**Week-7**

**1. Using the Hive Command-Line Interface (CLI)**

1. **Open the Terminal** (SSH into the Cloudera node if you're working remotely).
2. **Start Hive CLI** by simply typing:

Hive

This will launch the Hive command prompt, which looks like hive>.

**Execute Hive Commands**:

* Once you’re in the Hive prompt, you can type any Hive SQL commands. For example

CREATE DATABASE sample\_db;

USE sample\_db;

CREATE TABLE sample\_table (id INT, name STRING);

SHOW TABLES;

SELECT \* FROM sample\_table;

To exit the Hive CLI, type

exit;

Running Hive Commands Directly from the Terminal (Without Entering Hive CLI)

You can also run Hive commands directly from the shell prompt by using the -e or -f option with the hive command.

* **Using the -e Option** (Execute Command):

hive -e "SHOW DATABASES;"

**1. Creating Tables with Different Storage Formats**

* **Creating a Table Stored as ORC File**

CREATE TABLE orc\_table (

id INT,

name STRING,

salary FLOAT

)

STORED AS ORC;

**Creating a Table Stored as Parquet File**:

CREATE TABLE parquet\_table (

id INT,

name STRING,

age INT

)

STORED AS PARQUET;

**2. Loading Data into Tables**

* **Load Data into Table from HDFS Path**

LOAD DATA INPATH '/user/hive/warehouse/datafile.csv' INTO TABLE sample\_table;

**Load Local Data into Hive Table**:

**Inserting Data into Hive Tables**

* **Insert Data Using Values**

INSERT INTO TABLE sample\_table VALUES (1, 'Alice', 3000), (2, 'Bob', 4000);

**Basic Query Examples**

* **Filtering Data**

SELECT \* FROM sample\_table WHERE age > 30;

Using Aggregate Functions

SELECT department, COUNT(\*) AS num\_employees FROM employees GROUP BY department;

Using HAVING Clause

SELECT department, AVG(salary) AS avg\_salary

FROM employees

GROUP BY department

HAVING avg\_salary > 5000;

**Join Operations**

* **Inner Join**

SELECT a.id, a.name, b.salary

FROM employees a

JOIN salaries b ON (a.id = b.emp\_id);

**Partitioning and Bucketing**

* **Create Partitioned Table**

CREATE TABLE partitioned\_table (

id INT,

name STRING,

salary FLOAT

) PARTITIONED BY (department STRING);

**Working with Views**

* **Create a View**

CREATE VIEW high\_salary\_view AS

SELECT id, name, salary

FROM employees

WHERE salary > 5000;

**Altering Table Structure**

* **Rename Column**

ALTER TABLE sample\_table CHANGE COLUMN old\_column\_name new\_column\_name STRING;