PARQUET - Read From Parquet Files Using Materialized Views

Note: The following Table of Contents should exist in all documents

- Goal
- Solution
- References:

Goal

The goal of this document is to provide very simple way to read from an existing parquet file using Materialized views inside incorta.

In Example below, Materialized view is created to read the parquet file of SALES.SALES table provided by Sample Tenant Demo.

Solution

- 1. Ensure Spark is up & running and incorta is properly integrated with Spark.
- 2. Place the parquet file in shared folder or path, where it can be accessed by incorta
- 3. Login to incorta with user having permission to manage schemas (i.e. Super user, Schema Manager, or Analyzer)
- 4. Go to Schemas & Session Variables Page
- 5. Create new or Edit an Existing Schema
- 6. Add New Materialized view table as shown below

	O localhos							Distant	PR 04	D. Dista	ux 🖽 Songs & M	Lucia El COL	E DE LIVEOU	Tester?	DI Concettore	Dis Date	a film	* 1
incor		g_coad_issue	- neora				iearch	L Misc.		0 000	uk 🖂 songs a v				shboard Dia			٠
0	8.0	read_f	from_p	arqu	et			Last Lo Apr 17, 2	oad Sta 1018, 22:0			Loa 20s	ding Time	2	Rows 919K	× 	Schema V Table Alias	Wizard
8-0 Ⅲ ⊘	6	Table sales_mv_fr	rom_parqui	et		Lan	guage on				Colur 10			Joins 0	R c 9	200 000 119m	Join	zed View
0																		

6. Choose Language as Python. Then add code as shown below. Please note that

- parquetFile is a variable name that can be set to anything.
- Please provide correct full path of parquet files when using method: sqlContext.read.parquet or method: spark.read.parquet

Simple read sqlContext.read.parquet

```
# direct read the parquet content by defining a variable and adding the
parquet path. No extra processing is required
parquetFile = sqlContext.read.parquet("/Users/emanmoussa
/incorta_rel211_official/tenants/Demo/parquet/SALES/SALES/part-000000")
# Save the result
save(parquetFile)
```

Or:

Using sqlContext.read.parquet then processing result and saving it

```
#read the parquet content by defining a variable and adding the parquet
path
parquetFile = sqlContext.read.parquet("/Users/emanmoussa
/incorta_rel211_official/tenants/Demo/parquet/SALES/SALES/part-000000")
# Parquet files can also be registered as tables and then used in SQL
statements.
parquetFile.registerTempTable("parquetFile")
#Add your query. Below is a simple select from the parquet. You can
select specific fields if you need, or use specific functions
selectedSales= sqlContext.sql("SELECT * FROM parquetFile")
# Save the result
save(selectedSales)
```

Or:

Simple read using spark.read.parquet

```
#simple read using spark.read.parquet
df2 = spark.read.parquet('/home/incorta/IncortaAnalytics_3.1.7/tenants
/Demo/parquet/SALES/SALES/part-000000')
save(df2)
```

7. Optional: Add MV properties if needed, example is below

Inco			Data Source		×	+ Cancel Done				
0										
0	sales_mv_from_pa	rquet	Language:							
			Python							
80			incremental		0					
			Script:		8					
٢			<pre>parquetFile = sqlConte sa/incorta_rel211_offi /SALES/part-000000")</pre>	st.read.parquet(cial/tenants/Dem						
0	Name Label		parquetFile.registerTe			Source	Encrypted			
	QUANTITY,SOLD	Quantity Sole	10 1			Gibe				
	AMOUNT_SOLD Amount Sol		save(selectedSales)			Gibe				
	I IME,O	Time 1d				Gibe				
	CALENDAR, MONTH, NUMBER Calendar M		Properties:			take				
	CALENDAR, QUARTER, NUMBER	Calendar Qu	key: spark-executor-memory	value: 1g	8	Gibe				
	CALENDAR, MEAR	Calendar Yea	key: spark.executor.cores	value: 1	8	Gibe				
	COST_OF_60005	Cost Of Good				Gabre				
	Ø /%00,10	Prod 1d	keyc spark.cores.max	value: 1	8	Silve				
	CUSTOMER_ID Customer to			Add Property		Gibe				
	8 N	N				Gibe				
			Cancel	Delete	Save					

8. Click on Save on the Data Source pop-up then Done on Top right

Inco			Data Source		×	+ Cancel Done			
Ø									
8	sales_mv_from_pa	rquet	Language:		_				
			Python						
80			incremental						
			Script		8				
٢			<pre>parquetFile = sqlCostest. ss/incosts_rel211_officia /SALES/part-000000")</pre>						
0	S Name	Label	parquetFile.registerTempT	able("parquetFile")		Source Encrypted			
	QUANTITY, SOLD Quantity So		selectedSales- sqlContext le")	adilistret . they be	quetri				
	AMOUNT_SOLD	Amount Sold	save(selectedSales)						
	I IME,IO	Time 1d							
	CALENDAR, MONTH, NUMBER	Calendar Mo	Properties:						
	CALENDAR, QUARTER, NUMBER	Calendar Qu	key: spark-executor-memory	value: 1g	8				
	CALENDAR, YEAR	Calendar Yea	key: spark-everutor.cores	value: 1					
	COST_OF_60005	Cost Of Good							
	Ø /#00,10	Prod ki	keyc spark-cores.max	value: 1	8				
	CUSTOMERJD Customer ld		Add	Property	_				
	8 N	N							
			Cancel	Delete Save					

Note: Sample Schema is attached below. You will need to modify it before use as per steps 6-8.



References:

please refer to spark programming guide & Choose Spark as language: https://spark.apache.org/docs/1.6.1/sql-programming-guide.html