

Database Performance Audit Checklist

Comprehensive audit of your database performance covering query optimisation, indexing, storage I/O, connection pooling, caching, and monitoring — 50+ items across 6 sections to identify bottlenecks and improve responsiveness.

6AUDIT
SECTIONS**50+**ITEMS TO
CHECK**Score**EACH SECTION
OUT OF 10**Free**PRINT & USE
NO STRINGS

How to Use This Checklist

Work through each section with your DBA or managed service provider. Run the recommended diagnostics before scoring each area. Flag any section scoring below 6 for immediate remediation and build a prioritised performance improvement plan.

Need Help With Your IT?

Our team can help you implement the recommendations in this resource.

info@cloudswitched.com
+44 2030 043 450

New London House, 8 London St, London EC3R 7LP

1 Query Performance & Optimisation

Poorly performing queries are the most common cause of database slowdowns. Identify and resolve the worst offenders first.

- Enable and review the **slow query log** to identify queries exceeding acceptable execution times (threshold: 1 second for OLTP workloads)
- The top 20 most **resource-intensive queries** have been identified using execution plans and ranked by cumulative CPU and I/O cost
- All frequently executed queries use **parameterised statements** rather than dynamic SQL to enable plan caching and prevent SQL injection
- Complex queries have been reviewed for **unnecessary joins, subqueries, and correlated subqueries** that could be refactored for efficiency
- Queries returning large result sets include **appropriate pagination** rather than fetching all rows and filtering in the application layer
- The **query execution plan cache** hit ratio is monitored and consistently above 90% to avoid repeated compilation overhead
- Application code has been reviewed to eliminate **N+1 query patterns** where loops issue individual queries instead of batch operations
- Stored procedures and views are **periodically reviewed** for performance regression as data volumes grow (schedule quarterly reviews)
- Database **statistics are updated regularly** to ensure the query optimiser has accurate cardinality estimates for plan selection

Section Score: /10

2 Indexing Strategy

Effective indexing dramatically improves read performance, but over-indexing degrades writes. Strike the right balance for your workload.

- All tables have a **primary key defined** and clustered index strategy aligned with the most common access patterns
- Missing index recommendations from the **database engine's advisor** have been reviewed and selectively implemented based on workload analysis
- Unused and **duplicate indexes have been identified** and removed to reduce storage overhead and write amplification
- Composite indexes are ordered with the **most selective column first** and cover the most frequently filtered and sorted columns
- Covering indexes are used for **high-frequency queries** to enable index-only scans and eliminate table lookups
- Index **fragmentation levels are monitored** and maintenance (rebuild or reorganise) is scheduled during off-peak windows (rebuild above 30% fragmentation)
- The ratio of **index reads to index writes** is tracked to ensure indexes are delivering value relative to their maintenance cost
- Filtered indexes or **partial indexes** are considered for tables with large volumes of rarely queried historical data
- The impact of new indexes is **tested in a staging environment** before deployment to production to verify improvement without side effects

Section Score: /10

3 Storage & I/O Performance

Database performance is ultimately constrained by storage throughput. Ensure your I/O subsystem is not the bottleneck.

- Database files are hosted on **SSD or NVMe storage** with sufficient IOPS for peak workload demands (spinning disks are inadequate for production databases)
- Data files and **transaction log files are on separate physical volumes** to prevent I/O contention between read/write and logging operations
- TempDB or temporary tablespace is placed on **dedicated fast storage** as it handles sort operations, hash joins, and spills from memory
- Storage **latency is continuously monitored** and consistently below 5ms for read operations and 2ms for write operations
- The database **autogrowth settings** use fixed-size increments rather than percentage-based growth to prevent file fragmentation
- Database file sizes are **pre-allocated with adequate headroom** to minimise autogrowth events during business hours
- Disk queue lengths are monitored and **sustained queue depths above 2 per disk** are investigated as potential I/O bottlenecks
- Backup operations are scheduled to **avoid peak I/O periods** and use compression to reduce storage throughput requirements

Section Score: /10

4 Connection Management & Pooling

Efficient connection management prevents resource exhaustion and ensures consistent response times under load.

- Application connection strings use **connection pooling** with minimum and maximum pool sizes tuned to the workload (avoid unlimited max pool size)
- The **maximum concurrent connections** configured on the database server reflects actual peak demand plus a reasonable headroom margin
- Connection **timeout settings** are configured to release abandoned connections promptly rather than holding resources indefinitely
- The application properly **closes and returns connections** to the pool after each operation — no connection leaks identified
- Connection pool **health checks** validate connections before reuse to prevent application errors from stale or broken connections
- Long-running queries are **identified and terminated** if they exceed acceptable thresholds to prevent connection starvation
- Database server **resource governor or workload management** is configured to prevent any single application from monopolising connections
- Connection pool metrics (**active, idle, waiting, and timed-out**) are monitored and alerting is configured for pool exhaustion

Section Score: /10

5 Caching & Memory Configuration

Properly configured memory and caching reduce I/O pressure and deliver consistent sub-millisecond response times for frequently accessed data.

- The database server has **sufficient RAM allocated** to hold the active working set in the buffer pool without excessive paging to disk
- Buffer pool or **cache hit ratio** is monitored and consistently above 95% for OLTP workloads (below 90% indicates insufficient memory)
- Memory is allocated with a **maximum server memory setting** that leaves adequate headroom for the operating system and other processes
- Application-level caching (Redis, Memcached) is deployed for **frequently read, rarely changed data** to offload repetitive queries from the database
- Query plan cache is **appropriately sized** to retain execution plans for frequently used queries without consuming excessive memory
- The database **page life expectancy** metric is monitored and consistently above 300 seconds to confirm adequate buffer pool sizing
- Memory grants for **large sort and hash operations** are monitored to prevent excessive memory grants that starve other queries
- Cache invalidation strategies are implemented to ensure **stale data is not served** from application caches after database updates

Section Score: /10

6 Monitoring & Alerting

Proactive monitoring detects performance degradation before users notice. Reactive troubleshooting is always more expensive.

- A dedicated **database monitoring solution** is deployed (e.g., Datadog, SolarWinds DPA, pganalyze, or Percona Monitoring) with real-time dashboards
- Alerts are configured for **CPU utilisation, memory pressure, disk I/O latency, and connection count** exceeding defined thresholds
- Slow query alerts trigger when **query execution times exceed baseline thresholds** or when new queries appear in the slow query log
- Deadlock detection is enabled with **automatic alerting and logging** to identify and resolve contention between concurrent transactions
- Database **backup success and failure** is monitored with immediate alerting on any backup failure or missed backup window
- Replication lag (if applicable) is **monitored continuously** with alerts when lag exceeds acceptable thresholds for your RPO requirements
- Scheduled **performance reports** are generated weekly comparing key metrics against baseline to identify gradual degradation trends
- All monitoring data is **retained for at least 90 days** to support capacity planning, trend analysis, and post-incident investigation
- An **annual performance review** compares year-over-year metrics and feeds into capacity planning and budget discussions

Section Score: /10

7 Audit Summary & Action Plan

#	AUDIT AREA	SCORE	PRIORITY
1	Query Performance & Optimisation	/ 10	H / M / L
2	Indexing Strategy	/ 10	H / M / L
3	Storage & I/O Performance	/ 10	H / M / L
4	Connection Management & Pooling	/ 10	H / M / L
5	Caching & Memory Configuration	/ 10	H / M / L
6	Monitoring & Alerting	/ 10	H / M / L
TOTAL SCORE		/ 60	

Score Interpretation

80–100: Excellent. Your IT setup is well-managed. Focus on continuous improvement and emerging threats.

60–79: Good foundation but gaps exist. Prioritise areas scoring below 6 and create an action plan.

Below 60: Significant gaps that put your business at risk. Consider an urgent review with an IT specialist.

Top 3 Priority Actions:

- 1 _____
- 2 _____
- 3 _____

Additional Notes

Audit completed by: _____ Date: _____ Next review due: _____

Need Help With Your IT?

Our team can help you implement the recommendations in this resource.

info@cloudswitched.com
+44 2030 043 450

New London House, 6 London St, London EC3R 7LP