**Regularly used common commands:**

|  |  |
| --- | --- |
| **DESCRIPTION** | **COMMAND** |
| **Autocomplete** | hive> Press Tab key Display all 436 possibilities? (y or n)If you enter y, you’ll get a long list of all the keywords |
| **Navigation Keystrokes** | Use the up arrow and down arrow keys to scroll through previous commandsCtrl+A goes to the beginning of the lineCtrl+E goes to the end of the lineDelete key will delete the character to the left of the cursor |
| **Command History** | Hive saves the last 100,00 lines into a file $HOME/.hivehistory |
| **Shell Execution** | type **!** followed by the command and terminate the line with a semicolon (**;**) hive> ! /bin/echo “Hello World”; Hello World hive> ! pwd;/home/me/hiveplay (Note: Don’t invoke interactive commands that require user input. Shell “pipes” don’t work and neither do file “globs.”For example, ! ls \*.hql; will look for a file named ***\*.hql;***, rather than all files that end with the ***.hql*** extension.) |
| **To Print Current DB in use** | set hive.cli.print.current.db=true; (or) set hiveconf:hive.cli.print.current.db=true; |
| **To remove current db name display in hive shell** | set hiveconf:hive.cli.print.current.db=false; |
| **Specifying Metastore location for each user** | set hive.metastore.warehouse.dir=/user/myname/hive/warehouse; |
| ***system* Namespace (provides read-write access to Java system properties)** | set system:user.name; (or) set system:user.name=yourusername; |
| ***env* Namespace (provides read-only access to environment variables)** | set env:HOME; |
| **Hadoop dfs commands inside Hive shell** | Exclude hadoop keyword and end the command with semicolon(**;**) as below: hive> dfs -ls / ; (Note: This method of accessing hadoop commands is actually more efficient than using the ***hadoop dfs …***equivalent at the bash shell, because the latter starts up a new JVM instance each time, whereas Hive just runs the same code in its current process.) |
| **Execute hive queries from a *.hql*file** | source /unix-path/to/file/withqueries.hql; |
| **Print Column Headers** | set hive.cli.print.header=true; |
| **Show complete details of a table** | SHOW CREATE TABLE mytable; (or) DESCRIBE [FORMATTED] [db\_name.]table\_name[.complex\_col\_name …] (or) DESCRIBE EXTENDED mytable;  hive> SHOW CREATE TABLE employees;  OK  CREATE TABLE `employees`(  `emplid` string COMMENT 'from deserializer',  `name` string COMMENT 'from deserializer',  `age` string COMMENT 'from deserializer',  `salary` string COMMENT 'from deserializer',  `dept` string COMMENT 'from deserializer')  ROW FORMAT SERDE  'org.apache.hadoop.hive.contrib.serde2.RegexSerDe'  WITH SERDEPROPERTIES (  'input.regex'='(.{4})(.{35})(.{3})(.{11})(.{4})')  STORED AS INPUTFORMAT  'org.apache.hadoop.mapred.TextInputFormat'  OUTPUTFORMAT  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'  LOCATION  'hdfs://sandbox.hortonworks.com:8020/user/hue/tmp/fixed\_employees'  TBLPROPERTIES (  'COLUMN\_STATS\_ACCURATE'='true',  'numFiles'='0',  'totalSize'='0',  'transient\_lastDdlTime'='1455035524')  Time taken: 3.399 seconds, Fetched: 21 row(s) |
| **Get columns names of the table** | SHOW COLUMNS FROM mytable; |
| **Load data from a local file to the hive table** | LOAD DATA LOCAL INPATH ‘/unix-path/myfile’ INTO TABLE mytable; |
| **Load data from hdfs file to the hive table** | LOAD DATA INPATH ‘/hdfs-path/myfile’ INTO TABLE mytable; |
| **Data Types** | **Numeric Data Types:** >> **TINYINT** (1-byte signed integer, from -128 to 127) >> **SMALLINT** (2-byte signed integer, from -32,768 to 32,767) >> **INT** (4-byte signed integer, from -2,147,483,648 to 2,147,483,647) >> **BIGINT** (8-byte signed integer, from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807) >> **FLOAT** (4-byte single precision floating point number) >> **DOUBLE** (8-byte double precision floating point number) >> **DECIMAL** (or) DECIMAL(precision, scale) (Precision of 38 digits. User definable precision and scale)  **Date/Time Types:** >> **TIMESTAMP** (UTC time. Format ***‘YYYY-MM-DD HH:MM:SS.fffffffff’*** (9 decimal place precision) ***Ex: ‘2012-02-03 12:34:56.123456789’*** >> **DATE** (Format: ***‘YYYY-­MM-­DD’*** The range of values supported for the Date type is be ***0000-­01-­01 to 9999-­12-­31***, dependent on support by the primitive Java Date type.)  **String Types:** >> **STRING** >> **VARCHAR** (Length specifier between 1 and 65355) >> **CHAR** (Fixed-length. The maximum length is fixed at 255)  **Misc Types:** >> **BOOLEAN** >> **BINARY**  **Complex Types:** >> **arrays: ARRAY** >> **maps: MAP** >> **structs: STRUCT** >> **union: UNIONTYPE** |

**Dynamic Partition**  
Given below are the configuration properties for *dynamic partition* inserts. **Note** by default *dynamic partition* inserts are disabled.

|  |  |  |
| --- | --- | --- |
| **CONFIGURATION PROPERTY** | **DEFAULT** | **NOTE** |
| hive.exec.dynamic.partition | false | Needs to be set to true to enable dynamic partition inserts |
| hive.exec.dynamic.partition.mode | strict | In strict mode, the user must specify at least one static partition in case the user accidentally overwrites all partitions, in nonstrict mode all partitions are allowed to be dynamic |
| hive.exec.max.dynamic.partitions.pernode | 100 | Maximum number of dynamic partitions allowed to be created in each mapper/reducer node |
| hive.exec.max.dynamic.partitions | 1000 | Maximum number of dynamic partitions allowed to be created in total |
| hive.exec.max.created.files | 100000 | Maximum number of HDFS files created by all mappers/reducers in a MapReduce job |
| hive.error.on.empty.partition | false | Whether to throw an exception if dynamic partition insert generates empty results |

**Hive One Shot commands**

|  |  |
| --- | --- |
| **DESCRIPTION** | **COMMAND** |
| **To Print Current DB in use** | $ hive –hiveconf hive.cli.print.current.db=true |
| **Specify a file of commands for the CLI to run as it starts, before showing you the prompt** | $ cat hiveproperties.txt set hive.cli.print.current.db=true; set system:user.name;  $ hive -i hiveproperties.txt system:user.name=yourusername hive> |
| **Adding the *-e* execute Hive queries** | $ hive -e “SELECT \* FROM mytable LIMIT 3”; |
| **Adding the *-S* for silent mode removes the OK and Time taken … lines, as well as other inessential output** | $ hive -S -e “SELECT \* FROM mytable LIMIT 3” |
| **Useful trick for finding a property name that you can’t quite remember** | $ hive -S -e “set” | grep warehouse\_or\_pattern |
| **Comments in Hive scripts starts with double hyphen (‐‐) followed by space and then comment description. Hive scripts have the extension *.hql*** | $ cat hivescript.hql ‐‐ Comment line1 ‐‐ Comment line2 SELECT \* FROM mytable LIMIT 3; |
| **Executing Hive Queries from *.hql* files** | $ hive -f /unix-path/to/file/hivescript.hql |
| **Hive variables (The env namespace is useful as an alternative way to pass variable definitions to Hive)** | $ YEAR=2012 hive -e “SELECT \* FROM mytable WHERE year = ${env:YEAR}”; |