**Hive Data Definitions**

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

* CREATE Database and DROP Database
* CREATE Table and DROP Table
* ALTER Table and Alter Partition statements
* CREATE View and Drop View
* CREATE Function and Drop Function

REGEXP and RLIKE are non-reserved keywords prior to Hive 2.0.0 and reserved keywords starting in Hive 2.0.0

**Create/Drop/Alter/Use Database**

Create 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. CREATE DATABASE was added in Hive 0.6

**Drop Database:**

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

The uses of SCHEMA and DATABASE are interchangeable – they mean the same thing. DROP DATABASE was added in Hive 0.6 ([HIVE-675](https://issues.apache.org/jira/browse/HIVE-675)). The default behavior is RESTRICT, where DROP DATABASE will fail if the database is not empty. To drop the tables in the database as well, use DROP DATABASE ... CASCADE. Support for RESTRICT and CASCADE was added in Hive 0.8

**Alter Database**

|  |
| --- |
| ALTER (DATABASE|SCHEMA) database\_name SET DBPROPERTIES (property\_name=property\_value, ...);  ALTER (DATABASE|SCHEMA) database\_name SET OWNER [USER|ROLE] user\_or\_role;  ALTER (DATABASE|SCHEMA) database\_name SET LOCATION hdfs\_path; -- (Note: Hive 2.2.1, 2.4.0 and later) |

The uses of SCHEMA and DATABASE are interchangeable – they mean the same thing. ALTER SCHEMA was added in Hive 0.14 ([HIVE-6601](https://issues.apache.org/jira/browse/HIVE-6601)).

No other metadata about a database can be changed.

**Use Database:**

|  |
| --- |
| USE database\_name;  USE DEFAULT; |

USE sets the current database for all subsequent HiveQL statements. To revert to the default database, use the keyword "default" instead of a database name. To check which database is currently being used: SELECT [current\_database()](https://cwiki.apache.org/confluence/display/Hive/LanguageManual+UDF#LanguageManualUDF-Misc.Functions) (as of [Hive 0.13.0](https://issues.apache.org/jira/browse/HIVE-4144)).

USE database\_name was added in Hive 0.6

**Create Table**

CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS] [db\_name.]table\_name    -- (Note: TEMPORARY available in Hive 0.14.0 and later)

  [(col\_name data\_type [COMMENT col\_comment], ... [constraint\_specification])]

  [COMMENT table\_comment]

  [PARTITIONED BY (col\_name data\_type [COMMENT col\_comment], ...)]

  [CLUSTERED BY (col\_name, col\_name, ...) [SORTED BY (col\_name [ASC|DESC], ...)] INTO num\_buckets BUCKETS]

  [SKEWED BY (col\_name, col\_name, ...)                  -- (Note: Available in Hive 0.10.0 and later)]

     ON ((col\_value, col\_value, ...), (col\_value, col\_value, ...), ...)

     [STORED AS DIRECTORIES]

  [

   [ROW FORMAT row\_format]

   [STORED AS file\_format]

     | STORED BY 'storage.handler.class.name' [WITH SERDEPROPERTIES (...)]  -- (Note: Available in Hive 0.6.0 and later)

  ]

  [LOCATION hdfs\_path]

  [TBLPROPERTIES (property\_name=property\_value, ...)]   -- (Note: Available in Hive 0.6.0 and later)

  [AS select\_statement];   -- (Note: Available in Hive 0.5.0 and later; not supported for external tables)

CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS] [db\_name.]table\_name

  LIKE existing\_table\_or\_view\_name

  [LOCATION hdfs\_path];

data\_type

  : primitive\_type

  | array\_type

  | map\_type

  | struct\_type

  | union\_type  -- (Note: Available in Hive 0.7.0 and later)

primitive\_type

  : TINYINT

  | SMALLINT

  | INT

  | BIGINT

  | BOOLEAN

  | FLOAT

  | DOUBLE

  | DOUBLE PRECISION -- (Note: Available in Hive 2.2.0 and later)

  | STRING

  | BINARY      -- (Note: Available in Hive 0.8.0 and later)

  | TIMESTAMP   -- (Note: Available in Hive 0.8.0 and later)

  | DECIMAL     -- (Note: Available in Hive 0.11.0 and later)

  | DECIMAL(precision, scale)  -- (Note: Available in Hive 0.13.0 and later)

  | DATE        -- (Note: Available in Hive 0.12.0 and later)

  | VARCHAR     -- (Note: Available in Hive 0.12.0 and later)

  | CHAR        -- (Note: Available in Hive 0.13.0 and later)

array\_type

  : ARRAY < data\_type >

map\_type

  : MAP < primitive\_type, data\_type >

struct\_type

  : STRUCT < col\_name : data\_type [COMMENT col\_comment], ...>

union\_type

   : UNIONTYPE < data\_type, data\_type, ... >  -- (Note: Available in Hive 0.7.0 and later)

row\_format

  : DELIMITED [FIELDS TERMINATED BY char [ESCAPED BY char]] [COLLECTION ITEMS TERMINATED BY char]

        [MAP KEYS TERMINATED BY char] [LINES TERMINATED BY char]

        [NULL DEFINED AS char]   -- (Note: Available in Hive 0.13 and later)

  | SERDE serde\_name [WITH SERDEPROPERTIES (property\_name=property\_value, property\_name=property\_value, ...)]

file\_format:

  : SEQUENCEFILE

  | TEXTFILE    -- (Default, depending on hive.default.fileformat configuration)

  | RCFILE      -- (Note: Available in Hive 0.6.0 and later)

  | ORC         -- (Note: Available in Hive 0.11.0 and later)

  | PARQUET     -- (Note: Available in Hive 0.13.0 and later)

  | AVRO        -- (Note: Available in Hive 0.14.0 and later)

  | INPUTFORMAT input\_format\_classname OUTPUTFORMAT output\_format\_classname

constraint\_specification:

  : [, PRIMARY KEY (col\_name, ...) DISABLE NOVALIDATE ]

    [, CONSTRAINT constraint\_name FOREIGN KEY (col\_name, ...) REFERENCES table\_name(col\_name, ...) DISABLE NOVALIDATE

|  |
| --- |
|  |

#### Row Formats & SerDe:

You can create tables with a custom SerDe or using a native SerDe. A native SerDe is used if ROW FORMAT is not specified or ROW FORMAT DELIMITED is specified.   
Use the SERDE clause to create a table with a custom SerDe. For more information on SerDes see:

* [Hive SerDe](https://cwiki.apache.org/confluence/display/Hive/DeveloperGuide#DeveloperGuide-HiveSerDe)
* [SerDe](https://cwiki.apache.org/confluence/display/Hive/SerDe)
* [HCatalog Storage Formats](https://cwiki.apache.org/confluence/display/Hive/HCatalog+StorageFormats)

You must specify a list of columns for tables that use a native SerDe. Refer to the [Types](https://cwiki.apache.org/confluence/display/Hive/LanguageManual+Types) part of the User Guide for the allowable column types.   
A list of columns for tables that use a custom SerDe may be specified but Hive will query the SerDe to determine the actual list of columns for this table.

For general information about SerDes, see [Hive SerDe](https://cwiki.apache.org/confluence/display/Hive/DeveloperGuide#DeveloperGuide-HiveSerDe) in the Developer Guide. Also see [SerDe](https://cwiki.apache.org/confluence/display/Hive/SerDe) for details about input and output processing.

To change a table's SerDe or SERDEPROPERTIES, use the ALTER TABLE statement as described below in [Add SerDe Properties](https://cwiki.apache.org/confluence/display/Hive/LanguageManual+DDL#LanguageManualDDL-AddSerDeProperties).

| Row Format | Description |
| --- | --- |
| **RegEx**  ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe' WITH SERDEPROPERTIES  ( "input.regex" = "<regex>" ) STORED AS TEXTFILE; | Stored as plain text file, translated by Regular Expression.  The following example defines a table in the default Apache Weblog format.    CREATE TABLE apachelog (    host STRING,    identity STRING,    user STRING,    time STRING,    request STRING,    status STRING,    size STRING,    referer STRING,    agent STRING)  ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'  WITH SERDEPROPERTIES (    "input.regex" = "([^]\*) ([^]\*) ([^]\*) (-|\\[^\\]\*\\]) ([^ \"]\*|\"[^\"]\*\") (-|[0-9]\*) (-|[0-9]\*)(?: ([^ \"]\*|\".\*\") ([^ \"]\*|\".\*\"))?"  )  STORED AS TEXTFILE;  More about RegexSerDe can be found here in [HIVE-662](https://issues.apache.org/jira/browse/HIVE-662) and [HIVE-1719](https://issues.apache.org/jira/browse/HIVE-1719). |
| **JSON**   ROW FORMAT SERDE  'org.apache.hive.hcatalog.data.JsonSerDe'  STORED AS TEXTFILE | Stored as plain text file in JSON format.  The JsonSerDe for JSON files is available in [Hive 0.12](https://issues.apache.org/jira/browse/HIVE-4895) and later.  In some distributions, a reference to hive-hcatalog-core.jar is required.  ADD JAR /usr/lib/hive-hcatalog/lib/hive-hcatalog-core.jar;  CREATE TABLE my\_table(a string, b bigint, ...)  ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'  STORED AS TEXTFILE;  The JsonSerDe was moved to Hive from HCatalog and before it was in hive-contrib project. It was added to the Hive distribution by [HIVE-4895](https://issues.apache.org/jira/browse/HIVE-4895). An Amazon SerDe is available at s3://elasticmapreduce/samples/hive-ads/libs/jsonserde.jar for releases prior to 0.12.0.  The JsonSerDe for JSON files is available in [Hive 0.12](https://issues.apache.org/jira/browse/HIVE-4895) and later. |
| **CSV/TSV**  ROW FORMAT SERDE  'org.apache.hadoop.hive.serde2.OpenCSVSerde'  STORED AS TEXTFILE | Stored as plain text file in CSV / TSV format.  The CSVSerde is available in [Hive 0.14](https://issues.apache.org/jira/browse/HIVE-7777) and greater.  The following example creates a TSV (Tab-separated) file.    CREATE TABLE my\_table(a string, b string, ...) ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'  WITH SERDEPROPERTIES (     "separatorChar" = "\t",     "quoteChar"     = "'",     "escapeChar"    = "\\"  )  STORED AS TEXTFILE;  Default properties for SerDe is Comma-Separated (CSV) file  DEFAULT\_ESCAPE\_CHARACTER \  DEFAULT\_QUOTE\_CHARACTER  "  DEFAULT\_SEPARATOR        ,  This SerDe works for most CSV data, but does not handle embedded newlines. To use the SerDe, specify the fully qualified class name org.apache.hadoop.hive.serde2.OpenCSVSerde.  Documentation is based on original documentation at <https://github.com/ogrodnek/csv-serde>.  **Limitations**  This SerDe treats all columns to be of type String. Even if you create a table with non-string column types using this SerDe, the DESCRIBE TABLE output would show string column type.  The type information is retrieved from the SerDe.   To convert columns to the desired type in a table, you can create a view over the table that does the CAST to the desired type.  The CSV SerDe is based on <https://github.com/ogrodnek/csv-serde>, and was added to the Hive distribution in [HIVE-7777](https://issues.apache.org/jira/browse/HIVE-7777).  The CSVSerde has been built and tested against Hive 0.14 and later, and uses [Open-CSV](http://opencsv.sourceforge.net/) 2.3 which is bundled with the Hive distribution.  For general information about SerDes, see [Hive SerDe](https://cwiki.apache.org/confluence/display/Hive/DeveloperGuide#DeveloperGuide-HiveSerDe) in the Developer Guide. Also see [SerDe](https://cwiki.apache.org/confluence/display/Hive/SerDe) for details about input and output processing. |

#### Partitioned Tables:

Partitioned tables can be created using the PARTITIONED BY clause. A table can have one or more partition columns and a separate data directory is created for each distinct value combination in the partition columns. Further, tables or partitions can be bucketed using CLUSTERED BY columns, and data can be sorted within that bucket via SORT BY columns. This can improve performance on certain kinds of queries.

If, when creating a partitioned table, you get this error: "FAILED: Error in semantic analysis: Column repeated in partitioning columns," it means you are trying to include the partitioned column in the data of the table itself. You probably really do have the column defined. However, the partition you create makes a pseudocolumn on which you can query, so you must rename your table column to something else (that users should not query on!).

For example, suppose your original unpartitioned table had three columns: id, date, and name.

Example:

|  |
| --- |
| id     int,  date   date,  name   varchar |

Now you want to partition on date. Your Hive definition could use "dtDontQuery" as a column name so that "date" can be used for partitioning (and querying).

Example:

|  |
| --- |
|  |

create table table\_name (

  id                int,

  dtDontQuery       string,

  name              string

)

partitioned by (date string);