ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΗΣΗΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

ΕΡΓΑΣΙΑ ΜΑΘΗΜΑΤΟΣ

ΚΩΝΣΤΑΝΤΙΝΟΣ-ΜΠΟΡΙΣ ΣΟΛΔΑΤΟΣ

Π16132

ΕΚΦΩΝΗΣΗ

Ομάδα εργασιών 3 (Γενικά - Διάφορα)

Task 3.1 - Extra I/O and Primary Key rethink (*)

Υλοποίηση των παρακάτω λειτουργιών (functionalities):

1. Group by

2. Select Distinct

Επίσης, καλείστε να επανασχεδιάσετε τον τρόπο που αποθηκεύεται το πρωτεύον κλειδί (primary key), καθώς και να εντάξετε τη δυνατότητα ορισμού multicolumn primary keys.

ΕΠΕΞΗΓΗΣΗ

Τα ζητούμενα της εργασίας υλοποιήθηκαν στα ήδη υπάρχοντα αρχεία **database.py** και **table.py**.

1. Group by

Για το **group by** δημιούργησα την εξής μέθοδο στο αρχείο table.py:

```
341     # GROUP BY function for TASK 3.1
342     def group_by(self, column_name, asc=True):
343         column = self.columns[self.column_names.index(column_name)]
344 
345         # sort data
346         idx = sorted(range(len(column)), key=lambda k: column[k], reverse=not asc)
347         a = [self.data[i] for i in idx]
348 
349         # index of column we use to group by
350         idx_col = self.column_names.index(column_name)
351         b = []
352 
353         # check for and remove any duplicates
354         for i in range(len(a)-1):
355             if i != len(a)-2:
356                 if a[i][idx_col] != a[i+1][idx_col]:
357                     b.append(a[i])
358             elif i == len(a)-2:
359                 if a[i][idx_col] != a[i+1][idx_col]:
360                     b.append(a[i])
361                     b.append(a[i+1])
362                 else:
363                     b.append(a[i+1])
364 
365         # return table but arange data using idx list (sorted indexes)
366         dict = {(key):(b if key=="data" else value) for key, value in self.__dict__.items()}
367         return dict
```

Η παράμετρος column_name είναι η στήλη στην οποία βασίζεται η λειτουργεία group by.

Αρχικά, με βάση την στήλη αυτή (column_name) κάνω διάταξη όλων των δεδομένων σε αύξουσα σειρά και τα αποθηκεύω στο a. Όταν τα δεδομένα είναι ταξινομημένα είναι πιο εύκολο και απλό να βρεθούν οι διπλότυπες εγγραφές.

Έπειτα έχω μια for loop για να αφαιρέσω όλες τις διπλότυπες εγγραφές για την στήλη, με βάση την οποία κάνουμε το group by. Μέσα στην for loop γίνεται η διαδικασία του group by όπου για τις διπλότυπες εγγραφές κρατάμε την τελευταία εγγραφή και διαγράφουμε όλες τις προηγούμενες. Το τελικό αποτέλεσμα το αποθηκεύω στο b.

Τέλος επιστρέφω τον πίνακα ομαδοποιημένο (group-ed by) με βάση την στήλη column_name.

Παράδειγμα Εκτέλεσης

Για να τρέξουμε το select με την λειτουργία group by, μέσα στις παρενθέσεις του select γράφουμε **group_by='column'** όπου column η στήλη που θα επιλέξουμε για να γίνει η ομαδοποίηση:

db.select('student', '*', **group_by='name'**)

```
>>> db.select('student', '*', group_by='name')

## student ##
  ID (str) #PK#  name (str)    dept_name (str)     tot_cred (int)
---------------  ------------  -----------------  ---------------
          76653  Aoi           Elec. Eng.                      60
          98765  Bourikas      Elec. Eng.                      98
          19991  Brandt        History                         80
          76543  Brown         Comp. Sci.                      58
          23121  Chavez        Finance                        110
          45678  Levy          Physics                         46
          44553  Peltier       Physics                         56
          55739  Sanchez       Music                           38
          12345  Shankar       Comp. Sci.                      32
          70557  Snow          Physics                          0
          98988  Tanaka        Biology                        120
          54321  Williams      Comp. Sci.                      54
          00128  Zhang         Comp. Sci.                     102
```

db.select('student', '*', **group_by='dept_name'**)

```
>>> db.select('student', '*', group_by='dept_name')

## student ##
  ID (str) #PK#  name (str)    dept_name (str)     tot_cred (int)
---------------  ------------  -----------------  ---------------
          98988  Tanaka        Biology                        120
          76543  Brown         Comp. Sci.                      58
          98765  Bourikas      Elec. Eng.                      98
          23121  Chavez        Finance                        110
          19991  Brandt        History                         80
          55739  Sanchez       Music                           38
          70557  Snow          Physics                          0
```

2. Select Distinct

Για το **select distinct** στο αρχείο table.py μέσα στις μεθόδους **_select_where** και **_select_where_with_btree** έχω γράψει τον εξής κώδικα:

```
205         # SELECT DISTINCT TASK 3.1 #
206
207         drows = [[self.data[i][j] for j in return_cols] for i in rows]
208         #print(drows)
209         ndrows = []
210         for elem in drows:
211             if elem not in ndrows:
212                 ndrows.append(elem)
213
214         drows = ndrows
215
216         # END SELECT DISTINCT TASK 3.1 #
```

Εδώ για κάθε γραμμή με βάση αυτό που έχω επιλέξει (select … from student) ελέγχω αν υπάρχουν όμοιες γραμμές και αν υπάρχουν δεν τις κρατάω.

Παράδειγμα Εκτέλεσης

Αν θέλουμε η επιλογή μας να είναι **distinct,** μέσα στις παρενθέσεις του select γράφουμε **distinct=True.**

db.select('student', ['dept_name'], **distinct=True**)

```
>>> db.select('student', ['dept_name'], distinct=True)

## student ##
dept_name (str)
----------------
Comp. Sci.
History
Finance
Physics
Music
Elec. Eng.
Biology
```

3. Primary Key

Στο αρχείο table.py άλλαξα τον τρόπο με τον οποίο αποθηκεύεται το primary key. Αντί να αποθηκεύεται το index το κλειδιού αποθηκεύω το όνομα το κλειδιού και ένταξα την δυνατότητα ορισμού multicolumn primary key.

```
62              # PRIMARY KEY MULTI COLUMN SUPPORT AND NEW STORING METHOD TASK 3.1 #
63              if primary_key is not None:
64                  if not isinstance(primary_key, list):
65                      #self.pk_idx = self.column_names.index(primary_key)
66                      self.pk_idx = primary_key
67                  else:
68                      self.pk_idx = []
69                      for i in primary_key:
70                          #self.pk_idx.append(self.column_names.index(i))
71                          self.pk_idx.append(i)
72              # END OF PRIMARY KEY MULTI COLUMN SUPPORT AND NEW STORING METHOD TASK 3.1
```

Εδώ ελέγχω αν η μεταβλητή primary_key είναι λίστα ή όχι, αν είναι λίστα σημαίνει ότι το table έχει multicolumn primary key. Αν θέλουμε ένα table να έχει για παράδειγμα 2 primary keys γράφουμε μέσα στο create_tableQ **primary_key=[‘pk1’, ‘pk2’]**

Για παράδειγμα:

db.create_table('course', ['course_id', 'credits'], [str,int], primary_key=['course_id', 'credits'])

Για την προβολή του πίνακα με τις αλλαγές που έχω κάνει στη μέθοδο show τροποποίησα τον κώδικα για να δείχνει σωστά όλα τα primary keys, δηλαδή να προστεθεί δίπλα από το όνομα της κατάλληλης στήλης το #PK#.

```
450         # TASK 3.1 updated the addition of pk to each appropriate column for multi column keys
451         if self.pk_idx is not None:
452             if not isinstance(self.pk_idx, list):
453                 # table has a primary key, add PK next to the appropriate column
454                 for j in range(len(headers)):
455                     if (headers[j].split('(')[0]).strip() == self.pk_idx:
456                         headers[j] = headers[j] + ' #PK#'
457             else:
458                 for j in range(len(headers)):
459                     for i in range(len(self.pk_idx)):
460                         if (headers[j].split('(')[0]).strip() == self.pk_idx[i]:
461                             headers[j] = headers[j] + ' #PK#'
462         # TASK 3.1 END
```

Παράδειγμα Εκτέλεσης

```
db.create_table('course', ['course_id', 'title', 'dept_name', 'credits'], [str,str,str,int], primary_key=['course_id', 'credits'])
```

```
>>> db.select('course', '*')

## course ##
course_id (str) #PK#    title (str)                dept_name (str)      credits (int) #PK#
---------------------   ------------------------   ----------------   --------------------
BIO-101                 Intro. to Biology          Biology                              4
BIO-301                 Genetics                   Biology                              4
BIO-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
```

Εδώ βλέπουμε ότι έχει προστεθεί το #PK# στα κατάλληλα columns.

3.1 View Primary Key

Επιπλέον έχω δημιουργήσει μία μέθοδο για την προβολή του primary key για το table που θα επιλέξουμε. Η μέθοδος (βρίσκεται στο αρχείο table.py):

```
325     # function to print the primary key column for a specified table TASK 3.1
326     def showpk(self):
327         if self.pk_idx is not None:
328             rows = [i for i in range(len(self.columns[0]))]
329             if not isinstance(self.pk_idx, list):
330                 tt = self.column_names.index(self.pk_idx)
331                 pp = [self.data[i][tt] for i in rows]
332             else:
333                 tt = []
334                 for i in self.pk_idx:
335                     tt.append(self.column_names.index(i))
336                 pp = [[self.data[i][j] for j in tt] for i in rows]
337
338
339             print(pp)
```

Όπου αυτή η μέθοδος καλείται από μία άλλη μέθοδο στο αρχείο **database.py:**

```
336     # function to print the primary key column values TASK 3.1
337     def show_pk(self, table_name):
338         pk = self.tables[table_name].showpk()
```

Η μέθοδος αυτή εκτυπώνει στην κονσόλα την στήλη που έχει οριστεί ως primary key.

Παράδειγμα Εκτέλεσης

Το primary key για το student:

```
>>> db.show_pk('student')
['00128', '12345', '19991', '23121', '44553', '45678', '54321', '55739', '70557', '76543', '76653', '98765', '98988']
```

Το primary key για το course (multicolumn primary key):

```
>>> db.show_pk('course')
[['BIO-101', 4], ['BIO-301', 4], ['BIO-399', 3], ['CS-101', 4], ['CS-190', 4], ['CS-315', 3], ['CS-319', 3], ['CS-347', 3]
 ['EE-181', 3], ['FIN-201', 3], ['HIS-351', 3], ['MU-199', 3], ['PHY-101', 4]]
```