

## Task 2.4: Server - Client

Ομάδα:

Κικίδης Δημήτριος, Π18073

Πέττα Αρετή, Π18231

Για το εξής ερώτημα έχουν υλοποιηθεί 2 server και 1 client σε 3 αρχεία συνολικά. Συνεπώς, τα αρχεία που προστέθηκαν συνολικά είναι τα 'server1.py', 'server2.py', 'client.py' και αυτό το pdf, δηλαδή το 'doc.pdf'.

---

### *Client*

---

Η λογική επικοινωνίας των πελατών με κάποιο server είναι η εξής:

1. Ο πελάτης κάνει αίτημα σύνδεσης στον server.
2. Ο server αποδέχεται τον πελάτη και περιμένει το ερώτημά του.  
Έπειτα εκτελούνται επαναληπτικά τα βήματα:
  - a. Ο πελάτης στέλνει το ερώτημά του και έναν non-printable χαρακτήρα που σηματοδοτεί το τέλος του ερωτήματός του.
  - b. Ο server εκτελεί το ερώτημα του πελάτη αφού το λάβει ολόκληρο και του στέλνει την απάντηση, πάλι με έναν non-printable χαρακτήρα στο τέλος, που σηματοδοτεί το τέλος της απάντησης. Έπειτα ο server περιμένει ξανά ερώτημα από τον πελάτη.

Η επανάληψη συνεχίζεται μέχρι ο πελάτης να στείλει τη λέξη 'exit'. Τότε ο server τερματίζει τη σύνδεσή του με τον πελάτη χωρίς να στείλει πίσω κάποια απάντηση, ενώ το πρόγραμμα του πελάτη τερματίζεται.

Τα υποστηριζόμενα query του πελάτη αυτή τη στιγμή είναι της μορφής 'select \* from {table\_name}', όπου {table\_name} το όνομα του πίνακα που θέλει να δει ο πελάτης, καθώς και το 'exit' για τον τερματισμό της σύνδεσης.

Η ροή εκτέλεσης του πελάτη έχει ως εξής: Ο πελάτης συνδέεται στον server. Έπειτα εκτελεί ένα άπειρο while loop που τερματίζεται αν δοθεί ως είσοδος η λέξη 'exit'. Σε κάθε είσοδο του χρήστη ελέγχεται ώστε η μορφή των ερωτημάτων να είναι σωστή και, αν είναι, στέλνεται το ερώτημα στον server με τον non-printable χαρακτήρα στο τέλος. Μετά την αποστολή του ερωτήματος γίνεται αναμονή για τη λήψη της απάντησης. Όταν ληφθεί η απάντηση, εκτυπώνεται το αποτέλεσμα αφού αφαιρεθεί ο τελευταίος non-printable χαρακτήρας και το πρόγραμμα περιμένει πάλι είσοδο από το χρήστη. Οι εντολές client.sendall() και client.recv() είναι blocking και έχουν ως αποτέλεσμα το πρόγραμμα να συνεχίζει τη ροή της εκτέλεσης του μόνο όταν έχει αποσταλεί ολόκληρο το ερώτημα και έχει ληφθεί κάποιο αποτέλεσμα αντίστοιχα. Για τον έλεγχο του αποτελέσματος εκτελείται ένα while loop μέχρι να ληφθεί ο non-printable χαρακτήρας. Με τη λέξη 'exit' τερματίζεται η σύνδεση και το πρόγραμμα του πελάτη.

Ο πελάτης είναι υλοποιημένος στο αρχείο 'client.py'.

Εκτέλεση πελάτη: python client.py (στο directory του client.py)

---

## Servers

---

Server 1: Είναι υλοποιημένος στο αρχείο 'server1.py'. Η ροή εκτέλεσής του έχει ως εξής: Αρχικά φορτώνει την επιλεγμένη βάση και αρχικοποιεί το socket του server με τις εντολές:

```
db = Database("vsmdb", load = True)
# Port number.
PORT = 5050
# Socket creation.
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# Bind server to IP and Port.
server.bind("", PORT)
# Start listening for requests.
server.listen(5)
print("\n[Server] Server started.")
print("\n[Server] Active clients: 0.")
```

Εδώ φορτώνεται η βάση 'vsmdb' αλλά με την αλλαγή του ορίσματος μπορεί να φορτωθεί οποιαδήποτε υπάρχουσα βάση της miniDB. Μετά ξεκινά η εκτέλεση ενός άπειρου while loop.

```
try:
    while True:
        # Accept a request.
        connection, address = server.accept()
        # Create a new thread that will execute the 'client' function.
        thread = threading.Thread(target = client, args = (connection, address))
        print(f"\n[Server] Connected to |{address[0]} : {address[1]}|.")
        print(f"\n[Server] Active clients: {threading.activeCount()}")
        # Start the new thread.
        thread.start()
except:
    server.close()
    print("\n[Server] Server stopped.")
```

Ο server περιμένει στην εντολή server.accept() μέχρι να έρθει request από κάποιον πελάτη, γιατί η εντολή αυτή κάνει block το main thread. Όταν έρθει ένα request, το main thread ξεμπλοκάρει και δημιουργεί ένα νέο thread για να εξυπηρετήσει τον πελάτη. Το νέο thread θα εκτελέσει τη συνάρτηση client του κώδικα με τα ορίσματα της σύνδεσης και της

διεύθυνσης του χρήστη. Το main thread ξεκινά το νέο thread και έπειτα το main thread μπλοκάρει πάλι στην εντολή server.accept(). Ο server θα συνεχίσει αυτή την επαναληπτική διαδικασία μέχρι να διακοπεί η λειτουργία του, πατώντας τα κουμπιά Ctrl + C, ενώ τότε θα εκτελεστεί το κομμάτι του except, κλείνοντας το socket του server.

Εκτέλεση server 1: python server1.py (στο directory του server1.py)

Server 2: Είναι υλοποιημένος στο αρχείο 'server2.py'. Η ροή εκτέλεσής του έχει ως εξής: Αρχικά φορτώνει την επιλεγμένη βάση και αρχικοποιεί το socket του server με τις εντολές:

```
# Database to load.
db = Database("vsmdb", load = True)
# Port number.
PORT = 5050
# Socket creation.
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# Make the server socket non-blocking.
server.setblocking(0)
# Bind server to IP and Port.
server.bind("", PORT)
# Start listening for requests.
server.listen(5)
print("\n[Server] Server started.")
print("\n[Server] Active clients: 0.")
```

Εδώ φορτώνεται η βάση 'vsmdb' αλλά με την αλλαγή του ορίσματος μπορεί να φορτωθεί οποιαδήποτε υπάρχουσα βάση της miniDB.

Έπειτα γίνεται η αρχικοποίηση των 2 λιστών:

```
inputs = [server]
# The list of all the
outputs = []
```

Η λίστα inputs περιέχει αρχικά τον server και έπειτα προστίθεται για αρχή σε αυτήν κάθε πελάτης όταν συνδέεται. Εκτός από τον server, όλα τα άλλα sockets των πελατών μπαίνουν εδώ όταν θέλουμε να διαβάσουμε από αυτά, για αυτό και στην αρχή εισάγονται σε αυτή τη λίστα. Για τον server, σημαίνει ότι θέλουμε να κάνουμε αποδοχή πελατών. Όταν θέλουμε να στείλουμε απάντηση στους πελάτες,

αφαιρούνται από αυτή τη λίστα και εισάγονται στη λίστα outputs, η οποία περιέχει τα sockets των πελατών στα οποία θέλουμε να στείλουμε απάντηση. Όταν τους έχουμε στείλει την απάντηση και περιμένουμε πάλι να στείλουν ένα ερώτημα, μετακινούνται πάλι στη λίστα inputs.

Μετά αρχικοποιούνται και κάποια dictionaries με πληροφορίες για τους συνδεδεμένους πελάτες:

```
queries = {}  
# The dictionary of queries  
results = {}  
# The dictionary of results  
addresses = {}
```

Στη συνέχεια ξεκινά η εκτέλεση ενός άπειρου while loop. Μέσα σε αυτό πρώτα εκτελείται η εντολή:

```
readable, writable, exceptional = select.select(  
    inputs, outputs, inputs + outputs, 60)
```

Η οποία δέχεται ως ορίσματα τη λίστα των sockets από τα οποία ο server θέλει να διαβάσει(ή τον server για τον αποδοχή ενός πελάτη), τη λίστα των sockets στα τα οποία ο server θέλει να στείλει απάντηση και τη λίστα των socket που μπορεί να έχουν ένα exception(δηλαδή δίνονται όλα τα sockets, αφού δε ξέρουμε σε ποιο μπορεί να υπάρχει κάποιο exception). Το αποτέλεσμα που επιστρέφει είναι η λίστα readable που είναι τα έτοιμα sockets της λίστας inputs, η λίστα writable που είναι τα έτοιμα sockets της λίστας outputs και η λίστα exceptional που είναι όλα τα sockets που έχουν όντως κάποιο exception. Το τελευταίο όρισμα είναι το timeout 60 δευτερολέπτων, διότι η εντολή αυτή είναι blocking, οπότε αν σε 1 λεπτό δεν έχει επιστραφεί το αποτέλεσμα, επιστρέφονται 3 κενές λίστες και εκτελείται πάλι η ίδια εντολή αφού βρίσκεται στο while loop.

Η εντολή `server.setblocking(0)` κάνει το socket του server non-blocking στην αποδοχή του πελάτη(`s.accept()`).

```
connection, address = s.accept()
# Make the new socket non-blocking.
connection.setblocking(0)
```

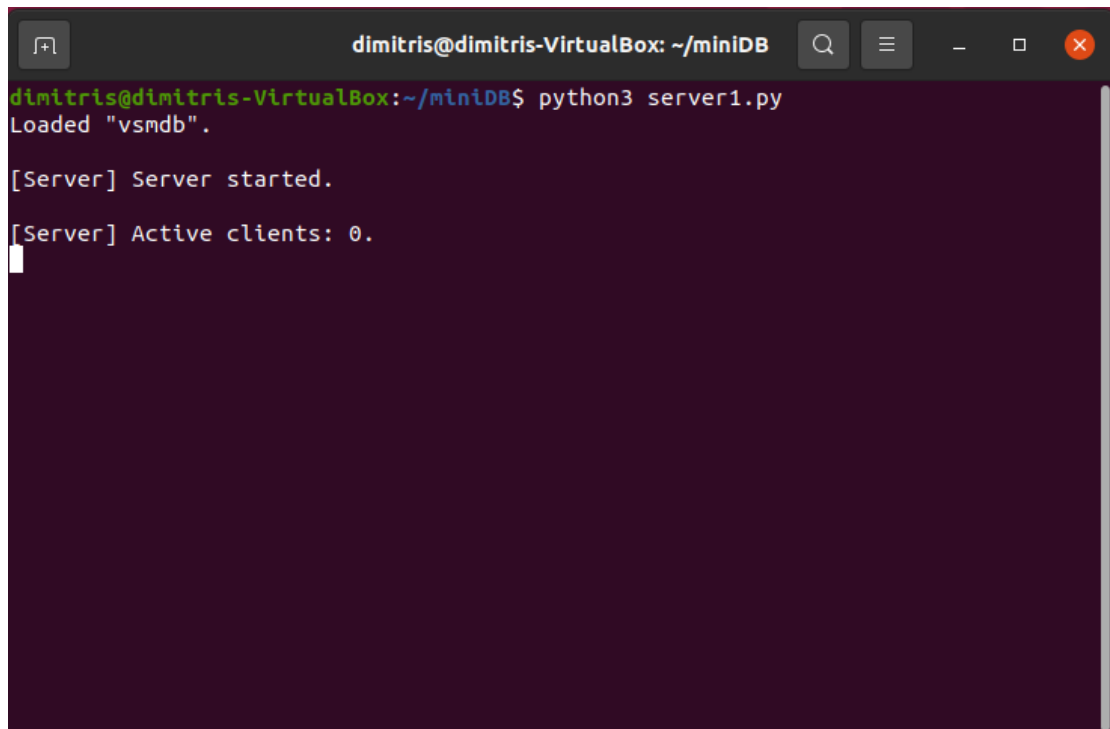
Η εντολή `connection.setblocking(0)` κάνει το socket του πελάτη που αποδέχτηκε ο server non-blocking, δηλαδή όταν ο server διαβάζει από αυτό ή γράφει σε αυτό το socket, δεν μπλοκάρει το main thread. Στην περίπτωση της αποστολής στο socket σημαίνει ότι μπορεί να σταλθούν λιγότερα byte από το μέγιστο αριθμό byte που μπορούν να σταλθούν σε μια εκτέλεση της εντολής `send` (το όριο είναι τα 512 byte στο πρόγραμμα). Στην περίπτωση διαβάσματος από το socket σημαίνει ότι μπορεί να διαβαστούν λιγότερα byte από το μέγιστο αριθμό byte που μπορούν να διαβαστούν σε μια εκτέλεση της εντολής `recv` (το όριο είναι τα 512 byte στο πρόγραμμα).

Σε κάθε επανάληψη του while loop, μετά τον καθορισμό των λιστών στις οποίες ο server μπορεί να γράψει ή να διαβάσει, αντίστοιχα για την κάθε λίστα σε κάθε socket της γίνεται αποστολή ή λήψη κάποιων δεδομένων. Αντίστοιχα τερματίζεται η σύνδεση με όσα sockets βρίσκονται στη λίστα `exceptional`.

Ο server θα συνεχίσει αυτή την επαναληπτική διαδικασία μέχρι να διακοπεί η λειτουργία του, πατώντας τα κουμπιά `Ctrl + C`, ενώ τότε θα εκτελεστεί το κομμάτι του `except`, κλείνοντας το socket του server.

Εκτέλεση server 2: `python server2.py` (στο directory του `server2.py`)

Παράδειγμα σύνδεσης δύο πελατών στον server 1:

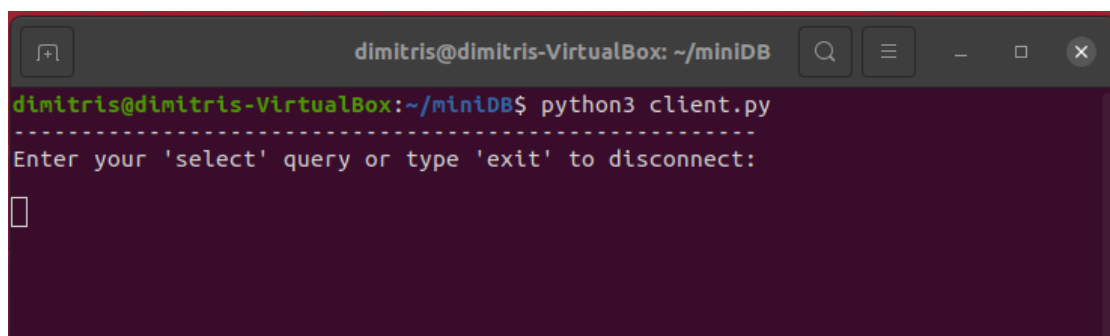
A terminal window titled 'dimitris@dimitris-VirtualBox: ~/miniDB'. The user has executed 'python3 server1.py'. The output shows 'Loaded "vsmdb".', '[Server] Server started.', and '[Server] Active clients: 0.'. A cursor is visible on the line following the last message.

```
dimitris@dimitris-VirtualBox: ~/miniDB
dimitris@dimitris-VirtualBox:~/miniDB$ python3 server1.py
Loaded "vsmdb".

[Server] Server started.

[Server] Active clients: 0.
```

Server 1

A terminal window titled 'dimitris@dimitris-VirtualBox: ~/miniDB'. The user has executed 'python3 client.py'. The output shows a dashed line separator, the prompt 'Enter your 'select' query or type 'exit' to disconnect:', and a cursor on the line below.

```
dimitris@dimitris-VirtualBox: ~/miniDB
dimitris@dimitris-VirtualBox:~/miniDB$ python3 client.py
-----
Enter your 'select' query or type 'exit' to disconnect:


```

Client 1

A terminal window titled 'dimitris@dimitris-VirtualBox: ~/miniDB'. The user has executed 'python3 server1.py'. The output shows 'Loaded "vsmdb".', '[Server] Server started.', '[Server] Active clients: 0.', '[Server] Connected to |127.0.0.1 : 55090|.', and '[Server] Active clients: 1.'. A cursor is visible on the line following the last message.

```
dimitris@dimitris-VirtualBox: ~/miniDB
dimitris@dimitris-VirtualBox:~/miniDB$ python3 server1.py
Loaded "vsmdb".

[Server] Server started.

[Server] Active clients: 0.

[Server] Connected to |127.0.0.1 : 55090|.

[Server] Active clients: 1.
```

Server 1

```
dimitris@dimitris-VirtualBox: ~/miniDB
dimitris@dimitris-VirtualBox:~/miniDB$ python3 client.py
-----
Enter your 'select' query or type 'exit' to disconnect:

select * from classroom

## classroom ##
building (str)      room_number (str)  capacity (int)
-----
Packard              101                500
Painter              514                 10
Taylor               3128                70
Watson               100                 30
Watson               120                 50

-----
Enter your 'select' query or type 'exit' to disconnect:
█
```

Client 1

```
dimitris@dimitris-VirtualBox: ~/miniDB
dimitris@dimitris-VirtualBox:~/miniDB$ python3 server1.py
Loaded "vsmdb".

[Server] Server started.
[Server] Active clients: 0.
[Server] Connected to |127.0.0.1 : 55090|.
[Server] Active clients: 1.
[Server] Received the query 'select * from classroom' from |127.0.0.1 : 55090|.
[Server] Sent the result for the query 'select * from classroom' to |127.0.0.1 : 55090|.
█
```

Server 1

```
dimitris@dimitris-VirtualBox: ~/miniDB
dimitris@dimitris-VirtualBox:~/miniDB$ python3 client.py
-----
Enter your 'select' query or type 'exit' to disconnect:
█
```

Client 2



```
dimitris@dimitris-VirtualBox: ~/miniDB
dimitris@dimitris-VirtualBox:~/miniDB$ python3 server1.py
Loaded "vsmdb".

[Server] Server started.

[Server] Active clients: 0.

[Server] Connected to |127.0.0.1 : 55090|.

[Server] Active clients: 1.

[Server] Received the query 'select * from classroom' from |127.0.0.1 : 55090|.

[Server] Sent the result for the query 'select * from classroom' to |127.0.0.1 : 55090|.

[Server] Connected to |127.0.0.1 : 55092|.

[Server] Active clients: 2.
█
```

## Server 1

```
dimitris@dimitris-VirtualBox: ~/miniDB
dimitris@dimitris-VirtualBox:~/miniDB$ python client.py
-----
Enter your 'select' query or type 'exit' to disconnect:

select * from classroom

## classroom ##
building (str)      room_number (str)  capacity (int)
-----
Packard              101                500
Painter              514                 10
Taylor               3128                70
Watson                100                 30
Watson                120                 50

-----
Enter your 'select' query or type 'exit' to disconnect:
█
```

## Client 2

```
dimitris@dimitris-VirtualBox: ~/miniDB
Loaded "vsmdb".
[Server] Server started.
[Server] Active clients: 0.
[Server] Connected to |127.0.0.1 : 55090|.
[Server] Active clients: 1.
[Server] Received the query 'select * from classroom' from |127.0.0.1 : 55090|.
[Server] Sent the result for the query 'select * from classroom' to |127.0.0.1 : 55090|.
[Server] Connected to |127.0.0.1 : 55092|.
[Server] Active clients: 2.
[Server] Received the query 'select * from classroom' from |127.0.0.1 : 55092|.
[Server] Sent the result for the query 'select * from classroom' to |127.0.0.1 : 55092|.
█
```

## Server 1

```
dimitris@dimitris-VirtualBox: ~/miniDB
select * from classroom

## classroom ##
building (str)      room_number (str)  capacity (int)
-----
Packard              101                 500
Painter              514                  10
Taylor               3128                 70
Watson                100                  30
Watson                120                  50

-----
Enter your 'select' query or type 'exit' to disconnect:
exit

Disconnected from the server.
dimitris@dimitris-VirtualBox:~/miniDB$ █
```

## Client 1

```
dimitris@dimitris-VirtualBox: ~/miniDB
## classroom ##
building (str)    room_number (str)    capacity (int)
-----
Packard           101                  500
Painter           514                  10
Taylor            3128                 70
Watson            100                  30
Watson            120                  50

-----
Enter your 'select' query or type 'exit' to disconnect:
exit

Disconnected from the server.
dimitris@dimitris-VirtualBox:~/miniDB$
```

## Client 2

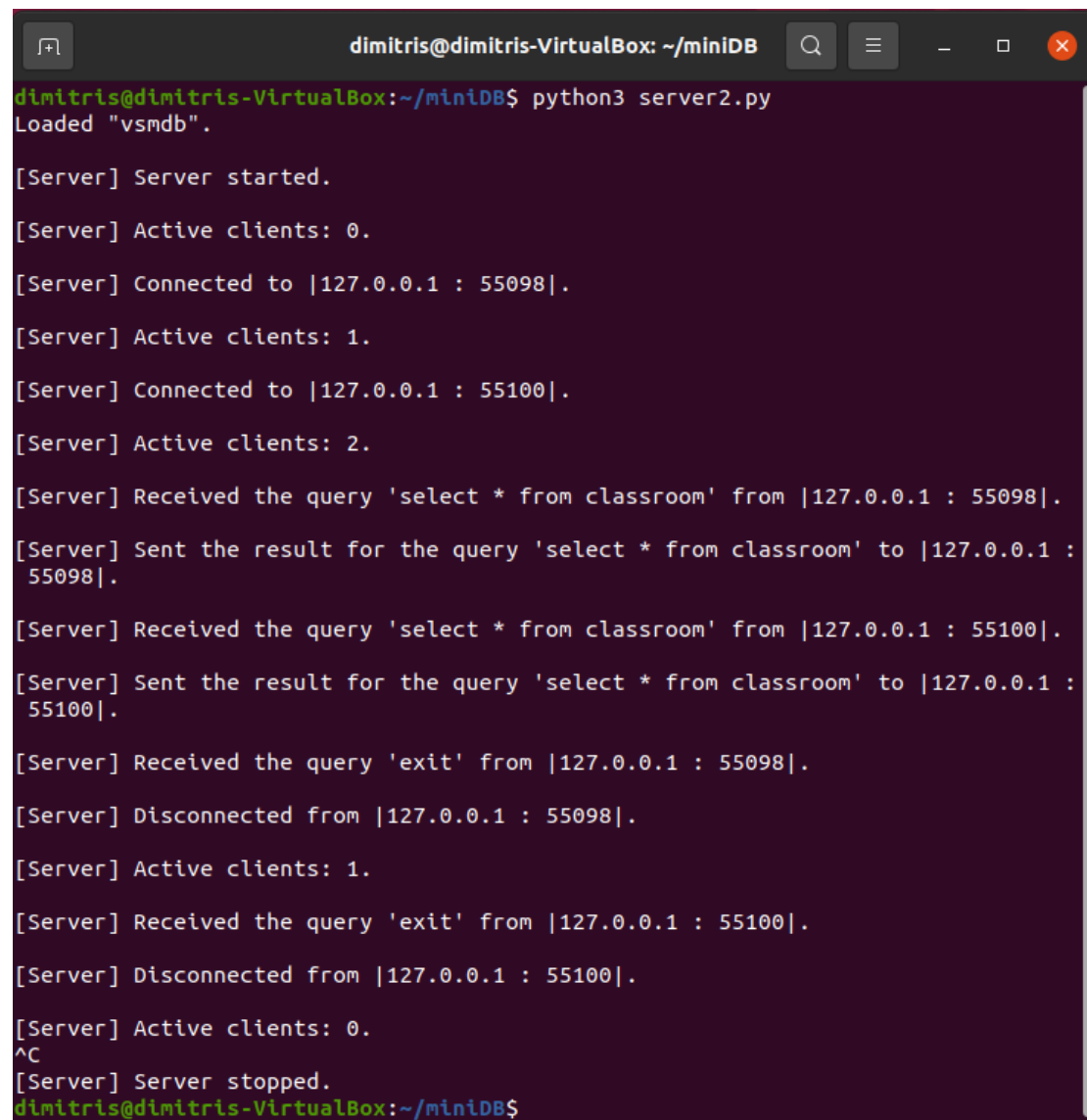
```
dimitris@dimitris-VirtualBox: ~/miniDB
[Server] Sent the result for the query 'select * from classroom' to |127.0.0.1 : 55090|.
[Server] Connected to |127.0.0.1 : 55092|.
[Server] Active clients: 2.
[Server] Received the query 'select * from classroom' from |127.0.0.1 : 55092|.
[Server] Sent the result for the query 'select * from classroom' to |127.0.0.1 : 55092|.
[Server] Received the query 'exit' from |127.0.0.1 : 55090|.
[Server] Disconnected from |127.0.0.1 : 55090|.
[Server] Active clients: 1.
[Server] Received the query 'exit' from |127.0.0.1 : 55092|.
[Server] Disconnected from |127.0.0.1 : 55092|.
[Server] Active clients: 0.
```

## Server 1

```
dimitris@dimitris-VirtualBox: ~/miniDB
[Server] Connected to |127.0.0.1 : 55092|.
[Server] Active clients: 2.
[Server] Received the query 'select * from classroom' from |127.0.0.1 : 55092|.
[Server] Sent the result for the query 'select * from classroom' to |127.0.0.1 : 55092|.
[Server] Received the query 'exit' from |127.0.0.1 : 55090|.
[Server] Disconnected from |127.0.0.1 : 55090|.
[Server] Active clients: 1.
[Server] Received the query 'exit' from |127.0.0.1 : 55092|.
[Server] Disconnected from |127.0.0.1 : 55092|.
[Server] Active clients: 0.
^C
[Server] Server stopped.
dimitris@dimitris-VirtualBox:~/miniDB$
```

Server 1

Αντίστοιχα λειτουργεί και ο server 2.

A terminal window titled 'dimitris@dimitris-VirtualBox: ~/miniDB' with standard window controls. The terminal shows the execution of 'python3 server2.py'. The output includes: 'Loaded "vsmdb".', '[Server] Server started.', '[Server] Active clients: 0.', '[Server] Connected to |127.0.0.1 : 55098|.', '[Server] Active clients: 1.', '[Server] Connected to |127.0.0.1 : 55100|.', '[Server] Active clients: 2.', '[Server] Received the query 'select \* from classroom' from |127.0.0.1 : 55098|.', '[Server] Sent the result for the query 'select \* from classroom' to |127.0.0.1 : 55098|.', '[Server] Received the query 'select \* from classroom' from |127.0.0.1 : 55100|.', '[Server] Sent the result for the query 'select \* from classroom' to |127.0.0.1 : 55100|.', '[Server] Received the query 'exit' from |127.0.0.1 : 55098|.', '[Server] Disconnected from |127.0.0.1 : 55098|.', '[Server] Active clients: 1.', '[Server] Received the query 'exit' from |127.0.0.1 : 55100|.', '[Server] Disconnected from |127.0.0.1 : 55100|.', '[Server] Active clients: 0.', '^C', '[Server] Server stopped.', and the prompt 'dimitris@dimitris-VirtualBox:~/miniDB\$'.

```
dimitris@dimitris-VirtualBox: ~/miniDB
dimitris@dimitris-VirtualBox:~/miniDB$ python3 server2.py
Loaded "vsmdb".

[Server] Server started.

[Server] Active clients: 0.

[Server] Connected to |127.0.0.1 : 55098|.

[Server] Active clients: 1.

[Server] Connected to |127.0.0.1 : 55100|.

[Server] Active clients: 2.

[Server] Received the query 'select * from classroom' from |127.0.0.1 : 55098|.

[Server] Sent the result for the query 'select * from classroom' to |127.0.0.1 : 55098|.

[Server] Received the query 'select * from classroom' from |127.0.0.1 : 55100|.

[Server] Sent the result for the query 'select * from classroom' to |127.0.0.1 : 55100|.

[Server] Received the query 'exit' from |127.0.0.1 : 55098|.

[Server] Disconnected from |127.0.0.1 : 55098|.

[Server] Active clients: 1.

[Server] Received the query 'exit' from |127.0.0.1 : 55100|.

[Server] Disconnected from |127.0.0.1 : 55100|.

[Server] Active clients: 0.
^C
[Server] Server stopped.
dimitris@dimitris-VirtualBox:~/miniDB$
```

Server 2(μετά το τέλος όλης της εκτέλεσης)

```
dimitris@dimitris-VirtualBox: ~/miniDB
dimitris@dimitris-VirtualBox:~/miniDB$ python3 client.py
-----
Enter your 'select' query or type 'exit' to disconnect:

select * from classroom

## classroom ##
building (str)      room_number (str)  capacity (int)
-----
Packard              101                500
Painter              514                 10
Taylor               3128                70
Watson               100                 30
Watson               120                 50

-----
Enter your 'select' query or type 'exit' to disconnect:

exit

Disconnected from the server.
dimitris@dimitris-VirtualBox:~/miniDB$
```

Client 1(μετά το τέλος όλης της εκτέλεσης)

```
dimitris@dimitris-VirtualBox: ~/miniDB
dimitris@dimitris-VirtualBox:~/miniDB$ python3 client.py
-----
Enter your 'select' query or type 'exit' to disconnect:

select * from classroom

## classroom ##
building (str)      room_number (str)  capacity (int)
-----
Packard              101                500
Painter              514                 10
Taylor               3128                70
Watson               100                 30
Watson               120                 50

-----
Enter your 'select' query or type 'exit' to disconnect:

exit

Disconnected from the server.
dimitris@dimitris-VirtualBox:~/miniDB$
```

Client 2(μετά το τέλος όλης της εκτέλεσης)