SQL START! 2022

Data Integrity with SQL Database Ledger

Gianluca Hotz

ghotz@ugiss.org | @glhotz

Who am I?

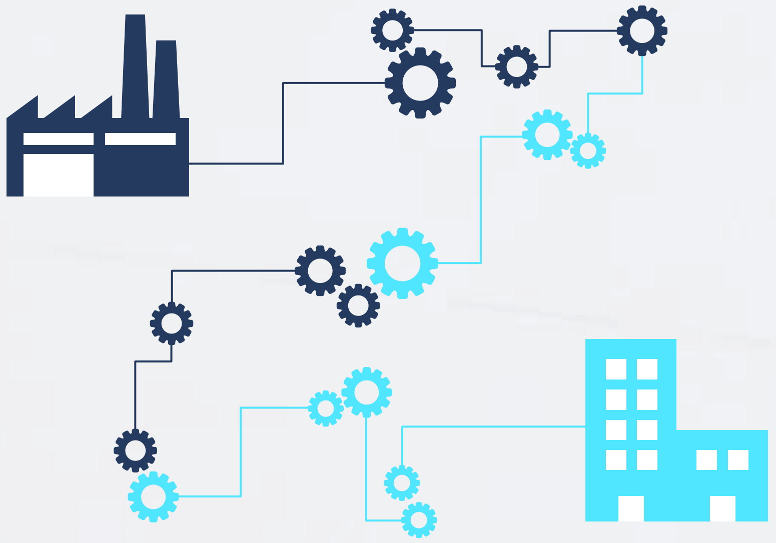
- Gianluca Hotz | @glhotz | ghotz@ugiss.org
- Independent Consultant
  - 25+ years on SQL Server (from 4.21 back in 1996)
  - Database modeling & development, sizing & administration, modernization (upgrades & migrations), performance tuning, security
- Community
  - 24 years Microsoft MVP SQL Server/Data Platform (from 1998)
  - VMware Experts SQL Server
  - Founder and president UGISS (ex «PASS Chapter»)

Partners

UNIVERSITÀ
POLITECNICA
DELLE MARCHE

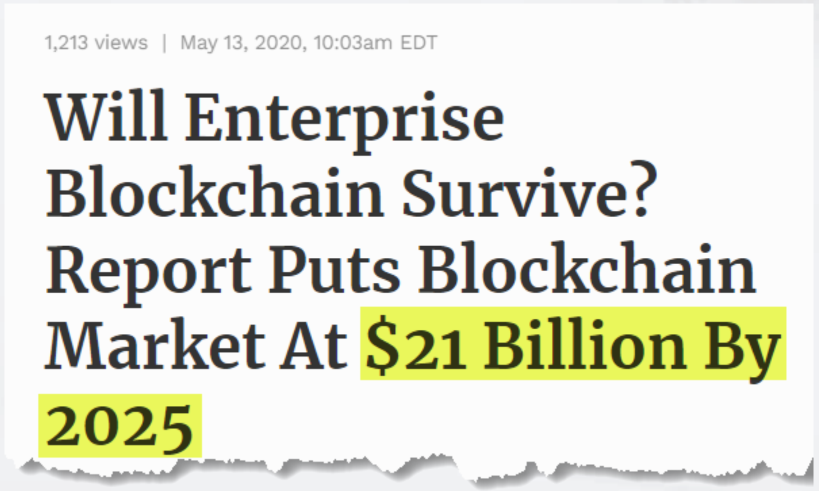
Ledger technologies enable digital trust

Companies are moving from intermediaries and manual auditing that are slow and costly...



...to ledger technologies that reduce cost, saves time, and lowers risk

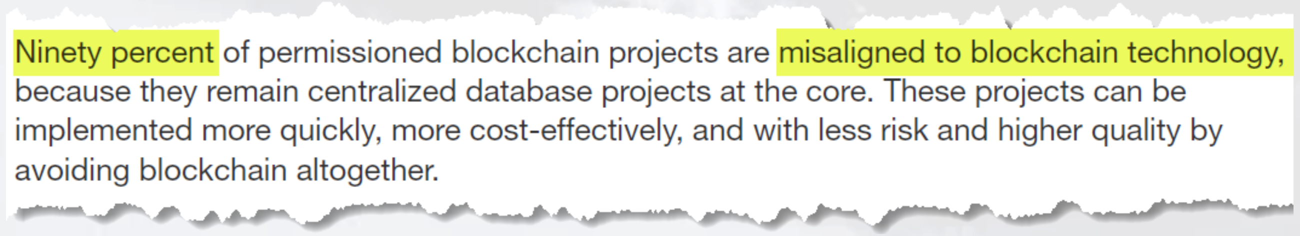
Blockchain market growth predictions are growing



1,213 views | May 13, 2020, 10:03am EDT

Will Enterprise Blockchain Survive? Report Puts Blockchain Market At $21 Billion By 2025

https://www.forbes.com/sites/robertanzalone/2020/05/13/will-enterprise-blockchain-survive-a-new-report-says-that-the-blockchain-technology-market-will-reach-21-billion-by-2025/#7a5f793954b8

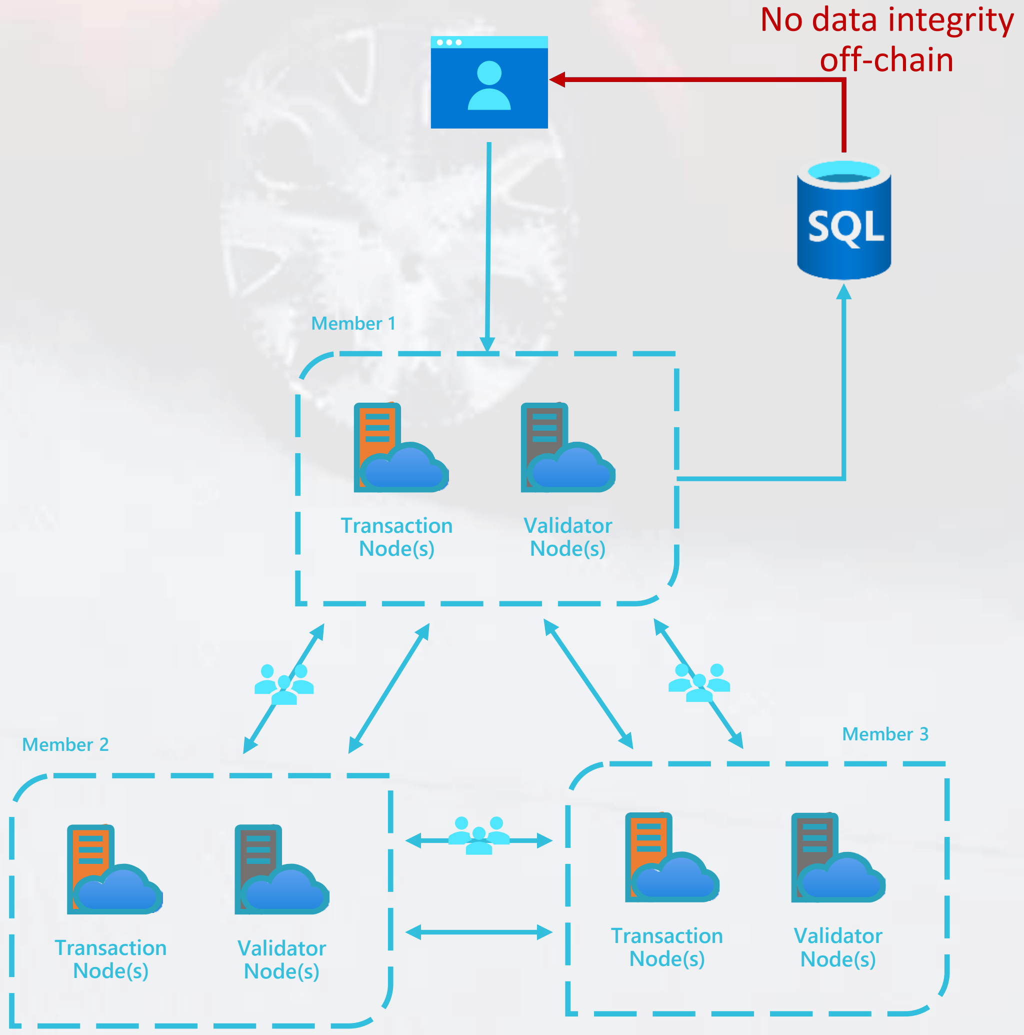


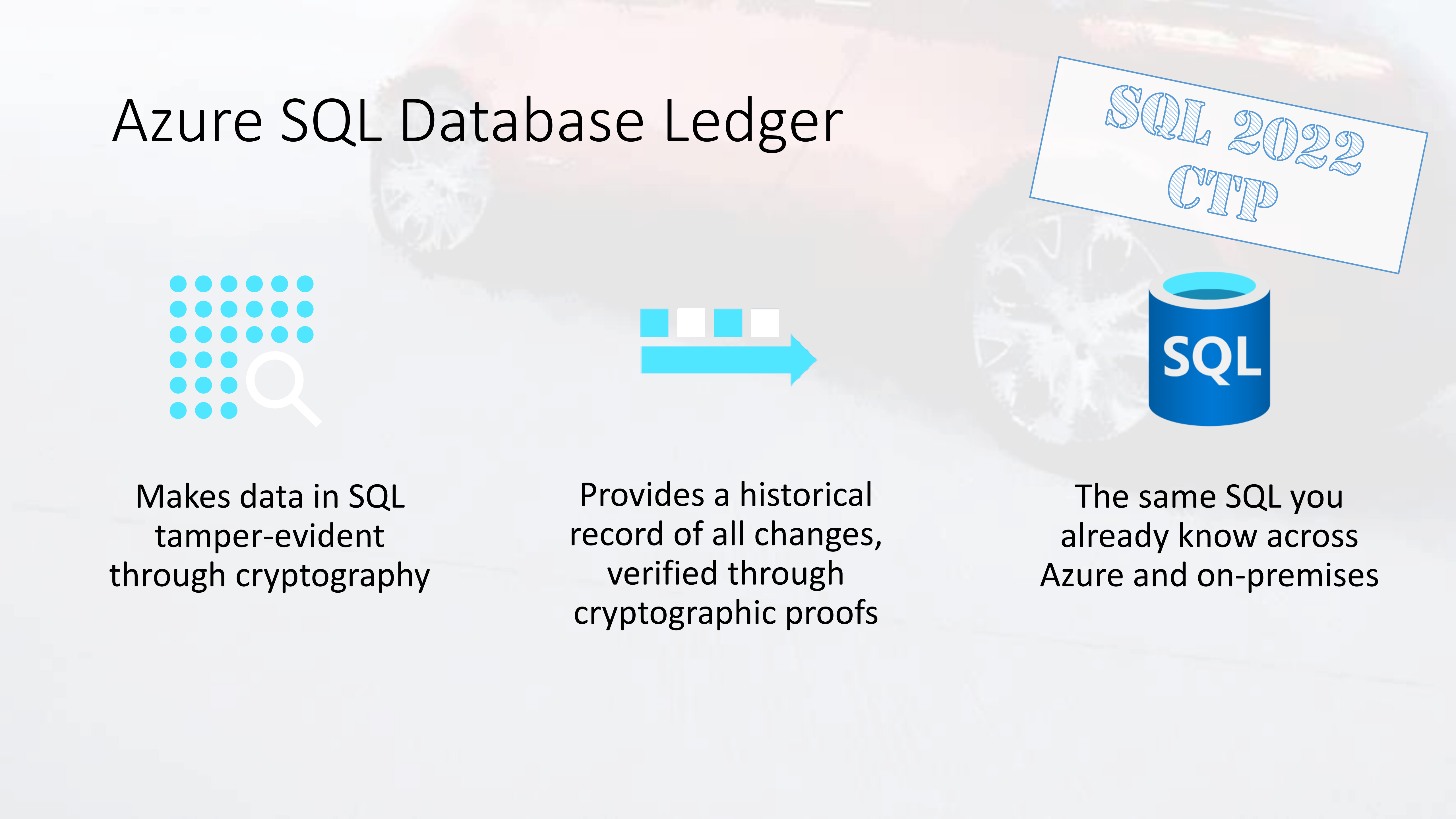
Ninety percent of permissioned blockchain projects are misaligned to blockchain technology, because they remain centralized database projects at the core. These projects can be implemented more quickly, more cost-effectively, and with less risk and higher quality by avoiding blockchain altogether.

Gartner Predicts 2019: Blockchain Technologies

Blockchains overkill for centralized scenarios

- Decentralization requires all parties to host nodes on the network to participate in consensus
- Governance rules must be established by the consortium and deployed/managed
- Latency associated with network consensus can impact transaction throughput (<1000 TPS for Ethereum)
- Off-chain storage patterns for querying data are a typical pattern, but data integrity is lost in the process
- Bespoke development with immature tooling makes development and management challenging





Azure SQL Database Ledger

SQL 2022
CTP

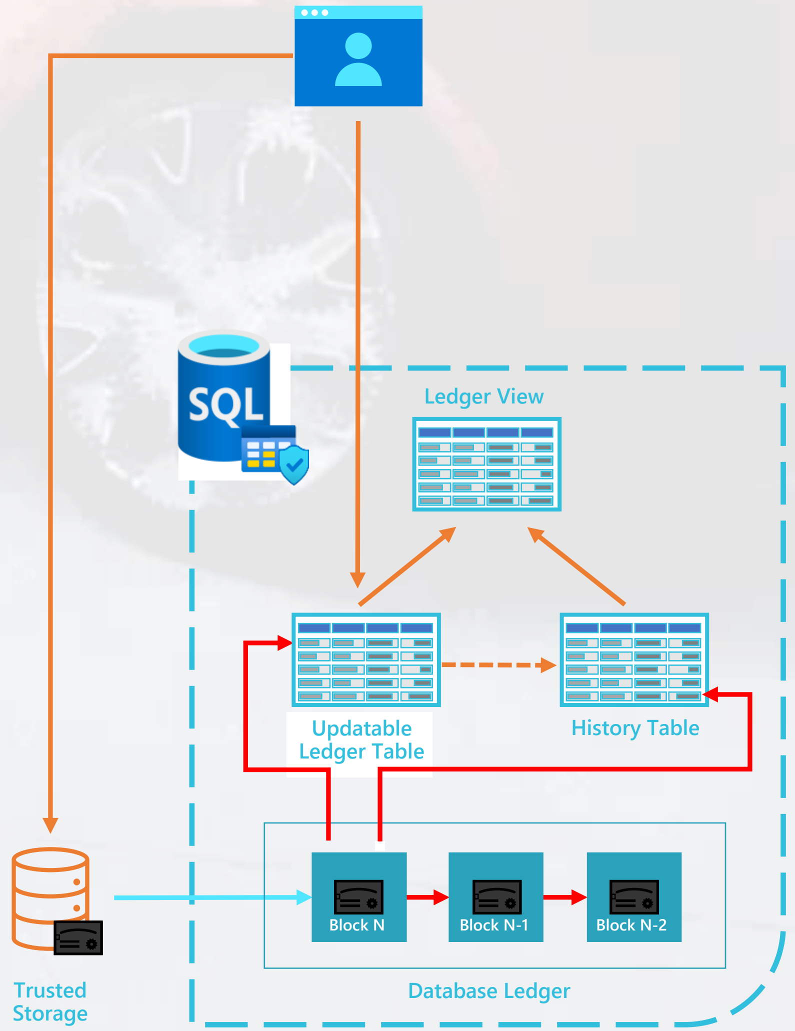
Makes data in SQL tamper-evident through cryptography

Provides a historical record of all changes, verified through cryptographic proofs

The same SQL you already know across Azure and on-premises

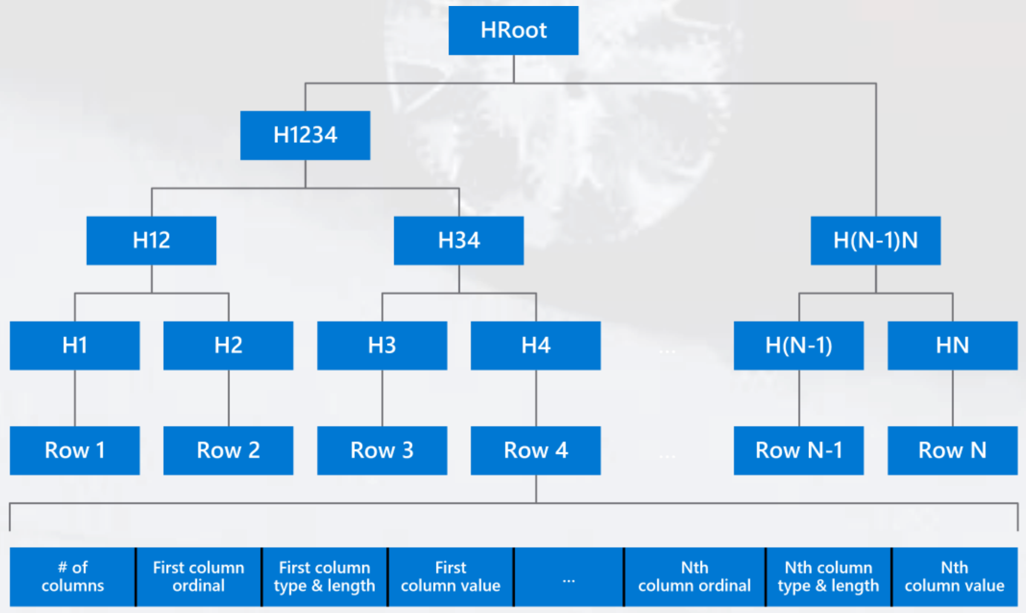
Ledger Tables

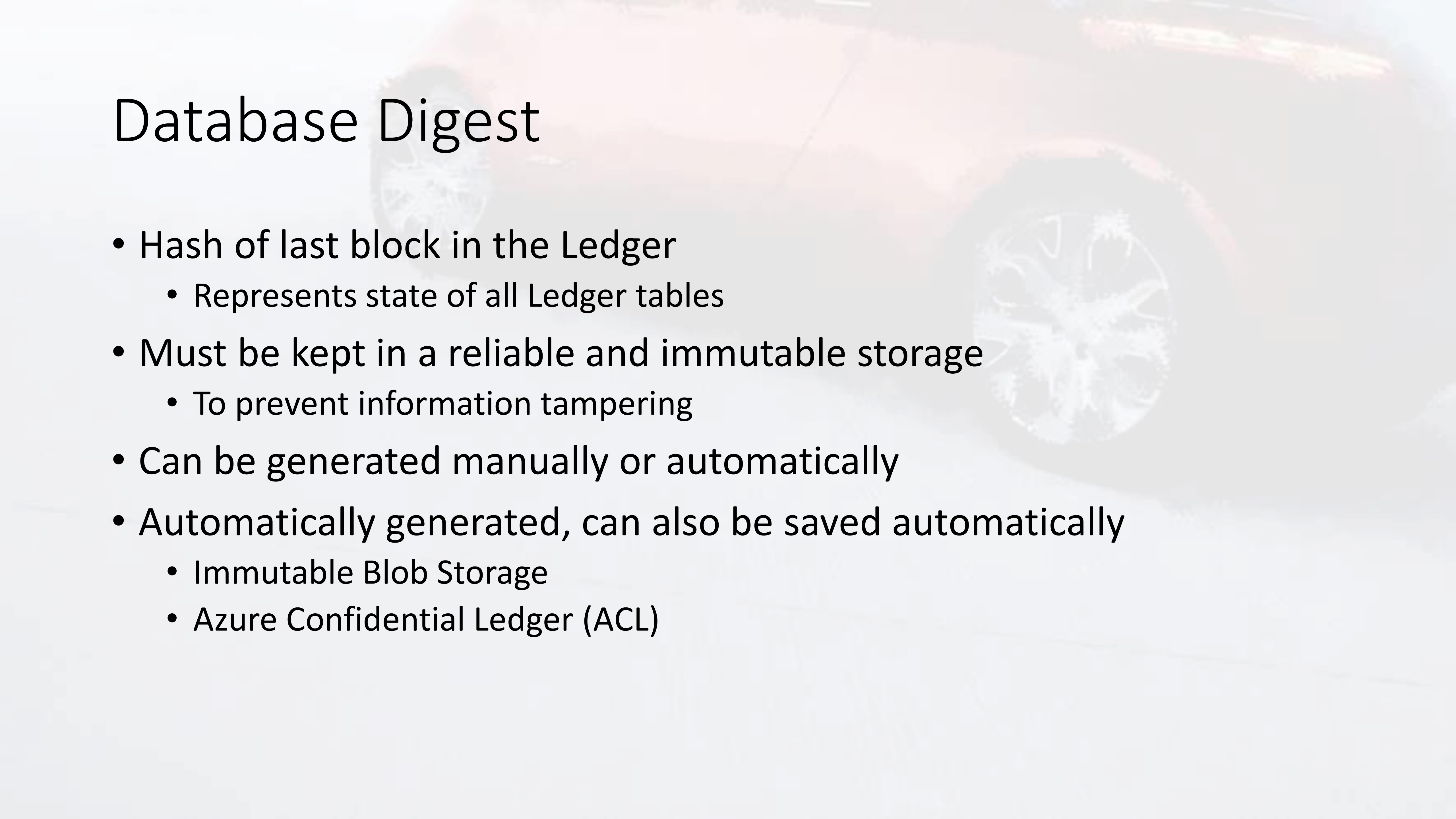
- **Updatable** allow insert/update/delete
- History of updated/deleted rows preserved in history table and easy-to-query Ledger View
- Integrity of updatable/history tables maintained through cryptographic links of the Database Ledger
- System can periodically upload digital receipts to a customer-configured trusted storage service
- Customer can use digital receipts to verify the integrity of the data
- **Append-Only** allow only insert
    - no need for a history table



Database Ledger

- Incrementally capture database state
  - At logical level
    - blockchain and Merkle Tree data structures
  - Captures also transaction metadata
    - e.g., timestamp, user
- Blocks and transaction information in system tables
  - sys.database_ledger_transactions
  - sys.database_ledger_blocks
- Block are closed
  - every 30 seconds
  - manually executing sys.sp_generate_database_ledger_digest





Database Digest

- Hash of last block in the Ledger
  - Represents state of all Ledger tables
- Must be kept in a reliable and immutable storage
  - To prevent information tampering
- Can be generated manually or automatically
- Automatically generated, can also be saved automatically
  - Immutable Blob Storage
  - Azure Confidential Ledger (ACL)

Reliable (trusted) Storage

«Immutable Blob Storage»

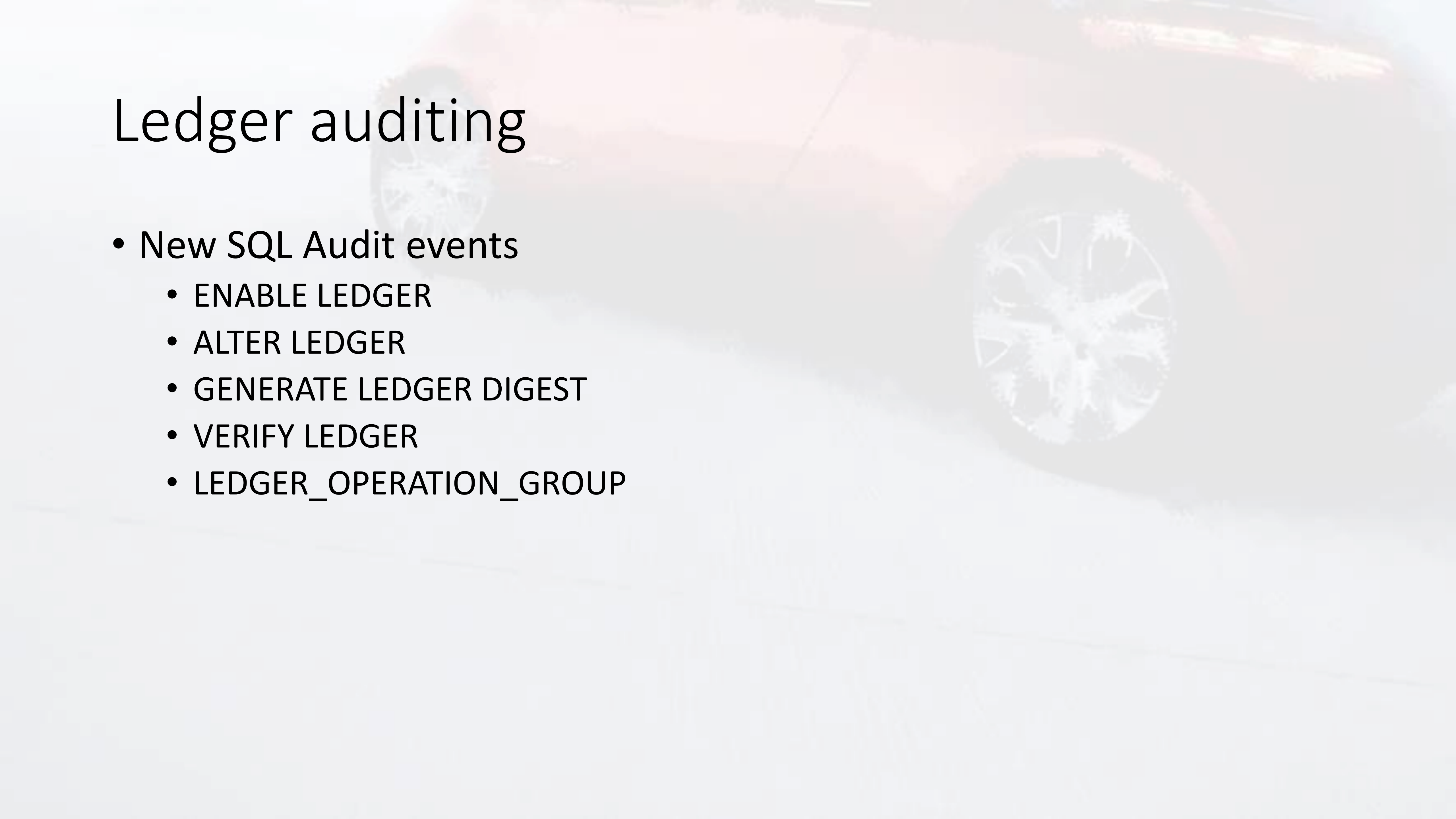
- Storage “Write Once, Read Many” based on **policies**
- BLOBs can be set as read-only for specified range
- Data blocked only functionally based on policy
- Support for audit logging but log creator must be trusted
- Microsoft is the TCB
  - Trusted Computing Base

«Azure Confidential Ledger» (ACL)

- Storage "Write Once, Read Many" **permanently**
- BLOBs written in the Ledger cannot be edited
- Uses tamper-proof “Confidential Enclaves”
- Create serialized Ledger files and transaction receipts that can be verified by customers
- Microsoft is outside the TCB
  - source code is open source (Confidential Consortium Framework)

Ledger verification

- Tampering possible even if changes forbidden, especially on-premises
  - e.g. direct modification of data files, DBCC WRITEPAGE, SQL Server process "hijacking", etc.
- Verify recomputes all hashes and compares them with digest
- When to verify
  - when necessary (e.g. suspected tampering, formal audit, litigation)
  - on a recurring basis (e.g. daily, hourly)
- Verification via system procedure depends on save mode
  - automatic: passing a BLOB Storage address
  - manual: passing JSON document containig the digest



Ledger auditing

- New SQL Audit events
  - ENABLE LEDGER
  - ALTER LEDGER
  - GENERATE LEDGER DIGEST
  - VERIFY LEDGER
  - LEDGER_OPERATION_GROUP

Demo

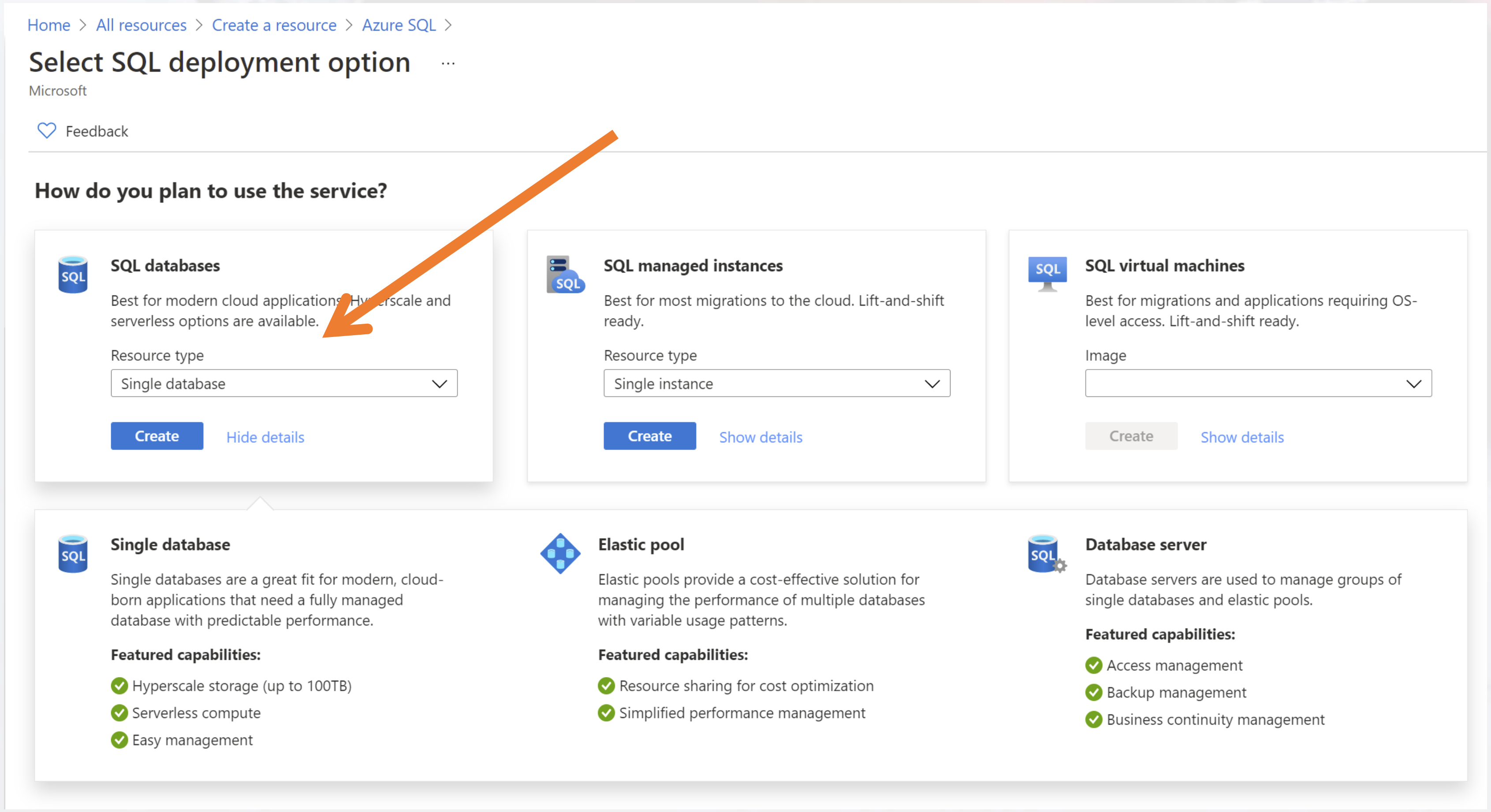
Enabling SQL Database Ledger

Ledger Tables

Demo

Enabling SQL Database Ledger

Select SQL deployment option



Home > All resources > Create a resource > Azure SQL >

Select SQL deployment option ...

Microsoft

Feedback

How do you plan to use the service?

SQL databases

Best for modern cloud applications. Hyperscale and serverless options are available.

Resource type: Single database

Create | Hide details

SQL managed instances

Best for most migrations to the cloud. Lift-and-shift ready.

Resource type: Single instance

Create | Show details

SQL virtual machines

Best for migrations and applications requiring OS-level access. Lift-and-shift ready.

Image:

Create | Show details

Single database

Single databases are a great fit for modern, cloud-born applications that need a fully managed database with predictable performance.

Featured capabilities:

- ✓ Hyperscale storage (up to 100TB)
- ✓ Serverless compute
- ✓ Easy management

Elastic pool

Elastic pools provide a cost-effective solution for managing the performance of multiple databases with variable usage patterns.

Featured capabilities:

- ✓ Resource sharing for cost optimization
- ✓ Simplified performance management

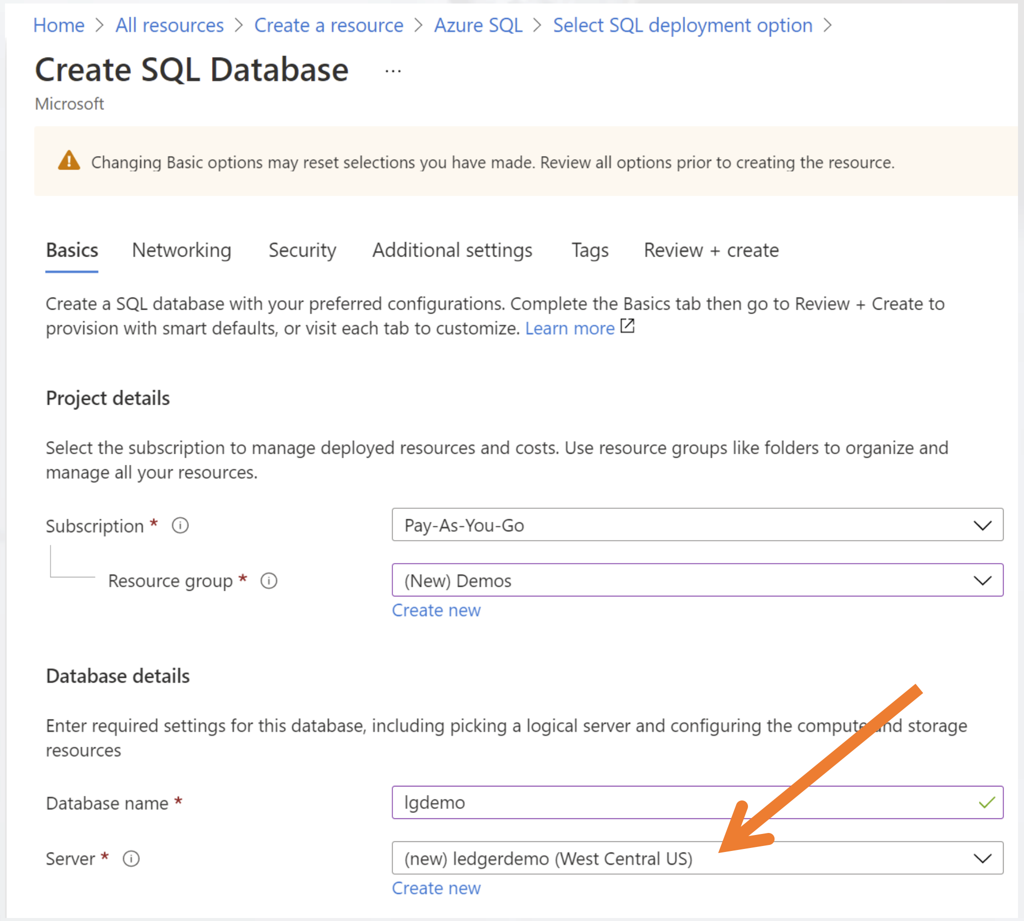
Database server

Database servers are used to manage groups of single databases and elastic pools.

Featured capabilities:

- ✓ Access management
- ✓ Backup management
- ✓ Business continuity management

Create SQL Database



Home > All resources > Create a resource > Azure SQL > Select SQL deployment option >

Create SQL Database ...

Microsoft

Changing Basic options may reset selections you have made. Review all options prior to creating the resource.

Create a SQL database with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults, or visit each tab to customize. Learn more

Project details

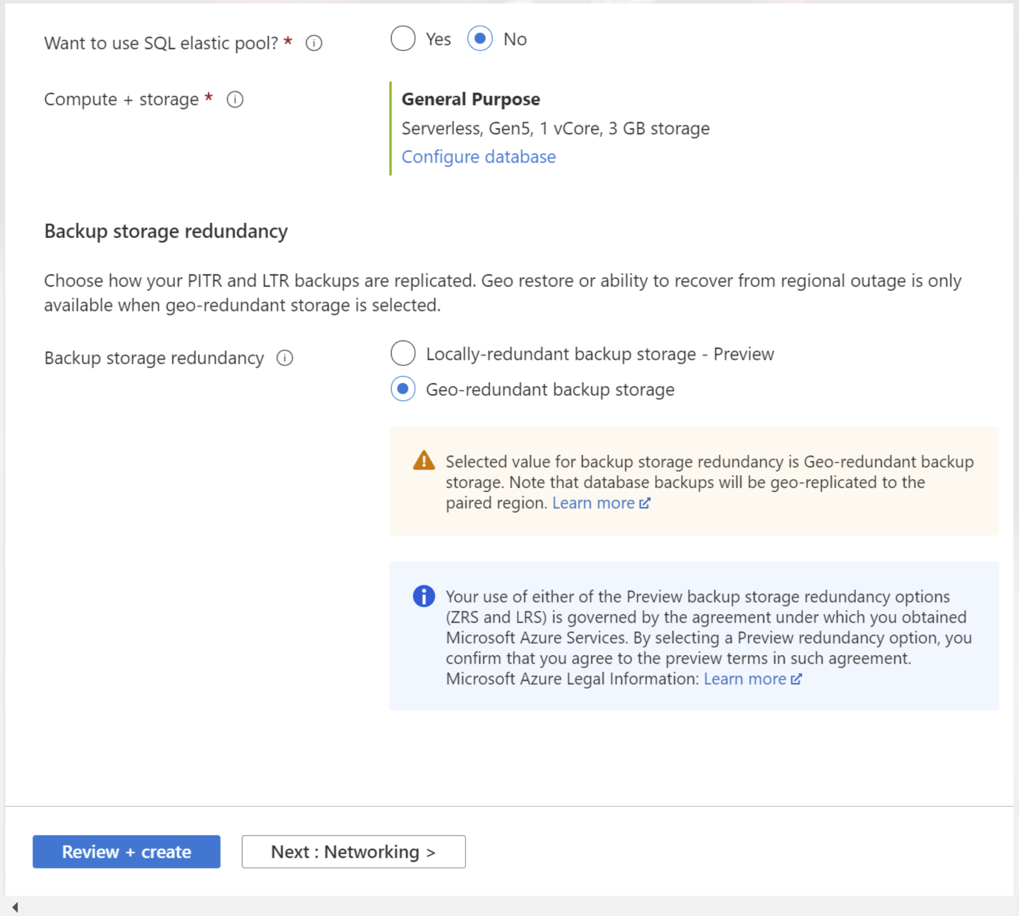
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	Pay-As-You-Go
Resource group *	(New) Demos
	Create new

Database details

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name *	lgdemo
Server *	(new) ledgerdemo (West Central US)
	Create new



Want to use SQL elastic pool? *	Yes [✓ No]
Compute + storage *	**General Purpose** Serverless, Gen5, 1 vCore, 3 GB storage Configure database

Backup storage redundancy

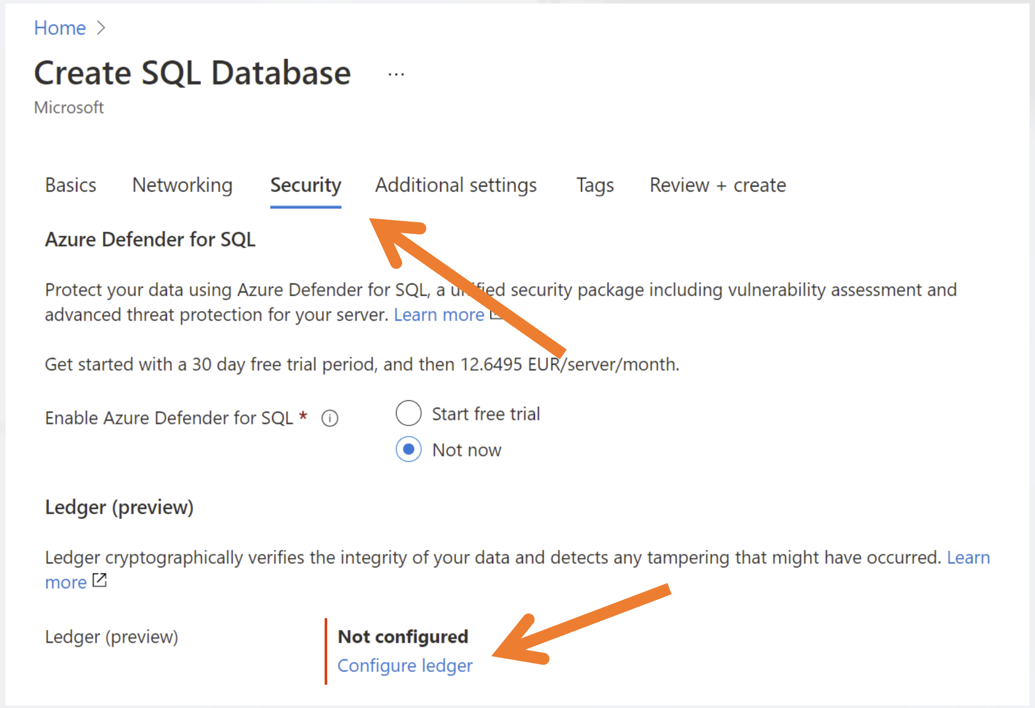
Choose how your PITR and LTR backups are replicated. Geo restore or ability to recover from regional outage is only available when geo-redundant storage is selected.

Backup storage redundancy: Locally-redundant backup storage - Preview [✓ Geo-redundant backup storage]

Selected value for backup storage redundancy is Geo-redundant backup storage. Note that database backups will be geo-replicated to the paired region. Learn more

Your use of either of the Preview backup storage redundancy options (ZRS and LRS) is governed by the agreement under which you obtained Microsoft Azure Services. By selecting a Preview redundancy option, you confirm that you agree to the preview terms in such agreement. Microsoft Azure Legal Information: Learn more

Ledger configuration


Home >

Create SQL Database

Microsoft

Basics | Networking | **Security** | Additional settings | Tags | Review + create

Azure Defender for SQL

Protect your data using Azure Defender for SQL, a unified security package including vulnerability assessment and advanced threat protection for your server. Learn more

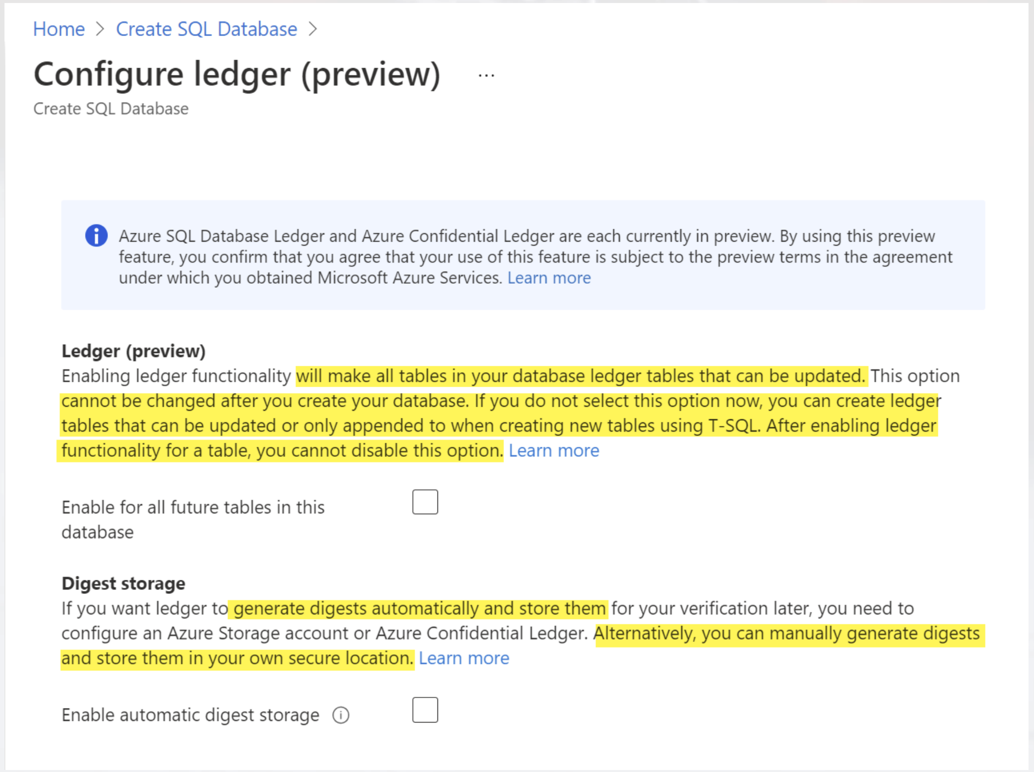
Get started with a 30 day free trial period, and then 12.6495 EUR/server/month.

Enable Azure Defender for SQL * : Start free trial [✓ Not now]

Ledger (preview)

Ledger cryptographically verifies the integrity of your data and detects any tampering that might have occurred. Learn more

Ledger (preview): **Not configured** Configure ledger


Home > Create SQL Database >

Configure ledger (preview)

Create SQL Database

Azure SQL Database Ledger and Azure Confidential Ledger are each currently in preview. By using this preview feature, you confirm that you agree that your use of this feature is subject to the preview terms in the agreement under which you obtained Microsoft Azure Services. Learn more

Ledger (preview)

Enabling ledger functionality will make all tables in your database ledger tables that can be updated. This option cannot be changed after you create your database. If you do not select this option now, you can create ledger tables that can be updated or only appended to when creating new tables using T-SQL. After enabling ledger functionality for a table, you cannot disable this option. Learn more

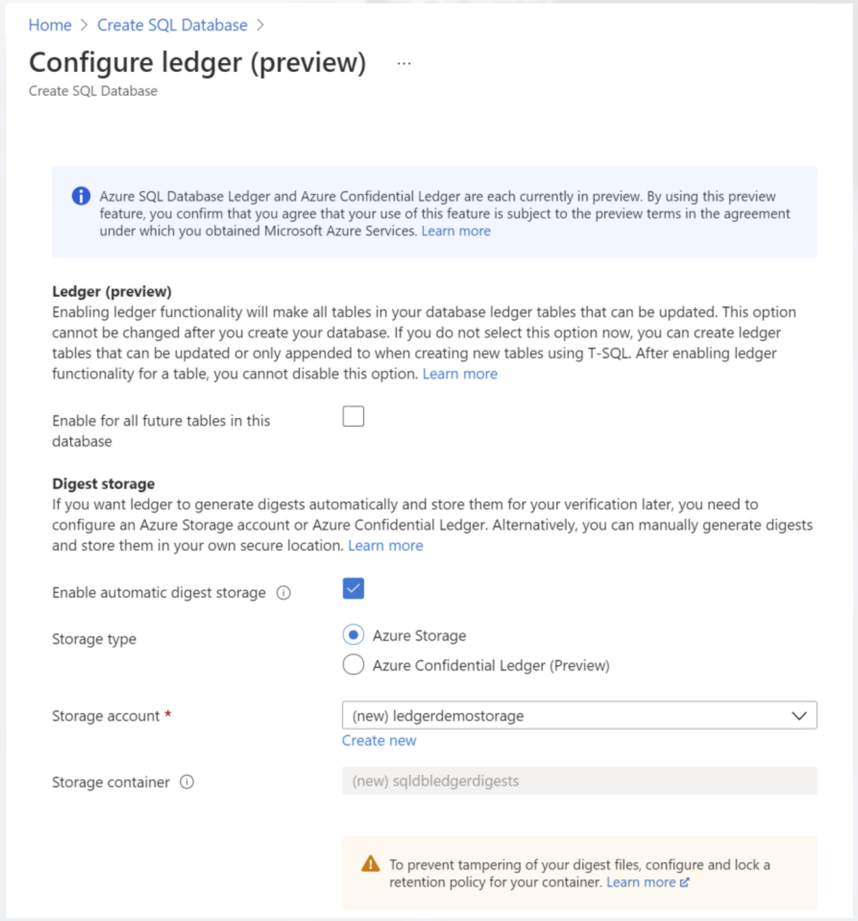
Enable for all future tables in this database ☐

Digest storage

If you want ledger to generate digests automatically and store them for your verification later, you need to configure an Azure Storage account or Azure Confidential Ledger. Alternatively, you can manually generate digests and store them in your own secure location. Learn more

Enable automatic digest storage ☐

Digest Storage



Home > Create SQL Database >

Configure ledger (preview) ...

Create SQL Database

Azure SQL Database Ledger and Azure Confidential Ledger are each currently in preview. By using this preview feature, you confirm that you agree that your use of this feature is subject to the preview terms in the agreement under which you obtained Microsoft Azure Services. Learn more

Ledger (preview)

Enabling ledger functionality will make all tables in your database ledger tables that can be updated. This option cannot be changed after you create your database. If you do not select this option now, you can create ledger tables that can be updated or only appended to when creating new tables using T-SQL. After enabling ledger functionality for a table, you cannot disable this option. Learn more

Enable for all future tables in this database ☐

Digest storage

If you want ledger to generate digests automatically and store them for your verification later, you need to configure an Azure Storage account or Azure Confidential Ledger. Alternatively, you can manually generate digests and store them in your own secure location. Learn more

Enable automatic digest storage ☑

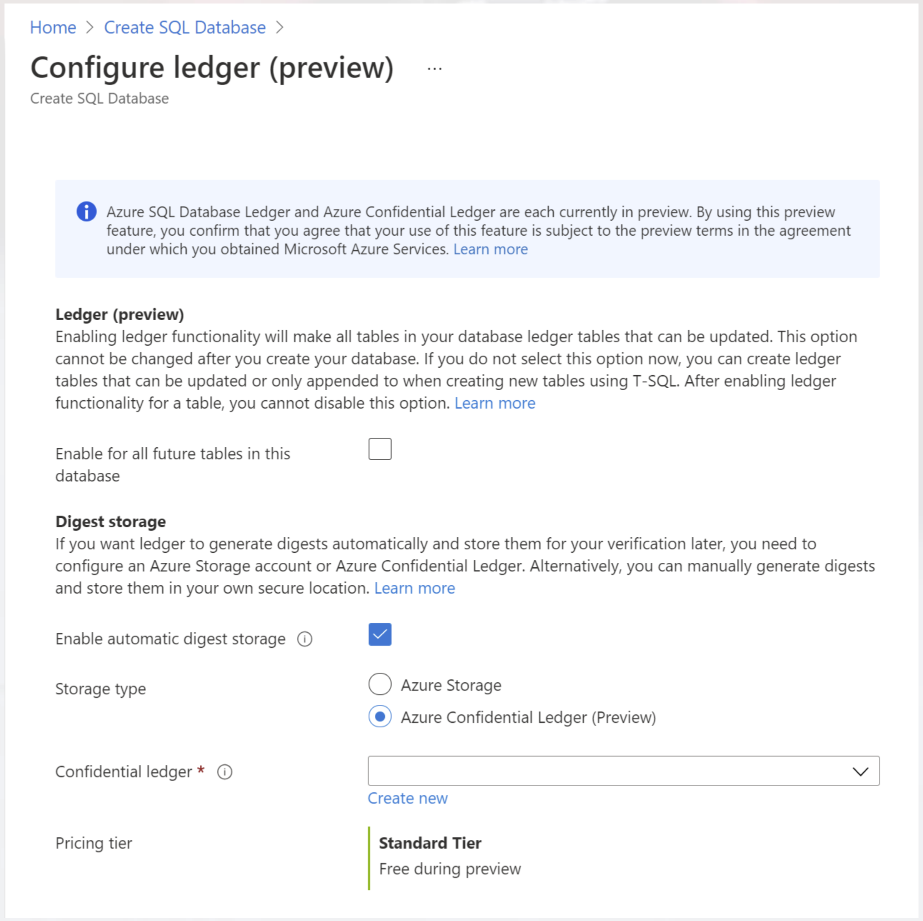
Storage type: [✓ Azure Storage] Azure Confidential Ledger (Preview)

Storage account *: (new) ledgerdemostorage

Create new

Storage container: (new) sqldbledgerdigests

To prevent tampering of your digest files, configure and lock a retention policy for your container. Learn more



Home > Create SQL Database >

Configure ledger (preview) ...

Create SQL Database

Azure SQL Database Ledger and Azure Confidential Ledger are each currently in preview. By using this preview feature, you confirm that you agree that your use of this feature is subject to the preview terms in the agreement under which you obtained Microsoft Azure Services. Learn more

Ledger (preview)

Enabling ledger functionality will make all tables in your database ledger tables that can be updated. This option cannot be changed after you create your database. If you do not select this option now, you can create ledger tables that can be updated or only appended to when creating new tables using T-SQL. After enabling ledger functionality for a table, you cannot disable this option. Learn more

Enable for all future tables in this database ☐

Digest storage

If you want ledger to generate digests automatically and store them for your verification later, you need to configure an Azure Storage account or Azure Confidential Ledger. Alternatively, you can manually generate digests and store them in your own secure location. Learn more

Enable automatic digest storage ☑

Storage type: Azure Storage [✓ Azure Confidential Ledger (Preview)]

Confidential ledger *:

Create new

Pricing tier: **Standard Tier** Free during preview

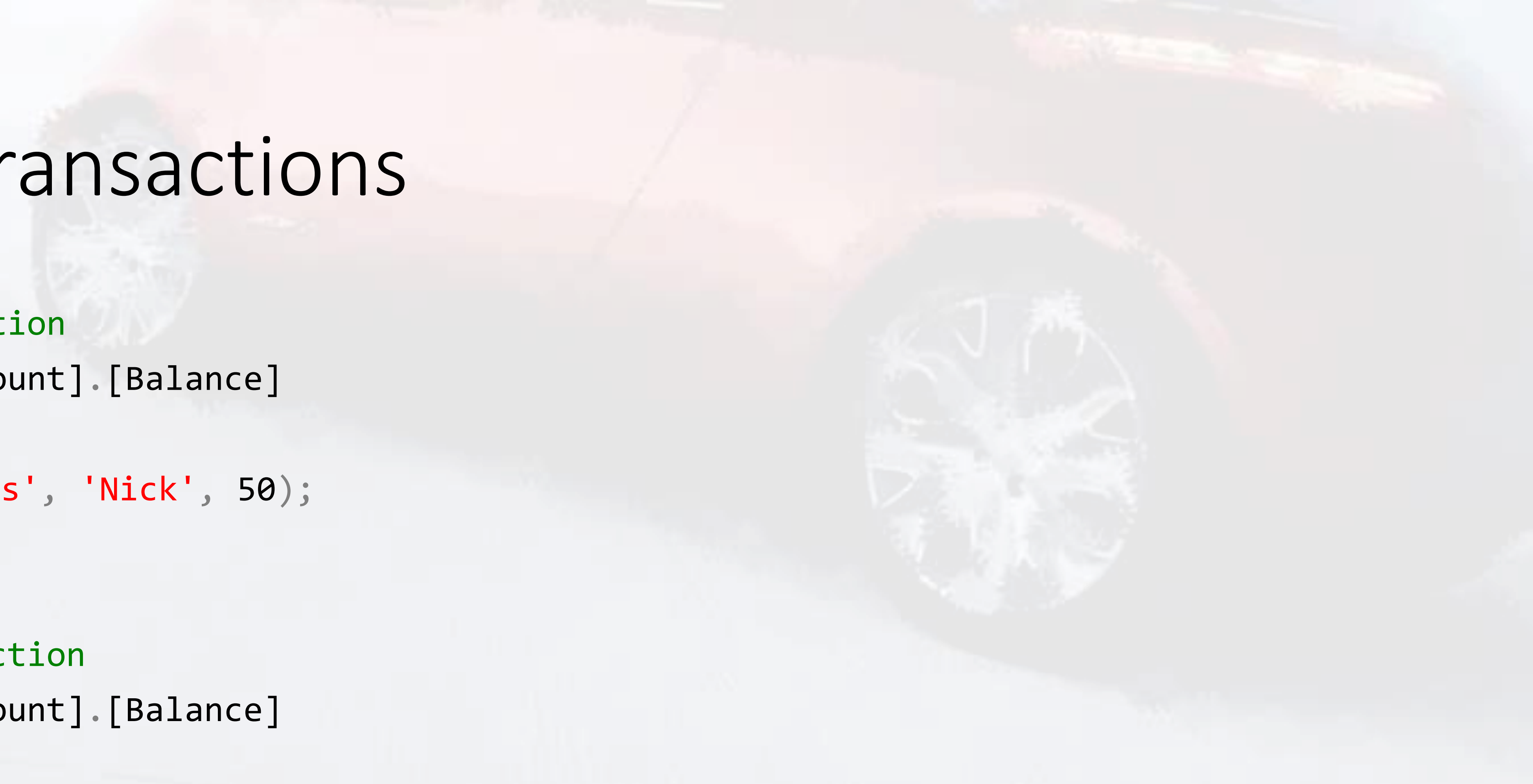
Demo

Ledger Tables

Updatable Ledger Table

```
CREATE SCHEMA [Account];
GO
CREATE TABLE [Account].[Balance]
(
        [CustomerID]   int            NOT NULL PRIMARY KEY CLUSTERED
,       [LastName]     varchar(50)    NOT NULL
,       [FirstName]    varchar(50)    NOT NULL
,       [Balance]      decimal(10,2)  NOT NULL
)
WITH (
        SYSTEM_VERSIONING = ON --(HISTORY_TABLE = [Account].[BalanceHistory])
,       LEDGER = ON --(LEDGER_VIEW = [Account].[BalanceLedgerView])
);
GO
```

INSERT transactions

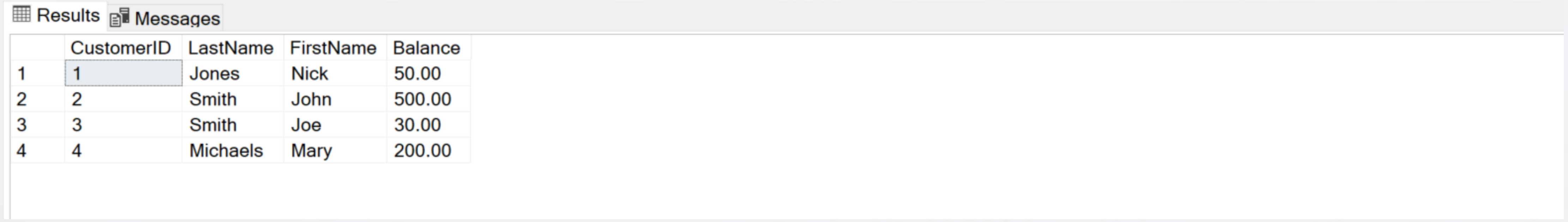


```sql
-- First transaction
INSERT INTO [Account].[Balance]
VALUES
        (1, 'Jones', 'Nick', 50);
GO

-- Second transaction
INSERT INTO [Account].[Balance]
VALUES
        (2, 'Smith', 'John', 500)
,       (3, 'Smith', 'Joe', 30)
,       (4, 'Michaels', 'Mary', 200);
GO
```

Selecting data

```
-- By default, columns with information relating to transactions are
-- not returned (provides transparency to applications)
SELECT *
FROM [Account].[Balance];
GO
```

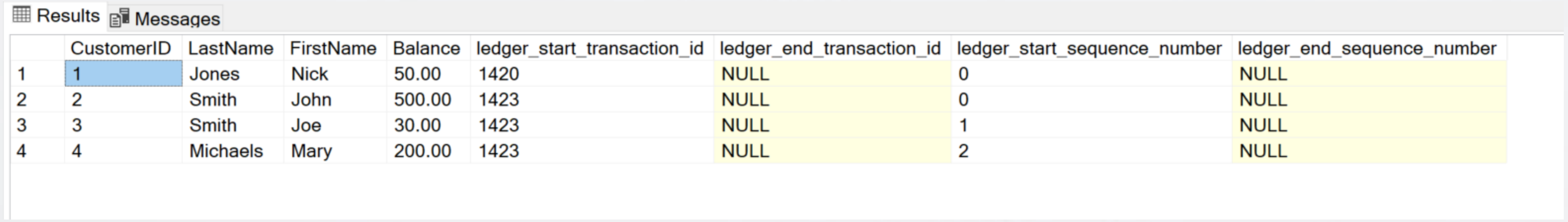


Results | Messages

	CustomerID	LastName	FirstName	Balance
1	1	Jones	Nick	50.00
2	2	Smith	John	500.00
3	3	Smith	Joe	30.00
4	4	Michaels	Mary	200.00

Selecting additional metadata

```
-- Metadata columns must be explicitly selected
SELECT *
,       [ledger_start_transaction_id]
,       [ledger_end_transaction_id]
,       [ledger_start_sequence_number]
,       [ledger_end_sequence_number]
FROM [Account].[Balance];
GO
```



Results Messages

	CustomerID	LastName	FirstName	Balance	ledger_start_transaction_id	ledger_end_transaction_id	ledger_start_sequence_number	ledger_end_sequence_number
1	1	Jones	Nick	50.00	1420	NULL	0	NULL
2	2	Smith	John	500.00	1423	NULL	0	NULL
3	3	Smith	Joe	30.00	1423	NULL	1	NULL
4	4	Michaels	Mary	200.00	1423	NULL	2	NULL

Updating data

```
UPDATE[Account].[Balance]
SET[Balance] = 100
WHERE[CustomerID] = 1;
GO
```

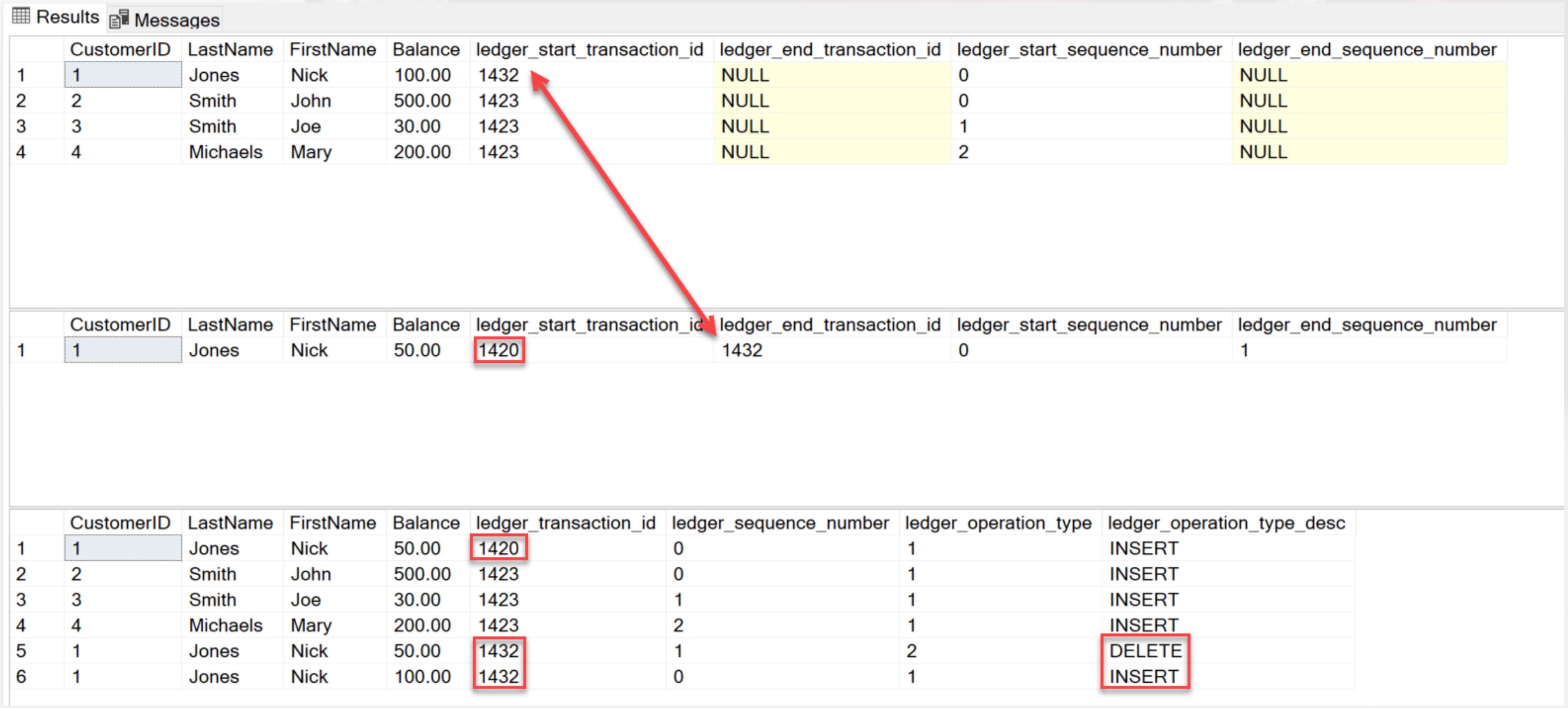
Query data & ledger metadata after updates

```
-- We query the updateable table, the history table and the ledger view
SELECT *
,[ledger_start_transaction_id]
,[ledger_end_transaction_id]
,[ledger_start_sequence_number]
,[ledger_end_sequence_number]
FROM [Account].[Balance];

SELECT * FROM [Account].[MSSQL_LedgerHistoryFor_1525580473];

SELECT * FROM [Account].[Balance_Ledger] ORDER BY [ledger_transaction_id];
GO
```

Data & ledger metadata after updates



Results Messages

	CustomerID	LastName	FirstName	Balance	ledger_start_transaction_id	ledger_end_transaction_id	ledger_start_sequence_number	ledger_end_sequence_number
1	1	Jones	Nick	100.00	1432	NULL	0	NULL
2	2	Smith	John	500.00	1423	NULL	0	NULL
3	3	Smith	Joe	30.00	1423	NULL	1	NULL
4	4	Michaels	Mary	200.00	1423	NULL	2	NULL

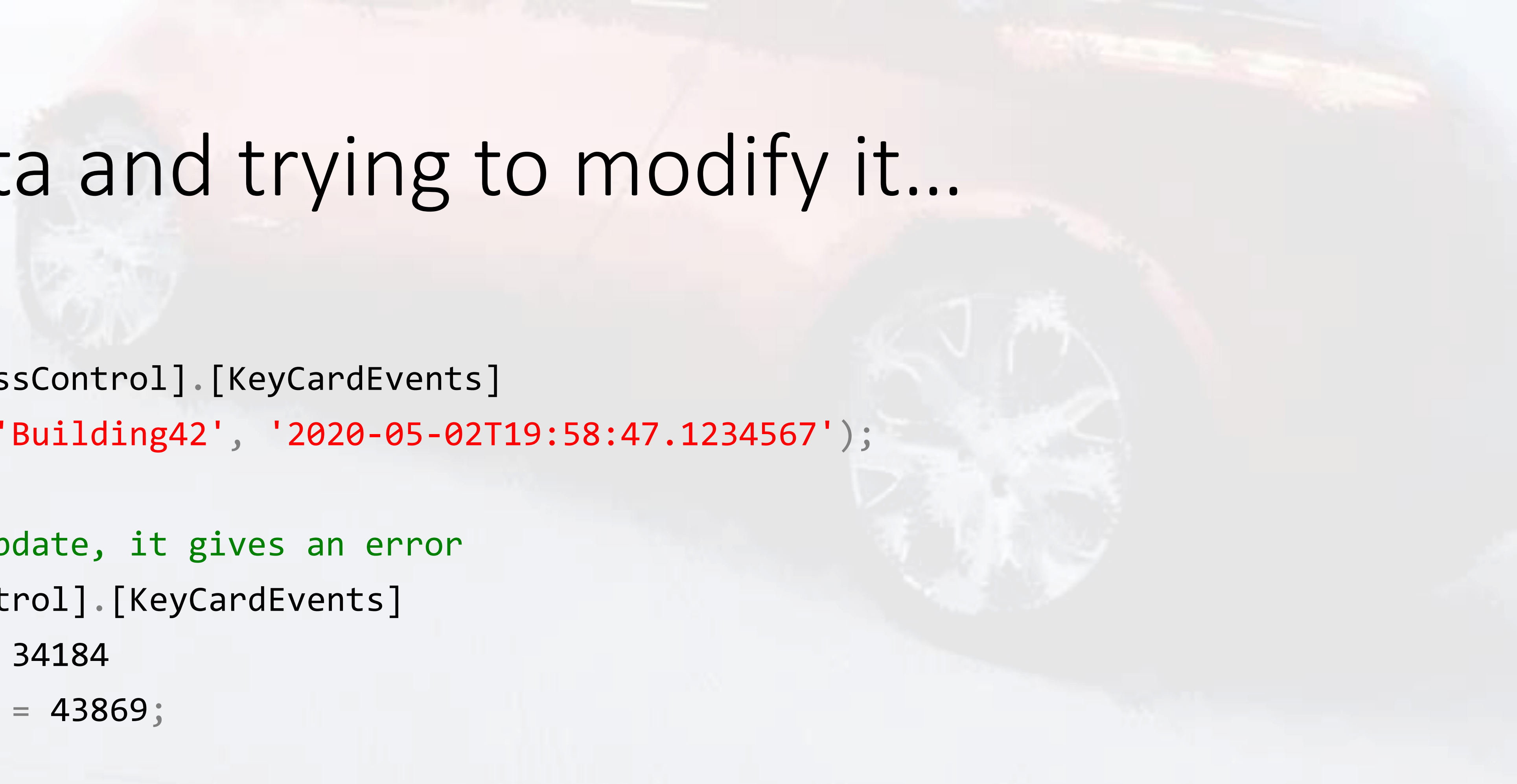
	CustomerID	LastName	FirstName	Balance	ledger_start_transaction_id	ledger_end_transaction_id	ledger_start_sequence_number	ledger_end_sequence_number
1	1	Jones	Nick	50.00	1420	1432	0	1

	CustomerID	LastName	FirstName	Balance	ledger_transaction_id	ledger_sequence_number	ledger_operation_type	ledger_operation_type_desc
1	1	Jones	Nick	50.00	1420	0	1	INSERT
2	2	Smith	John	500.00	1423	0	1	INSERT
3	3	Smith	Joe	30.00	1423	1	1	INSERT
4	4	Michaels	Mary	200.00	1423	2	1	INSERT
5	1	Jones	Nick	50.00	1432	1	2	DELETE
6	1	Jones	Nick	100.00	1432	0	1	INSERT

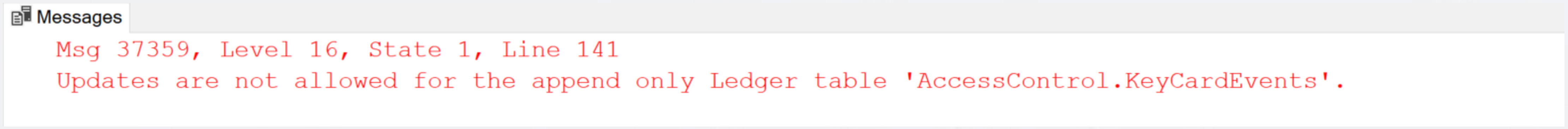
Append-only Ledger Table

```
CREATE SCHEMA [AccessControl];
GO
CREATE TABLE [AccessControl].[KeyCardEvents]
(
       [EmployeeID]                  int           NOT NULL PRIMARY KEY CLUSTERED
,      [AccessOperationDescription] nvarchar(MAX) NOT NULL
,      [Timestamp]                   datetime2     NOT NULL
)
WITH (
       LEDGER = ON (APPEND_ONLY = ON)
);
GO
```

Insert data and trying to modify it…



```sql
-- Insert a row
INSERT INTO [AccessControl].[KeyCardEvents]
VALUES ('43869', 'Building42', '2020-05-02T19:58:47.1234567');
GO
-- If we try to update, it gives an error
UPDATE [AccessControl].[KeyCardEvents]
SET[EmployeeID] = 34184
WHERE[EmployeeID] = 43869;
GO
```

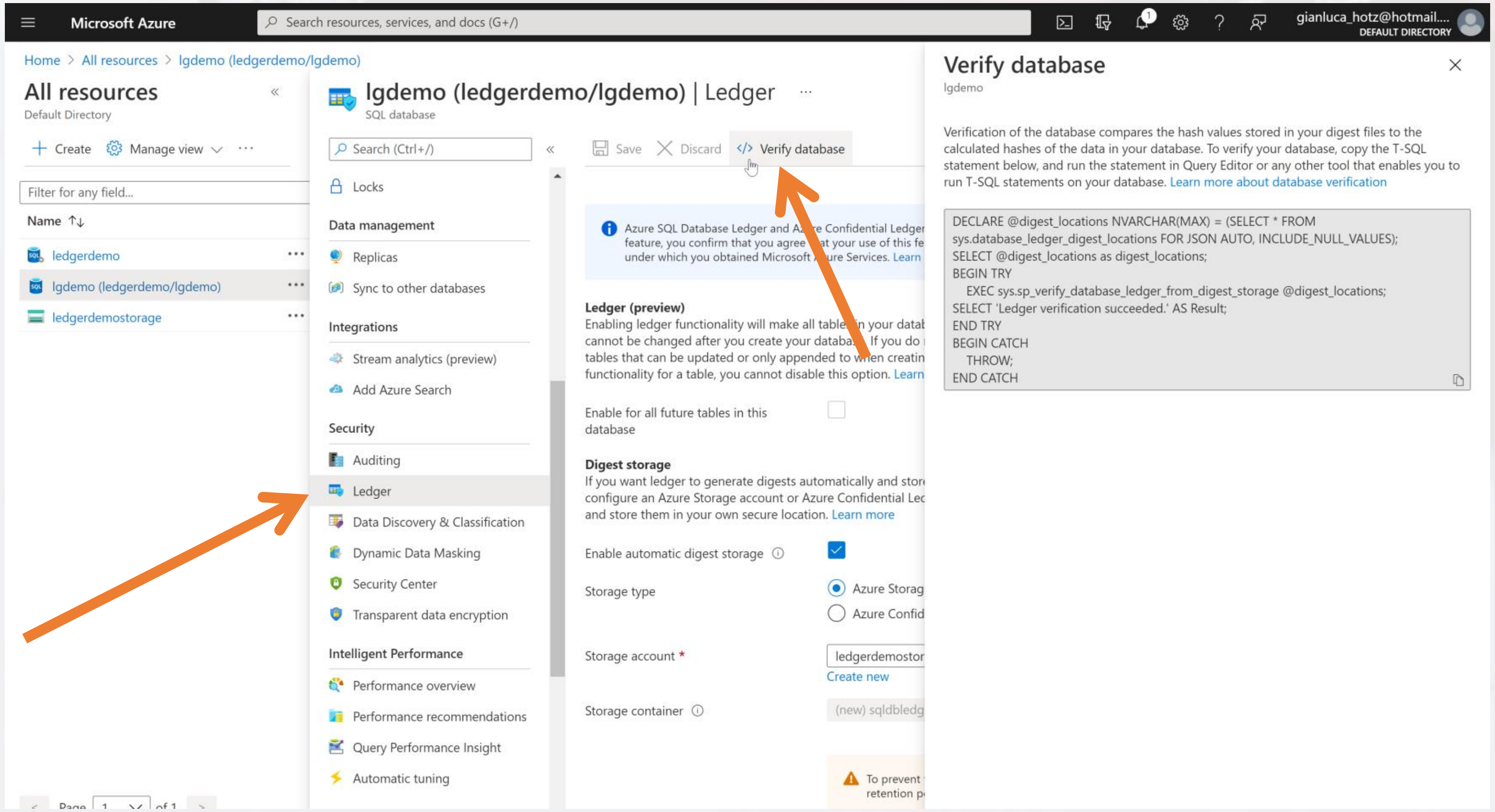


```
Msg 37359, Level 16, State 1, Line 141
Updates are not allowed for the append only Ledger table 'AccessControl.KeyCardEvents'.
```

Demo

Verify database

Verify Database



Azure SQL Database Ledger and Azure Confidential Ledger feature, you confirm that you agree that your use of this fe under which you obtained Microsoft Azure Services. Learn

Ledger (preview)
Enabling ledger functionality will make all tables in your datab cannot be changed after you create your database. If you do tables that can be updated or only appended to when creatin functionality for a table, you cannot disable this option. Learn

Enable for all future tables in this database ☐

Digest storage
If you want ledger to generate digests automatically and store configure an Azure Storage account or Azure Confidential Led and store them in your own secure location. Learn more

Enable automatic digest storage ✓

Storage type ● Azure Storag ○ Azure Confid

Storage account * ledgerdemostor
Create new

Storage container (new) sqldbledg

To prevent retention p

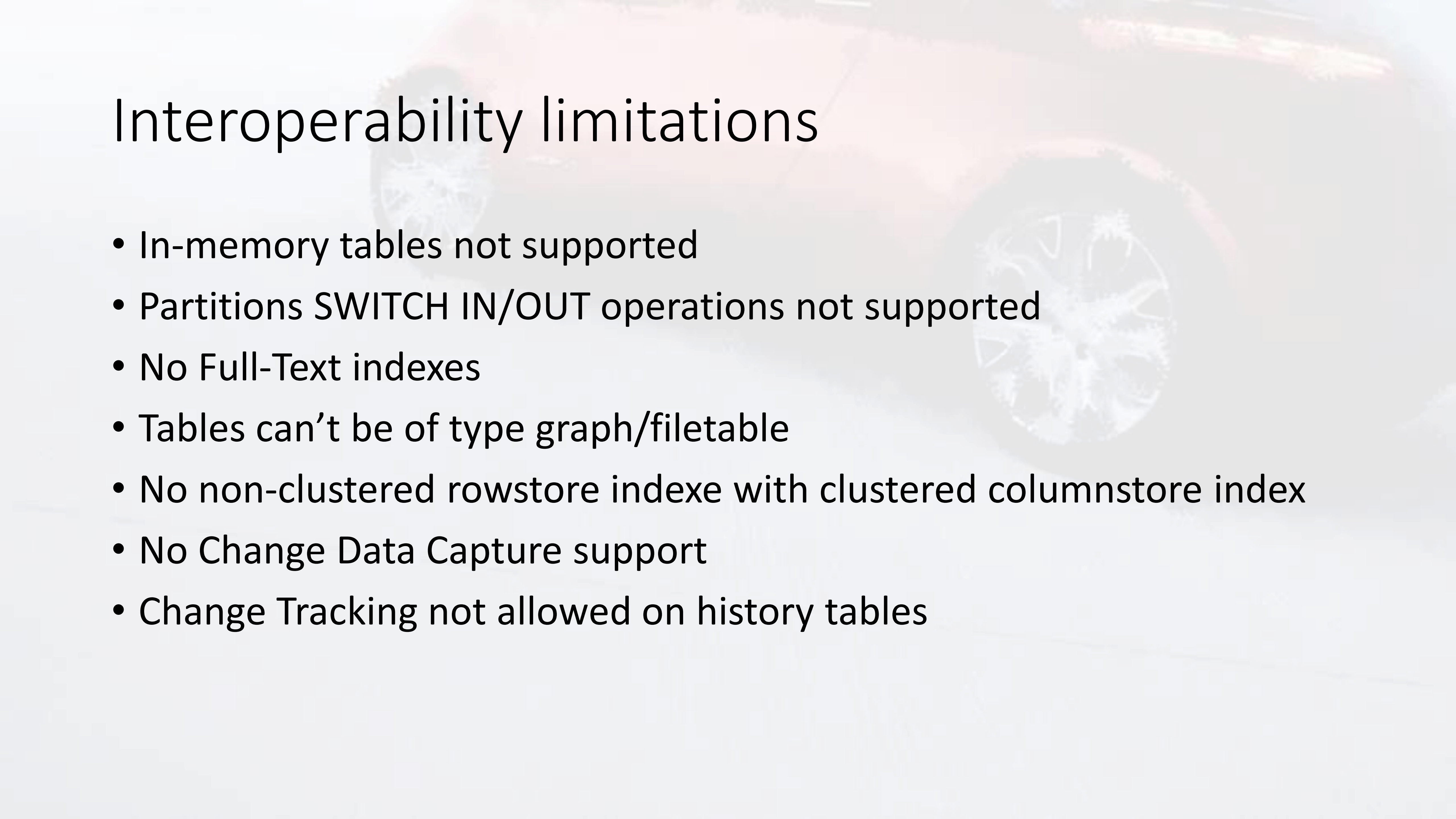
Verify database
lgdemo

Verification of the database compares the hash values stored in your digest files to the calculated hashes of the data in your database. To verify your database, copy the T-SQL statement below, and run the statement in Query Editor or any other tool that enables you to run T-SQL statements on your database. Learn more about database verification

```
DECLARE @digest_locations NVARCHAR(MAX) = (SELECT * FROM
sys.database_ledger_digest_locations FOR JSON AUTO, INCLUDE_NULL_VALUES);
SELECT @digest_locations as digest_locations;
BEGIN TRY
    EXEC sys.sp_verify_database_ledger_from_digest_storage @digest_locations;
SELECT 'Ledger verification succeeded.' AS Result;
END TRY
BEGIN CATCH
    THROW;
END CATCH
```

General limitations

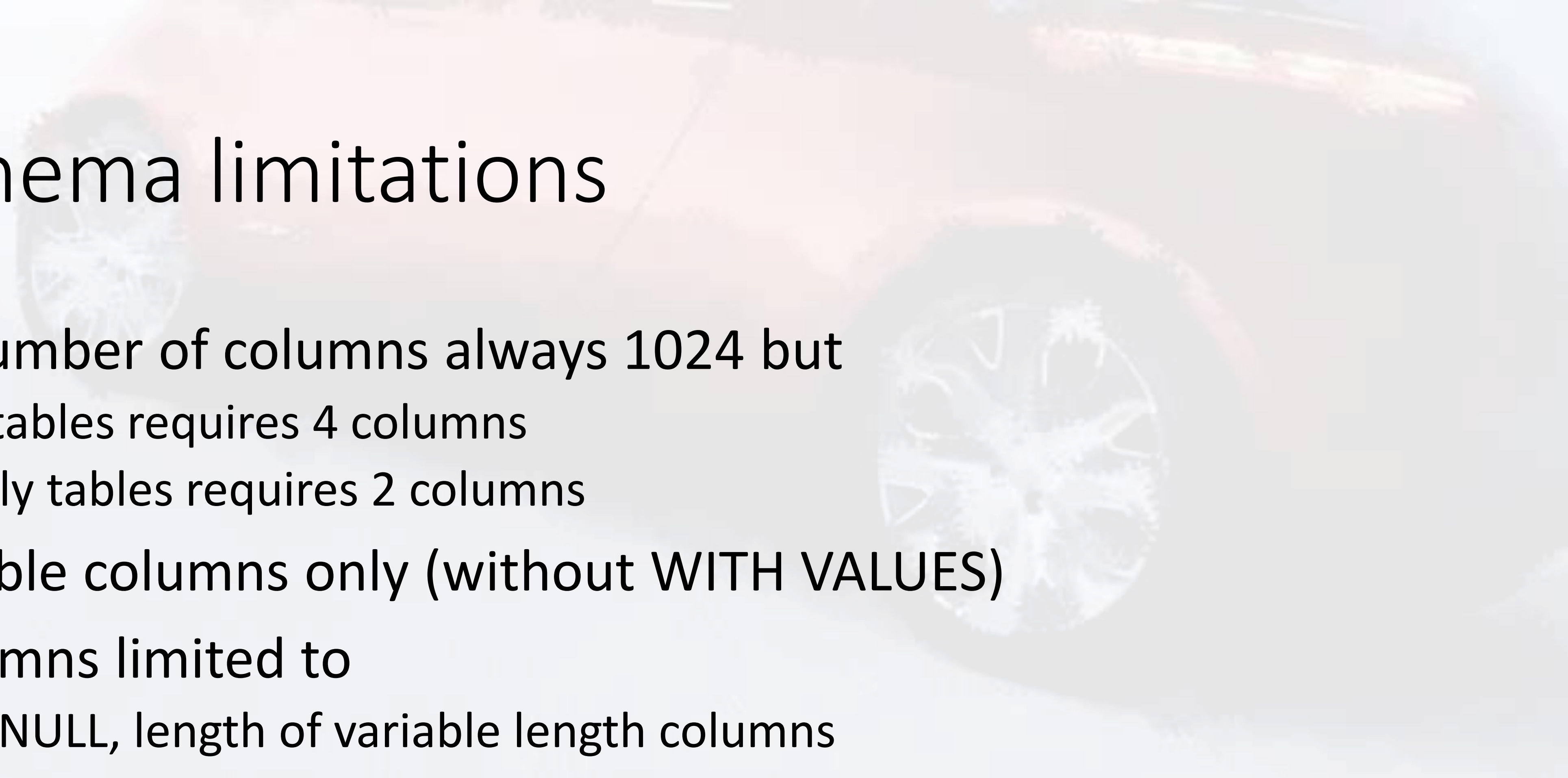
- Database level option forcing Ledger Tables cannot be disabled
- No conversion of existing tables (both ways)
  - Migrate with sys.sp_copy_data_in_batches system procedure
- No dropping of tables/columns
  - renamed/hidden but remain available for database verification
- Deleting older data from history tables forbidden
- Transaction can only update (!) 200 tables
- Updatable ledger tables are based on the technology of temporal tables and inherits most of the limitations



Interoperability limitations

- In-memory tables not supported
- Partitions SWITCH IN/OUT operations not supported
- No Full-Text indexes
- Tables can’t be of type graph/filetable
- No non-clustered rowstore indexe with clustered columnstore index
- No Change Data Capture support
- Change Tracking not allowed on history tables

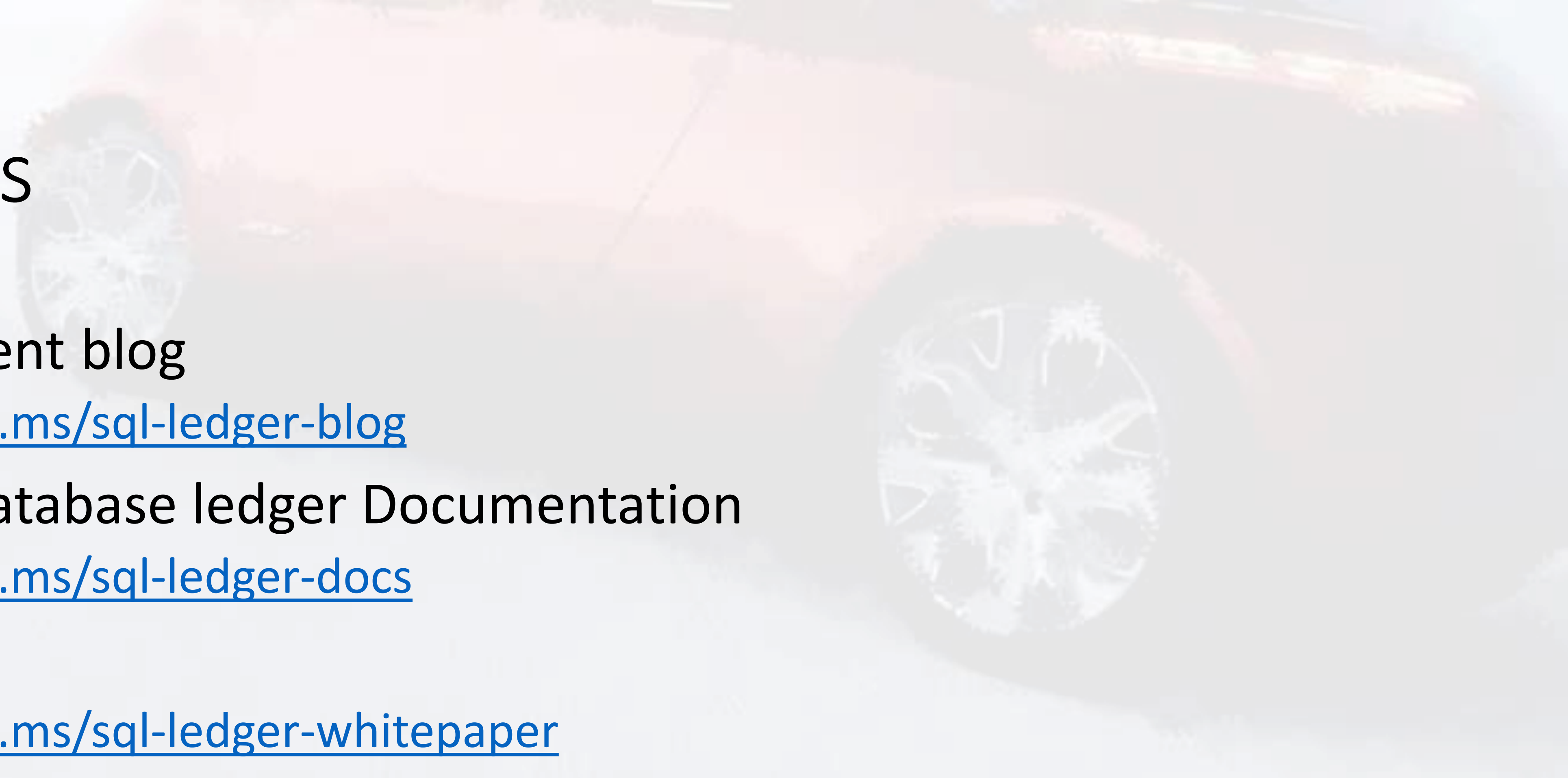
Other schema limitations



- Maximum number of columns always 1024 but
  - updatable tables requires 4 columns
  - Append-only tables requires 2 columns
- Adding nullable columns only (without WITH VALUES)
- Altering columns limited to
  - NULL/NOT NULL, length of variable length columns
  - Collation for Unicode strings
- XML, FILESTREAM, SqlVariant & user-defined types not supported
- Sparse Column Set not supported

Use cases for Ledger Tables

- In general: those who only need Forward Integrity
  - System reliable processing transactions and protected against future tampering
- Some examples
  - Streamlining audits
    - Provides data integrity cryptographic proof to auditors (internal or external)
  - Multiple-party business processes
    - Alternative to Blockchain for intrinsically centralized systems, "trust, but verify" perspective
  - Trusted off-chain storage for Blockchain
- Choosing an Azure ledger technology
  - https://techcommunity.microsoft.com/t5/azure-sql/choosing-an-azure-ledger-technology/ba-p/2450502



Resources

- Announcement blog
  - https://aka.ms/sql-ledger-blog
- Azure SQL Database ledger Documentation
  - https://aka.ms/sql-ledger-docs
- Whitepaper
  - https://aka.ms/sql-ledger-whitepaper

Thanks!

Q&A