Report - template

Assignment 3 - MongoDB

Group: 79
Students: Nirushaan Selvaratnam

Introduction

The task is to upload data and answer some questions using queries from a modified dataset based on Geolife GPS Trajectory dataset. I have hosted a MongoDB server using Docker and the Python class DbConnector which was provided with the task material. Most of the questions were answered purely using MongoDB queries, while some required some additional coding. The result of my work is provided below as pictures of the results I got.

Results

Part 1

User collection

```
{'_id': '135',
 'activities': [ObjectId('67079c558d3f5b99604ebe63'),
                ObjectId('67079c558d3f5b99604ebe64'),
                ObjectId('67079c558d3f5b99604ebe65'),
                ObjectId('67079c558d3f5b99604ebe66'),
                ObjectId('67079c558d3f5b99604ebe67'),
                ObjectId('67079c558d3f5b99604ebe68'),
                ObjectId('67079c558d3f5b99604ebe69'),
                ObjectId('67079c558d3f5b99604ebe6a')],
 'has_labels': False}

{'_id': '132',
 'activities': [ObjectId('67079c558d3f5b99604ebe6b'),
                ObjectId('67079c558d3f5b99604ebe6c'),
                ObjectId('67079c558d3f5b99604ebe6d')],
 'has_labels': False}
```

Here are some of the documents from my user collection. I only included 2 documents as this is sufficient to understand the layout I chose for that specific collection.

Activity collection

```
{'_id': ObjectId('67079c558d3f5b99604ebe63'),
 'end_date_time': datetime.datetime(2009, 1, 3, 5, 40, 31),
 'start_date_time': datetime.datetime(2009, 1, 3, 1, 21, 34),
 'transportation_mode': None,
 'user_id': '135'}

{'_id': ObjectId('67079c558d3f5b99604ebe64'),
 'end_date_time': datetime.datetime(2009, 1, 2, 4, 41, 5),
 'start_date_time': datetime.datetime(2009, 1, 2, 4, 31, 27),
 'transportation_mode': None,
 'user_id': '135'}

{'_id': ObjectId('67079c558d3f5b99604ebe65'),
 'end_date_time': datetime.datetime(2009, 1, 27, 4, 50, 32),
 'start_date_time': datetime.datetime(2009, 1, 27, 3, 0, 4),
 'transportation_mode': None,
 'user_id': '135'}

{'_id': ObjectId('67079c558d3f5b99604ebe66'),
 'end_date_time': datetime.datetime(2009, 1, 10, 4, 42, 47),
 'start_date_time': datetime.datetime(2009, 1, 10, 1, 19, 47),
 'transportation_mode': None,
 'user_id': '135'}

{'_id': ObjectId('67079c558d3f5b99604ebe67'),
 'end_date_time': datetime.datetime(2009, 1, 14, 12, 30, 53),
 'start_date_time': datetime.datetime(2009, 1, 14, 12, 17, 57),
 'transportation_mode': None,
 'user_id': '135'}
```

These are some documents from my activity collection. I chose to only include exact matches on start time and end time for labels.

TrackPoint collection

```
{'_id': ObjectId('67090a8619c619b15321daa0'),
 'activity_id': ObjectId('670906f7db8a688baa9b6795'),
 'altitude': 492.0,
 'coordinates': [39.974294, 116.399741],
 'date_time': datetime.datetime(2009, 1, 3, 1, 21, 34),
 'user_id': '135'}

{'_id': ObjectId('67090a8619c619b15321daa1'),
 'activity_id': ObjectId('670906f7db8a688baa9b6795'),
 'altitude': 492.0,
 'coordinates': [39.974292, 116.399592],
 'date_time': datetime.datetime(2009, 1, 3, 1, 21, 35),
 'user_id': '135'}
```

These are the first 2 documents in my TrackPoint collection. In the coordinates field the first value is latitude, and the second value is longitude.

Part 2

Task 1

```
User Count: 182
Activity Count: 16048
Trackpoint Count: 9681756
```

After inserting the data, I have 182 users, 16048 activities, and 9681756 TrackPoint's.

Task 2

Query result:

```
Average number of activities per user: 88.17582417582418
```

Task 3

Query result:

```
{'_id': '128', 'numActivities': 2102}
{'_id': '153', 'numActivities': 1793}
{'_id': '025', 'numActivities': 715}
{'_id': '163', 'numActivities': 704}
{'_id': '062', 'numActivities': 691}
{'_id': '144', 'numActivities': 563}
{'_id': '041', 'numActivities': 399}
{'_id': '085', 'numActivities': 364}
{'_id': '004', 'numActivities': 346}
{'_id': '140', 'numActivities': 345}
{'_id': '167', 'numActivities': 320}
{'_id': '068', 'numActivities': 280}
{'_id': '017', 'numActivities': 265}
{'_id': '003', 'numActivities': 261}
{'_id': '014', 'numActivities': 236}
{'_id': '126', 'numActivities': 215}
{'_id': '030', 'numActivities': 210}
{'_id': '112', 'numActivities': 208}
{'_id': '011', 'numActivities': 201}
{'_id': '039', 'numActivities': 198}
```

These are the top 20 users with the highest number of activities.

Task 4

Query result:

```
'010'
'058'
'062'
'078'
'080'
'085'
'098'
'111'
'128'
'163'
```

These are all the IDs for all the users that have at least one activity labeled with taxi.

Task 5

Query result:

```
{'mode_of_transport': 'run', 'total_activities': 1}
{'mode_of_transport': 'bike', 'total_activities': 262}
{'mode_of_transport': 'walk', 'total_activities': 481}
{'mode_of_transport': 'subway', 'total_activities': 133}
{'mode_of_transport': 'taxi', 'total_activities': 37}
{'mode_of_transport': 'bus', 'total_activities': 199}
{'mode_of_transport': 'airplane', 'total_activities': 3}
{'mode_of_transport': 'train', 'total_activities': 2}
{'mode_of_transport': 'boat', 'total_activities': 1}
{'mode_of_transport': 'car', 'total_activities': 419}
```

Here are all the transportation modes and the number of activities that were registered with the given transportation mode.

Task 6a

Query result:

```
{'_id': 2008, 'count': 5895}
{'_id': 2009, 'count': 5880}
{'_id': 2010, 'count': 1487}
{'_id': 2011, 'count': 1204}
{'_id': 2007, 'count': 994}
{'_id': 2012, 'count': 588}
{'_id': 2000, 'count': 1}
```

The year with the highest number of activities is 2008. My query takes into consideration if the start_date and end_date stretch from one year to another. If that is the case for an activity, it would be counted for both years.

Task 6b

Query result:

```
[(2009, 11612.629166666698),
 (2008, 9200.591666666674),
 (2007, 2315.418611111113),
 (2010, 1388.7275000000002),
 (2011, 1132.3516666666653),
 (2012, 711.2133333333336),
 (2000, 0.05111111111111114)]
```

These are the years and the recorded hours for each year. My query and code consider instances where activities are recorded over a year change. The year with the most recorded activities is not the year with the most recorded hours. 2009 has more hours recorded than 2008.

Task 7

Query result:

```
Total distance walked in 2008 by user 112: 115.47465961508004 km
```

Task 8

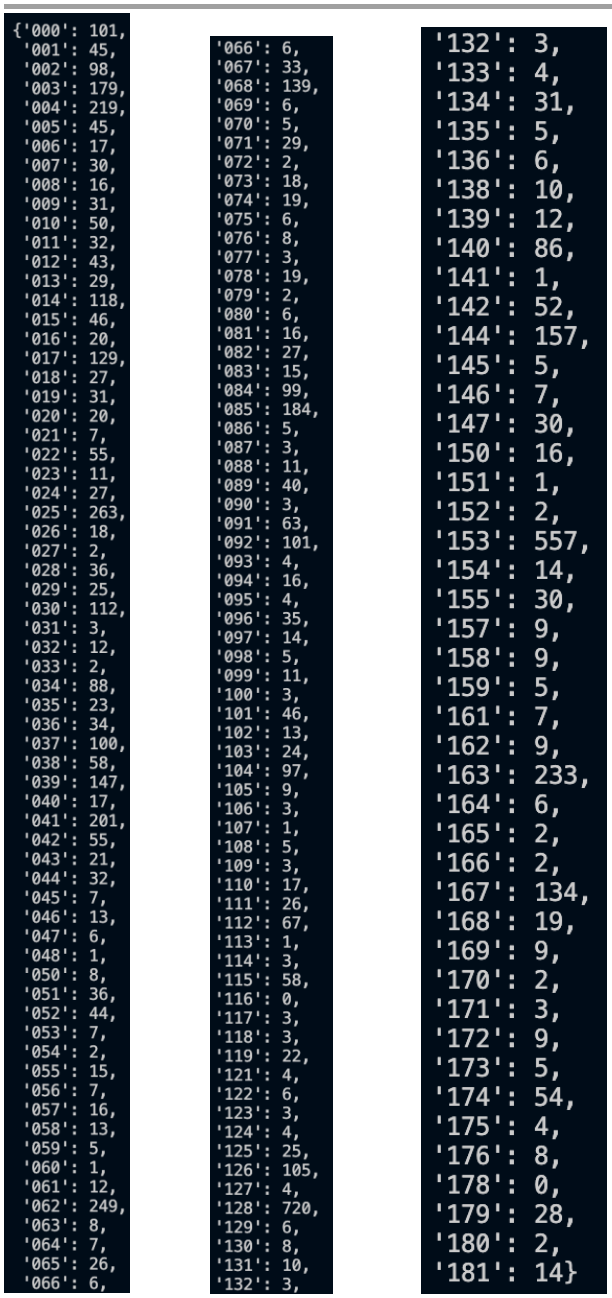
Query result:

```
[('128', 2135669.282417741),
 ('153', 1820736.9522737002),
 ('004', 1089358.0),
 ('041', 789924.1000003539),
 ('003', 766613.0),
 ('085', 714053.1000000071),
 ('163', 673472.3440420027),
 ('062', 596106.5999999233),
 ('144', 588718.9123359431),
 ('030', 576377.0),
 ('039', 481311.0),
 ('084', 430319.0),
 ('000', 398638.0),
 ('002', 377503.0),
 ('167', 370650.1136482952),
 ('025', 358131.7999999046),
 ('037', 325572.79999995086),
 ('140', 311175.52283458825),
 ('126', 272394.47427820024),
 ('017', 205319.39999998698)]
```

These are the top 20 users with the most altitude gained. I only considered the value of −777 to be invalid. It takes two consecutive TrackPoint's and check if the previous altitude is less than the current altitude and adds it to the altitude gain for the user.

Task 9

Query result:



These are all the users with invalid activities. An activity is invalid if two consecutive TrackPoint's were more than or equal to 5 minutes apart. The first value is the user_id and the second value is the number of invalid activities.

Task 10

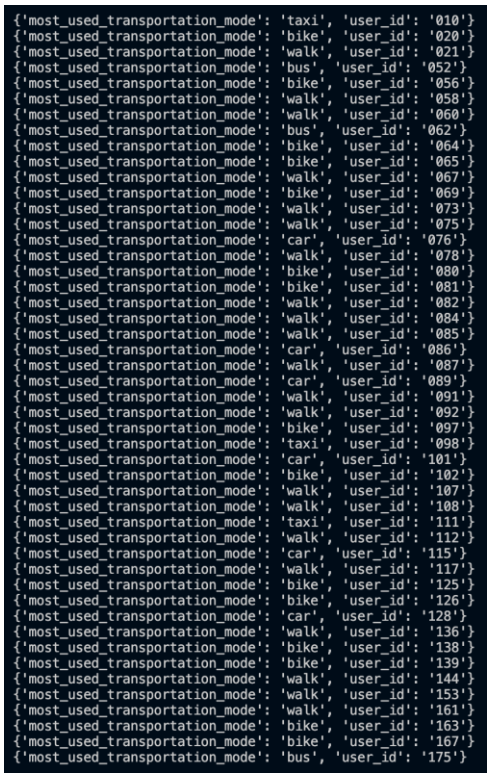
Query result:

```
{'_id': ['019']}
{'_id': ['018']}
```

These are the users with TrackPoint's registered within the forbidden city of Beijing. I did an exact match on where the latitude is 39,916 and the longitude is 116,397 where any decimal number after the given number can be whatever.

Task 11

Query result:



```
{'most_used_transportation_mode': 'taxi', 'user_id': '010'}
{'most_used_transportation_mode': 'bike', 'user_id': '020'}
{'most_used_transportation_mode': 'walk', 'user_id': '021'}
{'most_used_transportation_mode': 'bus', 'user_id': '052'}
{'most_used_transportation_mode': 'bike', 'user_id': '056'}
{'most_used_transportation_mode': 'walk', 'user_id': '058'}
{'most_used_transportation_mode': 'walk', 'user_id': '060'}
{'most_used_transportation_mode': 'bus', 'user_id': '062'}
{'most_used_transportation_mode': 'bike', 'user_id': '064'}
{'most_used_transportation_mode': 'bike', 'user_id': '065'}
{'most_used_transportation_mode': 'walk', 'user_id': '067'}
{'most_used_transportation_mode': 'bike', 'user_id': '069'}
{'most_used_transportation_mode': 'walk', 'user_id': '073'}
{'most_used_transportation_mode': 'walk', 'user_id': '075'}
{'most_used_transportation_mode': 'car', 'user_id': '076'}
{'most_used_transportation_mode': 'walk', 'user_id': '078'}
{'most_used_transportation_mode': 'bike', 'user_id': '080'}
{'most_used_transportation_mode': 'bike', 'user_id': '081'}
{'most_used_transportation_mode': 'walk', 'user_id': '082'}
{'most_used_transportation_mode': 'walk', 'user_id': '084'}
{'most_used_transportation_mode': 'walk', 'user_id': '085'}
{'most_used_transportation_mode': 'car', 'user_id': '086'}
{'most_used_transportation_mode': 'walk', 'user_id': '087'}
{'most_used_transportation_mode': 'car', 'user_id': '089'}
{'most_used_transportation_mode': 'walk', 'user_id': '091'}
{'most_used_transportation_mode': 'walk', 'user_id': '092'}
{'most_used_transportation_mode': 'bike', 'user_id': '097'}
{'most_used_transportation_mode': 'taxi', 'user_id': '098'}
{'most_used_transportation_mode': 'car', 'user_id': '101'}
{'most_used_transportation_mode': 'bike', 'user_id': '102'}
{'most_used_transportation_mode': 'walk', 'user_id': '107'}
{'most_used_transportation_mode': 'walk', 'user_id': '108'}
{'most_used_transportation_mode': 'taxi', 'user_id': '111'}
{'most_used_transportation_mode': 'walk', 'user_id': '112'}
{'most_used_transportation_mode': 'car', 'user_id': '115'}
{'most_used_transportation_mode': 'walk', 'user_id': '117'}
{'most_used_transportation_mode': 'bike', 'user_id': '125'}
{'most_used_transportation_mode': 'bike', 'user_id': '126'}
{'most_used_transportation_mode': 'car', 'user_id': '128'}
{'most_used_transportation_mode': 'walk', 'user_id': '136'}
{'most_used_transportation_mode': 'bike', 'user_id': '138'}
{'most_used_transportation_mode': 'bike', 'user_id': '139'}
{'most_used_transportation_mode': 'walk', 'user_id': '144'}
{'most_used_transportation_mode': 'walk', 'user_id': '153'}
{'most_used_transportation_mode': 'walk', 'user_id': '161'}
{'most_used_transportation_mode': 'bike', 'user_id': '163'}
{'most_used_transportation_mode': 'bike', 'user_id': '167'}
{'most_used_transportation_mode': 'bus', 'user_id': '175'}
```

These are all the users with registered transportation mode other than None, along with their most used transportation mode.

Discussion

If you look at my collections, I tried to have all the fields needed in one collection as doing join on NOSQL databases is very time-consuming. That is the reason for having additional user_id- and activity_id-field in the TrackPoint collection.

There were fewer pain points in this exercise as I already had the recipe for answering all the queries from the previous exercise. I simply just had to translate the tasks to using my new database layout and using MongoDB syntax. The only pain point was choosing between the multiple ways I could format my collections. I started by not including the user_id in the TrackPoint field, but after running a join query I decided to do it because of how long the queries took.

From my previous courses I knew that joins were faster in a SQL database compared to a NOSQL database but seeing it for myself after coding was a valuable experience.