Fall 2021 - CS 673: Scalable Databases

Assignment #4 HABSE

By

Pritam M. Channawar

Hbase:

* HBase is a distributed column-oriented database built on top of the Hadoop file system. It is an open-source project and is horizontally scalable.
* HBase is a data model that is similar to Google’s big table designed to provide quick random access to huge amounts of structured data. It leverages the fault tolerance provided by the Hadoop File System (HDFS).

Storage Mechanism in HBase:

HBase is a column-oriented database and the tables in it are sorted by row. The table schema defines only column families, which are the key value pairs. A table have multiple column families and each column family can have any number of columns. Subsequent column values are stored contiguously on the disk. Each cell value of the table has a timestamp. In short, in an HBase:

* Table is a collection of rows.
* Row is a collection of column families.
* Column family is a collection of columns.
* Column is a collection of key value pairs.

Database:

* I have created a tables in which ‘Movies’ represents table name and corresponds to column name “Movie\_Details”.
* Following table shows the details about table, column families and columns in my example.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RowId | Movies | Cast | | Production | Genre |
|  | Movie\_Details | Actor | Director | Producer | Genre\_Details |
| 1 | Title | FName | FName | FName | Movie\_Genre |
| 2 | Year | LName | LName | LName | Language |
| 3 | Budget | Gender | Gender | Gender | Country |
| 4 | Tagline | Earning | Earning | Prod\_Company | Rating |

Status:

* This command returns the status of the system including the details of the servers running on the system.

Graphical user interface, text

Description automatically generated with medium confidence

Version:

* This command returns the version of HBase used in your system.



Create Tables:

* Create table; pass table name, a dictionary of specifications per column family, and optionally a dictionary of table configuration.

Text

Description automatically generated

Syntax: create <tablename>, <columnfamilyname>

create 'Movies','Movie\_Details'

create 'Genre' , 'Genre\_Details'

create 'Cast' , 'Actor' , 'Director'

create 'Production' , 'Producer'

create ‘Person’, ‘Role'

List Tables:

Syntax: list

A picture containing chart

Description automatically generated

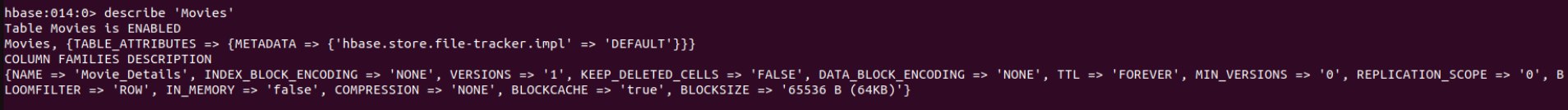
* “List” command will display all the tables that are present or created in HBase

Describe:

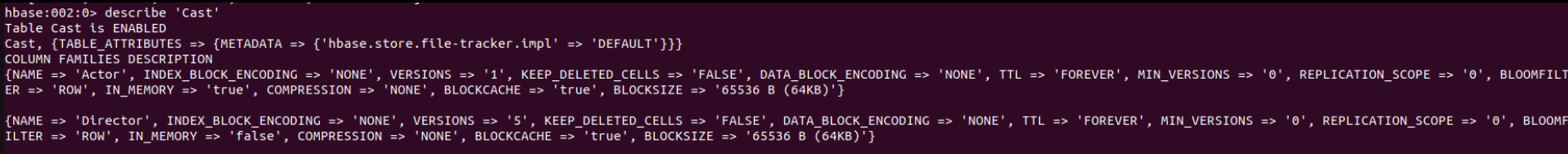
describe 'Movies'

describe 'Genre'

describe 'Cast'



* It will give more information about column families present in the mentioned table
* It will give information about table name with column families, associated filters, versions and some more details.



describe 'Production'

Scan:

* This command scans entire table and displays the table contents.
* We can pass several optional specifications to this scan command to get more information about the tables present in the system.

Syntax: scan <'tablename'>, {Optional parameters}

scan 'Movies'

scan 'Genre'

scan 'Cast'

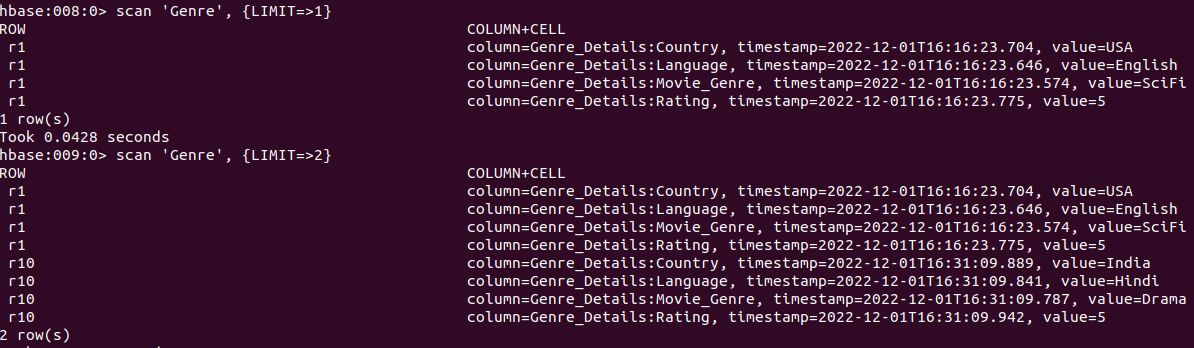
scan 'Production'

Text

Description automatically generated

scan 'Genre', {LIMIT=>1}

﻿scan 'Genre', {LIMIT=>2}

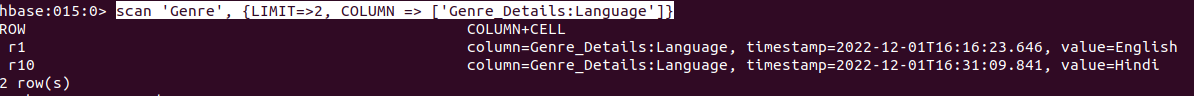


﻿

We can limit the results by passing scanner specification LIMIT, below example limits to one row.

* Abow command limits the results to 2 rows ‘r1’ and ‘r2’

﻿scan 'Genre', {LIMIT=>2, COLUMN => ['Genre\_Details:Language']}



* Using COLUMN specification, we can list the rows that have columns specified

﻿scan 'Genre', {LIMIT=>3, COLUMN => ['Genre\_Details:Language','Genre\_Details:Country'], STARTROW => 'r5'}

Text

Description automatically generated with low confidence

* This returns 3 rows with columns ‘Language’ and ‘Country’ with STARTROW ‘r5’

﻿scan 'Genre', {COLUMN => ['Genre\_Details:Language','Genre\_Details:Country'], STARTROW => 'r5', STOPROW => 'r9'}

Text

Description automatically generated

* This returns rows with columns ‘Language’ and ‘Country’ with STARTROW ‘r5’ and STOPROW ‘r9’

Put:

Syntax: put <'tablename'>,<'rowname'>,<'columnvalue'>,<'value'>

This command is used for following things

* It will put a cell ‘value’ at defined or specified table or row or column.
* It will optionally coordinate time stamp.

Text

Description automatically generated

* 10 rows are added in each table and code is as following

Row1:

put 'Movies' , 'r1' , 'Movie\_Details:Title' , 'The Matrix'

put 'Movies' , 'r1' , 'Movie\_Details:Year' , '1999'

put 'Movies' , 'r1' , 'Movie\_Details:Budget' , '500000000000'

put 'Movies' , 'r1' , 'Movie\_Details:Tagline' , 'Welcome to the Real World'

put 'Genre' , 'r1' , 'Genre\_Details:Movie\_Genre' , 'SciFi'

put 'Genre' , 'r1' , 'Genre\_Details:Language' , 'English'

put 'Genre' , 'r1' , 'Genre\_Details:Country' , 'USA'

put 'Genre' , 'r1' , 'Genre\_Details:Rating' , '5'

put 'Cast' , 'r1' , 'Actor:FName' , 'Keanu'

put 'Cast' , 'r1' , 'Actor:LName' , 'Reeves'

put 'Cast' , 'r1' , 'Actor:Gendre' , 'M'

put 'Cast' , 'r1' , 'Actor:Earning' , '4509766'

put 'Cast' , 'r1' , 'Director:FName' , 'Andy'

put 'Cast' , 'r1' , 'Director:LName' , 'Wachowski'

put 'Cast' , 'r1' , 'Director:Gendre' , 'M'

put 'Cast' , 'r1' , 'Director:Earning' , '6750382437'

put 'Production' , 'r1' , 'Producer:FName' , 'Joel'

put 'Production' , 'r1' , 'Producer:LName' , 'Silver'

put 'Production' , 'r1' , 'Producer:Gendre' , 'F'

put 'Production' , 'r1' , 'Producer:Prod\_Company' , 'MCompany'

Row2:

put 'Movies' , 'r2' , 'Movie\_Details:Title' , 'Top Gun'

put 'Movies' , 'r2' , 'Movie\_Details:Year' , '1986'

put 'Movies' , 'r2' , 'Movie\_Details:Budget' , '67450000000000'

put 'Movies' , 'r2' , 'Movie\_Details:Tagline' , 'I feel the need, the need for speed'

put 'Genre' , 'r2' , 'Genre\_Details:Movie\_Genre' , 'SciFi'

put 'Genre' , 'r2' , 'Genre\_Details:Language' , 'English'

put 'Genre' , 'r2' , 'Genre\_Details:Country' , 'USA'

put 'Genre' , 'r2' , 'Genre\_Details:Rating' , '5'

put 'Cast' , 'r2' , 'Actor:FName' , 'Tom'

put 'Cast' , 'r2' , 'Actor:LName' , 'Cruise'

put 'Cast' , 'r2' , 'Actor:Gendre' , 'M'

put 'Cast' , 'r2' , 'Actor:Earning' , '4500009766'

put 'Cast' , 'r2' , 'Director:FName' , 'Tony'

put 'Cast' , 'r2' , 'Director:LName' , 'Scott'

put 'Cast' , 'r2' , 'Director:Gendre' , 'M'

put 'Cast' , 'r2' , 'Director:Earning' , '896436045'

put 'Production' , 'r2' , 'Producer:FName' , 'Jim'

put 'Production' , 'r2' , 'Producer:LName' , 'Cash'

put 'Production' , 'r2' , 'Producer:Gendre' , 'M'

put 'Production' , 'r2' , 'Producer:Prod\_Company' , 'TopCompany'

Row3:

put 'Movies' , 'r3' , 'Movie\_Details:Title' , 'Jerry Maguire'

put 'Movies' , 'r3' , 'Movie\_Details:Year' , '2002'

put 'Movies' , 'r3' , 'Movie\_Details:Budget' , '8867000000000'

put 'Movies' , 'r3' , 'Movie\_Details:Tagline' , 'The rest of his life begins now.'

put 'Genre' , 'r3' , 'Genre\_Details:Movie\_Genre' , 'Romantic'

put 'Genre' , 'r3' , 'Genre\_Details:Language' , 'English'

put 'Genre' , 'r3' , 'Genre\_Details:Country' , 'USA'

put 'Genre' , 'r3' , 'Genre\_Details:Rating' , '4'

put 'Cast' , 'r3' , 'Actor:FName' , 'Renee'

put 'Cast' , 'r3' , 'Actor:LName' , 'Zellweger'

put 'Cast' , 'r3' , 'Actor:Gendre' , 'F'

put 'Cast' , 'r3' , 'Actor:Earning' , '56000004366'

put 'Cast' , 'r3' , 'Director:FName' , 'Cameron'

put 'Cast' , 'r3' , 'Director:LName' , 'Crowe'

put 'Cast' , 'r3' , 'Director:Gendre' , 'M'

put 'Cast' , 'r3' , 'Director:Earning' , '34700045'

put 'Production' , 'r3' , 'Producer:FName' , 'Cameron'

put 'Production' , 'r3' , 'Producer:LName' , 'Crowe'

put 'Production' , 'r3' , 'Producer:Gendre' , 'M'

put 'Production' , 'r3' , 'Producer:Prod\_Company' , 'JCompany'

Row4:

put 'Movies' , 'r4' , 'Movie\_Details:Title' , 'When Harry Met Sally'

put 'Movies' , 'r4' , 'Movie\_Details:Year' , '1998'

put 'Movies' , 'r4' , 'Movie\_Details:Budget' , '999000000000'

put 'Movies' , 'r4' , 'Movie\_Details:Tagline' , 'At odds in life... in love on-line.'

put 'Genre' , 'r4' , 'Genre\_Details:Movie\_Genre' , 'Romantic'

put 'Genre' , 'r4' , 'Genre\_Details:Language' , 'English'

put 'Genre' , 'r4' , 'Genre\_Details:Country' , 'USA'

put 'Genre' , 'r4' , 'Genre\_Details:Rating' , '4'

put 'Cast' , 'r4' , 'Actor:FName' , 'Billy'

put 'Cast' , 'r4' , 'Actor:LName' , 'Crystal'

put 'Cast' , 'r4' , 'Actor:Gendre' , 'F'

put 'Cast' , 'r4' , 'Actor:Earning' , '9000004366'

put 'Cast' , 'r4' , 'Director:FName' , 'Rob'

put 'Cast' , 'r4' , 'Director:LName' , 'Reiner'

put 'Cast' , 'r4' , 'Director:Gendre' , 'M'

put 'Cast' , 'r4' , 'Director:Earning' , '890700045'

put 'Production' , 'r4' , 'Producer:FName' , 'Nora'

put 'Production' , 'r4' , 'Producer:LName' , 'Ephron'

put 'Production' , 'r4' , 'Producer:Gendre' , 'F'

put 'Production' , 'r4' , 'Producer:Prod\_Company' , 'HCompany'

Text

Description automatically generated

Row5:

put 'Movies' , 'r5' , 'Movie\_Details:Title' , ' One Flew Over the Cuckoos Nest '

put 'Movies' , 'r5' , 'Movie\_Details:Year' , '1975'

put 'Movies' , 'r5' , 'Movie\_Details:Budget' , '786754000000000'

put 'Movies' , 'r5' , 'Movie\_Details:Tagline' , 'If hes crazy, what does that make you?'

put 'Genre' , 'r5' , 'Genre\_Details:Movie\_Genre' , 'Comedy'

put 'Genre' , 'r5' , 'Genre\_Details:Language' , 'English'

put 'Genre' , 'r5' , 'Genre\_Details:Country' , 'USA'

put 'Genre' , 'r5' , 'Genre\_Details:Rating' , '4'

put 'Cast' , 'r5' , 'Actor:FName' , 'Jack'

put 'Cast' , 'r5' , 'Actor:LName' , 'Nicholson'

put 'Cast' , 'r5' , 'Actor:Gendre' , 'M'

put 'Cast' , 'r5' , 'Actor:Earning' , '90000000066'

put 'Cast' , 'r5' , 'Director:FName' , 'Nancy'

put 'Cast' , 'r5' , 'Director:LName' , 'Meyers'

put 'Cast' , 'r5' , 'Director:Gendre' , 'M'

put 'Cast' , 'r5' , 'Director:Earning' , '89070000045'

put 'Production' , 'r5' , 'Producer:FName' , 'Nancy'

put 'Production' , 'r5' , 'Producer:LName' , 'Meyers'

put 'Production' , 'r5' , 'Producer:Gendre' , 'M'

put 'Production' , 'r5' , 'Producer:Prod\_Company' , 'OCompany'

Row 6:

put 'Movies' , 'r6' , 'Movie\_Details:Title' , 'Dilwale Dulhaniya Le Jaenge'

put 'Movies' , 'r6' , 'Movie\_Details:Year' , '1998'

put 'Movies' , 'r6' , 'Movie\_Details:Budget' , '7000000000'

put 'Movies' , 'r6' , 'Movie\_Details:Tagline' , 'DDLJ'

put 'Genre' , 'r6' , 'Genre\_Details:Movie\_Genre' , 'Romantic'

put 'Genre' , 'r6' , 'Genre\_Details:Language' , 'Hindi'

put 'Genre' , 'r6' , 'Genre\_Details:Country' , 'India'

put 'Genre' , 'r6' , 'Genre\_Details:Rating' , '4'

put 'Cast' , 'r6' , 'Actor:FName' , 'Kajol'

put 'Cast' , 'r6' , 'Actor:LName' , 'Devgan'

put 'Cast' , 'r6' , 'Actor:Gendre' , 'F'

put 'Cast' , 'r6' , 'Actor:Earning' , '700000'

put 'Cast' , 'r6' , 'Director:FName' , 'Yash'

put 'Cast' , 'r6' , 'Director:LName' , 'Chopra'

put 'Cast' , 'r6' , 'Director:Gendre' , 'M'

put 'Cast' , 'r6' , 'Director:Earning' , '89070000000045'

put 'Production' , 'r6' , 'Producer:FName' , 'Yash'

put 'Production' , 'r6' , 'Producer:LName' , 'Chopra'

put 'Production' , 'r6' , 'Producer:Gendre' , 'M'

put 'Production' , 'r6' , 'Producer:Prod\_Company' , 'YRF films'

Row 7:

put 'Movies' , 'r7' , 'Movie\_Details:Title' , 'Bahubali-the Beginning'

put 'Movies' , 'r7' , 'Movie\_Details:Year' , '2015'

put 'Movies' , 'r7' , 'Movie\_Details:Budget' , '100000000000000'

put 'Movies' , 'r7' , 'Movie\_Details:Tagline' , ' The boy he raised, the man he killed'

put 'Genre' , 'r7' , 'Genre\_Details:Movie\_Genre' , 'Action'

put 'Genre' , 'r7' , 'Genre\_Details:Language' , 'Hindi'

put 'Genre' , 'r7' , 'Genre\_Details:Country' , 'India'

put 'Genre' , 'r7' , 'Genre\_Details:Rating' , '5'

put 'Cast' , 'r7' , 'Actor:FName' , 'Prabhas'

put 'Cast' , 'r7' , 'Actor:LName' , ''

put 'Cast' , 'r7' , 'Actor:Gendre' , 'M'

put 'Cast' , 'r7' , 'Actor:Earning' , '900000000'

put 'Cast' , 'r7' , 'Director:FName' , 's.s'

put 'Cast' , 'r7' , 'Director:LName' , 'Rajamauli'

put 'Cast' , 'r7' , 'Director:Gendre' , 'M'

put 'Cast' , 'r7' , 'Director:Earning' , '76500000045'

put 'Production' , 'r7' , 'Producer:FName' , 'Prasad'

put 'Production' , 'r7' , 'Producer:LName' , 'Devineni'

put 'Production' , 'r7' , 'Producer:Gendre' , 'M'

put 'Production' , 'r7' , 'Producer:Prod\_Company' , 'B films'

Row 8:

put 'Movies' , 'r8' , 'Movie\_Details:Title' , 'Bahubali-the Conclusion'

put 'Movies' , 'r8' , 'Movie\_Details:Year' , '2017'

put 'Movies' , 'r8' , 'Movie\_Details:Budget' , '100000000000000'

put 'Movies' , 'r8' , 'Movie\_Details:Tagline' , ' The boy he raised, the man he killed'

put 'Genre' , 'r8' , 'Genre\_Details:Movie\_Genre' , 'Action'

put 'Genre' , 'r8' , 'Genre\_Details:Language' , 'Hindi'

put 'Genre' , 'r8' , 'Genre\_Details:Country' , 'India'

put 'Genre' , 'r8' , 'Genre\_Details:Rating' , '5'

put 'Cast' , 'r8' , 'Actor:FName' , 'Prabhas'

put 'Cast' , 'r8' , 'Actor:LName' , ''

put 'Cast' , 'r8' , 'Actor:Gendre' , 'M'

put 'Cast' , 'r8' , 'Actor:Earning' , '900000000'

put 'Cast' , 'r8' , 'Director:FName' , 's.s'

put 'Cast' , 'r8' , 'Director:LName' , 'Rajamauli'

put 'Cast' , 'r8' , 'Director:Gendre' , 'M'

put 'Cast' , 'r8' , 'Director:Earning' , '76500000045'

put 'Production' , 'r8' , 'Producer:FName' , 'Prasad'

put 'Production' , 'r8' , 'Producer:LName' , 'Devineni'

put 'Production' , 'r8' , 'Producer:Gendre' , 'M'

put 'Production' , 'r8' , 'Producer:Prod\_Company' , 'B films'

Row 9:

put 'Movies' , 'r9' , 'Movie\_Details:Title' , 'Sholey'

put 'Movies' , 'r9' , 'Movie\_Details:Year' , '1975'

put 'Movies' , 'r9' , 'Movie\_Details:Budget' , '645300000000000'

put 'Movies' , 'r9' , 'Movie\_Details:Tagline' , ' The greatest star cast ever assembled – the greatest story ever told '

put 'Genre' , 'r9' , 'Genre\_Details:Movie\_Genre' , 'Action'

put 'Genre' , 'r9' , 'Genre\_Details:Language' , 'Hindi'

put 'Genre' , 'r9' , 'Genre\_Details:Country' , 'India'

put 'Genre' , 'r9' , 'Genre\_Details:Rating' , '5'

put 'Cast' , 'r9' , 'Actor:FName' , 'Amitabh'

put 'Cast' , 'r9' , 'Actor:LName' , 'Bachachan'

put 'Cast' , 'r9' , 'Actor:Gendre' , 'M'

put 'Cast' , 'r9' , 'Actor:Earning' , '97800000000'

put 'Cast' , 'r9' , 'Director:FName' , 'Ramesh'

put 'Cast' , 'r9' , 'Director:LName' , 'Sippy'

put 'Cast' , 'r9' , 'Director:Gendre' , 'M'

put 'Cast' , 'r9' , 'Director:Earning' , '76500000045'

put 'Production' , 'r9' , 'Producer:FName' , 'United'

put 'Production' , 'r9' , 'Producer:LName' , ''

put 'Production' , 'r9' , 'Producer:Gendre' , ''

put 'Production' , 'r9' , 'Producer:Prod\_Company' , 'Sippy films'

Row10:

put 'Movies' , 'r10' , 'Movie\_Details:Title' , 'Devdas'

put 'Movies' , 'r10' , 'Movie\_Details:Year' , '2002'

put 'Movies' , 'r10' , 'Movie\_Details:Budget' , '540000000000000000'

put 'Movies' , 'r10' , 'Movie\_Details:Tagline' , 'A grand saga of timeless love'

put 'Genre' , 'r10' , 'Genre\_Details:Movie\_Genre' , 'Drama'

put 'Genre' , 'r10' , 'Genre\_Details:Language' , 'Hindi'

put 'Genre' , 'r10' , 'Genre\_Details:Country' , 'India'

put 'Genre' , 'r10' , 'Genre\_Details:Rating' , '5'

put 'Cast' , 'r10' , 'Actor:FName' , 'Aishwarya'

put 'Cast' , 'r10' , 'Actor:LName' , 'Rai'

put 'Cast' , 'r10' , 'Actor:Gendre' , 'F'

put 'Cast' , 'r10' , 'Actor:Earning' , '66000000000'

put 'Cast' , 'r10' , 'Director:FName' , 'Sanjay Leela'

put 'Cast' , 'r10' , 'Director:LName' , 'Bhansali'

put 'Cast' , 'r10' , 'Director:Gendre' , 'M'

put 'Cast' , 'r9' , 'Director:Earning' , '432956500000045'

put 'Production' , 'r10' , 'Producer:FName' , 'United'

put 'Production' , 'r10' , 'Producer:LName' , ''

put 'Production' , 'r10' , 'Producer:Gendre' , ''

put 'Production' , 'r10' , 'Producer:Prod\_Company' , 'Mega Bollywood'

Insert partial row:



﻿put 'Movies' , 'r11' , 'Movie\_Details:Title' , 'Bat-man'

Text

Description automatically generated

* ﻿get 'Movies' , 'r11'
* Disply partial row values

Alter:

Syntax: alter <tablename>, NAME=><column familyname>, VERSIONS=>5

* This command alters the column family schema.
* We can use MAX\_FILESIZE with the table as scope attribute as above
* The number represent in MAX\_FILESIZE is in term of memory in bytes

Text

Description automatically generated

alter 'Cast', {NAME => 'Actor', IN\_MEMORY => true}, {NAME => 'Director', VERSIONS => 5}

To change or add the ‘Actor’ column family in table ‘Cast’ from current value to keep a maximum of 5 cell VERSIONS,

* “Cast” is table name created with column name “Actor” previously

Here with the help of an alter command we are trying to change the column family schema to Actor from Cast

alter 'Movies', NAME='Movie\_Details', VERSIONS=>4



* You can also operate the alter command on several column families as well. For example, we will define two new column to our existing table “Movies”.

alter 'Genre', NAME='Genre\_Details', MAX\_FILESIZE=>'132545224'

alter 'Production', NAME='Producer', METHOD => 'table\_att\_unset'

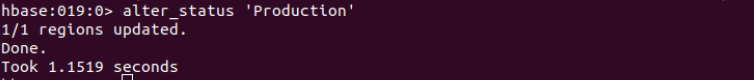
alter 'Person', NAME => 'Detail', METHOD => 'delete'

* You can change table-scope attributes like MAX\_FILESIZE, READONLY, MEMSTORE\_FLUSHSIZE, DEFERRED\_LOG\_FLUSH, etc. These can be put at the end;for example, to change the max size of a region to 128MB or any other memory value we use this command.

alter\_status:

Syntax: alter\_status 'education'

* Through this command, you can get the status of the alter command
* Which indicates the number of regions of the table that have received the updated schema pass table name
* Here in above screen shot it shows 1/1 regions updated. It means that it has updated one region. After that if it successful it will display comment done.



Disable:

* This command will start disabling the named table
* If table needs to be deleted or dropped, it has to disable first

Syntax: disable <tablename>

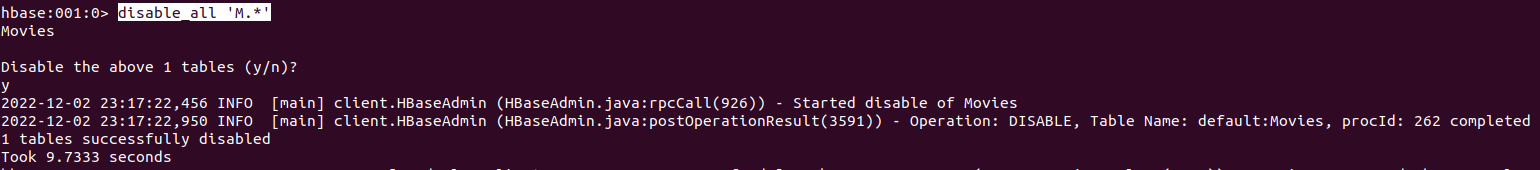


Disable\_all:

* This command will disable all the tables matching the given regex.
* The implementation is same as delete command (Except adding regex for matching)
* Once the table gets disable the user can able to delete the table from HBase
* Before delete or dropping table, it should be disabled first

Syntax: disable\_all<"matching regex"

﻿disable\_all 'M.\*'

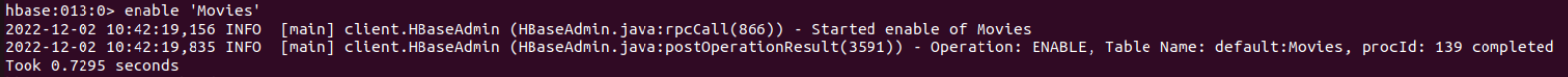


* Disables table movies as it starts with letter ‘M’

Ebable:

Syntax: enable <tablename>

* This command will start enabling the named table
* Whichever table is disabled, to retrieve back to its previous state we use this command
* If a table is disabled in the first instance and not deleted or dropped, and if we want to re-use the disabled table then we have to enable it by using this command.



﻿ Ebable\_all:

Enable all of the tables matching the given regex

enable\_all<"matching regex"

enable\_all 'M.\*'



Is\_enabled:

Syntax: is\_enabled 'Movies'

This command will verify whether the named table is enabled or not. Usually, there is a little confusion between “enable” and “is\_enabled” command action, which we clear here

* Suppose a table is disabled, to use that table we have to enable it by using enable command
* is\_enabled command will check either the table is enabled or not



Is\_disabled:

Syntax: is\_disabled 'Movies'

* Use is\_disabled to check if the table is disabled. When it disabled it returns ‘true‘

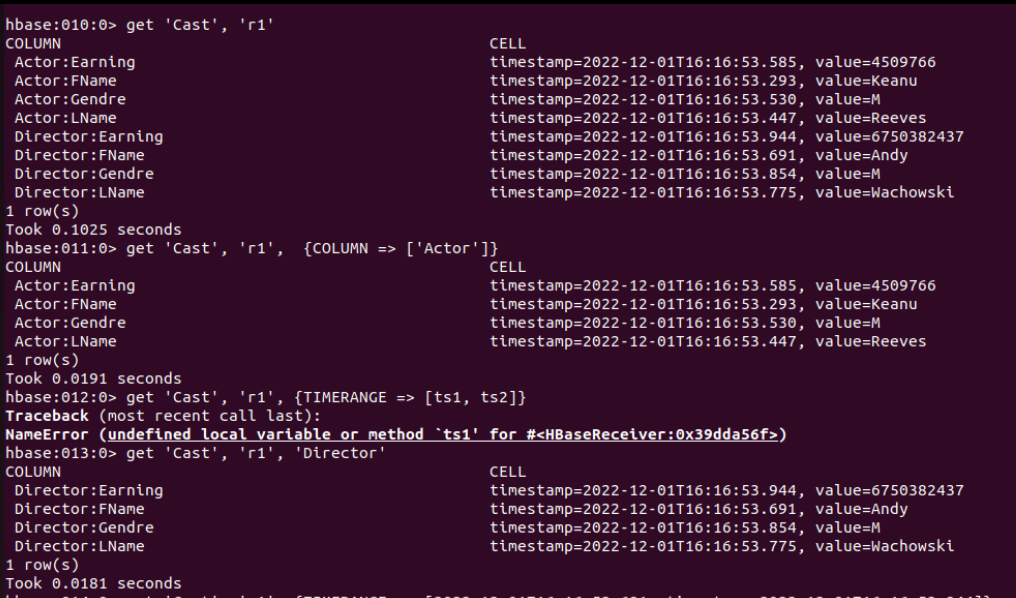
Text

Description automatically generated

Get:

Syntax: get <'tablename'>, <'rowname'>, {< Additional parameters>}

* Here <Additional Parameters> include TIMERANGE, TIMESTAMP, VERSIONS and FILTERS.
* By using this command, you will get a row or cell contents present in the table. In addition to that you can also add additional parameters to it like TIMESTAMP, TIMERANGE,VERSIONS, FILTERS, etc. to get a particular row or cell content.



﻿get 'Movies', 'r1', {TIMERANGE => [235445, 10]}

﻿get 'Movies', 'r1', {TIMERANGE => [100000000, 100000000000000]}

﻿get 'Movies', 'r1', {TIMERANGE => [10, 100000]}

Graphical user interface, text, application, chat or text message

Description automatically generated

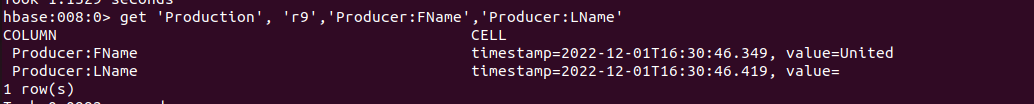
get 'Production', 'r9'

A screenshot of a computer

Description automatically generated with medium confidence

Get row 9 of table Production

﻿get 'Production', 'r9','Producer:FName','Producer:LName'



Get Fname and Lname of row 9 from production table

get 'Cast', 'r3',’Actor’

get 'Cast', 'r4',{column => ’Director’}

Text

Description automatically generated

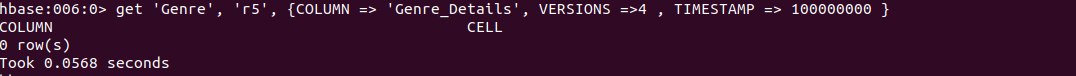
Get Director column of row 4 from table Cast

get ‘Genre’, ‘r5’ {column => ‘Genre\_Details’, VERSIONS >=4}

Text

Description automatically generated with medium confidence

﻿get 'Genre', 'r5', {COLUMN => 'Genre\_Details', VERSIONS =>4 , TIMESTAMP => 100000000 }



Count:

Syntax: count <'tablename'>, CACHE =>1000

* The command will retrieve the count of a number of rows in a table. The value returned by this one is the number of rows.
* Current count is shown per every 1000 rows by default.
* Count interval may be optionally specified.
* Default cache size is 10 rows.
* Count command will work fast when it is configured with right Cache.

count <'Cast'>, CACHE =>1000

count <'Cast'>, INTERVAL =>10

count <'Cast'>, CACHE =>1000, INTERVAL =>10

Text

Description automatically generated

Text

Description automatically generated with medium confidence

* Count ‘Production’
* Returns the number of rows in a table.
* Count ‘xyz’
* When table is not found it returns an error ‘Unknown table’

Show\_filter:

Syntax: show\_filters

This command displays all the filters present in HBase like ColumnPrefix Filter, TimestampsFilter, PageFilter, FamilyFilter, etc.

Text

Description automatically generated

Drop:

Syntax:drop <table name>

* To delete the table present in HBase, first we have to disable it
* To drop the table present in HBase, first we have to disable it
* So either table to drop or delete first the table should be disable using disable command

Text

Description automatically generated

* Drop ‘Person’
* To Drop person table you have to first disable it
* Check if Person table exists result is FALSE

Drop\_all:

Syntax: drop\_all<"regex">

* This command will drop all the tables matching the given regex
* Tables have to disable first before executing this command using disable\_all
* Tables with regex matching expressions are going to drop from HBase

Text

Description automatically generated

* ﻿drop\_all 'P.\*'
* Production table did not drop because it is not disabled

Delete:

Syntax:delete <'tablename'>,<'row name'>,<'column name'>

* This command will delete cell value at defined table of row or column.
* Delete must and should match the deleted cells coordinates exactly.
* When scanning, delete cell suppresses older versions of values.

﻿delete 'Movies','r11', 'Movie\_Details:Title'



* Specific data in table Movies row r11 'Movie\_Details:Title' got deleted

Deleteall:

Syntax: deleteall <'tablename'>, <'rowname'>

* This Command will delete all cells in a given row.
* We can define optionally column names and time stamp to the syntax.



* ﻿deleteall 'Production', 'r10'
* Row r10 got deleted from table Production

Exists:

* Use exists command to verify if the table present in the database.

Syntax: exists ‘<table\_name>’

Shape, rectangle

Description automatically generated with medium confidence

* Returns ‘false‘ when table not present in the database.

Truncate:

Syntax: truncate <tablename>

After truncate of an hbase table, the schema will present but not the records. This command performs 3 functions; those are listed below

* Disables table if it already presents
* Drops table if it already presents
* Recreates the mentioned table

﻿truncate 'Person'

Text

Description automatically generated

* Person table got deleted and then created again

Self Assesment:

Graphical user interface, application

Description automatically generated

Done

Done

Done

Done

Done

Done

Done