

|  |
| --- |
| * **Hive Data Definitions:** |
|  |

|  |
| --- |
| Data definition language is part of HiveQL, which are used for creating, altering, and dropping databases, tables, views, functions, and indexes. |
|  |

|  |
| --- |
| **EXAMPLES:** |
|  |

|  |
| --- |
| * **Create command** |
|  |

|  |
| --- |
| The CREATE TABLE statement follows SQL conventions, but Hive’s version offers significant extensions to support a wide range of flexibility where the data files for tables are stored, the formats used, etc. |
|  |

|  |
| --- |
|  |
|  |

|  |  |
| --- | --- |
| CREATE DATABASE department ; | |
|  |
| If you add the option IF NOT EXISTS with create command, Hive will silently ignore the statement if the table already exists. This is useful in scripts that should create a table the first time they run.  **CREATE TABLE IF NOT EXISTS mydb.employees2;**  Where mydb is our database and employee2 is tablename | |
|  | |

|  |
| --- |
|  |
|  |

|  |
| --- |
| * **Drop command** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| However, the clause has a gotcha you should know. If the schema specified differs from the schema in the table that already exists, Hive won’t warn you. If your intention is for this table to have the new schema, you’ll have to drop the old table, losing your data, and then re-create it. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **DROP DATABASE IF EXISTS human\_resource;** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| * **Alter command** |
|  |

|  |
| --- |
| Most table properties can be altered with ALTER TABLE statements, which change metadata about the table but not the data itself. These statements can be used to fix mistakes in schema, move partition locations, and do other operations. |
|  |

|  |
| --- |
| **ALTER TABLE messages RENAME TO logmsgs;** |
|  |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| * **Hive Data Manipulations:** |
|  |
|  |

|  |
| --- |
| The data manipulation language parts of HiveQL that are used to put data into Hive tables and to extract data to the file system, and how to explore and manipulate data with queries, grouping, filtering, joining, etc. load the data |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **EXAMPLES:** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| LOAD DATA LOCAL INPATH '${env:HOME}/cognizant-employees' |
|  |

|  |
| --- |
| OVERWRITE INTO TABLE employees |
|  |

|  |
| --- |
| PARTITION (country = 'US', state = 'CA'); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
| --------------------------------------------------------------------------------------------------------------------- |

|  |
| --- |
| * **HiveQL Manipulations:** |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **EXAMPLES:** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **• LOAD** |
|  |

|  |
| --- |
| 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. |
|  |

|  |
| --- |
| **Syntax** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| LOAD DATA [LOCAL] INPATH 'filepath' [OVERWRITE] INTO TABLE tablename |
|  |

|  |
| --- |
| [PARTITION (partcol1=val1, partcol2=val2 ...)] |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **• INSERT** |
|  |

|  |
| --- |
| Inserting data into Hive Tables from queries |
|  |

|  |
| --- |
| Query Results can be inserted into tables by using the insert clause. |
|  |
|  |

|  |
| --- |
| **Syntax** |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| INSERT OVERWRITE TABLE tablename1 [PARTITION (partcol1=val1, partcol2=val2 ...) |
|  |

|  |
| --- |
| [IF NOT EXISTS]] select\_statement1 FROM from\_statement; |
|  |

|  |
| --- |
| INSERT INTO TABLE tablename1 [PARTITION (partcol1=val1, partcol2=val2 ...)] |
|  |

|  |
| --- |
| select\_statement1 FROM from\_statement; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **• UPDATE** |
|  |
|  |

|  |
| --- |
| **Syntax** |
|  |

|  |
| --- |
|  |
|  |

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

|  |
| --- |
|  |
|  |

|  |
| --- |
| **• DELETE** |
|  |
|  |

|  |
| --- |
| **Syntax** |
|  |

|  |
| --- |
| DELETE FROM tablename [WHERE expression] |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| • **MERGE** |
|  |
|  |

|  |
| --- |
| **Syntax** |
|  |

|  |
| --- |
| 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> |
|  |