

# TDT4225 - Assignment 2

Group 35  
Henrik Larsen  
Julie Vy Tran  
Eirik Reiestad

# Contents

<b>Contents</b>	<b>i</b>
<b>Summary</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Results</b>	<b>2</b>
2.1 Part 1 . . . . .	2
2.2 Part 2 . . . . .	4
2.2.1 How many users, activities, and trackpoints are there in the dataset (after it is inserted into the database). . . . .	4
2.2.2 Find the average, maximum, and minimum number of trackpoints per user. .	4
2.2.3 Find the top 15 users with the highest number of activities. . . . .	11
2.2.4 Find all users who have taken a bus. . . . .	11
2.2.5 List the top 10 users by their amount of different transportation modes. . . .	12
2.2.6 Find activities that are registered multiple times. You should find the query even if it gives zero result. . . . .	13
2.2.7 a) Find the number of users that have started an activity in one day and ended the activity the next day. b) List the transportation mode, user id and duration for these activities. . . . .	13
2.2.8 Find the number of users which have been close to each other in time and space. Close is defined as the same space (50 meters) and for the same half minute (30 seconds) . . . . .	15

2.2.9	Find the top 15 users who have gained the most altitude meters. . . . .	16
2.2.10	Find the users that have traveled the longest total distance in one day for each transportation mode. . . . .	16
2.2.11	Find all users who have invalid activities, and the number of invalid activities per user . . . . .	17
2.2.12	Find all users who have registered transportation_mode and their most used transportation_mode. . . . .	19
<b>3</b>	<b>Discussion</b>	<b>22</b>
3.1	Part 1 . . . . .	22
3.2	Part 2 . . . . .	22
<b>4</b>	<b>Feedback</b>	<b>25</b>
	<b>Attachments</b>	<b>25</b>

# Summary

This report is for Assignment 2 in TDT4225. The report will display and discuss the results from the tasks in the assignment.

# Chapter 1

## Introduction

In Part 1, we have defined the tables in the database based on the assignment's suggestions. After that, we solved the task of cleaning and inserting the given dataset into the created tables. For Part 2, we have answered the different questions using both MySQL queries and Python code to manipulate the data in such a way that it corresponds to the answer to the different questions.

As a group, we worked jointly for Part 1 of the assignment because we perceived that the different sub-tasks were closely related to each other. For Part 2 of the assignment, we decided to divide the tasks equally to each member as the different sub-tasks can be solved individually. The solution were reviewed by the other group members after each member has finished implementing their solution.

In Section 2, the results from our implementation will be presented, while in Section 3 we will discuss our solutions with respect to assumptions and decisions we have made to achieve those results.

Our code implementation consists of three main files. The file `part1.py` corresponds to creating the tables and inserting the dataset. The file `part1.ipynb` corresponds to visualize the tables after the dataset has been inserted. The file `part2.ipynb` includes the queries for the tasks in Part 2.

Our code implementation has been published to a Github repository. The link to our repository is given here: <https://github.com/EirikReiestad/TDT4225>

# Chapter 2

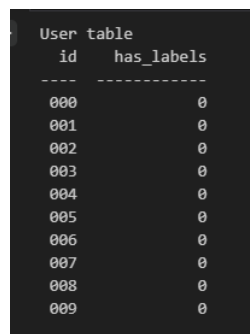
## Results

### 2.1 Part 1

The top 10 rows from the `User` table in text format are given below in Table 2.1 while the screenshot of the result is presented in Figure 2.1.

User table		
id	has	labels
000		0
001		0
002		0
003		0
004		0
005		0
006		0
007		0
008		0
009		0

**Table 2.1:** Result of inserting the data into the `User` table.



A screenshot of a terminal window showing the top 10 rows of the 'User' table. The table has two columns: 'id' and 'has\_labels'. The 'id' column contains values from 000 to 009, and the 'has\_labels' column contains the value 0 for all rows. The terminal output is as follows:

User table	
id	has_labels
000	0
001	0
002	0
003	0
004	0
005	0
006	0
007	0
008	0
009	0

**Figure 2.1:** Screenshot of the top 10 rows from the `User` table.

The top 10 rows from the **Activity** table in text format are given below in Table 2.2 while the screenshot of the result is presented in Figure 2.2.

Activity table				
id	user_id	transportation_mode	start_date_time	end_date_time
1	000		2008-10-23 02:53:04	2008-10-23 11:11:12
2	000		2008-10-24 02:09:59	2008-10-24 02:47:06
3	000		2008-10-26 13:44:07	2008-10-26 15:04:07
4	000		2008-10-27 11:54:49	2008-10-27 12:05:54
5	000		2008-10-28 00:38:26	2008-10-28 05:03:42
6	000		2008-10-29 09:21:38	2008-10-29 09:30:28
7	000		2008-10-29 09:30:38	2008-10-29 09:46:43
8	000		2008-11-03 10:13:36	2008-11-03 10:16:01
9	000		2008-11-03 23:21:53	2008-11-04 03:31:08
10	000		2008-11-10 01:36:37	2008-11-10 03:46:12

**Table 2.2:** Result of inserting the data into the Activity table.

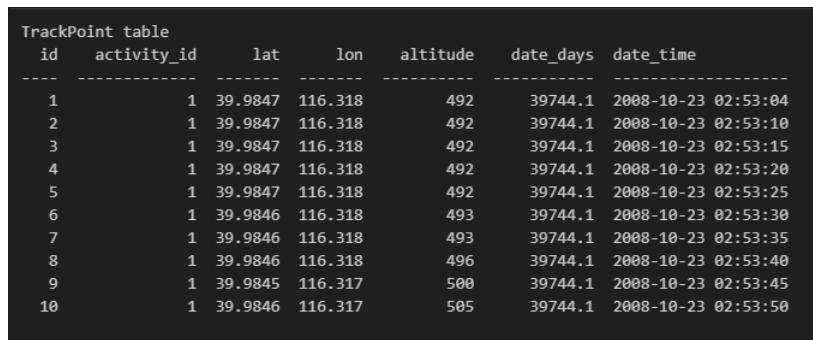
Activity table				
id	user_id	transportation_mode	start_date_time	end_date_time
1	000		2008-10-23 02:53:04	2008-10-23 11:11:12
2	000		2008-10-24 02:09:59	2008-10-24 02:47:06
3	000		2008-10-26 13:44:07	2008-10-26 15:04:07
4	000		2008-10-27 11:54:49	2008-10-27 12:05:54
5	000		2008-10-28 00:38:26	2008-10-28 05:03:42
6	000		2008-10-29 09:21:38	2008-10-29 09:30:28
7	000		2008-10-29 09:30:38	2008-10-29 09:46:43
8	000		2008-11-03 10:13:36	2008-11-03 10:16:01
9	000		2008-11-03 23:21:53	2008-11-04 03:31:08
10	000		2008-11-10 01:36:37	2008-11-10 03:46:12

**Figure 2.2:** Screenshot of the top 10 rows from the Activity table.

The top 10 rows from the **TrackPoint** table in text format are given below in Table 2.3 while the screenshot of the result is presented in Figure 2.3.

TrackPoint table						
id	activity_id	lat	lon	altitude	date_days	date_time
1	1	39.9847	116.318	492	39744.1	2008-10-23 02:53:04
2	1	39.9847	116.318	492	39744.1	2008-10-23 02:53:10
3	1	39.9847	116.318	492	39744.1	2008-10-23 02:53:15
4	1	39.9847	116.318	492	39744.1	2008-10-23 02:53:20
5	1	39.9847	116.318	492	39744.1	2008-10-23 02:53:25
6	1	39.9846	116.318	493	39744.1	2008-10-23 02:53:30
7	1	39.9846	116.318	493	39744.1	2008-10-23 02:53:35
8	1	39.9846	116.318	496	39744.1	2008-10-23 02:53:40
9	1	39.9845	116.317	500	39744.1	2008-10-23 02:53:45
10	1	39.9846	116.317	505	39744.1	2008-10-23 02:53:50

**Table 2.3:** Result of inserting the data into the TrackPoint table.



id	activity_id	lat	lon	altitude	date_days	date_time
1	1	39.9847	116.318	492	39744.1	2008-10-23 02:53:04
2	1	39.9847	116.318	492	39744.1	2008-10-23 02:53:10
3	1	39.9847	116.318	492	39744.1	2008-10-23 02:53:15
4	1	39.9847	116.318	492	39744.1	2008-10-23 02:53:20
5	1	39.9847	116.318	492	39744.1	2008-10-23 02:53:25
6	1	39.9846	116.318	493	39744.1	2008-10-23 02:53:30
7	1	39.9846	116.318	493	39744.1	2008-10-23 02:53:35
8	1	39.9846	116.318	496	39744.1	2008-10-23 02:53:40
9	1	39.9845	116.317	500	39744.1	2008-10-23 02:53:45
10	1	39.9846	116.317	505	39744.1	2008-10-23 02:53:50

**Figure 2.3:** Screenshot of the top 10 rows from the `TrackPoint` table.

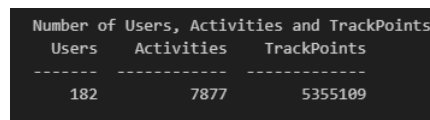
## 2.2 Part 2

### 2.2.1 How many users, activities, and trackpoints are there in the dataset (after it is inserted into the database).

The numbers of users, activities, and trackpoints there are in the dataset after it is inserted into the database is presented in text format below in Table 2.4 while the screenshot of the result is presented in Figure 2.4.

Users	Activities	TrackPoints
182	7877	5355109

**Table 2.4:** Result of Task 1 in text format.



Users	Activities	TrackPoints
182	7877	5355109

**Figure 2.4:** Screenshot of the result to Task 1.

### 2.2.2 Find the average, maximum, and minimum number of trackpoints per user.

#### Average trackpoints

The result of the average number of trackpoints per user in text format is given in Table 2.5, while the screenshot of the result is given in Figure 2.5.



<i>user_id</i>	<i>AVG(trackpoints)</i>	<i>user_id</i>	<i>AVG(trackpoints)</i>	<i>user_id</i>	<i>AVG(trackpoints)</i>
000	670.871	061	140	122	1321.17
001	824.965	062	439.054	123	1438
002	924.801	063	709.857	124	0
003	807.387	064	695	125	659.25
004	761.858	065	1114.94	126	437.182
005	587.548	066	889.375	127	335
006	801.125	067	1729	128	474.102
007	758.275	068	0	129	0
008	1376	069	12	130	771.714
009	1218.32	070	1185	131	858.4
010	1107.5	071	446.894	132	1125.33
011	451.756	072	40.5	133	765.75
012	629.783	073	197.302	134	55.44
013	1388.3	074	894.941	135	976.5
014	905.936	075	264	136	15.3333
015	806.583	076	77.3333	137	0
016	690.222	077	1048	138	98.5
017	868.245	078	304.667	139	21.6667
018	540.795	079	488.826	140	489.319
019	372.291	080	43.5	141	0
020	704.638	081	540.571	142	708.514
021	475	082	694.5	143	0
022	900.695	083	505.742	144	187.5
023	979.312	084	638.583	145	1075.2
024	860.408	085	367.882	146	27.2
025	606.295	086	63.6667	147	0
026	1034.71	087	19.8	148	0
027	2037	088	0	149	0
028	1195.36	089	74.75	150	127
029	1713.17	090	40	151	1322
030	871.352	091	9.5	152	782
031	489.5	092	53	153	374.6
032	1190.43	093	290.227	154	0
033	735.25	094	127	155	844.5
034	558.678	095	971.121	156	0
035	1449.71	096	0	157	192.308
036	803.75	097	27.7778	158	429.286
037	890.977	098	118	159	335.143
038	1071.51	099	79.1875	160	0
039	756.51	100	0	161	26
040	953.55	101	323.667	162	30.8182
041	797.416	102	148.692	163	95.2727
042	555.946	103	709.362	164	40.8571
043	795.094	104	0	165	1160.6
044	628.492	105	0	166	170
045	752.125	106	0	167	617.25
046	703.807	107	81	168	1537.19
047	65.8333	108	14.6667	169	1245.97
048	1118	109	148.75	170	0
049	0	110	0	171	35.4
050	760.833	111	29	172	758.278
051	231.3	112	351.028	173	161.667
052	177	113	605.193	174	0
053	0	114	0	175	7
054	1093	115	895.111	176	44
055	132.526	116	0	177	0
056	39.7333	117	24	178	84
057	127	118	0	179	0
058	38.5	119	987.469	180	293.25
059	0	120	0	181	45.8
060	8	121	1073.4		

Table 2.5: Result of Task 2a in text format.

Average trackpoints per user				id	average
---	-----	---	-----	---	-----
000	670.8710	060	8	121	1073.4
001	824.9649	061	140	122	1321.17
002	924.8014	062	439.054	123	1438
003	807.3870	063	709.857	124	0
004	761.8584	064	695	125	659.25
005	587.5479	065	1114.94	126	437.182
006	801.1250	066	889.375	127	335
007	758.2750	067	1729	128	474.102
008	1376.0000	068	0	129	0
009	1218.3243	069	12	130	771.714
010	1107.5000	070	1185	131	858.4
011	451.7562	071	446.894	132	1125.33
012	629.7826	072	40.5	133	765.75
013	1388.3025	073	197.302	134	55.44
014	905.9364	074	894.941	135	976.5
015	806.5833	075	264	136	15.3333
016	690.2222	076	77.3333	137	0
017	868.2453	077	1048	138	98.5
018	540.7955	078	304.667	139	21.6667
019	372.2911	079	488.826	140	489.319
020	704.6383	080	43.5	141	0
021	475.0000	081	540.571	142	708.514
022	900.6951	082	694.5	143	0
023	979.3125	083	505.742	144	187.5
024	860.4082	084	638.583	145	1075.2
025	606.2951	085	367.882	146	27.2
026	1034.7143	086	63.6667	147	0
027	2037.0000	087	19.8	148	0
028	1195.3585	088	0	149	0
029	1713.1667	089	74.75	150	127
030	871.3524	090	40	151	1322
031	489.5000	091	9.5	152	782
032	1190.4286	092	53	153	374.6
033	735.2500	093	290.227	154	0
034	558.6778	094	127	155	844.5
035	1449.7083	095	971.121	156	0
036	803.7500	096	0	157	192.308
037	890.9767	097	27.7778	158	429.286
038	1071.5139	098	118	159	335.143
039	756.5101	099	79.1875	160	0
040	953.5500	100	0	161	26
041	797.4160	101	323.667	162	30.8182
042	555.9455	102	148.692	163	95.2727
043	795.0938	103	709.362	164	40.8571
044	628.4918	104	0	165	1160.6
045	752.1250	105	0	166	170
046	703.8065	106	0	167	617.25
047	65.8333	107	81	168	1537.19
048	1118.0000	108	14.6667	169	1245.97
049	0.0000	109	148.75	170	0
050	760.8333	110	0	171	35.4
051	231.3000	111	29	172	758.278
052	177.0000	112	351.028	173	161.667
053	0.0000	113	605.193	174	0
054	1093.0000	114	0	175	7
055	132.5263	115	895.111	176	44
056	39.7333	116	0	177	0
057	127.0000	117	24	178	84
058	38.5000	118	0	179	0
059	0.0000	119	987.469	180	293.25
		120	0	181	45.8

**Figure 2.5:** Screenshot of the result to Task 2a. The first and third column also correspond to the `id` and the second and fourth correspond also correspond to the `average`.

## Maximum trackpoints

The result of the maximum number of trackpoints per user in text format is given in Table 2.6, while the screenshot of the result is given in Figure 2.6.

<i>user_id</i>	<i>MAX(trackpoints)</i>	<i>user_id</i>	<i>MAX(trackpoints)</i>	<i>user_id</i>	<i>MAX(trackpoints)</i>
000	2359	061	443	122	2328
001	2472	062	2447	123	2160
002	2438	063	1986	124	0
003	2485	064	695	125	1047
004	2482	065	2336	126	1629
005	2058	066	2009	127	565
006	2478	067	1729	128	2453
007	2228	068	0	129	0
008	2499	069	12	130	2106
009	2396	070	2486	131	2003
010	1964	071	1840	132	2119
011	2306	072	69	133	1390
012	2277	073	1214	134	599
013	2486	074	2314	135	2391
014	2499	075	264	136	18
015	2411	076	105	137	0
016	2360	077	1728	138	118
017	2471	078	1118	139	47
018	2245	079	897	140	2472
019	2276	080	47	141	0
020	2201	081	1047	142	2482
021	475	082	800	143	0
022	2421	083	2444	144	357
023	2215	084	2158	145	2227
024	2377	085	993	146	49
025	2464	086	72	147	0
026	2396	087	36	148	0
027	2480	088	36	149	0
028	2477	089	357	150	510
029	2440	090	132	151	1322
030	2454	091	11	152	2069
031	1254	092	54	153	1084
032	2358	093	1595	154	0
033	1205	094	510	155	2153
034	2448	095	1849	156	0
035	2493	096	0	157	806
036	2299	097	52	158	1832
037	2421	098	118	159	725
038	2500	099	648	160	0
039	2464	100	0	161	26
040	2434	101	617	162	110
041	2491	102	859	163	672
042	2350	103	1942	164	64
043	2005	104	0	165	2463
044	2206	105	0	166	631
045	1932	106	0	167	1331
046	2457	107	81	168	2495
047	227	108	20	169	2325
048	1118	109	307	170	0
049	0	110	0	171	97
050	2484	111	38	172	2456
051	951	112	1155	173	468
052	177	113	785	174	0
053	0	114	0	175	7
054	1527	115	2262	176	114
055	670	116	0	177	0
056	165	117	24	178	84
057	510	118	0	179	0
058	104	119	2442	180	1024
059	0	120	0	181	110
060	8	121	2457		

**Table 2.6:** Result of Task 2b in text format.

Maximum trackpoints per user					
				id	maximum
---	----	---	----	----	-----
000	2359	060	8	121	2457
001	2472	061	443	122	2328
002	2438	062	2447	123	2160
003	2485	063	1986	124	0
004	2482	064	695	125	1047
005	2058	065	2336	126	1629
006	2478	066	2009	127	565
007	2228	067	1729	128	2453
008	2499	068	0	129	0
009	2396	069	12	130	2106
010	1964	070	2486	131	2003
011	2306	071	1840	132	2119
012	2277	072	69	133	1390
013	2486	073	1214	134	599
014	2499	074	2314	135	2391
015	2411	075	264	136	18
016	2360	076	105	137	0
017	2471	077	1728	138	118
018	2245	078	1118	139	47
019	2276	079	897	140	2472
020	2201	080	47	141	0
021	475	081	1047	142	2482
022	2421	082	800	143	0
023	2215	083	2444	144	357
024	2377	084	2158	145	2227
025	2464	085	993	146	49
026	2396	086	72	147	0
027	2480	087	36	148	0
028	2477	088	0	149	0
029	2440	089	357	150	510
030	2454	090	132	151	1322
031	1254	091	11	152	2069
032	2358	092	54	153	1084
033	1205	093	1595	154	0
034	2448	094	510	155	2153
035	2493	095	1849	156	0
036	2299	096	0	157	806
037	2421	097	52	158	1832
038	2500	098	118	159	725
039	2464	099	648	160	0
040	2434	100	0	161	26
041	2491	101	617	162	110
042	2350	102	859	163	672
043	2005	103	1942	164	64
044	2206	104	0	165	2463
045	1932	105	0	166	631
046	2457	106	0	167	1331
047	227	107	81	168	2495
048	1118	108	20	169	2325
049	0	109	307	170	0
050	2484	110	0	171	97
051	951	111	38	172	2456
052	177	112	1155	173	468
053	0	113	785	174	0
054	1527	114	0	175	7
055	670	115	2262	176	114
056	165	116	0	177	0
057	510	117	24	178	84
058	104	118	0	179	0
059	0	119	2442	180	1024
		120	0	181	110

**Figure 2.6:** Screenshot of the result to Task 2b. The first and third columns also correspond to the `id` and the second and fourth correspond also correspond to the `maximum`.

## Minimum trackpoints

The result of the minimum number of trackpoints per user in text format is given in Table 2.7, while the screenshot of the result is given in Figure 2.7.

<i>user_id</i>	<i>MIN(trackpoints)</i>	<i>user_id</i>	<i>MIN(trackpoints)</i>	<i>user_id</i>	<i>MIN(trackpoints)</i>
000	5	061	10	122	161
001	33	062	4	123	15
002	4	063	13	124	0
003	3	064	695	125	106
004	4	065	149	126	39
005	5	066	17	127	90
006	14	067	1729	128	9
007	6	068	0	129	0
008	165	069	12	130	212
009	134	070	13	131	150
010	663	071	56	132	59
011	3	072	12	133	100
012	64	073	3	134	4
013	13	074	3	135	75
014	3	075	264	136	13
015	22	076	42	137	0
016	7	077	437	138	79
017	4	078	12	139	10
018	3	079	193	140	3
019	11	080	40	141	0
020	22	081	56	142	4
021	475	082	589	143	0
022	15	083	5	144	18
023	31	084	116	145	128
024	4	085	10	146	8
025	3	086	47	147	0
026	9	087	6	148	0
027	1760	088	0	149	0
028	32	089	15	150	9
029	4	090	4	151	1322
030	9	091	8	152	66
031	23	092	52	153	32
032	62	093	9	154	0
033	39	094	9	155	28
034	4	095	12	156	0
035	200	096	0	157	18
036	13	097	16	158	6
037	14	098	118	159	63
038	8	099	3	160	0
039	7	100	0	161	26
040	39	101	143	162	4
041	3	102	6	163	15
042	4	103	5	164	11
043	26	104	0	165	32
044	3	105	0	166	7
045	71	106	0	167	63
046	175	107	81	168	12
047	4	108	10	169	384
048	1118	109	26	170	0
049	0	110	0	171	3
050	3	111	24	172	5
051	5	112	46	173	14
052	177	113	271	174	0
053	0	114	0	175	7
054	659	115	29	176	8
055	5	116	0	177	0
056	16	117	24	178	84
057	9	118	0	179	0
058	8	119	26	180	15
059	0	120	0	181	3
060	8	121	175		

**Table 2.7:** Result of Task 2c in text format.

Minimum trackpoints per user					
				id	minimum
000	5	060	8	121	175
001	33	061	10	122	161
002	4	062	4	123	15
003	3	063	13	124	0
004	4	064	695	125	106
005	5	065	149	126	39
006	14	066	17	127	90
007	6	067	1729	128	9
008	165	068	0	129	0
009	134	069	12	130	212
010	663	070	13	131	150
011	3	071	56	132	59
012	64	072	12	133	100
013	13	073	3	134	4
014	3	074	3	135	75
015	22	075	264	136	13
016	7	076	42	137	0
017	4	077	437	138	79
018	3	078	12	139	10
019	11	079	193	140	3
020	22	080	40	141	0
021	475	081	56	142	4
022	15	082	589	143	0
023	31	083	5	144	18
024	4	084	116	145	128
025	3	085	10	146	8
026	9	086	47	147	0
027	1760	087	6	148	0
028	32	088	0	149	0
029	4	089	15	150	9
030	9	090	4	151	1322
031	23	091	8	152	66
032	62	092	52	153	32
033	39	093	9	154	0
034	4	094	9	155	28
035	200	095	12	156	0
036	13	096	0	157	18
037	14	097	16	158	6
038	8	098	118	159	63
039	7	099	3	160	0
040	39	100	0	161	26
041	3	101	143	162	4
042	4	102	6	163	15
043	26	103	5	164	11
044	3	104	0	165	32
045	71	105	0	166	7
046	175	106	0	167	63
047	4	107	81	168	12
048	1118	108	10	169	384
049	0	109	26	170	0
050	3	110	0	171	3
051	5	111	24	172	5
052	177	112	46	173	14
053	0	113	271	174	0
054	659	114	0	175	7
055	5	115	29	176	8
056	16	116	0	177	0
057	9	117	24	178	84
058	8	118	0	179	0
059	0	119	26	180	15
		120	0	181	3

**Figure 2.7:** Screenshot of the result to Task 2c. The first and third columns also correspond to the `id` and the second and fourth correspond also correspond to the `minimum`.

### 2.2.3 Find the top 15 users with the highest number of activities.

The top 15 users with the highest number of activities are presented in text format in Table 2.8 below while the screenshot of the result is presented in Figure 2.8.

Top	user_id	num_of_activities
1	025	715
2	128	519
3	062	406
4	041	399
5	004	346
6	140	345
7	017	265
8	003	261
9	014	236
10	030	210
11	011	201
12	039	198
13	034	180
14	000	155
15	002	146

**Table 2.8:** Result of Task 3 in text format.



Top 15 users with the highest number of activities		
Top	user_id	num_of_activities
1	025	715
2	128	519
3	062	406
4	041	399
5	004	346
6	140	345
7	017	265
8	003	261
9	014	236
10	030	210
11	011	201
12	039	198
13	034	180
14	000	155
15	002	146

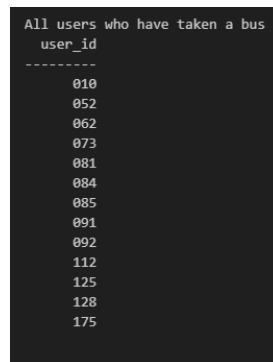
**Figure 2.8:** Screenshot of the result to Task 3.

### 2.2.4 Find all users who have taken a bus.

The users who have taken a bus are presented in text format in Table 2.9 below while the screenshot of the result is presented in Figure 2.9.

```
user_id
-----
010
052
062
073
081
084
085
091
092
112
125
128
175
```

**Table 2.9:** Result of Task 4 in text format.



```
All users who have taken a bus
user_id
-----
010
052
062
073
081
084
085
091
092
112
125
128
175
```

**Figure 2.9:** Screenshot of the result to Task 4.

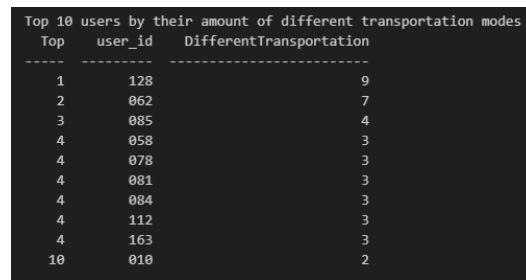
### 2.2.5 List the top 10 users by their amount of different transportation modes.

The top 10 users by their amount of different transportation modes are presented in text format in Table 2.10 below while the screenshot of the result is presented in Figure 2.2.5.

Top	user_id	DifferentTransportation
1	128	9
2	062	7
3	085	4
4	058	3
5	078	3
6	081	3
7	084	3
8	112	3
9	163	3
10	010	2

**Table 2.10:** Result of Task 5 in text format.





Top	user_id	DifferentTransportation
1	128	9
2	062	7
3	085	4
4	058	3
4	078	3
4	081	3
4	084	3
4	112	3
4	163	3
10	010	2

Figure 2.10: Screenshot of the result to Task 5.

### 2.2.6 Find activities that are registered multiple times. You should find the query even if it gives zero result.

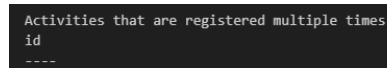
Activities that are registered multiple times are presented in text format in Table 2.11 below while the screenshot of the result is presented in Figure 2.11. The query gives zero result.

```

      id
      -

```

Table 2.11: Result of Task 6 in text format.



```

Activities that are registered multiple times
id
----

```

Figure 2.11: Screenshot of the result to Task 6.

### 2.2.7 a) Find the number of users that have started an activity in one day and ended the activity the next day. b) List the transportation mode, user id and duration for these activities.

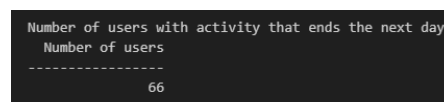
The number of users that have started activity in one day and ended the activity the next day is presented in text format in Table 2.12 below while the screenshot of the result is presented in Figure 2.12.

```

Number of users
-----
66

```

Table 2.12: Result of Task 7a in text format.



```

Number of users with activity that ends the next day
Number of users
-----
66

```

Figure 2.12: Screenshot of the result to Task 7a.

The transportation mode, user id, and duration for these activities are presented in text format in Table 2.13 below while the screenshot of the result is presented in Figure 2.13.

user_id	transportation_mode	duration
020	bike	10:10:01
021	walk	3:57:13
058	car	0:36:12
062	walk	1:23:44
085	bus	0:58:25
115	car	1:28:04
115	car	1:02:50
115	car	1:00:56
115	car	1:05:10
115	car	1:23:14
115	car	0:54:09
115	car	1:08:43
115	car	1:09:36
115	car	1:03:47
115	car	1:01:42
115	car	0:55:50
115	car	1:24:08
115	car	1:17:13
115	car	1:01:33
115	car	1:03:29
115	car	0:58:19
115	car	7:46:29
115	car	1:05:39
115	car	0:57:48
115	car	1:08:14
126	walk	0:07:20
128	car	1:04:19
128	car	0:44:11
128	bus	0:58:44
128	subway	0:30:57
128	car	0:33:08
128	car	1:40:04
128	airplane	5:31:49
128	subway	0:27:26
163	bike	0:11:18

**Table 2.13:** Result of Task 7b in text format.

User	Transportation Mode	Duration
020	bike	10:10:01
021	walk	3:57:13
058	car	0:36:12
062	walk	1:23:44
085	bus	0:58:25
115	car	1:28:04
115	car	1:02:50
115	car	1:00:56
115	car	1:05:10
115	car	1:23:14
115	car	0:54:09
115	car	1:08:43
115	car	1:09:36
115	car	1:03:47
115	car	1:01:42
115	car	0:55:50
115	car	1:24:08
115	car	1:17:13
115	car	1:01:33
115	car	1:03:29
115	car	0:58:19
115	car	7:46:29
115	car	1:05:39
115	car	0:57:48
115	car	1:08:14
126	walk	0:07:20
128	car	1:04:19
128	car	0:44:11
128	bus	0:58:44
128	subway	0:30:57
128	car	0:33:08
128	car	1:40:04
128	airplane	5:31:49
128	subway	0:27:26
163	bike	0:11:18

**Figure 2.13:** Screenshot of the result to Task 7b.

### 2.2.8 Find the number of users which have been close to each other in time and space. Close is defined as the same space (50 meters) and for the same half minute (30 seconds)

The number of users who have been close to each other in time and space is presented in text format below while the screenshot of the result is presented in Figure 2.14.

81 have been close to each other in time and space

81 have been close to each other in time and space

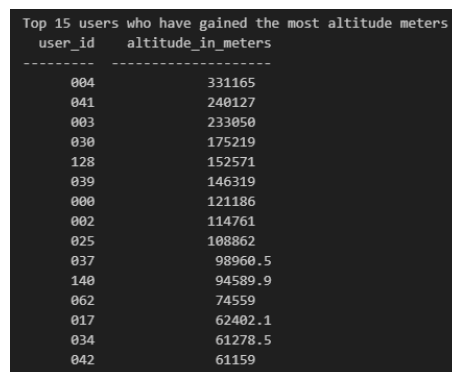
**Figure 2.14:** Screenshot of the result to Task 8.

### 2.2.9 Find the top 15 users who have gained the most altitude meters.

The top 15 users who have gained the most altitude meters are presented in text format in Table 2.14 below while the screenshot of the result is presented in Figure 2.15.

Top	user_id	altitude_in_meters
1	004	331165
2	041	240127
3	003	233050
4	030	175219
5	128	152571
6	039	146319
7	000	121186
8	002	114761
9	025	108862
10	037	98960.5
11	140	94589.9
12	062	74559
13	017	62402.1
14	034	61278.5
15	042	61159

**Table 2.14:** Result of Task 9 in text format.



Top 15 users who have gained the most altitude meters	
user_id	altitude_in_meters
004	331165
041	240127
003	233050
030	175219
128	152571
039	146319
000	121186
002	114761
025	108862
037	98960.5
140	94589.9
062	74559
017	62402.1
034	61278.5
042	61159

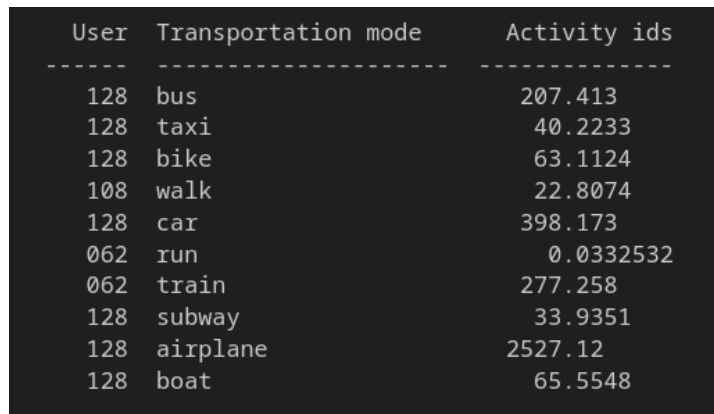
**Figure 2.15:** Screenshot of the result to Task 9.

### 2.2.10 Find the users that have traveled the longest total distance in one day for each transportation mode.

The users that have traveled the longest total distance in one day for each transportation mode are presented in text format in Table 2.15 below while the screenshot of the result is presented in Figure 2.16.

transportation_mode	user_id	distance(km)
128	bus	207.413
128	taxi	40.2233
139	walk	63.1124
128	bike	22.8074
128	car	398.173
062	run	0.0332532
062	train	277.258
128	subway	33.9351
128	airplane	2527.12
128	boat	65.5548

**Table 2.15:** Result of Task 10 in text format.



User	Transportation mode	Activity ids
128	bus	207.413
128	taxi	40.2233
128	bike	63.1124
108	walk	22.8074
128	car	398.173
062	run	0.0332532
062	train	277.258
128	subway	33.9351
128	airplane	2527.12
128	boat	65.5548

**Figure 2.16:** Screenshot of the result to Task 10.

### 2.2.11 Find all users who have invalid activities, and the number of invalid activities per user

The users who have invalid activities and the number of invalid activities per user are presented in text format in Table 2.16 below while the screenshot of the result is presented in Figure 2.17.

<i>user</i>	<i>id</i>	<i>count</i>	<i>user</i>	<i>id</i>	<i>count</i>	<i>user</i>	<i>id</i>	<i>count</i>
	000	442		047	19		112	18
	001	116		048	5		113	1
	002	295		050	17		115	60
	003	839		051	146		119	79
	004	1169		054	8		121	15
	005	126		055	28		122	21
	006	48		056	2		123	21
	007	133		057	60		126	6
	008	33		058	3		127	8
	009	114		060	1		128	215
	011	38		061	41		130	18
	012	163		062	206		131	59
	013	68		063	26		132	3
	014	328		065	6		133	18
	015	136		066	40		134	83
	016	60		067	1		135	21
	017	492		069	1		139	7
	018	97		070	15		140	195
	019	65		071	61		142	200
	020	15		072	6		144	1
	021	1		073	10		145	17
	022	188		074	19		146	28
	023	30		075	1		150	60
	024	73		076	5		151	4
	025	424		077	11		152	4
	026	51		078	7		153	2
	027	13		079	3		155	70
	028	90		080	2		157	65
	029	37		081	2		158	28
	030	509		083	56		159	7
	031	7		084	12		162	27
	032	36		085	17		163	4
	033	3		086	2		164	20
	034	240		087	1		165	4
	035	122		089	8		166	4
	036	125		090	15		167	4
	037	279		092	1		168	36
	038	264		093	4		169	12
	039	423		094	60		171	13
	040	50		095	20		172	21
	041	386		097	4		173	11
	042	117		099	113		175	1
	043	72		103	44		176	24
	044	80		108	7		180	2
	045	19		109	9		181	49
	046	26		111	2			

Table 2.16: Result of Task 11 in text format.

user_id	count(*)	user_id	count(*)	user_id	count(*)	user_id	count(*)
000	442	035	122	075	1	130	18
001	116	036	125	076	5	131	59
002	295	037	279	077	11	132	3
003	839	038	264	078	7	133	18
004	1169	039	423	079	3	134	83
005	126	040	50	080	2	135	21
006	48	041	386	081	2	139	7
007	133	042	117	083	56	140	195
008	33	043	72	084	12	142	200
009	114	044	80	085	17	144	1
011	38	045	19	086	2	145	17
012	163	046	26	087	1	146	28
013	68	047	19	089	8	150	60
014	328	048	5	090	15	151	4
015	136	050	17	092	1	152	4
016	60	051	146	093	4	153	2
017	492	054	8	094	60	155	70
018	97	055	28	095	20	157	65
019	65	056	2	097	4	158	28
020	15	057	60	099	113	159	7
021	1	058	3	103	44	162	27
022	188	060	1	108	7	163	4
023	30	061	41	109	9	164	20
024	73	062	206	111	2	165	4
025	424	063	26	112	18	166	4
026	51	065	6	113	1	167	4
027	13	066	40	115	60	168	36
028	90	067	1	119	79	169	12
029	37	069	1	121	15	171	13
030	509	070	15	122	21	172	21
031	7	071	61	123	21	173	11
032	36	072	6	126	6	175	1
033	3	073	10	127	8	176	24
034	240	074	19	128	215	180	2
						181	49

Figure 2.17: Screenshot of the result to Task 11.

### 2.2.12 Find all users who have registered transportation\_mode and their most used transportation\_mode.

The users who have registered transportation mode and their most used transportation mode are presented in text format in Table 2.17 below while the screenshot of the result is presented in Figure 2.18.

<u>user_id</u>	<u>most used transportation mode</u>	<u>user_id</u>	<u>most used transportation mode</u>
010	taxi	091	bus
020	bike	092	bus
021	walk	097	bike
052	bus	098	taxi
056	bike	101	car
058	car	102	bike
060	walk	107	walk
062	bus	108	walk
064	bike	111	taxi
065	bike	112	walk
067	walk	115	car
069	bike	117	walk
073	walk	125	bike
075	walk	126	bike
076	car	128	car
078	walk	136	walk
080	bike	138	bike
081	bike	139	bike
082	walk	144	walk
084	walk	153	walk
085	walk	161	walk
086	car	163	bike
087	walk	167	bike
089	car	175	bus

**Table 2.17:** Result of Task 12 in text format.



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

**Figure 2.18:** Screenshot of the result to Task 12.

# Chapter 3

## 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?

### 3.1 Part 1

For Part 1, we followed the assignment's suggestions and did not do anything differently. However, we did make assumptions based on our interpretation of the task's description. First of all, an assumption we made was that trackpoints of users without transportation labels should also be inserted into the tables just with an activity transportation mode as `NULL`. This is because, in the assignment, it is stated that each activity is stored as a `.plt` file in the Trajectory. In addition, in part 2, tasks 1, 2, 3, 6, 7, 8, and 9 do not restrict to only activities with transportation mode. We have therefore solved it this way. Another assumption based on the assignment text in part "Therefore we only want you to insert activities that have fewer than or exactly 2500 trackpoints in them, i.e. when inserting activities into the database, check that the size of the `.plt`-files do not exceed 2500 lines (excluding the headers, of course). When you insert TrackPoints, the same rule applies (you cannot link a trackpoint to an activity that is dropped, anyway)". As the assignment text says only insert activities that have fewer than or exactly 2500 trackpoints and if an activity is dropped then you cannot link a trackpoint, thus, we dropped all the trackpoints that belong to activities with more than 2500 trackpoints. We also dropped all activities which do not have any trackpoints as they are not relevant. A problem that occurred when we tried to insert track points was that there exists user(s) that had activities with different transportation modes but with the same `start_date_time` and `end_date_time`. Thus, we had to add a convention of taking the last inserted activity as the one the track points should be related to.

### 3.2 Part 2

We did not do anything differently than how it was explained in the assignment sheet, but we made some assumptions and decisions for some tasks based on how we interpret the different tasks.

In Task 6 where we are supposed to find activities that are registered multiple times, we assumed that the duplicates of activities will have the same transportation mode, end time, start time, and user. However, it is also possible to interpret it as two activities are the same if the end time, start time, and user are the same in cases where it is registered multiple times but just with different transportation modes. Then we only need to remove the `WHERE` clause for transportation mode to get this. However, we have chosen to interpret it based on the former and not the latter because it says registered multiple times and does not explicitly say register with different transportation modes.

In Task 7b, as the question explicitly mentions transportation mode, we do not count the rows in the `Activity` table with transportation mode equals `NULL` as we interpret it as they are not relevant to this question. Therefore, Task 7b has fewer rows than 66 rows. Although to include the rows with transportation mode as `NULL` as well, then we only need to remove the part `transportation_mode IS NOT NULL` of the solution query.

In Task 8, we extracted every trackpoint and connected it to the activity table by joining on `activity_id`. Then we sorted the output by date to make it easy to find the activities close in time to each other. Then we looped through each row and looked for the combinations within 30 seconds. If they were within 30 seconds we checked if the latitude and longitude were within 50 using the haversine method to find the distance between the two points

In Task 10, we were asked for the "total distance in one day", which can be interpreted in different ways. On one hand, it could mean a 24-hour interval where we count the distance for one activity and ensure that it doesn't exceed the timespan. However, it is also possible to interpret it differently. When we solved this task, we followed the same thought process as in Task 7, by restraining it to be within one date day and not exceeding that. In addition, the user is not restricted to a continuous activity, meaning a user could have a run in the morning, and another run in the evening, and those two activities would be added up to the total distance for one transportation mode. However, in the handed in code, we have also included a solution to the 24-hour interval, but because it was time consuming, we were not able to get a result out of it.

In Task 11, we assumed each trackpoint for an activity is inserted sequentially based on our observation from the dataset (actually our implementation in Part 1). Thus, we used the previous id as our `LAG` row. However, if the dataset was not insert sequentially then this solution will not work as it will compare the wrong rows. A solution to this could be by using the MySQL function `LAG` to retrieve the previous row based on the order of the `datetime`.

In Task 12, we decided that for users that have the same number of activities tagged on multiple transportation modes, the first transportation mode in alphabetical order should be chosen. We decided on this because it is a simple solution. However, a more complex solution could for example be choosing the transportation mode that is most recent. Thus, there are different ways to choose, and depends on the use case and what is most preferred. Another thing in this task is that we added an extra `JOIN` and `WHERE` for the `User` table even though we will get the same result without that part. This is because, even if the user has some activities with transportation mode in the `Activity` table, but has the property `has_labels` as `FALSE`. Then the user should not be counted because of the line "users who have registered transportation\_mode" in the assignment text. However, from Part 1, we have done so that all activities related to users that do not register transportation mode have transportation mode as `NULL`. Thus, the `WHERE` clause on the `Activity` should be sufficient enough. Despite that, if some rows in the `Activity` table do not follow the rule given in Part 1, then the query without `JOIN` might lead to a result that includes users who do not have the property

`has_labels` as `TRUE`.

We also had to figure out in the assignments, when it was suitable to only use pure SQL, and when to combine SQL queries with Python to solve the task. For example, tasks 8 and 10 were solved more efficiently and easier by combining SQL with Python than with just pure SQL.

## Chapter 4

### Feedback

We find the tasks quite diverse in difficulty which is a good thing as not all tasks are too difficult or too easy. However, there were some tasks that we find it difficult to interpret, for example like task 8 and 10 that was mentioned in the previous section. Except for that, we found the assignment relevant and learnt a lot from it.