Building Scalable Data Warehouses

Microsoft Fabric · From Architecture to Production

Jonathan Stewart · **@sqllocks** · Sound Bi · soundbi.com

About Me

25+ Years in Data & Analytics

Helping enterprises migrate to and scale on Microsoft Fabric — from architecture and data modeling to Power BI and production governance.

Creator/Founder: SQLBites Community

Building the next generation of data practitioners.

Migrating Enterprises to Fabric

Real production migrations across healthcare, finance, retail, and manufacturing.

Hundreds of Technical Talks Globally

International speaker on data architecture and analytics engineering.

Sound BI

Presentation Agenda

Here's what we'll cover today to help you master Microsoft Fabric for scalable data warehousing.

1. **The Scalability Problem**
   Why traditional warehouses fail under pressure.
2. **Microsoft Fabric Overview**
   Key innovations and features for 2025–2026.
3. **Data Modeling & Loading**
   Effective design and ingestion strategies.
4. **Concurrency & Pipelines**
   Managing data flow at production scale.
5. **Governance & Security**
   Identity management, access control, compliance.
6. **Architecture Decisions**
   Lakehouse vs. Warehouse: choosing the right path.
7. **Roadmap & Cost**
   Upcoming features and controlling expenditure.
8. **Action Plan & Takeaways**
   Immediate steps you can implement Monday.

Quick Poll

🤦 Managing a Traditional Warehouse?

SQL Server, Synapse, Snowflake, or on-prem — raise your hand if you're still running one of these today.

🚀 Already in Production with Fabric?

Fabric Warehouse running in production right now — who's already made the leap?

Good news: By the end of this session, you'll have a concrete plan regardless of where you are today.

CHAPTER 1

The Scalability Problem

How many of you have a warehouse that exhibits these symptoms?

Takes Forever to Load

Refresh windows stretch into hours, delaying every downstream report and frustrating the entire business.

Costs a Fortune to Run

Bills grow faster than the business value being delivered — and nobody can explain why.

Nobody Understands It Anymore

Tribal knowledge, undocumented logic, sprawling dependencies — held together by one person who's about to leave.

Here's the thing: Most "scalability" problems aren't really about size. They're about architecture.

The Architecture Problem

Traditional warehouses fail because of **coupling, not capacity**. When compute and storage are inseparable, every scaling decision becomes expensive and constrained.

Compute and Storage Are Bundled

You can't scale one without scaling both — paying for capacity you don't need, every time.

Vertical Scaling Hits a Ceiling

There's only so big a single machine can get. After that, you're stuck — no path forward without a full platform rethink.

Horizontal Scaling Is Complex

Distributed systems require specialized expertise and introduce new failure modes most teams aren't equipped to manage.

Performance Degrades Non-Linearly

As data grows, query times don't scale linearly — they compound exponentially. A 2× data increase can mean a 10× slowdown.

The real issue: Your architecture wasn't designed for the scale you need today.

What Scale Really Means

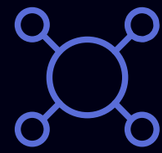
Scalability isn't a single dial — it's three distinct dimensions that must be designed for independently.

Data Volume

Handling terabytes of data without performance degradation as datasets grow over time. Not just today's size — tomorrow's.

User Concurrency

Sustaining 200+ simultaneous users querying live dashboards without throttling, timeouts, or degraded experience.



Query Complexity

Supporting complex analytical workloads — aggregations, multi-table joins, window functions — without dedicated tuning.

Traditional approach: Scale everything or nothing — an expensive and inflexible compromise that satisfies none of these dimensions well.

Real-World Example

A client scenario we encountered — this story will pay off at the end of the session.

The Setup

- 2TB data warehouse on SQL Server on-prem
- 200+ concurrent users across 5 regional offices
- Complex reporting queries averaging 15–30 seconds

What Broke

- Refresh window: 2 hours → 6+ hours and growing
- Queries timing out during peak business hours
- Hardware scale-up quoted at $300K+ — bought only 6 months of runway

Sound familiar? Stay with me — we'll come back to this story with the resolution at the end.

CHAPTER 2

Microsoft Fabric Changes the Game

GIGAOM RADAR 2025: LEADER & OUTPERFORMER

Independent Scaling

Compute and storage scale independently — pay only for what you use, when you use it.

OneLake Storage

Single unified storage layer — zero data duplication across all workloads and teams.

Delta Lake Foundation

ACID transactions, time travel, and schema evolution built in from day one. Open-source. Vendor-neutral.

Three Data Stores

Warehouse, SQL Database, and Lakehouse — all reading from the same OneLake data without duplication.

Microsoft shipped more in 2025 than most vendors ship in five years. This is not incremental progress.

OneLake: The Foundation

Think of OneLake as your **single source of truth** — one storage layer that every Fabric workload reads from and writes to.

What It Provides

- One storage layer across all Fabric workloads
- Built on Delta Lake — open-source, vendor-neutral
- Automatic replication across regions
- Built-in security and governance at the storage layer

Why It Matters

- No more data duplication between systems
- Consistency across all teams and tools
- Lower storage costs — one copy, many readers
- Simplified architecture — less to manage, less to break

Compute Separation Is Your Friend

The game-changer: compute scales independently from storage. This is the fundamental shift that makes Fabric economics work.

Traditional Architecture

- Storage + compute bundled together
- Scaling storage also scales compute cost
- Idle time still costs money — 24/7

Fabric Architecture

- Storage in OneLake (separate, predictable cost)
- Compute on-demand — pay per use only
- Auto-pause when not in use

$300K+
Old Annual Cost

6hr
Old Refresh Window

~$48K
New Annual Cost

45min
New Refresh Window

Result: Pay for what you actually use, not what you might need at peak. The savings compound over time.

Three Data Stores: Your Complete Toolkit

Fabric gives you three purpose-built data stores — not one-size-fits-all. Each is optimized for a distinct workload type.

Fabric Warehouse

ANALYTICS

Large-scale analytics, BI workloads, star schema design, T-SQL compatible, 200+ concurrent users. The go-to for Power BI and enterprise reporting.

SQL Database in Fabric

TRANSACTIONAL

Translytical (OLTP + analytics), AI apps, vector search, serverless. Operational workloads with built-in analytics — no ETL delay.

Lakehouse

DATA ENGINEERING

Mixed workloads, data science, Spark + notebooks, schema-on-read, ML training. Maximum flexibility for engineering and exploration.

Most organizations will use all three. They're designed to work **together** via OneLake, not compete.

Decision Matrix: Which Data Store?

Scenario	Use	Why
Enterprise BI, Power BI dashboards	**Fabric Warehouse**	Star schema, T-SQL
E-commerce app backend + analytics	**SQL Database**	Translytical, no ETL delay
Support chatbot, semantic search	**SQL Database**	Vector support, RAG patterns
Data engineering, ML model training	**Lakehouse**	Spark, notebooks, flexibility
Exploratory analysis, data science	**Lakehouse**	Schema-on-read
Legacy SQL Server migration	**Fabric Warehouse**	T-SQL compatibility
Small operational database (<4TB)	**SQL Database**	Serverless, auto-scaling

You're not locked in. Start with one data store, add others as your needs evolve. All share OneLake underneath.

CHAPTER 3

What's New: 2025–2026 Features

GENERALLY AVAILABLE

Data Clustering

Automatic physical data organization — no manual partitioning schemes required. One line of DDL activates it.

IDENTITY Columns

Auto-generated surrogate keys — eliminates custom scripts, sequences, and key collision risk.

MERGE Statement

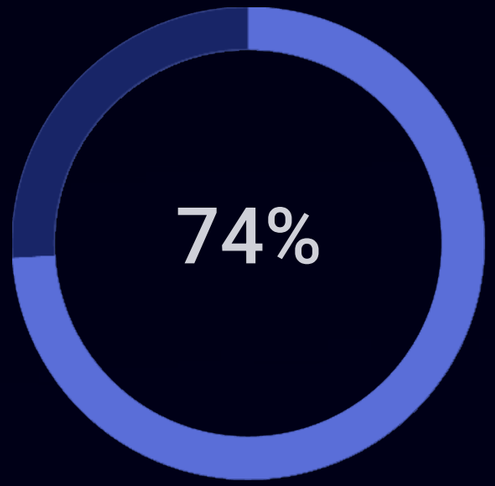
Native upsert patterns for incremental loads — replaces clunky delete+insert logic with a single atomic operation.

Warehouse Snapshots

Point-in-time consistency for auditing, compliance, and instant rollback. Configurable retention up to 90 days.

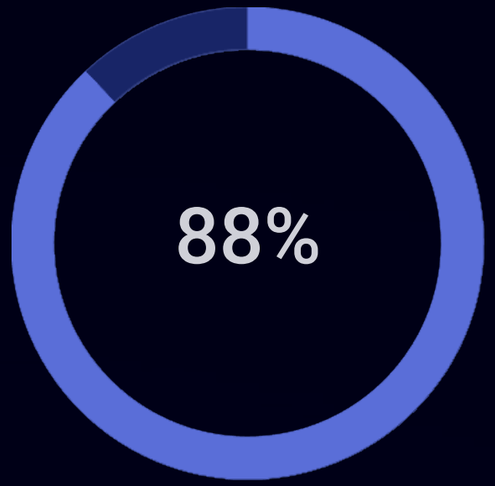
Data Clustering: The Biggest Win

Fabric analyzes your query patterns and physically reorganizes data. One line of DDL — no indexes, no manual partitioning strategies to maintain.



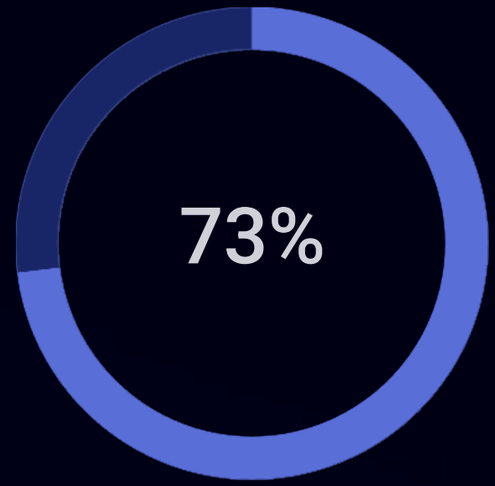
Faster Execution

47 sec → 8 sec on a 5B-row fact table



Fewer Rows Scanned

5B → ~600M rows read per query



Lower Compute Cost

150 CUs → 40 CUs per execution

```
CREATE TABLE FactSales (...) WITH ( CLUSTER BY (OrderDateKey) )
```

Your move: Enable clustering on your largest fact tables first. This is the fastest ROI in the entire platform.

MERGE + Incremental Loading

Full refreshes don't scale. MERGE handles insert, update, and delete in one atomic operation — the foundation of any production-grade incremental load.

```
MERGE INTO FactSales AS target
USING StagingSales AS source
ON target.SalesID = source.SalesID
WHEN MATCHED THEN UPDATE SET ...
WHEN NOT MATCHED THEN INSERT ...;
```

- Replaces the old "delete + insert" anti-pattern that causes gaps and duplicates under failure conditions.
- Guaranteed data consistency — no duplicates, no lost updates, no partial loads.
- Pair with watermark columns for efficient delta detection — load only what changed.

IDENTITY Columns & Warehouse Snapshots

IDENTITY Columns

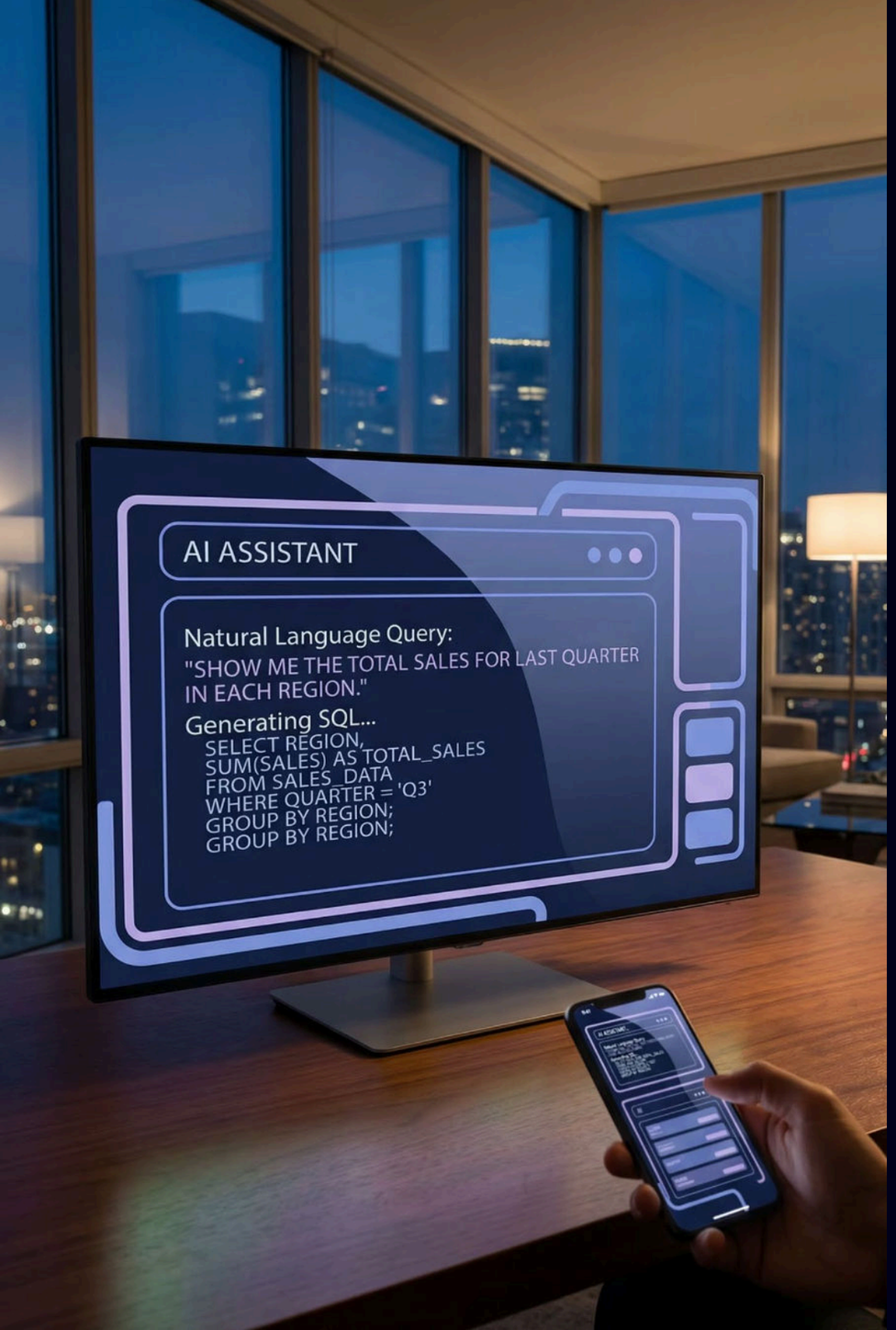
Auto-generated surrogate keys — no more MAX(ID)+1, no key collisions, no race conditions under concurrent load.

```
CREATE TABLE DimCustomer
(CustomerKey INT
IDENTITY(1,1), ...)
```

Warehouse Snapshots

Point-in-time consistency for your entire warehouse — instant creation, standard SQL queryable.

- Retention configurable: 7, 30, or 90 days
- Read-only, queryable with standard T-SQL
- Use cases: audit trails, rollback, month-end reporting locks



Copilot, Data Agent + Migration Assistant

Copilot + Data Agent

- Natural language querying for your org's data — no SQL required for business users
- Copilot writes SQL and suggests query optimizations in real time
- Data Agent lets business users query in plain English — serious productivity multiplier

Migration Assistant

- Schema migration — tables, views, indexes automated
- Data type mapping from SQL Server to Fabric
- Stored procedure conversion with AI assistance
- Weeks of migration work reduced to hours

Let's Be Honest: Current Limitations

Fabric is powerful, but it's not perfect. Here's what to know before you build your production architecture:

SQL Database in Fabric

4TB data size cap — fine for operational workloads, not suitable for petabyte-scale analytics.

Stored Procedures

Support is growing but still limited versus the full SQL Server feature set. Plan for refactoring.

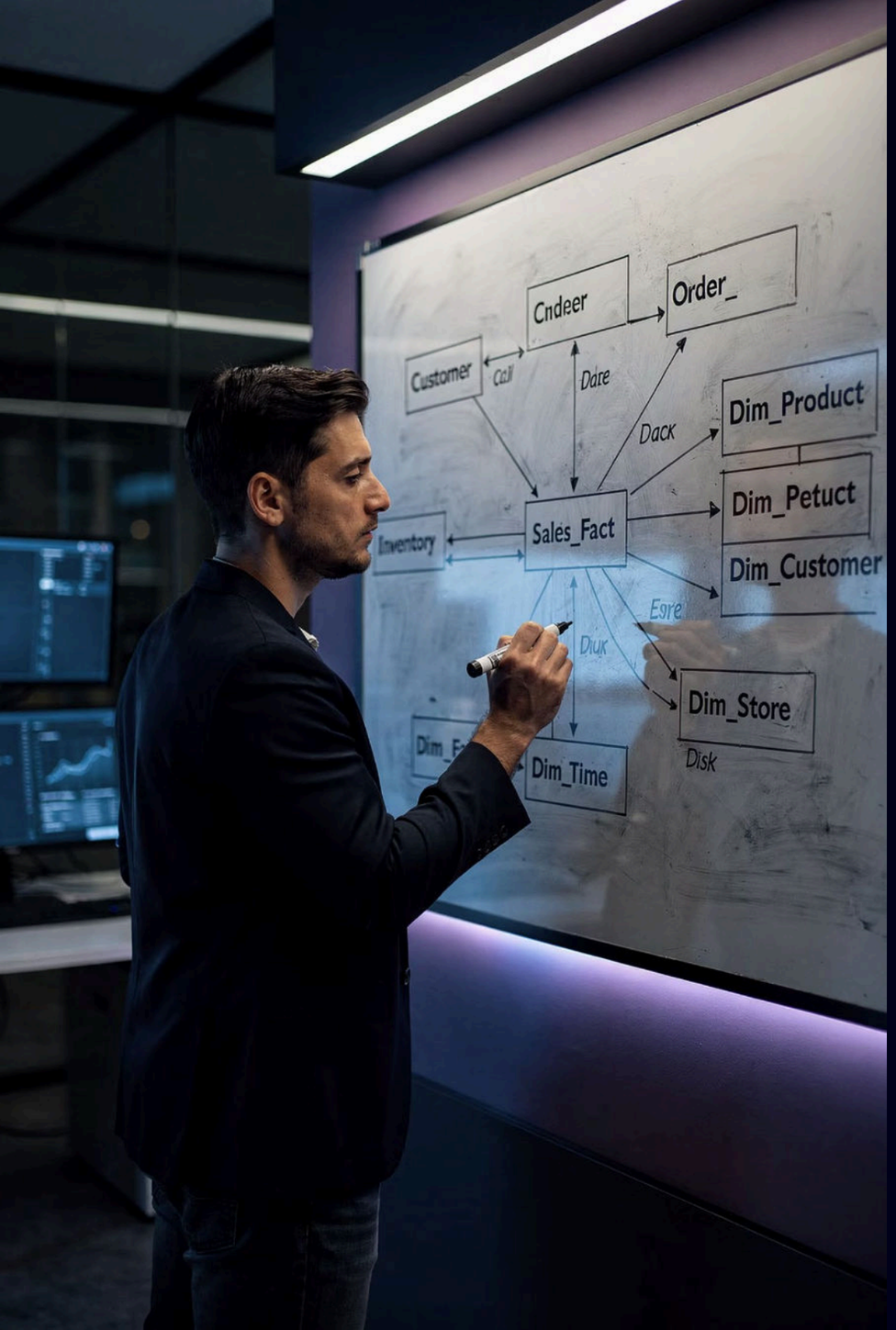
Elastic Pools

Not yet available (coming Q2–Q3 2026) — multi-warehouse compute sharing is still manual today.

Materialized Views + Cold Start

Materialized views are coming soon. High Concurrency Mode helps cold-start latency, but first-query-after-idle can still be slow.

None of these are dealbreakers. They are engineering trade-offs in a platform that is maturing fast — most are on the roadmap.



CHAPTER 3

Data Modeling Still Matters

Get the logical layer right. Let Fabric optimize the physical layer. The best platform in the world can't rescue a poorly designed schema.

Star Schema Remains King

Fact + dimensions. Power BI is literally optimized for this shape — don't fight it.

Right-Size Your Data Types

VARCHAR(50) vs VARCHAR(MAX) matters at 5 billion rows. Every byte multiplied by billions adds up.

Let Data Clustering Handle Physical Layout

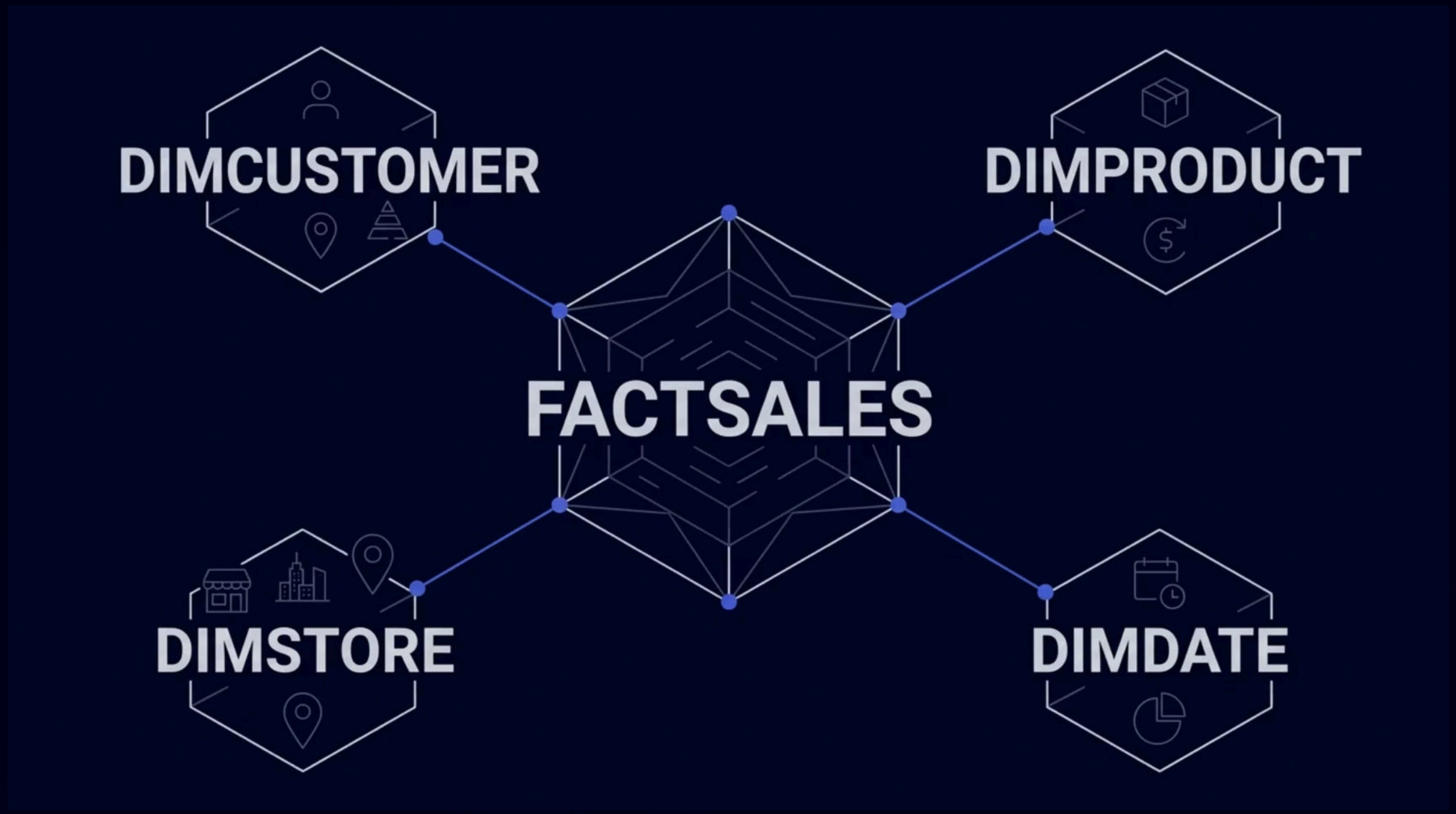
You focus on logical design — Fabric organizes the files. Don't introduce manual partitioning unless you have extremely specific needs.

Normalize with Purpose

Denormalize for performance where it makes sense. Don't introduce unnecessary complexity just because you can.

Star Schema: Still the Best Pattern

The star schema remains the gold standard for analytical workloads. Simple for end users, optimal for Power BI, and predictable query patterns at any scale.



The hub-and-spoke design minimizes joins, maximizes query optimizer efficiency, and makes the semantic layer easy to maintain as your data grows.

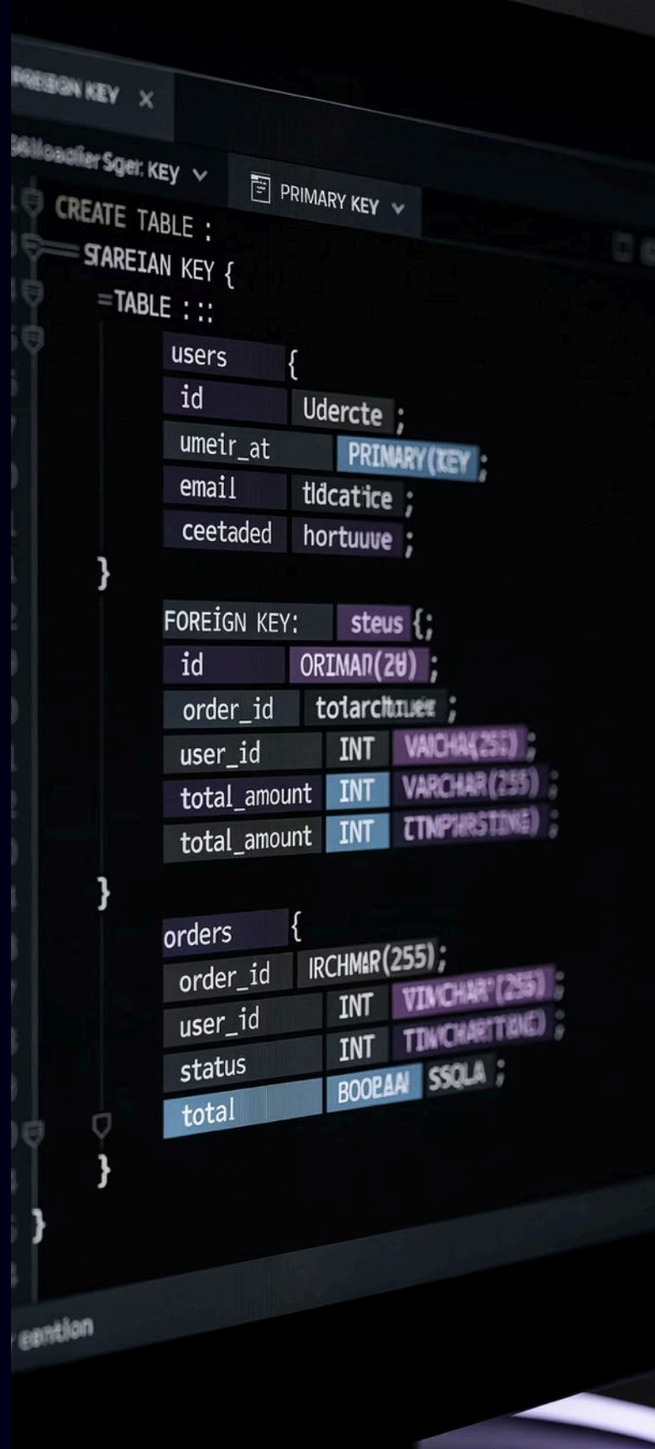
Right-Sizing Data Types

This matters more at scale than most architects expect. Small decisions at design time compound enormously at production volume.

Column Type	CustomerName	CustomerEmail	Storage at 100M rows
Over-sized (VARCHAR MAX)	2,000 bytes	500 bytes	~25 GB
Right-sized (VARCHAR 100)	100 bytes	100 bytes	~2 GB

Impact: 12× storage difference. Faster scans. Lower compute cost. Better compression ratios in Parquet.

- You define tables, columns, relationships, and data types — this is your domain.
- Fabric handles file layout, compression, and query optimization — let it.
- Avoid manual partitioning unless you have extremely specific, validated requirements.



COPY INTO + Watermarks

Use COPY INTO instead of row-by-row INSERT — the performance difference is dramatic and the pattern is simple to implement.

```
COPY INTO FactSales
FROM 'https://.../sales/*.parquet'
WITH (FILE_TYPE = 'PARQUET');
```

15× faster. 10M rows: 8 minutes with COPY INTO vs. 2 hours with INSERT loops.

Watermark Pattern for Delta Detection

01

Read Last Watermark

Query your control table for the timestamp or key of the last successful load.

02

Load Only New Rows

Filter source data: `WHERE LastModifiedDate > last_watermark` — skip everything already loaded.

03

Update Watermark

After a successful load, write the new high-water mark back to the control table.

File Ingestion Sweet Spot

File size and count dramatically impact load performance and parallelism. Getting this right is one of the highest-leverage tuning decisions in your pipeline.

1

Too Small (<100 MB)

High overhead, poor parallelism, excessive file count to manage. Metadata operations dominate execution time.

2

Sweet Spot (100 MB – 1 GB) ✅

Optimal parallelism, efficient memory use, fast recovery on failure. This is your target zone.

3

Too Large (>1 GB)

Reduced parallelism, memory pressure, longer recovery time when a single file load fails.

Key insight: 10 files × 500MB = 10-way parallelism. 1 file × 5GB = single-threaded. High file count beats large file size.

CHAPTER 4

Concurrency Breakthroughs

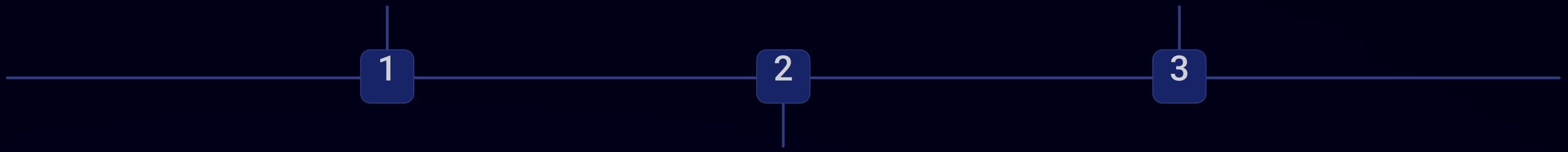
2025–2026 brings three breakthroughs that eliminate the most common production pain points teams have faced since Fabric launched.

Data Compaction Preemption

Fabric detects conflicting writes, pauses background compaction, lets your ETL complete cleanly. No more mysterious 3am failures.

High Concurrency Mode

Eliminates cold-start latency for Lakehouse. Queries execute immediately. Trade-off: slightly higher idle cost.



Workload Isolation

Separate ETL and reporting into different warehouses reading from the same OneLake data. Neither workload blocks the other.

Pipeline Architecture That Scales

Five rules that separate pipelines that survive production from those that fail at 3am on a Monday.

1

Modular Components

Small, reusable, single-responsibility pipelines. If it does more than one thing, split it.

2

Error Handling + Retry

Expect transient failures. Build automatic recovery with exponential backoff: wait 30s → retry (max 3) → alert + log.

3

Parallel Table Loads

Load independent tables simultaneously. Don't serialize what doesn't need to be serial.

4

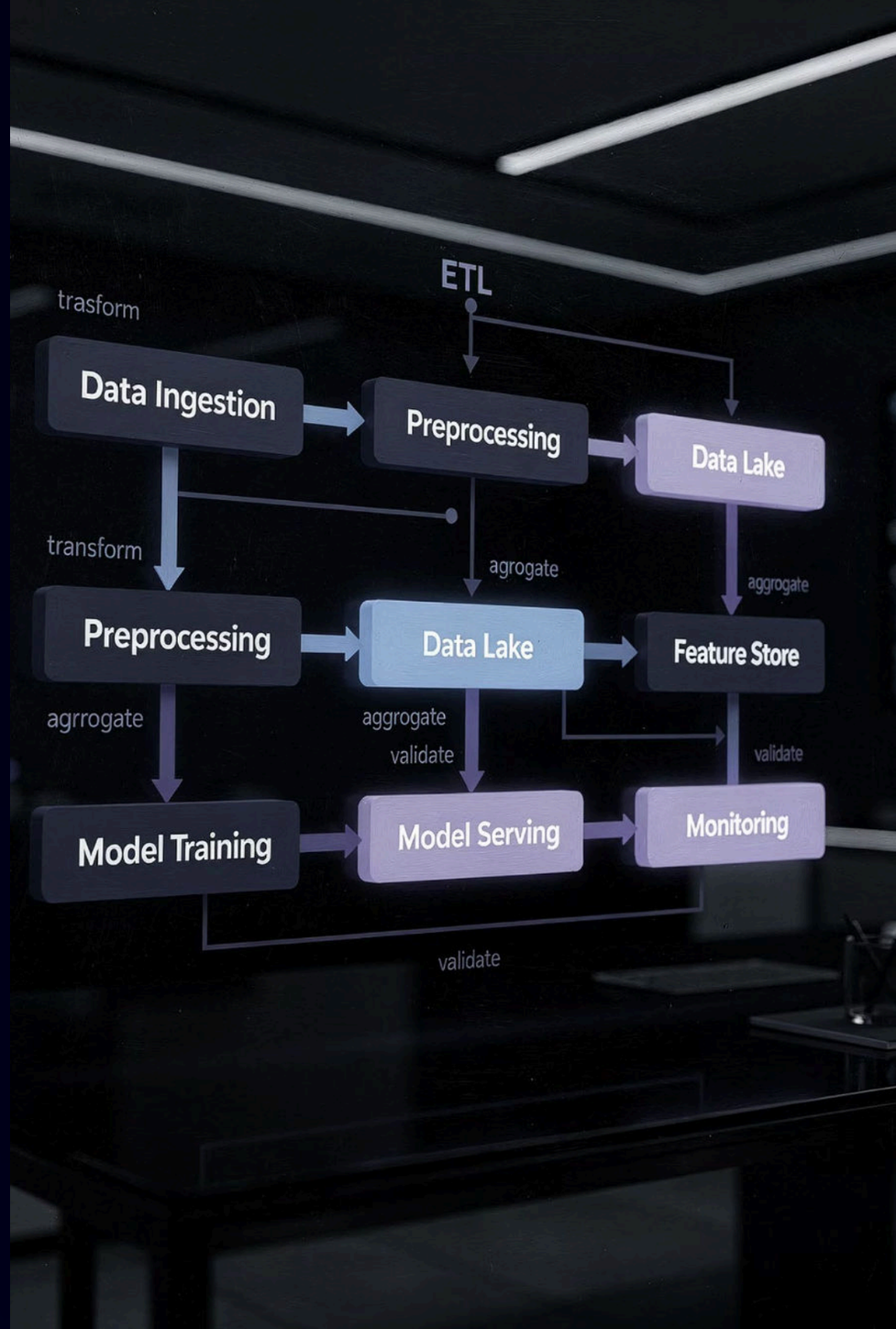
Avoid Giant Monoliths

"One pipeline does everything" is impossible to debug, test, or hand off to another engineer.

5

Set Workspace Collation Early

This is expensive to change later. Make the decision at project start, not after go-live.



Modular Design + Error Handling

1

Bad: Monolithic Pipeline

`Extract_Transform_Load_Validate_Notify`

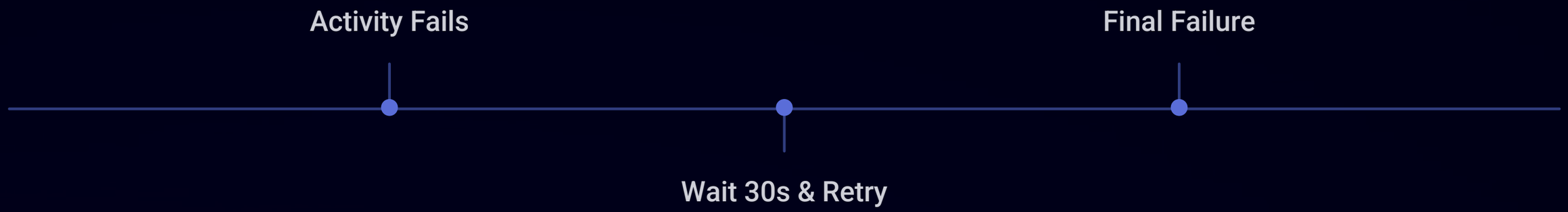
47 activities, 8 levels deep — impossible to troubleshoot, test, or hand off. One failure breaks everything downstream.

2

Good: Modular Design

- Pipeline 1: `Extract_Sources`
- Pipeline 2: `Transform_FactSales`
- Pipeline 3: `Validate_DataQuality`
- Pipeline 4: `Orchestrator` (coordinates all)

Error Handling Pattern



Build recovery into every activity. Silent failures in production pipelines are the leading cause of stale dashboards and data trust erosion.

CHAPTER 5

Governance: Easier Than You Think

SQL Audit Logs — 5-Minute Setup

1. Navigate to Workspace Settings → Monitoring → SQL Audit Logs
2. Toggle: Enable auditing
3. Select events: reads, writes, schema changes, permission changes
4. Set retention: 7, 30, or 90 days
5. Done. Logs start flowing immediately.

Identity + DevOps at Scale

- **Identity at Scale:** Expanded from 1,000 to 10,000 identities per tenant (Feb 2026) — enterprise-ready at any org size.
- **CI/CD for GraphQL APIs:** Git-enabled source control, PR workflows, full dev → test → prod promotion pipeline.

CHAPTER 6

Architecture: Retail E-Commerce

5M customers · 50K daily orders · Real-time inventory

SQL Database in Fabric

- Orders, Customers, Inventory (transactional writes)
- App writes directly — zero ETL delay
- Real-time inventory dashboards for ops teams

Fabric Warehouse

- Historical analytics (3+ years of order history)
- Star schema: FactOrders, DimCustomer, DimProduct
- Power BI semantic model, CLV analysis, campaign attribution

OneLake as shared storage: Both stores read from the same data — no sync jobs, no duplication, one governance model.

Architecture: SaaS Platform with AI

B2B SaaS · 10K tenants · AI assistant · Semantic search

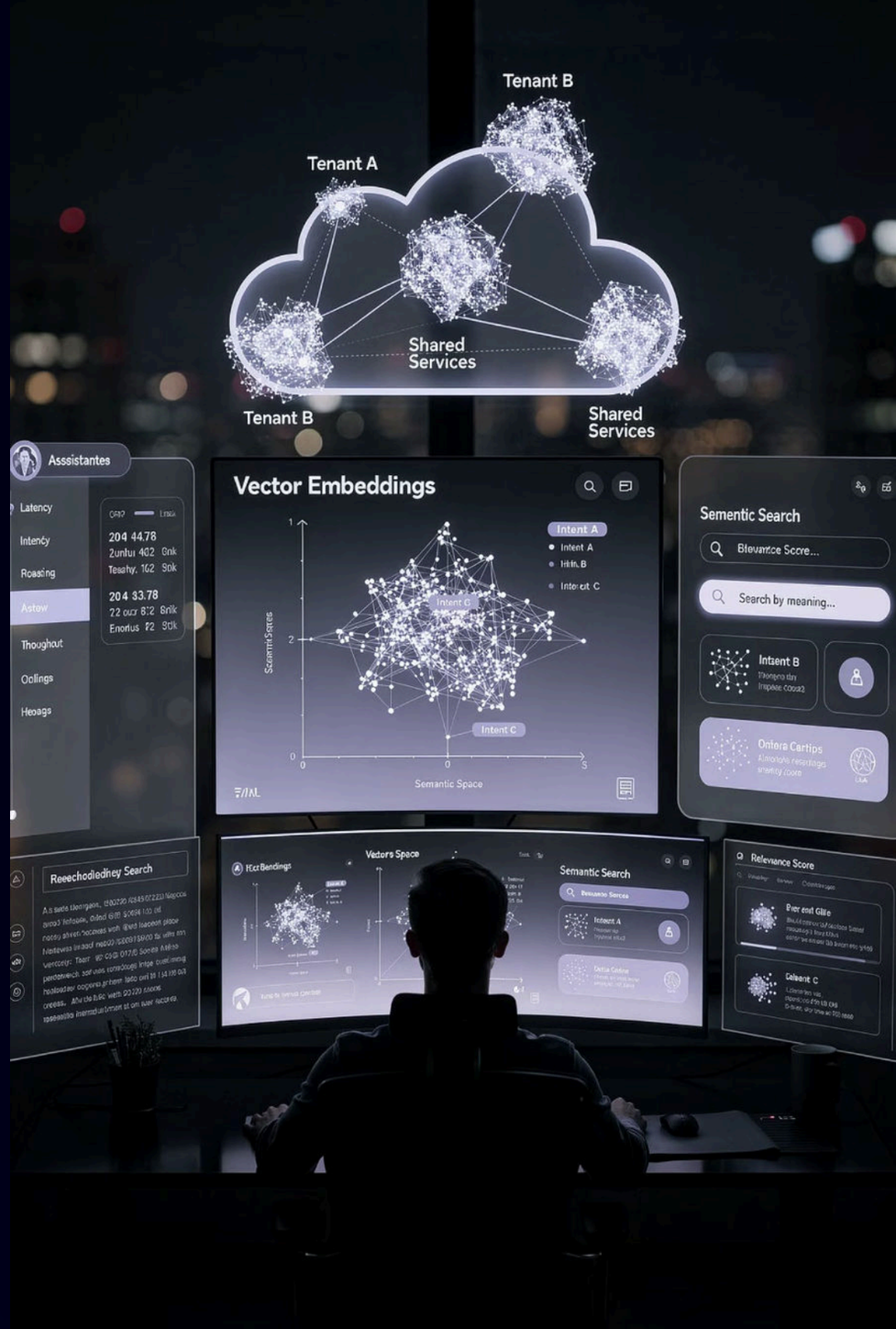
SQL Database in Fabric

- Multi-tenant isolation per customer
- Vector embeddings for RAG (retrieval-augmented generation)
- AI assistant powered by semantic search across tenant data

Fabric Warehouse

- Aggregated cross-tenant analytics
- Executive dashboards: MRR, churn, feature adoption
- Star schema: FactUsage, DimTenant, DimFeature

OneLake as shared storage: Operational AI data and analytical reporting coexist without duplication or sync complexity.



Architecture: Healthcare Analytics

Hospital system · Patient records · IoT devices · HIPAA compliance

SQL Database in Fabric

- Patient admissions and appointment scheduling
- Real-time bed availability for clinical operations
- SQL Audit Logs enabled for HIPAA compliance tracking

Fabric Warehouse

- Enterprise outcomes and cost reporting
- Star schema: FactEncounters, DimPatient, DimProvider
- Regulatory reports with warehouse snapshots for audit trail

OneLake as shared storage: One governance and security model spans all data — critical for HIPAA and Joint Commission audits.

Architecture: Financial Services

Investment firm · Trading data · Regulatory reporting

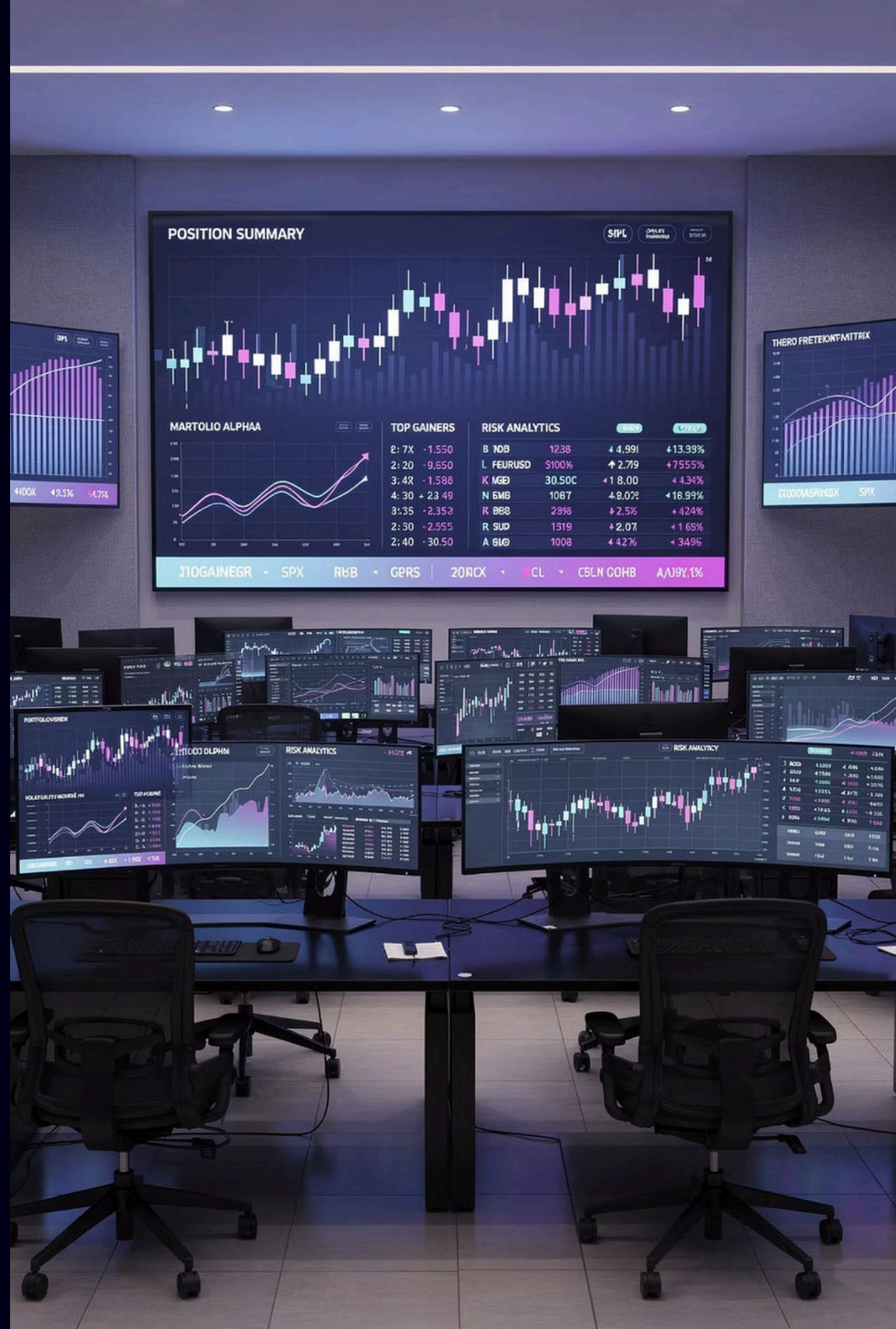
SQL Database in Fabric

- Trade execution, orders, and position management
- Real-time P&L and client portal data
- Sub-second latency for trading operations

Fabric Warehouse

- Regulatory reporting: MiFID II, Basel III
- Risk analytics: VaR, stress testing, exposure analysis
- GraphQL APIs with full CI/CD source control

OneLake as shared storage: Regulatory-grade audit trails and real-time trading data share one platform — one compliance posture.



Architecture: Manufacturing IoT

500+ machines · Predictive maintenance · Supply chain optimization

SQL Database in Fabric

- Production schedules and work order management
- Real-time machine status from IoT sensor streams
- Quality inspection records and defect tracking

Fabric Warehouse

- Manufacturing analytics: OEE, yield, downtime
- Supply chain reporting and procurement analysis
- Star schema: FactProduction, DimMachine, DimShift

OneLake as shared storage: Sensor data flows from IoT ingest through Lakehouse to Warehouse without duplication or transformation lag.

Key Insight: They Work Together

The pattern across all five architecture scenarios is the same. Each data store plays a distinct role — and together they cover every workload type your organization needs.

1

SQL Database in Fabric

Operational data, frequent writes, real-time queries, AI workloads, vector search, translytical patterns.

2

Lakehouse

Raw data, high-volume ingestion, data science, exploratory analysis, ML model training, Spark workloads.

3

Fabric Warehouse

Historical analytics, enterprise BI, regulatory reporting, star schema, Power BI semantic models.

One governance model. One security layer. One cost model. OneLake unifies everything underneath.

Remember That Client?

Let's close the loop on the story from the beginning. Here's what the migration actually delivered.

87%

Faster Refresh

6+ hours → 45 minutes

80%

Query Speedup

15–30 sec → 3–5 sec

84%

Cost Reduction

$300K+ → ~$48K/year

500+

Concurrent Users

Up from 200, no degradation

What we did: Star schema redesign, Data Clustering on 4 fact tables, MERGE patterns replacing full refreshes, workload isolation, and watermark-based incremental loads across the board.

The ROI Is Real: Industry Data

The Forrester Total Economic Impact Study (commissioned by Microsoft) measured real Fabric deployments — not projections:

379%

Overall ROI

$9.79M

Net Present Value

Over 3 years

$778K

Infrastructure Savings

From decommissioning legacy systems

$3.6M

Analyst Productivity

Reduction in time-to-output

$1.4M

Better Decisions

Profit from fresher, more reliable data

These numbers come from real Fabric deployments. The ROI case for migration is not theoretical — it is documented and reproducible.

CHAPTER 7

Coming Soon: 2026 Roadmap

Elastic Pools

Share compute across multiple warehouses. Expected 30–40% cost reduction for multi-warehouse environments. ETA: Q2–Q3 2026.

Materialized Views

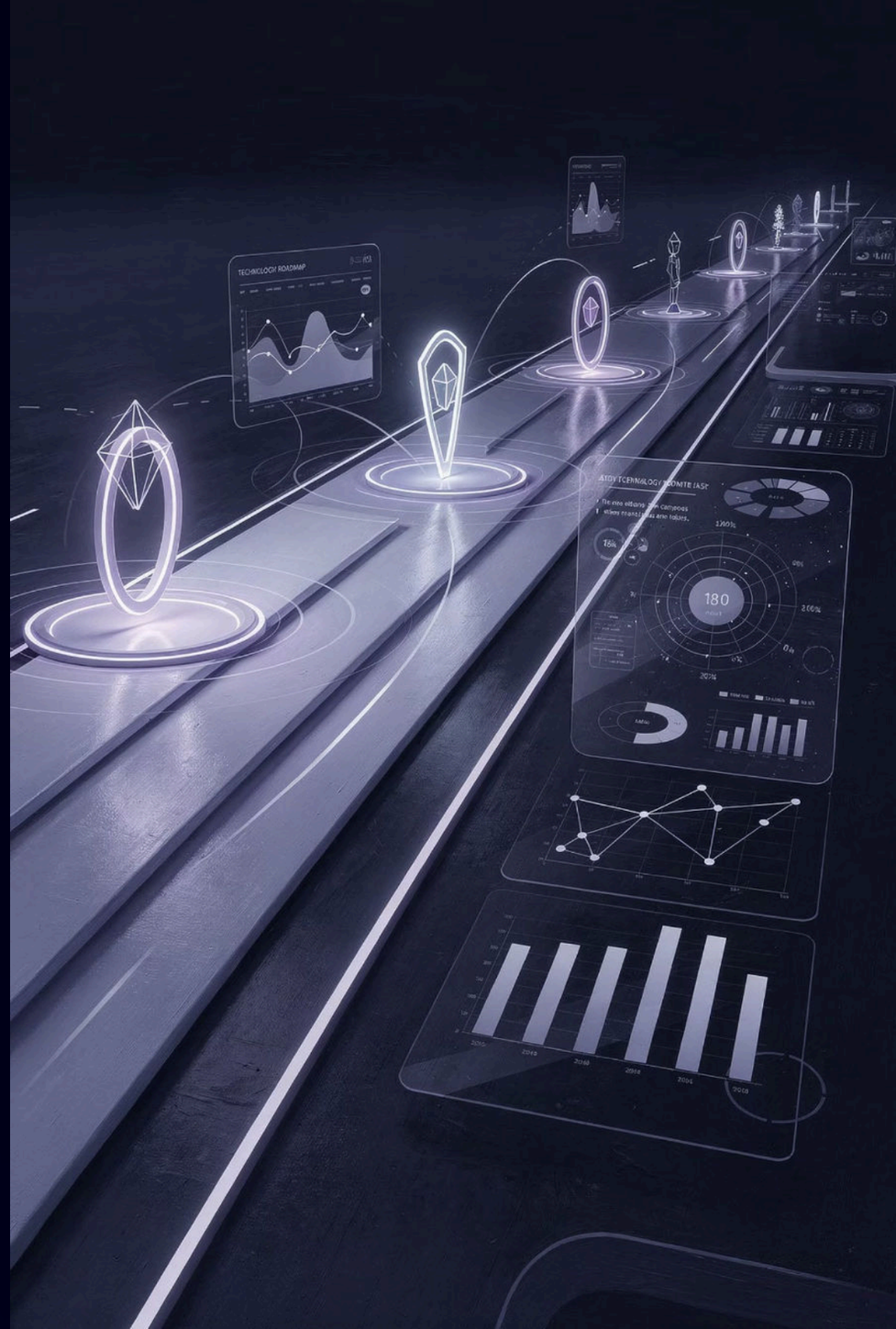
Pre-aggregated tables with automatic refresh. No more manual scheduling or summary table maintenance.

Full-Text Search

Native full-text querying across large text columns — critical for support, compliance, and content analytics.

Liquid Clustering GA

Automatic data organization for Lakehouse (currently Warehouse-only). Unified clustering story across both stores.



Cost Management: Four Rules

Fabric can scale infinitely — but so can your bill if you don't establish cost governance before you go to production.

1

Right-Size Your Capacity SKU

F2 for dev (~$500/mo), F8 for production (~$2K/mo), F64+ for enterprise ($16K+/mo). Don't over-provision on day one.

2

Enable Auto-Pause

For idle warehouses — trigger after 15 minutes of inactivity. Never pay for compute that isn't running.

3

Set Consumption Budgets and Alerts

Configure spending alerts before you need them, not after you receive an unexpected invoice. This is not optional.

4

Monitor Continuously

Track compute, storage, and query patterns daily. Use the Fabric Capacity Metrics app — it is purpose-built for this.

CHAPTER 8

Your Action Plan for Monday

Four concrete actions you can take this week — each with measurable impact and no downtime required.

Enable Data Clustering on Your Largest Fact Tables

Fastest ROI in the entire platform. One line of DDL per table. No downtime, no migration, no risk.

Convert One Full-Refresh Pipeline to MERGE + Watermarks

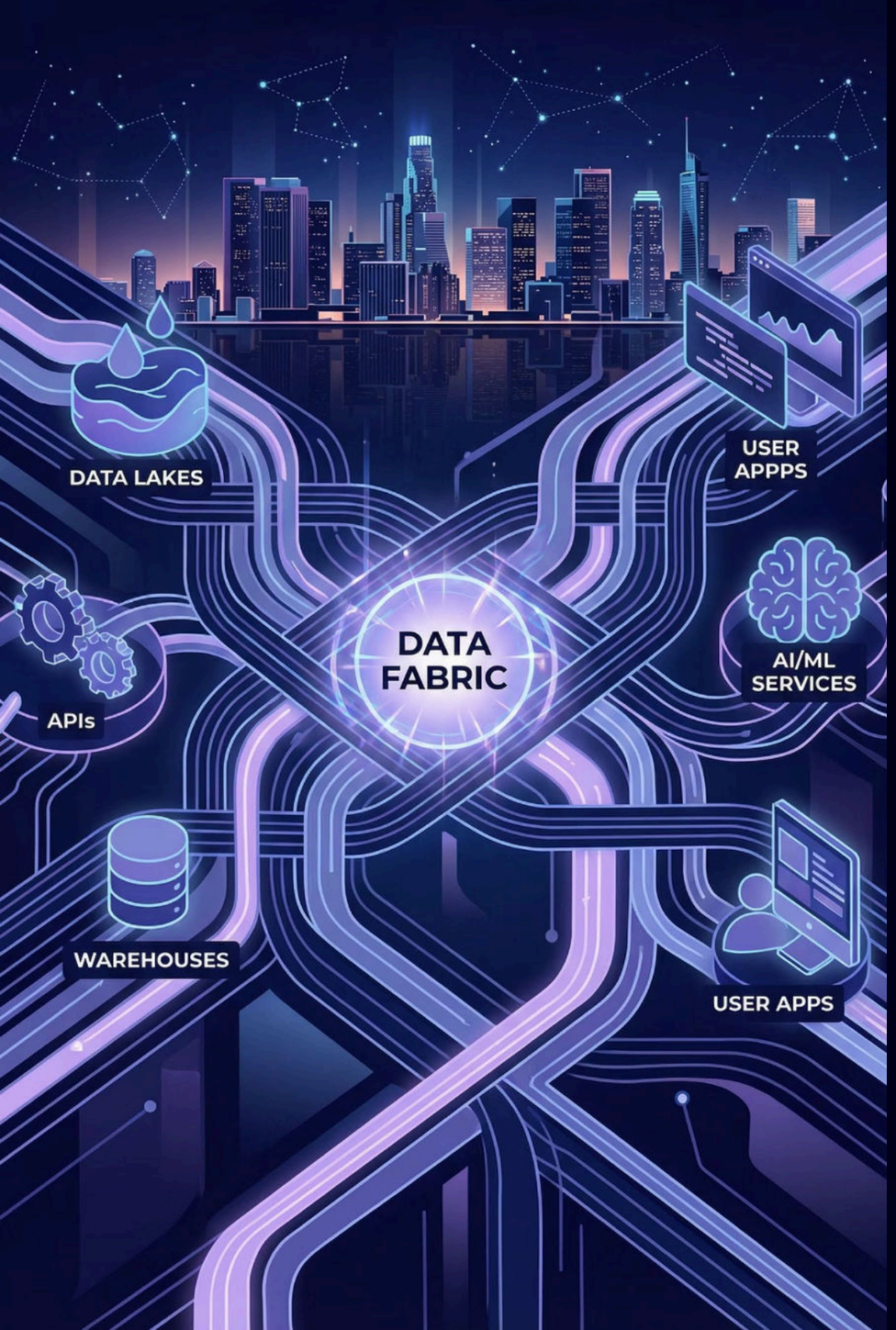
Replace 6-hour full loads with 2-minute incrementals. Pick your most painful pipeline and start there.

Audit Data Types — Right-Size Integers, Drop NVARCHAR

Smaller Parquet files = faster scans = lower compute cost. This is free performance you're leaving on the table.

Set Up Workload Isolation — Separate ETL from Reporting

Neither workload blocks the other. Both read from OneLake. Your business users will notice immediately.



Key Takeaways

1

Fabric Has Three Purpose-Built Data Stores

Warehouse (analytics), SQL Database (translytical + AI), Lakehouse (flexibility). Most organizations will use all three — they are designed to coexist.

2

They Work Together, Not Against Each Other

OneLake unifies storage. Data flows freely across all three stores. One governance model, one security layer, one bill.

3

Production-Ready Features Are Here

Data Clustering, MERGE, IDENTITY, Snapshots, SQL Database — all generally available today. You can build on this now.

4

Start Small, Scale Smart

One incremental load pattern. One data store. Monitor early, adjust fast. Complexity is the enemy of production stability.

The window to get ahead is now. Teams investing in Fabric architecture today will have a 12–18 month competitive advantage over those who wait.

Resources

Official Documentation

- learn.microsoft.com/fabric
- blog.fabric.microsoft.com
- learn.microsoft.com/fabric/database

Research

- Forrester TEI Study: "Total Economic Impact of Microsoft Fabric"
- Jimi Adeboyejo: "Maximizing ROI with Microsoft Fabric" (LinkedIn)

Community

- SQLBites Community: sqlbites.net
- Fabric Community Forums: community.fabric.microsoft.com
- @sqllocks on Twitter/X and LinkedIn

This Session

All code samples, architecture diagrams, and migration checklists from this talk are available at Jonathan.stewart@**soundbi.com**.

SOLLOCKS

SQLBites

JONATHAN STEWART

@sqllocks

Principal Consultant · Data Warehousing
Microsoft Fabric · Enterprise Analytics Architecture

</> BUILT DIFFERENT

CONNECT

LINKEDIN
Connect with me

SQLBITES
Microsoft Fabric content, insights & resources

BOOK A MEETING
Schedule time to talk data strategy

Questions? Let's Talk.

Jonathan Stewart · Principal Analytics Consultant, Sound Bi

Jonathan.stewart@soundbi.com· @sqllocks

→ Your Migration Scenario

→ Which Data Store Fits

→ Architecture Design

→ Performance Challenges

→ Where to Start

> *Keep learning. Keep building. Keep pushing the boundaries of what's possible with data.*

Sound BI