ΕΡΓΑΣΙΑ ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΘΑΝΑΣΙΟΣ ΡΟΥΒΙΝΕΤΗΣ **Π19145**

ΝΙΚΟΣ ΦΑΚΑΣ **Π19178**

ΠΕΡΙΟΕΧΟΜΕΝΑ

1)Ενότητα 1η

1.1) Κώδικας εργασίας

Database.py:

```
self.tb_name = name
self.tb2_name = name
self.str_con = name
self.action = name
```

Το συγκεκριμένο έχει προσθεθεί στο def __init__

```
def create_trigger(self, test, t1, t2, condition):
    """
    Create trigger that is going to execute an insert into a
table after the insert in a table
    Args:
      test: has the all the sentence that the user gave for the
trigger function
        t1: variable that is being used for the function to
take values
        t2: variable that is being used for the function to
take values
        condition: variable that is being used for the function
to take values and will be used to check the
        condition of the args that user gives

    """
    test1 = test.split()
    self.tb_name = test1[4]
    self.tb2_name = test1[8]
    self.str_con = condition
    self.action = test1[1]

def insert_into(self, table_name, row_str):
    '''
    Inserts data to given table.

    Args:
        table_name: string. Name of table (must be part of
database).
        row: list. A list of values to be inserted (will be
casted to a predifined type automatically).
        lock_load_save: boolean. If False, user needs to load,
lock and save the states of the database (CAUTION).
        Useful for bulk-loading.
    '''
    table_n2 = self.tb2_name
    con = self.str_con

    if self.action == 'before':

        if table_name == self.tb_name:
            print("you are in the trigger function")
            self.delete_from(table_n2, con)
            self.action = ''
```

```
            self.tb_name = ''

        row = row_str.strip().split(',')
        self.load_database()
        # fetch the insert_stack. For more info on the
insert_stack
        # check the insert_stack meta table
        lock_ownership = self.lock_table(table_name, mode='x')
        insert_stack =
self._get_insert_stack_for_table(table_name)

        try:
            self.tables[table_name]._insert(row, insert_stack)
        except Exception as e:
            logging.info(e)
            logging.info('ABORTED')
        self._update_meta_insert_stack_for_tb(table_name,
insert_stack[:-1])

        if lock_ownership:
            self.unlock_table(table_name)
        self._update()
        self.save_database()

    elif self.action == "instead":

        if table_name == self.tb_name:
            print("you are in the trigger function")
            self.delete_from(table_n2, con)
            self.action = ''
            self.tb_name = ''

        else:
            row = row_str.strip().split(',')
            self.load_database()
            # fetch the insert_stack. For more info on the
insert_stack
            # check the insert_stack meta table
            lock_ownership = self.lock_table(table_name,
mode='x')
            insert_stack =
self._get_insert_stack_for_table(table_name)

            try:
                self.tables[table_name]._insert(row,
insert_stack)
            except Exception as e:
                logging.info(e)
                logging.info('ABORTED')
            self._update_meta_insert_stack_for_tb(table_name,
insert_stack[:-1])

            if lock_ownership:
                self.unlock_table(table_name)
            self._update()
            self.save_database()

    elif self.action == 'after':
```

```
        row = row_str.strip().split(',')
        self.load_database()
        # fetch the insert_stack. For more info on the
insert_stack
        # check the insert_stack meta table
        lock_ownership = self.lock_table(table_name, mode='x')
        insert_stack =
self._get_insert_stack_for_table(table_name)

        try:
            self.tables[table_name]._insert(row, insert_stack)
        except Exception as e:
            logging.info(e)
            logging.info('ABORTED')
        self._update_meta_insert_stack_for_tb(table_name,
insert_stack[:-1])

        if lock_ownership:
            self.unlock_table(table_name)
        self._update()
        self.save_database()

        if table_name == self.tb_name:
            print("you are in the trigger function")
            self.delete_from(table_n2, con)
            self.action = ''
            self.tb_name = ''

    else:

        row = row_str.strip().split(',')
        self.load_database()
        # fetch the insert_stack. For more info on the
insert_stack
        # check the insert_stack meta table
        lock_ownership = self.lock_table(table_name, mode='x')
        insert_stack =
self._get_insert_stack_for_table(table_name)

        try:
            self.tables[table_name]._insert(row, insert_stack)
        except Exception as e:
            logging.info(e)
            logging.info('ABORTED')
        self._update_meta_insert_stack_for_tb(table_name,
insert_stack[:-1])



        if lock_ownership:
            self.unlock_table(table_name)
        self._update()
        self.save_database()
```

mdp.py:

```
'create trigger': ['create trigger', 'after insert on',
'execute delete from', 'where']
```

Το συγκεκριμένο έχει προσθεθεί στο def interpret

2)Ενότητα 2η

2.1) Επεξήγηση κώδικα

Στην συγκεκριμένη διαδικασία είμαστε υποχρεωμένη να δημιουργήσουμε και να εκτελέσουμε μια διαδικασία trigger. Για την εκτέλεση της διαδικασίας πραγματοποιήσαμε αλλαγές κυρίως στο αρχείο database.py, με την αλλαγή στην συνάρτηση insert_into και την δημιουργία της συνάρτησης create_trigger.

Το συγκεκριμένο trigger είναι υλοποιήσιμο για να κάνει διαγραφεί από κάποιο πίνακα όταν δοθεί η εντολή εισαγωγής. Η διαδικασία μπορεί να εκτελεστεί τόσο πριν την εντολή εισαγωγής αλλά τόσο αντί αυτής της εντολής και ύστερα της εντολής

Η συνάρτηση create_trigger δημιουργήθηκε για την ανάκτηση των δεδομένων όπου χρειαζόμαστε για την εκτέλεση της διαδικασίας όπου ζητείται με το create_trigger. Λαμβάνει δεδομένα από τον χρήστη στην μεταβλητή test και στην μεταβλητή condition όπου περιέχει τα δεδομένα που είναι αναγκαία ώστε να γίνει η διαγραφή ενός στοιχείου. Χρησιμοποιούμε την εντολή διάσπασης split ώστε να χωρίσουμε τις λέξεις όπου έδωσε ο χρήστης για πάρουμε τα ακριβείς δεδομένα που είναι απαραίτητα για την υλοποίηση της διαδικασίας. Αφού γίνει η διάσπαση καταχωρούμε τις τιμές μέσα στις μεταβλητές self.tb_name , self.tb2_name , self.str_con και self.action στα οποία περνάμε το όνομα του πίνακα όπου θα είναι για το insert , το όνομα του πίνακα όπου θα γίνει το delete , τα στοιχεία που χρειάζεται για να γίνει ο έλεγχος για την διαγραφή αλλά και πότε θέλει ο χρήστης να γίνει η διαδικασία αντίστοιχα. Με την χρήση των μεταβλητών self θα μπορούμε να τις περνάμε και στις υπόλοιπες συναρτήσεις ώστε να γίνουν οι διαδικασίες όπου χρειάζονται για την υλοποίηση του trigger.

Στην συνάρτηση insert_into πραγματοποιήθηκαν αλλαγές για τη εκτέλεση του trigger χωρίς να αλλάξουμε την διαδικασία της εισαγωγής δεδομένων, η διαδικασία εισαγωγής έχει εισχωρήσει μέσα στου ελέγχους ώστε να μην επηρεαστεί με το trigger που υλοποιείτε.

Στην insert_into πλέον περιλαμβάνεται από if όπου γίνεται έλεγχος για το trigger. Στο if γίνεται έλεγχος για το αν το self.action είναι after , before , instead αλλιώς αν δεν ισχύει κάτι από τα προηγούμενα κάνει απλά την διαδικασία του insert into. Έτσι αν δεν έχει εκτελεστεί η συνάρτηση trigger θα λειτουργήσει με τους ήδη υπάρχοντας δεδομένα που είχε το πρόγραμμα πριν την παρεμβολή μας αφού ο κώδικας δεν έχει αλλάξει. Στην περίπτωση που εισέλθει σε κάποιο if του action τότε ανάλογα σε ποιο είναι κάνει την διαδικασία όπου έχει ανατεθεί μέσα στο if . Στην περίπτωση όπου το action = before τότε εκτελεί πρώτα ένα if όπου ελέγχει αν ο πίνακας που γίνεται η εισαγωγή στοιχείων είναι ο πίνακας που περάσαμε στο self.tb_name στην περίπτωση που είναι τότε καλεί την συνάρτηση delete με δεδομένα το table_name2 και το con όπου έχουμε περάσει τα self.tb_name2 και το self.str_con αντίστοιχα, αλλάζει μετά και τις τιμές self.action και self.tb_name ώστε να μην ξανά μπει στο συγκεκριμένο if. Ύστερα εκτελεί την διαδικασία insert ΄όπως είναι κανονικά. Η ίδια ακριβώς διαδικασία γίνεται και στο if action = after με μοναδική διαφορά να το if όπου κάνει τον έλεγχο αν ο πίνακας που γίνεται η εισαγωγή στοιχείων είναι ο πίνακας που περάσαμε στο self.tb_name βρίσκετε στο τέλος και πρώτα γίνεται ο κώδικας του insert και μετά γίνεται η διαδικασία του if . Στην περίπτωση όταν το if είναι action = instead τότε βλέπουμε πως μέσα στο if περιέχει μόνο το δεύτερο if όπου κάνει τον έλεγχο αν ο πίνακας που γίνεται η εισαγωγή στοιχείων είναι ο πίνακας που περάσαμε στο self.tb_name και καλεί το delete με τα στοιχεία που έχει δεχτεί από τα self.tb_name2 και το self.str_con αντίστοιχα, αλλάζει μετά και τις τιμές self.action και self.tb_name ώστε να μην ξανά μπει στο συγκεκριμένο if. Στην περίπτωση που δεν ισχύει το if τότε έχουμε το else που εκτελεί κανονικά τις εντολές το insert.

Έτσι υλοποιείτε η διαδικασία trigger όταν θέλουμε να κάνουμε delete στην περίπτωση που γίνει ένα συγκεκριμένο insert σε ένα πίνακα.

3)Ενότητα 3η

3.1) Εκτέλεση εργασίας

```
(smdb)> SELECT * FROM student;
  id (str) #PK#  name (str)    dept_name (str)      tot_cred (int)
--------------  -----------  ----------------  ---------------
         00128  zhang        comp. sci.                      102
         12345  shankar      comp. sci.                       32
         19991  brandt       history                          80
         23121  chavez       finance                         110
         44553  peltier      physics                          56
         45678  levy         physics                          46
         54321  williams     comp. sci.                       54
         55739  sanchez      music                            38
         70557  snow         physics                           0
         76543  brown        comp. sci.                       58
         76653  aoi          elec. eng.                       60
         98765  bourikas     elec. eng.                       98
         98988  tanaka       biology                         120
         14564  deland       acos.                           143
         10005  lane         comp. sci.                      187
         12543  jenner       biology                         107
         12987  jenner       history                         109
         12786  toublis      elec. eng.                      189
         12776  toublia      elec. eng.                      169
         19876  harvy        bio. tech.                      189
         27890  jerald       tech.                           289
         27810  seretis      tech.                           209
         25432  june         teacher                         345
         12349  ronnald      biology                          21
         16453  bond         biotech                         100
         14587  lordi        accountant                      103
         14568  lord         accountant                      176
         19209  pord         biotech                          33

(smdb)> create trigger sttrigger instead insert on student execute delete from student where id=19209
(smdb)>
```

```
(smdb)> insert into student values(19220,Tailor,sci.,33)
you are in the trigger function
(smdb)> SELECT * FROM student;
  id (str) #PK#  name (str)    dept_name (str)      tot_cred (int)
--------------  -----------  ----------------  ---------------
         00128  zhang        comp. sci.                      102
         12345  shankar      comp. sci.                       32
         19991  brandt       history                          80
         23121  chavez       finance                         110
         44553  peltier      physics                          56
         45678  levy         physics                          46
         54321  williams     comp. sci.                       54
         55739  sanchez      music                            38
         70557  snow         physics                           0
         76543  brown        comp. sci.                       58
         76653  aoi          elec. eng.                       60
         98765  bourikas     elec. eng.                       98
         98988  tanaka       biology                         120
         14564  deland       acos.                           143
         10005  lane         comp. sci.                      187
         12543  jenner       biology                         107
         12987  jenner       history                         109
         12786  toublis      elec. eng.                      189
         12776  toublia      elec. eng.                      169
         19876  harvy        bio. tech.                      189
         27890  jerald       tech.                           289
         27810  seretis      tech.                           209
         25432  june         teacher                         345
         12349  ronnald      biology                          21
         16453  bond         biotech                         100
         14587  lordi        accountant                      103
         14568  lord         accountant                      176
```

```
(smdb)> create trigger sttrigger after insert on student execute delete from student where id=16453
(smdb)> insert into student values(19221,keren,comp. sci.,180)
you are in the trigger function
(smdb)> SELECT * FROM student;
  id (str) #PK#  name (str)    dept_name (str)      tot_cred (int)
--------------  ------------  -----------------  ----------------
         00128  zhang         comp. sci.                      102
         12345  shankar       comp. sci.                       32
         19991  brandt        history                          80
         23121  chavez        finance                         110
         44553  peltier       physics                          56
         45678  levy          physics                          46
         54321  williams      comp. sci.                       54
         55739  sanchez       music                            38
         70557  snow          physics                           0
         76543  brown         comp. sci.                       58
         76653  aoi           elec. eng.                       60
         98765  bourikas      elec. eng.                       98
         98988  tanaka        biology                         120
         14564  deland        acos.                           143
         10005  lane          comp. sci.                      187
         12543  jenner        biology                         107
         12987  jenner        history                         109
         12786  toublis       elec. eng.                      189
         12776  toublia       elec. eng.                      169
         19876  harvy         bio. tech.                      189
         27890  jerald        tech.                           289
         27810  seretis       tech.                           209
         25432  june          teacher                         345
         12349  ronnald       biology                          21
         14587  lordi         accountant                      103
         19221  keren         comp. sci.                      180
         19220  tailor        sci.                             33
```

```
(smdb)> create trigger sttrigger after insert on student execute delete from student where id=14568
(smdb)> insert into student values(19220,Tailor,sci.,33)
you are in the trigger function
(smdb)> SELECT * FROM student;
  id (str) #PK#  name (str)    dept_name (str)      tot_cred (int)
--------------  -----------  ----------------  ---------------
         00128  zhang        comp. sci.                      102
         12345  shankar      comp. sci.                       32
         19991  brandt       history                          80
         23121  chavez       finance                         110
         44553  peltier      physics                          56
         45678  levy         physics                          46
         54321  williams     comp. sci.                       54
         55739  sanchez      music                            38
         70557  snow         physics                           0
         76543  brown        comp. sci.                       58
         76653  aoi          elec. eng.                       60
         98765  bourikas     elec. eng.                       98
         98988  tanaka       biology                         120
         14564  deland       acos.                           143
         10005  lane         comp. sci.                      187
         12543  jenner       biology                         107
         12987  jenner       history                         109
         12786  toublis      elec. eng.                      189
         12776  toublia      elec. eng.                      169
         19876  harvy        bio. tech.                      189
         27890  jerald       tech.                           289
         27810  seretis      tech.                           209
         25432  june         teacher                         345
         12349  ronnald      biology                          21
         16453  bond         biotech                         100
         14587  lordi        accountant                      103
         19220  tailor       sci.                             33
```