

SQL ledger database tables and schema in Azure SQL Database

Last updated by | Vitor Tomaz | Jun 8, 2022 at 5:35 AM PDT

Contents

- [SQL ledger append-only ledger tables](#)
 - [Append-only ledger table schema](#)
 - [Ledger view](#)
 - [Ledger view schema](#)

SQL ledger append-only ledger tables

Append-only ledger tables allow only INSERT operations on your tables, ensuring that privileged users such as DBAs cannot alter data through traditional DML operations. Append-only ledger tables are ideal for systems that do not update or delete records, such as Security Information Event and Management (SIEM) systems or blockchain systems where data needs to be replicated from the blockchain to a database. Since there are no UPDATE or DELETE operations on an append-only table, there is no need for a corresponding history table as there is with [Updatable ledger tables](#).

Creating an append-only ledger table can be done through specifying the LEDGER = ON argument in your create database T-SQL statement specifying the the APPEND_ONLY = ON option.

For details on options available when specifying the LEDGER argument in your T-SQL statement, see [CREATE TABLE \(Transact-SQL\)](#).

[!IMPORTANT] When creating an append-only ledger table, system-generated columns will be created in your ledger table for tracking data lineage for forensics purposes.

Ledger tables cannot be reverted non-ledger tables. This is to ensure an attacker cannot temporarily remove ledger capabilities on a ledger table, make changes, and then re-enable ledger functionality.

Append-only ledger table schema

When created, append-only ledger tables will add 2 system-generated, hidden columns to your table. These columns contain metadata noting which transactions made changes to your append-only ledger tables and the order of operations by which rows were updated by the transaction. This data is useful for forensics purposes in understanding how data was inserted over time.

Column name	Data type	Description
ledger_start_transaction_id	bigint	The id of the transaction that created a row version.
ledger_start_sequence_number	bigint	The sequence number of an operation within a transaction that created a row version.

Ledger view

For every append-only ledger table, the system automatically generates a view, called the ledger view, that reports all row inserts that have occurred on the table. The ledger view is primarily helpful for updatable ledger tables, rather than append-only ledger tables, as the latter does not have any UPDATE or DELETE capabilities. The ledger view for append-only ledger tables is available for consistency between both updatable and append-only ledger tables.

Ledger view schema

Column name	Data type	Description
ledger_transaction_id	bigint	The id of the transaction that created or deleted a row version.
ledger_sequence_number	bigint	The sequence number of a row-level operation within the transaction on the table.
ledger_operation_type_id	tinyint	Contains 0 – INSERT or 1 – DELETE. Inserting a row into the ledger table produces a new row in the ledger view containing 0 (INSERT) in this column. Deleting a row from the ledger table produces a new row in the ledger view containing 1 (DELETE) in this column. Updating a row in the ledger table produces two new rows in the ledger view. One row contains 1 (DELETE) and the other row contains 1 (INSERT) in this column.
ledger_operation_type_desc	nvarchar(128)	Contains 'INSERT' or 'DELETE'. See above for details.

How good have you found this content?

