

# ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ

Σχολή Τεχνολογικών Εφαρμογών  
Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων



Πτυχιακή Εργασία

## “Οπτική Αναπαράσταση Βάσεων Δεδομένων με JGraph”

Κλειδής Γεώργιος  
Κουτεντάκης Μύρων

Επιβλέπων Καθηγητής : Ακουμιανάκης Δημοσθένης

Ηράκλειο Ιούνιος 2008

**Πίνακας Περιεχομένων**

<b>1</b>	<b>Εισαγωγή.....</b>	<b>4</b>
<b>2</b>	<b>Γενικά.....</b>	<b>5</b>
2.1	<i>Μερικά λόγια για τις Βάσεις Δεδομένων.....</i>	5
2.1.1	Χαρακτηριστικά σχεσιακών συστημάτων βάσεων δεδομένων.....	7
2.1.2	Γλώσσες διαχείρισης δεδομένων.....	7
2.2	<i>Αντικειμενοστραφής προγραμματισμός.....</i>	8
2.2.1	Χαρακτηριστικά του αντικειμενοστραφή προγραμματισμού.....	10
2.2.2	Java.....	11
2.3	<i>Visualizations.....</i>	13
2.3.1	Εφαρμογές σε χάρτες σκέψων.....	14
2.3.2	Εφαρμογές στην παρουσίαση ειδήσεων.....	14
2.3.3	Εφαρμογές στην παρουσίαση δεδομένων.....	16
2.3.4	Εφαρμογές παρουσίασης συνδέσεων.....	20
2.3.5	Εφαρμογές παρουσίασης ιστοσελίδων.....	23
2.3.6	Εργαλεία για απεικόνιση δεδομένων.....	24
<b>3</b>	<b>Αναπαράσταση δεδομένων με το JGraph.....</b>	<b>26</b>
3.1	<i>Εισαγωγικά.....</i>	26
3.2	<i>Ορισμοί γράφων.....</i>	26
3.3	<i>Τι είναι το JGraph.....</i>	28
3.4	<i>Το πρόβλημα που αντιμετωπίζει η πτυχιακή.....</i>	31
3.4.1	Ποία είναι τα δεδομένα.....	31
3.4.2	Πως αναπαριστώνται?.....	32
3.5	<i>Παρουσίαση συστήματος.....</i>	33
3.5.1	Διαδραστικά αντικείμενα της Διεπαφής μας.....	34
<b>4</b>	<b>Υλοποίηση και σενάρια χρήσης.....</b>	<b>39</b>
4.1	<i>Σύνδεση με RDBMS μέσω JDBC.....</i>	40
4.2	<i>Διάβασμα των δεδομένων των ΒΔ του RDBMS.....</i>	40
4.3	<i>Μετατροπή των δεδομένων μιας ΒΔ σε αντικείμενα JAVA.....</i>	40
4.4	<i>Οπτική αναπαράσταση των αντικειμένων μέσω JGraph.....</i>	40
4.5	<i>Αλλαγή χωροδιάταξης των οπτικών αντικειμένων.....</i>	40
4.6	<i>Δυνατότητα άμεσης εύρεσης σχεσιακών αντικειμένων, για κάθε αντικείμενο.....</i>	43
4.7	<i>Μορφοποίηση οπτικών αντικειμένων.....</i>	43
4.8	<i>Εμφάνιση / Απόκρυψη πίνακα από την αναπαράσταση.....</i>	44
<b>5</b>	<b>Σύνοψη και συμπεράσματα.....</b>	<b>45</b>
	<b>Βιβλιογραφία.....</b>	<b>46</b>

## Πίνακας Εικόνων

ΕΙΚΟΝΑ 1: ΈΝΑ ΣΥΣΤΗΜΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ. ....	6
ΕΙΚΟΝΑ 2: ΔΥΟ ΑΠΟ ΤΑ ΠΟΙΟ ΓΝΩΣΤΑ ΑΝΟΙΧΤΟΥ ΚΩΔΙΚΑ (OPEN SOURCE) ΣΧΕΣΙΑΚΑ ΛΟΓΙΣΜΙΚΑ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ (RDBMS) .....	6
ΕΙΚΟΝΑ 3: ΟΡΓΑΝΩΣΗ ΜΙΑΣ ΤΥΠΙΚΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΜΕ ΤΗΝ ΒΟΗΘΕΙΑ ΤΩΝ (RDBMS).....	7
ΕΙΚΟΝΑ 4: ΈΝΑ ΑΠΛΟ ΕΡΩΤΗΜΑ ΣΕ SQL ΓΛΩΣΣΑ. ....	8
ΕΙΚΟΝΑ 5 : ΤΕΣΣΕΡΑ ΑΝΤΙΚΕΙΜΕΝΑ(INSTANCE) ΤΗΣ ΚΛΑΣΗΣ ΛΟΓΑΡΙΑΣΜΟΣ. ....	9
ΕΙΚΟΝΑ 6 : ΜΕΡΙΚΕΣ ΓΛΩΣΣΕΣ ΒΑΣΙΣΜΕΝΕΣ ΣΕ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΥΠΟΔΕΙΓΜΑ. ....	10
ΕΙΚΟΝΑ 7 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ WEB TREND MAP 2007 VERSION 2.0 . ....	14
ΕΙΚΟΝΑ 8 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ ΔΕΔΟΜΕΝΩΝ NEWSMAP. ....	15
ΕΙΚΟΝΑ 9 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ ΔΕΔΟΜΕΝΩΝ VOYA. ....	15
ΕΙΚΟΝΑ 10 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ ΔΕΔΟΜΕΝΩΝ DIGG BIGSPY.....	16
ΕΙΚΟΝΑ 11 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ ΔΕΔΟΜΕΝΩΝ DIGG STACK. ....	16
ΕΙΚΟΝΑ 12 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ ΔΕΔΟΜΕΝΩΝ AMAZTYPE. ....	16
ΕΙΚΟΝΑ 13 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ ΔΕΔΟΜΕΝΩΝ FLIKRTIME. ....	17
ΕΙΚΟΝΑ 14 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ ΔΕΔΟΜΕΝΩΝ TIME MAGAZINE. ....	17
ΕΙΚΟΝΑ 15 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ ΔΕΔΟΜΕΝΩΝ CRAZYEGG.....	18
ΕΙΚΟΝΑ 16 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ ΔΕΔΟΜΕΝΩΝ THREE VIEWS. ....	18
ΕΙΚΟΝΑ 17 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ ΔΕΔΟΜΕΝΩΝ WEBSITES AS GRAPHS. ....	19
ΕΙΚΟΝΑ 18 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ ΔΕΔΟΜΕΝΩΝ INTERACTIVE HISTORY TIMELINE. ....	19
ΕΙΚΟΝΑ 19 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ ΔΕΔΟΜΕΝΩΝ ELASTIC LISTS .....	20
ΕΙΚΟΝΑ 20 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ ΔΕΔΟΜΕΝΩΝ MUNTERBUND.....	20
ΕΙΚΟΝΑ 21 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ ΔΕΔΟΜΕΝΩΝ BURST LABS. ....	21
ΕΙΚΟΝΑ 22 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ ΔΕΔΟΜΕΝΩΝ FIGD’T VISUALIZER.....	21
ΕΙΚΟΝΑ 23 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ ΔΕΔΟΜΕΝΩΝ ΤΟΥ LEE BYRON. ....	22
ΕΙΚΟΝΑ 24 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ ΔΕΔΟΜΕΝΩΝ THE SHAPE OF SONG.....	22
ΕΙΚΟΝΑ 25 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ ΔΕΔΟΜΕΝΩΝ MUSICMAP. ....	22
ΕΙΚΟΝΑ 26 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ ΔΕΔΟΜΕΝΩΝ SPACETIME.....	23
ΕΙΚΟΝΑ 27 : ΕΦΑΡΜΟΓΗ ΑΠΕΙΚΟΝΙΣΗΣ ΔΕΔΟΜΕΝΩΝ UBROWSER.....	23
ΕΙΚΟΝΑ 28 : ΔΙΑΓΡΑΜΜΑ ΑΠΕΙΚΟΝΙΣΗΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΤΟΥ ΜΟΝΤΕΛΟΥ ΑΝΑΦΟΡΑΣ ΤΟΥ PREFUSE.....	25
ΕΙΚΟΝΑ 29 : ΈΝΑΣ ΑΠΛΟΣ ΓΡΑΦΟΣ(ΚΑΤΕΥΘΥΝΟΜΕΝΟΣ). ....	26
ΕΙΚΟΝΑ 30 : (1 <sup>ΟΞ</sup> ΓΡΑΦΟΣ) ΜΗ ΚΑΤΕΥΘΥΝΟΜΕΝΟΣ ,(2 <sup>ΟΞ</sup> ΓΡΑΦΟΣ) ΚΑΤΕΥΘΥΝΟΜΕΝΟΣ.....	27
ΕΙΚΟΝΑ 31 : ΟΠΤΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΕΝΟΣ ΓΡΑΦΟΥ(RADIAL TREE LAYOUT) ΜΕ ΤΗ ΧΡΗΣΗ ΤΟΥ JGRAPH.....	29
ΕΙΚΟΝΑ 32 : ΠΡΟΒΟΛΗ ΑΛΛΑΓΗΣ ΜΕΓΕΘΟΥΣ ΚΕΛΙΩΝ ΕΝΟΣ ΓΡΑΦΟΥ. ....	30
ΕΙΚΟΝΑ 33 : ΜΕΡΙΚΑ ΑΠΟ ΤΑ ΑΝΤΙΚΕΙΜΕΝΑ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ ΤΗΣ JAVA.....	34
ΕΙΚΟΝΑ 34 : ΔΙΕΠΑΦΗ ΕΙΣΟΔΟΥ ΤΗΣ ΠΤΥΧΙΑΚΗΣ ΜΑΣ.....	34
ΕΙΚΟΝΑ 35 : ΚΥΡΙΑ ΔΙΕΠΑΦΗ ΤΗΣ ΠΤΥΧΙΑΚΗΣ. ....	36
ΕΙΚΟΝΑ 36 : ΑΛΥΣΙΔΑ ΤΩΝ ΔΙΑΔΙΚΑΣΙΩΝ ΤΗΣ ΠΤΥΧΙΑΚΗΣ ΜΑΣ. ....	39
ΕΙΚΟΝΑ 37 : ΜΥ LAYOUT 1, ΑΡΧΙΚΗ ΑΠΕΙΚΟΝΙΣΗ .....	41
ΕΙΚΟΝΑ 38 : ΑΛΛΑΓΗ ΣΕΙΡΑΣ ΠΙΝΑΚΩΝ .....	42
ΕΙΚΟΝΑ 39: ΜΥ LAYOUT 2 .....	42
ΕΙΚΟΝΑ 40 : ΜΕ ΚΙΤΡΙΝΕΣ ΓΡΑΜΜΕΣ ΦΑΙΝΟΝΤΑΙ ΟΙ ΣΥΝΔΕΣΕΙΣ ΤΟΥ ΣΥΓΚΕΚΡΙΜΕΝΟΥ ΚΕΛΙΟΥ .....	43

# 1 Εισαγωγή

Κάθε εποχή χαρακτηρίζεται ή και προσδιορίζεται από κυρίαρχες αντιλήψεις για τους όρους της Επιστήμης, της Τεχνολογίας και της Κοινωνίας που άμεσα ή έμμεσα την καθορίζουν. Σήμερα, βρισκόμαστε στην εποχή της πληροφορίας (Information Age) γεγονός που κάνει επιτακτικά σημαντικό, ο σημερινός άνθρωπος να μπορεί να προσανατολίζεται σ' έναν τεράστιο ιστό διαθέσιμων πληροφοριών ( δεδομένων ), οι περισσότερες απ' τις οποίες είναι ποσοτικές.

Ο τρόπος αποθήκευσης της πληθώρας των δεδομένων αποτελούσε για χρόνια ένα μεγάλο πρόβλημα. Στις μέρες μας η ραγδαία εξέλιξη και διάδοση των υπολογιστών, έχουν καταστήσει σαφές ότι η λύση βρίσκεται στη ψηφιοποίηση των δεδομένων. Για το σκοπό αυτό έχουν δημιουργηθεί βάσεις δεδομένων ( 2.1 ) για κάθε τύπο πληροφορίας. Έτσι με τη βοήθεια των λογισμικών διαχείρισης δεδομένων μπορούμε να αποθηκεύουμε , να επεξεργαζόμαστε και να κάνουμε αναζήτηση σε ψηφιακά δεδομένα.

Βρίσκοντας λύση στο τρόπο αποθήκευσης, ανέκυψε ένα άλλο θέμα. Το πόσο εύκολα μπορεί να επεξεργάζεται τα δεδομένα αυτά ο χρήστης, γενικά το θέμα αυτό είναι ένας ολόκληρος κλάδος στην επιστήμη των υπολογιστών. Ο κλάδος αυτός ασχολείται με το να κάνει την αλληλεπίδραση ανθρώπου-υπολογιστή ευκολότερη. Η απεικόνιση της πληροφορίας είναι ένα κομμάτι του κλάδου αυτού. Το θέμα της πτυχιακής έχει να κάνει με την απεικόνιση των δεδομένων, παρακάτω θα δούμε αναλυτικότερα την έννοια της( 2.3 ), καθώς και άλλες έννοιες της επιστήμης των υπολογιστών που έχουν να κάνουν με το θέμα μας.

## 2 Γενικά

Η παρούσα πτυχιακή εργασία βασίζεται σε δύο ευρέως διαδεδομένα πεδία της πληροφορικής – αυτό των βάσεων δεδομένων και της διαχείρισης δεδομένων με αντικειμενοστραφή μεθόδους και εργαλεία. Το ιδιαίτερο πεδίο αναφοράς της εργασίας μας απορρέει από τη σύζευξη των δύο αυτών γνωστικών αντικειμένων για την ανάπτυξη διαδραστικών οπτικών αναπαραστάσεων δεδομένων (interactive visualizations). Στο κεφάλαιο αυτό επιχειρείται μια συνοπτική παρουσίαση των πεδίων αυτών προκειμένου να εξειδικευτεί τόσο το τα όρια μελέτης της πτυχιακής εργασίας όσο και η συνεισφορά αυτών που υλοποιήθηκαν.

### 2.1 Μερικά λόγια για τις Βάσεις Δεδομένων

Είναι γνωστό ότι ο άνθρωπος στην καθημερινότητα του έχει την ανάγκη επικοινωνίας με το περιβάλλον. Η επικοινωνία αυτή γίνεται με πολλούς τρόπους όπως, ομιλία, γραπτός λόγος, κινήσεις του σώματος, χειρονομίες ή και συνδυασμός όλων των παραπάνω. Το κύριο χαρακτηριστικό αυτής της επικοινωνίας, είναι η ανταλλαγή πληροφοριών που λαμβάνει χώρα ανάμεσα στον κάθε άνθρωπο και το χώρο μέσα στον οποίο ζει και κινείται. Οι πληροφορίες αυτές μπορεί να είναι οποιασδήποτε μορφής, και να αφορούν σκέψεις, περιγραφές, ή ακόμη και συναισθήματα. Ωστόσο, πάντα έχουν ένα συγκεκριμένο περιεχόμενο. Το περιεχόμενο αυτό έχει να κάνει με το φυσικό σύστημα από το οποίο παίρνουμε την πληροφορία.

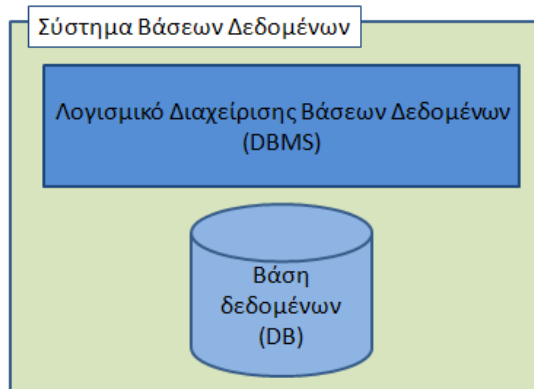
Στην επιστήμη των υπολογιστών οι πληροφορίες συνιστούν το ωφέλιμο αποτέλεσμα της επεξεργασίας ακατέργαστων δεδομένων (*data*). Τα δεδομένα μπορούν να οριστούν ως τρόποι αναπαράστασης εννοιών και γεγονότων που δύναται να υποστούν διαχείριση και επεξεργασία από τον άνθρωπο, ή από ένα αυτοματοποιημένο υπολογιστικό σύστημα. Η διαχείριση των δεδομένων αυτών γίνεται με ποικίλους τρόπους και εργαλεία.

Μια κατηγορία εξειδικευμένων εργαλείων διαχείρισης και επεξεργασίας δεδομένων είναι τα Συστήματα Διαχείρισης Βάσεων Δεδομένων (Database Management Systems DBMS). Τα συστήματα αυτά αναλαμβάνουν να διεκπεραιώσουν με υπολογιστικούς τρόπους σειρά καθηκόντων που επιτρέπουν τον ορισμό δεδομένων, την καταχώρησή τους σε φυσικά αποθηκευτικά μέσα, (π.χ. δίσκοι) και τη μετέπειτα ανάκληση και επεξεργασία τους από τους χρήστες.

Ένα σύστημα διαχείρισης βάσεων δεδομένων για να μπορέσει να διαχειριστεί τα δεδομένα σωστά και οργανωμένα τα χωρίζει σε τύπους. Ενδεικτικά παραδείγματα τύπων δεδομένων είναι το σύνολο των ακεραίων αριθμών, το σύνολο χαρακτήρων καθώς και ειδικοί τύποι όπως ημερομηνία. Επομένως, μια βάση δεδομένων που ορίζεται με τη χρήση ενός συστήματος διαχείρισης δεδομένων μπορεί να θεωρηθεί ως *ένα σύνολο δεδομένων που χαρακτηρίζονται από κάποια λογική οργάνωση και ομαδοποίηση έτσι ώστε να είναι εύκολη και αποτελεσματική η διαχείρισή τους*. Ευθύνη των συστημάτων διαχείρισης βάσεων δεδομένων είναι μεταξύ άλλων και να υποστηρίξουν κατάλληλες μεθόδους αποθήκευσης και προσπέλασης των δεδομένων.

Για το λόγο αυτό αξιοποιούνται συγκεκριμένα λογισμικά που συνιστούν τον πυρήνα ενός Συστήματος Διαχείρισης Βάσεων Δεδομένων.

Μια βάση δεδομένων και ένα λογισμικό διαχείρισης βάσεων δεδομένων, ονομάζεται σύστημα βάσεων δεδομένων (βλέπε Εικόνα 1). Σήμερα, υπάρχουν διαθέσιμα αρκετά λογισμικά διαχείρισης βάσεων δεδομένων στο εμπόριο από διάφορες εταιρείες από τα οποία τα περισσότερα υπακούουν και υποστηρίζουν το σχεσιακό μοντέλο δεδομένων. Ορισμένα από τα συστήματα αυτά διανέμονται υπό καθεστώς ελεύθερου και ανοικτού λογισμικού.



Εικόνα 1: Ένα σύστημα βάσεων δεδομένων.

Τα συστήματα βάσεων δεδομένων που διανέμονται υπό καθεστώς ελεύθερου και ανοικτού λογισμικού (βλέπε Εικόνα 2) μπορεί ο καθένας ελεύθερα να τα αξιοποιεί, να τα διανέμει, να τα αντιγράφει και να τα τροποποιεί ανάλογα με τις ανάγκες του, χωρίς να απαιτείται η απόκτηση άδειας. Η λογική τους εντάσσεται σε ένα εναλλακτικό μοντέλο ανάπτυξης και χρήσης λογισμικού, στο οποίο η δυνατότητα αλλαγών ή βελτιώσεων (ώστε να καλύπτονται οι ανάγκες αυτού που το χρησιμοποιεί) παρέχεται στο χρήστη μέσω της ελεύθερης διάθεσης και του πηγαίου κώδικα του λογισμικού. Γύρω από αυτή τη λογική δημιουργήθηκε μια τεράστια κοινότητα χρηστών και προγραμματιστών, με βάση το διαδίκτυο, οι οποίοι συμβάλλουν από κοινού στη συνεχή βελτίωση του λογισμικού, παρέχοντας δωρεάν τις γνώσεις και τη δουλειά τους σε όλους.

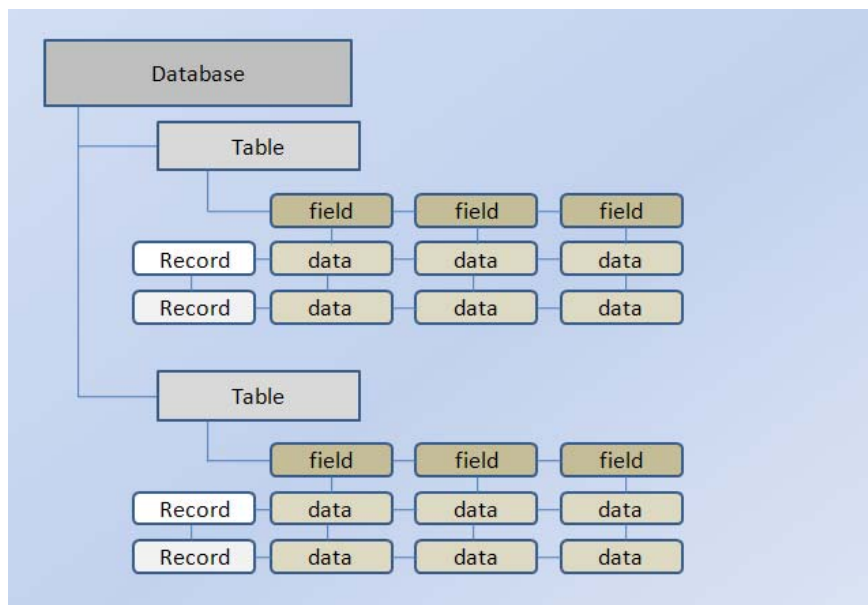


Εικόνα 2: Δυο από τα πιο γνωστά ανοικτού κώδικα (open source) σχεσιακά λογισμικά διαχείρισης βάσεων δεδομένων (RDBMS)

### 2.1.1 Χαρακτηριστικά σχεσιακών συστημάτων βάσεων δεδομένων

Τα σχεσιακά λογισμικά διαχείρισης βάσεων δεδομένων (RDBMS) είναι γρηγορότερα και πιο ευέλικτα στην διαχείριση των δεδομένων. Ο όρος σχεσιακή βάση δεδομένων προτάθηκε πρώτη φορά το 1970 από τον Edgar F.Codd σηματοδοτώντας μια προσέγγιση στη διαχείριση δεδομένων της οποίας οι βασικές αρχές χρησιμοποιούνται σε μεγάλο βαθμό και στις μέρες μας. Η δουλειά τους είναι να αποθηκεύουν τα δεδομένα σε χωριστούς πίνακες αναλόγως με την ιδιότητα των δεδομένων σε αντίθεση με τα κοινά λογισμικά (DBMS) όπου δεν γίνεται διαχωρισμός και αποθηκεύονται όλα σε ένα πίνακα. Η δομή των δεδομένων σε ξεχωριστούς πίνακες μπορεί να επεκταθεί και μέσα στους πίνακες με την βοήθεια των πεδίων (στήλες, υποκατηγορίες).

Οι έννοιες πίνακας και πεδίο (στήλη) είναι όροι στις σχεσιακές βάσεις δεδομένων. Αν φανταστούμε ένα πίνακα με τα πεδία του τότε με τον όρο εγγραφή θα εννοούμε μια (οριζόντια) γραμμή του. Οι εγγραφές είναι τα δεδομένα μιας βάσεις δεδομένων και τα πεδία είναι κάποια από τα χαρακτηριστικά των δεδομένων αυτών, τα οποία βρίσκονται όλα μαζί σε ένα πίνακα που έχει μόνο ίδιου τύπου δεδομένα, αυτή είναι και η λογική δομής ενός σχεσιακού λογισμικού διαχείρισης βάσεων δεδομένων (βλέπε Εικόνα 3).



Εικόνα 3: Οργάνωση μιας τυπικής βάσης δεδομένων με την βοήθεια των (RDBMS).

### 2.1.2 Γλώσσες διαχείρισης δεδομένων

Τα λογισμικά αυτά διαχειρίζονται μια τέτοια δομή μιας βάσης δεδομένων χρησιμοποιώντας ειδικές γλώσσες προγραμματισμού όπως η SQL (Structured Query Language). Η SQL είναι η πιο διαδεδομένη τυποποιημένη γλώσσα βάσεων δεδομένων που χρησιμοποιείται. Από το 1986 [SQL-86] η γλώσσα έχει τυποποιηθεί και μέχρι σήμερα έχει υποστεί σειρά τροποποιήσεων [SQL:2006]).

Η θεωρητική βάση της SQL γλώσσας βασίζεται στο μοντέλα που έχουν ρίζες σε μαθηματικά μοντέλα όπως η μαθηματική λογική και η άλγεβρα. Από αυτά τα

μοντέλα παράγονται και οι θεωρητικές σχεσιακές γλώσσες που βασίζονται στο σχεσιακό λογισμό (πεδίου τιμών ή πλειάδων) και τη σχεσιακή άλγεβρα. Ωστόσο, η χρήση μιας γλώσσας όπως η SQL καθιστά πιο εύκολη και ευέλικτη τη διαχείριση σχεσιακών δεδομένων. Οι δηλώσεις της SQL ακολουθούν συγκεκριμένο συντακτικό και καλύπτουν τόσο τον ορισμό όσο και τη διαχείριση δεδομένων. Οι δηλώσεις ορισμού δεδομένων επιτρέπουν τη δημιουργία ενός πίνακα με των πεδίων του, καθώς και τη προσθήκη, αφαίρεση και τροποποίηση δεδομένων (εγγραφών) σε αυτόν. Μια ακόμη εξίσου σημαντική δυνατότητα της SQL, είναι η ανάκτηση δεδομένων από μια βάση που επιτρέπουν την εξαγωγή χρήσιμης πληροφορίας. Η σύνταξη των SQL ερωτημάτων είναι πολύ ευέλικτη, έτσι η πληροφορίες που δίδουν μπορούν να προσαρμοστούν στις ανάγκες του χρήστη. Ειδικά για μεγάλες ΒΔ το να μπορούμε να παίρνουμε πληροφορίες από τα δεδομένα μας είναι που κάνει απαραίτητη την χρήση των σχεσιακών λογισμικών βάσεων δεδομένων και των SQL ερωτημάτων.

Για παράδειγμα έστω μια βιβλιοθήκη με χιλιάδες βιβλία και ένας χρήστης που επιθυμεί να εξάγει πληροφορία για τα βιβλία που κοστίζουν πάνω από 100 ευρώ. Το αντίστοιχο ερώτημα προς τη βάση δεδομένων μπορεί να αποδοθεί από την σύνταξη του SQL ερωτήματος που φαίνεται στην Εικόνα 4.

```
SELECT * FROM books
WHERE price > 100.00
ORDER BY title;
```

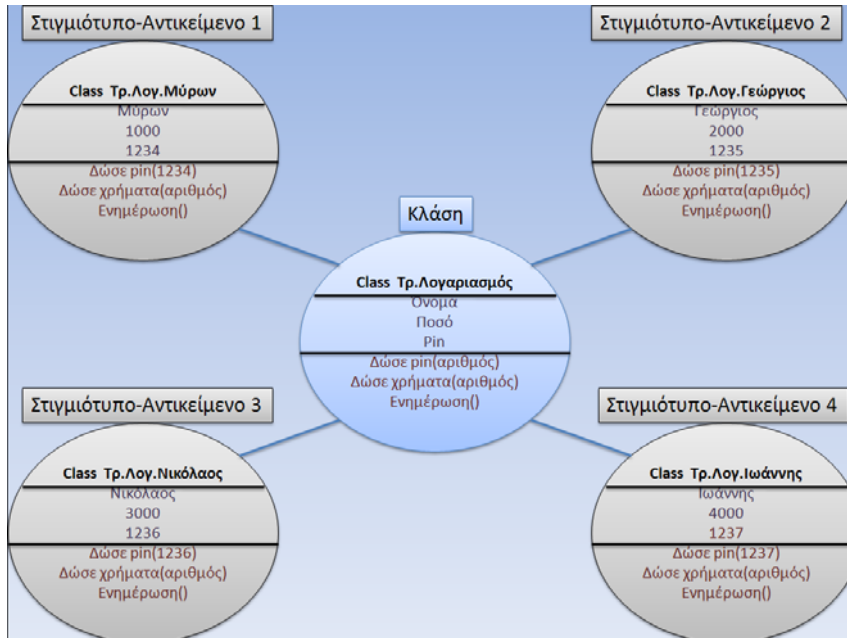
Εικόνα 4: Ένα απλό ερώτημα σε SQL γλώσσα.

## 2.2 Αντικειμενοστραφής προγραμματισμός

Ο αντικειμενοστραφής προγραμματισμός (oop, object-oriented programming) είναι ένα προγραμματιστικό υπόδειγμα που εμφανίστηκε το 1960 στην επιστήμη των υπολογιστών, καθιερώθηκε το 1990 και αντικατέστησε σε μεγάλο βαθμό το δομημένο προγραμματισμό. Στο προγραμματιστικό αυτό υπόδειγμα ο χειρισμός των σχετιζόμενων δεδομένων και των διαδικασιών που επενεργούν σε αυτά γίνεται από κοινού, μέσω μιας δομής δεδομένων που τα περιβάλλει ως αυτόνομη οντότητα με ταυτότητα και δικά της χαρακτηριστικά. Αυτή η δομή λέγεται *κλάση* και έχει δύο συνιστώσες, τα *πεδία* και τις *μεθόδους*. Τα πεδία περιγράφουν τι είναι η κλάση και οι μέθοδοι περιγράφουν τι κάνει η κλάση. Με άλλα λόγια τα πεδία είναι οι μεταβλητές που κάνει χρήση η κλάση αυτή. Οι μέθοδοι είναι οι λειτουργίες της κλάσης, αν φανταστούμε ότι μια κλάση περιγράφει έναν άνθρωπο τότε μια μέθοδος θα ήταν το περπάτημα, τρέξιμο, άλμα κ.α. , ενώ ένα πεδίο του θα ήταν το φύλο, χρώμα δέρματος, η καταγωγή του κ.α. . Αντικείμενα ονομάζονται όλα τα στιγμιότυπα μιας κλάσης, έτσι αντικείμενα της κλάσης άνθρωπος θα ήταν ένας λευκός(πεδίο) άντρας(πεδίο) από την Ελλάδα(πεδίο) με μεθόδους, τραγουδάω(μέθοδος) χορεύω συρτάκι(μέθοδος) για παράδειγμα(☺), ένα άλλο παράδειγμα φαίνεται στην (Εικόνα



5). Επομένως, κλάση είναι μια φόρμα ή ένα καλούπι που μέσο αυτού φτιάχνουμε τα αντίτυπα, εδώ αντικείμενα-στιγμιότυπα. Η χρήση των αντικειμένων του ΑΠ συμβάλει σε πιο σωστό προγραμματισμό με το να παρέχει κώδικα σύντομο και πιο αποδοτικό αποφεύγοντας την επαναχρησιμοποίηση κώδικα ιδιαίτερα σε μεγάλα και πολύπλοκα προγράμματα.



Εικόνα 5 : Τέσσερα αντικείμενα(instance) της κλάσης Λογαριασμός.

Η πρώτη γλώσσα προγραμματισμού που περιείχε αντικειμενοστραφής έννοιες ήταν η γλώσσα προγραμματισμού Simula 67, από την οποία επηρεάστηκε τη δεκαετία του '70 η ανάπτυξη της γλώσσας προγραμματισμού Smalltalk που εισήγαγε τον όρο αντικειμενοστραφής προγραμματισμός. Η Smalltalk αναπτύχθηκε από τον Άλαν Κέι της εταιρείας Xerox στο πλαίσιο μίας εργασίας με στόχο τη δημιουργία ενός χρήσιμου, αλλά και εύχρηστου, προσωπικού υπολογιστή. Το 1980 η τελική έκδοση της Smalltalk έγινε διαθέσιμη σηματοδοτώντας ότι η έρευνα για την αντικατάσταση του δομημένου προγραμματισμού με ένα πιο σύγχρονο υπόδειγμα απέδιδε αποτελέσματα. Στη γλώσσα αυτή καταργήθηκαν οι παραδοσιακές δομές δεδομένων από τις κλάσεις και έτσι όλα ήταν αντικείμενα.

Περίπου την ίδια εποχή και επίσης με επιρροές από την Simula, ολοκληρωνόταν η ανάπτυξη της C++ που ήταν η επέκταση της γλώσσας προγραμματισμού C με αντικειμενοστραφή χαρακτηριστικά. Με την επίδραση της C++ κατά την δεκαετία του '80 έβγαλαν πολλές γλώσσες της εποχής εκείνης αντικειμενοστραφής εκδόσεις. Το προγραμματιστικό υπόδειγμα του αντικειμενοστραφή προγραμματισμού επικράτησε το πρώτο ήμισυ της δεκαετίας του 1990 επειδή ήταν καταλληλότερο στην ανάπτυξη της τότε καθιέρωσης των γραφικών διασυνδέσεων χρήστη στους μικροϋπολογιστές. Το 1995 εμφανίζεται η Java ως μια πλήρως αντικειμενοστραφή γλώσσα που έμοιαζε στο συντακτικό της με την C++, προσέφερε πρωτοποριακές δυνατότητες για την εποχή και έδωσε νέα ώθηση στον αντικειμενοστραφή προγραμματισμό. Ταυτόχρονα εμφανίστηκαν ποικίλες άτυπες

βελτιώσεις στο βασικό προγραμματιστικό υπόδειγμα, όπως οι αντικειμενοστραφείς γλώσσες μοντελοποίησης λογισμικού και τα σχεδιαστικά πρότυπα. Το 2001 η Microsoft εστίασε την προσοχή της στην πλατφόρμα .NET, μία ανταγωνιστική της Java πλατφόρμα ανάπτυξης και εκτέλεσης λογισμικού η οποία ήταν εξολοκλήρου προσανατολισμένη στην αντικειμενοστρέφεια. Παρακάτω βλέπουμε μερικές από τις πιο γνωστές αντικειμενοστραφείς γλώσσες που αναφέραμε (Εικόνα 6).



Εικόνα 6 : Μερικές γλώσσες βασισμένες σε αντικειμενοστραφής υπόδειγμα.

### 2.2.1 Χαρακτηριστικά του αντικειμενοστραφή προγραμματισμού

Μια ιδιότητα που διαθέτουν οι κλάσεις είναι η *ενθυλάκωση δεδομένων* (data encapsulation), που τους επιτρέπει να κρύβουν τα δεδομένα τους από το υπόλοιπο πρόγραμμα και να εξασφαλίζουν ότι μόνο μέσω των δημοσίων μεθόδων τους αυτά μπορούν να προσπελαστούν. Μια κλάση μπορεί να επεκταθεί σε νέες κλάσεις, που δηλώνονται ως κληρονόμοι (υποκλάσεις ή 'παιδιά'), οι οποίες μπορούν να επαναχρησιμοποιήσουν τις μεταβιβάσιμες μεθόδους και ιδιότητες της γονικής τους κλάσης αλλά και να προσθέσουν δικές τους. Η ιδιότητα αυτή των κλάσεων λέγεται *κληρονομικότητα*. Τα στιγμιότυπα των κλάσεων 'παιδιών' αφού διαθέτουν τις ίδιες λειτουργίες είναι σε θέση και να χρησιμοποιηθούν αντί των γονιών (εφόσον τα παιδιά είναι κάποια πιο εξειδικευμένη έκδοση των γονιών) το αντίθετο δεν μπορεί να συμβεί. Παράδειγμα κληρονομικότητας είναι μία γονική κλάση Vehicle(=Όχημα) και οι δύο πιο εξειδικευμένες υποκλάσεις της Car (=Αυτοκίνητο) και Bicycle (=Ποδήλατο), οι οποίες λέμε ότι "κληρονομούν" από αυτήν. *Πολλαπλή κληρονομικότητα* είναι η δυνατότητα που έχει μια κλάση να κληρονομεί ταυτόχρονα από περισσότερες από μία γονικές. Από μία υποκλάση μπορούν να προκύψουν νέες υποκλάσεις που κληρονομούν από αυτήν, με αποτέλεσμα μία ιεραρχία κλάσεων που συνδέονται μεταξύ τους "ανά γενιά" με σχέσεις κληρονομικότητας.

Με την κληρονομικότητα, επιτυγχάνεται η επαναχρησιμοποίηση κώδικα ο οποίος έχει γραφεί πριν και είναι ένα από τα βασικά πλεονεκτήματα του αντικειμενοστραφή υποδείγματος προγραμματισμού.

- ❖ *Υπερφόρτωση μεθόδου* (method overloading) έχουμε όταν υπάρχουν στην ίδια ή σε διαφορετικές κλάσεις, μέθοδοι με το ίδιο όνομα αλλά διαφορετικά ορίσματα.

Διαφορετικά ορίσματα εννοούμε αριθμός(πλήθος) ορισμάτων και είδος του τύπου(ακέραιος, χαρακτήρας..) τους. Έτσι μπορεί μια μέθοδος να έχει το ίδιο όνομα άλλα ανάλογα με τι όρισμα θα της δώσουμε, όταν την χρησιμοποιούμε να αλλάζει η λειτουργία της.

- ❖ *Υποσκέλιση μεθόδου (method overriding)* είναι η κατάσταση κατά την οποία μία θυγατρική κλάση και η γονική της έχουν μία μέθοδο ομώνυμη και με τα ίδια ορίσματα. Χάρη στη δυνατότητα του πολυμορφισμού ο μεταγλωττιστής «ξέρει» πότε να καλέσει ποια μέθοδο, βασισμένος στον τύπο του τρέχοντος αντικειμένου. Δηλαδή πολυμορφισμός είναι η δυνατότητα των αντικειμενοστραφών μεταγλωττιστών να αποφασίζουν δυναμικά ποια είναι η κατάλληλη μέθοδος να κληθεί σε συνθήκες υποσκέλισης.
- ❖ *Αφηρημένη κλάση (abstract class)* είναι μία κλάση που ορίζεται μόνο για να κληρονομηθεί σε θυγατρικές υποκλάσεις και δεν υπάρχουν δικά της στιγμιότυπα (αντικείμενα). Μία αφηρημένη κλάση μπορεί να έχει και μη αφηρημένες μεθόδους οι οποίες υλοποιούνται στην ίδια την κλάση (αν και φυσικά μπορούν να υποσκελίζονται σε υποκλάσεις). Μία αφηρημένη κλάση που δεν έχει γνωρίσματα και όλες οι μέθοδοι της είναι αφηρημένες και δημόσιες καλείται *διασύνδεση (interface)*. Οι κλάσεις που κληρονομούν από μία διασύνδεση λέγεται ότι την "υλοποιούν".

## 2.2.2 Java

Η Java είναι μια νέα γλώσσα προγραμματισμού είναι ιδιαίτερα γνωστή όχι μόνο από τους υπολογιστές αλλά και από τα κινητά και πολλές άλλες συσκευές πολυμέσων καθώς και από το διαδίκτυο. Ανήκει στις αντικειμενοστραφείς γλώσσες προγραμματισμού και επιτρέπει σε οποιονδήποτε υπολογιστή, όπου και αν βρίσκεται αυτός, να έχει πρόσβαση και να χρησιμοποιεί μια εφαρμογή που είναι εγκατεστημένη σε κάποιο δίκτυο.

Η δημιουργία της Java ξεκίνησε το 1990 στην εταιρεία Sun από μια ομάδα ανθρώπων με επικεφαλή τον James Gosling. Ο σκοπός της ομάδας αυτής ήταν να δημιουργήσουν προγράμματα τα οποία θα έκαναν πιο "επαναστατική" τη χρήση και τη λειτουργία των οικιακών συσκευών(βίντεο, τηλεόραση, στερεοφωνικά κτλ). Στην αρχή χρησιμοποίησαν την C++ λόγω ταχύτητας αλλά ο Gosling, σύντομα διαπίστωσε ότι η C++ δεν ήταν η κατάλληλη γλώσσα προγραμματισμού για τα προγράμματα που ήθελαν να φτιάξουν. Οπότε με τη βοήθεια ενός άλλου προγραμματιστή, του Patrick Naughton τον Αύγουστο του 1991 δημιούργησε μια καινούργια γλώσσα την οποία ονόμασαν OAK (βελανιδιά, από μια βελανιδιά που υπήρχε στον κήπο του κτηρίου που ήταν το γραφείο τους). Κατάφεραν τελικά να φτιάξουν έναν υπολογιστή χειρός που έμοιαζε με τηλεχειριστήριο το οποίο όμως δεν παρουσίασε την αναμενόμενη εμπορική επιτυχία παρόλο που κατασκευαστικά πέτυχε.

Το 1993 η ομάδα προσπάθησε να προωθήσει την OAK στον τομέα Video On Demand (VOD) κάνοντας σχετική πρόταση στην εταιρεία Time Warner, η οποία

ενδιαφερόταν για τέτοια έργα την εποχή εκείνη .Επειδή ο τομέας αυτός όμως κρίθηκε εμπορικά μη αξιοποιήσιμος η πρόταση απορρίφθηκε. Την εποχή οπου το internet ξεφεύγει από την κατάσταση απλού κειμένου και αρχίζουν να εμφανίζονται τα γραφικά , ένας από τους ιδύνοντες της Sun εκείνη την περίοδο, διέβλεψε ότι η ΟΑΚ θα μπορούσε να χρησιμοποιηθεί στον παγκόσμιο ιστό. Έτσι η ομάδα το αντιλαμβάνεται και αλλάζει κατεύθυνση του έργου για τρίτη φορά, η φιλοσοφία της εταιρείας ήταν να διαθέτει τη γλώσσα δωρεάν ώστε να τυποποιηθεί. Κατόπιν μετονομάζεται σε Java τον Ιανουάριο του 1995, και λίγο αργότερα παρουσιάζεται επίσημα από την Sun ως εργαλείο ανάπτυξης εφαρμογών για το Internet. Η διάδοση της αρχίζει από τα προγράμματα πλοήγησης όπως Netscape Navigator (2.0), Internet Explorer(3.0) με υποστήριξη Java.

Όπως αναφέραμε και παραπάνω η Java από πλευράς δομής μοιάζει περισσότερο με την C/C++. Μοιάζει αρκετά στον τρόπο σύνταξης και χρησιμοποιεί πολλά στοιχεία της, δεν είναι όμως C/C++. Αποφεύγει πολλά χαρακτηριστικά της C/C++ για λόγους απλότητας και ασφάλειας, ενώ σε άλλα επεκτείνει τις λειτουργίες τους για να γίνει πιο ευέλικτη και πιο σύγχρονη γλώσσα. Έχει πάρει επίσης και στοιχεία από την γλώσσα προγραμματισμού Smalltalk όπως τον πηγαίο κώδικα (Byte code) και το Garbage Collection(αυτόματος καθαρισμός της μνήμης).

Μια από τις ιδιότητες που κάνουν την Java να ξεχωρίζει από άλλες γλώσσες προγραμματισμού, είναι τα applets, ειδικά προγράμματα που το σημαντικό τους χαρακτηριστικό είναι η ασφάλεια που διαθέτουν από ιούς και προγράμματα με κακό σκοπό. Αυτό συμβαίνει διότι τα applets δεν μπορούν να γράψουν στο σκληρό δίσκο αν δεν πάρουν άδεια. Για να τρέξει ένα applet χρειάζεται ένα πρόγραμμα πλοήγησης που να έχει σύνδεση στο διαδίκτυο ώστε να μπορούμε να το κατεβάσουμε. Η Java εκτός από τα applet μπορεί να χρησιμοποιηθεί και για την δημιουργία παντός φύσεως προγραμμάτων τα οποία καλούνται εφαρμογές (applications).

Η Java σήμερα είναι μια ιδιαίτερα δημοφιλής γλώσσα που αξιολογείται με ποικίλους τρόπους και για διάφορα πεδία εφαρμογών. Αυτό οφείλεται σε ορισμένα ιδιαίτερα χαρακτηριστικά που καθιστούν τη γλώσσα δημοφιλή, όπως τα ακόλουθα:

- ❖ *Απλότητα.* έννοιες όπως δείκτες, δομές και ενώσεις της C/C++ δεν συναντάμε στην Java, αυτό την κάνει μια σχετικά απλή γλώσσα.
- ❖ *Αντικειμενοστραφείς.* Η Java χρησιμοποιεί κλάσεις για να οργανώσει τον κώδικα σε λογικές ενότητες όπως και η C/C++. Όταν εκτελείται ένα πρόγραμμα δημιουργεί από τις κλάσεις αντικείμενα, τα οποία αποτελούνται από τα πεδία και τις μεθόδους. Τα πεδία λένε τι είναι το αντικείμενο και οι μέθοδοι τι κάνει το αντικείμενο. Οι κλάσεις μπορούν να κληρονομούν ιδιότητες από μια άλλη κλάση όχι όμως από πολλές(πολλαπλή κληρονομικότητα).
- ❖ *Μεταγλωττιζόμενη ή διερμηνευόμενη.* Μετά την μεταγλώττιση του πηγαίου κώδικα παράγεται ένας ειδικός κώδικας ο λεγόμενος κώδικας byte(bytecode). Ο κώδικας αυτός μοιάζει με τον κώδικα μηχανής μπορεί όμως να εκτελεστεί από οποιοδήποτε λειτουργικό σύστημα αρκεί να διαθέτει διερμηνευτή της Java.

- ❖ *Ασφάλεια.* Από τη αρχή η Java σχεδιάστηκε με σκοπό την ασφάλεια εκτέλεσης του κώδικα στο διαδίκτυο. Αυτό είχε ως αποτέλεσμα τον περιορισμό αρκετών χαρακτηριστικών που έχει η C/C++, έτσι δεν υπάρχουν δείκτες, ούτε μπορεί να γίνει αυθαίρετη προσπέλαση διευθύνσεων της μνήμης(προστασία από ιούς). Η Java χρησιμοποιεί επίσης ένα ισχυρό μηχανισμό για τον έλεγχο αναμενόμενων και μη αναμενόμενων σφαλμάτων(exceptions).
- ❖ Υποστηρίζει *πολυνημάτωση* (multithreading). Ένα πρόγραμμα μπορεί να αποτελείται από πολλές διαδικασίες οι οποίες εκτελούνται συνεχώς και ανεξάρτητα η μια από την άλλη.
- ❖ Συλλογή ακρήστων(garbage collection). Διαγράφονται αυτόματα ότι δεδομένα υπάρχουν στην μνήμη που δεν χρησιμοποιούνται
- ❖ Δημιουργεί γρήγορο κώδικα. Αυτό οφείλεται στην χρήση γρήγορων μεταγλωττιστών των λεγόμενων “just in time compilers”(JIT).
- ❖ Παρέχει βιβλιοθήκες κώδικα για διάφορες χρήσεις, όπως δημιουργία γραφικών, χειρισμό αλφαριθμητικών, μαθηματικές πράξεις, προσπέλαση αρχείων στο διαδίκτυο, χειρισμό σχεσιακών βάσεων δεδομένων, δημιουργία εφαρμογών πελάτη – εξυπηρετητή, κλήση απομακρυσμένων αντικειμένων κτλ.

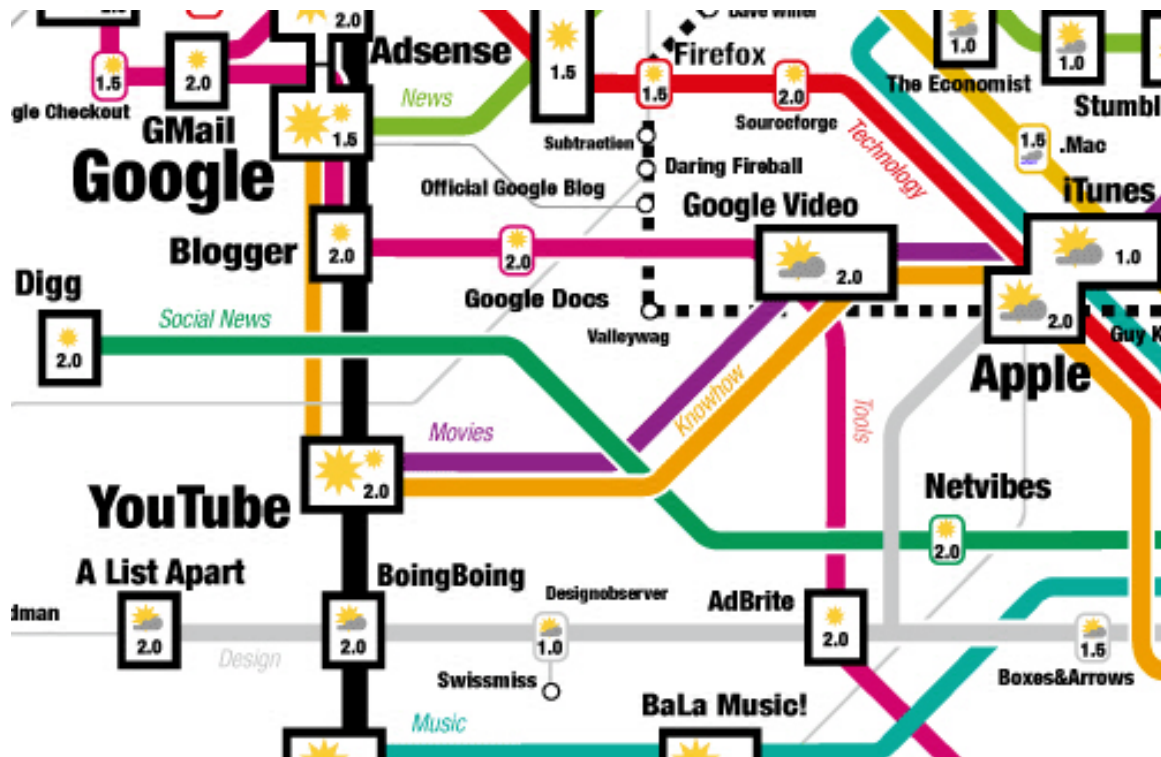
## 2.3 Visualizations

Η *παρουσίαση των δεδομένων* (data visualization) ανήκει στον κλάδο Αλληλεπίδρασης Ανθρώπου-Υπολογιστή της επιστήμης των υπολογιστών. Ο κλάδος αυτός ασχολείται με τις προκλήσεις που παρουσιάζονται στο να γίνουν οι υπολογιστές και οι υπολογισμοί χρήσιμοι, εύκολοι στη χρήση και ευρέως προσπελάσιμοι προς τους χρήστες. Στην αναφορά αυτή δεν θα γίνει ανάλυση ολόκληρου του κλάδου αυτού αλλά του κομματιού που έχει να κάνει με το θέμα της πτυχιακής. Το κομμάτι αυτό αφορά την ευκολότερη παρουσίαση δεδομένων μιας σχεσιακής βάσης, ώστε να γίνεται απλή η κατανόηση της πληροφορίας που περιέχεται.

Η υπολογιστική ισχύ των υπολογιστών έχει αυξηθεί πολύ στις μέρες μας, συνεπώς και οι δυνατότητες αποθήκευσης και επεξεργασίας μεγάλου όγκου δεδομένων. Οι δυνατότητες αυτές γίνονται πιο αποτελεσματικές όταν συνδυάζονται με κατάλληλη παρουσίαση της πληροφορίας προς τον χρήστη. Τα *γραφικά* είναι ένας παράγοντας που έχει βοηθήσει αρκετά στην παρουσίαση των δεδομένων. Καθημερινά παρουσιάζονται καινούργιοι και πιο σύγχρονοι τρόποι-τεχνικές απεικόνισης των δεδομένων. Οι επιλογές σε τεχνικές απεικόνισης σήμερα είναι πολλές και πραγματικά καταφέρνουν να παρουσιάσουν τα δεδομένα εντυπωσιακά, συναρπαστικά και ιδιαίτερα περιγραφικά. Ας δούμε μερικά παραδείγματα από την εφαρμογή των τεχνικών αυτών στο διαδίκτυο.

### 2.3.1 Εφαρμογές σε χάρτες σκέψεων

Η παρακάτω απεικόνιση δεδομένων (Εικόνα 7) παρουσιάζει τις 200 δημοφιλέστερες ιστοσελίδες στο διαδίκτυο, δομημένες ανά κατηγορία, σχέσης, επιτυχίας, δημοσιότητας και προοπτικής ως ένα χάρτη συγκοινωνιών(μετρό). Όσο πιο μεγάλος-κεντρικός σταθμός τόσο πιο δημοφιλής είναι η ιστοσελίδα.



Εικόνα 7 : Εφαρμογή απεικόνισης Web Trend Map 2007 Version 2.0 .

### 2.3.2 Εφαρμογές στην παρουσίαση ειδήσεων

Το Newsmap (Εικόνα 8) είναι μια εφαρμογή που αντιδράει αλλάζοντας οπτικά το χάρτη των ειδήσεων του Google. Τα διάφορα χρώματα απεικονίζουν τα διάφορα θέματα ειδήσεων(π.χ. πορτοκαλί ίσον τοπικά, κόκκινο ίσον εξωτερικά κτλ), όσο πιο ανοιχτό είναι το χρώμα τόσο πιο φρέσκια είναι η είδηση, ενώ το μέγεθος της γραμματοσειράς δείχνει πόσο δημοφιλές είναι το κάθε θέμα την συγκεκριμένη στιγμή.



Εικόνα 8 : Εφαρμογή απεικόνισης δεδομένων Newsmap.

Το Voyage (Εικόνα 9) είναι ένα RSS - feeder το οποίο παρουσιάζει τα πιο πρόσφατα νέα υπό μορφή διαφάνειας. Κάνοντας zoom ο χρήστης εισέρχεται σε άλλες διαφάνειες με παλαιότερα θέματα. Η πλοήγηση μπορεί να γίνει και με timeline, μια μπάρα που φέρει επάνω της όλες τις διαφάνειες με τα θέματα των ειδήσεων ανά μέρα.



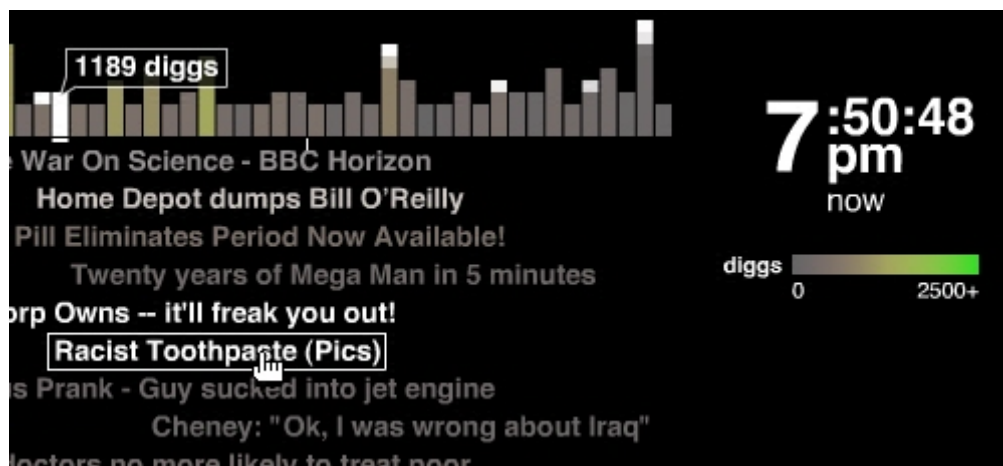
Εικόνα 9 : Εφαρμογή απεικόνισης δεδομένων Voyage.

Το Digg BigSpy (Εικόνα 10) τακτοποιεί διάφορες ιστορίες από το διαδίκτυο ανάλογα με το πόσο ο κόσμος της αναζητεί. Όσο πιο μεγάλες τόσο πιο πολλές αναζητήσεις, παράλληλα γράφετε και ο αριθμός των αναζητήσεων.



Εικόνα 10 : Εφαρμογή απεικόνισης δεδομένων Digg BigSpry.

Το Digg Stack(Εικόνα 11) είναι παρόμοιο με το Digg BigSpry. Εδώ παρουσιάζονται οι αναζητήσεις θεμάτων σε στοίβες. Όσο πιο πολλές αναζητήσεις τόσο μεγαλύτερες γίνονται οι στοίβες.



Εικόνα 11 : Εφαρμογή απεικόνισης δεδομένων Digg Stack.

### 2.3.3 Εφαρμογές στην παρουσίαση δεδομένων

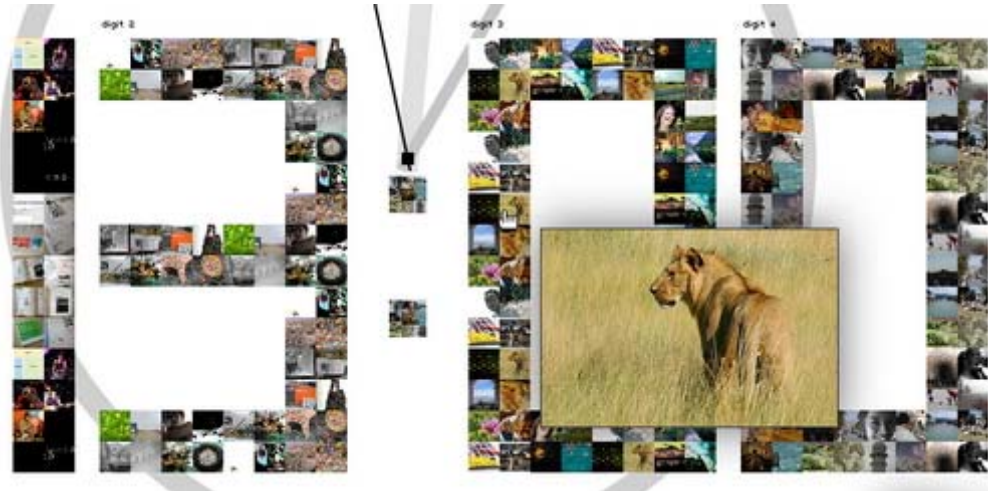
Το Amaztype (Εικόνα 12) εκτελεί μια τυπική αναζήτηση βιβλίων απεικονίζοντας τα αποτελέσματα όπως την λέξη αναζήτησης που του δίνουμε, αν θέλουμε να πάρουμε παραπάνω πληροφορία για ένα βιβλίο απλά το επιλέγουμε με το ποντίκι.



Εικόνα 12 : Εφαρμογή απεικόνισης δεδομένων Amaztype.

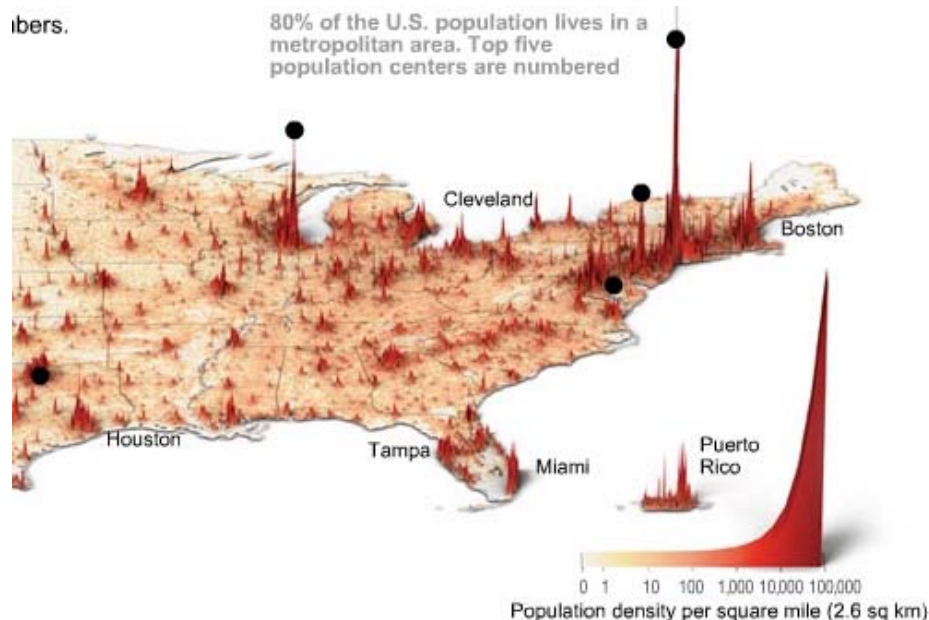


Παρόμοια ιδέα χρησιμοποιεί το flickrtime (Εικόνα 13) όπου τα δεδομένα είναι φωτογραφίες που ανεβάζουν οι επισκέπτες της σελίδας και η παρουσίαση είναι σε μορφή της ώρας εκείνης την στιγμής. Για επιπλέον πληροφορία για μια φωτογραφίας απλά την επιλέγουμε.



Εικόνα 13 : Εφαρμογή απεικόνισης δεδομένων flickrtime.

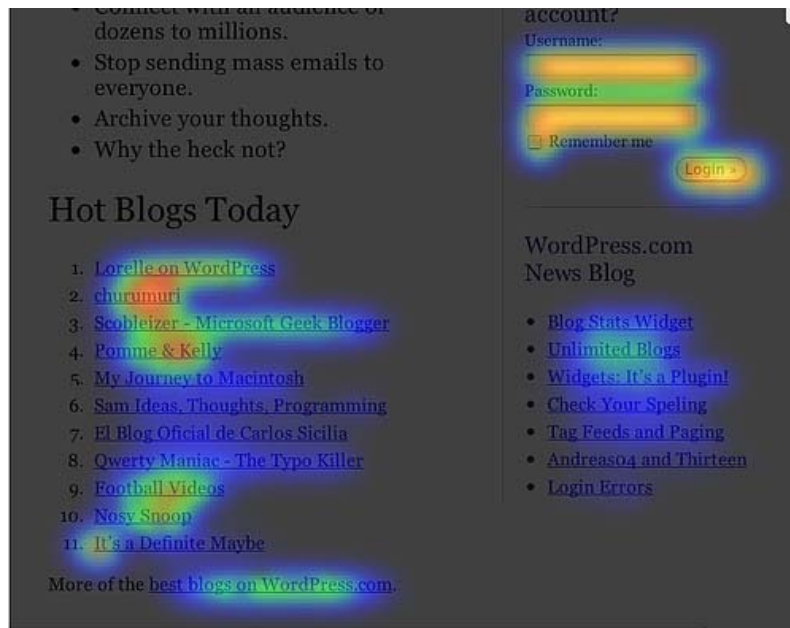
Το Time Magazine (Εικόνα 14) κάνει χρήση οπτικών κορυφών σε ένα χάρτη για την αναπαράσταση δεδομένων, τα οποία είναι ο πληθυσμός των πόλεων μιας χώρας (εδώ της Αμερικής).



Εικόνα 14 : Εφαρμογή απεικόνισης δεδομένων Time Magazine.

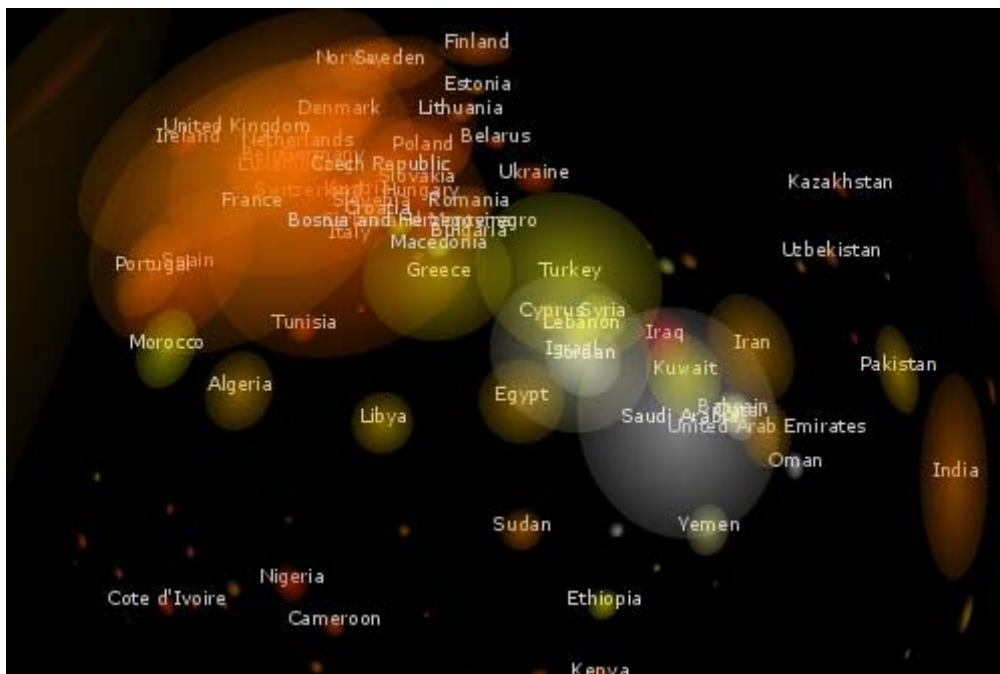
Το crazyEgg (Εικόνα 15) μπορεί να χρησιμοποιηθεί σε μια σελίδα με σκοπό να καταγράψει την συμπεριφορά των επισκεπτών. Όσο πιο πολλά τα κλικ των επισκεπτών τόσο πιο κόκκινη γίνεται η περιοχή γύρω από το link(σύνδεση), κάτι σαν μετρητή του πιο καυτού link της σελίδας. Η εφαρμογή αυτή διαθέτει και άλλους

τρόπους αναπαράστασης της πληροφορίας αυτής όπως σε λίστες ή σε μικρά παράθυρα.



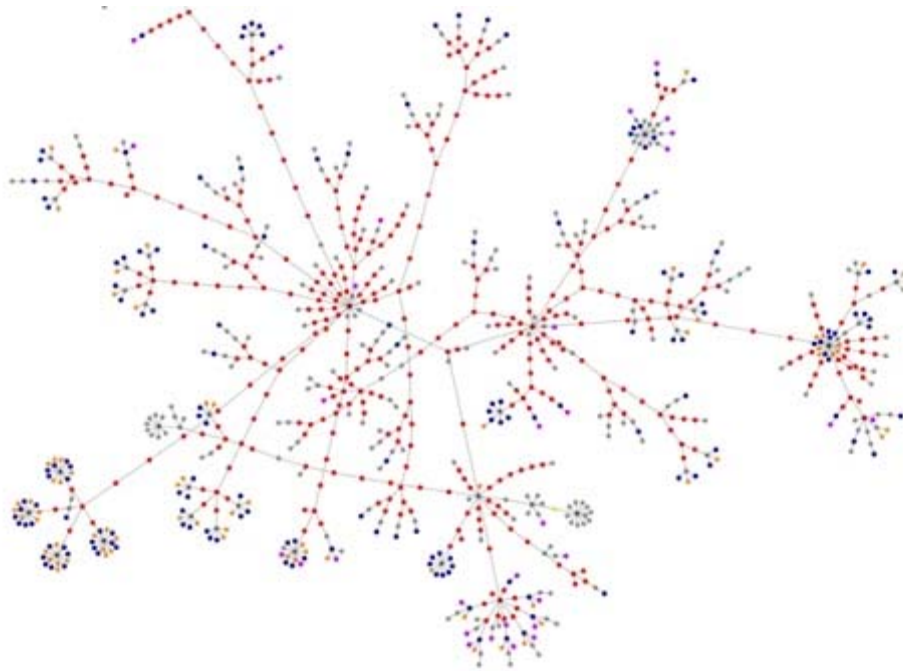
Εικόνα 15 : Εφαρμογή απεικόνισης δεδομένων crazyEgg.

Το Three Views (Εικόνα 16) παρουσιάζει δεδομένα σε χρωματιστούς κύκλους, τα δεδομένα είναι τα χρήματα που ξοδεύει η εκάστοτε χώρα για την αγορά στρατιωτικού εξοπλισμού(μέγεθος του κύκλου) και οι απολαβές από την χρήση του(χρώμα κύκλου).



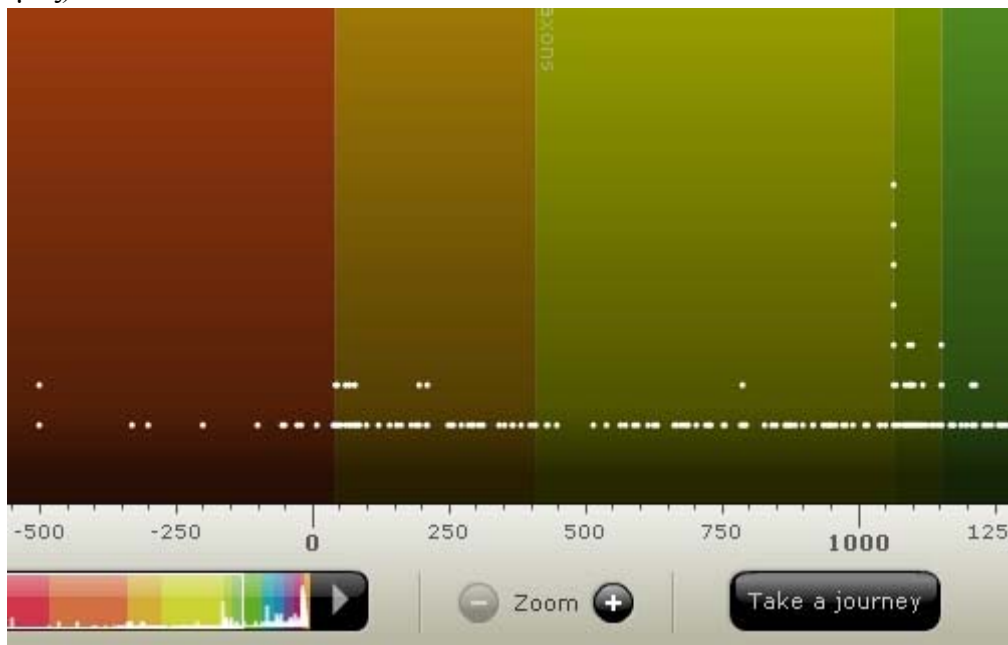
Εικόνα 16 : Εφαρμογή απεικόνισης δεδομένων Three Views.

Στο Websites as graphs (Εικόνα 17), η εφαρμογή παρουσιάζει μια ιστοσελίδα σε μορφή γράφου. Η ιεραρχία του γράφου έχει κατανημηθεί βάση των ιδιοτήτων μιας σελίδας όπως συνδέσεις, πίνακες, ετικέτες, φωτογραφίες, φόρμες.



Εικόνα 17 : Εφαρμογή απεικόνισης δεδομένων Websites as graphs.

Το Interactive History Timeline (Εικόνα 18) είναι μια διαδραστική μπάρα χρόνου. Αυτή που εμφανίζεται στο σχήμα παρουσιάζει την ιστορία της Αγγλίας. Τα διάφορα χρώματα δείχνουν μια συγκεκριμένη περίοδο εποχής της χώρας(π.χ. πόλεμος).



Εικόνα 18 : Εφαρμογή απεικόνισης δεδομένων Interactive History Timeline.

Το Elastic Lists(Εικόνα 19) παρουσιάζει ένα είδος ελαστικών λιστών με δεδομένα. Τα δεδομένα είναι όλοι οι νομπελίστες του κόσμου. Στην πρώτη λίστα οι νομπελίστες είναι χωρισμένοι ανά χώρα καταγωγής. Στη δεύτερη λίστα ο διαχωρισμός γίνεται ανά δεκαετίες. Στην τελευταία λίστα ο διαχωρισμός είναι ανά

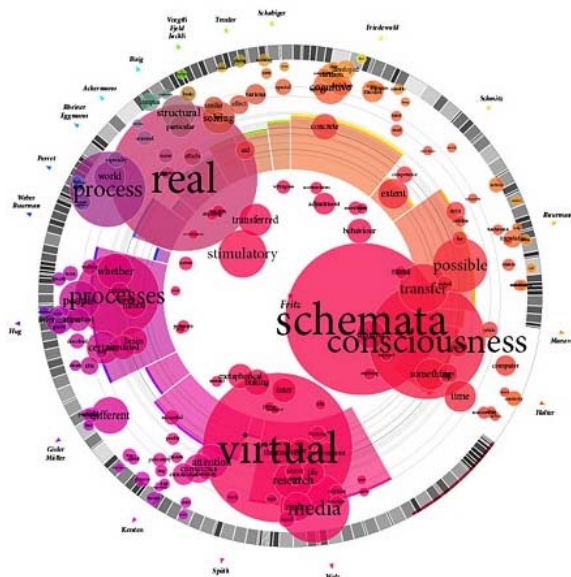
έτος. Σε όλες τις περιπτώσεις το τελικό αποτέλεσμα εξαρτάται από την επιλογή που κάνουμε στην προηγούμενη λίστα και που έχει ως αποτέλεσμα να αλλάζει η επόμενη λίστα. Επίσης, υποστηρίζονται διάφοροι τρόποι φιλτραρίσματος(π.χ. φύλλο).

COUNTRY	all	filter	lock	DECADE	all	filter	lock	YEAR	all	filter	lock
United States of America	281			2000s	56			2004	12		
				1990s	98			2003	9		
United Kingdom	105			1980s	97			2002	10		
France	49			1970s	103			2001	15		
Germany	44			1960s	79			2000	10		
Federal Republic of Germ	31			1950s	72			1999	7		
Switzerland	30			1940s	43			1998	12		
Sweden	30			1930s	56			1997	12		
the Netherlands	15			1920s	54			1996	13		
USSR	14			1910s	40			1995	10		
Italy	14			1900s	57			1994	10		
Denmark	13							1993	9		
Austria	12							1992	7		
Belgium	11							1991	7		
								1990	11		

Εικόνα 19 : Εφαρμογή απεικόνισης δεδομένων Elastic Lists

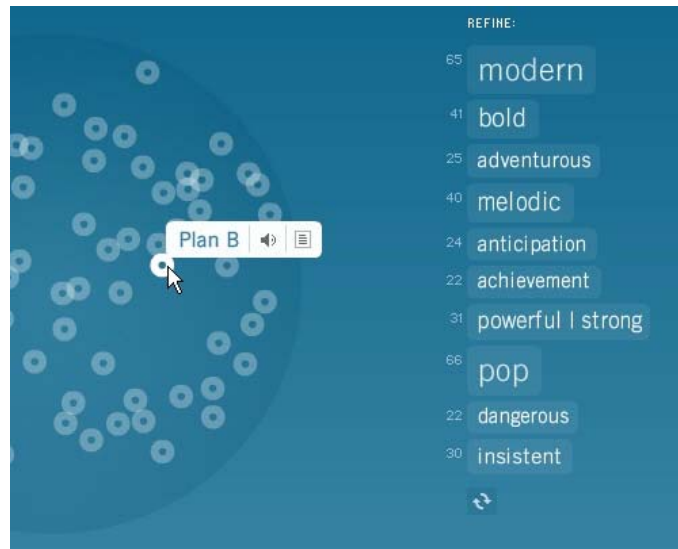
### 2.3.4 Εφαρμογές παρουσίασης συνδέσεων

Το Munterbund (Εικόνα 20) παρουσιάζει τα αποτελέσματα έρευνας γραφικών απεικονίσεων από ομοιότητες κείμενων σε δοκίμια βιβλίων. Η πρόκληση είναι να βρεις φόρμες γραφικών και τυπογραφικών παρουσιάσεων των ομοιοτήτων αυτών που να είναι όμορφες και ενημερωτικές. Μια προσπάθεια να γίνει ένα τέτοιο σύστημα αυτόματης δημιουργίας γραφικών προκαθορισμένων κανόνων είναι και αυτή η εφαρμογή.



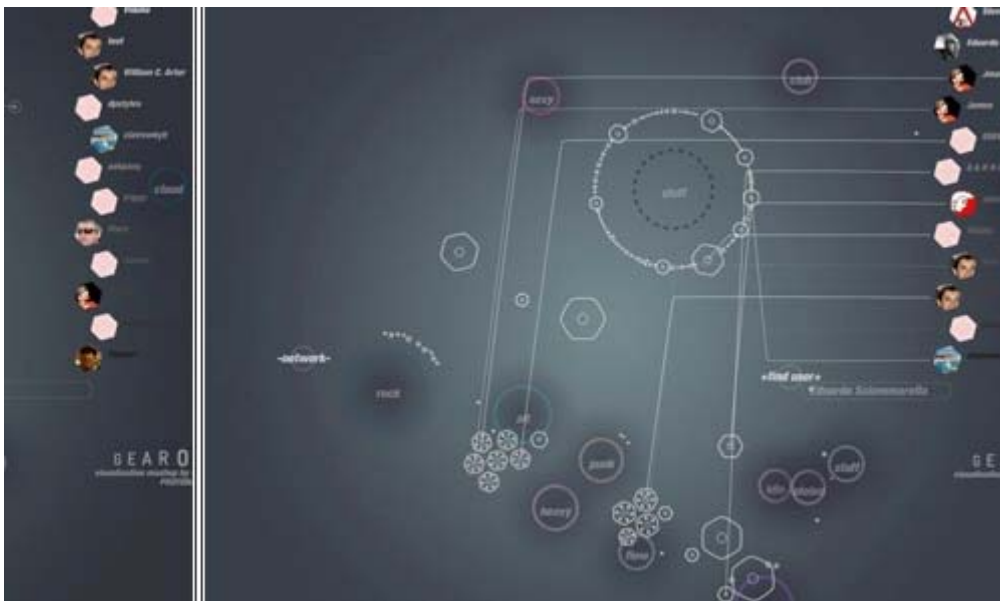
Εικόνα 20 : Εφαρμογή απεικόνισης δεδομένων Munterbund.

Το Burst Labs (Εικόνα 21) είναι μια εφαρμογή που προτείνει στους χρήστες όμοια ή συνδεδεμένα στοιχεία με την λέξη κλειδί αναζήτησης( αγαπημένοι τραγουδιστές, εκπομπές, ταινίες κτλ) μέσα σε μια φούσκα.



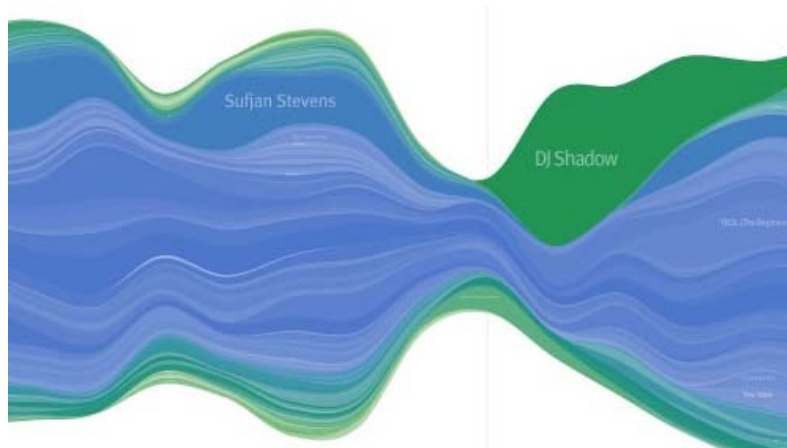
Εικόνα 21 : Εφαρμογή απεικόνισης δεδομένων Burst Labs.

Το Figd't Visualizer (Εικόνα 22) προσφέρει τη δυνατότητα οπτικής παρουσίασης ενός δικτύου. Η πληροφορία που παρουσιάζει ιδιαίτερα είναι οι τάσεις-προτιμήσεις μέσα στο διαδίκτυο των χρηστών. Όπως για παράδειγμα τι είδος μουσική ακούγεται περισσότερο (π.χ., ροκ ή κλασσική).



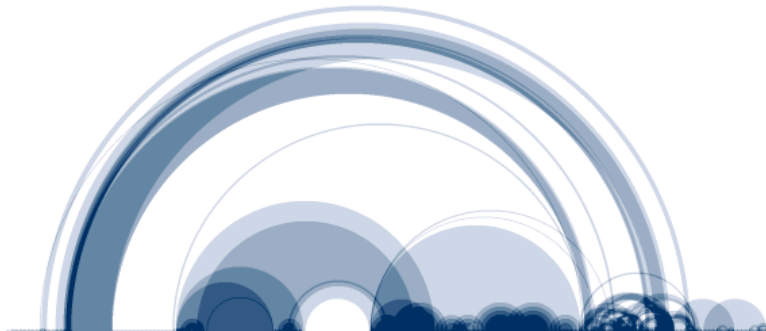
Εικόνα 22 : Εφαρμογή απεικόνισης δεδομένων Figd't Visualizer.

Στην Εικόνα 23 παρουσιάζεται ένα ιστόγραμμα το οποίο προσπαθεί να καταγράψει κατά προσέγγιση την περιγραφή του τι είδος μουσική ακούει αυτός που το δημιούργησε (Lee Byron).

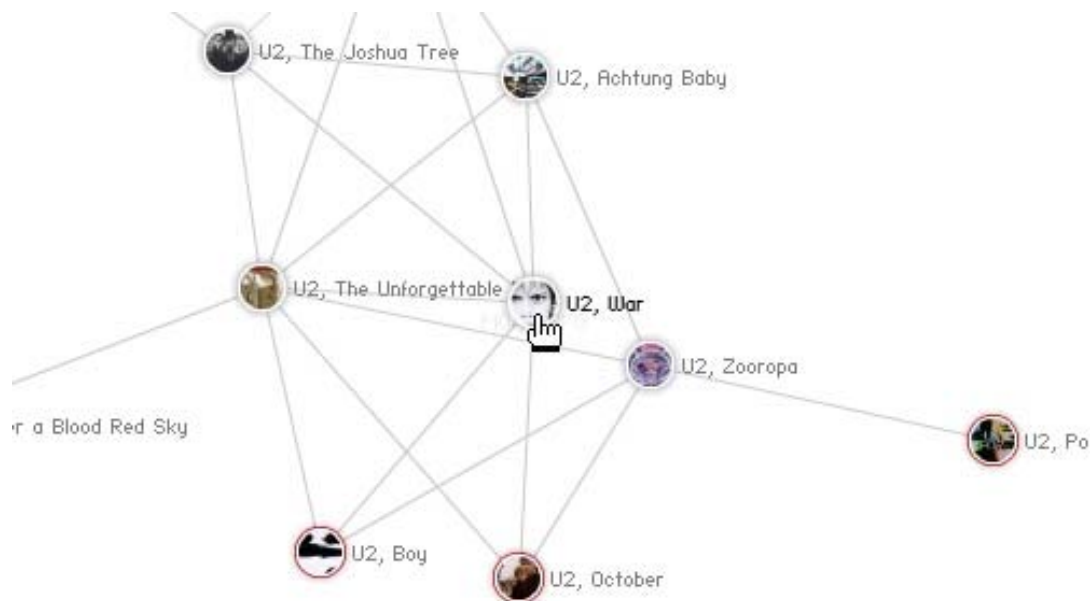


Εικόνα 23 : Εφαρμογή απεικόνισης δεδομένων του Lee Byron.

Shape of Song (σχήμα τραγουδιού) (Εικόνα 24), όπως λέει και το όνομα του παρουσιάζει ένα τραγούδι σε σχήμα. Τέλος, το Musicmap (Εικόνα 25), παρουσιάζει τις συνδέσεις ως συνδεδεμένες γραμμές, δημιουργώντας ένα δίκτυο.



Εικόνα 24 : Εφαρμογή απεικόνισης δεδομένων The Shape of Song.



Εικόνα 25 : Εφαρμογή απεικόνισης δεδομένων Musicmap.

### 2.3.5 Εφαρμογές παρουσίασης ιστοσελίδων

Η διαδραστική οπτική αναπαράσταση δεδομένων έχει χρησιμοποιηθεί ευρύτατα και σε εφαρμογές του Παγκόσμιου Ιστού. Ενδεικτικά το Spacetime (Εικόνα 26), παρουσιάζει το Google, Yahoo, Flickr, eBay και φωτογραφίες σε 3D απεικόνιση (3D πρόγραμμα πλοήγησης). Έτσι για παράδειγμα σε μια αναζήτηση του Google τα αποτελέσματα που θα βγουν θα είναι σε όμορφα 3D παράθυρα. Το UBrowser (Εικόνα 27), είναι μια εφαρμογή ανοιχτού κώδικα που προσφέρει ένα 3D πρόγραμμα πλοήγησης στο διαδίκτυο σε διάφορα σχήματα (π.χ. κύβος, κύκλος).



Εικόνα 26 : Εφαρμογή απεικόνισης δεδομένων Spacetime.



Εικόνα 27 : Εφαρμογή απεικόνισης δεδομένων UBrowser.

## 2.3.6 Εργαλεία για απεικόνιση δεδομένων

### 2.3.6.1 Prefuse

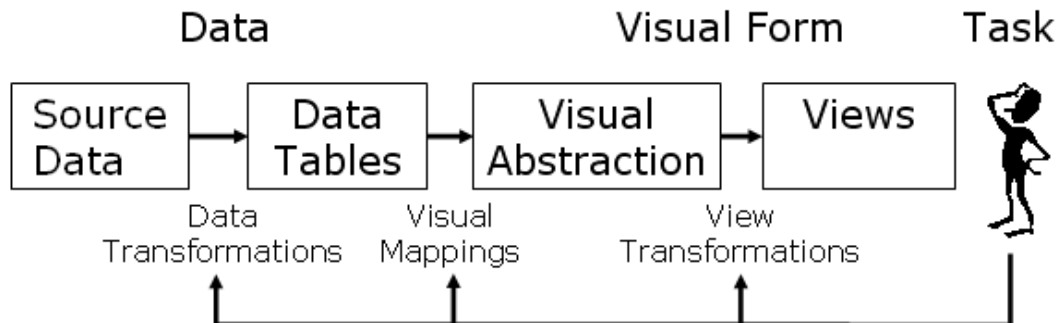
Ένα από τα σημαντικότερα εργαλεία (λογισμικό) για υλοποίηση πολλών από των παραπάνω διαδραστικών απεικονίσεων είναι το `prefuse`, χρησιμοποιεί την γλώσσα προγραμματισμού Java και είναι ανοιχτού κώδικα (open source). Διαθέτει επίσης και το `prefuse flare` εργαλείο που παρέχει αναπαραστάσεις και απεικονίσεις για ActionScript και το Adobe Flash Player. Το `prefuse` μπορεί να χρησιμοποιηθεί για να δημιουργήσει ανεξάρτητες εφαρμογές, απεικονίσεις ενσωματωμένες σε μεγαλύτερες εφαρμογές και web applets. Μερικά από τα κύρια χαρακτηριστικά του `prefuse` αναφέρονται παρακάτω.

- ❖ Πίνακες, γραφικές παραστάσεις, και δομές δεδομένων όπως δέντρα που υποστηρίζουν τις αυθαίρετες ιδιότητες στοιχείων, στοιχεία που συντάσσουν ευρετήριο και ερωτήσεις επιλογής, όλες με έναν πολύ αποδοτικό τρόπο διαχείρισης της μνήμης.
- ❖ Συστατικά για το σχεδιάγραμμα, το χρώμα, το μέγεθος, και την κωδικοποίηση μορφής, τις τεχνικές διαστρεβλώσεων, animation, και πολλά άλλα.
- ❖ Μια βιβλιοθήκη των ελέγχων αλληλεπίδρασης καθώς και διαδικασίες άμεσου-χειρισμού.
- ❖ Υποστήριξη animation μέσω μιας γενικής δραστηριότητας που σχεδιάζει το μηχανισμό.
- ❖ Δυνατότητα μετασχηματισμών που υποστηρίζουν panning και zooming συμπεριλαμβανομένης της γεωμετρικής και σημασιολογικής μεγέθυνσης.
- ❖ Δυναμικά ερωτήματα(query) για το φιλτράρισμα των στοιχείων.
- ❖ Ενσωματωμένη αναζήτηση κειμένου που χρησιμοποιεί διάφορες διαθέσιμες μηχανές.
- ❖ Μια μηχανή προσομοίωσης για την δυναμική απεικόνιση και το animation.
- ❖ Ευελιξία για τις πολλαπλές εμφανίσεις, περιλαμβανομένου "overview+detail" και "small multiples" απεικονίσεις.
- ❖ Ενσωματωμένη γλώσσα σαν την Sql για την υποβολή ερωτημάτων στις δομές δεδομένων του Prefuse και για τη δημιουργία πεδίων με αντλημένα δεδομένα(derived data fields).
- ❖ Η υποστήριξη για την ισχύ των ερωτημάτων στις βάσεις δεδομένων Sql και τη χαρτογράφηση αυτών των αποτελεσμάτων σε δομές δεδομένων Prefuse.
- ❖ Απλά, φιλικά APIs για τη δημιουργία επεξεργασίας συγκεκριμένων διαδικασιών, αλληλεπιδράσεων, και rendering συστατικών.

Ο σχεδιασμός του εργαλείου `prefuse` είναι βασισμένος πάνω στο μοντέλο πληροφορία-απεικόνιση-αναφορά, μια αρχιτεκτονική λογισμικού που σπάει την



διαδικασία της απεικόνισης σε συγκεκριμένα βήματα, από την απόκτηση δεδομένων και μοντελοποίηση σε οπτική κωδικοποίηση δεδομένων στην παρουσίαση διαδραστικών προβολών. Αυτή η διαδικασία περιγράφεται στην παρακάτω εικόνα (Εικόνα 28).



Εικόνα 28 : Διάγραμμα απεικόνισης της πληροφορίας του μοντέλου αναφοράς του prefuse.

### 2.3.6.2 Swivel

Αυτό το εργαλείο δημιουργεί εύκολα διαγράμματα πίτας(pie charts), απλά διαγράμματα(diagrams) και ιστογράμματα(histograms). Επίσης μέσω Swivel API διαθέτει την δυνατότητα προσθήκης είδη υπαρχόντων μεθόδων απεικόνισης.

### 2.3.6.3 JGraph

Το JGraph είναι ένα επιπλέον εργαλείο το οποίο μπορεί να απεικονίσει οπτικά δεδομένα. Επειδή η πτυχιακή κάνει χρήση αυτού του εργαλείου για την απεικόνιση των δεδομένων μας θα το αναλύσουμε παρακάτω.

## 3 Αναπαράσταση δεδομένων με το JGraph

### 3.1 Εισαγωγικά

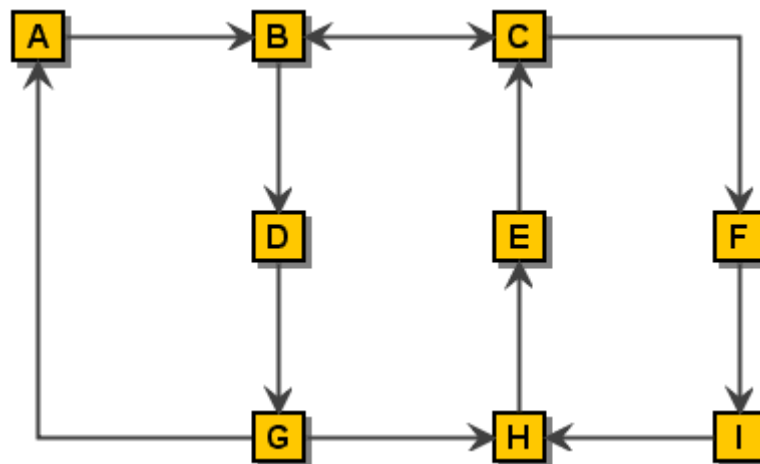
Επειδή η πτυχιακή αυτή χρησιμοποιεί έννοιες από την *θεωρία των γράφων* θα γίνει μια σύντομη περιγραφή της, ώστε να πάρουμε μια εικόνα.

Η αρχές της θεωρίας των γράφων παρουσιάστηκαν το 1736 από τον Leonhard Euler. Ο Euler είχε να λύσει το πρόβλημα των γεφυριών της πόλης Königsberg, και έπρεπε να βρει αν μπορεί να γίνει ο γύρος της πόλης περνώντας μόνο μια φορά από κάθε ένα των επτά γεφύρια του ποταμού (Pregel) της πόλης (έτσι έχουμε την διαδρομή Euler, *Euler path*). Κατάφερε με την βοήθεια των γράφων να θέσει ένα χρήσιμο όρο και έτσι αναβλήθηκε ο γύρος αυτός της πόλης.

Η θεωρία των γράφων είναι κομμάτι από τα μαθηματικά, που ενώνει τις έννοιες των γράφων και των σχέσεων τους και τις αναλύει. Οι γράφοι είναι μια μέθοδο για την διατύπωση και λύση πολλών προβλημάτων που συναντάμε, σε

- δίκτυα και συστήματα τηλεπικοινωνιών (π.χ. το Internet),
- χάρτες - επιλογή δρομολογίων,
- προγραμματισμό εργασιών (scheduling),
- ανάλυση προγραμμάτων (flow charts).

Γενικά ένας *γράφος* ( $G$ ) είναι ένα σύνολο από *κόμβους* ( $V$ ) που ενώνονται με *ακμές* ( $E$ ),  $G=(V,E)$ . Ένας απλός γράφος εμφανίζεται στην Εικόνα 29. Στη συνέχεια θα γίνει μια αναφορά σε βασικούς ορισμούς των γράφων.

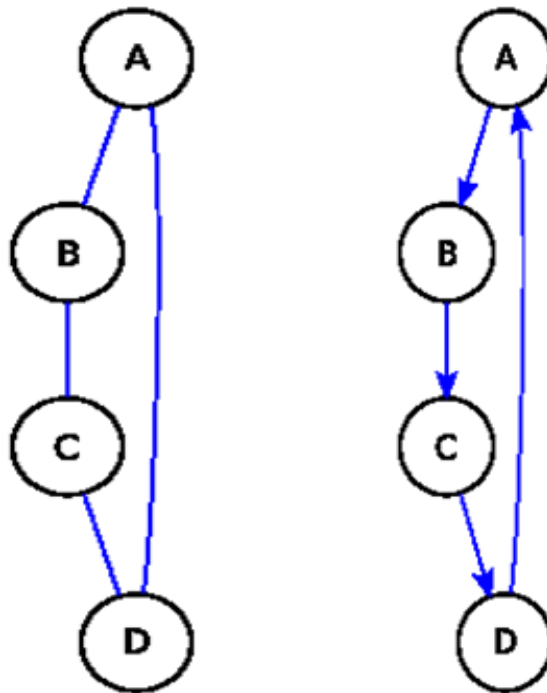


Εικόνα 29 : Ένας απλός γράφος(κατευθυνόμενος).

### 3.2 Ορισμοί γράφων

Η βασική ορολογία ενός γράφου περιλαμβάνει στοιχειώδεις έννοιες από τις οποίες μπορούμε να οριοθετήσουμε επιπλέον παράγωγες έννοιες. Αρχικά θα αναφερθούμε σε στοιχειώδεις έννοιες. Ένας γράφος ( graph ) είναι ένα σύνολο από κόμβους που ενώνονται με ακμές. Ο γράφος ορίζεται πλήρως από τους κόμβους και τη συνδεσμολογία τους. Αν οι ακμές είναι προσανατολισμένες (ορίζονται δηλαδή από

διατεταγμένα ζεύγη κόμβων) τότε ο γράφος λέγεται κατευθυνόμενος (directed) (Εικόνα 30). Αν οι ακμές δεν είναι προσανατολισμένες (ορίζονται δηλαδή από μη διατεταγμένα ζεύγη κόμβων) τότε ο γράφος λέγεται μη κατευθυνόμενος (undirected) (Εικόνα 30). Αν οι ακμές είναι συνδεδεμένες με κάποια αξία (βάρους) τότε ο γράφος λέγεται *σταθμισμένος* (weighted). *Πλήρης* (Complete) ορίζεται ο μη κατευθυνόμενος γράφος που περιέχει ακμές που ενώνουν κάθε ζεύγος κόμβων. Σε σχέση με την πυκνότητα υπάρχουν δύο κύρια είδη γράφων. *Αραιός* καλείται ο γράφος που περιέχει λίγες σχετικά ακμές (λ.χ. για  $n$  κόμβους ο αριθμός των ακμών  $< n \log n$ ). *Πυκνός* (Dense) καλείται ο γράφος από τον οποίο απουσιάζουν λίγες σχετικά ακμές.



Εικόνα 30 : (1<sup>ος</sup> γράφος) μη κατευθυνόμενος ,(2<sup>ος</sup> γράφος) κατευθυνόμενος.

Με βάση τα παραπάνω μπορούμε επίσης να ορίσουμε:

- ❖ *Διαδρομή* (path) καλείται μια διάταξη κόμβων οι οποίοι ενώνονται με ακμές.
- ❖ *Απλή διαδρομή* (simple path) καλείται η παραπάνω διάταξη όταν κάθε κόμβος εμφανίζεται μια μόνο φορά.
- ❖ *Κύκλος* (cycle) καλείται η διαδρομή δύο ή περισσότερων κόμβων που καταλήγει στον κόμβο αρχής.
- ❖ *Απόσταση* (distance) μεταξύ δύο κόμβων καλείται το μήκος της συντομότερης διαδρομής που τους ενώνει.
- ❖ *Διαδρομή Euler* (Euler path) καλείται η διαδρομή ή οποία περνάει από όλες τις ακμές του γράφου ακριβώς μια φορά.
- ❖ *Διαδρομή Hamilton* (Hamilton path) καλείται η διαδρομή ή οποία περνάει από όλους τους κόμβους του γράφου ακριβώς μια φορά.
- ❖ Γράφοι που περιέχουν τις παραπάνω διαδρομές ονομάζονται *αντίστοιχα*.

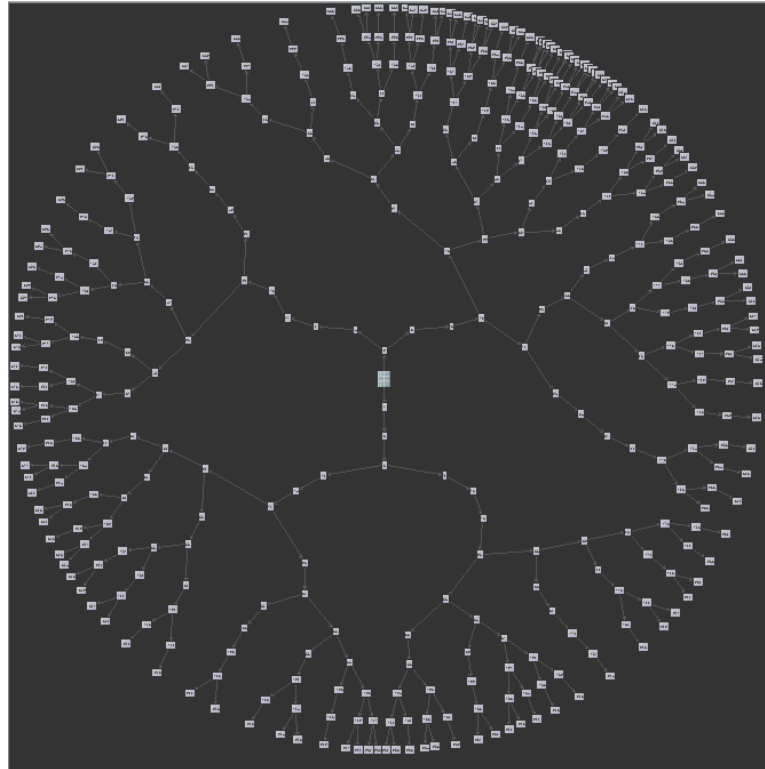
- ❖ *Συνεκτικός* (connected) ονομάζεται ο γράφος για τον οποίο υπάρχει διαδρομή από κάθε κόμβο σε κάθε άλλο κόμβο.
- ❖ *Βαθμός* ενός κόμβου σε έναν μη κατευθυνόμενο γράφο καλείται ο αριθμός των συνδεδεμένων με αυτόν ακμών.
- ❖ *Βαθμός εισόδου* (in-degree) σε έναν κατευθυνόμενο γράφο καλείται ο αριθμός των ακμών που καταλήγουν σε αυτόν.
- ❖ *Βαθμός εξόδου* (out-degree) σε έναν κατευθυνόμενο γράφο καλείται ο αριθμός των ακμών που ξεκινούν από αυτόν.
- ❖ *Υπογράφος* (subgraph)  $B$  ενός γράφου  $A$  καλείται ο γράφος του οποίου όλες οι ακμές και οι κόμβοι περιέχονται στον  $A$ .
- ❖ *Δένδρο σκελετός* (spanning tree) ενός γράφου καλείται ο υπογράφος που περιέχει όλους του κόμβους αλλά μόνο όσες ακμές απαιτούνται για να σχηματιστεί δένδρο.

Οι ορισμοί από την θεωρία των γράφων βρίσκονται και στις οπτικές απεικονίσεις δεδομένων που χρησιμοποιούν την βοήθεια των γράφων. Είναι πολλά τα εργαλεία-λογισμικά που χρησιμοποιούν τους γράφους για την παρουσίαση των δεδομένων, μερικά από τα αποτελέσματα που καταφέρνουν αυτά είδαμε παραπάνω. Παρακάτω θα πούμε για το JGraph το εργαλείο που χρησιμοποιήσαμε στην παρόν πτυχιακή για να παρουσιάσουμε τα δεδομένα μας.

### 3.3 Τι είναι το JGraph

Το JGraph είναι ένα λογισμικό που βοηθά στην παρουσιάζει δεδομένων σε μορφή γράφων. Είναι μια «ώριμη» και πλούσια σε χαρακτηριστικά βιβλιοθήκη κώδικα, είναι γραμμένη στην γλώσσα προγραμματισμού Java. Οι κλάσεις των πακέτων της βιβλιοθήκης του είναι γραμμένες με τέτοιο τρόπο ώστε να συνεργάζεται άψογα με τα εργαλεία του Swing πακέτου της Java, επίσης βασίζονται στο MVC (Model View Control) πρότυπο του Swing. Αυτό οφείλεται στο ότι το JGraph(η κλάση) είναι μια επέκταση του JComponent, που είναι η βασική κλάση του Swing για όλα τα συστατικά(components), σαν σχεδιασμός το JGraph έχει πολλά κοινά με αυτόν της κλάσης JTree.

Είναι λογισμικό ανοιχτού κώδικα (open source), οι εφαρμογές που χρησιμοποιούν το JGraph μπορούν να τρέξουν σε συστήματα που χρησιμοποιούν από την 1.4 έκδοση της Java και νεώτερες. Διαθέτει πλήθος λειτουργιών σχεδιασμού γράφων για client-side ή server-side εφαρμογές. Το JGraph είναι απλό στη χρήση έχοντας ταυτόχρονα ένα πολύ δυνατό API μέσω του οποίου μπορούμε να αναπαραστήσουμε, αλληλεπιδράσουμε, αυτόματα layout και να παράγουμε ανάλυση του γράφου (βλέπε Εικόνα 31).



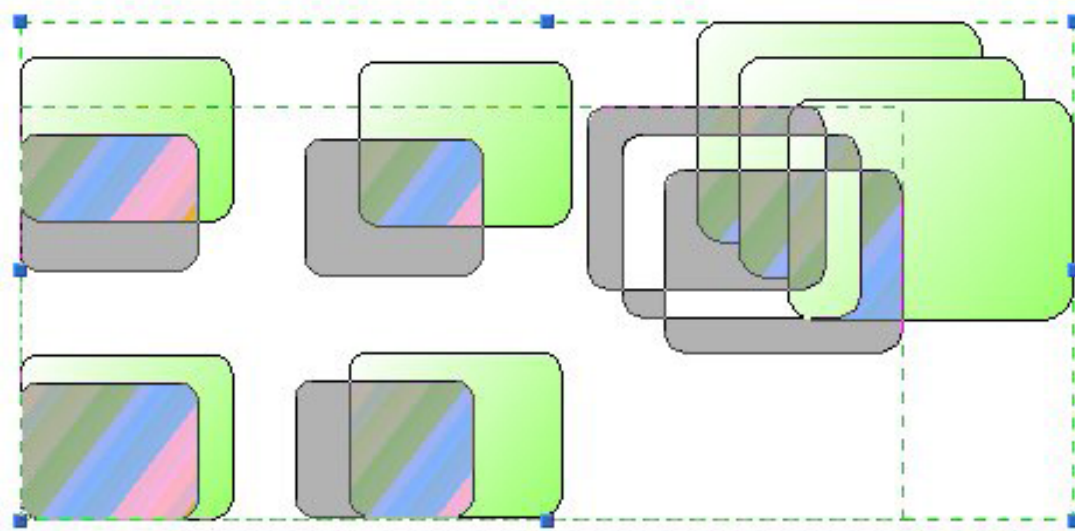
Εικόνα 31 : Οπτική αναπαράσταση ενός γράφου(radial tree layout) με τη χρήση του JGraph.

Τα δεδομένα που μπορούν να απεικονιστούν με την βοήθεια του JGraph μπορεί να προέρχονται από διάφορους κόσμους των πληροφοριών όπως:

- Διαγράμματα διαδικασιών
- Ροές εργασιών και BPM
- Πίνακες ρών
- Ροές κυκλοφορίας
- Αναπαράσταση βάσεων δεδομένων και WWW
- Παρουσίαση δικτύων και τηλεπικοινωνιών
- Χάρτες
- GIS
- Διαγράμματα UML
- Ηλεκτρονικά κυκλώματα
- VLSI
- CAD
- Χρηματιστήριο
- Βιοχημεία
- Οικολογία, κ.α.

Μια εφαρμογή φτιαγμένη με την χρήση του JGraph διαθέτει την ιδιότητα της *αλληλεπίδρασης*, δηλαδή διαθέτει την δυνατότητα τροποποίησης του μοντέλου του γράφου δια μέσο της εφαρμογής του GUI. Υποστηρίζει μεταφορά και κλωνοποίηση κελιών, μεγέθυνση και ανασχηματισμό, σύνδεση και αποσύνδεση, μεταφορά από εξωτερικές πηγές, αλλαγές στα κείμενα των κελιών(όνομα), αλλαγές χρώματος, φόντου κ.α (βλέπε Εικόνα 32). Είναι πολλά τα προσόντα που προσφέρει η

αλληλεπίδραση των εφαρμογών του JGraph και εξαρτάται από την ευελιξία προγραμματισμού της από τον εκάστοτε προγραμματιστή.



Εικόνα 32 : Προβολή αλλαγής μεγέθους κελιών ενός γράφου.

Όπως έχουμε προαναφέρει μια απεικόνιση ενός γράφου, αποτελείται από κόμβους(κελιά, cells) και τις συνδέσεις(που είναι γραμμές, edges) μεταξύ τους. Τα στοιχεία αυτά μπορεί να τοποθετηθούν οπουδήποτε στον χώρο απεικόνισης. Η διάταξη(layout) των στοιχείων ενός γράφου είναι πολύ σημαντική στην απόδοση πληροφορίας. Κάθε διαφορετικός τρόπος διάταξης αποδίδει διαφορετική πληροφορία σε αυτόν που βλέπει τον γράφο. Το JGraph διαθέτει μερικούς έτοιμους τρόπους διάταξης τους οποίους μπορεί ο χρήστης να προσθέσει εύκολα στον κώδικα της εφαρμογής του. Ακόμα δίνεται η δυνατότητα δημιουργίας καινούργιου τρόπου διάταξης ο οποίος να προσαρμόζεται στις ανάγκες του εκάστοτε χρήστη.

Η χωροδιάταξη των στοιχείων ενός γράφου αλλάζει σε κάθε τρόπο απεικόνισης(layout). Οι τρόποι απεικόνισης που διαθέτει το JGraph και χρησιμοποιούμε για τις ανάγκες της πτυχιακής είναι:

- Radial tree, ξεκινάει παίρνοντας το πρώτο κελί και τοποθετώντας στο κέντρο του σχεδιαγράμματος (κόμβος ρίζας). Στη συνέχεια, σχεδιάζει τα κελιά με τα οποία συνδέεται, σε ίσες αποστάσεις μεταξύ τους. Ακολούθως, παίρνει ένα – ένα τα προηγούμενα κελιά και τα συνδέει με τα σχετιζόμενα τους. Αυτή η διαδικασία συνεχίζεται για όσο υπάρχουν σχετιζόμενα κελιά. Τέλος, όταν δεν υπάρχουν άλλα σχετιζόμενα κελιά ξεκινάει πάλι απ' την αρχή, παίρνοντας τον επόμενο κόμβο ρίζας.
- Hierarchical, σε ένα ιεραρχικό μοντέλο δεδομένων (Hierarchical), τα δεδομένα είναι οργανωμένα σε ένα δέντρο, σαν δομή. Η δομή επιτρέπει την επανάληψη των πληροφοριών χρησιμοποιώντας γονέας - παιδί σχέσεις (κάθε γονέας μπορεί να έχει πολλά παιδιά, αλλά το κάθε παιδί έχει ένα μόνο γονέα).
- Simple Circle, στο συγκεκριμένο Layout, τα κελιά τοποθετούνται πάνω σ' ένα κύκλο σε σταθερές αποστάσεις μεταξύ τους.

- Fast Organic, η χρήση αυτού του τρόπου απεικόνισης εφαρμόζεται συνήθως σε μικρούς γράφους και απλούς γράφους. Το αποτέλεσμα της χωροδιάταξης βασίζεται σε αλγόριθμους και η μορφή του μοιάζει σαν το dna.
- Organic, βασίζεται στα παρακάτω κριτήρια, διανέμει τα κελιά ομαλά, το μήκος των συνδέσεων μεταξύ των κελιών ομοιόμορφο, σμικρύνει τα σταυροδρόμια των συνδέσεων, κρατάει τους κόμβους ώστε να μην έρθουν πολύ κοντά στους κόμβους.
- Self Organized,
- Spring Layout, είναι ένας γράφος που μιμείται ένα σύστημα μορίων και τοποθετεί τα κελιά.

### 3.4 Το πρόβλημα που αντιμετωπίζει η πτυχιακή

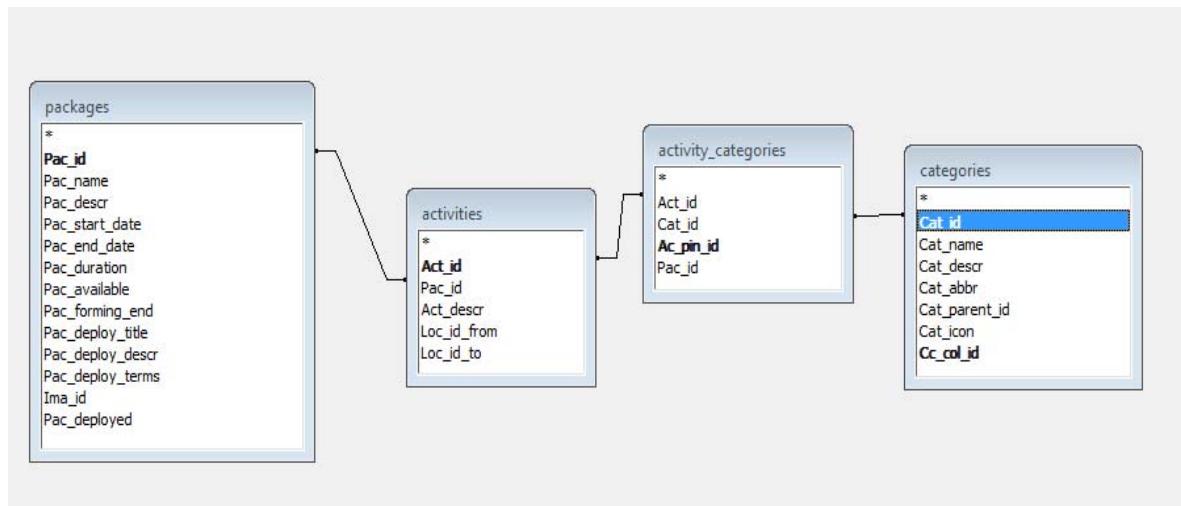
#### 3.4.1 Ποία είναι τα δεδομένα

Όπως αναφέρθηκε και στο 2.1 υπάρχουν πολλοί τρόποι να αποθηκεύσουμε και να διαχειριστούμε δεδομένα . Απ' αυτά εμείς θα ασχοληθούμε με τις βάσεις δεδομένων και πιο συγκεκριμένα , με δυο από τις πιο διαδεδομένες , τη MySQL και τον SQL Server2000. Η επιλογή της MySQL , έγινε λόγω του ότι είναι ανοιχτού κώδικα και της ευρείας διάδοσης της. Για τον SQL Server2000 , η επιλογή έγινε βάση της ευκολίας διαχείρισης .

Για να διαχειριστούμε τη MySQL βάση μας , θα χρησιμοποιήσουμε το MySQL administrator . Ένα πρόγραμμα, με GUI, που μας δίνει τη δυνατότητα να δημιουργήσουμε βάσεις δεδομένων, στις οποίες μπορούμε να προσθέσουμε πίνακες και μέσα σ' αυτούς πεδία, όπως επίσης και να προσδιορίσουμε τους τύπους των πεδίων και τις σχέσεις. Επίσης, έχουμε τη δυνατότητα να δημιουργήσουμε ερωτήματα, μέσω του MySQL Query Browser.

Όσο αφορά τον SQL Server2000 και τη διαχείριση μιας βάσης, χρησιμοποιούμε τον Enterprise Manager και το SQL Query Analyzer , που μας προσφέρουν παραπλήσιες δυνατότητες με το MySQL administrator και Query Browser, αντιστοίχως.

Η βάση, που χρησιμοποιήσαμε για δοκιμές, είναι μια παραλλαγή αυτής που χρησιμοποιείται στο portal eKoNEΣ . Σ ' αυτήν υπάρχουν τέσσερις πίνακες , ο activities , ο activities\_categories, ο categories και ο packages. Οι συνδέσεις μεταξύ τους είναι:



### 3.4.2 Πως αναπαριστώνται?

Οι υπολογιστές μπορούν να κάνουν μερικά πράγματα εύκολα, που άνθρωπος δεν μπορεί όπως να διαχειρίζονται δεδομένα. Από την άλλη ο άνθρωπος μπορεί να κάνει κάποια άλλα πράγματα καλύτερα από τον υπολογιστή για παράδειγμα να παράγει ιδέες και μεθόδους. Για να λυθούν καλύτερα διάφορα προβλήματα απαιτείται η συνεργασία και των δύο. Η συνεργασία αυτών απαιτεί την επικοινωνία μεταξύ τους. Αλληλεπίδραση -Επικοινωνία μεταξύ ανθρώπου και υπολογιστή (EAY) δεν είναι τίποτα άλλο από μια διάταξη μετατροπής στοιχείων μιας οντολογίας που κατανοεί ο άνθρωπος σε στοιχεία μιας άλλης οντολογίας που αντιλαμβάνεται και ερμηνεύει ο υπολογιστής. Για να μπορέσει όμως να υπάρξει επικοινωνία ανθρώπου-υπολογιστή θα πρέπει να γεφυρωθεί το χάσμα μεταξύ τους. Την γεφύρωση αυτή αναλαμβάνει να πραγματοποιήσει η *διεπαφή χρήστη* που ορίζεται ως:

*.. το τμήμα εκείνο της επικοινωνίας ανθρώπου-υπολογιστή που αφορά τους τρόπους, τις μεθόδους και τα εργαλεία που υλοποιούν την επικοινωνιακή διάδραση ανάμεσα στις οντότητες του χρήστη και του υπολογιστικού συστήματος.*

Αλλιώς θα λέγαμε ότι διεπαφή είναι το σύνολο των τμημάτων ενός υπολογιστή μέσω των οποίων έρχονται σε επαφή και αλληλεπιδρούν ο άνθρωπος-χρήστης με το υπολογιστικό σύστημα. Τα τμήματα, μέρη μιας διεπαφής μπορεί να είναι είτε υλικά είτε λογισμικά.

Στην επικοινωνία αυτή σήμερα έχει επικρατήσει η αλληλεπίδραση που βασίζεται κυρίως στο οπτικό κανάλι. Οι διεπαφές των πρώτων υπολογιστών διέθεταν διακόπτες που αναλόγως αν τους άνοιγες ή τους έκλεινες εκτελούσαν τις αντίστοιχες εντολές. Αργότερα οι διακόπτες αντικαταστήθηκαν από εντολές που πληκτρολογούσε ο χρήστης με την βοήθεια του πληκτρολογίου. Κατόπιν εμφανίστηκε ο τύπος διεπαφής με μενού όπου ο χρήστης ακολουθούσε ένα έτυμο δομημένο μονοπάτι από εντολές μέχρι να φτάσει σε αυτήν της επιλογής του, χωρίς την να πληκτρολογεί τίποτα. Τη δεκαετία του 90 εμφανίστηκαν οι διεπαφές WIMP (Windows, Icons, Menus, Pointing), εδώ πολλές διεπαφές ανταγωνίζονται για την διεκδίκηση περιορισμένων υπολογιστικών πόρων (π.χ. οθόνη γραφικών), οι εφαρμογές παρουσιάζονται σε



παράθυρα ή αντιπροσωπεύονται ως εικονίδια και ο χρήστης μπορεί με το ποντίκι ή κάποια άλλη συσκευή κατάδειξης να επιλέξει εντολές από μενού. Με τον απευθείας χειρισμό του υπολογιστή μέσω εικονιδίων την ίδια περίπου χρονική περίοδο ολοκληρώνεται η εξέλιξη ενός de facto προτύπου διεπαφής που ισχύει μέχρι και σήμερα.

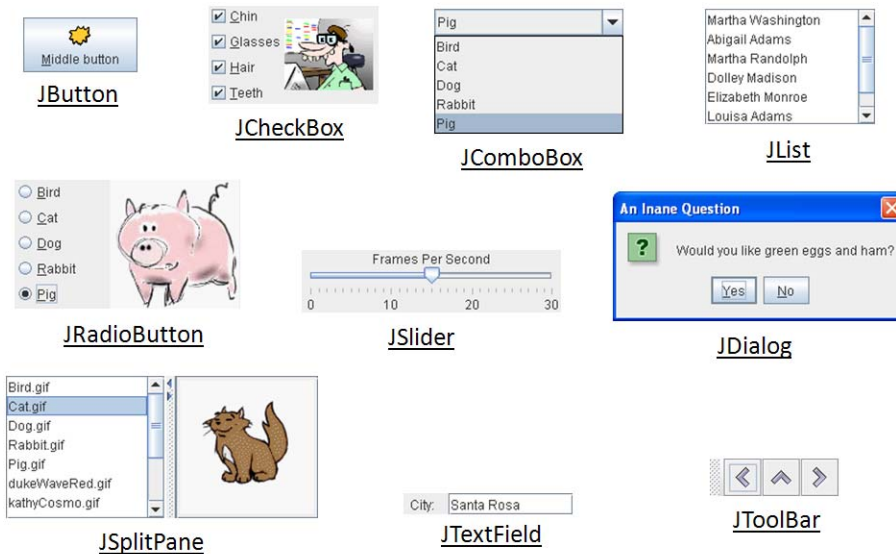
Ο άνθρωπος για να αλληλεπιδράσει με το περιβάλλον του και στην επικοινωνία με άλλους χρησιμοποιεί όλες του τις αισθήσεις. Έτσι και η διεπαφή ως εργαλείο αλληλεπίδρασης θα πρέπει να χρησιμοποιεί ερεθίσματα από όλες τις αισθήσεις του ανθρώπου ώστε να είναι πιο αποτελεσματική. Έτσι εκτός από την οπτική αλληλεπίδραση, σημαντικά αποτελέσματα φέρνει και η ακουστική, αφού είναι αποδεδειγμένο ότι ο άνθρωπος αντιδρά γρηγορότερα σε ακουστικά ερεθίσματα από ότι σε οπτικά.

Τα βασικά στυλ αλληλεπίδρασης διαχωρίζονται σε οπτικά και σε μη οπτικά. Πολλά στυλ από τα οπτικά στυλ αλληλεπίδρασης αναφέρθηκαν στην προ-προηγούμενη παράγραφο. Ένα άλλο δημοφιλές που συναντάμε συχνά στο διαδίκτυο είναι και οι φόρμες. Για τα μη οπτικά στυλ αλληλεπίδρασης δεν θα γίνει αναφορά αφού δεν χρησιμοποιούνται από την διεπαφή της παρόν πτυχιακής. Παρακάτω θα αναφερθούμε στο διαδραστικό σύστημα μια διεπαφής ώστε να δούμε και να μπορούμε αναλύσουμε αυτό της πτυχιακής αυτής.

### 3.5 Παρουσίαση συστήματος

Κάθε *διαδραστικό σύστημα* έχει τρία συνιστώσα μέρη, την παρουσίαση, το διάλογο, και τη συνιστώσα επικοινωνίας με το λειτουργικό μέρος του συστήματος. Ένα διαδραστικό σύστημα αποτελείται από τα διαδραστικά αντικείμενα ή αντικείμενα αλληλεπίδρασης. Είναι αυτά που συνθέτουν μια διεπαφή και τα οποία αναλαμβάνουν να μεταφέρουν στην εφαρμογή τις ενέργειες του χρήστη. Ο χειρισμός τους από τον χρήστη γίνεται με την βοήθεια διάφορων συσκευών εισόδου(πληκτρολόγιο, ποντίκι κ.α. ).Μερικά από τα βασικά αντικείμενα αλληλεπίδρασης που συναντάμε στην Java βλέπουμε στην επόμενη εικόνα(Εικόνα 33) Κάθε διαδραστικό αντικείμενο αποτελείται από τρεις μέρη :

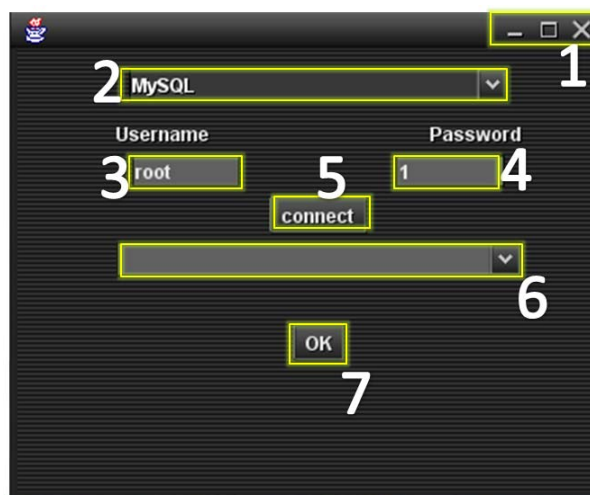
1. Την *όψη*, η οποία προσδιορίζει την εμφάνιση του αντικειμένου.
2. Την *συμπεριφορά* του αντικειμένου, η οποία προσδιορίζει τις μεθόδους αλληλεπίδρασης με το αντικείμενο
3. Την *διασύνδεση* με την εφαρμογή, η οποία προσδιορίζει τον τρόπο με τον οποίο ένα διαδραστικό αντικείμενο μεταφέρει τις δράσεις που δέχεται στο υπόλοιπο σύστημα.



Εικόνα 33 : Μερικά από τα αντικείμενα αλληλεπίδρασης της Java.

### 3.5.1 Διαδραστικά αντικείμενα της Διεπαφής μας

Παρακάτω θα δούμε τα αντικείμενα αλληλεπίδρασης που εμφανίζονται στην διεπαφή της πτυχιακής μας. Αφού θα δούμε την όψη τους, θα αναλύσουμε την συμπεριφορά τους και κατόπιν θα δούμε πως γίνεται η επικοινωνία με την εφαρμογή μας. Κάθε αντικείμενο συμβολίζεται και με ένα αριθμό τον οποίο θα αναφέρουμε κάθε φορά για το εκάστοτε αντικείμενο που γράφουμε. Η διεπαφή της πτυχιακής αποτελείται από δυο κομμάτια, αρχίζει από μια μικρή διεπαφή(Εικόνα 34) ή διεπαφή εισόδου και καταλήγει στην κύρια διεπαφή(Εικόνα 35) που διαθέτει το πλήθος των δυνατοτήτων της εφαρμογής.



Εικόνα 34 : Διεπαφή εισόδου της πτυχιακής μας.

Και οι δύο διεπαφές έχουν δημιουργηθεί όπως έχουμε πει με την βοήθεια της γλώσσας προγραμματισμού Java(Swing) και του JGraph.Όλα τα αντικείμενα είναι

τοποθετημένα μέσα σε διάφορα JPanel, τα οποία με την σειρά τους είναι όλα τοποθετημένα σε ένα JFrame, αυτό που φαίνεται στην εικόνα.

Το πρώτο αντικείμενο(νούμερο 1, Εικόνα 34) είναι γνωστό από όλες της παραθυρικές διεπαφές όσον αναφορά την όψη. Είναι τρία κουμπιά με τα τρία σύμβολά, τα οποία μπορούν να πατηθούν. Οι ενέργειές τους όταν πατηθούν(αρ.κλικ) στην εφαρμογή είναι ελαχιστοποίηση, μεγέθυνση παραθύρου και κλείσιμο εφαρμογής.

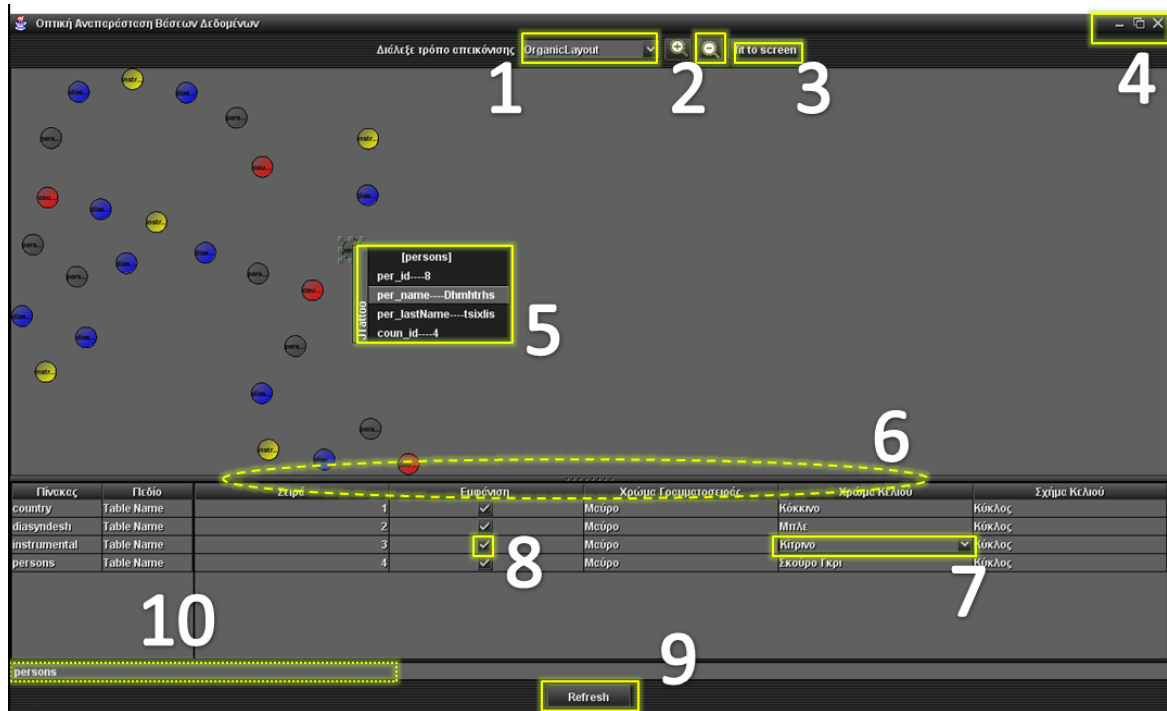
Η όψη του δεύτερου αντικείμενου(νούμερο 2, Εικόνα 34) παρουσιάζει ένα JComboBox. Στα δεξιά του υπάρχει ένα βελάκι, το οποίο αν το πατήσουμε(αρ.κλικ) θα εμφανίσει δύο επιλογές τις MySQL και SQL SERVER από τις οποίες μπορούμε να επιλέξουμε, με το ποντίκι πάλι, αυτή της αρεσκείας μας. Από μόνο του αυτό το αντικείμενο εκτός από το να κρατάει το επιλεγμένο από την λίστα (RDBMS) δεν προσφέρει τίποτα άλλο στην εφαρμογή.

Παρακάτω τα δύο επόμενα αντικείμενα (νούμερο 3-4, Εικόνα 34) είναι δυο JTextField. Μέσα σε JTextField μπορούμε να βάλουμε μέσω του πληκτρολογίου κείμενο αριθμούς και οτιδήποτε άλλους χαρακτήρες και αυτό να τους εμφανίσει, αφού το έχουμε επιλέξει με το ποντίκι. Η δουλειά τους είναι να κρατάνε τις αλλαγές που δέχονται από τον χρήστη σε κείμενο.

Επόμενο αντικείμενο είναι ένα κουμπί, JButton(νούμερο 5, Εικόνα 34). Το κουμπί αυτό μπορεί να πατηθεί, κάθε φορά που πατιέται είναι σε θέση να μεταβιβάσει την προγραμματισμένη ενέργεια στην εφαρμογή. Εδώ η ενέργεια αυτή είναι να διαβάσει την επιλογή που έχει το JComboBox και τους χαρακτήρες που έχουν τα δύο JTextField. Τα στοιχεία αυτά τα μεταφέρει σε μεταβλητές και ενεργοποιεί μέσω JDBC το RDBMS. Τα στοιχεία από τις μεταβλητές διαβάζονται από το RDBMS και αν είναι σωστά γίνεται η σύνδεση με αυτό. Σε περίπτωση σφάλματος εμφανίζεται παράθυρο λάθους JOptionPane με κατάλληλο μήνυμα, ώστε να ξαναδοκιμάσει ο χρήστης.

Αν τα στοιχεία που έδωσε είναι σωστά γίνεται η σύνδεση με το RDBMS, κατόπιν διαβάζονται όλες οι βάσεις(ΒΔ) που υπάρχουν σε αυτό και περνιούνται μια-μια με την σειρά στο επόμενο αντικείμενο αλληλεπίδρασης(νούμερο 6, Εικόνα 34). Ο χρήστης επιλέγει με ίδιο τρόπο όπως και στο νούμερο 2, αυτήν τη φορά όμως την βάση(ΒΔ) που θέλει.

Επόμενο αντικείμενο είναι πάλι ένα JButton(νούμερο 7, Εικόνα 34) που είναι προγραμματισμένο να παίρνει την επιλογή του JComboBox(νούμερο 6, Εικόνα 34). Την επιλογή αυτή του χρήστη την μεταφέρει στο RDBMS έτσι ώστε αυτό να διαθέσει στην εφαρμογή την επιλεγμένη βάση (ΒΔ). Επίσης ενεργοποιεί το δεύτερο(κύριο) μέρος της διεπαφής μας. Σε περίπτωση πάλι που η βάση(ΒΔ) που επέλεξε ο χρήστης δεν είναι έγκυρη ή είναι κενή, εμφανίζεται στον χρήστη παράθυρο λάθους JOptionPane με ανάλογο μήνυμα.



Εικόνα 35 : Κύρια διεπαφή της πτυχιακής.

Με το πάτημα του αντικείμενου νούμερο 7(Εικόνα 34) ενεργοποιείται η παραπάνω διεπαφή. Ταυτόχρονα η εφαρμογή διαβάζει την βάση δεδομένων(ΒΔ) που έχει επιλέξει ο χρήστης και δημιουργεί αντικείμενα για όλες της εγγραφές του κάθε πίνακα της. Τα αντικείμενα αυτά τα μεταβιβάζουμε στο JGraph το οποίο τα παρουσιάζει σε μορφή γράφου(κάθε αντικείμενο ένα κελί), η απεικόνιση αυτή γίνεται στο μεγάλο JPanel της διεπαφής όπως φαίνεται στην Εικόνα 35.

Αντικείμενο αλληλεπίδρασης νούμερο 1(Εικόνα 35). Από όψη το αντικείμενο αυτό είναι άλλο ένα JComboBox. Κάθε εγγραφή(Item) του JComboBox αντιπροσωπεύει ένα διαφορετικό τρόπο διάταξης(layout) του γράφου που απεικονίζει το JGraph. Κάθε φορά που ο χρήστης επιλέγει μέσω του αντικείμενου αυτού ένα διαφορετικό layout, η διάταξη του γράφου στο JPanel ανανεώνεται με τον καινούργιο αυτό layout.

Το επόμενο αντικείμενο είναι ένα JButton νούμερο 2(Εικόνα 35). Η ενέργεια που μεταφέρει, σε κάθε πάτημα στην εφαρμογή, είναι ότι ο χρήστης θέλει την σμίκρυνση(zoom out) του απεικονιζόμενου γράφου. Έτσι η εφαρμογή αλληλεπιδρά και μειώνει το μέγεθος του γράφου. Διπλά από αυτό από την αριστερή μεριά υπάρχει άλλο ένα JButton το οποίο κάνει την αντίστροφη δουλειά, δηλαδή μεγέθυνση(zoom in) του γράφου. Η μεγέθυνση-σμίκρυνση του γράφου γίνεται και με την ροδέλα του ποντικιού αναλόγως με την φορά που ρολάρουμε το wheel.

Συνεχίζοντας, έχουμε άλλο ένα JButton(νούμερο 3, Εικόνα 35) που με το πάτημα του φέρνουμε το μέγεθος του γράφου ίσο με το μέγεθος του JPanel που φιλοξενείται, με άλλα λόγια εφαρμόζεται στα όρια του πάνω. Αυτό μας βοηθά στο να έχουμε μια πλήρη εικόνα όλου του γράφου.

Το αντικείμενο αλληλεπίδρασης(νούμερο 4, Εικόνα 34) το έχουμε είδη αναφέρει πιο πάνω(νούμερο 1, Εικόνα 34).

Το επόμενο αντικείμενο είναι ένα JPopupMenu(νούμερο 5, Εικόνα 35). Το αντικείμενο αυτό εμφανίζεται μόνο όταν πατηθεί δεξή κλικ πάνω σε κελί του γράφου. Όταν λοιπόν αυτό πατηθεί εμφανίζεται ένα μενού με στοιχεία, τα οποία είναι οι διάφορες ιδιότητες του κελιού αυτού. Αλλιώς επειδή κάθε κελί είναι μια εγγραφή ενός πίνακα της βάσης(ΒΔ), τα στοιχεία του μενού είναι τα δεδομένα του κάθε πεδίου του πίνακα για την συγκεκριμένη εγγραφή.

Το επόμενο αντικείμενο είναι ένα JSplitPane (νούμερο 6, Εικόνα 35). Το αντικείμενο αυτό δεν διαφέρει από ένα κοινό JPanel, δέχεται και αυτό αντικείμενα, εδώ δέχεται ένα πίνακα JTable. Το JSplitPane μπορεί να αυξομειώσει το μέγεθος του, είτε προς τα πάνω, είτε προς τα κάτω. Ο πίνακας αυτός διαθέτει διάφορα διαδραστικά αντικείμενα, που σκοπός τους είναι η διαμόρφωση των κελιών του γράφου σύμφωνα με τις προτιμήσεις του χρήστη.

Πάμε να δούμε τα δυνατότητες μορφοποίησης της απεικόνισης του γράφου που είναι ενσωματωμένες μέσα στον πίνακα αυτό. Σαν διαδραστικό αντικείμενο ένα JTable αποτελείται από γραμμές και στήλες. Οι γραμμές αντιπροσωπεύουν τους πίνακες της επιλεγμένης βάσης(ΒΔ), ο αριθμός τους ισούται με τον αριθμό αυτών. Οι στήλες του πίνακα είναι επτά και περιέχουν μέσα αλλά αντικείμενα αλληλεπίδρασης.

Η πρώτη στήλη(Πίνακας), περιέχει το όνομα του κάθε πίνακα της βάσης(ΒΔ). Στην δεύτερη στήλη(Πεδίο), για κάθε εγγραφή υπάρχει ένα JComboBox(όπως νούμερο 7, Εικόνα 35) του οποίου τα στοιχεία είναι τα πεδία του κάθε πίνακα. Στην Τρίτη στήλη(Σειρά), σε κάθε εγγραφή υπάρχει ένα JComboBox(όπως νούμερο 7, Εικόνα 35) που τα στοιχεία του είναι αριθμοί (1, 2, 3,..μέχρι  $n$ , όπου  $n$  = αριθμός πινάκων). Η στήλη αυτή απευθύνεται μόνο στο MyLayout1, τι κάνει ακριβώς το layout αυτό θα δούμε παρακάτω. Σε ολόκληρη την επόμενη στήλη(Εμφάνιση) φιλοξενείται ένα JCheckBox(όπως νούμερο 8, Εικόνα 35) το οποίο κρατάει την επιλογή του χρήστη, αν θέλει δηλαδή να εμφανίζεται ή όχι ο πίνακας στον γράφο.

Σε κάθε μια από τις επόμενες τρεις στήλες(Χρώμα γραμματοσειράς, Χρώμα κελιού, Σχήμα κελιού) είναι ενσωματωμένο και ένα JComboBox(όπως νούμερο 7, Εικόνα 35) για κάθε εγγραφή του. Τα στοιχεία των JComboBox που υπάρχουν στις δυο στήλες (Χρώμα γραμματοσειράς, Χρώμα κελιού) είναι τα ίδια(Ανοιχτό Γκρι, Άσπρο, Βαθύ Πορφυρό, Γκρι, Κίτρινο, Κόκκινο, Μαύρο, Μπλε, Πορτοκαλί, Πράσινο, Ροζ, Σκούρο Γκρι) και αλλάζουν το χρώμα γραμματοσειράς και το χρώμα κελιού αντίστοιχα. Τα στοιχεία του JCheckBox της τελευταίας στήλης(Σχήμα κελιού) περιέχουν τα διάφορα σχήματα που μπορούν(Διαμάντη, Έλλειψη, Ορθογώνιο Παρα/γραμμο, Στρογγυλό Ορθογώνιο, Κύκλος) να παρουσιασθούν τα κελιά.

Παραπάνω αναφέραμε τις δυνατότητες μορφοποίησης του γράφου που υπάρχουν μέσα στο JTable. Ο χρήστης είναι ελεύθερος να διαλέξει την μορφοποίηση της αρεσκείας του και κατόπιν να πατήσει το κουμπί JButton(όπως νούμερο 9, Εικόνα 35). Με το πάτημα του κουμπιού μεταφέρονται όλες οι επιλογές που είναι στον πίνακα στην εφαρμογή και γίνεται η ανανέωση στην απεικόνιση του γράφου.

Το JTextField(όπως νούμερο 10, Εικόνα 35), κάθε φορά που θα πατηθεί ένα κελί στο γράφο μεταφέρει στον χρήστη το όνομα του επιλεγμένου αυτού κελιού.

Αυτά ήταν τα αντικείμενα αλληλεπίδρασης, που κάνει χρήση η διεπαφή μας. Παρακάτω θα δούμε πως έγινε η υλοποίηση της εφαρμογής καθώς και τα σενάρια χρήσης.

## 4 Υλοποίηση και σενάρια χρήσης

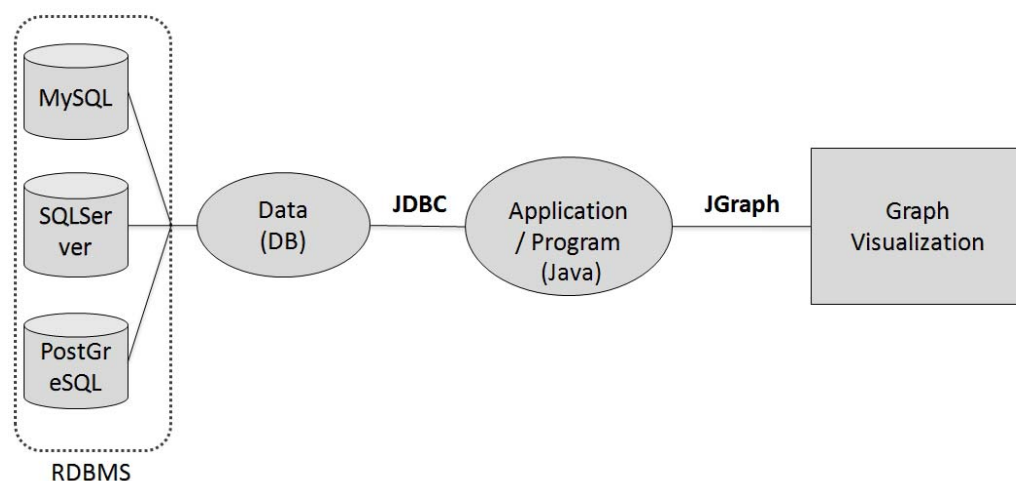
Στο κεφάλαιο αυτό θα δούμε πως υλοποιείται η εφαρμογή μας. Ακόμα θα δούμε τα σενάρια χρήσης της εφαρμογής.

Το σημαντικότερο κομμάτι υλοποίησης της πτυχιακής μας γίνεται μέσα από την εφαρμογή μας, συγκεκριμένα του κώδικα της που είναι γραμμένος σε γλώσσα προγραμματισμού Java. Για να μπορέσουμε να χειριστούμε την γλώσσα με ευκολία έγινε χρήση του εργαλείου NetBeansIDE 6.1. Η επιλογή του συγκεκριμένου έγινε λόγω ευκολίας χρήσης τόσο στον κώδικά όσο και στον χειρισμό των αντικειμένων αλληλεπίδρασης της Java(component palette).

Εκτός όμως από την Java, και για να μπορέσουμε να αναπαραστήσουμε οπτικά δεδομένα χρειαζόμαστε τα δεδομένα. Που πρέπει να είναι σε δομή μιας ΒΔ, άρα ή θα πρέπει να δημιουργήσουμε μια ΒΔ, ή να τα φορτώσουμε από μια είδη υπάρχον ΒΔ. Αυτές τις εργασίες τις χειριζόμαστε με την βοήθεια των RDBMS(βλέπε 2.1) όπως έχουμε αναφέρει και παραπάνω.

Επόμενο βήμα υλοποίησης είναι να μεταφέρουμε τα δεδομένα αυτά από το την ΒΔ ενός RDBMS στον χώρο της Java. Να γίνει δηλαδή η σύνδεση των δεδομένων μιας ΒΔ με την Java. Το κομμάτι αυτό υλοποιείται με την βοήθεια του JDBC, μια βιβλιοθήκη γραμμένη σε Java που με την χρήση των κατάλληλων κλάσεων της γίνεται η σύνδεση με την ΒΔ(παρόμοιο με ODBC).

Επόμενο βήμα υλοποίησης γίνεται με την χρήση του εργαλείου παρουσίασης JGraph. Με την βοήθεια του γίνεται η οπτική αναπαράσταση των δεδομένων που έχουμε φορτώσει από την ΒΔ σε μορφή γράφου.



Εικόνα 36 : Αλυσίδα των διαδικασιών της πτυχιακής μας.

Όπως φαίνεται(Εικόνα 36) ο πυρήνας υλοποίησης-λειτουργίας της πτυχιακής βασίζεται στην εφαρμογή(πρόγραμμα). Αυτό οφείλεται στο ότι τα παραπάνω

εργαλεία υλοποίησης που χρησιμοποιούμε τα χειριζόμαστε μέσα από την εφαρμογή(εκτός την δημιουργία της ΒΔ). Ο χειρισμός γίνεται με την χρήση των κατάλληλων κλάσεων, μεθόδων του εκάστοτε εργαλείου. Με άλλα λόγια όλα ελέγχονται από κατάλληλο κώδικα, είναι θέμα προγραμματισμού η υλοποίηση μέσα από τα εργαλεία αυτά.

Κάθε κομμάτι υλοποίησης είναι ένας κρίκος, το σύνολο όλων των κομματιών δίνει την παραπάνω αλυσίδα (Εικόνα 36). Αφού δημιουργηθεί η αλυσίδα και λειτουργούν όλα τα κομμάτια βγαίνουν τα παρακάτω σενάρια χρήσης της εφαρμογής μας.

#### **4.1 Σύνδεση με RDBMS μέσω JDBC**

Η εφαρμογή με την βοήθεια του JDBC ( Java Database Connector ) μπορεί να συνδεθεί με οποιοδήποτε RDBMS . Η σύνδεση με ένα RDBMS σημαίνει αυτόματα και σύνδεση με όλες τις υπάρχουσες ΒΔ του (βλέπε Εικόνα 36). Η εφαρμογή δίνει την δυνατότητα επιλογής της ΒΔ που θέλει να συνδεθεί ο χρήστης.

#### **4.2 Διάβασμα των δεδομένων των ΒΔ του RDBMS**

Αφού γίνει η σύνδεση με την ΒΔ ακολουθεί η μεταφορά των δεδομένων της στην εφαρμογή. Η εφαρμογή είναι σε θέση να πάρει όλες τις πληροφορίες που αφορούν τα δεδομένα της ΒΔ, καθώς και πληροφορίες που αφορούν την ίδια την ΒΔ.

#### **4.3 Μετατροπή των δεδομένων μιας ΒΔ σε αντικείμενα JAVA**

Η εφαρμογή τα δεδομένα που παίρνει από την ΒΔ μπορεί και τα μετατρέπει σε αντικείμενα της Java. Όταν λέμε δεδομένα εννοούμε τις εγγραφές μιας ΒΔ. Κάθε τέτοια έγγραφη διαθέτει όπως αναφέραμε και προηγούμενα διάφορα χαρακτηριστικά (πίνακα στον οποίο ανήκει, πεδίο, τύπος δεδομένου κάθε πεδίου κ.α.). Πολλά από τα χαρακτηριστικά και συγκεκριμένα αυτά που χρειαζόμαστε για τις ανάγκες της πτυχιακής τα μεταβιβάζουμε και στα δικά μας αντικείμενα που δημιουργούμε. Τα χαρακτηριστικά αυτά υπάρχουν και μπορούν να προσπελαστούν μέσω των μεταβλητών ή των μεθόδων που διαθέτει το κάθε αντικείμενο.

#### **4.4 Οπτική αναπαράσταση των αντικειμένων μέσω JGraph**

Η εφαρμογή καταφέρνει να δώσει οπτική αναπαράσταση στα αντικείμενα που έχει δημιουργήσει. Κάθε αντικείμενο που έχει δημιουργηθεί, μετατρέπεται με χρήση των κλάσεων και των μεθόδων του πακέτου του JGraph σε ένα δυσδιάστατο γραφικό αντικείμενο.

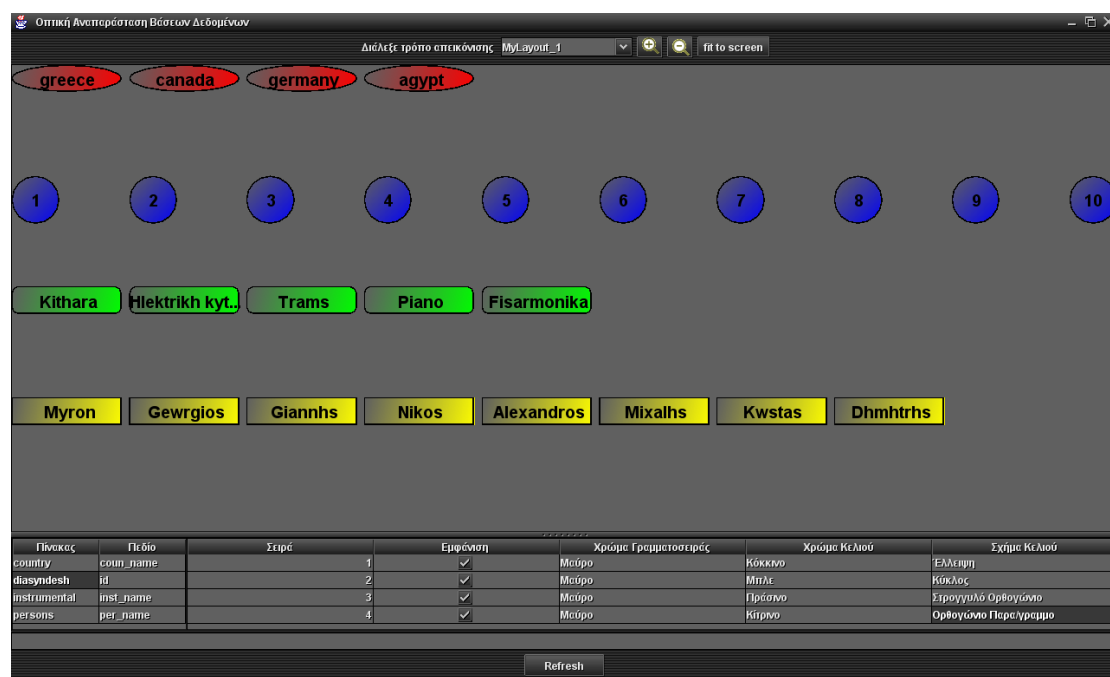
#### **4.5 Αλλαγή χωροδιάταξης των οπτικών αντικειμένων**

Η εφαρμογή διαθέτει την δυνατότητα να αλλάζει την χωροδιάταξη των οπτικών αντικειμένων. Η αλλαγές της θέσης απεικόνισης των οπτικών αντικειμένων στον χώρο διαχειρίζονται από ένα χειριστή διάταξης χώρου(layout manager). Ο οποίος διαθέτει διάφορους τρόπους απεικόνισης (layout). Παρακάτω θα αναφέρουμε τους



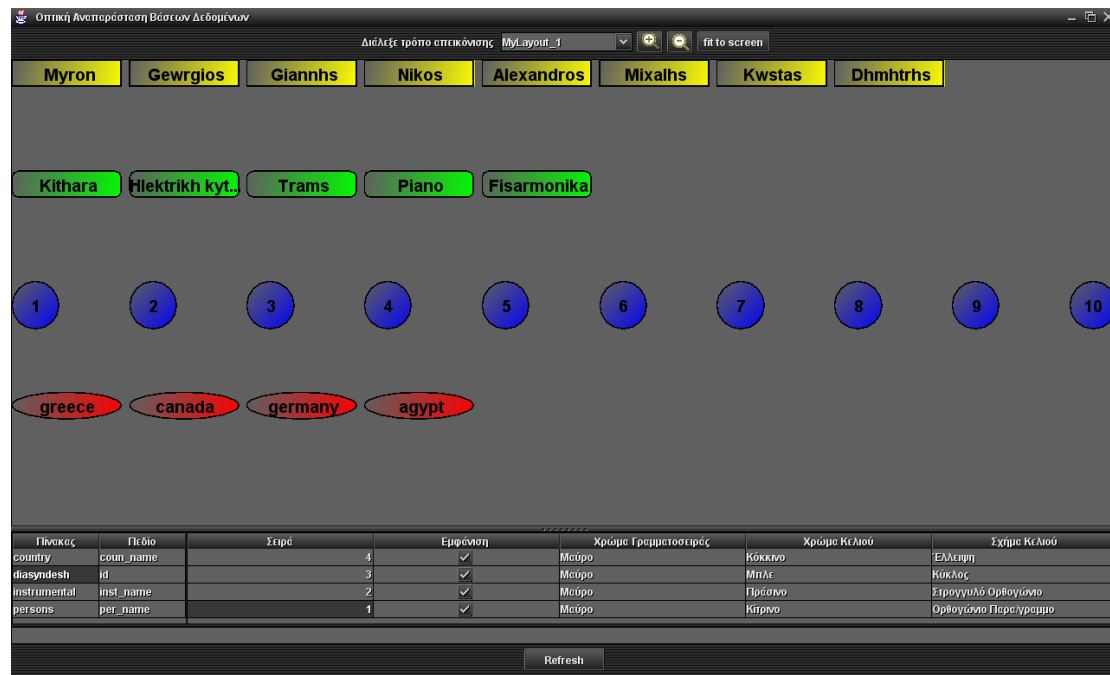
δύο δικούς μας τρόπους διάταξης των οπτικών αντικειμένων που δημιουργήσαμε για τις ανάγκες της πτυχιακής αυτής. Ως αναφορά τους έτοιμους τρόπους διάταξης που διαθέτει το JGraph, μπορούμε να πούμε ότι στην χρήση τους, που έγινε από την εφαρμογή αυτή, τα αποτελέσματα εξαρτώνται άμεσα από τα δεδομένα. Την δομή δηλαδή με την οποία έχουν προστεθεί στην ΒΔ, έτσι τα αποτελέσματα μπορεί να διαφέρουν από ΒΔ σε ΒΔ. Το γεγονός αυτό δεν εμφανίζεται στους δικούς μας τρόπους απεικόνισης άλλα και σε μερικούς από τους έτοιμους (Simple Circle Layout).

Ο πρώτος τρόπος απεικόνισης (MyLayout 1) τοποθετεί τις εγγραφές (κελιά) του πρώτου πίνακα (της ΒΔ που διαβάζει) τη μια δίπλα στην άλλη σε μια ευθεία και σε σταθερές αποστάσεις μεταξύ τους (διάταξη ροής). Μόλις τελειώσουν οι εγγραφές του πρώτου πίνακα, το σημείο τοποθέτησης μεταφέρεται ξανά στην αρχή αφού αφήσει ένα κενό και συνεχίζει την ίδια διαδικασία, αυτό γίνεται για όλους του πίνακες της ΒΔ (Εικόνα 37). Έτσι, έχουμε μια απεικόνιση, που εύκολα βλέπουμε το πλήθος των πινάκων (ανάλογος των γραμμών) και των εγγραφών (ανάλογος των κελιών κάθε γραμμής).



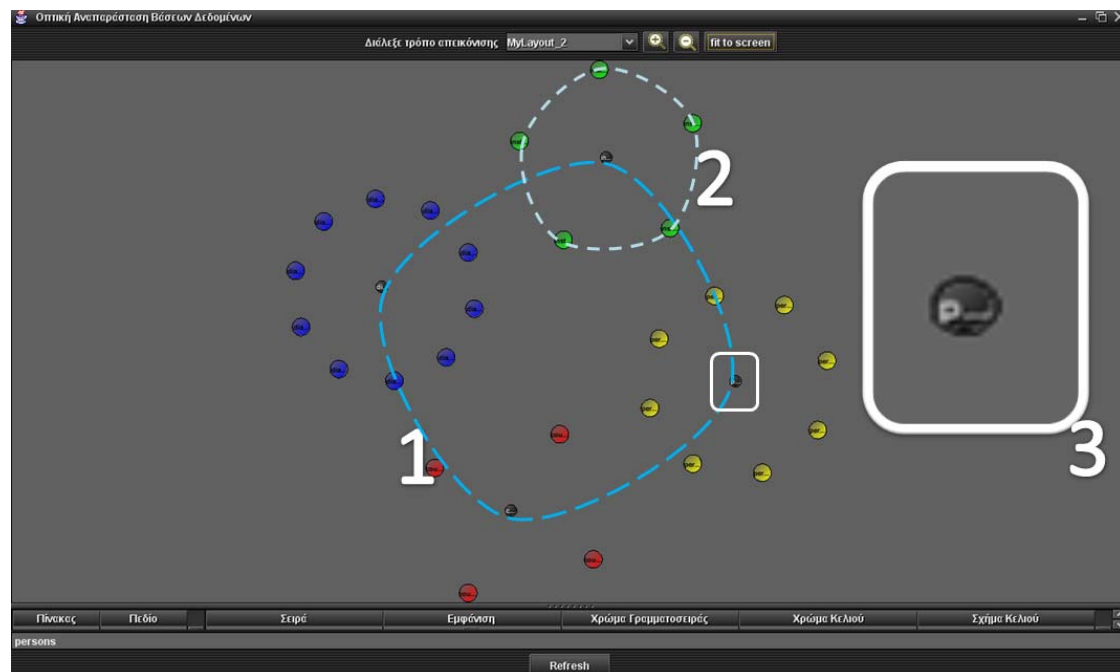
Εικόνα 37 : My Layout 1, αρχική απεικόνιση

Στη συνέχεια έχουμε τη δυνατότητα να αλλάζουμε τη σειρά, με την οποία εμφανίζονται οι πίνακες. Αυτό γίνεται από τη στήλη «Σειρά » , δίνοντας τον αριθμό ένα (1) στον πίνακα που θέλουμε να εμφανίζεται πρώτος, δύο (2) στο δεύτερο κ.τ.λ. Στη παρακάτω Εικόνα 38, βλέπουμε τους πίνακες να εμφανίζονται με την αντίστροφη σειρά απ’ ότι στην Εικόνα 37.



Εικόνα 38 : Αλλαγή σειράς πινάκων

Η χρηστικότητα του δεύτερου τρόπου απεικόνισης (My Layout 2), δίνει μια συνολική εικόνα των δεδομένων της βάσης. Έτσι μπορούμε εύκολα να δούμε το πλήθος, των δεδομένων της βάσης και των εγγραφών κάθε πίνακα. Η χωροδιάταξη εδώ βασίζεται στην κυκλική αποτύπωση των κελιών, συγκεκριμένα σε δύο νοητούς κύκλους. Ο πρώτος κύκλος (νούμερο 1) σχηματίζεται από το πλήθος των δεύτερων κύκλων (νούμερο 2).



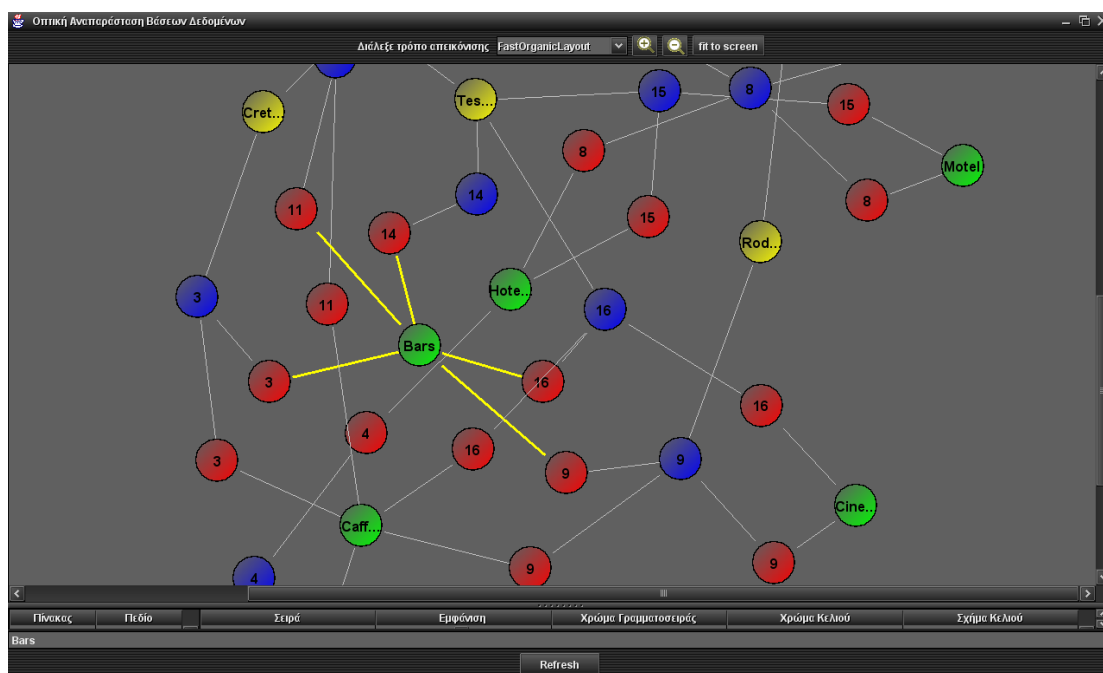
Εικόνα 39: My Layout 2

Έτσι όπως φαίνεται στην Εικόνα 39, πάνω στα όρια του πρώτου (νούμερο 1) τοποθετείται το κέντρο (νούμερο 3) των άλλων κύκλων (νούμερο 2). Τα σημεία του

πρώτου κύκλου είναι οι πίνακες και τα σημεία του δεύτερου κύκλου είναι οι εγγραφές των πινάκων. Μέσω του χαρακτηριστικού αυτού της εφαρμογής, η πληροφορία που παίρνει ο χρήστης από τα οπτικά αντικείμενα αλλάζει από απεικόνιση (layout) σε απεικόνιση. Το ότι ο χρήστης μπορεί να πάρει από τα ίδια οπτικά αντικείμενα, με την βοήθεια των διαφορετικών διατάξεων στο χώρο, διαφορετικές πληροφορίες από την ΒΔ, είναι πολύ σημαντικό στην κατανόηση της ΒΔ, κάτι που είναι και ο βασικός σκοπός της εφαρμογής.

#### 4.6 Δυνατότητα άμεσης εύρεσης σχεσιακών αντικειμένων, για κάθε αντικείμενο

Η εφαρμογή, με την βοήθεια της διεπαφής της, δίνει στον χρήστη την δυνατότητα να βρει για κάθε οπτικό αντικείμενο την σχέση του με άλλα οπτικά αντικείμενα (βλέπε Εικόνα 40). Αυτό γίνεται επιλέγοντας, με το πάτημα του ποντικιού (κλικ στο αριστερό κουμπί) το γραφικό αντικείμενο, άμεσα φωτίζονται (με κίτρινο χρώμα) οι συνδέσεις των σχεσιακών οπτικών αντικειμένων με αυτό. Επίσης σημαντική ιδιότητα στην κατανόηση των δεδομένων και των σχέσεων τους, που υπάρχουν σε μια ΒΔ.



Εικόνα 40 : Με κίτρινες γραμμές φαίνονται οι συνδέσεις του συγκεκριμένου κελιού

#### 4.7 Μορφοποίηση οπτικών αντικειμένων

Η εφαρμογή παρέχει στον χρήστη την δυνατότητα να τροποποιήσει την μορφή των γραφικών αντικειμένων. Κάθε αντικείμενο είναι σε θέση να αλλάξει σχήμα και χρώμα, επίσης μπορεί να αλλάξει το κείμενο περιγραφής του καθώς και το χρώμα του. Με τις αλλαγές αυτές ο χρήστης μπορεί να ξεχωρίσει τα δεδομένα του κάθε πίνακα, εξασφαλίζοντας ευκολότερη κατανόηση της ΒΔ.

#### **4.8 Εμφάνιση / Απόκρυψη πίνακα από την αναπαράσταση**

Η εφαρμογή διαθέτει την δυνατότητα να εμφανίζει οπτικά τις εγγραφές των πινάκων. Πολλές φορές όμως δεν θέλουμε να εμφανίζονται τα δεδομένα μερικών πινάκων. Τέτοιοι πίνακες μπορεί να αποδίδουν συσχετίσεις μεταξύ άλλων πινάκων της ΒΔ, δηλαδή δεν περιέχουν δεδομένα αλλά τις σχέσεις των δεδομένων άλλων πινάκων, ή σε μεγάλες ΒΔ με πολλούς πίνακες να μην θέλουμε να εμφανίζονται όλοι. Σε κάθε περίπτωση η επιλογή αφήνεται στον χρήστη, οποία στιγμή θέλει ξανά εμφανίζει τα δεδομένα που είχε επιλέξει να μην φαίνονται.

## 5 Σύνοψη και συμπεράσματα

Σκοπός της πτυχιακής ήταν να πειραματιστούμε με διάφορους τύπους βάσεων δεδομένων. Απ' τις οποίες μέσω σύνδεσης με τη java να πάρουμε τα δεδομένα τους και να δημιουργήσουμε τα αντίστοιχα αντικείμενα. Στη συνέχεια, να δημιουργήσουμε διάφορες οπτικές αναπαραστάσεις των αντικειμένων αυτών με τη βοήθεια του JGraph .

Αρχίζοντας με τις βάσεις, τα πλεονεκτήματα και τα μειονεκτήματα της MySQL έναντι του SqlServer 2000. Το μεγάλο της πλεονέκτημα είναι ότι είναι ανοιχτού κώδικα και διατίθεται δωρεάν. Επίσης είναι καλύτερη στη διαχείριση πόρων του συστήματος και πιο γρήγορη στην εκτέλεση ερωτημάτων . Στα μειονεκτήματα τις είναι η δημιουργία σχέσεων μεταξύ πινάκων , στην οποία υπερτερεί ο SqlServer 2000, όπως επίσης και στις δικλίδες ασφαλείας που παρέχει.

Στη συνέχεια, χρησιμοποιώντας τη Java κάναμε χρήση αρκετών χαρακτηριστικών της, που την έχουν καταστήσει τη κορυφαία αντικειμενοστραφής γλώσσα. Αρχικά , η εύκολη σύνδεση με βάσεις δεδομένων και άντληση στοιχείων απ' αυτές. Ακόμα η δημιουργία αντικειμένων και οι δυνατότητες χρήσεις τους. Επίσης , μέσω NetBeans , η απλουστευμένη διαδικασία υλοποίησης φορμών και η δυνατότητα παραμετροποίησης των. Τέλος , με τη βοήθεια της βιβλιοθήκης JGraph , η αναπαράσταση δεδομένων με διάφορους γράφους , που δίνουν στο χρήστη τη δυνατότητα να παρακολουθήσει ποσοτικά και ποιοτικά στοιχεία της βάσης.

## Βιβλιογραφία

### Βιβλία

1. Εισαγωγή στη Java 2, Γιώργος Λιακέας, Εκδόσεις Κλειδάριθμος.
2. Διεπαφή Χρήστη-Υπολογιστή, μια σύγχρονη προσέγγιση 2006, Δημοσθένης Ακουμιανάκης, Εκδόσεις Κλειδάριθμος.

### Διαδίκτυο

1. <http://www.smashingmagazine.com/2007/08/02/data-visualization-modern-approaches/>
2. <http://www.vrvis.at/via/resources/DA-NSahling/node3.html>
3. <http://www.jgraph.com/pub/jgraphmanual.pdf>
4. <http://www.jgraph.com/forum/>
5. <http://prefuse.org/>
6. <http://www.jtattoo.net/ScreenShots.html>