Node Guide : Select Rows

Overview

The **Select Rows** node is used to **retrieve data** from tables stored in the **Data Vault**. It supports both **manual SQL** and **visual query-building** modes, making it accessible to both technical and non-technical users.

Use this node to run select queries, test them during configuration, and store the results into agent variables for use in your automation flow.

What This Node Does

- Fetches data using SQL SELECT
- Offers a powerful **query builder UI**
- Validates and tests SQL statements
- Maps results into agent variables
- Enables dynamic data handling in your flows

Inputs

Accepts a **SQL query** in either query mode or UI mode to retrieve data from one or more tables in the Data Vault.

Outputs

- **Fetched Rows**: A variable to store the result set
- **Number of Rows Fetched**: A variable to store the count of rows retrieved
- **Status**: A variable indicating whether the query executed successfully or failed

---

Configuration

The node has two flexible modes for building and executing SQL queries:

---

1. Query Mode

- View a **list of tables** and their **schema** (column names and types)
- See available **agent variables** to insert into your query using `{{variable_name}}`
- Manually enter your **SQL query** in the editor
- Enter a **natural language description** and click **Generate** to auto-create a query using AI
- Click **Validate** to check for SQL syntax errors
- Click **Test** to preview the actual data returned from the query

---

2. UI Mode

- Choose one or more **tables** from a dropdown
- Select the **columns** you want to retrieve
  - If no columns are selected, `*` is used
- Choose an **aggregation type** from the dropdown (optional)
- Provide an **alias** for each selected column (**mandatory**)
- Add SQL clauses visually:
  - **Join**: INNER, LEFT, RIGHT, FULL
  - **Where**: Add filter conditions
  - **Group By** and **Having**
  - **Order By**: ASC or DESC
- Click **Show SQL** to view the auto-generated query

- Use **Test** to verify the query results

Example Flow

Use Case: You want to fetch all orders where the status is "Pending" and assign them to a support agent.

Flow Steps:

1. **Select Rows Node**

Write a query:

```
SELECT * FROM orders WHERE status = 'Pending'
```

- ○
- Store results in `pending_orders_list`
- Store count in `order_count`
- Check status in `query_status`

2. **Condition Node**
    - If `order_count > 0`, proceed
3. **For(Each) Node**
    - Loop through `pending_orders_list`
4. **Update Row Node**
    - Assign each order to a support agent

This flow ensures you dynamically pull pending orders and assign them without writing static data.

Description

The **Select Rows** node helps fetch structured data from your internal tables stored in the Data Vault. It supports both technical SQL users and non-technical users via its easy-to-use UI mode. It helps prevent flow failure by allowing you to **validate** and **test** your SQL during node configuration.

Use it when you need to:

- Retrieve data conditionally
- Drive decisions based on live data
- Feed downstream nodes with real-time values

> Note: This node is for **data retrieval only**. To manipulate data, use **Insert**, **Update**, or **Delete** nodes.