

Συστήματα Διαχείρισης Βάσεων Δεδομένων

Εργασία Εξεταστικής Χειμερινού Εξαμήνου 2020-2021

Ομάδα ενός ατόμου

Ονοματεπώνυμο: Γεώργιος Σαμαράς

A.M. : Π18134

Εργασία: Task 3-2 : Privileges

Github Account: GSamarasCS (<https://github.com/GSamarasCS>)

Εκφώνηση Εργασίας:

Δυνατότητα υποστήριξης διαφορετικών χρηστών (users) στη miniDB με διαφορετικά δικαιώματα πρόσβασης (privileges). Συγκεκριμένα:

1. Διαφορετικοί χρήστες και ομάδες χρηστών
2. Δικαιώματα χρηστών ανά βάση
3. Δικαιώματα χρηστών ανά πίνακα (table)

Υλοποίηση / Επεξήγηση

Η εργασία έχει 3 απαιτούμενα στοιχεία όπως φαίνεται και στην εκφώνηση:

- Χρειάζεται να υπάρχουν διαφορετικά ήδη χρηστών
- Τα οποία έχουν διαφορετικά δικαιώματα ανά βάση αλλά
- και ανά πίνακα

Η εύκολη λύση σε αυτό το πρόβλημα είναι αντί να δημιουργήσουμε μία νέα βάση με χρήστες οι οποίοι έχουν διαφορετικούς ρόλους, να γίνεται διερώτηση είδους χρήστη κάθε φορά που κάποιος προσπαθεί να εκτελέσει μια "μέθοδο" την οποία δεν θα μπορούσε ένας απλός χρήστης να εκτελέσει χωρίς άδειες διαχειριστή. Για αυτόν τον λόγο δημιουργήσα ένα function το οποίο διερωτά κάθε φορά που ο χρήστης πάει να εκτελέσει μία "μέθοδο" που μπορεί να προκαλέσει αλλαγές σε βάση ή πίνακα, αν θέλει να την εκτελέσει ως διαχειριστής ή όχι.

Ο κώδικας είναι ο παρακάτω:

```
def privileges(method_name):  
    print("Run " + method_name + " method as: \n1 --> Administrator \n2 --> User")  
    answer = input("Your Answer: ")  
    while answer != '1' and answer != '2':  
        print("Invalid answer. Please try again.")  
        answer = input("Your answer: ")  
    return answer
```

Ο χρήστης απαντάει κάθε φορά γράφοντας ένα string. Οποιοδήποτε string πέρα του '1' ή '2' απρρίπτεται και η ερώτηση επαναλαμβάνετε μέχρι να υπάρξει μία αποδεκτή απάντηση την οποία επιστρέφει.

Φυσικά, υπάρχει και συνέχεια στον κώδικα, η οποία γράφτηκε μέσα στον κώδικα των functions που μπορούν να αλλάξουν μία βάση ή πίνακα, όπως φαίνεται στην παρακάτω εικόνα:

```
83 def drop_db(self):  
84     answer = privileges("drop database")  
85     if answer == '2':  
86         print("You need Admin privileges to run this method.")  
87         return  
88     shutil.rmtree(self.savedir)  
89
```

Αυτό το κομμάτι του κώδικα γράφτηκε απευθείας μέσα στα αντιστοιχία functions καθώς δεν μπορεί αλλιώς να εκτελέσει την εντολή return για να σταματήσει την εκτέλεση του function και για να μην χρειάζεται η δημιουργία ενός δεύτερου if answer == '2' μέσα στον κώδικα.

Τέλος, παρατηρήθηκε πως και ο constructor __init__ μπορεί να μεταβάλει / αλλάξει πίνακες, για αυτό προστέθηκε και σε αυτόν το function privileges, με ένα παραπάνω restriction στην "μέθοδο" create_tables για πιο ομαλή εκτέλεση του constructor

```
101 def create_table(self, name=None, column_names=None, column_types=None, primary_key=None, load=None):
102     """
103     This method create a new table. This table is saved and can be accessed by
104     db_object.tables['table_name']
105     or
106     db_object.table_name
107     """
108     if name != "meta_length" and name != "meta_locks" and name != "meta_insert_stack" and name != "meta_indexes":
109         answer = privileges("create table")
110         if answer == '2':
111             print("You need Admin privileges to run this method.")
112             return
113     self.tables.update({name: Table(name=name, column_names=column_names, column_types=column_types, primary_key=primary_key,
114                                     # self._name = Table(name=name, column_names=column_names, column_types=column_types, load=load)
115                                     # check that new dynamic var doesnt exist already
116                                     if name not in self.__dir__():
117                                         setattr(self, name, self.tables[name])
118                                     else:
119                                         raise Exception(f'Attribute "{name}" already exists in class "{self.__class__.__name__}"')
120                                     # self.no_of_tables += 1
121                                     print(f'New table "{name}"')
122                                     self._update()
123                                     self.save()
124     })
```

Παράδειγμα Εκτέλεσης Κώδικα

Για το παρακάτω παράδειγμα δημιουργήθηκε το αρχείο python demo_database.py, του οποίου ο κώδικας είναι ο εξής:

```
1 from database import Database
2 # create db with name "DemoDB"
3 db = Database('DemoDB', load=False)
4 # create a single table named "classroom"
5 db.create_table('classroom', ['building', 'room_number', 'capacity'], [str, str, int])
6 # insert 5 rows
7 db.insert('classroom', ['Packard', '101', '500'])
8 db.insert('classroom', ['Painter', '514', '10'])
9 db.insert('classroom', ['Taylor', '3128', '70'])
10 db.insert('classroom', ['Watson', '100', '30'])
11 db.insert('classroom', ['Watson', '120', '50'])
12
13 db.select('classroom', '*')
14
```

1ο Παράδειγμα εκτέλεσης:

```
C:\Users\User\Desktop\miniDB-master\venv\Scripts\python.exe C:/Users/User/Desktop/miniDB-master/demo_database.py
Run 'create Database' method as:
1 --> Administrator
2 --> User
Your Answer: 1
Invalid answer. Please try again.
Your answer: sagehitsofh
Invalid answer. Please try again.
Your answer: 1
New table "meta_length"
New table "meta_locks"
New table "meta_insert_stack"
New table "meta_indexes"
Run 'create table' method as:
1 --> Administrator
2 --> User
Your Answer: 1
New table "classroom"
Run 'insert' method as:
1 --> Administrator
2 --> User
Your Answer: 1
Run 'insert' method as:
1 --> Administrator
2 --> User
Run 'insert' method as:
1 --> Administrator
2 --> User
Your Answer: 1
Run 'insert' method as:
1 --> Administrator
2 --> User
Your Answer: 1

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

Process finished with exit code 0
```

Στην αρχή του παραπάνω παραδείγματος δοκιμάζεται η εισαγωγή string διαφορετικού του '1' και '2' που είναι οι αποδεκτές απαντήσεις, με αποτέλεσμα η ερώτηση να επαναληφθεί. Στην συνέχεια τρέχουμε το υπόλοιπο πρόγραμμα απαντώντας πως είμαστε Administrator, με αποτέλεσμα οι υπόλοιπες "μέθοδοι" να εκτελεσθούν φυσιολογικά.

Στο παρακάτω παράδειγμα όμως, αν επιλέξουμε να τρέξουμε την "μέθοδο" create_database ως χρήστης, βγαίνει στην συνέχεια error καθώς δεν δημιουργήθηκε βάση εξ αρχής:

```
C:\Users\User\Desktop\miniDB-master\venv\Scripts\python.exe C:/Users/User/Desktop/miniDB-master/demo_database.py
Run 'create Database' method as:
1 --> Administrator
2 --> User
Your Answer: 2
You need Admin privileges to run this method.
Run 'create table' method as:
1 --> Administrator
2 --> User
Your Answer: 1
New table "classroom"
Traceback (most recent call last):
  File "C:/Users/User/Desktop/miniDB-master/demo_database.py", line 5, in <module>
    db.create_table('classroom', ['building', 'room_number', 'capacity'], [str,str,int])
  File "C:/Users/User/Desktop/miniDB-master/database.py", line 122, in create_table
    self.update()
  File "C:/Users/User/Desktop/miniDB-master/database.py", line 96, in _update
    self.update_meta_length()
  File "C:/Users/User/Desktop/miniDB-master/database.py", line 496, in _update_meta_length
    if table._name not in self.meta_length.table_name: # if new table, add record with 0 no. of rows
AttributeError: 'Database' object has no attribute 'meta_length'

Process finished with exit code 1
```

Αντίστοιχο παράδειγμα για την παράλειψη δημιουργίας πίνακα:

```
C:\Users\User\Desktop\miniDB-master\venv\Scripts\python.exe C:/Users/User/Desktop/miniDB-master/demo_database.py
Run 'create Database' method as:
1 --> Administrator
2 --> User
Your Answer: 2
New table "meta_length"
New table "meta_locks"
New table "meta_insert_stack"
New table "meta_indexes"
Run 'create table' method as:
1 --> Administrator
2 --> User
Your Answer: 2
You need Admin privileges to run this method.
Run 'insert' method as:
1 --> Administrator
2 --> User
Your Answer: 2
You need Admin privileges to run this method.
Run 'insert' method as:
1 --> Administrator
2 --> User
Your Answer: 2
You need Admin privileges to run this method.
Run 'insert' method as:
1 --> Administrator
2 --> User
Your Answer: 2
You need Admin privileges to run this method.
Run 'insert' method as:
1 --> Administrator
2 --> User
Your Answer: 2
You need Admin privileges to run this method.
Traceback (most recent call last):
  File "C:/Users/User/Desktop/miniDB-master/demo_database.py", line 13, in <module>
    db.select('classroom', '*')
  File "C:/Users/User/Desktop/miniDB-master/databse.py", line 363, in select
    table = self.tables[table_name]._select_where(columns, condition, order_by, asc, top_k)
KeyError: 'classroom'

Process finished with exit code 1
```