

ΠΡΟΧΩΡΗΜΕΝΑ ΘΕΜΑΤΑ  
ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΕΦΑΡΜΟΓΩΝ  
ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

---

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ  
ΤΟ ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2020-2021

---

---

ΟΜΑΔΑ SER

ΕΥΠΡΑΞΙΑ ΤΣΙΛΙΚΟΥΔΗ, ΑΜ 3353

ΡΟΔΟΠΗ ΚΩΣΤΕΛΗ, ΑΜ 3263

ΣΟΦΙΑ ΠΑΠΑΧΑΡΑΛΑΜΠΟΥΣ, ΑΜ 3312

---

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΜΑΪΟΣ 2021

## ΙΣΤΟΡΙΚΟ ΠΡΟΗΓΟΥΜΕΝΩΝ ΕΚΔΟΣΕΩΝ

Ημερομηνία	Έκδοση	Περιγραφή	Συγγραφέας
23/03/21	Φάση I	Αρχική οργάνωση. Εξαγωγή, μετασχηματισμός και φόρτωση δεδομένων (ETL).	SER team
26/04/21	Φάση II	Οπτικοποίηση σε χρονοδιάγραμμα με δυναμική δημιουργία ερωτήσεων. Περιβάλλον αλληλεπίδρασης για τις επιλογές του χρήστη.	SER team
23/05/21	Φάση III	Πλήρης υλοποίηση της εφαρμογής.	SER team

## 1 ΠΕΡΙΕΧΟΜΕΝΑ

---

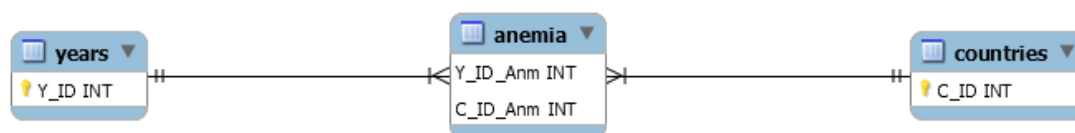
2	Βάση Δεδομένων.....	4
2.1	Σχεσιακό σχήμα σε λογικό επίπεδο.....	4
2.2	Σχεσιακό σχήμα σε φυσικό επίπεδο.....	6
2.2.1	Ρύθμιση των παραμέτρων του dbms.....	8
2.2.2	Ρύθμιση του φυσικού σχήματος της βάσης δεδομένων.....	9
2.2.3	Ρύθμιση ασφάλειας.....	9
3	Αρχιτεκτονική Λογισμικού.....	11
3.1	Αρχιτεκτονική και δομή ETL.....	11
3.2	Πακέτα και υποσυστήματα κεντρικής εφαρμογής.....	12
4	Υποδείγματα ερωτήσεων και απαντήσεων.....	14
5	Βιβλιογραφία.....	21

## 2 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Σ' αυτή την ενότητα περιγράφονται τα σχήματα της Βάσης Δεδομένων που χρησιμοποιούνται στο project.

### 2.1 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΛΟΓΙΚΟ ΕΠΙΠΕΔΟ

Το σχεσιακό σχήμα της Βάσης Δεδομένων σε λογικό επίπεδο περιλαμβάνει 15 οντότητες. Οι οντότητες "years" και "countries", αποτελούν τις βασικές οντότητες από τις οποίες εξαρτώνται όλες οι υπόλοιπες. Η σύνδεση των οντοτήτων με τις βασικές οντότητες ακολουθεί τον τύπο συσχέτισης ένα προς πολλά (1-N), δηλαδή μία οντότητα αναφέρεται σε πολλές χρονολογίες και πολλές χώρες. Στο παρακάτω σχεσιακό διάγραμμα (Σχήμα 1) αναπαρίσταται η συσχέτιση μεταξύ μίας οντότητας της Βάσης Δεδομένων με τις δύο βασικές οντότητες.



Σχήμα 1: Σχεσιακό Διάγραμμα σε Λογικό Επίπεδο

Η λογική δομή της βάσης αποτελείται από τις διασυνδέσεις του πίνακα κάθε δείκτη με τους lookup πίνακες με τις χώρες και τις χρονολογίες. Αφού δημιουργήσαμε όλους τους πίνακες (lookup tables και indexes tables), ακολουθήσαμε τα παρακάτω βήματα στην εφαρμογή διαχείρισης βάσεων δεδομένων MySQL Workbench, προκειμένου να εξάγουμε το σχήμα της βάσης (αρχείο "DataLoad.sql").

Αρχικά πηγαίνουμε στο tab Database > Forward Engineer. Στο παράθυρο που εμφανίζεται, στις σελίδες Connection Options και Options επιλέγουμε Next. Στη σελίδα Select Objects επιλέγουμε "Export MYSQL Table Objects" και στη συνέχεια Next, Save To File, έπειτα Next και Close.

Παρακάτω παρατίθενται οι εντολές κατασκευής πινάκων (Πίνακας 1) που βρίσκουμε στο αρχείο "Dataload.sql", το οποίο περιλαμβάνει το εξαγόμενο σχήμα της βάσης.

### ΕΠΕΞΗΓΗΣΗ ΕΝΤΟΛΩΝ ΚΑΤΑΣΚΕΥΗΣ ΠΙΝΑΚΩΝ

ΕΝΤΟΛΗ	ΕΠΕΞΗΓΗΣΗ
<b>CREATE SCHEMA IF NOT EXISTS 'database_name'</b>	Δημιουργία της βάσης "database_name" αν αυτή δεν υπάρχει.
<b>CREATE TABLE IF NOT EXISTS 'database_name'.`table_name`</b>	Δημιουργία του πίνακα "table_name" στην βάση "database_name" αν αυτός δεν υπάρχει.

<b>ID` INT NOT NULL</b>	Δηλώνεται ο τύπος του πεδίου ID ως ακέραιος με περιορισμό Not Null (απαγορεύονται οι κενές εγγραφές).
<b>PRIMARY KEY (ID`)</b>	Το πεδίο ID τίθεται ως πρωτεύον κλειδί.
<b>CONSTRAINT `FK_ID` FOREIGN KEY (ID_Idx`) REFERENCES `lookup_table` (ID`) ON DELETE CASCADE ON UPDATE CASCADE</b>	Καθορισμός ξένου κλειδιού με αναφορά στον lookup πίνακα. Σε περίπτωση διαγραφής εγγραφών του γονικού πίνακα, διαγράφονται και οι αντίστοιχες εγγραφές από τον θυγατρικό πίνακα.
<b>CREATE INDEX `FK_ID` ON `database_name`.`table_name` (ID_Idx` ASC);</b>	Οι εγγραφές του πίνακα "table_name" ταξινομούνται κατά αύξουσα σειρά, σύμφωνα με το πεδίο "Idx_ID". Με αυτόν τον τρόπο γίνεται πιο εύκολη η ανάκτηση εγγραφών.
<b>ENGINE = InnoDB;</b>	Η ρύθμιση του Engine σε InnoDB εξασφαλίζει τον τρόπο αποθήκευσης του πίνακα στον δίσκο. Σε περίπτωση που αναδημιουργήσουμε τον πίνακα (με Create Table) σε άλλο MySQL Server, με άγνωστη προκαθορισμένη ρύθμιση αποθήκευσης, ορίζεται ο InnoDB τύπος αποθήκευσης.

Πίνακας 1: Επεξήγηση Εντολών Κατασκευής Πινάκων

Παρακάτω παρατίθεται ένα παράδειγμα οργάνωσης κατασκευής πινάκων:

```
CREATE TABLE IF NOT EXISTS `nutrdb`.`Years` (
  `Y_ID` INT NOT NULL,
  `Year_Descr` INT NOT NULL,
  `FiveYSpan` INT NULL,
  `Decade` INT NULL,
  PRIMARY KEY (`Y_ID`))
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `nutrdb`.`Hapiscore` (
  `WHR` FLOAT NULL,
  `Y_ID_HS` INT NOT NULL,
  `C_ID_HS` INT NOT NULL,
```

PRIMARY KEY (`Y\_ID\_HS`,`C\_ID\_HS`),

CONSTRAINT `HS\_FK\_Y\_ID` FOREIGN KEY (`Y\_ID\_HS`) REFERENCES `years` (`Y\_ID`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `HS\_FK\_C\_ID` FOREIGN KEY (`C\_ID\_HS`) REFERENCES `countries` (`C\_ID`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE = InnoDB;

CREATE INDEX `HS\_FK\_Y\_ID` ON `nutrdb`.`Hapiscore` (`Y\_ID\_HS` ASC) ;

CREATE INDEX `HS\_FK\_C\_ID` ON `nutrdb`.`Hapiscore` (`C\_ID\_HS` ASC) ;

Υπάρχουν πολλές σχεδιαστικές λύσεις για την οργάνωση της πληροφορίας προκειμένου να είναι ακριβής, συνεπής και να καθιστά εύκολες τις ερωτήσεις. Ένας από τους τρόπους σχεδίασης βάσεων δεδομένων είναι η δημιουργία ενός μόνο πίνακα που περιλαμβάνει τις πληροφορίες για όλα τα αρχεία. Μια ακόμα ιδέα θα ήταν να διατηρούμε ένα πίνακα για κάθε δείκτη ο οποίος αποτελείται από μια στήλη-κλειδί με την πληροφορία για τη χώρα καθώς και στήλες για τις διαφορετικές χρονολογίες. Οι παραπάνω λύσεις παρ' ότι καθιστούν εύκολες τις ερωτήσεις (και δεν απαιτούν πράξεις join), αποτελούν μη συντηρήσιμη δομή βάσης. Για παράδειγμα, σε περίπτωση που θέλουμε να εισάγουμε μια νέα χρονολογία στη βάση θα πρέπει να επέμβουμε στον κώδικα. Επιπλέον, υπάρχουν τύποι ερωτημάτων (π.χ. μέσος όρος ανά δεκαετία/πενταετία) για τους οποίους απαιτείται προεπεξεργασία των δεδομένων.

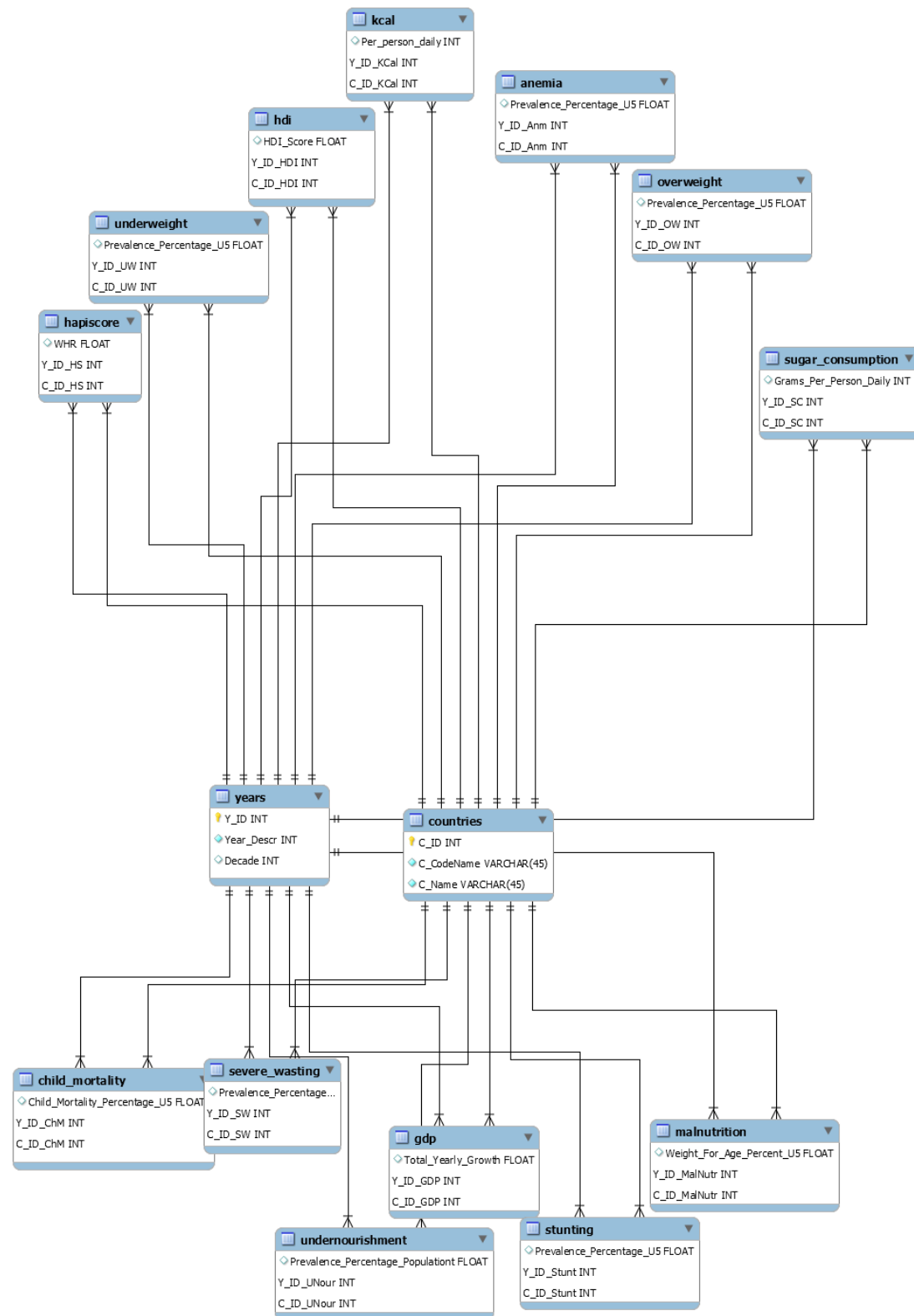
Η σχεδιαστική λύση που επιλέξαμε χρησιμοποιείται συχνότερα σε Βάσεις Δεδομένων διότι δεν απαιτείται συντήρηση κώδικα κατά την εισαγωγή/διαγραφή δεδομένων. Επιπλέον, παρ' ότι οι ερωτήσεις απαιτούν πράξεις join για να κατασκευαστούν, η Βάση Δεδομένων κατασκευάζει γρήγορα τις απαντήσεις. Συνεπώς, πραγματοποιούμε ελάχιστη προεπεξεργασία δεδομένων.

Επιλέξαμε την παραπάνω σχεδιαστική λύση για τους λόγους που διακρίνονται παρακάτω:

- Καλύτερη οργάνωση του κώδικα
- Υψηλή επεκτασιμότητα
- Συντηρήσιμος κώδικας
- Γρήγορη απάντηση σε queries, λιγότερη προεπεξεργασία δεδομένων
- Πρότυπο που χρησιμοποιείται συχνότερα σε βάσεις δεδομένων

## 2.2 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΦΥΣΙΚΟ ΕΠΙΠΕΔΟ

Στην ενότητα αυτή περιγράφουμε το σχεσιακό σχήμα της Βάσης Δεδομένων σε φυσικό επίπεδο. Στο Σχήμα 2 παρουσιάζεται η δομή των οντοτήτων, η οποία περιλαμβάνει για κάθε οντότητα τα χαρακτηριστικά της καθώς και τα πρωτεύοντα και δευτερεύοντα κλειδιά. Επιπλέον, διακρίνονται οι συσχετίσεις μεταξύ των οντοτήτων. Αναλυτικότερα, οι κύριες οντότητες "years" και "countries" έχουν μόνο πρωτεύοντα κλειδιά, ενώ οι υπόλοιπες οντότητες έχουν δευτερεύοντα κλειδιά στις βασικές οντότητες τα οποία συνιστούν το πρωτεύων κλειδί τους.



Σχήμα 2: Σχεσιακό Διάγραμμα σε Φυσικό Επίπεδο

### 2.2.1 ΡΥΘΜΙΣΗ ΤΩΝ ΠΑΡΑΜΕΤΡΩΝ ΤΟΥ DBMS

Για την αποθήκευση των πινάκων της Βάσης Δεδομένων, χρησιμοποιήσαμε ως μηχανή αποθήκευσης την InnoDB [1]. Η InnoDB αποτελεί μηχανή αποθήκευσης γενικού σκοπού που προσφέρει ταυτόχρονα υψηλή αξιοπιστία και απόδοση, καθώς και την προεπιλεγμένη μηχανή αποθήκευσης της MySQL. Τα κύρια πλεονεκτήματα που οδήγησαν στην επιλογή της παραπάνω μηχανής αποθήκευσης για τις δομές της Βάσης Δεδομένων μας είναι τα παρακάτω:

- Ακολουθεί το ACID μοντέλο. Η ατομικότητα, η συνέπεια, η απομόνωση και η μονιμότητα είναι ένα σύνολο ιδιοτήτων το οποίο εγγυάται ότι οι συναλλαγές στη βάση δεδομένων λειτουργούν αξιόπιστα.
- Οργανώνει τα δεδομένα στον δίσκο έτσι ώστε να βελτιστοποιήσουν τις ερωτήσεις με βάση τα πρωτεύοντα κλειδιά.
- Επιπλέον, υποστηρίζει δευτερεύοντα κλειδιά και ευρετήρια με χρήση δομών B-δέντρων.

Αφού στήσαμε και ρυθμίσαμε τη βάση δεδομένων μας, προχωρήσαμε στις παρακάτω ρυθμίσεις σε φυσικό επίπεδο στην μηχανή αποθήκευσης InnoDB.

Για μικρές βάσεις δεδομένων, οι οποίες χωρούν στην κύρια μνήμη ολόκληρες, είναι εφικτό να καθορίσουμε τους data buffers σε κάτι σαν 10% παραπάνω από το μέγεθος της βάσης. Βρήκαμε λοιπόν, το εν λόγω μέγεθος και ρυθμίσαμε το μέγεθος των data buffers ανάλογα με τη μνήμη του Η/Υ μας.

Χρησιμοποιώντας το παρακάτω query βρίσκουμε το μέγεθος της βάσης δεδομένων μας.

```
SELECT
    table_schema 'Database Name',
    SUM(data_length + index_length) 'Size in Bytes',
    ROUND(SUM(data_length + index_length) / 1024 / 1024, 2)
    'Size in MiB'
FROM information_schema.tables
GROUP BY table_schema;
```

Database Name	Size in Bytes	Size in MB
nutrdb	16138240	15.39

Έτσι, η παράμετρος innodb\_buffer\_pool\_size θα πάρει την τιμή: 15.39 MB + 0.01\*(15.39 MB) ≈ 16 MB = 16777216 Bytes.

Η αλλαγή της παραμέτρου γίνεται με την παρακάτω εντολή:

```
mysqld --innodb-buffer-pool-size=16777216 --innodb-buffer-pool-instances=1
--innodb-buffer-pool-chunk-size=16777216;
```

Εναλλακτικά, η αλλαγή της παραμέτρου μπορεί να γίνει με την εξής διαδικασία:

- Ανοίγουμε το αρχείο my.ini με δικαιώματα διαχειριστή
- Βρίσκουμε την παράμετρο innodb\_buffer\_pool\_size και αλλάζουμε την τιμή της
- Πηγαίνουμε στις «Υπηρεσίες»
- Βρίσκουμε τη MySQL και κάνουμε επανεκκίνηση



---

### 2.2.2 ΡΥΘΜΙΣΗ ΤΟΥ ΦΥΣΙΚΟΥ ΣΧΗΜΑΤΟΣ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

---

Τα ευρετήρια χρησιμοποιούνται έτσι ώστε να βρίσκουν γρήγορα γραμμές με συγκεκριμένες τιμές. Χωρίς τα ευρετήρια η αναζήτηση γίνεται με τη σειρά ξεκινώντας από την πρώτη γραμμή και στη συνέχεια διαβάζοντας όλο τον πίνακα μέχρι να βρεθεί σχετική πληροφορία. Όσο μεγαλύτερος είναι ο πίνακας, τόσο πιο χρονοβόρα γίνεται η παραπάνω διαδικασία. Ενώ, αν έχουμε κάποιο ευρετήριο η αναζήτηση γίνεται πιο εύκολα. Τα ευρετήρια που χρησιμοποιεί η mysql είναι το PRIMARY KEY, το UNIQUE INDEX και το FULLTEXT. Στον κώδικά μας χρησιμοποιούμε μόνο το PRIMARY KEY κυρίως για την περίπτωση που έχουμε WHERE συνθήκη. Τα πρωτεύοντα κλειδιά που ορίζουμε για κάθε πίνακα ξεχωριστά είναι το C\_ID (country id) και το Y\_ID (year id) για τους look up πίνακες ως έχει, αλλά και για τους υπόλοιπους πίνακες που τα κληρονομούν ως ξένα κλειδιά από τους look up πίνακες [2].

Οι όψεις (views) ή αλλιώς εικονικοί πίνακες (virtual views) δεν αποθηκεύουν δεδομένα αλλά ένα σύνολο αποτελεσμάτων που επιστρέφουν προκαθορισμένα ερωτήματα. Τις χρησιμοποιούμε συνήθως για ασφάλεια, για να κρύψουμε δεδομένα (στήλες) από τους χρήστες δίνοντας τους πρόσβαση σ' αυτές, αντί για τους ίδιους τους πίνακες. Επιπλέον, χρησιμοποιούνται στη κατασκευή απλοποιημένων, αφηρημένων διεπαφών, σε πιο πολύπλοκες βάσεις δεδομένων, αφού αποθηκεύουν αποτελέσματα που χρησιμοποιούνται συχνά από τη βάση. Στη βάση μας δεν δημιουργούμε όψεις, διότι δεν χρειάζεται να χρησιμοποιήσουμε αποτελέσματα από ένα συγκεκριμένο ερώτημα πολλαπλές φορές [3, 4].

---

### 2.2.3 ΡΥΘΜΙΣΗ ΑΣΦΑΛΕΙΑΣ

---

Στην ενότητα αυτή, παρουσιάζουμε τις ρυθμίσεις ασφαλείας που εφαρμόσαμε στην Βάση Δεδομένων. Ειδικότερα, περιγράφουμε τους χρήστες που δημιουργήσαμε για τη διαχείριση και την πρόσβαση στη βάση, καθώς και τη διαδικασία δημιουργίας αντιγράφων ασφαλείας.

Σε μια Βάση Δεδομένων μπορούν να χρησιμοποιηθούν πολλαπλοί λογαριασμοί χρηστών για μεγαλύτερη ασφάλεια. Σε κάθε λογαριασμό χρήστη μπορούν να ανατεθούν ρόλοι, οι οποίοι συσχετίζονται με διάφορα δικαιώματα πρόσβασης στη βάση [5]. Στη δική μας περίπτωση, έχουμε τον χρήστη διαχειριστή (root) με εξουσιοδότηση για όλες τις λειτουργίες πάνω στη βάση, καθώς και για τη διαχείριση των υπόλοιπων λογαριασμών. Επιπλέον, δημιουργήσαμε τρεις χρήστες:

1. Ένα χρήστη για τη διαχείριση της βάσης (manageruser), στον οποίο αναθέσαμε το ρόλο DBManager, και επιπλέον τα δικαιώματα REFERENCES, FILE.
2. Ένα χρήστη για τη δημιουργία των αντιγράφων ασφαλείας της βάσης (backupuser), στον οποίο αναθέσαμε το ρόλο BackupAdmin και το δικαίωμα FILE.
3. Και ένα χρήστη για την εκτέλεση ερωτήσεων στη βάση (appuser), στον οποίο αναθέσαμε μόνο το δικαίωμα SELECT.

Το backup ή αντίγραφο ασφαλείας αναφέρεται στην αντιγραφή φυσικών ή εικονικών αρχείων ή βάσεων δεδομένων σε δευτερεύουσα τοποθεσία για συντήρηση σε περίπτωση αποτυχίας εξοπλισμού ή άλλης καταστροφής. Η διαδικασία της δημιουργίας

αντιγράφων ασφαλείας δεδομένων είναι ζωτικής σημασίας καθώς μπορεί να επαναφέρει σημαντικές πληροφορίες της βάσης.

Η δημιουργία των backup αρχείων για κάθε ένα πίνακα της Βάσης Δεδομένων γίνεται με ένα script αρχείο ("create\_backups.py"). Αφού γίνει η σύνδεση στην βάση, η δημιουργία των backup δεδομένων γίνεται με χρήση της εντολής "SELECT \* INTO OUTFILE 'filename'". Επιλέξαμε τον παραπάνω τρόπο δημιουργίας των αρχείων backup, διότι πέρα από το γεγονός πως η βάση μας είναι μικρή και επομένως δεν τίθεται κάποιο ζήτημα όσον αφορά τον χρόνο δημιουργίας των αντιγράφων, μας επιτρέπει και να εξαγάγουμε αρχεία με συγκεκριμένη μορφή (χρησιμοποιώντας διαχωριστές στηλών και γραμμών. Ακόμη, το αρχείο δεν πρέπει να υπάρχει ήδη και δεν μπορεί να επεγγραφεί. Ο χρήστης που δημιουργεί τα backup αρχεία χρειάζεται να έχει τα δικαιώματα πρόσβασης στα αρχεία (FILE) και το ρόλο της διαχείρισης αντιγράφων ασφαλείας (BackupAdmin). Στη δική μας περίπτωση χρησιμοποιούμε τον χρήστη backupuser. Το αρχείο backup που θα δημιουργηθεί πρέπει να βρίσκεται στον ασφαλή φάκελο των μεταφορτώσεων αρχείων της MySQL. Ο φάκελός αυτός επιστρέφεται με την εντολή "show variables like 'secure file priv'" της MySQL.

### 3 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΛΟΓΙΣΜΙΚΟΥ

---

#### 3.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΔΟΜΗ ETL

---

Η σχεδίαση μίας Βάσης Δεδομένων βασίζεται στην οργάνωση των δεδομένων με βάση κάποιο μοντέλο σχεδίασης. Αρχικά, λαμβάνουμε υπόψιν την δομή των xlsx αρχείων. Ένα xlsx αρχείο αποτελείται από ένα δισδιάστατο πίνακα όπου αποθηκεύονται δεδομένα (αλφαριθμητικά και αριθμοί). Το σύνολο δεδομένων του Garpinder περιλαμβάνει πληθώρα δεδομένων με δείκτες που καλύπτουν ένα ευρύ θεματικό εύρος, από την οικονομία μέχρι την υγεία και την υποδομή [6].

Πιο συγκεκριμένα, κάθε αρχείο xlsx του Garpinder έχει μια συγκεκριμένη μορφή, όπου κάθε στήλη αναπαριστά μια χρονολογία και κάθε γραμμή αναπαριστά μια χώρα. Συνδυάζοντας τον δείκτη μιας γραμμής και μιας στήλης μπορούμε να αναφερθούμε στην τιμή μιας συγκεκριμένης χώρας για μια συγκεκριμένη χρονιά.

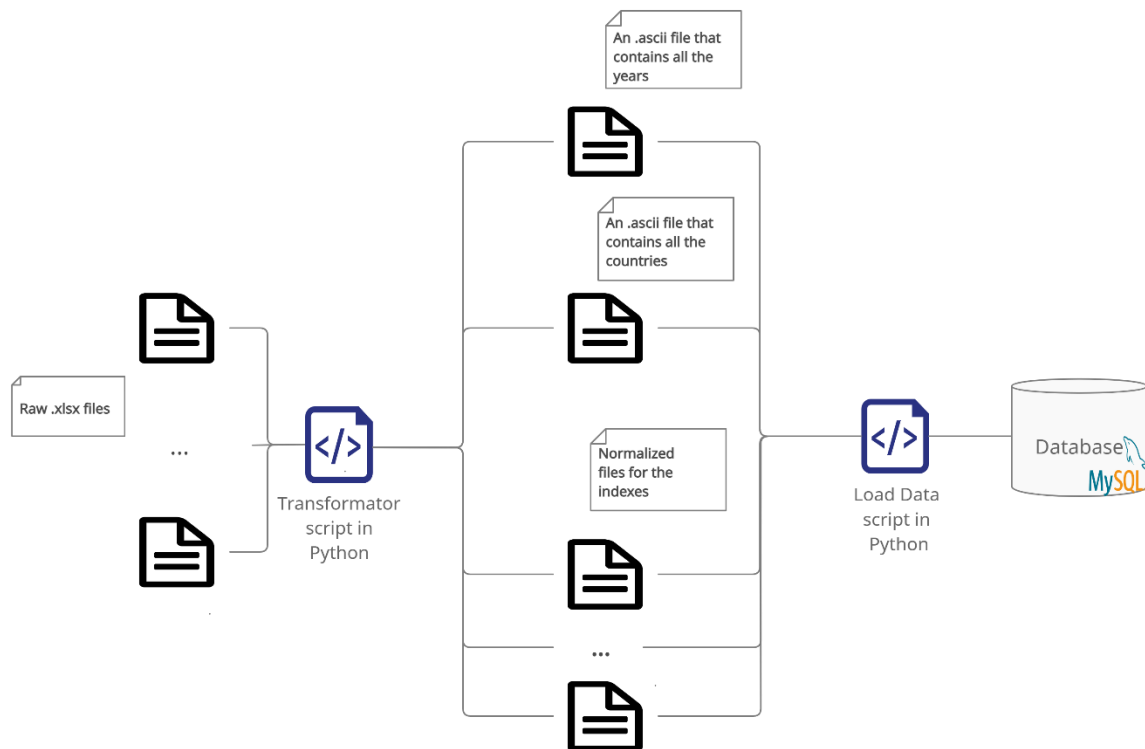
Στη δική μας βάση χρησιμοποιήσαμε δεδομένα που σχετίζονται με την υγεία και ειδικότερα με τη διατροφή για παιδιά ηλικίας κάτω των πέντε ετών. Κατά τη σχεδίαση του σχήματος της βάσης χρησιμοποιήσαμε έναν πίνακα για κάθε δείκτη και δύο lookup πίνακες για τις χρονολογίες και τις χώρες. Κατά την διαδικασία του μετασχηματισμού και μετά την προεπεξεργασία των δεδομένων των xlsx αρχείων, δημιουργούμε αρχεία ascii τόσο για τις χρονολογίες και τις χώρες, όσο και για τους δείκτες, όπως περιγράφεται παρακάτω. Για την υλοποίηση του μετασχηματισμού δημιουργήσαμε ένα αρχείο script ("transformation.py") με τη γλώσσα προγραμματισμού python

Στη συνέχεια, επεξεργαστήκαμε τα αρχεία δεδομένων xlsx με βάση τους δύο lookup πίνακες. Καταγράψαμε σε δύο σελίδες όλες τις χώρες και τις χρονολογίες που υπάρχουν σε όλα τα αρχεία του συνόλου δεδομένων. Έπειτα, αφού ταξινομήσαμε τα δύο σελίδες δημιουργήσαμε δύο λεξικά ένα με τις χρονολογίες και ένα με τα ονόματα των χωρών, ώστε να τα αντιστοιχίσουμε με ένα αναγνωριστικό. Ειδικότερα, στο αρχείο των χρονολογιών καταγράφεται το αναγνωριστικό, η χρονολογία και η δεκαετία στην οποία ανήκει, χωρισμένα με τον ειδικό χαρακτήρα "|". Με αντίστοιχο τρόπο, στο αρχείο των χωρών κρατάμε το αναγνωριστικό και το όνομα της χώρας. Επιπλέον, για το τελευταίο αρχείο, χρειάστηκε να μετατρέψουμε τα ονόματα σε μια κοινή μορφή αφαιρώντας από αυτά χαρακτήρες όπως κόμμα, τελεία, και απόστροφος. Ακολούθως, ο μετασχηματισμός των xlsx αρχείων για κάθε δείκτη γίνεται με βάση τα δύο λεξικά που δημιουργήσαμε για τις χρονολογίες και τις χώρες. Αρχικά, μετατρέπουμε τα δεδομένα σε ποσοστά σε όσα αρχεία δεικτών χρειάζεται, έτσι ώστε να έχουν όλα τα δεδομένα μας την ίδια μορφή. Στο αρχείο που δημιουργούμε για κάθε δείκτη, αποθηκεύουμε την τιμή του δείκτη, το αναγνωριστικό της χρονολογίας και της χώρας (με βάση τα λεξικά), που αντιστοιχούν στην δεδομένη τιμή χωρισμένα με τον ειδικό χαρακτήρα "|", όπως και προηγουμένως.

Μετά την διαδικασία του μετασχηματισμού, τα αρχεία είναι έτοιμα να φορτωθούν στην βάση. Αφού εγκαταστήσαμε την MySQL και την εφαρμογή MySQL Workbench, δημιουργήσαμε το σχήμα της βάσης στο Workbench, κατασκευάζοντας τους πίνακες για τους δείκτες, τις χρονολογίες και τις χώρες (lookup πίνακες) ορίζοντας τα πρωτεύοντα και δευτερεύοντα κλειδιά τους και τα χαρακτηριστικά τους στον τύπο δεδομένων που χρειάζονταν κάθε φορά. Στη συνέχεια, εξάγαμε το αρχείο "NutrLoadData.sql" με το Forward Engineer της εφαρμογής Workbench, το οποίο περιλαμβάνει όλες τις εντολές SQL για την δημιουργία του σχήματος. Το αρχείο αυτό

είναι χρήσιμο σε περίπτωση που χρειαστεί να δημιουργήσουμε την βάση από την αρχή. Το φόρτωμα των ascii αρχείων στη βάση γίνεται με ένα script αρχείο ("load\_data\_script.py"). Αφού γίνει η σύνδεση στην βάση, η φόρτωση των δεδομένων γίνεται με χρήση της εντολής "LOAD DATA INFILE".

Οι διαδικασίες που περιγράψαμε παραπάνω, παρουσιάζονται στο παρακάτω ETL διάγραμμα:

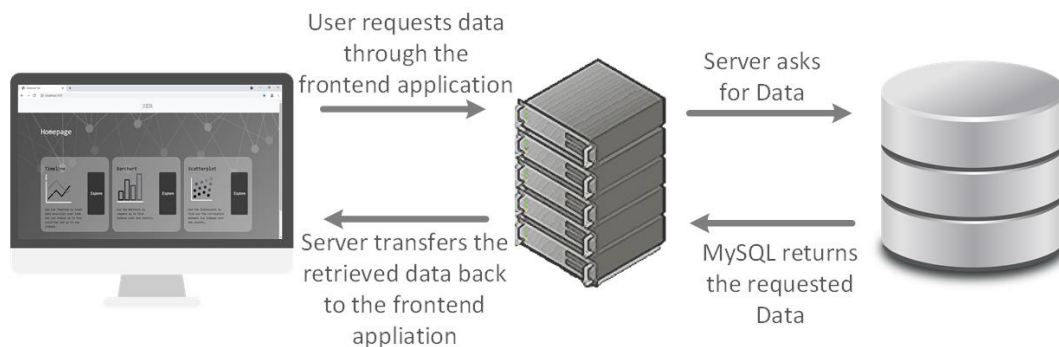


Σχήμα 3: ETL διάγραμμα

### 3.2 ΠΑΚΕΤΑ ΚΑΙ ΥΠΟΣΥΣΤΗΜΑΤΑ ΚΕΝΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ

Σ' αυτή την ενότητα περιγράφεται τόσο η επικοινωνία μεταξύ των υποσυστημάτων όσο και η high-level αρχιτεκτονική του συστήματος. Στο Σχήμα 4 παρουσιάζονται τα τμήματα από τα οποία αποτελείται το σύστημα, πιο συγκεκριμένα την εφαρμογή μας, τον εξυπηρετητή και την Βάση Δεδομένων, και τον τρόπο με τον οποίο επικοινωνούν μεταξύ τους. Η σειρά ενεργειών που ακολουθείτε κάθε φορά που αλληλεπιδρά ο χρήστης με την εφαρμογή είναι η παρακάτω:

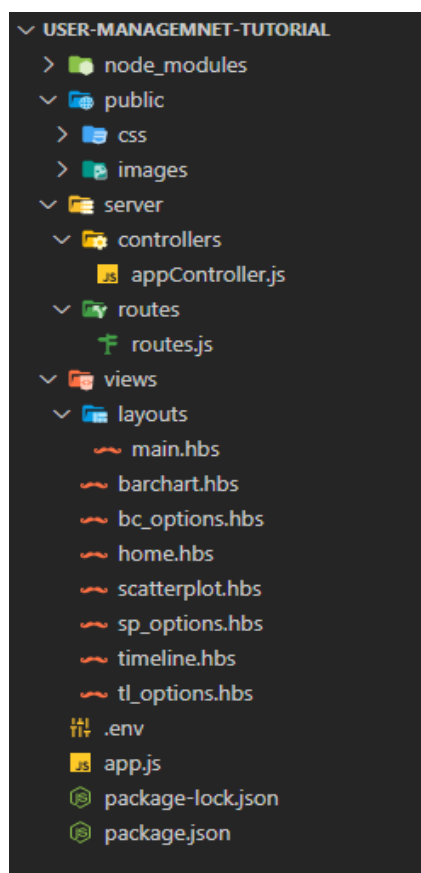
- Ο χρήστης κάνει κάποιο αίτημα μέσω τις εφαρμογής.
- Ο εξυπηρετητής ζητάει τα αιτούμενα δεδομένα από την Βάση Δεδομένων.
- Η βάση επιστρέφει τα δεδομένα που ζητήθηκαν μέσω των ερωτημάτων της MySQL.
- Τέλος, ο εξυπηρετητής μεταφέρει στην κεντρική εφαρμογή τα δεδομένα που ανέκτησε.



Σχήμα 4: Σύνδεση μεταξύ της εφαρμογής, του εξυπηρετητή και της βάσης δεδομένων

Το λογισμικό της εφαρμογής μας κατασκευάστηκε σε scripting γλώσσα Javascript με χρήση των βιβλιοθηκών express, doteenv, express-handlebars, body-parser, και mysql. Για την υλοποίηση των διαγραμμάτων με τα οποία οπτικοποιήσαμε τα αποτελέσματα κάθε ερώτησης, χρησιμοποιήσαμε την τεχνολογία d3 [7]. Τέλος, αξιοποιήθηκαν και τροποποιήθηκαν έτοιμα Notebooks από την πλατφόρμα Observable [8,9,10,11].

Το στιγμιότυπο με τα υποσυστήματα / πακέτα του λογισμικού της κεντρικής εφαρμογής επερώτησης παρατίθενται παρακάτω στο Σχήμα 5.



Σχήμα 5: Στιγμιότυπο με τα υποσυστήματα / πακέτα του λογισμικού της κεντρικής εφαρμογής επερώτησης

Αναλυτικότερα, τη βασική λειτουργία της εφαρμογής οργανώνει το αρχείο `app.js`. Συγκεκριμένα, σε αυτό το αρχείο εγκαθιδρύουμε την σύνδεση με τον Server μέσω της θύρας 3000 (<http://localhost:3000/>), ορίζουμε κατάλληλα τους συνδέσμους (routes), καθώς και τη διαδρομή για τα static files, στα οποία αποθηκεύουμε εικόνες και CSS αρχεία [12].

Στο αρχείο `route.js` γίνεται η σύνδεση μεταξύ των αιτημάτων GET και POST με τους αντίστοιχους χειριστές που βρίσκονται στο αρχείο `appController.js`. Στο δεύτερο, εκτελούνται οι βασικές λειτουργίες της εφαρμογής, συγκεκριμένα, η σύνδεση, και επομένως η επικοινωνία με την Βάση Δεδομένων, καθώς και άλλες λειτουργίες που εξασφαλίζουν την βέλτιστη απόκριση του συστήματος.

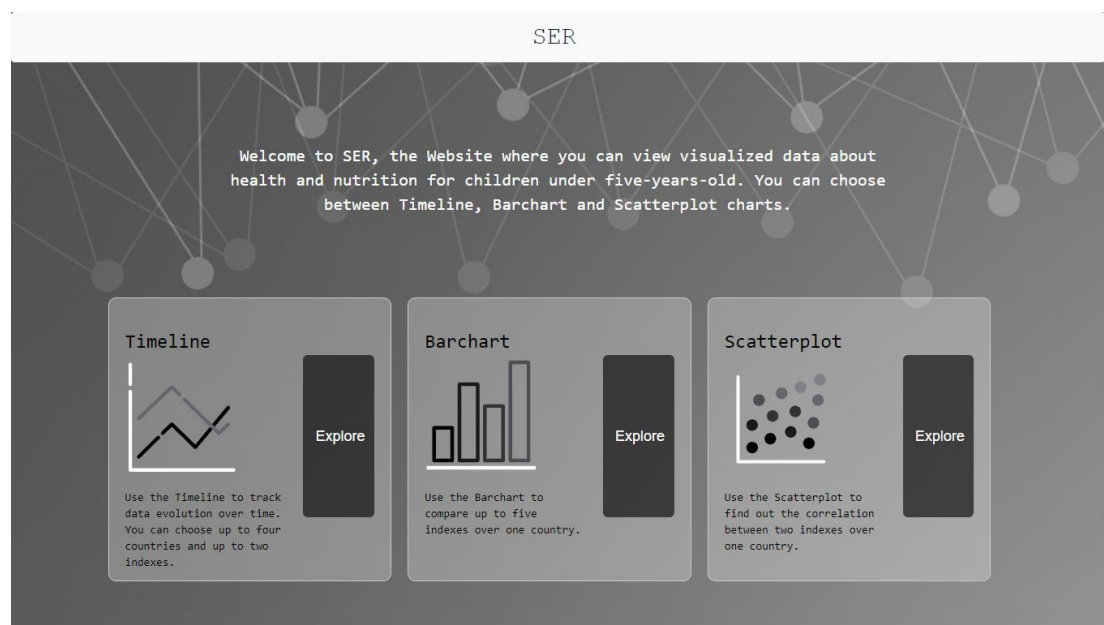
Για την οπτικοποίηση της εφαρμογής, χρησιμοποιήθηκαν `handlebars` αρχεία (`.hbs`) έτσι ώστε οι σελίδες HTML να κατασκευαστούν δυναμικά. Συγκεκριμένα, κάθε σελίδα έχει την ίδια βασική δομή που περιγράφεται από το αρχείο `main.hbs`, ωστόσο το περιεχόμενό της διαμορφώνεται ανάλογα με το `.hbs` αρχείο που τη τροφοδοτεί.

## 4 ΥΠΟΔΕΙΓΜΑΤΑ ΕΡΩΤΗΣΕΩΝ ΚΑΙ ΑΠΑΝΤΗΣΕΩΝ

Στην παρακάτω ενότητα παρουσιάζεται ένα σύντομος οδηγός χρήσης της εφαρμογής μας.

### Αρχική σελίδα

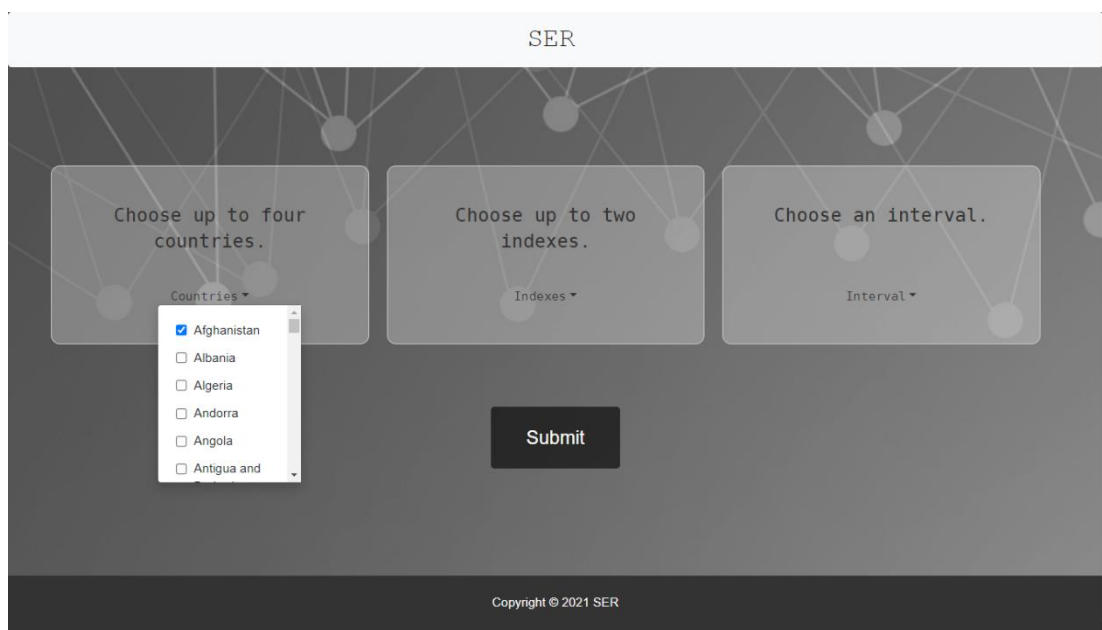
Η αρχική σελίδα της εφαρμογής μας (Σχήμα 6) δίνει τη δυνατότητα στον χρήστη να επιλέξει μεταξύ των τριών τρόπων οπτικοποίησης: Timeline για την απεικόνιση των δεδομένων στον χρόνο, Barchart για σύγκριση διαφόρων δεικτών για μία χώρα, και Scatterplot για ανίχνευση συσχετίσεων μεταξύ δύο διαφορετικών δεικτών για μία χώρα. Με το κουμπί Explore ο χρήστης μεταβαίνει στην σελίδα για την επιλογή ερωτήσεων. Πατώντας το κουμπί του λογότυπου, ο χρήστης μπορεί να μεταβεί στην αρχική σελίδα από οποιαδήποτε άλλη σελίδα βρίσκεται.



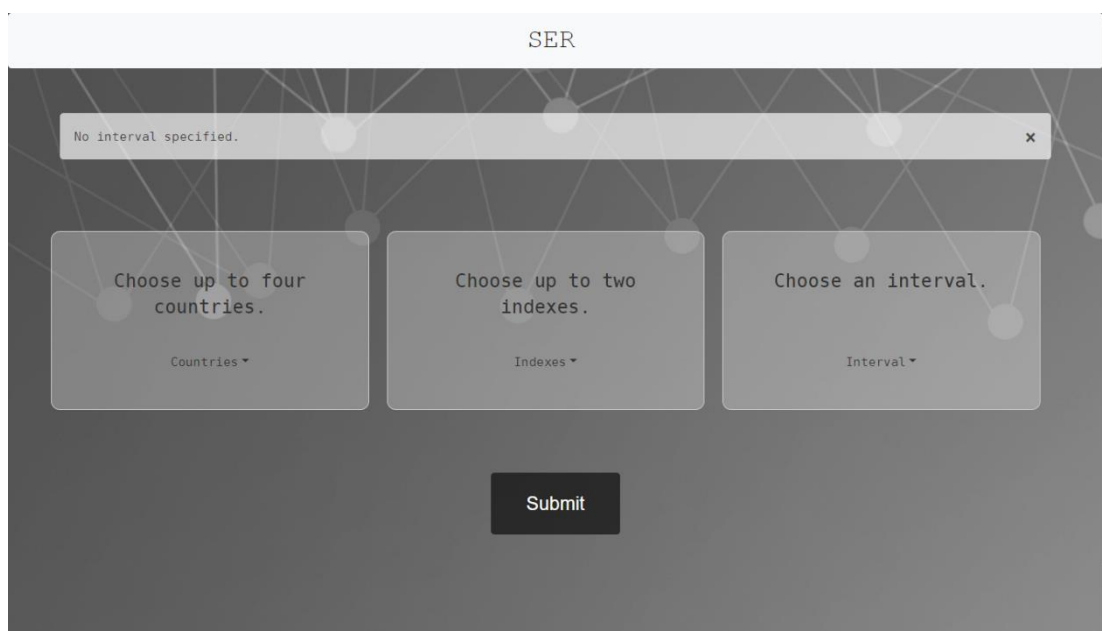
Σχήμα 6: Αρχική σελίδα εφαρμογής

### Επιλογή Timeline

Στην σελίδα timeline-options, ο χρήστης έχει τη δυνατότητα να κάνει μία ερώτηση διαλέγοντας μία έως τέσσερις χώρες, μέχρι δύο δείκτες και το μεσοδιάστημα που επιθυμεί. Η επιλογή για κάθε μία από τις παραπάνω επιλογές γίνεται πατώντας πάνω στο αντίστοιχο κουμπί, όπου ανοίγει ένα κουτί επιλογών όπως φαίνεται στο Σχήμα 7. Σε περίπτωση που ο χρήστης δεν επιλέξει κάποιο από τα απαιτούμενα πεδία για την δημιουργία του χρονοδιαγράμματος, εμφανίζεται ένα μήνυμα προειδοποίησης, όπως φαίνεται στο Σχήμα 8. Στη συνέχεια, μετά τη συμπλήρωση και των τριών πεδίων χρησιμοποιώντας το κουμπί Submit ο χρήστης μεταβαίνει στη σελίδα του Timeline.



Σχήμα 7: Σελίδα Timeline-options

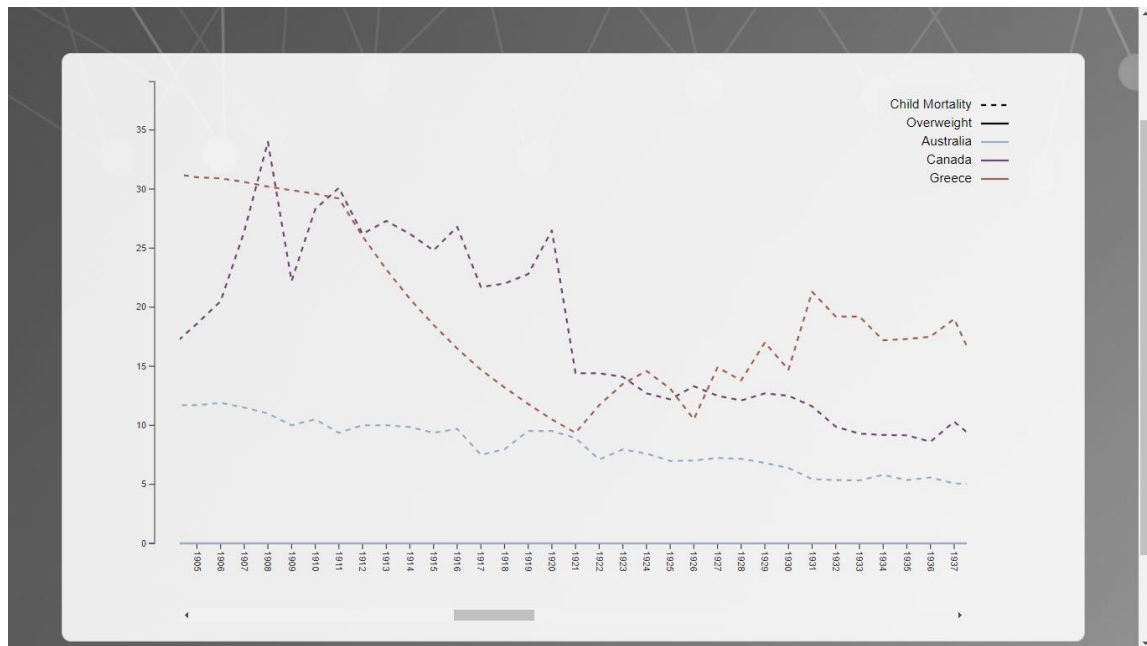


Σχήμα 8: Προειδοποιητικό μήνυμα μη συμπλήρωσης πεδίου



## Timeline

Στη σελίδα Timeline παρουσιάζεται το χρονοδιάγραμμα με τις επιλογές του χρήστη, οι οποίες εμφανίζονται σε υπόμνημα πάνω δεξιά στο τέλος του διαγράμματος. Πιο συγκεκριμένα, χρησιμοποιούνται διαφορετικές υφές για κάθε δείκτη και διαφορετικά χρώματα για κάθε χώρα. Ο χρήστης μπορεί να μετακινηθεί στο διάγραμμα με την χρήση μιας μπάρας .



Σχήμα 9: Σελίδα Timeline

Η ερώτηση που γίνεται στο backend για το παραπάνω χρονοδιάγραμμα είναι η παρακάτω:

```
SELECT years.Year_Descr as year, countries.C_Name as country,
child_mortality.Child_Mortality_Percentage_U5 as 'child_mortality',
overweight.Prevalence_Percentage_U5 as 'overweight'

FROM (countries, years)

LEFT JOIN child_mortality ON child_mortality.Y_ID_ChM = years.Y_ID
AND child_mortality.C_ID_ChM = countries.C_ID

LEFT JOIN overweight ON overweight.Y_ID_OW = years.Y_ID AND
overweight.C_ID_OW = countries.C_ID

WHERE ( countries.C_Name = 'Australia' OR countries.C_Name = 'Canada'
OR countries.C_Name = 'Greece' ) AND (
child_mortality.Child_Mortality_Percentage_U5 IS NOT NULL OR
overweight.Prevalence_Percentage_U5 IS NOT NULL )

ORDER BY year;
```



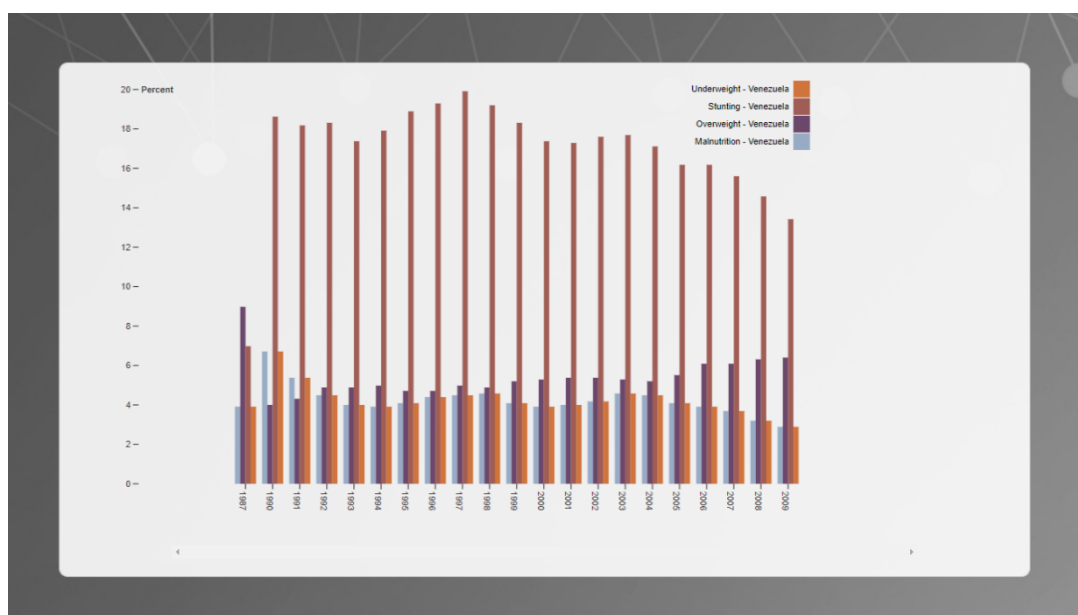
## Επιλογή Barchart

Στη σελίδα barchart-options ο χρήστης μπορεί να κάνει μία ερώτηση, επιλέγοντας μία μόνο χώρα, μέχρι πέντε δείκτες και ένα μεσοδιάστημα (Σχήμα 10). Σε περίπτωση, που δεν συμπληρωθεί κάποιο από τα απαιτούμενα πεδία για την δημιουργία ερώτησης, εμφανίζεται το κατάλληλο προειδοποιητικό μήνυμα. Επιπλέον, πατώντας το κουμπί Submit, ο χρήστης μεταβαίνει στην επόμενη σελίδα με το barchart.

Σχήμα 10: Σελίδα Barchart-options

## Barchart

Στη σελίδα barchart παρουσιάζεται το διάγραμμα σε μπάρες με τις επιλογές του χρήστη, οι οποίες εμφανίζονται σε υπόμνημα πάνω δεξιά στο τέλος του διαγράμματος. Πιο συγκεκριμένα, χρησιμοποιούνται διαφορετικά χρώματα για κάθε δείκτη.



*Σχήμα 11: Σελίδα Barchart*

Η ερώτηση που γίνεται στο backend για το παραπάνω χρονοδιάγραμμα είναι η παρακάτω:

```
SELECT years.Year_Descr as year, countries.C_Name as country,
malnutrition.Weight_For_Age_Percent_U5 as 'malnutrition',
overweight.Prevalence_Percentage_U5 as 'overweight',
stunting.Prevalence_Percentage_U5 as 'stunting',
underweight.Prevalence_Percentage_U5 as 'underweight'

FROM (countries, years)

LEFT JOIN malnutrition ON malnutrition.Y_ID_MalNutr = years.Y_ID AND
malnutrition.C_ID_MalNutr = countries.C_ID

LEFT JOIN overweight ON overweight.Y_ID_OW = years.Y_ID AND
overweight.C_ID_OW = countries.C_ID

LEFT JOIN stunting ON stunting.Y_ID_Stunt = years.Y_ID AND
stunting.C_ID_Stunt = countries.C_ID

LEFT JOIN underweight ON underweight.Y_ID_UW = years.Y_ID AND
underweight.C_ID_UW = countries.C_ID

WHERE ( countries.C_CodeName = 'Venezuela' ) AND (
malnutrition.Weight_For_Age_Percent_U5 IS NOT NULL OR
overweight.Prevalence_Percentage_U5 IS NOT NULL OR
stunting.Prevalence_Percentage_U5 IS NOT NULL OR
underweight.Prevalence_Percentage_U5 IS NOT NULL )

ORDER BY year
```

**Επιλογή Scatterplot**

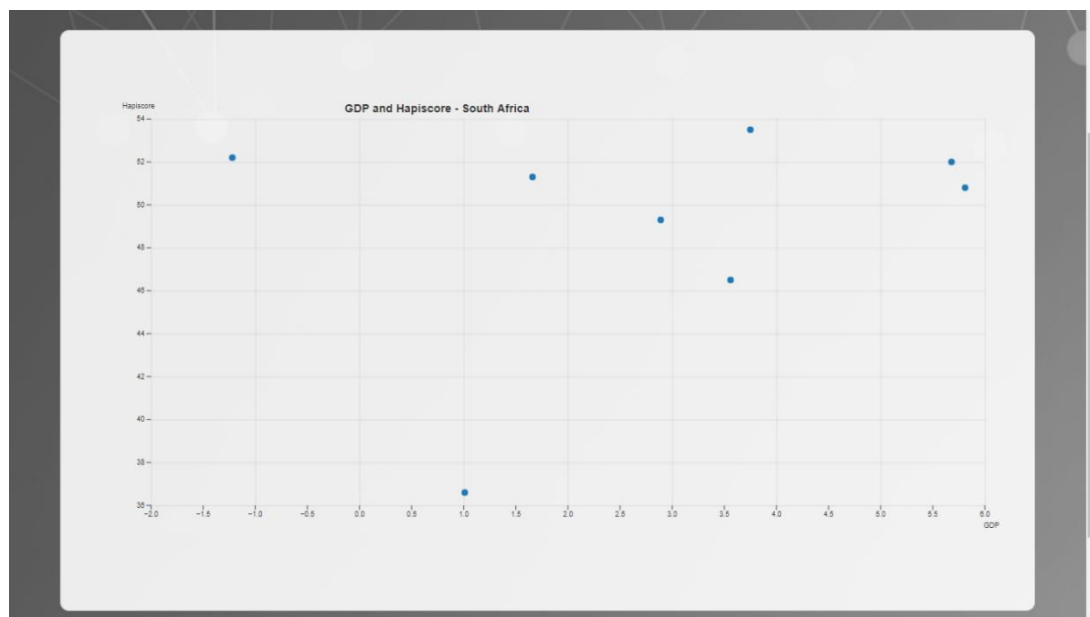
Στη σελίδα scatterplot-options ο χρήστης μπορεί να κάνει μία ερώτηση, επιλέγοντας μία μόνο χώρα, δύο δείκτες και ένα χρονικό μεσοδιάστημα (Σχήμα 12). Σε περίπτωση, που δεν συμπληρωθεί κάποιο από τα απαιτούμενα πεδία για την δημιουργία ερώτησης, εμφανίζεται το κατάλληλο προειδοποιητικό μήνυμα. Πατώντας το κουμπί Submit, ο χρήστης μεταβαίνει στην επόμενη σελίδα με το scatterplot.

The screenshot shows the SER web application interface. At the top, it says 'SER'. Below this, there are three main selection boxes: 'Choose a country.' with a 'Countries' dropdown, 'Choose two indexes.' with an 'Indexes' dropdown, and 'Choose an interval.' with an 'Interval' dropdown. The 'Interval' dropdown is open, showing three options: 'Yearly' (selected with a blue checkmark), 'Five-Year Span', and 'Ten-Year Span'. Below these boxes is a 'Submit' button. At the bottom, it says 'Copyright © 2021 SER'.

Σχήμα 12: Σελίδα Scatterplot-options

### Scatterplot

Στη σελίδα scatterplot παρουσιάζεται η συσχέτιση μεταξύ των δύο δεικτών που επέλεξε ο χρήστης, όπως φαίνεται στο Σχήμα 12.



Σχήμα 12: Σελίδα Scatterplot

Η ερώτηση που έγινε στο backend για το παραπάνω διάγραμμα είναι η παρακάτω:

```
SELECT years.Year_Descr as year, countries.C_Name as country,
gdp.Total_Yearly_Growth as 'GDP', hapiscore.WHR as 'Hapiscore'

FROM (countries, years)

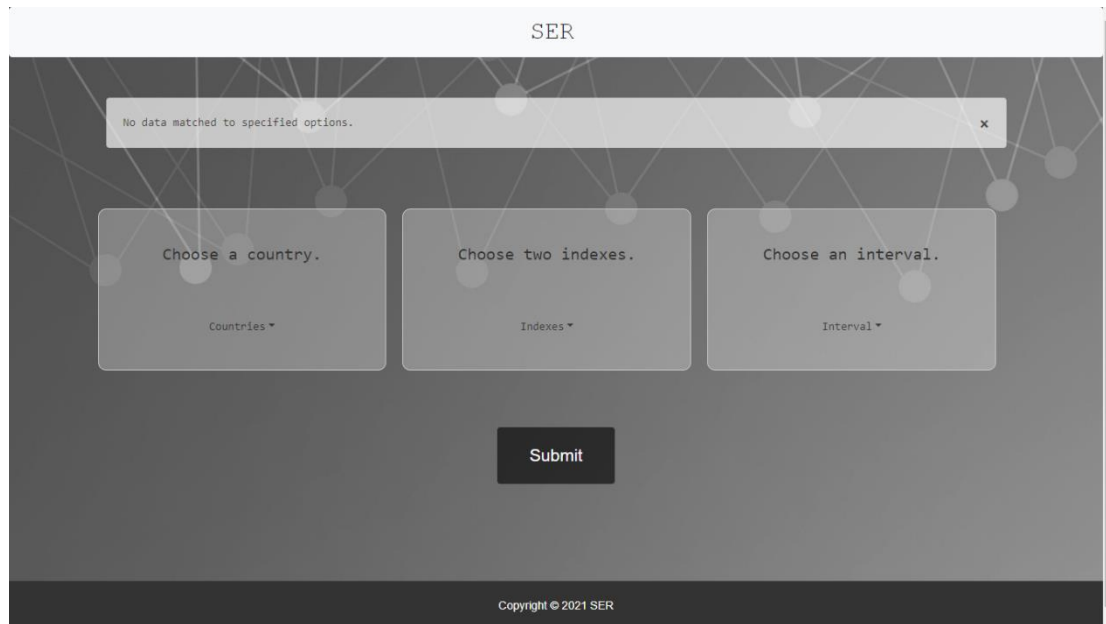
LEFT JOIN gdp ON gdp.Y_ID_GDP = years.Y_ID AND gdp.C_ID_GDP =
countries.C_ID
```

```
LEFT JOIN hapiscore ON hapiscore.Y_ID_HS = years.Y_ID AND
hapiscore.C_ID_HS = countries.C_ID

WHERE ( countries.C_CodeName = 'SouthAfrica' ) AND (
gdp.Total_Yearly_Growth IS NOT NULL AND hapiscore.WHR IS NOT NULL )

ORDER BY year
```

Επιπλέον, σε περίπτωση που δεν υπάρχουν δεδομένα που να απαντούν μία ερώτηση εμφανίζεται το παρακάτω προειδοποιητικό μήνυμα, όπως φαίνεται στο Σχήμα 13. Το ίδιο συμβαίνει σε οποιαδήποτε επιλογή οπτικοποίησης.



Σχήμα 13: Προειδοποιητικό μήνυμα για την μη ύπαρξη απάντησης σε μία ερώτηση

Μία τέτοια ερώτηση είναι η παρακάτω:

```
SELECT years.Year_Descr as year, countries.C_Name as country,
overweight.Prevalence_Percentage_U5 as 'Overweight',
stunting.Prevalence_Percentage_U5 as 'Stunting'

FROM (countries, years)

LEFT JOIN overweight ON overweight.Y_ID_OW = years.Y_ID AND
overweight.C_ID_OW = countries.C_ID

LEFT JOIN stunting ON stunting.Y_ID_Stunt = years.Y_ID AND
stunting.C_ID_Stunt = countries.C_ID

WHERE ( countries.C_CodeName = 'Bahamas' ) AND (
overweight.Prevalence_Percentage_U5 IS NOT NULL AND
stunting.Prevalence_Percentage_U5 IS NOT NULL )

ORDER BY year
```

## 5 ΒΙΒΛΙΟΓΡΑΦΙΑ

---

- [1] “MySQL Documentation”, <https://dev.mysql.com/doc/refman/8.0/en/innodb-storage-engine.html>
- [2] “How MySQL Uses Indexes”, <https://dev.mysql.com/doc/refman/8.0/en/mysql-indexes.html>
- [3] “Learn MySQL: The Basics of MySQL Views”, <https://www.sqlshack.com/learn-mysql-the-basics-of-mysql-views/>
- [4] “Using MySQL Views”, <https://www.a2hosting.com/kb/developer-corner/mysql/mysql-views>
- [5] “Using Roles”, <https://dev.mysql.com/doc/refman/8.0/en/roles.html>
- [6] “GAPMINDER”, <https://www.gapminder.org/>
- [7] “The D3.js Graph Gallery”, <https://www.d3-graph-gallery.com/>
- [8] “Observable”, <https://observablehq.com/>
- [9] “Directly labelling lines”, <https://observablehq.com/@harrystevens/directly-labelling-lines>
- [10] “Grouped Bar Chart”, <https://observablehq.com/@d3/grouped-bar-chart>
- [11] “Scatterplot with Shapes”, <https://observablehq.com/@d3/scatterplot-with-shapes>
- [12] “flaticon”, <https://www.flaticon.com/>