

ΠΡΟΧΩΡΗΜΕΝΑ ΘΕΜΑΤΑ  
ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ  
ΕΦΑΡΜΟΓΩΝ ΒΑΣΕΩΝ  
ΔΕΔΟΜΕΝΩΝ

---

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ  
ΕΡΓΑΣΙΑ ΓΙΑ ΤΟ  
ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2020-  
2021

---

---

TEAMCSE

ΒΛΑΧΟΘΑΝΑΣΗΣ ΝΙΚΟΛΑΟΣ, 2652

ΠΑΣΟΗ ΣΟΦΙΑ, 2798

---

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΜΑΪΟΣ 2021

#### ΙΣΤΟΡΙΚΟ ΠΡΟΗΓΟΥΜΕΝΩΝ ΕΚΔΟΣΕΩΝ

Ημερομηνία	Έκδοση	Περιγραφή	Συγγραφέας
2021/04/22	1.0	Υλοποίηση της Βάσης, του GUI και κατασκευή δυναμικών queries.	teamCSE
2021/05/27	2.0	Refactor του κώδικα και υλοποίηση των plots.	teamCSE

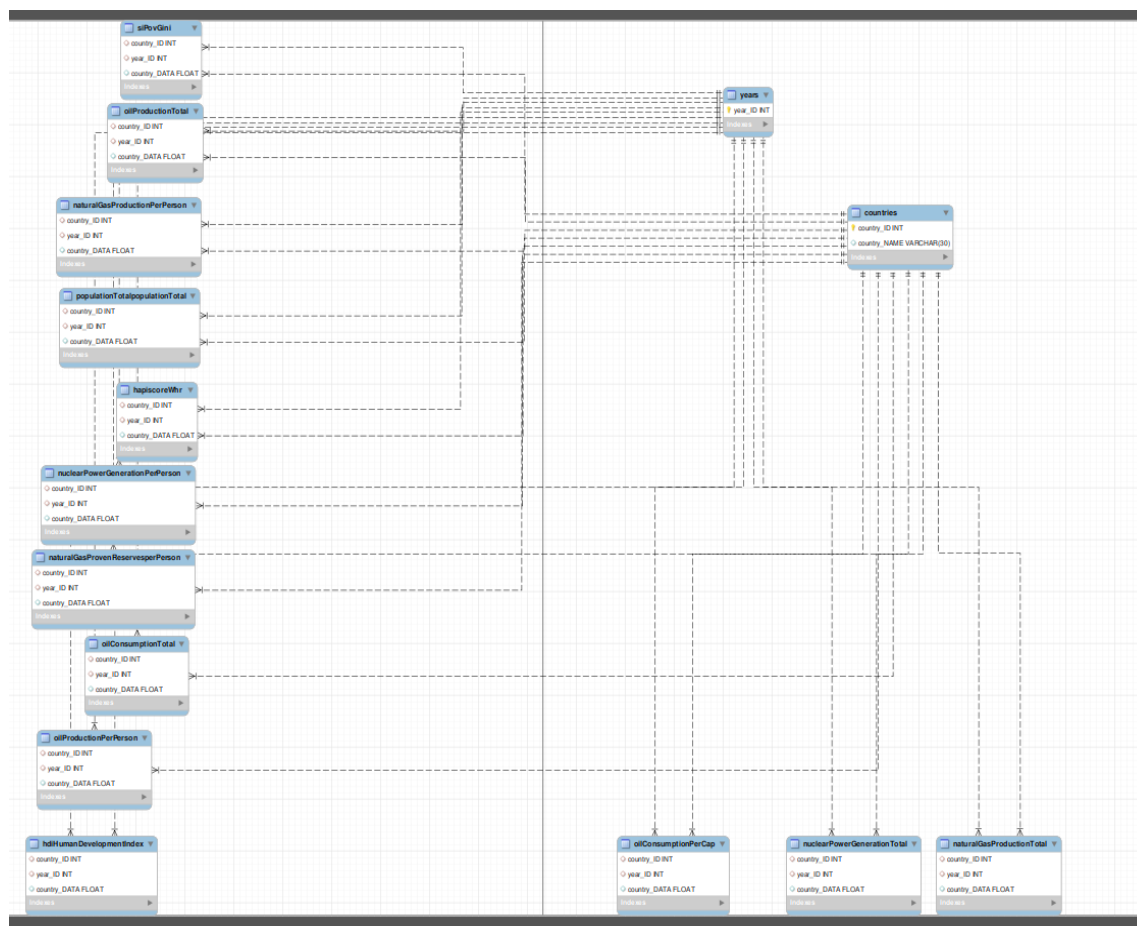
# 1 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Για τον σχεδιασμό της βάσης δεδομένων επιλέξαμε την προτεινόμενη σχεδιαστική λύση Α, από την εκφώνηση. Κρατάμε ένα πίνακα για κάθε πρόβλημα, καθώς και πίνακες για τις χώρες και τις χρονολογίες. Η επιλογή έγινε έπειτα από σύγκριση με τις υπόλοιπες προτάσεις. Αρχικά απορρίψαμε την πρόταση Β, διότι παρόλο που προσφέρει ευκολία και χώρο, σε μία μελλοντική επέκταση της βάσης, τα ερωτήματα θα γίνουν αρκετά αργά, πράγμα που θα κάνει αργή την εφαρμογή. Έπειτα απορρίψαμε την πρόταση Γ, διότι παρόλο που προσφέρει ευκολία ως προς την τροφοδότηση της βάσης(απλώς “ταίζουμε τη βάση με τα αρχικά .csv files), δεσμεύει αρκετό αποθηκευτικό χώρο καθώς για κάθε δείκτη πρέπει να κρατάμε πολλούς πίνακες.

Βέβαια αξίζει να αναφερθεί το γεγονός πως και η λύση που επιλέξαμε δεν είναι τέλεια. Από τη μία μας προσφέρει γρήγορα ερωτήματα(ακόμα και σε μεγάλες βάσεις), αλλά από την άλλη τα ερωτήματα της σχεσιακής άλγεβρας γίνονται πιο πολύπλοκα.

## 1.1 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΛΟΓΙΚΟ ΕΠΙΠΕΔΟ

Το σχεσιακό σχήμα της βάσης δεδομένων, φαίνεται στην παρακάτω εικόνα.



Η κατασκευή της βάσης και των πινάκων έγιναν με χρήση SQL scripts (βρίσκονται στον φάκελο SQLscripts του repository στο GitHub). Η τροφοδότηση των πινάκων έγινε μέσω του Workbench(δεξί κλικ στο όνομα του πίνακα, πατάμε την επιλογή “Table Data Import Wizard”, βάζουμε το path του αρχείου και πατάμε next.).

---

## 1.2 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΦΥΣΙΚΟ ΕΠΙΠΕΔΟ

---

Δεν πραγματοποιήσαμε καμία παραπάνω ρύθμιση, οπότε όλα έχουν μπει by Default.

## 2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΛΟΓΙΣΜΙΚΟΎ

### 2.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ

Όπως φαίνεται στην παρακάτω εικόνα, η δομή των αρχείων .csv που πήραμε από το GapMinder, δεν ήταν βολική ώστε να φορτωθούν τα δεδομένα απευθείας στη βάση.

	country	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
1	Afghanistan				0.372	0.44	0.476	0.383	0.378	0.357	0.313	0.398	0.422	0.266	0.269	0.257
2	Albania			0.463		0.549	0.527	0.587	0.551	0.455	0.481	0.461	0.451	0.464	0.5	0.488
3	Algeria						0.546	0.532	0.56		0.635		0.534	0.525	0.504	0.501
4	Angola							0.559	0.436	0.394	0.379					
5	Argentina		0.631	0.607	0.596	0.642	0.644	0.678	0.647	0.658	0.667	0.67	0.643	0.604	0.579	0.597
6	Armenia		0.429	0.488	0.465	0.418	0.437	0.426	0.432	0.428	0.445	0.435	0.433	0.429	0.506	0.468
7	Australia	0.734		0.729	0.725		0.745	0.741	0.72	0.736	0.729	0.731	0.725	0.726	0.718	0.722
8	Austria		0.712		0.718		0.73	0.747	0.74	0.75	0.695	0.708	0.705	0.729	0.74	0.729
9	Azerbaijan		0.473	0.457	0.482	0.457	0.422	0.468	0.491	0.548	0.525	0.515	0.53	0.515	0.517	0.516

countryId	countryName
0	Afghanistan
1	Albania
2	Algeria
3	Angola
4	Argentina
5	Armenia
6	Australia
7	Austria

Για κατασκευή του πίνακα με τις χώρες και τα id τους, υλοποιήσαμε ένα Python Script που μας δίνει ένα αρχείο στη μορφή που φαίνεται στη διπλανή εικόνα. Πήρε ως είσοδο όλα τα αρχεία των δεικτών, διάβασε τις χώρες, πέταξε τα διπλότυπα, τις ταξινόμησε αλφαβητικά και σε κάθε row του νέου csv έγραψε το όνομα της χώρας και ως id τον αριθμό του row στο οποίο γράφτηκε.

yearId
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812

Κάτι αντίστοιχο έγινε και για τον πίνακα με τις χρονολογίες, με τη μόνη διαφορά στο id. Εδώ κρατάμε μόνο τη χρονολογία.

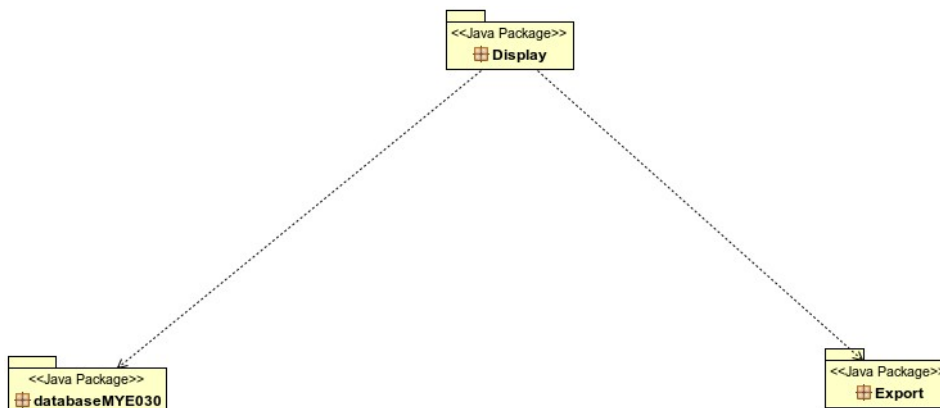
Id	Year	Data
0	2005	0
0	2006	0
0	2007	0
0	2008	0.372
0	2009	0.44
0	2010	0.476
0	2011	0.383
0	2012	0.378
0	2013	0.357
0	2014	0.313

Οι πίνακες για τους δείκτες έχουν τα εξής πεδία:

1. id της χώρας
2. id της χρονιάς
3. data

Για να είμαστε σε θέση να φορτώσουμε τα δεδομένα και οι τιμές των πεδίων να λάβουν τη σωστή τιμή υλοποιήσαμε ένα Python Script, το οποίο κατασκευάζει .csv files στην κατάλληλη μορφή, όπως φαίνεται στη διπλανή φωτογραφία.

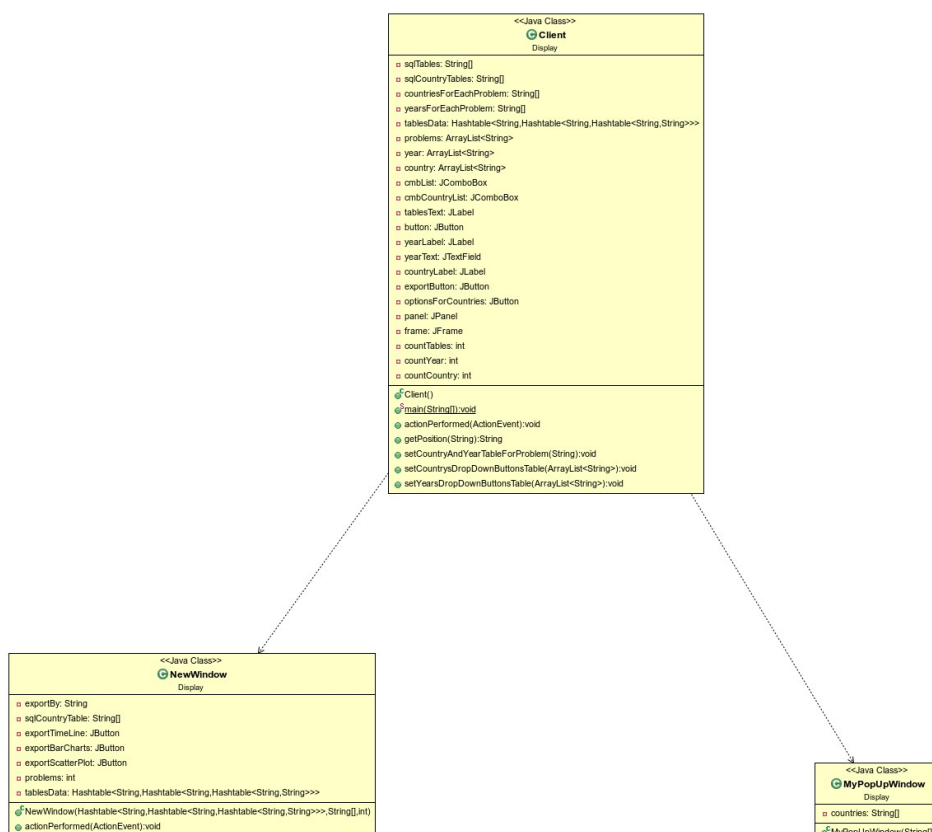
## 2.2 ΔΙΑΓΡΑΜΜΑΤΑ ΠΑΚΕΤΩΝ / ΥΠΟΣΥΣΤΗΜΑΤΩΝ ΚΕΝΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ



Το Package Display διαχειρίζεται τη γραφική διεπαφή(GUI), το Package Export τα plots που εμφανίζονται και το Package databaseMYE030 οποιαδήποτε λειτουργία σχετίζεται με τη βάση(σύνδεση, κατασκευή ερωτημάτων, κατασκευή δομής με τα δεδομένα που επιστρέφουν τα ερωτήματα).

## 2.3 ΔΙΑΓΡΑΜΜΑ(ΤΑ) ΚΛΑΣΕΩΝ ΚΕΝΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ

Package Display



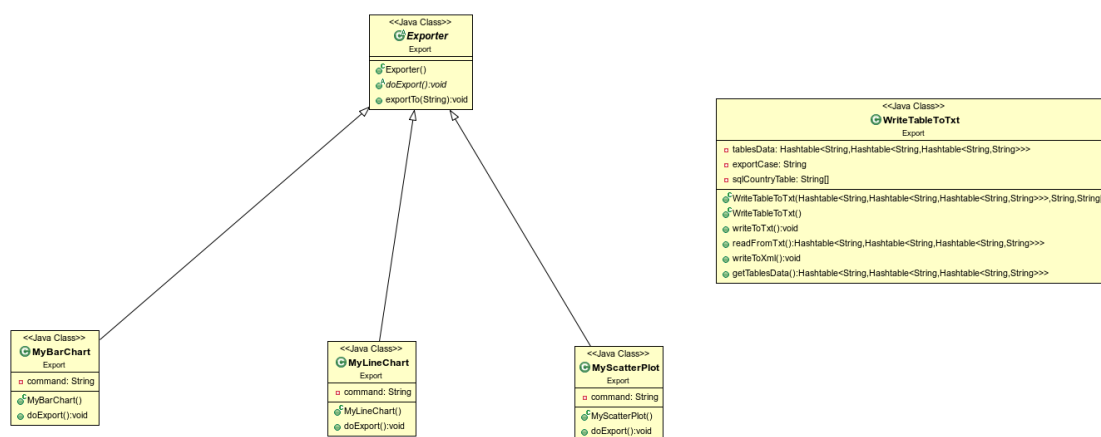


Η κλάση Client διαχειρίζεται το πρώτο παράθυρο του GUI. Πιο συγκεκριμένα, τα προβλήματα που επιλέγει ο χρήστης, τις ημερομηνίες και τις χώρες. Διαχειρίζεται ακόμα και το κουμπί της υποβολής μιας ερώτησης, της εμφάνισης των plots και της εμφάνισης των κοινών χωρών των προβλημάτων.

Η κλάση NewWindow διαχειρίζεται τα κουμπιά του παραθύρου που εμφανίζεται όταν ο χρήστης πατήσει το κουμπί εμφάνισης των plots στο αρχικό παράθυρο. Σε αυτό το παράθυρο ο χρήστης επιλέγει τον τύπο του plot.

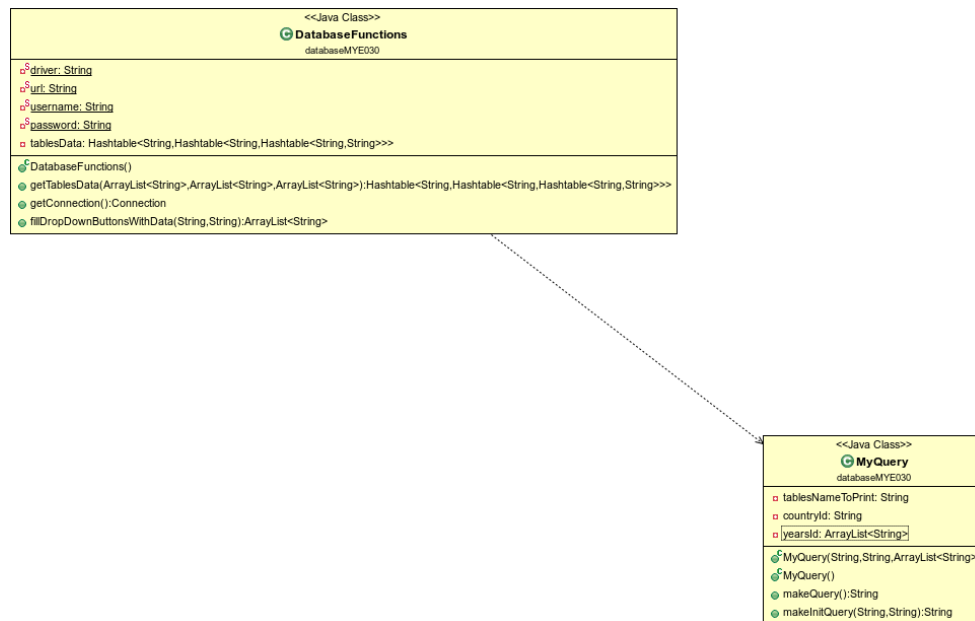
Η κλάση MyPopUpWindow φτιάχνει ένα νέο παράθυρο με τις κοινές χώρες των προβλημάτων, αφότου ο χρήστης πατήσει το κατάλληλο κουμπί στο αρχικό παράθυρο.

### Package Export



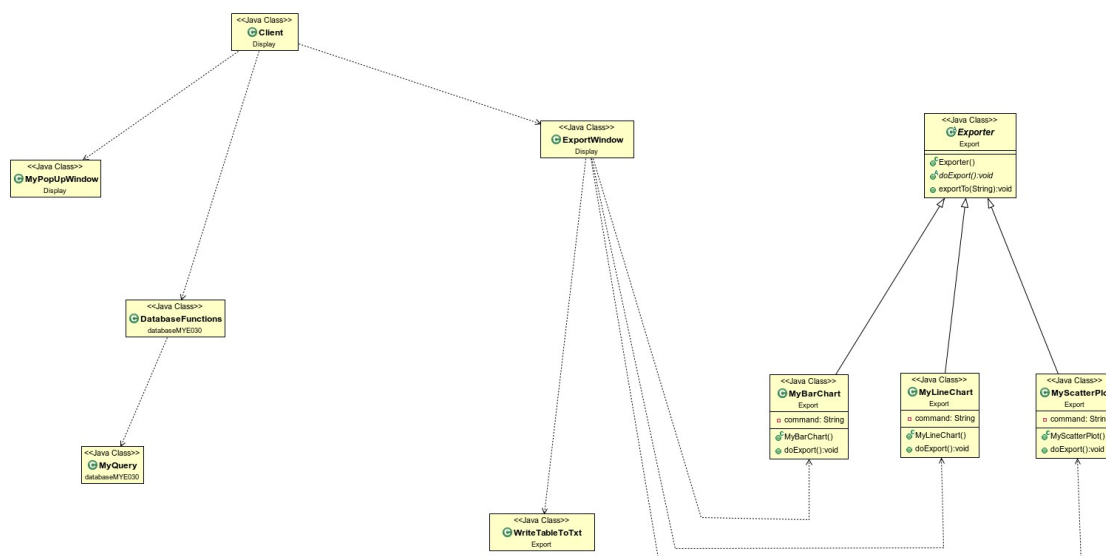
Η κλάση Exporter είναι μια Abstract Class, η οποία έχει μια μέθοδο που καλεί και εκτελεί ένα Python Script που κατασκευάζει τα plots. Τη μέθοδο αυτή υλοποιούν οι κλάσεις παιδιά, όπου καθένα από αυτά καλεί το κατάλληλο Python Script, ώστε να εμφανίσει τον επιλεγμένο τύπο plot. Η κλάση WriteTableToTxt γράφει σε ένα txt αρχείο το hash table με τα δεδομένα που θέλουμε, ώστε να το διαχειριστούν τα Scripts για να φτιάξουν τα plots.

## Package databaseMYE030



Η κλάση DatabaseFunctions καλείται με δεδομένα από το GUI, τα οποία τροφοδοτεί στην κλάση MyQuery ώστε να κατασκευάσει το ερώτημα και να το γυρίσει πίσω στην DatabaseFunctions. Επίσης, πραγματοποιεί τη σύνδεση με τη βάση, της υποβάλει το ερώτημα και κατασκευάζει τη δομή με τα δεδομένα.

## Total UML Diagram



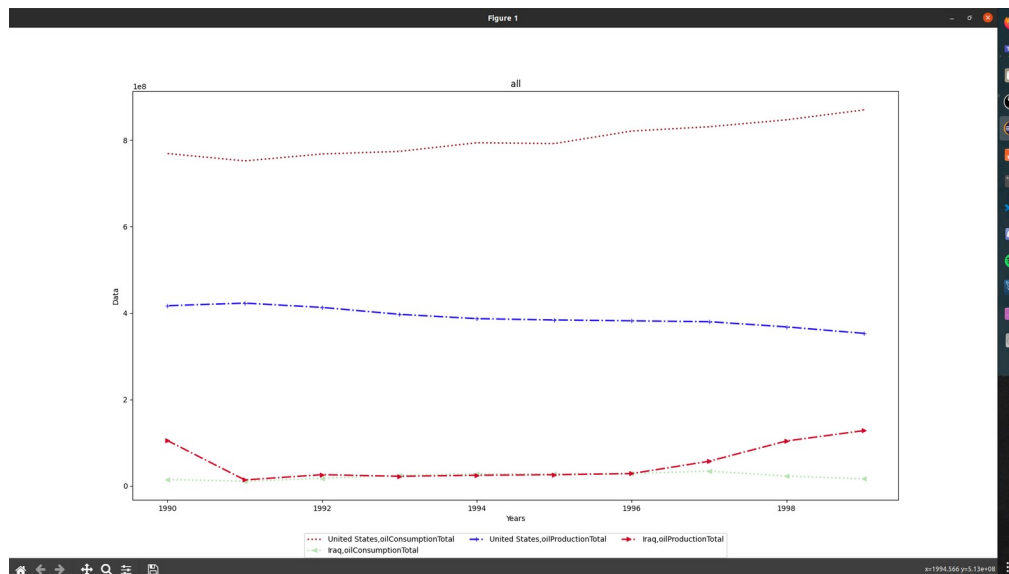
### 3 ΥΠΟΔΕΙΓΜΑΤΑ ΕΡΩΤΗΣΕΩΝ ΚΑΙ ΑΠΑΝΤΗΣΕΩΝ

Αρχικά εμφανίζεται το παρακάτω παράθυρο. Επιλέγουμε ποια προβλήματα θέλουμε.

Έπειτα, επιλέγουμε τις χρονολογίες που μας ενδιαφέρουν. Εμφανίζεται ένα επιπλέον παράθυρο, όπου μπορούμε να δούμε τις χώρες που μπορούμε να επιλέξουμε(ουσιαστικά τις κοινές χώρες των προβλημάτων.).

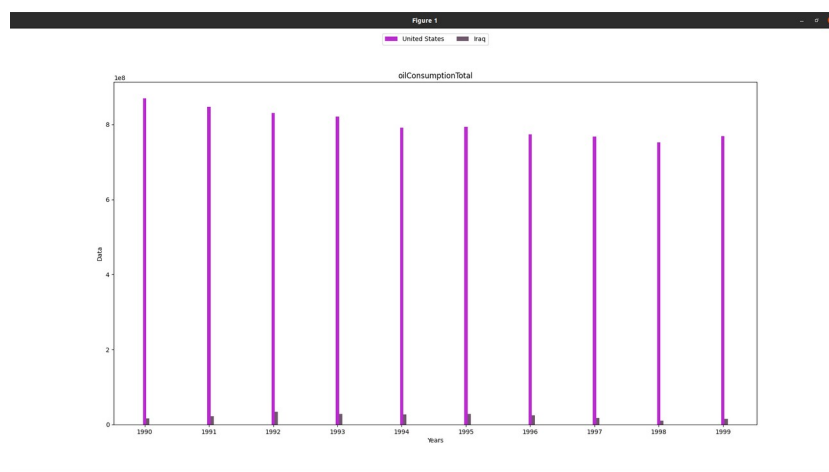
Πατάμε το κουμπί “Ask a Question” για να πάρουμε τα δεδομένα που θέλουμε από τη βάση. Μετά πατάμε το κουμπί “Export” για να επιλέξουμε ποιο plot θέλουμε να εμφανίσουμε.

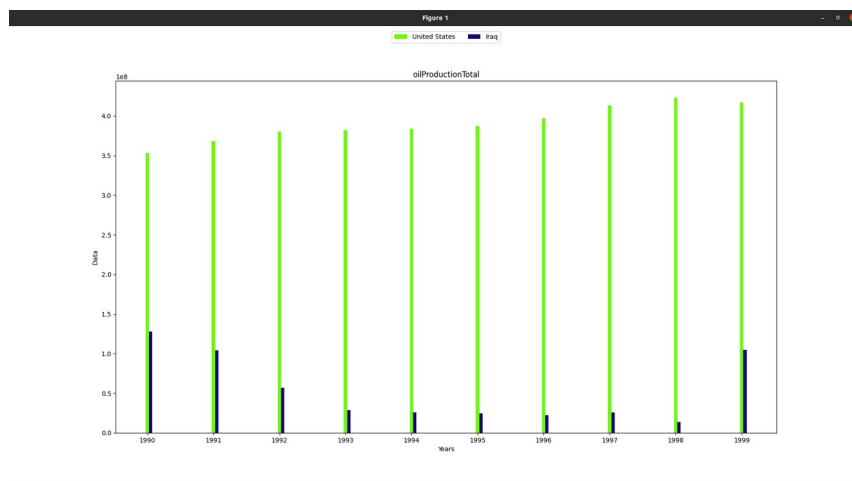
Πατώντας “Export To Time Line” το εμφανίζεται το παρακάτω:



Στον άξονα Y έχουμε τα δεδομένα των προβλημάτων κάθε χώρας και στον άξονα X τις χρονολογίες. Κάτω στο κέντρο μπορούμε να δούμε το όνομα, το χρώμα και τη μορφή της γραμμής κάθε καμπύλης. Για παράδειγμα, η διακεκομμένη πράσινη αναφέρεται στο πρόβλημα “oilConsumptionTotal” της χώρας Iraq.

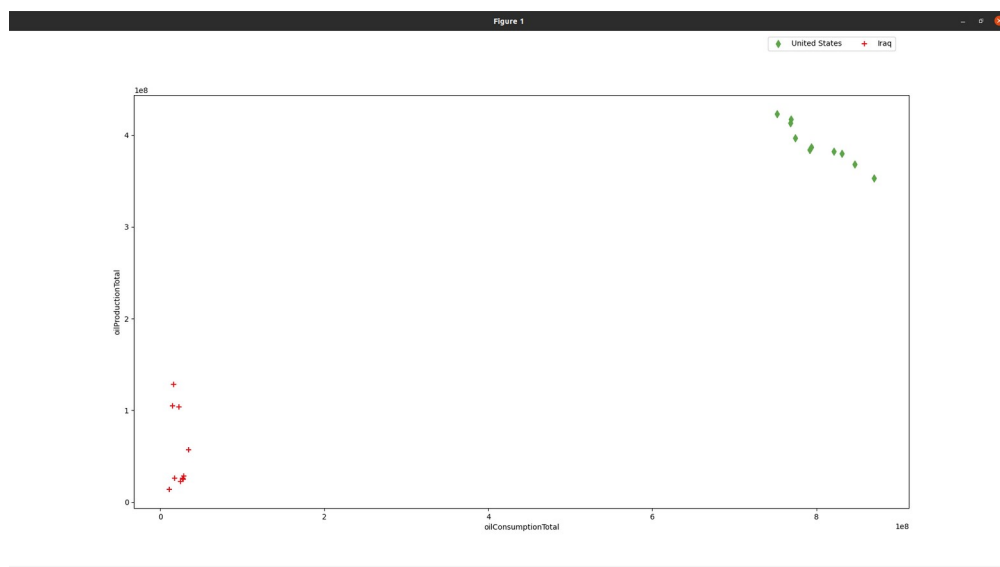
Πατώντας “Export To Bar Charts” το εμφανίζονται τα παρακάτω:





Το έχουμε υλοποιήσει με τέτοιο τρόπο ώστε να εμφανίζεται ένα plot για κάθε πρόβλημα (με όσες χώρες θέλουμε προφανώς). Διαφορετικά, δεν θα ήταν ευδιάκριτο. Στον άξονα Y έχουμε τα δεδομένα των προβλημάτων κάθε χώρας και στον άξονα X τις χρονολογίες. Πάνω στο κέντρο μπορούμε να δούμε το όνομα και το χρώμα του κάθε bar. Για παράδειγμα, το bar με λαχανί χρώμα αναφέρεται στο πρόβλημα “oilProductionTotal” της χώρας Iraq, ενώ το bar με σκούρο μώβ στο “oilConsumptionTotal” της ίδιας χώρας.

Πατώντας “Export To Scatter Plots” το εμφανίζεται το παρακάτω:



Στον άξονα Y έχουμε τα δεδομένα ενός προβλήματος και στον άξονα X ενός άλλου. Κάθε διαφορετικό χρώμα αντιπροσωπεύει 1 χώρα. Πάνω αριστερά μπορούμε να δούμε το όνομα και το χρώμα της κάθε χώρας. Για παράδειγμα, το κόκκινο plus αναφέρεται στη χώρα Iraq.

## 4 ΤΕΚΜΗΡΙΩΣΗ ΚΑΙ ΛΟΙΠΑ ΣΧΟΛΙΑ

---

Το frontend και το backend της εφαρμογής είναι γραμμένα σε Java. Τα scripts της μετατροπής των αρχικών csv αρχείων, καθώς και των plots είναι γραμμένα σε Python. Η βάση και οι πίνακες κατασκευάστηκαν με SQL scripts.

Τα περιβάλλοντα ανάπτυξης που χρησιμοποιήθηκαν είναι:

1. Workbench για την SQL.
2. VisualCodeStudio για την Python.
3. Eclipse για την Java.

Για πιθανές μελλοντικές αλλαγές έχουμε φτιάξει ένα TODO List:

1. Αλλαγή του GUI:
  - a. Να παίρνουμε περισσότερους περιορισμούς για τα countries και years.
  - b. Να τροποποιήσουμε τα κουμπιά ώστε να εμφανίζονται αυτόματα τα κοινά countries και years(χωρίς popups windows).
2. Να προσθέσουμε παραπάνω πεδία στον πίνακα years, όπως 5ετίες, 10ετίες κτλ. Αντίστοιχα, και κουμπιά στον GUI, για να διαλέγει ο χρήστης πως θα εμφανίζονται οι χρονολογίες.
3. Όσον αφορά τα Scatter Plots, να προσθέσουμε ένα πεδίο όπου ο χρήστης θα επιλέγει ένα πρόβλημα σαν σταθερό για να το συγκρίνει με όσα θέλει. Αυτή τη στιγμή είναι υλοποιημένο να εμφανίζεται μόνο όταν ο χρήστης επιλέγει 2 προβλήματα, διότι δεν ξέραμε ποιο να κρατήσουμε ως βάση.