Ex No 5

Create tables in Hive and write queries to access the data in the table

# AIM:

To create tables in Hive and write queries to access the data in the table.

# PROCEDURE:

**Step 1: Download and Install Hive**

1. **Download Hive**:

**Download Hive from the official website:**

wget https://downloads.apache.org/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz

1. **Extract Hive**:

tar -xvf apache-hive-3.1.2-bin.tar.gz

1. **Move Hive Directory**:

sudo mv apache-hive-3.1.2-bin /usr/local/hive

1. **Set Hive Environment Variables**: **Edit .bashrc to configure Hive:**

nano ~/.bashrc

**Add the following lines:**

export HIVE\_HOME=/usr/local/hive export PATH=$PATH:$HIVE\_HOME/bin **Apply the changes:**

source ~/.bashrc

1. **Configure Hive**:

**Configure Hive to use MySQL as its metastore by editing the Hive configuration file (hive- site.xml):**

nano $HIVE\_HOME/conf/hive-site.xml

**Add the following configuration for MySQL connection:**

<property>

<name>javax.jdo.option.ConnectionURL</name>

<value>jdbc:mysql://localhost/metastore</value>

</property>

<property>

<name>javax.jdo.option.ConnectionDriverName</name>

<value>com.mysql.jdbc.Driver</value>

</property>

<property>

<name>javax.jdo.option.ConnectionUserName</name>

<value>root</value>

</property>

<property>

<name>javax.jdo.option.ConnectionPassword</name>

<value>password</value>

</property>

1. **Start Hive**:

**Once everything is configured, start Hive by simply typing:**

hive

**Step 2: Create a Database and Table in Hive**

1. **Create a Database**:

**In the Hive terminal, create a new database:**

CREATE DATABASE financials;

1. **Use the Database**: USE financials;
2. **Create a Table**:

**Create a table to store financial data:**

CREATE TABLE finance\_table ( id INT,

name STRING

)

1. **Insert Data into the Table**:

**Insert sample data into the finance\_table:**

INSERT INTO TABLE finance\_table VALUES (1, 'Alice'), (2, 'Bob'), (3, 'Charlie');

**Step 3: Store the Output in HDFS**

1. **Create a Partitioned Table**:

**For optimized storage, create a partitioned table by year:**

CREATE TABLE partitioned\_finance\_table ( id INT,

name STRING

)

PARTITIONED BY (year INT)

1. **Insert Data into the Partitioned Table**:

INSERT INTO partitioned\_finance\_table PARTITION (year=2023) VALUES (1, 'Alice'), (2, 'Bob');

INSERT INTO partitioned\_finance\_table PARTITION (year=2024) VALUES (3, 'Charlie');

1. **Create a Bucketed Table**:

**Create a bucketed table to improve query performance:**

CREATE TABLE bucketed\_finance\_table ( id INT,

name STRING

)

CLUSTERED BY (id) INTO 4 BUCKETS

1. **Insert Data into the Bucketed Table**:

INSERT INTO TABLE bucketed\_finance\_table VALUES (1, 'Alice'), (2, 'Bob'), (3, 'Charlie');

**Step 4: View the Output in HDFS**

1. **Create an ORC Table**:

**Use ORC (Optimized Row Columnar) format for efficient storage:**

CREATE TABLE orc\_finance\_table ( id INT,

name STRING

)

1. **Insert Data into the ORC Table**:

INSERT INTO TABLE orc\_finance\_table SELECT \* FROM finance\_table;

1. **View the Output in HDFS**:

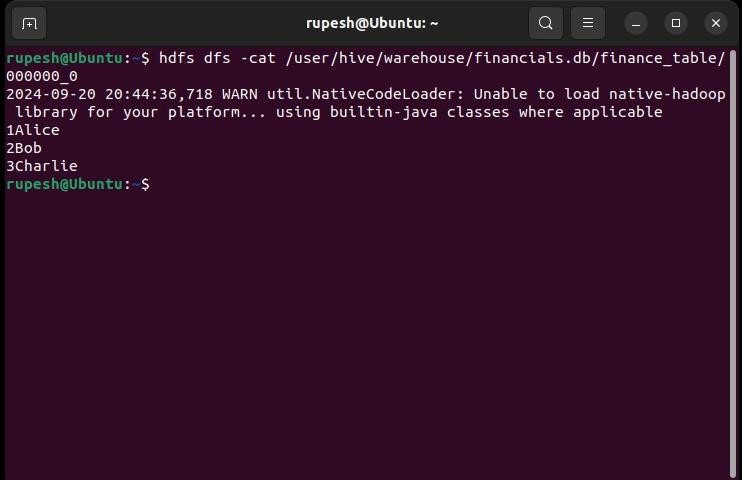
**You can view the output by navigating to the HDFS directory where Hive stores the data. Use the following command to view the stored data:**

hdfs dfs -ls /user/hive/warehouse/financials.db/finance\_table

**To view the contents of the ORC table:**

hdfs dfs -cat /user/hive/warehouse/financials.db/orc\_finance\_table/000000\_0

**OUTPUT:**



**RESULT:**

Thus, to create tables in Hive and write queries to access the data in the table was completed successfully.