Lesson 4

Spark DataFrame and RDDs

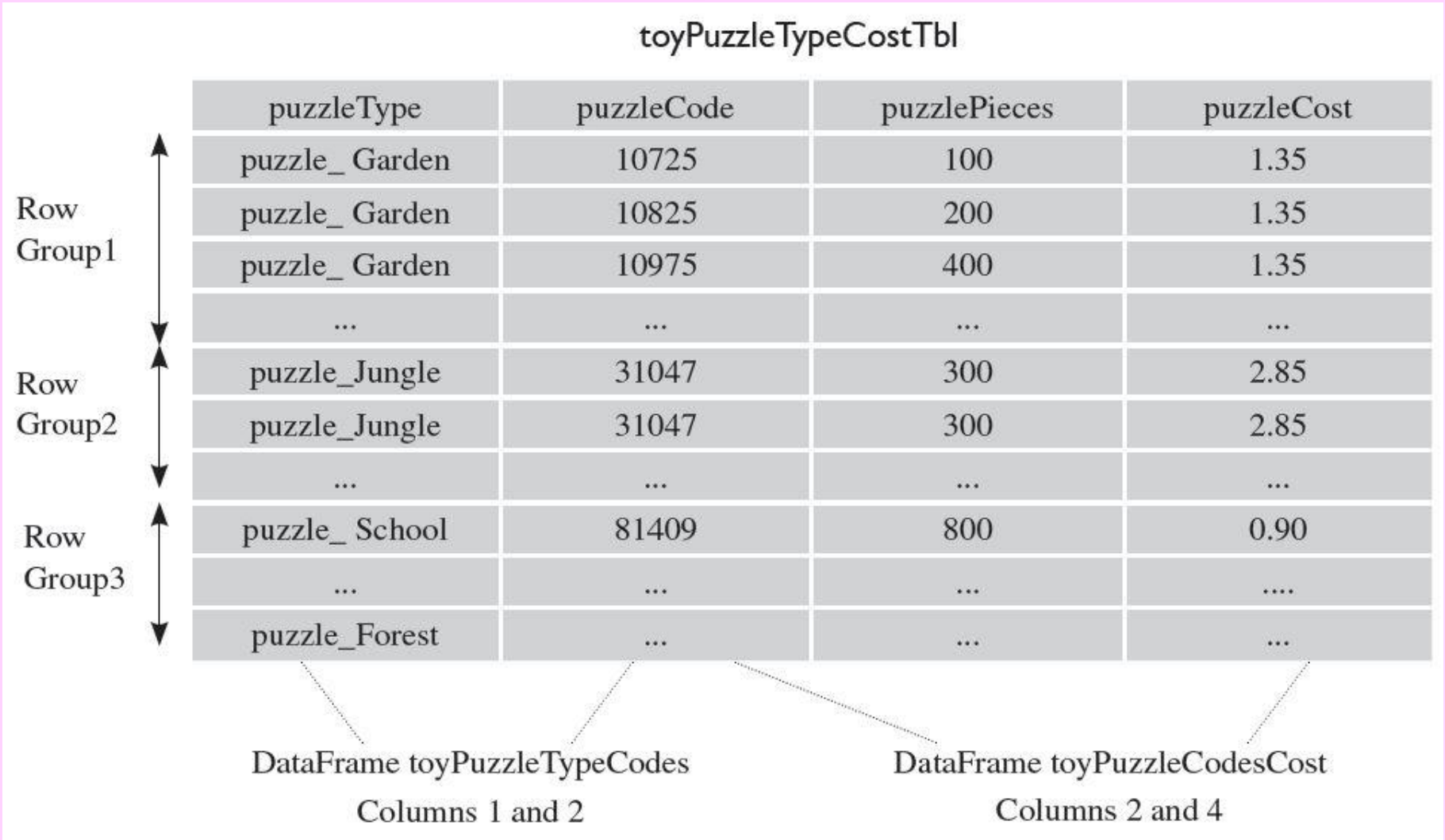
Spark DataFrame

- A distributed collection of data organized into named columns
- Used for transformation using filter, join, or groupBy aggregation functions
- Refer Section 10.4.1 for Merge and Join Functions for DataFrame Objects

DataFrames

- Created from several data sources,
- JSON datasets, Hive tables, Parquet row groups, structured data files, external Data Stores and RDDs
- Usages of DataFrames from the Parquet and JSON objects
- Section 10.3.3 for conversion from CSV format dataset

Figure 5.6 Sample table toyPuzzleTypeCostTbl rows, row groups in DataFrames



DataFrame (SchemaRDD)

- DataFrame, earlier named as SchemaRDD is similar to a table in a traditional database
- The schema is blueprint for the organization of data in an RDD (similar to traditional database schema)
- The schema tells how the RDD constructs
- Refer Section 10.3.4 for creating DataFrame from the RDD

DataFrame (SchemaRDD)

- The SchemaRDD returns on the queries loading or execution. A SchemaRDD is composed of row objects. The SchemaRDD has additionally the 'Data Type' information for each column. A row object wraps the arrays of basic data types.

Spark Resilient Distributed Dataset (RDD)

- A collection of objects distributed on many computing nodes
- Parallel structures on clusters
- immutable (thus read-only) and partitioned distributed collection of objects,

RDD Features

- Have an interface which enables transformations that apply the same to many data objects,
- Each RDD can split into multiple partitions, which may be computed in parallel on different nodes of a cluster
- computations,
-

RDD Features

- Fault-tolerant abstraction which enables In-Memory cluster
- create only through the deterministic operations on either (i) data in stable Data store such as file or (ii) operations on other RDDs,
- Enable efficient execution of iterative algorithms, and interactive data-mining

RDD Features

- Commands to them, enable the intermediate-results explicitly persist in memory, and
- Controls the partitioning so that placement of data optimizes and partitions can be manipulated using operators

Spark RDD immutability

- Not capable of or not susceptible to change
- A new RDD creates on transform and action commands

Commands for Creation of New RDD

(i) load an external dataset as a distributed collection of objects, or

(ii) use a driver (program) for distributing a collection of objects.

Transform operation

- Each dataset represents an object
- The transform-command invokes the methods using the objects to create new RDD(s)
- Transform operations create RDDs from each other
- Example 5.9

Action Operation

- (i) returns a value into a program or (ii) exports data to a Data Store.
- The action command does the computation when a first-time action takes place on an RDD and returns a value or sends data to a Data Store.
- Example 5.9

Removing Data

- Auto-monitoring in Spark auto-monitors the usages of caches
- Spark removes the caches using ‘least recently used partitions removed first’ strategy
- RDD.unpersist()

Spark data types and their descriptions.

- Table 5.4

Numeric Operations on RDD

- Table 5.6
- count (*), count (expr); sum (col), sum (DISTINCT col), avg (col), avg (DISTINCT col), min (col) and DOUBLE max(col) (Table 4.10).

Use of Statistical Functions

- The statistical functions stdev(), sampleStdev(), variance, sampleVariance() for analysis with DataFrames in input

Shared Variables

- Broadcast Variable:
- Created from a value denoted by a variable v and running method sc.broadcast (v). The broadcast variable is a wrapper around v.
- Example 5.11

Shared Variables

- Accumulator Variable:
- Accumulators are special variables. They add the values using associative and commutative operations. They also support parallel run: for example, in count() or sum()
- Example 5.12

Summary

We learnt :

- DataFrame
- Resilient Data Sets
- RDD Features
- Transformation and Action
- Shared Variables

End of Lesson 4 on

Spark DataFrame and RDDs