



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Εργασία για το μάθημα συστήματα
διαχείρισης βάσεων
(mini DB)

Σπουδαστές: Καλλένος Κωνσταντίνος, Π18179

Καλογήρου Στυλιανή, Π18181

Φεβρουάριος 2021

Ερώτημα που κλήθηκε να λύσει η ομάδα μας:

Task 3.3 - SQL compiler ()**

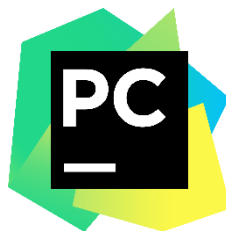
Υλοποίηση ενός SQL compiler για την υποστήριξη των I/O εντολών που υπάρχουν στην τρέχουσα έκδοση της miniDB. Ενδεικτικά, οι εντολές που μπορούν να υποστηριχθούν είναι οι εξής:

1. Select From Where
2. Update Where
3. Insert Into
4. Delete From Where
5. Create/Drop Table
6. Create Database
7. Select From Join
8. Create Index On

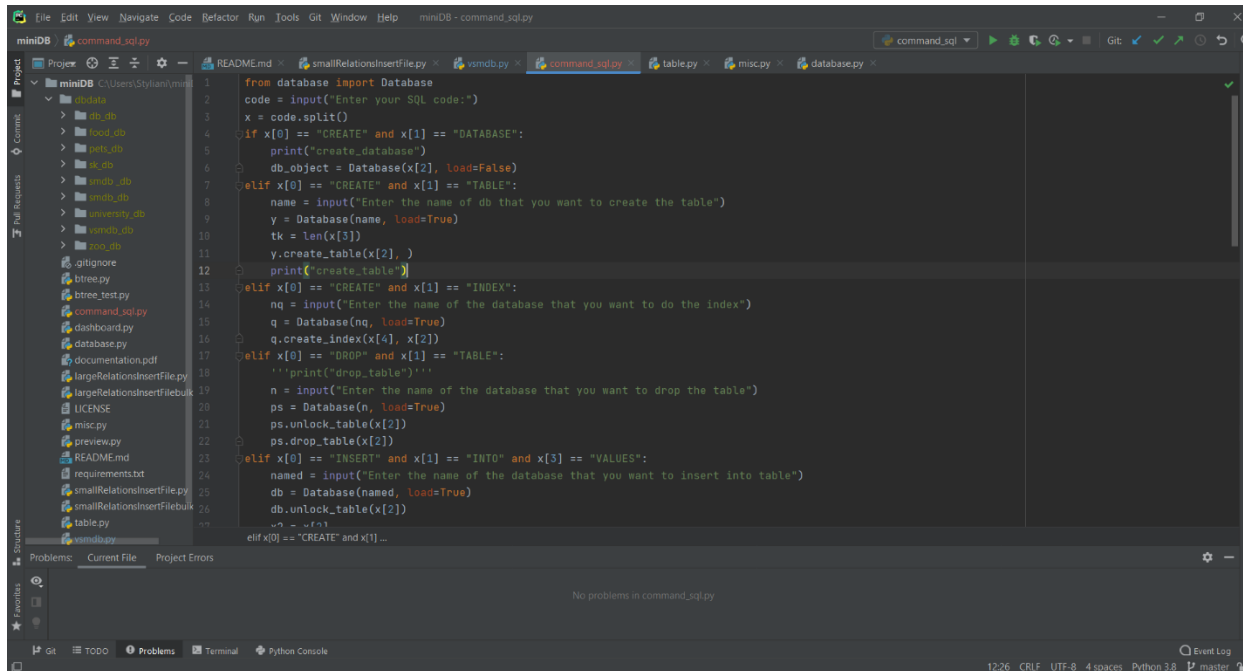
Υλοποίηση Ερωτημάτων

Αρχικά να αναφέρουμε ότι τα πιο πάνω υλοποιήθηκαν με την γλώσσα προγραμματισμού python 3.8 στο πρόγραμμα py Charm της Jet Brains.

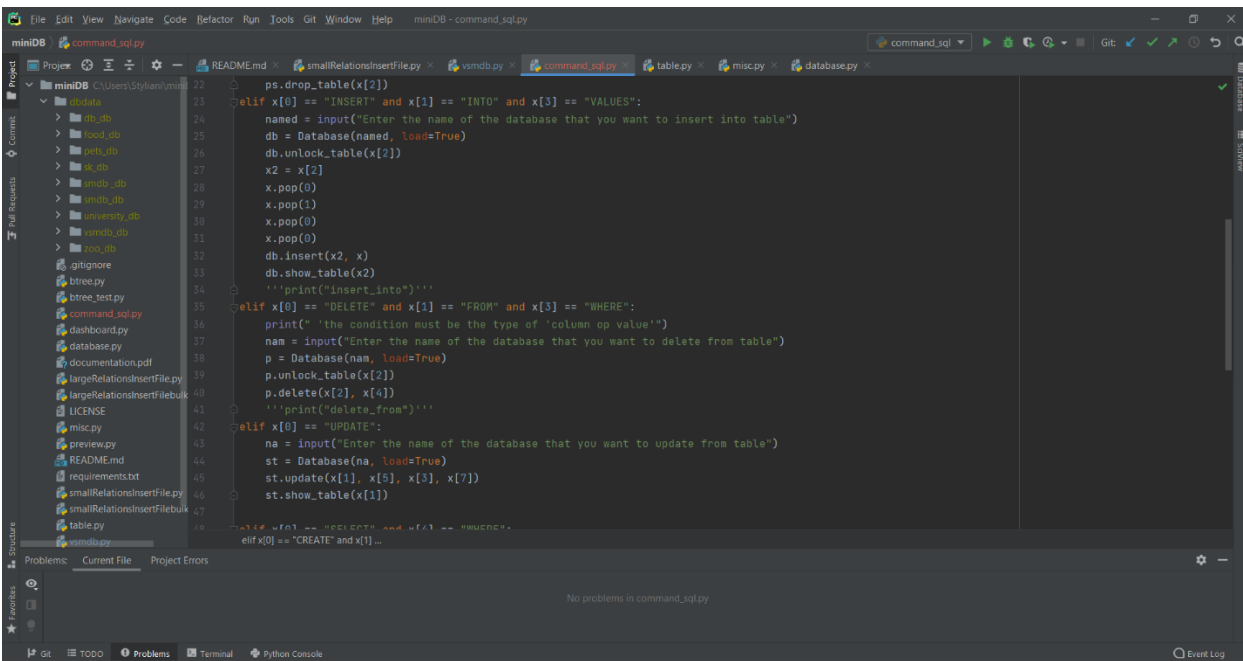
Για τα ερωτήματα μας αυτά έχει δημιουργηθεί το αρχείο `command_sql.py` που περιέχει τον εκτελέσιμο κώδικα για τα 8 ζητούμενα.



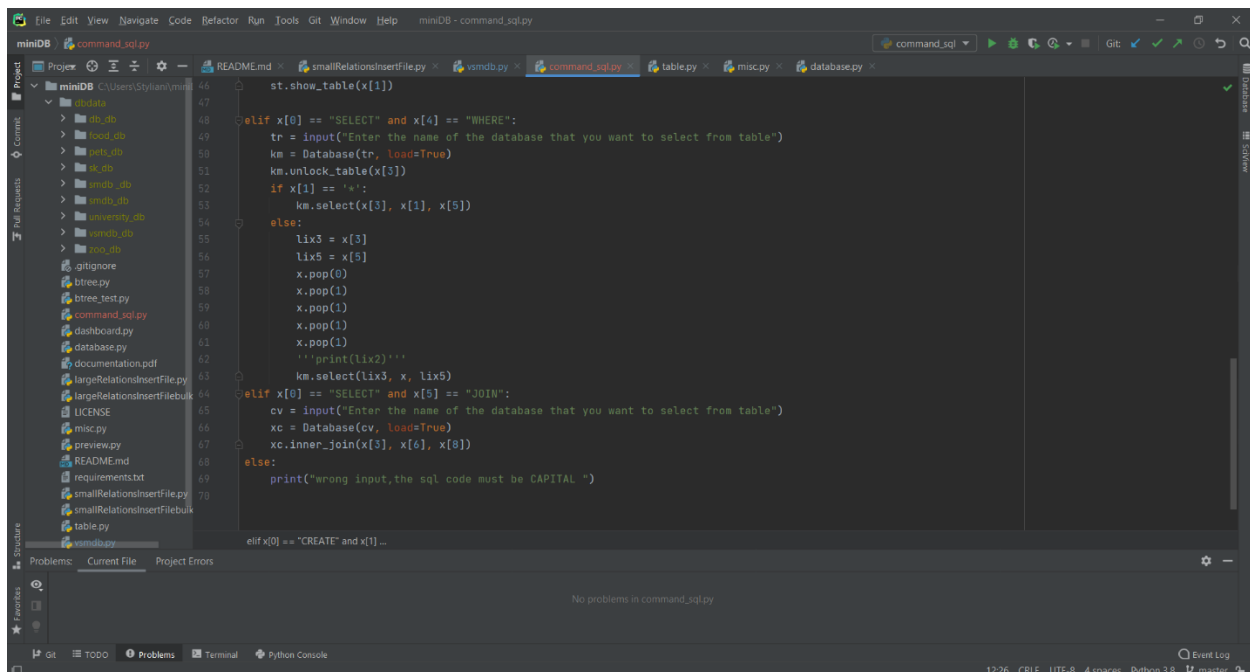
Κώδικας Αρχείου command_sql.py



```
1 from database import Database
2 code = input("Enter your SQL code:")
3 x = code.split()
4 if x[0] == "CREATE" and x[1] == "DATABASE":
5     print("create_database")
6     db_object = Database(x[2], load=False)
7 elif x[0] == "CREATE" and x[1] == "TABLE":
8     name = input("Enter the name of db that you want to create the table")
9     y = Database(name, load=True)
10    tk = len(x[3])
11    y.create_table(x[2], )
12    print("create_table")
13 elif x[0] == "CREATE" and x[1] == "INDEX":
14    nq = input("Enter the name of the database that you want to do the index")
15    q = Database(nq, load=True)
16    q.create_index(x[4], x[2])
17 elif x[0] == "DROP" and x[1] == "TABLE":
18    print("drop_table")
19    n = input("Enter the name of the database that you want to drop the table")
20    ps = Database(n, load=True)
21    ps.unlock_table(x[2])
22    ps.drop_table(x[2])
23 elif x[0] == "INSERT" and x[1] == "INTO" and x[3] == "VALUES":
24    named = input("Enter the name of the database that you want to insert into table")
25    db = Database(named, load=True)
26    db.unlock_table(x[2])
27    x = x[3:]
28    elif x[0] == "CREATE" and x[1] == "TABLE":
```



```
22 ps.drop_table(x[2])
23 elif x[0] == "INSERT" and x[1] == "INTO" and x[3] == "VALUES":
24    named = input("Enter the name of the database that you want to insert into table")
25    db = Database(named, load=True)
26    db.unlock_table(x[2])
27    x2 = x[2]
28    x.pop(0)
29    x.pop(1)
30    x.pop(0)
31    x.pop(0)
32    db.insert(x2, x)
33    db.show_table(x2)
34    print("insert_into")
35 elif x[0] == "DELETE" and x[1] == "FROM" and x[3] == "WHERE":
36    print("the condition must be the type of 'column op value'")
37    nam = input("Enter the name of the database that you want to delete from table")
38    p = Database(nam, load=True)
39    p.unlock_table(x[2])
40    p.delete(x[2], x[4])
41    print("delete_from")
42 elif x[0] == "UPDATE":
43    na = input("Enter the name of the database that you want to update from table")
44    st = Database(na, load=True)
45    st.update(x[1], x[5], x[3], x[7])
46    st.show_table(x[1])
47
48 # if x[0] == "SELECT" and x[1] == "WHERE":
49 #     elif x[0] == "CREATE" and x[1] == "TABLE":
```

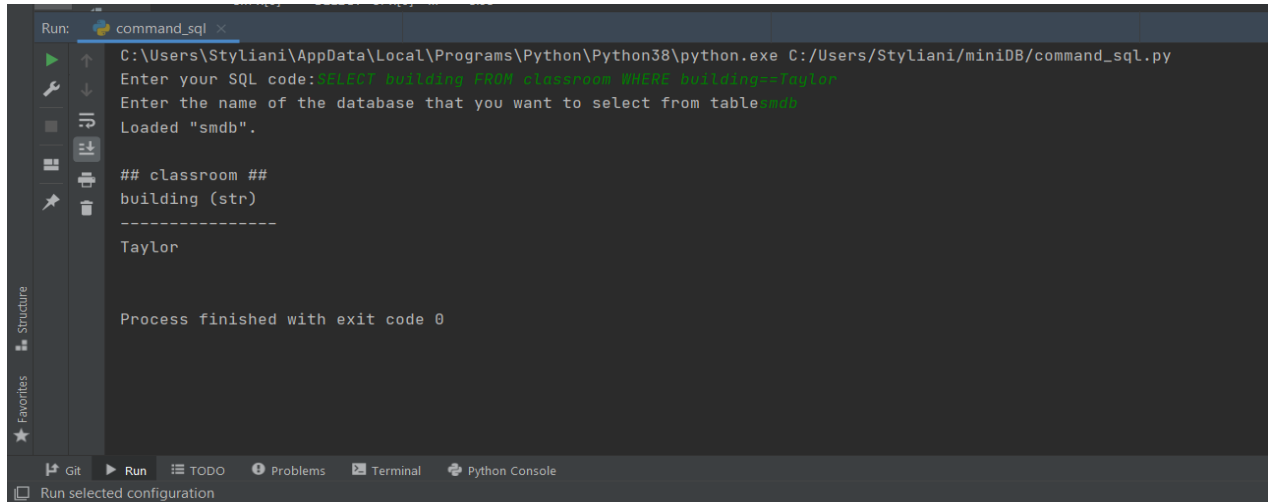


Λίγα λόγια για τον κώδικα του προγράμματος:

- Αρχικά κάνουμε split την εντολή της SQL που εισάγει ο χρήστης ,αν είναι λάθος εμφανίζει το κατάλληλο μήνυμα λάθους.
- Στην συνέχεια χωρίζονται οι εντολές ανάλογα με πιο θέλει ο χρήστης ελέγχουμε την 1 ή και την 2 λέξη για να αναγνωρίσουμε την εντολή
- Για την υλοποίηση των εντολών παίρνουμε τις μεταβλητές από το split που χωρίσαμε την εντολή που εισήγαγε ο χρήστης, για παράδειγμα η εντολή CREATE DATABASE pets , ελέγχουμε την πρώτη λέξη είναι το CREATE και παίρνουμε το x[2] που είναι το όνομα της καινούργιας βάσης δηλαδή pets,και τα εισάγουμε στον αντίστοιχο κώδικα της miniDB μας, db_object= Database(x[2], load=False), με τον ίδιο τρόπο έχουν υλοποιηθεί και άλλες εντολές.
- Στην περίπτωση των εντολών select from where και insert into έγινε η εξής τεχνική, για να απομονώσουμε τα rows, κάνουμε μια νέα λίστα με τα στοιχεία της εντολής που έδωσε ο χρήστης και αφαιρούμε με την μέθοδο pop() τις λέξεις INSERT,INTO,VALUES (στην περίπτωση του insert) και τα rows είναι αυτά που μένουν στην λίστα. Στο select from where γίνεται με την ίδια λογική για τα columns και αν είναι * παίρνει όλες τις στήλες χωρίς να χρειάζεται αυτή η διαδικασία.

Παραδείγματα σωστής εκτέλεσης κώδικα

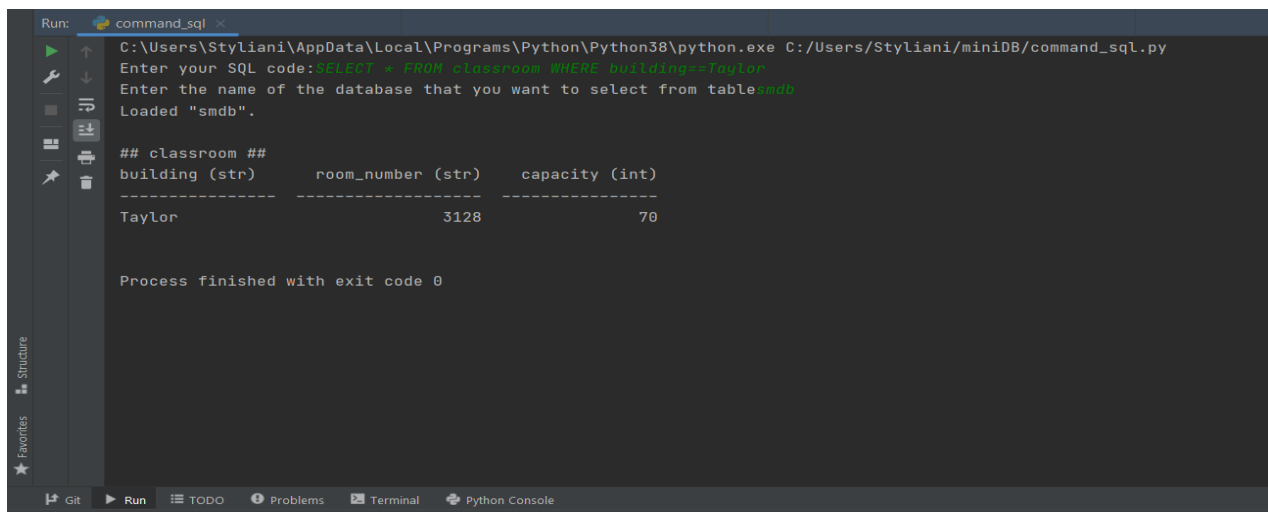
1. Select From Where



```
Run: command_sql x
C:\Users\Styliani\AppData\Local\Programs\Python\Python38\python.exe C:/Users/Styliani/miniDB/command_sql.py
Enter your SQL code: SELECT building FROM classroom WHERE building=Taylor
Enter the name of the database that you want to select from table: smdb
Loaded "smdb".

## classroom ##
building (str)
-----
Taylor

Process finished with exit code 0
```



```
Run: command_sql x
C:\Users\Styliani\AppData\Local\Programs\Python\Python38\python.exe C:/Users/Styliani/miniDB/command_sql.py
Enter your SQL code: SELECT * FROM classroom WHERE building=Taylor
Enter the name of the database that you want to select from table: smdb
Loaded "smdb".

## classroom ##
building (str)    room_number (str)    capacity (int)
-----
Taylor           3128                70

Process finished with exit code 0
```

2. Update Where

```
Run: command_sql x
C:\Users\Styliani\AppData\Local\Programs\Python\Python38\python.exe C:/Users/Styliani/miniDB/command_sql.py
Enter your SQL code:update advisor set i_ID = 26452 where i_ID=10101
Enter the name of the database that you want to update from tablesmdb
Loaded "smdb".

## advisor ##
s_ID (str) #PK#      i_ID (str)
-----
00128      45565
12345      26452
23121      76543
44553      22222
45678      22222
76543      45565
76653      98345
98765      98345
45800      63899
26782      34908
59596      51512

Process finished with exit code 0
```

3.Insert Into

```
Run: command_sql x
C:\Users\Styliani\AppData\Local\Programs\Python\Python38\python.exe C:/Users/Styliani/miniDB/command_sql.py
Enter your SQL code:INSERT INTO advisor (26782, 34908)
Enter the name of the database that you want to insert into tablesmdb
Loaded "smdb".
['26782', '34908']

## advisor ##
s_ID (str) #PK#      i_ID (str)
-----
00128      45565
12345      10101
23121      76543
44553      22222
45678      22222
76543      45565
76653      98345
98765      98345
into       12345
26782      34908

Process finished with exit code 0
```

```
Run: command_sql x
Enter your SQL code:insert into time_slot values A 7 12 0 14 30
Enter the name of the database that you want to insert into tablesmdb
Loaded "smdb".
['A', 'T', '12', '0', '14', '30']

## time_slot ##
time_slot_id (str)  day (str)      start_hr (int)  start_min (int)  end_hr (str)    end_min (str)
-----
A                   M                   8               0               8               50
A                   W                   8               0               8               50
A                   F                   8               0               8               50
B                   M                   9               0               9               50
B                   W                   9               0               9               50
B                   F                   9               0               9               50
C                   M                   11              0               11              50
C                   W                   11              0               11              50
C                   F                   11              0               11              50
D                   M                   13              0               13              50
D                   W                   13              0               13              50
D                   F                   13              0               13              50
E                   T                   10              30              11              45
E                   R                   10              30              11              45
F                   T                   14              30              15              45
F                   R                   14              30              15              45
G                   M                   16              0               16              50
G                   W                   16              0               16              50
G                   F                   16              0               16              50
H                   W                   10              0               12              30
A                   T                   12              0               14              30

Process finished with exit code 0
```

4.Delete From Where

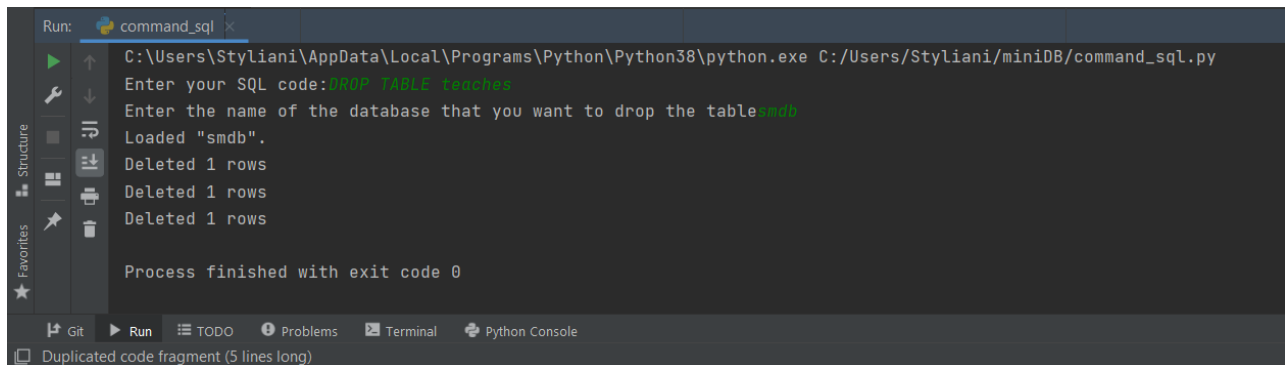
```
Run: command_sql x
C:\Users\Styliani\AppData\Local\Programs\Python\Python38\python.exe C:/Users/Styliani/miniDB/command_sql.py
Enter your SQL code:delete from advisor where i_ID==76766
'the condition must be the type of 'column op value'
Enter the name of the database that you want to delete from tablesmdb
Loaded "smdb".
Deleted 1 rows

Process finished with exit code 0
```

5.Create/Drop Table

Create Table

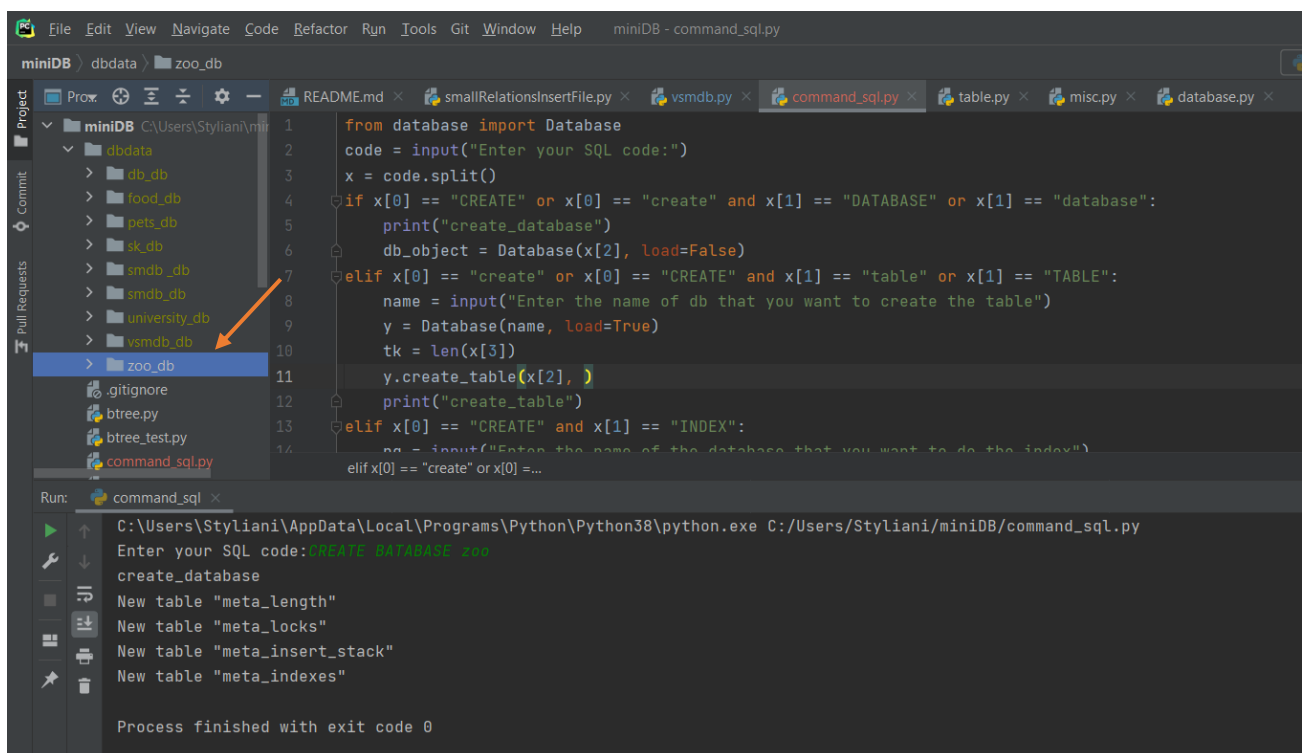
Drop Table



```
Run: command_sql x
C:\Users\Styliani\AppData\Local\Programs\Python\Python38\python.exe C:/Users/Styliani/miniDB/command_sql.py
Enter your SQL code: DROP TABLE teaches
Enter the name of the database that you want to drop the table: smdb
Loaded "smdb".
Deleted 1 rows
Deleted 1 rows
Deleted 1 rows

Process finished with exit code 0
```

6.Create Database



```
File Edit View Navigate Code Refactor Run Tools Git Window Help miniDB - command_sql.py
miniDB dbdata zoo_db
Project miniDB C:\Users\Styliani\miniDB
  dbdata
  db_db
  food_db
  pets_db
  sk_db
  smdb_db
  smdb_db
  university_db
  vsmdb_db
  zoo_db
  .gitignore
  btree.py
  btree_test.py
  command_sql.py
  README.md
  smallRelationsInsertFile.py
  vsmdb.py
  command_sql.py
  table.py
  misc.py
  database.py

1 from database import Database
2 code = input("Enter your SQL code:")
3 x = code.split()
4 if x[0] == "CREATE" or x[0] == "create" and x[1] == "DATABASE" or x[1] == "database":
5     print("create_database")
6     db_object = Database(x[2], load=False)
7 elif x[0] == "create" or x[0] == "CREATE" and x[1] == "table" or x[1] == "TABLE":
8     name = input("Enter the name of db that you want to create the table")
9     y = Database(name, load=True)
10    tk = len(x[3])
11    y.create_table(x[2], )
12    print("create_table")
13 elif x[0] == "CREATE" and x[1] == "INDEX":
14    name = input("Enter the name of the database that you want to do the index")
15 elif x[0] == "create" or x[0] == "CREATE":
16    pass

Run: command_sql x
C:\Users\Styliani\AppData\Local\Programs\Python\Python38\python.exe C:/Users/Styliani/miniDB/command_sql.py
Enter your SQL code: CREATE DATABASE zoo
create_database
New table "meta_length"
New table "meta_locks"
New table "meta_insert_stack"
New table "meta_indexes"

Process finished with exit code 0
```

7.Select From Inner Join


```
Run: command_sql x
C:\Users\Styliani\AppData\Local\Programs\Python\Python38\python.exe C:/Users/Styliani/miniDB/command_sql.py
Enter your SQL code:SELECT * FROM course INNER JOIN prereq ON course.course_id=prereq.course_id
Enter the name of the database that you want to select from table:advis
Loaded "smbd".

## course ##
course_id (str) #PK#    title (str)            dept_name (str)        credits (int)
-----
BI0-101                Intro. to Biology      Biology                4
BI0-301                Genetics               Biology                4
BI0-399                Computational Biology   Biology                3
CS-101                 Intro. to Computer Science Comp. Sci.            4
CS-190                 Game Design            Comp. Sci.            4
CS-315                 Robotics               Comp. Sci.            3
CS-319                 Image Processing        Comp. Sci.            3
CS-347                 Database System Concepts Comp. Sci.            3
EE-181                 Intro. to Digital Systems Elec. Eng.            3
FIN-201                Investment Banking      Finance                3
HIS-351                World History           History                3
MU-199                 Music Video Production  Music                 3
PHY-101                Physical Principles     Physics                4
```

8.Create Index On

```
Run: command_sql x
C:\Users\Styliani\AppData\Local\Programs\Python\Python38\python.exe C:/Users/Styliani/miniDB/command_sql.py
Enter your SQL code:CREATE INDEX ADS on advisor
Enter the name of the database that you want to do the index:smbd
Loaded "smbd".
Creating Btree index.

Process finished with exit code 0
|
```