Introduction to Databases

Chahed Nedra-

Relational DBMS

- Relational database management systems (RDBMS) support the relational (=table-oriented) data model. The schema of a table (=relation schema) is defined by the table name and a fixed number of attributes with fixed data types. A record (=entity) corresponds to a row in the table and consists of the values of each attribute. A relation thus consists of a set of uniform records.
- **Most popular examples**
    - Oracle
    - MySQL
    - Microsoft SQL Server
    - PostgreSQL
    - IBM Db2

MySQL & Functionalities

- **MySQL** is the most famous large-scale database server. It is an open-source system that powers a large number of applications and websites online. It is easy to set up and requires minimal fine-tuning to achieve excellent performance levels. Third-party GUI tools such as Adminer, HeidiSQL, MySQL Workbench, and dbForge Studio make MySQL even more straightforward to get started with the database.

MySQL features

❖ Easy to use

❖ It is secure

❖ Client/ Server Architecture

❖ It is scalable

❖ High Flexibility

❖ Compatible on many operating systems

❖ High Performance

❖ GUI Support

PostgreSQL & Functionalities

- **PostgreSQL** is considered the most advanced and powerful SQL compliant and open-source objective-RDBMS. It has become the first choice for corporations that perform complex and high-volume data operations due to its powerful underlying technology.. PostgreSQL is extremely capable of handling multiple tasks simultaneously and efficiently, due to which it is trusted by business giants such as **Apple, Yahoo!, Instagram, Facebook and Instagram** along with numerous other telecoms, financial and government institutions.

PostgreSQL features

❖User-defined types

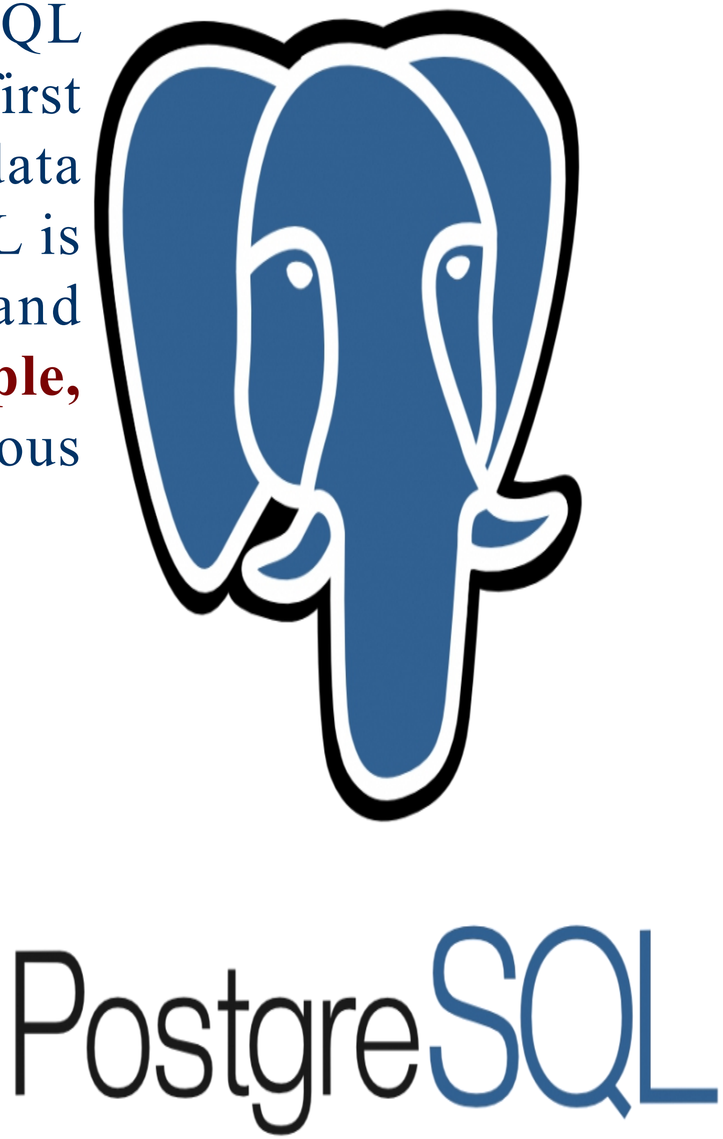
❖Table inheritance

❖Multi-version concurrency control (MVCC)

❖Foreign key referential integrity

❖Asynchronous replication

❖Sophisticated locking mechanism

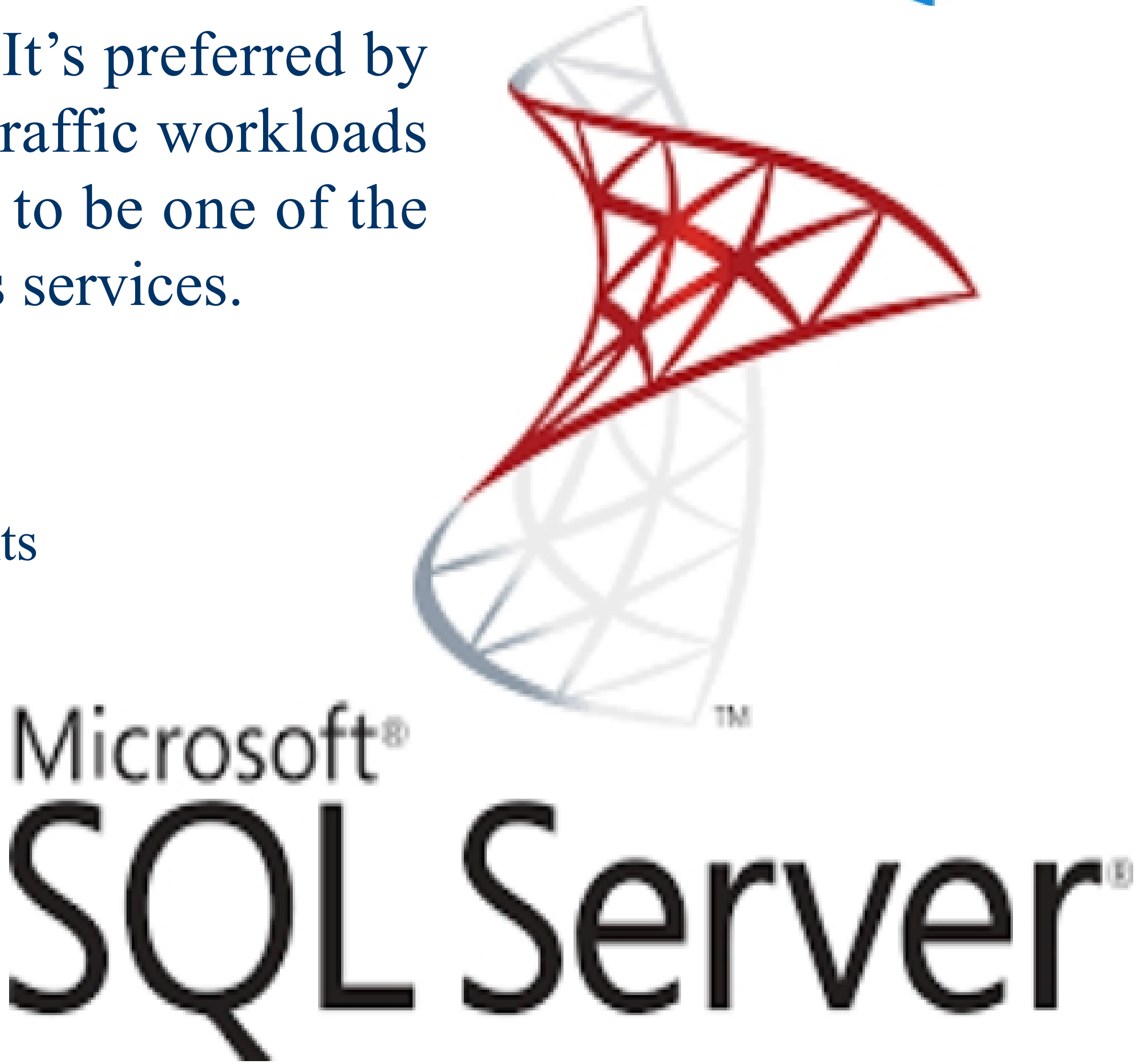


SQL Server & Functionalities

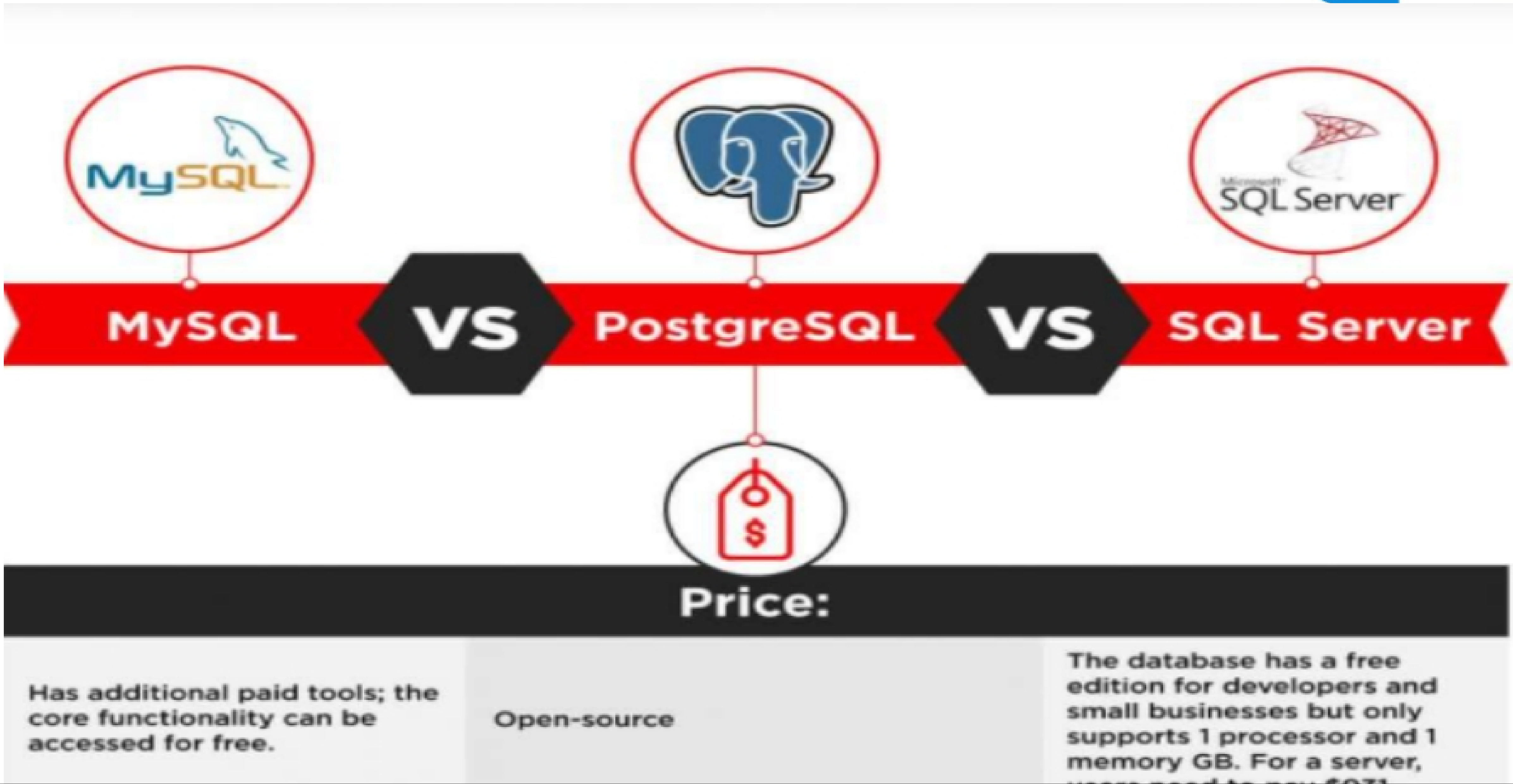
- **SQL Server** is a commercial solution. It’s preferred by companies who are dealing with large traffic workloads on a regular basis. It’s also considered to be one of the most compatible systems with Windows services.

SQL Server features

- Intelligent Query Processing Enhancements
- Accelerated Database Recovery (ADR)
- AlwaysEncrypted with secure enclaves
- Memory-optimized Tempdb metadata
- Query Store custom capture policies
- Resumable index build
- Data virtualization with Polybase



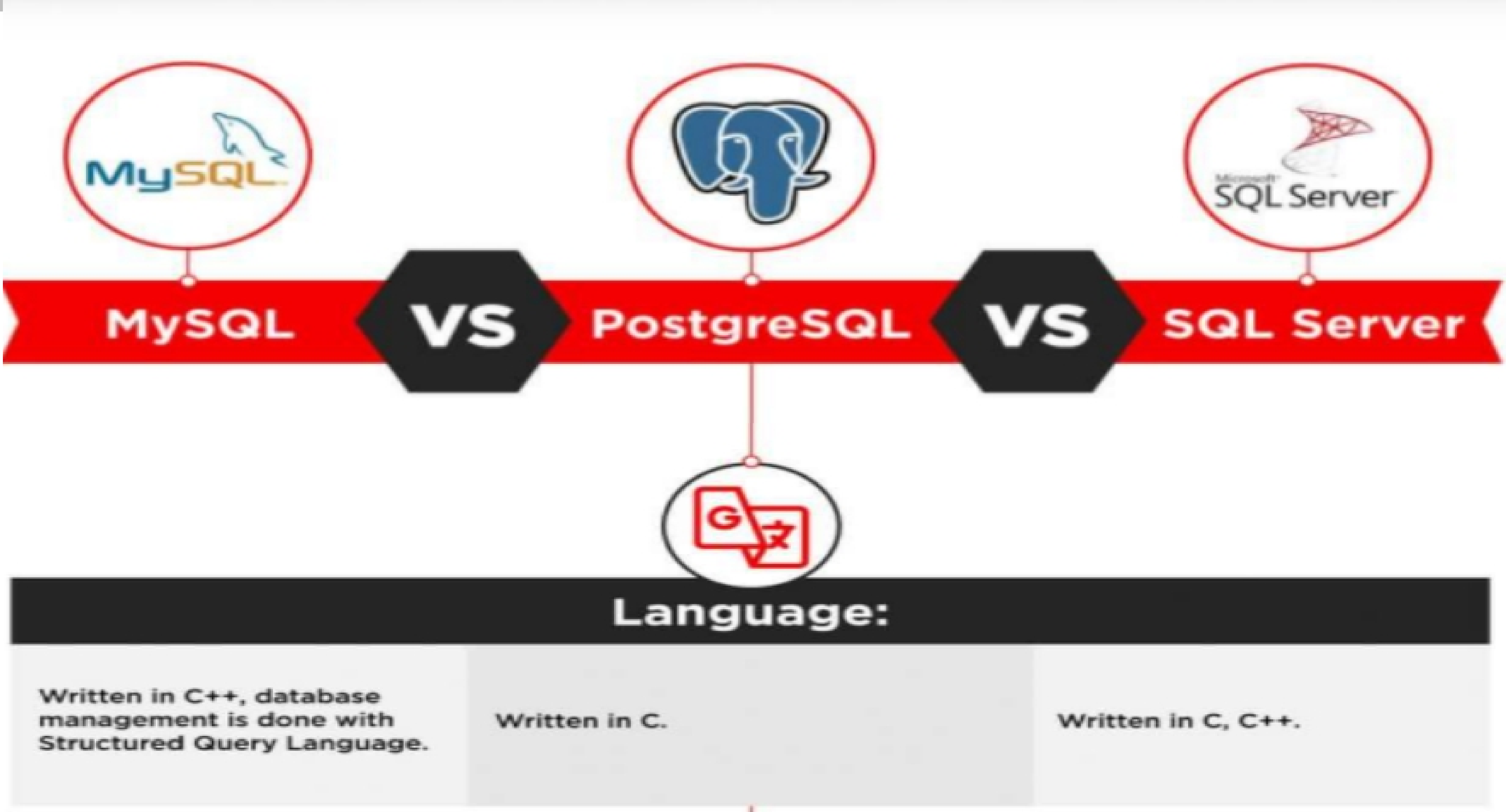
General information for MySQL, PostgreSQL and SQL Server



Price:

MySQL	PostgreSQL	SQL Server
Has additional paid tools; the core functionality can be accessed for free.	Open-source	The database has a free edition for developers and small businesses but only supports 1 processor and 1 memory GB. For a server,

Language

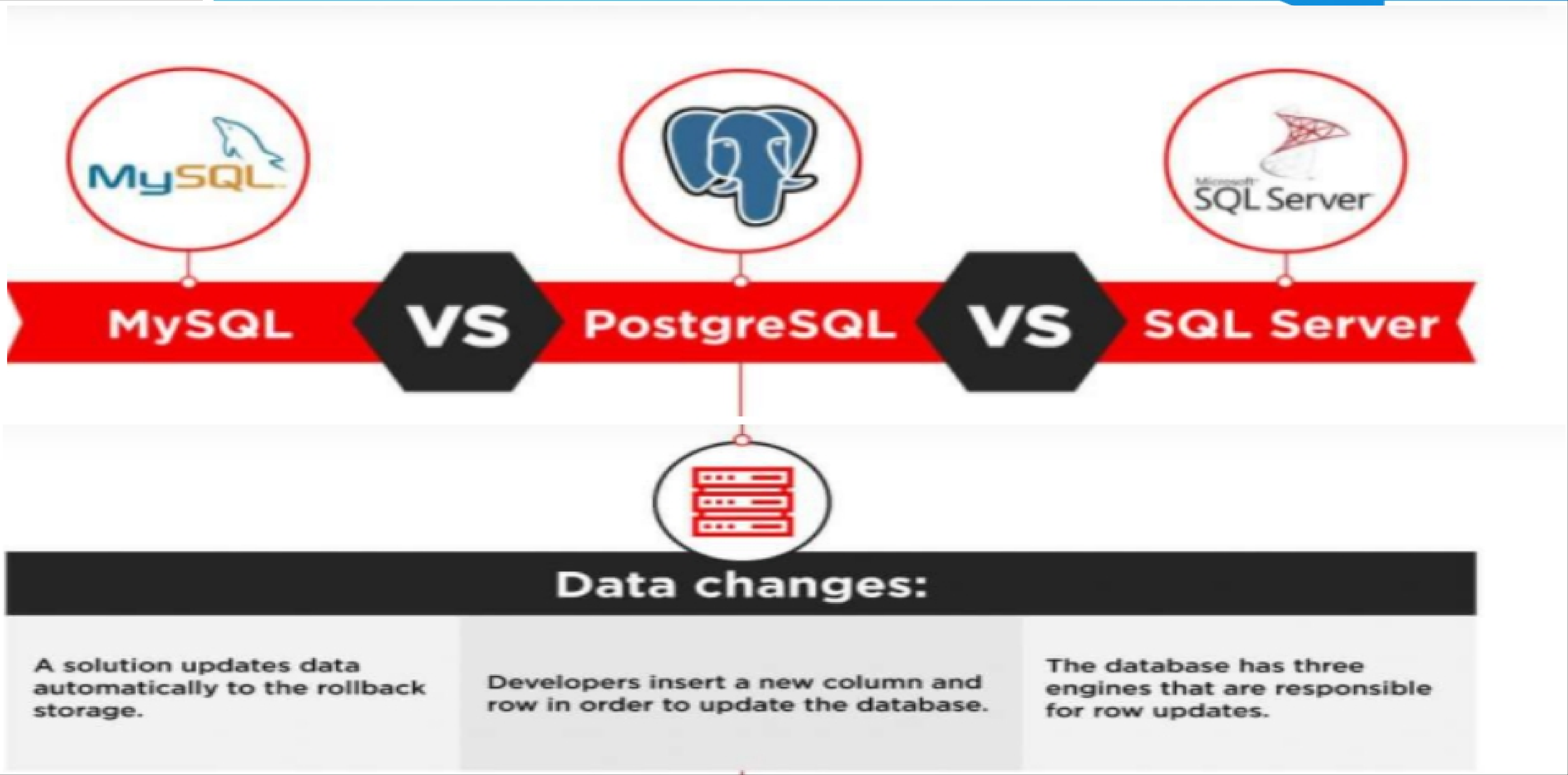

MySQL
PostgreSQL
Microsoft SQL Server

MySQL VS PostgreSQL VS SQL Server

Language:

Written in C++, database management is done with Structured Query Language.	Written in C.	Written in C, C++.

Data changes

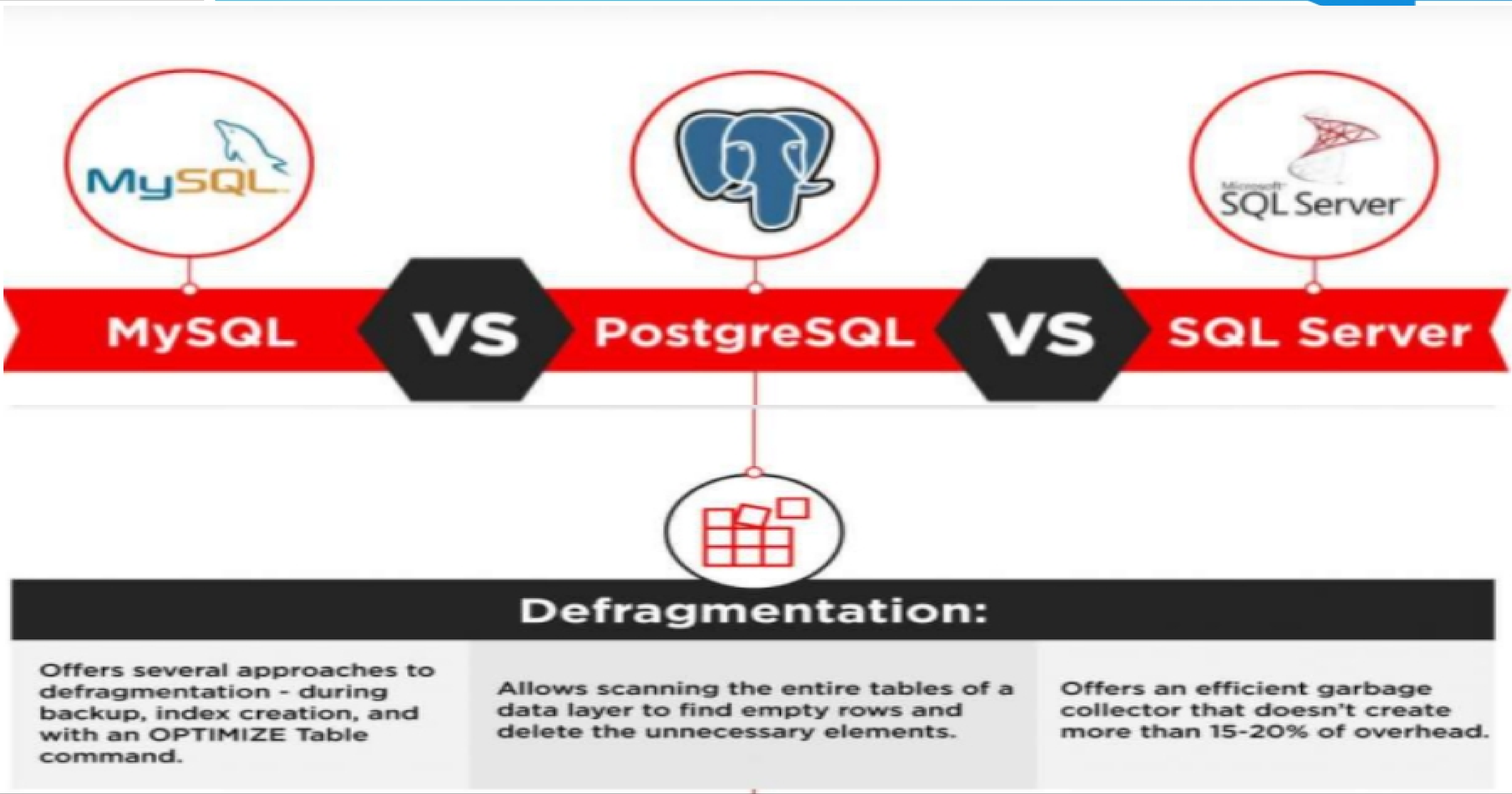


MySQL VS PostgreSQL VS SQL Server

Data changes:

A solution updates data automatically to the rollback storage.	Developers insert a new column and row in order to update the database.	The database has three engines that are responsible for row updates.

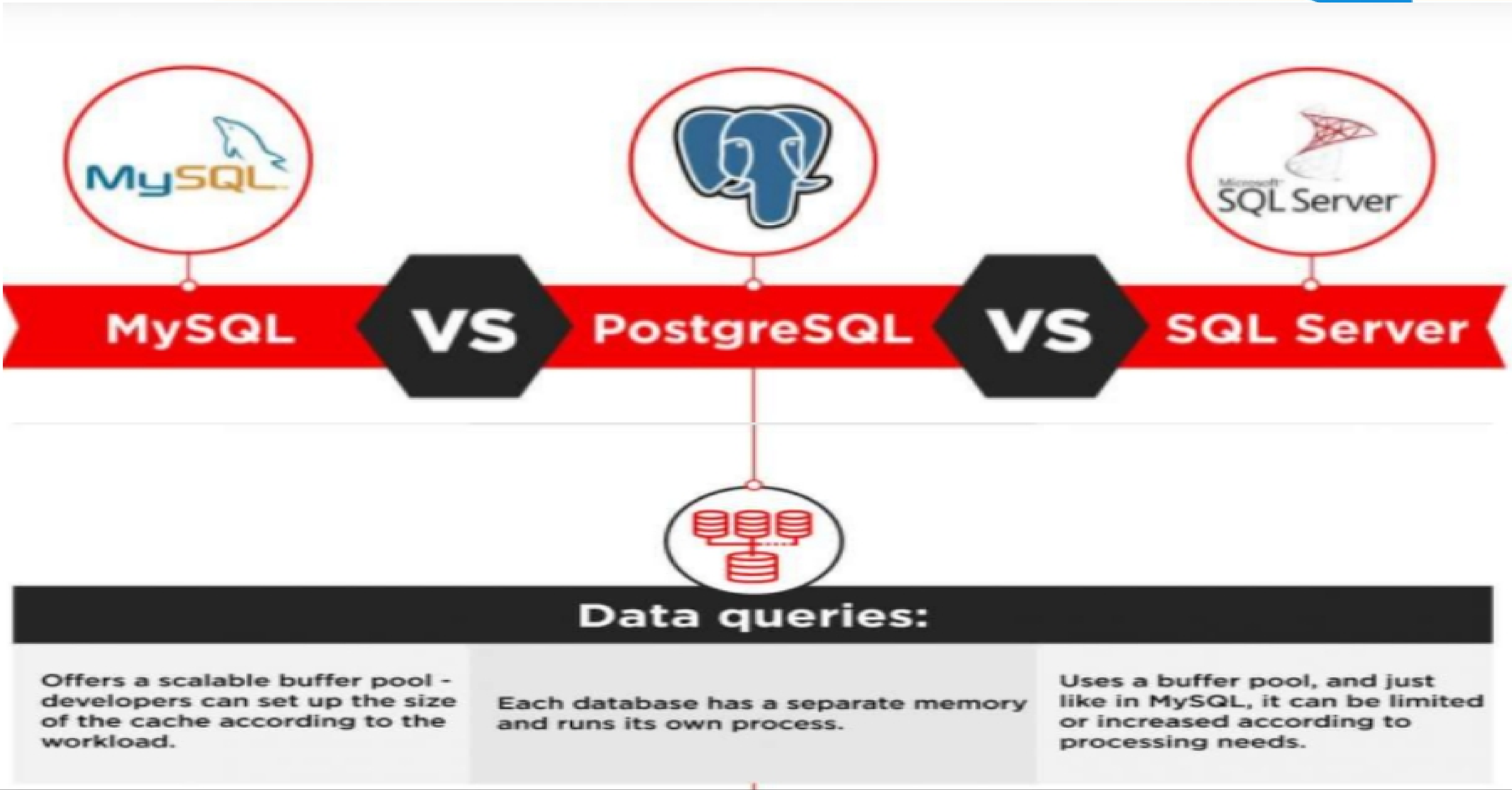
Defragmentation



MySQL	PostgreSQL	SQL Server
Offers several approaches 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.

MySQL
VS
PostgreSQL
VS
SQL Server
Defragmentation:

Data queries

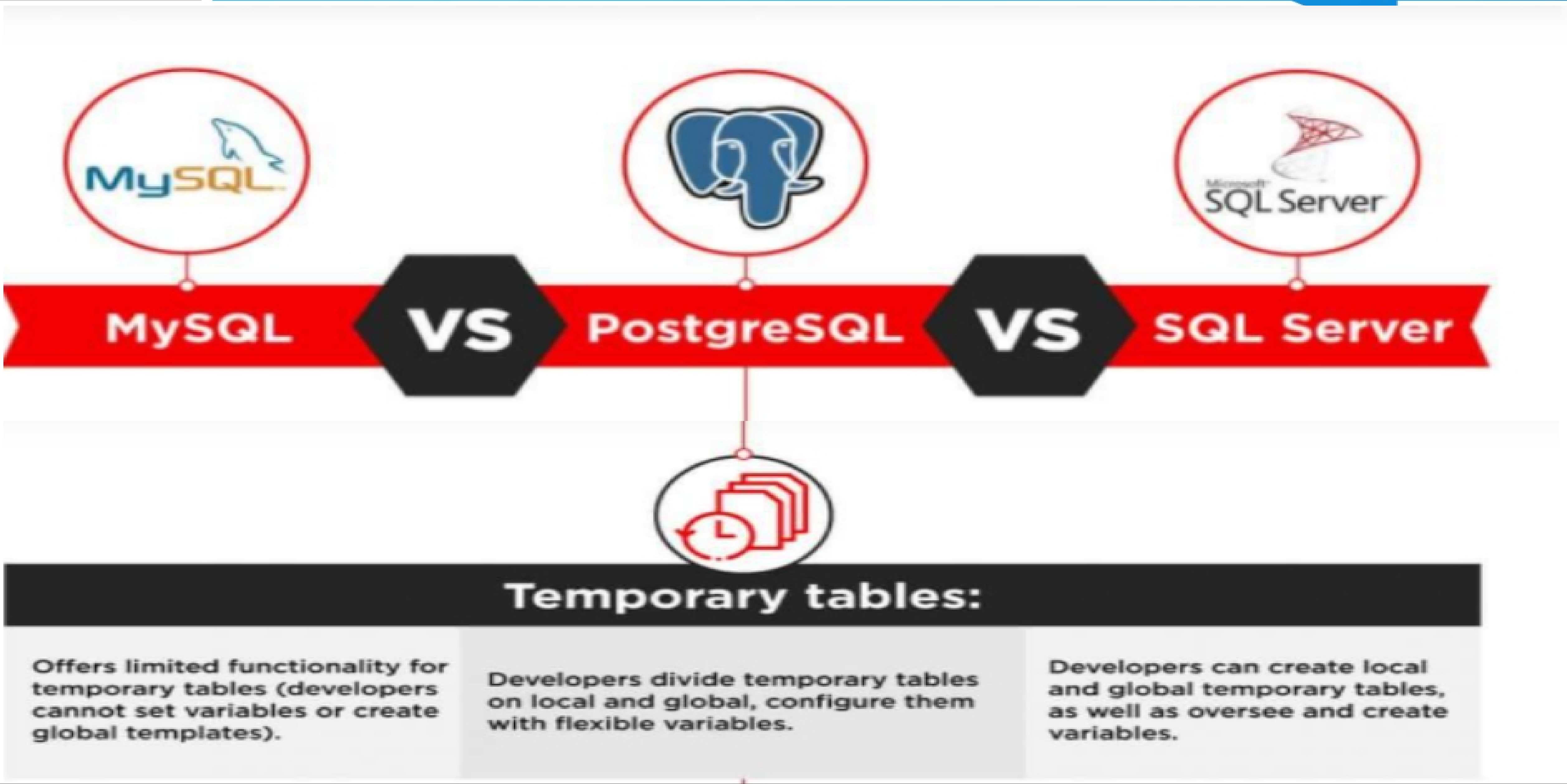

MySQL
Microsoft
SQL Server

MySQL VS PostgreSQL VS SQL Server

Data queries:

MySQL	PostgreSQL	SQL Server
Offers a scalable buffer pool - developers can set up the size of the cache according to the workload.	Each database has a separate memory and runs its own process.	Uses a buffer pool, and just like in MySQL, it can be limited or increased according to processing needs.

Temporary tables



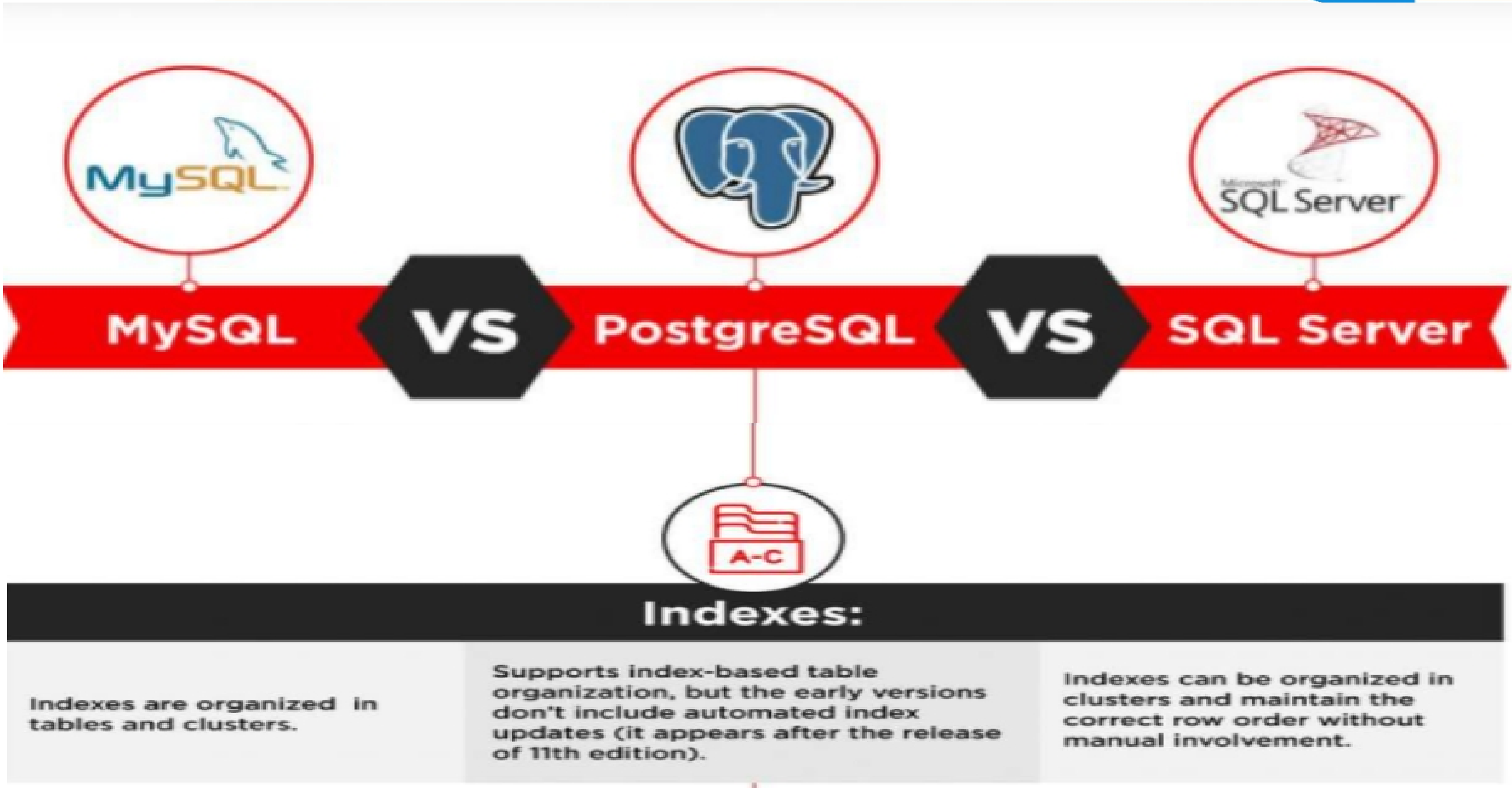
Temporary tables:

Offers limited functionality for temporary tables (developers cannot set variables or create global templates).

Developers divide temporary tables on local and global, configure them with flexible variables.

Developers can create local and global temporary tables, as well as oversee and create variables.

Indexes

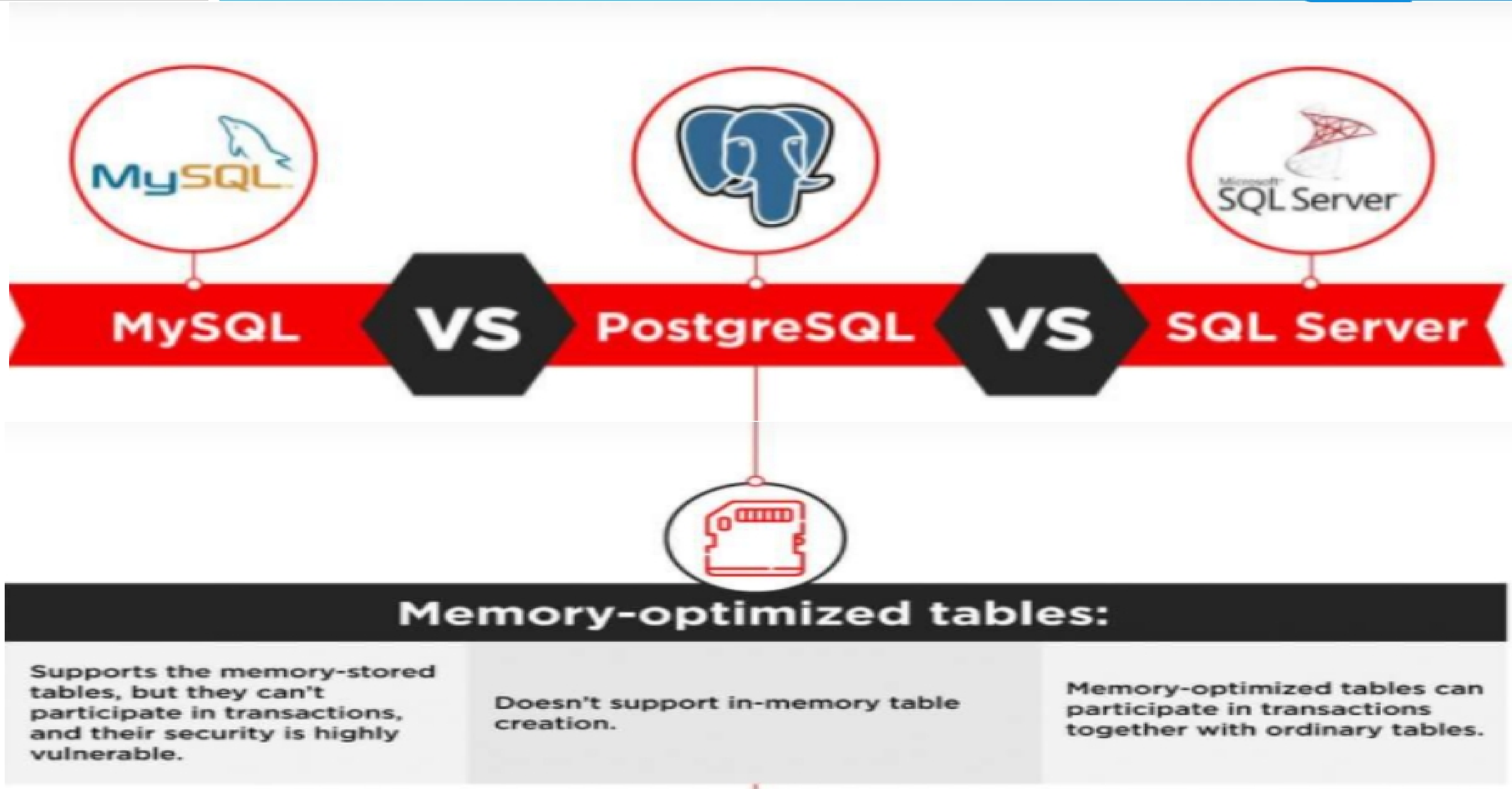


MySQL VS PostgreSQL VS SQL Server

Indexes:

MySQL	PostgreSQL	SQL Server
Indexes are organized in tables and clusters.	Supports index-based table organization, but the early versions don't include automated index updates (it appears after the release of 11th edition).	Indexes can be organized in clusters and maintain the correct row order without manual involvement.

Memory optimized tables

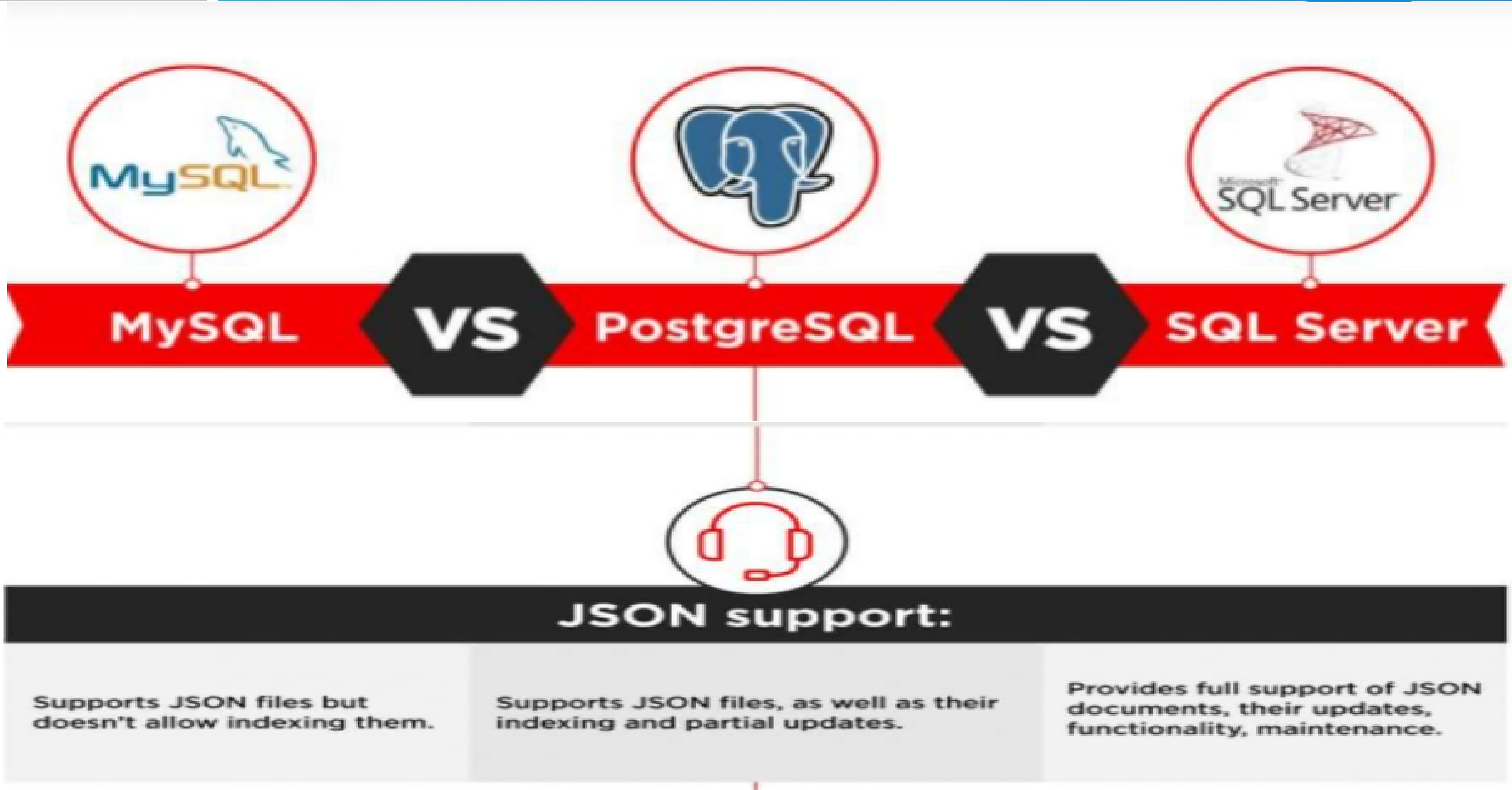


MySQL VS PostgreSQL VS SQL Server

Memory-optimized tables:

MySQL	PostgreSQL	SQL Server
Supports the memory-stored tables, but they can't participate in transactions, and their security is highly vulnerable.	Doesn't support in-memory table creation.	Memory-optimized tables can participate in transactions together with ordinary tables.

JSON support

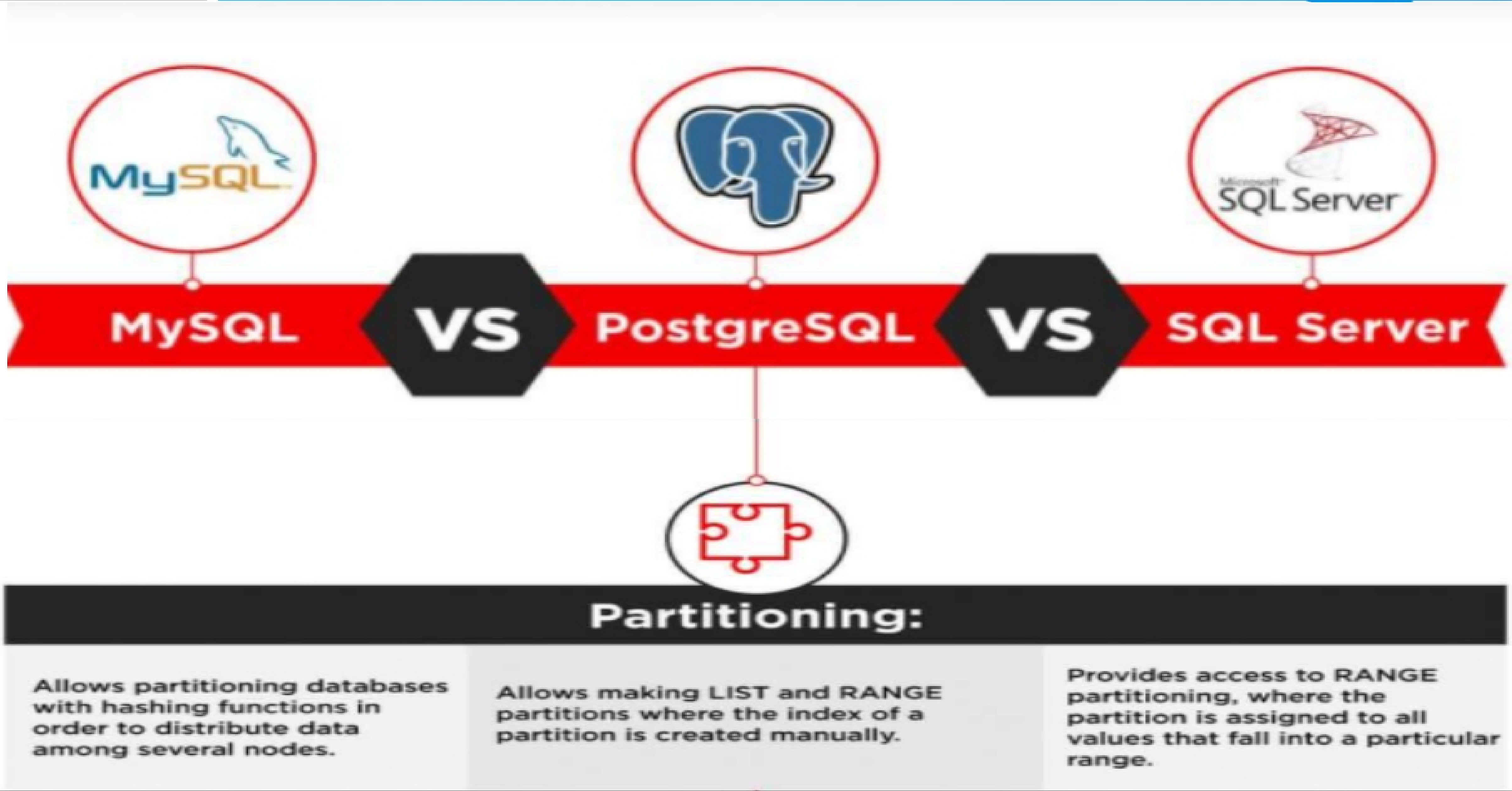


MySQL VS PostgreSQL VS SQL Server

JSON support:

MySQL	PostgreSQL	SQL Server
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, their updates, functionality, maintenance.

Partitioning



MySQL VS PostgreSQL VS SQL Server

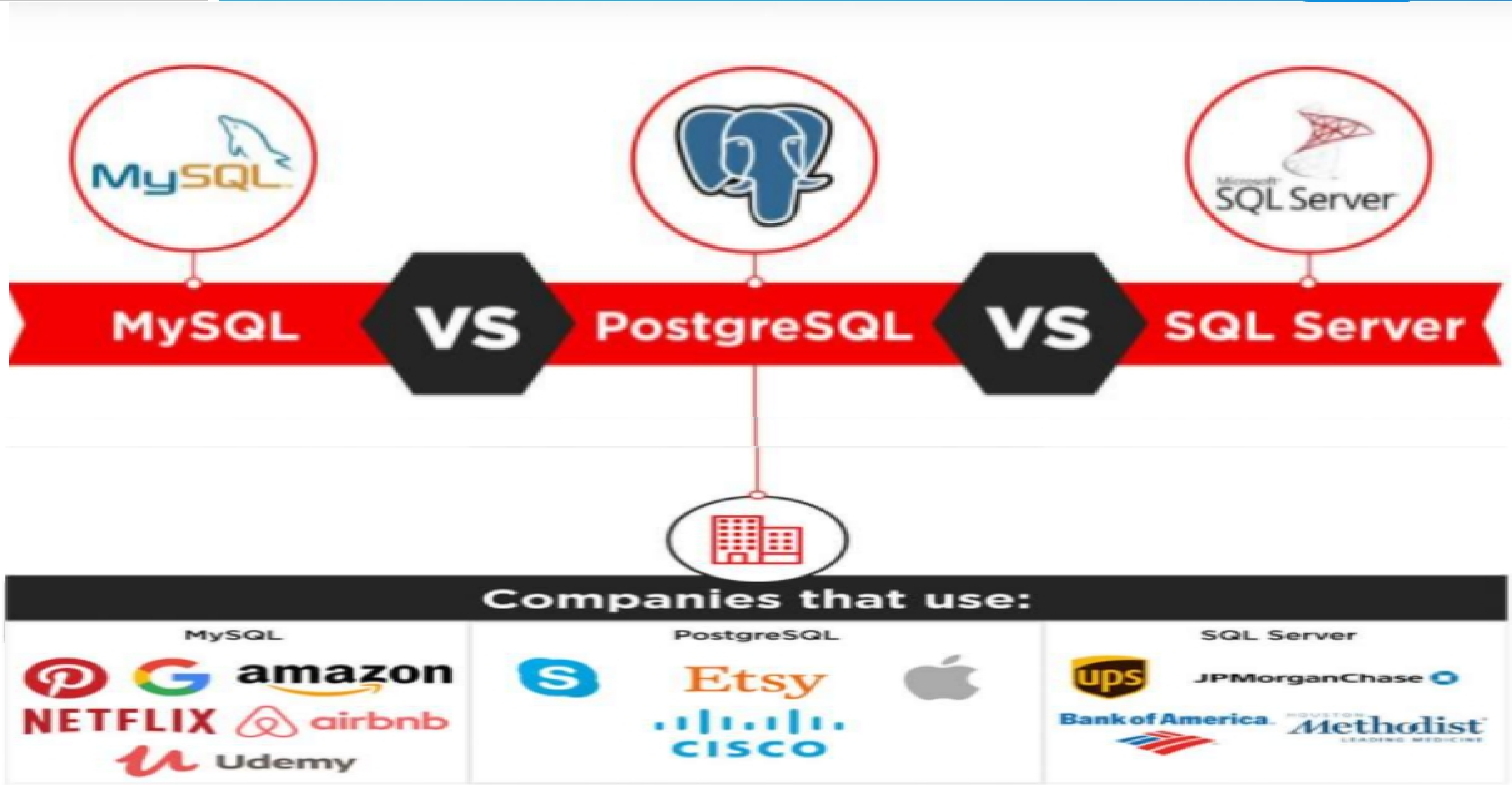
Partitioning:

Allows partitioning databases 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.

Companies that use



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.