

# Lecture 11. Database monitoring

# Definition: Database monitoring

**Database monitoring** is the tracking of database performance and resources in order to create and maintain a high performance and highly available application infrastructure.

Databases are at the heart of many enterprises' most critical business processes. Application complexity is on the rise and IT infrastructures are becoming far more diverse. With complex applications working across heterogeneous infrastructures, database monitoring that provides fast, accurate problem resolution is critical to helping IT troubleshoot problems before they affect end-users

# Database monitoring

Monitoring the state of the database ensures better performance and availability of the database. The process of monitoring databases involves checking the availability of free space for all the table spaces and checking for database and server errors.

To prevent critical database error messages or data loss, ensure that there is sufficient free space for all of the table spaces, especially the USERS, and INDX table spaces. You can check the log files that are generated by the database regularly to check if the problems that are encountered by your system are related to database failure.

## Typical monitoring tasks

You must constantly monitor the database to ensure that it is up and running and is not bogged down with heavy jobs that slows performance. The database administrator can receive notification alerts when system errors occur and when resources are being used excessively. The database administrator must also run their own checklists of monitoring tasks, which can include the following:

- Monitor free space in table spaces (especially USERS and INDX because they are most likely to run out of space)
- Check log files for database errors
- Check for free disk space and disk usage (know when the database is reaching a shortage of free disk space)
- Monitor server resources usage and errors (processor, memory, swapping, disk I/O, network I/O)
- Check the status of periodic back ups
- Monitor rollback usage (when rollbacks occur and if they take too long)
- Set alerts (for example, when unique constraints are violated, locks are held for a long time or deadlocks occur, queries run slowly, or resources are used excessively)

## Checking for free space in table spaces

You must constantly monitor the available free space of all table spaces, especially the USERS, and the INDX table spaces, because they are most likely to run out of space. You can check the availability of free space at regular intervals that depend on the initial configuration of free space and the workload. Sufficient free space prevents critical error messages and data loss.

### Listing table space usage in DB2®

You can check for table space usage from the command line by using any of the following commands:

- `db2 connect to dbname`
- `db2 list table spaces show detail`
- `db2pd -db dbname -tablespaces`

## Monitoring slow running queries

There are several reasons for slow running queries, including table fragmentation, data volume, application server process, or server load. You can monitor the `db.log` log file for DELAYED QUERY log entries to locate and determine if there are slow running queries. Each service has its own `db.log` log file. Each of the `db.log` files is located in their respective `$TOP/logs/service/service_Name` directories, where `service_Name` is the unique service name of one of your services.

To ensure that you have current statistics in the `db.log` file, you should update your database statistics before viewing your service's `db.log` file to check for slow running queries.

## Checking for database errors

As the database administrator, you must check the log files that are provided by Oracle or DB2 regularly.

You can find all the DB2 log files at `DB2 instance\sqllib\db2dump`. You can also set up alerts and be notified when unique constraints are violated, when locks are held for a long time or deadlocks occur, and when queries start to run slow. You can look for all Oracle error messages in `$ORACLE_BASE/admin/oracle SID/directory`.

# What is included in monitoring?

For example, categories for SQL Server, MySQL and Oracle database monitoring include:

- Query details (top CPU, slow running, and most frequent)
- Session details (current user connections and locks)
- Scheduled jobs
- Replication details
- Database performance ([buffer](#), [cache](#), [connection](#), [lock](#), and [latch](#))



The following outline is a list of items to take into account when implementing a database monitoring system:

How should you monitor?

- Automatically collect key database, server, and [virtualization](#) performance metrics
- Alert on performance or availability problems for both database and server components and optionally take corrective action
- Generate comprehensive reports to show database utilization and capacity issues
- Correlate database issues with end user response metrics for accurate assessment of application performance

What constitutes a problem?

- KPIs that exceed threshold values
- Alarms or [alerts](#) generated by the database
- Inability to access the database
- Poor database performance and response time
- Running out of key resources the database needs to perform properly ([CPU](#), [Memory](#), [Storage](#), [IO capacity](#))

## What should you do when a problem is identified?

- Prioritize and escalate high severity alerts with text messages or email alerts
- For recurring problems build detailed notes into the alert to speed resolution
- Automate with OS commands or scripts to fix the problem if possible

## What are the benefits of monitoring and tracking?

- Reduce the time and resources needed for supporting databases and the underlying IT infrastructure
- Improve end-user performance
- Improve capacity planning by determining whether issues can be resolved by upsizing database or server configurations
- Proactively troubleshoot performance problems before they reach end-users

Thank you for your attention!