

Troubleshooting Common Issues

Sure! Let's talk about troubleshooting common issues in databases.

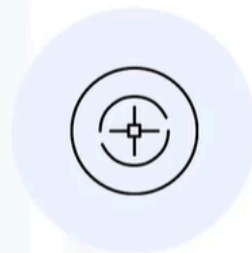
When we troubleshoot a database, it's like being a detective trying to solve a mystery. We start by asking questions to understand the problem better. For example, we might ask: What exactly is going wrong? Is it slow? Is it not connecting? Who is experiencing the issue? Is it just one person or many? By gathering this information, we can pinpoint where the problem is happening and under what conditions.

Common issues often arise from three main areas: poor performance, improper configuration, and poor connectivity. Imagine if your favorite restaurant had a slow kitchen (poor performance), the menu was all mixed up (improper configuration), or the delivery service was unreliable (poor connectivity). Each of these problems can lead to a frustrating experience. Similarly, in databases, if the server is not set up correctly or if there's a bad connection, it can lead to slow responses or errors.

Troubleshooting basics

Troubleshooting is a process of identifying and then solving a problem

It begins by answering the following questions:



Troubleshooting basics

- What are the symptoms?
- Where is the problem happening?
- When does the problem happen?
- Under which conditions does the problem happen?
- Is the problem reproducible?



- Troubleshooting is a process of identifying and solving a problem. It begins by answering the following questions: What are the symptoms? Who or what is reporting the problem? Where is the problem happening? Is it specific to a platform, environment, application, or something else? When does the problem happen? Does it happen at a specific time, multiple times, or under specific conditions? Under which conditions does the problem happen? Is the problem reproducible?

Common problems



Performance



Configuration



Connectivity

- The most common problems encountered with databases are typically caused by one or more of the following: Poor Performance. Poor performance can result in slow response to user queries or applications accessing the database. Improper Configuration. Improperly configured clients, servers, or databases can cause a wide range of problems, including poor performance, crashes, errors, or even database corruption. Poor Connectivity. Poor connectivity can cause poor performance, time outs, or a variety of errors when interacting with the database.

Common performance issues

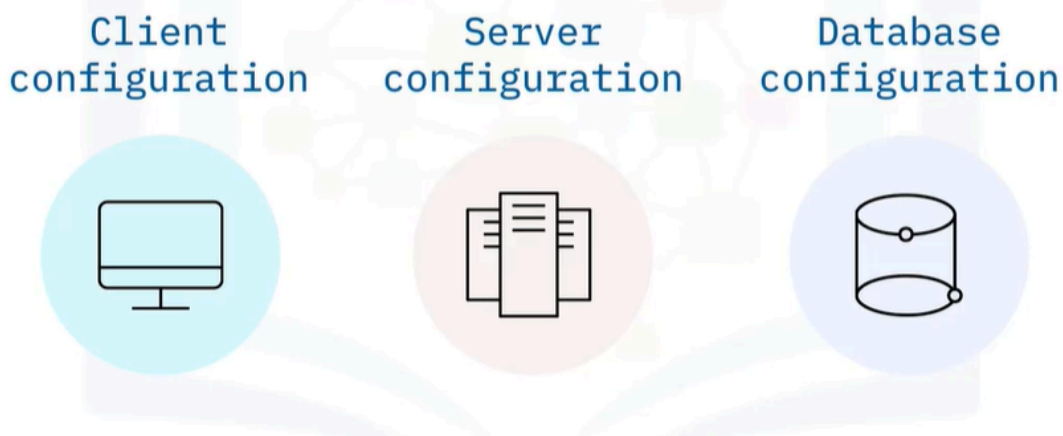
Poor performance is often caused by:



- Poor performance is usually caused by high latency for disk reads and writes, slow processing time by the server, or a poor connection between the server and client. Any of these could be rooted in one or more of the following: Inadequate server hardware or configuration. For example, the server could be running out of disk space, memory, or processing power. Increasing these resources on the database server is called vertical scaling. Adding additional database partitions and shards, to improve performance is called horizontal scaling. Improper configuration. A poorly configured database may be operational but still unable to meet demand. For example, it may need to allow more connections, or it may need changes to its buffering and indexing settings to keep up with queries and return results quickly. Connectivity. A slow or poor network connection, or limited bandwidth between the client and

database can cause high latency and processing times. Queries and Application Logic. A poorly written database query or improper application logic (such as unnecessary locking of database objects) can also result in performance issues.

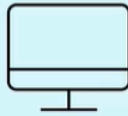
Common configuration issues



- An improperly or sub-optimally configured client, server, or database can cause any number of problems that can manifest in many ways. For example: A user with an incorrectly configured client or incorrect driver might be unable to connect to the database. A poorly or incorrectly configured server can reduce performance, cause time outs or any number of possible errors. The database may need to be configured to allow more connections or other settings, like caching or indexing, and may need to be adjusted to correct problems or improve overall performance.

Client configuration issues

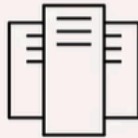
- Incorrect login, password or authentication type
- Incorrect connection configuration
- Incorrect driver version



-
- Common issues with client configuration can be caused by incorrect login credentials, an incorrect host name or IP address, or even a corrupt or outdated connection driver. To fix these problems, check the client's driver configuration and verify the following: The username and password specified in the connection settings are correct. Be sure to verify that the client is also configured to use the correct type of authentication, such as Windows or SQL authentication, for example. The connection configuration settings, such as IP address, host name, and server name, are correct. The driver version for the database application is up to date and correct.

Server configuration issues

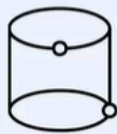
- Out of memory
- Out of disk space
- Inadequate processing power
- Disk fragmentation
- Improper storage configuration
- Bugs in OS or RDBMS software



- Server configuration also significantly impacts performance and operation. Some examples of things you might change or configure to improve performance or correct a problem include: Add more physical RAM or increase the memory assigned to the server. Add more physical disk space or assign additional disk space to the server. Consider upgrading the CPU or assigning more processing power to the server. Consider defragmenting the hard disk. Fragmented data degrades overall performance. Sometimes configuring the storage system appropriately can alleviate performance issues, for example, placing frequently accessed tables on a faster disk. Bugs in operating systems or in RDBMS software can result in errors and server crashes, so ensure you regularly apply software patches and security updates to guard against this.

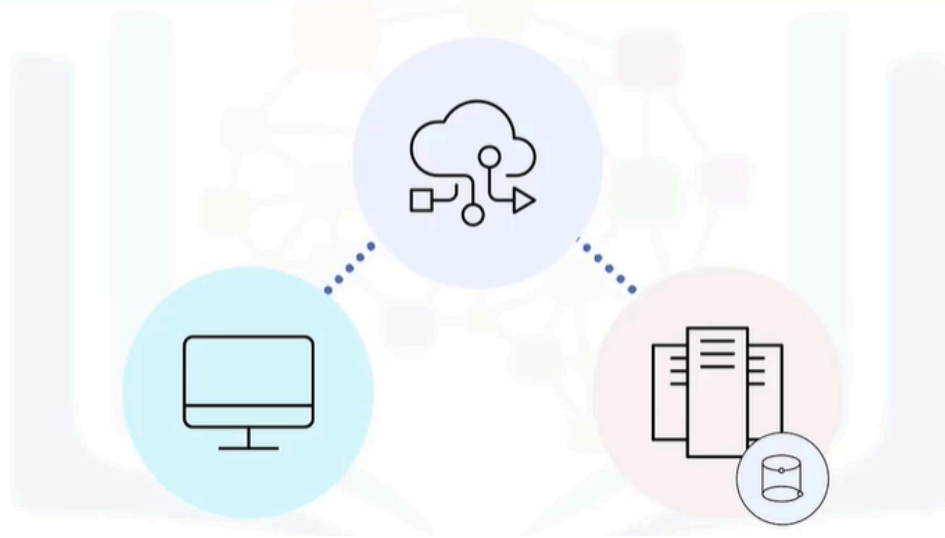
Database configuration issues

- Insufficient database connections
- Insufficient buffering
- Indexing



- The configuration of the database is something you need to monitor and continually evaluate to ensure it meets demand. Some examples of configuration settings you might need to change or correct are: Increase the number of allowed connections to the database to meet increasing demand. Change database buffering to improve performance. Change database indexing to improve performance.

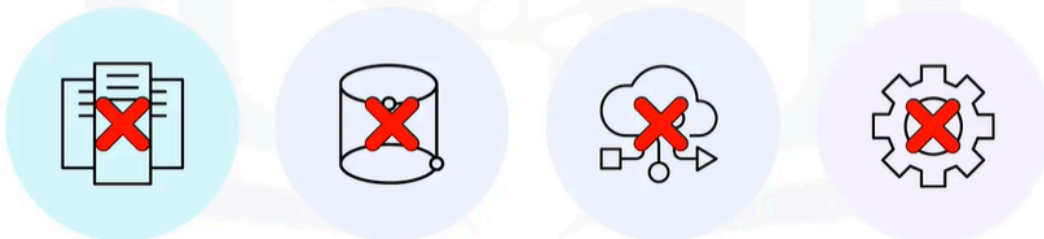
Connectivity issues



- Poor connectivity between the client and the database server can cause a wide range of problems, including poor performance, error messages, or loss of function.

Examples of connectivity issues

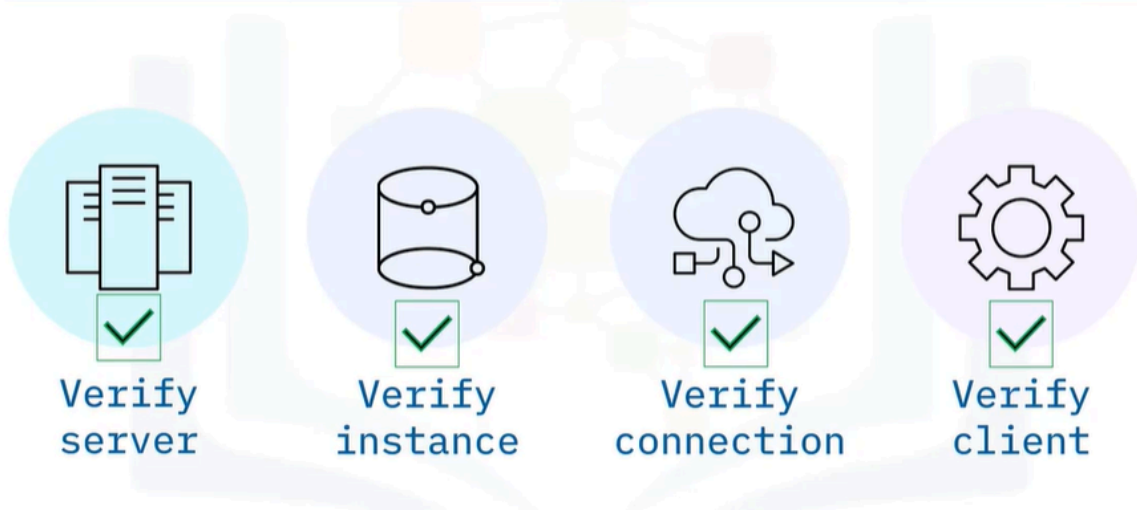
- Database server cannot be reached
- Database instance is not running
- Client login credentials or security are incorrect
- Client configuration is incorrect



- Some of the most common connectivity problems are typically caused by one of the following: The database server cannot be reached or is not running

properly. The database instance on the server is not running properly. The client login credentials are incorrect, or missing security settings such as for SSL connections. The client configuration is incorrect.

Common connectivity solutions



- Here are some common methods to help troubleshoot and solve basic connectivity issues. Verify that the database server is running properly. The exact procedure depends on your configuration and environment. For example, you may need to physically check an on-premises server. Or you may need to verify that a virtual machine in a cloud service is running. Next, verify that the database instance on the server is running. This process varies depending on operating system and database. For example, on a Windows-based system you could use the Task Manager to verify that the instance is running. On a Db2 configuration, you could run `db2cmd.exe` and then issue commands in the command line. Verify that the database can be reached from the client. A common method is to use the PING command from the client to communicate with the server's IP address or host name. Finally, verify that the client connection driver is configured correctly. For example, make sure the user name and password for the connection are correct, and that other settings like IP address or host name, or security and encryption protocols are also correct.

Troubleshooting tools



- Performance monitoring, reports, and server and database logs help identify performance bottlenecks and determine the best way to correct them. Performance monitoring helps identify potential network, server, and database issues before they occur and helps determine where improvements can be made. Dashboards can monitor databases in real time and provide an early warning system for problems before they affect users, in addition to tracking historical performance and other metrics. Server and database logs help identify a problem and when it occurred.

Summary

In this video, you learned that:

- Common database problems caused by poor performance, improper configuration, poor connectivity
 - Poor performance caused by high latency for disk r/w, slow server processing, poor client/server connection
 - Server issues such as inadequate hardware resources or misconfigured settings impact performance
 - Common connectivity problems are inability to connect to database, database server/instance not running, and client logins
 - Performance monitoring, dashboards and reports, and server/database logs help identify bottlenecks
-