CHAPTER 6

UNIT VI

Advanced Analytics - Technology and Tools

Syllabus Topics

Analytics for unstructured data - Use cases, MapReduce, Apache Hadoop, The Hadoop Ecosystem - Pig, HIVE, HBase, Mahout, NoSQL. An Analytics Project - Communicating, operationalizing, creating final deliverables.

Syllabus Topic : Analytics for Unstructured Data

6.1 Analytics for Unstructured Data

Q. 6.1.1 Write note on analytics for unstructured data. (Refer section 6.1) **(4 Marks)**

- Before proceeding to data analysis, it is necessary to collect and process the data to extract the useful information.
- The extent of initial processing as well as data preparation is based on the size of data, and the extent of straightforwardness to recognize the structure of the data.
- There are four types of data structures :

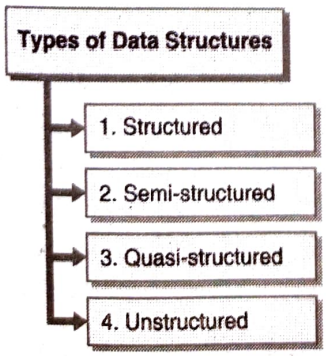

Fig. 6.1.1 : Types of Data Structures

➜ **1. Structured**

A precise and steady format (e.g. a data table)

➜ **2. Semi-structured**

A format which self-descries itself (e.g. an XML file)

➜ **3. Quasi-structured**

A little bit inconsistent format (e.g. a hyperlink)

➜ **4. Unstructured**

An inconsistent format (e.g. text or video)

- For interpretation purpose, the easiest format is structured data like RDBMS (relational database management system) tables.
- On the other hand, practically it is still important to understand the different values that may present in a specific column and what representations of these values in different situations (depends, e.g. on the values of other columns for the similar record).
- It is also possible that some columns have some unstructured text or stored objects, like images or videos.
- Tools discussed in this chapter focus on unstructured data.

Syllabus Topic : Use Cases

6.1.1 Use Cases

Q. 6.1.2 Explain use cases in analytics of unstructured data. (Refer section 6.1.1) **(4 Marks)**

- There are various use cases for MapReduce as discussed in the following content.
- The MapReduce paradigm provides a way to divide a large task into smaller ones, run those tasks simultaneously, and merge the results of the individual tasks into single final output.
- Apache Hadoop provides a software implementation for MapReduce.

☞ IBM Watson

- In 2011, IBM's computer system Watson takes part in a television game show of USA known as Jeopardy opposite to two best champions in the history of Jeopardy show.
- In the show, the contestants were given a clue like "He likes his martinis shaken, not stirred" and the correct response, phrased in the form of a question, would be, "Who is James Bond?"
- In the 3 day tournament, human contestants were defeated by the Watson.
- For the purpose of educating Watson, use of Hadoop is preferred to process several data sources like encyclopedias, dictionaries, news wire feeds, literature, as well as the entire contents of Wikipedia.
- In the game show, for each clue provided, Watson had responsibility to carry out following tasks in less than three seconds :
  - Deconstruct the clue given by game show into words and phrases
  - Built grammatical relationship between the words and the phrases
  - Generate a group of similar terms to refer in Watson's search for a response
  - Take help of Hadoop to manage the search for a response across terabytes of data
  - Conclude probable responses and allocate their likelihood of being correct
  - Actuate the buzzer
  - Provide a syntactically correct response in English
- Among other applications, is the use of Watson become recommended option in medical profession for the purpose of diagnosing patients and provides treatment recommendations.

☞ LinkedIn

- LinkedIn is a business and employment-oriented service of more than 250 million users in near about 200 countries.
- LinkedIn offers various free as well as subscription - based services, like company information pages, job postings, talent searches, social graphs of one's contacts, personally tailored news feeds, and access to discussion groups, including a Hadoop users group.
- In LinkedIn the use of Hadoop is for following purposes :
  - To process production database transaction logs on daily basis.
  - Keep the track of users' several activities like views and clicks.
  - Feed the extracted data back to the production systems.
  - Restructure the data for the purpose of adding it to an analytical database.
  - Develop as well as test analytical models.

☞ Yahoo

- Yahoo has largest publicly announced Hadoop deployments at thousand of nodes across various clusters consuming hundreds petabytes of raw storage.
- Yahoo's Hadoop applications include the following :
  - Search index creation as well as maintenance
  - Optimization of Web page content

- Optimization of Web ad placement
- Spam filters
- Ad-hoc analysis as well as analytic model development

- Before the deploying of Hadoop, it took near about twenty six days to process 3 years' worth of log data.
- By the use of Hadoop, the processing time was reduced to twenty minutes.

Syllabus Topic : Apache Hadoop

6.1.2 Apache Hadoop

Q. 6.1.3 What is Hadoop? (Refer section 6.1.2) **(2 Marks)**

- **Hadoop** is an open source, Java-based programming framework which supports the processing and storage of extremely large sets of data in a distributed computing environment using simple programming models.
- Hadoop has very strong processing power and the ability to handle virtually unlimited parallel tasks.
- With the help of Hadoop, applications can be run on systems with thousands of commodity hardware nodes. It can handle thousands of terabytes of data. Hadoop has distributed file system which facilitates rapid data transfer rates among nodes. This allows the system to proceed even in case one or more nodes get failed. This approach avoids the unexpected data loss.
- Hadoop has quickly emerged as a foundation for big data processing tasks like scientific analytics of data, planning of business and sales, and processing enormous volumes of data including social media data.

History of Hadoop

- Computer scientists Doug Cutting and Mike Cafarella created Hadoop in 2006 to support distribution for the Nutch (search engine). The main aim is to increase the speed of search results by the distribution of data and implement calculations on different computers by multitasking.
- Later on Cutting joined Yahoo but him still works on the Nutch project with the ideas based on Google's early work with automating distributed data storage and processing.
- The Nutch Project divided into two parts –
  - Nutch - Web crawler portion
  - Hadoop - Distributed computing and processing portion
- In 2008, Yahoo released Hadoop as an open-source project. Now Apache Software Foundation (ASF) manages the Hadoop's framework and ecosystem of technologies.

6.1.2(A) Modules (Components) of Hadoop

Q. 6.1.4 Explain different components of HADOOP. (Refer section 6.1.2(A)) **(7 Marks)**

Q. 6.1.5 Explain in brief different building blocks of HADOOP. (Refer section 6.1.2(A)) **(7 Marks)**

HDFS

HDFS stands for Hadoop Distributed File System. It states that the files will be broken into blocks and stored in nodes over the distributed architecture. It provides high-throughput access to application data.

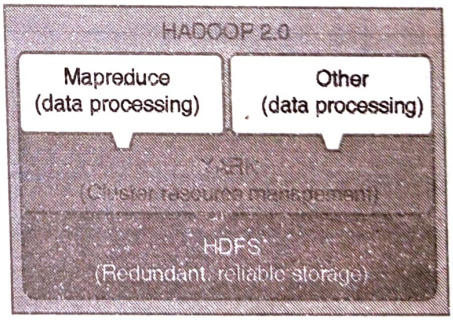

Fig. 6.1.2

- **Yarn** : Yarn stands for "Yet another Resource Negotiator". It is used for job scheduling and managing the cluster (multiple nodes).
- **MapReduce :** This is YARN-based system for parallel processing of large data set using key value pair. The Map task takes input data and converts it into a data set which can be computed in Key value pair.

- **Hadoop Common :** These Java libraries and utilities are used to start Hadoop. These are used by other Hadoop modules. These libraries provide file system and OS level abstractions.

Advantages of Hadoop

1. Huge amounts of any kind of data can be stored and processed quickly.
2. **Computing power :** Hadoop's distributed computing model processes big data fast.
3. **Fault Tolerance :** In case of failure of any node the tasks are automatically redirected to other nodes.
4. **Flexibility :** Any kind of unstructured data like text, images and videos can be stored.
5. **Low cost :** This open-source framework is free.
6. **Scalability :** New nodes can be easily added to handle big tasks.

Disadvantage of Hadoop

1. Hadoop is rough in manner because the software is under active development.
2. Programming model is very restrictive.
3. Joins of multiple datasets are tricky and slow.
4. Cluster management is hard : In the cluster, operations like debugging, distributing. software, collection logs etc are too hard.
5. Requires care and may limit scaling.

Syllabus Topic : MapReduce

6.1.3 MapReduce

Q. 6.1.6 Write a short note on : MapReduce. (Refer section 6.1.3) **(5 Marks)**

- MapReduce is an important part of Hadoop. It is a software framework which is used to write applications easily to process huge amount of data (multi-terabyte data-sets) simultaneously on large clusters (thousands of nodes) in reliable, fault-tolerant manner.
- MapReduce basically referes to two tasks performed by the Hadoop programs. One is map and another is reduce.

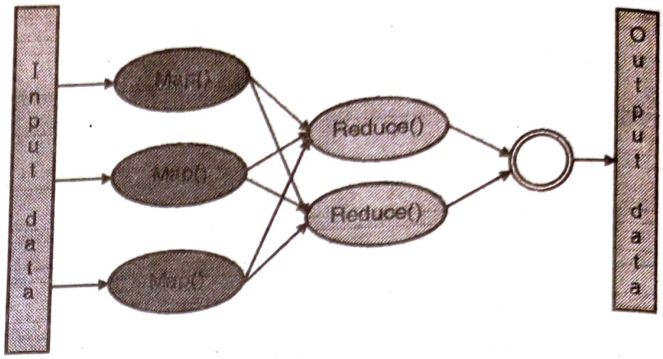

Fig. 6.1.3

- Hadoop programs perform following two tasks on **MapReduce** :
  - **The Map Task :** This is the first task, which takes a set of data and converts it into another set of data in which individual elements are broken down into tuples (key/value pairs).
  - **The reduce :** This job takes the output of previously executed map task as input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce job is always performed after the map job.
- In MapReduce framework, the data of input and output is stored in a file system. The framework handles the scheduling of all the tasks, monitoring these tasks and if fails, re-executes them.
- The main advantage of MapReduce is that it is simple to scale data processing over multiple computing nodes. The data processing primitives are known as mappers and reducers in the MapReduce model.
- The MapReduce framework consists of single master JobTracker and one slave TaskTracker per cluster-node.
- **Master JobTracker :** The tasks under master are
  - Managing the resources.
  - Tracking consumption and availability of resources.

- Scheduling the jobs component tasks on the slaves.
- Monitoring the tasks and re-executing the failed tasks.

- **The slaves TaskTracker :** It execute the tasks as per the directions of the master and provide task-status information to the master periodically.
- The JobTracker is very important in Hadoop MapReduce service. If JobTracker goes down, all running tasks get halted.

☞ Advantages of MapReduce

1. Scalable.
2. Fault tolerant.
3. Simple coding model.
4. Supports unstructured data.

6.1.4 Hadoop Distributed File System (HDFS)

Q. 6.1.7 Explain HDFS with suitable diagram. (Refer section 6.1.4) **(4 Marks)**

- The HDFS is the primary storage system used by Hadoop applications. HDFS is a distributed file system and a framework provided by Hadoop for the analysis and transformation of huge data sets which uses the MapReduce paradigm. The HDFS is based on Google File System (GFS). It provides high-performance access to data across Hadoop clusters (thousands of computers), HDFS has become a key tool for managing pools of big data and supporting big data analytics applications.
- HDFS is usually deployed on commodity hardware of low-cost where the possibility of server failures is common. The file system is designed to be highly fault-tolerant. The HDFS facilitates the rapid transfer of data between different computer nodes and enables Hadoop systems to proceed its execution even if one or more nodes get failed. That decreases the risk of catastrophic failure, even in the event that numerous nodes fail.

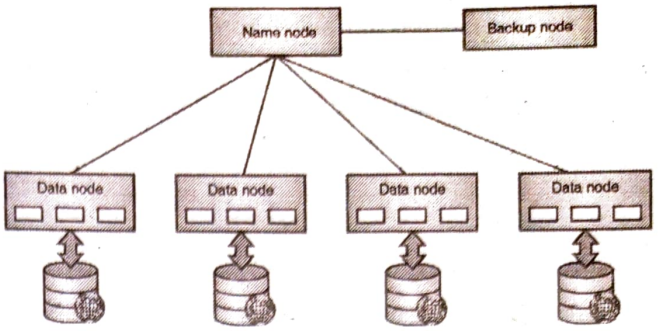

Fig. 6.1.4 : HDFS

- The architecture used by HDFS is known as master/slave architecture.
- NameNode which manages the metadata of file system and DataNode which stores the actual data.
- The HDFS namespace is a hierarchy of files and directories. Inodes are used to represent these file and directories. Inodes are used to record attributes such as permissions, modification and access times etc. The file content is split into large blocks and each block of the file is independently replicated at multiple DataNodes.
- The tree structure of namespace is maintained by the NameNode. It maps the blocks to DataNodes. In a cluster there may be hundreds of DataNodes and thousands of HDFS clients per cluster, as number of application tasks can be executed by each DataNode simultaneously.

☞ Advantages of HDFS

1. High scalability.
2. Low limitation.
3. Open source.
4. Low cost.

☞ Disadvantages of HDFS

1. Still rough - means software under active development.
2. Programming model is very restrictive.
3. Cluster management is high.

Syllabus Topic : The Hadoop Ecosystem

6.2 The Hadoop Ecosystem

Q. 6.2.1 Explain the term : Hadoop Ecosystem. (Refer section 6.2) **(4 Marks)**

- Hadoop is a framework. The Hadoop ecosystem provides the furnishings that turn the framework into a comfortable place for big data activity that reflects the specific needs and tastes of users.
- The Hadoop ecosystem includes both official Apache open source projects and a wide range of commercial tools and solutions.
- Some of the best-known open source examples include Hive, Pig, HBase and Mahout.
- Commercial Hadoop offerings are even more diverse and include platforms and packaged distributions from vendors such as Cloudera, Hortonworks, and MapR, plus a variety of tools for specific Hadoop development, production, and maintenance tasks.
- Hadoop Ecosystem is neither a programming language nor a service, it is a platform or framework which solves big data problems.
- We can consider it as a suite which encompasses a number of services (ingesting, storing, analyzing and maintaining) inside it.
- Let us discuss and get a brief idea about how the services work individually and in collaboration.
  - **Pig :** Provides a high-level data-flow programming language
  - **Hive :** Provides SOL-like access
  - **Mahout :** Provides analytical tools
  - **HBase :** Provides real-time reads and writes

Syllabus Topic : Pig

6.2.1 Pig

Q. 6.2.2 Explain pig with suitable example. (Refer section 6.2.1) **(4 Marks)**

- Apache Pig is a high-level platform for creating programs that run on Apache Hadoop.
- The language for this platform is called Pig Latin.
- Pig can execute its Hadoop jobs in MapReduce, Apache Tez, or Apache Spark].
- Pig Latin abstracts the programming from the Java MapReduce idiom into a notation which makes MapReduce programming high level, similar to that of SQL for relational database management systems.
- Pig Latin can be extended using user-defined functions (UDFs) which the user can write in Java, Python, JavaScript or Ruby and then call directly from the language.

☞ **History**

- Apache Pig was originally developed at Yahoo Research around 2006 for researchers to have an ad-hoc way of creating and executing MapReduce jobs on very large data sets.
- In 2007, it was moved into the Apache Software Foundation.
- **Example :** An example of a "Word Count" program in Pig Latin:

```
input_lines = LOAD '/tmp/my-copy-of-all-pages-on-internet'
AS (line:chararray);

-- Extract words from each line and put them into a pig bag
-- datatype, then flatten the bag to get one word on each row
words = FOREACH input_lines GENERATE
FLATTEN(TOKENIZE(line)) AS word;

-- filter out any words that are just white spaces
filtered_words = FILTER words BY word MATCHES '\\w+';

-- create a group for each word
word_groups = GROUP filtered_words BY word;

-- count the entries in each group
```

```
word_count = FOREACH word_groups GENERATE
COUNT(filtered_words) AS count, group AS word;

-- order the records by count
ordered_word_count = ORDER word_count BY count DESC;
STORE ordered_word_count INTO '/tmp/number-of-words-
on-internet';
```

- The above program will generate parallel executable tasks which can be distributed across multiple machines in a Hadoop cluster to count the number of words in a dataset such as all the webpages on the internet.

Syllabus Topic : HIVE

6.2.2 HIVE

Q. 6.2.3 Explain HIVE with its architecture. (Refer section 6.2.2) **(8 Marks)**

- Apache Hive is a data warehouse software project built on top of Apache Hadoop for providing data summarization, query, and analysis.
- Hive gives an SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop.
- Traditional SQL queries must be implemented in the MapReduce Java API to execute SQL applications and queries over distributed data.
- Hive provides the necessary SQL abstraction to integrate SQL-like queries (HiveQL) into the underlying Java without the need to implement queries in the low-level Java API.
- Since most data warehousing applications work with SQL-based querying languages, Hive aids portability of SQL-based applications to Hadoop.

Features of Hive

Apache Hive supports analysis of large datasets stored in Hadoop's HDFS and compatible file systems such as Amazon S3 filesystem. An SQL-like query language called HiveQL is provided by Hive with schema on read and transparently converts queries to MapReduce, Apache Tez and Spark jobs. All three execution engines can run in Hadoop YARN. To accelerate queries, it provides indexes, including bitmap indexes. Other features of Hive include:

Indexing to provide acceleration

- Different storage types such as plain text, RCFile, HBase, ORC, and others.
- Metadata storage in a relational database management system, significantly reducing the time to perform semantic checks during query execution.
- Operating on compressed data stored into the Hadoop ecosystem using different algorithms.
- Built-in user-defined functions (UDFs) to manipulate dates, strings, and other data-mining tools. Hive supports extending the UDF set to handle use-cases not supported by built-in functions.

Architecture of Hive

Major components of the Hive architecture are :

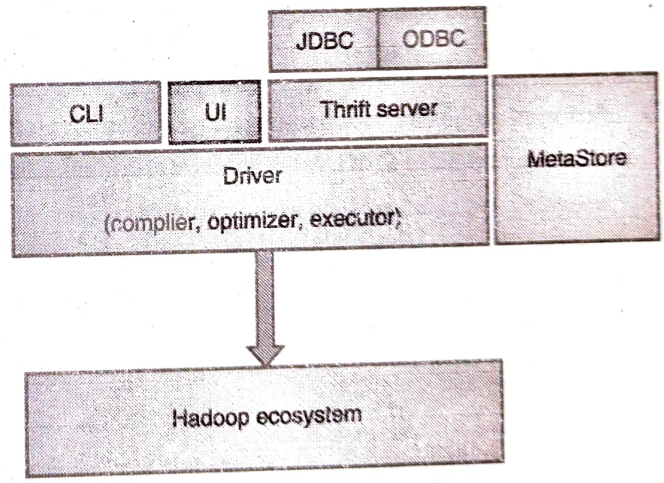

Fig. 6.2.1 : Architecture of HIVE

Metastore

- Stores metadata for each of the tables such as their schema and location. It also includes the partition metadata which helps the driver to track the progress of various data sets distributed over the cluster. The data is stored in a traditional RDBMS format. The metadata helps the driver to keep a track of the data and it is highly crucial. Hence, a backup server regularly replicates the data which can be retrieved in case of data loss.

Driver

- Acts like a controller which receives the HiveQL statements. It starts the execution of statement by creating sessions and monitors the life cycle and progress of the execution. It stores the necessary metadata generated during the execution of an HiveQL statement. The driver also acts as a collection point of data or query result obtained after the Reduce operation.

Compiler

- Performs compilation of the HiveQL query, which converts the query to an execution plan. This plan contains the tasks and steps needed to be performed by the Hadoop MapReduce to get the output as translated by the query. The compiler converts the query to an Abstract syntax tree (AST). After checking for compatibility and compile time errors, it converts the AST to a directed acyclic graph (DAG). DAG divides operators to MapReduce stages and tasks based on the input query and data.

Optimizer

- Performs various transformations on the execution plan to get an optimized DAG. Various transformations can be aggregated together, such as converting a pipeline of joins by a single join, for better performance. It can also split the tasks, such as applying a transformation on data before a reduce operation, to provide better performance and scalability.

Executor

- After compilation and Optimization, the Executor executes the tasks according to the DAG. It interacts with the job tracker of Hadoop to schedule tasks to be run. It takes care of pipelining the tasks by making sure that a task with dependency gets executed only if all other prerequisites are run.

CLI, UI, and Thrift Server

- Command Line Interface and UI (User Interface) allow an external user to interact with Hive by submitting queries, instructions and monitoring the process status. Thrift server allows external clients to interact with Hive.

Syllabus Topic : HBase

6.2.3 HBase

Q. 6.2.4 What is Hbase? Discuss various Hbase Data Model and applications. **(5 Marks)**

(Refer section 6.2.3)

- For data storage and maintenance related problems RDBMS is the solution since 1970.After the evolution of Big Data, the tools like Hadoop get widely spread. For storing big data Hadoop uses distributed file System. Mapreduce is used to process such data.
- As we have seen in previous section, Hadoop can store huge data of various formats such as arbitrary, semi-, or even unstructured.

Limitations of Hadoop

- Hadoop performs batch processing and there is only one way to access the data and that is nothing but sequential way. Random access of data is not possible in Hadoop. This increases the time required to search any data in huge databases.
- There was a requirement of any solution on this problem which can facilitate the random access of data.

What is HBase?

- HBase is the database that store huge amounts of data and access the data in a random manner.
- HBase built on top of the Hadoop file system.
- It is a distributed column-oriented database.
- It is an open-source project and is horizontally scalable.
- HBase leverages the fault tolerance provided by the Hadoop File System (HDFS).
- Hbase is a part of the Hadoop ecosystem that provides random real-time read/write access to data in the Hadoop File System.
- HBase is used to store data in HDFS which can be easily read and write randomly.

Features of HBase

- Strictly consistent reads and writes.
- Automatic and configurable sharding of tables
- Linear and modular scalability.
- Block cache and Bloom Filters for real-time queries.
- Automatic failover support between RegionServers.
- Easy to use Java API for client access.
- Query predicate push down via server side Filters
- Convenient base classes for backing Hadoop MapReduce jobs with Apache HBase tables.

6.2.3(A) HBase Data Model

Q. 6.2.5 Discuss Hbase data model. (Refer section 6.2.3(A)) **(5 Marks)**

HBase stores data in table format. Tables are combinations of rows and columns.

HBase Data Model Terminology

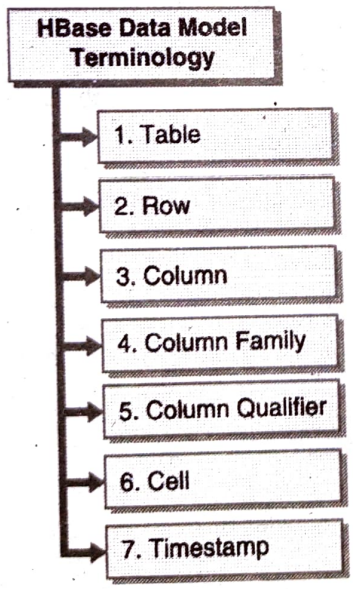

Fig. 6.2.2 : HBase Data Model Terminology

1. Table

An HBase table consists of multiple rows.

2. Row

In HBase the row contains a row key and one or more columns with values associated with them. Records are stored in sorted format depending upon the row key. The main focus while storing data is that the related rows should be stored near to each other.

3. Column

In HBase column contains a column family and a column qualifier. These are delimited by a colon(:) character.

4. Column Family

Column families contain set of columns and their values. The column family has set of storage properties, such as whether its values should be cached in memory, how its data is compressed or its row keys are encoded, and others.

The column families are same in all the tables even though do not have any value. At the time of table creation, the column families are specified.

5. Column Qualifier

To provide the index for a given piece of data, a column qualifier is added to a column family. Column qualifiers are mutable and different column qualifiers may assigned to different rows.

6. Cell

Cell contains a value and a timestamp, which represents the value's version. A cell is a combination of row, column family, and column qualifier.

7. Timestamp

A timestamp is the identifier for a given version of a value. It is written alongside each value. On the RegionServer, the timestamp is by default the time when the data was written. It is also possible to specify different timestamp value while adding data into the cell.

Conceptual View

Each cell has multiple versions, typically represented by the timestamp of when they were inserted into the table					
		Timestamp 1		Timestamp 2	
		Column family - personal		Column family – office	
	Row Key	Name	Residence-phone	Phone	Address
The table is lexicographically sorted on the row keys	00001	Anand	020-48154055	020-21255544	1021 Satara road
	00002	Rahul	020-55453340	020-22222223	1021 Satara road
	00003	Amar	020-37337337	020-17486878	1021 Satara road
	00004	Nitin	020-56789108	020-98886552	4455 Sadashiv Peth
	00005	Sagar	020-27374855	020-45484548	4455 Sadashiv Peth
	00006	Suraj	020-48495567	020-47411112	543 Narayan Peth

Cells

Fig. 6.2.3

In the Fig. 6.2.3, the table contains two column families : Personal and office. Each column family has two columns. Cell is the entity where the actual data is stored. The rows are sorted based on the row keys.

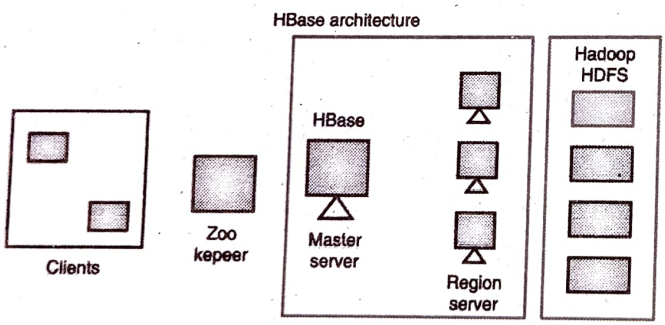

Fig. 6.2.4

- In HBase, tables are divided into regions. These regions are served by the region servers.
- Regions are vertically divided by column families into "Stores". In HDFS the stores are saved as files.
- There are three major components in HBase
  1. Client library
  2. Master server
  3. Region servers.

☞ **MasterServer**

The master server

- It assigns regions to the region servers with the help of ZooKeeper
- IT handles load balancing of the regions across region servers. The busy servers are unloaded and regions are shifted to less occupied servers.
- The state of the cluster is maintained by negotiating the load balancing.
- Is responsible for schema changes and other metadata operations such as creation of tables and column families.

6.2.3(B) HBase Regions

These are the tables that are split up and spread across the region servers.

☞ **Region server**

Functionality of Region Server

- It interacts with the client and handle data-related operations.

- It handles all read and write requests for the regions which are under it.
- The size of the region is decided by it by following the region size thresholds.

The internal structure of region server is as follows :

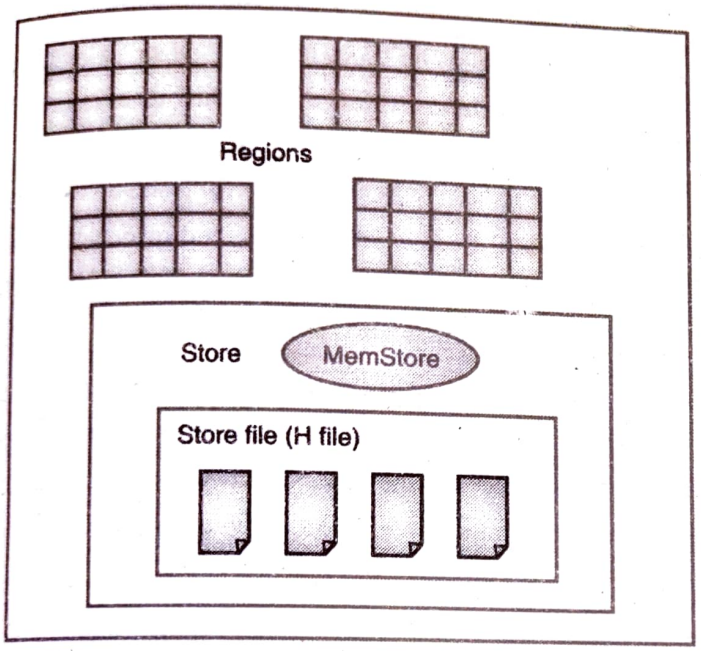

Fig. 6.2.5

The memory store and HFiles are placed in the store. The MemStore works like cache memory. Initially whatever which enters into HBase is stored here. Later on that data is moved to Hfiles as blocks and the memStore is flushed.

Zookeeper

- Zookeeper is an open-source project. It provides services like naming, providing distributed synchronization, maintaining configuration information, etc.
- There are temporary nodes in Zookeeper which represents different region servers. These nodes are used by Master Servers to discover available servers.
- The server failure or network partitions are also tracked with the help of nodes.
- Zookeeper helps to set communication between clients and region.

Syllabus Topic : Mahout

6.2.4 Mahout

Q. 6.2.6 Write note on Mahout. (Refer section 6.2.4) **(3 Marks)**

- As soon as the dataset is available in HDFS, the further action is to apply an analytical technique on the data.
- For small datasets, tools like R are effective, but in case of large datasets, there may be performance issue with these tools.
- In the environment of Hadoop, Apache Mahout is good option to apply the analytical techniques.
- Mahout offers executable Java libraries to apply analytical techniques in a scalable manner to Big Data.
- As we know, a mahout is a nothing but a person who can control an elephant.
- Apache Mahout is considered as a toolset which directs Hadoop (the elephant in this case), to get meaningful analytic results.
- Java code is provided by Mahout which implements the algorithms for various techniques in the following three categories:

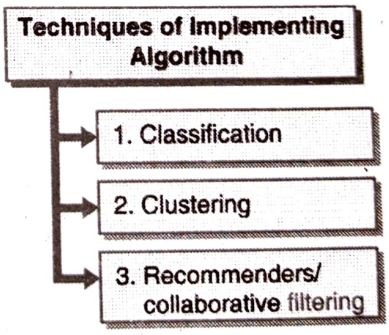

Fig. 6.2.6 : Techniques of implementing algorithm

1. Classification

(a) Logistic regression

(b) Na"lve Bayes

(c) Random forests

(d) Hidden Markov models

➜ **2. Clustering**

(a) Canopy clustering

(b) K-means clustering

(c) Fuzzy k-means

(d) Expectation maximization (EM)

➜ **3. Recommenders/collaborative filtering**

(a) Nondistributed recommenders

(b) Distributed item-based collaborative filtering

Syllabus Topic : NoSQL

6.3 NoSQL

Q. 6.3.1 Explain NoSQL with advantages. (Refer section 6.3) **(8 Marks)**

The relational databases are widely used in software industry. The design of relational databases is not such that which can cope with the scale and agility challenges that face modern real time applications, nor were they built to obtain benefit of the commodity storage and processing power available now a day. Now a days the data management becomes very difficult because of the tremendously increase in the size of data in various emerging fields like social networking, e-commerce etc.

- NoSQL is also known as "non SQL" or "non relational" or "Not Only SQL". NoSQL is database which provides a complete different mechanism for storing and retrieval of data which is modelled in means other than the tabular relations used in relational database management systems.
- Sometimes the term NOSQL seems to be confusing as handling data without SQL is out of imagination for some people. But actually the meaning of SQL is Not Only SQL.
- NoSQL challenges the dominance of relational databases. NoSQL databases are increasingly used in big data and real-time web applications.
- In NoSQL the data structures used like key-value, column, graph or document are different from those used in traditional relational database system which makes some operations faster in NoSQL.
- Usually the data structures used by NoSQL databases are also more flexible than relational database system.
- NoSQL allows the insertion of data without a predefined schema (design). In this database, it is possible to make significant application changes in the system without having worry about service interruptions. Because of it, the speed of development increases, the integration of code becomes more reliable and time of database administration decreases.
- To enforce data quality controls, developer has to add application-side code. NoSQL supports validation rules to be applied on the database which helps user to control the data while maintaining the advantage of a dynamic schema.
- NoSQL databases are more concentrated on availability, partition tolerance, and speed for which they may compromise the consistency. The basic reason of no wide adoption of NoSQL is use of low level query language rather than SQL.

☞ History of NoSQL

- Such database have existed since year 1960, but not known as "NoSQL". Need of this database comes in picture with the rise of web related applications like Google, Facebook, Amazon etc.
- In 1998 Carlo Strozzi first introduced a lightweight, open source relational database system which did not expose the standard Structured Query Language (SQL) interface. Carlo Strozzi gives the name as "NoSQL" to it. He suggested that as the NoSQL is differ from the traditional relational model, it should be called as "NoREAL" means "No Relational".
- Ohan Oskarsson, then a developer at Last.fm, reintroduced the term *NoSQL* in early 2009 when he organized an event to discuss "open source distributed, non relational databases". The name attempted to label the emergence of an increasing number of

non-relational, distributed data stores, including open source clones of Google's BigTable/ MapReduce and Amazon's Dynamo.

- Most of the early NoSQL systems did not attempt to provide atomicity, consistency, isolation and durability guarantees, contrary to the prevailing practice among relational database systems.
- Based on 2014 revenue, the NoSQL market leaders are MarkLogic, MongoDB, and Datastax. Based on 2015 popularity rankings, the most popular NoSQL databases are MongoDB, Apache Cassandra, and Redis.

Use NoSQL when needs are like

1. Decentralized applications (e.g. Web and mobile)
2. Continuous availability; no downtime
3. High velocity data (devices, sensors, etc.)
4. Data coming in from many locations
5. Structured data is available with some semi/unstructured data.
6. To maintain high data volumes; retain forever.

What is NoSQL ?

- NoSQL systems are also called as "Not only SQL" which indicates that they may support query languages like SQL.
- NoSQL is not depending on column, rows or schema for structure, it is no-relational database management systems. The data models of NoSQL are more flexible.
- NoSQL provides scalability, availability and fault tolerance and emerges as a alternative for relational database.
- The speciality of NoSQL is that, it may not require fixed table schemas avoids join operations, and also scale horizontally.
- Developers are working with applications that create massive volumes of new, rapidly changing data types - structured, semi-structured, unstructured and polymorphic data. NoSQL is useful for data which is growing far more rapidly or unstructured data or data which does not store in the relational schemas of RDBMS. There are common types of unstructured data: user and session data; chat, messaging, and log data; time series data such as IoT and device data; and large objects such as video and images.

Features of NoSQL

- Design simplicity.
- Simpler "horizontal" scaling to clusters of machines. This was a problem in relational databases.
- More control over data availability.

Observations regarding NoSQL

- It does not use the relational model.
- It runs well on clusters.
- NoSQL is Mostly open-source.
- It is Schema-less.

Why NoSQL ?

- All IT professionals and industry database experts come to know that NoSQL is here to stay.
- A recent study performed on NoSQL market growth forecasts a very strong compound annual growth rate of 21 percent for NoSQL technology from 2013-2018. It shows bright future for NoSQL.
- NoSQL databases has been much more clearly articulated today. NoSQL database refers to groups of databases that are not based on relational database model.
- The data storage model used by NoSQL database is not some fixed data model, but the common features among the NoSQL database is that the relational and tabular database model of SQL based database is not used.

Advantages of NoSQL

We cannot consider that NoSQL databases are straight substitution for relational database management system (RDBMS). But for many issues regarding data, NoSQL seem to be better.

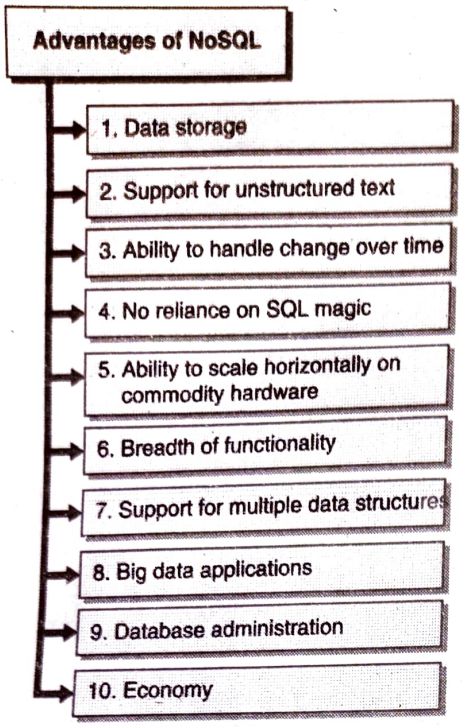

Fig. 6.3.1 : Advantages of NoSQL

➜ **1. Data storage**

NoSQL databases supports storing data "in the form of Key value pair which give ability to store simple data structures. The document NoSQL database provides the ability to handle a range of flat or nested structures.

➜ **2. Support for unstructured text**

NoSQL databases can handle unstructured text easily. This ability increases information effectively and can help organizations make better decisions.

➜ **3. Ability to handle change over time**

NoSQL databases are capable of managing changes because of the systematic storage system.

➜ **4. No reliance on SQL magic**

SQL(Structured Query Language) is the predominant language which is used to write queries in relational database management systems. Even if several NoSQL databases provide support for SQL access, they do so for compatibility with existing applications like business intelligence (BI) tools. NoSQL is not dependent on SQL for processing. NoSQL databases support their own query languages that can support data processing.

➜ **5. Ability to scale horizontally on commodity hardware**

NoSQL databases supports distribution of a database across several servers. Hence if there is requirement of more data storage, then number of servers can be increased and connect them to database cluster (horizontal scaling) making them work as a single data service.

➜ **6. Breadth of functionality**

Near about all the relational databases support the same characteristics but in a slightly different way, so they are all similar. In contrast, the NoSQL databases come in different core types: key-value, document store and graph. Out of these types, the one select to suit our requirements is not hard.

➜ **7. Support for multiple data structures**

There is requirement of simple as well as complex data structures. NoSQL databases provide support for a range of data structures. Key-value stores can handle Simple binary values, lists, maps, and strings. Document databases can manage highly complex parent-child hierarchal structures Graph stores can describe the web of interrelated information.

➜ **8. Big data applications**

In some systems the data grows rapidly. Such big volume data can be easily handled by NoSQL databases.

➜ **9. Database administration**

The NoSQL has data distribution and auto repair capabilities, simplified data models and fewer tuning and administration requirements. This leads to less requirement of hands-on management.

➜ **10. Economy**

These databases are designed to be used with low-cost commodity hardware.

It is difficult for application developer to find match between the relational data structures and the in-memory data structures. Using NoSQL databases they can develop the system without having to convert in-memory structures to relational structures.

☞ Disadvantages of NoSQL

1. No standard schema.
2. Less use of SQL.

☞ Companies using NoSQL

Now a day's many companies using NoSQL. Some of them are :

- Google
- Facebook
- Adobe
- Foursquare
- Digg
- Vermont public radio
- LinkedIn
- Mozilla

6.3.1 Types and Examples of NoSQL Database

There have been various approaches to classify NoSQL databases, each with different categories and subcategories.

Following are some types of NoSQL :

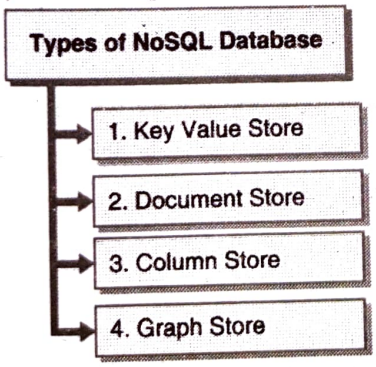

Fig. 6.3.2 : Types of NoSQL database

➜ 1. Key Value Store

Q. 6.3.2 Explain key value store NoSQL data model. (Refer section 6.3.1) **(5 Marks)**

- The Key-value (KV) store system uses the concept of associative array, as their fundamental data model. In this model, data is represented as a collection of key-value pairs. Every single item in the database is stored as an attribute name (or 'key'), together with its value.
- In NoSQL database, a table exists with two columns : one is the Key and the other is Value.
- The key in the key-value pair should not be repeated because it the unique identifier that helps to access the value associated with that key uniquely.
- The Key value stores allow the developer of application to store schema-less data.
- Key-value databases are the simplest form of the NoSQL databases. Other advanced models are mostly extensions to this key-value model.
- **Examples** include Memcached, Riak, ArangoDB, InfinityDB, Oracle NoSQL Database, Redis and dbm.
- All key-value databases are not similar; there are major differences between these databases. For example: Data in **MemcacheDB** is not persistent while in **Riak** it is persistant. Such features are useful when implementing some solutions. It is important to not only choose a key-value database based on your requirements, it is also important to choose which key-value database.
- Examples of key-value NoSQL database applications :
  - Dynamo
  - MemcacheDB
  - Redis
  - Riak
  - FairCom
- Four main core operations perform on key-value store :
  1. **Get(key) :** It return the single value of given key.
  2. **Put(key,value) :** It assigns value to key.
  3. **Multi-get(key1,key2,key3,..,keyn) :** It returns multiple values of given multiple keys.
  4. **Delete(key) :** It deletes both key and value present in it.

Example

This is a simple example for key-value store. Here keys are the names of employees and values are their contact numbers.

Key	Value
Sam	(234) 567-8901
jack	(134)526-6845
Ron	(245)452-4584
kenny	(356)584-1458

Where key value store is used ?

Key-value databases can be used in many scenarios. Such as,

General Web / Computers

- User profiles
- Article/blog comments
- Emails

E-commerce

- Shopping cart contents
- Product categories
- Product details

Advantages of Key Value Store

- Key value store is the simplest type of NoSQL.
- Supports simple queries very efficiently.
- Extended form of key-value stores is able to sort the keys.
- It is specially designed for storing data as a schema free data.
- Very simple data-modeling pattern should be understandable by anyone.
- With little or no maintained indexes, the key-value stores are designed to be more scalable and extremely fast.
- Suitable for system where data is not highly related.

Disadvantages of Key Value Store

- The indexing and scanning capabilities are absent. It does not help if we want to perform more operation as per user requirement than the basic CRUD (Create, Read, Update, Delete) operations.
- Only one row simple queries can be executed efficiently.
- Difficult to perform SQL operations like JOINS, GROUP BY etc.
- Selecting appropriate data type for value is difficult.
- Difficult to use constraints like FOREIGN KEY or NOT NULL.
- More application code is required to reassemble collections of key-value pairs into objects.

2. Document Store

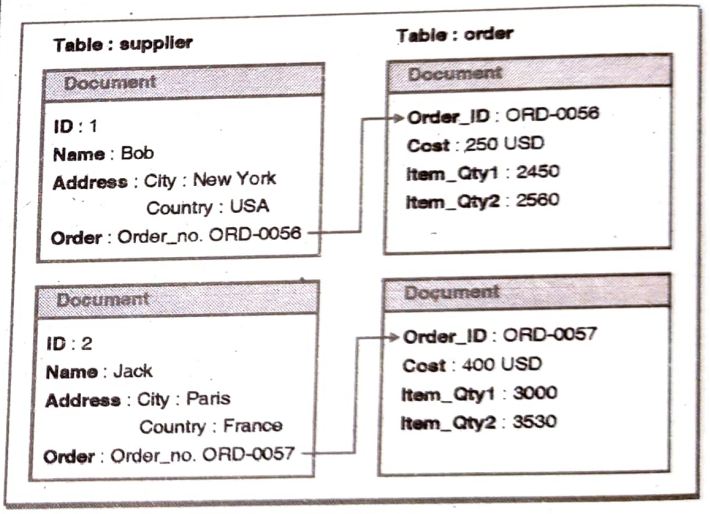

Fig. 6.3.3

- These databases store records as "documents" where a document can generally be thought of as a grouping of key-value pairs.
- The documents are identified by the unique keys which represents them. One defining characteristics of a document-oriented database is that in addition to the key lookup performed by a key-value store, the database offers an API or query language that retrieves documents based on their contents.
- Document databases are extension to key-value store. Addition to query capabilities of key-value databases,

they provide indexing and the ability to filter documents based on attributes in the document.

- Examples of Document Store NoSQL database applications :
  - MongoDB
  - Couchbase

☞ **Advantages of Document Store**

- The performance is good and the distribution across various servers becomes a lot easier.
- There is no need of translation between object in SQL and application. The object can directly be converted into document.
- They have strong indexing features and can rapidly execute different queries.

➜ **3. Column Store**

- Column store is column oriented NoSQL database. Data is stored in cells grouped in columns of data rather than as rows of data. The logical grouping of columns is created in column families. There is no limitation on number of columns in a column family.
- These columns can be created at runtime. The operations like read write are performed using columns rather than rows.
- The column store structure gives the advantage of fast search / access and data aggregation over the row format data storage of relational databases.
- Relational databases store a single row as a continuous disk entry. Different rows are stored in different places on disk while column store database store all the cells related to a column as a continuous disk entry which makes the search/access faster.
- **For example :** Displaying titles from a bunch of a million articles will be a tedious and time wasting task while using relational databases as it will go over each location to get item titles. While in column store, title of all the items can be obtained with just one disk access.
- Examples of column store NoSQL database applications :
  - HBase
  - BgTable
  - HyperTable

☞ **Advantages of Column Store**

- Efficient storage and data compression.
- Fast data loads.
- Simple configuration.

☞ **Disadvantages of Column Store**

- Queries with table joins can reduce high performance.
- Transactions are to be avoided or just not supported.

➜ **4. Graph Store**

- The Graph Store NoSQL database technology is designed to handle very large sets of data which may be structured, semi-structured or unstructured.
- In a Graph Base NoSQL Database, rigid format of SQL or the tables and columns representation does not exists. Rather a flexible graphical representation is used which is perfect to address scalability concerns. The graph structure contains edges, nodes and properties.
- The graph store is helpful in transferring the data from one model to other very easily.
- Graph store stores the entities and relationships between these entities. Entities are also known as nodes, which have properties. Nodes represent entities such as people, businesses, accounts, or any other item to be tracked.
- They are roughly the equivalent of the record, relation, or row in a relational database, or the document in a document database.
- Relations are known as edges that can have properties.
- Edges are denoting directions. Nodes are generally organized with the help of relationships. The data is stored once by the organization of graph. This data can be interpreted in different ways depending upon the relationships between nodes.

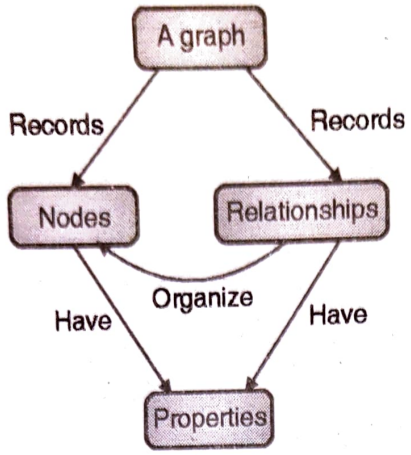

Fig. 6.3.4

- Key points related to graph store.
- Edges and nodes are used by these databases to represent and store data.
- The nodes are organised with the help of some relationships in between them, which is represented by edges between the nodes.
- Both the nodes and the relationships have some definite properties.
- Examples of Graph store NoSQL database applications :
  - Neo4j
  - Polyglot

☞ Advantage of Graph Store

- Performance
- Flexibility
- Agility(ability to move quickly or easily)

☞ Comparison of all the four NoSQL Databases

Database model	Performance	Scalability	Flexibility
Key value store database	High	High	High
Column store database	High	High	Moderate
Document store database	High	Variable(high)	High
Graph database	Variable	Variable	High

6.3.2 Comparative Study of SQL and NoSQL

For managing the database SQL has been most widely used programming language over the last few decades. It is a Relational Database Management System. But in today's era NoSQL has arises for an option to the SQL. There is very high difference between SQL and NoSQL. In this topic we will see the comparative study of SQL and NoSQL.

☞ The conceptual difference

A framework of a relational database which is setup with the defined categories used by SQL. The tables of SQL are idle for storing data which is structured. The structured data like name, address fits into the SQL format perfectly.

But when data is unstructured it needed another format which is not dependant on the relationships of the data. When this situation occurs that time we can use NoSQL. NoSQL allows for the storage of an unstructured data without categorizing the data into fixed tables. NoSQL database scales horizontally. NoSQL is also known as Non-relational database or distributed database.

☞ Factual difference

- In SQL databases are structured in the form of tables, but in NoSQL databases are structured in the form of documents, graphs, or key-value pairs.
- In SQL Database there is a standard definition of schema which must be worked with the structured data. While In NoSQL there is no standard definition for schema which must be worked with the structured data.
- SQL database have predefined schema for structured data while NoSQL database have dynamic schema for unstructured data.
- SQL databases has feature of vertical scaling while NoSQL databases has feature of horizontal scaling.
- SQL database are designed and managed with SQL (Structured Query language) while NoSQL database are designed and managed with the UnQL (Unstructured Query Language).
- The syntax of SQL does not vary with database, while the syntax of UnQL varies with database.

- SQL is preferable for handling complex query while NoSQL cannot handle complex query. SQL queries are more powerful than NoSQL queries.
- Examples of SQL databases are : MySQL, Oracle, MS-SQL while examples of NoSQL are : MongoDB, BigTable, Redis, Hbase, Neo4i and CouchDb.
- SQL cannot manage big data which is stored in hierarchical manner while NoSQL handles hierarchical data better than SQL. Hence, NoSQL is preferable to SQL when managing big data.

☞ Comparision between SQL and NoSQL

Q. 6.3.3 Compare SQL and NoSQL. (Refer section 6.3.2) **(4 Marks)**

Parameters	SQL	NoSQL
Long form	SQL means structured query language.	NoSQL means NOT only SQL language.
Data Format	Data is in structured and organized form.	Data is in unstructured form.
Use	It is used for small to medium scale data set effectively.	It is used for large set of data.
Schema	SQL is schema based.	NoSQL is schema less.
Data Type	Data is stored as row and column in table where each column is of specific type.	The data model is depending upon the database type.
Forms	Relational database is table based.	Key value pair, storage, column, document store, graph database.
Exmaple	SQL server Oracle	MongoDB HBase

6.3.3 NoSQL Data Models

Q. 6.3.4 Explain NoSQL Data Model. (Refer section 6.3.3) **(4 Marks)**

☞ Data Model

Data model is the mode which organizes the data. It is a representation which is used to understand and manipulate the data. The data model helps to :

- Represent the data elements in analytical view.
- Maintain relationship of these elements.
- Describe the way by which we interact with the database.

In RDBMS the relational model was used to represent the data. The data is represented in the form of tables (combination of rows and columns). Each row represents some entity while columns represent relationships in the same table or with another table. The relational data modeling has been a well-defined discipline for many years. However now a days with the emergence of NoSQL databases, need of a new data model arises.

☞ NoSQL Data Model

The NoSQL data model is different from traditional relational data model. There are different data models :

- Key-value
- Document
- Column - family
- Graph

The three model Key-value, document and column-family share common features of Aggregate Orientation.

☞ NoSQL Agreegate Model

- In this model it considered that we have to manage more complex data compared to relational model. The complex data structure are in the form of map, list etc. The aggregate model uses this complex structure. Aggregate is a collection of data objects which are treated as a single unit to manage and manipulate. Atomic operations are expected to update aggregates. Using aggregate it is easy to work on cluster which is unit of machines. Aggregates helps application developer by solving the mismatch problem which usually occur in relational model.
- When the aggregate model runs on a cluster, it gives several advantages on computation power and data distribution. While gathering data, it requires minimizing the number of nodes. The aggregate gives an important view that which data should be stored together.

- The **Key-Value** and **Document** databases are strongly aggregate oriented. These databases contain number of aggregates with a key to get the data. In Key-Value we can store any type of object.
- The Column-Family has two level aggregate structure. The first key is the row identifier. The second level values are defined to as columns.

Important point related to NoSQL Data Aggregate Model

- All these models use aggregated index by a key.
- The key is used for searching the data.
- The Aggregate acts as a atomic unit for modification.
- In the document model, the document is treated as single unit of storage. This model makes document transparent for querying.
- In Column-Family model, the columns are divided into column families and treated as single units.
- All models improve the accessibility of data.

Syllabus Topic : An Analytics Project - Communicating, Operationalizing

6.4 An Analytics Project - Communicating, Operationalizing

Q. 6.4.1 Explain an analytic project with communicating and operationalizing. (Refer section 6.4) **(8 Marks)**

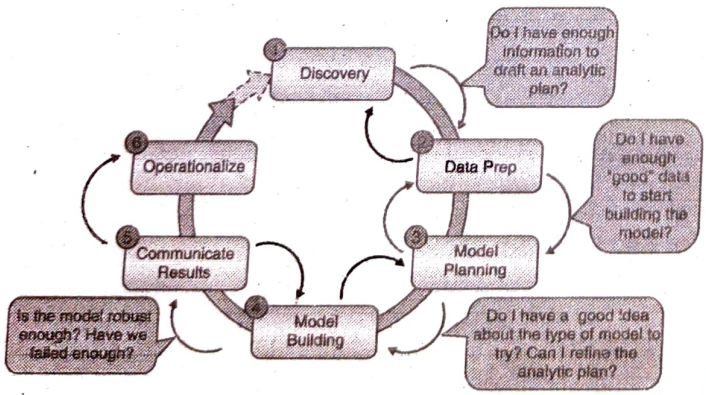

Fig. 6.4.1 : Data Analytic Lifecycle

- As we have seen in the first chapter, the last phase of Data Analytic Life Cycle is Operationalize.
- In this last phase, the various teams require to assess the benefits of the project work and set up a pilot for the purpose of deploying the models in a managed way prior to broadening the work and sharing it with all the employees of the enterprise or ecosystem of users.
- Here, a pilot project indicates a project before a full-scale rollout of the newly added algorithms or functionality.
- This pilot project may have limited scope and rollout to the lines of business, products, or services which may be affected because of newly added models.
- The ability of team regarding quantifying the benefits and share them in manageable way with the stakeholders will decide the future of pilot project and ultimately the work will move forward in a production environment.
- Hence it is very important to identify the benefits and state them with clear and unambiguous manner in the final presentations.
- As the team tries to deploy the analytical model as a pilot project, there is also necessity to consider running the model in a production environment for a specific quantity of products which will give the exact evaluation in a live setting.
- This helps the team to understand the adjustments they should do before deployment.
- In this phase a new set of team members is formed mostly consisting of those engineers who are responsible for the production environment and having new issues or concerns.
- The core interest of this group is to ensure that running the model fits in comfortable way into the production environment and it is easily possible to integrate model into downstream processes.
- When the model is being executed in the production environment, it is responsibility of the team to detect input anomalies before providing to the model, assess run times, and also measure competition for getting the available resources with other processes in the production environment.

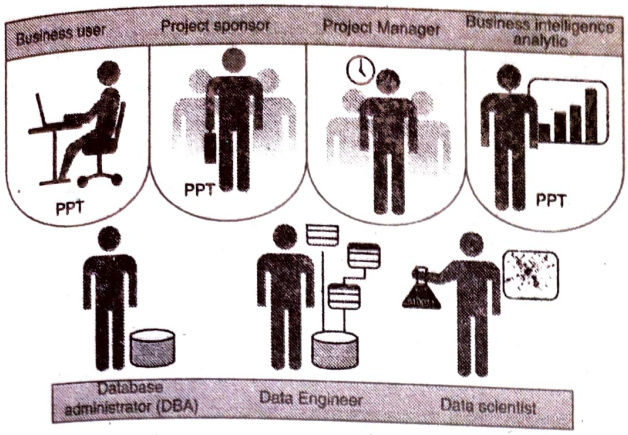

Fig. 6.4.2 : 12-2 Key outputs from a successful analytic project

- Here we will see a brief review regarding the key outputs of all the important stakeholders of an analytics project and what is their expectation at the conclusion of a project:
- **Business User** : They are interested in determining the benefits and implications of the findings to the business.
- **Project Sponsor** : These stakeholders are mostly interested in the questions regarding the business impact of the project, the risks and ROI (return on investment) and the way by which the project can be implemented within the organization and beyond.
- **Project Manager** : They need to conclude if the project completion will be on time and within budget.
- **Business Intelligence Analyst** : Then they need the information about the reports and dashboards managed by them; whether they will be impacted and need to change.
- **Data Engineer and Database Administrator (DBA) :** They have typically need of sharing the code from the analytical project and generate technical documents which will describe the way to implement the code.
- **Data Scientists** : They have typically need of sharing the code and explain the detail information of model to their peers, managers, and other stakeholders.
- Four **important deliverables** are as follows in which most of the stakeholders are interested :

1. Presentation related to Project Sponsors includes high-level takeaways which are useful for executive-level stakeholders, with a few important messages. It supports their decision-making process.
2. Presentation for Analysts, which contains the information regarding changes to business processes and reports. The data scientists which interact with this presentation are well-known to technical graphs and will be interested in the details.
3. Code for the use of technical people like engineers and others which are responsible for managing the production environment.
4. Technical specifications for the purpose of implementing the code.

Syllabus Topic : Creating Final Deliverables

6.5 Creating Final Deliverables

Q. 6.5.1 Explain an Final Deliverables in an analytic project. (Refer section 6.5) **(8 Marks)**

- After the process of reviewing the list of major stakeholders for data science projects and main deliverables, now we are going to see deliverables in detail.
- To understand it properly, we will discuss an example of scenario of a fictional bank, YoyoDyne Bank.
- Churn rate here indicates the frequency with which customers cut their relationship as customers of YoyoDyne Bank or switch to another competing bank.

☞ Synopsis of YoyoDyne Bank case study

- YoyoDyne Bank is a retail bank which desire to enhance its NPV (Net Present value) as well as customer retention rate.
- For this purpose it likes to set an effective marketing campaign which should target the customers to decrease the chum rate by minimum five percent.
- The bank wants to conclude whether it is worth to retain those customers.

- Also the bank wants to analyze the important reasons because of which customers are leaving and want to resolve them.
- The bank wish to establish a data warehouse to enhance marketing and other associated customer care groups.
- As per the this information, the team of data science may generate an analytics plan as given in Fig.6.5.1 . during the project :

Components of Analytic Plan	Retail Banking : YoyoDyne Bank
Discovery Business Problem Framed	How can the bank identify customers with the highest likelihood for churn ?
Initial Hypotheses	Transaction volume and type are key predictors of churn rates.
Data and Scope	5 months of customer account history.
Model Planning Analytic Technique	Logistic regression to identify most influential factors predicting churn.
Result and key findings	Key predictors of churn are : 1. Once customers stop using their accounts for gas and groceries, their account holdings quickly diminish and the customers churn. 2. If the customers use their debit card fewer than 5 times per month, they will leave the bank within 60 days.
Business impact	By targeting customers who are at high risk for churn customer attrition can be reduced by 23%. This would save $3 million in lost customer revenue and avoid $1.5 million in new customer acquisition costs each year for the bank.

Fig. 6.5.1 : Analytics plan for YoyoDyne Bank case study

6.5.1 Developing Core Material for Multiple Audiences

- Since few components of the projects may be used for different audiences, it will be better to generate a core set of materials related to the project, which can be used to make presentations for different elements such as technical audience or executive sponsor.
- In the Table 6.5.1 we can see the main components of the final presentations for the project sponsor and an analyst audience.
- Remember that teams have seven areas to create a core set of materials which will be useful for the two presentation audiences.
- It is possible to use three areas (Project Goals, Main Findings, and Model Description) for both presentations.
- For other areas there is need of additional elaboration, like the Approach.
- Still other areas, for example Key Points need special levels of details for analysts and data scientists than for the project sponsor.

Table 6.5.1 : Comparison of Materials for Sponsor and Analyst Presentations

Presentation Sponsor Component	Project Sponsor Presentation	Analyst Presentation
Project Goals	List most important 3-4 agreed-upon goals.	
Main Findings	Emphasize key messages.	
Approach	High-level Methodology	High-level methodology Appropriate particulars about modeling techniques and technology
Model Description	Overview of the modeling technique	
Key Points Supported with Data	Support key points by the use of simple charts and graphics such as bar charts.	Display details to support the key points. Analyst-oriented charts and graphs like ROC curves and histograms. Visuals and significance of key variables.

Model Details	Skip this particular section, or discuss only at a high level.	Display the code or main logic regarding model, and comprise model type, variables, and technology used to execute the model and score data. Recognize important variables and their impact. Describe anticipated model performance. Detailed description regarding the modeling technique Discuss variables, scope, and predictive power.
Recommendations	Concentrate on business impact, considering risks and ROI. Give the sponsor most important points to implement work within the organization.	Provide recommendations with implications for the modeling or for the purpose of deploying in a production environment.

6.5.2 Project Goals

1. Build a predictive model for the purpose of determining customers which are most likely to churn.
2. It is necessary that the predictive power of the model must be necessarily good as existing customer retention techniques used by the bank.
3. On weekly basis, in production environment, the model should be able to run on entire data set.

Situation & Project Goals

1. YoyoDyne Bank desires to increase the NPV (Net Present Value) and retention rate of the customers
2. In the last 5 weeks, YoyoDyne Bank has lost 10 of its 100 key customers and is experiencing increased competition from its biggest competitor.
3. In the absence of a fast remediation plan, YoyoDyne Bank has risk of losing its dominant position in three key markets.

Goals of YoyoDyne "Churn Project"

1. Build a predictive model for the purpose of determining customers which are most likely to churn.
2. It is necessary that the predictive power of the model must be necessarily good as existing customer retention techniques used by the bank.
3. On weekly basis, in production environment, the model should be able to run on entire data set.

6.5.3 Approach

- In the Approach portion of the presentation, there is need of explaining the methodology which has been implemented on the project by the team.
- It can contain interactions with domain experts, the various groups which are collaborating within the organization, and a some description about the solution developed.
- Any extra comments associated with working assumptions must be included by the team which it is following while performing the work.
- For the project sponsors, the level of discussion should be high while explaining the solution.
- When explaining the solution, the discussion should remain at a high level for the project sponsors.
- If the data is being presented to analysts or data scientists, then there is need to provide additional detail regarding the model type, with the technology as well as the real performance of the model throughout the various tests taken.
- At the end, as part of the description regarding the approach, the team can illustrate various types of constraints from several elements such as systems, tools, or current processes and any type of implications for any updations in these things to work with the project.
- In the following example we will see the way to describe the methodology implemented in a data science project to a sponsor audience.

Approach (for Sponsors)

- Communicated with 20 members from retail lending team for the purpose of understanding YoyoDyne's lending policies as well as marketing practices for customer retention.
- Make collaboration with IT for the purpose of identifying appropriate datasets and review data quality and availability
- Build churn model to recognize customers who are most likely to leave the bank
  - Categorize most of the influential factors
  - Make available the high explanatory power for the analysis of impact of several factors on churn
- Insert effective type of social media data to the model so as to improve predictive power.
- Worked with IT for the purpose of replicating model performance within YoyoDyne's production environment.
- Note that while describing the approach with analysts some technical details may get added

Approach (for Analysts)

- Communicated with 20 members from retail lending team for the purpose of understanding YoyoDyne's lending policies as well as marketing practices for customer retention.
- Make collaboration with IT for the purpose of identifying appropriate datasets and review data quality and availability
- Build churn model in R using a Generalized Addictive Modeling technique
  - Minimizes variable transformations and binning.
  - Make available the high explanatory power for the analysis of impact of several factors on churn
- Insert effective type of social media data to the model so as to improve predictive power and examine its impact.
- Worked with IT for the purpose of replicating model performance within YoyoDyne's production environment.
- The model can be quickly scored in the DB over big datasets with the help of a SQL (Structure Query Language) code generator.

6.5.4 Model Description

- After the process of describing the project approach, teams usually describe the model that was used.
- **Overview of Basic Methodology :** Guess the probability of churn for each customer. Find the customers having greater likelihood for churn and make comparison with actual churn outcomes to train the algorithm.
- **Model :** Logistic Regression Model
- **Dependent variable :** Binary variable, of churn/no churn

Scope

- 1,20000 Yoyodyne bank customers, as per churn within 100 day period after 1/31/2016
- 1,20000 Customers with all churners through 6/30/16.
- All selected customers were Active, Suspended or Pendtng as of 1/31/2016

Sampling

- **Training sample:** 20,000 subscribers
- **Testing sample:** 30,000 subscribers
- The new model developed has predictive power at least as good as the bank's current churn model
- In the next step key points are identified depending upon insights as well as observations which are basically resulted from the data and model scoring results.

□□□