



# Big Data

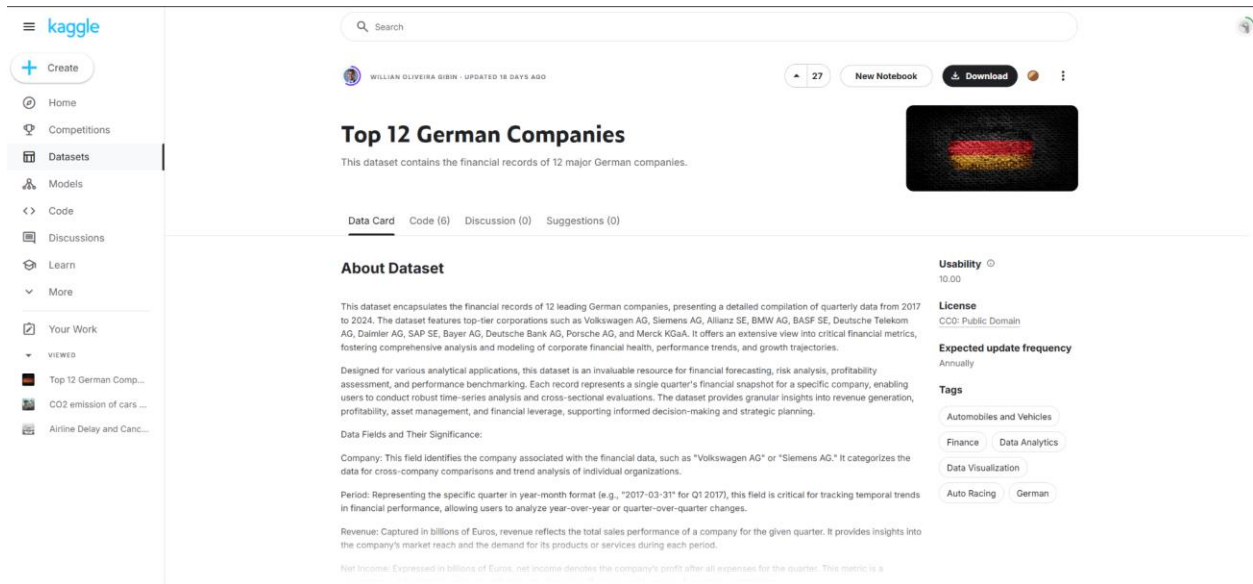
06/12/2024

Jon Fernandez de Gorostiza

<b>Dataset</b>	<b>2</b>
<b>Cluster</b>	<b>4</b>
<b>HUE</b>	<b>5</b>
<b>Manual de CRUD</b>	<b>6</b>
<b>HIVE</b>	<b>7</b>
CRUD	9
Create	9
Read	10
Update	10
Delete	11
<b>HBASE</b>	<b>12</b>
CRUD	14
Create	14
Read	15
Update	15
Delete	16

## Dataset

Descargamos el dataset desde Kaggle, en nuestro caso son datos financieros de 12 empresas alemanas.



The screenshot shows the Kaggle dataset page for "Top 12 German Companies". The page is titled "Top 12 German Companies" and includes a description: "This dataset contains the financial records of 12 major German companies." The dataset is created by WILLIAN OLIVEIRA BIBIN, updated 16 days ago, and has 27 versions. It is available for download and has a new notebook created. The page also features a "Data Card" section with an "About Dataset" tab selected. The "About Dataset" section describes the dataset as a compilation of quarterly data from 2017 to 2024 for 12 leading German companies, including Volkswagen AG, Siemens AG, Allianz SE, BMW AG, BASF SE, Deutsche Telekom AG, Daimler AG, SAP SE, Bayer AG, Deutsche Bank AG, Porsche AG, and Merck KGaA. It mentions that the dataset is useful for financial forecasting, risk analysis, profitability assessment, and performance benchmarking. The "Data Fields and Their Significance" section lists fields like Company, Period, Revenue, and Net Income, providing brief descriptions for each. On the right side, there is a "Usability" section showing a score of 10.00, a "License" section indicating "CC0: Public Domain", an "Expected update frequency" of "Annually", and a "Tags" section with categories like "Automobiles and Vehicles", "Finance", "Data Analytics", "Data Visualization", "Auto Racing", and "German".

Al cargar el archivo en excel vemos que hay algunas columnas que como contienen comas, al hacer la división sobre ellas los datos se pasan de columnas. Para solucionar este problema vamos a eliminar la última columna ya que esta es la causante de estos problemas, además de que también contiene el carácter % que nos puede traer mas problemas aun al importar los datos después. Como vamos a crear nosotros la tabla a mano la primera fila, que es la de los nombres de las columnas, la vamos a borrar.



## Cluster

Creamos un clúster de hadoop de la manera que los hemos detallado en los anteriores ejercicios y subimos el archivo csv que hemos descargado de Kaggle al entorno de hadoop.

```
scp -i /home/iabd/Descargas/ClavesHadoop.pem
/home/iabd/Descargas/Top_12_German_Companies.csv hadoop@ec2-54-158-105-
211.compute-1.amazonaws.com:/home/hadoop
```

```
(base) iabd@dm2-13:~$ scp -i /home/iabd/Descargas/ClavesHadoop.pem /home/iabd/De
scargas/Top_12_German_Companies.csv hadoop@ec2-54-172-174-211.compute-1.amazona
ws.com:/home/hadoop
Top_12_German_Companies.csv          100%  49KB 165.2KB/s   00:00
(base) iabd@dm2-13:~$
```

Tenemos que subirlo ahora del entorno al gestor de archivos de hadoop, así que creamos una carpeta y subimos el archivo.

```
hdfs dfs -mkdir Proyecto
```

```
hdfs dfs -put Top_12_German_Companies.csv Proyecto
```

```
[hadoop@ip-172-31-85-225 ~]$ hdfs dfs -mkdir Proyecto
[hadoop@ip-172-31-85-225 ~]$ hdfs dfs -put Top_12_German_Companies.csv Proyecto
[hadoop@ip-172-31-85-225 ~]$ hdfs dfs -ls Proyecto
Found 1 items
-rw-r--r--  1 hadoop hdfsadmingroup    47012 2024-12-03 18:16 Proyecto/Top_12
German_Companies.csv
[hadoop@ip-172-31-85-225 ~]$
```

## HUE

Ahora venimos a aplicaciones del clúster de hadoop en AWS y pinchamos en Tonalidad.

**IU de la aplicación en el nodo principal**  
 Estas requieren que el túnel de SSH esté habilitado.

Habilitar una conexión SSH

Aplicación	URL de la IU
Administrador de recursos	<a href="http://ec2-54-158-105-211.compute-1.amazonaws.com:8088/">http://ec2-54-158-105-211.compute-1.amazonaws.com:8088/</a>
HBase	<a href="http://ec2-54-158-105-211.compute-1.amazonaws.com:16010/">http://ec2-54-158-105-211.compute-1.amazonaws.com:16010/</a>
Nodo del nombre de HDFS	<a href="http://ec2-54-158-105-211.compute-1.amazonaws.com:9870/">http://ec2-54-158-105-211.compute-1.amazonaws.com:9870/</a>
Tonalidad	<a href="http://ec2-54-158-105-211.compute-1.amazonaws.com:8888/">http://ec2-54-158-105-211.compute-1.amazonaws.com:8888/</a>

Nos pide crear un usuario.

HUE

User: Hadoop

Password: Hadoop1@

Query. Explore. Repeat.

Since this is your first time logging in, pick any username and password. Be sure to remember these, as **they will become your Hue superuser credentials.**

The password must be at least 8 characters long, and must contain both uppercase and lowercase letters, at least one number, and at least one special character.

Hadoop

.....|

The password must be at least 8 characters long, and must contain both uppercase and lowercase letters, at least one number, and at least one special character.

Create Account

Accedemos a la interfaz de usuario de HUE donde desde esta podemos acceder a HIVE y HBase.

## Manual de CRUD

Operación	HBase	HIVE
<b>Create</b>	<pre>create 'table_name', 'column_family'  put 'table_name', 'row_key', 'column_family:column_name', 'value'</pre>	<pre>CREATE TABLE table_name (column_name data_type, column_name data_type)  CREATE TABLE new_table AS SELECT * FROM existing_table</pre>
<b>Read</b>	<pre>get 'table_name', 'row_key'  scan 'table_name'</pre>	<pre>SELECT * FROM table_name</pre>
<b>Update</b>	<pre>put 'table_name', 'row_key', 'column_family:column_name', 'new_value'</pre>	<pre>UPDATE table_name SET column_name = new_value WHERE condition</pre>
<b>Delete</b>	<pre>delete 'table_name', 'row_key', 'column_family:column_name'  deleteall 'table_name', 'row_key'</pre>	<pre>DELETE FROM table_name WHERE condition</pre>



## HIVE

Desde la interfaz de usuario de HUE, seleccionamos la BBDD de HIVE y creamos una tabla con desde la línea de comandos el siguiente código:

```
CREATE TABLE Top_12_German_Companies (
```

```
    Company STRING,
```

```
    Period STRING,
```

```
    Revenue DOUBLE,
```

```
    Net_Income DOUBLE,
```

```
    Liabilities DOUBLE,
```

```
    Assets DOUBLE,
```

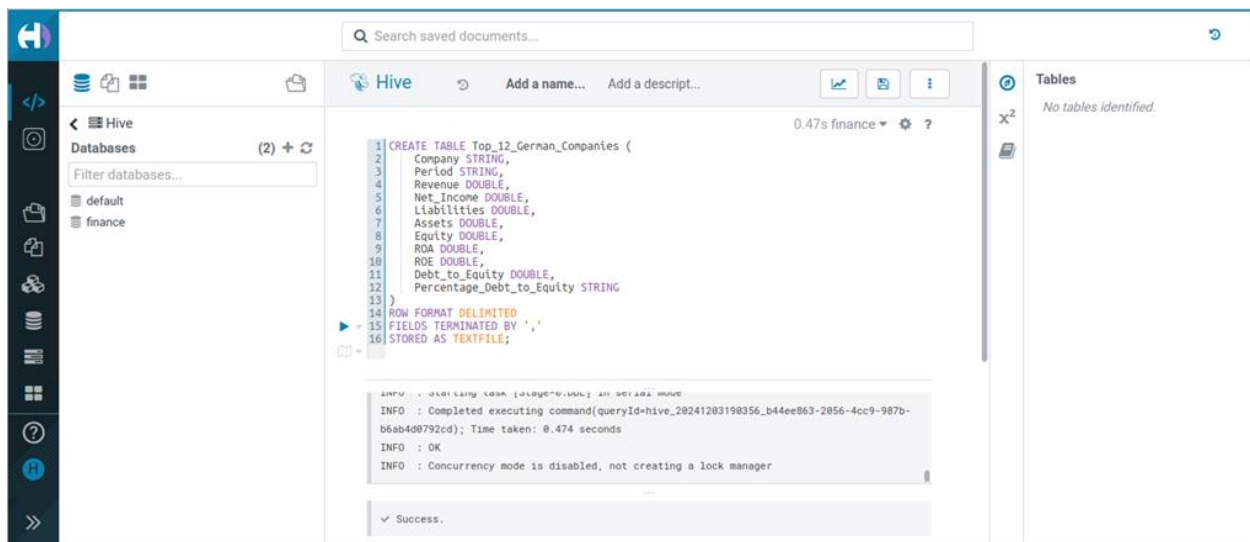
```
    Equity DOUBLE
```

```
)
```

```
ROW FORMAT DELIMITED
```

```
FIELDS TERMINATED BY ','
```

```
STORED AS TEXTFILE;
```





Una vez creada la tabla comprobamos que nos ha añadido todas las columnas y los tipos de datos de cada una correctamente.

The screenshot shows the Hive Table Browser interface. On the left, a sidebar lists the tables in the 'default' database, including 'top\_12\_german\_companies'. The main panel displays the details for this table. The 'PROPERTIES' section indicates it is a table managed and stored in location, created by hadoop on 05/12/2024 2:20 +01:00. The 'STATS' section shows 0 files, 0 rows, and a total size of 0 B. The 'SCHEMA' section shows a table with 10 columns: company (string), period (string), revenue (double), net\_income (double), liabilities (double), assets (double), equity (double), roa (double), roe (double), and debt\_to\_equity (double). Each column has a description 'Add a description...'.

Column (10)	Type	Description	Sample
company	string	Add a description...	
period	string	Add a description...	
revenue	double	Add a description...	
net_income	double	Add a description...	
liabilities	double	Add a description...	
assets	double	Add a description...	
equity	double	Add a description...	
roa	double	Add a description...	
roe	double	Add a description...	
debt_to_equity	double	Add a description...	

Ahora tenemos que cargar los datos que tenemos dentro del hdfs.

```
LOAD DATA INPATH '/user/hadoop/Proyecto/Top_12_German_Companies.csv'
```

```
INTO TABLE top_12_german_companies;
```

The screenshot shows the Hive query execution interface. The query entered is:

```
1 LOAD DATA INPATH '/user/hadoop/Proyecto/Top_12_German_Companies.csv'
2 INTO TABLE top_12_german_companies;
```


The execution status is 'Success'. The logs show the following information:

```
INFO : Completed executing command(queryId=hive_20241203190544_72f19c82-1943-4b29-9d79-31bb3754b3f8); Time taken: 1.384 seconds
INFO : OK
INFO : Concurrency mode is disabled, not creating a lock manager
```

The 'Query History' section shows the query was executed 'hace unos segundos' (a few seconds ago) and was successful. The 'Saved Queries' section shows the query was saved with the name 'CREATE TABLE Top\_12\_German\_Companies (company STRING, period STRING, revenue DOUBLE, net\_income DOUBLE, liabilities DOUBLE, assets DOUBLE, equity DOUBLE, roa DOUBLE, roe DOUBLE, debt\_to\_equity DOUBLE)'. The right sidebar shows the table structure for 'finance.top\_12\_german\_companies' with the following columns and data types:




Column	Type
company	string
period	string
revenue	double
net_income	double
liabilities	double
assets	double
equity	double
roa	double
roe	double
debt_to_equity	double
percentage_debt_to_equity	string



Vemos que los datos insertados son correctos

 **Hive**



Execute and watch

Add a description...

0.19s default  

```
1 | SELECT * FROM 'default'.'top_12_german_companies' LIMIT 100;
```

```
INFO : Connecting hive to database, not creating a lock manager
INFO : Executing command(queryId=hive_20241205014120_dbf3bdd6-8927-4dd4-93bc-439339488fb7): SELECT * FROM 'default'.'top_12_german_companies' LIMIT 100
INFO : Completed executing command(queryId=hive_20241205014120_dbf3bdd6-8927-4dd4-93bc-439339488fb7); Time taken: 0.001 seconds
INFO : OK
INFO : Concurrency mode is disabled, not creating a lock manager
```

Query History

Saved Queries

Results (100+)

	top_12_german_companies.company	top_12_german_companies.period	top_12_german_companies.revenue	top_12_german_companies.net_income	top_12_german_compan
1	Siemens AG	12/31/2017	19716237464	1276840007	45009303223
2	Allianz SE	12/31/2017	19458831198	1600107100	48538978480
3	BMW AG	12/31/2017	18808147150	9601843496	35382107627
4	BASF SE	12/31/2017	16895580815	1797081911	28309420014
5	Deutsche Telekom AG	12/31/2017	11505351408	1425682028	36680109736
6	Daimler AG	12/31/2017	17133317238	1743084807	19707492188
7	SAP SE	12/31/2017	17560385805	2276360916	40828269592
8	Bayer AG	12/31/2017	18251254610	2670535587	16524775630
9	Deutsche Bank AG	12/31/2017	9318303083	9585083337	11123712659
10	Porsche AG	12/31/2017	12119018742	8940910803	28361763209
11	Merck KGaA	12/31/2017	11760953647	812341476	29296072481
12	Volkswagen AG	12/31/2018	15132692627	1296423798	22603216585
13	Siemens AG	12/31/2018	9738110535	1128540207	47796930083
14	Allianz SE	12/31/2018	11358222175	1091045445	25112465464
15	BMW AG	12/31/2018	6557505339	6188836049	33315961378

## CRUD

### Create

Al crear la BBDD lo hemos hecho, las capturas están arriba.

### Read

Hacemos una consulta que nos devuelva todos los datos del 2018, tenemos que hacer la transformación de la fecha porque en nuestra BBDD está como String y no como date.

SELECT \*

FROM `default`.`top\_12\_german\_companies`

WHERE YEAR(TO\_DATE(FROM\_UNIXTIME(UNIX\_TIMESTAMP(period, 'MM/dd/yyyy')))) = 2018;

The screenshot shows the Hive query interface. The query executed is:

```
SELECT *
FROM `default`.`top_12_german_companies`
WHERE YEAR(TO_DATE(FROM_UNIXTIME(UNIX_TIMESTAMP(period, 'MM/dd/yyyy')))) = 2018;
```

The results are displayed in a table with 5 columns: `top_12_german_companies.company`, `top_12_german_companies.period`, `top_12_german_companies.revenue`, `top_12_german_companies.net_income`, and `top_12_german_companies...`. The table contains 15 rows of data, sorted by revenue.

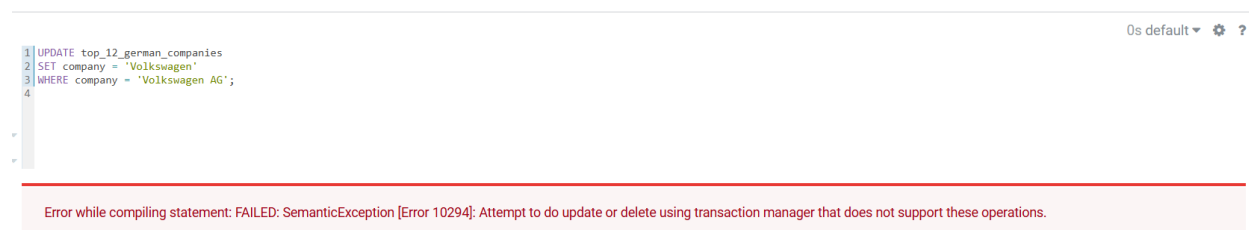
	top_12_german_companies.company	top_12_german_companies.period	top_12_german_companies.revenue	top_12_german_companies.net_income	top_12_german_companies...
1	Volkswagen AG	12/31/2018	15132692627	1296423798	22603216585
2	Siemens AG	12/31/2018	9738110535	1128540207	47796930083
3	Allianz SE	12/31/2018	11358222175	1091045445	25112465464
4	BMW AG	12/31/2018	6557505339	6188836049	33315961378
5	BASF SE	12/31/2018	10270182691	7088632771	42708053991
6	Deutsche Telekom AG	12/31/2018	15273880507	1698666925	35514608139
7	Daimler AG	12/31/2018	14737735881	1726421482	41539802392
8	SAP SE	12/31/2018	10184334320	1365996842	44932510908
9	Bayer AG	12/31/2018	18541451915	2242441772	28896911499
10	Deutsche Bank AG	12/31/2018	8808021584	1274941171	36770248755
11	Porsche AG	12/31/2018	7955302121	9397117624	35954156213
12	Merck KGaA	12/31/2018	14241334382	1165009578	37401670447
13	Volkswagen AG	3/31/2018	15989256775	1291419013	42659096630
14	Siemens AG	3/31/2018	10138254767	1282116952	45296373926
15	Allianz SE	3/31/2018	17610247426	2015329868	40259793264

## Update

```
UPDATE top_12_german_companies
```

```
SET company = 'Volkswagen'
```

```
WHERE company = 'Volkswagen AG';
```



The screenshot shows a SQL IDE interface. At the top right, there is a dropdown menu set to '0s default' and two icons (a gear and a question mark). The main area contains a SQL statement with line numbers 1 through 4: 

```
1 UPDATE top_12_german_companies  
2 SET company = 'Volkswagen'  
3 WHERE company = 'Volkswagen AG';  
4
```

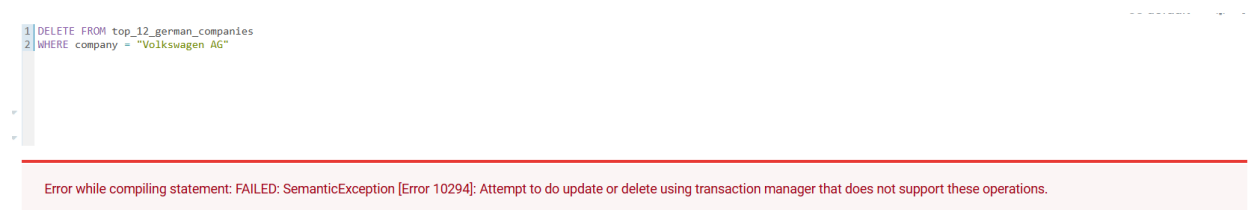
 Below the code, a red error bar displays the message: 'Error while compiling statement: FAILED: SemanticException [Error 10294]: Attempt to do update or delete using transaction manager that does not support these operations.'

Al intentar hacer un update nos produce un error, aunque la consulta sea correcta.

## Delete

```
DELETE FROM top_12_german_companies
```

```
WHERE company = "Volkswagen AG"
```



The screenshot shows a SQL IDE interface. At the top right, there is a dropdown menu set to '0s default' and two icons (a gear and a question mark). The main area contains a SQL statement with line numbers 1 through 2: 

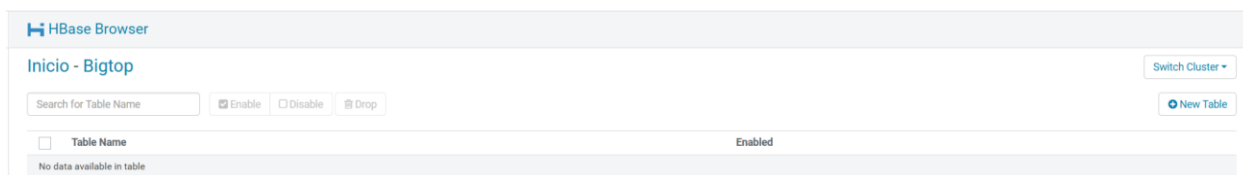
```
1 DELETE FROM top_12_german_companies  
2 WHERE company = "Volkswagen AG"
```

 Below the code, a red error bar displays the message: 'Error while compiling statement: FAILED: SemanticException [Error 10294]: Attempt to do update or delete using transaction manager that does not support these operations.'

Al igual que con las modificaciones, tampoco nos deja eliminar filas.

## HBASE

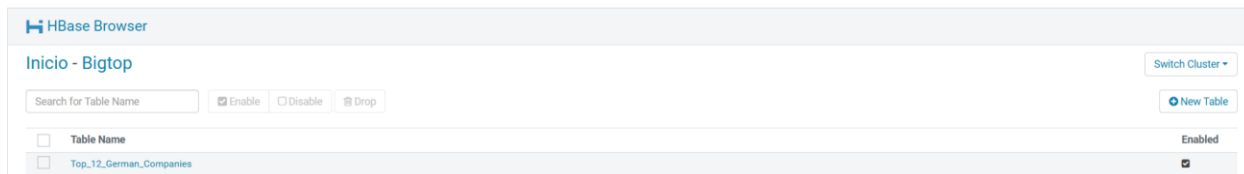
Dentro de HUE nos dirigimos a la pestaña de HBase y lo primero que tenemos que hacer es crear una nueva tabla.



Dentro de la creación le indicamos cuantas familias de columnas le queremos añadir.

The screenshot shows the 'Create New Table' dialog box. It has a title bar with a close button. Inside, there's a 'Table Name' field with the text 'Top\_12\_German\_Companies'. Below that, there's a 'Column Families' section. It has two input fields: the first contains 'cf\_general' and the second contains 'cf\_financials'. Each field has a blue 'x' icon to its left and a blue '+ Add a column property' link to its right. Below these fields, there's a blue '+ Add an additional column family' link. At the bottom right, there are two buttons: 'Cancel' and 'Submit'.

Ya podemos ver que nos la ha creado.



Ahora nos vamos a la línea de comandos, al entorno donde tenemos instalado el hadoop y cargamos el csv que tenemos en hdfs a la tabla que hemos creado de HBase.

```
sudo hbase org.apache.hadoop.hbase.mapreduce.ImportTsv \ -
Dimporttsv.columns=HBASE_ROW_KEY,cf_general:Company,cf_general:Period,cf_financials:
Revenue,cf_financials:Net_Income,cf_financials:Liabilities,cf_financials:Assets,cf_financials:E
quity \
```

```
-Dimporttsv.separator=',' \
```

```
Top_12_German_Companies \
```

```
/user/hadoop/Top_12_German_Companies.csv
```

```
[hadoop@ip-172-31-88-39 ~]$ sudo hbase org.apache.hadoop.hbase.mapreduce.ImportTsv
-Dimporttsv.columns=HBASE_ROW_KEY,cf_general:Company,cf_general:Period,cf_financials:Revenue,cf_financials:Net_Income,cf_financials:Liabilities,cf_financials:Assets,cf_financials:Equity
-Dimporttsv.separator=',' Top_12_German_Companies /user/hadoop/Top_12_German_Companies.csv
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/hbase/lib/client-facing-thirdparty/log4j-slf4j-impl-2.17.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Reload4jLoggerFactory]
[hadoop@ip-172-31-88-39 ~]$
```

cf\_financials van a contener los datos Revenue, Net\_Income, Liabilities, Assets y Equity, mientras que la columna cf\_general va a contener información adicional como Company y Period.

Allianz SE						
cf_financials	cf_financials	cf_financials	cf_financials	cf_general	cf_general	
Assets	Liabilities	Net_Income	Revenue	Company	Period	
17576276882	58775396940	41199120058	1727338520	9/30/2024	16485330446	
BASF SE						
cf_financials	cf_financials	cf_financials	cf_financials	cf_general	cf_general	
Assets	Liabilities	Net_Income	Revenue	Company	Period	
19004181470	42677324477	23673143006	7466675451	9/30/2024	9332023907	
Bayer AG						
cf_financials	cf_financials	cf_financials	cf_financials	cf_general	cf_general	
Assets	Liabilities	Net_Income	Revenue	Company	Period	
35755800815	69399402463	33643521649	2639441820	9/30/2024	17989542164	
BMW AG						
cf_financials	cf_financials	cf_financials	cf_financials	cf_general	cf_general	
Assets	Liabilities	Net_Income	Revenue	Company	Period	
41125290082	51913035226	10787737144	2101927829	9/30/2024	14640665936	
Daimler AG						
cf_financials	cf_financials	cf_financials	cf_financials	cf_general	cf_general	
Assets	Liabilities	Net_Income	Revenue	Company	Period	
12576276882	58775396940	41199120058	1727338520	9/30/2024	16485330446	

Ahora nos vamos a conectar a la shell de HBase para hacer las consultas del CRUD.

```
[hadoop@ip-172-31-88-39 ~]$ hbase shell
```



## CRUD

### Create

```
put 'Top_12_German_Companies', 'Siemens AG_12/31/2017', 'cf_general:Company',
'Siemens AG'
```

```
hbase:028:0> put 'Top_12_German_Companies', 'Siemens AG_12/31/2017', 'cf_general:
Company', 'Siemens AG'
Took 0.4440 seconds
hbase:029:0>
```

### Read

```
get 'Top_12_German_Companies', 'Siemens AG_12/31/2017'
```

```
hbase:029:0> get 'Top_12_German_Companies', 'Siemens AG_12/31/2017'
COLUMN          CELL
cf_general:Company  timestamp=2024-12-05T19:20:48.404, value=Siemens AG
1 row(s)
Took 0.0389 seconds
```

### Update

En hbase el comando put crea en caso de que no exista pero también modifica en el caso en el que si lo haga.

```
put 'Top_12_German_Companies', 'Siemens AG_12/31/2017', 'cf_financials:Net_Income',
'1300000000'
```

```
hbase:030:0> put 'Top_12_German_Companies', 'Siemens AG_12/31/2017', 'cf_financia
ls:Net_Income', '1300000'
Took 0.0095 seconds
```

## Comprobación

```
hbase:031:0> get 'Top_12_German_Companies', 'Siemens AG_12/312017'
COLUMN                                CELL
cf_financials:Net_In timestamp=2024-12-05T19:25:05.555, value=1300000
come
cf_general:Company   timestamp=2024-12-05T19:20:48.404, value=Siemens AG
1 row(s)
Took 0.0070 seconds
```

## Delete

deleteall 'Top\_12\_German\_Companies', 'Siemens AG\_12/31/2017'

Para eliminar toda la fila con la clave Siemens AG\_12/31/2017:

```
hbase:032:0> deleteall 'Top_12_German_Companies', 'Siemens AG_12/312017'
Took 0.0133 seconds
hbase:033:0> get 'Top_12_German_Companies', 'Siemens AG_12/312017'
COLUMN                                CELL
0 row(s)
Took 0.0049 seconds
hbase:034:0>
```