

## Assignment 15.3

### Hive Data Definitions:

**Hive Data Definition** Language. **Hive Data Definition** Language (DDL) is a subset of **Hive** SQL statements that describe the **data** structure in **Hive** by creating, deleting, or altering schema objects such as databases, tables, views, partitions, and buckets.

Example: Create Database and table

```
CREATE DATABASES IF NOT EXISTS custom;
USE custom;
CREATE TABLE olympic_data
(
  name STRING,
  age INT,
  country STRING,
  year INT,
  closingDate STRING,
  sport STRING,
  gold INT,
  silver INT,
  bronze INT,
  total INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t';
```

### Hive Data Manipulations:

**Hive Data Manipulations** Language. **Hive Data Manipulations** Language (DML) is a subset of **Hive** SQL statements that is used for loading data in the hive schemas.

Example:

```
LOAD DATA
LOCAL INPATH '/home/acadgild/hive/olympix_data.csv'
INTO TABLE olympic_data;
```

### HiveQL Manipulations:

#### 1. Loading Data into Managed Tables.

Since Hive has no row level insert, update and delete operations, the only way to put data into table is to use one of the "bulk" load operations. Or you can just write files in the correct directories by other means.

Example:

```
LOAD DATA LOCAL INPATH "/user/acadgild/data/employee.txt"
OVERWRITE INTO TABLE employees
PARTITION (country='US', state='CA');
```

The command will first create the directory for the partition, if it doesn't already exist, then copy data to it. If the target table is not partitioned, you omit the PARTITION clause.

## 2. Inserting Data into Tables from Queries.

The INSERT statement lets you load data into a table from a query.

here is an example of Oregon, where we presume data is already in another table called staged\_employees.

```
INSERT OVERWRITE TABLE employee
PARTITION (country='US', state='OR')
SELECT * FROM staged_employees se
WHERE se.cnty='US' AND se.st='OR';
```

with OVERWRITE, any previous contents of the partition (or whole table if not partitioned) are replaced. If you drop the keyword OVERWRITE or replace it with INTO, hive appends the data rather than replaces it. This feature is only available in Hive v0.8.0 or later.

### i. Dynamic Partition Inserts.

There's still one problem with this syntax: if you have a lot of partitions to create, you have to write a lot of SQL!

Fortunately, Hive also supports dynamic partition feature, where it can infer the partitions to create based on query parameters. By comparison, up until now we have considered only static partitions.

```
INSERT OVERWRITE TABLE employee
PARTITION (country, state)
SELECT ..., se.cnty, se.st FROM staged_employees se;
```

## 3. Creating Tables and Loading Them in One Query.

You can also create a table and insert query results into it in one statement:

```
CREATE TABLE ca_employees
AS SELECT name, salary, address
FROM employees
WHERE state='CA';
```

This table contains just the name, salary and address column from the employee table records for employee in California. The schema for the new table is taken from the SELECT clause.

The common use of this feature is to extract a convenient subset of data from a larger, more unwieldy table.

## 4. Exporting Data.

How do we get data out of tables? If data files are already formatted the way you want, then it's simple enough to copy the directories or files.

```
hadoop fs -cp source_path target_path
```

Otherwise, you can use INSERT....DIRECTORY...., as in this example:

```
INSERT OVERWRITE LOCAL DIRECTORY '/user/acadgild/export/ca_employees'
SELECT name,salary,address
FROM employees
WHERE state='CA'
```