

Prepared by Dr. Nermin Ewais

CYB 770
IT Forensics

Lecture 07



Database Forensic

Lecture 07

CYB 770
IT Forensics

Prepared by
Dr. Nermin Ewais

- Database Overview
- Database Forensics Overview
- Relation Database
- Database Engines
- MS SQL Server
- MySQL

- A database is a systematic collection of data.
- They support electronic storage and manipulation of data.
- Databases make data management easy.

Database Types

- **Relational database:** It is the most efficient way to access structured information. The data is organized into a set of tables that has columns and rows.
- **NoSQL database:** NoSQL is a nonrelational database that contains unstructured and semi-structured data. It rose in popularity as web applications came to be commonly used and became more complex.

- **Database Management System or DBMS** is a type of software that helps manage a database. It is used to find and store information within a database. It can be modified according to the user's needs. It adds a layer of security to the database.

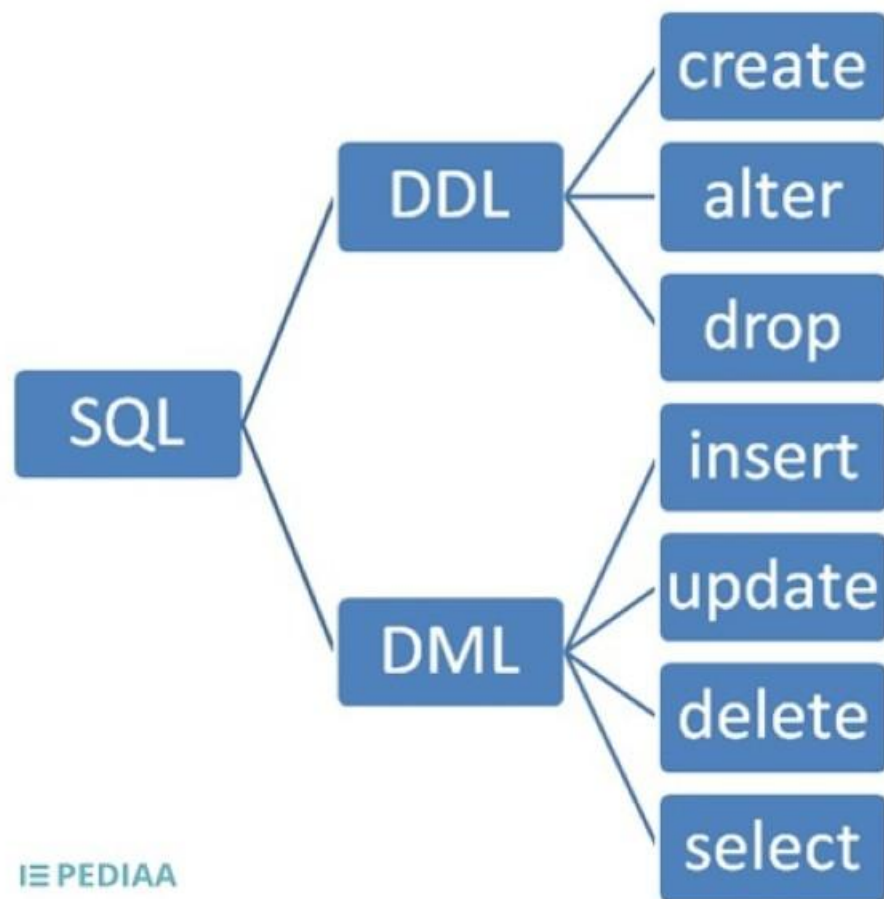
Example of DBMS

- **Microsoft SQL Server** – SQL Server, developed by Microsoft, is a Relational Database Management System. It is built on SQL, the standard query language for Database Management Systems.
- **Oracle Database** – Developed by Oracle Corporation, Oracle Database is based on a multi-model DBMS. It is widely used when processing online transactions.
- **MySQL** – Based on Structured Query Language (SQL), MySQL is a Relational Database Management System. It is used in e-commerce platforms, data warehousing, etc. It is widely used as a web Database Management System.
- **IBM Db2** – Db2 is a Relational Database Management System developed by IBM. It is designed to analyze, store and retrieve data efficiently.

- Database forensics is a subfield of digital forensic science concerned with the forensic examination of databases and their metadata.

- It is the use of electronic data stored in the database to reconstruct the clues, detect crime, and accomplish case cracking.
- The database forensic follows the standard forensic method and employs investigative techniques on database contents and metadata.
- Forensic analysis of a database may involve inspecting and validating the timestamps associated with the update time of a row in a relational table to validate a database user's actions.

- Directly focuses on the identification, preservation and analysis of database data
- Identify data **pre and post transaction**
- Retrace user **DML & DDL operations (DDL is Data Definition Language, DML is Data Manipulation Language).**
- Recover previously deleted data rows
- Can help prove/disprove the occurrence of a data security breach.



- **DDL is Data Definition Language** which is used to define data structures. For example: create table, alter table are instructions in SQL.
- **DML: DML is Data Manipulation Language** which is used to manipulate data itself. For example: insert, update, delete are instructions in SQL.

CREATE table

table_name

```
(  
Column1 datatype (size),  
column2 datatype (size),  
.  
.  
columnN datatype(size)  
);
```

```
INSERT INTO table_name VALUES  
(value1, value2, value3,...);
```

Database System

Volatile Data

MEMORY
Data

Running
Processes

RAM

Status-
information

Query-
Cache

Non-volatile Data

Data

Database
Structre

DBMS
Version

Log Files

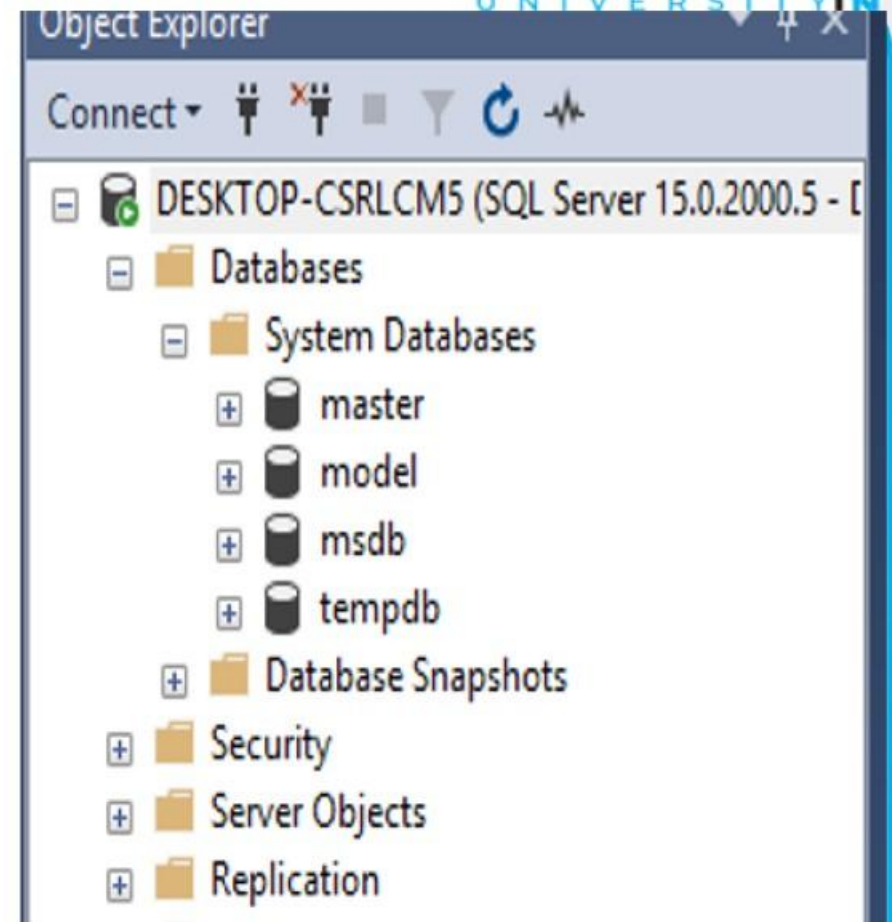
- MS SQL Server
- MySQL

MS SQL Server

- MS SQL Server is a database server
- Product of Microsoft
- Enables user to write queries and other SQL statements and execute them

By default, SQL server has 4 databases

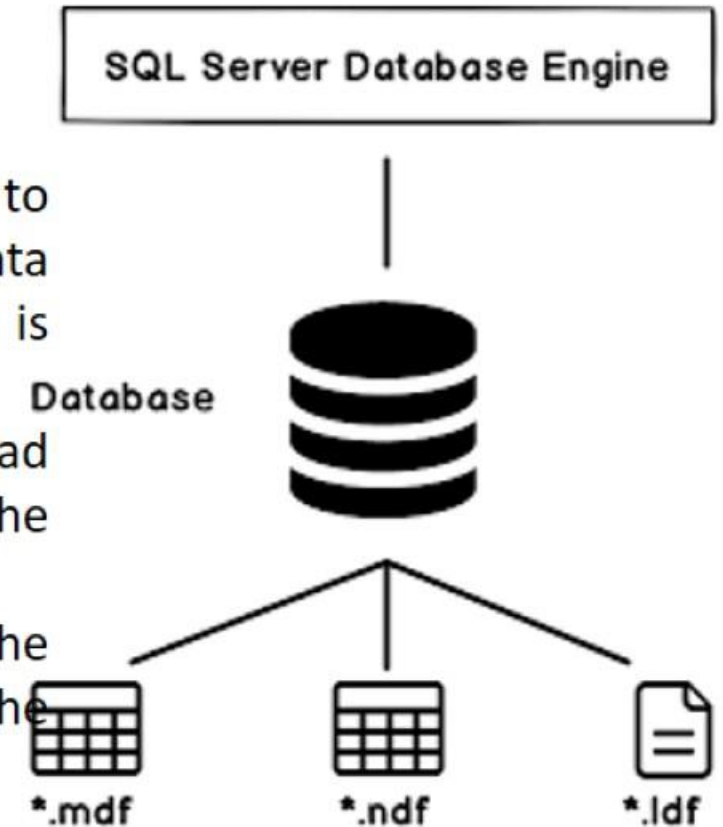
- **Master** : System defined stored procedures, login details, configuration settings etc
- **Model** : Template for creating a database
- **Tempdb** : Stores temporary tables. This db is created when the server starts and dropped when the server shuts down
- **Msdb** : Has tables that have details with respect to alerts, jobs. Deals with SQL Server Agent Service



- By default the size of a database is 1 MB

A database consists of:

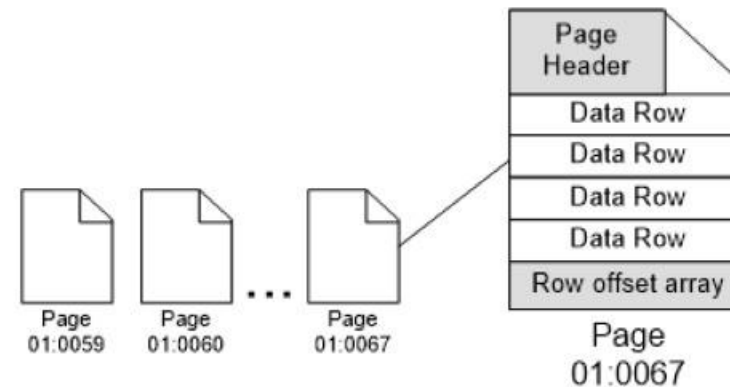
- **Primary:** Contains startup information for the database and points to the other files in the database. Every database has one primary data file. The recommended file name extension for primary data files is .mdf.
- **Secondary** Optional user-defined data files. Data can be spread across multiple disks by putting each file on a different disk drive. The recommended file name extension for secondary data files is .ndf.
- **Transaction Log** The log holds information used to recover the database. There must be at least one log file for each database. The recommended file name extension for transaction logs is .ldf.



Database files

- Data files (.mdf) contain the actual data
- Consists of multiple data pages
- Data rows can be fixed or variable length

- Each **data file** contains multiple data pages (about 8 kb)
- Data pages are divided into :
- **Page Header** : presents page ID , page type
- **Data row** : Store the actual data
- **Offset Table** : Points the actual location of actual data



- Log files (.ldf) hold data required to roll-back transactions and recover the database
- Physical log files consist of multiple Virtual Log Files (VLF)
- A VLF is the unit of truncation for the transaction log

VLF #1 (Inactive)	VLF #2 (Inactive)	VLF #3 (Active)	VLF #4 (Inactive)	Free Space
----------------------	----------------------	--------------------	----------------------	---------------

SQL Server data resides natively within SQL Server and stored externally within the native Windows operating system

- Trace files
- System event logs
- SQL Server error logs

- **Purpose:** Monitoring and Diagnostics
- **What it captures:** Database **activity** and events
- **Content:** SQL queries, login attempts, errors, performance metrics, deadlock graphs
- **Analogy:** A **security camera recording** or **phone tap** - it records what people are doing and saying
- **Usage:** For performance tuning, debugging, auditing, troubleshooting
- **Optional:** You create it only when you need to monitor something

Transaction Log File

- **Purpose:** Data Integrity and Recovery
- **What it captures:** Database **changes** at the physical level
- **Content:** "Before and after" images of modified data, transaction begin/end markers
- **Analogy:** A **bank ledger** - it records every financial transaction to ensure accuracy and can reconstruct account balances
- **Usage:** For crash recovery, rolling back transactions, database restoration, maintaining ACID properties
- **Mandatory:** Every database **MUST** have one - it's critical for operation

- The transaction log records all transactions and the modifications made to the database by each transaction. It is crucial for:
- **Individual transaction recovery:** Rolling back incomplete transactions.
- **Database recovery:** Bringing the database to a consistent state after a system failure.
- **Point-in-time recovery:** Restoring a database to a specific point in time (in Full or Bulk-Logged recovery models).
- **High availability and disaster recovery solutions:** Supporting features like Always On Availability Groups, database mirroring, and log shipping.

```
Select * FROM sys.fn_dblog(NULL,NULL)
```

	Current LSN	Operation	Context	Transaction ID	LogBlockGeneration	Tag Bits	Log Record Fixed Length	Log Record Length
1	00000058:000001a8:0001	LOP_BEGIN_XACT	LCX_NULL	0000:000057bb	0	0x0000	76	168
2	00000058:000001a8:0002	LOP_BEGIN_XACT	LCX_NULL	0000:000057bc	0	0x0000	76	168
3	00000058:000001a8:0003	LOP_MODIFY_ROW	LCX_BOOT_PAGE	0000:000057bc	0	0x0000	62	100
4	00000058:000001a8:0004	LOP_MODIFY_ROW	LCX_BOOT_PAGE	0000:000057bc	0	0x0000	62	100
5	00000058:000001a8:0005	LOP_MODIFY_ROW	LCX_BOOT_PAGE	0000:000057bc	0	0x0000	62	100
6	00000058:000001a8:0006	LOP_MODIFY_ROW	LCX_BOOT_PAGE	0000:000057bc	0	0x0000	62	92
7	00000058:000001a8:0007	LOP_COMMIT_XACT	LCX_NULL	0000:000057bc	0	0x0000	80	84
8	00000058:000001a8:0008	LOP_COMMIT_XACT	LCX_NULL	0000:000057bb	0	0x0000	80	84
9	00000058:000001aa:0001	LOP_SHRINK_NOOP	LCX_DIAGNOSTICS	0000:00000000	0	0x0000	24	128

SQL Server Error Log:

- **Purpose:**
- This log records information about SQL Server startup, shutdown, errors, warnings, and other significant events related to the SQL Server instance itself.

Select logs

- ☒ SQL Server
 - ☐ Current - 3/13/2017 1:15:00 PM
 - ☒ Archive #1 - 3/9/2017 11:08:00 AM
 - ☐ Archive #2 - 2/10/2017 12:02:00 PM
 - ☐ Archive #3 - 1/17/2017 12:16:00 PM
 - ☐ Archive #4 - 1/13/2017 2:40:00 PM
 - ☐ Archive #5 - 1/12/2017 2:23:00 PM
 - ☐ Archive #6 - 1/12/2017 2:22:00 PM
- ☐ SQL Server Agent
- ☐ Database Mail
- ☐ Windows NT

< III >

Status

Last Refresh:

3/13/2017 1:15:54 PM

Load Log Export Refresh Filter ... Search ... Stop

Log file summary: No filter applied

Date	Source	Message
3/9/2017 9:26:03 AM	Logon	Login failed for user 'S[REDACTED]'.
3/9/2017 9:26:03 AM	Logon	Error: 18456, Severity: 14, State: 38.
3/9/2017 9:26:02 AM	Logon	Login failed for user 'S[REDACTED]'.
3/9/2017 9:26:02 AM	Logon	Error: 18456, Severity: 14, State: 38.
3/9/2017 9:26:02 AM	Logon	Login failed for user 'S[REDACTED]'.
3/9/2017 9:26:02 AM	Logon	Error: 18456, Severity: 14, State: 38.
3/9/2017 9:26:02 AM	Logon	Login failed for user 'S[REDACTED]'.
3/9/2017 9:26:02 AM	Logon	Error: 18456, Severity: 14, State: 38.
3/9/2017 9:26:02 AM	Logon	Login failed for user 'S[REDACTED]'.
3/9/2017 9:26:02 AM	Logon	Error: 18456, Severity: 14, State: 38.
3/9/2017 9:26:01 AM	Logon	Login failed for user 'S[REDACTED]'.
3/9/2017 9:26:01 AM	Logon	Error: 18456, Severity: 14, State: 38.
3/9/2017 9:26:01 AM	Logon	Login failed for user 'S[REDACTED]'.
3/9/2017 9:26:01 AM	Logon	Error: 18456, Severity: 14, State: 38.
3/9/2017 9:26:01 AM	Logon	Login failed for user 'S[REDACTED]'.
3/9/2017 9:26:01 AM	Logon	Error: 18456, Severity: 14, State: 38.
3/9/2017 9:26:01 AM	Logon	Login failed for user 'S[REDACTED]'.
3/9/2017 9:26:01 AM	Logon	Error: 18456, Severity: 14, State: 38.
3/9/2017 9:26:01 AM	Logon	Login failed for user 'S[REDACTED]'.
3/9/2017 9:26:01 AM	Logon	Error: 18456, Severity: 14, State: 38.
3/9/2017 9:26:00 AM	Logon	Login failed for user 'S[REDACTED]'.
3/9/2017 9:26:00 AM	Logon	Error: 18456, Severity: 14, State: 38.
3/9/2017 9:26:00 AM	Logon	Login failed for user 'S[REDACTED]'.
3/9/2017 9:26:00 AM	Logon	Error: 18456, Severity: 14, State: 38.
3/9/2017 9:26:00 AM	Logon	Login failed for user 'S[REDACTED]'.
3/9/2017 9:26:00 AM	Logon	Error: 18456, Severity: 14, State: 38.
3/9/2017 9:26:00 AM	Logon	Login failed for user 'S[REDACTED]'.

Location of Files to Restore the Evidence

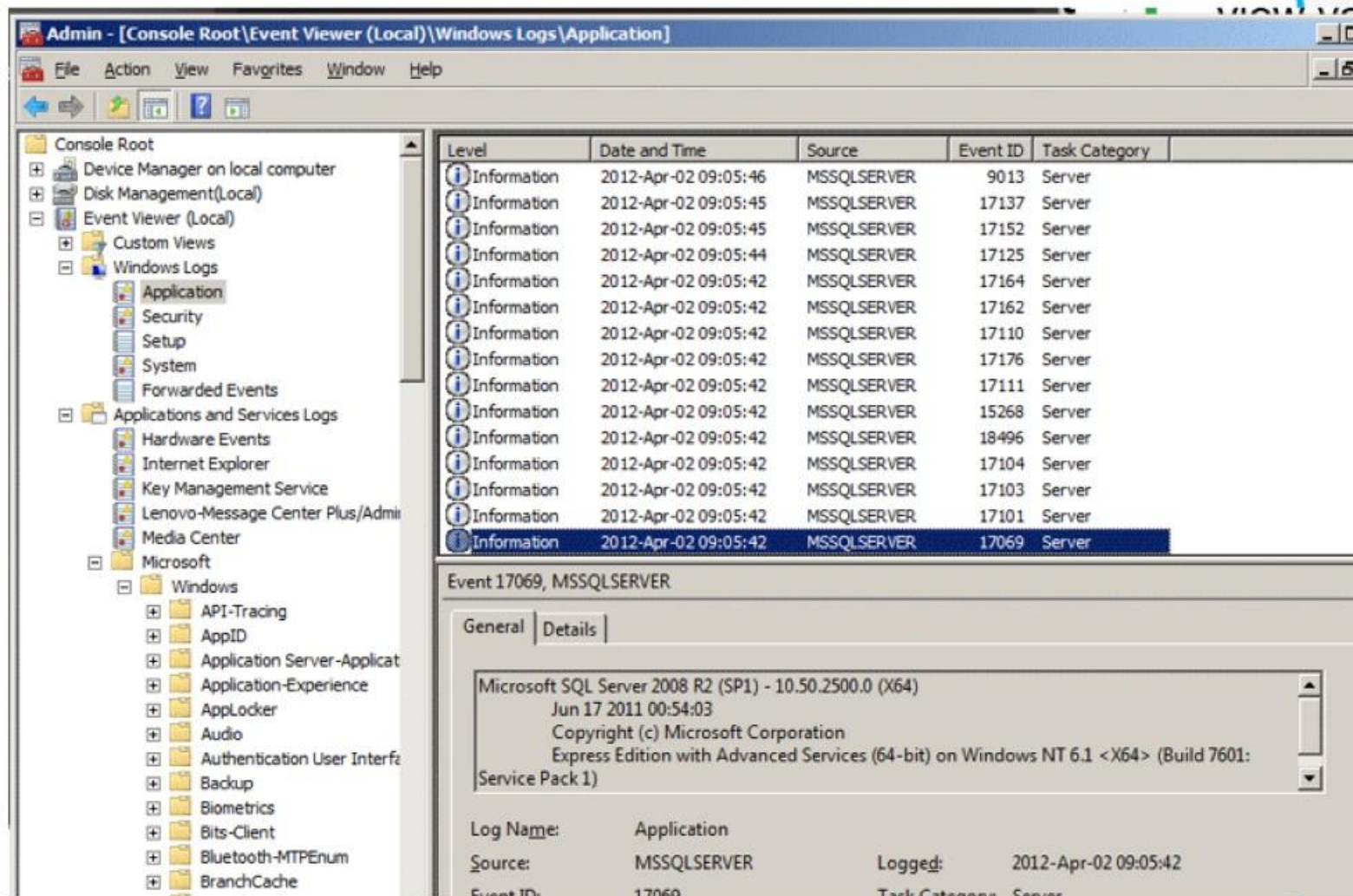
Along with the Volatile database data, Windows logs and Database plan cache, investigators can examine the following files to have an insight of the activities occurred on the database:

- ❑ **Trace files:** \\Microsoft SQL Server\MSSQL11.MSSOLSERVER\MSSQL\LOG\LOG_#.TRC
- ❑ **Database & logs files:** \\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\ DATA*.MDF | *.LDF
- ❑ **SQL Server error logs:** \\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\ LOG\ERRORLOG

Windows logs store the logon events performed on **SQL server** launch event viewer expand **windows Logs** node and view various **windows event logs**



event viewer



Admin - [Console Root\Event Viewer (Local)\Windows Logs\Application]

Level	Date and Time	Source	Event ID	Task Category
Information	2012-Apr-02 09:05:46	MSSQLSERVER	9013	Server
Information	2012-Apr-02 09:05:45	MSSQLSERVER	17137	Server
Information	2012-Apr-02 09:05:45	MSSQLSERVER	17152	Server
Information	2012-Apr-02 09:05:44	MSSQLSERVER	17125	Server
Information	2012-Apr-02 09:05:42	MSSQLSERVER	17164	Server
Information	2012-Apr-02 09:05:42	MSSQLSERVER	17162	Server
Information	2012-Apr-02 09:05:42	MSSQLSERVER	17110	Server
Information	2012-Apr-02 09:05:42	MSSQLSERVER	17176	Server
Information	2012-Apr-02 09:05:42	MSSQLSERVER	17111	Server
Information	2012-Apr-02 09:05:42	MSSQLSERVER	15268	Server
Information	2012-Apr-02 09:05:42	MSSQLSERVER	18496	Server
Information	2012-Apr-02 09:05:42	MSSQLSERVER	17104	Server
Information	2012-Apr-02 09:05:42	MSSQLSERVER	17103	Server
Information	2012-Apr-02 09:05:42	MSSQLSERVER	17101	Server
Information	2012-Apr-02 09:05:42	MSSQLSERVER	17069	Server

Event 17069, MSSQLSERVER

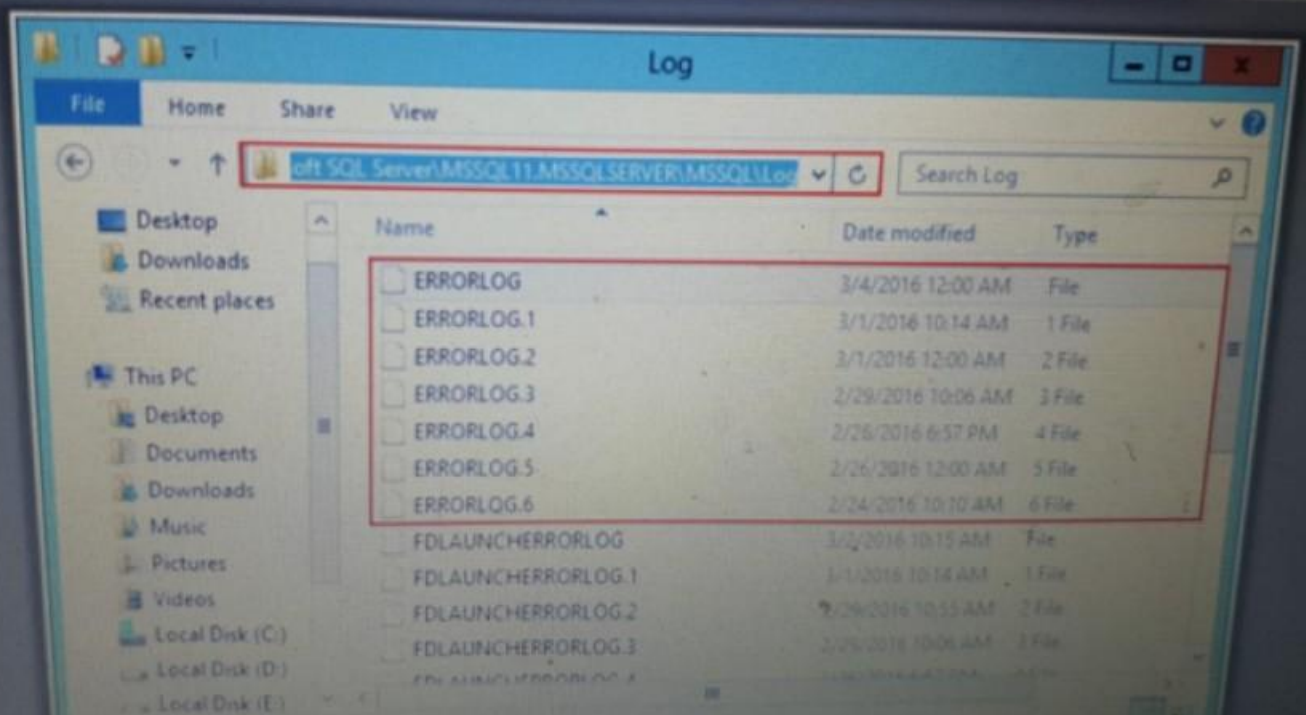
General Details

Microsoft SQL Server 2008 R2 (SP1) - 10.50.2500.0 (X64)
 Jun 17 2011 00:54:03
 Copyright (c) Microsoft Corporation
 Express Edition with Advanced Services (64-bit) on Windows NT 6.1 <X64> (Build 7601: Service Pack 1)

Log Name: Application
 Source: MSSQLSERVER
 Event ID: 17069
 Logged: 2012-Apr-02 09:05:42
 Task Category: Server

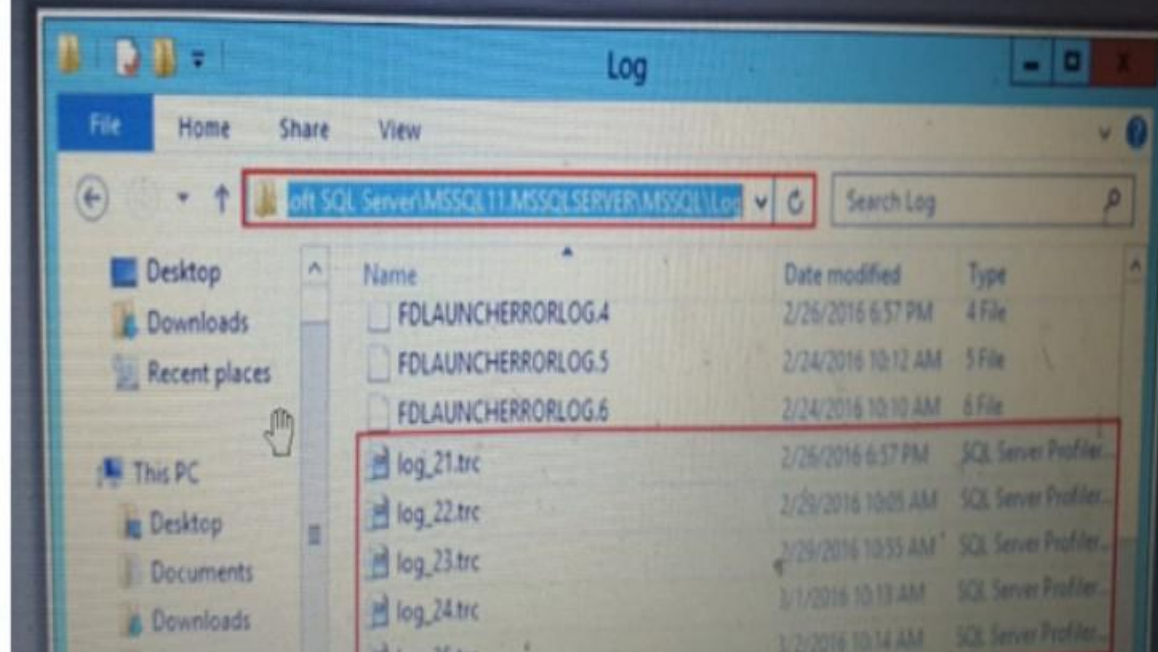
To collect the SQL Server error logs navigate to **C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\LOG**

The SQL Server error logs contain user defined events and specific system events



To collect the trace files (.trc) navigate to **C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\LOG**

The trace files contain the events occurred on a SQL server and the host databases



SQL Server :

- Volatile database
- primary data file and active transaction log
- Database plan cache

- Volatile Database is a RAM-style memory, which usually loses all its contents on power cuts.
- Investigators can track the volatile database information like **login sessions of an account and the transactions** using **ApexSQL** DBA's **ApexSQL** Audit application.

Common reports Custom reports Options

General

- Complete audit trail
- Backup and DBCC activities

Change and activity auditing

- DML history
- DDL history
- Access history

Security

- Security configuration history**
- Logon activity history
- Permission changes
- Unauthorized access

Auditing overview

- Audit settings history
- Audit integrity checks

Date range: 06/07/2015 00:00 to 06/07/2015 23:59

Server: [Dropdown]
Database: [Dropdown]
Schema: [Dropdown]

Object: [Dropdown]
Login: [Dropdown]
Client host: [Dropdown]
Application: [Dropdown]

☒ Security

- ☒ Assembly load
- ☒ Audit login
- ☒ Audit login failed
- ☒ Audit logout
- ☒ Audit server alter trace event
- ☒ Audit server starts and stops

Preview Generate PDF

Hide filter

Date	Server	Database	Login	Application	Client host	Schema	Object	Operation
06/07/2015 06:40:51.983 AM	SERVER2012R2	master	NT AUTHORITY\ANONYMOUS LOGON	Report Server	SERVER2012R2-L1			Audit login failed
Login failed for user 'NT AUTHORITY\ANONYMOUS LOGON'. Reason: Could not find a login matching the name provided. [CLIENT: 192.168.1.7]								
06/07/2015 06:40:14.223 AM	SERVER2012R2	master	SERVER2012R2\Administrator	Microsoft SQL Server Management Studio - Query	SERVER2012R2			Revoke server permissions
REVOKE ALTER ANY CONNECTION TO [1] AS [sa]								
06/07/2015 06:40:14.223 AM	SERVER2012R2	master	SERVER2012R2\Administrator	Microsoft SQL Server Management Studio - Query	SERVER2012R2		sa	Server principal impersonation
REVOKE ALTER ANY CONNECTION TO [1] AS [sa]								
06/07/2015 06:40:14.223 AM	SERVER2012R2	master	SERVER2012R2\Administrator	Microsoft SQL Server Management Studio - Query	SERVER2012R2		sa	Server principal impersonation
REVOKE ALTER ANY CONNECTION TO [1] AS [sa]								
06/07/2015 06:40:14.223 AM	SERVER2012R2	master	SERVER2012R2\Administrator	Microsoft SQL Server Management Studio - Query	SERVER2012R2			Revoke server permissions
REVOKE ALTER ANY AVAILABILITY GROUP TO [1] CASCADE AS [sa]								
06/07/2015 06:40:14.220 AM	SERVER2012R2	master	SERVER2012R2\Administrator	Microsoft SQL Server Management Studio - Query	SERVER2012R2		sa	Server principal impersonation
REVOKE ALTER ANY AVAILABILITY GROUP TO [1] CASCADE AS [sa]								
06/07/2015 06:40:14.220 AM	SERVER2012R2	master	SERVER2012R2\Administrator	Microsoft SQL Server Management Studio - Query	SERVER2012R2		sa	Server principal impersonation
REVOKE ALTER ANY AVAILABILITY GROUP TO [1] CASCADE AS [sa]								
06/07/2015 06:40:14.220 AM	SERVER2012R2	master	SERVER2012R2\Administrator	Microsoft SQL Server Management Studio - Query	SERVER2012R2			Revoke server permissions
REVOKE ADMINISTER BULK OPERATIONS TO [1] AS [sa]								

Collecting Primary Data File and Active Transaction Logs

- The primary data file (mdf) and active transaction logs (ldf) play a key role in the forensic investigation.
- These files offer sufficient information to a forensic examiner for dealing with the investigation.
- A forensic examiner needs to know the location of mdf and ldf associated with a database, before proceeding with the investigation.
- The SQLCMD application helps an investigator to obtain the location of these files.

Collecting Primary Data File and Active Transaction Logs

Collecting the active transaction log

Determine on disk locations of the transaction log files

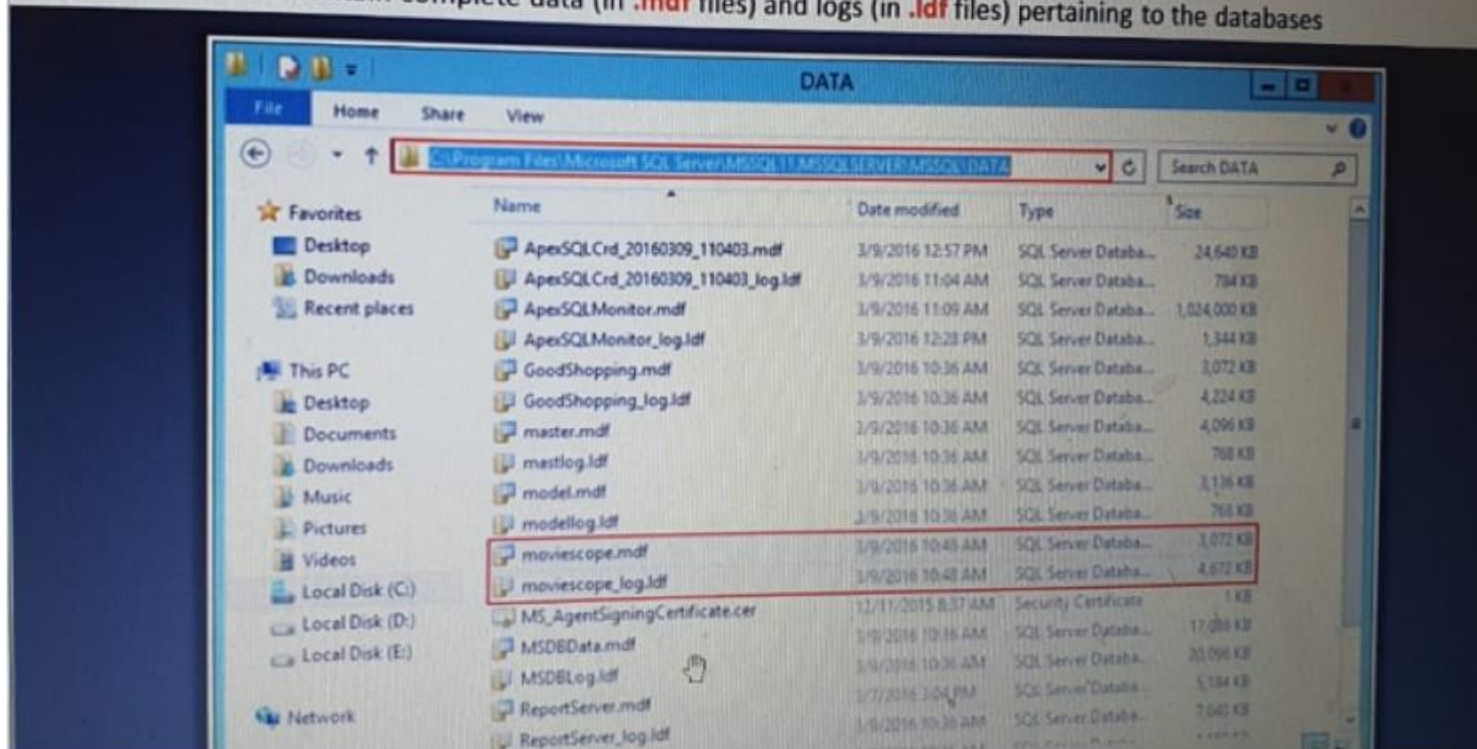
The **SQLCMD** application lets investigators load and establish a connection with the server.

```
D:\FResponse>sqlcmd -S RZ-SQL-2005 -e -s"," -E
1> sp_helpdb OnlineSales
2> go
```

name	fileid	filename	...
OnlineSales	1	C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\OnlineSales.mdf	...
OnlineSales_log	2	C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\OnlineSales_log.ldf	...
OnlineSales_log2	3	C:\OtherLogs\OnlineSales_log2.ldf	...

Collecting Primary Data File and Active Transaction Logs

- Collect the database files (.mdf) and log files (.ldf) from **C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA**
- These files contain complete data (in .mdf files) and logs (in .ldf files) pertaining to the databases



Collecting Primary Data File and Active Transaction Logs

- SQL Server Management Studio (SSMS) is an integrated environment for accessing, configuring, managing, administering, and developing all components of SQL Server and Azure SQL Database.
- Forensic investigators need to have good knowledge of how to use various functions (such as **dbcc log, fn_dblog, etc.**) in the SSMS to view and analyze the logs in plain text format.
- **Fn_dblog** : function allow to retrieve the action portion of the transaction log file.
- **DBCC Log**: command allow to retrieve the action transaction log files for specified database.

Chapter 07

Collecting Primary Data File and Active

The screenshot displays the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'ALEXB-PC\SQLEXPRESS2008 (SQL Server 10.50.1600)'. The central query window shows a query executed against the 'sa (51)' user. The query results are displayed in the Results pane, showing two rows of data from the fn_dblog function. The Properties pane on the right shows the connection details for the current session.

Query Executed:

```
Select * from fn_dblog('000000CB:00000055:0002', '000000CC:00000088:000A')  
Select * from fn_dblog(NULL, NULL)
```

Query Results:

	Current LSN	Operation	Context	Transaction ID	LogBlockGeneration	Tag Bits	Log Record Fixed Length	Log Record Length
1	000000ce:000000ac:0001	LOP_BEGIN_CKPT	LCX_NULL	0000:00000000	0	0x0000	96	96
2	000000ce:000000ad:0001	LOP_END_CKPT	LCX_NULL	0000:00000000	0	0x0000	136	136

Connection Properties:

- Current connection parameters
- Aggregate Status
 - Connection failures
 - Elapsed time: 00:00:00.1410081
 - Finish time: 27/12/2013 11:09:30
 - Name: ALEXB-PC\SQLEXPRESS2008
 - Rows returned: 2
 - Start time: 27/12/2013 11:09:30
 - State: Open
- Connection
 - Connection name: ALEXB-PC\SQLEXPRESS2008
- Connection Details
 - Connection elapsed: 00:00:00.1410081
 - Connection finish time: 27/12/2013 11:09:30
 - Connection rows returned: 2
 - Connection start time: 27/12/2013 11:09:30
 - Connection state: Open
 - Display name: ALEXB-PC\SQLEXPRESS2008
 - Login name: sa
 - Server name: ALEXB-PC\SQLEXPRESS2008
 - Server version: 10.50.1600
 - Session Tracking ID
 - SPID: 51

Status Bar: Query executed successfully. ALEXB-PC\SQLEXPRESS2008 (10... sa (51) lwtsFC 00:00:00 2 rows

Collecting Database plan cache

- Whenever a query is run for the first time in SQL Server, it is compiled, and a query plan is generated for the query.
- Every query requires a query plan before it is executed.
- This query plan is stored in SQL Server **query plan cache**.
- This way when that query is run again, SQL Server doesn't need to create another query plan; rather it uses the cached query plan which improved database performance.

SQL Server provides the following **dynamic management views and functions** that can be used to find out what is in the plan cache at any given time.

- sys.dm_exec_cached_plans
- sys.dm_exec_sql_text
- sys.dm_exec_query_plan

`sys.dm_exec_cached_plans`

- Returns a row for each query plan that is cached by SQL Server for faster query execution.
- You can use this dynamic management view to find cached query plans, cached query text, the amount of memory taken by cached plans, and the reuse count of the cached plans.

Ex: -- How many plans are in the cache

```
SELECT COUNT(*) FROM sys.dm_exec_cached_plans;
```

Collecting Database plan cache

```
SELECT cplan.usecounts, cplan.objtype, qtext.text, qplan.query_plan
FROM sys.dm_exec_cached_plans AS cplan
CROSS APPLY sys.dm_exec_sql_text(plan_handle) AS qtext
CROSS APPLY sys.dm_exec_query_plan(plan_handle) AS qplan
ORDER BY cplan.usecounts DESC
```

usecounts	cacheobjtype	objtype	text	query_plan
3	Compiled Plan	Adhoc	DECLARE @edition sysname; SET @edition = cast(SERV...	<ShowPlanXML xmlns="http://schemas.microsoft.com...
2	Compiled Plan	Prepared	(@_msparam_0 nvarchar(4000))SELECT dtb.collation_nam...	<ShowPlanXML xmlns="http://schemas.microsoft.com...
1	Compiled Plan	Adhoc	declare @MasterPath nvarchar(512) declare @...	<ShowPlanXML xmlns="http://schemas.microsoft.com...
1	Compiled Plan	Adhoc	SELECT dtb.name AS [Name], dtb.state AS [State] FROM ...	<ShowPlanXML xmlns="http://schemas.microsoft.com...
1	Compiled Plan	Adhoc	SELECT cp.usecounts, cp.cacheobjtype, cp.objtype, st.te...	<ShowPlanXML xmlns="http://schemas.microsoft.com...
1	Compiled Plan	Adhoc	SELECT log.name AS [Name], log.principal_id AS [ID], ISN...	<ShowPlanXML xmlns="http://schemas.microsoft.com...
1	Compiled Plan	Adhoc	SELECT dtb.name AS [Name], dtb.database_id AS [ID], CA...	<ShowPlanXML xmlns="http://schemas.microsoft.com...
1	Compiled Plan	Prepared	(@dbname nvarchar(6))SELECT COLLATIONPROPERTY((...	<ShowPlanXML xmlns="http://schemas.microsoft.com...
1	Compiled Plan	Adhoc	SELECT dtb.name AS [Name], dtb.database_id AS [ID], CA...	<ShowPlanXML xmlns="http://schemas.microsoft.com...

Mysql

- MySQL is the world's most popular **open source** database.
- MySQL powers many of the most accessed applications, including Facebook, Twitter, Netflix, Uber, Airbnb, Shopify, and Booking.com.

- **Database forensics** is the process of **examining, analyzing, and recovering evidence** from a database system without altering its data.
- In **MySQL**, investigates the server, logs, tables, and storage files to understand what happened—whether a breach, data modification, or unauthorized access.

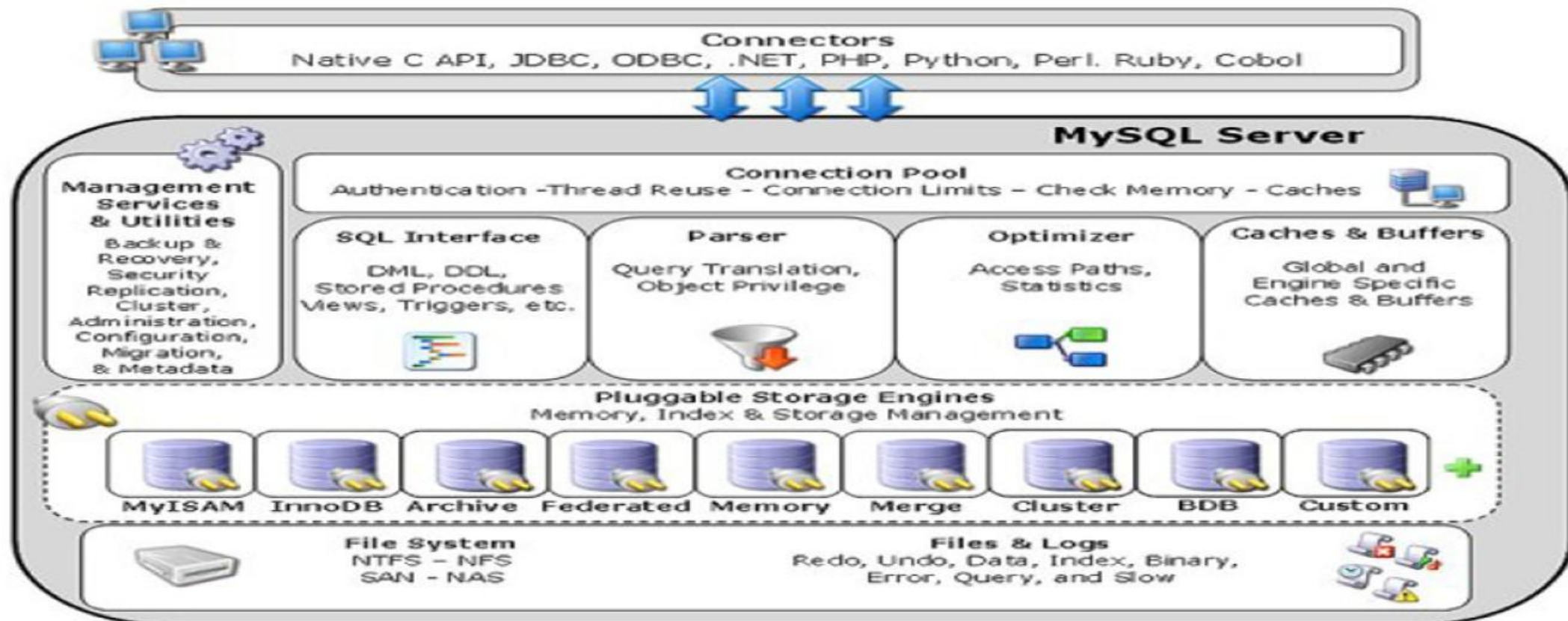
MySQL generates several logs that are critical in forensics:

- **Error Log:** Shows startup errors, crashes, failed logins, etc.
- **Binary Log:** Records all write operations (INSERT, UPDATE, DELETE).
- **Slow Query Log:** Logs queries taking unusually long.
May show unexpected or malicious operations.
- **General Query Log:** Logs all SQL queries executed.

Investigators check:

- Accounts created
- Privilege changes
- Failed login attempts
- Suspicious privilege escalations

The architecture of MySQL is based on a tiered architecture, which is the combination of subsystems and support components interacting with one another to read, analyze and execute the queries made to the database server, and return the results





Storage engines are used to create, read, and update data within a database. Various Storage engines supported by MySQL are:

1. InnoDB (default storage engine in MySQL version $\geq 5.5.5$)

Is often used when you need to use transactions

2. MyISAM (default storage engine in MySQL version $< 5.5.5$)

Provides unlimited data storage

3. Memory

It is an in-memory table that implements a hashing mechanism for faster retrieval of commonly used data



4. BDB

Considered an alternative to InnoDB

Supports additional transaction methods such as COMMIT and ROLLBACK\

5. ARCHIVE

Well known for storing and retrieving huge volumes of seldom-accessed archival or historical data

6. BLACKHOLE

7. MERGE

8. FEDERATED

Status and log files stored in data directory include:

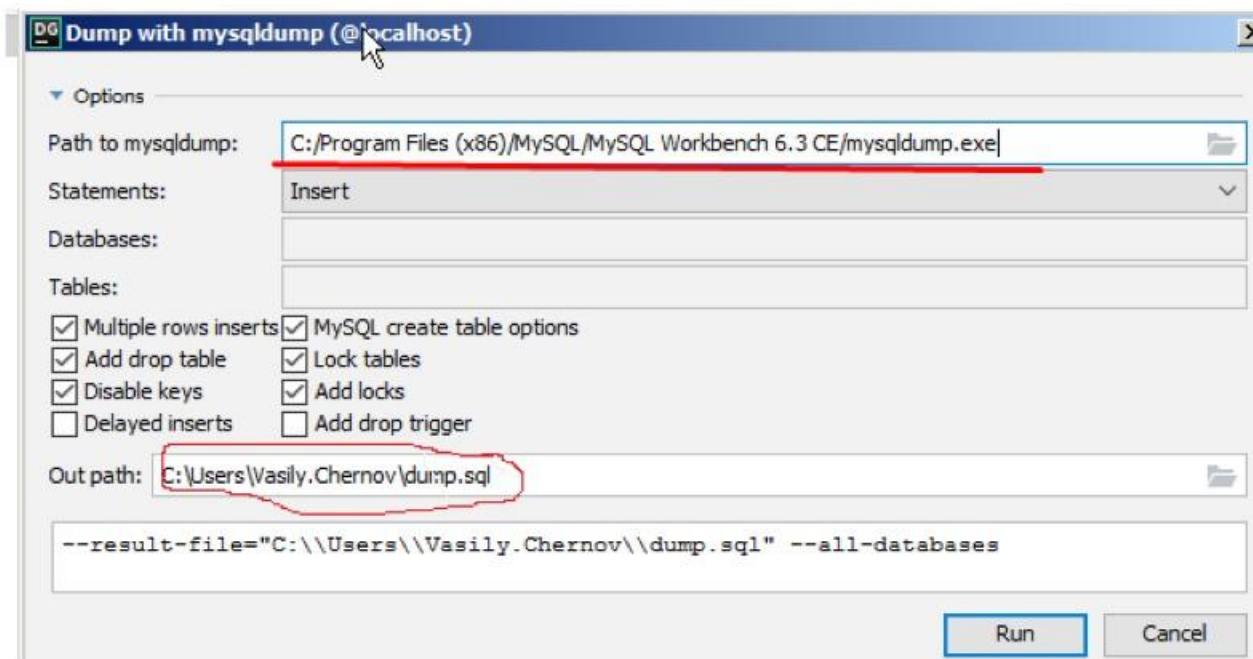
- **Process ID file (HOSTNAME.pid)**, contains the process ID created when the server starts
- **Error log (HOSTNAME.err)**, contains the information associated with the startup and shutdown events, and errors
- **General query log (HOSTNAME.log)**, logs the client connections and activities
- **Binary log index (HOSTNAME-bin.index)**, contains the list of all the binary log files currently available in the data directory

- **Relay log** (HOSTNAMErelay-bin.n), contains the events that describe the changes occurred in the database
- **Relay log index** (HOSTNAMErelay-bin.index), contains the list of all the relay log files currently available in the data directory
- **Master info file** (master.info) created by a replication slave server, that contains the essential parameters used for connecting to the master slave
- **Relay log info file** (relay-log.info) created by a replication slave server, that contains the status of relay log processing
- **Slow query log** (HOSTNAMEslow.log), a text file that contains statements which take longer processing time

The following utility programs help investigators to perform the forensic examination.

1. Mysqldump

- The utility allows you to dump a database or a collection of databases for backup purposes
- It generates a **.sql** file with CREATE table, DROP table and INSERT into the SQL statement of the source database
- It executes the **.sql** file on the destination database to restore the original database.



The following utility programs help investigators to perform the forensic examination.

2. **mysqlaccess**

- Checks the access privileges defined for host name, user name, etc.
- Validates access using the user, db, and host tables

3. **myisamlog**

Processes the contents of MyISAM log file and perform recovery operation, display version information, , depending on the situation

4. **myisamchk**

- Views the status of the MyISAM table or checks, repairs, or optimizes them.

- <https://info-savvy.com/>



Thank You