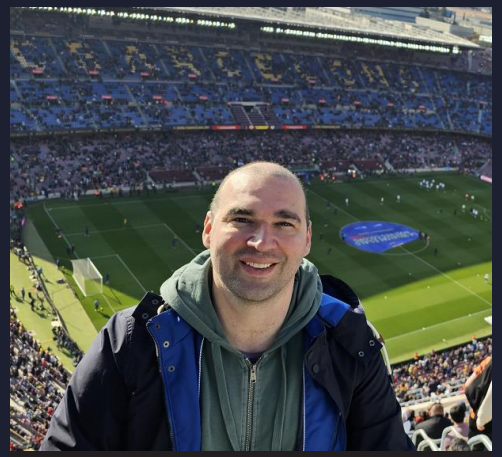
It's just a SQL – Crash Course on Fabric Warehouse for T-SQL Ninjas!

Nikola Ilic

Data Mozart, Microsoft Data Platform MVP

Data Grillen
May 16th 2024



Microsoft® Most Valuable Professional

Nikola Ilic

Consultant & Trainer

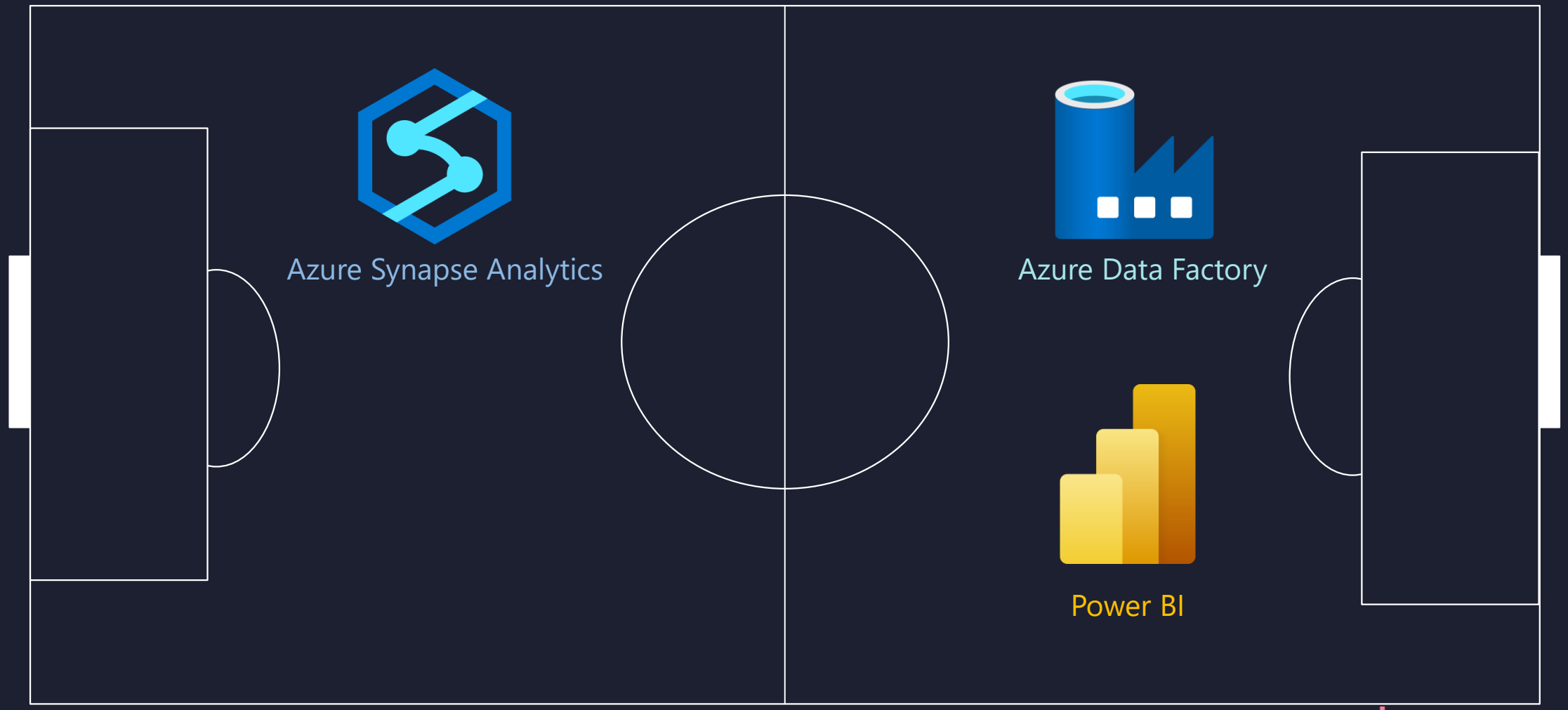
data-mozart.com

@DataMozart

learn.data-mozart.com

- *I'm making music from the data*!
- Power BI and SQL addict, blogger, speaker...
- Father of 2, Barca & Leo Messi fan...

What is a Microsoft Fabric?



@DataMozart

“Players” in Microsoft Fabric

Data Factory

Data Engineering

Data Warehouse

Data Science

Real Time Analytics

Data Activator

Power BI

Microsoft Synapse

OneLake

You'll NOT learn to write SQL!

```sql
SELECT cu.FirstName
    , cu.LastName
    , geo.RegionCountryName AS country
    , geo.CityName AS city
    , SUM(fso.SalesAmount) AS salesAmount
FROM dbo.FactOnlineSales AS fso
    INNER JOIN dbo.DimCustomer AS cu ON cu.CustomerKey = fso.CustomerKey
    --INNER JOIN dbo.DimProduct AS pr ON pr.ProductKey = fso.ProductKey
    INNER JOIN dbo.DimGeography AS geo ON geo.GeographyKey = cu.GeographyKey
WHERE cu.EducationLevel = 'Bachelors'
    AND geo.RegionCountryName = 'United States'
GROUP BY cu.CustomerKey
    , cu.FirstName
    , cu.LastName
    , geo.RegionCountryName
    , geo.CityName
HAVING SUM(fso.SalesAmount) > 100
ORDER BY salesAmount DESC
```

Based on a “non-real-life” story!

OLAP SQL
≠
OLTP SQL

OLAP

```sql
SELECT p.Color
    , SUM(fis.SalesAmount) AS SalesAmount
FROM FactInternetSales AS fis
INNER JOIN DimProduct AS p on p.ProductKey = fis.ProductKey
GROUP BY p.Color
```

OLTP

```sql
INSERT INTO InternetSales
(SalesKey
, CustomerKey
, ProductKey
, SalesAmount
, ....
)

VALUES (....)
```

What is a Warehouse in Fabric?

It’s a…data warehouse!

What is a Warehouse in Fabric?

Stores data in a proprietary format

Stores data in an open format (Delta)

What is a Warehouse in Fabric?

Stores data in an open format (Delta)

Why Parquet Format?

Data compression

Columnar storage

Language agnostic

Open-source format

Support for complex data types

Row-Based Storage

	Product	Customer	Country	Date	Sales Amount
Row 1	Ball	John Doe	USA	2023-01-01	100
Row 2	T-Shirt	John Doe	USA	2023-01-02	200
Row 3	Socks	Maria Adams	UK	2023-01-01	300
Row 4	Socks	Antonio Grant	USA	2023-01-03	100
Row 5	T-Shirt	Maria Adams	UK	2023-01-02	500
Row 6	Socks	John Doe	USA	2023-01-05	200

Row-Based Storage

	Product	Customer	Country	Date	Sales Amount
Row 1	Ball	John Doe	USA	2023-01-01	100
Row 2	T-Shirt	John Doe	USA	2023-01-02	200
Row 3	Socks	Maria Adams	UK	2023-01-01	300
Row 4	Socks	Antonio Grant	USA	2023-01-03	100
Row 5	T-Shirt	Maria Adams	UK	2023-01-02	500
Row 6	Socks	John Doe	USA	2023-01-05	200

Row-Based Storage

	Product	Customer	Country	Date	Sales Amount
Row 1	Ball	John Doe	USA	2023-01-01	100
Row 2	T-Shirt	John Doe	USA	2023-01-02	200
Row 3	Socks	Maria Adams	UK	2023-01-01	300
Row 4	Socks	Antonio Grant	USA	2023-01-03	100
Row 5	T-Shirt	Maria Adams	UK	2023-01-02	500
Row 6	Socks	John Doe	USA	2023-01-05	200

Column-Based Storage

Column 1	Column 2	Column 3	Column 4	Column 5
Product	Customer	Country	Date	Sales Amount
Ball	John Doe	USA	2023-01-01	100
T-Shirt	John Doe	USA	2023-01-02	200
Socks	Maria Adams	UK	2023-01-01	300
Socks	Antonio Grant	USA	2023-01-03	100
T-Shirt	Maria Adams	UK	2023-01-02	500
Socks	John Doe	USA	2023-01-05	200

Parquet is a columnar format that stores the data in row groups!

Parquet Storage

	Column 1	Column 2	Column 3	Column 4	Column 5
	Product	Customer	Country	Date	Sales Amount
Row group 1	Ball	John Doe	USA	2023-01-01	100
	T-Shirt	John Doe	USA	2023-01-02	200
Row group 2	Socks	Maria Adams	UK	2023-01-01	300
	Socks	Antonio Grant	USA	2023-01-03	100
Row group 3	T-Shirt	Maria Adams	UK	2023-01-02	500
	Socks	John Doe	USA	2023-01-05	200

Projection and Predicate(s)

Projection = SELECT

Predicate(s) = WHERE

	Column 1	Column 2	Column 3	Column 4	Column 5
	Product	Customer	Country	Date	Sales Amount
Row group 1	Ball	John Doe	USA	2023-01-01	100
	T-Shirt	John Doe	USA	2023-01-02	200
Row group 2	Socks	Maria Adams	UK	2023-01-02	300
	Socks	Antonio Gran	USA	2023-01-05	100
Row group 3	T-Shirt	Maria Adams	UK	2023-01-02	500
	Socks	John Doe	USA	2023-01-05	200

The engine will skip scanning these records!

@DataMozart

Row-Based Storage: 5 Columns + 6 Rows

	Product	Customer	Country	Date	Sales Amount
Row 1	Ball	John Doe	USA	2023-01-01	100
Row 2	T-Shirt	John Doe	USA	2023-01-02	200
Row 3	Socks	Maria Adams	UK	2023-01-01	300
Row 4	Socks	Antonio Grant	USA	2023-01-03	100
Row 5	T-Shirt	Maria Adams	UK	2023-01-02	500
Row 6	Socks	John Doe	USA	2023-01-05	200

Column-Based Storage: 2 Columns + 6 Rows

Column 1	Column 2	Column 3	Column 4	Column 5
Product	Customer	Country	Date	Sales Amount
Ball	John Doe	USA	2023-01-01	100
T-Shirt	John Doe	USA	2023-01-02	200
Socks	Maria Adams	UK	2023-01-01	300
Socks	Antonio Grant	USA	2023-01-03	100
T-Shirt	Maria Adams	UK	2023-01-02	500
Socks	John Doe	USA	2023-01-05	200

Column Storage With Row Groups: 2 Columns + 4 Rows

	Column 1	Column 2	Column 3	Column 4	Column 5
	Product	Customer	Country	Date	Sales Amount
Row group 1	Ball	John Doe	USA	2023-01-01	100
	T-Shirt	John Doe	USA	2023-01-02	200
Row group 2	Socks	Maria Adams	UK	2023-01-02	300
	Socks	Antonio Gran	USA	2023-01-05	100
Row group 3	T-Shirt	Maria Adams	UK	2023-01-02	500
	Socks	John Doe	USA	2023-01-05	200

The engine will skip scanning these records!

@DataMozart

How Parquet “Knows” Which Row Group to Scan?

- **Parquet contains metadata**
  - Data about data
- **Min and max values**
- **Footer**
  - Format version
  - Schema information
  - Column metadata

Performance tip!

- Merge multiple smaller files into one bigger
- A few hundred MBs

Can It Be Better Than This?

Data compression

1. Dictionary encoding

2. Run-Length encoding

Product	Index
Ball	0
T-Shirt	1
Socks	2
Socks	2
T-Shirt	1
Socks	2

Index	Product
0	Ball
1	T-Shirt
2	Socks

Long arm T-shirt with application on the neck

Can It Be Better Than THIS?!

Delta format

Parquet format on steroids!

- ✓ Versioning of Parquet files
- ✓ Stores transaction log
- ✓ Tracks all changes

Two Types of Warehouses in Fabric

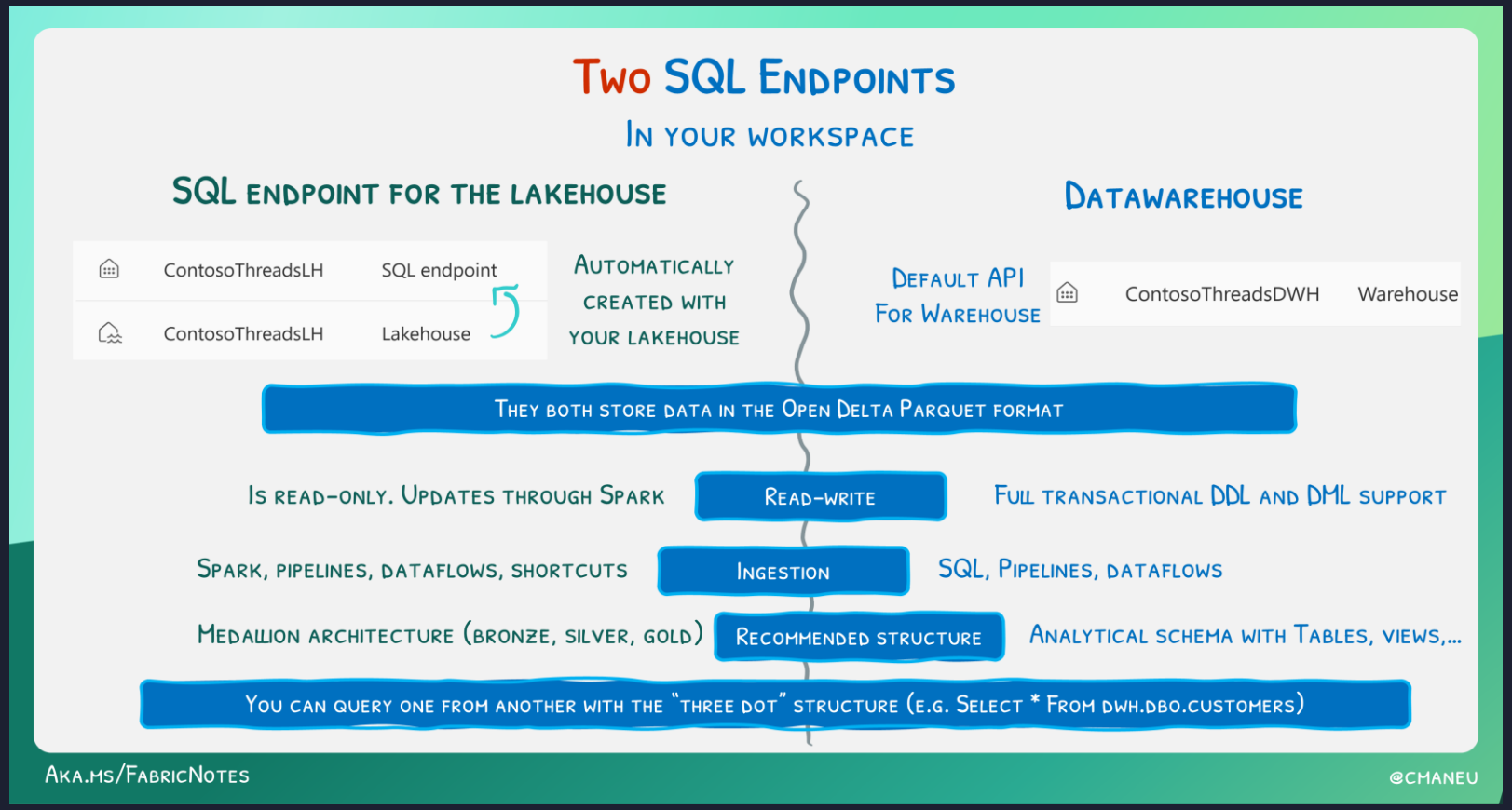
SQL Endpoint of the Lakehouse

- ✓ Automatically generated
- ✓ Supports ONLY read operations
- ✓ Views, inline TVFs, procs…
- ✓ Manage permissions

Synapse Data Warehouse

- ✓ Full transactional support
- ✓ DDL/DML operations
- ✓ Traditional data warehousing workloads

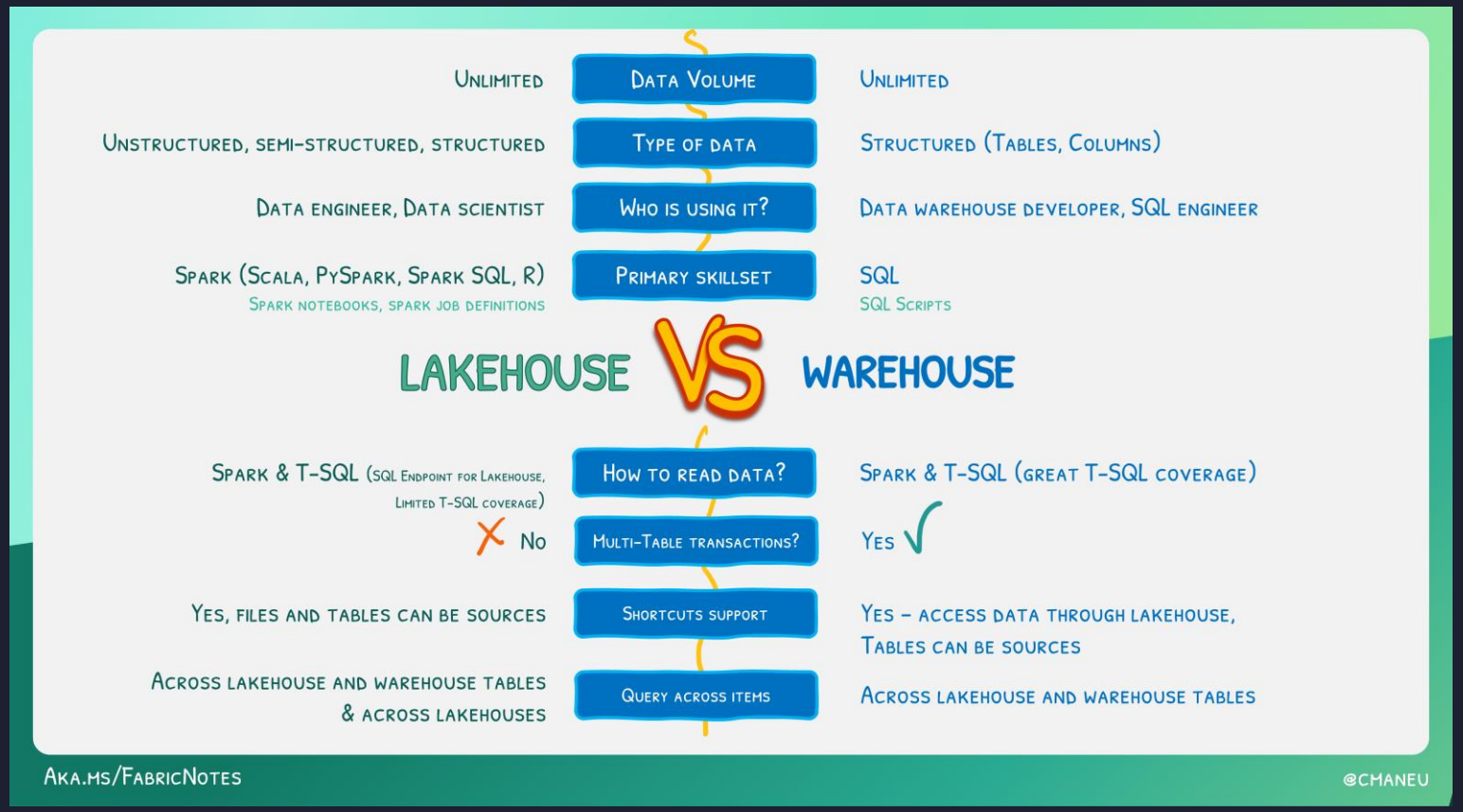
Tale of Two SQL Endpoints in Fabric



Credit: Christopher Maneu

@DataMozart

Do I Need a Warehouse or Lakehouse?



Credit: Christopher Maneu

@DataMozart

What Happens Behind the Scenes?

Efficient Query Optimizer

+

Efficient Query Processor

×

Multi-node distribution (MPP)

What Happens Behind the Scenes?

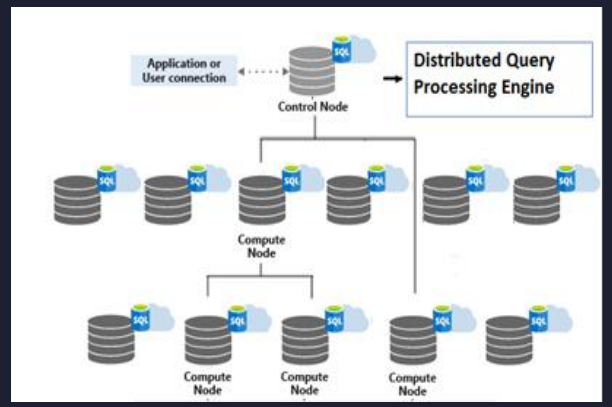
SQL Server Query Optimizer

+

VertiPaq

×

Polaris Engine (Synapse Serverless SQL)



Marriage...

Even happily ever after?

What Happens Behind the Scenes?

Query Processing

1

Parquet

Read Parquet files

Transcoding

2

VertiPaq (Tabular)

CCI (SQL Server)

Fabric Engines (Power BI, SQL, Spark, KQL)

V-Ordering

RLE

3

Parquet

Performance Considerations

- ✓ Statistics (automatic & manual)
- ✓ Data types
- ✓ Reducing query result set (max 10K rows in browser)
- ✓ PK, FK, Unique constraints*

*Not enforced!

Table Constraints

- ✓ PRIMARY KEY supported only with NONCLUSTERED and NOT ENFORCED
- ✓ UNIQUE supported only with NONCLUSTERED and NOT ENFORCED
- ✓ FOREIGN KEY with NOT ENFORCED
- ✓ No default constraints!

```
CREATE TABLE PrimaryKeyTable (c1 INT NOT NULL, c2 INT);

ALTER TABLE PrimaryKeyTable ADD CONSTRAINT PK_PrimaryKeyTable PRIMARY KEY NONCLUSTERED (c1) NOT ENFORCED;
```

I’m a SQL Server Ninja – What’s (not) in for me?

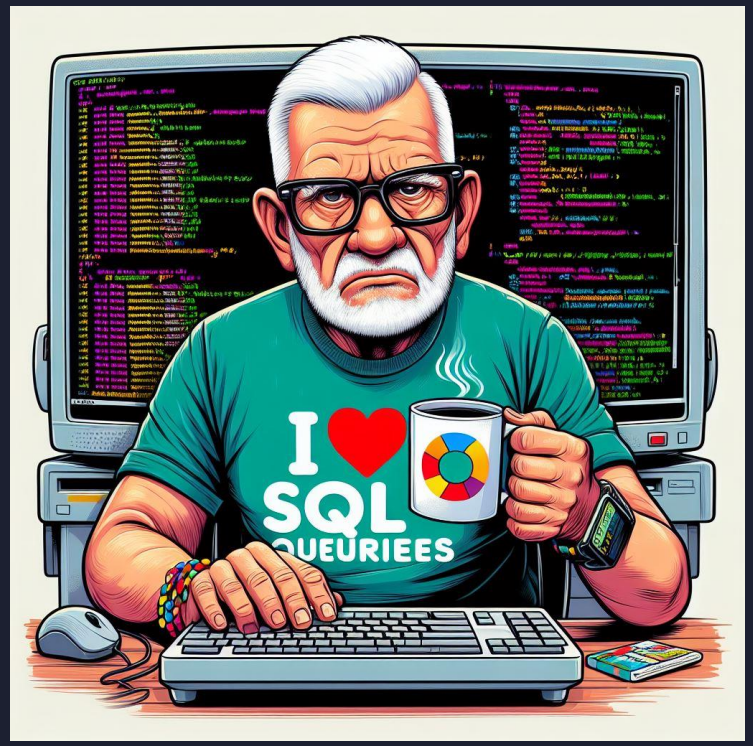


Table limitations

- Computed columns
- Indexed views
- Partitioned tables
- Indexes
- Sequence
- Temp tables
- Triggers
- User-defined data types

T-SQL limitations

- ALTER TABLE ADD/ALTER COLUMN
- Identity columns
- Hints
- MERGE
- Recursive queries
- SET ROWCOUNT
- SET TRANSACTION ISOLATION LEVEL
- TRUNCATE

I’m a Synapse SQL Ninja – What’s (not) in for me?

T-SQL limitations

- OPENROWSET
- PREDICT

Ingest Data Into Warehouse

COPY (T-SQL)

- ✓ Code-rich data ingestion
- ✓ Highest possible throughput
- ✓ Data ingestion as part of T-SQL logic

Dataflows Gen2

- ✓ Low-code, no-code solution
- ✓ Custom transformations
- ✓ Power Query-like experience

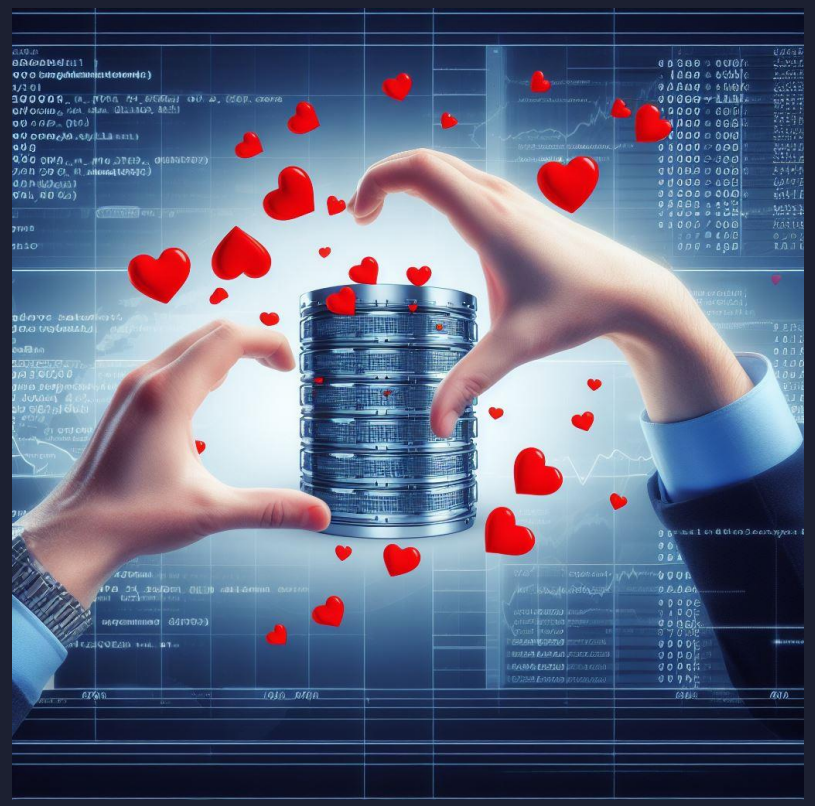
Pipelines

- ✓ Low-code, no-code solution
- ✓ Repeatedly run workflows
- ✓ Large volumes of data

Cross-warehouse

- ✓ Code-rich
- ✓ INSERT...SELECT
- ✓ SELECT INTO
- ✓ CTAS

From Fabric DWH with love…



Feature innovations…

- Mirroring
- Table clone
- Automatic data compaction

Mirroring in Microsoft Fabric

- ✓ Overcomes the “limitation” of Shortcuts to read Delta-only
- ✓ Connects to a “wonderful world” of proprietary formats and databases

Enable some kind of CDC on the source database!

- ✓ Fabric transforms proprietary format to Delta on the fly and stores in OneLake
- ✓ Entire database or specific tables
- ✓ All features supported, including cross-joining and Direct Lake!

Mirroring in Microsoft Fabric

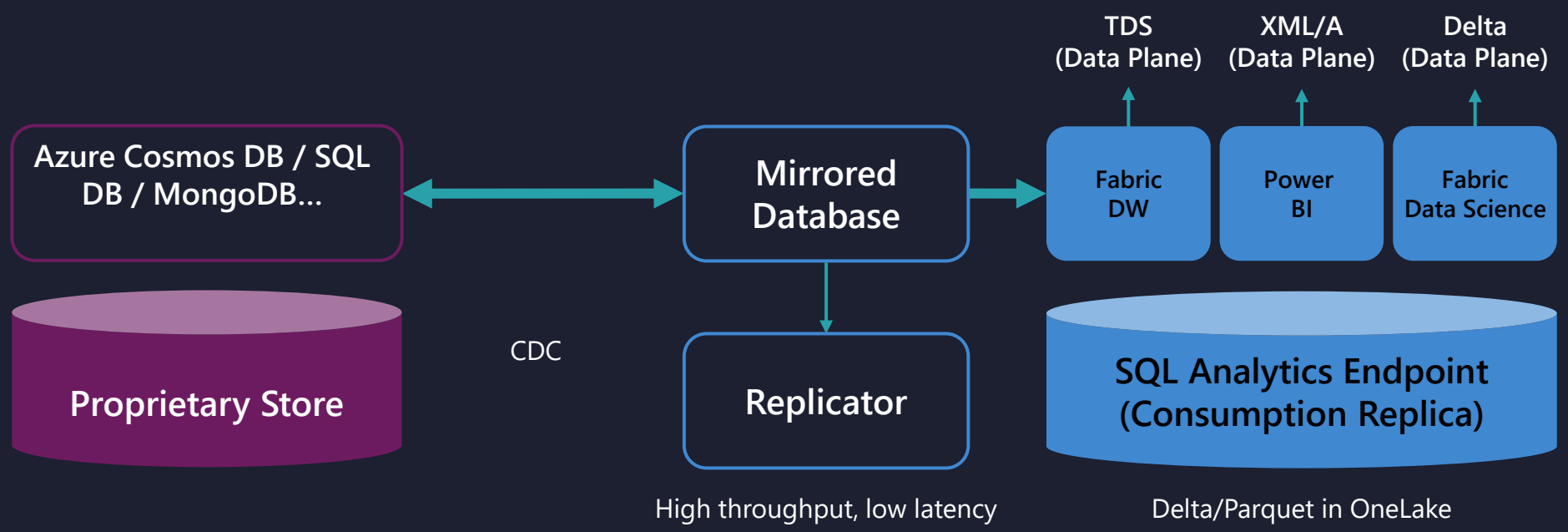


Illustration taken from Microsoft's presentation at Ignite 2023

Mirroring Pricing

1 TB per CU for FREE!

- F2 = 2 TB, F8 = 8 TB, F64 = 64 TB…
- ***If you pause the capacity, you pay for storage then!***

Table Cloning in Microsoft Fabric

- ✓ Creates a table replica with metadata
- ✓ Physical data still stored in OneLake – clone is just referencing it

  - *Test/Dev environment*
  - *Reporting/ML workloads*
  - *Point-in-time data for compliance requirements*
  - *Data recovery*

- ✓ Limitations
  - No clones across warehouses
  - No Fabric Lakehouse SQL Endpoint
  - No RLS/DDM inheritance

Automatic Data Compaction

- ✓ Parquet file can't be changed
- ✓ 1000s or 10000s of small Parquet files
- ✓ Reading metadata is slow and inefficient!

- ➢ *Rewrites many smaller Parquet files into a few bigger ones*
- ➢ *Removes deleted rows*

What's Coming…

Fabric Release Plan – Data Warehouse

Feature	ETA
Case insensitive collation	Q3 2024
TRUNCATE	Q3 2024
Result set caching	Q3 2024
Nested CTEs	Q3 2024
Notebook integration	Q3 2024
Mirroring GA	Q4 2024
Copilot (Public preview)	Q2 2024

Thank you

Nikola Ilic

@DataMozart

www.data-mozart.com

www.learn.data-mozart.com