

# **Setup n8n For OEE Monitoring System**

**Yaser Ali Husen**

# **n8n Installation**

## **Linux (VPS)**

# n8n Installation - Linux

## 1. Update & Upgrade

```
sudo apt update && sudo apt upgrade -y
```

Source for Linux VPS: <https://www.youtube.com/watch?v=R6HsF8Ndeas>

## 2. Create swap memory (For low memory only)

```
sudo fallocate -l 4G /swapfile
```

```
sudo chmod 600 /swapfile
```

```
sudo mkswap /swapfile
```

```
sudo swapon /swapfile
```

```
# Add to /etc/fstab for persistensi:
```

```
echo '/swapfile none swap sw 0 0' | sudo tee -a /etc/fstab
```

## 3. Install Docker

Instal required packages

```
sudo apt install -y apt-transport-https ca-certificates curl gnupg
```

## 4. Add GPG key Docker

```
sudo install -m 0755 -d /etc/apt/keyrings
```

```
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o  
/etc/apt/keyrings/docker.gpg
```

```
sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

# n8n Installation - Linux

## 5. Setup repository

```
echo \  
  "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg]  
  https://download.docker.com/linux/debian \  
  "$(./etc/os-release && echo "$VERSION_CODENAME")" stable" | \  
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

## 6. Update dan instal Docker

```
sudo apt update  
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

## 7. Verification Installation

```
sudo docker run hello-world
```

# n8n Installation - Linux

#Run n8n in Docker-----

```
mkdir -p ~/n8n_data
```

8. Run n8n container with memory limitation

```
sudo docker run -d \
--name n8n \
--restart unless-stopped \
--memory=1g \
--memory-swap=1.5g \
-p 5678:5678 \
-v ~/n8n_data:/home/node/.n8n \
-e N8N_SECURE_COOKIE=false \
-e N8N_BASIC_AUTH_ACTIVE=true \
-e N8N_RUNNERS_ENABLED=true \
n8nio/n8n
```

#Check docker

```
sudo docker ps
```

```
sudo docker logs n8n
```

#Test using curl

```
curl -v http://1<IP_Address>:5678
```

# n8n Installation - Linux

Trouble shooting-----

#If want to reset or delete

```
sudo docker stop n8n && sudo docker rm n8n
```

```
sudo rm -rf ~/n8n_data
```

```
mkdir ~/n8n_data
```

```
sudo docker stop n8n && sudo docker rm n8n
```

# Run n8n container#Change port to 80, some vps blocks port 5678

```
sudo docker stop n8n && sudo docker rm n8n
```

```
sudo docker run -d \
```

```
--name n8n \
```

```
--restart unless-stopped \
```

```
--memory=1g \
```

```
--memory-swap=1.5g \
```

```
-p 80:5678 \
```

```
-e N8N_PORT=5678 \
```

```
-e N8N_HOST=0.0.0.0 \
```

```
-v ~/n8n_data:/home/node/.n8n \
```

```
-e N8N_SECURE_COOKIE=false \
```

```
-e N8N_BASIC_AUTH_ACTIVE=true \
```

```
-e N8N_RUNNERS_ENABLED=true \
```

```
n8nio/n8n
```

# n8n Installation Windows

# n8n Installation – Windows (With Docker)

1. Download Docker Desktop for Windows

Website: <https://docs.docker.com/desktop/setup/install/windows-install/>

2. Install Docker Desktop

3. Install n8n in Docker Desktop

# n8n Installation – Windows (Without Docker)

1. Install Nodejs. (version: 20.19 or 22.19)

Website: <https://nodejs.org/en/download>

2. Install n8n

Open cmd prompt:

*npm install n8n -g*

3. Run n8n

In cmd prompt:

*n8n*

4. Open n8n in Browser

Address: 127.0.0.1:5678

# Signup Chat Model

# Gemini AI

1. Visit website: <https://aistudio.google.com/>
2. Create API Key

The screenshot shows the Google AI Studio interface for managing API keys. On the left, there's a sidebar with options like 'API Keys' (which is selected and highlighted in blue), 'Usage & Billing', and 'Changelog'. The main content area is titled 'API Keys' and contains a section for testing the Gemini API with a 'curl' command example. Below this, a large red circle highlights the 'Create API key' button, which is located in the top right corner of the main content area. At the bottom, there's a table showing a list of API keys, with columns for 'Project number', 'Project name', 'API key', 'Created', and 'Plan'. A 'Look up API Key for project' button is also present at the bottom.

View status	Project number	Project name	API key	Created	Plan
<					

1. Visit website: <https://console.groq.com/>
2. Create API Key

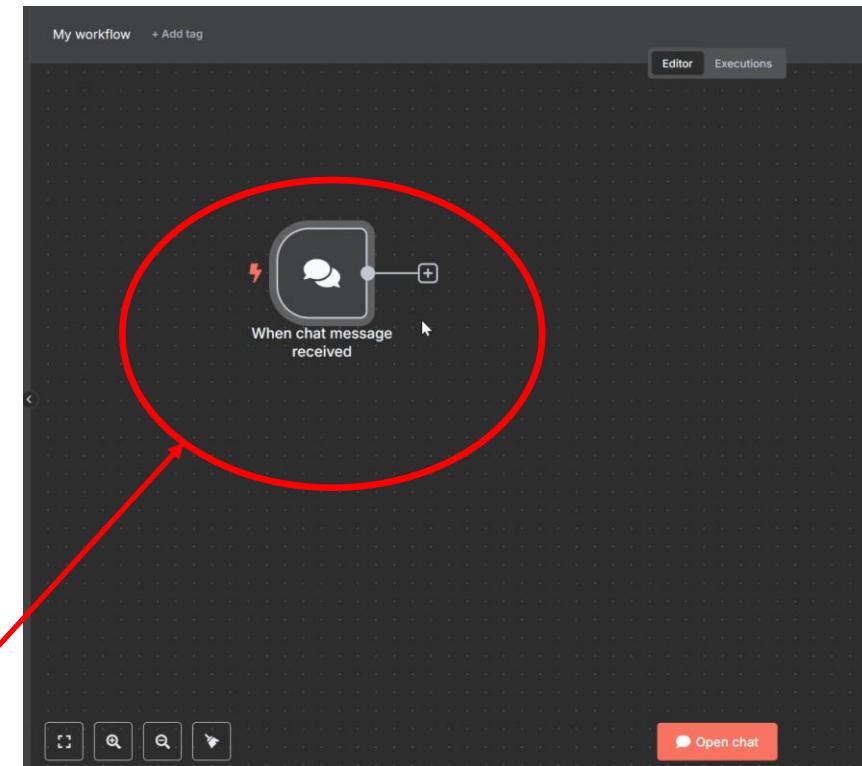
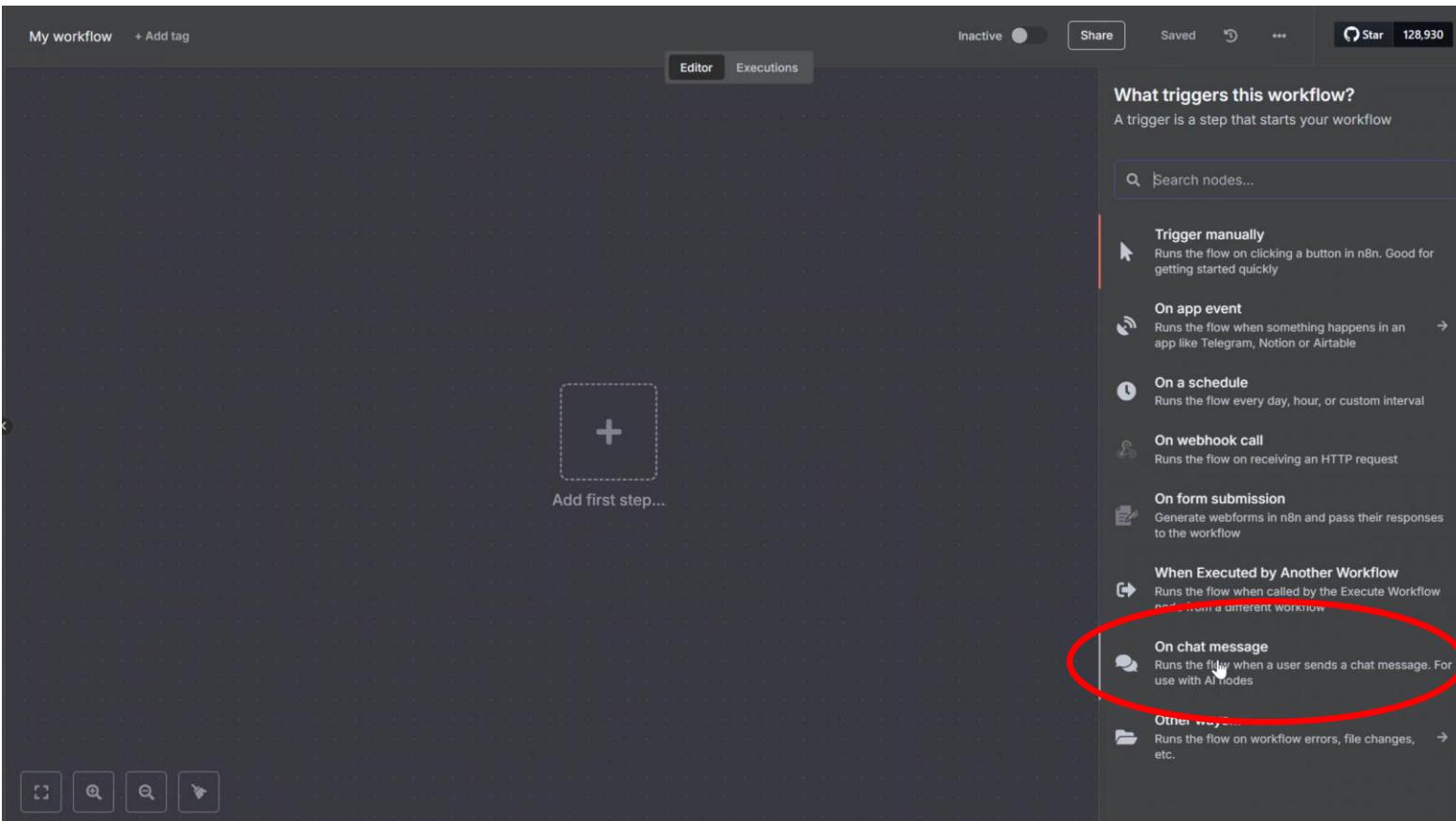
The screenshot shows the Groq AI console interface. At the top, there is a navigation bar with icons for back, forward, refresh, and search, followed by the URL 'console.groq.com/keys'. To the right of the URL are a star icon, a folder icon, a user profile icon, and a three-dot menu icon. Below the navigation bar, the header includes 'Personal' (with a red '9' badge), 'Default Project', and tabs for 'Playground', 'API Keys' (which is highlighted in red), 'Dashboard', 'Docs', and a gear icon. A red oval highlights the 'Create API Key' button, which is located in the top right corner of the main content area. The main content area is titled 'API Keys' and contains a sub-instruction: 'Manage your project API keys. Remember to keep your API keys safe to prevent unauthorized access.' Below this, there is a table with columns: NAME, SECRET KEY, CREATED, LAST USED, and USAGE (24HRS). One row is visible, showing 'n8n\_oee', 'gsk\_...Fp10', '8/9/2025', '8/9/2025', and '0 API Calls'. To the right of this row are two small icons: a pencil and a trash can.

NAME	SECRET KEY	CREATED	LAST USED	USAGE (24HRS)
n8n_oee	gsk_...Fp10	8/9/2025	8/9/2025	0 API Calls

# n8n Flow

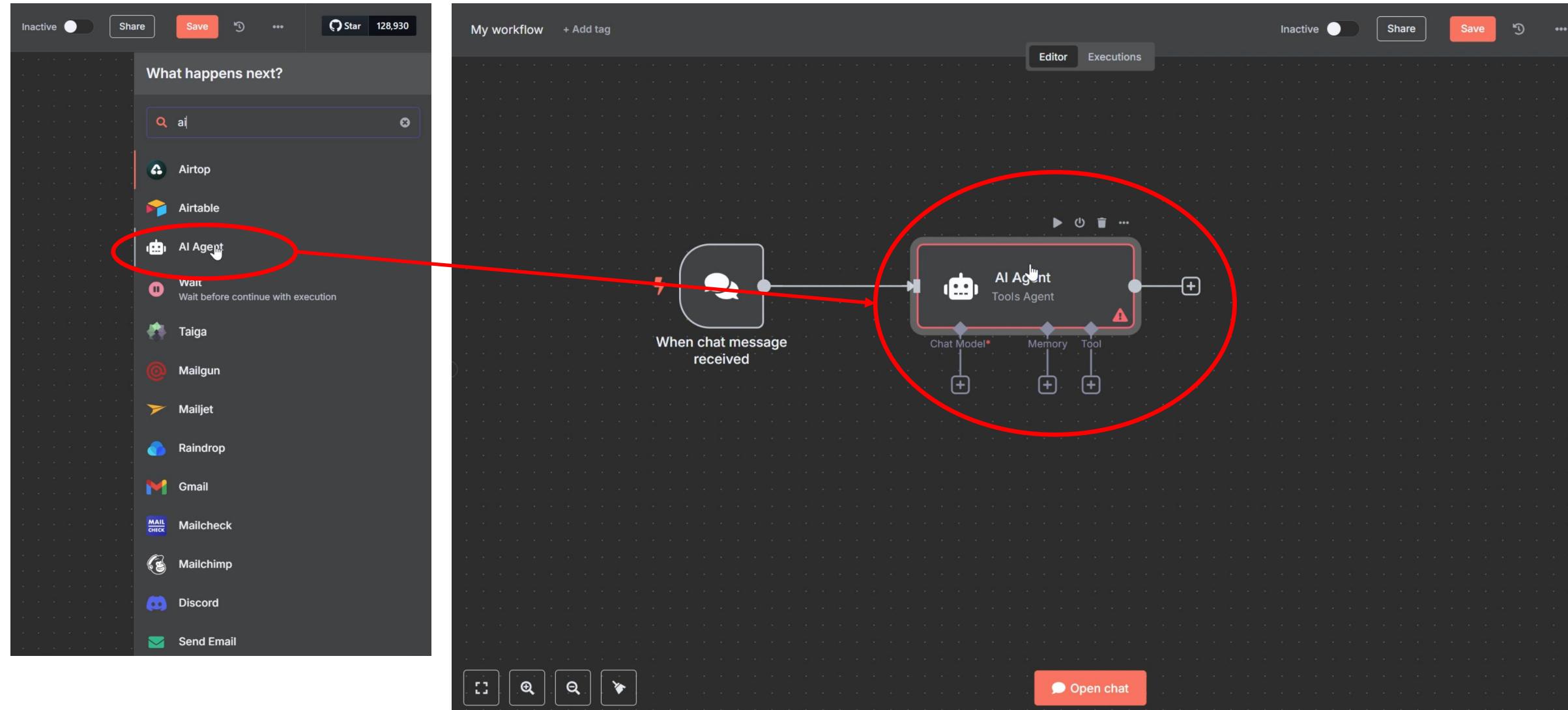
# n8n Flow - Chat

## 1. Add On chat message



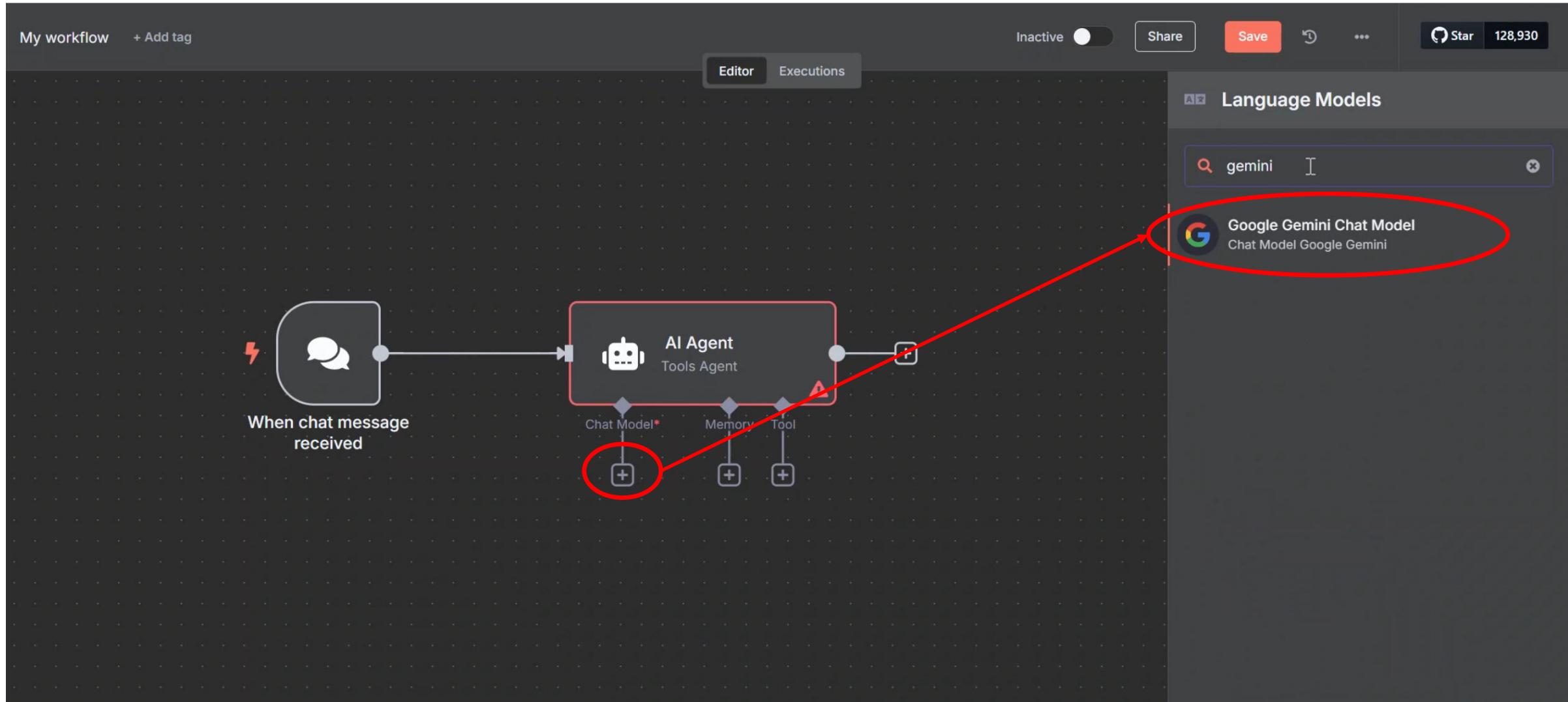
# n8n Flow – AI Agent

## 2. Add AI Agent



# n8n Flow – Chat Model

## 3. Add AI Chat Model



# n8n Flow – Chat Model

## Chat Model Configuration

AI Model:

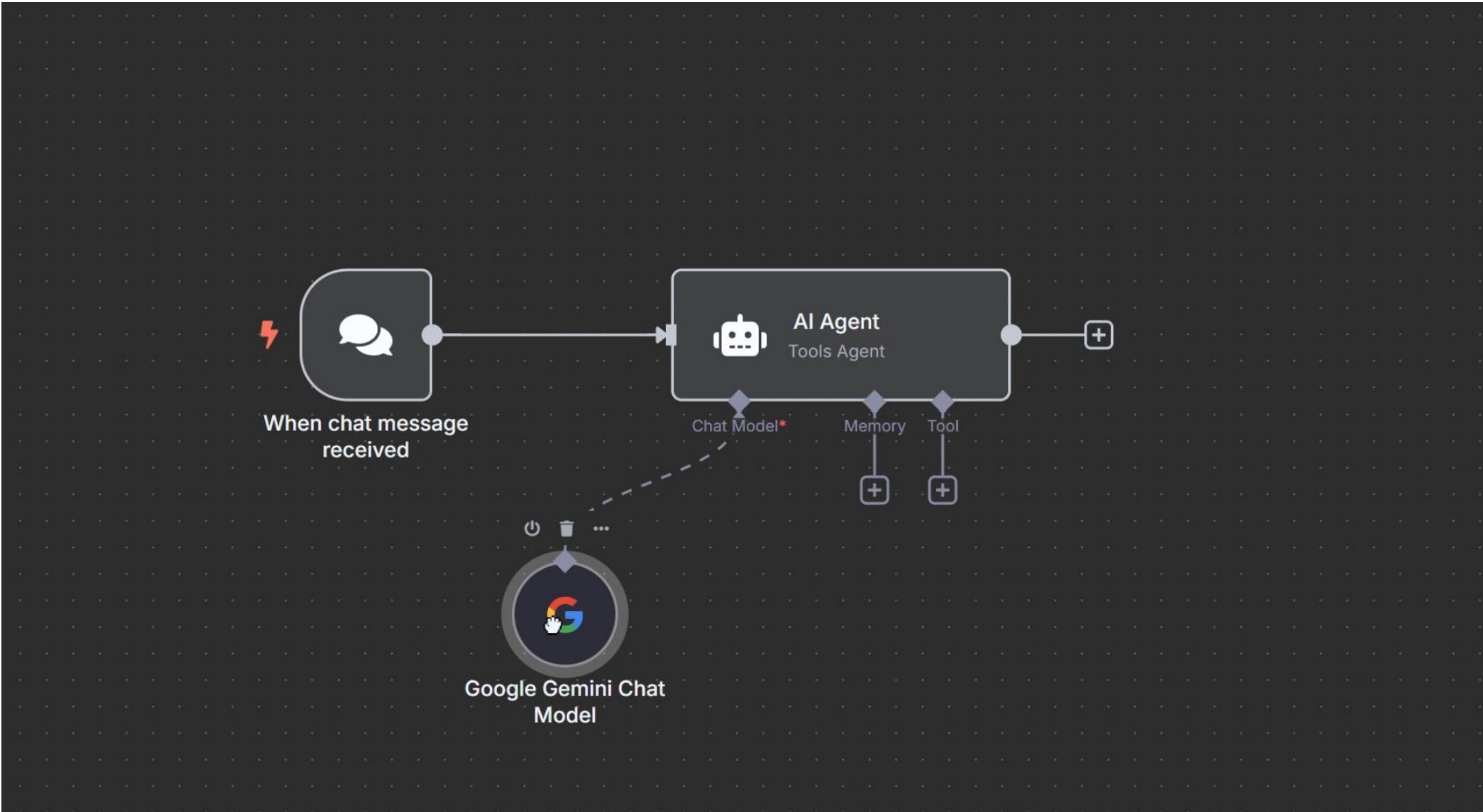
The screenshot shows the 'Parameters' tab of the 'Google Gemini Chat Model' configuration. It includes a 'Credential to connect with' dropdown set to 'Google Gemini(PaLM) Api account', a 'Model' dropdown set to 'models/gemini-1.0-pro', and a detailed description of the model. A red circle highlights the 'models/gemini-2.0-flash' option in the list below, which is described as 'Gemini 2.0 Flash'. A cursor icon is positioned over this highlighted option.

Temperature:

The screenshot shows the 'Parameters' tab of the 'Google Gemini Chat Model' configuration. It includes a 'Credential to connect with' dropdown set to 'Google Gemini(PaLM) Api account'. In the 'Options' section, there is a 'Sampling Temperature' input field set to '0.4', which is circled in red. A red arrow points from this circled field to the 'Host' field in the credential configuration window shown below.

The screenshot shows the 'Google Gemini(PaLM) Api account' configuration window. It displays a green status bar indicating 'Connection tested successfully'. Below it, there are sections for 'Connection' (with a green checkmark), 'Sharing', and 'Details'. The 'Details' section contains fields for 'Host' (set to 'https://generativelanguage.googleapis.com') and 'API Key' (represented by a series of dots). A red oval highlights the 'API Key' field. Another red arrow points from the 'Host' field in the credential configuration window above to this highlighted 'API Key' field.

# n8n Flow – Chat Model



# n8n Flow - Memory

## 4. Add Memory

The screenshot shows the n8n Flow interface with a dark theme. On the left, a workflow is displayed:

- A trigger node "When chat message received" (represented by a speech bubble icon with a lightning bolt) is connected to an "AI Agent Tools Agent" node.
- The "AI Agent Tools Agent" node has three outgoing connections:
  - "Chat Model\*" (indicated by a dashed arrow pointing to a "Google Gemini Chat Model" node below)
  - "Memory" (indicated by a solid arrow pointing to a "Memory" node)
  - "Tool" (indicated by a solid arrow pointing to a "Tool" node)
- The "Memory" and "Tool" nodes each have a plus sign icon, indicating they can be expanded or added.

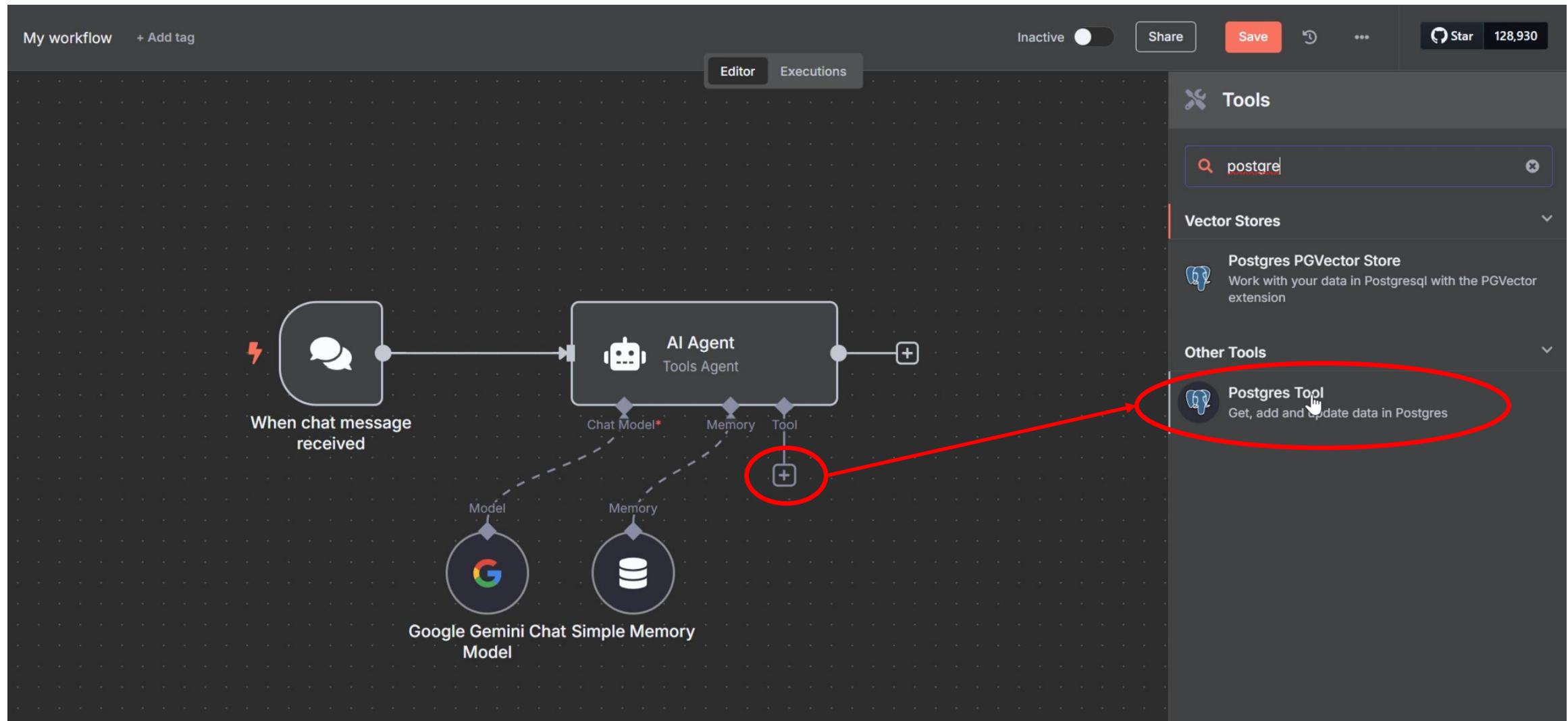
A red circle highlights the plus sign icon on the "Memory" node. A red arrow points from this highlighted area to the "Simple Memory" section in the sidebar.

The sidebar on the right is titled "Memory" and contains the following information:

- Memory**: Memory allows an AI model to remember and reference past interactions with it.
- For beginners** (highlighted with a red oval):
  - Simple Memory**: Stores in n8n memory, so no credentials required.
- Other memories**:
  - MongoDB Chat Memory**: Stores the chat history in MongoDB collection.
  - Motorhead**: Use Motorhead Memory.
  - Postgres Chat Memory**: Stores the chat history in Postgres table.
  - Redis Chat Memory**: Stores the chat history in Redis.
  - Xata**: Use Xata Memory.
  - Zep**: Use Zep Memory.

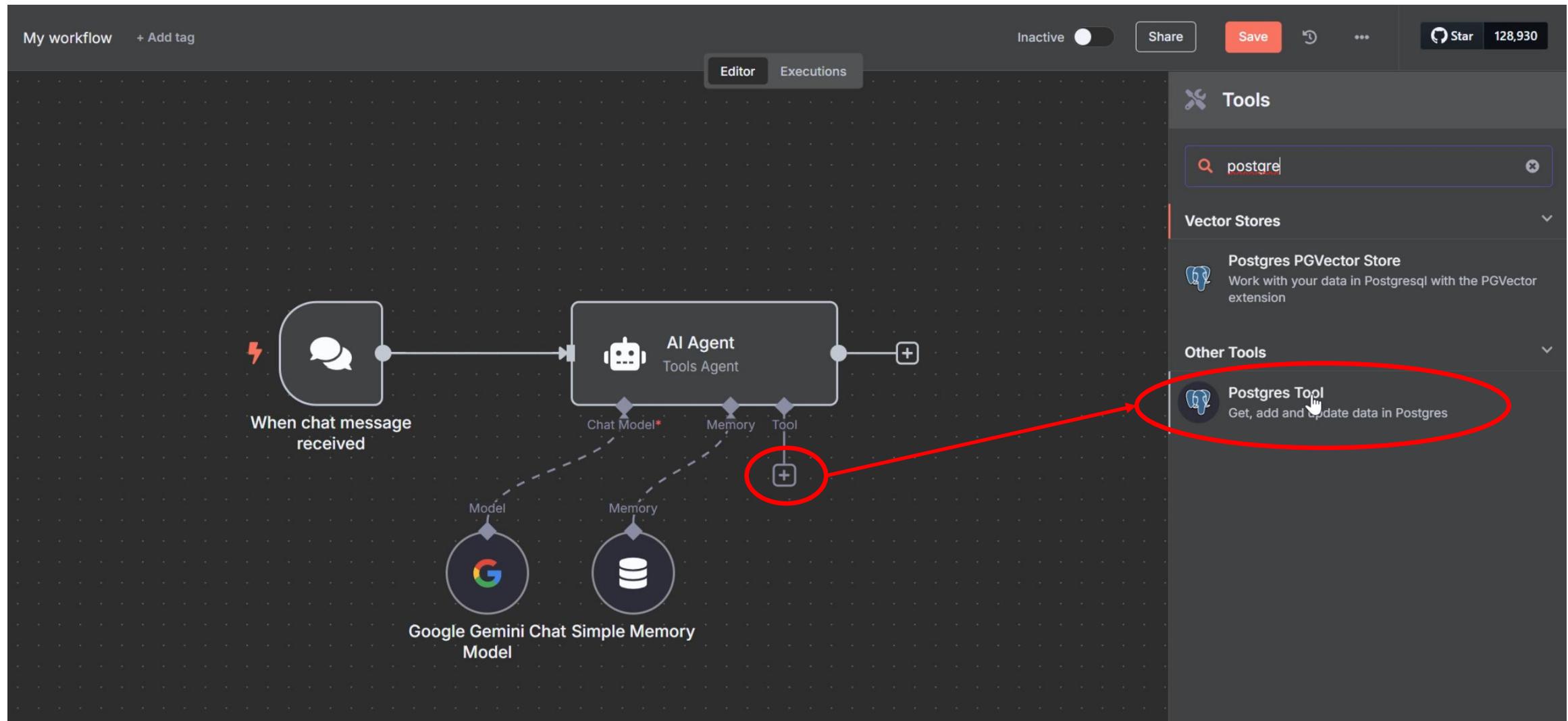
# n8n Flow – Tool: Table List

## 5. Add Tool: Table List



# n8n Flow – Tool: Table List

## 5. Add Tool: Table List



# n8n Flow – Tool: Table List

## Table List Configuration

The screenshot shows the 'Table List Configuration' screen for the 'Postgres' tool. At the top, there's a 'Postgres' icon, a 'Test step' button, and a 'Docs' link. Below that, there are tabs for 'Parameters' (which is selected) and 'Settings'. Under 'Parameters', it says 'Credential to connect with' and 'Postgres account'. In the 'Tool Description' section, 'Set Manually' is selected. The 'Description' section contains a button labeled 'Get Tables List'. The 'Operation' section has a 'Execute Query' button. The 'Query' section contains a code editor with the following SQL query:

```
1 SELECT table_name from information_schema.tables where table_schem
```

The 'Result' section shows the output of the query:

```
SELECT table_name from information_schema.tables where table_schema = 'p
```

A tip at the bottom says: 'Tip: Anything inside {{ }} is JavaScript. Learn more.'

At the bottom right of the main window is a 'Add option' button.

If using n8n on docker, please user database IP address instead localhost  
Get IP Cmd: ip route

## Database Connection:

The screenshot shows the 'Postgres account' configuration screen. It has tabs for 'Connection' (selected), 'Sharing', and 'Details'. The 'Connection' tab shows a green success message: 'Connection tested successfully'. Below it, there's a note: 'Need help filling out these fields? Open docs'. The 'Details' tab shows the following connection settings:

- Host: 103.172.204.67
- Database: oee
- User: postgres
- Password: (redacted)
- Maximum Number of Connections: (redacted)

A red arrow points from the 'Edit' icon in the top right of the main configuration window to the 'Edit' icon in the top right of this configuration window.

# n8n Flow – Tool: Table List

## Table List Configuration

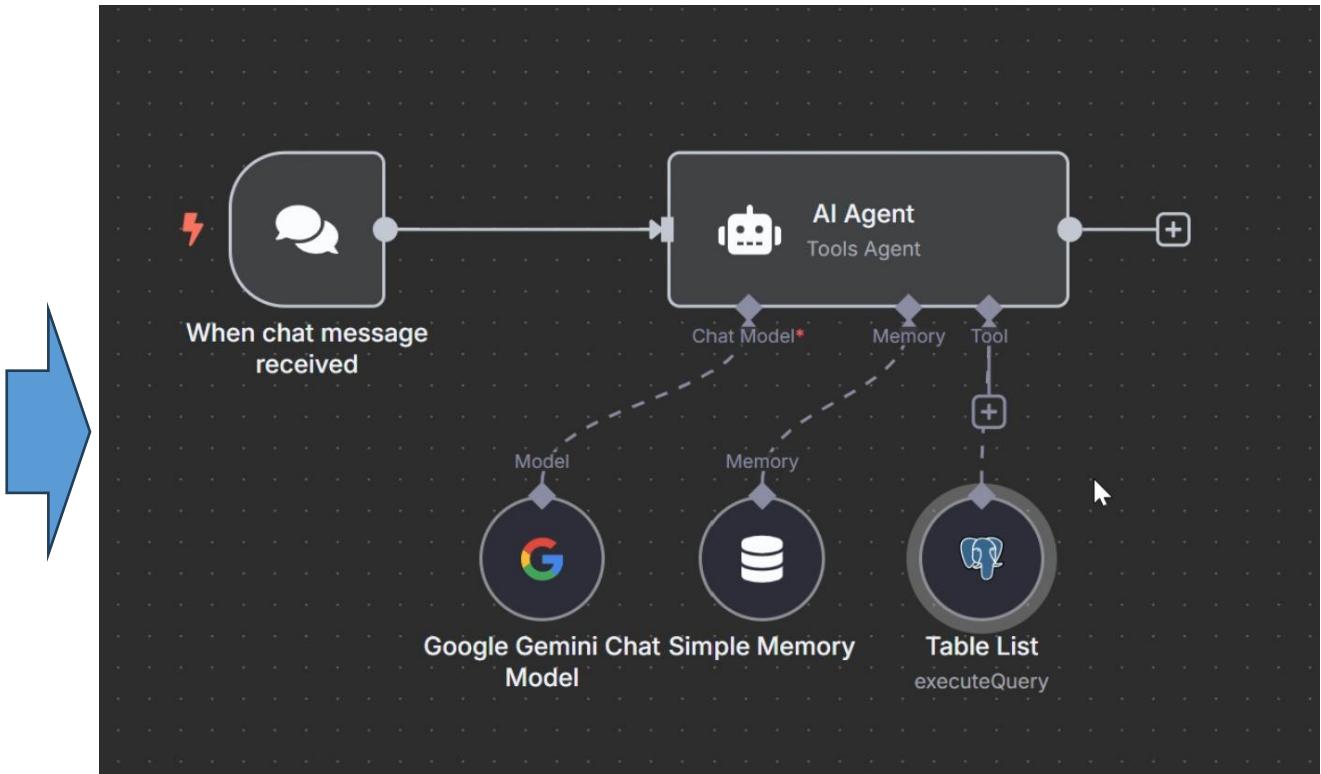
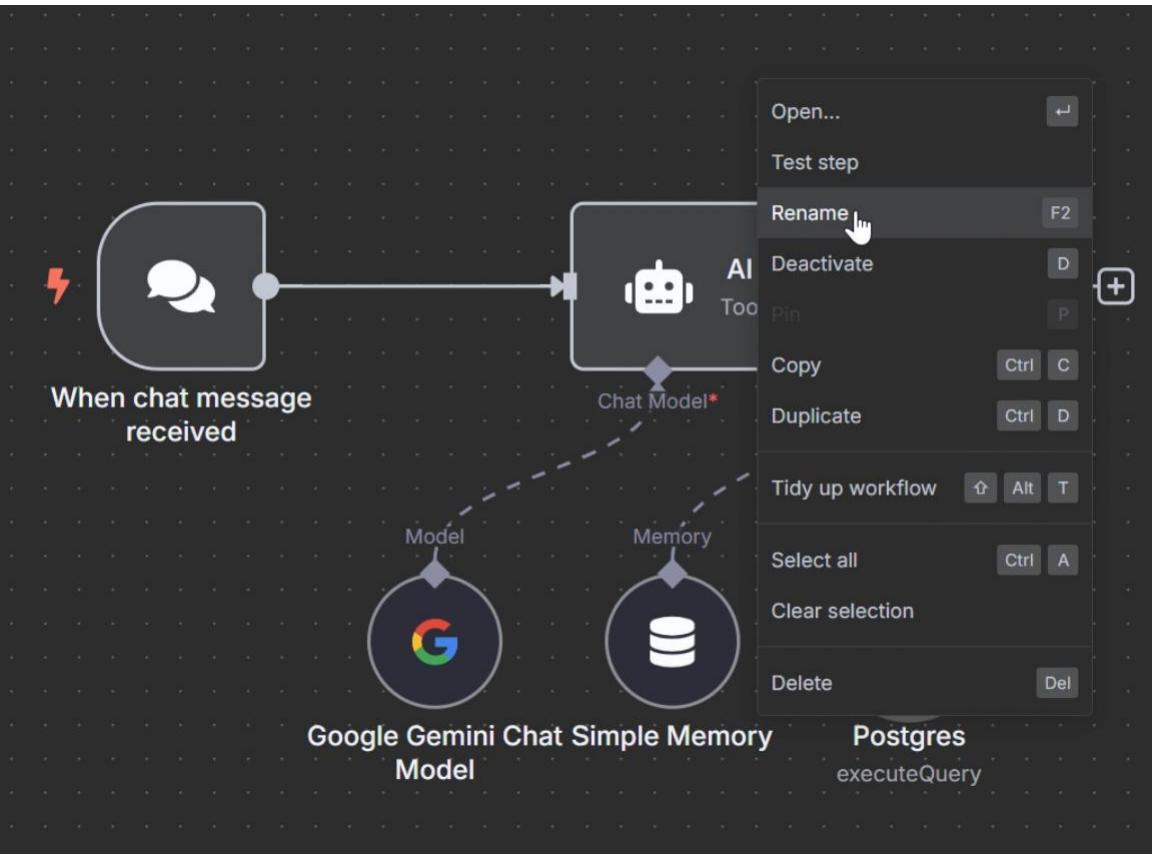
The screenshot shows the configuration interface for a 'Table List' tool in n8n. The top navigation bar includes a 'Postgres' icon, a 'Test step' button, and a 'Docs' link. Below the navigation, there are tabs for 'Parameters' (which is selected) and 'Settings'. The 'Parameters' section contains fields for 'Credential to connect with' (set to 'Postgres account'), 'Tool Description' (set to 'Set Manually'), 'Description' (set to 'Get Tables List'), 'Operation' (set to 'Execute Query'), and a 'Query' section. The 'Query' section displays the SQL command: `SELECT table_name from information_schema.tables where table_schema = 'public'`. A 'Result' section below shows the same query again. A tip at the bottom states: 'Tip: Anything inside {{ }} is JavaScript. Learn more'.

Query:

*SELECT table\_name from information\_schema.tables  
where table\_schema = 'public'*

# n8n Flow – Tool: Table List

## Rename Tool



# n8n Flow – Tool: Table Schema

## 6. Add Tool: Table Schema

The screenshot shows the configuration interface for the 'Table Schema' tool in n8n. The 'Parameters' tab is selected. Under 'Credential to connect with', 'Postgres account' is chosen. The 'Tool Description' field contains 'Set Manually'. The 'Description' field has 'Read Table Schema' selected. The 'Operation' field is set to 'Execute Query'. In the 'Query' section, the SQL query 'SELECT column\_name, data\_type from information\_schema.columns WHERE table\_name=\$1' is entered. Below the query, a note says 'Consider using query parameters to prevent SQL injection attacks. Add them in the options below'. In the 'Options' section, under 'Query Parameters', there is a field containing the expression '{{ \$fromAI('table\_name','name of table to select') }}'. There are also 'Fixed' and 'Expression' buttons, and a 'Add option' button.

Query:

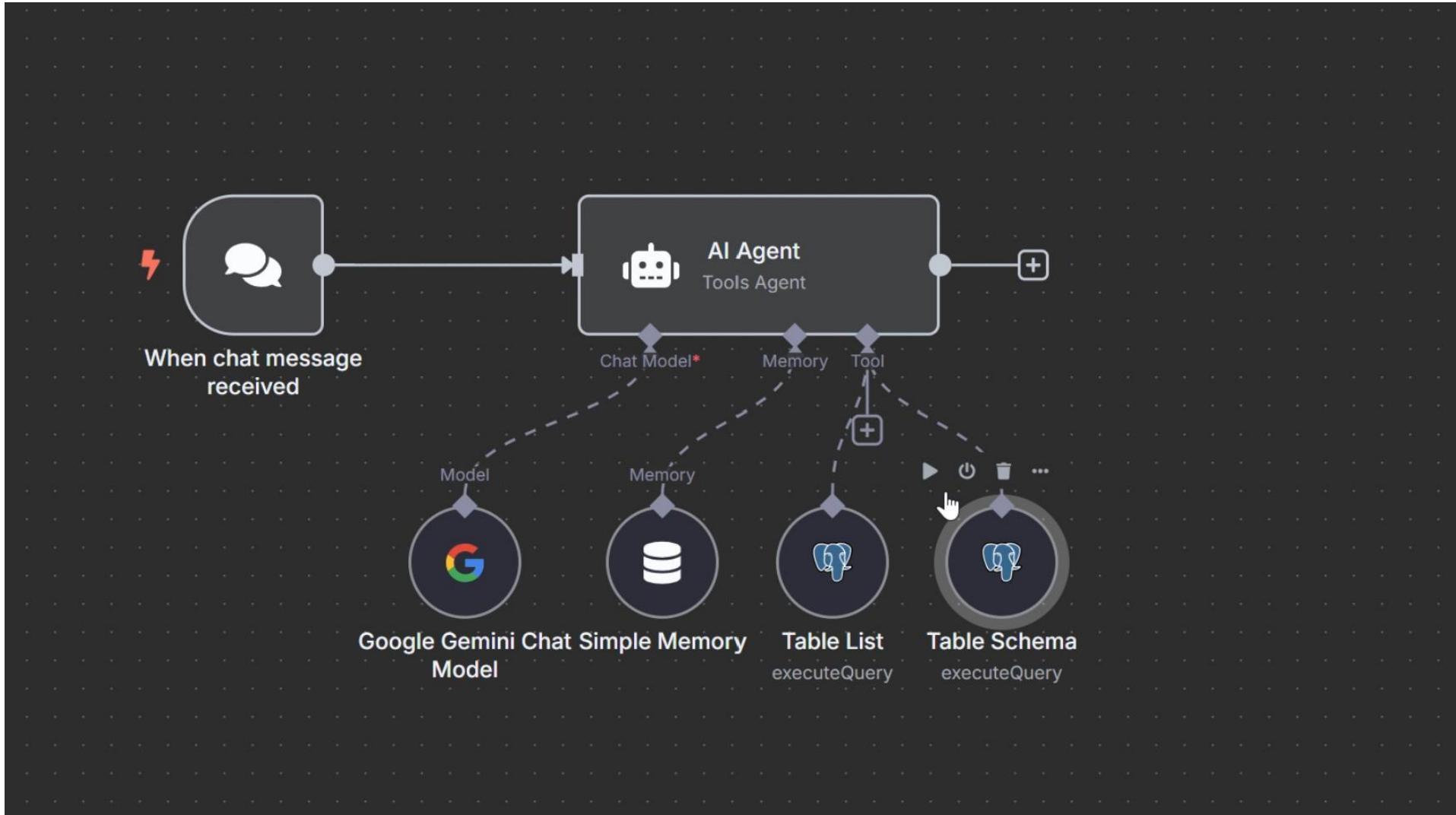
*SELECT column\_name, data\_type from information\_schema.columns WHERE table\_name=\$1*

Add option → Query Parameters:

*{{ \$fromAI('table\_name','name of table to select') }}*

# n8n Flow – Tool: Table Schema

## Table Schema



# n8n Flow – Tool: Execute Query

## 7. Add Tool: Execute Query

The screenshot shows the n8n Flow interface for configuring a 'Postgres' tool. The 'Parameters' tab is selected. Under 'Credential to connect with', 'Postgres account' is chosen. In the 'Tool Description' section, 'Set Manually' is selected. The 'Description' field contains 'Execute Query'. The 'Operation' dropdown is set to 'Execute Query'. The 'Query' section displays the following code:

```
1 v SELECT {{ $fromAI("fields") }}  
2   FROM {{ $fromAI('table_name','name of table to select') }}  
3   {{ $fromAI("condition") && $fromAI("condition").trim() ? "WHERE " + $fromAI("condition")  
4 : "" }}  
4   {{ $fromAI("group_by") && $fromAI("group_by").trim() ? "GROUP BY " + $fromAI("group_by")  
5 : "" }}  
5
```

The 'Result' section shows the generated SQL query:

```
SELECT [Execute previous nodes for preview]  
FROM [Execute previous nodes for preview]  
[Execute previous nodes for preview]  
[Execute previous nodes for preview]
```

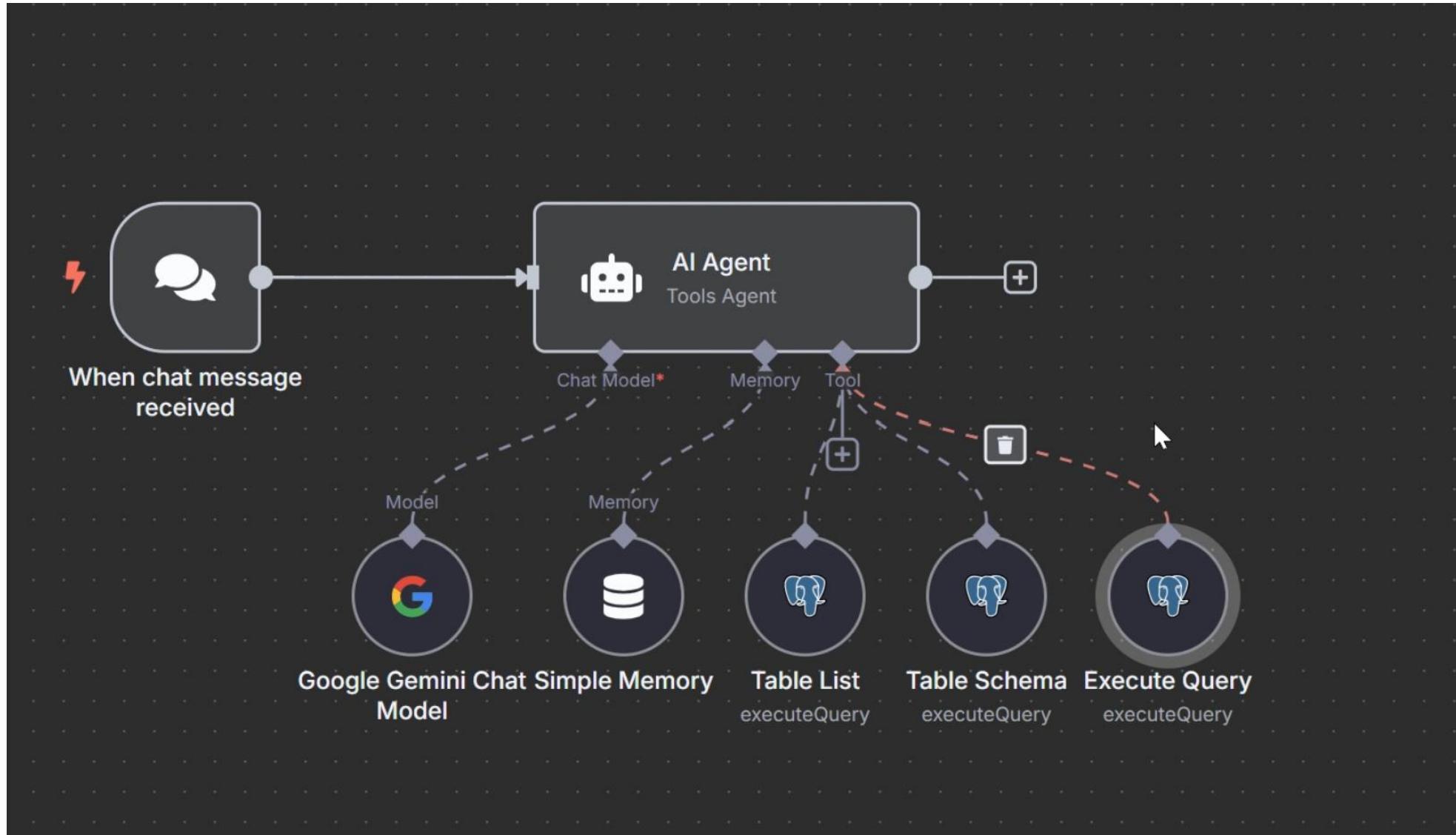
A tip at the bottom states: 'Tip: Anything inside {{ }} is JavaScript. [Learn more](#)'.

Query:

```
SELECT {{ $fromAI("fields") }}FROM  
{{ $fromAI('table_name','name of table to  
select') }}{{ $fromAI("condition") &&  
$fromAI("condition").trim() ? "WHERE " +  
$fromAI("condition") : "" }}{{ $fromAI("group_by") &&  
$fromAI("group_by").trim() ? "GROUP BY " +  
$fromAI("group_by") : "" }}
```

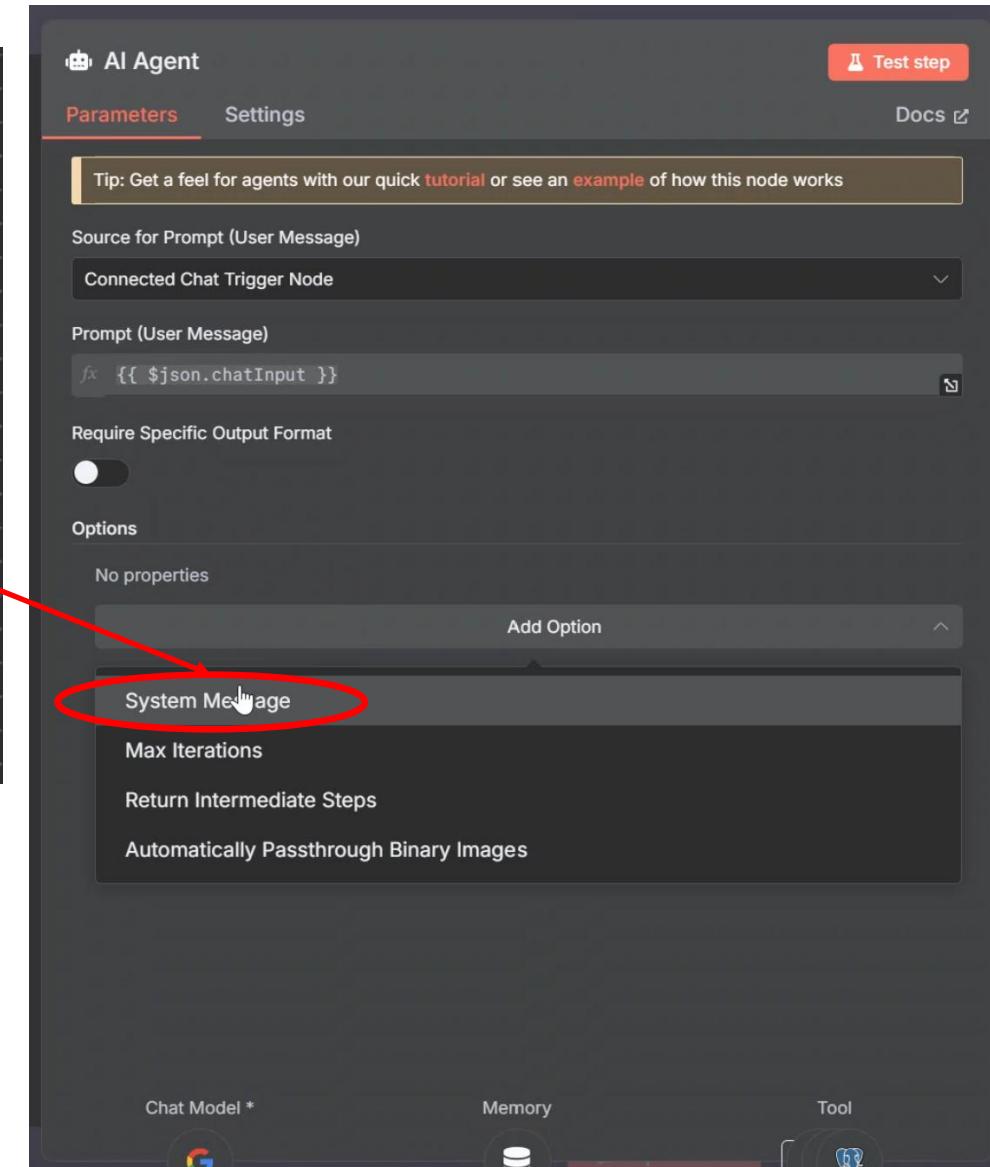
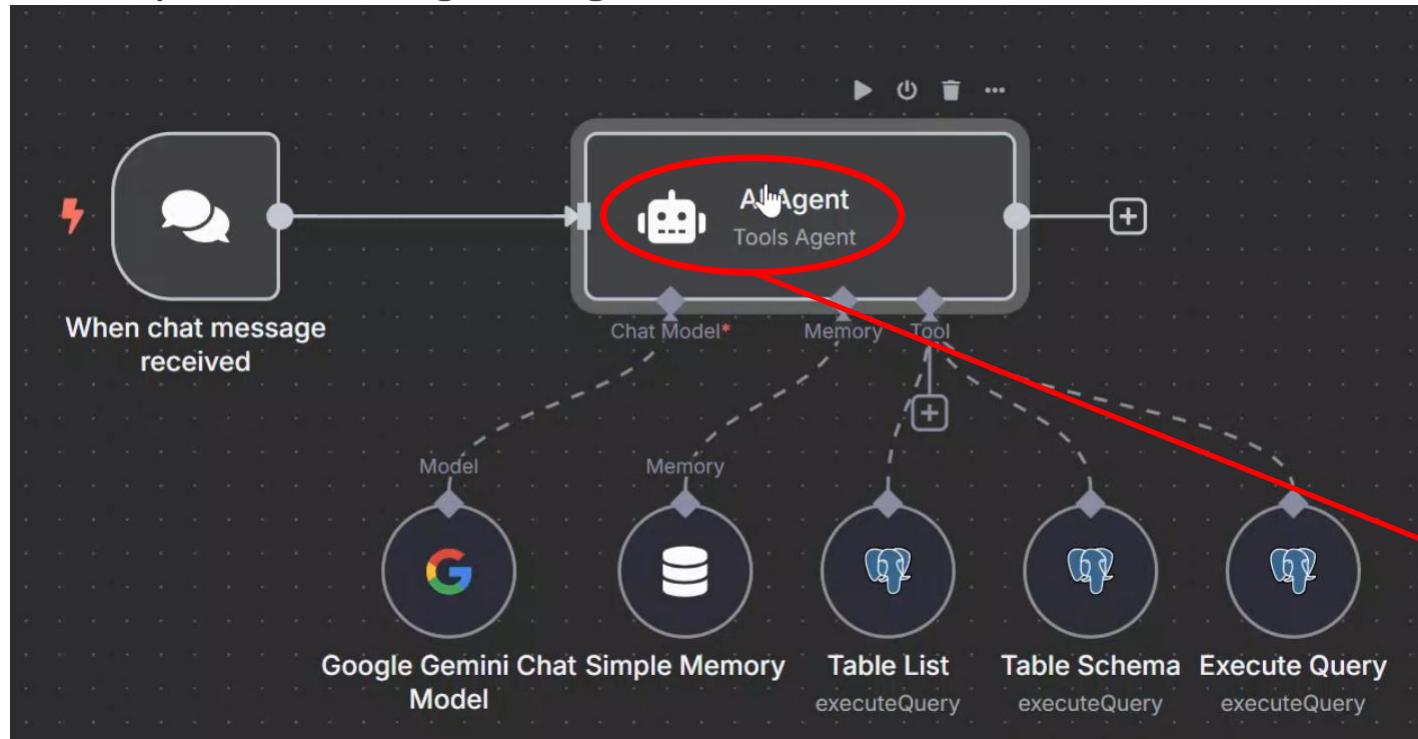
# n8n Flow – Tool: Execute Query

## Execute Query



# n8n Flow – System Message

## 8. Add System Message AI Agent



# n8n Flow – System Message

## System Message AI Agent

The screenshot shows the configuration of an 'AI Agent' node in the n8n Flow interface. The node is set up to receive a 'User Message' from a 'Connected Chat Trigger Node' and generate a response using the expression `{{{ $json.chatInput }}}`. The 'System Message' option is selected, displaying the following text:

You are a data analyst and an OEE calculation expert.  
Use only available tools. If asked for data, query it.

Don't ask the user for the table structure.  
Run Table Schema automatically if necessary to avoid field errors during queries.  
Then, run Execute Query directly to retrieve the results.  
Never issue a query, but provide the final results.

1. To determine the table structure, DO NOT ask the user.  
Run the Table Schema tool directly without confirmation.

2. To read data from the database, use the Execute Query tool.  
This is the only way to read data.  
You MUST run Execute Query every time you want to retrieve values or final results from the database.

Below the message content, there are fields for 'Chat Model \*' (set to Google), 'Memory' (set to a specific node), and 'Tool' (set to a specific node).

# n8n Flow – System Message

## System Message AI Agent (Indonesia):

Kamu adalah data analyst dan ahli perhitungan OEE.

Gunakan hanya tool yang tersedia, jika ditanya data harus melakukan query.

Jangan tanya struktur tabel ke user,

Jalankan Table\_Schema otomatis jika perlu agar tidak salah field saat query, Setelah itu langsung jalankan Execute\_Query untuk ambil hasil, Jangan pernah keluarkan query, tapi berikan hasil akhir.

1. Untuk mengetahui struktur tabel, JANGAN tanya user.

Jalankan tool Table\_Schema secara langsung tanpa konfirmasi.

2. Untuk membaca data dari database, gunakan tool Execute\_Query.

Ini adalah satu-satunya cara untuk membaca data. Kamu HARUS menjalankan Execute\_Query setiap kali ingin mengambil nilai atau hasil akhir dari database.

3. Data OEE ada di tabel oee\_date.

- Saat query di table oee\_date perlu grouping atau where, HARUS menggunakan id\_machine, jangan machine\_name atau machine\_no.

- Jika menggunakan machine\_name atau machine\_no, kamu HARUS JOIN ke machine\_master berdasarkan id\_machine.

- Jika menggunakan fungsi agregasi (SUM, AVG, MAX, MIN, COUNT), pastikan menambahkan GROUP BY untuk kolom non-agregasi yang di-select, gunakan id\_machine jika berdasarkan mesin.

- Jika ditanya berdasarkan nama mesin, ambil nama mesinnya di machine\_master, lakukan join berdasarkan id\_machine.

4. Saat gunakan tool Execute\_Query:

- Jangan satukan klausula WHERE ke dalam 'table\_name'.

- Output harus memiliki key 'condition' untuk bagian WHERE, dan 'table\_name' hanya berisi nama tabel dan JOIN saja.

- Contoh output JSON:

```
{  
  "fields": "COUNT(DISTINCT date)",  
  "table_name": "oee_date JOIN machine_master ON oee_date.id_machine = machine_master.id_machine",  
  "condition": "machine_master.machine_name = 'MACHINING-03' AND EXTRACT(YEAR FROM date) = 2025 AND EXTRACT(MONTH FROM date) = 7",  
  "group_by": ""  
}
```

5. Data id\_machine, machine\_name, machine\_no ada di tabel machine\_master.

6. Jika ingin mengambil data dari tabel oee\_date:

- Pertama, ambil id\_machine dari tabel machine\_master sesuai machine\_name atau machine\_no.

- Setelah itu, gunakan id\_machine tersebut untuk query ke tabel yang dimaksud.

- Jika ingin melakukan group mesin, group berdasarkan id\_machine.

- loading\_time dan stop\_time satunya menit.

- Untuk nilai actual output nama kolomnya qty\_output

- Untuk nilai quantity defect nama kolomnya qty\_defect

7. Saat menjawab pertanyaan user:

- Jika tidak tahu kolom tabel, langsung jalankan Table\_Schema.

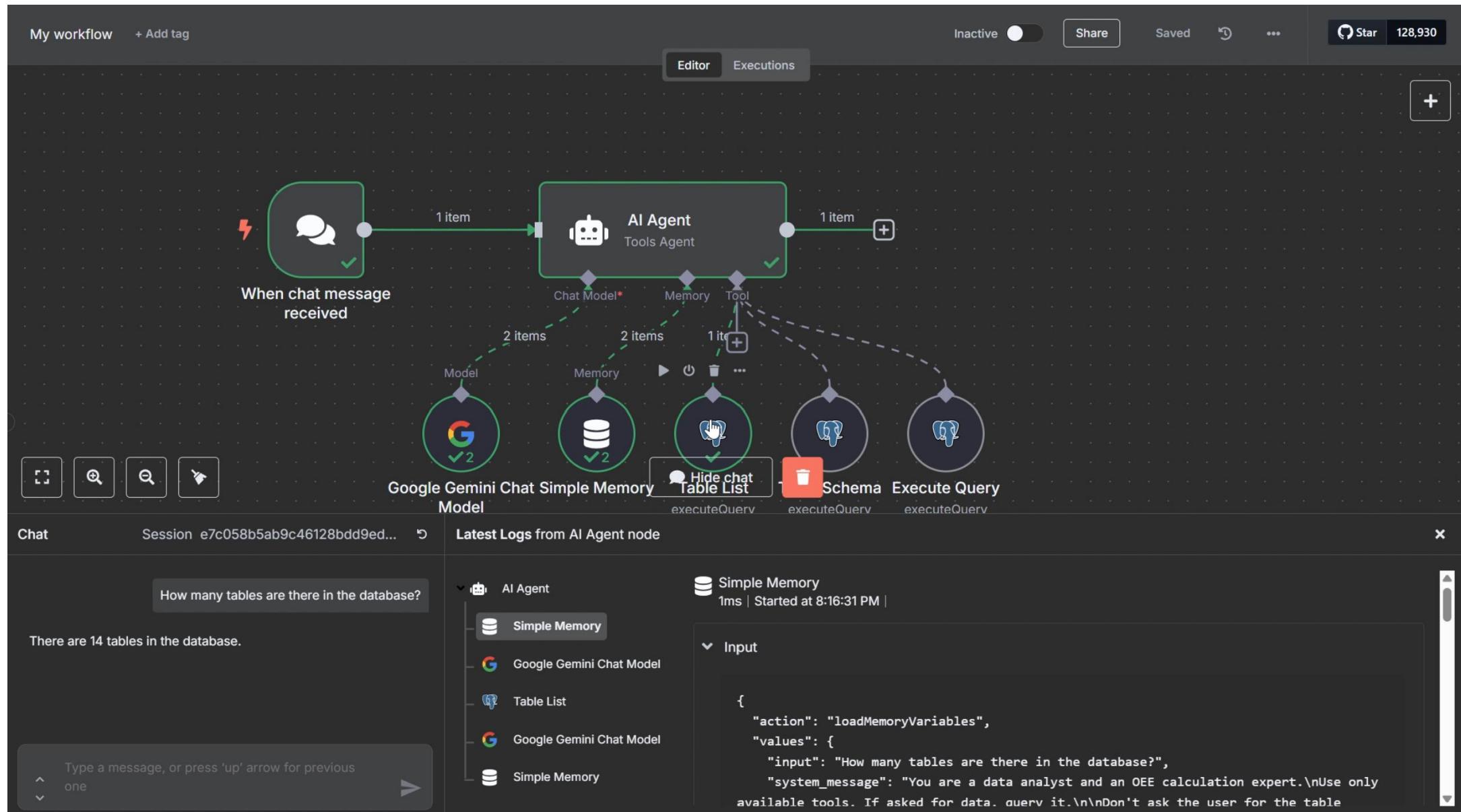
- Setelah tahu kolom, langsung jalankan Execute\_Query.

- Jangan pernah menanyakan struktur tabel atau query ke user.

8. Jika ada fungsi agregasi di fields, wajib sertakan kolom non-agregasi di group\_by.

9. Jika ada filter berdasarkan nama mesin, gunakan join ke machine\_master dan taruh filter di condition menggunakan machine\_master.machine\_name.

# n8n Flow – Test



# n8n Flow – Test

My workflow + Add tag

Inactive Share Save ⚙ ... Star 128,930

Editor Executions

When chat message received → AI Agent Tools Agent

AI Agent Tools Agent

Chat Model Memory Tool

Model Memory

Google Gemini Chat Simple Memory Model

Hide chat Table List Schema Execute Query executeQuery executeQuery executeQuery

Node executed successfully

Chat Session e7c058b5ab9c46128bdd9ed... Latest Logs from AI Agent node

List the columns and their data types in the table view\_oee\_date.

i OK. The table view\_oee\_date has the following columns:

- quality (numeric)
- id\_machine (smallint)
- date (date)
- loading\_time (numeric)

Type a message, or press 'up' arrow for previous

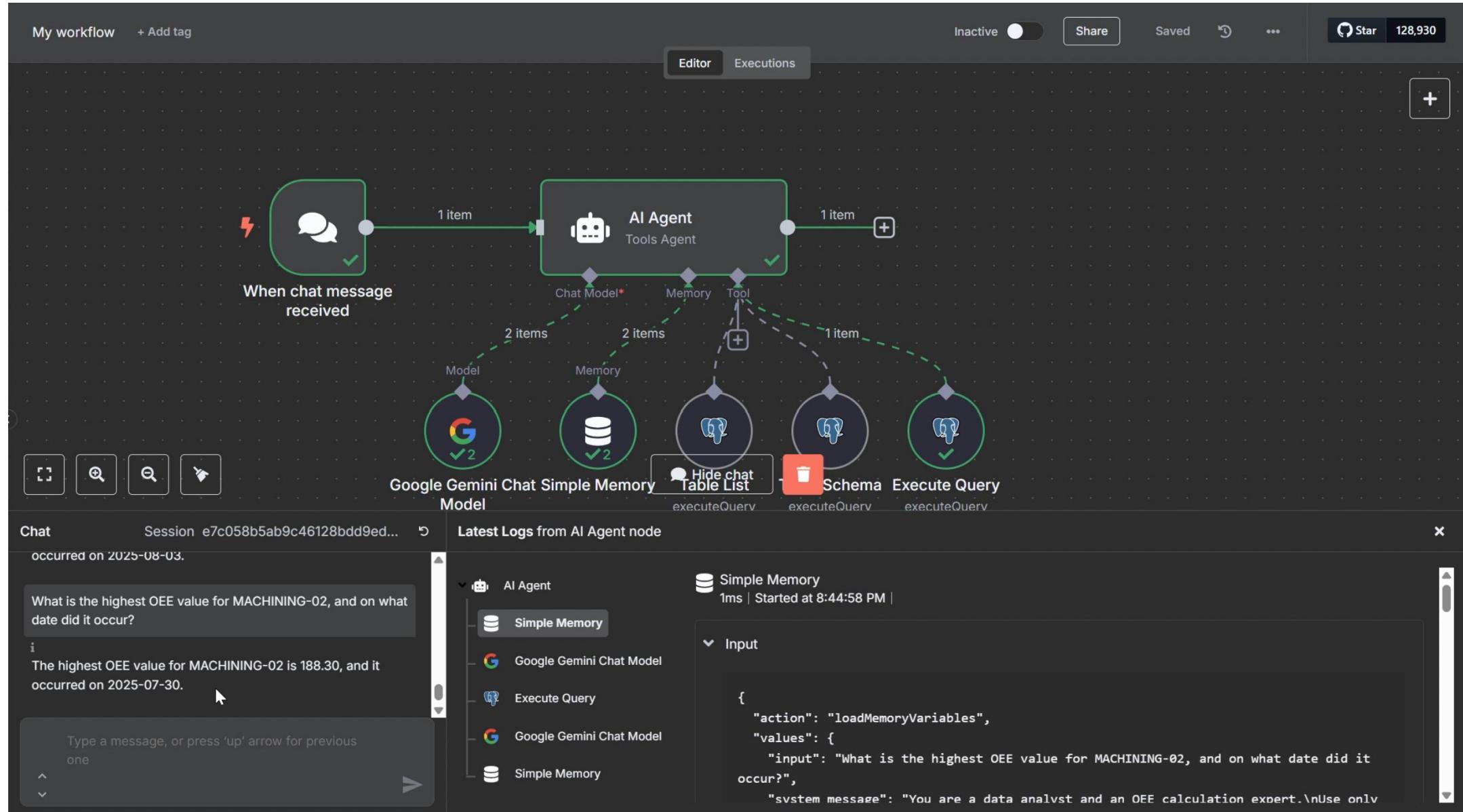
one

Simple Memory 1ms | Started at 8:21:19 PM |

Input

```
{ "action": "loadMemoryVariables", "values": { "input": "List the columns and their data types in the table view_oee_date.", "system_message": "You are a data analyst and an OEE calculation expert.\nUse only available tools. If asked for data. query it.\n\nDon't ask the user for the table" }}
```

# n8n Flow – Test



# Thank You

 +628128717487