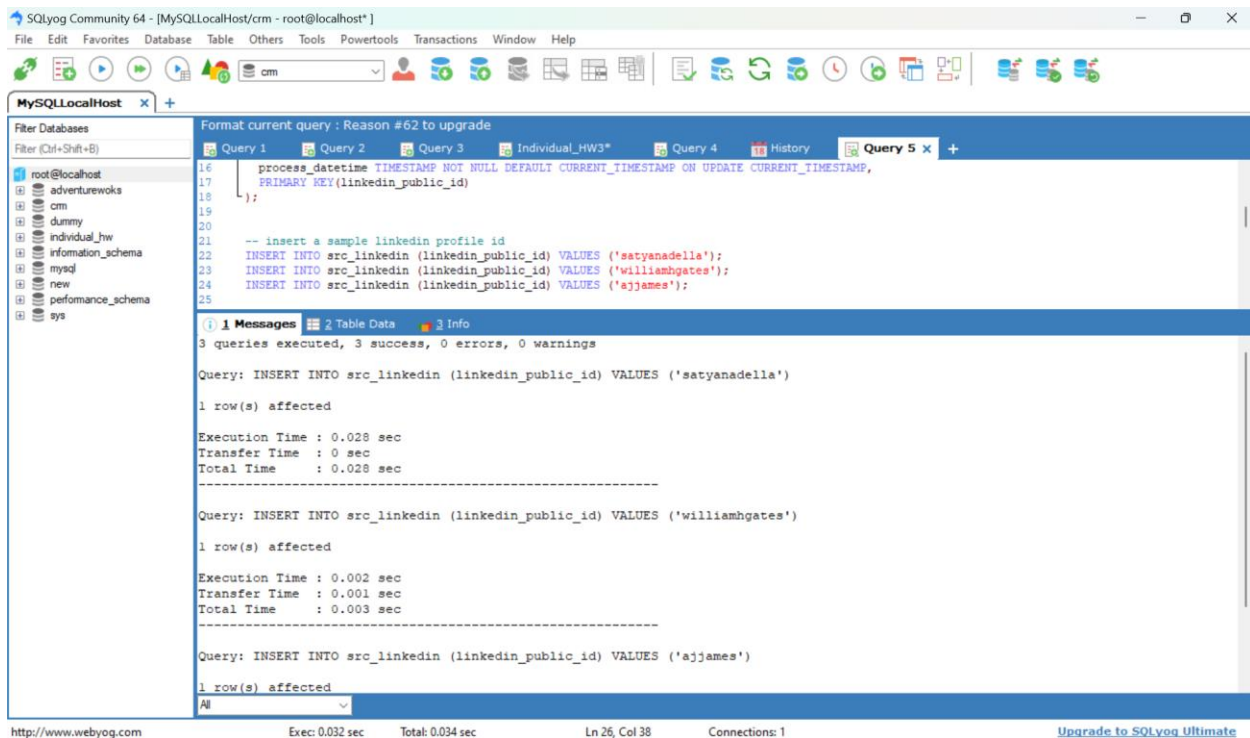


## Ingesting LinkedIn Profile

### 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:



The screenshot shows the SQLyog Community 64 interface. The left sidebar displays the database structure, including the 'crm' database. The main window shows the execution of three SQL queries to insert data into the 'src\_linkedin' table. The queries are as follows:

```
Query 1: process_datetime TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP, PRIMARY KEY(linkedin_public_id);
```

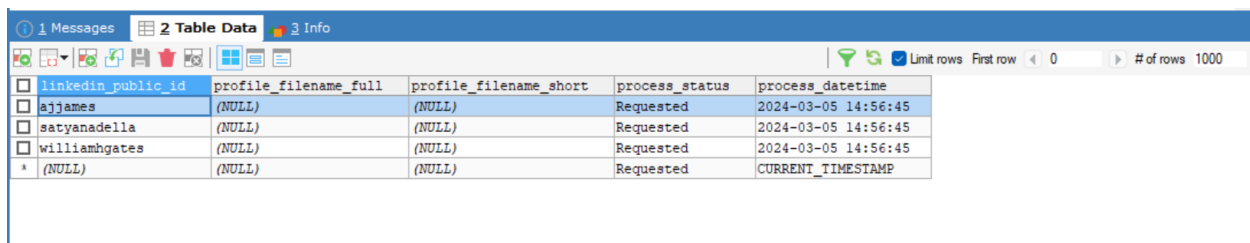
```
Query 2: -- insert a sample linkedin profile id
```

```
Query 3: INSERT INTO src_linkedin (linkedin_public_id) VALUES ('satyanadella');
INSERT INTO src_linkedin (linkedin_public_id) VALUES ('williamhgates');
INSERT INTO src_linkedin (linkedin_public_id) VALUES ('ajjames');
```

The bottom panel shows the execution results for each query, indicating that 1 row(s) were affected for each insert operation. The total execution time for all queries was 0.034 seconds.

### 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.

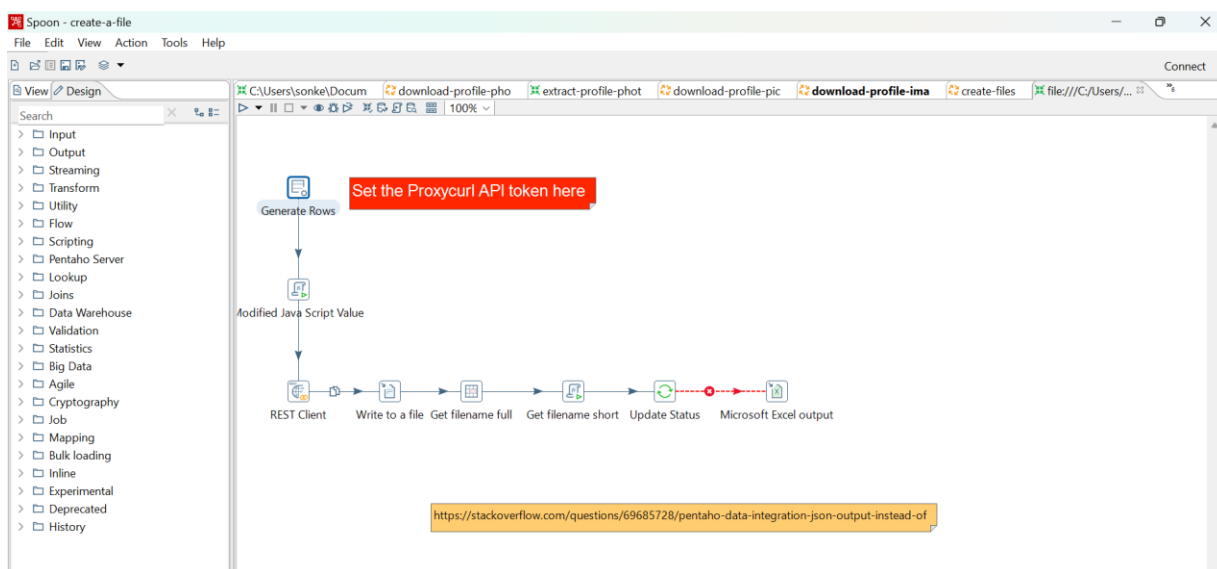
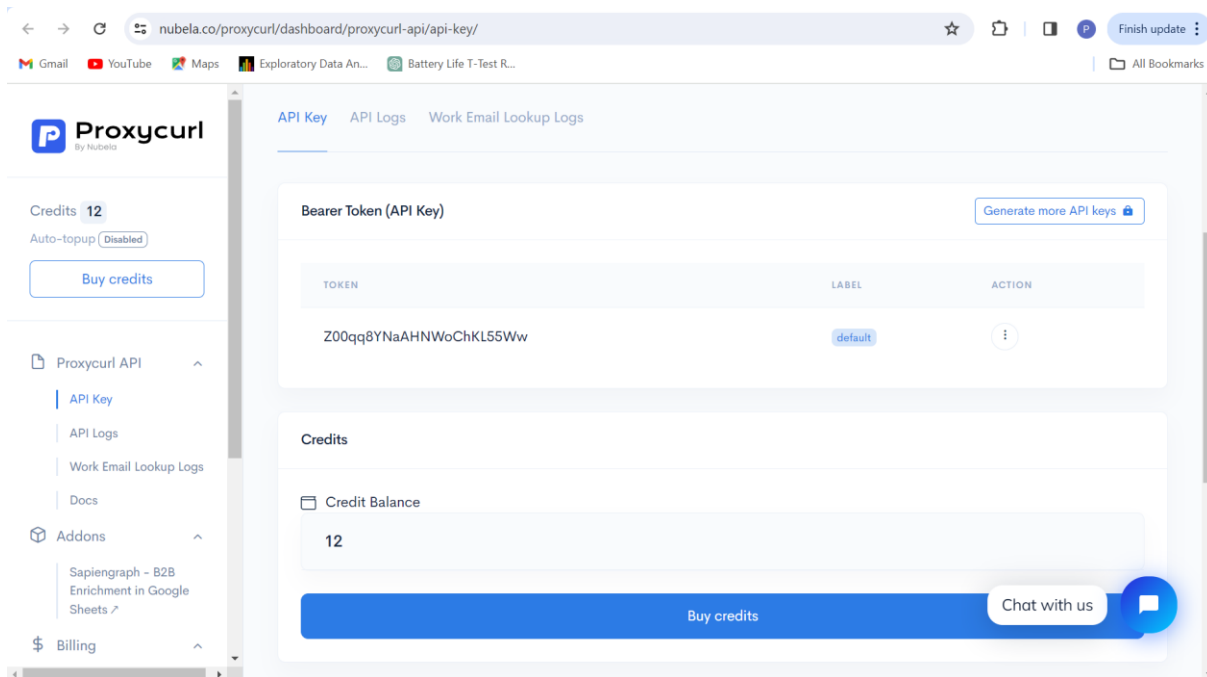


The screenshot shows the 'Table Data' view for the 'src\_linkedin' table. The table contains the following data:

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.

Step name: Microsoft Excel output

Filename: C:\Users\sonke\Documents\Warehous Browse...

Create Parent folder: ☐

Do not create file at start: ☒

Extension: xls

Include stepnr in filename?: ☐

Include date in filename?: ☒

Include time in filename?: ☐

Specify Date time format: ☐

Date time format:

Show filename(s)...

Add filenames to result: ☒

Help OK Cancel

Here we have provided that token.

Step name: Generate Rows

Limit: 1

Never stop generating rows: ☐

Interval in ms (delay): 5000

Current row time field name: now

Previous row time field name: FiveSecondsAgo

Fields:

#	Name	Type	Value
1	dummy	String	dummy
2	headerAuth	String	Bearer Z00qq8YNaAHNWwCHKL55Ww
3	statusDownloaded	String	Downloaded
4	jsonFileExtension	String	json

### 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.

The screenshot displays the Apache NiFi web interface for a workflow named "download-raw-files". The top navigation bar includes "File", "Edit", "View", "Action", "Tools", and "Help". The left sidebar shows a "Search" bar and a tree view of the NiFi catalog, including categories like General, Mail, File management, Conditions, Scripting, Bulk loading, Big Data, Modeling, XML, Utility, Repository, File transfer, File encryption, and Deprecated.

The main workspace shows a workflow design with the following steps:

- main-refresh-linkedin
- download-raw-files
- get-linkedin-urls
- create-a-file
- download-profile-photo
- extract-profile-photo-url
- download-profile-picture

The "get-linkedin-urls" step is highlighted with a green box, and the "create-a-file" step is highlighted with a red box. A text box explains the data flow:

Get a list of LinkedIn URLs to extract the profiles.  
Two following parameters get passed to the next step, "create-files".

- linkedin\_public\_id
- linkedin\_profile\_url

The two parameters above get mapped to the user defined parameters below:

- PAR\_LINKEDIN\_PROFILE\_URL
- PAR\_LINKEDIN\_PUBLIC\_ID

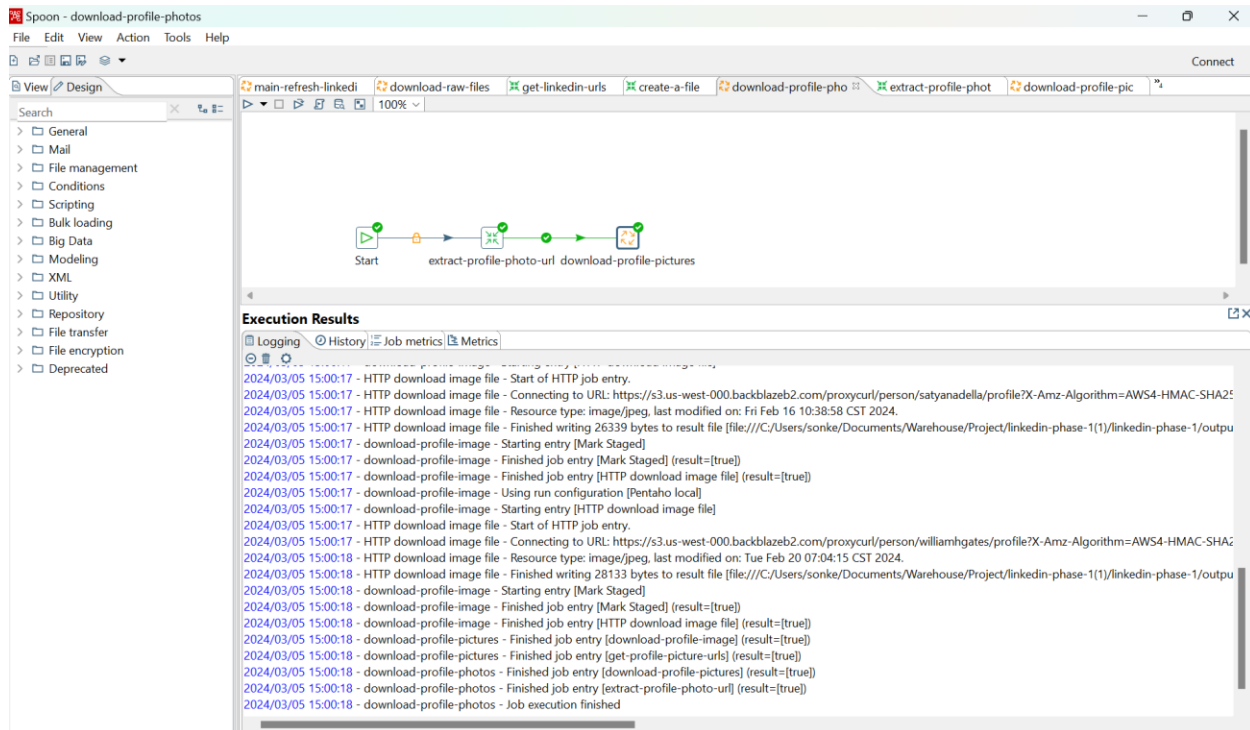
Then, it downloads its LinkedIn profiles in JSON to a local folder /output/profiles

The bottom section shows the "Execution Results" tab, which includes a "Logging" section with the following log entries:

```
2024/03/05 14:59:55 - download-raw-files - Starting entry [create-a-file]
2024/03/05 14:59:55 - create-files - Using run configuration [Pentaho local]
2024/03/05 14:59:55 - create-files - Starting entry [create a file]
2024/03/05 14:59:55 - create a file - Using run configuration [Pentaho local]
2024/03/05 14:59:55 - create a file - Using legacy execution engine
2024/03/05 14:59:55 - create-a-file - Dispatching started for transformation [create-a-file]
2024/03/05 14:59:55 - Generate Rows.0 - Finished processing (I=0, O=0, R=0, W=1, U=0, E=0)
2024/03/05 14:59:55 - Modified Java Script Value.0 - Optimization level set to 9.
2024/03/05 14:59:55 - Modified Java Script Value.0 - Finished processing (I=0, O=0, R=1, W=1, U=0, E=0)
2024/03/05 14:59:56 - REST Client.0 - Finished processing (I=0, O=0, R=1, W=1, U=0, E=0)
2024/03/05 14:59:56 - Get filename short.0 - Optimization level set to 9.
2024/03/05 14:59:56 - Write to a file.0 - Finished processing (I=0, O=1, R=1, W=1, U=0, E=0)
2024/03/05 14:59:56 - Get filename full.0 - Finished processing (I=0, O=0, R=1, W=1, U=0, E=0)
2024/03/05 14:59:56 - Get filename short.0 - Finished processing (I=0, O=0, R=1, W=1, U=0, E=0)
2024/03/05 14:59:56 - Update Status.0 - Finished processing (I=1, O=0, R=1, W=1, U=1, E=0)
2024/03/05 14:59:56 - create-a-file - Finished job entry [create a file] (result=true)
```

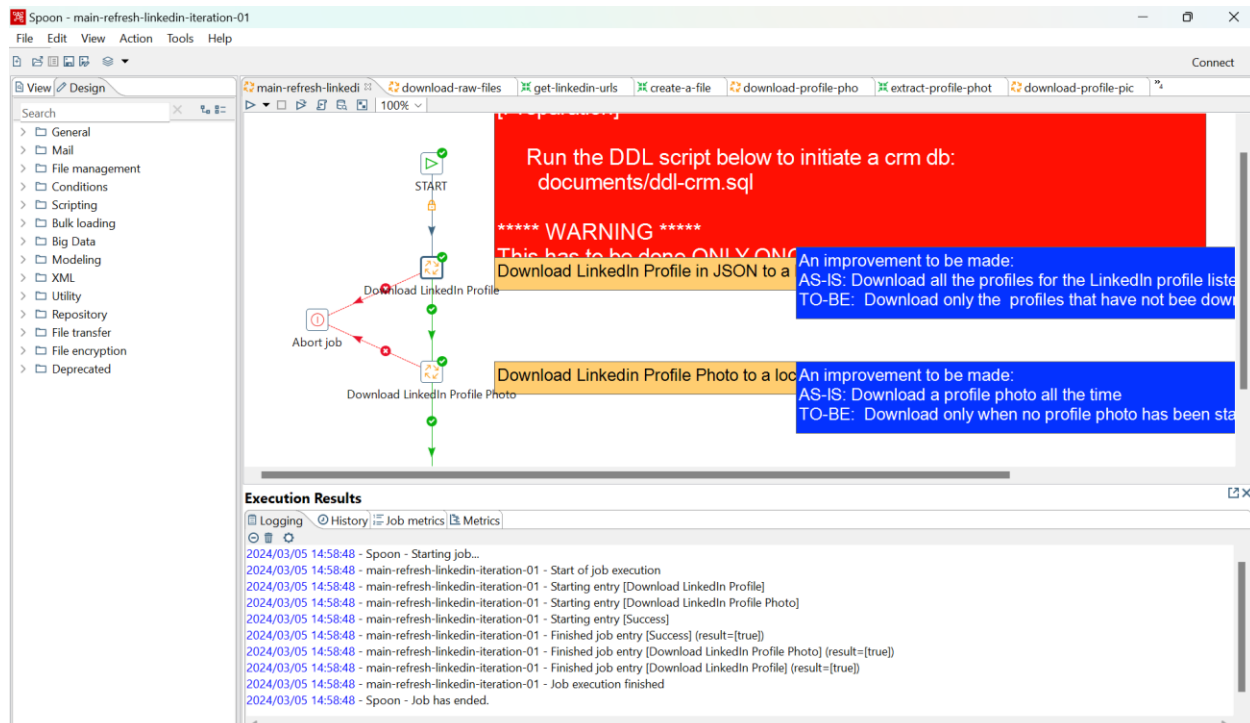
### Step 4:

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.



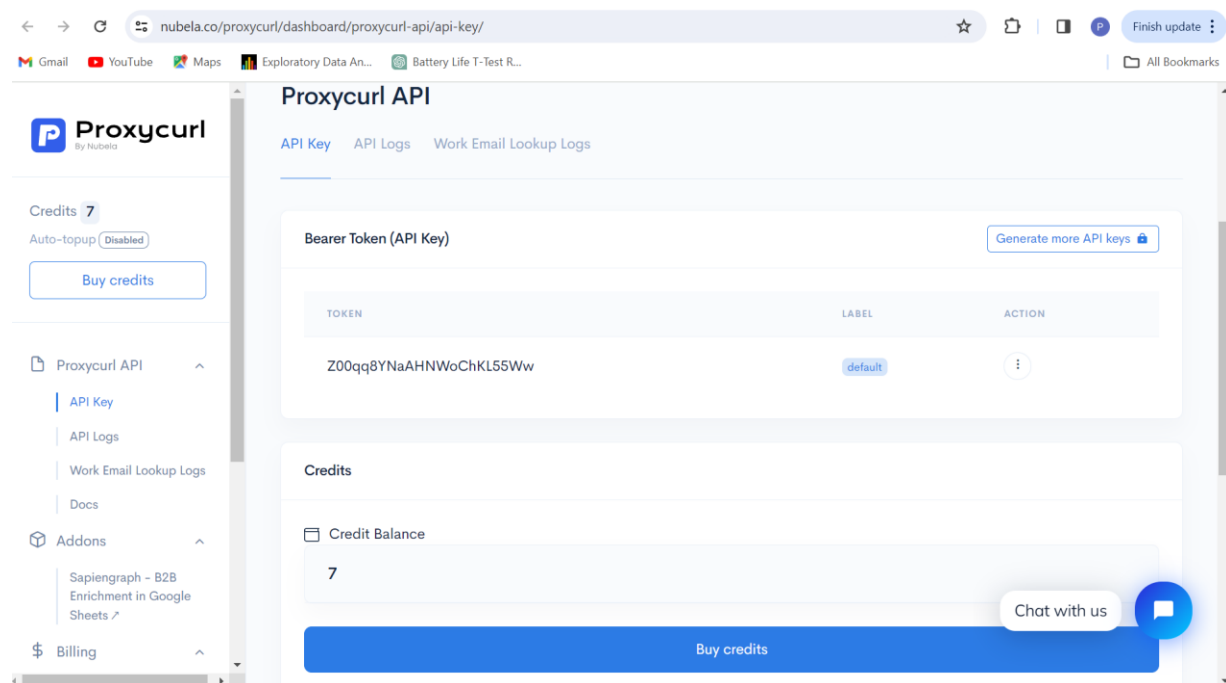
## Step 5:

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.

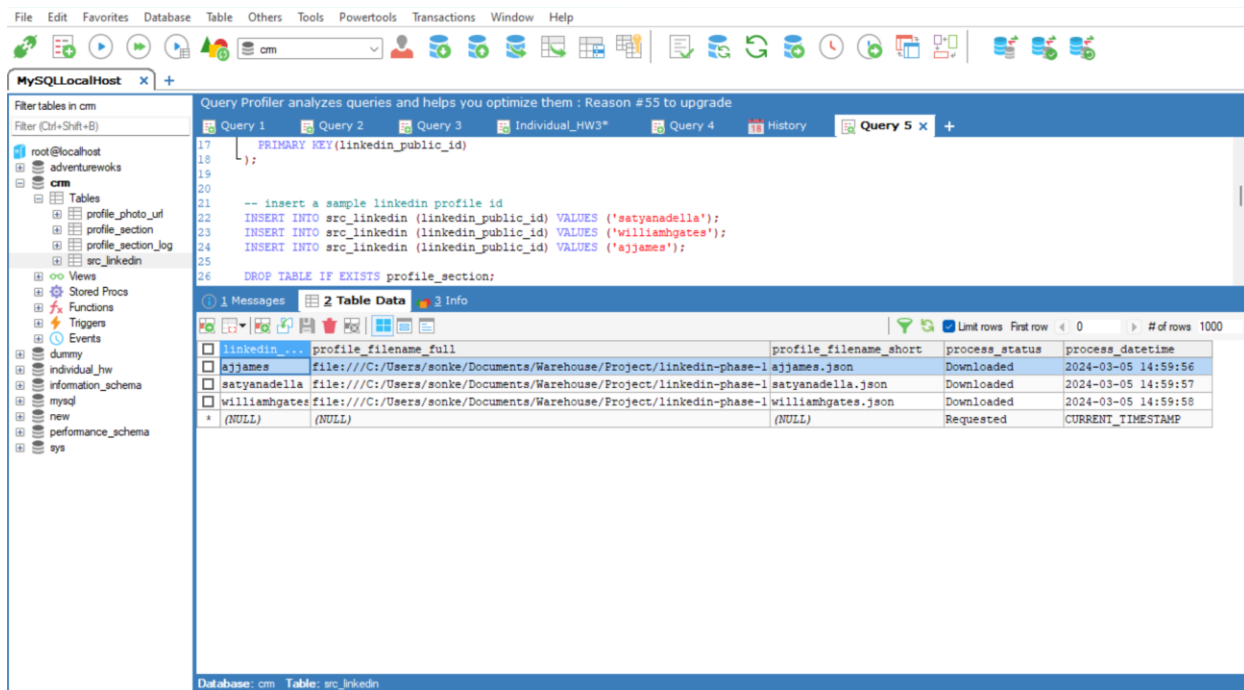


## Step 6:

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 “williamhgate” 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.

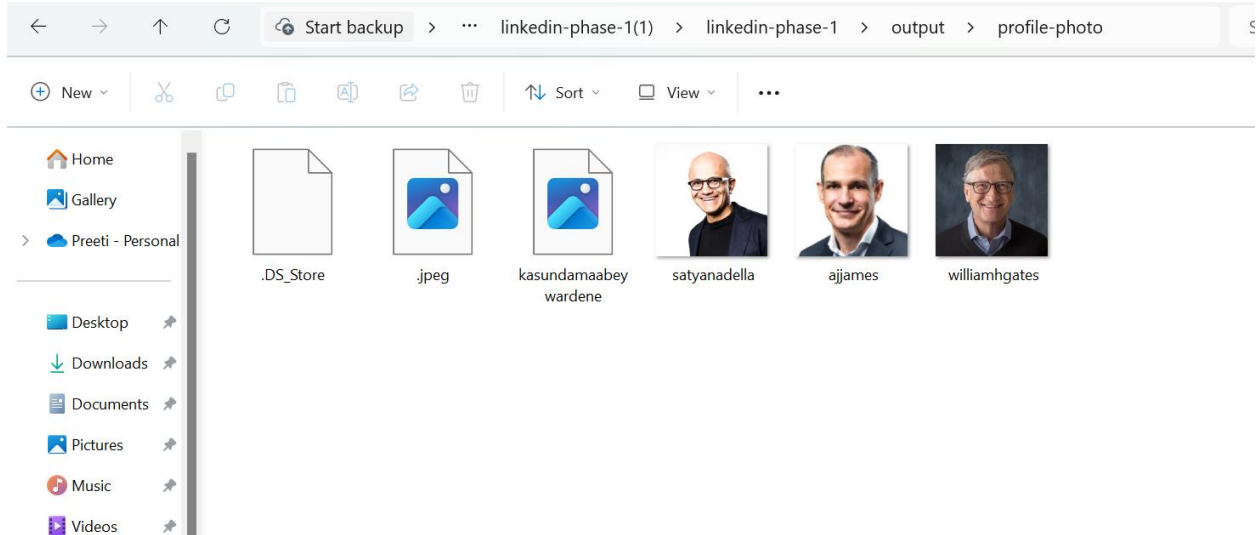


## Step 7:

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:**



## Output -> Profile:

Start backup > ... Project > linkedin-phase-1(1) > linkedin-phase-1 > output > profile

New | Cut | Copy | Paste | Delete | Sort | View | ...

Name	Date modified	Type	Size
.DS_Store	2/27/2024 11:21 PM	DS_STORE File	7 KB
json	3/4/2024 10:13 PM	JSON File	1 KB
ajames	3/5/2024 2:59 PM	JSON File	34 KB
kasundamaabeywardene	3/4/2024 10:20 PM	JSON File	1 KB
satyanadella	3/5/2024 2:59 PM	JSON File	8 KB
williamhgates	3/5/2024 2:59 PM	JSON File	5 KB

Home | Gallery | Preeti - Personal

Desktop | Downloads | Documents | Pictures | Music | Videos | Work