

How to Repair SQL Database: A Practical Workflow



- "Repair" means restoring accessibility, consistency, and serviceability without avoidable data loss.



- Scope: SQL Server corrupt pages, log failures, failed restores, suspect states.



- Six-stage workflow: detect, assess, contain, recover, validate, harden.



- Backup-first rule: verify backups before any repair; never repair the only copy.



- Lab safety: disposable databases, snapshots, isolated instances only — no production risk.



Why Databases Corrupt: Causes and Failure Signatures



- Corruption usually starts outside SQL Server, deep in the I/O path



- Failing disks, RAID cache, faulty RAM, bad drivers, improper shutdowns



- Physical corruption damages pages; logical corruption breaks metadata



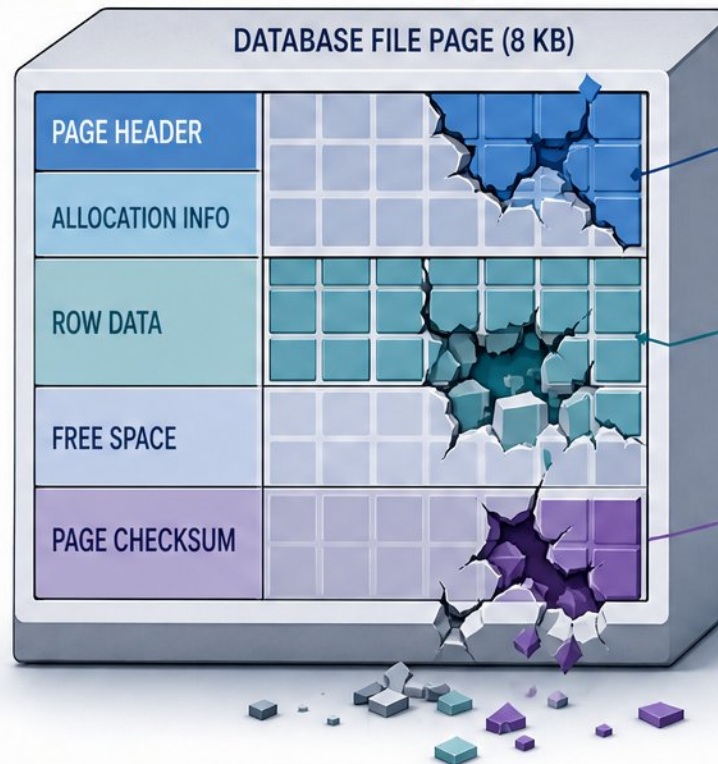
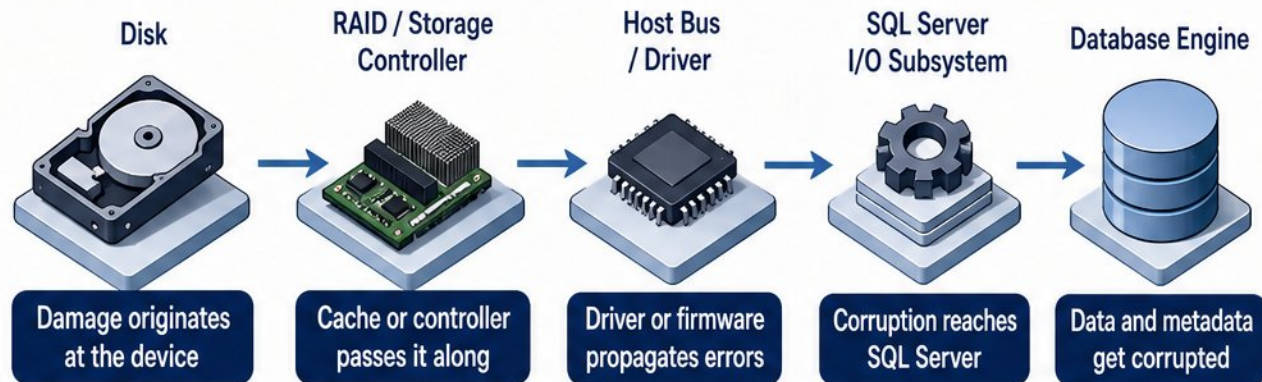
- Page checksums, torn-page detection, backup checksums reveal the damage



- Watch for Msg 823, 824, 825, 832, 8904, 8905 in the error log



- Silent corruption spreads into every backup taken afterward



TORN-PAGE DAMAGE

Partial writes leave pages inconsistent



ALLOCATION DAMAGE

Extent or allocation map corruption



CHECKSUM MISMATCH

Stored checksum does not match the page



Reading the Error Messages: 823, 824, 825, and Friends



Msg 823

- Msg 823: OS reported I/O failure — storage admitted it could not complete the request



Msg 824

- Msg 824: I/O succeeded but page failed logical check — wrong data returned as healthy



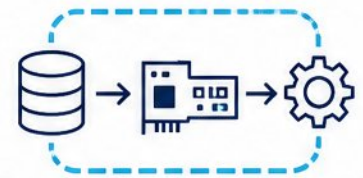
Msg 825

- Msg 825: read failed then succeeded on retry — early hardware warning, not noise



Msg 605

- Msg 605 and 829: allocation-level damage — a restore conversation, not a repair one



Msg 829

- All point at the storage path — disk, controller, driver — investigate before any DB fix



Focus on the storage path first.
Hardware issues surface as these messages.





Core Concepts and Toolbox



- DBCC CHECKDB options: PHYSICAL_ONLY, DATA_PURITY, ESTIMATEONLY, NO_INFOMSGS, ALL_ERRORMSGs



- Repair levels: REPAIR_FAST does nothing; REPAIR_REBUILD safe; REPAIR_ALLOW_DATA_LOSS last resort



- Emergency mode: READ_ONLY, sysadmin-only; single-user mode required for REPAIR_* options



- Tail-log backup preserves transactions after the last log backup; page restore is lower impact



- RESTORE DATABASE vs RESTORE PAGE vs piecemeal/filegroup restore



- Escalate to Microsoft or a recovery specialist when repair is unsafe; preserve all evidence first

Detection: Running and Scheduling Integrity Checks



- Run DBCC CHECKDB on a schedule, not just after symptoms appear



- Keep CHECKDB interval shorter than your backup retention window



- PHYSICAL_ONLY for frequent checks; full logical check on longer cadence



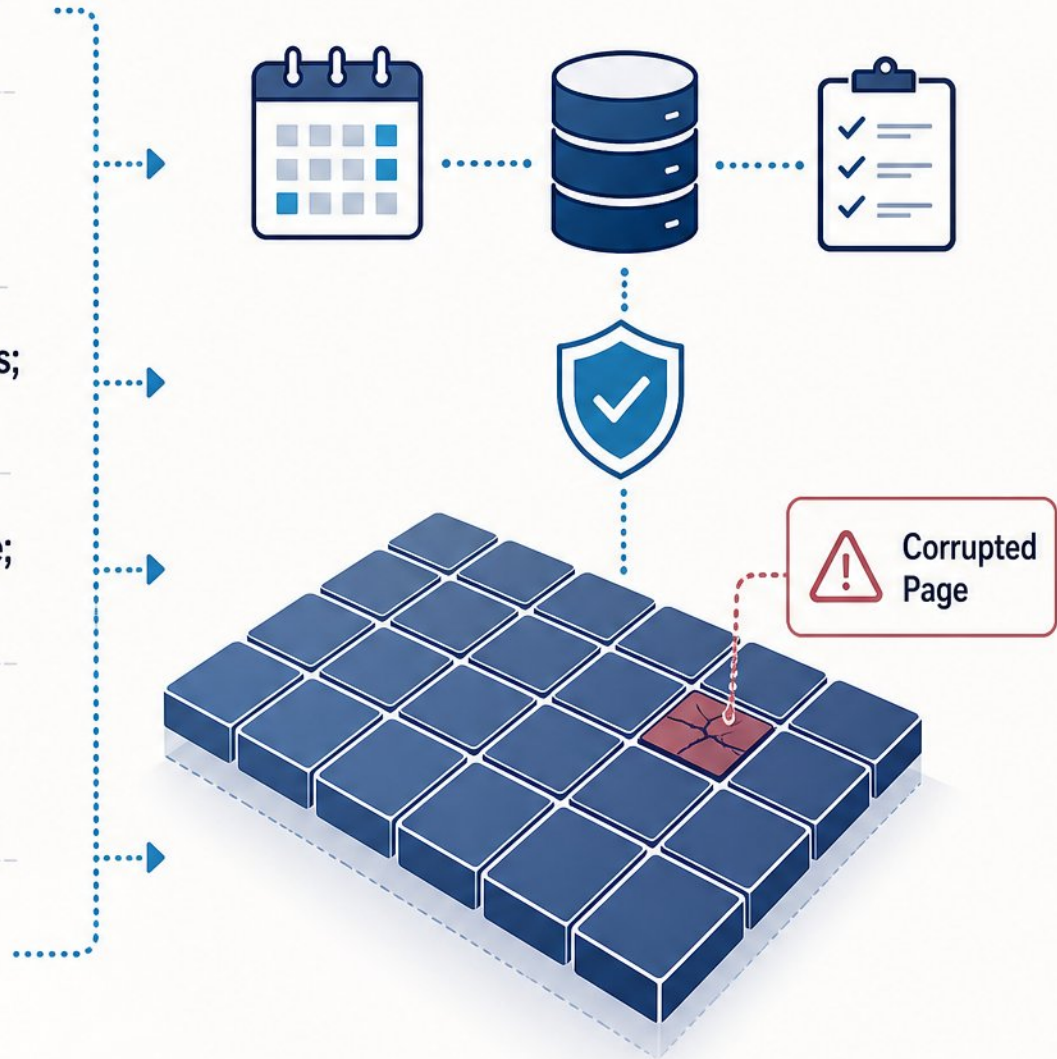
- Filter sys.databases for ONLINE state; be deliberate with system databases



- Verify backups separately: RESTORE VERIFYONLY plus checksummed backups



- Watch msdb.dbo.suspect_pages as the earliest storage warning



Assessment: Interpreting Output and Choosing a Recovery Path



- Read the full DBCC CHECKDB output, not just the final repair-level line



- Earlier messages identify objects, indexes, pages, and error types



- Minimum repair level states what repair would take, not what to run



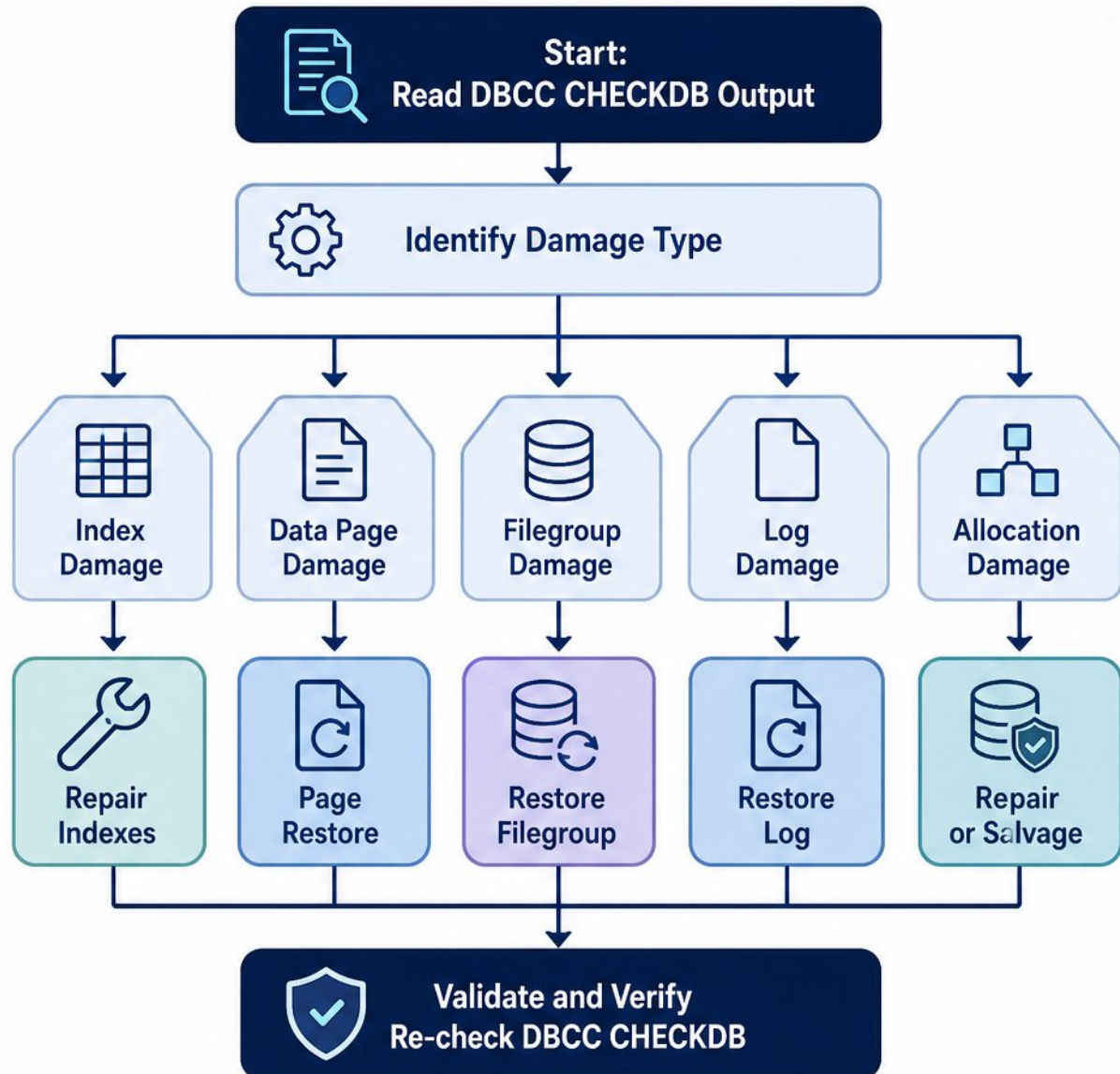
- Decide by damage type: index, data page, filegroup, log, allocation



- Check database and log state via sys.databases and sys.master_files



- Isolate root cause before repeating; recurring corruption means hardware



Containment: Stabilize, Preserve, and Communicate



- Restrict to **EMERGENCY** or **SINGLE_USER** only with a clear rollback path



- Back up the tail log where the log is accessible and recovery model permits



- Use **NORECOVERY**, **NO_TRUNCATE**, or **CONTINUE_AFTER_ERROR** as the situation dictates



- Preserve original data and log files; avoid detach/attach or manual log deletion



- Never repair the only copy — disable backup-deletion jobs, communicate data-loss risk



Recovery Path 1: Restore from Backup



- Known-good backup restore is the primary corruption fix — prefer it over repair



- Take a tail-log backup first to capture unbacked transactions and protect the chain



- Restore full, then differential, then log backups in order — all WITH NORECOVERY



- Apply RECOVERY only once, at the very end of the restore sequence

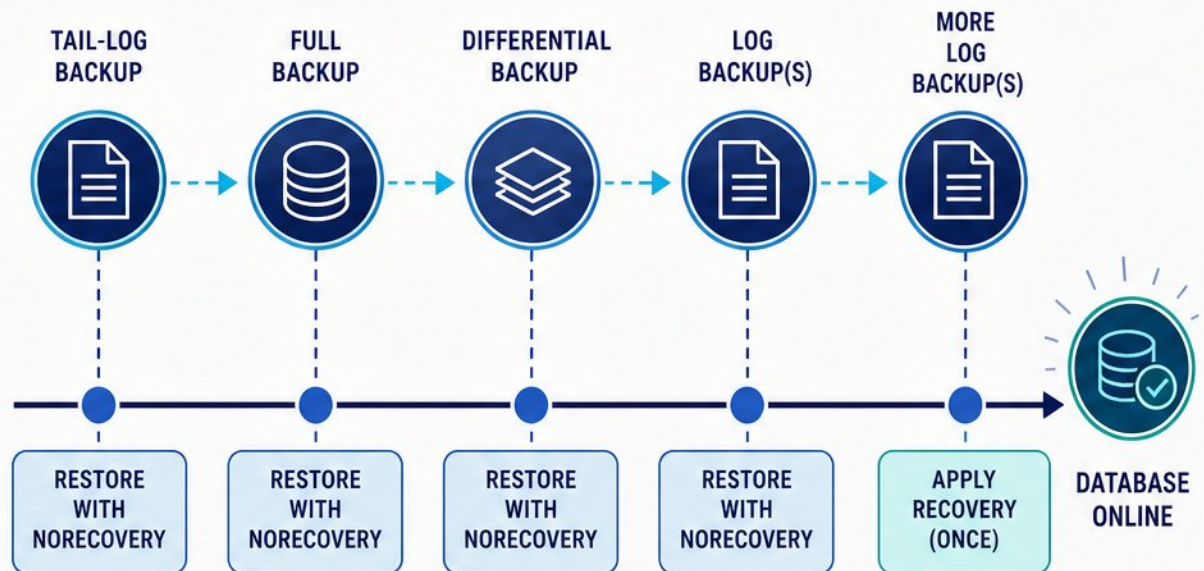


- Restore to a point in time before the bad write or corruption occurred



- Validate the restored database with DBCC CHECKDB before pointing applications at it

BACKUP CHAIN TIMELINE



RESTORE TO A POINT IN TIME BEFORE THE BAD WRITE OR CORRUPTION



BAD WRITE /
CORRUPTION OCCURRED



Recovery Path 2: Page Restore and Index Rebuilds



- Page restore targets isolated damaged pages — faster than a full restore



- Requires FULL or BULK_LOGGED recovery plus an unbroken log chain



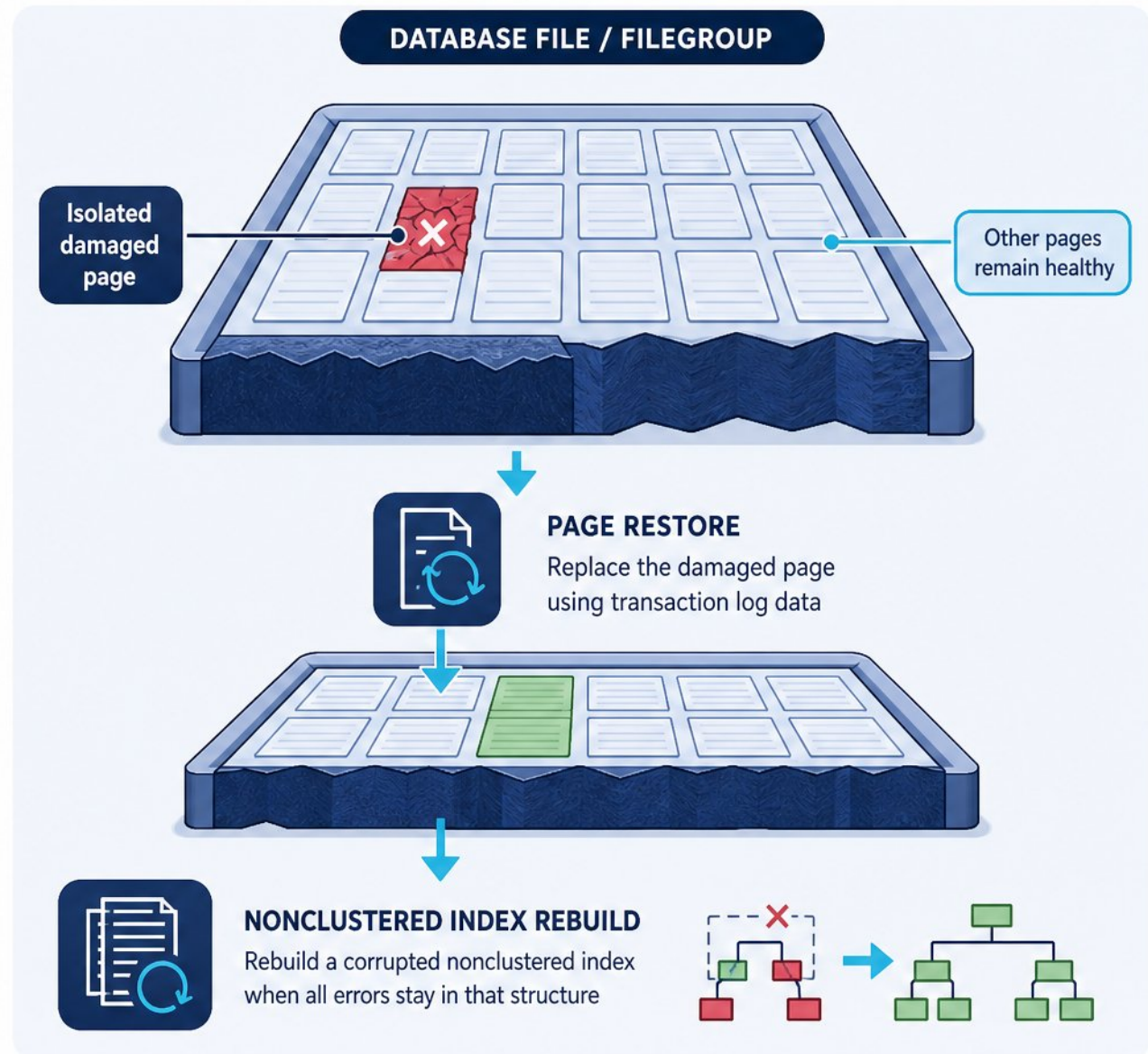
- Online page restore needs Enterprise Edition; offline works in all editions



- Cannot fix allocation pages, boot pages, file headers, or full-text catalogs



- Rebuild a corrupted nonclustered index when all errors stay in that structure





Recovery Path 3: DBCC Repair as Last Resort



DETECT



ASSESS



CONTAIN



RECOVER



VALIDATE



HARDEN

- - REPAIR_* options only when no restore path exists; REPAIR_FAST performs no repairs.
- - REPAIR_REBUILD: no data loss, but only a narrow class of errors qualify.
- - REPAIR_ALLOW_DATA_LOSS: deallocates damaged pages, losing every row on them.
- - Database must be in single-user mode; wrap in an explicit transaction where supported.
- - Emergency-mode repair cannot be rolled back, can rebuild the log, breaks ACID.

SINGLE-USER MODE REQUIRED



REPAIR_REBUILD



no data loss, but only a narrow class of errors qualify.

REPAIR_ALLOW_DATA_LOSS



deallocates damaged pages, losing every row on them.



Last resort action. Use only when no other restore path exists.

Salvage: Exporting Data When Repair Is Not Safe



- When DBCC cannot repair cleanly and no backup exists, salvage into a clean database



- Script the schema out, then export healthy tables with BCP, SSIS, or SELECT INTO



- EMERGENCY mode is the read window; access limited to sysadmin, logging disabled



- Skip broken tables and expect logical inconsistencies across relationships



- Treat any REPAIR_ALLOW_DATA_LOSS database as salvaged, not production-ready



- Rebuild indexes and validate dependencies intensively before reusing the data



Validation After Repair or Recovery



- Re-run DBCC CHECKDB with no repair option; confirm zero errors remain



- "Fixed" means structurally consistent, not recovered



- Run DBCC CHECKCONSTRAINTS and reconciliation to catch logical breakage



- Compare row counts and critical aggregates against a known-good source



- Take a checksummed full backup, restore-test it, monitor for recurring 823/824/825



DETECT



ASSESS



CONTAIN



RECOVER



VALIDATE



HARDEN

Prevention and Hardening



- Backup chain: full, differential, and log backups, each written **WITH CHECKSUM**



- Keep backups long enough to survive late-discovered corruption



- Enable **PAGE_VERIFY CHECKSUM** on every database; migrate old **NONE/TORN_PAGE** settings



- Configure alerts for 823, 824, and 825 — SQL Server ships with none



- Run **CHECKDB** often enough that corruption is found while good backups remain



- Patch drivers and firmware, use ECC memory and UPS protection



- Do restore drills and keep runbooks; an untested backup is a hope

DEFENSE IN DEPTH



CHECKSUMMED BACKUPS
Full, Differential, and Log
Each Written **WITH CHECKSUM**



RETENTION
Keep Long Enough to Survive
Late-Discovered Corruption



PAGE_VERIFY CHECKSUM
Enabled on Every Database
Migrate Old **NONE/TORN_PAGE** Settings



CORRUPTION ALERTS
Configure Alerts for 823, 824, and 825 —
SQL Server Ships with None



REGULAR CHECKDB
Run Often Enough That Corruption Is Found
While Good Backups Remain



INFRASTRUCTURE HYGIENE
Patch Drivers and Firmware,
Use ECC Memory and UPS Protection



RESTORE DRILLS & RUNBOOKS
Do Restore Drills and Keep Runbooks;
an Untested Backup Is a Hope

I/O PATH CHAIN



DISK / STORAGE



STORAGE CONTROLLER

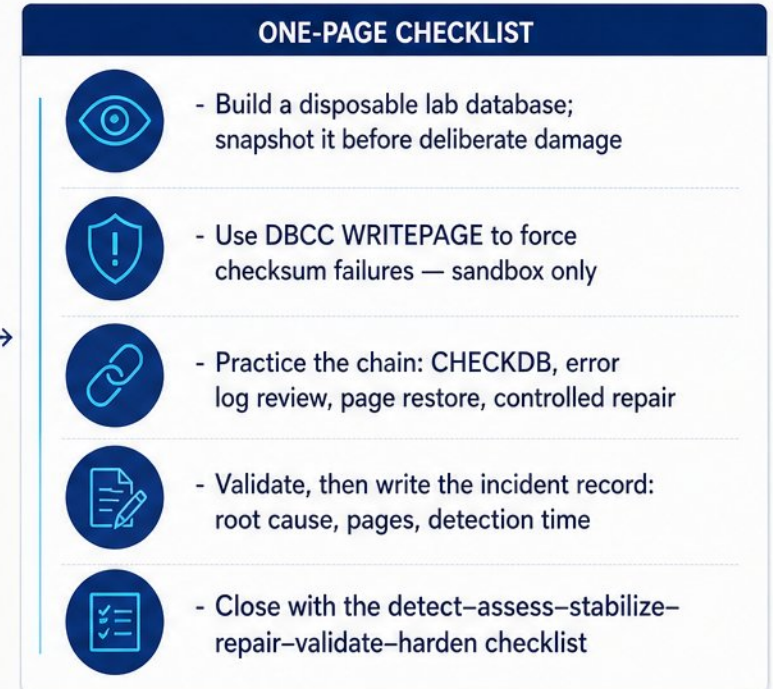
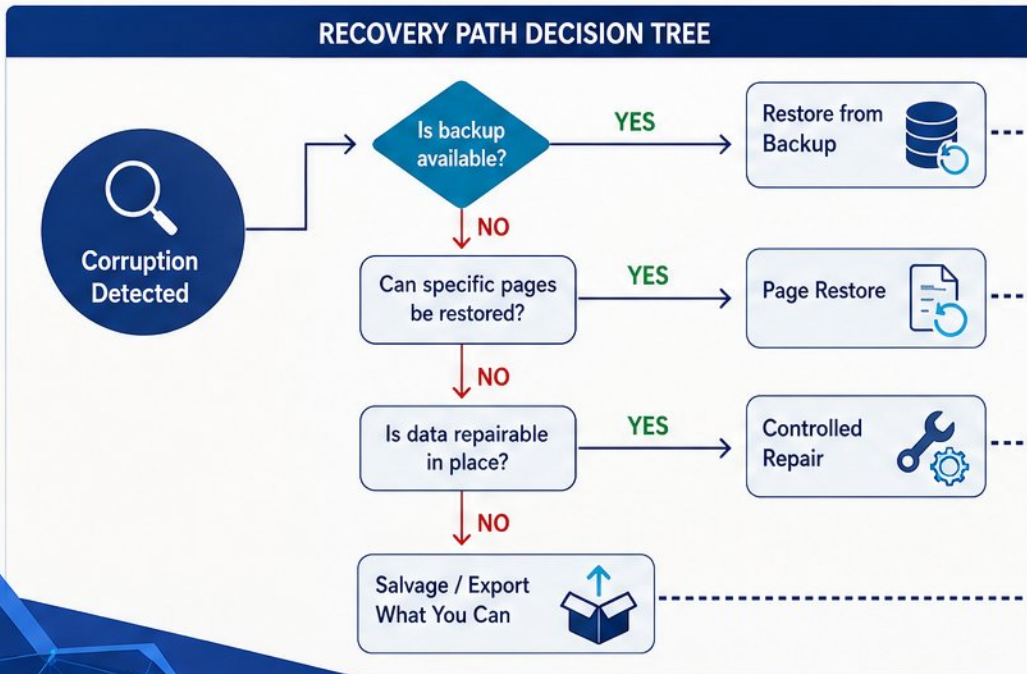
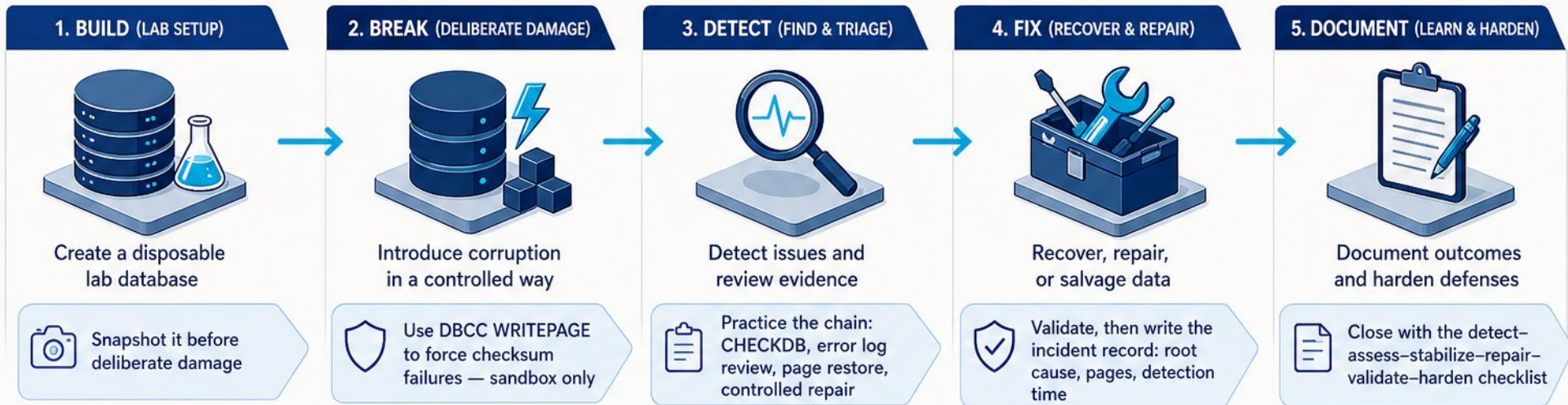


SYSTEM / HOST



SQL SERVER

Hands-On Lab: Break It, Detect It, Fix It, Document It



PersonWise.

PersonWise • AI digital-human course platform



Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/b69d7080/public