

B I G D A T A



ANALYSE DE DONNÉES AVEC HIVE



AGENDA



➤ ANALYSE DE DONNÉES AVEC HIVE

Objectif

Qu'est-ce qu'Hive ?

Architecture Hive

Hive Metastore et stockage de données

Cas d'utilisation d'Hive

Base de données et Tables Hive

Syntaxe HiveQL basique

Types de données dans Hive

Fonctions communes prédéfinies

Formats de données Hive




Apache Hive



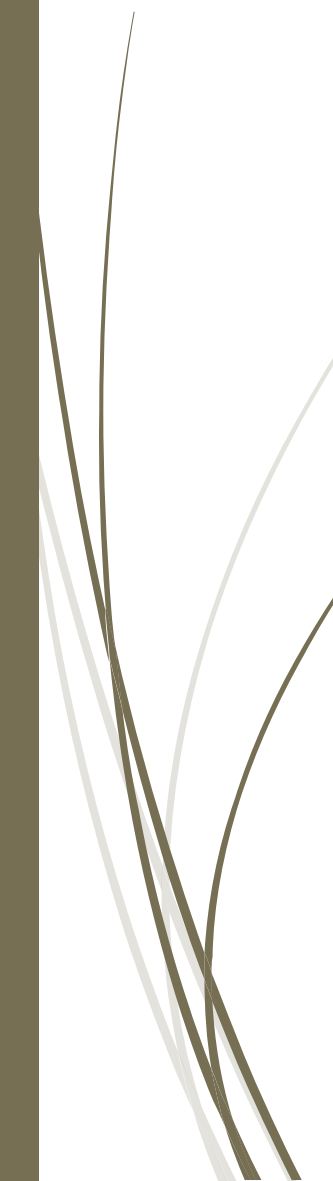


Apache Hive

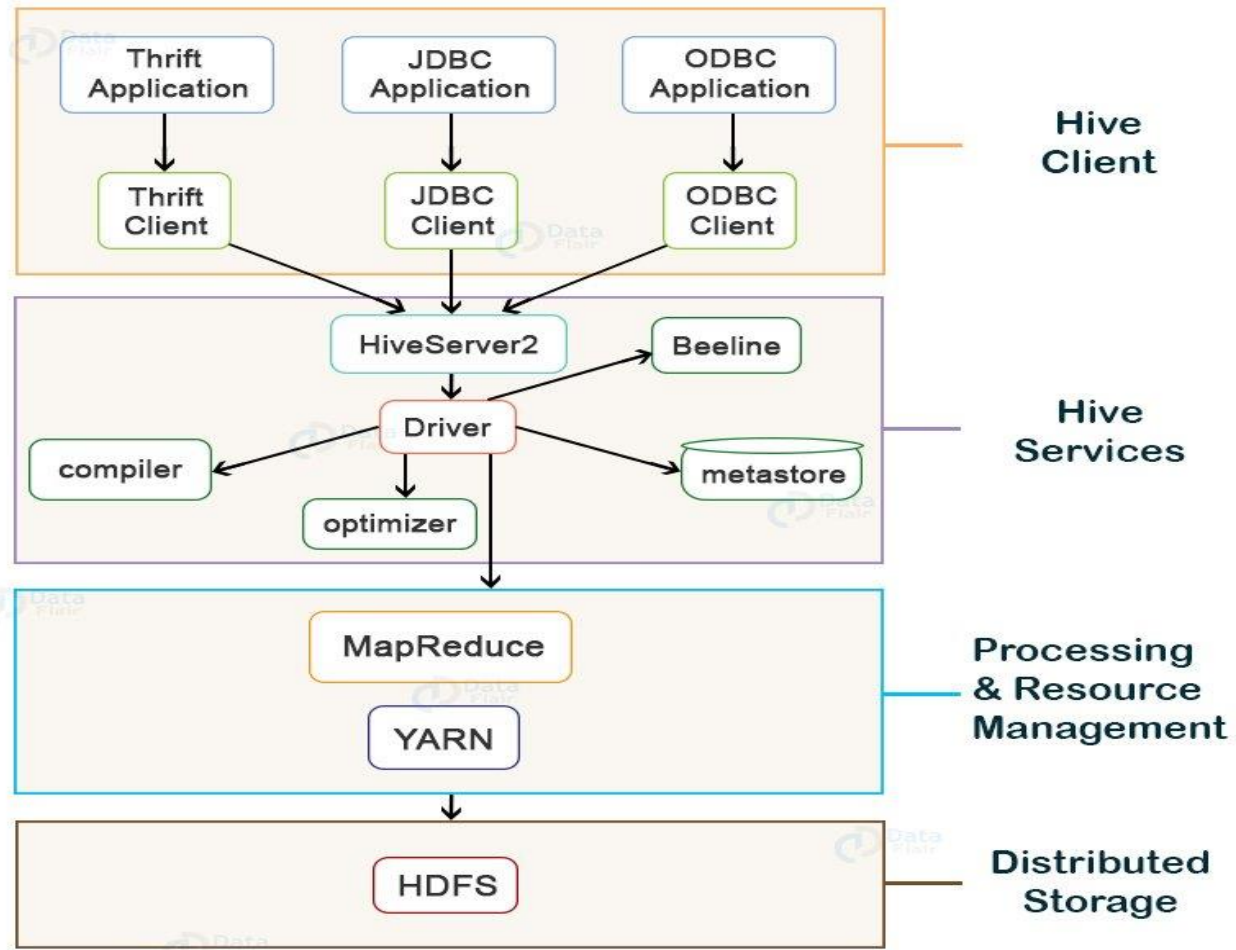
- Hive est un système d'entrepôt de données pour Hadoop.
 - Hive n'est pas une base de données relationnelle; il ne conserve que les informations de métadonnées sur les données volumineuses stockées sur HDFS.
 - Hive permet de traiter les données volumineuses comme des tables et effectuer des opérations de type SQL sur les données à l'aide d'un langage de script appelé HiveQL.
- 



Apache Hive

- Hive a été créé pour réduire la complexité de la programmation MapReduce.
 - Un programme MapReduce de 100 lignes de code peut être résumé en 11 lignes de code peut être résumé en une seule commande Hive.
 - Hive est facile à utiliser par des Analystes qui n'ont pas des connaissances Java
- 


Architecture Hive



Hive Architecture & Its Components



Hive Metastore et stockage de données

- 
- Hive utilise une base de données (appelée metastore) pour stocker les metadatas des données analysées .
 - Une table Hive est constituée d'un schéma stocké dans le metastore et de données stockées sur HDFS
 - Hive convertit les commandes HiveQL en tâches MapReduce ou Tez pour accéder aux données



Hive Metastore et stockage de données

Hive permet de lire ou créer des données sur le HDFS en spécifiant le format :

- ✓ Text File
 - ✓ SequenceFile
 - ✓ RCFile
 - ✓ Avro Files
 - ✓ ORC Files
 - ✓ Parquet
- 



Hive Metastore et stockage de données

- ✓ Hive permet l'accès aux fichiers compressés tel que Gzip, Bzip2.
 - ✓ Hive peut compresser les fichiers au moment du stockage
- 

Hive vs bases des données traditionnelles

Caractéristiques	Hive	Bases de données Traditionnelles
Langage de requêtes	Not Only SQL (NOSql)	SQL
Application du Schéma	Au moment de lecture	Au moment d'écriture
Notion Lecture/Ecriture	Une seule Ecriture et Plusieurs Lectures	Plusieurs Ecritures et Plusieurs Lectures
Manipulation des lignes	Pas de suppression ou Maj (En cours d'amélioration avec l'option ACID)	Tout est possible
Capacité de stockage	Peut atteindre facilement les 100 Peta	10 Tera Maximum
Traitement Transactionnel et Analytique	Pas de OLTP (Online Transaction Processing) Pas idéal pour le OLAP (Online Analytical Processing)	OLTP et OLAP
Scalabilité	Facile et pas chère	Difficile et Couteuse

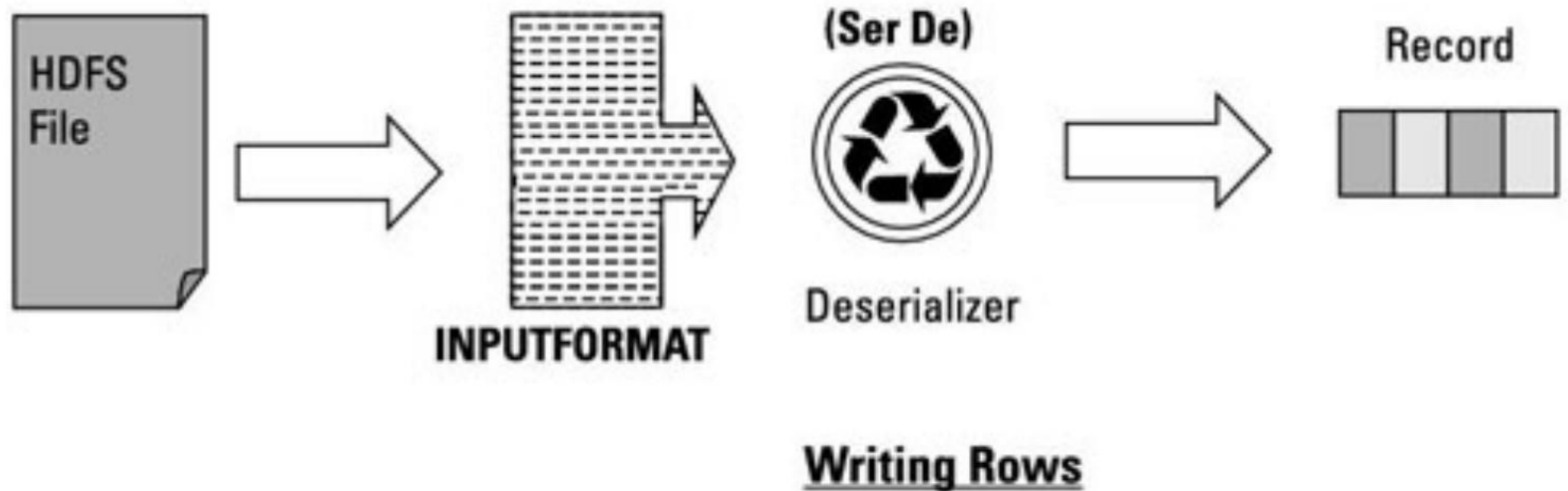


Cas d'utilisation d'Hive

- Hive est bien performant pour l'analyse à froid des données :
 - ✓ Mode batch
 - ✓ Accès aux raw data
 - ✓ Calcul des KPI
 - ✓ Générer de vues métiers
 - ✓ Hive n'est pas adapté pour les traitements au fil de l'eau (streaming)

Hive Metastore et stockage de données

- Hive utilise SerDe (Serializer et Deserializer) pour lire et écrire des lignes dans des tables



BIG DATA

Lab

▲ 18.75

▼ 25.01

▼ 22.10

▲ 07.28

▲ 25.21

▼ 30.12

▲ 29.79

▼ 18.07

▲ 24.78

Lab1 : Interagir avec Hive

- Comment lancer des requêtes Hive ?
 - Commande Hive

```
[root@sandbox-hdp ~]# hive
log4j:WARN No such property [maxFileSize] in org.apache.log4j.DailyRollingFileAppender.

Logging initialized using configuration in file:/etc/hive/2.6.5.0-292/0/hive-log4j.properties
hive> █
```




Base de données et Tables Hive

Hive, comme les SGBDR, permet de manipuler des bases de données :

➤ Création :

- ✓ CREATE DATABASE [IF NOT EXISTS] DB_NAME;
- ✓ CREATE SCHEMA [IF NOT EXISTS] DB_NAME



Base de données et Tables Hive

Suppression ou DROP :

- ✓ `DROP DATABASE [IF EXISTS] DB_NAME [RESTRICT | CASCADE];`
- ✓ `DROP SCHEMA [IF NOT EXISTS] DB_NAME [RESTRICT | CASCADE];`
- ✓ L'option `CASCADE` permet de dropper les tables avant le drop de la base



Base de données et Tables Hive

- Hive permet l'utilisation de 3 types de tables .

Tables Internes :

- ✓ Les metadatas et les données sont gérés par Hive.
- ✓ Les données sont stockées dans le répertoire HDFS de la base.
- ✓ Un drop d'une table interne = suppression de la metadata et les données de la table.



Base de données et Tables Hive

- Hive permet l'utilisation de 3 types de tables .

Tables Internes :

- ✓ Les metadatas et les données sont gérés par Hive.
- ✓ Les données sont stockées dans le répertoire HDFS de la base.
- ✓ Un drop d'une table interne = suppression de la metadata et les données de la table.

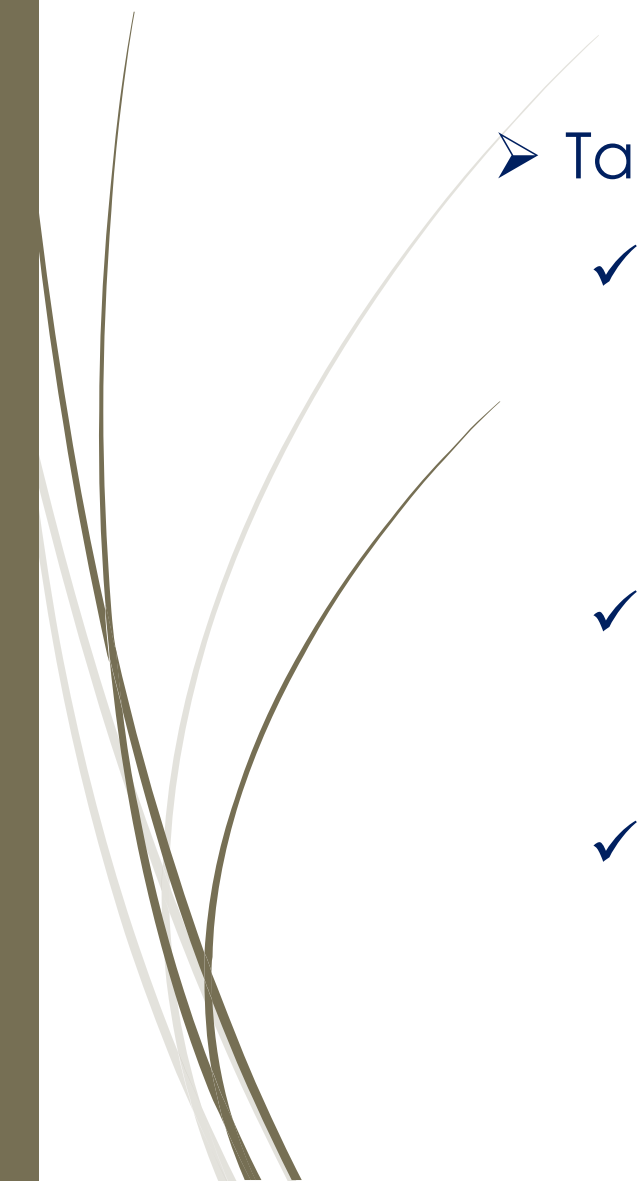


Base de données et Tables Hive

- Tables Externes :
 - ✓ Hive ne gère pas les données de la table.
 - ✓ A la création de la table externes, Hive s'occupe de la création de la metadata.
 - ✓ Un drop d'une table Externe = suppression
 - ✓ Un drop d'une table Externe = suppression de la metada seulement.



Base de données et Tables Hive

- Tables Temporaires :
 - ✓ Hive permet de créer des tables temporaires pour un besoin d'un stockage intermédiaire dans des requêtes complexes.
 - ✓ Hive supprime automatiquement les tables temporaires à la fin de la session.
 - ✓ Une table temporaire est une table interne qui ne supporte pas les options de partitions et indexes.
- 

Base de données et Tables Hive

- Le syntaxe de la création de la table :

```
CREATE [TEMPORARY] [EXTERNAL] TABLE [IF  
NOT EXISTS] [db_name.] table_name  
[(col_name data_type [COMMENT  
col_comment], ...)] [COMMENT  
table_comment] [ROW FORMAT row_format]  
[STORED AS file_format]
```

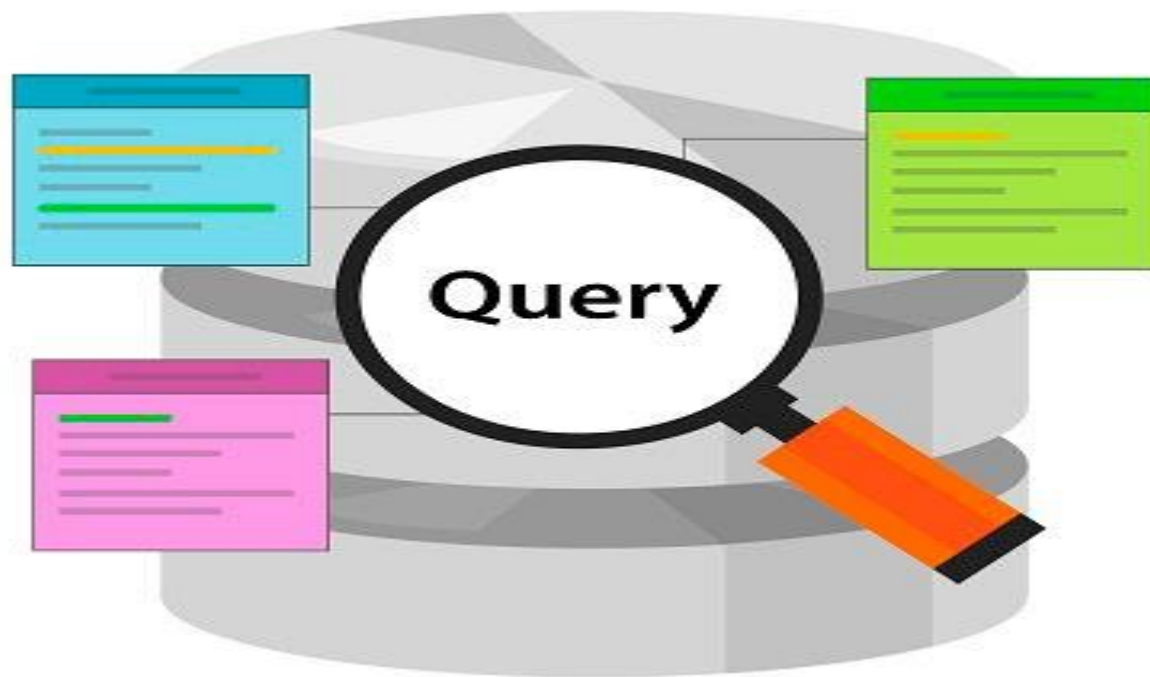

Base de données et Tables Hive

- Le syntaxe de la création de la table :

```
CREATE TABLE IF NOT EXISTS employee ( eid int,  
name String, salary String, destination String)  
COMMENT 'Employee details' ROW FORMAT  
DELIMITED FIELDS TERMINATED BY '\t' LINES  
TERMINATED BY '\n' STORED AS TEXTFILE;
```

Syntaxe HiveQL basique

- Le syntaxe de la création de la table :



Syntaxe HiveQL basique

- Le syntaxe de la création de la table :

Très proche de SQL

- ✓ Create, drop, use database
- ✓ Create, drop, alter table
- ✓ Select * from db_name.table_name
- ✓ JOIN
- ✓ UNION
- ✓ LIMIT

Types de données dans Hive

Numérique :

- ✓ TINYINT (1-byte signed integer, from -128 to 127)
- ✓ SMALLINT (2-byte signed integer, from -32,768 to 32,767)
- ✓ INT/INTEGER (4-byte signed integer, from - 2,147,483,648 to 2,147,483,647)
- ✓ BIGINT (8-byte signed integer, from - 9,223,372,036,854,775,808 to 9,223,372,036,854,775,807)
- ✓ FLOAT (4-byte single precision floating point number



Types de données dans Hive

- DOUBLE (8-byte double precision floating point number)
- DOUBLE PRECISION (alias for DOUBLE, only available starting with Hive 2.2.0)
- DECIMAL
 - ✓ Introduced in Hive 0.11.0 with a precision of 38 digits
 - ✓ Hive 0.13.0 introduced user-definable precision and scale
- NUMERIC (same as DECIMAL, starting with Hive 3.0.0)



Types de données dans Hive

➤ Date :

- ✓ TIMESTAMP
- ✓ DATE
- ✓ INTERVAL

➤ String

- ✓ STRING
- ✓ VARCHAR
- ✓ CHAR

BIG DATA

Lab

▲ 18.75

▼ 25.01

▼ 22.10

▲ 25.21

▼ 30.12

▲ 29.79


▼ 18.07

▲ 07.28

▲ 24.78



Lab 2


- Création de la base lab 2
 - Création d'une table externe
 - Création d'une table interne
 - Requête select
- 

Formats de données Hive





Formats de données Hive

- Les formats utilisés par Hive sont :
 - ✓ Text File
 - ✓ SequenceFile
 - ✓ RCFile
 - ✓ Avro Files
 - ✓ ORC Files
 - ✓ Parquet
- 

Formats de données Hive

➤ **Text File**

- ✓ Format par défaut à la création d'une table sans spécifier le format
- ✓ Create table tb_text (a string, b string);
- ✓ Describe formatted tb_text;

Formats de données Hive

```
hive> create table tb_text(a string, b string);
```

```
OK
```

```
Time taken: 4.005 seconds
```

```
hive>
```

```
hive> describe formatted tb_text;
```

```
OK
```

#	col_name	data_type	comment
	a	string	
	b	string	

```
# Detailed Table Information
```

```
Database:          default
```

```
Owner:             root
```



Formats de données Hive

► **SequenceFile**

- Create table tb_seqFile (a string, b string) STORED AS SEQUENCEFILE;
- DESCRIBE FORMATTED tb_seqFile;

Formats de données Hive

```
hive> describe formatted tb_seqFile;
OK
# col_name          data_type          comment

a                   string
b                   string

# Detailed Table Information
Database:           default
Owner:              root
LastAccessTime:     UNKNOWN
Protect Mode:       None
Retention:          0
Location:            hdfs://sandbox-hdp.hortonworks.com:8020/apps/hive/warehouse/tb_seqfile
Table Type:         MANAGED_TABLE
Table Parameters:
    COLUMN_STATS_ACCURATE {\"BASIC_STATS\": \"true\"}
    numFiles              0
    numRows               0
    rawDataSize           0
    totalSize             0
    transient_lastDdlTime 1658866468

# Storage Information
SerDe Library:      org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat:        org.apache.hadoop.mapred.SequenceFileInputFormat
OutputFormat:       org.apache.hadoop.hive ql.io.HiveSequenceFileOutputFormat
Compressed:         No
Num Buckets:        -1
Bucket Columns:     []
Sort Columns:       []
Storage Desc Params:
    serialization.format 1
Time taken: 0.564 seconds, Fetched: 32 row(s)
```


Formats de données Hive

➤ **RCFile**

- Create table tb_rcFile (a string, b string) STORED AS RCFILE;
- DESCRIBE FORMATTED tb_rcFile;

Formats de données Hive

➤ **AvroFile**

- ✓ Create table tb_avroFile (a string, b string) STORED AS AVROFILE;
- ✓ DESCRIBE FORMATTED tb_avroFile;

Formats de données Hive

➤ **OrcFile**

- ✓ Create table tb_orcFile (a string, b string) STORED AS ORCFILE;
- ✓ DESCRIBE FORMATTED tb_orcFile;

Formats de données Hive

➤ ParquetFile

- ✓ Create table tb_parquetFile (a string, b string)
STORED AS PARQUETFILE;
- ✓ DESCRIBE FORMATTED tb_parquetFile;

Formats de données Hive

➤ ParquetFile

- ✓ Create table tb_parquetFile (a string, b string)
STORED AS PARQUETFILE;
- ✓ DESCRIBE FORMATTED tb_parquetFile;

Formats de données Hive

➤ ParquetFile

- ✓ Create table tb_parquetFile (a string, b string)
STORED AS PARQUETFILE;
- ✓ DESCRIBE FORMATTED tb_parquetFile;

BIG DATA

QUESTION ?



Merci !!

D A T A

