# DataFrames and Spark SQL

In this lesson, you learned that:

RDDs are Spark’s primary data abstraction partitioned across the cluster’s nodes.

Spark uses directed acyclic graphs (DAGs) to enable fault tolerance. When a node goes down, Spark replicates the DAG and restores the node.

Transformations undergo lazy evaluation, meaning they are only evaluated when the driver function calls an action.

A data set is a distributed collection of data that provides the combined benefits of both RDDs and SparkSQL.

Consisting of strongly typed JVM objects, data sets use DataFrame typesafe capabilities and extend object-oriented API capabilities.

Data sets work with both Scala and Java APIs.  DataFrames are not typesafe. You can use APIs in Java, Scala, and Python. Data sets are Spark’s latest data abstraction.

Spark SQL optimization’s primary goal is to improve a SQL query’s run-time performance by reducing the query’s time and memory consumption, saving organizations time and money.

Catalyst is the Spark SQL built-in rule-based query optimizer. Catalyst performs analysis, logical optimization, physical planning, and code generation.

Tungsten is the Spark built-in cost-based optimizer for CPU and memory usage that enables cache-friendly computation of algorithms and data structures.

Basic DataFrame operations are reading, analysis, transformation, loading, and writing.

You can use a Pandas DataFrame in Python to load a data set and apply the `printschema`, `select`, or `show` function for data analysis.

Keep only relevant data for transform tasks and apply functions such as filters, joins, column operations, grouping and aggregations, and other functions.

Spark SQL consists of Spark modules for structured data processing that can run SQL queries on Spark DataFrames and are usable in Java, Scala, Python, and R.

Spark SQL supports both temporary views and global temporary views. Use a DataFrame function or an SQL Query and Table view for data aggregation. Spark SQL supports Parquet files, JSON data sets, and Hive tables.