ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΥΕ030/ ΠΛΕ045 **ΠΡΟΧ. ΘΕΜΑΤΑ ΤΕΧΝΟΛΟΓΙΑΣ & ΕΦΑΡΜΟΓΩΝ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ**
ΑΝΟΙΞΗ 2018 **ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΑΣΚΗΣΗ**

Ημερομηνία Εξέτασης: Δευτέρα 21-05-2018 **Π. Βασιλειάδης**

Η προγραμματιστική άσκηση για το μάθημα είναι **υποχρεωτική** και αφορά τη σχεδίαση, υλοποίηση και ρύθμιση ενός ολοκληρωμένου πληροφοριακού συστήματος (κατασκευή βάσης δεδομένων, διαπροσωπεία, ρύθμιση λειτουργίας). 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.

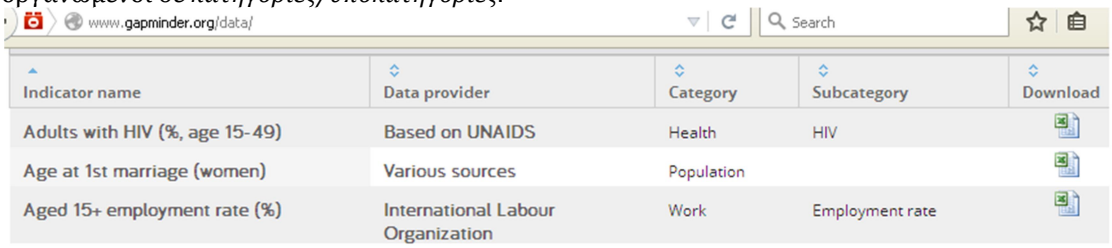
Είναι υποχρεωτικό να υλοποιήσετε τουλάχιστον ένα σύστημα με σχεσιακό back-end και γραφική διαπροσωπεία + την τελική αναφορά (βλ. στο τέλος της εκφώνησης).

Για φέτος, αποφάσισα ότι η έμφαση θα δοθεί στο πρόβλημα της ***οπτικοποίησης δεδομένων***. 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.***

Το project που θα κληθείτε να υλοποιήσετε στηρίζεται στα δεδομένα του οργανισμού GapMinder. Στην τοποθεσία http://www.gapminder.org/data θα βρείτε πλείστα όσα αρχεία για διάφορα είδη δεδομένων που χαρακτηρίζουν τον κόσμο μας τα τελευταία 70 χρόνια. Η GapMinder έχει ένα εξαιρετικό σύστημα για να βλέπει κανείς οπτικά την εξέλιξη των δεδομένων οπτικά, όμως, είναι δύσκολο να κάνουμε πιο ενδιαφέρουσες ερωτήσεις στο σύστημά της. Τα αρχεία από μόνα τους, βέβαια, δεν προσφέρονται ούτε για την απάντηση ερωτήσεων, ούτε για διαδραστικές οπτικοποιήσεις. Ως εκ τούτου, ***για να μπορούμε να απαντήσουμε ενδιαφέρουσες ερωτήσεις, πρέπει να οργανώσομε τα δεδομένα σε μια βάση δεδομένων και να χτίσουμε μια εφαρμογή γύρω τους!***

Δεδομένα

Τα δεδομένα της GapMinder αφορούν *δείκτες* (*indicators*), οι οποίοι με τη σειρά τους είναι οργανωμένοι σε *κατηγορίες/υποκατηγορίες*.



Επιλογή δεδομένων. ***Κάθε ομάδα θα πρέπει να διαλέξει δεδομένα από περίπου 10 δείκτες της ίδιας υποκατηγορίας.***
Επίσης θα πρέπει να συλλέξετε και 2 δείκτες με στοιχεία αναφοράς (π.χ., μπορείτε να διαλέξετε από Population, GDP, Gini index, Literacy/out-of-school) ***ώστε μετά να κάνετε αναλύσεις*** (π.χ., how are malaria deaths related to gini/gdp/region of the world/...)

Προσέξτε τι γίνεται όταν υπάρχουν και συγκεντρωτικοί και αναλυτικοί δείκτες: θα πρέπει να διαλέξετε όλους τους αναλυτικούς δείκτες που μπορούν να ανασυνθέσουν και το συνολικό, και όχι μόνο μερικούς από αυτούς.
Επίσης, αν θέλουμε να ανασυνθέσουμε πλήρως ένα δείκτη από τις επί μέρους κατηγορίες θα πρέπει να είμαστε πολύ προσεκτικοί. Ένα παράδειγμα σε σχέση με τον πληθυσμό. Π.χ., υπάρχει (α) Population aged 0-4 years, both sexes (%) και (β) Population aged 0-4 years, male (%) και Population aged 0-4 years, female (%). Στην περίπτωση αυτή, το (β) έχει ποσοστό επί γυναικών και ποσοστό επί ανδρών. Αν θέλουμε να έχουμε την κατανομή ανά κατηγορία ηλικίας (είτε σε απόλυτο αριθμό, είτε σε ποσοστό επί του συνολικού πληθυσμού), ανά φύλο, ανά έτος και ανά χώρα, πρέπει να βρούμε τα αρχεία με τον απόλυτο αριθμό ανθρώπων ανά ηλικιακή κατηγορία, να βρούμε το (male/female) sex ratio, να υπολογίσουμε τον απόλυτο αριθμό ή το ποσοστό του πληθυσμού ανά ηλικιακή κατηγορία και απο εκεί να βγάλουμε το επιθυμητό. Εκεί υπάρχει και μια δεύτερη δυσκολία: οι ηλικιακές κατηγορίες δεν είναι ακριβώς ίδιες. Οπότε πρέπει να απεικονίσετε κατάλληλα τις ομάδες από τα διαφορετικά αρχεία πληροφορίας, ώστε να βγει σωστά η ζητούμενη πληροφορία.

Στόχος

Ο τελικός σκοπός σας ως ομάδες είναι να μπορέσετε να υλοποιήσετε μια εφαρμογή οπτικής εξαγωγής συμπερασμάτων η οποία θα αξιοποιεί δεδομένα που θα έχουν ενσωματωθεί σε μια βάση δεδομένων.
Το project έχει τρεις φάσεις: (α) setup & προεπεξεργασία DBMS και δεδομένων, (β) σχεδίαση και φόρτωση δεδομένων και (γ) ανάπτυξη εφαρμογής.

ΦΑΣΗ Ι: αρχική οργάνωση

Κάθε ομάδα πρέπει να προβεί στις παρακάτω ενέργειες:

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

Σχεδίαση Βάσης Δεδομένων. Το κάθε αρχείο έχει δεδομένα σε ένα συγκεκριμένο format. Η απεικόνισή του σε ένα σχεσιακό σχήμα δεν είναι μονόδρομος. Για παράδειγμα: στο αρχείο GDP per employee, παίρνω ένα μικρό υποσύνολο από τις 3 πρώτες γραμμές:

GDP per employee, (constant 1990$)	1980	1981	1982	…
Albania	6690	6668	6674	…
Algeria	12352	12150	12444	…
Angola	1865	1769	1630	…
…				

Τα προβλήματα που έχουμε είναι:

- Έχουμε πολλές χώρες με τις οποίες θα ασχοληθούμε
- Έχουμε πολλά χρόνια, για τα οποία επιπλέον, σας ζητείται υποχρεωτικά να τα οργανώσετε σε 5ετίες, 10ετίες, 20ετίες
- Έχουμε πολλούς δείκτες που μας απασχολούν

Υπάρχουν πολλές σχεδιαστικές λύσεις για το πώς θα οργανώσετε την πληροφορία ώστε να είναι (α) ακριβής και συνεπής και (β) εύκολα επερωτήσιμη. Μερικές ενδεικτικές ιδέες ακολουθούν.

Α. Είναι εφικτό να έχετε την πληροφορία ανά δείκτη, χώρα και έτος (και ενδεχομένως άλλα χαρακτηριστικά, όπως π.χ., φύλλο) με τη χρήση lookup πινάκων. Π.χ., ξανά για το GDP per employee μπορώ να έχω την εξής οργάνωση:

Προσέξτε πώς τα έτη και οι χώρες αποθηκεύονται σε ένα lookup πίνακα. Προσέξτε ότι, αντί για τιμές μόνο (Albania, Algeria …), ο πίνακας `Countries` έχει μέσα (i) numeric primary key, (ii) το όνομα, φυσικά, καθώς και (iii) άλλες πληροφορίες (αυτοσχεδιάζω στα επιπλέον πεδία):

01	Albania	Europe	Tirana	…
02	Algeria	Africa	Algiers	…
…				

Προσέξτε επίσης πως το αρχείο εισόδου θα αποθηκευθεί πλέον σε ένα fact πίνακα με ένα foreign key σε κάθε lookup πίνακα που το αφορά. Η σχεδιαστική λύση αυτή έχει και αυτή πλεονεκτήματα (ποια?) και, ως συνήθως, δεν είναι δωρεάν (με πρώτο εμφανές κόστος ότι τα δεδομένα θέλουν ευρύτερους μετασχηματισμούς).

Β. Υπάρχει επίσης η σχεδιαστική δυνατότητα, πολλοί τέτοιοι fact πίνακες να συνδυαστούν για λόγους ευκολίας και χώρου σε ένα πίνακα με πολλά measures.

Γ. Υπάρχει η σχεδιαστική δυνατότητα, να διατηρήσετε τη δομή του αρχείου σε ένα πίνακα (still, think: θα έχετε πολλά αρχεία, ένα ανά δείκτη).

There is no silver bullet. Κάθε σχεδιαστική λύση αυτή έχει πλεονεκτήματα αλλά, ως συνήθως, δεν είναι δωρεάν – κάτι πληρώνουμε και κάτι κερδίζουμε. ***Στην σχεδίαση που θα κάνετε, σκεφτείτε τι θα πράξετε και τεκμηριώστε στην αναφορά (συνοπτικά), ποιες εναλλακτικές σχεδιάσεις σκεφθήκατε και γιατί προκρίνατε τελικά αυτή που τελικώς επιλέξατε. Ούτως ή άλλως, θα τα συζητήσουμε εγκαίρως και στο μάθημα.***

Extract – Transform – Load (ETL) the data. Τα πηγαία δεδομένα ΔΕΝ είναι στη δομή της τελικής τους μορφής μέσα στη βάση δεδομένων. Έτσι 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 συνήθης 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.

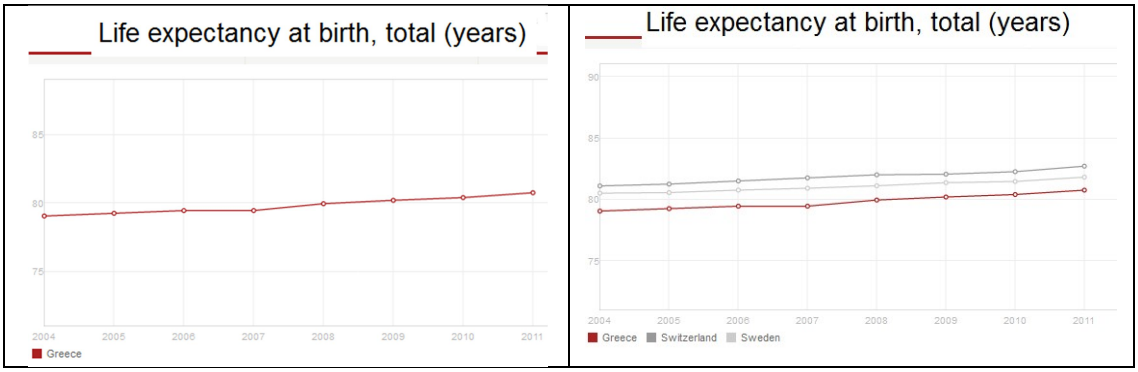
Θα χρειαστεί να σχεδιάσετε και να τεκμηριώσετε καλά τη διαδικασία μετασχηματισμού και φόρτωσης των δεδομένων. Δείτε το συνοδευτικό κείμενο για ETL που καλύπτει (α) τη διαδικασία σχεδίασης και (β) εργαλεία που υποστηρίζουν την εκτέλεση ETL ροών.

ΦΑΣΕΙΣ ΙΙ και ΙΙΙ: υλοποίηση εφαρμογής

Στη φάση ΙΙ θα φτιάξετε κάποια γρήγορα prototypes από τις ερωτήσεις και τις οπτικοποίησεις που απαιτούνται (όπως θα δείτε παρακάτω). Στην φάση ΙΙΙ θα αξιοποιήσετε πλήρως τα

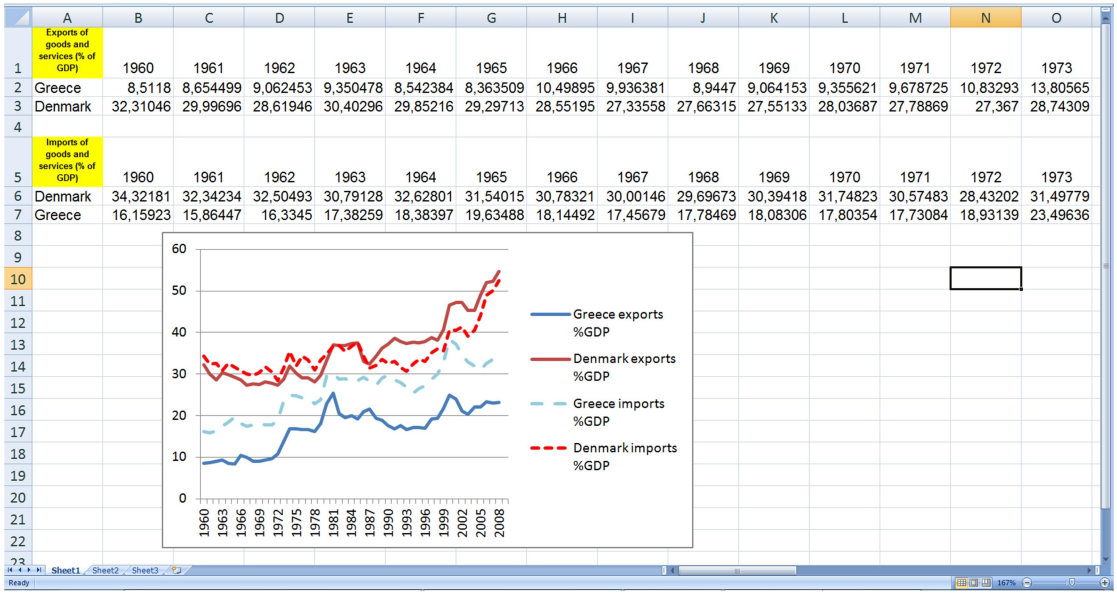
δεδομένα για την εξαγωγή συμπερασμάτων και θα εμπλουτίσετε την εφαρμογή σας με την πλήρη γκάμα από ερωτήσεις και οπτικοποιήσεις που ζητούνται.

Timelines / trendlines. Αν θέλουμε να δείξουμε την εξέλιξη ενός ή περισσοτέρων 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.



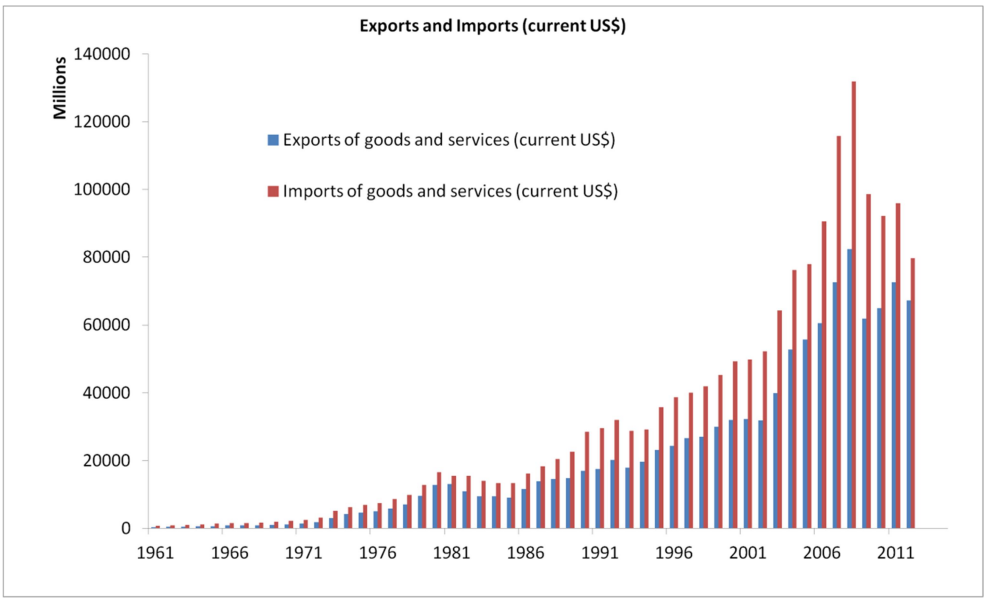
Οι υποκατηγορίες που μπορεί να έχουμε είναι:

- Για k-το πλήθος χώρες, για ένα δείκτη, δείξτε πώς εξελίσσεται στο χρόνο (απλή περίπτωση: 1 χώρα)
- Για k_C -το πλήθος χώρες και για k_m -το πλήθος δείκτες, δείξτε πώς εξελίσσεται στο χρόνο ο καθένας (το παράδειγμα εδώ είναι από άλλο data set, αλλά ενδεικτικό του τι εννοούμε)

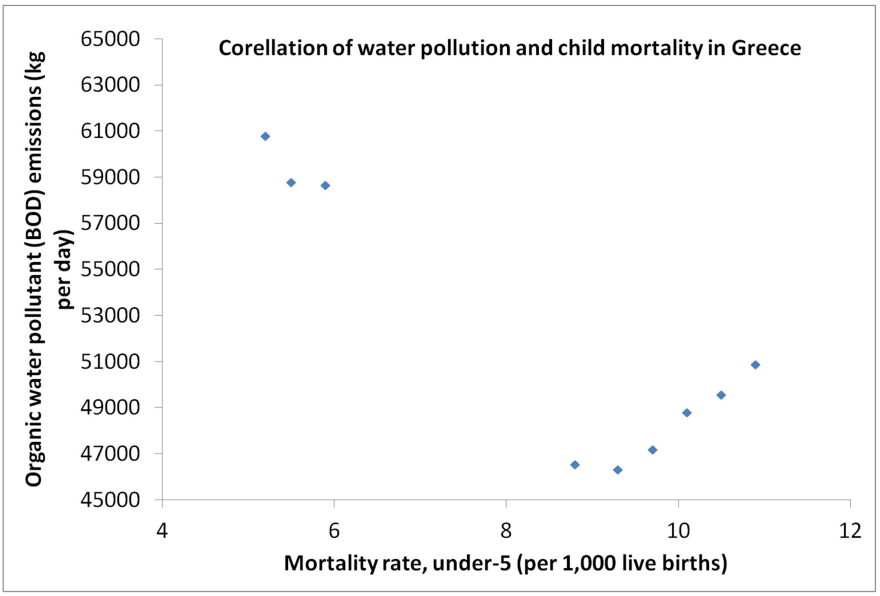


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Exports of goods and services (% of GDP)	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	1970	1971	1972	1973
2	Greece	8,5118	8,654499	9,062453	9,350478	8,542384	8,363509	10,49895	9,936381	8,9447	9,064153	9,355621	9,678725	10,83293	13,80565
3	Denmark	32,31046	29,99696	28,61946	30,40296	29,85216	29,29713	28,55195	27,33558	27,66315	27,55133	28,03687	27,78869	27,367	28,74309
4															
5	Imports of goods and services (% of GDP)	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	1970	1971	1972	1973
6	Denmark	34,32181	32,34234	32,50493	30,79128	32,62801	31,54015	30,78321	30,00146	29,69673	30,39418	31,74823	30,57483	28,43202	31,49779
7	Greece	16,15923	15,86447	16,3345	17,38259	18,38397	19,63488	18,14492	17,45679	17,78469	18,08306	17,80354	17,73084	18,93139	23,49636

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). Στην περίπτωσή μας, μπορούμε πάλι να έχουμε ένα συνδυασμό από χώρες και δείκτες.



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). Η συσχέτιση προκύπτει βάζοντας τις τιμές για τον ένα δείκτη στον ένα άξονα και τις τιμές για τον άλλο δείκτη στον άλλο άξονα.



Οδηγίες προς ναυτιλλομένους

Στο τέλος της εργασίας, θέλουμε ο χρήστης να μπορεί να επιλέξει (α) χώρες, (β) δείκτες και (γ) χρονικό εύρος και να απεικονίζεται το αποτέλεσμα είτε ανά χρόνο, είτε ανά πενταετία, κοκ. Κατασκευάστε αρχικά από ένα τέτοιο γράφημα ανά περίπτωση (ξεκινήστε από τα πιο απλά), με fixed query πίσω του, και δοκιμάστε να οπτικοποιήσετε το αποτέλεσμα. Μετά, ΠΡΟΟΔΕΥΤΙΚΑ, προσθέστε τη δυνατότητα επιλογών για τα (α)-(γ), ώστε η ερώτηση να κατασκευάζεται δυναμικά.

Από τις πολύ συχνά χρησιμοποιούμενες βιβλιοθήκες οπτικοποίησης είναι οι d3.js (javascript) for web development και οι JavaFX (built-in Java) ή jfreechart (Java library) για Java. **Για φέτος θα χρησιμοποιήσουμε JavaFX σε περιβάλλον eclipse** (βλ. ένα εξαιρετικό tutorial στο σύνδεσμο: http://code.makery.ch/java/javafx-8-tutorial-intro/).

Στη φάση ΙΙ, 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 προγραμματάκια που να τρέχουν
4. Κατασκευή του πρώτου script που προσπελάζει τη βάση δεδομένων και (α) συνδέεται, (β) υποβάλει μια ερώτηση, (γ) διαχειρίζεται το αποτέλεσμά της
5. Κατασκευή του πρώτου script που οπτικοποιεί δεδομένα (όχι απαραίτητα αποτελέσματα ερωτήσεων σε βάση) με τον επιθυμητό τρόπο.
6. Προοδευτική σύνδεση των παραπάνω

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

Μπορείτε να έχετε έτοιμες κάποιες ***προκατασκευασμένες βοηθητικές όψεις*** (views) ή να χρησιμοποιείτε ***προσωρινές όψεις ανάλογα με το ερώτημα*** (CREATE VIEW … -- SELECT … -- DROP VIEW …) ώστε να κάνετε την προγραμματιστική δουλειά πιο εύκολη.

Μπορείτε να ***αναλύσετε τις ερωτήσεις σας*** και αν διαπιστώσετε ότι μπορεί να ***παραμετροποιήσετε την κατασκευή τους*** (π.χ., ανάλογα με το τι δίνει ο χρήστης, να προσαρμόζονται τα πεδία του SELECT / GROUP-BY / …) και να φτιάξετε μεθόδους/συναρτήσεις που κατασκευάζουν την ερώτηση στη βάση με βάση τις παραμέτρους αυτές (με προφανές όφελος: write once, test a few times, safely use for ever)

Στο τελικό report αναμένεται να έχετε ρυθμίσει το DBMS && τη βάση (memory allocation, index creation, access rights, …) και να καταγράψετε τις ρυθμίσεις αυτές.

Χρονοδιάγραμμα

Στη συνέχεια παρατίθενται στάδια της ανάπτυξης, ενδιάμεσες προθεσμίες (milestones) και καταληκτικές ημερομηνίες ολοκλήρωσης (deadlines).

[**12/02**]	Εκφώνηση
Κάντε την Φάση Ι και ότι μπορείτε από ΙΙ	Εκτέλεση των βημάτων της ΦΑΣΗΣ Ι Παραδοτέα: Π1.1: Exported Σχήμα + workbench screenshot Π1.2: Φορτωμένη βάση για τα backbone data backup Π1.3: scripts + 1-page diagram for the transformation process
[**19/03**]	**Milestone**: **ΟΛΟΚΛΗΡΩΣΗ ΦΑΣΗΣ Ι**
Η αρχή είναι το ήμισυ του παντός	Μία αρχική οπτικοποίηση με fixed queries ανά κατηγορία 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* Παραδοτέα: Π1.4: Project setup of the application Π1.5. Code containing the above forms/charts
[**16/04**]	**Milestone**: **Ενδιάμεσα στη φάση ΙΙ**
Αν έχεις μία αναφορά, οι άλλες είναι εύκολες	Τουλάχιστον μία οπτικοποίηση με dynamically constructed queries Περιβάλλον αλληλεπίδρασης για τις επιλογές του χρήστη Παραδοτέα: Π2.1: Application code containing the above Π2.2. Πρώτη εκδοχή του τόμου με τα (α) – (δ) (βλ. παρακάτω)
[**30/04**]	**Hard Deadline**: **ΟΛΟΚΛΗΡΩΣΗ ΦΑΣΗΣ ΙΙ**
	Πλήρης υλοποίηση της εφαρμογής Παραδοτέα: **Π3.1**: το **σύστημα** εν λειτουργία **Π3.2**: **τελική αναφορά** εκτυπωμένη (όπως στο σχετικό πρότυπο που βρίσκεται αναρτημένο στο δικτυακό τόπο του μαθήματος) **Π3.3**: **DVD** με τον κώδικα, τα scripts, τα δεδομένα (input, output, backups) και την τελική αναφορά
[**21/05**]	**Hard Deadline**: **ΟΛΟΚΛΗΡΩΣΗ ΦΑΣΗΣ ΙΙΙ + ΕΠΙΔΕΙΞΗ @ 2018/05/21**

ΚΑΛΗ ΕΠΙΤΥΧΙΑ!!