Hive & SQL

Sl. No.	Agenda Title
1	Introduction
2	Hive DDL
3	Demo: Databases.ddl
4	Demo: Tables.ddl
5	Hive Views
6	Demo: Views.ddl
7	Primary Data Types
8	Data Load
9	Demo: ImportExport.dml
10	Demo: HiveQueries.dml
11	Demo: Explain.hql
12	Table Types

Sl. No.	Agenda Title
13	Demo: ExternalTable.ddl
14	Complex Data Types
15	Demo: Working with Complex Datatypes
16	Hive Variables
17	Demo: Working with Hive Variables
18	Hive Variables and Execution Customisation
19	Demo: Working with Hive Execution

INTRODUCTION

A data warehouse solution built on top of Hadoop - by Facebook.

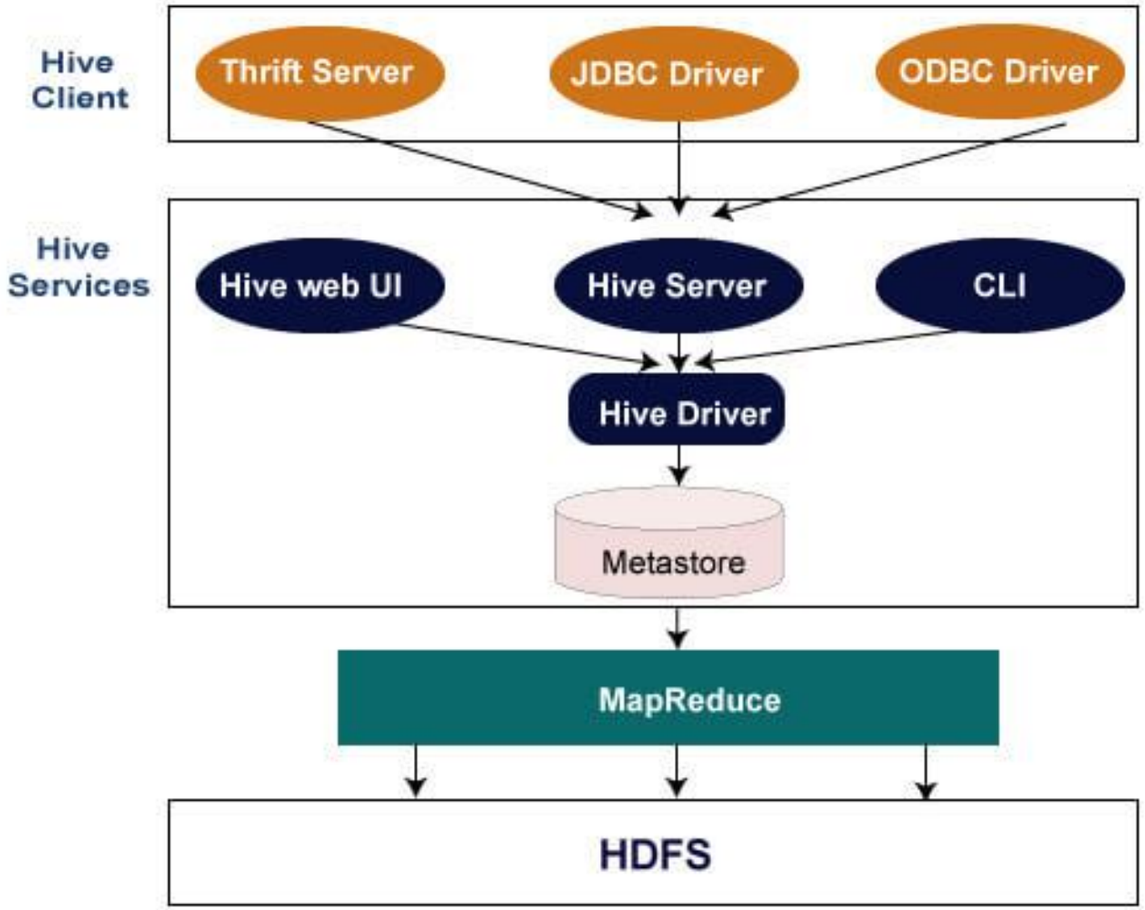
An essential tool in the Hadoop ecosystem that provides an SQL (Structured QueryLanguage) dialect (called as Hive Query Language) for querying data stored in the Hadoop Distributed Filesystem (HDFS).

Most data warehouse applications are implemented using relational databases that use SQL as the query language. Hive lowers the barrier for moving these applications to Hadoop. People who know SQL can learn Hive easily.

Automatically uses HDFS for storage, but stores all the meta information about database and table in metadata DB locally to Hive.

Hive is most suited for data warehouse applications, where relatively static data is analyzed, fast response times are not required, and when the data is not changing rapidly.

HIVE ARCHITECTURE



HIVE DDL

DATABASES

It's also common to use databases to organize production tables into logical groups.

If you don't specify a database, the default database is used.

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

```
CREATE DATABASE ineuron_db;
```

Hive will throw an error if ineuron_db already exists.

You can suppress these warnings with this variation:

```
CREATE DATABASE IF NOT EXISTS ineuron_db;
```

```
SHOW DATABASES LIKE 'a.*';
```

HIVE DDL (CONTD.)

DATABASES

Tables in that database will be stored in subdirectories of the database directory.

The exception is tables in the default database, which doesn't have its own directory.

The database directory is created under a top-level directory specified by the property

hive.metastore.warehouse.dir

set hive.metastore.warehouse.dir;

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

```
CREATE DATABASE ineuron_db
LOCATION '/user/ineuron/mydb';
```

HIVE DDL (CONTD.)

DATABASES

DESCRIBE DATABASE ineuron_db;

The USE command sets a database as your working database, analogous to changing working directories in a filesystem.

USE ineuron_db;

set hive.cli.print.current.db=true;

DROP DATABASE IF EXISTS ineuron_db CASCADE;

HIVE DDL (CONTD.)

The IF EXISTS is optional and suppresses warnings if ineuron_db doesn't exist.

By default, Hive won't permit you to drop a database if it contains tables.

You can either drop the tables first or append the CASCADE keyword to the command,which will cause the Hive to drop the tables in the database first:

```
DROP DATABASE IF EXISTS ineuron_db CASCADE;
```

When a database is dropped, its directory is also dropped.

HIVE DDL (CONTD.)

TABLES

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 table if not exists emp_details
(
emp_name string,
unit string,
exp int,
location string
)
row format delimited
fields terminated by ',';
DROP TABLE IF EXISTS emp_details;
```

HIVE DDL (CONTD.)

TABLES

Most table properties can be altered with ALTER TABLE statements, which change metadata about the table but not the data itself.

Renaming a Table

Use this statement to rename the table emp_details to employee_details:

```
ALTER TABLE emp_details RENAME TO employee_details;
```

HIVE DDL (CONTD.)

Changing Columns

You can rename a column, change its position, type, or comment:

```
ALTER TABLE emp_details
CHANGE COLUMN emp_name emp_name STRING
COMMENT 'Employee Name'
 AFTER unit;
```

You have to specify the old name, a new name, and the type, even if the name or type is not changing.

HIVE VIEWS

A common use case for views is restricting the result rows based on the value of one or more columns.

When a query becomes long or complicated, a view may be used to hide the complexity by dividing the query into smaller, more manageable pieces; similar to writing a function in a programming language or the concept of layered design in software.

```
CREATE VIEW joined_view AS
SELECT * FROM people JOIN cart
ON (cart.people_id=people.id)
WHERE firstname='john';
```

HIVE VIEWS (CONTD.)

As part of Hive's query optimization, the clauses of both the query and view may be combined together into a single actual query.

The conceptual view still applies when the view and a query that uses it both contain an ORDER BY clause or a LIMIT clause.

The view's clauses are evaluated before the using query's clauses.

For example, if the view has a LIMIT 100 clause and the query has a LIMIT 200 clause, you'll get at most 100 results.

While defining a view doesn't "materialize" any data, the view is frozen to any subsequent changes to any tables and columns that the view uses.

Hence, a query using a view can fail if the referenced tables or columns no longer exist.

QUIZ

Q. When is it suggested to use a View?

1. When a query is composed of many inner and complex queries.
2. When we want to fetch specific records or columns from a table.
3. When we want to know table details along with query result.
4. When a table stores temporary data.

ANSWERS

Q.When is it suggested to use a View?

1. When a query is composed of many inner and complex queries.
2. When we want to fetch specific records or columns from a table.
3. When we want to know table details along with query result.
4. When a table stores temporary data.

PRIMARY DATA TYPES

Primary Data Types are further classified into four categories. They are:

1. Numeric Types
2. String Types
3. Date/Time Types
4. Miscellaneous Types

Numeric Data Types

Integral types are – TINYINT, SMALLINT, INT & BIGINT- they store integer values

Equivalent to Java's datatype i.e. byte , short , int , and long primitive types

Floating types are – FLOAT, DOUBLE & DECIMAL.

Equivalent to Java's float and double , and SQL's Decimal respectively.

DECIMAL(5,3) represents total of 5 digits, out of which 3 are decimal digits. E.g. 13.345

PRIMARY DATA TYPES (CONTD.)

PRIMITIVE NUMERIC DATATYPE

Type	Size	Range	Examples
TINYINT	1 Byte signed integer	-128 to 127	100
SMALLINT	2 Bytes signed integer	-32,768 to 32,767	100, 1000
INT	4 Bytes signed integer	-2,147,483,648 to 2,147,483,647	100, 1000, 50000
BIGINT	8-byte signed integer	$-9.2*10^{18}$ to $9.2*10^{18}$	100, $1000*10^{10}$
FLOAT	4-byte single precision float	$1.4*e^{-45}$ to $3.4*e^{+38}$	1500.00
DOUBLE	8-byte double precision float	$4.94e^{-324}$ to $1.79e^{+308}$	750000.00
DECIMAL	17 Bytes Precision upto 38 digits	$-10^{38}+1$ to $10^{38}-1$	DECIMAL(5,2)

PRIMARY DATA TYPES (CONTD.)

PRIMITIVESTRING DATATYPE

Type	Description	Examples
STRING	Sequence of characters. Either single quotes (') or double quotes (") can be used to enclose characters	'Welcome to Hadooptutorial.info'
VARCHAR	Max length is specified in braces. Similar to SQL's VARCHAR. Max length allowed is 65355 bytes	'Welcome to Hadooptutorial.info tutorials'
CHAR	Similar to SQL's CHAR with fixed-length. i.e values shorter than the specified length are padded with spaces	'Hadooptutorial.info'

PRIMARY DATA TYPES (CONTD.)

Date/Time Types

Hive provides DATE and TIMESTAMP data types in traditional (UNIX time stamp).

UNIX time stamp format for date/time related fields in hive.

DATE values are represented in the form YYYY-MM-DD. Example: DATE '2014-12-07'.

Date ranges allowed are 0000-01-01 to 9999-12-31.

TIMESTAMP use the format yyyy-mm-dd hh:mm:ss[.f...].

We can also cast the String, Time-stamp values to Date format if they match format.

PRIMARY DATA TYPES (CONTD.)

Miscellaneous Types

Hive supports two more primitive data types, BOOLEAN and BINARY. Similar to Java's Boolean, BOOLEAN in hive stores true or false values only.

BINARY is an array of Bytes and similar to VARBINARY in many RDBMSs