

ΠΡΟΧΩΡΗΜΕΝΑ ΘΕΜΑΤΑ
ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΕΦΑΡΜΟΓΩΝ
ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ
ΤΟ ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2023

ΟΜΑΔΑ

ΠΑΝΑΓΙΩΤΗΣ ΚΟΛΟΚΟΥΡΗΣ, 4914

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΜΑΪΟΣ 2023

ΙΣΤΟΡΙΚΟ ΠΡΟΗΓΟΥΜΕΝΩΝ ΕΚΔΟΣΕΩΝ

Ημερομηνία	Έκδοση	Περιγραφή	Συγγραφέας
2023/05/18	0.1	Εκκίνηση	ΠΚ
2023/05/26	0,2	Τελική Αναφορά	ΠΚ

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1	Βαση δεδομενων	4
1.1	Σχεσιακό σχήμα σε λογικό επίπεδο	4
1.2	Σχεσιακό σχήμα σε φυσικό επίπεδο.....	5
1.2.1	Ρύθμιση των παραμέτρων του dbms.....	5
1.2.2	Ρύθμιση του φυσικού σχήματος της βάσης δεδομένων	6
1.2.3	Ρύθμιση ασφάλειας	6
2	Αρχιτεκτονική Λογισμικού.....	7
2.1	Extract Transform and Load Data (ETL)	7
2.1.1	Μετασχηματισμοί Ακατέργαστων Δεδομένων	8
2.1.2	Αποκλεισμός Δεδομένων	11
2.1.3	Αρχεία Φόρτωσης Δεδομένων στην Βάση Δεδομένων	11
2.1.4	Εξαγωγή Backup της Βάσης.....	12
2.2	API Βάσης Δεδομένων	13
2.3	Web Εφαρμογή (Front End)	14
2.3.1	Διάγραμμα Πακέτων Angular	14
3	Υποδείγματα ερωτήσεων και απαντήσεων.....	16
3.1	Επιλογή Στατιστικού Δείκτη	16
3.2	Line Charts	16
3.3	Bar Charts	17
3.4	Scatter Plot.....	17
4	Μελέτη Δεδομένων.....	18
4.1	Κατά Κεφαλήν Εισόδημα	18
4.2	Brain Drain.....	18
4.3	Ανάπτυξη και Πληθυσμός.....	19
5	Λοιπά σχόλια.....	21
6	Παράρτημα Α.....	22
7	Παράρτημα Β.....	33

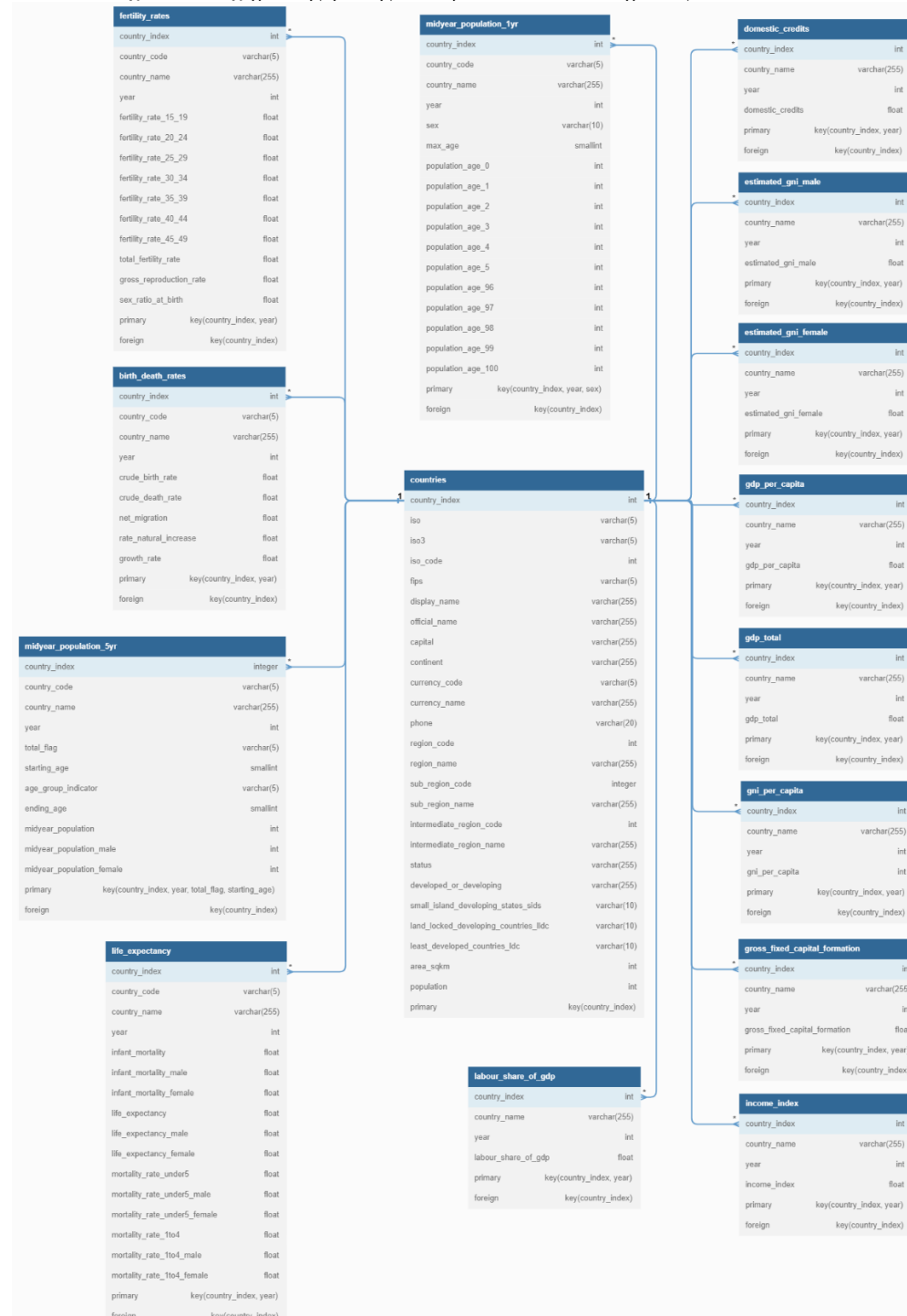
1 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Στην παρούσα ενότητα περιγράφεται το σχήμα της βάσης δεδομένων που χρησιμοποιείται στην εργασία.

1.1 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΛΟΓΙΚΟ ΕΠΙΠΕΔΟ

Στην Εικόνα 1 παρουσιάζεται το σχεσιακό σχήμα της βάσης δεδομένων.

Εικόνα 1: Σχεσιακό σχήμα της βάσης δεδομένων του συστήματος



Σημειώνεται ότι ο πίνακας `midyear_population_1yr` στην Εικόνα 1 παρουσιάζεται περικομμένος για να χωρέσει ολόκληρο το σχήμα στην σελίδα. Στην πραγματικότητα υπάρχουν στήλες για κάθε ηλικία από 0 έως 100.

Τα αρχεία κώδικα SQL που χρησιμοποιήθηκαν για να δημιουργηθεί το σχήμα της βάσης καθώς και να φορτωθούν τα αρχεία δεδομένων παρατίθεται στο

Παράρτημα Α.

1.2 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΦΥΣΙΚΟ ΕΠΙΠΕΔΟ

Στην παρούσα ενότητα καταγράφονται οι ρυθμίσεις σε φυσικό επίπεδο.

1.2.1 ΡΥΘΜΙΣΗ ΤΩΝ ΠΑΡΑΜΕΤΡΩΝ ΤΟΥ DBMS

Το σύστημα διαχείρισης βάσεων δεδομένων (DBMS) που χρησιμοποιήθηκε είναι η MySQL.

Η storage engine της MySQL είναι (από προεπιλογή) η InnoDB η οποία και χρησιμοποιήθηκε στην παρούσα εργασία.

Η ανάθεση μνήμης στον buffer pool της InnoDB προσδιορίζεται με τον συνδυασμό των μεταβλητών

- *innodb_buffer_pool_size*,
- *innodb_buffer_pool_chunk_size* και
- *innodb_buffer_pool_instances*.

Η σχέση που συνδέει τις τρεις μεταβλητές είναι: η τιμή της *innodb_buffer_pool_size* πρέπει να είναι πολλαπλάσιο του γινομένου *innodb_buffer_pool_chunk_size* * *innodb_buffer_pool_instances*.

Οι τιμές των παραπάνω μεταβλητών που έχουν τεθεί στην MySQL είναι οι εξής:

- *innodb_buffer_pool_size* = 8G
- *innodb_buffer_pool_chunk_size* = 128MB
- *innodb_buffer_pool_instances* = 8

Αυτές οι τιμές είχαν επιλεγεί σαν μια καλή συμβιβαστική λύση για την γενική χρήση της MySQL και σύμφωνα με την συνολική διαθέσιμη μνήμη του υπολογιστή (16GB).

Οι παραπάνω τιμές είναι υπεραρκετές για τις απαιτήσεις της δεδομένης βάσης. Η βάση έχει μέγεθος στον δίσκο περίπου 75 MB. Σημειώνεται ότι δεν χρησιμοποιήθηκε το μεγάλο αρχείο δημογραφικών δεδομένων για λόγους που εξηγούνται στην παράγραφο 0.

1.2.2 ΡΥΘΜΙΣΗ ΤΟΥ ΦΥΣΙΚΟΥ ΣΧΗΜΑΤΟΣ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Δεν χρησιμοποιήθηκαν επιπλέον ευρετήρια εκτός από το πρωτεύον κλειδί κάθε πίνακα. Τα δεδομένα των πινάκων είναι σχετικά μικρά και οι ερωτήσεις στην βάση, στα πλαίσια της εργασίας, πολύ γρήγορες ($<1s$). Συνεπώς δεν προστέθηκαν ευρετήρια σε έξτρα παραμέτρους των πινάκων που χρησιμοποιούνται για αναζήτηση ή ταξινόμηση πχ. Όνομα, χώρα ή χρονολογία.

Τέλος, δεν χρησιμοποιήθηκαν καθόλου όψεις.

1.2.3 ΡΥΘΜΙΣΗ ΑΣΦΑΛΕΙΑΣ

Δεν χρησιμοποιήθηκαν κάποιες ρυθμίσεις ασφαλείας.

2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΛΟΓΙΣΜΙΚΟΥ

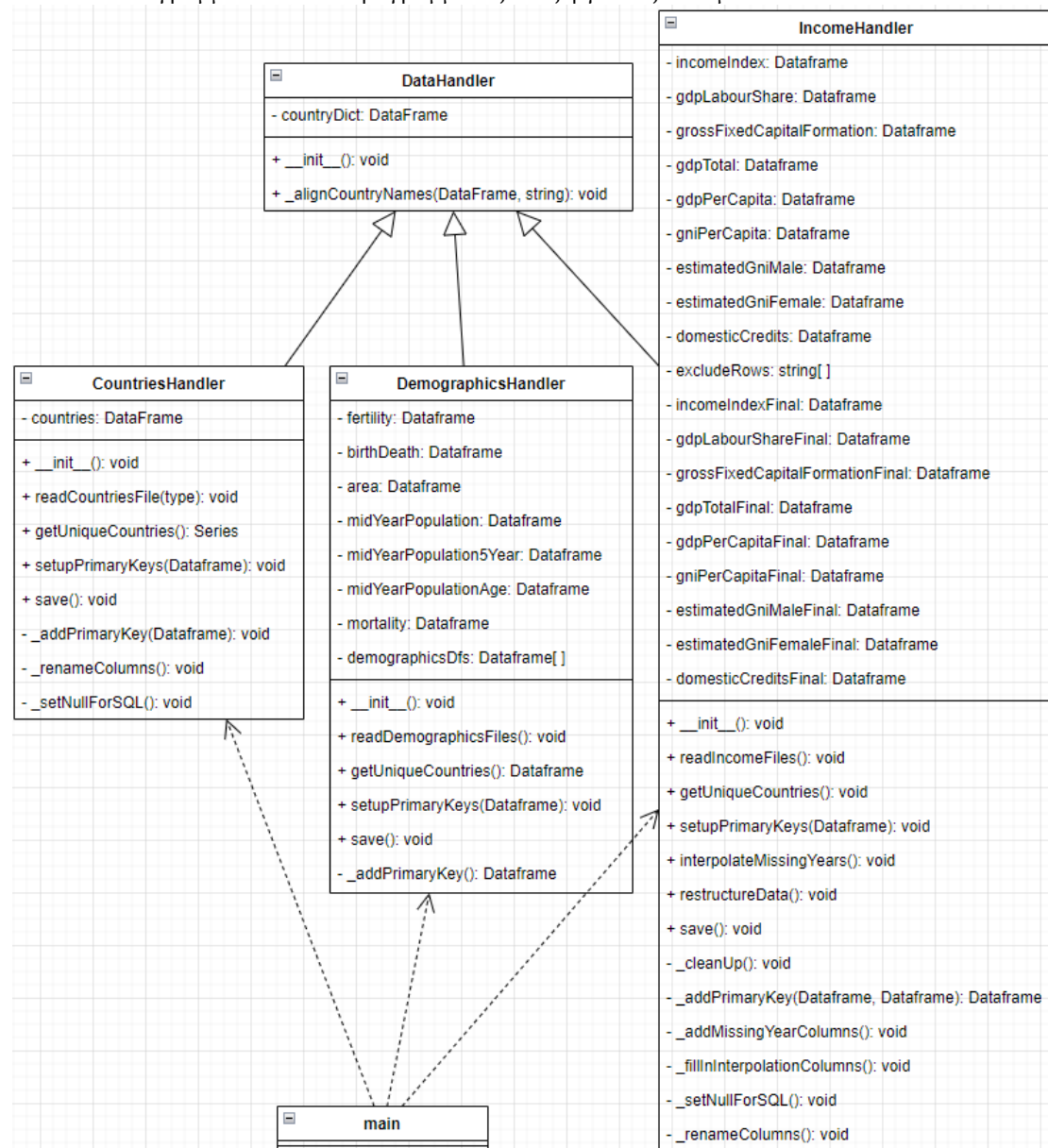
2.1 EXTRACT TRANSFORM AND LOAD DATA (ETL)

Τα δεδομένα πάνω στα οποία βασίστηκε η παρούσα εργασία προέρχονται από την ιστοσελίδα Kaggle (www.kaggle.com).

Τα ακατέργαστα αυτά δεδομένα έπρεπε να μετασχηματιστούν σε κατάλληλη μορφή για να φορτωθούν σε επερωτήσιμους πίνακες σε μια βάση δεδομένων. Για τον μετασχηματισμό αυτό αναπτύχθηκε αντικειμενοστραφές πρόγραμμα Python με χρήση της βιβλιοθήκης pandas.

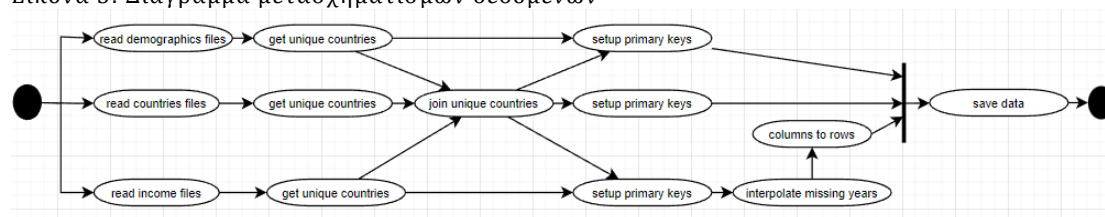
Το διάγραμμα κλάσεων του προγράμματος παρουσιάζεται στην Εικόνα 2.

Εικόνα 2: Διάγραμμα κλάσεων προγράμματος επεξεργασίας δεδομένων



Ένα υψηλού επιπέδου διάγραμμα των μετασχηματισμών που υπέστησαν τα δεδομένα παρουσιάζεται στην Εικόνα 3. Μια πιο αναλυτική περιγραφή των μετασχηματισμών παρουσιάζεται στην ενότητα 2.1.1.

Εικόνα 3: Διάγραμμα μετασχηματισμών δεδομένων



2.1.1 ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΙ ΑΚΑΤΕΡΓΑΣΤΩΝ ΔΕΔΟΜΕΝΩΝ

Τα ακατέργαστα δεδομένα προέρχονται από τρεις πηγές και αφορούν τις ανάλογες τρεις κατηγορίες:

- Στοιχεία κρατών
- Δημογραφικά στοιχεία κρατών
- Οικονομικά στοιχεία κρατών

Παρακάτω περιγράφονται διάφορα ζητήματα που αφορούν την προετοιμασία των δεδομένων και αιτιολογούνται οι διάφορες αποφάσεις που πάρθηκαν.

2.1.1.1 ΠΡΩΤΕΥΟΝΤΑ ΚΛΕΙΔΙΑ

Όλα τα διαθέσιμα δεδομένα αφορούν χώρες του κόσμου.

Στις διαφορετικές κατηγορίες δεδομένων υπάρχουν διαθέσιμες διαφορετικές χώρες και σε κάποιες περιπτώσεις διαφοροποιείται εν μέρει ακόμα και το όνομα κάποιων χωρών.

Ταυτόχρονα δημιουργείται η ανάγκη να υπάρχει αντιστοίχιση μεταξύ των χωρών σε κάθε κατηγορία δεδομένων.

Από τους παραπάνω λόγους προκύπτει η ανάγκη δημιουργίας ενός μοναδικού αναγνωριστικού για κάθε χώρα, δηλαδή ενός πρωτεύοντος κλειδιού. Η καλύτερη, από άποψη επιδόσεων, επιλογή πρωτεύοντος κλειδιού για τους πίνακες της βάσης δεδομένων είναι ένας αύξων ακέραιος αριθμός, μοναδικός για κάθε χώρα. Για την απόδοση, στις χώρες, ενός τέτοιου πρωτεύοντος κλειδιού πραγματοποιήθηκαν οι εξής ενέργειες.

- Διαβάστηκαν όλα τα δεδομένα και των τριών κατηγοριών και αποθηκεύτηκαν σε κατάλληλες δομές στην μνήμη (Pandas Dataframes).
- Σαρώθηκαν όλα τα δεδομένα και αποθηκεύτηκαν τα μοναδικά ονόματα χωρών που βρέθηκαν σε κάθε κατηγορία δεδομένων.
- Τα ονόματα κάποιων χωρών διαφοροποιούνται ανάμεσα στις διαφορετικές κατηγορίες δεδομένων. Έτσι αναγνωρίστηκαν οι εν λόγω περιπτώσεις και δημιουργήθηκε ένα λεξικό αντιστοίχισης μεταξύ των ονομάτων το οποίο εφαρμόζεται αυτόματα σε όλα τα ονόματα των χωρών κάθε κατηγορίας δεδομένων. Αυτό το λεξικό βρίσκεται στην κλάση *DataHandler*.
- Δημιουργήθηκε μια λίστα με τα συνολικά μοναδικά και εναρμονισμένα ονόματα χωρών. Η λίστα ταξινομήθηκε και σε κάθε χώρα αποδόθηκε ένας μοναδικός αύξων αριθμός, το πρωτεύον κλειδί.

- Με χρήση της παραπάνω λίστας, σαρώθηκαν όλα τα δεδομένα όλων των κατηγοριών και δίπλα σε κάθε όνομα χώρας προσαρτήθηκε ο μοναδικός ακέραιος αριθμός, ήτοι το πρωτεύον κλειδί.

2.1.1.2 ΧΡΟΝΙΚΗ ΔΙΑΘΕΣΙΜΟΤΗΤΑ ΟΙΚΟΝΟΜΙΚΩΝ ΔΕΔΟΜΕΝΩΝ

Ενώ τα δημογραφικά δεδομένα είναι διαθέσιμα ανά έτος, δεν ισχύει το ίδιο για τα οικονομικά στοιχεία. Σε κάποιες περιπτώσεις διατίθενται σε χρονικά διαστήματα πενταετίας.

Ο μεγαλύτερος όγκος δεδομένων διατίθεται ανά έτος. Για τις περιπτώσεις δεδομένων ανά πενταετία αποφασίστηκε να χρησιμοποιηθεί η τεχνική της γραμμικής παρεμβολής (interpolation) για την συμπλήρωση των χρόνων που λείπουν. Είναι φανερό ότι τα δεδομένα που θα προκύψουν θα είναι μόνο μια εκτίμηση των πραγματικών τιμών. Παρόλα αυτά αυτή η εκτίμηση δεν μπορεί να απέχει πολύ από την πραγματικότητα.

Η εναλλακτική λύση της περικοπής όλων των δημογραφικών δεδομένων ώστε να εναρμονιστούν ανά πενταετία εξετάστηκε και απορρίφθηκε με την αιτιολογία ότι έτσι θυσιάζονται πολλά δεδομένα.

2.1.1.3 ΟΡΓΑΝΩΣΗ ΤΩΝ ΟΙΚΟΝΟΜΙΚΩΝ ΔΕΔΟΜΕΝΩΝ

Τα δημογραφικά δεδομένα παρέχονται με οργάνωση διαφορετικών γραμμών για χώρα και έτος πχ. υπάρχει μια γραμμή για την Ελλάδα του 2020 και ξεχωριστή γραμμή για την Ελλάδα του 2021.

Τα οικονομικά δεδομένα παρέχονται με οργάνωση μια γραμμή για κάθε χώρα και πολλές στήλες για κάθε χρονιά.

Γίνεται φανερό ότι οι δυο κατηγορίες θα πρέπει να εναρμονισθούν ως προς την αναπαράσταση των δεδομένων ώστε να προκύψει όσο τον δυνατόν πιο εναρμονισμένη βάση δεδομένων. Επιλέχθηκε να υιοθετηθεί η υπάρχουσα οργάνωση των δημογραφικών δεδομένων. Το επιχείρημα υπέρ των πολλών γραμμών για χώρα και χρονιά είναι ότι διευκολύνεται η εισαγωγή/διαγραφή μιας συγκεκριμένης χρονιάς αν αυτό απαιτηθεί. Για παράδειγμα, για την εισαγωγή ενός έτους δεν χρειάζεται να υπάρχουν/δοθούν τιμές για όλα τα υπόλοιπα έτη/στήλες.

Επίσης επιλέχθηκε να δημιουργηθούν πολλοί (μικροί) πίνακες για κάθε οικονομικό δεδομένο σε αντίθεση με έναν μεγάλο πίνακα με πολλές στήλες. Το επιχείρημα είναι ανάλογο του παραπάνω, ήτοι διευκολύνεται η εισαγωγή/διαγραφή δεδομένων. Το αντεπιχείρημα είναι ότι έτσι απαιτούνται joins για την ενδεχόμενη συνένωση ξανά των δεδομένων αλλά στο παρόν project αυτό δεν απαιτείται.

2.1.1.4 ΟΡΓΑΝΩΣΗ ΦΑΚΕΛΩΝ ΔΕΔΟΜΕΝΩΝ

Τα ακατέργαστα δεδομένα που χρησιμοποιήθηκαν οργανώθηκαν σε φακέλους ως εξής:

- *Data*
 - *original*
 - *countries*
 - *income*
 - *international*

Τα επεξεργασμένα δεδομένα οργανώθηκαν σε φακέλους αντίστοιχα:

- *Data*
 - *processed*
 - *countries*
 - *income*
 - *international*

2.1.1.5 ΕΚΤΕΛΕΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ ΜΕΤΑΣΧΗΜΑΤΙΣΜΩΝ

Η εκτέλεση του προγράμματος Python (version 3.7.9) από το τερματικό απαιτεί την εγκατάσταση των πακέτων *pandas* και *openpyxl* η οποία πραγματοποιείται με τις εξής εντολές (τερματικό στην τοποθεσία του προγράμματος Python ήτοι *src/ETL/*)

- *pip3 install pandas*
- *pip3 install openpyxl*

Η εκτέλεση του προγράμματος πραγματοποιείται από το τερματικό (στην τοποθεσία του προγράμματος Python ήτοι *src/ETL/*) με την εντολή:

python main.py

Εναλλακτικά, η εκτέλεση μπορεί να πραγματοποιηθεί με την χρήση κάποιου IDE όπως ο Spyder που χρησιμοποιήθηκε για την ανάπτυξη του κώδικα.

2.1.1.6 ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕΤΑΣΧΗΜΑΤΙΣΜΩΝ

Ο μετασχηματισμός των δοθέντων ακατέργαστων δεδομένων γίνεται αυτόματα με την εκτέλεση του προγράμματος Python. Δεν απαιτούνται manual βήματα. Παράγονται τα εξής αρχεία ανά κατηγορία:

- Στοιχεία κρατών:
 - *countriesFinal.csv*
- Δημογραφικά στοιχεία κρατών:
 - *age_specific_fertility_rates_final.csv*
 - *birth_death_growth_rates_final.csv*
 - *country_names_area_final.csv*
 - *midyear_population_5yr_age_sex_final.csv*
 - *midyear_population_age_sex_final.csv*
 - *midyear_population_final.csv*
 - *mortality_life_expectancy_final.csv*
- Οικονομικά στοιχεία κρατών:
 - *domestic_credits_final.csv*
 - *estimated_gni_female_final.csv*
 - *estimated_gni_male_final.csv*
 - *gdp_per_capita_final.csv*
 - *gdp_total_final.csv*
 - *gni_per_capita_final.csv*
 - *gross_fixed_capital_formation_final.csv*
 - *income_index_final.csv*
 - *labour_share_of_gdp_final.csv*

2.1.2 ΑΠΟΚΛΕΙΣΜΟΣ ΔΕΔΟΜΕΝΩΝ

Πρέπει να σημειωθεί ότι στην εφαρμογή που ακολουθεί επιλέχθηκε να μην χρησιμοποιηθούν τα εξής σύνολα δεδομένων:

- Στοιχεία έκτασης χωρών:
Αρχικά η ανάπτυξη της εφαρμογής εστιάσθηκε στην απεικόνιση της μεταβολής των δεδομένων στον χρόνο και όταν αυτή ολοκληρώθηκε ικανοποιητικά δεν υπήρχε χρόνος για την ενσωμάτωση των στοιχείων έκτασης.
- Πληθυσμιακά δεδομένα (midyear_population.csv)
Το εν λόγω αρχείο είναι περιττό. Περιέχει συνολικά δεδομένα πληθυσμού, αθροισμένα και για τα δύο φύλα. Όλη αυτή η πληροφορία περιέχεται στο αρχείο midyear_population_5yr_age_sex.csv το οποίο έχει ενσωματωθεί στην εφαρμογή.
- Αρχείο δεδομένων πληθυσμού (midyear_population_age_country_code.csv):
Ανάμεσα στα παρεχόμενα δεδομένα βρίσκεται και ένα μεγάλο αρχείο πληθυσμιακών δεδομένων ανά ηλικιακό έτος. Αναγνωρίζεται ότι η ενσωμάτωση αυτού του αρχείου στην βάση θα ήταν καλή άσκηση για την αντιμετώπιση των αντίστοιχων χρονοβόρων ερωτήσεων. Παρόλα αυτά παρατηρήθηκε ότι αυτό το αρχείο δεν παρέχει παραπάνω πληροφορία από το κατά πολύ μικρότερο αρχείο midyear_population_age_sex.csv. Το πρώτο (μεγάλο) παρέχει διπλότυπα δεδομένα για όλα τα ηλικιακά έτη που διαφοροποιούνται μόνο ως προς τις στήλες age και population για τις οποίες δεν υπάρχει πληροφόρηση ως προς το τι περιγράφουν συνεπώς και δεν μπορούν να χρησιμοποιηθούν.

2.1.3 ΑΡΧΕΙΑ ΦΟΡΤΩΣΗΣ ΔΕΔΟΜΕΝΩΝ ΣΤΗΝ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Δημιουργήθηκαν sql αρχεία φόρτωσης των επεξεργασμένων δεδομένων στην βάση δεδομένων. Αυτά παρατίθενται στο Παράρτημα Α.

Τα αρχεία φόρτωσης δεδομένων στην βάση βρίσκονται στην εξής τοποθεσία:

- *ETL*
 - *db_load*

Η βάση δεδομένων ονομάζεται *mye030_countries_stats* και περιέχει τους εξής πίνακες:

- *birth_death_rates*
- *countries*
- *country_areas*
- *domestic_credits*
- *estimated_gni_female*
- *estimated_gni_male*
- *fertility_rates*
- *gdp_per_capita*
- *gdp_total*
- *gni_per_capita*
- *gross_fixed_capital_formation*
- *income_index*
- *labour_share_of_gdp*
- *life_expectancy*

- *midyear_population*
- *midyear_population_1yr*
- *midyear_population_5yr*

Το σχήμα της βάσης παρουσιάζεται στην Εικόνα 1.

Σημειώνεται ο πίνακας *countries* έχει primary key ένα μοναδικό ακέραιο αριθμό και όλοι οι υπόλοιποι πίνακες έχουν foreign keys προς αυτό.

2.1.4 ΕΞΑΓΩΓΗ BACKUP ΤΗΣ ΒΑΣΗΣ

Δημιουργήθηκε backup της βάσης δεδομένων σε αρχείο *.sql*. Η δημιουργία του έγινε αυτόματα από το MySQL Workbench.

Η εκτέλεση του αρχείου αναδημιουργεί την βάση σε περίπτωση καταστροφής ή απώλειας της. Το αρχείο αυτό βρίσκεται στην παρακάτω τοποθεσία και είναι πολύ μεγάλο για να παρουσιαστεί στην αναφορά:

- *ETL*
 - *db_backup*

2.2 API ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Αναπτύχθηκε πρόγραμμα – API σε Node JS για την επικοινωνία με την βάση δεδομένων. Το πρόγραμμα τρέχει έναν (τοπικό) server που ακούει και απαντάει σε http requests.

Η τεκμηρίωση του API παρατίθεται στο Παράρτημα Β και έχει αναρτηθεί επίσης στην ιστοσελίδα: <https://documenter.getpostman.com/view/25718180/2s93kxd6vE>.

Το API της βάσης δεδομένων έχει αναπτυχθεί ώστε να απαντάει σε τριών ειδών ερωτήσεις:

- Ερώτηση για όλες τις διαθέσιμες χώρες στην βάση.
- Ερωτήσεις για τα διαθέσιμα έτη ανάλογα με την χώρα και το στατιστικό μέγεθος.
- Ερωτήσεις για τα στατιστικά μεγέθη ανάλογα με την χώρα, το χρονικό διάστημα και ενδεχομένως φύλο, ηλικία/ηλικιακό γκρουπ.

Το πρόγραμμα χρησιμοποιεί την βιβλιοθήκη express για την διαχείριση των εισερχόμενων διαδρομών (routes).

Τα εισερχόμενα requests δρομολογούνται στα κατάλληλα controller modules:

- *countriesControllers*
- *yearsControllers*
- *statisticsControllers*

μέσω των αντίστοιχων router modules:

- *countriesRoutes*
- *yearsRoutes*
- *statisticsRoutes*

2.3 WEB ΕΦΑΡΜΟΓΗ (FRONT END)

Η διεπαφή χρήστη αναπτύχθηκε ως web εφαρμογή με χρήση Angular.

Η εφαρμογή δίνει την δυνατότητα στον χρήστη να δημιουργήσει τριών ειδών γραφήματα:

- Line charts
- Bar charts
- Scatter plots

Η συμπεριφορά των line charts και bar charts παρουσιάζει αρκετά κοινά σημεία.

- Ο χρήστης έχει την δυνατότητα να απεικονίσει ένα ή περισσότερα στατιστικά μεγέθη (δημογραφικά/οικονομικά).
- Και στα δύο είδη γραφημάτων, παρουσιάζεται η εξέλιξη των στατιστικών μεγεθών στον χρόνο.
- Η μόνη διαφοροποίηση των δύο είναι ότι τα bar charts περιορίζονται στην απεικόνιση πέντε (5) στατιστικών μεγεθών. Ένας μεγαλύτερος αριθμός θα έκανε το γράφημα δύσκολο στην ανάγνωση και κατανόηση.

Η απεικόνιση των scatter plots, εξ' ορισμού συγκρίνει την συσχέτιση δύο μόνο στατιστικών μεγεθών.

Η επιλογή στατιστικού μεγέθους είναι κοινή σε όλες τις περιπτώσεις και περιλαμβάνει τα εξής σημεία:

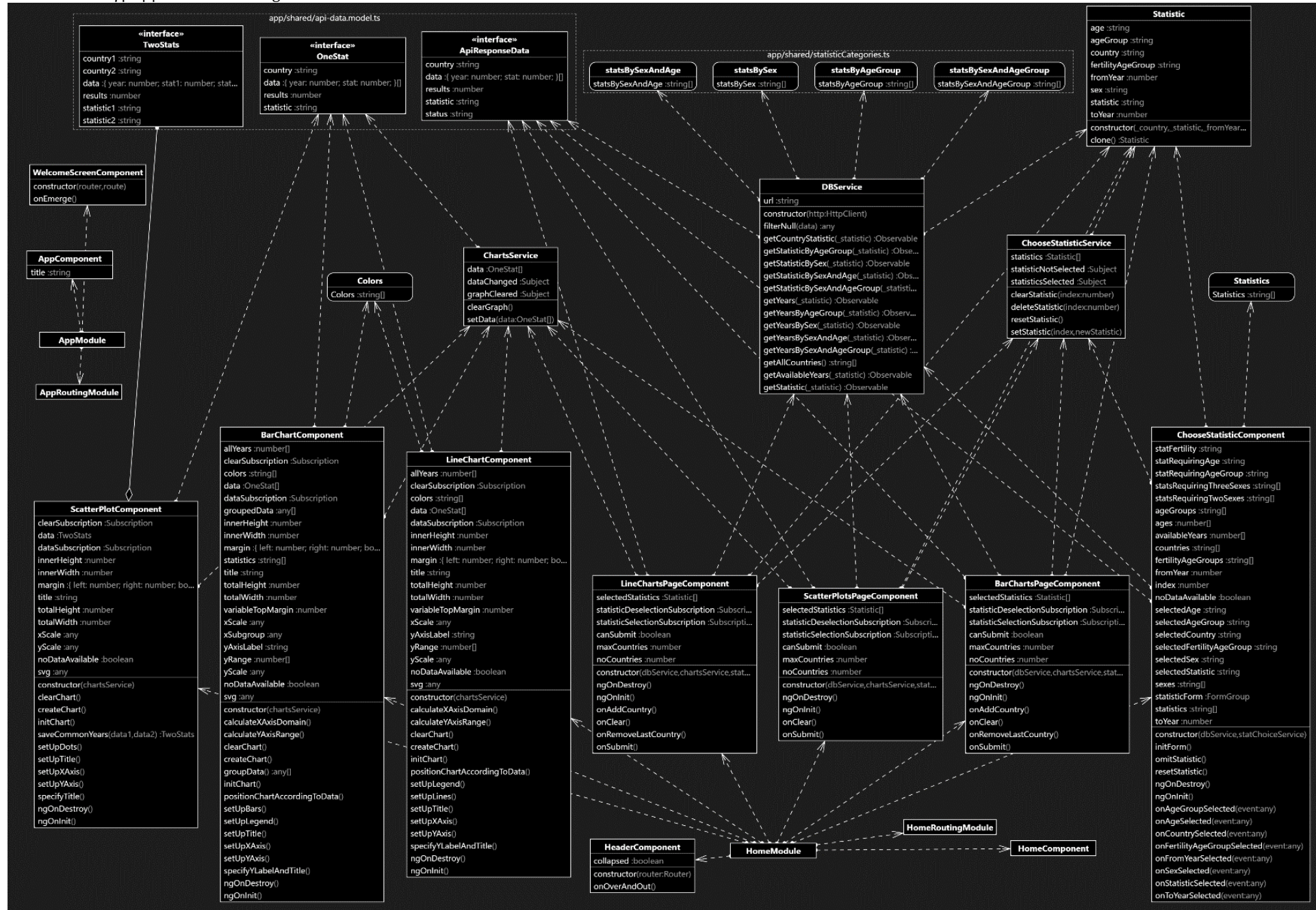
- Αρχικά επιλογή χώρας. Η λίστα των διαθέσιμων χωρών επιστρέφεται από ερώτηση στο API της βάση δεδομένων (στο background).
- Έπειτα επιλογή στατιστικού μεγέθους ανάμεσα στα διαθέσιμα στατιστικά μεγέθη. Η λίστα με τα στατιστικά μεγέθη είναι στατική και δεν προέρχεται από ερώτηση στην βάση.
- Επιλογή χρονικής περιόδου ανάλογα με την χώρα και το στατιστικό μέγεθος που επιλέχθηκε. Σημειώνεται ότι διαφορετικά ζεύγη χώρα-δείκτης έχουν διαφορετικές χρονικές περιόδους διαθέσιμες. Οι δύο λίστες επιλογής από χρόνο έως χρόνο καταρτίζονται μετά από ερώτηση στο API.
- Ανάλογα με το στατιστικό δείκτη που επιλέχθηκε μπορεί ο χρήστης να επιλέξει επιμέρους διαθέσιμες υποκατηγορίες όπως πχ. ηλικία, ηλικιακή ομάδα και φύλο.

Εφόσον η επιλογή των παραπάνω ολοκληρωθεί, ο χρήστης με το πάτημα ενός κουμπιού μπορεί να δει το γράφημα με την πληροφορία που επέλεξε.

2.3.1 ΔΙΑΓΡΑΜΜΑ ΠΑΚΕΤΩΝ ANGULAR

Η ανάπτυξη της εφαρμογής δεν έγινε αντικειμενοστρεφώς. Η Angular χρησιμοποιεί modules, components και services μεταξύ άλλων. Ένα ενδεικτικό διάγραμμα πακέτων παρουσιάζεται στην Εικόνα 4

Εικόνα 4: Διάγραμμα πακέτων Angular



3 ΥΠΟΔΕΙΓΜΑΤΑ ΕΡΩΤΗΣΕΩΝ ΚΑΙ ΑΠΑΝΤΗΣΕΩΝ

Σε αυτή την ενότητα παρατίθενται παραδείγματα των τριών κατηγοριών γραφημάτων για διαφορετικές επιλογές στατιστικών δεικτών.

3.1 ΕΠΙΛΟΓΗ ΣΤΑΤΙΣΤΙΚΟΥ ΔΕΙΚΤΗ

Μια ενδεικτική επιλογή στατιστικού δείκτη παρουσιάζεται στην Εικόνα 5.

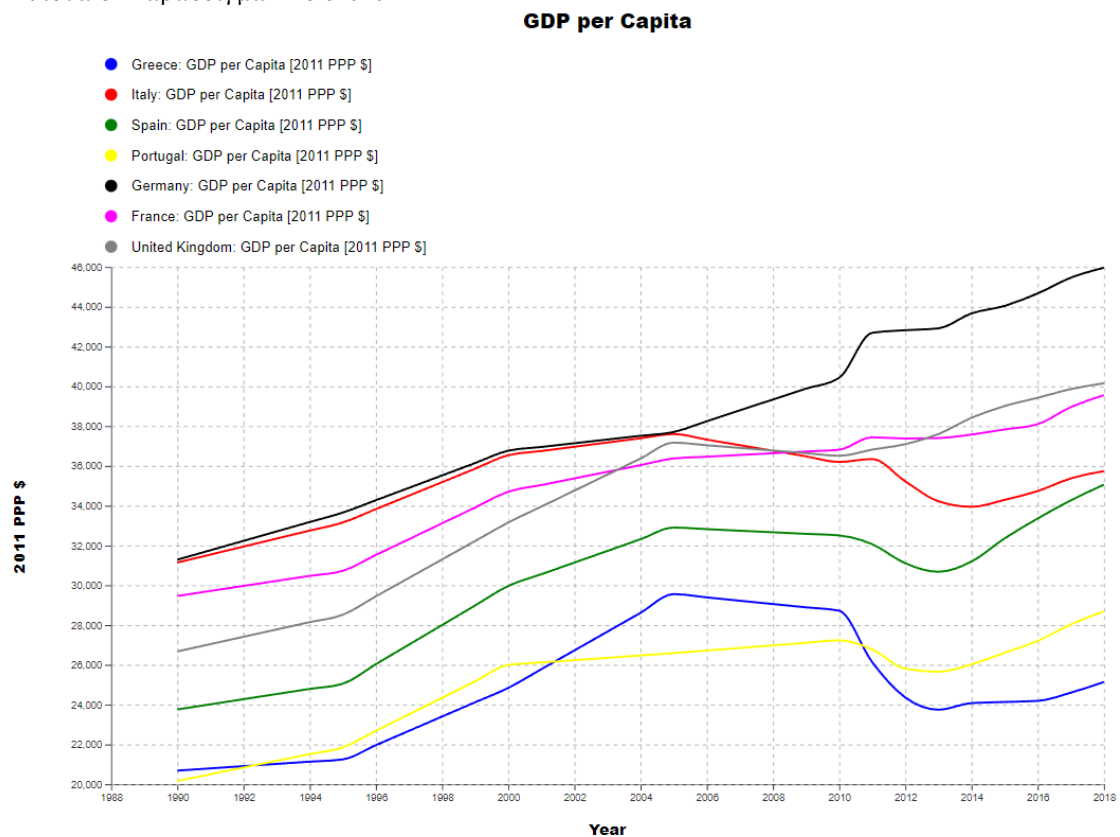
Εικόνα 5: Παράδειγμα επιλογής στατιστικού δείκτη

Country	Statistic	From Year	To Year
Nepal	Mid-Year Population (by Age Group)	1971	2050
		Sex	Age Group
		Both Sexes	All Ages

3.2 LINE CHARTS

Ένα παράδειγμα απεικόνισης line chart για σύγκριση του κατά κεφαλή εισοδήματος ευρωπαϊκών χωρών παρουσιάζεται στην Εικόνα 6.

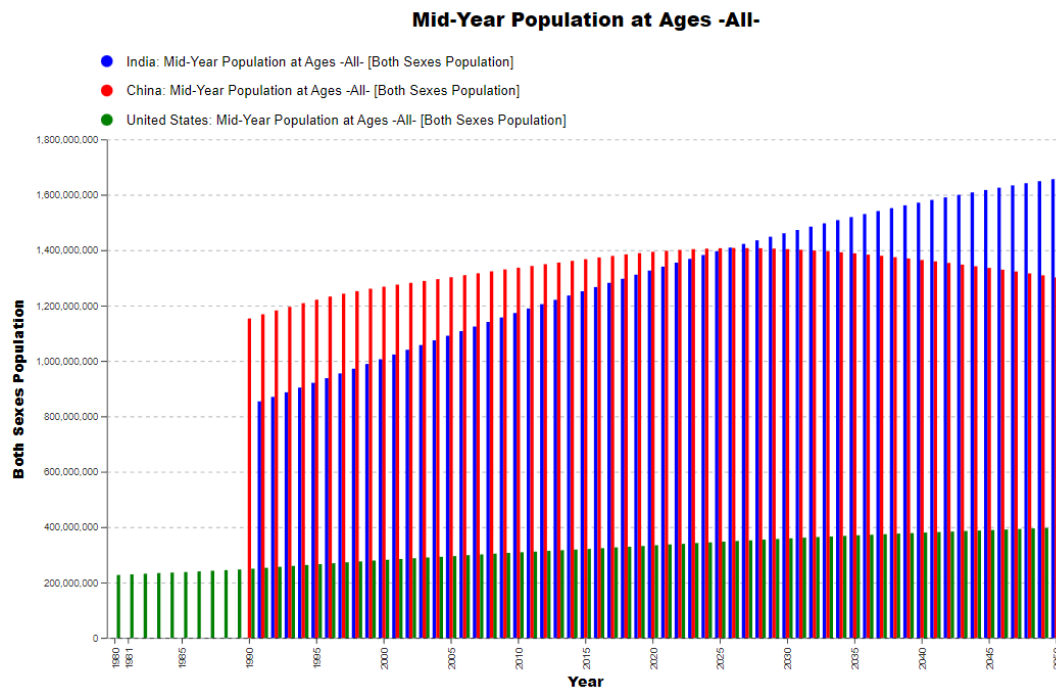
Εικόνα 6: Παράδειγμα line chart



3.3 BAR CHARTS

Ένα παράδειγμα απεικόνισης bar chart, για σύγκριση της πληθυσμιακή εξέλιξης στις ραγδαία αναπτυσσόμενες χώρες Ινδία και Κίνα απέναντι στις Ηνωμένες Πολιτείες, παρουσιάζεται στην Εικόνα 7.

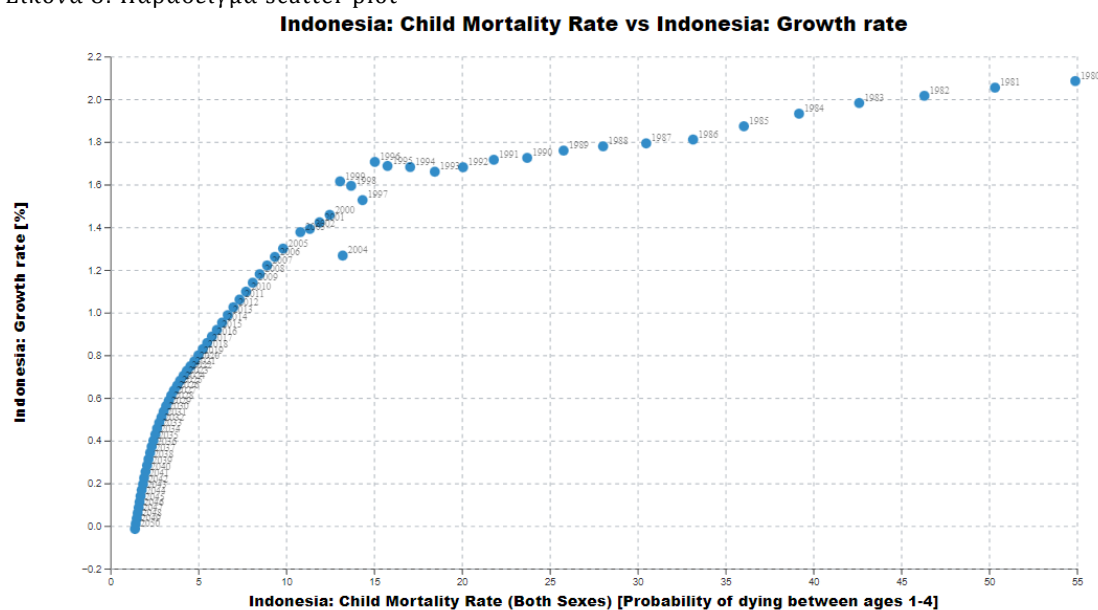
Εικόνα 7: Παράδειγμα bar chart



3.4 SCATTER PLOT

Ένα παράδειγμα απεικόνισης scatter plot για την συσχέτιση ρυθμού ανάπτυξης με παιδική θνησιμότητα στην Ινδονησία παρουσιάζεται στην Εικόνα 8.

Εικόνα 8: Παράδειγμα scatter plot



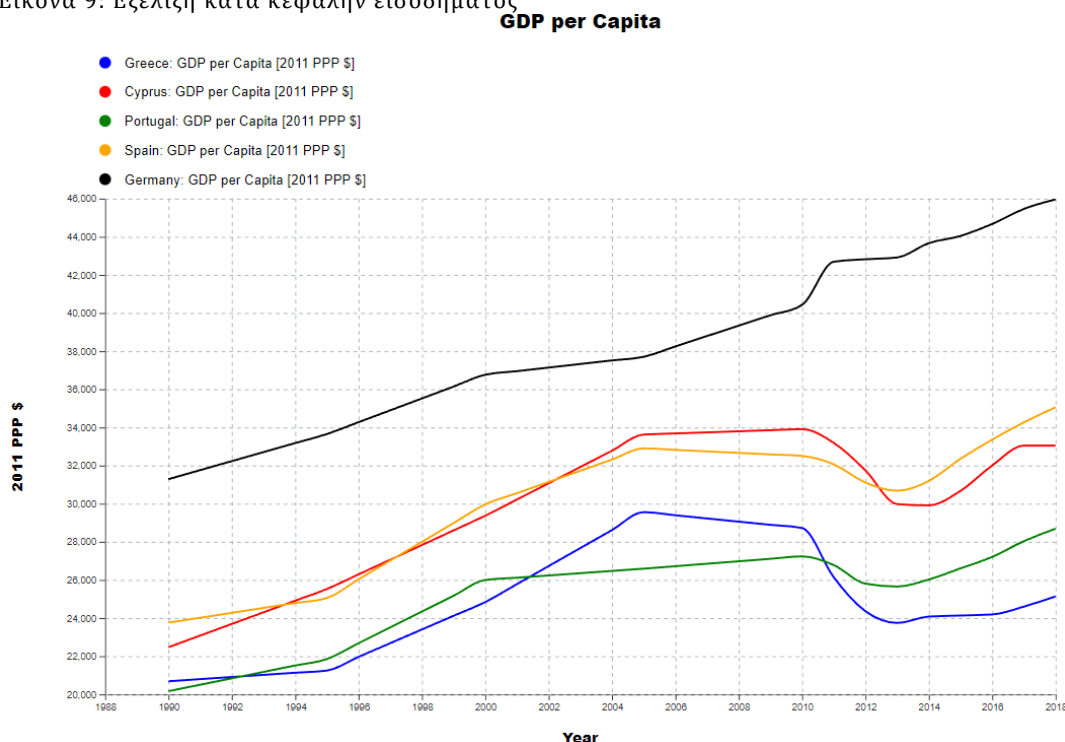
4 ΜΕΛΕΤΗ ΔΕΔΟΜΕΝΩΝ

Επιλέγεται να γίνει μια σύντομη μελέτη της οικονομικής κρίσης του 2008 και πως επηρέασε αυτή τις χώρες του ευρωπαϊκού νότου (Ελλάδα, Πορτογαλία, Ισπανία, Κύπρος) σε αντίθεση με τις χώρες της βόρειας Ευρώπης.

4.1 ΚΑΤΑ ΚΕΦΑΛΗΝ ΕΙΣΟΔΗΜΑ

Αρχικά παρουσιάζεται στην Εικόνα 9 η εξέλιξη στον χρόνο του κατά κεφαλή εισοδήματος στις χώρες του νότου σε σύγκριση με την Γερμανία για όλη την περίοδο που υπάρχουν στατιστικά δεδομένα.

Εικόνα 9: Εξέλιξη κατά κεφαλήν εισοδήματος



Η διαφορά ευρωπαϊκού νότου και βορρά ως προς την ανθεκτικότητα της οικονομίας είναι φανερή.

Η Γερμανία παρουσιάζει μόνο μια στασιμότητα στο κατά κεφαλήν εισόδημα την περίοδο 2011-2013 και αμέσως μετά συνεχίζει ανοδική πορεία. Σε αντίθεση, οι 4 χώρες του ευρωπαϊκού νότου υπόκεινται σημαντική μείωση των εισοδημάτων τους την περίοδο 2010-2013. Έπειτα αρχίζουν να ανακάμπτουν με τους δικούς τους ρυθμούς η κάθε μια.

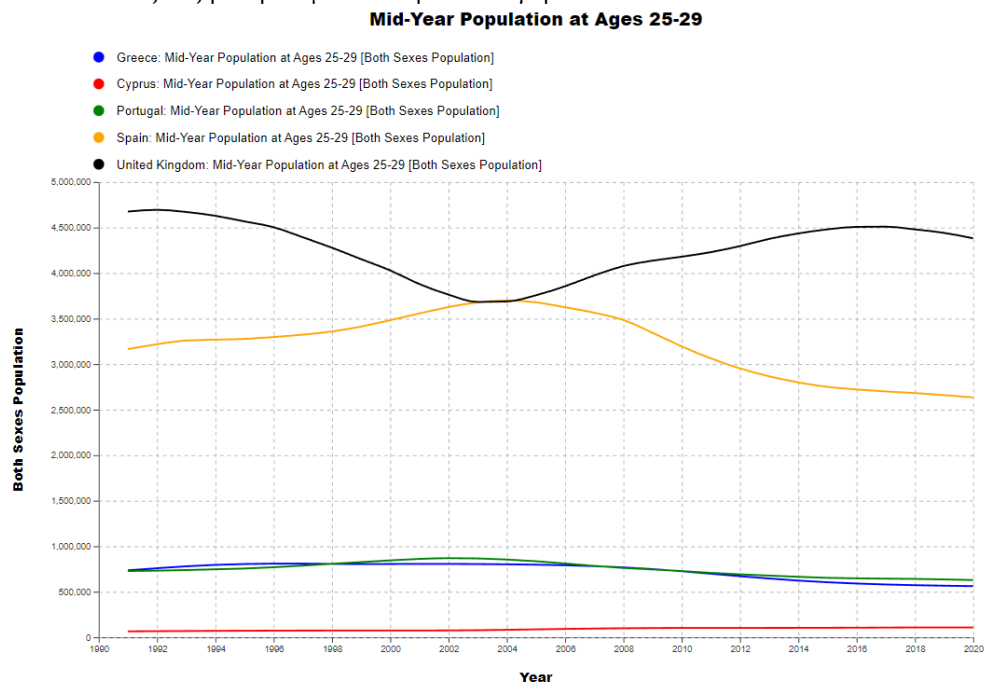
4.2 BRAIN DRAIN

Ένα φαινόμενο που παρατηρήθηκε έντονα τα χρόνια της οικονομικής κρίσης και μετά είναι το λεγόμενο brain drain, ήτοι η οικονομική μετανάστευση ατόμων υψηλής πανεπιστημιακής κατάρτισης.

Στο γράφημα της Εικόνα 10 παρουσιάζεται η εξέλιξη του πληθυσμού ηλικίας 25-29 κατά την περίοδο πριν και μετά την οικονομική κρίση για τις χώρες του ευρωπαϊκού

νότου με αντίθεση με το Ηνωμένο Βασίλειο, έναν από τους πιο δημοφιλής προορισμούς νέων ευρωπαϊών οικονομικών μεταναστών.

Εικόνα 10: Εξέλιξη πληθυσμού στο ηλικιακό γκρούπ 25-29



Στο γράφημα φαίνεται η μείωση του πληθυσμού σε Ισπανία, Ελλάδα και Πορτογαλία σε αντίθεση με την σημαντική αύξηση του πληθυσμού στις ηλικίες 25-29 στο Ηνωμένο Βασίλειο. Αυτές οι τάσεις ξεκινάνε από περίπου το 2003, δηλαδή πριν το οικονομικό κραχ του 2008, αλλά συνεχίζουν για αρκετά χρόνια μετά.

4.3 ΑΝΑΠΤΥΞΗ ΚΑΙ ΠΛΗΘΥΣΜΟΣ

Η οικονομική θεωρία υποστηρίζει ότι η οικονομική ανάπτυξη συνδέεται με το υψηλής ειδίκευσης ανθρώπινο δυναμικό μιας χώρας.

Στα παρακάτω γραφήματα παρουσιάζονται συσχετιζόμενοι οι δείκτες κατά κεφαλή εισόδημα με πληθυσμιακή εξέλιξη στις ηλικίες 25-29 για τις χώρες Ελλάδα, Ισπανία και Ηνωμένο Βασίλειο.

Εδώ τα συμπεράσματα είναι πιο δύσκολα.

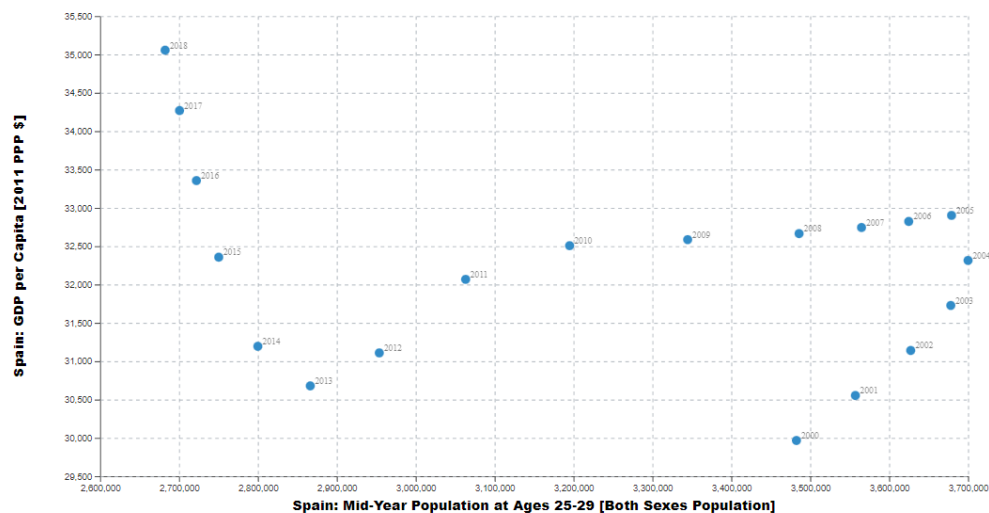
- Για την Ελλάδα παρατηρείται μια συσχέτιση της πληθυσμιακής εξέλιξης στις ηλικίες 25-29 με το κατά κεφαλή εισόδημα την περίοδο 2005 – 2013. Όσο μειώνεται ο πληθυσμός, μειώνεται και το κατά κεφαλή εισόδημα.
- Στην Ισπανία παρατηρείται μια στασιμότητα στο κατά κεφαλή εισόδημα από το 2005 έως το 2010 ενώ ο πληθυσμός μειώνεται και μια μείωση του εισοδήματος την περίοδο 2010-2013.
- Στο Ηνωμένο Βασίλειο αντίθετα παρατηρείται αύξηση στο κατά κεφαλήν εισόδημα την περίοδο 2010-2016 καθώς ο πληθυσμός (ηλικίες 25-29) αυξάνεται.

Φυσικά πρέπει να σημειωθεί εδώ ότι συσχέτιση και σημαίνει και αιτιότητα.

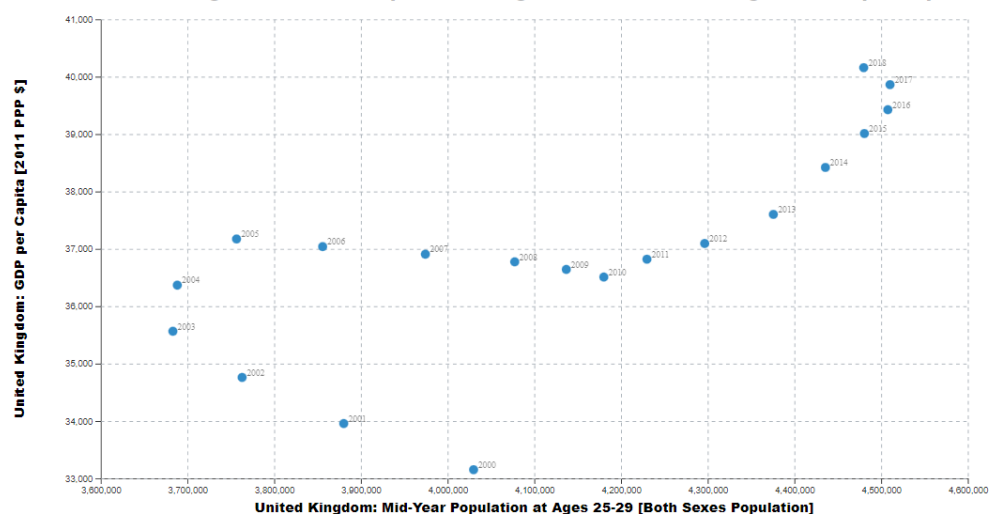
Εικόνα 11: Κατά κεφαλή εισόδημα με πληθυσμιακή εξέλιξη για την Ελλάδα

Greece: Mid-Year Population at Ages 25-29 vs Greece: GDP per Capita

Εικόνα 12: Κατά κεφαλή εισόδημα με πληθυσμιακή εξέλιξη για την Ισπανία

Spain: Mid-Year Population at Ages 25-29 vs Spain: GDP per Capita

Εικόνα 13: Κατά κεφαλή εισόδημα με πληθυσμιακή εξέλιξη για το Ηνωμένο Βασίλειο

United Kingdom: Mid-Year Population at Ages 25-29 vs United Kingdom: GDP per Capita

5 ΛΟΙΠΑ ΣΧΟΛΙΑ

5.1 TO DO LIST

Παρακάτω παρατίθενται ορισμένες εκκρεμότητες που έχουν αναγνωριστεί αλλά δεν υπήρξε χρόνος να υλοποιηθούν

- Ενσωμάτωση και χρήση των στοιχείων έκτασης στην γραφική απεικόνιση.
- Προσθήκη κουμπιού επιλογής «Remove All Statistic» αντί η διαγραφή να γίνεται ένα-ένα στο front-end.
- Προσθήκη μικρών κύκλων στα line charts στα σημεία που υπάρχουν δεδομένα.
- Προσθήκη tooltip με ανάλογη πληροφορία όταν ο χρήστης κάνει hover το ποντίκι πάνω από ένα line/bar στα αντίστοιχα charts.

6 ΠΑΡΑΡΤΗΜΑ Α

Η κατασκευή του πίνακα countries και το φόρτωμα των αντίστοιχων δεδομένων έγινε με τον παρακάτω κώδικα sql.

```
drop database if exists mye030_countries_stats;  
create database mye030_countries_stats;  
use mye030_countries_stats;
```

```
drop table if exists countries;  
create table countries (  
    country_index int not null,  
    iso varchar(5),  
    iso3 varchar(5),  
    iso_code int,  
    fips varchar(5),  
    display_name varchar(255),  
    official_name varchar(255),  
    capital varchar(255),  
    continent varchar(255),  
    currency_code varchar(5),  
    currency_name varchar(255),  
    phone varchar(20),  
    region_code int,  
    region_name varchar(255),  
    sub_region_code int,  
    sub_region_name varchar(255),  
    intermediate_region_code int,  
    intermediate_region_name varchar(255),  
    status varchar(255),  
    developed_or_developing varchar(255),  
    small_island_developing_states_sids varchar(10),  
    land_locked_developing_countries_lldc varchar(10),  
    least_developed_countries_ldc varchar(10),  
    area_sqkm int,  
    population int,  
    primary key (country_index)  
);
```

```
load data local infile  
'C:/Users/panou/Dropbox/02_Edu/01_Uni/06_Uoi/04_Courses/MYE030/Project/  
mye030_CountriesStats/Data/processed/countries/countriesFinal.csv'  
into table countries  
fields terminated by ','  
optionally enclosed by '"'  
lines terminated by '\r\n'  
IGNORE 1 LINES;
```

Η κατασκευή των πινάκων των δημογραφικών στοιχείων και το φόρτωμα των αντίστοιχων δεδομένων έγινε με τον παρακάτω κώδικα sql.

```
use mye030_countries_stats;
```

```
drop table if exists fertility_rates; show warnings;
```

```
create table fertility_rates (  
    country_index int not null,  
    country_code varchar(5),  
    country_name varchar(255),  
    year int not null,  
    fertility_rate_15_19 float,  
    fertility_rate_20_24 float,  
    fertility_rate_25_29 float,  
    fertility_rate_30_34 float,  
    fertility_rate_35_39 float,  
    fertility_rate_40_44 float,  
    fertility_rate_45_49 float,  
    total_fertility_rate float,  
    gross_reproduction_rate float,  
    sex_ratio_at_birth float,  
    primary key (country_index, year),  
    foreign key (country_index) references countries(country_index)  
    on delete cascade  
);
```

```
load data local infile
```

```
'C:/Users/panou/Dropbox/02_Edu/01_Uni/06_Uol/04_Courses/MYE030/Project/  
mye030_CountriesStats/Data/processed/international/age_specific_fertility_rates_final.csv'
```

```
into table fertility_rates
```

```
fields terminated by ','
```

```
lines terminated by '\r\n'
```

```
IGNORE 1 LINES;
```

```
drop table if exists birth_death_rates; show warnings;
```

```
create table birth_death_rates (  
    country_index int not null,  
    country_code varchar(5),  
    country_name varchar(255),  
    year int not null,  
    crude_birth_rate float,  
    crude_death_rate float,  
    net_migration float,  
    rate_natural_increase float,  
    growth_rate float,  
    primary key (country_index, year),  
    foreign key (country_index) references countries(country_index)  
    on delete cascade  
);
```



```
load data local infile
'C:/Users/panou/Dropbox/02_Edu/01_Uni/06_Uol/04_Courses/MYE030/Project/
mye030_CountriesStats/Data/processed/international/birth_death_growth_rates_final.csv'
into table birth_death_rates
fields terminated by ','
lines terminated by '\r\n'
IGNORE 1 LINES;
```

```
drop table if exists country_areas;
create table country_areas (
    country_index int not null,
    country_code varchar(5),
    country_name varchar(255),
    country_area int,
    primary key (country_index),
    foreign key (country_index) references countries(country_index)
    on delete cascade
);
```

```
load data local infile
'C:/Users/panou/Dropbox/02_Edu/01_Uni/06_Uol/04_Courses/MYE030/Project/
mye030_CountriesStats/Data/processed/international/country_names_area_final.csv'
into table country_areas
fields terminated by ','
lines terminated by '\r\n'
IGNORE 1 LINES;
```

```
drop table if exists midyear_population_5yr;
create table midyear_population_5yr (
    country_index int not null,
    country_code varchar(5),
    country_name varchar(255),
    year int not null,
    total_flag varchar(5) not null,
    starting_age smallint not null,
    age_group_indicator varchar(5),
    ending_age smallint,
    midyear_population int,
    midyear_population_male int,
    midyear_population_female int,
    primary key (country_index, year, total_flag, starting_age),
    foreign key (country_index) references countries(country_index)
    on delete cascade
);
```

```
load data local infile
'C:/Users/panou/Dropbox/02_Edu/01_Uni/06_Uol/04_Courses/MYE030/Project/
mye030_CountriesStats/Data/processed/international/midyear_population_5yr_age_sex_final.csv'
into table midyear_population_5yr
fields terminated by ','
lines terminated by '\r\n' IGNORE 1 LINES;
```

```
drop table if exists midyear_population_1yr;
create table midyear_population_1yr (
    country_index int not null,
    country_code varchar(5),
    country_name varchar(255),
    year int not null,
    sex varchar(10) not null,
    max_age smallint,
    population_age_0 int,
    population_age_1 int,
    population_age_2 int,
    population_age_3 int,
    population_age_4 int,
    population_age_5 int,
    population_age_6 int,
    population_age_7 int,
    population_age_8 int,
    population_age_9 int,
    population_age_10 int,
    population_age_11 int,
    population_age_12 int,
    population_age_13 int,
    population_age_14 int,
    population_age_15 int,
    population_age_16 int,
    population_age_17 int,
    population_age_18 int,
    population_age_19 int,
    population_age_20 int,
    population_age_21 int,
    population_age_22 int,
    population_age_23 int,
    population_age_24 int,
    population_age_25 int,
    population_age_26 int,
    population_age_27 int,
    population_age_28 int,
    population_age_29 int,
    population_age_30 int,
    population_age_31 int,
    population_age_32 int,
    population_age_33 int,
    population_age_34 int,
    population_age_35 int,
    population_age_36 int,
    population_age_37 int,
    population_age_38 int,
    population_age_39 int,
    population_age_40 int,
    population_age_41 int,
    population_age_42 int,
```

population_age_43 int,
population_age_44 int,
population_age_45 int,
population_age_46 int,
population_age_47 int,
population_age_48 int,
population_age_49 int,
population_age_50 int,
population_age_51 int,
population_age_52 int,
population_age_53 int,
population_age_54 int,
population_age_55 int,
population_age_56 int,
population_age_57 int,
population_age_58 int,
population_age_59 int,
population_age_60 int,
population_age_61 int,
population_age_62 int,
population_age_63 int,
population_age_64 int,
population_age_65 int,
population_age_66 int,
population_age_67 int,
population_age_68 int,
population_age_69 int,
population_age_70 int,
population_age_71 int,
population_age_72 int,
population_age_73 int,
population_age_74 int,
population_age_75 int,
population_age_76 int,
population_age_77 int,
population_age_78 int,
population_age_79 int,
population_age_80 int,
population_age_81 int,
population_age_82 int,
population_age_83 int,
population_age_84 int,
population_age_85 int,
population_age_86 int,
population_age_87 int,
population_age_88 int,
population_age_89 int,
population_age_90 int,
population_age_91 int,
population_age_92 int,
population_age_93 int,

```
population_age_94 int,
population_age_95 int,
population_age_96 int,
population_age_97 int,
population_age_98 int,
population_age_99 int,
population_age_100 int,
primary key (country_index, year, sex),
foreign key (country_index) references countries(country_index)
on delete cascade
);

load data local infile
'C:/Users/panou/Dropbox/02_Edu/01_Uni/06_Uol/04_Courses/MYE030/Project/
mye030_CountriesStats/Data/processed/international/midyear_population_age_sex_final.csv'
into table midyear_population_1yr
fields terminated by ','
lines terminated by '\r\n'
IGNORE 1 LINES;

drop table if exists midyear_population;
create table midyear_population (
    country_index int not null,
    country_code varchar(5),
    country_name varchar(255),
    year int not null,
    midyear_population int,
    primary key (country_index, year),
    foreign key (country_index) references countries(country_index)
    on delete cascade
);

'C:/Users/panou/Dropbox/02_Edu/01_Uni/06_Uol/04_Courses/MYE030/Project/
mye030_CountriesStats/Data/processed/international/midyear_population_final.csv'
into table midyear_population
fields terminated by ','
lines terminated by '\r\n'
IGNORE 1 LINES;

drop table if exists life_expectancy;
create table life_expectancy (
    country_index int not null,
    country_code varchar(5),
    country_name varchar(255),
    year int not null,
    infant_mortality float,
    infant_mortality_male float,
    infant_mortality_female float,
    life_expectancy float,
    life_expectancy_male float,
    life_expectancy_female float,
```

```
mortality_rate_under5 float,  
mortality_rate_under5_male float,  
mortality_rate_under5_female float,  
mortality_rate_1to4 float,  
mortality_rate_1to4_male float,  
mortality_rate_1to4_female float,  
primary key (country_index, year),  
foreign key (country_index) references countries(country_index)  
on delete cascade  
);  
  
load data local infile  
'C:/Users/panou/Dropbox/02_Edu/01_Uni/06_Uol/04_Courses/MYE030/Project/  
mye030_CountriesStats/Data/processed/international/mortality_life_expectancy_final.csv'  
into table life_expectancy  
fields terminated by ','  
lines terminated by '\r\n'  
IGNORE 1 LINES;
```

Η κατασκευή των πινάκων των οικονομικών στοιχείων και το φόρτωμα των αντίστοιχων δεδομένων έγινε με τον παρακάτω κώδικα sql.

```
use mye030_countries_stats;
```

```
drop table if exists domestic_credits;
create table domestic_credits (
    country_index int not null,
    country_name varchar(255),
    year int not null,
    domestic_credits float,
    primary key (country_index, year),
    foreign key (country_index) references countries(country_index)
    on delete cascade
);
```

```
load data local infile
'C:/Users/panou/Dropbox/02_Edu/01_Uni/06_UoI/04_Courses/MYE030/Project/
mye030_CountriesStats/Data/processed/income/domestic_credits_final.csv'
into table domestic_credits
fields terminated by ','
lines terminated by '\r\n'
IGNORE 1 LINES;
```

```
drop table if exists estimated_gni_female;
create table estimated_gni_female (
    country_index int not null,
    country_name varchar(255),
    year int not null,
    estimated_gni_female float,
    primary key (country_index, year),
    foreign key (country_index) references countries(country_index)
    on delete cascade
);
```

```
load data local infile
'C:/Users/panou/Dropbox/02_Edu/01_Uni/06_UoI/04_Courses/MYE030/Project/
mye030_CountriesStats/Data/processed/income/estimated_gni_female_final.csv'
into table estimated_gni_female
fields terminated by ','
lines terminated by '\r\n'
IGNORE 1 LINES;
```

```
drop table if exists estimated_gni_male; show warnings;
create table estimated_gni_male (
    country_index int not null,
    country_name varchar(255),
    year int not null,
    estimated_gni_male float,
    primary key (country_index, year),
    foreign key (country_index) references countries(country_index)
    on delete cascade
);
```

```
load data local infile
'C:/Users/panou/Dropbox/02_Edu/01_Uni/06_Uol/04_Courses/MYE030/Project/
mye030_CountriesStats/Data/processed/income/estimated_gni_male_final.csv'
into table estimated_gni_male
fields terminated by ','
lines terminated by '\r\n'
IGNORE 1 LINES;
```

```
drop table if exists gdp_per_capita; show warnings;
create table gdp_per_capita (
    country_index int not null,
    country_name varchar(255),
    year int not null,
    gdp_per_capita float,
    primary key (country_index, year),
    foreign key (country_index) references countries(country_index)
    on delete cascade
);
```

```
load data local infile
'C:/Users/panou/Dropbox/02_Edu/01_Uni/06_Uol/04_Courses/MYE030/Project/
mye030_CountriesStats/Data/processed/income/gdp_per_capita_final.csv'
into table gdp_per_capita
fields terminated by ','
lines terminated by '\r\n'
IGNORE 1 LINES;
```

```
drop table if exists gdp_total; show warnings;
create table gdp_total (
    country_index int not null,
    country_name varchar(255),
    year int not null,
    gdp_total float,
    primary key (country_index, year),
    foreign key (country_index) references countries(country_index)
    on delete cascade
);
```

```
load data local infile
'C:/Users/panou/Dropbox/02_Edu/01_Uni/06_Uol/04_Courses/MYE030/Project/
mye030_CountriesStats/Data/processed/income/gdp_total_final.csv'
into table gdp_total
fields terminated by ','
lines terminated by '\r\n'
IGNORE 1 LINES;
```

```
drop table if exists gni_per_capita;
create table gni_per_capita (
    country_index int not null,
    country_name varchar(255),
    year int not null,
    gni_per_capita int,
    primary key (country_index, year),
    foreign key (country_index) references countries(country_index)
    on delete cascade
);
```

```
load data local infile
'C:/Users/panou/Dropbox/02_Edu/01_Uni/06_Uol/04_Courses/MYE030/Project/
mye030_CountriesStats/Data/processed/income/gni_per_capita_final.csv'
into table gni_per_capita
fields terminated by ','
lines terminated by '\r\n'
IGNORE 1 LINES;
```

```
drop table if exists gross_fixed_capital_formation; show warnings;
create table gross_fixed_capital_formation (
    country_index int not null,
    country_name varchar(255),
    year int not null,
    gross_fixed_capital_formation float,
    primary key (country_index, year),
    foreign key (country_index) references countries(country_index)
    on delete cascade
);
```

```
load data local infile
'C:/Users/panou/Dropbox/02_Edu/01_Uni/06_Uol/04_Courses/MYE030/Project/
mye030_CountriesStats/Data/processed/income/gross_fixed_capital_formation_final.csv'
into table gross_fixed_capital_formation
fields terminated by ','
lines terminated by '\r\n'
IGNORE 1 LINES;
```

```
drop table if exists income_index;
create table income_index (
    country_index int not null,
    country_name varchar(255),
    year int not null,
    income_index float,
    primary key (country_index, year),
    foreign key (country_index) references countries(country_index)
    on delete cascade
);
```



```
load data local infile
'C:/Users/panou/Dropbox/02_Edu/01_Uni/06_Uol/04_Courses/MYE030/Project/
mye030_CountriesStats/Data/processed/income/income_index_final.csv'
into table income_index
fields terminated by ','
lines terminated by '\r\n'
IGNORE 1 LINES;
```

```
drop table if exists labour_share_of_gdp; show warnings;
create table labour_share_of_gdp (
    country_index int not null,
    country_name varchar(255),
    year int not null,
    labour_share_of_gdp float,
    primary key (country_index, year),
    foreign key (country_index) references countries(country_index)
    on delete cascade
);
```

```
load data local infile
'C:/Users/panou/Dropbox/02_Edu/01_Uni/06_Uol/04_Courses/MYE030/Project/
mye030_CountriesStats/Data/processed/income/labour_share_of_gdp_final.csv'
into table labour_share_of_gdp
fields terminated by ','
lines terminated by '\r\n'
IGNORE 1 LINES;
```

Σημειώνεται ότι για την φόρτωση των δεδομένων πρέπει να τεθεί η μεταβλητή *local_infile* σε 1 με την εντολή *SET GLOBAL local_infile = 1;*

7 ΠΑΡΑΡΤΗΜΑ Β

GET getAllCountries

127.0.0.1:3000/api/v1/countries

Return a list of all available countries in the database. For some countries data may not be available.

Response: 200 Application/Json

```
json
{
  "status": "success",
  "results": 254,
  "data": [
    {
      "countries": [
        {
          "display_name": "Afghanistan",
          ..
        }
      ]
    }
  ]
}
```

GET getYears

127.0.0.1:3000/api/v1/years/Canada/GDP per Capita

Generic Request:

127.0.0.1:3000/api/v1/years/country/statistic/sex/age-group/starting-age

Return list of years for which the database holds records for country = "Canada" and statistic = "GDP per Capita".

statistic can be any of:

- Gross Reproduction Rate
- Sex Ratio at Birth
- Crude Birth Rate
- Crude Death Rate
- Net Migration Rate
- Rate of Natural Increase
- Growth Rate
- Income index
- Domestic Credit Provided by Financial Sector
- GDP per Capita
- Gross Domestic Product (GDP) (total)
- Gross Fixed Capital Formation
- Gross National Income (GNI) per Capita
- Labour Share of GDP

Response: 200 Application/Json

```
json
{
  "status": "success",
  "country": "Canada",
  "statistic": "GDP per Capita [2011 PPP $]",
  "results": 29,
  "data": [
    {
      "year": 1990,
      ..
    }
  ]
}
```

GET `getYearsBySex``127.0.0.1:3000/api/v1/years/Turkey/Life Expectancy at Birth/Male`

Generic Request:

`127.0.0.1:3000/api/v1/years/country/statistic/sex`

Return list of years for which the database holds records for country = "Turkey", statistic = "Life Expectancy at Birth" and sex = "Male".

statistic can be any of:

- Infant Mortality Rate
- Life Expectancy at Birth
- Under-5 Mortality Rate
- Child Mortality Rate
- Estimated Gross National Income per Capita

sex can be any of:

- Both Sexes
- Male
- Female

Response: 200 Application/Json

json

```
{
  "status": "success",
  "country": "Turkey",
  "statistic": "Life Expectancy at Birth (Male) [Years]",
  "results": 71,
  "data": [{ "year": 1980 }, ..]
}
```

GET `getYearsByAgeGroup``127.0.0.1:3000/api/v1/years/South Korea/Fertility Rate/age-group/20`

Generic Request:

`127.0.0.1:3000/api/v1/years/country/statistic/age-group/starting-age`

Return list of years for which the database holds records for country = "South Korea", statistic = "Fertility Rate" and age group starting at starting-age 20 i.e. [20-24].

statistic can be any of:

- Fertility Rate

starting age can be any of:

- Total
- 15
- 20
- 25
- 30
- 35
- 40
- 45

Response: 200 Application/Json

json

```
{
  "status": "success",
  "country": "South Korea",
  "statistic": "Fertility Rate between ages 20-24 [Births per 1,000 population]",
  "results": 61,
  "data": [{ "year": 1990 }, ..]
}
```

GET getYearsBySexAndAge

```
127.0.0.1:3000/api/v1/years/Italy/Mid-Year Population (by Age)/Male/age/37
```

Generic Request:

```
127.0.0.1:3000/api/v1/years/country/statistic/sex/age/age
```

Return list of years for which the database holds records for country = "Italy", statistic = "Mid-Year Population (by Age)", sex = "Male" and age = "37".

statistic can be any of:

- Mid-Year Population (by Age)

starting age can be any of: 0 ... 100

sex can be any of:

- Male
- Female

Response: 200 Application/Json

json

```
{
  "status": "success",
  "country": "Italy",
  "statistic": "Mid-Year Population at Age 37 [Male Population]",
  "results": 60,
  "data": [{ "year": 1991 }, ..]
}
```

GET getYearsBySexAndAgeGroup

```
127.0.0.1:3000/api/v1/years/Germany/Mid-Year Population (by Age Group)/Female/age-group/25
```

Generic Request:

```
127.0.0.1:3000/api/v1/years/country/statistic/sex/age-group/starting-age
```

Return list of years for which the database holds records for country = "Germany", statistic = "Mid-Year Population (by Age Group)", sex = "Female" and age group starting at age 25 i.e. [25-29].

statistic can be any of:

- Mid-Year Population (by Age Group)

starting age can be any of:

- all
- 0
- 5
- 15
- 20
- 25
- ...
- 100

sex can be any of:

- Both Sexes
- Male
- Female

Response: 200 Application/Json

json

```
{
  "status": "success",
  "country": "Germany",
  "statistic": "Mid-Year Population at Ages 25-29 [Female Population]",
  "results": 60,
  "data": [{ "year": 1991 }, ..]
}
```

GET `getCountryStatistic``127.0.0.1:3000/api/v1/statistics/Armenia/Sex Ratio at Birth/1989/2050`

Generic Request:

`127.0.0.1:3000/api/v1/statistics/country/statistic/fromYear/toYear`

Return a list of year - statistic pairs for country = "Armenia" and statistic = "Sex Ratio at Birth" for time interval between fromYear = "1989" and toYear = "2050".

statistic can be any of:

- Gross Reproduction Rate
- Sex Ratio at Birth
- Crude Birth Rate
- Crude Death Rate
- Net Migration Rate
- Rate of Natural Increase
- Growth Rate
- Income index
- Domestic Credit Provided by Financial Sector
- GDP per Capita
- Gross Domestic Product (GDP) (total)
- Gross Fixed Capital Formation
- Gross National Income (GNI) per Capita
- Labour Share of GDP

Response: 200 Application/Json

e.g.

```
json
{
  "status": "success",
  "country": "Armenia",
  "statistic": "Sex Ratio at Birth [Male births per female birth]",
  "results": 62,
  "data": [ { "year": 1989, "stat": 1.0879 }, ... ]
}
```

GET `getCountryStatisticBySex``127.0.0.1:3000/api/v1/statistics/Austria/Child Mortality Rate/1989/2050/Both Sexes`

Generic Request:

`127.0.0.1:3000/api/v1/statistics/country/statistic/fromYear/toYear/sex`

Return a list of year - statistic pairs for country = "Austria", statistic = "Child Mortality Rate" for time interval between fromYear = "1989" and toYear = "2050" and for sex = "Both Sexes".

statistic can be any of:

- Infant Mortality Rate
- Life Expectancy at Birth
- Under-5 Mortality Rate
- Child Mortality Rate
- Estimated Gross National Income per Capita

sex can be any of:

- Both Sexes
- Male
- Female

Response: 200 Application/Json

```
json
{
  "status": "success",
  "country": "Austria",
  "statistic": "Child Mortality Rate (Both Sexes) [Probability of dying between ages 1-4]",
  "results": 60,
  "data": [ { "year": 1991, "stat": 1.54 }, ... ]
}
```

GET getCountryStatisticByAgeGroup

```
127.0.0.1:3000/api/v1/statistics/France/Fertility Rate/1989/2050/age-group/20
```

Generic Request:

```
127.0.0.1:3000/api/v1/statistics/country/statistic/fromYear/toYear/age-group/starting-age
```

Return a list of year - statistic pairs for country = "France", statistic = "Fertility Rate" for time interval between fromYear = "1989" and toYear = "2050" and age group starting at age = "20" i.e. [20-24].

statistic can be any of:

- Fertility Rate

starting age can be any of:

- Total
- 15
- 20
- 25
- 30
- 35
- 40
- 45

Response: 200 Application/Json

e.g.

json

```
{
  "status": "success",
  "country": "France",
  "statistic": "Fertility Rate between ages 20-24 [Births per 1,000 population]",
  "results": 61,
  "data": [ { "year": 1990, "stat": 76.7 }, ... ]
}
```

GET getCountryStatisticBySexAndAge

```
127.0.0.1:3000/api/v1/statistics/Poland/Mid-Year Population (by Age)/1989/2050/Male/age/20
```

Generic Request:

```
127.0.0.1:3000/api/v1/statistics/country/statistic/fromYear/toYear/sex/age/age
```

Return a list of year - statistic pairs for country = "Poland" and statistic = "Mid-Year Population (by Age)" for time interval between fromYear = "1989" and toYear = "2050", sex = Male and age = "20".

statistic can be any of:

- Mid-Year Population (by Age)

starting age can be any of: 0 ... 100

sex can be any of:

- Male
- Female

Response: 200 Application/Json

e.g.

json

```
{
  "status": "success",
  "country": "Poland",
  "statistic": "Mid-Year Population at Age 20 [Male Population]",
  "results": 62,
  "data": [ { "year": 1989, "stat": 252707 }, ... ]
}
```

GET `getCountryStatisticBySexAndAgeGroup`

```
127.0.0.1:3000/api/v1/statistics/Greece/Mid-Year Population (by Age Group)/1989/2050/Female/age-group/20
```

Generic Request:

```
127.0.0.1:3000/api/v1/statistics/country/statistic/fromYear/toYear/sex/age-group/starting-age
```

Return a list of year - statistic pairs for country = "Greece" and statistic = "Mid-Year Population (by Age Group)" for time interval between fromYear = "1989" and toYear = "2050" and age group starting with starting-age = "20" i.e. [20-24].

statistic can be any of:

- Mid-Year Population (by Age Group)

starting age can be any of:

- all
- 0
- 5
- 15
- 20
- 25
- ...
- 100

sex can be any of:

- Both Sexes
- Male
- Female

Response: 200 Application/Json

e.g.

json

```
{
  "status": "success",
  "country": "Greece",
  "statistic": "Mid-Year Population at Ages 20-24 [Female Population]",
  "results": 60,
  "data": [ { "year": 1991, "stat": 387490 }, ... ]
}
```