**ASSIGNMENT 24.1**

**Explain with an example in brief.**

**● Hive Data Definitions**

**● Hive Data Manipulations**

**● HiveQL Manipulations**

Ans)

HiveQL is the Hive query language. Like all SQL dialects in widespread use, it doesn’t fully conform to any particular revision of the ANSI SQL standard. It is perhaps closest to MySQL’s dialect, but with significant differences. Hive offers no support for row-level inserts, updates, and deletes. Hive doesn’t support transactions. Hive adds extensions to provide better performance in the context of Hadoop and to integrate with custom extensions and even external programs.

This chapter starts with the so-called data definition language parts of HiveQL, which are used for creating, altering, and dropping databases, tables, views, functions, and indexes.

Databases in Hive

The simplest syntax for creating a database is shown in the following example:

hive> **CREATE** **DATABASE** financials;

Hive will throw an error if financials already exists. You can suppress these warnings with this variation:

hive> **CREATE** **DATABASE** IF **NOT** **EXISTS** financials;

At any time, you can see the databases that already exist as follows:

hive> **SHOW** DATABASES;

**default**

financials

hive> **CREATE** **DATABASE** human\_resources;

hive> **SHOW** DATABASES;

**default**

financials

human\_resources

If you have a lot of databases, you can restrict the ones listed using a *regular expression*, a concept we’ll explain in [LIKE and RLIKE](https://www.safaribooksonline.com/library/view/programming-hive/9781449326944/ch06.html#LIKE-RLIKE), if it is new to you. The following example lists only those databases that start with the letter h and end with any other characters (the .\* part):

hive> **SHOW** DATABASES **LIKE** 'h.\*';

human\_resources

hive> ...

You can override this default location for the new directory as shown in this example:

hive> **CREATE** **DATABASE** financials

> **LOCATION** '/my/preferred/directory';

You can add a descriptive comment to the database, which will be shown by the DESCRIBE DATABASE <database> command.

hive> **CREATE** **DATABASE** financials

> **COMMENT** 'Holds all financial tables';

hive> **DESCRIBE** **DATABASE** financials;

financials Holds **all** financial tables

hdfs://master-server/**user**/hive/warehouse/financials.db

Alter Database

You can set key-value pairs in the DBPROPERTIES associated with a database using the ALTER DATABASE command. No other metadata about the database can be changed, including its name and directory location:

hive> **ALTER** **DATABASE** financials **SET** DBPROPERTIES ('edited-by' = 'Joe Dba');

Creating Tables

**CREATE** **TABLE** IF **NOT** **EXISTS** mydb.employees (

name STRING **COMMENT** 'Employee name',

salary FLOAT **COMMENT** 'Employee salary',

subordinates ARRAY<STRING> **COMMENT** 'Names of subordinates',

deductions **MAP**<STRING, FLOAT>

**COMMENT** 'Keys are deductions names, values are percentages',

address STRUCT<street:STRING, city:STRING, **state**:STRING, zip:INT>

**COMMENT** 'Home address')

**COMMENT** 'Description of the table'

TBLPROPERTIES ('creator'='me', 'created\_at'='2012-01-02 10:00:00', ...)

**LOCATION** '/user/hive/warehouse/mydb.db/employees';

HiveQL: Data Manipulation

# the Hive query language, focusing on the data manipulation language parts that are used to put data into tables aLoading Data into Managed Tables

Since Hive has no row-level insert, update, and delete operations, the only way to put data into an table is to use one of the “bulk” load operations.

and to extract data from tables to the filesystem.

**LOAD** **DATA** **LOCAL** INPATH '${env:HOME}/california-employees'

OVERWRITE **INTO** **TABLE** employees

PARTITION (country = 'US', **state** = 'CA');

LOAD DATA- Load operations are currently pure copy/move operations that move data files into locations corresponding to Hive tables.

LOAD DATA [LOCAL] INPATH 'filepath' [OVERWRITE] INTO TABLE tablename [PARTITION (partcol1=val1, partcol2=val2 ...)] •

file path can refer to a file (in which case Hive will move the file into the table) or it can be a directory (in which case Hive will move all the files within that directory into the table).

If the keyword LOCAL is specified, then:

the load command will look for filepath in the local file system.

If a relative path is specified, it will be interpreted relative to the user's current working directory.

The user can specify a full URI for local files as well, for example: file:///user/hive/project/data1

the load command will try to copy all the files addressed by filepath to the target filesystem.

If the OVERWRITE keyword is used then the contents of the target table (or partition) will be deleted and replaced by the files referred to by filepath, otherwise the files referred by filepath will be added to the table.

Query results can be inserted into filesystem directories.

INSERT OVERWRITE [LOCAL] DIRECTORY directory1 [ROW FORMAT row\_format] [STORED AS file\_format] (Note: Only available starting with Hive 0.11.0) SELECT ... FROM ...

**HIVEQL MANIPULATION**

The Hive Query Language (HiveQL) is a query language for Hive to process and analyze structured data in a Metastore.

1. SELECT- SELECT statement is used to retrieve the data from a table.

SELECT \* FROM emp\_details;

This statement will give all the data stored in table. Some clauses are used with SELECT statement to select data based on some condition.

1. WHERE- WHERE clause works similar to a condition. It filters the data using the condition and gives you a finite result.
2. ORDER BY- ORDER BY clause is used to retrieve the details based on one column and sort the result set by ascending or descending order.
3. GROUP BY- GROUP BY clause is used to group all the records in a result set using a particular collection column. It is used to query a group of records.