

Ejercicios de HBase

Usando la SHELL de Hbase

1. Comencemos abriendo la terminal.

2. Abrimos Hbase

```
[cloudera@quickstart ~]$ hbase shell
2024-07-16 10:23:52,420 INFO [main] Configuration.deprecation: hadoop.native.lib
is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.13.0, rUnknown, Wed Oct 4 11:16:18 PDT 2017
```

3. Vamos a crear nuestra primera tabla.

Crear una tabla llamada `primeraTabla` que tenga una columna de familia llamada `primeraFamilia`.

```
hbase(main):001:0> create 'primeraTabla', 'primeraFamilia'
0 row(s) in 1.4660 seconds

=> Hbase::Table - primeraTabla
```

4. Comprueba que la tabla se ha creado correctamente. Puedes usar el comando `describe`.

```
hbase(main):002:0> describe primeraTabla
NameError: undefined local variable or method `primeraTabla' for #<Object:0x7874
d9d>

hbase(main):003:0>
hbase(main):004:0* describe 'primeraTabla'
Table primeraTabla is ENABLED
primeraTabla
COLUMN FAMILIES DESCRIPTION
{NAME => 'primeraFamilia', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'ROW',
REPLICATION_SCOPE => '0', VERSIONS => '1', COMPRESSION => 'NONE', MIN_VERSIONS =
> '0', TTL => 'FOREVER', KEEP_DELETED_CELLS => 'FALSE', BLOCKSIZE => '65536', IN
MEMORY => 'false', BLOCKCACHE => 'true'}
1 row(s) in 0.1780 seconds
```

5. Creamos un namespace de nombre prueba:

```
hbase(main):005:0> create_namespace 'prueba'
0 row(s) in 0.0570 seconds
```

6. Deshabilitamos la tabla que hemos creado anteriormente.

```
hbase(main):006:0> disable 'primeraTabla'
0 row(s) in 2.2740 seconds
```

7. Borrar la primera tabla que hemos creado.

```
hbase(main):007:0> drop 'primeraTabla'
0 row(s) in 1.2650 seconds
```

8. Crea una nueva tabla llamada ciudad.

```
hbase(main):008:0> create 'ciudad', 'cf1'
0 row(s) in 1.2280 seconds

=> Hbase::Table - ciudad
```

9. Modifica la tabla ciudad y añade una nueva familia de columna que se llame localización.

```
hbase(main):022:0> disable 'ciudad'
0 row(s) in 2.2620 seconds

hbase(main):023:0> alter 'ciudad', 'localizacion'
Updating all regions with the new schema...
1/1 regions updated.
Done.
0 row(s) in 1.9040 seconds

hbase(main):024:0> enable 'ciudad'
0 row(s) in 1.2870 seconds
```

10. Intenta ver todas las tablas del sistema.

```
hbase(main):010:0> list
TABLE
ciudad
1 row(s) in 0.0140 seconds

=> ["ciudad"]
```

11. Modificar la familia de columna localización para que solo guarde una versión.

```
hbase(main):026:0> disable 'ciudad'
0 row(s) in 2.2520 seconds

hbase(main):027:0> alter 'ciudad', NAME => 'localizacion', VERSIONS => 1
Updating all regions with the new schema...
1/1 regions updated.
Done.
0 row(s) in 1.9040 seconds

hbase(main):028:0> enable 'ciudad'
0 row(s) in 1.2540 seconds
```

12. Borra la familia localización de la tabla ciudad.

```
hbase(main):029:0> disable 'ciudad'
0 row(s) in 2.2650 seconds

hbase(main):030:0> alter 'ciudad', NAME => 'localizacion', METHOD => 'delete'
Updating all regions with the new schema...
1/1 regions updated.
Done.
0 row(s) in 1.9060 seconds

hbase(main):031:0> enable 'ciudad'
0 row(s) in 1.2490 seconds
```

13. Comprueba que los cambios se han realizado.

```
hbase(main):013:0> describe 'ciudad'
Table ciudad is ENABLED
ciudad
COLUMN FAMILIES DESCRIPTION
{NAME => 'cf1', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'ROW', REPLICATION_SCOPE => '0', VERSIONS => '1', COMPRESSION => 'NONE', MIN_VERSIONS => '0', TTL => 'FOREVER', KEEP_DELETED_CELLS => 'FALSE', BLOCKSIZE => '65536', IN_MEMORY => 'false', BLOCKCACHE => 'true'}
1 row(s) in 0.0260 seconds
```

Acceso a datos

1. Abre la Shell de Hbase.

```
2024-07-16 10:43:26,715 INFO [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.13.0, rUnknown, Wed Oct 4 11:16:18 PDT 2017

hbase(main):001:0> █
```

2. Creamos una tabla llamada mundo, con dos familias de columnas; ciudad y población.

```
hbase(main):001:0> create 'mundo', 'ciudad', 'poblacion'
0 row(s) in 1.4280 seconds

=> Hbase::Table - mundo
hbase(main):002:0> █
```

3. Obtener la fila 1000 de la tabla mundo. Debería no existir.

```
hbase(main):002:0> get 'mundo', '1000'
COLUMN          CELL
0 row(s) in 0.1140 seconds

hbase(main):003:0> █
```

4. Introducimos en la tabla mundo una fila con las siguientes características:

- Nombre de la tabla: 'mundo'
- Clave: '10'
- Column Family: 'ciudad'
- Column descriptor: 'nombre' y el valor 'Pamplona'
- Column descriptor: 'region' y el valor 'Navarra'

```
hbase(main):003:0> put 'mundo', '10', 'ciudad:nombre', 'Pamplona'
0 row(s) in 0.0590 seconds
```

```
hbase(main):004:0> put 'mundo', '10', 'ciudad:region', 'Navarra'
0 row(s) in 0.0090 seconds
```

5. Comprueba que la tabla se ha creado correctamente.

```
hbase(main):005:0> get 'mundo', '10'
COLUMN                                CELL
ciudad:nombre                        timestamp=1721119649299, value=Pamplona
ciudad:region                        timestamp=1721119683397, value=Navarra
2 row(s) in 0.0130 seconds
```

6. Añade ahora a esa fila, otros dos descriptores; uno para las coordenadas del eje X y otro para las coordenadas del eje Y. Los valores que debes introducir son: 42°49'00"N y 1°39'00"O respectivamente. Para no tener problemas con el formato de las coordenadas introducir los números así: 42.49.00.

```
hbase(main):007:0> put 'mundo', '10', 'ciudad:coordenadasX', '42.49.00'
0 row(s) in 0.0050 seconds

hbase(main):008:0> put 'mundo', '10', 'ciudad:coordenadasY', '1.39.00'
0 row(s) in 0.0040 seconds
```

7. Además, añade a esa fila la población de pamplona: 197 138 habitantes.

```
hbase(main):013:0> put 'mundo', '10', 'poblacion:habitantes', '197138'
0 row(s) in 0.0080 seconds
```

8. Comprueba finalmente que los datos se han insertado correctamente.

```
hbase(main):014:0> get 'mundo', '10'
COLUMN                                CELL
ciudad:coordenadasX                  timestamp=1721119880153, value=42.49.00
ciudad:coordenadasY                  timestamp=1721119904118, value=1.39.00
ciudad:nombre                        timestamp=1721119649299, value=Pamplona
ciudad:region                        timestamp=1721119683397, value=Navarra
poblacion:habitantes                 timestamp=1721121956514, value=197138
5 row(s) in 0.0290 seconds
```

9. Supongamos que la población de Pamplona ha cambiado en los últimos años y hay que actualizar la base de datos. Ahora Pamplona debe tener 200.000 habitantes.

```
hbase(main):022:0> put 'mundo', '10', 'poblacion:habitantes', '200000'
0 row(s) in 0.0100 seconds
```

10. Obtener la fila con clave 10 (Pamplona), mostrando los 3 últimos valores de habitantes que han sido almacenados.

```
hbase(main):023:0> get 'mundo', '10', {COLUMN => 'poblacion:habitantes', VERSIONS => 3}
COLUMN                                CELL
poblacion:habitantes                 timestamp=1721122431407, value=200000
1 row(s) in 0.0100 seconds
```

11. Borrar los campos de coordenadas.

```
hbase(main):024:0> delete 'mundo', '10', 'ciudad:coordenadasx'
0 row(s) in 0.0370 seconds

hbase(main):025:0> delete 'mundo', '10', 'ciudad:coordenadasy'
0 row(s) in 0.0110 seconds
```

12. Comprueba que se han borrado dichos campos.

```
hbase(main):026:0> get 'mundo', '10'  
COLUMN          CELL  
ciudad:coordenadasX timestamp=1721119880153, value=42.49.00  
ciudad:coordenadasY timestamp=1721119904118, value=1.39.00  
ciudad:nombre    timestamp=1721119649299, value=Pamplona  
ciudad:region    timestamp=1721119683397, value=Navarra  
poblacion:habitantes timestamp=1721122431407, value=200000  
5 row(s) in 0.0130 seconds
```

13. A continuación, borra la fila entera correspondiente a Pamplona (key = 10)

```
hbase(main):027:0> deleteall 'mundo', '10'  
0 row(s) in 0.0080 seconds
```

14. Comprueba que la operación se ha realizado correctamente.

```
hbase(main):028:0> get 'mundo', '10'  
COLUMN          CELL  
0 row(s) in 0.0060 seconds
```

Usando el API del desarrollador

He empezado a hacerlo pero me he encontrado con que no estan los proyectos instalados, por lo que he intentado crearlo, pero necesito unos jars que no puedo obtener debido a restricciones del navegador.

He podido instalar dos .jar pero necesitaria mas para el proyecto, y el navegador me pone lo siguiente:

mvnrepository.com

⚠ Your browser is out of date!

Update your browser to view this website correctly. [More Information.](#)

1. Abre una terminal.

2. Los proyectos de eclipse para los ejercicios ya han sido instalados pero debemos refrescar. Por ello, introduce el siguiente comando:

3. Busca el icono de eclipse en tu escritorio y haz doble click.

4. Para crear un archivo jar, haz click derecho en el nombre del proyecto y selecciona Export -> Java -> jarFile y después hacer click en Next. El archivo se creará en el directorio por defecto. De esta forma, podremos ejecutar el proyecto:

5. Para comenzar repasaremos un poco la sintaxis de Java. Para ello, dentro del proyecto Mundo, en el paquete solution, crearemos una clase llamada Ciudad. Esa clase tendrá que tener tres atributos privados; el nombre de la ciudad (string), el país (string) y la población (int). Hacer también los constructores y los getters y setters.