


Data Integrity in Azure SQL Database

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:28 AM PST

[[TOC]]

The content of this article was copied from [Data Integrity in Azure SQL Database](#) .

Data Integrity in Azure SQL Database

Posted on October 2, 2017 Peter Carlin CVP Azure Database Services

Microsoft takes data integrity very seriously. While there are some traditional techniques used by DBAs in SQL Server to monitor data integrity (e.g. DBCC CHECKDB) and various methods to recover from database corruptions, the Azure SQL Database engineering team has been able to develop new techniques that can handle some classes of corruption automatically and without data loss. These techniques are used as part of the service to avoid data loss and downtime in the cases where they can be avoided.

This blog post outlines some of those techniques, how they work, and what impact it has on customers concerned with what steps they should take to safeguard their data in Azure SQL Database.

How we manage data integrity for Azure SQL Database

The job of protecting data integrity in Azure SQL Database involves a combination of techniques and evolving methods:

Extensive data integrity error alert monitoring. Azure SQL Database emits alerts for all errors and unhandled exceptions that indicate data integrity concerns. Each alert is routed to the engineering team for manual handling and investigation (and the general process followed is explained below). This alerting system has caught more integrity issues than anything else described in this blog post.

Backup and restore integrity checks. On an ongoing basis, the Azure SQL Database engineering team automatically tests the restore of automated database backups of databases across the service. Upon restore, databases also receive integrity checks using DBCC CHECKDB. Any issues found during the integrity check will result in an alert to the engineering team.

I/O system “lost write” detection. Azure SQL Database has additional functionality to detect what has been the most common cause of observed physical corruption issues; I/O system “lost writes”. This functionality tracks page writes and their associated LSNs (Log Sequence Numbers). Any subsequent read of a data page from disk will be compared with the page’s expected LSN. If there is a mismatch in LSNs between what is on disk and what is expected, the page will be considered stale, resulting in an immediate alert to the engineering team.

Automatic Page Repair. Azure SQL Database leverages database replicas behind-the-scenes for business continuity purposes, and so the service also then leverages automatic page repair, which is the same technology used for SQL Server database mirroring and availability groups. In the event that a replica cannot read a page due to a data integrity issue, a fresh copy of the page will be retrieved from another replica, replacing the unreadable page without data loss or customer downtime. This functionality applies to premium tier databases and standard tier databases with geo-secondaries.

Data integrity at rest and in transit. Databases created in the service are by default set to verify pages with the CHECKSUM setting, calculating the checksum over the entire page and storing in the page header. Transport Layer Security (TLS) is also used for all communication in addition to the base transport level checksums provided by TCP/IP.

How we handle data integrity incidents

Incidents of wrong results or corruption are treated with the highest severity, with 24x7 work and support from across all Azure engineering teams. The goals when handling integrity incidents are to minimize unavailability and minimize the amount of data loss.

Addressing System Data Integrity Issues

Incidents that do not impact customer data or database availability will be corrected without notifying customers. Examples include those incidents that automatic page repair can address, or corruption to internal database metadata or telemetry that does not affect customer data or query results.

Addressing Customer Data Integrity Issues

All wrong-results and customer data corruption issues are communicated to impacted customers as soon after confirmed detection as possible. Azure SQL DB engineers will:

1. Work directly with the customer to explain the scope of corruption, outlining recovery options, and allowing the customer to choose the option that works best for their application and scenario.
2. Immediately initiate a restore to the point just prior to corruption, and make that available free of charge to the customer under a different database name. Customers for whom availability is the highest priority often choose to begin using this version of the database while recovery operations on the version with some corrupted data occur in-parallel. In this case, after repair of the original database, the support engineer will assist the customer in identifying what data has diverged between the two and what operations should occur to reconcile.
3. Where possible, assist the customer in understanding the scope of impact to their application, for example identifying whether data corruption has caused the application to change other data in an unexpected way.

Repair is achieved using various methods, and with steps taken in conjunction with customers. Options can include but are not limited to:

- Rebuilding the index – for example for a non-clustered index where the clustered index or heap is not also corrupted. Running DBCC CHECKDB with REPAIR_REBUILD where the repair has no possibility of data loss.
- Running DBCC CHECKDB with REPAIR_ALLOW_DATA_LOSS where repairs can cause some data loss.
- For scenarios where DBCC CHECKDB cannot be used to repair the data integrity issue, engineers use point-in-time restore to the point before the data integrity issue occurred plus manual replay of relevant transactions from the transaction log. An example when this occurs is when the transaction log has been corrupted in a way that prevents automatic replay but does not affect customer data. Issues leading to wrong-results or data corruption receive detailed post mortems from the engineering team and associated repair items created because of the issue are closely tracked. Many significant enhancements have occurred because of these post mortems, including the “lost writes” functionality described earlier.

Running integrity checks in Azure SQL Database

The Azure SQL Database engineering team takes responsibility for managing data integrity. As such, it is not necessary for customers to run integrity checks in Azure SQL Database.

With the existing monitoring and protection provided by the service, customers can still choose to execute user-initiated integrity checks in Azure SQL Database. For example, customers may optionally run DBCC CHECKDB.

Evolving methodologies

The Azure SQL Database engineering team regularly reviews and continues to enhance the data integrity issue detection capabilities of the software and hardware used in the service. Although rare, if a data integrity error is encountered prior to receiving notification from Microsoft customer support, customers should file a support case.

If you have feedback to share regarding Microsoft's data integrity strategy in Azure SQL Database, we would like to hear from you. To contact the engineering team with feedback or comments on this subject, please email: SQLDBArchitects@microsoft.com.

How good have you found this content?

