

EVOLUTION OF DATA LAKEHOUSES



**ADVANCING
ANALYTICS**



Microsoft
Partner



Gold Data Analytics
Gold Data Platform
Silver Cloud Platform



databricks



Microsoft®
Most Valuable
Professional



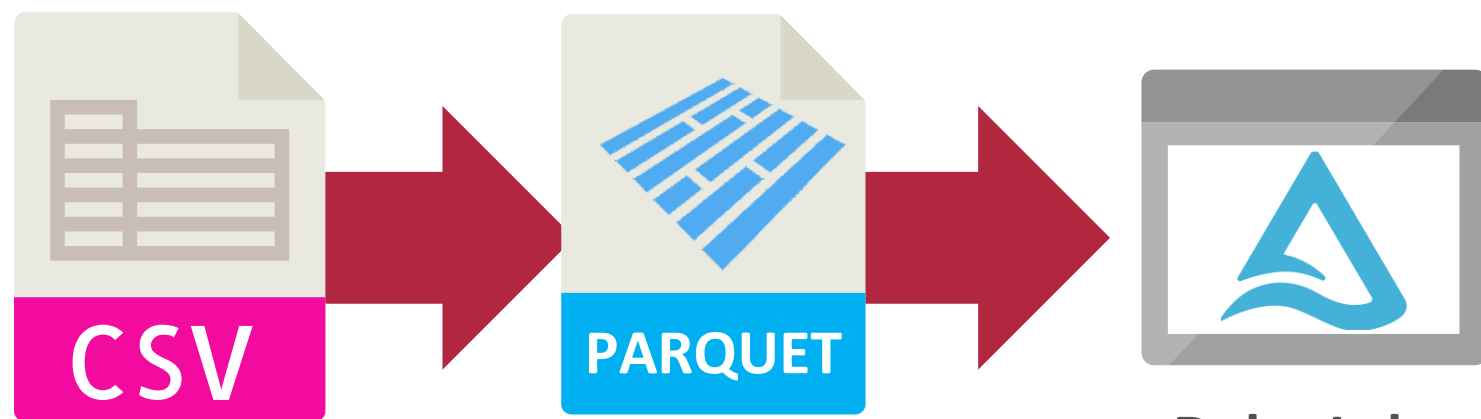
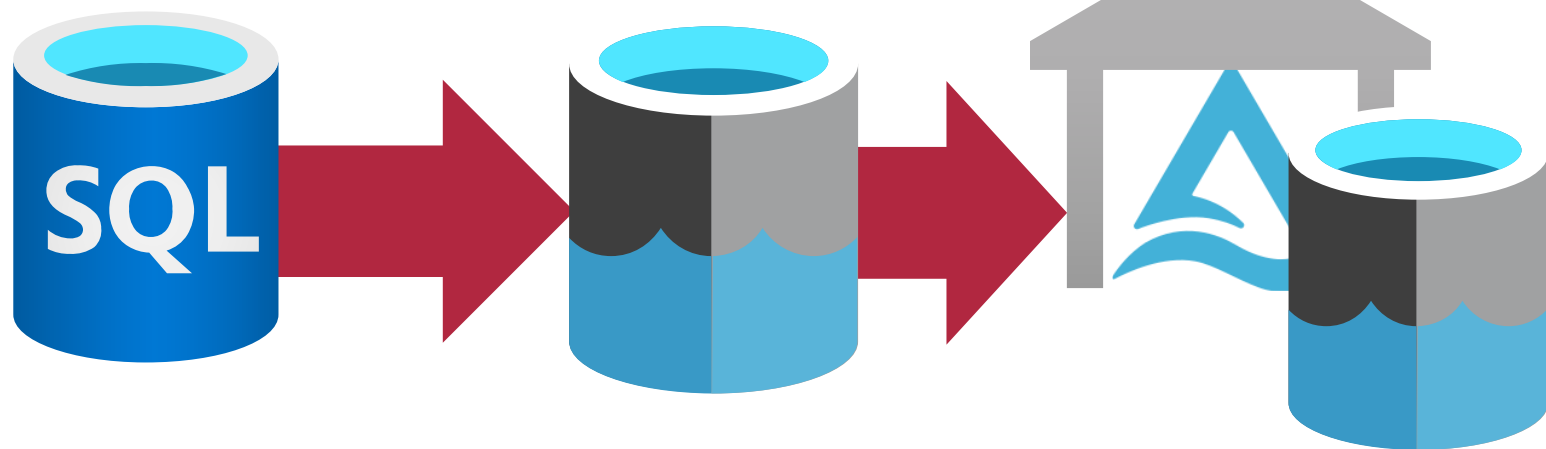
@ADVANCINGANALYTICS



@ADVANALYTICSUK

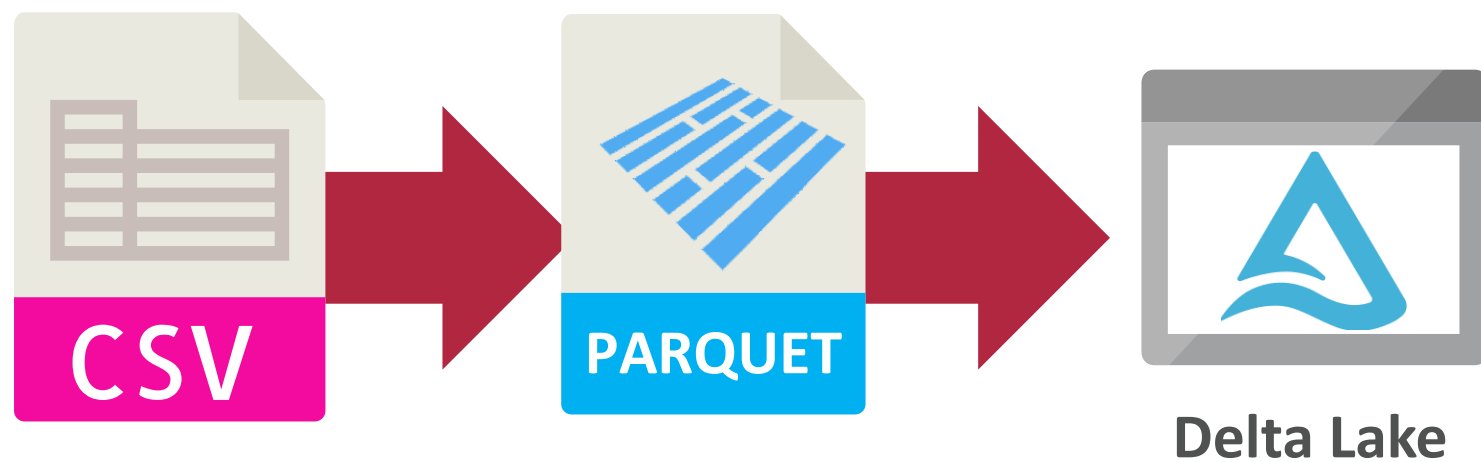
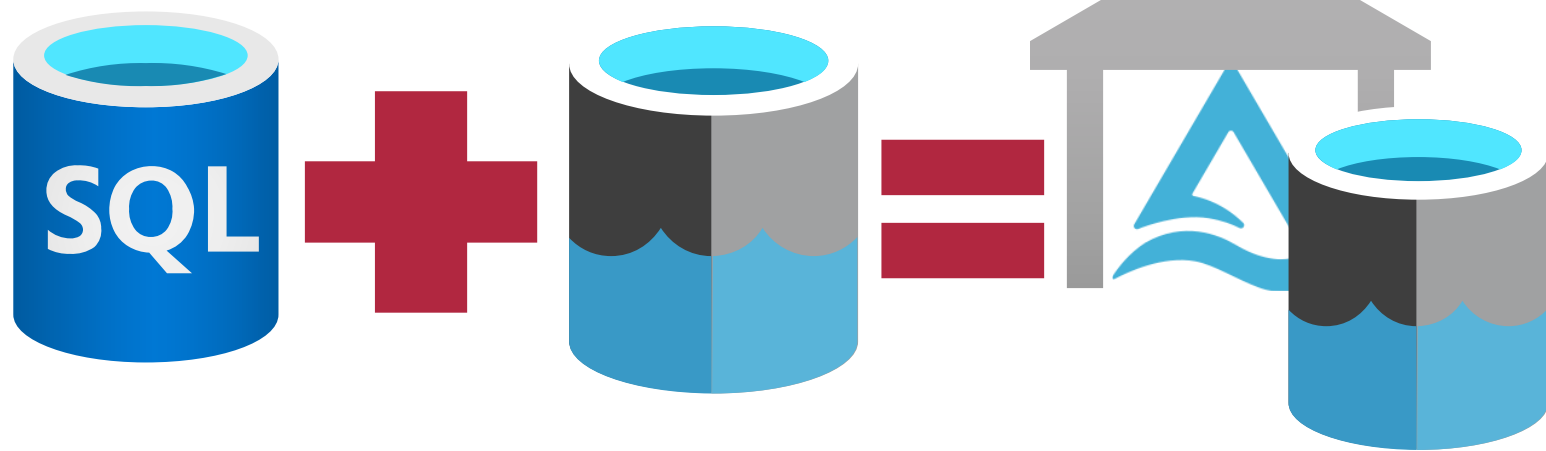


/ADVANCING ANALYTICS



Delta Lake





THE DATA LAKEHOUSE



- Structured
- Governed
- Familiar
- Fast (for specific jobs)

- Flexible
- Cheap
- Scalable
- Fast (for specific jobs)



THE DATA LAKEHOUSE

- Structured
- Governed
- Familiar
- Fast
- Flexible
- Cheap
- Scalable





www.advancinganalytics.co.uk

SO WHAT'S WRONG WITH PARQUET?



@ADVANCINGANALYTICS

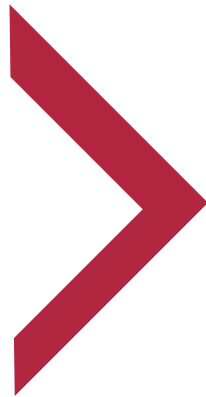


@ADVANALYTICSUK

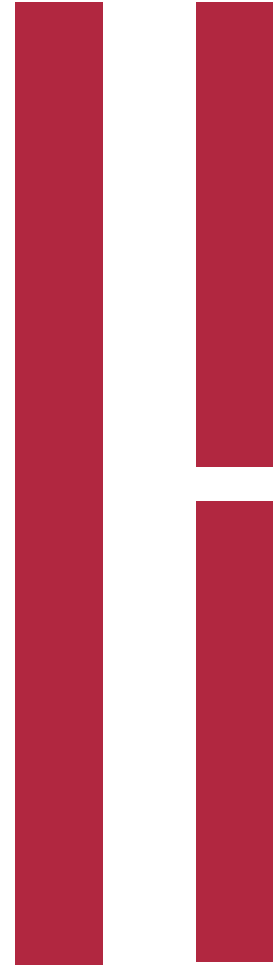


/ADVANCING ANALYTICS

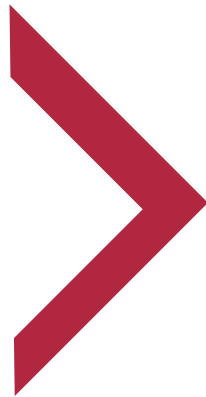
THE PROBLEMS OF PARQUET



Small files have a heavy performance impact

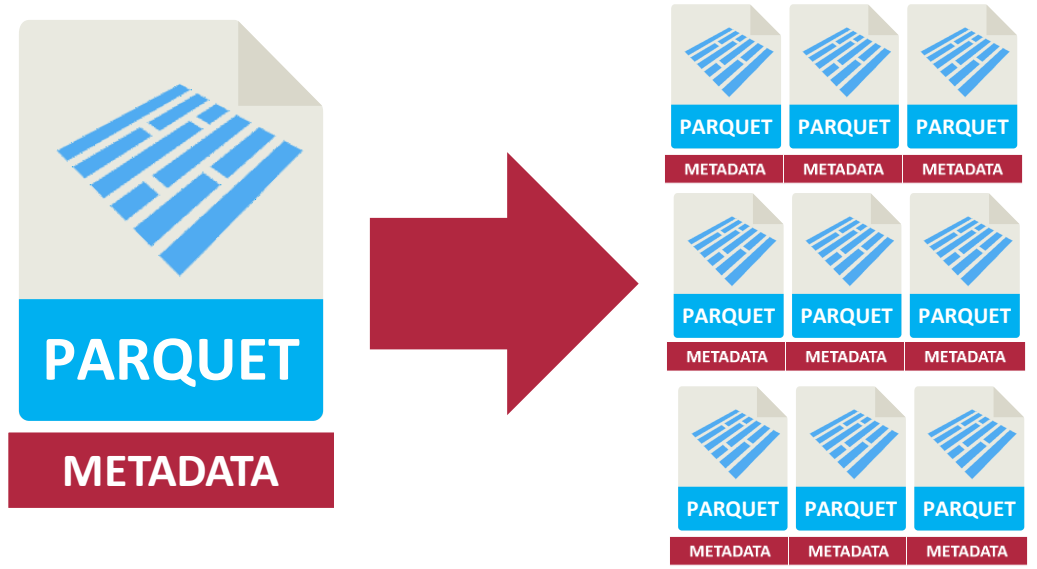


THE PROBLEMS OF PARQUET



Small files have a heavy performance impact

METADATA IN DATA FILES



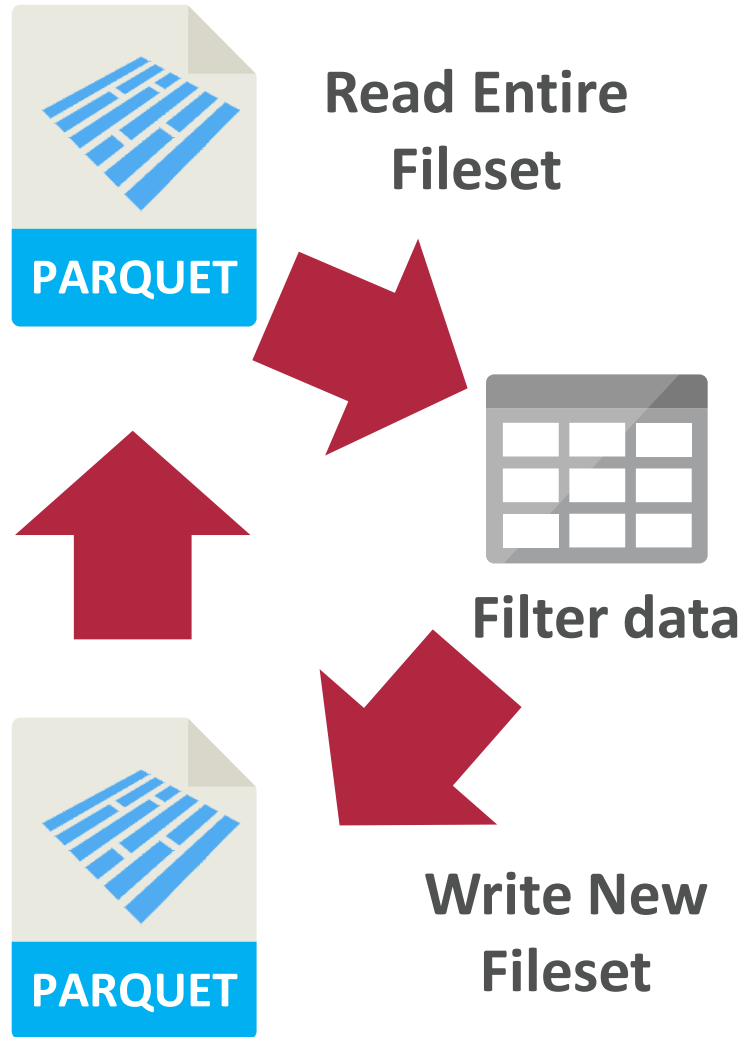
Metadata scan = reading all files



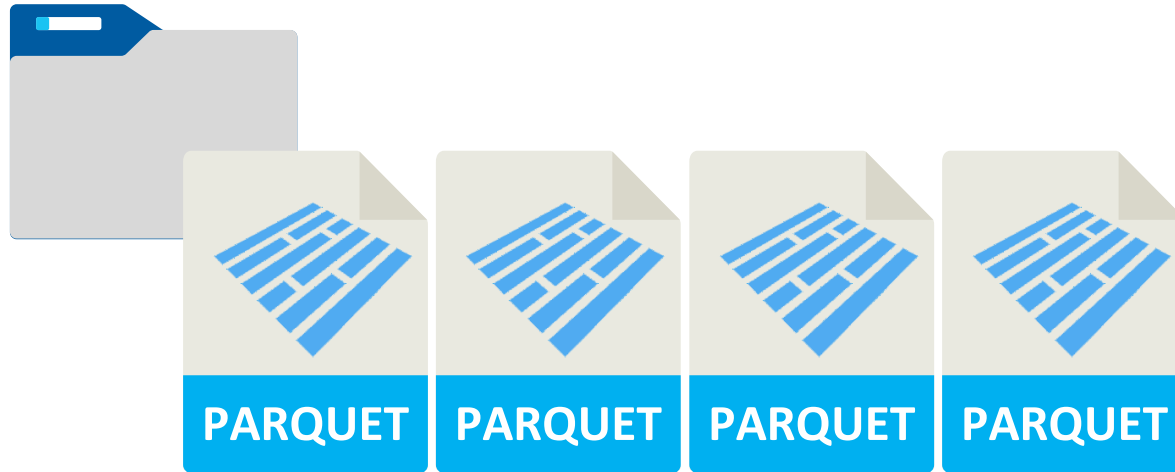
OPERATIONAL COMPLEXITY



DELETE FROM
MyTable WHERE...



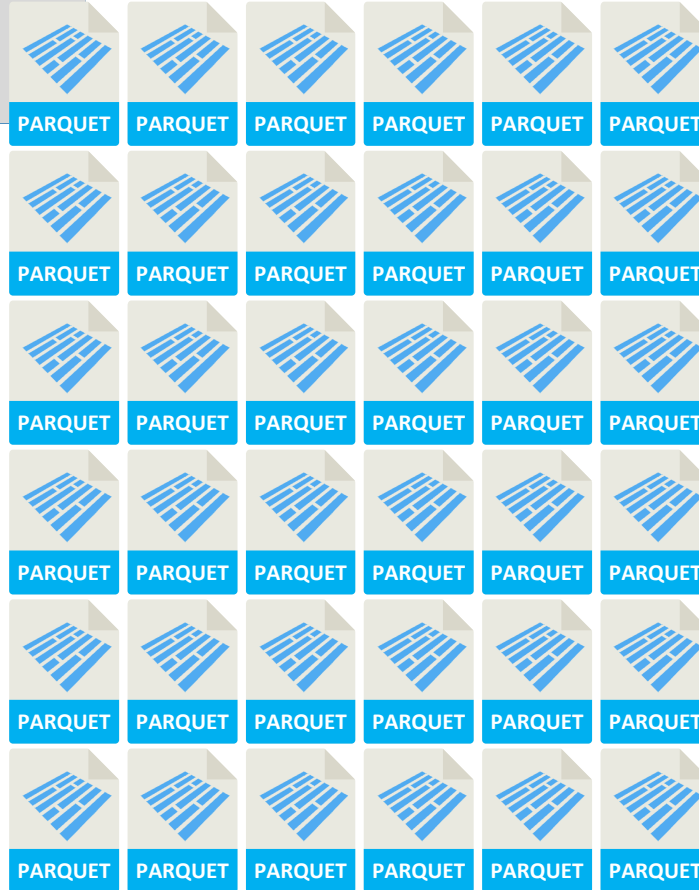
AUDITING & RECOVERY



DELETE FROM MyTable WHERE ID = 132



NO INDEXES



**SELECT * FROM
MyTable WHERE...**



HENCE WHY WE NEED...



- Structured
- Governed
- Familiar
- Fast (for specific jobs)





www.advancinganalytics.co.uk

WHAT IS DELTA?



@ADVANCINGANALYTICS



@ADVANALYTICSUK



/ADVANCING ANALYTICS

WHAT IS DELTA?

"Databricks Delta is a unified data management system that brings reliability and performance(**10-100x** faster than Spark on Parquet) to cloud data lakes.

Delta's core abstraction is a Spark table with built-in reliability and performance optimizations."



WHAT IS DELTA?

*Delta Lake is an **optimised,**
managed format for organising &
working with **Parquet** files*

"It's Parquet, but better"





www.advancinganalytics.co.uk

SOUNDS FANCY... HOW DOES IT WORK?



@ADVANCINGANALYTICS

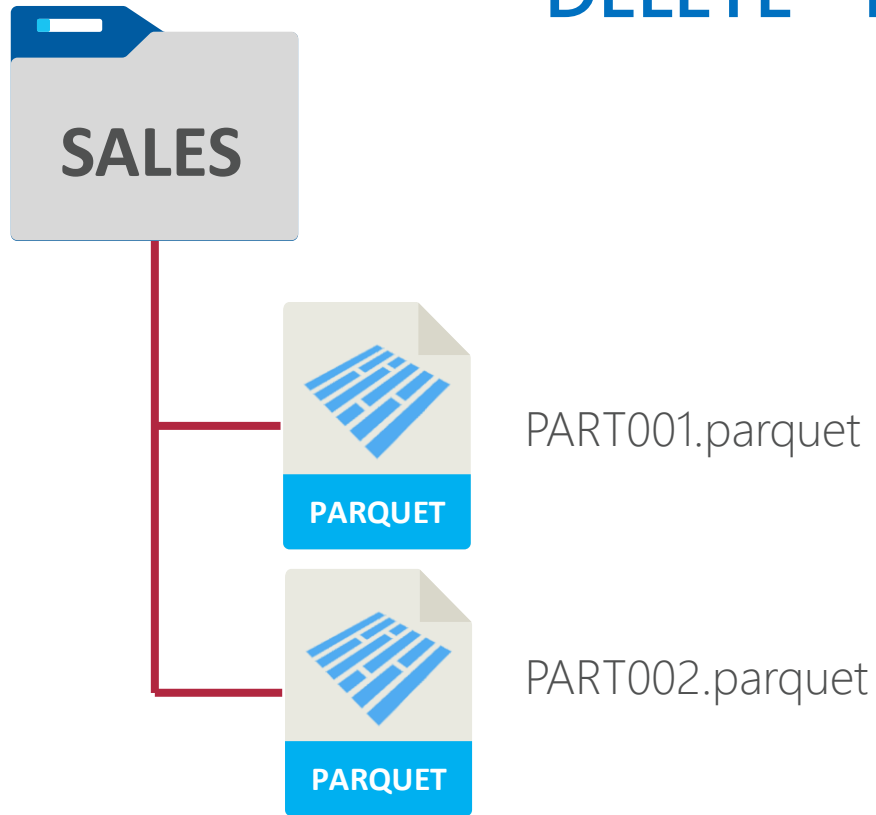


@ADVANALYTICSUK

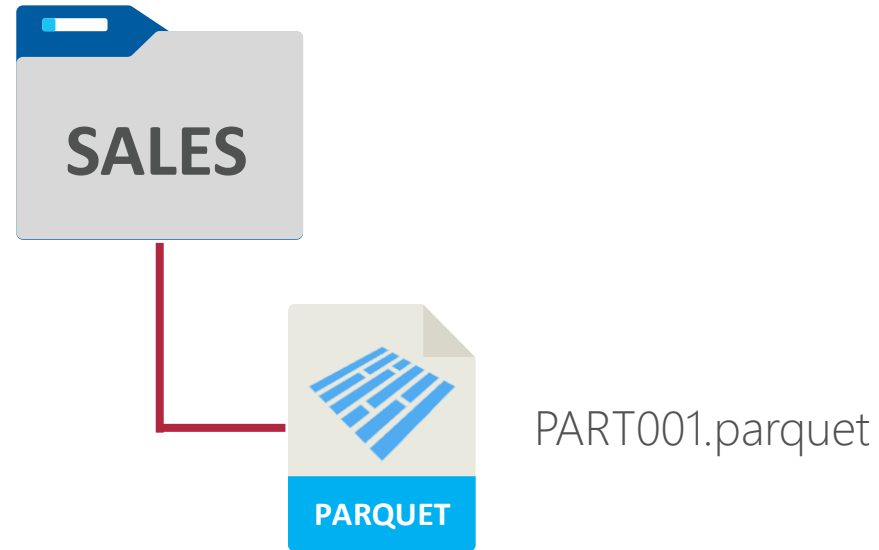


/ADVANCING ANALYTICS

BEFORE DELTA



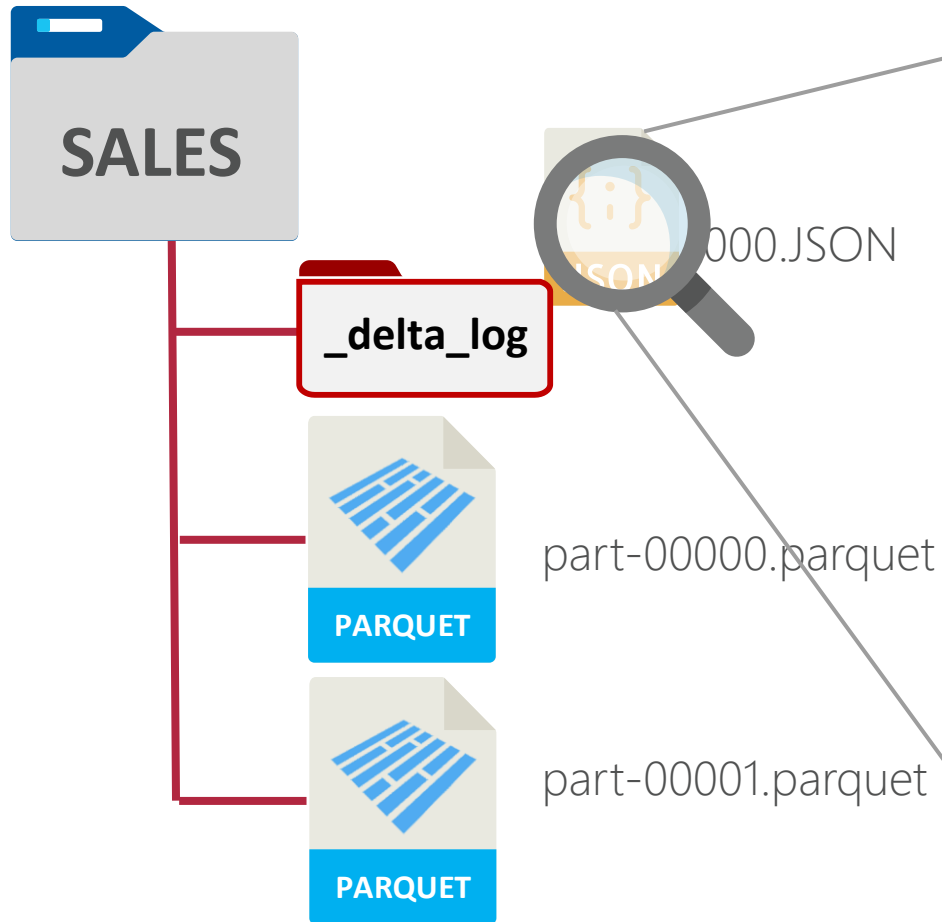
DELETE * **FROM** SALES **WHERE** Segment = 3



Only way to delete is to replace the existing files with a new file containing the non-deleted data



BUT WHAT ACTUALLY IS IT? - WITH DELTA:

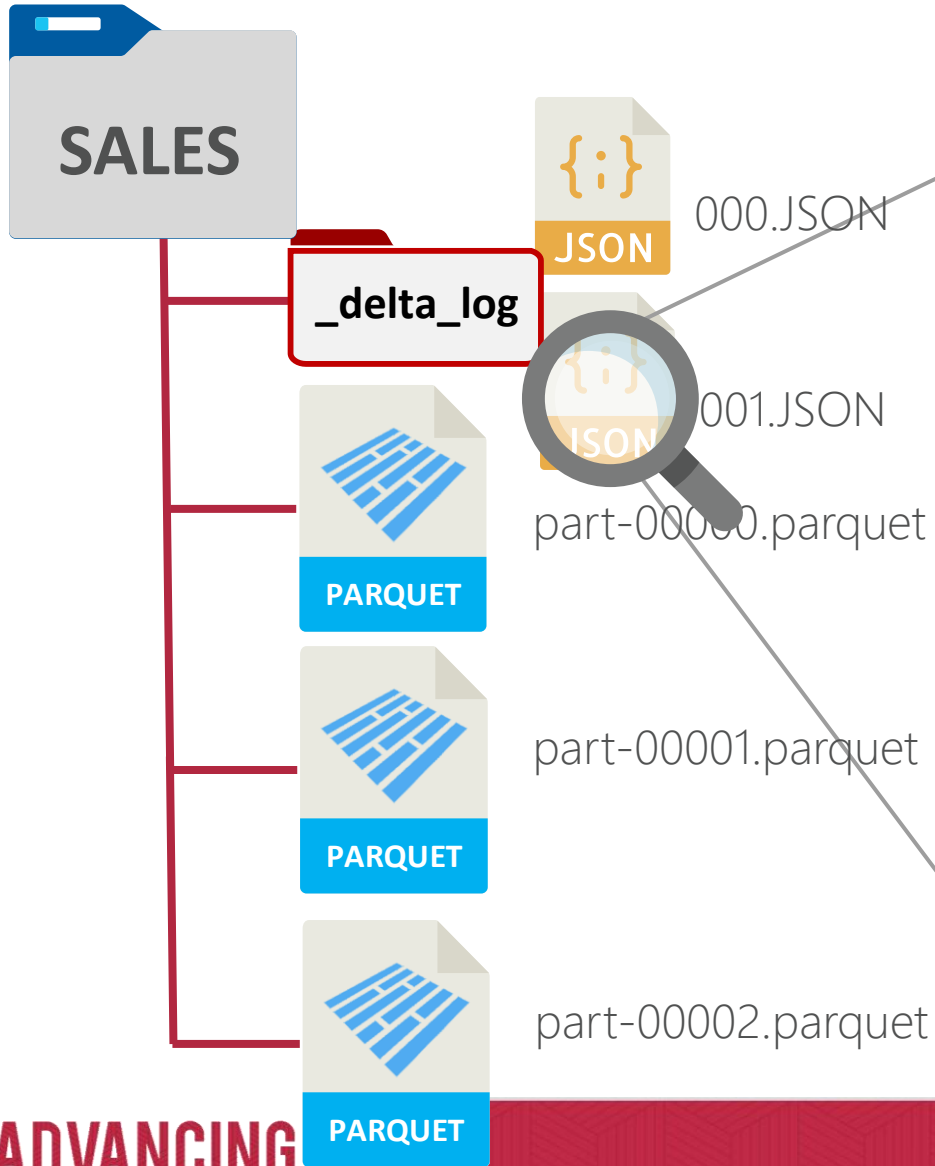


```
...  
{  
  "add": {  
    "path": "part-00000.parquet",  
    "partitionValues": {},  
    "size": 255520,  
    "modificationTime": 1572823237000,  
    "dataChange": true,  
    "stats": [...]},  
  "add": {  
    "path": "part-00001.parquet",  
    "partitionValues": {},  
    "size": 242520,  
    "modificationTime": 1572823237000,  
    "dataChange": true,  
    "stats": [...]}  
}
```



BUT WHAT ACTUALLY IS IT? - WITH DELTA:

DELETE * FROM SALES WHERE
Segment = 3



```
...
{
  "add": {
    "path": "part-00002.parquet",
    "partitionValues": {},
    "size": 255520,
    "modificationTime": 1572823237000,
    "dataChange": true,
    "stats": [...]},
  "delete": {
    "path": "part-00000.parquet",
    "modificationTime": 1572823237000,
    "dataChange": true},
  "delete": {
    "path": "part-00001.parquet",
    "modificationTime": 1572823237000,
    "dataChange": true}
}
```



SELECT * FROM SALES



000.JSON



001.JSON



part-00000.parquet



part-00001.parquet



part-00002.parquet





000.JSON



001.JSON



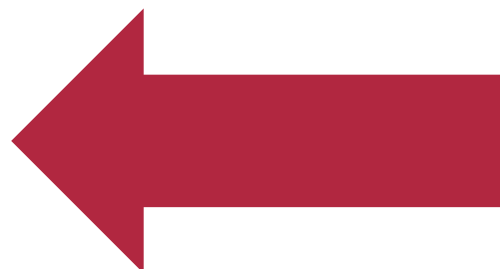
part-00000.parquet



part-00001.parquet



part-00002.parquet



SELECT * FROM SALES
TIMESTAMP AS OF
"2018-10-18T22:15:12.013Z"



DEMO: WORKING WITH DELTA

- Creating a Delta Table
- The Delta Transaction Log
- Updating Delta Tables





www.advancinganalytics.co.uk

KILLER DELTA FEATURES



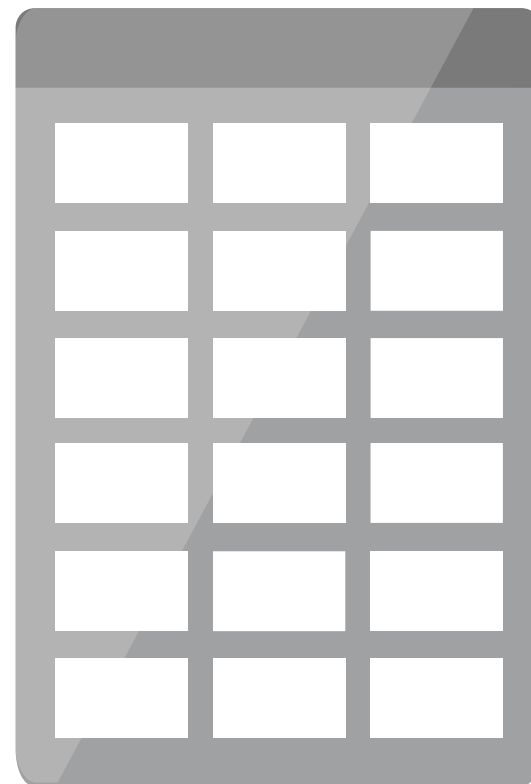
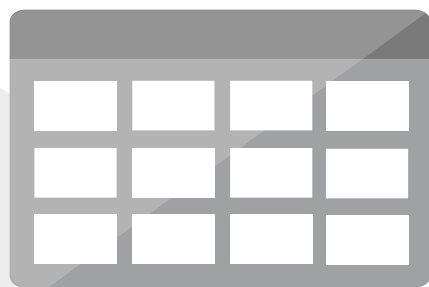
@ADVANCINGANALYTICS



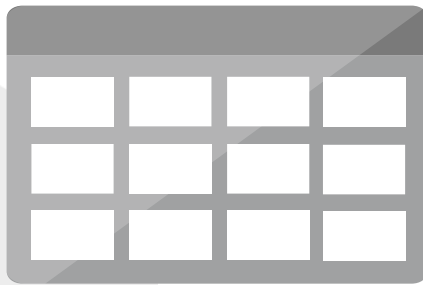
@ADVANALYTICSUK



/ADVANCING ANALYTICS



Delta supports minor
schema drift



DELTA MERGE



```
MERGE INTO <target_table>
USING <source_table>
ON <merge_condition>
[ WHEN MATCHED [ AND <condition> ] THEN <matched_action> ]
[ WHEN MATCHED [ AND <condition> ] THEN <matched_action> ]
[ WHEN NOT MATCHED [ AND <condition> ] THEN <not_matched_action> ]
```

where

```
<matched_action> =
    DELETE |
    UPDATE SET * |
    UPDATE SET column1 = value1 [, column2 = value2 ...]
```

```
<not_matched_action> =
    INSERT * |
    INSERT (column1 [, column2 ...]) VALUES (value1 [, value2 ...])
```



DEMO: DELTA UPDATES

- Schema Drift
- Merge into a Delta Table
- TIME TRAVEL



www.advancinganalytics.co.uk

MANAGING DELTA TABLES



@ADVANCINGANALYTICS

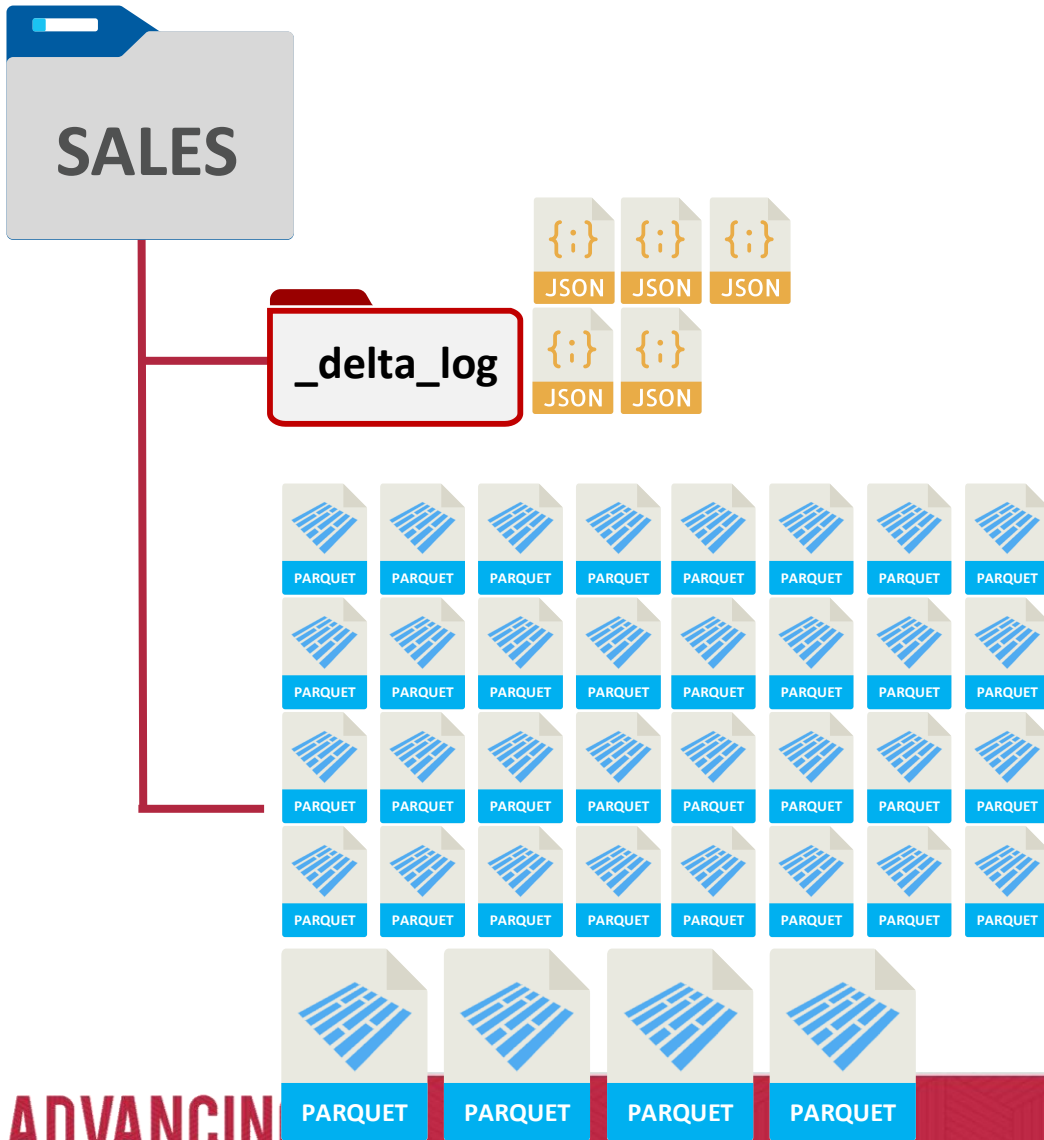


@ADVANALYTICSUK



/ADVANCING ANALYTICS

SMALL/INCREMENTAL UPDATES



The **Optimize** command compacts small files into larger, better compressed files

This is treated like all other updates, files are NOT deleted



Z-ORDERING

Whilst it sounds complicated, we can summarize Z-ordering as:

"Sort the data on specific columns before writing to files, to optimize data skipping"

```
--Optimize an entire table  
OPTIMIZE [database].[table] ZORDER BY [ColumnName]
```



Z-ORDERING EXPLAINED

"select count() from Employees
where Name = 'Brad'"*

A	1	Bob
A	2	Fred



A	1	Andy
A	2	Tom



A	1	Brad
A	2	Tim



A	1	Dan
A	2	Jan



In this example, we have
not ordered our small files,
so data skipping only hits
occasional lucky successes



Z-ORDERING EXPLAINED

A	1	Bob
A	2	Fred

A	1	Andy
A	2	Tom

A	1	Brad
A	2	Tim

A	2	Dan
A	1	Jan



ZOrder
by
Name

A	1	Andy
A	1	Bob
A	1	Brad
A	2	Dan

A	2	Fred
A	1	Jan
A	2	Tim
A	2	Tom



"select count() from
Employees where
Name = 'Brad'"*



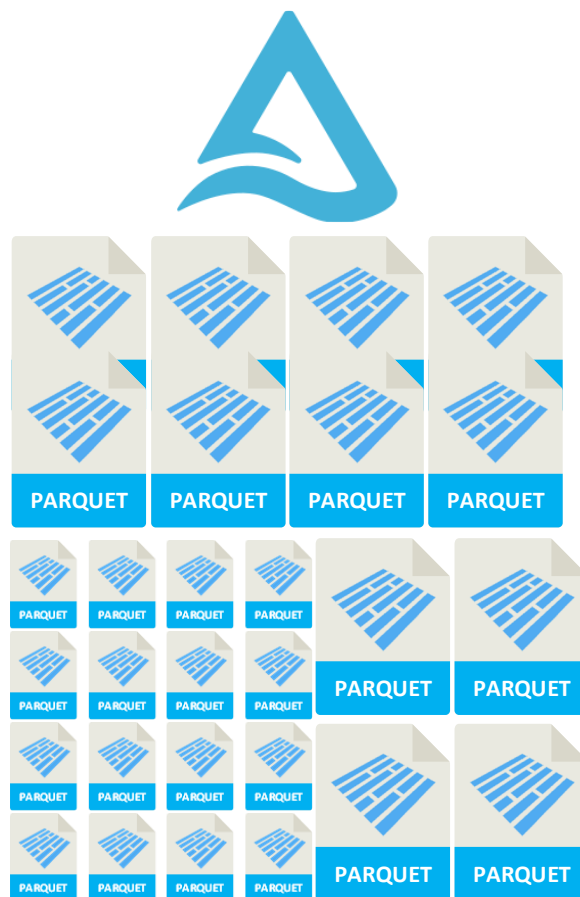
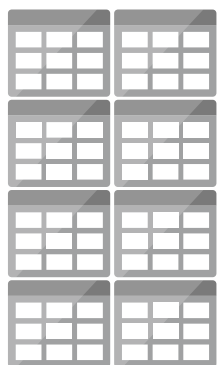


```
--Enable AutoOptimize
```

```
set spark.databricks.delta.properties.defaults.autoOptimize.optimizeWrite = true;  
set spark.databricks.delta.properties.defaults.autoOptimize.autoCompact = true;
```

Optimized Writes - Change the query plan to attempt to write out files of at least 128Mb (this is less than the 1Gb default that Optimize uses!)

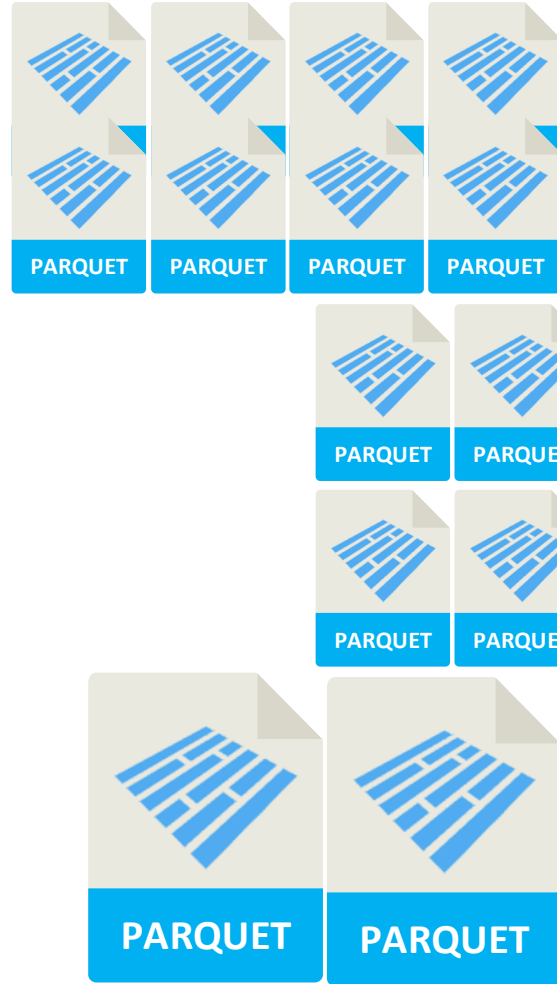
Auto Compaction - Run a lightweight optimize job after the write has finished, looking for further file compaction opportunities, again with 128mb



Optimised Writes

Auto Compaction





Optimize





DELTA MANAGEMENT

- View Delta Metadata
- Optimize a Delta Table



www.advancinganalytics.co.uk

THE DATA LAKEHOUSE



@ADVANCINGANALYTICS



@ADVANALYTICSUK



/ADVANCING ANALYTICS

THE DATA LAKEHOUSE

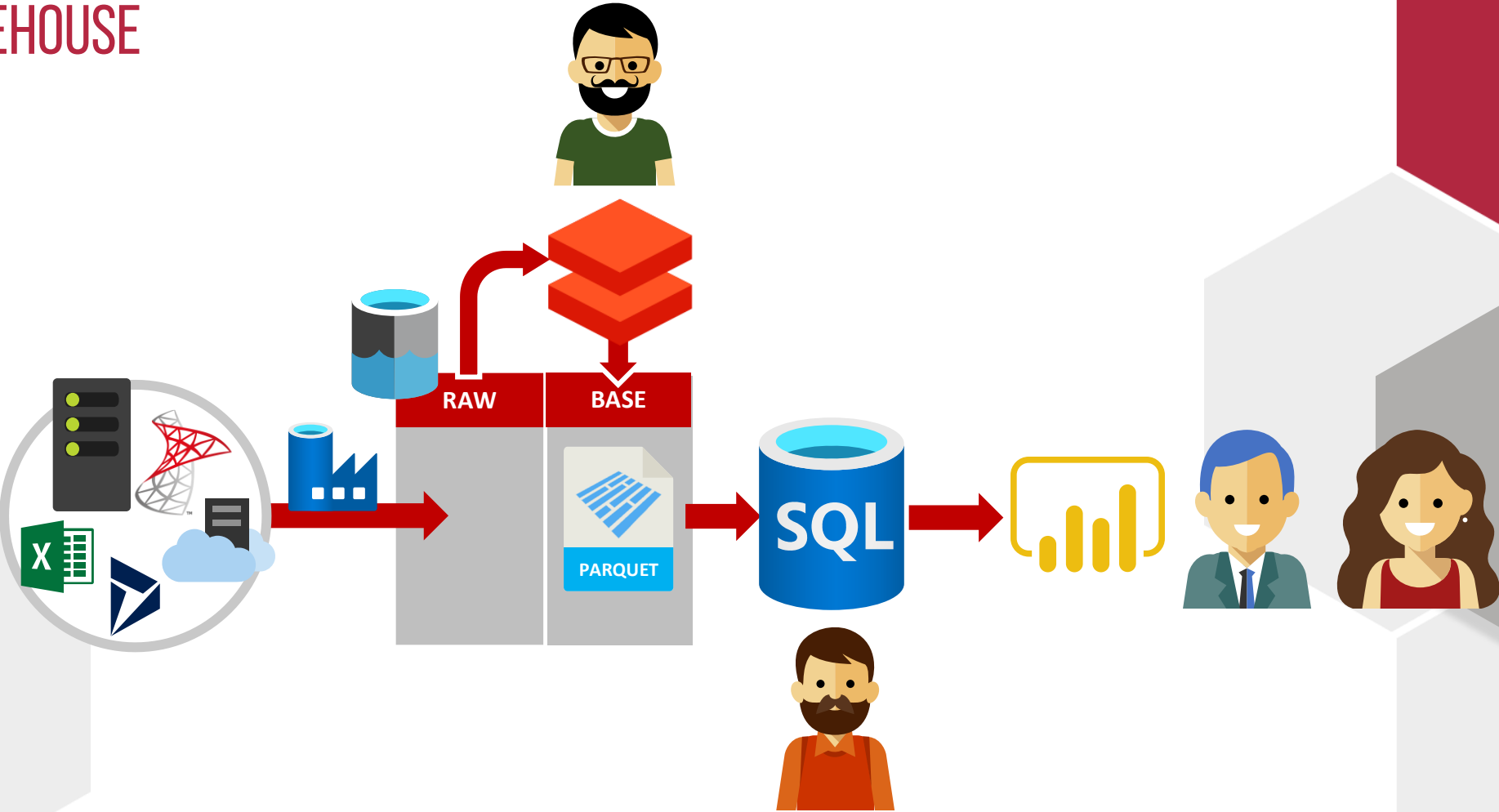
- Structured
- Governed
- Familiar
- Fast
- Flexible
- Cheap
- Scalable



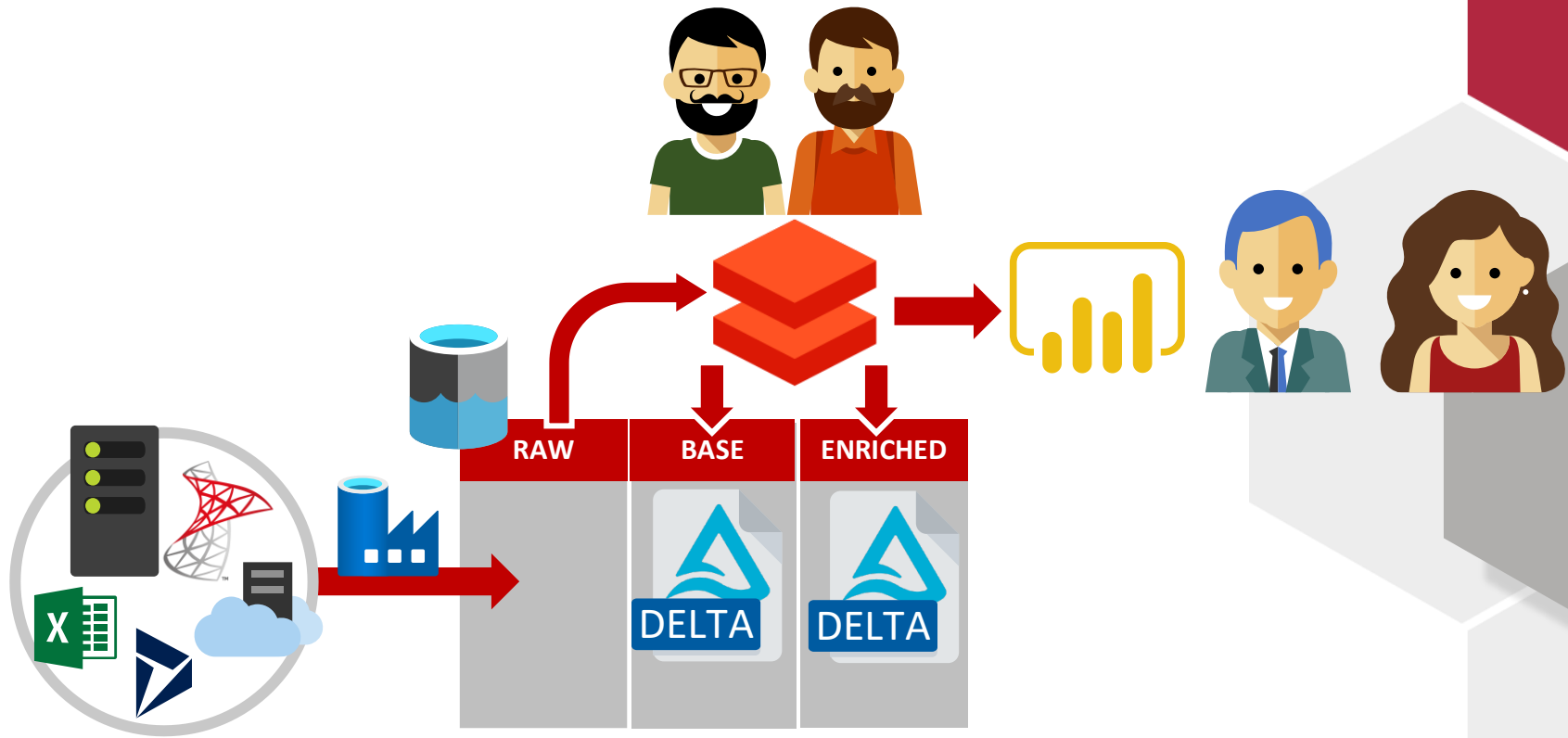
- Small Files
- Operational Complexity
- Metadata
- Indexing



MODERN WAREHOUSE



THE DATA LAKEHOUSE



THANKS FOR LISTENING



Twitter: @MrSiWhiteley

youtube.com/c/AdvancingAnalytics

AdvancingAnalytics.co.uk



@ADVANCINGANALYTICS



@ADVANALYTICSUK



/ADVANCING ANALYTICS