Relational Databases (RDBMS)

by Kelechi Precious Nwachukwu

- Definition of RDBMS
- The 3 well known RDBMS (MySQL, PostgreSQL, SQL SERVER) and their functionalities
- A comparison between the three RDBMS

Definition of RDBMS

RDBMS stands for “Relational Database Management System” which is designed specifically for relational databases.

A database is relational if it stores data in a structured format, using rows and columns. This makes it easy to locate and access specific values within the database.

It is “relational” because the values within each table are related to each other which let tables also to be related to other tables.

The Most well known RDBMS are : MySQL, PostgreSQL and Microsoft SQL server

MySQL

- MySQL, pronounced either « My S-Q-L » or « My Sequel » is an open source RDBM. It is based on the structure query language, which is used for adding, removing and modifying information in the database.
- MySQL can be used for a variety of applications, but is most commonly found on web servers. A website that uses MySQL may include web pages that access information from a database (those pages are referred to as “dynamic”.
- MySQL is open-source and free software under the GNU license supported by Oracle.

PostgreSQL

PostgreSQL features transactions with Atomicity, Consistency, Isolation, Durability (ACID) properties.

It is designed to handle a range of workloads, from single machines to data warehouses or Web services with many concurrent users.

PostgreSQL comes with many features aimed to help developers build applications, administrators to protect data integrity, and build fault-tolerant environments, also help you manage your data no matter how big or small the dataset. In addition to being free and open source, PostgreSQL is highly extensible. For example, you can define your own data types, build out custom functions, even write code from different programming languages without recompiling your database

SQL Server

SQL Server is a RDBMS created by Microsoft. It is used as a central location to save and obtain data needed for applications. It uses SQL (structured query language) for queries that store or retrieve data Microsoft SQL Server also allows user-defined composite types (UDTs) to be defined and used. It also makes server statistics available as virtual tables and views (called Dynamic Management Views or DMVs). In addition to tables, a database can also contain other objects including views, stored procedures, indexes and constraints, along with a transaction log.

Comparison between the RDBMS

SQL and No-SQL database have differences in their structure. However, when we are looking into several SQL solutions, the criteria are a lot more distorted. Now we will consider the aspects more precisely and analyze the underlying functionality, by taking a look at their comparisons in the tables below:

	MySQL	PostgreSQL	SQL Server
Price	Has additional paid tools; the core functionality can be accessed for free.	open-source	The database has a free edition for developers and small business but only supports 1 processor and 1GB memory.
Language	C++	C	C++
Used by companies	Google, Udemy, Netflix, AirBnb, Amazon	Apple, Skype, Cisco	Bank of America, UPS
Partitioning	Allows partitioning database with hashing functions in order to distribute data among several nodes.	Allows making LIST and RANGE partitions where the index of a partition is created manually.	Provides access to RANGE partitioning, where the partition is assigned to all values that fall into a particular range.

Defragmentation	Offers several approach to defragmentation - during backup, index creation, and with an OPTIMIZE Table command	Allows scanning the entire tables of a data layer to find empty rows and delete the unnecessary elements.	Offers an efficient garbage collector that doesn't create more than 15-20% of overhead.
JSON Support	Supports JSON files but doesn't allow indexing them.	Supports JSON files, as well as their indexing and partial updates.	Provides full support of JSON documents
Data queries	Offers a scalable buffer pool - developers can set up the size of the cache according to the workload	Isolates processes even further than MySQL by treating them as a separate OS process.	Also uses a buffer pool, and just like in MySQL, it can be limited or increased according to processing needs.

Indexes	Organized indexes in tables and clusters. Developers can automatically locate and update indexes in their databases	Support index-based table organization, but the early version don’t include automated index updates	Offers rich automated functionality for index management. They can organize in clusters and maintain the correct row order without manual involvement.
Temporary Tables	Offers limited functionality for temporary tables. Developers cannot set variables or create global templates.	Offers a lot more functionality when it comes to temporary content. You divide temporary tables into local and global, then configure with flexible variables.	Offers rich functionality for temporary table management. You can create local and global temporary tables, as well as oversee and create variables.

Conclusion

The choice between the three most popular databases ultimately boils down to the comparison of the functionality, use cases, and ecosystems. Companies that prioritize flexibility, cost-efficiency, and innovation usually choose open-source solutions. They can be integrated with multiple free add-ons, have active user communities, and are continuously updated.