

Otto Friedrich University Bamberg



Data Streams and Complex Event Processing

Assignment - 02

Supervisor:

Prof. Dr. Daniela Nicklas
Mobile Software Systems

Submitted By :

Manasa Reddy Bandari - 1942692
Akshay Sharma - 1943984
Kummari Daniel Raj - 1973448

Nov 29, 2018

Contents

1	Exercise 1 :	2
1.1	Reports descriptions of auctions made in the last 10 minutes by luitpold Martucci	2
1.2	Find all items in the auctions stream that have an initial bid of more than 120 dollars.	2
1.3	Return the number of bids received for each auction.	2
1.4	Find auctions which had bids 2 minutes before their expiry date.	2
1.5	Return the number of bids received for each auction coming from each state.	3
2	Exercise 2 :	3
2.1	Do not assume any heartbeats or punctuations.	3
2.1.1	09:45	3
2.1.2	10:55	3
2.1.3	11:41	4
2.2	What would change if there is a heartbeat every second? Please fill the template again.	4
2.2.1	09:45	4
2.2.2	10:55	4
2.2.3	11:41	5
3	Exercise 3:	5
3.1	Why do events require their occurrence time to be included? In which cases is it important? Provide an example.	5
3.2	Why do some events require spatial properties? (When is that used, provide examples).	5
3.3	A data stream can contain punctuations. Define what punctuations are, and give an example of how they can be used.	5

1 Exercise 1 :

Based on the dataset generated by Nexmark, write the following queries on Odysseus using the PQL language (The same queries of the exercise 3 in the first sheet).

1.1 Reports descriptions of auctions made in the last 10 minutes by luitpold Martucci

```
1 ap_join = JOIN ({PREDICATE = 'auction.seller = person.id'},
    auction, person)
2 join_time = TIMEWINDOW ({SIZE = [10, 'minutes']}, ap_join)
3 person_select = SELECT ({PREDICATE = 'name == "Luitpold
    Martucci"'}, join_time)
4 final_output = PROJECT ({attributes = ['name', 'itemname', '
    description']}, person_select)
```

1.2 Find all items in the auctions stream that have an initial bid of more than 120 dollars.

```
1 item_select = SELECT ({PREDICATE = 'initialbid >= 120'},
    auction)
2 final_output = PROJECT ({attributes = ['id', 'itemname', '
    initialbid']}, item_select)
```

1.3 Return the number of bids received for each auction.

```
1 time_window = TIMEWINDOW ({SIZE = [5, 'minutes']}, bid)
2 final_output = AGGREGATE ({
3 group_by = [ 'auction'],
4 aggregations = [['count', 'bidder', 'bid_numbers', 'integer'
    ]]], time_window)
```

1.4 Find auctions which had bids 2 minutes before their expiry date.

```
1 ab_join = JOIN ({PREDICATE = 'auction.id = bid.auction'},
    auction, bid)
2
3 final_output = SELECT ({PREDICATE = 'auction.expires-bid.
    datetime <= 120000'}, ab_join)
```

1.5 Return the number of bids received for each auction coming from each state.

```

1 bp_join = join ({PREDICATE = 'person.id = bid.bidder'}, bid,
2               person)
3 join_time = TIMEWINDOW ({SIZE = [5, 'minutes']}, bp_join)
4
5 final_output = AGGREGATE ({group_by = ['state', 'auction'],
6                             aggregations = [['count', 'bidder', 'bid_numbers', 'Integer'
7                                             ]]], join_time)

```

2 Exercise 2 :

We have two incoming streams from the library example used in the first exercise. We want to find the books that were checked in and checked out within half an hour. We present here a sample of elements from both streams. Using the ripple join try and compute the expected results and the status of sweep areas (buffers) for both streams at the times 09:45, 10:55 and 11:41.

2.1 Do not assume any heartbeats or punctuations.

2.1.1 09:45

Sweep Area 1

in_006	Inf_121_003	[09:35-10:05]
in_007	Aut_175_101	[09:37-10:07]
in_008	Inf_113_021	[09:42-10:12]

2.1.2 10:55

Sweep Area 1

in_011	Siv_014_035	[10:33-11:03]
in_012	Wirt_025_021	[10:43-11:13]
in_013	Med_147_102	[10:50-11:20]

Sweep Area 2

out_027	Aut_175_101	[10:21-10:51]
out_028	Mab_024_014	[10:31-11:01]
out_029	Inf_001_174	[10:54-11:24]

2.1.3 11:41

Sweep Area 1 :

in_012	Wirt_025_021	[10:43-11:13]
in_013	Med_147_102	[10:50-11:20]
in_014	Aut_401_201	[11:02-11:32]
in_015	Arc_104_715	[11:05-11:35]

Sweep Area 2:

out_029	Inf_001_174	[10:54-11:24]
out_030	Wirt_025_021	[11:11-11:41]

Output Buffer :

in_012	out_030	Wirt_025_021	[11:11 - 11:13]
--------	---------	--------------	-----------------

2.2 What would change if there is a heartbeat every second? Please fill the template again.

2.2.1 09:45

Sweep Area 1:

in_006	Inf_121_003	[09:35-10:05]
in_007	Aut_175_101	[09:37-10:07]
in_008	Inf_113_021	[09:42-10:12]

2.2.2 10:55

Sweep Area 1:

in_011	Siv_014_035	[10:33-11:03]
in_012	Wirt_025_021	[10:43-11:13]
in_013	Med_147_102	[10:50-11:20]

Sweep Area 2:

out_028	Mab_024_014	[10:31-11:01]
out_029	Inf_001_174	[10:54-11:24]

2.2.3 11:41

Sweep Area 2:

out_030	Wirt_025_021	[11:11-11:41]
---------	--------------	---------------

3 Exercise 3:

Answer the following questions:

3.1 Why do events require their occurrence time to be included? In which cases is it important? Provide an example.

For efficient orchestration of events , we do require their occurrence time. We can take Pizza Delivery boy tracking system in which his location and time needs to be present in order to calculate the estimated delivery time.

3.2 Why do some events require spatial properties? (When is that used, provide examples).

Some events do require spatial properties as they are essential for specific scenarios and situations. We can take an example of Google Maps for this one.

3.3 A data stream can contain punctuations. Define what punctuations are, and give an example of how they can be used.

Punctuation is being used to unblock the operator in case if some item is not required then it can send it as blank making the stream alive. Like in a navigation system , we can restrict some intervals in which location is not getting into the stream.