

ΠΡΟΧΩΡΗΜΕΝΑ ΘΕΜΑΤΑ  
ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΕΦΑΡΜΟΓΩΝ  
ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

---

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ  
ΤΟ ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2022-2023

---

---

ΟΜΑΔΑ

ΒΛΑΧΟΠΟΥΛΟΣ ΛΑΜΠΡΟΣ 1, 2948

ΜΠΟΖ ΝΤΟΥΡΑΝ 2, 2310

---

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΜΑΪΟΣ 2023

## ΙΣΤΟΡΙΚΟ ΠΡΟΗΓΟΥΜΕΝΩΝ ΕΚΔΟΣΕΩΝ

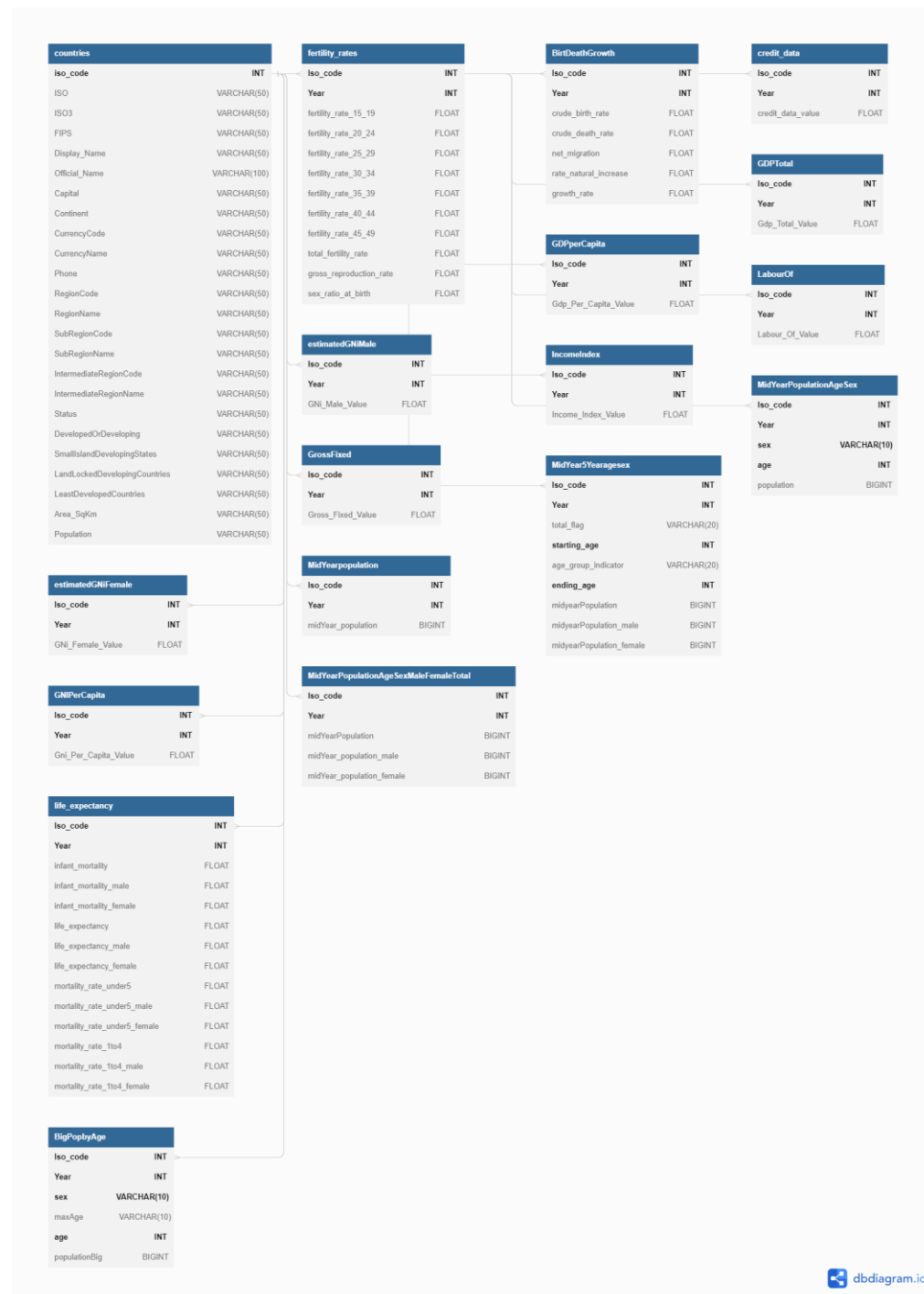
Ημερομηνία	Έκδοση	Περιγραφή	Συγγραφέας
10/5/23	V0	Εισαγωγή πληροφοριών Σχεσιακού μοντέλου της βάσης	Μποζ Ντουράν
20/5/23	V1	Εισαγωγή πληροφοριών σχετικά με τα Scripts	Λάμπρος Βλαχόπουλος
31/5/2023	V2	Ολοκλήρωση της αναφοράς	Λάμπρος Βλαχόπουλος Μποζ Ντουράν

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

# 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

```
CREATE TABLE `countries` (
  `iso_code` INT PRIMARY KEY NOT NULL,
  `ISO` VARCHAR(50),
  `ISO3` VARCHAR(50),
  `FIPS` VARCHAR(50),
  `Display Name` VARCHAR(50),
  `Official Name` VARCHAR(100),
  `Capital` VARCHAR(50),
  `Continent` VARCHAR(50),
  `CurrencyCode` VARCHAR(50),
  `CurrencyName` VARCHAR(50),
  `Phone` VARCHAR(50),
  `RegionCode` VARCHAR(50),
  `RegionName` VARCHAR(50),
  `SubRegionCode` VARCHAR(50),
  `SubRegionName` VARCHAR(50),
  `IntermediateRegionCode` VARCHAR(50),
  `IntermediateRegionName` VARCHAR(50),
  `Status` VARCHAR(50),
  `DevelopedOrDeveloping` VARCHAR(50),
  `SmallIslandDevelopingStates` VARCHAR(50),
  `LandLockedDevelopingCountries` VARCHAR(50),
  `LeastDevelopedCountries` VARCHAR(50),
  `Area_SqKm` VARCHAR(50),
  `Population` VARCHAR(50)
);
----
CREATE TABLE `fertility_rates` (
  `Iso_code` INT NOT NULL,
  `Year` INT NOT NULL,
  `fertility_rate_15_19` FLOAT,
  `fertility_rate_20_24` FLOAT,
  `fertility_rate_25_29` FLOAT,
  `fertility_rate_30_34` FLOAT,
  `fertility_rate_35_39` FLOAT,
  `fertility_rate_40_44` FLOAT,
  `fertility_rate_45_49` FLOAT,
  `total_fertility_rate` FLOAT,
  `gross_reproduction_rate` FLOAT,
  `sex_ratio_at_birth` FLOAT,
  PRIMARY KEY (`Iso_code`, `Year`)
);
----
CREATE TABLE `BirtDeathGrowth` (
  `Iso_code` INT NOT NULL,
  `Year` INT NOT NULL,
  `crude_birth_rate` FLOAT,
  `crude_death_rate` FLOAT,
  `net_migration` FLOAT,
  `rate_natural_increase` FLOAT,
  `growth_rate` FLOAT,
  PRIMARY KEY (`Iso_code`, `Year`)
);
----
CREATE TABLE `credit_data` (
  `Iso_code` INT NOT NULL,
  `Year` INT NOT NULL,
  `credit_data_value` FLOAT,
  PRIMARY KEY (`Iso_code`, `Year`)
);
----
CREATE TABLE `estimatedGNIFemale` (
  `Iso_code` INT NOT NULL,
  `Year` INT NOT NULL,
  `GNI_Female_Value` FLOAT,
  PRIMARY KEY (`Iso_code`, `Year`)
);
```

```
-----
CREATE TABLE `estimatedGniMale` (
    `Iso_code` INT NOT NULL,
    `Year` INT NOT NULL,
    `Gni_Male_Value` FLOAT,
    PRIMARY KEY (`Iso_code`, `Year`)
);
-----
CREATE TABLE `GDPperCapita` (
    `Iso_code` INT NOT NULL,
    `Year` INT NOT NULL,
    `Gdp_Per_Capita_Value` FLOAT,
    PRIMARY KEY (`Iso_code`, `Year`)
);
-----
CREATE TABLE `GDPTotal` (
    `Iso_code` INT NOT NULL,
    `Year` INT NOT NULL,
    `Gdp_Total_Value` FLOAT,
    PRIMARY KEY (`Iso_code`, `Year`)
);
-----
CREATE TABLE `GNIperCapita` (
    `Iso_code` INT NOT NULL,
    `Year` INT NOT NULL,
    `Gni_Per_Capita_Value` FLOAT,
    PRIMARY KEY (`Iso_code`, `Year`)
);
-----
CREATE TABLE `GrossFixed` (
    `Iso_code` INT NOT NULL,
    `Year` INT NOT NULL,
    `Gross_Fixed_Value` FLOAT,
    PRIMARY KEY (`Iso_code`, `Year`)
);
-----
CREATE TABLE `IncomeIndex` (
    `Iso_code` INT NOT NULL,
    `Year` INT NOT NULL,
    `Income_Index_Value` FLOAT,
    PRIMARY KEY (`Iso_code`, `Year`)
);
-----
CREATE TABLE `LabourOf` (
    `Iso_code` INT NOT NULL,
    `Year` INT NOT NULL,
    `Labour_Of_Value` FLOAT,
    PRIMARY KEY (`Iso_code`, `Year`)
);
-----
CREATE TABLE `life_expectancy` (
    `Iso_code` INT NOT NULL,
    `Year` INT NOT NULL,
    `infant_mortality` FLOAT,
    `infant_mortality_male` FLOAT,
    `infant_mortality_female` FLOAT,
    `life_expectancy` FLOAT,
    `life_expectancy_male` FLOAT,
    `life_expectancy_female` FLOAT,
    `mortality_rate_under5` FLOAT,
    `mortality_rate_under5_male` FLOAT,
    `mortality_rate_under5_female` FLOAT,
    `mortality_rate_1to4` FLOAT,
    `mortality_rate_1to4_male` FLOAT,
    `mortality_rate_1to4_female` FLOAT,
    PRIMARY KEY (`Iso_code`, `Year`)
);
```

```
-----
CREATE TABLE `MidYearpopulation` (
  `Iso_code` INT NOT NULL,
  `Year` INT NOT NULL,
  `midYear_population` BIGINT,
  PRIMARY KEY (`Iso_code`, `Year`)
);
-----
CREATE TABLE `MidYear5Yearagesex` (
  `Iso_code` INT NOT NULL,
  `Year` INT NOT NULL,
  `total_flag` VARCHAR(20),
  `starting_age` INT NOT NULL,
  `age_group_indicator` VARCHAR(20),
  `ending_age` INT NOT NULL,
  `midyearPopulation` BIGINT,
  `midyearPopulation_male` BIGINT,
  `midyearPopulation_female` BIGINT,
  PRIMARY KEY (`Iso_code`, `Year`, `starting_age`,
`ending_age`)
);
-----
CREATE TABLE `MidYearPopulationAgeSex` (
  `Iso_code` INT NOT NULL,
  `Year` INT NOT NULL,
  `sex` VARCHAR(10) NOT NULL,
  `age` INT NOT NULL,
  `population` BIGINT,
  PRIMARY KEY (`Iso_code`, `Year`, `sex`,
`age`)
);
-----
CREATE TABLE `BigPopbyAge` (
  `Iso_code` INT NOT NULL,
  `Year` INT NOT NULL,
  `sex` VARCHAR(10) NOT NULL,
  `maxAge` VARCHAR(10) NOT NULL,
  `age` INT NOT NULL,
  `populationBig` BIGINT,
  PRIMARY KEY (`Iso_code`, `Year`, `sex`,
`age`)
);
-----
CREATE TABLE `MidYearPopulationAgeSexMaleFemaleTotal` (
  `Iso_code` INT NOT NULL,
  `Year` INT NOT NULL,
  `midYearPopulation` BIGINT,
  `midYear_population_male`
BIGINT,
  `midYear_population_female`
BIGINT,
  PRIMARY KEY (`Iso_code`,
`Year`)
);
-----
ALTER TABLE `fertility_rates` ADD FOREIGN KEY (`Iso_code`) REFERENCES `countries`
(`iso_code`);
-----
ALTER TABLE `BirtDeathGrowth` ADD FOREIGN KEY (`Iso_code`) REFERENCES `countries`
(`iso_code`);
-----
ALTER TABLE `credit_data` ADD FOREIGN KEY (`Iso_code`) REFERENCES `countries`
(`iso_code`);
-----
ALTER TABLE `estimatedGNIfemale` ADD FOREIGN KEY (`Iso_code`) REFERENCES `countries`
(`iso_code`);
-----
ALTER TABLE `estimatedGNIMale` ADD FOREIGN KEY (`Iso_code`) REFERENCES `countries`
(`iso_code`);
-----
ALTER TABLE `GDPperCapita` ADD FOREIGN KEY (`Iso_code`) REFERENCES `countries`
(`iso_code`);
```

```
-----
ALTER TABLE `GDPTotal` ADD FOREIGN KEY (`Iso_code`) REFERENCES `countries`
(`iso_code`);
-----
ALTER TABLE `GNIPerCapita` ADD FOREIGN KEY (`Iso_code`) REFERENCES `countries`
(`iso_code`);
-----
ALTER TABLE `GrossFixed` ADD FOREIGN KEY (`Iso_code`) REFERENCES `countries`
(`iso_code`);
-----
ALTER TABLE `IncomeIndex` ADD FOREIGN KEY (`Iso_code`) REFERENCES `countries`
(`iso_code`);
-----
ALTER TABLE `LabourOf` ADD FOREIGN KEY (`Iso_code`) REFERENCES `countries`
(`iso_code`);
-----
ALTER TABLE `life_expectancy` ADD FOREIGN KEY (`Iso_code`) REFERENCES `countries`
(`iso_code`);
-----
ALTER TABLE `MidYearpopulation` ADD FOREIGN KEY (`Iso_code`) REFERENCES `countries`
(`iso_code`);
-----
ALTER TABLE `MidYear5Yearagesex` ADD FOREIGN KEY (`Iso_code`) REFERENCES `countries`
(`iso_code`);
-----
ALTER TABLE `MidYearPopulationAgeSex` ADD FOREIGN KEY (`Iso code`) REFERENCES
`countries` (`iso_code`);
-----
ALTER TABLE `BigPopbyAge` ADD FOREIGN KEY (`Iso_code`) REFERENCES `countries`
(`iso_code`);
-----
ALTER TABLE `MidYearPopulationAgeSexMaleFemaleTotal` ADD FOREIGN KEY (`Iso code`)
REFERENCES `countries` (`iso_code`);
-----
```

```
LOAD DATA LOCAL INFILE 'src/main/resources/countriesTable.csv'
INTO TABLE countries
CHARACTER SET latin1
FIELDS OPTIONALLY ENCLOSED BY '"' TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
-----
LOAD DATA LOCAL INFILE 'src/main/resources/AgeFertility_rates.csv'
INTO TABLE fertility_rates
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
-----
LOAD DATA LOCAL INFILE 'src/main/resources/BirthDeathGrowth.csv'
INTO TABLE BirtDeathGrowth
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
-----
LOAD DATA LOCAL INFILE 'src/main/resources/DomesticCredits.csv'
INTO TABLE credit_data
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
-----
LOAD DATA LOCAL INFILE 'src/main/resources/EstimatedGniFemale.csv'
INTO TABLE estimatedGniFemale
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

```
----  
LOAD DATA LOCAL INFILE 'src/main/resources/EstimatedGniMale.csv'  
  INTO TABLE estimatedGniMale  
  FIELDS TERMINATED BY ','  
  LINES TERMINATED BY '\n'  
  IGNORE 1 ROWS;  
----  
LOAD DATA LOCAL INFILE 'src/main/resources/GniPerCapita.csv'  
  INTO TABLE GniPerCapita  
  FIELDS TERMINATED BY ','  
  LINES TERMINATED BY '\n'  
  IGNORE 1 ROWS;  
----  
LOAD DATA LOCAL INFILE 'src/main/resources/GdpPerCapita.csv'  
  INTO TABLE GDPperCapita  
  FIELDS TERMINATED BY ','  
  LINES TERMINATED BY '\n'  
  IGNORE 1 ROWS;  
----  
LOAD DATA LOCAL INFILE 'src/main/resources/GdpTotal.csv'  
  INTO TABLE GDPTotal  
  FIELDS TERMINATED BY ','  
  LINES TERMINATED BY '\n'  
  IGNORE 1 ROWS;  
----  
LOAD DATA LOCAL INFILE 'src/main/resources/IncomeIndex.csv'  
  INTO TABLE IncomeIndex  
  FIELDS TERMINATED BY ','  
  LINES TERMINATED BY '\n'  
  IGNORE 1 ROWS;  
----  
LOAD DATA LOCAL INFILE 'src/main/resources/LabourOf.csv'  
  INTO TABLE LabourOf  
  FIELDS TERMINATED BY ','  
  LINES TERMINATED BY '\n'  
  IGNORE 1 ROWS;  
----  
LOAD DATA LOCAL INFILE 'src/main/resources/GrossFixed.csv'  
  INTO TABLE GrossFixed  
  FIELDS TERMINATED BY ','  
  LINES TERMINATED BY '\n'  
  IGNORE 1 ROWS;  
----  
LOAD DATA LOCAL INFILE 'src/main/resources/Midyear.csv'  
  INTO TABLE MidYearpopulation  
  FIELDS TERMINATED BY ','  
  LINES TERMINATED BY '\n'  
  IGNORE 1 ROWS;  
----  
LOAD DATA LOCAL INFILE  
'src/main/resources/midyearpopulation5yragesexWITHOUTONLYTOTAL.csv'  
  INTO TABLE MidYear5Yearagesex  
  FIELDS TERMINATED BY ','  
  LINES TERMINATED BY '\n'  
  IGNORE 1 ROWS;
```



```
----  
LOAD DATA LOCAL INFILE  
'src/main/resources/midyearpopulationagesex.csv'  
  INTO TABLE MidYearPopulationAgeSex  
  FIELDS TERMINATED BY ','  
  LINES TERMINATED BY '\n'  
  IGNORE 1 ROWS;  
----  
LOAD DATA LOCAL INFILE  
'src/main/resources/midyearpopulationMaleFemaletotal.csv'  
  INTO TABLE MidYearPopulationAgeSexMaleFemaleTotal  
  FIELDS TERMINATED BY ','  
  LINES TERMINATED BY '\n'  
  IGNORE 1 ROWS;  
----  
LOAD DATA LOCAL INFILE 'src/main/resources/LifeExpectansy.csv'  
  INTO TABLE life_expectancy  
  FIELDS TERMINATED BY ','  
  LINES TERMINATED BY '\n'  
  IGNORE 1 ROWS;  
----  
LOAD DATA LOCAL INFILE  
'src/main/resources/com/example/guiproject/Big.csv'  
  INTO TABLE BigPopbyAge  
  FIELDS TERMINATED BY ','  
  LINES TERMINATED BY '\n'  
  IGNORE 1 ROWS;  
----
```

---

## 1.2 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΦΥΣΙΚΟ ΕΠΙΠΕΔΟ

---

---

### 1.2.1 ΡΥΘΜΙΣΗ ΤΩΝ ΠΑΡΑΜΕΤΡΩΝ ΤΟΥ DBMS

---

set global local\_infile=True; //για να μπορούμε να φορτώνουμε τοπικά αρχεία

By default InnoDB.

MySQL session : set all parameters 600 seconds. // για να διαβάζουμε τα αρχεία. Μην κλείνει η σύνδεση και το read αποτύχει.

---

### 1.2.2 ΡΥΘΜΙΣΗ ΤΟΥ ΦΥΣΙΚΟΥ ΣΧΗΜΑΤΟΣ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

---

Δεν κάναμε χρήση Indexes , ούτε views.

Βέβαια με την χρήση ως primary key country\_code (Int), θα ήταν αρκετά εύχρηστο ένα ευρετήριο ως προς αυτό καθώς θα επιτάχυνε την αναζήτηση κάθε φορά. Σχετικά με την χρήση των views αυτό που θα μπορούσαμε να κάνουμε είναι :

Κατά την δημιουργία της βάσης και πριν γίνει κάποιο ερώτημα από το χρήστη θα μπορούσαμε να κάνουμε χρήση ερωτημάτων της μορφής

```
CREATE VIEW country_statistics AS
SELECT c.iso_code, c.Display_Name, c.Continent, c.Population,
       cd.credit_data_value, ef.GNI_Female_Value, em.GNI_Male_Value,
       gpc.Gdp_Per_Capita_Value, gt.Gdp_Total_Value, gni.Gni_Per_Capita_Value,
       gf.Gross_Fixed_Value, ii.Income_Index_Value, lo.Labour_Of_Value
FROM countries c
LEFT JOIN credit_data cd ON c.iso_code = cd.Iso_code
LEFT JOIN estimatedGNIFemale ef ON c.iso_code = ef.Iso_code
LEFT JOIN estimatedGNI_Male em ON c.iso_code = em.Iso_code
LEFT JOIN GDPPerCapita gpc ON c.iso_code = gpc.Iso_code
LEFT JOIN GDPTotal gt ON c.iso_code = gt.Iso_code
LEFT JOIN GNIPerCapita gni ON c.iso_code = gni.Iso_code
LEFT JOIN GrossFixed gf ON c.iso_code = gf.Iso_code
LEFT JOIN IncomeIndex ii ON c.iso_code = ii.Iso_code
LEFT JOIN LabourOf lo ON c.iso_code = lo.Iso_code;
```

Να συνδυάσουμε δηλαδή σε έναν πίνακα στην μνήμη κάποιους πίνακες που θα χρησιμοποιήσουμε περισσότερο στα ερωτήματα. Και όταν ο χρήστης κάνει ερώτηση που η πληροφορία αντλείται σε αυτούς τους πίνακες να εξάγουμε το αποτέλεσμα μέσω αυτού του πίνακα και όχι να εκτελούμε το ερώτημα στην βάση. Με αυτή την σκέψη δεν θα εκτελούσαμε πολλές φορές τα ίδια join. Πχ στο παραπάνω θα μπορούσαμε να αντλήσουμε δεδομένα για όλα τα οικονομικά δεδομένα των χωρών χωρίς να πρέπει να εκτελούμε κάθε φορά join για κάθε διαφορετικό ερώτημα που αφορά οικονομικά στοιχεία.

## 2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΛΟΓΙΣΜΙΚΟΥ

### 2.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΔΟΜΗ ETL

Είναι σημαντικό να επισημάνουμε εδώ ότι όλη η επεξεργασία των αρχείων έως ότου προκύψουν τα εν τέλη φορτώσιμα έγινε μέσω προγραμμάτων στην java και δεν χρησιμοποιήθηκαν ενδιάμεσοι πίνακες στην βάση.

Με το pivotCSV.java παράγουμε αρχείο της μορφής από το countries.csv.

ISO_Code	FIPS	Display_Name	Official_Name
----------	------	--------------	---------------

Αυτό το αρχείο το χρησιμοποιούμε για να κάνουμε map το iso\_code από όλα τα αρχεία.

Σε όλα τα αρχεία δημιουργούμε ένα map με τα στοιχεία

```
countryMap.put(officialName, isoCode);  
countryMap.put(displayName, isoCode);  
countryMap.put(fips, isoCode);  
countryMap.put(officialName+" "+displayName, isoCode);
```

προσπαθούμε δηλαδή να κάνουμε mapping το iso\_code με ένα από οποιοδήποτε χαρακτηριστικό υπάρχει στο εκάστοτε αρχείο. Να σημειωθεί ότι για την περίπτωση του officialName, displayName ελέγχουμε με Jaccard similarity δηλαδή αναζητούμε ομοιότητα.

Τα εν τέλη φορτώσιμα αρχεία έχουν την μορφή:

#### Countries:

```
`iso_code`,  
`ISO`,  
`ISO3`,  
`FIPS`,  
`Display_Name`,  
`Official_Name`,  
`Capital`,  
`Continent`,  
`CurrencyCode`,  
`CurrencyName`,  
`Phone`,  
`RegionCode`,  
`RegionName`,  
`SubRegionCode`,  
`SubRegionName`
```

`IntermediateRegionCode`  
`IntermediateRegionName`  
`Status`,  
`DevelopedOrDeveloping`  
`SmallIslandDevelopingStates`  
`LandLockedDevelopingCountries`  
`LeastDevelopedCountries`  
`Area\_SqKm`  
`Population`

**fertility\_rates:**

`Iso\_code` ,  
`Year`,  
`fertility\_rate\_15\_19`  
`fertility\_rate\_20\_24`  
`fertility\_rate\_25\_29`  
`fertility\_rate\_30\_34`  
`fertility\_rate\_35\_39`  
`fertility\_rate\_40\_44`  
`fertility\_rate\_45\_49`  
`total\_fertility\_rate`  
`gross\_reproduction\_rate`  
`sex\_ratio\_at\_birth`

**BirtDeathGrowth**

`Iso\_code`,  
`Year` INT,  
`crude\_birth\_rate`,  
`crude\_death\_rate`,  
`net\_migration`,  
`rate\_natural\_increase`,  
`growth\_rate`,

**life\_expectancy**

`Iso\_code`,  
`Year`,  
`infant\_mortality`,  
`infant\_mortality\_male`  
`infant\_mortality\_female`,  
`life\_expectancy`,  
`life\_expectancy\_male`,  
`life\_expectancy\_female`,  
`mortality\_rate\_under5`  
`mortality\_rate\_under5\_male`  
`mortality\_rate\_under5\_female`,  
`mortality\_rate\_1to4`,  
`mortality\_rate\_1to4\_male`,  
`mortality\_rate\_1to4\_female`

**MidYearpopulation**

```
`Iso_code`,  
`Year`,  
`midYear_population`
```

**MidYear5Yearagesex**

```
`Iso_code`,  
`Year`,  
`total_flag`,  
`starting_age`,  
`age_group_indicator`,  
`ending_age`,  
`midyearPopulation`,  
`midyearPopulation_male`,  
`midyearPopulation_female`
```

Εδώ σημαντικό να τονίσουμε ότι αφαιρέθηκαν δεδομένα που δεν περιείχαν την απαραίτητη πληροφορία. Δηλαδή κρατήσαμε σε αυτόν εγγραφές που όντως έχουν πληροφορία για τον πληθυσμό ανά ηλικιακό group.

Οι εγγραφές που περιέχουν \* μεταφέρθηκαν σε διαφορετικό αρχείο καθώς είχαν μόνο πληροφορίες για τον συνολικό πληθυσμό και έτσι έγινε διαφορετικός πίνακας .  
(\*\*MidYearPopulationAgeSexMaleFemaleTotal).

**MidYearPopulationAgeSex**

```
Iso_code`,  
`Year`,  
`sex`,  
`age`,  
`population`
```

**BigPopbyAge**

```
`Iso_code`,  
`Year`,  
`sex`,  
`maxAge`,  
`age`,  
`populationBig`
```

Στο αρχείο αυτό παρατηρήσαμε ότι η πληροφορία για τον πληθυσμό κρατούνταν και ως ηλικιακή ομάδα δηλ πληθυσμος για 0 , 1 , κτλ αλλά και ως συνολικό άθροισμα.

Εμείς αυτό που κάναμε είναι ένα pivot τις στήλες γραμμές.

**MidYearPopulationAgeSexMaleFemaleTotal**

```
`Iso_code`,  
`Year`,  
`midYearPopulation`,  
`midYear_population_male`,  
`midYear_population_female`
```

**\*\*Αυτός ο πίνακας περιέχει τις εγγραφές του αρχείου `MidYear5Yearagesex.csv` που έχουν μόνο συνολικά στοιχεία για τον πληθυσμό .**

Για τα οικονομικά δεδομένα δημιουργήθηκε πίνακας για κάθε δεδομένο ξεχωριστά της μορφής

```
`Iso_code`,  
`Year`,  
`Value`
```

Σημαντικό να αναφέρουμε πως ότι δεδομένο σχετικά υπήρχε στα οικονομικά δεδομένα που αφορούσαν Human Development , Regions έγιναν σε διαφορετικά αρχεία πίνακες απλά δεν χρησιμοποιήθηκαν καθώς θεωρήθηκαν αρκετά ελλιπής.

Να πούμε εδώ ότι προτιμήσαμε:

- 1)πίνακα για κάθε οικονομικό στοιχείο
- 2) pivot στηλών σε γραμμές σε αρχεία που αφορούν πληθυσμό

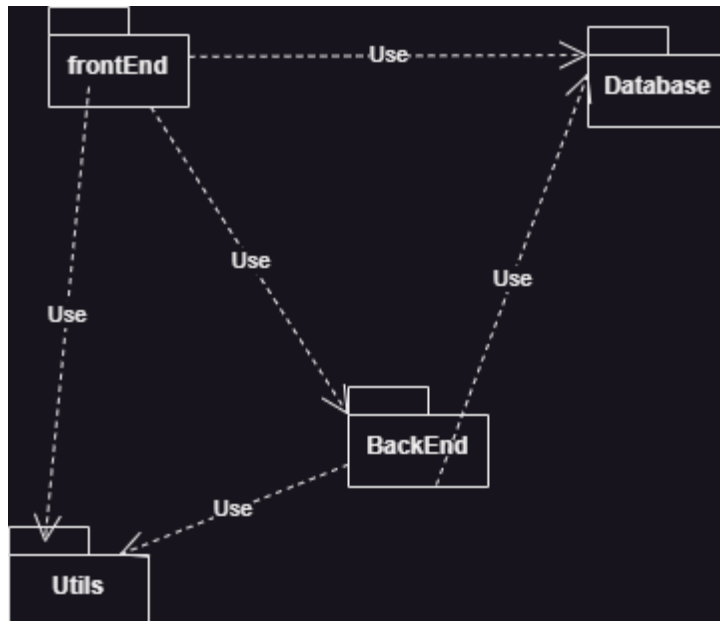
καθώς έτσι απλοποιήσαμε την λειτουργία Load στην βάση.

Ωστόσο , για την ανάκτηση πληροφοριών μέσω ερωτημάτων εκτελούμε πολλαπλά Join και επιβαρύνουμε συνέχεια την βάση με αυτό.

**\*\* Δεν κατανοήσαμε ακριβώς πως εξάγονται τα διαγράμματα Etl για αυτό δεν παραθέσαμε.**

## 2.2 ΔΙΑΓΡΑΜΜΑΤΑ ΠΑΚΕΤΩΝ / ΥΠΟΣΥΣΤΗΜΑΤΩΝ ΚΕΝΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ

Package Diagram:



### 2.3 ΔΙΑΓΡΑΜΜΑ(ΤΑ) ΚΛΑΣΕΩΝ ΚΕΝΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ

## Business Logic Class



Ουσιαστικά οι κλάσεις του FrontEnd δημιουργούν την διεπαφή με το χρήστη και μέσω της κλάσης Listener λαμβάνουμε από αυτόν τις απαραίτητες πληροφορίες που χρειαζόμαστε για να δημιουργήσουμε το sqlQuery και εν τέλει να παράξουμε το διάγραμμα.

Η κλάση Engine είναι ο ενδιάμεσος μεταξύ του υποσυστήματος και της διεπαφής και καλεί τις αντίστοιχες λειτουργίες.

### 3 ΥΠΟΔΕΙΓΜΑΤΑ ΕΡΩΤΗΣΕΩΝ ΚΑΙ ΑΠΑΝΤΗΣΕΩΝ

Το συνολικό ερώτημα που υποστηρίζει το πρόγραμμα μας είναι της μορφής :

```
SELECT t1.Display_Name,t1.year_group,t4.avg_0 ,t4.avg_1 ,t4.avg_5 ,t2.avg_2 ,t1.avg_3 ,t3.avg_4
FROM (SELECT countries.Display_Name,
CEIL(MidYearPopulationAgeSex.Year / 1) * 1 AS year_group,AVG(MidYearPopulationAgeSex.population) AS avg_3
FROM countries
LEFT JOIN MidYearPopulationAgeSex
ON countries.iso_code = MidYearPopulationAgeSex.Iso_code
WHERE countries.Display_Name IN ( 'Greece','Japan' )
AND MidYearPopulationAgeSex.Year BETWEEN 1990 AND 2020
AND MidYearPopulationAgeSex.sex = 'female'
AND MidYearPopulationAgeSex.age = 4
GROUP BY countries.Display_Name, year_group) t1
LEFT JOIN (
SELECT countries.Display_Name,
CEIL(MidYear5Yearagesex.Year / 1) * 1 AS year_group,SUM(MidYear5Yearagesex.midyearPopulation)/1 AS avg_2
FROM countries
LEFT JOIN MidYear5Yearagesex
ON countries.iso_code = MidYear5Yearagesex.Iso_code
WHERE countries.Display_Name IN ( 'Greece','Japan' )
AND MidYear5Yearagesex.Year BETWEEN 1990 AND 2020
AND MidYear5Yearagesex.starting_age >= 0 AND MidYear5Yearagesex.ending_age <= 100
GROUP BY countries.Display_Name, year_group) t2
ON t2.Display_Name = t1.Display_Name AND t2.year_group = t1.year_group LEFT JOIN (
SELECT countries.Display_Name,
CEIL(BigPopbyAge.Year / 1) * 1 AS year_group, SUM(BigPopbyAge.populationBig)/1 AS avg_4
FROM countries
LEFT JOIN BigPopbyAge
ON countries.iso_code = BigPopbyAge.Iso_code
WHERE countries.Display_Name IN ( 'Greece','Japan' )
AND BigPopbyAge.Year BETWEEN 1990 AND 2020
AND BigPopbyAge.age >= 1
AND BigPopbyAge.age <= 10
AND BigPopbyAge.sex IN ('male', 'female')
GROUP BY countries.Display_Name, year_group) t3
ON t3.Display_Name = t2.Display_Name AND t3.year_group = t2.year_group LEFT JOIN (
SELECT countries.Display_Name,
CEIL(life_expectancy.Year / 1) * 1 AS year_group,
AVG(life_expectancy.infant_mortality) AS avg_0 ,
AVG(BirtDeathGrowth.crude_birth_rate) AS avg_1 ,
AVG(GDPTotal.Gdp_Total_Value) AS avg_5 FROM countries
LEFT JOIN life_expectancy ON countries.iso_code = life_expectancy.Iso_code
LEFT JOIN BirtDeathGrowth ON countries.iso_code = BirtDeathGrowth.Iso_code AND BirtDeathGrowth.Year = life_expectancy.Year
LEFT JOIN GDPTotal ON countries.iso_code = GDPTotal.Iso_code AND GDPTotal.Year = life_expectancy.Year
WHERE countries.Display_Name IN ( 'Greece','Japan' ) AND life_expectancy.Year BETWEEN 1990 AND 2020
GROUP BY countries.Display_Name, year_group
) t4
ON t4.Display_Name = t3.Display_Name AND t4.year_group = t3.year_group
ORDER BY t4.avg_0
```

Τα t1,t2,t3,t4 δημιουργούνται δυναμικά ανάλογα από τους πίνακες που συμμετέχουν στο ερώτημα. Αν οι πίνακες του ερωτήματος δεν περιέχουν

- 1) MidYearPopulationAgeSex
- 2) MidYear5Yearagesex
- 3) BigPopbyAge

Δεν έχουν λόγω ύπαρξης και προφανώς δεν δημιουργούνται. Όλοι οι άλλοι πίνακες γίνονται Join στο tNumber που δημιουργείται.

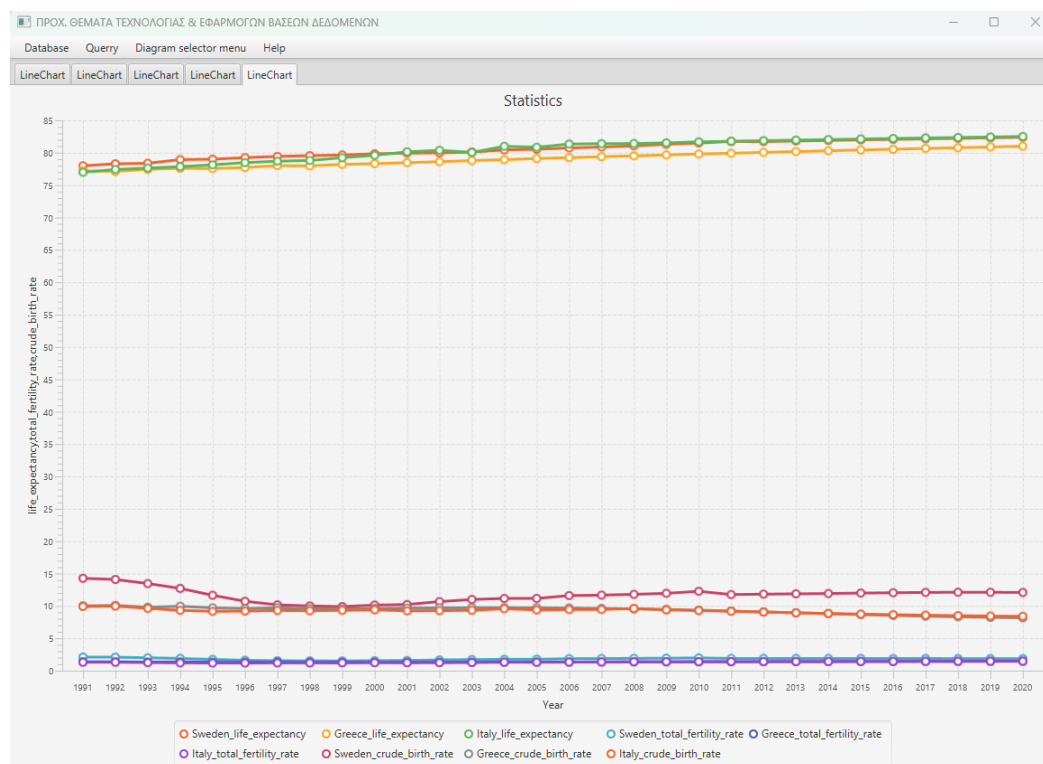


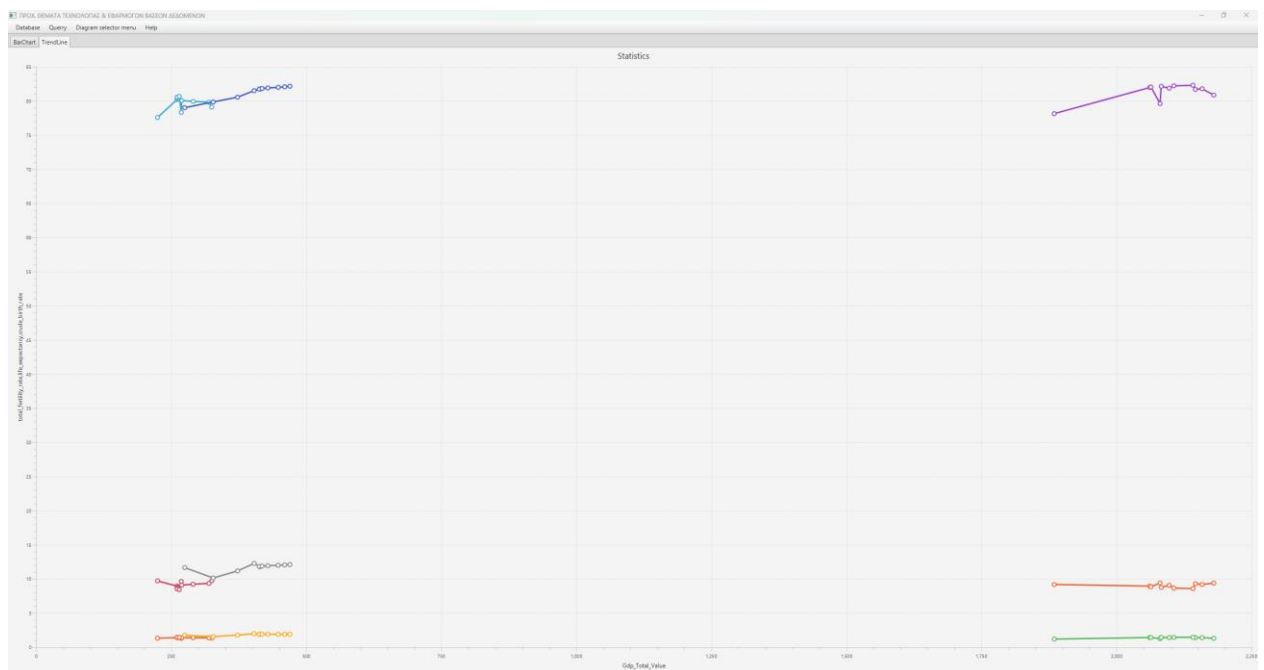
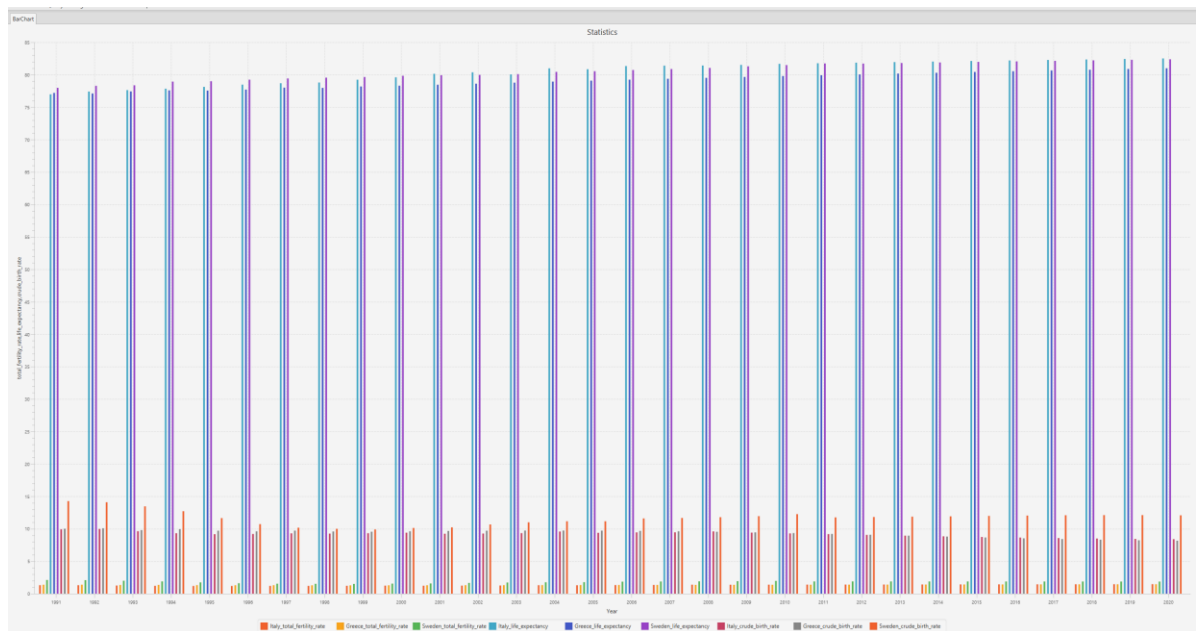
## 4 ΛΟΙΠΑ ΣΧΟΛΙΑ

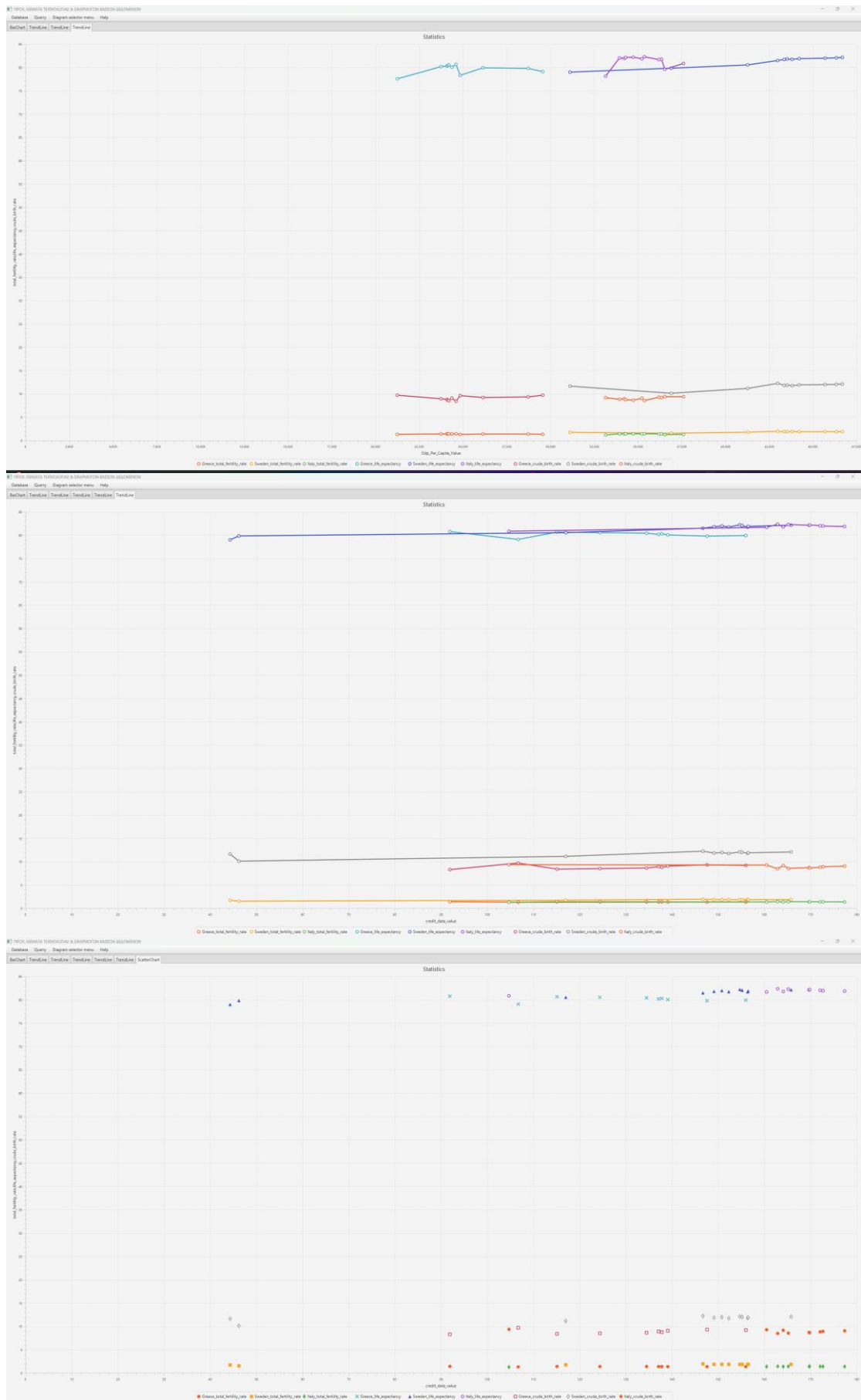
\*\* Critical

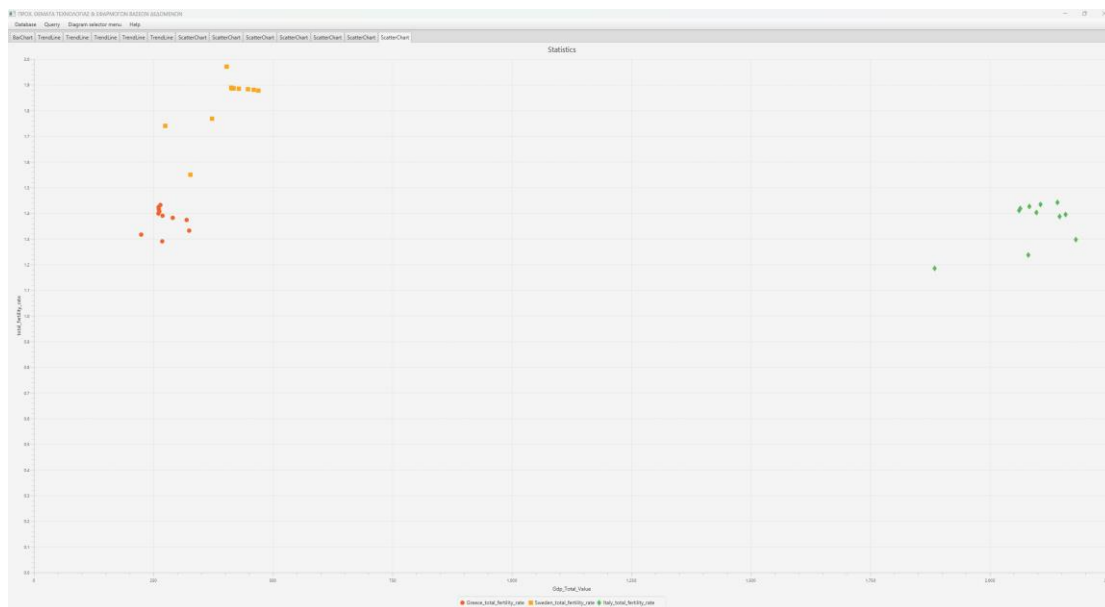
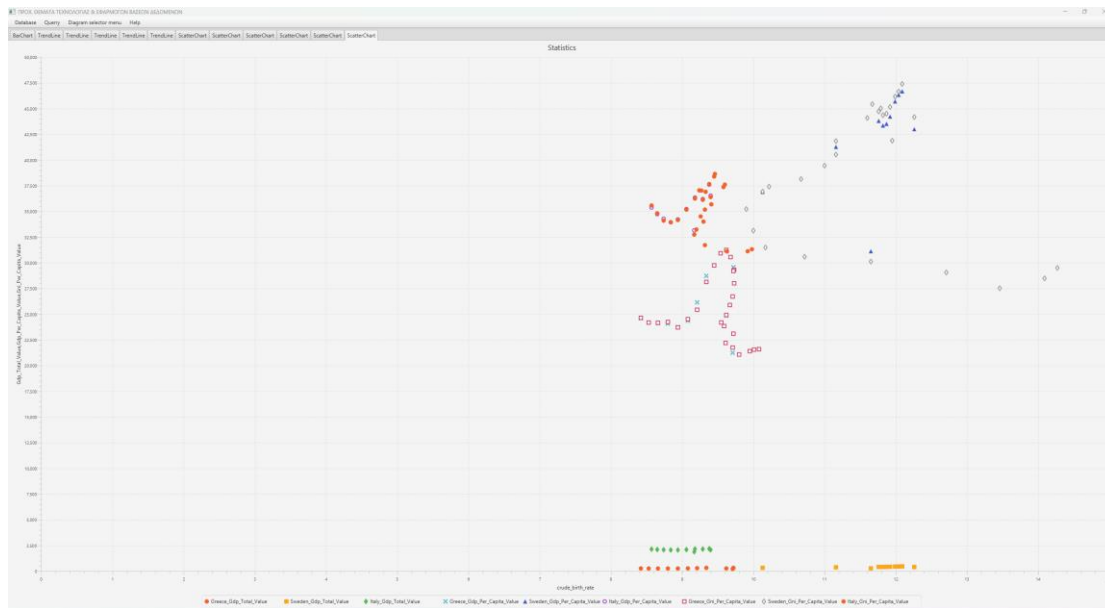
- 1) Τα Scripts τρέχουν ένα ένα και όχι όλα μαζί.
- 2) Επίσης για την παραγωγή των οικονομικών δεδομένων και πως τρέχει το Script θα σας κάνουμε αναφορά και στο βίντεο.
- 3) Επίσης παρατηρήσαμε ότι στα αρχεία υπήρχε η εγγραφή Gaza Strip η οποία δεν υπάρχει στο countries οπότε δε μπορούσαμε να κάνουμε map. Αφαιρέσαμε τις εγγραφές που αφορούν αυτή. Σκεφτήκαμε να περάσουμε χειροκίνητα την εγγραφή αυτή στο countries με iso\_code τύπου Error\_code της μορφής -1 και να κάνουμε map την εγγραφή αυτή με το iso\_code αλλά εν τέλει δεν το υλοποιήσαμε.
- 4) Επίσης στα Scripts περνάμε ελέγχουμε χειροκίνητα αν κάνουμε map με την εγγραφή ως display\_name = "Hong Kong; China (SAR)", καθώς ήταν η μοναδική που δε μπορούσαμε να κάνουμε mapping με βάση το name.
- 5) Το άτομο που εν τέλει είχαμε δηλώσει ομάδα άλλαξε γνώμη για το μάθημα.

Μελέτη Οικονομικών στοιχείων:









Περισσότερα για τα διαγράμματα θα γίνουν αναφορά και στο βίντεο.