ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΡΟΧΩΡΗΜΕΝΑ ΘΕΜΑΤΑ ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΕΦΑΡΜΟΓΩΝ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ ΤΟ ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2021-2022

ΟΜΑΔΑ LABOOK

ΤΗΛΕΜΑΧΟΣ-ΜΑΡΚΟΣ ΜΠΑΖΑΚΑΣ, 3281

ΒΑΣΙΛΕΙΟΣ ΝΤΟΝΤΗΣ, 3300

ΖΗΣΗΣ-ΠΡΟΚΟΠΙΟΣ ΤΑΛΑΜΑΓΚΑΣ, 3340

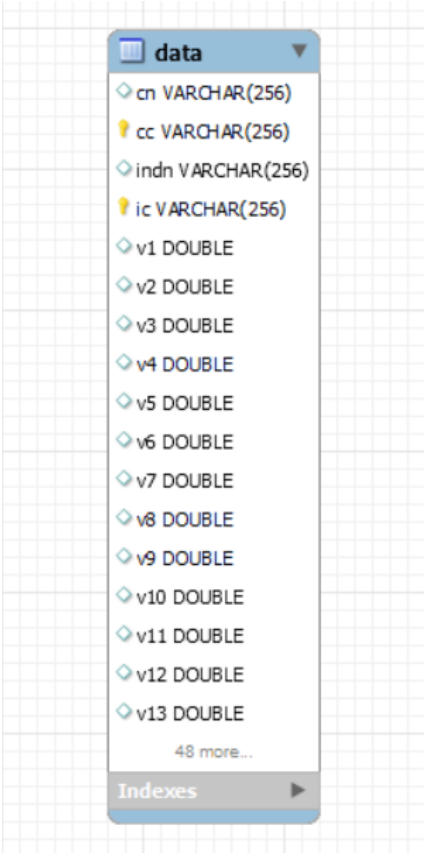
ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΜΑΪΟΣ 2022

1 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Στην παρούσα ενότητα περιγράφονται τα σχήματα της βάσης (ή βάσεων, αν είναι παραπάνω από μία) δεδομένων που χρησιμοποιούνται στο project.

1.1 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΛΟΓΙΚΟ ΕΠΙΠΕΔΟ



Σχήμα 1.1 Σχεσιακό σχήμα της βάσης δεδομένων του συστήματος

Η βάση δεδομένων μας αποτελείται από έναν πίνακα τον οποίο βλέπουμε παραπάνω. Σε αυτόν περιέχονται όλες οι στήλες του τελικού csv που καλούμαστε να διαχειριστούμε. Για παράδειγμα, συγκρίνοντας με την εικόνα 1.2 που δείχνει τα δεδομένα του csv, οι τιμές αντιστοιχούνται :

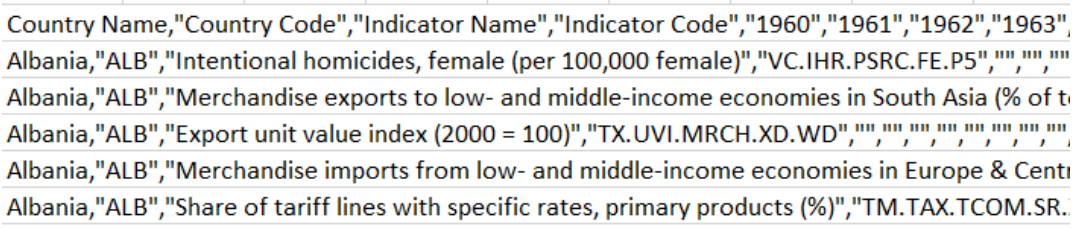
VARCHAR(256) cn: Country Name

VARCHAR(256) cc: Country Code

VARCHAR(256) indn: Indicator Name

VARCHAR(256) ic: Indicator Code

DOUBLE v1,v2,v3 .. : η τιμή το 1960,1961,1962 ...



Εικόνα 1.2

Το μοναδικό primary key που χρησιμοποιούμε σε αυτήν την βάση δεδομένων είναι ο συνδυασμός του Country Code και του Indicator Code, δηλαδή PRIMARY KEY (cc, ic)

```
-- MySQL Script generated by MySQL Workbench

-- Sat May  7 18:13:12 2022

-- Model: New Model    Version: 1.0

-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,
UNIQUE_CHECKS=0;

SET
@OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,N
O_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_Z
ERO,NO_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema mydata

-- -----------------------------------------------------

DROP SCHEMA IF EXISTS `mydata` ;

-- -----------------------------------------------------

-- Schema mydata

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `mydata` DEFAULT
CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci ;

USE `mydata` ;

-- -----------------------------------------------------

-- Table `mydata`.`data`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydata`.`data` ;
```

```
CREATE TABLE IF NOT EXISTS `mydata`.`data` (
  `cn` VARCHAR(256) NULL DEFAULT NULL,
  `cc` VARCHAR(256) NOT NULL,
  `indn` VARCHAR(256) NULL DEFAULT NULL,
  `ic` VARCHAR(256) NOT NULL,
  `v1` DOUBLE NULL DEFAULT NULL,
  `v2` DOUBLE NULL DEFAULT NULL,
  `v3` DOUBLE NULL DEFAULT NULL,
  `v4` DOUBLE NULL DEFAULT NULL,
  `v5` DOUBLE NULL DEFAULT NULL,
  `v6` DOUBLE NULL DEFAULT NULL,
  `v7` DOUBLE NULL DEFAULT NULL,
  `v8` DOUBLE NULL DEFAULT NULL,
  `v9` DOUBLE NULL DEFAULT NULL,
  `v10` DOUBLE NULL DEFAULT NULL,
  `v11` DOUBLE NULL DEFAULT NULL,
  `v12` DOUBLE NULL DEFAULT NULL,
  `v13` DOUBLE NULL DEFAULT NULL,
  `v14` DOUBLE NULL DEFAULT NULL,
  `v15` DOUBLE NULL DEFAULT NULL,
  `v16` DOUBLE NULL DEFAULT NULL,
  `v17` DOUBLE NULL DEFAULT NULL,
  `v18` DOUBLE NULL DEFAULT NULL,
  `v19` DOUBLE NULL DEFAULT NULL,
  `v20` DOUBLE NULL DEFAULT NULL,
  `v21` DOUBLE NULL DEFAULT NULL,
  `v22` DOUBLE NULL DEFAULT NULL,
  `v23` DOUBLE NULL DEFAULT NULL,
  `v24` DOUBLE NULL DEFAULT NULL,
  `v25` DOUBLE NULL DEFAULT NULL,
  `v26` DOUBLE NULL DEFAULT NULL,
  `v27` DOUBLE NULL DEFAULT NULL,
  `v28` DOUBLE NULL DEFAULT NULL,
  `v29` DOUBLE NULL DEFAULT NULL,
  `v30` DOUBLE NULL DEFAULT NULL,
  `v31` DOUBLE NULL DEFAULT NULL,
  `v32` DOUBLE NULL DEFAULT NULL,
  `v33` DOUBLE NULL DEFAULT NULL,
  `v34` DOUBLE NULL DEFAULT NULL,
  `v35` DOUBLE NULL DEFAULT NULL,
  `v36` DOUBLE NULL DEFAULT NULL,
  `v37` DOUBLE NULL DEFAULT NULL,
  `v38` DOUBLE NULL DEFAULT NULL,
  `v39` DOUBLE NULL DEFAULT NULL,
  `v40` DOUBLE NULL DEFAULT NULL,
  `v41` DOUBLE NULL DEFAULT NULL,
  `v42` DOUBLE NULL DEFAULT NULL,
  `v43` DOUBLE NULL DEFAULT NULL,
  `v44` DOUBLE NULL DEFAULT NULL,
  `v45` DOUBLE NULL DEFAULT NULL,
  `v46` DOUBLE NULL DEFAULT NULL,
  `v47` DOUBLE NULL DEFAULT NULL,
  `v48` DOUBLE NULL DEFAULT NULL,
  `v49` DOUBLE NULL DEFAULT NULL,
  `v50` DOUBLE NULL DEFAULT NULL,
  `v51` DOUBLE NULL DEFAULT NULL,
  `v52` DOUBLE NULL DEFAULT NULL,
  `v53` DOUBLE NULL DEFAULT NULL,
  `v54` DOUBLE NULL DEFAULT NULL,
  `v55` DOUBLE NULL DEFAULT NULL,
  `v56` DOUBLE NULL DEFAULT NULL,
  `v57` DOUBLE NULL DEFAULT NULL,
  `v58` DOUBLE NULL DEFAULT NULL,
  `v59` DOUBLE NULL DEFAULT NULL,
  `v60` DOUBLE NULL DEFAULT NULL,
  `v61` DOUBLE NULL DEFAULT NULL,
  PRIMARY KEY (`cc`, `ic`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECK
```

1.2 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΦΥΣΙΚΟ ΕΠΙΠΕΔΟ

1.2.1 ΡΥΘΜΙΣΗ ΤΩΝ ΠΑΡΑΜΕΤΡΩΝ ΤΟΥ DBMS

Δεν αλλάξαμε τις παραμέτρους του DBMS, χρησιμοποιήθηκαν οι αρχικοί παράμετροι, όπως ορίζονται από το MYSQL Workbench.

1.2.2 ΡΥΘΜΙΣΗ ΑΣΦΑΛΕΙΑΣ

Φτιάξαμε το backup της βάσης, που το έχουμε στον φάκελο backup του project.

2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΛΟΓΙΣΜΙΚΟΥ

2.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΔΟΜΗ ETL

Αρχικά πρέπει να φέρουμε τα δεδομένα μέσα στη βάση μας για περαιτέρω επεξεργασία. Εδώ καταγράφεται η αρχιτεκτονική της ETL διαδικασίας . Για την πραγματοποίηση του ETL, γράψαμε δύο scripts σε Python, με ονόματα finalcombinecsv.py και mysqlconnector.py . Το πρώτο script(finalcombinecsv.py), διαβάζει όλα τα csv με τα δεδομένα για κάθε χώρα, αλλάζει τις κενές τιμές σε 0 καθώς υπάρχει πρόβλημα στον server, σβήνει τις γραμμές που δεν αντιστοιχούν στις οικογένειες μετρικών που έχουμε διαλέξει με αυτές να δουλέψουμε και τελικά δημιουργεί ένα csv αρχείο(Finalcombinedcsv2.csv) με τα δεδομένα για όλες τις χώρες που χρειαζόμαστε. Έπειτα με την εκτέλεση του δεύτερου script (mysqlconnector.py) που είναι υπεύθυνο για την σύνδεση στο mysql workbench, την δημιουργία της βάσης δεδομένου και την δημιουργία και κλήση της εντολής που θα φορτώσει τα δεδομένα που θέλουμε στον κατάλληλο πίνακα της βάσης.

Επιλέξαμε να διαχειριστούμε 26 χώρες της Ευρώπης τις οποίες θα μπορέσετε να δείτε στο front-end του προγράμματος και τις οικογένειες μετρικών με αρχικά EN, SH και SP. Για να επιλεγούν άλλες οικογένειες μετρικών αρκεί να αλλαχθούν οι πίνακες familyarray και familyarray2 του finalcombinedcsv script.

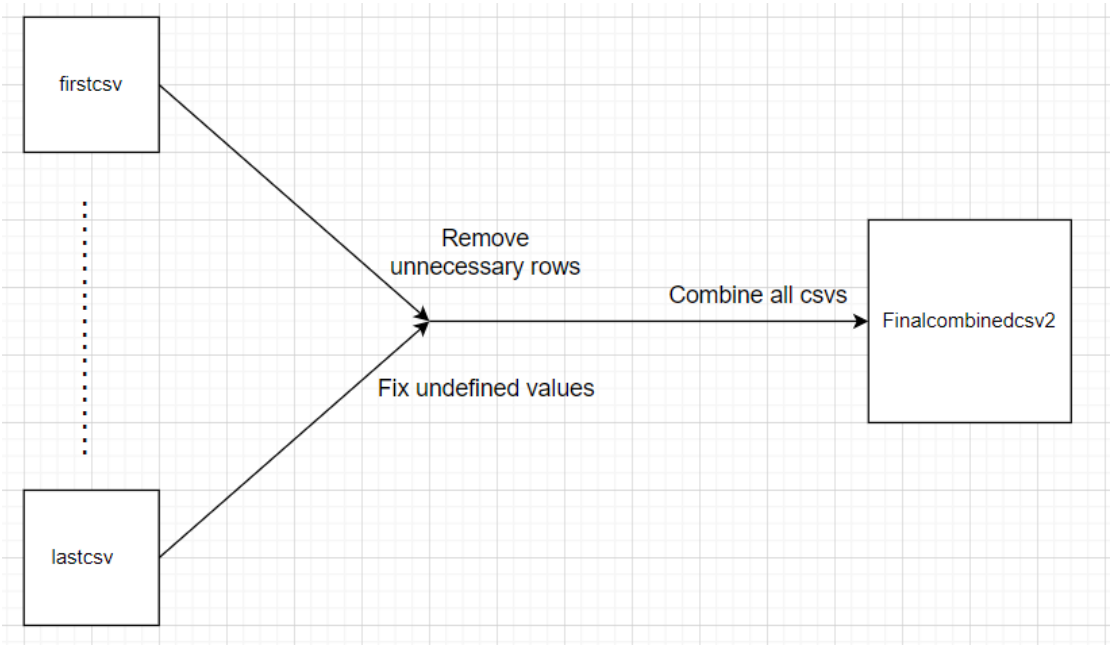
```
familyarray = ['EN', 'SH', 'SP']
familyarray2 = [' EN', ' SH', ' SP']
```

Έπειτα, με την συνάρτηση tablecomm(), φτιάχνουμε την εντολή για την δημιουργία του πίνακα data, το οποίο γίνεται εν μέρη manually, καθώς αν έχουμε άλλα headers στο csv δεν θα τρέχει

σωστά. Αλλά αν απλά στο csv τοποθετηθούν και άλλες χρονιές (π.χ csv από τον ίδιο ιστότοπο σε 10 χρόνια) θα πρέπει να αλλάξει το 62 στο for, καθώς αντιπροσωπεύει το πόσες χρονιές υπάρχουν στο csv.

```
def tablecomm():
    tablecommand = "CREATE TABLE data(cn VARCHAR(256),cc VARCHAR(256),indn VARCHAR(256),ic VARCHAR(256),"
    for i in range(1, 62):
        v = " v"+str(i)+" DOUBLE,"
        tablecommand += v
    tablecommand += " primary key(cc,ic))";
    return tablecommand
```

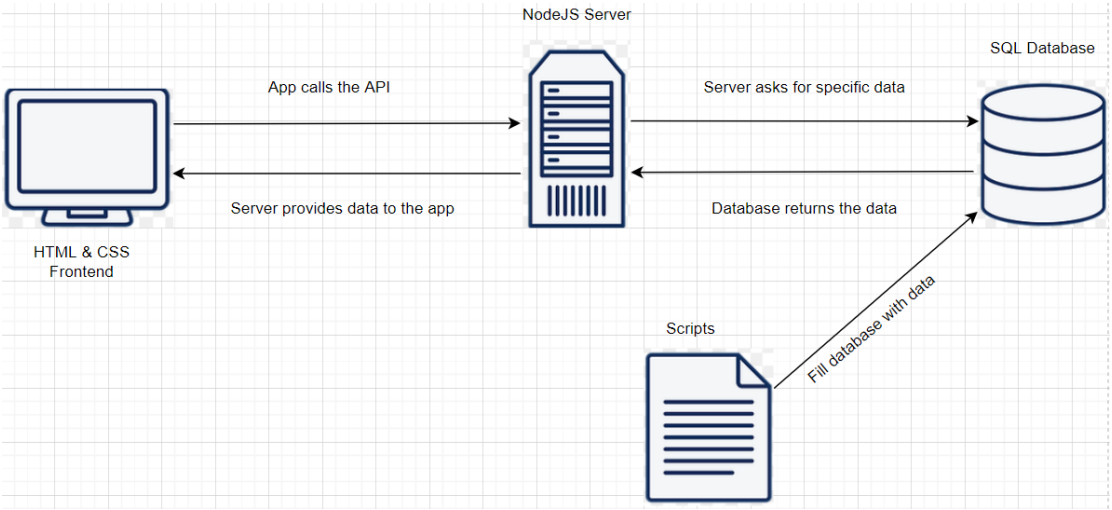
Στο Σχήμα 2.1 απεικονίζεται η αρχιτεκτονική ETL.



Σχήμα 2.1Παράδειγμα τεκμηρίωσης των μετασχηματισμών ETL με ένα απλό σχήμα.

2.2 ΔΙΑΓΡΑΜΜΑΤΑ ΠΑΚΕΤΩΝ / ΥΠΟΣΥΣΤΗΜΑΤΩΝ ΚΕΝΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ

Το διάγραμμα για τα υποσυστήματα του λογισμικού.



Σχήμα 2.2 Διάγραμμα εφαρμογής

Το frontend παίρνει δεδομένα από την φόρμα που συμπληρώνει ο χρήστης και τροφοδοτεί τον server με τα δεδομένα αυτά. Ο οποίος, σύμφωνα με τα δεδομένα αποφασίζει τι ερώτηση θα κάνει στην βάση δεδομένων έτσι ώστε να λάβει το επιθυμητό αποτέλεσμα και τι θα εμφανίσει ξανά στον χρήστη. Η επεξεργασία του γραφήματος που θα εμφανιστεί στον χρήστη γίνεται εντός της HTML, δηλαδή στο frontend.

3 ΥΠΟΔΕΙΓΜΑΤΑ ΕΡΩΤΗΣΕΩΝ ΚΑΙ ΑΠΑΝΤΗΣΕΩΝ

Στην εικόνα 3.1 υποδεικνύουμε την σύνδεση του server με την βάση, την ερώτηση στην βάση και την επεξεργασία του αποτελέσματος.

```
var mysql      = require('mysql2');
var connection = mysql.createConnection({
  host     : '127.0.0.1',
  user     : 'root',
  password : '',
  database : 'mydata'
});

connection.connect();

connection.query('SELECT DISTINCT cn FROM data', function (error, results, fields) {
  if (error) throw error;
  var x = results.length;
  for(i=0; i<x; i++){
    countries.push(results[i].cn);
  }
  //console.log('Countries :', countries);

});
```

Εικόνα 3.1

Στην εικόνα 3.2 στο app.get(), με τις παραμέτρους countries και comparisons εντός του res.render() δίνουμε δεδομένα στο front-end μας, για να δώσει επιλογές στον χρήστη να επιλέξει στην φόρμα. Το app.post(), εκτελείται αφού κάνει submit την φόρμα ο χρήστης και επεξεργαζόμαστε τα δεδομένα που λάβαμε στον server, με σκοπό να τα φέρουμε στην μορφή που θέλουμε. Παρόμοιος είναι και ο κώδικας για timeline και scatterplot.

```
app.get("/barchartindex", (req, res) => {
  res.render("barchartindex", {country: countries, comparison: comparisons})
});
app.post('/barchartindex', (req, res) => {
  //console.log(req.body)
  countryy = JSON.stringify(req.body.fcountry);
  stringg = JSON.stringify(req.body.comp);
  stringg = stringg.substr(2,stringg.length-3);
  stringg2 = JSON.stringify(req.body.scomp);
  stringg2 = stringg2.substr(2,stringg2.length-3);
  stringg3 = JSON.stringify(req.body.tcomp);
  stringg3 = stringg3.substr(2,stringg3.length-3);
  showyears = JSON.stringify(req.body.showyears);
  fyear = parseFloat(req.body.fyear);
  lyear = parseFloat(req.body.lyear);
  res.redirect('/barchart');
});
```

Εικόνα 3.2

Στην εικόνα 3.3 είναι η συνέχεια της εικόνας 3.2, συνεχίζουμε την επεξεργασία των δεδομένων, κάνουμε τις απαραίτητες ερωτήσεις στην βάση και αποθηκεύουμε τα αποτελέσματα τους σε πίνακες.

```
app.get("/barchart", (req, res) => {
  stringg = ' '+stringg;
  stringg2 = ' '+stringg2;
  stringg3 = ' '+stringg3;
  stringg = JSON.stringify(stringg);
  stringg2 = JSON.stringify(stringg2);
  stringg3 = JSON.stringify(stringg3);
  querystring = `SELECT * FROM data WHERE cn = ${countryy} AND indn = ${stringg}`;
  querystring2 = `SELECT * FROM data WHERE cn = ${countryy} AND indn = ${stringg2}`;
  querystring3 = `SELECT * FROM data WHERE cn = ${countryy} AND indn = ${stringg3}`;
  connection.connect();
  connection.query(querystring, function (error, results, fields) {
    if (error) throw error;
    var xxx = [];
    var x = results.length;
    for(i=0; i<x; i++){
      xxx.push(results[i]);
    }
    if(stringg !== '" --"'){
      arrayxxx = Object.values(xxx[0]);
    }
  });
```

Εικόνα 3.3

Στην εικόνα 3.4 δείχνουμε το τελικό στάδιο επεξεργασίας δεδομένων πριν δωθούν στο front-end. Στο datarray, τοποθετούμε το lyear-fyear+1 στην πρώτη θέση που υποδεικνύει το πόσες τιμές έχει ο κάθε πίνακας(κάθε arrayxxx). Η μεταβλητή i αρχικοποιείται fyear-yeartoarrayposition(1960-1956) που είναι 4 γιατί οι θέσεις 0-3 του πίνακα έχουν δεδομένα με το ποια χώρα είναι, κωδικό κτλ. Που δεν τα χρειαζόμαστε για το γράφημα, για αυτό δεν τα τοποθετούμε στην τελική λίστα. Τέλος δημιουργούμε έναν πίνακα stringarray, που κρατάει τα strings της χώρας και των ερωτήσεων που έγιναν στην βάση και καλούμε το frontend, με τις απαραίτητες παραμέτρους(δεδομένα).

```
  datarray = [];
  datarray.push(lyear - fyear + 1);
  for(var i = fyear - yeartoarrayposition; i <= lyear - yeartoarrayposition; i++){
    datarray.push(arrayxxx[i]);
    datarray.push(arrayxxx2[i]);
    datarray.push(arrayxxx3[i]);
  }
  stringarray = [stringg, stringg2, stringg3, countryy];
  res.render("barchart", {x: datarray, strings: stringarray, showyeaar: showyears, fyeaar: fyear, syeaar: lyear});
```

Εικόνα 3.4

Στην συνέχεια, στην εικόνα 3.5 σας δείχνουμε ένα script σε javascript, που μας βοήθησε στην δημιουργία των select της φόρμας σε κάθε html σελίδα. Έχει χρησιμοποιηθεί σχεδόν σε κάθε σελίδα που έχουμε. Με το [var array = “<%= locals.country %>”], περνιέται ο πίνακας country που είχαμε δώσει ως παράμετρο στο κάλεσμα του frontend από τον server και με αυτόν τον τρόπο λαμβάνουμε αυτόν τον πίνακα που αποθηκεύουμε στην μεταβλητή array για να μπορέσουμε να το

διαχειριστούμε. Έπειτα στο fcountry(όνομα που αντιστοιχεί σε ένα πεδίο της φόρμας) περνάμε όλες τις επιλογές που θα εμφανιστούν στον χρήστη σε αυτό το πεδίο.

```
<script>var array = "<%= locals.country %>";</script>
<script>var comparray = "<%= locals.comparison %>";</script>
<script type = "text/javascript">
 (function countriesselect() {
   var elm = document.getElementById('fcountry'),
       df = document.createDocumentFragment();
       var option = document.createElement('option');
       option.value = '---';
       option.appendChild(document.createTextNode('---'));
       df.appendChild(option);
       var arr = array.split(',');
     for (var i = 0; i < arr.length; i++) {
       var option = document.createElement('option');
       option.value = arr[i];
       option.appendChild(document.createTextNode(arr[i]));
       df.appendChild(option);
     }
   elm.appendChild(df);
}());
  </script>
```

```
<label for="fcountry">Country</label>
<select id="fcountry" name="fcountry"></select>
```

Εικόνα 3.5

4 ΛΟΙΠΑ ΣΧΟΛΙΑ

Το μόνο κομμάτι που δεν έχουμε καταφέρει να υλοποιήσουμε καθόλου στο project μας είναι η διαχείριση του time aggregation. Έτσι όπως χειριστήκαμε τα γραφήματα μέσω της d3, μας ήταν πολύ δύσκολο να αλλάξουμε τον x άξονα, καθώς γέμιζε αυτόματα με τις εμφανίσεις των γραφημάτων.