Part 2 - Create a Virtual Warehouse and Run Queries [45 minutes]

Overview: What is Cloudera Data Warehouse?

We will explore features of Cloudera Data Warehouse (CDW) by performing some data exploration and create dashboards to share our results to a wider audience
We will be taking a look at a generated data set from a mock airline company containing flights information from its fleet of aircraft.

A virtual warehouse represents virtual compute resources to access data that is stored in a database catalog. This lets you create or destroy compute resources, auto-scale, or separate resources across different workloads, all running on the same underlying data.

CDW let's you choose from a set of default resources based on your predicted workload as well as give you fine grained control over autoscaling and timeout features so you can fine tune your system to be most cost effective.

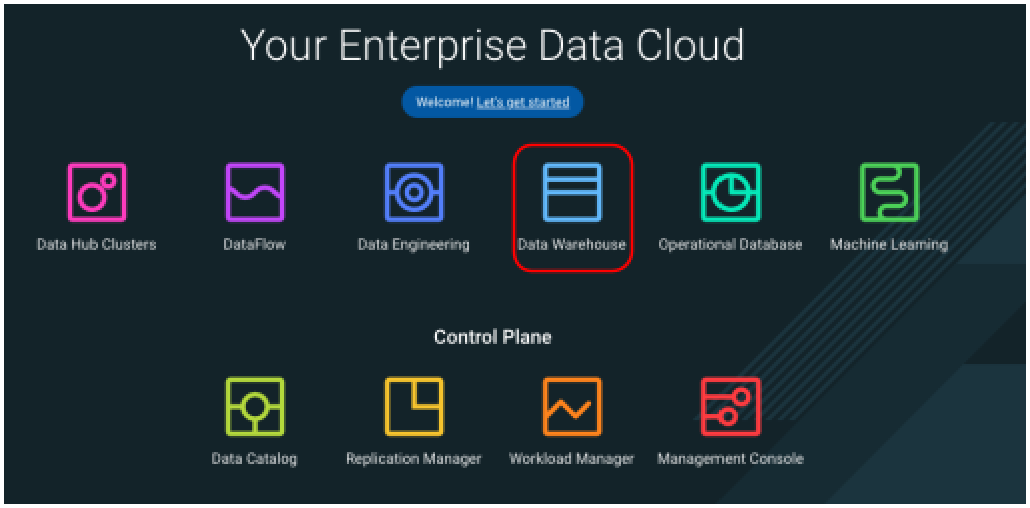
Purpose: Create a virtual warehouse and run queries, answering the questions below:

- What are the top 5 visited destinations by year from (1995-2008)?
- What are the top 10 routes (origin and dest) that have seen maximum diversions?
- Which three months have seen the most number of cancellation due to bad weather?

1) Open CDP, using the “admin” user within the Test Drive link.

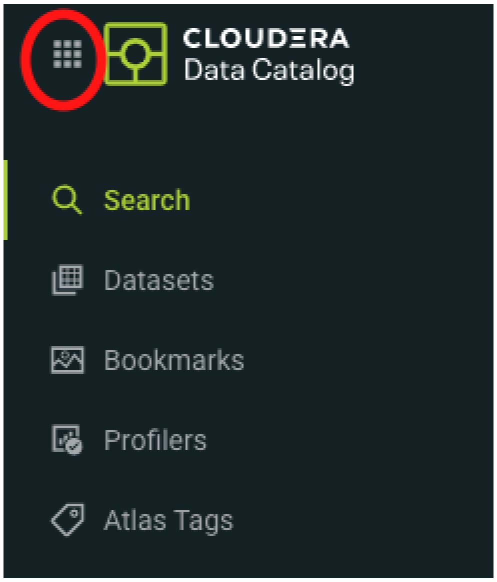
Your link should look something like (remember click the link in your email not the link below)
http://login.trycdp.com/auth/realms/trycdp-trialxx/protocol/saml/clients/samlclient?tn=trialxx_admin@trycdp.com&p=X
*xx represents the trial user #
*X represents the password

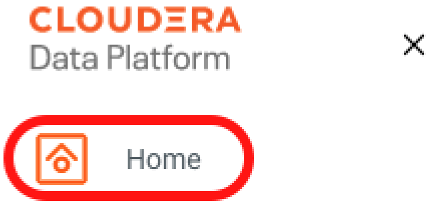
2) Click the “Data Warehouse” within the CDP Home Screen



How do you get to the CDP Home Screen?

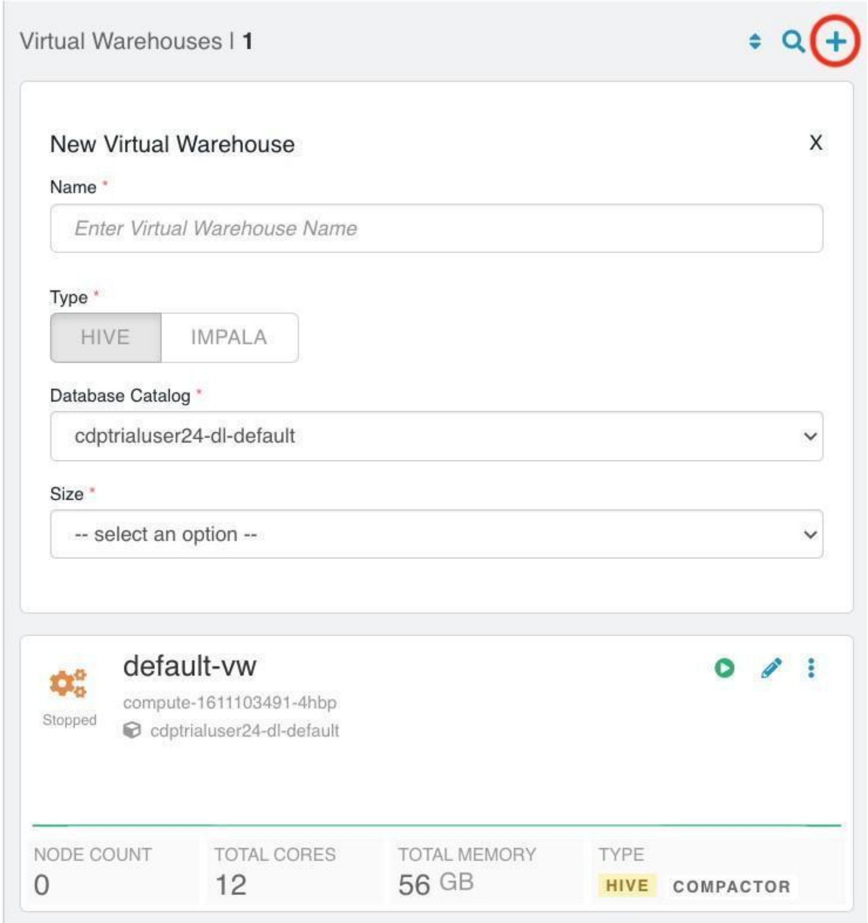
- From any experience such as “Data Catalog”, click the 9 square at the top left and then click “Home”




CLOUDERA
Data Platform
×
Home

WE HAVE DONE THIS FOR YOU – DO NOT CREATE A NEW VIRUTAL WAREHOUSE – READ THROUGH THIS FOR BACKGROUND INFO ONLY…

3) DO NOT Click the “+” at the top right next to “Virtual Warehouses”



Virtual Warehouses | **1**

New Virtual Warehouse X

Name *

Enter Virtual Warehouse Name

Type *

HIVE IMPALA

Database Catalog *

cdptrialuser24-dl-default

Size *

-- select an option --

default-vw

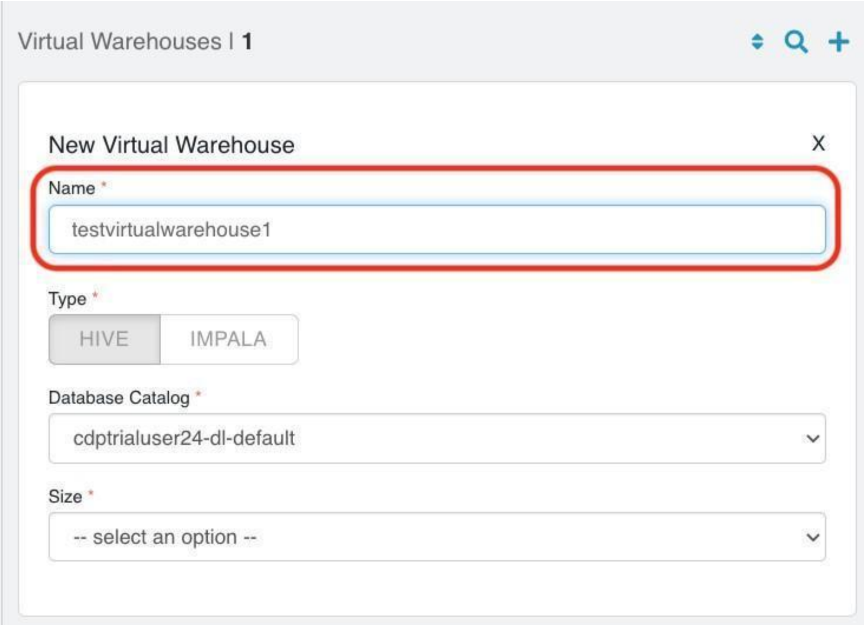
Stopped

compute-1611103491-4hbp

cdptrialuser24-dl-default

NODE COUNT	TOTAL CORES	TOTAL MEMORY	TYPE
0	12	56 GB	HIVE COMPACTOR

4) DO NOT Enter a name for your New Virtual Warehouse



Virtual Warehouses | 1

New Virtual Warehouse

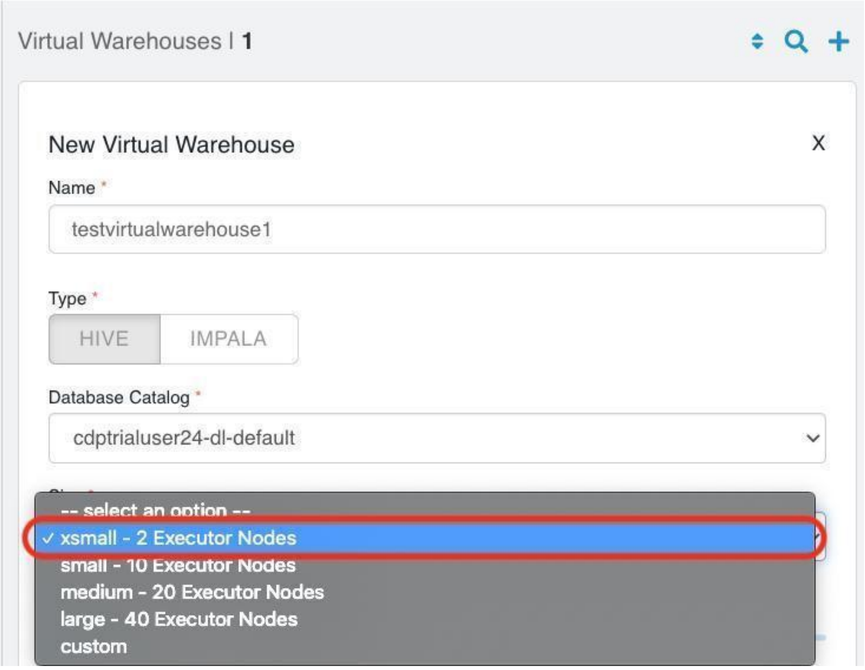
Name *
testvirtualwarehouse1

Type *
HIVE | IMPALA

Database Catalog *
cdptrialuser24-dl-default

Size *
-- select an option --

5)) DO NOT Select the Size of “xsmall - 2 Executor Nodes”
*How do I choose a size? Initial concurrent users



Virtual Warehouses | 1

New Virtual Warehouse

Name *
testvirtualwarehouse1

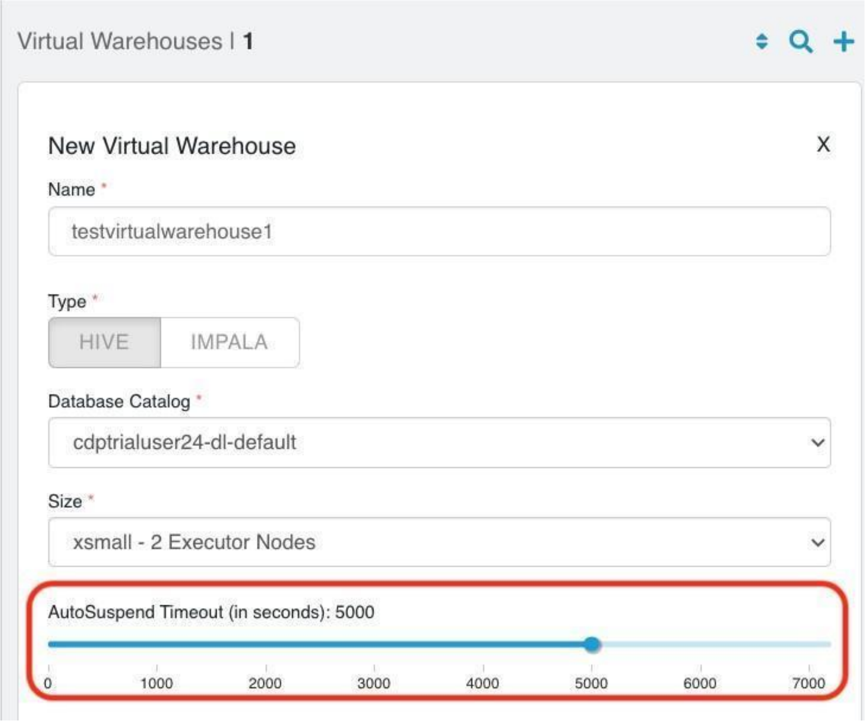
Type *
HIVE | IMPALA

Database Catalog *
cdptrialuser24-dl-default

-- select an option --
✓ xsmall - 2 Executor Nodes
small - 10 Executor Nodes
medium - 20 Executor Nodes
large - 40 Executor Nodes
custom

6) To save money you stop the instances you aren't using. Cloudera lets you define if you spin down to zero, if you have some Kubernetes pods running all the time, and how long these live when there is no workload. DO NOT Set the AutoSuspend Timeout (in seconds) between 4500 and 5500:

*What is AutoSuspend Timeout? Automatically spin-down unused resources after timeout occurs.



Virtual Warehouses | 1

New Virtual Warehouse

Name *: testvirtualwarehouse1

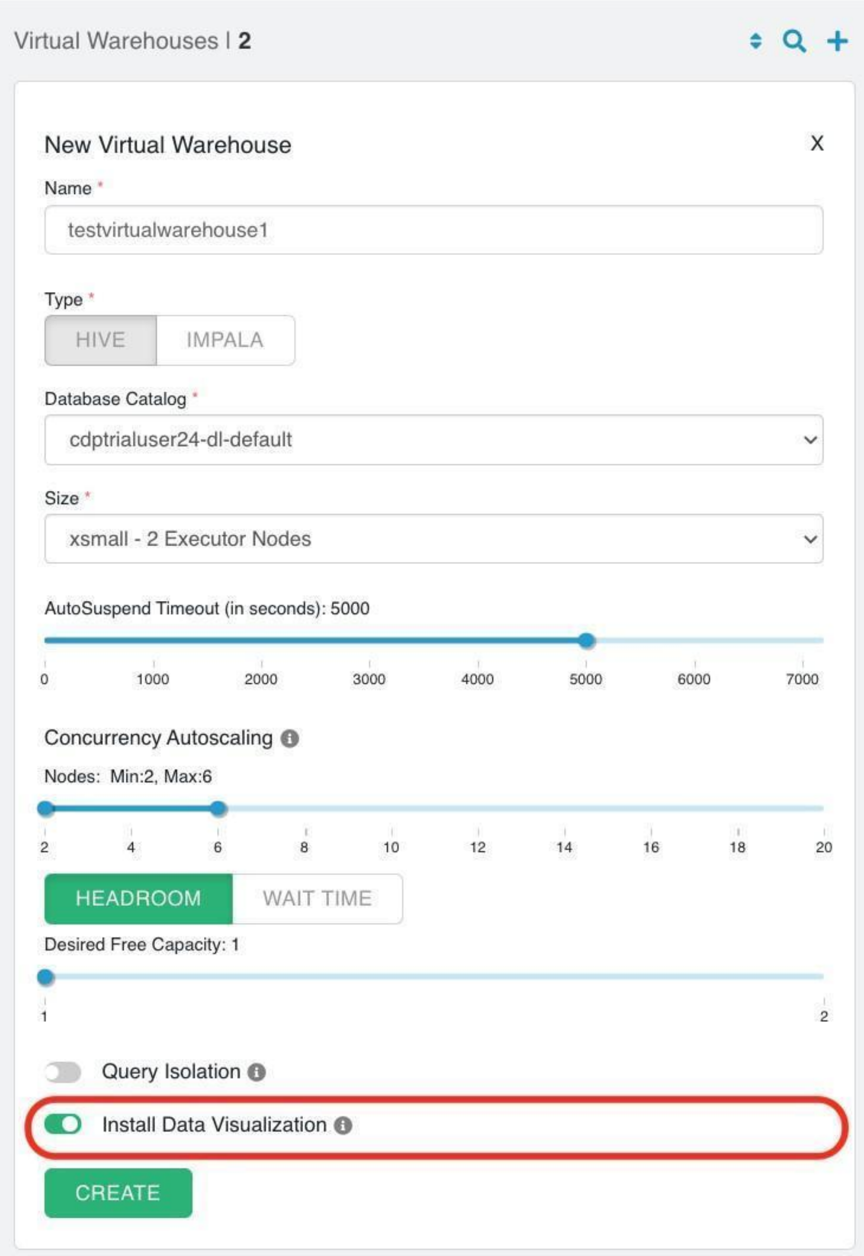
Type *: HIVE | IMPALA

Database Catalog *: cdptrialuser24-dl-default

Size *: xsmall - 2 Executor Nodes

AutoSuspend Timeout (in seconds): 5000

7) DO NOT Choose “Install Data Visualization” to be on *Allowing for Data Visualizations in Part 3



Virtual Warehouses | 2

New Virtual Warehouse

Name *

testvirtualwarehouse1

Type *

HIVE IMPALA

Database Catalog *

cdptrialuser24-dl-default

Size *

xsmall - 2 Executor Nodes

AutoSuspend Timeout (in seconds): 5000

Concurrency Autoscaling

Nodes: Min:2, Max:6

HEADROOM WAIT TIME

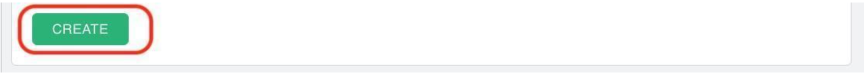
Desired Free Capacity: 1

○ Query Isolation

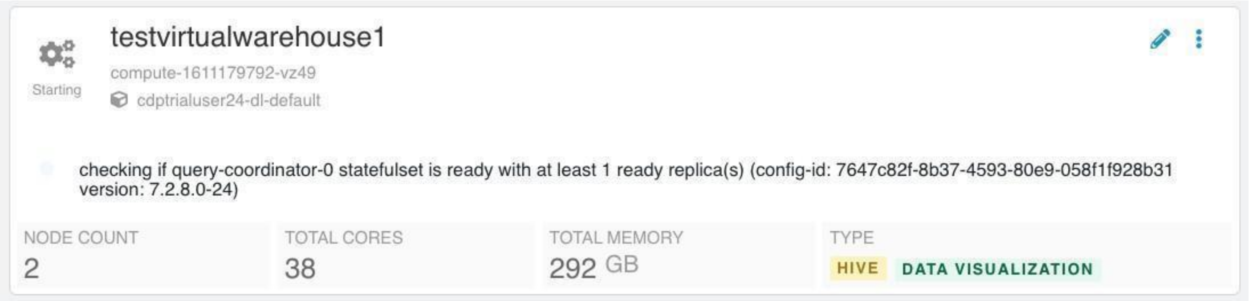
● Install Data Visualization

8) DO NOT , DO NOT , DO NOT, REALLY DO NOT: Click “Create” to create your Virtual Warehouse

*Allow for approximately 5 minutes for your Virtual Warehouse to become available for use

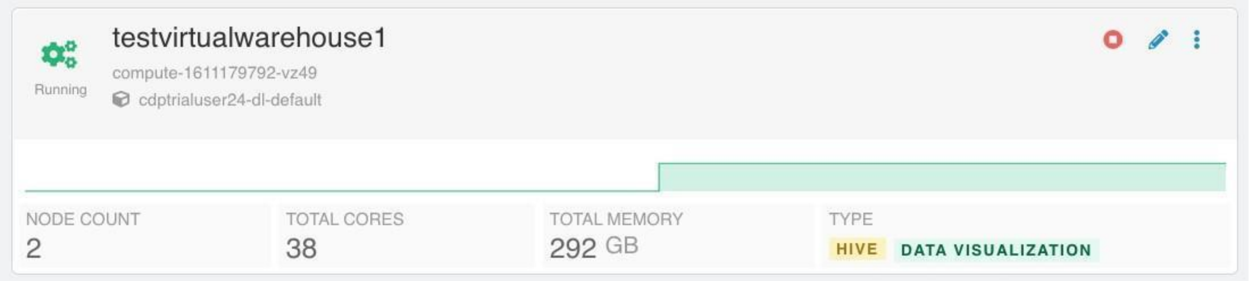


When available for use, “Starting” will change to “Running” as shown below


testvirtualwarehouse1
compute-1611179792-vz49
cdptrialuser24-dl-default
Starting

checking if query-coordinator-0 statefulset is ready with at least 1 ready replica(s) (config-id: 7647c82f-8b37-4593-80e9-058f1f928b31 version: 7.2.8.0-24)

NODE COUNT	TOTAL CORES	TOTAL MEMORY	TYPE
2	38	292 GB	HIVE DATA VISUALIZATION


testvirtualwarehouse1
compute-1611179792-vz49
cdptrialuser24-dl-default
Running

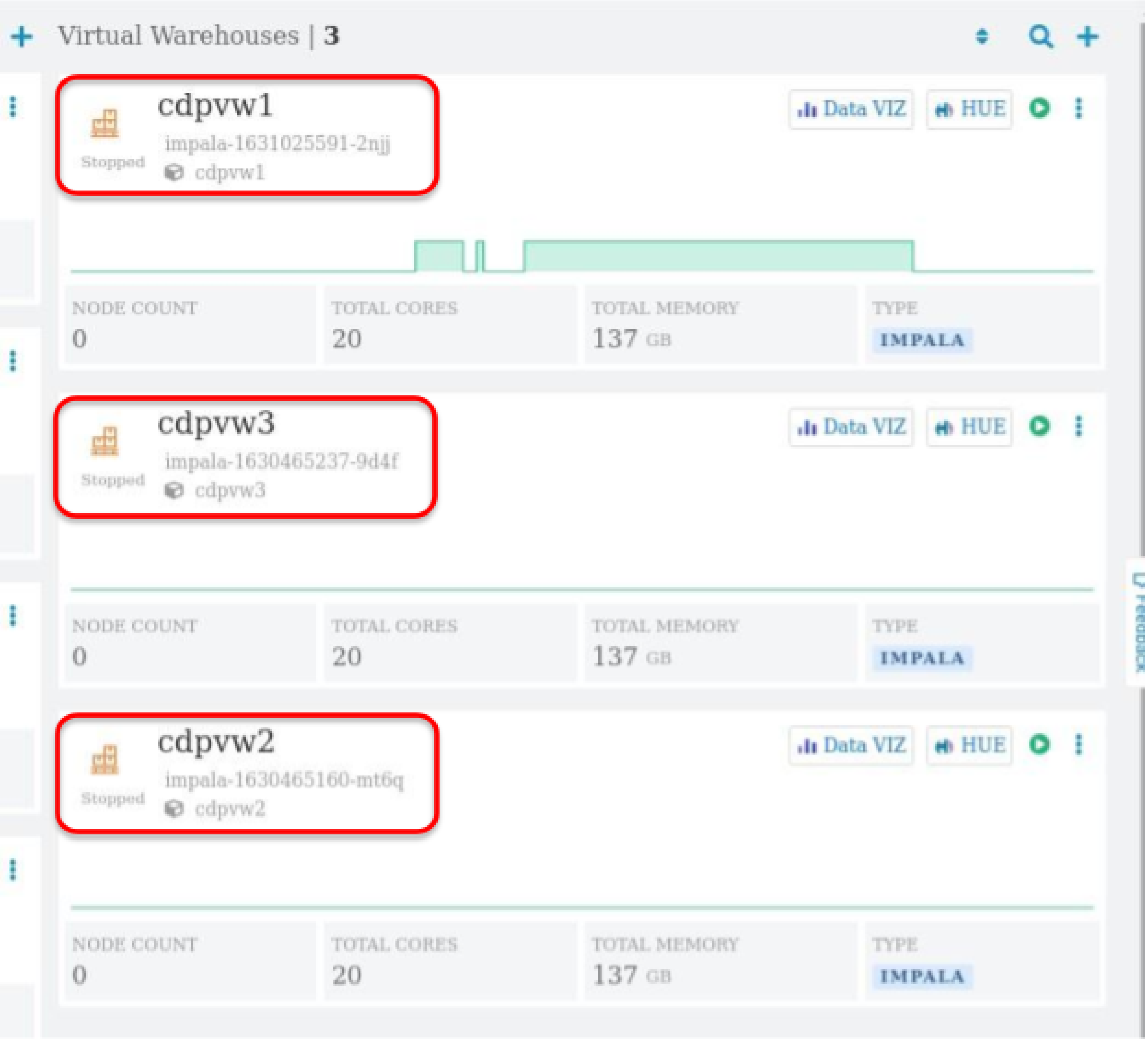
NODE COUNT	TOTAL CORES	TOTAL MEMORY	TYPE
2	38	292 GB	HIVE DATA VISUALIZATION

If you can read this it is the end of the background information and it is

TIME FOR YOU TO JUMP BACK IN AND DO THE LAB. *Please, Please, not drop any tables. Do not alter any tables. This is a shared environment. You all have*

the same userid and you have admin powers.

9) Notice there multiple Virtual Warehouses (VWs). You will be working on one of the VWs. If you are in Zoom Breakout Room 1 use cdpvw1, room2 uses cdpvw2 etc



Virtual Warehouses | 3

cdpvw1
Stopped
impala-1631025591-2njj
cdpvw1

NODE COUNT	TOTAL CORES	TOTAL MEMORY	TYPE
0	20	137 GB	IMPALA

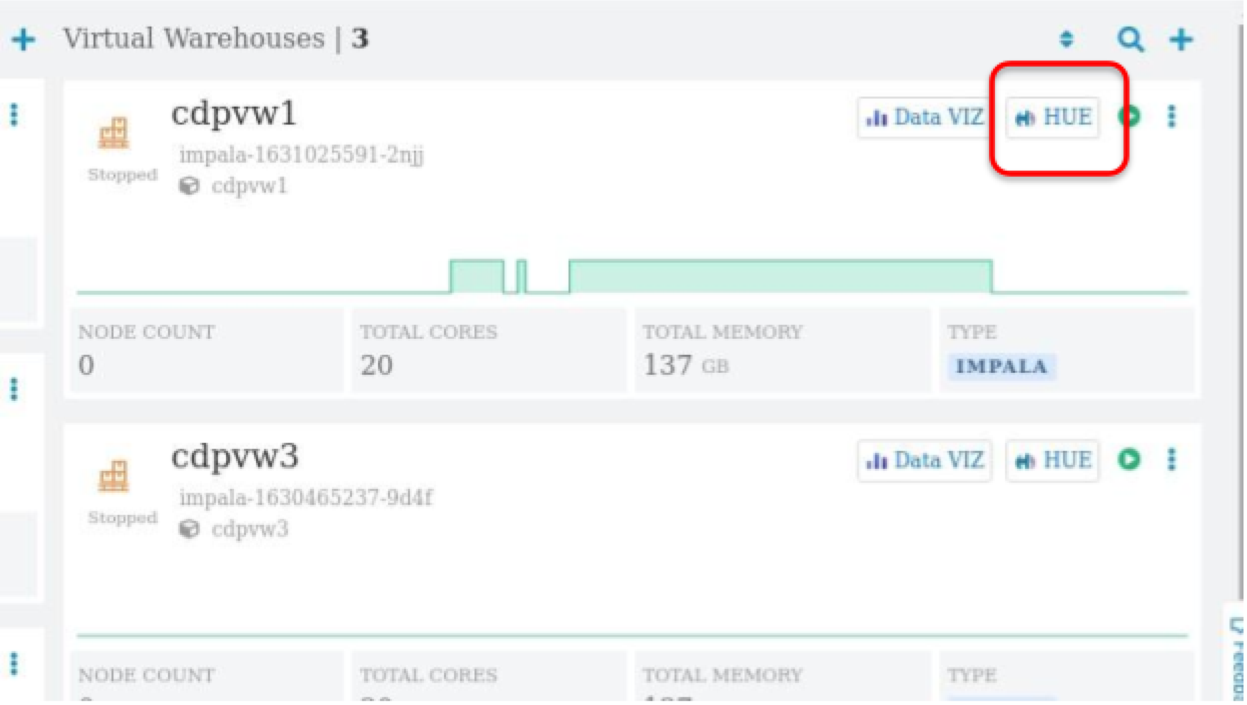
cdpvw3
Stopped
impala-1630465237-9d4f
cdpvw3

NODE COUNT	TOTAL CORES	TOTAL MEMORY	TYPE
0	20	137 GB	IMPALA

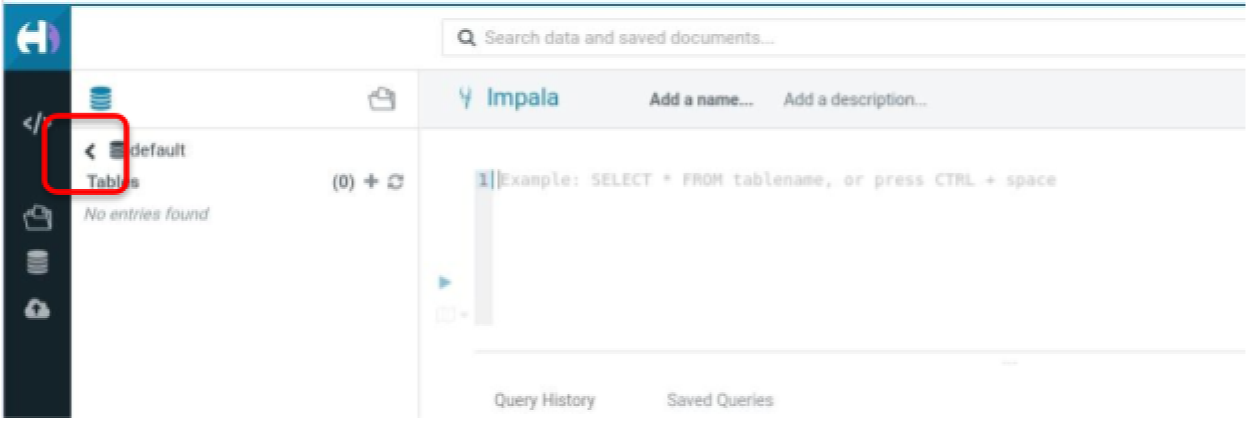
cdpvw2
Stopped
impala-1630465160-mt6q
cdpvw2

NODE COUNT	TOTAL CORES	TOTAL MEMORY	TYPE
0	20	137 GB	IMPALA

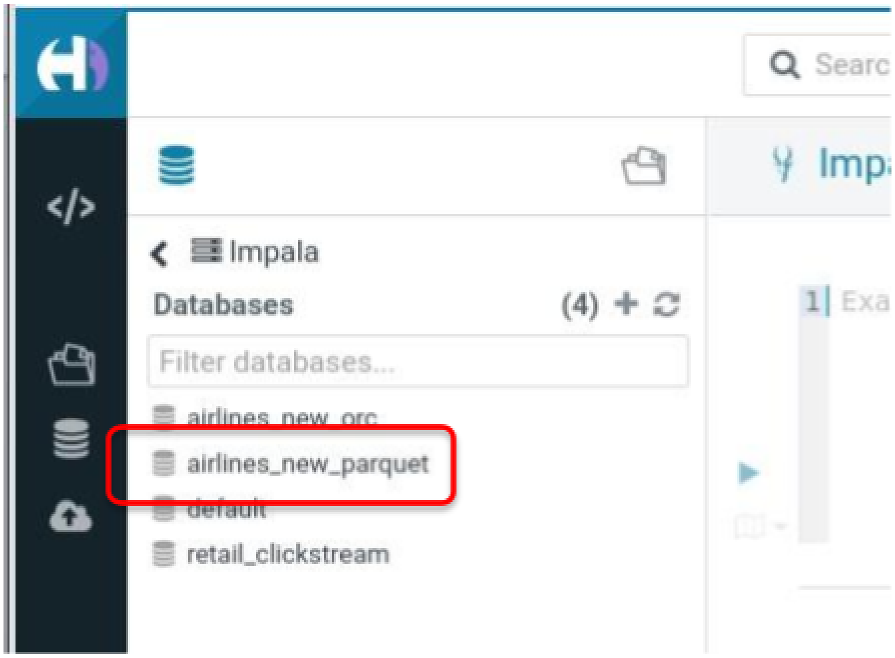
10) Click on HUE to enter the “Hadoop User Experience” in your designated room.



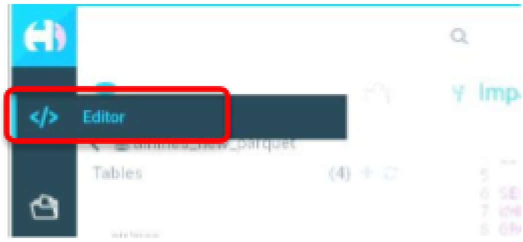
11) The landing page takes you to the “default” database. Click on the **<** to the left of the default database to select a different database



12) Click on the database “airlines_new_parquet” that we saw in Part 1 “Data Catalog.” Both Impala and Hive work with both Parquet and ORC files. As a rule of thumb if you’re mostly using Impala use Parquet format or Kudu. When working with Hive ORC is the preferred format.



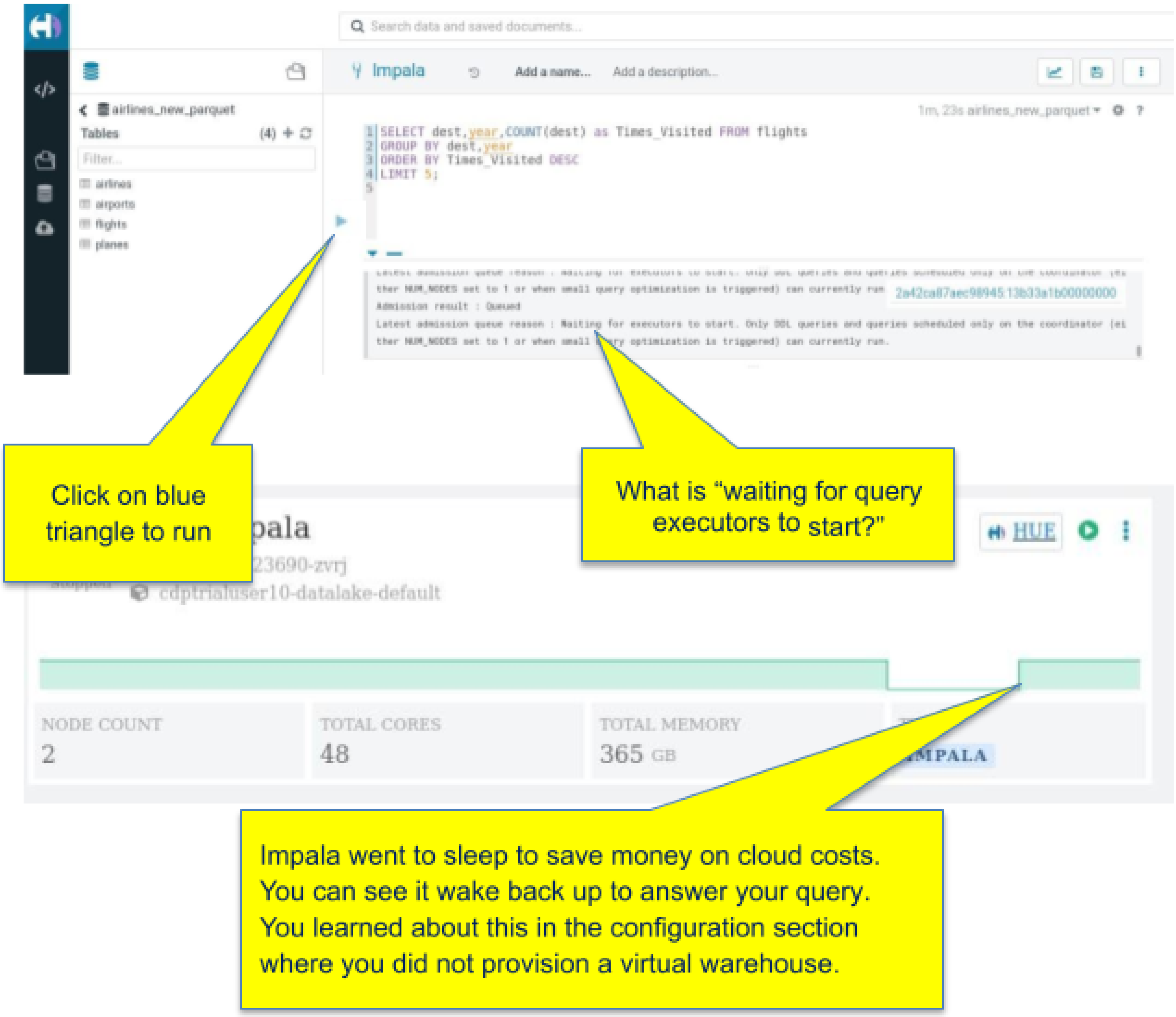
If you ever need to get back to this screen layout click on the “editor” shown here:



13) Enter the following query, answering the question “show me the top 5 visited destination by year from (1995-2008)”   Click on the blue triangle to run the query.

```
SELECT dest,year,COUNT(dest) as Times_Visited FROM flights
GROUP BY dest,year
ORDER BY Times_Visited DESC
LIMIT 5;
```

Why doesn’t it run right away? You promised it was fast ☺



```
SELECT dest,year,COUNT(dest) as Times_Visited FROM flights
GROUP BY dest,year
ORDER BY Times_Visited DESC
LIMIT 5;
```

```
Latest admission queue reason : Waiting for executors to start. Only DDL queries and queries scheduled only on the coordinator (ei
ther NUM_NODES set to 1 or when small query optimization is triggered) can currently run  2a42ca87aec98945:13b33a1b00000000
Admission result : Queued
Latest admission queue reason : Waiting for executors to start. Only DDL queries and queries scheduled only on the coordinator (ei
ther NUM_NODES set to 1 or when small query optimization is triggered) can currently run.
```

NODE COUNT 2

TOTAL CORES 48

TOTAL MEMORY 365 GB

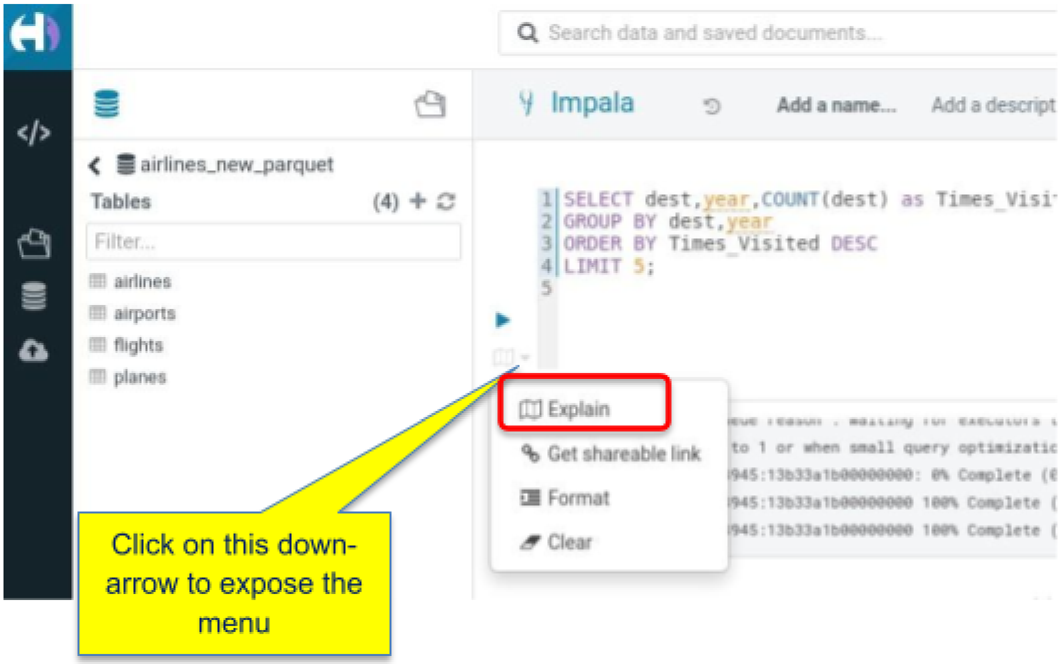
Click on blue triangle to run

What is “waiting for query executors to start?”

Impala went to sleep to save money on cloud costs. You can see it wake back up to answer your query. You learned about this in the configuration section where you did not provision a virtual warehouse.

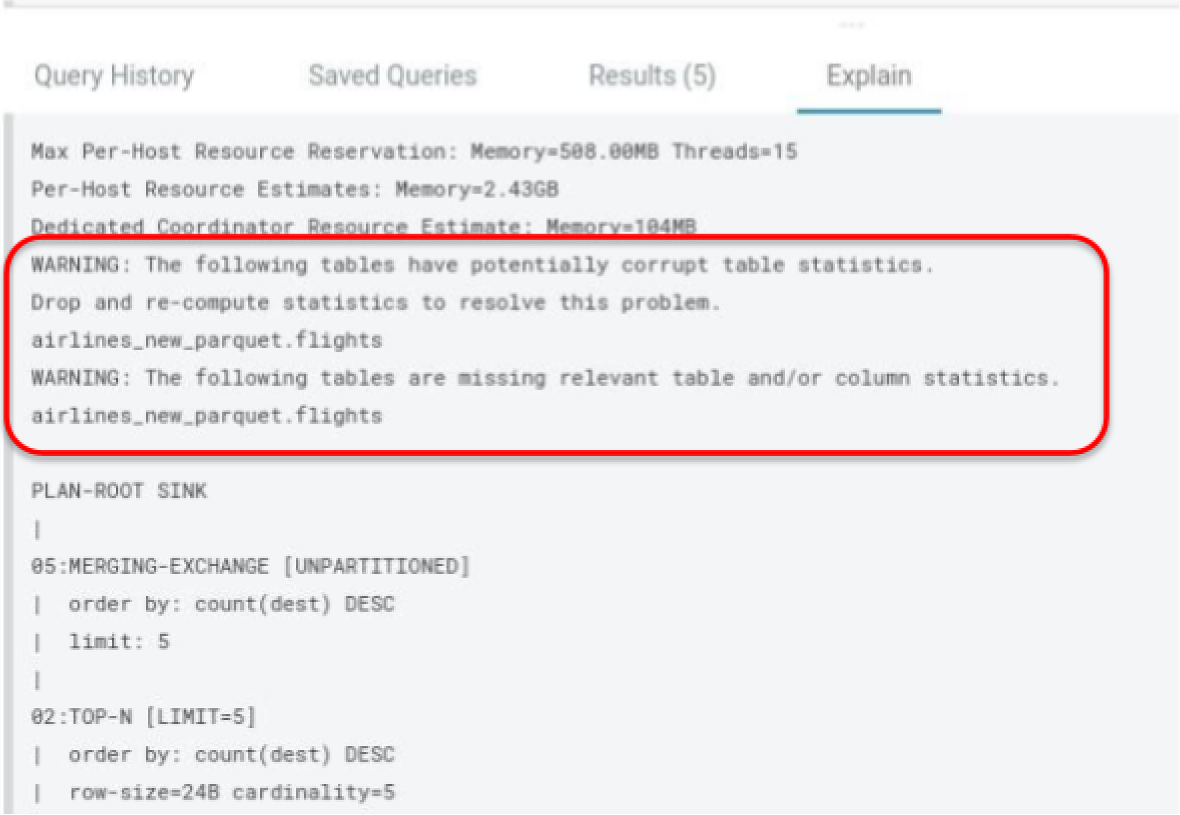
You may get the query result without having to wait for Impala to “wake up.” You get to decide how long an idle time you want to wait before you scale down to zero, or if you have the budget you can have Impala always available.

14) Click “EXPLAIN” to see the explain plan prior to running the query
*Not required to execute the query - this gives us a plan on exactly what the query is doing



```
SELECT dest,year,COUNT(dest) as Times_Visi
GROUP BY dest,year
ORDER BY Times_Visited DESC
LIMIT 5;
```

Click on this down-arrow to expose the menu



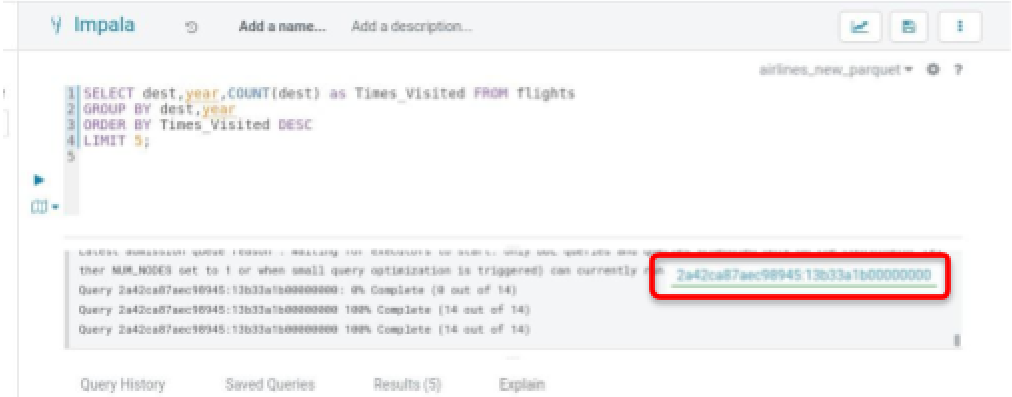
```
Max Per-Host Resource Reservation: Memory=508.00MB Threads=15
Per-Host Resource Estimates: Memory=2.43GB
Dedicated Coordinator Resource Estimate: Memory=104MB
WARNING: The following tables have potentially corrupt table statistics.
Drop and re-compute statistics to resolve this problem.
airlines_new_parquet.flights
WARNING: The following tables are missing relevant table and/or column statistics.
airlines_new_parquet.flights

PLAN-ROOT SINK
|
05:MERGING-EXCHANGE [UNPARTITIONED]
|  order by: count(dest) DESC
|  limit: 5
|
02:TOP-N [LIMIT=5]
|  order by: count(dest) DESC
|  row-size=24B cardinality=5
```

The “Explain Plan” shows you how the query will execute. It is asking you to update statistics for the tables in the query. This is a good idea for performance. After everyone reaches this point in the lab designate one person to run: `compute stats airlines_new_parquet.flights;`

We are in a shared environment. If someone else has done “compute stats” you won’t see the message. In STEP 4, after the DataViz lab, you will have a chance to create your own tables and work on compute stats with your unique tables.

We can discuss optimizer paths and table statistic in more detail as part of the breakout room.

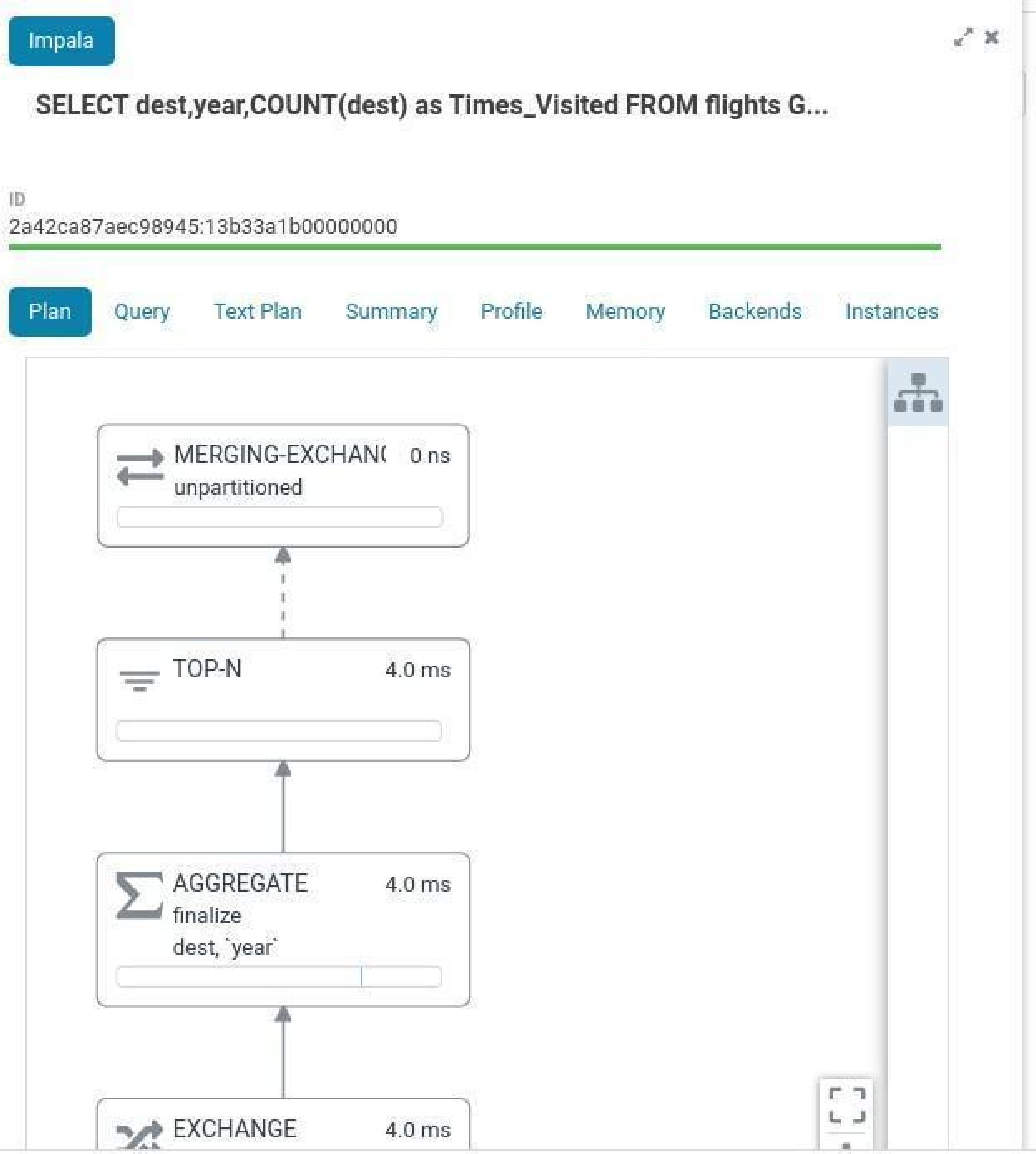
15) After you run the query you will have a link to lots of information about the query. Click on the link shown below



```
SELECT dest,year,COUNT(dest) as Times_Visited FROM flights
GROUP BY dest,year
ORDER BY Times_Visited DESC
LIMIT 5;
```

```
ther NUM_NODES set to 1 or when small query optimization is triggered) can currently r  2a42ca87aec98945:13b33a1b00000000
Query 2a42ca87aec98945:13b33a1b00000000: 0% Complete (0 out of 14)
Query 2a42ca87aec98945:13b33a1b00000000 100% Complete (14 out of 14)
Query 2a42ca87aec98945:13b33a1b00000000 100% Complete (14 out of 14)
```

16) Explore the different tabs in the query pop-up. The Visual Plan shows how the query was executed. These get more interesting with multi table joins. The Summary show how much time was spent in each stage of the query, the memory used, and the rows produced. The Profile includes the summary and great detail of everything that happened on each node running the query.



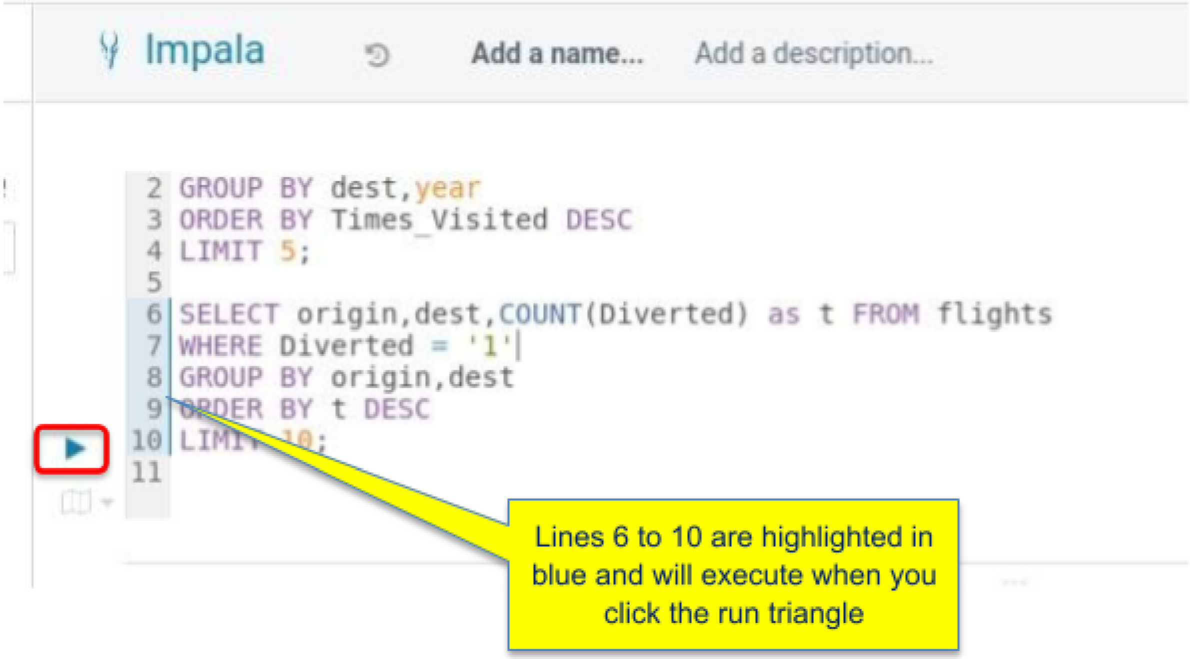
SELECT dest,year,COUNT(dest) as Times_Visited FROM flights G...

ID
2a42ca87aec98945:13b33a1b00000000

17) Add another query to the editor

```
SELECT origin,dest,COUNT(Diverted) as t FROM flights
WHERE Diverted = "1"
GROUP BY origin,dest
ORDER BY t DESC
LIMIT 10;
```

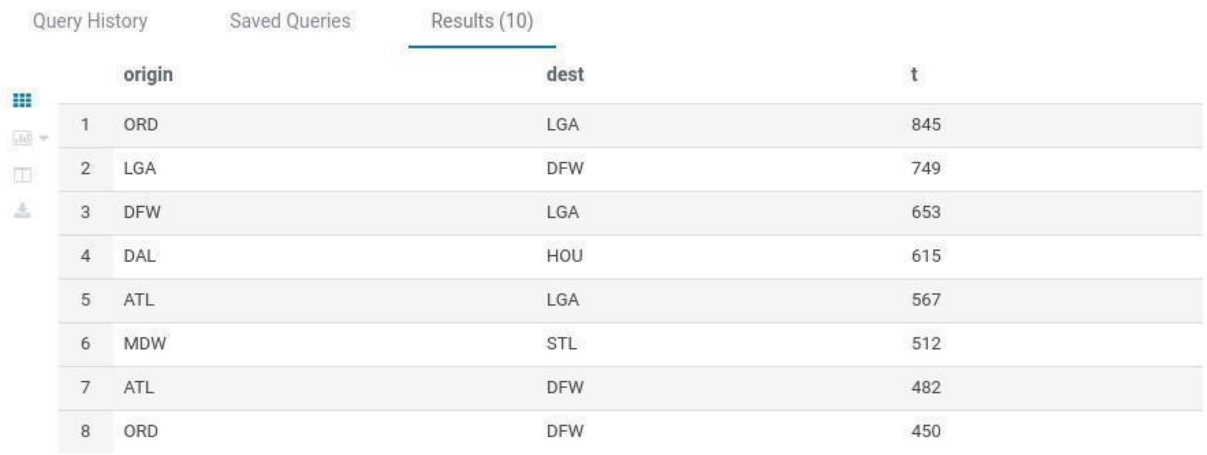
18) Notice the highlighting on the left edge. HUE is parsing based on the semi-colon and the execution arrow will run whatever is highlighted in blue, or whatever has been highlighted by the cursor. This way you can have multiple queries in the same canvas.



```
 2 GROUP BY dest,year
 3 ORDER BY Times_Visited DESC
 4 LIMIT 5;
 5
 6 SELECT origin,dest,COUNT(Diverted) as t FROM flights
 7 WHERE Diverted = '1'
 8 GROUP BY origin,dest
 9 ORDER BY t DESC
10 LIMIT 10;
11
```

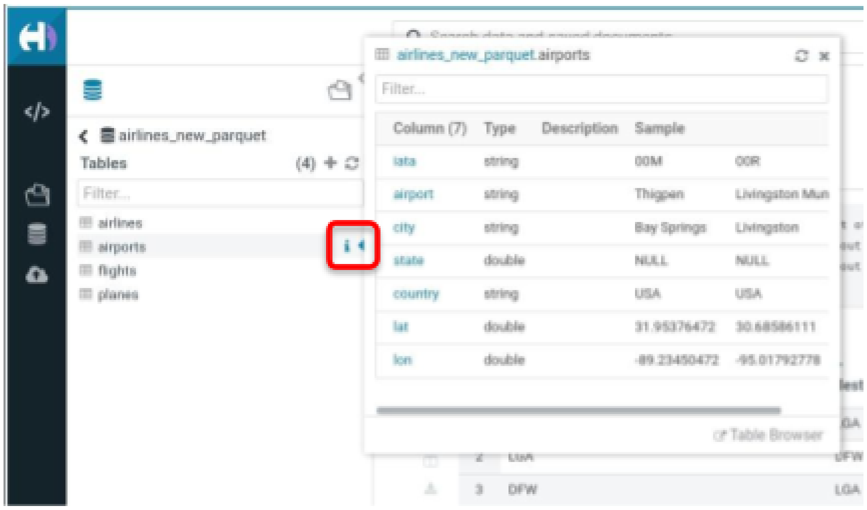
Lines 6 to 10 are highlighted in blue and will execute when you click the run triangle

19) Click the blue triangle to execute the query, answering the question “What are the top 10 routes (origin and dest) that have seen maximum diversions?”



	origin	dest	t
1	ORD	LGA	845
2	LGA	DFW	749
3	DFW	LGA	653
4	DAL	HOU	615
5	ATL	LGA	567
6	MDW	STL	512
7	ATL	DFW	482
8	ORD	DFW	450

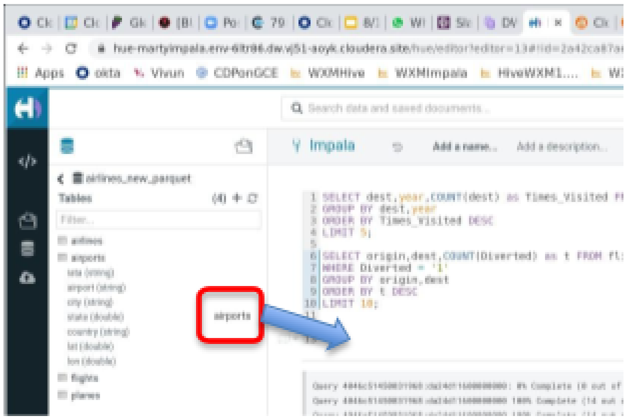
20) Hover over the disappearing **i** next to the airports table to see more information about the table. We’re going to use the geo location of the airports to do a marker map in HUE.



Column (7)	Type	Description	Sample	
iata	string		00M	00R
airport	string		Thigpen	Livingston Mun
city	string		Bay Springs	Livingston
state	double		NULL	NULL
country	string		USA	USA
lat	double		31.95376472	30.68586111
lon	double		-89.23450472	-95.01792778

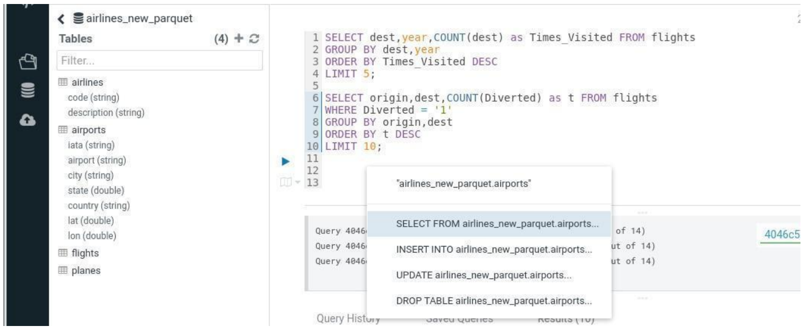
You can also click on the table name to expand all the columns. On the far right of the browser is help tooling.

21) Try out the “drag and drop” option, drag and drop the table name “airports over to line 12



```
1 SELECT dest,year,COUNT(dest) as Times_Visited F
2 GROUP BY dest,year
3 ORDER BY Times_Visited DESC
4 LIMIT 5;
5
6 SELECT origin,dest,COUNT(Diverted) as t FROM fl
7 WHERE Diverted = '1'
8 GROUP BY origin,dest
9 ORDER BY t DESC
10 LIMIT 10;
```

When you drop the table name you’ll get to choose what SQL you want auto generated. Take the “select” option and run the query with the blue-triangle



```
1 SELECT dest,year,COUNT(dest) as Times_Visited FROM flights
2 GROUP BY dest,year
3 ORDER BY Times_Visited DESC
4 LIMIT 5;
5
6 SELECT origin,dest,COUNT(Diverted) as t FROM flights
7 WHERE Diverted = '1'
8 GROUP BY origin,dest
9 ORDER BY t DESC
10 LIMIT 10;
11
12
13
```

"airlines_new_parquet.airports"

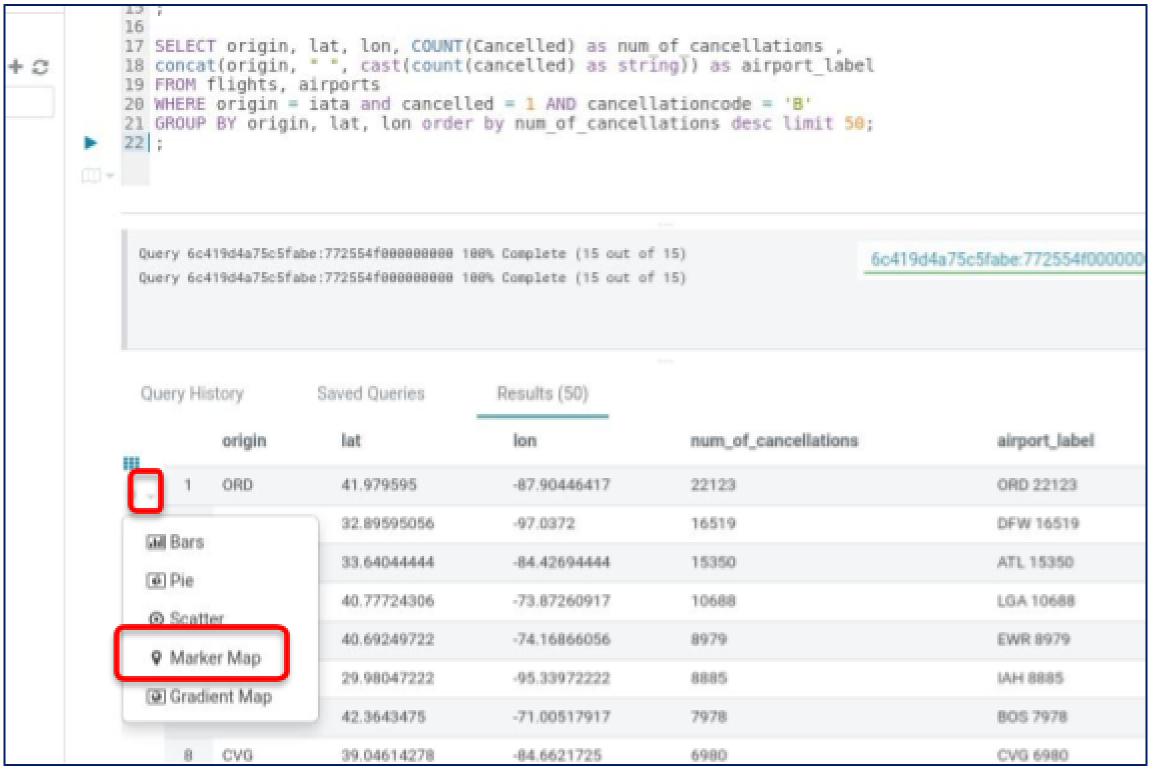
SELECT FROM airlines_new_parquet.airports...
INSERT INTO airlines_new_parquet.airports...
UPDATE airlines_new_parquet.airports...
DROP TABLE airlines_new_parquet.airports...

```
Query 4046      of 14)
Query 4046     ut of 14)
Query 4046     ut of 14)
```

22) Let’s now build a marker map of the airports with the most cancellations. This will correlate with the airports that have the most flights. Run the SQL shown below

```
SELECT origin, lat, lon, COUNT(Cancelled) as num_of_cancellations ,
concat(origin, " ", cast(count(cancelled) as string)) as airport_label
FROM flights, airports
WHERE origin = iata and cancelled = 1 AND cancellationcode = 'B'
GROUP BY origin, lat, lon order by num_of_cancellations desc limit 50;
```

23) After you run the SQL use the down arrow to choose the type of output formatting. You’ve been using the data grid, we’re now going to choose the marker-map

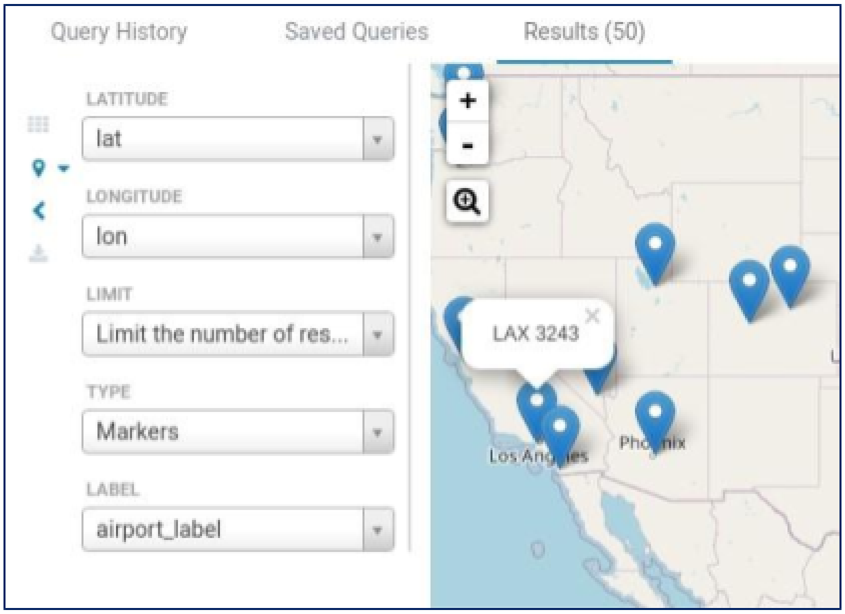


```
;

SELECT origin, lat, lon, COUNT(Cancelled) as num_of_cancellations ,
concat(origin, " ", cast(count(cancelled) as string)) as airport_label
FROM flights, airports
WHERE origin = iata and cancelled = 1 AND cancellationcode = 'B'
GROUP BY origin, lat, lon order by num_of_cancellations desc limit 50;
;
```

```
Query 6c419d4a75c5fabe:772554f000000000 100% Complete (15 out of 15)
Query 6c419d4a75c5fabe:772554f000000000 100% Complete (15 out of 15)
```

	origin	lat	lon	num_of_cancellations	airport_label
1	ORD	41.979595	-87.90446417	22123	ORD 22123
		32.89595056	-97.0372	16519	DFW 16519
		33.64044444	-84.42694444	15350	ATL 15350
		40.77724306	-73.87260917	10688	LGA 10688
		40.69249722	-74.16866056	8979	EWR 8979
		29.98047222	-95.33972222	8885	IAH 8885
		42.3643475	-71.00517917	7978	BOS 7978
8	CVG	39.04614278	-84.6621725	6980	CVG 6980

24) Configure the marker map per shown below. Clicking on one of the markers will pop up the value of the “ariport_label” column



LATITUDE: lat

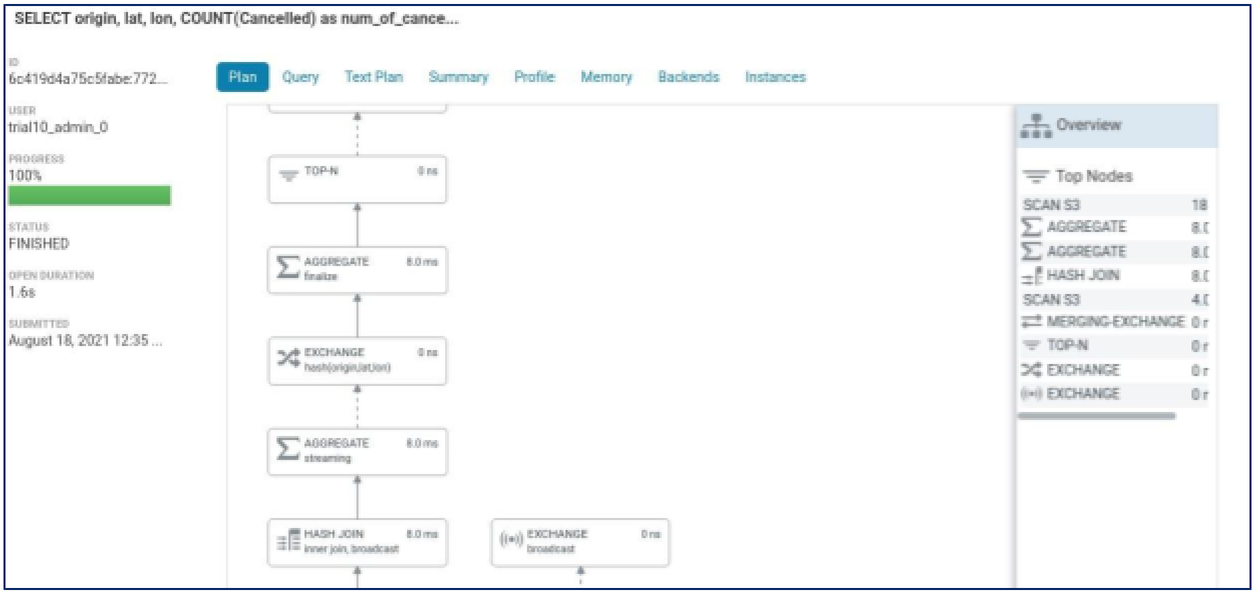
LONGITUDE: lon

LIMIT: Limit the number of res...

TYPE: Markers

LABEL: airport_label

25) Look at the query plan – notice we now have a join in the tree



SELECT origin, lat, lon, COUNT(Cancelled) as num_of_cance...

ID
6c419d4a75c5fabe:772...

USER
trial10_admin_0

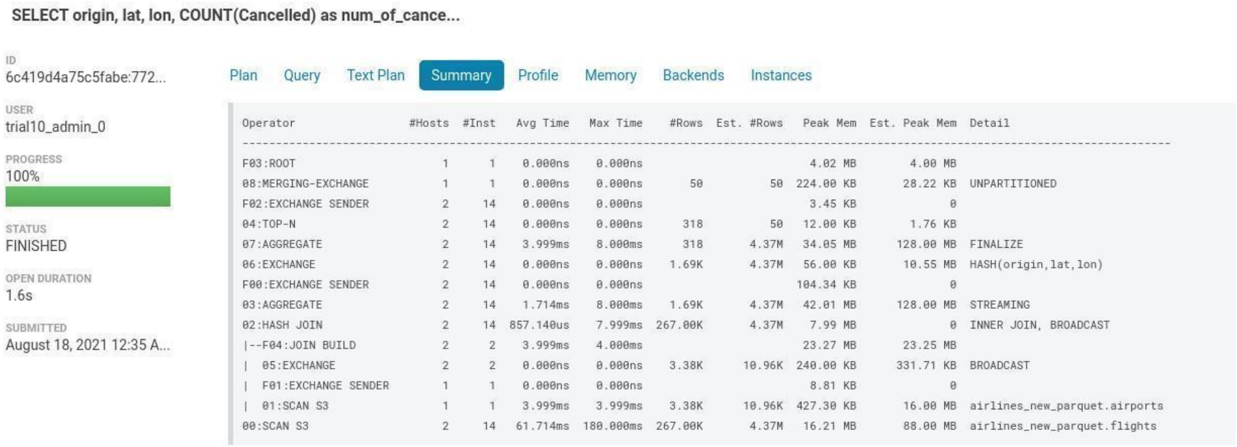
PROGRESS
100%

STATUS
FINISHED

OPEN DURATION
1.6s

SUBMITTED
August 18, 2021 12:35 ...

26) The summary shows details of all the stages in the join and their metrics



SELECT origin, lat, lon, COUNT(Cancelled) as num_of_cance...

ID
6c419d4a75c5fabe:772...

USER
trial10_admin_0

PROGRESS
100%

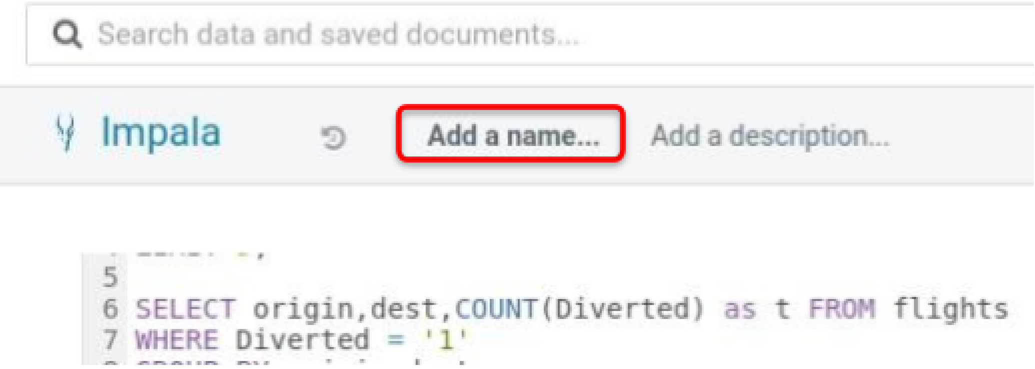
STATUS
FINISHED

OPEN DURATION
1.6s

SUBMITTED
August 18, 2021 12:35 A...

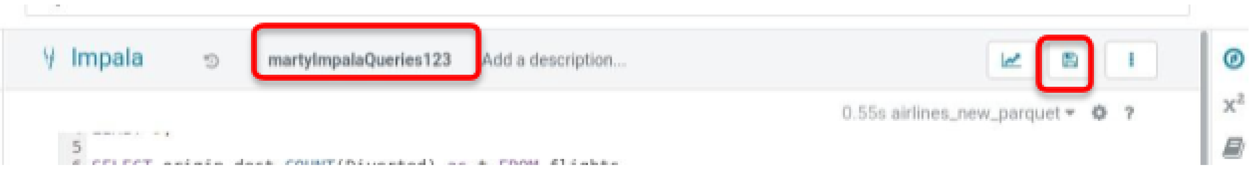
```
Operator              #Hosts  #Inst   Avg Time   Max Time    #Rows  Est. #Rows   Peak Mem  Est. Peak Mem  Detail
--------------------------------------------------------------------------------------------------------------------
F03:ROOT                   1      1    0.000ns    0.000ns                         4.02 MB        4.00 MB
08:MERGING-EXCHANGE        1      1    0.000ns    0.000ns       50          50  224.00 KB       28.22 KB  UNPARTITIONED
F02:EXCHANGE SENDER        2     14    0.000ns    0.000ns                         3.45 KB              0
04:TOP-N                   2     14    0.000ns    0.000ns      318          50   12.00 KB        1.76 KB
07:AGGREGATE               2     14    3.999ms    8.000ms      318       4.37M   34.05 MB      128.00 MB  FINALIZE
06:EXCHANGE                2     14    0.000ns    0.000ns    1.69K       4.37M   56.00 KB       10.55 MB  HASH(origin,lat,lon)
F00:EXCHANGE SENDER        2     14    0.000ns    0.000ns                       104.34 KB              0
03:AGGREGATE               2     14    1.714ms    8.000ms    1.69K       4.37M   42.01 MB      128.00 MB  STREAMING
02:HASH JOIN               2     14  857.140us    7.999ms  267.00K       4.37M    7.99 MB              0  INNER JOIN, BROADCAST
|--F04:JOIN BUILD          2      2    3.999ms    4.000ms                        23.27 MB       23.25 MB
|  05:EXCHANGE             2      2    0.000ns    0.000ns    3.38K      10.96K  240.00 KB      331.71 KB  BROADCAST
|  F01:EXCHANGE SENDER     1      1    0.000ns    0.000ns                         8.81 KB              0
|  01:SCAN S3              1      1    3.999ms    3.999ms    3.38K      10.96K  427.30 KB       16.00 MB  airlines_new_parquet.airports
00:SCAN S3                 2     14   61.714ms  180.000ms  267.00K       4.37M   16.21 MB       88.00 MB  airlines_new_parquet.flights
```

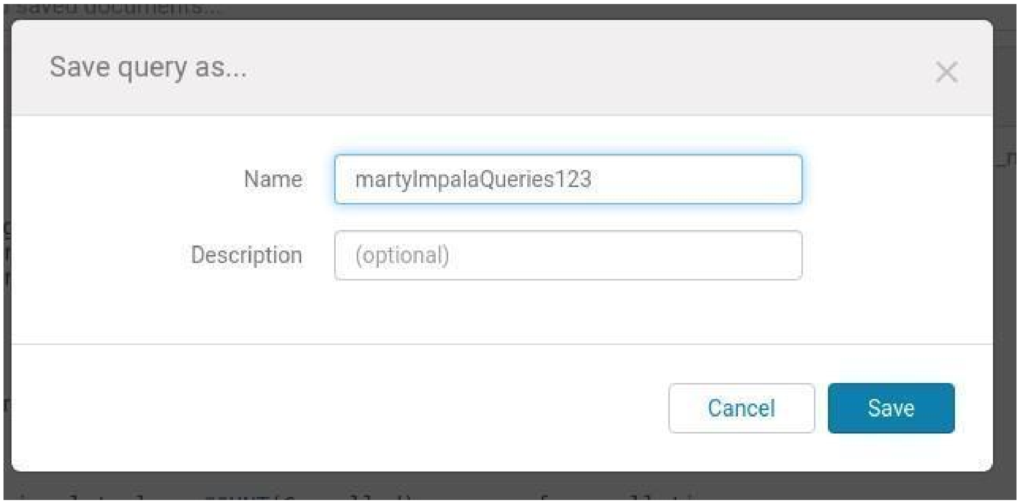
27) Time to save our work. Give your Impala SQL a name. For the lab use yourNameXXX where XXX is where you get to be creative. Use something unique, this is a multi user environment.



```
SELECT origin,dest,COUNT(Diverted) as t FROM flights
WHERE Diverted = '1'
```

Then click “Save” and then “Save” in the popup

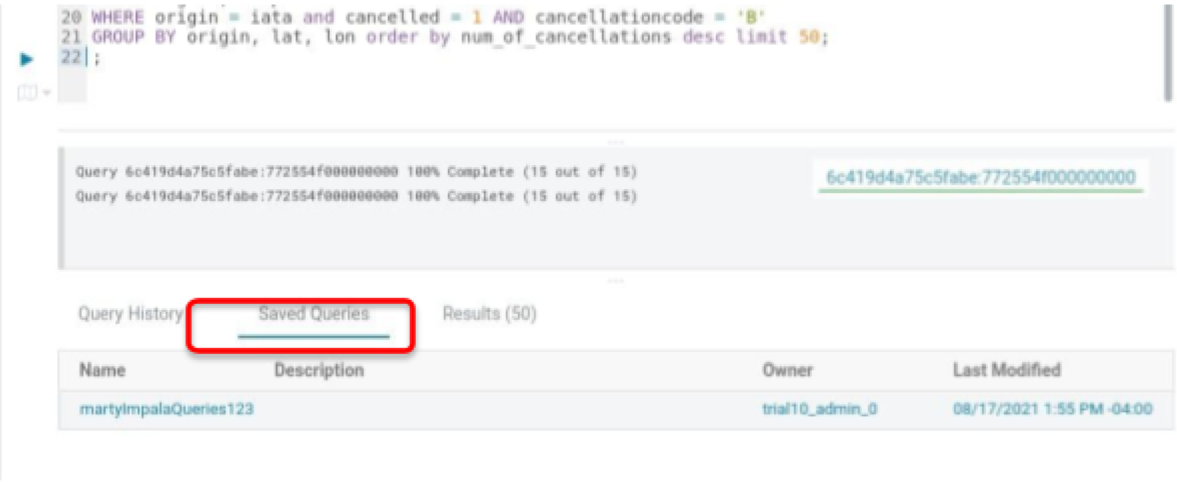




Name: martyImpalaQueries123

Description: (optional)

Your saved queries will show up under the “Saved Queries” heading.



```
WHERE origin = iata and cancelled = 1 AND cancellationcode = 'B'
GROUP BY origin, lat, lon order by num_of_cancellations desc limit 50;
;
```

```
Query 6c419d4a75c5fabe:772554f000000000 100% Complete (15 out of 15)
Query 6c419d4a75c5fabe:772554f000000000 100% Complete (15 out of 15)
```

Name	Description	Owner	Last Modified
martyImpalaQueries123		trial10_admin_0	08/17/2021 1:55 PM -04:00