1. What is hive?

* Hive is basically a data warehousing packaging built on the top of hadoop.
* Used for managing and querying large structured data sets.
* It uses sql kinda of language to achieve this called HIVEQL.

2. What is data warehouse?

A large store of data accumulated from a wide range of sources within a company and used to guide management decisions.

Inmon described a data warehouse as being a subject-oriented, integrated, time-variant and [nonvolatile](http://searchstorage.techtarget.com/definition/nonvolatile-storage) collection of data that supports management's decision-making process.

**Subject-Oriented**: A data warehouse can be used to analyze a particular subject area. For example, "sales" can be a particular subject.

**Integrated**: A data warehouse integrates data from multiple data sources. For example, source A and source B may have different ways of identifying a product, but in a data warehouse, there will be only a single way of identifying a product.

**Time-Variant**: Historical data is kept in a data warehouse. For example, one can retrieve data from 3 months, 6 months, 12 months, or even older data from a data warehouse. This contrasts with a transactions system, where often only the most recent data is kept. For example, a transaction system may hold the most recent address of a customer, where a data warehouse can hold all addresses associated with a customer.

**Non-volatile**: Once data is in the data warehouse, it will not change. So, historical data in a data warehouse should never be altered.

3. Where Hive is best suitable?

* When you are doing data warehousing applications where most probably you

are receiving static data instead of dynamic data.

* When the application is on high latency(i.e where query response time is high)
* Where large data sets are maintained and mined for insights and reports.

4. What is static and dynamic data?

Static data is that, memory is fixed in size.This results in the maximum size needing to be known in advance, as memory cannot be reallocated at a later point

A key advantage of static data structures is that with [memory allocation](http://www.webopedia.com/TERM/A/allocated_memory.html) fixed, no control or oversight is needed to prevent potential overflow or underflow issues when adding new items or removing existing ones. This makes static data structures easier to program but at the expense of potential efficiency in terms of memory consumption.

Dynamic data - memory that has the flexibility to grow or shrink in size, enabling a programmer to control exactly how much memory is utilized.

Static data structures are ideal for storing a fixed number of data items, but they lack the dynamic data structure’s flexibility to consume additional memory if needed or free up memory when possible for improved efficiency.

5. Where Hive is not used?

* Where it doesnt supports OLTP and supports only OLAP

OLAP - Online Analytical Processing.

OLTP - online transaction processings

* If application requires OLTP -then switch to NoSql databases.
* HQL queries have higher latency due to mapreduce jobs.

6. What are OLTP and OLAP?

OLTP: OLTP is characterized by a large number of short on-line transactions **(INSERT, UPDATE, DELETE).**

* The main emphasis for OLTP systems is put on **very fast query processing**, maintaining data integrity in multi-access environments and an **effectiveness measured by number of transactions per second.**

OLAP:

* deals with Historical Data or Archival Data
* Queries are often very complex and involve **aggregations**.
* OLAP applications are widely used by Data Mining techniques
* Sometime query need to access large amount of data in Management records like what was the **profit of your company in last year.**

Let's consider two examples scenarios :

***Scenario 1 :***

You are building an online store/website, and you want to be able to :

* store user data, passwords, previous transactions...
* store actual products, their associated price

You want to be able to find data for a particular user, change it's name... Basically perform INSERT, UPDATE, DELETE operations on a user data. Same with products, etc.

You want to be able to make transactions, possibly involving a user buying a product (that's a relation). Then OLTP is probably a good fit (think SQL databases).

***Scenario 2 :***

You have an online store/website, and you want to compute things like

* the "total money spend for all users"
* "what is the most sold product"

This falls into the analytics/business intelligence domain, therefore OLAP is probably more suited.

* **OLTP databases are meant to be used to do many small transactions, and usually serve as a "single source of truth".**
* **OLAP databases on the other hand are more suited for analytics, data mining, less queries but they are usually bigger (they operate on more data).**

Best component for OLAP ------ hive

Best component for OLTP ------- NoSql, cassandra??

**7. Differences between Sort by and order by, Distribute by and cluster by.**

**Order By:**

Uses single reducer to guarantee total sorted order in output. If we are in strict mode then **LIMIT** has to be used to minimize sort time.

It performs a total ordering of the query result set. This means that all the data is passed through a single reducer, which may take an unacceptably long time to execute for larger data sets.

Hive adds an alternative, SORT BY that orders the data only within each reducer, thereby performing a local ordering, where each reducer’s output will be sorted. Better performance is traded for total ordering.

In both cases, the syntax differs only by the use of the ORDER or SORT keyword. You can specify any columns you wish and specify whether or not the columns are ascending using the ASC keyword (the default) or descending using the DESC keyword.

**Distribute By:**

Ensures each of N reducers gets non-overlapping ranges of x, but doesn't sort the output of each reducer. You end up with N or unsorted files with non-overlapping ranges.

**Cluster BY:**

Ensures each of N reducers gets non-overlapping ranges then sorts by those ranges at the reducers. This gives you global ordering, and is the same as doing (DISTRIBUTE BY x and SORT BY x). You end up with N or more sorted files with non-overlapping ranges.

So **CLUSTER BY** is basically the more scalable version of **ORDER BY**.

NOTE: Suppose to mention how many reducers we need :

**SET mapred.reduce.tasks = no of reducers**

**8. When to use sort by instead of order by?**

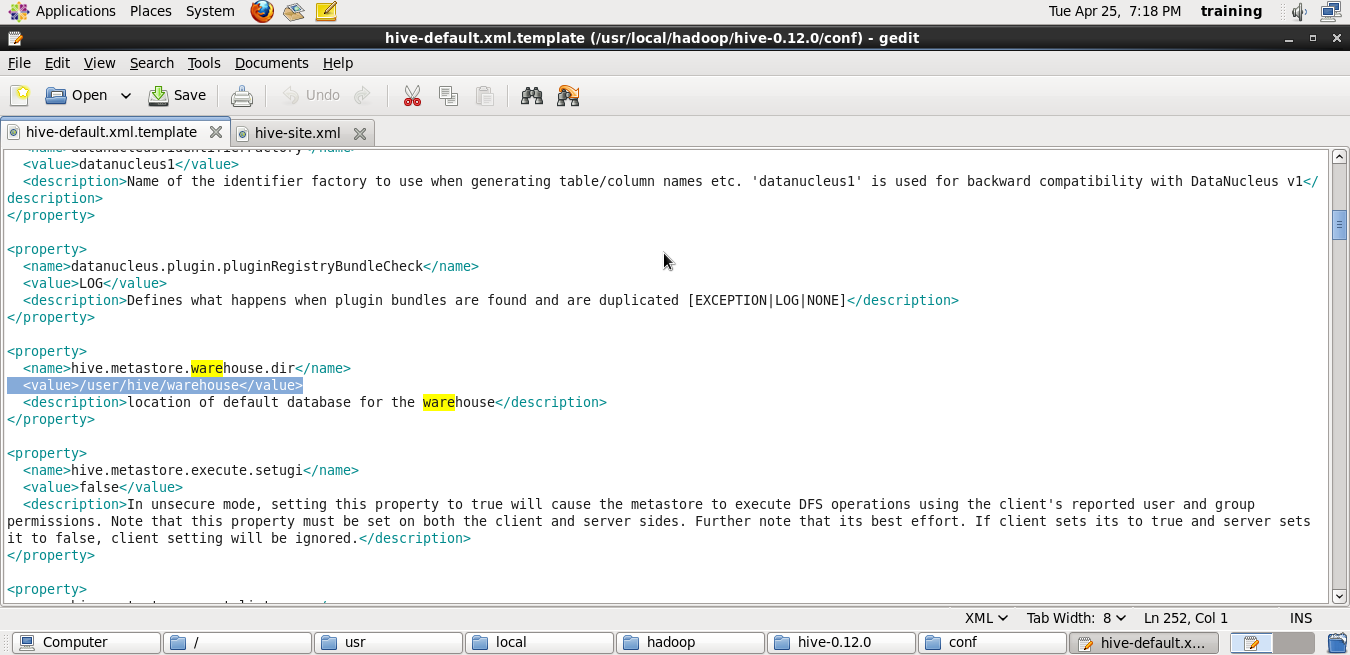
Sort by clause is used to sort very huge data sets as the sort by clause sorts the data using multiple reducers.

But order by sorts all the data together using a single reducer.

Sort by in hive is same as secondary sort in Map reduce.

**9. Where does the data of a Hive table gets stored?**

* By default hive table is stored in HDFS path : /user/hive/warehouse.
* It is specified in **hive.metastore.warehouse.dir** configuration paraMETER PRESENT IN HIVE-SITE.XML



9**. Why HDS is not used to store Hive meta store? I.e Why metasore is stored in RDBMS rather than storing in HDFS?**

* Editing or CRUD operations for files / data which is present in HDFS is not allowed.
* Metastore stores metadata in RDBMS to provide low query latency.
* HDFS read and write operations are time consuming.

10. Differences between internal and external table.

12. Difference between Partition and Buckets.

13. Consider a scenario:

Consider a table - transaction with columns like ====== customer\_id, total\_amount, month, country.

Lets insert 50000 tuples into this tables. So I want to know the total revenue generated for each month. But hive is taking too much time to process it. So how to solve this problem.

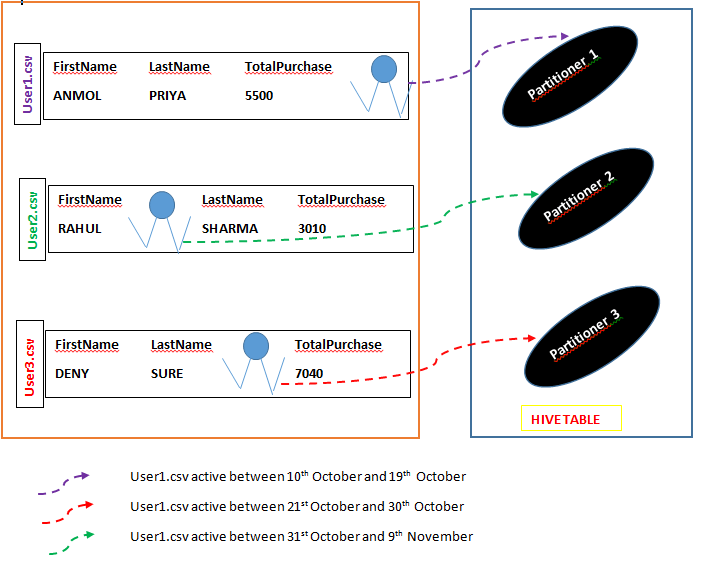
* Create partition table-------- partition\_transaction for month.
* Query ------select sum(amount) from partition\_transaction where month = January;

14. What is partitioning and when it is used?

A scenario for partitioning:

Let’s take a scenario:

* Data is present in hdfs coming in from various ecommerce companies.
* We need to run the HiveQl queries on user buying pattern.
* We need to analyse data coming in from last 10 days.

In the above scenario instead of running the queries which involves scanning of entire table, an approach should be followed where query runs on only last 10 days of data.

15. What is dynamic partition and when it is used?

* Values for partitioning are known during runtime.
* It is used when loading data from an existing non partitioned table to improve query latency.
* And when the values of partition are not known before and hence finding these unknown partition values manually from huge data sets is a tedious task.

16. How Hive distributes rows into bucketing?

With hasing function

Hash)\_Function ( of bucketed column ) **MOD**  ( number of buckets)

17. Consider, I have a csv file present in /temp

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Firts name | Last name | mail | gender | IP |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |

So how will you consume this csv file into Hive warehouse using built in serde?

A SerDe allows us to convert unstructured bytes into a record that we can process using HIVE.

**ROW FORMAT SERDE ‘ORG.APACHE.HADOOP.HIVE.SERDE2.OpenCSVSerDe’**

STORES AS TEXTFILE LOCATION ‘/TEMP’

18. Scenario:

I have a lot small files. Data is of the format {id, name, email, country}

Now as we know hadoop performance will decrease in case of small files. So how to overcome in hive?

Use sequence files.

19. **What is the difference between local and remote metastore?**

*Local Metastore:*

In local metastore configuration, the metastore service runs in the same JVM in which the Hive service is running and connects to a database running in a separate JVM, either on the same machine or on a remote machine.

*Remote Metastore:*

In the remote metastore configuration, the metastore service runs on its own separate JVM and not in the Hive service JVM. Other processes communicate with the metastore server using Thrift Network APIs. You can have one or more metastore servers in this case to provide more availability.

20. **8. Scenario:**

*Suppose I have installed Apache Hive on top of my Hadoop cluster using default metastore configuration. Then, what will happen if we have multiple clients trying to access Hive at the same time?*

The default metastore configuration allows only one Hive session to be opened at a time for accessing the metastore. Therefore, if multiple clients try to access the metastore at the same time, they will get an error. One has to use a standalone metastore, i.e. Local or remote metastore configuration in Apache Hive for allowing access to multiple clients concurrently.

Following are the steps to configure MySQL database as the local metastore in Apache Hive:

* One should make the following changes in hive-site.xml:
  + *javax.jdo.option.ConnectionURL* property should be set to jdbc:*mysql*:*//host/*dbname?createDataba
  + *seIfNotExist=true.*
  + *javax.jdo.option.ConnectionDriverName* property should be set to *com.mysql.jdbc.Driver.*
  + One should also set the username and password as:
    - javax.jdo.option.ConnectionUserName is set to desired username.
    - javax.jdo.option.ConnectionPassword is set to the desired password.
* The JDBC driver JAR file for MySQL must be on the Hive’s classpath, i.e. The jar file should be copied into the Hive’s lib directory.
* Now, after restarting the Hive shell, it will automatically connect to the MySQL database which is running as a standalone metastore.

21. When we do select \* from table; in hive why doesnt invoke HDFS?

22. How Hive reads and writes records?

This is achieved by hive-SerDe using serialize and deserialize methods.

23. What is SerDe?

24. Can we use hive for unstructured data?

**25. What are the ways to achieve optimization in hive?**

1. Partitioning and bucketing
2. Applying filtering and denormalization to data before joining.
3. Use ORC file….Using ORC files for every hive tables is extremely beneficial to get faster response times for our hive queries.
4. Use Tez ------ A processing engine.
5. Use vectorization.:

Normally hive rad data row by row, so it takes a lot of time and is less efficient.

Vectorization means reading data in a batch consisting of around 1000 rows rather than reading one by one.

Vectorized query execution improves performance of operations like scans, aggregations, filters and joins, by performing them in batches of 1024 rows at once instead of single row each time.

26. What is Beeline in hive?

<http://blog.cloudera.com/blog/2014/02/migrating-from-hive-cli-to-beeline-a-primer/>

Apache Hive was a heavyweight command-line tool that accepted queries and executed them utilizing MapReduce.

* **HiveServer1** is the server (responsible for compiling and monitoring MapReduce jobs.
* **Hive CLI** is the command-line interface (sends SQL to the server).

Recently Hive community [introduced HiveServer2](http://blog.cloudera.com/blog/2013/07/how-hiveserver2-brings-security-and-concurrency-to-apache-hive/), an enhanced Hive server designed for multi-client concurrency and improved authentication that also provides better support for clients connecting through JDBC and ODBC

Now HiveServer2, with Beeline as the command-line interface, is the recommended solution.

Beeline was developed specifically to interact with the new server. Unlike **Hive CLI, which is an Apache Thrift-based client**,

Beeline is a JDBC client based on the [SQLLine](http://sqlline.sourceforge.net/) CLI — although the JDBC driver used communicates with HiveServer2 using HiveServer2’s Thrift APIs.

## **Use Cases: Hive CLI versus Beeline**

**1. Server Connection**

Hive CLI connects to a remote HiveServer1 instance using the Thrift protocol. To connect to a server, you specify the host name and optionally the port number of the remote server:

> hive -h <hostname> -p

<port>

In contrast, Beeline connects to a remote HiveServer2 instance using JDBC. Thus, the connection parameter is a JDBC URL that’s common in JDBC-based clients:

> beeline -u <url> -n <username> -p

<password>

jdbc:hive2://ubuntu:11000/db2?hive.cli.conf.printheader=true;hive.exec.mode.local.auto.inputbytes.max=9999#stab=salesTable;icol=customerID

jdbc:hive2://?hive.cli.conf.printheader=true;hive.exec.mode.local.auto.inputbytes.max=9999#stab=s