

ΠΡΟΧΩΡΗΜΕΝΑ ΘΕΜΑΤΑ
ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΕΦΑΡΜΟΓΩΝ
ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ για
το ακαδημαϊκό έτος 2024-2025

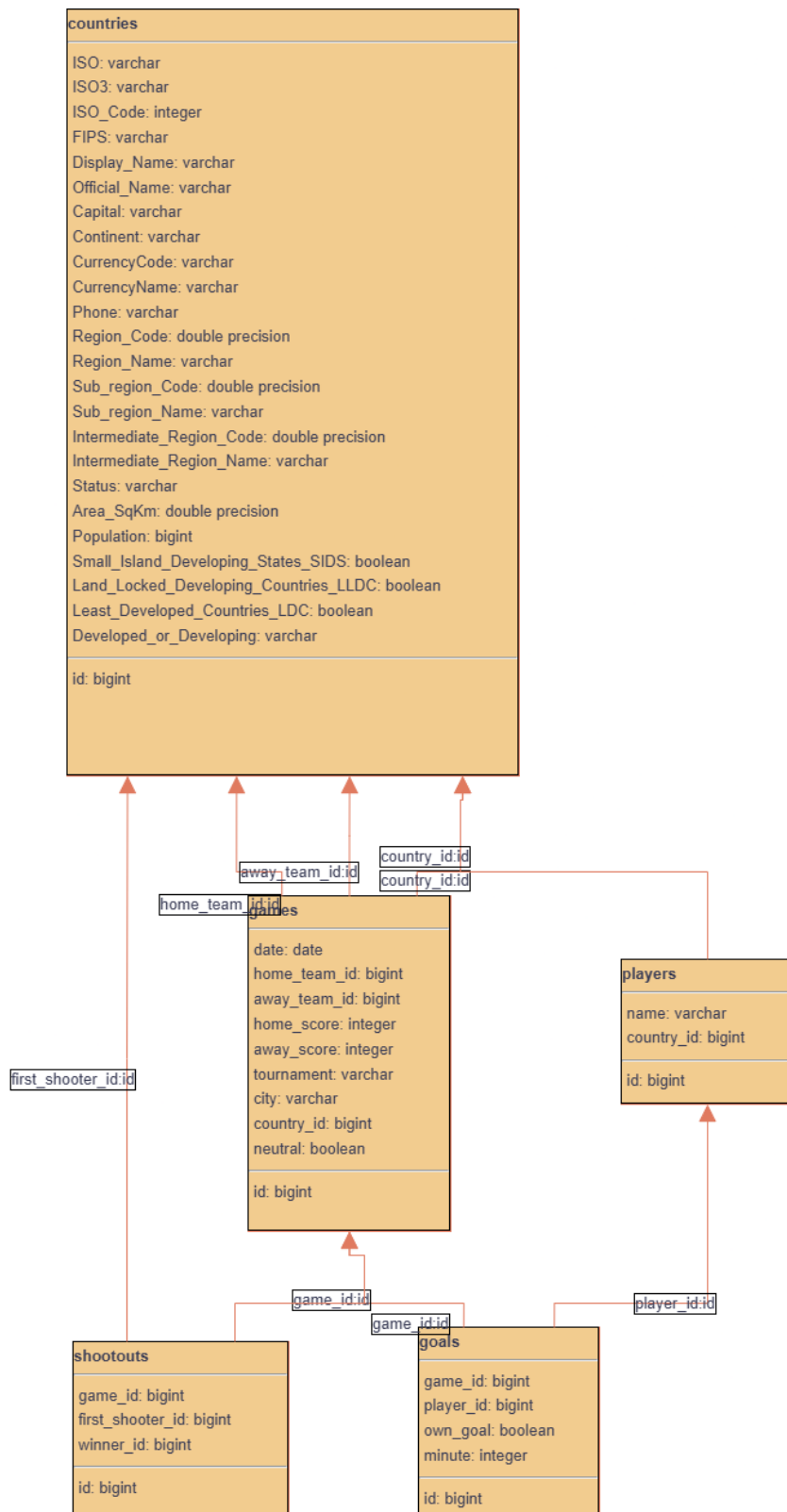
ΑΘΑΝΑΣΟΠΟΥΛΟΣ ΑΝΤΥΡΑΣ
ΦΙΛΙΠΠΟΣ, 5113

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΜΑΪΟΣ 2025

1 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

1.1 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΦΥΣΙΚΟ ΕΠΙΠΕΔΟ



2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΛΟΓΙΣΜΙΚΟΥ

2.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΔΟΜΗ ETL

Η διαδικασία ενσωμάτωσης των δεδομένων εκτελείται από ένα **Python** script με την βοήθεια της βιβλιοθήκης **pandas**. Απαραίτητη προϋπόθεση είναι να έχει δημιουργηθεί το σχήμα της βάσης από τα DDL αρχεία.

Για τις χώρες, σε κάθε πίνακα αντικαθιστούμε former names χωρών με τα current names. Στη συνέχεια βρίσκουμε όλες τις χώρες που υπάρχουν στους υπόλοιπους πίνακες αλλά όχι στον πίνακα countries και τις προσθέτουμε. Τέλος παράγουμε primary key για κάθε χώρα.

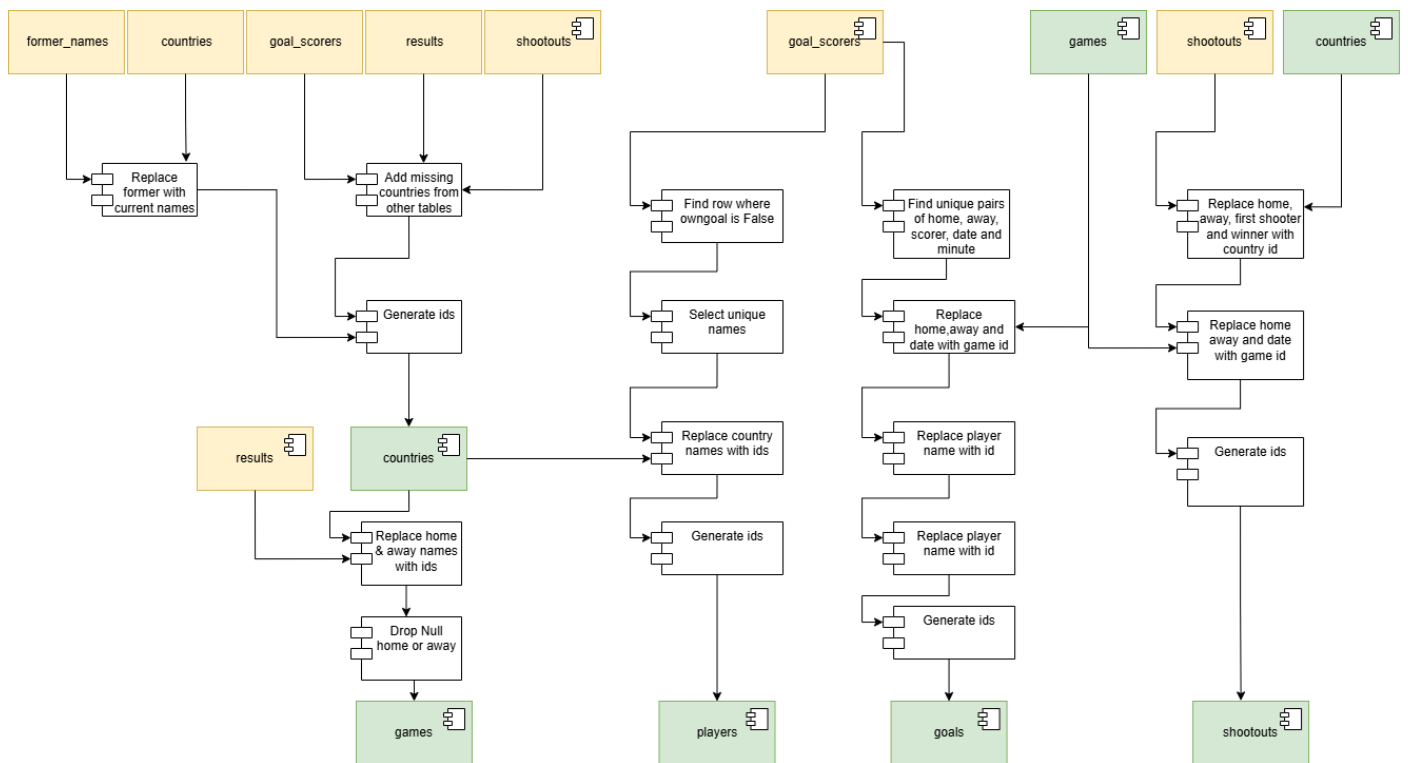
Στους αγώνες αντικαθιστούμε τα ονόματα των χωρών με τα primary key της κάθε χώρας. Αφαιρούμε αγώνους για τους οποίους δεν γνωρίζουμε home ή away χώρα, ή αν home και away είναι ίδια. Τέλος παράγουμε primary key για κάθε αγώνα με βάση home team, away team και date.

Για τους παίκτες, βρίσκουμε γκολ που δεν είναι αυτογκόλ και κρατάμε το όνομα του παίκτη και της ομάδας του. Αντικαθιστούμε το όνομα της χώρας με το id της και παράγουμε id για κάθε παίκτη.

Στον πίνακα με τα γκολ αντικαθιστούμε παίκτες, ομάδες και αγώνες με τα αντίστοιχα id από τον κάθε πίνακα. Παράγουμε τα id.

Στο πίνακα πέναλτι επίσης αντικαθιστούμε ομάδες και ημερομηνία με το κατάλληλο id αγώνα και τα first shooter και winner με τα αντίστοιχα country id. Παράγουμε τα ids.

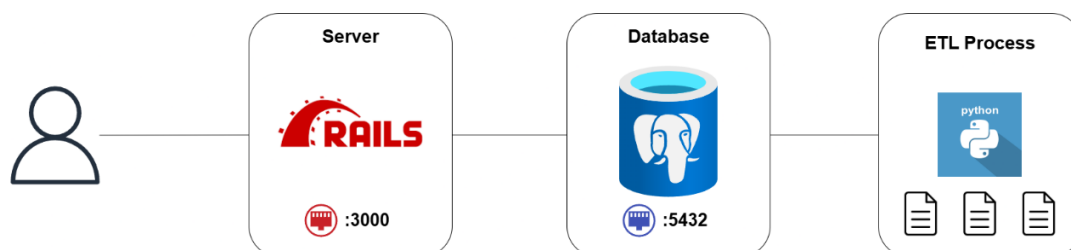
Τα τελικά dataframes φορτώνονται στους αντίστοιχους πίνακες στην βάση μας.



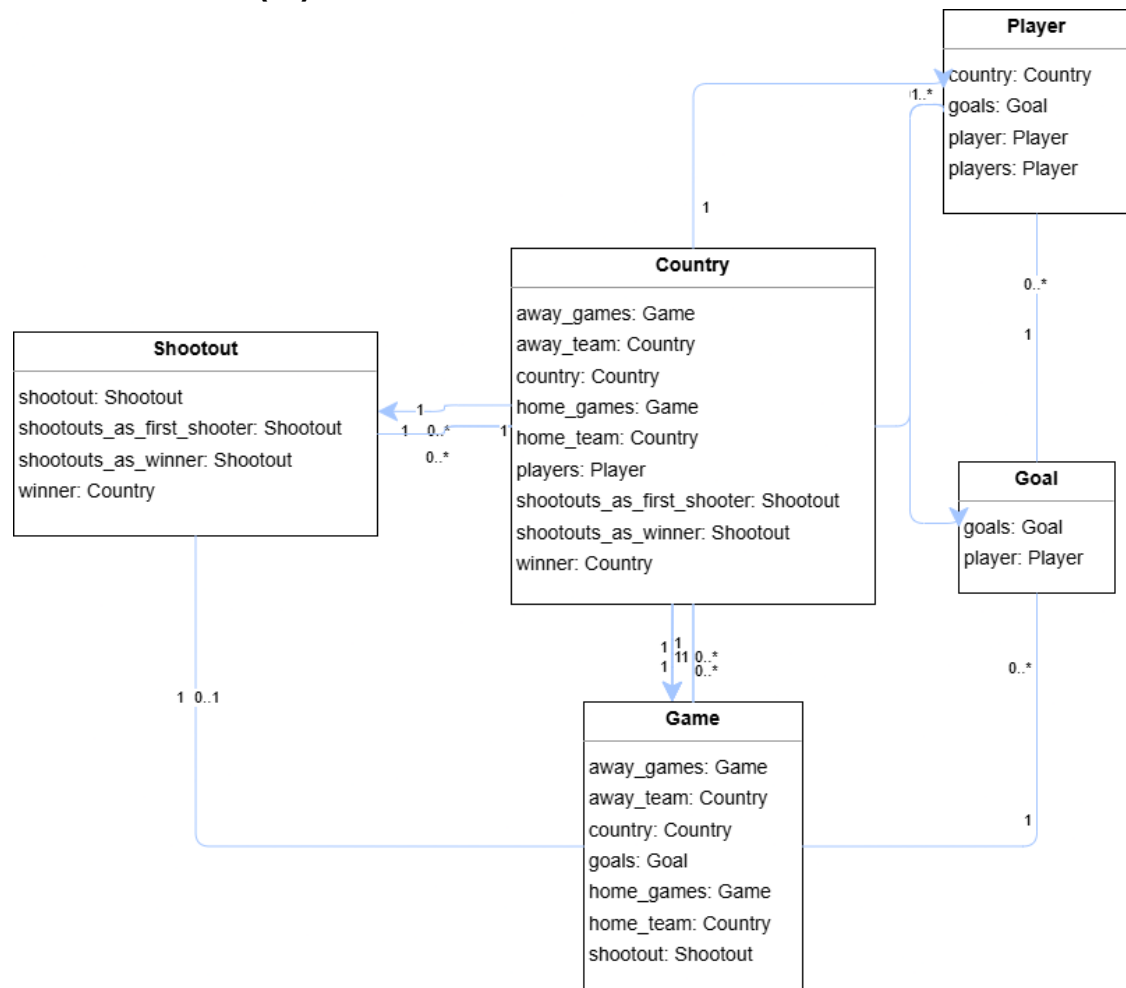
2.2 ΔΙΑΓΡΑΜΜΑΤΑ ΠΑΚΕΤΩΝ / ΥΠΟΣΥΣΤΗΜΑΤΩΝ ΚΕΝΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ

Η εφαρμογή συνολικά αποτελείται από ένα **Ruby on Rails** backend application και μια βάση δεδομένων **PostgreSQL**. Το backend εκτελεί queries στην βάση και δημιουργεί και επιστρέφει τα κατάλληλα views στον χρήστη. Το Rails επιλέχθηκε ως framework λόγω της γρήγορης αυτοματοποιημένης δημιουργίας κλάσεων, views και controllers. Επίσης παρέχει το **Active Record** interface που μας επιτρέπει να γραφούμε κώδικα Ruby που μοιάζει πολύ με **SQL**. Τα views είναι server site rendered με την χρήση embedded Ruby(ERB) προκειμένου να αποφύγουμε την χρήση κάποιου ανεξάρτητου frontend framework.

Για την κατασκευή διαγραμμάτων (bar, line, maps) χρησιμοποιήθηκε η βιβλιοθήκη **D3.js** για το styling της εφαρμογής γενικά η βιβλιοθήκη **simple css**.



2.3 ΔΙΑΓΡΑΜΜΑ(ΤΑ) ΚΛΑΣΕΩΝ ΚΕΝΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ



3 ΥΠΟΔΕΙΓΜΑΤΑ ΕΡΩΤΗΣΕΩΝ ΚΑΙ ΑΠΑΝΤΗΣΕΩΝ

Players:

```
def year_span_of_scoring
  Goal.joins(:game)
    .where(player_id: id)
    .pluck(Arel.sql(
      "CAST(MIN(EXTRACT(YEAR FROM games.date)) AS INT),
        CAST(MAX(EXTRACT(YEAR FROM games.date)) AS INT)"
    ))
    .first
end
```

Figure 1: Εύρος χρονιών που ένας παίκτης σκόραρε

```
def goals_to_games_per_year_stats
  goals.joins("INNER JOIN games ON games.id = goals.game_id")
    .group("EXTRACT(YEAR FROM games.date)")
    .select("EXTRACT(YEAR FROM games.date) AS year,
      COUNT(goals.id) * 1.0 / COUNT(DISTINCT games.id) AS ratio")
end
```

Figure 2: Τέρματα / αγώνες της ομάδας του για το εύρος χρονιών που ο παίκτης αγωνίστηκε

Countries:

```
def wins_per_year_as_home
  home_games.where("home_score > away_score").group("CAST(EXTRACT(year FROM date)AS int)").count
end

def losses_per_year_as_home
  home_games.where("home_score < away_score").group("CAST(EXTRACT(year FROM date)AS int)").count
end

def ties_per_year_as_home
  home_games.where(neutral: true).group("CAST(EXTRACT(year FROM date)AS int)").count
end
```

Figure 3: Συνολικά πόσες νίκες, ήττες, ισοπαλίες, αγώνες είχε η χώρα ως γηπεδούχος

```
def yearly_statistics
  years = (home_games.pluck(:date).map(&:year) + away_games.pluck(:date).map(&:year)).uniq.sort

  wins = Game.where("(home_team_id = :id AND home_score > away_score) OR (away_team_id = :id AND away_score > home_score)", id: id)
    .group("CAST(EXTRACT(YEAR FROM date) AS int)")
    .count

  losses = Game.where("(home_team_id = :id AND home_score < away_score) OR (away_team_id = :id AND away_score < home_score)", id: id)
    .group("CAST(EXTRACT(YEAR FROM date) AS int)")
    .count

  ties = Game.where("home_score = away_score AND (home_team_id = :id OR away_team_id = :id)", id: id)
    .group("CAST(EXTRACT(YEAR FROM date) AS int)")
    .count
end
```

Figure 4: Linechart με νίκες, ήττες, ισοπαλίες, αγώνες ανά χρονιά

```
def show
  start_year = params[:start_year].to_i
  end_year = params[:end_year].to_i

  @games = Game.joins('INNER JOIN countries AS home_countries ON games.home_team_id = home_countries.id')
    .joins('INNER JOIN countries AS away_countries ON games.away_team_id = away_countries.id')
    .where('games.home_team_id = :country_id OR games.away_team_id = :country_id', country_id: @country.id)
    .where('EXTRACT(YEAR FROM games.date) BETWEEN :start_year AND :end_year', start_year: start_year, end_year: end_year)

end
```

Figure 5: Αγώνες χώρας για συγκεκριμένο διάστημα

```
def index
  if params[:query].present?
    @countries = Country.where("Display_Name ILIKE ?", "%#{params[:query]}%")
  else
    @countries = Country.all
  end
end
```

Figure 6: Αναζήτηση χώρας με βάση το όνομα