In [1]:
```
from pyspark.sql import SparkSession

if __name__ == '__main__':
#Create the spark session
     spark = SparkSession.builder.appName("map transformation").getOrCreate()

#Create a dataset
     data = spark.sparkContext.range(1,5)

#Show the dataset
     print('Dataset')
     print(data.collect())
     print('-------------')

#Use the map function
     rdd = data.map(lambda x: (x, x*x, x*x*x))

#Show the new dataset after the map function
     print('New Dataset')
     print(rdd.collect())
```

```
22/10/10 07:32:33 WARN Utils: Your hostname, Vaishalis-MacBook-Pro.local resolves to a l
oopback address: 127.0.0.1; using 192.168.0.105 instead (on interface en0)
22/10/10 07:32:33 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLev
el).
22/10/10 07:32:33 WARN NativeCodeLoader: Unable to load native-hadoop library for your p
latform... using builtin-java classes where applicable
22/10/10 07:32:34 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting
port 4041.
22/10/10 07:32:34 WARN Utils: Service 'SparkUI' could not bind on port 4041. Attempting
port 4042.
22/10/10 07:32:34 WARN Utils: Service 'SparkUI' could not bind on port 4042. Attempting
port 4043.
Dataset

[1, 2, 3, 4]
-------------
New Dataset
[(1, 1, 1), (2, 4, 8), (3, 9, 27), (4, 16, 64)]
```

In [2]:
```
#New Dataset
data = [(1, 1, 1), (2, 4, 8), (3, 9, 27), (4, 16, 64)]

columns = ["number","squared","cubed"]

#Create DataFrame
df = spark.createDataFrame(data = data, schema = columns)

#show() displays the contents of the DataFrame in a Table Row and Column Format
df.show()
```

```
+------+-------+-----+
|number|squared|cubed|
+------+-------+-----+
|     1|      1|    1|
|     2|      4|    8|
|     3|      9|   27|
|     4|     16|   64|
+------+-------+-----+
```

In [3]:
```
#Double Column 2 and Column 3 and return a new DataFrame
rdd2 = df.rdd.map(lambda x: (x[0], x[1]*2, x[2]*2))
df2 = rdd2.toDF(["number","square_doubled", "cube_doubled"]).show()
```

```
+------+--------------+------------+
|number|square_doubled|cube_doubled|
+------+--------------+------------+
|     1|             2|           2|
|     2|             8|          16|
|     3|            18|          54|
|     4|            32|         128|
+------+--------------+------------+
```

In [4]:
```
data = [('1','1','1'),
        ('2','4','8'),
        ('3','9','27'),
        ('4','16','64')
       ]
columns = ["number","squared","cubed"]

df = spark.createDataFrame(data = data, schema = columns)
df.show()

#Referring Column Names
rdd2 =df.rdd.map(lambda x:
                 (x["number"],x["squared"],x["cubed"]*2)
                )
def func1(x):
    number = x.number
    square = x.squared
    newNumber = x.cubed*2
    return (number,square, newNumber)

rdd2 = df.rdd.map(lambda x: func1(x)).toDF(["number","sqaured_number","new_number"]).sho
```

```
+------+-------+-----+
|number|squared|cubed|
+------+-------+-----+
|     1|      1|    1|
|     2|      4|    8|
|     3|      9|   27|
|     4|     16|   64|
+------+-------+-----+

+------+--------------+----------+
|number|sqaured_number|new_number|
+------+--------------+----------+
|     1|             1|        11|
|     2|             4|        88|
|     3|             9|      2727|
|     4|            16|      6464|
+------+--------------+----------+
```

In [ ]:

In [ ]: