**Hive QL Manipulations**

**HiveQL: Data Manipulation:**

1. Loading Data into Managed Tables.  
  
2. Inserting Data into Tables from Queries.  
I. Dynamic Partition Inserts.  
  
3. Creating Tables and Loading Them in One Query.  
  
4. Exporting Data.  
  
  
**CONTENTS**  
Since Hive has no row-level insert, update, and delete operations, the only way to put  
data into a table is to use one of the “bulk” load operations. Or you can just write files  
in the correct directories by other means. Loading Data into Managed Tables.  
The INSERT statement lets you load data into a table from a query. Inserting Data into Tables from Queries how do we get data out of tables? If the 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 '/tmp/ca\_employees'  
SELECT name, salary, address  
FROM employees  
WHERE se.state = 'CA';  
Exporting Data

**HivQL: Data Manipulation**  
LOAD DATA LOCAL INPATH   
'${env:HOME}/california-employees'  
OVERWRITE INTO TABLE employees  
PARTITION (country = '  
US  
', state = '  
CA  
');  
This command will first create the directory for the partition, if it doesn’t already exist, then copy the data to it. If the target table is not partitioned, you omit the PARTITION clause. Here is an example for the state of Oregon, where we presume the data is already in another table called staged\_employees.

**INSERT OVERWRITE TABLE employees 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 a  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.  
Consider this change to the previous example:  
  
**INSERT OVERWRITE TABLE employees  
PARTITION (country, state)  
SELECT ..., se.cnty, se.st  
FROM staged\_employees se;**  
  
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 se.state = 'CA';**  
This table contains just the name, salary, and address columns from the employee table records for employees in California. The schema for the new table is taken from the SELECT clause. A common use for this feature is to extract a convenient subset of data from a larger, more unwieldy table. The specified path can also be a full URI (e.g., hdfs://master-server/tmp/ca\_employees). As a reminder, we can look at the results from  
within the hive CLI:  
  
**hive> ! ls /tmp/ca\_employees;  
000000\_0  
hive> ! cat /tmp/payroll/000000\_0  
John Doe100000.0201 San Antonio CircleMountain ViewCA94040  
Mary Smith80000.01 Infinity LoopCupertinoCA95014**