

Cloudera Runtime 7.1.9

Using Hue

Date published: 2020-07-28

Date modified: 2023-09-07

CLOUDERA

<https://docs.cloudera.com/>

Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

About using Hue.....	4
Accessing and using Hue.....	4
Viewing Hive query details.....	10
Viewing Hive query history.....	10
Viewing Hive query information.....	10
Viewing explain plan for a Hive query.....	11
Viewing Hive query timeline.....	12
Viewing configurations for a Hive query.....	12
Viewing DAG information for a Hive query.....	14
Terminating Hive queries.....	17
Comparing Hive queries in Hue.....	17
Enable stored procedures in Hue.....	18
Run stored procedure from Hue.....	19
Using SQL to query HBase from Hue.....	20
Querying existing HBase tables.....	21
Enabling the SQL editor autocompleter.....	21
Using governance-based data discovery.....	22
Searching metadata tags.....	22
Supported non-ASCII and special characters in Hue.....	23
Options to rerun Oozie workflows in Hue.....	25
Creating Iceberg tables using Hue.....	26

About using Hue

Hue provides a one-stop querying experience in Cloudera Data Platform (CDP) that leverages Hive and Impala SQL queries.

Accessing and using Hue

Get started using Hue by analyzing and visualizing your data with Impala and Hive SQL query engines.

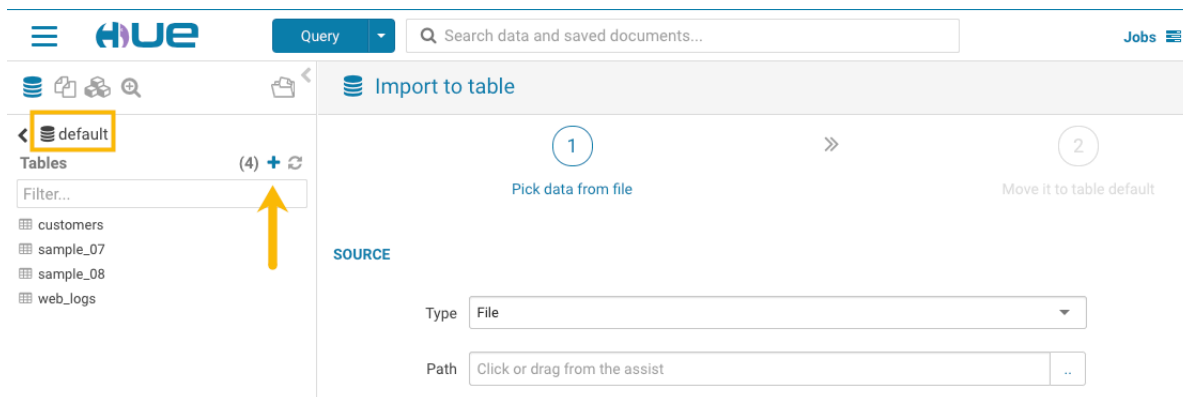
About this task


To try Hue without having an account, try running sample queries on <http://demo.gethue.com/>.

Procedure

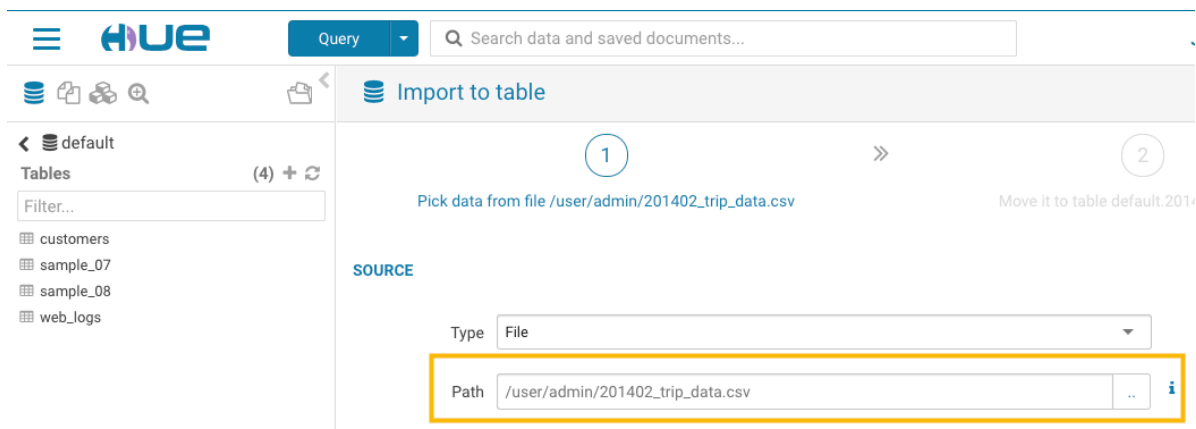
1. Download and unzip [one year of bike trips](#) from the Bay Area Bike Share program. This file is about 80 MB in size.

2. Create a table from the `~/babs_open_data_year_1/201402_babs_open_data/201402_trip_data.csv` file found in the unzipped `babs_open_data_year_1.zip` file:
 - a) In the Cloudera Manager, select Hue WebUI Hue Load Balanced to launch Hue.
 - b) In the left navigation panel of Hue, make sure the default database is selected, and click the plus sign to create a table as shown in the following image:



If the default database is not selected, click the "less than" icon  that is next to the database icon in the left panel. This enables you to select the default database.

- c) In the center panel Importer UI, set Type to File.
- d) Drag the `201402_trip_data.csv` file to the Path field as shown in the following image:



- e) Set the formats as follows:
 - Field Separator = Comma (,)
 - Record Separator = New line
 - Quote Character = Double Quote

Then click Next at the bottom of the page.

- f) Set the properties Format = Text.
- g) Edit the FIELDS as follows:
 - Rename Bike # to Bike ID
 - Change the data type of ZipCode to string.
 - Remove all of the spaces in the Name fields.


Then click Submit at the bottom of the page.













3. Click Query at the top of the page and select Editor Hive to open the Hive editor and then create a query.

- Enter the following query into the editor window:

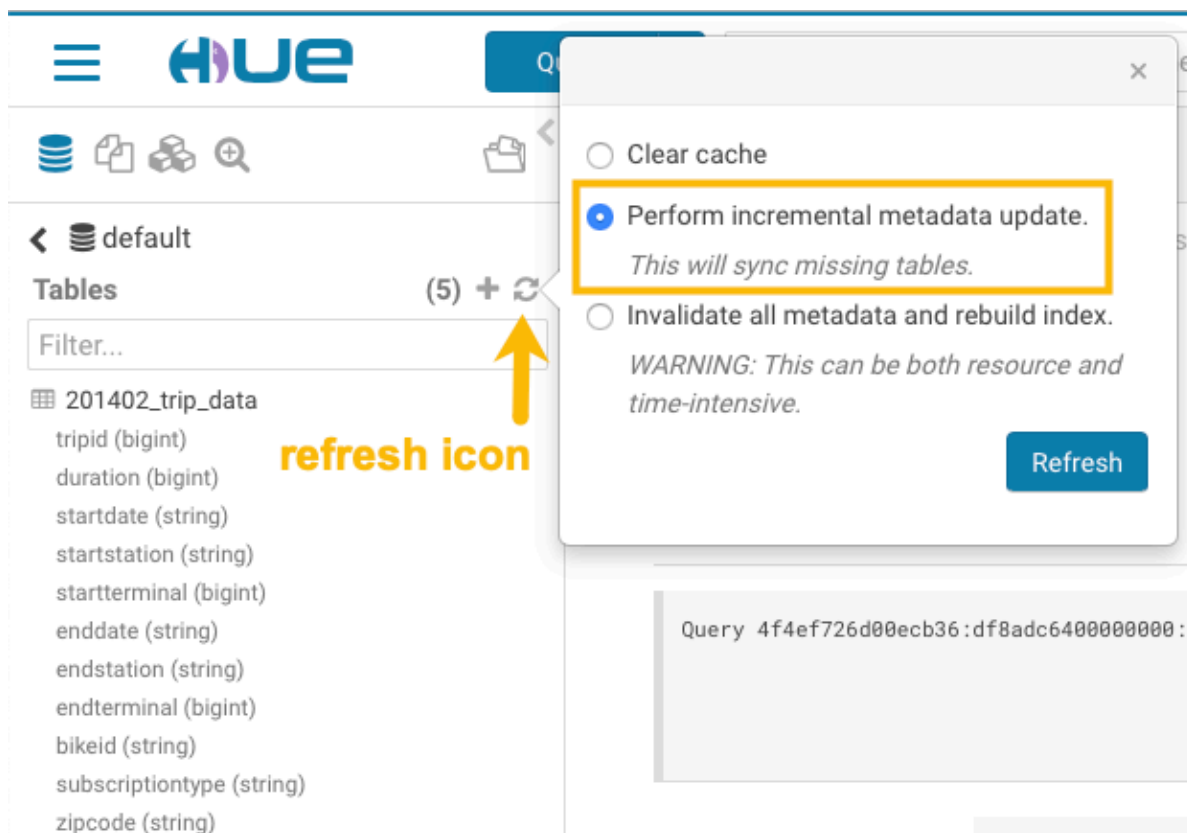
```
SELECT * FROM default.201402_trip_data
LIMIT 10;
```

-

Click the execute icon  to run the query. The following rows are returned:

	Query History	Saved Queries	Results (10)																																																						
			<table><thead><tr><th></th><th>201402_trip_data.tripid</th><th>201402_trip_data.duration</th><th>201402_trip_data</th></tr></thead><tbody><tr><td></td><td>1</td><td>4576</td><td>63</td><td>8/29/2013 14:13</td></tr><tr><td></td><td>2</td><td>4607</td><td>70</td><td>8/29/2013 14:42</td></tr><tr><td></td><td>3</td><td>4130</td><td>71</td><td>8/29/2013 10:16</td></tr><tr><td></td><td>4</td><td>4251</td><td>77</td><td>8/29/2013 11:29</td></tr><tr><td></td><td>5</td><td>4299</td><td>83</td><td>8/29/2013 12:02</td></tr><tr><td></td><td>6</td><td>4927</td><td>103</td><td>8/29/2013 18:54</td></tr><tr><td></td><td>7</td><td>4500</td><td>109</td><td>8/29/2013 13:25</td></tr><tr><td></td><td>8</td><td>4563</td><td>111</td><td>8/29/2013 14:02</td></tr><tr><td></td><td>9</td><td>4760</td><td>113</td><td>8/29/2013 17:01</td></tr><tr><td></td><td>10</td><td>4258</td><td>114</td><td>8/29/2013 11:33</td></tr></tbody></table>		201402_trip_data.tripid	201402_trip_data.duration	201402_trip_data		1	4576	63	8/29/2013 14:13		2	4607	70	8/29/2013 14:42		3	4130	71	8/29/2013 10:16		4	4251	77	8/29/2013 11:29		5	4299	83	8/29/2013 12:02		6	4927	103	8/29/2013 18:54		7	4500	109	8/29/2013 13:25		8	4563	111	8/29/2013 14:02		9	4760	113	8/29/2013 17:01		10	4258	114	8/29/2013 11:33
	201402_trip_data.tripid	201402_trip_data.duration	201402_trip_data																																																						
	1	4576	63	8/29/2013 14:13																																																					
	2	4607	70	8/29/2013 14:42																																																					
	3	4130	71	8/29/2013 10:16																																																					
	4	4251	77	8/29/2013 11:29																																																					
	5	4299	83	8/29/2013 12:02																																																					
	6	4927	103	8/29/2013 18:54																																																					
	7	4500	109	8/29/2013 13:25																																																					
	8	4563	111	8/29/2013 14:02																																																					
	9	4760	113	8/29/2013 17:01																																																					
	10	4258	114	8/29/2013 11:33																																																					

4. Click Query at the top of the page and select Editor Impala to open the Impala SQL editor and then create a query.
 - a. In the left panel, click the refresh icon and select Perform incremental metadata update to make the new table visible to Impala:

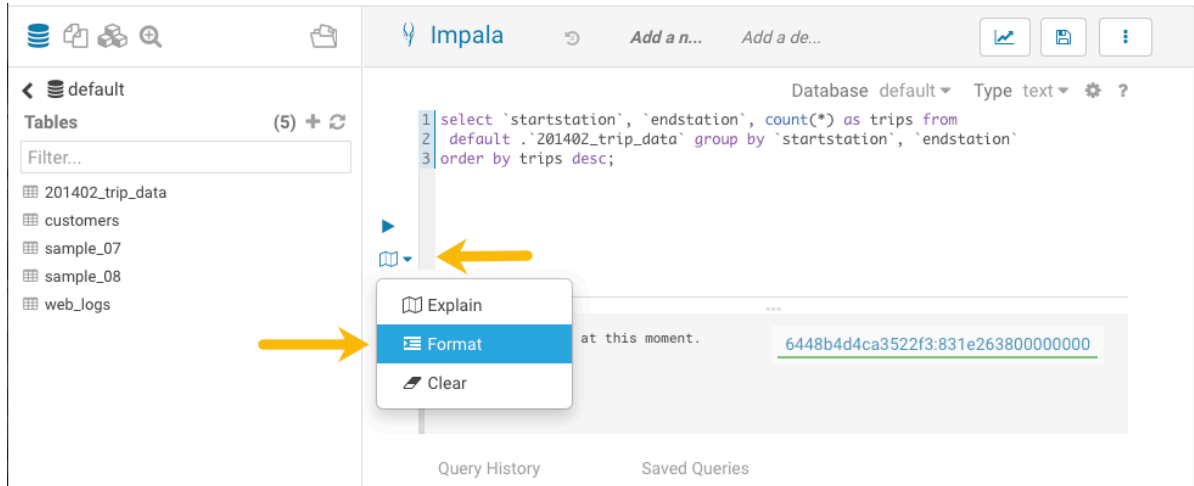


The screenshot shows the Hue interface with a modal dialog box open. The dialog box has three radio button options: "Clear cache", "Perform incremental metadata update.", and "Invalidate all metadata and rebuild index.". The "Perform incremental metadata update." option is selected and highlighted with a yellow box. Below it, there is a warning: "WARNING: This can be both resource and time-intensive.". A "Refresh" button is located at the bottom right of the dialog. In the background, the left panel shows a table named "201402_trip_data" with various columns. A yellow arrow points to the refresh icon (a circular arrow) next to the table name, with the text "refresh icon" written in yellow below it.

- b. Enter the following query into the editor window:

```
select 'startstation', 'endstation', count(*) as trips from default.'201402_trip_data'
group by 'startstation', 'endstation' order by trips desc;
```

- c. Click the down arrow just under the execution icon and select Format:

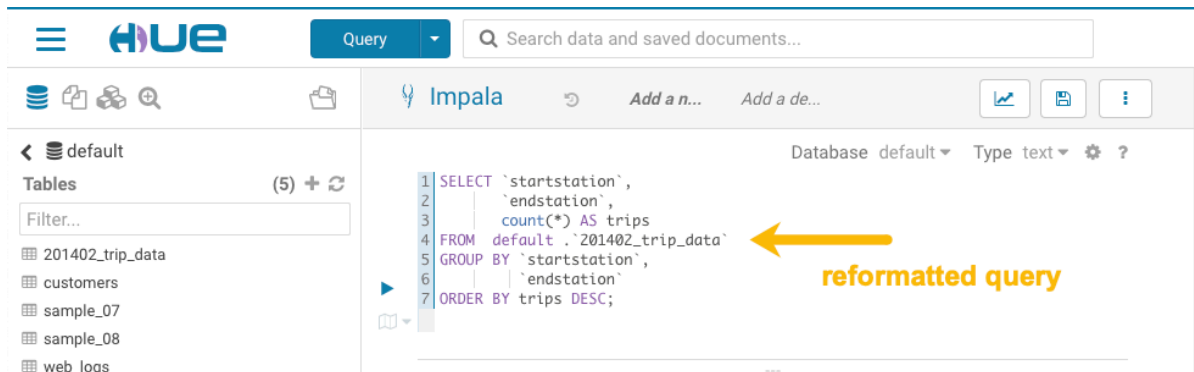


The screenshot shows the Hue Impala interface. On the left, there is a sidebar with a 'Tables' section containing a list of tables: 201402_trip_data, customers, sample_07, sample_08, and web_logs. The main area displays a SQL query in a text editor:

```
1 select `startstation`, `endstation`, count(*) as trips from
2 default.`201402_trip_data` group by `startstation`, `endstation`
3 order by trips desc;
```

Below the query editor, a dropdown menu is open, showing three options: Explain, Format (highlighted in blue), and Clear. A yellow arrow points to the 'Format' option. Another yellow arrow points to the dropdown arrow icon in the query editor. Below the query editor, there is a 'Query History' section with a 'Saved Queries' section containing a single entry with a long alphanumeric ID: 6448b4d4ca3522f3:831e263800000000.

This reformats the query:




The screenshot shows the Hue Impala interface with the same SQL query as in the previous screenshot, but now reformatted. The query is displayed as follows:

```
1 SELECT `startstation`,
2       `endstation`,
3       count(*) AS trips
4 FROM default.`201402_trip_data`
5 GROUP BY `startstation`,
6          `endstation`
7 ORDER BY trips DESC;
```

A yellow arrow points to the reformatted query text, with the label 'reformatted query' written in yellow. The interface also shows the Hue logo and a search bar at the top.

- d.

Click the save icon , enter a query name, and click Save.

- e.

Click the execute icon  to run the query.

5. Create a bar chart that is based on the query results:

a.



Click the chart icon and then select Bars.

Impala interface showing a query and its results. The query is:

```

1 SELECT `startstation`,
2        `endstation`,
3        count(*) AS trips
4 FROM default.`201402_trip_data`
5 GROUP BY `startstation`,
6          `endstation`
7 ORDER BY trips DESC;
    
```

The results table shows the top 10 stations by trip count:

startstation	endstation
1 Harry Bridges Plaza (Ferry Building)	Embarcadero at Sansome
2 Townsend at 7th	San Francisco Caltrain (Townsend at 4th)
3 San Francisco Caltrain 2 (330 Townsend)	Townsend at 7th
4 Market at Sansome	2nd at South Park
5 Embarcadero at Sansome	Steuart at Market
6 2nd at South Park	Market at Sansome
7 San Francisco Caltrain (Townsend at 4th)	Harry Bridges Plaza (Ferry Building)
8 2nd at Townsend	Harry Bridges Plaza (Ferry Building)

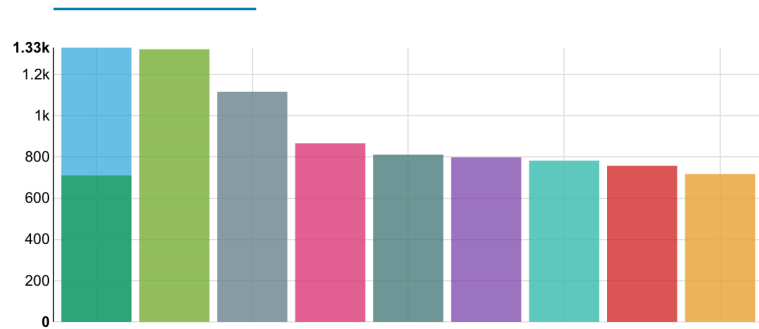
The 'Columns' panel on the left shows 'startstation' selected, and the 'Bars' chart type is highlighted with a yellow box and arrow.

b. Set the bar chart elements as follows:

- X-AXIS = startstation
- Y-AXIS = trips
- LIMIT = 10

Configuration panel for the bar chart:

- TYPE: Bars
- X-AXIS: startstation
- Y-AXIS: trips
- GROUP: Choose a column to pivo...
- LIMIT: 10
- SORTING: [Icons for sorting options]




6.



Create a pie chart by clicking the chart icon again and then select Pie.

7.

Download the query results by clicking  and selecting in what format you want to download, copy, or export the results.

Viewing Hive query details

You can search Hive query history, compare two queries, download debug bundles for troubleshooting, and view query details, a graphical representation of the query execution plan, and DAG information on the Job Browser page in Hue.

Viewing Hive query history

The Queries tab on the Job Browser page in Hue displays all the queries that were run from various query interfaces, such as Beeline, Hive Warehouse Connector (HWC), Tableau, Hue, and other JDBC BI clients and tools.

About this task

Only Query Processor Administrators can view historical queries of all users to monitor resource utilization and control costs from the Hue Job Browser. Non-admin users can view only their queries.

Procedure

1. Log in to the Hue web interface.
2. Click Jobs from the left assist panel.
The **Job Browser** page is displayed.
3. Click Queries.

The Hive queries that were run for the past seven days are displayed. You can select the time period for which you want to view the historical data.

You can also filter queries by their status.

Related Information

[Adding Query Store Administrator users on Private Cloud Base clusters](#)

Viewing Hive query information

The Query Info tab provides information such as, the Hive query ID, the user who executed the query, the start time, the end time, the total time taken to execute the query, the tables that were read and written, application ID, Directed Acyclic Graph (DAG) IDs, session ID, LLAP app ID, thread ID, and the queue against which the query was run.

Procedure

1. Log in to the Hue web interface.
2. Click Jobs from the left assist panel.
The **Job Browser** page is displayed.

- Go to the **Queries** tab and click on the query for which you want to view the query details.

The following image shows the **Query Info** tab on the Hue web interface:

The screenshot displays the Hue web interface with the 'Job Browser' header and 'Jobs' and 'Queries' tabs. The 'Queries' tab is active, showing a table with columns for QUERY ID, USER, and STATUS. The first row shows a query ID, a user, and a 'SUCCESS' status. Below the table, the 'Query Info' tab is selected, displaying the query text on the left and various execution metrics on the right.

QUERY ID	USER	STATUS
hive_20220509083516_a9d00c94-657a-4d80-9cc2-51851ec711eb		✓ SUCCESS

Query Info tabs: Query Info, Visual Explain, Timeline, Query Config, DAG Info, DAG Flow, DAG Swimlane, DAG Counters, DAG Configurations

```

QUERY
SELECT
*
FROM
customer
WHERE
c_nationkey = 15

```

START TIME	3 minutes ago
END TIME	2 minutes ago
DURATION	50s
TABLES READ	customer (default)
TABLES WRITTEN	-
APPLICATION ID	application_1652085158072_0001
DAG ID	dag_1652085158072_0001_2
SESSION ID	76e59bed-40e6-4387-8c35-52606ecacaf4
LLAP APP ID	
THREAD ID	HiveServer2-Background-Pool: Thread-297
QUEUE	None

Viewing explain plan for a Hive query

The Visual Explain feature provides a graphical representation of the query execution plan. The Explain plan is read from right to left. It provides details about every stage of query execution.


Procedure

- Log in to the Hue web interface.
- Click Jobs from the left assist panel.
The **Job Browser** page is displayed.
- Go to the **Queries** tab and click on the query for which you want to view the query details.

4. Click on Visual Explain.

The following image shows the **Visual Explain** tab on the Hue web interface:



5. (Optional) Click  to download the query explain plan in JSON format.

Viewing Hive query timeline

The Timeline tab provides a visual representation of Hive performance logs and shows the time taken by each stage of the query execution.

About this task

Following are the stages in which a query is executed:

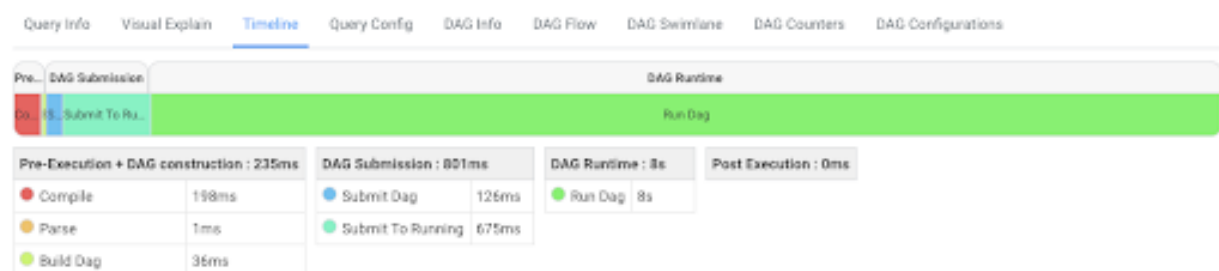
- Pre-execution and DAG construction: It is the first phase of query execution and is executed on the Hive engine. It constitutes the time taken to compile, parse, and build the Directed Acyclic Graph (DAG) for the next phase of the query execution.
- DAG submission: It is the second phase in which the DAG that was generated in Hive is submitted to the Tez engine for execution.
- DAG runtime: It shows the time taken by the Tez engine to execute the DAG.
- Post-execution: It is the last phase of query execution in which the files in S3/ABFS are moved or renamed.

Duration data about each phase are distilled into more granular metrics based on query execution logs.

Procedure

1. Log in to the Hue web interface.
2. Click Jobs from the left assist panel.
The **Job Browser** page is displayed.
3. Go to the **Queries** tab and click on the query for which you want to view the query details.
4. Click on Timeline.

The following image shows the **Timeline** tab on the Hue web interface:



Viewing configurations for a Hive query

The Query Config tab provides the configuration properties and settings that are used in a Hive query. You can use this tab to verify that configuration property values align with your expectations.

Procedure

1. Log in to the Hue web interface.
2. Click Jobs from the left assist panel.
The **Job Browser** page is displayed.
3. Go to the **Queries** tab and click on the query for which you want to view the query details.
4. Click on Query Config.

The following image shows the **Query Config** tab on the Hue web interface:

Config Name	Config Value
hadoop.security.group.mapping.ldap.posix.attr...	uidNumber
dfs.block.invalidate.limit	1000
yarn.admin.acl	*
hive.repl.dump.metadata.only.for.external.table	true
hive.exec.stagingdir	.hive-staging
hive.druid.rollup	true
yarn.federation.enabled	false
yarn.app.mapreduce.am.job.committer.cancel...	60000
hive.druid.broker.address.default	localhost:8082
dfs.disk.balancer.max.disk.throughputInMBper...	10
dfs.qjournal.select-input-streams.timeout.ms	20000
hive.llap.io.orc.time.counters	true
hive.repl.retain.prev.dump.dir	false
hive.vectorized.execution.mapjoin.native.fast...	true
dfs.provided.aliasmap.inmemory.leveldb.dir	/tmp
yarn.nodemanager.process-kill-wait.ms	5000
yarn.minicluster.use-rpc	false
io.map.index.interval	128

Viewing DAG information for a Hive query

Directed Acyclic Graph (DAG) is created by the Hive engine every time you query the Hive Virtual Warehouse. The Hive SQL queries are compiled and converted into a Tez execution graph also known as a DAG. DAG is a collection of vertices where each vertex executes a fragment of the query or script. Hue provides a web interface to view detailed information about DAGs.

About this task

Directed connections between vertices determine the order in which they are executed. For example, the vertex to read a table must be run before a filter can be applied to the rows of that table. As another example, consider a vertex that reads a user table that is very large and distributed across multiple computers and multiple racks. Reading the table is achieved by running many tasks in parallel.



Important: The DAG information tabs (**DAG Info**, **DAG Flow**, **DAG Swimlane**, **DAG Counters**, **DAG Configurations**) are displayed only if the Tez engine is used for query execution. The Tez engine is typically utilized for complex queries.

Procedure

1. Log in to the Hue web interface.
2. Click Jobs from the left assist panel.
The **Job Browser** page is displayed.
3. Go to the **Queries** tab and click on the query for which you want to view the query details.
4. Click DAG Info to see the DAG ID, DAG name, the status of the query, the time taken to execute the DAG, start time, and end time.

The following image shows the **DAG Info** tab on the Hue web interface:

The screenshot shows the Hue web interface with the DAG Info tab selected. The query details are as follows:

QUERY ID	USER	STATUS
hive_20220509083516_a9d00c94-657a-4d80-9cc2-51851ec711eb	[REDACTED]	✓ SUCCESS

Navigation tabs: Query Info, Visual Explain, Timeline, Query Config, **DAG Info**, DAG Flow, DAG Swimlane, DAG Counters, DAG Configurations

DAG ID	DAG NAME
dag_1652085158072_0001_2	SELECT * FROM customer WHERE c_nationke...15 (Stage-1)

STATUS	DURATION
SUCCEEDED	00:00:50

START TIME	END TIME
3 minutes ago	2 minutes ago

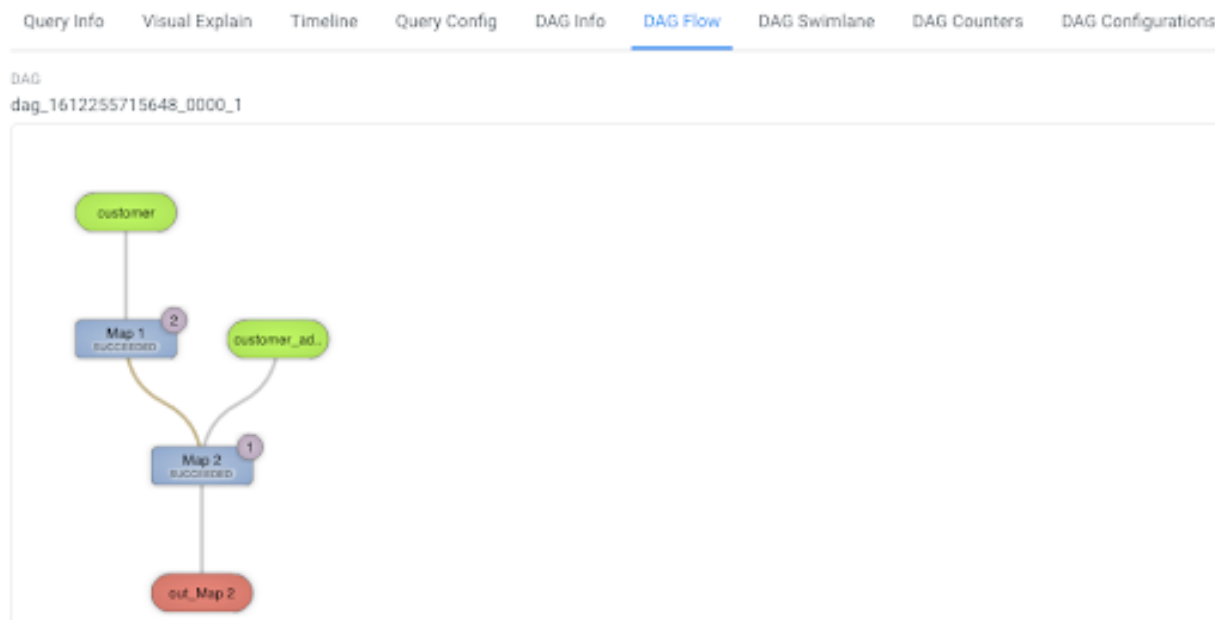
The following table lists and describes the status of the Tez job:

Status	Description
Submitted	The DAG is submitted to Tez but is not running
Running	The DAG is currently running
Succeeded	The DAG was completed successfully
Failed	The DAG failed to complete successfully
Killed	The DAG was stopped manually
Error	An internal error occurred when executing the DAG

5. Click DAG Flow to see the DAG in the form of a flowchart.

You can gain insight into the complexity and the progress of executing jobs, and investigate the vertices that have failures or are taking a long time to complete.

The following image shows the **DAG Flow** tab on the Hue web interface::



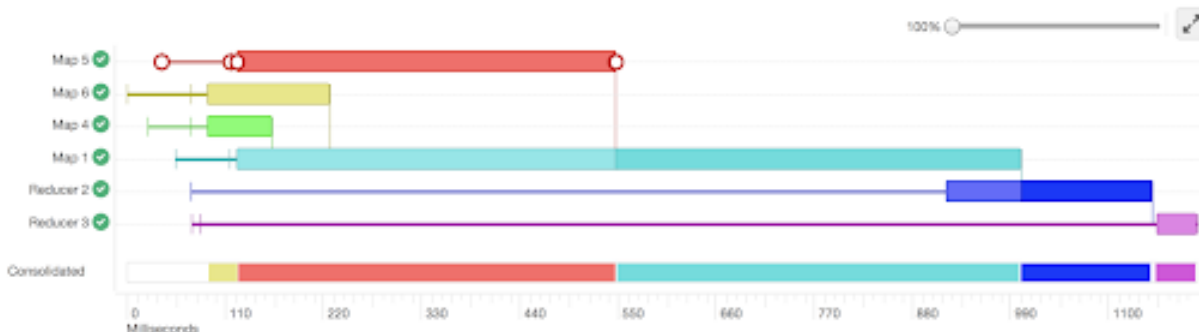
Here, the input to vertices Map 1 and Map 2 are the tables displayed in green boxes. Next, Map 2 depends on the result set generated by Map 1. Map 2 is the last vertex in the DAG flow and after it completes its execution, the query output is written to a file in a filesystem such as S3 or ABFS.

There are a few options to change the layout of the DAG flow. You can hide the input and the output nodes to view only the task vertices by clicking the Toggle source/sink visibility button. You can switch between the horizontal and vertical orientation by clicking the Toggle orientation button.

6. Click DAG Swimlane to see the DAG of the vertices against time.

Each mapping and reducing task is a vertex. Each horizontal bar of the swimlane represents the total time taken by the vertex to complete the execution. The vertical lines indicate the time when the vertex was initialized, the time when the vertex started, the time when the first task started, the time when the last task was completed, and the time when the vertex finished its execution. When you mouse over the vertical line, the bubble displays the stage of the vertex execution and provides a timestamp. The vertical lines connecting two vertices denote the dependency of a vertex on another vertex.

The following image shows the **DAG Swimlane** tab on the Hue web interface:



In this example, Map 1 depends on the results of Map 5. Map 1 will finish its execution only when Map 5 finishes its execution successfully. Similarly, Reducer 2 depends on Map 1 to complete its execution.

The consolidated timeline shows the percentage of time each vertex took to complete executing.

- Click DAG Counters to see details such as the number of bytes read and written, number of tasks that initiated and ran successfully, amount of CPU and memory consumed, and so on.

The **DAG Counters** tab provides a way to measure the progress or the number of operations that occur within a generated DAG. Counters are used to gather statistics for quality control purposes or problem diagnosis.

The following image shows the **DAG Counters** tab on the Hue web interface:

Group Name	Counter Name	DAG : dag_1652085158072_0001_2
org.apache.tez.common.counters.DAGCounter	NUM_SUCCEEDED_TASKS	54
org.apache.tez.common.counters.DAGCounter	TOTAL_LAUNCHED_TASKS	54
org.apache.tez.common.counters.DAGCounter	DATA_LOCAL_TASKS	54
org.apache.tez.common.counters.DAGCounter	AM_CPU_MILLISECONDS	5890
org.apache.tez.common.counters.DAGCounter	AM_GC_TIME_MILLIS	22
org.apache.tez.common.counters.FileSystem...	FILE_BYTES_WRITTEN	1074416910
org.apache.tez.common.counters.FileSystem...	S3A_BYTES_READ	6742239795
org.apache.tez.common.counters.FileSystem...	S3A_READ_OPS	1124
org.apache.tez.common.counters.TaskCounter	TASK_DURATION_MILLIS	759357
org.apache.tez.common.counters.TaskCounter	INPUT_RECORDS_PROCESSED	146519
org.apache.tez.common.counters.TaskCounter	INPUT_SPLIT_LENGTH_BYTES	12387638515
HIVE	CREATED_FILES	36
HIVE	RECORDS_IN_Map_1	150000000
HIVE	RECORDS_OUT_0	6003115
HIVE	RECORDS_OUT_OPERATOR_FIL_5	6003115
HIVE	RECORDS_OUT_OPERATOR_FS_7	6003115
HIVE	RECORDS_OUT_OPERATOR_SEL_6	6003115
HIVE	RECORDS_OUT_OPERATOR_TS_0	150000000

8. Click DAG Configurations to see the Tez configuration details for a query that has a DAG associated with it. The following image shows the **DAG Configurations** tab on the Hue web interface:

Query Info	Visual Explain	Timeline	Query Config	DAG Info	DAG Flow	DAG Swimlane	DAG Counters	DAG Configurations
Config Name		DAG : dag_1612255715648_0000_1						
dfs.namenode.fs-limits.max-xattrs-per-inode		32						
dfs.namenode.delegation.token.always-use		false						
yam.nodemanager.runtime.linux.docker.delaye..		false						
yam.timeline-service.handler-thread-count		10						
yam.timeline-service.webapp.rest-csrf.custom..		X-XSRF-Header						
fs.s3a.retry.limit		7						
dfs.client.write.byte-array-manager.count-reset..		10000						
yam.nodemanager.linux-container-executor.cg..		/hadoop-yam						
mapreduce.shuffle.connection-keep-alive.time..		5						
mapreduce.client.libjars.wildcard		true						
hive.zookeeper.kerberos.enabled		false						

Terminating Hive queries

If a query is running for longer than expected, or you have accidentally triggered it, then you can stop the query to free up the resources. Hue also allows you to stop multiple queries at once.

About this task



Note: This feature is available only for Hive queries. Only admin users or Hue superusers can stop running queries.

Procedure

1. Log in to the Hue web interface.
2. Click Jobs from the left assist panel.
The **Job Browser** page is displayed.
3. Go to the **Queries** tab.
A list of queries that were run is displayed.
4. Select the queries that you want to stop and click Kill.

Comparing Hive queries in Hue

You can compare two queries to know how each query is performing in terms of speed and cost-effectiveness. Hue compares various aspects of the two queries, based on which you can identify what changed between the executions of those two queries, and you can debug performance-related issues between different runs of the same query.

About this task

The query comparison report provides you a detailed side-by-side comparison of your queries.

For Hive queries, it includes recommendations for optimizing each query, metadata about the queries, visual explain for each query, query timeline, query configuration, Directed Acyclic Graph (DAG) information, DAG flows, DAG swimlanes, DAG counters, and DAG configurations.

For Impala queries, the query comparison report includes query details, execution plan details, and the aggregated metrics for both the queries and provides a variance between the two.

Procedure

1. Log in to the Hue web interface.
2. Click Jobs from the left assist panel.
The **Job Browser** page is displayed.
3. Go to the **Queries** tab.
A list of queries that were run is displayed.
4. Select the two queries you want to compare and click Compare.

Query comparison report for Hive queries:

The screenshot shows the Hue web interface for query comparison. At the top, there are two query cards with their IDs and users. Below them is a navigation menu with options: Query Info, Visual Explain, Timeline, Query Config, DAG Info, DAG Flow, DAG Swimlane, DAG Counters, and DAG Configurations. The main content area is split into two columns, each showing details for a query. The left column shows the SQL query: `SELECT * FROM customer WHERE c_nattokeny = 15`. The right column shows the query details: START TIME (2 hours ago), END TIME (2 hours ago), DURATION (148ms), TABLES READ (customer (default)), TABLES WRITTEN (-), APPLICATION ID (application_1652085158072_0001), DAG ID (dag_1652085158072_0001_2), SESSION ID (76e59bed-40e6-4387-8c35-52606ecacaf4), and THREAD ID (HiveServer2-Background-Pool: Thread-245).

Enabling stored procedures for Hive on CDP Private Cloud Base

To create, edit, and drop procedures and functions that are written in Hive Hybrid Procedural SQL (HPL/SQL) using the Hue query editor, you must enable the `hpsql` option in the Hue Advanced Configuration Snippet.

About this task



Note: Hue enables you to switch between Hive and HPL/SQL interpreters. By default, the regular Hive interpreter is enabled when you add the Hue service to your cluster. To enable the HPL/SQL interpreter, you must update Hue's Advanced Configuration Snippet in Cloudera Manager. However, updating Hue's Advanced Configuration Snippet overrides the default configuration. Therefore, to use both Hive and HPL/SQL interpreters, you must enable both by updating Hue's Advanced Configuration Snippet.

Procedure

1. Log in to Cloudera Manager as an administrator.
2. Go to [Clusters Hue Configuration](#) and add the following lines in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

```
[notebook]
  [[interpreters]]
    [[[hive]]]
      name=Hive
      interface=hiveserver2
    [[[hplsql]]]
      name=Hplsql
      interface=hiveserver2
```

3. Click Save Changes.
4. Restart the Hue service.
5. Go to [Clusters Hive on Tez Configuration](#) and add the following property name and its value in the Hive Service Advanced Configuration Snippet (Safety Valve) for hive-site.xml field:

Name: hive.security.authorization.sqlstd.confwhitelist.append

Value: QUERY_EXECUTOR|HPLSQL
6. Click Save Changes.
7. Restart the Hive on Tez service.

How to run a stored procedure from Hue in CDP Private Cloud Base

HPL/SQL allows you to implement business logic using variables, expressions, flow-of-control statements, and iterations. HPL/SQL makes SQL-on-Hadoop more dynamic. You can leverage your existing procedural SQL skills, and use functions and statements to make your typical ETL development more productive. Hue provides a smart interface to run stored procedures.



Note: This feature is available only for Hive queries.

To run stored procedures from Hue, enable the HPL/SQL interpreter by configuring Hue's Advanced Configuration Snippet in Cloudera Manager as described in [Enabling stored procedures for Hive on CDP Private Cloud Base](#).

The following example creates a procedure and returns records by passing a cursor:

```
print 'Hello world';/
CREATE PROCEDURE greet(name STRING)
BEGIN
  PRINT 'Hello ' || name;
END;/
CREATE PROCEDURE even(cur OUT SYS_REFCURSOR)
BEGIN
  OPEN cur FOR
  SELECT n FROM NUMBERS
  WHERE MOD(n, 2) == 0;
END;/
CREATE PROCEDURE set_message(IN name STRING, OUT result STRING)
BEGIN
  SET result = 'Hello, ' || name || '!';
END;
-- Call the procedure and print the results
DECLARE str STRING;
CALL set_message('world', str);
PRINT str;
```



Attention: In the hplsql mode, you must terminate the commands using the forward slash (/). The semicolon (;) is used throughout procedure declarations and can no longer be relied upon to terminate a query in the editor.



Note: HPL/SQL does not support all types of Hive statements, such as JOIN or EXPLAIN. Refer to the [HPL/SQL Reference](#) for more information.

Related Information

[Enabling stored procedures for Hive on CDP Private Cloud Base](#)

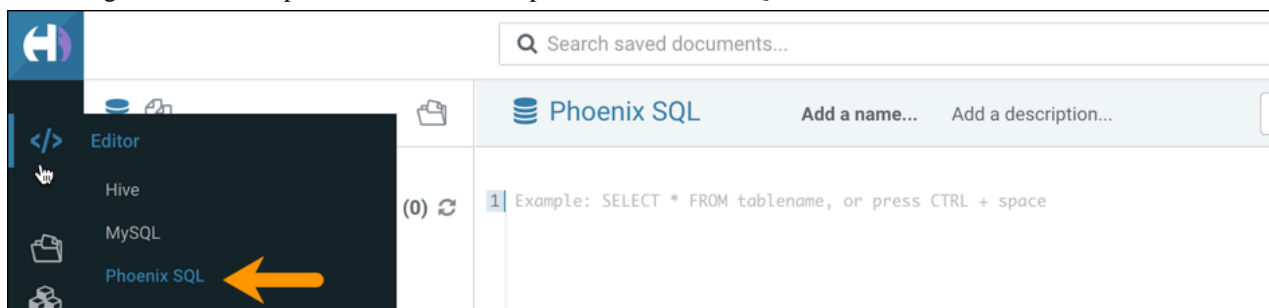
Using SQL to query HBase from Hue

Hue provides a simple SQL interface to create and manipulate SQL tables that are stored in HBase, and define and manipulate views on HBase tables using Apache Phoenix in addition to HBase shell and database API.

Cloudera does not recommend manipulating Phoenix tables from HBase as this can lead to data loss.

The SQL connector is shipped with Hue so that you do not have to download and configure it yourself.

Following are some examples to create and manipulate the Phoenix SQL tables from the Hue editor:



Creating a table

```
CREATE TABLE IF NOT EXISTS Company (company_id INTEGER PRIMARY KEY, name VARCHAR(225));
```

Upserting values in the table

```
UPSERT INTO Company VALUES(1, 'Cloudera');
UPSERT INTO Company VALUES(2, 'Apache');
```

Querying the table

```
SELECT * FROM Company;
```

Deleting a record

```
DELETE FROM Company WHERE COMPANY_ID=1;
```

Dropping the table

```
DROP TABLE Company;
```

Querying existing HBase tables

To use SQL for querying data from existing HBase tables, you must create a view in Phoenix pointing to the HBase table.

To map the existing tables to the views, run the following statement using the Phoenix editor on the Hue web interface:

```
CREATE VIEW if not exists "[***HBASE-TABLE-NAME***]" ( pk VARCHAR PRIMARY KE
Y, val VARCHAR );
```

Enabling the SQL editor autocompleter

Autocompleter provides finely tuned SQL suggestions for Hive and Impala dialects while you enter queries into the editor window. See [Brand new Autocompleter for Hive and Impala](#) in the Hue blog.

About this task

Autocompleter is enabled by default. To manually enable or disable it, open the editor configuration panel and edit settings as follows:

Procedure

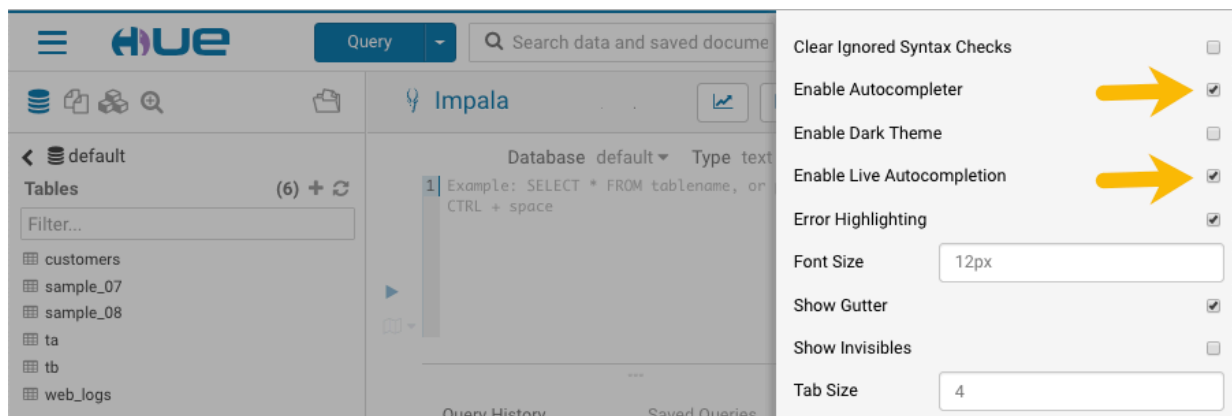
1. Log in to Hue and go to either the Hive or Impala editor.
2. Place your cursor in the editor window and then use one of the following keyboard shortcuts to open the editor configuration panel:

- On a Mac system, use the Command key followed by a hyphen and then a comma:
Command-,
- On a Windows system, use the Ctrl key followed by a hyphen and then a comma:
Ctrl-,



Tip: Type a question mark (?) anywhere but in the active editor window to open a menu of editor keyboard shortcuts.

3. To enable autocompletion, check the box adjacent to Enable Autocompleter. When you check Enable Autocompleter, Enable Live Autocompletion is automatically enabled as well. Place your cursor in the editor window to close the configuration panel.



4. To disable autocompletion:

- Uncheck Enable Live Autocompletion but leave Enable Autocompleter checked, and then place your cursor in the editor window to close the configuration panel. This disables live autocompletion, but if you want to use

autocompletion while building your queries in the editor, enter the following key stroke sequence to activate autocompletion: Ctrl + Space Key

- Uncheck both Enable AutoCompleter and Enable Live Autocompletion, and then click in the editor to close the configuration panel. This disables all autocompletion functionality.

Using governance-based data discovery

Hue can use the metadata tagging, indexing, and search features available in Apache Atlas data management. After integrating Hue with Atlas, classifications and indexed entities can be accessed and viewed in Hue. This topic shows you how to use metadata classifications in Hue.

Integration between Hue and Atlas is enabled by default, but if your administrator has disabled it, it must be re-enabled before you can use governance-based data discovery.

You can create tags to classify your data both from Atlas and Hue.

Searching metadata tags

The SQL Editor in Hue provides a search text box where you can search on the metadata tags or classifications that are associated with your databases, tables, and columns.

About this task

You can search for tags or classifications in either the Hive or the Impala editors.

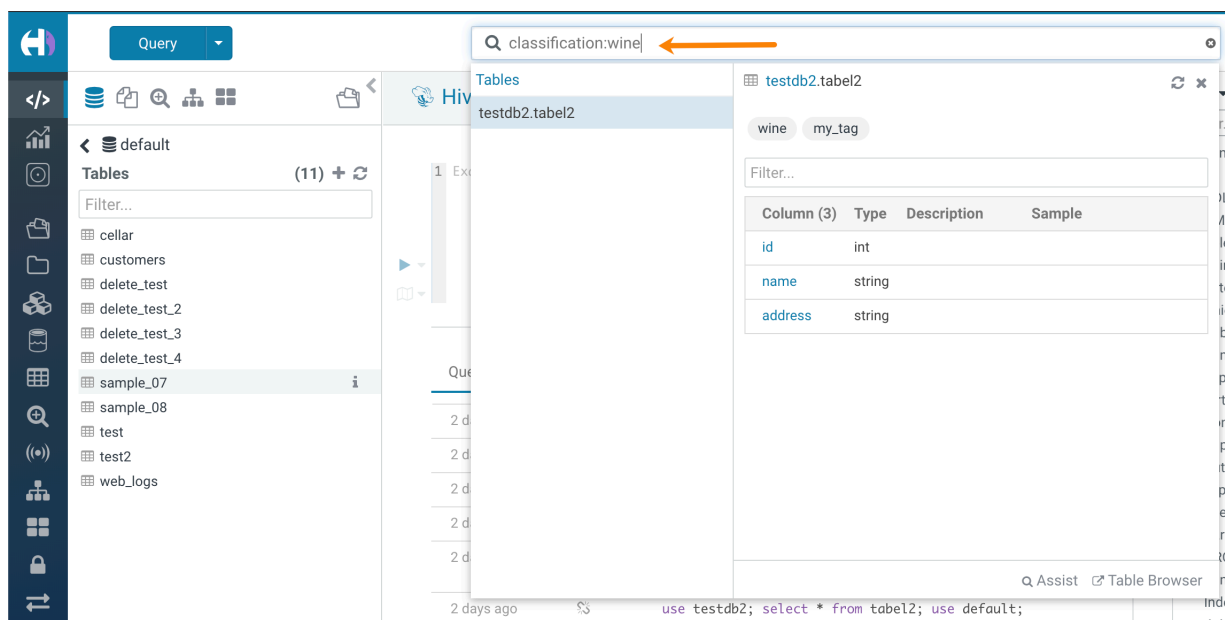


Note: On clusters that use Apache Ranger for role-based access control, the Search mechanism does not display counts of popular values. Ranger ensures that Hue users can view only entities to which their user role (as configured and managed by Ranger) has been granted specific permissions.

Procedure

1. Go to Query Editor Impala or Hive.

- To locate the tags or classifications in Apache Atlas, in the metadata search box located just to the right of the Query drop-down menu, type a tag: or classification: facet followed by its name. For example, type classification:wine as shown in the following image:



After you type the search facet and the tag or classification name in the search box, the `<database>.<table>` where the tag or classification is found is returned. Click the `<database>.<table>` to view the tags and classifications that have been defined for it.

Supported non-ASCII and special characters in Hue

Auto-generated files may often introduce non-alphanumeric characters in the file and directory names that Hue does not support. This might cause the files or directories to not appear on the Hue File Browser. Review the list of non-alphanumeric, non-ASCII, diacritics (accents) characters supported in Hue for the following operations: upload, create, list in folder, view, and rename.

Table 1: Supported characters on HDFS

Special character symbol	Description	Filename support	Folder name support
~	Tilde	Fully supported	Fully supported
@	Ampersat	Fully supported	Fully supported
#	Hash	Partially supported. Not supported for rename operations.	Unsupported
\$	Dollar sign	Fully supported	Fully supported
&	Ampersand	Fully supported	Fully supported
(Left paranthesis	Fully supported	Fully supported
)	Right paranthesis	Fully supported	Fully supported
*	Asterisk	Fully supported	Fully supported
!	Exclamation mark	Fully supported	Fully supported
+	Plus	Fully supported	Fully supported
=	Equal	Fully supported	Fully supported
:	Colon	Unsupported	Unsupported

Special character symbol	Description	Filename support	Folder name support
;	Semicolon	Fully supported	Fully supported
,	Comma	Fully supported	Fully supported
.	Period	Fully supported	Fully supported
?	Question mark Not supported with Knox.	Fully supported	Fully supported
/	Forward slash	Unsupported	Unsupported
\	Backslash	Unsupported	Unsupported
'	Apostrophe or single quote	Fully supported	Fully supported
T#####-ääö	Non-standard alphabets with diacritics and accents.	Fully supported	Fully supported

Table 2: Supported characters on S3

Special character symbol	Description	Filename support	Folder name support
~	Tilde	Fully supported	Fully supported
@	Ampersat	Fully supported	Fully supported
#	Hash	Partially supported. Not supported for view and rename operations.	Unsupported
\$	Dollar sign	Fully supported	Fully supported
&	Ampersand	Fully supported	Fully supported
(Left paranthesis	Fully supported	Fully supported
)	Right paranthesis	Fully supported	Fully supported
*	Asterisk	Fully supported	Fully supported
!	Exclamation mark	Fully supported	Fully supported
+	Plus	Fully supported. Not supported for create operations on RAZ-enabled environments.	Fully supported
=	Equal	Fully supported	Fully supported
:	Colon	Unsupported	Unsupported
;	Semicolon	Fully supported	Fully supported
,	Comma	Fully supported	Fully supported
.	Period	Fully supported	Fully supported
?	Question mark Not supported with Knox.	Fully supported	Partially supported. Not supported for list, upload, and view operations.
/	Forward slash	Unsupported	Unsupported
\	Backslash	Partially supported. Not supported for upload operations.	Partially supported. Not supported for upload operations.
'	Apostrophe or single quote	Fully supported	Fully supported
T#####-ääö	Non-standard alphabets with diacritics and accents.	Fully supported	Fully supported

Table 3: Supported characters on ABFS

Special character symbol	Description	Filename support	Folder name support
~	Tilde	Fully supported	Fully supported
@	Ampersat	Fully supported	Fully supported
#	Hash	Partially supported. Not supported for view and rename operations.	Unsupported
\$	Dollar sign	Fully supported	Fully supported
&	Ampersand	Fully supported	Fully supported
(Left paranthesis	Fully supported	Fully supported
)	Right paranthesis	Fully supported	Fully supported
*	Asterisk	Fully supported	Fully supported
!	Exclamation mark	Fully supported	Fully supported
+	Plus	Fully supported	Fully supported
=	Equal	Fully supported	Fully supported
:	Colon	Unsupported	Unsupported
;	Semicolon	Fully supported	Fully supported
,	Comma	Fully supported	Fully supported
.	Period	Fully supported	Fully supported
?	Question mark Not supported with Knox.	Partially supported. Not supported for view and rename operations.	Partially supported. Not supported for list, rename, and view operations.
/	Forward slash	Unsupported	Unsupported
\	Backslash	Unsupported	Unsupported
'	Apostrophe or single quote	Fully supported	Fully supported
T#####-ääö	Non-standard alphabets with diacritics and accents.	Fully supported	Fully supported

Options to rerun Oozie workflows in Hue

Oozie workflows consume time and resources to run. You can optimize the rerun of a failed Oozie workflow by selecting one of the following two options: “All or skip successful” or “Only failed”. These options enable you to select individual actions within your workflow that you want to rerun.



Important: The option to rerun the an Oozie workflow is enabled in the following conditions:

- In case of an external workflow, the rerun button is enabled if the workflow was run by the same user who is logged in, or if the user is a Hue superuser.
- If the workflow is created in Hue, then the rerun button is enabled only for the user who originally ran the workflow, or for the Hue superuser.

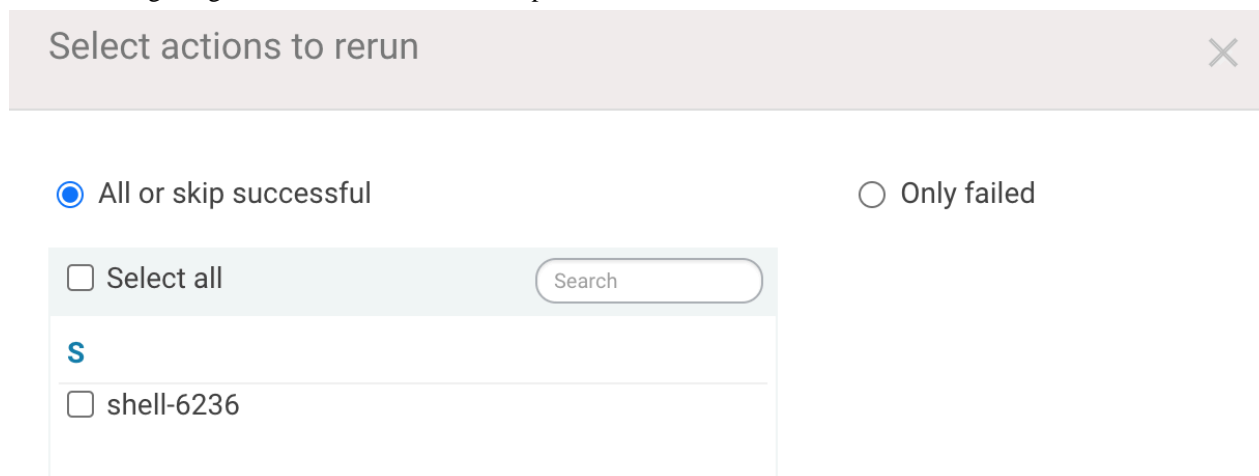
The All or skip successful option enables you to either rerun both the successful and failed actions or skip the actions that ran successfully and run only the failed ones. Using this option, you have more control over selecting the actions that you want to rerun from the list of successfully run actions.

If you select the All or skip successful option but do not select any or all of the successfully run actions, then Hue reruns the whole Oozie workflow.

If you select the All or skip successful option and select some of the successfully run actions, then Hue reruns the selected actions and the failed actions.

Select the Only failed option to only rerun the failed actions within your workflow.

The following image shows the available rerun options on the Hue web interface:



Creating Iceberg tables using Hue

You can create Iceberg tables in Hue by using a CREATE TABLE statement or by importing CSV files using the Hue importer.

About this task



Note: In the CDP 7.1.9 release, Iceberg is not supported with Hive. Parquet is the only supported format for writing to Impala tables. For more information about the supported table formats, sample queries, and syntax, see *Create table feature*.

Following is a sample Impala query to create an Iceberg table in the default Parquet format. You can run this query from the Impala editor in Hue:

```
CREATE TABLE ice_t2 (i int, s string, ts timestamp, d date)
STORED BY ICEBERG;
```

Alternatively, you can create an Iceberg table by importing a CSV file in Hue. You can either use an existing file present on your filesystem or upload a new file into Hue.

Procedure

1. Log in to the Hue web interface.
2. Click Importer on the left-assist panel.
3. Select Remote File from the Type drop-down menu.
4. Click .. adjoining to the Path option.
The **Choose a file** modal is displayed.
5. Upload a CSV file or select an available file present on your filesystem.
Format and preview details are displayed.
At this point, you can select the characters that should be used for the field separator, record separator, and quotes.
6. Click Next.
7. Specify the type and name of your table under the DESTINATION section.
8. Expand the Extras section under the PROPERTIES section and select the Iceberg table option.
Verify other entries and make the necessary modifications.

9. Click Submit.

The CREATE TABLE query is submitted with your preferences.

Related Information

[Create table feature \(syntax and sample queries\)](#)