

Orders

```
... .trigger(processingTime='%s seconds' % freq) \
... .option("checkpointLocation", checkpoint_location + "/" + date) \
... .options(truncate=False) \
... .start()
...
>>> stream = console_output(parsed_orders, 10)
21/09/19 10:52:46 WARN shortcircuit.DomainSocketFactory: The short-circuit local reads feature cannot be used because libhadoop cannot be loaded.

-----
Batch: 0
-----
+-----+-----+-----+-----+-----+-----+
|order_id|customer_id|order_status|order_purchase_timestamp|order_approved_at|order_delivered_
carrier_date|order_delivered_customer_date|order_estimated_delivery_date|
+-----+-----+-----+-----+-----+-----+
|949d5b44dbf5de918fe9c16f97b45f8a|f88197465ea7920adcdbec7375364d82|delivered|2017-11-18 19:28:06|2017-11-18 19:45:59|2017-11-22 13:39
:59|2017-12-02 00:28:42|2017-12-15 00:00:00|
|83018ec114eee8641c97e08f7b4e926f|7f8c8b9c2ae27bf3300f670c3d478be8|delivered|2017-10-26 15:54:26|2017-10-26 16:08:14|2017-10-26 21:46
:53|2017-11-08 22:22:00|2017-11-23 00:00:00|
|95266dbfb7e20354baba07964dac78d5|a166da34890074091a942054b36e4265|delivered|2018-01-08 07:55:29|2018-01-08 08:07:31|2018-01-24 23:16
:37|2018-01-26 17:32:38|2018-02-21 00:00:00|
|f3e7c359154d965827355f39d6b1fdac|62b423aab58096ca514ba6aa06be2f98|delivered|2018-08-09 11:44:40|2018-08-10 03:24:51|2018-08-10 12:29
```

customer names

read from cassandra

```
+-----+-----+
|cid|full_name|
+-----+-----+
|55490de47a9c2abff0a27a5296ae9c2c|Terrance Trump|
|ffdc2af7160aed3bad21cbc8ec68c0c5|Philip Clegg|
|d4faa220408c20e53595d2950f361f3b|May Pollock|
|59cf1a479d104e38587b9fe9d60febe3|Michael Dillon|
|db925a533d94cee42707026949892c7e|Richard Beere|
|48c83ab1910a8d95f8702dde180d658b|Martina Moore|
|6eae55140163e2af86907a15a478f2d2|Caroline Brown|
|4fd75fb5ef1f01c7585bf746092a1544|Andrew Booth|
|2c5491a76ec0a56db58282ea7ae70617|David Matthews|
|4d833833ec5cc5bb9e6c230820e4bc54|Tracey Groves|
|fac60d7c4df3896a5af5db6c91e03797|Jane Gill|
|aa3d8c9e8e8a5e90b117efe5e97a560|Lachlan McLaren|
|686ca7499141a82f95123c370af061b0|Ibrar Majid|
|2b47ff70d422cc3ada976dea40f48782|David Francis|
|5072cf2f4cbec30b8ba917d5d7b6b125|Michal Buras|
|20b5aae6a3e31111009f9a7ecc31a232|Ann Peterson 2|
|bf141bf67f7be428d558bcf0e018eab60|Michael McGrane|
|e40265c6eebf38ea830b695d9340bda|Sharon Lambe|
|6c347ef65dd574fb9c2d654c3650fc43|Archie MacGregor|
|7e016f9ea275279784f42e0642214284|Maurice Savage|
```

Static join

```
+-----+-----+-----+
|oid|cid|full_name|
+-----+-----+-----+
|e481f51cbd5c54678b7cc49136f2d6af7|9ef432eb6251297304e76186b10a928d|Kevin Maguire|
|76c6e866289321a7c93b82b54852dc33|f54a9f0e6b351c431402b8461ea51999|George Rae|
|5ff96c15d0b717ac6ad1f3d77225a350|19402a48fe860416adf93348aba37740|Wanda Woods|
+-----+-----+-----+
```

write joined df to Cassandra as Stream by Batch

```
>>> names_df = spark.sql("select 20b5aae6a3e31111009f9a7ecc31a232 as cid, Ann Peterson 2 as full_name")
>>> names_df.show()
+-----+-----+
|cid|full_name|
+-----+-----+
|20b5aae6a3e31111009f9a7ecc31a232|Ann Peterson 2|
+-----+-----+
```

```
# read all names
```

```
>>> all_names_df.show()
+-----+-----+
|          cid|    full_name|
+-----+-----+
|55490de47a9c2abff...| Terrance Trump|
|ffdc2af7160aed3ba...|  Philip Clegg|
|d4faa220408c20e53...|   May Pollock|
|59cf1a479d104e385...| Michael Dillon|
|db925a533d94cee42...| Richard Beere|
|48c83ab1910a8d95f...|  Martina Moore|
|6eae55140163e2af8...| Caroline Brown|
|4fd75fb5ef1f01c75...|  Andrew Booth|
|2c5491a76ec0a56db...| David Matthews|
|4d833833ec5cc5bb9...| Tracey Groves|
|fac60d7c4df3896a5...|   Jane Gill|
|aa3d8c9eec8e52e90...| Lachlan McLaren|
|686ca7499141a82f9...|  Ibrar Majid|
|2b47ff70d422cc3ad...| David Francis|
|5072cf2f4cbec30b8...|  Michal Buras|
|20b5aae6a3e311110...| Ann Peterson 2|
|bf141bf67fbe428d5...| Michael McGrane|
|e40265c6ceebf38ea...| Sharon Lambe|
|6c347ef65dd574fb9...| Archie MacGregor|
|7e016f9ea27527978...| Maurice Savage|
```

```
# filter by cid
```

```
>>> c_name_df.show()
+-----+-----+
|          cid|    full_name|
+-----+-----+
|20b5aae6a3e311110...| Ann Peterson 2|
+-----+-----+
```

```
# filter by non-partition key
```

```
>>> jane_df.show() # only first 20
+---+-----+
|cid|full_name|
+---+-----+
+---+-----+

>>> jane_df.count() # all records
```

between doesn't work with PushDown filter

cid	full_name
55490de47a9c2abff...	Terrance Trump
59cf1a479d104e385...	Michael Dillon
48c83ab1910a8d95f...	Martina Moore
6eae55140163e2af8...	Caroline Brown
4fd75fb5ef1f01c75...	Andrew Booth
2c5491a76ec0a56db...	David Matthews
4d833833ec5cc5bb9...	Tracey Groves
aa3d8c9eec8e52e90...	Lachlan McLaren
686ca7499141a82f9...	Ibrar Majid
2b47ff70d422cc3ad...	David Francis
5072cf2f4cbec30b8...	Michal Buras
20b5aae6a3e311110...	Ann Peterson 2
6c347ef65dd574fb9...	Archie MacGregor
7e016f9ea27527978...	Maurice Savage
ae2164e850f39dce4...	Mohammed Hoque

in works with PushDown filter

```
0717d7cc51a2127b0701d4a5ca0004e707043007030c177, readschema: strictRelatString,full_nameString/
>>> in_select.show()
+-----+-----+
|      cid|    full_name|
+-----+-----+
|20b5aae6a3e311110...|Ann Peterson 2|
|b89010d4a6acaa06d...| Stephen Wood|
+-----+-----+

>>> in_select.count()
21/09/19 11:49:15 WARN core.RequestHandler: Query '[2 bound values] SELECT count(*) FROM "streaming_student782_3"."customer_names" WHERE "cid"
" IN (?, ?)  ALLOW FILTERING;' generated server side warning(s): Aggregation query used on multiple partition keys (IN restriction)
2
>>>
>>>
```