Travaux pratiques : HBase

Jonathan Lejeune

Objectifs

Ce sujet de travaux pratiques vous permettra de vous initier à l'utilisation de Hbase dans un environnement UNIX. Pour faire ce TP vous n'oublierez pas au préalable de démarrer le HDFS (le reformater si nécessaire) et de démarrer Hbase avec 2 RegionServer (cf. TP préparatoire).

Vous pouvez au choix coder en Java ou en scala.

Exercice 1 – Création et remplissage d'une table Hbase

Dans cet exercice nous allons créer une table HBase pour stocker des données de LastFM . Nous souhaitons donc avoir la table *ecoute* ayant le schéma suivant :

	cf1				
rowkey	UserId	TrackId	LocalListening	RadioListening	Skip
...	...	...	...	...	...

On considérera une seule famille de colonne appelée cf1

Question 1

Grâce au générateur de données fourni (nécessite une version $\geq$ 2.11 de scala), produire un fichier texte de 1 ko respectant le format des logs de LastFM :

```
generateTextfile.sh -t lastfm -n 1 -s 1
```

Question 2

Nous considérerons dans cette question qu'il n 'y a pas d'accès concurrent à la base. Écrire un programme (java ou scala) :

- créant la table `ecoute` si cette dernière n'existe pas déjà dans le namespace par défaut.
- lit ligne par ligne le fichier généré précédemment
- pour chaque ligne lue, ajoute une nouvelle entrée dans la table. Le rowkey sera une concaténation du `UserId` et du `TrackId` . Il faudra prendre en compte le cas où la clé existe déjà : dans ce cas ne pas écraser l'ancienne valeur mais incrémenter les compteurs existants. Pour tester si une rowkey existe il faut utiliser la méthode `isEmpty` de la classe `Result`.

NB : Le tableau ci-dessous vous indique comment lire un fichier ligne par ligne.

En java	`BufferedReader br = new BufferedReader(new FileReader("fichier"));`
	`String line;`
	`while ((line = br.readLine()) != null) {`
	`  //traitement de line`
	`}`
	`br.close();`

En scala	`for(line <- Source.fromFile("fichier").getLines){`
	`  //traitement de line`
	`}`

Question 3

Repérez dans le fichier texte généré précédemment deux lignes ayant les mêmes userid et trackid.

Question 4

Ouvrez un shell Hbase :

`hbase shell`

et affichez le contenu de la table `ecoute`

`scan 'ecoute'`

Question 5

Contrôlez la cohérence des valeurs dans la base par rapport aux lignes précédemment identifiées à la question 3.

Question 6

Que se passerait-il si on lançait ce programme plusieurs fois en parallèle ?

Question 7

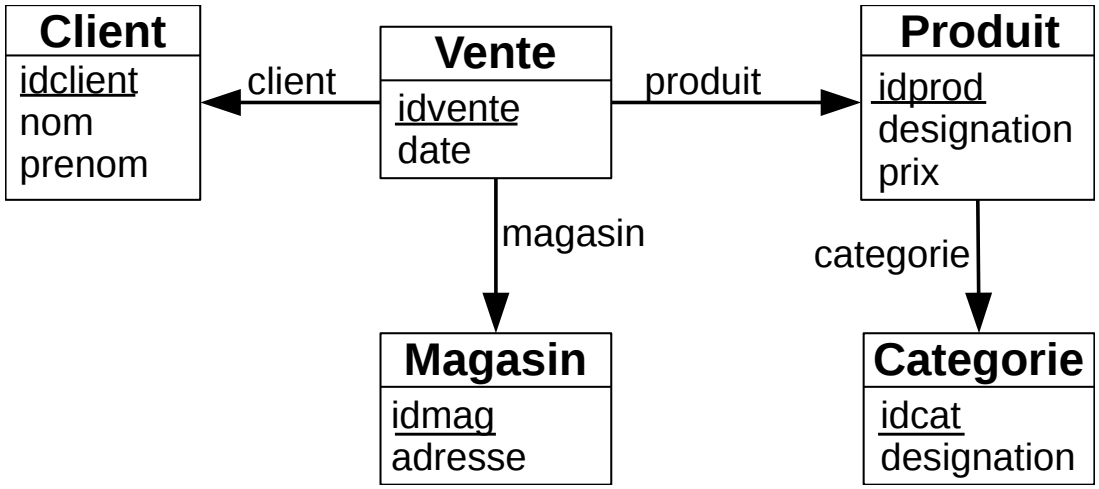
Pour résoudre le problème de la question précédente, modifiez votre programme en utilisant des mutations de type `Increment`. Vous trouverez une documentation à cette URL :

https://hbase.apache.org/apidocs/org/apache/hadoop/hbase/client/Increment.html.

NB : Il faudra que les colonnes `LocalListening`, `RadioListening` et `Skip` soient la représentation d'un long.

Exercice 2 – Requête de jointure et dénormalisation de schéma

On s'intéresse dans cet exercice à une base de données normalisée dont le schéma est donné ci-dessous.



La schéma considéré modélise une enseigne de commerce. Un produit mise à la vente est associé à une seule catégorie. L'enseigne possède plusieurs magasins et enregistre la liste de ses clients. La vente d'un produit lie le produit acheté, le magasin dans lequel l'achat a eu lieu, le client concerné ainsi que la date. Pour appliquer ce schéma dans Hbase, on considérera donc 5 tables où le rowkey d'une ligne sera égale à l'identifiant de l'élément. On considérera une seule famille de colonne par table qu'on appellera `defaultcf`. Ainsi on a les tables suivantes (avec une ligne d'exemple) :

- la table **client** :

	defaultcf		
rowkey	**idclient**	**nom**	**prenom**
client42	client42	Dupont	Paul
...	...	...	...

- la table **magasin** :

	defaultcf	
rowkey	**idmag**	**adresse**
magasin21	magasin21	Paris15ème
...	...	...

- la table **categorie** :

	defaultcf	
rowkey	**idcat**	**designation**
categorie12	categorie12	informatique
...	...	...

- la table **produit** :

	defaultcf			
rowkey	**idprod**	**designation**	**prix**	**categorie**
produit10	produit10	clé usb	19.90	categorie12
...	...	...	...	...

- la table **vente** :

	defaultcf				
rowkey	**idvente**	**client**	**produit**	**magasin**	**date**
vente34	vente34	client42	produit10	magasin21	12/12/2012 16 :34
...	...	...	...	...	...

Question 1

Générez ce schéma avec le script `generateSchemaVente.sh` et vérifiez grâce au shell hbase que toutes les tables décrites précédemment existent bien et sont correctement remplies.

Question 2

Écrire un programme qui affiche le nombre de ventes par nom de catégorie.

Question 3

Chronométrez le temps nécessaire à exécuter cette requête sur la base générée à la question précédente.

Question 4

Proposez deux solutions de dénormalisation de ce schéma pour accélérer le temps de la requête précédente. Assurez-vous que ces solutions soient plus rapides en comparant leur temps d'exécution avec la version normalisée.

Question 5

Que devons-nous faire pour que la base reste cohérente à chaque nouvel enregistrement d'une vente ?