

ΠΡΟΧΩΡΗΜΕΝΑ ΘΕΜΑΤΑ
ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΕΦΑΡΜΟΓΩΝ
ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ
ΤΟ ΑΚΑΔΗΜΑΪΚΟ ΈΤΟΣ 2015-2016

ΟΜΑΔΑ “LACHESIS”

ΝΙΚΟΛΑΟΣ ΖΩΗΣ, 2054

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΜΑΪΟΣ 2016

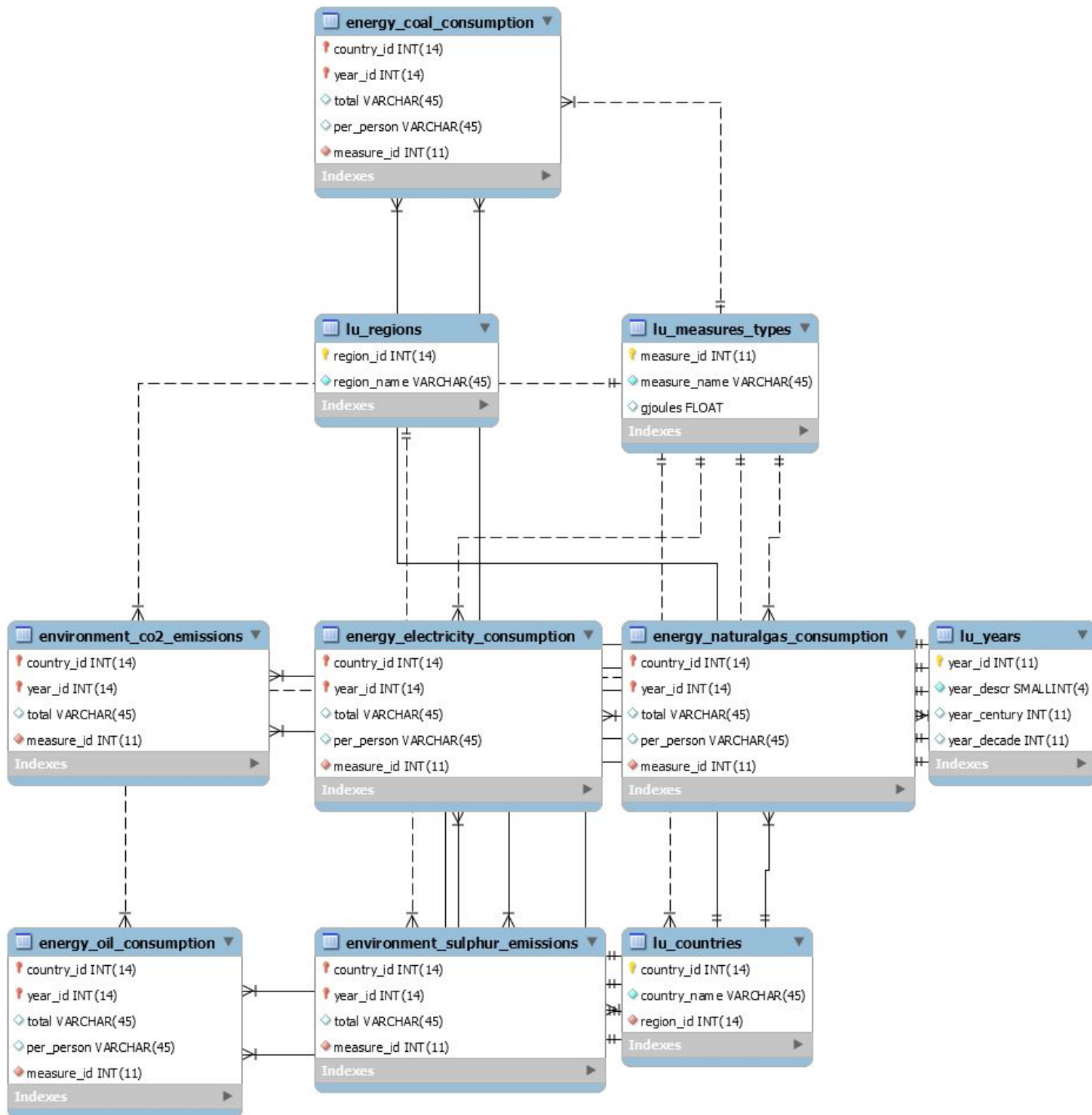
ΙΣΤΟΡΙΚΟ ΠΡΟΗΓΟΥΜΕΝΩΝ ΕΚΔΟΣΕΩΝ

Ημερομηνία	Έκδοση	Περιγραφή	Συγγραφέας
yyyy/mm/dd	x.x		

Το κείμενο συμπληρώνεται προοδευτικά, όπως προχωρείτε στις φάσεις του Project.

1 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

1.1 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΛΟΓΙΚΟ ΕΠΙΠΕΔΟ



1.2 ΣΧΕΣΙΑΚΌ ΣΧΉΜΑ ΣΕ ΦΥΣΙΚΟ ΕΠΊΠΕΔΟ

1.2.1 ΡΥΘΜΙΣΗ ΤΩΝ ΠΑΡΑΜΈΤΡΩΝ ΤΟΥ DBMS

Συμβουλευτήκα την σελίδα <https://www.percona.com> , καλή για αρχάριους, δίνει μια ικανοποιητική ρύθμιση ανάλογα τον υπολογιστή.

default-storage-engine	= InnoDB	innodb-log-files-in-group	= 2
max-allowed-packet	= 16M	innodb-log-file-size	= 64M
max-connect-errors	= 1000000	innodb-flush-log-at-trx-commit	= 1
tmp-table-size	= 32M	innodb-file-per-table	= 1
max-heap-table-size	= 32M	innodb-buffer-pool-size	= 512M
max-connections	= 500		
thread-cache-size	= 50		
open-files-limit	= 65535		
table-cache	= 2048		

1.2.2 ΡΥΘΜΙΣΗ ΤΟΥ ΦΥΣΙΚΟΎ ΣΧΉΜΑΤΟΣ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΈΝΩΝ

Δημιουργία procedures, για αυτόματη φόρτωση δεδομένων, και αναθεση ξένων κλειδιών όπου χρειαστεί, με ασφάλεια.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `create_year`(IN year_descr SMALLINT(4),OUT year_id INT(11))
BEGIN
    DECLARE year_exists int;
    DECLARE century int;
    DECLARE decade int;
    SET year_exists = 0;
    SET century = (year_descr - (year_descr % 100))/100;
    SET decade = (year_descr % 100)/10;

    SELECT count(*) INTO year_exists from lu_years AS t WHERE t.year_descr = year_descr;

    IF (year_exists > 0) THEN
        #SELECT t.year_id FROM lu_years AS t WHERE t.year_descr = year_descr;
        SELECT t.year_id INTO year_id FROM lu_years AS t WHERE t.year_descr = year_descr;
    ELSE
        INSERT INTO lu_years(year_descr,year_century,year_decade) VALUES(year_descr, century, decade);
        #SELECT LAST_INSERT_ID() AS year_id;
        SET year_id = LAST_INSERT_ID();
    END IF;
END
```

```
CREATE DEFINER='root'@'localhost' PROCEDURE `INSERT_ENERGY_RECORD`(IN target_table varchar(45), IN country varchar(45),IN year_descr
BEGIN
    DECLARE country_id INT;
    DECLARE year_id INT;
    DECLARE measure_id INT;

    CALL create_country(country,country_id);
    CALL create_year(year_descr,year_id);
    CALL create_measure(measure,measure_id);

    SELECT country_id,year_id,measure_id;

    SET @query_ex = CONCAT("INSERT INTO ",target_table," (country_id,year_id,total,per_person,measure_id) VALUES ('",country_id,"",year_descr,
    ",",measure_id,"')");

    PREPARE not_prefixed FROM @query_ex;
    EXECUTE not_prefixed;

    DEALLOCATE PREPARE not_prefixed;
END
```

```

CREATE DEFINER='root'@'localhost' PROCEDURE `INSERT_ENVIRONMENT_RECORD`(IN target_table varchar(45), IN country varchar(45), IN year
BEGIN
DECLARE country_id INT;
DECLARE year_id INT;
DECLARE measure_id INT;

CALL create_country(country,country_id);
CALL create_year(year_descr,year_id);
CALL create_measure(measure,measure_id);

SELECT country_id,year_id,measure_id;

SET @query_ex = CONCAT("INSERT INTO ",target_table," (country_id,year_id,total,measure_id) VALUES ('",country_id,"','year_id','to

PREPARE not_prefixed FROM @query_ex;
EXECUTE not_prefixed;

DEALLOCATE PREPARE not_prefixed;
END

CREATE DEFINER='root'@'localhost' PROCEDURE `create_country`(IN country varchar(45),OUT country_id INT(11))
BEGIN
DECLARE country_exists int;
SET country_exists = 0;
SELECT count(*) INTO country_exists from lu_countries AS t WHERE t.country_name = country;

IF (country_exists > 0) THEN
#SELECT t.country_id FROM lu_countries AS t WHERE t.country_name = country;
SELECT t.country_id INTO country_id FROM lu_countries AS t WHERE t.country_name = country;
ELSE
INSERT INTO lu_countries(country_name) VALUES(country);
#SELECT LAST_INSERT_ID() AS country_id;
SET country_id = LAST_INSERT_ID();
END IF;
END

CREATE DEFINER='root'@'localhost' PROCEDURE `create_measure`(IN measure varchar(45),OUT measure_id INT(11))
BEGIN
DECLARE measure_exists int;
SET measure_exists = 0;

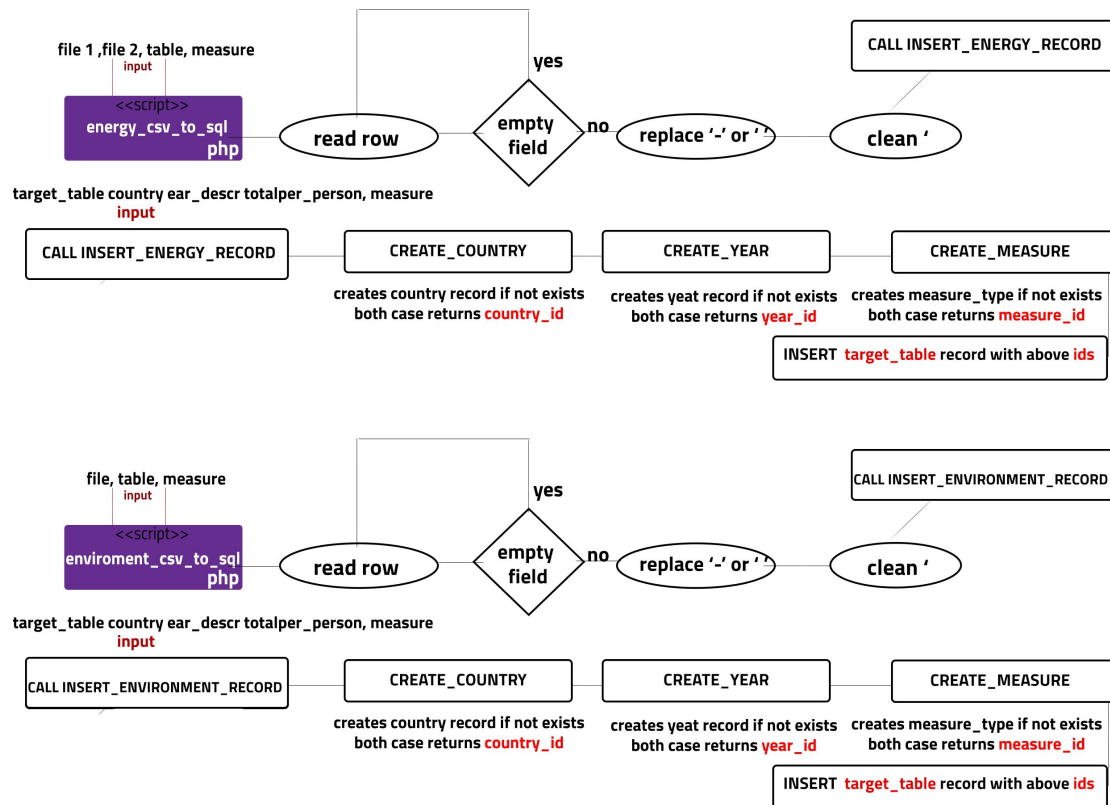
SELECT count(*) INTO measure_exists from lu_measures_types AS t WHERE t.measure_name = measure;

IF (measure_exists > 0) THEN
#SELECT t.measure_id FROM lu_measures_types AS t WHERE t.measure_name = measure;
SELECT t.measure_id INTO measure_id FROM lu_measures_types AS t WHERE t.measure_name = measure;
ELSE
INSERT INTO lu_measures_types(measure_name) VALUES(measure);
#SELECT LAST_INSERT_ID() AS measure_id;
SET measure_id = LAST_INSERT_ID();
END IF;
END

```

2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΛΟΓΙΣΜΙΚΟΥ

2.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΔΟΜΗ ETL



Σχετικά με το enviroment_csv_to_sql.php

* This Script is working only with the following arguments and rules

* file_input_path --a CSV file with 'total' values

* table -- the Table Name from the database which is gonna be affected

* measure -- the measure types of the indicators e.x kWh,tons,ote,etc..

* REQUIREMENTS: You need the SQL Procedures in order to make it work

Σχετικά με το enviroment_csv_to_sql.php

* This Script is working only with the following arguments and rules

* file1_input_path --a CSV file with 'total' values

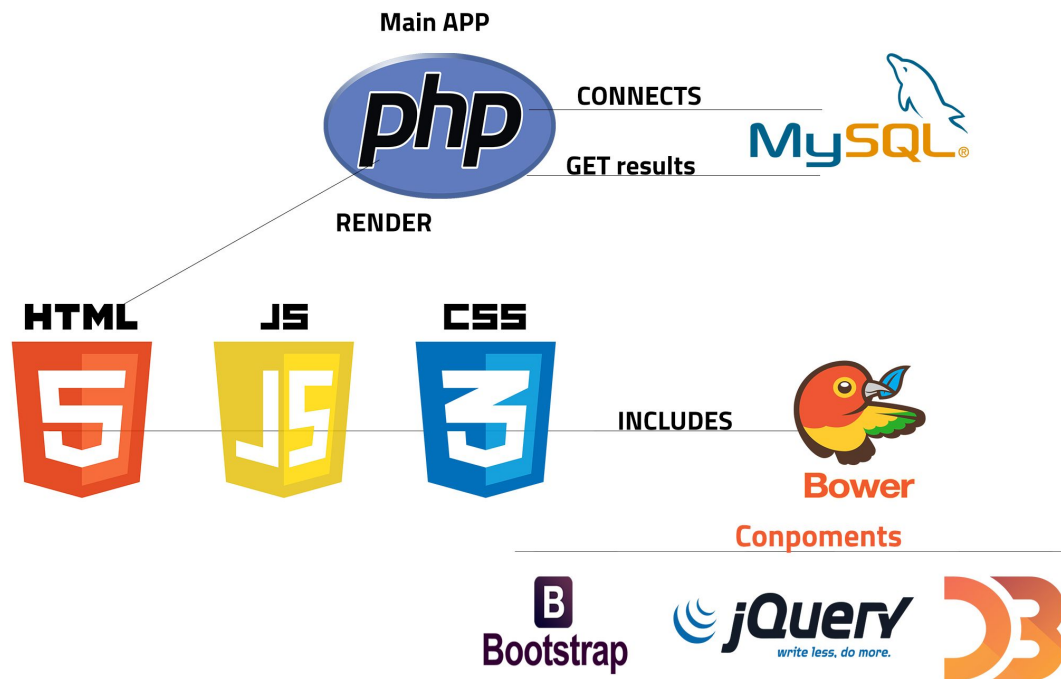
* file2_input_path --a CSV file with 'per person' values

* table -- the Table Name from the database which is gonna be affected

* measure -- the measure types of the indicators e.x kWh,tons,ote,etc..

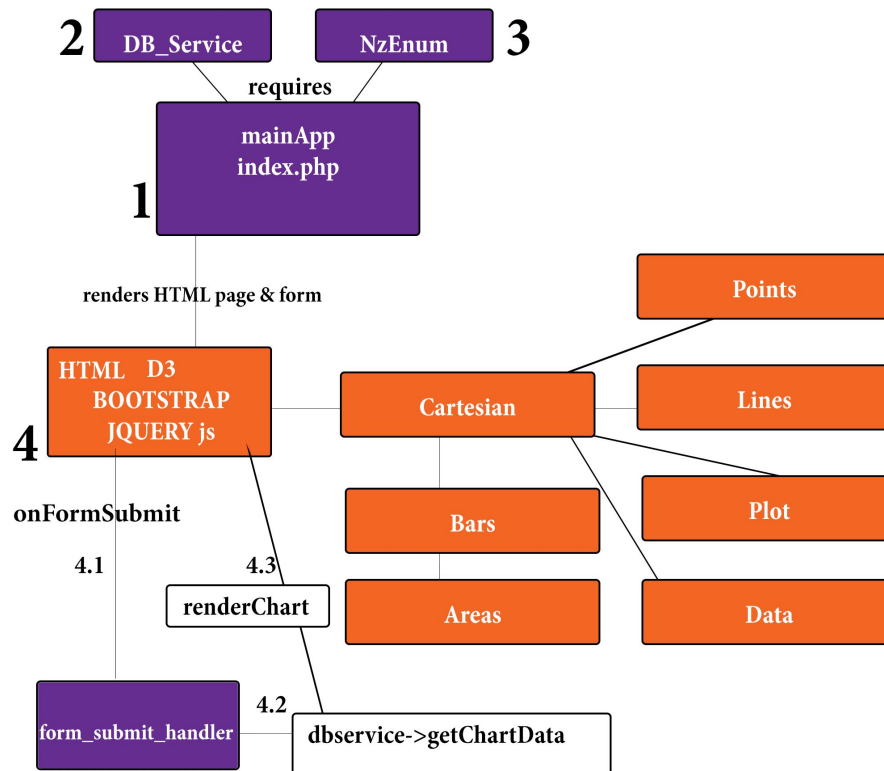
* REQUIREMENTS: You need the SQL Procedures in order to make it work

2.2 ΔΙΑΓΡΑΜΜΑΤΑ ΠΑΚΕΤΩΝ / ΥΠΟΣΥΣΤΗΜΑΤΩΝ ΚΕΝΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ



Η εφαρμογή είναι σε php, βασικό file είναι το "index.php" το οποίο κάνει load το loader.php, μέσα σε αυτό γίνεται establish connection με sql DB αλλά επίσης και Enum custom class, καθώς επίσης και function για τη επιλογή χρώματος ανά πίνακα.

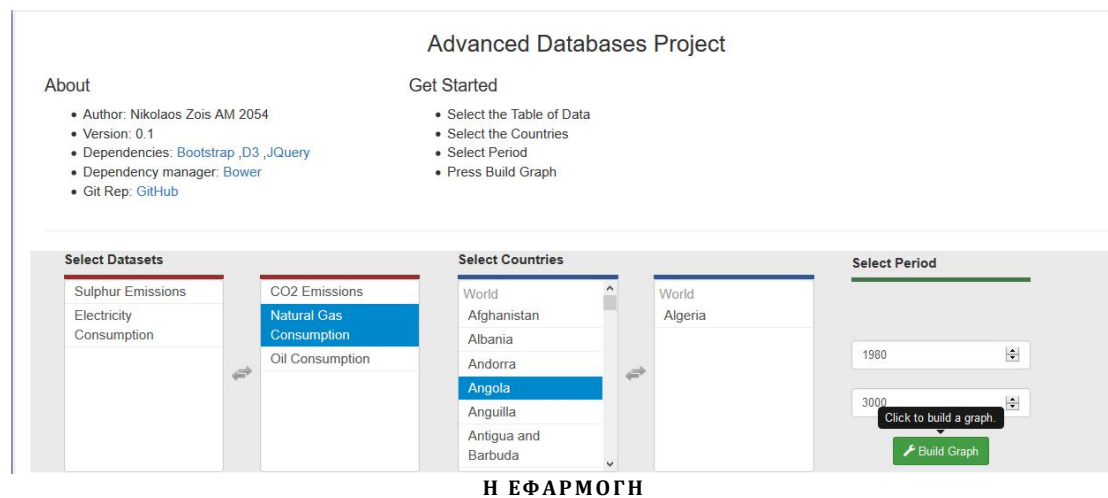
2.3 ΔΙΑΓΡΑΜΜΑ(ΤΑ) ΚΛΑΣΕΩΝ ΚΕΝΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ



Work flow

1. Η εφαρμογή μας ξεκινά με την εκτέλεση του index.php
 - a) Κάνει load το DBService και το NzEnum(Custom Enumeration)
2. Γίνετε η επαλήθευση σύνδεσης με την βάση δεδομένων μας.
3. Φορτώνει πληροφορίες όπως είναι τα datasets,countries,years
4. Γίνετε το Render HTML της σελίδας μας. Αν γίνει submit η φόρμα (αφότου έχουμε επιλέξει τιμές στα πεδία)
 - a) Καλούμε το form_submit_handler.php με **ajax request**.
 - b) Με την βοήθεια του dbservice παίρνουμε τα κατάλληλα δεδομένα.
 - i. Γίνετε μορφοποίηση της πληροφορίας σε **json object** ώστε η βιβλιοθήκη JS PEEK να αποδεχτεί το response.
 - c) Η **json μορφή** γίνετε handle από το ChartFactory το οποίο κάνει implement την Peek Lib
 - d) Εμφανίζουμε δυναμικά τα chart μας.

3 ΥΠΟΔΕΙΓΜΑΤΑ ΕΡΩΤΗΣΕΩΝ ΚΑΙ ΑΠΑΝΤΗΣΕΩΝ



4 ΤΕΚΜΗΡΙΩΣΗ ΚΑΙ ΛΟΙΠΑ ΣΧΌΛΙΑ

Αναβαθμίσεις στο <https://github.com/nikoszoisse/Advanced-Databases>

[Bootstrap](#) (Για το UI)

[D3 - Data Driven Documents](#) (Για γραφικά)

[jQuery](#) (Για το UI)

[Peek - a chart library based on D3.js](#) (Για εύκολα γραφήματα)

[Bower Dependency Package Manager](#) (Για διαχείριση version όλων των παραπαίω)

4.1 ΠΕΡΙ PEEK

Η Peek είναι μία πολύ χρήσιμη βιβλιοθήκη -βασισμένη στην D3-, με την οποία μπορούμε να φτιάξουμε γραφήματα τύπου: scatter, line, area, bar, pie, and donut charts in multiple, σε διάφορους άξονες.

Αυτό που διακρίνει την Peek είναι ότι όλα τα γραφήματα παίρνουν την ίδια μορφή input, και το μόνο που έχουμε να κάνουμε είναι να δώσουμε τη σωστή μορφή input (json) αλλά και τον τύπο chart που μας αρέσει.

Ένα μικρό παράδειγμα setup για ένα line chart είναι το παρακάτω, τόσο απλά!

```
<script>
$(document).ready(function() {
  var data = getFirstGroupData();
  var chart = new Cartesian("#line-chart");
  chart.plot.axes.x.showTicks = false;
  chart.plot.axes.y.showTicks = false;
  chart.draw(data);
  var legend = new Legend("#line-legend");
  legend.draw(data);
});
</script>

<div id="line-chart"></div>
<div id="line-legend"></div>
```

