

TP n9 en Mise en œuvre des Framework d'IA et Big data

Hbase

- Utilisation du référentiel <https://github.com/big-data-europe/docker-hbase>
`C:\>git clone https://github.com/big-data-europe/docker-hbase.git`
- Lancer Docker desktop
 Accéder à la ligne de commande du container cloner, et lancer docker-compose-distributed-local.yml up -d
`C:\docker-hbase-master>docker-compose -f docker-compose-distributed-local.yml up -d`
- Lister le detail de chaque container demmare (ID, nPort, nom, ...) :docker container ls
- Vérification que quelque chose est à l'écoute : http://localhost:16010/master-status
- Accéder à la ligne de commande du serveur Hbase et démarrer hbase-regionserver
`C:\docker-hbase-master>docker exec -it hbase-regionserver bash`
- Lancer le shell de HBase: ...] \$ **hbase shell**
`root@hbase-regionserver:/# hbase shell`
- Créer la table emp ayant deux familles de colonnes : « **données personnelles** » et « **données professionnelles** »
`hbase(main):001:0> create 'emp', 'personalData', 'professionalData'`
 OU
`hbase(main):001:0> > create 'emp', {NAME=>'personneData'}, {NAME=>'professionnelData'}`
- Vérifier la création de la table en listant les noms de tables existantes : hbase> list
`hbase> list 'emp'`
 ou hbase(main):012:0> exists 'emp' pour vérifier l'existence de la table emp
- Afficher la structure de la table emp:
`hbase> describe 'emp'`

On aura les propriétés de chaque famille de colonnes (VERSIONS, COMPRESSION, ...)

10. Modification de la structure de la table emp

Alter est la commande utilisée pour apporter des modifications à une table existante. À l'aide de cette commande, on peut modifier le nombre maximal de cellules d'une famille de colonnes, définir et supprimer des opérateurs d'étendue de table et supprimer une famille de colonnes d'une table.

Ajout une autre famille de colonnes	<code>alter 'table_name', {NAME => 'family_name'}</code>
Modification du nombre maximal de cellules d'une famille de colonnes	<code>alter 'table_name', NAME => 'f1', VERSIONS => 5</code>
Rendre une table en lecture seule.	<code>alter 'table_name', READONLY</code>
Suppression d'une famille de colonnes	<code>alter 'table_name', 'delete' => 'column family'</code>
Suppression d'opérateurs d'étendue de table	<code>alter 'table_name', METHOD => 'table_att_unset', NAME => 'MAX_FILESIZE'</code>

a. Modifier le nombre maximal de cellules d'une famille de colonnes, et vérifier la modification

`hbase(main):008:0> alter 'emp', NAME => 'personnalData', VERSIONS => 5`

b. Ajouter la famille de colonnes departement, et verifier

`hbase> alter 'emp', {NAME => 'departement'}`

11. Insertion des données avec la commande put: put 'HBase_table_name', 'row_key', 'colfamily:colname', 'value'

<code>put 'emp','1','personalData:ename','Miller'</code>	<code>put 'emp','2','personalData:ename','Clark'</code>
<code>put 'emp','1','personalData:hiredate','23-01-1982'</code>	<code>put 'emp','2','personalData:hiredate','09-06-1981'</code>
<code>put 'emp','1','professionalData:job','Clerck'</code>	<code>put 'emp','2','professionalData:job','Manager'</code>
<code>put 'emp','1','professionalData:sal','1300'</code>	<code>put 'emp','2','professionalData:sal','2450'</code>
<code>put 'emp','1','departement:deptno','10'</code>	<code>put 'emp','2','departement:deptno','10'</code>

Insérer les lignes restantes à l'aide de la commande put de la même manière

<code>put 'emp','3','personalData:ename','King'</code>	<code>put 'emp','4','personalData:ename','Scott'</code>
<code>put 'emp','3','personalData:hiredate','17-11-1981'</code>	<code>put 'emp','4','personalData:hiredate','13-07-1987'</code>
<code>put 'emp','3','professionalData:job','President'</code>	<code>put 'emp','4','professionalData:job','Analyst'</code>
<code>put 'emp','3','professionalData:sal','5000'</code>	<code>put 'emp','4','professionalData:sal','3000'</code>
<code>put 'emp','3','departement:deptno','10'</code>	<code>put 'emp','4','departement:deptno','20'</code>

put 'emp','5','personalData:ename','Ford' put 'emp','5','personalData:hiredate','03-12-1981' put 'emp','5','professionalData:job','Analyst' put 'emp','5','professionalData:sal','3000' put 'emp','5','departement:deptno','20'	put 'emp','6','personalData:ename','Smith' put 'emp','6','personalData:hiredate','17-12-1980' put 'emp','6','professionalData:job','Clerck' put 'emp','6','professionalData:sal','800' put 'emp','6','departement:deptno','20'
---	--

12. Lire les données de la table emp avec la commande get: get '<table name>','row1'

hbase> get 'emp', '1'

13. Lire une colonne spécifique : get 'table name', 'rowid', {COLUMN => 'column family:column name' }

hbase> get 'emp', '1', {COLUMN => 'personalData:ename' }

14. Modifier la valeur de salaire de l'employé de numero '1'

Hbase>put 'emp','1','professionalData:sal','4000'

15. Lire les données avec la commande scan: scan 'table_name', {attribute => 'value', ... }

hbase> scan 'emp'

hbase> scan 'emp', {COLUMN => 'personalData:ename' }

16. Compter le nombre de lignes dans la table emp : count '<table name>'

17. Utilisation de quelques filtres dans la commande scan via l'attribut FILTER:

a. Lister toutes les informations de l'employé 2

hbase>scan 'emp', {FILTER => "PrefixFilter ('2')"}

b. Lister les employes de salaire = 5000

hbase>scan 'emp', { COLUMNS => 'professionalData:sal', FILTER => "ValueFilter(=,'binary:5000')"}

18. Utilisation du filtre SingleColumnValueFilter :

SingleColumnValueFilter ('<family>', '<qualifier>', <compare operator>, '<comparator>')

a. Lister les employes de departement 10:

hbase> scan 'emp',{COLUMNS => ['personalData:ename','professionalData:job','departement:deptno'],
FILTER =>"SingleColumnValueFilter ('departement','deptno','=', 'binary:10') " }

b. Lister les employes de job Analyst

hbase> scan 'emp',{COLUMNS => ['personalData','professionalData'], FILTER
=>"SingleColumnValueFilter ('professionalData','job','=', 'regexstring:Analyst') " }

c. Lister les employes de departement 10 et de fonction Analyst :

hbase> scan 'emp',{COLUMNS =>
['personalData:ename','professionalData:job','departement:deptno'],FILTER =>"SingleColumnValueFilter
('departement','deptno','=', 'binary:10') AND
SingleColumnValueFilter('professionalData','job','=', 'regexstring:Analyst') " }

19. Suppression des lignes, des cellules ou des familles de cellules :

Suppression d'une cellule spécifique d'un tableau	delete 'table_name', 'row_key', 'column_name', time_stamp_value
Suppression de toutes les cellules	deleteall 'table_name', 'row_key'
Supprimer des familles de colonnes	alter 'table_name', {NAME => 'column_familly', METHOD => 'delete' ou alter 'table_name', 'delete'=> 'column_familly'

a. Supprimer la colonne ename de la cellule personalData, pour employé de numero 2,

hbase> delete 'emp','2','personalData:ename',1701441458637

b. Supprimer toutes les cellules du premier employé

hbase> deleteall 'emp', '1'

c. Supprimer la familles de colonnes 'departement', et verifier la suppression

hbase> alter 'emp', 'delete'=> 'departement'

20. Creation de référence à une table : référence = get_table 'table_name'

hbase> t1 = get_table 'emp'

hbase> t1.count

hbase> t1.scan FILTER => "PrefixFilter ('2') AND ColumnPrefixFilter('na')"

hbase> t1.put '7', 'personalData:ename', 'Ford'

21. Vider la table emp avec la commande : hbase> truncate 'emp' (Remarquer les étapes de vidage).

22. Verifier la suppression

23. Supprimer la table emp avec la commande : hbase> drop 'emp'

Que se passe-t-il?

24. Désactiver la table avec la commande disable et refaire la suppression.

25. Vérifier si la table emp existe toujours ou non : hbase> exists 'emp'

Exercice 2.

1. Lancer le shell de Hbase
2. Afficher le nom d'utilisateur connecté
3. Créer une table nommée htable avec 3 familles de colonnes
4. Lister la structure de la table
5. Désactiver le tableau
6. Modifier le tableau à l'aide de ces commandes :
 - a. modifier 'htable', {NAME => 'col_fam1', COMPRESSION => 'GZ' }
 - b. modifier 'htable', {NAME => 'col_fam1', IN_MEMORY => 'true'}
 - c. modifier 'h_table', {NAME => 'col_fam1', VERSIONS => 3}, {NAME => 'col_fam2', VERSIONS => 2} }
7. Activer le tableau
8. Insérer quelques valeurs dans le tableau à l'aide de la commande put
9. Compter le nombre de lignes dans le tableau
10. Lister la ligne 1 du tableau
11. Scanner le tableau
12. Modifier certaines valeurs dans le tableau à l'aide de la commande put
13. Scanner le tableau

Exercice 3.

1. Copier le fichier (emp_data.csv) dans le conteneur namenode
`C:\docker-hbase-master>docker cp D:\MyData\Cours\BigData\Data\emp_data.csv namenode:/tmp/`
2. Accéder au conteneur namenode et exécuter-le de manière interactive à l'aide de la commande suivante ;
`c:\>docker-hadoop-spark>docker exec -it namenode bash`
3. Dans le terminal bash résultant, créer le dossier d'entrée input pour héberger des fichiers
`root@fdc43977c871:/# hdfs dfs -mkdir -p /user/cloudera/hbase_data`
`root@fdc43977c871:/# cd tmp`
4. Copier le fichier emp.csv de /tmp vers le dossier d'entrée
`root@fdc43977c871:/tmp# hdfs dfs -put emp_data.csv /user/root/input`
Visualiser le fichier sur le site : **<http://localhost:50070/explorer.htm>**
5. Lancer le shell HBase, et créer la table Hbase: emp_data (pers, prof)
6. La commande ImportTsv permet de charger un fichier csv dans une table HBase.
Utiliser la commande suivante dans un terminal (Shell Linux):

```
[cloudera@quickstart ~]$ hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -
Dimporttsv.separator=', ' -
Dimporttsv.columns='HBASE_ROW_KEY,pers:ename,pers :manager,pers:hire_date,prof:designatio
n,prof:sal,prof:deptno' emp_data /user/cloudera/hbase_data/emp_data.csv
```

7. Faire un scan de la table emp_data
8. Afficher la ligne dont le Row Key est '7844'
9. Afficher les employes ayant les salaires supérieurs à 2000
10. Afficher les noms des employes des fonctions ‘
11. Afficher les noms des employes recrutés après la date 30/12/1982
12. Désactiver la table emp_data et faire ensuite un scan pour avoir son contenu.
13. Activer la table emp_data avec la commande enable et faire ensuite un scan