A database is a collection of organized information for easy access and management. Computer databases generally consist of aggregated data or files that contain information about customers, transactions, or inventories. Regular monitoring of the **database’s performance** is necessary to ensure that it is running properly and to detect issues as they arise.

Here is a short **database monitoring guide** that can assist you in choosing the right tools.

# What is Database Monitoring

**Database monitoring** is the act of measuring a database’s performance in real time to identify problems and any other factors that can cause a problem in future. It helps understand the areas in the database that can be optimized or enhanced to improve efficiency and performance.

This is generally done with **monitoring software tools** that are either inbuilt with the database or installed separately from third-party vendors. Through these tools and interfaces, database administrators can monitor the performance of different database areas that serve various applications. They allow to see the performance of an application’s database through a specialized web interface.

The main purpose of **database monitoring** is to assess how a database server performs, for both software and hardware. This includes taking snapshots of performance indicators at regular intervals to make it easy for determining the exact time when a problem or a bottleneck occurs.

It also helps to gain an insight on the exact cause of a problem at that time and to find the most **optimum solution**. This is necessary because different problems occur at different intervals of time, and it is not possible for a single administrator to monitor the entire database in real time.

# Why Database Monitoring is Important?

Generally, a database is an important part of an information system. It is used to store, query, and update **sensitive business data**. For any database administrator, the performance, availability, and security of a database are important areas of concern. Database monitoring has a number of benefits that signify its importance.

* It helps determine if the performance of a database can be improved
* It identifies security flaws by evaluating the user activity and the application
* It troubleshoots and debugs problems in the application components that use the database.
* It is important to have a **proactive monitoring approach** than a reactive one, since it helps identify issues before they can turn into big accidents.
* When an application slows down or malfunctions, the database is always the first area that is investigated. If monitoring is performed regularly, it is easier to find out any database-related problems and fix the issues.
* Besides performance and security, it also helps automatically check database backups.

# What Database Metrics Should be Monitored?

## Infrastructure

Infrastructure-related **database monitoring metrics** include:

* Memory available
* Percent CPU time that is used by database processes
* Available disk space
* Percent virtual memory use
* Disk queue for waiting I/O

## Availability

## Before you can look at other counters, you should ensure that your database is accessible and available. It also saves you from the trouble of complaining customers before finding out an outage. Availability metrics include:

## Accessibility of database end point and port (e.g. 5432 for PostgreSQL, 3306 for MySQL, etc.)

## Accessibility of database node(s) that use common protocols like Telnet or Ping

## Failover event for master node or upgrade event for peer/slave nodes in clusters with multi-nodes.

## Throughput

## It is measured to create a normal production performance baseline. Metrics vary according to database platforms, but there are some common metrics among all platforms such as:

## Number of active database connections

## Average time for completing a read query

## Connection wait time for endpoints

## Number of read queries in progress or received

## Average time for completing an update, insert or delete command

## Number of update, insert, or delete commands that are in progress or received

## Number of transactions completed

## Heap memory used

## Replication lag between a primary and secondary node

## Data growth and transaction log size percent rate

## Percentage of number of times in-memory data cache was accessed

## Performance

## Performance metrics also vary according to different databases and have to be reported in a particular time-scale. They indicate any potential bottlenecks and it is recommended to create baselines for them.

## Common performance metrics include:

## Number of database lock timeouts

## Number of read or write queries that are blocked or currently waiting

## Number of deadlocks

## Percentage of times the disk-based virtual memory is accessed

## Application traces

## Skewed data distribution in the nodes

## Warnings for unusable indexes or outdated statistics

## Queries that run slower than a certain threshold

## Scheduled Tasks

## These are repetitive tasks in a database. Some systems such as Oracle or Microsoft SQL Server have inbuilt job scheduling options, whereas other use cron jobs or third-party schedulers. Some scheduled jobs to monitor include:

## Database maintenance tasks such as vacuuming, analyzing and updating statistics, reindexing, log rotation, database integrity checks, compaction, etc.

## Application-specific tasks such as archiving and data exportation, etc.

## Regardless of the type of job, the outcome of a scheduled task has to be monitored in terms of success or failure.

## Security

## Database security needs to be monitored in accordance with enterprise security goals. As a bare minimum, it is important to monitor the following:

## New user accounts creation

## Password changes

## Number of failed login attempts

## Database configuration change events

## Logs

## All database engines have some types of logs where they record the information. Logs can consist of one or more files. Database monitoring services should collect, parse and store logs to create metrics from the events that are exposed as a result. Logs can consist of valuable information such as

## User and system queries

## Outputs of scheduled jobs

## Database system events such as errors, startup, shutdown, etc.

# What Tools to Use

Effective **database monitoring tools** are those that can capture, store and analyze the events and metrics and create meaningful results from it. When choosing your database tools, there are different options to consider.

* Some of the database software is included as a value-added product with monitoring tool
* Some database services have their own monitoring tools such as, Oracle Enterprise Manger
* Open-source databases that are commercially distributed can have monitoring tools as well.
* Some database administrators use their own tasks and scripts to capture the metrics and build personalized dashboards.
* **Enterprise Application Performance Monitoring** tools such as the one by [VirtualMetric](https://www.virtualmetric.com/database-monitoring), also provide database plugins.

Whichever tools you use, a good **database monitoring tool** should:

* Gather and store logs and metrics
* Be able to operate without a lot of maintenance by the customer
* Use short probes or light agents for collecting the data
* Have a wide plugin and hook range
* Be able to correlate logs with performance metrics from the database
* Create dashboards that have logs and metrics

VirtualMetric provides a powerful [database monitoring and management solution](https://www.virtualmetric.com/database-monitoring). To find out how it works, get a free 30-day trial [here](https://www.virtualmetric.com/trial).