
Bug Report: MindsDB Job Scheduler Bug Report- Data Ingestion Not Triggering Until Docker Restart

Reported By: Rajesh Adhikari / [rajesh-adk-137](#)

Date: June 29, 2025

MindsDB Version: 8.0.17

1. Overview

This report outlines an issue where scheduled jobs in MindsDB, particularly for Knowledge Base data ingestion, fail to reflect updates in the target Knowledge Base immediately after execution. Although job logs indicate successful execution and the source data is accessible to MindsDB, changes are only reflected after a Docker Desktop restart.

2. Prerequisites & Setup

To reproduce this issue, ensure the CharacterKB project is properly set up using the `README.md` and `MINDSDB_SETUP.md` guides available in the repository: [CharacterKB GitHub](#).

Key setup requirements:

- MindsDB is running via Docker Desktop.
 - A database named `character_sheet_10000` is connected to a Google Sheet containing `character_data_10000`.
 - The `character_kb_10000` Knowledge Base is created and initialized.
-

3. Steps to Reproduce

1. Create and Start the Scheduled Job

Execute the following SQL in MindsDB Studio:

```
CREATE JOB character_kb_job AS (  
  INSERT INTO character_kb_10000  
  SELECT unique_id, media_type, genre, character_name, description  
  FROM character_sheet_10000.character_data_10000  
  WHERE unique_id > 10000 -- Adjust based on your dataset  
)  
EVERY 1 minute;
```

2. Verify Job Status

```
SHOW JOBS;  
SELECT * FROM log.jobs_history WHERE name = 'character_kb_job';
```

Expected: Jobs show as successfully executed with no errors.

3. **Add New Data to Source** Manually append new rows in character_data_10000 on Google Sheets with unique_id > 10000 (e.g., 10001, 10002, etc.).

4. **Verify New Data Visibility**

```
SELECT *  
FROM character_sheet_10000.character_data_10000  
ORDER BY unique_id DESC  
LIMIT 5;
```

Expected: Newly added data should appear in query results.

5. **Query the Knowledge Base** Wait over a minute (or a few job cycles), then run:

```
SELECT *  
FROM character_kb_10000  
WHERE content = 'pineapple'  
LIMIT 5;
```

Expected: Newly added data should be returned.

Actual: New data is not returned.

6. **Restart Docker Desktop** Restart Docker Desktop and wait for MindsDB to come back online.

7. **Re-query Knowledge Base** Re-run the semantic search query:

```
SELECT *  
FROM character_kb_10000  
WHERE content = 'pineapple'  
LIMIT 5;
```

Expected: Newly added data is now returned.

4. Expected Behavior

The scheduled job character_kb_job should automatically ingest new data every minute from character_data_10000 into character_kb_10000. Updates should be immediately reflected in the Knowledge Base following each job execution—without requiring a Docker restart.

5. Actual Behavior

- Scheduled jobs run successfully with no errors.

- New data is visible in the source (character_data_10000) but does not appear in the Knowledge Base.
- Restarting Docker causes the data to appear in the Knowledge Base.
- Manual INSERT statements immediately update the Knowledge Base without needing a restart.

6. Diagnostic Steps & Observations

- **Manual Insertion Test:** Running INSERT INTO character_kb_10000 ... manually results in immediate update to the Knowledge Base.
- **Source Data Visibility:** Queries confirm real-time visibility of newly added rows.
- **Job Logs:** log.jobs_history shows successful runs with no errors.
- **Docker Dependency:** A container restart is required for scheduled inserts to be reflected in the Knowledge Base.

7. Supporting Evidence

Images provided below follow this sequence:

1. Job creation

```
56
57 CREATE JOB character_kb_job AS (
58   ... INSERT INTO character_kb_10000
59   ... SELECT unique_id, media_type, genre, character_name, description
60   ... FROM character_sheet_10000.character_data_10000
61   ... WHERE unique_id > 10000;
62 )
63 EVERY 1 min;
64
```

Last Query: CREATE JOB character_kb_job AS (INSERT INTO character_kb_10000 SELECT unique_id, media_type,... Export ▾

Message	
1	Query successfully completed

2. Addition of test data

10000	9999	Blogs	Detective and M	Miss Fisher	Phryne Fisher is a glamorous and adventurous priv
10001	10000	Movies	Psychological Tr	Clarice Starling	Clarice Starling is a young FBI trainee who is deter
10002	10001	pineapple	pineapple	pineapple	pineapple
10003					

3. Verification of source data visibility

```
72 SELECT *
73 FROM character_sheet_10000.character_data_10000
74 ORDER BY unique_id DESC
75 LIMIT 5;
76
```

Last Query: `SELECT * FROM character_sheet_10000.character_data_10000 ORDER BY unique_id DESC LIMIT 5;` Export ▾

	unique_id	media_type	genre	character_name	description
1	10001	pineapple	pineapple	pineapple	pineapple
2	10000	Movies	Psychological Thrillers	Clarice Starling	Clarice Starling "Buffalo Bill"

4. Failure of Knowledge Base to reflect updates

```
65
66 SELECT *
67 FROM character_kb_10000
68 WHERE content = 'pineapple'
69 LIMIT 5
70
```

Last Query: `SELECT * FROM character_kb_10000 WHERE content = 'pineapple' LIMIT 5` Export ▾

	id	chunk_id	chunk_content
1	6539	6539:description:1of1:0to335	Tony Tony Chopper is a doctor and a reindeer who and is known for his adorable appearance and incro do anything to help his crew.

5. Docker restart

Container CPU usage 0.48% / 1600% (16 CPUs available) Container memory usage 1.72GB / 7.27GB Show charts

Search [] [] Only show running containers

	Name	Image	Status	Port(s)	CPU (%)	Memory (%)	Actions
	backend		Running (1/1)		0.6%	51%	<div>View details Pause Restart Open in VSCode</div>

6. Knowledge Base reflecting updates post-restart

```
65 SELECT *
66 FROM character_kb_10000
67 WHERE content = 'pineapple'
68 LIMIT 5
69
70
```

Last Query: `SELECT * FROM character_kb_10000 WHERE content = 'pineapple' LIMIT 5` Export ▾

	id	chunk_id	chunk_content
1	10001	10001:description:1of1:0to9	pineapple

Also we have verified Manual INSERT who have shown immediate effect without restart.

```
48
49 --- manual insertion works instantly
50
51 INSERT INTO character_kb_10000
52 SELECT unique_id, media_type, genre, character_name, description
53 FROM character_sheet_10000.character_data_10000
54 WHERE unique_id > 10000;
55
56
```

8. Potential Cause (Hypothesis)

It is hypothesized that the job scheduler interacts with a cached or non-persistent version of the data, leading to a delay in applying updates to the Knowledge Base. However, this is inconsistent with the immediate effect seen in manual INSERT operations, suggesting that the issue may stem from the job scheduling mechanism specifically, possibly due to a sync or flushing problem within the scheduler internals.
