Inference Document for Implementation Using Hive and Sqoop for the Online Shopping System

Overview

This document explains the implementation of a Big Data solution for an **Online Shopping System** using **Hive and Sqoop frameworks**. The aim is to:

- Import product data from a relational database (RDBMS) into the Hadoop ecosystem using Sqoop.
- Use **Hive** to query and analyze the product data by categories (purpose of the product) and to count the total number of products available.

---

2. Problem Statement

The Online Shopping System has an inventory of products listed in an RDBMS. The **goal** is to:

1. Identify the total number of products available on the portal.
2. Segregate these products according to their category/purpose (e.g., Electronics, Clothing, Books).

---

3. Frameworks and Tools Used

- **Sqoop**: Sqoop is a tool designed to efficiently transfer bulk data between Hadoop and structured data stores such as relational databases.
- **Hive**: Hive is a data warehouse system built on top of Hadoop. It provides a SQL-like interface to query data stored in Hadoop's HDFS.

---

4. Steps Involved in the Implementation

Step 1: Use Sqoop to Import Data from RDBMS to HDFS

Assume the product data resides in a relational database (MySQL, PostgreSQL, etc.). The data will be transferred into Hadoop's HDFS for further analysis and querying.

1. Table Structure in RDBMS: The products table in the relational database has the following fields:
    - product_id: Unique identifier for each product.
    - product_name: Name of the product.
    - category: Category of the product (e.g., Electronics, Clothing, Books).
    - price: Price of the product.
    - description: Detailed description of the product.

2. **Using Sqoop to Import Data: To import the product data from the products table in MySQL into HDFS, we use the following Sqoop command:**

sqoop import --connect jdbc:mysql://localhost:3306/online_shopping --username root --password password \

--table products --target-dir /user/hadoop/products_data --fields-terminated-by ',' --m 1

- Explanation of Parameters:
  - --connect jdbc:mysql://localhost:3306/online_shopping: Specifies the JDBC connection string to the MySQL database.
  - --username root --password cloudera: MySQL login credentials.
  - --table products: Specifies the products table to import.
  - --target-dir /user/hadoop/products_data: The location in HDFS where data will be stored.
  - --fields-terminated-by ',': Specifies the delimiter used in the table (comma).
  - --m 1: Number of mappers to use during import (set to 1 for small data).

This imports the data into HDFS, under the /user/hadoop/products_data directory.

Step 2: Create Hive Table

Once the data is in HDFS, we can create a Hive table that will allow us to query this data using Hive's SQL-like syntax.

1. **Hive Table Creation Query:**

```
CREATE EXTERNAL TABLE products (
   product_id INT,
   product_name STRING,
   category STRING,
   price DECIMAL,
   description STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LOCATION '/user/hadoop/products_data';
```

- Explanation of the Table Creation:

- CREATE EXTERNAL TABLE: Indicates that the data already exists in HDFS and should be referenced.
- FIELDS TERMINATED BY ',': Specifies the delimiter for the data.
- LOCATION '/user/hadoop/products_data': Points to the directory in HDFS where the imported data is stored.

Step 3: Count the Total Number of Products

To find the total number of products available in the shopping portal, we can use a simple Hive query to count the rows in the products table.

1. Query to Count Products:

```
SELECT COUNT(*) FROM products;
```

   - Explanation: This query counts the total number of products in the products table. The output will give the total number of products available in the portal.

Step 4: Segregate Products Based on Category (Purpose)

The next task is to segregate the products based on their category or purpose. This can be done using the GROUP BY clause in Hive to group products by their category.

1. Query to Group Products by Category:

```
SELECT category, COUNT(*) AS product_count
FROM products
GROUP BY category;
```

   - Explanation:
     - GROUP BY category: Groups the products by their category field.
     - COUNT(*): Counts the number of products in each category.

The query will return the number of products in each category, such as:

category	product_count
Electronics	500
Clothing	300
Books	200

5. Conclusion

In this document, we demonstrated how to use Sqoop to import data from a relational database into HDFS and then use Hive to query the data for product count and segregation based on product categories (purpose). The solution provides the following insights:

- The total number of products available on the portal.
- The segregation of products into categories such as Electronics, Clothing, and Books.

This approach allows businesses to leverage the Hadoop ecosystem for scalable, efficient data processing, enabling insightful analysis of large datasets from their online shopping system.