May 6, 2014

HareDB HBase Client Web Version (1.94.03 & 1.94.01.02)

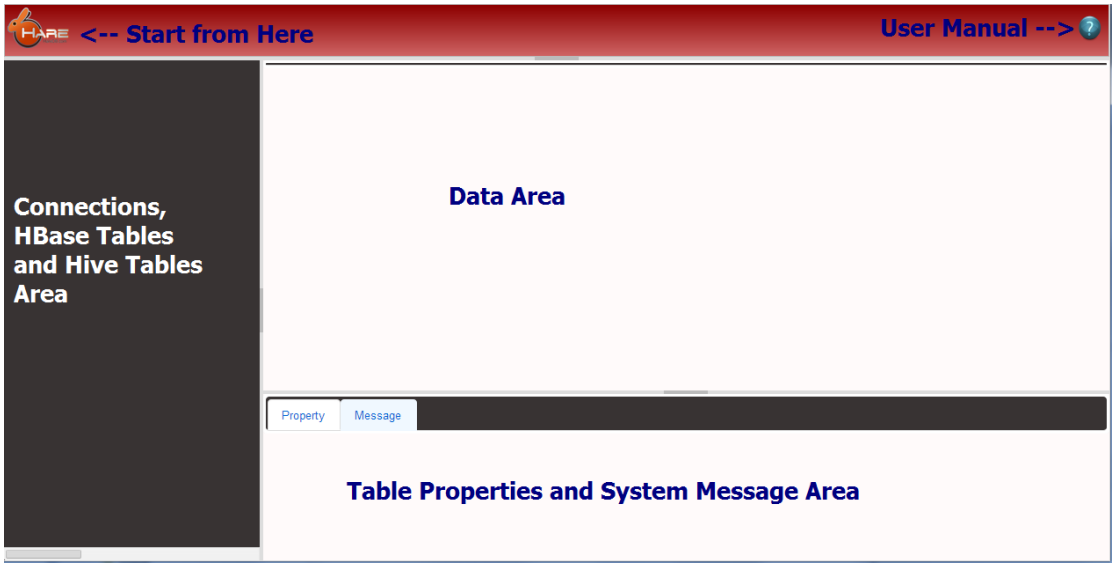
USER MANUAL

HareDB Team

内容

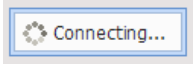
Over All....2
Connect to HBase....2
Connection....3
Connection Manager....3
Add a new Connection....4
Alter Connection....6
Delete Connection....6
Clone Connection....7
Rename Connection....8
Tables....9
Create Table....9
Alter Table....11
Add Column Family....12
Edit Column Family's Properties....14
Drop Table....15
Table metadata....16
Create Meta....16
Create Hive Table....18
Query Data....19
Register a coprocessor....19
Open HQL Command (HareQL Command)....21
Open HTable....23
Open Hive Table....24
Data Operation....26
Put data....26
Modify data....26
New Qualifier....27
Bulkload (only in Full Function Version)....28
Appendix A -- Introduction of HareQL....34
What is HareQL?....34
Why Hare?....34
What do we support right now?....34
Syntax....34
Operators....35
Functions....35

Over All

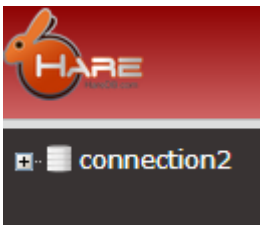


Connect to HBase

1. Click the icon  .
2. Select a connection you want to connect to.

3. The system will try to connect to HBase and show 

4. After connection is succeed, the connection node will show up.

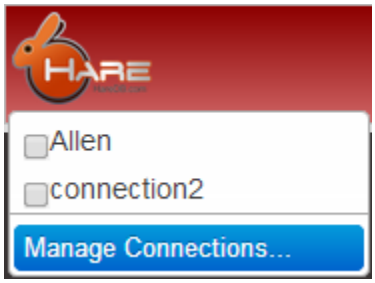


Connection

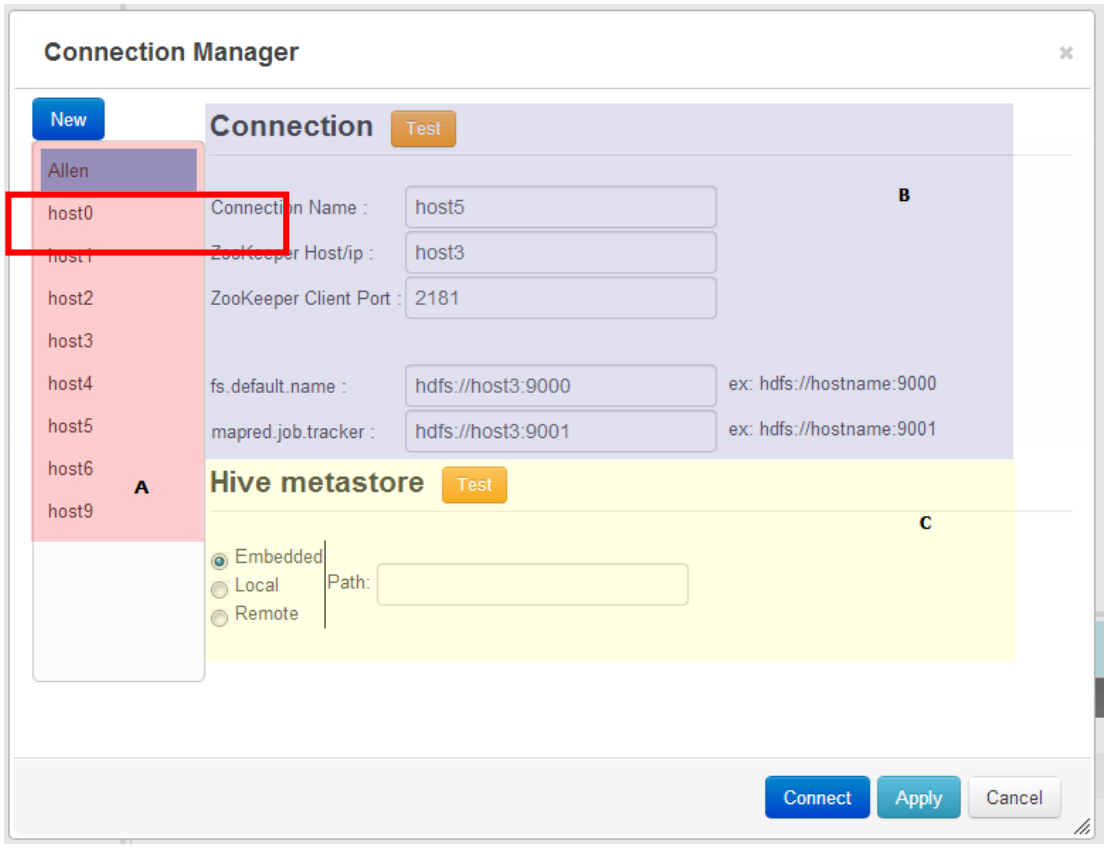
We define a “Connection” which has two parts of information.
One is for Hadoop and HBase Cluster.
Another is for Hive metastore which is a place to store meta.

Connection Manager

1. Connection Manager is a component which can help us to add、modify and delete a connection.
2. There is an item “Manage Connections” in the bottom of the connection list.
3. Please select “Manage Connections”



4. After step 2, System will pop up a Connection Manager windows.



There are 3 areas in the “Connection Manager” window.

Area A:

It is a connection list which contents all connections you have.

Area B:

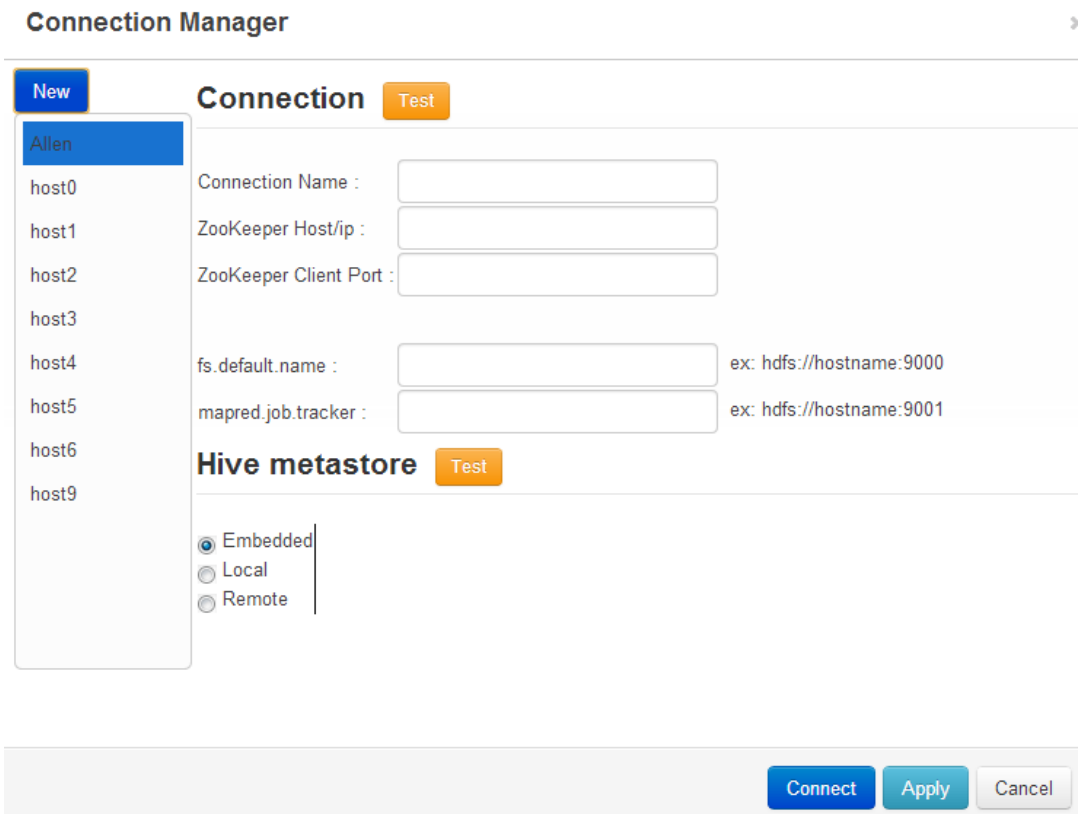
There are information of the connection you selected related with Hadoop and HBase Cluster

Area C:

There are information of the connection you selected related hive metastore.

Add a new Connection

1. to open a “Connection Manager”
2. Click the button 



Here are the information you need.

Connection

Connection Name: Just a name which will show up at the connection list.

Zookeeper Host/ip: Input zookeeper’s host name or ip.

ZooKeeper Client Port: Input the port of zookeeper’s that your request can through

Hive metastore

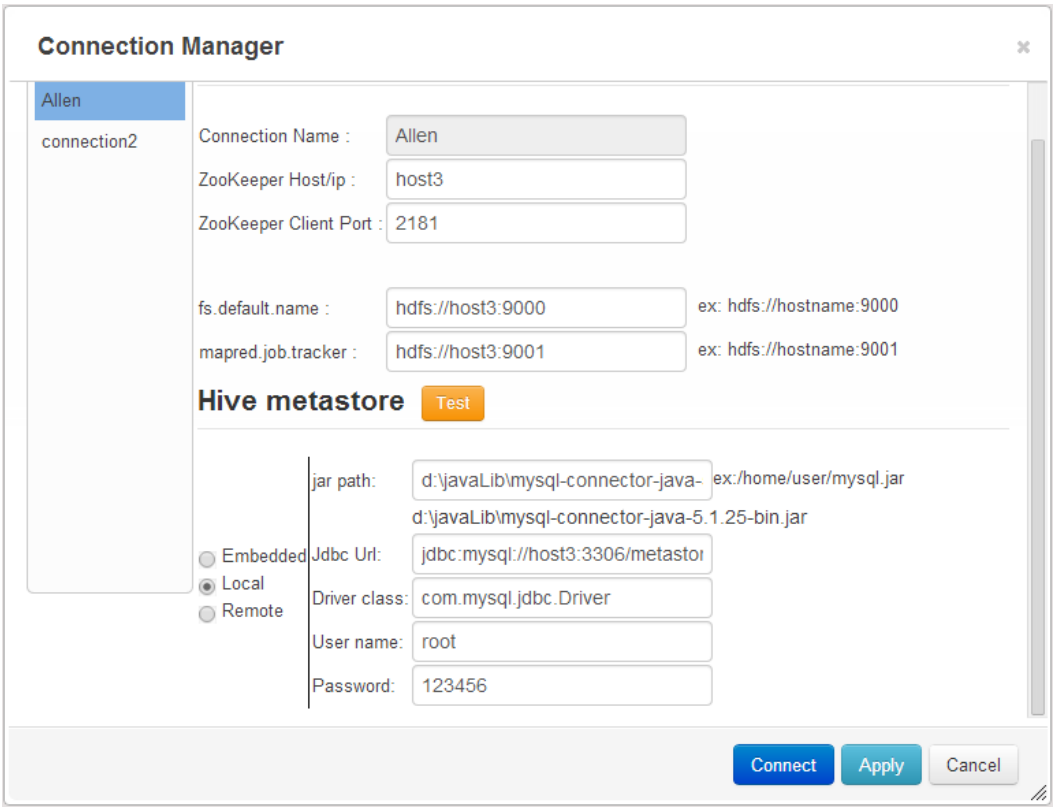
These information is for metadata of the HTable. We rely on the metadata database of Apache Hive.

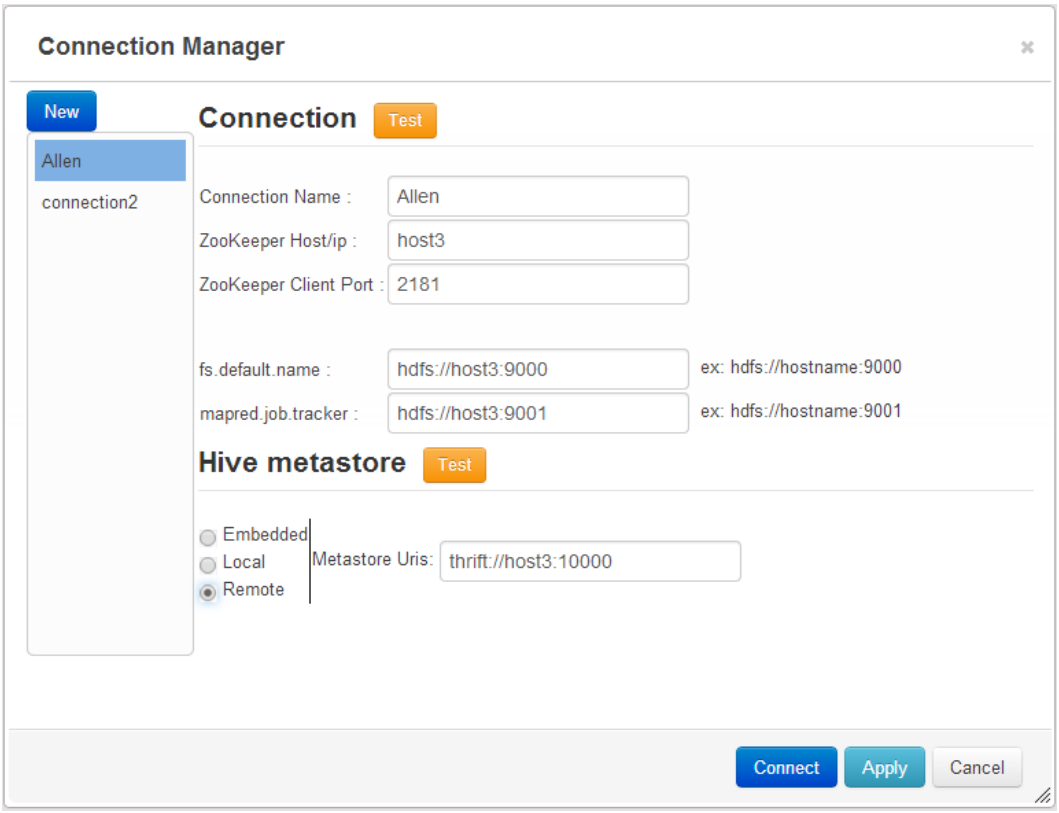
We support all kinds of hive metadata database.

First, it's "Embedded". If you don't have any hive metadata database or you didn't install hive or even you don't know what is Hive. Please select "Embedded"

Second, it's "Local". If you have an RDB to be a Hive metadata database, please select "Local". And please input the rdb connection information.

Third, it's "Remote". Apache Hive provide a thrift service to interactive with metadata database. So, please provide the url of thrift server.
For example: http://192.168.1.11:10000
192.168.1.11 => your hive thrift server.
10000 => Thrift server port.





After filling the information, we have to click the button 

Hereafter, the new connection will appear to the connection list in the left side of "Connection Manager".

Alter Connection

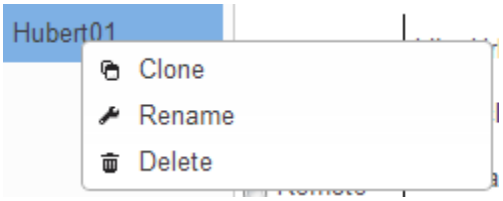
1. Open "Connection Manager" (Reference: Connection Manager)
2. Select a connection you want to alter in the connection list.

Furthermore, the information of the connection you selected will show up at the right side of the Connection Manager.

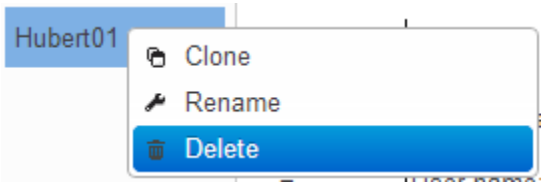
3. To modify any information you want to, then click the button 

Delete Connection

1. Open "Connection Manager" (Reference: Connection Manager)
2. Right click on the connection you want to delete. Then it will pop up a menu.

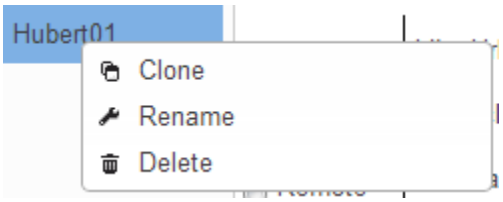


3. Please select “Delete” and it’s done.

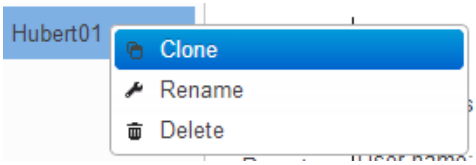


Clone Connection

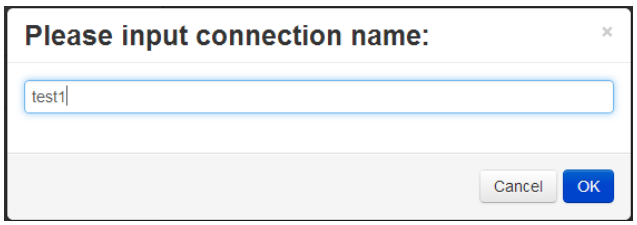
1. Open “Connection Manager” (Reference: Connection Manager)
2. Right click on the connection you want to clone. Then it will pop up a menu.



3. Please select “Clone”. System will up an input window.

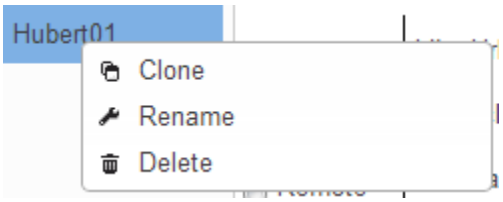


4. Please input a connection name

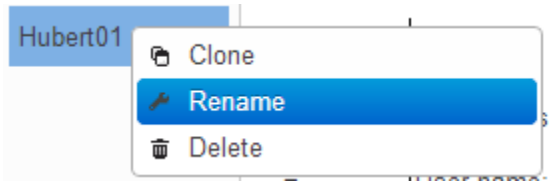


Rename Connection

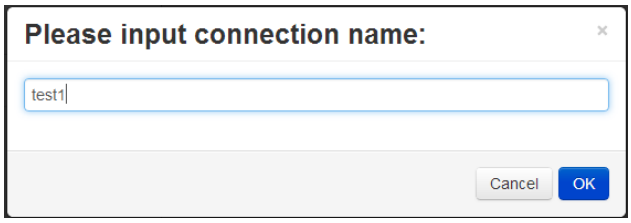
1. Open “Connection Manager” (Reference: Connection Manager)
2. Right click on the connection you want to rename. Then it will pop up a menu.



3. Please select “Rename”. System will up an input window.



4. Please input a new connection name



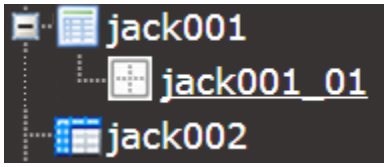
Tables

We defines 3 kinds of table. Each kinds of table has different icon.

 It's an HBase table with no table metadata.

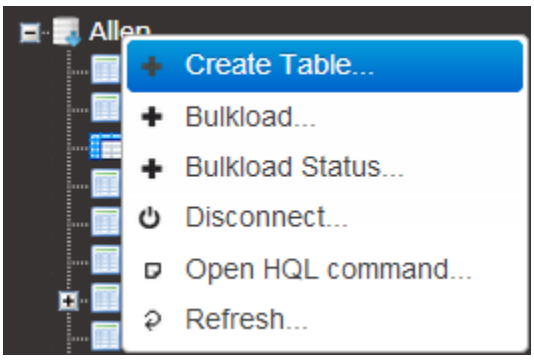
 It's an HBase table with table metadata

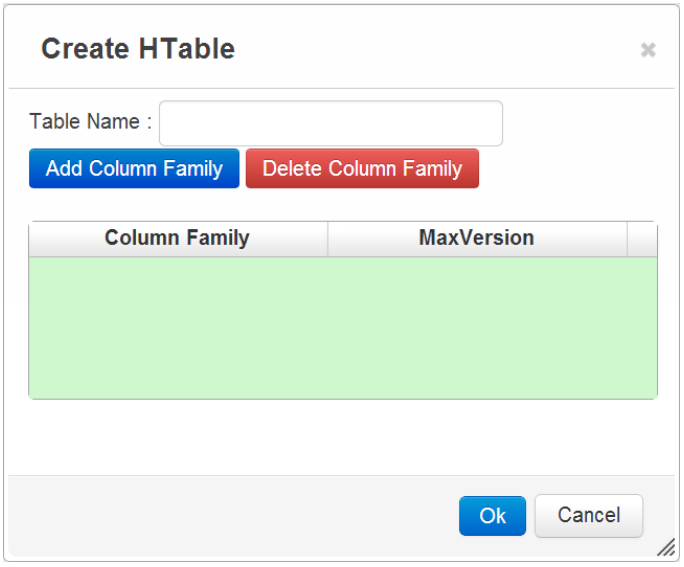
 It's a Hive table with table metadata. This hive table is a child of HBase table.



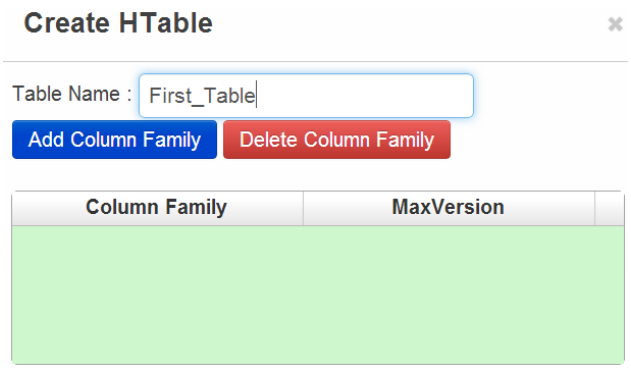
Create Table

1. Select a connection and connect to HBase. (Reference "Connect to HBase")
2. Right click on the connection icon and it will pop up a menu.
3. Please select "Create Table", and it will pop up a windows.

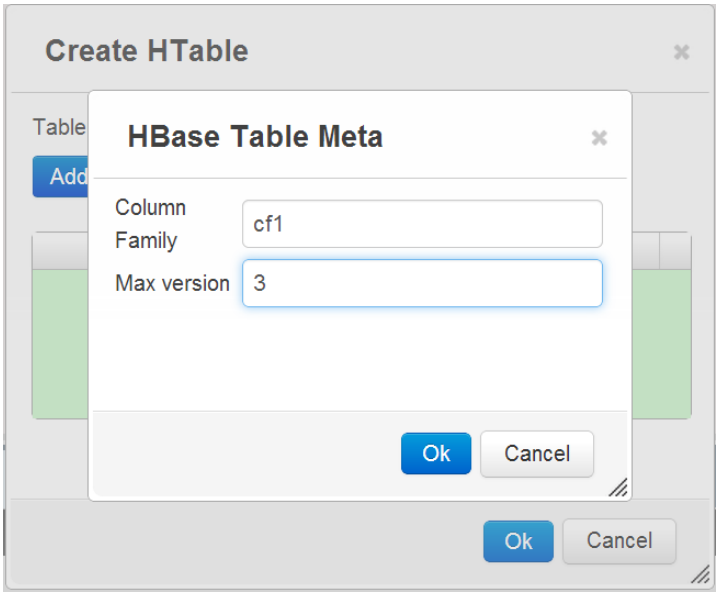


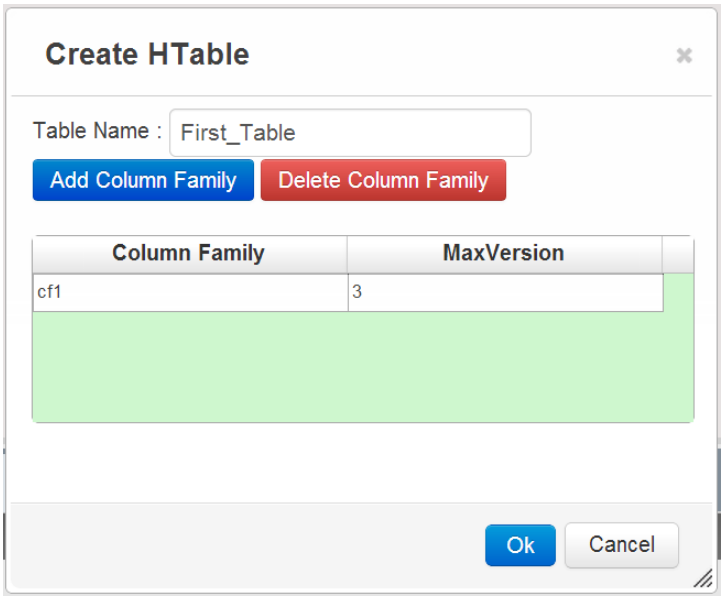


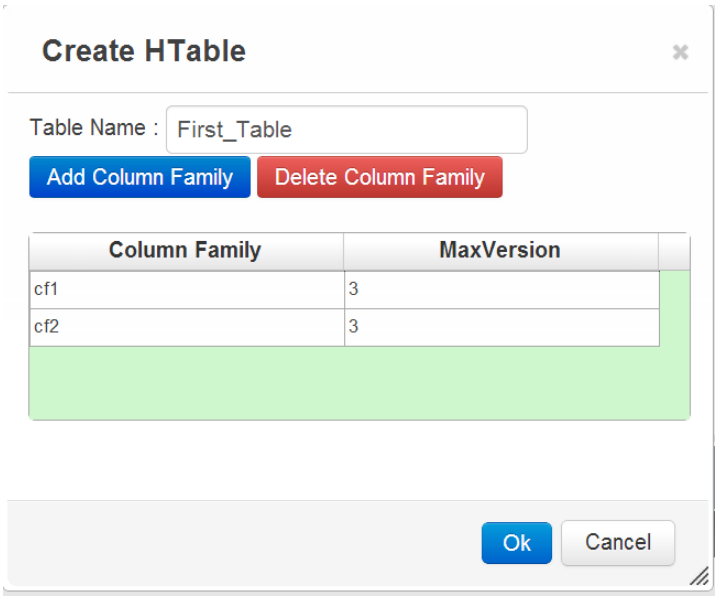
4. Input a table name.



5. At least input a column family. Of course you can delete column family





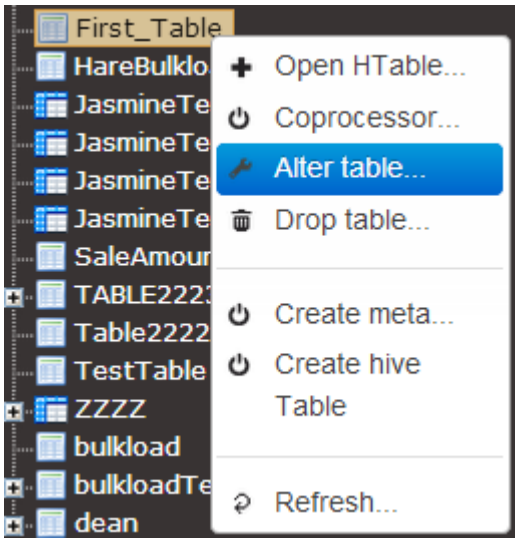


6. Press “OK”
7. You can check the table you just created in the table list

Alter Table

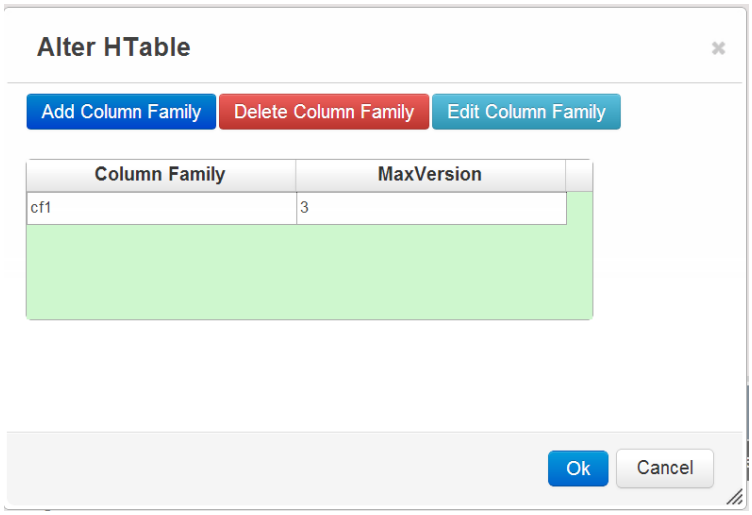
1. Select an connection to connect to HBase (Reference: ”Connect to HBase”)

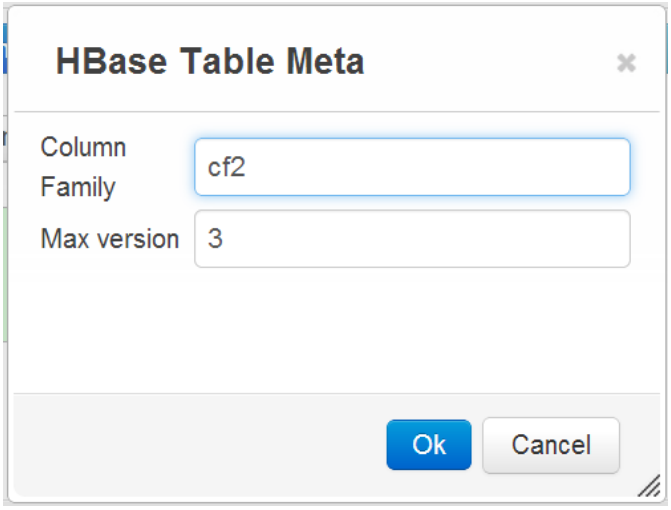
2. Right click on the HTable you want to alter and system will pop up a menu.
3. Please select “Alter Table”, and it will pop up a windows.

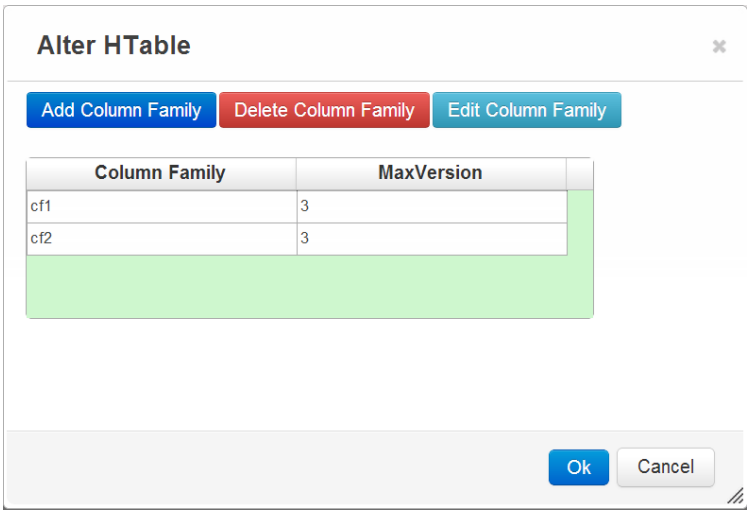


4. You can add column family or delete column family here.
   However, please be careful while you are deleting column family.
   Deleting column family will lose your data which is belong to the column family.

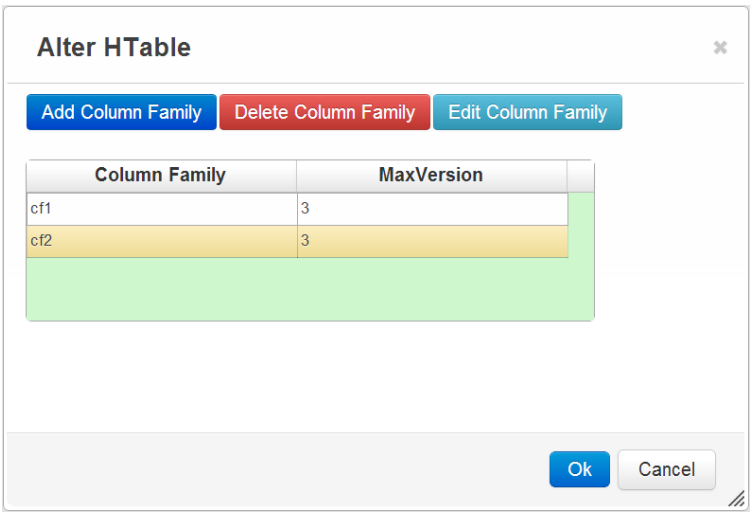
Add Column Family

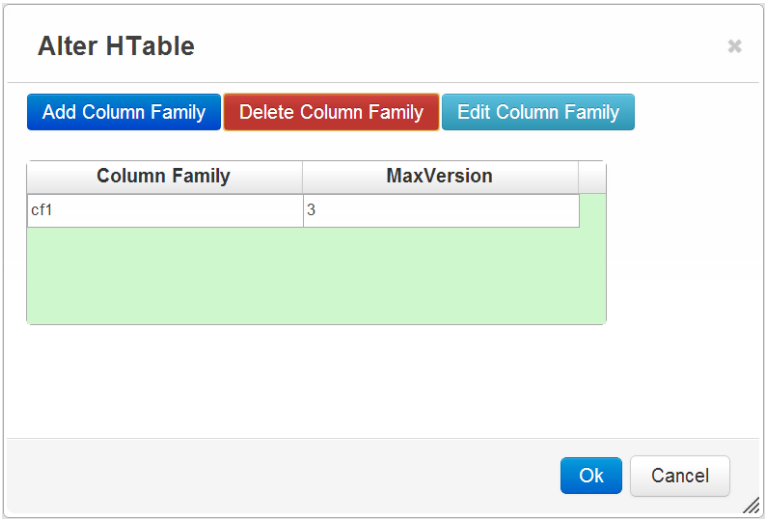






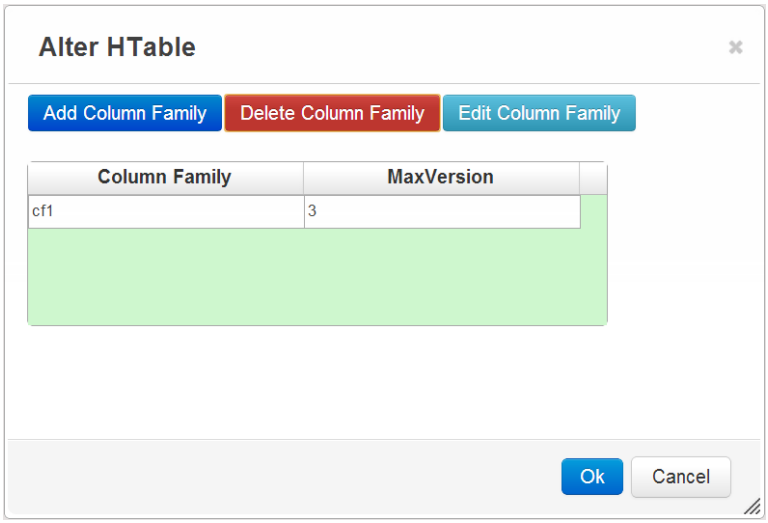
Drop Column Family



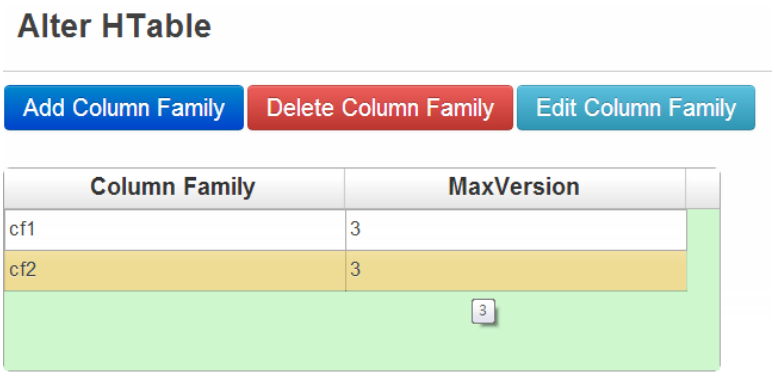


Edit Column Family's Properties

1. Please open the Alter Table Windows (Reference: "Alter Table")



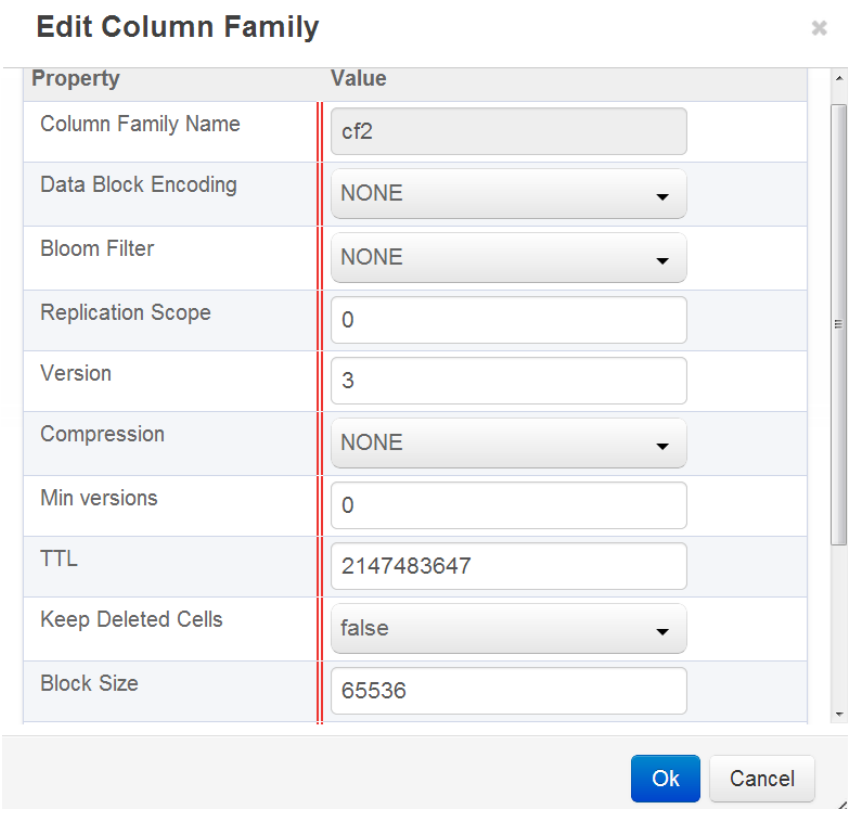
2. Select a Column Family



3. Click 

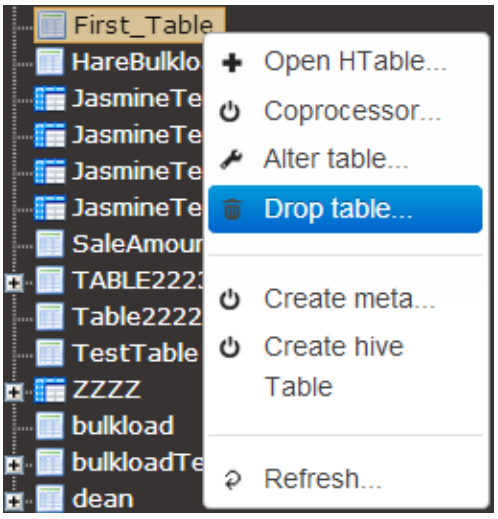
, and system will pop up a window.

4. You can modify most properties.



Drop Table

1. Select a connection and connect to HBase. (Reference "Connect to HBase")
2. Right click on the connection icon and it will pop up a menu.



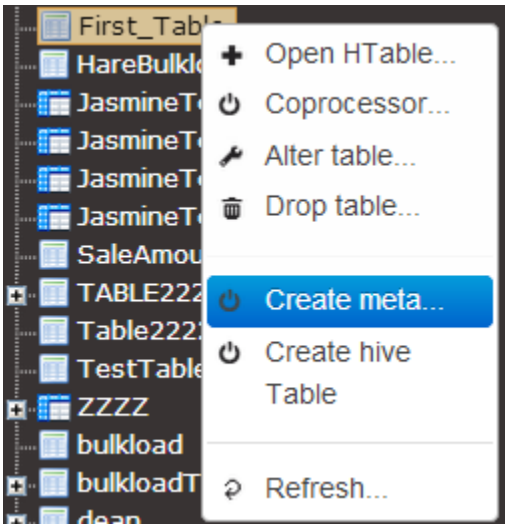
3. Please select "Drop Table", and it will pop up a message.

Table metadata

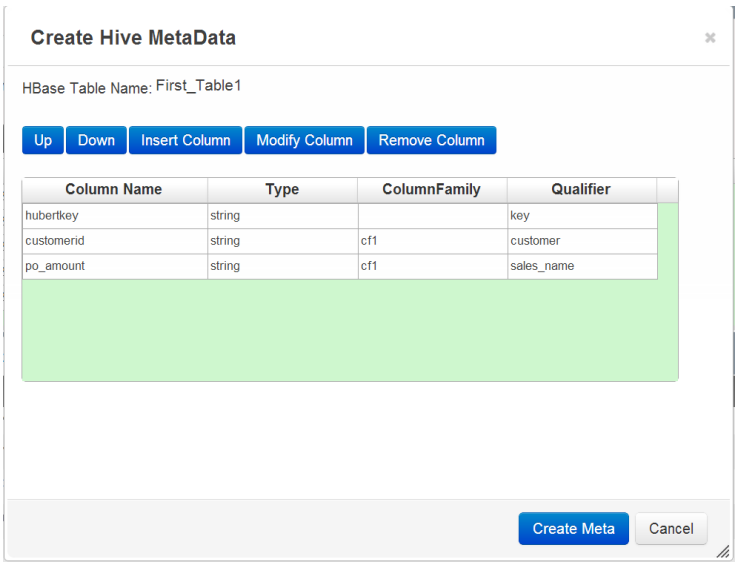
As we know, there is no schema in HBase. We only define table name and column families while we are creating HBase table. Furthermore, there is no data type in HBase. Everything is stored as a byte [] in HBase. However if we don't know the correct type, we can't cast the byte [] to the correct type and we can't read it correctly. So, we provide some metadata manage tool for you.

Create Meta

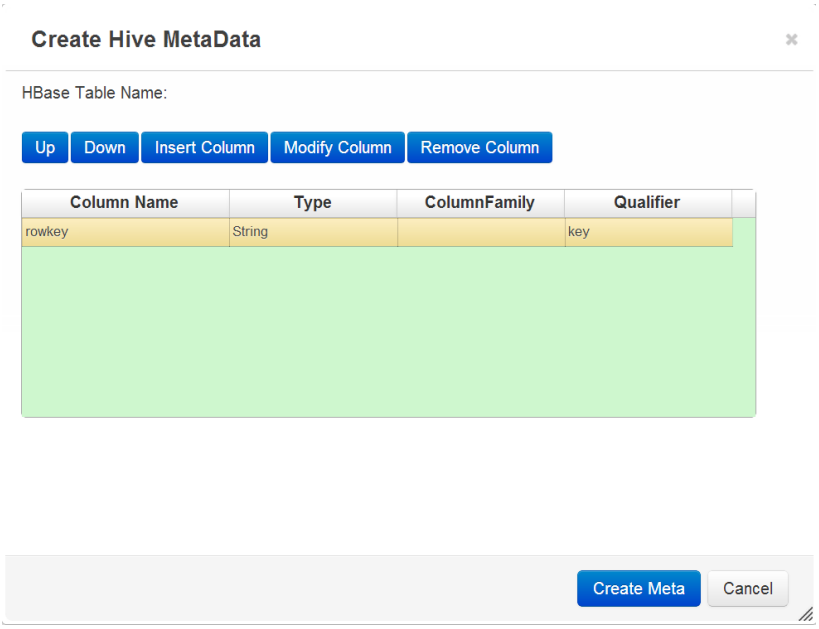
1. Select a connection and connect to HBase. (Reference "Connect to HBase")
2. Right click on the table which you want to create meta and it will pop up a menu.



3. Click "Create meta", and it will pop up a windows.
   If the table has some hive tables already, system will give you a suggestion according to those hive tables.



Screenshot of Create Meta with suggestion

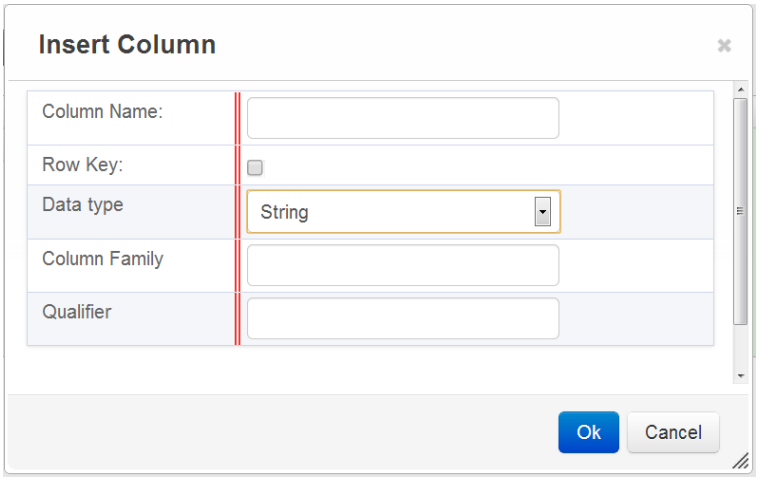


Screenshot of Create Meta without suggestion

4. Rowkey is a must row, don't delete it.
5. Insert Column

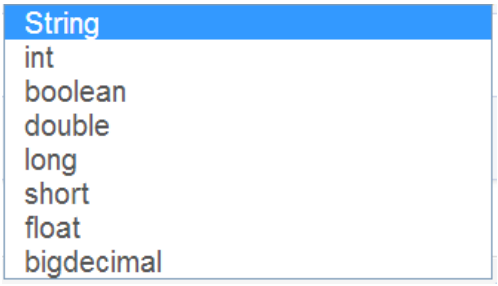
   - Click 

   - System will pop up a window



   - Column Name: Any name you want
   - Data Type: this type must the same with the type while you input it to HBase.

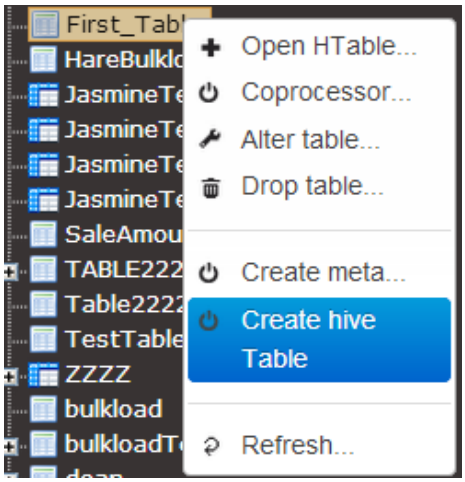
- We support these types as below.



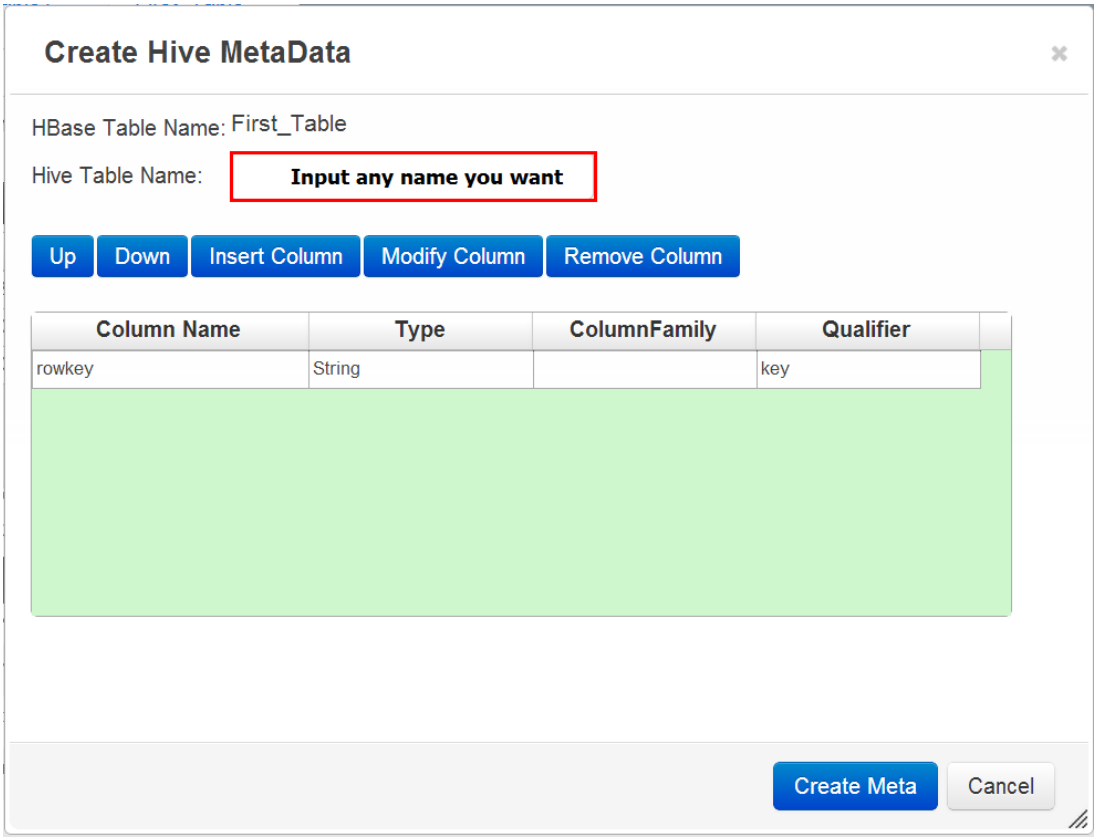
- Column Family and Qualifier must be the same with your HBase column which you want to map. Be careful it's case sensitive.

Create Hive Table

1. Select a connection and connect to HBase. (Reference "Connect to HBase")
2. Right click on the table which you want to create hive table for it and system will pop up a menu.
3. Click "Create hive table", and system will pop up a windows.



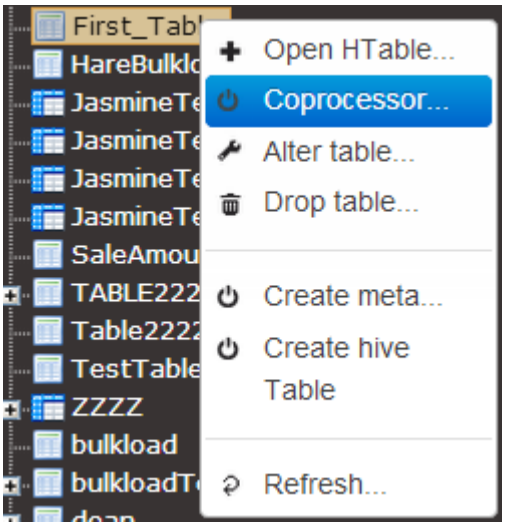
4. Input a table name
5. And you can insert, modify column and delete column (Reference "Create Meta")



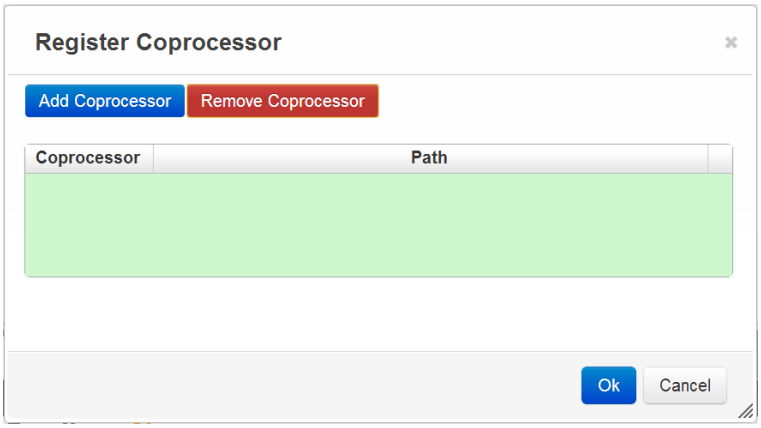
Query Data

Register a coprocessor

1. According to the HBase, we must register coprocessor for the table first before using it.
2. Select a connection and connect to HBase. (Reference "Connect to HBase")
3. Right click on the table which you want to register for it and system will pop up a menu.

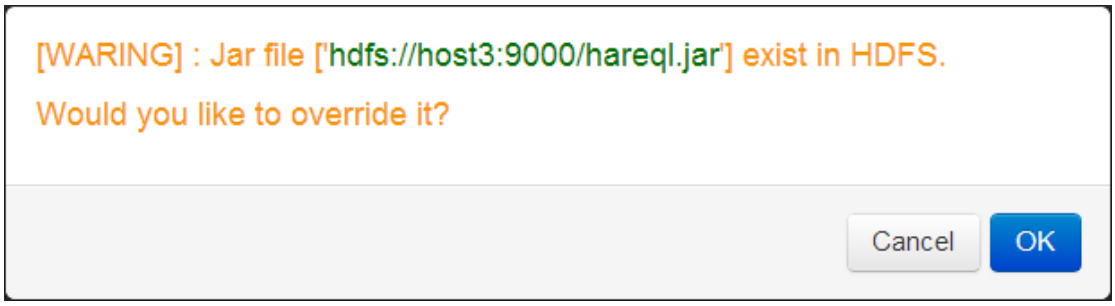


4. Click “Coprocessor” and system will pop up a window.

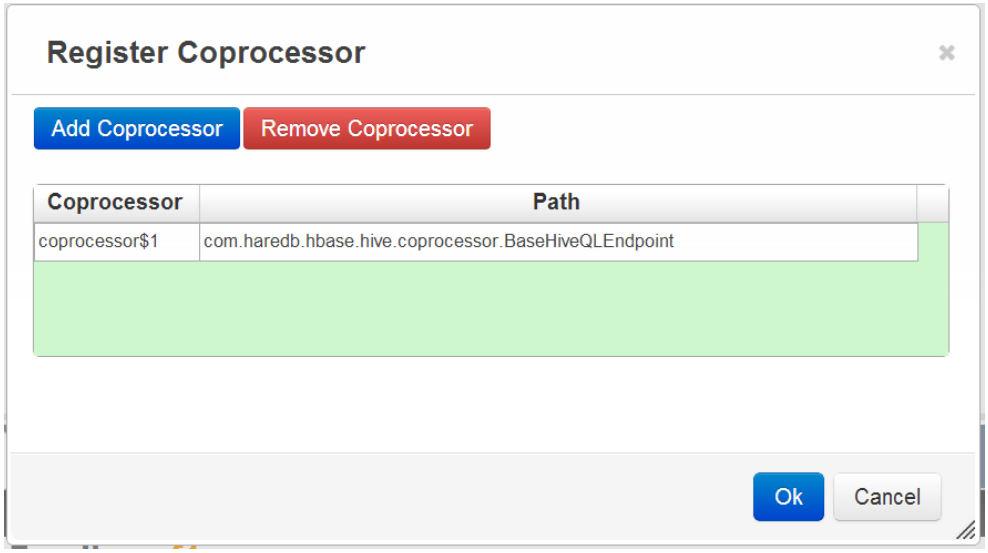


5. Click “Add Coprocessor” and system will pop up another window.
6. Please select “BaseHiveQLEndpoint” and fill the “Coprocessor Path”.
   Example: hdfs://host3:9000/hareql.jar
   PS: If you installed other version before, please reinstall and overwrite the coprocessor jar.

If the name which you gave has already existed, system will pop up an message as below. Usually, we override it.



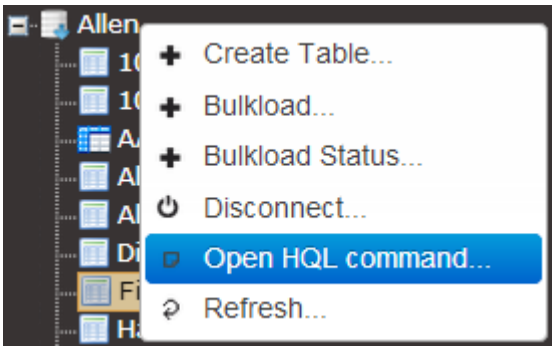
7. Press OK, here is the success screenshot as below.



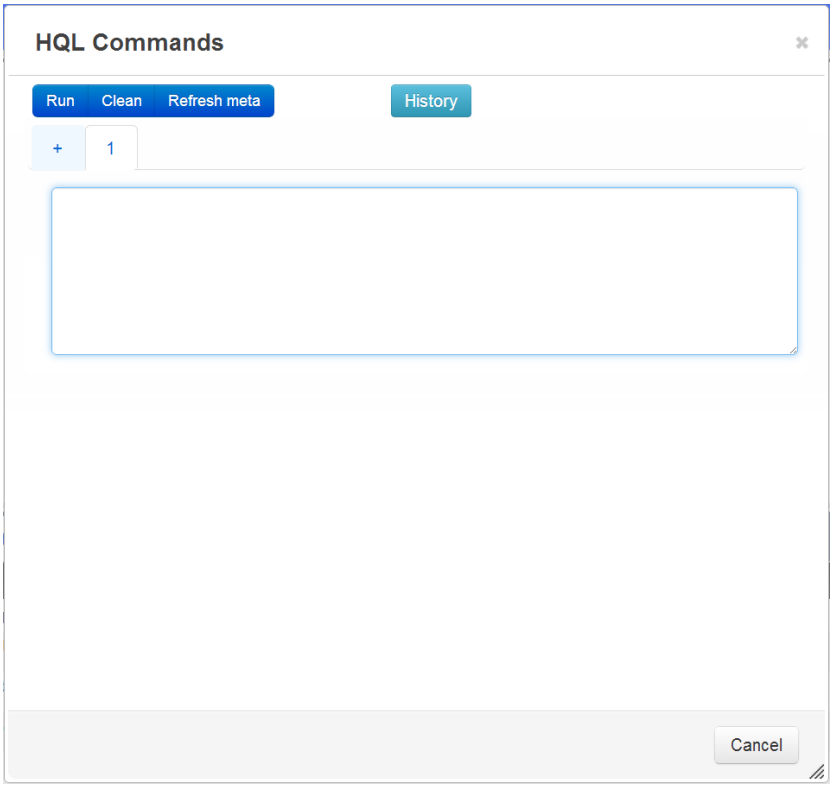
Open HQL Command (HareQL Command)

1. Register a coprocessor
   Our HQL (HareQL Command) is implemented by HBase coprocessor.
   According to the HBase, we must register coprocessor first before using it.
   (Reference "Register Coprocessor")
   PS: If you are using previous version, please remove coprocessor first, and re-register again.

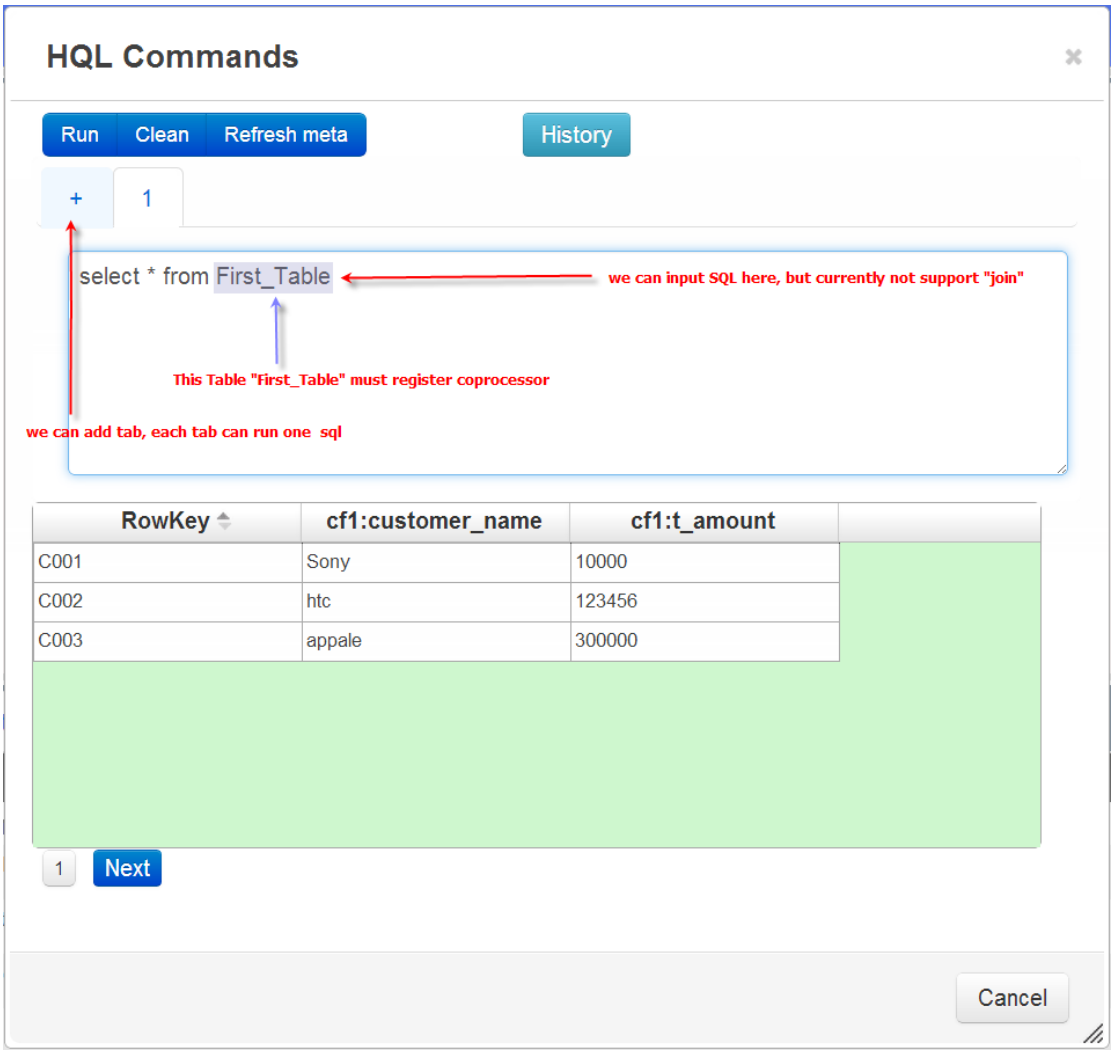
2. Select a connection and connect to HBase. (Reference "Connect to HBase")
3. Right click on the connection icon and system will pop up a menu.



4. Click "Open HQL command" and system will pop up a window like fellow.



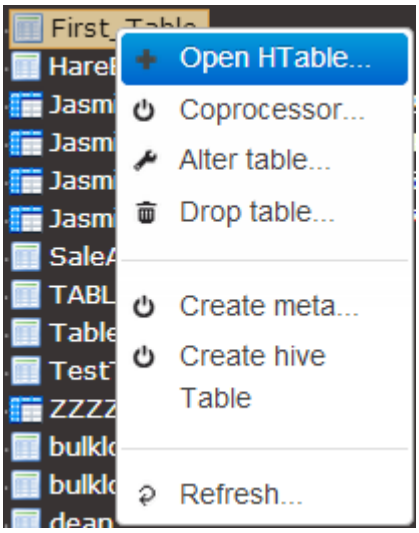
5. You can use sql here to query data, but just “Select” no “Insert、Update、Delete” and “DDL”.



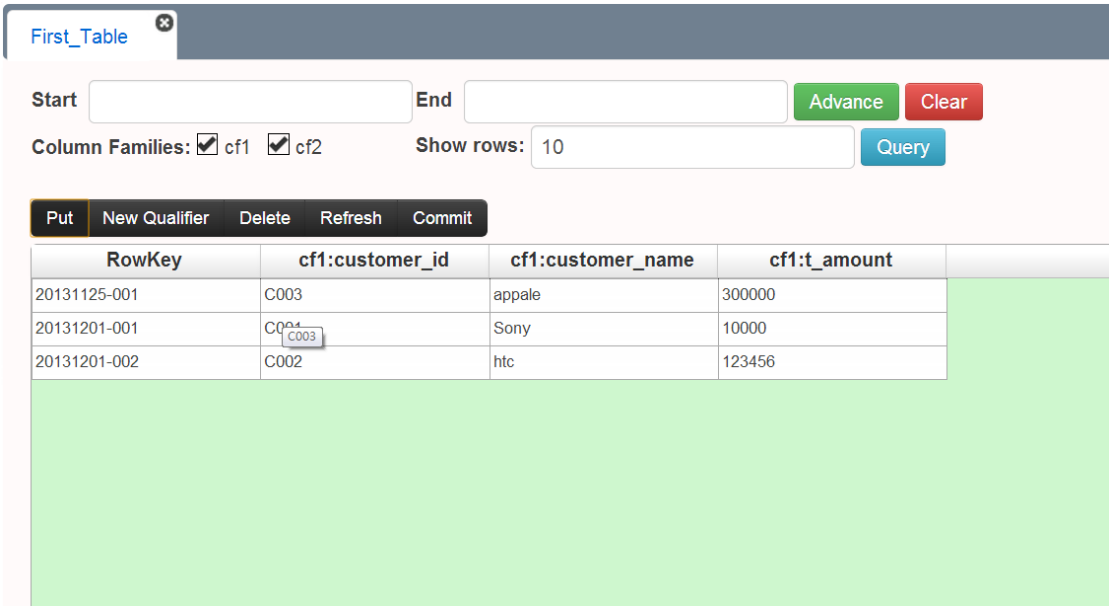
6. After you input the sql, press the RUN.

Open HTable

1. Select a connection and connect to HBase. (Reference “Connect to HBase”)
2. Double click the table or right click on the table and system will pop up a menu. Select “OpenHTable”.

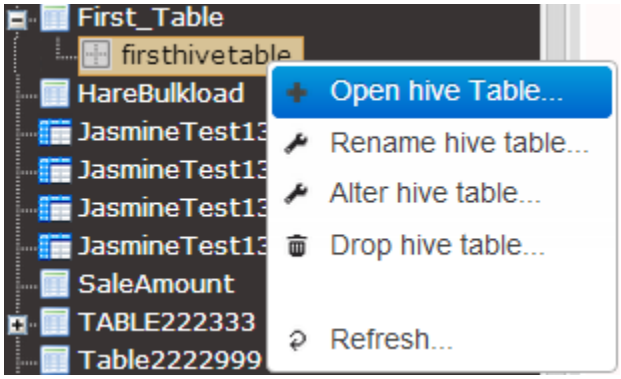


3. You can click Query. Than data will show up as below.

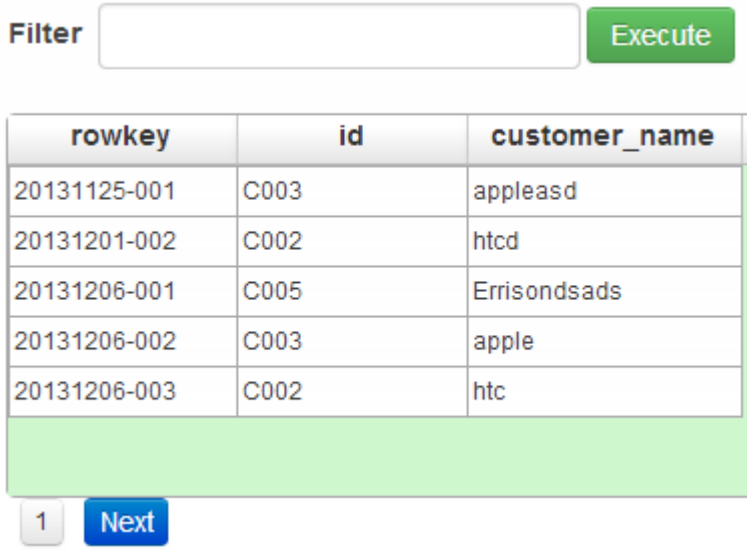


Open Hive Table

1. Select a connection and connect to HBase. (Reference "Connect to HBase")
2. Double click the Hive table or right click on the table and system will pop up a menu and select "Open hive Table".



3. System will show the hive table with data in the right side as fellow.

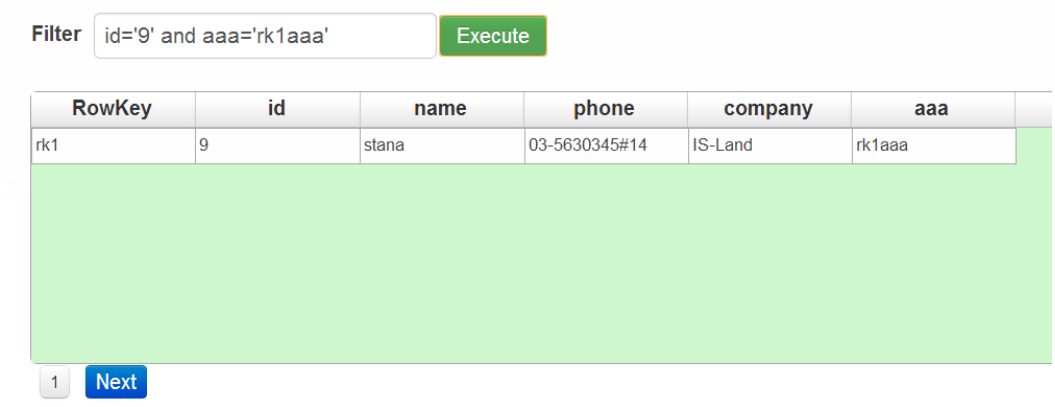


4. You can input some criteria in filter area than click “Execute”. You will get the result.

   PS: This table must register our coprocessor first, or this function will not work.

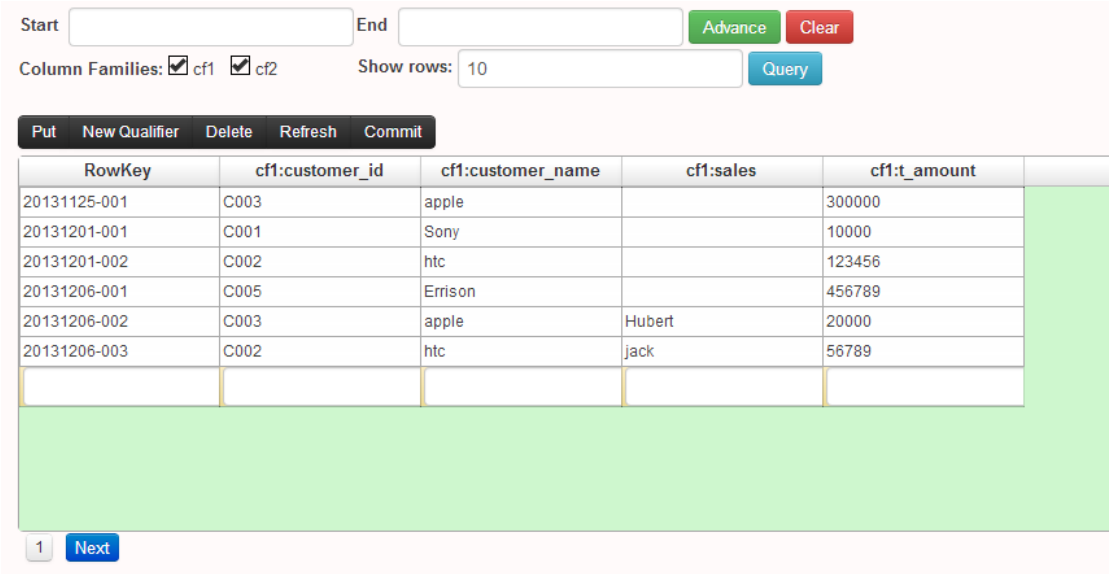
   (Reference: Register a coprocessor)

   If you are using previous version, please remove coprocessor first, and re-register again.



Data Operation

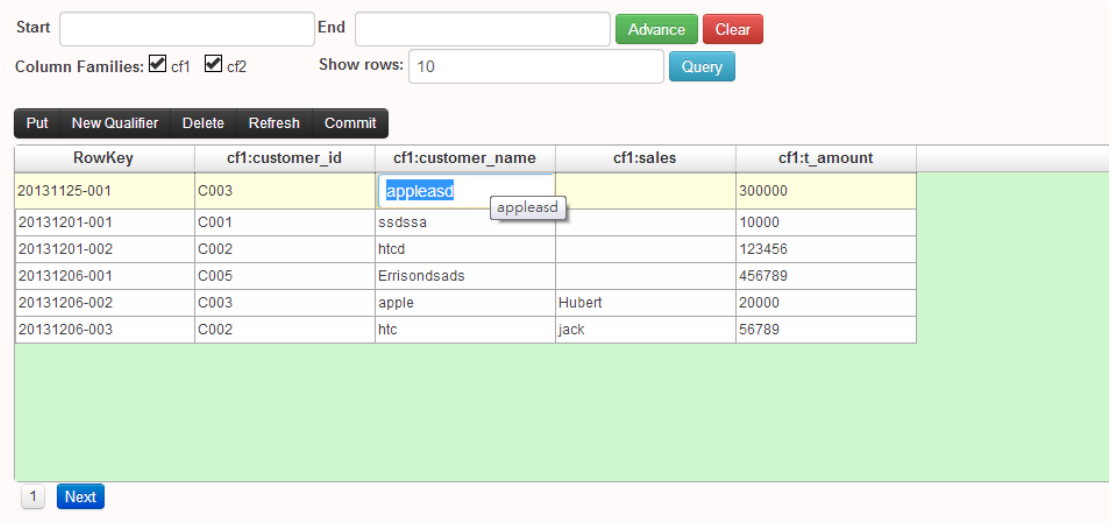
Put data



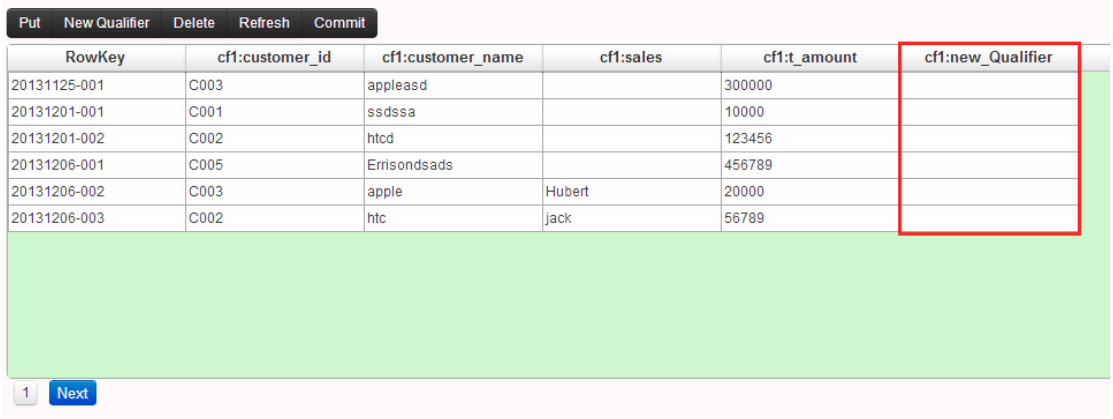
After input data to the cell, please remember to click “Enter”.

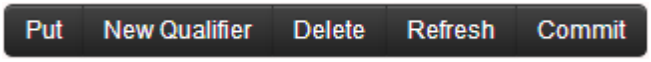
Currently, if you click your mouse in other location, maybe you will lose the values which you just enter. (We are working on this bug…)

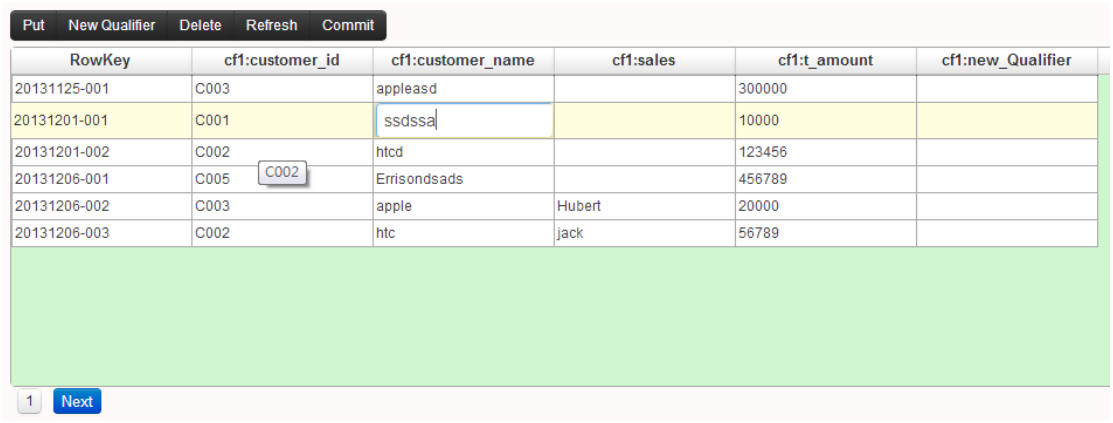
Modify data

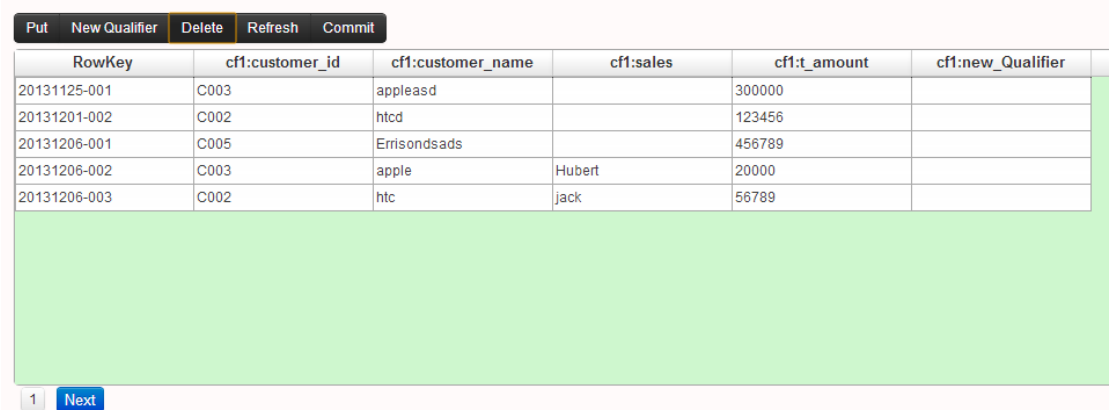


New Qualifier





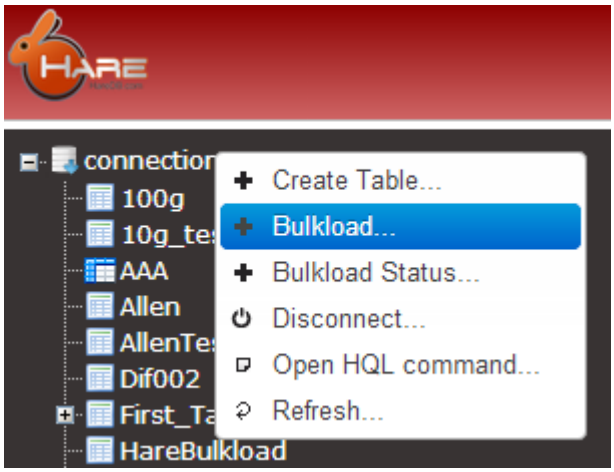




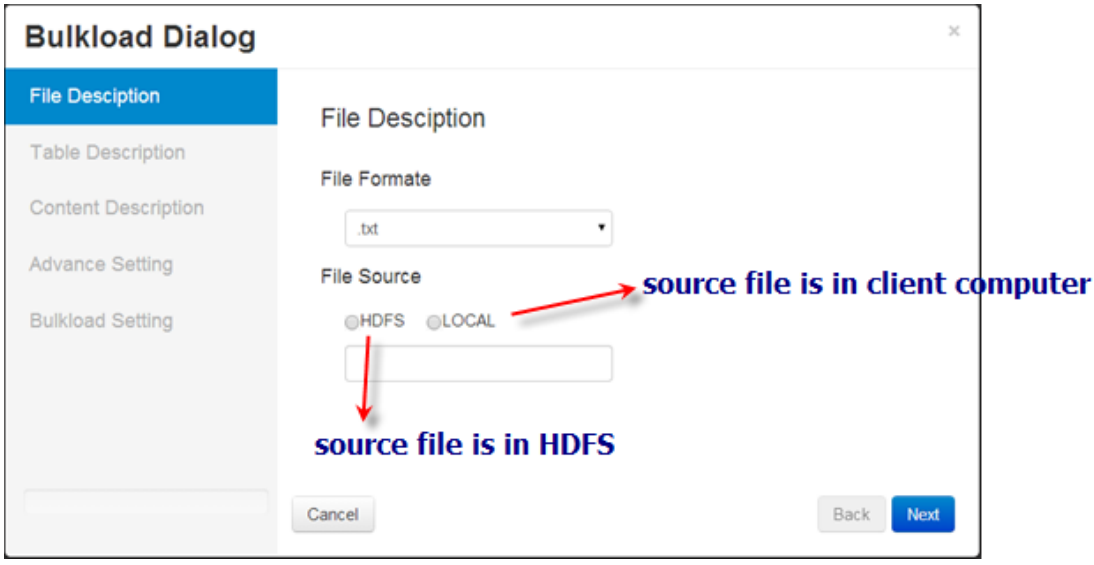
Bulkload (only in Full Function Version)

Bulkload is our new function. It's a fast way to import big data from a structure file.

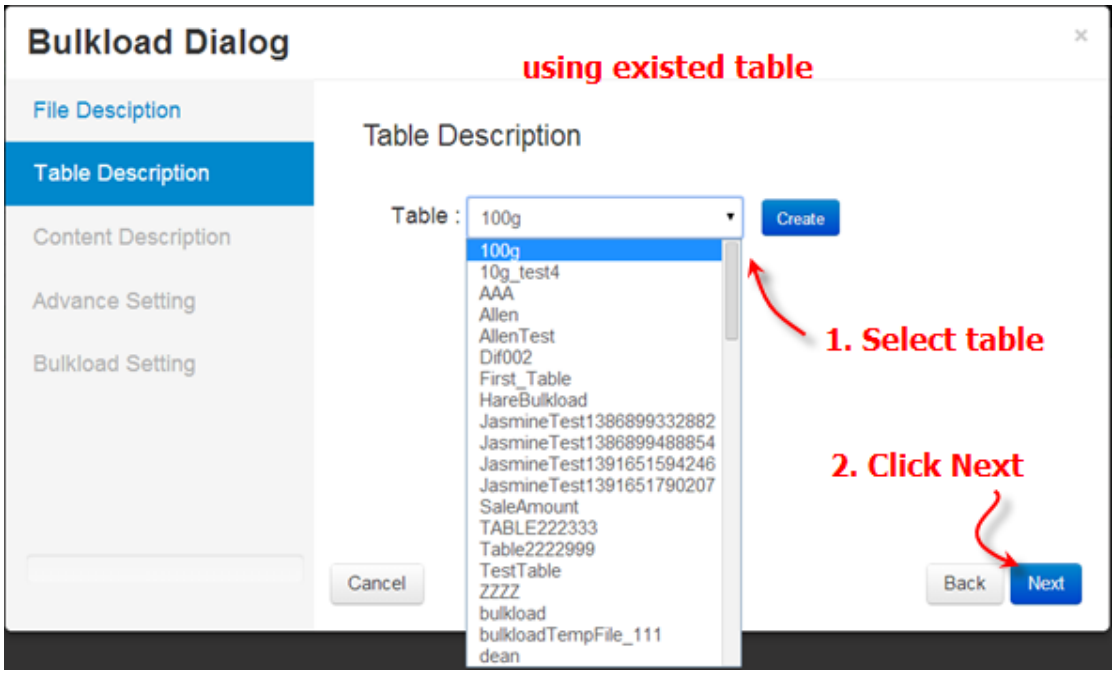
1. Right click on the connection icon and system will pop up a menu.
2. Select "Bulkload", and system will pop up a dialog.

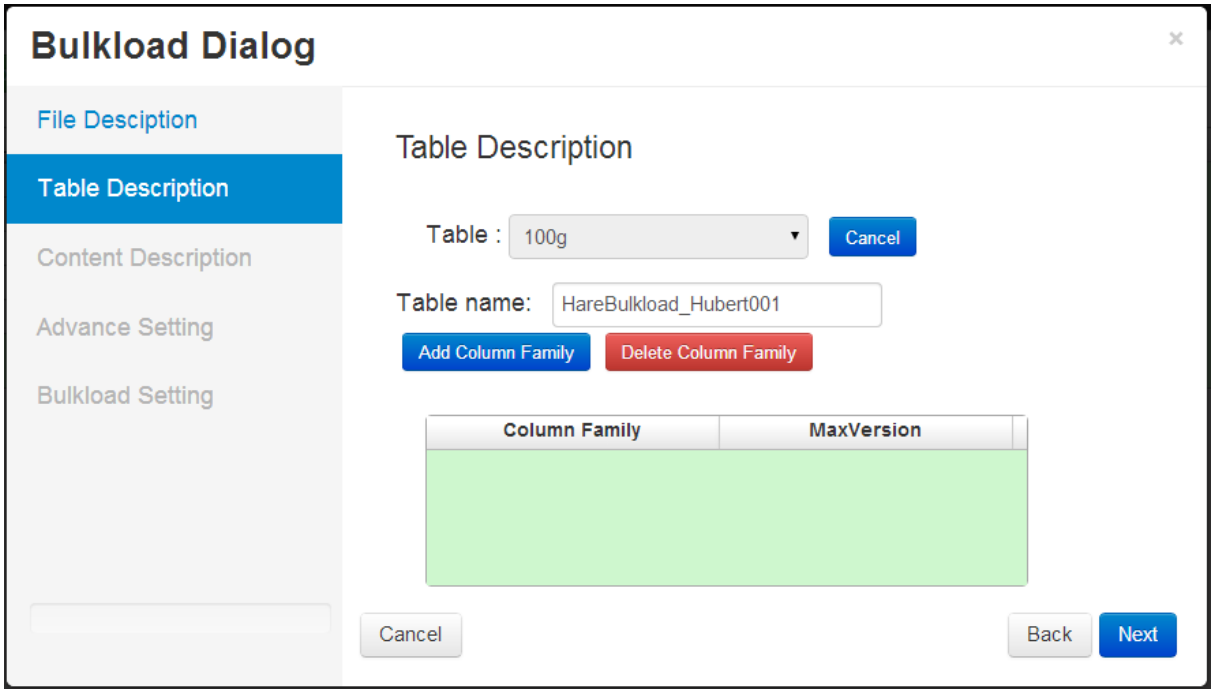


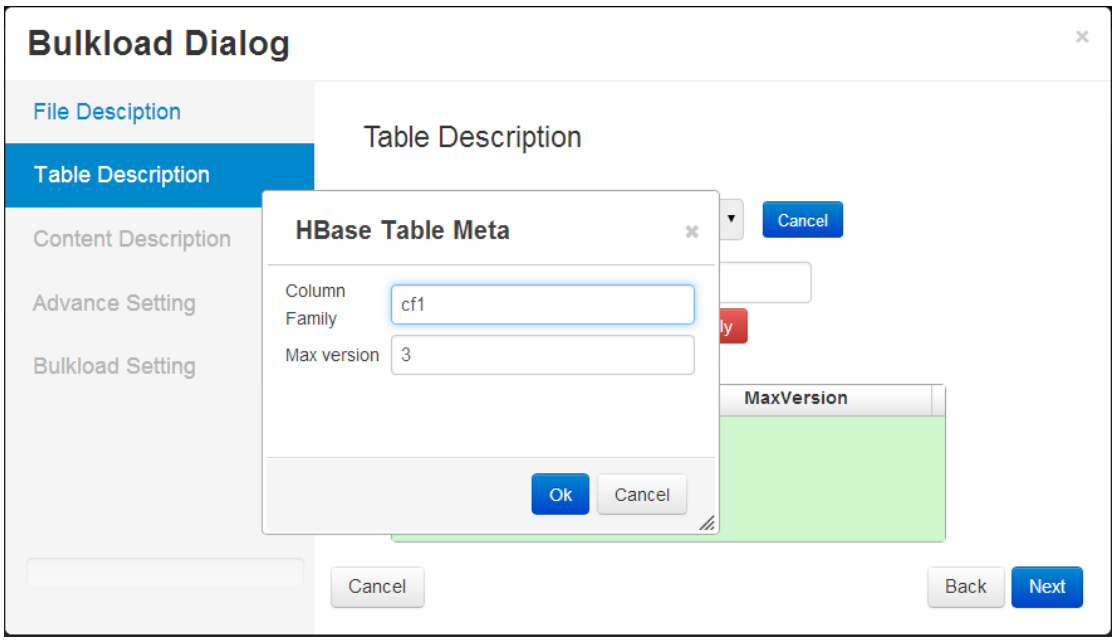
3. There are 5 steps to import data in the bulkload function.
4. First, it's File Description . Currently we only support text file, but we don't care the extension of the source file.
5. Then you have to tell system where is the source file. You can upload the file to HDFS by yourself, then you should select "HDFS". If your source file is in your computer, then you should select "LOCAL". System will upload the file for you, and start bulkload automatically.



6. Second, It's "Table Description". This table is the target table. In other words, this table is the HBase table which you want to put data from the source file.
7. You can select an existed table or create a new table.

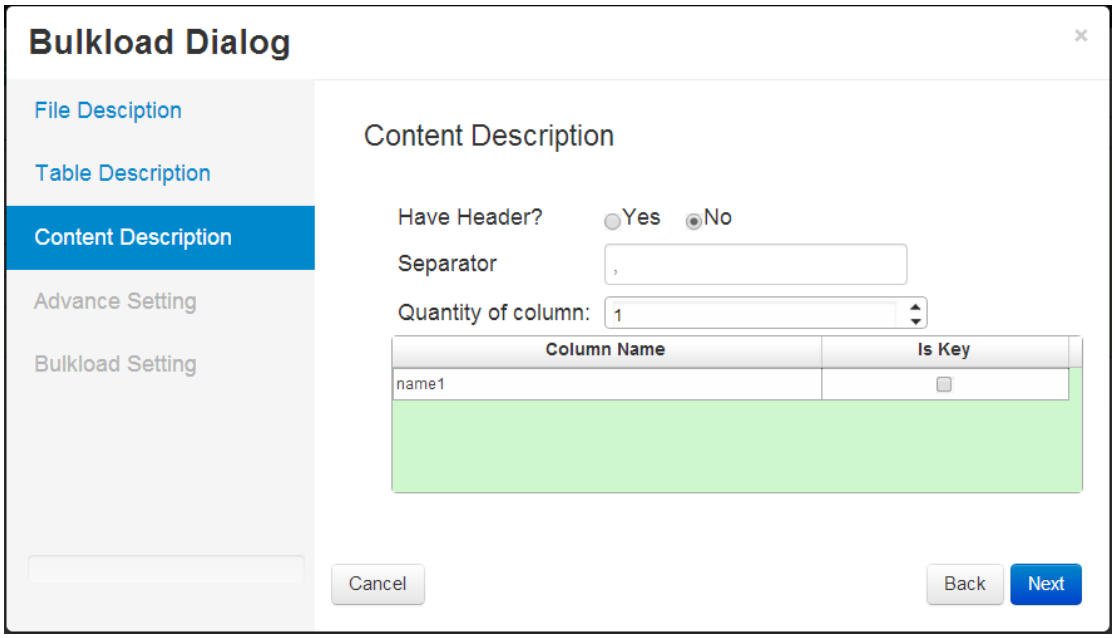


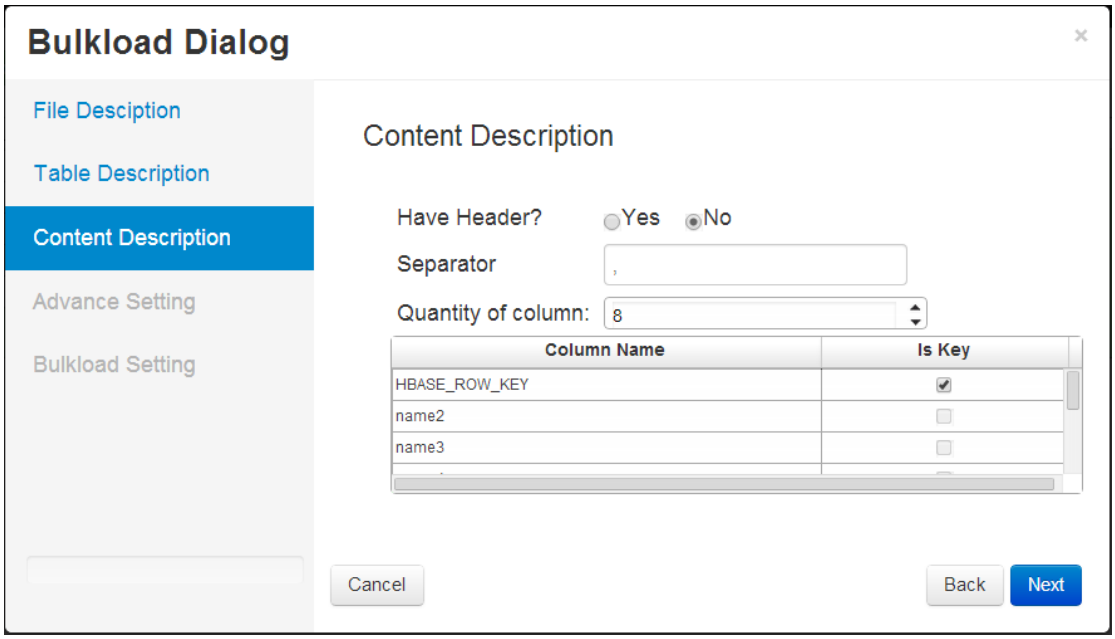




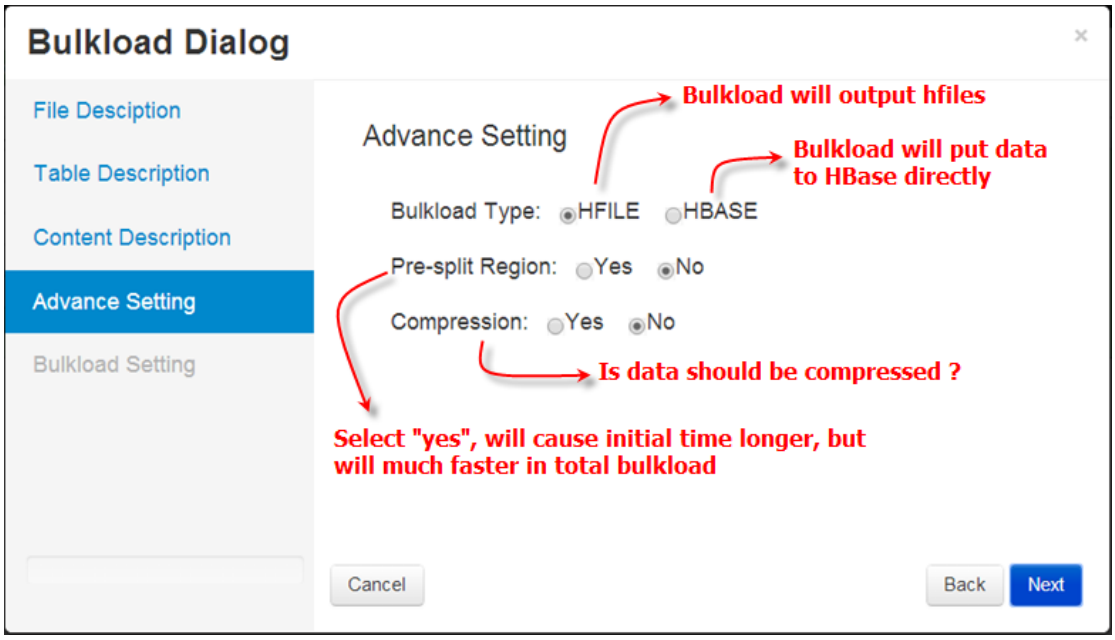
8. Third, it's "Content Description". System has to understand the format of source file.
   Currently, the format of source file should be the image as below.
   Each row is a row records.
   Separator is "," and there are 8 column in each records.

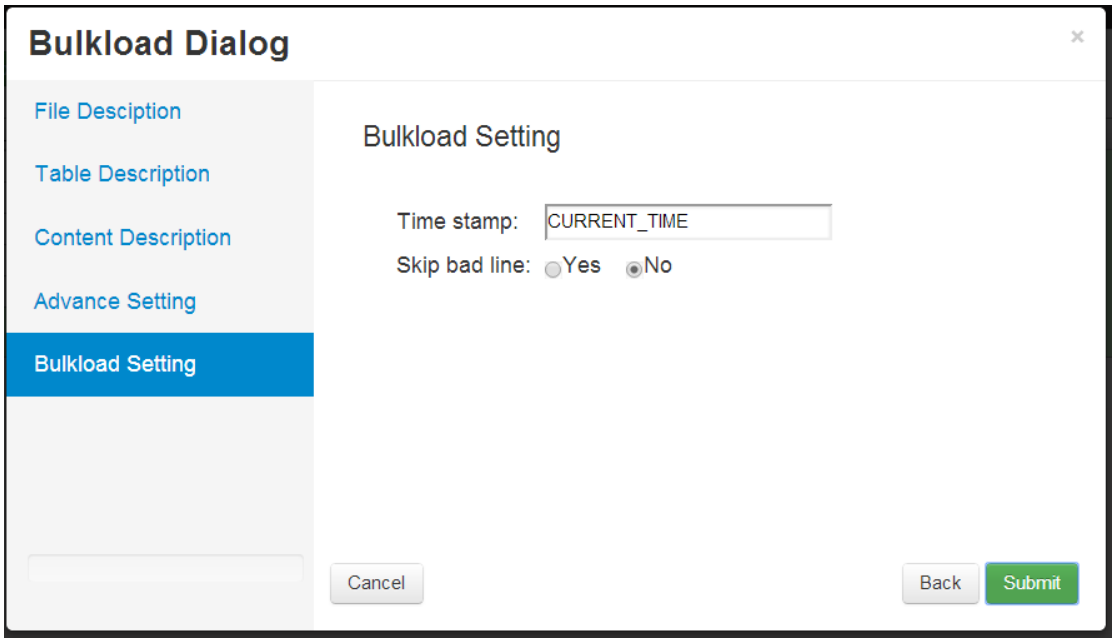
```
7741,6735,9361,9526,645,3110,6853,5708
890,8026,6707,1078,9674,3908,7828,5973
1859,3738,2302,3990,3123,1954,9848,1270
5581,1106,8571,7407,447,8158,3821,6962
4652,8206,8666,8148,4467,8205,4936,7577
790,4918,4633,9029,9862,5384,7427,3448
9699,1566,1041,4845,8858,9666,1230,4414
6498,1210,6513,17,2688,4914,2535,4644
3345,8697,5694,13,4943,6541,7489,7728
5067,5275,1274,964,4761,1634,6490,3285
6940,7442,2677,4923,4643,5568,8054,5542
8584,5184,571,257,2663,6008,5193,3493
```

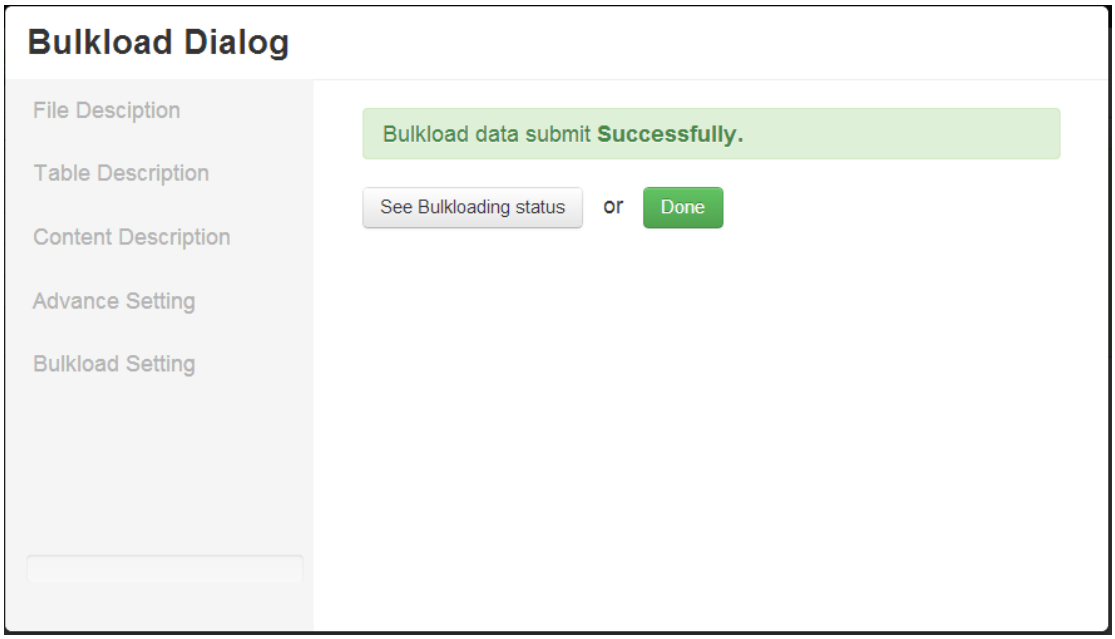




9. Fourth, it's "Advance Setting". Actually, you can keep it default.






Bulkload Dialog
File Desciption
Table Description
Content Description
Advance Setting
Bulkload Setting
Bulkload data submit Successfully.
See Bulkloading status
or
Done

Appendix A -- Introduction of HareQL

What is HareQL?

HareQL is hiveQL+HBase Coprocessor.
We can use hiveQL which is a subset of ANSI SQL to query data.

Why Hare?

The reason is high speed and easy install.
HareQL is using hiveQL, but 3~80X faster than hive.
Actually, you don't have to do anything about installation.
All you have to do is install this client tool and register the coprocessor "BaseHiveQLEndpoint".

What do we support right now?

Syntax

Select

```
SELECT [ ALL | DISTINCT] select_expr as alias, select_expr, ...
FROM table_reference
[WHERE where_condition]
[GROUP BY col_list [HAVING condition]]
[SORT BY | ORDER BY col_list]
[LIMIT number]
```

Union

```
select_statement UNION ALL select_statement UNION ALL
select_statement ...
```

UNION is used to combine the result from multiple SELECT statements into a single result set. We currently only support “UNION ALL”.
Currently, we are not support sub-query. So we don’t support the query like this:

```
SELECT *
FROM (
  select_statement
  UNION ALL
  select_statement
)
```

Operators

Logical Operators

AND、OR、&&、 || 、NOT A

Relational Operators

A = B、A <=> B
A <> B、A != B
A < B、A <= B、A > B、A >= B
A IS NULL、A IS NOT NULL
A LIKE B
A RLIKE B
A REGEXP B

Arithmetic Operators

The following operators support various common arithmetic operations on the operands. A and B are columns in HBase Table. No matter what type of A and B are number (like integer or float…) or string in Hive metastore, their value in HBase cell must be the same type with hive meta. We all return String types; if any of the operands are NULL, then the result is also NULL.

A + B, A – B ,
A * B, A / B,
A % B, A & B,
A | B, A ^ B,
~A

Functions

Aggregate Functions

count(*), count(expr), count(DISTINCT expr[, expr_.])
sum(col), sum(DISTINCT col)
avg(col), avg(DISTINCT col)
min(col), max(col)