

ΠΡΟΧΩΡΗΜΕΝΑ ΘΕΜΑΤΑ
ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΕΦΑΡΜΟΓΩΝ
ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ
ΤΟ ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2019-2020

ΔΗΜΟΣ ΓΚΙΤΣΑΚΗΣ, ΑΜ 2425

ΣΠΥΡΙΔΩΝ ΚΑΛΟΥΔΗΣ, ΑΜ 2447

ΙΩΑΝΝΗΣ ΧΟΥΛΙΑΡΑΣ, ΑΜ 2631

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΜΑΪΟΣ 2020

ΙΣΤΟΡΙΚΟ ΠΡΟΗΓΟΥΜΕΝΩΝ ΕΚΔΟΣΕΩΝ

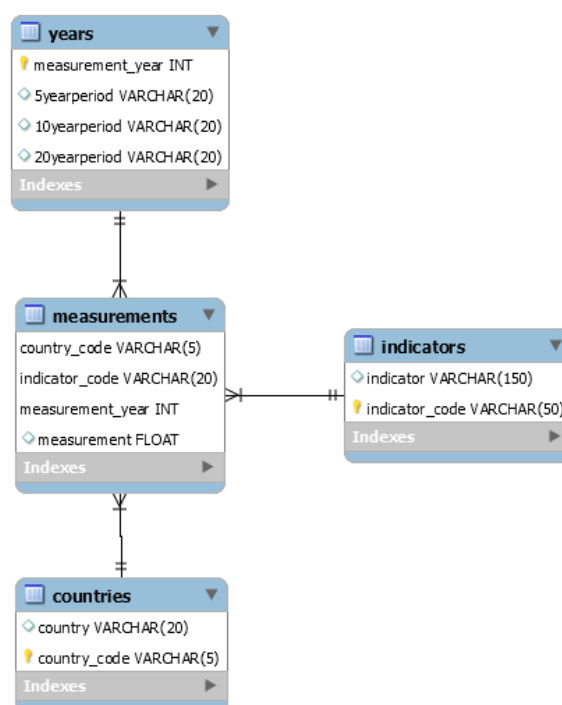
Ημερομηνία	Έκδοση	Περιγραφή	Συγγραφέας
2020/05/25	v1.0	Ολοκλήρωση φάσης 1,2,3.	Dev Team:
			Δήμος Γκιτσάκης
			Σπυρίδων Καλούδης
			Ιωάννης Χουλιάρας

1. ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Για την υλοποίηση της βάσης δημιουργήσαμε το database με όνομα WORLDBANK το οποίο έχει το παρακάτω σχήμα.

1.1. ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΛΟΓΙΚΟ ΕΠΙΠΕΔΟ

Το σχήμα της βάσης είναι το εξής:



Το παραπάνω σχήμα προέκυψε έπειτα από screenshot του MySQL workbench. Τα δεδομένα τα οποία χρησιμοποιήσαμε μετασχηματίστηκαν και φορτώθηκαν στη βάση μέσω κάποιων scripts που περιγράφονται παρακάτω.

Σαν τύπο αποθήκευσης στην MySQL χρησιμοποιήσαμε InnoDB. Οι υπόλοιπες ρυθμίσεις είναι οι default της MySQL.

2. ΤΡΟΠΟΠΟΙΗΣΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΦΟΡΤΩΣΗ ΒΑΣΗΣ

2.1. ΠΕΡΙΓΡΑΦΗ SCRIPT (PYTHON) : CORRECTCSVFORMAT.PY

Αρχικά έχουμε δύο πίνακες, έναν με indicator codes και έναν με τα ονόματα των αρχείων κάθε χώρας.

Η διαδικασία αρχίζει με την κλήση της συνάρτησης `bigParser()`, η οποία διαβάζει τα αρχεία των χωρών και για κάθε indicator γράφει τα δεδομένα (Indicator Name, Indicator Code) στο αρχείο `indicators.csv`. Έπειτα, καλεί την συνάρτηση `parser(line)`, η οποία χωρίζει σωστά τη γραμμή και για κάθε χρονιά κατασκευάζει μία γραμμή με τα πεδία Country Code, Indicator Code, χρονιά της μέτρησης, μέτρηση. Επειδή υπήρχαν πολλά null πεδία, υπάρχουν οι κατάλληλοι έλεγχοι και έχουν αντικατασταθεί με μηδενικά. Στη συνέχεια, αυτές οι γραμμές αποθηκεύονται σε ένα πίνακα και καλείται η συνάρτηση `writer(array)`, η οποία γράφει στο αρχείο `measurements.csv` τις γραμμές αυτού του πίνακα.

Αμέσως μετά, καλείται η συνάρτηση `writeCountries()`, η οποία γράφει στο αρχείο `countries.csv` τα πεδία Country Name, Country Code.

Τέλος, καλείται η συνάρτηση `writeYears()` η οποία με χρήση πολλαπλών if και flags, γράφει στο αρχείο `years.csv` τα πεδία χρονιά, γκρουπ πενταετίας, γκρουπ δεκαετίας, γκρουπ εικοσαετίας

2.2. ΠΕΡΙΓΡΑΦΗ SCRIPT (PYTHON) : FILLDATABASE.PY

Αρχικά γίνεται η σύνδεση στην βάση WORLDBANK που αναφέραμε παραπάνω. Μια σημαντική σημείωση είναι ότι για να φορτωθεί η βάση σε διαφορετικό μηχάνημα πρέπει να αλλάξει η μεταβλητή `passwd` στο σωστό κωδικό του root user της MySQL. Για παράδειγμα χρησιμοποιήθηκε το "screamformeioannina".

Μέσω της εντολής `mydb.cursor().execute("εντολή")` οποιαδήποτε όρισμα δώσουμε είναι μια εντολή σε SQL η οποία εκτελείται στη βάση.

Εάν έχουμε ήδη δημιουργήσει τους πίνακες της βάσης το script ξεκινά με την συνάρτηση `dropTables()` η οποία διαγράφει τους πίνακες.

Καλείται η `createTables()` η οποία δημιουργεί τους πίνακες της βάσης με τα επιθυμητά πεδία. Τα σωστά primary keys, foreign keys και σωστούς ελέγχους για διαγραφή εγγραφών (on delete cascade on update cascade).

Στη συνέχεια καλούνται οι συναρτήσεις: `fillCountries()`, `fillIndicators()`, `fillYears()` και `fillMeasurements()` οι οποίες θα εισάγουν στους πίνακες της βάσης τα δεδομένα.

Τέλος γίνεται commit και κλείσιμο της σύνδεσης με τη βάση.

Τα παραπάνω scripts και csv αρχεία περιλαμβάνονται στον φάκελο project-folder/python και για η εκτέλεση των scripts γίνονται με τις εντολές : python3 CorrectCSVFormat.py και python3 fillDatabase.py

3. NODE.JS (API) : ΕΠΙΚΟΙΝΩΝΙΑ ΜΕ ΤΗΝ ΒΑΣΗ

Στο project-folder/server υπάρχει ένα αρχείο database.js όπου θα υλοποιηθεί το API μέσω του node.js

Για την υλοποίηση χρειαζόμαστε μία σύνδεση στην βάση δεδομένων. Αυτό επιτυγχάνεται με την κλήση `mysql.createPool({"εντολές"})` όπου στις εντολές συμπληρώνουμε τα πεδία όπως host, user, password, database και port. Σε κάθε σύστημα θα πρέπει να συμπληρώνονται οι σωστές τιμές.

Επίσης χρησιμοποιούμε την μέθοδο `express()` όπου την κρατάμε σε μία μεταβλητή `app`. Τα δεδομένα που θα επιστρέφονται θα είναι σε μορφή JSON.

Για την πραγματοποίηση μίας ερώτησης χρησιμοποιούμε την εντολή `app.route("url")` όπου στο url θα έχουμε συγκεκριμένες τιμές (χώρες, indicators, years etc) . Μετά από επεξεργασία του url είμαστε έτοιμοι να στείλουμε την ερώτηση μέσω του `app.query()`. Η μέθοδος επιστρέφει είτε error είτε ένα JSON αρχείο με τα δεδομένα της κάθε ερώτησης.

Τέλος πρέπει να ορίσουμε που "ακούει" το api μας. Αυτό επιτυγχάνεται με την συνάρτηση `app.listen("port")` όπου στο port τοποθετούμε έναν αριθμό (πχ 8000).

Για να τρέξει το αρχείο θα πρέπει από το τερματικό να πλοηγηθούμε στον φάκελο και να τρέξουμε την εντολή: `run npm dev`

4. ANGULAR ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑ

Η angular είναι ένα web framework με κύρια γλώσσα προγραμματισμού την TypeScript.

Κάθε component αποτελείται από html - css και typescript αρχεία. Χρησιμοποιήσαμε 4 components για την δημιουργία των home page, bar chart , line chart και scatter plot.

Για κάθε ξεχωριστό component στο αρχείο .html υλοποιήσαμε έναν απλό stepper. Σε κάθε βήμα και μετά από επιλογές που έχει κάνει ο χρήστης, εκτελούνται από το αρχείο .ts κάποιες συναρτήσεις. Η κάθε μία συνάρτηση έχει και μία ερώτηση προς το API.

Η σύνδεση μεταξύ του backend με το frontend υλοποιείται στο αρχείο `data.service.ts` μέσω του `HttpClient`. Η κλάση περιέχει κάποιες συναρτήσεις όπου για ορίσματα υπάρχουν οι επιλογές του χρήστη μέσω του stepper. Έχοντας το URL που οδηγεί στο API προσθέτουμε τις κατάλληλες τιμές ώστε να γίνει η ερώτηση. Κάθε συνάρτηση επιστρέφει έναν πίνακα από Interfaces που έχουμε δημιουργήσει στο αρχείο `data.ts`

Για να ξεκινήσει ο server από το τερματικό θα πρέπει ο χρήστης να εκτελέσει την εντολή `ng server -o` ώστε να γίνει το compile και να ανοίξει ο web browser.

5. D3.JS : DATA VISUALIZATION

Για την απεικόνιση των γραφημάτων κάνουμε χρήση της βιβλιοθήκης d3.js.

Μέσω των συναρτήσεων που παρέχει η βιβλιοθήκη, μπορούμε να προσθέσουμε html κώδικα. Έτσι για την δημιουργία γραφημάτων προσθέτουμε ένα svg element όπου μπορούμε να προσθέσουμε σχήματα.

Η d3 μπορεί να διαχειριστεί δεδομένα είτε σε μορφή JSON είτε σε μορφή πίνακα και για κάθε πλειάδα να δημιουργήσει ένα σχήμα.

Για το bar chart και το line chart επειδή ο χρήστης έχει την δυνατότητα να επιλέξει πολλές χώρες και indicators, κάναμε χρήση της d3.nest το οποίο ομαδοποιεί σωστά τα δεδομένα. Στον άξονα X απεικονίζονται οι χρονιές ενώ στον άξονα Y οι μετρήσεις. Αντιθέτως στο scatterplot επειδή ο χρήστης μπορεί να επιλέξει μόνο μία χώρα και δύο indicators, ο κάθε άξονας απεικονίζει τις μετρήσεις ενός indicator.