

Troubleshooting Common Issues

Objectives

After watching this video, you will be able to:

- Describe common performance, configuration, and connection issues
- Describe basic troubleshooting and solutions for common issues
- Describe tools used to identify, prevent, and solve common database issues

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?

Common problems

wide range of problems such as poor performance, crashes, errors, or even database corruption



slow response to queries
or accessing the database



Configuration



Connectivity

Common performance issues

Poor performance is often caused by:



Inadequate hardware



Server or database configuration



Network connectivity



Queries and application logic

Common configuration issues

Client configuration



Server configuration



Database configuration



Client configuration issues

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



IP, host name,
server name

Server configuration issues

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

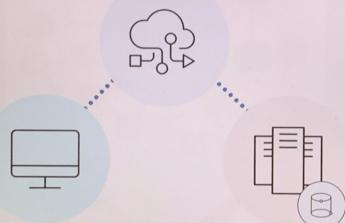


Database configuration issues

- Insufficient database connections
- Insufficient buffering
- Indexing



Connectivity issues



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



Common connectivity solutions



Verify server



Verify instance



Verify connection



Verify client

Troubleshooting tools



Monitoring tools



Dashboards and reports



Server and database logs

Using Status Variables, Error Codes, and Documentation

Objectives

After watching this video, you will be able to:

- Describe how to obtain server status information
- Explain how to retrieve error codes and information
- Describe how to use RDBMS documentation for troubleshooting

Getting server status

- Databases have utilities to check health and operational status
 - Command line
 - GUI

Getting server status

```
# SERVICE MYSQL STATUS  
[dbadm@example etc]$ su - root  
Password:  
[root@example ~]#  
[root@example ~]#  
[root@example ~]# service mysql  
status  
MySQL running (5089) [ OK ]  
[root@example ~]#
```

Getting server status

→ command to monitor the status in each database

- Db2
 - > db2pd
- MySQL
 - # SHOW STATUS
- PostgreSQL
 - > Pg_isready

Using status variables

```
# SHOW STATUS
```

represent status for some aspect of server itself

```
SHOW GLOBAL STATUS
```

represent value for the current connection

```
SHOW SESSION STATUS
```

can be called Local status

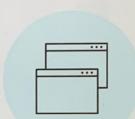
Using status variables

```
SHOW STATUS LIKE 'pattern';  
SHOW STATUS LIKE 'Key%';
```

```
1 mysql> SHOW STATUS LIKE 'Key%';  
2 +-----+  
3 | Variable_name | Value |  
4 +-----+  
5 | Key_blocks_used | 14955 |  
6 | Key_read_requests | 96854827 |  
7 | Key_reads | 162040 |  
8 | Key_write_requests | 7589728 |  
9 | Key_writes | 3813196 |  
10 +-----+
```

Using GUIs to get server status

Windows SQL Server GUI Tools



- Activity Monitor
- System Monitor

Reviewing logs

log general server activity, connectivity, and other aspect of the server



Server and OS Logs

log information and error specific to the database being operated, such as MySQL, PostgreSQL



Database Error Logs

Log file examples (SQL)

Created every time server started



contains info. and error events



an optional log file that tracks all database configuration changes



Error Log Event Log Trace Log

Error code example (SQL)

SQL Server Login name of procedure that caused the problem



Connection Failed:
SQL State: '08004'
SQL Server Error: 4060
Server rejected the connection;
Access to the requested database
has been denied

Decoding errors

Where to find documentation and troubleshooting help:

- Db2: ibm.com/docs/db2
- PostgreSQL: postgresql.org/docs
- MSSQL: docs.microsoft.com/sql
- MySQL: dev.mysql.com/doc/

Using Logs for Troubleshooting

Objectives

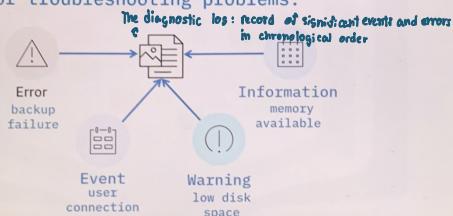
- After watching this video, you will be able to:
- Describe diagnostic logs and their purpose
 - List various types of diagnostic logs
 - Identify information in a database diagnostic logs
 - Explain where diagnostic logs are stored and how to access them

Troubleshooting with logs

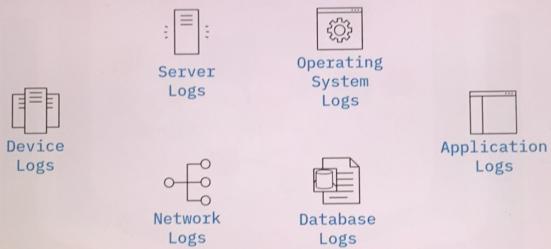
- Systematic process used to locate faults or errors and provide detail on how to correct hardware and software issues
- Approaching troubleshooting logically and methodically is essential to successful resolution
- Troubleshooting is the main reason to create diagnostic logs

What are diagnostic logs?

Diagnostic logs provide chronological records of events and errors in a particular component or application and can be used for troubleshooting problems.

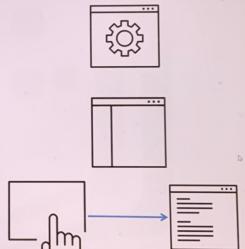


Types of diagnostic logs

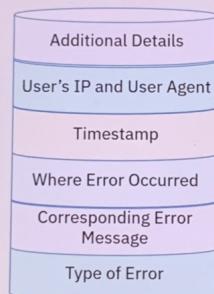


Working with log files

- May be possible to configure location of log
- Many log files are in plain text format
- Some logs may require special tools to view or filter contents



Diagnostic log contents



Db2 diagnostic log

```
additional details such as the DB2 software
2006-02-09-18.07.31.059000-300 TIME stamp: LEVEL: Event
PID : 2140 TID : 2064 PROC : db2start.exe-> type of message
INSTANCE: DB2 NODE : 000
FUNCTION: Db2, RAS/PD component, _pdlogint, probe:120
START : New Diagnostic Log file
DATA #1 : Build Level: 1564 bytes
REFERENCE: 0x00000000-0x00000000 Db2 code release "SQL09010"
with level identifier "01010107".
Informational tokens are "DB2 v9.1.0.190", "s060121", "", Fix Pack "0".
DATA #2 : System Info, 1564 bytes
System: WIN32_NT MYSRVR Service Pack 2 5.1 x86 Family 15, model 2, stepping 4
Physical Memory(MB): total:1624 free:1527 available:617
Virtual Memory(MB): total:2462 free:2336
Swap Memory(MB): total:1438 free:2213
Information in this record is only valid at the time when this file was created
(see this record's time stamp)

$> db2 get DBM CFG show detail | grep DIAG
Diagnostic data directory path (DIAGPATH) = /home/db2inst1/sql1ib/db2dump
Diagnostic error capture level (DIAGLEVEL) = 4
```

PostgreSQL server logs

- log_destination
 - syslog (Linux/Unix)
 - event log (Windows)
 - csvlog
 - stderr (default)
- log_directory
- log_filename

```
postgres=# show log destination ;
-----
stderr
(1 row)

postgres=# show log directory ;
log directory -----
-----
pg_log
(1 row)

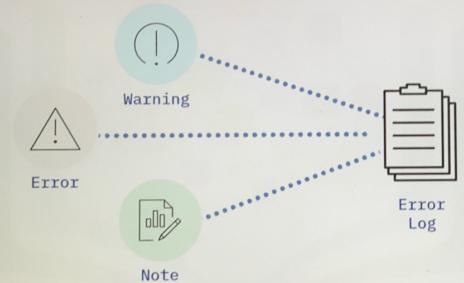
postgres=# show log filename ;
log filename -----
-----
postgresql-%Y-%m-%d_%H%M%S.log
(1 row)
```

MySQL server logs

MySQL has several log for troubleshooting:

- General Query log
 - Connections and queries received from clients
- Slow Query log
 - Queries that take longer than log_query_time
- Error log
 - Diagnostic messages for mysqld

MySQL error log



MySQL error log

- Can be configured to write to:
 - event log - enabled by default in Windows
 - syslog - used on Linux or Unix
 - stderr - enabled by default on Linux or Unix
 - specific file
- Specify using log_error

```
[mysqld]
log_error = /var/log/mysql/error.log
```

MySQL error log

Configure how much is stored in the log:

Level	Includes	Default
1	Errors	
2	Errors and warnings	
3	Errors, warnings and notes	Default

- Use log_err_verbosity:

```
[mysqld]
log_error_verbosity=2 # error and warning messages only
```

Overview of Automating Database Tasks

Objectives

After watching this video, you will be able to:

- Identify what database automated tasks are
- Explain the benefits of using scripting to automate tasks
- List some of the database automation tasks
- Describe automating maintenance tasks benefits
- Recognize the vital role automation plays in database testing

Identifying automated tasks

Leverages unattended processes and self-updating procedures
to make administration tasks simpler and quicker



Fewer deployment errors/higher reliability and speed on change implementations



Enables staff to focus on more important tasks and coding

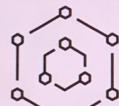


Ideal tasks to automate are time-consuming and repetitive ones such as database health check, alert and server/database maintenance



Using script to automate tasks

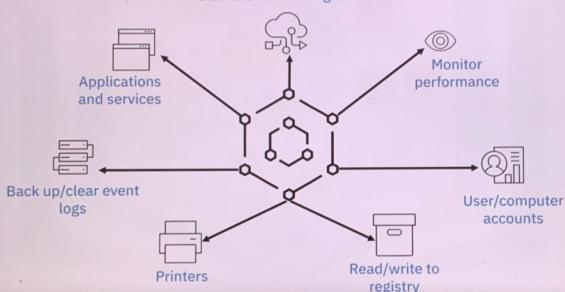
↳ process of using software to leverage and reuse existing scripts
↳ deliver automation



Automation Software

Using script to automate tasks

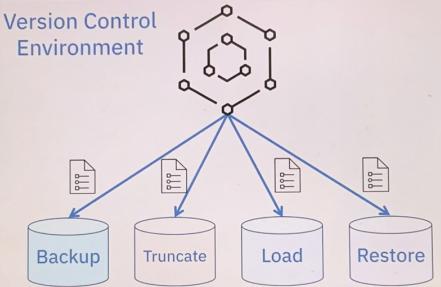
Automate networking tasks



Using script to automate tasks

Version Control Environment

- Keep database in sync with code



Using script to automate tasks

Database Scripts

↳ automation tasks
DBA tasks

Error Notifications

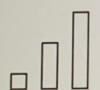


Cron Jobs

Moving Archive Logs
Scheduling
shell Scripts

Advantages of automating tasks

Increase throughput and productivity



Improve quality or increase predictability of quality



Improve consistency of process or product



Advantages of automating tasks

Increase consistency of outputs or results



Free up staff to perform other activities



Provide higher-level jobs in automated processes



Automated database task examples



Database health check
→ Now healthy and efficient the system is



Typical automated database tasks



Trace file cleanup

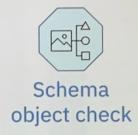
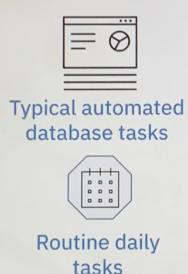
the process of deleting the chronological log of messages and errors written out by the database



the process where the system gathers a collection of names, definitions and attr. dictionary statistics



Automated database task examples



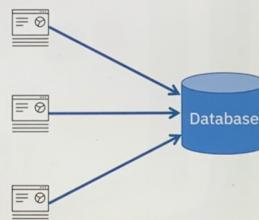
Routine daily tasks

Automating database testing

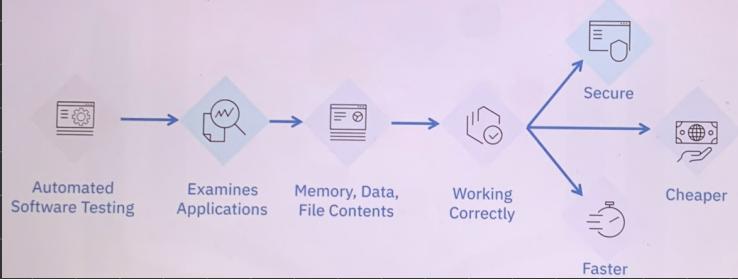
- Information in database is correct and running properly within controlled testing environment
- Schema
- Tables
- Triggers

Automating database testing

- Prevents data loss
- Saves aborted transaction data
- Prohibits unauthorized access
- Checks data integrity and consistency



Automating database testing benefits

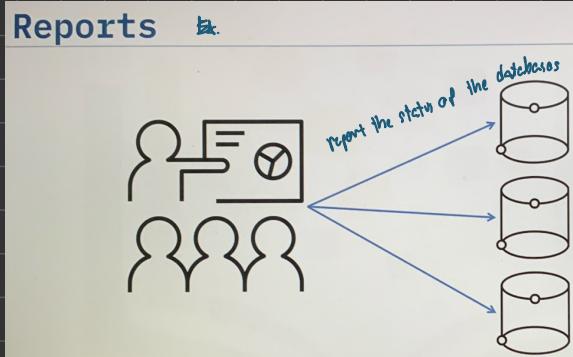


Automating Reports and Alerts

Objectives

- After watching this video, you will be able to:
- Distinguish between reports, notifications, and alerts
 - Describe how DBAs use reports, notifications, and alerts
 - Identify some of the common alerts based on best practices
 - Explain some ways of automating reports, notifications, and alerts

Reports

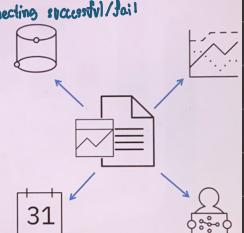


Automated reports



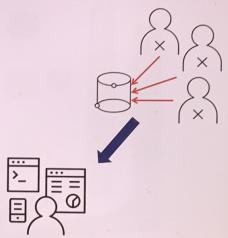
Reports

- Health status of databases
- No. of users connecting successful/fail
- Address issues/problems
- Keep track of trends over time
- Predict future needs
- Regular schedule: daily, weekly, or monthly



Notifications

- Bring an event to the DBA's attention
- Raise awareness of specific events
- Display on dashboards or sent through email



Alerts

Configurable severity thresholds

Alerts

Alerts bring urgent issues to your attention



Alerts

Configurable severity thresholds

best practice is to send out warning alerts when threshold for the specified event reaches

85%



Configurable severity thresholds

Alerts

Configurable severity thresholds



95%



Automate reports

- Configure content and frequency
- Use a sample report
- Configure your own



Automation

- Configure through:
 - Graphical interface
 - Command line
 - Script
- Varies by RDBMS

