



Hive Installation

- To install Hive on your local machine follow the link: <https://cwiki.apache.org/confluence/display/Hive/GettingStarted#GettingStarted-InstallationandConfiguration>
- To use Hive from Cloudera, there are 2 main methods:
 - Using Terminal, open a terminal and type hive
 - Using Hue:
 - 1- Open the browser and click on Hue link.
 - 2- Click on the "Query Editors" and select Hive.

2

Simple Hive Query Example

- We will execute simple query on the sample tables in the default database that comes with Cloudera and as follows:
 - Click on Hue from the browser (use the word "cloudera" as a username and password to sign in Hue)and click on next to go to step 2.
 - Click on Hive Editor to download Hive examples.
 - Select Hive from "Query Editor".
 - Click on refresh and you will see two sample tables.

3

Simple Hive Query Example



Execute the following SQL statement

```
SELECT sample_07.description,  
       sample_07.salary  
FROM  
  sample_07  
WHERE  
  (sample_07.salary>100000)  
ORDER BY sample_07.salary DESC  
LIMIT 100;
```

4

Creating Database in Hive



- We will create new database and tables using either terminal or Hue as follows:
 - Using the terminal, type the following:
hive
hive> CREATE DATABASE empdb;
hive> DROPE DATABASE empdb;
 - Using Hue, do the following:
 - Now open the browser and click on Hue then select Hive from “Query Editors” and type CREATE DATABASE empdb; click on Execute then click on refresh then select empdb from the database list.

5

Creating Table in Hive



- Write the following SQL statement to create a table:

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

- Click on Execute

6

Creating Text File to Hold Data



- To load the table with data, create a text file by clicking on Places from Cloudera and select "Home Folder" then R.C. and select "create Document" then "Empty File" and write the following values separated by tabs:

| | | |
|---|----|------|
| 1 | a1 | 1000 |
| 2 | a2 | 1000 |
| 3 | b1 | 1000 |
| 4 | b2 | 2000 |
| 5 | z1 | 2000 |
| 6 | x2 | 2000 |

7

Uploading Text File to Hive



- Now we need to upload this file to hadoop and then use it with Hive, so click on the employee table then click on the eye icon.
- Click on "import data", click on the two dots button then click on "Upload a file" and select the text file.
- The file will be uploaded to Hadoop and will be shown in the list, so select the file from the list and click on submit

8

Hive Partition Table Example



- To use partition, first we need to write and execute:

```
set hive.exec.dynamic.partition.mode=nonstrict;  
set hive.exec.dynamic.partition=true;
```
- Next, we create a new table to hold partitions:

```
CREATE TABLE emparts (eid int, name String)  
PARTITIONED BY (salary String);
```
- Fill table with data by:

```
INSERT OVERWRITE TABLE emparts  
PARTITION (salary) SELECT * FROM  
employee;
```

9

Hive Java Example



- Open eclipse and create new Java project.
- R.C. on project and select "Build Path" then select "Add External Archives...". Now click on "File System" then select the path: "/usr/lib/hadoop" and select all jar files then click OK.
- Do the same as the above step, but select jar files in the path: "/usr/lib/hadoop/lib".
- Do the same as the above step, but select jar files in the path: "/usr/lib/hive/lib".
- Create a new Java class file and write the program in the next slide.

10

Hive Java Example



```
import java.sql.SQLException;
import java.sql.Connection;
import java.sql.Statement;
import java.sql.DriverManager;

public class Main {
    private static String driverName = "org.apache.hive.jdbc.HiveDriver";

    public static void main(String[] args) throws SQLException {
        try {
            Class.forName(driverName);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
            System.exit(1);
        }
        Connection con = DriverManager.getConnection(
            "jdbc:hive2://localhost:10000/userdb2", "", "");
        Statement stmt = con.createStatement();
        stmt.executeQuery("CREATE TABLE emp1 (eid int, name String, salary String) "
            + "COMMENT 'Employee details'"
            + "ROW FORMAT DELIMITED"
            + "FIELDS TERMINATED BY '\t'"
            + "LINES TERMINATED BY '\n'"
            + "STORED AS TEXTFILE");
        System.out.println("Table emp1 Created Successfully");
        con.close();
    }
}
```

11

Thank You.....



Any Question?
