

ΠΡΟΧΩΡΗΜΕΝΑ ΘΕΜΑΤΑ
ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΕΦΑΡΜΟΓΩΝ
ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ
ΤΟ ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2019-2020

ΟΜΑΔΑ BEAST

ΧΡΗΣΤΟΣ ΓΡΗΓΟΡΙΑΔΗΣ,
2860

ΔΗΜΗΤΡΗΣ ΤΣΑΚΤΣΙΡΑΣ, 2838

ΧΡΗΣΤΟΣ ΣΟΥΡΗΣ, 2819

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

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

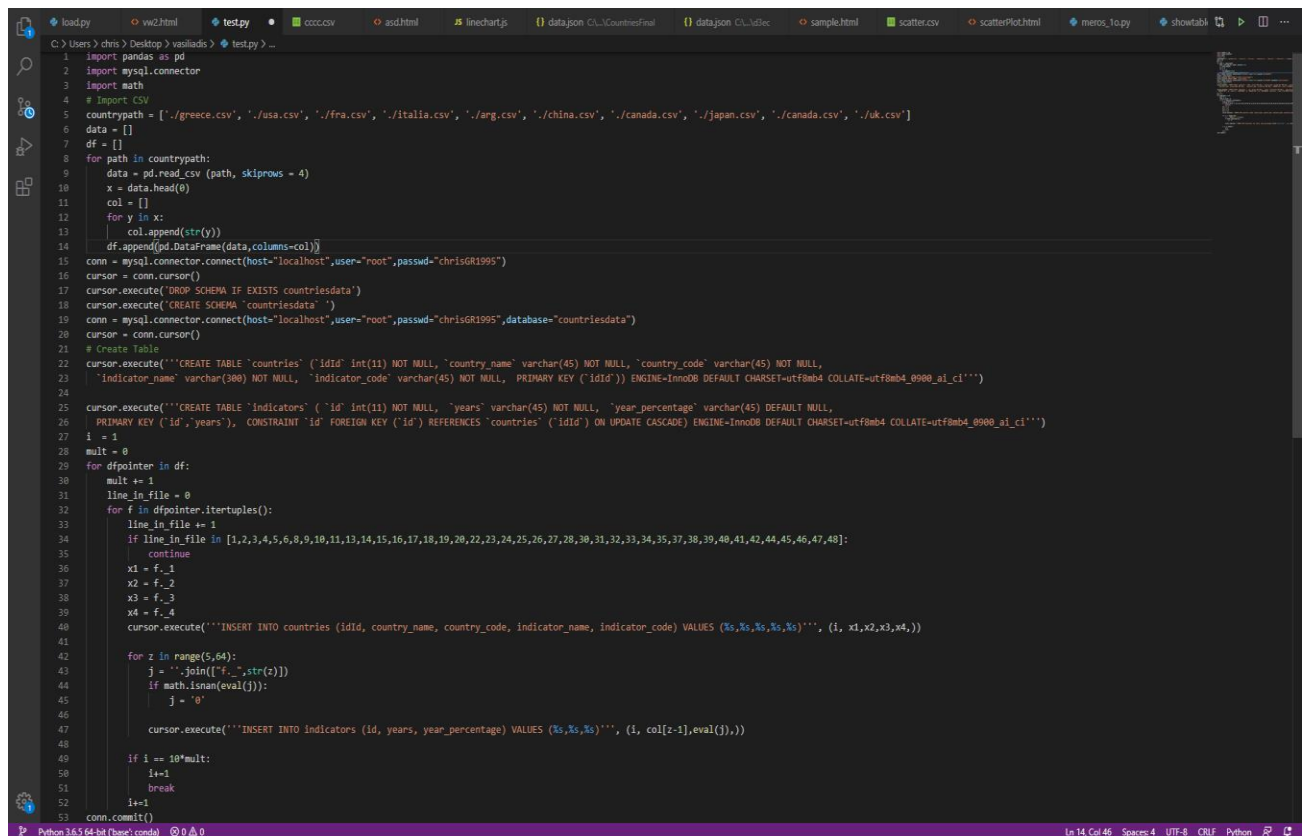
Εδώ περιγράφονται τα σχήματα της βάσης (ή βάσεων, αν είναι παραπάνω από μία) δεδομένων που χρησιμοποιούνται στο project.

1.1 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΛΟΓΙΚΟ ΕΠΙΠΕΔΟ

Παρακάτω παραθέτουμε ένα python script το οποίο δημιουργεί ένα νέο σχήμα (countriesdata.db) , δημιουργεί τα απαραίτητα tables και τα γεμίζει με τα δεδομένα που φορτώνουμε από τα csv αρχεία.

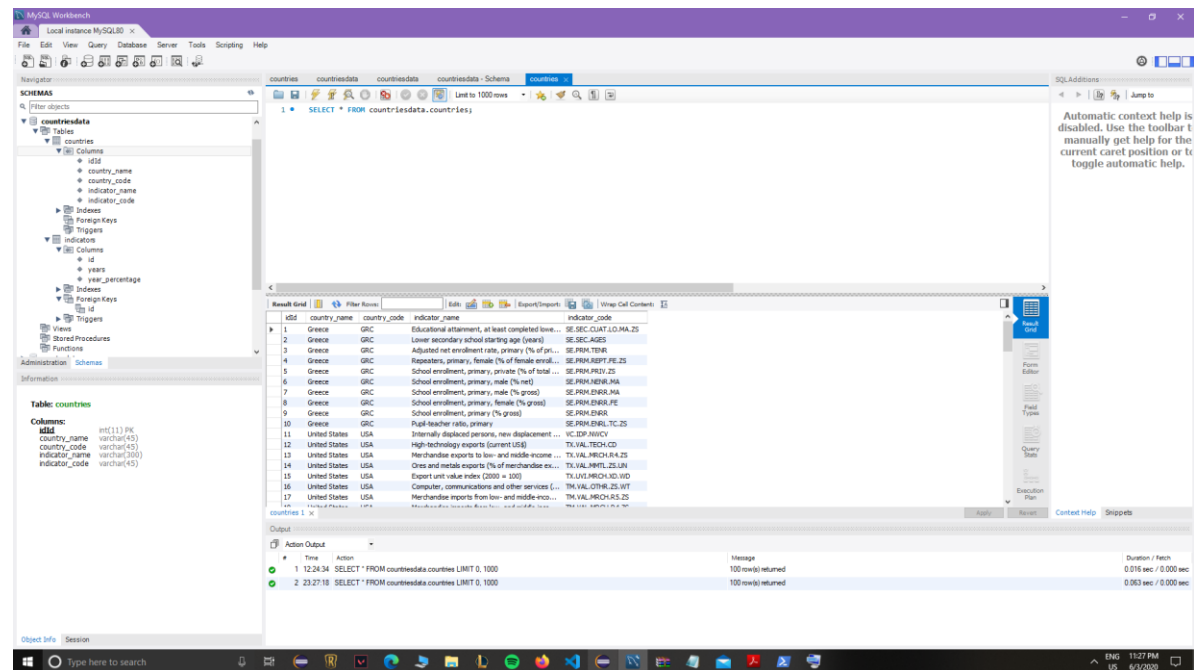
Απαραίτητες προϋποθέσεις:

- Να τρέχει ο sql server
- Να υπάρχει το mysql module της python (pip install mysql-connector-python)
- Σωστά στοιχεία για την σύνδεση με το server (γραμμή 15 και 19)

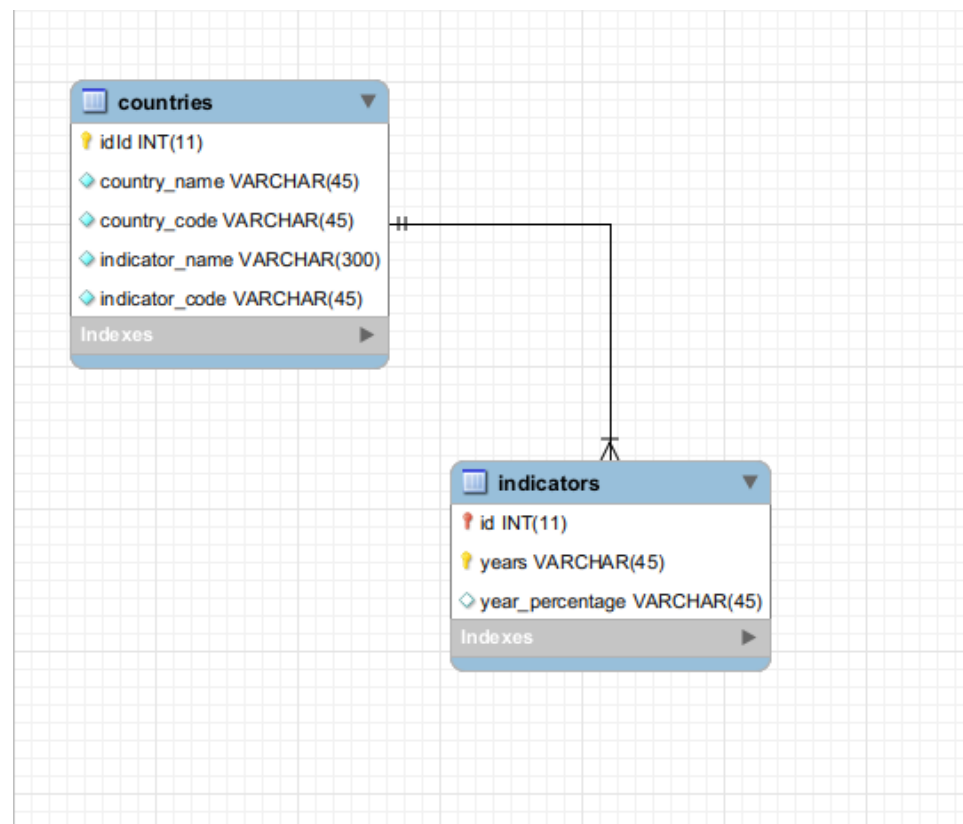


```
1 import pandas as pd
2 import mysql.connector
3 import math
4 # Import CSV
5 countrypath = ['./greece.csv', './usa.csv', './fra.csv', './italia.csv', './arg.csv', './china.csv', './canada.csv', './japan.csv', './canada.csv', './uk.csv']
6 data = []
7 df = []
8 for path in countrypath:
9     data = pd.read_csv(path, skiprows = 4)
10    x = data.head(0)
11    col = []
12    for y in x:
13        col.append(str(y))
14    df.append(pd.DataFrame(data, columns=col))
15 conn = mysql.connector.connect(host="localhost", user="root", passwd="chrisGR1995")
16 cursor = conn.cursor()
17 cursor.execute("DROP SCHEMA IF EXISTS countriesdata")
18 cursor.execute("CREATE SCHEMA 'countriesdata' ")
19 conn = mysql.connector.connect(host="localhost", user="root", passwd="chrisGR1995", database="countriesdata")
20 cursor = conn.cursor()
21 # Create Table
22 cursor.execute("""CREATE TABLE 'countries' ('id' int(11) NOT NULL, 'country_name' varchar(45) NOT NULL, 'country_code' varchar(45) NOT NULL,
23 'indicator_name' varchar(300) NOT NULL, 'indicator_code' varchar(45) NOT NULL, PRIMARY KEY ('id')) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci""")
24
25 cursor.execute("""CREATE TABLE 'indicators' ('id' int(11) NOT NULL, 'years' varchar(45) NOT NULL, 'year_percentage' varchar(45) DEFAULT NULL,
26 PRIMARY KEY ('id','years'), CONSTRAINT 'id' FOREIGN KEY ('id') REFERENCES 'countries' ('id') ON UPDATE CASCADE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci""")
27 i = 1
28 mult = 0
29 for dfpointer in df:
30     mult += 1
31     line_in_file = 0
32     for f in dfpointer.itertuples():
33         line_in_file += 1
34         if line_in_file in [1,2,3,4,5,6,8,9,10,11,13,14,15,16,17,18,19,20,22,23,24,25,26,27,28,30,31,32,33,34,35,37,38,39,40,41,42,44,45,46,47,48]:
35             continue
36             x1 = f._1
37             x2 = f._2
38             x3 = f._3
39             x4 = f._4
40             cursor.execute("INSERT INTO countries (idid, country_name, country_code, indicator_name, indicator_code) VALUES (%s,%s,%s,%s,%s)", (i, x1,x2,x3,x4))
41
42             for z in range(5,64):
43                 j = ''.join(["f._",str(z)])
44                 if math.isnan(eval(j)):
45                     j = '0'
46
47                 cursor.execute("INSERT INTO indicators (id, years, year_percentage) VALUES (%s,%s,%s)", (i, col[z-1],eval(j)))
48
49             if i == 10*mult:
50                 i+=1
51                 break
52             i+=1
53 conn.commit()
```

Όταν τελειώσει το script το workbench θα είναι ως εξής:



1.2 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΦΥΣΙΚΟ ΕΠΙΠΕΔΟ



1.2.1 ΡΥΘΜΙΣΗ ΤΩΝ ΠΑΡΑΜΕΤΡΩΝ ΤΟΥ DBMS

Στο python script όταν δημιουργούμε τα tables στο τέλος της ερώτησης προσθέτουμε `ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci` (γραμμή 25)

storage engine, memory allocation (of various kinds), ...

1.2.2 ΡΥΘΜΙΣΗ ΤΟΥ ΦΥΣΙΚΟΥ ΣΧΗΜΑΤΟΣ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

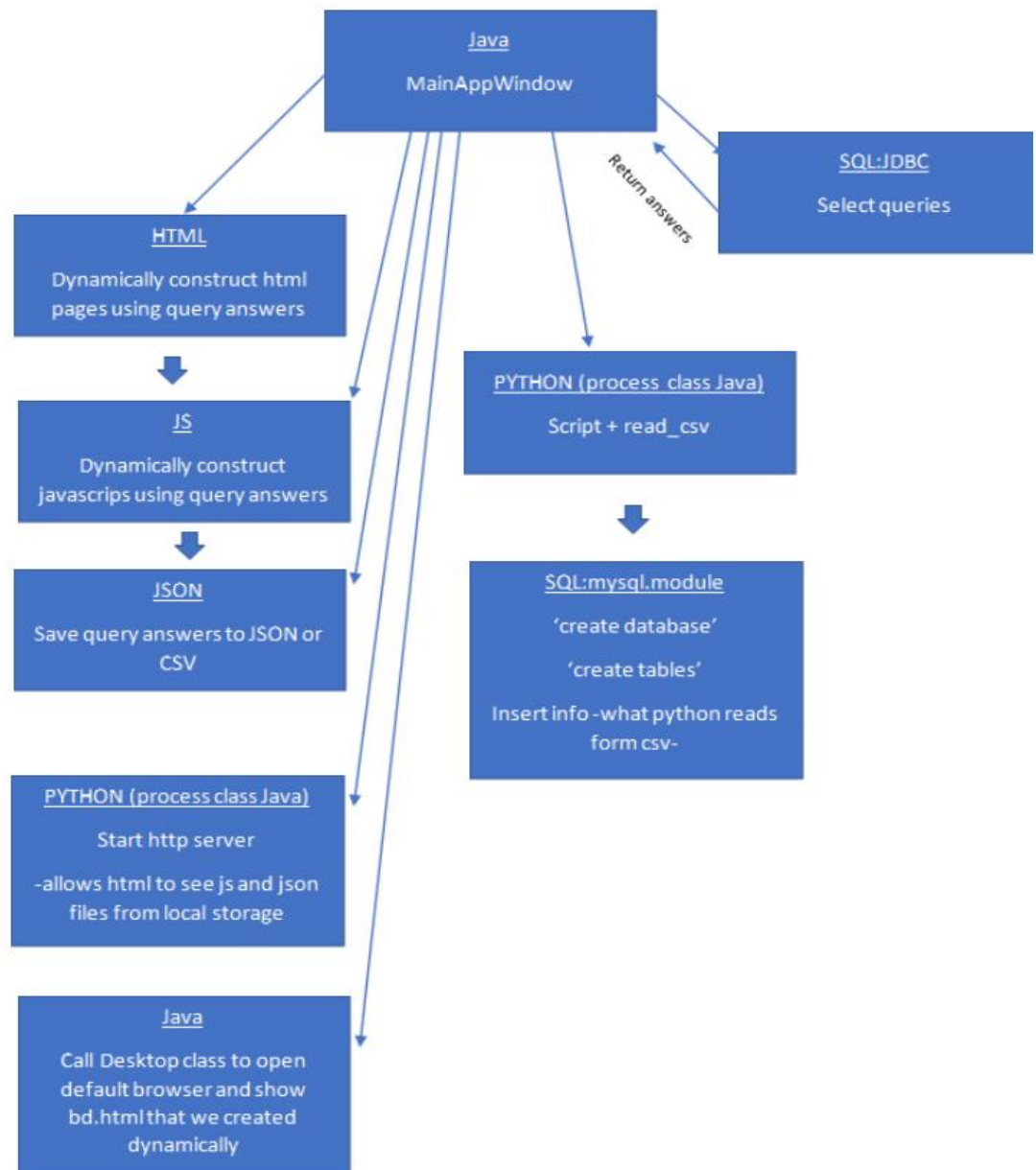
ορισμός πιθανών ευρετηρίων (indexes), όψεων (views) που είναι υλοποιημένες ή μη, αλλαγές στο σχήμα των πινάκων για λόγους απόδοσης, κλπ. Τεκμηριώστε τα παραπάνω με βάση τα πλάνα από τα ερωτήματα που καθυστερούν ή με βάση την εσωτερική δομή του κώδικα και της δυσκολίας συγγραφής του.

1.2.3 ΡΥΘΜΙΣΗ ΑΣΦΑΛΕΙΑΣ

ορισμός δικαιωμάτων καταχώρησης ή ανάκτησης δεδομένων σε διαφορετικούς ρόλους και χρήστες του συστήματος.

2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΛΟΓΙΣΜΙΚΟΥ

2.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΔΟΜΗ ETL



Αρχικά πρέπει να φέρουμε τα δεδομένα μέσα στη βάση μας για περαιτέρω επεξεργασία. Εδώ καταγράφεται η αρχιτεκτονική της ETL διαδικασίας (είτε μέσω εργαλείου, είτε μέσω των όποιων scripts προεπεξεργασίας και φόρτωσης δεδομένων φτιάξετε).

Είναι σημαντικό η διαδικασία να καταγραφεί με ακρίβεια στις λεπτομέρειες. Μπορείτε να χρησιμοποιήσετε UML-based / BPMN / ETL-specific formalisms για τη διαγραμματική απεικόνιση. Δείτε τις σχετικές οδηγίες στο συνοδευτικό κείμενο στο web site του μαθήματος.

Λογικά, για ότι είναι αυτοματοποιημένο, αρκεί να πείτε τι ρυθμίσεις χρειάζονται.

Αν έχετε όμως παρεμβάσεις που γίνονται manually, πρέπει να καταγραφούν επίσης οι λεπτομέρειες.

2.2 ΔΙΑΓΡΑΜΜΑΤΑ ΠΑΚΕΤΩΝ / ΥΠΟΣΥΣΤΗΜΑΤΩΝ ΚΕΝΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ

Το διάγραμμα για τα υποσυστήματα / πακέτα του λογισμικού που κατασκευάσατε ως κεντρική εφαρμογή επερώτησης. Ο στόχος είναι να φανεί η high-level αρχιτεκτονική του συστήματος, χωρίς λεπτομέρειες των επί μέρους κλάσεων.

2.3 ΔΙΑΓΡΑΜΜΑ(ΤΑ) ΚΛΑΣΕΩΝ ΚΕΝΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ

Αν η ανάπτυξη γίνει αντικειμενοστρεφώς, εδώ μπαίνουν τα διαγράμματα κλάσεων + ο σχολιασμός της κεντρικής εφαρμογής. Αλλιώς μπαίνουν διαγράμματα που διευκολύνουν την κατανόηση της εσωτερικής αρχιτεκτονικής του λογισμικού (π.χ., component/ deployment diagrams / ...)

3 ΥΠΟΔΕΙΓΜΑΤΑ ΕΡΩΤΗΣΕΩΝ ΚΑΙ ΑΠΑΝΤΗΣΕΩΝ

Select Chart:

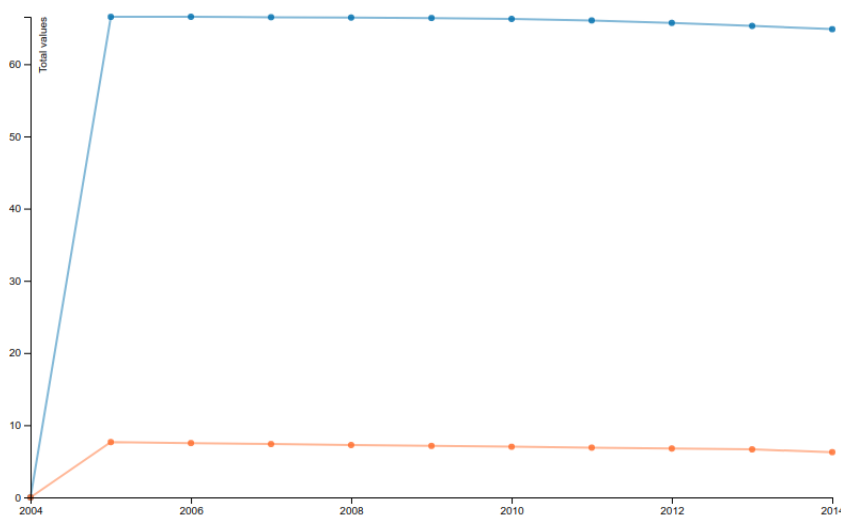
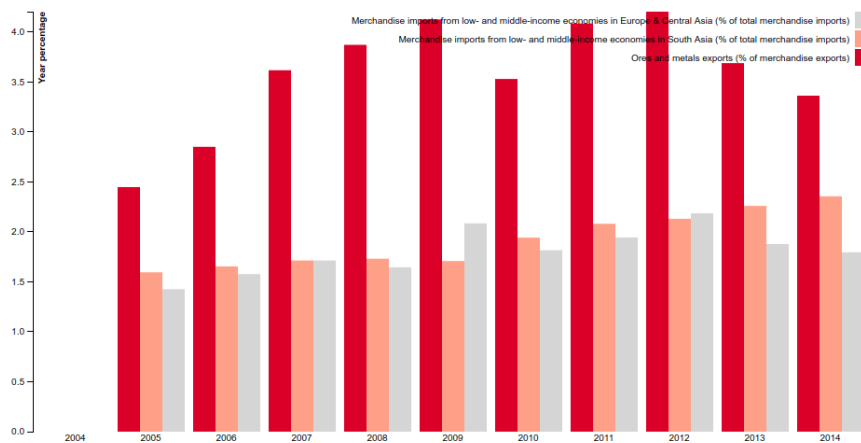
Select Countries:

Select Indicators:

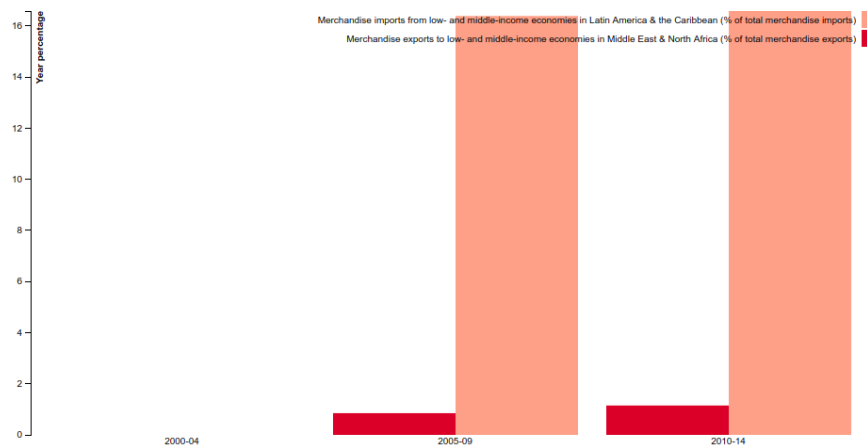
Select Presentation:

Select Year From:

Select Year To:



Ανά 5 χρόνια:



4 ΤΕΚΜΗΡΙΩΣΗ ΚΑΙ ΛΟΙΠΑ ΣΧΟΛΙΑ