● Hive Data Definitions:

DDL commands are the statements that are responsible for defining and changing the structure of a database or table in Hive.

|  |  |
| --- | --- |
| CREATE | Database,Table |
| DROP | Database,Table |
| TRUNCATE | Table |
| ALTER | Database,Table |
| SHOW | Databases,Tables,Table Properties,Partitions,Functions,Index |
| DESCRIBE | Database, Table ,View |
| DDL Commands in Hive  Example | |

**Create Database in Hive**

As the name implies, this DDL command in Hive is used for creating databases.

CREATE (DATABASE) [IF NOT EXISTS] database\_name

  [COMMENT database\_comment]

  [LOCATION hdfs\_path]

  [WITH DBPROPERTIES (property\_name=property\_value, ...)];

In the above syntax for create database command, the values mentioned in square brackets [] are optional.

**Usage of Create Database Command in Hive**

hive> create database if not exists firstDB comment "This is my first demo" location '/user/hive/warehouse/newdb' with DBPROPERTIES ('createdby'='abhay','createdfor'='dezyre');

OK

Time taken: 0.092 seconds

**Drop Database in Hive**

This command is used for deleting an already created database in Hive and the syntax is as follows -

DROP (DATABASE) [IF EXISTS] database\_name [RESTRICT|CASCADE];

**Usage of Drop Database Command in Hive**

hive> drop database if exists firstDB CASCADE;

OK

Time taken: 0.099 seconds

In Hadoop Hive, the mode is set as RESTRICT by default and users cannot delete it unless it is non-empty. For deleting a database in Hive along with the existing tables, users must change the mode from RESTRICT to CASCADE.

In the syntax for drop database Hive command, “if exists” clause is used to avoid any errors that might occur if the programmer tries to delete a database which does not exist.

● Hive Data Manipulations/ HiveQL Manipulations

DML (Data Manipulation Language) commands in Hive are used for inserting and querying the data from hive tables once the structure and architecture of the database has been defined using the DDL commands listed above.

Data can be loaded into Hive tables using –

* LOAD command
* Insert command

Example

**Usage of LOAD Command for Inserting Data Into Hive Tables**

#### Syntax for Load Command in Hive

LOAD DATA [LOCAL] INPATH 'hdfsfilepath/localfilepath' [OVERWRITE] INTO TABLE existing\_table\_name

Let’s load a structured file that contains information about different students.

Let’s take a look at the data present in the file –

ID|name|age|fee|city|state |zip

1|Kendall|22|25874|Kulti-Barakar|WB|451333

2|Mikayla|25|35367|Jalgaon|Maharastra|710179

3|Raven|20|49103|Rewa|Madhya Pradesh|392423

4|Carla|19|27121|Pilibhit|UP|769853

5|Edward|21|32053|Tuticorin|Tamil Nadu|368262

6|Wynter|21|43956|Surendranagar|GJ|457441

7|Patrick|19|19050|Mumbai|MH|580220

8|Hayfa|18|15590|Amroha|UP|470705

9|Raven|16|37836|Cuddalore|TN|787001

* The file is a ‘|’ delimited file where each row  can be inserted as a table record.
* First let’s create a table student based on the contents in the file –
* The **ROW FORMAT DELIMITED** must appear before any of the other clauses, with the exception of the STORED AS … clause.
* The clause **ROW FORMAT DELIMITED FIELDS TERMINATED BY '|** means  I character will be used as field separator by hive.
* The clause **LINES TERMINATED BY ‘\n'**means that the line delimiter will be new line.
* The clause **LINES TERMINATED BY ‘\n'** and **STORED AS …** do not require the ROW FORMAT DELIMITED keywords.

hive> CREATE TABLE IF NOT EXISTS college.students (

> ID BIGINT COMMENT 'unique id for each student',

> name STRING COMMENT 'student name',

> age INT COMMENT 'sudent age between 16-26',

> fee DOUBLE COMMENT 'student college fee',

> city STRING COMMENT 'cities to which students belongs',

> state STRING COMMENT 'student home address state s',

> zip BIGINT COMMENT 'student address zip code'

> )

> COMMENT 'This table holds the demography info for each student'

> ROW FORMAT DELIMITED

> FIELDS TERMINATED BY '|'

> LINES TERMINATED BY '\n'

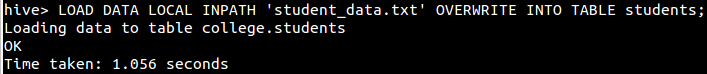
> STORED AS TEXTFILE

> LOCATION '/user/hive/warehouse/college.db/students';

OK

Time taken: 0.112 seconds

* Let’s  load the file into the student table –



If the keyword LOCAL is not specified, then Hive will need absolute URI of the file. However, if local is specified then it assumes the following rules -

* It will assume it’s an HDFS path and will try to search for the file in HDFS.
* If the path is not absolute, then hive will try to locate the file in the /user/ in HDFS.

Using the OVERWRITE keyword while importing means the data will be ingested i.e. it will delete old data and put new data otherwise it would just append the new data. The contents of the target table will be deleted and replaced by the files referred to by file path; otherwise the files referred by file path will be added to the table.

Let’s check if the data has been inserted into the table –

hive> select \* from students;

:

:

:

596 Stephen 25 16573.0 Gaya BR 874761

597 Colby 25 19929.0 New Bombay Maharastra 868698

598 Drake 21 49260.0 Nagaon Assam 157775

599 Tanek 18 12535.0 Gurgaon Haryana 201260

600 Hedda 23 43896.0 Ajmer RJ 697025

Time taken: 0.132 seconds, Fetched: 601 row(s)

2. Check first 5 records:

Now, let's try to retrieve only 5 records using the limit option -

hive> select \* from students limit 5;

OK

NULL name NULL NULL city state NULL

1 Kendall 22 25874.0 Kulti-Barakar WB 451333

2 Mikayla 25 35367.0 Jalgaon Maharastra 710179

3 Raven 20 49103.0 Rewa Madhya Pradesh 392423

4 Carla 19 27121.0 Pilibhit UP 769853

Time taken: 0.144 seconds, Fetched: