

# RDBMS and their functionalities

# Definition of RDBMS

Though not strictly based on relational theory, RDBMS is a product that showcases data arranged as a collection of rows and columns. And these RDBMS products follow some of the Codd's 12 rules.

With an RDBMS, it is possible to embed a collection of programs or capabilities, enabling IT and other related teams to create, edit, update, manage and interrelate with a relational database.

Structured query language (SQL) is implemented by most commercial RDBMS systems for accessing the database. Some experts are of the opinion that RDBMS can run by itself without the intervention of SQL as it was developed after the introduction of the relational model.

one of the most popular and widely used definitions of the elements of an RDBMS comprises Codd's 12 rules. But most of these rules and the early models of the RDBMS seemed to be non-compliant, which led the term to be grouped under a broader class of database systems:

- The data would be presented to the user in a tabular form, also known as relations, which is a collection of tables and each table consists of a set of rows and columns.
- Relational operators would be provided to organize the data in the afore-mentioned tabular form.

Here are some of the implementations that were adopted as part of the early relational model systems:

- Micro DBMS
- IS1 (1970–72) and PRTV (1973–79)

It is known that the very first RDBMS system sold was Multics Relational Data Store followed by Ingres and IBM BS12.

Do you prioritize the ease-of-use when choosing an open-source RDBMS? You should use SQLite. Note that MySQL and PostgreSQL offer user-friendliness too. Many 3rd party tools support these RDBMSs, furthermore, you can find extensive documentation.

However, SQLite remains ahead of both in this regard. You don't need to configure it. SQLite doesn't execute as a server-side process. As a result, you don't need to stop, start, or restart it. Your developers don't need to manage complex configuration files either.

Do you want excellent community support for the open-source RDBMS you choose? After all, you aren't using a commercial product. Robust community support has importance for you. You should choose PostgreSQL in this case since a large and vibrant community supports it. MySQL and SQLite have their developer communities too. However, PostgreSQL maintains a lead in this regard.