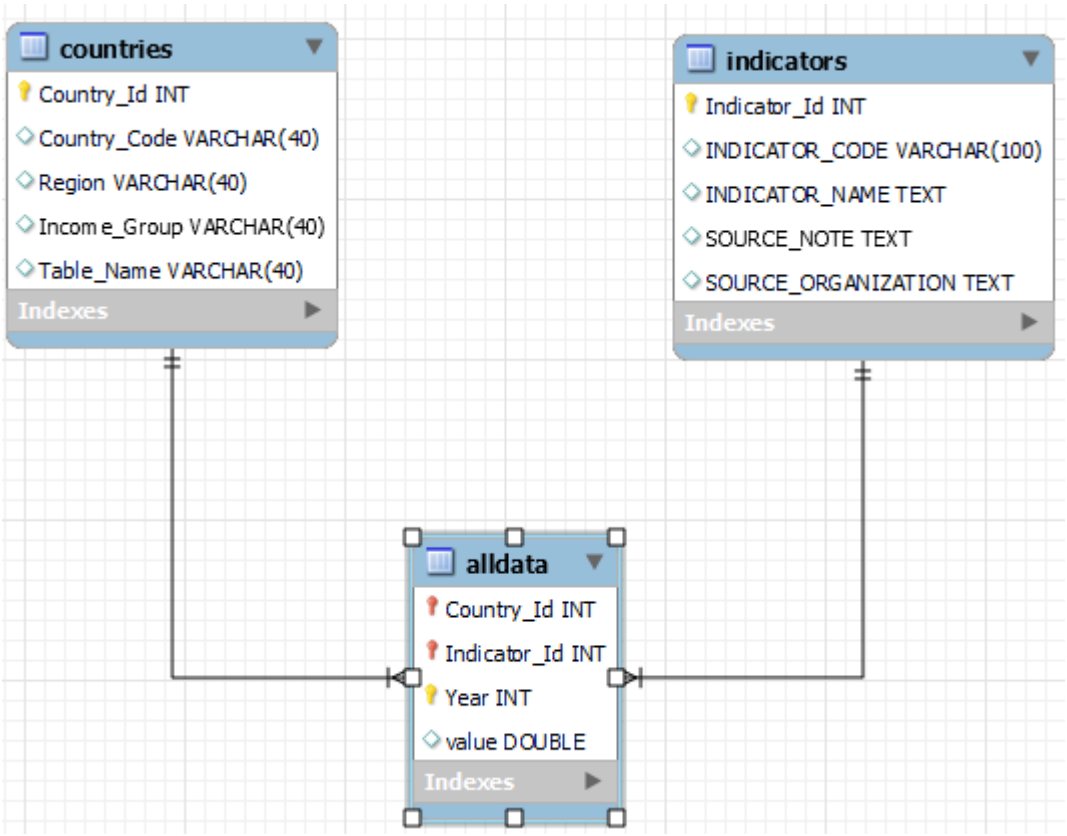
1 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Έχουμε υλοποίηση μια βάση δεδομένων , η οποία περιέχει 3 tables :

- **Countries** : Περιέχει ένα μοναδικό Id (Country_Id) και όλες τις απαραίτητες πληροφορίες για κάθε μια από τις χώρες.
- **Indicators** :Περιέχει ένα μοναδικό Id (Indicator_Id) και όλες τις απαραίτητες πληροφορίες για κάθε μία από τις μετρικές.
- **AllData** : Είναι το main table μας και περιέχει τις μετρήσεις για κάθε χώρα, μετρική και χρονιά. Για να συνδυαστεί όμως με τα παραπάνω table (ώστε να μπορούμε να πάρουμε διάφορα δεδομένα) έχουμε βάλει 2 foreign keys ως προς τα Country_Id(countries) και Indicator_Id(Indicators).

Έτσι καταλήγουμε σε μια δομή βάσης 3NF , ώστε να μπορέσουμε να διαχειριστούμε καλύτερα τα δεδομένα μας.

1.1 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΛΟΓΙΚΟ ΕΠΙΠΕΔΟ



Σχήμα 1-1 Σχεσιακό σχήμα της βάσης δεδομένων του συστήματος

```
-- MySQL Script generated by MySQL Workbench

-- 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_T
ABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FO
R_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- Schema mydb

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

-- Schema project

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

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

-- Schema project

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

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

USE `project` ;

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

-- Table `project`.`countries`

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

CREATE TABLE IF NOT EXISTS `project`.`countries` (

 `Country_Id` INT NOT NULL,

 `Country_Code` VARCHAR(40) NULL DEFAULT NULL,

 `Region` VARCHAR(40) NULL DEFAULT NULL,

 `Income_Group` VARCHAR(40) NULL DEFAULT NULL,

 `Table_Name` VARCHAR(40) NULL DEFAULT NULL,

 PRIMARY KEY (`Country_Id`),

 UNIQUE INDEX `Country_Code` (`Country_Code` ASC)
VISIBLE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4_0900_ai_ci;

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

-- Table `project`.`indicators`

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

CREATE TABLE IF NOT EXISTS `project`.`indicators` (

 `Indicator_Id` INT NOT NULL,

 `INDICATOR_CODE` VARCHAR(100) NULL DEFAULT
NULL,

 `INDICATOR_NAME` TEXT NULL DEFAULT NULL,

 `SOURCE_NOTE` TEXT NULL DEFAULT NULL,

 `SOURCE_ORGANIZATION` TEXT NULL DEFAULT
NULL,

 PRIMARY KEY (`Indicator_Id`),

 UNIQUE INDEX `INDICATOR_CODE`
(`INDICATOR_CODE` ASC) VISIBLE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4_0900_ai_ci;

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

-- Table `project`.`alldata`

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

CREATE TABLE IF NOT EXISTS `project`.`alldata` (

 `Country_Id` INT NOT NULL,

 `Indicator_Id` INT NOT NULL,

 `Year` INT NOT NULL,

 `value` DOUBLE NULL DEFAULT NULL,

 PRIMARY KEY (`Country_Id`, `Indicator_Id`, `Year`),

 INDEX `Indicator_Id` (`Indicator_Id` ASC) VISIBLE,

 CONSTRAINT `alldata_ibfk_1`

   FOREIGN KEY (`Country_Id`)

   REFERENCES `project`.`countries` (`Country_Id`),

 CONSTRAINT `alldata_ibfk_2`

   FOREIGN KEY (`Indicator_Id`)

   REFERENCES `project`.`indicators` (`Indicator_Id`))

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_CHECKS;
```

1.2 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΦΥΣΙΚΟ ΕΠΙΠΕΔΟ

1.2.1 ΡΥΘΜΙΣΗ ΤΩΝ ΠΑΡΑΜΕΤΡΩΝ ΤΟΥ DBMS

Δεν χρειάστηκε να κάνουμε κάποια ρύθμιση παραμέτρων , αφού το storage engine που θέλαμε να χρησιμοποιήσουμε ήταν το InnoDB , το οποίο είναι το Default engine τις mysql.

1.2.2 ΡΥΘΜΙΣΗ ΤΟΥ ΦΥΣΙΚΟΥ ΣΧΗΜΑΤΟΣ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Επειδή τα δεδομένα μας ήταν πολλά στον main πίνακα και υπήρχε μεγάλη καθυστέρηση, αναγκαστήκαμε να εκτελέσουμε την εντολή
“SET @@FOREIGN_KEY_CHECKS = 0”. Έτσι δεν χρειάζεται η βάση να ελέγχει τα ξένα κλειδιά και φορτώνει τα δεδομένα αρκετά πιο γρήγορα.

1.2.3 ΡΥΘΜΙΣΗ ΑΣΦΑΛΕΙΑΣ

Όσον αφορά τις ρυθμίσεις ασφαλείας , εκτελέσαμε την εντολή
‘SET GLOBAL local_infile = 1’ και στην συνέχεια ‘πειράζοντας’ λίγο το αρχείο my.cnf επιτρέψαμε να μπορούν να φορτώνονται αρχεία μέσα στην βάση , ασχέτως από το που βρίσκονται. Έτσι καταφέραμε να χρησιμοποιήσουμε την εντολή LOAD DATA LOCAL INFILE.

2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΛΟΓΙΣΜΙΚΟΥ

2.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΔΟΜΗ ETL

Για να φορτωθούν τα δεδομένα μέσα στην βάση δημιουργήσαμε 2 scripts :

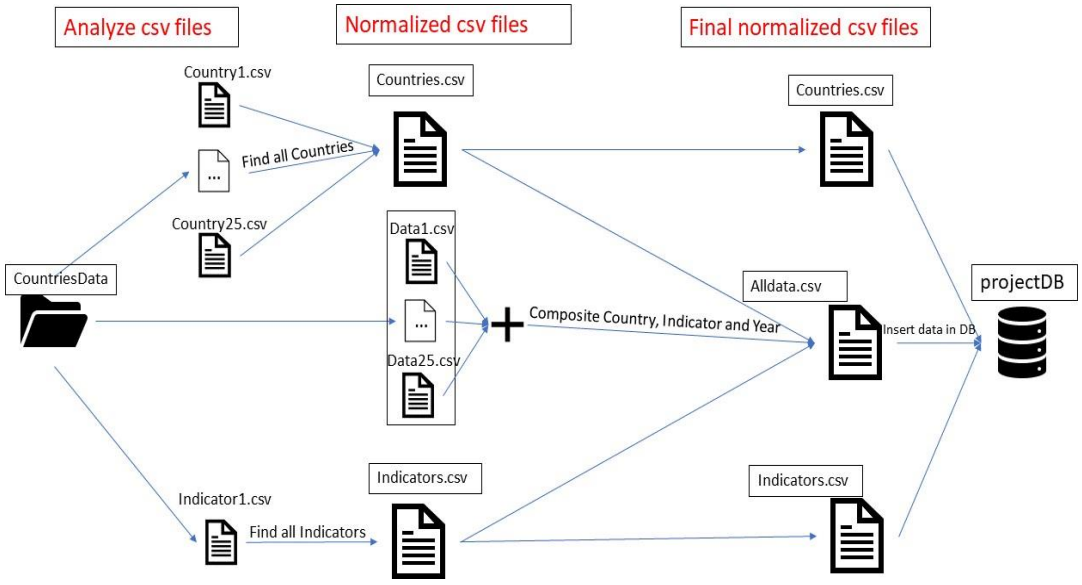
- **ReadyForLoad** : Το οποίο φέρνει τα δεδομένα μας σε συνεπή μορφή για να φορτωθούν στην βάση.
- **loadData** : Το οποίο φορτώνει τα δεδομένα στην βάση , όπως αυτά έχουν δημιουργηθεί από το παραπάνω script.

Αρχικά πήραμε πληροφορίες για την κάθε χώρα ξεχωριστά από τα αρχεία metadata_Country και με την βοήθεια τις βιβλιοθήκης pandas τα συνθέσαμε όλα σε ένα αρχείο Country.csv, το οποίο περιέχει πληροφορίες για όλες τις χώρες.

Στην συνέχεια κάναμε την ίδια διαδικασία για το αρχείο Indicator.csv, το οποίο περιέχει πληροφορίες για όλα τα indicators. Σε αυτήν την περίπτωση δεν χρειάστηκε να ανοίξουμε όλα τα αρχεία metadata_Indicators , αφού όλα περιέχουν τις ίδιες πληροφορίες.

Τέλος δημιουργήσαμε τα αρχεία Country.csv και Indicators.csv ,τα συνδυάσαμε με όλα τα αρχεία που περιέχουν τα βασικά data για κάθε χώρα (API) και δημιουργήσαμε ένα αρχείο AllData, το οποίο περιέχει όλες τις μετρήσεις για κάθε χρονιά, μετρική και χώρα.

Αφού ετοιμάσαμε τα τελικά αρχεία που θα φορτωθούν στην βάση , χρησιμοποιήσαμε το loadData script , στο οποίο αρχικά δημιουργήσαμε μια νέα βάση και 3 tables για να μπουν τα δεδομένα των 3 csv αρχείων μας. Και με την βοήθεια της εντολής "LOAD DATA LOCAL INFILE " φορτώσαμε τα αρχεία μας στην βάση.



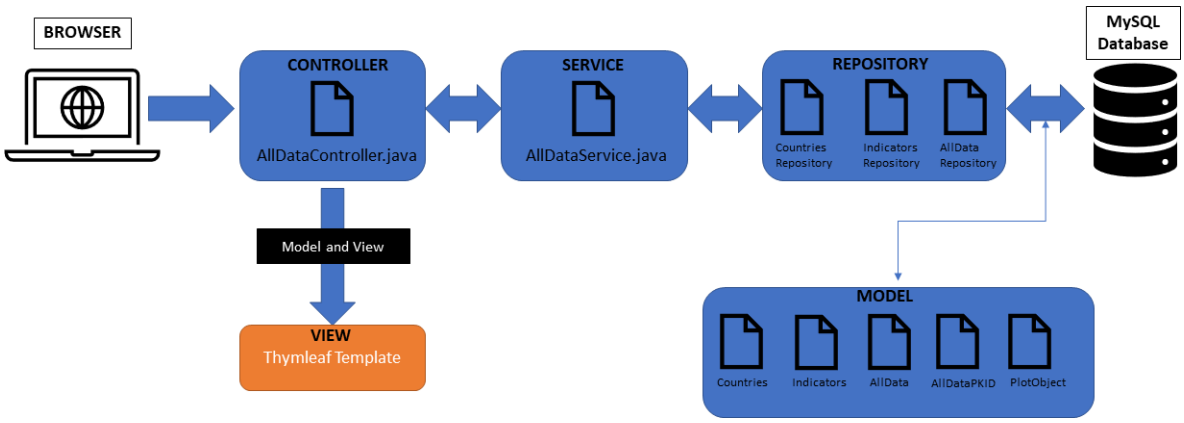
Σχήμα 2-1 ETL diagram

Για να μπορέσει να γίνει όλη η παραπάνω διαδικασία αρκεί μόνο να τρέξουμε το αρχείο main.py.

2.2 ΔΙΑΓΡΑΜΜΑΤΑ ΠΑΚΕΤΩΝ / ΥΠΟΣΥΣΤΗΜΑΤΩΝ ΚΕΝΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ

Όπως φαίνεται και στο σχήμα 2.2 το project βασίζεται στην αρχιτεκτονική της Spring Boot στην οποία το package Controller είναι αρμόδιο να διαχειριστεί τα request από τον browser και με τη βοηθεια των Model and View αντικειμένων να απεικονισει τα

αποτελεσματα των request. Στο service package υλοποιείται η busines logic ,σε συνδιασμό με το Model το οποιο αποτελειται από τα Entity που συνδέονται με τα tables της βάσης μας. Στο Repositry υλοποιούντε όλες οι ερωτήσεις που γίνονται στη βαση και τα αποτελέσματα επιστρέφονται στον controller.



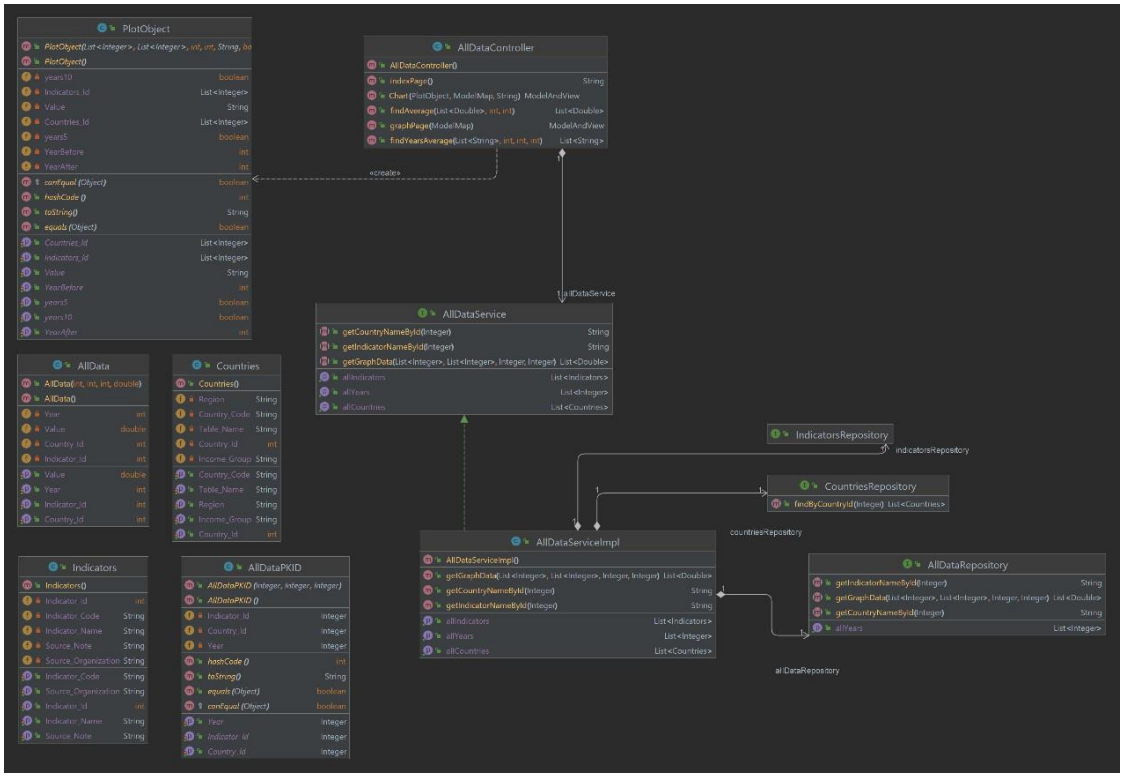
Σχήμα 2-2 Spring boot MVC architecture diagram

2.3 ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΕΩΝ ΚΕΝΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ

Αρχικά στο πακέτο Model έχουμε κατασκευάσει όλες τις οντότητες οι οποίες συνδέονται και αναπαριστούν τα tables της βάσης μας. Όπως αναφέραμε και στην ενότητα 1.1 τα tables της βάσης μας είναι countries,indicators,alldata. Κατά συνέπεια και τα Model είναι Countries, Indicators, AllData με πεδία τα ονόματα των column των αντίστοιχων table. Επίσης κατασκευάσαμε άλλο ένα αντικείμενο Plot Object το οποίο κρατάει τα δεδομένα που εισάγει ο χρήστης για το εκάστοτε plot που έχει επιλέξει.

Ο AllDataController είναι αρμόδιος για τη επικοινωνία του front end με το back end. Αρχικά κάνει κάποιες ερωτήσεις στην βάση(ώστε να πάρει όλα τα διαθέσιμα δεδομένα με τα οποία μπορεί ο χρήστης να φτιάξει διαγράμματα) , περνάει τα αποτελέσματα μέσω του Model and View και με την βοήθεια του thymeleaf δείχνει τα αποτελέσματα στο χρήστη . Στην συνέχεια γεμίζει το PlotObject χρησιμοποιώντας το AllDataService το οποίο κάνει ερωτήσεις στη βάση μέσω των Repositories. Ανάλογα με το ποιο γράφημα θα ζητήσει να δει ο χρήστης, καλείτε και το αντίστοιχο view(BarChart.html, LineChart.html, ScatterChart.html) .Τέλος είναι υλοποιημένες και οι μέθοδοι για των υπολογισμό του μέσου όρου αν το επιλέξει ο χρήστης(5 years, 10 years).

Στο Repository κατασκευάζουμε τα query's που συλλέγουν τα δεδομένα από τη βάση και οπού χρειαστούν καλούνται μέσω του AllDataService αντικειμένου.



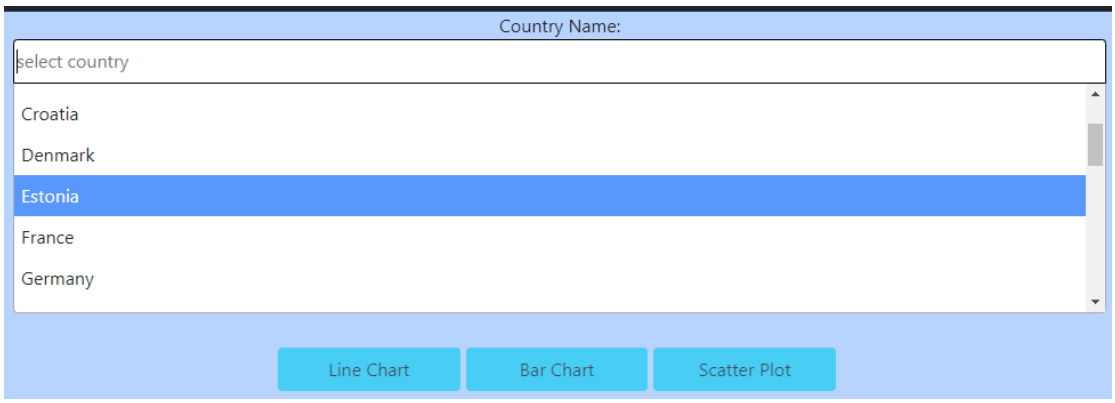
Σχήμα 2-3 Class Diagram

3 ΥΠΟΔΕΙΓΜΑΤΑ ΕΡΩΤΗΣΕΩΝ ΚΑΙ ΑΠΑΝΤΗΣΕΩΝ

Αρχικά για να παρουσιάσουμε στον χρήστη τις χώρες τις οποίες μπορεί να επιλέξει κάνουμε την παρακάτω ερώτηση στη βάση.

```
public List<Countries> getAllCountries() {
    return countriesRepository.findAll();
}
```

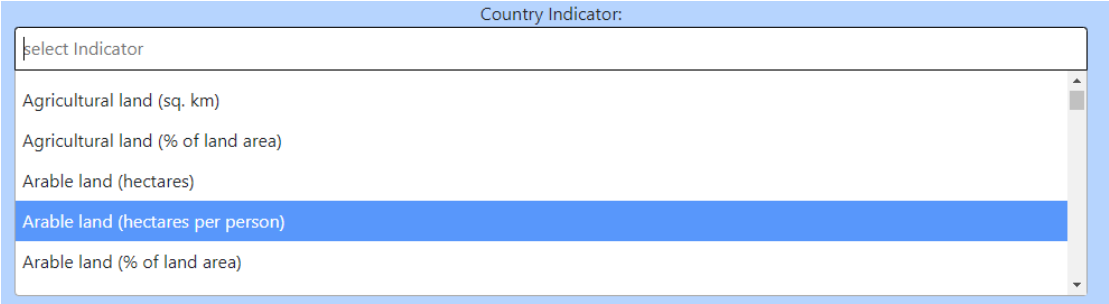
Η παραπάνω εντολή χρησιμοποιεί τη μέθοδο find All του JPARepository Interface η οποία επιστρέφει όλα τα entities του συγκεκριμένου table.Το αποτέλεσμα το εισάγουμε σε ένα dropdown menu για να μπορεί να επιλέξει μια η περισσότερες χώρες ο χρήστης όπως φαίνεται και παρακάτω.



Αντίστοιχη δουλειά γίνεται και για τους δείκτες.

```
public List<Indicators> getAllIndicators() {
    return indicatorsRepository.findAll();
}
```

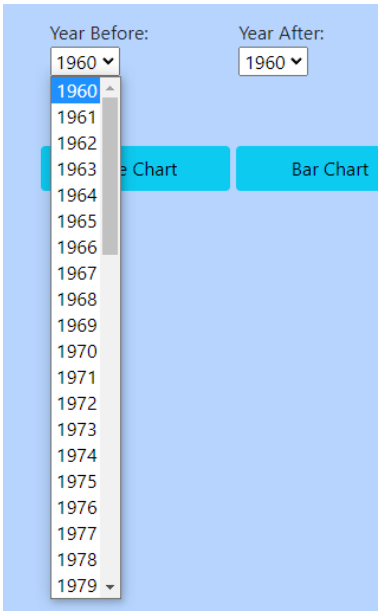
Με αποτέλεσμα :



Όσον αφορά τα years αποθηκεύουμε τη λίστα με τις διαθέσιμες χρονιές σε μια λίστα κάνοντας την παρακάτω ερώτηση στη βάση.

```
@Query(value = "SELECT Year FROM AllData WHERE Country_Id=1 AND
Indicator_Id=1",nativeQuery = true)
```

Και πάλι την παρουσιάζουμε στο χρήστη με ένα dropdown menu.



Για να πάρουμε τις τιμές μιας μέτρησης που έχει επιλέξει ο χρήστης ώστε να την προωθήσουμε στο εκάστοτε γράφημα κάνουμε την παρακάτω ερώτηση στη βάση.

```
@Query(value = "SELECT a.Value\n"+
        " FROM AllData as a\n" +
        "INNER JOIN countries as c ON a.Country_Id=c.Country_Id\n" +
        "INNER JOIN Indicators as i ON a.Indicator_Id=i.Indicator_Id\n" +
        "WHERE a.Country_Id in (:Countries_Id) AND a.Indicator_Id in
(:Indicators_Id)\n" +
        "AND (a.Year BETWEEN :YearBefore AND :YearAfter)",nativeQuery =true)
```

Η παραπάνω ερώτηση επιστρέφει τις τιμές ενός indicator μιας χώρας για το χρονικό εύρος που επέλεξε ο χρήστης σαν μια λίστα από Double.

Τέλος χρησιμοποιήσαμε και δυο ακόμα βοηθητικές ερωτήσεις οι οποίες μας επιστρέφουν το όνομα μιας Country η ενός Indicator με βάση το primary key τους αντίστοιχα.

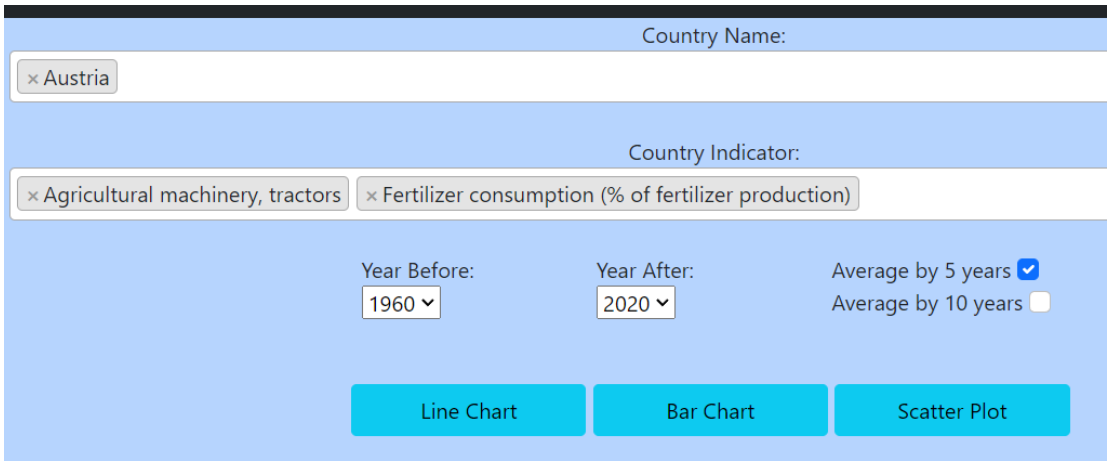
```
@Query(value = "SELECT DISTINCT c.Table_Name FROM AllData as a INNER JOIN
countries as c ON a.Country_Id=c.Country_Id WHERE a.Country_Id =
:countryId",nativeQuery = true)
```

```
@Query(value = "SELECT DISTINCT i.INDICATOR_NAME FROM AllData as a INNER
JOIN indicators as i ON a.Indicator_Id=i.Indicator_Id WHERE a.Indicator_Id =
:indicatorId",nativeQuery = true)
```

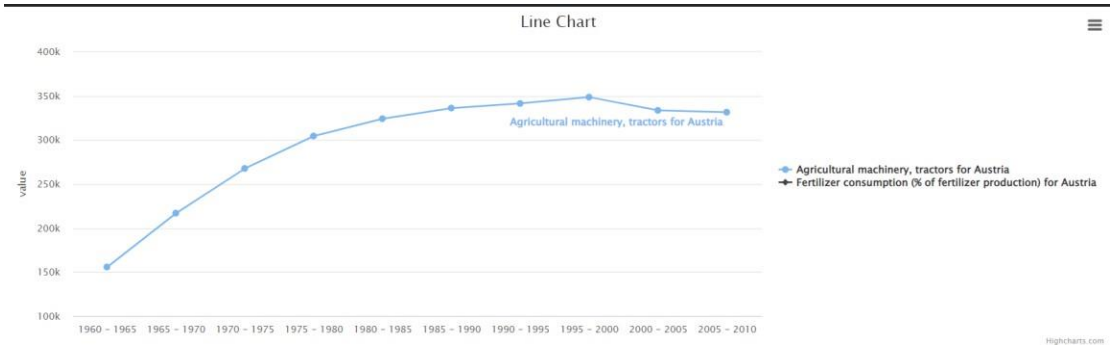
4 ΛΟΙΠΑ ΣΧΟΛΙΑ

Όταν ο χρήστης ζητήσει να δει το διάγραμμα ως μέσο όρο 5 ή 10 ετών δεν λαμβάνουμε καθόλου υπόψιν μας τις null τιμές, έτσι αν τα δεδομένα που επέλεξε να δει έχουν όλα null τιμές , δεν θα δημιουργηθεί το διάγραμμα.

Παραδείγματος χάρη αν ο χρήστης επιλέξει τα παρακάτω



Θα εμφανιστεί το παρακάτω διάγραμμα γιατί η μετρική fertilizer consumption έχει μόνο null τιμές για την συγκεκριμένη χώρα και το συγκεκριμένο διάστημα.



Αν με τις ίδιες επιλογές ο χρήστης θελήσει να δει το scatter plot , δεν θα εμφανιστεί καθόλου το διάγραμμα, αφού ουσιαστικά το scatter plot χρησιμοποιείται για να συγκριθούν 2 μετρικές και σε αυτήν την περίπτωση δεν υπάρχει κάτι να συγκρίνει.

Όσον αναφορά τα κουμπιά για να δημιουργηθούν τα διαγράμματα , όπως και τα κουμπιά για τον μέσο όρο 5 ή 10 χρόνων , γίνονται disable και enable ανάλογα με τις επιλογές του χρήστη :

- **Line Chart :** γίνεται enable όταν έχουν επιλεχτεί μια ή περισσότερες χώρες και indicators.
- **Bar Chart :** γίνεται enable όταν έχουν επιλεχτεί μια ή περισσότερες χώρες και indicators.
- **Scatter Plot :** γίνεται enable όταν έχουν επιλεχτεί μια ή περισσότερες χώρες και ακριβώς 2 indicators.
- **Average by 5 year :** γίνεται enable όταν το διάστημα των ετών που έχει επιλέξει ο χρήστης είναι μεγαλύτερο ή ίσο των 5.
- **Average by 10 year :** γίνεται enable όταν το διάστημα των ετών που έχει επιλέξει ο χρήστης είναι μεγαλύτερο ή ίσο των 10.

Τέλος όλα τα παραπάνω γίνονται enable όταν το year After είναι μεγαλύτερο ή ίσο του year Before.