Visual Studio LIVE!
EXPERT SOLUTIONS FOR ENTERPRISE DEVELOPERS
Las Vegas

What's New for Developers in SQL Server 2022?

Leonard Lobel
Chief Technology Officer
Sleek Technologies

Level:
Intermediate

March 23, 2023
8:00am – 9:15 am

Your Code Powers the World.
Our Training Powers You.

#VSLIVE

About Me

sleek technologies
EY
MVP Microsoft® Most Valuable Professional
pluralsight
Microsoft CERTIFIED AZURE COSMOS DB DEVELOPER SPECIALTY

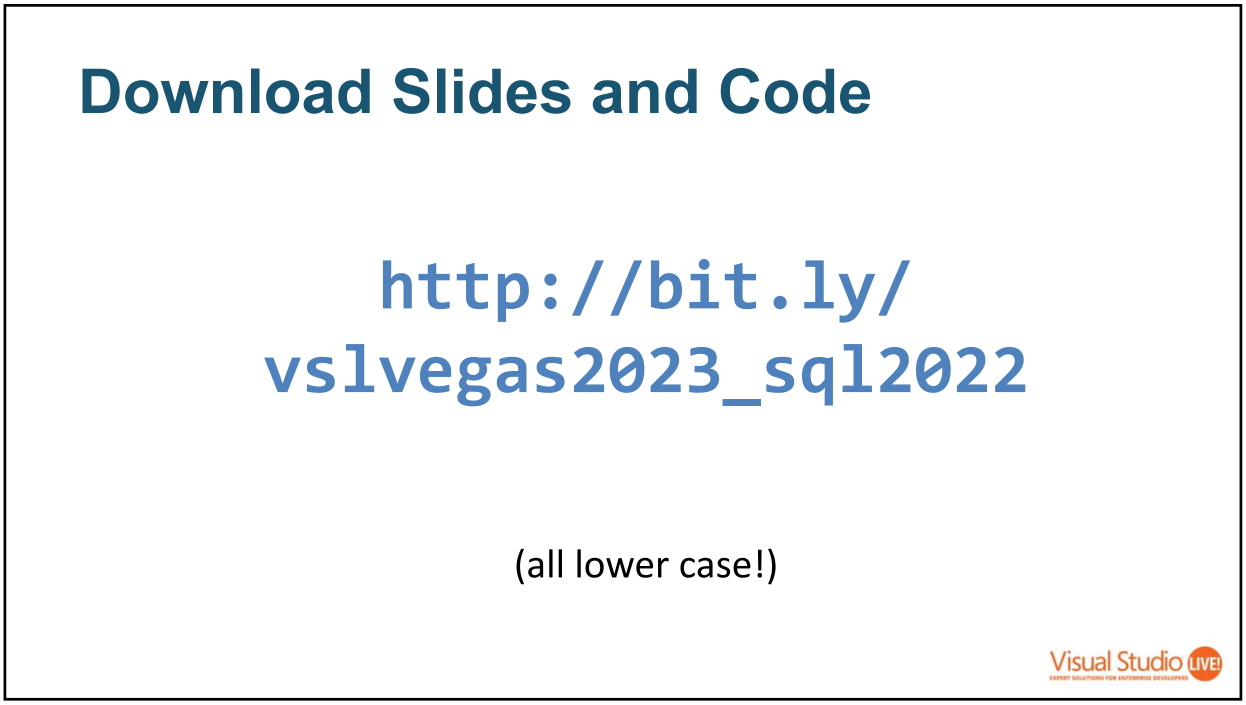
Leonard Lobel

- CTO & Co-Founder
  - Sleek Technologies, Inc.
- Microsoft Consultant
  - Ernst & Young
- Microsoft MVP
  - Data Platform
- Trainer/Speaker/Author
- Programming since 1979

Contact

- Email
  - lenni.lobel@sleektech.com
- Blog
  - lennilobel.wordpress.com
- Twitter
  - @lennilobel

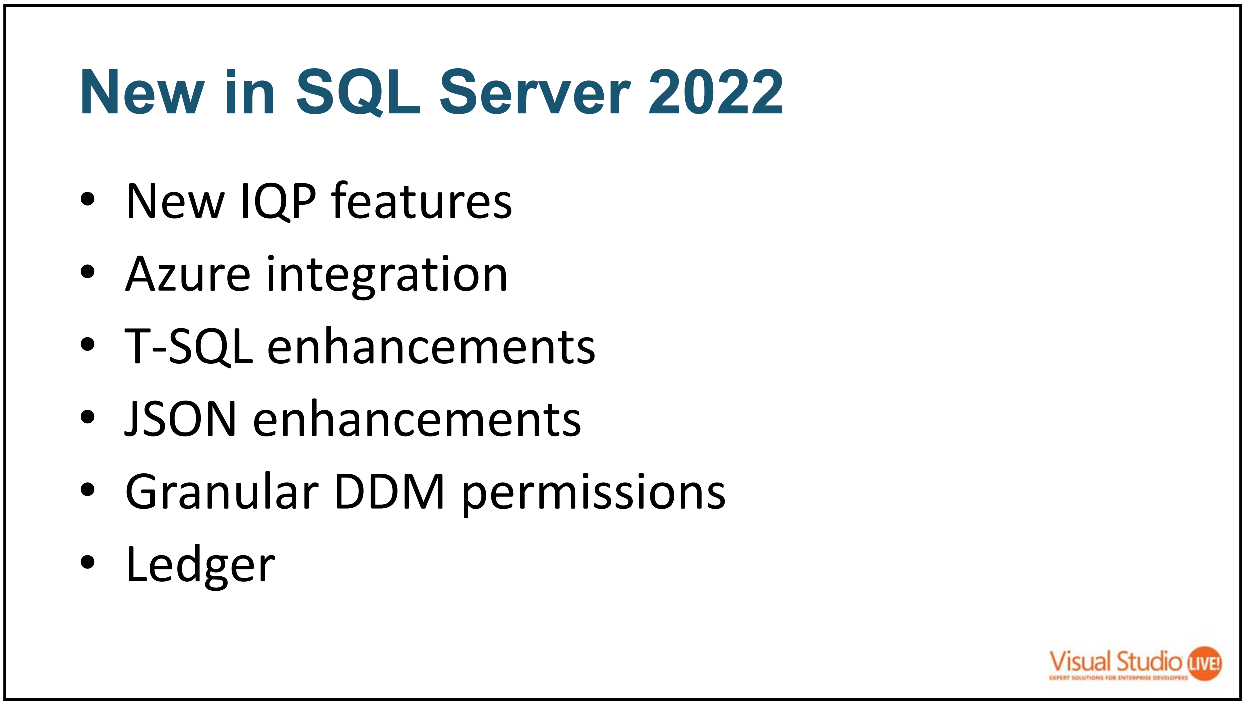
Programming Microsoft SQL Server 2008
Programming Microsoft SQL Server 2012
Developer //Step by step Microsoft Azure SQL Database



Download Slides and Code

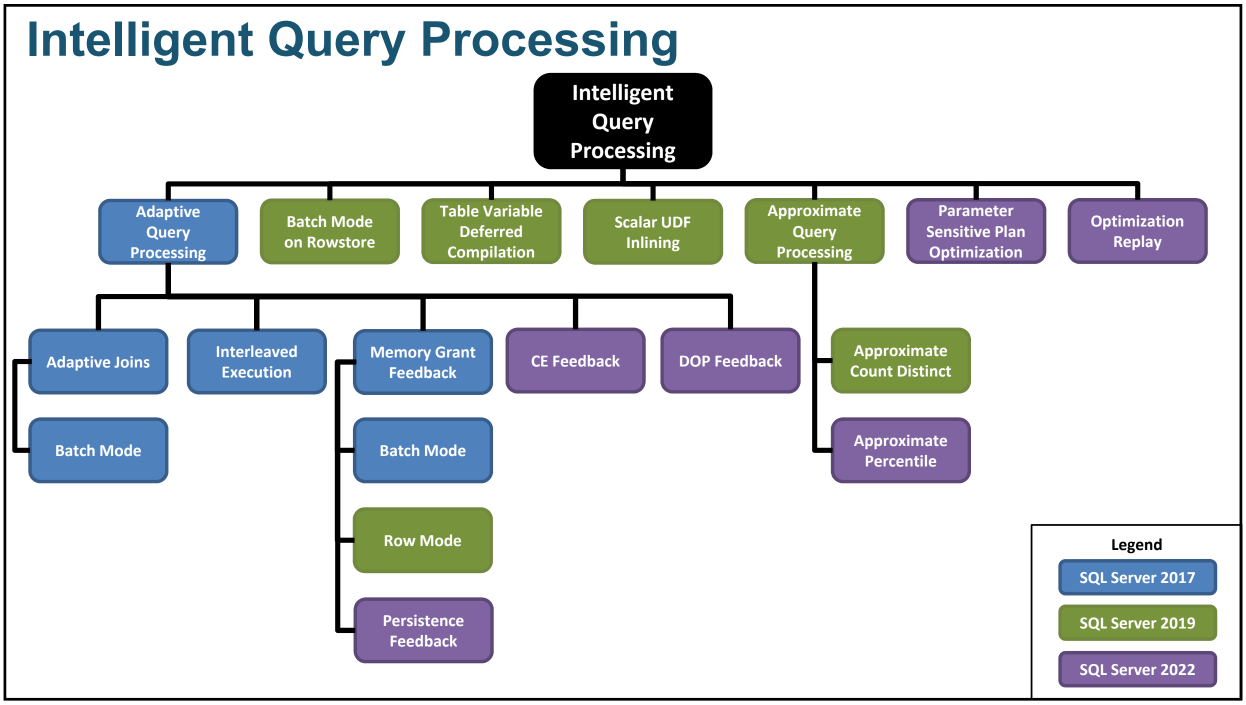
`http://bit.ly/vslvegas2023_sql2022`

(all lower case!)

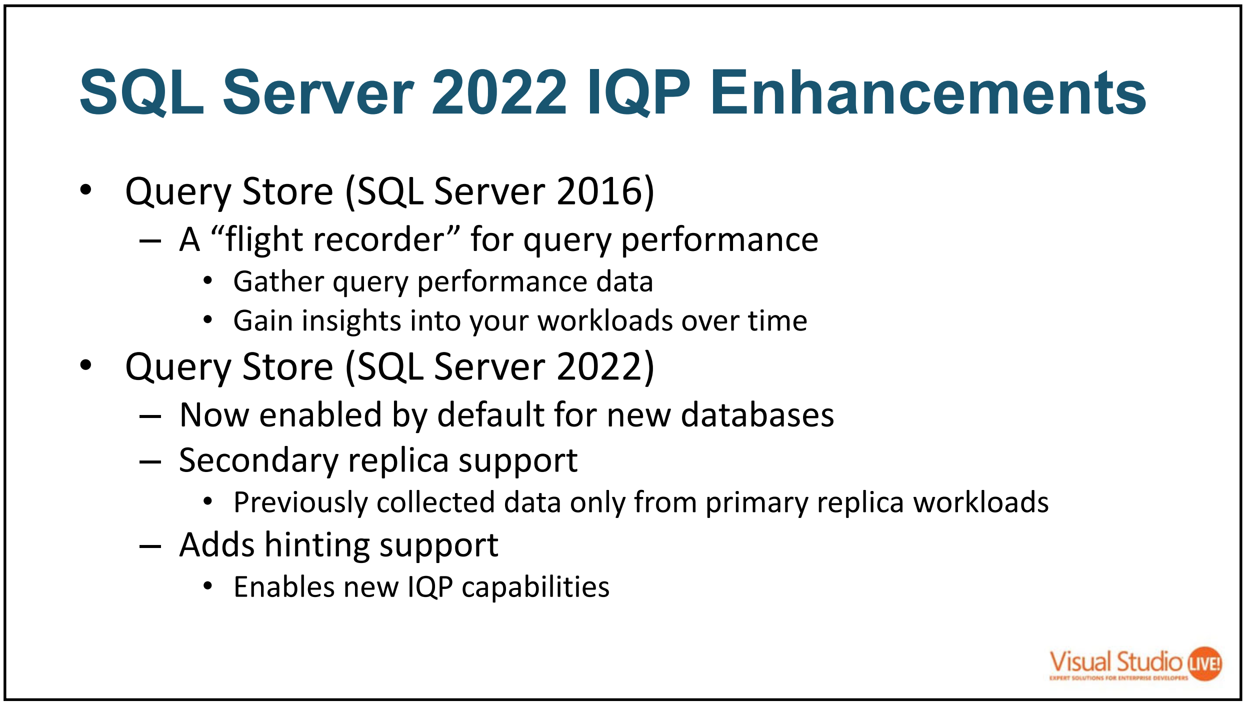


New in SQL Server 2022

- New IQP features
- Azure integration
- T-SQL enhancements
- JSON enhancements
- Granular DDM permissions
- Ledger

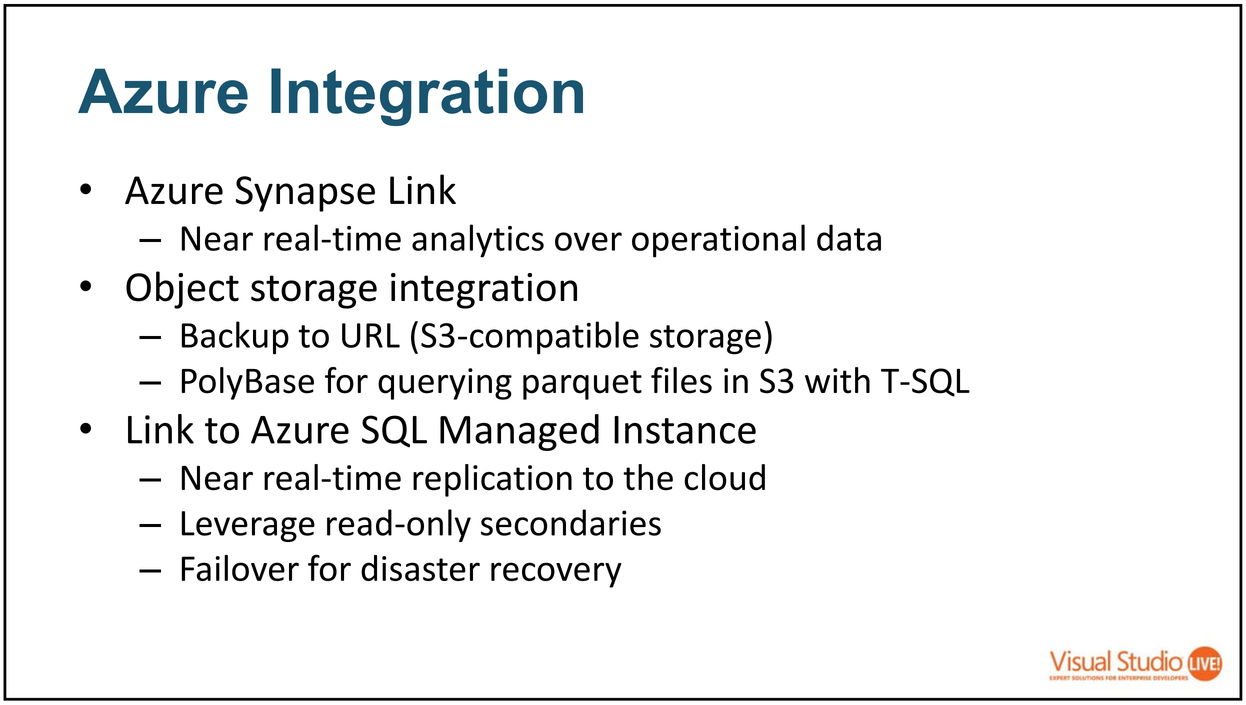


Intelligent Query Processing



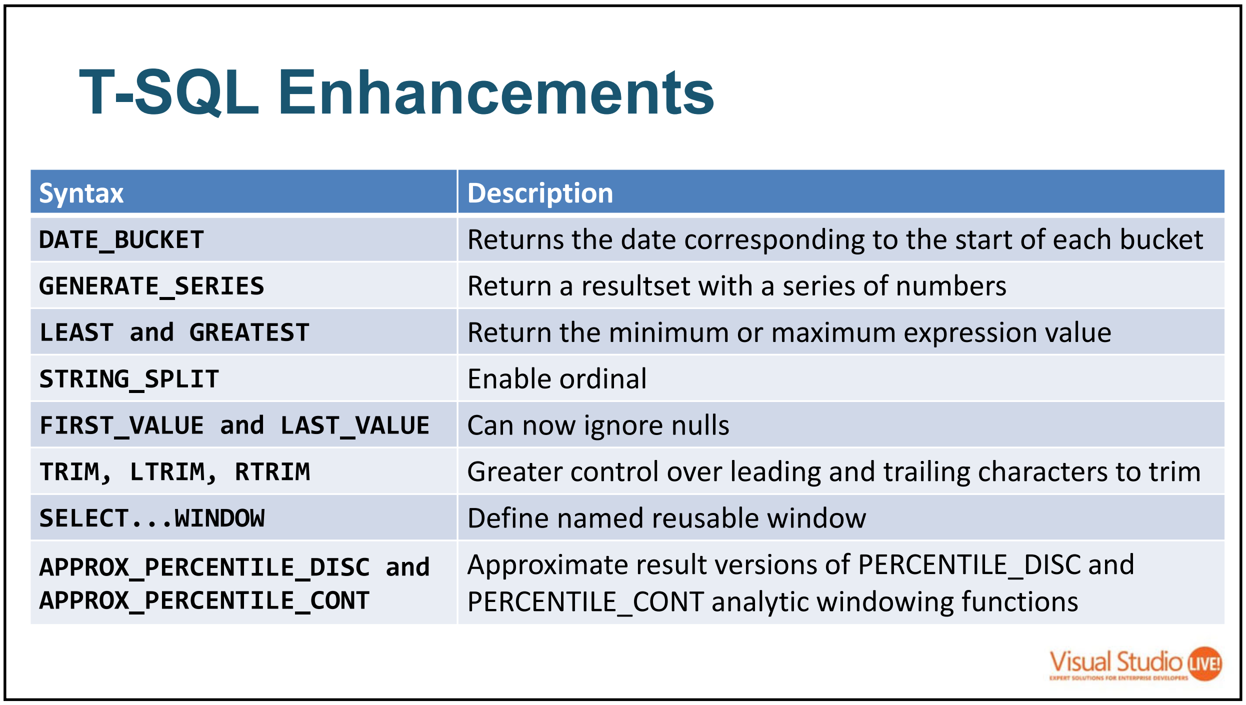
SQL Server 2022 IQP Enhancements

- Query Store (SQL Server 2016)
  - A “flight recorder” for query performance
    - Gather query performance data
    - Gain insights into your workloads over time
- Query Store (SQL Server 2022)
  - Now enabled by default for new databases
  - Secondary replica support
    - Previously collected data only from primary replica workloads
  - Adds hinting support
    - Enables new IQP capabilities



Azure Integration

- Azure Synapse Link
  - Near real-time analytics over operational data
- Object storage integration
  - Backup to URL (S3-compatible storage)
  - PolyBase for querying parquet files in S3 with T-SQL
- Link to Azure SQL Managed Instance
  - Near real-time replication to the cloud
  - Leverage read-only secondaries
  - Failover for disaster recovery



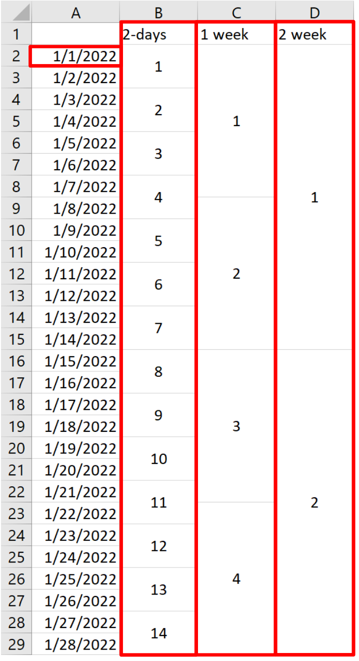
T-SQL Enhancements

Syntax	Description
`DATE_BUCKET`	Returns the date corresponding to the start of each bucket
`GENERATE_SERIES`	Return a resultset with a series of numbers
`LEAST and GREATEST`	Return the minimum or maximum expression value
`STRING_SPLIT`	Enable ordinal
`FIRST_VALUE and LAST_VALUE`	Can now ignore nulls
`TRIM, LTRIM, RTRIM`	Greater control over leading and trailing characters to trim
`SELECT...WINDOW`	Define named reusable window

`APPROX_PERCENTILE_DISC and`
`APPROX_PERCENTILE_CONT`
	Approximate result versions of PERCENTILE_DISC and PERCENTILE_CONT analytic windowing functions

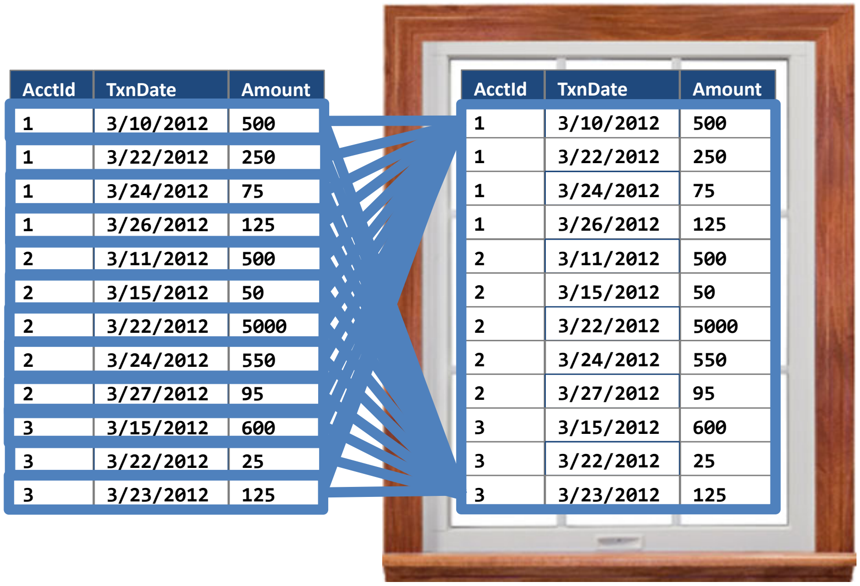
DATE_BUCKET

```
DATE_BUCKET(DAY, 2, DateColumn, '2022-01-01')
DATE_BUCKET(WEEK, 1, DateColumn, '2022-01-01')
DATE_BUCKET(WEEK, 2, DateColumn, '2022-01-01')
```



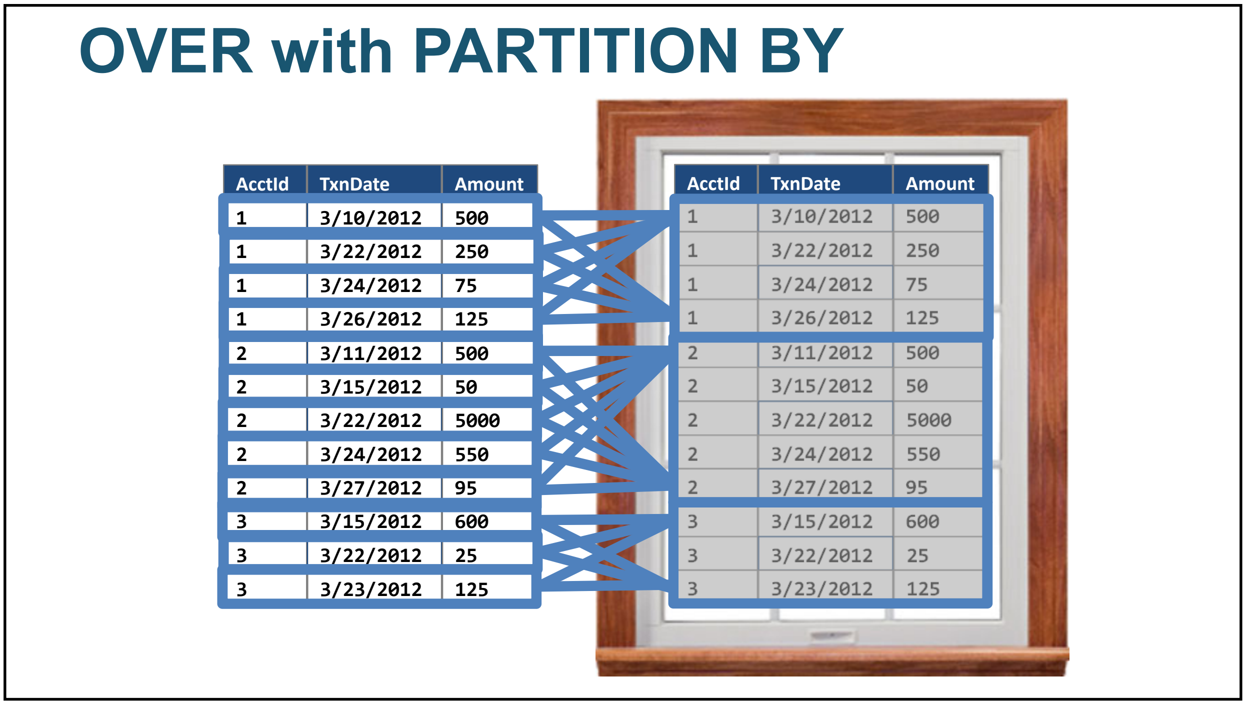
	A	B	C	D
1		2-days	1 week	2 week
2	1/1/2022	1	1	1
3	1/2/2022			
4	1/3/2022	2		
5	1/4/2022			
6	1/5/2022	3		
7	1/6/2022			
8	1/7/2022	4		
9	1/8/2022		2	
10	1/9/2022	5		
11	1/10/2022			
12	1/11/2022	6		
13	1/12/2022			
14	1/13/2022	7		
15	1/14/2022			
16	1/15/2022	8	3	2
17	1/16/2022			
18	1/17/2022	9		
19	1/18/2022			
20	1/19/2022	10		
21	1/20/2022			
22	1/21/2022	11		
23	1/22/2022		4	
24	1/23/2022	12		
25	1/24/2022			
26	1/25/2022	13		
27	1/26/2022			
28	1/27/2022	14		
29	1/28/2022			

OVER



AcctId	TxnDate	Amount
1	3/10/2012	500
1	3/22/2012	250
1	3/24/2012	75
1	3/26/2012	125
2	3/11/2012	500
2	3/15/2012	50
2	3/22/2012	5000
2	3/24/2012	550
2	3/27/2012	95
3	3/15/2012	600
3	3/22/2012	25
3	3/23/2012	125

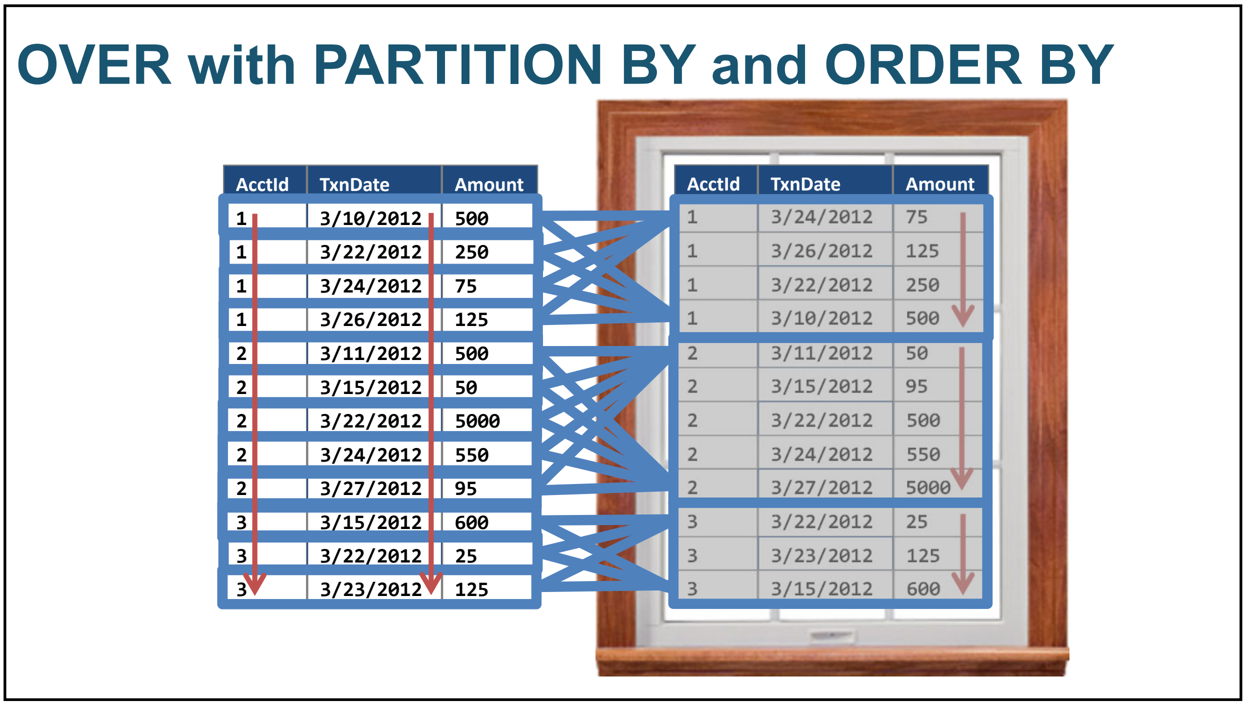
AcctId	TxnDate	Amount
1	3/10/2012	500
1	3/22/2012	250
1	3/24/2012	75
1	3/26/2012	125
2	3/11/2012	500
2	3/15/2012	50
2	3/22/2012	5000
2	3/24/2012	550
2	3/27/2012	95
3	3/15/2012	600
3	3/22/2012	25
3	3/23/2012	125



OVER with PARTITION BY

AcctId	TxnDate	Amount
1	3/10/2012	500
1	3/22/2012	250
1	3/24/2012	75
1	3/26/2012	125
2	3/11/2012	500
2	3/15/2012	50
2	3/22/2012	5000
2	3/24/2012	550
2	3/27/2012	95
3	3/15/2012	600
3	3/22/2012	25
3	3/23/2012	125

AcctId	TxnDate	Amount
1	3/10/2012	500
1	3/22/2012	250
1	3/24/2012	75
1	3/26/2012	125
2	3/11/2012	500
2	3/15/2012	50
2	3/22/2012	5000
2	3/24/2012	550
2	3/27/2012	95
3	3/15/2012	600
3	3/22/2012	25
3	3/23/2012	125



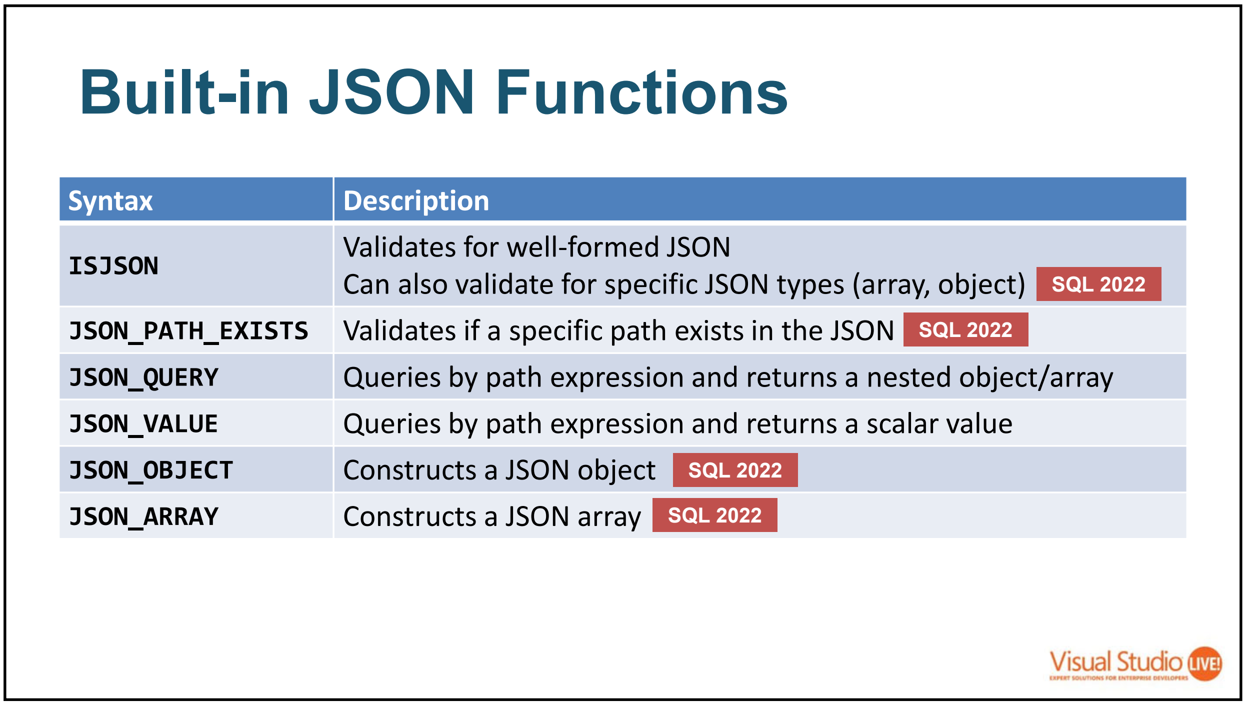
OVER with PARTITION BY and ORDER BY

AcctId	TxnDate	Amount
1	3/10/2012	500
1	3/22/2012	250
1	3/24/2012	75
1	3/26/2012	125
2	3/11/2012	500
2	3/15/2012	50
2	3/22/2012	5000
2	3/24/2012	550
2	3/27/2012	95
3	3/15/2012	600
3	3/22/2012	25
3	3/23/2012	125

AcctId	TxnDate	Amount
1	3/24/2012	75
1	3/26/2012	125
1	3/22/2012	250
1	3/10/2012	500
2	3/11/2012	50
2	3/15/2012	95
2	3/22/2012	500
2	3/24/2012	550
2	3/27/2012	5000
3	3/22/2012	25
3	3/23/2012	125
3	3/15/2012	600

T-SQL Enhancements

demo

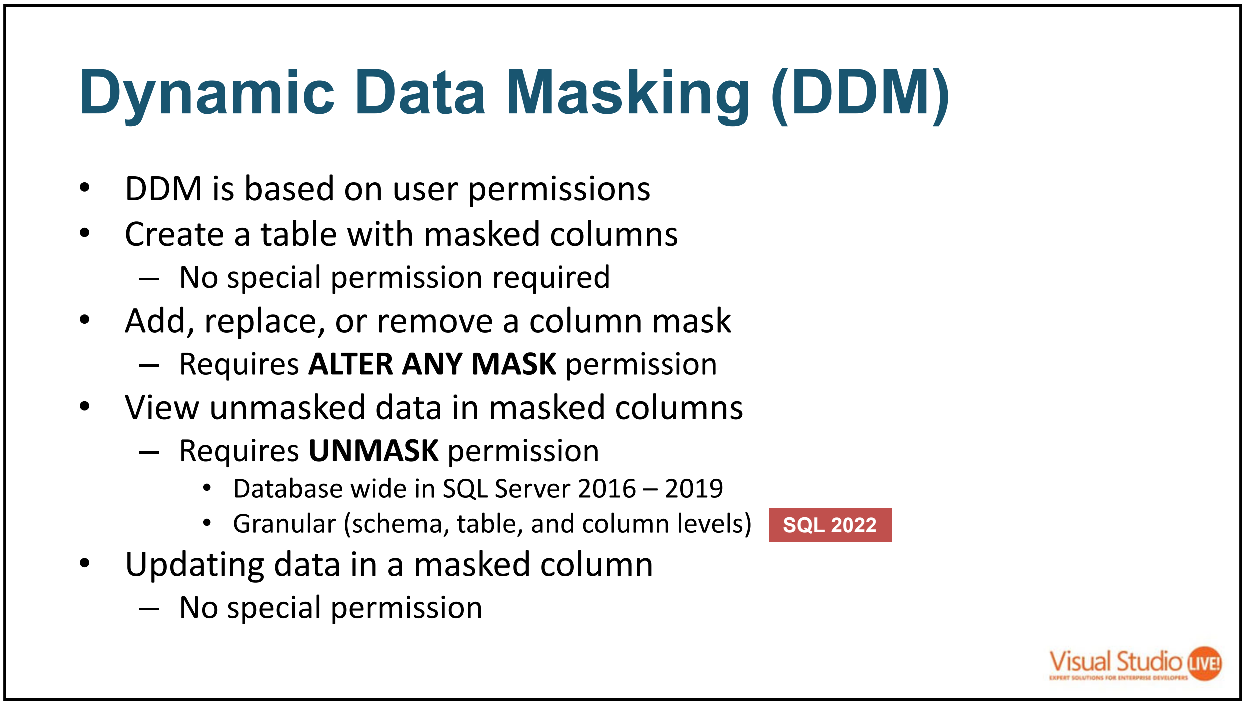


Built-in JSON Functions

Syntax	Description
`ISJSON`	Validates for well-formed JSON Can also validate for specific JSON types (array, object) SQL 2022
`JSON_PATH_EXISTS`	Validates if a specific path exists in the JSON SQL 2022
`JSON_QUERY`	Queries by path expression and returns a nested object/array
`JSON_VALUE`	Queries by path expression and returns a scalar value
`JSON_OBJECT`	Constructs a JSON object SQL 2022
`JSON_ARRAY`	Constructs a JSON array SQL 2022

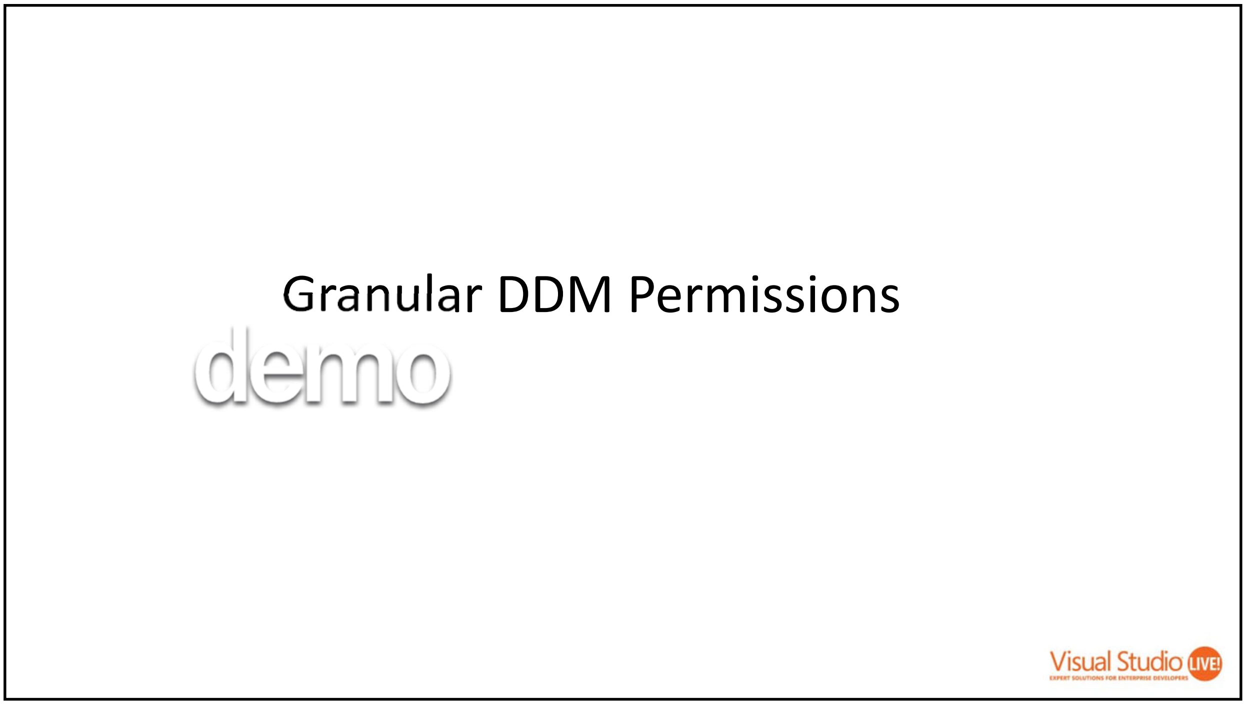
JSON Enhancements

demo



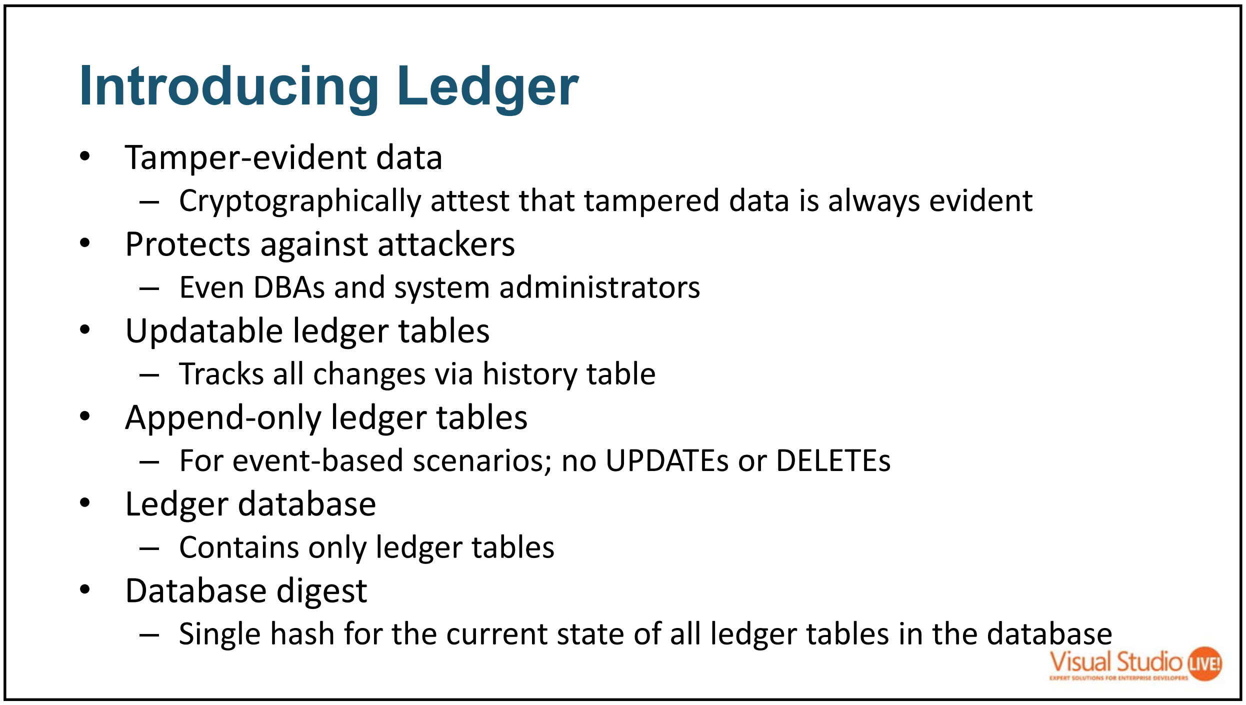
Dynamic Data Masking (DDM)

- DDM is based on user permissions
- Create a table with masked columns
  - No special permission required
- Add, replace, or remove a column mask
  - Requires **ALTER ANY MASK** permission
- View unmasked data in masked columns
  - Requires **UNMASK** permission
    - Database wide in SQL Server 2016 – 2019
    - Granular (schema, table, and column levels) SQL 2022
- Updating data in a masked column
  - No special permission



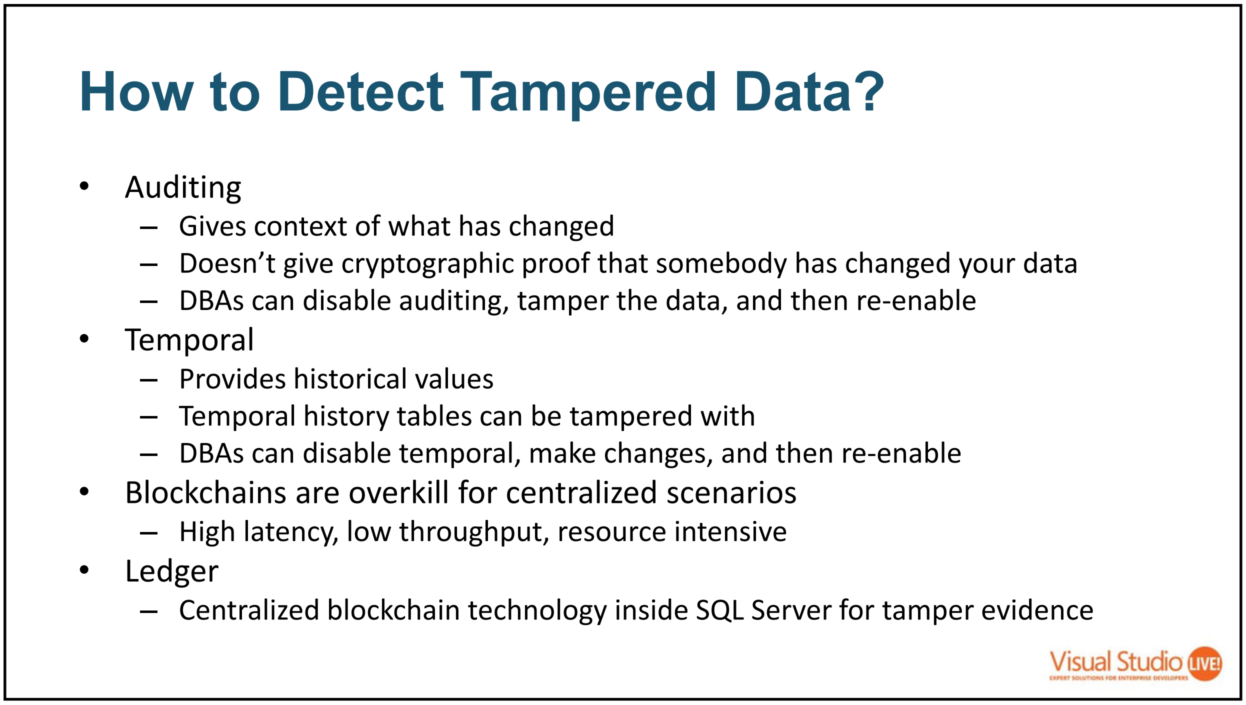
Granular DDM Permissions

demo



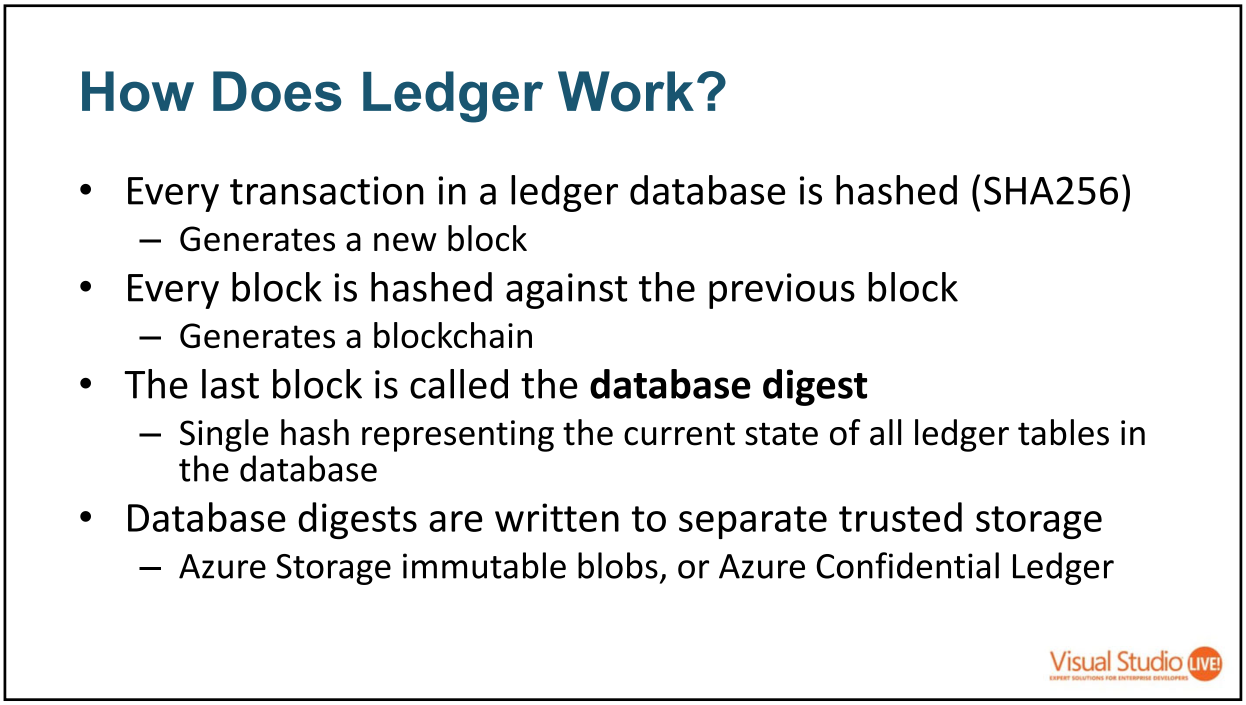
Introducing Ledger

- Tamper-evident data
  - Cryptographically attest that tampered data is always evident
- Protects against attackers
  - Even DBAs and system administrators
- Updatable ledger tables
  - Tracks all changes via history table
- Append-only ledger tables
  - For event-based scenarios; no UPDATEs or DELETEs
- Ledger database
  - Contains only ledger tables
- Database digest
  - Single hash for the current state of all ledger tables in the database



How to Detect Tampered Data?

- Auditing
  - Gives context of what has changed
  - Doesn’t give cryptographic proof that somebody has changed your data
  - DBAs can disable auditing, tamper the data, and then re-enable
- Temporal
  - Provides historical values
  - Temporal history tables can be tampered with
  - DBAs can disable temporal, make changes, and then re-enable
- Blockchains are overkill for centralized scenarios
  - High latency, low throughput, resource intensive
- Ledger
  - Centralized blockchain technology inside SQL Server for tamper evidence



How Does Ledger Work?

- Every transaction in a ledger database is hashed (SHA256)
  - Generates a new block
- Every block is hashed against the previous block
  - Generates a blockchain
- The last block is called the **database digest**
  - Single hash representing the current state of all ledger tables in the database
- Database digests are written to separate trusted storage
  - Azure Storage immutable blobs, or Azure Confidential Ledger

Ledger

demo

Thank You!

- Contact me
  - lenni.lobel@sleektech.com
- Visit my blog
  - lennilobel.wordpress.com
- Follow me on Twitter
  - @lennilobel
- Thanks for coming! ☺