BDA - 2018

Q.1) What is NoSQL?

- NoSQL DB, also called Not Only SQL, is an approach to data management & database design that's useful for very large sets of distributed data.
- NoSQL is not relational DB.
- It's non-relational DBMS, that doesn't req fixed schema.
- The major purpose of of using NoSQL DB is for distributed data stores with humongous data storage needs.
- NoSQL is used for Big data & real time web apps.
  Eg: Twitter, Facebook & Google, Netflix, Linkenin collects TB's of user data every single day.
- NoSQL DB system encompasses that can store structured, semi-structured, unstructred & polymorphic data.

Advantages:

① NoSQL is Non-relational

Non-relational, in other words, you can call it as table-less, these NoSQL DB vary from SQL DB.
- They provide ease of management while ensuring high level of flexibility with data models that are new.

② NoSQL is Low Cost

- While being low cost, NoSQL is also an open-source DB, that provides awesome soluth for smaller enterprise to opt this at affordable price.
- Various kind of NoSQL DB available in market include Couchbase, Amazon's Dynamo DB, Mongo DB to provide for processing big data app that are cost-effective.

③ Scalability is Easier

- NoSQL has been gaining popularity because of elasticity & scalability that it offers over other kind of DB that are available.
- It's been designed to perform exceptionally well under any condition including low cost hlw.
- You can easily create DB without actually developing any detailed DB models when using NoSQL DB. This will save lot of time & effort.

④ To store large data:

- Organizatn are finding valuable to capture more data & process it more quickly.
- They find it expensive to do with ~~RDB~~ Relational DB, because it req more cost to run large data & computing loads on clusters.
- Many NoSQL DB are designed explicitly to run on clusters, so they make a better fit for big data scenarios.

⑤ Analytics:

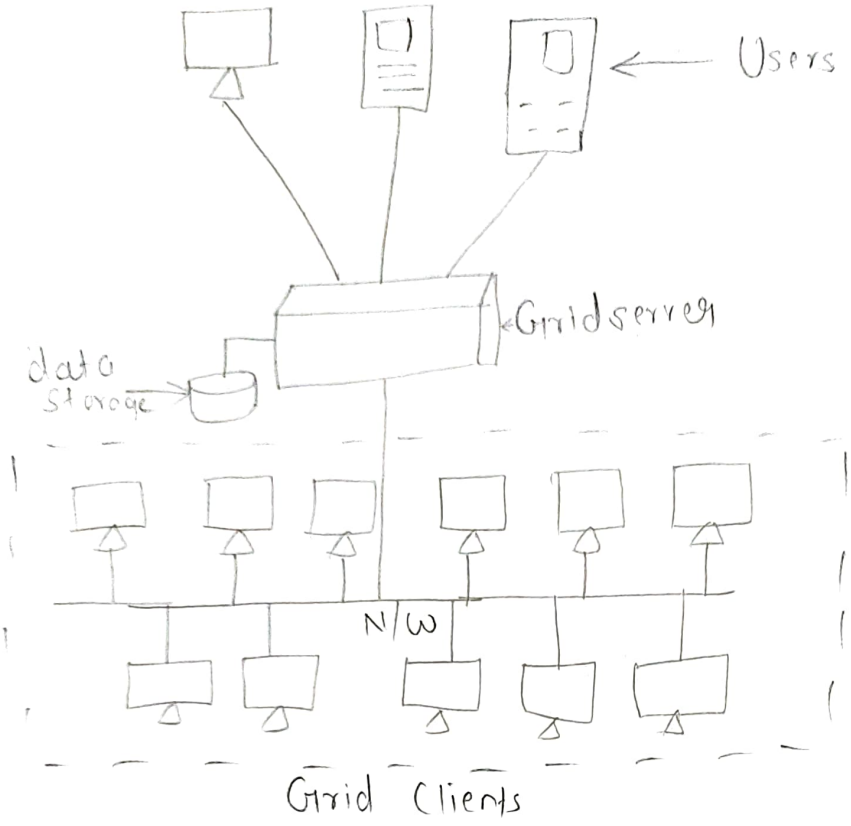
One of reason to consider adding a NoSQL DB to your corporate infraiture is that many NoSQL DB are well suited to performing analytical queries.

(0.2) Note:

(i) Grid Computing:

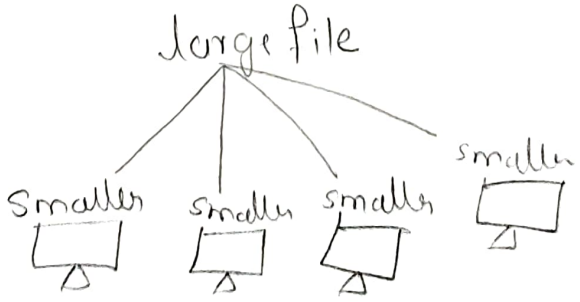
- Grid computing is the collectn of computer resources from multiple locatn to reach common goal.
- Grid canbe thought of distributed system with non-interactive workloads that involve large number of files
- A computational grid is a Hlw & slw infra that provide dependable, consistent & inexpensive access to high end computational capabilities
- A grid links together computing resources (PC's, workstatn) servers, storage) & provide mechanism to access those resources.

How Grid computing works?



- In general grid computing system requires:
- ① At least one computer, usually a server, which handles all the administrative ~~facilities~~ in the system.
- ② A ~~special~~ n/w of computers running special grid computing n/w s/w.
- ③ A collectn of computer s/w called middleware

eg:



- large file will be broken down into smaller pieces & distributed throughout the computing resources. for execution
- This process is executed in parallel as their is no interactive workload betn each system. As a result final o/p is achieved in smaller amount of time.

Volunteer Computing:

- Volunteer computing is a type of distributed computing in which people donate their computers unused resources to research-oriented project.
- Thus it's establishment of technology that enables ordinary citizens donate their computing resources to one or more "projects".

VC includes:

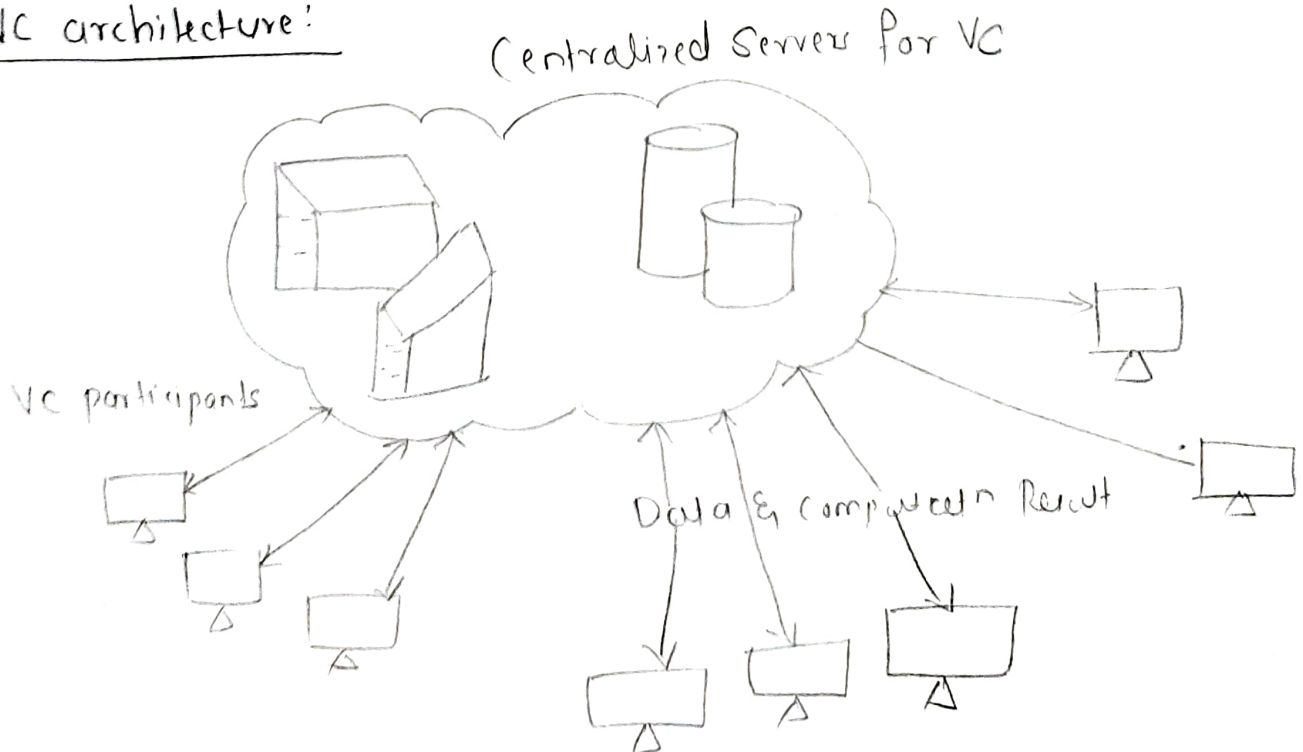
① Group of computing resources
② Voluntary share the resource
③ Trust betn peers
④ Incentive volunteers.

Eg: ① SETI@home ② storage@home ③ BONIC

- BONIC is widely used middle-ware system.
- a client prog run on the volunteer's computer.

However, data is centrally stored

VC architecture:



- It's very CPU intensive, which make it suitable for running on hundreds of thousands of computers across the world.
- Volunteers here are donating CPU cycles not bandwidth.
- It runs a perpetual computatn on untrusted machines on Internet with highly variable connectn speeds & no data locality.

Q4) Apache Cassandra

- ~~Apache~~ Cassandra data store is an open source Apache project available at http://cassandra.apache.org
- Cassandra originated at Facebook in 2007 to solve that company's inbox search problem
- Originally designed to handle large volume of data in a way that was difficult to scale with traditional methods

Decentralized:
- Every node in cluster has same role. There is no single point of failure.
- Data is distributed across the cluster, but there is no master as every node can service any request

Supports replicatn & multi data center replicatn:
- Cassandra is designed as a distributed system, for deployment of large no. of nodes across multiple data centers.

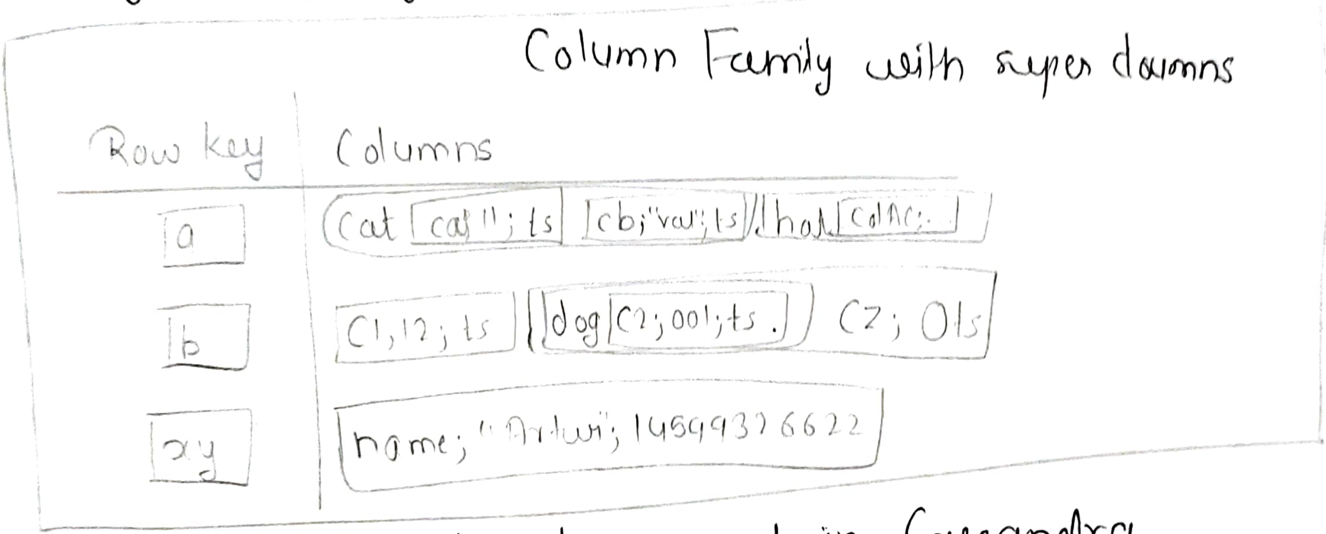
Scalability:
Read & write throughput both increase linearly as new machines are added, with no downtime (or) interuptn to apps.

Fault tolerant
- Data is automatically replicated to multiple nodes for fault tolerance.
- Replicatn across multiple data centers is supported.
- Failed nodes can be replaced with no downtime.

Map Reduce support:

- Cassandra has Hadoop integratn, with MapReduce Support.
- There is support also for Apache Pig & Apache Hive.

- Cassandra is a hybrid betn a key-value & Column-oriented NoSQL DB.
- key value nature is represented by row obj, in which value would be generally organized in columns.



Following are some key terms used in Cassandra
- Keyspace can be seen as DB Schema in SQL.
- Column family resembles a table in SQL world.
- Row has a key & as a value set of Cassandra column.
- Column is a triplet consist of (name, value, timestamp)
- DataTypes: Validators & Comparators.

Q.12a) What is Apache Pig?

- Pig is a simple to understand data flow lang used in the analysis of large data sets.
- Pig scripts are automatically converted into MapReduce jobs by Pig interpreter, so you can analyze data in Hadoop cluster even if you aren't familiar with MapReduce.

- Pig is used to
  - proces web log
  - Build user behaviour models
  - Proces images
  - Data minning
- Pig is made up of 2 components:
  - 1st is language itself which is called as PigLatin
  - 2nd is runtime environment where PigLatin prog are executed
- Pig execut^n environment has 2 modes.

① Local mode: All scripts are run on single machine, Hadoop MapReduce & HDFS are no required.

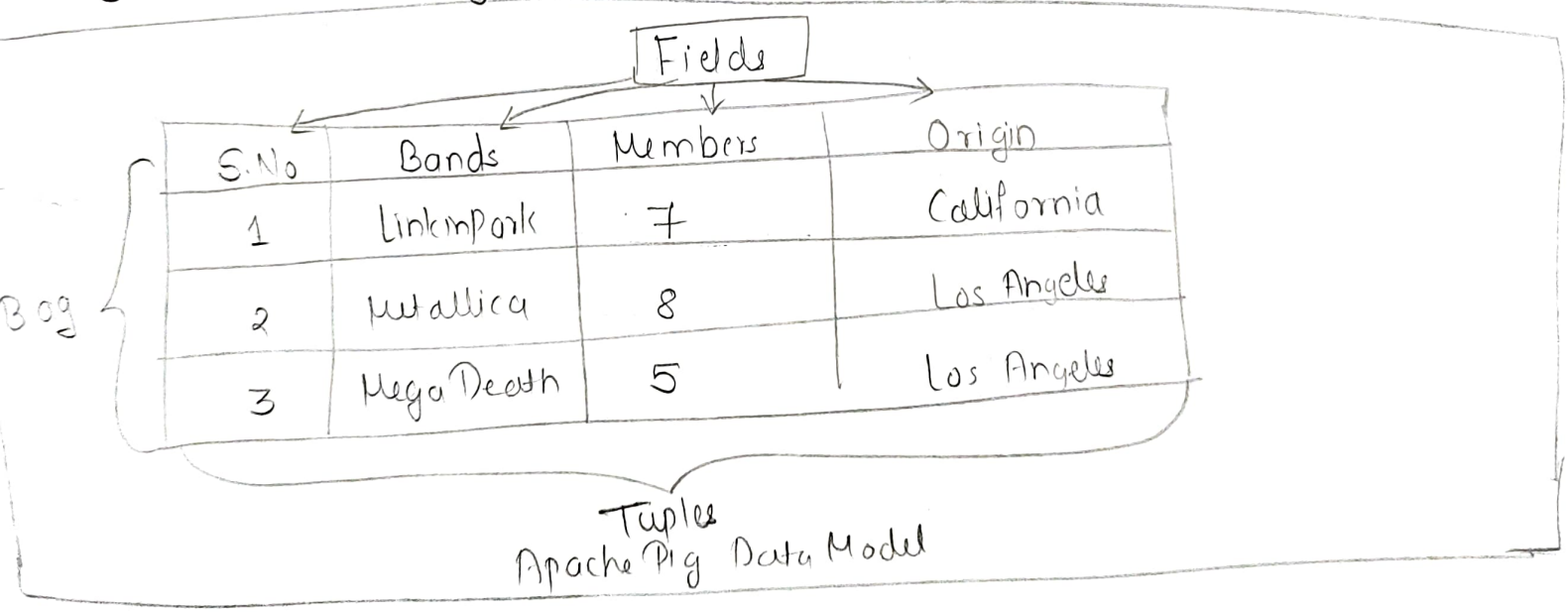
② Hadoop: Also called MapReduce mode, all scripts are run on a given Hadoop cluster.

Apache Pig	MapReduce
① Apache Pig is a data flow lang	① It's a data processing paradigm.
② It's a high level lang	② It's low level lang
③ Performing join operat^n in Apache Pig is pretty simple	③ It's quite difficult in MapReduce to perform join operat^n bet^n datasets.
④ Any novice programmer with basic knowledge of SQL can work conveinetly on Apache pig.	④ Exposure to Java must to work with MapReduce

⑤ Apache pig use multi-query approach thus reducing length of code to greater extent.	- It will require almost 20 times more no. of lines to perform same task.
⑥ Pig Scripts internally convert into MapReduce prog & get executed.	- MapReduce prog is compiled & executed directly.
⑦ Writing & executing Pig script is simple task compared to MapReduce	- Writing & executing MapReduce prog is bit complex task.

Q. 12b) Pig Data Model

- The data model of Pig Latin is fully nested & it allows complex non-atomic datatypes such as map & tuple.
- Given below is diagrammatical representation of Pig Latin's data model.



Apache Pig Data Model

Atom:

- Any single value in Pig Latin, irrespective of their data, type is known as Atom. It's stored as string & can be used as string & number.
- int, float, double, chararray are atomic values of Pig.
- A piece of data or simple atomic value is known as field.
  Eg: 'Linkin Park' or '7'

Tuple:

- A record that is formed by an ordered set of field is known as a tuple, fields can be of any type.

- A tuple is similar to a row in table of RDBMS

Eg: Metallica, 8, Los Angles

Bag: A bag is an unordered set of tuples. In other words, a collectn of tuples (non-unique) is known as a bag.

- A bag is represented by '{}'.
- Unlike table in RDBMS, it's not necessary that every tuple contain same no. of fields or that fields in same posith (column) have same type.

Eg: {(Linkin Park, 7), (Metallica, 8)}

Map: A map (or data map) is a set of key-value pairs.

- The key needs to be type chararray & should be unique.

The value might be of any type. It's represented by '[]'

Eg: [Bands # Linkin Park, Members # 7]

2019 Q.12 a)

HBASE	HDFS
① HBase provides low latency access to small operatn of data within large datasets	① HDFS provides high latency operatn.
② It supports read & write	② It supports WORM (Write once Read Many or Multiple times)
③ HBase is accessed through shell commands, Java, API, REST	③ HDFS is accessed through MapReduce jobs.
④ HBASE is Java based Not only SQL DB.	④ HDFS is Java-based file system utilized for storing large data sets
⑤ HBase is partially tolerant & highly consistent	⑤ HDFS is highly fault-tolerant & cost effective.
⑥ HBase is dynamic changes	⑥ It's rigid architecture
⑦ HBase is preferable for real time processing	⑦ HDFS is preferable for offline batch processing

2019 Q.12 b)

Cassandra	RDBMS
① Cassandra is used to deal with unstructred data	① RDBMS is used to deal with structured data
② It's flexible schema	② It's has fixed data
③ It has, a table is a list of "nested key-value pairs"	③ RDBMS, a table is an array of arrays
④ keyspace is outer most container which contain data corresponding to an app	④ RDBMS, dB is outer most container which contain data corresponding to an app.
⑤ In Cassandra, tables or column families are entity of keyspace	⑤ RDBMS, tables are entities of DB.
⑥ Row is unit of replicatn	⑥ Row is an individual record
⑦ Column is a unit of storage	⑦ Column represents attributes of a relatn
⑧ Relatnship are represented using collectn	⑧ There is concept of foreign key joins etc.

Q.13 2018 b)

b) Hive QL

- HiveQL is Hive query language
- Hadoop is an open source framework for the distributed processing of large amount of data across a cluster.
- It relies upon MapReduce paradigm to reduce complex tasks into smaller parallel tasks that can be executed concurrently across multiple machines.
- However, writing MapReduce tasks on top of Hadoop for processing data is not for everyone since it requires learning new framework

- What is needed is an easy-to-use abstractn on top of Hadoop that allows ppl not familiar with it to use its capabilities as easily.
- Hive aims to solve this problem by offering an SQL-like interface, called HiveSQL, on top of Hadoop.
- Hive achieves this task by converting queries written in HiveQL into MapReduce tasks that are then run across Hadoop cluster to fetch desired results.
- It's best suited for batch processing large amount of data (such as data warehousing) but is not ideally suitable as a routine transactional DB because of its slow response times.
- A common task for which Hive is used is processing of logs of web servers.
- Hive query lang (HiveQL) supports SQL features like CREATE tables, DROP tables, SELECT... FROM... WHERE clause, Joins (inner, left outer, right outer & outer joins), Cartesian products, GROUP BY, SORT BY, aggregatn, union & many useful fundn on primitive as well as complex data types.
- Metadata browsing features such as list DB, tables & so on are also provided.
- HiveQL does have limitatn compared with tradional RDBMSSQL.
- HiveQL allow creatn of new tables in accordance with partitions as well as buckets & allow insertn of data in single (or) multiple tables but doesn't allow deletn (or) updating of data.

HiveQL: Data definitn (2017 Q.7)

1. 1st open the hive console by typing:
   $ hive
2. Once the hive console is opened, like
   hive >
   you need to run the query to create the table
3. Create & Show DB

They are very useful for larger clusters with multiple teams
users, as a way of avoiding table name collisions. It's also comm
to use DB to organize producth tables into logical groups.
If you don't specify a DB, default DB is used.

```
hive > CREATE DB IF NOT EXISTS FINANCIALS;
```

At any time, you can see DB that already exist as follow:

```
hive> SHOW DBASES;
```

o/p is:

```
default
financials
hive> CREATE DATABASE human_resources;
hive> SHOW DATABASES;
```

o/p is:

```
default
financials
human_resources
```

4. DESCRIBE database
   _ shows the directory locatn for the database.

```
hive > DESCRIBE DATABASE financials;
```

o/p is: hdfs://master-server/user/hive/warehouse/financials.db

HiveQL: Data definitn (2017 Q.7)

1. 1st open the hive console by typing:
   $ hive

2. Once the hive console is opened, like
   hive >
   you need to run the query to create the table

3. Create & Show DB

They are very useful for larger clusters with multiple teams & users, as a way of avoiding table name collisions. It's also common to use DB to organize producth tables into logical groups.
If you don't specify a DB, default DB is used.

```
hive > CREATE DB IF NOT EXISTS FINANCIALS;
```

At any time, you can see DB that already exist as follow:

```
hive> SHOW DBASES;
```

o/p is:

```
default
financials
hive> CREATE DATABASE human_resources;
hive> SHOW DATABASES;
```

o/p is:

```
default
financials
human_resources
```

4. DESCRIBE database
   - shows the directory locatn for the database.

```
hive > DESCRIBE DATABASE financials;
```

o/p is: hdfs://master-server/user/hive/warehouse/financials.db

5. Use Database

You can Use command sets DB as your working db, analogous to changing working directories in a filesystem

hive> USE financials;

6. DROP Database – you can drop a database:

hive> DROP DATABASE IF EXISTS financials;

The IF EXISTS is optional & suppresses warning if financials doesn't exist.

7. ALTER Database

You can set key-value pair in DBPROPERTIES associated with a db using ALTER DATABASE command. No other metadata about db can be changed, including it's name & directory location:

hive> ALTER DATABASE financials SET DBPROPERTIES('edited-by=
'active steps');

8. Create Tables

The CREATE TABLE statement follows SQL conventions, but Hive's version offers significant extensions to support a wide range of flexibility where data files for table are stored, the format used etc.

•) Managed Tables

- The tables we have created so far are called managed tables (or) sometimes called internal tables, because Hive controls lifecycle of their data.
- As we've seen, Hive stores data for these tables in subdirectory under directory defined by hive.metastore.warehouse.dir (eg: /user/hive/warehouse) by default.
- When we drop a managed table, Hive ~~db~~ delete data in the table
- Managed tables are less convenient for sharing with other tools.

•) External Tables

```
CREATE EXTERNAL TABLE NOT EXISTS stocks (
    exchange STRING,
    symbol STRING,
    ymd STRING,
    price_open FLOAT,
    price_high FLOAT,
    price_low FLOAT,
    volume INT,
    price_adj_close FLOAT
)
ROW FORMAT DELIMITED FIELDS TERMINATED ','
LOCATION '/data/stocks/';
```

The EXTERNAL keywords tells Hive this table is external & the LOCATION clause is required to tell Hive where it's located. Because it's external

Dropping Tables:

The familiar DROP TABLE command from SQL is supported:

```
DROP TABLE IF EXISTS financials;
```

Q. 14 2019 Avro

- Avro is a language-neutral data serializatn system.
- It can be processed by many lang (currently C, C++, C#, Java, Python & Ruby)
- Avro creates binary structured format that is both compressible & splittable. Hence it can be efficiently used as i/p to Hadoop MapReduce jobs.
- Avro provides rich data structures. For eg: you can create a record that contains an array, an enumerated type & a sub record.
- These datatypes can be created in any lang, can be processed in Hadoop & results can be fed to a third lang.

- Avro schemas defined in JSON, facilitate implementatn in the lang that already have JSON libraries.
- Avro creates a self-describing file named Avro Data File, in which it stores data along with its schema in the metadata section.
- Avro is used in Remote Procedure Calls (RPCs). During RPC, client & server exchange schema in the connectn handshake.

Avro work flow

Step: 1 - Create schemas. Here you need to design Avro schema according to your data.

Step: 2 - Read the schema into your prog. It's done in two ways-

- By Generating a class Corresponding to schema - Compile the schema using Avro. This generatn a class file corresponding to schema.
- By using Parsers Library - You can directly read the schema using parsers library.

Step 3 - Serialize the data using serializatn API provided for Avro, which is found in package org.apache.avro.specific.

Step: 4 - Deserialize data using deserilizatn API provided for Avro, which is found in package org.apache.avro.specific.

Serializatn

Serializatn is req to be -

- Compact - To make best use of nlw bandwidth, which is most scarce resource in a data center.
- Fast - Since communicatn between the nodes is crucial in distributed systems, serializatn & deserializatn proces should be quick, producing less overhead
- Extensible - Protocols change over time to meet new req, so it should be straightforward to evolve protocol in a controlled manner for clients & servers.

•) Interoperable - The message format should support the nodes that are written in different languages.

Serlizatn definition:

Serlizatn is the proces of converting structured objects into byte stream
It's done basically for 2 purpose:
① For tramission over a n/w (interprocess communicatn)
② And for writing to persisitent storage.

- In Hadoop, the Interprocess communicatn betn nodes in the system is done by using remote procedure callsie RPC's.
- RPC protocol uses serlizatn to make message into binary stream to be sent to remote node, which receives & deserializes binary stream into orignal message.

RPC serlizatn format is expected to be:
① Compact ② Fast ③ Extensible ④ Interoperatable.

Persisten storage

It's a digital storage facility that doesn't lose it's data with loss of power supply. Files, folders, db are example of persistent storage

Writable Interface

This interface in Hadoop which provides methods for serlizatn & deserlizatn. The following table describe the methods -

SNo.	Methods & Descriptn
1.	void readFields(DataInput in) This method is used to declize the fields of given object.
2.	void write(DataOutput out) This method is used to serialize fields of given object.

Writable Comparable Interface

- It's the combinatn of Writable & Comparable interfaces. This interface inherits Writable interface of Hadoop as well as Comparable interface of Java. Thus it provides methods for data serializatn, deserializatn & comparision.

SNo.	Method & Descriptn
1.	int compareTo(class obj) This method compares current obj with given object obj.