Practical Lab Exercise: Monitoring, Logging & Auditing in MS SQL Server

Block 1: Create a test database and table

```
USE master;
GO

IF EXISTS (SELECT name FROM sys.databases WHERE name = 'DemoAudit')
    DROP DATABASE DemoAudit;
GO

CREATE DATABASE DemoAudit;
GO

USE DemoAudit;
GO

-- Original Users table
CREATE TABLE Users (
    UserID INT PRIMARY KEY,
    Username VARCHAR(50),
    PasswordHash VARCHAR(100)
);
GO
```

```
INSERT INTO Users VALUES (1, 'admin', 'hash123'), (2, 'student', 'hash456');
GO

-- NEW: Activity log table (replaces DMV monitoring)
CREATE TABLE ActivityLog (
    LogID INT IDENTITY(1,1) PRIMARY KEY,
    EventTime DATETIME DEFAULT GETDATE(),
    EventType VARCHAR(50),
    UserName VARCHAR(100),
    Details VARCHAR(500)
);
GO

-- NEW: Error log table (replaces sp_readerrorlog)
CREATE TABLE ErrorLog (
    ErrorID INT IDENTITY PRIMARY KEY,
    ErrorTime DATETIME DEFAULT GETDATE(),
    ErrorNumber INT,
    ErrorMessage VARCHAR(500)
);
GO
```

Block 2: Stored Procedure that Logs Reads (Auditing + Monitoring)

```
CREATE PROCEDURE sp_GetUser
    @UserID INT
AS
BEGIN
    -- Log the access (this is your AUDIT trail)
    INSERT INTO ActivityLog (EventType, UserName, Details)
    VALUES ('SELECT', SUSER_NAME(), 'UserID = ' + CAST(@UserID AS VARCHAR));

    -- Return the data
    SELECT * FROM Users WHERE UserID = @UserID;
END
GO
```

Every time someone calls `sp_GetUser`, a record is written to `ActivityLog`. This replaces the need for server-level auditing or default trace.

Block 3: Test the Audited Procedure

```
-- Run the procedure a few times
EXEC sp_GetUser 1;
EXEC sp_GetUser 2;
EXEC sp_GetUser 1;
GO

-- View the activity log (this is your MONITORING output)
SELECT * FROM ActivityLog;
GO
```

Block 4: Simulate Error Logging (Replaces `sp_readerrorlog`)

```
-- Create a procedure that logs errors into our custom ErrorLog table
CREATE PROCEDURE sp_FaultyProcedure
AS
BEGIN
    BEGIN TRY
        -- Cause an error (divide by zero)
        SELECT 1/0;
    END TRY
    BEGIN CATCH
        INSERT INTO ErrorLog (ErrorNumber, ErrorMessage)
        VALUES (ERROR_NUMBER(), ERROR_MESSAGE());
    END CATCH
END
GO

-- Run it to generate an error log entry
EXEC sp_FaultyProcedure;
GO

-- View the error log (this replaces reading the SQL Server error log)
SELECT * FROM ErrorLog;
GO
```

Block 5: Real-time Monitoring (Activity Log)

```
-- See recent activity – who accessed what and when
SELECT
    LogID,
    EventTime,
    EventType,
    UserName,
    Details
FROM ActivityLog
ORDER BY EventTime DESC;
GO
```

Block 7: Clean up the demo database

```
USE master;
GO
DROP DATABASE DemoAudit;
GO
```