DOCUMENTATION ΆΣΚΗΣΗΣ 2.2

Η εργασία υλοποιήθηκε από τους φοιτητές:

Π18007 Παναγιώτης Ανδρεάτος

Π18111 Μιχαήλ-Βασίλειος Μπούφας

Π18205 Ευθύμιος Καζάκος

Ο κώδικας υλοποιήθηκε από κοινού.

Για την εργασίας υλοποιήθηκε ο κώδικας log.py , τα αρχεία logger.txt, tableforms.data, stack.data, Analitikolog.txt και τροποποιήθηκε το database.py

Στο database.py :

1. Κάναμε import το log και datetime
```
import log
import datetime
```
2. Στην πλειοψηφία των μεθόδων προσθέσαμε ένα μήνυμα που το περνάμε στο Αναλυτικό logger (Analitikolog.txt). Το μήνυμα αυτό περνάει την ημερομηνία και την ώρα όπου έγινε η ενέργεια, το όνομα της ενέργειας, καθώς και όλα τα άλλα στοιχεία που μπορεί να είναι απαραίτητα (Όνομα πίνακα, όνομα βάσης κ.τ.λ.).
Παράδειγμα:
```
log.Log.trlog("DATE_AND_TIME: "+str(x)+" FUNCTION: database_creation "+"DATABASE_NAME: "+str(name))
```
3. Σε κάποιες από αυτές τις μεθόδους υπάρχει αντίστοιχα και ένα επιπλέον μήνυμα το οποίο το περνάμε στο logger (logger.txt) (Το logger αυτό χρησιμοποιείται στο rollback το οποίο θα αναλύσουμε στην συνέχεια.).Αυτή η λειτουργία γίνεται με 2 τρόπους.
Παράδειγμα:
```
f = open("logger.txt", "a")
f.write("database_creation,"+str(name))
f.write("\n")
f.close()
```
Ή
```
log.Log.save_table_data(self, table_name)
```

Στο log.py:

1. Δημιουργία της μεθόδου save_table_data η οποία σώζει στο αρχείο tableforms.data το τρέχον στιγμιότυπο του πίνακα, την τιμή του meta_insert_stack του στο αρχείο stack.data και στο αρχείο logger.txt το όνομα του πίνακα και το όνομα της βάσης στην οποία ανήκει.
2. Δημιουργία της μεθόδου rollback:
Υποστηρίζεται rollback στις εντολές:
__init__
drop_db
create_table
table_from_csv
drop_table
cast_column
insert
update
delete
Αρχικά τραβάμε τα περιεχόμενα του logger.txt και τα αποθηκεύουμε σε μια λίστα την οποία αντιστρέφουμε. Έπειτα ελέγχουμε αν το πλήθος των στοιχείων της λίστας (αριθμός εγγραφών στο logger.txt) είναι μικρότερο από τον αριθμό των εντολών (παράμετρος N) για τον οποίο ο χρήστης επιθυμεί να κάνει rollback. Αν συμβαίνει κάτι τέτοιο τυπώνουμε μήνυμα λάθους και διακόπτουμε την λειτουργία. Σε αντίθετη περίπτωση ελέγχουμε αν ο

χρήστης επιθυμεί rollback κατά βάση ή κατά πίνακα (Τιμές παραμέτρων database_name,table_name). Στο rollback κατά πίνακα υποστηρίζονται οι λειτουργίες:
create_table
table_from_csv
drop_table
cast_column
insert
update
delete
Για τις λειτουργίες create_table & table_from_csv , διαγράφουμε τον πίνακα που δημιουργήθηκε. Για τις υπόλοιπες επαναφέρουμε το τελευταίο αποθηκευμένο στιγμιότυπο του πίνακα και του meta_insert_stack του. Διαγράφουμε τον πίνακα από την βάση και χρησιμοποιώντας την μέθοδο table_from_object τον ξαναδημιουργούμε σύμφωνα με το στιγμιότυπο που ανακτήσαμε καθώς και ενημερώνουμε το meta_insert_stack του. Το meta_length του ενημερώνεται με την κλήση της μεθόδου _update που υπάρχει στο database.py . Δεν κάνουμε τίποτα για το meta_index, καθώς η drop_table δεν το επηρεάζει.
Στο rollback κατά βάση υποστηρίζονται όλες οι παραπάνω λειτουργίες καθώς επίσης:
__init__
drop_db
Για την λειτουργία __init__ διαγράφουμε τη βάση.
Για την λειτουργία drop_db εμφανίζουμε ότι δεν μπορούμε να κάνουμε rollback, καθώς αν και έχουμε την δυνατότητα να αποθηκεύσουμε κάποιο στιγμιότυπο της βάσης, δεν γίνεται να το επαναφέρουμε.
Όπου χρειάζεται να επαναφέρουμε στιγμιότυπα, στη συνέχεια τα διαγράφουμε από τα αντίστοιχα αρχεία.
Τέλος, διαγράφουμε από το logger.txt όσες εγγραφές χρησιμοποιήθηκαν στο rollback.

3. Δημιουργία της μεθόδου trlog για να περνάμε το ιστορικό εντολών της βάσης στο αρχείο Analitikolog.txt . Καλείται από την database.py
4. Δημιουργία της μεθόδου show_log για την εμφάνιση του περιεχομένου του Analitikolog.txt

Παράδειγμα του κώδικα που τρέχει:

vsmdb.py - C:\Users\user\Desktop\miniDB\vsmdb.py (3.8.3)

File Edit Format Run Options Window Help

```
from database import Database
from log import Log
# 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],'capacity')
# insert 5 rows
db.create_index('classroom','leksh')
db.insert('classroom', ['Packard', '101', '500'])
db.insert('classroom', ['Painter', '514', '10'])
db.insert('classroom', ['Taylor', '3128', '70'])
db.insert('classroom', ['Watson', '100', '30'])
db.delete('classroom','capacity==30')
db.meta_insert_stack.show()
db.insert('classroom', ['Watson', '120', '50'])
db.meta_insert_stack.show()
db.update('classroom','20','capacity','capacity==50')
db.cast_column('classroom','capacity',str)
#db.table_to_csv('classroom')
#db.drop_table('classroom')
#db.table_from_csv('classroom.csv','classroom',[str,str,int],'capacity')
logger=Log()
logger.rollback(3,'vsmdb')
db.show_table('classroom')
db.meta_insert_stack.show()
db.meta_length.show()
logger.show_log()
```

```
========== RESTART: C:\Users\user\Desktop\mini
New table "meta_length"
New table "meta_locks"
New table "meta_insert_stack"
New table "meta_indexes"
New table "classroom"
Creating Btree index.
Deleted 1 rows

## meta_insert_stack ##
table_name (str)    indexes (list)
------------------  ----------------
classroom           [3]


## meta_insert_stack ##
table_name (str)    indexes (list)
------------------  ----------------
classroom           []

Starting rollback...
Loaded "vsmdb".
Deleted 1 rows
Deleted 1 rows
Deleted 1 rows
Loaded "vsmdb".
Deleted 1 rows
Deleted 1 rows
Deleted 1 rows
Loaded "vsmdb".
Deleted 1 rows
Deleted 1 rows
Deleted 1 rows
End of rollback.

## classroom ##
building (str)      room_number (str)    capacity (int) #PK#
----------------  --------------------  ---------------------
Packard                          101                     500
Painter                          514                      10
Taylor                          3128                      70


## meta_insert_stack ##
table_name (str)    indexes (list)
------------------  ----------------
classroom           []


## meta_insert_stack ##
table_name (str)    indexes (list)
------------------  ----------------
classroom           []


## meta_length ##
table_name (str)     no_of_rows (int)
------------------  ------------------
classroom                           3

DATE_AND_TIME: 2021-02-05 11:39:04.099696 FUNCTION: database_creation DATABASE_NAME: vsmdb
DATE_AND_TIME: 2021-02-05 11:39:04.174722 FUNCTION: table_creation DATABASE_NAME: vsmdb TABLE_NAME: classroom
DATE_AND_TIME: 2021-02-05 11:39:04.210730 FUNCTION: create_index DATABASE_NAME: vsmdb TABLE_NAME: classroom INDEX_NAME: leksh
DATE_AND_TIME: 2021-02-05 11:39:04.259088 FUNCTION: has_index DATABASE_NAME: vsmdb TABLE_NAME: meta_locks
DATE_AND_TIME: 2021-02-05 11:39:04.240083 FUNCTION: insert DATABASE_NAME: vsmdb TABLE_NAME: classroom INSERT_ROW: ['Packard', '101', 500]
DATE_AND_TIME: 2021-02-05 11:39:04.300097 FUNCTION: has_index DATABASE_NAME: vsmdb TABLE_NAME: meta_locks
DATE_AND_TIME: 2021-02-05 11:39:04.284093 FUNCTION: insert DATABASE_NAME: vsmdb TABLE_NAME: classroom INSERT_ROW: ['Painter', '514', 10]
DATE_AND_TIME: 2021-02-05 11:39:04.334105 FUNCTION: has_index DATABASE_NAME: vsmdb TABLE_NAME: meta_locks
DATE_AND_TIME: 2021-02-05 11:39:04.318102 FUNCTION: insert DATABASE_NAME: vsmdb TABLE_NAME: classroom INSERT_ROW: ['Taylor', '3128', 70]
DATE_AND_TIME: 2021-02-05 11:39:04.372113 FUNCTION: has_index DATABASE_NAME: vsmdb TABLE_NAME: meta_locks
DATE_AND_TIME: 2021-02-05 11:39:04.356110 FUNCTION: insert DATABASE_NAME: vsmdb TABLE_NAME: classroom INSERT_ROW: ['Watson', '100', 30]
DATE_AND_TIME: 2021-02-05 11:39:04.399119 FUNCTION: delete DATABASE_NAME: vsmdb TABLE_NAME: classroom
DATE_AND_TIME: 2021-02-05 11:39:04.423125 FUNCTION: has_index DATABASE_NAME: vsmdb TABLE_NAME: meta_locks
DATE_AND_TIME: 2021-02-05 11:39:04.497141 FUNCTION: has_index DATABASE_NAME: vsmdb TABLE_NAME: meta_locks
DATE_AND_TIME: 2021-02-05 11:39:04.481138 FUNCTION: insert DATABASE_NAME: vsmdb TABLE_NAME: classroom INSERT_ROW: ['Watson', '120', 50]
DATE_AND_TIME: 2021-02-05 11:39:04.555154 FUNCTION: has_index DATABASE_NAME: vsmdb TABLE_NAME: meta_locks
DATE_AND_TIME: 2021-02-05 11:39:04.539151 FUNCTION: update DATABASE_NAME: vsmdb TABLE_NAME: classroom SET_COLUMN capacity
DATE_AND_TIME: 2021-02-05 11:39:04.590163 FUNCTION: has_index DATABASE_NAME: vsmdb TABLE_NAME: meta_locks
DATE_AND_TIME: 2021-02-05 11:39:04.575159 FUNCTION: cast_column DATABASE_NAME: vsmdb TABLE_NAME: classroom COLUMN_NAME: capacity CAST_TYPE: <class 'str'>
DATE_AND_TIME: 2021-02-05 11:39:04.669203 FUNCTION: has_index DATABASE_NAME: vsmdb TABLE_NAME: meta_locks
DATE_AND_TIME: 2021-02-05 11:39:04.667201 FUNCTION: table_drop DATABASE_NAME: vsmdb TABLE_NAME: classroom
DATE_AND_TIME: 2021-02-05 11:39:04.745219 FUNCTION: table_creation DATABASE_NAME: vsmdb TABLE_NAME: classroom
DATE_AND_TIME: 2021-02-05 11:39:04.772224 FUNCTION: has_index DATABASE_NAME: vsmdb TABLE_NAME: meta_locks
DATE_AND_TIME: 2021-02-05 11:39:04.770224 FUNCTION: table_drop DATABASE_NAME: vsmdb TABLE_NAME: classroom
DATE_AND_TIME: 2021-02-05 11:39:04.845009 FUNCTION: table_creation DATABASE_NAME: vsmdb TABLE_NAME: classroom
DATE_AND_TIME: 2021-02-05 11:39:04.872063 FUNCTION: has_index DATABASE_NAME: vsmdb TABLE_NAME: meta_locks
DATE_AND_TIME: 2021-02-05 11:39:04.869887 FUNCTION: table_drop DATABASE_NAME: vsmdb TABLE_NAME: classroom
DATE_AND_TIME: 2021-02-05 11:39:04.938724 FUNCTION: table_creation DATABASE_NAME: vsmdb TABLE_NAME: classroom
DATE_AND_TIME: 2021-02-05 11:39:04.964364 FUNCTION: has_index DATABASE_NAME: vsmdb TABLE_NAME: meta_locks
DATE_AND_TIME: 2021-02-05 11:39:04.972357 FUNCTION: has_index DATABASE_NAME: vsmdb TABLE_NAME: meta_locks
DATE_AND_TIME: 2021-02-05 11:39:04.962351 FUNCTION: show_table DATABASE_NAME: vsmdb TABLE_NAME: classroom NO_OF_ROWS: None
```