# Relational Database Management System (RDBMS)



## What is a Relational Database Management System (RDBMS)?

A relational database management system (RDBMS) is a program that allows you to create, update, and administer a relational database.RDBMS is based on relational model as invented by E.F.Codd of IBM’s San Jose Research Laboratory.

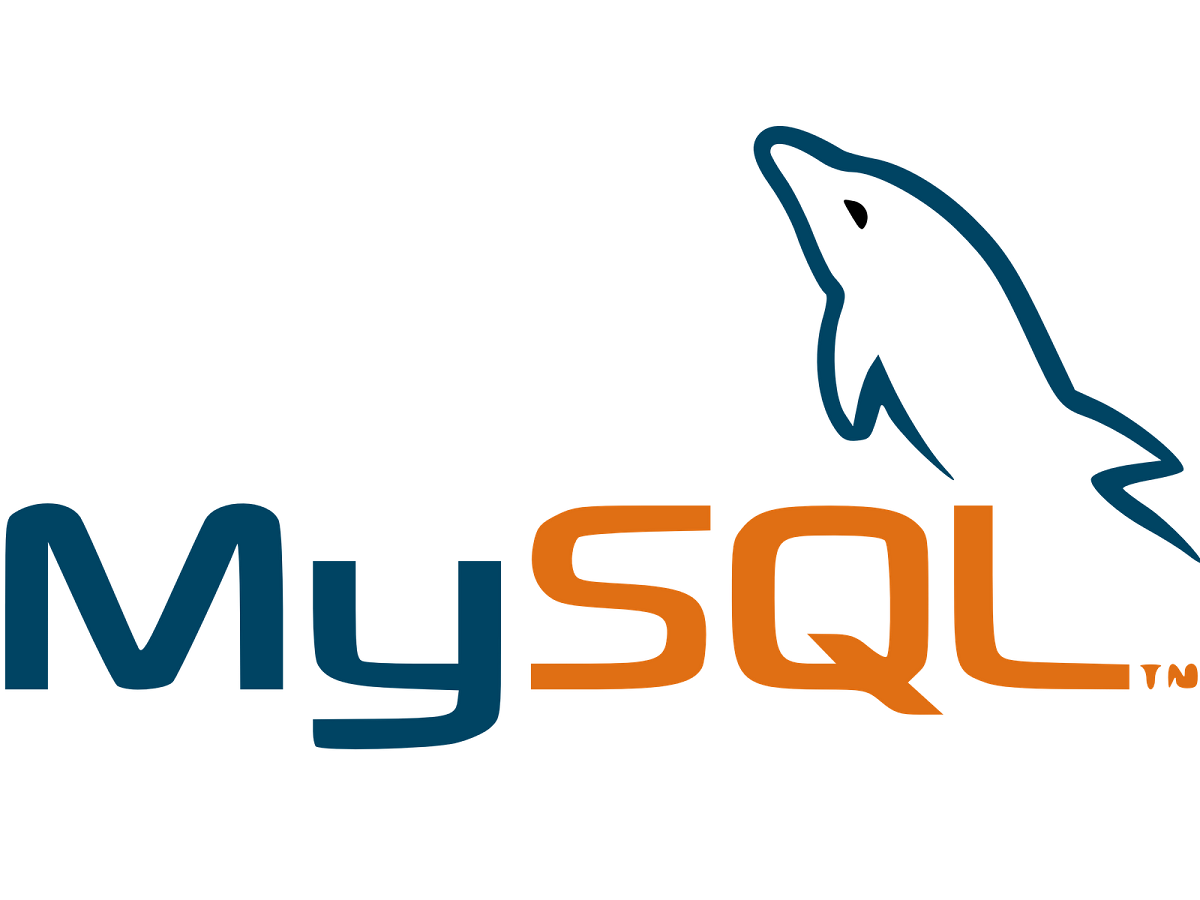
There is so many Relational Database Management Systems. This few pages will present a brief but rich description for 3 well known relational relational RDMS which are MySQL, PostgreSQL and SQL SERVER .

***Popular Relational Database Management Systems :***

Most relational database management systems use the SQL language to access the database.

SQL (Structured Query Language) is a programming language used to communicate with data stored in a relational database management system. SQL syntax may differ slightly depending on which RDBMS you are using.

[**MySQL**](https://www.mysql.com/)



**What Is MySQL?**

MySQL is an oracle-backed open source [relational database](https://www.techtarget.com/searchdatamanagement/definition/relational-database) management system ([RDBMS](https://www.techtarget.com/searchdatamanagement/definition/RDBMS-relational-database-management-system)) based on structured query language ([SQL](https://searchsqlserver.techtarget.com/definition/SQL)). MySQL runs on virtually all platforms, including [linux](https://www.techtarget.com/searchdatacenter/definition/Linux-operating-system), [UNIX](https://www.techtarget.com/searchdatacenter/definition/Unix) and [windows](https://www.techtarget.com/searchwindowsserver/definition/Windows). Although it can be used in a wide range of applications, mysql is most often associated with web applications and online publishing.

MySQL is an important component of an open source enterprise stack called [LAMP](https://www.techtarget.com/whatis/definition/LAMP-Linux-Apache-MySQL-PHP). LAMP is a web development platform that uses Linux as the operating system, [Apache](https://www.techtarget.com/whatis/definition/Apache) as the web server, MySQL as the relational database management system and PHP as the object-oriented scripting language.(Sometimes [Perl](https://www.techtarget.com/whatis/definition/Perl) or [Python](https://www.techtarget.com/whatis/definition/Python) is used instead of PHP.)  
MySQL is a combination of ‘My’ the name of co-founder [Michael Widenius](https://en.wikipedia.org/wiki/Michael_Widenius)'s daughter and ‘SQL.’ MySQL is a relational database management system developed and released back in 1995. It was originally conceived by the Swedish company MySQL AB, MySQL was acquired by Sun Microsystems in 2008 and then by Oracle when it bought Sun in 2010. Developers can use MySQL under the GNU General Public License ([GPL](https://www.techtarget.com/searchdatacenter/definition/GNU-General-Public-License-GNU-GPL-or-simply-GPL)), but enterprises must obtain a commercial license from Oracle.

Today, MySQL is used by many database-driven web applications including Durpal, Joomla, php and WordPress. MySQL is also the RDBMS behind many of the top websites in the world and countless corporate and consumer-facing web-based applications, including Facebook, Twitter and Youtube.  
Some of the disadvantages are that it has been known to suffer from poor performance when scaling, open source development has lagged since Oracle has taken control of MySQL, and it does not include some advanced features that developers may be used to.

**What are MySQL functionalities ?**

The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications. The most common use for MySQL however, is for the purpose of a web database. It can be used to store anything from a single record of information to an entire inventory of available products for an online store. In association with a scripting language such as PHP or Perl ,it is possible to create websites which will interact in real-time with a MySQL database to rapidly display categorized and searchable information to a website user.

[**PostgreSQL**](https://www.postgresql.org/)

****

**What is Postgres?**

According to the [PostgreSQL website](https://www.postgresql.org/), “PostgreSQL is a powerful, open source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.”  
 PostgreSQL  also known as Postgres, is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source_software) [relational database managementsystem](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS)emphasizing [extensibility](https://en.wikipedia.org/wiki/Extensibility) and [SQL compliance](https://en.wikipedia.org/wiki/SQL_compliance). It was originally named Postgres, referring to its origins as a successor to the [Ingres](https://en.wikipedia.org/wiki/Ingres_(database)) database developed at the [University of California, Berkeley](https://en.wikipedia.org/wiki/University_of_California,_Berkeley) by a computer science professor named Michael Stonebraker. Stonebraker started Postgres in 1986 as a follow-up project to its predecessor, Ingres. In 1996, the project was renamed to PostgreSQL to reflect its support for [SQL](https://en.wikipedia.org/wiki/SQL). After a review in 2007, the development team decided to keep the name PostgreSQL and the alias Postgres.   
 Postgres is currently used in production by a wide variety of enterprises across many industries, including financial services, information technology, government and media and communications , be it small startups or large organizations such as Apple, Instagram, Twitch, and Reddit.

**What are PostgreSQL functionalities ?**

PostgreSQL As a database server, its primary function is to store data, securely and supporting best practices, and retrieve it later, as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the internet). It can handle workloads ranging from small single-machine applications to large internet-facing applications with many concurrent users. Recent versions also provide replication of the database itself for security and scalability.  
 PostgreSQL Implements the majority of the SQL:2011 standard, is acid-compliant and transactional (including most DDL statements) avoiding locking issues using multiversion concurrency control (MVCC), provides immunity to dirty reads and full serializability; handles complex SQL queries using many indexing methods that are not available in other databases; has updateable views and materialized views, triggers, foreign keys; supports functions and stored procedures, and other expandability, and has a large number of extensions written by third parties. In addition to the possibility of working with the major proprietary and open source databases, PostgreSQL supports migration from them, by its extensive standard SQL support and available migration tools. And if proprietary extensions had been used, by its extensibility that can emulate many through some built-in and third-party open source compatibility extensions, such as for Oracle.

[**SQL Server**](https://www.postgresql.org/)



**What is SQL Server?**

SQL Server is a relational database management system (RDBMS) developed by microsoft. It is primarily designed and developed to compete with mysql and oracle database. SQL server supports ANSI SQL, which is the standard SQL (structured query language) language. However, SQL Server comes with its own implementation of the SQL language, T-SQL (transact-sql).  
T-SQL is a microsoft propriety language known as Transact-SQL. It provides further capabilities of declaring variable, exception handling, stored procedure, etc.  
SQL Server management studio (SSMS) is the main interface tool for SQL Server, and it supports both 32-bit and 64-bit environments.

**What exactly is Microsoft SQL Server for?**

Microsoft SQL server is ideal for storing all the desired information in relational databases, as well as to manage such data without complications, thanks to its visual interface and the options and tools it has. This is vital, especially for websites that have the option of registering users to log in.  
  
For companies, using this tool is essential because of the facilities it offers and the utilities it has. If you have a list of customers, a product catalog or even a large selection of multimedia content available, Microsoft SQL server helps to manage absolutely everything. It is essential for the proper functioning of a website or any application.

Its main component is composed of a relational engine in charge of processing commands, queries, as well as the storage of files, bb.Dd., Tables and data buffers. Its secondary levels are intended for memory management, programming and administration of request and response interactions with the servers that host the databases.

**Functions and Features of Microsoft SQL Server:**

Some of the main functions that distinguish Microsoft SQL Server, are its variety of tools aimed at data management and analysis, as well as business intelligence with which to gain insights about your business and customers supported by machine learning.  
Microsoft SQL server allows you to easily integrate your data into applications and take advantage of a broad set of cognitive services to leverage artificial intelligence at any data scale, both on-premises and cloud environments thanks to its integration with azure AI.  
 Generally, SQL SERVER offer the user high availability to enable faster switching processes. Its in-memory capabilities allow for increased flexibility and ease of use by providing seamless integration with the microsoft server family of servers.  
Being based on open source it is very easy to access and the vast majority of programmers working in web development have used microsoft SQL server in some of their projects, as well as being very widespread also has a large community that offers support to other users.

Now that you know what Microsoft SQL Server is and what it is used for, you should know some advantages that make it very interesting for developers. The clearest one is that it works with relational databases, that is, using multiple interconnected tables to store information and organize it correctly.

**Comparison of MySQL vs. PostgreSQL vs. SQL Server**

General information for MySQL, PostgreSQL and SQL Server

|  |  |  |  |
| --- | --- | --- | --- |
|  | **MySQL** | **PostgreSQL** | **SQL Server** |
| Maturity | Initial release was in 1995 | Initial release was in 1989 | MSMS SQL Server for OS/2 was released in 1989 (together with Sybase)  SQL Server 6.0 was released in 1995 marking the end of collaboration with Sybase. |
| Language | Written in C, has a few C++ modules | Written in C | Mostly C++ with a few exceptions |
| Cost | Open source / Owned by Oracle and has several paid editions | Completely free / Open source | SQL Server Express is a free edition, but it is limited to using 1 processor, 1 GB memory and 10 GB database files. |

Data changes for MySQL, PostgreSQL and SQL Server

|  |  |  |  |
| --- | --- | --- | --- |
|  | **MySQL** | **PostgreSQL** | **SQL Server** |
| Row Updates | Updates happen in place, changed data is copied to the rollback segment. This makes vacuuming and index compaction very efficient. MySQL is slower for reads, but writes are atomic and if columns in a secondary index change, this does not require changes to all indexes. | Updates are being implemented as inserts + mark as delete for vacuum. All indexes have a link to the physical id of the row. This has an update amplifying effect because when the column gets updated, new row with new physical id gets created and all indexes require updates, even those which are not referring to the changed column to get a pointer to the new row physical id. | Row-Store database engine:  In-Memory database engine: updates implemented as insert + mark for delete. Garbage collector is not non-blocking and parallel  Columnstore database engine: in-place updates |
| Vacuum / Defragmentation | Vacuuming and index compaction are very efficient. | Vacuum performs full tables scans to find the deleted rows and quite heavy process/might impact users’ workload. | In-memory garbage collector might add max ~15% overhead, usually much less. |

Querying the data for MySQL, PostgreSQL and SQL Server

|  |  |  |  |
| --- | --- | --- | --- |
|  | **MySQL** | **PostgreSQL** | **SQL Server** |
| The buffer pool / cache that serves queries | MySQL cache that serves user queries is called a buffer pool. This cache can be set to the size as large as needs, leaving only enough memory for other processes on the server. You can split the buffer pool into multiple parts to minimize contention for memory structures and you can pin tables to the buffer pool. Table scan or mysqldump evicts older data. | PostgreSQL maintains shared memory for data pages and, due to the fact that it is a process-based system, each connection has a native OS process of its own and has its own memory. Process is releasing the memory after the execution has finished. Therefore, has problems scaling past hundreds of active connections. | SQL Server memory is called buffer pool and its size can be set as large as needed, no option to set multiple buffer pools. |
| Constraints support | Supports primary keys, foreign keys, not-null constraints, unique constraints, default constraints, does not support CHECK constraints | Supports primary keys, foreign keys, not-null constraints, check constraints, unique constraints, default constraints, [exclusion constraints](https://www.cybertec-postgresql.com/en/postgresql-exclude-beyond-unique/) | Supports primary keys, foreign keys, not-null constraints, check constraints, unique constraints, default constraints |
| Temporary tables | Supports CTE, No support for global temp tables (available outside the session scope) and no table variables.  Interesting fact: You cannot refer to a TEMPORARY table more than once in the same query. For example, the following does not work: SELECT \* FROM temp\_table JOIN temp\_table AS t2; | Supports CTE, Global and local temporary tables and table variables (using table name as a type name).  Interesting fact: if you create two tables with the same name, one is temporary and another one is regular table CREATE TEMP TABLE X (…) and CREATE TABLE X (…), "select \* from x" will always bring data from temporary table. | Supports CTE, Global and local temporary tables and table variables. |
| Window / Analytical functions | Supports:  CUME\_DIST, FIRST\_VALUE, LAG, LAST\_VALUE, LEAD, PERCENT\_RANK, ROW\_NUMBER, RANK, DENSE\_RANK, NTILE, NTH\_VALUE No PERCENTILE\_CONT, PERCENTILE\_DISC functions. | Supports functions:  CUME\_DIST, FIRST\_VALUE, LAG, LAST\_VALUE, LEAD, PERCENTILE\_CONT, PERCENTILE\_DISC, PERCENT\_RANK, ROW\_NUMBER, RANK, DENSE\_RANK, NTILE, NTH\_VALUE | Supports functions:  CUME\_DIST, FIRST\_VALUE, LAG, LAST\_VALUE, LEAD, PERCENTILE\_CONT, PERCENTILE\_DISC, PERCENT\_RANK, ROW\_NUMBER, RANK, DENSE\_RANK, NTILE. Yet no NTH\_VALUE function |
| Parallel query execution | MySQL will usually use 1 CPU per query. | Query plans can leverage multiple CPUs | Query plans can leverage multiple CPUs |
| Indexes | Supports index-organized tables - clustered indexes.  Does not support persisted indexes / materialized views | Supports index-organized table, but updates are manual until ProstgreSQL 11 when it is automatic.Supports persisted indexes/materialized views. | Supports index-organized tables - clustered indexes that automatically maintains rows order. |
| Multiple indexes usage in single query | Multiple indexes might be used for the single query. | Multiple indexes might be used for the single query. If we have separate indexes on x and y, one possible implementation of a query like WHERE x = 5 AND y = 6 is to use each index with the appropriate query clause and then AND together the index results to identify the result rows. | Multiple indexes might be used for a single query (index intersection feature). |
| Multicolumn indexes | Multi-column indexes can have up to 16 columns | Multi-column indexes can have up to 32 columns | Multi-column indexes can have up to 16 columns |
| Partial indexes (an index built over a subset of a table using filter) | Does not support partial indexes | Supports partial indexes | Supports partial indexes |
| Join algorithms | MySQL executes joins between tables using only a nested-loop algorithm or variations of it. | Supports nested-loop joins, Hash joins and merge joins algorithms. | Supports nested-loop joins, hash joins and merge joins algorithms. |
| Query execution plan reuse | Maintains caches for prepared statements and stored programs on a per-session basis. Statements cached for one session are not accessible to other sessions. | Caches query plans only as long as the prepared statement is open. The query plan is disposed when the prepared statement is closed. | Has shared execution plan cache to enable queries to reuse execution plans |
| Statistics | Maintains persistent and non-persistent statistics (cleared on server restart) | Maintains statistics used by the planner, they are being updated by ANALYZE or VACUUM or CREATE INDEX | Maintains persistent statistics |
| Memory-optimized tables | MySQL has got an ability to store tables in memory. The tables that are created in memory do not support transactions, their data is vulnerable to crashes. Those tables should be used as a temporary area or as a read-only caches. | Does not offer any in-memory engine. | In-memory OLTP is integrated into SQL Server’s database engine |
| Columnstore or row- store | MariaDB have recently launched the column store engine for MySQL which was designed as a massively parallel database in an environment with multiple servers. It can be used instead of InnoDB storage engine. | Row-store. Does not offer any columnar storage engine. | SQL Server offers column store indexes to query large tables |

JSON and Data Type Support for MySQL, PostgreSQL and SQL Server

|  |  |  |  |
| --- | --- | --- | --- |
|  | **MySQL** | **Postgresql** | **SQL Server** |
| JSON data type | MySQL has JSON data type support and also supports in place partial updates over the JSON instead of replacing the whole document however there are many limitations. It does not support indexing for JSON but there are workarounds. | PostgreSQL supports JSON data type and supports partial updates | SQL Server supports JSON data type and supports partial updates |
| Additional Advanced data types | Supports Geospatial data type. No user-defined types. | Supports Geospatial and lots of advanced data types, such as multi-dimensional arrays, user-defined types, etc. | Supports Geospatial data type, Hierarchical data |

Sharding / Partitioning / Replication for MySQL, PostgreSQL and SQL Server

|  |  |  |  |
| --- | --- | --- | --- |
|  | **MySQL** | **PostgreSQL** | **SQL Server** |
| Partitioning support | Supports HASH partitioning (use HASH function on any column to split table into N partitions), RANGE or LIST partitioning that can be based on several columns and KEY partitioning which is similar to HASH but based on some auto generated number. | Supports RANGE and LIST partitioning but partitions and indexes on them must be manually created and old-style partitioning via table inheritance (when querying the parent table, all children tables are being queries as well, children tables have constraints on partitioning column. Interesting fact: Children tables can have more columns that parent table and indexes must be applied separately on children tables.) | Supports RANGE partitioning. |
| Sharding support | No good sharding implementation (MySQL Cluster is rarely deployed due to many limitations) | There are dozens of forks of Postgres which implement sharding but none of them yet haven’t been added to the community release. | No standard sharding implementation. |
| Replication | Master-slave replication based on statements or based on changed rows  Group replication with master server automatic election | Master - slave replication based on changed rows and log shipping. | Database level: Availability Groups master-multiple slaves  Log shipping  On data level: Master-slave / Bi-directional master-slave/ and master-master (merge) replication |