

ΠΡΟΧΩΡΗΜΕΝΑ ΘΕΜΑΤΑ
ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΕΦΑΡΜΟΓΩΝ
ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ
ΤΟ ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2019-2020

ΓΚΑΡΤΖΙΟΣ ΧΡΙΣΤΟΣ , 2666`

ΜΠΑΡΖΑΣ ΓΙΑΝΝΗΣ , 2765

ΜΑΚΡΗΣ ΓΡΗΓΟΡΙΟΣ , 3022

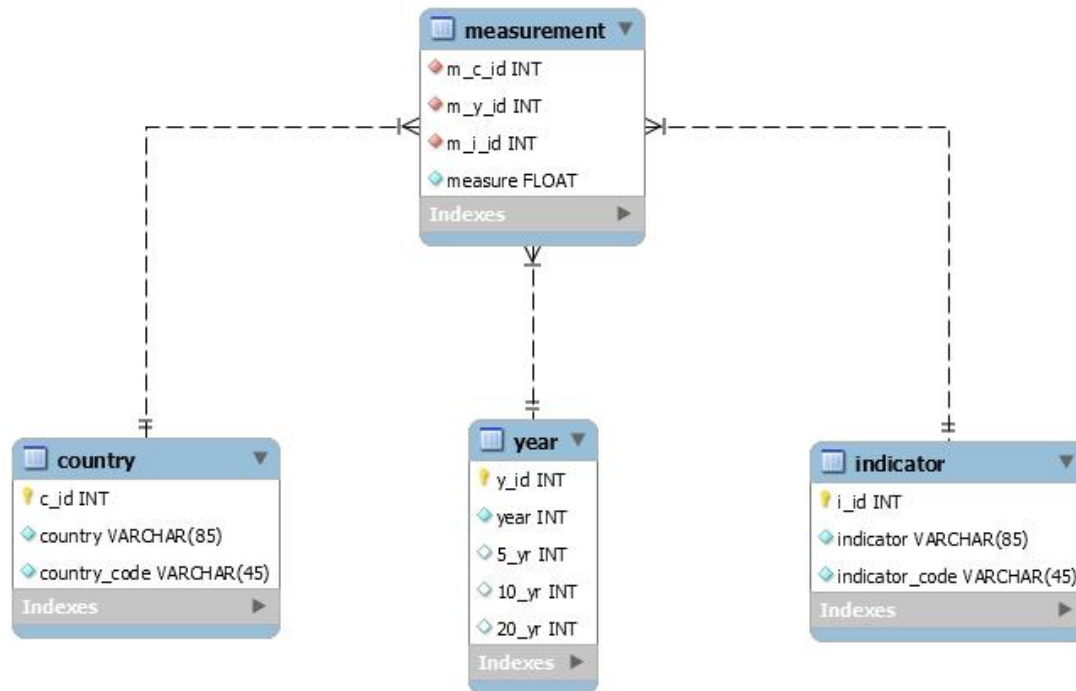
ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΙΟΥΝΙΟΣ 2020

1 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

1.1 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΛΟΓΙΚΟ ΕΠΙΠΕΔΟ

Αρχικά βλέπουμε στο σχεσιακό σχήμα της βάσης δεδομένων όπως αυτό προκύπτει μέσω εργαλείων της MySQL Workbench.



Παρακάτω ακολουθούν η αναλυτική λίστα των εντολών κατασκευής πινάκων.

```

CREATE SCHEMA `gdb_schema` ;

CREATE TABLE `gdb_schema`.`year` (
  `y_id` INT NOT NULL,
  `year` INT NOT NULL,
  `5yr` VARCHAR(45) NULL,
  `10yr` VARCHAR(45) NULL,
  `20yr` VARCHAR(45) NULL,
  PRIMARY KEY (`y_id`));

CREATE TABLE `gdb_schema`.`country` (
  `c_id` INT NOT NULL,
  `country` VARCHAR(45) NOT NULL,
  `c_code` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`c_id`));

CREATE TABLE `gdb_schema`.`indicator` (
  `i_id` INT NOT NULL,
  `indicator_name` VARCHAR(45) NOT NULL,
  `i_code` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`i_id`));
  
```

```
CREATE TABLE `gdb_schema`.`measurement` (  
  `c_id` INT NOT NULL,  
  `measurementcol` VARCHAR(45) NULL,  
  INDEX `c_id_idx` (`c_id` ASC) VISIBLE,  
  CONSTRAINT `c_id`  
    FOREIGN KEY (`c_id`)  
      REFERENCES `gdb_schema`.`country` (`c_id`)  
      ON DELETE CASCADE  
      ON UPDATE CASCADE);  
  
ALTER TABLE `gdb_schema`.`measurement`  
ADD COLUMN `measurementcol1` VARCHAR(45) NULL AFTER `y_id`,  
CHANGE COLUMN `measurementcol` `y_id` INT NOT NULL ,  
ADD INDEX `y_id_idx` (`y_id` ASC) VISIBLE;  
;  
  
ALTER TABLE `gdb_schema`.`measurement`  
ADD CONSTRAINT `y_id`  
  FOREIGN KEY (`y_id`)  
    REFERENCES `gdb_schema`.`year` (`y_id`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE;  
  
ALTER TABLE `gdb_schema`.`measurement`  
ADD COLUMN `measurementcol1` VARCHAR(45) NULL AFTER `y_id`,  
CHANGE COLUMN `measurementcol` `y_id` INT NOT NULL ,  
ADD INDEX `y_id_idx` (`y_id` ASC) VISIBLE;  
;  
  
ALTER TABLE `gdb_schema`.`measurement`  
ADD CONSTRAINT `y_id`  
  FOREIGN KEY (`y_id`)  
    REFERENCES `gdb_schema`.`year` (`y_id`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE;  
ALTER TABLE `gdb_schema`.`measurement`  
CHANGE COLUMN `measurementcol1` `i_id` INT NOT NULL ,  
ADD INDEX `i_id_idx` (`i_id` ASC) VISIBLE;  
;  
  
ALTER TABLE `gdb_schema`.`measurement`  
ADD CONSTRAINT `i_id`  
  FOREIGN KEY (`i_id`)  
    REFERENCES `gdb_schema`.`indicator` (`i_id`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE;  
ALTER TABLE `gdb_schema`.`measurement`  
ADD PRIMARY KEY (`c_id`, `y_id`, `i_id`);  
;
```

1.2 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΦΥΣΙΚΟ ΕΠΙΠΕΔΟ

1.2.1 ΡΥΘΜΙΣΗ ΤΩΝ ΠΑΡΑΜΕΤΡΩΝ ΤΟΥ DBMS

Οι ρυθμίσεις των παραμέτρων που κάναμε στο DBMS μας είναι οι εξής:

- `set global local_infile = 1;`
- `secure_file_priv=secure_file_priv = " "` στο αρχείο `my.ini`
- `innodb_buffer_pool_size = 3GB` στο αρχείο `my.ini`

Η πρώτη παράμετρος τροποποιήθηκε κατά τον τρόπο που φαίνεται παραπάνω ώστε η βάση να μην παραπονιέται κατά την φόρτωση μεγάλου όγκου αρχείων.

Στο αρχείο `my.ini` υπάρχει μια μεταβλητή `secure_file_priv=secure_file_priv` στην οποία είναι αποθηκευμένο το path από το οποίο μπορούν να φορτωθούν ασφαλή δεδομένα στην βάση. Για να μπορούμε λοιπόν να φορτώνουμε δεδομένα από όπου επιθυμούμε θέτουμε όπως φαίνεται παραπάνω το string αυτό σε κενό. Τέλος θέσαμε την μεταβλητή `innodb_buffer_pool_size` τόσο μεγάλη όσο χρειαζόνταν για τις ανάγκες του project.

1.2.2 ΡΥΘΜΙΣΗ ΤΟΥ ΦΥΣΙΚΟΥ ΣΧΗΜΑΤΟΣ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Κατά τη δημιουργία της βάσης ορίσαμε για κάθε πίνακα ένα ευρετήριο με βάση το primary key του. Όσων αφορά την χρήση των όψεων τις αποφύγαμε καθώς σύμφωνα με την δομή της βάσης μας δεν υπάρχει κάποια σπάνια ερώτηση πέραν των ερωτήσεων για τις διαθέσιμες χώρες, χρονολογίες (indicators) που ζητούνται πάντα για την φόρτωση του μενού, που όμως ακόμη και αυτά αναφέρονται σε μικρούς πίνακες και δεν αποτελούν ακριβώς ερωτήματα.

1.2.3 ΡΥΘΜΙΣΗ ΑΣΦΑΛΕΙΑΣ

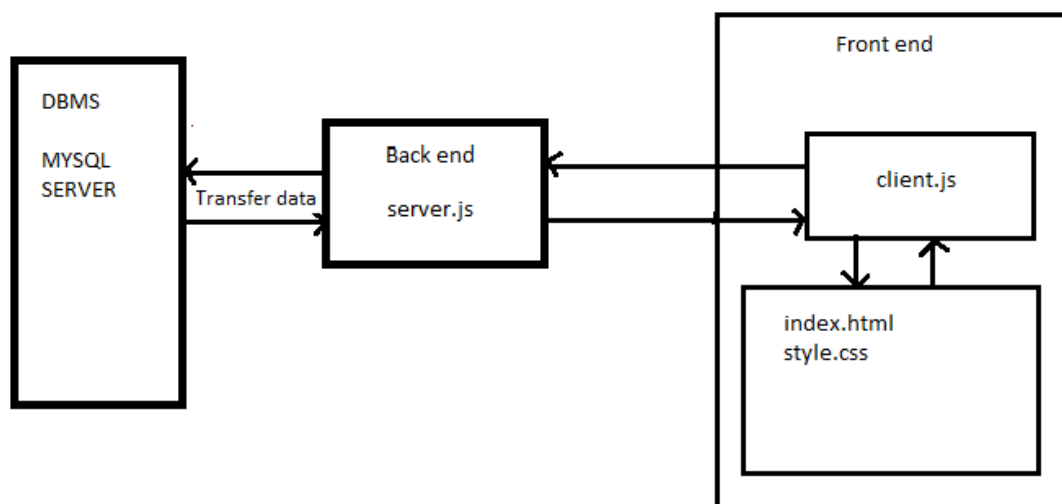
Σύμφωνα με την αρχιτεκτονική του web app ο χρήστης δεν μπορεί να κάνει επίθεση στη βάση καθώς οι μεταβλητές του ερωτήματος δεν εμφανίζονται στο path. Επιπλέον η δημιουργία των ερωτημάτων γίνεται αυτόματα από την εφαρμογή με την συλλογή των επιλογών του χρήστη από το menu επιλογών. Τέλος χωρίζουμε τα ερωτήματα σε 3 κομμάτια ένα `select` ένα για το `from` και ένα για το `where`, με αυτό τον τρόπο καταφέρνουμε να μην υπάρχει η δυνατότητα να παραποιηθεί το είδος του ερωτήματος.

2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΛΟΓΙΣΜΙΚΟΥ

2.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΔΟΜΗ ETL

Αρχικά, όπως αναφέρεται και στην εκφώνηση κατεβάσαμε τα δεδομένα που μας ενδιέφεραν από την σελίδα <https://data.worldbank.org/>. Συγκεκριμένα τα δεδομένα που επιλέχθηκαν αναφέρονται στα GDP(ΑΕΠ) 24 διαφορετικών χωρών καθώς και σε ορισμένα ποσοστά αξιοποίησης του GDP για κάλυψη διάφορων αναγκών (Current Health expenditure (% of GDP), Military expenditure (% of GDP), Government expenditure on education (% of GDP)). Στην συνέχεια επεξεργαστήκαμε κατάλληλα τα δεδομένα αυτά ώστε να ταιριάζουν στην δομή της βάσης δεδομένων μας. Η παραπάνω διαδικασία έγινε μέσω ενός script γραμμένου σε python. Τέλος για την φόρτωση των δεδομένων (στην πλέον κατάλληλη μορφή) στην βάση μας, αξιοποιήσαμε την built-in δυνατότητα που δίνεται μέσω του MySQL Workbench για φόρτωση δεδομένων μέσω Self-Contained Files.

2.2 ΔΙΑΓΡΑΜΜΑΤΑ ΠΑΚΕΤΩΝ / ΥΠΟΣΥΣΤΗΜΑΤΩΝ ΚΕΝΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ



3 ΥΠΟΔΕΙΓΜΑΤΑ ΕΡΩΤΗΣΕΩΝ ΚΑΙ ΑΠΑΝΤΗΣΕΩΝ

Η δομή του application χωρίζεται σε 3 μέρη, αυτό της Βάσης που περιέχει τη βάση και τον Server, το κομμάτι του Back end που περιέχει τα αρχεία nodejs και server.js, το οποίο μεταφέρει και λαμβάνει δεδομένα από και προς τη βάση και το τρίτο και τελευταίο κομμάτι του Front end όπου περιέχονται τα αρχεία client.js, index.html, style.css και άλλα αρχεία modules και packages.

Αρχικά κατά την εκκίνηση ο client κάνει ερωτήσεις στον server ο οποίος με την σειρά του κάνει στην βάση ώστε να λάβει τα διαθέσιμα χρόνια και χώρες (indicators) έτσι ώστε να φορτωθεί το μενού με τα κατάλληλα δεδομένα. Η φόρτωση του μενού γίνεται μέσω querySelectors.

Η επικοινωνία ανάμεσα στον client και τον server γίνεται μέσω Socket-io και χρήση κατάλληλων events. Όταν είναι έτοιμο το αίτημα του χρήστη λαμβάνεται στο event answer και μετασχηματίζεται η δομή του δυναμικά και εισάγεται στην λίστα List_combs έτσι ώστε τα δεδομένα να φορτωθούν κατάλληλα σε ένα από τα τρία διαθέσιμα d3 διαγράμματα ανάλογα την επιλογή του χρήστη. Η δομή της List_combs είναι μια γραμμή για κάθε χρόνια και μια τριάδα από(indicator,country,measure) που αντιπροσωπεύει τον συνδυασμό indicator,country. Η σύνθεση της SQL ερώτησης γίνεται χωρισμένη σε 2 περιπτώσεις αν είναι ομαδοποιημένη με βάση πενταετία, δεκαετία, εικοσαετία ή αν είναι ανά χρονιά.

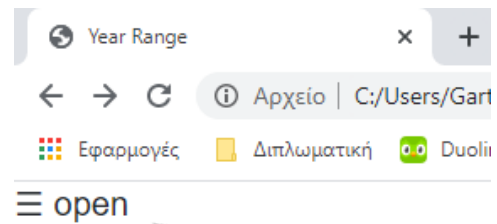
Για να εισάγεται με σωστό τρόπο κρατάμε σε ένα βοηθητικό πίνακα όλους τους συνδυασμούς indicator,country και δημιουργούμε τα κατάλληλα πεδία σε κάθε γραμμή χρησιμοποιώντας ένα μετρητή για να διαφοροποιείται το label.

Η αποστολή του αιτήματος γίνεται όταν πατηθεί το κουμπί show data και εφόσον έχουμε ελέγξει ότι όλα τα πεδία είναι συμπληρωμένα από τον χρήστη. Σε διαφορετική περίπτωση εμφανίζεται στον χρήστη ανάλογο μήνυμα. Οι διαθέσιμες επιλογές στα d3 γραφήματα είναι: scatter plot, bar chart, timeline. Τέλος έχουμε προσθέσει στο index.html στις επιλογές του μενού ανάλογα με τις επιλογές του χρήστη να καλούνται και κατάλληλες συναρτήσεις που να περνάνε τις επιλογές του στον client.js.

Σύντομη παρουσίαση του περιβάλλοντος:

Κατά την εκκίνηση του web app ο χρήστης βλέπει στο πάνω αριστερά μέρος του browser του ένα κουμπί (open) με το οποίο ανοίγει το μενού όπως φαίνεται στην διπλανή εικόνα.

Πατώντας το κουμπί αυτό έχουμε πλέον την εικόνα που ακολουθεί παρακάτω και στην οποία φαίνονται τόσο οι επιλογές που δίνονται στον χρήστη για την διαμόρφωση του ερωτήματος του όσο και το πώς η κάθε επιλογή διαμορφώνει το SQL ερώτημα.



Year Range

Price: 1960 1960

choose countries

choose indicators

Please select a group way:

by year

by 5 years

by 10 years

by 20 years

Please select graph:

timeline

bar chart

scatter plot

Connection established.+..

show data!

Year FILTERING

WHERE

ORDER BY

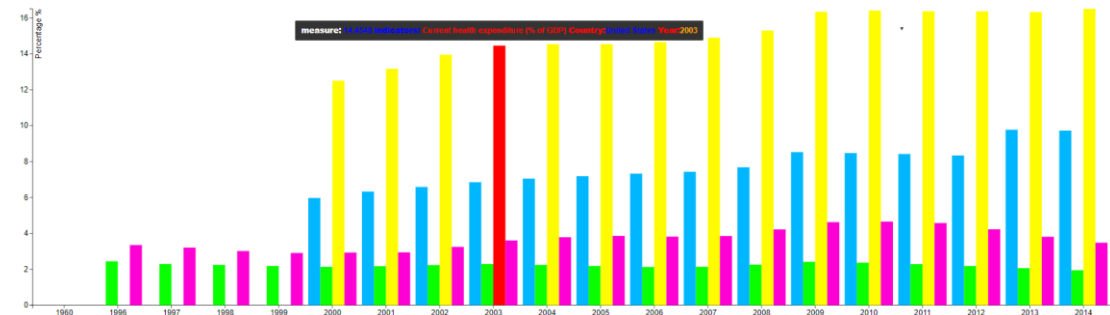
Way of Presentation

query COMMIT

Παρακάτω ακολουθούν κάποια παραδείγματα ερωτήσεων με τις απαντήσεις τους.

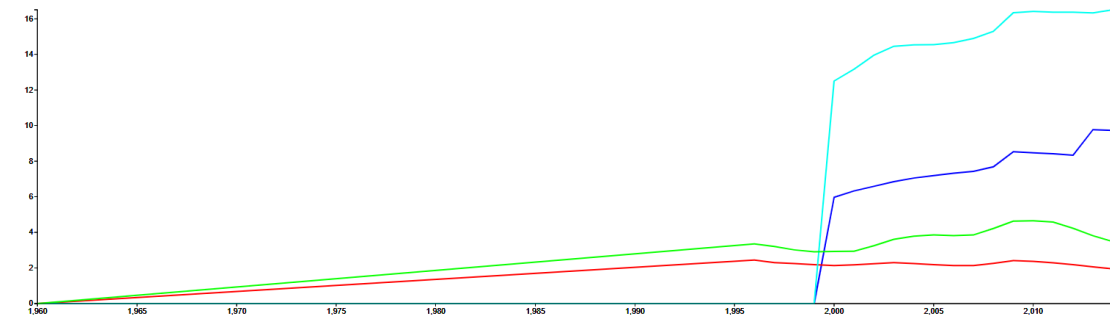
Παράδειγμα με Bar chart:

≡ open



Παράδειγμα με Timeline:

≡ open



Παράδειγμα με Scatter Plot:

≡ open

