

Π18226 Χριστίνα-Μαρία Λαγού

Π18166 Παναγιώτα Φουρνάρη

Εργασία συστήματα διαχείρισης βάσεων δεδομένων.

Task 3.1 - Extra I/O and Primary Key rethink

Οι επεξεργασία έγινε στο αρχείο table.py και στο database.py

SELECT DISTINCT

```
292 def select_distinct(self, column_name):
293     column = self.columns[self.column_names.index(column_name)]
294     unique_values = {}
295     for idx, value in enumerate(column):
296         if value not in unique_values:
297             unique_values[value] = idx
298     dict = {(key): ([self.data[i] for i in unique_values.values() if key=="data" else value) for key, value in self._dict_.items())}
299     print(dict)
300     return Table(load=dict)
```

Η Select distinct παίρνει δυο ορίσματα (self, column_name). Ορίζουμε μια κενή λίστα και χρησιμοποιούμε επανάληψη για να ελέγξουμε όλα τα πεδία . Για κάθε value αν δεν υπάρχει στην λίστα τότε το βάζει σε αυτή .

#db.select('classroom','*')

```
## classroom ##
building (str)    room_number (str)    capacity (int)
-----
Packard          101          500
Painter          514          10
Taylor           3128         70
Watson           100          30
Watson           120          50
panagiota        20           50
giorgos          12           50
```

#db.select('classroom','*',select_distinct='building')

```
## classroom ##
building (str)    room_number (str)    capacity (int)
-----
Packard          101          500
Painter          514          10
Taylor           3128         70
Watson           100          30
panagiota        20           50
giorgos          12           50
```

Το Watson εμφανίζεται μόνο μια φορά και όχι δυο που ήταν στο αρχικό πίνακα .

GROUP_BY

Το διαφορετικό που κάνει από την Select distinct είναι ότι έχει και τα aggregate functions δηλαδή ανάλογα τι aggregate θα του βάλουμε θα μας βγάλει και μια ακόμα κολώνα στον πίνακα με αυτό που του ζητάμε .

Το group by τα σορτάρει και τα εμφανίζει ταξινομημένα .

Μια ιδέα είναι να δημιουργήσουμε μια νέα λίστα και να κάνουμε order by την στήλη πχ θα παίρναμε το column_number και θα έκανε order by συμφώνα με το αριθμό της στήλης που έχουμε . Επιπλέον μπορούμε να πάρουμε το unique_values και μετά να κάνω sort. Δυστυχώς καμία από αυτές τις σκέψεις δεν μπορέσαμε να υλοποιήσουμε .

```
286
287 def group_by(self, column_name, asc=False):
288     print('group by')
289     column = self.columns[self.column_names.index(column_name)]
290
291     unique_values = {}
292     for idx, value in enumerate(column):
293         if value not in unique_values:
294             unique_values[value] = idx
295
296     unique_values = sorted(range(len(column)), reverse=not asc)
297     rows = []
298     rows.sort(key=itemgetter(column_number))
299
300     dict = {(key):([self.data[i] for i in unique_values.values() if key=="data" else value] for key, value in self.__dict__.items())}
301     return Table(load=dict)
```