NOSQL

Classical Relational Database

- A database transaction, must be atomic, consistent, isolated and durable. As Discussed Below

- **Atomic** : A transaction is a logical unit of work which must be either completed with all of its data modifications, or none of them is performed.

- **Consistent** : At the end of the transaction, all data must be left in a consistent state.

Cont…

- **Isolated** : Modifications of data performed by a transaction must be independent of another transaction. Unless this happens, the outcome of a transaction may be erroneous.

- **Durable** : When the transaction is completed, effects of the modifications performed by the transaction must be permanent in the system.

- Often these four properties of a transaction are acronymed as **ACID**.

Distributed Systems

- A distributed system consists of multiple computers and software components that communicate through a computer network (a local network or by a wide area network).

- A distributed system can consist of any number of possible configurations, such as mainframes, workstations, personal computers, and so on.

- The computers interact with each other and share the resources of the system to achieve a common goal.

Advantages of Distributed Computing

- **Reliability (fault tolerance) :**
  The important advantage of distributed computing system is reliability. If some of the machines within the system crash, the rest of the computers remain unaffected and work does not stop.

- **Scalability :**
  In distributed computing the system can easily be expanded by adding more machines as needed.

- **Sharing of Resources :**
  Shared data is essential to many applications such as banking, reservation system. As data or resources are shared in distributed system, other resources can be also shared (e.g. expensive printers).

Cont…

- **Speed :**
  A distributed computing system can have more computing power and it's speed makes it different than other systems.
- **Open system :**
  As it is open system, every service is equally accessible to every client i.e. local or remote.
- **Performance :**
  The collection of processors in the system can provide higher performance (and better price/performance ratio) than a centralized computer.

Disadvantages of Distributed Computing

- **Software :**
  Less software support is the main disadvantage of distributed computing system.
- **Networking :**
  The network infrastructure can create several problems such as transmission problem, overloading, loss of messages.
- **Security :**
  Easy access in distributed computing system increases the risk of security and sharing of data generates the problem of data security

NOSQL

- NoSQL is a non-relational database management systems, different from traditional relational database management systems in some significant ways.
- It is designed for distributed data stores where very large scale of data storing needs (for example Google or Facebook which collects terabits of data every day for their users).
- These type of data storing may not require fixed schema, avoid join operations and typically scale horizontally.

Why NoSQL?

- In today's time data is becoming easier to access and capture through third parties such as Facebook, Google+ and others.
- Personal user information, social graphs, geo location data, user-generated content and machine logging data are just a few examples where the data has been increasing exponentially.
- To avail the above service properly, it is required to process huge amount of data.
- The evolution of NoSql databases is to handle these huge data properly.
-

Example 1

Social-network graph :

Each record: UserID1, UserID2

Separate records: UserID, first_name, last_name, age, gender,...

Task: Find all friends of friends of friends of ... friends of a given user.

Example 2

Wikipedia pages

Large collection of documents

Combination of structured and unstructured data

Task: Retrieve all pages regarding athletics of Summer Olympic before 1950.

RDBMS vs NoSQL

- **RDBMS**
  - Structured and organized data
  - Structured query language (SQL)
  - Data and its relationships are stored in separate tables.
  - Data Manipulation Language, Data Definition Language
  - Tight Consistency

RDBMS vs NOSQL

- **NoSQL**
  - Stands for Not Only SQL
  - No declarative query language
  - No predefined schema
  - Key-Value pair storage, Column Store, Document Store, Graph databases
  - Eventual consistency rather ACID property
  - Unstructured and unpredictable data
  - CAP Theorem
  - Prioritizes high performance, high availability and scalability
  - BASE Transaction

CAP Theorem (Brewer's Theorem)

- CAP theorem states that there are three basic requirements which exist in a special relation when designing applications for a distributed architecture.

CAP Stands for

- Consistency
- Availability
- Partition Tolerance

CAP Theorem Cont…

- **Consistency** - This means that the data in the database remains consistent after the execution of an operation. For example after an update operation all clients see the same data.
- **Availability** - This means that the system is always on (service guarantee availability), no downtime.
- **Partition Tolerance** - This means that the system continues to function even the communication among the servers is unreliable, i.e. the servers may be partitioned into multiple groups that cannot communicate with one another.

CAP Theorem Cont…

- CAP provides the basic requirements for a distributed system to follow 2 of the 3 requirements.

- Therefore all the current NoSQL database follow the different combinations of the C, A, P from the CAP theorem.

- Here is the brief description of three combinations CA, CP, AP :

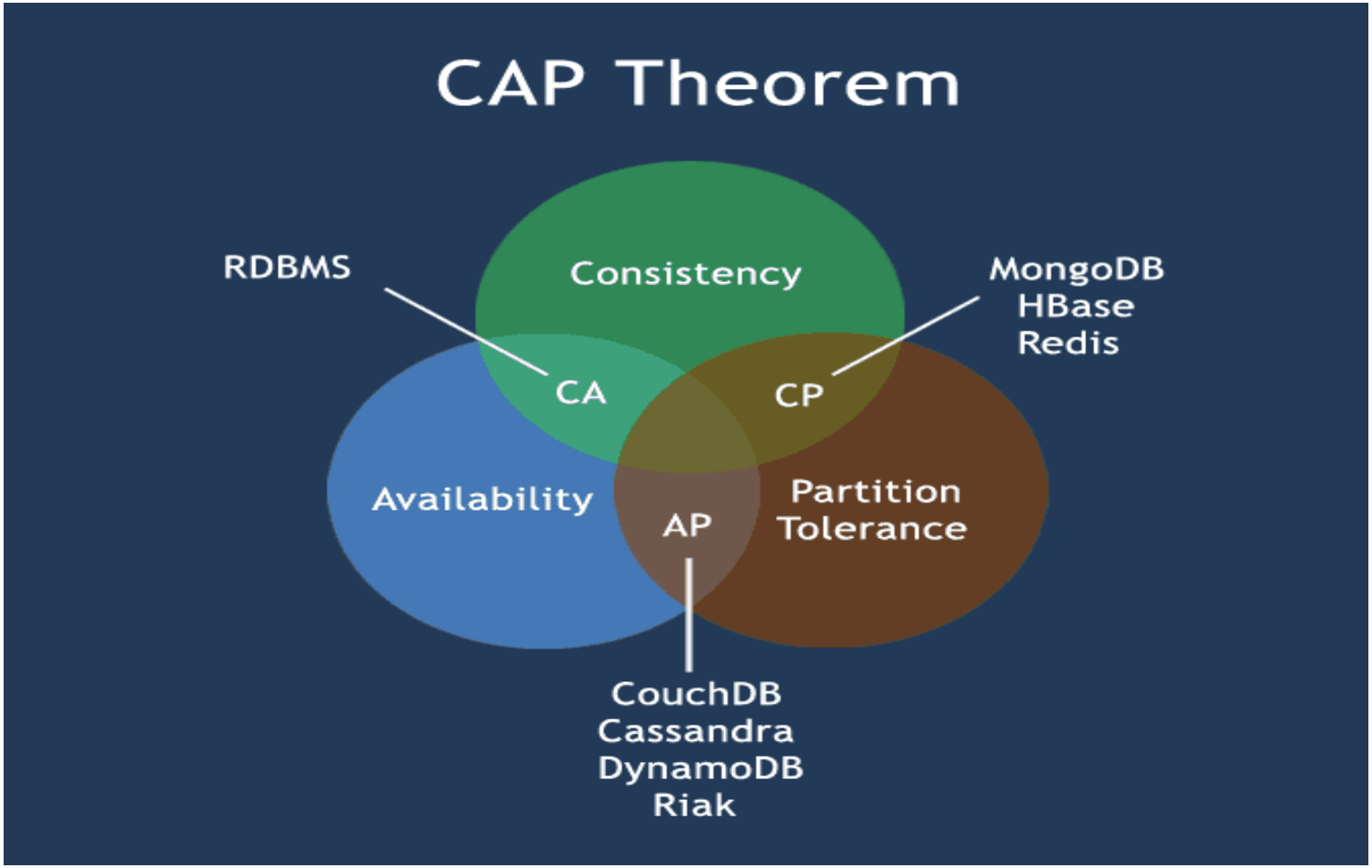
Cont…

CA - Single site cluster, therefore all nodes are always in contact. When a partition occurs, the system blocks.

CP - Some data may not be accessible, but the rest is still consistent/accurate.

AP - System is still available under partitioning, but some of the data returned may be inaccurate.

Cont...


CAP Theorem
RDBMS
Consistency
MongoDB
HBase
Redis
CA
CP
Availability
Partition
Tolerance
AP
CouchDB
Cassandra
DynamoDB
Riak

NoSQL pros/cons

- **Advantages** :
  - High scalability
  - Distributed Computing
  - Lower cost
  - Schema flexibility, semi-structure data
  - No complicated Relationships

Cont…

- **Disadvantages**
  - No standardization
  - Limited query capabilities (so far)
  - Eventual consistent is not intuitive to program for

The BASE

- The BASE acronym was defined by Eric Brewer, who is also known for formulating the CAP theorem.

- The CAP theorem states that a distributed computer system cannot guarantee all of the following three properties at the same time:
  - Consistency
  - Availability
  - Partition tolerance

Cont…

- ***Basic Availability:-***

  The database appears to work most of the time.

- ***Soft-state :-***

  Stores don’t have to be write-consistent, nor do different replicas have to be mutually consistent all the time.

- ***Eventual consistency***

  Stores exhibit consistency at some later point

ACID vs BASE

ACID	BASE
Atomic	**B**asically **A**vailable
Consistency	**S**oft state
Isolation	**E**ventual consistency
Durable	

NoSQL Categories

- There are four general types (most common categories) of NoSQL databases.
  - Key-value stores
  - Column-oriented
  - Graph
  - Document oriented

Key-value stores

- Key-value stores are most basic types of NoSQL databases.
- Designed to handle huge amounts of data.
- Based on Amazon's Dynamo paper.
- Key value stores allow developer to store schema-less data.
- In the key-value storage, database stores data as hash table where each key is unique and the value can be string, JSON, BLOB (Binary Large OBject) etc.
- A key may be strings, hashes, lists, sets, sorted sets and values are stored against these keys.

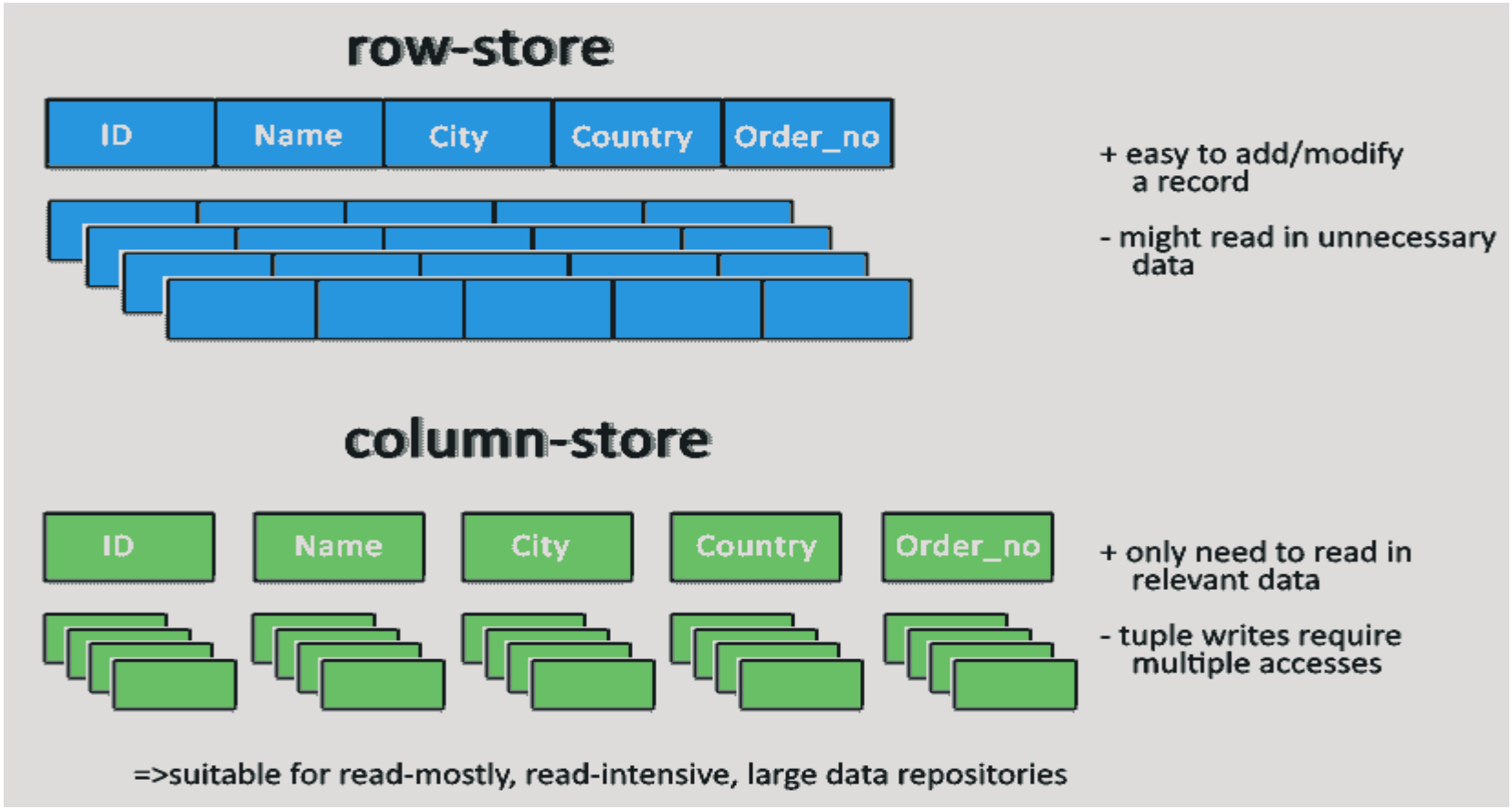
Cont…

- For example a key-value pair might consist of a key like "Name" that is associated with a value like "Robin".
- Key-Value stores can be used as collections, dictionaries, associative arrays etc.
- Key-Value stores follow the 'Availability' and 'Partition' aspects of CAP theorem.
- Key-Values stores would work well for shopping cart contents, or individual values like color schemes, a landing page URI, or a default account number.

Cont…

- **Example of Key-value store Data Base**

  Redis, Dynamo, Riak. etc.



Column-oriented databases

- Column-oriented databases primarily work on columns and every column is treated individually.
- Values of a single column are stored contiguously.
- Column stores data in column specific files.
- In Column stores, query processors work on columns too.
- All data within each column datafile have the same type which makes it ideal for compression.

Cont…

- Column stores can improve the performance of queries as it can access specific column data.
- High performance on aggregation queries (e.g. COUNT, SUM, AVG, MIN, MAX).
- Works on data warehouses and business intelligence, customer relationship management (CRM), Library card catalogs etc.

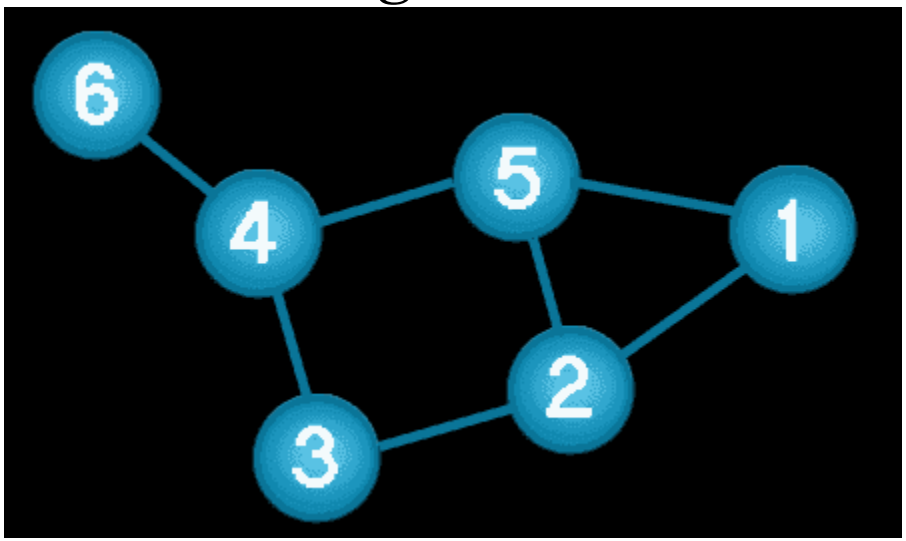
Cont…

- **Example of Column-oriented databases**

  BigTable, Cassandra, SimpleDB etc.

Graph databases

- A graph data structure consists of a finite (and possibly mutable) set of ordered pairs, called edges or arcs, of certain entities called nodes or vertices.
- The following picture presents a labeled graph of 6 vertices and 7 edges.



What is a Graph Databases?

- A graph database stores data in a graph.
- A graph database is a collection of nodes and edges
- Each node represents an entity (such as a student or business) and each edge represents a connection or relationship between two nodes.
- Every node and edge are defined by a unique identifier.
- Each node knows its adjacent nodes.
- As the number of nodes increases, the cost of a local step (or hop) remains the same.
- Index for lookups.

Comparison between the classic relational model and the graph model

Relational model	**Graph model**
Tables	Vertices and Edges set
Rows	Vertices
Columns	Key/value pairs
Joins	Edges

Example of Graph databases : OrientDB, Neo4J, Titan.etc.

Document Oriented databases

- A collection of documents
- Data in this model is stored inside documents.
- A document is a key value collection where the key allows access to its value.
- Documents are not typically forced to have a schema and therefore are flexible and easy to change.
- Documents are stored into collections in order to group different kinds of data.
- Documents can contain many different key-value pairs, or key-array pairs, or even nested documents.

Comparison between the classic relational model and the document model :

Relational model	**Document model**
Tables	Collections
Rows	Documents
Columns	Key/value pairs
Joins	not available

Example of Document Oriented databases : Mongo DB, Couch DB etc.