SparkSQL Cheatsheet

Table of Contents

[Case Statements 1](#_Toc482788840)

[DataFrame (pyspark.sql.dataframe.DataFrame) 1](#_Toc482788841)

[Read a Parquet File into a DataFrame 1](#_Toc482788842)

[Store DataFrame as Temp Table for Later Querying 1](#_Toc482788843)

[Select an Array Element 2](#_Toc482788844)

[Parquet 2](#_Toc482788845)

[Explode a list or array data field 2](#_Toc482788846)

[Strings 3](#_Toc482788847)

[Concatenate strings 3](#_Toc482788848)

[Substring 3](#_Toc482788849)

[Timestamps 3](#_Toc482788850)

[Convert string to timestamp 3](#_Toc482788851)

# Case Statements

shipment\_modes\_df = sqlContext.sql(''' SELECT poiFromId, poiToId,

truckCount / (truckCount + intermodalCount + unknownCount) as truckPct

FROM

(SELECT poiFromId, poiToId,

SUM(CASE WHEN mode = 'truck' THEN 1 ELSE 0 END) AS truckCount,

SUM(CASE WHEN mode = 'intermodal' THEN 1 ELSE 0 END) AS intermodalCount,

SUM(CASE WHEN mode = 'unknown' THEN 1 ELSE 0 END) AS unknownCount

FROM

(SELECT shipmentId, poiFromId, poiToId,

explode(modes) as mode

FROM shipment\_sums\_table

WHERE legCount = 1)

GROUP BY poiFromId, poiToId

ORDER BY poiFromId, poiToId

)

order by truckPct

''')

# DataFrame (pyspark.sql.dataframe.DataFrame)

## Read a Parquet File into a DataFrame

shipment\_summaries\_path = '/lambda/summaries-parquet/PG/shipment-summaries/\*'

shipment\_sums = sqlContext.read.parquet(shipment\_summaries\_path) # pyspark.sql.dataframe.DataFrame

## Store DataFrame as Temp Table for Later Querying

shipment\_sums.createOrReplaceTempView("shipment\_sums\_table")

Query from the Temp Table:

test\_df = sqlContext.sql('''SELECT shipmentId FROM shipment\_sums\_table limit 1''')

test\_df.head()

Output:

Row(shipmentId=u'0306098015')

# Select an Array Element

df.printSchema()

## root

## |-- stuff: array (nullable = true)

## | |-- element: struct (containsNull = true)

## | | |-- a: long (nullable = true)

## | | |-- b: long (nullable = true)

## | | |-- c: long (nullable = true)

sqlContext.sql("SELECT stuff[0].a FROM df").show()

## +---+

## |\_c0|

## +---+

## | 1|

## +---+

# Parquet

## Explode a list or array data field

shipment\_df = sqlContext.sql('''SELECT shipmentId, modes

FROM shipment\_sums\_table

where legCount = 2

limit 5

''')

shipment\_df.show()

Output:

+----------+--------------+

|shipmentId| modes|

+----------+--------------+

|0306080481|[truck, truck]|

|0306120361|[truck, truck]|

|0306120411|[truck, truck]|

|0306103320|[truck, truck]|

|0306078031|[truck, truck]|

+----------+--------------+

# What modes are available?

modes\_df = sqlContext.sql('''SELECT shipmentId,

explode(distinct modes)as mode

FROM shipment\_sums\_table

WHERE legCount = 2

limit 6

''')

modes\_df.show()

Output:

+----------+-----+

|shipmentId| mode|

+----------+-----+

|0306080481|truck|

|0306080481|truck|

|0306120361|truck|

|0306120361|truck|

|0306120411|truck|

|0306120411|truck|

+----------+-----+

# Strings

## Concatenate strings

CONCAT(SUBSTR(smfPayloadData.timestamp.asOf,1,10), ' ',

SUBSTR(smfPayloadData.timestamp.asOf, 12, 8)) as test

## Substring

SUBSTR(smfPayloadData.timestamp.asOf,1,10)

output: “2016-03-01”

# Timestamps

## Convert string to timestamp

shipment\_carr = sqlContext.sql('''

SELECT smfPayloadData.paired.shipmentId,

smfPayloadData.paired.shipper,

smfPayloadData.paired.carrier,

smfPayloadData.timestamp.asOf as dest\_arr\_utc,

to\_utc\_timestamp('2016-03-01 00:00:00','UTC') as example\_timestamp

FROM mapped

where smfPayloadData.events.DeliveryLocationArrival is not null

and smfPayloadData.timestamp.asOf >= to\_utc\_timestamp('2016-03-01 00:00:00','UTC')

and smfPayloadData.timestamp.asOf < to\_utc\_timestamp('2016-03-02 00:00:00','UTC')

limit 1

''')

printResultsAsJson(shipment\_carr)

output:

{

**"example\_timestamp": "2016-03-01 00:00:00.0"**,

"carrier": "SCNN",

"shipper": "PG",

"dest\_arr\_utc": "2016-03-01T23:51:00.000Z",

"shipmentId": "Jackie will add"

}