**Introduction :**

Apache Cassandra is a highly scalable, high-performance distributed database designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. It is a type of NoSQL database. Let us first understand what a NoSQL database does.

**What is NoSQL?**

A NoSQL database (Not Only SQL) is a database that provides a mechanism to store and retrieve data other than the tabular relations used in relational databases. These databases are schema-free, support easy replication, have simple API, eventually consistent, and can handle huge amounts of data.

The primary objective of a NoSQL database is to have:

* Simplicity of design
* Horizontal scaling is possible
* finer control over availability

**Types of NoSQL Databases:**

* Key-Value store
* Document Databases
* Column Databases
* Graph Databases

**NoSQL vs. Relational Databases:**

**NoSQL:**

* Supports very simple query language.
* No fixed schema.
* No “ACID”, it is only “eventually consistent”.
* Does not support transactions.

**Relational Databases:**

* Supports powerful query language.
* It has a fixed schema.
* Follows “ACID”.
* Supports transactions.

**NP:** *There is no winner in this comparison, it really depends on the kind of application you are building and the data you are storing.*

**What is Apache Cassandra?**

* Apache Cassandra is an open source, distributed and decentralized/distributed storage system (database), for handling huge amounts of structured data spread out across the world. It provides highly available service with no single point of failure.
* Cassandra is a NoSQL database and can be described as a mix between a “*Key-Value Store”* and *“Column-Orientated”* database.

**The notable points of Apache Cassandra:**

* It is scalable, fault-tolerant, and consistent.
* It is a column-oriented database.
* Its distribution design is based on Amazon’s Dynamo and its data model on Google’s Bigtable.
* Created at Facebook, it differs sharply from relational database management systems.
* Cassandra implements a Dynamo-style replication model with no single point of failure, but adds a more powerful “column family” data model.
* Cassandra is being used by some of the biggest companies such as Facebook, Twitter, Cisco, Rackspace, ebay, Twitter, Netflix, and more.

**Features of Cassandra:**

* **Scalability** - New machines can easily be added with no downtime or interruption.
* **Decentralized** -No master & no single point of failure. Data is distributed across the cluster
* **Fast linear-scale performance** - Cassandra is linearly scalable, i.e., it increases your throughput as you increase the number of nodes in the cluster. Therefore it maintains a quick response time.
* **Flexible data storage** - Cassandra accommodates all possible data formats including: structured, semi-structured, and unstructured. It can dynamically accommodate changes to your data structures according to your need.
* **Easy data distribution** - Cassandra provides the flexibility to distribute data where you need by replicating data across multiple data centers.
* **Transaction support** - Cassandra supports properties like Atomicity, Consistency, Isolation, and Durability (ACID).
* **Cassandra Query Language (CQL)** – An SQL-like alternative

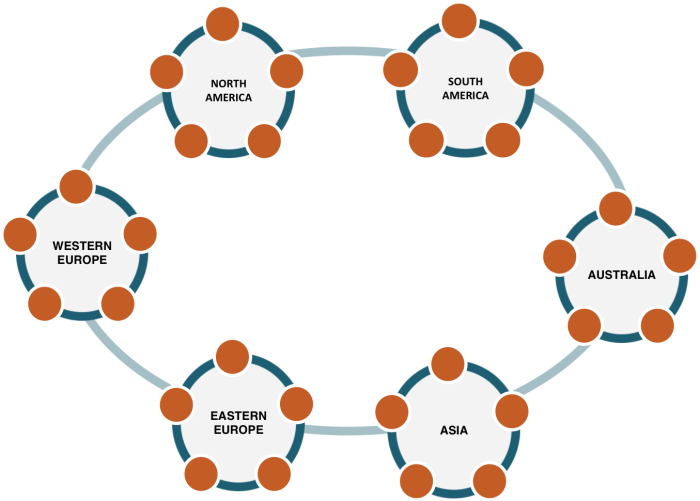
**Cassandra Query Language (CQL):**

* *CQL* is very similar to *SQL* in terms of syntax and commands*.*
* Statements directly change data and/or change the way data is stored.
* All statements end with semi-colon “**;**”.

SELECT \* FROM tablename;

**Cassandra Architecture:**

* Built with the understanding that hardware & software failures can happen.
* Per to Per Architecture.
* All the nodes are the same.
* Read/Write anywhere.
* Gossip protocol.
* Commit Log Captures All Activity.



**How data is stored?**

* **Keyspaces** – A Keyspace is similar to a **“Schema”**
* **Columns** – A “**Column Family”** is similar to a **“Table”** in a relational database except much more flexible
* Cassandra encourages **“Denormalization”** as oppose to relational **“Normalization”**
* **Columns** can be created on the fly.