

Proyecto Hadoop 2\_6





Guillermo Fora Goncer

ÍNDICE

[**Introducción 3**](#_5f2sqzmefdcg)

[**Creación de la máquina 3**](#_opxvj9b5r49g)

[**Dataset 3**](#_iptqcjgw5sv2)

[Descargar Dataset de Kaggle 3](#_k5t7mg9g7pn)

[Explorar archivo CSV 4](#_ya0wb2to2jaf)

[Subir CSV del Dataset 4](#_1tsi7bnwafkh)

[Subir CSV al HDFS 6](#_qr9mtz69kw1h)

[**Manual de instancias CRUB HBase 6**](#_kugb87u6pg1t)

[**Manual de instancias CRUB Hive 8**](#_4op888dzb5a9)

[**HIVE 9**](#_1xtqq8msv5gw)

[**Creación de la tabla en Hive con HUE 9**](#_h3pq37y103lh)

[Consultas en Hive 15](#_c0uw4mbcguh8)

[**Hbase 18**](#_7l3dj9m94r0k)

[Creación de la tabla en Hbase con HUE 18](#_qatcabcfijqj)

[Consultas HBase 21](#_abluuf7u0us5)

# Introducción

Hemos visto que Hadoop es un ecosistema que consta de diferentes herramientas para almacenar, procesar y analizar datos Big Data (Petabytes: 1015 byte).

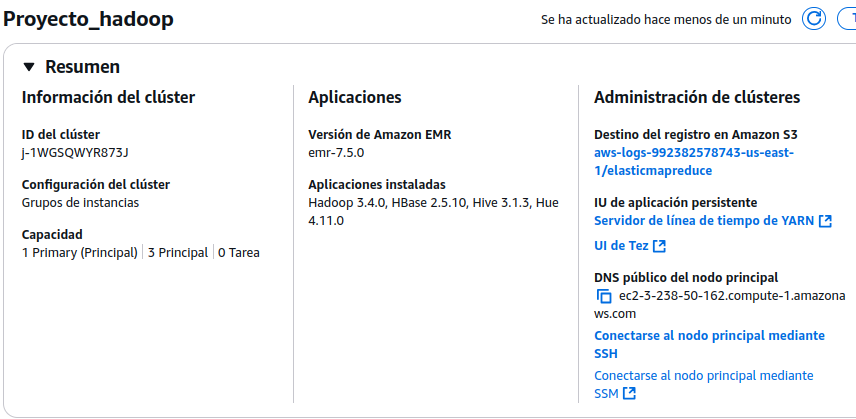
Hasta ahora hemos aprendido cómo instalar un clúster Hadoop en máquinas virtuales, así como en AWS y cómo lanzar tareas utilizando el motor MapReduce.

Para avanzar en el estudio de las herramientas del ecosistema Hadoop en este proyecto nos centraremos en Hive (SQL), HBase(NoSQL) y HUE (interfaz gráfica).

Lo que vamos a hacer es descargarnos desde Kaggle algunos ficheros, subirlos a Hadoop y utilizando Hive y HBase haremos consultas sobre sus datos, la herramienta HUE la podemos usar como interfaz.

Utilizaré este *tipo de formato para los comandos*.

# Creación de la máquina



# Dataset

## Descargar Dataset de Kaggle

Para este proyecto he elegido este dataset (Películas de marvel)

<https://www.kaggle.com/datasets/sarthakbharad/marvel-movies-dat>

## Explorar archivo CSV

Vamos a hacer una exploración del CSV del Dataset para que luego sea más fácil el proceso de creación de tablas.

Comprobamos que el .csv esta en ingles y tienes los siguientes nombres de columnas:

index,Title,Director (1),Director (2),Release Date (DD-MM-YYYY),IMDb (scored out of 10), IMDB Metascore (scored out of 100), Rotten Tomatoes - Critics (scored out of 100%), Rotten Tomatoes - Audience (scored out of 100%), Letterboxd (scored out of 5), CinemaScore (grades A+ to F), Budget (in million $), Domestic Gross (in million $), Worldwide Gross (in million $)

Esta es la descripción de las columnas:

* **Nombre de película, director 1, director 2 y fecha de lanzamiento**
* **IMBd** es un sitio web donde se puntúan las películas del 1 al 10.
* **IMBd Metascore** es la puntuación basada en reseñas de críticos de cine y otros factores. La puntuación de del 0 al 100
* **Rotten Tomatoes** es una plataforma que se especializa en críticas de cine. Tendremos dos columnas, una para críticos (critics) y otra de la audiencia (audience). La puntuación es sobre el 100%
* **Letterboxd** es una plataforma social donde los usuarios califican películas en una escala de 1 a 5.
* **CinemaScore** es un servicio que encarga encuestas a la audiencia que asiste al estreno de la película, obteniendo una calificación que varía de A+ (la mejor) a F (la peor).
* **Budget** es el presupuesto de la película en millones de dólares.
* **Domestic Gross** son ingresos obtenidos por la película solo en el mercado nacional (es decir, en el país de origen). Este valor también está en millones de dólares y muestra cuántos beneficios generó la película en su país de estreno.
* **Worldwide Gross** es la cantidad total de dinero recaudado a nivel mundial. Esto incluye las ganancias tanto a nivel nacional como internacional (otros países fuera del país de origen). También está en millones de dólares.

Limpiamos el CSV para que sea más fácil de manejar eliminando la columna index,dejamos la fecha en formato DD/MM/YY, quitamos paréntesis de los nombres de columnas, cambiamos los espacios por barra baja (\_), cambiamos las comas de los números decimales por puntos y quitamos todas las comillas. Comprobamos que en el director\_2 aparecen las columnas como nan, esto tendremos que cambiarlo por null para que Hbase no de error al importar el CSV

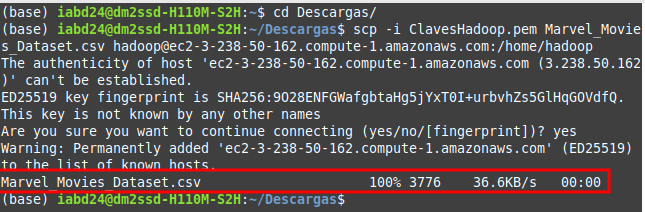
## Subir CSV del Dataset

Una vez creada la máquina, vamos a subir el Dataset descargado del link mencionado arriba. \*\* Hay que tener en cuenta que cada vez que clones el EMR tendrás que cambiar el nombre de la máquina por el nuevo (Lo que pones después del hadoop@...) para poder subir los datos y realizar la conexión SSH\*\*

Para ello vamos utilizar la terminal de nuestra máquina y pondremos la siguiente línea:

*scp -i ClavesHadoop.pem Marvel\_Movies\_Dataset.csv* [*hadoop@ec2-3-238-50-162.compute-1.amazonaws.com*](mailto:hadoop@ec2-3-238-50-162.compute-1.amazonaws.com)*:/home/hadoop*

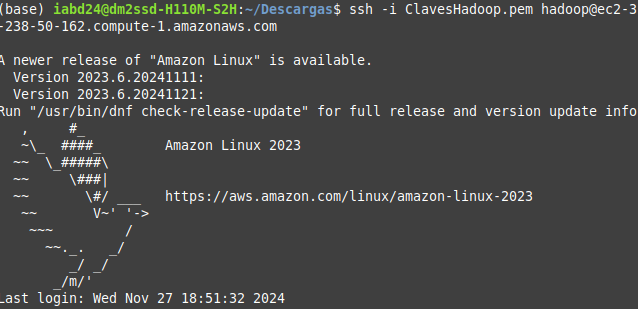
Para que este comando funcione, tendremos que tener en la carpeta de descargas las claves SSH que hemos creado en la creación de la máquina en AWS y el dataset descomprimido (el archivo .csv)



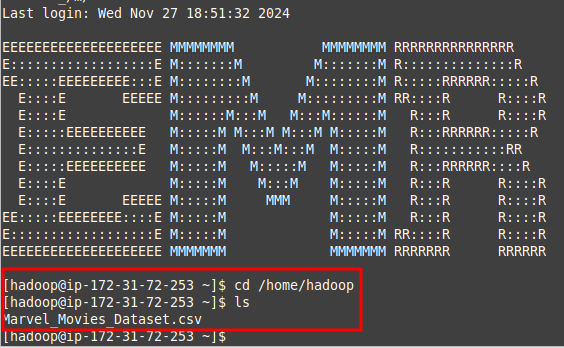
Ahora vamos a habilitar una conexión SSH con la máquina de AWS para comprobar que se ha realizado correctamente.

Para habilitarlo utilizamos el siguiente comando:

*ssh -i ClavesHadoop.pem* [*hadoop@ec2-3-238-50-162.compute-1.amazonaws.com*](mailto:hadoop@ec2-3-238-50-162.compute-1.amazonaws.com)



Nos dirigimos al directorio donde hemos copiado el .csv



## Subir CSV al HDFS

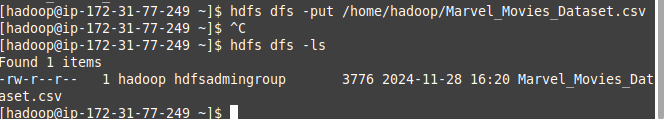
Una vez subido a la máquina de AWS, ahora tenemos que subirlo al HDFS, para ello usamos el siguiente comando:

*hdfs dfs -put /home/hadoop/Marvel\_Movies\_Dataset.csv*

\*\*Puedes escribir la ruta absoluta o la ruta relativa sin el /home/hadoop ya que se subirá a /user/hadoop\*\*

Si queremos verificar que se ha subido correctamente usamos este comando:

*hdfs dfs -ls*



# Manual de instancias CRUB HBase

Vamos a mencionar los comandos más usados en HBase.

*start-hbase.sh → Iniciar el servidor*

*hbase shell → Acceder al shell*

| **Operación** | **Comando** | **Descripción** | **Ejemplo** |
| --- | --- | --- | --- |
| **CREATE** | create | Crea una tabla en HBase especificando las familias de columnas. | create 'empleados', 'datos\_personales', 'datos\_laborales' |
| **INSERTAR** | put | Inserta un valor en una fila y columna específica. | put 'empleados', 'emp1', 'datos\_personales:nombre', 'Juan Perez' |
| **LEER (fila específica)** | get | Obtiene los datos de una fila completa o una columna específica. | get 'empleados', 'emp1' |
| **LEER (toda la tabla)** | scan | Escanea y muestra todas las filas de una tabla. | scan 'empleados' |
| **ACTUALIZAR** | put | Sobrescribe un valor existente en una columna. | put 'empleados', 'emp1', 'datos\_laborales:salario', '55000' |
| **ELIMINAR (valor específico)** | delete | Elimina un valor de una columna específica. | delete 'empleados', 'emp1', 'datos\_laborales:puesto' |
| **ELIMINAR (toda la fila)** | deleteall | Borra todos los datos asociados con una fila. | deleteall 'empleados', 'emp1' |
| **DESACTIVAR TABLA** | disable | Desactiva una tabla para operaciones. | disable 'empleados' |
| **ELIMINAR TABLA** | drop | Elimina una tabla previamente desactivada. | drop 'empleados' |
| **DESCRIBIR TABLA** | describe | Muestra el esquema de una tabla. | describe 'empleados' |
| **CONTAR FILAS** | count | Cuenta el número de filas en una tabla. | count 'empleados' |

# Manual de instancias CRUB Hive

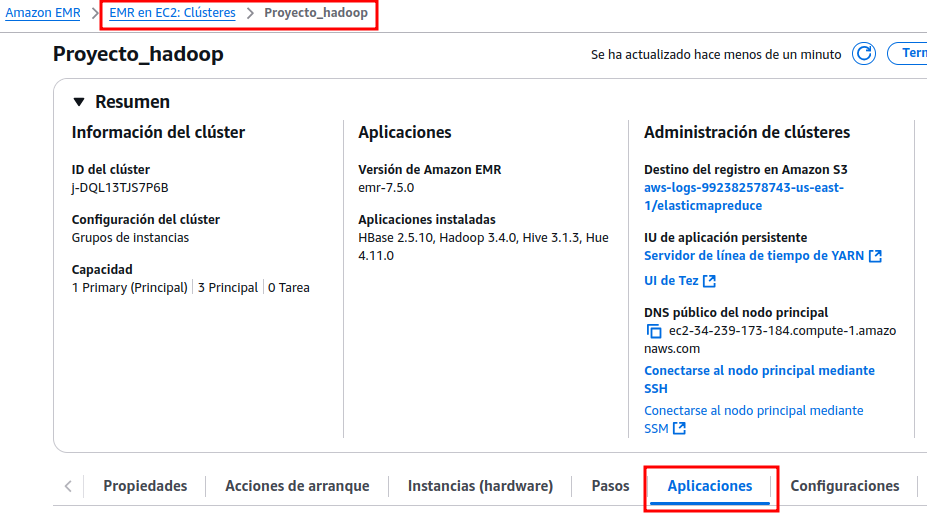
*hive → Iniciar shell*

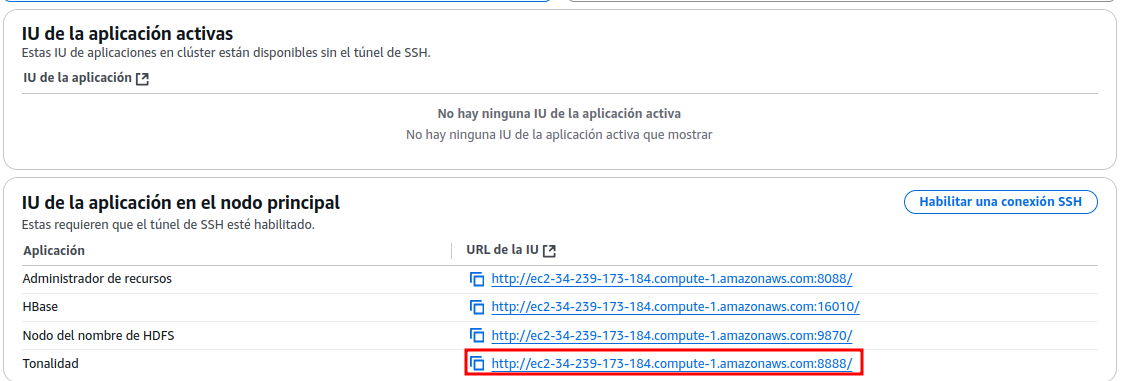
| **Operación** | **Comando** | **Descripción** | **Ejemplo** |
| --- | --- | --- | --- |
| **CREATE** | CREATE TABLE | Crea una tabla especificando columnas, tipos de datos, y formato de almacenamiento. | CREATE TABLE empleados (id INT, nombre STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ','; |
| **LOAD DATA** | LOAD DATA | Carga datos desde un archivo local o HDFS a una tabla. | LOAD DATA LOCAL INPATH '/ruta/empleados.csv' INTO TABLE empleados; |
| **SELECT** | SELECT | Consulta datos de una tabla con opciones de filtrado, agrupación y orden. | SELECT \* FROM empleados WHERE salario > 50000; |
| **DESCRIBE** | DESCRIBE | Muestra el esquema de una tabla (columnas y tipos de datos). | DESCRIBE empleados; |
| **UPDATE** | UPDATE | Actualiza valores en filas específicas de tablas transaccionales. | UPDATE empleados\_txn SET salario = 60000 WHERE id = 1; |
| **DELETE** | DELETE | Elimina filas específicas de tablas transaccionales. | DELETE FROM empleados\_txn WHERE edad > 30; |
| **TRUNCATE** | TRUNCATE | Vacía todos los datos de una tabla sin eliminarla. | TRUNCATE TABLE empleados; |
| **DROP** | DROP TABLE | Elimina una tabla y todos sus datos permanentemente. | DROP TABLE empleados; |
| **SHOW DATABASES** | SHOW DATABASES | Lista todas las bases de datos disponibles. | SHOW DATABASES; |
| **SHOW TABLES** | SHOW TABLES | Lista todas las tablas de la base de datos actual. | SHOW TABLES; |
| **INSERT OVERWRITE** | INSERT OVERWRITE | Exporta los resultados de una consulta a un archivo o directorio. | INSERT OVERWRITE LOCAL DIRECTORY '/ruta/salida' SELECT \* FROM empleados; |
| **COUNT** | SELECT COUNT(\*) | Cuenta el número total de filas en una tabla. | SELECT COUNT(\*) FROM empleados; |

# HIVE

## Creación de la tabla en Hive con HUE

Accedemos a la interfaz de HUE, para ello vamos al EMR → Aplicaciones → Tonalidad

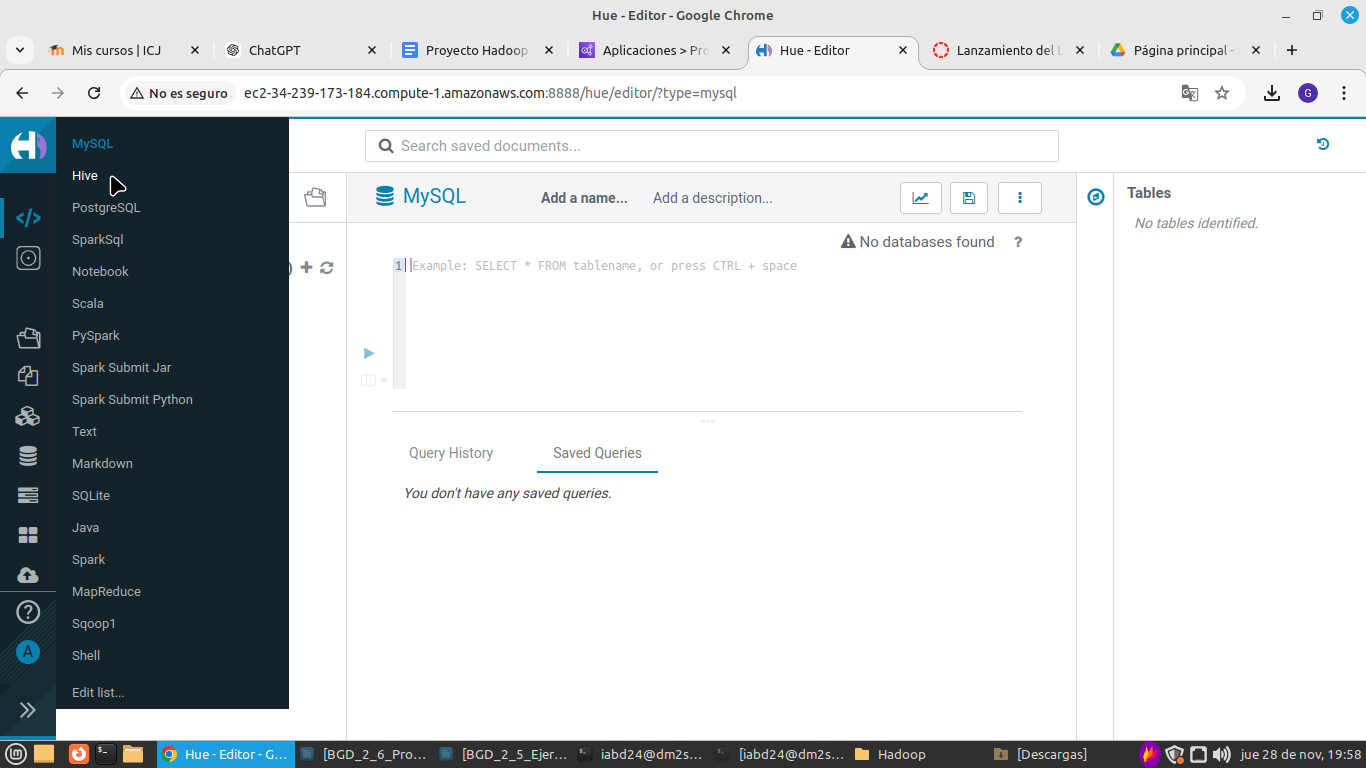




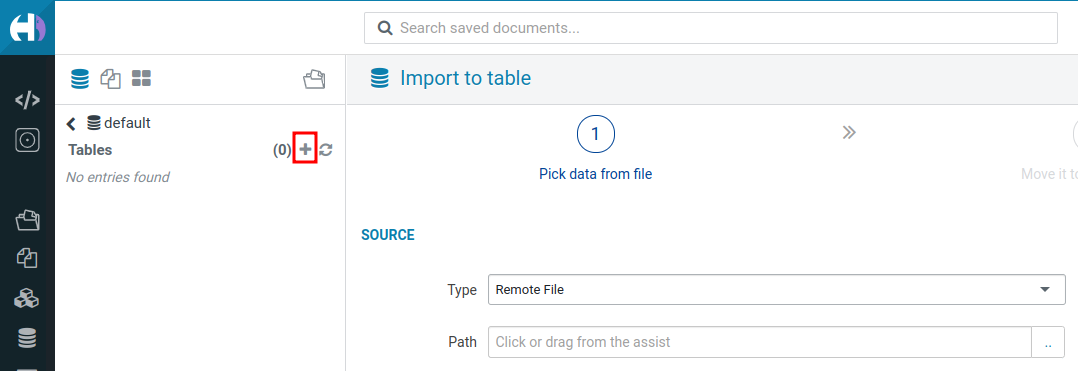
Cuando accedemos a HUE tenemos que crear un usuario.

Yo le pondré siempre admin → Admin.10

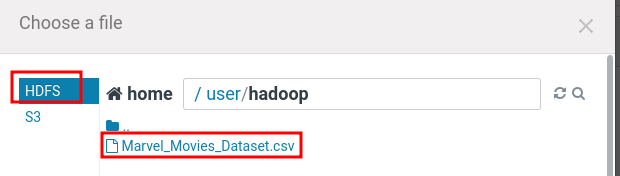
Ahora nos dirigimos a la base de datos Hive.



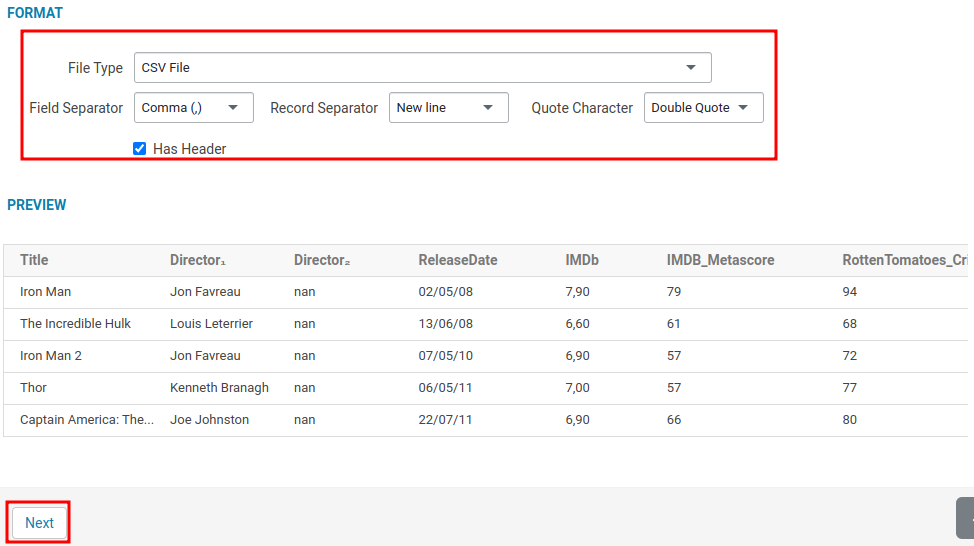
Y le damos a crear tabla



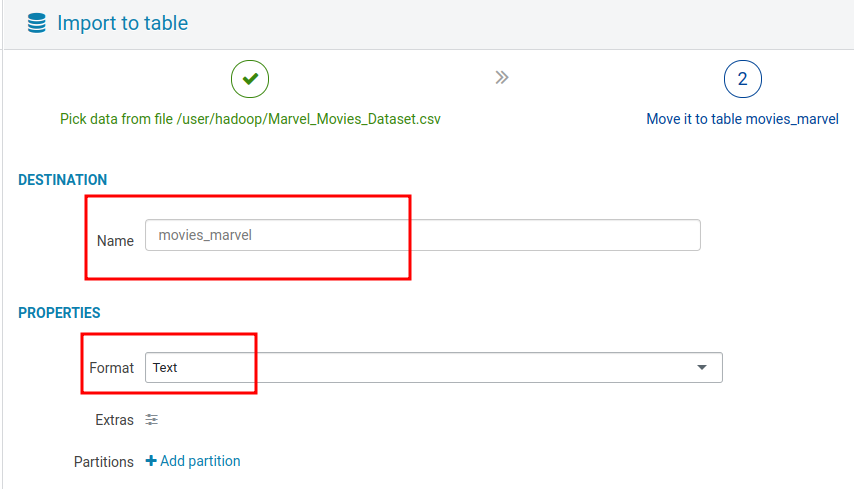
Ahora seleccionamos la ruta de donde está el csv, que debería estar en el hdfs.



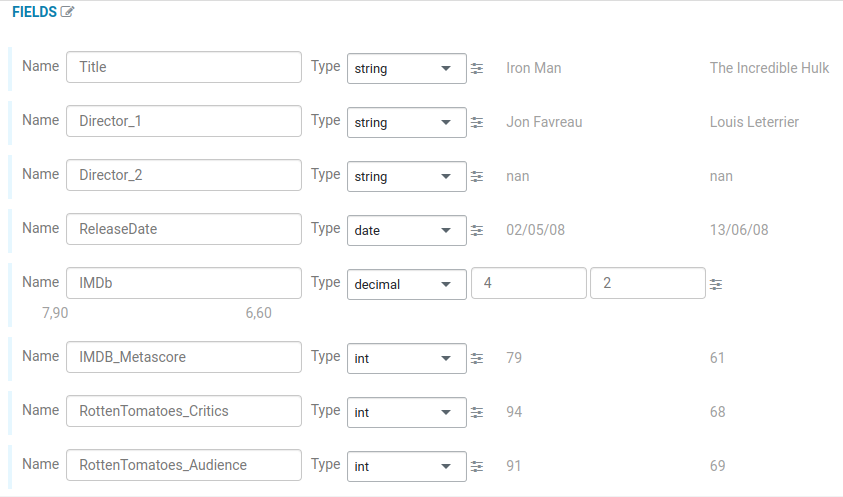
Ahora seleccionamos el tipo de archivo que es, los separadores que utiliza y le damos a next.

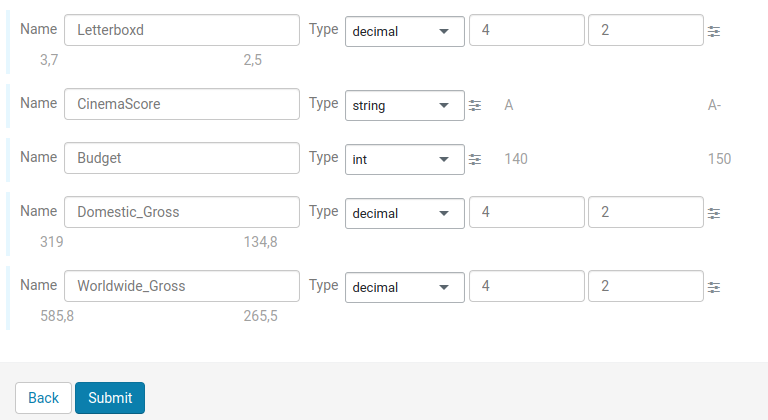


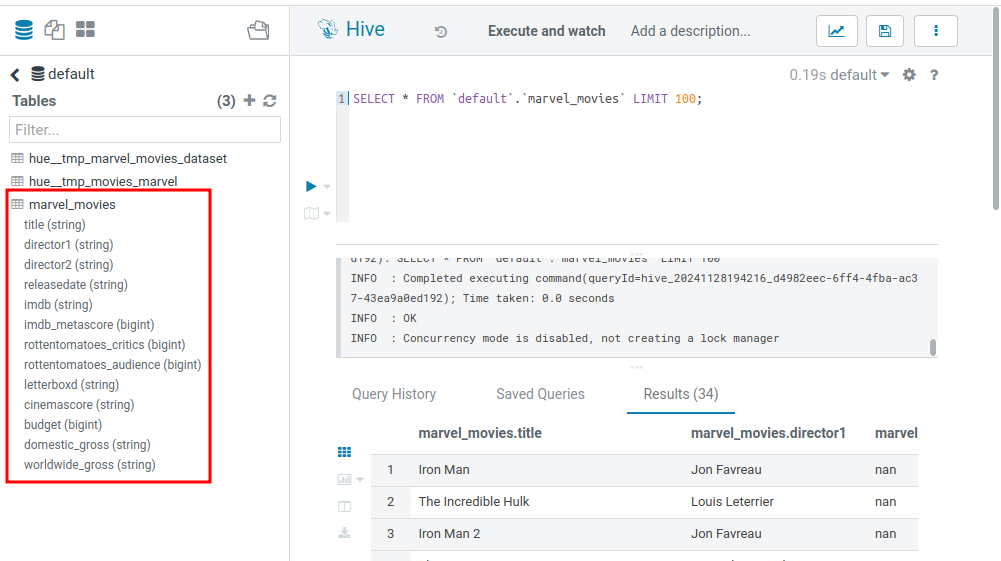
Ahora seleccionamos como se va a llamar la tabla y el formato



Más abajo, podrás modificar el nombre y el formato de cada columna. (No puede haber espacios o guiones en los nombres de las columnas)





Comprobamos que se ha creado correctamente  


De esta forma da un error de que no existe la base de datos y te crea una base de datos temporal.

Si quieres crearla sin que de errores puedes hacer de la siguiente forma:

Crea la tabla con la consola de Hive.



Este es el codigo que tienes que poner:

*CREATE TABLE marvel\_movies (*

*title STRING,*

*director\_1 STRING,*

*director\_2 STRING,*

*releasedate STRING,*

*imdb FLOAT,*

*imdb\_metascore INT,*

*rottentomatoes\_critics FLOAT,*

*rottentomatoes\_audience FLOAT,*

*letterboxd FLOAT,*

*cinemascore STRING,*

*budget FLOAT,*

*domestic\_gross FLOAT,*

*worldwide\_gross FLOAT*

*)*

*ROW FORMAT DELIMITED*

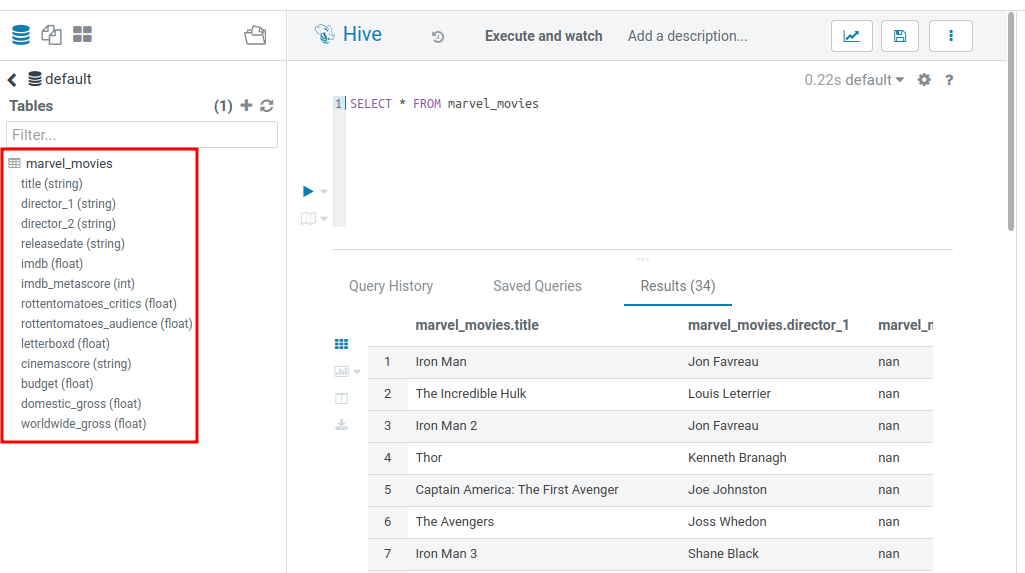
*FIELDS TERMINATED BY ','*

*STORED AS TEXTFILE;*

En el CSV tendrás que borrar el nombre de las columnas para importarlo ya que si están el nombre de las columnas dará error al importarlo.

*LOAD DATA INPATH '/user/hadoop/Marvel\_Movies\_Dataset.csv' INTO TABLE marvel\_movies*

Comprobamos que se ha creado correctamente y que se han insertado bien todas las filas.

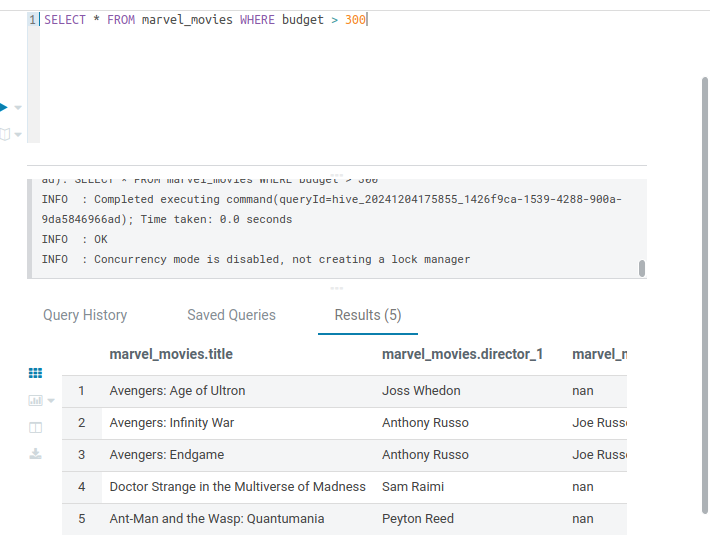
**

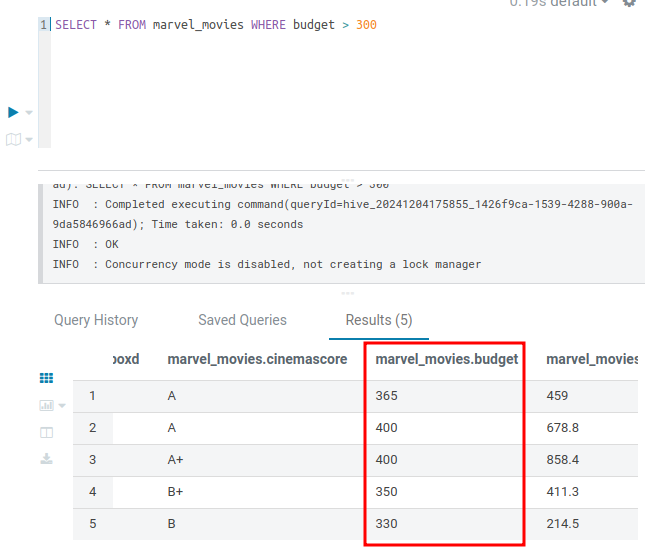
## 

## Consultas en Hive

Puede ir al apartado [Manual de instancias CRUB Hive](#_4op888dzb5a9) para ver como realizar las consultas.

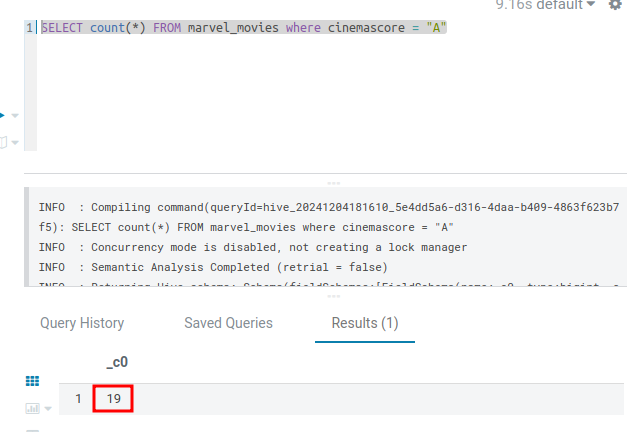
*SELECT \* FROM marvel\_movies WHERE budget > 300*





*SELECT count(\*) FROM marvel\_movies where cinemascore = "A"*

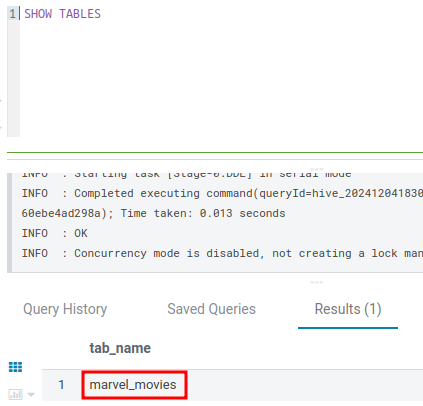




*SHOW DATABASES*

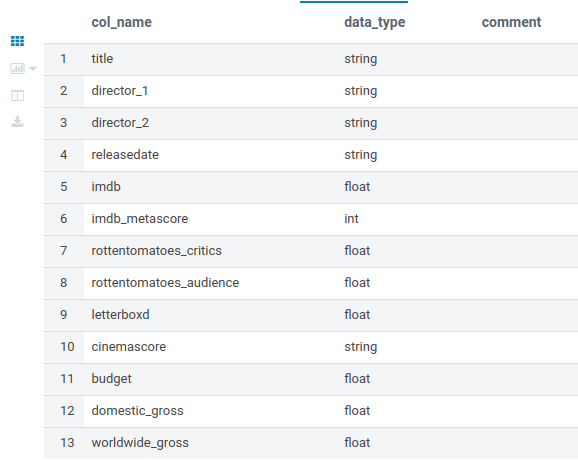


*SHOW TABLES*



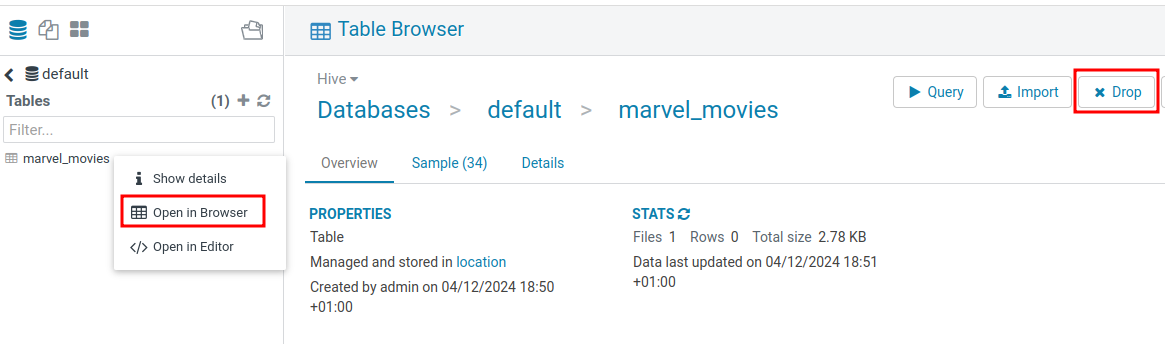
*DESCRIBE marvel\_movies*





Los comandos delete, update y truncate no te deja realizarlos porque el formato de la tabla no es ORC y para cambiar que puedas usar ese formato tiene que cambiar la configuración del archivo hive-site.xml

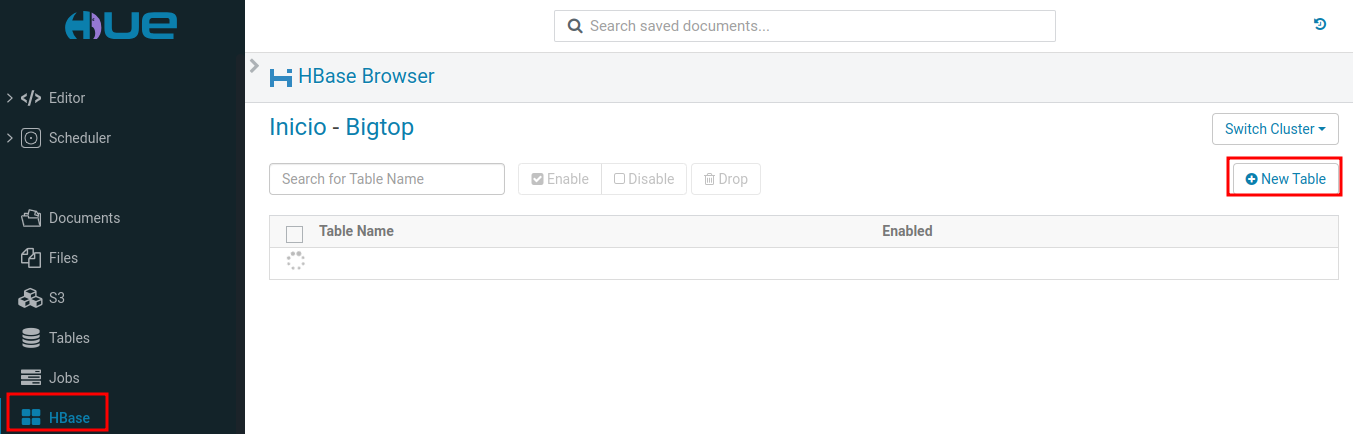
Para borrar la tabla tendrás que hacer click derecho en la tabla, seleccionar open in Browser y a la derecha aparece Drop.



# Hbase

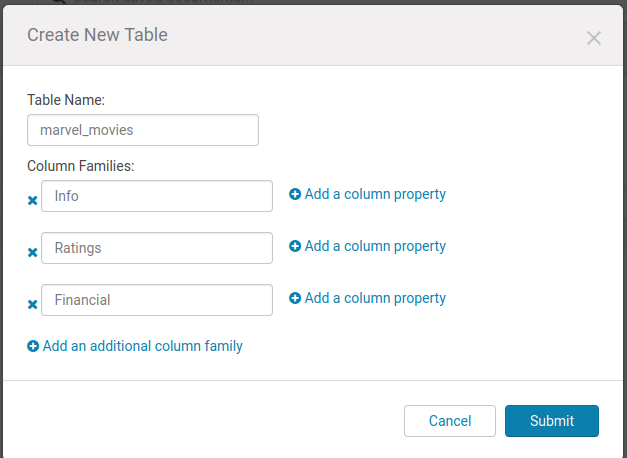
## Creación de la tabla en Hbase con HUE

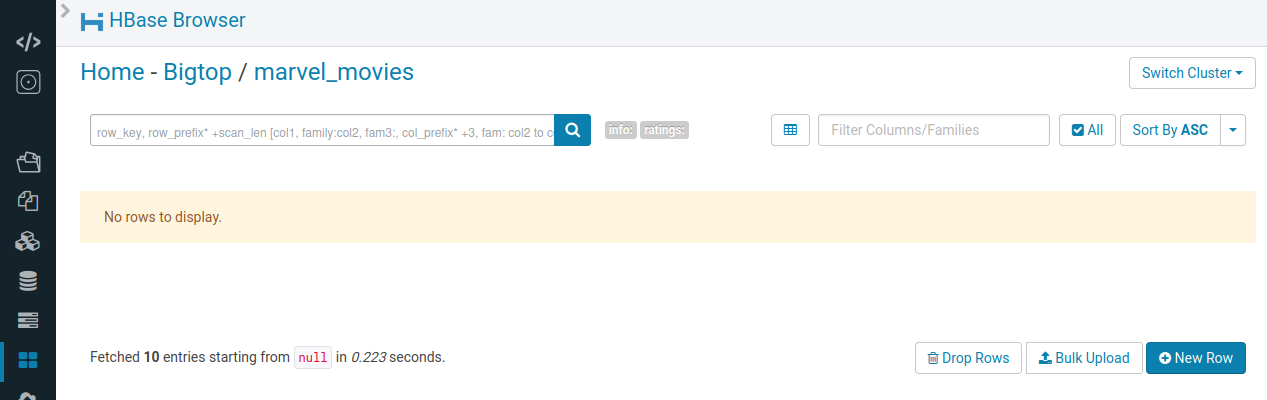
En la interfaz de HUE, seleccionamos Hbase que es el último icono de la barra de la izquierda (los 4 cuadrados) y seleccionamos New table a la derecha del todo.



Seleccionamos el nombre de la tabla y metemos el nombre de los grupos en los que queremos agrupar las columnas. Vamos a elegir como key\_row el title.

* Info
  + director\_1
  + director\_2
  + releasedate
* Ratings
  + IMBd
  + IMBd\_Metascore
  + RottenTomatoes\_critics
  + Letterboxd
  + CinemaScore
* Financial
  + Budge
  + DomesticGross
  + WorldwideGross



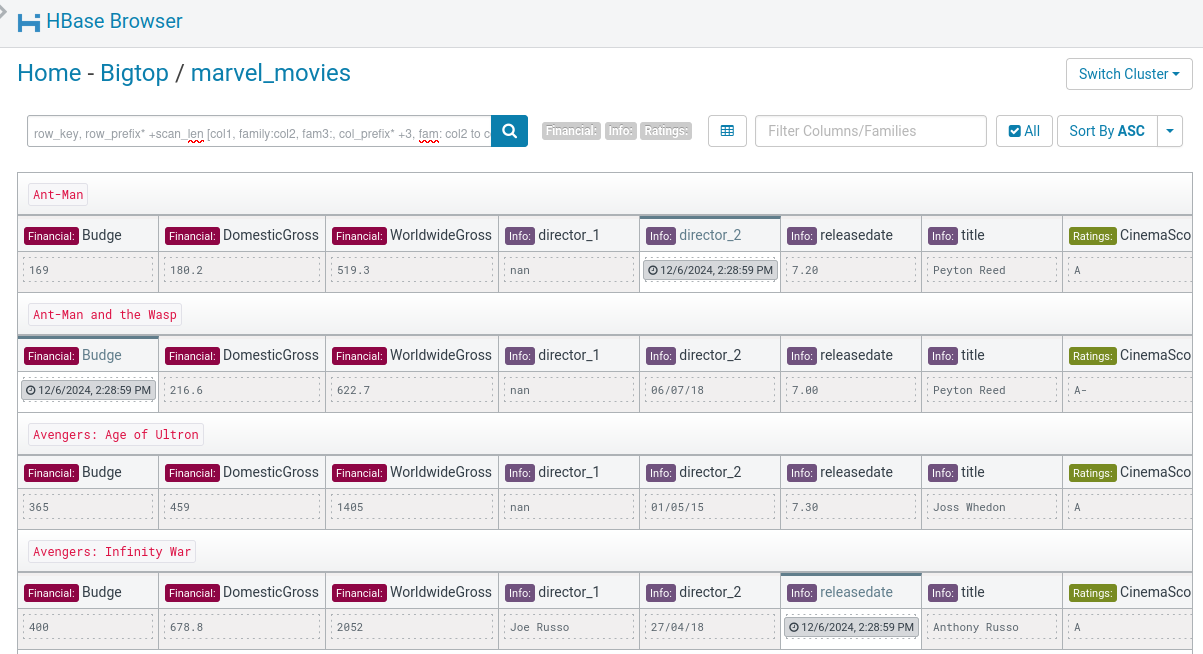


Probamos a importar el CSV desde el HUE pero parece que no funciona asi que lo haremos a través de la terminal. Nos conectamos por SSH al cluster de AWS como explicamos en el punto de [Subir CSV del dataset](#_1tsi7bnwafkh)

Para importar el CSV desde el HDFS usaremos el siguiente comando:

*sudo hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.columns=HBASE\_ROW\_KEY,Info:title,Info:director\_1,Info:director\_2,Info:releasedate,Ratings:IMBd,Ratings:IMBd\_Metascore,Ratings:RottenTomatoes\_critics,Ratings:Letterboxd,Ratings:CinemaScore,Financial:Budge,Financial:DomesticGross,Financial:WorldwideGross -Dimporttsv.separator=',' marvel\_movies hdfs:///user/hadoop/Marvel\_Movies\_Dataset.csv*

Comprobamos que se han importado bien los datos



## Consultas HBase

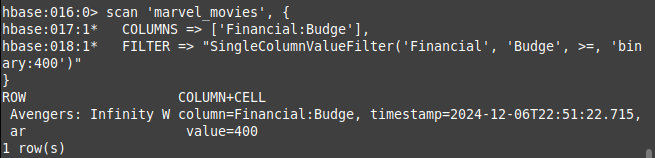
Puede ir al apartado [Manual de instancias CRUB HBase](#_kugb87u6pg1t) para ver como realizar las consultas.

*scan 'marvel\_movies', {*

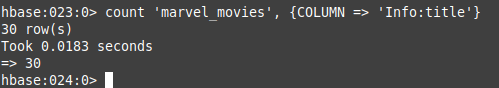
*COLUMNS => ['Financial:Budge'],*

*FILTER => "SingleColumnValueFilter('Financial', 'Budge', >=, 'binary:400')"*

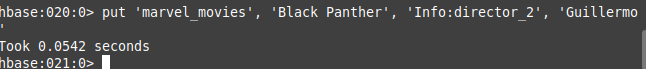
*}*

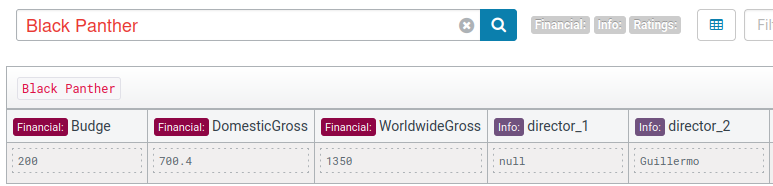


*count 'marvel\_movies', {COLUMN => 'Info:title'}*

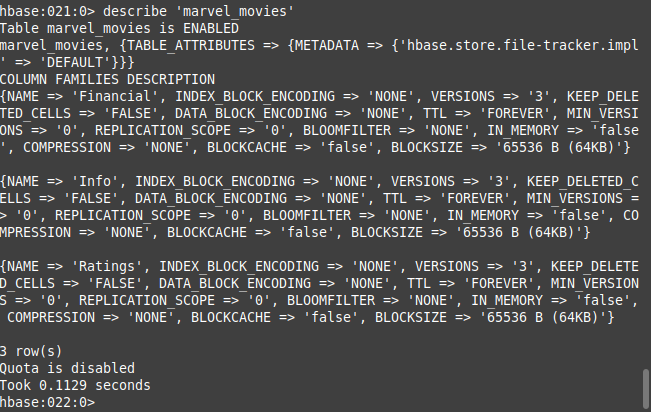


*put 'marvel\_movies', 'Black Panther', 'Info:director\_2', 'Guillermo'*



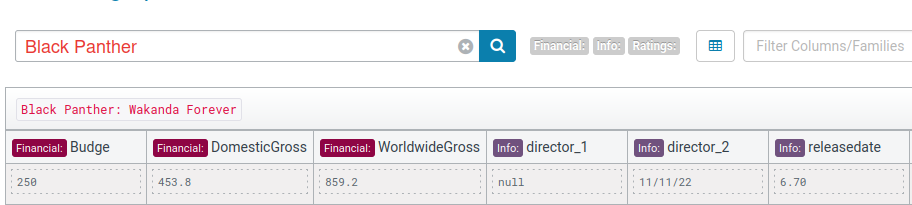
**

*describe ‘marvel\_movies’*



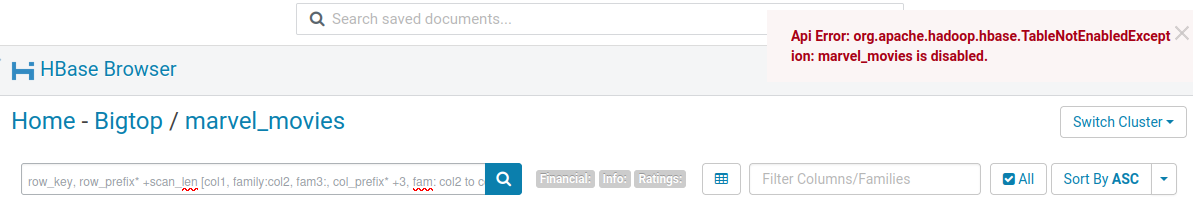
*deleteall 'marvel\_movies', 'Black Panther'*





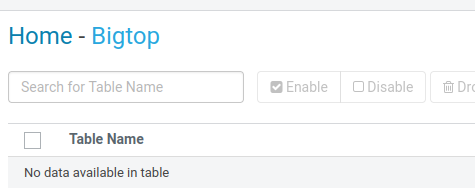
*disable ‘marvel\_movies’*





*drop ‘marvel\_movies*





’