

## Running Additional Queries



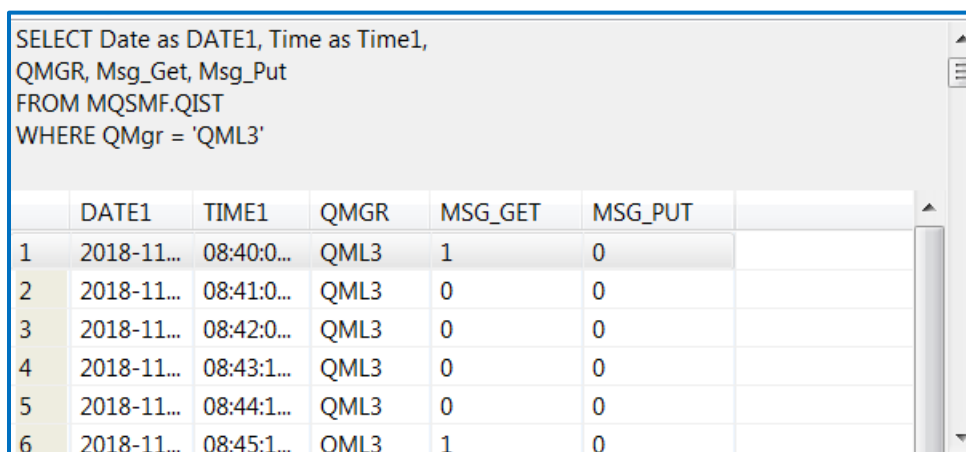
## Lab Objective

This lab takes you thru the steps to run some additional queries of the primary queries, using Data Studio.

## Lab Steps

- 1) Yesterday we heard about the MQ puts and gets total from the data manager – those are the puts and gets that are passed from the messages manager to the data manager, because they can potentially be successful. So the first query will be to retrieve that information from the data manager table and compare it with the message manager data.
- 2) In Data Studio, on the MQPERF1\_SQL tab, enter the following query:  

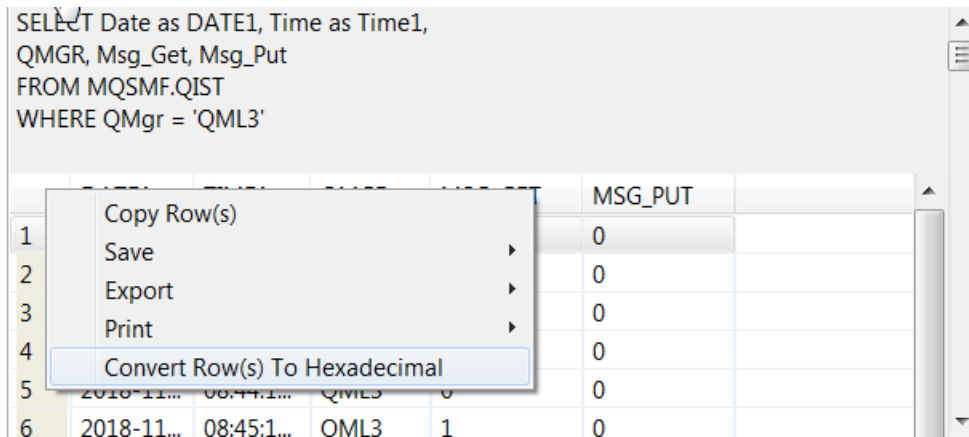
```
SELECT Date as DATE1, Time as Time1,  
       QMGR, Msg_Get, Msg_Put  
FROM MQSMF.QIST  
WHERE QMgr = 'QML3' ;
```
- 3) Run the query by hitting the green arrow.
- 4) The results pane should look something like this:



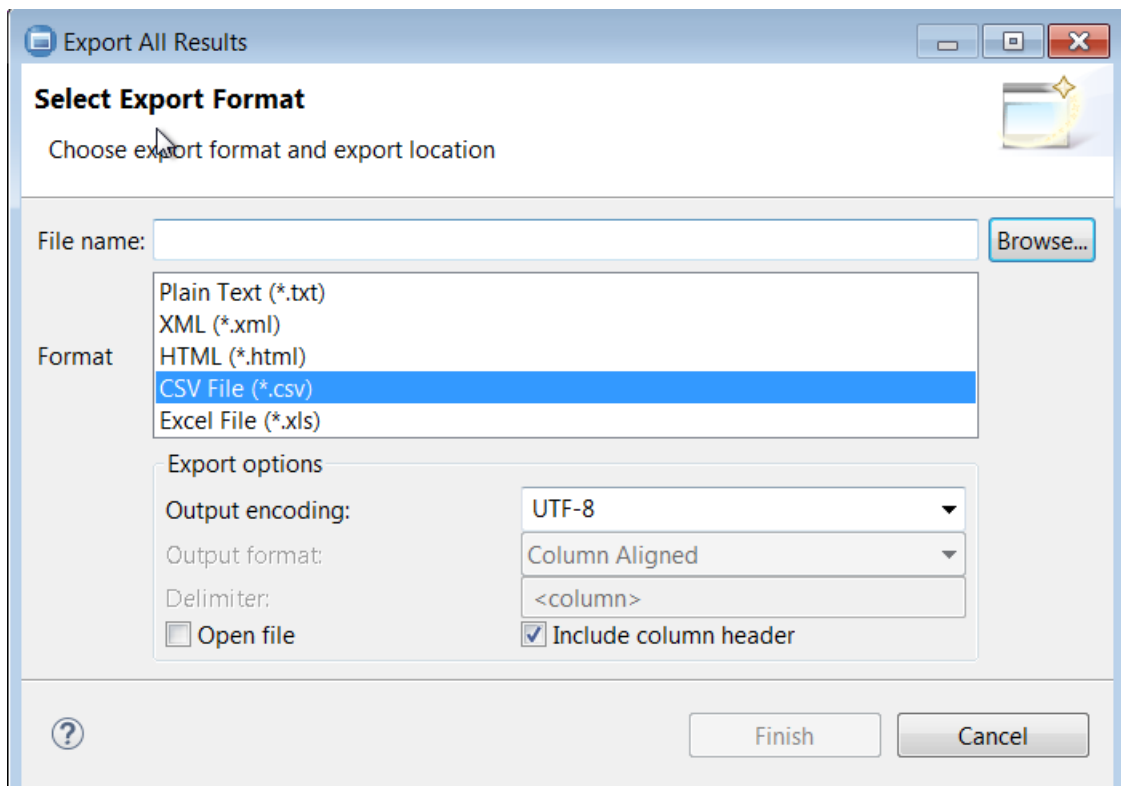
The screenshot shows a Data Studio interface with a SQL query editor at the top and a results pane below it. The query is: `SELECT Date as DATE1, Time as Time1, QMGR, Msg_Get, Msg_Put FROM MQSMF.QIST WHERE QMgr = 'QML3' ;`. The results pane displays a table with 6 rows and 6 columns: an index column, DATE1, TIME1, QMGR, MSG\_GET, and MSG\_PUT. The data shows two successful puts (rows 1 and 6) and four successful gets (rows 2, 3, 4, and 5).

	DATE1	TIME1	QMGR	MSG_GET	MSG_PUT
1	2018-11...	08:40:0...	QML3	1	0
2	2018-11...	08:41:0...	QML3	0	0
3	2018-11...	08:42:0...	QML3	0	0
4	2018-11...	08:43:1...	QML3	0	0
5	2018-11...	08:44:1...	QML3	0	0
6	2018-11...	08:45:1...	QML3	1	0

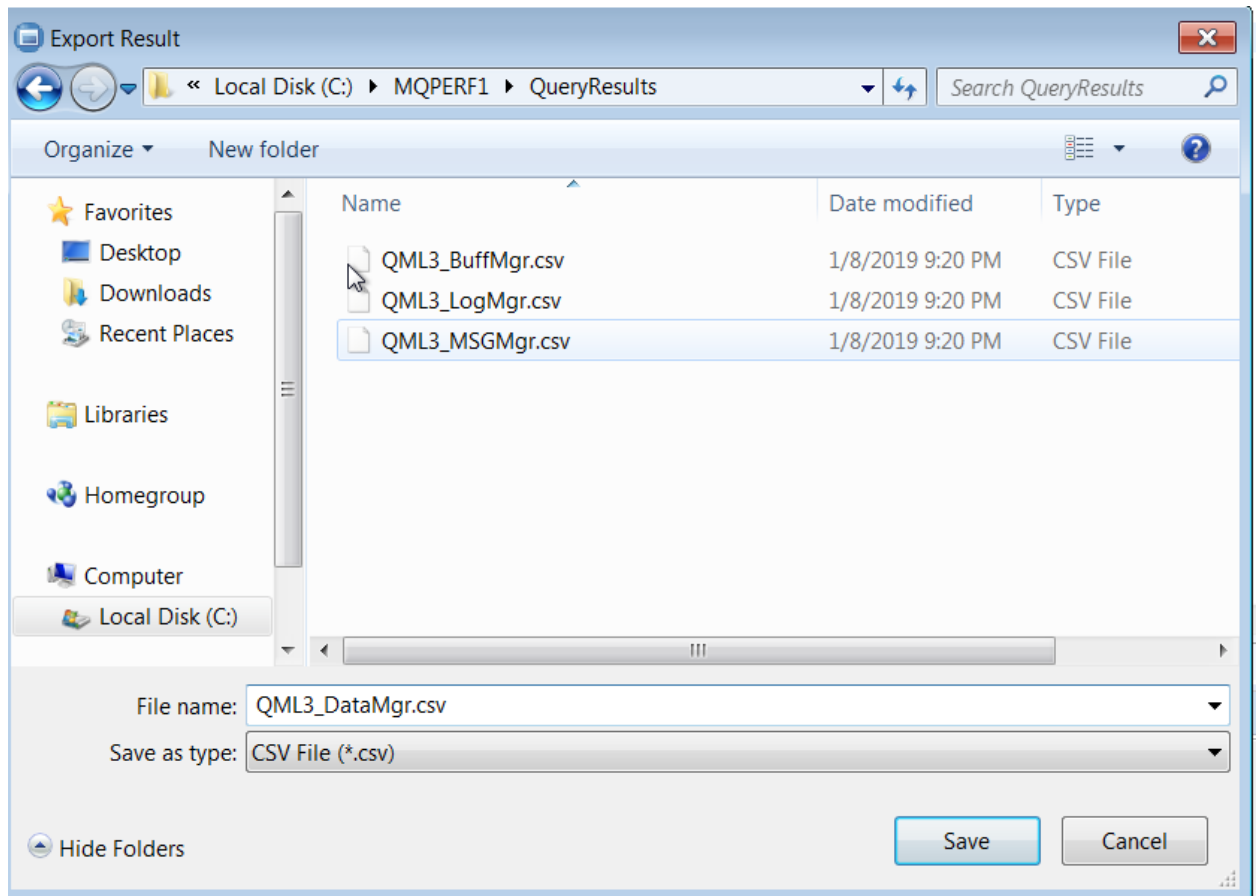
- 5) Because there should be a limited number of rows, less than the default of 500, the next step is to export the results set. In the rectangle to the left of the 'DATE1' label, right click and chose Export, then 'All Results (not shown as it could not be captured).



- 6) The 'Export All Results' pan should be shown, select CSV file and hit the Browse button.



- 7) Navigate to the MQPERF1/QueryResults directory and enter the file name of QML3\_DataMgr and hit the Save button.



- 8) That will return to the 'Export all Results' pane, from there hit the 'Finish' button.
- 9) In Open Office Calc, open the QML3\_DataMgr CSV file. Scrolling towards the bottom, you can see some put and get activity as shown:

	A	B	C	D	E
1	DATE1	TIME1	QMGR	MSG_GET	MSG_PUT
245	2018-11-13	13:06:03,600000	QML3	161	122
246	2018-11-13	13:06:57,660000	QML3	97	73
247	2018-11-13	13:08:05,230000	QML3	123	94
248	2018-11-13	13:09:06,050000	QML3	123	93
249	2018-11-13	13:10:06,870000	QML3	111	85
250	2018-11-13	13:11:07,680000	QML3	102	75
251	2018-11-13	13:11:42,880000	QML3	64	48
252	2018-11-13	13:13:09,320000	QML3	1640718	1640727
253	2018-11-13	13:14:10,140000	QML3	1279241	1279234
254	2018-11-13	13:15:10,950000	QML3	1434370	1434359
255	2018-11-13	13:16:11,770000	QML3	753461	753453
256	2018-11-13	13:17:12,590000	QML3	94	72
257	2018-11-13	13:18:13,410000	QML3	103	78
258	2018-11-13	13:19:14,220000	QML3	119	90
259	2018-11-13	13:20:15,040000	QML3	102	78

- 10) While we can manually compare these values with the equivalent Message Manager data, it would be easier to craft a query to associate the data from the two tables.
- 11) Returning to our query pane, a new query will be created to associate the values from the two tables. Note that the text for this query is in a text file in the Queries directory under the MQPERF1 directory:

```

SELECT DM.Date as DATE1, DM.Time as Time1,
DM.QMGR AS QMGR_1, DM.Msg_Get AS DataMGR_GET,
MM.Get as MsgMgr_Get,
DM.Msg_Put AS DataMgr_Put,
MM.Put AS MsgMgr_Put,
MM.Put1 AS MsgMgr_Put1
FROM MQSMF.QIST DM, MQSMF.QMST MM
WHERE (DM.Date = MM.Date AND DM.Time = MM.Time AND DM.QMGR = 'QML3' );

```

- 12) Run the query and export the results as illustrated above, call the exported file QML3\_DMandMM.
- 13)

14) Opening the file with Open Office shows the following:

QML3\_DMandMM.csv - OpenOffice Calc

	A	B	C	D	E	F	G	H
	DATE1	TIME1	QMGR_1	DATAMGR_GET	MSGMGR_GET	DATAMGR_PUT	MSGMGR_PUT	MSGMGR_PUT1
1	2018-11-13	13:16:11,770000	QML3	753461	753445	753453	753453	
2	2018-11-12	08:40:07,780000	QML3	1	17	0	0	
3	2018-11-12	08:41:08,600000	QML3	0	17	0	0	
4	2018-11-12	08:42:09,420000	QML3	0	17	0	0	
5	2018-11-12	08:43:10,230000	QML3	0	17	0	0	
6	2018-11-12	08:44:11,050000	QML3	0	17	0	0	
7	2018-11-12	08:45:11,870000	QML3	1	17	0	0	
8	2018-11-12	08:46:12,680000	QML3	0	18	0	0	
9	2018-11-12	08:47:13,500000	QML3	2	21	1	1	
10	2018-11-12	08:48:14,320000	QML3	0	17	0	0	
11	2018-11-12	08:49:15,140000	QML3	0	17	0	0	
12	2018-11-12	08:50:15,950000	QML3	1	17	0	0	
13	2018-11-12	08:51:16,770000	QML3	0	18	0	0	
14	2018-11-12	08:52:17,590000	QML3	0	17	0	0	
15	2018-11-12	08:53:18,410000	QML3	0	17	0	0	
16	2018-11-12	08:54:19,220000	QML3	0	17	0	0	
17	2018-11-12	08:54:19,220000	QML3	0	17	0	0	

Sheet 1 / 1      Default      STD      Sum=753445      100 %

- 15) It may be necessary to sort the data, but as you can see the number of MQPUTs for both sources in the 13:16 rows, while the number of message manager gets is slightly larger than the data manager. Because we know the tasks that were running, this slight difference is the normal get with wait processing that goes unfulfilled by the OEMPUT program.
- 16) On some rows, the number of MQPUTs and MQPUT1 from the message manager will need to be added to compare to the data manager put count, that is something that can be added to the query or done within the spreadsheet.
- 17) Congratulations, you are now an expert!