

# 5η Εργασία: Κανονικοποίηση και Ερωτήματα Ομαδοποίησης και Συνάθροισης

**Προθεσμία: θα ανακοινωθεί σύντομα μέσω eclass**

## Σκοπός:

Συνεχίζουμε με μερικές ακόμα αλλαγές κανονικοποίησης και έπειτα τρέχουμε ερωτήματα ομαδοποίησης και συνάθροισης.

## Προαπαιτούμενα:

Θεωρούμε πως η βάση είναι στην κατάσταση που περιγράφεται στην 4η Εργασία.

## Ζητούμενα Εργασίας:

### Μέρος Α

- Παρατηρήστε ότι οι τιμές του πεδίου amenities στον Πίνακα room\_table που δημιουργήσαμε στην προηγούμενη εργασία δεν είναι ατομικές. Προκειμένου να γίνει κανονικοποίηση, δημιουργήστε έναν νέο Πίνακα Amenity με πεδία amenity\_id και amenity\_name. Αποθηκεύστε στο amenity\_name όλες τις μοναδικές ατομικές τιμές του πεδίου amenities αφού επεξεργαστούν κατάλληλα (Μπορείτε να χρησιμοποιήσετε τις String functions της postgres regexp\_split\_to\_table(), regexp\_replace() ή/και όποιες άλλες θεωρείτε ότι σας βολεύουν). Δηλώστε το amenity\_id ως πεδίο αυτόματης προσαύξησης και κάντε το Primary Key.
- Κάντε τις ενέργειες που είναι απαραίτητες ώστε να υπάρχει σύνδεση του Πίνακα room\_table με τον Amenity. Οι ενέργειες αυτές μπορεί να περιλαμβάνουν δημιουργία νέων πεδίων, ορισμό Primary/Foreign Keys μέχρι και δημιουργία νέου Πίνακα.
- Διαγράψτε το πεδίο amenities από τον room\_table.
- Ενημερώστε το ER διάγραμμα της βάσης με τις αλλαγές.
- Ερώτηση: Βρίσκεται η βάση μας σε Κανονική Μορφή BCNF; Δικαιολογείστε για κάθε πίνακα. Εάν όχι, εντοπίστε τις “κακές” Συναρτησιακές Εξαρτήσεις.

Γράψτε όλες τις εντολές διαμόρφωσης της βάσης του Μέρους Α σε ένα αρχείο **part1.sql** και την τελευταία ερώτηση για BCNF σε ένα αρχείο **BCNF.docx** ή **BCNF.pdf**. Για το ER παραδώστε το τελικό αρχείο εικόνας.

### Μέρος Β

Φτιάξτε 5 ερωτήματα συνάθροισης που να έχουν κάποιο νόημα.

- Όλα πρέπει να έχουν τουλάχιστον ένα join
- 2 από τα ερωτήματα θα πρέπει να έχουν τουλάχιστον ένα outer join.

- Τουλάχιστον 2 από τα ερωτήματα θα πρέπει να έχουν ένα WHERE.
- Τουλάχιστον 2 από τα ερωτήματα θα πρέπει να έχουν ένα GROUP BY..
- Τουλάχιστον 2 από τα ερωτήματα θα πρέπει να έχουν ένα HAVING.
- Τουλάχιστον σε 1 από τα ερωτήματα πρέπει να χρησιμοποιηθεί ο πίνακας geolocation.
- Θα πρέπει να χρησιμοποιηθούν όλοι οι πίνακες που φτιάξατε στην προηγούμενη άσκηση (table Host, table Room, table Price, table Location)

Γράψτε όλα τα ερωτήματα σε ένα αρχείο **part2.sql**.

Προσθέστε τις σύντομες περιγραφές των ερωτημάτων και τα πλήθη των αποτελεσμάτων στο ίδιο αρχείο με τη μορφή σχολίων. Π.χ:

```
/* Find all rentings from all houses that took place in the 31st of
December 2017
```

```
Output: 9663 rows
```

```
*/
```

```
SELECT listings.id, listings.listing_url, calendar.date_attr
FROM listings
INNER JOIN calendar
ON listings.id = calendar.listing_id
WHERE calendar.date_attr = '2017-12-31';
```

## Εργαλεία:

- AWS RDS Postgres instance
- Postgres psql ή/και PgAdmin
- [Draw.io](https://draw.io)

## Συμβουλές για την υλοποίηση:

- Προτού αρχίσετε να κάνετε αλλαγές σε κάθε πίνακα θα μπορούσατε να φτιάχνετε ένα αντίγραφο αυτού. Εάν γίνει κάποιο λάθος και σας είναι δύσκολο να επιστρέψετε στην προηγούμενη κατάσταση που ήταν ο πίνακας σας απλώς ξεκινάτε να δουλεύετε με το αντίγραφό του. Σε περίπτωση που φτιάξετε τέτοια αντίγραφα πινάκων, όμως, όταν ολοκληρώσετε την Εργασία παρακαλούμε ενημερώστε μας καταλλήλως για να διευκολυνθούμε στην διόρθωση.
- Επιβεβαιώστε ότι ο χρήστης της βάσης του οποίου μας στέλνετε τα credentials όντως έχει πρόσβαση στη βάση σας και μπορεί να τρέχει `SELECT` queries.
- Τρέξτε και ελέγξτε κάθε ερώτημα στην Airbnb βάση σας.
- Επιβεβαιώστε ότι κανένα ερώτημα δεν είναι άνευ ουσίας όσον αφορά την εξερεύνηση των δεδομένων με την έννοια ότι δεν είναι απλή εμφάνιση κάποιου πίνακα. Το ζητούμενο είναι να υπάρχει συνδυασμός κριτηρίων ώστε να εξάγεται κάποια γνώση. αντι-π.χ: `select * from listings;` ή `select * from listings where id="123";`.
- Επιβεβαιώστε ότι ο χρήστης της βάσης του οποίου μας στέλνετε τα credentials όντως έχει πρόσβαση στη βάση σας.

## Χρήσιμα links:

<https://www.postgresql.org/docs/9.3/static/functions-matching.html>

<http://www.postgresqlonline.com/journal/archives/152-Regular-Expressions-in-PostgreSQL.html>

## Παραδοτέα:

- Δημιουργήστε ένα .txt αρχείο στο οποίο θα αναγράφονται το endpoint του AWS instance σας (μπορείτε να το δείτε στο AWS console, *RDS > Databases > db\_identifier > Connectivity section*), το όνομα της βάσης σας και το username και το password ενός χρήστη με read-only δικαιώματα, ώστε να μπορούμε να δούμε τους πίνακες της βάσης σας. Το .txt αρχείο θα πρέπει να έχει την παρακάτω μορφή:

Endpoint: <name\_of\_the\_endpoint>

Username: <username>

Password: <password>

Database: <name\_of\_the\_database>

- Βάλτε τα αρχεία part1.sql, BCNF.docx/pdf, part2.sql, το αρχείο εικόνας του ER και το αρχείο .txt σε ένα φάκελο. Το όνομα του φακέλου πρέπει να αποτελείται από τους αριθμούς μητρώου σας χωρισμένους με παύλα, δηλαδή *αριθμός\_μητρώου\_1-αριθμός\_μητρώου\_2*. Δημιουργήστε ένα .zip αρχείο αυτού του φακέλου, το οποίο θα έχει το ίδιο όνομα με τον φάκελο.
- Κάντε υποβολή το .zip αρχείο στο eclass στην ενότητα *Εργασίες / 5η Εργασία*.