Top 3 RDBMS

MySQL

MySQL happens to be one of the most popular databases, according to DB Engines Ranking. It's a definite leader among SQL solutions, used by Google, LinkedIn, Amazon, Netflix, Twitter, and others. MySQL popularity has been growing a lot because teams increasingly prefer open-source solutions instead of commercial ones.

Price: the database solution is developed by Oracle and has additional paid tools; the core functionality can be accessed for free.

Language: MySQL is written in C++; database management is done with Structured Query Language.

PostgreSQL

A tried-and-proven relational database that is known for supporting a lot of data types, intuitive storage of schemeless data, and rich functionality. Some developers go even as far as to claim that it's the most advanced open-source database on the market. We wouldn't go that far, but it's definitely a highly universal solution.

Price: open-source

Language: C

SQL Server

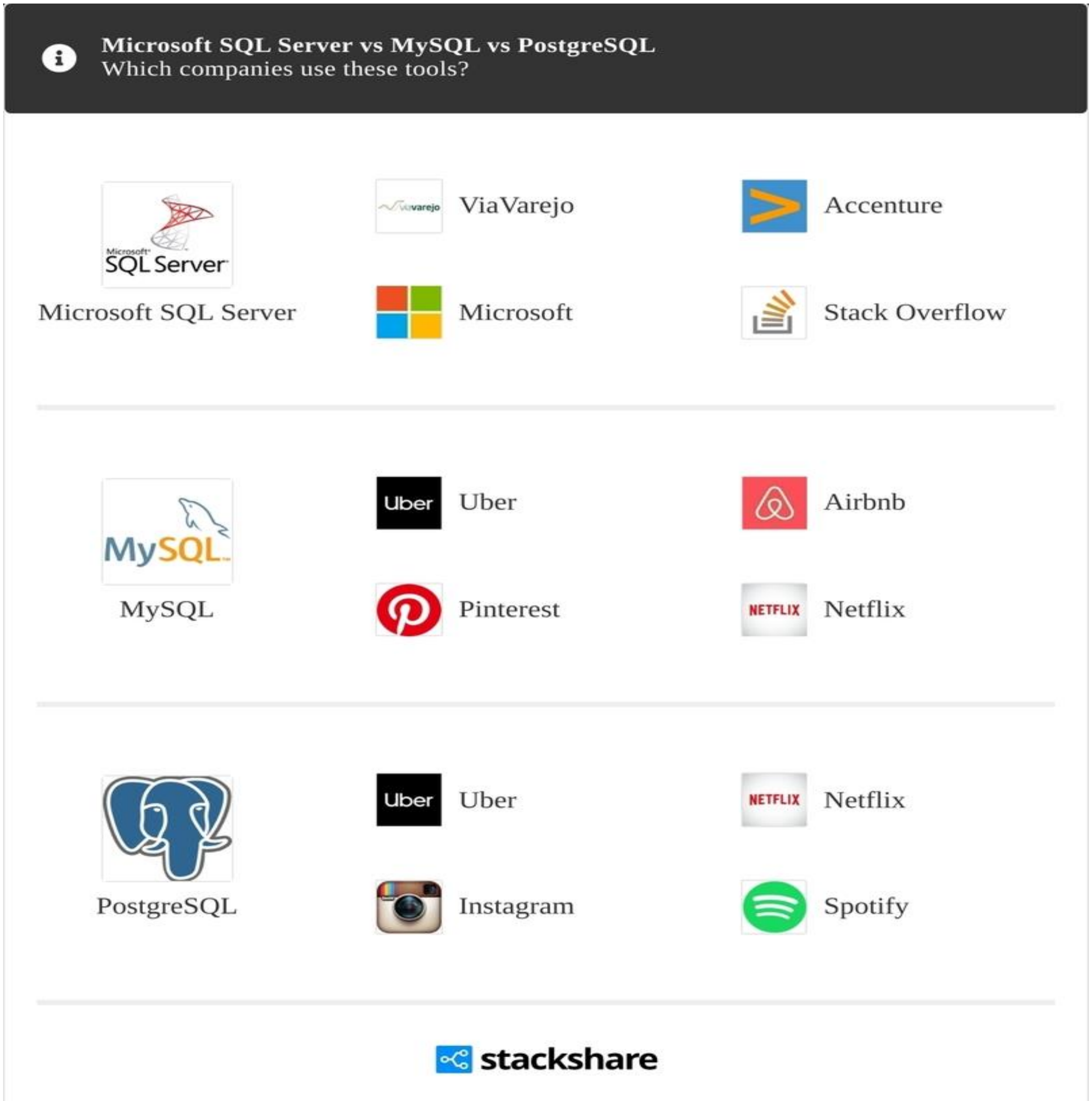
Unlike PostgreSQL vs MySQL, SQL Server is a commercial solution. It's preferred by companies who are dealing with large traffic workloads on a regular basis. It's also considered to be one of the most compatible systems with Windows services.

The SQL Server infrastructure includes a lot of additional tools, like reporting services, integration systems, and analytics. For companies that manage multiple teams, these tools make a big difference in day-to-day work.

Price: the database has a free edition for developers and small businesses but only supports 1 processor, 1GB of maximum memory used by the database engine and 10GB maximum database size.

. For a server, users need to pay $931.

Companies Using These Tools



Microsoft SQL Server vs MySQL vs PostgreSQL
Which companies use these tools?

Microsoft SQL Server	ViaVarejo	Accenture
	Microsoft	Stack Overflow
MySQL	Uber	Airbnb
	Pinterest	Netflix
PostgreSQL	Uber	Netflix
	Instagram	Spotify

stackshare

Side-by-side Comparison of SQL Tools

In this comparison, we'll take a look at the functionality of the three most popular SQL databases, examine their use cases, respective advantages, and disadvantages. Firstly, we'll start by exploring the in-depth functionality

Data Changes

Here we evaluate the ease that the data can be modified with and the database defragmented. The key priority is the systems' flexibility, security, and usability.

Row updates

This criterion refers to the algorithms that a database uses to update its contents, speed, and efficiency.

In the **MySQL** case**,** a solution updates data automatically to the rollback storage. If something goes wrong, developers can always go back to the previous version.

PostgreSQL: developers insert a new column and row in order to update the database. All updated rows have unique IDs. This multiplies the number of columns and rows and increases the size of the database, but in turn, developers benefit from higher readability.

SQL Server: the database has three engines that are responsible for row updates. The ROW Store handles the information on all previous row updates, IDs, and modified content. The in-memory engine allows analyzing the quality of an updated database with a garbage collector. The column-store database lets store updates in columns, like in column-driven databases.

=> Among these three, SQL Server offers perhaps the most flexibility and efficiency, because it allows monitoring updated rows and columns, collecting errors, and automating the process. The difference between SQL Server and MySQL and PostgreSQL lies mainly in customizing the positions – SQL Server offers a lot more than others.

Defragmentation

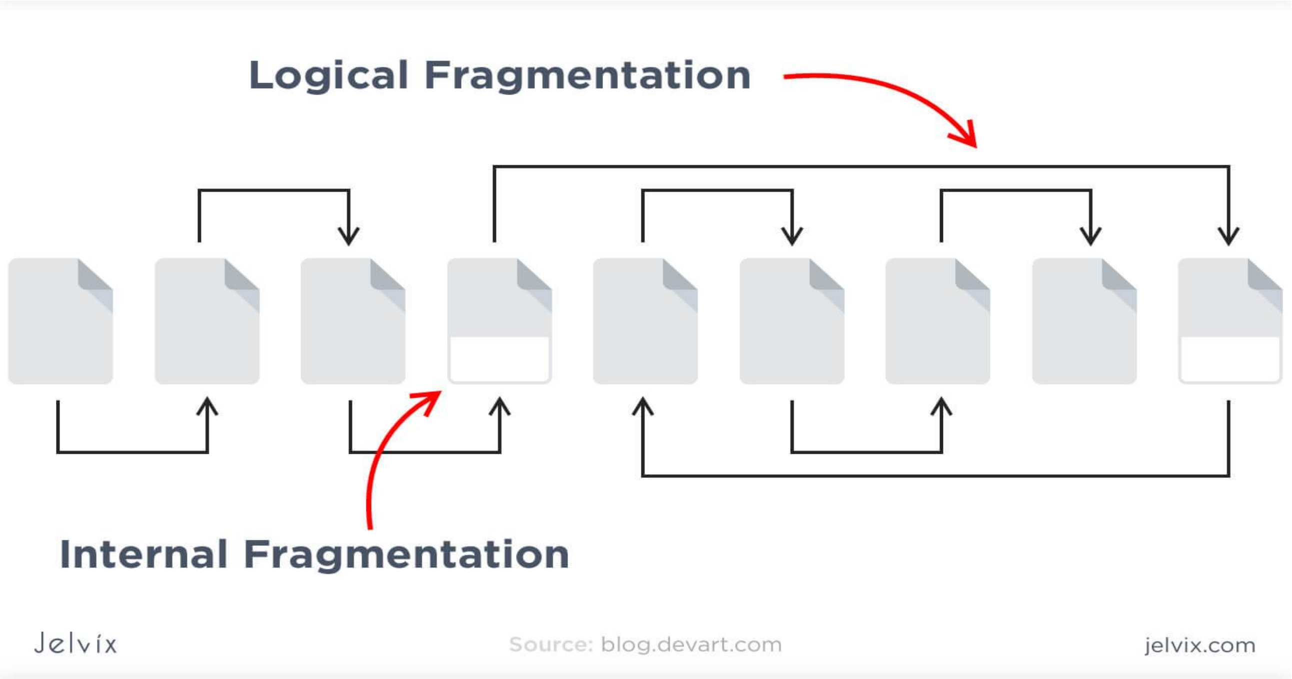
When developers update different parts of an SQL database, the changes occur at different points of the systems and can be hard to read, track, and manage. Therefore, maintenance should include defragmentation – the process of unifying the updated database by assigning indexes, revisiting the structure, and creating new pages. The database frees up the disk space that is not used properly so that a database can run faster.

MySQL offers several approaches to defragmentation – during backup, index creation, and with an OPTIMIZE Table command. Without going into much detail, we'll just say that having that many options for table maintenance is convenient for developers, and it surely saves a lot of time.

PostgreSQL allows scanning the entire tables of a data layer to find empty rows and delete the unnecessary elements. By doing so, the system frees up the disk space. However, the method requires a lot of CPU and can affect the application's performance

SQL Server offers an efficient garbage collector that doesn't create more than 15-20% of overhead. Technically, developers can even run garbage collector on a continuous basis, because it's that efficient.

=> Overall, MySQL and SQL Server offer more of defragmentation methods that Postgresql does. They consume less CPU and provide more flexible settings



Source: blog.devart.com

JSON Support

The use of JSON files allows developers to store non-numeric data and achieve faster performance. JSON documents don't have to be parsed, which contributes to much higher processing speed. They are easily readable and accessible, which is why JSON support simplifies maintenance. JSON files are mostly used in non-relational databases, but lately, SQL solutions have supported this format as well.

MySQL supports JSON files but doesn't allow indexing them. Overall, the functionality for JSON files in MySQL is very limited, and developers mostly prefer using classical strings. Similarly to non-relational databases, MySQL also allows working with geospatial data, although handling it isn't quite as intuitive.

PostgreSQL supports JSON files, as well as their indexing and partial updates. The database supports even more additional data than MySQL. Users can upload user-defined types, geospatial data, create multi-dimensional arrays, and a lot more.

SQL Server also provides full support of JSON documents, their updates, functionality, and maintenance. It has a lot of additional features for GPS data, user-defined types, hierarchical information, etc.

Overall, all three solutions are pretty universal and offer a lot of functionality for non-standard data types. MySQL, however, puts multiple limitations for JSON files, but other than that, it’s highly compatible with advanced data