

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΥΕ030/
ΠΛΕ045

ΠΡΟΧ. ΘΕΜΑΤΑ ΤΕΧΝΟΛΟΓΙΑΣ & ΕΦΑΡΜΟΓΩΝ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ
Π. Βασιλειάδης

ΑΝΟΙΞΗ 2025

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΑΣΚΗΣΗ

Ημερομηνία Παράδοσης: Τρίτη 27-05-2025, 16.59

Η προγραμματιστική άσκηση για το μάθημα είναι **υποχρεωτική** και αφορά τη σχεδίαση, υλοποίηση και ρύθμιση ενός ολοκληρωμένου πληροφοριακού συστήματος. Η εργαστηριακή άσκηση προσφέρει **3 μονάδες** στον τελικό βαθμό του μαθήματος. Φυσικά, πρέπει να πιάσετε τουλάχιστον τη βάση στην εργασία, όπως και στο διαγώνισμα. Σε περιπτώσεις εξαιρετικών εργασιών, η επίδοση επιβραβεύεται με bonus που μπορεί να φτάσει ως και μία μονάδα στον τελικό βαθμό.

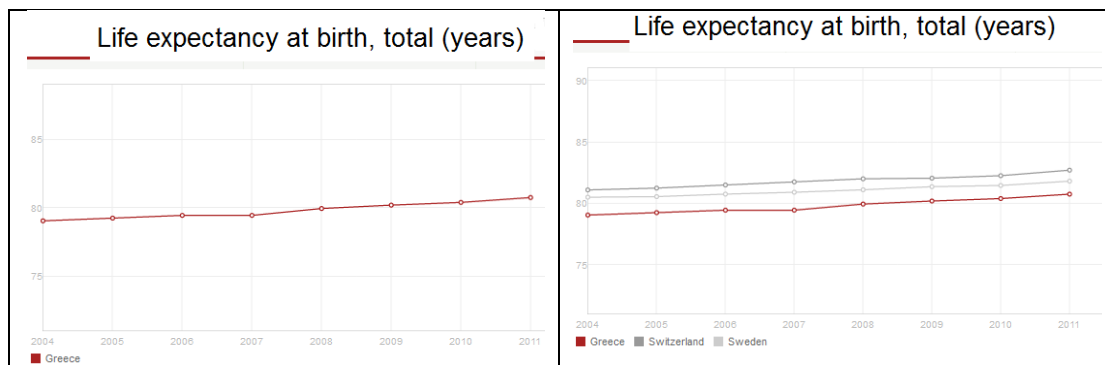
Οι προθεσμίες είναι ιερές.

Είναι υποχρεωτικό να υλοποιήσετε τουλάχιστον ένα σύστημα με σχεσιακό back-end business logic, και γραφική διαπροσωπεία καθώς και μια τελική αναφορά.

Η προγραμματιστική άσκηση του μαθήματος ασχολείται με το πρόβλημα (α) της **ενοποίησης** και (β) της **οπτικοποίησης δεδομένων**. Ο στόχος των τεχνικών ενοποίησης είναι να εντάξουν πληροφορίες με διαφορετική δομή και κωδικοποίηση σε ένα ενιαίο σχήμα, ώστε να είναι ενιαία επερωτήσιμες. Ο στόχος των τεχνικών οπτικοποίησης είναι να δώσουν στον χρήστη την πληροφορία με τρόπο που αναδεικνύει οπτικά ιδιότητες, τάσεις και πρότυπα που βρίσκονται κρυμμένα στα δεδομένα. *Why bother? Κυρίως, γιατί ζούμε σε μια εποχή που έχουμε όλο και πιο πολλά δεδομένα γύρω μας, και γίνεται όλο και πιο δύσκολο να τα αξιοποιήσουμε, ρωτώντας τα. Οι απαντήσεις στις ερωτήσεις πλέον δεν αρκούν: στους χρήστες πρέπει να παρουσιάζονται και ενδιαφέρουσες ιδιότητες εντός των δεδομένων.*

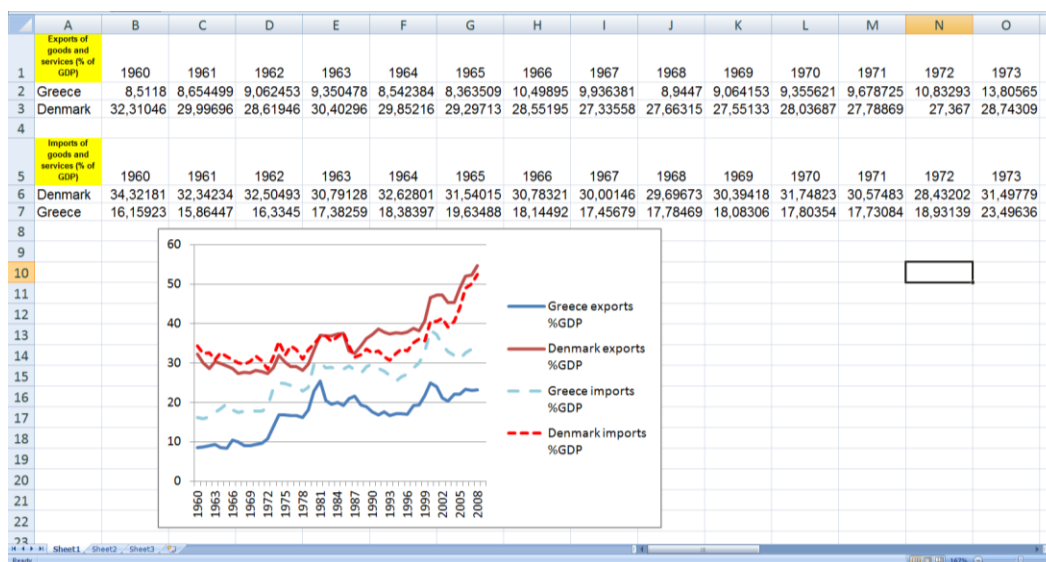
Γενικά περί Οπτικοποίησης

Timelines / trendlines. Αν θέλουμε να δείξουμε την εξέλιξη ενός ή περισσότερων δεικτών στο χρόνο, το πιο συχνά χρησιμοποιούμενο μέσο είναι οι timelines. Ο χρόνος απεικονίζεται στον άξονα των x και το μετρούμενο μέγεθος στον άξονα των y . Αν αντί για χρόνο έχουμε άλλο ποσό στον άξονα των x (π.χ., ο πληθυσμός μιας χώρας, η έκτασή της κλπ.) τότε εμπίπτουμε στη γενικότερη κατηγορία των trendlines. Εδώ έχουμε προσδόκιμο ζωής για (α) μία και (β) 3 χώρες στο χρόνο.

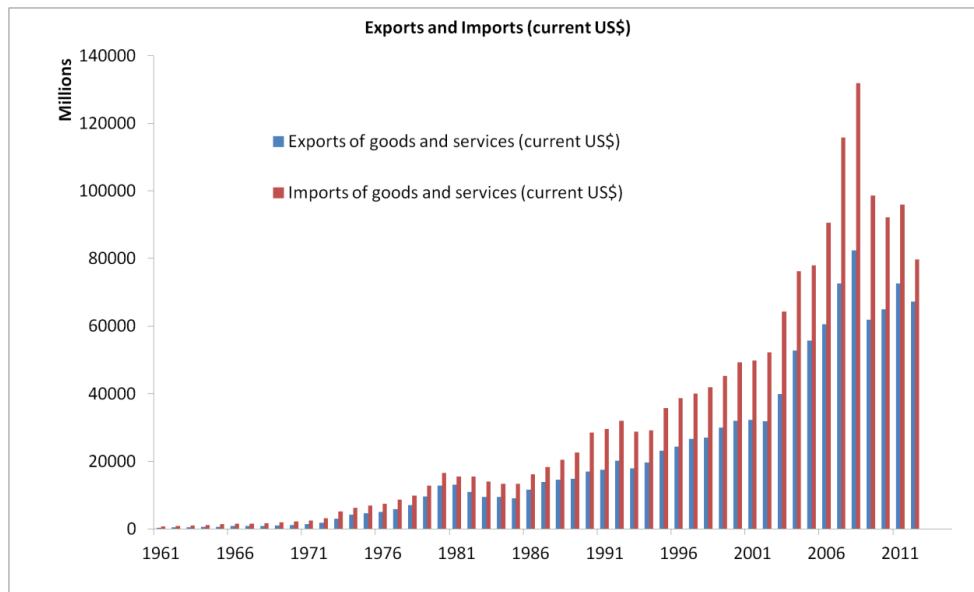


Οι υποκατηγορίες που μπορεί να έχουμε είναι:

- Για k -το πλήθος χώρες, για ένα δείκτη, πώς εξελίσσεται στο χρόνο (απλή περίπτωση: 1 χώρα)
- Για k_c -το πλήθος χώρες και για k_m -το πλήθος δείκτες, πώς εξελίσσεται στο χρόνο ο καθένας

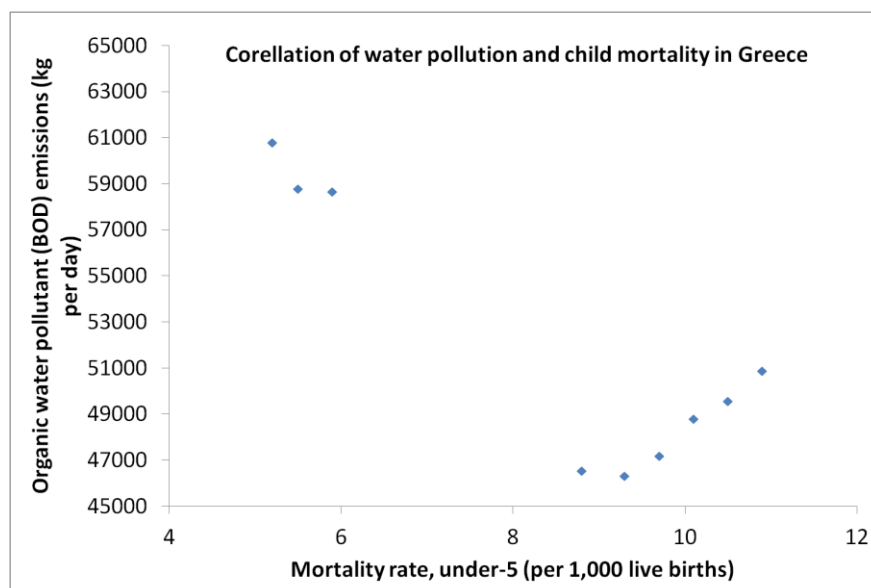


Bar charts. Η εν λόγω τεχνική χρησιμοποιείται για να συγκρίνει δύο ή περισσότερες μετρικές (y -axis) πάνω στις (ίδιες τιμές του άξονα των x). Γενικά, μπορούμε να γενικεύσουμε το παραπάνω σε περισσότερες από 2 μετρικές, k -το πλήθος στη γενική περίπτωση, αλλά με μικρό k (σκεφθείτε πόσο άσχημο θα ήταν το διάγραμμα για την περίπτωση 2 δεικτών και 2 χωρών).



Scatter Plots. Η εν λόγω τεχνική οπτικοποίησης συσχετίζει περισσότερες της μίας μετρικές και προσπαθεί να δείξει το βαθμό συσχέτισής τους. Για παράδειγμα, αν θέλουμε να δούμε πώς σχετίζεται η παιδική θνησιμότητα με την πρόσβαση σε «υψηλής ποιότητας» νερό για μια συγκεκριμένη χώρα, πρέπει να κάνουμε μια ερώτηση που να επιστρέφει για κάθε έτος το ποσοστό παιδικής θνησιμότητας και το ποσοστό πρόσβασης σε νερό υψηλής ποιότητας (κάθε εγγραφή του αποτελέσματος λέει έτος, παιδ. θνησ., μόλυνση ύδατος). Κάθε σημείο είναι ένα έτος για τη χώρα αυτή. Η συσχέτιση προκύπτει βάζοντας τις τιμές για τον ένα δείκτη στον ένα άξονα και τις τιμές για τον άλλο δείκτη στον άλλο άξονα.

Επίσης, μπορείτε να φιξάρετε ένα έτος και να ρωτήσετε για όλες τις χώρες. Σε αυτή την περίπτωση, ο ένας από τους 2 δείκτες μπορεί να είναι μια ιδιότητα των χωρών, π.χ., πληθυσμός. Κάθε σημείο είναι μια χώρα στην περίπτωση αυτή.



Δεδομένα που θα χρησιμοποιήσουμε φέτος

Θα χρησιμοποιήσουμε κάποια σύνολα δεδομένων που θα κατεβάσετε από το Kaggle. Θα βρείτε όλα τα δεδομένα που χρειαζόμαστε στο σχετικό φάκελο που εμφανίζεται στη σελίδα της εκφώνησης.

A. Στο <https://www.kaggle.com/datasets/sshashankrajak/countries> έχουμε στοιχεία για χώρες

ISO	ISO3	ISO_ Code	FIPS	Display_ Name	...	Region Name	...	Status	Developed /Developing	Population
AF	AFG	4	AF	Afghanistan		Asia		Independent	Developing	29121286
AX	ALA	248		Aland Islands		Europe		Part of FI	Developed	26711
AL	ALB	8	AL	Albania		Europe		Independent	Developed	2986952

B. Επίσης μας ενδιαφέρουν δεδομένα αγώνων ποδοσφαίρου από το

<https://www.kaggle.com/datasets/martj42/international-football-results-from-1872-to-2017>

ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΓΩΝΩΝ

date	home team	away team	home score	away score	tournament	city	country	neutral
06-02-2008	Georgia	Latvia	1	3	Friendly	Tbilisi	Georgia	FALSE
06-02-2008	Greece	Finland	2	1	Friendly	Nicosia	Cyprus	TRUE
06-02-2008	Honduras	Paraguay	2	0	Friendly	San Pedro Sula	Honduras	FALSE
06-02-2008	Hungary	Slovakia	1	1	Friendly	Limassol	Cyprus	TRUE
06-02-2008	Iran	Syria	0	0	FIFA World Cup qualification	Tehran	Iran	FALSE
06-02-2008	Iraq	China PR	1	1	FIFA World Cup qualification	Dubai	United Arab Emirates	TRUE

ΠΕΝΑΛΤΙ ΣΕ ΑΓΩΝΕΣ ΠΟΥ ΗΡΘΑΝ ΙΣΟΠΑΛΙΑ

date	home_team	away_team	winner	first_shooter
04-07-2024	Argentina	Ecuador	Argentina	Argentina
05-07-2024	Portugal	France	France	France
05-07-2024	Venezuela	Canada	Canada	Venezuela
06-07-2024	England	Switzerland	England	England
06-07-2024	Uruguay	Brazil	Uruguay	Uruguay
13-07-2024	Canada	Uruguay	Uruguay	Canada

Για όσους/όσες δεν ξέρουν από αυτά: αν ένας αγώνας πρέπει να καταλήξει σε πρόκριση μιας από τις δύο ομάδες και έρθει ισοπαλία, πάει στην παράταση. Αν κι εκεί δεν υπάρξει νικητής, βαράμε πέναλτι (shootouts), με μια διαδικασία που δεν ενδιαφέρει περεταίρω, μέχρι κάποιος να κερδίσει...

ΣΚΟΡΕΡ

date	home_team	away_team	team	scorer	minute	own_goal	penalty
28-10-1978	Greece	Hungary	Greece	Maik Galakos	58	FALSE	FALSE
28-10-1978	Greece	Hungary	Greece	Maik Galakos	67	FALSE	FALSE
28-10-1978	Greece	Hungary	Greece	Christos Ardzoglou	71	FALSE	FALSE
28-10-1978	Greece	Hungary	Greece	Thomas Mavros	89	FALSE	FALSE
28-10-1978	Greece	Hungary	Hungary	BÃ©la VÃ¡rady	90	FALSE	FALSE
15-11-1978	Austria	Portugal	Portugal	NenÃ©	30	FALSE	FALSE
15-11-1978	Austria	Portugal	Austria	Walter Schachner	71	FALSE	FALSE
15-11-1978	Austria	Portugal	Portugal	Alberto Fonseca	90	FALSE	FALSE

Στόχος

Ο τελικός σκοπός σας είναι να μπορέσετε να υλοποιήσετε μια εφαρμογή ενοποίησης και οπτικής εξαγωγής συμπερασμάτων η οποία θα αξιοποιεί δεδομένα που θα έχουν ενσωματωθεί σε μια βάση δεδομένων. Στο τέλος θέλουμε: (1) τα δεδομένα να έχουν φορτωθεί σε μια (α) καθαρή και (β) επερωτήσιμη βάση δεδομένων, (2) να έχει αναπτυχθεί πληροφοριακό σύστημα με front- και back-end, καθώς και ενδιαμέση λογική, και (3) να υπάρχει διαδραστική εφαρμογή που να μπορεί να επιτρέψει την πλοήγηση στα δεδομένα.

Θα χρειαστεί να υποστηρίξουμε ένα αναλυτή με έτοιμες αναφορές για διάφορα επί μέρους ερωτήματα που μπορεί να τον ενδιαφέρουν.

ΧΩΡΕΣ

Για κάθε χώρα, ο αναλυτής να μπορεί να δει το προφίλ της, τουτέστιν

- πόσες χρονιές αγωνίζεται η χώρα αυτή [από .. έως],
- συνολικά πόσες νίκες, ήττες, ισοπαλίες, αγώνες είχε η χώρα αυτή,
- το ίδιο, αλλά σπασμένο ως γηπεδούχος και φιλοξενούμενη
- ... άλλα στατιστικά που μπορεί να προσθέσετε... , και,
- οπωσδήποτε θέλουμε τα σχετικά linechart με νίκες, ήττες, ισοπαλίες, αγώνες ανά χρονιά

Επίσης, για κάθε χώρα, και με φίλτρα για το εύρος χρονιών, να μπορούμε να δούμε σαν πινακάκι, τις πληροφορίες για όλους τους αγώνες (όχι αναγκαστικά στην αναφορά του προφίλ, μπορεί με ένα κλικ να ανοίγει παραδίπλα ένα dependent report)

ΧΡΟΝΙΕΣ

Για μια χρονιά, ο αναλυτής να μπορεί να δει το προφίλ της, τουτέστιν να μπορούμε να δούμε

- πόσοι αγώνες, πόσοι ισόπαλοι, πόσοι με πέναλτι, ... διεξήχθησαν τη χρονιά αυτή
- οπωσδήποτε, για τα παραπάνω στατιστικά αγώνων ανά χώρα, θέλουμε τα σχετικά barchart (ενδεχομένως και νίκες/ήττες/κλπ.)
- κάπου χωριστά, τις πληροφορίες για όλους τους αγώνες, σαν πινακάκι, έχοντας και φίλτρα για τις χώρες (π.χ., Region Name, Status, Developed /Developing, ...) /* εξ ου και χρειαζόμαστε οι χώρες να φορτωθούν από το σχετικό αρχείο */ ,και,

ΚΑΘΟΛΙΚΑ ΣΤΑΤΙΣΤΙΚΑ

Αν υποθέσουμε μια scoring function με 3 πόντους ανά νίκη και 1 πόντο ανά ισοπαλία, για κάθε χώρα μπορούμε να βγάλουμε βαθμολογία. Ταξινομήστε τις χώρες

- με βάση αριθμό νικών
- με βάση το παραπάνω σκορ
- με βάση νίκες / χρονιές που αγωνίζεται η χώρα
- με βάση σκορ/ χρονιές που αγωνίζεται η χώρα
- ...

Υποχρεωτικά, βρείτε το τοπ-10 σε κάθε κατάταξη και δώστε τα σχετικά barcharts.

Υποχρεωτικά, φτιάξτε το σχετικό scatterplot {νίκες ή σκορ} X { Area_SqKm ή Population}.

ΠΑΙΧΤΕΣ

Για κάθε σκόρερ, προφίλ:

- Εύρος χρονιών που σκόραρε
- Σύνολο τερμάτων, μέγιστος αρ. τερμάτων ανά αγώνα
- Τέρματα / αγώνες της ομάδας του για το εύρος χρονιών που ο παίκτης αγωνίστηκε
- Το ίδιο ανά χρονιά, και με το σχετικό linechart

ΦΑΣΕΙΣ

Το project έχει τρεις φάσεις: (I) setup & προεπεξεργασία DBMS και δεδομένων, (II) υλοποίηση του κορμού μιας εφαρμογής και (III) τελειοποίηση της εφαρμογής και μελέτη.

ΦΑΣΗ I: ενοποίηση σε μία (1) ενιαία βάση δεδομένων

Η ενοποίηση έχει στόχο (α) να φτιαχτεί ένας **reference** πίνακας που να περιέχει τις χώρες με τα στοιχεία τους (και το σωστό πρωτεύον κλειδί) *ως **reference, single-version-of-the-truth, table for countries***, και (β) να μετασχηματισθούν οι υπόλοιπες πληροφορίες ώστε να έχουμε πίνακες με καθαρά δεδομένα και εγγυήσεις ορθότητας.

Κάθε ομάδα πρέπει να προβεί στις παρακάτω ενέργειες για το φόρτωμα:

1. Στήσιμο της βάσης και ενός γραφικού εργαλείου διαχείρισης (π.χ., MySQL & MySQL Workbench) στο μηχάνημά σας.
2. Download το κομμάτι των δεδομένων που θα χρησιμοποιήσετε – τα αντίστοιχα αρχεία δηλαδή.
3. Δημιουργήστε το σχήμα της βάσης για τα δεδομένα σας – όπως θα συζητήσουμε στο μάθημα (βλ. υποδείξεις στο παρακάτω παράδειγμα). Χρησιμοποιήστε InnoDB τύπο αποθήκευσης στη MySQL.
4. Δημιουργία scripts που μετατρέπουν τα εισερχόμενα αρχεία σε αρχεία φόρτωσης δεδομένων – αρχεία δηλαδή, στα οποία τα δεδομένα είναι έτοιμα προς φόρτωση
5. Δημιουργία scripts φόρτωσης των αρχείων φόρτωσης (π.χ., δείτε την εντολή LOAD DATA INFILE στη MySQL)
6. Φόρτωση των αρχείων και εξαγωγή backup της βάσης

Προσέξτε το codepage και το file encoding: UTF-8 is not enough.

A. Σχεδίαση. Μερικά ζητήματα σχεδίασης σε σχέση με τα δεδομένα.

- **Χώρες.** Τι θα κάνετε με το ότι κάποια δεδομένα είναι χώρας και κάποια είναι ομάδας χωρών (π.χ. Ευρώπη, G20, ...)?
- **Πέναλτι.** Τι θα κάνετε για εγγραφές που έχει το αρχείο πέναλτι, αλλά ο αγώνας δεν υπάρχει στο αρχείο με τα αποτελέσματα? Κανονικά πρέπει να τις βρείτε, καθώς και να απομονωθούν κάπου χώρια, σε ένα αρχείο καραντίνας.
- Αν τα πέναλτι μπουκ σε χωριστό πίνακα, ο σωστός τρόπος είναι να έχει foreign key στον πίνακα αποτελεσμάτων.
- **Κλειδιά.** Τι θα είναι primary key στον πίνακα countries? Τα πρωτεύοντα κλειδιά που είναι συμβολοσειρές είναι απολύτως ακατάλληλα για τη δουλειά. Επίσης πρέπει να δείτε μήπως υπάρχουν χώρες που βρίσκονται στη μία κατηγορία αρχείων και όχι στις άλλες. *Οι αγώνες περιέχουν και χώρες που το countries.csv δεν έχει.* Άρα θα πρέπει να έχετε μετασχηματισμούς που να καταφέρουν να φέρουν τα δεδομένα σε ένα (1) ενιαίο κλειδί αναφοράς για τις χώρες.
- Τι θα κάνετε με το primary key των αγώνων (και όχι μόνο). Σκεφτείτε: ποιο είναι το primary key αν απλά ακολουθήσετε την ανθρώπινη διαίσθηση? Όταν θα χρειαστεί να κάνετε join τον πίνακα αποτελεσμάτων με άλλους πίνακες, θα κάνετε join χρησιμοποιώντας όλο αυτό?
- *For the database that will ultimately hold your single-version-of-the-truth tables (i.e., not the ones that you will use for loading and transformation purposes, but for the ones hosting the clean v. of the data that will ultimately be queried by the users for info): Primary and Foreign keys are obligatory!!!*

Το να αφήσετε τη δομή των πινάκων ολόδια με τη δομή των πηγαίων αρχείων είναι λόγος αποτυχίας. Σε ό,τι αφορά τη σχεδίαση της βάσης:

(α) υποχρεωτικά reference table for the countries, με τα δεδομένα από το kaggle

(β) υποχρεωτικά **τα σωστά primary keys και foreign keys**

(γ) υποχρεωτικά μετασχηματισμός των υπολοίπων δεδομένων ώστε να φορτωθούν έχοντας αντί για τα απλά ονόματα των χωρών, τα σωστά κλειδιά

B. Extract – Transform – Load (ETL) the data. Τα πηγαία δεδομένα ΔΕΝ είναι στη δομή της τελικής τους μορφής μέσα στη βάση δεδομένων => πιθανώς θα πρέπει να μετασχηματισθούν.

- Μια πιθανή μέθοδος είναι χρησιμοποιήσετε κάποιους βοηθητικούς πίνακες υποδοχής, ώστε να φέρετε τα δεδομένα στη βάση, και μετά από κάποιες μετατροπές (με scripts / views / ... ό,τι χρειαστεί τέλος πάντων) να τροφοδοτήσετε τους πίνακες που θα χρησιμοποιηθούν για τις ερωτήσεις των χρηστών.
- Ο πιο συνηθισμένος τρόπος είναι να κατασκευασθούν κάποιες ροές εργασίας ETL που κάνουν αυτή τη δουλειά. Στο πλαίσιο της εργασίας αυτής, μπορείτε να το κάνετε εύκολα με κάποια script. Μπορείτε όμως και να χρησιμοποιήσετε κάποιο σχετικό εργαλείο.

Θα χρειαστεί να σχεδιάσετε και να τεκμηριώσετε καλά τη διαδικασία μετασχηματισμού και φόρτωσης των δεδομένων. Δείτε το [συνοδευτικό κείμενο](#) και τη [σελίδα project στο site του μαθήματος](#) για ETL που καλύπτει (α) τη διαδικασία σχεδίασης και (β) εργαλεία που υποστηρίζουν την εκτέλεση ETL ροών.

Υποχρεωτικά στο τέλος της διαδικασίας πρέπει να έχετε (i) DDL scripts for the creation of the database schema if needed, (ii) ETL scripts for the automated loading of the data in the database, (iii) backup of the loaded database. / if there is ever a failure, you (in fact: anyone) need to be able to recreate the database automatically */*

Γ. Query design. Με βάση το σχήμα της βάσης και το τι φορτώθηκε, πρέπει να μπορείτε να απαντήσετε τα βασικά ερωτήματα που έχει η εφαρμογή σας. *Άρα δοκιμάστε σε ένα περιβάλλον επερώτησης και κρατήστε σαν script αρχεία τις ερωτήσεις που θα χρειαστείτε στη συνέχεια.* Αυτό θα αποδειχθεί πιο δύσκολο απ' όση φαίνεται, κρατήστε χρόνο και γι' αυτό.

Θέλουμε ιδανικά όλα τα ερωτήματα, κι αν όχι όλα, όσο δυνατόν περισσότερα, να απαντώνται με SQL views ή έστω direct queries ΚΑΙ ΟΧΙ «φέρνω όλα τα δεδομένα σε μια συλλογή στην κύρια μνήμη και κάνω τη δουλειά εκεί». Άρα **το back-end data processing πρέπει να γίνεται ΜΕΣΑ στο DBMS**. Αυτό δεν μπορεί να γίνει καθολικά, αλλά η ιδέα είναι να μην πετάξουμε το μπαλάκι στον host κώδικα, όσο αυτό είναι εφικτό.

Οι views είναι ένα εξαιρετικό προγραμματιστικό εργαλείο για το σκοπό αυτό. Το ίδιο και οι ερωτήσεις με WITH statement καθώς και οι προσωρινοί πίνακες

<https://dev.mysql.com/doc/refman/8.4/en/with.html>

<https://dev.mysql.com/doc/refman/8.4/en/create-temporary-table.html>

Engineering is all about tradeoffs. You must judge the tradeoffs for whichever part of your design that you decide, and document the decision accordingly.

Ενδεικτικό παράδειγμα. Θα χρειαστεί σίγουρα να υπολογιστεί ποια χώρα νίκησε σε ένα αγώνα. Αυτό περιλαμβάνει υπολογισμούς: σε κάθε αγώνα το home_score – away_score μπορεί να το πει αυτό: αν είναι θετικό κέρδισε η γηπεδούχος, αρνητικό η φιλοξενούμενη, εκτός κι αν έρθει ισοπαλία, όπου αναλόγως το τουρνουά, μπορεί και έχουν υπάρξει πέναλτι, οπότε πρέπει να πας να το βρεις στα σχετικά δεδομένα για τα πέναλτι. Αυτό είναι κάτι που μπορεί να αποφασίσει κανείς πώς θα το υλοποιήσει, μιας και υπάρχουν εναλλακτικές με διαφορετικά οφέλη και κόστη:

- Να προσθέσει ως στήλη στον πίνακα, και να το υπολογίσει στο φόρτωμα
- Να φτιάξει μια (ή πολλές) view και να το υπολογίζει στο query runtime
- ...

<p><i>Η υποεντολή CASE μπορεί να σας φανεί χρήσιμη.</i></p>	<pre>SELECT r.*, home_score-away_score AS GoalDifference, CASE WHEN (home_score-away_score = 0) THEN NULL WHEN (home_score-away_score > 0) THEN home_team WHEN (home_score-away_score < 0) THEN away_team END AS winner FROM results r;</pre> <p>https://dev.mysql.com/doc/refman/8.4/en/case.html</p>
---	---

ΦΑΣΗ II: υλοποίηση κορμού εφαρμογής

Στη φάση II θα φτιάξετε κάποια γρήγορα prototypes από τις ερωτήσεις και τις οπτικοποιήσεις που απαιτούνται (όπως θα δείτε παρακάτω). Θα χρειαστεί:

1. Στήσιμο του προγραμματιστικού περιβάλλοντος στο οποίο θα γίνει η ανάπτυξη.
2. Στήσιμο του περιβάλλοντος στο οποίο θα στηθεί και θα τρέξει η εφαρμογή σας (ενδεχομένως το ίδιο).
3. Πειραματισμός με έτοιμα παραδείγματα από την τεκμηρίωση των τεχνολογιών που θα χρησιμοποιήσετε: φτιάξτε μικρά προγράμματα που να τρέχουν.
4. Κατασκευή του κώδικα που προσπελάζει τη βάση δεδομένων και (α) συνδέεται, (β) υποβάλει μια ερώτηση, (γ) διαχειρίζεται το αποτέλεσμα της.
5. Κατασκευή του κώδικα που οπτικοποιεί δεδομένα (όχι απαραίτητα αποτελέσματα ερωτήσεων σε βάση) με τον επιθυμητό τρόπο.
6. Προοδευτική σύνδεση των παραπάνω.

/ Στην φάση III θα εμπλουτίσετε την εφαρμογή σας με την πλήρη γκάμα από ερωτήσεις και οπτικοποιήσεις που ζητούνται */*

Τεχνολογίες που μπορείτε να χρησιμοποιήσετε (ενδεικτικά & όχι περιοριστικά):

- **d3** (<https://d3js.org/>) (αξίζει να αφιερώσετε ώρα να περιηγηθείτε στα παραδείγματα του <https://observablehq.com/@d3/gallery> τα οποία έχουν, όλα, και τον κώδικά τους μαζί).
- **JavaFX σε περιβάλλον Eclipse** (<https://openjfx.io/> && ένα εξαιρετικό tutorial στο σύνδεσμο: <https://code.makery.ch/library/javafx-tutorial/>). Προσοχή: μετά τη Java 8, δεν υπάρχει built-in support της JavaFX στο jdk και πρέπει να συμπεριλάβετε τα σχετικά jar στο project.
- **Swing**. Το Java Swing είναι ένα απλό και πανίσχυρο προγραμματιστικό εργαλείο (<https://docs.oracle.com/javase/tutorial/uiswing/components/index.html> και ιδίως <https://docs.oracle.com/javase/tutorial/uiswing/components/menu.html> <https://docs.oracle.com/javase/tutorial/uiswing/examples/components/>). Υπάρχει και η πολύ απλή library JFreeChart <https://www.baeldung.com/jfreechart-visualize-data>.
- **Οτιδήποτε άλλο περιβάλλον για front-end/application logic σας είναι οικείο.**

ΦΑΣΗ III: τελειοποίηση της εφαρμογής

Στη φάση III, θα πρέπει να προσθέσετε και ένα βαθμό διαδραστικότητας στο παραπάνω. Προσθέστε μενού επιλογής (ή άλλους τρόπους επιλογής) και χρησιμοποιήστε γραφικούς τρόπους αλληλεπίδρασης (π.χ., φόρμες και drop-down list boxes) ώστε να πάρετε από το χρήστη τι ακριβώς επιθυμεί να δει. Συνδέστε το κομμάτι αυτό με ερωτήσεις και οπτικοποιήσεις.

Παραδοτέα

Turnin: θα χρειαστεί να κάνετε turnin ένα text file που θα περιέχει: (α) AM, Ονόματα της ομάδας, (β) το link@Github, στο οποίο θα βρω αυτά που ζητάω στη σχετική λιστούλα, (γ) αν υπάρχει κάτι άλλο που θέλετε να μου πείτε. **turnin prj@mye030 AM1_AM2_AM3_prj.txt**

Στο **<project> @ GitHub** για το project σας (κατά προτίμηση με ένα διακριτό όνομα) θα αναμένω να δω (Α) το αρχικό README.md όπου θα πρέπει να φαίνονται οι συντελεστές, και οι όποιες τεχνικές λεπτομέρειες, (Β) το φάκελο src και το φάκελο data (τουλάχιστον) και (Γ) ένα φάκελο deliverables που θα περιέχει (α) το pdf για το τελικό παραδοτέο, (β) ένα link to a mp4 video, approx. 15' length, showing your system at work, along with an explanation of the internal structure and design. **Checklist:**

1. Ο κώδικας + δεδομένα + README.md μέσω ενός GitHub project
2. Τελική αναφορά με όνομα AM1_AM2_AM3_projectReport.pdf
3. Ένα link to video with your system at work, similarly named

Θα σας πρότεινα να μπειτε στη διαδικασία να κάνετε το project σας public, οπότε να μεριμνήσετε να έχετε test + documentation όπως πρέπει. Μπορείτε να δείτε παραδείγματα Readme.md στο [OOAD](#), στο [Pythia](#), και για πιο απλά, στο [Delian Cubes](#) και στο [Hecate](#).

Οδηγίες για το video. Προτεινόμενη δομή για το video που θα φτιάξετε.

ΜΕΡΟΣ Α (περίπου 5'): υπόβαθρο του project

- Ποιοι είμαστε στην ομάδα
- Τι αρχεία διαλέξαμε, τι δείκτες και χώρες
- Σχήμα της βάσης δεδομένων
- Πώς μετασχηματίσαμε και φορτώσαμε τα δεδομένα

ΜΕΡΟΣ Β (περίπου 5'): οργάνωση του project

- Αρχιτεκτονική του κώδικα (π.χ., δείχνοντας modules/packages/... μέσα σε ένα IDE)
- Εξήγηση της βασικής λειτουργικότητας του project μέσα στον κώδικα (δείχνοντας τα σχετικά κύρια σημεία μέσα σε ένα IDE)

ΜΕΡΟΣ Γ (περίπου 5'): Demo

- Demo

Προφανώς όλα τα παραπάνω είναι ενδεικτικά.

PROJECT CHECKLIST

REPEATABILITY		FRONT END	
<input type="checkbox"/> Scripts ETL			
<input type="checkbox"/> Scripts DDL		COUNTRIES	SCORERS
<input type="checkbox"/> Backup of the DB		<input type="checkbox"/> Profile	<input type="checkbox"/> Stats
		<input type="checkbox"/> Details	<input type="checkbox"/> Charts
ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ		<input type="checkbox"/> Charts	
<input type="checkbox"/> Εξηγήστε τι σχεδίαση που ακολουθήσατε και τα trade-offs			
<input type="checkbox"/> Έχετε PKs? FK's?		YEARS	GLOBAL STATS
<input type="checkbox"/> Έχετε σχέδιο για το πώς θα ρωτάτε?		<input type="checkbox"/> Profile	<input type="checkbox"/> Charts
		<input type="checkbox"/> Details	
ΠΑΡΑΔΟΤΕΑ		<input type="checkbox"/> Charts	
<input type="checkbox"/> Git repo with code			
<input type="checkbox"/> Report			
<input type="checkbox"/> Video			

Χρονοδιάγραμμα

Στη συνέχεια παρατίθενται ενδεικτικά στάδια της ανάπτυξης με ενδιάμεσες προθεσμίες (milestones) και (το μόνο υποχρεωτικό και παραδόσιμο) καταληκτική ημερομηνία ολοκλήρωσης.

<p>[11/02]</p> <p>Κάντε την Φάση I και ότι μπορείτε από II</p>	<p>Εκφώνηση</p> <hr/> <p>Εκτέλεση των βημάτων της ΦΑΣΗΣ I. Στόχοι:</p> <ul style="list-style-type: none"> • Exported Σχήμα + workbench screenshot • Φορτωμένη βάση για τα backbone data backup • scripts + 1-page diagram for the transformation process • First cut of view definitions and main queries
<p>[11/03]</p> <p>Η αρχή είναι το ήμισυ του παντός</p>	<p>Milestone: ΟΛΟΚΛΗΡΩΣΗ ΦΑΣΗΣ I</p> <hr/> <p>Στήσιμο προγραμματιστικού περιβάλλοντος/framework και κατανόησή τους 60% του χρόνου να τρέξει η πρώτη αναφορά, 20% του χρόνου να τρέξει η δεύτερη, the rest of the time for the rest. Στόχοι:</p> <ul style="list-style-type: none"> • Project setup of the application • Code containing the above forms/charts • Πρώτη οπτικοποίηση με dynamically constructed queries • Περιβάλλον αλληλεπίδρασης για τις επιλογές του χρήστη • Πρώτη έκδοση της αναφοράς
<p>[11/04]</p>	<p>Milestone: ΟΛΟΚΛΗΡΩΣΗ ΦΑΣΗΣ II</p> <hr/> <p>Πλήρης υλοποίηση της εφαρμογής</p> <p>Παραδοτέα:</p> <p>Π3.1: το σύστημα ως ολοκληρωμένος κώδικας (και ως back-end, και ως front-end, και τα scripts για το μετασχηματισμό και φόρτωση των δεδομένων), καθώς και τα δεδομένα (input, output, backups))</p> <p>Π3.2: τελική αναφορά (όπως στο σχετικό πρότυπο που βρίσκεται αναρτημένο στο δικτυακό τόπο του μαθήματος)</p> <p>Π3.3: video με τον σύστημα σε δράση, και την τελική αναφορά</p>
<p>[27/05]</p>	<p>Hard Deadline: ΟΛΟΚΛΗΡΩΣΗ ΦΑΣΗΣ III</p> <hr/>

ΚΑΛΗ ΕΠΙΤΥΧΙΑ!!