Using Explode, User Defined Functions, and Pivot

Lesson Objectives

- After completing this lesson, you should be able to use these methods on DataFrames:
  - `explode()`
  - User Defined Functions
  - `pivot()`

How can we turn this sales data …

1	Acme	2013	1000	Jim,Tom
2	Lumos	2013	1100	Fred,Ann
3	Acme	2014	2800	Jim
4	Lumos	2014	1200	Ann
5	Acme	2014	4200	Fred,Sally

… into this report?

1	Acme	2013	1000	Jim,Tom
2	Lumos	2013	1100	Fred,Ann
3	Acme	2014	2800	Jim
4	Lumos	2014	1200	Ann
5	Acme	2014	4200	Fred,Sally

Ann	550	1200
Fred	550	2100
Jim	500	2800
Sally		2100
Tom	500	

Setting up the example

```
Sales = namedtuple("Sales",["id","account","year","commission","sales_reps"])
```

```
sales = sc.parallelize([Sales(1, "Acme", "2013", 1000, ["Jim", "Tom"]),
         Sales(2, "Lumos", "2013", 1100, ["Fred", "Ann"]),
         Sales(3, "Acme", "2014", 2800, ["Jim"]),
         Sales(4, "Lumos", "2014", 1200, ["Ann"]),
         Sales(5, "Acme", "2014", 4200, ["Fred", "Sally"])]).toDF()
sales.show()
```

```
+---+-------+----+----------+-------------+
| id|account|year|commission|   sales_reps|
+---+-------+----+----------+-------------+
|  1|   Acme|2013|      1000|   [Jim, Tom]|
|  2|  Lumos|2013|      1100|  [Fred, Ann]|
|  3|   Acme|2014|      2800|        [Jim]|
|  4|  Lumos|2014|      1200|        [Ann]|
|  5|   Acme|2014|      4200|[Fred, Sally]|
+---+-------+----+----------+-------------+
```

explode()

```
from pyspark.sql.functions import explode
sales.select("id","account","year","commission",explode("sales_reps").alias("sales_rep")).show()
```

```
+---+-------+----+----------+---------+
| id|account|year|commission|sales_rep|
+---+-------+----+----------+---------+
|  1|   Acme|2013|      1000|      Jim|
|  1|   Acme|2013|      1000|      Tom|
|  2|  Lumos|2013|      1100|     Fred|
|  2|  Lumos|2013|      1100|      Ann|
|  3|   Acme|2014|      2800|      Jim|
|  4|  Lumos|2014|      1200|      Ann|
|  5|   Acme|2014|      4200|     Fred|
|  5|   Acme|2014|      4200|    Sally|
+---+-------+----+----------+---------+
```

User Defined Functions

```
from pyspark.sql.functions import UserDefinedFunction
from pyspark.sql.types import IntegerType

column_len = UserDefinedFunction(lambda x: len(x), IntegerType())

exploded = sales.select("id","account","year","commission",
             column_len("sales_reps").alias("num_reps"),
             explode("sales_reps").alias("sales_rep"))
exploded.show()
```

```
+---+-------+----+----------+--------+---------+
| id|account|year|commission|num_reps|sales_rep|
+---+-------+----+----------+--------+---------+
|  1|   Acme|2013|      1000|       2|      Jim|
|  1|   Acme|2013|      1000|       2|      Tom|
|  2|  Lumos|2013|      1100|       2|     Fred|
|  2|  Lumos|2013|      1100|       2|      Ann|
|  3|   Acme|2014|      2800|       1|      Jim|
|  4|  Lumos|2014|      1200|       1|      Ann|
|  5|   Acme|2014|      4200|       2|     Fred|
|  5|   Acme|2014|      4200|       2|    Sally|
+---+-------+----+----------+--------+---------+
```

… and it produces this

```
exploded = exploded.withColumn("share", exploded.commission / exploded.num_reps).drop("num_reps")
exploded.show()
```

```
+---+-------+----+----------+---------+------+
| id|account|year|commission|sales_rep| share|
+---+-------+----+----------+---------+------+
|  1|   Acme|2013|      1000|      Jim| 500.0|
|  1|   Acme|2013|      1000|      Tom| 500.0|
|  2|  Lumos|2013|      1100|     Fred| 550.0|
|  2|  Lumos|2013|      1100|      Ann| 550.0|
|  3|   Acme|2014|      2800|      Jim|2800.0|
|  4|  Lumos|2014|      1200|      Ann|1200.0|
|  5|   Acme|2014|      4200|     Fred|2100.0|
|  5|   Acme|2014|      4200|    Sally|2100.0|
+---+-------+----+----------+---------+------+
```

A quick reminder

Our goal is to produce this 

Ann	550	1200
…		

2	Lumos	2013	1100	550	Ann
4	Lumos	2014	1200	1200	Ann
…					

and so far we have this

pivot()

```
exploded.groupBy("sales_rep").pivot("year").sum("share").orderBy("sales_rep").show()
```

```
+---------+-----+------+
|sales_rep| 2013|  2014|
+---------+-----+------+
|      Ann|550.0|1200.0|
|     Fred|550.0|2100.0|
|      Jim|500.0|2800.0|
|    Sally| null|2100.0|
|      Tom|500.0|  null|
+---------+-----+------+
```

Pivoting with groupBy on two columns

```
exploded.groupBy("account", "sales_rep").pivot("year").sum("share").orderBy("account", "sales_rep").show()
```

```
+-------+---------+-----+------+
|account|sales_rep| 2013|  2014|
+-------+---------+-----+------+
|   Acme|     Fred| null|2100.0|
|   Acme|      Jim|500.0|2800.0|
|   Acme|    Sally| null|2100.0|
|   Acme|      Tom|500.0|  null|
|  Lumos|      Ann|550.0|1200.0|
|  Lumos|     Fred|550.0|  null|
+-------+---------+-----+------+
```

Lesson Summary

Having completed this lesson, you should understand the role of

- `explode()`
- User Defined Functions
- `pivot()`

in preparing data for further analysis.