

CHEATSHEET SQL MONITORING & OPTIMIZATION

SQL HEALTH-CHECK

BE PROACTIVE! Maintain a robust and reliable database system by monitoring SQL metrics and get valuable insights into the health and efficiency of your SQL instance (Postgres, MySQL, or SQL Server):



CPU Usage

- High CPU usage may indicate:
 - 1. Inefficient query** execution
 - 2. Resource contention**
 - 3. Suboptimal query** plans
- Set an alert when capacity reaches 70% to make proactive decisions and improve CPU utilization.



Locking Activity

- **Monitor to prevent** contention issues and deadlocks.



Memory Availability

- Ensure there's **enough memory** for efficient operations.
- **Low memory** can lead to swapping & degrade performance.
- Ensure all (unclustered) indexes fit into **internal memory**.



Buffer Cache Hit Ratio

- Indicates how often **SQL Instance** fulfills requests from memory instead of disk.



Disk Usage

- Monitor especially for **table and index bloat**.
- Slow disks impact **SQL instance's speed**.



Network Volume

- Understand the volume of **incoming and outgoing data**.
- Troubleshoot network-related issues and plan for **network capacity**.



Query Throughput

- Track read and write query throughput.
- Analyze query execution time to identify slow queries.



Replication Lag

- Ensure data consistency across replicas.
- Check for replication errors to ensure reliable data synchronization.

QUERY OPTIMIZATION



Avoid Select*

- Explicitly specify needed columns to reduce resource usage.



Use Wildcards Judiciously

- Wildcards like % in LIKE clauses can be resource-intensive.



Prefer Numeric Comparisons

- Use numeric values for comparisons instead of characters for better performance.



Replace Subqueries with JOINs

- Subqueries can be performance bottlenecks, especially with large result sets.
- Use EXISTS instead of IN for subqueries to stop evaluating once a match is found.



Limit or Paginate Results

- Restrict the number of rows returned to only what's necessary.



Filter Data in Database

- Let the database fetch only the necessary data; avoid filtering data using code.



Simplify JOIN Statements

- Both complex JOIN statements & multiple JOINs can reduce readability & performance.
- Prefer INNER JOINs over OUTER JOINs (LEFT, RIGHT, FULL).

Learn more & take your data management to the next level!

Contact us now

