|  |  |
| --- | --- |
| Week-8 | Hive Data Definition Language(DDL) Commands |
| 8a | ImplementData Definition Language (DDL) Commandsfor databases in Hadoop Hiveframework using Cloudera. |

* Open Virtual box and then start **clouderaquickstartTerminal**and type “hive” to launch hive shell

**DDL Commands for Databases**

**1)** **CREATE** database Statement is used to create a database in Hive. A database in Hive is a namespace or a collection or catalog of tables.

Syntax: **CREATE DATABASE|SCHEMA [IF NOT EXISTS] database\_name**

**[COMMENT database\_comment]**

**[LOCATION hdfs\_path]**

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

[ ] are optional clauses. We can use SCHEMA in place of DATABASE in this command. The following query is executed to create a database named employee. If everything went good, you will see a ‘OK’ message, else you will see relevant error message.

Simple creation

hive> CREATE DATABASE facultycse;

OK

Time taken: 0.033 seconds

hive> CREATE DATABASE facultyece;

Full creation

hive> CREATE DATABASE IF NOT EXISTS employee COMMENT ‘this is employee database’ LOCATION ‘/user/hive/warehouse/hivedir/’ WITH DBPROPERTIES (‘creator’=‘Bhanu’, ‘date’=‘2020-12-07’);

**2)** **SHOW** databases statement lists all the databases present in the metastore.

Syn: **SHOW (DATABASES/SCHEMAS) [LIKE ‘wildcards'];**

* Wildcards in the regular expression can only be '\*' for any character(s) or '|' for a choice. Examples are 'employees', 'emp\*', 'emp\*|\*ees', all of which will match the database named 'employees’:

|  |  |
| --- | --- |
| hive> SHOW DATABASES;  default  employee  facultycse  facultyece | hive> SHOW DATABASES LIKE ‘\*ee’;  employee |
| hive> SHOW DATABASES LIKE ‘fac\*’;  facultycse  facultyece |

**3) DESCRIBE** database statement in Hive shows the name of Database in Hive, its comment (if set), its location, its owner name, owner type and its properties.

Syn: **DESCRIBE DATABASE/SCHEMA [EXTENDED] db\_name;**

* EXTENDED can be used to get the database properties.

hive>DESCRIBE DATABASE facultycse;

facultycse hdfs://quickstart.cloudera:8020/user/hive/warehouse/faculty.db cloudera USER

hive>DESCRIBE DATABASE EXTENDED employee;

employee this is employee database hdfs://quickstart.cloudera:8020/user/hive/warehouse/ cloudera USER {date=2020-12-07, creator=Bhanu};

**4)USE** database statement in Hive is used to select the specific database for a session on which all subsequent HiveQL statements would be executed.

Syn: **USE db\_name;**

hive> USE employee;

OK

**5)DROP** database statement in Hive is used to Drop (delete) the database. The default behavior is RESTRICT which means that the database is dropped only when it is empty. To drop the database with tables, we can use CASCADE.

Syn: **DROP (DATABASE|SCHEMA) [IF EXISTS] db\_name [RESTRICT|CASCADE];**

hive>DROP DATABASE facultyece;

OK

hive> DROP DATABASE IF EXISTS facultycse CASCADE;

OK

**6) ALTER** database statement in Hive is used to change the metadata associated with the database in Hive.

Syntax for changing Database Properties:

**ALTER (DATABASE|SCHEMA) db\_name SET DBPROPERTIES (property\_name=property\_value, ...);**

hive> ALTER DATABASE employee SET DBPROPERTIES (‘creator’=‘Bhanu Prasad’, ‘date’=‘07-12-2020’);

employee this is employee database hdfs://quickstart.cloudera:8020 /user/hive/warehouse/hivedir/ cloudera USER {date= **07-12-2020**, creator=**Bhanu Prasad**};

Syn for changing Database owner:

**ALTER (DATABASE|SCHEMA) database\_name SET OWNER [USER|ROLE] user\_or\_role;**

hive> ALTER DATABASE employee SET OWNER **USERclient**;

employee this is employee database hdfs://quickstart.cloudera:8020 /user/hive/warehouse/hivedir/ **client USER**{date= 07-12-2020, creator=Bhanu Prasad};

hive> ALTER DATABASE employee SET OWNER **ROLE Admin**;

employee this is employee database hdfs://quickstart.cloudera:8020 /user/hive/warehouse/hivedir/ **Admin ROLE** {date= 07-12-2020, creator=Bhanu Prasad};

|  |  |
| --- | --- |
| 8b | ImplementData Definition Language (DDL) Commands for tables in Hadoop Hive framework using Cloudera. |

**DDL Commands for Tables**

**1) CREATE TABLE** statement in Hive is used to create a table with the given name. If a table or view already exists with the same name, then the error is thrown. We can use IF NOT EXISTS to skip the error.

Syn: **CREATE TABLE [IF NOT EXISTS] [db\_name.] table\_name [(col\_namedata\_type [COMMENT col\_comment], ... [COMMENT col\_comment])]**

**[COMMENT table\_comment]**

**[ROW FORMAT row\_format]**

**[STORED AS file\_format]**

**[LOCATION hdfs\_path];**

hive> CREATE TABLE IF NOT EXISTS employee.emptable (emp\_id STRING COMMENT ‘This is Employee ID’, emp\_name STRING COMMENT ‘This is Employee Name’, emp\_sal FLOAT COMMENT ‘This is Employee Salary’)

COMMENT ‘This table contains Employees Data’

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ‘,’

STORED AS TEXTFILE;

**2)** **SHOW** tables statement in Hive lists all the base tables and views in the current database.

Syn: **SHOW TABLES [IN database\_name];**

hive> SHOW TABLES IN employee;

OK

emptable

**3)** **DESCRIBE** table statement in Hive shows the lists of columns for the specified table.

Syn: **DESCRIBE [EXTENDED|FORMATTED] [db\_name.] table\_name[.col\_name ( [.field\_name])];**

hive> DESCRIBE employee.emptable;

emp\_id string This is Employee ID

emp\_name string This is Employee Name

emp\_sal float This is Employee Salary

hive> DESCRIBE EXTENDED employee.emptable;

hive> DESCRIBE FORMATTED employee.emptable;

**4) ALTER** table statement in Hive enables you to change the structure of an existing table, rename the table, add columns to the table, change the table properties, etc.

Syntax for Rename a table:

**ALTER TABLE table\_name RENAME TO new\_table\_name;**

hive> ALTER TABLE employee.emptable RENAME TO employee.facultytable;

Syn to Add columns to a table:

**ALTER TABLE table\_name ADD COLUMNS (column1, column2) ;**

hive> ALTER TABLE employee.facultytable ADD COLUMNS (emp\_post string COMMENT ‘This is employee post’, emp\_age INT COMMENT ‘This is employee age’);

Syn to set table properties:

**ALTER TABLE table\_name SET TBLPROPERTIES (‘property\_key’=’property\_new\_value’);**

hive> ALTER TABLE employee.facultytable SET TBLPROPERTIES (‘table for’=’faculty data’);

**5) DROP**tablestatement in Hive deletes the data for a particular table and remove all metadata associated with it from Hive metastore.

* If PURGE is not specified, then the data is actually moved to the .Trash/current directory.
* If PURGE is specified, then data is lost completely.

Syn: **DROP TABLE [IF EXISTS] table\_name [PURGE];**

hive> DROP TABLE IF EXISTS employee.emptable PURGE;

OK

**6) TRUNCATE** table statement in Hive removes all the rows from the table or partition.

Syn: **TRUNCATE TABLE table\_name;**

hive> TRUNCATE TABLE employee.emptable;

OK

|  |  |
| --- | --- |
| Week-9 | Hive Data Manipulation Language(DML) Commands |
| 9 | ImplementData Manipulation Language (DML) Commandsfortablesin Hadoop Hiveframework using Cloudera. |

* Open Virtual box and then start **clouderaquickstartTerminal**and type “hive” to launch hive shell

**DML Commands for Tables**

**1) LOAD** statement in Hive is used to copy/move data files into the locations corresponding to Hive tables.

Syn: **LOAD DATA [LOCAL] INPATH 'filepath' [OVERWRITE] INTO TABLE tablename [PARTITION (partcol1=val1, partcol2=val2 ...)];**

LOCAL keyword = file path in the local filesystem.

LOCAL not specified = file path in the hdfs

OVERWRITE contents of the target table (or partition) will be deleted and replaced by the files otherwise contents are added to the table

hive> LOAD DATA LOCAL INPATH ‘/home/cloudera/HiveDir/emptextdata' INTO TABLE employee.facultytable;

OK

emptextdata contents

1,bob,25000.00,asstprof,35,male

2,mary,35000.00,assocprof,38,female

3,mike,50000.00,prof,45,male

**2)** **SELECT** statement in Hive is similar to the SELECT statement in SQL used for retrieving data from the database.

Syn: **SELECT \* FROM tablename;** //displays all records

hive> SELECT \* FROM employee.facultytable;

1 bob 25000.00 asstprof 35 male

2 mary 35000.00 assocprof 38 female

3 mike 50000.00 prof 45 male

**SELECT col1,col2 FROM tablename;** //Retrieves only specified columns data

hive> SELECT emp\_name,emp\_salary FROM employee.facultytable;

bob 25000.00

mary 35000.00

mike 50000.00

**3) a) INSERT INTO** statement appends the data into existing data in the table or partition.

Syn: **INSERT INTO TABLE tablename [PARTITION (partcol1=val1, partcol2=val2 ...)] VALUES (col1value,col2value,…)**

hive> INSERT INTO TABLE employee.facultytable VALUES (4, ‘jessy’, 45000.00, ‘assocprof’, 40, ‘female’);

hive> SELECT \* FROM employee.facultytable;

4 jessy 45000.00 assocprof 40 female

1 bob 25000.00 asstprof 35 male

2 mary 35000.00 assocprof 38 female

3 mike 50000.00 prof 45 male

**b) INSERT OVERWRITE** table overwrites the existing data in the table or partition.

Syn: **INSERT OVERWRITE TABLE tablename1 [PARTITION (partcol1=val1, ..) [IF NOT EXISTS]] select\_statement FROM from\_statement;**

**4)** **DELETE** statement in Hive deletes the table data. If the WHERE clause is specified, then it deletes the rows that satisfy the condition in where clause.

Syn: **DELETE FROM tablename [WHERE expression];**

hive> DELETE FROM employee.facultytable WHERE emp\_age=38;

hive> SELECT \* FROM employee.facultytable;

4 jessy 45000.00 assocprof 40 female

1 bob 25000.00 asstprof 35 male

3 mike 50000.00 prof 45 male

**5)UPDATE** statement in Hive deletes the table data. If the WHERE clause is specified, then it updates the column of the rows that satisfy the condition in WHERE clause. Partitioning and Bucketing columns cannot be updated.

Syn: **UPDATE tablename SET column = value [, column = value ...] [WHERE expression];**

hive> UPDATE employee.facultytable SET emp\_name = ‘mike tyson’ WHERE emp\_age=45;

hive> SELECT \* FROM employee.facultytable;

4 jessy 45000.00 assocprof 40 female

1 bob 25000.00 asstprof 35 male

3 mike tyson 50000.00 prof 45 male

**6)EXPORT** statement exports the table or partition data along with the metadata to the specified output location in the HDFS. Metadata is exported in a \_metadata file, and data is exported in a subdirectory ‘data.’

Syn: **EXPORT TABLE tablename [PARTITION (part\_column="value"[, ...])] TO 'export\_target\_path' [ FOR replication('eventid') ];**

hive> EXPORT TABLE employee.drivertable TO ‘/user/hive/warehouse’;

**7)IMPORT** command imports the data from a specified location to a new table or already existing table.

Syn: **IMPORT [[EXTERNAL] TABLE new\_or\_original\_tablename [PARTITION (part\_column="value"[, ...])]] FROM 'source\_path' [LOCATION 'import\_target\_path’];**

hive> IMPORT TABLE employee.importedtable FROM ‘/user/hive/warehouse’;