Cloudera Runtime 7.3.2

Troubleshooting Apache Atlas

Date published: 2020-07-28
Date modified: 2026-03-31

https://docs.cloudera.com/

Legal Notice

Contents

Atlas index repair configuration

You can use reindexing to troubleshoot Apache Atlas basic search inconsistency.

Rebuilding the whole Atlas index

In your Cloudera Manager instance running the Atlas service, add the following in Atlas Server Advanced Configuration Snippet (Safety Valve) for conf/atlas-application.properties.

atlas.rebuild.index=true

atlas.patch.numWorkers=3

atlas.patch.batchSize=300

Later, restart the Atlas Service.

Attention:

- You must revert back this configuration once the reindexing is completed, else the reindexing takes place on every restart.
- The reindexing process will be done during Atlas restart, so Atlas will not be reachable till reindexing process is completed.
- The time taken for reindexing depends upon the amount of data.

Rebuilding the index for particular GUID

Incorrect search results related to a particular GUID can be repaired by limiting the reindex to that element.

atlas-index-repair/repair_index.py [-g <***GUID***>]

Note:

Atlas will use REST APIs to fetch the entity, which will need the correct authentication mechanism to be specified based on the installation.

For an Atlas installation with username and password use the following:

atlas-index-repair/repair_index.py [-g <***GUID***>] [-u <***USER***>] [-p <***PASSWORD***>] * guid: [optional]

Example:

atlas-index-repair/repair_index.py -u admin -p admin123 -g 13d77457-2a45-4e92-ad53-a172c7cb70a5

For Atlas installations using Kerberos as authentication mode, use the following:

kinit -kt /etc/security/keytabs/atlas.service.keytab atlas/fqdn@DOMAIN

Example:

kinit -kt /etc/security/keytabs/atlas.service.keytab atlas/fqdn@EXAMPLE.com

atlas-index-repair/repair_index.py -g 13d77457-2a45-4e92-ad53-a172c7cb70a5

Note: In case of many affected entities, it is recommended to rebuild the whole index instead.

Activating Concurrent Message Ingestion

Atlas hook message processing can take a long time with a high number of messages, greatly increasing wait times. This can improved by turning on Concurrent Message Ingestion.

Atlas Hook message processing has linear complexity for consuming the messages. This can lead to the following:

- Enforcing authorization policies takes longer.
- Metadata showing up in Atlas takes an unpredictable amount of time.

Concurrent Message Ingestion can offset these issues by the enabling the following after determining dependencies within incoming messages:

- Dependent messages are processed serially.
- Messages without dependencies are processed concurrently.

Enabling Concurrent Message Ingestion

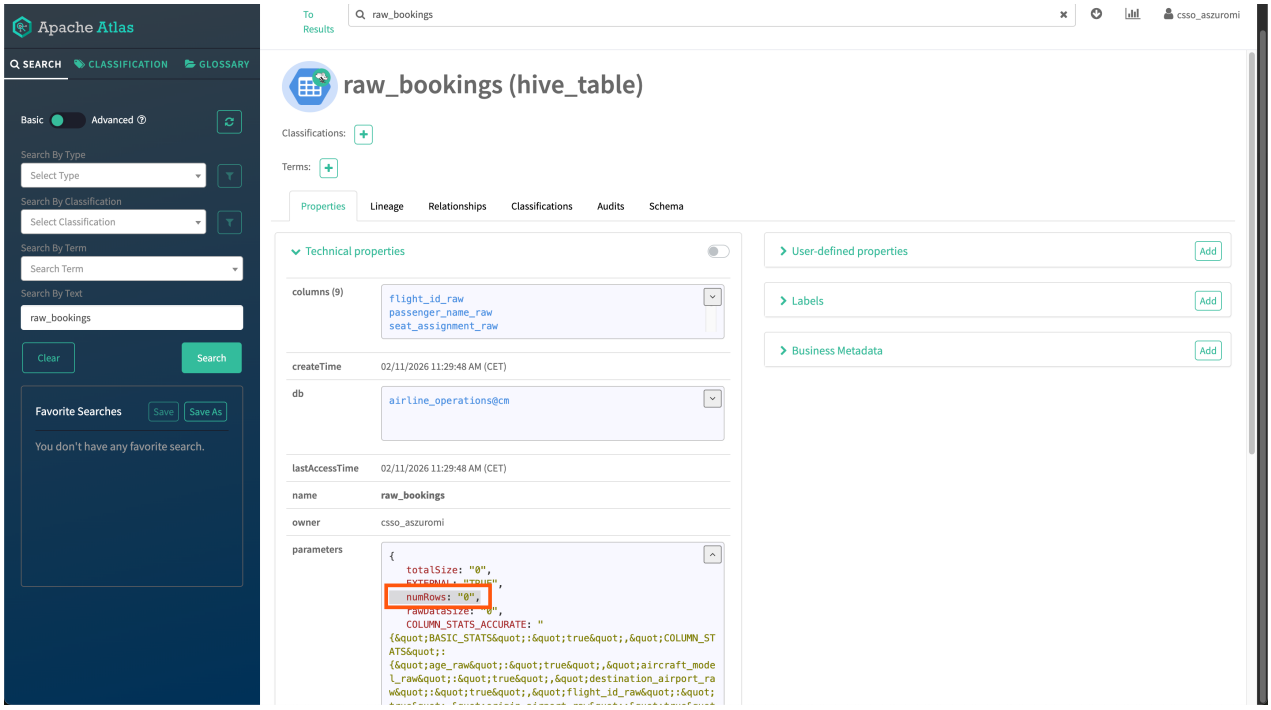
1. Go to Cloudera Manager Clusters Atlas Configuration .
2. Add the following property atlas.notifications.concurrent=true.

Real-time update of entity details

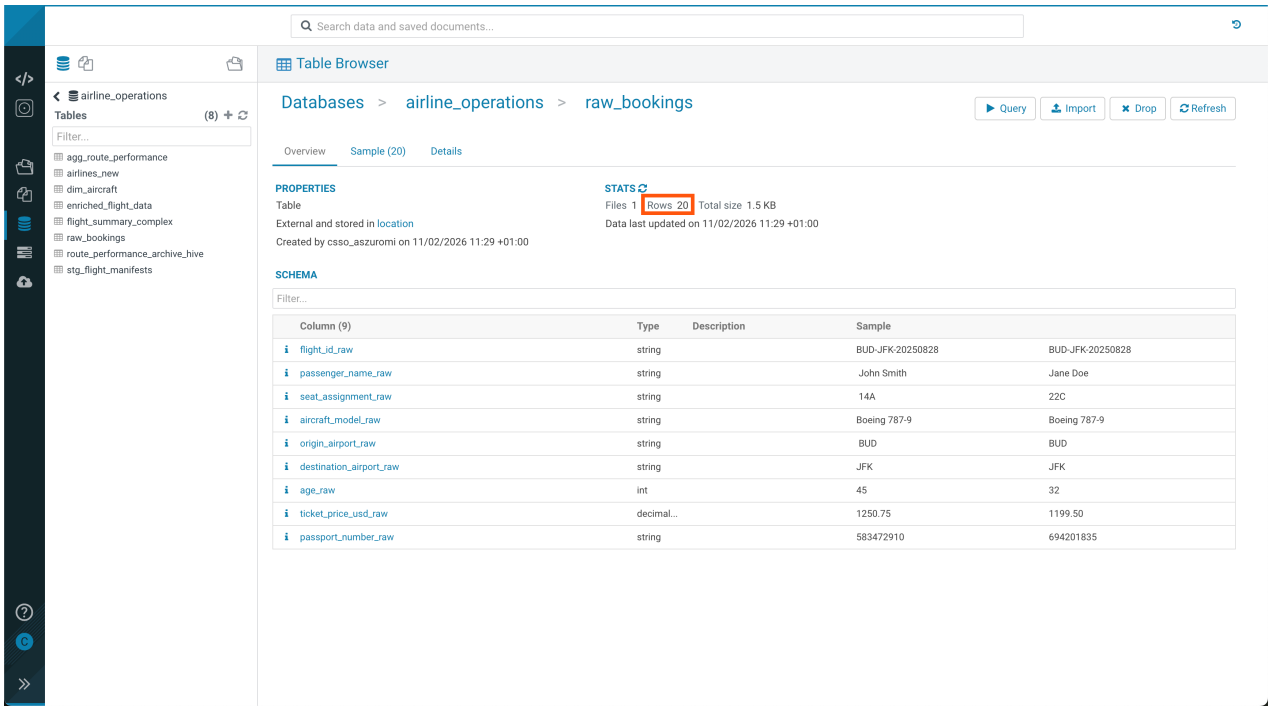
Atlas cannot update row numbers for entities unless a Data Definition Language operation is made on the entity.

Atlas can only monitor events caused by the DDL commands. This means that Atlas is not updated when Data Manipulation Language (DML) operations (SELECT/INSERT/UPDATE/DELETE) cause changes in Hive. Details not updated include also the operational properties, for example, **lastAccessTime**, **last_modified_by**, **last_modified_time**. The displayed details in Atlas (for example, number of rows) can be triggered by Data Definition Language (DDL) operations (CREATE/ALTER/DROP).

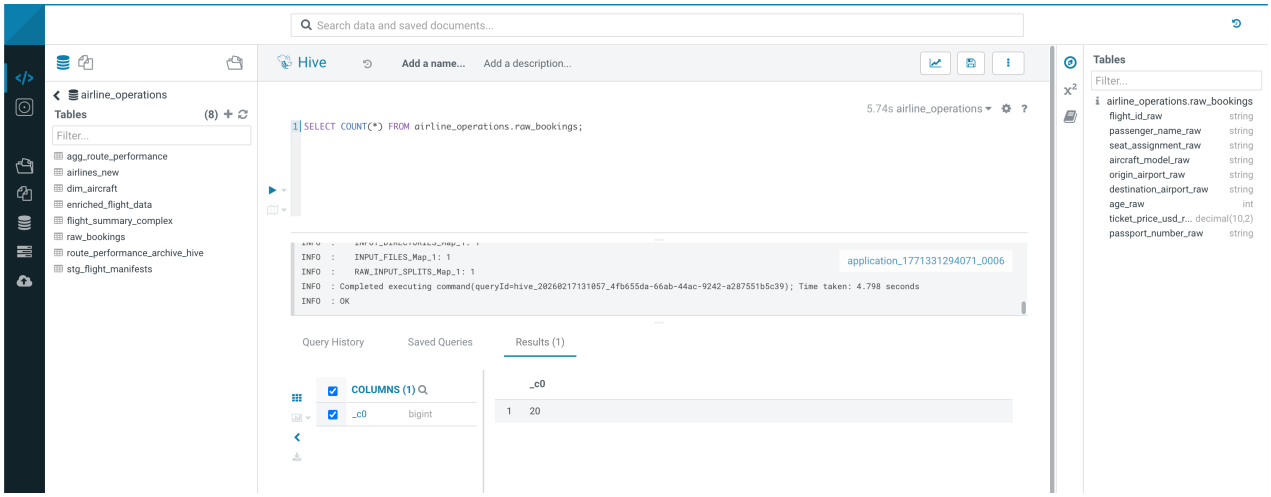
Figure 1: Hive table without any rows in Atlas



You can check the number of rows in Hue with the **Table Browser**.



You can also check the number of rows by using the COUNT query.



Note: Running an ALTER operation on the specific table, for example adding new column, triggers the recalculation of rows in Atlas' **parameters** field.