ScriptPreflight

Preflight report

Synthetic sample. Not a customer script.

Script
purge-audit-log.sql

Report ID
RPT-00000000

Date
24 November 2026, 10:27 SAST (08:27 UTC)

Version
Engine 0.1.0+scriptdom.180.140.0 · ruleset 0.16.6

Result

Critical risk · Do not run as written

Deletes every row of dbo.AuditLog, on line 9.

What it changes

Deletes every row of dbo.AuditLog.

`DELETE · line 9`

How far it reaches

Every row of dbo.AuditLog.

`Every row of dbo.AuditLog`

Can you undo it

Partly. Some changes can be undone and some can't.

`reversibility: partial`

What you meant, and what it can reach

You said **no count given**

It can reach **every row of dbo.AuditLog**

What we could check: all of it

Every part of the script was read by the rules.

Also note

- Synthetic sample, not a real customer script. Every finding, the blast radius, coverage and reversibility are real engine output; no AI explanation, rewrite or plans were added.
- T-SQL pack: SQL Server and Azure SQL syntax is parsed with Microsoft ScriptDom; no server is contacted, so row counts, object existence and permissions are not known.
- Row reach is judged from the WHERE clause text only: a condition on a column that matches most rows is reported as filtered.
- Procedures, functions and triggers not defined in the script are not analysed; sqlcmd :r includes are not analysed, and :!! shell commands only when they run powershell -Command.
- Oracle, PostgreSQL and MySQL scripts are refused, not analysed.

Findings (5)

Most severe first: 2 critical, 1 high, 1 medium, 1 low.

Critical Deletes every row of dbo.AuditLog

SPF-TSQL-003 · version 0.7.0 · line 9

```
  9 DELETE FROM dbo.AuditLog;
```

Why it matters

DELETE on dbo.AuditLog reaches every row: no WHERE condition. Once committed, the rows can only be recovered from a backup.

Recommendation

Add a WHERE condition that names exactly the rows to remove, check the count with a SELECT first, and run it in a transaction you can roll back.

Critical Empties table dbo.AuditLogStaging

SPF-TSQL-002 · version 0.7.0 · line 12

```
 12 TRUNCATE TABLE dbo.AuditLogStaging;
```

Why it matters

TRUNCATE TABLE removes every row of dbo.AuditLogStaging at once, with no WHERE condition and no row-by-row log. Outside a transaction that is rolled back, the rows can only be recovered from a backup.

Recommendation

Confirm the table is the one you mean (not a production table with a similar name) and copy the rows first.

High Switches SalesReporting to the SIMPLE recovery model

SPF-TSQL-014 · version 0.7.0 · line 17

```
 17 ALTER DATABASE SalesReporting SET RECOVERY SIMPLE;
```

Why it matters

The log chain is broken: point-in-time restore is impossible until a new full backup is taken, and log backups stop working. Nothing later in the script switches back.

Recommendation

Keep FULL recovery for databases that need point-in-time restore; take a full backup straight after switching back.

Medium Transaction has no error handling

SPF-TSQL-022 · version 0.7.0 · line 7

```
  7 BEGIN TRANSACTION;
```

Why it matters

Without TRY...CATCH or SET XACT_ABORT ON, most errors do not roll the transaction back: later statements still run and the earlier changes can be committed or left open. The transaction makes 2 changes, so a failure in one can commit the others on their own.

Recommendation

Add SET XACT_ABORT ON, and wrap the transaction in BEGIN TRY ... COMMIT ... END TRY BEGIN CATCH ROLLBACK; THROW; END CATCH.

Low DBCC SHRINKFILE shrinks database files

SPF-TSQL-015 · version 0.7.0 · line 18

```
 18 DBCC SHRINKFILE (SalesReporting_log, 1);
```

Why it matters

Shrinking moves pages and fragments indexes; the file often grows again.

Suggested fix

No suggested fix in this report.

Sign-off

Reviewed by

Approved by

Change ref

Date