

What is it?

RDDs are the fundamental data structure in Spark, representing distributed collections of objects.

- Key Features:
- ✓ Immutable and distributed across multiple nodes.
- ✓ Supports lazy evaluation (executions are triggered only when an action is performed).

- ✓ Fault-tolerant (recomputes lost data automatically).
- ✓ Can use transformations (map, filter, reduceByKey, etc.) and actions (collect, count, etc.).
- ✓ Supports both structured and unstructured data.
- Limitations:
- X No inbuilt optimization like DataFrames.
- X Performance is lower compared to DataFrames and Datasets



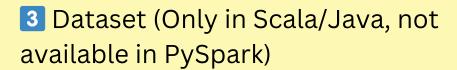
What is it?

A DataFrame is a distributed table-like structure similar to pandas DataFrames, built on top of RDDs but optimized using Spark's Catalyst optimizer.

- Key Features:
- ✓ Schema-based, similar to SQL tables (uses columns and rows).
- ✓ Can handle structured and semi-structured data (JSON, CSV, etc.).
- ✓ Supports Catalyst optimizer for better performance.



- ✓ High-level API for easier operations.
- Limitations:
- X Less flexible than RDDs when dealing with low-level transformations.
- X Not type-safe (in Python, no compile-time type checking).



What is it?

A Dataset is a type-safe version of a DataFrame in Spark. It offers the performance benefits of DataFrames while ensuring compile-time type safety.

- Key Features:
- ✓ Type-safe and object-oriented (avoids runtime errors).
- ✓ Optimized using the Catalyst optimizer.

- ✓ Uses Encoders for efficient serialization.
- ✓ Offers both functional (like RDDs) and SQL-like (like DataFrames) operations.
- Limitations:
- X Not available in PySpark (since Python is dynamically typed).
- X Slightly slower than DataFrames because of type enforcement.