**Big Data and NoSQL Concept**

**5. What is Big Data? Explain properties/characteristics of Big Data?**

**Big Data** is a collection of data that is huge in volume, yet growing exponentially with time. It is a data with so large size and complexity that none of traditional data management tools can store it or process it efficiently. Big data is also a data but with huge size.

**Characteristics of Big Data**

Big data can be described by the following characteristics:

* Volume
* Variety
* Velocity
* Variability

**(i) Volume –** The name Big Data itself is related to a size which is enormous. Size of data plays a very crucial role in determining value out of data. Also, whether a particular data can actually be considered as a Big Data or not, is dependent upon the volume of data. Hence, **'Volume'** is one characteristic which needs to be considered while dealing with Big Data.

**(ii) Variety –** The next aspect of Big Data is its **variety**.

Variety refers to heterogeneous sources and the nature of data, both structured and unstructured. During earlier days, spreadsheets and databases were the only sources of data considered by most of the applications. Nowadays, data in the form of emails, photos, videos, monitoring devices, PDFs, audio, etc. are also being considered in the analysis applications. This variety of unstructured data poses certain issues for storage, mining and analyzing data.

**(iii) Velocity –** The term **'velocity'** refers to the speed of generation of data. How fast the data is generated and processed to meet the demands, determines real potential in the data.

Big Data Velocity deals with the speed at which data flows in from sources like business processes, application logs, networks, and social media sites, sensors,[Mobile](https://www.guru99.com/mobile-testing.html)devices, etc. The flow of data is massive and continuous.

**(iv) Variability –** This refers to the inconsistency which can be shown by the data at times, thus hampering the process of being able to handle and manage the data effectively.

**------------------------------------------------------------------------------------------------**

**6. Examples of Big Data?**

**Examples of Big-Data:**

* The **New York Stock Exchange** generates about **one terabyte** of new trade data per day.

**Social Media**

* The statistic shows that **500+terabytes** of new data get ingested into the databases of social media site **Facebook**, every day. This data is mainly generated in terms of photo and video uploads, message exchanges, putting comments etc.
* A single **Jet engine** can generate **10+terabytes** of data in **30 minutes** of flight time. With many thousand flights per day, generation of data reaches up to many **Petabytes.**

**Healthcare**

* Big Data has already started to create a huge difference in the healthcare sector. With the help of predictive analytics, medical professionals and HCPs are now able to provide personalized healthcare services to individual patients. Apart from that, fitness wearables, telemedicine, remote monitoring – all powered by Big Data and AI – are helping change lives for the better.

**Academia**

* Big Data is also helping enhance education today. Education is no more limited to the physical bounds of the classroom – there are numerous online educational courses to learn from. Academic institutions are investing in digital courses powered by Big Data technologies to aid the all-round development of budding learners.

**Banking**

* The banking sector relies on Big Data for fraud detection. Big Data tools can efficiently detect fraudulent acts in real-time such as misuse of credit/debit cards, archival of inspection tracks, faulty alteration in customer stats, etc.

**Manufacturing**

* **According to TCS Global Trend Study, the most significant benefit of Big Data in manufacturing is improving the supply strategies and product quality.** In the manufacturing sector, Big data helps create a transparent infrastructure, thereby, predicting uncertainties and incompetencies that can affect the business adversely**.**

**IT**

* One of the largest users of Big Data, IT companies around the world are using Big Data to optimize their functioning, enhance employee productivity, and minimize risks in business operations. By combining Big Data technologies with ML and AI, the IT sector is continually powering innovation to find solutions even for the most complex of problems.

**Retail**

* Big Data has changed the way of working in traditional brick and mortar retail stores. Over the years, retailers have collected vast amounts of data from local demographic surveys, POS scanners, RFID, customer loyalty cards, store inventory, and so on. Now, they’ve started to leverage this data to create personalized customer experiences, boost sales, increase revenue, and deliver outstanding customer service.
* Retailers are even using smart sensors and Wi-Fi to track the movement of customers, the most frequented aisles, for how long customers linger in the aisles, among other things. They also gather social media data to understand what customers are saying about their brand, their services, and tweak their product design and marketing strategies accordingly.

**Transportation**

* Big Data Analytics holds immense value for the transportation industry. In countries across the world, both private and government-run transportation companies use Big Data technologies to optimize route planning, control traffic, manage road congestion, and improve services. Additionally, transportation services even use Big Data to revenue management, drive technological innovation, enhance logistics, and of course, to gain the upper hand in the market.

**------------------------------------------------------------------------------------------------**

**7. Types of Big Data?**

Following are the types of Big Data:

1. **Structured**
2. **Unstructured**
3. **Semi-structured**

### Structured

Structured is one of the types of big data and By structured data, we mean data that can be processed, stored, and retrieved in a fixed format. It refers to highly organized information that can be readily and seamlessly stored and accessed from a database by simple search engine algorithms. **For instance, the employee table in a company database will be structured as the employee details, their job positions, their salaries, etc.,** will be present in an organized manner.

**Examples Of Structured Data**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Employee\_ID** | **Employee\_Name** | **Gender** | **Department** | **Salary\_In\_lacs** |
| 2365 | Rajesh Kulkarni | Male | Finance | 650000 |
| 3398 | Pratibha Joshi | Female | Admin | 650000 |
| 7465 | Shushil Roy | Male | Admin | 500000 |
| 7500 | Shubhojit Das | Male | Finance | 500000 |
| 7699 | Priya Sane | Female | Finance | 550000 |

### Unstructured

Unstructured data refers to the data that lacks any specific form or structure whatsoever. This makes it very difficult and time-consuming to process and analyse unstructured data**. Email** is an example of unstructured data. A typical example of unstructured data is a heterogeneous data source containing a combination of **simple text files,** **images, videos and the output returned by 'Google Search' etc**. Structured and unstructured are two important types of big data.

**Semi-Structured**

Semi structured is the third type of big data. Semi-structured data pertains to the data containing both the formats mentioned above, that is, structured and unstructured data. We can see semi-structured data as a structured in form but it is actually not defined with e.g. a table definition in relational [DBMS](https://www.guru99.com/what-is-dbms.html). Example of semi-structured data is **a data represented in an XML file.**

Examples Of Semi-structured Data

Personal data stored in an XML file-

<rec><name>Prashant Rao</name><sex>Male</sex><age>35</age></rec>

<rec><name>Seema R.</name><sex>Female</sex><age>41</age></rec>

<rec><name>Satish Mane</name><sex>Male</sex><age>29</age></rec>

<rec><name>Subrato Roy</name><sex>Male</sex><age>26</age></rec>

<rec><name>Jeremiah J.</name><sex>Male</sex><age>35</age></rec>

**------------------------------------------------------------------------------------------**

**8. Benefits Of Big Data Processing.**

**Benefits Of Big Data Processing.**

* Ability to process Big Data brings in multiple benefits, such as-
  + Businesses can utilize outside intelligence while taking decisions
* Access to social data from search engines and sites like facebook, twitter are enabling organizations to fine tune their business strategies.
  + Improved customer service
* Traditional customer feedback systems are getting replaced by new systems designed with Big Data technologies. In these new systems, Big Data and natural language processing technologies are being used to read and evaluate consumer responses.
  + Early identification of risk to the product/services, if any
  + Better operational efficiency
* Identifying the root causes of failures and issues in real time
* Fully understanding the potential of data-driven marketing
* Generating customer offers based on their buying habits
* Improving [customer engagement](http://www.entrepreneur.com/article/229311) and increasing [customer loyalty](https://www.smartdatacollective.com/using-big-data-analytics-to-drive-customer-loyalty-business/)
* Re-evaluating risk portfolios quickly
* Personalizing the [customer experience](http://thenextweb.com/insider/2015/03/29/4-ways-to-improve-customer-experience-with-data/)
* Adding value to online and offline customer interactions
* Big Data technologies can be used for creating a staging area or landing zone for new data before identifying what data should be moved to the [data warehouse](https://www.guru99.com/data-warehousing.html). In addition, such integration of Big Data technologies and data warehouse helps an organization to offload infrequently accessed data.

**------------------------------------------------------------------------------------------------**

**9.What is NoSQL? Explain its features and application in brief?**

**NoSQL** Database is a non-relational Data Management System, that does not require a fixed schema. It avoids joins, and is easy to scale. The major purpose of using a NoSQL database is for distributed data stores with humongous data storage needs. NoSQL is used for Big data and real-time web apps. For example, companies like Twitter, Facebook and Google collect terabytes of user data every single day.

**NoSQL database** stands for "Not Only SQL" or "Not SQL."

**Features of NoSQL**

**Non-relational**

* NoSQL databases never follow the [relational model](https://www.guru99.com/relational-data-model-dbms.html)
* Never provide tables with flat fixed-column records
* Work with self-contained aggregates or BLOBs
* Doesn't require object-relational mapping and data normalization
* No complex features like query languages, query planners,referential integrity joins, ACID

**Schema-free**

* NoSQL databases are either schema-free or have relaxed schemas
* Do not require any sort of definition of the schema of the data
* Offers heterogeneous structures of data in the same domain

**Simple API**

* Offers easy to use interfaces for storage and querying data provided
* APIs allow low-level data manipulation & selection methods
* Text-based protocols mostly used with HTTP REST with JSON
* Mostly used no standard based NoSQL query language
* Web-enabled databases running as internet-facing services

**Distributed**

* Multiple NoSQL databases can be executed in a distributed fashion
* Offers auto-scaling and fail-over capabilities
* Often ACID concept can be sacrificed for scalability and throughput
* Mostly no synchronous replication between distributed nodes Asynchronous Multi-Master Replication, peer-to-peer, HDFS Replication
* Only providing eventual consistency
* Shared Nothing Architecture. This enables less coordination and higher distribution.

**Replication**

* The database allows automatic database replication.
* It is done to maintain availability in case of outages.
* Some sophisticated NoSQL databases provide automated recovery and are fully self- healing.
* To enable data localization and to withstand regional failures, it can distribute the database across multiple geographic regions.
* NoSQL does not require a separate application to implement replication.

**Dynamic schemas**

* NoSQL databases permit to insert the data without the predefined schema.
* Real-time application changes can be made easily without the need to worry about service interruptions.
* This makes development faster, more reliable and less time consuming for the database administrator.

**Auto-sharding**

* Horizontal scaling is done in a NoSQL database i.e. servers are added instead of increasing the capacity of a single server.
* It provides an auto-sharding feature i.e. it automatically spread data across various numbers of servers.
* Application need not be aware of the composition of the server pool.
* A load of data and queries are automatically balanced among the servers. If any server fails, then it is replaced quickly and transparently without disrupting the application.

**Applications:**

**E-Commerce**

* E-commerce companies use NoSQL for store huge volume of data and large amount of request from user.

**Social Gaming**

* Data-intensive applications such as social games which can grow users to millions. Such a growth in number of users as well as amount of data requires a database system which can store such data and can be scaled to incorporate number of growing users NOSQL is suitable for such applications.
* NOSQL has been used by some of the mobile gaming companies like, electronic arts, Zynga and Tencent.

**Ad Targeting**

* Displaying ads or offers on the current web page is a decision with direct income To determine what group of users to target, on web page where to display ads, the platforms gathers behavioral and demographic characteristics of users.
* A NoSQL database enables ad companies to track user details and also place the very quickly and increases the probability of clicks.
* AOL, Media mind and PayPal are some of the ad targeting companies which uses NoSQL

**Session Store**

* Managing session data using relational database is very difficult, especially in case where applications are grown very much.
* In such cases the right approach is to use a global session store, which manages session information for every user who visits the site.
* NOSQL is suitable for storing such web application session information very is large in size.
* Since the session data is unstructured in form, so it is easy to store it in schema less documents rather than in relation database record.

**User Profile Store**

* To enable online transactions, user preferences, authentication of user and more, it is required to store the user profile by web and mobile application.
* In recent time users of web and mobile application are grown very rapidly. The relational database could not handle such large volume of user profile data which growing rapidly, as it is limited to single server.
* Using NOSQL capacity can be easily increased by adding server, which makes scaling cost effective

**Content and Metadata Store**

* Many companies like publication houses require a place where they can store large amount of data, which include articles, digital content and e-books, in order to merge various tools for learning in single platform
* The applications which are content based, for such application metadata is very frequently accessed data which need less response times.
* For building applications based on content, use of NoSQL provide flexibility in faster access to data and to store different types of contents

**Mobile Applications**

* Since the smartphone users are increasing very rapidly, mobile applications face problems related to growth and volume.
* Using NoSQL database mobile application development can be started with small size and can be easily expanded as the number of user increases, which is very difficult if you consider relational databases.
* Since NoSQL database store the data in schema-less for the application developer can update the apps without having to do major modification in database.

The mobile app companies like Kobo and Playtika, uses NOSQL and serving millions of users across the world.

**Internet of Things**

* Today, billions of devices are connected to internet, such as smartphones, tablets, home appliances, systems installed in hospitals, cars and warehouses. For such devices large volume and variety of data is generated and keep on generating.
* Relational databases are unable to store such data. The NOSQL permits organizations to expand concurrent access to data from billions of devices and systems which are connected, store huge amount of data and meet the required performance.

**---------------------------------------------------------------------------------------**

**10.Compare NoSQL VS SQL.**

|  |  |  |
| --- | --- | --- |
| **Index** | **SQL** | **NoSQL** |
| 1) | Databases are categorized as Relational Database Management System (RDBMS). | NoSQL databases are categorized as Non-relational or distributed database system. |
| 2) | SQL databases have fixed or static or predefined schema. | NoSQL databases have dynamic schema. |
| 3) | SQL databases display data in form of tables so it is known as table-based database. | NoSQL databases display data as collection of key-value pair, documents, graph databases or wide-column stores. |
| 4) | SQL databases are vertically scalable. | NoSQL databases are horizontally scalable. |
| 5) | SQL databases use a powerful language "Structured Query Language" to define and manipulate the data. | In NoSQL databases, collection of documents are used to query the data. It is also called unstructured query language. It varies from database to database. |
| 6) | SQL databases are best suited for complex queries. | NoSQL databases are not so good for complex queries because these are not as powerful as SQL queries. |
| 7) | SQL databases are not best suited for hierarchical data storage. | NoSQL databases are best suited for hierarchical data storage. |
| 8) | MySQL, Oracle, Sqlite, PostgreSQL and MS-SQL etc. are the example of SQL database. | MongoDB, BigTable, Redis, RavenDB, Cassandra, Hbase, Neo4j, CouchDB etc. are the example of nosql database |

**--------------------------------------------------------------------------------------------**

**11.Which are types of NoSQL database? Enlist NoSQL based database?**

**NoSQL Databases** are mainly categorized into four types: Key-value pair, Column-oriented, Graph-based and Document-oriented. Every category has its unique attributes and limitations. None of the above-specified database is better to solve all the problems. Users should select the database based on their product needs.

Types of NoSQL Databases:

* Key-value Pair Based
* Column-oriented Graph
* Graphs based
* Document-oriented

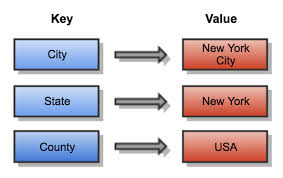
**Key-value Pair Based**

Data is stored in key/value pairs. It is designed in such a way to handle lots of data and heavy load.

Key-value pair storage databases store data as a hash table where each key is unique, and the value can be a JSON, BLOB(Binary Large Objects), string, etc.

It is one of the most basic NoSQL database example. This kind of NoSQL database is used as a collection, dictionaries, associative arrays, etc. Key value stores help the developer to store schema-less data. They work best for shopping cart contents.

**Redis, Dynamo, Riak** are some NoSQL examples of key-value store DataBases. They are all based on Amazon's Dynamo paper.



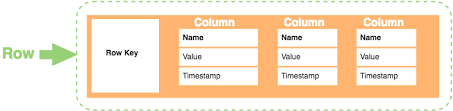
**Column-Based**

Column-oriented databases work on columns and are based on BigTable paper by Google. Every column is treated separately. Values of single column databases are stored contiguously.

They deliver high performance on aggregation queries like SUM, COUNT, AVG, MIN etc. as the data is readily available in a column.

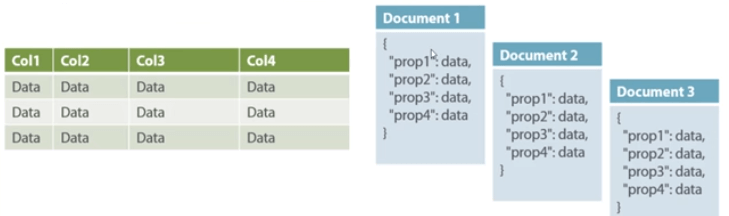
Column-based NoSQL databases are widely used to manage data warehouses, [business intelligence](https://www.guru99.com/business-intelligence-definition-example.html), CRM, Library card catalogs,

**HBase, Cassandra, Hypertable** are NoSQL query examples of column based database.



**Document-oriented**

Document-Oriented NoSQL DB stores and retrieves data as a key value pair but the value part is stored as a document. The document is stored in JSON or XML formats. The value is understood by the DB and can be queried.

[](https://www.guru99.com/images/1/101818_0537_NoSQLTutori8.png)

Relational Vs. Document

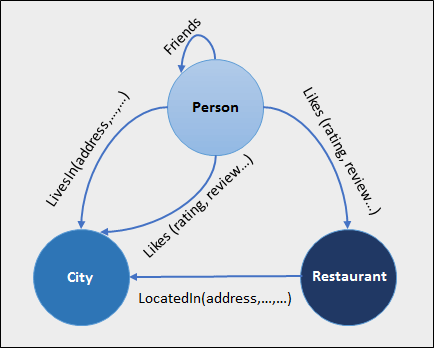
In this diagram on your left you can see we have rows and columns, and in the right, we have a document database which has a similar structure to JSON. Now for the relational database, you have to know what columns you have and so on. However, for a document database, you have data store like JSON object. You do not require to define which make it flexible.

The document type is mostly used for CMS systems, blogging platforms, real-time analytics & e-commerce applications. It should not use for complex transactions which require multiple operations or queries against varying aggregate structures.

**Amazon SimpleDB, CouchDB, MongoDB, Riak, Lotus Notes, MongoDB,** are popular Document originated DBMS systems.

**Graph-Based**

A graph type database stores entities as well the relations amongst those entities. The entity is stored as a node with the relationship as edges. An edge gives a relationship between nodes. Every node and edge has a unique identifier.

[](https://www.guru99.com/images/1/101818_0537_NoSQLTutori9.png)

Compared to a relational database where tables are loosely connected, a Graph database is a multi-relational in nature. Traversing relationship is fast as they are already captured into the DB, and there is no need to calculate them.

Graph base database mostly used for social networks, logistics, spatial data.

**Neo4J, Infinite Graph, OrientDB, FlockDB** are some popular graph-based databases.

**NoSQL Databases:**

MongoDB, CouchDB, CouchBase, Cassandra, HBase, Redis, Riak, Neo4J are the popular **NoSQL databases examples**.

MongoDB, CouchDB, Couchbase , Amazon SimpleDB, Riak, Lotus Notes are **document-oriented** **NoSQL databases.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MONGODB

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1: Explain the features of MONGODB.

a. Schema-less Database: A Schema-less database means one collection can hold different types of documents in it.

It is not necessary that the one document is similar to another document like in the relational databases.

b. Document Oriented: In MongoDB, all the data stored in the documents instead of tables like in RDBMS.

In these documents, the data is stored in fields(key-value pair) instead of rows and columns which make the data much more flexible in comparison to RDBMS.

And each document contains its unique object id.

c. Indexing: In MongoDB database, every field in the documents is indexed with primary and secondary indices this makes easier and takes less time to get or search data from the pool of the data.

d. Scalabiltiy: MongoDB provides horizontal scalability with the help of sharding.

Sharding means to distribute data on multiple servers, here a large amount of data is partitioned into data chunks using the shard key, and these data chunks are evenly distributed across shards that reside across many physical servers. It will also add new machines to a running database.

e. Replication: MongoDB provides high availability and redundancy with the help of replication, it creates multiple copies of the data and sends these copies to a different server so that if one server fails, then the data is retrieved from another server.

f. Aggregation: It allows to perform operations on the grouped data and get a single result or computed result.

It provides three different aggregations i.e, aggregation pipeline, map-reduce function, and single-purpose aggregation methods.

g. High Performance: The performance of MongoDB is very high and data persistence as compared to another database due to its features like scalability, indexing, replication, etc.

================================================================

2. Explain CRUD in MongoDB with syntax & suitable example.

MongoDB CURD Operations: CURD is Create Update, Read and Delete.

a. Create Operations : Create or insert operations add new documents to a collection. If the collection does not currently exist, insert operations will create the collection.

Eg: db.collection.insert();

> db.Film.insert({"Film Id":106,"Title of Film":"Article15","Year of Release":2019,"genre":["Crime","Action"],"actors":[{"fname":"Ayushmann","lname":"Khurrana"},{"fname":"Isha","lname":"Talwar"}],"directors":[{"fname":" Anubhav","lname":"Sinha"}],"release details":[{"places of release":["Mumbai","Pune"]},{"dates":["11/11/2019","12/12/2019"]},{"rating of film":"A"}]})

b. Update Operations : Update operations modify existing documents in a collection. MongoDB provides the following methods to update documents of a collection:

db.collection.replaceOne()

db.collection.updateMany()

db.collection.updateOne()

c. Read Operations : Read operations retrieve documents from a collection; i.e. query a collection for documents.

MongoDB provides the following methods to read documents from a collection:

Eg: db.collection.find()

db.Film.find();

d. Delete Operations : Delete operations remove documents from a collection. MongoDB provides the following methods to delete documents of a collection:

Eg : db.collection.deleteOne()

db.collection.deleteMany()

================================================================

3. Explain sorting in MongoDB with suitable example.

In MongoDB, sorting is done by the sort() method. The sort() method consists of two basic building blocks.

These building blocks are fields to be sorted and the sort order.

The sorting order in MongoDB is defined by either a one (1) or a minus (-1).

Here the positive one represents the ascending order, while the negative one represents the descending order.

Eg: db.collection\_name.find().sort({field\_name: sort order})

db.Film.find().sort({"Title of Film":1,"Year of Release":-1}).pretty()

================================================================

4. Explain skip() and limit() in MongoDB with suitable example.

The limit() function in MongoDB is used to specify the maximum number of results to be returned.

Only one parameter is required for this function to return the number of the desired result.

Eg : > db.Film.find({"actors.fname":"Madhuri"},{\_id:0,"Title of Film":1}).sort({\_id:-1}).limit(2)

The skip() method is used for skipping the given number of documents in the Query result.

Eg: > db.Film.find({"actors.fname":"Madhuri"},{\_id:0,"Title of Film":1}).sort({\_id:-1}).limit(2).skip(1);

================================================================

5. Explain Aggregation in MONGODB with syntax & example.

Aggregation basically groups the data from multiple documents and operates in many ways on those grouped data in order to return one combined result.

Eg: db.collection\_name.aggregate(aggregate\_operation);

Different expressions used by Aggregate function :

$sum : Summates the defined values from all the documents in a collection.

$avg : Calculates the average values from all the documents in a collection.

$min : Return the minimum of all values of documents in a collection.

$max : Return the maximum of all values of documents in a collection.

$push : Inserts values to an array in the resulting document.

$first : Returns the first document from the source document.

$last : Returns the last document from the source document.

================================================================

6. How regex can be used in MongoDB?

Provides regular expression capabilities for pattern matching strings in queries.

MongoDB uses PCRE(Perl compatible regular expressions).

The following <options> are available for use with regular expression.

a) i: Case insensitivity to match upper and lower cases.

Eg : db.customer.find( { name: { $regex: /^ABC/i } } )

b) m: For patterns that include anchors (i.e. ^ for the start, $ for the end),

match at the beginning or end of each line for strings with multiline values.

Eg : db.customer.find( { item: { $regex: /^S/, $options: 'm' } } )

c) x: “Extended” capability to ignore all white space characters in the $regex pattern

unless escaped or included in a character class.

================================================================

7. Explain the usage of Indexing in MONGODB.

An index in MongoDB is a special data structure that holds the data of few fields of documents on which the index is created. Indexes improve the speed of search operations in database because instead of searching the whole document, the search is performed on the indexes that holds only few .

syntax:

db.collection\_name.createIndex({field\_name: 1 or -1})

getIndexes() method to find all the indexes created on a collection.

syntax:

db.collection\_name.getIndexes()

dropIndex() method is used for dropping the index.

syntax:

db.collection\_name.dropIndex({index\_name: 1})

================================================================

8. How cursor can be used in MongoDB?

The Cursor is a MongoDB Collection of the document which is returned upon the find method execution.

By default, it is automatically executed as a loop.

Cursor Method:

cursor.count() : Modifies the cursor to return the number of documents in the result set rather than the documents themselves.

cursor.forEach() : Applies a JavaScript function for every document in a cursor.

cursor.hasNext() : Returns true if the cursor has documents and can be iterated.

cursor.limit() : Constrains the size of a cursor’s result set.

cursor.map() : Applies a function to each document in a cursor and collects the return values in an array.

cursor.max() : Specifies an exclusive upper index bound for a cursor.

cursor.next() : Returns the next document in a cursor.

cursor.pretty() : Configures the cursor to display results in an easy-to-read format.

cursor.skip() : Returns a cursor that begins returning results only after passing or skipping a number of documents.

cursor.sort() : Returns results ordered according to a sort specification.

================================================================

9. What is map reduce in MongoDB?

Map-reduce is a data processing paradigm for condensing large volumes of data into useful aggregated results.

In map-reduce operation, MongoDB applies the map phase to each input document.The map function emits key-value pairs.

For those keys that have multiple values, MongoDB applies the reduce phase, which collects and condenses the aggregated data.

Eg:

var m1 = function() {{emit(this.Transaction\_Detail.Item\_Name, this.Transaction\_Detail.Quantity)}; }

var r1 = function(name,quantity) {{return Array.sum(quantity)}}

db.Transaction.mapReduce(m1,r1,{out:"totalll"})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Distributed Database Concepts

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. What is Distributed Database? Write Chracteristic of Distributed database.

A Distributed Database Management System (DDBMS) contains a single logical

database that is divided into a number of fragments.

Every fragment gets stored on one or more computers under the control of a separate DBMS,

with the computers connected by a communications network.

Chracteristic of Distributed database:

1. logically interrealted

2. Data fregmenents

3. Fragment may get replicated

4. replicas & Fragments are allocated to site

5. connected network.

6. DBMS

7. Site Automous

================================================================

2. Explain the need of Distributed DBMS.

A distributed database is basically a database that is not limited to one system,

it is spread over different sites, i.e, on multiple computers or over a network of computers.

Distributed databases offer some key advantages over centralized databases.

a. Reliability – Building an infrastructure is similar to investing: diversify to reduce your chances of loss.

b. Security – We can give permissions to single sections of the overall database, for better internal and external protection.

c. Cost-effective – Bandwidth prices go down because users are accessing remote data less frequently.

d. Local access - if there is a failure in the umbrella network, you can still get access to your portion of the database.

e. Growth – If you add a new location to your business, it’s simple to create an additional node within the database, making distribution highly scalable.

f. Speed & resource efficiency – Most requests and other interactivity with the database are performed at a local level, also decreasing remote traffic.

1. Write advantages & disadvantages of DDBS.

Advantages

-------------

1. Reflections of organiasation structure

2. Imporved sharing,local autonomoy

3. Improved Availability of data

4. Reliability of data

5. Performance of backed is improved

6. Economics support

7. Modular growth

Disadvantages

--------------

1. Complexity(database design)

2. security

3. Lack of standards

4. What are the types of DDBS?

There are two types of DDBS:

1. Homogeneous

2. Heterogeneous

================================================================

5. Explain the architectures of DDBS.

Three architectures are used in distributed database systems:

a. Client/server Architecture : Client/server architectures are those in which a DBMS-related workload is split into two logical

components namely client and server, each of which typically executes on different systems.

Client is the user of the resource whereas the server is a provider of the resource.

b. Peer- to-Peer Architecture for DDBMS: In these systems, each peer acts both as a client and a server for imparting database services.

The peers share their resource with other peers and co-ordinate their activities.

c. Multi - DBMS Architectures : This is an integrated database system formed by a collection of two or more autonomous database systems.

================================================================

6. Which are the design stategies of DDBS?

DESIGN OF DDBMS(distributed databases)

Straegies:

1. Fragmentation

2.Allocation

3.Replication

================================================================

7. What is data allocation? Explain various allocation strategies.

Allocation : After fragmentation each partition or the fragmment is store at a site (how much )

Straegies Local of reference Reliability & Availability Performance Storage cost Communication Cost

centralized Lowest level Lowest Unsatisfactory Lowest Highest

Fragmented High High Satisfactory low Low

Complete Replication Highest Highest Best Highest Lowest

Selective Replication High High Satisfactory Low Low

================================================================

8. What is data replication? Explain various replication schemes.

Replication :Data replication is the process in which the data is copied at multiple locations (Different computers or servers) to improve the availability of data.

a. Partial replication : Partial replication means only some fragments are replicated from the database.

b. Full replication : In full replication scheme, the database is available to almost every location or user in communication network.

================================================================

9. What is Fragmentation? What are the types of fragmentation?

Fragmentation : Divide the whole db into subrelations

a. Horizontal

b.Vertical

c. Hybrid(based on predicate and projection)

================================================================

10. Explain Horizontal Fragmentation with suitable example.

Horizontal Fragmentation is splitting of tables horizontally that is into tuples or rows.

For example, a COMPANY table having 1000 records can be horizontally fragmented into ten fragments, each fragment having 100 unique records.

we can use SELECT statement, with the WHERE clause on a single attribute.

emp\_id | emp\_name | job\_name | manager\_id | hire\_date | salary | commission | dep\_id

--------+----------+-----------+------------+------------+---------+------------+--------

68319 | KAYLING | PRESIDENT | | 1991-11-18 | 6000.00 | | 1001

66928 | BLAZE | MANAGER | 68319 | 1991-05-01 | 2750.00 | | 3001

67832 | CLARE | MANAGER | 68319 | 1991-06-09 | 2550.00 | | 1001

65646 | JONAS | MANAGER | 68319 | 1991-04-02 | 2957.00 | | 2001

67858 | SCARLET | ANALYST | 65646 | 1997-04-19 | 3100.00 | | 2001

69062 | FRANK | ANALYST | 65646 | 1991-12-03 | 3100.00 | | 2001

63679 | SANDRINE | CLERK | 69062 | 1990-12-18 | 900.00 | | 2001

64989 | ADELYN | SALESMAN | 66928 | 1991-02-20 | 1700.00 | 400.00 | 3001

here horizontal fragmentation is :

a. 66928 | BLAZE | MANAGER | 68319 | 1991-05-01 | 2750.00 | | 3001

b. 64989 | ADELYN | SALESMAN | 66928 | 1991-02-20 | 1700.00 | 400.00 | 3001

every row is example of horizontal fragmentation.

================================================================

11. Explain Vertical Fragmentation with suitable example.

Vertical Fragmentation is fragmenting of table into columns known as set or site, where every site must have at least one column in common such as the primary key attribute column (so that when the fragmented sites when needed can again be formed to a whole.

For eg :

emp\_id | emp\_name | job\_name | manager\_id | hire\_date | salary | commission | dep\_id

--------+----------+-----------+------------+------------+---------+------------+--------

68319 | KAYLING | PRESIDENT | | 1991-11-18 | 6000.00 | | 1001

66928 | BLAZE | MANAGER | 68319 | 1991-05-01 | 2750.00 | | 3001

67832 | CLARE | MANAGER | 68319 | 1991-06-09 | 2550.00 | | 1001

65646 | JONAS | MANAGER | 68319 | 1991-04-02 | 2957.00 | | 2001

67858 | SCARLET | ANALYST | 65646 | 1997-04-19 | 3100.00 | | 2001

69062 | FRANK | ANALYST | 65646 | 1991-12-03 | 3100.00 | | 2001

63679 | SANDRINE | CLERK | 69062 | 1990-12-18 | 900.00 | | 2001

64989 | ADELYN | SALESMAN | 66928 | 1991-02-20 | 1700.00 | 400.00 | 3001

here vertical fragmentation is :

emp\_name | job\_name

-------- +----------+

66928 | BLAZE |

67832 | CLARE |

65646 | JONAS |

67858 | SCARLET |

69062 | FRANK |

63679 | SANDRINE |

64989 | ADELYN

================================================================

12. Explain Hybrid Fragmentation with suitable example.

In hybrid fragmentation, a combination of horizontal and vertical fragmentation techniques are used.

This is the most flexible fragmentation technique since it generates fragments with minimal extraneous information.

Eg:

emp\_id | emp\_name | job\_name | manager\_id | hire\_date | salary | commission | dep\_id

------+----------+-----------+------------+------------+---------+------------+--------

68319 | KAYLING | PRESIDENT | | 1991-11-18 | 6000.00 | | 1001

66928 | BLAZE | MANAGER | 68319 | 1991-05-01 | 2750.00 | | 3001

67832 | CLARE | MANAGER | 68319 | 1991-06-09 | 2550.00 | | 1001

65646 | JONAS | MANAGER | 68319 | 1991-04-02 | 2957.00 | | 2001

67858 | SCARLET | ANALYST | 65646 | 1997-04-19 | 3100.00 | | 2001

69062 | FRANK | ANALYST | 65646 | 1991-12-03 | 3100.00 | | 2001

63679 | SANDRINE | CLERK | 69062 | 1990-12-18 | 900.00 | | 2001

64989 | ADELYN | SALESMAN | 66928 | 1991-02-20 | 1700.00 | 400.00 | 3001

================================================================

================================================================

ASSIGNMENT 4

1. Define Scale up, Speed up in Parallel Database.

Ans:

**Scale up**

Scale up is the ability of an application to retain response time as the job size or the transaction volume increases by adding additional processors and disks. Scale up is calculated using the following formula:

Scale up = Volume m / Volume 1

In database applications, scale up can be either batch or transactional.

Batch scale up: Batch scale up, larger batch jobs can be supported without a loss of response time.

Transaction scale up: Transaction scale up, larger numbers of transactions can be supported without loss of response time.

In both cases, response time is maintained by the addition of more processors.

**Speed up**

Speed up is defined as the ratio between the runtime with one processor and the runtime using multiple processors. It measures the performance improvement gained using multiple processors instead of a single processor and is calculated using formula:

Speed up = Time1 / Time M

Time1 => Time1 is the time it takes to execute a task using only one processor.

Time M => Time M is the time it takes to execute that same task using M processors.

--------------------------------------------------------------------------------------------------------------------------------------

2. Goals of parallel DB.

Ans:

* Improve performance: The performance of the system can be improved by connecting multiple CPU and disks in parallel. Many small processors can also be connected in parallel.
* Improve availability of data: Data can be copied to multiple locations to improve the availability of data.
* Improve reliability: Reliability of system is improved with completeness, accuracy and availability of data.
* Provide distributed access of data: Companies having many branches in multiple cities can access data with the help of parallel database system.

--------------------------------------------------------------------------------------------------------------------------------------

3. Explain architecture of parallel DB.

Ans:

In parallel database architecture, there are multiple central processing units (CPUs) connected to a computer system. There are several architectural models for parallel machines. Three of the most prominent ones are listed below:

* Shared-memory multiple CPU.
* Shared-disk multiple CPU.
* Shared-nothing multiple CPU.

--------------------------------------------------------------------------------------------------------------------------------------

4. Shared memory system, Shared disk system, Shared nothing system.

Ans:

# Shared memory system

* Shared memory system uses multiple processors which is attached to a global shared memory via intercommunication channel or communication bus.
* Shared memory system have large amount of cache memory at each processors, so referencing of the shared memory is avoided.
* If a processor performs a write operation to memory location, the data should be updated or removed from that location.

# shared memory system

# Advantages of Shared memory system

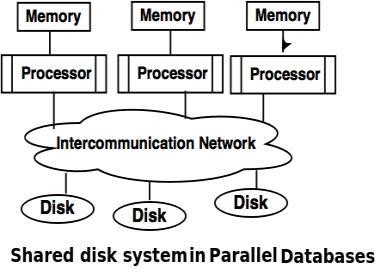
* Data is easily accessible to any processor.
* One processor can send message to other efficiently.

# Disadvantages of Shared memory system

* Waiting time of processors is increased due to more number of processors.

**Shared Disk System**

* Shared disk system uses multiple processors which are accessible to multiple disks via intercommunication channel and every processor has local memory.
* Each processor has its own memory so the data sharing is efficient.
* The system built around this system is called as clusters.



# Advantages of Shared Disk System

* Fault tolerance is achieved using shared disk system.

**Fault tolerance:** If a processor or its memory fails, the other processor can complete the task. This is called as fault tolerance.

# Disadvantage of Shared Disk System

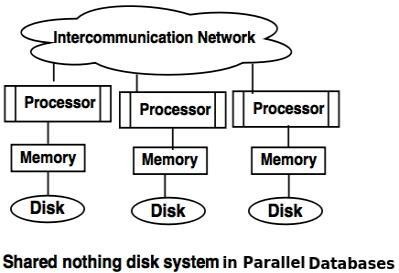
* Shared disk system has limited scalability as large amount of data travels through the interconnection channel.
* If more processors are added the existing processors are slowed down.

# Applications of Shared Disk System

Digital Equipment Corporation (DEC):DEC cluster running relational databases use the shared disk system and now owned by Oracle.

**Shared nothing disk system**

* Each processor in the shared nothing system has its own local memory and local disk.
* Processors can communicate with each other through intercommunication channel.
* Any processor can act as a server to serve the data which is stored on local disk.



# Advantages of Shared nothing disk system

* Number of processors and disk can be connected as per the requirement in share nothing disk system.
* Shared nothing disk system can support for many processor, which makes the system more scalable.

# Disadvantages of Shared nothing disk system

* Data partitioning is required in shared nothing disk system.
* Cost of communication for accessing local disk is much higher.

# Applications of Shared nothing disk system

* Tera data database machine.
* The Grace and Gamma research prototypes.

--------------------------------------------------------------------------------------------------------------------------------------

5. Explain various methods of achieving parallelism in Quries.

Ans:

**I/O Parallelism**

* Reduce the time required to retrieve relations from disk by partitioning
* The relation on multiple disk.

**Inter query Parallelism**

* Queries/transactions execute in parallel with one another.
* Increases transaction throughput; used primarily to scale up a transaction processing system to support a larger number of transactions per second.
* Easiest form of parallelism to support, particularly in a shared memory parallel database, because even sequential database systems support concurrent processing.
* More complicated to implement on shared-disk or shared-nothing architectures

**Intra query Parallelism**

* Execution of a single query in parallel on multiple processors/disks, important for speeding up long-running queries

.

**Intra operation Parallelism**

* Parallelize the execution of each individual operation in the query.

**Interoperation Parallelism**

* Execute the different operations in a query expression in parallel.

=================================================================