Guardian One SQL Server Best Practices

Guardian One documentation | Version 1.1 | Reviewed 14 August 2026

Purpose and Decision Rule

Current version	1.1
Last reviewed	14 August 2026
Document owner	Product owner: Adam Timothy
Change summary	Manager-readability and controlled-publication review

Version history

Version	Date	Change	Status
1.0	Initial issue	Initial controlled document	Superseded
1.1	14 August 2026	Manager-readability and publication review	Current

This standard provides a review baseline for SQL Server 2012 and later. A best practice is not a universal configuration value. Apply a recommendation only after establishing workload evidence, platform and edition support, recovery objectives, security requirements, an accountable owner, test results, change approval, and a rollback plan.

Guardian One performs scheduled, point-in-time assessment and assurance. It does not provide continuous monitoring or alerting. Customers must retain an owned monitoring, alerting, incident-response, backup, and disaster-recovery capability.

Server-Level Standards

Control	Baseline	Guardian One evidence	Required decision
Memory	Set max server memory so Windows, drivers, security, backup, and other local services retain measured headroom.	Configured and committed SQL memory, Windows memory, paging, and grant evidence where available.	Size from representative demand; do not apply a fixed percentage blindly.
Parallelism	Review MAXDOP and cost threshold together using CPU, waits, Query Store, topology, and workload tests.	Configuration, CPU, waits, and governed Query Store candidates.	Retain only a measured benefit; document scope and rollback.
TempDB	Use appropriately sized data files with consistent size and fixed growth; separate storage when risk and workload justify it.	File count, allocation, I/O latency, growth settings, placement, and trace growth events.	Add files only for evidenced contention; do not assume one file per processor is always correct.
Trace flags	Maintain an approved inventory tied to the exact SQL Server version and support position.	Configuration and supplementary diagnostic evidence.	Remove obsolete or undocumented flags only after dependency testing.
Security	Prefer named Windows identities, least privilege, managed service accounts, controlled sysadmin membership, secure protocols, and auditable emergency access.	Principals, roles, configuration, compliance controls, and default-trace security changes.	Validate application identities before changing access.
Patching	Maintain supported builds and a tested patch cadence covering engine, OS, drivers, HA, backup, and security dependencies.	Version/build and assessment findings.	Test representative workloads and recovery; record rollback and post-change validation.

Database-Level Standards

Control	Baseline	Guardian One evidence	Required decision
Recovery model	Align FULL, BULK_LOGGED, or SIMPLE with approved recovery-point and point-in-time restore requirements.	Recovery model, backup history, log evidence, and database state.	Changing recovery model changes the backup strategy; validate the chain afterwards.
File layout and growth	Pre-size for expected demand. Prefer workload-tested fixed-MB growth over percentages. Avoid routine shrink.	Current/max size, growth mode, volume capacity, file placement, and growth/shrink trace events.	Size data and log independently; unlimited growth does not reserve disk capacity.
Naming	Use stable, meaningful names that follow the organisation's conventions and avoid embedding transient server or employee identities.	Inventory and application assessment context.	Treat renames as dependency changes, not cosmetic edits.
Schemas	Use schemas as ownership and permission boundaries; avoid placing all application objects in dbo by default.	Schema/object inventory and security evidence where collected.	Validate ownership chains and deployment tooling.
Compression and partitioning	Use only when edition/version support, CPU cost, I/O benefit, maintenance, and data lifecycle justify it.	Size, workload, index, and Query Store evidence.	Test representative reads, writes, rebuilds, backups, and restores.
Database options	Disable AUTO_SHRINK and normally AUTO_CLOSE; use CHECKSUM page verification. Treat every exception as an application dependency.	Native application-database checks.	Document rationale, owner, evidence, and rollback.

Object-Level Standards

Control	Baseline	Guardian One evidence	Required decision
Indexing	Design for the actual workload; remove redundancy carefully and retain indexes required for constraints or critical access paths.	Duplicate/overlapping, disabled, hypothetical, missing-index, usage, and query evidence where available.	Validate writes, reads, storage, maintenance, and plan regressions before change.
Constraints and triggers	Prefer declarative keys and constraints for data integrity. Use triggers only for a documented requirement with set-based, multi-row-safe logic.	Constraint state and application findings.	Never enable or trust a constraint without validating existing data.
Stored procedures	Use schema-qualified, parameterised, set-based code; control transaction scope, error handling, permissions, and result contracts.	Query and schema evidence where entitled and available.	Test parameter sensitivity, concurrency, failure paths, and rollback.
Anti-patterns	Review RBAR processing, scalar UDFs, implicit conversions, non-SARGable predicates, broad selects, and unbounded transactions.	Query Store candidates, plan-related findings, waits, and application assessment.	Refactor only with a representative performance and correctness test.

Operational Standards

Control	Baseline	Guardian One evidence	Required decision
Backup and restore	Define RPO/RTO, protect backup destinations, use checksums where supported, retain appropriate full/differential/log chains, and test restores.	Bounded backup history, job/log findings, and recovery configuration.	A successful backup is not proof of recoverability; retain restore evidence.
Monitoring and alerting	Operate continuous coverage for availability, capacity, backups, corruption, Agent failures, HA health, security, and performance.	Guardian One supplies scheduled assessment evidence only.	Integrate findings with the customer's existing monitoring and incident process.
Performance baseline	Compare representative busy and quiet periods, business cycles, deployments, and infrastructure changes.	Repeatable assessments, waits, I/O, activity, Query Store, and repository history.	Do not diagnose a transient incident from one snapshot.
Deployment and source control	Version database code and configuration; use peer review, pre-production validation, controlled credentials, rollback, and post-deployment checks.	Default-trace change insights and assessment differences.	The cyclical default trace is supporting evidence, not a durable deployment audit.
Disaster recovery	Test restore, failover, dependency recovery, access, runbooks, communications, and return-to-service at an approved cadence.	Backup, availability, configuration, compliance, and repository evidence.	Record the achieved RPO/RTO and remediate gaps through owned plans.
Capacity	Forecast data, log, TempDB, backup, and repository growth with application owners.	File size/growth settings, volume capacity, trace events, and assessment history.	Set thresholds in customer monitoring and review forecasts after workload change.

Evidence and Escalation

For each exception, record the observed evidence, business impact, affected service, accountable owner, proposed action, dependencies, test plan, approval, maintenance window, rollback, and validation result. Escalate suspected corruption, errors 823/824/825, failed recovery, unexplained privilege elevation, exhausted storage, or repeated service interruption through the customer’s incident process.