ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ
ΕΡΓΑΣΙΑ ΜΑΘΗΜΑΤΟΣ 2022-2023

ΝΑΠΟΛΕΩΝ ΚΟΥΤΣΟΥΡΙΔΗΣ Π16058
ΣΠΥΡΙΔΩΝ ΣΑΛΑΓΙΑΝΝΗΣ Π16197

#1 Enrich WHERE statement

(a)NOT,BETWEEN

Ο τελεστής NOT βρίσκεται ενσωματωμένος στο πακέτο operators,το οποίο χρησιμοποιείται και για τους υπόλοιπους τελεστές της συνθήκης where,δηλαδή τα <,>,= και τα υπόλοιπα.Επομένως με δήλωση του τελεστή not στις μεθόδους split_condition και get_op που βρίσκονται στο αρχείο misc.py μπορούμε να χρησιμοποιήσουμε τον τελεστή not.

```
(smdb)> select * from course where credits not 4
course_id (str) #PK#    title (str)              dept_name (str)      credits (int)
----------------------  -----------------------  -----------------  --------------
bio-399                 computational biology    biology                         3
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
```

Ο τελεστής BETWEEN δεν βρίσκεται στο πακέτο operators,οπότε θα δημιουργήσουμε μία συνάρτηση η οποία θα πραγματοποιεί την λειτουργία αυτή. Λόγω των αλλαγών που θα έπρεπε να υλοποιήσουμε για να λειτουργεί όπως στην MySQL (select * from _ where _ between _ and _) επιλέξαμε να ορίσουμε το query ώς εξής: select * from _ where _ between _,_ και να επεξεργαστούμε την συνθήκη τοπικά στην συνάρτηση between,την οποία υλοποιήσαμε στο αρχείο misc.py.
Για να μπορέσουμε να επεξεργαστούμε στην συνάρτηση between το κομμάτι της συνθήκης τροποποιήσαμε την μέθοδο parse_table στην κλάση Table.py τροποποιώντας το return ώστε να επιστρέφει string στην περίπτωση που ο τελεστής είναι το between.

```
(smdb)> select * from student where name between a,d
  id (str) #PK#  name (str)    dept_name (str)      tot_cred (int)
---------------  ------------  -----------------  ----------------
          19991  brandt        history                          80
          23121  chavez        finance                         110
          76543  brown         comp. sci.                       58
          76653  aoi           elec. eng.                       60
          98765  bourikas      elec. eng.                       98

(smdb)> select * from student where tot_cred between 10,50
  id (str) #PK#  name (str)    dept_name (str)      tot_cred (int)
---------------  ------------  -----------------  ----------------
          12345  shankar       comp. sci.                       32
          45678  levy          physics                          46
          55739  sanchez       music                            38
```

(b)AND,OR

Οι τελεστές AND και OR χρησιμοποιούνται ώστε να συνενώνονται εντολές,δηλαδή δύο ή παραπάνω τελεστές στην σειρά.Η λογική που ακολουθήσαμε είναι να αντιμετωπίσουμε τα επιμέρους υποερωτήματα ώς κανονικά queries και συνδυάσουμε τα αποτελέσματα.Για αυτόν τον λόγο δημιουργήσαμε την μέθοδο get_op_results στο αρχείο Table.py η οποία αντί να εκτυπώνει το αποτέλεσμα με την μέθοδο του πακέτου tabulate,επιστρέφει το αποτέλεσμα.Επιπλέον ορίσαμε την συνάρτηση get_op_headers,ώστε να έχουμε το όνομα του κάθε πεδίου ώστε να εκτυπώσουμε με τον ίδιο τρόπο με πρίν τον τελικό πίνακα.Στην συνέχεια με κατάλληλη συνθήκη στο αρχείο mdb.py χωρίζουμε το σύνθετο query σε επιμέρους απλά queries και συνδυάζουμε τα αποτελέσματα κάνοντας τις αντίστοιχες πράξεις για τον κάθε operator.

```
(smdb)> select * from student
  id (str) #PK#  name (str)    dept_name (str)      tot_cred (int)
---------------  ------------  -----------------  ----------------
          00128  zhang         comp. sci.                      102
          12345  shankar       comp. sci.                       32
          19991  brandt        history                          80
          23121  chavez        finance                         110
          44553  peltier       physics                          56
          45678  levy          physics                          46
          54321  williams      comp. sci.                       54
          55739  sanchez       music                            38
          70557  snow          physics                           0
          76543  brown         comp. sci.                       58
          76653  aoi           elec. eng.                       60
          98765  bourikas      elec. eng.                       98
          98988  tanaka        biology                         120

(smdb)> select * from student where tot_cred between 10,50 or name=zhang
  id (str) #PK#  name (str)    dept_name (str)      tot_cred (int)
---------------  ------------  -----------------  ----------------
          12345  shankar       comp. sci.                       32
          45678  levy          physics                          46
          55739  sanchez       music                            38
          00128  zhang         comp. sci.                      102
(smdb)> select * from student where tot_cred between 10,50 and name=levy
  id (str) #PK#  name (str)    dept_name (str)      tot_cred (int)
---------------  ------------  -----------------  ----------------
          45678  levy          physics                          46
```

Το παραπάνω στιγμιότυπο οθόνης παρουσιάζει ένα απλό select,το οποίο εκτυπώνει ολόκληρο τον πίνακα,ένα select με την χρήση του OR και ένα ακόμη με την χρήση του AND,ώστε να παρουσιαστεί η λειτουργικότητα.

#Enrich indexing functionality

(a)Btree index over unique(non-PK) columns

Η ιδιότητα ενός πεδίου unique είναι ότι κάθε εγγραφή παρουσιάζεται μόνο μία φορά,επομένως όπως και στο πεδίο που λειτουργεί ως πρωτεύων κλειδί(PK),έτσι και σε ένα πεδίο unique μπορεί να υποστηριχθεί ευρετήριο καθώς δεν υπάρχουν διπλότυπες εγγραφές.

Για να μπορούμε να καλέσουμε ένα τέτοιο ευρετήριο πρέπει να προσθέσουμε την ιδιότητα unique ως λειτουργικότητα στην miniDB.Για να μπορέσουμε να έχουμε την ιδιότητα unique τροποποιήσαμε τις μεθόδους __init__ show και get_op_headers της κλάσης Table.py,στην κλάση Database.py τροποποιήσαμε την μέθοδο __init__ για να δέχεται το unique ως όρισμα και στο αρχείο mdb.py.

Οπότε έχουμε αποτέλεσμα σαν το ακόλουθο.

```
(smdb)> create table unique_test(col1 int,col2 str unique)
Created table "unique_test".
(smdb)> insert into unique_test values(1,'a')
(smdb)> insert into unique_test values(2,'b')
(smdb)> select * from unique_test
  col1 (int)  col2 (str)#UNIQUE#
------------  --------------------
           1  'a'
           2  'b'

(smdb)>
```

Εφόσον το πεδίο με το χαρακτηριστικο PK(primary key) έχει τις ίδιες ιδιότητες με αυτό του UNIQUE, όσον αφορά την λειτουργία που μας ενδιαφέρει,μπορούμε να χρησιμοποιήσουμε την ίδια μέθοδο και για αυτό το πεδίο.Επομένως με αναζήτηση πάνω στο πεδίο με τον περιορισμό unique πραγματοποιούμε αναζήτηση με Btree.

```
(smdb)> select col2 from unique_test
col2 (str)
------------
'a'
'b'
```