

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ  
Τμήμα Πληροφορικής



Εργασία Μαθήματος  
**Συστήματα Διαχείρισης Βάσεων Δεδομένων**

Αριθμός εργασίας – Τίτλος εργασίας	<i>miniDB-Τελική εργασία</i>
Όνομα ομάδας:	<i>WebFlexers</i>
Μέλη ομάδας:	Μιχάλης Στυλιανίδης – Π19165 Λευτέρης Κοντούρης – Π19077 Κωνσταντίνος Καλογερόπουλος – Π19057
Ημερομηνία παράδοσης	13/2/2022



## Εκφώνηση εργασίας

Να υλοποιήσετε τα παρακάτω ερωτήματα:

- *NOT NULL, UNIQUE constraints (+ Btree functionality) - 30/50 #79*

Add support for not null and unique columns in CREATE TABLE command. For unique columns, add Btree functionality (already there but currently available for primary key only)

- *INSERT INTO table SELECT ... - 30/50 #78*

Add support for massive insertion of rows (results of a SELECT query) in a table.



## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1	Εισαγωγή .....	4
2	Περιγραφή του προγράμματος .....	4
2.1	Not null, unique περιορισμοί(constraints) στις στήλες ενός πίνακα.....	4
2.2	Insert into <i>table1</i> select [ <i>columns</i> ] from <i>table2</i> where ( <i>condition</i> ) .....	4
3	Επίδειξη της λύσης .....	5
3.1	Not null, unique περιορισμοί(constraints) στις στήλες ενός πίνακα.....	5
3.2	Insert into <i>table2</i> select ( <i>condition</i> ) from <i>table1</i> where ( <i>condition</i> ).....	5



## 1 Εισαγωγή

Το miniDB είναι ένα project γραμμένο σε python, το οποίο προσομοιάζει τις λειτουργίες μίας βάσης δεδομένων. Επιτρέπει την ταχεία ανάπτυξη και δοκιμή λειτουργιών πάνω σε αυτή.

## 2 Περιγραφή του προγράμματος

### 2.1 Not null, unique περιορισμοί(constraints) στις στήλες ενός πίνακα

Για την υλοποίηση των παραπάνω περιορισμών, δημιουργήθηκε η κλάση ColumnConstraints η οποία περιέχει μια λίστα με τα ονόματα των στηλών ενός πίνακα για το not\_null και μια αντίστοιχη για το unique. Αρχικά στο [mdb.py](#) εντοπίζονται οι στήλες που περιέχουν περιορισμούς και τα όνομα τους αποθηκεύονται στην αντίστοιχη λίστα που αναφέρθηκε πιο πάνω. Το αντικείμενο της κλάσης ColumnConstraints αποθηκεύεται στο λεξικό(γραμμή 120).

Στην συνέχεια, για την υποστήριξη των νέων περιορισμών τροποποιήθηκε η μέθοδος create\_table([database.py](#), γραμμές 103-105 ). Προστέθηκε το αντικείμενο *column\_constraints* ως όρισμα και ενημερώθηκε ο constructor της κλάσης Table με τα νέα δεδομένα.

Επίσης, στον constructor του [table.py](#) έγινε χρήση της μεθόδου generate\_dictionary() της κλάσης ColumnConstraints, η οποία δημιουργεί ένα λεξικό από τα δεδομένα του αντικειμένου. Αυτό γίνεται επειδή οι κλάση ColumnConstraints δεν είναι σειριοποιήσιμη(serializable).

Ενημερώθηκαν οι μέθοδοι insert και update(υπεύθυνες για την εισαγωγή και τροποποίηση δεδομένων μέσα σε έναν πίνακα), ώστε να υποστηρίζουν τους νέους περιορισμούς. Για την διευκόλυνση ανάγνωσης του κώδικα, δημιουργήθηκε η μέθοδος check\_constraints, η οποία εκτελεί τον έλεγχο εγκυρότητας των δεδομένων που πρόκειται να εισαχθούν σύμφωνα με τους περιορισμούς που υπάρχουν σε κάθε στήλη.

### 2.2 Insert into *table1* select [*columns*] from *table2* where (*condition*)

Αρχικά, γίνεται ταυτοποίηση([mdb.py](#), γραμμές140-159) του 'insert\_into\_select' query από τον interpret. Δημιουργείται ένα dictionary με τα στοιχεία της ερώτησης(query). Αν δεν δηλωθούν οι στήλες στις οποίες θα εισαχθούν δεδομένα, τότε επιλέγονται αυτόματα όλες.

Έπειτα, καλείται η συνάρτηση insert\_into\_select() [[database.py](#), γραμμή 278-320] με ορίσματα τις τιμές του λεξικού. Η συνάρτηση αρχικά παίρνει τα δεδομένα από τον δεύτερο πίνακα(table2) μέσω ενός select query. Στην συνέχεια δημιουργεί τη λίστα temp, η οποία εκφράζει τη θέση της κάθε στήλης(column) του πρώτου πίνακα(table1) που έδωσε ο χρήστης.



Για παράδειγμα, έστω οι στήλες που έδωσε ο χρήστης `target_cols=[z,y]` και έστω `home_cols=[x,y,z,k]` οι στήλες του αρχικού πίνακα. Τότε η λίστα αυτή θα είναι η `temp = [null, 1, 0, null]`, όπου 1 η θέση του στοιχείου `y` στην λίστα `home_cols`. Το `null`, υποδηλώνει ότι το στοιχείο δεν υπάρχει στην λίστα που δόθηκε και θα περαστεί ως `null`.

Επομένως, όταν θα έρθει η στιγμή να περαστεί μία γραμμή στοιχείων(`row`) από τον δεύτερο πίνακα(`table2`) στον πρώτο(`table1`), αν τα στοιχεία του δεν είναι στη σωστή σειρά, μετά το πέρας της επανάληψης θα είναι([database.py](#), γραμμές 310-317).

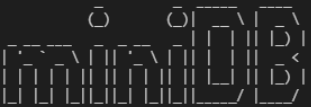
Παραδείγματος χάρη, δεδομένου ότι ο αρχικός πίνακας έχει στήλες της μορφής `[id,name,salary]` για να εισαχθεί το `row ['leftieris', '2525']`, όπου `leftieris=name` και `2525=id`, πρέπει να έρθει σε μορφή `[2525, leftieris, null]`. Αυτό επιτυγχάνεται μέσω της `temp` και άρα η τελευταία γραμμή γίνεται `[data[temp[1]], data[temp[0]], null]`, όπου `data` είναι τα στοιχεία(`values`) της κάθε γραμμής που εισάγονται.

### 3 Επίδειξη της λύσης

#### 3.1 Not null, unique περιορισμοί(constraints) στις στήλες ενός πίνακα

```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Michalis\Documents\GitHub\miniDB> & C:/Users/Michalis/AppData/Local/Programs/Python/Python39/python.exe c:/Users/Michalis/Documents/GitHub/miniDB/mdb.py
None
INFO:root:Loaded "None".

 2021 - v3.2

(None)> .cdb new
INFO:root:Loaded "new".
(new)> create table person(id int primary key, first_name str not null, last_name str not null unique, afm int unique)
Created table "person".
(new)> insert into person values(1, Jack, Pelter, 136)
(new)> insert into person values(2, Jordan, Peterson, 246)
(new)> insert into person values(3, null, Peterson, 246)
INFO:root:Tried to add a null value into a not null column
INFO:root:ABORTED
(new)> insert into person values(3, Jack, Peterson, 246)
INFO:root:Tried to add a duplicate value into a unique column
INFO:root:ABORTED
(new)> insert into person values(3, Jack, , 246)
INFO:root:Tried to add a null value into a not null column
INFO:root:ABORTED
(new)> insert into person values(3, Jack, null, 246)
INFO:root:Tried to add a null value into a not null column
INFO:root:ABORTED
```

#### 3.2 Insert into `table2` select (`condition`) from `table1` where (`condition`)

```
(mySchool)> create table teachers (id int primary key, name str not null, salary int)
Created table "teachers".
(mySchool)> create table temporary_teachers (id int primary key, name str not null, salary int,
duration int)
Created table "temporary_teachers".
(mySchool)> insert into teachers values (2525, Leftieris, 20010)
```



```
(mySchool)> insert into teachers values (1234, Maraki, 20810)
(mySchool)> insert into teachers values (0909, Ioanna, 23456)
(mySchool)> insert into temporary_teachers values (8900, Kostas, 21210, 6)
(mySchool)> insert into temporary_teachers values (1230, Mary, 14510, 12)
(mySchool)> insert into temporary_teachers values (1780, George, 1400, 24)
(mySchool)> select * from temporary_teachers
  id (int) #PK#  name (str)      salary (int)  duration (int)
-----
      8900  kostas              21210         6
      1230  mary               14510        12
      1780  george             1400         24

(mySchool)> select * from teachers
  id (int) #PK#  name (str)      salary (int)
-----
      2525  lefteris          20010
      1234  maraki            20810
      0909  ioanna            23456

(mySchool)> insert into temporary_teachers select * from teachers
(mySchool)> select * from temporary_teachers
  id (int) #PK#  name (str)      salary (int)  duration (int)
-----
      8900  kostas              21210         6
      1230  mary               14510        12
      1780  george             1400         24
      2525  lefteris          20010        null
      1234  maraki            20810        null
      0909  ioanna            23456        null

(mySchool)>
```