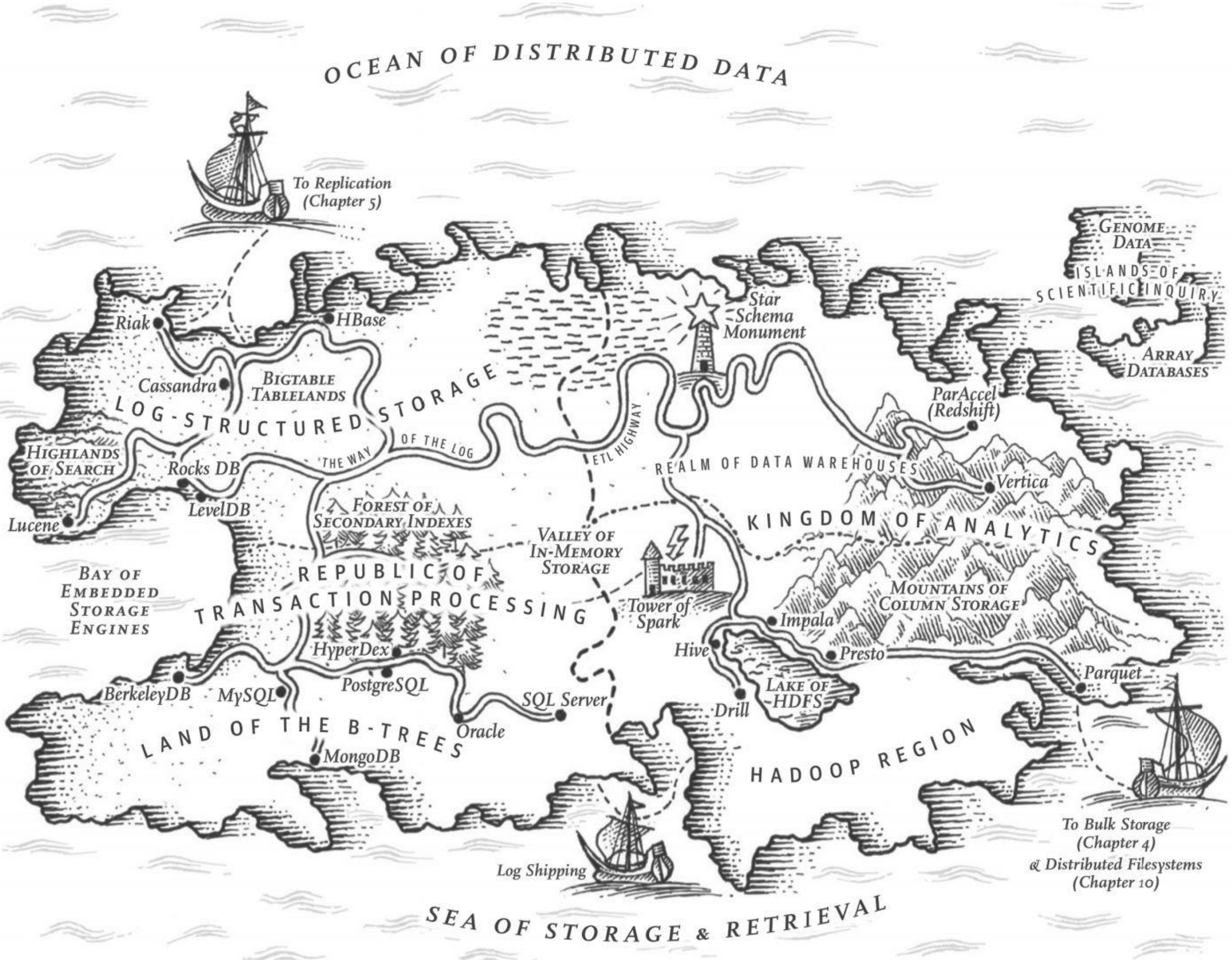
A gentle introduction to

Apache Spark, Databricks and Delta Lake

Personal Introduction

- Senior Consultant at Avanade
- MSc in Financial & Management Engineering
- Microsoft Certified Professional
- 12 years long journey, in the vast continent of Data

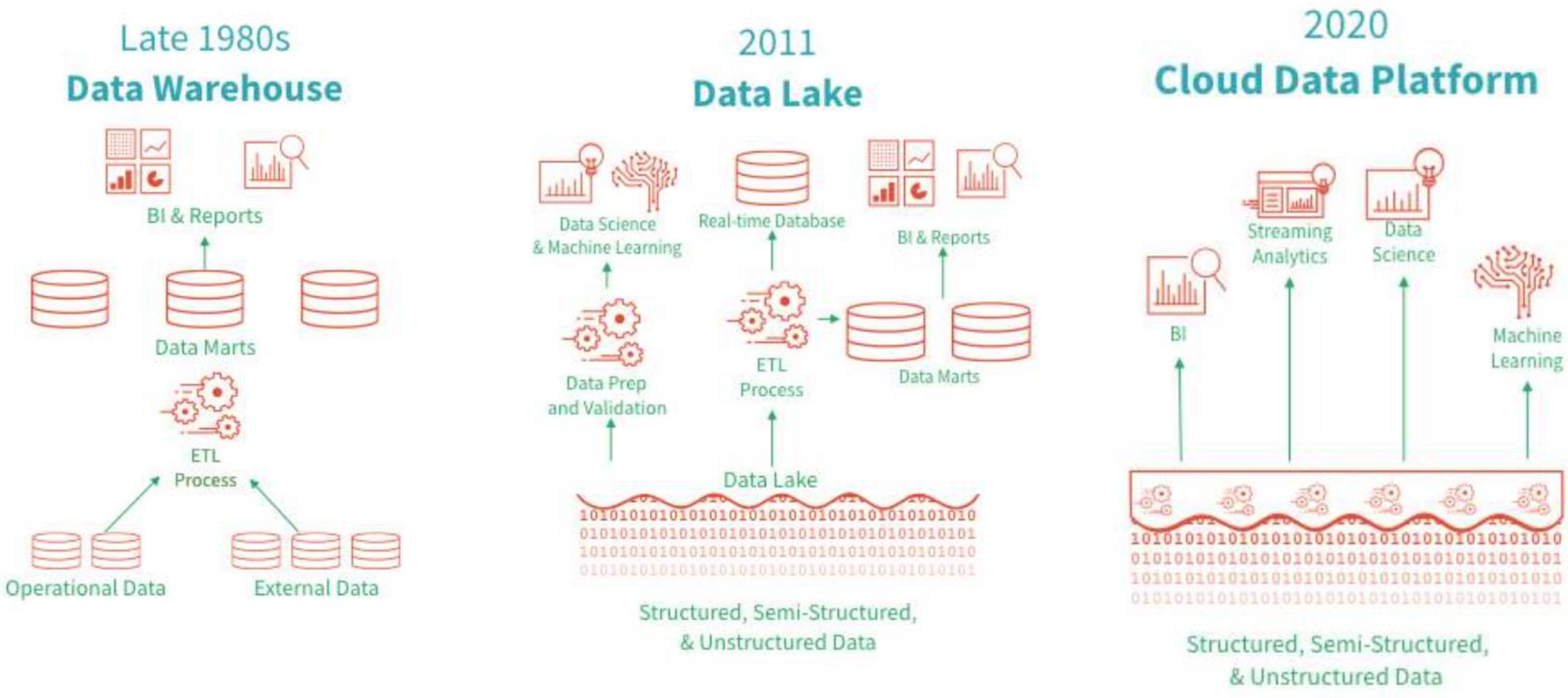


**Designing Data-Intensive Applications by Martin Kleppmann*

Agenda

- The big picture
- A peak into Apache Spark using Databricks
- Delta Lake
- Where do I go from here?

The evolution of Data Management

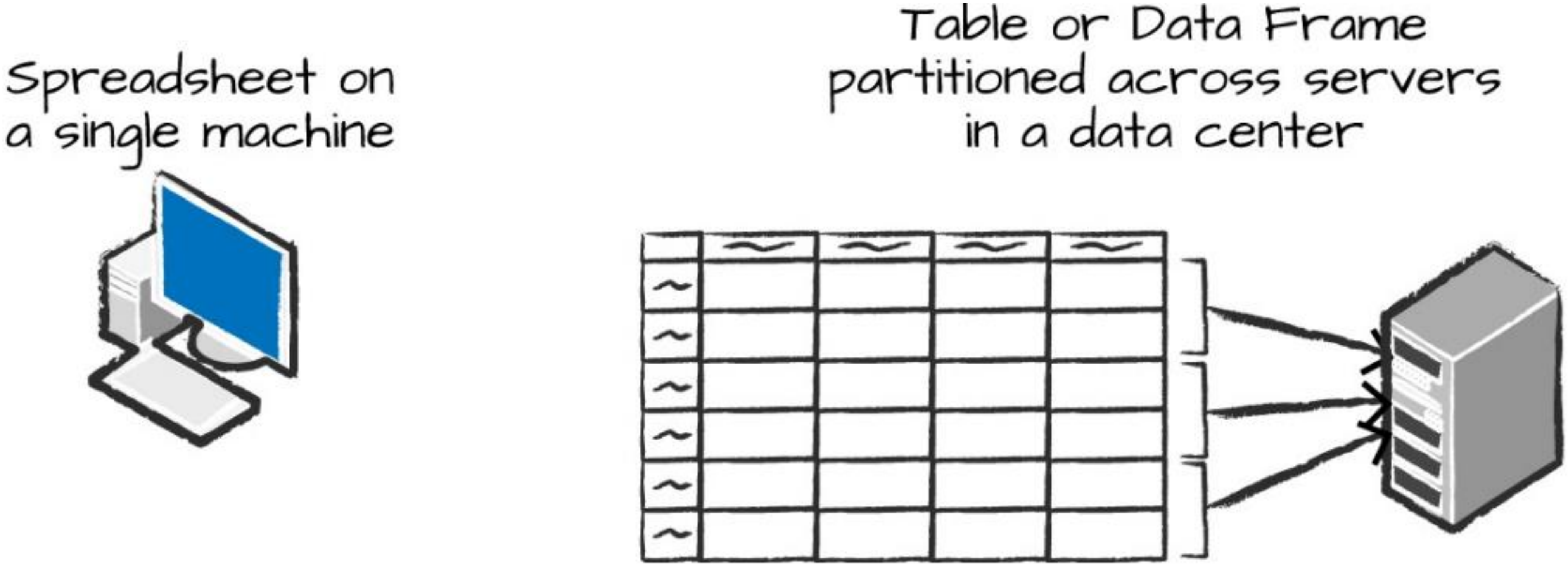


A brief history of Apache Spark

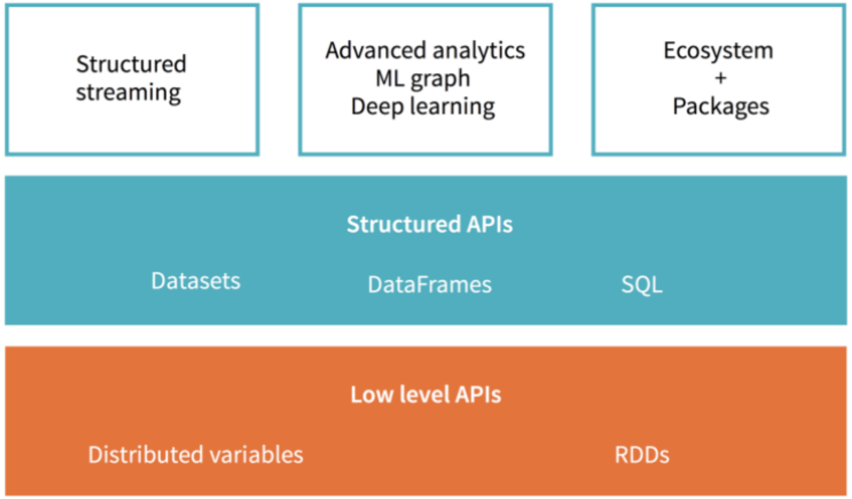
- Began in 2009 at UC Berkeley as a research project
- At that time Hadoop was the dominant paradigm
- Databricks was founded in 2013
- Spark 1.0 released in 2014
- Spark 2.0 released in 2016
- Spark 3.0 released in 2020

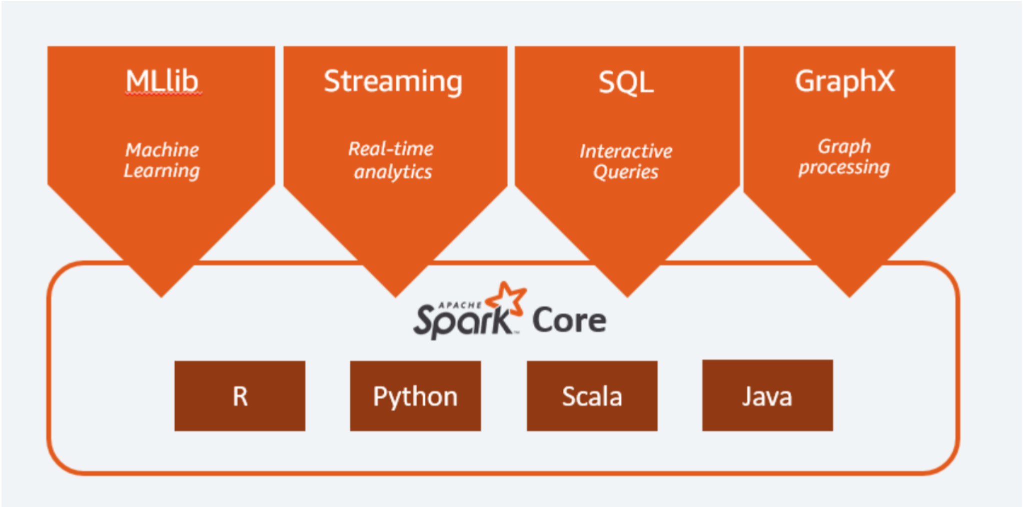
What is Spark and how does it work?

A simple example

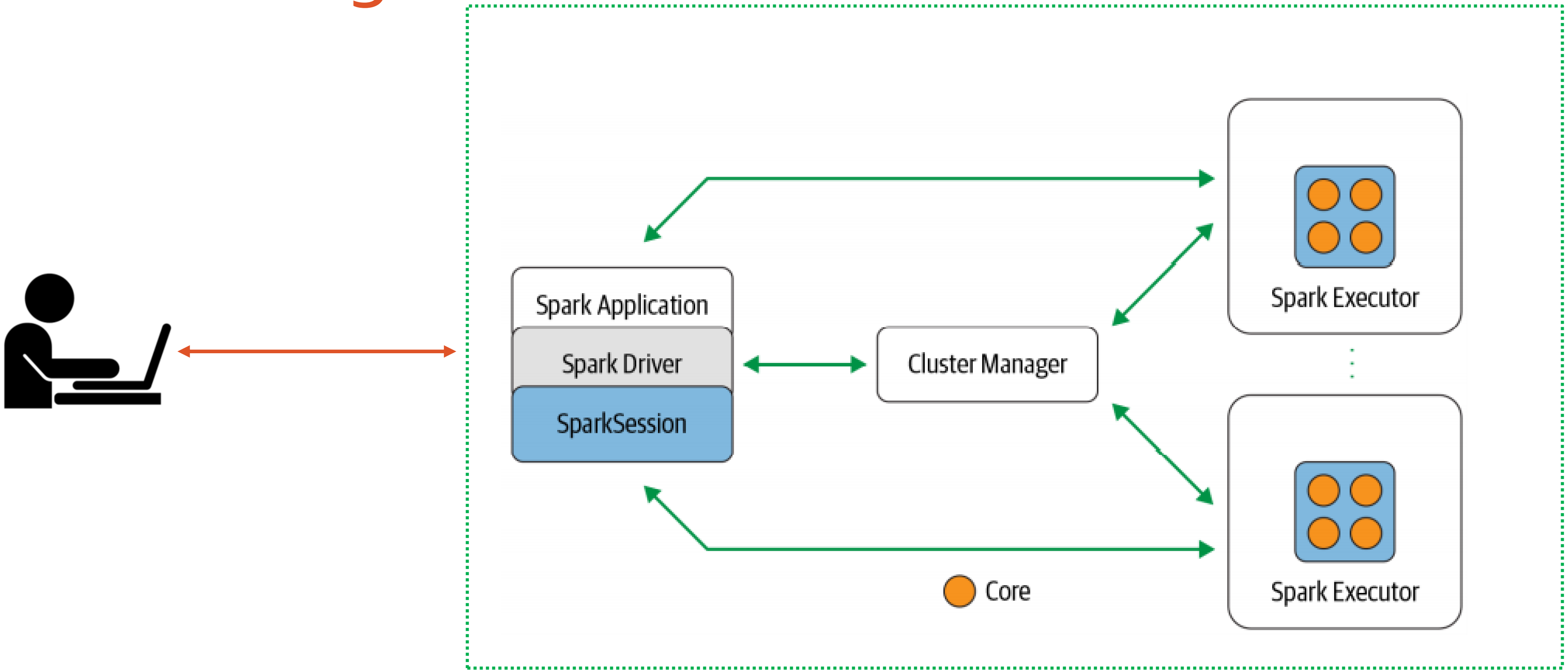

Spreadsheet on a single machine
Table or Data Frame partitioned across servers in a data center

How do we talk to Spark?

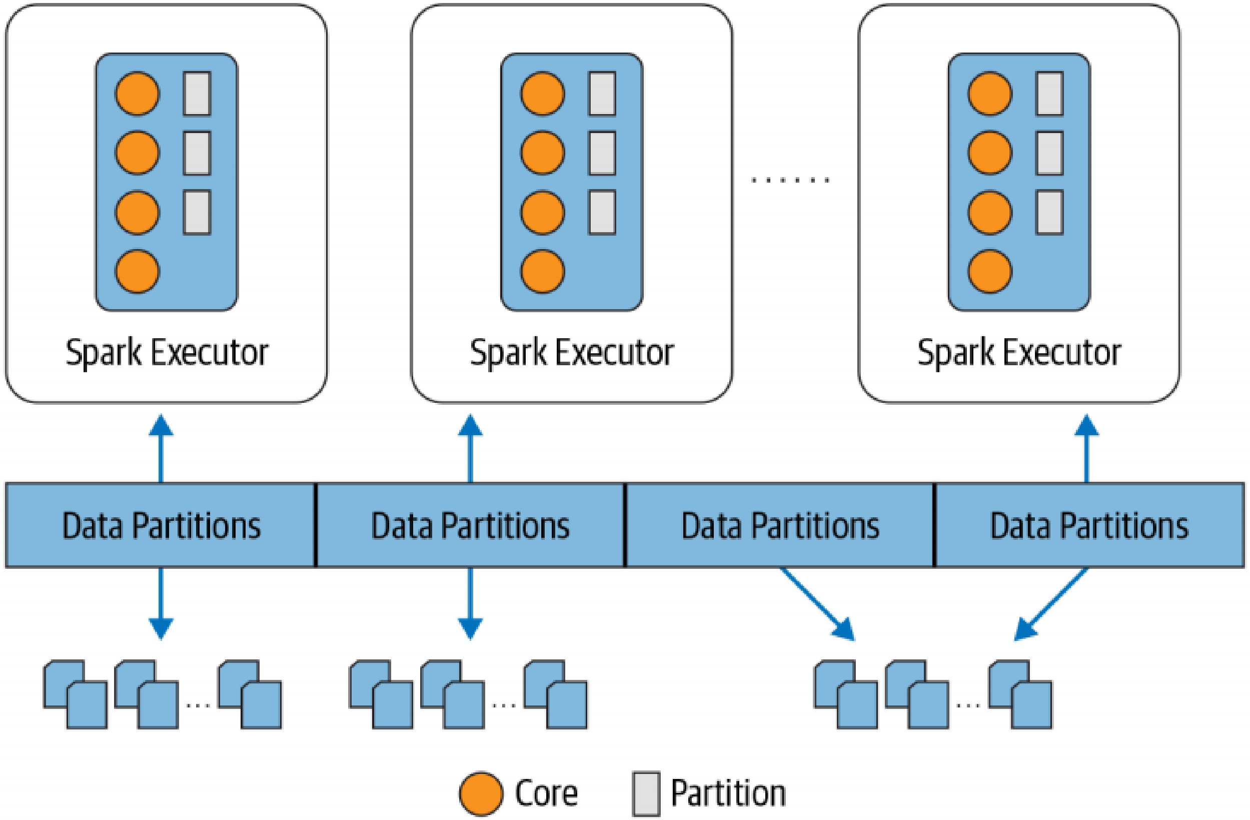




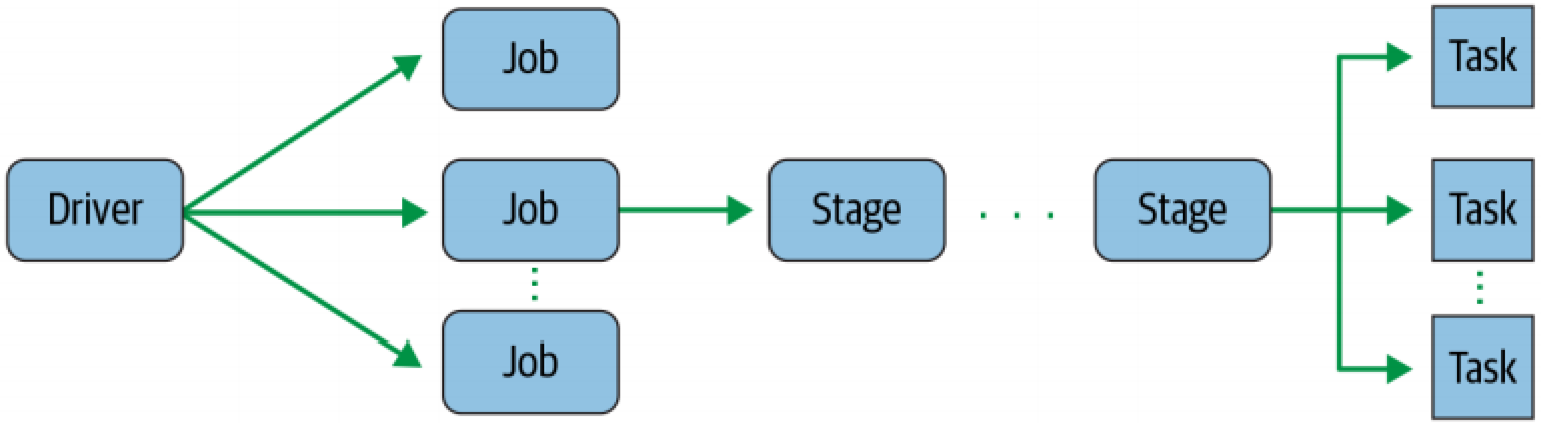
Distributing work



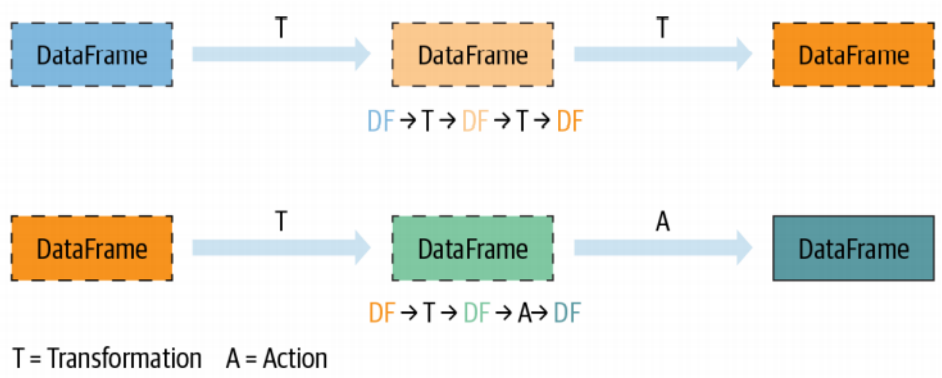
Distributing work

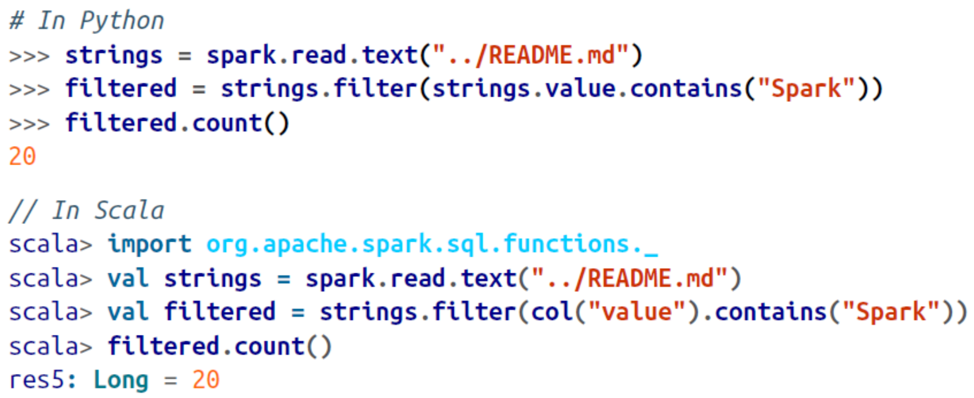


Spark stages, jobs and tasks



Transformations, Actions and Lazy Evaluation

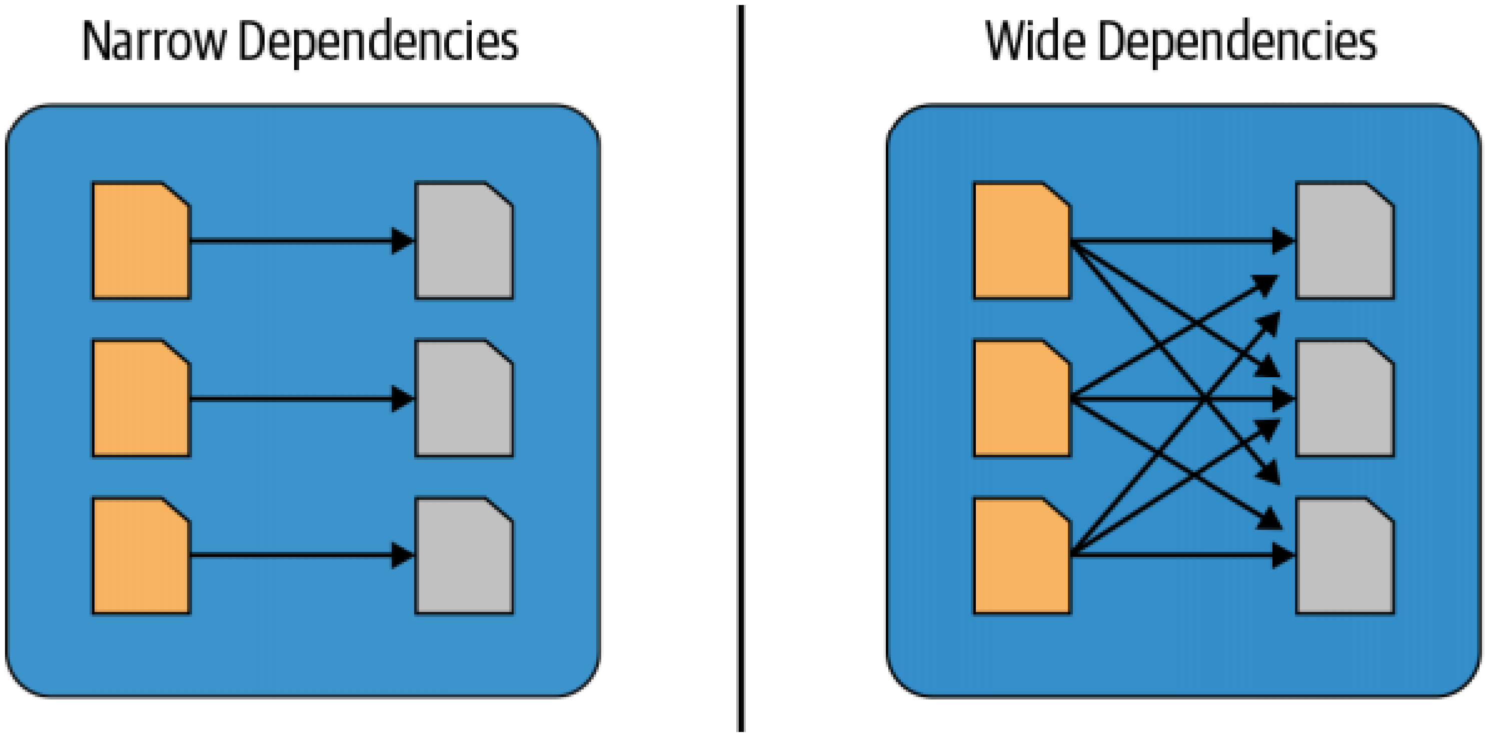




```
# In Python
>>> strings = spark.read.text("../README.md")
>>> filtered = strings.filter(strings.value.contains("Spark"))
>>> filtered.count()
20

// In Scala
scala> import org.apache.spark.sql.functions._
scala> val strings = spark.read.text("../README.md")
scala> val filtered = strings.filter(col("value").contains("Spark"))
scala> filtered.count()
res5: Long = 20
```

Transformations, Actions and Lazy Evaluation



An example on narrow transformations:

"Select data where age = 37"

Node 1

Name	Age	City
Arnold	37	Amsterdam
Mohamed	37	London
John	25	Athens

Node 2

Name	Age	City
Lara	37	New York
George	31	London
Seif	45	Cairo

Node 3

Name	Age	City
Ankur	37	Mumbai
Jack	67	London
Lian	24	Beijing

Node 1

Name	Age	City
Arnold	37	Amsterdam
Mohamed	37	London

Node 2

Name	Age	City
Lara	37	New York

Node 3

Name	Age	City
Ankur	37	Mumbai

Final Result

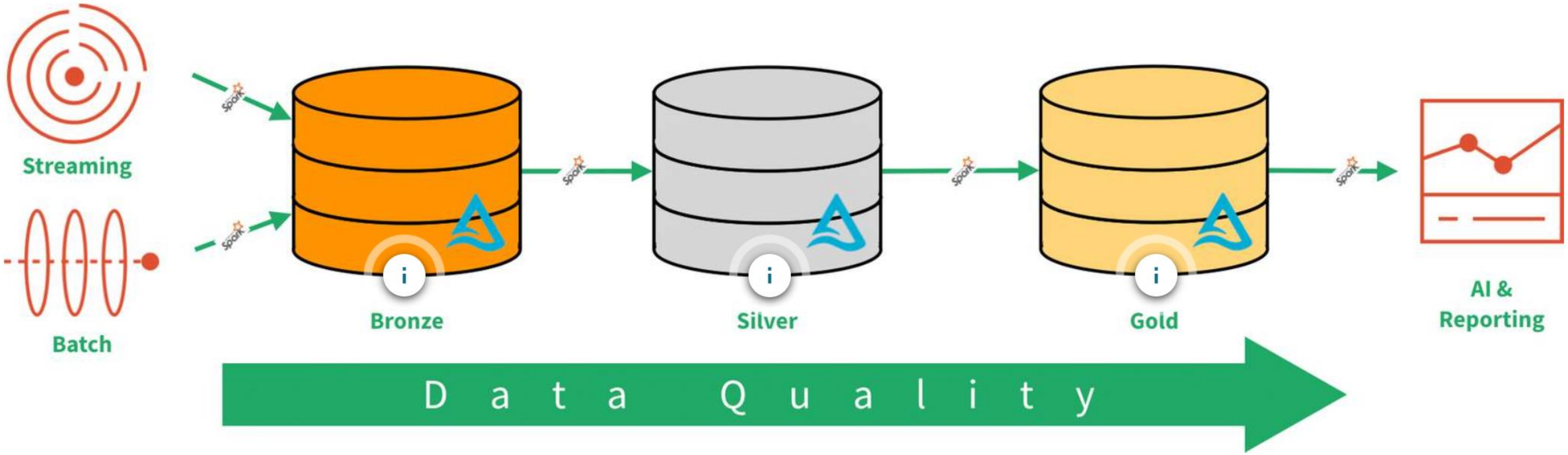
Name	Age	City
Arnold	37	Amsterdam
Mohamed	37	London
Lara	37	New York
Ankur	37	Mumbai

Delta Lake

What is Delta Lake?

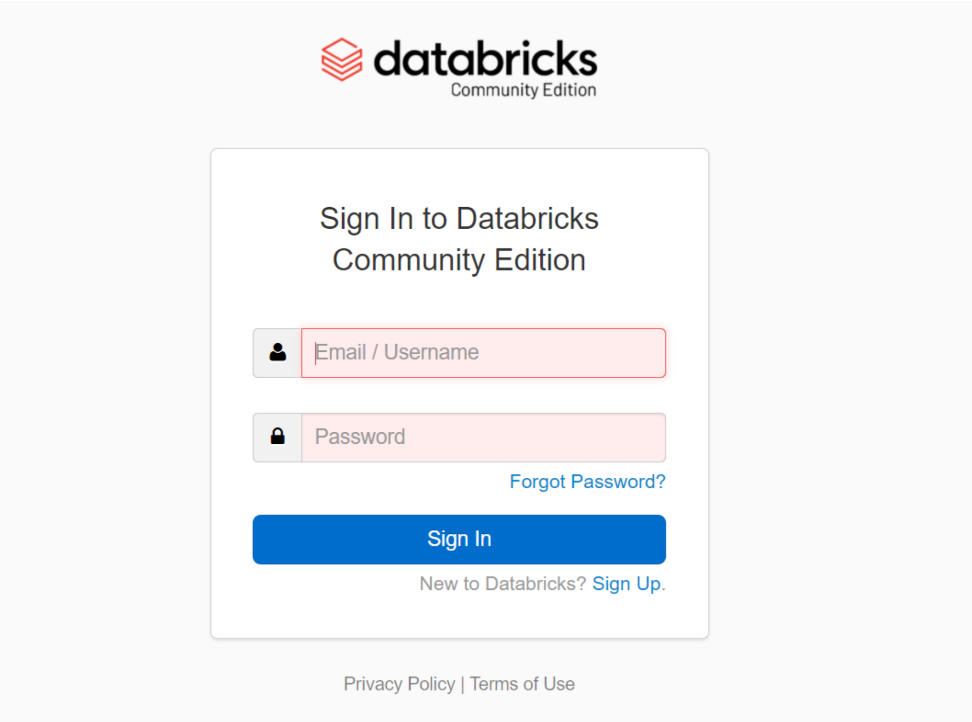
- Specifically designed to work with Apache Spark
- ACID compliant
- Streaming and batch unification
- Schema enforcement
- Time travel
- Upserts and deletes

Example Delta Lake architecture



How can I use Spark?

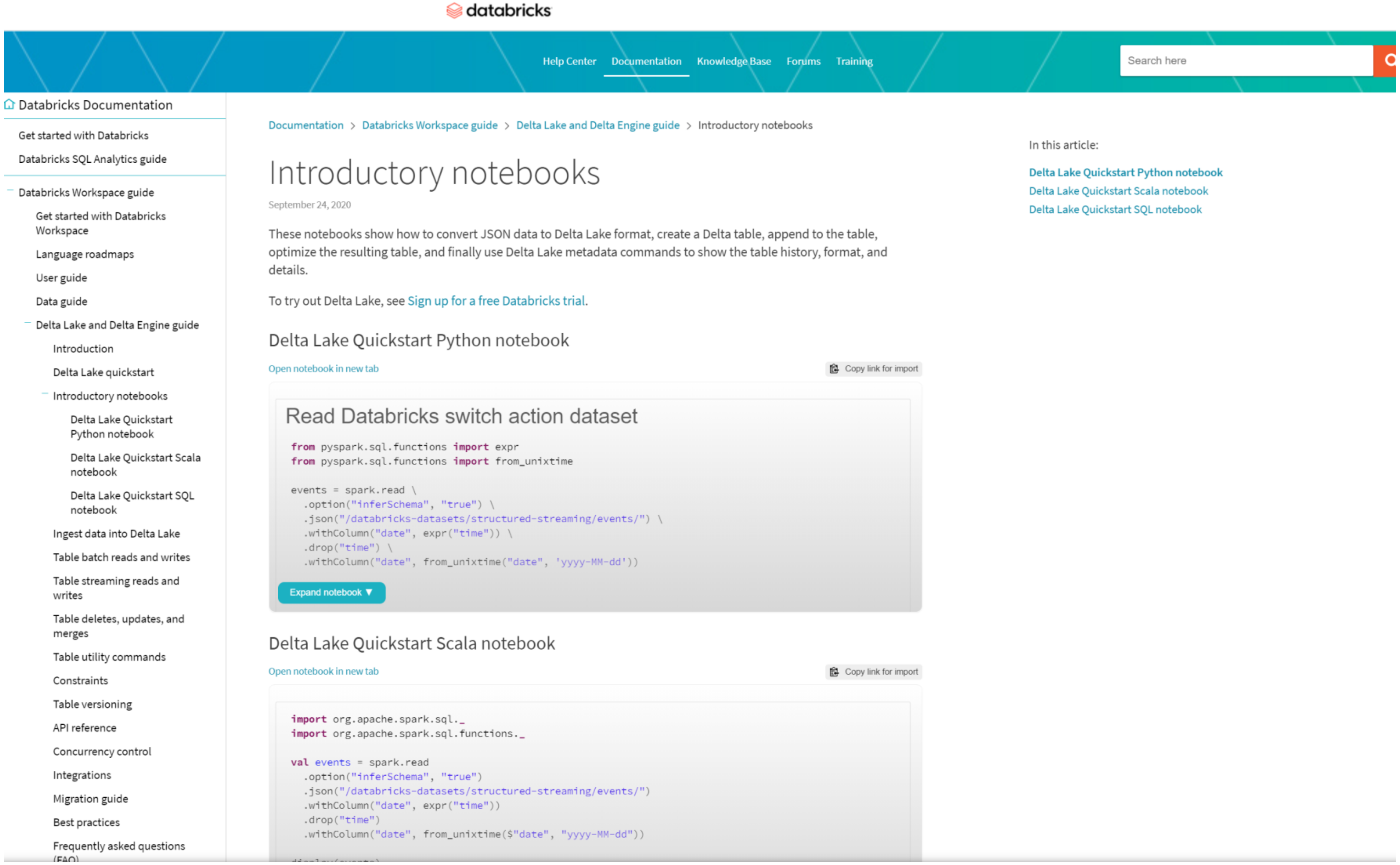
- The ‘unmanaged way’
- The ‘managed way’



Sign In to Databricks Community Edition

Demo

Source: Databricks Documentation


databricks
Help Center
Documentation
Knowledge Base
Forums
Training
Search here
Databricks Documentation
Get started with Databricks
Databricks SQL Analytics guide
Databricks Workspace guide
Get started with Databricks Workspace
Language roadmaps
User guide
Data guide
Delta Lake and Delta Engine guide
Introduction
Delta Lake quickstart
Introductory notebooks
Delta Lake Quickstart Python notebook
Delta Lake Quickstart Scala notebook
Delta Lake Quickstart SQL notebook
Ingest data into Delta Lake
Table batch reads and writes
Table streaming reads and writes
Table deletes, updates, and merges
Table utility commands
Constraints
Table versioning
API reference
Concurrency control
Integrations
Migration guide
Best practices
Frequently asked questions (FAQ)
Documentation > Databricks Workspace guide > Delta Lake and Delta Engine guide > Introductory notebooks
In this article:
Delta Lake Quickstart Python notebook
Delta Lake Quickstart Scala notebook
Delta Lake Quickstart SQL notebook
Open notebook in new tab
Copy link for import
Expand notebook ▼
Open notebook in new tab
Copy link for import

Introductory notebooks

September 24, 2020

These notebooks show how to convert JSON data to Delta Lake format, create a Delta table, append to the table, optimize the resulting table, and finally use Delta Lake metadata commands to show the table history, format, and details.

To try out Delta Lake, see Sign up for a free Databricks trial.

Delta Lake Quickstart Python notebook

Read Databricks switch action dataset

```python
from pyspark.sql.functions import expr
from pyspark.sql.functions import from_unixtime

events = spark.read \
  .option("inferSchema", "true") \
  .json("/databricks-datasets/structured-streaming/events/") \
  .withColumn("date", expr("time")) \
  .drop("time") \
  .withColumn("date", from_unixtime("date", 'yyyy-MM-dd'))
```

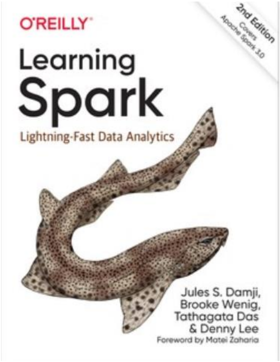
Delta Lake Quickstart Scala notebook

```scala
import org.apache.spark.sql._
import org.apache.spark.sql.functions._

val events = spark.read
  .option("inferSchema", "true")
  .json("/databricks-datasets/structured-streaming/events/")
  .withColumn("date", expr("time"))
  .drop("time")
  .withColumn("date", from_unixtime($"date", "yyyy-MM-dd"))
```

Where do I go from here

- https://www.oreilly.com/library/view/learning-spark-2nd/9781492050032/



O'REILLY

2nd Edition
Covers Apache Spark 3.0

Learning Spark

Lightning-Fast Data Analytics

Jules S. Damji, Brooke Wenig, Tathagata Das & Denny Lee

Foreword by Matei Zaharia

- https://academy.databricks.com/pathway/how-to-build-a-cloud-data-platform

How to Build a Cloud Data Platform for BI and ML

Enroll Now

To register for this pathway please click "Start" below. If you are registering for someone else please check "This is for someone else".

This training is free.

FREE

Categories

All Categories
Public Trainings
Self Paced
Certifications

- https://docs.databricks.com/index.html

Databricks Documentation

Get started with Databricks
Databricks SQL Analytics guide
Databricks Workspace guide
Resources

Updated Jan 12, 2021

Send us feedback

Documentation > Databricks documentation

Databricks documentation

January 05, 2021

This documentation site provides how-to guidance and reference information for Databricks SQL Analytics and Databricks Workspace.

- Get started with Databricks

Thank you!

Any questions?