

## DEMO: Flowise with GraphRAG

# DATABASES

Relational Database: PostgreSQL

Vector Database: PGVector (PostgreSQL extension)

Graph Database: neo4j

Administration and UI:

PostgreSQL, PGVector - pgAdmin

neo4j - has own web UI

# COGNIFY TOOL



Hi there! How can I help?



текст.txt

COGNIFY()



Process Flow



cognify

I've initiated the cognify process to transform your text into a structured knowledge graph. The background task has been launched successfully, and I've logged the operation for your reference.

To monitor the progress of this operation, you can use the [cognify\\_status](#) tool. Alternatively, you can check the detailed log file at: [/app/.venv/lib/python3.12/site-packages/logs/2025-09-11\\_00-47-49.log](#).

Once the cognify process completes, we'll be able to search and query the structured knowledge graph for insights about the text content. Would you like me to proceed with a search for specific information from this text?

# COGNIFY TOOL RELATION DB UPDATE

Object Explorer

public.EdgeType\_relationship\_name/cognee\_db/cognee@postgres x

text tree table

Занпо

1

2

3

4

5

6

Total rows: 10 Query complete 00:00:00.206

CRLF Ln 1, Col 1

relationship (edge)

```
{
  id : 014f8baa-f01b-5272-b60e-4e2a9dc4327d
  created_at : 1757552476163
  updated_at : 1757552476163
  ontology_valid : false
  version : 1
  topological_rank : 0
  metadata : {
    index_fields : [ 1 item
      0 : text
    ]
  }
  type : IndexSchema
  belongs_to_set : null
  text : first_appeared_in
}
```

Scratch Pad x

Page No: 1 of 1

ontology\_valid": false, "version": 1, "topological\_rank": 0, "metadata": {"index

ontology\_valid": false, "version": 1, "topological\_rank": 0, "metadata": {"index

ontology\_valid": false, "version": 1, "topological\_rank": 0, "metadata": {"index

id": "4ff5b30c-5d1c-5cd0-a87a-7505cdcc7b68", "created\_at": 1757552446650, "updated\_at": 1757552446650, "ontology\_valid": false, "version": 1, "topological\_rank": 0, "metadata": {"index

id": "6154e9f7-8f39-5962-9549-fc6361a4fa01", "created\_at": 1757552476163, "updated\_at": 1757552476163, "ontology\_valid": false, "version": 1, "topological\_rank": 0, "metadata": {"index

Отмена OK

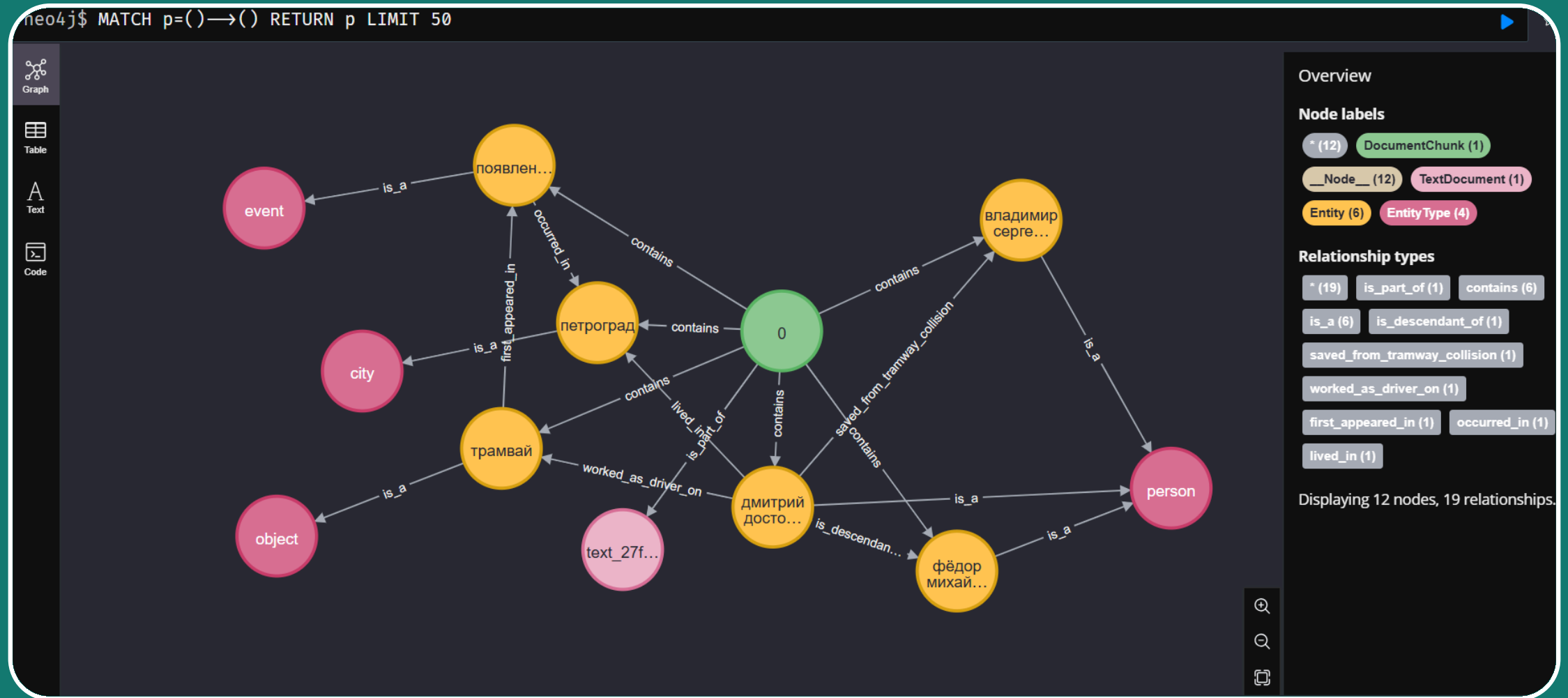
# COGNIFY TOOL VECTOR DB UPDATE

The screenshot displays the Cognify tool interface. On the left is the 'Object Explorer' showing a database schema with tables like 'EdgeType\_relationship\_name'. The main area shows a SQL query: `SELECT * FROM public."EdgeType_relationship_name" ORDER BY id ASC`. Below the query, the 'Data Output' tab shows a table of results. A green arrow points to the 'vector' column in the results table, with the text 'embeddings for relationships (edges)' next to it.

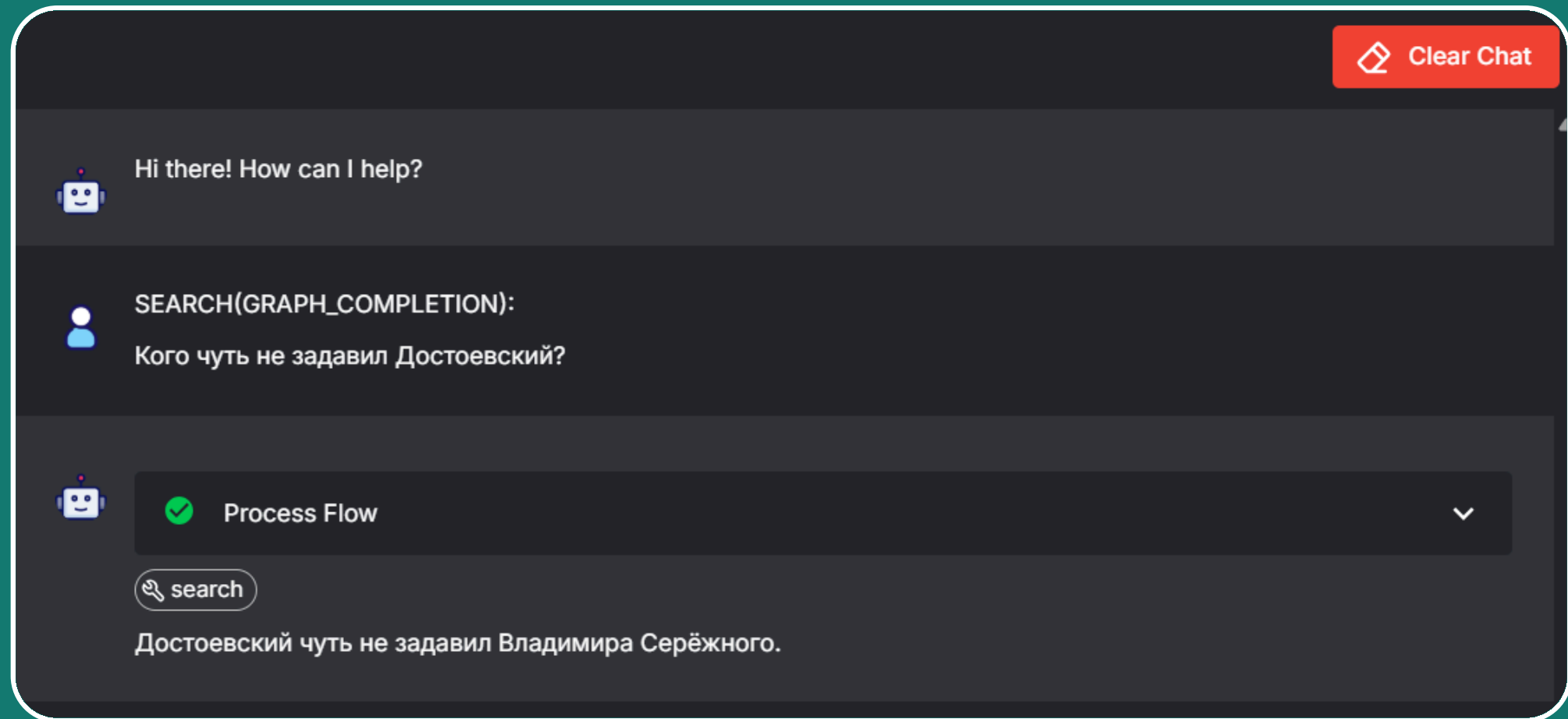
embeddings for relationships (edges)

	vector
1	["index_fiel...", [-0.083805345,-0.029250324,-7.553161,1.907051,-0.43000162,3.2445948,5.396335,-4.2860813,3.6174388,-5.006088,0.94685453,0.13209248,-0.0027009852,1.2837968,
2	["index_fiel...", [-0.09626067,-2.1899843,3.015535,-3.1507804,-0.52285457,5.2188425,7.6990557,-3.012274,2.9376793,-1.8482565,2.8399625,-2.6217759,0.08594765,-5.5840063,13.92
3	["index_fiel...", [-0.022312393,-2.929693,-3.8393953,-5.2029953,-0.19519696,5.965074,0.9933882,0.39846998,4.8819675,-0.17068268,-6.854203,3.4810648,0.3021149,-1.8696035,5.42
4	["index_fiel...", [-0.094649255,-0.05901919,-5.1721506,-1.3326005,-0.5033157,5.655086,3.438326,1.187798,4.45696,-5.8160033,-0.73448884,3.1503563,0.18799098,2.6304805,11.0696
5	["index_fiel...", [0.14098944,4.0320354,-2.5898368,-1.0656737,0.22838041,0.8435495,3.8470442,-20.48795,10.540073,-0.710742,-1.9729574,0.6199873,3.067029,-1.7971572,6.498746
6	index_field...", [-0.08686823,-0.2403075,-4.4718513,-3.6269088,-0.5096364,5.319092,4.43235,2.089379,4.1586094,-2.513923,-3.0625486,0.39058486,0.05895237,-4.27275,10.45061,1.

# COGNIFY TOOL GRAPH DB UPDATE



# SEARCH(GRAPH\_COMPLETION) TOOL



# SEARCH TYPES

`SearchType.SUMMARIES` - Getting concise overviews of topics. Summarizing large amounts of information. Quick understanding of complex subjects.

`SearchType.INSIGHTS` - Discovering how entities are connected. Understanding relationships between concepts. Exploring the structure of your knowledge graph.

`SearchType.CHUNKS` - Finding specific facts. Getting direct answers to questions. Retrieving precise information.

`SearchType.COMPLETION` - Getting detailed explanations. Combining multiple pieces of information. Generating comprehensive answers

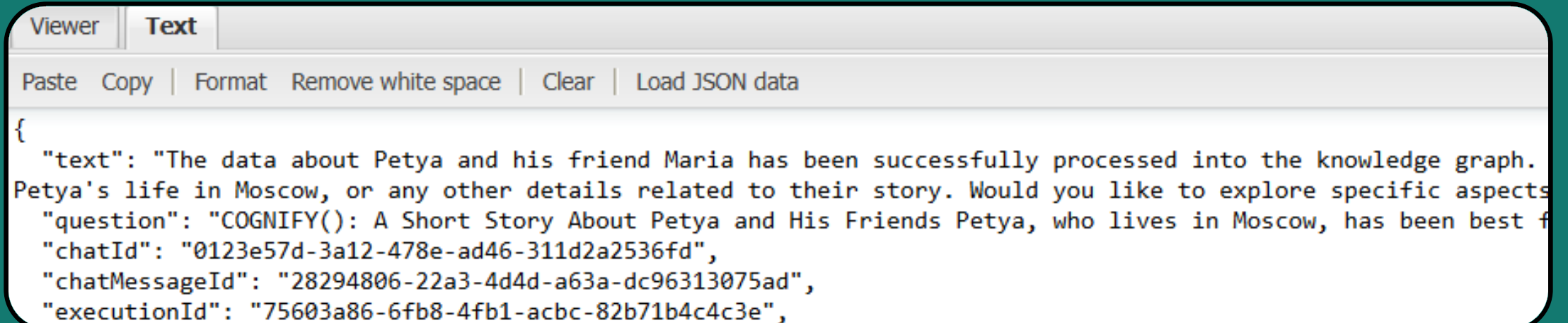
`SearchType.GRAPH_COMPLETION` - Complex queries requiring graph traversal. Questions that benefit from understanding relationships. Queries where context from connected entities matters

`SearchType.CYPHER` - Executing precise graph queries with full control. Leveraging Cypher features and functions. Getting raw data directly from the graph database



# CURL QUERY

```
curl http://localhost:3000/api/v1/prediction/b2ea9565-02ef-45d7-ae9e-32d92b5b3140 \
-X POST \
-d '{"question": "COGNIFY(): A Short Story About Petya and His Friends Petya, who lives in Moscow, has been best friends with Maria since childhood."}' \
-H "Content-Type: application/json" \
-H "Authorization: Bearer RO6maTmLiESQonJOU-BNmPa13YzlrQS-aK37pB2_nyU"
```

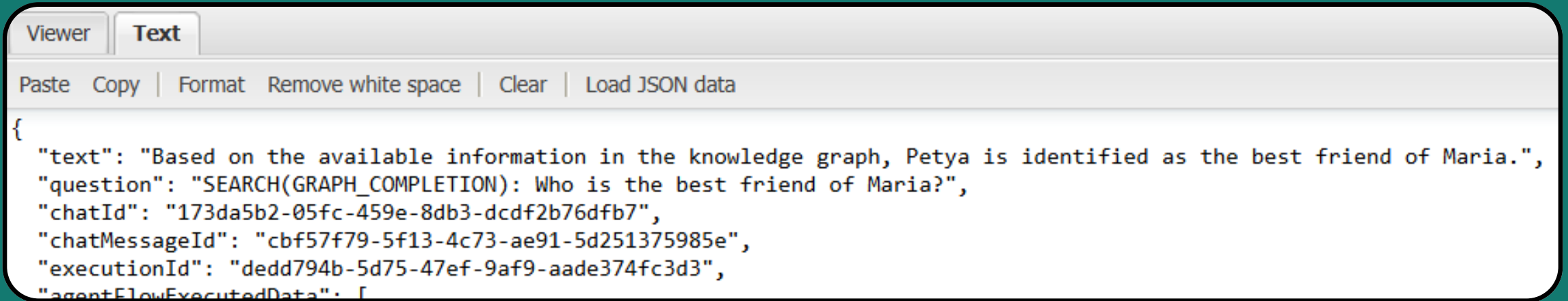


The screenshot shows a web-based JSON viewer. At the top, there are two tabs: 'Viewer' and 'Text', with 'Text' being the active tab. Below the tabs is a toolbar with buttons for 'Paste', 'Copy', 'Format', 'Remove white space', 'Clear', and 'Load JSON data'. The main area displays a JSON object with the following structure:

```
{
  "text": "The data about Petya and his friend Maria has been successfully processed into the knowledge graph. Petya's life in Moscow, or any other details related to their story. Would you like to explore specific aspects",
  "question": "COGNIFY(): A Short Story About Petya and His Friends Petya, who lives in Moscow, has been best f",
  "chatId": "0123e57d-3a12-478e-ad46-311d2a2536fd",
  "chatMessageId": "28294806-22a3-4d4d-a63a-dc96313075ad",
  "executionId": "75603a86-6fb8-4fb1-acbc-82b71b4c4c3e",
```

# CURL QUERY

```
curl http://localhost:3000/api/v1/prediction/b2ea9565-02ef-45d7-ae9e-32d92b5b3140 \
-X POST \
-d '{"question": "SEARCH(GRAPH_COMPLETION): Who is the best friend of Maria?"}' \
-H "Content-Type: application/json" \
-H "Authorization: Bearer RO6maTmLiESQonJOU-BNmPa13YzlrQS-aK37pB2_nyU"
```



The screenshot shows a web-based JSON viewer interface. At the top, there are two tabs: 'Viewer' and 'Text', with 'Text' being the active tab. Below the tabs is a toolbar with buttons for 'Paste', 'Copy', 'Format', 'Remove white space', 'Clear', and 'Load JSON data'. The main area displays a JSON object with the following fields:

```
{
  "text": "Based on the available information in the knowledge graph, Petya is identified as the best friend of Maria.",
  "question": "SEARCH(GRAPH_COMPLETION): Who is the best friend of Maria?",
  "chatId": "173da5b2-05fc-459e-8db3-dcdf2b76dfb7",
  "chatMessageId": "cbf57f79-5f13-4c73-ae91-5d251375985e",
  "executionId": "dedd794b-5d75-47ef-9af9-aade374fc3d3",
  "agentFlowExecutedData": [
```