Introduction To Databases - Checkpoint

May 27, 2022

Objective

In this checkpoint, you are asked to prepare a presentation that deals with 3 well known relational RDBMS which are MySQL, PostgreSQL and SQL SERVER .

Instructions

1. Present each of the RDBMS and their functionalities
2. A comparison between the three RDBMS.

Description of MySQL, PostgreSQL and SQL SERVER

MySQL

- Free and Open-source RDBMS that is supported by Oracle Corporation (also available under a variety of proprietary licenses). It is used by many popular websites, including Facebook, Twitter and Youtube

PostgreSQL

- Free and open-source RDBMS designed to handle a range of workloads, from single machines to data warehouses or Web services with many concurrent users. It is the default database for macOS Server

SQL Server

- RDBMS developed by Microsoft, with at least a dozen different editions, each aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users

Comparison between MySQL, PostgreSQL and SQL SERVER

	MySQL	PostgreSQL	SQL Server
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.
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
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

Comparison between MySQL, PostgreSQL and SQL SERVER

	MySQL	PostgreSQL	SQL Server
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
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.
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.
Parallel query execution	MySQL will usually use 1 CPU per query.	Query plans can leverage multiple CPUs	Query plans can leverage multiple CPUs

Comparison between 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
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.

Comparison between MySQL, PostgreSQL and SQL SERVER

	MySQL	PostgreSQL	SQL Server
Indexes	Supports index-organized tables - clustered indexes. Does not support persisted indexes / materialized views	Supports index-organized table, but updates are manual until PostgreSQL 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).

Comparison between MySQL, PostgreSQL and SQL SERVER

	MySQL	PostgreSQL	SQL Server
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
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

Comparison between MySQL, PostgreSQL and SQL SERVER

	MySQL	PostgreSQL	SQL Server
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
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

Thank You.