Presenting each of the RDBMS and their functionalities

	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.

A comparison between the three RDBMS

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	Supports primary keys, foreign keys, not-null constraints, check constraints, unique constraints, default constraints
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

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