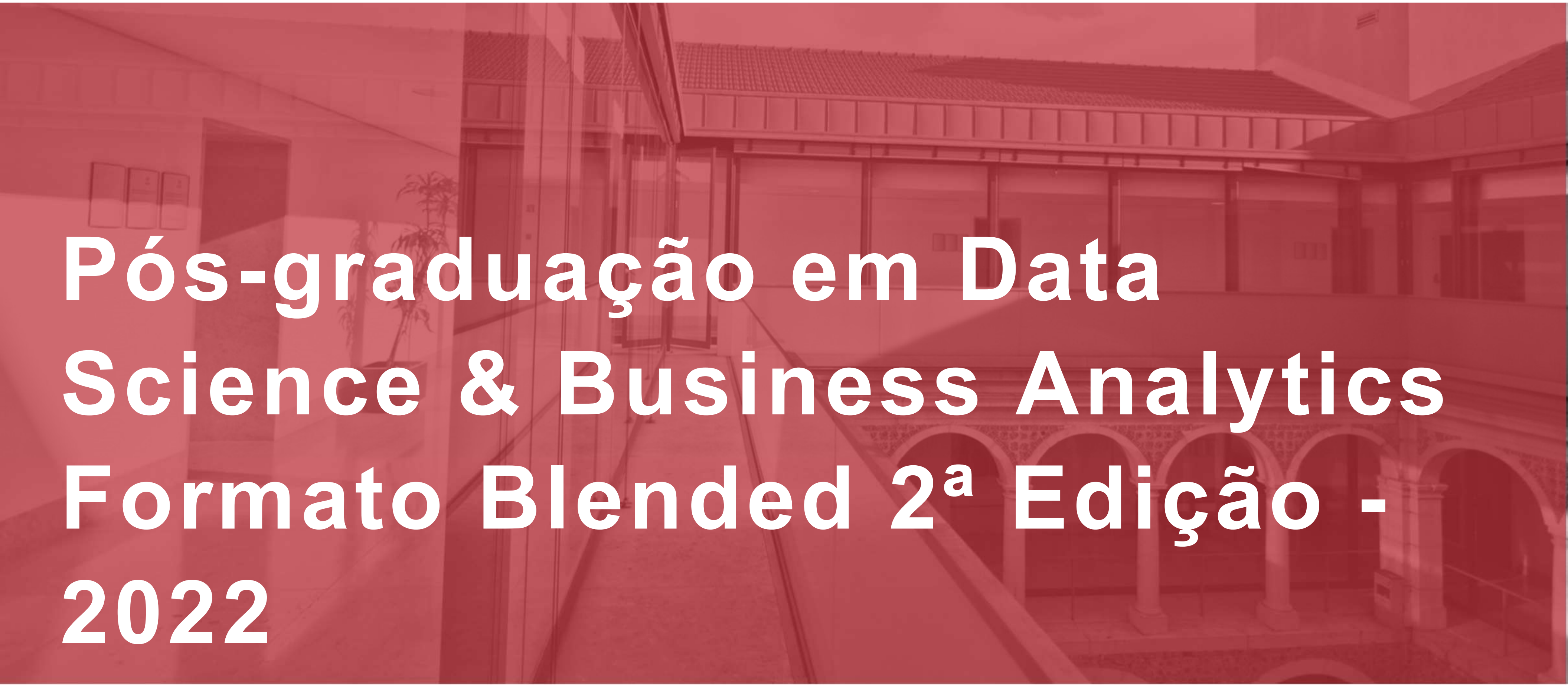


Pós-graduação em Data Science & Business Analytics Formato Blended 2ª Edição - 2022

ISEG Executive Education

Data Warehousing

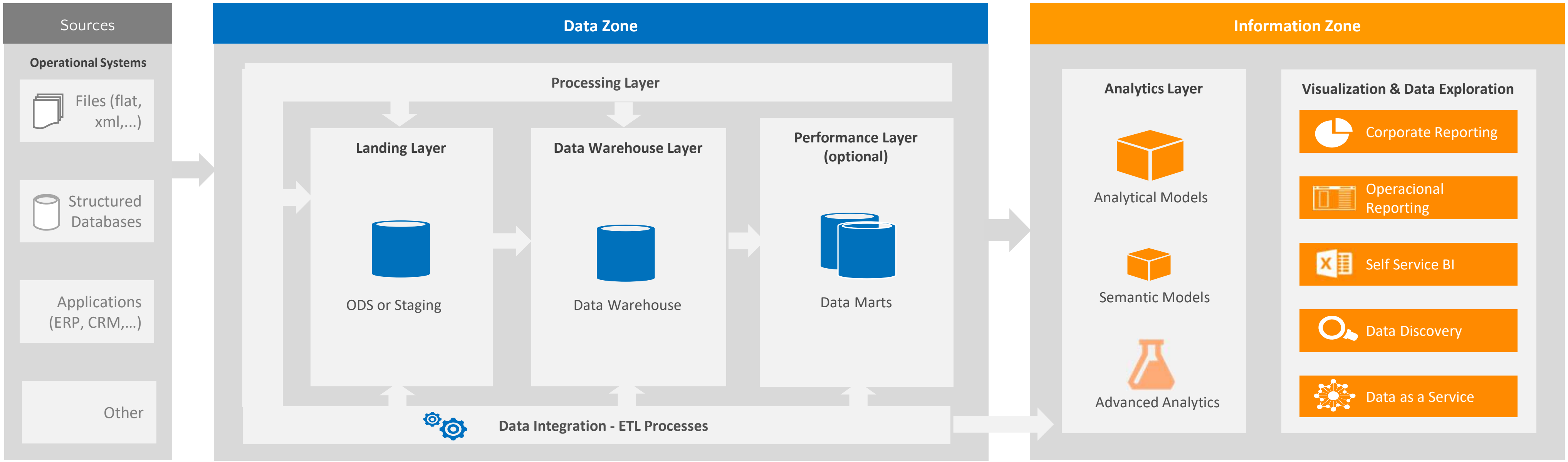
Artur Vieira

Índice

- Traditional Data Warehouse Architectures
- Data Integration
- SQL Server Integration Services
- Azure Data Factory

Executive Education

Traditional Data Warehouse Architecture

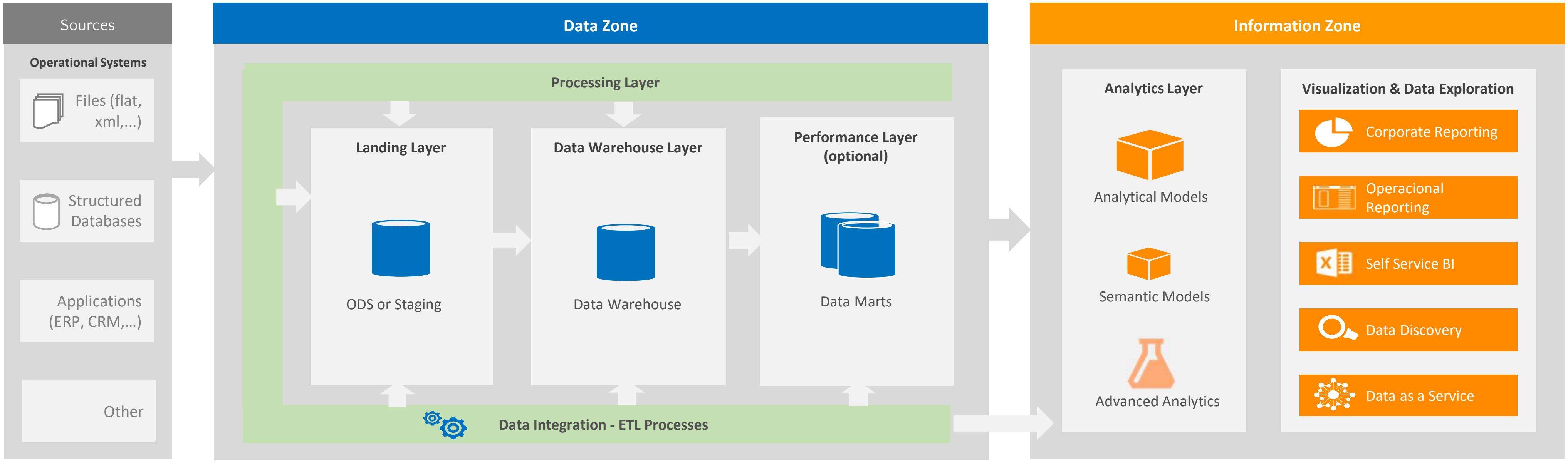


ODS – operational data store

Windows iOS Android HTML5

Executive Education

Traditional Data Warehouse Architecture



Sources

Operational Systems

- Files (flat, xml,...)
- Structured Databases
- Applications (ERP, CRM,...)
- Other

Data Zone

Processing Layer

- Landing Layer: ODS or Staging
- Data Warehouse Layer: Data Warehouse
- Performance Layer (optional): Data Marts

Data Integration - ETL Processes

Information Zone

Analytics Layer

- Analytical Models
- Semantic Models
- Advanced Analytics

Visualization & Data Exploration

- Corporate Reporting
- Operacional Reporting
- Self Service BI
- Data Discovery
- Data as a Service

ODS – operational data store

Windows iOS Android HTML5

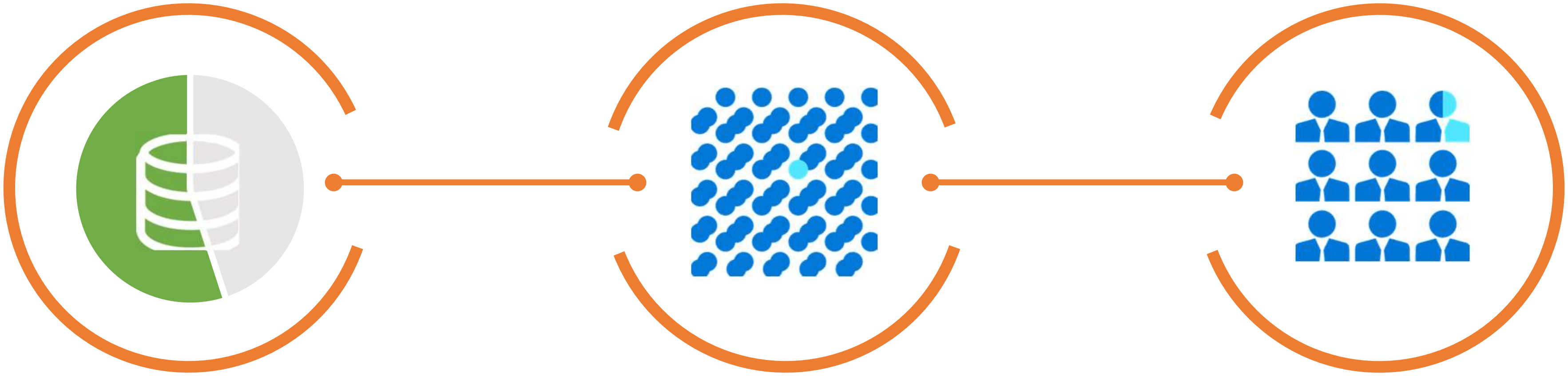
Data Integration

Data integration is a process in which heterogeneous data is retrieved and combined as an incorporated form and structure.

- **Extract, Transform and load (ETL)**
- Integrate structured and unstructured data
- Multiple sources
- Multiple destinations
- Data Modeling
- Data profiling
- Data Cleansing, Data Merging / Data Enrichment

Executive Education

Access to data remains top issue



Less than half of structured data is actively used in decision-making

Less than 1% of the unstructured data is analyzed or used

97%* of executives find data silos harmful to their organization

*83% of executives confirm their organizations have data silos

Harvard Business Review, 2017:
https://hbr.org/2017/05/whats-your-data-strategy

American Management Association
2017 survey

There are barriers to getting value from data

- Data silos
- Incongruent data types
- Complexity of solutions
- Multi cloud environment
- Rising costs

Executive Education

Derive real value from your data

⚠	⚠	⚠	⚠	⚠
Data silos	Incongruent data types	Complexity of solutions	Multi cloud environment	Rising costs
✓	✓	✓	✓	✓
One hub for all data	Support for diverse types of data	Unlimited data scale	Familiar tools and ecosystem	Lower TCO

On-premises, hybrid, Azure

SQL Server Integration Services (SSIS)

SQL Server Integration Services (SSIS) is a component of the Microsoft SQL Server database software that can be used to perform a broad range of data integration and data transformation task.

- **Data integration -** it combines the data residing in different sources and provide users with a unified view of these data

- **Data transformation - it transforms the ingested data by applying logic to fit the data objectives**

Azure Data Factory

Connect with confidence

All-inclusive connectivity that prioritizes security and compliance

Reduce integration costs

Serverless, scales on demand to focus on the data, not infrastructure

Work efficiently

Intuitive, visual environment for everyone

Productive & trusted hybrid data integration service that simplifies ETL with any data, from any source, at scale.

Connect with confidence

All-inclusive connectivity

More than 80 natively built and fully managed connectors, no added cost, new connectors added monthly

Efficient and resilient data transfer by leveraging the full capacity of underlying network bandwidth, up to 2 GB/sec throughput

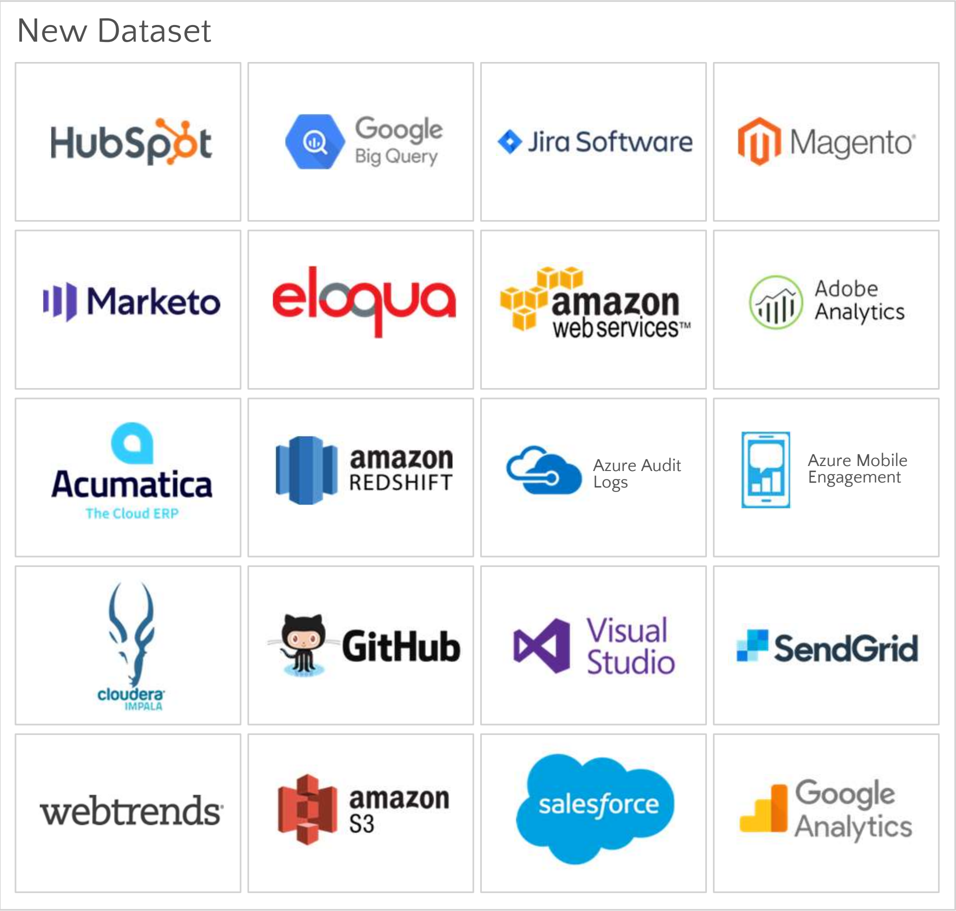
Trusted, global cloud presence

Data Factory availability in 25+ regions, with data movement available globally to help ensure compliance & reduced network egress costs.

Security & compliance peace of mind

Native integration with Azure Active Directory (AAD) and Azure Key Vault (AKV) for identity and access management to cloud solutions & applications, based on centralized policy and rules

HIPAA, HITECH, ISO/IEC 27001, ISO/IEC 27018, CSA STAR certification.



Reduce integration costs

Serverless, fully managed service

No infrastructure to manage, no hardware to upgrade
Scales on demand
Pay only for what you use.

One data integration service for everyone

Reduce integration tool fragmentation & costs
Flexibility to work how you please, visually or using code (Python, .NET or ARM)

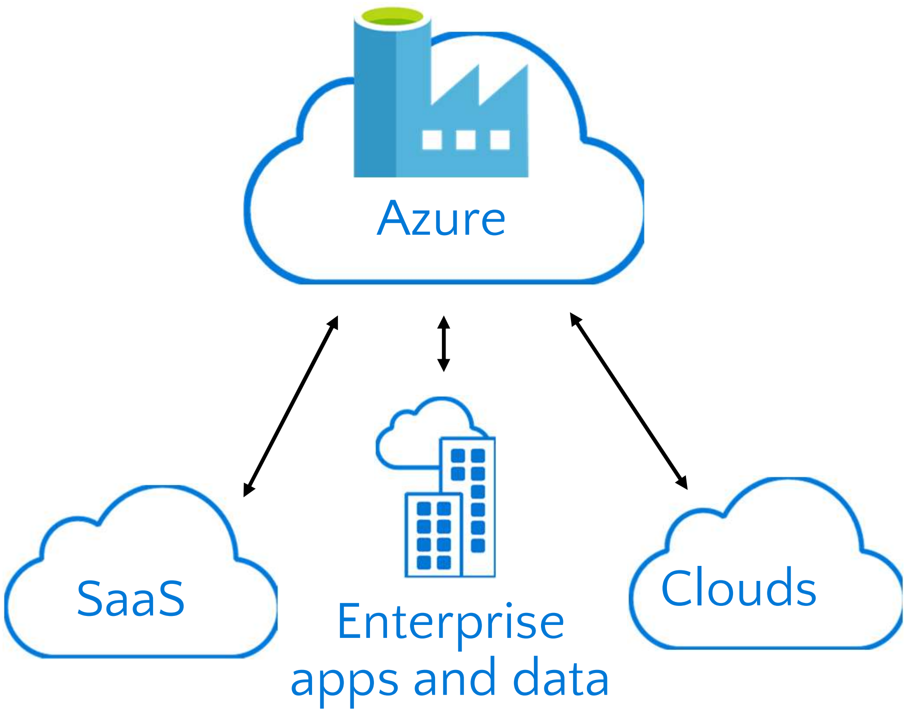
Fast and scalable transformations with Spark

Azure Databricks' Spark engine powers data transformations for fast and fully managed data transformations

.

Reduce development overhead

Migrate to the cloud by moving SSIS packages into Azure without redevelopment
Use existing tools for new development.
Full integration with GitHub for team collaboration.



Executive Education

Work efficiently

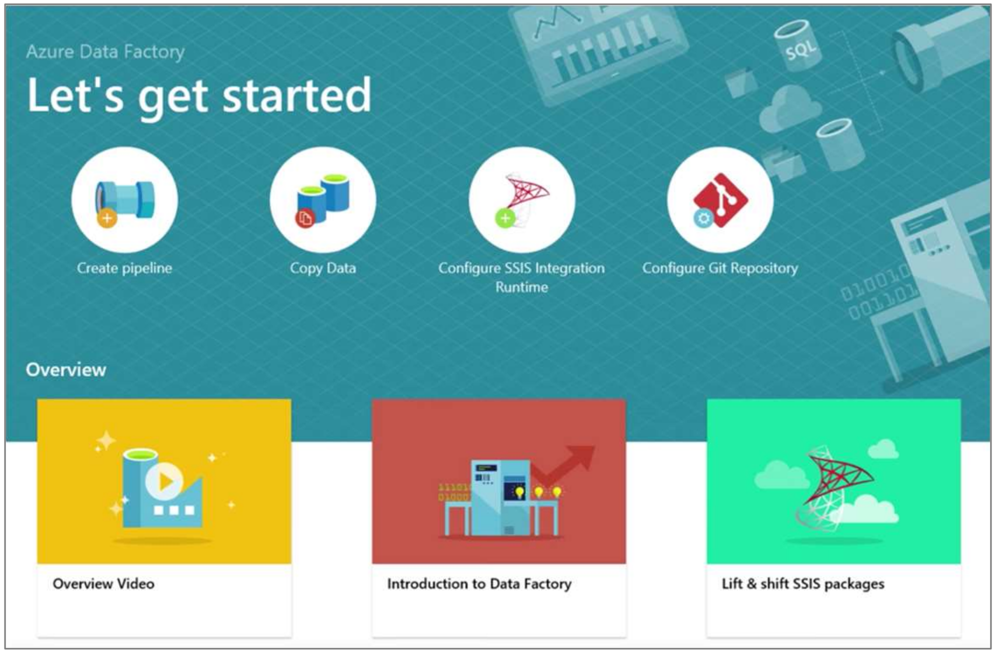
Simple to get started

Azure Data Factory Dashboard: Use tutorials, quick starts, predefined templates, leverage & share best practices & patterns.

Easy to be productive

Visual environment: Ingest, move, prepare, transform and process your data with just a few clicks.

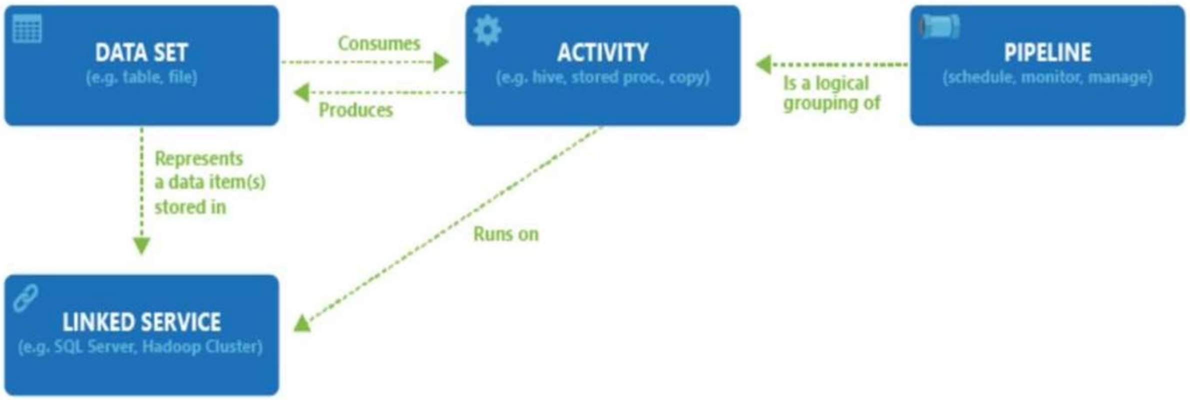
- **Data orchestration**: Visually construct workflows to orchestrate integration and transformation.
- **Data transformation**: Mapping Data Flows to visually create complex pipelines and transforms. Native handling of data evolution / schema drift & for non-relational data, Rich & granular monitoring and management
- **Pipeline automation**: Automate pipeline runs with Triggers
- **Intelligent Data preparation**: Visually explore data with Wrangling Data Flows
- **CI/CD**: Simple dev ops integration with built in support with Azure Monitor, API, PowerShell, Azure Monitor logs, and health panes on the Azure Portal, Git integration



Azure Data Factory - Data Integration Service

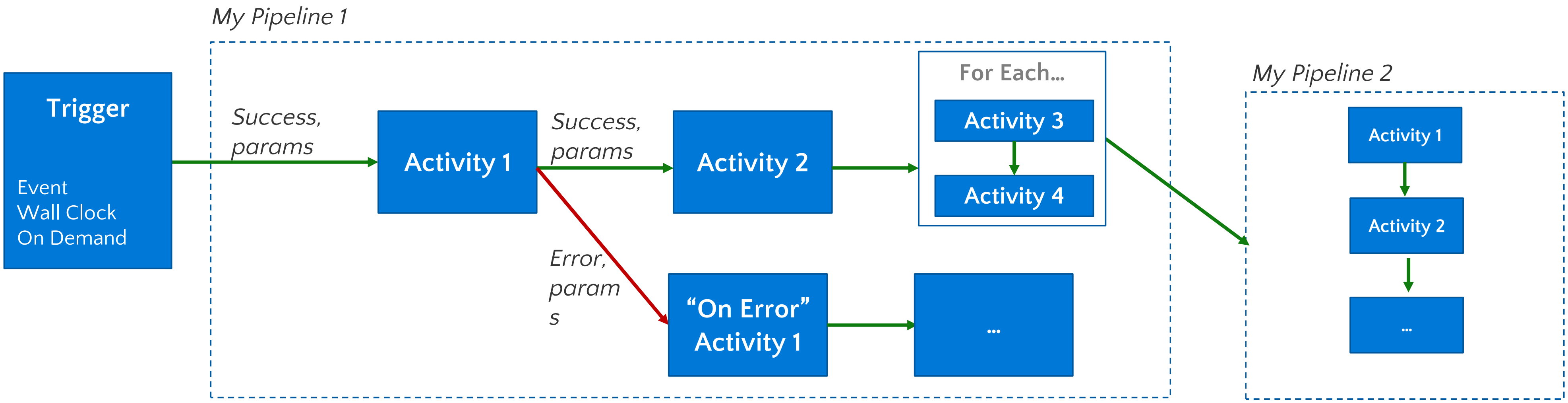
Azure Data Factory (ADF) is a cloud-based data integration service **that orchestrates and automates the movement and transformation of data**.

It orchestrates existing services that collect raw data and transform it into ready-to-use information. ADF is used to **collect data from many different data sources, ingest and prepare it, organize and analyze it with a range of transformations, then publish ready-to-use data for consumption.**

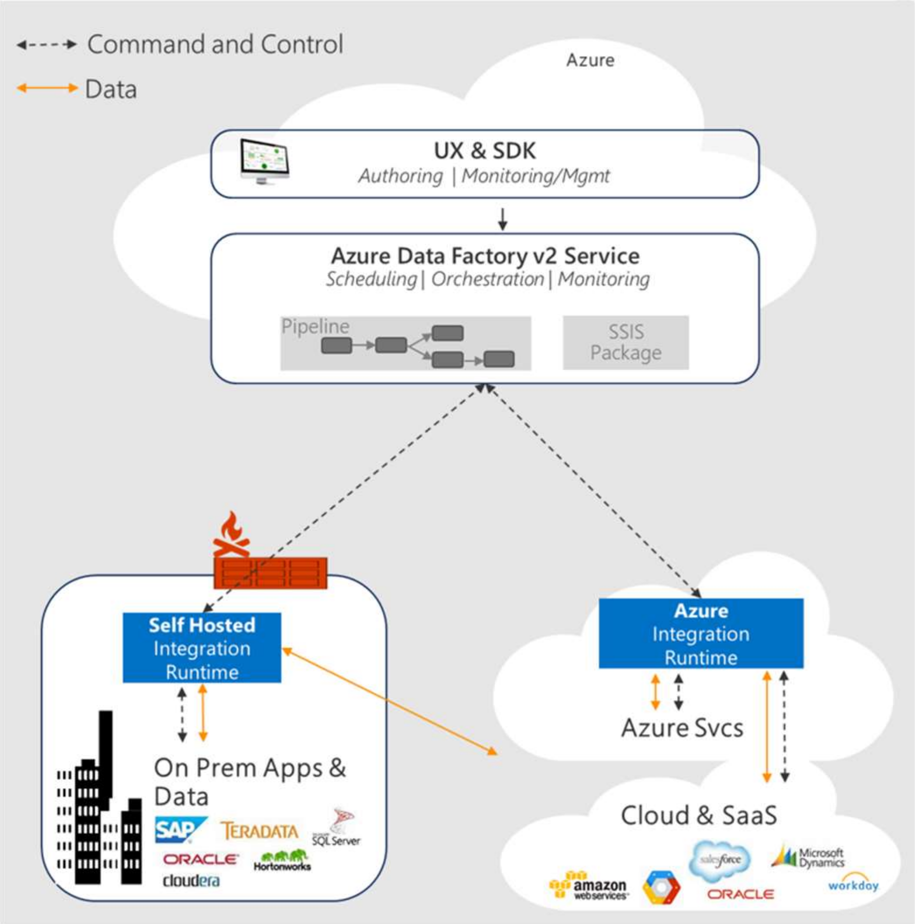

DATA SET
(e.g. table, file)
Consumes
Produces
ACTIVITY
(e.g. hive, stored proc., copy)
Is a logical grouping of
PIPELINE
(schedule, monitor, manage)
Represents a data item(s) stored in
Runs on
LINKED SERVICE
(e.g. SQL Server, Hadoop Cluster)

Azure Data Factory – Control Flow

Coordinate pipeline activities into **finite execution steps to enable looping, conditionals and chaining while separating data transformations into individual data flows**



Azure Data Factory - Data Integration Service



Data Factory

A data integration account.

Location of orchestration, service metadata

Integration Runtime (IR)

ADF’s execution engine

Three core capabilities:

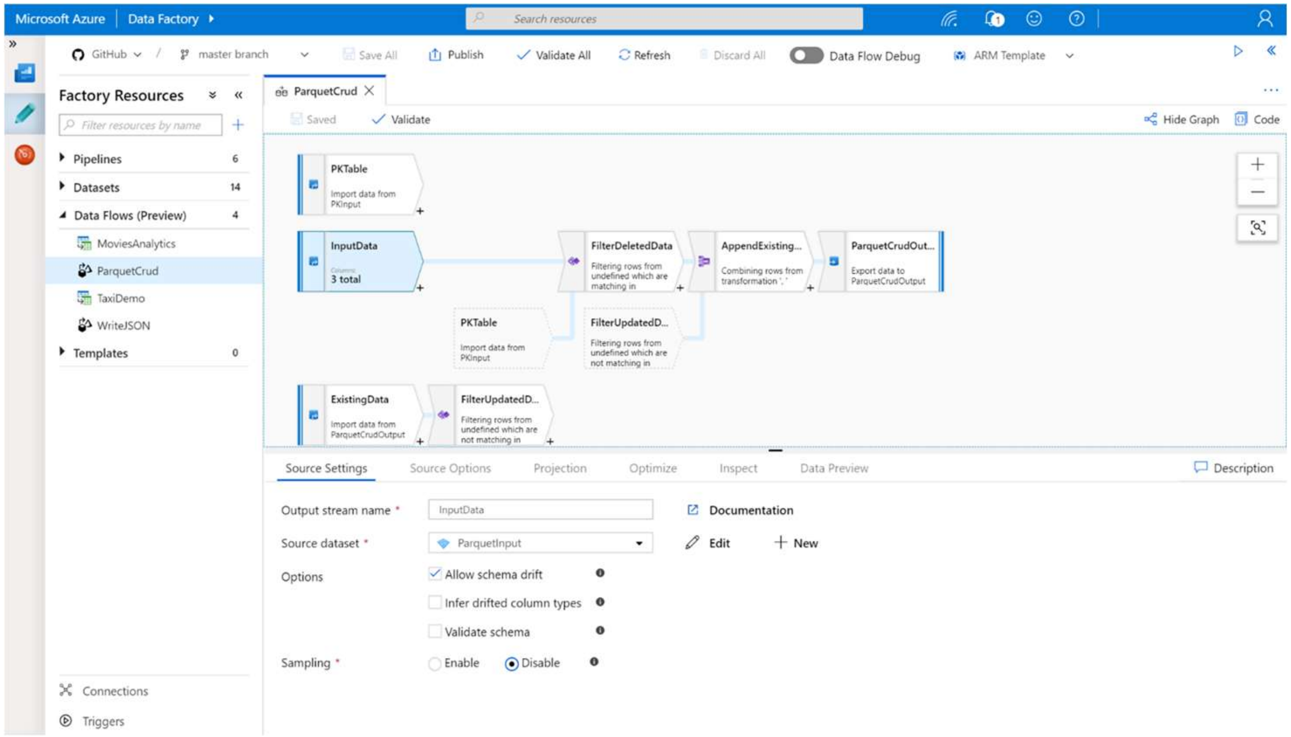
- data movement
- pipeline activity execution
- SSIS package execution

Executive Education

What are Mapping Data Flows?

Data Flow is a new feature of Azure Data Factory to build data transformations in a visual user interface

- Transform at scale, in the cloud
- Code-free pipelines do NOT require understanding of Spark / Scala / Python / Java
- Serverless scale-out transformation execution engine
- Resilient data transformation Flows built for big data scenarios with unstructured data requirements
- Operationalized with Data Factory scheduling, control flow and monitoring



Output stream name * InputData

Source dataset * ParquetInput

Options ☑ Allow schema drift

□ Infer drifted column types

□ Validate schema

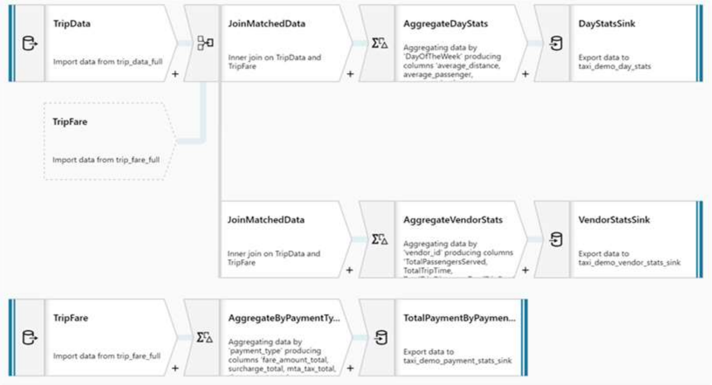
Sampling * ○ Enable ● Disable

Executive Education

Mapping & Wrangling Data Flows

MAPPING DATAFLOW

Code-free data transformation @scale



TripData
Import data from trip_data_full

JoinMatchedData
Inner join on TripData and TripFare

AggregateDayStats
Aggregating data by 'DayOfTheWeek' producing columns 'average_distance, average_passenger,

DayStatsSink
Export data to taxi_demo_day_stats

TripFare
Import data from trip_fare_full

JoinMatchedData
Inner join on TripData and TripFare

AggregateVendorStats
Aggregating data by 'vendor_id' producing columns 'TotalPassengersServed, TotalTripTime,

VendorStatsSink
Export data to taxi_demo_vendor_stats_sink

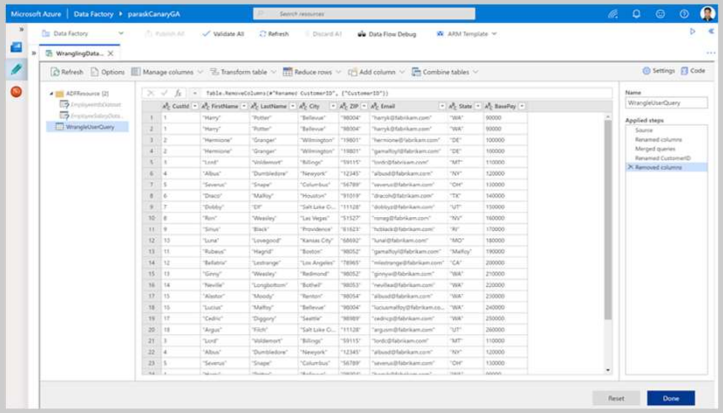
TripFare
Import data from trip_fare_full

AggregateByPaymentTy...
Aggregating data by 'payment_type' producing columns 'fare_amount_total, surcharge_total, mta_tax_total,

TotalPaymentByPaymen...
Export data to taxi_demo_payment_stats_sink

WRANGLING DATAFLOW

Code-free data preparation @scale



```
Table.RemoveColumns(#"Renamed CustomerID", {"CustomerID"})
```

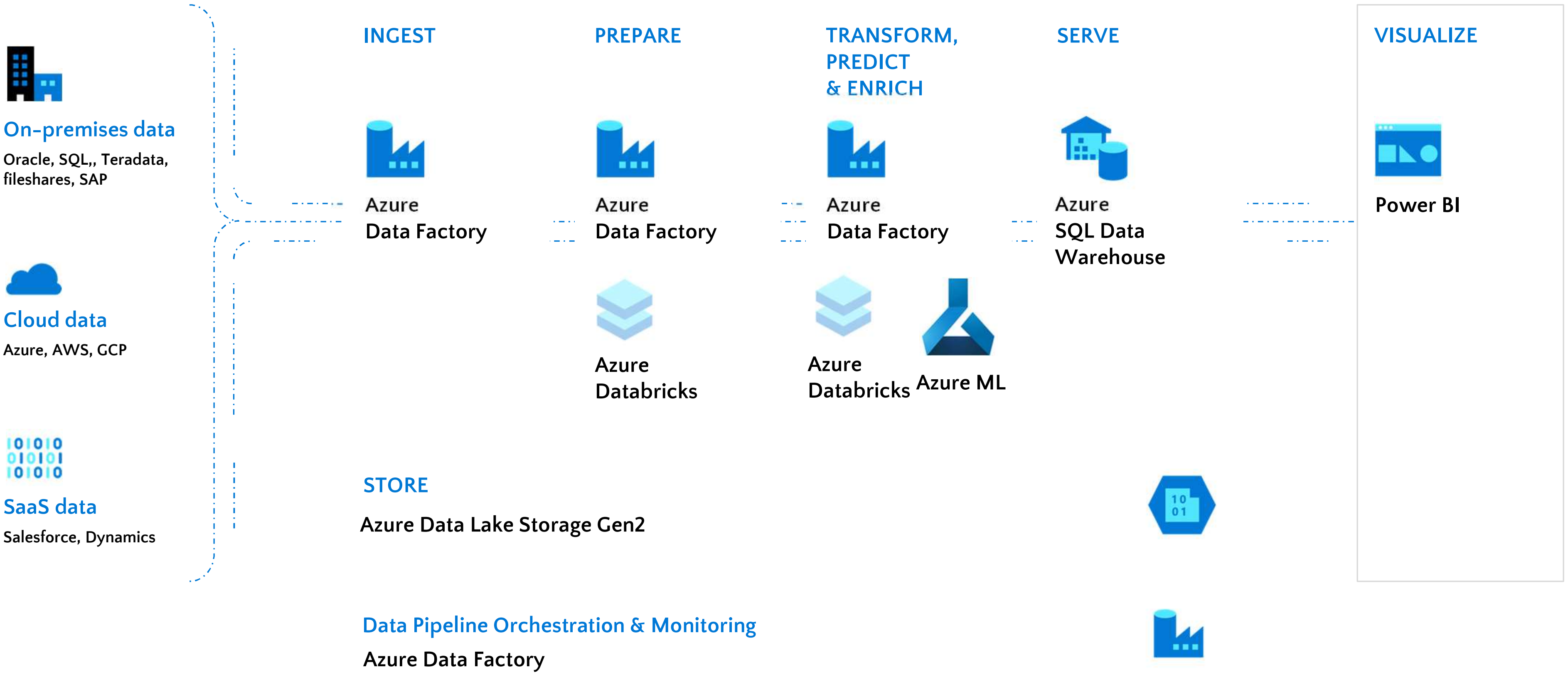
	CustId	FirstName	LastName	City	ZIP	Email	State	BasePay
1	1	"Harry"	"Potter"	"Bellevue"	"98004"	"harryk@fabrikam.com"	"WA"	90000
2	1	"Harry"	"Potter"	"Bellevue"	"98004"	"harryk@fabrikam.com"	"WA"	90000
3	2	"Hermione"	"Granger"	"Wilmington"	"19801"	"hermione@fabrikam.com"	"DE"	100000
4	2	"Hermione"	"Granger"	"Wilmington"	"19801"	"gamalfoyl@fabrikam.com"	"DE"	100000
5	3	"Lord"	"Voldemort"	"Billings"	"59115"	"lordc@fabrikam.com"	"MT"	110000
6	4	"Albus"	"Dumbledore"	"Newyork"	"12345"	"albusd@fabrikam.com"	"NY"	120000
7	5	"Severus"	"Snape"	"Columbus"	"56789"	"severus@fabrikam.com"	"OH"	130000
8	6	"Draco"	"Malfoy"	"Houston"	"91019"	"dracoh@fabrikam.com"	"TX"	140000
9	7	"Dobby"	"Elf"	"Salt Lake C...	"11128"	"dobbyz@fabrikam.com"	"UT"	150000
10	8	"Ron"	"Weasley"	"Las Vegas"	"51527"	"ronag@fabrikam.com"	"NV"	160000
11	9	"Sirius"	"Black"	"Providence"	"61623"	"hcblack@fabrikam.com"	"RI"	170000
12	10	"Luna"	"Lovegood"	"Kansas City"	"68692"	"lunal@fabrikam.com"	"MO"	180000
13	11	"Rubeus"	"Hagrid"	"Boston"	"98052"	"gamalfoyl@fabrikam.com"	"Malfoy"	190000
14	12	"Bellatrix"	"Lestrange"	"Los Angeles"	"78965"	"mlestrange@fabrikam.com"	"CA"	200000
15	13	"Ginny"	"Weasley"	"Redmond"	"98052"	"ginnyw@fabrikam.com"	"WA"	210000
16	14	"Neville"	"Longbottom"	"Bothell"	"98053"	"nevillea@fabrikam.com"	"WA"	220000
17	15	"Alastor"	"Moody"	"Renton"	"98054"	"albusd@fabrikam.com"	"WA"	230000
18	16	"Lucius"	"Malfoy"	"Bellevue"	"98004"	"luciusmalfoy@fabrikam.co...	"WA"	240000
19	17	"Cedric"	"Diggory"	"Seattle"	"98989"	"cedricp@fabrikam.com"	"WA"	250000
20	18	"Argus"	"Filch"	"Salt Lake C...	"11128"	"argusm@fabrikam.com"	"UT"	260000
21	3	"Lord"	"Voldemort"	"Billings"	"59115"	"lordc@fabrikam.com"	"MT"	110000
22	4	"Albus"	"Dumbledore"	"Newyork"	"12345"	"albusd@fabrikam.com"	"NY"	120000
23	5	"Severus"	"Snape"	"Columbus"	"56789"	"severus@fabrikam.com"	"OH"	130000

Name: WrangleUserQuery

Applied steps
- Source
- Renamed columns
- Merged queries
- Renamed CustomerID
- ✕ Removed columns

PUBLIC PREVIEW

Modern Data Warehouse (MDW)



On-premises data

Oracle, SQL,, Teradata, fileshares, SAP

Cloud data

Azure, AWS, GCP

SaaS data

Salesforce, Dynamics

INGEST

Azure Data Factory

PREPARE

Azure Data Factory

Azure Databricks

TRANSFORM, PREDICT & ENRICH

Azure Data Factory

Azure Databricks

Azure ML

SERVE

Azure SQL Data Warehouse

VISUALIZE

Power BI

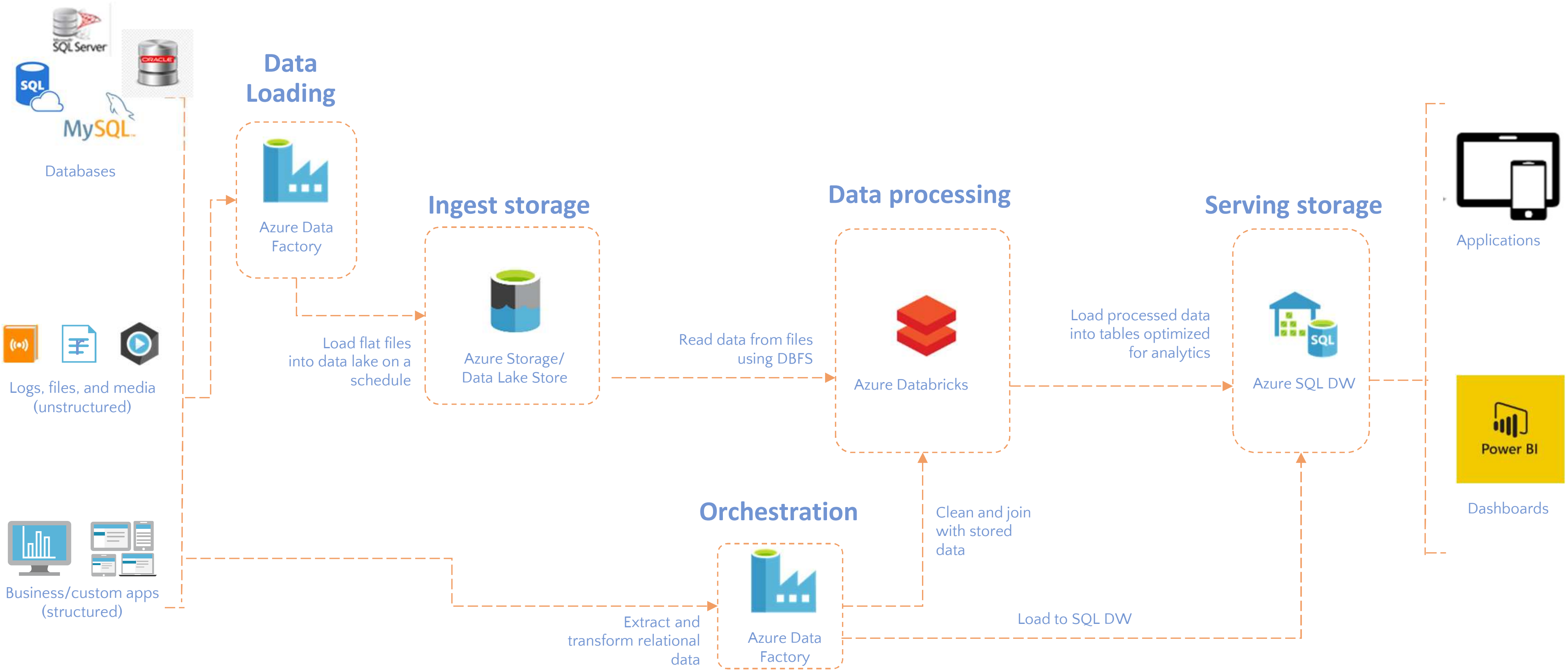
STORE

Azure Data Lake Storage Gen2

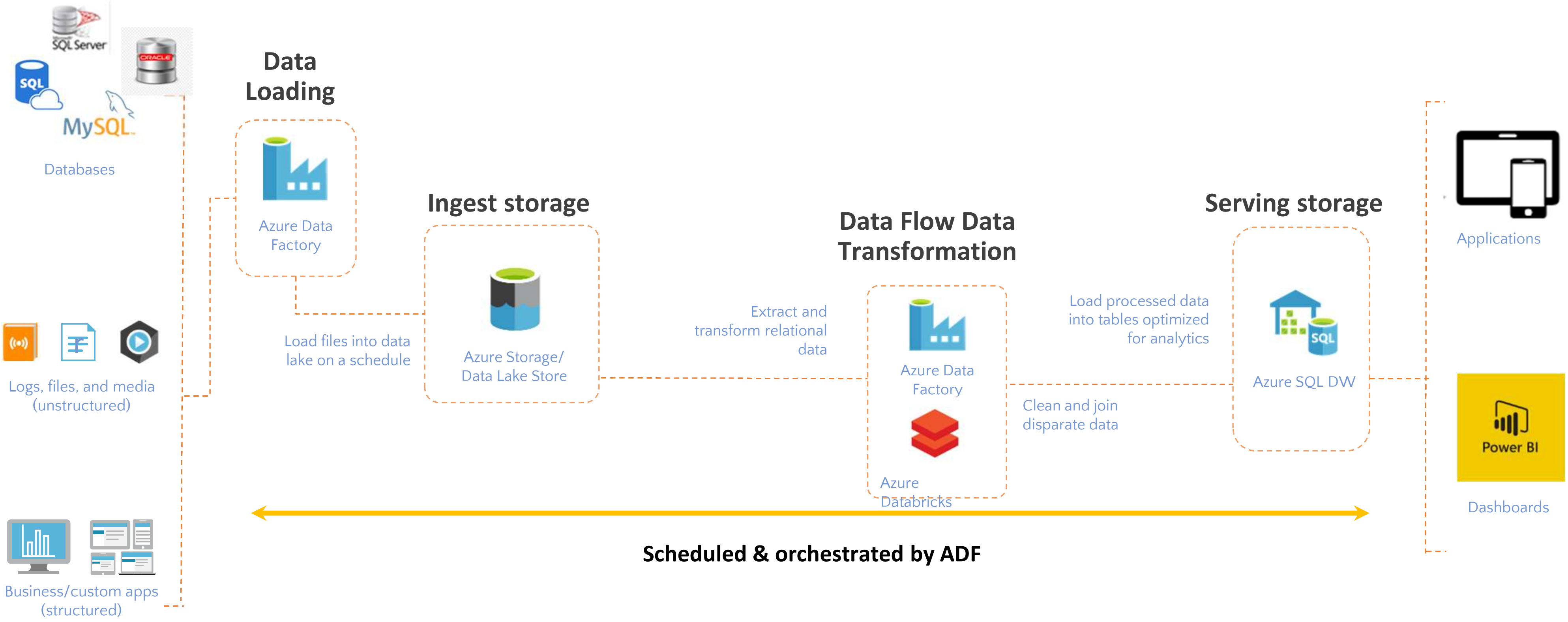
Data Pipeline Orchestration & Monitoring

Azure Data Factory

Modern Data Warehouse Pattern Today



Modern Data Warehouse Pattern with Mapping Data Flows



Databases

Logs, files, and media (unstructured)

Business/custom apps (structured)

Data Loading

Azure Data Factory

Load files into data lake on a schedule

Ingest storage

Azure Storage/ Data Lake Store

Extract and transform relational data

Data Flow Data Transformation

Azure Data Factory

Azure Databricks

Load processed data into tables optimized for analytics

Clean and join disparate data

Serving storage

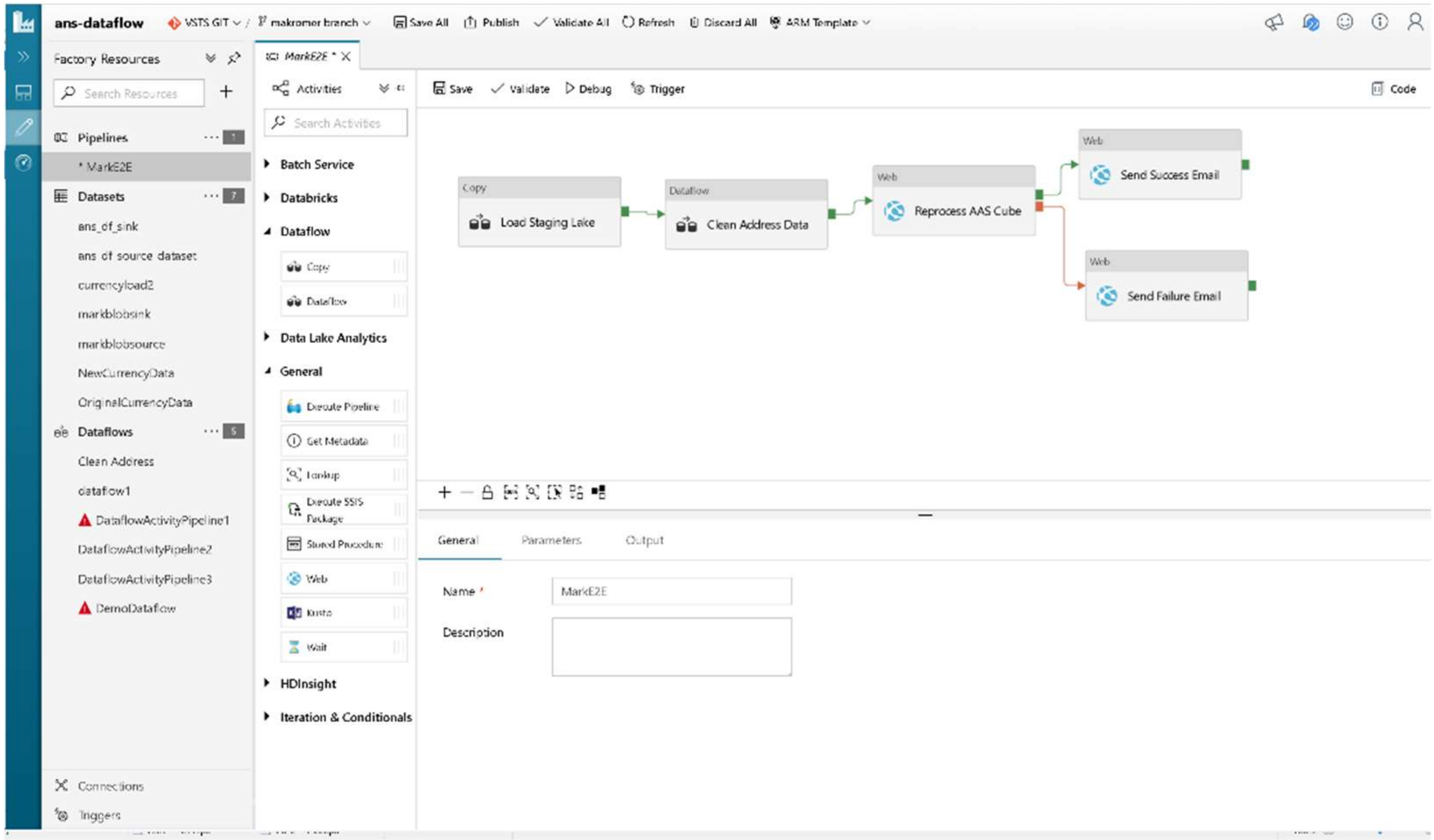
Azure SQL DW

Applications

Dashboards

Scheduled & orchestrated by ADF

Pipeline execution of a Data Flow Activity



Name * MarkE2E

Description

- Design code-free ETL workflows
- Copy data from on-prem, other clouds and Azure
- Stage data for transformation
- Build visual data transformations
- Schedule triggers for your pipeline execution
- Monitor processes and configure alerts
- All within ADF

Best in class monitoring and management

Monitor Pipeline and Activity Runs

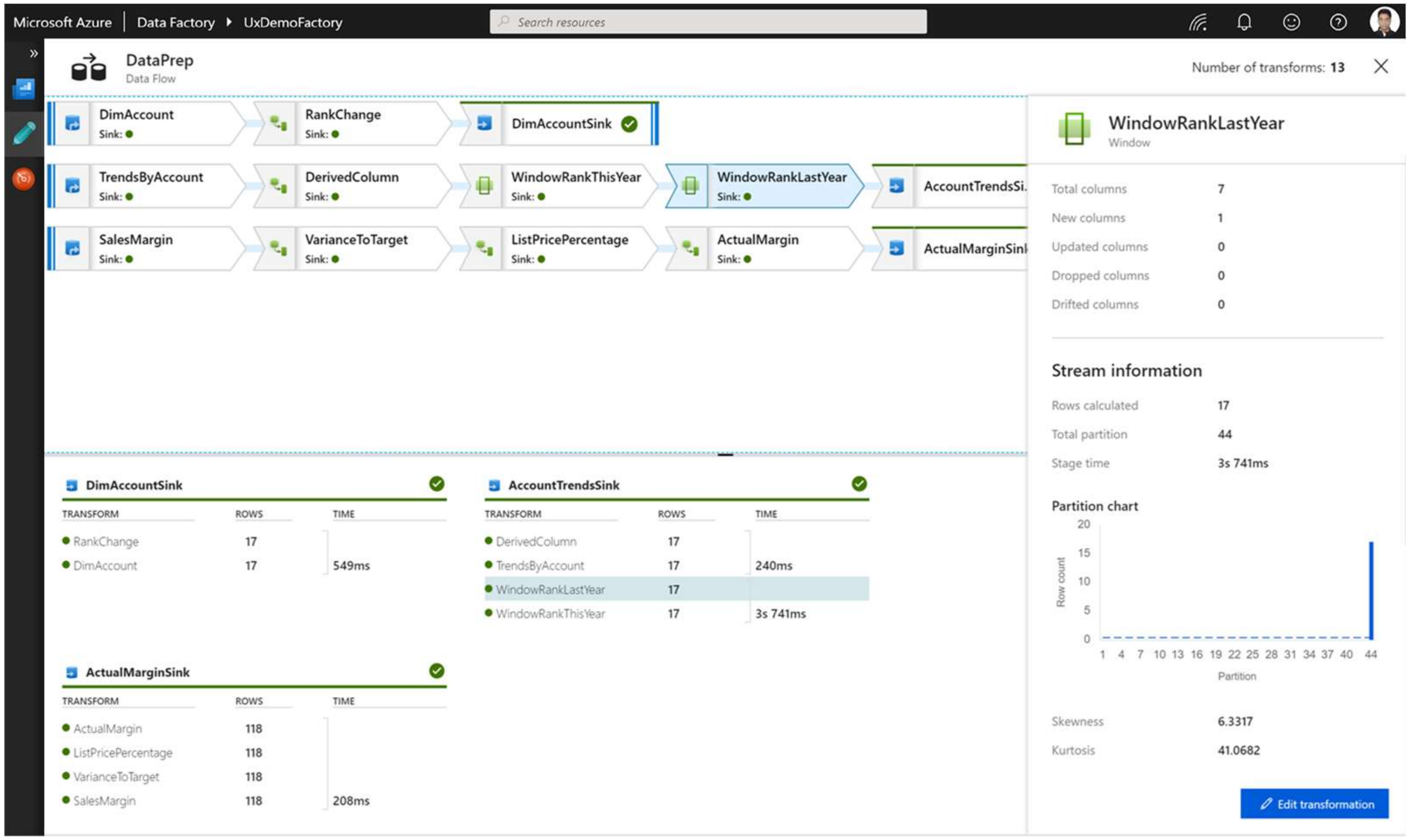
Rich language to query Runs

Operational lineage between parent-child pipelines

Azure Monitor Integration

- Diagnostics logging
- Metrics & Alerts
- Events

Restate Pipeline and Activities



Number of transforms: 13

DimAccountSink ✓

TRANSFORM	ROWS	TIME
● RankChange	17	
● DimAccount	17	549ms

AccountTrendsSink ✓

TRANSFORM	ROWS	TIME
● DerivedColumn	17	
● TrendsByAccount	17	240ms
● WindowRankLastYear	17	
● WindowRankThisYear	17	3s 741ms

ActualMarginSink ✓

TRANSFORM	ROWS	TIME
● ActualMargin	118	
● ListPricePercentage	118	
● VarianceToTarget	118	
● SalesMargin	118	208ms

WindowRankLastYear
Window

Total columns	7
New columns	1
Updated columns	0
Dropped columns	0
Drifted columns	0

Stream information

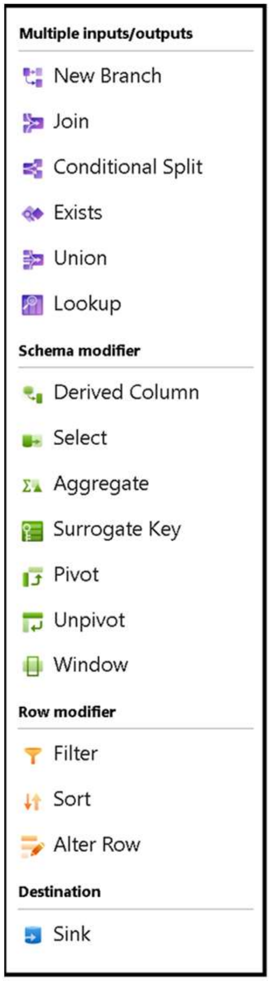
Rows calculated	17
Total partition	44
Stage time	3s 741ms

Partition chart

Skewness	6.3317
Kurtosis	41.0682

Data Transformations, Expression Language and Debugging

Azure Data Factory Continues to Extend Data Flow Library



Multiple inputs/outputs

- New Branch
- Join
- Conditional Split
- Exists
- Union
- Lookup

Schema modifier

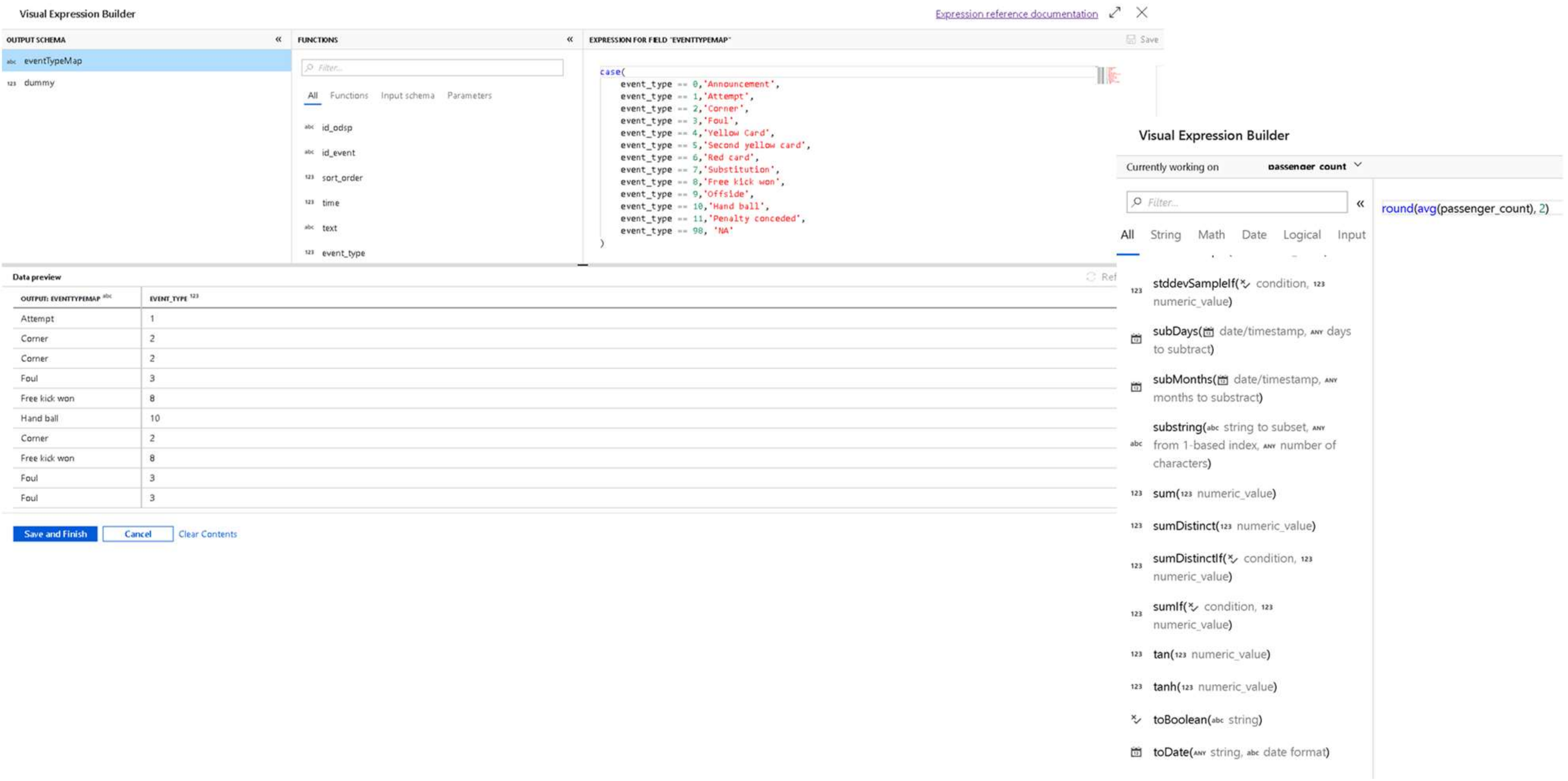
- Derived Column
- Select
- Aggregate
- Surrogate Key
- Pivot
- Unpivot
- Window

Row modifier

- Filter
- Sort
- Alter Row

Destination

- Sink



```
case(
    event_type == 0,'Announcement',
    event_type == 1,'Attempt',
    event_type == 2,'Corner',
    event_type == 3,'Foul',
    event_type == 4,'Yellow Card',
    event_type == 5,'Second yellow card',
    event_type == 6,'Red card',
    event_type == 7,'Substitution',
    event_type == 8,'Free kick won',
    event_type == 9,'Offside',
    event_type == 10,'Hand ball',
    event_type == 11,'Penalty conceded',
    event_type == 98, 'NA'
)
```

Data preview

OUTPUT: EVENTTYPEMAP abc	EVENT_TYPE 123
Attempt	1
Corner	2
Corner	2
Foul	3
Free kick won	8
Hand ball	10
Corner	2
Free kick won	8
Foul	3
Foul	3

Currently working on passenger count

```
round(avg(passenger_count), 2)
```

- stddevSampleIf(condition, numeric_value)
- subDays(date/timestamp, days to subtract)
- subMonths(date/timestamp, months to substract)
- substring(string to subset, from 1-based index, number of characters)
- sum(numeric_value)
- sumDistinct(numeric_value)
- sumDistinctIf(condition, numeric_value)
- sumIf(condition, numeric_value)
- tan(numeric_value)
- tanh(numeric_value)
- toBoolean(string)
- toDate(string, date format)

Executive Education

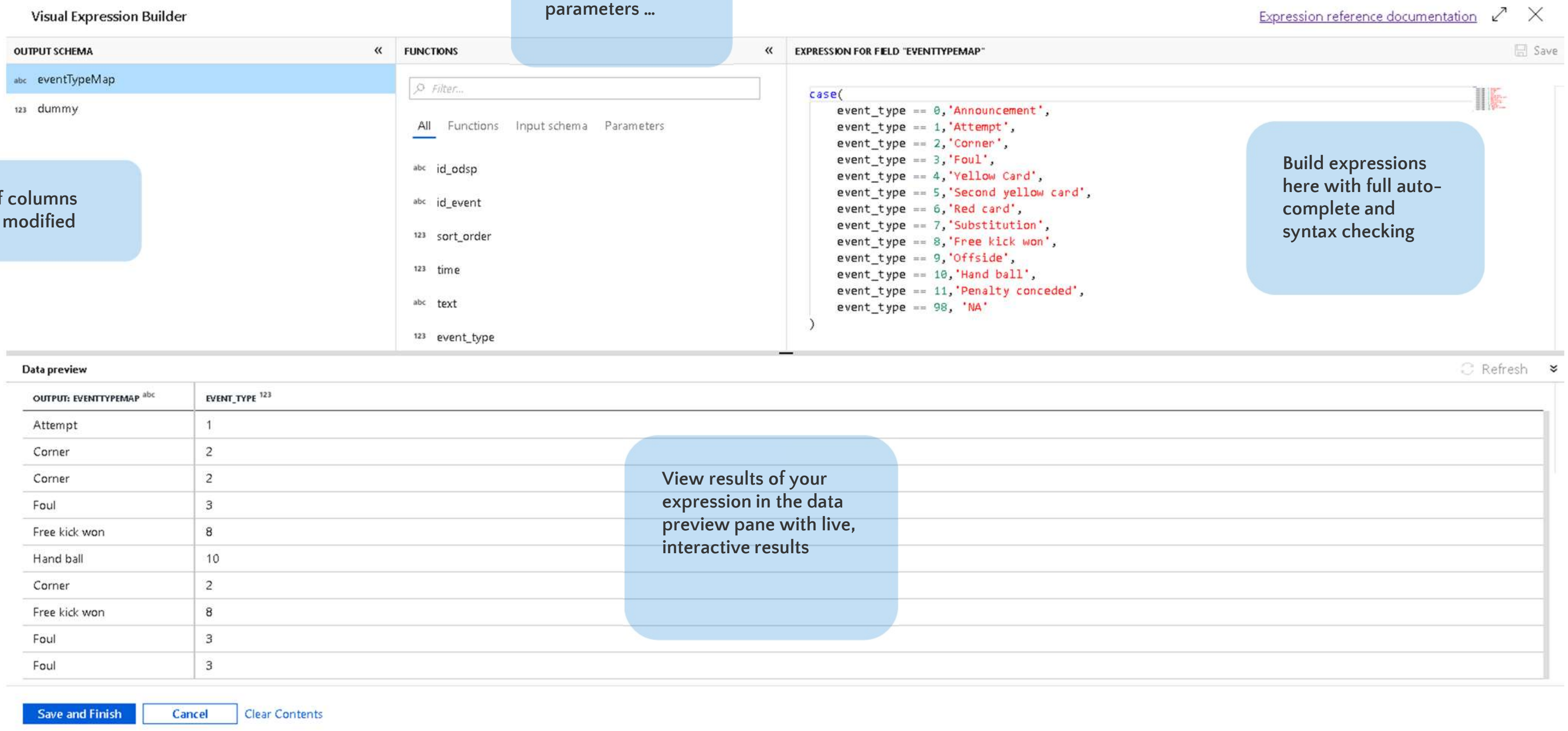
Expression builder

All available functions, fields, parameters …

List of columns being modified

Build expressions here with full auto-complete and syntax checking

View results of your expression in the data preview pane with live, interactive results

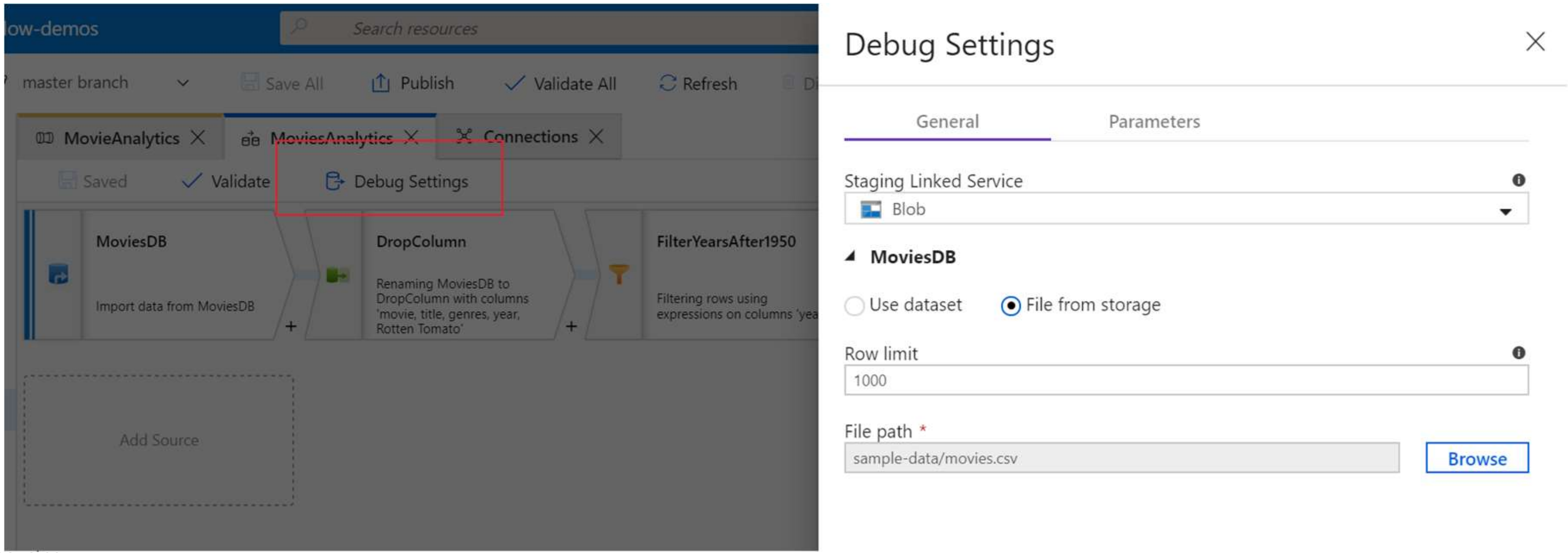


```
case(
    event_type == 0,'Announcement',
    event_type == 1,'Attempt',
    event_type == 2,'Corner',
    event_type == 3,'Foul',
    event_type == 4,'Yellow Card',
    event_type == 5,'Second yellow card',
    event_type == 6,'Red card',
    event_type == 7,'Substitution',
    event_type == 8,'Free kick won',
    event_type == 9,'Offside',
    event_type == 10,'Hand ball',
    event_type == 11,'Penalty conceded',
    event_type == 98, 'NA'
)
```

OUTPUT: EVENTTYPEMAP abc	EVENT_TYPE 123
Attempt	1
Corner	2
Corner	2
Foul	3
Free kick won	8
Hand ball	10
Corner	2
Free kick won	8
Foul	3
Foul	3

Switch to Debug Mode and select sample data to work with for debugging

- Set Parameter values and sample data in debug settings
  - Change # of rows used per source
  - Replace source with debug dataset
  - Assign debug parameter values



Debug Settings

General | Parameters

Staging Linked Service: Blob

MoviesDB

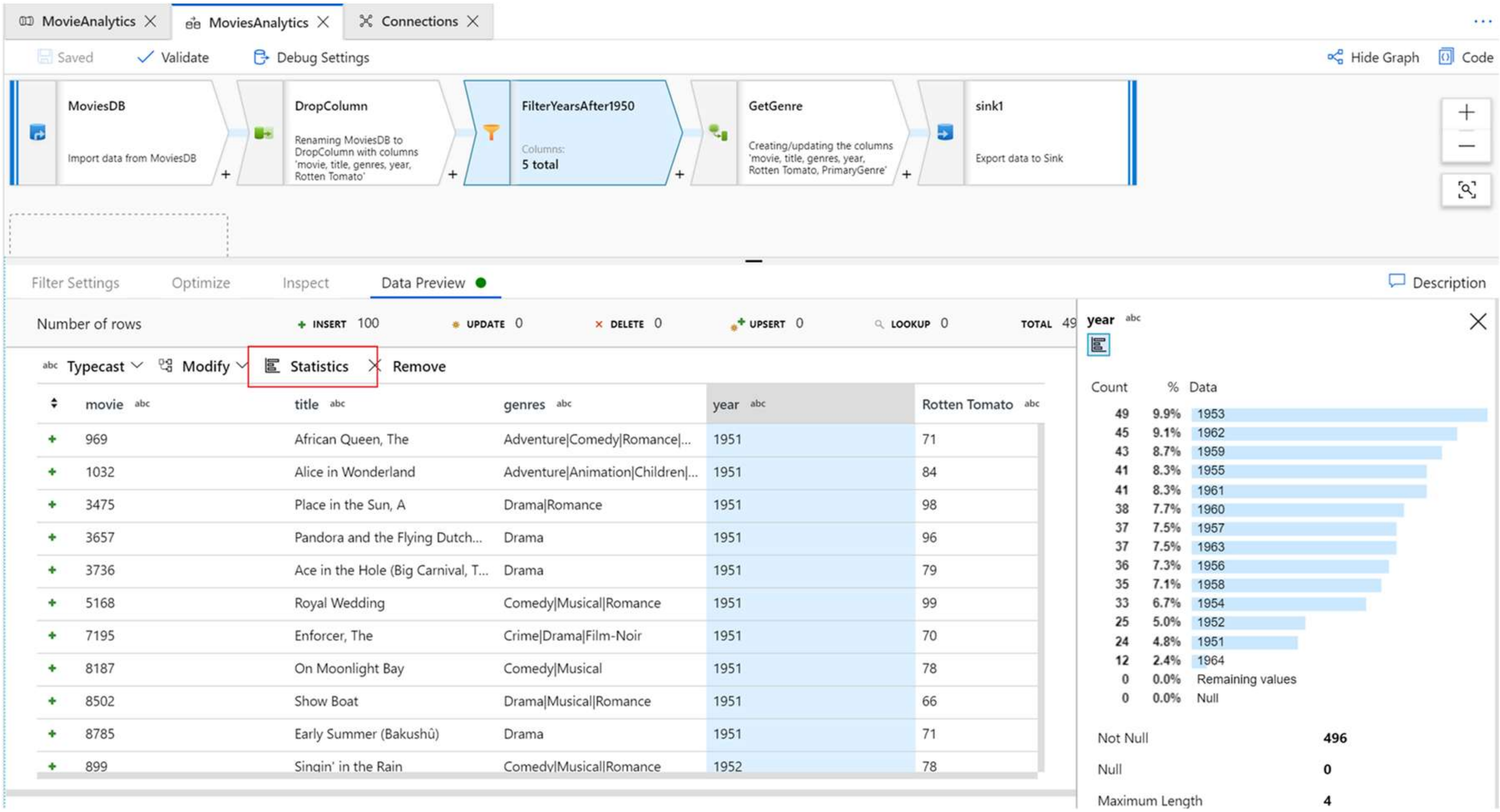
○ Use dataset ● File from storage

Row limit: 1000

File path *: sample-data/movies.csv

Browse

Debug Data Flows with Data Preview and Data Sampling



MoviesDB
Import data from MoviesDB

DropColumn
Renaming MoviesDB to DropColumn with columns 'movie, title, genres, year, Rotten Tomato'

FilterYearsAfter1950
Columns:
5 total

GetGenre
Creating/updating the columns 'movie, title, genres, year, Rotten Tomato, PrimaryGenre'

sink1
Export data to Sink

Number of rows: INSERT 100 · UPDATE 0 · DELETE 0 · UPSERT 0 · LOOKUP 0 · TOTAL 49

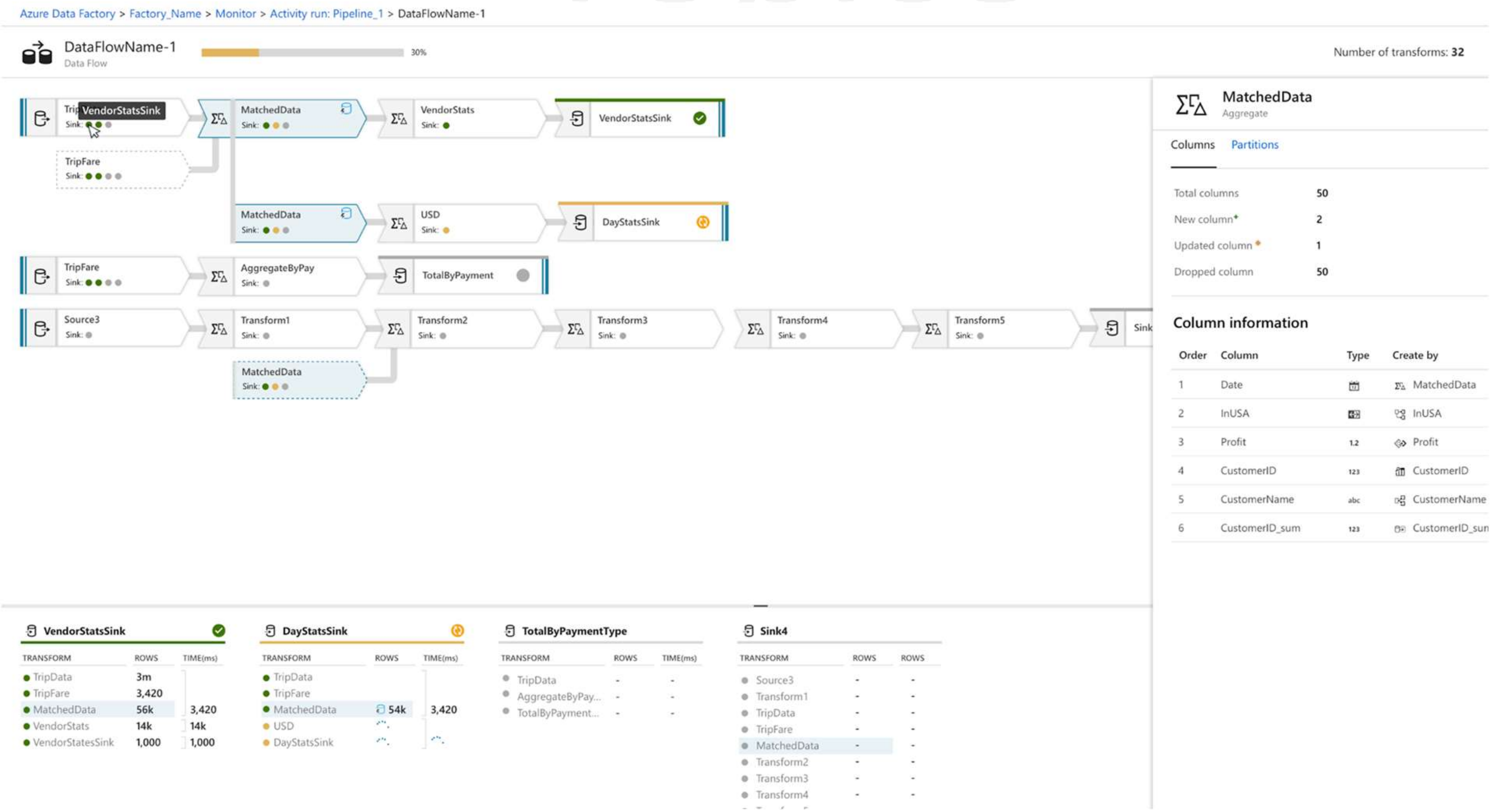
	movie abc	title abc	genres abc	year abc	Rotten Tomato abc
+	969	African Queen, The	Adventure|Comedy|Romance|...	1951	71
+	1032	Alice in Wonderland	Adventure|Animation|Children|...	1951	84
+	3475	Place in the Sun, A	Drama|Romance	1951	98
+	3657	Pandora and the Flying Dutch...	Drama	1951	96
+	3736	Ace in the Hole (Big Carnival, T...	Drama	1951	79
+	5168	Royal Wedding	Comedy|Musical|Romance	1951	99
+	7195	Enforcer, The	Crime|Drama|Film-Noir	1951	70
+	8187	On Moonlight Bay	Comedy|Musical	1951	78
+	8502	Show Boat	Drama|Musical|Romance	1951	66
+	8785	Early Summer (Bakushû)	Drama	1951	71
+	899	Singin' in the Rain	Comedy|Musical|Romance	1952	78

year abc

Count	%	Data
49	9.9%	1953
45	9.1%	1962
43	8.7%	1959
41	8.3%	1955
41	8.3%	1961
38	7.7%	1960
37	7.5%	1957
37	7.5%	1963
36	7.3%	1956
35	7.1%	1958
33	6.7%	1954
25	5.0%	1952
24	4.8%	1951
12	2.4%	1964
0	0.0%	Remaining values
0	0.0%	Null

Not Null	496
Null	0
Maximum Length	4

Deep Monitoring Introspection of Data Transformations



Number of transforms: **32**

MatchedData
Aggregate

Columns | Partitions

Total columns	50
New column*	2
Updated column *	1
Dropped column	50

Column information

Order	Column	Type	Create by
1	Date		MatchedData
2	InUSA		InUSA
3	Profit	1.2	Profit
4	CustomerID	123	CustomerID
5	CustomerName	abc	CustomerName
6	CustomerID_sum	123	CustomerID_sun

VendorStatsSink ✓

TRANSFORM	ROWS	TIME(ms)
● TripData	3m	
● TripFare	3,420	
● MatchedData	56k	3,420
● VendorStats	14k	14k
● VendorStatesSink	1,000	1,000

DayStatsSink

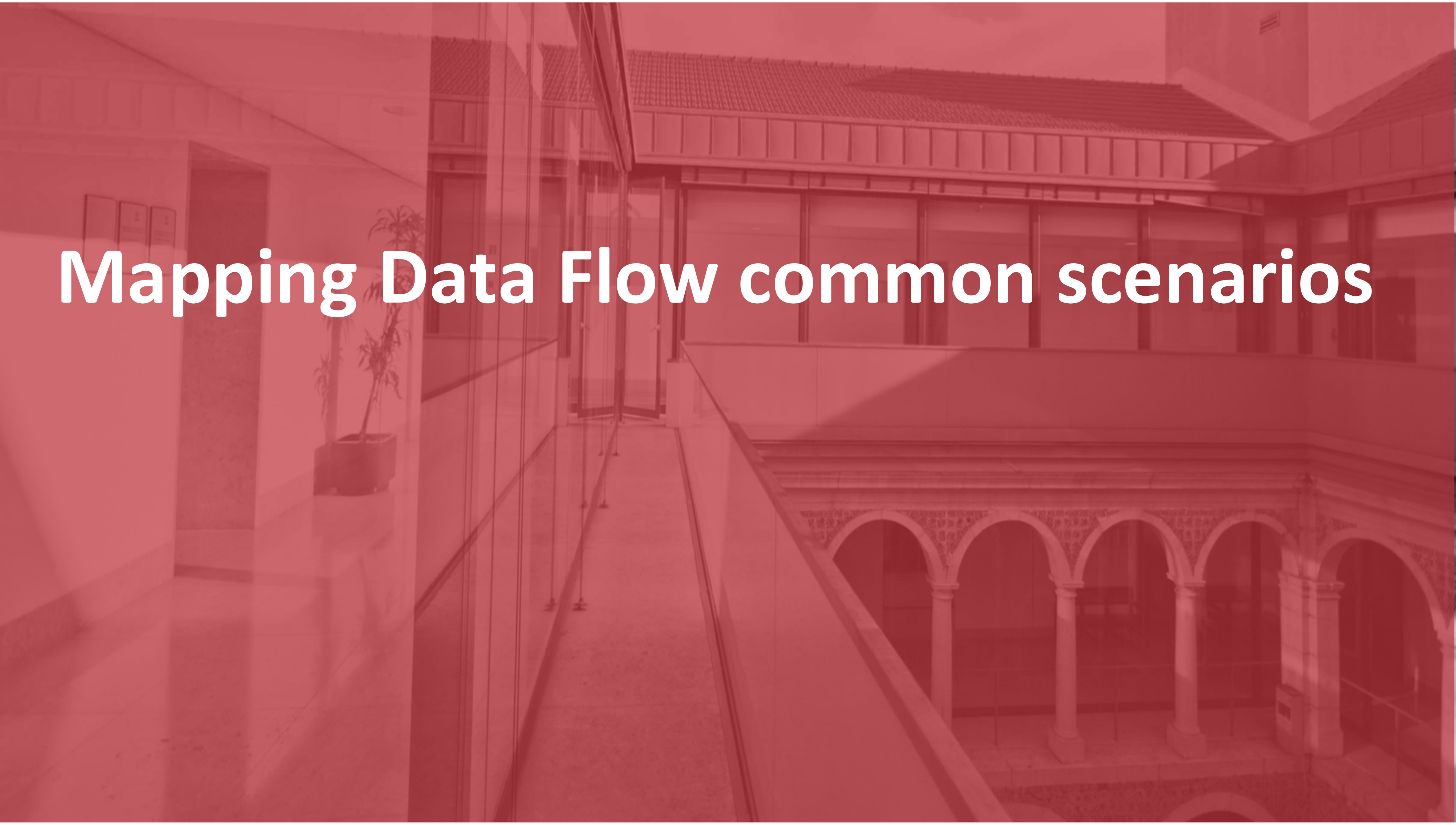
TRANSFORM	ROWS	TIME(ms)
● TripData		
● TripFare		
● MatchedData	54k	3,420
● USD		
● DayStatsSink		

TotalByPaymentType

TRANSFORM	ROWS	TIME(ms)
● TripData	-	-
● AggregateByPay...	-	-
● TotalByPayment...	-	-

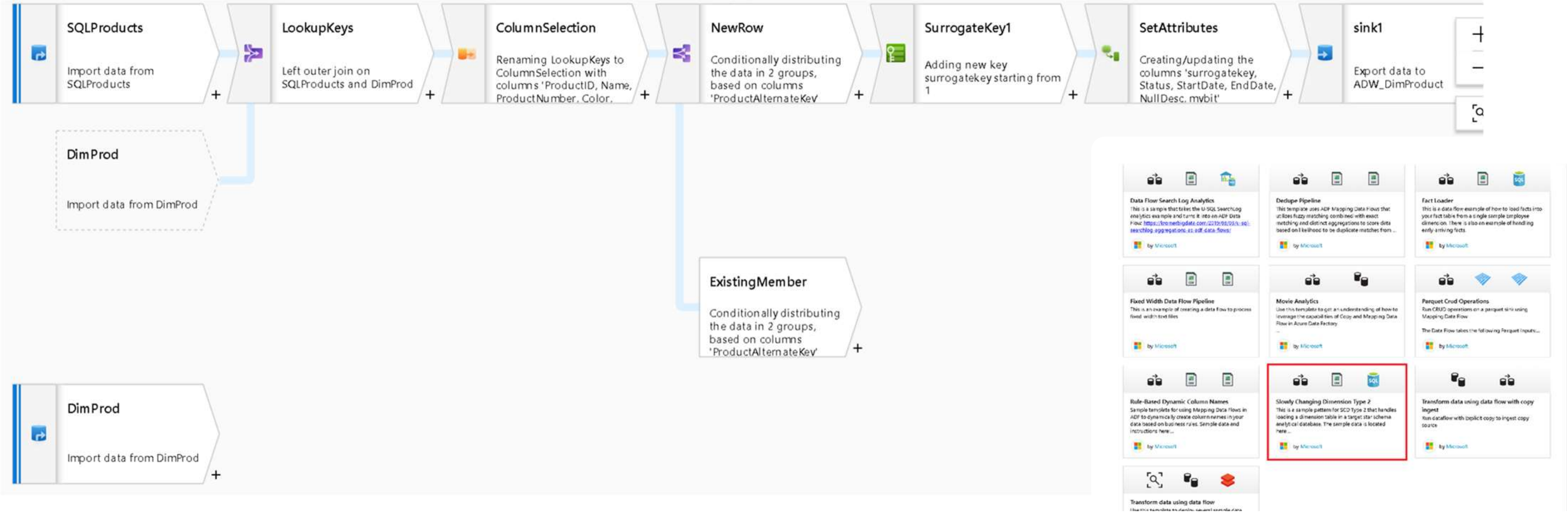
Sink4

TRANSFORM	ROWS	ROWS
● Source3	-	-
● Transform1	-	-
● TripData	-	-
● TripFare	-	-
● MatchedData	-	-
● Transform2	-	-
● Transform3	-	-
● Transform4	-	-



Mapping Data Flow common scenarios

Slowly Changing Dimension Scenario



SQLProducts
Import data from SQLProducts

LookupKeys
Left outer join on SQLProducts and DimProd

ColumnSelection
Renaming LookupKeys to ColumnSelection with columns 'ProductID, Name, ProductNumber, Color,

NewRow
Conditionally distributing the data in 2 groups, based on columns 'ProductAlternateKey'

SurrogateKey1
Adding new key surrogatekey starting from 1

SetAttributes
Creating/updating the columns 'surrogatekey, Status, StartDate, EndDate, NullDesc, mvbit'

sink1
Export data to ADW_DimProduct

DimProd
Import data from DimProd

ExistingMember
Conditionally distributing the data in 2 groups, based on columns 'ProductAlternateKey'

DimProd
Import data from DimProd

Data Flow Search Log Analytics
This is a sample that takes the U-SQL SearchLog analytics example and turns it into an ADF Data Flow: https://kromerbigdata.com/2019/03/03/u-sql-searchlog-aggregations-as-adf-data-flows/
by Microsoft

Dedupe Pipeline
This template uses ADF Mapping Data Flows that utilizes fuzzy matching combined with exact matching and distinct aggregations to score data based on likelihood to be duplicate matches from ...
by Microsoft

Fact Loader
This is a data flow example of how to load facts into your fact table from a single sample employee dimension. There is also an example of handling early arriving facts.
by Microsoft

Fixed Width Data Flow Pipeline
This is an example of creating a data flow to process fixed width text files
by Microsoft

Movie Analytics
Use this template to get an understanding of how to leverage the capabilities of Copy and Mapping Data Flow in Azure Data Factory
...
by Microsoft

Parquet Crud Operations
Run CRUD operations on a parquet sink using Mapping Data Flow

The Data Flow takes the following Parquet inputs:...
by Microsoft

Rule-Based Dynamic Column Names
Sample template for using Mapping Data Flows in ADF to dynamically create column names in your data based on business rules. Sample data and instructions here...
by Microsoft

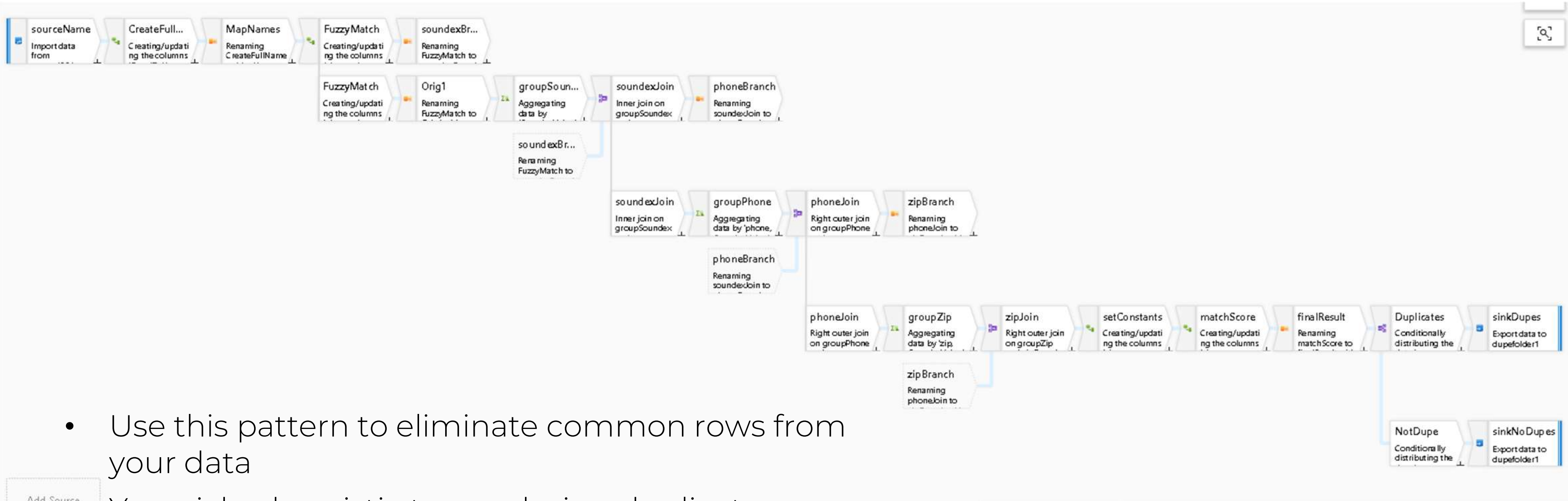
Slowly Changing Dimension Type 2
This is a sample pattern for SCD Type 2 that handles loading a dimension table in a target star schema analytical database. The sample data is located here ...
by Microsoft

Transform data using data flow with copy ingest
Run dataflow with explicit copy to ingest copy source
by Microsoft

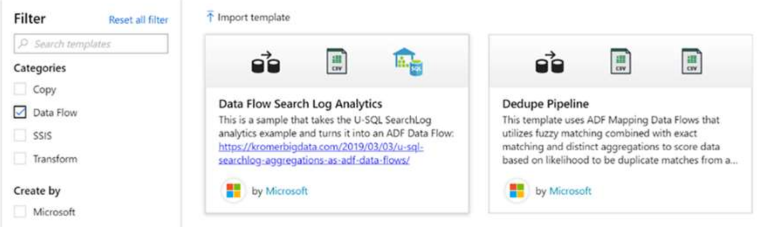
Transform data using data flow
Use this template to deploy several sample data

- Common DW pattern to manage changing attributes to dimension members
- Graphically build code-free SCD ETL pattern to load your data warehouse
- Connect directly to Azure SQL DB and Azure SQL DW
- Use Lookup, Surrogate Key, Derived Column and Select transforms

Data De-Duplication



- Use this pattern to eliminate common rows from your data
- You pick a heuristic to use during duplicate matching
- You can tag rows and/or remove duplicate rows
- Use exact matching and/or fuzzy matching
- Available as pipeline template *Dedupe Pipeline*



Filter Reset all filter

Search templates

Categories

☐ Copy

☑ Data Flow

☐ SSIS

☐ Transform

Create by

☐ Microsoft

Import template

Data Flow Search Log Analytics

This is a sample that takes the U-SQL SearchLog analytics example and turns it into an ADF Data Flow: https://kromerbigdata.com/2019/03/03/u-sql-searchlog-aggregations-as-adf-data-flows/

by Microsoft

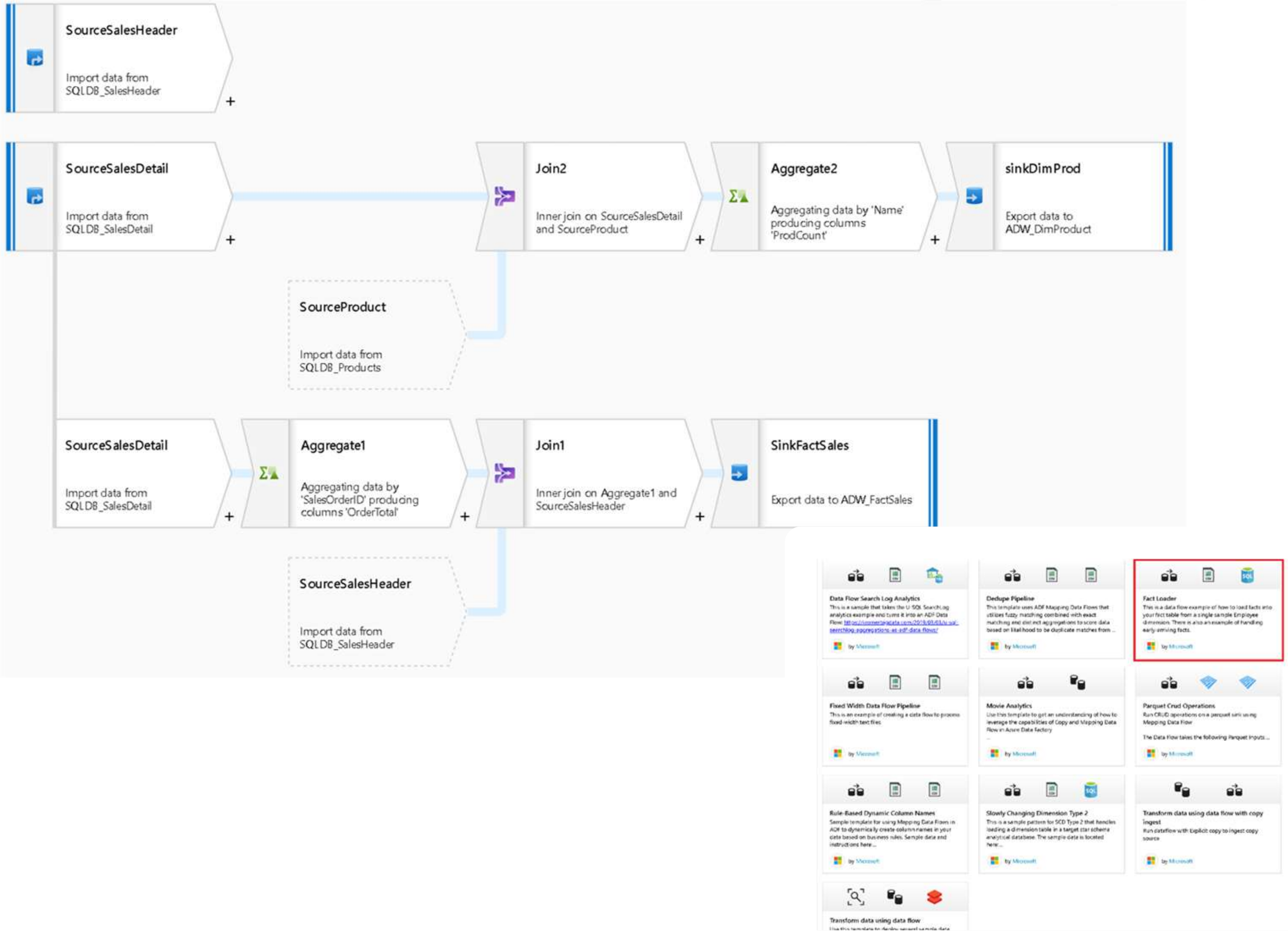
Dedupe Pipeline

This template uses ADF Mapping Data Flows that utilizes fuzzy matching combined with exact matching and distinct aggregations to score data based on likelihood to be duplicate matches from a...

by Microsoft

Executive Education

Load Fact Table in DW Scenario


SourceSalesHeader
Import data from SQLDB_SalesHeader
SourceSalesDetail
Import data from SQLDB_SalesDetail
Join2
Inner join on SourceSalesDetail and SourceProduct
Aggregate2
Aggregating data by 'Name' producing columns 'ProdCount'
sinkDimProd
Export data to ADW_DimProduct
SourceProduct
Import data from SQLDB_Products
SourceSalesDetail
Import data from SQLDB_SalesDetail
Aggregate1
Aggregating data by 'SalesOrderID' producing columns 'OrderTotal'
Join1
Inner join on Aggregate1 and SourceSalesHeader
SinkFactSales
Export data to ADW_FactSales
SourceSalesHeader
Import data from SQLDB_SalesHeader

Data Flow Search Log Analytics
This is a sample that takes the U-SQL SearchLog analytics example and turns it into an ADF Data Flow: https://kromerbigdata.com/2019/03/03/u-sql-searchlog-aggregations-as-adf-data-flows/
by Microsoft

Dedupe Pipeline
This template uses ADF Mapping Data Flows that utilizes fuzzy matching combined with exact matching and distinct aggregations to score data based on likelihood to be duplicate matches from ...
by Microsoft

Fact Loader
This is a data flow example of how to load facts into your fact table from a single sample Employee dimension. There is also an example of handling early-arriving facts.
by Microsoft

Fixed Width Data Flow Pipeline
This is an example of creating a data flow to process fixed-width text files
by Microsoft

Movie Analytics
Use this template to get an understanding of how to leverage the capabilities of Copy and Mapping Data Flow in Azure Data Factory
...
by Microsoft

Parquet Crud Operations
Run CRUD operations on a parquet sink using Mapping Data Flow
The Data Flow takes the following Parquet inputs ...
by Microsoft

Rule-Based Dynamic Column Names
Sample template for using Mapping Data Flows in ADF to dynamically create column names in your data based on business rules. Sample data and instructions here...
by Microsoft

Slowly Changing Dimension Type 2
This is a sample pattern for SCD Type 2 that handles loading a dimension table in a target star schema analytical database. The sample data is located here...
by Microsoft

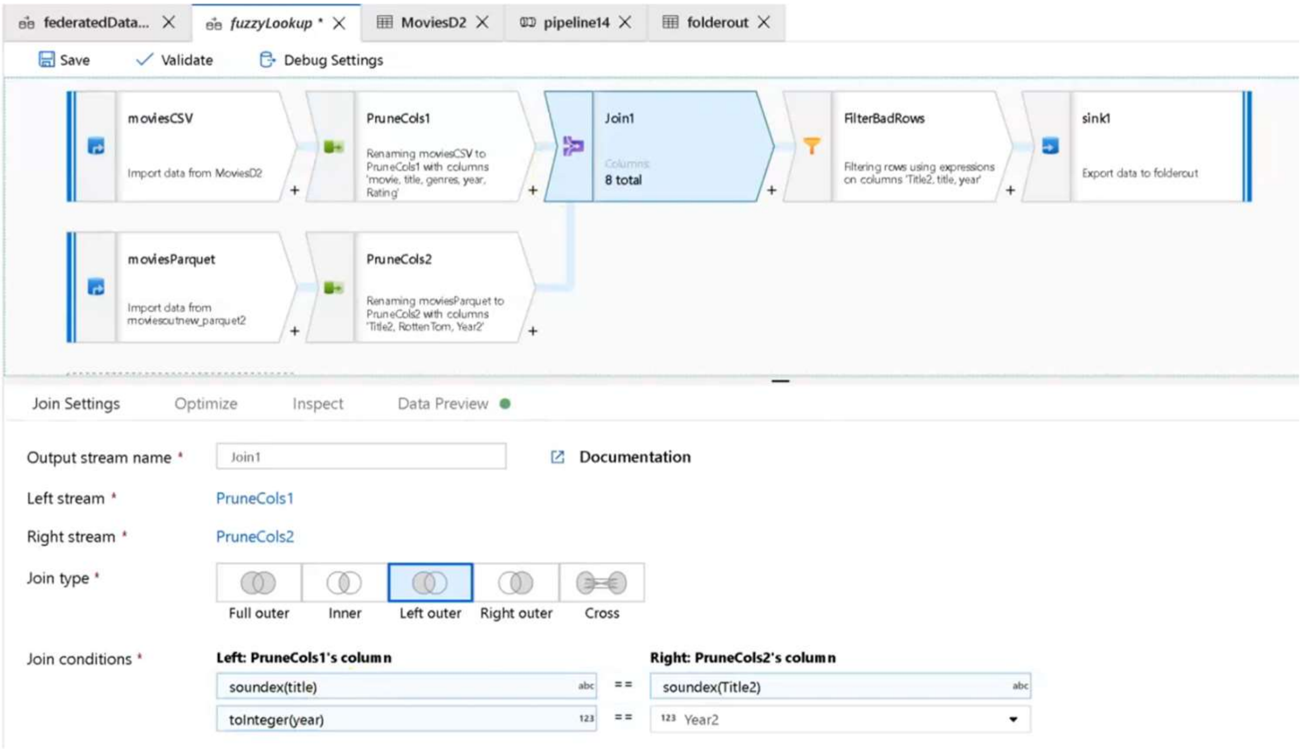
Transform data using data flow with copy ingest
Run dataflow with Explicit copy to ingest copy source
by Microsoft

Transform data using data flow
Use this template to deploy several sample data

- Classic ETL pattern is easy to build in ADF's code-free Data Flow visual data transformation environment
- Add Aggregate transforms to produce calculations that you store in your analytical database schema
- Use Join transform to combine data from multiple data sources and data streams inside your data flow
- Land your data in your Lake folders or direct to Azure SQL DW

Fuzzy Lookups

- Sometime when performing inline lookups, you don’t have exact matches when looking for references
- Fuzzy Lookups with Soundex helps find matches based on phonetic algorithms
- Very useful in data lake scenarios where joins and lookups are against data that is not normalized or cleaned


federatedData...
fuzzyLookup *
MoviesD2
pipeline14
folderout
Save
Validate
Debug Settings
moviesCSV
Import data from MoviesD2
PruneCols1
Renaming moviesCSV to PruneCols1 with columns 'movie, title, genres, year, Rating'
Join1
Columns
8 total
FilterBadRows
Filtering rows using expressions on columns 'Title2, title, year'
sink1
Export data to folderout
moviesParquet
Import data from moviescutnew_parquet2
PruneCols2
Renaming moviesParquet to PruneCols2 with columns 'Title2, RottenTom, Year2'
Join Settings
Optimize
Inspect
Data Preview
Documentation
Full outer
Inner
Left outer
Right outer
Cross

Output stream name * Join1

Left stream * PruneCols1

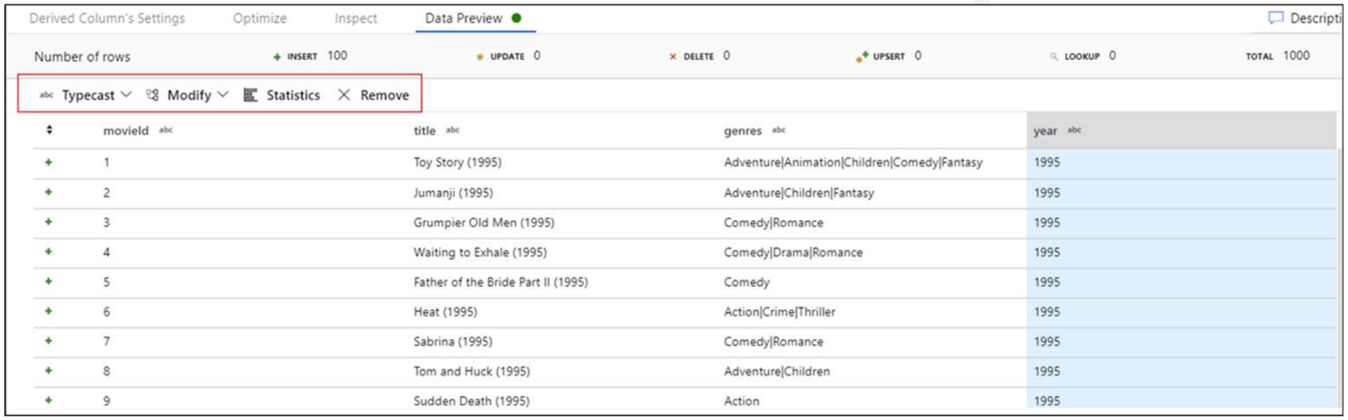
Right stream * PruneCols2

Join type * Left outer

Join conditions *

Left: PruneCols1's column		Right: PruneCols2's column
soundex(title) abc	==	soundex(Title2) abc
toInteger(year) 123	==	123 Year2

Data Lake Data Science Scenario

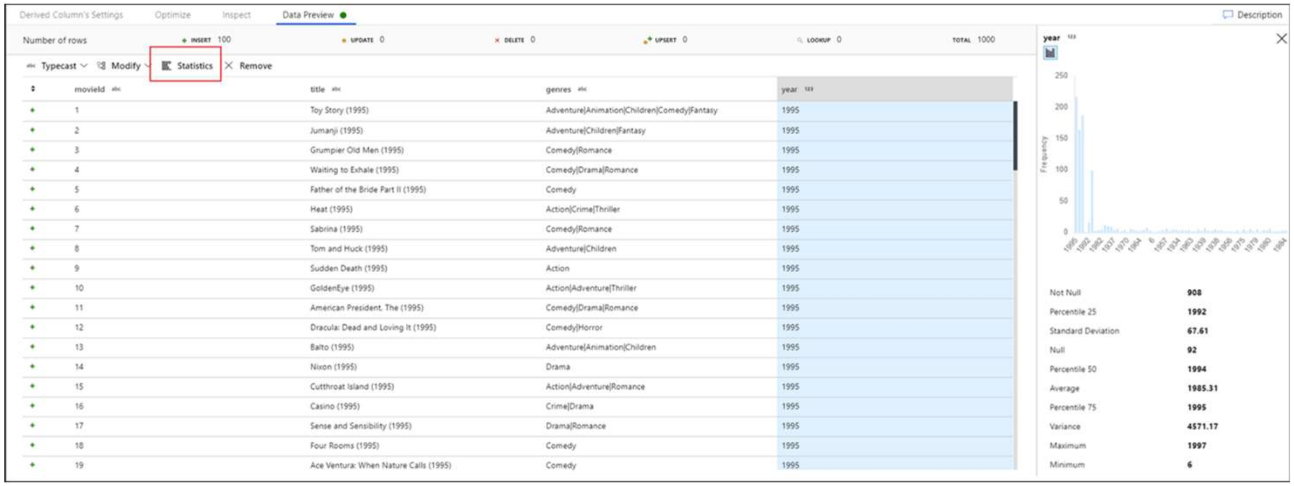


Derived Column's Settings · Optimize · Inspect · Data Preview ● · Descripti

Number of rows · INSERT 100 · UPDATE 0 · DELETE 0 · UPSERT 0 · LOOKUP 0 · TOTAL 1000

abc Typecast ˅ · Modify ˅ · Statistics · ✕ Remove

⇕	movieId abc	title abc	genres abc	year abc
+	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1995
+	2	Jumanji (1995)	Adventure|Children|Fantasy	1995
+	3	Grumpier Old Men (1995)	Comedy|Romance	1995
+	4	Waiting to Exhale (1995)	Comedy|Drama|Romance	1995
+	5	Father of the Bride Part II (1995)	Comedy	1995
+	6	Heat (1995)	Action|Crime|Thriller	1995
+	7	Sabrina (1995)	Comedy|Romance	1995
+	8	Tom and Huck (1995)	Adventure|Children	1995
+	9	Sudden Death (1995)	Action	1995



Derived Column's Settings · Optimize · Inspect · Data Preview ● · Description

Number of rows · INSERT 100 · UPDATE 0 · DELETE 0 · UPSERT 0 · LOOKUP 0 · TOTAL 1000

abc Typecast ˅ · Modify ˅ · Statistics · ✕ Remove

⇕	movieId abc	title abc	genres abc	year 123
+	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1995
+	2	Jumanji (1995)	Adventure|Children|Fantasy	1995
+	3	Grumpier Old Men (1995)	Comedy|Romance	1995
+	4	Waiting to Exhale (1995)	Comedy|Drama|Romance	1995
+	5	Father of the Bride Part II (1995)	Comedy	1995
+	6	Heat (1995)	Action|Crime|Thriller	1995
+	7	Sabrina (1995)	Comedy|Romance	1995
+	8	Tom and Huck (1995)	Adventure|Children	1995
+	9	Sudden Death (1995)	Action	1995
+	10	GoldenEye (1995)	Action|Adventure|Thriller	1995
+	11	American President, The (1995)	Comedy|Drama|Romance	1995
+	12	Dracula: Dead and Loving It (1995)	Comedy|Horror	1995
+	13	Balto (1995)	Adventure|Animation|Children	1995
+	14	Nixon (1995)	Drama	1995
+	15	Cutthroat Island (1995)	Action|Adventure|Romance	1995
+	16	Casino (1995)	Crime|Drama	1995
+	17	Sense and Sensibility (1995)	Drama|Romance	1995
+	18	Four Rooms (1995)	Comedy	1995
+	19	Ace Ventura: When Nature Calls (1995)	Comedy	1995

Not Null	908
Percentile 25	1992
Standard Deviation	67.61
Null	92
Percentile 50	1994
Average	1985.31
Percentile 75	1995
Variance	4571.17
Maximum	1997
Minimum	6

- ADF supports building visual data transformations against your data directly in Data Lake locations (i.e. Azure Blob Store, Azure Data Lake Store)
- Built-in handling of schema drift for frequent changes in data lake file formats, columns, and data types
- Perform data exploration and data profiling across your data lake in ADF Data Flow with interactive debug data preview and quick actions

Schema Drift

Schema Drift

In most real-world data integration solutions, source and target data stores will change shape

- Source data fields will change name
- Number of columns will change over time

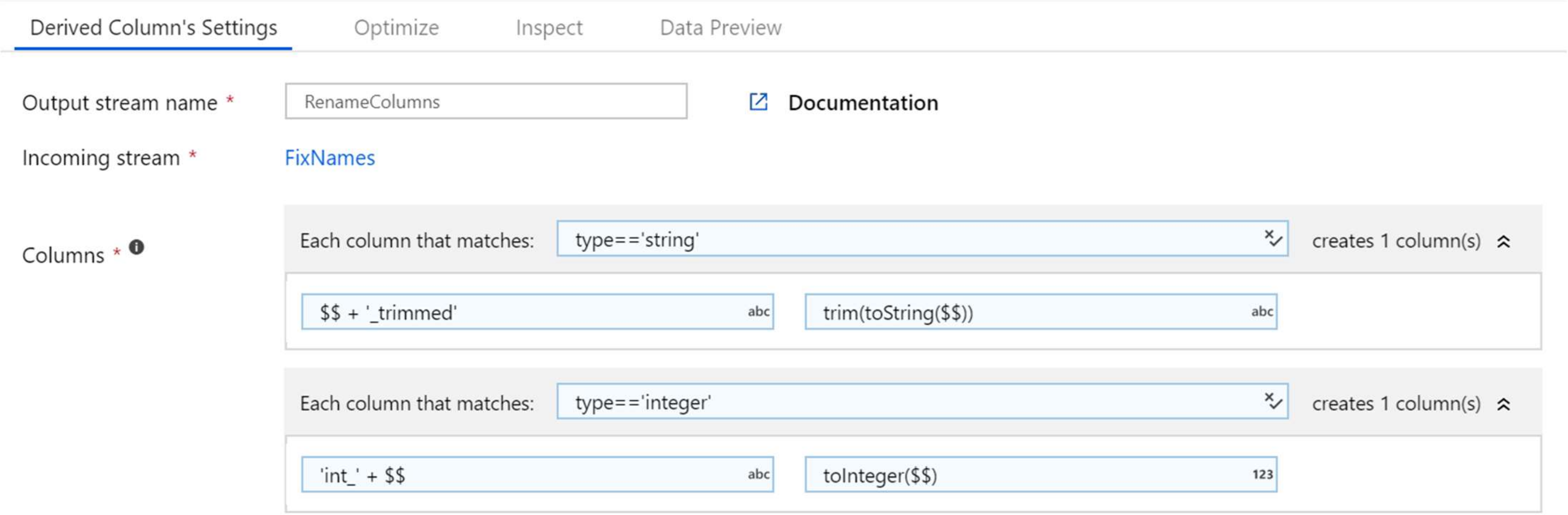
Traditional ETL processes break when schemas drift

Mapping Data Flow has built-in facilities for flexible schemas to handle schema drift

- Patterns, rule-based mapping, byName function
- Source: Read additional columns on top of what is defined in the dataset source
- Sink: Write additional columns on top of what is defined in the dataset sink

Pattern matching

- Match by name, type, stream, ordinal position



Output stream name * RenameColumns

Incoming stream * FixNames

Columns *

Each column that matches: type=='string' creates 1 column(s)

$$ + '_trimmed' abc | trim(toString($$)) abc

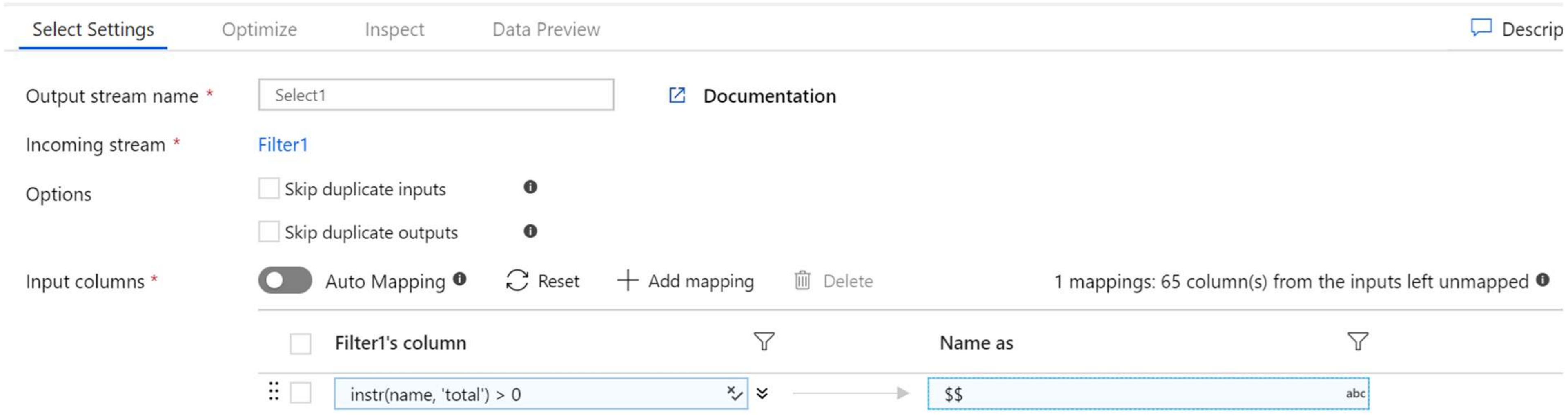
Each column that matches: type=='integer' creates 1 column(s)

'int_' + $$ abc | toInteger($$) 123

Executive Education

Rule-based mapping

- Rather than pick and choose columns for transformations one-by-one, build policies that collect columns based on matching rules.



Select Settings | Optimize | Inspect | Data Preview | Descrip

Output stream name * Select1 Documentation

Incoming stream * Filter1

Options
□ Skip duplicate inputs
□ Skip duplicate outputs

Input columns * Auto Mapping | Reset | + Add mapping | Delete | 1 mappings: 65 column(s) from the inputs left unmapped

□	Filter1's column	Name as
□	instr(name, 'total') > 0	$$

Prepare Practical Classes

Prepare Practical Classes

- Access the URL https://portal.azure.com/#@isegulisboacloud.onmicrosoft.com/resource/subscriptions/000a9103-fec4-4ce8-aba5-d7b0908f5aca/resourceGroups/DSBAFB2_DW/overview

- User your Power BI student Account ex. ixxxxx@students.isegexecutive.education
- Accept the access to Azure Portal resources
- Validate you are able to see the resource group DW

Resources

- Book: UNDERSTANDING AZURE DATA FACTORY, Rawat, Sudhir, Narain, Abhishek
- Patterns: http://aka.ms/dataflowpatterns
- Documentation: https://docs.microsoft.com/en-us/azure/data-factory/concepts-data-flow-overview

Executive Education

www.isegexecutive.education
Rua do Quelhas, 6
1200-781 Lisboa

(+351) 213 922 891
info@executive.education