



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΗΜΜΥ

ΕΡΓΑΣΤΗΡΙΟ ΔΙΑΝΕΜΗΜΕΝΩΝ

ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΕΦΑΡΜΟΓΩΝ

ΙΣΤΟΧΩΡΟΣ ΜΑΘΗΜΑΤΟΣ <https://www.eclass.tuc.gr/courses/HMMY167/>

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ - ΠΛΗ303

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

ΥΠΟΣΤΗΡΙΞΗ ΕΡΓΑΣΤΗΡΙΟΥ: Καζάσης Φώτης, Παππάς Νίκος

5^η εργαστηριακή άσκηση

Στόχος

Ο στόχος της άσκησης είναι η εξάσκηση στη δημιουργία και χρήση αποθηκευμένων συναρτήσεων. Στόχος είναι επίσης η χρήση συναρτήσεων σε CHECK περιορισμούς και η δημιουργία συναρτήσεων ενανυσμάτων στην PostgreSQL.

Αντικείμενο

A. Εξάσκηση σε συναρτήσεις

1. Δημιουργήστε τους πίνακες:
AB(id integer, item_type_id integer, item_type varchar(5))
typeA(typeA_id integer)
typeB(typeB_id integer)

Επιθυμούμε σε κάθε πλειάδα το πεδίο item_type_id να αναφέρεται (να δείχνει) στο πεδίο typeA_id του πίνακα typeA όταν η τιμή του πεδίου item_type στην πλειάδα είναι 'Α' και στο πεδίο typeB_id του πίνακα typeB όταν η τιμή του πεδίου item_type στην πλειάδα είναι 'Β'. Υλοποιήστε τον παραπάνω περιορισμό με την δημιουργία ενός check constraint στον πίνακα AB που θα ελέγχει την ορθότητα των τιμών που εισάγονται ή ενημερώνονται στα πεδία item_type_id, item_type.

Υπόδειξη: δημιουργήστε μια boolean συνάρτηση check_type(id integer, tp varchar(5)) που θα επιστρέφει true ή false ανάλογα αν επιτρέπεται να εισαχθεί αυτός ο συνδυασμός τιμών στα πεδία item_type_id, item_type ή όχι. Εκτελέστε την κατάλληλη SQL εντολή (ALTER), ώστε να τροποποιήσετε τον πίνακα AB για να εισάγετε το CHECK CONSTRAINT που αξιοποιεί την προηγούμενη συνάρτηση.

2. Δημιουργήστε ένα πίνακα **Student** με πεδία (am, name, surname) που κρατά στοιχεία των φοιτητών. Δημιουργήστε επίσης τους πίνακες **Name** με πεδία (name, sex) και **Surname** με πεδίο (surname), που κρατούν δεδομένα ελληνικών ονομάτων (και φύλου) και επωνύμων αντίστοιχα. Εισάγετε τα δεδομένα των δύο αρχείων csv (names, surnames) που συνοδεύουν την άσκηση στους πίνακες Name και Surname αντίστοιχα. Η εισαγωγή μπορεί να γίνει είτε γραφικά μέσω της διαδικασίας import/export του εργαλείου pgAdmin είτε μέσω της εντολής Copy (<https://www.postgresql.org/docs/13/sql-copy.html>).



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΗΜΜΥ

ΕΡΓΑΣΤΗΡΙΟ ΔΙΑΝΕΜΗΜΕΝΩΝ

ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΕΦΑΡΜΟΓΩΝ

ΙΣΤΟΧΩΡΟΣ ΜΑΘΗΜΑΤΟΣ <https://www.eclass.tuc.gr/courses/HMMY167/>

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ - ΠΛΗ303

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

ΥΠΟΣΤΗΡΙΞΗ ΕΡΓΑΣΤΗΡΙΟΥ: Καζάσης Φώτης, Παππάς Νίκος

3. Εκτελέστε τον κώδικα που σας δίνεται στο Παράρτημα για να δημιουργήσετε τις συναρτήσεις `random_surnames()`, `random_names()`, `create_am()`, `adapt_surname()`. Μελετήστε τον κώδικα των συναρτήσεων αυτών και καλέστε τις ώστε να καταλάβετε την λειτουργία τους.
4. Κατασκευάστε μία συνάρτηση `create_students()` η οποία θα επιστρέφει αυτόματα η εγγραφές φοιτητών όπου το `n` θα είναι μια παράμετρος εισόδου της συνάρτησης. Η συνάρτηση αυτή θα δέχεται και μία δεύτερη παράμετρο εισόδου `year` η οποία αντιστοιχεί στο έτος εισαγωγής, κοινό για όλες τις εγγραφές φοιτητών που θα δημιουργηθούν, ώστε να συντίθεται αυτόματα και ο αριθμός μητρώου της μορφής EEEEEAAAAA όπου EEEE το έτος εγγραφής και AAAAAA η εξαψήφια μορφή του αύξοντα αριθμού κάθε φοιτητή στο ίδιο έτος εισαγωγής συμπληρώνοντας από δεξιά με μηδενικά ώστε να συμπληρωθούν τα έξι ψηφία. Για τη δημιουργία της ζητούμενης συνάρτησης μπορείτε να χρησιμοποιήσετε τις συναρτήσεις που δημιουργήσατε στο βήμα (3). Χρησιμοποιήστε κατάλληλα τη συνάρτηση `create_students()` για να εισάγετε 1000 πλειάδες στον πίνακα `Student`.

Β. Εξάσκηση στη δημιουργία εναυσμάτων και όψεων στην PostgreSQL

Δημιουργήστε ένα πίνακα `student_audit` με πεδία (`id`, `operation`, `operation_time`, `am`, `message`) που θα αποτελεί το αρχείο των λειτουργιών που λαμβάνουν χώρα στα δεδομένα των φοιτητών. Το πεδίο `id` του πίνακα `student_audit` αποτελεί πρωτεύον χαρακτηριστικό και κρατάει τον αύξοντα αριθμό της επιχειρούμενης ενέργειας. Το πεδίο `operation` θα κρατάει την ένδειξη της ενημέρωσης και θα παίρνει τις τιμές 'D', 'U', 'I' για τις λειτουργίες `delete`, `update` και `insert` αντίστοιχα. Το πεδίο `operation_time` θα κρατάει το χρονικό σημείο που έλαβε χώρα η ενέργεια. Τέλος, το πεδίο `am` θα κρατάει τον αριθμό μητρώου του φοιτητή τον οποίο αφορά η ενέργεια και το πεδίο `message` ένα μήνυμα όπως περιγράφεται παρακάτω.

Υλοποιήστε τους παρακάτω περιορισμούς (με χρήση `Triggers`) σχετικά με την εισαγωγή, διαγραφή και ενημέρωση των στοιχείων των φοιτητών:

- Κάθε ενέργεια διαγραφής φοιτητή θα αγνοείται και θα καταγράφεται στον πίνακα `student_audit` με μήνυμα "Deletion Attempted".
- Κατά την ενημέρωση στοιχείων φοιτητή θα καταγράφεται στο `student_audit` ως μήνυμα ολόκληρο το νέο ονοματεπώνυμο του φοιτητή. Η ενημέρωση στα στοιχεία του φοιτητή επιτρέπεται μόνο στο όνομα.



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΗΜΜΥ

ΕΡΓΑΣΤΗΡΙΟ ΔΙΑΝΕΜΗΜΕΝΩΝ

ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΕΦΑΡΜΟΓΩΝ

ΙΣΤΟΧΩΡΟΣ ΜΑΘΗΜΑΤΟΣ <https://www.eclass.tuc.gr/courses/HMMY167/>

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ - ΠΛΗ303

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

ΥΠΟΣΤΗΡΙΞΗ ΕΡΓΑΣΤΗΡΙΟΥ: Καζάσης Φώτης, Παππάς Νίκος

- Αν επιχειρηθεί αλλαγή στο επώνυμο, αυτή πρέπει να αγνοείται αλλά το μήνυμα στο student audit θα καταγράφεται κανονικά.
- Αν υπάρχει αλλαγή στον ΑΜ θα πρέπει να αγνοείται και στο student_audit θα καταγράφεται μήνυμα "AM update denied:" και θα ακολουθεί ο ΑΜ που αγνοήθηκε.
- Η εισαγωγή νέου φοιτητή επιτρέπεται μόνο αν ο ΑΜ αντιστοιχεί στο τρέχον έτος και θα καταγράφεται στο student_audit. Σε διαφορετική περίπτωση θα δημιουργείται κατάλληλο μήνυμα εξαίρεσης.



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΗΜΜΥ

ΕΡΓΑΣΤΗΡΙΟ ΔΙΑΝΕΜΗΜΕΝΩΝ

ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΕΦΑΡΜΟΓΩΝ

ΙΣΤΟΧΩΡΟΣ ΜΑΘΗΜΑΤΟΣ <https://www.eclass.tuc.gr/courses/HMMY167/>

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ - ΠΛΗ303

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

ΥΠΟΣΤΗΡΙΞΗ ΕΡΓΑΣΤΗΡΙΟΥ: Καζάσης Φώτης, Παππάς Νίκος

ΠΑΡΑΡΤΗΜΑ

(α) Συνάρτηση random_surnames η οποία επιστρέφει η τυχαία επώνυμο με κατάληξη σε -ΗΣ για να μπορέσουμε εύκολα να παράγουμε την θηλυκή εκδοχή τους

```
CREATE OR REPLACE FUNCTION random_surnames(n integer)
  RETURNS TABLE(surname character(50), id integer) AS
$$
BEGIN
  RETURN QUERY
  SELECT snam.surname, row_number() OVER ()::integer
  FROM (SELECT "Surname".surname
  FROM "Surname"
  WHERE right("Surname".surname,2)='ΗΣ'
  ORDER BY random() LIMIT n) as snam;
END;
$$
LANGUAGE 'plpgsql' VOLATILE;
```

(β) Συνάρτηση random_names η οποία επιστρέφει η τυχαία ονόματα

```
CREATE OR REPLACE FUNCTION random_names(n integer)
  RETURNS TABLE(name character(30), sex character(1), id integer) AS
$$
BEGIN
  RETURN QUERY
  SELECT nam.name, nam.sex, row_number() OVER ()::integer
  FROM (SELECT "Name".name, "Name".sex
  FROM "Name"
  ORDER BY random() LIMIT n) as nam;
END;
$$
LANGUAGE 'plpgsql' VOLATILE;
```

(γ) Συνάρτηση δημιουργίας αριθμού μητρώου φοιτητή με βάση το έτος εισαγωγής και τον αύξοντα αριθμό του φοιτητή στο έτος

```
CREATE OR REPLACE FUNCTION create_am(year integer, num integer)
  RETURNS character(10) AS
$$
BEGIN
  RETURN concat(year::character(4), lpad(num::text, 6, '0'));
END;
$$
LANGUAGE 'plpgsql' IMMUTABLE;
```

(δ) Συνάρτηση προσαρμογής επωνύμου σε αρσενικό ή θηλυκό γένος

```
CREATE OR REPLACE FUNCTION adapt_surname(surname character(50),
```



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΗΜΜΥ

ΕΡΓΑΣΤΗΡΙΟ ΔΙΑΝΕΜΗΜΕΝΩΝ

ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΕΦΑΡΜΟΓΩΝ

ΙΣΤΟΧΩΡΟΣ ΜΑΘΗΜΑΤΟΣ <https://www.eclass.tuc.gr/courses/HMMY167/>

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ - ΠΛΗ303

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

ΥΠΟΣΤΗΡΙΞΗ ΕΡΓΑΣΤΗΡΙΟΥ: Καζάσης Φώτης, Παππάς Νίκος

```
sex character(1)) RETURNS character(50) AS
$$
DECLARE
    result character(50);
BEGIN
    result = surname;
    IF right(surname,2)<>'ΗΣ' THEN
        RAISE NOTICE 'Cannot handle this surname';
    ELSIF sex='F' THEN
        result = left(surname,-1);
    ELSIF sex<>'M' THEN
        RAISE NOTICE 'Wrong sex parameter';
    END IF;
    RETURN result;
END;
$$
LANGUAGE 'plpgsql' IMMUTABLE;
```