
miniDBProfiler use and information.

Project Contributors: Dimitriadou Eleftheria P18035 and Panagiotis Kalogeridis P18052.

To profile a miniDB command:

- 1.from miniDBProfiler import miniDBProfiler
- 2.create a miniDBProfiler instance: mp=miniDBProfiler()
- 3.use method profileMiniDB() to profile a command: mp.profileMiniDB("your_command",your_db)

Example to create new table in a database loaded on an instance called db1:

```
mp.profileMiniDB("db.create_table('new_table',['col1','col2'],[str,str])", db1)
```

Note: in the command string you must use db as a database instance name(ex. db.your_command, db=your_command)

and then give the database instance name you are using as the second argument.

When you use miniDBProfiler, the command you profile will be executed and you will see its results.

Under the result you will see the profiling results.

Those are the date and time the profiling happened, the number of all function calls used for the execution,

total time of execution followed by in-depth information on miniDB methods called.

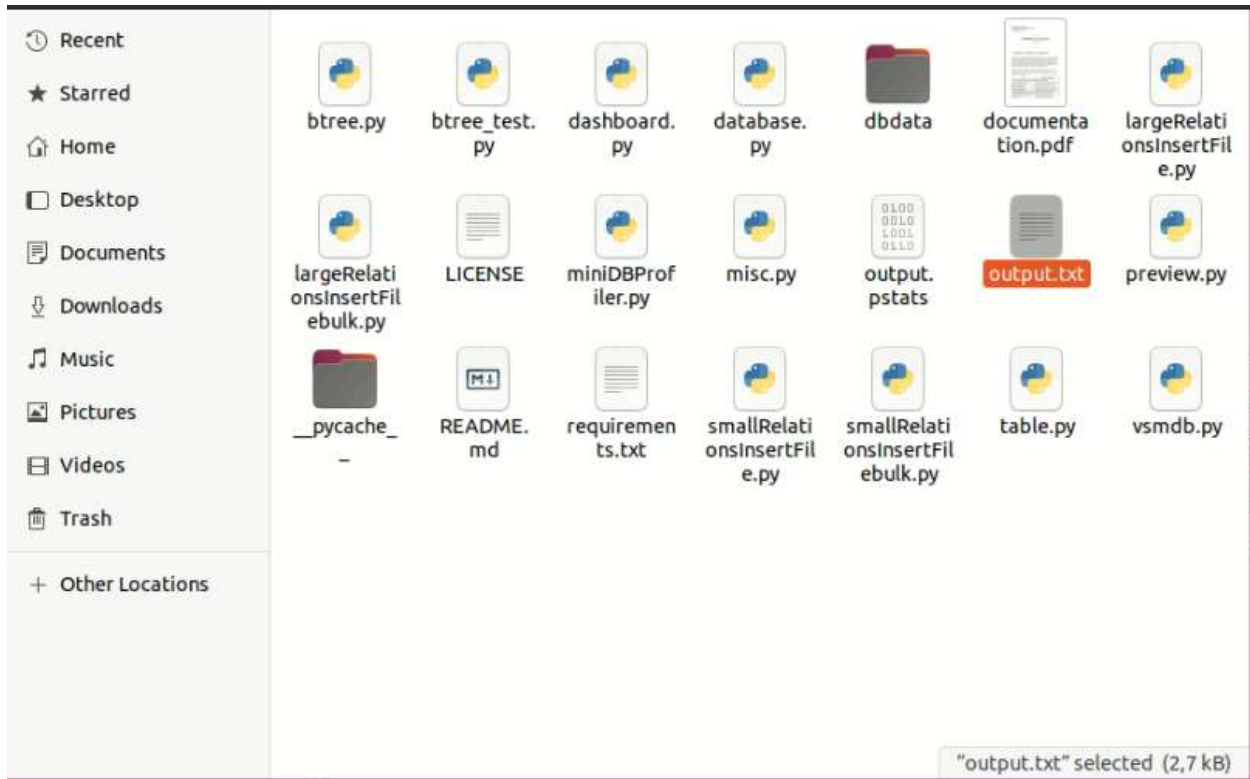
```
>>> from miniDBProfiler import miniDBProfiler
>>> mp = miniDBProfiler()
>>> mp.profileMiniDB("db.create_table('new_table',['col1','col2'],[str,str])", db)
New table "new_table"

-----
Sun Feb 21 11:37:56 2021    output.pstats
122 function calls in 0.002 seconds
From Which miniDB:

ncalls      tottime    percall    cuntime    percall    filename
-----
      2         0         0         0         0    /home/panagiotis/miniDB/misc.py:18(split_condition)
      2         0         0         0         0    /home/panagiotis/miniDB/table.py:354(_parse_condition)
      3         0         0         0         0    /home/panagiotis/miniDB/misc.py:3(get_op)
      2         0         0         0         0    /home/panagiotis/miniDB/table.py:120(_update_row)
      1         0         0    0.001     0.001    /home/panagiotis/miniDB/database.py:48(save)
      1         0         0         0         0    /home/panagiotis/miniDB/database.py:83(_update)
      1         0         0    0.001     0.001    /home/panagiotis/miniDB/database.py:92(create_table)
      2         0         0         0         0    /home/panagiotis/miniDB/database.py:453(<listcomp>)
      1         0         0         0         0    /home/panagiotis/miniDB/database.py:441(_update_meta_length)
      1         0         0         0         0    /home/panagiotis/miniDB/database.py:457(_update_meta_locks)
      1         0         0         0         0    /home/panagiotis/miniDB/database.py:469(_update_meta_insert_stack)
      1         0         0         0         0    /home/panagiotis/miniDB/table.py:24(__init__)
      6         0         0         0         0    /home/panagiotis/miniDB/table.py:76(<listcomp>)
      6         0         0         0         0    /home/panagiotis/miniDB/table.py:72(_update)
      3         0         0         0         0    /home/panagiotis/miniDB/table.py:94(_insert)
>>> █
```

- Specifically:
- ncalls = number of times a method was called
 - tottime = total time of execution all times a method was called
 - percall = time for one call (tottime/ncalls)
 - cumtime = total time spent on function and all subfunctions (accurate even for recursive functions)
 - percall = time for primitive functions for one call(cumtime/primitive calls)
 - filename = function data (path, name of file and name of function)
-

A file named output.txt is created and saved with information about all functions called (not only miniDB's) and can be accessed any time (Though it's re-written every time you use miniDBProfiler).



```
Open [F] output.txt ~miniDB
1 Sun Feb 21 11:40:24 2021 output.pstats
2
3 153 function calls in 0.002 seconds
4
5 Random listing order was used
6
7 nccalls tottime perccall cuntime perccall filename:lineno(function)
8 8 0.000 0.000 0.000 0.000 {method 'append' of 'list' objects}
9 12 0.000 0.000 0.000 0.000 {method '__dir__' of 'object' objects}
10 5 0.000 0.000 0.000 0.000 {method 'items' of 'dict' objects}
11 12 0.000 0.000 0.000 0.000 {method 'values' of 'dict' objects}
12 4 0.000 0.000 0.000 0.000 {method 'update' of 'dict' objects}
13 1 0.000 0.000 0.002 0.002 {built-in method builtins.exec}
14 12 0.000 0.000 0.000 0.000 {built-in method builtins.len}
15 4 0.000 0.000 0.000 0.000 {built-in method builtins.print}
16 20 0.000 0.000 0.000 0.000 {built-in method builtins setattr}
17 1 0.000 0.000 0.000 0.000 {built-in method posix.stat}
18 1 0.000 0.000 0.000 0.000 {built-in method posix.mkdir}
19 14 0.000 0.000 0.000 0.000 {built-in method _pickle.dump}
20 15 0.001 0.000 0.001 0.000 {built-in method io.open}
21 1 0.000 0.000 0.002 0.002 <string>:1(<module>)
22 1 0.000 0.000 0.000 0.000 /usr/lib/python3.8/cProfile.py:44(dump_stats)
23 1 0.000 0.000 0.000 0.000 /usr/lib/python3.8/cProfile.py:50(create_stats)
24 1 0.000 0.000 0.002 0.002 /home/panagiotis/miniDB/database.py:15(__init__)
25 5 0.001 0.000 0.002 0.000 /home/panagiotis/miniDB/database.py:40(save)
26 4 0.000 0.000 0.000 0.000 /home/panagiotis/miniDB/database.py:83(update)
27 4 0.000 0.000 0.002 0.000 /home/panagiotis/miniDB/database.py:92(create_table)
28 4 0.000 0.000 0.000 0.000 /home/panagiotis/miniDB/database.py:441(update_meta_length)
29 4 0.000 0.000 0.000 0.000 /home/panagiotis/miniDB/database.py:457(update_meta_locks)
30 4 0.000 0.000 0.000 0.000 /home/panagiotis/miniDB/database.py:469(update_meta_insert_stack)
31 4 0.000 0.000 0.000 0.000 /home/panagiotis/miniDB/table.py:24(__init__)
32 4 0.000 0.000 0.000 0.000 /home/panagiotis/miniDB/table.py:76(listcomp)
33 4 0.000 0.000 0.000 0.000 /home/panagiotis/miniDB/table.py:72(update)
34 1 0.000 0.000 0.000 0.000 {method 'enable' of '_lsprof.Profiler' objects}
35 1 0.000 0.000 0.000 0.000 {method 'disable' of '_lsprof.Profiler' objects}
36 1 0.000 0.000 0.000 0.000 /usr/lib/python3.8/genericpath.py:16(exists)
37
38
```

You can also only call function createPrint() to see only miniDB profile that was saved the last time you used miniDBProfiler: database_instance.createPrint()
example: mp.createPrint()

```
>>> mp.createPrint()

-----
Sun Feb 21 11:40:24 2021    output.pstats
153 function calls in 0.002 seconds
From Which miniDB:

ncalls      tottime    percall    cumtime    percall    filename
-----
1          0          0          0.002      0.002      /home/panagiotis/miniDB/database.py:15(__init__)
5         0.001          0          0.002      0          /home/panagiotis/miniDB/database.py:48(save)
4          0          0          0          0          /home/panagiotis/miniDB/database.py:83(_update)
4          0          0          0.002      0          /home/panagiotis/miniDB/database.py:92(create_table)
4          0          0          0          0          /home/panagiotis/miniDB/database.py:441(_update_meta_length)
4          0          0          0          0          /home/panagiotis/miniDB/database.py:457(_update_meta_locks)
4          0          0          0          0          /home/panagiotis/miniDB/database.py:469(_update_meta_insert_stack)
4          0          0          0          0          /home/panagiotis/miniDB/table.py:24(__init__)
4          0          0          0          0          /home/panagiotis/miniDB/table.py:76(<listcomp>)
4          0          0          0          0          /home/panagiotis/miniDB/table.py:72(_update)
>>> 
```

Other Comments:

When you use miniDBProfiler and create a new database, you have to load it to get the result in a database instance.

```
>>> mp.profileMiniDB("db1=Database('new_db',load=False)", db)
New table "meta_length"
New table "meta_locks"
New table "meta_insert_stack"
New table "meta_indexes"

-----
Sun Feb 21 11:40:24 2021      output.pstats
153 function calls in 0.002 seconds
From Which miniDB:

ncalls      tottime      percall      cumtime      percall      filename
-----
1          0          0          0.002        0.002        /home/panagiotis/miniDB/database.py:15(__init__)
5         0.001          0          0.002          0        /home/panagiotis/miniDB/database.py:48(save)
4          0          0          0          0        /home/panagiotis/miniDB/database.py:83(_update)
4          0          0          0.002          0        /home/panagiotis/miniDB/database.py:92(create_table)
4          0          0          0          0        /home/panagiotis/miniDB/database.py:441(_update_meta_length)
4          0          0          0          0        /home/panagiotis/miniDB/database.py:457(_update_meta_locks)
4          0          0          0          0        /home/panagiotis/miniDB/database.py:469(_update_meta_insert_stack)
4          0          0          0          0        /home/panagiotis/miniDB/table.py:24(__init__)
4          0          0          0          0        /home/panagiotis/miniDB/table.py:76(<listcomp>)
4          0          0          0          0        /home/panagiotis/miniDB/table.py:72(_update)
>>> |
```

```
>>> db1
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'db1' is not defined
>>> db1=Database('new_db',load=True)
Loaded "new_db".
>>> db1
<database.Database object at 0x7f3afa203f40>
>>> |
```