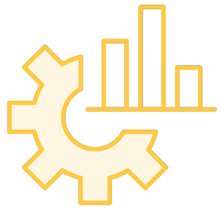
SQL Meets Big Data

Gerald Britton
Pluralsight Author

@GeraldBritton www.linkedin.com/in/geraldbritton

Big Data Origin and Evolution

Exploding digital data



Business analytics

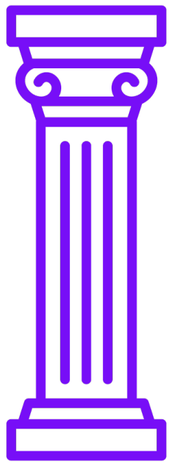
Advancing technology

Data science, machine learning and AI

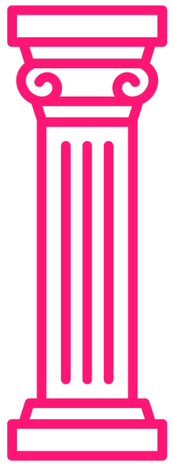
Innovations from open source

Internet of things

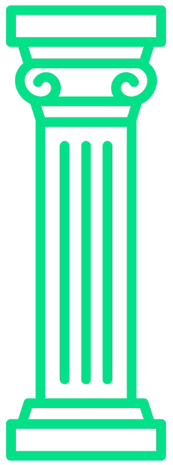
35% market share



HDFS



YARN



MapReduce

Common

Hadoop image licensed under the Apache License, Version 2.0

Using Apache Hive to Query Hadoop

Understanding the Hadoop Interactive Virtual Environment

Highlights of Hive

SQL for querying

Schema on read

Flexible data storage

Batch processing

Integrated with Hadoop

```
CREATE EXTERNAL TABLE IF NOT EXISTS
  employee_data (
    employee_id INT,
    employee_name STRING,
    department STRING,
    salary FLOAT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LOCATION
  '/user/hive/warehouse/employee_data';

SELECT department,
       AVG(salary) AS average_salary
FROM employee_data
GROUP BY department;

ROW FORMAT SERDE
  'org.apache.hive.hcatalog.data.JsonSerDe'
LOCATION
  '/user/hive/warehouse/employee_data';
```

- Create an external table to store data
- Define format of data as CSV
- Specify location of data
- Get average salary per department
- Define JSON format

Sparking Powerful Analyses

Utilizing Apache Spark for increased performance, flexibility and advanced analytics

Sparking Your Interest

Performance: in-memory computation

Flexibility: APIs in Python, Java, Scala and R

Real-time processing: stream data as it arrives

Advanced analytics: complex analyses and machine learning

Integration: Hadoop, Kafka, Cassandra and others

```
from pyspark.sql import SparkSession

spark = SparkSession.builder \
    .appName("InteractiveQueryExample") \
    .getOrCreate()

df = spark.read.csv("sample.csv",
   header=True, inferSchema=True)

df.createOrReplaceTempView("people")

result = spark.sql(
    "SELECT * FROM people WHERE age < 30")

result.show()
```

- Import the SparkSession module
- Initialize Spark session
- Load a sample dataset
- Query the data set

```
val df = spark.read.option(
    "header", "true")
    .csv("sample.csv")
df.createOrReplaceTempView("people")
val result = spark.sql(
    "SELECT *
    FROM people
    WHERE age < 30")
result.show()
```

◀ **Load the sample dataset**

◀ **Query the data set**

Taking a Splash in a Data Lake

Keeping data in its original, raw format without structuring it beforehand.

Data Lake Characteristics

Scalable

Flexible

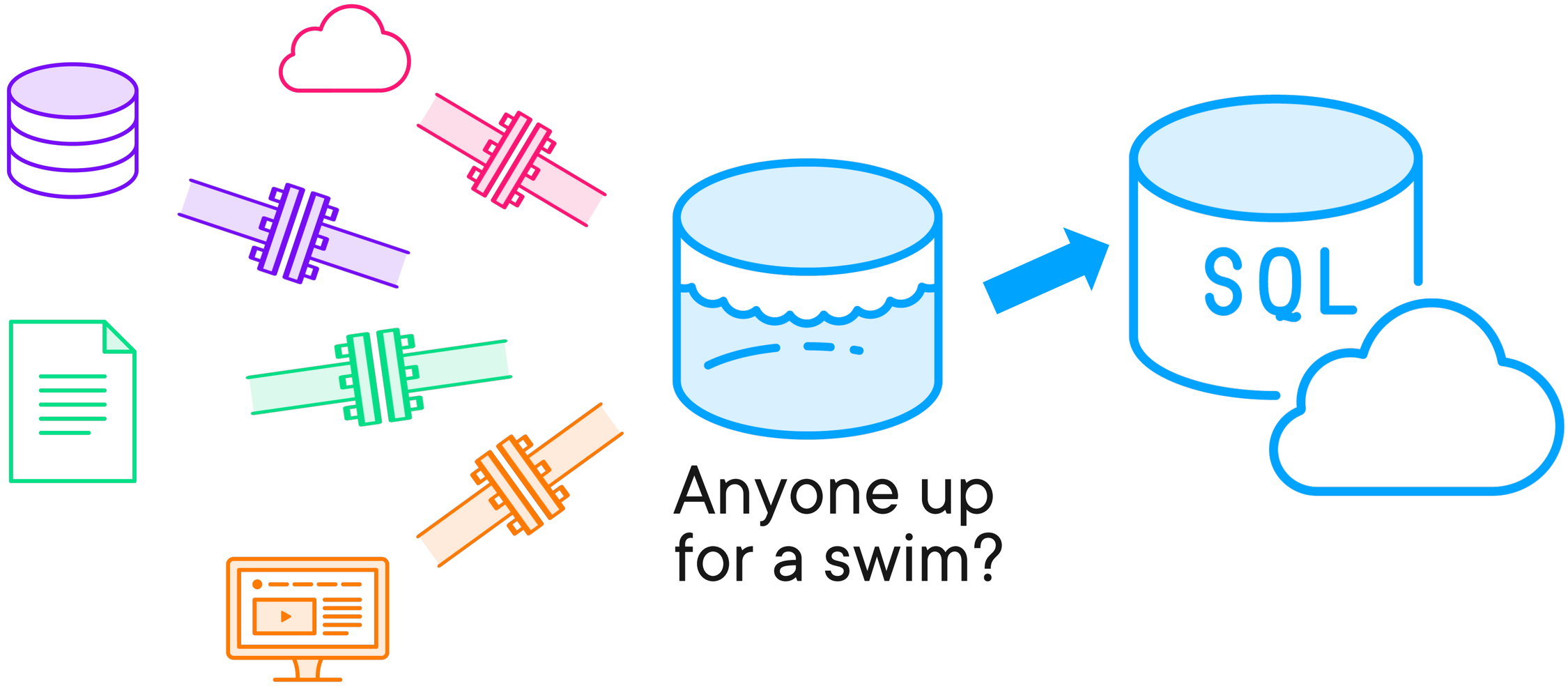
Cost-effective

Schema on Read

Exploration and Analysis

Governance and Security

Data Lake



Decentralization with Data Meshes

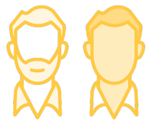
Organizing data based on business domains rather than technology or organizational structures

Data Mesh Characteristics

Decentralization – No one team handling the data

Domain oriented – Data organized by business segments

Data as a product – High-quality data for business consumers



Self-serve infrastructure – Each domain free to manage their data

Governance – Quality, security and compliance

SQL for Data Engineers

Gerald Britton
Pluralsight Author

@GeraldBritton www.linkedin.com/in/geraldbritton