ASSALAMU’ALAIKUM

Introduction to NoSQL Databases and PySpark

DR. RAHMAD KURNIAWAN, ST., MIT., (MTA., CISDV.)

Learning Objectives

- Understand the basics of NoSQL databases by interacting with a simple MongoDB instance.
- Get hands-on experience with distributed computing by running basic operations using PySpark in Python.

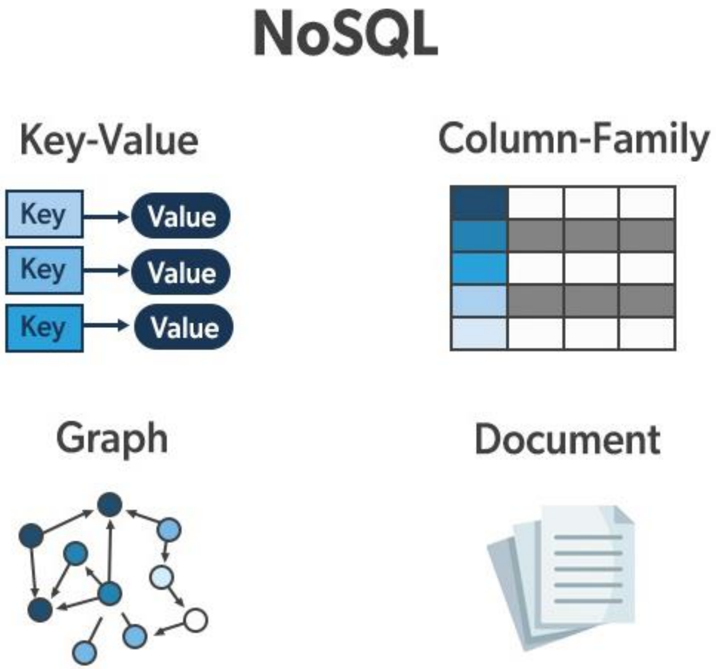
Outline

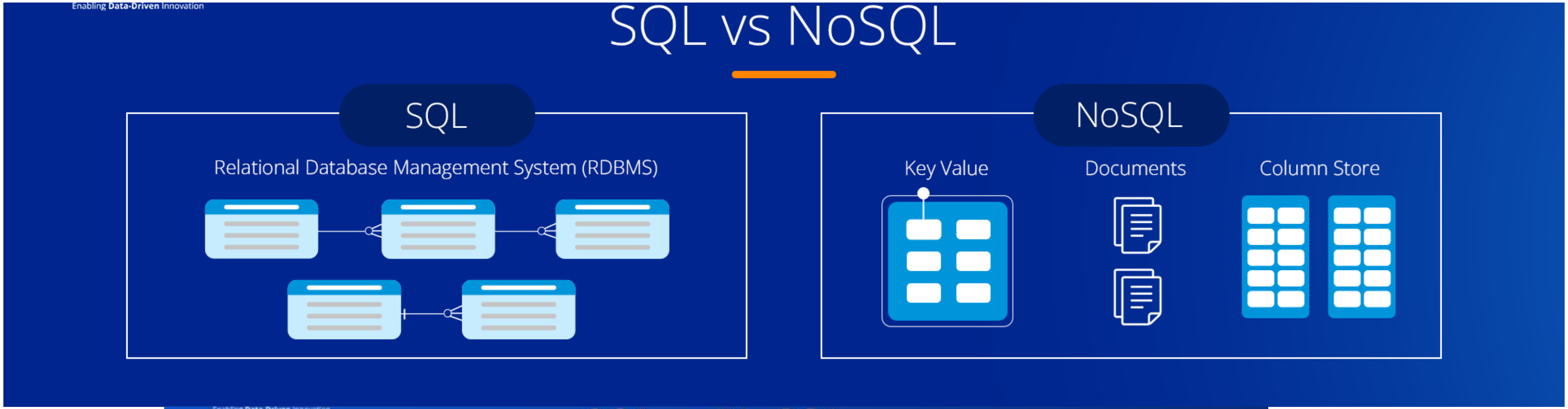
- What are NoSQL Databases?
- Types of NoSQL Databases
- Key Features of NoSQL
- Introduction to PySpark
- Why Use NoSQL with PySpark?
- Practical Applications

What are NoSQL Databases?

What is NoSQL?

- NoSQL stands for "Not Only SQL"
- A NoSQL database provides a mechanism for storage and retrieval of data that is modeled differently from relational databases
- NoSQL databases are used for handling large amounts of unstructured, semi-structured, or structured data





Enabling Data-Driven Innovation

SQL vs NoSQL

SQL

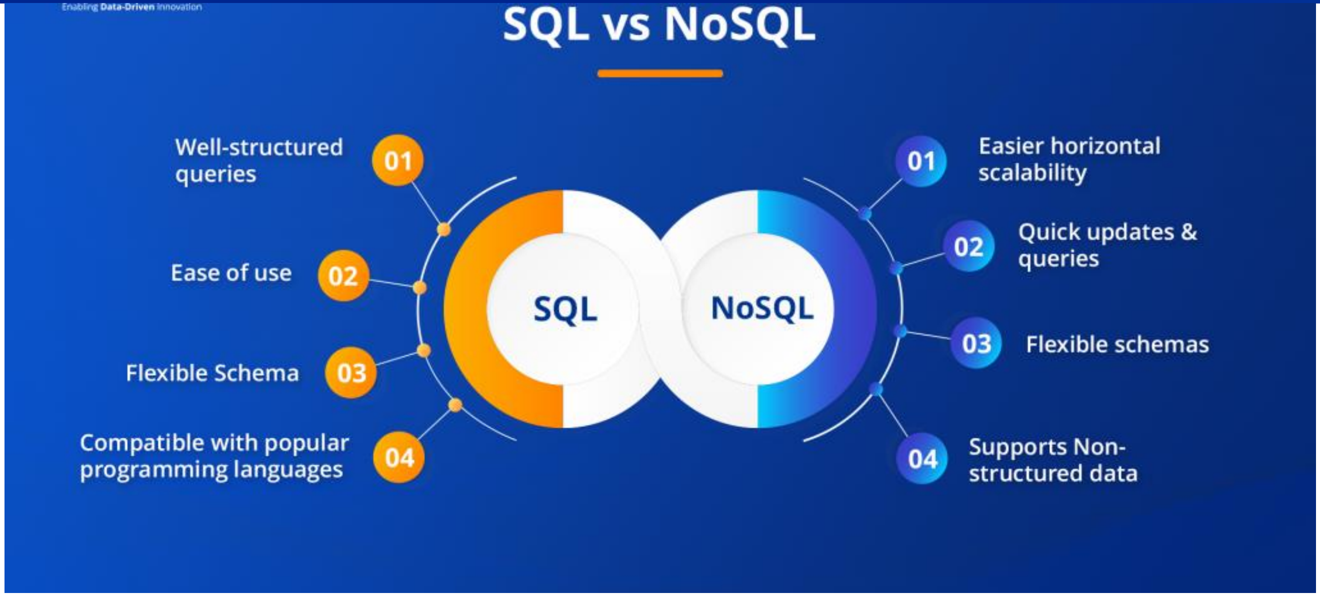
Relational Database Management System (RDBMS)

NoSQL

Key Value

Documents

Column Store



Enabling Data-Driven Innovation

SQL vs NoSQL

SQL

01 Well-structured queries

02 Ease of use

03 Flexible Schema

04 Compatible with popular programming languages

NoSQL

01 Easier horizontal scalability

02 Quick updates & queries

03 Flexible schemas

04 Supports Non-structured data

Differences between SQL and NoSQL

Feature	SQL Databases	NoSQL Databases
Data Model	Relational (tables)	Non-relational (document, key-value, graph, etc.)
Schema	Predefined schema	Dynamic schema
Scalability	Vertical scaling	Horizontal scaling
Use Case	Structured data	Unstructured, dynamic data
ACID Compliance	Strict	Eventual consistency

Types of NoSQL Databases

- **Document-Oriented Databases (e.g., MongoDB)**
  - Stores data as documents, typically in JSON or BSON format
- **Key-Value Databases (e.g., Redis, DynamoDB)**
  - Stores data as a collection of key-value pairs
- **Column-Oriented Databases (e.g., Cassandra)**
  - Organizes data in columns rather than rows
- **Graph Databases (e.g., Neo4j)**
  - Focuses on relationships between data nodes

Key Features of NoSQL Databases

Scalability

- Horizontally scalable, handling large-scale data across multiple servers.

Flexibility

- No fixed schema, allowing more flexibility with data types and structure.

High Performance

- Optimized for big data and high-velocity data.

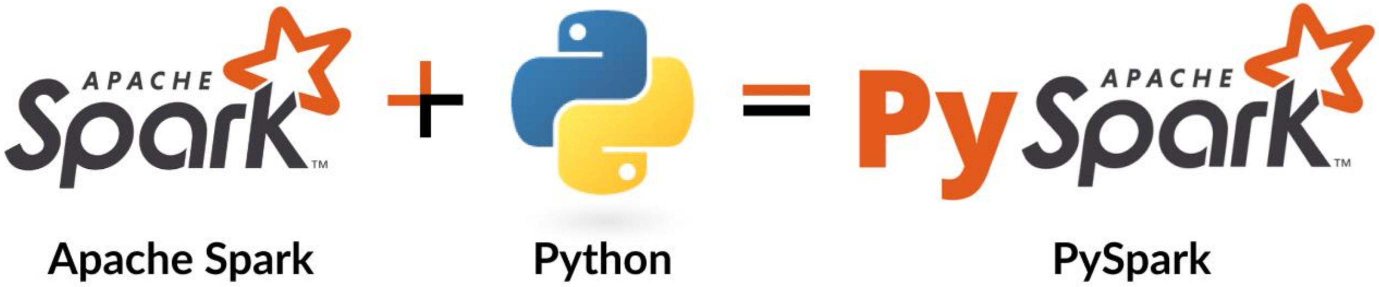
Distributed Architecture

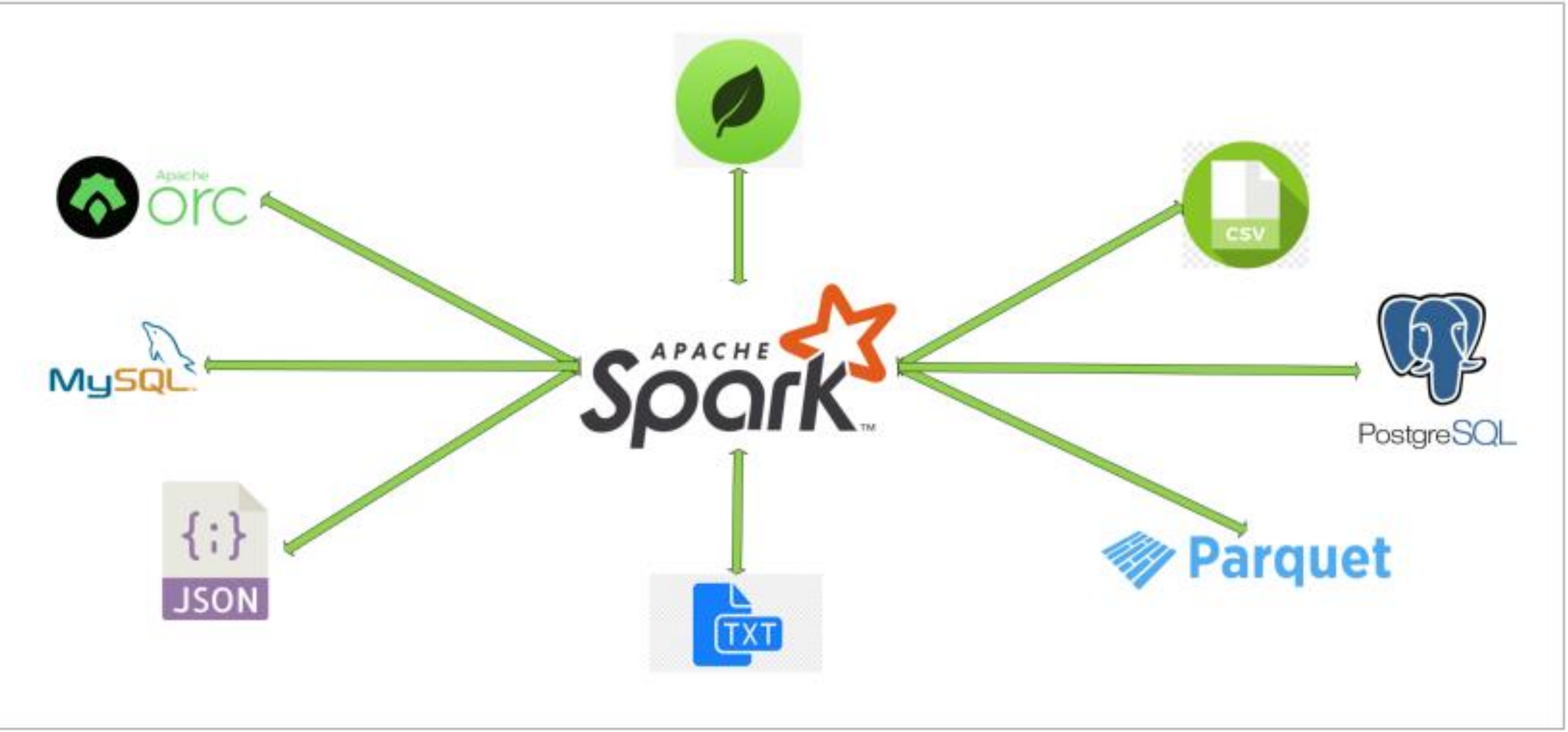
- Data can be spread across multiple locations for better reliability and performance.

Introduction to PySpark

❖ What is PySpark?

- PySpark is the Python API for Apache Spark, a powerful distributed computing framework.
- It allows the processing of big data in a distributed fashion using the Spark engine.
- PySpark can handle large-scale data processing tasks that are too big for traditional systems.





Apache orc
MySQL
JSON
APACHE Spark
TXT
CSV
PostgreSQL
Parquet

Key Features of PySpark

In-Memory Computing:

- Data is cached in memory for faster processing.

Fault Tolerance:

- Automatically handles node failures with its Resilient Distributed Datasets (RDD).

Distributed Processing:

- Works across a cluster of machines for better efficiency.

Flexible APIs:

- Supports multiple languages including Python, Java, Scala, and R.

Why Use NoSQL with PySpark?

Handling Unstructured Data:

- NoSQL is great for unstructured data, which PySpark can process at scale.

Scalability:

- Both NoSQL and PySpark are highly scalable, making them ideal for distributed big data systems.

Real-Time Processing:

- PySpark allows for real-time data streaming and batch processing of NoSQL data.

Integration:

- PySpark integrates well with NoSQL databases like MongoDB and Cassandra.

Example Architecture

- **Data Ingestion:**
  - Data is ingested from various sources (e.g., IoT devices, social media, logs).
- **NoSQL Database:**
  - Stores data in a distributed, flexible, and scalable NoSQL database (e.g., MongoDB).
- **PySpark Processing:**
  - Data is processed in real-time or batch mode using PySpark for analytics.
- **Visualization:**
  - Processed data is visualized or sent to other systems for decision-making.

Practical Applications

Social Media Analytics

- Using NoSQL databases for storing and analyzing social media data with PySpark.

IoT Data Processing

- Handling large streams of IoT data using NoSQL databases and PySpark for real-time analytics.

Recommendation Systems

- Building real-time recommendation engines using PySpark and NoSQL databases like Cassandra.

- Thank you