M to PySpark

Cheatsheet

Data Meerkat

Adam's Bubble

Task	Power Query	PySpark
Download data from Web / API	Web.Contents(<url>, optional <options>)	import requests requests.request(<method>, <url>,...)
Import data from json	Json.Document(<jsonText>,<encoding>)	spark.read.json(<path-to-json>)
Import data from csv	Csv.Document(<srouce>,<columns>, optional <delimiter>, optional <extraValues>, optional <encoding>)	spark.read.csv(<path>)
Create table / dataframe	#table(<columns>,<rows>)	spark.createDataFrame(<data>,<schema>)
Send SQL query against database	Value.NativeQuery(<target>,<query>, optional <parameters>, optional <options>)	spark.sql(<sql-text-query>)
Pivot data	Table.Pivot(<table>,<pivotValues>, <attributeColumns, <valueColumn>, optional <aggregationFunction>)	df.groupBy(<grouping_column>)\ .pivot(<pivot_column>)\ .agg({"agg_column": "agg_function"})
Un-pivot data	Table.Unpivot(<table>,<unpivotColumns>, <attributeColumn>,<valueColumn>)	df.unpivot(<stable-columns>, <unpivotColumns>, <attributeColumn>, <valueColumn>)

Task	Power Query	PySpark
Change Data Type of Column	Table.TransformColumnTypes(<table>, <typeTransforations>,optional <culture>)	df.withColumn(<name>,<column>.cast(<type>)) df.select(<column>.cast(<type>))
Remove Column(s)	Table.RemoveColumns(<table>,<columns>, optional <missingField>)	df.drop(<labels>,<axis>,<index>,<columns>)
Select Column(s)	Table.SelectColumns(<table>, <columns>, optional <missingField>)	df.select(<include>/<exclude>)
Add New Column	Table.AddColumn(<step>,<new-column-name>, <expression>, <type-of-column>)	df.withColumn(<name>, <value_provider>)
Transform Column(s)	Table.TransformColumns(<table>, <operations>, opional <defaultTransformation>, optional <missingField>)	df.withColumn(<name>, <value_provider>)
Transpose Table	Table.Transpose(<table>,optional <columns>)	df.transpose()
Joining Tables	Table.Join(<table1>,<key1>,<table2>,<key2>, optional <joinKind>, optional <joinAlgorithm>, optional <keyExqualityComparers>)	df.join(<right>,<on>,<joinKind>,<lsuffix>,<rsuffix>)

Task	Power Query	PySpark
Append Tables	<table> & <table> Table.Combine(<tables>,optional <columns>)	df.append(<other>)
Group By	Table.Group(<table>,<key>,<aggregatedColumns>, optional<groupKind>,optional <comparer>)	df.groupby(<by>,<axis>,<as_index>,<dropna>)
Extract Year from Date	Date.Year(<date>)	df.withColumn(<name>, year(<date>))
Extract Month from Date	Date.Month(<date>)	df.withColumn(<name>, month(<date>))
Extract Day from Date	Date.Day(<date>)	df.withColumn(<name>, dayofyear(<date>))
Date To Month Name	Date.MonthName(<date>,<encoding>)	df.withColumn(<name>, date_format(<date>, 'MMMM'))
Date To Text	Date.ToText(<date>,<format>,<encoding>)	df.withColumn(<name>, date_format(<date>, 'MM/dd/yyyy'))

Task	Power Query	PySpark
Date Difference	Duration.Days(<endDate> - <startDate>)	df.withColumn("datesDiff", datediff(<date1>,<date2>))
Conditions	if <condition> then <positive> else <negative>	when(<condition>, <positive>) .otherwise(<negative>)
Error Handling	try <expression> otherwise <expression>	try: <expression> except: <expression>
Filter Table	Table.SelectRows(<table>, <condition>)	df.filter(<items>,<like>,<regex>,<axis>)
Reorder Columns	Table.ReorderColumns(<table>,<columnOrder>, optional <missingField>)	df.select(<columns-in-specified-order>)
Table Rows Count	Table.RowCount(<table>)	df.count()
Remove Top N Rows	Table.Skip(<table>, optional <countOrCondition>)	df.tail(df.count()-<N to skip>(

Task	Power Query	PySpark
Sort Rows in Table	Table.Sort(<table>,<comparisonCriteria>)	df.sort(col("<column-name>").desc(), <more-column-sortings>)
Split Column by Delimiter	Table.SplitColumn(<table>, <sourceColumn>, Splitter.SplitTextByDelimiter(<delimiter>, QuoteStyle.Csv), <listOfNewColumns>)	df.withColumn("<new-column-name>", split("<source-column-name>", "<delimiter>").getItem(0))
Split Column by Position	Table.SplitColumn(<table>, <sourceColumn>, Splitter.SplitTextByPositions(<listOfPositions>), <listOfNewColumns>)	df.withColumn("<new-column-name>", substring("<source-column-name>", <start-position>, <count-of-characters>))
Replace values in Column	Table.Replace(<table>,<oldValue>,<newValue>, <replacer>,<columnsToSearch>)	df.withColumn("<new-column-name>",when(<predicate>, regexp_replace(<replaced-column>, "<old-value>", "<new-value>")))
Combine Multiple Arrays	List.Combine(<list-of-lists>)	flatten(<array-of-arrays>)
Expand Record	Table.ExpandListColumn(<table>,<column>)	df.select(complexField.value1, complexField.value2, ...)
Text Concatening in Column	<column1-or-value> & <column2-or-value>	concat(<column1-or-value>,<column2-or-value>)

Task	Power Query	PySpark
Fill Down **Fill Up**	Table.FillDown(<table>,<list-of-columns>) Table.FillUp(<table>,<list-of-columns>)	df.ffill() df.bfill()
Coalesce	<column1-or-value> ?? <column2-or-value>	coalesce(<column1>, <column2>, ...)
Statistical Info about Table	Table.Profile(<table>(	df.summary().show()
Remove Duplicate Rows	Table.Distinct(<table>)	df.distinct()
Table Column Rename	Table.RenameColumns(<table>, {{<column>, <new-name>},...})	df.select(<column>.alias(<new-name>))
List of Table Columns	Table.ColumnNames(<table>)	df.columns
Return Table Columns with Their Data Types	Table.Schema(<table>)	df.schema