

ICP-8 REPORT

The image displays two sequential screenshots of a Google Colab notebook titled "ICP-8.ipynb".

Top Screenshot: The notebook shows a code cell with the following Python code:

```
from pyspark.sql import SparkSession

# Create SparkSession (this will also initialize SparkContext)
spark = SparkSession.builder.master("local").appName("RDD Example").getOrCreate()

# Access the SparkContext from the SparkSession
sc = spark.sparkContext

# Create RDD with the first 15 natural numbers
numbers_rdd = sc.parallelize(range(1, 16))

# Collect and print the elements of the RDD
print(numbers_rdd.collect())

# Stop the Spark session (optional)
spark.stop()
```

The output of the code cell is displayed below it:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
```

Bottom Screenshot: This screenshot shows the notebook after further operations. The first code cell now contains:

```
[ ] # Show elements
print(numbers_rdd.collect())

# Show number of partitions
print("Number of partitions: ", numbers_rdd.getNumPartitions())
```

The output for this cell is:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
Number of partitions: 1
```

A second code cell below shows the re-initialization and a new operation:

```
from pyspark.sql import SparkSession

# Re-initialize SparkSession and SparkContext
spark = SparkSession.builder.master("local").appName("RDD Example").getOrCreate()
sc = spark.sparkContext

# Re-create the RDD
numbers_rdd = sc.parallelize(range(1, 16))

# Now perform operations on the RDD
first_element = numbers_rdd.first()
print("First element in the RDD:", first_element)
```

The output for this second cell is:

```
First element in the RDD: 1
```

 ICP-8.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

Connect T4 Gemini

+

Code

+

Text


Even numbers: [2, 4, 6, 8, 10, 12, 14]

Squared numbers: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225]

Sum of elements: 120

numbers_rdd.saveAsTextFile("numbers_rdd_output.txt")

another_rdd = sc.parallelize([16, 17, 18, 19, 20])
combined_rdd = numbers_rdd.union(another_rdd)

 ICP-8.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

Connect T4 Gemini

+

Code

+

Text

Combined RDD: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]

Cartesian Product: [(1, 16), (1, 17), (1, 18), (1, 19), (1, 20), (2, 16), (2, 17), (2, 18), (2, 19), (2, 20), (3, 16), (3, 17), (3, 18), (3, 19), (3, 20),

Dictionary RDD: [{'a': 1}, {'b': 2}, {'c': 3}]

data = sc.parallelize([1, 2, 2, 3, 3, 4])
count_rdd = data.map(lambda x: (x, 1)).reduceByKey(lambda a, b: a + b)
print("Unique values with counts: ", count_rdd.collect())

The screenshot displays a Jupyter Notebook with three distinct code cells, each demonstrating a different Spark operation. The interface includes a left sidebar with navigation icons, a top menu bar, and a right sidebar with utility icons.

Cell 1: Parallelize and ReduceByKey

```
[ ] data = sc.parallelize([1, 2, 2, 3, 3, 3, 4])
count_rdd = data.map(lambda x: (x, 1)).reduceByKey(lambda a, b: a + b)
print("Unique values with counts: ", count_rdd.collect())
```

Output: Unique values with counts: [(1, 1), (2, 2), (3, 3), (4, 1)]

Cell 2: Text Files

```
# Create RDD by combining multiple text files (file1.txt and file2.txt)
rdd_from_files = sc.textFile("file1.txt,file2.txt")

# Show the content from the text files
print("RDD from text files:", rdd_from_files.collect())
```

Output: RDD from text files: ['This is the first file.', 'It contains some text data.', 'Spark is awesome.', 'This is the second file.', 'It also contains text data.']

```
[ ] # Use the 'take' action to retrieve the first 5 lines
first_five_lines = rdd_from_files.take(5)
print("First 5 lines of the RDD:", first_five_lines)
```

Output: First 5 lines of the RDD: ['This is the first file.', 'It contains some text data.', 'Spark is awesome.', 'This is the second file.', 'It also contains text data.']

Cell 3: DataFrame

```
# Create a DataFrame
data = [("sachin", 10), ("rohit", 45), ("virat", 18)]
df = spark.createDataFrame(data, ["Name", "Jersey"])

# Show the DataFrame
df.show()
```

Output:

Name	Jersey
sachin	10
rohit	45
virat	18

Cell 4: RDD Example (low-level operations)

```
# RDD Example (low-level operations)
rdd_example = sc.parallelize([("sachin", 10), ("rohit", 45), ("virat", 18)])
df_example = spark.createDataFrame(rdd_example, ["Name", "jersey"])
df_example.show()
```

Output:

Name	jersey
sachin	10
rohit	45
virat	18

My github link : <https://github.com/PraveenDondapati/bda.git>