ΕΡΓΑΣΙΑ ΜΑΘΗΜΑΤΟΣ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τμήμα Πληροφορικής

Μάθημα: «ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΗΣΗΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ(5ο εξ.)»

Π18101 – ΑΝΑΣΤΑΣΙΑ ΙΩΑΝΝΑ ΜΕΞΑ

Π18078 – ΑΘΑΝΑΣΙΑ ΚΟΜΜΑΤΙΔΟΥ

Π18123 – ΒΑΣΙΛΙΚΗ ΠΑΣΙΑ

Ομάδα εργασιών 3 (Γενικά - Διάφορα)

Task 3.3 - SQL compiler (**)

Έχουμε φτιάξει το αρχείο sql_compiler.py στο οποίο υλοποιούνται όλα τα ζητούμενα της εργασίας. Συγκεκριμένα φτιάχνουμε 2 ενδεικτικούς πίνακες στην βάση, τον classroom και restroom. Ο classroom έχει τα πεδία [Building, Room number, Capacity] με πρωτεύον κλειδί το Room number, ενώ ο restroom έχει τα πεδία [Building, Gender] χωρίς να έχει πρωτεύον κλειδί. Αναλυτικά η δημιουργία των πινάκων μαζί με τις εγγραφές τους φαίνονται στο παρακάτω screenshot:

```
# Creating the tables
db.create_table('classroom', ['building', 'room_number', 'capacity'], [str,str,int], primary_key='room_number')
# insert 5 rows
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.insert('classroom', ['Watson', '120', '50'])

db.create_table('restroom', ['building', 'gender'], [str,str])
# insert 5 rows
db.insert('restroom', ['Packard', 'male'])
db.insert('restroom', ['Painter', 'female'])
db.insert('restroom', ['Taylor', 'male'])
db.insert('restroom', ['Watson', 'male'])
db.insert('restroom', ['Watson', 'female'])
```

Σημαντικές σημειώσεις:

- Κανονικά στην SQL μπορούμε να γράψουμε και κεφαλαία και μικρά γράμματα (είναι case insensitive). Εμείς έχουμε υλοποιήσει μόνο για μικρά γράμματα, δηλαδή είναι case sensitive. Επίσης, δεν υποστηρίζουμε την δυνατότητα να γράψουμε μια εντολή σε διαφορετικές γραμμές και δεν χρειαζόμαστε το ; στο τέλος κάθε εντολής, από την στιγμή που οι εντολές πρέπει να γράφονται στην ίδια γραμμή. Τέλος, αντί για varchar που χρησιμοποιεί η SQL για τα αλφαριθμητικά, εμείς γράφουμε str.

- Οπουδήποτε χρησιμοποιούνται operators δεν έχουμε αλλάξει το == σε = όπως είναι κανονικά στην SQL. Για παράδειγμα, κανονικά στην SQL θα γράφαμε

  ```
  UPDATE table_name
  SET column1 = value1, column2 = value2, ...
  WHERE condition;
  ```

  ενώ τώρα πρέπει να γράψουμε

  ```
  update table_name set column1 == value1, column2 == value2, … where condition
  ```

- Κανονικά στην SQL όπου θέλουμε να αναφερθούμε σε ένα στοιχείο εγγραφής που είναι varchar, βάζουμε ''. Εδώ δεν το έχουμε υλοποιήσει αυτό, επομένως γράφουμε χωρίς ''. Για παράδειγμα, κανονικά στην SQL θα γράφαμε

```
INSERT INTO Customers
VALUES ('Cardinal', 'Stavanger', 'Norway');
```

ενώ τώρα πρέπει να γράψουμε

```
insert into Customers values (Cardinal, Stavanger, Norway)
```

- Κανονικά στην SQL όπου θέλουμε να αναφερθούμε σε μια στήλη ενός πίνακα (π.χ. στα joins) γράφαμε το όνομα του πίνακα, τελεία (.) και το όνομα της στήλης. Εδώ αντί για τελεία (.) χρησιμοποιούμε την κάτω παύλα (_). Για παράδειγμα, κανονικά στην SQL θα γράφαμε

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

ενώ τώρα πρέπει να γράψουμε

```
select column_name(s) from table1 inner join table2 on table1_column_name
== table2_column_name
```

Όταν τρέξει ο χρήστης το πρόγραμμα του εμφανίζεται ένα μήνυμα, που τον ενημερώνει ότι μπορεί να γράψει την εντολή SQL της επιλογής του και ότι αν θέλει να τερματίσει το πρόγραμμα, μπορεί να δώσει ως είσοδο την λέξη ‘exit’.

```
Write an sql command (low case letters only!)
If you want to terminate the program, write 'exit'
```

```
Write an sql command (low case letters only!)
If you want to terminate the program, write 'exit'

exit
Deleted 1 rows
Deleted 1 rows
Deleted 1 rows

Process finished with exit code 0
```

Αν ο χρήστης γράψει λάθος την εντολή που δώσει ως είσοδο, θα του εμφανιστεί κατάλληλο μήνυμα και μπορεί να ξαναπροσπαθήσει.

```
Write an sql command (low case letters only!)
If you want to terminate the program, write 'exit'

selct * from classroom

Wrong syntax!

Write an sql command (low case letters only!)
If you want to terminate the program, write 'exit'

```

1. Select From Where και Select From Join

   Στην εντολή select, έχουμε υλοποιήσει όλα τα πιθανά σενάρια. Δηλαδή και αν γίνεται χρήση των where condition, into (save as), order by, asc/desc και top καθώς και inner join. Παρακάτω υπάρχουν screenshots που δείχνουν την επιτυχή εκτέλεση τριών διαφορετικών και παράλληλα σύνθετων εντολών select.

```
Write an sql command (low case letters only!)
If you want to terminate the program, write 'exit'

select building, room_number into table3 from classroom where building == Watson order by room_number asc

## table3 ##
building (str)      room_number (str) #PK#
---------------  ----------------------
Watson                             100
Watson                             120
```

```
Write an sql command (low case letters only!)
If you want to terminate the program, write 'exit'

select top 3 building, room_number from classroom

## classroom ##
building (str)      room_number (str) #PK#
----------------  ------------------------
Packard                             101
Painter                             514
Taylor                             3128
```

```
Write an sql command (low case letters only!)
If you want to terminate the program, write 'exit'

select restroom_gender, classroom_room_number from classroom inner join restroom on building == building
## Select ops no. -> 25
# Left table size -> 5
# Right table size -> 5

## testtable ##
restroom_gender (str)      classroom_room_number (str)
----------------------  ----------------------------
male                                             101
female                                           514
male                                            3128
male                                             100
female                                           100
male                                             120
female                                           120

Deleted 1 rows
Deleted 1 rows
Deleted 1 rows
```

Πρέπει να εστιάσουμε λίγο στην περίπτωση του select from join (το 3ο από τα παραπάνω screenshots). Από την στιγμή που για το inner join πρέπει να χρησιμοποιήσουμε την εντολή select, αποφασίσαμε να τα εντάξουμε στο ίδιο κομμάτι κώδικα, ώστε να μην επαναλαμβάνεται κώδικας χωρίς λόγο. Άρα, καλούμε πρώτα μέσα στον κώδικά μας την συνάρτηση db.inner_join() δίνοντας ως ορίσματα τον αριστερό πίνακα, τον δεξί πίνακα, την συνθήκη του join και το string "testtable", όπου είναι το όνομα το πίνακα που θα αποθηκευτεί το αποτέλεσμα του inner join. Έπειτα, εκτελούμε το κομμάτι του select στον πίνακα testtable και μόλις ολοκληρωθεί η διαδικασία, διαγράφουμε από την βάση τον πίνακα testtable.

```
if "inner join" in after_from[1]: # There is inner join in the command
    in_join = after_from[1].split("inner join") # in_join = ['left table', 'right table and condition']
    left_table = in_join[0].replace(" ", "") # in_join[0] = 'left table'
    in_join_temp = in_join[1].split("on") # in_join_temp = ['right table', 'condition']
    right_table = in_join_temp[0].replace(" ", "") # in_join_temp[0] = 'right table'
    in_join_condition = in_join_temp[1].replace(" ", "") # in_join_temp[1] = 'condition'
    db.inner_join(left_table, right_table, in_join_condition, "testtable")
    table = "testtable" # the name of the table, result of inner join
```

```
# Drop the table formed as a result of inner join
if table == "testtable":
    db.drop_table("testtable")
```

2. Update Where

Έχουμε υλοποιήσει την εντολή update where, όπου μπορείτε να αλλάξετε την τιμή ενός πεδίου μιας εγγραφής. Δεν γίνεται να αλλάξετε 2 πεδία της ίδιας εγγραφής με μία εντολή update, θα πρέπει να γίνει με 2 ξεχωριστές εντολές. Για παράδειγμα, έστω ότι θέλουμε να αλλάξουμε στον πίνακα classroom το capacity = 2130 και room_number = 150 όπου building = Taylor. Η εντολή *update classroom set capacity == 2130, room_number == 150 where building == Taylor* δεν θα λειτουργήσει. Πρέπει να εκτελέσουμε ξεχωριστά τις παρακάτω εντολές:

update classroom set capacity == 2130 where building == Taylor

update classroom set room_number == 150 where building == Taylor

Το αποτέλεσμα των εντολών αυτών, είναι ίδιο με το αν έτρεχε η εντολή *update classroom set capacity == 2130, room_number == 150 where building == Taylor* και φαίνεται στο παρακάτω screenshot:

```
Write an sql command (low case letters only!)
If you want to terminate the program, write 'exit'

update classroom set room_number == 150 where building == Taylor

## classroom ##
building (str)      room_number (str) #PK#    capacity (int)
---------------  ------------------------  ----------------
Packard                              101                500
Painter                              514                 10
Taylor                               150               2130
Watson                               100                 30
Watson                               120                 50
```

3. Insert Into

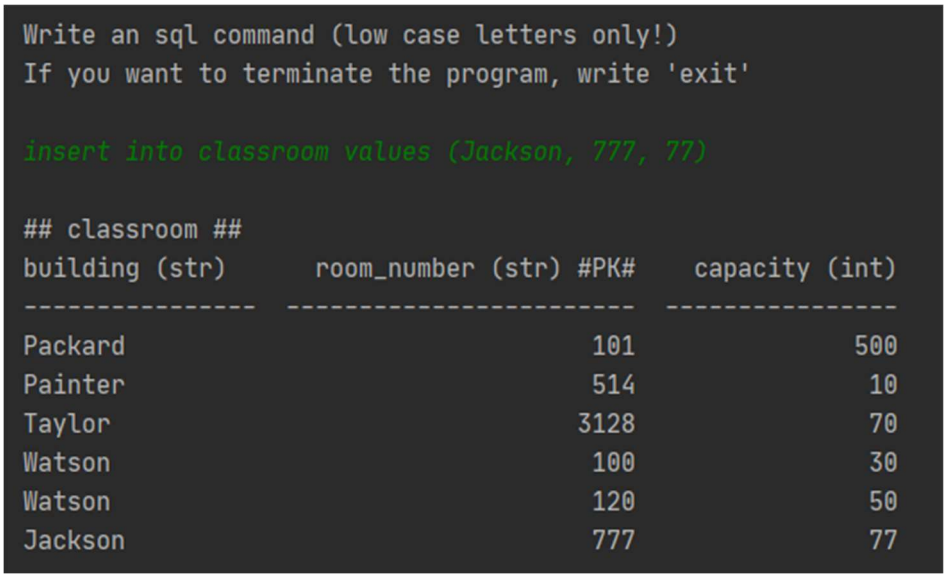
Έχουμε υλοποιήσει την εντολή insert into, όπου μπορείτε να εισάγετε εγγραφές στους πίνακες της βάσεις. Πρέπει να σημειωθεί, ότι από την στιγμή και στην υλοποίησή σας στην Python πρέπει κάθε φορά να εισαχθεί μια τιμή σε όλα τα πεδία του πίνακα, δεν χρειάζεται να γράψετε την εντολή insert into σε αύτη την μορφή

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

δεν το υποστηρίζουμε. Γράφουμε κατευθείαν

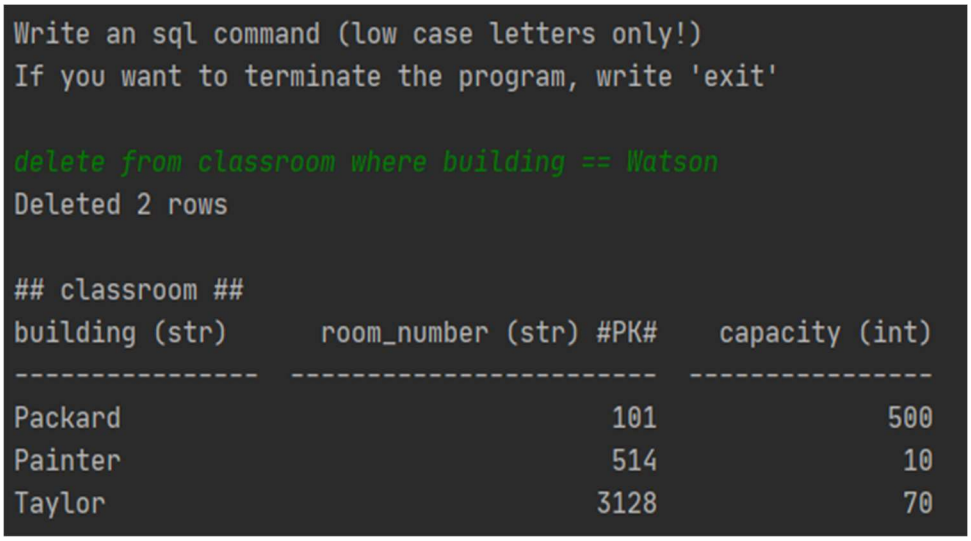
```
insert into table_name values (value1, value2, value3, ...)
```

Για παράδειγμα, στο παρακάτω screenshot φαίνεται η εκτέλεση της εντολής `insert into classroom values (Jackson, 777, 77)`:



4. *Delete From Where*

Έχουμε υλοποιήσει την εντολή delete from where, όπου μπορείτε να διαγράψετε μια εγγραφή από έναν πίνακα στην βάση. Στο παρακάτω screenshot φαίνεται η εκτέλεση την ενδεικτικής εντολής `delete from classroom where building == Watson`:



5. Create/Drop Table

Έχουμε υλοποιήσει την εντολή create table και drop table, όπου μπορείτε να δημιουργήσετε και να διαγράψετε αντίστοιχα κάποιον πίνακα. Ξεκινώντας με την create table, έχουμε υλοποιήσει και την περίπτωση που ο πίνακας δεν θα έχει πρωτεύον κλειδί και την περίπτωση που θα έχει. Στα παρακάτω screenshots φαίνονται παραδείγματα και για τις δύο περιπτώσεις:

```
Write an sql command (low case letters only!)
If you want to terminate the program, write 'exit'

create table department (dept_name str, building str, budget int, primary key (dept_name))
New table "department"
```

Με πρωτεύον κλειδί

```
Write an sql command (low case letters only!)
If you want to terminate the program, write 'exit'

create table department (dept_name str, building str, budget int)
New table "department"
```

Χωρίς πρωτεύον κλειδί

Συνεχίζοντας με την drop table, την έχουμε υλοποιήσει και στο παρακάτω screenshot φαίνεται ένα παράδειγμα:

```
Write an sql command (low case letters only!)
If you want to terminate the program, write 'exit'

drop department
Deleted 1 rows
Deleted 1 rows
Deleted 1 rows
```

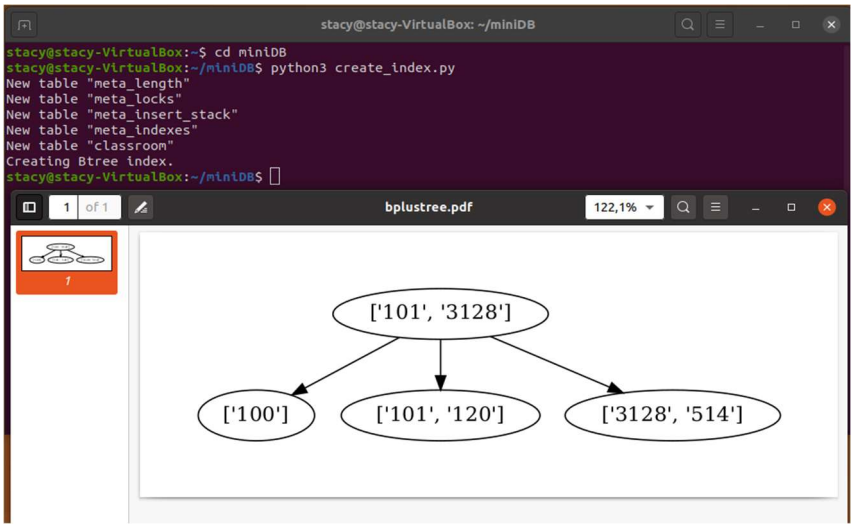
6. *Create Database*

Έχουμε υλοποιήσει την εντολή create database, όπου μπορείτε να δημιουργήσετε μια καινούργια βάση. Στο παρακάτω screenshot φαίνεται ένα παράδειγμα:

```
Write an sql command (low case letters only!)
If you want to terminate the program, write 'exit'

create database testDB
New table "meta_length"
New table "meta_locks"
New table "meta_insert_stack"
New table "meta_indexes"
```

7. *Create Index On*

Έχουμε υλοποιήσει την εντολή create index on, όπου μπορείτε να δημιουργήσετε ένα καινούργιο index. Στο παρακάτω screenshot φαίνεται ένα παράδειγμα, που τρέχουμε την εντολή `create index index1 on classroom`: (Το screenshot είναι από τα Linux, δεν καταφέραμε να το κάνουμε να τρέχει σε Windows…)



Ομάδα εργασιών 2 (Διαχείριση Δοσοληψιών, Αρχιτεκτονικές ΣΔΒΔ)

Task 2.4 - Server - Client (***)

Σας είχαμε πει στο email που στείλαμε για να μας ανατεθεί θέμα για την εργασία, ότι θα προσπαθήσουμε να ασχοληθούμε και με το task 2.4. Καταφέραμε να δημιουργήσουμε σύνδεση μεταξύ server και client. Πιο συγκεκριμένα, έχουμε δυο αρχεία python τα client.py και server.py τα οποία μπορούν να υλοποιήσουν μόνο την λειτουργία drop table. Στο αρχείο του server φτιάχνουμε τον ίδιο πίνακα classroom στην αρχή και εκεί περιέχεται ο κώδικας για την εντολή drop. Ενώ στο αρχείο του client το μόνο που υπάρχει, είναι να δέχεται input από τον χρήστη την εντολή drop. Πρέπει πρώτα να τρέξει ο server και εμφανίζει το εξής μήνυμα:

```
Please run client
```

Έπειτα, πρέπει να τρέξει ο client και εμφανίζεται το εξής:

```
Write query
```

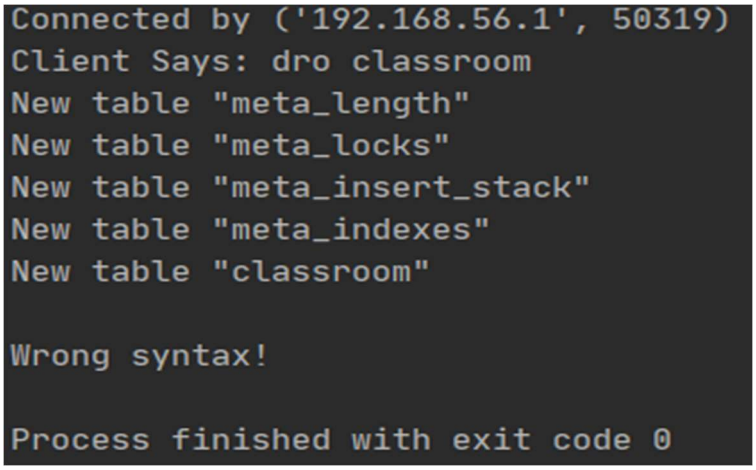
Άρα τώρα, μπορούμε να γράψουμε την εντολή `drop classroom`. Τα αποτελέσματα και για τον server και για τον client φαίνονται στα screenshots παρακάτω:

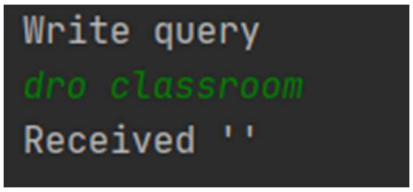
```
Please run client
Connected by ('192.168.56.1', 50250)
Client Says: drop classroom
New table "meta_length"
New table "meta_locks"
New table "meta_insert_stack"
New table "meta_indexes"
New table "classroom"
Deleted 1 rows
Deleted 1 rows
Deleted 1 rows

Process finished with exit code 0
```

```
Write query
drop classroom
Received 'Done'
```

Αποτελέσματα server Αποτελέσματα client

Αν ο χρήστης δώσει λάθος input τότε του εμφανίζεται αντίστοιχο μήνυμα:





Αποτελέσματα server Αποτελέσματα client

Θα πρέπει να ξανατρέξει πρώτα τον server και μετά τον client και να ξαναπροσπαθήσει.