Combining and Shaping Data

EXPLORING TECHNIQUES TO COMBINE AND SHAPE DATA

Overview

Wide data formats

Long data formats

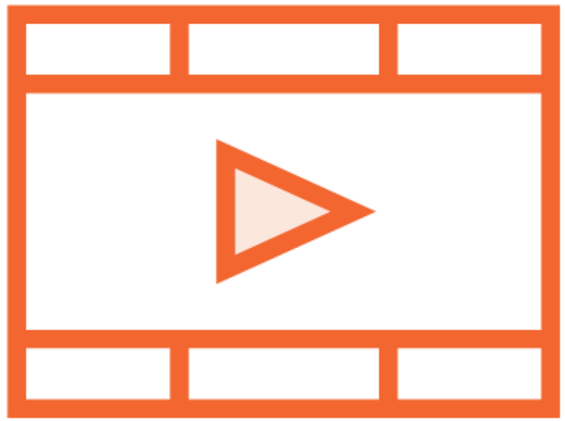
Types and uses of joins

Grouping operations

Aggregation operations

Prerequisites and Course Outline

Prerequisites



Basic Excel spreadsheets

Basic Python programming

Basic SQL to work with relational databases and data warehouses

High school math

Course Outline

Principles in combining and shaping data

Combining and shaping data in

- Spreadsheets
- Relational databases
- Python

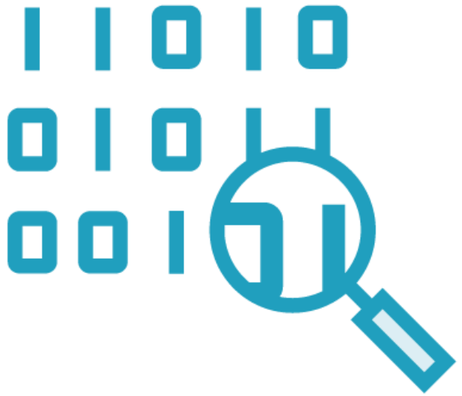
Integrating disparate sources using ETL pipelines

Working with streaming data using a data warehouse

“My mind is made up. Don’t confuse me with the facts.”

Some powerful person

Thoughtful, Fact-based Point of View



Fact-based

Built with painstakingly collected data

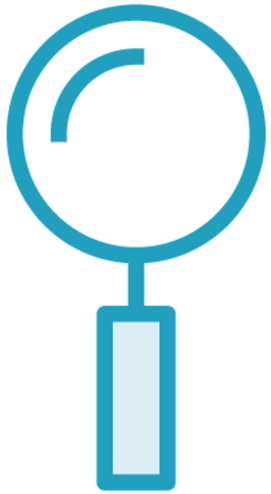
Thoughtful

Balanced, weighing pros and cons

Point of View

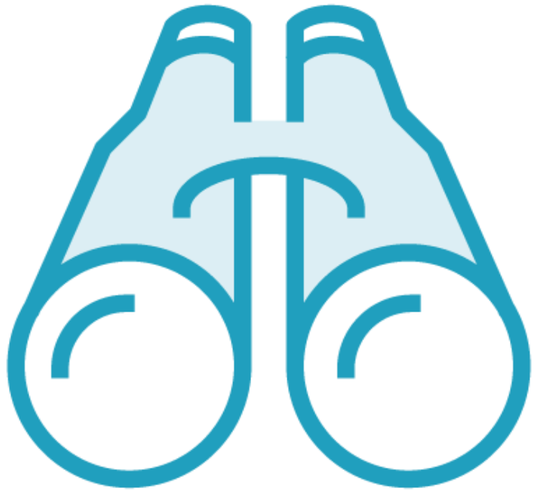
Prediction, recommendation, call to action

Two Sets of Statistical Tools



Descriptive Statistics

Identify important elements in a dataset



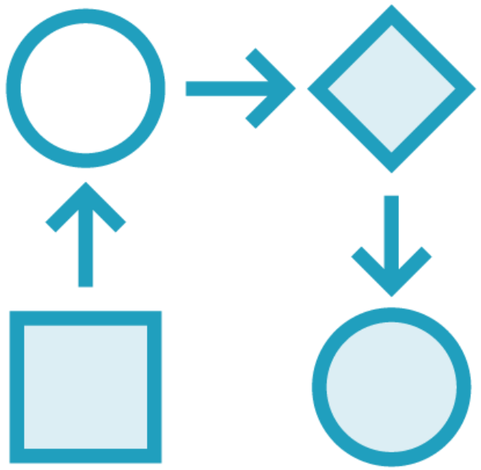
Inferential Statistics

Explain those elements via relationships with other elements

Two Hats of a Data Professional

Find the Dots

Identify important elements in a dataset



Connect the Dots

Explain those elements via relationships with other elements

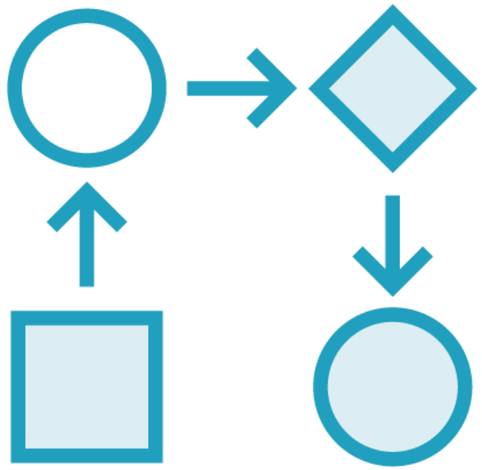
Finding the Dots

Data is more and more plentiful

However careful handling is needed

- Missing values
- Outliers
  - Genuine outliers
  - Erroneously measured points

Connecting the Dots



Spreadsheets

Programming languages

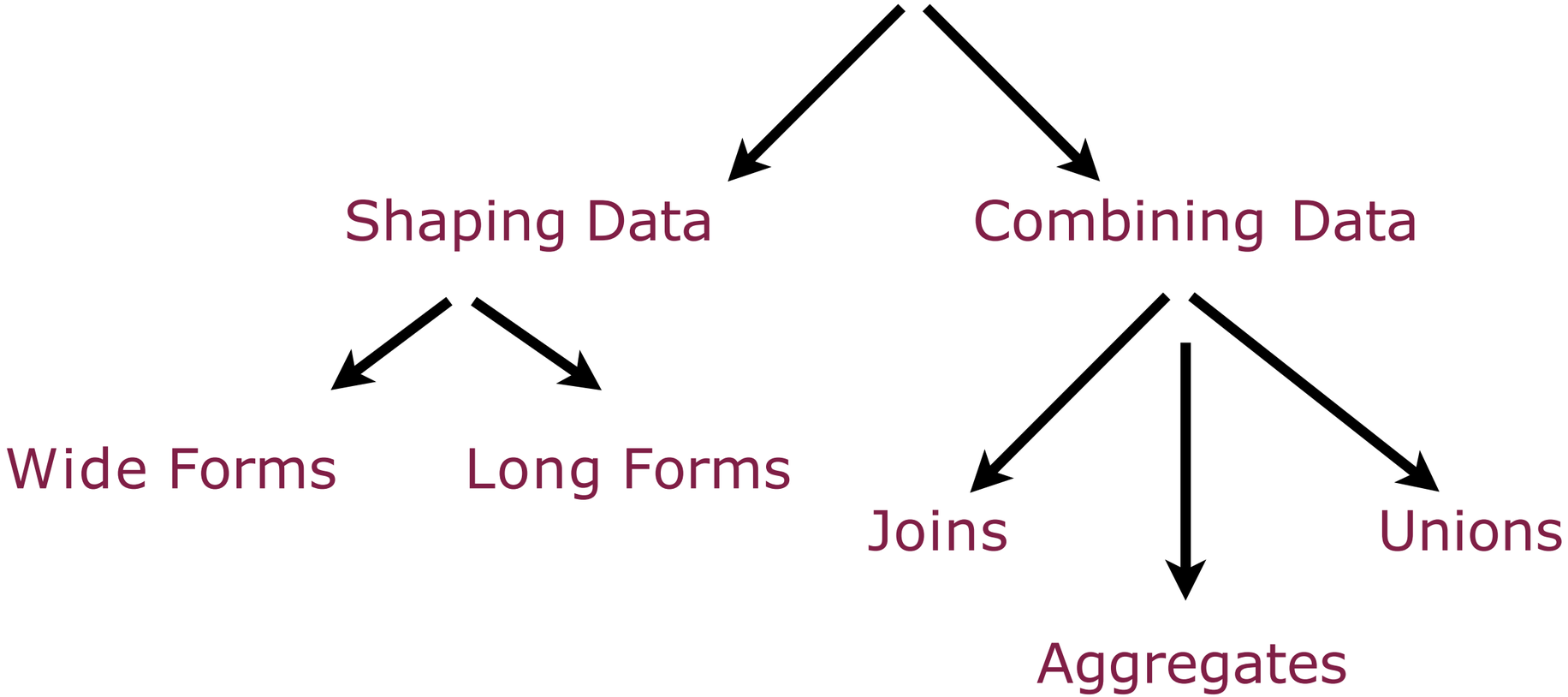
- In-memory processing
- Distributed processing

SQL

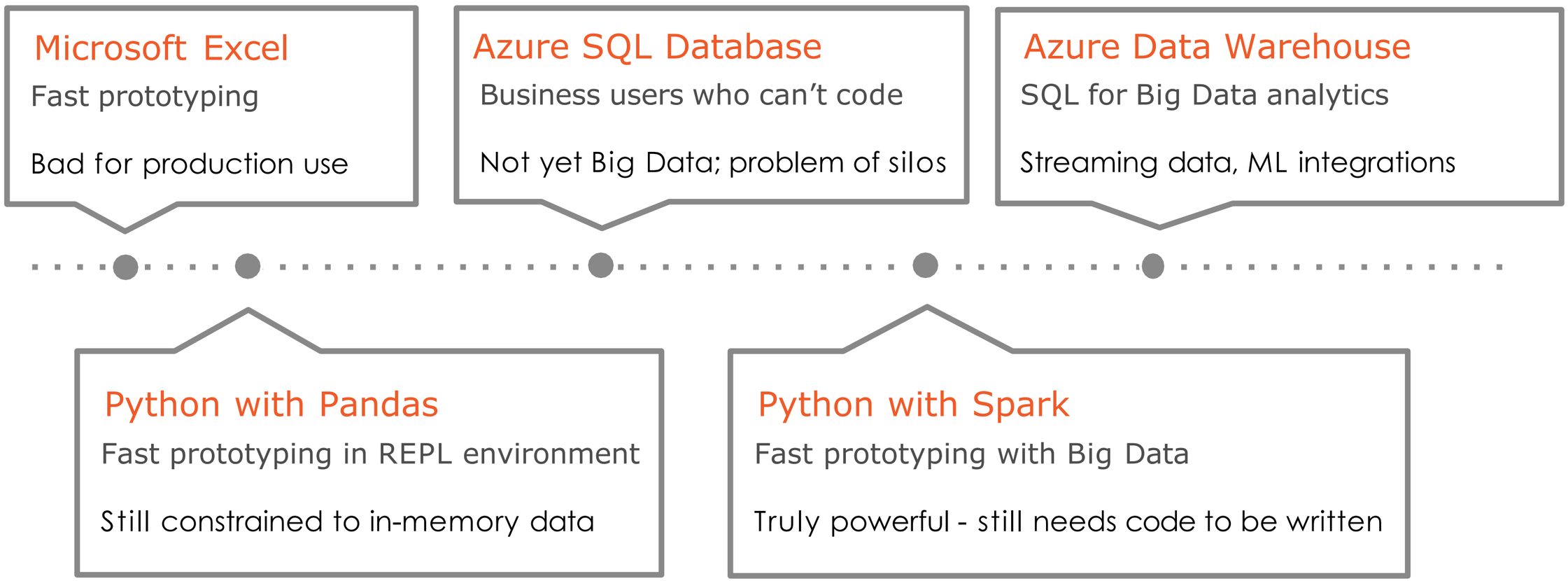
- Relational databases
- Data warehouses

Combining and shaping data smartly make it easy to connect the dots

Connecting the Dots

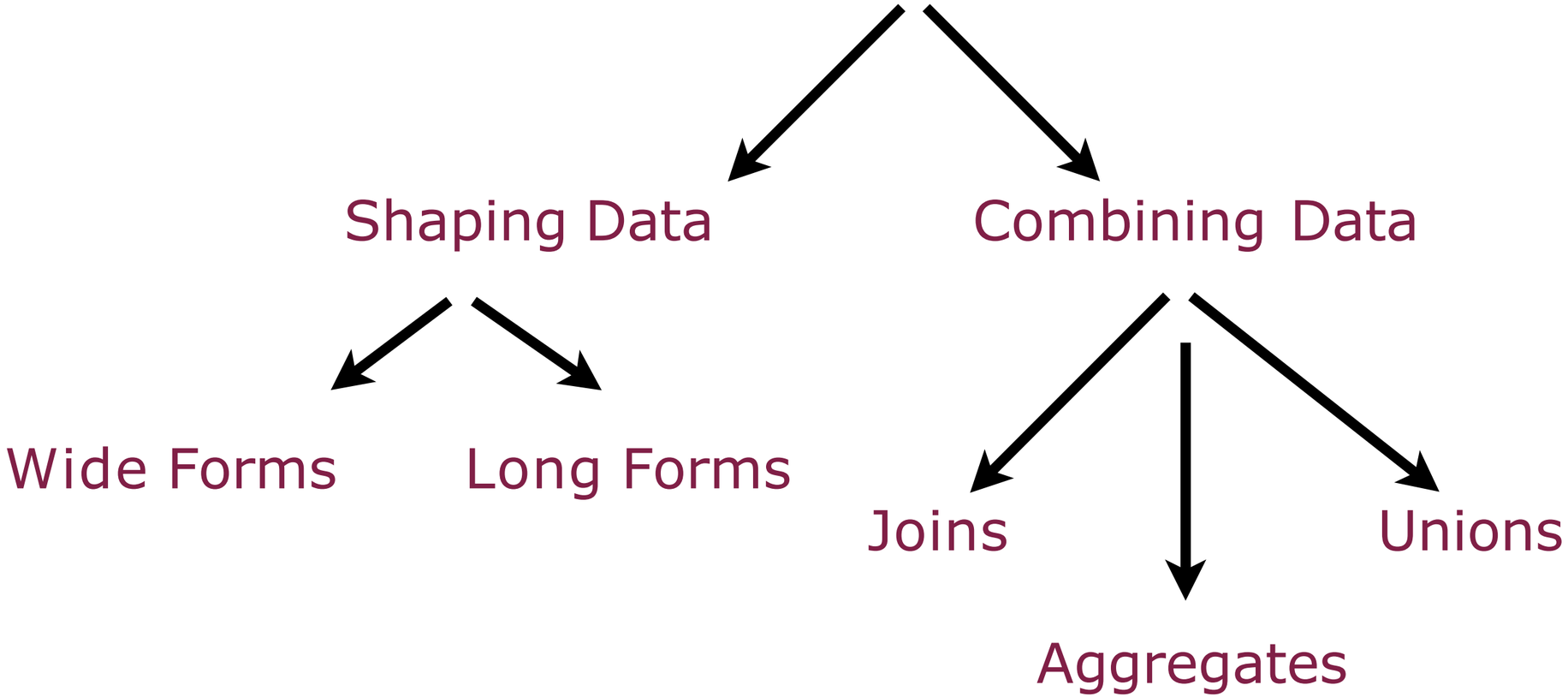


Choices of Technology

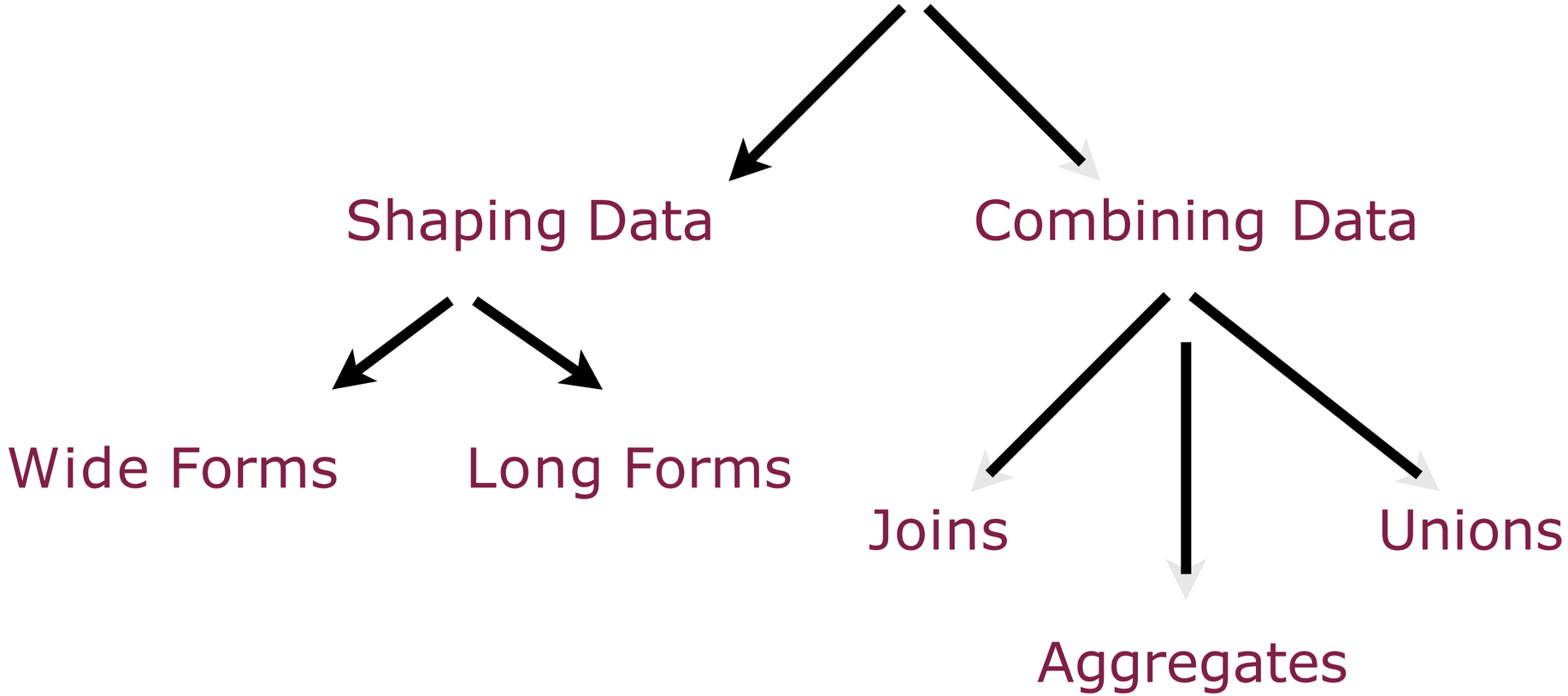


Wide vs. Long Data

Connecting the Dots



Connecting the Dots



Traditional Wide Form

Id	To	Type	Content
1	mike	offer	Mobile offer
2	john	sale	Redmi sale
3	jill	order	Order delivered
4	megan	sale	Clothes sale

As columns are added, table gets wider

Traditional Wide Form

Id	To	Type	Content
1	mike	offer	Mobile offer
2	john	sale	Redmi sale
3	jill	order	Order delivered
4	megan	sale	Clothes sale

2-D indexing to access a particular value

From Wide To Long

Id	To	Type	Content
1	mike	offer	Mobile offer
2	john	sale	Redmi sale
3	jill	order	Order delivered
4	megan	sale	Clothes sale

Id	Column	Value
1	To	mike
1	Type	offer
1	Content	Mobile offer
2	To	john
2	Type	sale
2	Content	Redmi sale
3	To	jill
3	Type	order
3	Content	Order delivered
4	To	megan
4	Type	sale
4	Content	Clothes sale

From Wide To Long

Id	To	Type	Content
1	mike	offer	Mobile offer
2	john	sale	Redmi sale
3	jill	order	Order delivered
4	megan	sale	Clothes sale

Id	Column	Value
1	To	mike
1	Type	offer
1	Content	Mobile offer
2	To	john
2	Type	sale
2	Content	Redmi sale
3	To	jill
3	Type	order
3	Content	Order delivered
4	To	megan
4	Type	sale
4	Content	Clothes sale

Flexible Schema

Id	To	Type	Content
1	mike	offer	Mobile offer
2	john	sale	Redmi sale
3	jill	order	Order delivered
4	megan	sale	Clothes sale

Id	Column	Value
1	To	mike
1	Type	offer
1	Content	Mobile offer
2	To	john
2	Type	sale
2	Content	Redmi sale
3	To	jill
3	Type	order
3	Content	Order delivered
4	To	megan
4	Type	sale
4	Content	Clothes sale

Flexible Schema

Id	To	Type	Content
1	mike	offer	Mobile offer
2	john	sale	Redmi sale
3	jill	order	Order delivered
4	megan	sale	Clothes sale

Id	Column	Value
1	To	mike
1	Type	offer
1	Content	Mobile offer
2	To	john
2	Type	sale
2	Content	Redmi sale
3	To	jill
3	Type	order
3	Content	Order delivered
4	To	megan
4	Type	sale
4	Content	Clothes sale

Multiple Rows Per Record

Id	To	Type	Content
1	mike	offer	Mobile offer
2	john	sale	Redmi sale
3	jill	order	Order delivered
4	megan	sale	Clothes sale

Id	Column	Value
1	To	mike
1	Type	offer
1	Content	Mobile offer
2	To	john
2	Type	sale
2	Content	Redmi sale
3	To	jill
3	Type	order
3	Content	Order delivered
4	To	megan
4	Type	sale
4	Content	Clothes sale

Efficient for Sparse Data

Id	To	Type	Content
1	mike		Mobile offer
2	john	sale	Redmi sale
3	jill	order	
4	megan	sale	

Id	Column	Value
1	To	mike
1	Content	Mobile offer
2	To	john
2	Type	sale
2	Content	Redmi sale
3	To	jill
3	Content	Order delivered
4	To	megan
4	Type	sale

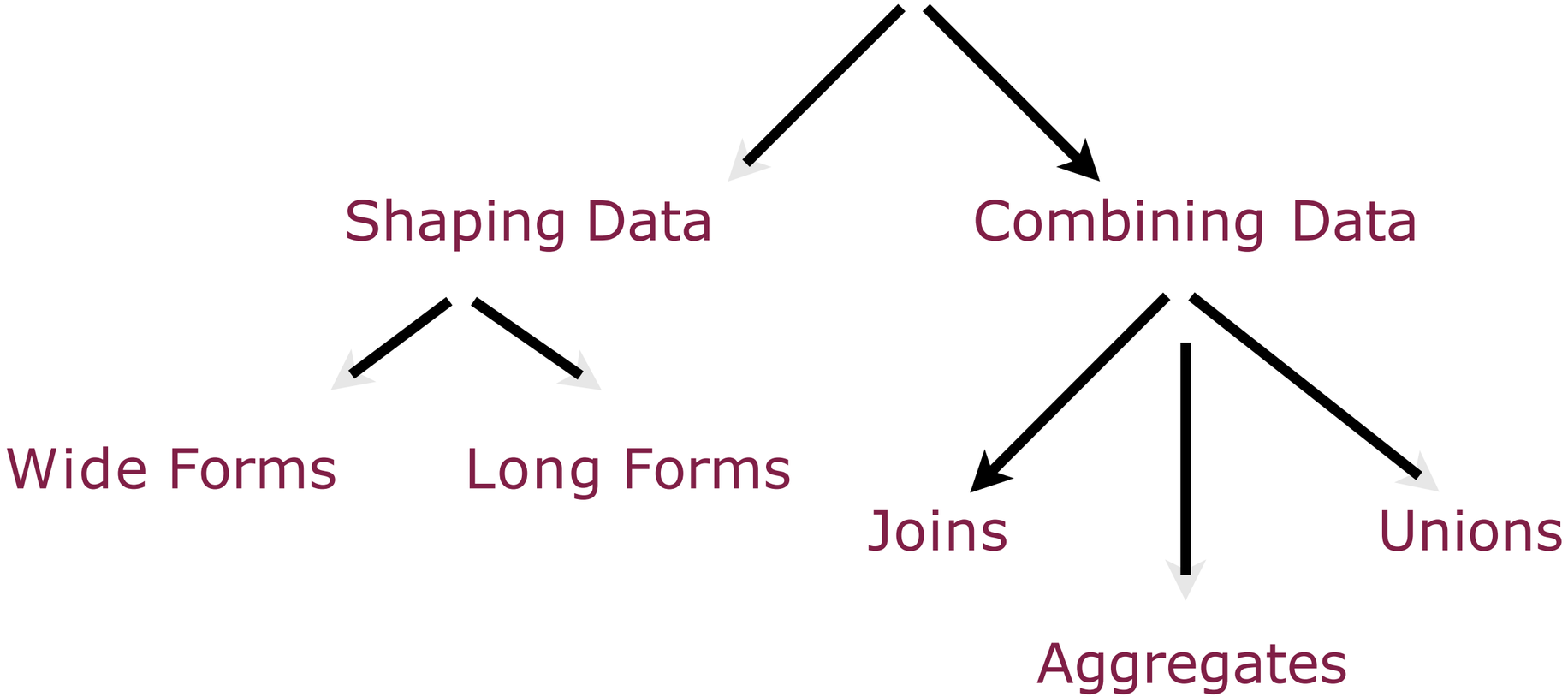
Missing values are not stored in the table

Choose wide form for
well-defined, **dense** data

Choose long form for **sparse**
data and **flexible** schema

Joins

Connecting the Dots



Joins

Name	Salary
Tom	1
John	1
Judy	150m

Name	Department
Judy	Google
Tom	GoogleX
John	Alphabet

Joins

Name	Salary	Department
Tom	1	GoogleX
John	1	Alphabet
Judy	150m	Google

Records from each relation matched on the join column

Joins

Name	Salary	Department
Tom	1	GoogleX
John	1	Alphabet
Judy	150m	Google

The most common type are equi-joins

Types of Joins

Left Outer Join

Right Outer Join

Full Outer Join

Self Join

Cross Join

Types of Joins

Left Outer Join

Right Outer Join

Full Outer Join

Self Join

Cross Join

Left Outer Join

Name	Salary
Tom	1
John	1
Judy	150m

Name	Department
Emily	Google
John	GoogleX
Tom	Alphabet

Left Outer Join

Name	Salary
Tom	1
John	1
Judy	150m

Every record on the left table will be present in the result

- with a matching record
- padded with nulls

Left Outer Join

Name	Salary	Department
Tom	1	Alphabet
John	1	GoogleX
Judy	150m	NULL

Types of Joins

Left Outer Join

Right Outer Join

Full Outer Join

Self Join

Cross Join

Right Outer Join

Name	Salary
Tom	1
John	1
Judy	150m

Name	Department
Emily	Google
John	GoogleX
Tom	Alphabet

Right Outer Join

Every record on the right table will be present in the result

- with a matching record
- padded with nulls

Name	Department
Emily	Google
John	GoogleX
Tom	Alphabet

Right Outer Join

Name	Salary	Department
Emily	NULL	Google
John	1	GoogleX
Tom	1	Alphabet

Types of Joins

Left Outer Join

Right Outer Join

Full Outer Join

Self Join

Cross Join

Full Outer Join

Name	Salary
Tom	1
John	1
Judy	150m

Name	Department
Emily	Google
John	GoogleX
Tom	Alphabet

Full Outer Join

Name	Salary
Tom	1
John	1
Judy	150m

Name	Department
Emily	Google
John	GoogleX
Tom	Alphabet

Records from both tables will be present in the result

- with a matching record
- padded with nulls

Full Outer Join

Name	Salary	Department
Emily	NULL	Google
John	1	GoogleX
Tom	1	Alphabet
Judy	150m	NULL

Types of Joins

Left Outer Join

Right Outer Join

Full Outer Join

Self Join

Cross Join

Self Join

Name	Salary
Tom	1
John	1
Judy	150m

Name	Salary
Tom	1
John	1
Judy	150m

Self Join

Name	Salary	Salary
Tom	1	1
John	1	1
Judy	150m	150m

Types of Joins

Left Outer Join

Right Outer Join

Full Outer Join

Self Join

Cross Join

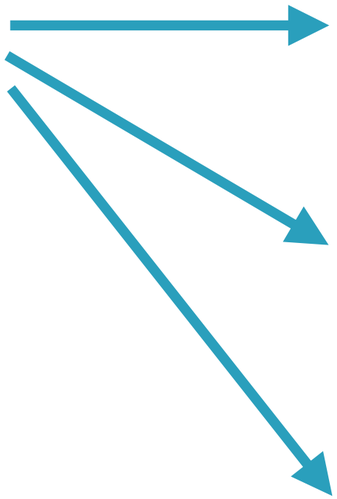
Cross Join

Name	Salary
Tom	1
John	1
Judy	150m

Name	Department
Emily	Google
John	GoogleX
Tom	Alphabet

Cross Join

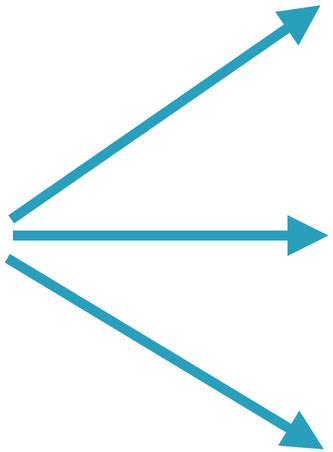
Name	Salary
Tom	1
John	1
Judy	150m



Name	Department
Emily	Google
John	GoogleX
Tom	Alphabet

Cross Join

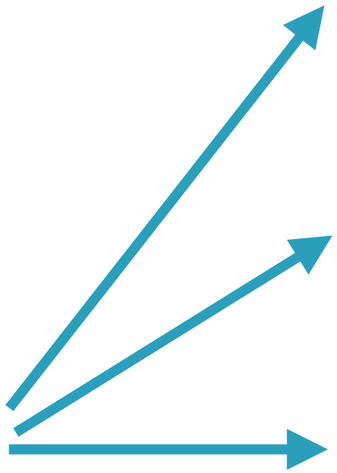
Name	Salary
Tom	1
John	1
Judy	150m



Name	Department
Emily	Google
John	GoogleX
Tom	Alphabet

Cross Join

Name	Salary
Tom	1
John	1
Judy	150m



Name	Department
Emily	Google
John	GoogleX
Tom	Alphabet

Cross Join

Name	Salary	Name	Department
Tom	1	Emily	Google
John	1	John	GoogleX
Judy	150m	Tom	Alphabet
Tom	1	Emily	Google
John	1	John	GoogleX
Judy	150m	Tom	Alphabet
Tom	1	Emily	Google
John	1	John	GoogleX
Judy	150m	Tom	Alphabet

Types of Joins

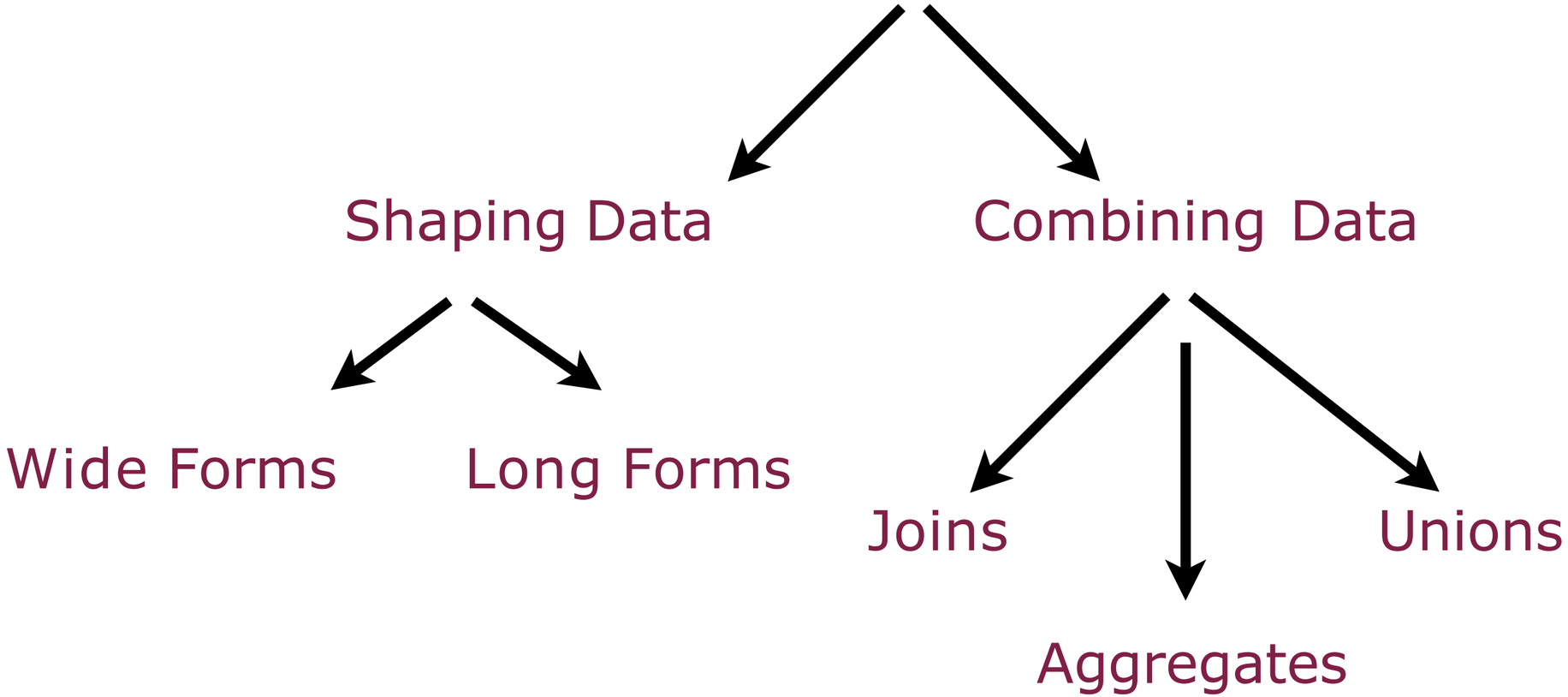
One-to-one

One-to-many

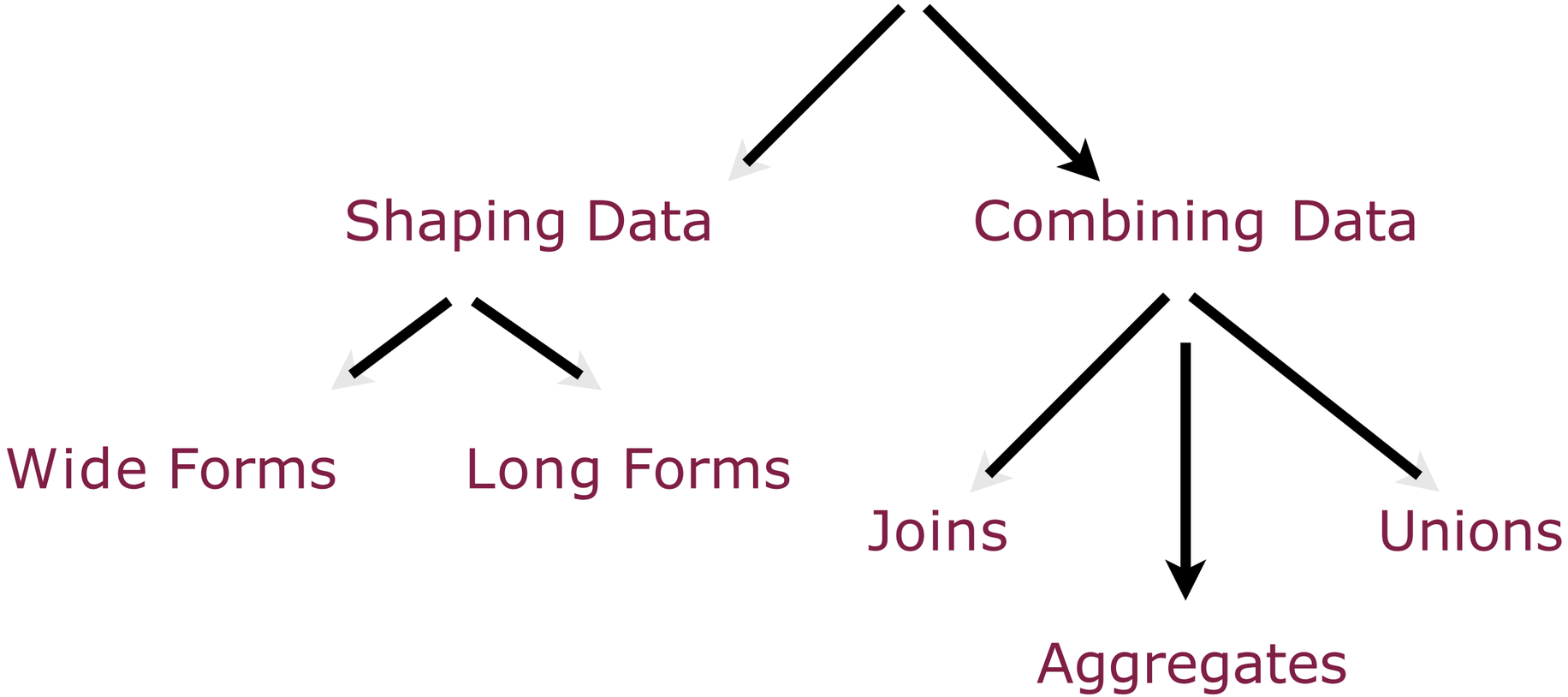
Many-to-many

Aggregations

Connecting the Dots



Connecting the Dots



Group By

ID	Product_ID	Quantity	Amount
o1	phone	1	199
o1	shoes	1	69
o2	book	2	22
o3	phone	1	149
o3	belt	2	19

Record of fields

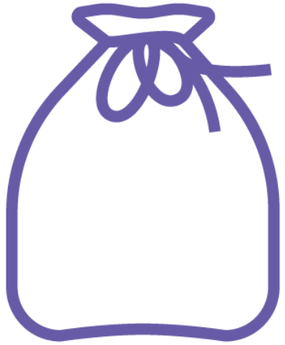
Group By

ID	Product_ID	Quantity	Amount
o1	phone	1	199
o1	shoes	1	69
o2	book	2	22
o3	phone	1	149
o3	belt	2	19

group orders by ID

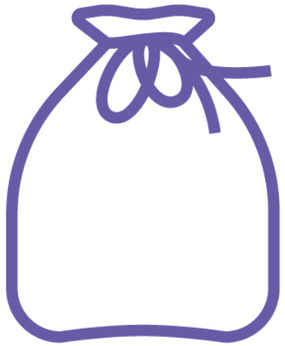
Group By

o1



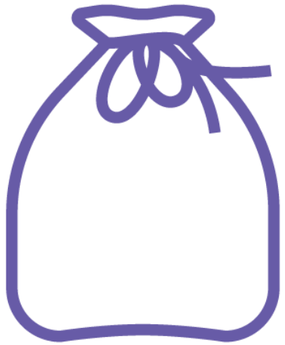
o1	phone	1	199
o1	shoes	1	69

o2



o2	book	2	22

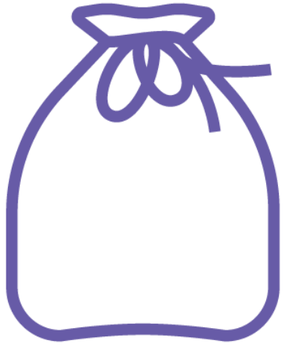
o3



o3	phone	1	149
o3	belt	2	19

Group By

o1

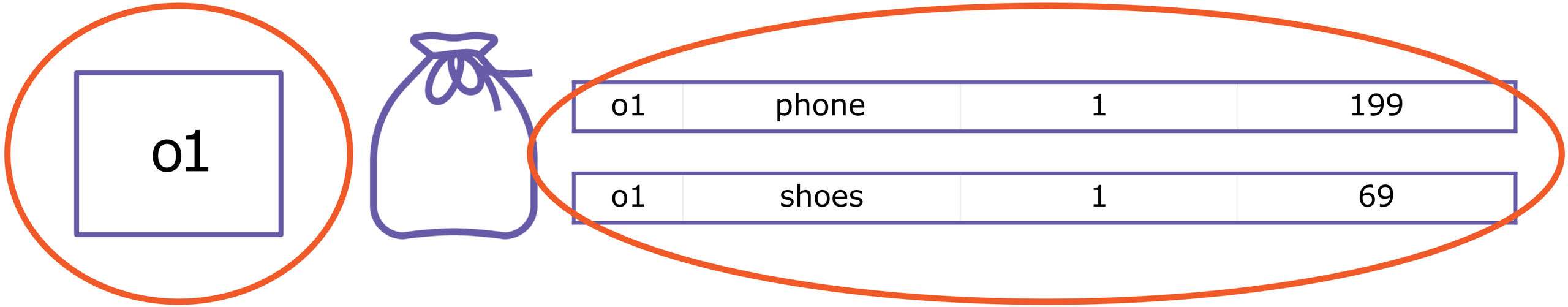


o1	phone	1	199

o1	shoes	1	69

All records with the same key are grouped

Group By

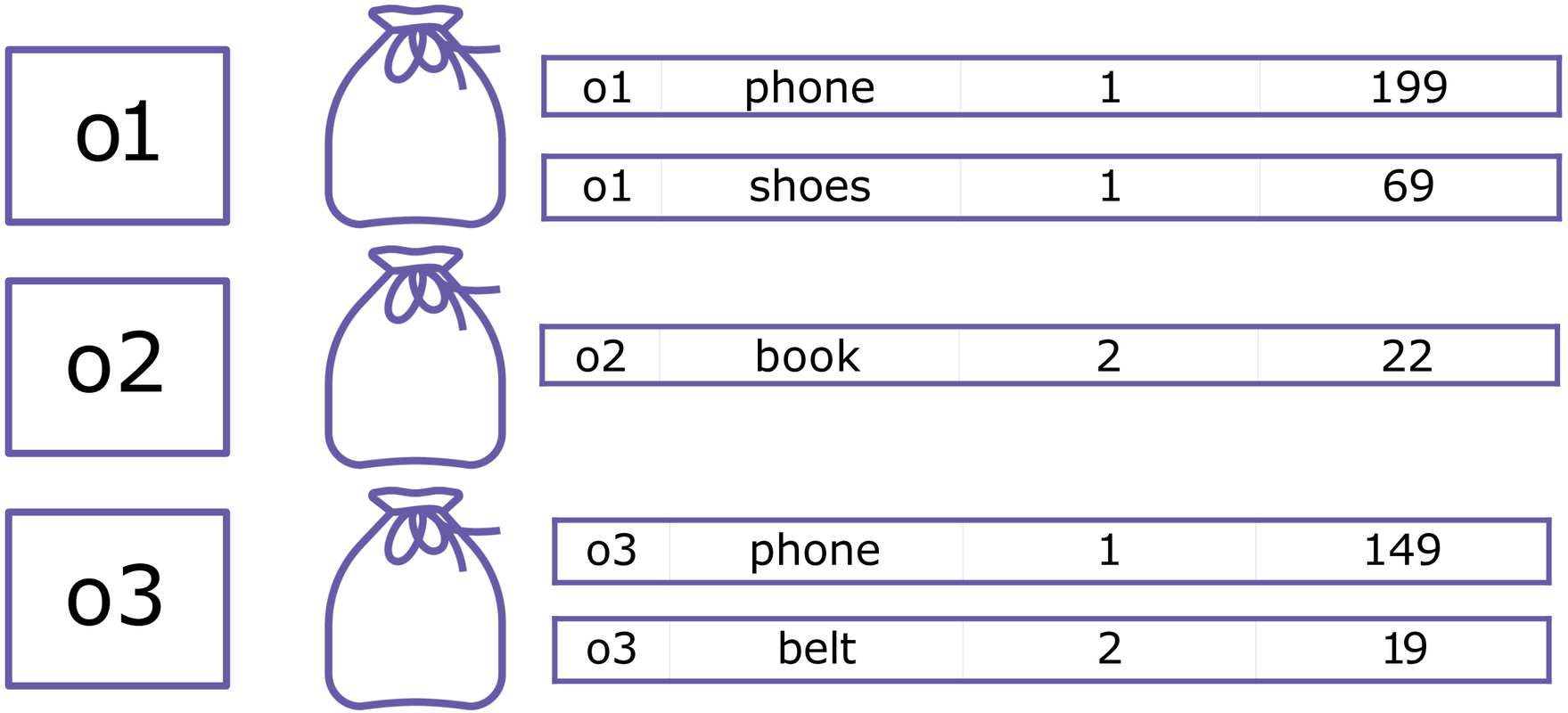


`group orders by ID` creates a grouping with:

the field used in group by i.e. the key

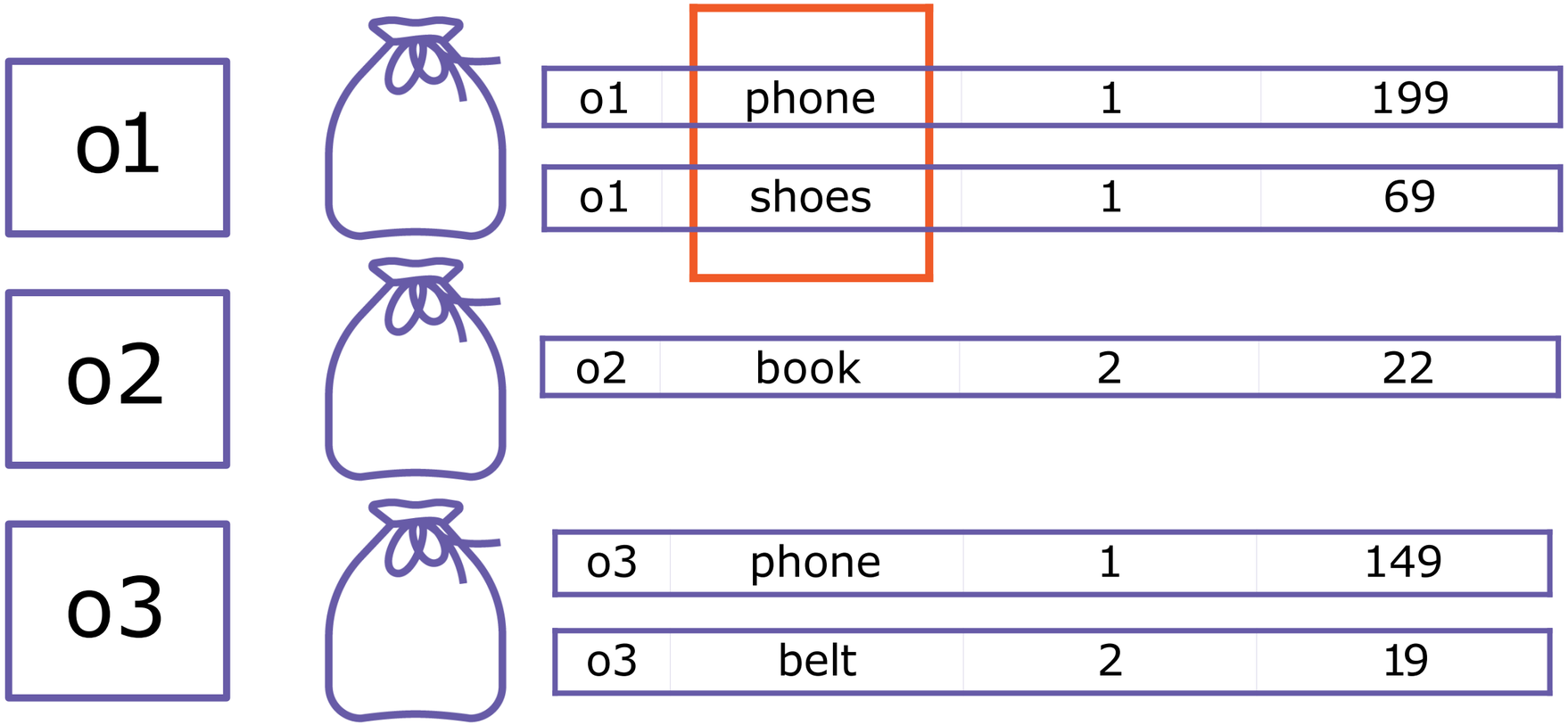
the values associated with the same key

Aggregations on Groups



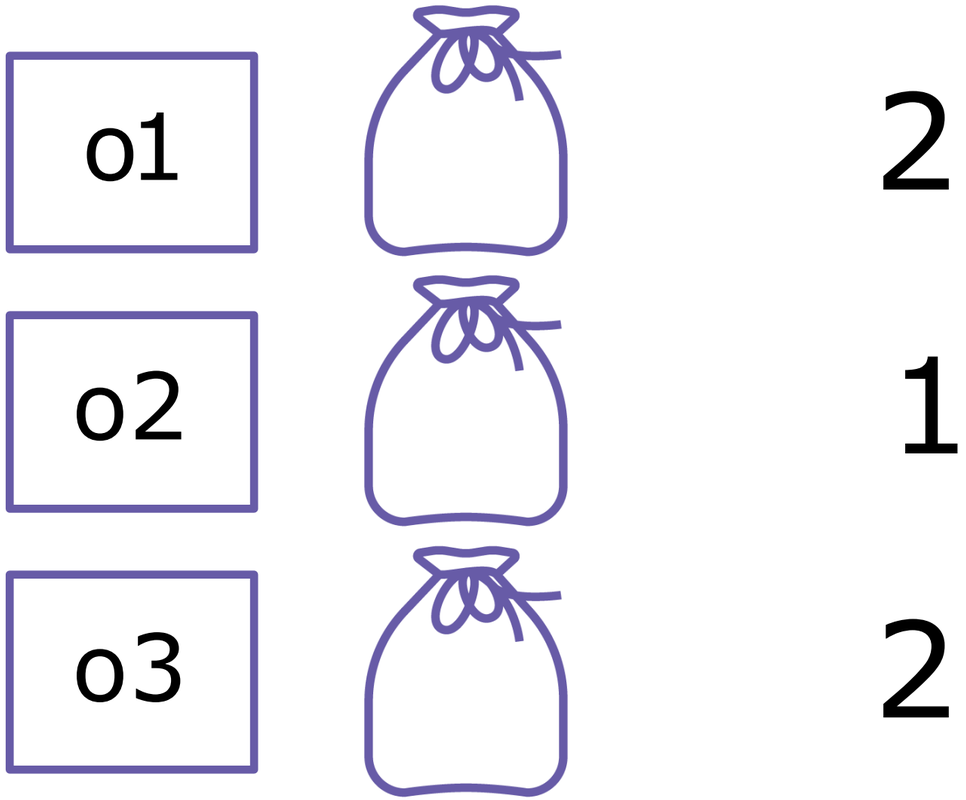
Aggregations are functions which can be applied to field values from multiple records which belong to a group

Aggregations on Groups



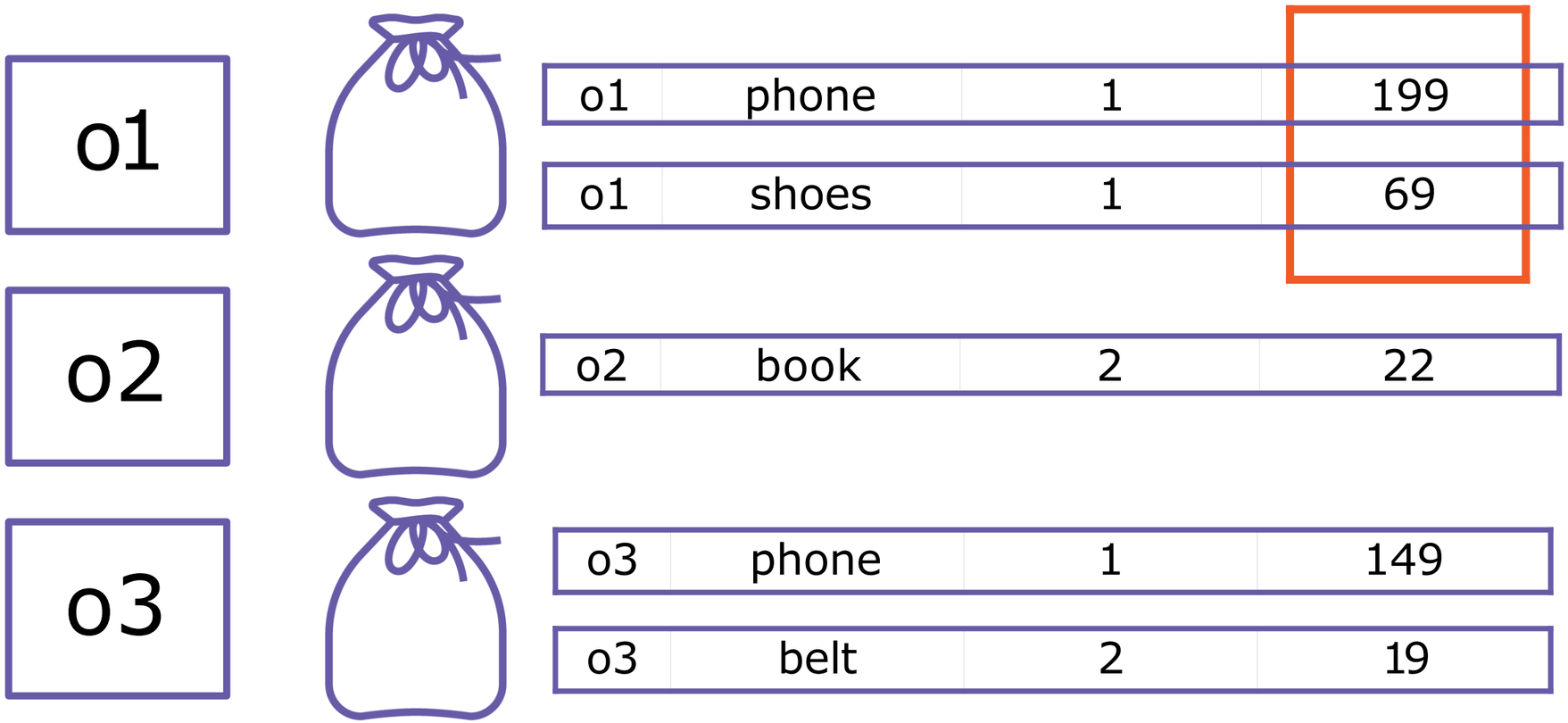
COUNT() the number of different products in each order

Aggregations on Groups



COUNT() the number of different products in each order

Aggregations on Groups



SUM() the total amount spent per order

Aggregations on Groups

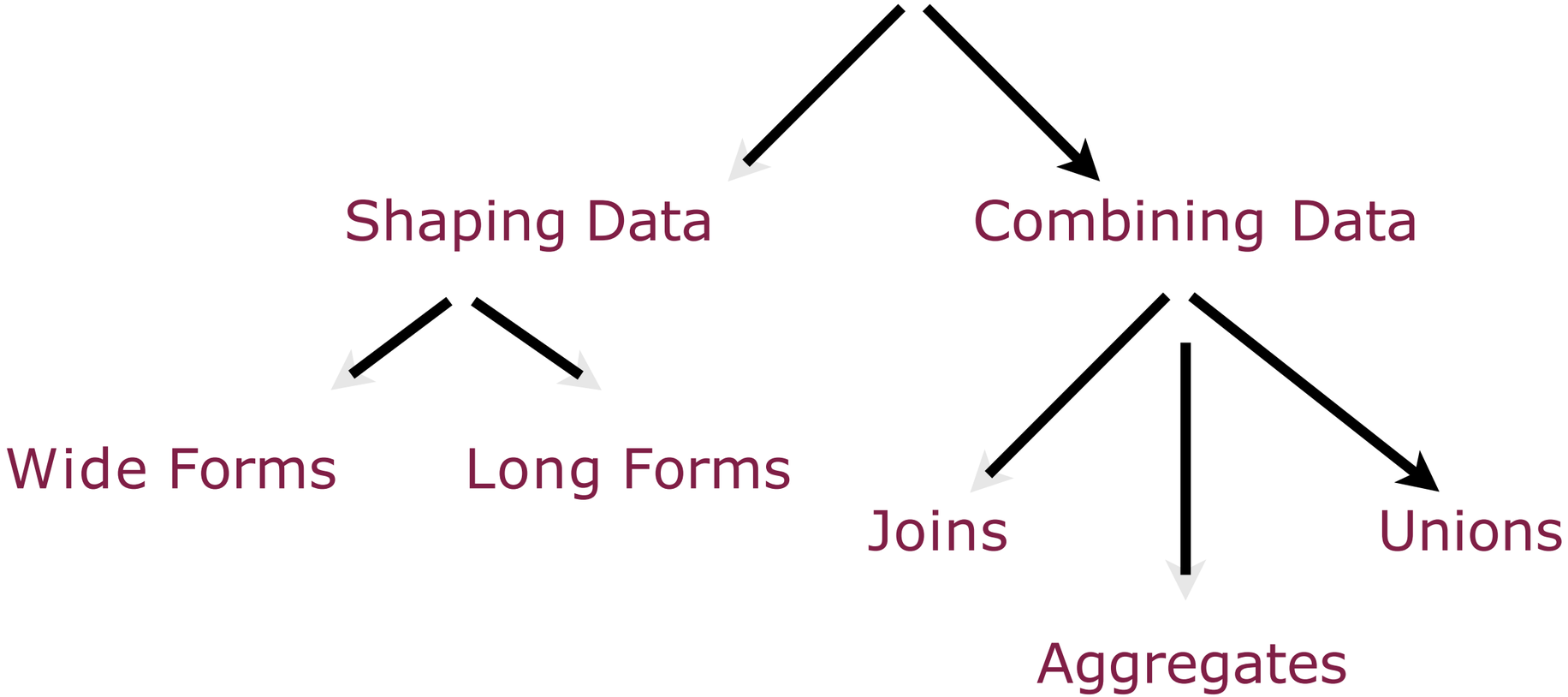
SUM() the total amount spent per order

Pivot operations convert data from long form to wide form data to enable aggregations

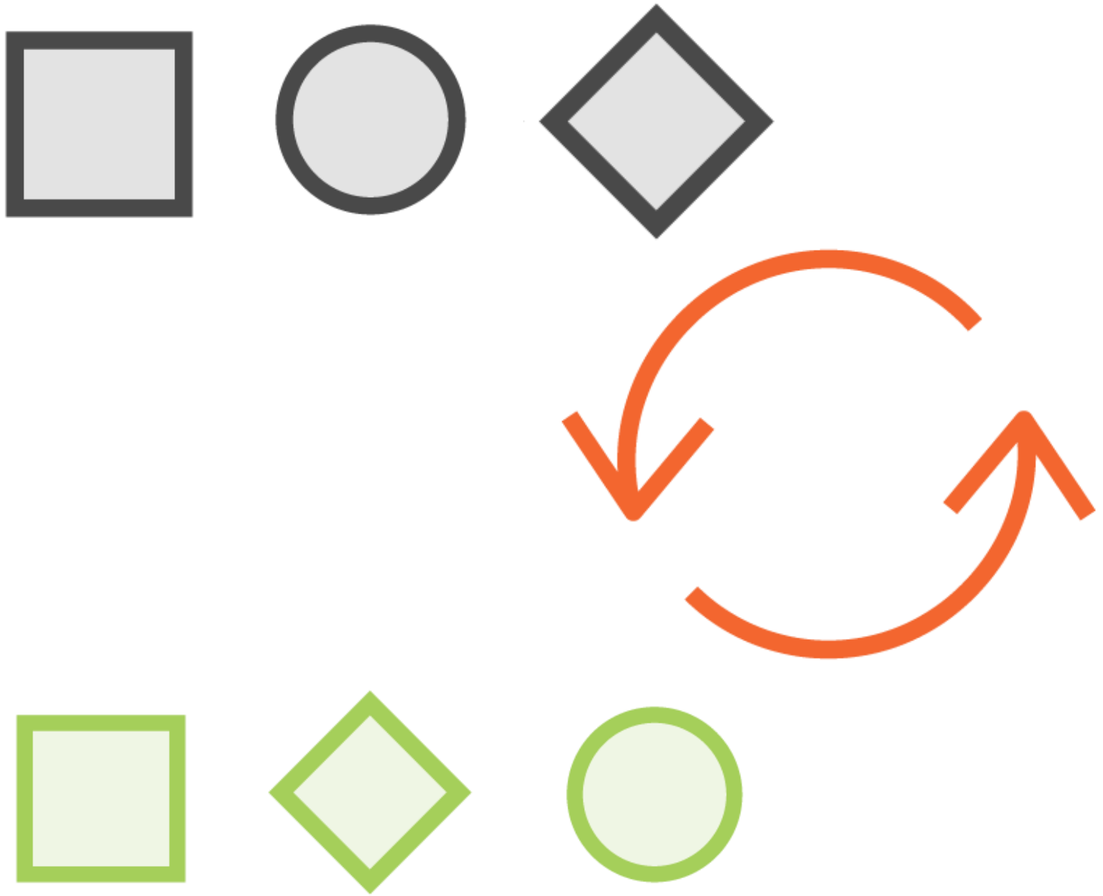
Unpivot operations convert data from wide form to long form data

Unions

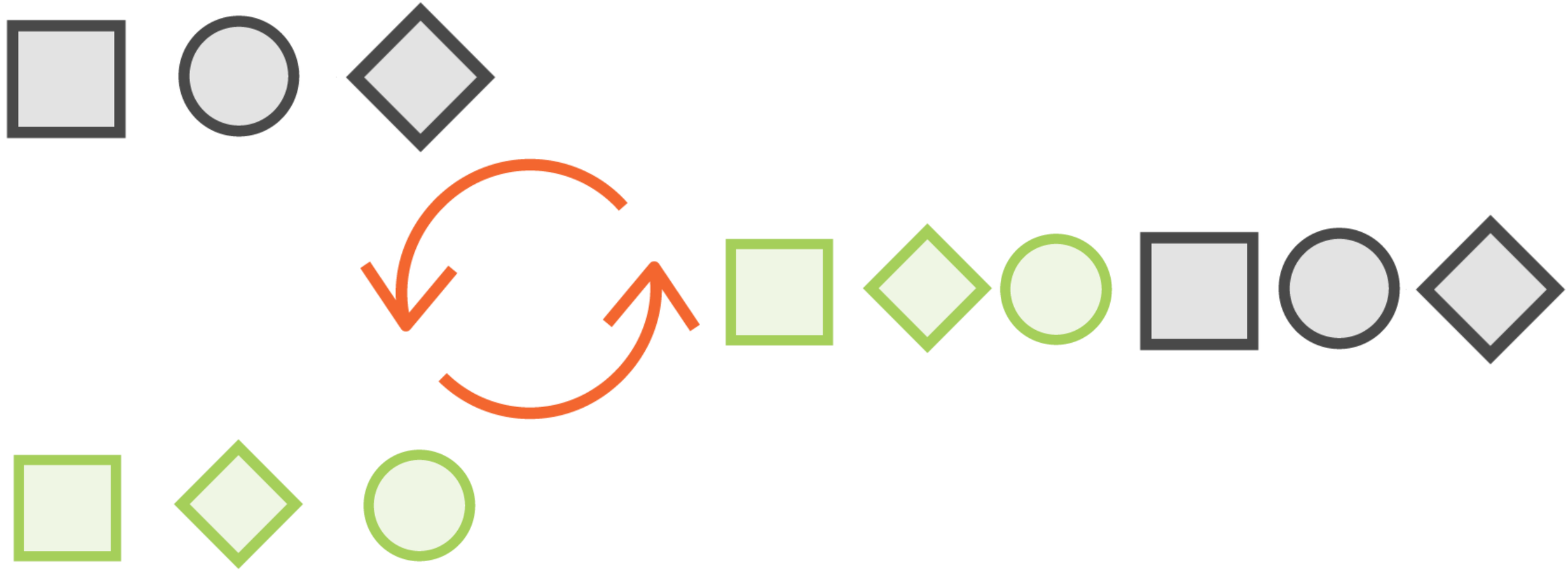
Connecting the Dots



Union



Union



Summary

Wide data formats

Long data formats

Types and uses of joins

Grouping operations

Aggregation operations

Combining and Shaping Data Using Python

Overview

Data cleaning and preparation in Python

Filling in missing values

Detecting and coping with outliers Working with imbalanced data

Data Cleaning

Data Cleaning and Preparation

Missing Data

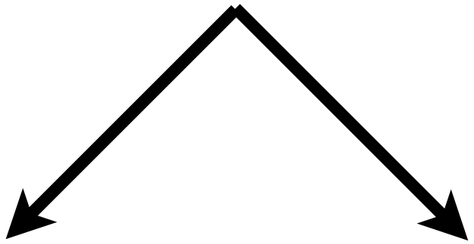
Outlier Data

Data Cleaning and Preparation

Missing Data

Outlier Data

Missing Data



Deletion

Imputation

Deletion a.k.a Listwise Deletion

Delete an entire record (row) if a single value (column) is missing. Simple but can lead to bias.

Listwise Deletion

Most common method in practice

Can reduce sample size significantly

If values are not missing at random, can introduce significant bias

Imputation

Fill in missing column values, rather than deleting records with missing values.

Imputation

Methods range from very simple to very complex

Simplest method: Use column average

Can interpolate from nearby values

Can even build model to predict missing values

Data Cleaning and Preparation

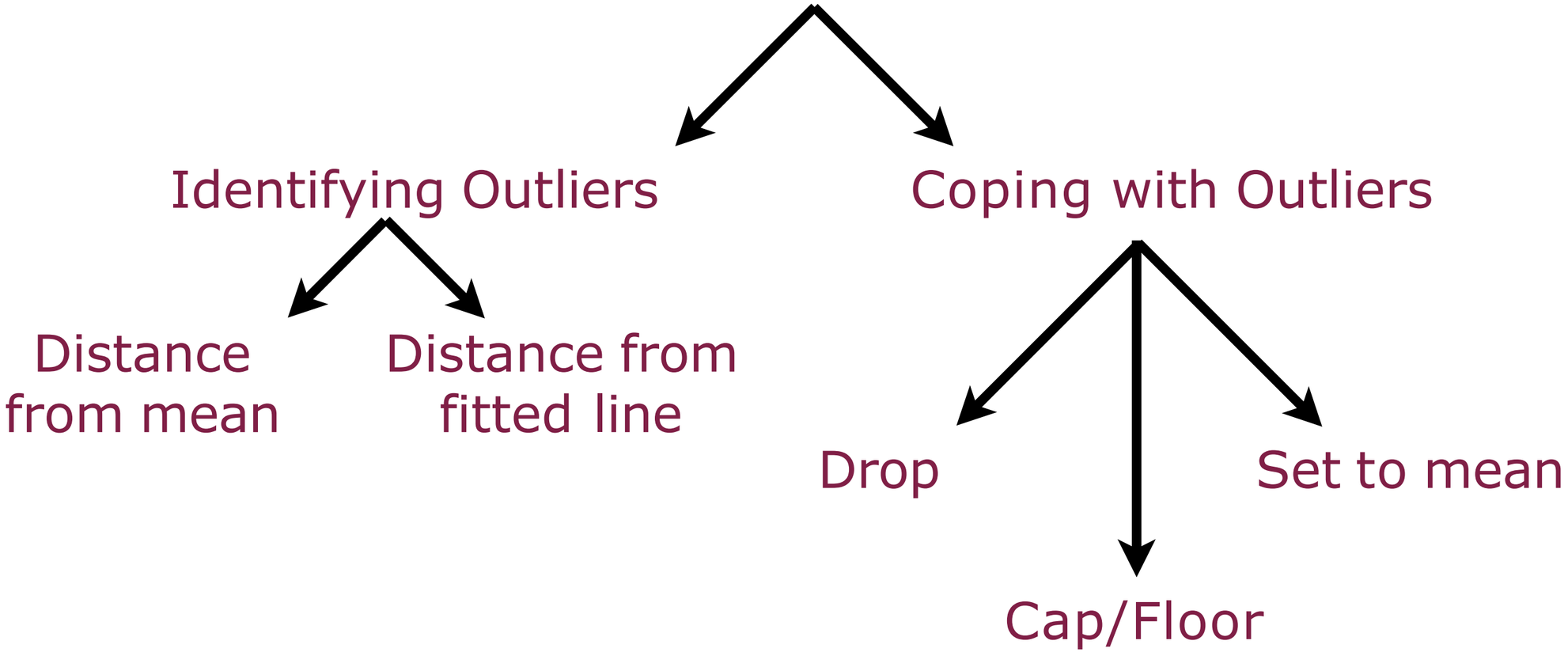
Missing Data

Outlier Data

Outlier

A data point that differs significantly from other data points in the same data set.

Outliers


Identifying Outliers
Coping with Outliers
Distance from mean
Distance from fitted line
Drop
Cap/Floor
Set to mean

Coping with Outliers

Always start by scrutinizing outliers

If erroneous observation

- Drop if all attributes of that point are erroneous
- Set to mean if only one attribute is erroneous

Coping with Outliers

If genuine, legitimate outlier

- Leave as-is if model not distorted
- Cap/Floor if model is distorted
  - Need to first standardize data
  - Cap positiver outliers to +3
  - Floor negative outliers to -3

Combining data and removing outliers

Detecting outliers using z-scores

Handling missing values in Python

Performing data cleaning and formatting operations

Oversampling and Undersampling

From Sample to Population

Population

All the data out there in the universe

Sample

A subset - hopefully representative - of the population

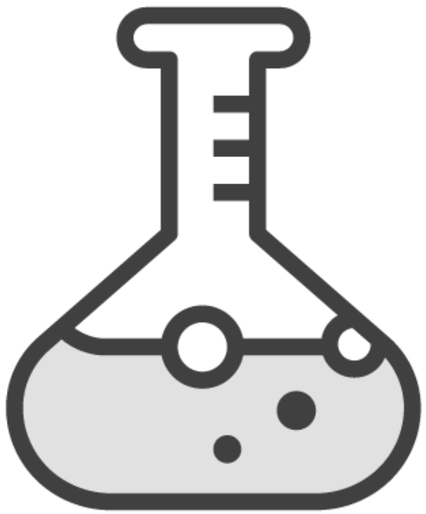
From Sample to Population

Population

Representative Sample

Biased Sample

When Unbiased Samples Make It Hard



A study seeks to measure health effect of a certain chemical

Exposure to chemical is random and extremely rare

For a meaningful test, an unbiased sample would need to be huge

Could we focus on the few exposed instances? (Case studies)

When Unbiased Samples Make It Hard

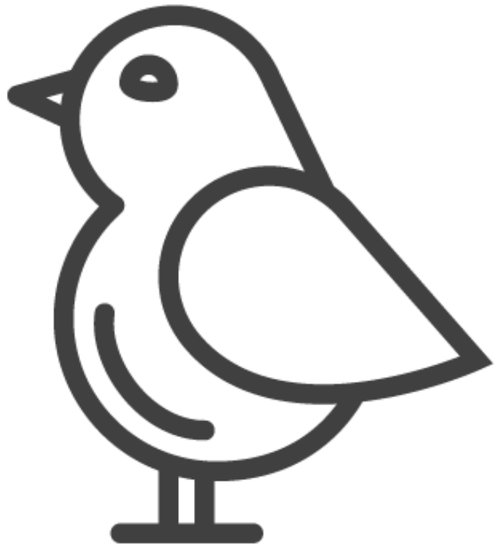
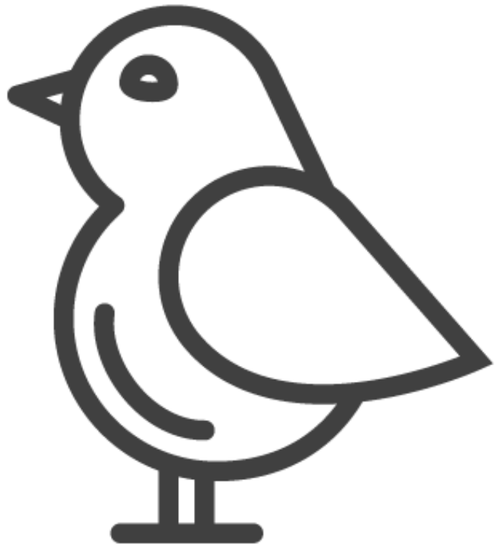


Image classifier looking for photos with Hawaiian Crow

One of the rarest birds on earth, looks a lot like the Common Crow

When Unbiased Samples Make It Hard



Training corpus has millions of images with the Common Crow

Only a dozen images with the Hawaiian Crow

Could we re-use images of the Hawaiian Crow?

Balancing Datasets

Oversampling of uncommon x or y values

Undersampling of common x or y values

Forcibly Balanced Datasets

Oversampling and undersampling tend to

- Reduce accuracy
- Increase precision and recall

Related techniques include

- Case studies
- Stratified sampling

Summary

Data cleaning and preparation in Python

Filling in missing values

Detecting and coping with outliers Working with imbalanced data

Integrating Data from Disparate Sources into a Data Warehouse

Overview

Transactional vs. analytical processing

Data warehouses vs. relational databases

Integrating data from different sources into an Azure SQL Data Warehouse

Transactions vs. Analytical Processing

Transactional and Analytical Processing

Order Management Support

John is responsible for tracking and delivering orders on time

Revenue Analyst

Anna is responsible for tracking and monitoring revenues

Order Management Support

3 customers want to ship orders to a different address

John updates the address on the shipments and re-routes them

Revenue Analyst

Management asks if the new TV ads this year were successful

Anna checks customer signups for a jump when the campaigns were run

Transactional and Analytical Processing

Transactional Processing

Analytical Processing

Transactional and Analytical Processing

Transactional Processing

Ensure correctness of individual entries

Access to recent data, from the last few hours or days

Updates data

Fast real-time access

Usually a single data source

Analytical Processing

Analyzes large batches of data

Access to older data going back months, or even years

Mostly reads data

Long running jobs

Multiple data sources

Transactional and Analytical Processing

Small Data

Both these objectives could be achieved
using the same database system

Small Data

Single machine with backup

Structured, well-defined data

Can access individual records or the entire dataset

No replication, updated data available instantaneously

Different tables store data from different sources

Transactional and Analytical Processing

Big Data

Very hard to meet all requirements
with the same database system

Big Data

Data distributed on a cluster with multiple machines

Semi-structured or unstructured data

No random access to data

Data replicated, propagation of updates take time

Different sources may have different unknown formats

3 Vs of Big Data

Volume: Amount of data

Variety: Number and type of sources

Velocity: Batch and streaming

Transactional and Analytical Processing

Transactional Processing

Traditional
RDBMS

Analytical Processing

Data Warehouse

Data Warehouse

Structured data store used for analytical processing and reporting; usually hold transformed data fed in from disparate sources via ETL Pipelines.

ETL Pipelines

Programs or scripts with business logic to automatedly *e*xtract data from disparate sources, *t*ransform it to satisfy a schema, then *l*oad it into a data warehouse.

Summary

Transactional vs. analytical processing

Data warehouses vs. relational databases

Working with Streaming Data Using a Data Warehouse

Overview

Batch and streaming data

Window operations on streaming data

event time, ingestion time and processing time

Ingest streaming data with Stream Analytics

Visualize data using Power BI

Batch and Stream Processing

Bounded datasets are processed in **batches**

Unbounded datasets are processed as **streams**

Batch vs. Stream Processing

Batch	Stream
Bounded, finite datasets	Unbounded, infinite datasets
Slow pipeline from data ingestion to analysis	Processing immediate, as data is received
Periodic updates as jobs complete	Continuous updates as jobs run constantly

Batch vs. Stream Processing

	Batch	Stream
Order of data received	unimportant	Order important, out of order arrival tracked
Single global state of the world	at any point in time	No global state, only history of events received

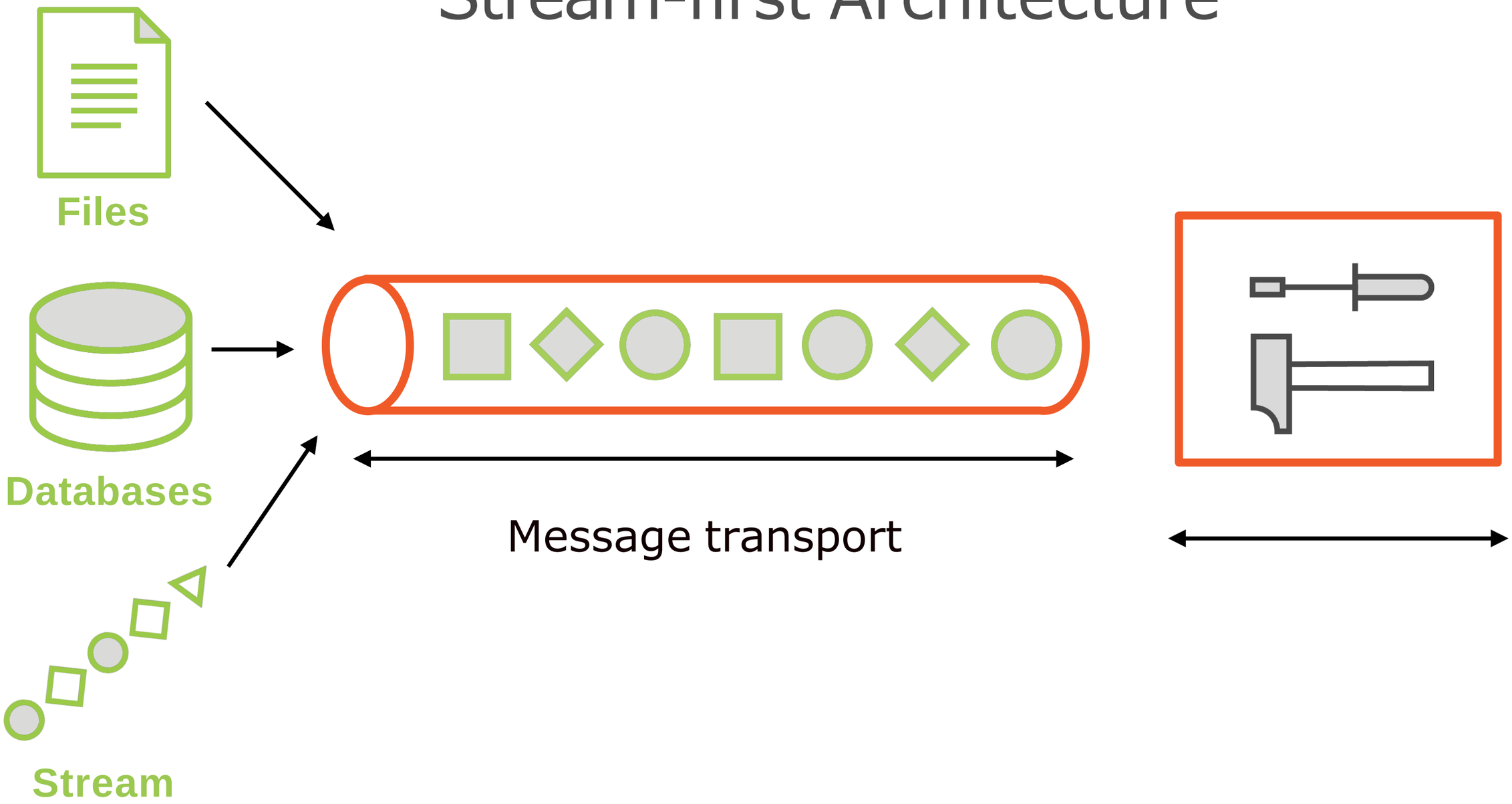
Traditional Systems

Files

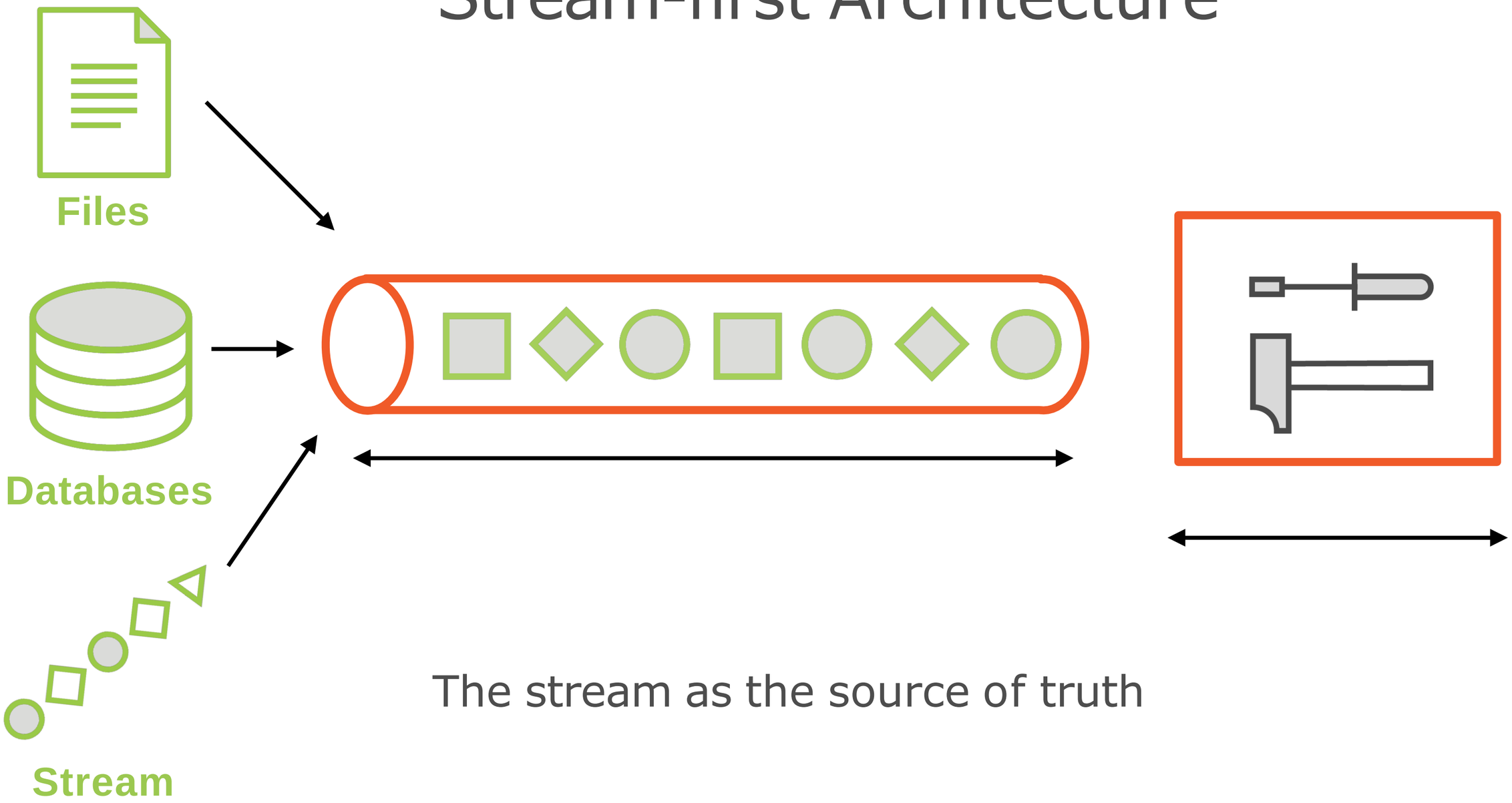
Databases

Reliable storage as the source of truth

Stream-first Architecture



Stream-first Architecture



The stream as the source of truth

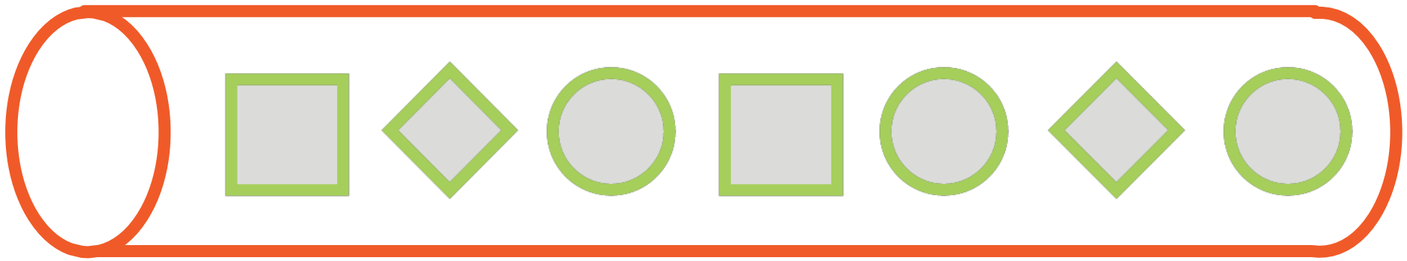
Message Transport

Buffer for event data

Performant and persistent

Decoupling multiple sources from processing

Kafka, MapR streams



Stream Processing

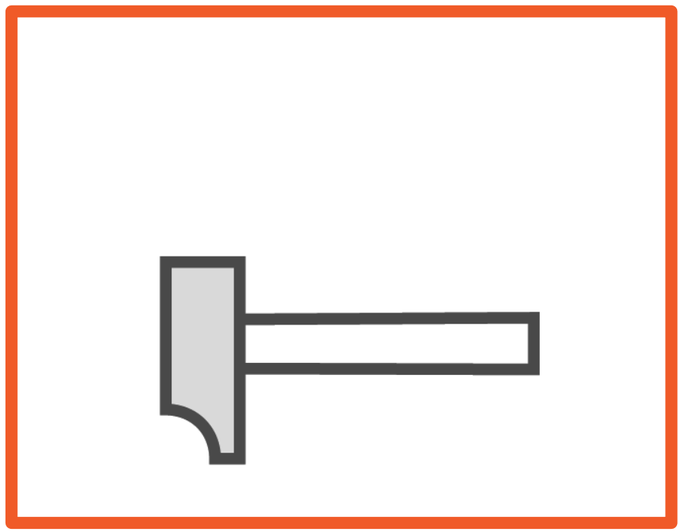
High throughput, low latency

Fault tolerance with low overhead

Manage out of order events

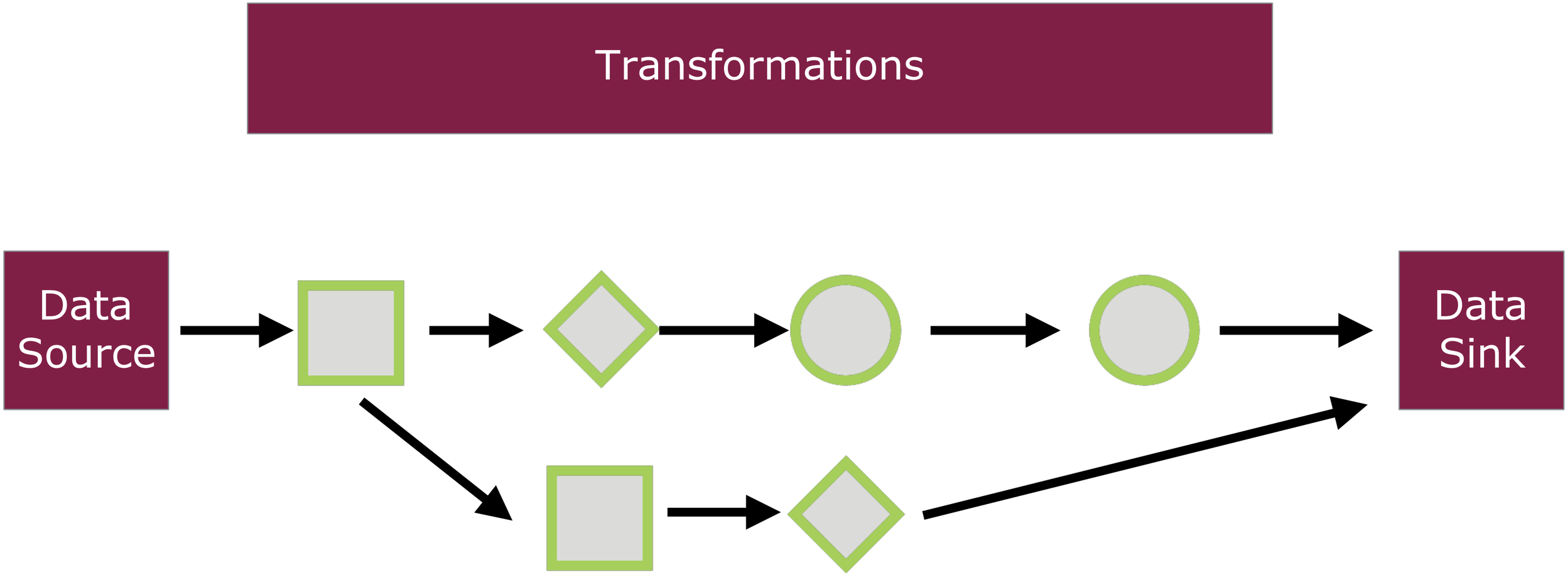
Easy to use, maintainable

Replay streams



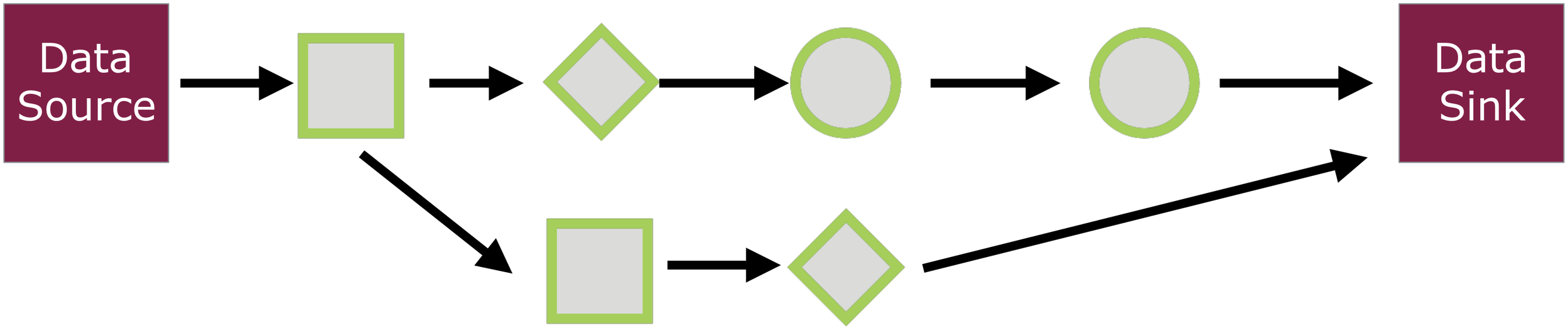
Stream Processing Model

Stream Processing Model



Transformations

A directed-acyclic graph



Window Operations on Streaming Data

Transformations

Stateless

Transformations which are applied on a single stream entity

Stateful

Transformations which accumulate across multiple stream entities

Window Transformations

Accumulate information across a window in a stream

Stateless Transformations

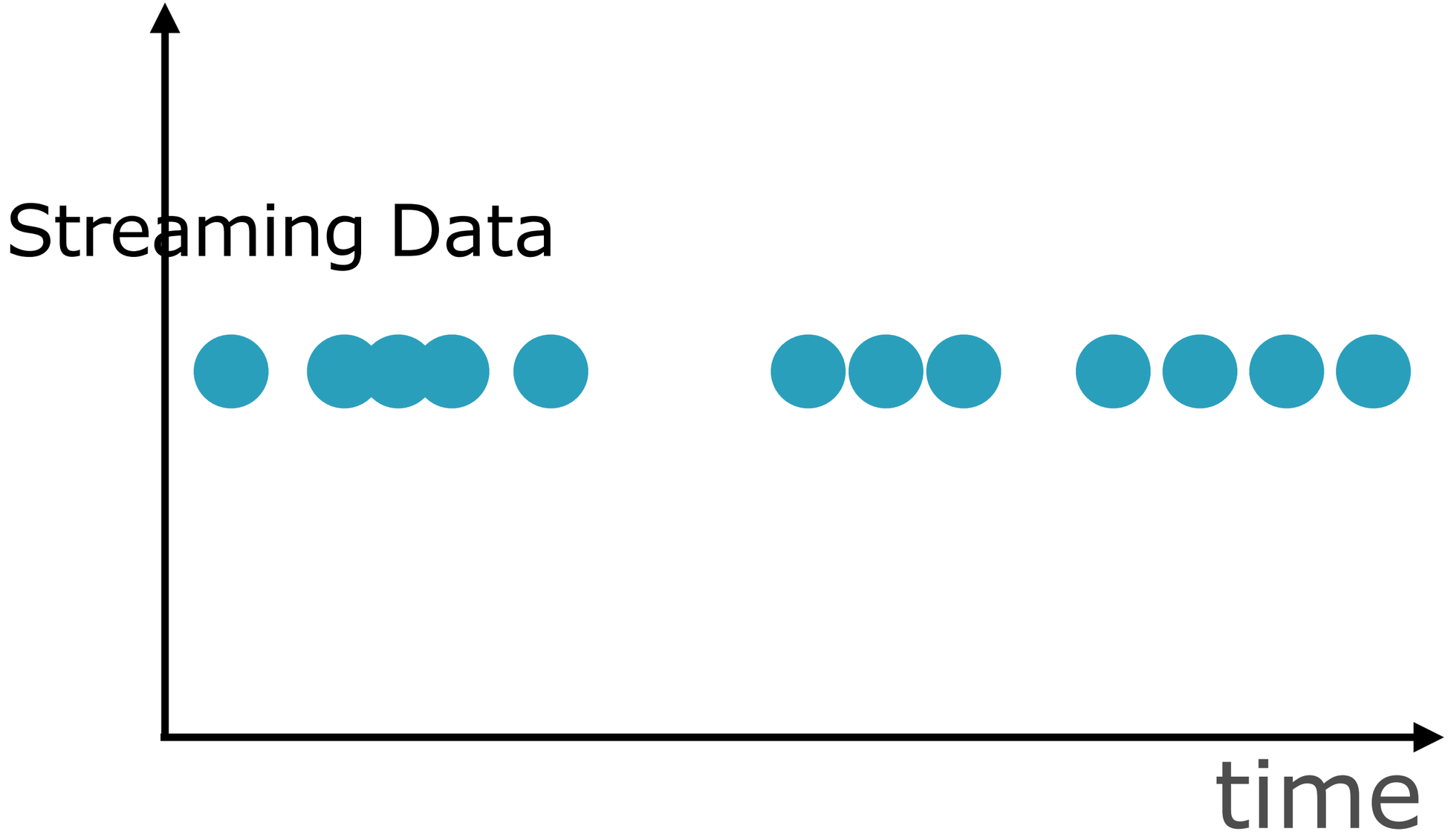
Each entity is operated on standalone

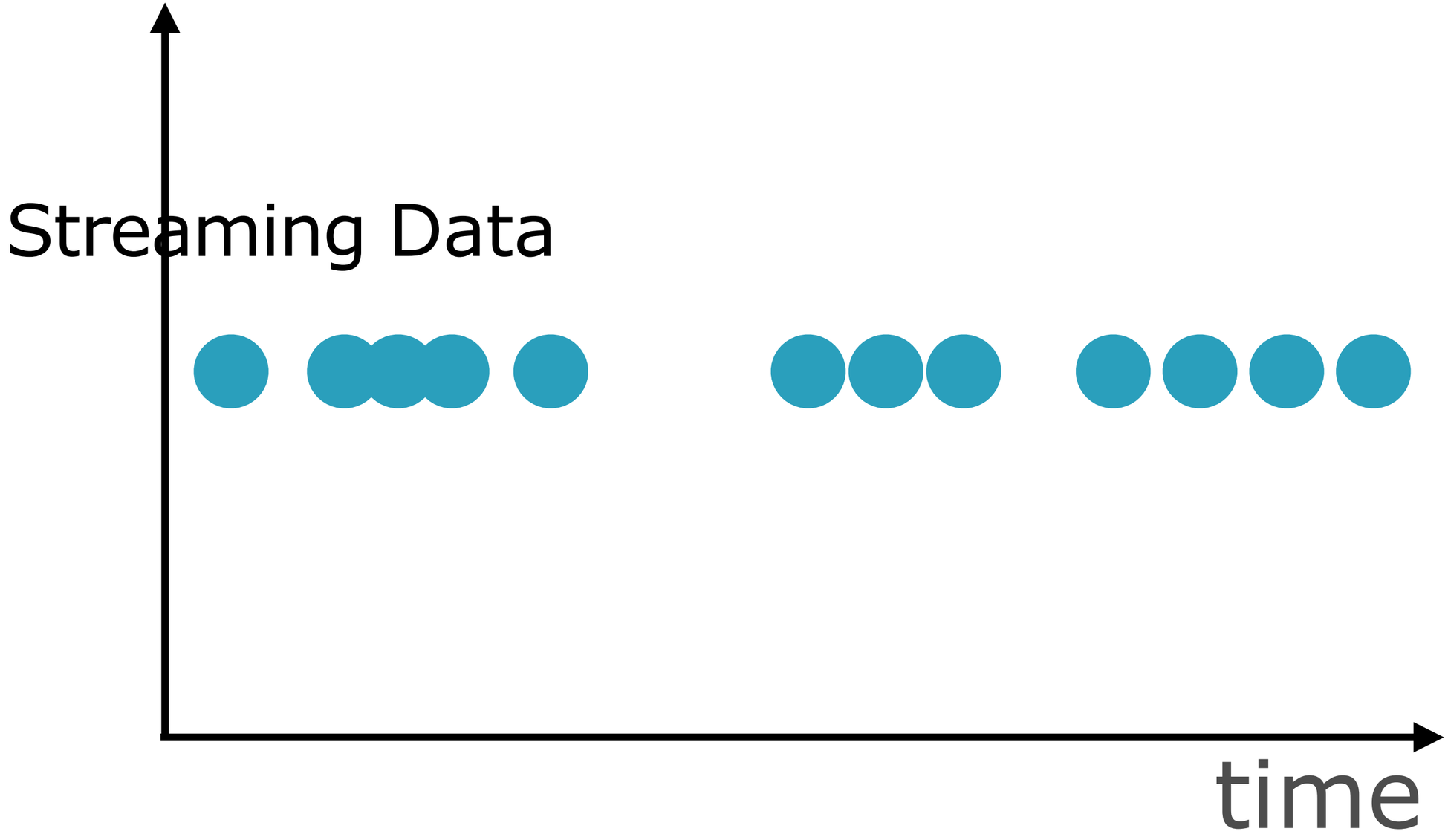
Speed exceeded? Alert triggered

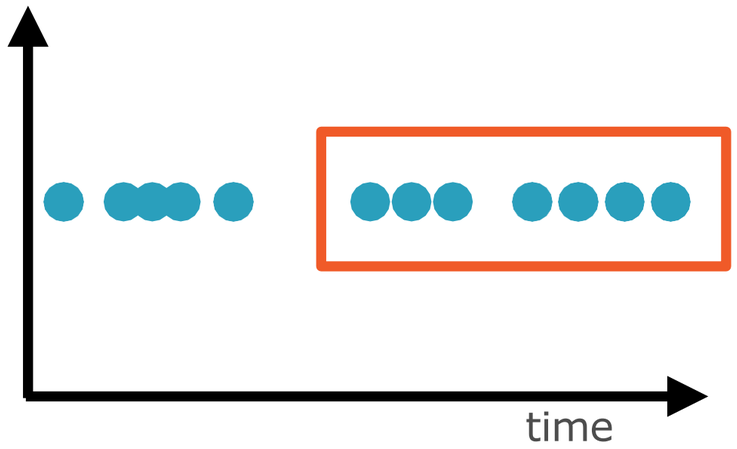
1

2

4

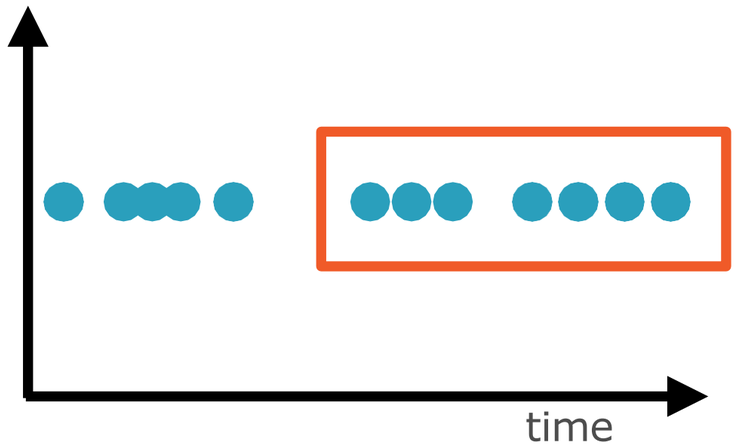

Streaming Data
time


Streaming Data
time



A window is a subset of a stream based on

- Time interval
- Count of entities
- Interval between entities



Transformations can be applied on all entities within a window

- sum, min, max, average

Types of Windows

Tumbling Window

Sliding Window

Count Window

Session Window

Global Window

Types of Windows

Tumbling Window

Sliding Window

Count Window

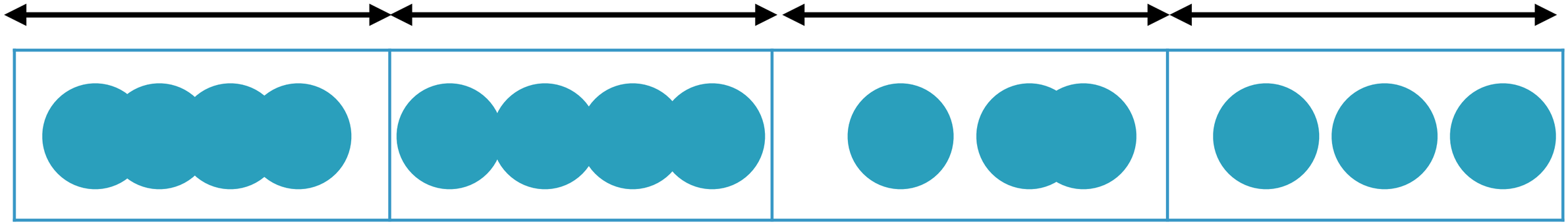
Session Window

Global Window

Types of Windows

A stream of data

Tumbling Window



Fixed window size

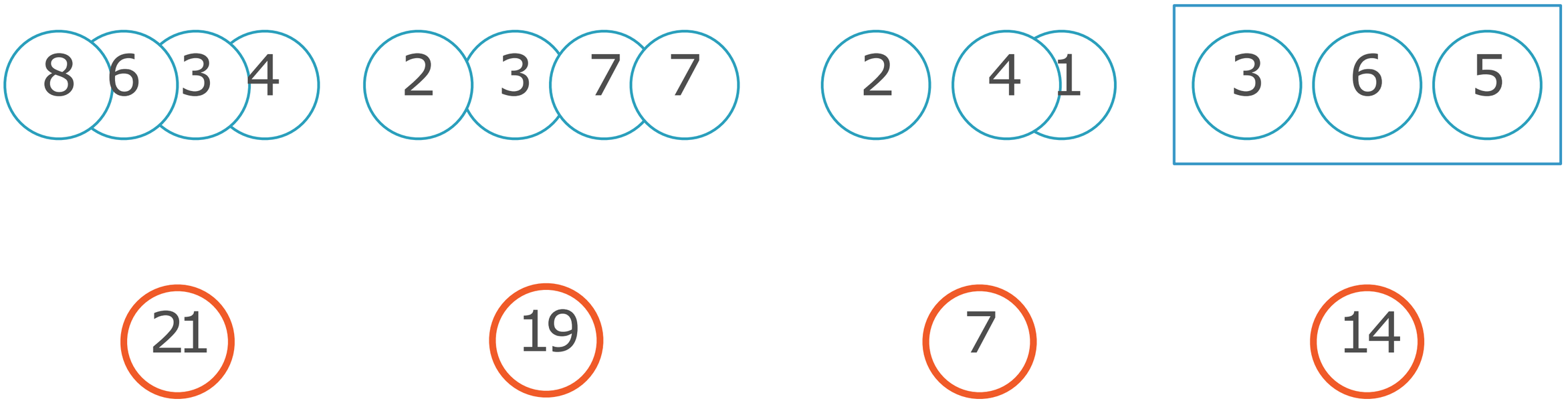
Non-overlapping time

Number of entities differ within a window

Tumbling Window

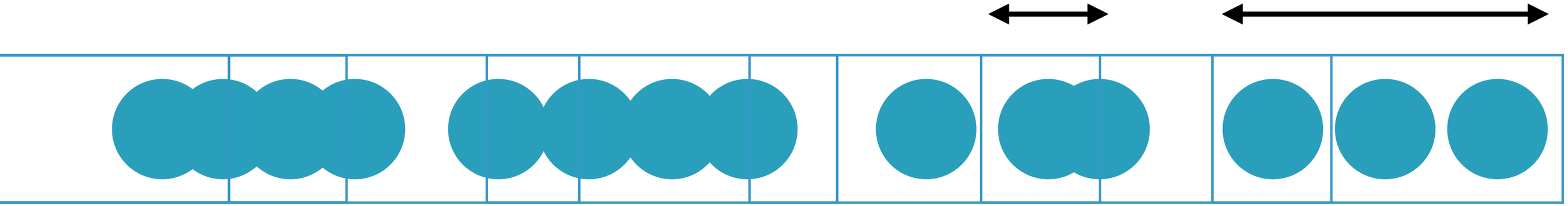
The window tumbles over the data, in a non-overlapping manner

Tumbling Window



Apply the sum() operation on each window

Sliding Window

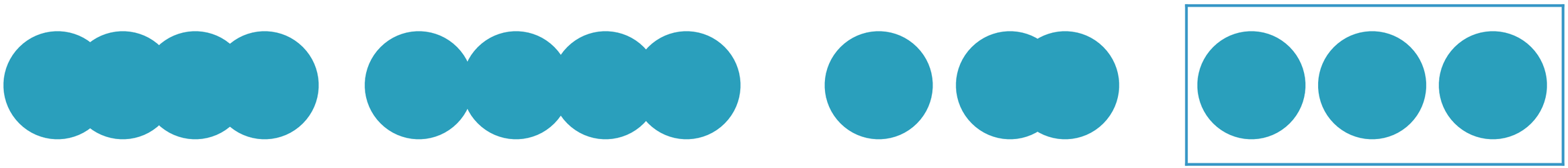


Fixed window size

Overlapping time - sliding interval

Number of entities differ within a window

Sliding Window

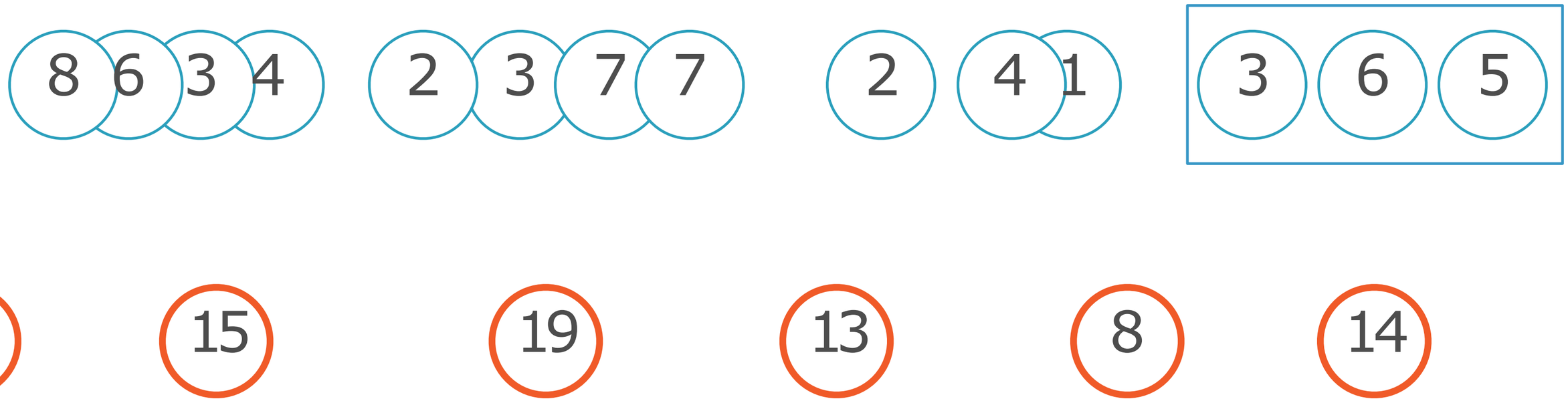


Fixed window size

Overlapping time - sliding interval

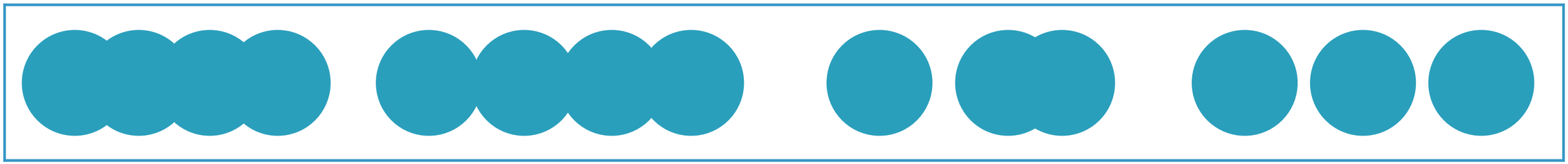
Number of entities differ within a window

Sliding Window



Apply the sum() operation on each window

Global Window



All data in the stream in one window

Types of Windows

Tumbling Window

Sliding Window

Count Window

Session Window

Global Window

Types of Windows

Tumbling Window

Sliding Window

Count Window

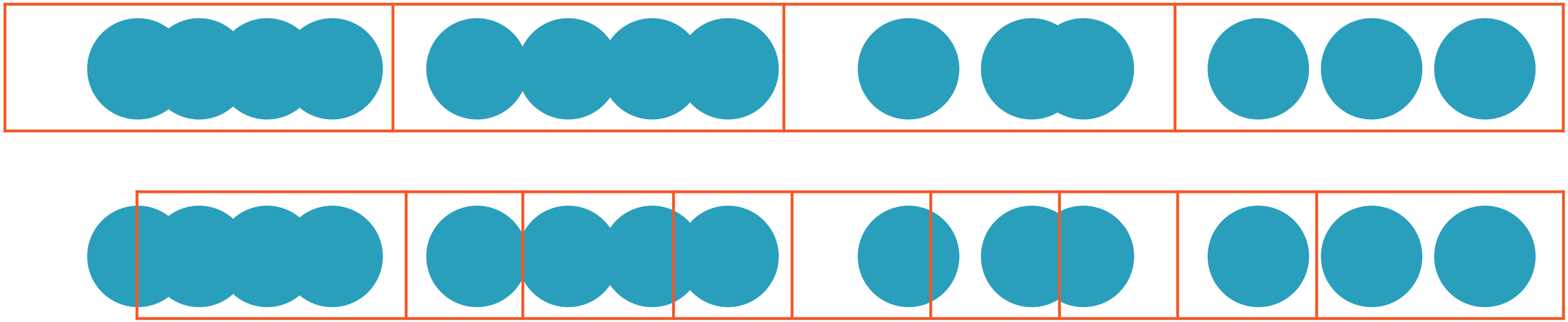
Session Window

Global Window

Event Time and Processing Time

Time-based Windows

Tumbling and sliding windows consider entities in a fixed interval of time

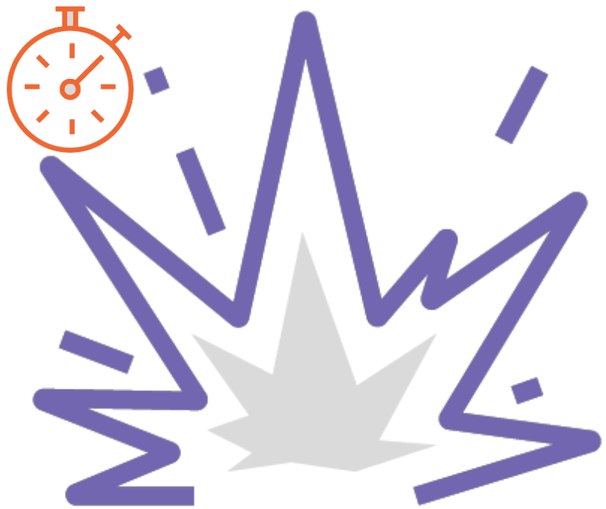


Time-based Windows

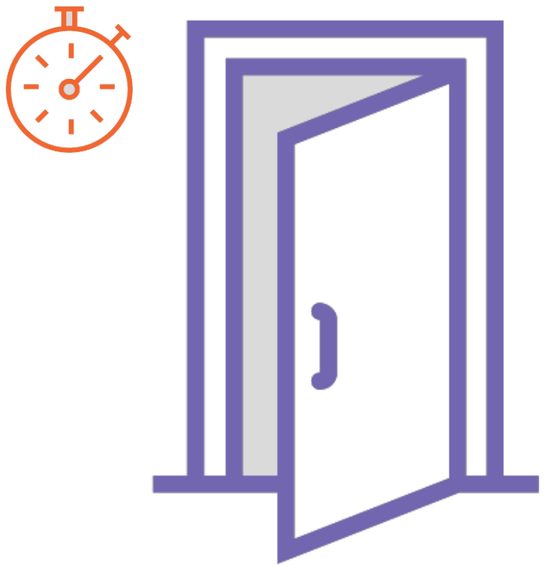
Tumbling and sliding windows consider entities in a fixed interval of time

There are different notions of time that can apply to entities in a stream

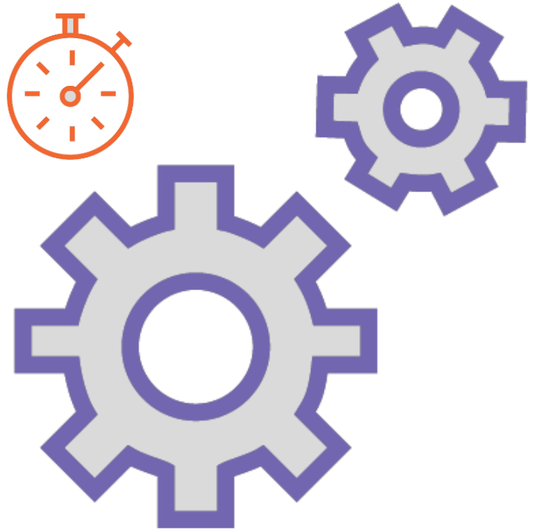
Time



Event Time



Ingestion Time



Processing Time

Event Time

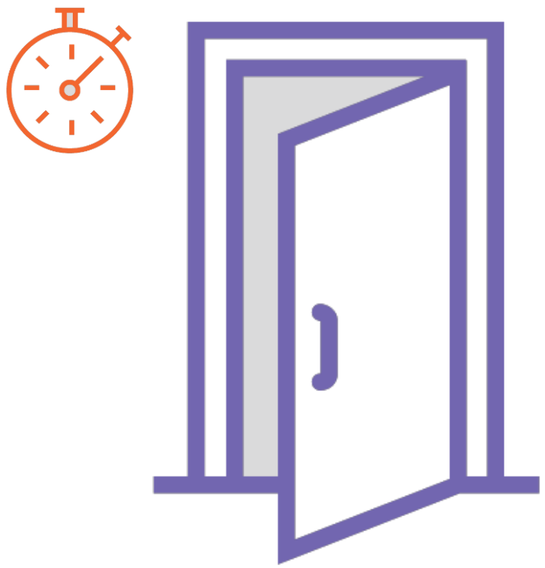
The time at which the event occurred at its original source

- Mobile phone, sensor, website

Usually embedded within records

Gives correct results in case of out of order or late events

Ingestion Time

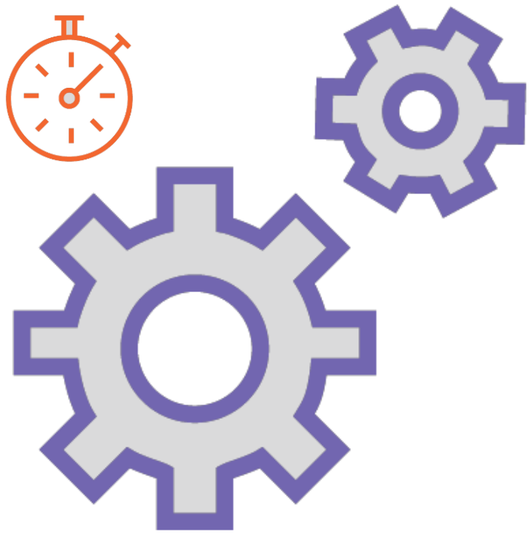


The time at which the event enters the system via a source

Timestamp given by system chronologically after the event time

Cannot handle out of order events

Processing Time



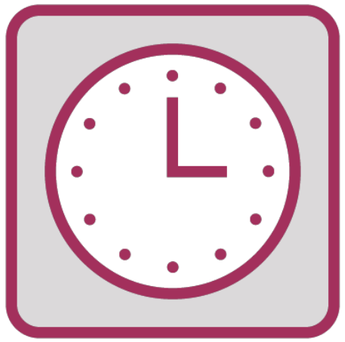
The system time of the machine processing entities

Chronologically after event time and ingestion time

Non-deterministic, depends on when data arrives, how long operations take

Simple, no coordination between streams and processors

How Late Is Late?



Class At 9 am

Class starts when clock strikes 9

Is 9:01 Late?

Realistically, at least some folks are going to be a minute late

Is 10:10 late?

A student is an hour late - allow in or send back?

The professor “knows” what lateness is reasonable

Students entering within this reasonable lateness are late but OK

Students entering after this reasonable lateness are too late

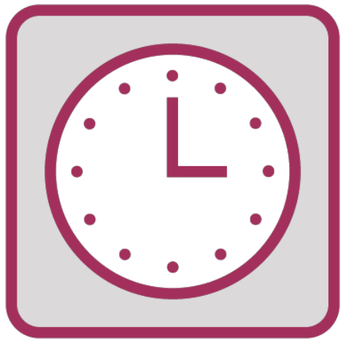
“Allowed Lateness”

Dealing with excessive lateness

A student is too late

- Option 1: Send back home
- Option 2: Allow in, continue class
- Option 3: Allow in, restart class(!)

How Late Is Late?



Trigger

Class starts when clock strikes 9

Allowed Lateness

Realistically, at least some folks are going to be a minute late

Unacceptable Lateness

A student is an hour late - allow in or send back?

Watermarks and Late Data

The system “knows” what lateness is reasonable

Data entering within this reasonable lateness is late but OK

Data entering after this reasonable lateness is too late

Watermarks and Late Data

Watermark

Threshold of allowed lateness (event time)

Late Data

Data within watermark is aggregated

Dropped Data

Data outside watermark is dropped

Summary

Batch and streaming data

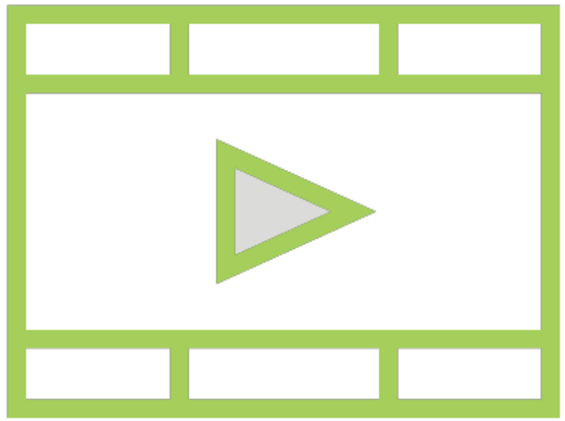
Window operations on streaming data

event time, ingestion time and processing time

Ingest streaming data with Stream Analytics

Visualize data using Power BI

Related Courses



Representing, Processing and Preparing Data

Summarizing Data and Deducing Probabilities