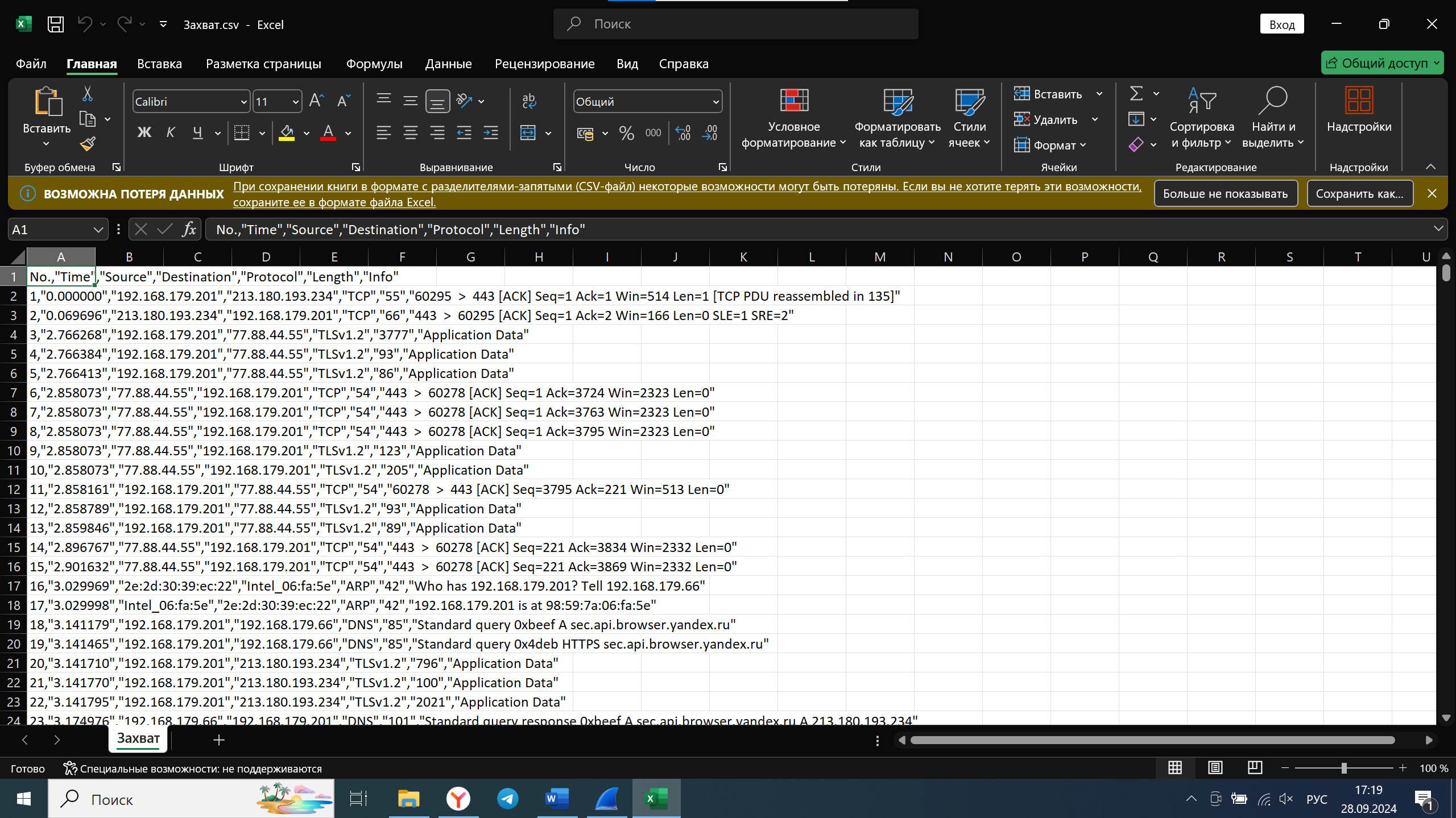
1. Запустите захват пакетов с интерфейса, который используется для выхода в сеть Интернет.



1. Создайте несколько файлов (минимум 3) размером 11000000 + (номер по списку \* 1 КБ + N) байт, где N – номер файла. Имя файлов должно иметь следующий вид: FIO\_N.txt, где F – первая буква фамилии на латинице; I – первая буква имени на латинице; O – первая буква отчества на латинице (при наличии), N – номер файла (1, 2, 3, …).

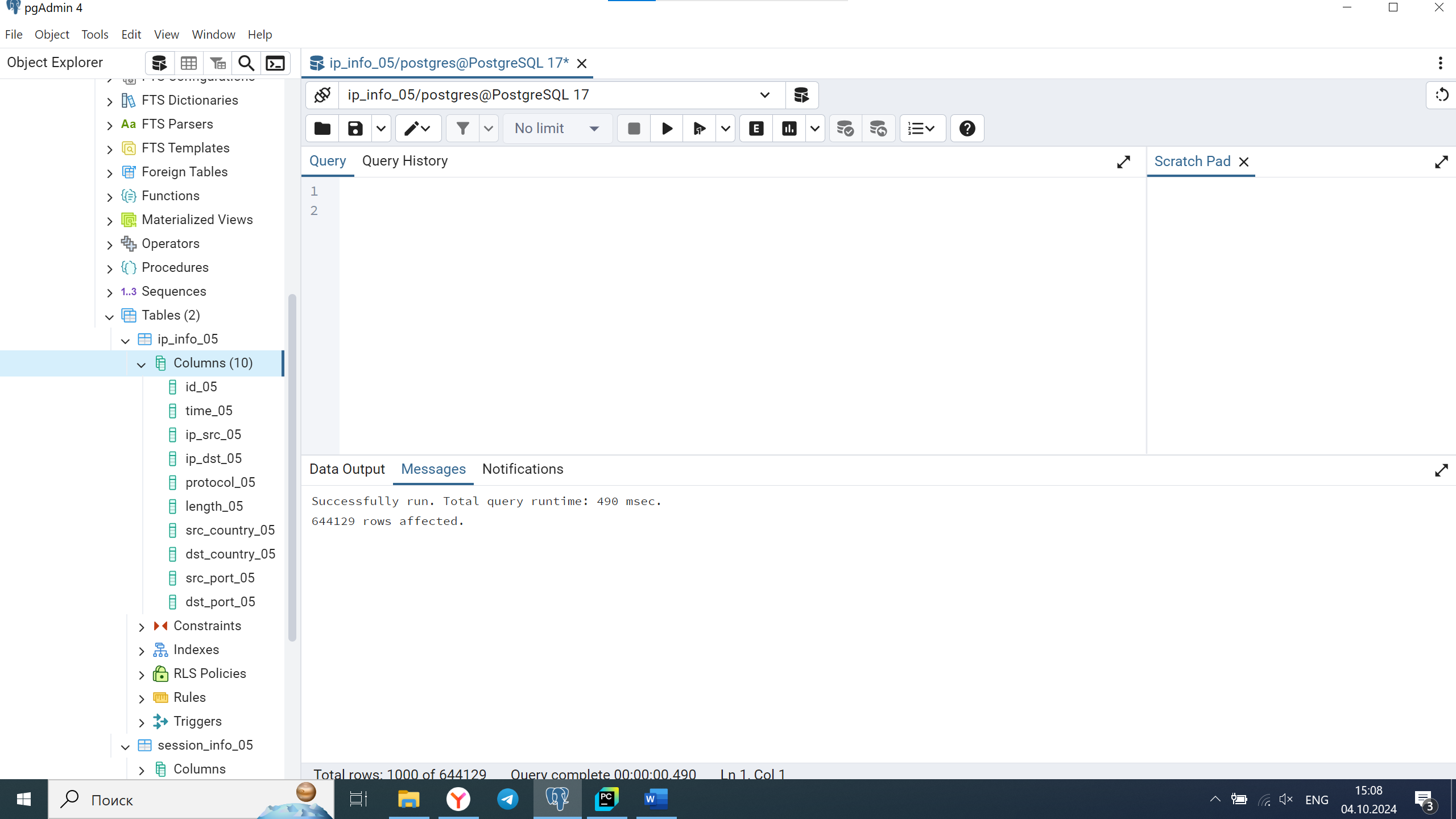


1. Остановите захват пакетов и экспортируйте информацию о сетевых пакетах в формате CSV и JSON. Проанализируйте структуру файлов.



1. Создайте базу данных в PostgreSQL с именем ip\_info\_XX (XX – порядковый номер студента в группе).

Создайте таблицу для хранения информации об IP-соединениях. В Приложении 2 приведен пример создания таблицы в pgAdmin. Минимальный набор полей: IPадрес отправителя, IP-адрес получателя, страна адреса отправителя, страна адреса получателя, порт отправителя, порт получателя, протокол, размер пакета (сессии). К названию каждого поля добавляем «\_XX» (XX – порядковый номер студента в группе). Например: ip\_src\_07, ip\_dst\_07, country\_from\_07, country\_to\_07, length\_07 и т.д



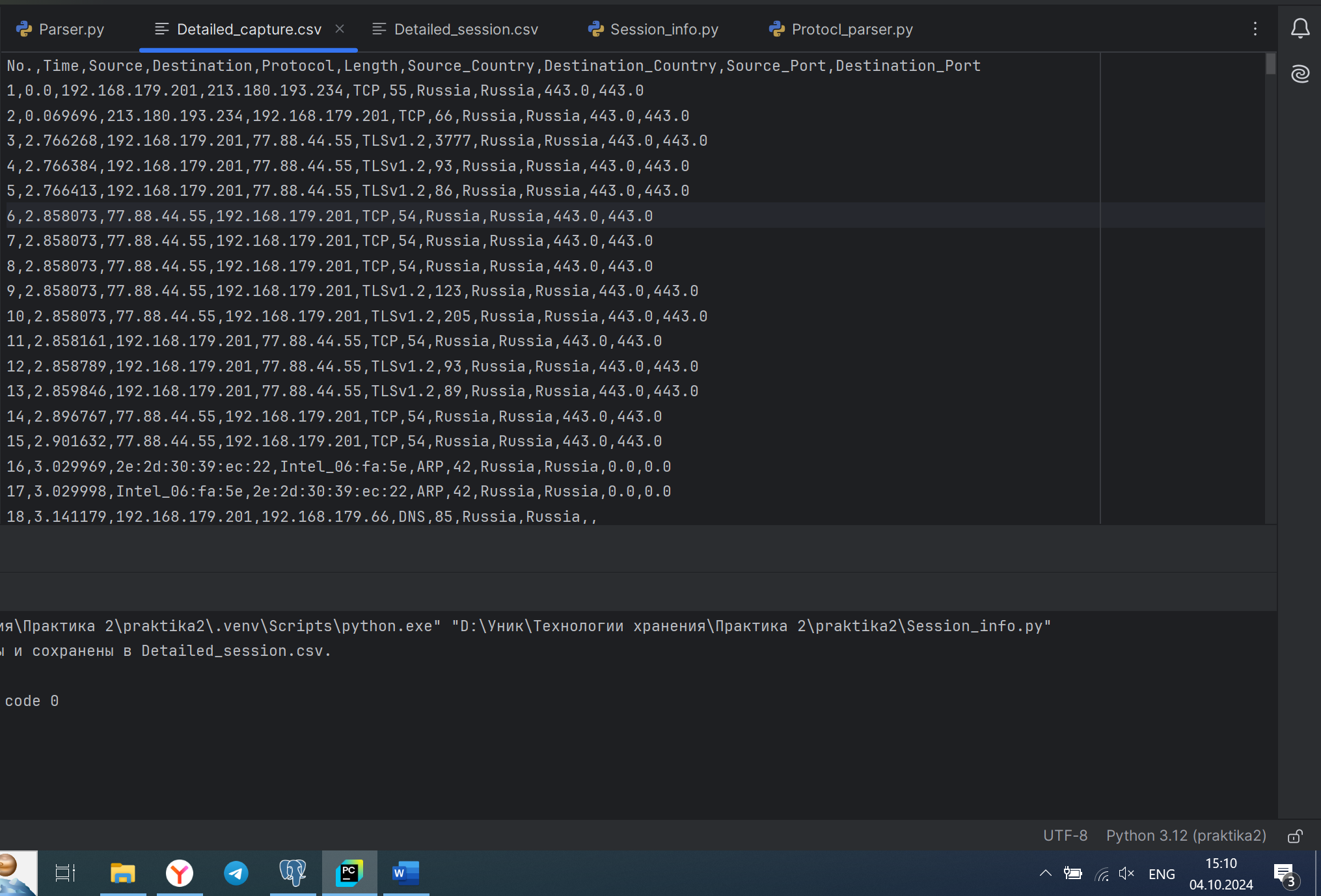
1. Напишите программу/скрипт для парсинга информации об IP-соединениях из ранее сохраненного файла CSV или JSON. Определитесь, когда будете вычислять информацию о размере сессии: перед загрузкой в БД или после. Для вычисления размера сессии необходимо просуммировать размеры пакетов, относящиеся к одной сессии. В программе/скрипте необходимо реализовать обогащение IPадресов информацией (как минимум о стране).

Для реализации обогащения ip адресов был написана программа на языке python.

Листинг программы:

import geoip2.database  
import pandas as pd  
  
# Путь к базе данных GeoLite2  
geoip\_db\_path = 'GeoLite2-Country.mmdb'  
  
# Путь к входному CSV файлу с данными Wireshark  
input\_csv\_path = 'Capture.csv'  
  
# Путь к выходному CSV файлу с добавленной информацией о стране и портах  
output\_csv\_path = 'Detailed\_capture.csv'  
  
  
# Функция для получения информации о стране по IP адресу  
def get\_country\_by\_ip(ip\_address, reader):  
 try:  
 response = reader.country(ip\_address)  
 return response.country.name  
 except Exception:  
 return "Russia"  
  
  
# Функция для получения порта по протоколу  
def get\_ports\_by\_protocol(protocol):  
 protocol\_ports = {  
 'TCP': (443,443), # HTTPS (443)  
 'ARP':(0,0), #Протокол канального уровня  
 'TLSv1.2': (443, 443), # TLS работает на 443 порту  
 'SSDP': (1900,1900), #1900(UDP)  
 'TLSv1.3': (443, 443), # TLS работает на 443 порту  
 'MDNS': (5353,5353), #UDP  
 'QUIC': (443,443), #UDP  
 'SSLv2': (443,443), #443(HTTPS)  
 'UDP': (53, 53), # DNS обычно использует UDP на порту 53  
 }  
 return protocol\_ports.get(protocol, (None, None)) # Если протокол неизвестен, вернем None  
  
  
# Открываем базу данных GeoLite2  
with geoip2.database.Reader(geoip\_db\_path) as reader:  
 # Читаем данные из CSV файла  
 df = pd.read\_csv(input\_csv\_path)  
  
 # Добавляем информацию о стране для Source и Destination IP-адресов  
 df['Source\_Country'] = df['Source'].apply(lambda x: get\_country\_by\_ip(x, reader))  
 df['Destination\_Country'] = df['Destination'].apply(lambda x: get\_country\_by\_ip(x, reader))  
  
 # Добавляем информацию о портах, исходя из протокола  
 df[['Source\_Port', 'Destination\_Port']] = df['Protocol'].apply(lambda x: pd.Series(get\_ports\_by\_protocol(x)))  
  
# Сохраняем результат в новый CSV файл  
df.to\_csv(output\_csv\_path, index=False)  
  
print(f"Данные с информацией о стране и портах успешно сохранены в {output\_csv\_path}")

Результат работы (чтение полученного файла):

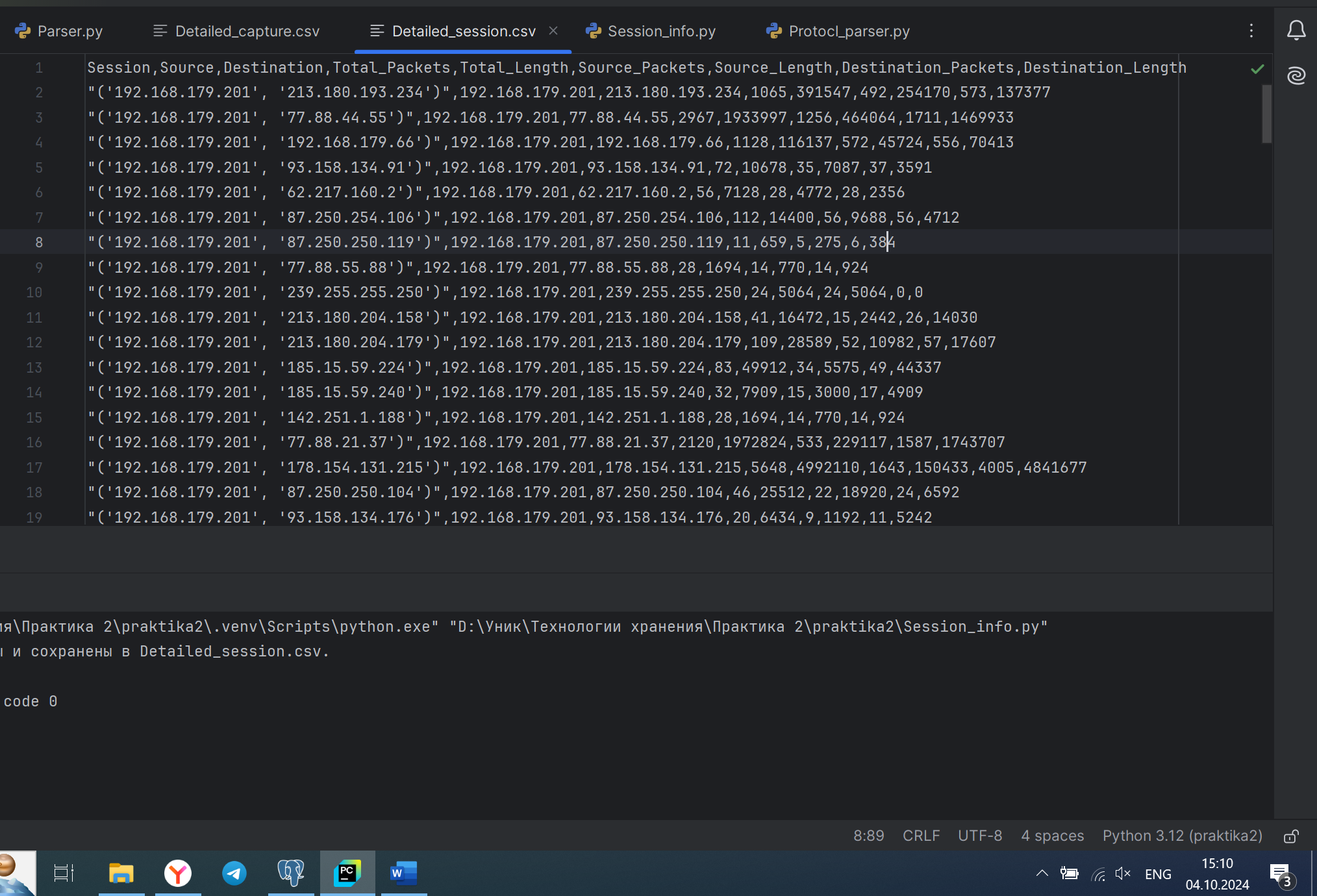


Для подсчета размера сессии была написана вторая программа на языке python

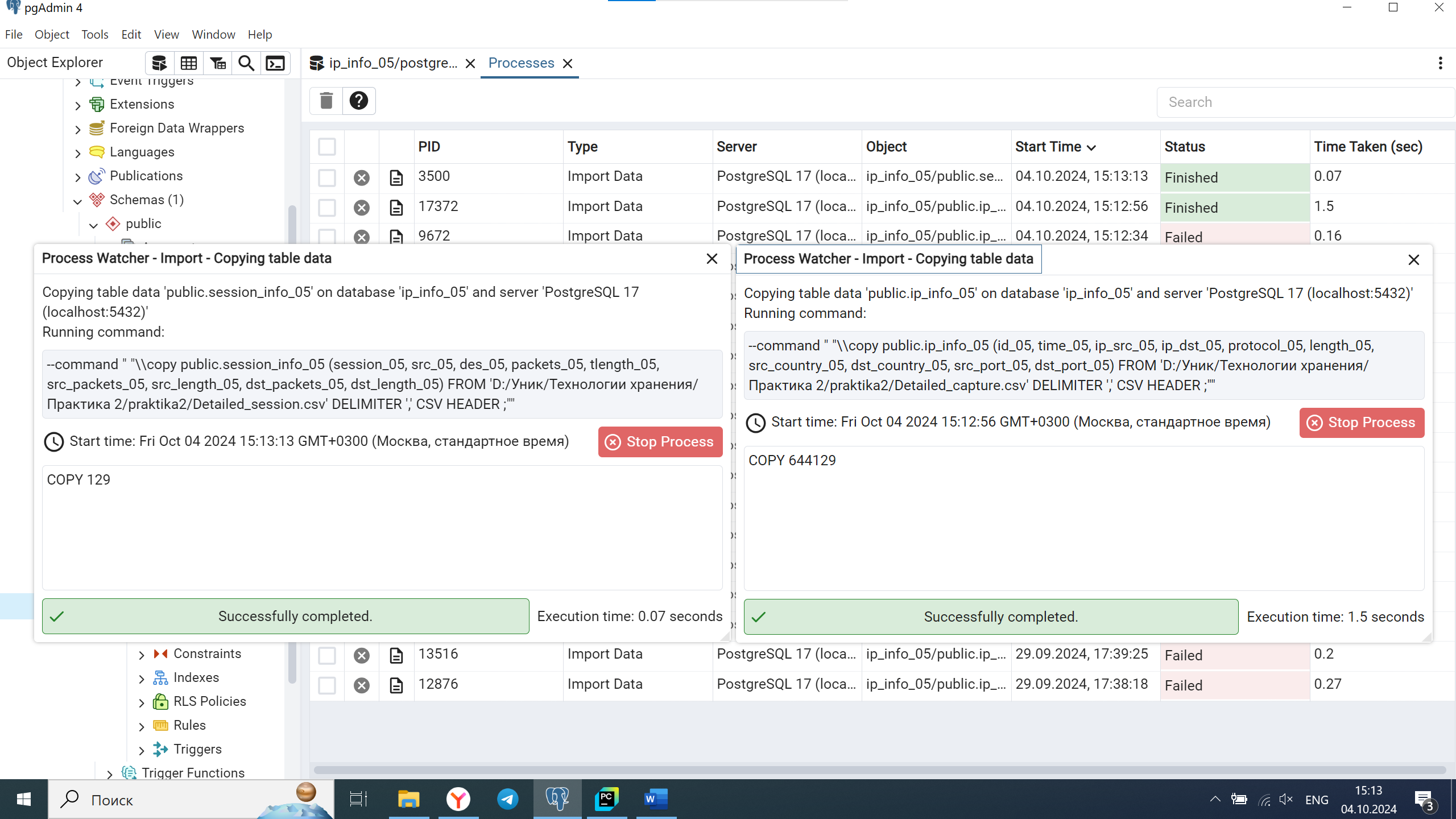
Листинг программы:

import pandas as pd  
  
# IP-адрес хоста, который всегда должен быть Source  
host\_ip = "192.168.179.201"  
  
# Чтение исходного CSV файла  
input\_csv\_path = 'Detailed\_capture.csv' # Укажите путь к вашему файлу  
df = pd.read\_csv(input\_csv\_path)  
  
# Создаем словарь для хранения информации о сессиях  
sessions = {}  
  
# Проходим по каждой строке исходного файла  
for index, row in df.iterrows():  
 src = row['Source']  
 dst = row['Destination']  
 length = int(row['Length'])  
 # Определяем, кто из них является хостом (192.168.179.201)  
 if src == host\_ip:  
 # Хост отправляет данные, сервер принимает  
 session\_key = (src, dst)  
 source\_packets = 1  
 source\_length = length  
 destination\_packets = 0  
 destination\_length = 0  
 elif dst == host\_ip:  
 # Сервер отправляет данные, хост принимает  
 session\_key = (dst, src)  
 source\_packets = 0  
 source\_length = 0  
 destination\_packets = 1  
 destination\_length = length  
 else:  
 # Если ни один из IP не является хостом, пропускаем строку  
 continue  
  
 # Обновляем информацию о сессии  
 if session\_key not in sessions:  
 sessions[session\_key] = {  
 'Source\_Packets': source\_packets,  
 'Source\_Length': source\_length,  
 'Destination\_Packets': destination\_packets,  
 'Destination\_Length': destination\_length,  
 }  
 else:  
 sessions[session\_key]['Source\_Packets'] += source\_packets  
 sessions[session\_key]['Source\_Length'] += source\_length  
 sessions[session\_key]['Destination\_Packets'] += destination\_packets  
 sessions[session\_key]['Destination\_Length'] += destination\_length  
  
# Формируем данные для записи в CSV  
session\_data = []  
for session\_key, data in sessions.items():  
 session\_data.append({  
 'Session': f"{session\_key}",  
 'Source': session\_key[0], # Хост (192.168.179.201)  
 'Destination': session\_key[1], # Сервер  
 'Total\_Packets': data['Source\_Packets'] + data['Destination\_Packets'],  
 'Total\_Length': data['Source\_Length'] + data['Destination\_Length'],  
 'Source\_Packets': data['Source\_Packets'],  
 'Source\_Length': data['Source\_Length'],  
 'Destination\_Packets': data['Destination\_Packets'],  
 'Destination\_Length': data['Destination\_Length']  
 })  
  
# Создаем DataFrame с результатами  
output\_df = pd.DataFrame(session\_data)  
  
# Сохраняем результат в новый CSV файл  
output\_csv\_path = 'Detailed\_session.csv' # Укажите путь для сохранения файла  
output\_df.to\_csv(output\_csv\_path, index=False)  
  
print(f"Сессии успешно сформированы и сохранены в {output\_csv\_path}.")

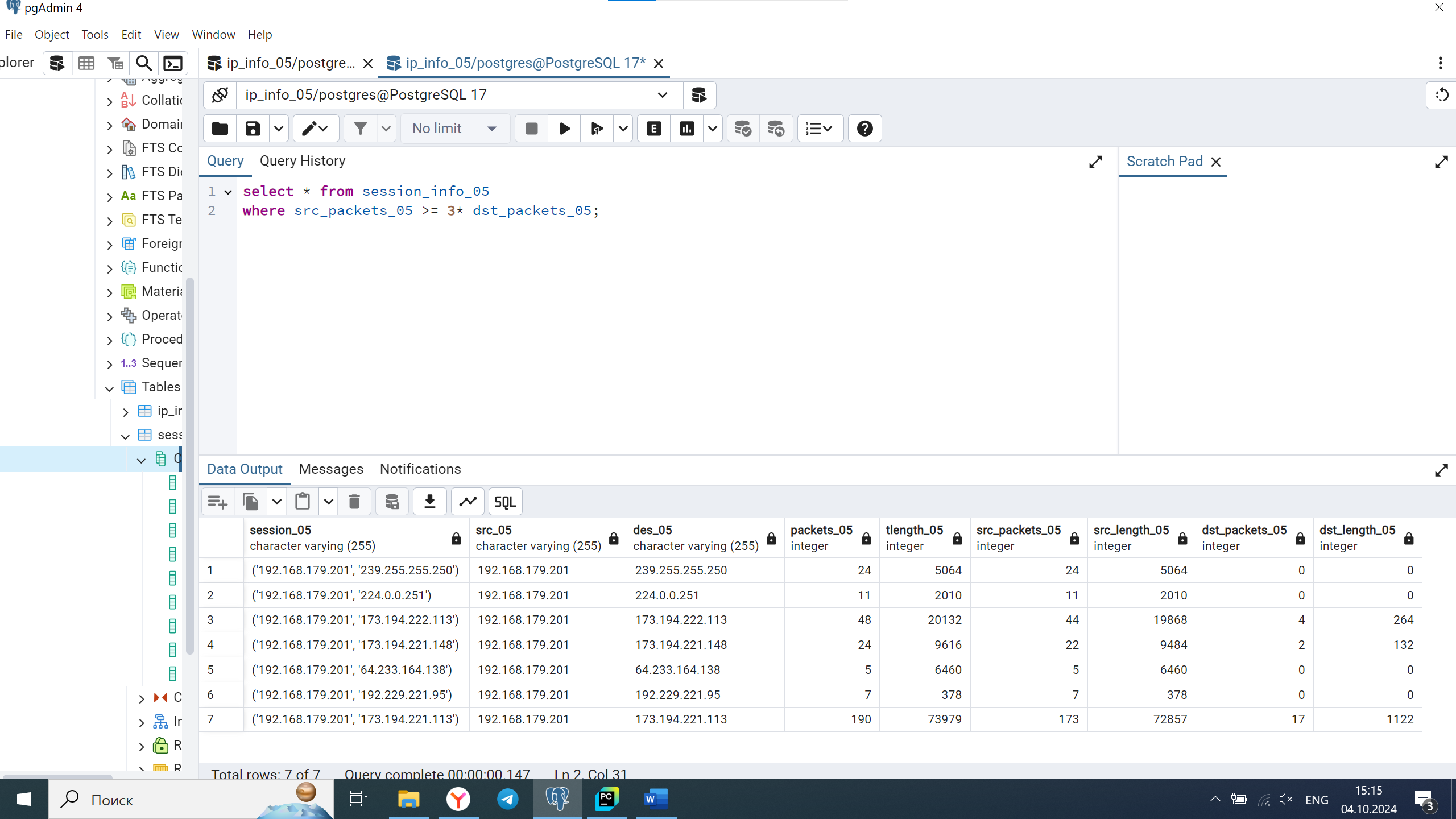
Результат работы программы (чтение файла):

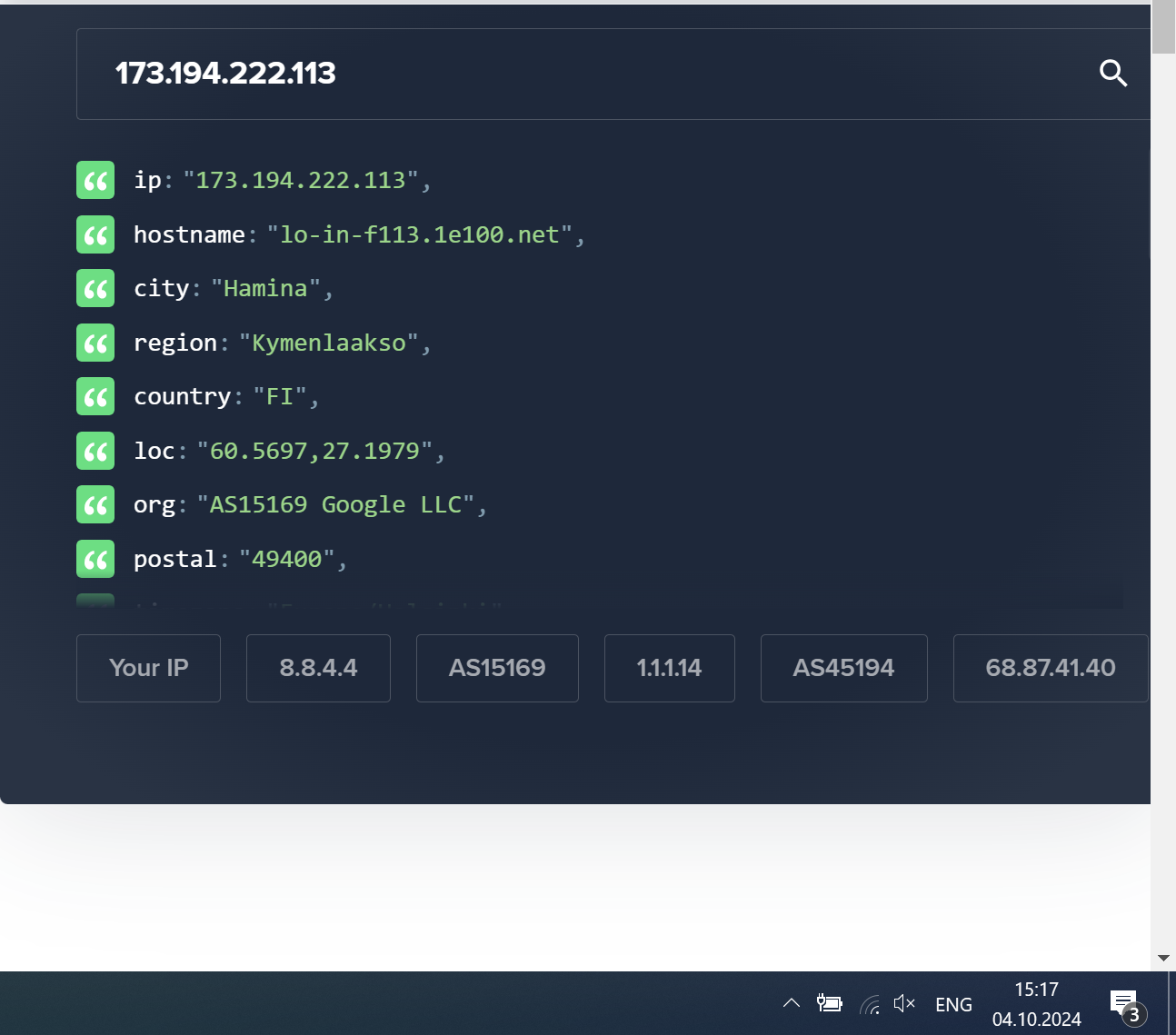


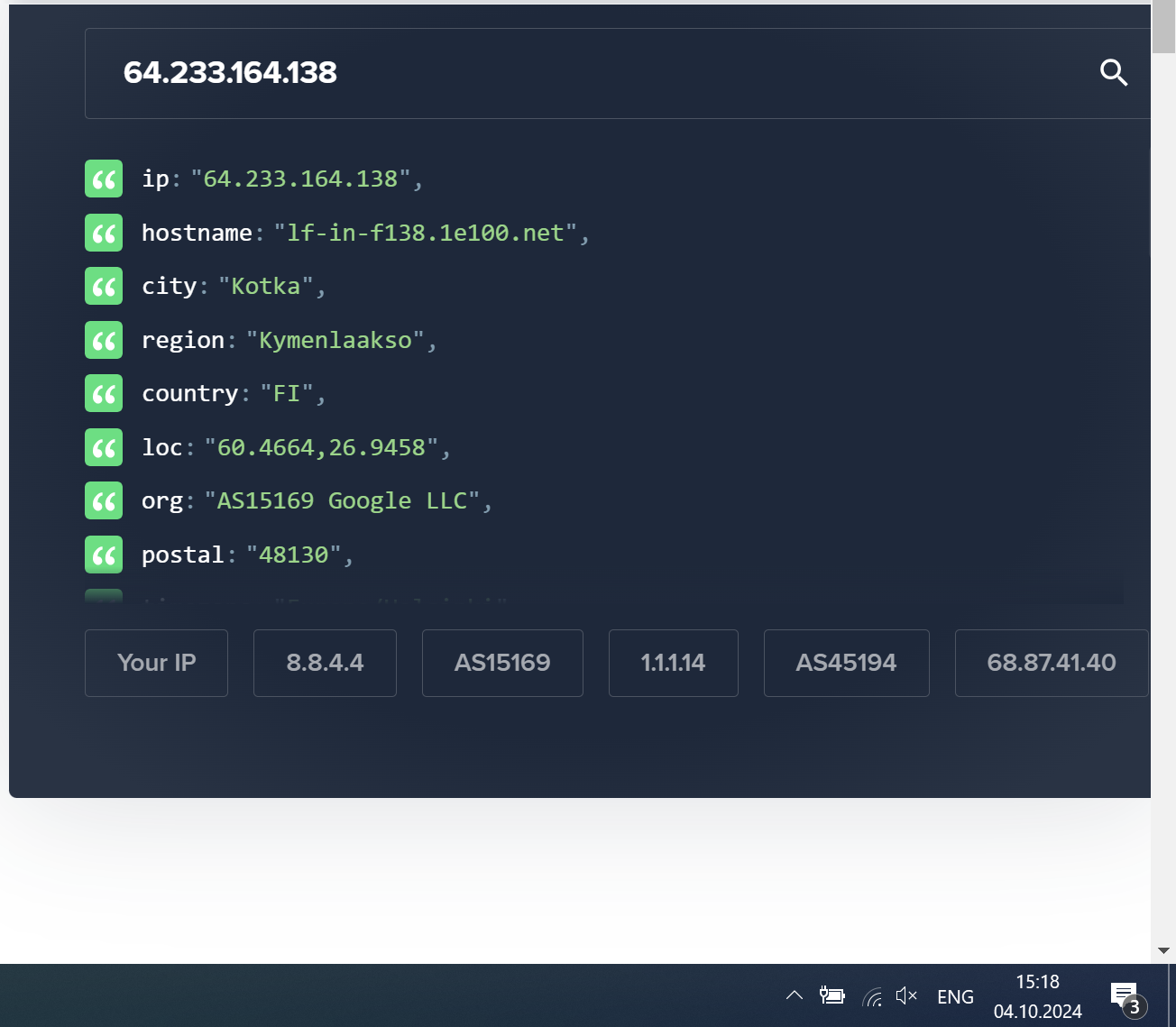
1. Загрузите данные в БД.



1. Напишите программу/скрипт для вычисления потенциальных утечек информации на основе собранных данных (объем данных, передающихся на сервер, во много раз больше объема данных, полученных от сервера). Соберите информацию о выявленных IP-адресах серверов (например, с сайта <https://ipinfo.io>).

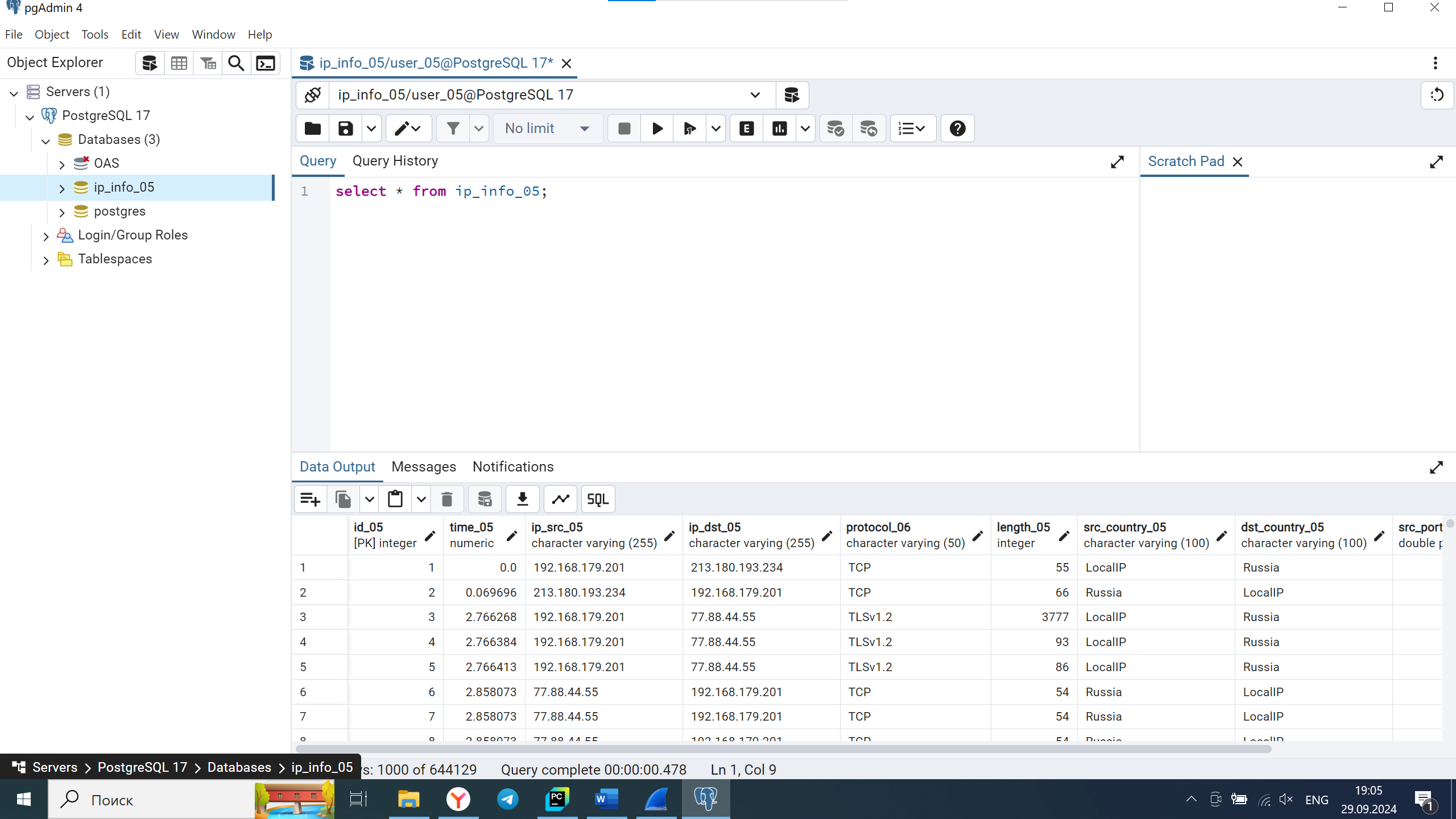






1. Создайте пользователя user\_XX (XX – порядковый номер студента в группе), имеющего права только на чтение данных. Авторизуйтесь под созданным пользователем. Выполните запросы на чтение и удаление/изменение данных.

Чтение данных



Изменение данных

