
MiniDB Server Client Documentation

v.1.0 (Jan. 2021)

1. Παράδειγμα χρήσης Server-Client για την miniDB

Αρχικά μετακινούμαστε τον φάκελο του project. Για να χρησιμοποιηθεί κάποια από τις βάσεις της miniDB όπως η vsmdb πρέπει να τρέξουμε το αντίστοιχο αρχείο της python (πχ vsmdb.py) για να δημιουργηθεί η βάση με τα δεδομένα.

```
kontilenia@kontilenia:~$ cd github/miniDB
kontilenia@kontilenia:~/github/miniDB$ python3 vsmdb.py
New table "meta_length"
New table "meta_locks"
New table "meta_insert_stack"
New table "meta_indexes"
New table "classroom"
```

Εικόνα 1. Δημιουργία βάσης

Στην συνέχεια τρέχουμε πρώτα το αρχείο του server (server.py) όπου ζητάει το όνομα της βάσης που θέλουμε να τρέξουμε και μετά του client (client.py) σε ξεχωριστά terminal.

```
kontilenia@kontilenia: ~/github/miniDB
File Edit View Search Terminal Help
kontilenia@kontilenia:~$ cd github/miniDB
kontilenia@kontilenia:~/github/miniDB$ python3 server.py
Give me the database name:
vsmdb
Loaded "vsmdb".
socket binded to 8181
Connected by ('127.0.0.1', 53886)
```

Εικόνα 2. Άνοιγμα server

Στον client μπορούμε να τρέξουμε όλες τις εντολές με μορφή:

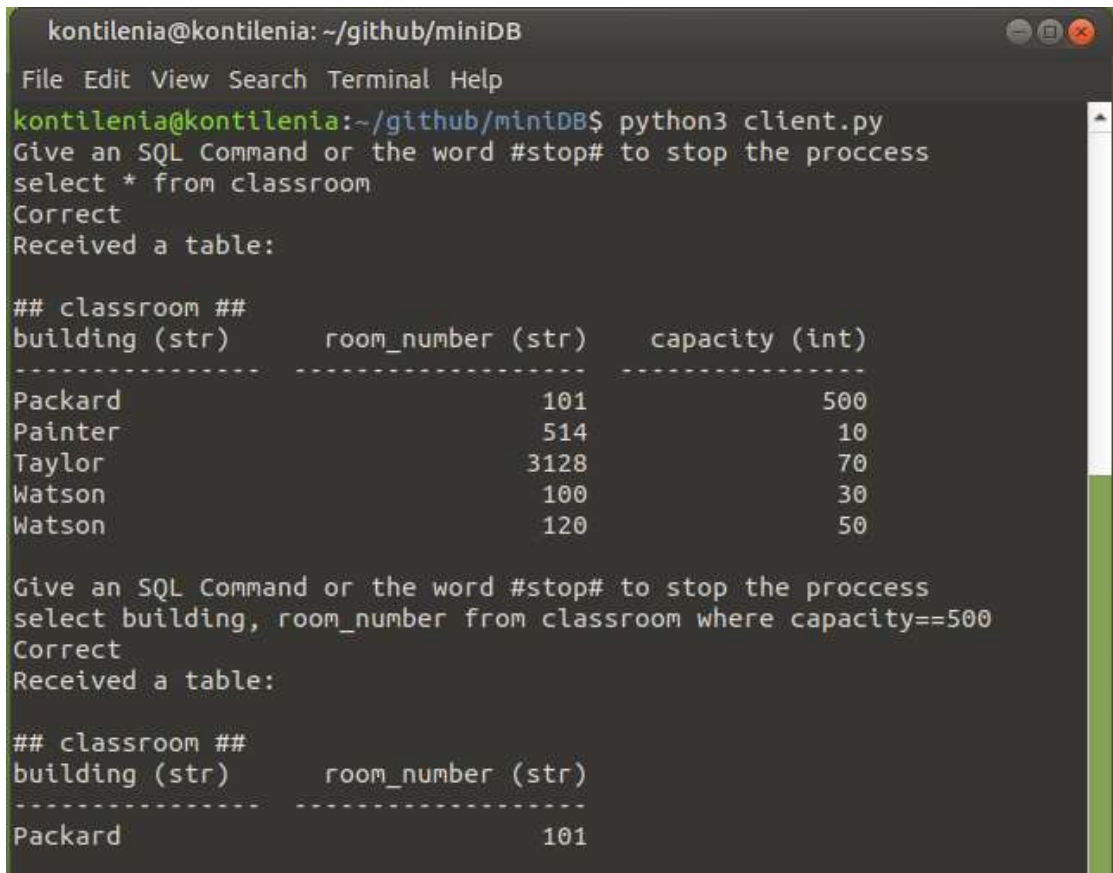
- select columns from table
- select columns from table where condition.

Το πεδίο `columns` πρέπει να περιέχει στήλες του πίνακα `table` χωρισμένες με κόμμα ή το σύμβολο `*`. Επίσης το πεδίο `condition` δέχεται όλες τις συνθήκες που έχουν υλοποιηθεί στην `miniDB`.

```
condition -> a condition using the following format :  
'column[<,<=,==,>=,>]value'
```

Εικόνα 3. Συνθήκες στην `miniDB`

Δεν έχει σημασία αν θα γραφτούν οι εντολές σε μικρά ή κεφαλαία γράμματα. Ωστόσο πρέπει να αφήνουμε ακριβώς ένα κενό από τις λέξεις `select`, `from`, `where` και οι στήλες να διαχωρίζονται από κόμμα με το πολύ ένα κενό. Διαφορετικά θα χτυπήσει ο `compiler`.



```
kontilenia@kontilenia: ~/github/miniDB  
File Edit View Search Terminal Help  
kontilenia@kontilenia:~/github/miniDB$ python3 client.py  
Give an SQL Command or the word #stop# to stop the proccess  
select * from classroom  
Correct  
Received a table:  
  
## classroom ##  
building (str)      room_number (str)      capacity (int)  
-----  
Packard              101                    500  
Painter              514                    10  
Taylor               3128                   70  
Watson               100                    30  
Watson               120                    50  
  
Give an SQL Command or the word #stop# to stop the proccess  
select building, room_number from classroom where capacity==500  
Correct  
Received a table:  
  
## classroom ##  
building (str)      room_number (str)  
-----  
Packard              101
```

Εικόνα 4. Άνοιγμα `client` και εκτέλεση δυο παραδειγμάτων

Τα αποτελέσματα που εμφανίζονται στον `server` είναι καθαρά για `debugging`.

```

1,room_number,building,@,classroom
-----
case1
The table has been sent
Closing the connection
Connected by ('127.0.0.1', 54024)
1,*,@,classroom
-----
case1
The table has been sent
2,building,room_number,@,classroom,@,capacity==500
-----
case2
The table has been sent

```

Εικόνα 5. Τα μηνύματα στον server

Σε περίπτωση που υπάρξει κάποιο συντακτικό λάθος της SQL εντολής ή αν υπάρχουν παραπάνω κενά στην εντολή θα χτυπήσει ο compiler.

```

select from classroom
Wrong Command Please try again!

```

Εικόνα 6. Συντακτικό λάθος

Αν υπάρχει λάθος σε στήλη, πίνακα ή συνθήκη ελέγχεται από τον server και επιστρέφεται μήνυμα λάθους.

```

select col from classroom
Correct
It seems there was a problem pulling the data from the database.
'col' is not in list
Give an SQL Command or the word #stop# to stop the proccess

```

Εικόνα 7. Μήνυμα λάθους εξαιτίας λάθος στήλης

Για να σταματήσει ο client πληκτρολογούμε stop. Όσο ο server παραμένει ανοιχτός μπορούμε να τρέξουμε και καινούριο client.

2. Σύντομη επεξήγηση κώδικα

Ο client και ο server συνδέονται μέσω μιας port με την βοήθεια socket όπου επικοινωνούν ανταλλάζοντας string και pickled αντικείμενα.

Ο compiler βρίσκεται μέσα στο αρχείο του client και ελέγχει αν η σύνταξη της εντολής select είναι σωστή μέσω regex.

Αν είναι τότε χωρίζουμε δύο περιπτώσεις. Αν η εντολή select έχει where (περίπτωση 2^η) ή όχι (περίπτωση 1^η). Ανάλογα με την περίπτωση στέλνεται και το string επικοινωνίας από τον client στον server το οποίο ξεκινάει με τον αριθμό της περίπτωσης. Στην συνέχεια του string αυτού βρίσκονται οι στήλες, το όνομα του πίνακα και η συνθήκη διαχωρισμένα με τον χαρακτήρα @. Αν υπάρχουν παραπάνω από μια στήλες χωρίζονται με κόμμα.

Ο server λαμβάνει το string και καλεί την συνάρτηση select με τα κατάλληλα ορίσματα. Το αποτέλεσμα γίνεται pickle και επιστρέφεται στον client (μπορεί να είναι exception ή πίνακας).

Ο client τέλος παίρνει το pickled αντικείμενο και το εμφανίζει αναλόγως.

Έχει δοθεί προσοχή ώστε να μην σταματήσει ο server σε περίπτωση που παρουσιαστεί exception είτε λόγο στήλης που δεν υπάρχει είτε πίνακα που δεν υπάρχει ή συνθήκης αλλά ταυτόχρονα και ο client να ενημερώνεται ποιο είναι το πρόβλημα. Επιπλέον σε περίπτωση λάθους ξεκλειδώνεται ο πίνακας που κλειδώθηκε για να μπορούν να συνεχιστούν τα προγράμματα.