databricks Spark Dataset Example

Dataset Using Range

```
val ds = spark.range(3)
ds.show()
```

```
+---+
| id|
+---+
|  0|
|  1|
|  2|
+---+
```

```
ds: org.apache.spark.sql.Dataset[Long] = [id: bigint]
```

Dataset Using Sequence

```
val ds = Seq(11, 22, 33).toDS()
ds.show()
```

```
+-----+
|value|
+-----+
|   11|
|   22|
|   33|
+-----+
```

```
ds: org.apache.spark.sql.Dataset[Int] = [value: int]
```

Dataset Using List

```
val ds = List(10,20,30).toDS
ds.show()
```

```
+-----+
|value|
+-----+
|   10|
|   20|
|   30|
+-----+
```

```
ds: org.apache.spark.sql.Dataset[Int] = [value: int]
```

Dataset Using Sequence of Case Classes

```
case class Book(name: String, cost: Int)
val bookDS = Seq(Book("Scala", 400), Book("Spark", 500), Book("Kafka",
300)).toDS()
bookDS.show()
```

```
+-----+----+
| name|cost|
+-----+----+
|Scala| 400|
|Spark| 500|
|Kafka| 300|
+-----+----+

defined class Book
bookDS: org.apache.spark.sql.Dataset[Book] = [name: string, cost: int]
```

Dataset Using RDD

```
val rdd = sc.parallelize(Seq(("Spark",500), ("Scala",400),("Kafka",300)))
val integerDS = rdd.toDS()
integerDS.show()
```

```
+-----+---+
|   _1| _2|
+-----+---+
|Spark|500|
|Scala|400|
|Kafka|300|
+-----+---+

rdd: org.apache.spark.rdd.RDD[(String, Int)] = ParallelCollectionRDD[15] at
parallelize at command-1988893286232020:1
integerDS: org.apache.spark.sql.Dataset[(String, Int)] = [_1: string, _2: in
t]
```

Dataset from Dataframe using Case Class

```
// Seq[Book] -> RDD[Book] -> Dataframe -> Dataset[Book]
case class Book(name: String, cost: Int)
val bookSeq = Seq(Book("Scala", 400), Book("Spark", 500), Book("Kafka",
300))
val bookRDD = sc.parallelize(bookSeq)
val bookDF = bookRDD.toDF()
val bookDS = bookDF.as[Book]
bookDS.show()
```

```
+-----+----+
| name|cost|
+-----+----+
|Scala| 400|
```

```
|Spark| 500|
|Kafka| 300|
+-----+----+

defined class Book
bookSeq: Seq[Book] = List(Book(Scala,400), Book(Spark,500), Book(Kafka,300))
bookRDD: org.apache.spark.rdd.RDD[Book] = ParallelCollectionRDD[0] at parall
elize at command-1988893286232021:4
bookDF: org.apache.spark.sql.DataFrame = [name: string, cost: int]
bookDS: org.apache.spark.sql.Dataset[Book] = [name: string, cost: int]
```

Dataset from Dataframe using Tuples

```
// Seq[(String, Int)] -> RDD[(String, Int)] -> Dataframe -> Dataset[(String,
Int)]
val bookSeq = Seq(("Scala", 400), ("Spark", 500), ("Kafka", 300))
val bookRDD = sc.parallelize(bookSeq)
val bookDF = bookRDD.toDF("Id", "Name")
val bookDS = bookDF.as[(String, Int)]
bookDS.show()

+-----+----+
|   Id|Name|
+-----+----+
|Scala| 400|
|Spark| 500|
|Kafka| 300|
+-----+----+

bookSeq: Seq[(String, Int)] = List((Scala,400), (Spark,500), (Kafka,300))
bookRDD: org.apache.spark.rdd.RDD[(String, Int)] = ParallelCollectionRDD[12]
at parallelize at command-702698878027611:3
bookDF: org.apache.spark.sql.DataFrame = [Id: string, Name: int]
bookDS: org.apache.spark.sql.Dataset[(String, Int)] = [Id: string, Name: in
t]
```

Word Count Example using Dataset

```
val linesDS = sc.parallelize(Seq("Spark is fast", "Spark has Dataset",
"Spark Dataset is typesafe")).toDS()
val wordsDS = linesDS.flatMap(_.toLowerCase.split(" ")).filter(_ != "")
val groupedDS = wordsDS.groupBy("value")
val countsDS = groupedDS.count()
countsDS.show()

+--------+-----+
|   value|count|
+--------+-----+
|typesafe|    1|
|    fast|    1|
```

```
|      is|    2|
| dataset|    2|
|   spark|    3|
|     has|    1|
+--------+-----+

linesDS: org.apache.spark.sql.Dataset[String] = [value: string]
wordsDS: org.apache.spark.sql.Dataset[String] = [value: string]
groupedDS: org.apache.spark.sql.RelationalGroupedDataset = RelationalGrouped
Dataset: [grouping expressions: [value: string], value: [value: string], typ
e: GroupBy]
countsDS: org.apache.spark.sql.DataFrame = [value: string, count: bigint]
```

Convert Dataset to Dataframe

```
+--------+-----+
|   value|count|
+--------+-----+
|   spark|    3|
| dataset|    2|
|      is|    2|
|typesafe|    1|
|    fast|    1|
|     has|    1|
+--------+-----+

warning: there was one feature warning; re-run with -feature for details
countsDF: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [value: s
tring, count: bigint]
```