RÉPUBLIQUE DU SÉNÉGAL

UNIVERSITÉ GASTON BERGER DE SAINT-LOUIS

INSTITUT POLYTECHNIQUE DE SAINT-LOUIS

Projet : Ingestion des données dans Big Data

Présenté par :

Bara Sow

Professeur : M. Mboup **Année universitaire** : 2023-2024

INSTITUT POLYTECHNIQUE DE SAINT-LOUIS

Plan

I. Introduction

Le but de ce projet est de mettre en évidence les compétences en ingestion et transformation des données dans un contexte Big Data en utilisant les outils Apache Sqoop et Apache Hive. Le but principal consiste à importer des informations d'une base de données relationnelle externe dans un système Big Data, puis à les traiter et à les transformer en utilisant Hive.

II. Prérequis pour le projet

Avant de commencer le projet, assurez-vous que vous avez installé les logiciels suivants :

- **Apache Sqoop** : Pour l'ingestion des données depuis la base de données vers le Big Data.
- **Apache Hive** : Pour le traitement et la transformation des données dans le Big Data.
- **MariaDB** : Base de données relationnelle utilisée dans ce projet.

Vous aurez également besoin des outils suivants sur votre machine locale :

- **SGBD relationnel** : MySQL ou MariaDB
- **Éditeur de base de données** : MySQL Workbench, Heidi SQL ou PhpMyAdmin

III. Ingestion des données avec Apache Sqoop

Dans ce chapitre, nous allons expliquer en détail comment intégrer des données depuis une base de données MySQL vers le système de fichiers distribués Hadoop (HDFS) en utilisant Apache Sqoop. L'outil de transfert de données Apache Sqoop est spécialement conçu pour importer et exporter des informations entre des bases de données relationnelles et Hadoop.

1. Démarrage du SGBD MySQL

La première étape consiste à démarrer notre MySQL en mode console avec l'utilisateur **root**, faire la création de compte utilisateur **retail_user**, créer une base de données **retail_db,** ajouter les droits d'utilisateur sur la base de données retail_db.

- MySQL en mode console

```
C:\Users\sowba>mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

- Création d'un utilisateur

```
mysql> CREATE user retail_user identified by 'hadoop';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

- Création d'une base de données

```
mysql> CREATE database retail_db;
```

- Ajout des droits d'utilisateur

```
mysql> GRANT ALL PRIVILEGES ON retail_db.* TO 'retail_user'@'%';
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

La prochaine étape consiste à se connecter avec notre utilisateur **retail_user** afin de pouvoir faire le chargement des données notre fichier **retail_db.sql** donné.

- Connexion en tant que retail user

```
C:\Users\sowba>mysql -u retail_user -phadoop
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
```

- Chargement des données

```
mysql> source C:/Users/sowba/OneDrive/Bureau/ProjetBigData/retail_db.sql
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

- Affichage des tables

```
mysql> use retail_db;
Database changed
mysql> show tables;
+---------------------+
| Tables_in_retail_db |
+---------------------+
| categories          |
| customers           |
| departments         |
| order_items         |
| orders              |
| products            |
+---------------------+
6 rows in set (0.02 sec)
```

2. Démarrage de Vagrant et SSH dans la machine virtuelle

L'étape suivante consiste à démarrer notre machine virtuelle Vagrant et à se connecter en SSH. Cela nous permet de configurer l'environnement Hadoop et de préparer l'utilisation de Sqoop.

- Démarrage de Vagrant

```
C:\Users\sowba\hadoopVagrant>vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
```

- Connexion sur notre environnement

```
C:\Users\sowba\hadoopVagrant>vagrant ssh
Last login: Fri Jul 19 11:35:12 2024 from 10.0.2.2
[vagrant@10 ~]$
```

- Démarrage des services Hadoop

```
[vagrant@10 shareFolder]$ start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
^[[Starting secondary namenodes [10.0.2.15]
[vagrant@10 shareFolder]$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
```

3. Configuration de notre base de données avec Sqoop

Après avoir configuré notre environnement, nous employons Apache Sqoop afin d'importer les données de la base de données MySQL vers HDFS. Dans cette partie, nous aurons besoin du **hostname** de la machine locale. Pour le savoir, nous allons au **cmd** et saisir **hostname** et ça nous l'affichera qui est **DESKTOP-SUOJOJB** pour notre machine**.**

- Vérification du réseau de la machine locale et de la machine virtuelle

```
[vagrant@10 shareFolder]$ sqoop list-databases \
> --connect "jdbc:mysql://DESKTOP-SUOJOJB:3306" \
> --username retail_user \
> --password hadoop
Warning: /usr/lib/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/lib/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/lib/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2024-07-25 12:02:41,017 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
2024-07-25 12:02:41,140 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
2024-07-25 12:02:41,280 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
Thu Jul 25 12:02:58 UTC 2024 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+ and 5.7.6+
 requirements SSL connection must be established by default if explicit option isn't set. For compliance with existing applications not using SSL the verifyServerCertif
icate property is set to 'false'. You need either to explicitly disable SSL by setting useSSL=false, or set useSSL=true and provide truststore for server certificate ve
rification.
mysql
information_schema
performance_schema
sys
sante_connect
contact
cinema
etudiant_db
dbsymfony
db_agence_immobiliere
retail_db
lara
laravel
[vagrant@10 shareFolder]$
```

Nous parvenons à voir la base **retail_db** donc nos deux machines sont dans le même réseau.

- Affichage de la liste des tables dans retail_db

```
[vagrant@10 shareFolder]$ sqoop list-tables \
> --connect "jdbc:mysql://DESKTOP-SUOJOJB:3306/retail_db" \
> --username retail_user \
> --password hadoop
Warning: /usr/lib/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/lib/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/lib/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2024-07-25 12:12:10,630 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
2024-07-25 12:12:10,758 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
2024-07-25 12:12:10,999 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
Thu Jul 25 12:12:27 UTC 2024 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+ and 5.7.6+
 requirements SSL connection must be established by default if explicit option isn't set. For compliance with existing applications not using SSL the verifyServerCertif
icate property is set to 'false'. You need either to explicitly disable SSL by setting useSSL=false, or set useSSL=true and provide truststore for server certificate ve
rification.
categories
customers
departments
order_items
orders
products
[vagrant@10 shareFolder]$
```

Toutes les tables sont présentes dans la base de données.

- <u>Importation des tables de la base de données retail_db</u>

Voici un exemple avec la table products. Ainsi de suite, nous allons faire pareil avec les tables categories, orders, order_items, departments, customers.
Le **localhost** constitue notre @IP_hostname .

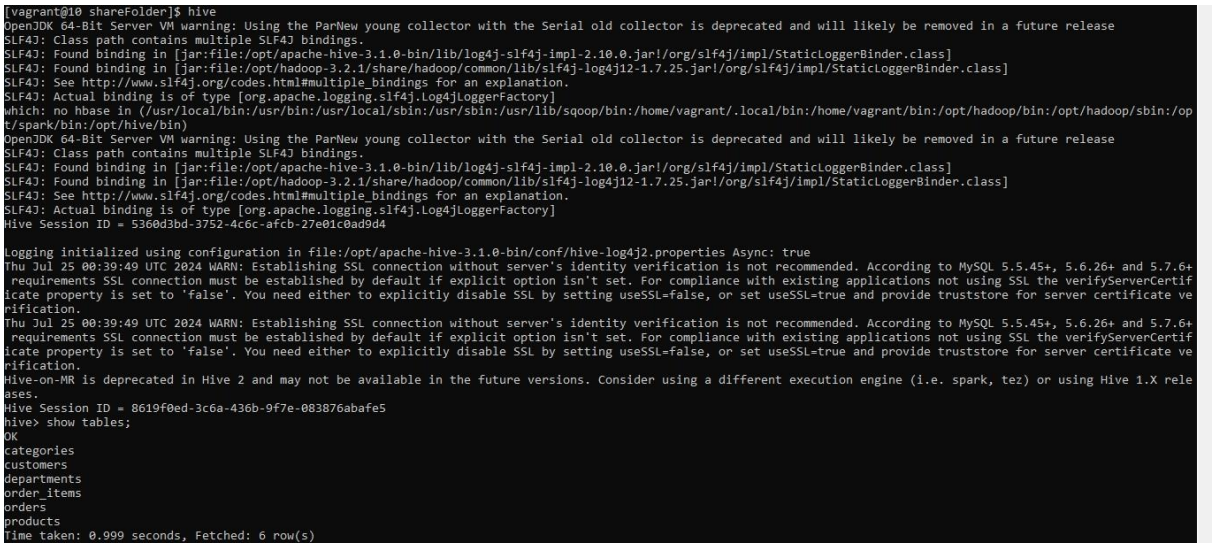
```
[vagrant@10 shareFolder]$ sqoop import \
>   --connect "jdbc:mysql://localhost:3306/retail_db" \
>   --username retail_user \
>   --password hadoop \
>   --table products \
>   --as-parquetfile \
>   --target-dir /user/hive/warehouse/retail_db/products \
>   --delete-target-dir
Warning: /usr/lib/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/lib/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/lib/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2024-07-25 00:20:40,204 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
2024-07-25 00:20:40,495 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
2024-07-25 00:20:40,851 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
2024-07-25 00:20:40,852 INFO tool.CodeGenTool: Beginning code generation
2024-07-25 00:20:40,852 INFO tool.CodeGenTool: Will generate java class as codegen_products
Thu Jul 25 00:20:41 UTC 2024 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+ and 5.7.6+
 requirements SSL connection must be established by default if explicit option isn't set. For compliance with existing applications not using SSL the verifyServerCertif
icate property is set to 'false'. You need either to explicitly disable SSL by setting useSSL=false, or set useSSL=true and provide truststore for server certificate ve
rification.
2024-07-25 00:20:43,125 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `products` AS t LIMIT 1
2024-07-25 00:20:43,224 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `products` AS t LIMIT 1
2024-07-25 00:20:43,274 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /opt/hadoop-3.2.1
```

- <u>Vérification de l'ingestion des données dans warehouse de hive</u>

```
[vagrant@10 shareFolder]$ hdfs dfs -ls /user/hive/warehouse/retail_db/products
Found 6 items
drwxr-xr-x   - vagrant supergroup          0 2024-07-25 00:20 /user/hive/warehouse/retail_db/products/.metadata
drwxr-xr-x   - vagrant supergroup          0 2024-07-25 00:22 /user/hive/warehouse/retail_db/products/.signals
-rw-r--r--   1 vagrant supergroup      17008 2024-07-25 00:22 /user/hive/warehouse/retail_db/products/aa8b7bdb-f8a1-4daa-8455-d7b8c56a6831.parquet
-rw-r--r--   1 vagrant supergroup      16764 2024-07-25 00:22 /user/hive/warehouse/retail_db/products/e8d275ec-4803-42cf-99ef-c2f60d2e3793.parquet
-rw-r--r--   1 vagrant supergroup      13625 2024-07-25 00:22 /user/hive/warehouse/retail_db/products/fe46fb6f-df22-434d-a1ec-ccad57af2558.parquet
-rw-r--r--   1 vagrant supergroup      13925 2024-07-25 00:22 /user/hive/warehouse/retail_db/products/fe76076c-3014-47c9-b96c-e44a2701e07a.parquet
[vagrant@10 shareFolder]$ hdfs dfs -ls /user/hive/warehouse/retail_db/categories
Found 6 items
drwxr-xr-x   - vagrant supergroup          0 2024-07-24 23:59 /user/hive/warehouse/retail_db/categories/.metadata
drwxr-xr-x   - vagrant supergroup          0 2024-07-25 00:00 /user/hive/warehouse/retail_db/categories/.signals
-rw-r--r--   1 vagrant supergroup       1283 2024-07-24 23:59 /user/hive/warehouse/retail_db/categories/0e3784d7-a731-416e-9141-09efa154b86d.parquet
-rw-r--r--   1 vagrant supergroup       1258 2024-07-25 00:00 /user/hive/warehouse/retail_db/categories/4e78889d-79f1-4c3c-9795-f1ab1df77e79.parquet
-rw-r--r--   1 vagrant supergroup       1281 2024-07-24 23:59 /user/hive/warehouse/retail_db/categories/6f7db690-c88c-4faf-a86f-c74a4398308a.parquet
-rw-r--r--   1 vagrant supergroup       1329 2024-07-24 23:59 /user/hive/warehouse/retail_db/categories/995d2dc5-10d7-465b-8a66-84b64f32e35f.parquet
[vagrant@10 shareFolder]$ hdfs dfs -ls /user/hive/warehouse/retail_db/orders
Found 6 items
drwxr-xr-x   - vagrant supergroup          0 2024-07-25 00:23 /user/hive/warehouse/retail_db/orders/.metadata
drwxr-xr-x   - vagrant supergroup          0 2024-07-25 00:24 /user/hive/warehouse/retail_db/orders/.signals
-rw-r--r--   1 vagrant supergroup     147482 2024-07-25 00:24 /user/hive/warehouse/retail_db/orders/067ce50a-06c1-4a67-b940-7996f9a52bf0.parquet
-rw-r--r--   1 vagrant supergroup     147329 2024-07-25 00:24 /user/hive/warehouse/retail_db/orders/67087e87-84ae-4da1-835e-1f8cb320aa79.parquet
-rw-r--r--   1 vagrant supergroup     151760 2024-07-25 00:24 /user/hive/warehouse/retail_db/orders/78bd0e09-7a02-4f84-ab94-c04f86247d7f.parquet
-rw-r--r--   1 vagrant supergroup     147281 2024-07-25 00:24 /user/hive/warehouse/retail_db/orders/a8a4bf3e-05d6-4e9b-b3fa-961b1a366050.parquet
[vagrant@10 shareFolder]$
```

Donc les données ont été bien ingérées.

- <u>Connexion et vérification des tables dans hive</u>



De là nous pouvons apercevoir que les tales ont été bien créées.

VI. Data Processing avec Apache Hive

Dans cette section du projet, notre attention sera portée sur la gestion des données en utilisant Apache Hive. Une fois que les données de la base de données ont été importées dans HDFS via Sqoop, il est essentiel de les organiser et de les préparer afin de les analyser de manière efficace. L'outil Apache Hive, qui permet d'exécuter des requêtes SQL sur des données massives stockées dans Hadoop, est très utile pour cette tâche.

Notre Vagrant démarré ainsi que notre environnement de développement en exécution connecté, nous pouvons commencer à exécuter les requêtes SQL vue que toutes les tables sont disponibles dans Hive.

Exécution des requêtes SQL :

1) Trouver le nombre total de commandes passées par chaque client au cours de l'année 2014.

```
SELECT
  c.customer_fname,
  c.customer_lname,order_status,
  COUNT(o.order_id) AS total_orders
FROM customers AS c
JOIN orders AS o
  ON c.customer_id = o.order_customer_id
WHERE
  o.order_status = 'COMPLETE' AND YEAR(o.order_date) = 2014
GROUP BY
  c.customer_fname,
  c.customer_lname
ORDER BY
  c.customer_fname;
```

2) Afficher le nom et le prénom des clients qui n'ont passé aucune commande, triés par customer_lname puis customer_fname.

```
SELECT
c.customer_fname,
c.customer_lname
FROM
customers c
LEFT JOIN
orders o ON c.customer_id = o.order_customer_id
WHERE
o.order_id IS NULL
ORDER BY
 c.customer_lname,
 c.customer_fname;
```

3) Afficher les détails des top 5 clients par revenue pour chaque mois. Vous devez obtenir tous les détails du client ainsi que le mois et les revenus par mois. Les données doivent être triées par mois dans l'ordre croissant et les revenus par mois dans l'ordre décroissant

```
SELECT
   YEAR(o.order_date) AS annee,
   MONTH(o.order_date) AS mois,
   c.customer_id,
   c.customer_fname,
c.customer_lname,
   SUM(oi.order_item_subtotal) AS revenus_mensuels
FROM
   orders o
JOIN
   order_items oi ON o.order_id = oi.order_item_order_id
JOIN
   customers c ON o.order_customer_id = c.customer_id
GROUP BY
   YEAR(o.order_date),
   MONTH(o.order_date),
   c.customer_id,
c.customer_lname,
   c.customer_fname
ORDER BY
   annee,
   mois,
   revenus_mensuels DESC
LIMIT 5;
```

4) Trouver toutes les commandes terminées ou fermées (completed ou closed), puis calculez le revenu total pour chaque jour pour chaque département. La sortie doit afficher : order_date, department_name et order_revenue

```
SELECT
   o.order_date,
   d.department_name, order_status,
   SUM(oi.order_item_subtotal) AS order_revenue
FROM
   orders o
JOIN
   order_items oi ON o.order_id = oi.order_item_order_id
JOIN
   products p ON oi.order_item_product_id = p.product_id
JOIN
   categories c ON p.product_category_id = c.category_id
JOIN
   departments d ON c.category_department_id = d.department_id
WHERE
   o.order_status IN ('COMPLETE', 'CLOSED')
GROUP BY
   o.order_date, d.department_name
ORDER BY
   o.order_date, d.department_name;
```

5) Trouver le rank de chaque catégorie par revenue obtenue dans chaque département à partir de toutes les transactions. Affichez les résultats par department_name et classez-les par ordre croissant.

```
SELECT
   d.department_name,
   c.category_name,
   SUM(oi.order_item_subtotal) AS category_revenue,
   RANK() OVER (PARTITION BY d.department_name ORDER BY
SUM(oi.order_item_subtotal) DESC) AS category_rank
FROM
   order_items oi
JOIN
   products p ON oi.order_item_product_id = p.product_id
JOIN
   categories c ON p.product_category_id = c.category_id
JOIN
   departments d ON c.category_department_id = d.department_id
GROUP BY
   d.department_name, c.category_name
ORDER BY
   d.department_name, category_rank;
```

6) Afficher le pourcentage de chaque catégorie par revenue dans chaque département. Afficher les résultats par department_name et pourcentage par ordre décroissant.

```
SELECT
  d.department_name,
  c.category_name,
  SUM(oi.order_item_subtotal) AS category_revenue,
  SUM(oi.order_item_subtotal) / SUM(SUM(oi.order_item_subtotal)) OVER
(PARTITION BY d.department_name) * 100 AS percentage
FROM
  order_items oi
JOIN
  products p ON oi.order_item_product_id = p.product_id
JOIN
  categories c ON p.product_category_id = c.category_id
JOIN
  departments d ON c.category_department_id = d.department_id
GROUP BY
  d.department_name, c.category_name
ORDER BY
  d.department_name, percentage DESC;
```

7) Afficher tous les clients qui ont passé une commande d'un montant supérieur à 200 $.

```
SELECT DISTINCT
  c.customer_id,
  c.customer_fname,
  c.customer_lname,
  c.customer_email
FROM
  customers c
JOIN
  orders o ON c.customer_id = o.order_customer_id
JOIN
  order_items oi ON o.order_id = oi.order_item_order_id
GROUP BY
  c.customer_id, c.customer_fname, c.customer_lname, c.customer_email
HAVING
  SUM(oi.order_item_subtotal) > 200;
```

8) Afficher les clients de la "customers" dont les noms customer_fname commence par "Rich"

```
SELECT
   customer_id,
   customer_fname,
   customer_lname,
   customer_email
FROM
   customers
WHERE
   customer_fname LIKE 'Rich%';
```

9) Fournir le nombre total de clients dans chaque état (state) dont le prénom commence par « M »

```
SELECT
   customer_state,
customer_fname,
customer_lname,
   COUNT(*) AS total_customers
FROM
   customers
WHERE
   customer_fname LIKE 'M%'
GROUP BY
   customer_state;
```

10) Trouver le produit le plus cher dans chaque catégorie

```
SELECT c.category_name, p.product_name, p.product_price
FROM products p
JOIN categories c ON p.product_category_id = c.category_id
WHERE (p.product_category_id, p.product_price) IN (
   SELECT product_category_id, MAX(product_price)
   FROM products
   GROUP BY product_category_id
);
```

11) Trouvez les 10 meilleurs produits qui ont généré les revenus les plus élevés.

```
SELECT p.product_name, SUM(oi.order_item_subtotal) AS total_revenue
FROM products p
JOIN order_items oi ON p.product_id = oi.order_item_product_id
GROUP BY p.product_name
ORDER BY total_revenue DESC
LIMIT 10;
```

VII. Conclusion

Ce projet met en évidence les bénéfices de la combinaison d'outils Big Data tels que Sqoop et Hive pour l'ingestion, le stockage et le traitement de grandes quantités de données. La maîtrise des compétences acquises est cruciale pour toute personne voulant travailler dans le domaine du Big Data, car elles fournissent une fondation solide pour analyser des données avancées et gérer des pipelines de données à grande échelle.