

V O L V O

# Jesienna konferencja COMMON POLSKA

20 Paźdernika 2025, Zakopane

Sztuka dzielenia się

## Tworzenie hybrydowych przepływów pracy AI + n8n + IBM i

MARCIN OGOŃ





# n8n (n-eight-n)

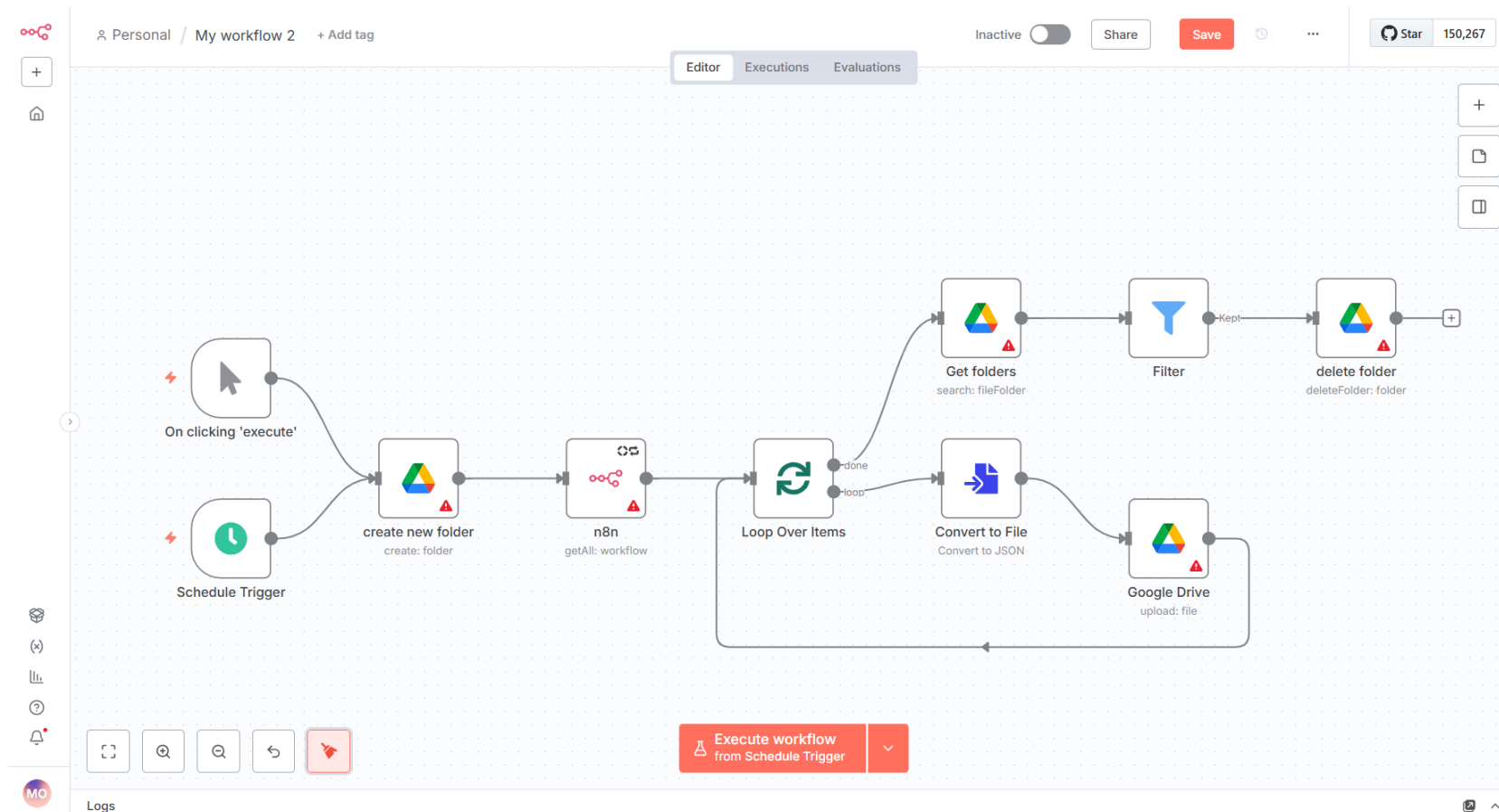
## Automatyzacja procesów bez kodowania i ograniczeń

- Wizualne narzędzie do automatyzacji zadań i procesów („low-code”).
- Pozwala łączyć ze sobą setki różnych aplikacji i usług.
- Działa na zasadzie „węzłów” (nodes) – każdy węzeł to jeden krok w Twoim procesie.
- Jest to rozwiązanie open-source, co oznacza, że możesz je hostować na własnym serwerze.
- Nadanie AI „rąk”: Sam model AI potrafi tylko generować tekst. n8n pozwala mu działać.
- Szybkie prototypowanie: Możesz błyskawicznie budować i testować pomysły na automatyzacje z AI.



# n8n

## Automatyzacja procesów bez kodowania i ograniczeń





# Mapepire

Nowoczesna warstwa dostępu do bazy danych

- Łatwa instalacja i zarządzanie (uruchamiany jako prosty Daemon).
- Optymalizacja pod kątem szybkości dostępu.
- Pozwala aplikacjom napisanym w .NET Core, Node.js czy Python na bezpośredni dostęp do danych na IBM i.
- Ułatwia tworzenie nowych aplikacji (np. Webowych lub mikroservisów) korzystających z istniejącej bazy danych „legacy” bez skomplikowanej konfiguracji.



# Mapepire

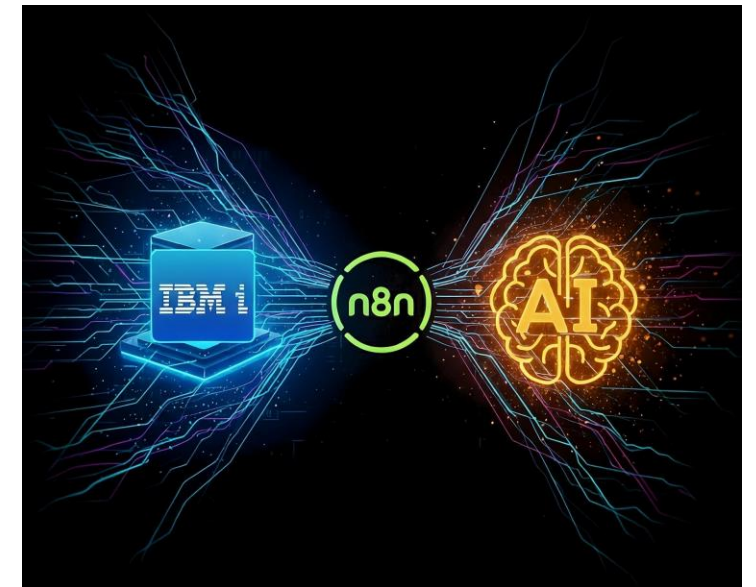
Nowoczesna warstwa dostępu do bazy danych

	JDBC	ODBC	Mapepire
Wymaga tylko jednego portu			✓
Dane są zawsze szyfrowane			✓
Zarządzalne poprzez punkty wyjścia systemu	✓	✓	✓
Rozszerzone wsparcie dla CCSID	✓		✓
Działa w notebookach Jupyter WatsonX.ai			✓
Działa w lekkich kontenerach (np. Alpine Linux)	✓		✓
Bezpośrednio obsługuje wiele języków klienckich			✓

# Integracja z IBM i: n8n-nodes-ibmi-mapepire

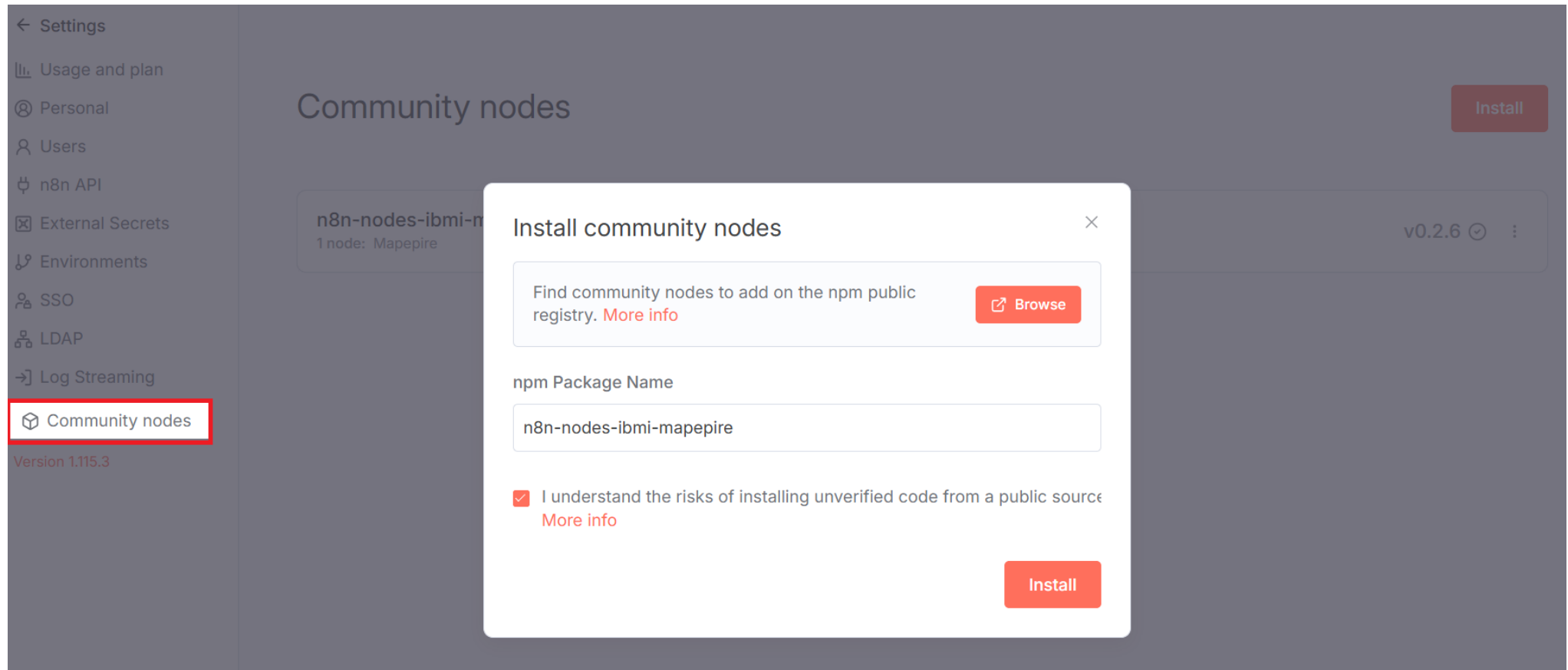
Łatwa integracja i automatyzacja zadań na systemie IBM i.

- Jest to dodatkowy, społecznościowy węzeł (community node) do n8n.
- Został stworzony specjalnie do komunikacji z systemami IBM i.
  - Uruchamianie poleceń SQL
  - Wykonywanie poleceń CL
- Działa jako „most” (klient) łączący n8n z serwerem Mapepire (@ibm/mapepire-js), który ma bezpośredni dostęp do systemu IBM i.
- Potrafi automatycznie stronicować wyniki.
- Proste narzędzie do „wystawienia” logiki i danych z systemu IBM i do nowoczesnych aplikacji i procesów



# Instalacja


AI + n8n + IBM i






# Poświadczenia

AI + n8n + IBM i

 Mapepire Server account 1  
Mapepire Server

 ×

---

Connection

Sharing

Details

Host \*

pub400.com

Port \*

8076

User \*

MOGON

Password \*

.....

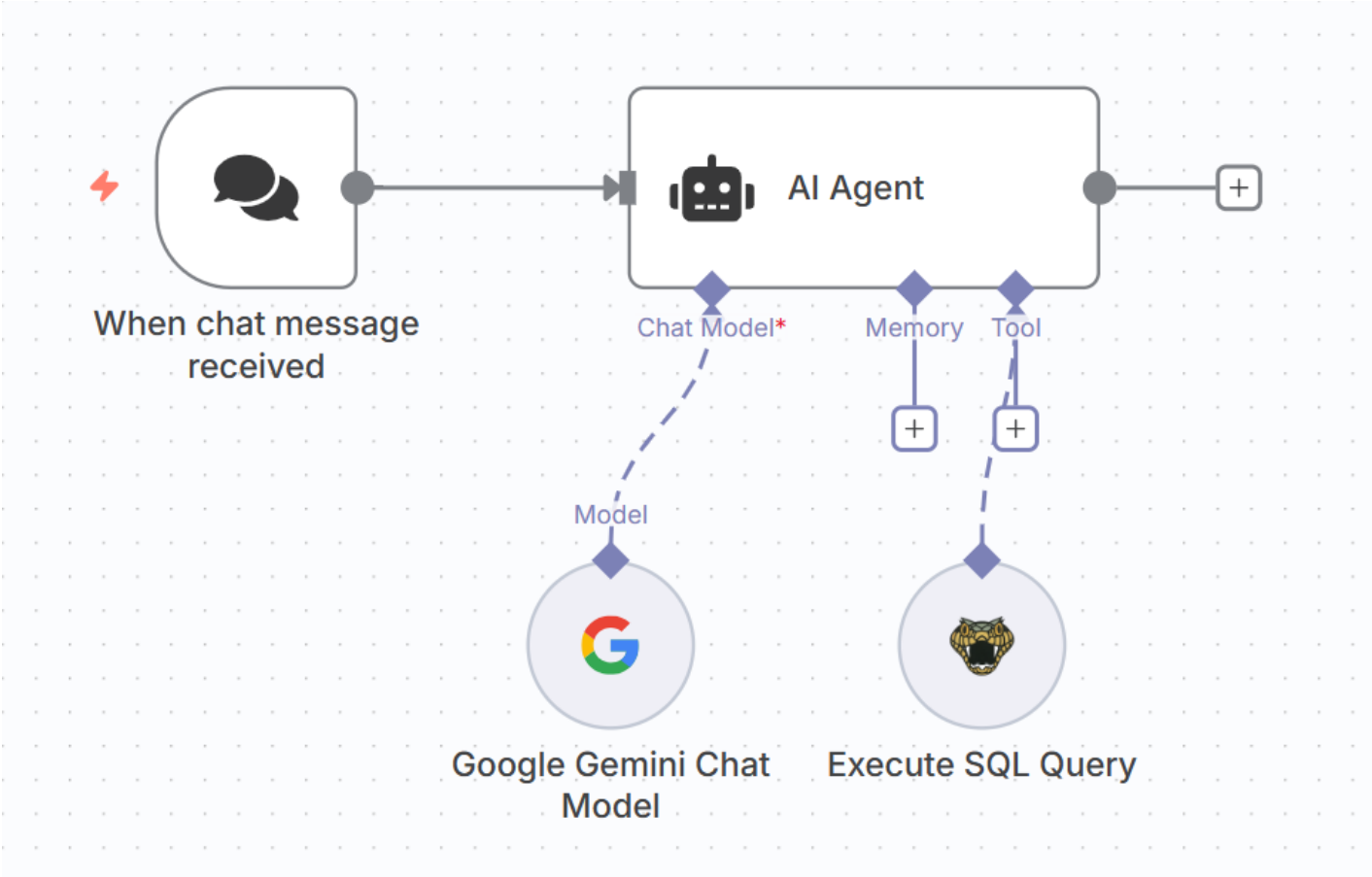
Ignore Unauthorized TLS

☒

CA Certificate (PEM)

# Przepływ pracy – odpytywanie bazy danych

# AI + n8n + IBM i



# Prompt

AI + n8n + IBM i

 AI Agent

[Execute step](#)

Parameters

Settings

[Docs](#)

Source for Prompt (User Message)

Define below

Prompt (User Message)


*fx*

You are end user helper. You need to write and run complex queries in DB2 on IBMi aligned with user requests to reduce query number run. Basing on available tools aggregate data and response to user in easy to read manner. Make sure every table uses MOGON1 schema prefix. Don't share schema name with user.

User request:{{ \$json.chatInput }}

# Model i narzędzia

AI + n8n + IBM i

 Google Gemini Chat Model

Parameters Settings

Credential to connect with

Google Gemini(PaLM) Api account

Model

models/gemini-2.5-flash

Options

No properties

Add Option

 Execute SQL Query

Parameters Settings

Execute step

Docs

Credential to connect with

Mapepire Server account 1

Description

Get all the data from DB2 on IBM i, make sure you append the tables in MOGON1 schema.  
Don't use ';' at the end.

Mode

SQL

SQL

`fx {{ $fromAI("sql_query", "SQL Query") }}`

undefined

Fetch Size

100

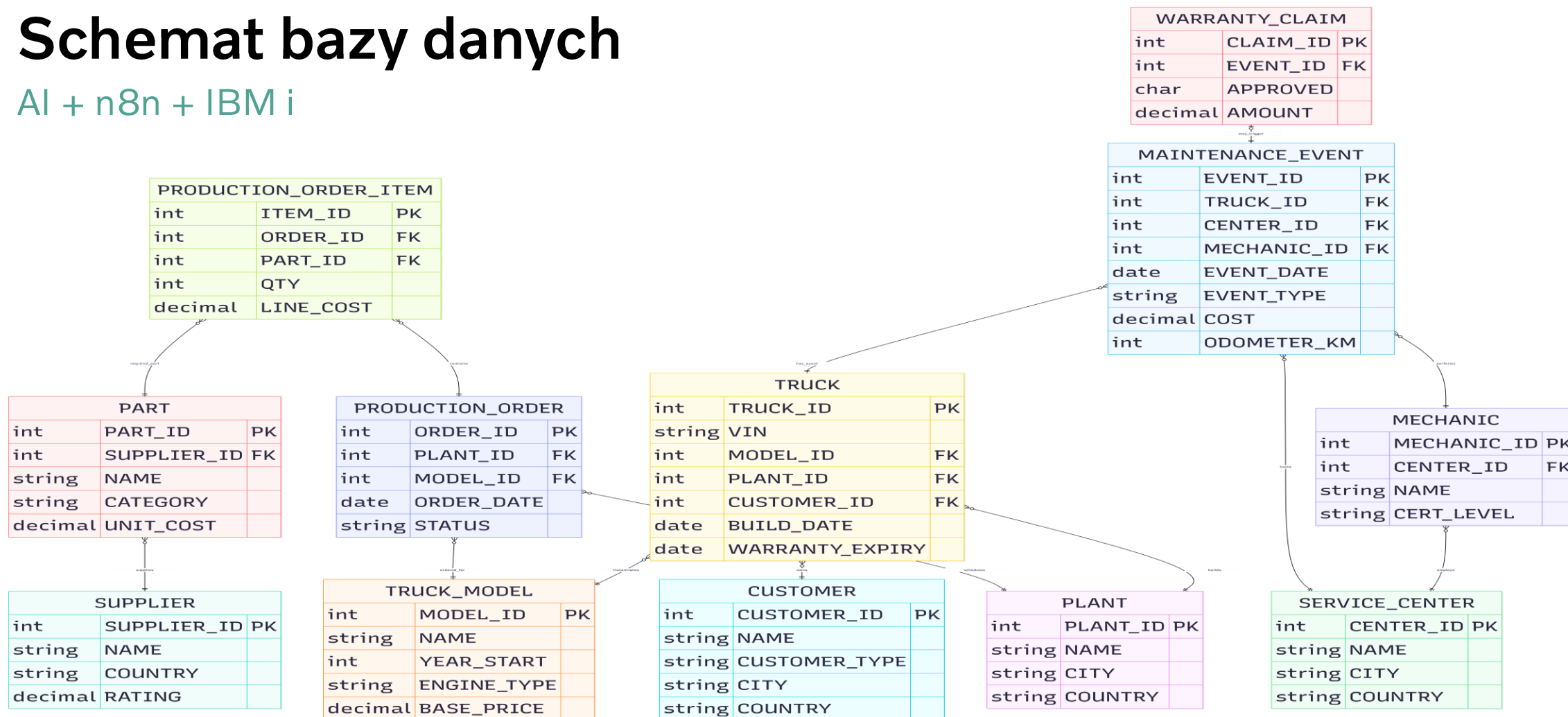
Additional Fields

No properties

Add Field

# Schemat bazy danych

AI + n8n + IBM i



# Realizacja

AI + n8n + IBM i

ChatConnect [Truck Database](#) Admin Assistant

### Truck Database Chat

Ask questions and get information from the truck database.

Hi there! 🙋

Start a chat. We're here to help you 24/7.

Hi there! 🙋

My name is Nathan. How can I assist you today?

For the last 3 years, show the top 10 truck models by number of maintenance events — return model name and event count.

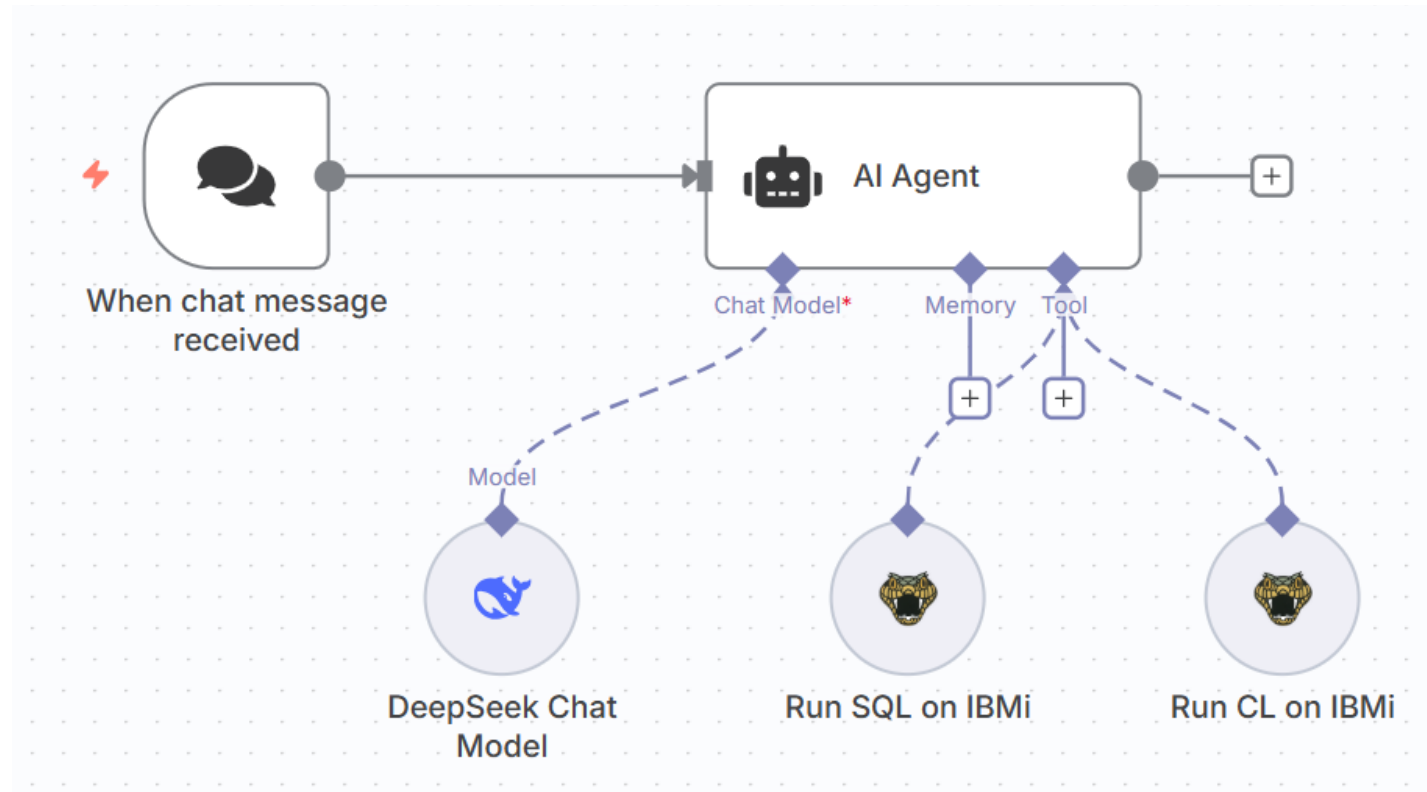
For the last 3 years, the top 10 truck models by number of maintenance events are:

- Model 221: 8 events
- Model 238: 8 events
- Model 214: 8 events
- Model 783: 8 events
- Model 989: 8 events
- Model 463: 8 events




# Przeływ pracy – uruchamianie komend

AI + n8n + IBM i



# Konfiguracja narzędzia

AI + n8n + IBM i

 Run CL on IBMi

Execute step

Parameters

Settings

Docs

Credential to connect with

Mapepire Server account 2

Description

Run CL command on IBM i

Mode

CL Command

CL Command

fx

{{ \$fromAI("cl\_command", "CL Command") }}

undefined

Additional Fields

Terse Results

Include Metadata

Add Field

# Analiza zapytania

AI + n8n + IBM i

Logs

Clear execution

Success in 36.673s | 33,556 Tokens

When chat message received

AI Agent

- DeepSeek Chat Model
- Run SQL on IBMi
- DeepSeek Chat Model
- Run SQL on IBMi
- DeepSeek Chat Model

When chat message received Success in 0s

OUTPUT

1 item

sessionId	action	chatInput
a4b08483b47e4aa882d721b5c8487e3b	sendMessage	List all active job

# Analiza zapytania

AI + n8n + IBM i

Logs

Clear execution

Success in 36.673s | 33,556 Tokens

When chat message received

AI Agent

DeepSeek Chat Model

Run SQL on IBMi

DeepSeek Chat Model

Run SQL on IBMi

DeepSeek Chat Model

DeepSeek Chat Model

Success in 4.636s | 410 Tokens

Input

Output

...

▼

INPUT

Q

≡

≡

≡

{}

1 item

Human: You are supporting tool. Choose best CL commands and SQL statements for IBM i release: 7.5 to finalize task and execute like on ACS Run SQL (or STRSQL / ODBC) on user with no \*ALLOBJ. Optimize and minimize SQL and command output if not needed. For every table function use TABLE(table function) to get data like SELECT \* FROM TABLE(QSYS2.OBJECT\_STATISTICS('MJATST','JRN')) AS X.\n\nQuestion: List all active job

OUTPUT

Q

≡

≡

≡

{}

1 item

```
[
  {
    "response": {
      "generations": [
        [
          {
            "text": "",
            "generationInfo": {
              "finish_reason": "tool_calls"
            }
          }
        ]
      ],
      "tokenUsage": {
        "completionTokens": 41,
        "promptTokens": 369,
        "totalTokens": 410
      }
    }
  }
]
```

# Analiza zapytania

AI + n8n + IBM i

The screenshot displays the n8n workflow editor interface during the execution of a workflow. On the left, the 'Logs' panel shows a sequence of steps: 'When chat message received', 'AI Agent', and a loop of 'DeepSeek Chat Model' and 'Run SQL on IBMi'. The 'Run SQL on IBMi' step is currently selected and highlighted. The main workspace is divided into 'INPUT' and 'OUTPUT' sections. The 'INPUT' section shows the SQL query: `SELECT * FROM TABLE(QSYS2.ACTIVE_JOB_INFO(JOB_NAME _FILTER => '%')) AS X`. The 'OUTPUT' section displays the JSON response from the IBM i database, which includes a 'response' object with 'rows' (an empty array), 'metadata' (containing 'column\_count': 125, 'job': '431323/QUSER/QZDASOINIT', and 'columns' array), and a 'name' field 'JOB\_NAME'.

Logs Clear execution Run SQL on IBMi Success in 6.593s Input Output ... ▼

Success in 36.673s | 33,556 Tokens

When chat message received

AI Agent

- DeepSeek Chat Model
- Run SQL on IBMi**
- DeepSeek Chat Model
- Run SQL on IBMi
- DeepSeek Chat Model

**INPUT** 1 item

**SQL**

```
SELECT * FROM
TABLE(QSYS2.ACTIVE_JOB_INFO(JOB_NAME
_FILTER => '%')) AS X
```

**OUTPUT** 1 item

```
[
{
  "response": [
    {
      "rows": [
      ],
      "metadata": {
        "column_count": 125,
        "job": "431323/QUSER/QZDASOINIT",
        "columns": [
          {
            "name": "ORDINAL_POSITION",
            "type": "INTEGER",
            "display_size": 11,
            "label": "ORDINAL_POSITION",
            "precision": 10,
            "scale": 0,
            "autoIncrement": false,
            "nullable": 1,
            "readOnly": true,
            "writeable": false
          },
          {
            "name": "JOB_NAME"
          }
        ]
      }
    }
  ]
}
```

# Analiza zapytania

AI + n8n + IBM i

Logs

Clear execution

Success in 36.673s | 33,5

Overview

Details

When chat message received

AI Agent

DeepSeek Chat Model

Run SQL on IBMi

DeepSeek Chat Model

Run SQL on IBMi

DeepSeek Chat Model

DeepSeek Chat Model

Success in 5.365s | 7,622 Tokens

Input

Output

...

▼

INPUT

1 item

Human: You are supporting tool. Choose best CL commands and SQL statements for IBM i release: 7.5 to finalize task and execute like on ACS Run SQL (or STRSQL / ODBC) on user with no \*ALLOBJ. Optimize and minimize SQL and command output if not needed. For every table function use TABLE(table function) to get data like SELECT \* FROM TABLE (QSYS2.OBJECT\_STATISTICS('MJATST','JRN')) AS X.\n\nQuestion: List all active job\n\nAI: \n\nTool: [{"rows": [], "metadata": {"column\_count": 125, "job": "431323/QUSER/QZDASOINIT", "columns": [{"name": "ORDINAL\_POSITION", "type": "INTEGER", "display\_size": 10, "label": "Ordinal position"}, {"name": "JOB\_NAME", "type": "VARCHAR", "display\_size": 28, "label": "Job name"}, {"name": "JOB\_NAME\_SHORT", "type": "VARCHAR", "display\_size": 10, "label": "Job name short"}, {"name": "JOB\_USER", "type": "VARCHAR", "display\_size": 10, "label": "Job user"}, {"name": "JOB\_NUMBER", "type": "VARCHAR", "display\_size": 6, "label": "Job number"}, {"name": "INTERNAL\_JOB\_ID", "type": "BINARY", "display\_size": 10, "label": "Internal job ID"}, {"name": "SUBSYSTEM", "type": "VARCHAR", "display\_size": 10, "label": "Subsystem"}, {"name": "SUBSYSTEM\_LIBRARY\_NAME", "type": "VARCHAR", "display\_size": 20, "label": "Subsystem library name"}, {"name": "AUTHORIZATION\_NAME", "type": "VARCHAR", "display\_size": 10, "label": "Authorization name"}, {"name": "JOB\_TYPE", "type": "VARCHAR", "display\_size": 3, "label": "Job type"}, {"name": "FUNCTION\_TYPE", "type": "VARCHAR", "display\_size": 10, "label": "Function type"}, {"name": "FUNCTION", "type": "VARCHAR", "display\_size": 10, "label": "Function"}, {"name": "JOB\_STATUS", "type": "VARCHAR", "display\_size": 4, "label": "Job status"}, {"name": "MEMORY\_POOL", "type": "VARCHAR", "display\_size": 9, "label": "Memory pool"}, {"name": "PRIORITY", "type": "INTEGER", "display\_size": 11, "label": "Priority"}]}

OUTPUT

1 item

The query returned an empty result set, which suggests there might be an issue with the filter or the table function. Let me try a more direct approach using a simpler SQL query:



# Analiza zapytania

AI + n8n + IBM i

Logs

Clear execution

Success in 36.673s | 33,556 Tokens

When chat message received

AI Agent

DeepSeek Chat Model

Run SQL on IBMi

DeepSeek Chat Model

Run SQL on IBMi

DeepSeek Chat Model

DeepSeek Chat Model Success in 5.365s | 7,622 Tokens

INPUT

1 item

Human: You are supporting tool. Choose best CL commands and SQL statements for IBM i release: 7.5 to finalize task and execute like on ACS Run SQL (or STRSQL / ODBC) on user with no \*ALLOBJ. Optimize and minimize SQL and command output if not needed. For every table function use TABLE(table function) to get data like SELECT \* FROM TABLE (QSYS2.OBJECT\_STATISTICS('MJATST','JRN')) AS X.\n\nQuestion: List all active job\n\nAI: \n\nTool: [{"rows": [], "metadata": {"column\_count": 125, "job": "431323/QUSER/QZDASOINIT", "columns": [{"name": "ORDINAL\_POSITION", "type": "INTEGER", "display\_size": 10, "label": "Ordinal Position", "is\_nullable": "NO"}, {"name": "JOB\_NAME", "type": "VARCHAR", "display\_size": 28, "label": "Job Name", "is\_nullable": "NO"}, {"name": "JOB\_NAME\_SHORT", "type": "VARCHAR", "display\_size": 10, "label": "Job Name Short", "is\_nullable": "NO"}, {"name": "JOB\_USER", "type": "VARCHAR", "display\_size": 10, "label": "Job User", "is\_nullable": "NO"}, {"name": "JOB\_NUMBER", "type": "VARCHAR", "display\_size": 6, "label": "Job Number", "is\_nullable": "NO"}, {"name": "INTERNAL\_JOB\_ID", "type": "BINARY", "display\_size": 10, "label": "Internal Job ID", "is\_nullable": "NO"}, {"name": "SUBSYSTEM", "type": "VARCHAR", "display\_size": 10, "label": "Subsystem", "is\_nullable": "NO"}, {"name": "SUBSYSTEM\_LIBRARY\_NAME", "type": "VARCHAR", "display\_size": 10, "label": "Subsystem Library Name", "is\_nullable": "NO"}, {"name": "AUTHORIZATION\_NAME", "type": "VARCHAR", "display\_size": 10, "label": "Authorization Name", "is\_nullable": "NO"}, {"name": "JOB\_TYPE", "type": "VARCHAR", "display\_size": 3, "label": "Job Type", "is\_nullable": "NO"}, {"name": "FUNCTION\_TYPE", "type": "VARCHAR", "display\_size": 10, "label": "Function Type", "is\_nullable": "NO"}, {"name": "FUNCTION", "type": "VARCHAR", "display\_size": 10, "label": "Function", "is\_nullable": "NO"}, {"name": "JOB\_STATUS", "type": "VARCHAR", "display\_size": 4, "label": "Job Status", "is\_nullable": "NO"}, {"name": "MEMORY\_POOL", "type": "VARCHAR", "display\_size": 9, "label": "Memory Pool", "is\_nullable": "NO"}, {"name": "RUN\_PRIORITY", "type": "INTEGER", "display\_size": 11, "label": "Run Priority", "is\_nullable": "NO"}, {"name": "THREAD\_COUNT", "type": "INTEGER", "display\_size": 11, "label": "Thread Count", "is\_nullable": "NO"}]}

OUTPUT

1 item

```
[
  {
    "response": {
      "generations": [
        {
          "text": "The query returned an empty result set, which suggests there might be an issue with the filter or the table function. Let me try a more direct approach using a simpler SQL query:",
          "generationInfo": {
            "finish_reason": "tool_calls"
          }
        }
      ]
    },
    "tokenUsage": {
      "completionTokens": 84,
      "promptTokens": 7538,
      "totalTokens": 7622
    }
  }
]
```

# Analiza zapytania

AI + n8n + IBM i

The screenshot displays the n8n workflow execution interface. On the left, the 'Logs' panel shows a successful execution in 36.673s with 33,556 tokens. The workflow consists of an 'AI Agent' node followed by a 'Run SQL on IBMi' node, which is currently selected. The main panel shows the 'Run SQL on IBMi' node's execution details, including the input SQL query and the resulting JSON output.

**Logs:** Success in 36.673s | 33,556 Tokens

**Workflow:** AI Agent → Run SQL on IBMi (selected) → DeepSeek Chat Model

**Run SQL on IBMi: Success in 1.793s**

**INPUT:** SQL

```
SELECT JOB_NAME, JOB_USER,
JOB_NUMBER, JOB_STATUS FROM
TABLE(QSYS2.ACTIVE_JOB_INFO()) AS X
```

**OUTPUT:** 1 item

```
[
  {
    "response": [
      {
        "rows": [
          {
            "JOB_NAME": "000000/QSYS/SCPF",
            "JOB_USER": "QSYS",
            "JOB_NUMBER": "000000",
            "JOB_STATUS": "EVTW"
          },
          {
            "JOB_NAME": "128669/QSYS/QSYSARB3",
            "JOB_USER": "QSYS",
            "JOB_NUMBER": "128669",
            "JOB_STATUS": "EVTW"
          },
          {
            "JOB_NAME": "128675/QSYS/QDBSRV01",
            "JOB_USER": "QSYS",
            "JOB_NUMBER": "128675",
            "JOB_STATUS": "EVTW"
          }
        ]
      }
    ]
  }
]
```

# Analiza zapytania

AI + n8n + IBM i

Logs

Clear execution

Success in 36.673s | 33,556 Tokens

When chat message received

AI Agent

DeepSeek Chat Model

Run SQL on IBMi

DeepSeek Chat Model

Run SQL on IBMi

DeepSeek Chat Model

DeepSeek Chat Model

Success in 17.36s | 25,524 Tokens

Input

Output

...

▼

INPUT

1 item

Human: You are supporting tool. Choose best CL commands and SQL statements for IBM i release: 7.5 to finalize task and execute like on ACS Run SQL (or STRSQL / ODBC) on user with no \*ALLOBJ. Optimize and minimize SQL and command output if not needed. For every table function use TABLE(table function) to get data like SELECT \* FROM TABLE (QSYS2.OBJECT\_STATISTICS('MJATST','JRN')) AS X.\n\n Question: List all active job\n AI: \n Tool: [{"rows": [], "metadata": {"column\_count": 125, "job": "431323/QUSER/QZDASOINIT", "columns": [{"name": "ORDINAL\_POSITION", "type": "INTEGER", "display\_size": 10, "label": "Ordinal Position"}, {"name": "JOB\_NAME", "type": "VARCHAR", "display\_size": 28, "label": "Job Name"}, {"name": "JOB\_NAME\_SHORT", "type": "VARCHAR", "display\_size": 10, "label": "Job Name Short"}, {"name": "JOB\_USER", "type": "VARCHAR", "display\_size": 10, "label": "Job User"}, {"name": "JOB\_NUMBER", "type": "VARCHAR", "display\_size": 6, "label": "Job Number"}, {"name": "INTERNAL\_JOB\_ID", "type": "BINARY", "display\_size": 10, "label": "Internal Job ID"}, {"name": "SUBSYSTEM", "type": "VARCHAR", "display\_size": 10, "label": "Subsystem"}, {"name": "SUBSYSTEM\_LIBRARY\_NAME", "type": "VARCHAR", "display\_size": 10, "label": "Subsystem Library Name"}, {"name": "AUTHORIZATION\_NAME", "type": "VARCHAR", "display\_size": 10, "label": "Authorization Name"}, {"name": "JOB\_TYPE", "type": "VARCHAR", "display\_size": 3, "label": "Job Type"}, {"name": "FUNCTION\_TYPE", "type": "VARCHAR", "display\_size": 10, "label": "Function Type"}, {"name": "FUNCTION", "type": "VARCHAR", "display\_size": 10, "label": "Function"}, {"name": "JOB\_STATUS", "type": "VARCHAR", "display\_size": 4, "label": "Job Status"}, {"name": "MEMORY\_POOL", "type": "VARCHAR", "display\_size": 9, "label": "Memory Pool"}, {"name": "RUN\_PRIORITY", "type": "INTEGER", "display\_size": 11, "label": "Run Priority"}, {"name": "THREAD\_COUNT", "type": "INTEGER", "display\_size": 11, "label": "Thread Count"}]}

OUTPUT

1 item

I've successfully retrieved the list of all active jobs on your IBM i system. Here's a summary of what was found:

### Active Jobs Summary

The query returned **333 active jobs** currently running on your IBM i 7.5 system. The results show:

- Job Name** (format: job\_number/user/job\_name)
- Job User** (user profile that owns the job)
- Job Number** (system-assigned job identifier)
- Job Status** (current status of each job)

**Key Observations:**

- System Jobs:** Many system-level jobs from QSYS, QUSER, and other system users
- Batch Jobs:** Several QBATCH jobs in DEQW status (waiting on job queue)
- Interactive Jobs:** Multiple QINTER jobs and user sessions
- Database Jobs:** QDBSRV\* jobs for database services
- Network Services:** TCP/IP jobs (QTCP), web services (QTMHHTTP), and other network services
- User Sessions:** Various user interactive sessions (QPAD\* jobs) in DSPW status (display wait)

# Odnośniki

- [AI Workflow Automation Platform & Tools - n8n](#)
- [Docker | n8n Docs](#)
- [marcin-ogon/n8n-nodes-ibmi-mapepire: n8n community nodes to run IBM i SQL and CL commands via Mapepire](#)
- [PUB400.COM - Your public IBM i server](#)

**V O L V O**