

## **Experiment 11**

**Name:** Ayush Jain

**SAP ID:** 60004200132

**Batch:** B1

**Subject:** DBMS

---

**Aim:** Case study on recent DBMS and one real life application where one of these DBMS used.

**Theory:**

1) Tabular comparison of different database systems.

<b><u>Name of Database</u></b>	<b><u>Database Type</u></b>	<b><u>Licensing</u></b>	<b><u>Scalability</u></b>	<b><u>Data Types Supported</u></b>
MySQL	SQL	GNU Generally public license	Vertical, complex	Structured, Semi - Structured
Oracle	Multi-Model, SQL	Proprietary	Vertical	Structured, Semi – Structured, Unstructured
MongoDB	NoSQL, document - oriented	SSPL	Horizontal	Structured, Semi – Structured, Unstructured
Firebase	NoSQL, real – time database	Open - Source	Horizontal	Structured, Semi – Structured, Unstructured
Maria DB	SQL	GNU Generally public license	Vertical	Structured, Semi - Structured
PostgreSQL	Object – Relational, SQL	Open Source	Vertical	Structured, Semi – Structured, Unstructured
Redis	NoSQL, key - value	Open Source, BSD 3 - clause	Horizontal	Structured, Semi – Structured, Unstructured

## 2) Describe DBMS used by any one organization solving real life problem.

Netflix is the big Kahuna of a Web media businesses, with 33 million subscribers in more than 40 countries. As Netflix's "watch now" streaming service has grown, the company has had to rethink its data and storage strategies to cope with ballooning workloads managed in the cloud. Today, the company is nearly complete in its migration from Oracle to the NoSQL database Cassandra, improving availability and essentially eliminating downtime incurred by database schema changes.

**Cassandra** is a free and open-source, distributed, wide-column store, NoSQL database management system designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. Cassandra offers support for clusters spanning multiple data center, with asynchronous masterless replication allowing low latency operations for all clients. Cassandra was designed to implement a combination of Amazon's Dynamo distributed storage and replication techniques combined with Google's Bigtable data and storage engine model.

### Main features of Cassandra

---

#### **Distributed -**

Every node in the cluster has the same role. There is no single point of failure. Data is distributed across the cluster (so each node contains different data), but there is no master as every node can service any request.

#### **Supports replication and multi data center replication -**

Replication strategies are configurable.<sup>[17]</sup> Cassandra is designed as a distributed system, for deployment of large numbers of nodes across multiple data centers. Key features of Cassandra's distributed architecture are specifically tailored for multiple-data center deployment, for redundancy, for failover and disaster recovery.

#### **Scalability -**

Designed to have read and write throughput both increase linearly as new machines are added, with the aim of no downtime or interruption to applications.

#### **Fault-tolerant -**

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

#### **Tunable consistency -**

Cassandra is typically classified as an AP system, meaning that availability and partition tolerance are generally considered to be more important than consistency in Cassandra.<sup>[18]</sup> Writes and reads offer a tunable level of consistency, all the way from "writes never fail" to "block for all replicas to be readable", with the quorum level in the middle.<sup>[19]</sup>

#### **MapReduce support -**

Cassandra has Hadoop integration, with MapReduce support. There is support also for Apache Pig and Apache Hive.<sup>[20]</sup>

#### **Query language -**

Cassandra introduced the Cassandra Query Language (CQL). CQL is a simple interface for accessing Cassandra, as an alternative to the traditional Structured Query Language (SQL).

#### **Eventual consistency -**

Cassandra manages eventual consistency of reads, upserts and deletes through Tombstones.

**Conclusion:** Hence, we learnt about different databases and understood how they are being used by different companies to solve many real – life problems.