

#FABCONSQLCON2026

FABCON

Microsoft Fabric
COMMUNITY CONFERENCE

SQLCON

Microsoft SQL
COMMUNITY CONFERENCE

ATLANTA MARCH 16 - 20, 2026

Get Two Fabric Certifications for FREE

Attendees of FABCON can take the Fabric Analytics Engineer or Fabric Data Engineer exam for free. Be part of the 2 fastest growing role-based certifications in Microsoft history.

Request your voucher by March 23, 2026.

<https://aka.ms/fabcon/cert100>



Materialized Lake Views Simplifying Your Medallion ETL

Declarative, auto-orchestrated smart tables for Fabric Lakehouse



ProcureSQL

Justin Cunningham

Director, Analytics & Engineering

- Understand where MLVs fit in your ETL
- See refresh + lineage in a practical Fabric workflow
- Leave with a concrete when to use MLVs decision framework

Why ETL Gets Fragile

- Hard-coded dependency chains
- Manual retries and orchestration glue
- Pipeline drift over time
- Slow incident triage

What Modern Teams Need

- Declarative SQL-first transforms
- Platform-managed refresh behavior
- Transparent lineage and impact
- Lower operational overhead

FOUNDATION

Medallion + Materialized Lake Views

Medallion Overview

MLVs can be used to declaratively formalize layer transitions

Layer	Goal	MLV Fit
Bronze	raw immutable history, minimal transformations	
Silver	standardized and quality-controlled	
Gold	business-ready curated outputs	

- Define transformations as SQL, persist results as smart tables
- Let them manage refresh dependencies
- Use lineage to quickly evaluate downstream impact
- Reduce the number of notebooks/pipelines you have to maintain

Refresh Gains

- More predictable update behavior
- Fewer custom scheduler dependencies
- Consistent data freshness posture
- Improved resilience during failures

Lineage Gains

- Faster root-cause analysis
- Safer schema change planning
- Better governance visibility
- Cleaner stakeholder communication

CHOICE

Decision Framework

MLVs are ideal when...

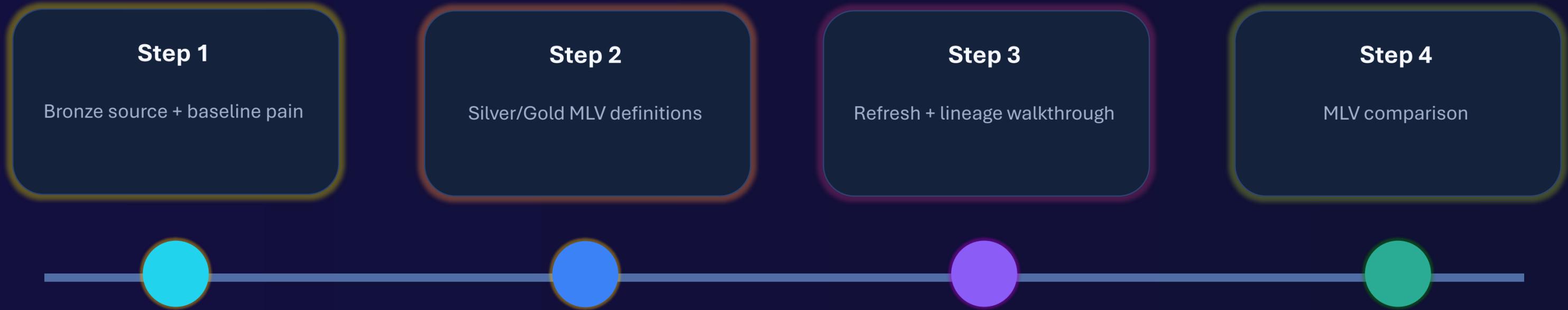
- Transforms are SQL-centric
- Maintainability > orchestration control
- Built-in dependency handling is wanted
- Lineage transparency is important

Prefer other options when...

- Heavy procedural logic is needed
- ML/non-SQL processing dominates
- Custom runtime control is required
- Warehouse semantics are mandatory

- MLVs simplify operations but reduce low-level runtime tuning flexibility
- Performance depends on query shape, partitioning, and workload patterns
- Operational simplicity often compounds value over time
- Start where orchestration pain is highest to maximize ROI

Demo Narrative





Demo

- Pick 1–2 brittle ETL flows as pilot candidates
- Refactor logic into declarative MLV statements
- Validate refresh and lineage expectations
- Standardize patterns and then scale

Materialized Lake Views: Simplifying Your Medallion ETL

- Materialized Lake Views can simplify medallion ETL significantly
- Refresh + lineage improve reliability, operability, and trust
- Use a decision framework — fit-for-purpose wins
- Pilot fast, prove value, then scale with standards

- Where is orchestration fragility hurting your team most today?
- What workloads are best first candidates for MLV adoption?
- What would successful implementation look like in your org?



ProcureSQL

Justin Cunningham

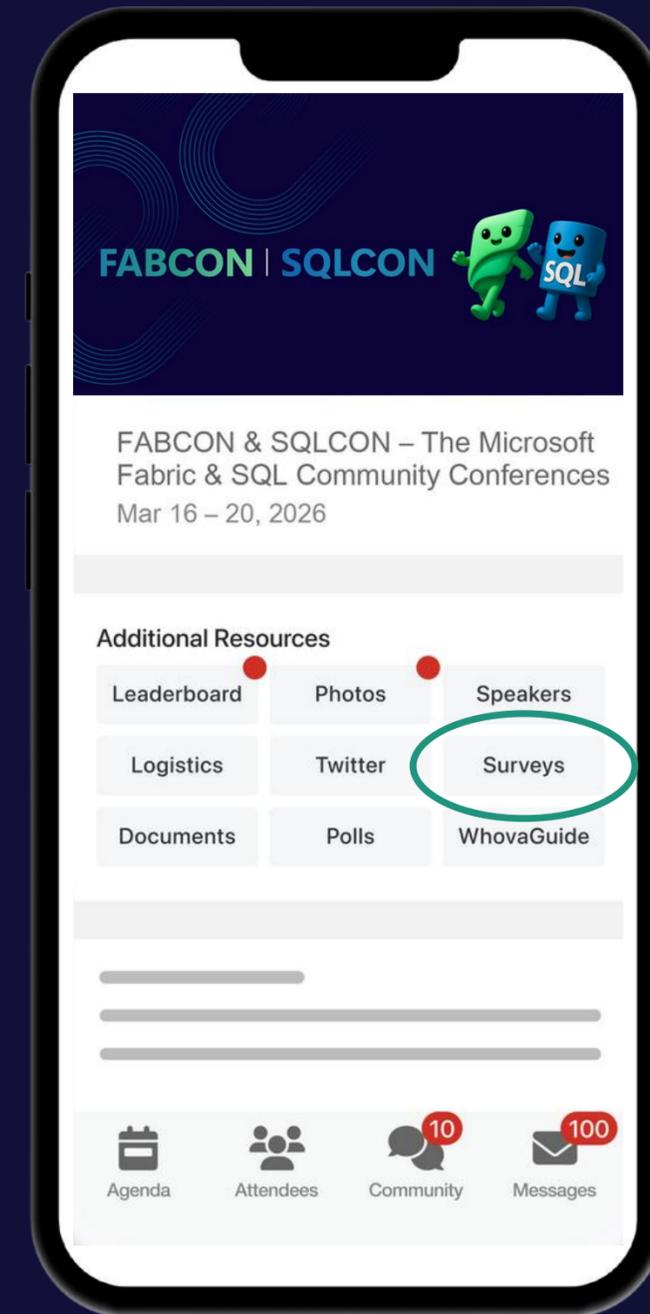
Director, Analytics & Engineering

Thank you for attending this session!

How was the session?



Complete Session Surveys in
Whova for your chance to WIN
PRIZES!



Sound off.
The mic is all yours.
Influence the product roadmap.

Join the Fabric User Panel



Share your feedback directly with our Fabric product group and researchers.

<https://aka.ms/JoinFabricUserPanel>

Join the SQL User Panel



Influence our SQL roadmap and ensure it meets your real-life needs

<https://aka.ms/JoinSQLUserPanel>