

CLOUDERA DATA PLATFORM

DATA WAREHOUSE LAB –

Experimental, you are welcome to try this lab or use the prior version

Step-by-step instructions:

Part 1 - Data Catalog [20 minutes]

Overview: What is Cloudera Data Catalog?

Data Catalog is a service that enables you to understand, manage, secure, and govern data assets across the enterprise. Data Catalog helps you understand data across multiple clusters and across multiple CDP environments. You can search to locate relevant data of interest based on various parameters. Using Data Catalog, you can understand how data is interpreted for use, how it is created and modified, and how data access is secured and protected.

Purpose: Search for a dataset (table) in Data Catalog, called “flights”.

- Find what database(s) the table “flights” is located.
- Find out at least one year that the “flights” table was generated from.
- Find out how many columns the table “flights” contains.

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 Catalog” within the CDP Home Screen



3) Type “flights” in the search box and press enter on your keyboard

Data Catalog / Search

Launch Profilers Action ▾

Q flights

Entities

- actualelapsedtime (hive_column)
- securitydelay (hive_column)
- year (hive_column)
- cancelled (hive_column)
- weatherdelay (hive_column)

Suggestions

- flights

4) Click “Hive Table” under Filters on the left

Launch Profilers

Q flights

Data Lakes

	Type	Name	Location
<input type="checkbox"/>	Hive Table	flights	/airlines_new_orc
<input type="checkbox"/>	Hive Table	flights	/airlines_new_parquet

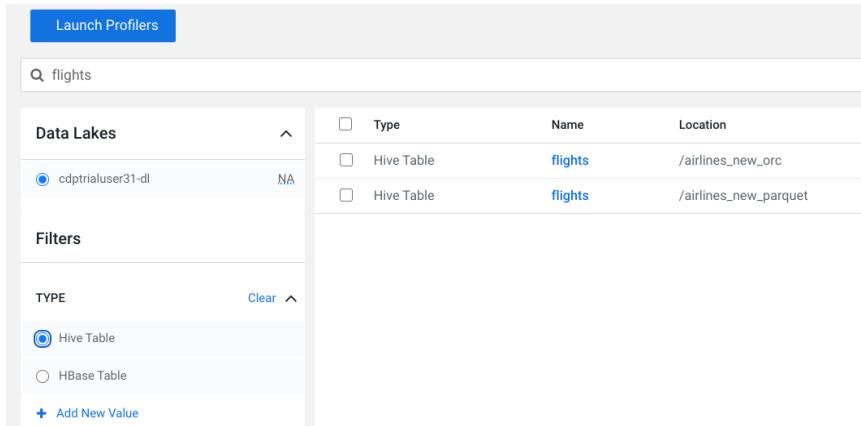
Filters

TYPE

Hive Table

HBase Table

[+ Add New Value](#)



*Find what database(s) the table “flights” is located.

5) Click “flights” where the Location = /airlines_new_orc

Launch Profilers

Q flights

Data Lakes

	Type	Name	Location
<input type="checkbox"/>	Hive Table	flights	/airlines_new_orc
<input type="checkbox"/>	Hive Table	flights	/airlines_new_parquet

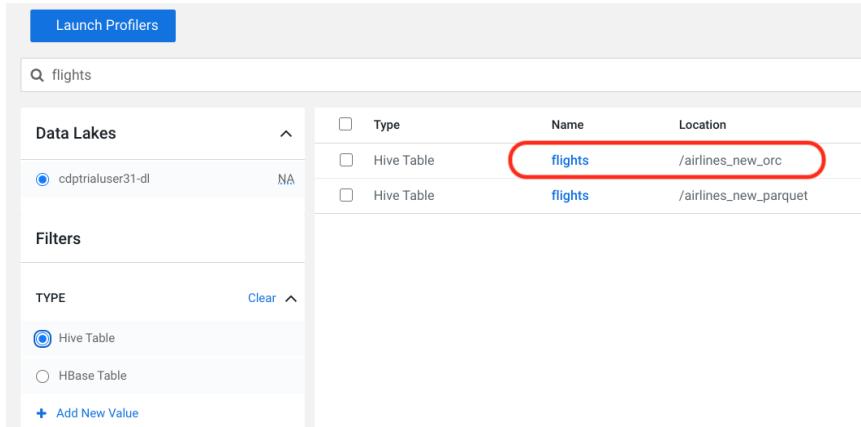
Filters

TYPE

Hive Table

HBase Table

[+ Add New Value](#)



6) Zoom into the Lineage and scroll over one of the /cdp-lake/data, clicking the “i” for more information

The screenshot shows the 'flights' table details in the Cloudera Data Catalog. Key information includes:

- Name:** flights
- Type:** HIVE TABLE
- Data Lake:** cdptrialuser31-dl
- Dataset:** 0
- Number of Columns:** 29
- Asset Properties:**
 - Owner: csso_trialuser31
 - Qualified Name: airlines_new_orc.flights@cm
 - Created On: Wed Jan 13 2021 01:10:57 GMT-0600 (Central Standard Time)
 - Last Access Time: Wed Jan 13 2021 01:10:57 GMT-0600 (Central Standard Time)
- Table Type:** MANAGED_TABLE
- Database:** airlines_new_orc
- DB Catalog:** cm
- Parent:** airlines_new_orc
- Managed Classifications:** 0
- Propagated Tags:**

Lineage: A diagram showing the lineage of the 'flights' table. It has five blue circular nodes representing data sources, each connected by a green line to a central green circular node labeled 'flights'. A red circle highlights one of the source nodes.

Entity Detail: A modal window for the entity '/cdp-lake/data/airlines/airlines_new_orc.db/flights/year=1997'.

Guid:	be5a2094-f572-432e-9fa4-96498f7db650
Type Name:	aws_s3_pseudo_dir
Classifications(0):	-
Owner:	-NA-
Qualified Name:	s3://prod-cdptrialuser31-trycdp-com/cdp-lake/data/airlines/airlines_new_orc.db/flights/year=1997@cm
Created On:	-NA-
Update Time:	-NA-
Created By:	csso_trialuser31
Updated By:	csso_trialuser31

*Find out at least one year that the “flights” table was generated from.

*Find out how many columns the table “flights” contains.

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 #

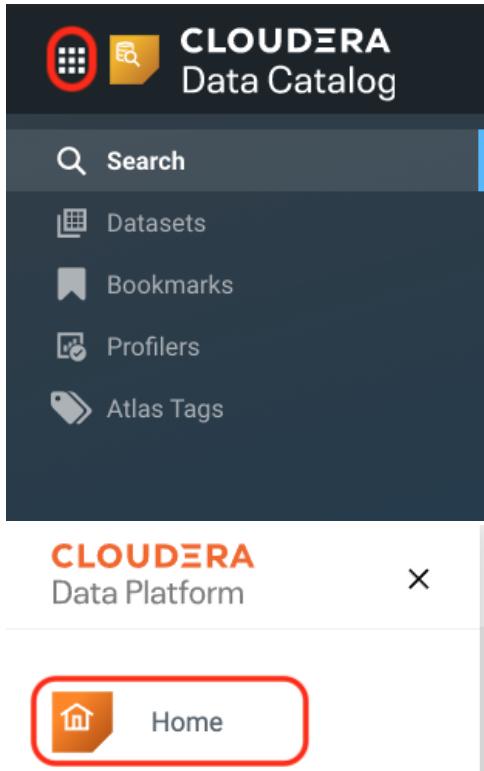
*X represents the password

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”



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

3) Click the “+” at the top right next to “Virtual Warehouses”

The screenshot shows the 'Virtual Warehouses' creation interface. At the top, there is a header 'Virtual Warehouses | 1' and a red circle with a plus sign (+) button. Below the header, there are fields for 'Name' (with placeholder 'Enter Virtual Warehouse Name'), 'Type' (with options 'HIVE' and 'IMPALA' where 'HIVE' is selected), 'Database Catalog' (set to 'cdptrialuser24-dl-default'), and 'Size' (set to '-- select an option --'). Below these fields, a table lists a single virtual warehouse named 'default-vw'. The table includes columns for 'NODE COUNT' (0), 'TOTAL CORES' (12), 'TOTAL MEMORY' (56 GB), and 'TYPE' (HIVE COMPACTOR). The 'default-vw' row has a gear icon, a stop icon, and a three-dot menu icon.

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

4) 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) Select the Size of “xsmall - 2 Executor Nodes”

*How do I choose a size? Initial concurrent users

Virtual Warehouses | 1

New Virtual Warehouse

Name *

Type *

HIVE IMPALA

Database Catalog *

cdptrialuser24-dl-default

Size *

-- select an option --

✓ xsmall - 2 Executor Nodes

small - 10 Executor Nodes

medium - 20 Executor Nodes

large - 40 Executor Nodes

custom

6) 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 *

Type *

HIVE IMPALA

Database Catalog *

cdptrialuser24-dl-default

Size *

xsmall - 2 Executor Nodes

AutoSuspend Timeout (in seconds): 5000

0 1000 2000 3000 4000 5000 6000 7000

7) Choose “Install Data Visualization” to be on

*Allowing for Data Visualizations in Part 3

Virtual Warehouses | 2

New Virtual Warehouse

Name *

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

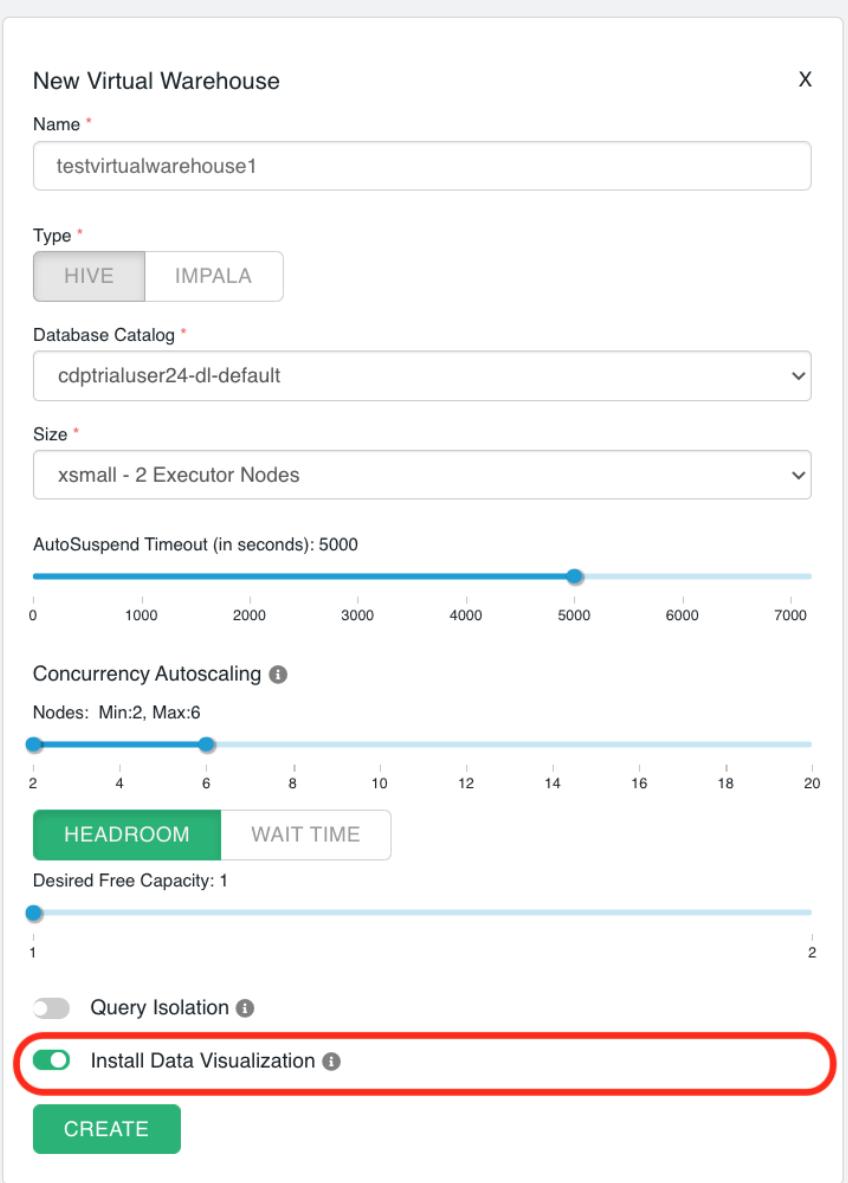
HEADROOM WAIT TIME

Desired Free Capacity: 1

Query Isolation ⓘ

Install Data Visualization ⓘ

CREATE



8) 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

OK, TIME FOR YOU TO JUMP BACK IN. THE DATA ANALYTICS STUDIO (DAS) IS DEPRICATED. IT IS REPLACED WITH HUE. THE INSTRUCTIONS BELOW SHOW YOU HUE WITH IMPALA. IF YOU WANT TO USE DAS AND HIVE THOSE INSTRUCTIONS ARE AFTER THE IMPALA LABS

9) Notice there are two virtual warehouses built on the cdptrial data catalog. Only the “default” virtual warehouse is enabled for Data VIZ

+ Virtual Warehouses | 30

	martyimpala	HUE	⋮
Running	impala-1629223690-zvrj cdptrialuser10-datalake-default		
NODE COUNT 2 TOTAL CORES 48 TOTAL MEMORY 365 GB TYPE IMPALA			
	default	DAS Data VIZ	⋮
Running	compute-1629217181-qsn5 cdptrialuser10-datalake-default		
NODE COUNT 2 TOTAL CORES 38 TOTAL MEMORY 292 GB TYPE HIVE UNIFIED ANALYTICS COMPACTOR			

10) Click on HUE to enter the “Hadoop User Experience”

The screenshot shows the Hue interface with a list of virtual warehouses. At the top, there's a search bar and a '+' button. Below it, the 'Virtual Warehouses' section lists two entries:

- martyimpala**: Running, impala-1629223690-zvrj, cdptrialuser10-datalake-default. It has 2 nodes, 48 cores, 365 GB memory, and is of type IMPALA.
- default**: Running, compute-1629217181-qsn5, cdptrialuser10-datalake-default. It has 2 nodes, 38 cores, 292 GB memory, and is of type HIVE, UNIFIED ANALYTICS, COMPACTOR.

A red box highlights the 'HUE' button in the top right corner of the interface.

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

The screenshot shows the Impala database landing page in Hue. On the left, there's a sidebar with icons for Home, Databases, Tables, and Cloud Storage. A red box highlights the 'Tables' icon. The main area shows the 'Impala' database selected. The left sidebar shows the 'default' database selected. The right side has a search bar, a text input field with placeholder text 'Example: SELECT * FROM tablename, or press CTRL + space', and buttons for 'Add a name...', 'Add a description...', 'Query History', and 'Saved Queries'.

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.

The screenshot shows the Impala UI interface. On the left is a sidebar with icons for databases, tables, and clouds. The main area displays a list of databases under the heading 'Databases'. The 'airlines_new_parquet' database is listed and has a red box drawn around it. Other databases shown are 'airlines new orc', 'default', and 'retail_clickstream'. A search bar at the top right contains the word 'Search'.

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

This screenshot shows the same Impala UI interface, but the 'Editor' tab is now selected, indicated by a red box around its label. The rest of the interface remains the same, showing the databases list.

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 😊

This screenshot shows the Impala UI with a query entered in the editor. The query is:

```
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;
```

Below the editor, a status message indicates: "Latest admission queue reason : Waiting for executors to start. Only DDL queries and queries scheduled only on the coordinator (either NUM_NODES set to 1 or when small query optimization is triggered) can currently run. Admission result : Queued".

This will be a
blue triangle,
click it to run

What is “waiting for query
executors to start?”

martyimpala
Stopped
impala-1629223690-zvrj
cdptrialuser10-datalake-default

NODE COUNT	TOTAL CORES	TOTAL MEMORY	TYPE
2	48	365 GB	IDLE

Impala went to sleep to save money on cloud costs. You can see it wake back up to answer your query

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

Search data and saved documents...

Impala

Add a name... Add a description

Tables (4) +

airlines_new_parquet

Filter...

airlines airports flights planes

1 SELECT dest,year,COUNT(dest) as Times_Visited
2 GROUP BY dest,year
3 ORDER BY Times_Visited DESC
4 LIMIT 5;

Explain Get shareable link Format Clear

Click on this down-arrow to expose the menu

Query History Saved Queries Results (5) Explain

```

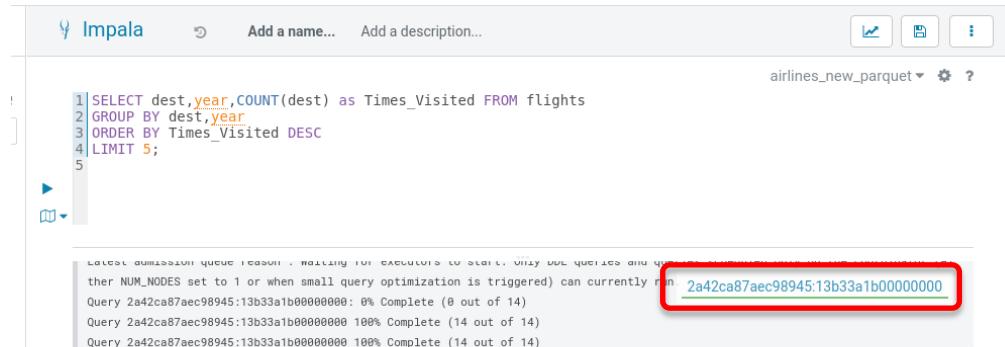
Max Per-Host Resource Reservation: Memory=508.00MB Threads=15
Per-Host Resource Estimates: Memory=2.43GB
Dedicated Coordinator Resource Estimate: Memory=104MB
WARNING: The following tables have potentially corrupt table statistics.
Drop and re-compute statistics to resolve this problem.
airlines_new_parquet.flights
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. We can discuss this in more detail.

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



Impala Add a name... Add a description... 

airlines_new_parquet ▾ * ?

```

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

```

Latest submitted queue reason : Waiting for executors to start. Only one queries and queri
ther NUM_NODES set to 1 or when small query optimization is triggered) can currently run [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)

Query History Saved Queries Results (5) Explain

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.

Impala

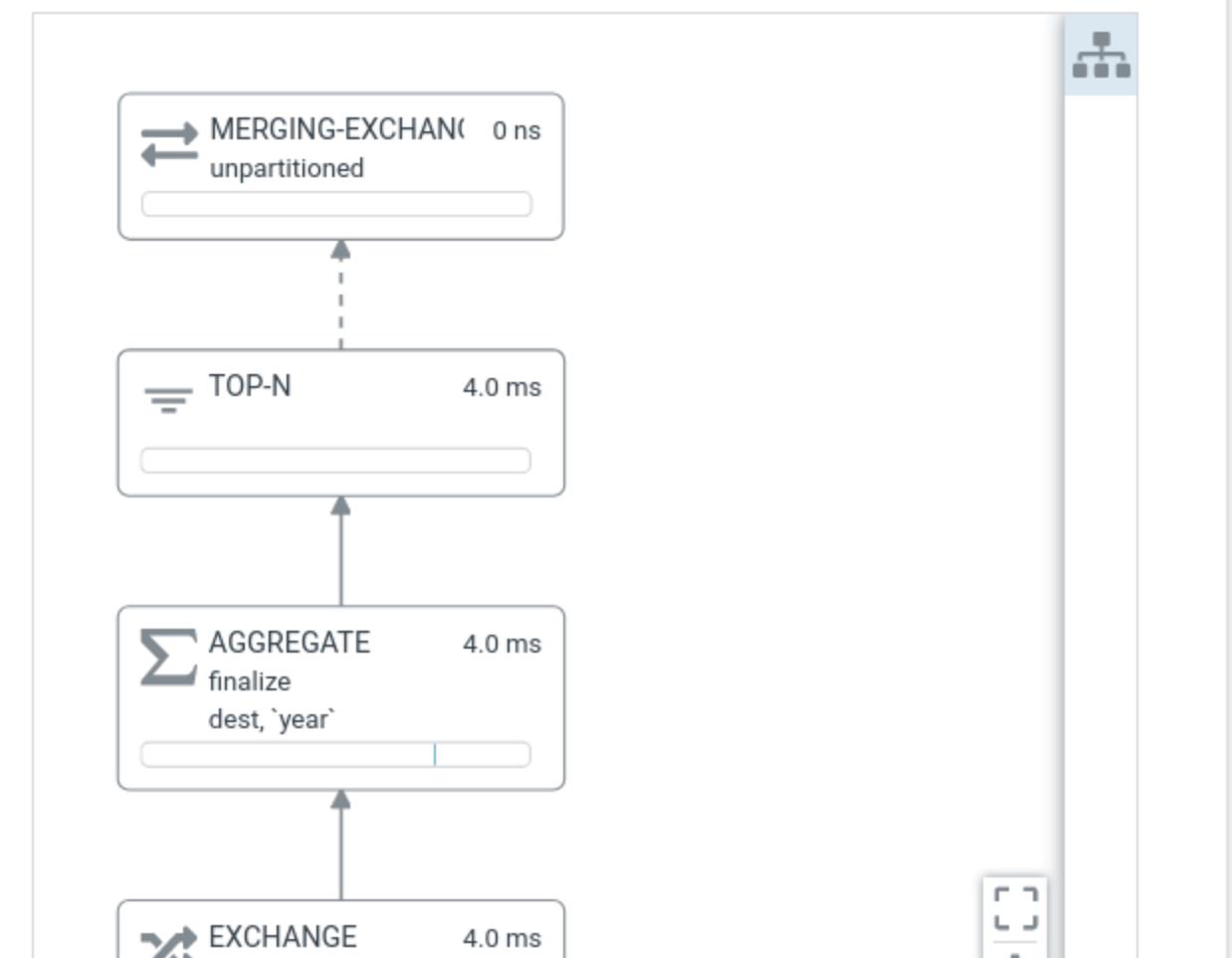


SELECT dest,year,COUNT(dest) as Times_Visited FROM flights G...

ID

2a42ca87aec98945:13b33a1b00000000

Plan Query Text Plan Summary Profile Memory Backends Instances



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
```

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

Query History Saved Queries Results (10)

	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.

The screenshot shows the Hue interface with the 'airlines_new_parquet' database selected. The 'airports' table is currently active. A modal window displays the schema for the 'airports' table, which includes columns: iata, airport, city, state, country, lat, and lon. The 'info' icon (i) located in the top right corner of the table list is highlighted with a red box.

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

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

The screenshot shows the Hue interface with the 'airlines_new_parquet' database selected. The 'airports' table is currently active. A modal window displays the schema for the 'airports' table, which includes columns: iata, airport, city, state, country, lat, and lon. The table name 'airports' is highlighted with a red box.

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

The screenshot shows the Hue interface with the 'airlines_new_parquet' database selected. The 'airports' table is currently active. A modal window displays the schema for the 'airports' table, which includes columns: iata, airport, city, state, country, lat, and lon. The table name 'airports' is highlighted with a red box.

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;

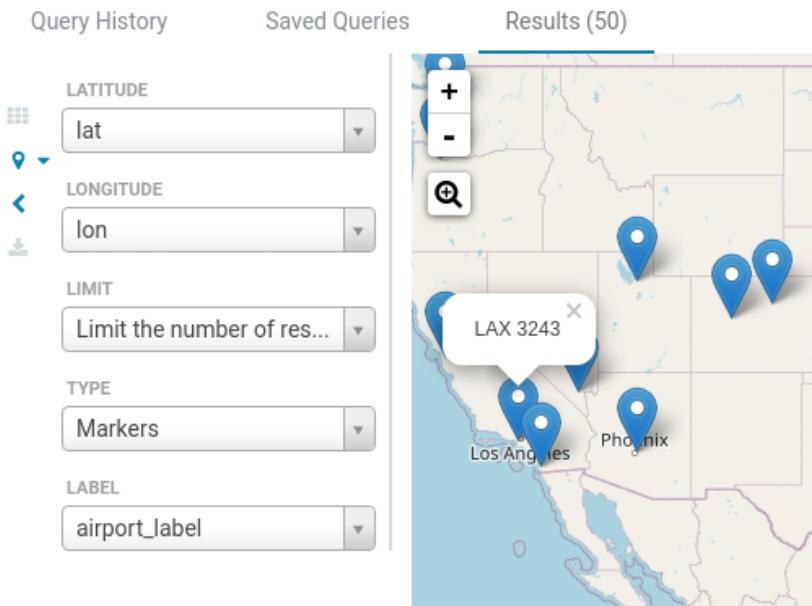
```

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

The screenshot shows a data visualization interface with a code editor at the top containing the provided SQL query. Below the code editor is a status bar indicating two queries are complete. The main area is a table titled 'Results (50)' showing flight data. To the left of the table is a vertical toolbar with several icons. A dropdown menu is open over the toolbar, listing five options: 'Bars', 'Pie', 'Scatter', 'Marker Map' (which is highlighted with a red box), and 'Gradient Map'. The 'Marker Map' option is the one intended for selection.

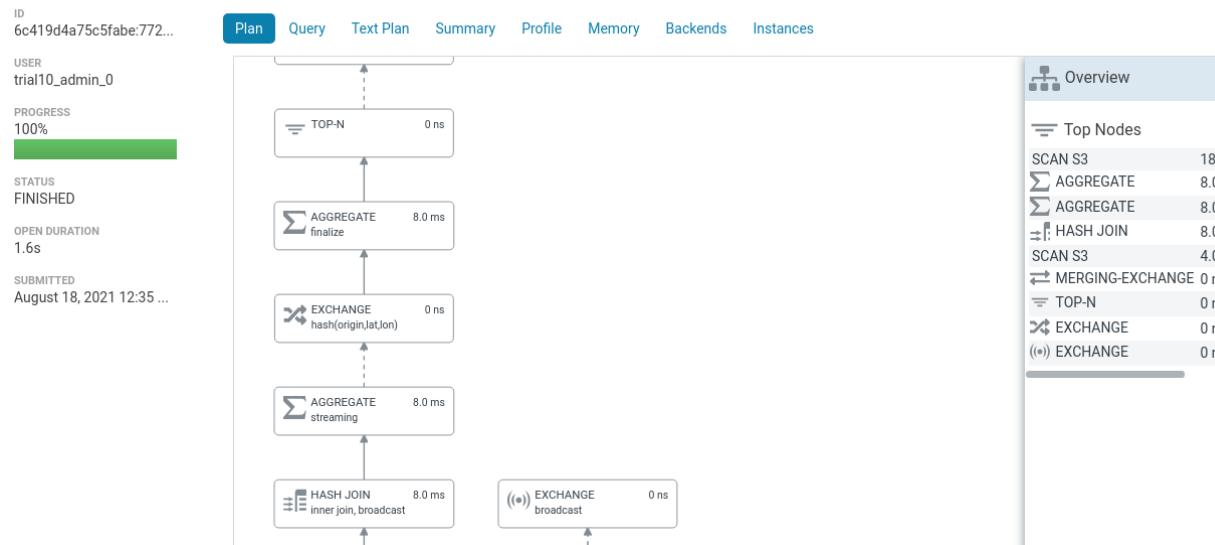
	origin	lat	lon	num_of_cancellations	airport_label
1	ORD	41.979595	-87.90446417	22123	ORD 22123
		32.89595056	-97.0372	16519	DFW 16519
		33.64044444	-84.42694444	15350	ATL 15350
		40.77724306	-73.87260917	10688	LGA 10688
		40.69249722	-74.16866056	8979	EWR 8979
		29.98047222	-95.33972222	8885	IAH 8885
		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 “airport_label” column



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...
```



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	Plan	Query	Text Plan	Summary	Profile	Memory	Backends	Instances						
USER				Operator	#Hosts	#Inst	Avg Time	Max Time	#Rows	Est.	#Rows	Peak Mem	Est. Peak Mem	Detail
trial10_admin_0				F03:ROOT	1	1	0.000ns	0.000ns		4.02	MB	4.00	MB	
PROGRESS	100%			08:MERGING-EXCHANGE	1	1	0.000ns	0.000ns	50	50	224.00	KB	28.22	KB UNPARTITIONED
STATUS	FINISHED			F02:EXCHANGE SENDER	2	14	0.000ns	0.000ns		3.45	KB	0		
OPEN DURATION	1.6s			04:TOP-N	2	14	0.000ns	0.000ns	318	50	12.00	KB	1.76	KB
SUBMITTED	August 18, 2021 12:35 A...			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		184.34	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	
				F05:EXCHANGE	2	2	0.000ns	0.000ns	3.38K	10.96K	248.00	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

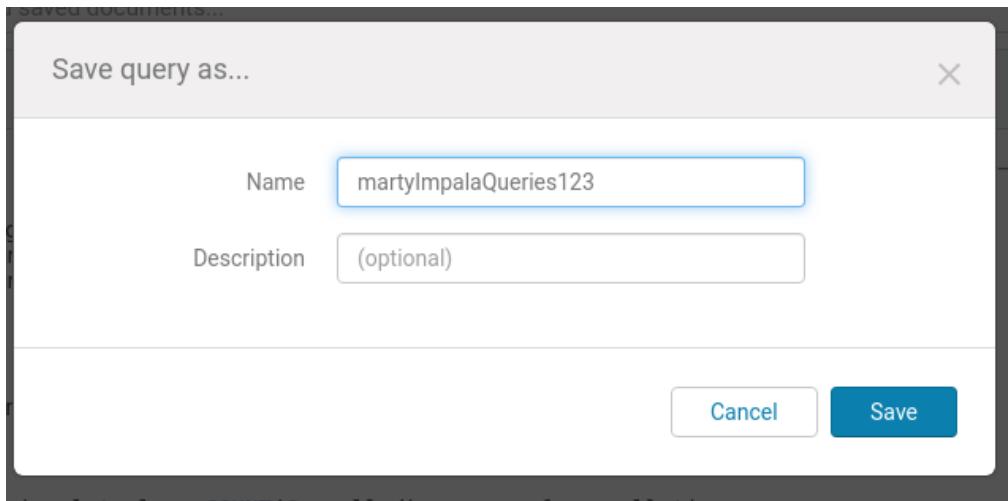
27) Time to save our work. Give your Impala session a name. Use something unique as this is a multi-user environment.

The screenshot shows the Impala interface. At the top, there is a search bar with the placeholder "Search data and saved documents...". Below the search bar, the connection name "Impala" is displayed next to a gear icon. To the right of the connection name is a button labeled "Add a name..." which is highlighted with a red rectangular border. Further to the right is a button labeled "Add a description...". The main area of the interface is a query editor containing the following SQL code:

```
5
6 SELECT origin,dest,COUNT(Diverted) as t FROM flights
7 WHERE Diverted = '1'
```

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

A screenshot of the Impala UI interface. At the top, there's a header bar with the 'Impala' logo, a search bar containing 'martyImpalaQueries123', and a button to 'Add a description...'. To the right of the search bar are three icons: a blue square with a white arrow (refresh), a blue square with a white document (save, highlighted with a red box), and a blue square with three vertical dots (more options). Below the header, the main area shows a query editor with a partially visible SQL statement and a results table. The results table has a red box around its top-left corner. On the far right of the results table, there are icons for a gear (settings), a question mark (help), and an 'x' (cancel). The bottom right corner of the screen shows a small 'x^2' icon.



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

A screenshot of the Data Visualization Lab interface. On the left, a code editor window displays a query: 20 WHERE origin = iata and cancelled = 1 AND cancellationcode = 'B'; 21 GROUP BY origin, lat, lon ORDER BY num_of_cancellations DESC LIMIT 50; 22;. Below the code editor, a message indicates two queries are complete. At the bottom, a navigation bar has tabs for "Query History", "Saved Queries" (which is highlighted with a red box), and "Results (50)". A table below lists the saved query: Name: martyImpalaQueries123, Description: (empty), Owner: trial10_admin_0, Last Modified: 08/17/2021 1:55 PM -04:00.

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

SKIP TO THE DATA VISUALIZATION LAB. DAS INSTRUCTIONS BELOW. DAS IS DEPRECATED SO I URGE YOU TO JUMP TO THE DATA VISUALIZATION... so I don't really have strong feelings about this...

- 9) Once your Virtual Warehouse is “Running”, click the line in the top right and then click “Open DAS”

Virtual Warehouses | 3

	testvirtualwarehouse1	mschoeni-iso-1
Running	compute-1611179792-vz49 cdptrialuser24-dl-default	Stopped compute-1611173596-dbly cdptrialuser24-dl-default
NODE COUNT	2	38
TOTAL CORES	38	292 GB
TOTAL MEMORY	292 GB	
TYPE	HIVE	DA

- 10) Enter the login information from step #1 above using the user, then click “LOGIN”
 *Changing “trialxx_admin” to the trail user you’re using and password defined by “X” in #1 above

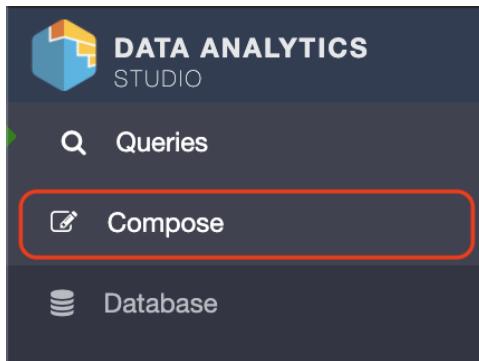
DATA ANALYTICS STUDIO

Username
trialxx_admin

Password

LOGIN

- 11) Click on “Compose”, to write the queries below to answer questions on the table “flights”



- 12) Choose the database “airlines_new_orc” that we found in Part 1 “Data Catalog”

The screenshot shows the Data Analytics Studio interface. The left sidebar has 'Queries', 'Compose' (which is selected), and 'Database'. The main area is titled 'Compose' with 'LAST UPDATE 6 sec ago'. It shows a list of databases: 'default', 'airlines_new_orc' (highlighted with a red box), 'airlines_new_parquet', 'retail_clickstream', 'default', and 'information_schema'. At the bottom are 'EXECUTE' and 'EXPLAIN' buttons.

- 13) Enter the following query in Worksheet1, answering the question “show me the top 5 visited destination by year from (1995-2008)”

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

The screenshot shows Worksheet1 in Data Analytics Studio. The query entered is:
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;
The entire query is highlighted with a red box. At the bottom are 'EXECUTE' and 'EXPLAIN' buttons.

14) Click “EXPLAIN” to see the visual 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

The screenshot shows a database interface with a query editor and an explain plan visualization.

Query Editor:

```
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;
```

Control Buttons:

EXECUTE EXPLAIN (The EXPLAIN button is highlighted with a red rectangle.)

Explain Plan Visualization:

```
graph LR; Fetch[Fetch] --> Limit[Limit  
Rows: 5]; Limit --> PS1["Partition/Sort  
Rows: 5.5m"]; PS1 --> TNK[Top N Key  
Rows: 5.5m]; TNK --> GB1[Group By  
Rows: 5.5m]; GB1 --> PS2["Partition/Sort  
Rows: 11m"]; PS2 --> GB2[Group By  
Rows: 11m]
```

The explain plan shows the following steps from left to right: Fetch, Limit (Rows: 5), Partition/Sort (Rows: 5.5m), Top N Key (Rows: 5.5m), Group By (Rows: 5.5m), Partition/Sort (Rows: 11m), and finally Group By (Rows: 11m).

- 15) Click “EXECUTE” to execute the query, answering the question “show me the top 5 visited destination by year from (1995-2008)”

The screenshot shows a database worksheet titled "Worksheet1*". At the top, there are buttons for "Saved", "Worksheet1*", a save icon, a close button, and a plus sign for adding a new worksheet. Below this is a code editor containing the following SQL query:

```
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;
```

At the bottom of the worksheet, there are two buttons: "EXECUTE" (highlighted with a red box) and "EXPLAIN".

- 16) Click the download button on the top right, to download the results as a CSV file

The screenshot shows a results table titled "Results". The table has three columns: DEST, YEAR, and TIMES_VISITED. The data is as follows:

DEST	YEAR	TIMES_VISITED
ATL	2005	429800
ATL	2004	416989
ATL	2008	414521
ATL	2007	413805
ATL	2006	404829

- 17) Going back to “Worksheet 1”, click the “+” to add another Worksheet for the next query

Saved ▾ Worksheet1*

```
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
```

EXECUTE **EXPLAIN**

18) In “Worksheet 2”, Choose the database “airlines_new_orc” then copy-and-paste the following query, answering the question “What are the top 10 routes (origin and dest) that have seen maximum diversions?”

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

Saved ▾ Worksheet1* Worksheet2*

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

EXECUTE **EXPLAIN**

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

The screenshot shows a user interface for executing SQL queries. At the top, there are two buttons: "EXECUTE" (highlighted with a red box) and "EXPLAIN". Below the buttons is a section titled "Results" which contains a table of flight data.

ORIGIN	DEST	T
ORD	LGA	845
LGA	DFW	749
DFW	LGA	653
DAL	HOU	615
ATL	LGA	567
MDW	STL	512
ATL	DFW	482
ORD	DFW	450
LAX	JFK	450
MIA	LGA	449

- 20) Going back to “Worksheet 2”, click the “+” to add another Worksheet for the final query

The screenshot shows a "Worksheet 2" tab open in a browser window. The tab bar includes "Saved", "Worksheet1*", "Worksheet2*", and a plus sign icon. The main area displays a SQL query:

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

21) In “Worksheet 3”, Choose the database “airlines_new_orc” then copy-and-paste the following query, answering the question “Which three months have seen the most number of cancellation due to bad weather?”

```
SELECT month, COUNT(Cancelled) as num_of_cancellations FROM flights
WHERE Cancelled = 1 AND CancellationCode = 'B'
GROUP BY month
ORDER BY num_of_cancellations DESC
LIMIT 3;
```

22) Click “EXECUTE” to execute the query, answering the question “Which three months have seen the most number of cancellation due to bad weather?”

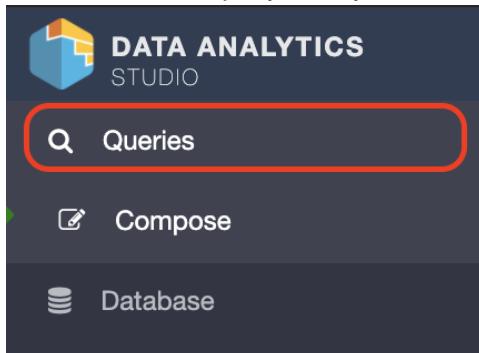
The screenshot shows a user interface for executing SQL queries. At the top, there are two buttons: "EXECUTE" (highlighted with a red box) and "EXPLAIN". Below them is a button labeled "Execute Query". The main area is titled "Results" and contains a table with the following data:

MONTH	NUM_OF_CANCELLATIONS
12	48868
1	42641
2	38234

22) Click “EXECUTE” a second time - this will lead us to our last portion of Part 2

23) Click on “Queries” on the top left navigation bar

*We'll look at our query history



24) Click the “Compare” on the right of your last query run (query at the top)

QUERIES (159)										COMPOSE QUERY
QUERY	STATUS	QUEUE	USER	TABLES READ	TABLES WRITTEN	START TIME	DURATION	DAG ID	ACTIONS	
SELECT month,COUNT(Cancelled) as n...	SUCCESS	None	trial24_admin	flights (airlines_new_o...	Not Available!	8 seconds ago	00:00:00	Not Available!		
SELECT month,COUNT(Cancelled) as n...	SUCCESS	None	trial24_admin	flights (airlines_new_o...	Not Available!	4 minutes ago	00:00:02	dag_161123645		
SELECT origin,dest,COUNT(Diverted) as...	SUCCESS	None	trial24_admin	flights (airlines_new_o...	Not Available!	8 minutes ago	00:00:03	dag_161123645		
SELECT dest,year,COUNT(dest) as Time...	SUCCESS	None	trial24_admin	flights (airlines_new_o...	Not Available!	20 minutes ago	00:00:00	Not Available!		
SELECT dest,year,COUNT(dest) as Time...	SUCCESS	None	trial24_admin	flights (airlines_new_o...	Not Available!	21 minutes ago	00:00:00	Not Available!		

25) Click the “Compare” on the right of the query (second to the top)

QUERIES (159)										COMPOSE QUERY
QUERY	STATUS	QUEUE	USER	TABLES READ	TABLES WRITTEN	START TIME	DURATION	DAG ID	ACTIONS	
SELECT month,COUNT(Cancelled) as n...	SUCCESS	None	trial24_admin	flights (airlines_new_o...	Not Available!	4 minutes ago	00:00:00	Not Available!		
SELECT month,COUNT(Cancelled) as n...	SUCCESS	None	trial24_admin	flights (airlines_new_o...	Not Available!	8 minutes ago	00:00:02	dag_161123645		
SELECT origin,dest,COUNT(Diverted) as...	SUCCESS	None	trial24_admin	flights (airlines_new_o...	Not Available!	13 minutes ago	00:00:03	dag_161123645		
SELECT dest,year,COUNT(dest) as Time...	SUCCESS	None	trial24_admin	flights (airlines_new_o...	Not Available!	24 minutes ago	00:00:00	Not Available!		
SELECT dest,year,COUNT(dest) as Time...	SUCCESS	None	trial24_admin	flights (airlines_new_o...	Not Available!	25 minutes ago	00:00:00	Not Available!		

26) Click on the “COMPARE” button to compare the two queries

Queries

```
SELECT month,COUNT(Cancelled) as num_of_cancellations ! x
```

```
SELECT month,COUNT(Cancelled) as num_of_cancellations ! x
```

COMPARE

[Compare two queries](#)

27) Notice the run duration is different between the two, let's find out why

Query Details - A		Query Details - B	
QUERY ID	hive_20210121153926_e3a56b9d-71f2-45dc-b23e-2c2e1146d61e	QUERY ID	hive_20210121153513_37aefec1-0284-4897-bfbb-bf9bb5797252
USER	trial24_admin	USER	trial24_admin
STATUS	SUCCESS	STATUS	SUCCESS
START TIME	21 Jan 2021 09:39:26	START TIME	21 Jan 2021 09:35:13
END TIME	21 Jan 2021 09:39:26	END TIME	21 Jan 2021 09:35:15
DURATION	118ms	DURATION	2s 311ms

28) Click on “timeline” at the top



As shown, the faster query only did “compile and parse”, while the slower query did “compile, parse, build dag, submit dag, submit to running, run dag”. Why? Because if you run the same exact query twice, the results are cached (if the data didn’t change). CDW knows if the data changed.



WELCOME TO THE DATA VISUALIZATION LAB

Part 3 - Data Visualization [25 minutes]

Overview: What is Data Visualization and how do we use it with our data?

Purpose: Create visualization using the flight information answering the question (visually with a density graph):

- What were the most number of flights from destination to origin between (1995-2008) - Route Density

- 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 #

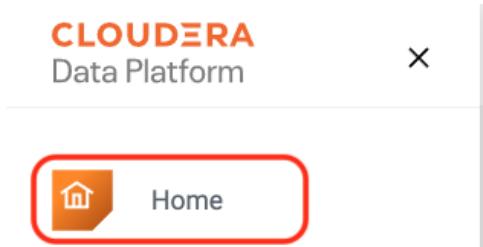
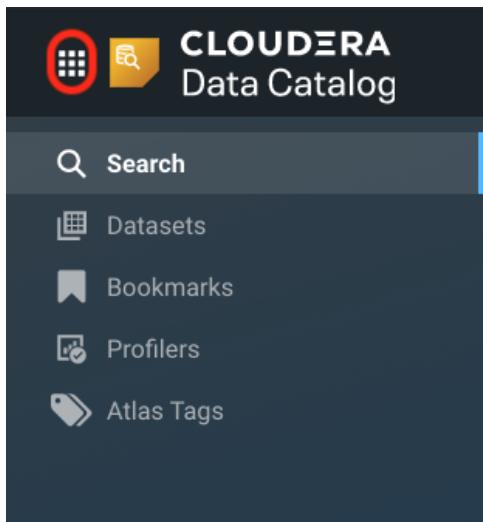
*X represents the password

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”



NOTES FOR NEW LAB: You will autologin, and when you get to create new dataset use the impalacdw and corresponding setting shown here:

New Dataset

Create a dataset from data on this connection. You need to create a dataset before you can create dashboards or apps.

Dataset title *

Dataset Source

From Table

Select Database

Select Table

CANCEL CREATE

3) Click “Open Data Visualization” on your existing “Running” Virtual Warehouse

Virtual Warehouses

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

mschoeni-iso-1
Stopped
compute-1611173596-dbtv
cdptrialuser24-dl-default

NODE COUNT	TOTAL CORES	TOTAL MEMORY	TYPE
2	38	292 GB	HIVE DA

NODE COUNT TOTAL CORES TOTAL MEMORY

Suspend
Clone
Edit
Delete
Upgrade
Copy JDBC URL
Download JDBC Jar
Open DAS
Open Data Visualization

- 4) Enter the login information from step #1 above using the user, then click “LOGIN”
*Changing “trialxx_admin” to the trail user you’re using and password defined by “X” in #1 above

CLOUDERA
Data Visualization

LOGIN

Username
trialxx_admin

Password
[Redacted]

Invalid login

[Forgot your password?](#)

Remember me on this computer

LOGIN

- 5) Click “DATA” the top navigation bar

The screenshot shows the Cloudera Data Visualization web application. At the top, there is a navigation bar with links for HOME, VISUALS, and DATA. The DATA link is highlighted with a red circle. To the right of the navigation is a search bar with placeholder text "find titles, viz types, datasets, authors...". Below the navigation, on the left, is a sidebar titled "All Connections" which lists "Default Hive VW" and "samples". In the main content area, there is a "Datasets" section with a count of 11, followed by a table header for "Title/Table", "Created", and "Last Modified".

6) Click “Default Hive VW” to add our dataset

This screenshot shows the same interface as above, but the "Default Hive VW" connection in the sidebar has been selected, indicated by a red circle around its row. The main content area remains the same, showing the Datasets section and the table headers.

7) Click “NEW DATASET” to add our “flights” data

This screenshot shows the interface again with the "Default Hive VW" connection selected. The "NEW DATASET" button in the top right of the main content area is highlighted with a red circle. The rest of the interface, including the sidebar and the table headers, remains consistent.

8) Enter a name for the Dataset title naming “airline_new_orc.flights”

*Can be any name you choose

New Dataset

Create a dataset from data on this connection. You need to create a dataset before you can create dashboards or apps.

Dataset title *

airlines_new_orc.flights

Dataset Source

From Table

Select Database

airlines_new_orc

Select Table

flights

CANCEL CREATE

9) Choose the database “airlines_new_orc”

New Dataset

Create a dataset from data on this connection. You need to create a dataset before you can create dashboards or apps.

Dataset title *

airlines_new_orc.flights

Dataset Source

From Table

Select Database

airlines_new_orc

Select Table

flights

CANCEL CREATE

10) Choose the table “flights”

*Need to import multiple databases and tables? You'd use Dataset Source = SQL

New Dataset

Create a dataset from data on this connection. You need to create a dataset before you can create dashboards or apps.

Dataset title *

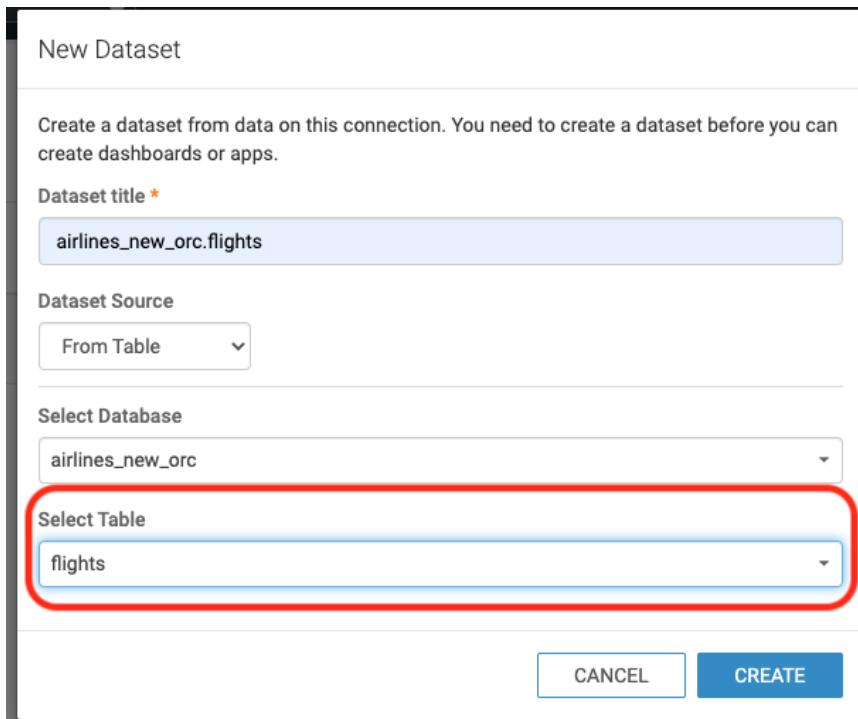
Dataset Source

From Table

Select Database

Select Table

CANCEL CREATE



11) Click “CREATE”

New Dataset

Create a dataset from data on this connection. You need to create a dataset before you can create dashboards or apps.

Dataset title *

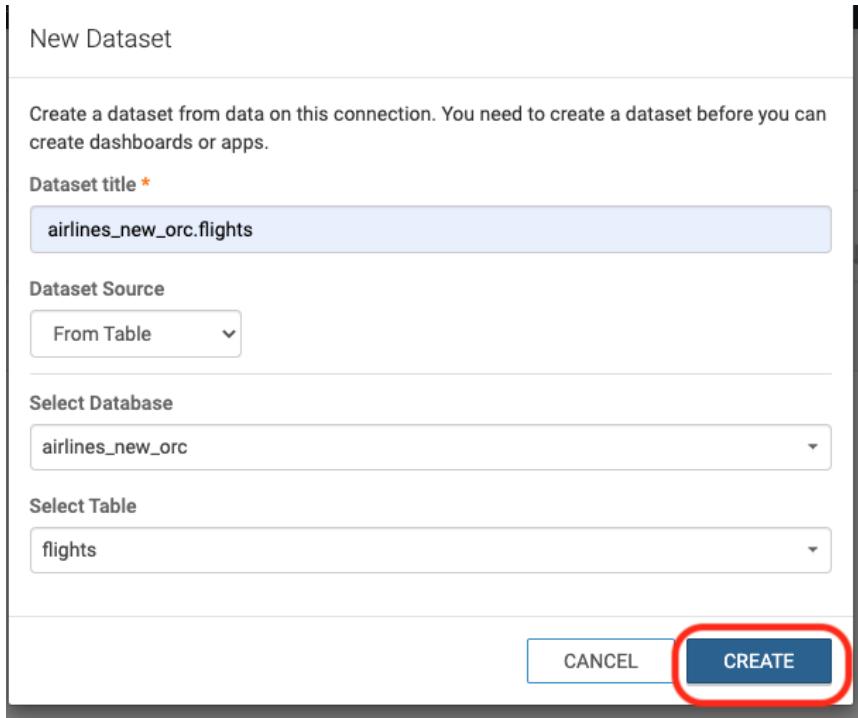
Dataset Source

From Table

Select Database

Select Table

CANCEL CREATE



12) Click “+” to create a New Dashboard

The screenshot shows the Data Studio interface. At the top, there are buttons for 'NEW DATASET' and 'ADD DATA'. Below that, a navigation bar includes 'Datasets' (with a red circle around it), 'Connection Explorer', and other options. A table lists datasets: 'airlines_new.orc.flights' (selected) and 'airlines_new.orc.flights'. The table columns include 'Title/Table', 'Created', 'Last Updated', 'Modified By', '# Visuals', and a delete icon.

13) Choose “Treemap” under “VISUALS”

The screenshot shows the 'Dashboard Designer' interface. On the left, a sidebar titled 'VISUALS' has a red circle around the 'Treemap' icon. The main area shows a 'DATA' panel with a dataset 'airlines_new.orc.flights' and a 'Dimensions' section containing fields: flights, uniquecarrier, tailnum, origin, dest, cancellationcode, and diverted. Below that is a 'Measures' section with 24 items including Record Count, month, dayofmonth, dayofweek, deptime, crsdeptime, arrtime, crsarrrtime, flightnum, and actualelapsedtime. At the bottom left, there's a 'REFRESH VISUAL' button.

14) Drag-and-drop both “dest” and “origin” from Dimensions->Flights into Dimensions under Visuals

Dashboard Designer

VISUALS

- Table
- 1234 LABEL
- WORD
- PIVOT
- CHART
- MAP
- ACTION
- SQL

Dimensions

- dest
- origin

Measures

drag fields to add here

Tooltips

drag fields to add here

Filters

drag fields to add here

Limit: 100

REFRESH VISUAL

DATA

airlines_new_orc.flights

Sample Mode: OFF

Search

Dimensions 6

- flights
- uniquecarrier
- tailnum
- origin
- dest
- cancellationcode
- diverted

Measures 24

- flights
- Record Count
- month
- dayofmonth
- dayofweek
- deptime
- crsdeptime
- arrtime
- crsarrtime
- flightnum

DASH.

Visuals

Filters

Settings

Style

BUILD

Settings

Style

Favorites

- AirDrop
- Recents
- Application:
- Desktop
- Documents
- Downloads

Locations

- Network

Tags

- Red
- Orange
- Yellow
- Green
- Blue

15) Drag-and-drop “Record Count” from Measures->Flights into Measures under Visuals

Dashboard Designer

VISUALS

 Table

 1234     

DATA

 airlines_new_orc.flights  

Sample Mode: OFF

Dimensions

- flights
 - A uniquecarrier
 - A tailnum
 - A origin
 - A dest
 - A cancellationcode
 - A diverted

Measures

- flights
 - # Record Count
 - # month
 - # dayofmonth
 - # dayofweek
 - # deptime
 - # crsdeptime
 - # arrtime
 - # crsarrrtime
 - # flightnum
 - # actualelapsedtime
 - # crselapsedtime
 - # airtime
 - # arrdelay

DASH.
Visuals
Filters
Settings

Style
Build
Settings
Style

Favorites
Recent
Applications
Desktop
Documents
Downloads

Locations
Network

Tags
Red
Orange
Yellow
Green
Blue

16) Click the right arrow next to Record Count and select “Descending” under Order and Top K

The screenshot shows the Tableau Dashboard Designer interface. On the left, there's a sidebar with various visualizations like Treemap, 1234 LABEL, WORD, and SQL. Below that are sections for Dimensions (dest, origin) and Measures (Record Count). There are also sections for Tooltips, X Trellis, Y Trellis, and Filters, each with a "drag fields to add here" placeholder. At the bottom is a blue "REFRESH VISUAL" button. The main area is titled "FIELD PROPERTIES" and contains a tree view of properties. A red box highlights the "Order and Top K" section under "Change Type". This section has two options: "Descending" (selected) and "Ascending". Under "Descending", there are input fields for "Top K:" (eg. 100) and "Bottom K:" (eg. 100). A note below says "Top K/Bottom K applies to granular dimensions". To the right of the properties panel, there are tabs for DASH., VISUAL, and other settings.

17) Click "REFRESH VISUAL"

*Notice - you can have other Visuals chosen to be displayed with the Dimensions and Measure(s), then click REFRESH VISUALS

Dashboard Designer

VISUALS	DATA	DASH.
Treemap	airlines_new_orc.flights Edit Delete	+ Visuals
	Sample Mode: OFF	+ Filters
Search	Dimensions 6	Settings
	flights	
	uniquecarrier	
	tailnum	
	origin	
	dest	
	cancellationcode	
	diverted	
	Measures 24	Style
	flights	
	# Record Count	Build
	# month	
	# dayofmonth	
	# dayofweek	
	# deptime	
	# crsdeptime	
	# arrtime	
	# crsarrrtime	
	# flightnum	
	# actualelapsedtime	
	# crselapsedtime	
	# airtime	
	# arrdelay	

* Dimensions

dest
origin

* Measure

Record Count

Tooltips

drag fields to add here

X Trellis

drag fields to add here

Y Trellis

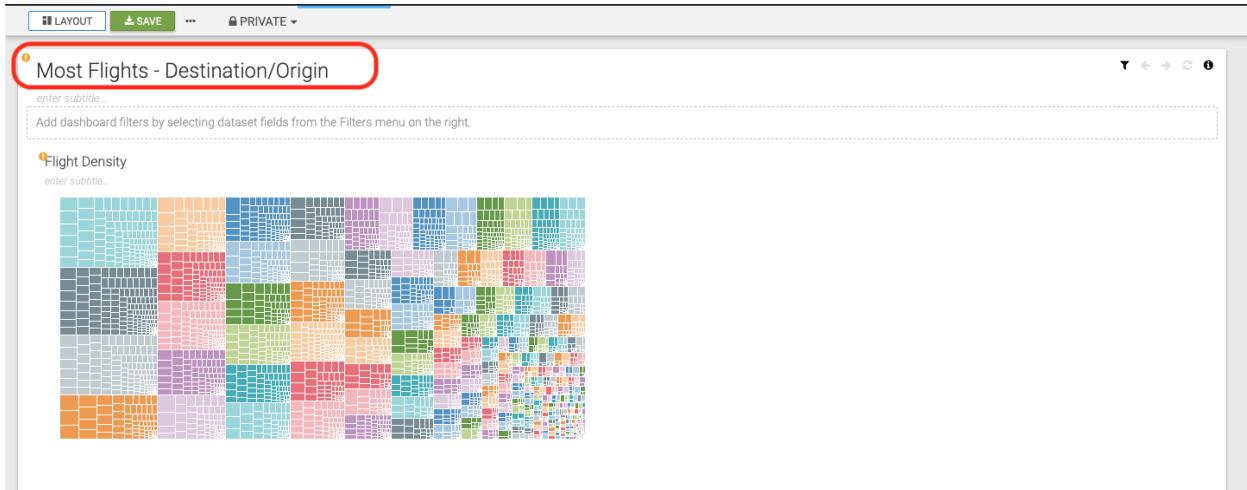
drag fields to add here

Filters

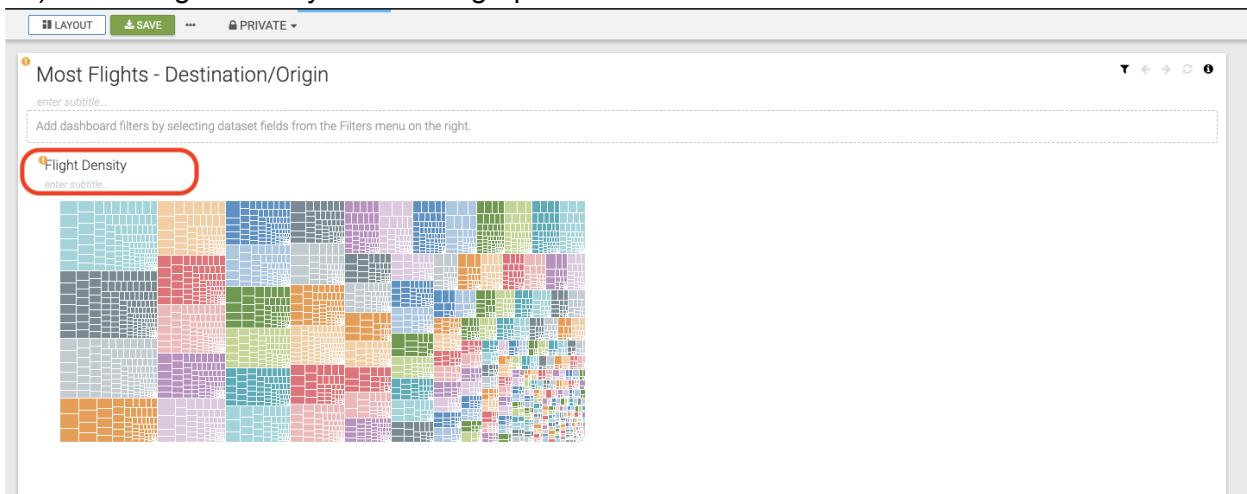
drag fields to add here

REFRESH VISUAL

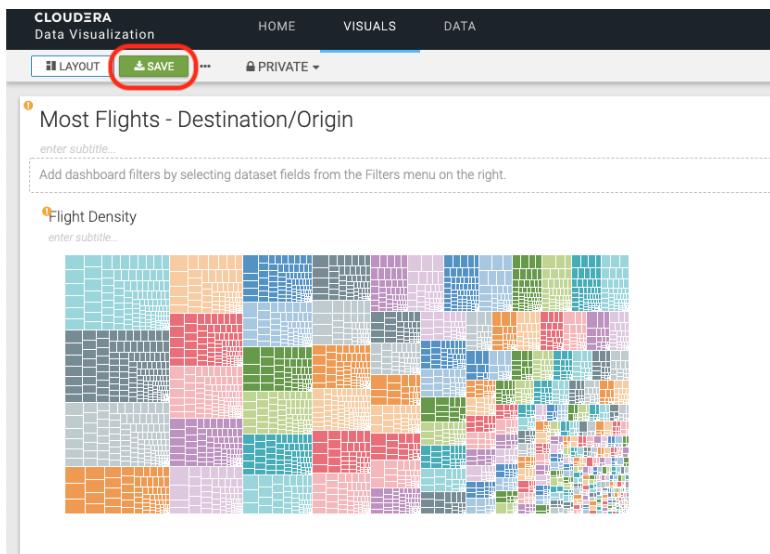
18) Enter a title “Most Flights - Destination/Origin”



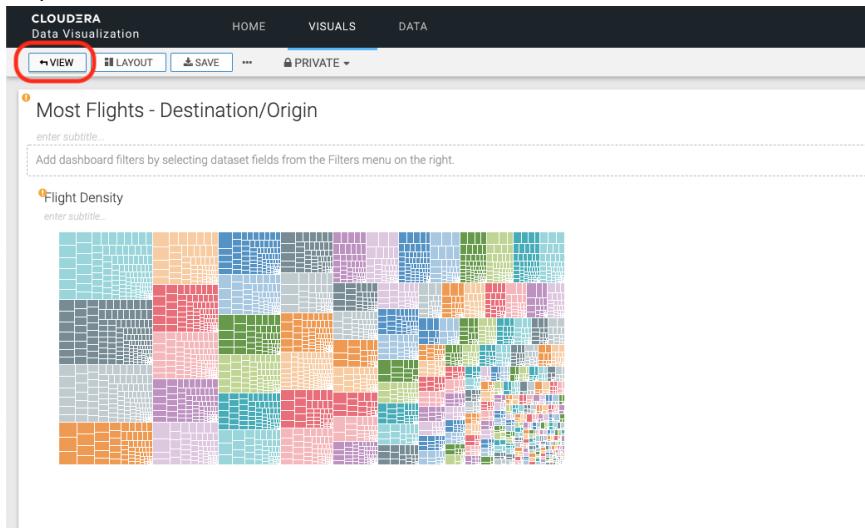
19) Enter “Flight Density” under the graph’s title



20) Click “SAVE”

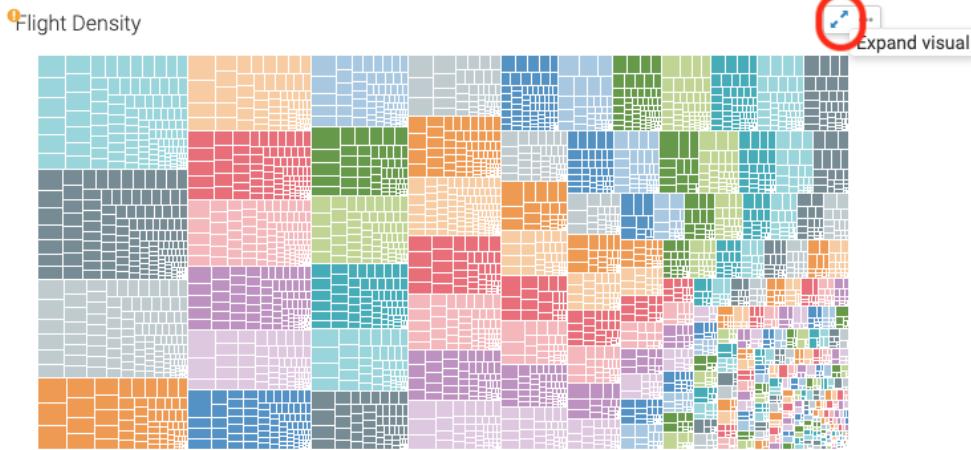


21) Click “VIEW”



22) Scroll over the graph and click “Expand Visual”

Most Flights - Destination/Origin



Destinations are displayed



OPTIONAL DATA IMPORT LAB USING DAS. YOU WILL DO THIS DATA IMPORT IN SPARK TOMORROW USING THE CLOUDERA MACHINE LEARNING.

Part 4 - Import a File into a Table [15 minutes]

Overview: How do we import data (csv file), creating a table?

- 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 #

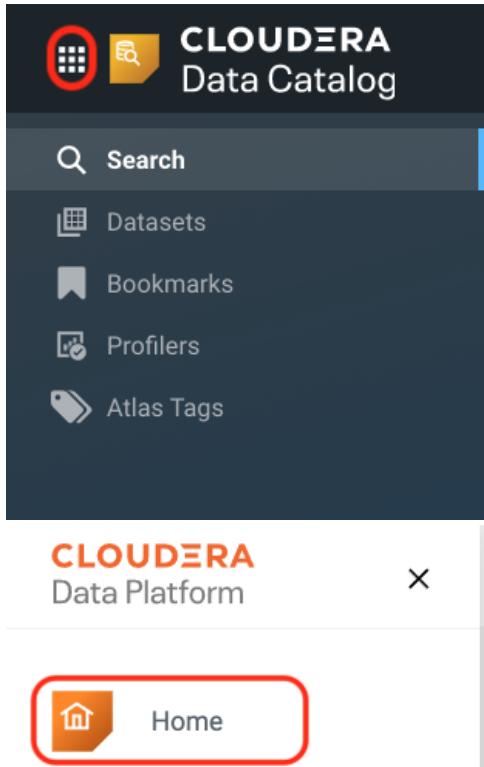
*X represents the password

- 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”



- 3) Click “Open DAS” on your existing “Running” Virtual Warehouse
*The same steps you did in Part 2 to Open DAS

The screenshot shows the Cloudera Data Platform (CDP) interface for managing virtual warehouses. The 'Virtual Warehouses' section lists three entries:

- testvirtualwarehouse1**: Status: Running, compute-1611179792-vz49, cdptrialuser24-dl-default. It has 2 nodes, 38 cores, and 292 GB of memory.
- mschoeni-iso-1**: Status: Stopped, compute-1611173596-dbtv, cdptrialuser24-dl-default. It has 0 nodes, 12 cores, and 56 GB of memory.
- default-vw**: Status: Stopped, compute-1611103491-4hbp. It has 0 nodes, 0 cores, and 0 GB of memory.

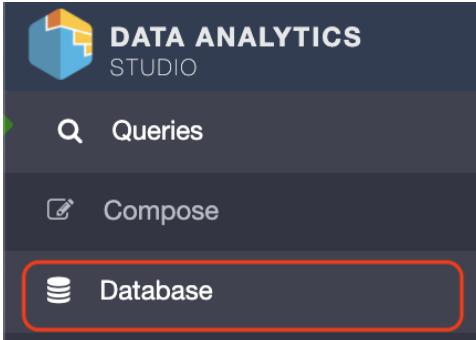
A context menu is open over the first entry, showing the following options:

- Suspend
- Clone
- Edit
- Delete
- Upgrade
- Copy JDBC URL
- Download JDBC Jar
- Open DAS** (highlighted with a red box)
- Open Data Visualization
- Set Compactor
- Run AutoScaling Demo
- Collect Diagnostic Bundle

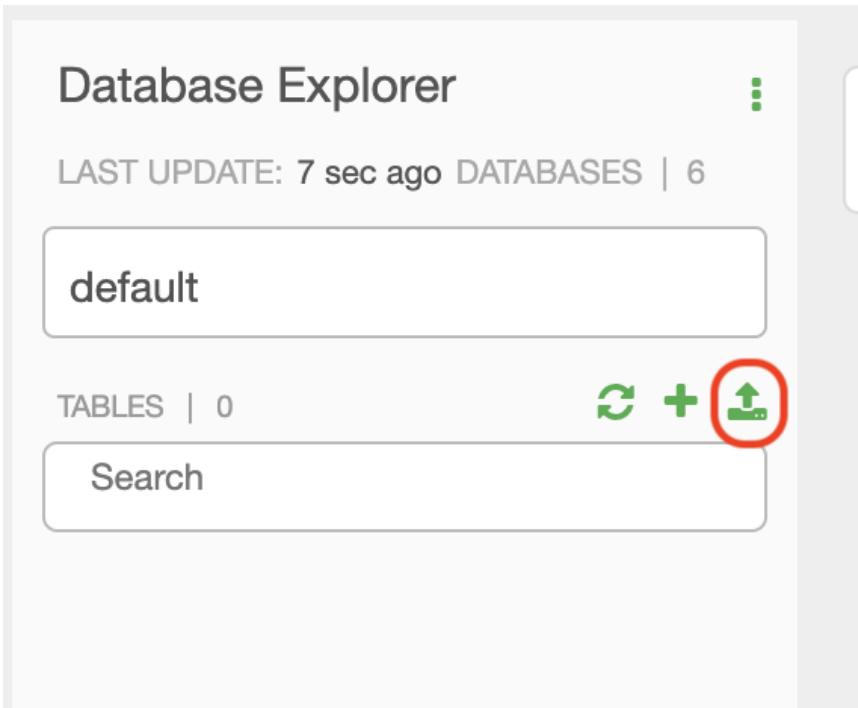
- 4) Enter the login information from step #1 above using the user, then click “LOGIN”
*You’ll likely already be authenticated from Part 2, you may not need to enter credentials
*Changing “trialxx_admin” to the trail user you’re using and password defined by “X” in #1 above

The screenshot shows the login screen for the Data Analytics Studio. It features a logo and the text "DATA ANALYTICS STUDIO". The form includes fields for "Username" (containing "trialxx_admin") and "Password" (an empty field). A large green "LOGIN" button is at the bottom, with a red circle highlighting it.

5) Click on Database on the left navigation bar



6) Click on “Upload Table”, using the “default” database



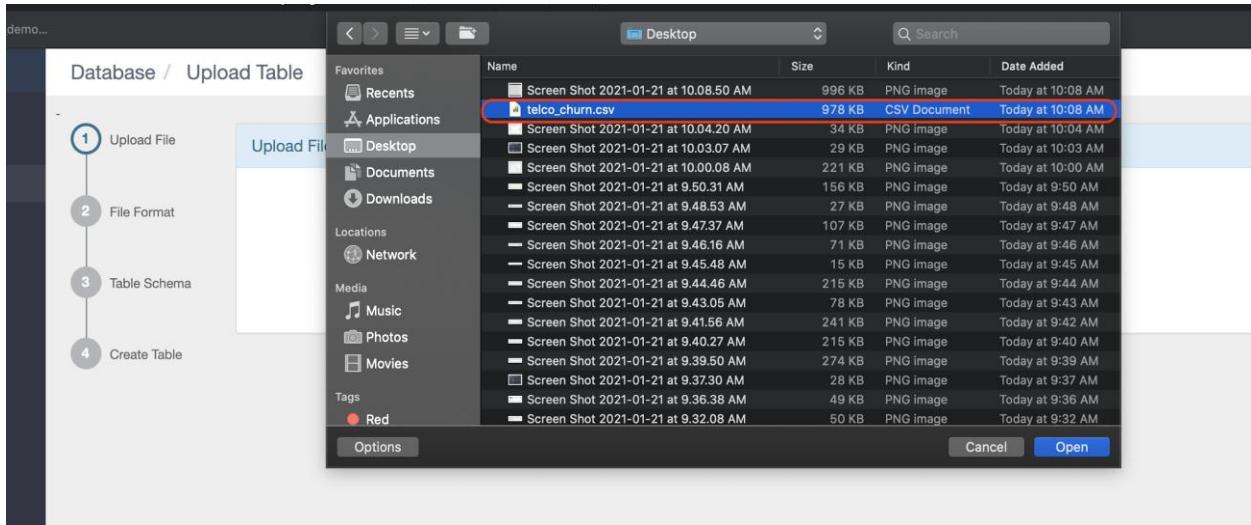
7) In a new browser window or tab, download the CSV file, saving to your desktop as “telco_churn.csv”

https://raw.githubusercontent.com/andy-hansen/cdp/master/cml/raw/WA_Fn-UseC_-Telco-Customer-Churn-.csv

The screenshot shows a web browser window with the following details:

- Address Bar:** raw.githubusercontent.com/andy-hansen/cdp/master/cml/raw/WA_Fn-UseC_-Telco-Customer-Churn-.csv
- Tab:** Apps - hms-mirror_demo...
- Content Area:** Displays the raw CSV data for the 'telco_churn.csv' file. The data consists of approximately 700 rows of customer information, including fields such as customerID, gender, SeniorCitizen, Partner, Dependents, tenure, PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, MonthlyCharges, TotalCharges, and Churn.
- Bottom Navigation:** Shows a file icon and the filename 'telco_churn.csv'.

8) Going back to the window from step 6 above, upload the file "telco_churn.csv"



9) Click the “Is first row header?”, since the first row is a header

File type	CSV	Clear
Field Delimiter	,	Clear
Escape Character	\	Clear
Quote Character	"	Clear
<input checked="" type="checkbox"/> Is first row header?		
<input type="checkbox"/> Contains endlines?		
PREVIEW		

10) Click “PREVIEW” prior to creating the table

PREVIEW

Table Preview

CUSTOMERID	GENDER	SENIORCITIZEN	PARTNER	DEPENDENTS	TENURE	PHONESERVICE	MULTIPLELINES	INTERNETSERVICE	ONLINESECURITY	ONLINEBACKUP	DEVICEPROTECTION	TECHSUPPORT	STREAMINGTV	STREAMINGMOVIES	CONTR/
7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes	No	No	No	No	Month-month
5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes	No	No	No	One year
3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	No	No	No	Month-month
7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	No	Yes	Yes	No	No	One year
9237-	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No	No	No	No	Month-

BACK **NEXT →** **X CANCEL**

11) Click “NEXT”

PREVIEW

Table Preview

CUSTOMERID	GENDER	SENIORCITIZEN	PARTNER	DEPENDENTS	TENURE	PHONESERVICE	MULTIPLELINES	INTERNETSERVICE	ONLINESECURITY	ONLINEBACKUP	DEVICEPROTECTION	TECHSUPPORT	STREAMINGTV	STREAMINGMOVIES	CONTR/
7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes	No	No	No	No	Month-month
5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes	No	No	No	One year
3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	No	No	No	Month-month
7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	No	Yes	Yes	No	No	One year
9237-	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No	No	No	No	Month-

BACK **NEXT →** **X CANCEL**

12) Enter ‘telco_churn’ as the Table Name. Click “CREATE”

Database / Upload Table

Table Name: telco_churn

COLUMNS ADVANCED TABLE PROPERTIES

COLUMN NAME	DATA TYPE	SIZE	ADVANCED	ACTION
customerID	STRING		<input type="checkbox"/> Allow complex datatypes	DELETE
gender	STRING		<input type="checkbox"/> Allow complex datatypes	DELETE
SeniorCitizen	INT		<input type="checkbox"/> Allow complex datatypes	DELETE
Partner	STRING		<input type="checkbox"/> Allow complex datatypes	DELETE
Dependents	STRING		<input type="checkbox"/> Allow complex datatypes	DELETE
tenure	INT		<input type="checkbox"/> Allow complex datatypes	DELETE
PhoneService	STRING		<input type="checkbox"/> Allow complex datatypes	DELETE

BACK + CREATE CANCEL

13) Wait for about 2 minutes then Go-to “Compose” and within “Worksheet 1” run the following query on the new table

```
select * from telco_churn limit 10;
```

Compose

Saved - Worksheet1

DATABASES | 6

default

TABLES | 1

Search Tables

= telco_churn (21)

EXECUTE EXPLAIN

Results

TELCO_CHURN.CUSTOMERID	TELCO_CHURN.GENDER	TELCO_CHURN.SENIORCITIZEN	TELCO_CHURN.PARTNER	TELCO_CHURN.DEPENDENTS	TELCO_CHURN.TENURE	TELCO_CHURN.PHONESERVICE	TELCO_CHURN.MULTIPLELINES	TELCO_CHURN.INTERNET
7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL
5575-QNVDIE	Male	0	No	No	34	Yes	No	DSL
3668-QPYBK	Male	0	No	No	2	Yes	No	DSL

Parking lot items

- [Show Impala/Hue - for CDH customers](#)
- [Add - Result Cache explanation](#)
- [Import CML github csv file](#)