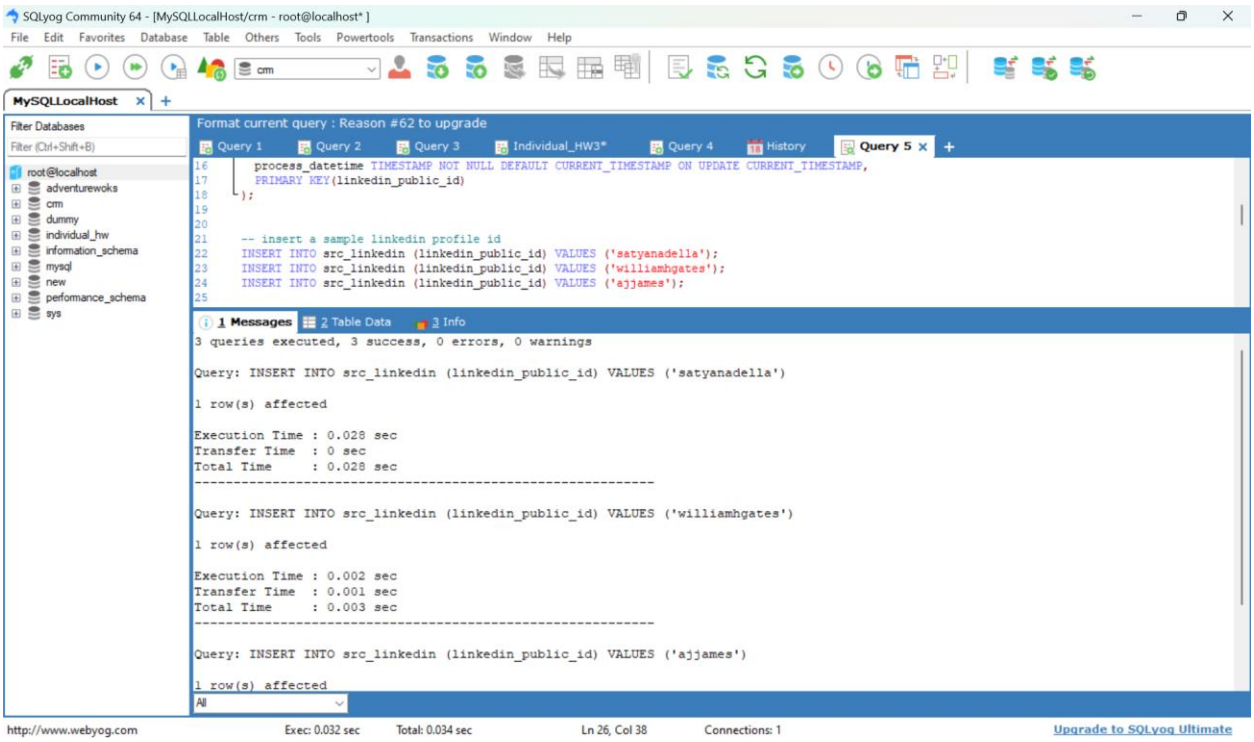
Group Project 1 Group No: 08
Members: Apurva Patil, Preeti Sonke, Durga Sai Sri Bommi

Ingesting LinkedIn Profile

Step 1:

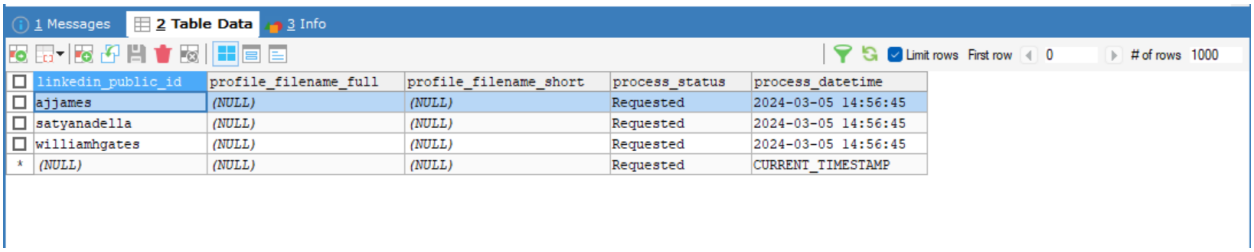
The SQL script initializes a database named **crm** for managing LinkedIn profiles, deleting any existing **crm** database first. It then creates the **crm** database with the **utf8mb4** character set and selects it for use. Four tables are created: **src_linkedin** to store LinkedIn profile IDs and associated filenames; **profile_section** to categorize different LinkedIn profile sections; **profile_section_log** to log processing statuses for profile sections; and **profile_photo_url** to store URLs of profile photos. Sample data is inserted into **src_linkedin** and **profile_section**. The script establishes relationships between tables using primary and foreign keys, ensuring data integrity. This setup is crucial for managing and tracking the ETL (Extract, Transform, Load) processes related to LinkedIn profile data within the project.

Inserted 3 LinkedIn profile Id's into the table src_linkedin:



Observation 1:

The table displays LinkedIn profiles that are marked with the status "Requested," indicating that an action, indicating the start of the ETL process, has been initiated for them.



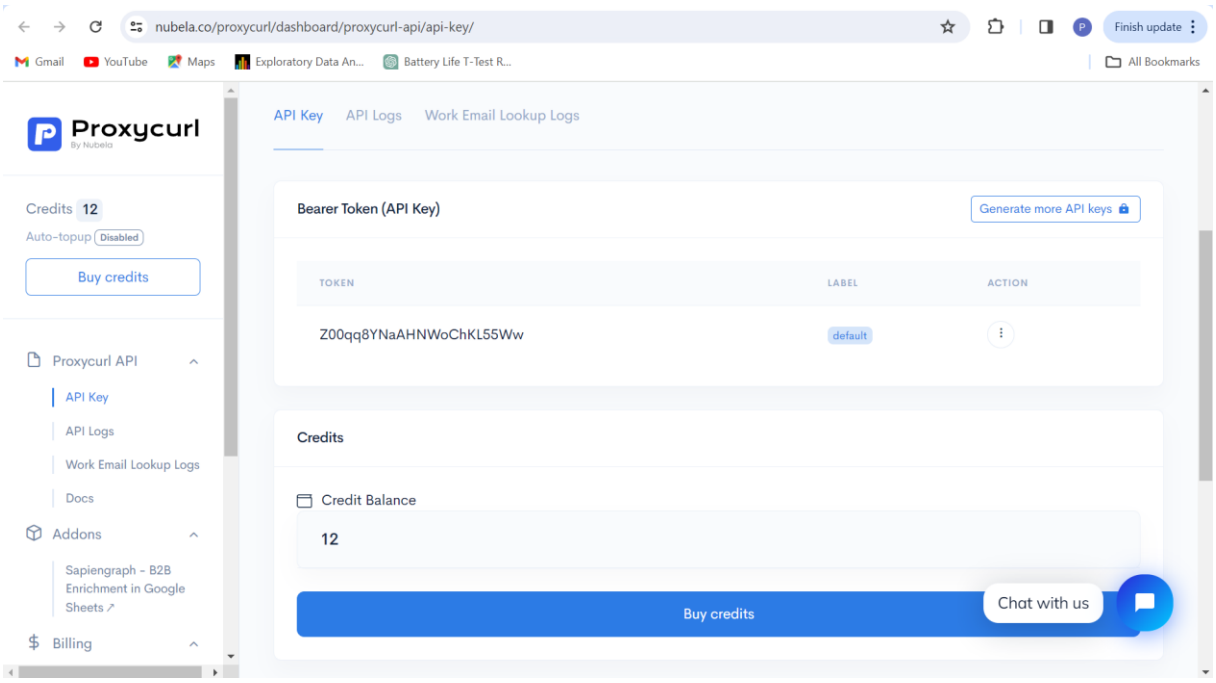
1 Messages | 2 Table Data | 3 Info

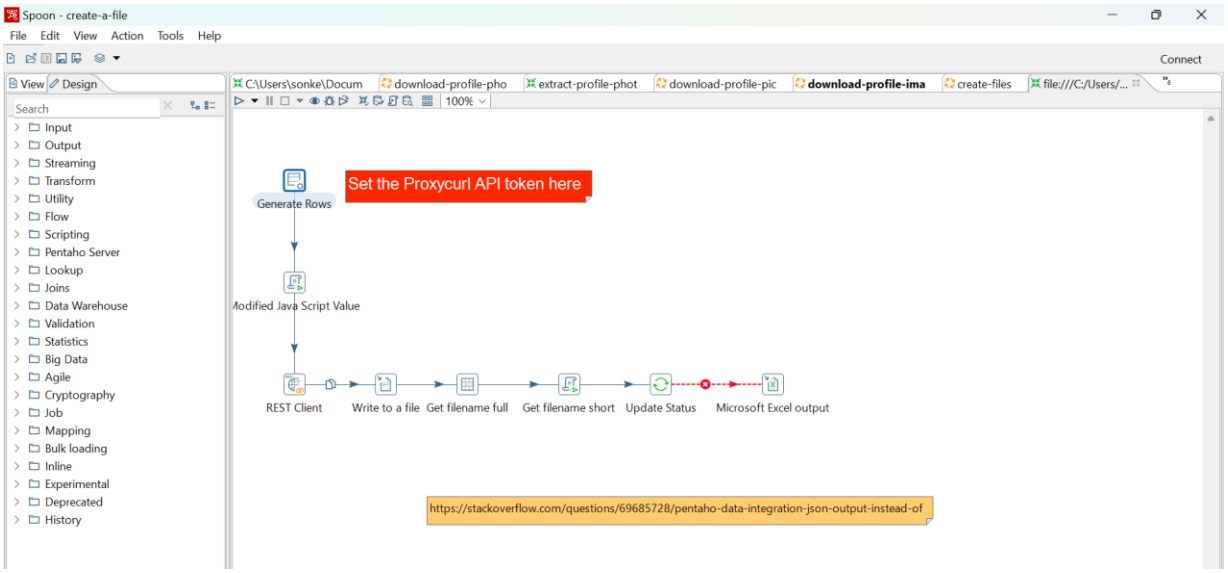
Limit rows First row 0 # of rows 1000

linkedin_public_id	profile_filename_full	profile_filename_short	process_status	process_datetime
ajjames	(NULL)	(NULL)	Requested	2024-03-05 14:56:45
satyanadella	(NULL)	(NULL)	Requested	2024-03-05 14:56:45
williamhgates	(NULL)	(NULL)	Requested	2024-03-05 14:56:45
* (NULL)	(NULL)	(NULL)	Requested	CURRENT_TIMESTAMP

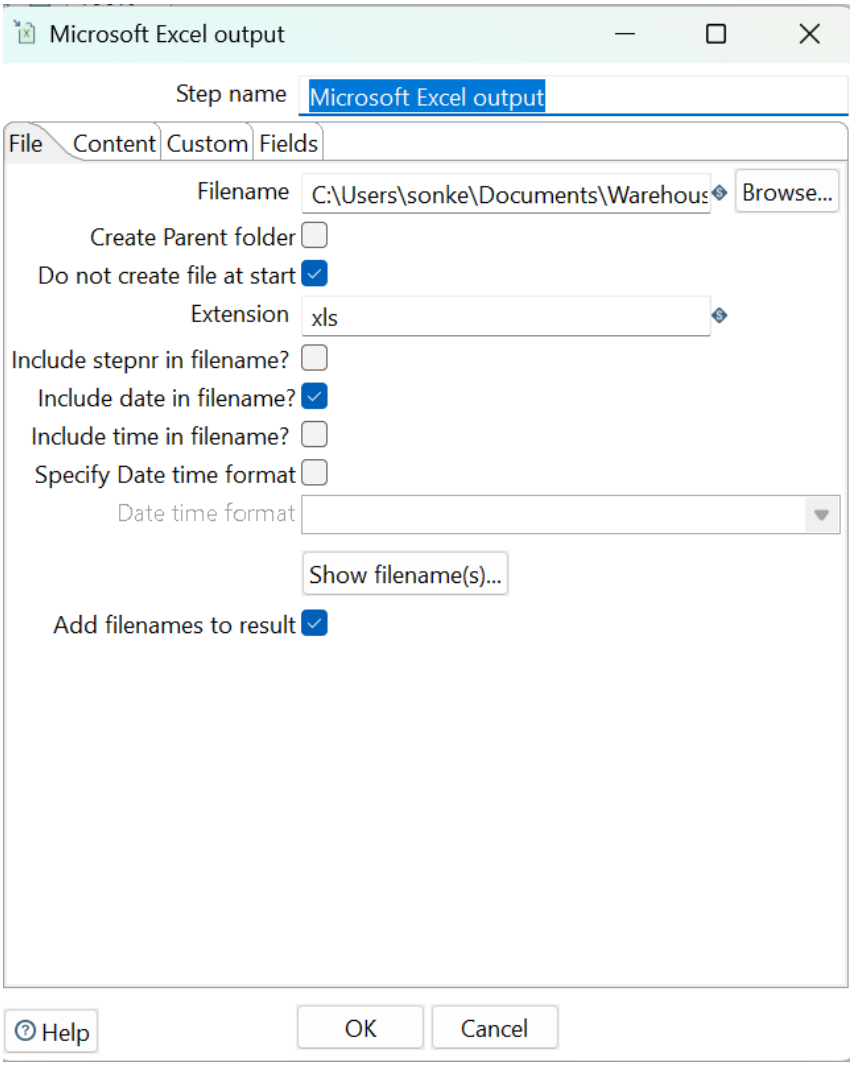
Step 2:

This is Proxycurl, where we can see that we have 12 credits available. Additionally, there's a token provided, which we need to use to access LinkedIn data through Proxycurl.

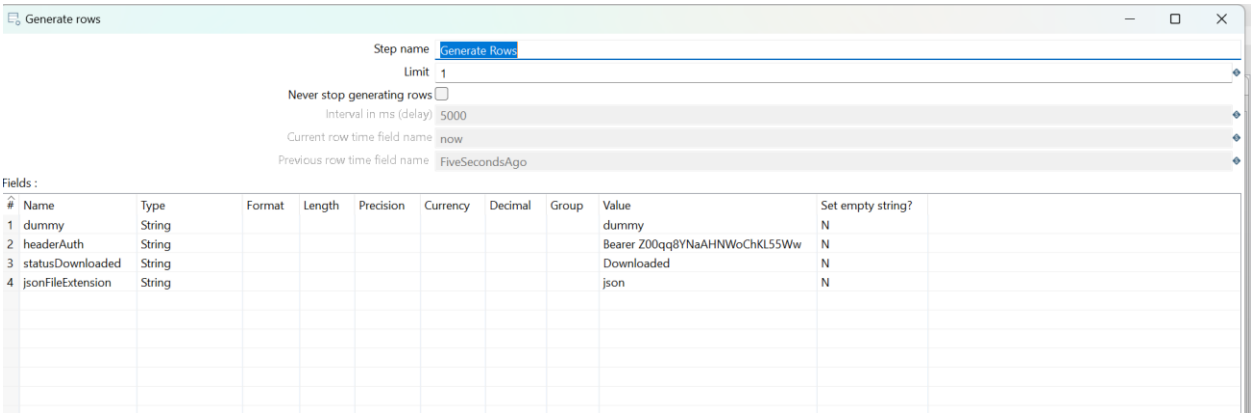




Here we have also provided the path where the logs will change and save.

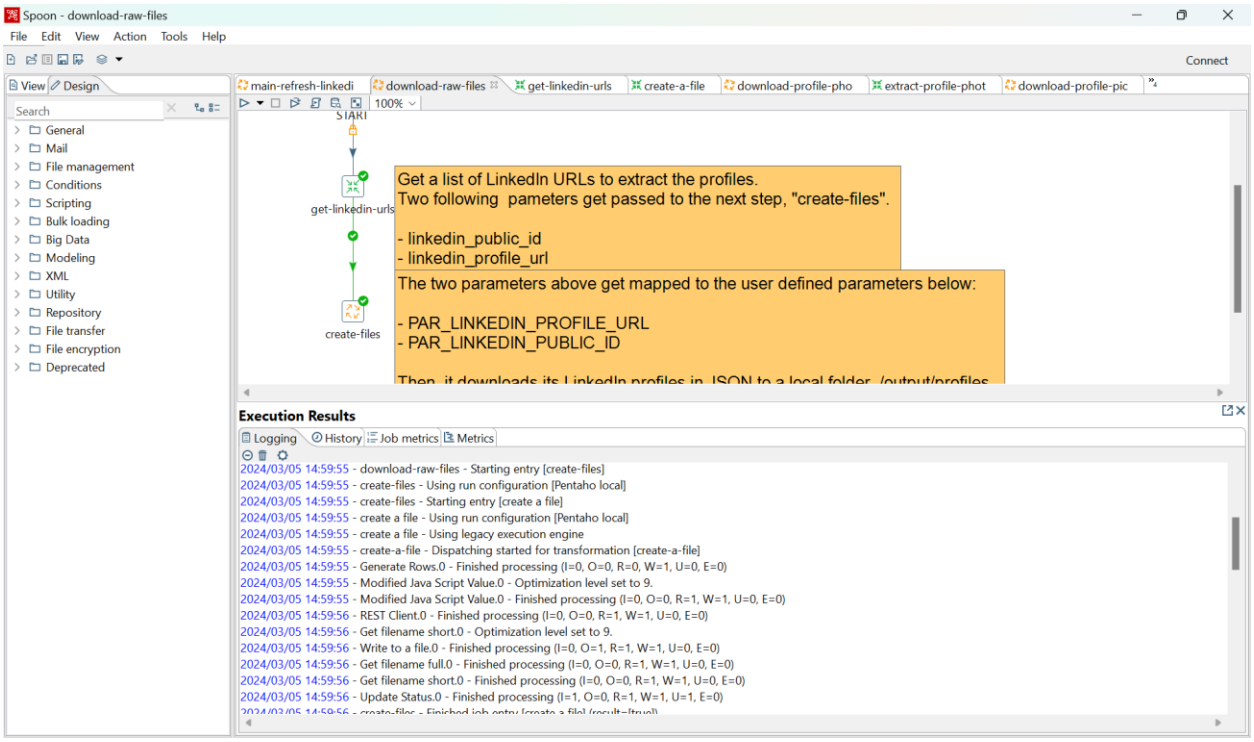


Here we have provided that token.



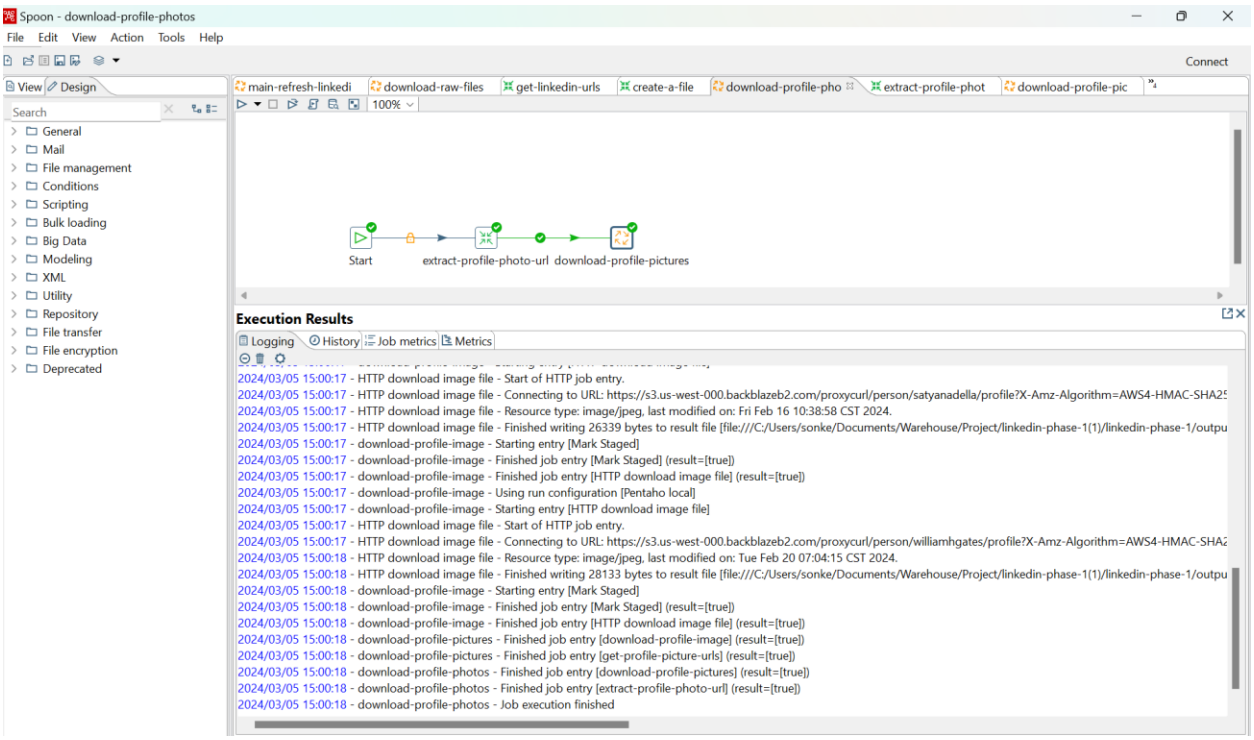
Step 3:

The process starts by obtaining a list of LinkedIn URLs to extract profiles using the "get-linkedin-urls" step. Then, it passes two specific parameters, the LinkedIn public ID and profile URL, to the next step called "create-files". The final action mentioned is downloading LinkedIn profiles in JSON format to a local folder, likely as part of organizing and storing the data extracted from LinkedIn.



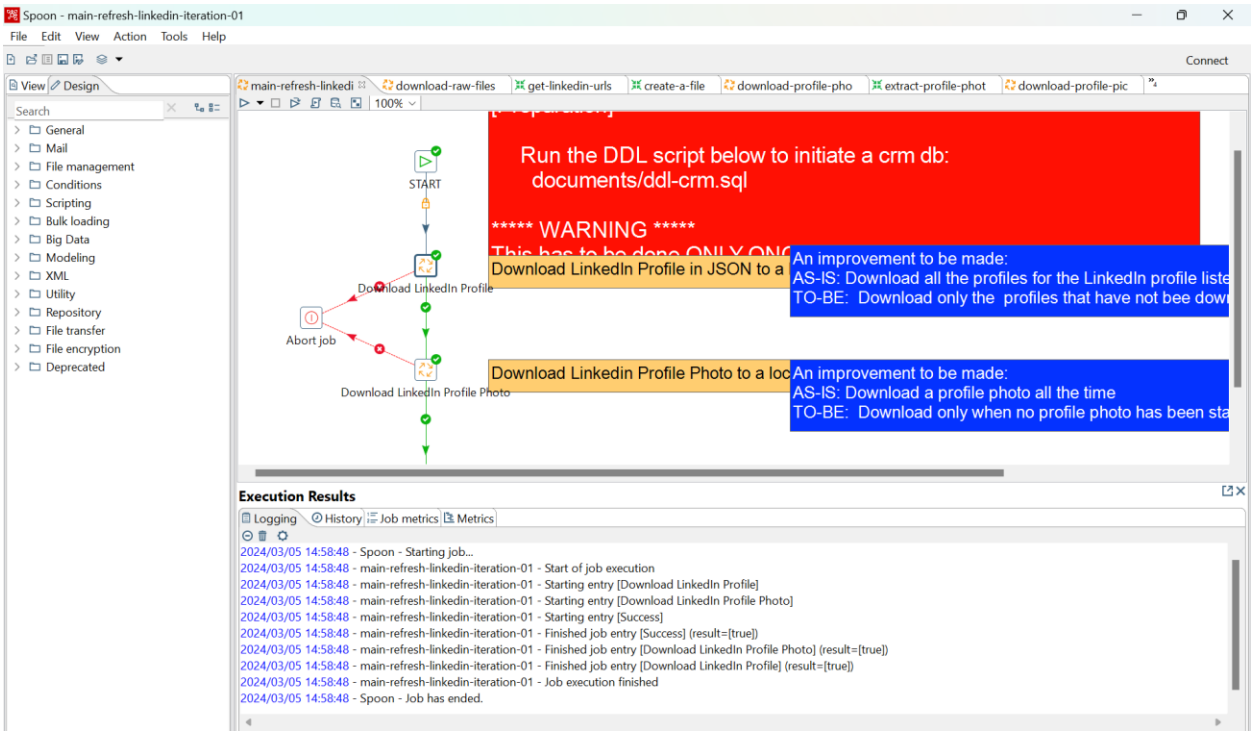
Step 4:

The process includes a step called "extract-profile-photo-url" followed by "download-profile-pictures," which suggests that the tool is extracting URLs of LinkedIn profile photos and then downloading those images. The log indicates successful HTTP requests and file downloads, showing that LinkedIn profile pictures have been retrieved and saved to a specified local directory.



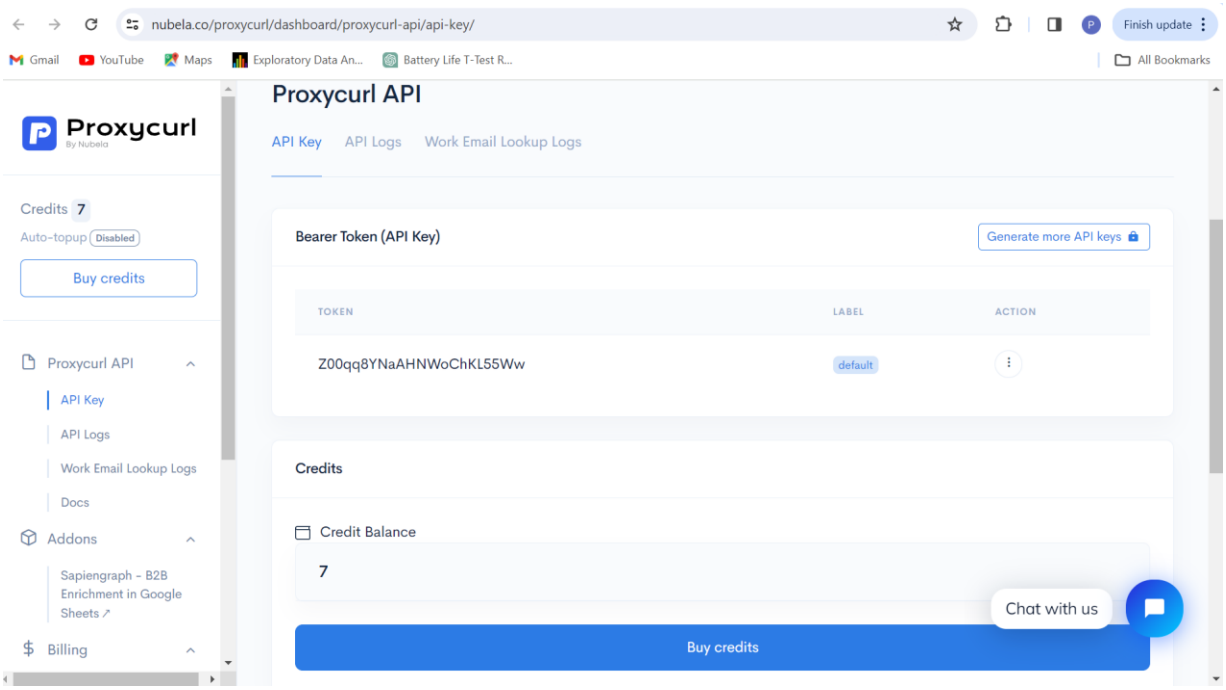
Step 5:

In this process, an ETL job is being executed which consists of downloading LinkedIn profiles and their corresponding profile photos.



Observation 2:

Based on the previous mention of having 12 credits, the status showing 7 credits suggests that 5 credits have been used.

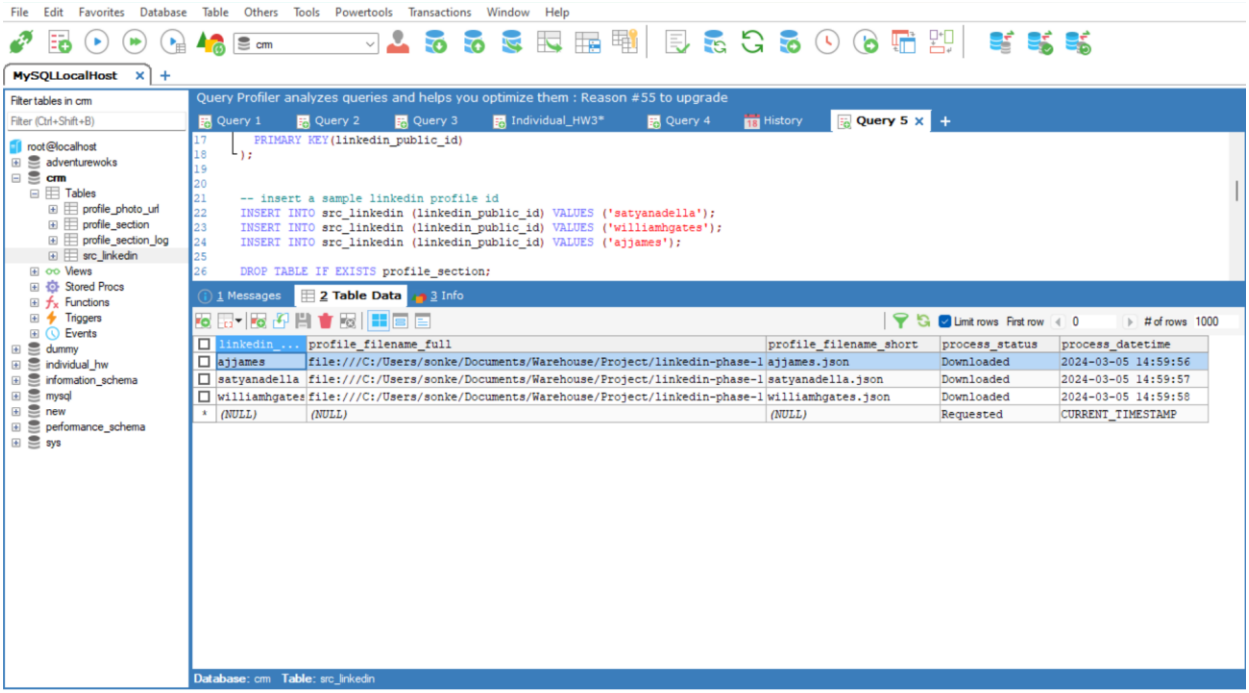


Step 6:

Go back to SQL Server and refresh the “src_linkedin” table to check if the URL is generated and the photos status is “Downloaded”.

Observation 3:

The database table shows that LinkedIn profiles for 'ajjames' and 'satyanadella' and ‘'williamhgates''have been processed and marked as 'Downloaded', with their respective JSON filenames recorded. It also shows that it pulls profiles incrementally as the timestamp column shows that there is minor second changes between them.



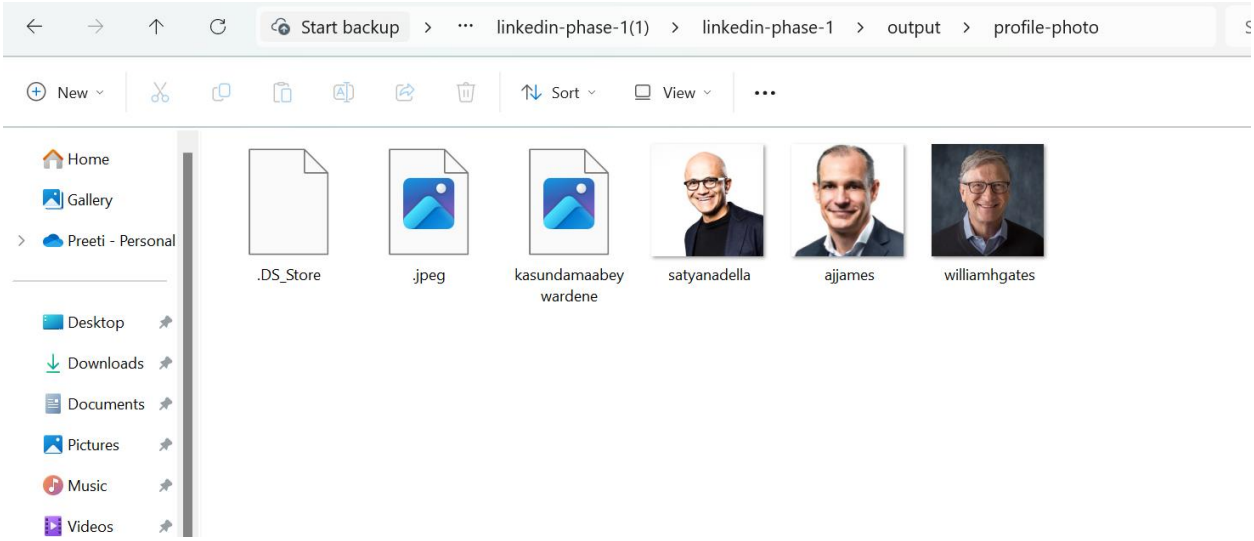
Step 7:

Go to “Output” folder in “linkedin_phase_1” folder to check the images and JSON file is created or not.

In the screenshots below we can see that the images and JSON file for requested person’s LinkedIn profile have been generated successfully.

Output -> profile-photo:

Group Project 1 Group No: 08
Members: Apurva Patil, Preeti Sonke, Durga Sai Sri Bommi



Output -> Profile: