Report - template

Assignment 2 - MySQL

Group: 51
Students: Henrik Borge, Torbjørn Grande, Sondre Rogde

Introduction

Briefly explain the task and the problems you have solved. How did you work as a group? If you used Git, a link to the repository would be nice.

The task was to implement a structure for storing data on activities. Each activity was related to a user and potentially multiple trackpoints. The trackpoints contains a timestamp with the coordinates of the user's position along with altitude. This was done keeping in mind that we were storing data for an application similar to Strava (more on this under Discussion).

All team members knew each other well from before, so working as a group posed no problems. Being a team of three people really helped in discussing the technical details of how to store the data and what we had to consider. For most problems in part 2, we implemented a simple solution assuming the data was clean and made sure that solution worked. After implementing this simple solution, we expanded on it to deal with edge cases, invalid data and other things that could invalidate the output. This is discussed in greater detail under Results and Discussion.

We used Github for code collaboration and version control. The repository on Github can be found here: https://github.com/Sondringsen/StoreDistribuerteOvinger. The repository should be publicly available, but please let us know if there is something wrong with the access. The repository also contains a README.md containing all documentation required for running the code.

Results

Add your results from the tasks, both as text and screenshots. Short sentences are sufficient.

The following is small excerpts from the tables. We decided to implement the database in a slightly different way than the suggested structure to make the application a bit more flexible. This will be further discussed under Discussion.

First 10 entries of User

id	has_labels
000	0
001	0
002	0
003	0
004	0
005	0
006	0
007	0
008	0
009	0
010	1

First 10 entries of Activity

id	user_id	walk	bike	bus	taxi	car	subway	train	airplane	boat	run	motorcycle	valid_activity	start_date_time	end_date_time
1	000	0	0	0	0	0	0	0	0	0	0	0	1	2009-04-12 07:33:03	2009-04-12 17:24:59
2	000	0	0	0	0	0	0	0	0	0	0	0	1	2009-05-10 02:02:06	2009-05-10 06:20:11
3	000	0	0	0	0	0	0	0	0	0	0	0	1	2008-12-01 11:18:27	2008-12-01 11:35:15
4	000	0	0	0	0	0	0	0	0	0	0	0	1	2009-06-21 10:14:07	2009-06-21 14:47:12
5	000	0	0	0	0	0	0	0	0	0	0	0	1	2008-12-11 12:14:32	2008-12-11 12:24:32
6	000	0	0	0	0	0	0	0	0	0	0	0	1	2009-06-29 01:16:30	2009-06-29 11:13:12
7	000	0	0	0	0	0	0	0	0	0	0	0	1	2009-05-09 18:16:36	2009-05-09 18:30:36
8	000	0	0	0	0	0	0	0	0	0	0	0	1	2009-07-03 00:28:00	2009-07-03 08:37:23
9	000	0	0	0	0	0	0	0	0	0	0	0	1	2009-06-07 07:52:55	2009-06-07 09:46:03
10	000	0	0	0	0	0	0	0	0	0	0	0	1	2009-06-11 22:12:04	2009-06-11 22:34:54

First 10 entries of TrackPoint

id	activity_id	lat	lon	altitude	transportation_mode	date_days	date_time
1	1	40	116.327	105		39915.3	2009-04-12 07:33:03
2	1	40.0002	116.327	80		39915.3	2009-04-12 07:33:09
3	1	40.0001	116.327	99		39915.3	2009-04-12 07:33:14
4	1	40	116.327	109		39915.3	2009-04-12 07:33:19
5	1	40	116.327	111		39915.3	2009-04-12 07:33:24
6	1	40	116.327	114		39915.3	2009-04-12 07:33:29
7	1	39.9999	116.327	120		39915.3	2009-04-12 07:33:34
8	1	39.9997	116.327	125		39915.3	2009-04-12 07:33:39
9	1	39.9997	116.327	126		39915.3	2009-04-12 07:33:44
10	1	39.9995	116.327	127		39915.3	2009-04-12 07:33:49

Question 1:

Notice that there is a high number of activities. The reason for this will be discussed in the Discussion and has to do with our goal of making the application similar to Strava.

```
Table User has 182 rows
Table Activity has 25004 rows
Table TrackPoint has 9644128 rows
```

Question 2:

Average activities per user

avg_activity_count
144.532

Question 3:
We see that 163 is on top here, but if we only include activities that has trackpoints, 128 would be on top.

Activity count for all activites:

Users with the most activities

user_id	activity_count
163	3640
153	2294
128	2283
062	1046
085	949
167	854
068	853
025	715
144	569
075	509
126	450
052	404
041	399
084	391
004	346
140	345
010	335
112	308
147	291
017	265

Activity count for only activities which have trackpoints:

Users with the most activities

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

Question 4:
Including all activities:

```
Users who have taken taxi
  id
----
 010
 020
 021
 052
 056
 058
 062
 065
 068
 075
 078
 080
 082
 084
 085
 091
 098
 100
 102
 104
 105
 111
 114
 118
 126
 128
 139
 147
 153
 154
 161
 163
 167
 175
 179
```

Only including activities which have trackpoint:

Users who have taken taxi

id
010
021
052
056
058
062
065
078
080
084
085
098
102
105
111
114
126
128
139
153
161
163
167
175

Question 5:
For this question our implementation worked very well since we only had to sum over all the binary variables in the Activity table.

All activities:

Number of times each transportation mode has been used

walk	bike	bus	taxi	car	subway	train	airplane	boat	run	motorcycle
5704	1850	2727	1126	934	764	293	16	5	6	2

Only including activities with trackpoints:

Number of times each transportation mode has been used

walk	bike	bus	taxi	car	subway	train	airplane	boat	run	motorcycle
1668	748	839	246	545	357	60	4	2	2	0

Question 6a:

In both question 6a and 6b we see that there is data registered for the year 2000. In the description of the dataset, it said that all activities were between 2007-2011, but we have an activity from 2000 and 616 activities from 2012 which does not make sense. However, after taking a closer look at some of these activities, it did not seem like there was anything wrong with them and we decided to keep them. For instance, all the trackpoints to the activity from 2000 is shown below. It is only 3 trackpoints, but they all seem valid. Also, the numbers are quite high here and would be almost halfed if we did not include activities that has no trackpoints.

Total activities:

Number of activities per year

year	activity_count
2008	11229
2009	7732
2007	2021
2011	1872
2010	1511
2012	616
	22
2000	1

Only including activities which have trackpoints:

Number of activities per year

year	activity_count
2008	5885
2009	5868
2010	1487
2011	1203
2007	994
2012	588
	22
2000	1

Activity from the year 2000:

activity_id	date_time	lat	lon	altitude
20544	2000-01-01 23:12:19	39.989	116.327	129
20544	2000-01-01 23:13:21	39.991	116.327	221
20544	2000-01-01 23:15:23	39.9932	116.327	217

Question 6b:

Depending on whether we include only the activities with trackpoints or all the activities the year with most time spent changes from 2008 to 2009.

Time spent on all activities:

```
Time spent on activities per year
  year    time_spent
------  ------------
  2008    15130.4
  2009    13564.3
  2007     4054.78
  2010     1717.33
  2011     1565.14
  2012      719.856
  2000        0.0511
```

Time spent on activities which have trackpoints:

```
Time spent on activities per year
  year    time_spent
------  ------------
  2009    11598.7
  2008     9180.19
  2007     2314.67
  2010     1388.04
  2011     1132.18
  2012      711.186
  2000        0.0511
```

Question 7:
For this question only 10km moves between two trackpoints were allowed. This was to avoid any faulty data where two consecutive trackpoints was too far from each other. For this specific user, it did not matter, however, it could matter for other users. Most of this task is done using pandas.

```
Total kilometers traveled by user 112 in 2008:
189.26566171958515
```

Question 8:
For this question only altitudes between -300 and 50,000 feet were allowed. All altitudes of -777 altitudes were dropped. Also, if the altitude changed with more than 300 feet it was discarded. For this question, the only terms in the sum are the terms

where there was a positive difference in altitude between two trackpoints, i.e., where the user ascends. Most of this task is done using pandas.

```
Users who have gained the most meters
user_id
128    423558.9192
153    420090.2952
004    246928.8432
163    174334.6272
003    166500.6576
085    157895.5440
030    130048.4064
144    113926.6200
084    106138.0656
039    104687.8272
167     98995.0776
002     85703.0544
000     84559.7496
041     70956.8304
126     67963.0848
025     60717.3792
062     56684.2656
013     44247.5112
140     43981.1160
028     43863.4632
dtype: float64
```

Question 9:
We found that most users have invalid activities. This task was also done using some pandas.

```
Invalid activities per user (only includes users who has invalid activities)
    user_id  invalid_activity_count
0       000                     101
1       001                      45
2       002                     100
3       003                     179
4       004                     219
..      ...                     ...
168     176                       8
169     178                       0
170     179                      28
171     180                       2
172     181                      14
```

Question 10:
For this question we allowed the latitude to be between 39.915 and 39.917 and the longitude to be between 116.396 and 116.398.

```
Users who have registered activities in the Forbidden City
  user_id
---------
      004
      018
      019
      131
```

Question 11:
Including all activities:

```
Users most used transportation mode
id
010    walk
020    bike
021    walk
052     bus
053    walk
       ...
167    walk
170    walk
174     car
175     bus
179    walk
Length: 69, dtype: object
```

Only including activities with trackpoints:

```
Users most used transportation mode
id
010    taxi
020    bike
021    walk
052     bus
053    walk
       ...
167    bike
170    walk
174     car
175     bus
179    walk
Length: 69, dtype: object
```

Discussion

Discuss your solutions. Did you do anything differently than how it was explained in the assignment sheet, in that case why and how did that work? Were there any pain points or problems? What did you learn from this assignment?

What we did differently

We did some things a bit differently than what was described in the given description. They were only minor changes implemented to make the solution more realistic for an app similar to Strava. In Strava you can have multiple transportation modes per activity. You can for instance run and cycle in one activity in Strava. To better meet this requirement the table storing activities contained a column for each transportation mode stored as a BIT(). A different solution is to store a comma-separated list, but this is seen as an antipattern in relational databases. In addition, we stored the transportation mode for a given trackpoint as a string (VARCHAR). Trackpoints can only have one transportation mode since it is just a point in time and not an interval. This way we still have the ability to keep track of how much a user has walked and allow multiple transportation modes during one activity.

To make the app more similar to Strava where you can register activities without tracking them (with trackpoints), we decided to let an activity be defined in two ways. Either, you have a plt file with trackpoints where all trackpoints is one activity (which can be either labeled or not). Or, you have a labeled activity without any trackpoints. This is also why the number of activities may seem inflated in some questions. If we did not allow this the number of activities would be much lower. In fact, it would only be 16048 as shown below.

```
 count_activity
________________
          16048
```

Please see the figures below for a detailed description of the tables and datatypes.

Column	Datatype
id	VARCHAR(4)
has_labels	BIT(1)

Column	Datatype
id	SMALLINT()
user_id	VARCHAR(4)
walk	BIT(1)
bike	BIT(1)
bus	BIT(1)
taxi	BIT(1)
car	BIT(1)
subway	BIT(1)
train	BIT(1)
airplane	BIT(1)
boat	BIT(1)
run	BIT(1)
motorcycle	BIT(1)
valid_activity	BIT(1)
start_date_time	DATETIME
end_date_time	DATETIME

Column	Datatype
id	INT
activity_id	SMALLINT
lat	DOUBLE(8,6)
lon	DOUBLE(9,6)
altitude	MEDIUMINT
transportation_mode	VARCHAR(30)
date_days	DOUBLE(15,10)
date_time	DATETIME

Handling messy data

To make the functionality most similar to Strava we let any activity be defined by the .plt files with start time and end time equal to the min and max timestamp of that plt.file respectively. If an activity was labeled, but did not correspond to any TrackPoints, we added it as an activity, but flagged it as invalid. We did this because you can add activities in Strava without having any GPS tracking (i.e. no trackpoints). Before inserting the data, any trackpoints with highly unlikely values, for instance, the altitude was restricted to be between -300 and 50,000 feet were removed. The coordinates were also verified to be between -90 and 90 and -180 and 180 for latitude and longitude respectively. There is also some consecutive trackpoints which does not make sense, for instance, where the distance between them is unrealistically high. This is not handled when inserting the data but is handled during the queries

where the distance between trackpoint-coordinates must be within a certain range. The same goes for altitude.

Pain points

Most of the exercise was completed without any great obstacles. The only pain point experienced was during part 2 with some of the more complicated queries. Those queries often required extracting large chunks of data and manipulating it in pandas. Since extracting and manipulating such large quantities of data is computationally intensive and time consuming it required careful attention to detail when coding to avoid spending too long debugging.

What we learned

We learned a lot doing this project. The most important one and relating to this course would be how to actually implement and maintain a database and make choices about how to store data most efficiently with the application as context. Since we did not have access to a virtual machine from the start, we maintained a database locally hosted using Docker. Experience and knowledge about these things are hard to gain from a book and is best learned through doing projects. These skills are also essential for future employers. Additionally, we gained more experience with working with data in python and using pandas which is also highly appreciated by employers and very applicable to all sorts of problems be it academic or professional.

Feedback

Optional - give us feedback on the task if you have any. The assignment is new this semester and we would love to improve if there were any problems.

This was a very interesting exercise with the appropriate workload and difficulty.