

HiveQL: TRAVAIL PRATIQUE

Databases in Hive

Le concept Hive d'une base de données est essentiellement un **catalogue** ou un namespace de tables. Cependant, ils sont très utiles pour les clusters avec plusieurs utilisateurs, en tant que moyen d'éviter les collisions de nom de table. Il est également courant d'utiliser des bases de données pour organiser les tables de production en groupes logiques.

Si vous ne spécifiez pas de base de données, la base de données **par défaut** est utilisée.

La syntaxe la plus simple pour créer une base de données est illustrée dans l'exemple suivant:

```
hive> CREATE DATABASE financials;
```

Hive lancera une erreur si les données financières existent déjà. Vous pouvez supprimer ces avertissements avec cette variante:

```
hive> CREATE DATABASE IF NOT EXISTS financials;
```

A tout moment, vous pouvez voir les bases de données existantes comme suit:

```
hive> SHOW DATABASES;
default
financials
```

```
hive> CREATE DATABASE human_resources;
```

```
hive> SHOW DATABASES;
default
financials
human_resources
```

Si vous avez plusieurs base de données vous pourrez utiliser les restrictions avec LIKE

```
hive> SHOW DATABASES LIKE 'h.*';
human_resources
hive> ...
```

Hive créera un **répertoire** pour chaque **base de données**. Les **tables** de cette **base de données** seront stockées dans **sous-répertoires** du **répertoire** de la **base de données**. L'exception concerne les **tables** de la **base de données par défaut**, qui n'a pas son propre répertoire.

Le répertoire de la base de données est créé dans un répertoire de niveau supérieur spécifié par la propriété hive.metastore.warehouse.dir

Vous pouvez remplacer cet emplacement par défaut pour un nouveau répertoire, comme suit :

```
hive> CREATE DATABASE financials
> LOCATION '/my/preferred/directory';
```

Vous pouvez ajouter un commentaire descriptif à la base de données, qui sera affiché par le DESCRIBE DATABASE <base de données>.

```
hive> CREATE DATABASE financials
> COMMENT 'Holds all financial tables';

hive> DESCRIBE DATABASE financials;
financials Holds all financial tables
hdfs://master-server/user/hive/warehouse/financials.db
```

La commande USE définit une base de données comme base de working, analogue à la modification répertoires de travail dans un filesystem :

```
hive> USE financials;
hive> SHOW TABLES;
```

Supprimer une base de données

```
hive> DROP DATABASE IF EXISTS financials;
```

Par défaut, Hive ne vous autorisera pas à supprimer une base de données si elle contient des tables. Tu peux soit supprimer d'abord les tables ou ajoutez le mot clé CASCADE à la commande, ce qui entraînera Hive à supprimer les tables dans la base de données d'abord:

```
hive> DROP DATABASE IF EXISTS financials CASCADE;
```

Création de tables

L'instruction CREATE TABLE suit les conventions SQL, mais la version de Hive offre d'avantages extensions pour prendre en charge un large éventail de flexibilité lorsque les fichiers de données des tables sont stockés, les formats utilisés, etc.

Exemple :

```
CREATE TABLE IF NOT EXISTS mydb.employees (
  name STRING COMMENT 'Employee name',
  salary FLOAT COMMENT 'Employee salary',
  subordinates ARRAY<STRING> COMMENT 'Names of subordinates',
  deductions MAP<STRING, FLOAT>
  COMMENT 'Keys are deductions names, values are percentages',
  address STRUCT<street:STRING, city:STRING, state:STRING, zip:INT>
  COMMENT 'Home address')
COMMENT 'Description of the table'
TBLPROPERTIES ('creator'='me', 'created_at'='2012-01-02 10:00:00', ...)
LOCATION '/user/hive/warehouse/mydb.db/employees';
```

Vous pouvez également copier le schéma (mais pas les données) d'une table existante:

```
CREATE TABLE IF NOT EXISTS mydb.employees2
LIKE mydb.employees;
```

```
hive> USE mydb;
hive> SHOW TABLES;
```

Si nous ne sommes pas dans la même base de données, nous pouvons toujours lister les tables de cette base de données:

```
hive> USE default;
hive> SHOW TABLES IN mydb;
```

Si vous avez plusieurs tables, vous pourrez faire des restrictions en utilisant LIKE

```
hive> USE mydb;
hive> SHOW TABLES LIKE 'empl.*';
employees
```

Pour voir le détail d'une table on utilise DESCRIBE EXTENDED mydb.employees

```
hive> DESCRIBE EXTENDED mydb.employees;
name string Employee name
salary float Employee salary
subordinates array<string> Names of subordinates
deductions map<string,float> Keys are deductions names, values are percentages
address struct<street:string,city:string,state:string,zip:int> Home address
Detailed Table Information Table(tableName:employees, dbName:mydb, owner:me,
...
location:hdfs://master-server/user/hive/warehouse/mydb.db/employees,
parameters:{creator=me, created_at='2012-01-02 10:00:00',
last_modified_user=me, last_modified_time=1337544510,
comment:Description of the table, ...}, ...)
```

Pour uniquement le schéma d'une colonne il faut faire :

```
hive> DESCRIBE mydb.employees.salary;
salary float Employee salary
```