PySpark CHEAT SHEET FOR DATA ENGINEERS

Master these essential PySpark commands to build efficient and scalable data pipelines!

Abhishek Agrawal
Data Engineer

SparkSession (Starting Point)

```
from pyspark.sql import SparkSession

# Create a SparkSession
spark = SparkSession.builder.appName("Example").getOrCreate()
```

DataFrame Operations

Command	Description	Example
`df.show()`	Display the DataFrame	`df.show(5)`
`df.printSchema()`	Print the schema of the DataFrame	`df.printSchema()`
`df.select()`	Select specific columns	`df.select("name", "age").show()`

Command	Description	Example
`df.filter()`	Filter rows based on conditions	`df.filter(df.age > 18).show()`
`df.withColumn()`	Add or modify a column	`df.withColumn("discount", df.price * 0.1)`
`df.drop()`	Drop a column	`df.drop("column_name")`
`df.distinct()`	Get distinct rows	`df.distinct().show()`
`df.sort()`	Sort DataFrame by columns	`df.sort(df["price"].desc()).show()`
`df.groupBy()`	Group rows and apply aggregations	`df.groupBy("region").sum("sales").show()`

DataFrame Joins

Join Type	Description	Example
`inner`	Inner join	`df1.join(df2, "id", "inner")`
`left`	Left join	`df1.join(df2, "id", "left")`
`right`	Right join	`df1.join(df2, "id", "right")`
`full`	Full outer join	`df1.join(df2, "id", "full")`

Data Transformation Commands

Command	Description	Example
`df.union()`	Combine two DataFrames	`df1.union(df2).show()`
`df.repartition()`	Repartition the DataFrame	`df.repartition(4)`

Command	Description	Example
df.cache()	Cache the DataFrame in memory	df.cache()
df.persist()	Persist the DataFrame to memory/disk	df.persist()
df.dropDuplicates()	Drop duplicate rows	df.dropDuplicates(["id"]).show()

SQL Queries in PySpark

Command	Description	Example
createOrReplaceTempView()	Create a temporary SQL view	df.createOrReplaceTempView("table_name")
spark.sql()	Run SQL queries on the temporary view	spark.sql("SELECT * FROM table_name").show()

PySpark Window Functions

Function	Description	Example
`row_number()`	Assigns a unique number to each row	`df.withColumn("row_num", row_number().over(windowSpec))`
`rank()`	Ranks rows based on specified criteria	`df.withColumn("rank", rank().over(windowSpec))`
`lead() / lag()`	Access next/previous row in a partition	`df.withColumn("next_val", lead("sales").over(windowSpec))`

RDD Operations (for Advanced Use Cases)

Command	Description	Example
`rdd.map()`	Apply a function to each element	`rdd.map(lambda x: x * 2).collect()`
`rdd.filter()`	Filter elements based on a condition	`rdd.filter(lambda x: x > 10).collect()`
`rdd.reduce()`	Apply a function to reduce elements	`rdd.reduce(lambda x, y: x + y)`
`rdd.collect()`	Return all elements as a list	`rdd.collect()`

Quick Tips for Optimizing PySpark Jobs

✅ Use **repartition()** for large datasets.

✅ Use **broadcast()** for small lookup tables.

✅ Cache DataFrames you reuse frequently.

✅ Avoid wide transformations like **groupBy()** unless necessary.

✅ Use **Delta Lake** for ACID transactions and time travel.

Abhishek Agrawal | Data Engineer

Follow for more content like this

Abhishek Agrawal

Azure Data Engineer