● **Hive Data Definitions**

The Driver for Apache Hive supports a broad set of DDL, including (but not limited to) the following:

1. CREATE Database and DROP Database
2. CREATE Table and DROP Table
3. ALTER Table and Alter Partition statements
4. CREATE View and Drop View
5. CREATE Function and Drop Function

CREATE Database and DROP Database:

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

  [COMMENT database\_comment]

  [LOCATION hdfs\_path]

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

The uses of SCHEMA and DATABASE are interchangeable – they mean the same thing.

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

CREATE Table and DROP Table:

e.g.

CREATE TABLE IF NOT EXISTS

custom.olympix\_data (athlete\_name string,athlete\_age int,athlete\_country string, year int,closing\_date string,sport string,gold\_medals int,

silver\_medals int,bronze\_medals int,total\_medals int)

ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';

This query will create a table in the database custom with table name olympix\_data.

DROP TABLE custom.olympix\_data;

It will drop the table olympix\_data from the database “custom”.

### Alter Table

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

This statement will change the name of a table to a different name.

### Alter Partition

Partitions can be added, renamed, exchanged (moved), dropped, or (un)archived by using the PARTITION clause in an ALTER TABLE statement

ALTER TABLE table\_name ADD [IF NOT EXISTS] PARTITION partition\_spec [LOCATION 'location'][, PARTITION partition\_spec [LOCATION 'location'], ...];

partition\_spec:

  : (partition\_column = partition\_col\_value, partition\_column = partition\_col\_value, ...)

**Create View**

|  |
| --- |
| CREATE VIEW [IF NOT EXISTS] [db\_name.]view\_name [(column\_name [COMMENT column\_comment], ...) ]    [COMMENT view\_comment]    [TBLPROPERTIES (property\_name = property\_value, ...)]    AS SELECT ...; |

CREATE VIEW creates a view with the given name. An error is thrown if a table or view with the same name already exists. we can use IF NOT EXISTS to skip the error.

### Drop View

DROP VIEW removes metadata for the specified view.

DROP VIEW [IF EXISTS] [db\_name.]view\_name;

### Create Function

This statement lets you create a function that is implemented by the class\_name. Jars, files, or archives which need to be added to the environment can be specified with the USING clause; when the function is referenced for the first time by a Hive session, these resources will be added to the environment as if add jar file had been issued.

CREATE FUNCTION [db\_name.]function\_name AS class\_name

  [USING JAR|FILE|ARCHIVE 'file\_uri' [, JAR|FILE|ARCHIVE 'file\_uri'] ];

● **Hive Data Manipulations**

Loading files into tables

Hive does not do any transformation while loading data into tables. Load operations are currently pure copy/move operations that move datafiles into locations corresponding to Hive tables.

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

### Update

Updates can only be performed on tables that support ACID.

UPDATE tablename SET column = value [, column = value ...] [WHERE expression]

### Delete

Deletes can only be performed on tables that support ACID

DELETE FROM tablename [WHERE expression]

### Merge

Merge can only be performed on tables that support ACID

MERGE INTO <target table> AS T USING <source expression/table> AS S

ON <boolean expression1>

WHEN MATCHED [AND <boolean expression2>] THEN UPDATE SET <set clause list>

WHEN MATCHED [AND <boolean expression3>] THEN DELETE

WHEN NOT MATCHED [AND <boolean expression4>] THEN INSERT VALUES<value list>

● HiveQL Manipulations

1. Loading Data into Managed Tables.  
  
2. Inserting Data into Tables from Queries.  
  
  
3. Creating Tables and Loading Them in One Query.  
  
4. Exporting Data.

Loading Data into Managed Tables:

LOAD DATA LOCAL INPATH '/home/acadgild/Desktop/assignments\_work/Data\_from\_acadgild/6.1/dataset\_Session.txt' into table CUSTOM.TEMPERATURE\_DATA;

It will load data in the table TEMPERATURE\_DATA of the custom database. Data will be loaded from the text file dataset\_Session.txt.

Inserting Data into Tables from Queries:

INSERT OVERWRITE TABLE employees  
PARTITION (country, state)  
SELECT ..., se.cnty, se.st  
FROM staged\_employees se;

Creating Tables and Loading Them in One Query

CREATE TABLE ca\_employees  
AS SELECT name, salary, address  
FROM employees  
WHERE se.state = 'CA';

Exporting Data

hive -e 'select \* from temperature\_data\_vw;' | sed 's/[[:space:]]\+/\|/g'  > /home/sahil/Desktop/temperature\_data\_vw.txt;

It will store the data fetched from the query in a text file named temperature\_data\_vw.txt on the desktop of the local machine.