

Εργασία Συστήματα Διαχείρισης Βάσεων Δεδομένων

Ονοματεπώνυμο: Αλέξιος Ζορμπάς

Αριθμός Μητρώου: π18045

Task που υλοποιήθηκε: 1.1 Ordered Files

Στο repository υπαρχει ενα zip miniDB-master που μεσα περιεχει τα αρχεια της miniDB με τον κωδικα αλλαγμενο ωστε να συμβαινει οτι περιγραφεται παρακατω

Αρχικά προσεθηκε στο αρχείο Table.py ,η insert stack και μια μεταβλητη s (η οποια βοηθαι στην ταξινομηση του πινακα)

```
class Table:
    istack=[]
    s=None#μεταβλητη που χρησιμοποιειται ωστε να ταξινομειται ο πινακας μο'
    '''
    Table object represents a table inside a database

    A Table object can be created either by assigning:
    - a table name (string)
    - column names (list of strings)
    - column types (list of functions like str/int etc)
    - primary (name of the primary key column)
```

Παρακατω φαινεται πως γινονται οι εισοδοι των στοιχεων(αρχικα στην insert stack,και στη συνεχεια οταν αυτη εχει μεγεθος 5 στον πινακα(τοτε η insert stack αδειαζει))

οι αλλαγες παρακατω ειναι στο αρχείο database.py

```
def insert(self, table_name, row, lock_load_save=True):
    '''
    Inserts into table

    table_name -> table's name (needs to exist in database)
    row -> a list of the values that are going to be inserted (will be automatically casted to predifi:
    lock_load_save -> If false, user need to load, lock and save the states of the database (CAUTION).
    '''
    if lock_load_save:
        self.load(self.savedir)
        if self.is_locked(table_name):
            return
        # fetch the insert_stack. For more info on the insert_stack
        # check the insert_stack meta table
        self.lockX_table(table_name)
        insert_stack = self._get_insert_stack_for_table(table_name)
    try:
        self.tables[table_name].istack.append(row)#εισαγωγη των εγγραφων αρχικα στην insert stack
        self.tables[table_name].s=0#η μεταβλητη s του αντικειμενου table(αρχικοποιειται ως 0)

        if (len(self.tables[table_name].istack)==5):#μεταφορα στον πινακα και ταξινομηση του και αδεια
            count=0
            for row in self.tables[table_name].istack:
                if (count==len(self.tables[table_name].istack)-1):
                    self.tables[table_name].s=1#η μεταβλητη s του αντικειμενου table(γινεται

                    self.tables[table_name]._insert(row, insert_stack)
                    count=count+1

            self.tables[table_name].istack.clear()

    except Exception as e:
        print(e)
```

παραπανω η s του table γινεται 0 καθε φορα που κατι μπαινει στην insertstack και οταν αυτη αδειζει αυτη γινεται 1(το γιατι εξηγειται παρακατω)

Παρακατω φαίνεται πως υλοποιούνται τα ordered files(στην insert του αρχείου table.py)

```
def _insert(self, row, insert_stack=[]):
    """
    Insert row to table
    """
    if len(row) != self._no_of_columns:
        raise ValueError(f'ERROR -> Cannot insert {len(row)} values. Only {self._no_of_columns} columns exist')

    for i in range(len(row)):
        # for each value, cast and replace it in row.
        try:
            row[i] = self.column_types[i](row[i])

        except:
            raise ValueError(f'ERROR -> Value {row[i]} is not of type {self.column_types[i]}.')

        # if value is to be appended to the primary_key column, check that it doesnt already exist (no duplicate prima
        if i == self.pk_idx and row[i] in self.column_by_name(self.pk):
            raise ValueError(f'## ERROR -> Value {row[i]} already exists in primary key column.')

    # if insert_stack is not empty, append to its last index
    if insert_stack != []:
        self.data[insert_stack[-1]] = row
    else: # else append to the end
        self.data.append(row)
    # self._update()
    if self.s == 1: # ο πίνακας ταξινομείται όταν η s γίνει 1 , μέσω της συνάρτησης insert ή select του database.py
        new_list = self.data.copy()
        self.data = sorted(new_list, key=lambda row: row[0])
```

Όταν η insert stack αδειάζει (και η s γίνεται 1), ο πίνακας ταξινομείται με βάση το 0ο στοιχείο

Η s χρησιμοποιείται για λόγους χρόνου , αν δεν υπήρχε, κάθε φορά που μια εγγραφή θα μεταφερόταν στον πίνακα από την insert stack , θα γινόταν ταξινόμηση, που θα σήμαινε ότι κάθε φορά που θα γινόταν αδειασμα του insert stack, θα γινόταν 5 sort (ενώ τώρα γίνεται 1)

Τελος, παρακατω φαινεται πως τα select λειτουργουν σωστα

```
self.load(self.savedir)
if self.is_locked(table_name):
    return
self.lockX_table(table_name)
if condition is not None:
    condition_column = split_condition(condition)[0]
if self._has_index(table_name) and condition_column==self.tables[table_name].column_names[self.tables[table_name].p]:
    index_name = self.select('meta_indexes', '*', f'table_name=={table_name}', return_object=True).index_name[0]
    bt = self.load_idx(index_name)
    table = self.tables[table_name]._select_where_with_btree(columns, bt, condition, order_by, asc, top_k)
else:
    if (len(self.tables[table_name].istack) !=0):#ελεγχος αν η insert stack εχει στοιχεια
        insert_stack = self._get_insert_stack_for_table(table_name)
        self.tables[table_name].s=0#η μεταβλητη s του αντικειμενου table(αρχικοποιειται ως 0)
        count=0
        for row in self.tables[table_name].istack:#μεταφορα στον πινακα και ταξινομηση του,(προσωρινα μονο για να λι
            if (count==len(self.tables[table_name].istack)-1):
                self.tables[table_name].s=1#η μεταβλητη s του αντικειμενου table(γινεται 1 οταν εισαγεται
                self.tables[table_name]._insert(row, insert_stack)
                count=count+1
            #self.tables[table_name].istack.clear()
        table = self.tables[table_name]._select_where(columns, condition, order_by, asc, top_k)
self.unlock_table(table_name)
if save_as is not None:
    table._name = save_as
    self.table_from_object(table)
else:
    if return_object:
        return table
    else:
```

Γινεται η ιδια διαδικασια με την εισοδο των εγγραφων,με διαφορα οτι :

-υλοποιειται στην select του database.py και οχι στην insert

-η insert stack δεν αδειαζει ,διοτι νεος πινακας που δημιουργειται απο το merge με την insert stack ειναι προσωρινος(δεν αποθηκευεται μονιμα)

Τρεξιμο του κωδικα:

Στο repository ,έχει ανεβεί ένα αρχείο vsmdb.py που περιέχει τον παρακατω κωδικα

```
from database import Database
# create db with name "smdb"
db = Database('vsmdb', load=False)
# create a single table named "classroom"
db.create_table('classroom', ['building', 'room_number', 'capacity'], [str,str,int])
# insert 5 rows
db.insert('classroom', ['B', '101', '500'])
db.insert('classroom', ['A', '514', '10'])
db.insert('classroom', ['Taylor', '3128', '70'])
db.insert('classroom', ['Watson', '100', '30'])
db.insert('classroom', ['C', '120', '50'])
db.insert('classroom', ['B1', '101', '500'])
db.insert('classroom', ['A1', '514', '101'])
#db.insert('classroom', ['Taylor1', '3128', '70'])
#db.insert('classroom', ['Watson1', '100', '30'])
#db.insert('classroom', ['C1', '120', '50'])
db.select('classroom', '*')
db.select('classroom', '*', 'capacity>100')
```

που αν τον τρεξουμε βλέπουμε τα παρακατω:

```
New table "meta_length"
New table "meta_locks"
New table "meta_insert_stack"
New table "meta_indexes"
New table "classroom"
```

```
## classroom ##
building (str)      room_number (str)      capacity (int)
-----
A                   514                      10
A1                  514                      101
B                   101                      500
B1                  101                      500
C                   120                      50
Taylor              3128                     70
Watson              100                      30
```

```
## classroom ##
building (str)      room_number (str)      capacity (int)
-----
A1                  514                      101
B                   101                      500
B1                  101                      500
```

```
>>>
```

Παρατηρούμε οτι οι εγγραφες είναι ταξινομημένες με βάση το πρώτο στοιχείο και οτι τα select λειτουργούν σωστά

Στο συγκεκριμένο σημείο ,η insertstack έχει μέσα 2 στοιχεία (τα 2 τελευταία που γίνονται insert) και ο πίνακας έχει 5 (τα πρώτα 5 που έγιναν insert στον κωδικά) αλλά τα select λειτουργούν σωστά.