Spark Example on Quickstart VM - Using Real World Sample

This document is to run Spark on CDH.

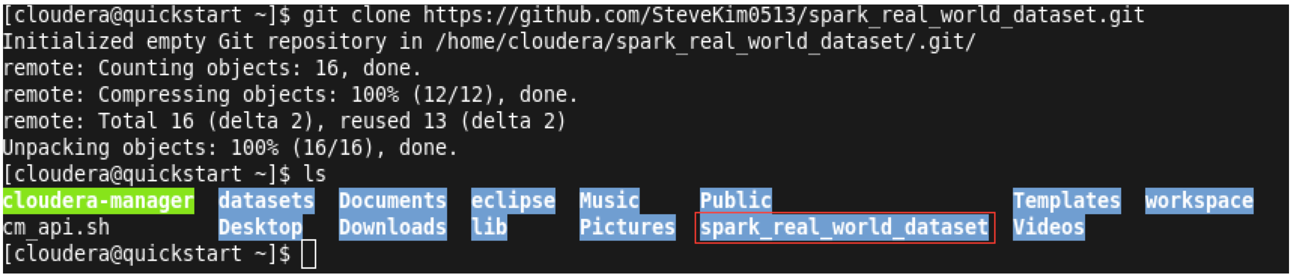
Apache Spark has been receiving a lot of deserved attention lately. It is very understandable given the huge importance of distributed data processing for many companies and the pursuit for faster, cheaper and easier to use technologies aiming replace or complement the widely adopted Hadoop ecosystem and its MapReduce paradigm.

You need to some preparation to run this example.

1. VMware Download(http://www.vmware.com/kr) and Install
2. Quickstart tutorial Download(http://www.cloudera.com/content/cloudera/en/downloads.html)
3. Open the Quickstart tutorial file(/cloudera-quickstart-vm-5.3.0-0-vmware/cloudera-quickstart-vm-5.3.0-0-vmware.vmx) on VMware.
4. Download the dataset on Github.*

* Download dataset for running spark example.
Open the terminal on virtual machine of QuickStart tutorial. And input command.

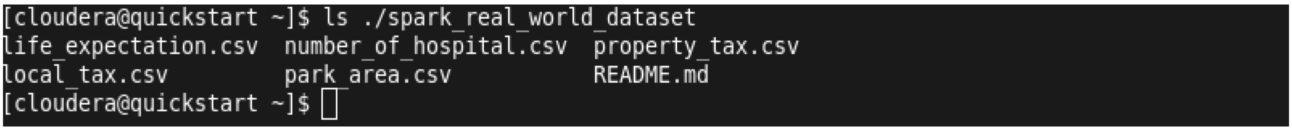
```
[cloudera@quickstart ~] git clone https://github.com/SteveKim0513/spark_real_world_dataset.git
```



```
[cloudera@quickstart ~]$ git clone https://github.com/SteveKim0513/spark_real_world_dataset.git
Initialized empty Git repository in /home/cloudera/spark_real_world_dataset/.git/
remote: Counting objects: 16, done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 16 (delta 2), reused 13 (delta 2)
Unpacking objects: 100% (16/16), done.
[cloudera@quickstart ~]$ ls
cloudera-manager  datasets  Documents  eclipse  Music     Public                    Templates  workspace
cm_api.sh         Desktop   Downloads  lib      Pictures  spark_real_world_dataset  Videos
[cloudera@quickstart ~]$ 
```

You can see the new folder named spark_real_world_dataset

```
[cloudera@quickstart ~] ls ./spark_real_world_dataset
```



```
[cloudera@quickstart ~]$ ls ./spark_real_world_dataset
life_expectation.csv  number_of_hospital.csv  property_tax.csv
local_tax.csv         park_area.csv           README.md
[cloudera@quickstart ~]$ 
```

These datasets are downloaded from “http://data.seoul.go.kr”.

< Scenario >

Life expectancy is the most commonly used measure to describe population health. Life expectancy measures how long, on average, a person is expected to live based on current age and sex-specific death rates. It is often expressed as the number of years of life a person born today is expected to live.

The starting point for calculating life expectancy is the age-specific death rates of the population members. If a large amount of data is available, the age-specific death rates can be simply taken as the mortality rates actually experienced at each age.

But life expectancy is not a simple. There are various factors to affect the result. For example, medical treatment level, dietary life, crime rate, etc. So I wonder that how well the value of life expectancy describes real world.

<Assumption>

Wealth, the condition of medical treatment, surrounding environment(e,g. green area ratio) may affect the result of life expectancy.

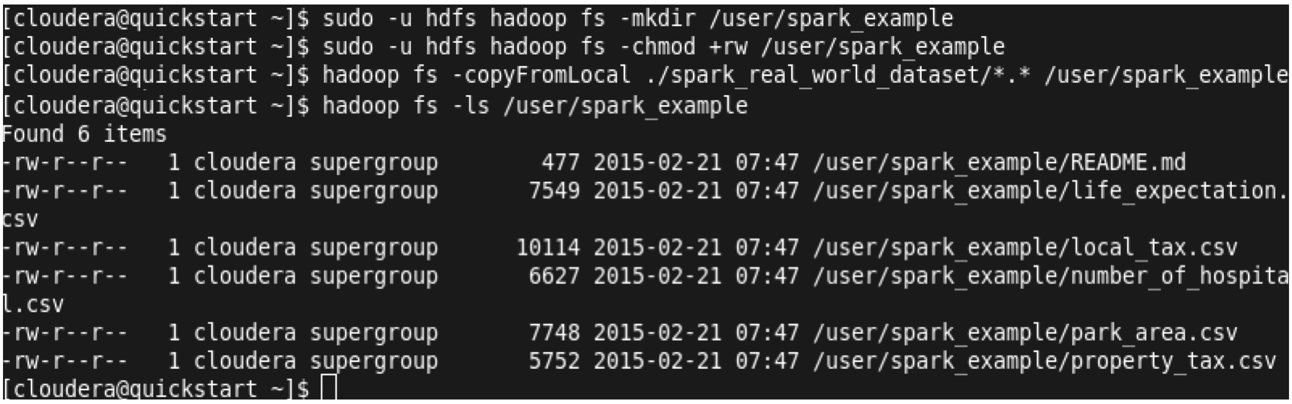
<How to measure>

Analyze between independent variables and the dependent variable using linear regression on Spark.

<Implementation>

Step 1. Upload the data to HDFS

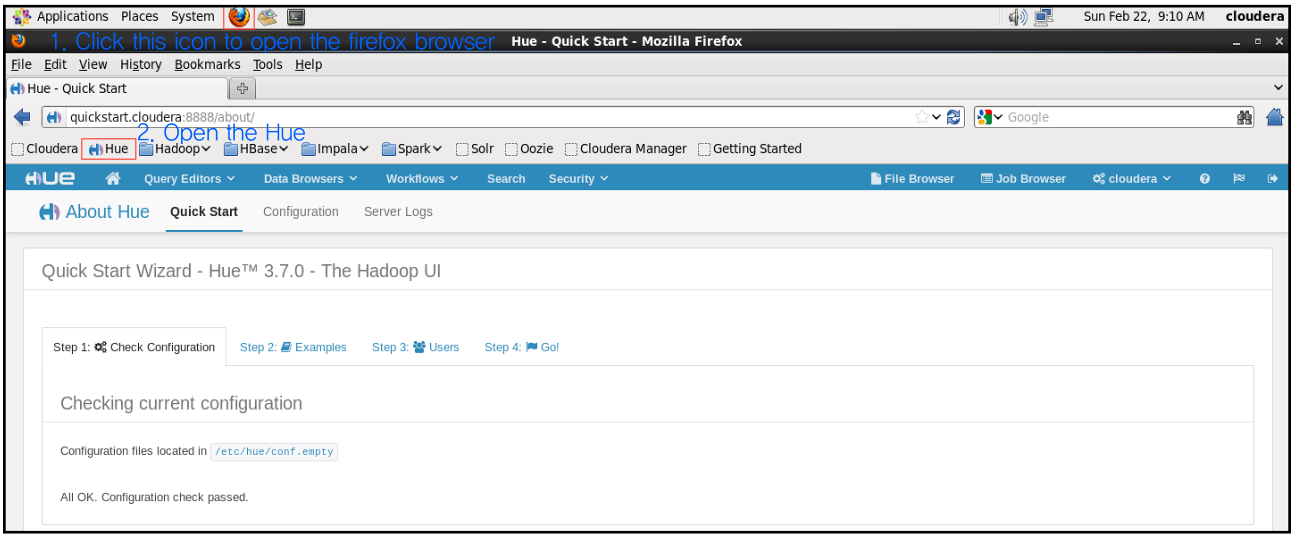
```
//make the folder at hadoop file system
[cloudera@quickstart ~] sudo -u hdfs hadoop fs -mkdir /user/spark_example

//change the authority of this folder.
[cloudera@quickstart ~] sudo -u hdfs hadoop fs -chmod +rw /user/spark_example

//copy the local dataset to hadoop file system
[cloudera@quickstart ~] hadoop fs -copyFromLocal ./spark_real_world_dataset/ /spark_example

//check the files
[cloudera@quickstart ~] hadoop fs -ls /user/spark_example
```



```
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -mkdir /user/spark_example
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -chmod +rw /user/spark_example
[cloudera@quickstart ~]$ hadoop fs -copyFromLocal ./spark_real_world_dataset/*.* /user/spark_example
[cloudera@quickstart ~]$ hadoop fs -ls /user/spark_example
Found 6 items
-rw-r--r--   1 cloudera supergroup        477 2015-02-21 07:47 /user/spark_example/README.md
-rw-r--r--   1 cloudera supergroup       7549 2015-02-21 07:47 /user/spark_example/life_expectation.
csv
-rw-r--r--   1 cloudera supergroup      10114 2015-02-21 07:47 /user/spark_example/local_tax.csv
-rw-r--r--   1 cloudera supergroup       6627 2015-02-21 07:47 /user/spark_example/number_of_hospita
l.csv
-rw-r--r--   1 cloudera supergroup       7748 2015-02-21 07:47 /user/spark_example/park_area.csv
-rw-r--r--   1 cloudera supergroup       5752 2015-02-21 07:47 /user/spark_example/property_tax.csv
[cloudera@quickstart ~]$ 
```

Step 2. Refine the data using Hive

1) Open the Hue to run Hive on GUI environment



1. Click this icon to open the firefox browser

2. Open the Hue

Applications Places System
Sun Feb 22, 9:10 AM cloudera
Hue - Quick Start - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Hue - Quick Start
quickstart.cloudera:8888/about/
Google
Cloudera Hue Hadoop HBase Impala Spark Solr Oozie Cloudera Manager Getting Started
HUE
Query Editors
Data Browsers
Workflows
Search
Security
File Browser
Job Browser
cloudera
About Hue
Quick Start
Configuration
Server Logs
Step 1: Check Configuration
Step 2: Examples
Step 3: Users
Step 4: Go!

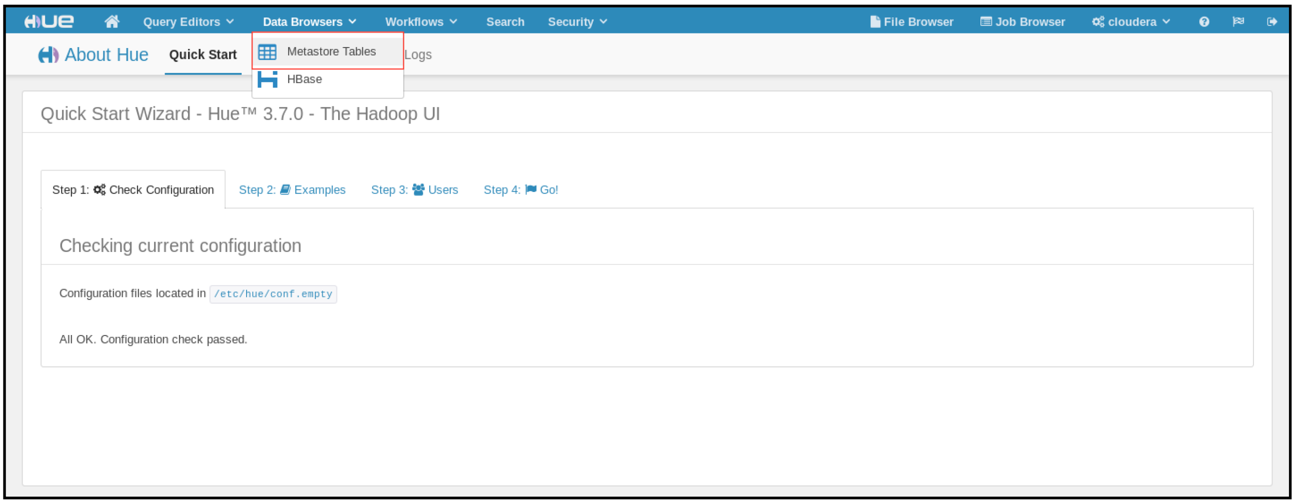
Quick Start Wizard - Hue™ 3.7.0 - The Hadoop UI

Checking current configuration

Configuration files located in `/etc/hue/conf.empty`

All OK. Configuration check passed.

2) Click the meatball to create the table



Quick Start Wizard - Hue™ 3.7.0 - The Hadoop UI

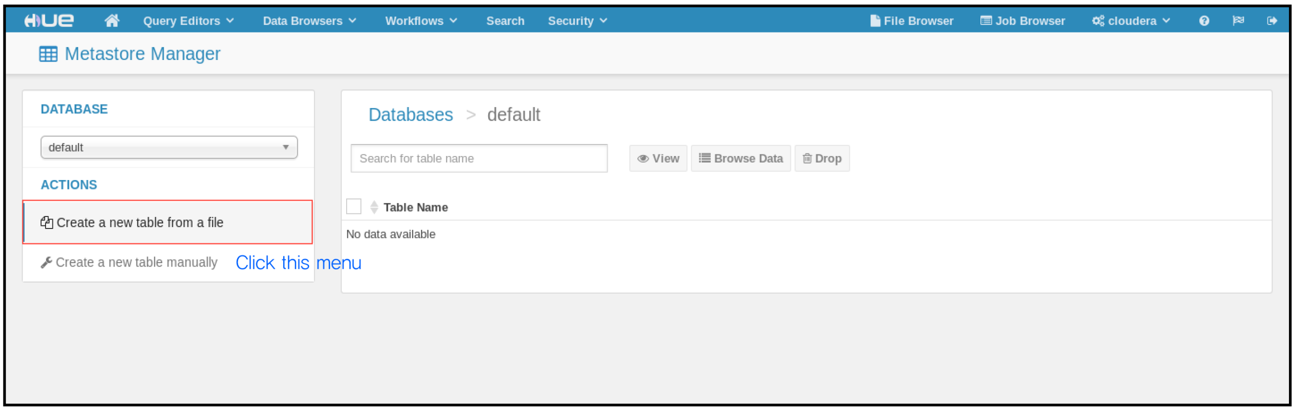
Step 1: Check Configuration | Step 2: Examples | Step 3: Users | Step 4: Go!

Checking current configuration

Configuration files located in `/etc/hue/conf.empty`

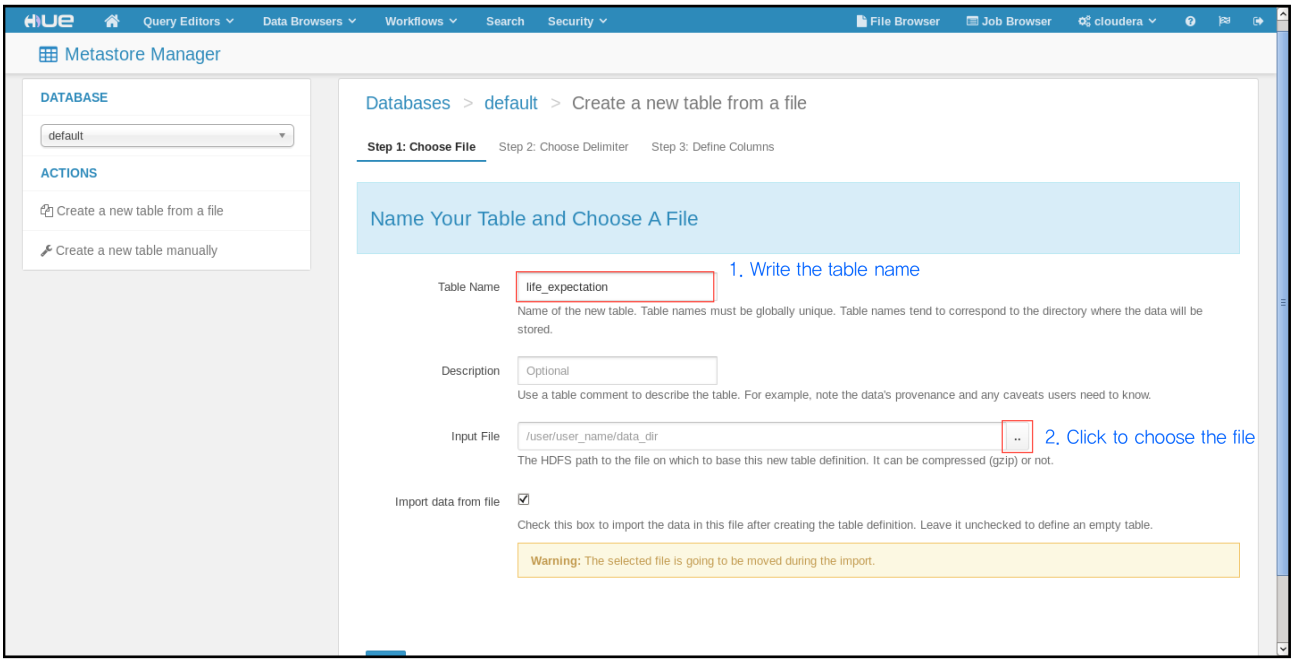
All OK. Configuration check passed.

3) Create the table from a data file(life_expectation)



No data available

Click this menu



Databases > default > Create a new table from a file

Step 1: Choose File | Step 2: Choose Delimiter | Step 3: Define Columns

Name Your Table and Choose A File

Table Name: life_expectation

1. Write the table name

Name of the new table. Table names must be globally unique. Table names tend to correspond to the directory where the data will be stored.

Description: Optional

Use a table comment to describe the table. For example, note the data's provenance and any caveats users need to know.

Input File: /user/user_name/data_dir

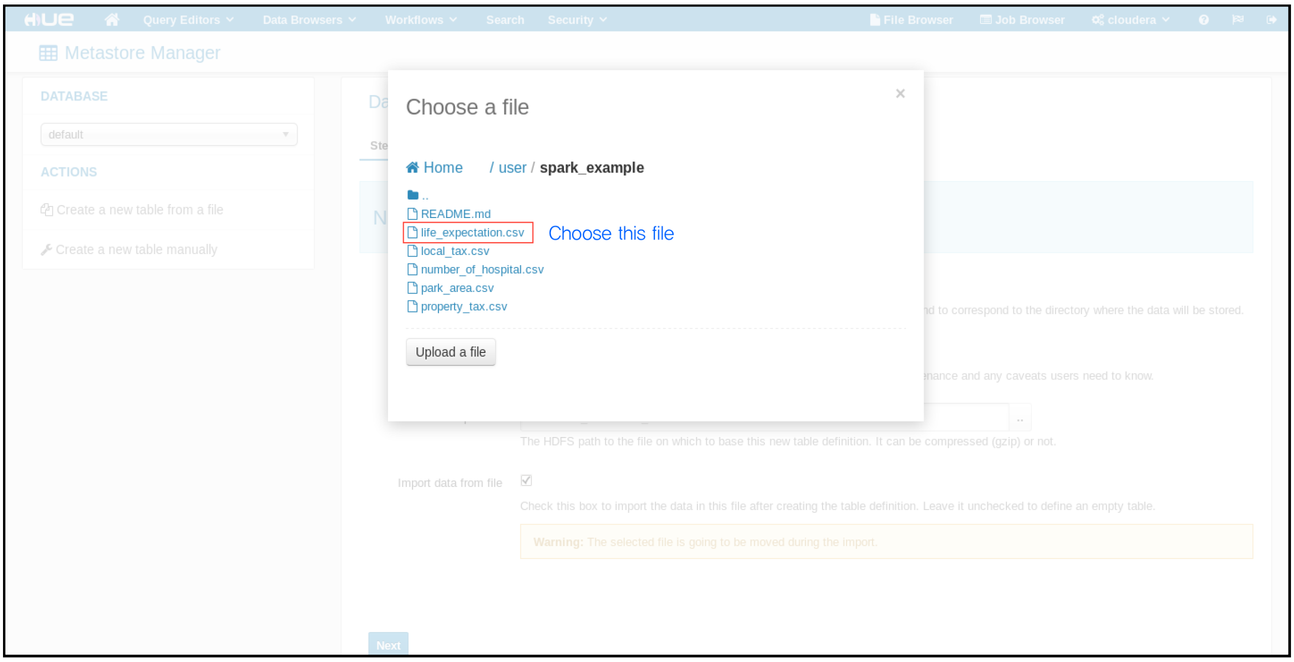
2. Click to choose the file

The HDFS path to the file on which to base this new table definition. It can be compressed (gzip) or not.

Import data from file ☑

Check this box to import the data in this file after creating the table definition. Leave it unchecked to define an empty table.

Warning: The selected file is going to be moved during the import.



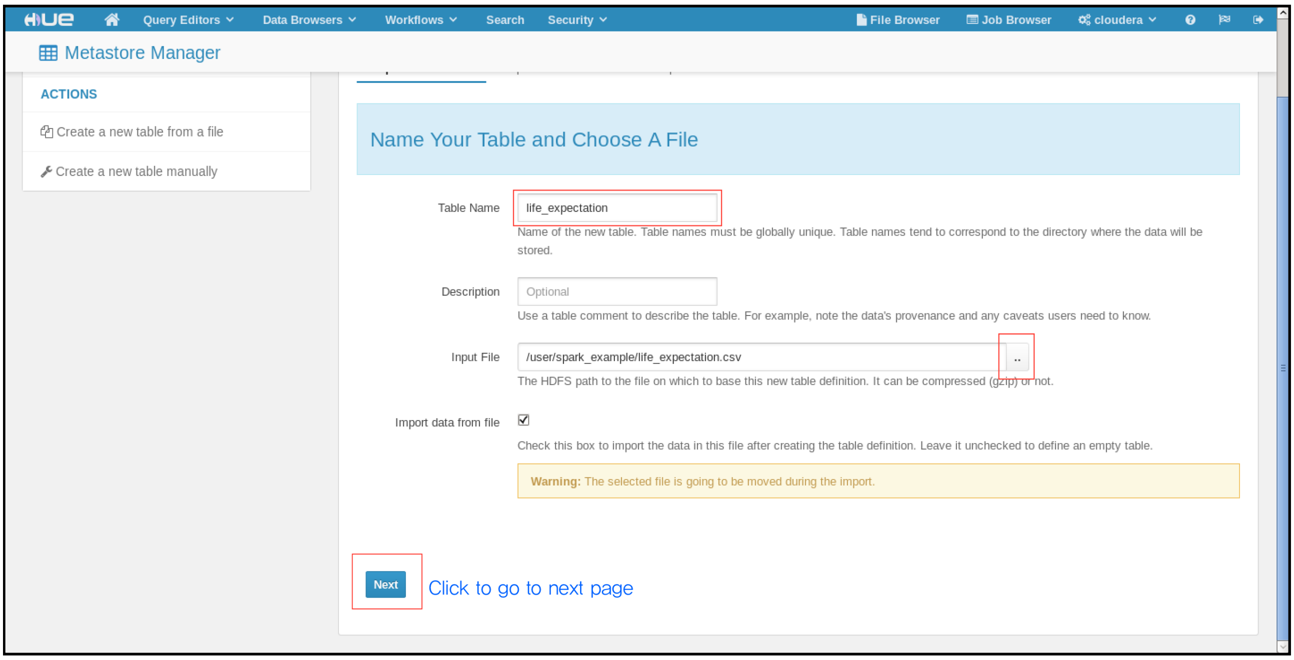
Choose a file

Home / user / spark_example

..
README.md
life_expectation.csv
local_tax.csv
number_of_hospital.csv
park_area.csv
property_tax.csv

Upload a file

Choose this file



Name Your Table and Choose A File

Table Name: life_expectation

Name of the new table. Table names must be globally unique. Table names tend to correspond to the directory where the data will be stored.

Description: Optional

Use a table comment to describe the table. For example, note the data's provenance and any caveats users need to know.

Input File: /user/spark_example/life_expectation.csv ..

The HDFS path to the file on which to base this new table definition. It can be compressed (gzip) or not.

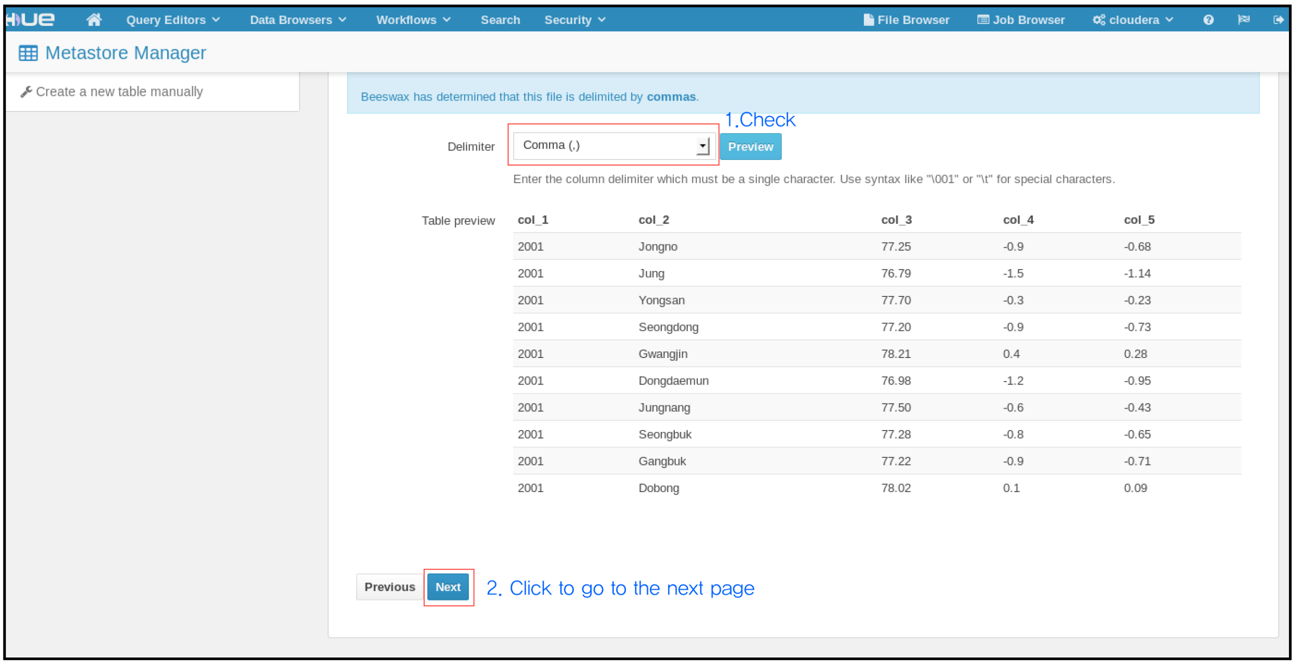
Import data from file ☑

Check this box to import the data in this file after creating the table definition. Leave it unchecked to define an empty table.

Warning: The selected file is going to be moved during the import.

Next

Click to go to next page



Metastore Manager

Create a new table manually

Beeswax has determined that this file is delimited by **commas**.

Delimiter: Comma (,) — Preview

1.Check

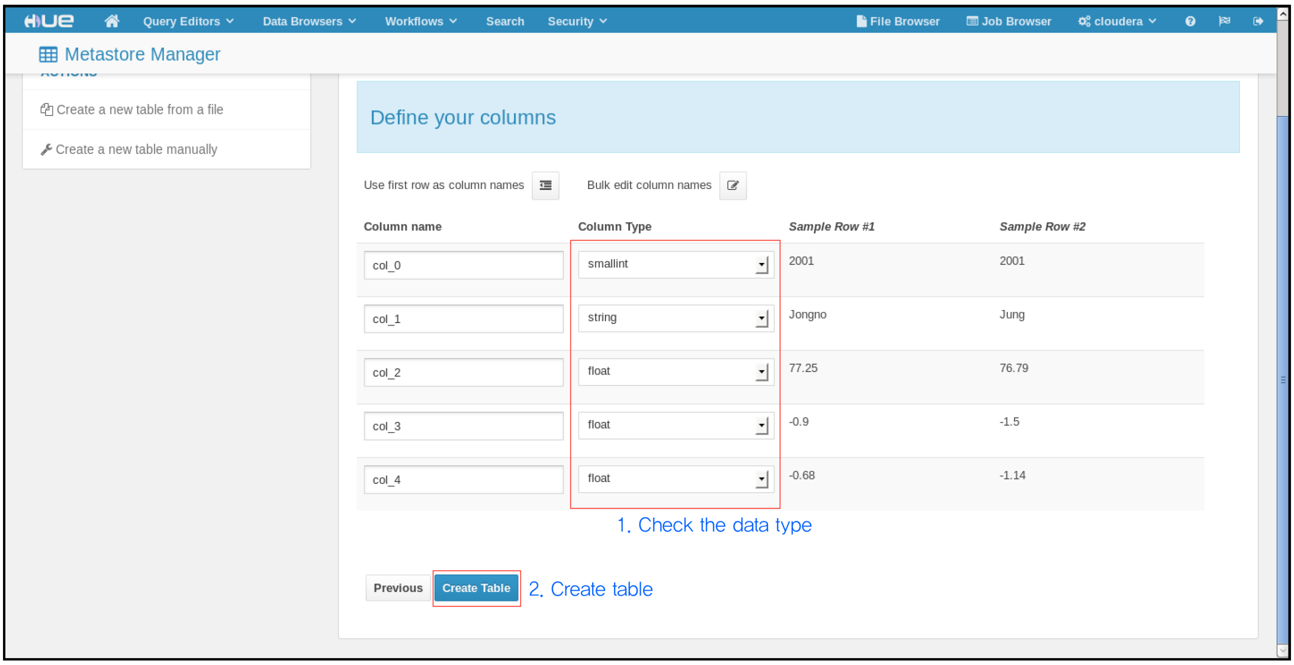
Enter the column delimiter which must be a single character. Use syntax like "\001" or "\t" for special characters.

Table preview

col_1	col_2	col_3	col_4	col_5
2001	Jongno	77.25	-0.9	-0.68
2001	Jung	76.79	-1.5	-1.14
2001	Yongsan	77.70	-0.3	-0.23
2001	Seongdong	77.20	-0.9	-0.73
2001	Gwangjin	78.21	0.4	0.28
2001	Dongdaemun	76.98	-1.2	-0.95
2001	Jungnang	77.50	-0.6	-0.43
2001	Seongbuk	77.28	-0.8	-0.65
2001	Gangbuk	77.22	-0.9	-0.71
2001	Dobong	78.02	0.1	0.09

Previous | Next

2. Click to go to the next page



Metastore Manager

ACTIONS

Create a new table from a file

Create a new table manually

Define your columns

Use first row as column names

Bulk edit column names

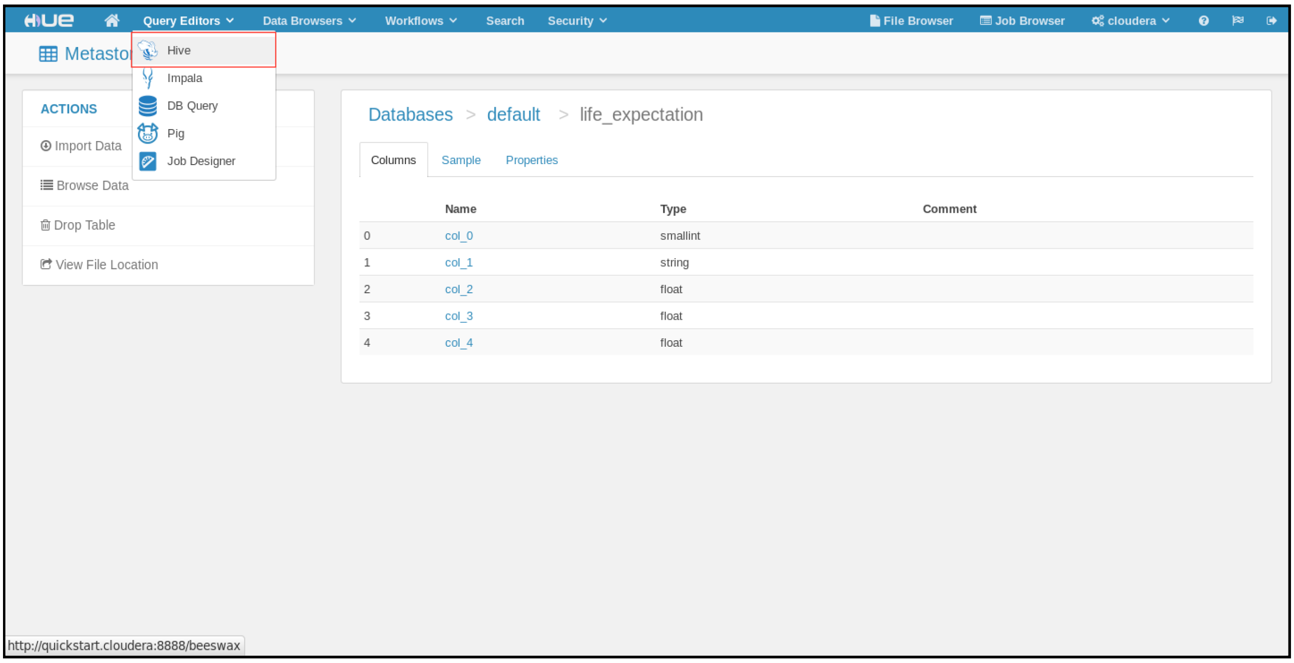
Column name	Column Type	*Sample Row #1*	*Sample Row #2*
col_0	smallint	2001	2001
col_1	string	Jongno	Jung
col_2	float	77.25	76.79
col_3	float	-0.9	-1.5
col_4	float	-0.68	-1.14

1. Check the data type

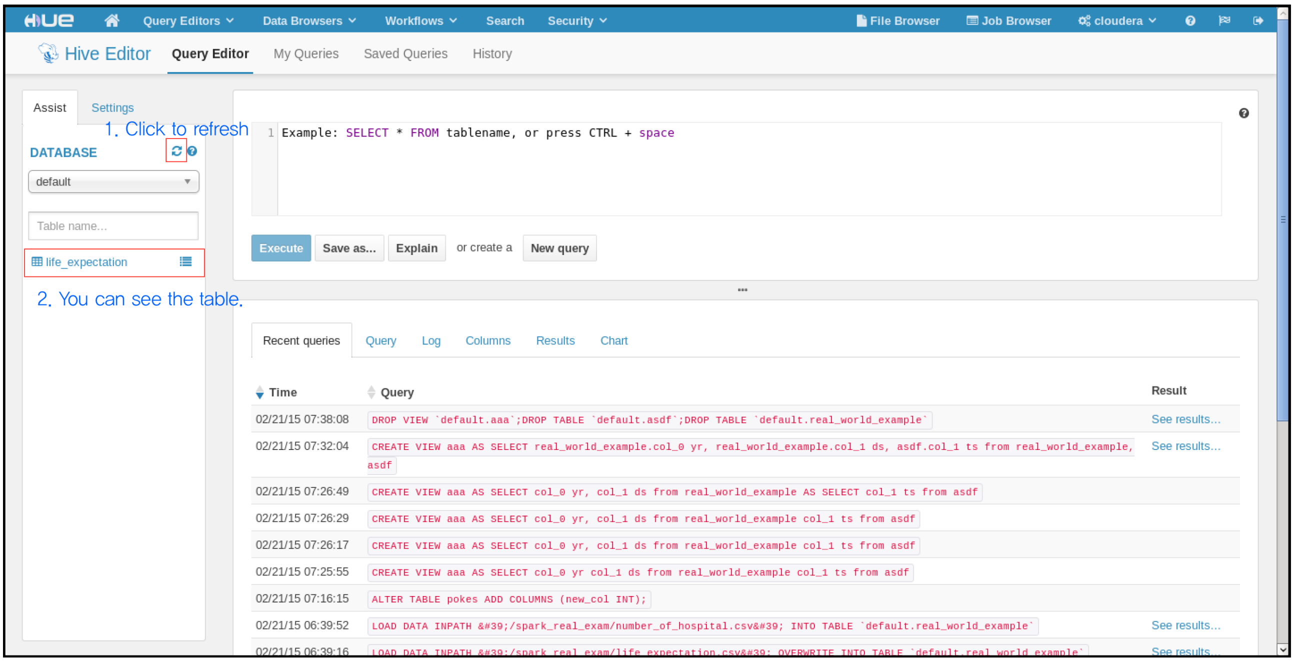
Previous | Create Table

2. Create table

4) Check the created table



	Name	Type	Comment
0	col_0	smallint	
1	col_1	string	
2	col_2	float	
3	col_3	float	
4	col_4	float	



1. Click to refresh

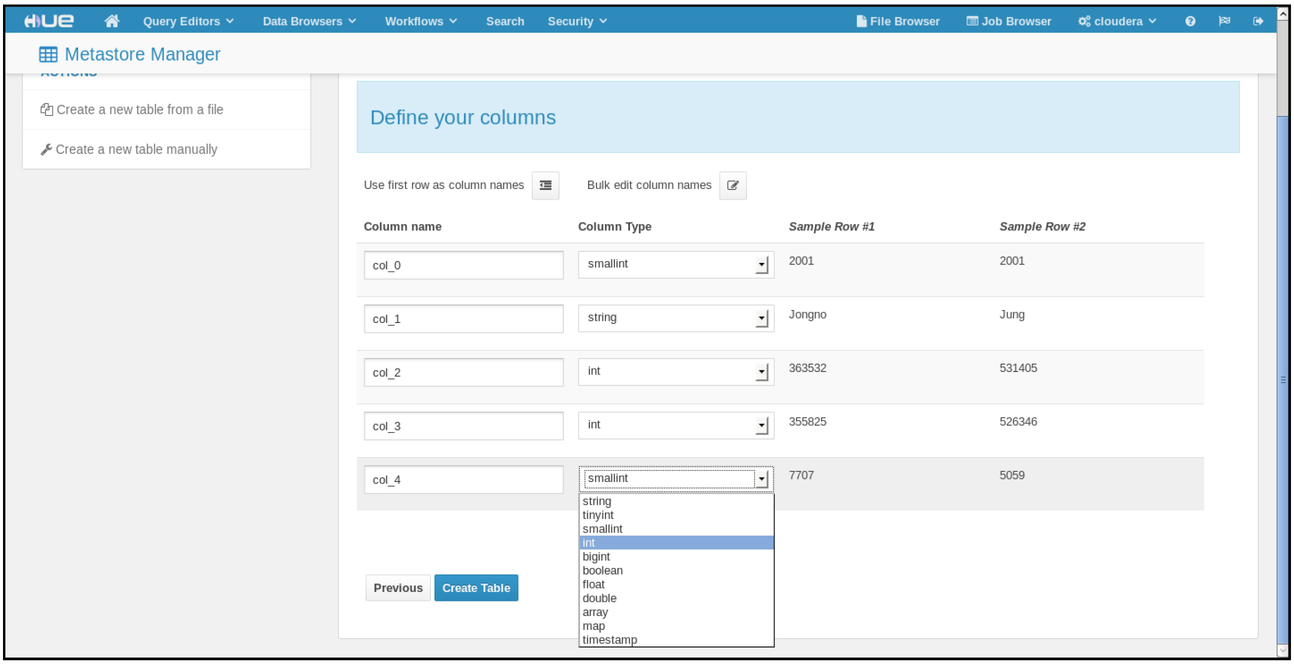
2. You can see the table.

```
Example: SELECT * FROM tablename, or press CTRL + space
```

Time	Query	Result
02/21/15 07:38:08	`DROP VIEW ` + "`default.aaa`;DROP TABLE `default.asdf`;DROP TABLE `default.real_world_example`"	See results...
02/21/15 07:32:04	CREATE VIEW aaa AS SELECT real_world_example.col_0 yr, real_world_example.col_1 ds, asdf.col_1 ts from real_world_example, asdf	See results...
02/21/15 07:26:49	CREATE VIEW aaa AS SELECT col_0 yr, col_1 ds from real_world_example AS SELECT col_1 ts from asdf	
02/21/15 07:26:29	CREATE VIEW aaa AS SELECT col_0 yr, col_1 ds from real_world_example col_1 ts from asdf	
02/21/15 07:26:17	CREATE VIEW aaa AS SELECT col_0 yr, col_1 ds from real_world_example col_1 ts from asdf	
02/21/15 07:25:55	CREATE VIEW aaa AS SELECT col_0 yr col_1 ds from real_world_example col_1 ts from asdf	
02/21/15 07:16:15	ALTER TABLE pokes ADD COLUMNS (new_col INT);	
02/21/15 06:39:52	LOAD DATA INPATH &#39;/spark_real_exam/number_of_hospital.csv&#39; INTO TABLE `default.real_world_example`	See results...
02/21/15 06:39:16	LOAD DATA INPATH &#39;/spark_real_exam/life_expectation.csv&#39; OVERWRITE INTO TABLE `default.real_world_example`	See results...

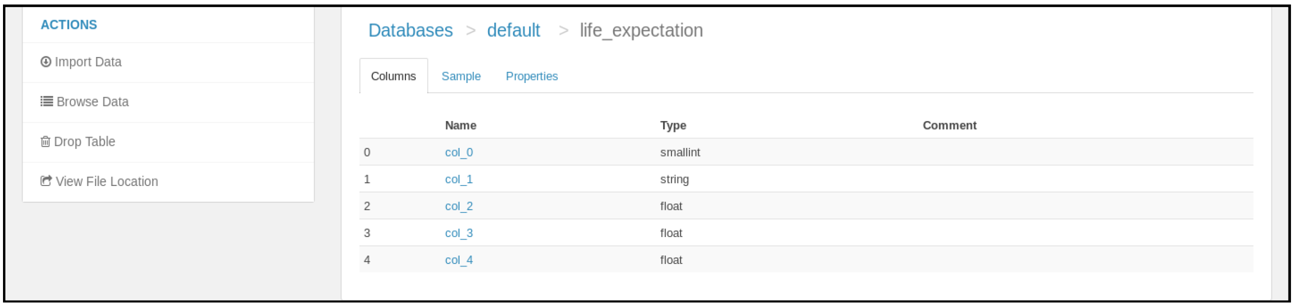
You have to create other tables using uploaded files. You can create the tables on the same way. And you must check the data type and change it if needed.

*cf. Change data type



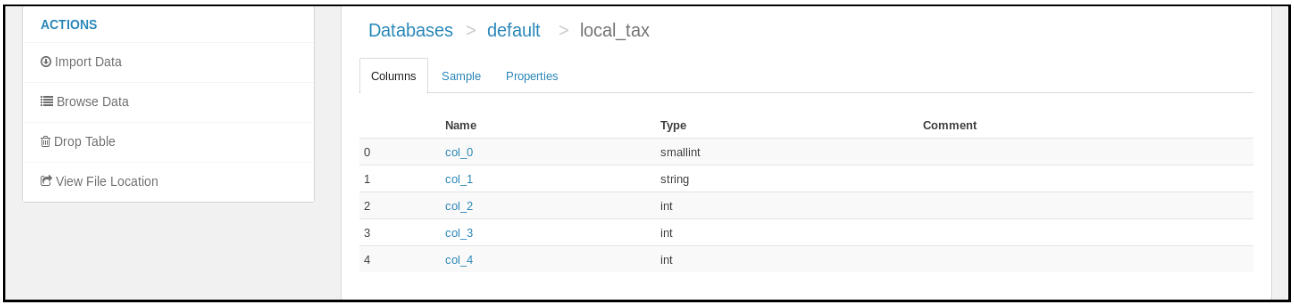
Column name	Column Type	Sample Row #1	Sample Row #2
col_0	smallint	2001	2001
col_1	string	Jongno	Jung
col_2	int	363532	531405
col_3	int	355825	526346
col_4	smallint	7707	5059

*cf. Final data type



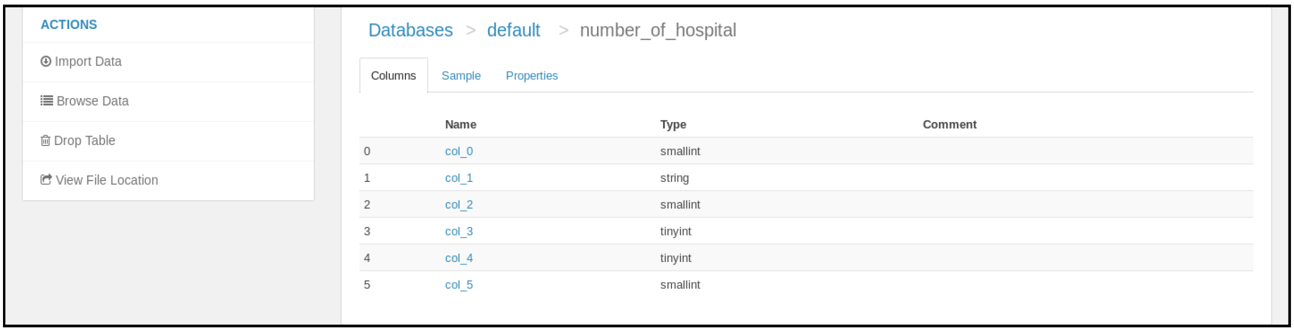
Databases > default > life_expectation

	Name	Type	Comment
0	col_0	smallint	
1	col_1	string	
2	col_2	float	
3	col_3	float	
4	col_4	float	



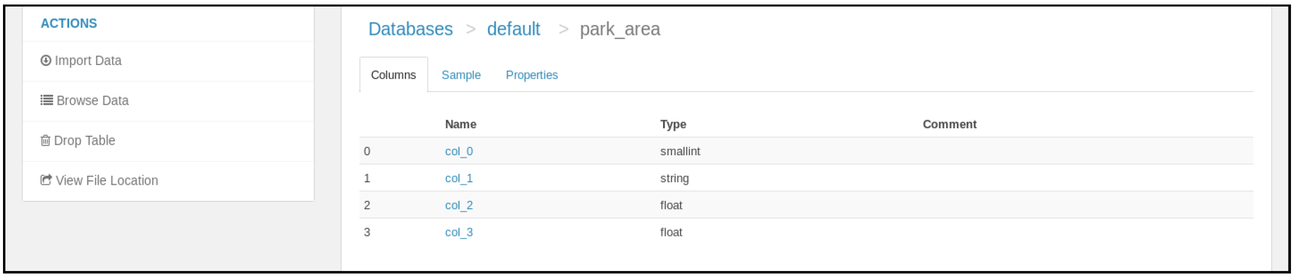
Databases > default > local_tax

	Name	Type	Comment
0	col_0	smallint	
1	col_1	string	
2	col_2	int	
3	col_3	int	
4	col_4	int	

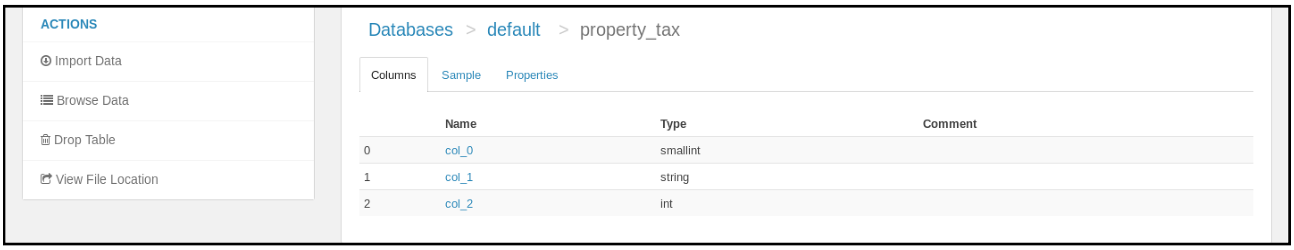


Databases > default > number_of_hospital

	Name	Type	Comment
0	col_0	smallint	
1	col_1	string	
2	col_2	smallint	
3	col_3	tinyint	
4	col_4	tinyint	
5	col_5	smallint	

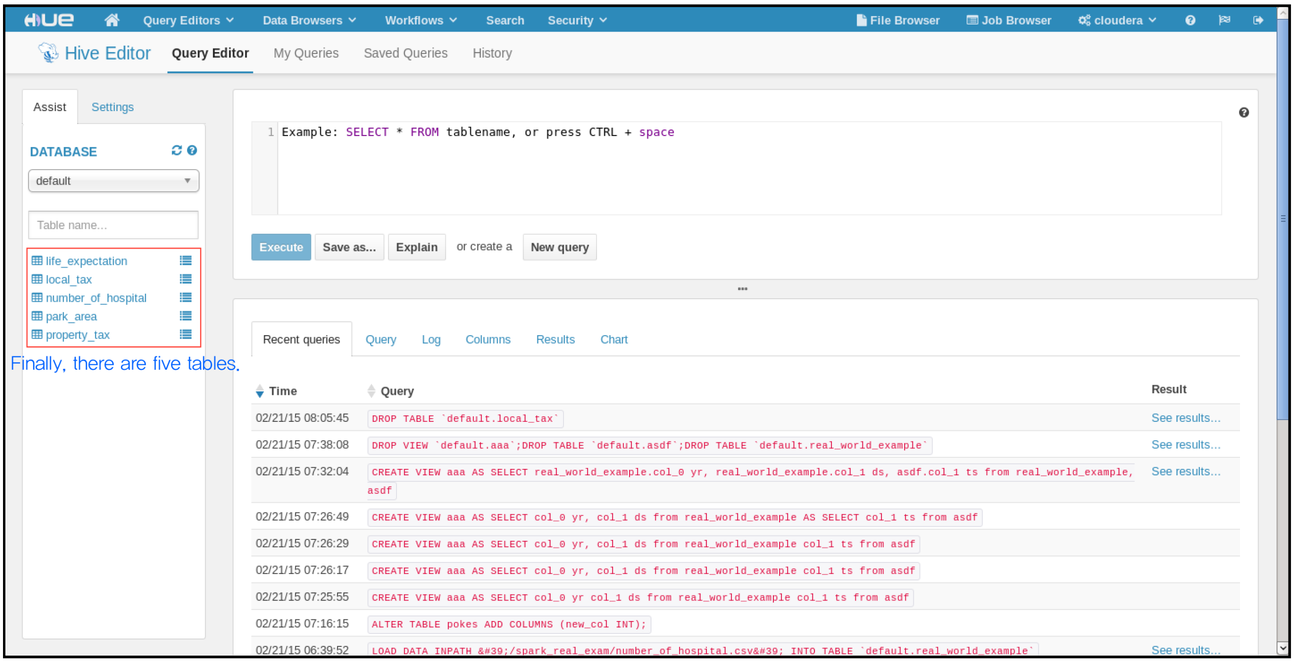


	Name	Type	Comment
0	col_0	smallint	
1	col_1	string	
2	col_2	float	
3	col_3	float	



	Name	Type	Comment
0	col_0	smallint	
1	col_1	string	
2	col_2	int	

5) Result of creating tables



```
Example: SELECT * FROM tablename, or press CTRL + space
```

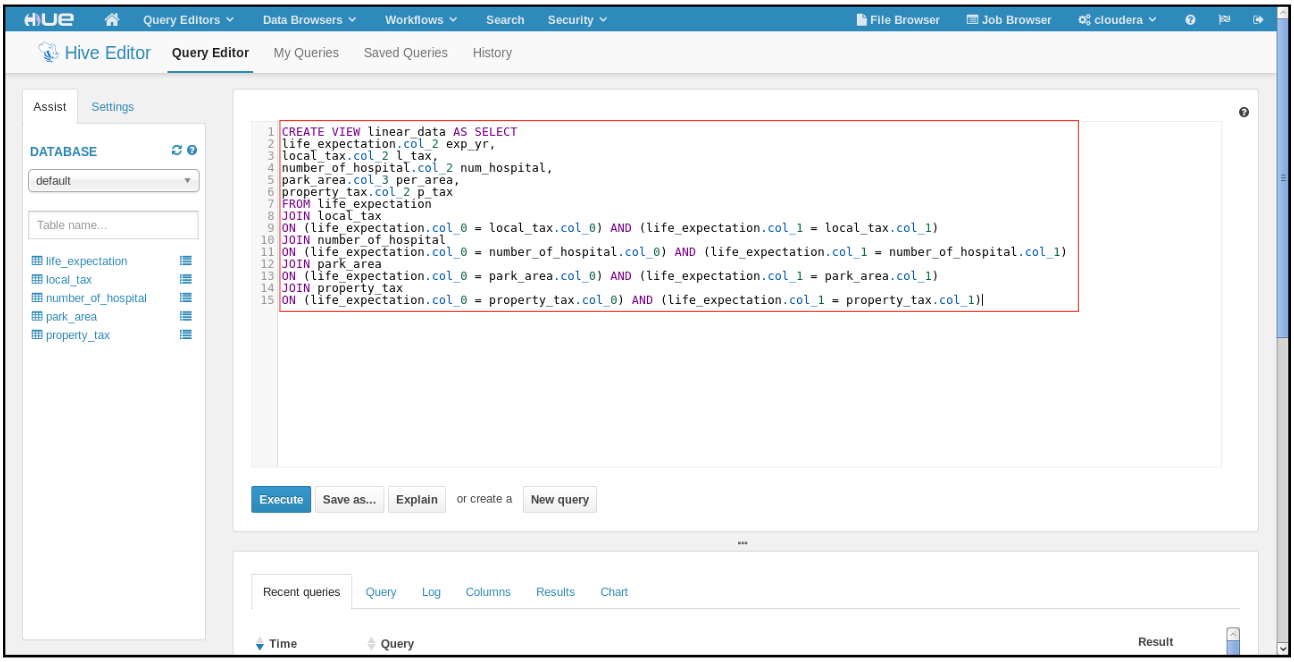
Finally, there are five tables.

Time	Query	Result
02/21/15 08:05:45	`DROP TABLE `default.local_tax``	See results...
02/21/15 07:38:08	`DROP VIEW `default.aaa`;DROP TABLE `default.asdf`;DROP TABLE `default.real_world_example``	See results...
02/21/15 07:32:04	`CREATE VIEW aaa AS SELECT real_world_example.col_0 yr, real_world_example.col_1 ds, asdf.col_1 ts from real_world_example, asdf`	See results...
02/21/15 07:26:49	`CREATE VIEW aaa AS SELECT col_0 yr, col_1 ds from real_world_example AS SELECT col_1 ts from asdf`	
02/21/15 07:26:29	`CREATE VIEW aaa AS SELECT col_0 yr, col_1 ds from real_world_example col_1 ts from asdf`	
02/21/15 07:26:17	`CREATE VIEW aaa AS SELECT col_0 yr, col_1 ds from real_world_example col_1 ts from asdf`	
02/21/15 07:25:55	`CREATE VIEW aaa AS SELECT col_0 yr col_1 ds from real_world_example col_1 ts from asdf`	
02/21/15 07:16:15	`ALTER TABLE pokes ADD COLUMNS (new_col INT);`	
02/21/15 06:39:52	`LOAD DATA INPATH &#39;/spark_real_exam/number_of_hospital.csv&#39; INTO TABLE `default.real_world_example``	See results...

6) Create view to refine data

Five data tables have a lot of data and some of them are not needed. So you have gather the needed data into one file.

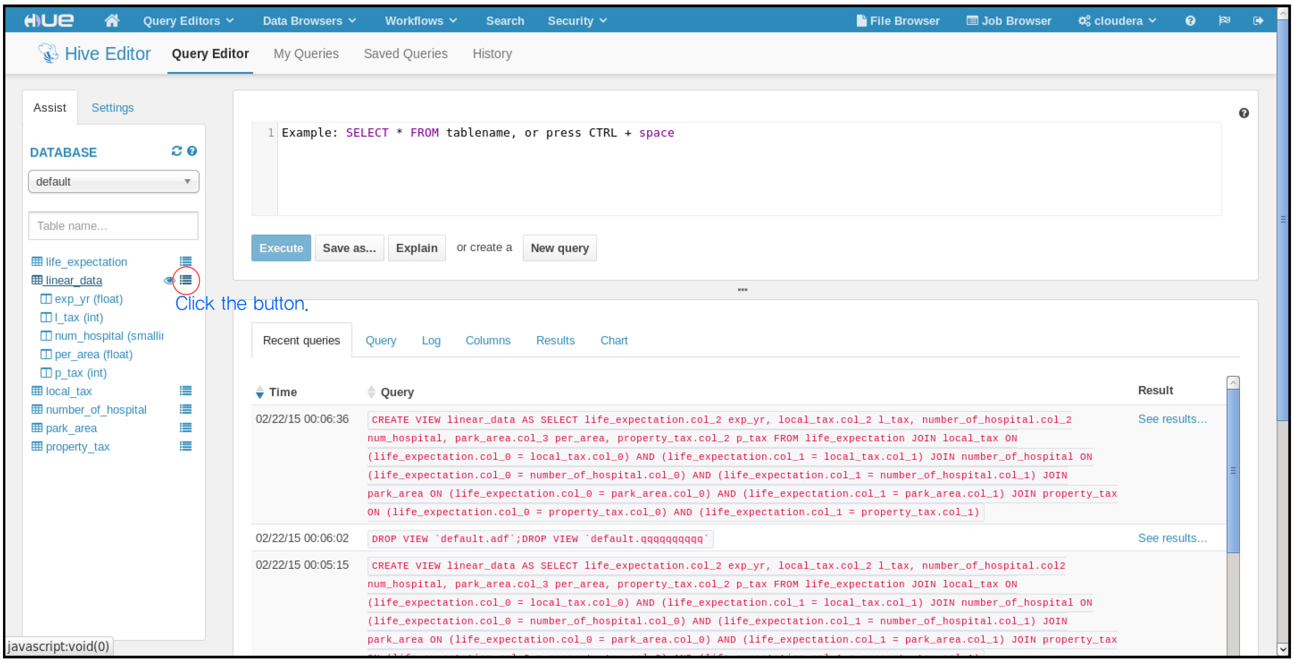
```
CREATE VIEW linear_data AS SELECT
life_expectation.col_2 exp_yr,
local_tax.col_2 l_tax,
number_of_hospital.col_2 num_hospital,
park_area.col_3 per_area,
property_tax.col_2 p_tax
FROM life_expectation
JOIN local_tax
ON (life_expectation.col_0 = local_tax.col_0) AND (life_expectation.col_1 = local_tax.col_1)
JOIN number_of_hospital
ON (life_expectation.col_0 = number_of_hospital.col_0) AND (life_expectation.col_1 =
number_of_hospital.col_1)
JOIN park_area
ON (life_expectation.col_0 = park_area.col_0) AND (life_expectation.col_1 = park_area.col_1)
JOIN property_tax
ON (life_expectation.col_0 = property_tax.col_0) AND (life_expectation.col_1 = property_tax.col_1)
```



```
CREATE VIEW linear_data AS SELECT
life_expectation.col_2 exp_yr,
local_tax.col_2 l_tax,
number_of_hospital.col_2 num_hospital,
park_area.col_3 per_area,
property_tax.col_2 p_tax
FROM life_expectation
JOIN local_tax
ON (life_expectation.col_0 = local_tax.col_0) AND (life_expectation.col_1 = local_tax.col_1)
JOIN number_of_hospital
ON (life_expectation.col_0 = number_of_hospital.col_0) AND (life_expectation.col_1 = number_of_hospital.col_1)
JOIN park_area
ON (life_expectation.col_0 = park_area.col_0) AND (life_expectation.col_1 = park_area.col_1)
JOIN property_tax
ON (life_expectation.col_0 = property_tax.col_0) AND (life_expectation.col_1 = property_tax.col_1)
```

Click the “Excute" button.

7) Check the result of creating view

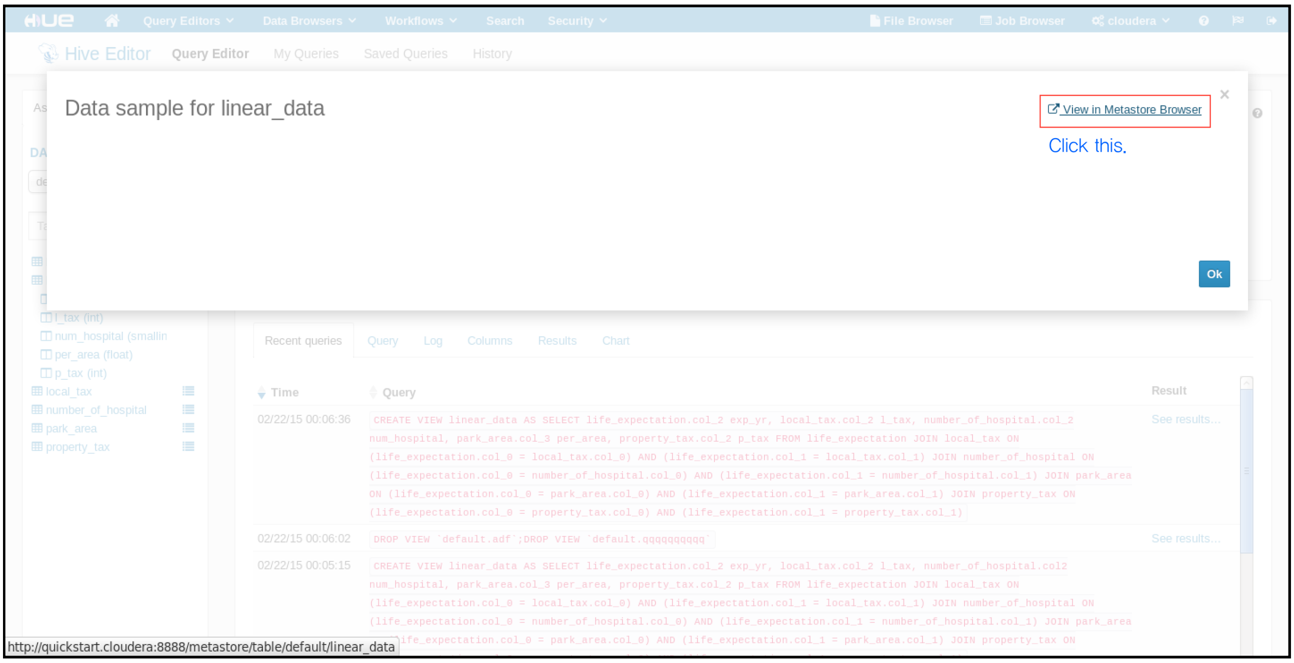


```
Example: SELECT * FROM tablename, or press CTRL + space
```

Click the button.

Time	Query	Result
02/22/15 00:06:36	`CREATE VIEW linear_data AS SELECT life_expectation.col_2 exp_yr, local_tax.col_2 l_tax, number_of_hospital.col_2 num_hospital, park_area.col_3 per_area, property_tax.col_2 p_tax FROM life_expectation JOIN local_tax ON (life_expectation.col_0 = local_tax.col_0) AND (life_expectation.col_1 = local_tax.col_1) JOIN number_of_hospital ON (life_expectation.col_0 = number_of_hospital.col_0) AND (life_expectation.col_1 = number_of_hospital.col_1) JOIN park_area ON (life_expectation.col_0 = park_area.col_0) AND (life_expectation.col_1 = park_area.col_1) JOIN property_tax ON (life_expectation.col_0 = property_tax.col_0) AND (life_expectation.col_1 = property_tax.col_1)`	See results...
02/22/15 00:06:02	``DROP VIEW `default.adf`;DROP VIEW `default.qqqqqqqqqq` ``	See results...
02/22/15 00:05:15	`CREATE VIEW linear_data AS SELECT life_expectation.col_2 exp_yr, local_tax.col_2 l_tax, number_of_hospital.col2 num_hospital, park_area.col_3 per_area, property_tax.col_2 p_tax FROM life_expectation JOIN local_tax ON (life_expectation.col_0 = local_tax.col_0) AND (life_expectation.col_1 = local_tax.col_1) JOIN number_of_hospital ON (life_expectation.col_0 = number_of_hospital.col_0) AND (life_expectation.col_1 = number_of_hospital.col_1) JOIN park_area ON (life_expectation.col_0 = park_area.col_0) AND (life_expectation.col_1 = park_area.col_1) JOIN property_tax`	

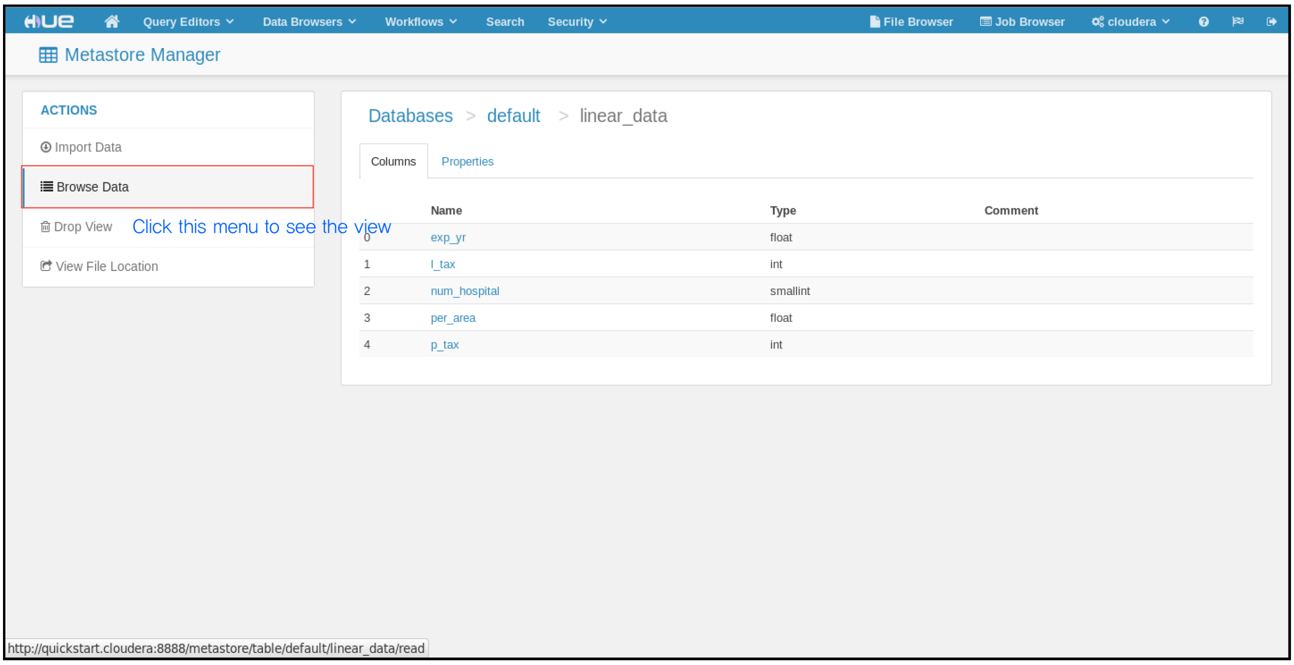
javascript:void(0)



Data sample for linear_data

Click this.

http://quickstart.cloudera:8888/metastore/table/default/linear_data



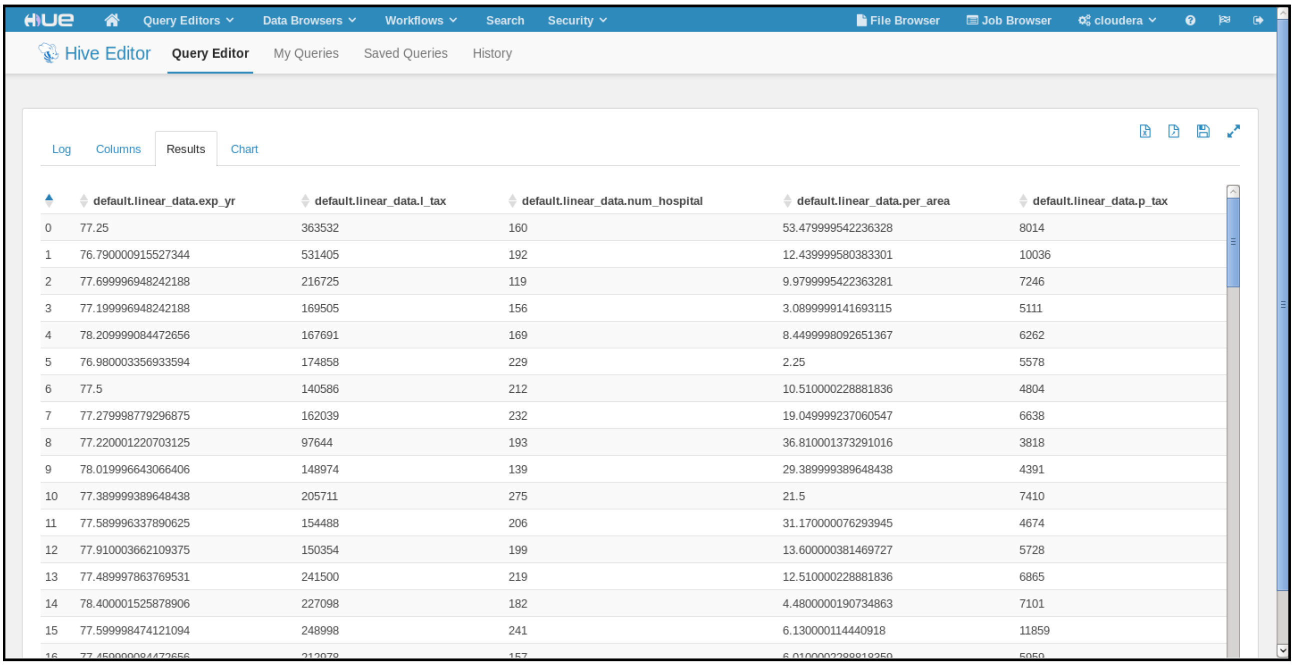
Click this menu to see the view

Databases > default > linear_data

	Name	Type	Comment
0	exp_yr	float	
1	l_tax	int	
2	num_hospital	smallint	
3	per_area	float	
4	p_tax	int	

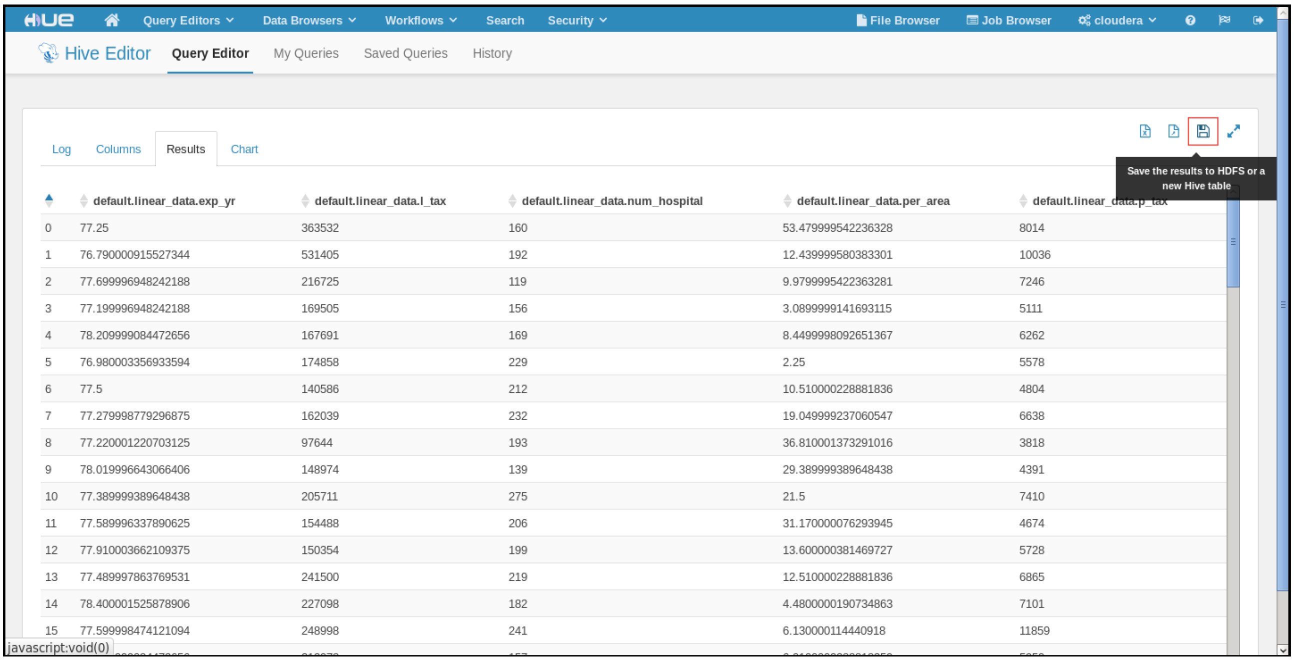
http://quickstart.cloudera:8888/metastore/table/default/linear_data/read

Now, you can see the refined dataset. There are five columns.
Column 1. Life expectation (y data)
Column 2. Local tax (x1 data)
Column 3. Number of hospital (x2 data)
Column4. Park area per a person (x3 data)
Column5. Property tax (x4 data)

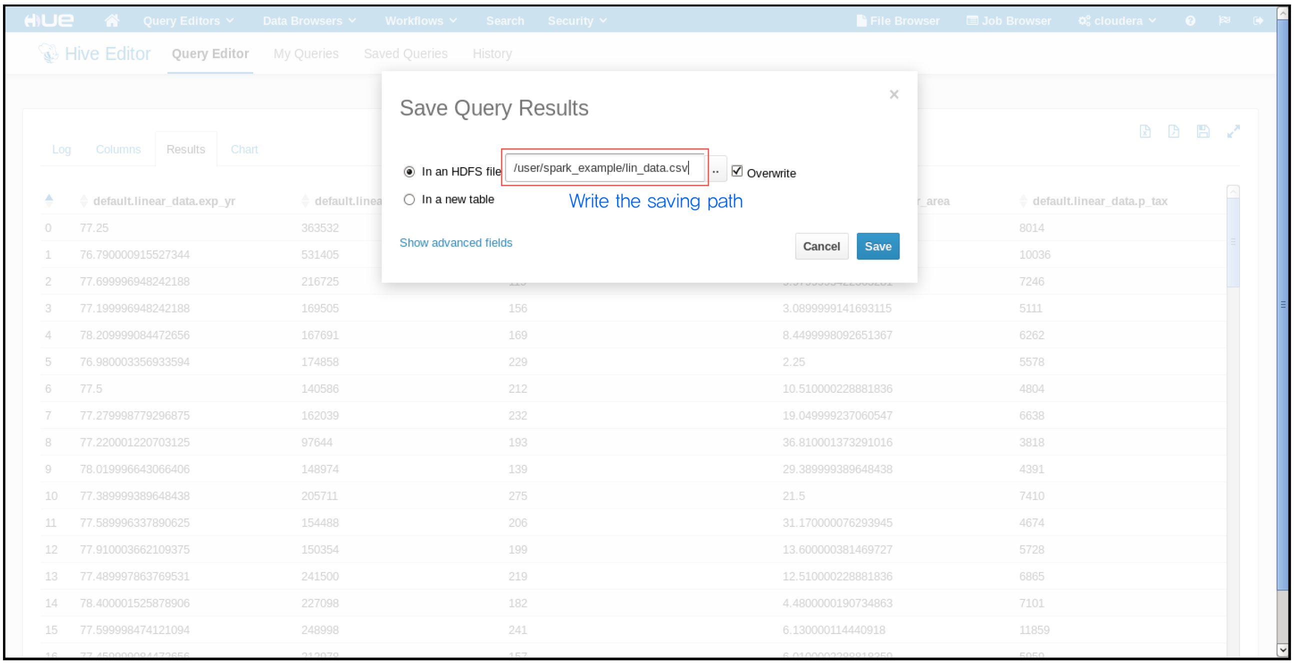


	default.linear_data.exp_yr	default.linear_data.l_tax	default.linear_data.num_hospital	default.linear_data.per_area	default.linear_data.p_tax
0	77.25	363532	160	53.479999542236328	8014
1	76.790000915527344	531405	192	12.439999580383301	10036
2	77.699996948242188	216725	119	9.9799995422363281	7246
3	77.199996948242188	169505	156	3.0899999141693115	5111
4	78.209999084472656	167691	169	8.4499998092651367	6262
5	76.980003356933594	174858	229	2.25	5578
6	77.5	140586	212	10.510000228881836	4804
7	77.279998779296875	162039	232	19.049999237060547	6638
8	77.220001220703125	97644	193	36.810001373291016	3818
9	78.019996643066406	148974	139	29.389999389648438	4391
10	77.389999389648438	205711	275	21.5	7410
11	77.589996337890625	154488	206	31.170000076293945	4674
12	77.910003662109375	150354	199	13.600000381469727	5728
13	77.489997863769531	241500	219	12.510000228881836	6865
14	78.400001525878906	227098	182	4.4800000190734863	7101
15	77.599998474121094	248998	241	6.130000114440918	11859
16	77.459999084472656	212078	157	6.0100002288818359	5959

8) Save the refined data to file on HDFS



	default.linear_data.exp_yr	default.linear_data.l_tax	default.linear_data.num_hospital	default.linear_data.per_area	default.linear_data.p_tax
0	77.25	363532	160	53.479999542236328	8014
1	76.790000915527344	531405	192	12.439999580383301	10036
2	77.699996948242188	216725	119	9.9799995422363281	7246
3	77.199996948242188	169505	156	3.0899999141693115	5111
4	78.209999084472656	167691	169	8.4499998092651367	6262
5	76.980003356933594	174858	229	2.25	5578
6	77.5	140586	212	10.510000228881836	4804
7	77.279998779296875	162039	232	19.049999237060547	6638
8	77.220001220703125	97644	193	36.810001373291016	3818
9	78.019996643066406	148974	139	29.389999389648438	4391
10	77.389999389648438	205711	275	21.5	7410
11	77.589996337890625	154488	206	31.170000076293945	4674
12	77.910003662109375	150354	199	13.600000381469727	5728
13	77.489997863769531	241500	219	12.510000228881836	6865
14	78.400001525878906	227098	182	4.4800000190734863	7101
15	77.599998474121094	248998	241	6.130000114440918	11859



Save Query Results

● In an HDFS file /user/spark_example/lin_data.csv ☑ Overwrite

○ In a new table

Write the saving path

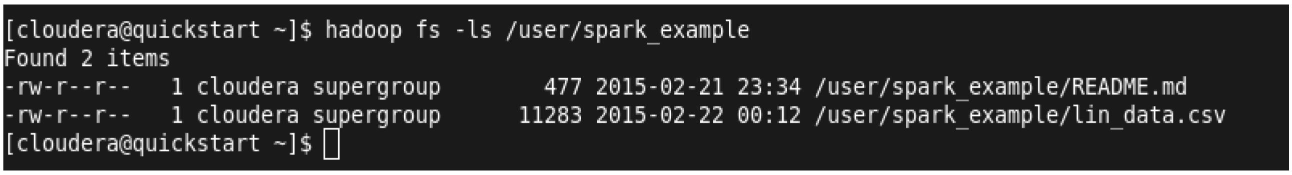
Show advanced fields

Cancel Save

	default.linear_data.exp_yr	default.linea		_area	default.linear_data.p_tax
0	77.25	363532			8014
1	76.790000915527344	531405			10036
2	77.699996948242188	216725			7246
3	77.199996948242188	169505	156	3.0899999141693115	5111
4	78.209999084472656	167691	169	8.4499998092651367	6262
5	76.980003356933594	174858	229	2.25	5578
6	77.5	140586	212	10.510000228881836	4804
7	77.279998779296875	162039	232	19.049999237060547	6638
8	77.220001220703125	97644	193	36.810001373291016	3818
9	78.019996643066406	148974	139	29.389999389648438	4391
10	77.389999389648438	205711	275	21.5	7410
11	77.589996337890625	154488	206	31.170000076293945	4674
12	77.910003662109375	150354	199	13.600000381469727	5728
13	77.489997863769531	241500	219	12.510000228881836	6865
14	78.400001525878906	227098	182	4.4800000190734863	7101
15	77.599998474121094	248998	241	6.130000114440918	11859

9) Check the refined file on HDFS

```
[cloudera@quickstart ~] hadoop fs -ls /user/spark_example
```



```
[cloudera@quickstart ~]$ hadoop fs -ls /user/spark_example
Found 2 items
-rw-r--r--   1 cloudera supergroup        477 2015-02-21 23:34 /user/spark_example/README.md
-rw-r--r--   1 cloudera supergroup      11283 2015-02-22 00:12 /user/spark_example/lin_data.csv
[cloudera@quickstart ~]$ 
```

Credits : JihoKim