

# Reaction404

Ευρυβιάδης Λιάπης ΑΕΜ: 03646

Χρήστος Μουντάκης ΑΕΜ: 03801

## Επιλογή Θέματος:

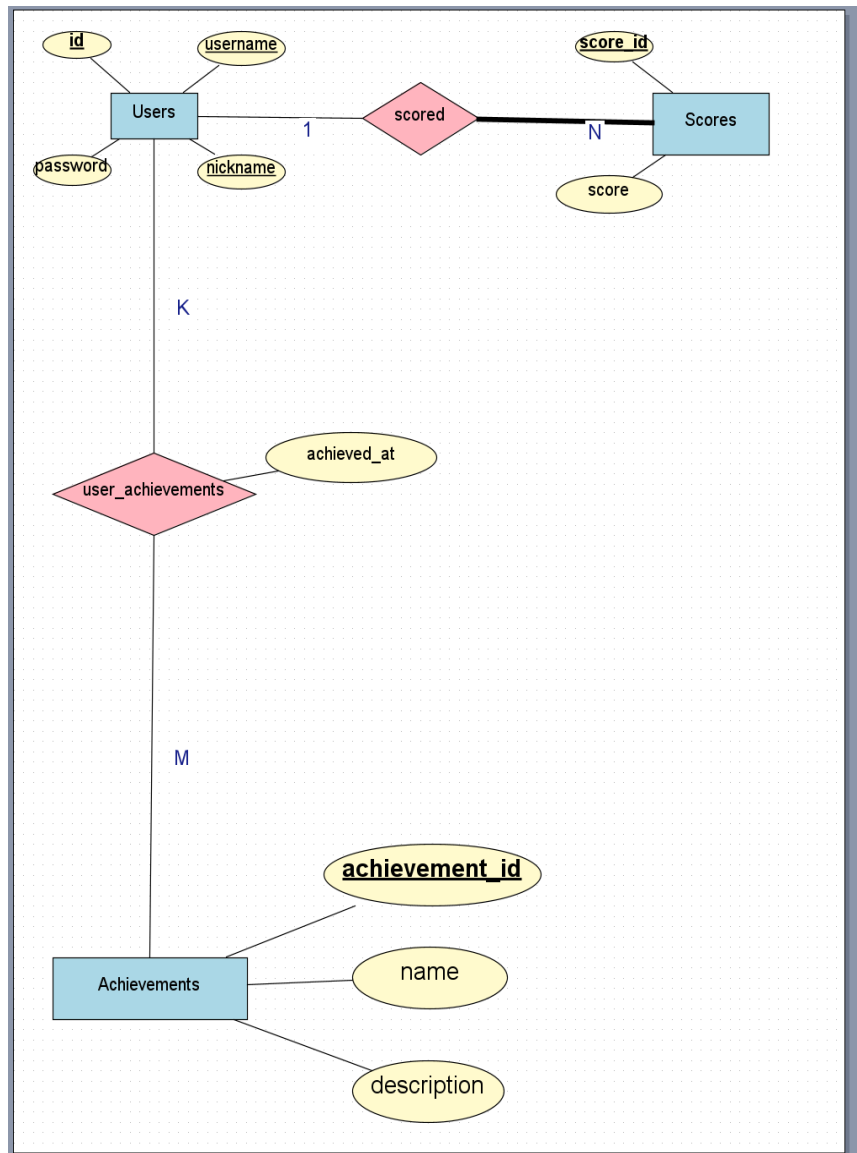
Αποφασίσαμε να αναπτύξουμε ένα παιχνίδι στο οποίο οι χρήστες θα ανταγωνίζονται για το ποιος μπορεί να αντιδράσει πιο γρήγορα στο πάτημα ενός κουμπιού. Το παιχνίδι είναι ανταγωνιστικό, με στόχο την αξιολόγηση της ταχύτητας αντίδρασης των παικτών. Επίσης, περιλαμβάνει συστήματα τίτλων και επιτευγμάτων, προσφέροντας στους χρήστες την ευκαιρία να κερδίσουν αναγνωρίσεις με βάση τις επιδόσεις τους. Τέλος υποστηρίζεται λειτουργία αναζήτησης παικτών και των επιτευγμάτων τους όπως και κατάταξη όλων των βαθμολογιών.

## Περιγραφή:

Κάθε χρήστης της εφαρμογής μας μπορεί να δημιουργήσει τον δικό του λογαριασμό, εισάγοντας ένα **username**, **nickname** και **password**. Το **nickname** θα χρησιμοποιείται για την εμφάνισή του στους άλλους παίκτες, καθώς και για την αλληλεπίδρασή του στο γενικότερο πλαίσιο του παιχνιδιού.

Αφού συνδεθεί στον λογαριασμό του, μπορεί να πατήσει το κουμπί **Play** και να προσπαθήσει να αντιδράσει με τον καλύτερο δυνατό χρόνο, να ξεκλειδώσει τα **achievements** που είναι διαθέσιμα, και να ανέβει όσο πιο ψηλά μπορεί στο **leaderboard**.

## Σχήμα ΔΟΣ:



## Μετατροπή σε ΣΣ:

```
CREATE TABLE users (  
    id SERIAL PRIMARY KEY,  
    username VARCHAR(50) NOT NULL UNIQUE,  
    nickname VARCHAR(50) NOT NULL UNIQUE,  
    password VARCHAR(255) NOT NULL  
);  
  
CREATE TABLE scores (  
    score_id SERIAL PRIMARY KEY,  
    user_id INT NOT NULL,  
    score INT NOT NULL,  
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE  
);  
  
CREATE TABLE achievements (  
    id INT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    description TEXT  
);  
  
CREATE TABLE user_achievements (  
    user_id INT NOT NULL,  
    achievement_id INT NOT NULL,  
    achieved_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY (user_id, achievement_id),  
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,  
    FOREIGN KEY (achievement_id) REFERENCES achievements(id) ON DELETE CASCADE  
);
```

### Users:

**Περιορισμός Πρωτεύοντος κλειδιού (Primary Key):** Το πεδίο `id` είναι το πρωτεύον κλειδί του πίνακα `users`. Είναι αυξανόμενο αυτόματα με τη χρήση του `SERIAL`, διασφαλίζοντας μοναδικότητα για κάθε εγγραφή και ότι κάθε χρήστης θα έχει μοναδική ταυτότητα.

**Περιορισμός Μη Κενών Τιμών (NOT NULL):** Τα πεδία `username`, `nickname` και `password` δεν επιτρέπουν NULL τιμές. Αυτό εξασφαλίζει ότι κάθε χρήστης πρέπει να έχει όνομα χρήστη, ψευδώνυμο και κωδικό πρόσβασης.

**Περιορισμός Μοναδικότητας (UNIQUE):** Τα πεδία `username` και `nickname` έχουν τον περιορισμό `UNIQUE`, πράγμα που διασφαλίζει ότι δεν μπορούν να υπάρχουν δύο

χρήστες με το ίδιο όνομα χρήστη ή ψευδώνυμο. Αυτό εξασφαλίζει ότι κάθε χρήστης έχει μοναδικό όνομα και ψευδώνυμο.

## Scores:

**Περιορισμός Πρωτεύοντος κλειδιού (Primary Key):** Το πεδίο `score_id` είναι το πρωτεύον κλειδί του πίνακα `scores`. Είναι αυξανόμενο αυτόματα με τη χρήση του `SERIAL`, εξασφαλίζοντας μοναδικότητα για κάθε εγγραφή και ότι κάθε βαθμολογία θα έχει μοναδικό ID χωρίς την ανάγκη να το ορίσει ο χρήστης.

**Περιορισμός Μη Κενών Τιμών (NOT NULL):** Τα πεδία `user_id` και `score` δεν επιτρέπουν NULL τιμές. Αυτό εξασφαλίζει ότι κάθε βαθμολογία πρέπει να έχει ένα χρήστη στον οποίο αναφέρεται και μία βαθμολογία, ώστε η βαθμολογία να καταχωρείται σωστά.

**Περιορισμός Ξένου κλειδιού (Foreign Key):** Το πεδίο `user_id` είναι ξένο κλειδί που αναφέρεται στο πεδίο `id` του πίνακα `users`. Αυτό διασφαλίζει ότι κάθε βαθμολογία είναι συσχετισμένη με έναν έγκυρο χρήστη από τον πίνακα `users`. Η εντολή `ON DELETE CASCADE` εξασφαλίζει ότι όταν διαγράφεται ένας χρήστης από τον πίνακα `users`, οι σχετικές εγγραφές στον πίνακα `scores` διαγράφονται αυτόματα. Αυτό αποτρέπει την ύπαρξη βαθμολογιών που δεν σχετίζονται με κάποιον υπάρχοντα χρήστη.

## Achievements

**Περιορισμός Πρωτεύοντος Κλειδιού (Primary Key):** Το πεδίο `id` είναι το πρωτεύον κλειδί του πίνακα `achievements`. Είναι τύπου INT και διασφαλίζει μοναδικότητα για κάθε επίτευγμα στον πίνακα. Κάθε επίτευγμα θα έχει έναν μοναδικό αναγνωριστικό αριθμό για να το διακρίνει από τα υπόλοιπα επιτεύγματα.

**Περιορισμός Μη Κενών Τιμών (NOT NULL):** Το πεδίο `name` δεν επιτρέπει NULL τιμές. Αυτό εξασφαλίζει ότι κάθε επίτευγμα θα έχει ένα όνομα, που είναι υποχρεωτικό για την καταχώρηση του.

## User\_Achievements

**Περιορισμός Πρωτεύοντος Κλειδιού (Primary Key):** Ο πίνακας `user_achievements` έχει σύνθετο πρωτεύον κλειδί που αποτελείται από τα πεδία `user_id` και

**achievement\_id.** Αυτό εξασφαλίζει ότι κάθε συνδυασμός χρήστη και επιτεύγματος θα είναι μοναδικός, αποτρέποντας την καταχώρηση διπλών εγγραφών για τον ίδιο χρήστη και το ίδιο επίτευγμα.

**Περιορισμός Μη Κενών Τιμών (NOT NULL):** Τα πεδία **user\_id** και **achievement\_id** δεν επιτρέπουν NULL τιμές. Αυτό εξασφαλίζει ότι κάθε εγγραφή στον πίνακα θα περιλαμβάνει υποχρεωτικά έναν έγκυρο χρήστη και ένα έγκυρο επίτευγμα.

**Περιορισμός Ξένου Κλειδιού (Foreign Key):** Τα πεδία **user\_id** και **achievement\_id** είναι ξένα κλειδιά που αναφέρονται στα πεδία **id** του πίνακα **users** και **id** του πίνακα **achievements** αντίστοιχα. Αυτό διασφαλίζει ότι κάθε εγγραφή στον πίνακα **user\_achievements** αντιστοιχεί σε έναν έγκυρο χρήστη από τον πίνακα **users** και σε ένα έγκυρο επίτευγμα από τον πίνακα **achievements**. Η εντολή **ON DELETE CASCADE** στα δύο αυτά πεδία, διασφαλίζει ότι αν ένας χρήστης διαγραφεί από τον πίνακα **users** ή ένα επίτευγμα διαγραφεί από τον πίνακα **achievements**, οι σχετικές εγγραφές στον πίνακα **user\_achievements** διαγράφονται αυτόματα.

**Περιορισμός Προεπιλεγμένων Τιμών (Default Value):** Το πεδίο **achieved\_at** έχει προκαθορισμένη τιμή (default value) την τρέχουσα χρονική στιγμή με τη χρήση της εντολής **DEFAULT CURRENT\_TIMESTAMP**. Αυτό διασφαλίζει ότι η ημερομηνία και η ώρα κατά την οποία καταχωρήθηκε το επίτευγμα θα καταγράφονται αυτόματα, χωρίς να απαιτείται ρητή εισαγωγή από τον χρήστη.

---

## Καταγραφή Συναρτησιακών Εξαρτήσεων (FDs):

### Users:

Κλειδί: **id**

Συναρτησιακές εξαρτήσεις:

- **id** → **username**, **nickname**, **password**
- **username** → **id**, **nickname**, **password**
- **nickname** → **id**, **username**, **password**

### Scores:

Κλειδί: `score_id`

Συναρτησιακές εξαρτήσεις:

- `score_id → user_id, score`
- `user_id → score`

## **Achievements:**

Κλειδί: `id`

Συναρτησιακές εξαρτήσεις:

- `id → name, description`
- `name → id, description`

## **User\_Achievements:**

Σύνθετο κλειδί: (`user_id, achievement_id`)

Συναρτησιακές εξαρτήσεις:

- `user_id, achievement_id → achieved_at`
- `user_id → achieved_at`

**Οι πίνακες είναι ήδη κανονικοποιημένοι σε BCNF**

# Περιγραφή Queries:

## 1. SELECT Queries

- `SELECT * FROM users;`  
Επιστρέφει όλες τις εγγραφές από τον πίνακα `users`. Χρησιμοποιείται για την εμφάνιση των δεδομένων των χρηστών.
- `SELECT * FROM achievements;`  
Επιστρέφει όλες τις εγγραφές από τον πίνακα `achievements`. Χρησιμοποιείται για την εμφάνιση όλων των διαθέσιμων επιτευγμάτων.
- `SELECT * FROM users WHERE id = $1;`  
Επιστρέφει τις πληροφορίες ενός χρήστη με βάση το μοναδικό του ID. Χρησιμοποιείται για την ανάκτηση δεδομένων ενός συγκεκριμένου χρήστη.
- `SELECT * FROM users WHERE username = $1;`  
Επιστρέφει τις πληροφορίες ενός χρήστη με βάση το `username`. Χρησιμοποιείται για εύρεση χρήστη με το όνομα χρήστη του.
- `SELECT * FROM users WHERE nickname = $1;`  
Επιστρέφει τις πληροφορίες ενός χρήστη με βάση το `nickname`. Χρησιμοποιείται για εύρεση χρήστη με το ψευδώνυμο του.
- `SELECT u.nickname, MIN(s.score) AS highest_score FROM scores ... LIMIT 10;`  
Επιστρέφει τους 10 καλύτερους χρήστες με βάση το χαμηλότερο (καλύτερο) σκορ. Χρησιμοποιεί `JOIN` για συσχέτιση δεδομένων χρηστών και σκορ.
- `SELECT s.score FROM scores WHERE s.user_id = $1 ORDER BY s.score ASC LIMIT 3;`  
Επιστρέφει τα τρία καλύτερα (χαμηλότερα) σκορ ενός χρήστη, ταξινομημένα σε αύξουσα σειρά.
- `SELECT COUNT(*) AS games_played FROM scores WHERE user_id = $1;`  
Επιστρέφει τον αριθμό των παιχνιδιών που έχει παίξει ένας συγκεκριμένος χρήστης.
- `SELECT s.score FROM scores WHERE s.user_id = $1 ORDER BY s.score ASC LIMIT 1;`  
Επιστρέφει το καλύτερο (χαμηλότερο) σκορ ενός συγκεκριμένου χρήστη.
- `SELECT 1 FROM (SELECT user_id FROM scores ORDER BY score ASC LIMIT 10) ... WHERE user_id = $1;`

Ελέγχει αν ένας συγκεκριμένος χρήστης ανήκει στους 10 καλύτερους χρήστες με βάση το σκορ του.

- `SELECT achievement_id, achieved_at FROM user_achievements WHERE user_id = $1;`

Επιστρέφει όλα τα επιτεύγματα που έχει κερδίσει ένας χρήστης μαζί με την ημερομηνία που τα πέτυχε.

## **2. INSERT Queries**

- `INSERT INTO users (username, nickname, password) VALUES ($1, $2, $3);`

Προσθέτει έναν νέο χρήστη στη βάση δεδομένων με `username`, `nickname`, και `password`.

- `INSERT INTO scores (user_id, score) VALUES ($1, $2);`

Καταγράφει ένα νέο σκορ για έναν χρήστη στη βάση δεδομένων.

- `INSERT INTO user_achievements (user_id, achievement_id) VALUES ($1, $2);`

Καταχωρεί ότι ένας χρήστης πέτυχε ένα συγκεκριμένο επίτευγμα.



## **Οδηγίες χρήσης για την εκκίνηση της εφαρμογής:**

1. Κατεβάστε τον φάκελο **Reaction404** στον υπολογιστή σας.
2. Δημιουργήστε μία νέα βάση δεδομένων στην PostgreSQL.
3. Εκκινήστε την και εισάγετε τα δεδομένα από το αρχείο **dump.sql** για να δημιουργήσετε όλα τα απαραίτητα tables στην βάση σας.
4. Κατευθυνθείτε στον φάκελο **db\_handler.js** ο οποίος βρίσκεται στο path: **Reaction404/Database/db\_handler.js**. Εκεί θα βρείτε όλα τα στοιχεία που πρέπει να συμπληρωθούν για την βάση προκειμένου να συνδεθεί με τον server (Database, Username, password, port, host). Συμπληρώστε τα ανάλογα με την βάση που δημιουργήσατε (θα υπάρχει ένα παράδειγμα ήδη).
5. Ανοίξτε ένα terminal και μεταβείτε στο Reaction404 (π.χ. `cd C:\Users\user\Documents\Reaction404>`).
6. Γράψτε τις εξής εντολές : **npm install, npm run dev**, προκειμένου να ξεκινήσει το API του back-end το οποίο τρέχει στο port: 8080 (Εάν το port 8080 είναι ήδη σε χρήση, παρακαλούμε ελευθερώστε το πριν προχωρήσετε.).
7. Ανοίξτε ένα νέο terminal χωρίς να κλείσετε το προηγούμενο, κατευθυνθείτε στο vue folder path: **Reaction404/vue** και γράψτε τις εξής εντολές : **npm install , npm run serve** (το οποίο παίρνει δυναμικά port).
8. Αυτή η εντολή θα εκκινήσει το front-end στην προεπιλεγμένη θύρα και θα σας εμφανίσει έναν σύνδεσμο στον οποίο μπορείτε να ανοίξετε το παιχνίδι μέσω του browser της επιλογής σας.
9. Καλή διασκέδαση!