ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

**Προχωρημένα Θέματα Τεχνολογίας και Εφαρμογών Βάσεων Δεδομένων**

**ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ για το ακαδημαϊκο έτος** 2021-2022

ΠΗΝΕΛΟΠΗ ΕΛΕΥΘΕΡΙΑΔΗ, 3221

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΜΑΪΟΣ 2022

ΙΣΤΟΡΙΚΟ ΠΡΟΗΓΟΥΜΕΝΩΝ ΕΚΔΟΣΕΩΝ

|  |  |  |  |
| --- | --- | --- | --- |
| Ημερομηνία | Έκδοση | Περιγραφή | Συγγραφέας |
| 22/3/2022 | 1.1 | Πρώτη επαφή με MySQL και τα δεδομένα. | Πηνελόπη Ελευθεριάδη |
| 18/4/2022 | 1.2 | Στήσιμο της βάσης δεδομένων. | Πηνελόπη Ελευθεριάδη |
| 30/4/2022 | 1.3 | Στήσιμο της βάσης δεδομένων και ETL. | Πηνελόπη Ελευθεριάδη |
| 15/5/2022 | 1.4 | Στήσιμο front-end και δημιουργία διαγραμμάτων με χρήση JavaFX. | Πηνελόπη Ελευθεριάδη |
| 20/5/2022 | 1.5 | Διορθώσεις και τελική μορφή εφαρμογής. | Πηνελόπη Ελευθεριάδη |

# βαση δεδομενων

Αρχικά, επέλεξα 25 χώρες και 3 δείκτες (SP, TM, TX).

Οι χώρες που επέλεξα είναι:

Austria, Belgium, Bulgaria, Cyprus, Denmark, Finland, France, Germany, Greece, Hungary, Ireland, Italy, Liechtenstein, Luxembourg, Malta, Monaco, Montenegro, Netherlands, Norway, Poland, Portugal, Romania, Spain, Sweden.

Τις πληροφορίες ανά χώρα τις κατέβασα σε μορφή Excel Spreadsheet. Τροποποίησα τα αρχεία Excel κάθε χώρας ώστε στην καρτέλα Countries τα στοιχεία να είναι στην σωστή σειρά. Συγκεκριμένα, οι στήλες Special Notes και TableName σε κάποιες χώρες ήταν ανάποδα και δημιουργούσαν πρόβλημα. Επίσης , αντικαταστάθηκαν τα κενά πεδία με «\Ν», για να δείξουμε στην MySQL ότι το πεδίο είναι κενό (Νull).

Έπειτα, αποθηκεύσαμε τις καρτέλες Country και Data σε αρχεία της μορφής Text(Tab delimited)(\*.txt) στο folder Uploads του MySQL Workbench.

## Σχεσιακό σχήμα σε λογικό επίπεδο

Diagram

Description automatically generated

Αυτό είναι το τελικό μου schema.

Τα παρακάτω βρίσκονται στο αρχείο ETL\_LoadDataPhase.

ΕΝΤΟΛΕΣ ΔΗΜΙΟΥΡΓΙΑΣ ΠΙΝΑΚΩΝ

CREATE TABLE `incomegroups` (

`ID` int NOT NULL AUTO\_INCREMENT,

`Name` varchar(225) NOT NULL,

PRIMARY KEY (`ID`),

UNIQUE KEY `Name\_UNIQUE` (`Name`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='Income Groups Lookup table';

INSERT INTO `incomegroups`(`Name`)

SELECT `Name` FROM `wdi\_transform`.`incomegroups`;

CREATE TABLE `regions` (

`ID` int NOT NULL AUTO\_INCREMENT,

`Name` varchar(45) NOT NULL,

PRIMARY KEY (`ID`),

UNIQUE KEY `Name\_UNIQUE` (`Name`)

) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `regions`(`Name`)

SELECT `Name` FROM `wdi\_transform`.`regions`;

CREATE TABLE `countries` (

`ID` int NOT NULL AUTO\_INCREMENT,

`Code` varchar(3) NOT NULL,

`Name` varchar(45) NOT NULL,

`RegionID` int DEFAULT NULL,

`IncomeGroupID` int DEFAULT NULL,

`SpecialNotes` varchar(512) DEFAULT NULL,

PRIMARY KEY (`ID`),

UNIQUE KEY `Code\_UNIQUE` (`Code`),

UNIQUE KEY `Name\_UNIQUE` (`Name`),

KEY `RegionID\_idx` (`RegionID`),

KEY `IncomeGroupID\_idx` (`IncomeGroupID`),

CONSTRAINT `FK\_IncomeGroup` FOREIGN KEY (`IncomeGroupID`) REFERENCES `incomegroups` (`ID`),

CONSTRAINT `FK\_Region` FOREIGN KEY (`RegionID`) REFERENCES `regions` (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=34 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `countries`(`Code`,`Name`,`RegionID`, `IncomeGroupID`, `SpecialNotes`)

SELECT `Code`,`Name`,`RegionID`, `IncomeGroupID`, `SpecialNotes` FROM `wdi\_transform`.`countries`;

CREATE TABLE `indicatorcategories` (

`ID` int NOT NULL AUTO\_INCREMENT,

`Name` char(2) NOT NULL,

PRIMARY KEY (`ID`),

UNIQUE KEY `Name\_UNIQUE` (`Name`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `indicatorcategories`(`Name`)

SELECT `Name` FROM `wdi\_transform`.`indicatorcategories`;

/\* Creation and Population of table `indicators`\*/

CREATE TABLE `indicators` (

`ID` int NOT NULL AUTO\_INCREMENT,

`IndicatorCategoryID` int NOT NULL,

`Code` varchar(45) NOT NULL,

`Name` varchar(255) DEFAULT NULL,

PRIMARY KEY (`ID`),

UNIQUE KEY `Code\_UNIQUE` (`Code`),

KEY `indicatorCategoryID\_idx` (`IndicatorCategoryID`),

CONSTRAINT `FK\_IndicatorCategory` FOREIGN KEY (`IndicatorCategoryID`) REFERENCES `indicatorcategories` (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=256 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `indicators`(`IndicatorCategoryID`,`Code`,`Name`)

SELECT `IndicatorCategoryID`,`Code`,`Name` FROM `wdi\_transform`.`indicators`;

CREATE TABLE `years` (

`ID` INT NOT NULL AUTO\_INCREMENT,

`Name` VARCHAR(4) NOT NULL,

`Lustrum` INT NOT NULL,

`Decade` INT NOT NULL,

`Vicennial` INT NOT NULL,

PRIMARY KEY (`ID`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `years`(`Name`,`Lustrum`,`Decade`,`Vicennial`)

SELECT `Name`,`Lustrum`,`Decade`,`Vicennial` FROM `wdi\_transform`.`years`;

CREATE TABLE `ciy` (

`ID` int NOT NULL AUTO\_INCREMENT,

`CountryID` int NOT NULL,

`IndicatorID` int NOT NULL,

`YearID` int NOT NULL,

`Value` double DEFAULT NULL,

PRIMARY KEY (`ID`),

UNIQUE KEY `CIY\_INDEX` (`CountryID`,`IndicatorID`,`YearID`),

KEY `CountryID\_idx` (`CountryID`),

KEY `Indicator\_idx` (`IndicatorID`),

KEY `Year\_idx` (`YearID`),

CONSTRAINT `FK\_Country` FOREIGN KEY (`CountryID`) REFERENCES `countries` (`ID`),

CONSTRAINT `FK\_Indicator` FOREIGN KEY (`IndicatorID`) REFERENCES `indicators` (`ID`),

CONSTRAINT `FK\_Year` FOREIGN KEY (`YearID`) REFERENCES `years` (`ID`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `ciy`(`CountryID`,`IndicatorID`,`YearID`,`Value`)

SELECT `CountryID`,`IndicatorID`,`YearID`,`Value` FROM `wdi\_transform`.`ciy`;

## Σχεσιακό σχήμα σε φυσικο επίπεδο

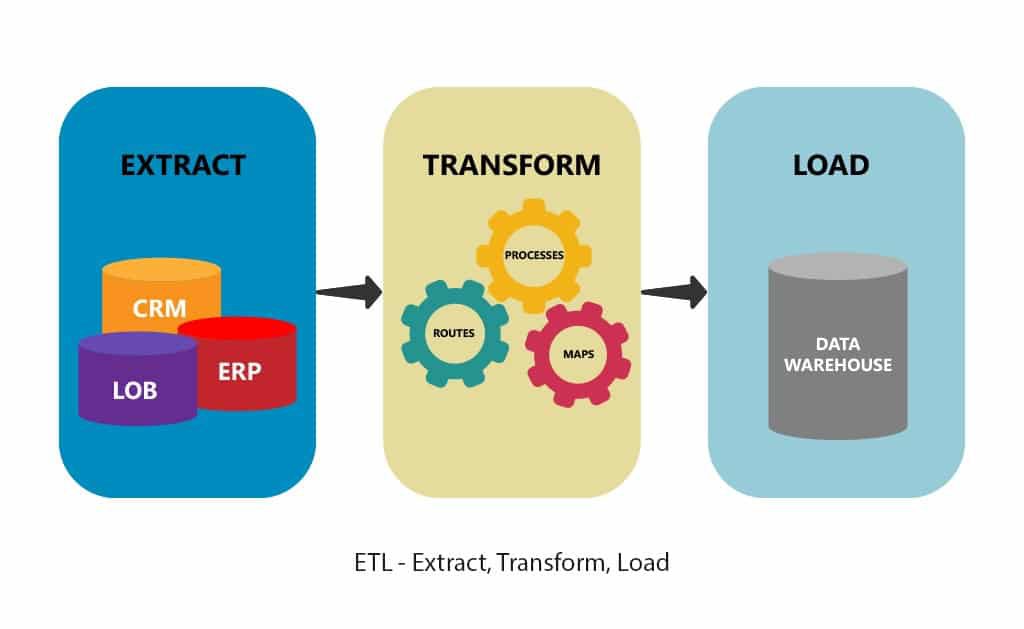
### Ρύθμιση ασφάλειας

Σε όλη την διάρκεια ανάπτυξης της βάσης υπάρχει ένας χρήστης στην MySQL με μοναδικό κωδικό.

Απαραίτητη η αποθηκεύση των καρτέλων Country και Data σε αρχεία της μορφής Text(Tab delimited)(\*.txt) στο folder Uploads του MySQL Workbench.

# αρχιτεκτονικη Λογισμικού

## αρχιτεκτονικη και δομη ETL



Με την λογική που μας υποδεικνύει το ETL Process δημιούργησα τα παρακάτω SQL scripts, ένα για κάθε φάση της μεθόδου ETL:

ETL\_ExtractDataPhase, ETL\_TransformDatPhase, ETL\_LoadDataPhase.

**ETL\_ExtractDataPhase**

Σε αυτό το script δημιουργώ την database wdi\_extract (αφού πρώτα την διέγραψα, στην περίπτωση που ήδη υπάρχει).

Κατόπιν, δημιουργώ τους πίνακες indata\_countries και indata\_data στην παραπάνω βάση δεδομένων. Οι πίνακες αυτοί έχουν πεδία συμβατά με αυτά των καρτελών Country και Data των Text αρχείων.

* indata\_countries (Code, Name, Region, IncomeGroup, SpecialNotes)
* indata\_data (CountryName, CountryCode, IndicatorName, IndicatorCode, 1960,…,2020)

Τέλος, με χρήση της εντολής LOAD DATA INFILE φορτώνω τα δεδομένα ανά χωρά. Το script αγνοεί τις γραμμές που είναι επικεφαλίδες, δηλαδή από τα αρχεία Data τις πρώτες τέσσερεις γραμμές και από τα αρχεία Country την πρώτη γραμμή.

**ETL\_TransformDatPhase**

Όπως και προηγούμενος, δημιουργώ την database wdi\_transform.

Η δουλειά που υλοποιεί αυτό το script είναι η κανονικοποίηση των δεδομένων.

Για τον σκοπό αυτό δημιουργεί τους ακόλουθους lookup πίνακες:

* incomegroups(ID, Name): Δημιουργώ ένα PK(ID) για κάθε income groups που εμφανίζεται στα δεδομένα μου.
* regions(ID, Name) : Δημιουργώ ένα PK(ID) για κάθε region που εμφανίζεται στα δεδομένα μου.
* countries(ID, Code, Name, RegionID, IncomeGroupID): Δημιουργώ ένα PK(ID) για κάθε country που εμφανίζεται στα δεδομένα μου.

Ο πίνακας έχει τα FK RegionID, IncomeGroupID ως αναφορές στα PKs incomegroups .ID και regions.ID.

* indicatorcategories (ID, Name): Δημιουργώ ένα PK(ID) για κάθε indicator category που εμφανίζεται στα δεδομένα μου.

Indicators(ID, IndicatorCategoryID, Code ,Name): Δημιουργώ ένα PK(ID) για κάθε indicator που εμφανίζεται στα δεδομένα μου. Ο πίνακας έχει το FK IndicatorCategoryID ως αναφορά στο PK indicatorcategories.ID .

* Years(ID, Name, Lustrum, Decade, Vicennial): Δημιουργώ ένα PK(ID) για κάθε year(από 1960 μέχρι 2020) που εμφανίζεται στα δεδομένα μου. Για κάθε year αποθηκευονται στα πεδια Lustrum, Decade, Vicennial η πενταετία, δεκαετία και εικοσαετία στην οποία ανήκει.

Επίσης δημιουργεί μια ενδιάμεση, μη κανονικοποιημένη (\_nn: non normalised), μορφή του κύριου πίνακα δεδομένων ciy (country-indicator-year). To script κάνει collumn unpivoting των στηλών 1960…2020 για όλες τις non null τιμές αυτών.

* ciy\_nn(CountryCode, IndicatorCode, YearNo, IndicatorValue): Σε κάθε record αυτού του πίνακα έχω πλέον μια μόνο χρονιά.

Τέλος, δημιουργεί τον κανονικοποιημένο κύριο πίνακα δεδομένων ciy(country-indicator-year)

* ciy (ID, CountryID, IndicatorID, YearID , Value) : Είναι η μορφή του τελικού πίνακα οπού για κάθε χώρα(CountryID), μετρική(IndicatorID) και χρονιά (YearID)έχουμε την αντίστοιχη τιμή της μετρικής(Value). Ο πίνακας έχει το FKs CountryID, IndicatorID, YearID ως αναφορές στα PK countries.ID, indicators.ID, years.ID .

**ETL\_LoadDataPhase:**

Δημιουργώ την database wdi. Αυτή είναι και η τελική μορφή της βάσης.

Πρακτικά φορτώνω τους πίνακες που δημιούργησα στο προηγούμενο script, εκτός από τον ciy\_nn,ο οποίος ήταν βοηθητικός για να μπορέσω να δημιουργήσω τον τελικό πίνακα ciy.

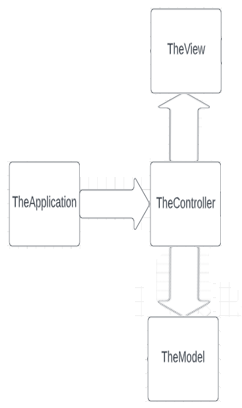
## **Διαγράμματα ΠΑΚΕΤΩΝ / υποσυστημάτων kentρικησ εφαρμογησ**

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε για back-end η ΜySQL και για front-end μια εφαρμογή γραμμένη σε JavaFX με την βοήθεια του Eclipse IDE.

Για την σύνδεση του Eclipse με την MySQL χρησιμοποιήθηκε η mysql-connector.

Για το στήσιμο του user interface χρησιμοποιήθηκε o SceneBuilder και η βιβλιοθήκη JavaFX .

## Διαγράμμα(τα) Κλάσεων κεντρικησ εφαρμογησ



Το pattern που χρησιμοποιήθηκε για την εφαρμογή είναι το MVC (Model View Controller). Συγκεκριμένα έχουμε τις κλάσεις :

1. TheApplication: Αυτή είναι η κλάση της εφαρμογής. Περιέχει την main η οποία καλεί την Application.launch, η οποία με την σειρά της καλεί την start οπού δημιουργείται η Scene και το Stage.
2. TheModel: Η κλάση αυτή έχει τα δεδομένα/status της εφαρμογής. Επίσης συνδέεται με και ανακτά δεδομένα από το back-end .Περιέχει τον constructor αυτής και τα απαραίτητα getters/setters των πεδίων της κλάσης.

Με την συνάρτηση connect2Database() πραγματοποιείται η σύνδεση με την MySQL.

Υπάρχουν, μια σειρά από συναρτήσεις με ονόματα της μορφής geXXXXs() με σκοπό την ανάκτηση όλων των τιμών για την συμπλήρωση(population) τον comboboxes.

Για παράδειγμά, με τον παρακάτω κώδικα παίρνω το αποτέλεσμα του query που ζητά την στήλη Name από τον πίνακα years.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

H βοηθητική συνάρτηση getData(), την οποία καλούν αυτές οι συναρτήσεις, εκτελεί ένα δεδομένο SQL Statement και επιστρέφει το αντίστοιχο ResultSet.

Οι συναρτήσεις getCIYData() και getCIYCollectionData() δημιουργούν δυναμικά, με βάση τις επιλογές του χρήστη, το SQL Statement για την ανάκτηση των δεδομένων του Chart. Θα γίνει πιο κατανοητό παρακάτω , που θα εξηγηθεί η δομή και οι επιλογές του menu. H getCIYCollectionData() είναι ειδική για την ανάκτηση των δεδομένων για Scatter Chart και Bubble Chart.

1. TheView.fxml: Αυτό το αρχείο αντιπροσωπεύει την κλάση View. Το View object είναι το Scene που δημιουργείται από τον FXMLLoader. Με την χρήση του SceneBuilder δημιουργήθηκε το user interface και δόθηκαν ονόματα (fx:Id) σε όλα τα controls (ComboBoxes, Buttons, TextAreas, Chart) και η αντίστοιχη callback συνάρτηση για την διαχείριση των events αυτών, στην κλάση TheController.
2. TheController: Αυτή είναι η κλάση που διαχειρίζεται τα αντικείμενα του UI και συνεργάζεται με τo model ώστε να διαχειριστεί τα δεδομένα και να ενημερώνονται σωστά τα πεδία κατά την χρήση της εφαρμογής. Εμπεριέχει την λογική κάθε εφαρμογής.

Η συνάρτηση initialize κάνει την αρχικοποίησή όλων των controls (Combo Boxes, Buttons etc.)

Ο ρόλος της συνάρτησης populateComboBox είναι η διαχείριση των ComboBoxes.

Οι συναρτήσεις handleXXXXOnAction είναι οι event handlers για κάθε αντικείμενο του UI.

Όταν επιλεχθεί από τον χρήστη η επιλογή [Add Series] ο event handlers handleAddSeriesButton είναι υπεύθυνος, με την βοήθεια του TheModel, να δημιουργήσει το κατάλληλο Chart ανάλογα την επιλογή του χρήστη.

Επίσης, υπάρχουν οι εξής enumerators :

-CharType: έχει τις μορφές chart που θέλουμε.

-DomainFunction: έχει τις επιλογές για το Control Box Group by.

-TimeInterval: έχει τις επιλογές για το Control Box Group by, όσον αφορά τις επιλογές grouping των δεδομένων σε 5ετιες, 10ετιες και 20ετιες.

# Υποδείγματα ερωτήσεων και απαντήσεων

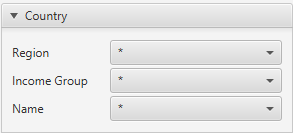
Το Stage της εφαρμογής έχει την εξής μορφή

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

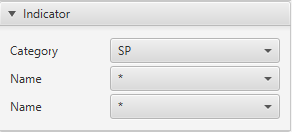
Οι επιλογές menu όπως και το διάγραμμα εμφανίζονται στο ίδιο παράθυρο. Τα μοναδικά pop up παράθυρα που εμφανίζονται είναι τα alert boxes που μας δίνουν μηνύματα λάθους ή πληροφορίας.

Επιλογή Χωράς:



Αρχικά, ο χρήστης επιλεγεί για ποια χωρά θέλει πληροφορίες. Για να μειώσουμε τις επιλογές του και για να μπορεί να μειώσει το εύρος επιλογής, υπάρχουν οι επιλογές Region και Income Group. Στην επιλογή Region μπορεί να επιλέξει αναμεσά σε χώρες της Ευρώπης και της κεντρικής Ασίας και σε χώρες της Μέσης Ανατολής και της Βορείας Αφρικής, ενώ στην Income Group επιλεγεί χώρες με υψηλό ή μεσαίο εισόδημα.

Επιλογή Μετρικής(ων)



Με την ιδιά λογική συνεχίζουμε για την επιλογή Indicator, οπού μειώνουμε τις επιλογές με την επιλογή της κατηγορίας , αναμεσά στις SP,TM,TX. Επιλεγεί ο χρήστης την μετρική που επιθυμεί ,και στην περίπτωση που θέλει Scatter Chart, θα πρέπει να επιλέξει και την δεύτερη.

Επιλογή χρονικής περιόδου

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Έπειτα, επιλεγεί την χρονική περίοδο που θέλει να δει τα αποτελέσματα.

Επιλογή χρονικών διαστημάτων και τιμής.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Και εδώ επιλέγεται η οργάνωση που θέλουμε για τις πληροφορίες μας. Με την επιλογή Year και Value θα εμφανιστούν κανονικά τα διαγράμματα μας με την χρονική περίοδο που έχουμε θέσει και τον indicator που έχουμε επιλέξει. Με την επιλογή 5ετιας, 10ετιας ή 20ετιας έχουμε δυο επιλογές :

-SUM : οπού θα εμφανιστεί το άθροισμά των values ανά 5ετια ή 10ετια ή 20ετια.

-Average : οπού θα εμφανιστεί ο μέσος ορός των values ανά 5ετια ή 10ετια ή 20ετια.

Επιλογή τύπου διαγράμματος και τίτλων διαγράμματος και σειρών

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Εδώ γίνεται η επιλογή του τύπου διαγράμματος και αυτόματα συμπληρώνονται ο τίτλος του διαγράμματος ανάλογα με τον indicator που έχουμε επιλέξει και η χωρά για την οποία ζητάμε τις πληροφορίες. Ο χρήστης μπορεί να επεξεργαστεί τα πεδία τίτλων.

Διαχείριση σειρών

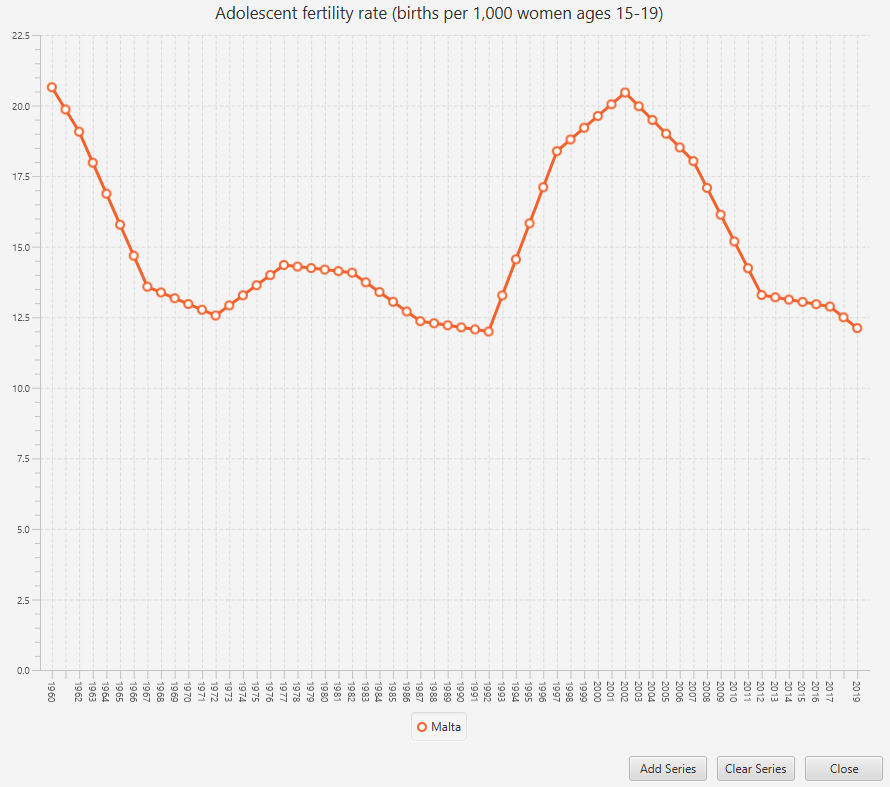


Η επιλογή [Add Series] προσθέτει μια σειρά στο διάγραμμα.

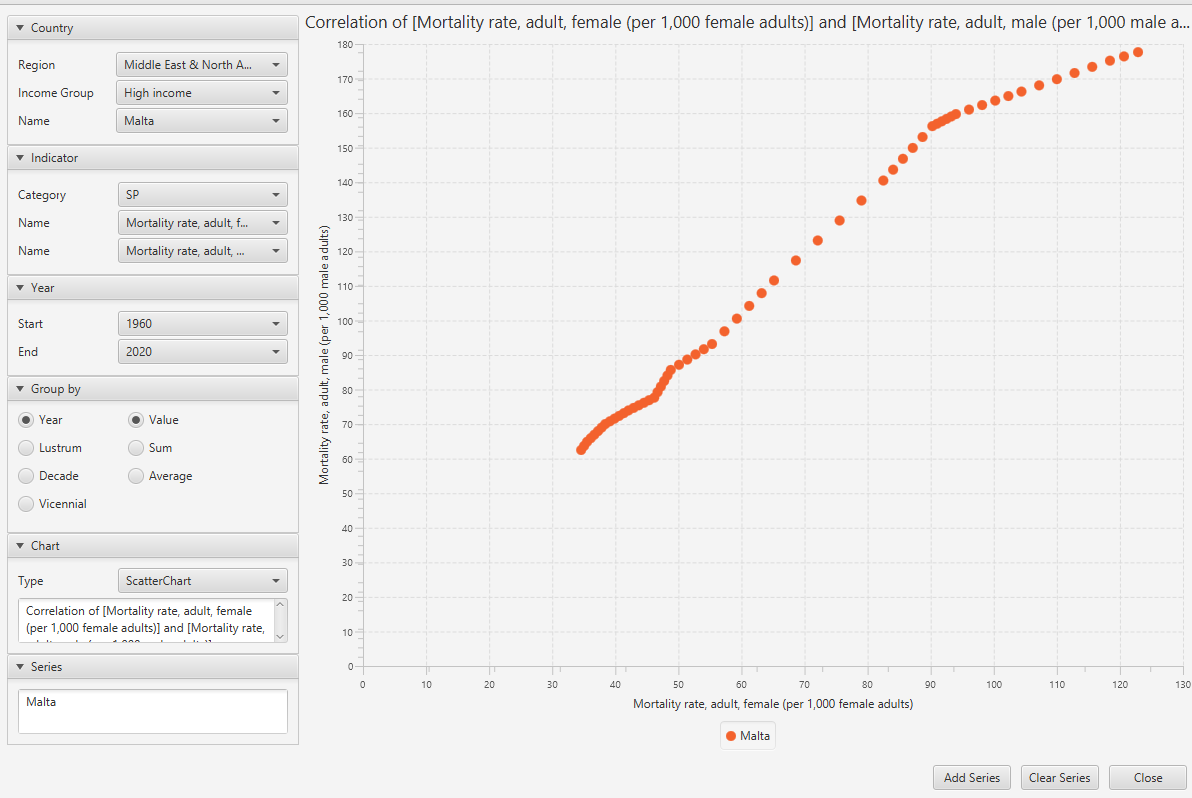
Η επιλογή [Clear Series] καθαρίζει το διάγραμμα.

Έξοδος από την εφαρμογής

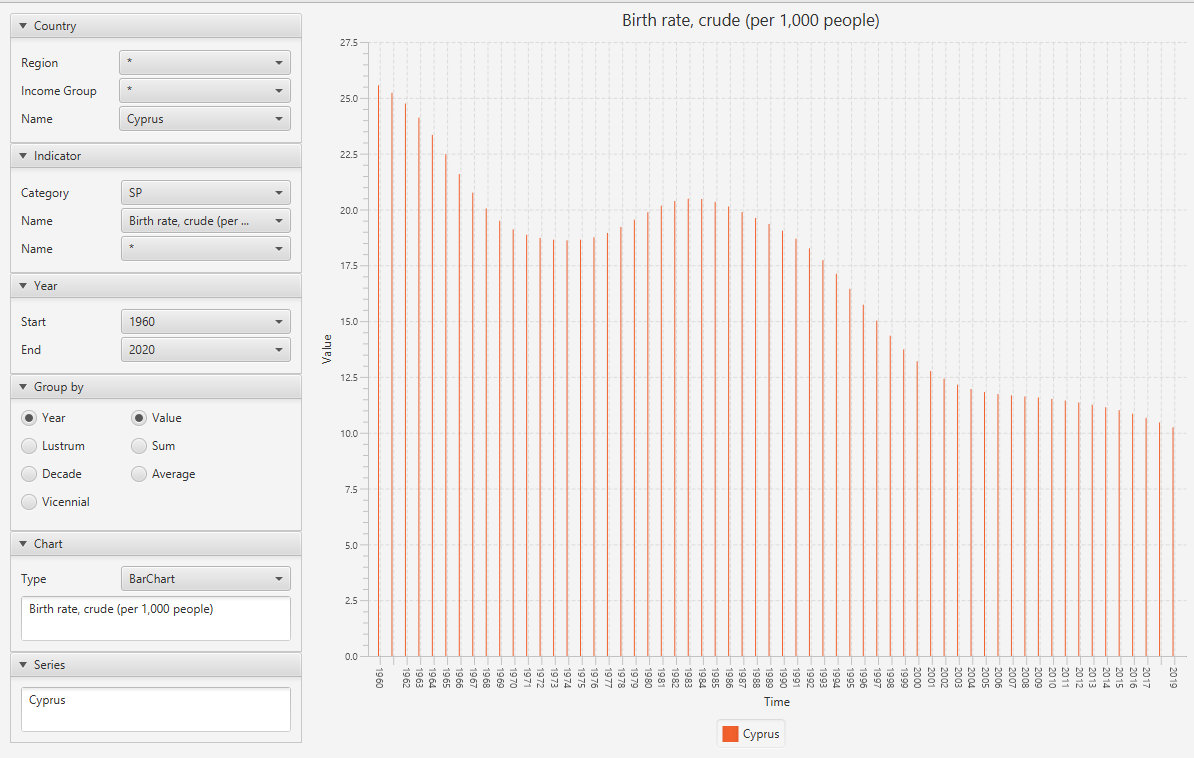
Η επιλογή [Close] και [x] κλείνουν την εφαρμογή.



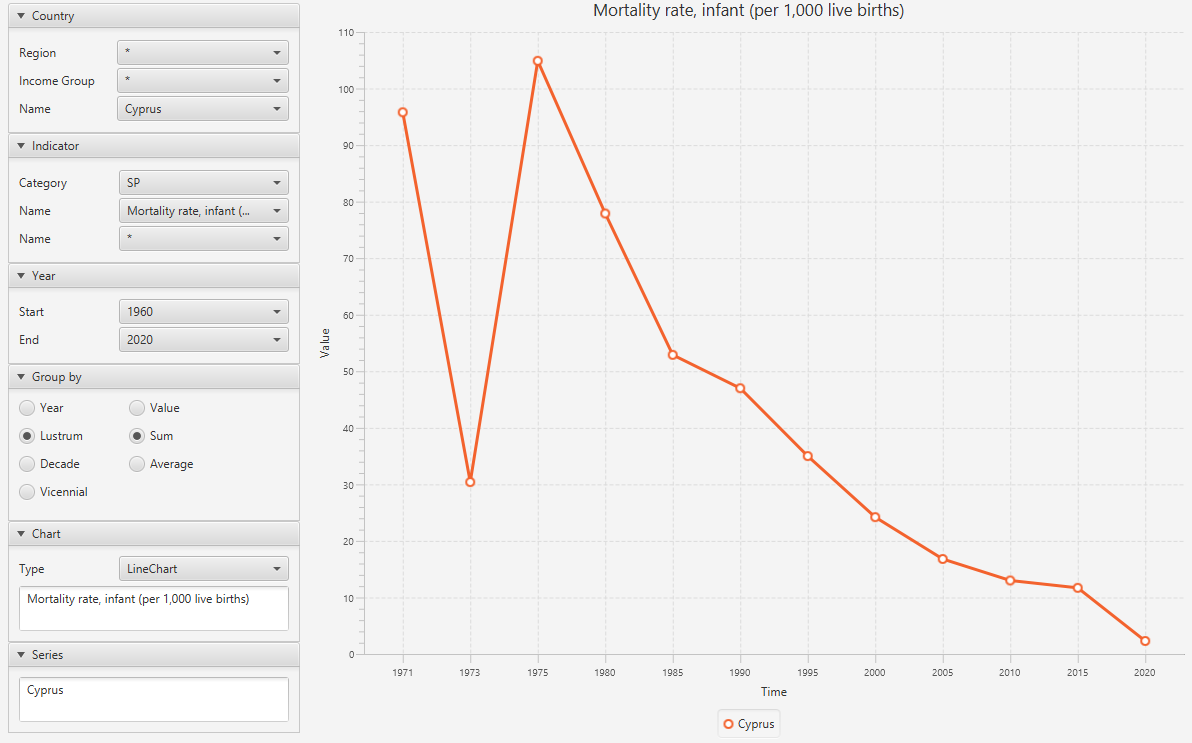
Το αποτέλεσμα ως ένα Line Chart.



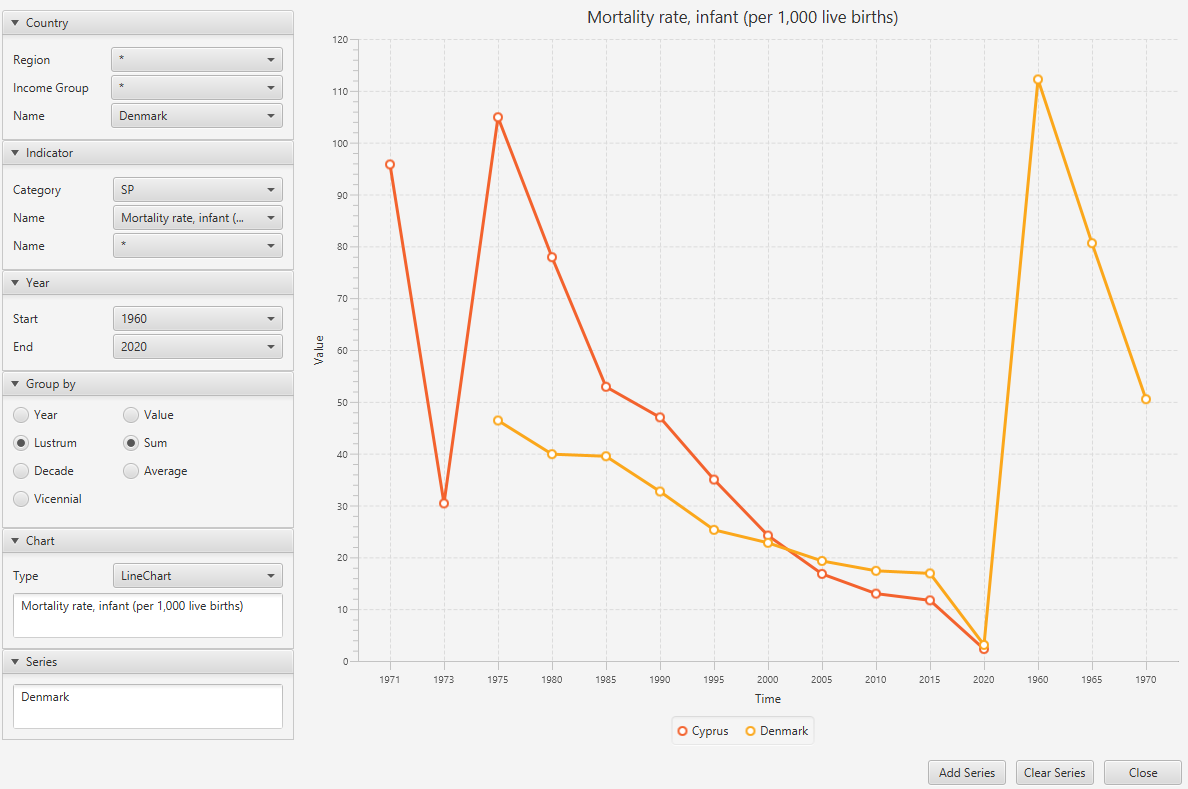
Και εδώ το αποτέλεσμα ενός Scatter Chart(Συσχέτιση δυο μετρικών).



Τα αποτελέσματα ενός Bar Chart με τις επιλογές που βλέπουμε στα αριστερά.



Ένα παράδειγμά παρουσίασης αποτελεσμάτων με τις επιλογή Group by Lustrum(5ετια) και sum.



Και ένα παράδειγμά με την εμφάνιση chart για δυο χώρες για τον ίδιο indicator.

# λοιπά σχόλια

Δεν έχει υλοποιηθεί η επιλογή Bubble Chart. Κατά την επιλογή του εμφανίζεται το αντίστοιχο μήνυμα ενημέρωσης του χρήστη.