

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ  
Σχολή Τεχνολογικών Εφαρμογών  
Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων



*Πτυχιακή εργασία*

Σχεδίαση και ανάπτυξη χωρικών βάσεων δεδομένων και μεθόδων επερωτήσεων

Ψαράκης Δημήτρης  
Τσάκωνας Φώτης

Επιβλέπων καθηγητής : Ακουμιανάκης Δημοσθένης

Ηράκλειο, Οκτώβρης 2006

## **Ευχαριστίες**

Θα θέλαμε να ευχαριστήσουμε τον Καθηγητή κ. Ακουμιανακη Δημοσθένη για την ανάθεση της εργασίας, την στήριξη και συνεργασία που συνέβαλλαν στην ολοκλήρωση της.

Επίσης θα θέλαμε να ευχαριστήσουμε θερμά τους συναδέλφους Μαυραγάνη Κων/νο , Αεράκη Μιχαήλ , Γκαγκα Δημήτριο , Λουδάρο Νικόλαο και Ρομαντζη Ελένη για την στήριξη και βοήθεια που μας παρείχαν.

Τέλος θα θέλαμε να ευχαριστήσουμε τις οικογένειες μας για την ηθική και υλική υποστήριξη που μας παρείχαν κατά την διάρκεια των σπουδών μας .

# Πίνακας περιεχομένων

<b>1</b>	<b>ΕΙΣΑΓΩΓΗ</b>	<b>6</b>
<b>2</b>	<b>ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ &amp; ΜΗΧΑΝΙΣΜΟΙ ΕΠΕΡΩΤΗΣΕΩΝ</b>	<b>8</b>
2.1	Γενικά περί βάσεων δεδομένων μια μικρή εισαγωγή	8
2.2	Κλασικοί τρόποι υποβολής ερωτημάτων	9
2.3	Γραφικοί τρόποι υποβολής ερωτημάτων	10
2.3.1	Συμβολικές γλώσσες	10
2.3.2	Χάρτες	11
2.4	Τρόποι αναπαράστασης αποτελέσματος	16
2.4.1	Ο κλασικός πίνακας	16
2.4.2	Γραφικοί τρόποι (οπτικές αναπαραστάσεις)	17
2.4.3	Μηχανισμοί θέασης γραφικής αναπαράστασης	28
<b>3</b>	<b>ΤΟ ΣΥΣΤΗΜΑ E-TOUR</b>	<b>30</b>
3.1	Αρχιτεκτονική συστήματος	30
3.2	Σχεδίαση βάσης δεδομένων	30
3.3	Σχεδίαση Διεπαφής	35
3.4	Υλοποίηση συστήματος	39
3.4.1	MySQL	39
3.4.2	NetBeans	40
3.4.3	SQLyog	41
<b>4</b>	<b>ΣΕΝΑΡΙΑ ΛΕΙΤΟΥΡΓΙΑΣ</b>	<b>42</b>
4.1	Απλή Αναζήτηση δραστηριότητας μέσω Ημερομηνίας	42
4.2	Πολλαπλή αναζήτηση δραστηριότητας μέσω Ημερομηνίας	47
4.3	Αναζήτηση δραστηριότητας μέσω τοποθεσίας	50
4.4	Φόρμες παρουσίασης των αποτελεσμάτων	52
<b>5</b>	<b>ΣΥΜΠΕΡΑΣΜΑΤΑ</b>	<b>58</b>
<b>6</b>	<b>ΒΙΒΛΙΟΓΡΑΦΙΑ</b>	<b>60</b>

## Ευρετήριο εικόνων

Εικόνα 1: Παράδειγμα συμβολικής επερώτησης.....	11
Εικόνα 2: Οι βασικές αρχές του art museum (a) το αρχικό σκίτσο (b) Το αποτέλεσμα της εξαγωγής γραμμών με το οποίο συγκρίνεται (c)Τα μεταδεδομένα τα οποία τραβήχτηκαν απ’ τον Αρχικό πίνακα (d).....	13
Εικόνα 3: Παράδειγμα κανονικοποίησης ενός ανακτημένου πολύγωνα mehrotra and gary .....	14
Εικόνα 4: Αποσύνθεση της εικόνας σε τοπικές περιοχές χρώματος(a)Η αρχική εικόνα (b) η αποσυντεθειμένη εικόνα (c) Η αποκοιμισμένη εικόνα .....	14
Εικόνα 5: Η διεπαφή χρήστη του cigales.....	15
Εικόνα 6: Ένα σετ εντολών σε μορφή χειρονομιών που προσφέρει το electronic cocktail parkin .....	15
Εικόνα 7: Ανιχνευμένα σχήματα που μοιάζουν με μπουκάλια τα οποία ανακτήθηκαν από μία βιβλιοθήκη πινάκων βασισμένα σ’ ένα ελεύθερο σχέδιο ενός μπουκαλιού(δε φαίνεται στο σχήμα) .....	16
Εικόνα 8: Παράδειγμα πίνακα που περιέχει πληροφορίες για βάση δεδομένων.....	17
Εικόνα 9: Οι τρεις βασικές δραστηριότητες του χρήστη που αφορούν την αντίληψη (a) ανάγνωση (b) σάρωση (c) ακρόαση .....	17
Εικόνα 10: Η χρήση της φόρμας για την εισαγωγή δεδομένων και τον προσδιορισμό των κριτηρίων μιας ερώτησης .....	19
Εικόνα 11: Πληροφορίες για τους πελάτες ενός ξενοδοχείων κατά τη διάρκεια ενός έτους.....	19
Εικόνα 12 Παράδειγμα μιας Form – Based αναπαράστασης στο EMBS .....	20
Εικόνα 13: Φόρμα για αντιπροσώπευση διαγραμμάτων - παραγωγή κρέατος σε μερικές ευρωπαϊκές χώρες .....	21
Εικόνα 14: Χρήση διαγράμματος Ο-Σ για το σχηματισμό σχεσιακού σχήματος.....	21
Εικόνα 15: Παραδείγματα των εικόνων .....	22
Εικόνα 16: Υβριδική προσέγγιση στο SKI .....	24
Εικόνα 17: Παράδειγμα SICON .....	24
Εικόνα 18: Παράδειγμα εφαρμογής της τεχνικής ‘dynamic querying’ .....	25
Εικόνα 19: Αναπαράσταση αποτελεσμάτων του HomeFinder υπό μορφή κειμένου..	26
Εικόνα 20: Dynamaps .....	27
Εικόνα 21: Filmfinder .....	27
Εικόνα 22: Οι τρεις βασικές κατηγορίες αλλαγής τρόπου θέασης (a) αλλαγή προοπτικής (b) αλλαγή θεματικού επιπέδου (c) αλλαγή επιπέδου λεπτομέρειας	28
Εικόνα 23: Αρχιτεκτονική δομή συστήματος.....	30
Εικόνα 24: Μοντέλο Ο-Σ.....	31
Εικόνα 25: Σχεσιακό μοντέλο Βάσης Δεδομένων .....	32
Εικόνα 26: Μοντέλο της βάσης .....	32
Εικόνα 27: Στιγμιότυπο του πίνακα ‘Δραστηριότητα’ .....	33
Εικόνα 28: Στιγμιότυπο του πίνακα ‘κατηγορία’ .....	33
Εικόνα 29: Στιγμιότυπο του πίνακα ‘τοποθεσία’ .....	34
Εικόνα 30: Στιγμιότυπο του πίνακα ‘Template’ .....	34
Εικόνα 31: Στιγμιότυπο του πίνακα ‘Type’ .....	35
Εικόνα 32: Σκαρίφημα διεπαφής .....	36
Εικόνα 33: Πρωτότυπο υψηλής πιστότητας .....	37

Εικόνα 34: Φόρμα παρουσίασης Συναυλιών .....	38
Εικόνα 35: Φόρμα παρουσίασης κινηματογραφικής ταινίας .....	38
Εικόνα 36: Φόρμα Παρουσίασης Θεατρικής Παράστασης.....	39
Εικόνα 37: SQLyog main data tab .....	41
Εικόνα 38: Ο χάρτης του Ηρακλείου .....	43
Εικόνα 39: Ενημερωμένα πεδία της διεπαφής.....	45
Εικόνα 40: Ενημέρωση χάρτη .....	46
Εικόνα 41: Ενημέρωση χάρτη διεπαφής.....	47
Εικόνα 42: Αναζήτηση μέσω ημερομηνία.....	48
Εικόνα 43: Ένδειξη σημείων με δραστηριότητες σε συγκεκριμένες ημερομηνίες .....	49
Εικόνα 44: Επιλογή από λίστα δραστηριοτήτων συγκεκριμένης ημερομηνίας .....	50
Εικόνα 45: Αναζήτηση δρώμενων ανά τοποθεσία .....	51
Εικόνα 46: Παρουσίαση όλων των δραστηριοτήτων .....	51
Εικόνα 47: Επιλογή κατηγορίας δραστηριότητας .....	52
Εικόνα 48: Παράδειγμα παρουσίασης θεατρικής δραστηριότητας.....	53
Εικόνα 49: Παράδειγμα παρουσίασης συναυλίας .....	54

# 1 Εισαγωγή

---

Στην εποχή μας οι χωρικές βάσεις δεδομένων, δηλαδή οι βάσεις που περιέχουν χωρικά, γεωγραφικά δεδομένα (συντεταγμένες, γεοπληροφοριακά στοιχεία κ.α.), είναι ευρέως διαδεδομένες και χρησιμοποιούνται σε πολλούς τομείς της πληροφορικής και της τεχνολογίας γενικότερα.

Παρόλα αυτά η χρησιμότητά τους αυτή μπορεί να επισκιαστεί αν ακολουθήσουμε τις παραδοσιακές, κουραστικές μεθόδους επερωτήσεων. Οι σημερινοί μέθοδοι για την αλληλεπίδραση με γεωγραφικά συστήματα πληροφοριών (GIS) και με χωρικές βάσεις δεδομένων είναι κυρίως μη χωρικές, καθώς απαιτούν απ' τους χρήστες να ασχοληθούν με γεωγραφικά δεδομένα κυρίως μεσ' από τη χρήση γλωσσών αλφαμετρικών εντολών. Η διενέργεια χωρικών επερωτήσεων πληκτρολογώντας μια εντολή σε μια χωρική γλώσσα εντολών είναι μια κουραστική διαδικασία, κυρίως γιατί απαιτεί εκτεταμένη εξάσκηση στη γλώσσα αυτή, έτσι ώστε να χρησιμοποιηθεί σωστά το λεξιλόγιό της, και επίσης γιατί αναγκάζει το χρήστη να μετατρέπει χωρικές εικόνες, που πιθανώς έχει στο μυαλό του, σε μη χωρικές γλώσσες. Επίσης οι λεκτικές περιγραφές γεωγραφικών-χωρικών καταστάσεων είναι συνήθως διφορούμενες και μπορούν πολύ συχνά να οδηγήσουν σε παρερμηνεύσεις. Αυτά τα μειονεκτήματα κάνουν τις κλασικές μεθόδους επερωτήσεων επιρρεπής στα λάθη και δύσκολες στη χρήση τους.

Για να αποφύγουμε αυτό το εννοιολογικό κενό, εμείς σ' αυτή την εργασία προβάλλουμε ένα τρόπο υποβολής επερωτήσεων σε χωρική βάση δεδομένων μέσω αλληλεπίδρασης με οπτικά-συμβολικά αντικείμενα τοποθετημένα πάνω σε δοσμένο χάρτη αλλά και με άλλους οπτικούς τρόπους υποβολής επερωτήσεων όπως με τη χρήση combo boxes, μπάρων κύλισης κ.α. Αυτός ο τρόπος επερωτήσεων υποστηρίζει την πιο άμεση ανθρώπινη χωρική σκέψη, που είναι σημαντικό, καθώς οι χρήστες συχνά έχουν οπτικές αναπαραστάσεις στο μυαλό τους όταν κάνουν ερωτήσεις σχετικές με χωρικά δεδομένα.

Σε αυτήν την εργασία εξετάζουμε τη δημιουργία της βάσης για ένα ολοκληρωμένο πολιτιστικό-τουριστικό οδηγό με τη χρήση χάρτη. Η εφαρμογή, ανάλογα με την επερώτηση που θα έχει γίνει προς τη βάση μας, μας θα υποδεικνύει σημεία πάνω σ' αυτόν.

Τα σημεία αυτά θα αντιπροσωπεύουν τον τόπο στον οποίο λαμβάνει χώρα μια δραστηριότητα αλλά και το είδος της δραστηριότητας αυτής. Στη συνέχεια μέσα από την αλληλεπίδραση με τα σημεία αυτά αλλά και με άλλα γραφικά μέσα της διεπαφής μας θα εμφανίζονται στο χρήστη πληροφορίες σχετικά με τις δραστηριότητες που τον ενδιαφέρουν.

Με αυτόν τον τρόπο προσπαθούμε να επιτύχουμε την εύκολη και γρήγορη αλληλεπίδραση χρήστη - συστήματος. Απ' τη μία μέσω των γραφικών μεθόδων επερωτήσεων απαλλάσσουμε το χρήστη απ' την ανάγκη γνώσης μιας γλώσσας

επερωτήσεων και αποκρύπτουμε την πολυπλοκότητα που υπάρχει στο υπόστρωμα και απ' την άλλη μέσω των γραφικών μεθόδων αναπαραστάσεως της πληροφορίας τα δεδομένα τα οποία ζητήθηκαν γίνονται εμφανή με τρόπο ευπαρουσίαστο και κατανοητό από κάθε άνθρωπο που χρησιμοποιεί το πρόγραμμα αυτό.

Στα επόμενα κεφάλαια της εργασίας αυτής θα δούμε μια περιγραφή των γραφικών / χωρικών μεθόδων επερωτήσεων αλλά και των κλασικών μεθόδων και των βάσεων δεδομένων (κεφάλαιο 2). Στη συνέχεια στο κεφάλαιο 3 γίνεται μια περιγραφή της αρχιτεκτονικής του συστήματος και της διαμόρφωσης της βάσης δεδομένων. Στο κεφάλαιο 4 παρουσιάζονται συνοπτικά επιλεγμένα σενάρια χρήσης του συστήματος που αναπτύχθηκε. Τέλος, το κεφάλαιο 5 συνοψίζει συμπεράσματα και δυνατότητες περαιτέρω ανάπτυξης του συστήματος.

## **2 Βάσεις δεδομένων & μηχανισμοί επερωτήσεων**

---

Στο κεφάλαιο αυτό επιχειρείται μια συνοπτική επισκόπηση των γνωστικών αντικειμένων που αποτελούν τη θεωρητική βάση της παρούσας πτυχιακής εργασίας. Συγκεκριμένα θα παρουσιάσουμε τις βασικές / παραδοσιακές αρχές βάσεων δεδομένων αλλά και σύγχρονες τάσεις στη ανάπτυξη συστημάτων υποβολής ερωτημάτων σε περιβάλλοντα και εφαρμογές σχεσιακών βάσεων δεδομένων.

### **2.1 Γενικά περί βάσεων δεδομένων μια μικρή εισαγωγή**

Ένας πιθανός ορισμός για μια βάση δεδομένων είναι ότι αποτελεί μια συλλογή από καταγραφές που αποθηκεύονται σε έναν υπολογιστή με συστηματικό τρόπο, έτσι ώστε μια εφαρμογή να μπορεί να την συμβουλευτεί για να απαντήσει σε ερωτήματα. Οι καταγραφές που ανακτώνται απαντώντας σε ερωτήματα γίνονται πληροφορίες που μπορούν να χρησιμοποιηθούν για τη λήψη αποφάσεων. Το πρόγραμμα του υπολογιστή το οποίο διαχειρίζεται και διαμορφώνει ερωτήματα σε μια βάση δεδομένων ονομάζεται Σύστημα Διαχείρισης Βάσεων Δεδομένων ή DBMS (database management system).

Η κεντρική ιδέα πίσω από μια βάση δεδομένων είναι ότι αποτελεί μια συλλογή από κομμάτια γνώσης. Τυπικά υπάρχει μια δομημένη περιγραφή του τύπου των καταγραφών και γεγονότων που υπάρχουν σε μια βάση. Αυτή η περιγραφή είναι γνωστή σαν σχήμα (schema) της βάσης. Το σχήμα αυτό περιγράφει τα αντικείμενα τα οποία αναπαριστώνται σε μια βάση δεδομένων, καθώς και τις μεταξύ τους σχέσεις. Ο τρόπος με τον οποίο είναι οργανωμένο το σχήμα αποτελεί και το μοντέλο της βάσης δεδομένων.

Το πιο ευρέως χρησιμοποιούμενο μοντέλο σήμερα είναι το σχεσιακό μοντέλο. Το μοντέλο αυτό αναπαριστά όλες τις πληροφορίες με την μορφή πολλαπλών σχετιζόμενων πινάκων, όπου κάθε πίνακας αποτελείται από γραμμές και στήλες. Επίσης, αναπαριστά τις σχέσεις μεταξύ αντικείμενων μετά την χρήση κοινών τιμών σε παραπάνω από έναν πίνακα.

Ένα άλλο δημοφιλές μοντέλο το οποίο έχει αναπτυχθεί τα τελευταία χρόνια είναι το αντικειμενοστραφές μοντέλο το οποίο προσπάθησε και έφερε κοντά τον κόσμο των βάσεων δεδομένων καθώς και των προγραμματιστικών εφαρμογών. Αυτό το κατάφερε κυρίως γιατί διασφάλισε ότι μια βάση δεδομένων χρησιμοποιεί το ίδιο σύστημα τύπων με το πρόγραμμα εφαρμογών με σκοπό την αποφυγή μετατροπών της αναπαράστασης της πληροφορίας από μια Β.Δ. στο περιβάλλον μιας εφαρμογής. Την ίδια στιγμή το αντικειμενοστραφές μοντέλο εισάγει τις βασικές ιδέες του αντικειμενοστραφούς προγραμματισμού όπως την ενθυλάκωση και τον πολυμορφισμό στον κόσμο των βάσεων δεδομένων. Επιπροσθέτως, είναι σημαντικό να αναφερθεί ότι τον τελευταίο καιρό έχει αναπτυχθεί το, ιδιαίτερος χρήσιμο,



μοντέλο των αντικειμενο-σχεσιακών βάσεων δεδομένων το οποίο αποτελεί συνδυασμό του κλασσικού σχεσιακού και του αμιγώς αντικειμεστρεφούς μοντέλου.

## 2.2 Κλασικοί τρόποι υποβολής ερωτημάτων

Σημαντικό είναι σ' ένα μοντέλο αλληλεπίδρασης χρήστη με το σύστημα να υπάρχει η δυνατότητα παροχής της μέγιστης δυνατής διευκόλυνσης του χρήστη για την αποκόμιση των κρίσιμων πληροφοριών και την επίτευξη των στόχων τους με ευκολία, ταχύτητα και αποτελεσματικότητα. Ένα σύστημα επερωτήσεων, θα πρέπει να διαθέτει χαρακτηριστικά τα οποία βοηθούν στη σύνταξη των κατάλληλων ερωτήσεων για την αποκόμιση των επιθυμητών απαντήσεων πράγμα που προϋποθέτει την σωστή κατανόηση της βάσης από τον χρήστη. Ένα επίσης σημαντικό χαρακτηριστικό είναι η απόδοση των αποτελεσμάτων με τέτοιο τρόπο ώστε να γίνονται απολύτως κατανοητά, και έτσι να μπορούν να χρησιμοποιηθούν σε μετέπειτα εφαρμογές.

Για να υποβληθούν και να διατυπωθούν τα ερωτήματα αυτά, χρησιμοποιούνται κάποιες δομημένες γλώσσες. Οι γλώσσες αυτές δημιουργούν, αλλάζουν, καταχωρούν και χειρίζονται δεδομένα από ένα RDBMS. Η σημαντικότερη δομημένη γλώσσα διαχείρισης δεδομένων σε περιβάλλοντα σχεσιακών συστημάτων είναι η SQL (structure query language). Πρέπει να σημειωθεί ότι, οι γλώσσες της κατηγορίας αυτής δεν αποτελούν ολοκληρωμένες γλώσσες προγραμματισμού, αν και διαθέτουν πολλά από τα χαρακτηριστικά τους όπως γραμματική, συντακτικό άλλα και χρησιμοποιούνται για προγραμματιστικούς σκοπούς.

Γενικότερα, η λειτουργική ικανότητα των γλωσσών χειρισμού δεδομένων και μεθόδων επερωτήσεων βασίζεται στις αρχικές λέξεις μιας δήλωσης, που είναι σχεδόν πάντα ένα ρήμα. Για παράδειγμα στην περίπτωση της SQL υπάρχουν ενδεικτικά τα ρήματα: SELECT , INSERT , UPDATE και DELETE. Με την εντολή SELECT ανακτούμε τα δεδομένα από μια ή περισσότερες γραμμές ενός η περισσότερων πινάκων : π.χ. :

```
SELECT * From Books where prices > 100.00.
```

Σε αυτή την περίπτωση θα εμφανιστούν όλες οι καταχωρήσεις του πίνακα Books όπου το πεδίο price έχει τιμή πάνω από 100.00. Με την INSERT προσθέτουμε παραπάνω γραμμές σε έναν υπάρχον πίνακα : π.χ.

```
INSERT into Books (Field1, Field2, Field3) values  
( 'test' , 'N' , Null)
```

Την εντολή UPDATE την χρησιμοποιούμε όταν θέλουμε να αλλάξουμε τιμές στις ήδη υπάρχουσες γραμμές ενός πίνακα: π.χ.

```
UPDATE Books SET field1 = 'x' where Field2 = 'N' .
```

Τέλος η DELETE αφαιρεί γραμμές από τον πίνακα π.χ.

```
DELETE From Books where field2 = 'N'
```

Με τα παραπάνω παραδείγματα διαπιστώνουμε ότι η φύση των γλωσσών αυτών γενικότερα αποτελείται από ένα σύνολο δηλώσεων και εντολών προς την βάση δεδομένων, οι οποίες ελέγχονται ως προς το συντακτικό και το λεκτικό τους και οι ορθά διατυπωμένες δηλώσεις εκτελούνται προκειμένου να παραχθεί το επιθυμητό αποτέλεσμα.

Παρ' όλα αυτά και παρά την φαινομενική απλότητα των εντολών αυτών είναι σκόπιμο να ειπωθεί ότι οι περισσότεροι άνθρωποι που αλληλεπιδρούν με υπολογιστές δεν έρχονται σε επαφή με τέτοιου είδους λεπτομέρειες. Αυτό που βλέπουν, και αυτό που ουσιαστικά τους ενδιαφέρει είναι μια απλή, γρήγορη και λειτουργική διεπαφή ώστε να ολοκληρώσουν με συνεπεία τις εκάστοτε εργασίες τους. Για αυτόν τον λόγο αυτές οι κλασικές μέθοδοι επερωτήσεων που εξετάσαμε παραπάνω θεωρούνται πλέον ξεπερασμένες και υπάρχουν μονό σαν ένα υπόστρωμα διεργασιών που αποκρύπτονται από τον εκάστοτε χρήστη, συνήθως μέσω μιας γραφικής διεπαφής .

## **2.3 Γραφικοί τρόποι υποβολής ερωτημάτων**

Οι γραφικοί τρόποι υποβολής ερωτημάτων (Visual Querying Systems) αποτελούν εναλλακτικά συστήματα επερωτήσεων για βάσεις δεδομένων και χρησιμοποιούν οπτικές αναπαραστάσεις για να τονίσουν το πεδίο του ενδιαφέροντος άλλα και για να ικανοποιήσουν τις απαιτήσεις του χρήστη. Τα συστήματα αυτά σχεδιάζονται για να βελτιώσουν την επικοινωνία ανθρώπου – υπολογιστή.

Τα πιο σημαντικά τους χαρακτηριστικά είναι εκείνα που καθορίζουν την φύση του διάλογου μεταξύ ανθρώπου και υπολογιστή. Τέτοια συστήματα εκμεταλλεύονται το υψηλό εύρος ζώνης του καναλιού της ανθρωπίνης όρασης, επιτρέποντας έτσι, την αναγνώριση και τον χειρισμό μεγάλων ποσοτήτων πληροφοριών. Τα VQS παρέχουν και την γλώσσα για να εκφραστούν επερωτήσεις στη βάση δεδομένων με οπτικό τρόπο άλλα και μια ποικιλία από δυνατότητες για την διευκόλυνση της αναπαράστασης αποτελεσμάτων συνεισφέροντας με τον τρόπο αυτό στη βελτίωση της αλληλεπίδρασης χρήστη – συστήματος. Έτσι, προορίζονται για μια ευρεία γκάμα χρηστών, κυρίως σε αρχάριους, οι οποίοι έχουν περιορισμένη γνώση υπολογιστών και αδιαφορούν για την εσωτερική δομή των βάσεων δεδομένων. Τα τελευταία χρόνια τα VQS έχουν υιοθετήσει μια μεγάλη γκάμα από στρατηγικές οπτικής αναπαράστασης δεδομένων άλλα και αλληλεπιδράσεις με τον χρήστη.

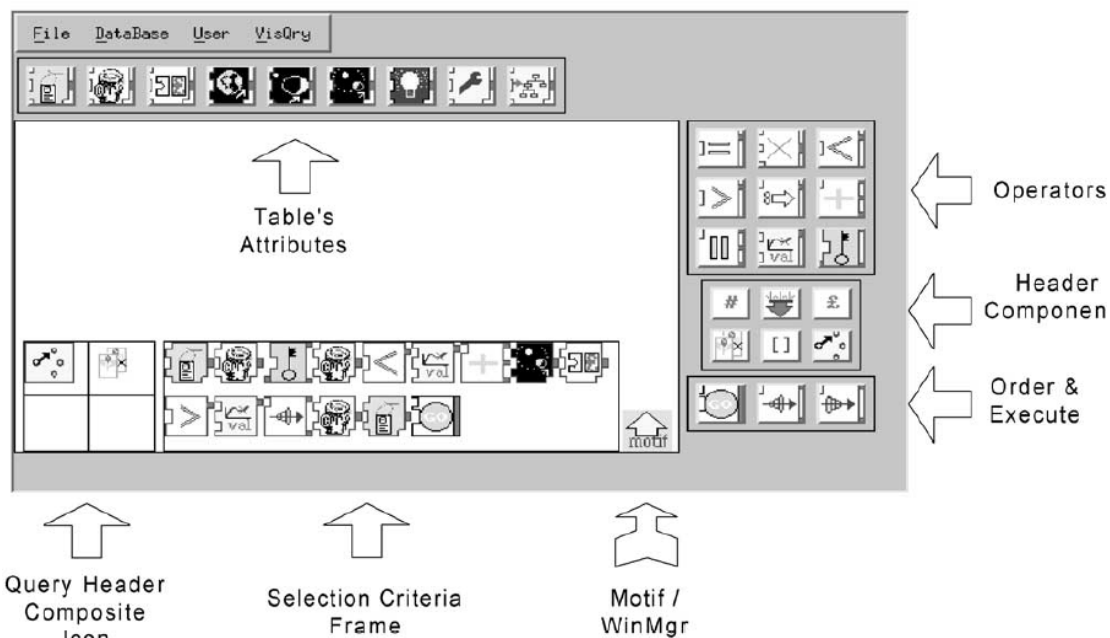
### **2.3.1 Συμβολικές γλώσσες**

Οι συμβολικές γλώσσες χρησιμοποιούνται με σκοπό την απλή και άμεση πρόσβαση του χρήστη στις πληροφορίες μιας βάσης δεδομένων, μέσω της συμβολικής αναπαράστασης επερωτήσεων. Οι πληροφορίες αυτές που βρίσκονται μέσα στην δομή του συστήματος μπορούν με αυτόν τον τρόπο αυτό να γίνουν εύκολα προσβάσιμες ακόμα και από έναν χρήστη ο οποίος δεν έχει ιδιαίτερη εμπειρία και γνώσεις επί του θέματος. Οι άνθρωποι γενικότερα προτιμούν να αλληλεπιδρούν με κάτι απλό και κατανοητό που αντιπροσωπεύει τον πραγματικό κόσμο στον οποίο ζούνε από το να έρχονται σε επαφή με την πολυπλοκότητα των μοντέλων βάσεων

Οι συμβολικές γλώσσες έχουν το πλεονέκτημα να αντιπροσωπεύουν μεγάλη πληθώρα εντολών και αιτημάτων προς το σύστημα πράγμα που οφείλεται στην ευρεία αντίληψη των οπτικών ερεθισμάτων που λαμβάνει ο εκάστοτε χρήστης. Οι εντολές μπορεί να διαφέρουν ανάλογα με το χρώμα το μέγεθος και το σχήμα του σύμβολου ή ακόμα, από την συνολική του εμφάνιση ή από το σημείο στο οποίο είναι τοποθετημένο.

Με βάση αυτά υπάρχει η δυνατότητα από ένα σύνολο ταξινομημένων κατανοήσιμων και σαφώς ορισμένων συμβόλων να δημιουργηθεί ολόκληρο λεξιλόγιο επερωτήσεων. Αυτό μπορεί να επιτευχθεί καταρχήν με την ταξινόμηση των συμβόλων σε ομάδες ανάλογα με την χρησιμότητα τους μέσα στην έκφραση της ερώτησης. Δηλαδή,

κάποια εικονίδια θα αντιπροσωπεύουν ρήματα (SELECT, UPDATE, DELETE, κτλ), κάποια άλλα άρθρα (From, Where), ενώ κάποια άλλα αντωνυμίες (all κ.α.) έτσι, με τον συνδυασμό διαφορετικών συμβόλων από κάθε ομάδα θα έχουμε πολλούς διαφορετικούς συνδυασμούς εκφράσεων με αποτέλεσμα να μπορούμε να εκτελέσουμε μεγάλη πληθώρα επερωτήσεων προς τη βάση δεδομένων.



**Εικόνα 1: Παράδειγμα συμβολικής επερώτησης**

Στην Εικόνα 1 παρουσιάζεται ένα παράδειγμα εικονικής τεχνικής υποβολής ερωτημάτων όπου σχηματίζεται μια επερώτηση με βάση τη διαφορετική λειτουργία του κάθε εικονιδίου. Αυτός ο τρόπος σύνταξης επερωτήσεων αναφέρεται εδώ ενδεικτικά κυρίως για να δείξει τις μεγάλες δυνατότητες που προσφέρουν οι συμβολικές γλώσσες όσον αφορά την ευκολία και την φιλικότητα προς όλους τους χρήστες που αλληλεπιδρούν με το σύστημα ανεξαρτήτως γνώσεων πληροφορικής.

Στην συγκεκριμένη εργασία τα σύμβολα χρησιμοποιούνται και γίνονται αντιληπτά απ' τον χρήστη με βάση τις έξι ιδιότητες τους όπως θέση στο χώρο, σχήμα, μορφή, χρώμα, κλπ. Ο στόχος του κάθε σύμβολου είναι να αναπαραστήσει μέσω των ιδιοτήτων του μια συγκεκριμένη έννοια. Στην προκειμένη περίπτωση η έννοια αυτή μπορεί να είναι είτε μια συγκεκριμένη τοποθεσία είτε μια συγκεκριμένη δραστηριότητα, είτε ο συνδυασμός τους, δηλαδή μια δραστηριότητα που συμβαίνει σε συγκεκριμένη τοποθεσία. Με τον τρόπο αυτό ο χρήστης αλληλεπιδρά με το σύστημα μέσω απλών κινήσεων επιτυγχάνοντας την υποβολή ερωτημάτων προς τη βάση δεδομένων με εύκολο τρόπο χωρίς να απαιτείται ο χρήστης να γνωρίζει τις διεργασίες που λαμβάνουν χώρα στο υπόστρωμα της εφαρμογής.

### 2.3.2 Χάρτες

Κατά τη διάρκεια των τελευταίων 20 χρόνων τα χωρικά δεδομένα γίνονται όλο και πιο σημαντικά ενώ οι χρήστες συνειδητοποιούν ότι είναι πιο αποδοτικό να αναφέρονται σε πληροφορίες χωρικά παρά σε αφηρημένες φόρμες. Αυτή η τάση προήγαγε την ανάπτυξη των συστημάτων ώστε να επιτρέπουν στους χρήστες να

χειρίζονται, να λαμβάνουν, να αναλύουν και να παρουσιάζουν χωρικά δεδομένα. Συστήματα με τέτοιες ιδιότητες ονομάζονται GIS (geographical information systems).

Τα πρώιμα GIS χρησιμοποιούνταν για την παραγωγή χαρτών, δημογραφικών και άλλων γεωγραφικών καταστάσεων. Σήμερα έχουν αναπτυχθεί και βρήκαν το δρόμο τους και προς τις προγραμματιστικές εφαρμογές, διευρύνοντας έτσι την γκάμα των χρηστών τους από μια μικρή ελίτ τεχνικά εκπαιδευμένων μηχανικών σ' ένα μεγάλο εύρος καθημερινών ανθρώπων.

Η αποθήκευση, η διαχείριση και ανάλυση των πληροφοριών είναι σημαντικό ζήτημα για κάθε σύστημα GIS. Παρόλα αυτά η πληροφορία είναι χρήσιμη μόνο όταν μπορεί να ανακτηθεί. Η διαδικασία της ανάκτησης της πληροφορίας αποτελείται από τρία βασικά στάδια. Το πρώτο αφορά στη διαμόρφωση των ερωτήσεων. Σ' αυτό το στάδιο ο χρήστης προσδιορίζει τι ακριβώς ψάχνει. Ακολούθως, απαιτείται απ' το σύστημα να ψάξει μέσα στα διαθέσιμα δεδομένα του και να ανακτήσει τις εγγραφές εκείνες οι οποίες ταιριάζουν με τις απαιτήσεις ή τα κριτήρια που ο χρήστης έθεσε. Αν αυτές βρεθούν τότε η παρουσίαση των αποτελεσμάτων αποτελεί το τρίτο βήμα που ολοκληρώνει τη διαδικασία.

Είναι επίσης καλό να οριστούν και κάποιες απαιτήσεις συστήματος για καθ' ένα απ' τα παραπάνω βήματα. Η διαδικασία διαμόρφωσης ερωτήσεων πρέπει να είναι απλή, κατανοητή και εκφραστική έτσι ώστε να ερωτώνται οι σωστές και επιθυμητές ερωτήσεις. Κατά τη διάρκεια της διερεύνησης της βάσης δεδομένων έμφαση πρέπει να δοθεί στην ακρίβεια και την αποτελεσματικότητα, αφού τα αποτελέσματα θα πρέπει να απαντούν ακριβώς στα ερωτηματολόγια του χρήστη. Η αποτελεσματικότητα της διερεύνησης εξαρτάται απ' το κατά πόσο το περιεχόμενο της ερώτησης συγκρίνεται σωστά με τις πληροφορίες που είναι αποθηκευμένες στο σύστημα. Στο βήμα της παρουσίασης των αποτελεσμάτων είναι απαραίτητο οι αποκομισμένες πληροφορίες να παρουσιάζονται με κατανοητό τρόπο, χωρίς να απασχολούν το χρήστη με περιττές λεπτομέρειες.

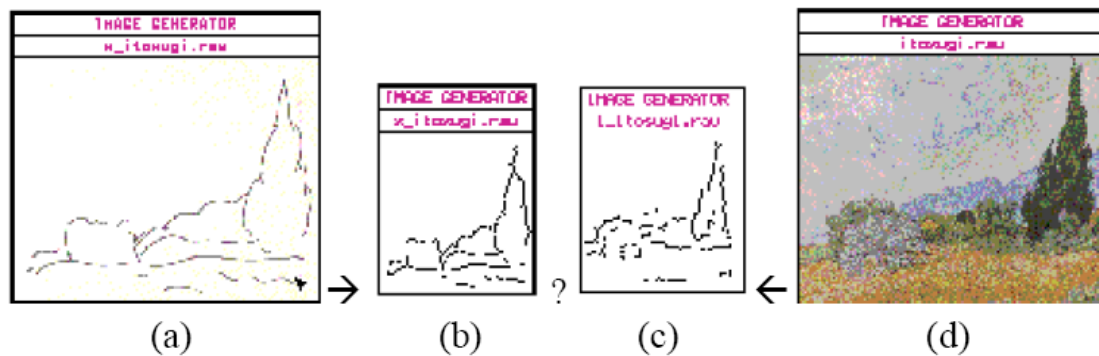
Γενικότερα τα συστήματα αποκόμισης χωρικών πληροφοριών μπορούν να χωριστούν σ' αυτά τα συστήματα που επιτρέπουν τη δημιουργία ερωτημάτων που είναι επικεντρωμένα στα πεδία και σ' εκείνα που είναι επικεντρωμένα στα αντικείμενα. Ο λόγος για τον οποίον κάποια συστήματα προτιμούν τη μία προσέγγιση απ' την άλλη εξαρτάται κυρίως απ' τον τύπο των δεδομένων που απαιτούνται κάθε φορά. Τα πεδία είναι περιοχές χώρου που χαρακτηρίζονται από συγκεκριμένες ιδιότητες. Συγκεκριμένα σημεία μπορούν να περιλαμβάνουν πολλαπλά χαρακτηριστικά αλλά δεν περιλαμβάνουν καμιά σημαντική πληροφορία για το περιβάλλον τους. Απλές εικόνες ή ψηφιακά γήινα μοντέλα αποτελούν τυπικά παραδείγματα πεδίων. Τα αντικείμενα απ' την άλλη, είναι φυσικές αναπαραστάσεις οντοτήτων και συχνά περιέχουν πληροφορίες υψηλού επιπέδου. Συνήθως φτιάχνονται μετά από δύλιση και ερμηνεία των δεδομένων. Τα αντικείμενα έχουν μοναδική ταυτότητα και έχουν όχι μόνο ιδιότητες και χαρακτηριστικά αλλά και συγκεκριμένες λειτουργίες. Έτσι, ένα αντικείμενο μπορεί να περιέχει γνώση σχετικά με το περιβάλλον του.

Αν ένα σύστημα ανάκτησης πληροφορίας επιτρέπει στο χρήστη να ρωτήσει για πολλαπλά χαρακτηριστικά (πεδία) ή περιπτώσεις (αντικείμενα) στην ίδια πρόταση, τότε μπορούν να ληφθούν υπόψη και οι σχέσεις τους. Αυτό το παραπάνω χαρακτηριστικό μας επιτρέπει να κατατάξουμε τα συστήματα αυτά σε τέσσερις διαφορετικές κατηγορίες:

- οπτικά ερωτηματολόγια σχετικά με την ύπαρξη πεδίων
- οπτικά ερωτηματολόγια σχετικά με τα πεδία και τις χωρικές σχέσεις
- οπτικά ερωτηματολόγια σχετικά με την ύπαρξη αντικειμένων
- οπτικά ερωτηματολόγια σχετικά με την ύπαρξη αντικειμένων και τις χωρικές σχέσεις

## Ύπαρξη πεδίων

Τα συστήματα που βασίζονται στη χρήση πεδίων ή χαρακτηριστικών για την ανάκτηση δεδομένων στρέφουν το ενδιαφέρον τους κυρίως στην ύπαρξη τοπικών χαρακτηριστικών. Αν πολλαπλά χαρακτηριστικά ερωτηθούν τότε οι σχέσεις τους δεν λαμβάνονται υπόψη. Παρόλα αυτά, κάποια συστήματα επιτρέπουν σ' ένα χρήστη να ορίσει τη τοποθεσία των χαρακτηριστικών αυτών. Σ' αυτές τις περιπτώσεις στη σχέση με το συγκεκριμένο πλαίσιο συνήθως μια εικόνα εγκαθιδρύεται. Ένα χαρακτηριστικό παράδειγμα τέτοιων συστημάτων είναι το trademark και το art museum. Το σύστημα trademark σχεδιάστηκε για να ανακτά γραφικά σύμβολα ενώ το art museum (Εικόνα 2) για να ψάχνει πίνακες ζωγραφικής σε μια μεγάλη βάση εικόνων. Το art museum επιτρέπει στο χρήστη να ζωγραφίσει περίπου την εικόνα για την οποία ενδιαφέρεται.



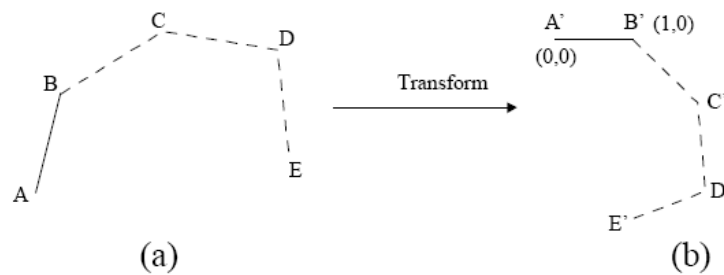
Εικόνα 2: Οι βασικές αρχές του art museum (a) το αρχικό σκίτσο (b) Το αποτέλεσμα της εξαγωγής γραμμών με το οποίο συγκρίνεται (c) Τα μεταδεδομένα τα οποία τραβήχτηκαν απ' τον Αρχικό πίνακα (d)

Μερικά συστήματα που βασίζονται σε πεδία ή χαρακτηριστικά και επιτρέπουν στους χρήστες να ρωτήσουν για πολλαπλά πεδία την ίδια στιγμή, παίρνουν υπόψη και τις χωρικές σχέσεις μεταξύ των πεδίων. Γι' αυτό το σκοπό θεωρούνται περιοχές έτσι ώστε να μπορούν να έχουν χωρικές αναφορές μεταξύ τους. Οι χωρικές σχέσεις μεταξύ πεδίων τυπικά ορίζονται βάση της σχετικής θέσης των κέντρων τους. Τέτοια παραδείγματα είναι το qbic (query by image content) και το image query by semantical content τα οποία είναι συστήματα παρόμοια με το art museum, μόνο που λαμβάνουν υπόψη και ιδιότητες όπως ποιότητα χρώματος, υφή κ.α.

Μια άλλη κατηγορία συστημάτων ασχολούνται με την ύπαρξη των αντικειμένων, όπως και με ορισμένες ιδιότητες των αντικειμένων π.χ. σχήμα και γεωμετρία. Αυτά είναι παρόμοια με την αντίστοιχη προσέγγιση των συστημάτων βασισμένων στα πεδία. Αν ερωτηθούν πολλαπλά αντικείμενα, τότε οι χωρικές σχέσεις δεν λαμβάνονται υπόψη. Χωρικές σχέσεις μπορεί να θεωρηθεί ότι έχουμε όταν η επερώτηση περιέχει παραπάνω από ένα αντικείμενα. Ένα τέτοιο σύστημα είναι και η

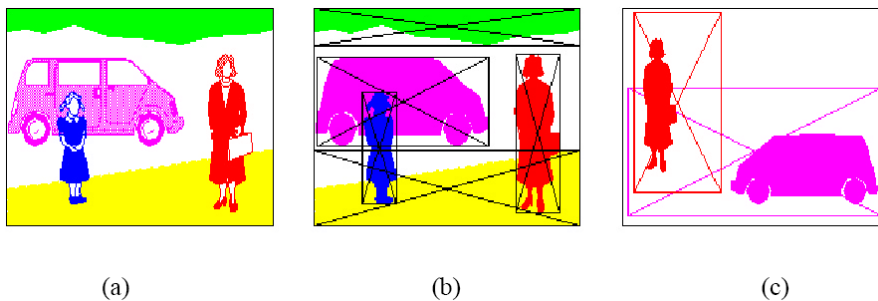
μέθοδος mehrotra and gary (Εικόνα 3) που προσπαθεί να τραβήξει παρόμοια σχήματα από μία βάση δεδομένων με εικόνες.

Τα συστήματα που περιέχουν τα μέσα για να κάνουν ερωτήσεις για αντικείμενα και τις χωρικές τους σχέσεις παρέχουν το μεγαλύτερο βαθμό ευχρηστίας για την ανάκτηση χωρικών δεδομένων. Μπορούμε να διακρίνουμε δύο τύπους. Τα συστήματα με συμβολικές αναπαραστάσεις επιτρέπουν στο χρήστη να φτιάξει ένα ερωτηματολόγιο χρησιμοποιώντας συνδυασμούς εικονιδίων και συμβόλων που αντιπροσωπεύουν αντικείμενα. Τέτοια παραδείγματα είναι το visualseek (Εικόνα 4) που επιτρέπει στο χρήστη να ψάξει για εικόνες συγκρίνοντας τοπικά και γενικά χαρακτηριστικά.

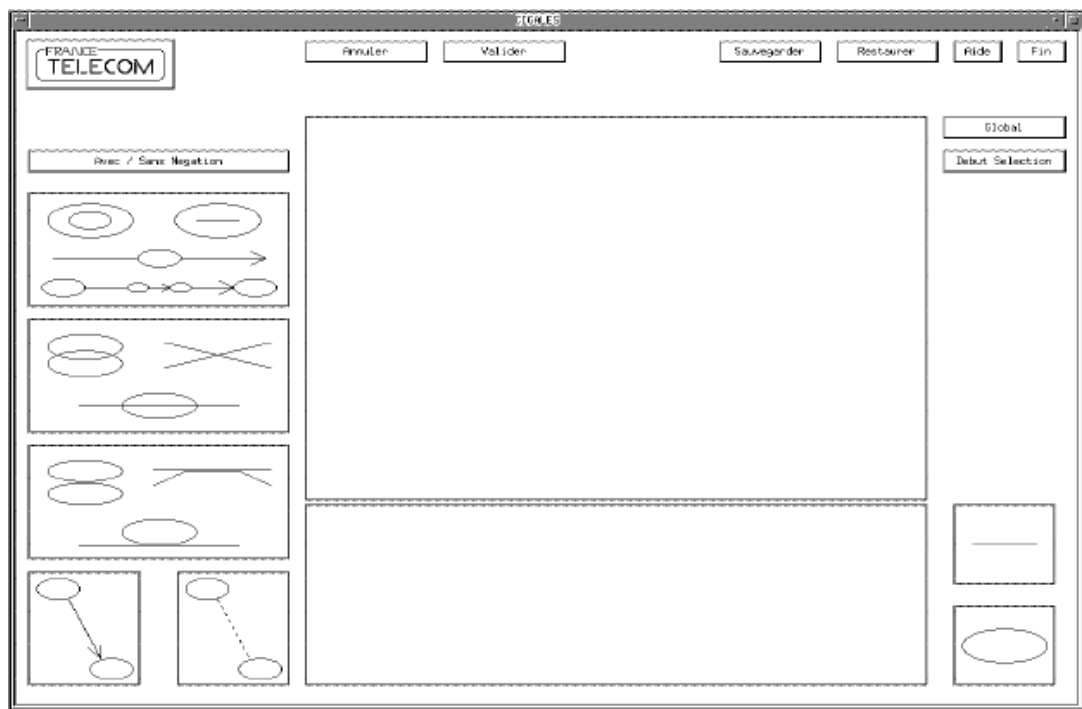


**Εικόνα 3: Παράδειγμα κανονικοποίησης ενός ανακτημένου πολύγону mehrotra and gary**

Επίσης υπάρχει και το sigales (Εικόνα 5) το οποίο βασίζεται στην ίδια ιδέα με το visualseek αλλά επικεντρώνεται κυρίως σε γεωγραφικές αναπαραστάσεις παρά σε εικόνες. Ένας χρήστης μπορεί να φτιάξει ένα χωρικό ερωτηματολόγιο με μια μικρή συλλογή από εικονίδια που βρίσκονται πάνω σε μια γραφική διεπαφή χρήστη.

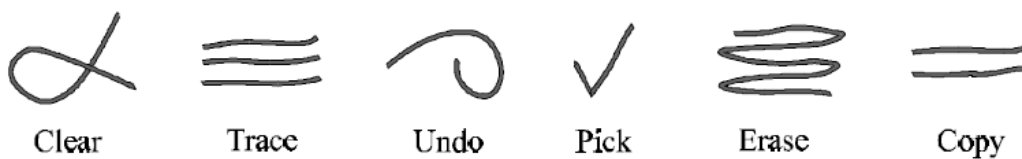


**Εικόνα 4: Αποσύνθεση της εικόνας σε τοπικές περιοχές χρώματος (a) Η αρχική εικόνα (b) η αποσυντεθειμένη εικόνα (c) Η αποκοιμισμένη εικόνα**



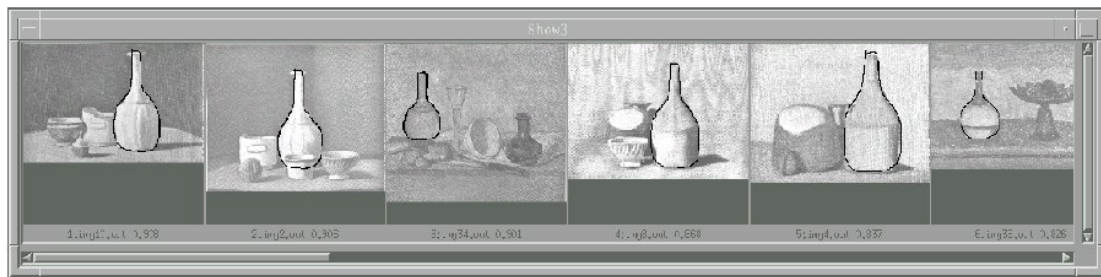
**Εικόνα 5: Η διεπαφή χρήστη του cigales**

Η δεύτερη κατηγορία συστημάτων είναι αυτή που χρησιμοποιείται η τεχνική του σκισσαρίσματος για να διαμορφωθούν οι επερωτήσεις. Το σκισσάρισμα είναι μια λιγότερο περιοριστική μέθοδος για να φτιαχτεί ένα ερωτηματολόγιο απ' το να φτιαχτεί χρησιμοποιώντας εικονίδια. Άρα τα συστήματα βασισμένα στο σκισσάρισμα επιτρέπουν τους χρήστες να εκφραστούν πιο ελεύθερα. Παραδείγματα τέτοιων συστημάτων είναι το electronic cocktail napkin (Εικόνα 6) που επιτρέπει στους σχεδιαστές να μεταφέρουν τις ιδέες τους σε ηλεκτρονικό χαρτί. Το σύστημα αυτό μπορεί να αναγνωρίσει, να ερμηνεύσει και να διαχειριστεί σχεδιαγράμματα, χειρονομίες και διαγράμματα.



**Εικόνα 6: Ένα σετ εντολών σε μορφή χειρονομιών που προσφέρει το electronic cocktail napkin**

Σε αυτή την κατηγορία εντάσσονται και τα συστήματα που υποστηρίζουν ανάκτηση εικόνων μέσω σκισσαρίσματος (π.χ. βλέπε Εικόνα 7) που επιτρέπει σ' ένα χρήστη να ανακτήσει αντικείμενα βασισμένα σε σχέδια. Το σχέδιο του αντικειμένου μετατρέπεται σ' ένα ελαστικό πίνακα ο οποίος ταιριάζετε μ' άλλα ανακτημένα σχήματα αντικειμένων από εικόνες.



**Εικόνα 7:** Ανιχνευμένα σχήματα που μοιάζουν με μπουκάλια τα οποία ανακτήθηκαν από μία βιβλιοθήκη πινάκων βασισμένα σ' ένα ελεύθερο σχέδιο ενός μπουκαλιού(δε φαίνεται στο σχήμα)

## **2.4 Τρόποι αναπαράστασης αποτελέσματος**

### **2.4.1 Ο κλασικός πίνακας**

Ο πίνακας αποτελεί ένα στοιχειώδη και απλό τρόπο ομαδοποίησης και αναπαράστασης αποτελεσμάτων. Είναι ένας τρόπος οπτικής επικοινωνίας ο οποίος χαρτογραφεί τη λογική δομή μιας ομάδας δεδομένων σ' ένα ιεραρχικό πλέγμα. Η χρήση του πίνακα για παρουσίαση αποτελεσμάτων βοηθάει ακόμα και τον μη εξοικειωμένο χρήστη να κατανοήσει τα περιεχόμενά του και αυτό χάρη στην οργανωτική του δομή και στον ευδιάκριτο διαχωρισμό των στοιχείων του. Εκμεταλλεύεται δηλαδή τη φυσική ροπή του ανθρώπου να οργανώνει τις πληροφορίες σε πίνακες. Τα δεδομένα του πίνακα αποτελούνται από διακριτά δεδομένα και από συνδυασμό εξαρτημένων και ανεξάρτητων μεταβλητών. Ένας μονοδιάστατος πίνακας ,όπως μια λίστα ή ένας πίνακας περιεχομένων έχει μία μοναδική ιεραρχία. Ένας δισδιάστατος πίνακας έχει δύο ή περισσότερες ιεραρχίες.

Στις σχεσιακές βάσεις δεδομένων ο πίνακας είναι μια ομάδα δεδομένων η οποία οργανώνεται σ' ένα μοντέλο από οριζόντιες γραμμές και κάθετες στήλες. Οι στήλες αναγνωρίζονται με ένα όνομα ενώ οι γραμμές απ' τις τιμές οι οποίες εμφανίζονται σε κάθε καταχώριση. Ένας πίνακας έχει συγκεκριμένο αριθμό στηλών αλλά απεριόριστο αριθμό γραμμών.

Οι πίνακες ,αν και όπως είπαμε πιο πάνω αντιπροσωπεύουν στοιχειώδη τρόπο αναπαράστασης αποτελεσμάτων, δεν έχουν χάσει την αξία τους. Αποτελούν ένα χρήσιμο εργαλείο κυρίως στον τομέα της πληροφορικής(βάσεις δεδομένων και αλλού) αλλά και σε άλλους επιστημονικούς τομείς όπως στατιστική, η μετεωρολογία, η βιολογία, η χημεία κ.α.



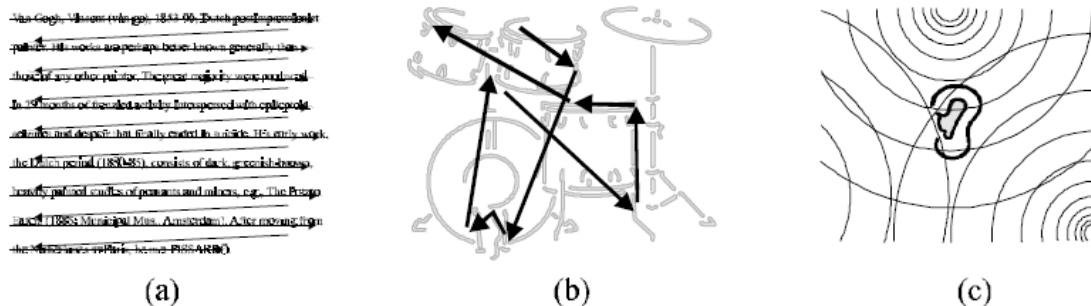
```
/*Database Information For - information_schema*/
```

Name	Engine	Version	Row_format	Rows	Avg_row_length
CHARACTER_SETS	MEMORY	0	Fixed	(NULL)	576
COLLATIONS	MEMORY	0	Fixed	(NULL)	423
COLLATION_CHARACTER_SET_APPLICABILITY	MEMORY	0	Fixed	(NULL)	387
COLUMNS	MyISAM	0	Dynamic	(NULL)	0
COLUMN_PRIVILEGES	MEMORY	0	Fixed	(NULL)	2565
KEY_COLUMN_USAGE	MEMORY	0	Fixed	(NULL)	4637
ROUTINES	MyISAM	0	Dynamic	(NULL)	0
SCHEMATA	MEMORY	0	Fixed	(NULL)	3656
SCHEMA_PRIVILEGES	MEMORY	0	Fixed	(NULL)	2179
STATISTICS	MEMORY	0	Fixed	(NULL)	2679
TABLES	MEMORY	0	Fixed	(NULL)	3641
TABLE_CONSTRAINTS	MEMORY	0	Fixed	(NULL)	2504
TABLE_PRIVILEGES	MEMORY	0	Fixed	(NULL)	2372
TRIGGERS	MyISAM	0	Dynamic	(NULL)	0
VIEWS	MyISAM	0	Dynamic	(NULL)	0
USER_PRIVILEGES	MEMORY	0	Fixed	(NULL)	1986

Εικόνα 8: Παράδειγμα πίνακα που περιέχει πληροφορίες για βάση δεδομένων

## 2.4.2 Γραφικοί τρόποι (οπτικές αναπαραστάσεις)

Πολλοί συγγραφείς συμφωνούν ότι οι οπτικές αναπαραστάσεις δεδομένων είναι αποτελεσματικές στο να παρουσιάζουν διαφορετικά είδη γνώσης. Ένα απ' τα κύρια πλεονεκτήματα τους βασίζεται στην ικανότητά τους να επιτρέπουν την εξαγωγή συμπερασμάτων μέσω της αντίληψης αντικαθιστώντας έτσι τις κουραστικές συγκρίσεις και υπολογισμούς.



Εικόνα 9: Οι τρεις βασικές δραστηριότητες του χρήστη που αφορούν την αντίληψη (a) ανάγνωση (b) σάρωση (c) ακρόαση

Η καλή αντίληψη είναι βασική για την σωστή αλληλεπίδραση με τον υπολογιστή. Ενώ μερικές ενέργειες του χρήστη είναι προαιρετικές, η αντίληψη δεν είναι. Η ικανότητα των ανθρώπων να αντιλαμβάνονται εξαρτάται βασικά απ' την όραση και την ακοή τους, γι' αυτό οι τρεις βασικές δραστηριότητες που αφορούν την αντίληψη είναι (βλέπε Εικόνα 9) η ανάγνωση ερεθισμάτων (π.χ. διαδοχικών λέξεων), η σάρωση που επιτρέπει εστίαση στα βασικά σημεία ενδιαφέροντος όπως τίτλους και επικεφαλίδες ή σε μέρη των γραφικών που τους έχει δοθεί έμφαση και τέλος η ακρόαση που συνήθως αφορά την αντίληψη του ήχου.

Η γραφική αναπαράσταση των δεδομένων μιας εφαρμογής μπορεί να χρησιμοποιήσει ποικίλες τεχνικές για να τραβήξει την προσοχή του χρήστη σ' ένα συγκεκριμένο μήνυμα. Μια δυνατότητα για να επικεντρωθεί οπτικά η προσοχή του χρήστη είναι να δημιουργηθεί τοπικά μια αντίθεση στη διεπαφή. Αυτό μπορεί να περιλαμβάνει αντίθεση στο χρώμα, το σχήμα ή το μέγεθος των υπάρχοντων στοιχείων ή την

εμφάνιση νέων γραφικών στην οθόνη. Κινούμενα στοιχεία και αντικείμενα σε μια κατά τα άλλα στατική παρουσίαση, όπως μια μπάρα προόδου, τραβούν επίσης τη προσοχή του χρήστη.

Παρακάτω παρουσιάζουμε τους βασικούς τρόπους γραφικής αναπαράστασης δεδομένων σύμφωνα με την άποψη πολλών συγγραφέων και ειδικών.

### **Παραστάσεις βασισμένες σε φόρμες**

Μια φόρμα (form) είναι μια ονομαστική συλλογή από αντικείμενα που έχουν την ίδια δομή. Είναι η πρώτη προσπάθεια να αφήσουμε το μονοδιάστατο χώρο του κειμένου, εκμεταλλευόμενοι το δισδιάστατο χαρακτηριστικό της οθόνης ενός υπολογιστή. Αυτό διευκολύνει τους μη ειδικούς χρήστες αφού αξιοποιείται η φυσική τάση των ανθρώπων να χρησιμοποιήσουν κανονικές δομές ή /και να οργανώνουν τα δεδομένα σε πίνακες. Το κύριο χαρακτηριστικό των διεπαφών χρήστη-υπολογιστή τύπου φόρμας είναι ότι αποδίδουν μια δομημένη αναπαράσταση που αντιστοιχεί σε μια αφαίρεση των συμβατικών εντύπων. Μια φόρμα μπορεί να θεωρηθεί ως ορθογώνιο πλέγμα που έχει συστατικά μέρη που μπορεί να είναι οποιοσδήποτε συνδυασμός από κυψέλες (cells) και ομάδες από κυψέλες ή sub Form. Ένα form cell είναι η μικρότερη μονάδα δεδομένων που μπορεί να αναφέρει σε μια εφαρμογή ενός χρήστη. Μια form είναι μια γενίκευση ενός πίνακα (table), με την έννοια ότι τα μέρη ενός πίνακα είναι συνήθως στοιχειώδη cells και δεν επιτρέπεται να ενσωματωθούν. Οι σχέσεις που αναπαριστούνται μέσω μιας φόρμας μπορεί να είναι είτε ανάμεσα σε κυψέλες, των υποσυνόλων, είτε του γενικού συνόλου, παρέχοντας έτσι στο χρήστη τρία επίπεδα πληροφοριών. Αυτό επιτρέπει σε ερωτήματα που αφορούν είτε τις μεμονωμένες κυψέλες (στοιχειώδης ερώτηση), τα υποσύνολα των κύψελων (ενδιάμεση ερώτηση), είτε το γενικό σύνολο των κύψελων (γενικό ερώτημα) για να απαντηθούν.

Η φόρμα μπορεί να χρησιμοποιηθεί είτε ως μηχανισμός υποβολής ερωτημάτων είτε ως μέσο αναπαράστασης των αποτελεσμάτων μιας ερώτησης. Στην Εικόνα 10 παρουσιάζεται ένα παράδειγμα της πρώτης περίπτωσης όπου η φόρμα χρησιμοποιείται για την υποβολή της ερώτησης προς το σύστημα διαχείρισης μιας βάσης δομένων. Προφανώς, αυτό που υπονοείται είναι ότι τα δεδομένα της φόρμας υποβάλλονται σε προεργασία προκειμένου να καταλήξουν σε μια κλασική διατύπωση ενός ερωτήματος (π.χ. σε SQL). Αντίθετα, στην Εικόνα 11 η φόρμα χρησιμοποιείται για την αναπαράσταση αποτελεσμάτων. Η συγκεκριμένη φόρμα αναπαριστά την πληρότητα του ξενοδοχείου σε ένα έτος. Μια στοιχειώδης ερώτηση θα μπορούσε να είναι: «*Ποιο είναι το ποσοστό των πελατών που είναι πάνω από 55 χρονών το μήνα Μάιο;*». Η απάντηση σε μια τέτοια ερώτηση μπορεί να βρεθεί εύκολα από μια ενιαία αντιστοίχιση μέσα στη φόρμα. Μια ενδιάμεση ερώτηση θα μπορούσε να είναι:

*«Ποιο είναι το ποσοστό των θηλυκών πελατών το χειμώνα;».*

Σε αυτή την περίπτωση, η απάντηση προκύπτει παρατηρώντας μια ομάδα από αντιστοιχίες στη φόρμα.

Εικόνα 10: Η χρήση της φόρμας για την εισαγωγή δεδομένων και τον προσδιορισμό των κριτηρίων μιας ερώτησης

Ja	Fe	Ma	Ap	Ma	Jun	Ju	Au	Se	Oc	No	De	
26	21	26	28	20	20	20	20	20	40	15	40	% Female Clients
69	70	77	71	37	36	39	39	55	60	68	72	% Local Clients
7	6	3	6	23	14	19	14	9	6	8	8	% European Clients
0	0	0	0	0	8	6	6	2	4	0	0	% North American Clients
20	15	14	15	23	27	22	30	27	19	19	17	% South American Clients
1	0	0	8	6	4	6	6	2	1	0	1	% African Clients
3	9	6	0	1	1	8	5	3	12	5	2	% Oriental Clients
78	80	85	86	85	87	70	76	87	85	87	80	% For Vacations
22	20	15	14	15	13	30	24	13	15	13	20	% For Business
2	2	4	2	2	1	1	2	2	4	2	5	% Age<20
25	27	37	35	25	25	27	28	24	30	24	30	% Age 20-35
48	49	42	48	54	55	53	51	55	46	55	43	% Age 35 - 55
25	22	17	15	19	19	19	19	19	20	19	22	% Age >55
67	82	70	83	74	77	56	62	90	92	78	55	% Room occupation

Εικόνα 11: Πληροφορίες για τους πελάτες ενός ξενοδοχείων κατά τη διάρκεια ενός έτους

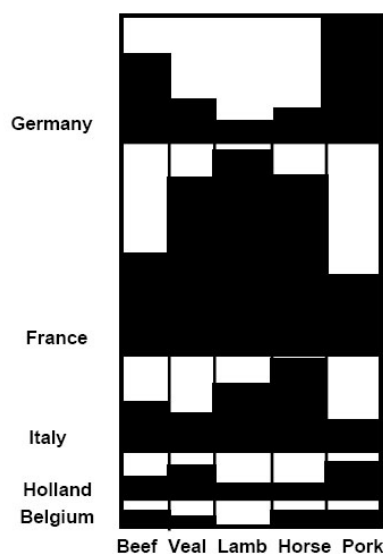
Μια ενδιαφέρουσα προσέγγιση που προτείνεται στο EMBS στοχεύει στο να ξεπεράσει τον έμφυτο περιορισμό των φορμών στους συσχετισμούς αναπαράστασεων μεταξύ των δεδομένων. Το EMBS επιδεικνύει δύο χαρακτηριστικά στοιχεία (βλέπε Εικόνα 12): κυψέλες και κουμπιά. Οι κυψέλες περιέχουν τιμές στοιχείων (που οργανώνονται σε έναν πίνακα) οι οποίες μπορούν να καθοριστούν από το χρήστη συμπληρώνοντας τις τιμές του παραδείγματος. Επιλέγοντας ένα κουμπί ή αλλάζοντας την τιμή μιας κυψέλης προκαλεί τον χειρισμό των δεδομένων. Έξι πρωτόγονες λειτουργίες δίνονται, οι οποίες μπορούν να συνδυαστούν περαιτέρω για να εκτελέσουν τις σύνθετες ενέργειες.

Εικόνα 12 Παράδειγμα μιας Form – Based αναπαράστασης στο EMBS

### Παραστάσεις βασισμένες σε διαγράμματα

Τα δεδομένα που περιλαμβάνονται σε μια φόρμα μπορούν να κατανοηθούν καλύτερα εάν χρησιμοποιούμε κάποιο είδος γραφικής αναπαράστασης που μπορεί να παρουσιάσει καλύτερα τις σχέσεις μεταξύ των δεδομένων αυτών. Παραδείγματος χάριν, οι σειρές των αριθμών μπορούν να μετασχηματιστούν σε ένα σχεδιάγραμμα στο οποίο το ύψος κάθε στήλης είναι ανάλογο προς την αξία του αντίστοιχου αριθμού. Ένα παράδειγμα αναπαράστασης αυτής της κατηγορίας είναι η παρουσίαση της παραγωγής κρέατος στις ευρωπαϊκές χώρες (βλέπε Εικόνα 13).

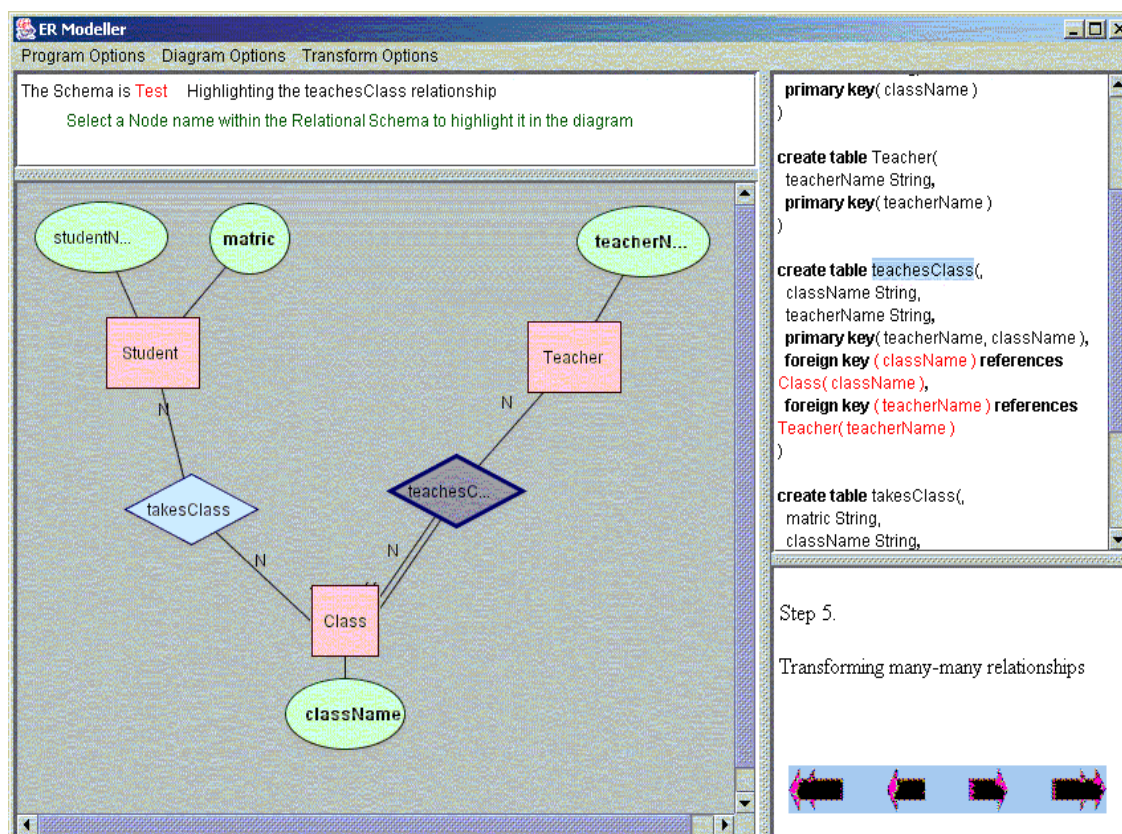
	Beef	Veal	Lamb	Horse	Pork
Germany	32	15	5	6	45
France	38	60	70	60	29
Italy	17	12	20	24	9
Holland	7	10	4	4	11
Belgium	6	3	1	6	6



Beef Veal Lamb Horse Pork

### Εικόνα 13: Φόρμα για αντιπροσώπηση διαγραμμάτων - παραγωγή κρέατος σε μερικές ευρωπαϊκές χώρες

Ακόμα, αυτά τα σχεδιαγράμματα μπορούν να τακτοποιηθούν με διαφορετικούς τρόπους για να τονίσουν άλλες σχέσεις μεταξύ των δεδομένων. Τα διαγράμματα που υιοθετούνται συχνά στα VQS χρησιμοποιούν σαν βασικά (οπτικά) συστατικά μέρη είτε σημεία είτε απλά γεωμετρικά σχήματα (τετράγωνα, ορθογώνια, κύκλοι, κ.λ.π.). Γενικά, ένα διάγραμμα χρησιμοποιεί τα οπτικά συστατικά που έχουν ένα προς ένα αντιστοιχία με τους συγκεκριμένους εννοιολογικούς τύπους. Οι γραμμές δείχνουν τους λογικούς τύπους σχέσης μεταξύ των στοιχείων. Μερικές φορές οι ετικέτες περιέχονται επίσης στα διαγράμματα για δηλωτικούς σκοπούς (βλ. στην Εικόνα 14β ένα παράδειγμα ενός διαγράμματος όπου τα ορθογώνια αντιπροσωπεύουν τις οντότητες στο πρότυπο του μοντέλου οντοτήτων-συσχετίσεων (ER) και τα διαμάντια αναπαριστούν τις σχέσεις). Στο παράδειγμα παρατηρούμε ότι ο χρήστης χρησιμοποιεί αποκλειστικά και μόνο το διάγραμμα Ο-Σ για να διαχειριστεί το σχήμα της βάσης δεδομένων το οποίο παράγεται αυτόματα από το σύστημα όπως φαίνεται στο παράθυρο πάνω δεξιά.



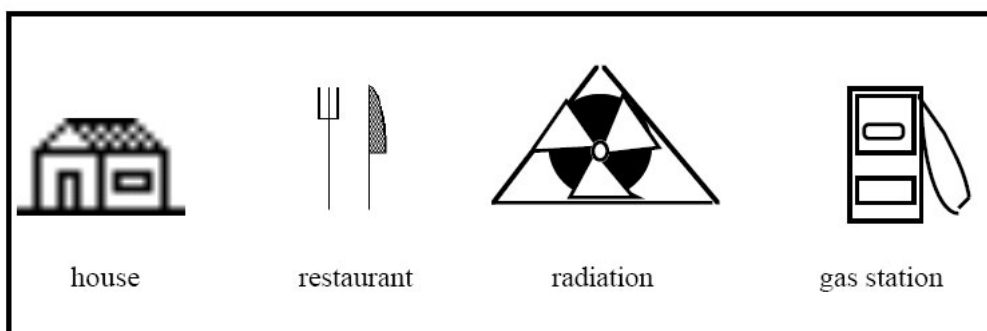
### Εικόνα 14: Χρήση διαγράμματος Ο-Σ για το σχηματισμό σχεσιακού σχήματος

Σε ένα διάγραμμα, εάν τροποποιούμε το σχεδιάγραμμά του με την ακολουθία ορισμένων κανόνων, το περιεχόμενό του μπορεί να παρουσιάσει νέες σχέσεις. Παραδείγματος χάριν, με τη βοήθεια της χωρικής εγγύτητας, το σχεδιάγραμμα μπορεί εύκολα να διαβιβάσει την έννοια της ομοιότητας και την παρουσία μιας έννοιας-πυρήνα γύρω από την οποία όλα τα άλλα συστατικά μέρη έλκονται. Το διάγραμμα στην Εικόνα 14 μπορεί να τροποποιηθεί είτε αποκρύπτοντας τα χαρακτηριστικά μιας οντότητας οπότε η έμφαση είναι στις βασικές σχέσεις της βάσης, είτε προσδιορίζοντας λεπτομερώς την οντότητα παρουσιάζοντας την σε πλήρη μορφή

μαζί με τα γνωρίσματά της και την κατηγορία αυτών (π.χ. γνωρίσματα κλειδιά, ξένα κλειδιά, τύπους συσχετίσεων όπως μερική / ολική συμμετοχή, κλπ).

### Παραστάσεις βασισμένες σε εικόνες

Στην επιστήμη των υπολογιστών, μια εικόνα μπορεί να οριστεί ως μια χωρισμένη σε τμήματα, τυποποιημένη εικόνα. Ο τεμαχισμός της εικόνας υπονοεί την εξαγωγή ενός μόνο συστατικού από το υπόβαθρο ενώ η τυποποίηση αναφέρεται σε μια αναπαράσταση που έχει γίνει από έναν μικρό αριθμό σημαντικών γραμμών που στο σύνολο τους αποτελούν την εικόνα. Δείτε, παραδείγματος χάριν, την εικόνα του σπιτιού στο αριστερό τμήμα στην εικόνα, όπου μόνο απλά γεωμετρικά σχήματα έχουν χρησιμοποιηθεί. Τα VQS πρέπει να αναπαριστούν όχι μόνο εικόνες πραγματικών αντικειμένων, αλλά και αφηρημένες έννοιες, δράσεις ή διαδικασίες. Εάν θέλουμε να αναπαραστήσουμε μια διαδικασία του υπολογιστή, αναγκάζομαστε να επεκτείνουμε το πλαίσιο της εικόνας, δεδομένου ότι δεν έχουμε κάποιο φυσικό οπτικό αντίστοιχο και δεν μπορούμε να εκμεταλλευτούμε την ομοιότητα στην εικόνα μεταξύ μιας τέτοιας εικόνας και της αφηρημένης έννοιας την οποία συστήνουμε. Νέες εικόνες πρέπει να σχεδιαστούν για να εκμεταλλευτούμε τις διαφορετικές μορφές συσχετισμού όπως η αναλογία, η σύμβαση, κ.λ.π., οι οποίες, σε γενικές γραμμές, είναι πιο χαρακτηριστικές για σύμβολα.



**Εικόνα 15: Παραδείγματα των εικόνων**

Ενδεικτικά παραδείγματα παρουσιάζονται στην Εικόνα 15 όπου, πηγαίνοντας από αριστερά στα δεξιά, η δεύτερη εικόνα έχει σχεδιαστεί για να δείξει ένα εστιατόριο, η τρίτη ραδιενέργεια από σύμβαση, και το τελευταίο ένας σταθμός βενζίνης. Εν περίληψη, μπορούμε να συμφωνήσουμε με τον ακόλουθο ορισμό μιας εικόνας στα πλαίσια της επιστήμης των υπολογιστών: «η εικόνα είναι ένα οπτικά τεμαχισμένο αντικείμενο που ενημερώνει το θεατή για ένα εσωτερικό μήνυμα ή πληροφορία (έννοια, λειτουργία, κατάσταση, τρόπος, κ.λ.π.) που έχει οριστεί από το σχεδιαστή».

### Μικτές (Hybrid) προσεγγίσεις

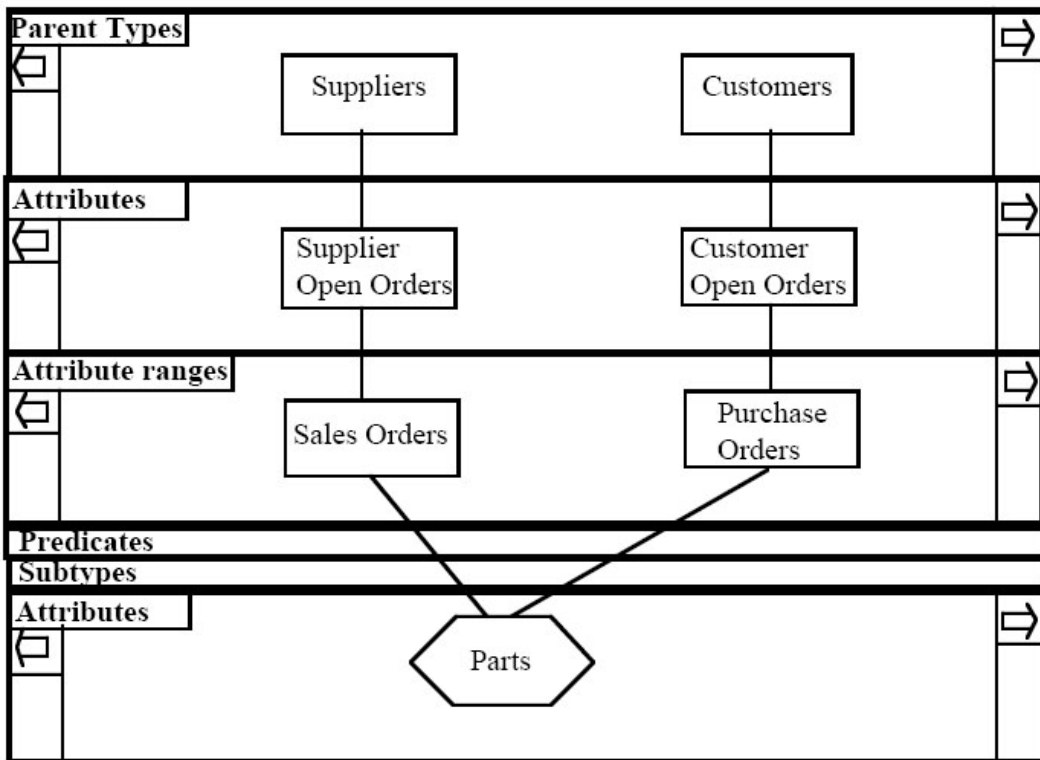
Η υβριδική αναπαράσταση χρησιμοποιεί έναν αυθαίρετο συνδυασμό των τριών παραπάνω εικονικών αναπαραστάσεων, είτε προσφέροντας στο χρήστη ποικίλες εναλλακτικές αναπαραστάσεις των Βάσεων Δεδομένων και των Queries, είτε συνδυάζοντας διαφορετικές εικονικές αναπαραστάσεις σε μια ενιαία αναπαράσταση. Από μια ανάλυση των VQS, προκύπτει ότι τα περισσότερα συστήματα υιοθετούν περισσότερες από μια εικονικές αναπαραστάσεις, αλλά συχνά ένας από αυτούς είναι ο κυρίαρχος. Σε αυτήν την περίπτωση, το σύστημα μπορεί να κατηγοριοποιηθεί

υιοθετώντας το παράδειγμα αναπαράστασης που βασίζεται σε αυτή τη συγκεκριμένη αναπαράσταση. Εντούτοις, στα αποκαλούμενα υβριδικά VQS, οι διαφορετικοί εικονικές αναπαραστάσεις έχουν την ίδια σημασία. Παρακάτω, υπογραμμίζουμε τους διάφορους τύπους υβριδικών αναπαραστάσεων που υιοθετούνται σε VQS μέχρι σήμερα, δηλαδή αναπαράστασης χρησιμοποιώντας:

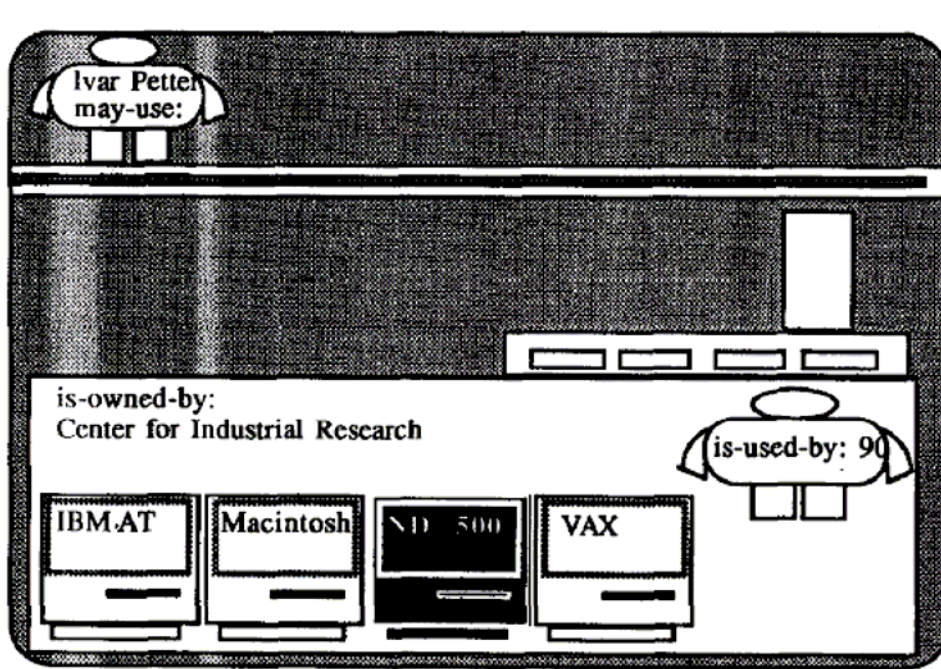
- μορφές και διαγράμματα
- διαγράμματα και εικόνες
- μορφές, διαγράμματα, και εικόνες

Στην πρώτη περίπτωση, τα διαγράμματα χρησιμοποιούνται γενικά για να περιγράψουν το σχεδιάγραμμα της Βάσης Δεδομένων. Οι forms υιοθετούνται κυρίως είτε για να δείξουν πιο λεπτομερείς πληροφορίες, όπως παραδείγματος χάριν ιδιότητες μιας οντότητας, τις περιπτώσεις τους, κ.λ.π., είτε για να σχηματίσουν τα Queries. Μια ιδιαίτερη χρήση των Forms και των διαγραμμάτων προτείνεται στο SKI.

Ένα ενδεικτικό παράδειγμα παρουσιάζεται στην Εικόνα 16. Όπως φαίνεται στο σχήμα, η οθόνη χωρίζεται σε ένα μεταβλητό αριθμό οριζόντιων γραμμών. Παραδείγματος χάριν, η πρώτη γραμμή αναπαριστά το πρώτο τύπο. Η επόμενη λωρίδα αντιπροσωπεύει τις ιδιότητες των πρώτων τύπων που μπορούν να παρουσιαστούν, τροποποιηθούν, κ.λ.π.... Ο χρήστης μπορεί να κάνει scroll πάνω η κάτω για να δει όλες τις λωρίδες. Οποιοδήποτε λωρίδα μπορεί να καταρρεύσει από την οθόνη. Η συγκεκριμένη γραφική αναπαράσταση εξηγεί πώς οι δύο τύποι *προμηθευτών* και *πελατών* (που έχουν επιλεγθεί νωρίτερα από το χρήστη) συσχετίζονται. Το τετράγωνο στο τέλος κάθε λωρίδας δείχνει ότι ορισμένα σημεία έχουν πέσει έξω από τη λωρίδα. Χρησιμοποιώντας ένα βέλος, ο χρήστης μπορεί να κάνει scroll για να δει το τέλος της λωρίδας και να βρει τα σημεία που θέλει.



Εικόνα 16: Υβριδική προσέγγιση στο SKI



Εικόνα 17: Παράδειγμα SICON

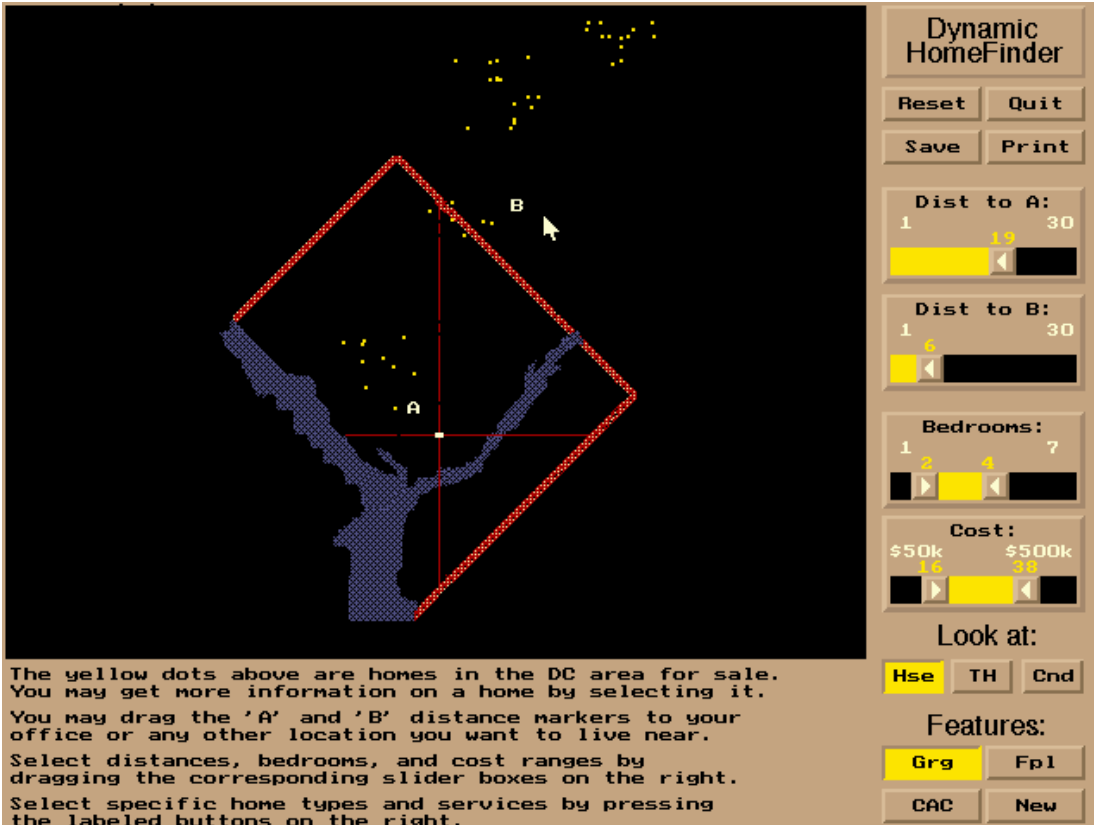
Στα VQS που χρησιμοποιούν διαγράμματα και εικόνες, τα διαγράμματα περιγράφουν πάλι το σχεδιάγραμμα της Βάσης Δεδομένων, ενώ οι εικόνες χρησιμοποιούνται είτε για να αναπαραστήσουν συγκεκριμένα πρωτότυπα αντικείμενα (π.χ. μια γενική περίπτωση ενός προσώπου) είτε για να δείξουν τις ενέργειες που εκτελούνται.



Παραδείγματος χάριν, το SICON (Εικόνα 17), αναπαριστά ένα σχήμα χρησιμοποιώντας και τα διαγράμματα και τις εικόνες. Πράγματι, ο χρήστης μπορεί να απεικονίσει το σχήμα στη διαγραμματική αναπαράσταση, να διαλέξει ένα τύπο οντοτήτων και να εκφράσει ένα σύνθετο ερώτημα ή μπορεί απλά να σχηματίσει το ερώτημα επιλέγοντας τις κατάλληλες εικόνες. Και στις δύο περιπτώσεις το αποτέλεσμα του Query παρουσιάζεται με τις εικόνες.

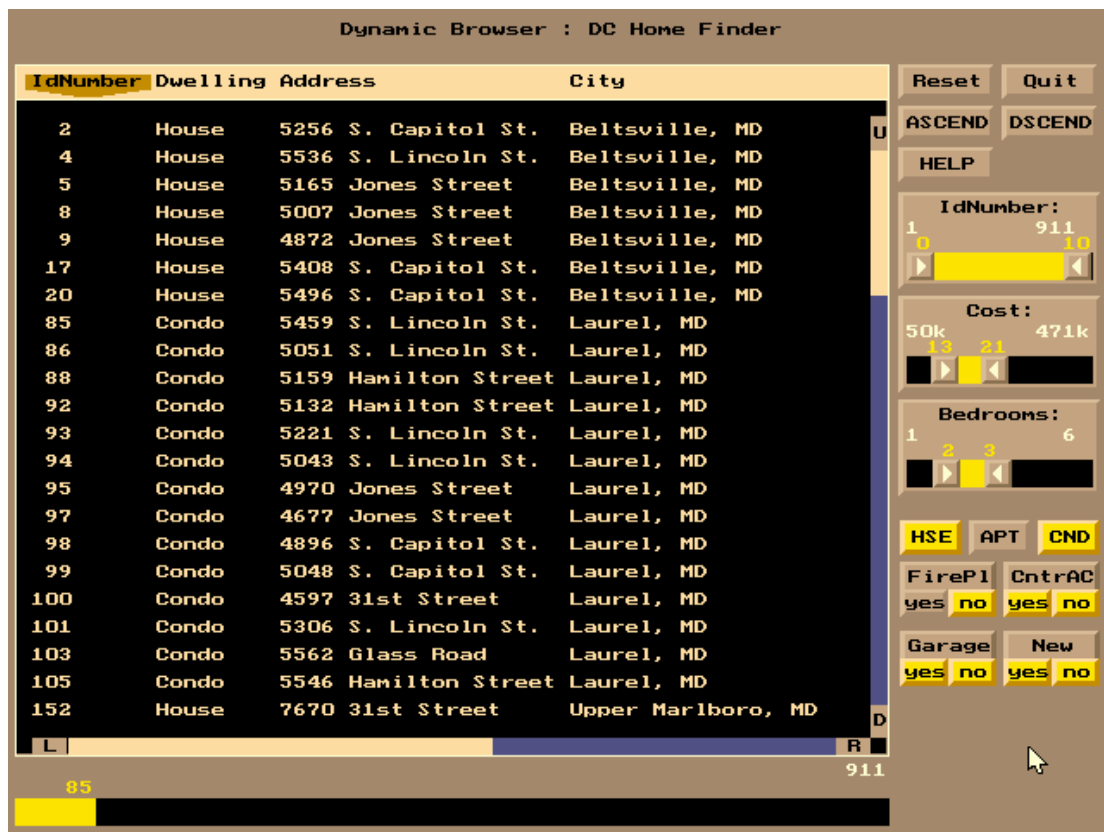
### Δυναμικοί μηχανισμοί επερώτησης

Μια καινοτόμα προσέγγιση που ενοποιούσε τόσο τον μηχανισμό υποβολής ερωτήσεων όσο και την αναπαράσταση δεδομένων υπήρξε αυτή των δυναμικών ερωτήσεων (dynamic queries). Η τεχνική αυτή εφαρμόστηκε με ιδιαίτερη επιτυχία σε σειρά εφαρμογών καταγράφοντας σημαντική βελτίωση στην αποτελεσματικότητα και στην υποκειμενική ικανοποίηση των χρηστών. Η τεχνική βασίζεται στον προσδιορισμό χαρακτηριστικών παραμετροποιημένων ερωτημάτων με γραφικό τρόπο και την άμεση ενημέρωση μιας γραφικής προσομοίωσης των αποτελεσμάτων. Ένα ενδεικτικό παράδειγμα εφαρμογής της τεχνικής παρουσιάζεται στην Εικόνα 18 όπου ο χρήστης διερευνά βάση κριτηρίων εναλλακτικές λύσεις εξεύρεσης σπιτιού. Το panel που βρίσκεται αριστερά επιτρέπει στο χρήστη να προσδιορίσει τα κριτήρια, ενώ η οπτική αναπαράσταση αριστερά ενημερώνεται αυτόματα με τα αποτελέσματα που ικανοποιούν το συγκεκριμένο ερώτημα. Ας σημειωθεί στο σημείο αυτό ότι το σύστημα υποστηρίζει και αναπαράσταση των αποτελεσμάτων υπό μορφή κειμένου (βλέπε Εικόνα 19). Ωστόσο η επιλογή αυτή δεν μπορεί να συγκριθεί με το μηχανισμό αλληλεπίδρασης που παρουσιάζεται στην Εικόνα 18.



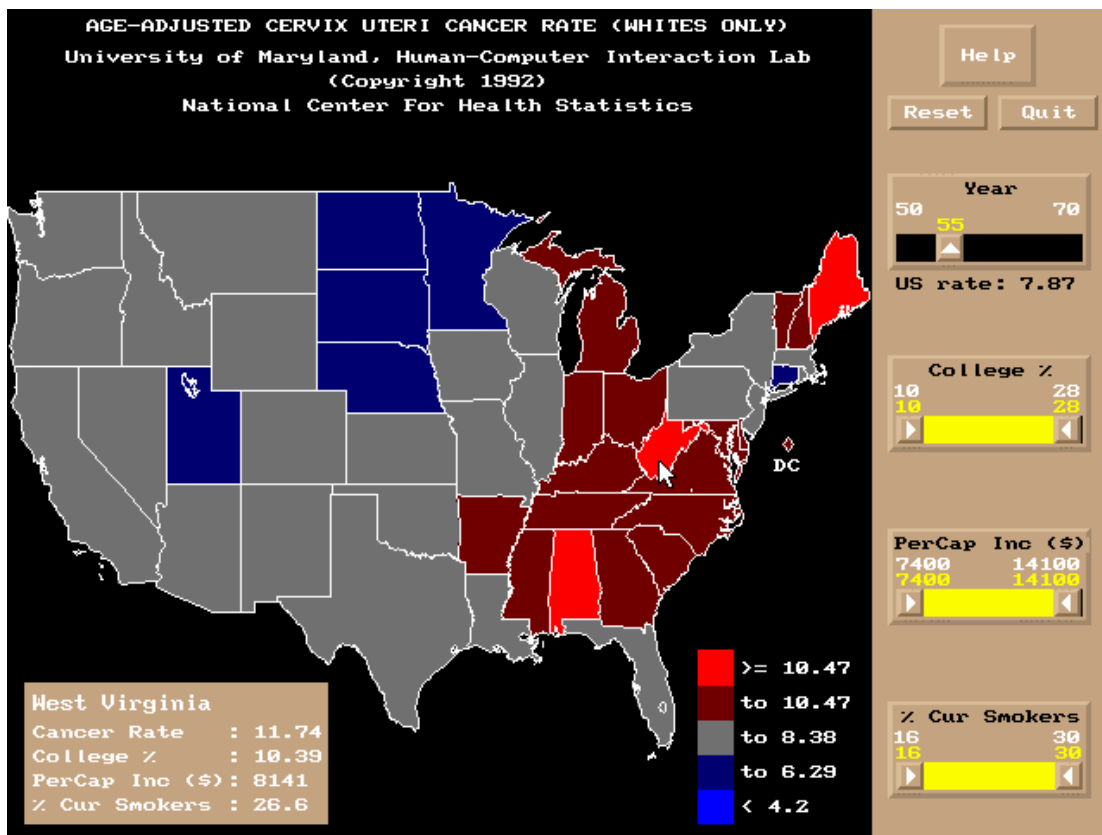
The yellow dots above are homes in the DC area for sale. You may get more information on a home by selecting it. You may drag the 'A' and 'B' distance markers to your office or any other location you want to live near. Select distances, bedrooms, and cost ranges by dragging the corresponding slider boxes on the right. Select specific home types and services by pressing the labeled buttons on the right.

Εικόνα 18: Παράδειγμα εφαρμογής της τεχνικής ‘dynamic querying’

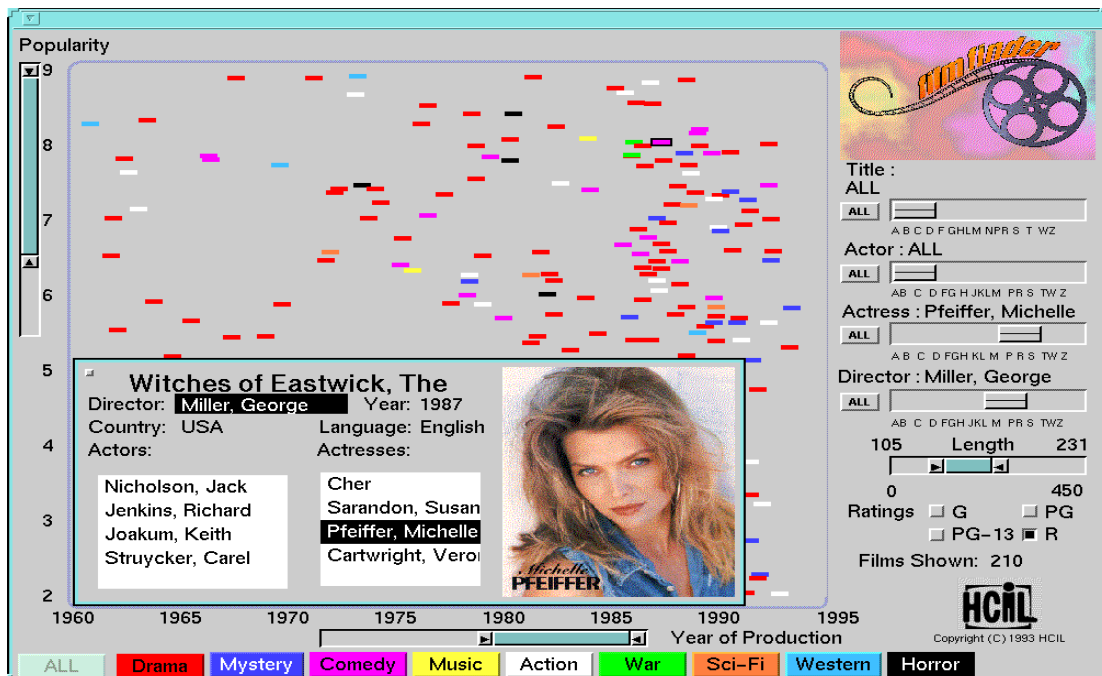


Εικόνα 19: Αναπαράσταση αποτελεσμάτων του HomeFinder υπό μορφή κειμένου

Η τεχνική των δυναμικών ερωτήσεων εξειδικεύτηκε και υποστήριξε σειρά εφαρμογών με ιδιαίτερη επιτυχία. Ενδεικτικά, αναφέρονται η έκδοση της τεχνικής για δυναμικούς χάρτες (dynamaps) όπως φαίνεται στην Εικόνα 20, καθώς και η έκδοση της τεχνικής για εντοπισμό κινηματογραφικών ταινιών (βλέπε Εικόνα 21).



Εικόνα 20: Dynamaps



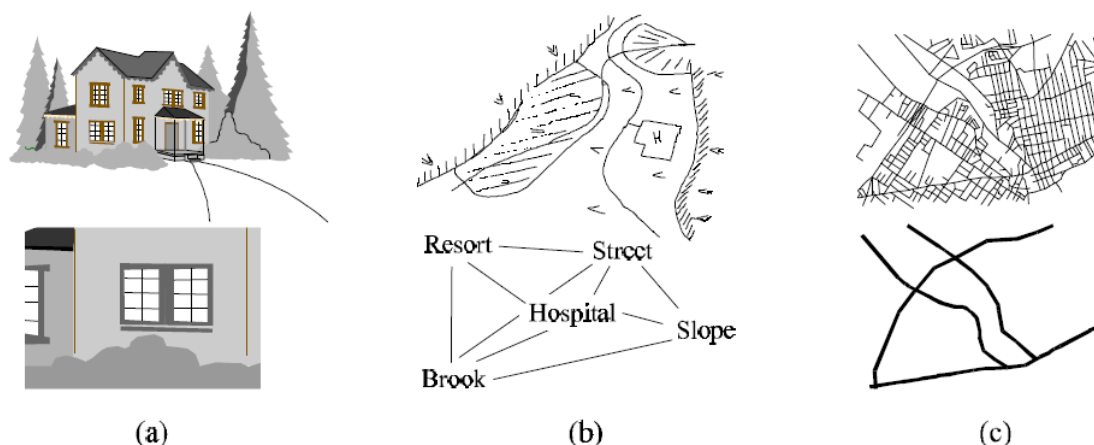
Εικόνα 21: Filmfinder

Η μελέτη των συστημάτων αυτών κατέδειξε σειρά επιθυμητών πλεονεκτημάτων της βασικής τεχνικής των δυναμικών ερωτήσεων που είχαν ως αποτέλεσμα της καταξίωση της μεθόδου και την ευρύτερη εφαρμογή της. Μερικά από τα πλεονεκτήματα αυτά συνοψίζονται ως εξής. Πρώτον, η τεχνική υποστηρίζει την

οπτική αναπαράσταση των τμημάτων μιας ερώτησης γεγονός που δεν απαιτεί από τον χρήστη τη λεπτομερή γνώση μιας γλώσσας υποβολής ερωτημάτων όπως η SQL. Δεύτερο, τα αποτελέσματα της εκτέλεσης μιας ερώτησης έχουν γραφική αναπαράσταση που επιτρέπει γρήγορη και αποδοτικότερη ερμηνεία των αποτελεσμάτων. Τρίτο πλεονέκτημα είναι η δυνατότητα υποβολής ενός ερωτήματος με αυξητικό τρόπο που ο χρήστης μπορεί να βλέπει τα αποτελέσματα των ερωτημάτων του όπως κατασκευάζεται η ερώτηση και ανάλογα να προσθέτει, να αφαιρεί ή να προσδιορίζει κριτήρια. Τέλος ένα άλλο θετικό χαρακτηριστικό της μεθόδου είναι η δυνατότητα γρήγορης ανάκλησης μιας δράσης που επιτρέπει το χρήστη να αναιρέσει κάτι το οποίο δεν επέφερε το επιθυμητό αποτέλεσμα και να οδηγηθεί σε ένα προγενέστερο αποδεκτό σημείο αντί της εκ' νέου διατύπωσης του ερωτήματος.

### 2.4.3 Μηχανισμοί θέασης γραφικής αναπαράστασης

Η ανάπτυξη γραφικών τεχνικών διαχείρισης των δεδομένων μιας βάσης δεδομένων επέτρεψε την επαύξηση των κλασικών τύπων ερωτημάτων. Συγκεκριμένα, έγινε δυνατή η υποστήριξη νέων τελεστών που αφορούν αποκλειστικά την παρουσίαση των γραφικών δεδομένων, επιτρέποντας έτσι στους χρήστες να μπορούν να μεταβάλλουν την όψη ή τον τρόπο που βλέπουν τα πράγματα ανάλογα με τις εργασίες που έχουν να εκτελέσουν. Η αλλαγή τρόπου θέασης είναι ένας βασικός τομέας στην αναπαράσταση των δεδομένων σε κάθε εφαρμογή. Για παράδειγμα αν κάποιος έχει μια διπλωμένη εφημερίδα θα μπορεί να διαβάσει την επικεφαλίδα αλλά όχι και τα περιεχόμενά της. Για να γίνει αυτό θα πρέπει να τη ξεδιπλώσει. Το ίδιο συμβαίνει και στις διεπαφές χρήστη-υπολογιστή όπου ο χειριστής μπορεί να χρειαστεί να αλλάξει το τρόπο θέασης ανάλογα με την κατάσταση. Υπάρχουν τρεις βασικοί τομείς στην αλλαγή τρόπου θέασης που αφορούν την αλλαγή προοπτικής, την αλλαγή θεματικού επιπέδου και την αλλαγή επιπέδου λεπτομέρειας (βλέπε Εικόνα 22 για παραδείγματα κάθε κατηγορίας).



Εικόνα 22: Οι τρεις βασικές κατηγορίες αλλαγής τρόπου θέασης (a) αλλαγή προοπτικής (b) αλλαγή θεματικού επιπέδου (c) αλλαγή επιπέδου λεπτομέρειας

Η αλλαγή προοπτικής περιλαμβάνει δραστηριότητες όπως περιστροφή, αλλαγή σημείου επιθεώρησης, παρατήρηση από μακριά κ.α. Η αλλαγή προοπτικής δε θα πρέπει να μπερδεύεται με την αλλαγή της γεωμετρίας ή των χωρικών ρυθμίσεων των αντικειμένων. Η αλλαγή προοπτικής μπορεί να εφαρμοστεί και σε κείμενο και σε γραφικά. Σε ένα δισδιάστατο περιβάλλον, όπως μια εικόνα, μπορεί να περιλαμβάνει

δραστηριότητες όπως η περιστροφή, το ζουμάρισμα κ.α. Σε ένα τρισδιάστατο περιβάλλον ο χρήστης θα μπορεί πραγματικά να πετάει μέσα στο χώρο.

Η αλλαγή θεματικού επιπέδου επιτρέπει την επισκόπηση των ίδιων δεδομένων μέσω διαφορετικών οπτικών φακών ή φίλτρων. Μία επισκόπηση ή μια διαφορετική θέαση για παράδειγμα μπορούν να θεωρηθούν θεματικά επίπεδα. Η αλλαγή θέασης είναι απαραίτητη μόνο όταν τα επιλεγμένα στοιχεία έχουν ενδιαφέρον. Παραδείγματος χάρη όταν η πυκνότητα των δεδομένων είναι πολύ μεγάλη τότε ενδεχομένως μια αλλαγή του θεματικού επιπέδου να επιτρέπει την διάκριση κάποιων δεδομένων ιδιαίτερου ενδιαφέροντος.

Τέλος, η αλλαγή επιπέδου λεπτομέρειας αφορά την δυνατότητα εστίασης (δηλαδή ζουμ) μέσα ή έξω στην πληροφορία. Το αν θα εξαφανιστεί ή θα εμφανιστεί ένα αντικείμενο δεν εξαρτάται μόνο απ' την αλλαγή του γεωμετρικού του μεγέθους. Την απόφαση για το ποιες ιδιότητες του αντικειμένου θα φανούν και ποιες θα αποκρυφτούν εξαρτάται απ' το περιεχόμενο και τη λειτουργία του. Ο σκοπός είναι να κρατηθεί η συνοχή των δεδομένων άθικτη. Αφού η αλλαγή λεπτομέρειας είναι μια ενέργεια που γίνεται συχνά, θα πρέπει να εκτελείται αυτόματα απ' το σύστημα. Αν η αναπαράσταση είναι αρκετά πολύπλοκη τότε η αλλαγή αυτή μπορεί να χρειαστεί να γίνει χειρωνακτικά.

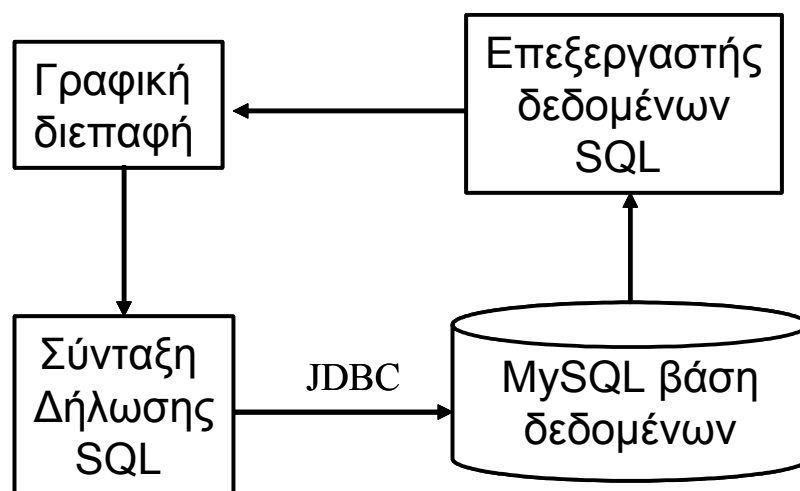
## 3 Το σύστημα e-tour

---

Στο κεφάλαιο αυτό παρουσιάζουμε συνοπτικά την αρχιτεκτονική δομή του συστήματος και τη σχεδίαση επιλεγμένων τμημάτων κορμού, όπως η βάση δεδομένων και η διεπαφή χρήστη – υπολογιστή.

### 3.1 Αρχιτεκτονική συστήματος

Η βασική αρχιτεκτονική του συστήματος διαχωρίζει μεταξύ της διεπαφής που υποστηρίζει ένα γραφικό στυλ αλληλεπίδρασης και του διαχειριστικού περιβάλλοντος της MySQL που αναλαμβάνει να εκτελέσει ερωτήματα και να συλλέξει δεδομένα από τους πίνακες της βάσης δεδομένων (βλέπε επόμενο τμήμα). Στην Εικόνα 23 παρουσιάζεται συνοπτικά η αρχιτεκτονική με σαφή διαχωρισμό του συστήματος διαχείρισης της βάσης δεδομένων και των λοιπών τμημάτων που υλοποιήθηκαν στα πλαίσια της παρούσας πτυχιακής. Ο συντάκτης δηλώσεων SQL είναι το τμήμα εκείνο το οποίο αναλαμβάνει τη διερμηνεία γραφικών καθηκόντων και την μετάφραση αυτών σε σημασιολογικά αντίστοιχες δηλώσεις SQL. Οι παράγωγες δηλώσεις διαβιβάζονται και εκτελούνται από το σχεσιακό περιβάλλον διαχείρισης της MySQL. Το τμήμα της επεξεργασίας των δεδομένων αφορά στην υλοποίηση των γραφικών τεχνικών ενημέρωσης της διεπαφής (π.χ. γραφική απεικόνιση επιλεγμένης τοποθεσίας, δραστηριοτήτων που φιλοξενούνται κλπ.). Επίσης, το τμήμα αυτό υλοποιεί και τα περιγράμματα παρουσίασης δεδομένων που παρουσιάζονται συνοπτικά παρακάτω.

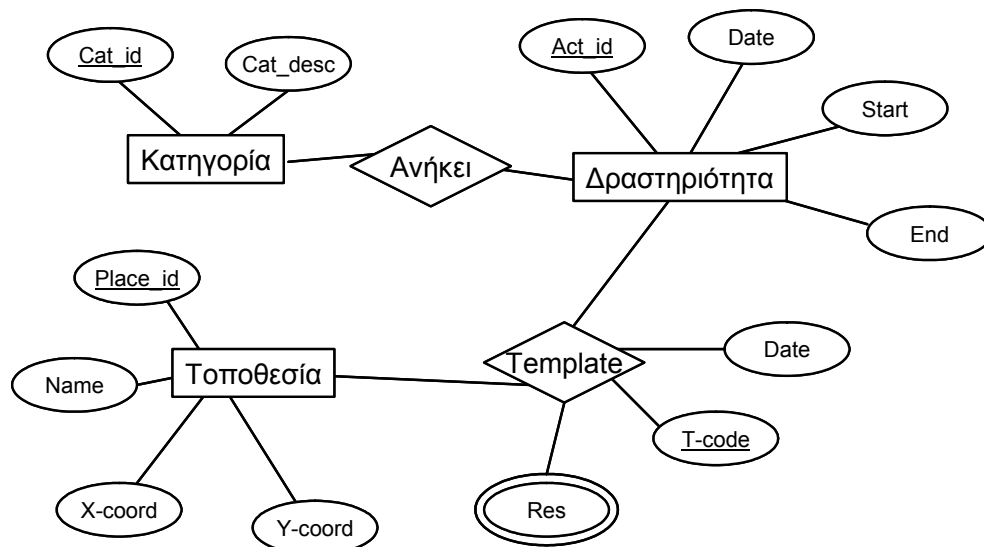


Εικόνα 23: Αρχιτεκτονική δομή συστήματος

### 3.2 Σχεδίαση βάσης δεδομένων

Η σχεδίαση της βάσης δεδομένων που χρησιμοποιεί το σύστημα πρέπει να υποστηρίζει καταρχήν αρχειοθέτηση δεδομένων διαφορετικών τύπων (π.χ. κείμενο, εικόνα, ήχο) καθώς και τα περιγράμματα που αξιοποιούνται από την διεπαφή για την

παρουσίαση των αποτελεσμάτων των ερωτήσεων. Το γεγονός αυτό παρουσιάζεται και στο μοντέλο Οντοτήτων-Συσχετίσεων στην Εικόνα 24.



Εικόνα 24: Μοντέλο Ο-Σ

Παρατηρούμε ότι οι βασικές οντότητες της βάσης δεδομένων είναι οι ακόλουθες:

- Κατηγορία που δηλώνει τύπο δραστηριότητας όπως παραδείγματος χάριν συναυλία, θεατρική παράσταση, κλπ
- Δραστηριότητα που δηλώνει μια συγκεκριμένη εκδήλωση που πραγματοποιείται μια ημερομηνία με σαφή έναρξη και τέλος
- Τοποθεσία που δηλώνει φυσικούς χώρους που φιλοξενούν δραστηριότητες

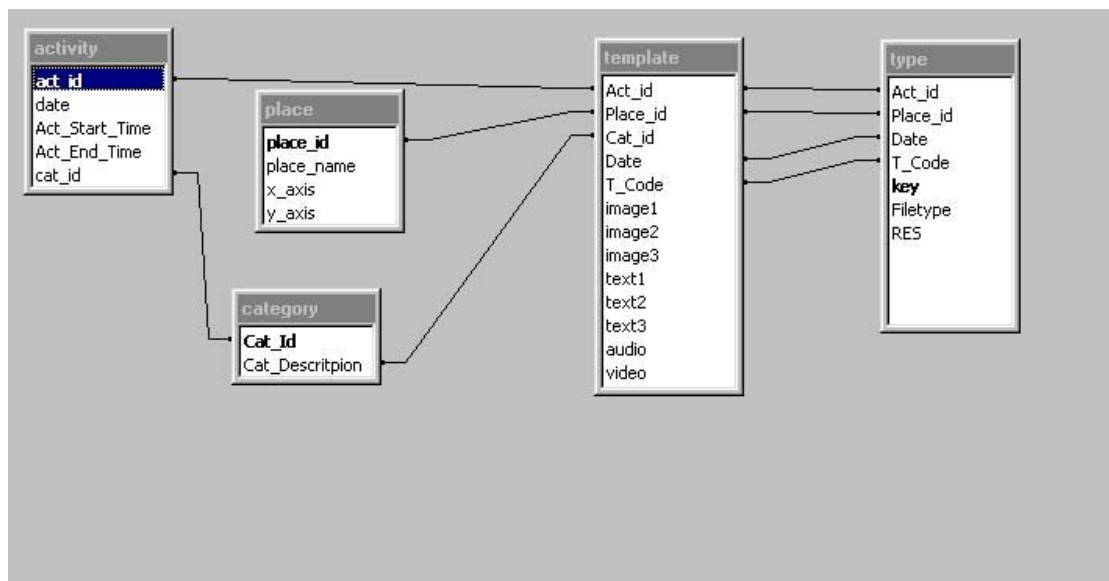
Δύο συσχετίσεις προσδιορίζουν την κατηγορία στην οποία ανήκει μια συγκεκριμένη δραστηριότητα και το περίγραμμα παρουσίασης της δραστηριότητας. Για λόγους απλούστευσης η πρώτη συσχέτιση απαλείφεται και το πρωτεύον κλειδί της οντότητας κατηγορία μεταφέρεται ως ξένο κλειδί στη οντότητα δραστηριότητα.

Το πρόγραμμα μας καλείται να καλύψει σενάρια παρουσίασης που λαμβάνουν χώρα σε συγκεκριμένες τοποθεσίες στο Ηράκλειο. Για να το επιτύχουμε αυτό πρέπει να γνωρίζουμε βασικά στοιχεία όπως την κατηγορία της δραστηριότητας (ΘΕΑΤΡΟ, ΚΙΝΗΜΑΤΟΓΡΑΦΟΣ, ΣΥΝΑΥΛΙΑ κ.λ.π), τον τόπο στον οποίο γίνεται καθώς και περαιτέρω πληροφορίες που αφορούν την ίδια την δραστηριότητα (ημερομηνία, ώρα, συντελεστές, κ.α). Μια βάση δεδομένων που ικανοποιεί τα παραπάνω παρουσιάζεται στην Εικόνα 25.

Όνομα Πίνακα	Πεδία Πίνακα
<b>Activity</b>	Act_id , Date , Act_Start_Time , Act_End_Time , Cat_id
<b>Category</b>	Cat_id , Cat_Description
<b>Place</b>	Place_id , Place_Name , X_axis , Y_axis
<b>Template</b>	Act_id , Cat_id , Place_id , Date , T_Code , image1, image2 , image3 , text1, text2 , text3 , audio , video
<b>Type</b>	Act_id , Place_id , T_Code, Date , key , Filetype , RES

Εικόνα 25: Σχεσιακό μοντέλο Βάσης Δεδομένων

Το σχήμα και οι σχέσεις τις βάσης καθορίζονται στην Εικόνα 26, ενώ τα στιγμιότυπα της βάσης παρουσιάζονται στην εικόνες Εικόνα 27 έως Εικόνα 31.



Εικόνα 26: Μοντέλο της βάσης



SQLyog - Free MySQL GUI - [New Connection - root@localhost]

her\_fest\_prog

1

1 Result 2 Messages 3 Table Data 4 Objects 5 History

Show All or Limit 0 50 Refresh

Act_id	Date	Act_StartTime	Act_EndTime	Cat_id
1	2006-08-06	21:00:00	(NULL)	3
2	2006-07-28	22:00:00	(NULL)	3
3	2006-08-30	20:00:00	22:00:00	2
4	2006-07-30	20:00:00	22:00:00	2
5	2006-08-14	20:00:00	22:00:00	1
6	2006-08-14	20:00:00	22:00:00	1
7	2006-08-30	20:00:00	20:00:00	2
8	2006-07-05	19:00:00	22:00:00	2
*	(NULL)	(NULL)	(NULL)	(NULL)

Ready 0 ms 8 row(s) Ln 1, Col 1 Connections : 1

Εικόνα 27: Στιγμιότυπο του πίνακα ‘Δραστηριότητα’

SQLyog - Free MySQL GUI - [New Connection - root@localhost]

her\_fest\_prog

1

1 Result 2 Messages 3 Table Data 4 Objects 5 History

Show All or Limit 0 50 Refresh

Cat_id	Cat_Description
1	CINEMA
2	THEATER
3	CONCERT
*	(NULL)

Ready 0 ms 3 row(s) Ln 1, Col 1 Connections : 1

Εικόνα 28: Στιγμιότυπο του πίνακα ‘κατηγορία’

The screenshot shows the SQLyog interface with the 'place' table selected in the left-hand tree. The main window displays the following data:

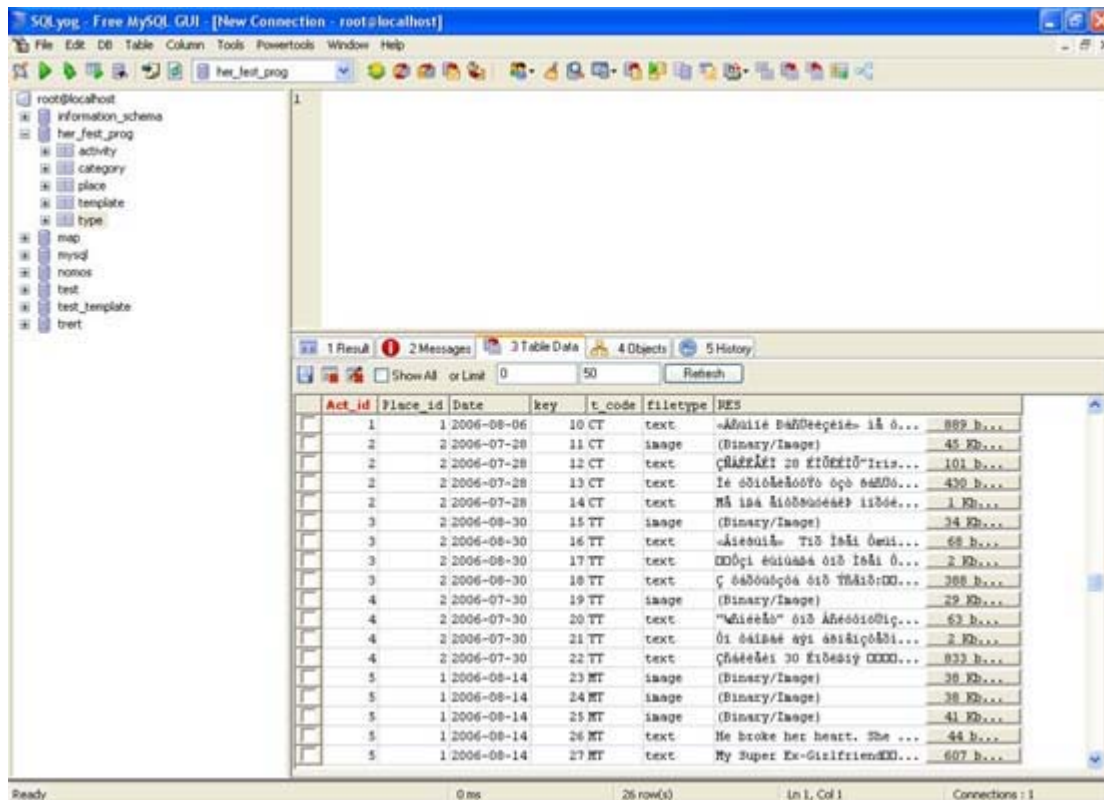
Place_id	Place_name	X_axis	Y_axis
1	ASTORIA	7 b...	610 330
2	ΣΦΡΟΤΗΕΑΤΡΟ ΚΑΖΑΝΤΣΑΡΔΙ	22 b...	510 550
3	ΦΡΟΝΑΜΟΝΑΣ ΠΑΝΟΚΡΑΤΟΡΑ	23 b...	230 490
*	(NULL) (NULL)	0 b...	(NULL) (NULL)

Εικόνα 29: Στιγμιότυπο του πίνακα ‘τοποθεσία’

The screenshot shows the SQLyog interface with the 'template' table selected in the left-hand tree. The main window displays the following data:

Act_id	Place_id	cat_id	Date	T_code	image1	image2	image3	text1	text2	text3	audio	video
1	1	3	2006-08-06	CT	1	1	1	1	1	1	1	0
2	2	3	2006-07-29	CT	1	0	0	1	1	1	0	0
3	2	2	2006-08-30	TT	1	0	0	1	1	1	0	0
4	2	2	2006-07-30	TT	1	0	0	1	1	1	0	0
5	1	1	2006-08-14	MT	1	1	1	1	0	1	0	0
6	3	1	2006-08-14	MT	0	0	0	1	0	0	0	0
7	3	2	2006-08-30	TT	1	1	1	1	1	1	0	0
8	2	2	2006-07-05	TT	1	0	0	1	(NULL)	1	0	(NULL)
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Εικόνα 30: Στιγμιότυπο του πίνακα ‘Template’



Εικόνα 31: Στιγμιότυπο του πίνακα 'Type'

### 3.3 Σχεδίαση Διεπαφής

Η εφαρμογή που αναπτύχθηκε στη παρούσα πτυχιακή εργασία έχει ως στόχο, όπως ήδη έχουμε αναφέρει να επιτρέψει σε ένα περιηγητή της πόλης του Ηρακλείου να διερευνήσει πολιτιστικές εκδηλώσεις που πραγματοποιούνται σε διαφορετικά σημεία της πόλης. Για να ικανοποιηθεί ο σκοπός αυτός πρέπει να δημιουργηθεί μια διεπαφή που θα επιτρέψει στο χρήστη να υποβάλει τις ερωτήσεις προς την βάση δεδομένων με τρόπο απλό και εύχρηστο αποκρύπτοντας του πολύπλοκους μηχανισμούς μιας κλασικής γλώσσας υποβολής ερωτήσεων όπως είναι η SQL και επιτρέποντας του να αποφύγει περιττές ενέργειες. Τέλος τα αποτελέσματα πρέπει να παρουσιάζονται με τρόπο ευδιάκριτο και σαφή.

Η διεπαφή που αναπτύχθηκε για να ικανοποιήσει τις παραπάνω απαιτήσεις ακολουθεί τη σχεδιαστική λογική της τεχνικής των δυναμικών ερωτήσεων που συνατήσαμε στο προηγούμενο κεφάλαιο. Έτσι, η διεπαφή αποτελείται από ένα κεντρικό τμήμα που περιέχει τον χάρτη και ένα πλαίσιο (jPanel), όπου εκεί γίνεται από τον χρήστη η επιλογή του σεναρίου χρήσης και ο προσδιορισμός των κριτηρίων μιας ερώτησης. Έχουμε, τέλος, τοποθετήσει μια ετικέτα (JLabel) στο κάτω δεξιά τμήμα της διεπαφής που αναφέρονται στοιχεία για την εκάστοτε τοποθεσία. Ένα σκαρίφημα της διεπαφής ερωτήσεων παρουσιάζεται στην Εικόνα 33 όπου καταγράφονται τα βασικά μέρη της διεπαφής ενώ, ένα πρωτότυπο υψηλής πιστότητας του τελικού συστήματος εμφανίζεται στην Εικόνα 33.

<h1>Χάρτης &amp; Ενεργα Στοιχεία Προγράμματος</h1>	<p>Τμήμα Επιλογής Σεναρίων</p>
	<p>Τμήμα Επιλογής Δεδομένων</p>
	<p>Ετικέτα Παρουσίασης Στοιχείων Τοποθεσίας</p>

Εικόνα 32: Σκαρίφημα διεπαφής



### **Εικόνα 33: Πρωτότυπο υψηλής πιστότητας**

Συγκεκριμένα, η διεπαφή του προγράμματος αποτελείται από 4 ενεργές περιοχές που συνοψίζονται ως εξής:

- το Τμήμα Επιλογής Σεναρίων
- το Τμήμα Επιλογής Δεδομένων ,
- την Ετικέτα Παρουσίασης Στοιχείων Τοποθεσίας
- το Χάρτη

Το Τμήμα Επιλογής Σεναρίων είναι το κομμάτι που καθορίζει τις λειτουργίες εκείνες που θα εκτελεστούν από τον χρήστη. Το Τμήμα Επιλογής Δεδομένων παρουσιάζει στον χρήστη (ανάλογα και με το σενάριο) όλα τα διαθέσιμα δεδομένα έτσι ώστε να επιλέξει δραστηριότητα με βάση τα δικά του κριτήρια. Το τμήμα του Χάρτη είναι το σημείο που ενεργοποιείτε και αλλάζει ανάλογα με τις προτιμήσεις του χρήστη. Τέλος Ετικέτα Παρουσίασης Στοιχείων Τοποθεσίας παρουσιάζει γενικότερα στοιχεία (τηλέφωνο, διεύθυνση κ.λ.π.) για τις περιοχές εκείνες που λαμβάνουν χώρα οι δραστηριότητες. Αξίζει να αναφέρουμε ότι σε ένα από τα σενάρια που παρουσιάζονται αργότερα, και αφορούν την αναζήτηση δραστηριότητας μέσω τοποθεσίας οι ρόλοι του τμήματος του χάρτη και του τμήματος επιλογής δεδομένων αλλάζουν γιατί ο χρήστης καλείτε να επιλέξει από όλα τα ενεργά στοιχεία του χάρτη εκείνα που επιθυμεί και έπειτα οι διαθέσιμες δραστηριότητες εμφανίζονται στο τμήμα επιλογής δεδομένων.

Όταν ο χρήστης επιλέξει την δραστηριότητα που εκείνος επιθυμεί , αναλόγως τον τύπο της , τα δεδομένα εμφανίζονται σε μία φόρμα παρουσίασης. Τα σκαριφήματα αυτών των φορμών παρουσιάζονται παρακάτω (βλέπε Εικόνα 34, Εικόνα 35 και Εικόνα 36).

ΤΙΤΛΟΣ		
Στοιχεία Συναυλίας		
Image 1	Image 2	Image 3
Γενική Περιγραφή Καλιτεχνών		
Audio Sample	Video Sample	

Εικόνα 34: Φόρμα παρουσίασης Συναυλιών

ΤΙΤΛΟΣ ΤΑΙΝΙΑΣ		
Image 1	ΣΤΟΙΧΕΙΑ ΤΑΙΝΙΑΣ	
ΠΕΡΙΓΡΑΦΗ ΤΑΙΝΙΑΣ		
Image 2	Video Sample	Image 3

Εικόνα 35: Φόρμα παρουσίασης κινηματογραφικής ταινίας

<b>ΤΙΤΛΟΣ</b>		
<b>Image 1</b>	<b>Στοιχεία Συντελεστών παραστασης</b>	
<b>Περιγραφή Παραστασης</b>		
<b>image 2</b>	<input type="checkbox"/> video Sample <input type="checkbox"/> audio Sample	<b>image 3</b>

Εικόνα 36: Φόρμα Παρουσίασης Θεατρικής Παράστασης

Οι παραπάνω διεπαφές έχουν σκοπό μόνο να παρουσιάσουν τα στοιχεία της δραστηριότητας που επέλεξε ο χρήστης και ο χρήστης μπορεί να αλληλεπιδράσει μαζί τους μόνο στην περίπτωση που υπάρξει οπτικό-ακουστικό υλικό( audio ή video sample ).

### 3.4 Υλοποίηση συστήματος

Για την υλοποίηση της εφαρμογής χρησιμοποιήσαμε 3 εργαλεία την MySQL ,τα NetBeans και το SQLyog.

#### 3.4.1 MySQL

Η MySQL είναι ένα σύστημα διαχείρισης βάσεων δεδομένων πολλών χρηστών, το οποίο η MySQL AB καθιστά διαθέσιμο σαν ελεύθερο λογισμικό κάτω από την άδεια ευρέως κοινού GNU. Επιλέξαμε για την δημιουργία και την διαχείριση της βάσης δεδομένων την MySQL γιατί προσφέρει αρκετά πλεονεκτήματα όπως :

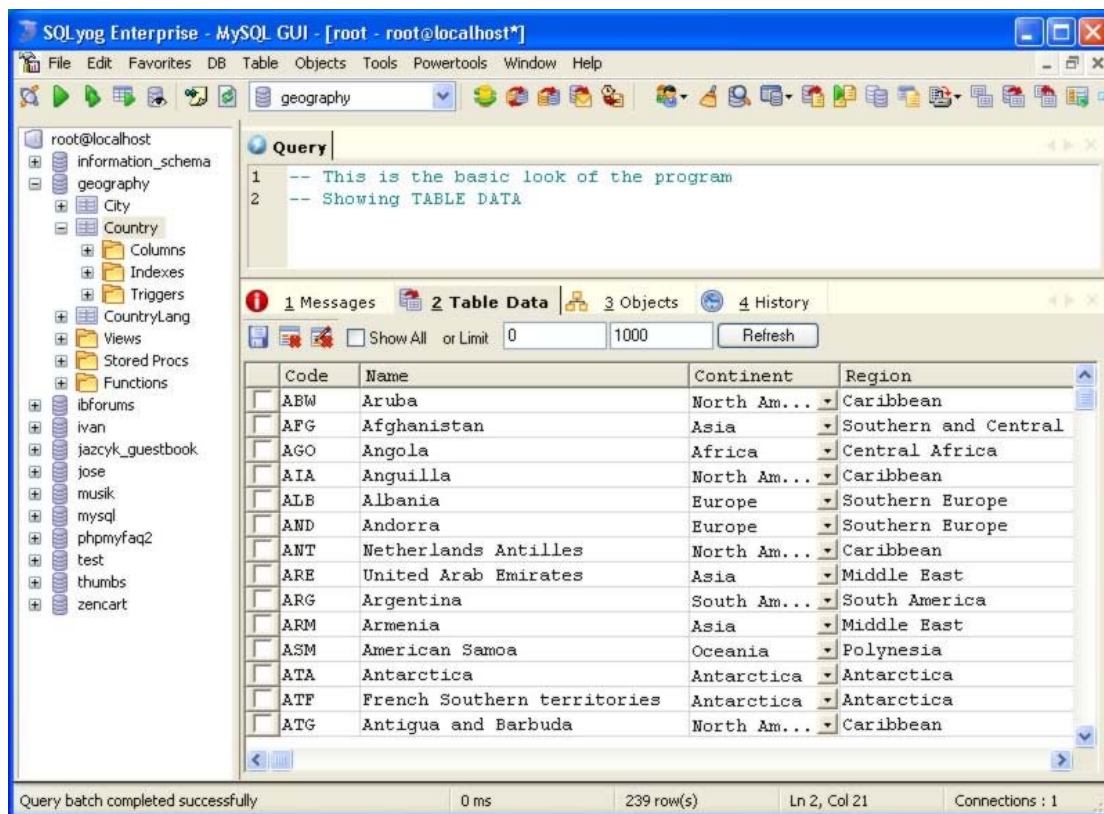
- Ευκολία χρήσης: Ενώ βασικές γνώσεις SQL απαιτούνται (όπως και στα περισσότερα DBMS ) δεν χρειάζεται παραπάνω από μερικές δηλώσεις SQL για να αλληλεπιδράσει κάποιος με την βάση.
- Ασφάλεια: Η MySQL περιλαμβάνει στερεά στρώματα ασφάλειας στοιχείων που προστατεύουν τα ευαίσθητα στοιχεία από εισβολείς . Τα δικαιώματα μπορούν να ρυθμιστούν ,έτσι ώστε να επιτρέπουν μερικά ή όλα τα προνόμια είτε στα άτομα ,είτε σε ομάδες. Οι κωδικοί πρόσβασης κωδικοποιούνται.

- Ταχύτητα: Για χάρη της ταχύτητας, οι σχεδιαστές της MySQL αποφάσισαν να προσφέρουν λιγότερα χαρακτηριστικά γνωρίσματα από άλλους ανταγωνιστές, όπως η Sybase και η Oracle , χωρίς να παραβλέπει τα βασικά γνωρίσματα που απαιτούνται από τους περισσότερους Database administrators.
- Δυνατότητα χειρισμού μεγάλου όγκου δεδομένων: Η MySQL μπορεί να χειριστεί σχεδόν οποιοδήποτε αριθμό στοιχείων, μέχρι και 50 εκατομμύρια σειρές και πάνω. Το βασικό όριο μεγέθους αρχείου είναι 4 GB. Παρ'όλα αυτά, μπορεί αυτό να αυξηθεί σε ένα θεωρητικό όριο των 8 TB δεδομένων .
- Βέλτιστη χρήση μνήμης: Ένας MySQL server έχει εξεταστεί λεπτομερώς για να αποτρέψει τις διαρροές μνήμης.
- Δυνατότητα εγκατάστασης σε πολλά λειτουργικά συστήματα: Linux , τα περισσότερα Unix και τα Windows επιτρέπουν την εγκατάσταση της MySQL.
- Υποστήριξη πολλών διεπαφών ανάπτυξης : JDBC , ODBC , Perl , PHP κ.α. επιτρέπουν την δημιουργία πληθώρας εφαρμογών για τις πιο σημαντικές πλατφόρμες .
- Πολλαπλές μηχανές αποθήκευσης: MyISAM , InnoDB , DBD , Merge και άλλες, δίνοντας την δυνατότητα επιλογής της αποτελεσματικότερης για τον κάθε πίνακα.

### 3.4.2 NetBeans

Τα NetBeans είναι ένα ενσωματωμένο περιβάλλον ανάπτυξης για το γράψιμο , την σύνταξη , τη δοκιμή και το debugging δικτυακών και μη εφαρμογών της πλατφόρμας Java. Περιλαμβάνει έναν πλήρη συντάκτη κειμένου με έλεγχο λαθών και σύνταξης , οπτικά εργαλεία σχεδίασης και άλλα γνωρίσματα.





Εικόνα 37: SQLyog main data tab

### 3.4.3 SQLyog

Το SQLyog είναι ένα γραφικό περιβάλλον για ένα MySQL Database Server. Το πρόγραμμα στοχεύει σε εξοικειωμένους, με δηλώσεις SQL, χρήστες που αναζητούν μια εύχρηστη γραφική διεπαφή για να εκτελούν τις ερωτήσεις τους. Το SQLyog επιτρέπει την εκτέλεση πολλαπλών, επιλεγμένων (highlighted) ή απλών ερωτήσεων και προσφέρει έναν template explorer για να μπορεί ο χρήστης να εισάγει γρήγορα συχνά χρησιμοποιημένες ερωτήσεις όπως, CREATE TABLE, INSERT, DROP TABLE και άλλες. Επιπλέον, επιτρέπει την εξαγωγή των δεδομένων σε CSV, HTML και XML format και εισαγωγή από αρχείο κειμένου. Τέλος, το SQLyog επιτρέπει την διάγνωση και βελτιστοποίηση των βάσεων δεδομένων καθώς και την αποθήκευση των προτιμώμενων SQL scripts και άλλα. Ένα ενδεικτικό στιγμιότυπο του εργαλείου παρουσιάζεται στην Εικόνα 37.

## 4 Σενάρια Λειτουργίας

---

Τα σενάρια που θα παρουσιάσουμε ενδεικτικά σε αυτήν την ενότητα είναι τα εξής η απλή αναζήτηση δραστηριότητας μέσω ημερομηνίας, η πολλαπλή αναζήτηση δραστηριότητας μέσω ημερομηνίας και η αναζήτηση δραστηριότητας μέσω τοποθεσίας. Σε κάθε περίπτωση θα παρουσιάσουμε τόσο τον τρόπο υποβολής ερωτήσεων μέσω γραφικών καθηκόντων όσο και τη μετατροπή των ερωτημάτων σε καλά δομημένες SQL ερωτήσεις και εκτέλεση των SQL ερωτημάτων.

Προτού ξεκινήσουμε να περιγράφουμε τα βασικά σενάρια ανάκτησης πληροφορίας θα παραθέσουμε τον κώδικα που χρησιμοποιεί το σύστημα μας για να επικοινωνήσει με την βάση δεδομένων έτσι ώστε να πάρει τα αποτελέσματα που θέλουμε:

```
try
{
    Class.forName("com.mysql.jdbc.Driver");
    Connection databaseConnection=DriverManager.getConnection
("jdbc:mysql://localhost/her_fest_prog","username","password");
    Statement statement = databaseConnection.createStatement();
    String sql=" Το εκάστοτε query(SELECT FROM WHERE etc)";
    ResultSet result = statement.executeQuery(sql);
    ΤοΌνομαΤηςΜεθόδου(result);
    databaseConnection.close();
}
catch(Exception e)
{
    System.out.println(e);
}
```

Όπου η εκάστοτε μέθοδος είναι της μορφής:

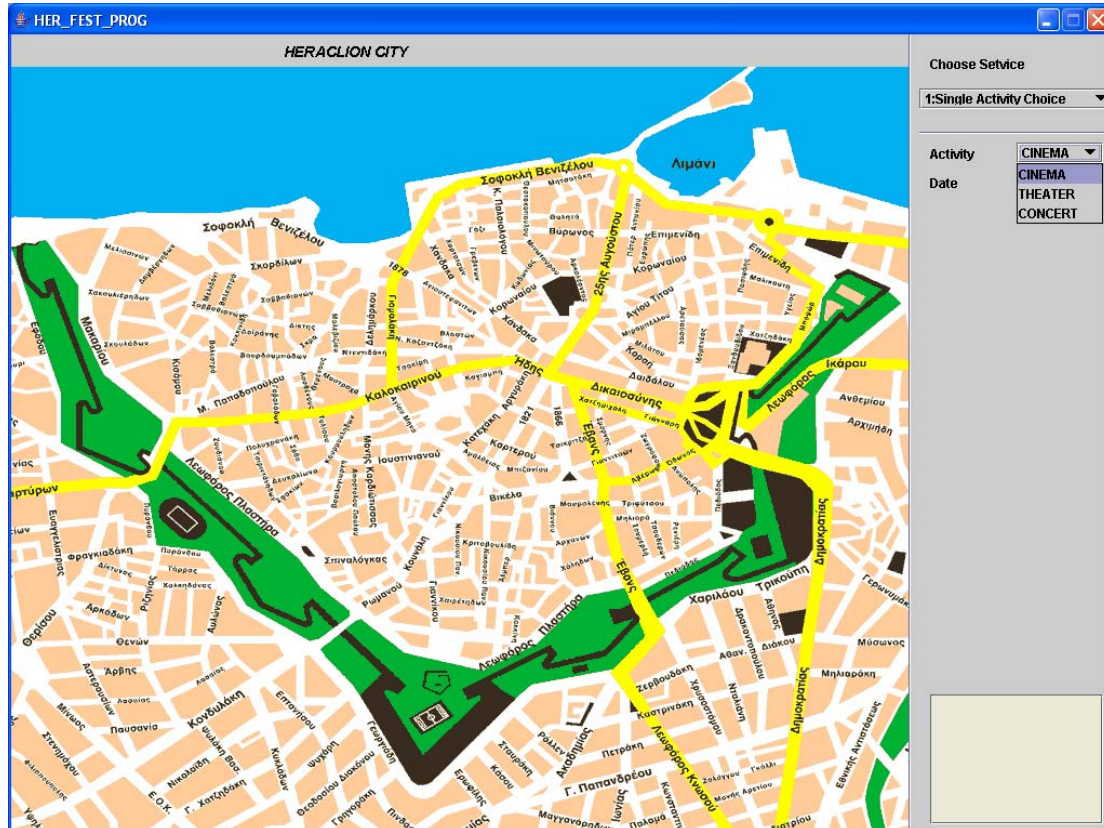
```
public void ΤοΌνομαΤηςΜεθόδου(ResultSet r) throws SQLException{
    ResultSetMetaData rmeta = r.getMetaData();
    int numColumns = rmeta.getColumnCount();
    while (r.next())
    {
        for (int i=1;i<=numColumns;i++)
        {
            r.getInt(); ή r.getString(); αναλόγως τα δεδομένα που
            θέλουμε να πάρουμε
        }
    }
}
```

### 4.1 Απλή Αναζήτηση δραστηριότητας μέσω Ημερομηνίας

Κατά το άνοιγμα της φόρμας εκτελείται το παρακάτω query έτσι ώστε να γεμίσει το πρώτο JComboBox με τις διαθέσιμες δραστηριότητες.

```
SELECT CAT_ID , CAT_DESCRIPTION FROM CATEGORY
```

Χρησιμοποιώντας τον κώδικα που παραθέσαμε στην αρχή της ενότητας παράγονται δυο πίνακες που περιέχουν ο ένας τις κατηγορίες δραστηριοτήτων και ο άλλος τα ID των δραστηριοτήτων και εισάγουμε τον πίνακα με τις κατηγορίες στο πρώτο jComboBox (βλέπε Εικόνα 38).



**Εικόνα 38: Ο χάρτης του Ηρακλείου**

Έπειτα, επιλέγουμε την δραστηριότητα που θέλουμε χρησιμοποιώντας το combobox, για παράδειγμα THEATER και δημιουργείται η ακόλουθη ερώτηση σε SQL

```
SELECT ACT_ID, DATE
FROM ACTIVITY
WHERE ACTIVITY.CAT_ID=cat_id[index]
```

Όπου cat\_id[] ο πίνακας που περιέχει τα Ids των κατηγοριών και η μεταβλητή index το index του πρώτου jComboBox όπου cat\_id[index]=2 επειδή το cat\_id για THEATER είναι 2.

Τις τιμές που θα πάρουμε από το query τις τοποθετούμε σε δυο πίνακες και έπειτα καλούμε την συνάρτηση:

```
CreateFinalArays(String[] tempdates , int[] tempid)
```

Η CreateFinalArays παίρνει τους δυο πίνακες , ελέγχει τον πίνακα με τις ημερομηνίες για διπλές τιμές και φτιάχνει αντίστοιχα και τον πίνακα με τα Ids, καταχωρώντας τα τελικά αποτελέσματα σε δυο Vectors, για παράδειγμα εάν για τα θέατρα τα αποτελέσματα που θα παίρναμε ήταν της μορφής:

TempID
3
4
7
8

Tempdates
2006-08-30
2006-07-30
2006-08-30
2006-07-05

Θα τους μετέτρεπε στους εξής δυο Vectors:

FinalDates
2006-08-14
2006-07-20
2006-07-05

FinalID
3,7
4
8

Αναλυτικότερα, οι συναρτήσεις που κάνουν τον διαχωρισμό είναι οι:

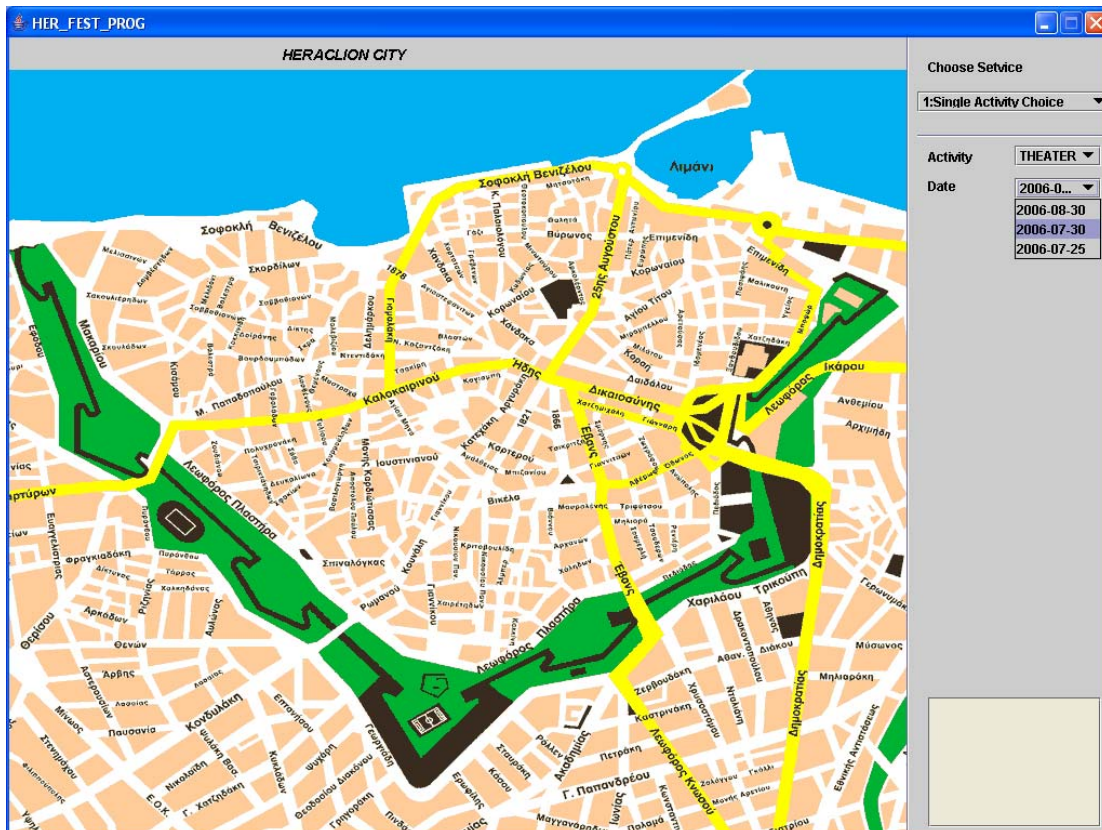
```
public void createFinalArrays(String[] tempdates,int[] tempid)
{
    FinalDates=new Vector();
    FinalID=new Vector();
    for(int i=0;i<tempdates.length;i++)
    {
        Vector v=new Vector();
        String temp=new String();
        boolean flag=false;
        v=WhereExists(tempdates, tempdates[i]);
        if(v.size(>1)
        {
            for(int j=0;j<v.size();j++)
            {
                int place=Integer.parseInt(v.get(j).toString());
                temp+=tempid[place] + ",";
                if(!flag)
                {
                    flag=!flag;
                }
                else
                {
                    tempdates[place]="";
                }
            }
        }
        else
        {
            temp="" +tempid[i];
        }
        if(!tempdates[i].equals(""))
        {
            FinalDates.add(tempdates[i]);
            FinalID.add(temp);
        }
    }
}
private static Vector WhereExists(String[] body, String key)
{
```

```

Vector places=new Vector();
for(int i=0;i<body.length;i++)
{
    if(!key.equals("") && key.equals(body[i]))
    {
        places.add(new Integer(i));
    }
}
return places;
}
}

```

Τέλος, τοποθετούμε το vector FinalDates στο δεύτερο JComboBox όπως φαίνεται στην Εικόνα 39.



**Εικόνα 39: Ενημερωμένα πεδία της διεπαφής**

Αφού επιλέξουμε ημερομηνία, εάν μόνο μία δραστηριότητα λαμβάνει χώρα τη συγκεκριμένη χρονική στιγμή (το γνωρίζουμε ελέγχοντας το μήκος του Vector σε εκείνη τη θέση) , για παράδειγμα στις 30/7/2006 εκτελούμε τα αντίστοιχα queries:

```

SELECT PLACE
FROM TEMPLATE
WHERE TEMPLATE.PLACE_id=
Integer.parseInt(FinalID.elementAt(index).toString())

```

Οπου `Integer.parseInt(FinalID.elementAt(index).toString())` είναι η τιμή του `act_id` για την συγκεκριμένη ημερομηνία και είναι ίσο με 4. Το οποίο καταχωρεί την τιμή του `Place_id` στην μεταβλητή `CurPlace`. Εδώ το `CurPlace` παίρνει την τιμή 2.

Και μετά το:

```

SELECT PLACE.X_AXIS, PLACE.Y_AXIS

```

```
FROM PLACE
WHERE PLACE_ID=CurPlace
```

Εφόσον, έχουμε τις συντεταγμένες των τοποθεσιών, χρησιμοποιούμε μια δομή επιλογής (case structure) που ενεργοποιεί τα αντίστοιχα σημεία στον χάρτη και αποθηκεύουμε τις τιμές CurPlace, το Act\_id της δραστηριότητας και την ημερομηνία για να τις χρησιμοποιήσουμε στις φόρμες παρουσίασης που θα δούμε αργότερα (βλέπε Εικόνα 40).



Εικόνα 40: Ενημέρωση χάρτη

Στην περίπτωση που παραπάνω από μια δραστηριότητες περιλαμβάνονται στο ημερήσιο πρόγραμμα, όπως π.χ. για την ημερομηνία 2006/08/30, καλούμε την μέθοδο **getEventID\_ByDate(int index)**, η οποία θα χωρίσει τα Ids και θα τα καταχωρίσει σε ένα πίνακα. Χρησιμοποιώντας τις τιμές του προηγούμενου παραδείγματος, στην πρώτη θέση του Vector FinalID θα μετατρέψει την τιμή 3,7 σε ένα πίνακα 2 θέσεων με τιμές 3 και 7. Η μέθοδος **getEventID\_ByDate** περιγράφεται παρακάτω:

```
private static String[] getEventID_ByDate(int index)
{
    String str=FinalID.get(index).toString();
    String[] id=str.split(",");
    return id;
}
```

Έπειτα, χρησιμοποιούμε μια loop, η οποία εκτελεί τα παρακάτω queries τόσες φορές όσες είναι και οι τιμές του πίνακα:

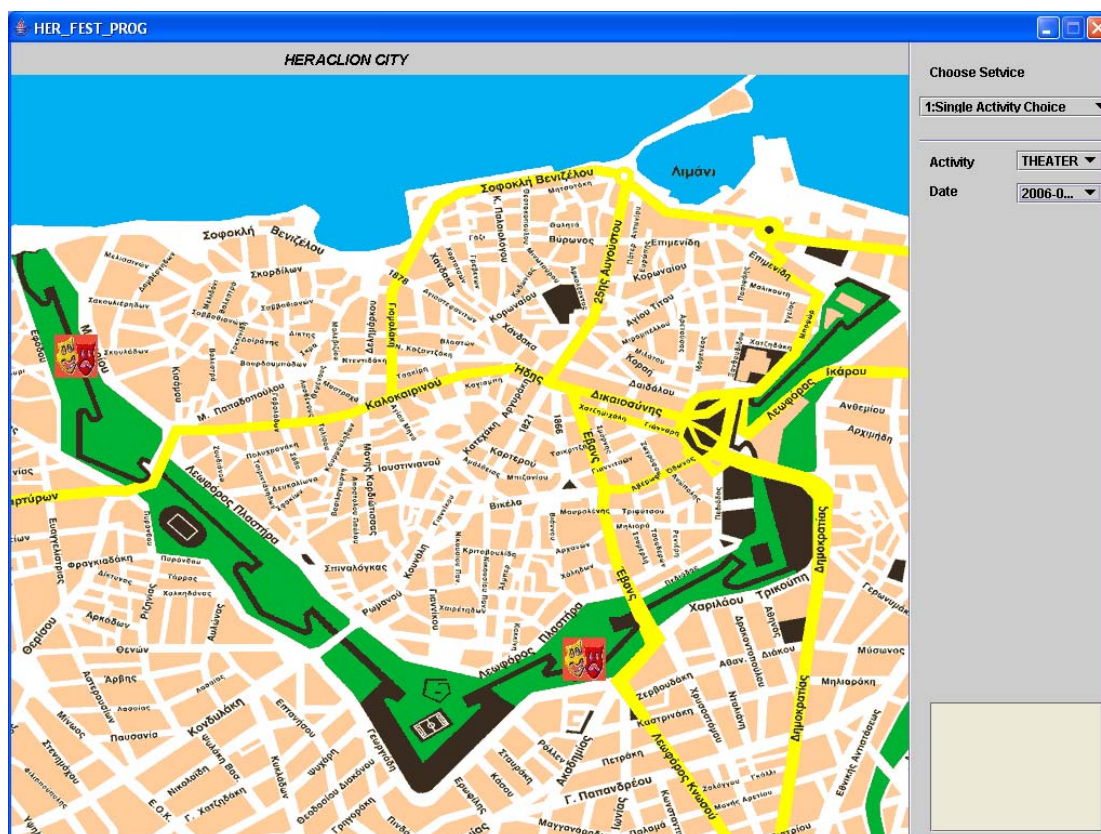
```
SELECT PLACE
FROM TEMPLATE
```

```
WHERE TEMPLATE.PLACE_id= test[qpoint]
```

Όπου test[] ,ο πίνακας με τα (χωρισμένα πλέον) Ids και qpoint ένας pointer που μας υποδεικνύει την σωστή τιμή και test[qpoint] παίρνει διαδοχικά τις τιμές 3 και 7. Μετά την εκτέλεση του πρώτου query το αποτέλεσμα καταχωρείται στην μεταβλητή CurPlace, δηλαδή 2 και 3 ,και εκτελείται το επόμενο query:

```
SELECT PLACE.X_AXIS,PLACE.Y_AXIS  
FROM PLACE  
WHERE PLACE_ID=CurPlace
```

Τέλος, ενεργοποιούμε τα αντίστοιχα σημεία και αποθηκεύουμε την τιμή του CurPlace καθώς και την Ημερομηνία (βλέπε Εικόνα 41).



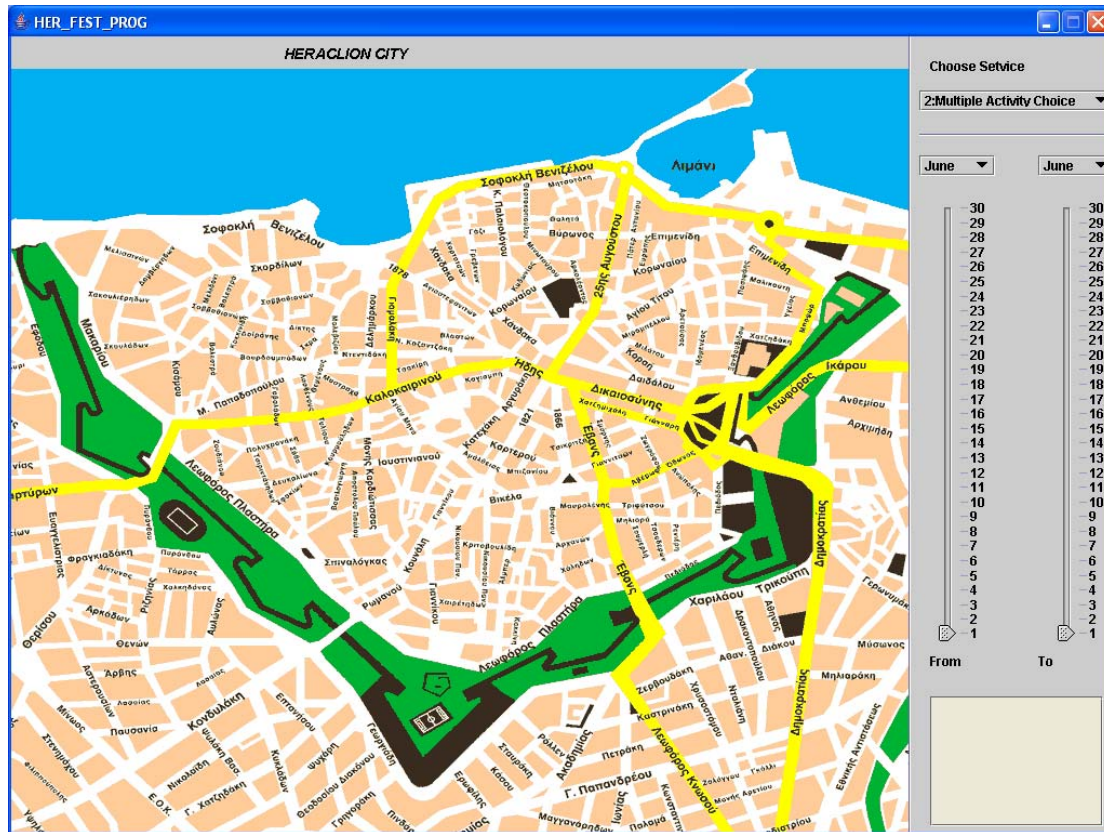
Εικόνα 41: Ενημέρωση χάρτη διεπαφής

## 4.2 Πολλαπλή αναζήτηση δραστηριότητας μέσω Ημερομηνίας

Στο σενάριο αυτό, όπως φαίνεται και στην Εικόνα 42, το δεξί μέρος της οθόνης αποτελείται από δυο jSlider όπου μας βοηθάνε να ορίσουμε τις ημερομηνίες ανάμεσα στις οποίες θα αναζητήσουμε τις δραστηριότητες μας. Για την δήλωση μιας συνθήκης επιλογής αρχικά, μετακινούμε το πρώτο jSlider έτσι ώστε να πάρει τιμή και έπειτα κάθε φορά που αλλάζουμε την κατάσταση του δεύτερου jSlider εκτελείται το query:

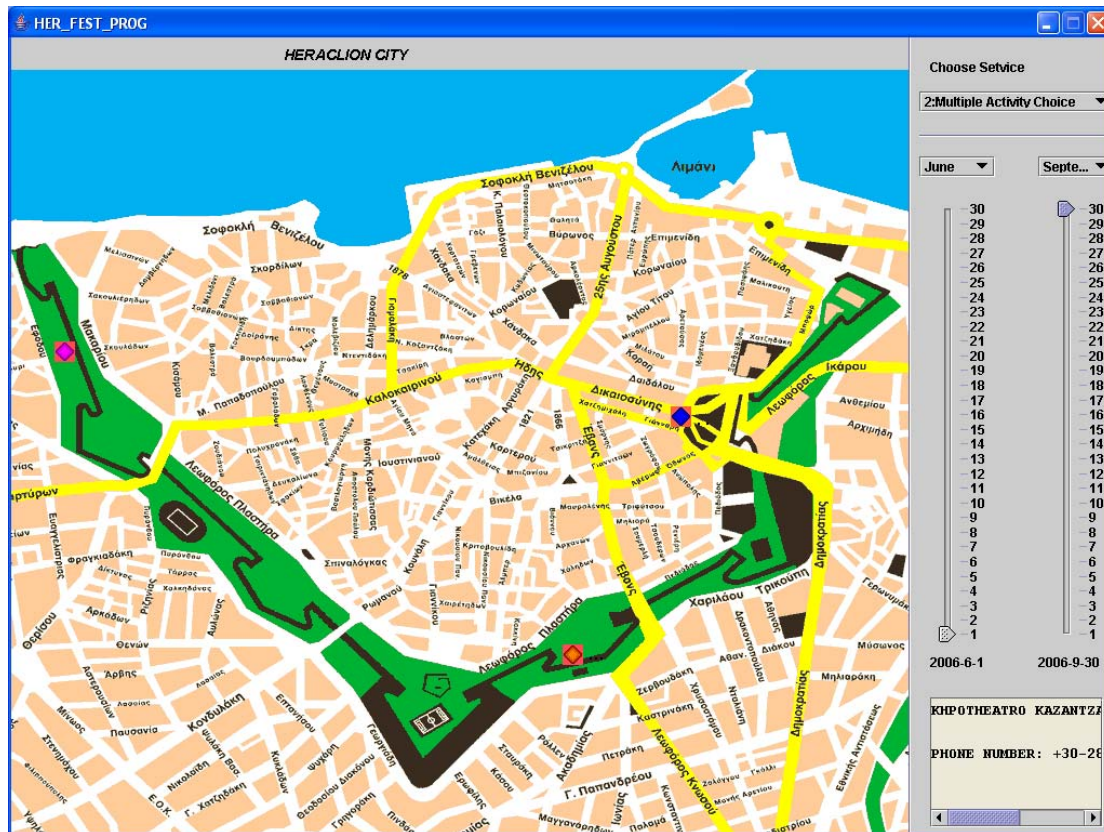
```
SELECT PLACE_ID  
FROM TEPLATE  
WHERE DATE > 'a' AND DATE <'b'
```

Οπού a η τιμή από το πρώτο jSlider και b η τιμή από το δεύτερο jSlider. Στο παράδειγμά μας a=2006-06-01 και b=2006-08-31. Στη συνέχεια ενεργοποιούνται τα σημεία εκείνα τα οποία φιλοξενούν δραστηριότητες στο χρονικό διάστημα που έχουμε ορίσει (βλέπε Εικόνα 43).



Εικόνα 42: Αναζήτηση μέσω ημερομηνία



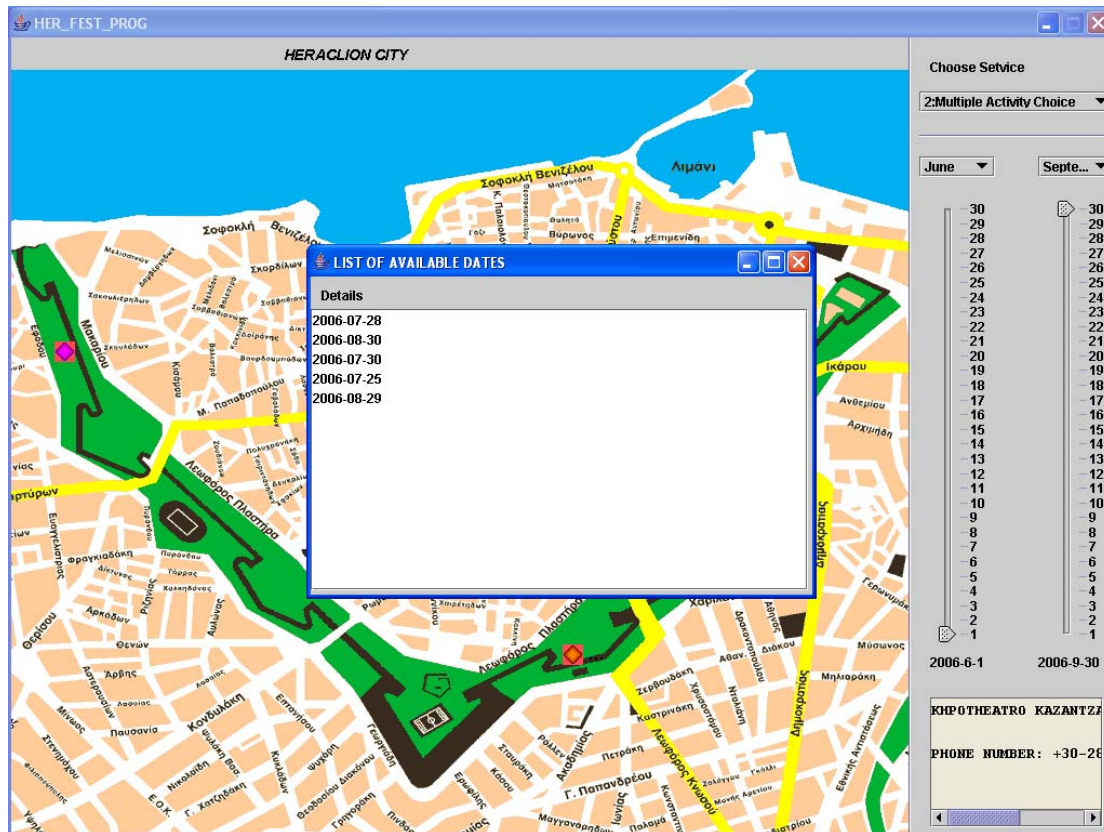


Εικόνα 43: Ένδειξη σημείων με δραστηριότητες σε συγκεκριμένες ημερομηνίες

Εάν τώρα κάνουμε click σε ένα από τα ενεργά σημεία του χάρτη το `place_id`, η αρχική και η τελική ημερομηνία καταχωρούνται στις μεταβλητές `plat`, `datebeg` και `dateend` τις φόρμας `intermediate`, η οποία εκτελεί το παρακάτω query έτσι ώστε να γεμίσει μια `jList` με τα αποτελέσματα (βλέπε Εικόνα 44).

```
SELECT DATE,ACT_ID
FROM TEMPLATE
WHERE TEMPLATE.DATE >'datebeg' AND TEMPLATE.DATE <'dateend'
and PLACE.ID='plat'
```

Οπού `datebeg=2006/06/01`, `dateend=2006/08/31` και `plat=2`.



Εικόνα 44: Επιλογή από λίστα δραστηριοτήτων συγκεκριμένης ημερομηνίας

Τέλος, επιλέγουμε (από το διάλογο στην Εικόνα 44) την ημερομηνία που επιθυμούμε και όπως και στις προηγούμενες περιπτώσεις αποθηκεύονται τα place\_id , Act\_id και η ημερομηνία για να χρησιμοποιηθούν στις φόρμες παρουσίασης.

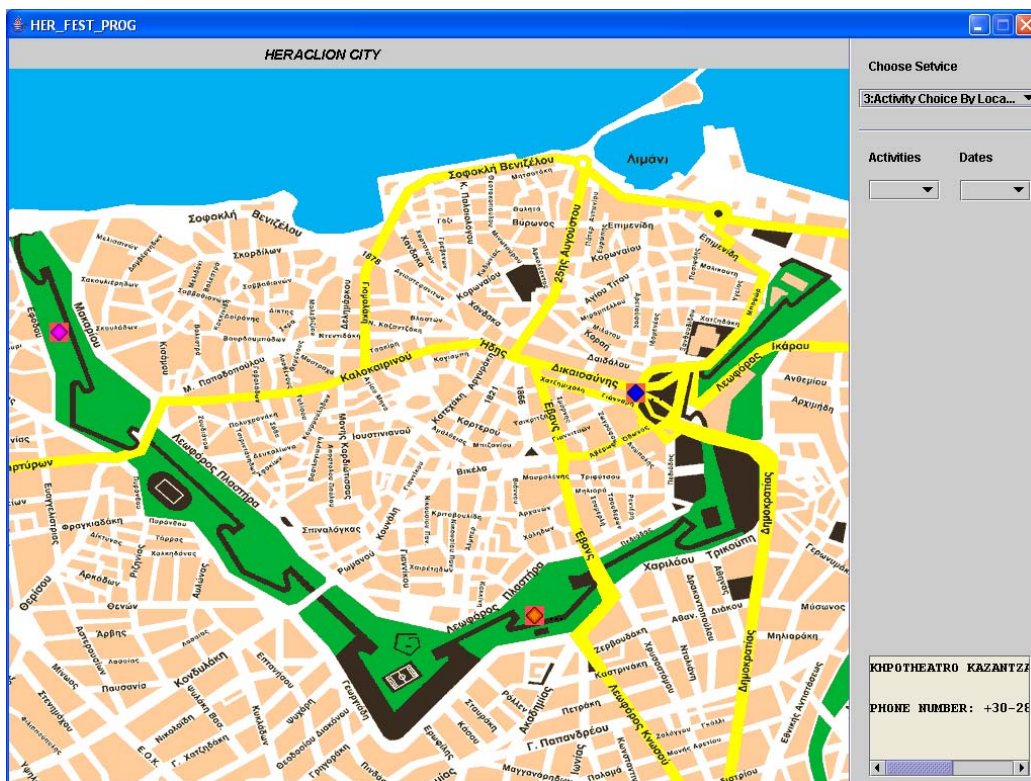
### 4.3 Αναζήτηση δραστηριότητας μέσω τοποθεσίας

Το σενάριο αυτό επιτρέπει στο χρήστη να αναζητήσει δρώμενα που γίνονται σε μια συγκεκριμένη τοποθεσία ή χώρο. Στη περίπτωση αυτή όπως βλέπουμε και στην Εικόνα 45, όλα τα σημεία παρουσιάζονται πάνω στον χάρτη. Κάνοντας click πάνω σε ένα από αυτά γεμίζουμε (βλέπε Εικόνα 46) τα δυο jComboBox στα δεξιά της οθόνης, το ένα με όλες τις κατηγορίες δραστηριοτήτων θέτοντας ως προεπιλογή την τιμή ALL και το άλλο με τις ημερομηνίες όλων των δραστηριοτήτων που γίνονται σε αυτήν την τοποθεσία. Αυτό γίνεται με τα εξής queries:

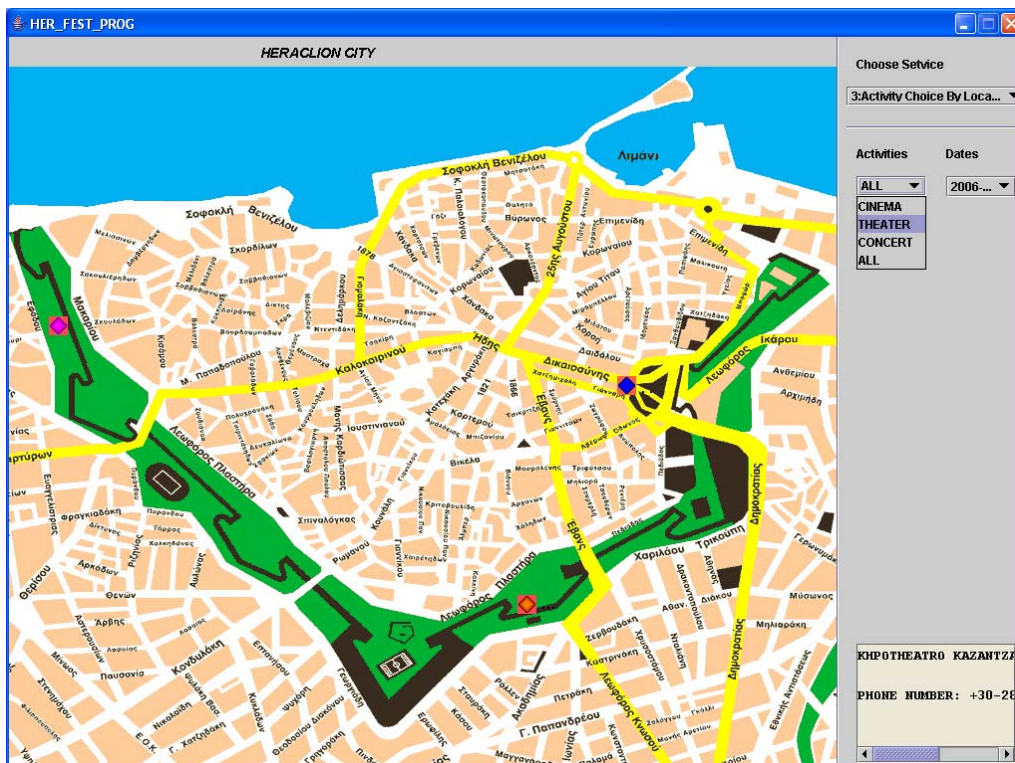
```
SELECT CAT_ID,CAT_DESCRIPTION
FROM CATEGORY
```

```
SELECT DATE
FROM TEMPLATE
WHERE TEMPLATE.PLACE_ID=aid
```

Όπου aid μια μεταβλητή που έχει τιμή το Place\_id του επιλεγμένου σημείου.



Εικόνα 45: Αναζήτηση δρόμων ανά τοποθεσία



Εικόνα 46: Παρουσίαση όλων των δραστηριοτήτων

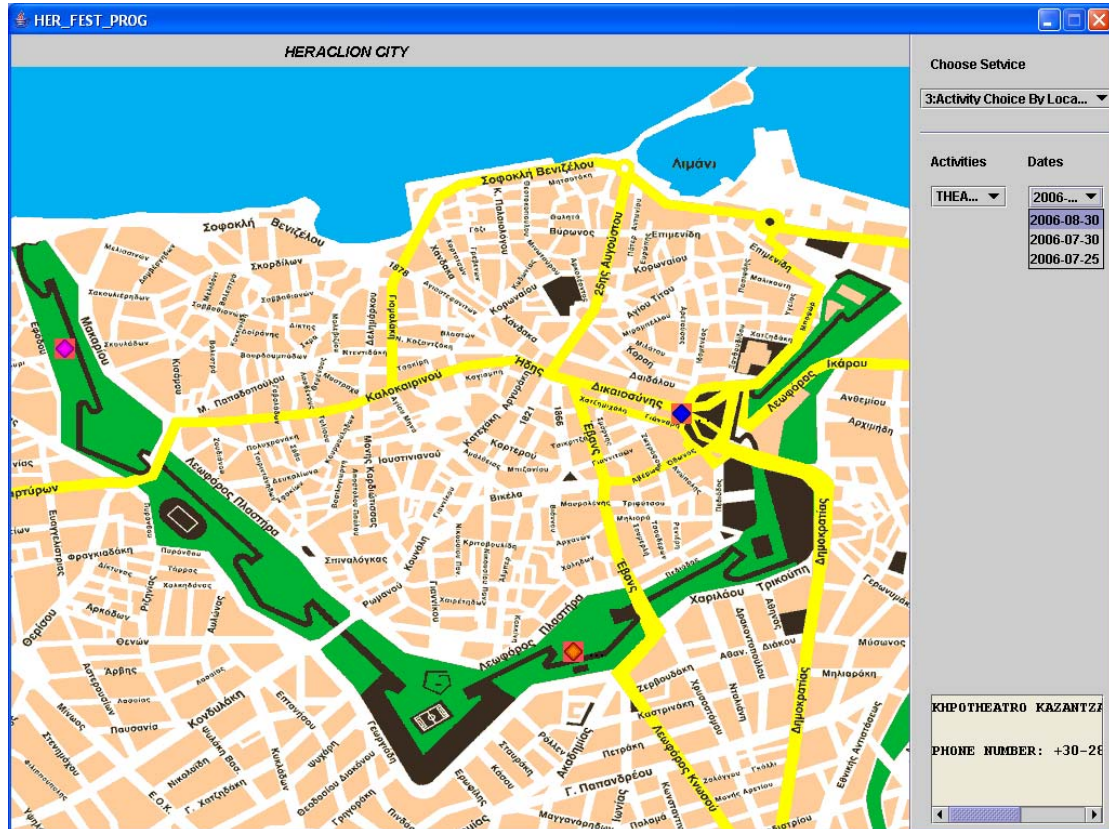
Έπειτα εάν θέλουμε να επιλέξουμε τύπο δραστηριότητας από το πρώτο query, επιλέγοντας για παράδειγμα όλα τα θέατρα που γίνονται σε αυτό το μέρος ,επιλέγουμε το THEATER και εκτελείται το query (βλέπε Εικόνα 47):

```

SELECT A_ID, DATE
FROM TEMPLATE
WHERE TEMPLATE.CAT_ID=index+1
TEMPLATE.PLACE_ID=aid

```

Οπού index+1 η τιμή του cat\_id εδώ 2 και aid η τιμή του place\_id εδώ 2.



Εικόνα 47: Επιλογή κατηγορίας δραστηριότητας

Κάνοντας click στην ημερομηνία που επιθυμούμε αποθηκεύουμε το place\_id , act\_id , και την ημερομηνία για να καλέσουμε μια φόρμα παρουσίασης των αποτελεσμάτων.

#### 4.4 Φόρμες παρουσίασης των αποτελεσμάτων

Χρησιμοποιώντας ως παράδειγμα τα δεδομένα του τελευταίου σεναρίου θα αναλύσουμε το τι γίνεται με τα δεδομένα που παίρνουμε από την βάση μας. Όπως έχουμε αναφέρει χρειαζόμαστε σε κάθε σενάριο τα Place\_id , act\_id καθώς και την ημερομηνία για να πάρουμε τα δεδομένα της εκάστοτε δραστηριότητας από την βάση και να τα παρουσιάσουμε στον χρήστη. Κάνοντας click λοιπόν στην ημερομηνία που επιθυμούμε εκτελείται το query:

```

SELECT T_CODE FROM TEMPLATE WHERE
TEMPLATE.ACT_ID=tmp[thr_index] AND
TEMPLATE.PLACE_ID=thr_id AND
TEMPLATE.DATE=act[thr_index]

```

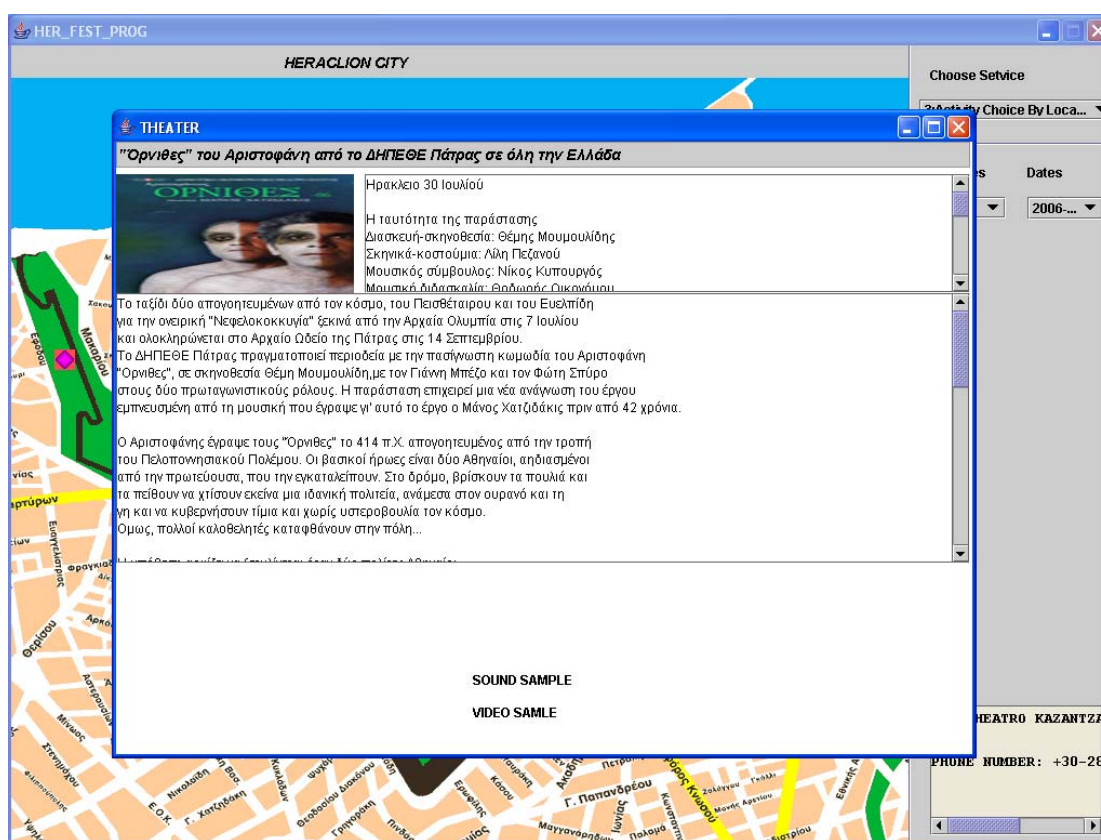
Όπου tmp[thr\_index] η τιμή του act\_id, thr\_id του place\_id και act[thr\_index] η ημερομηνία .Στο παράδειγμα μας έχουν τιμές 4 , 2 , 2006/07/30 αντίστοιχα. Το αποτέλεσμα αυτού του query υποδεικνύει στο πρόγραμμα ποια φόρμα να ανοίξει για να παρουσιάσει τα δεδομένα και περνάει σε αυτήν τις τιμές που θέλουμε. Εδώ επειδή

πρόκειται για θεατρική παράσταση το αποτέλεσμα είναι TT έτσι ώστε να μας ανοίξει την φόρμα παρουσίασης θεάτρων.

Έπειτα εκτελείται η δήλωση:

```
SELECT FILETYPE, RES
FROM TYPE
WHERE TYPE.ACT_ID=tact
AND TYPE.PLACE_ID=tpla
AND TYPE.DATE=tDA
```

Πάλι με τιμές 4,2,2006/07/30. Στη συνέχεια τα αποτελέσματα φιλτράρονται με βάση το Filetype ,σώζονται με τις αντίστοιχες καταλήξεις σε έναν φάκελο με το όνομα Her\_temp που δημιουργεί στο C:\ και τοποθετούνται στα κατάλληλα σημεία στη φόρμα.

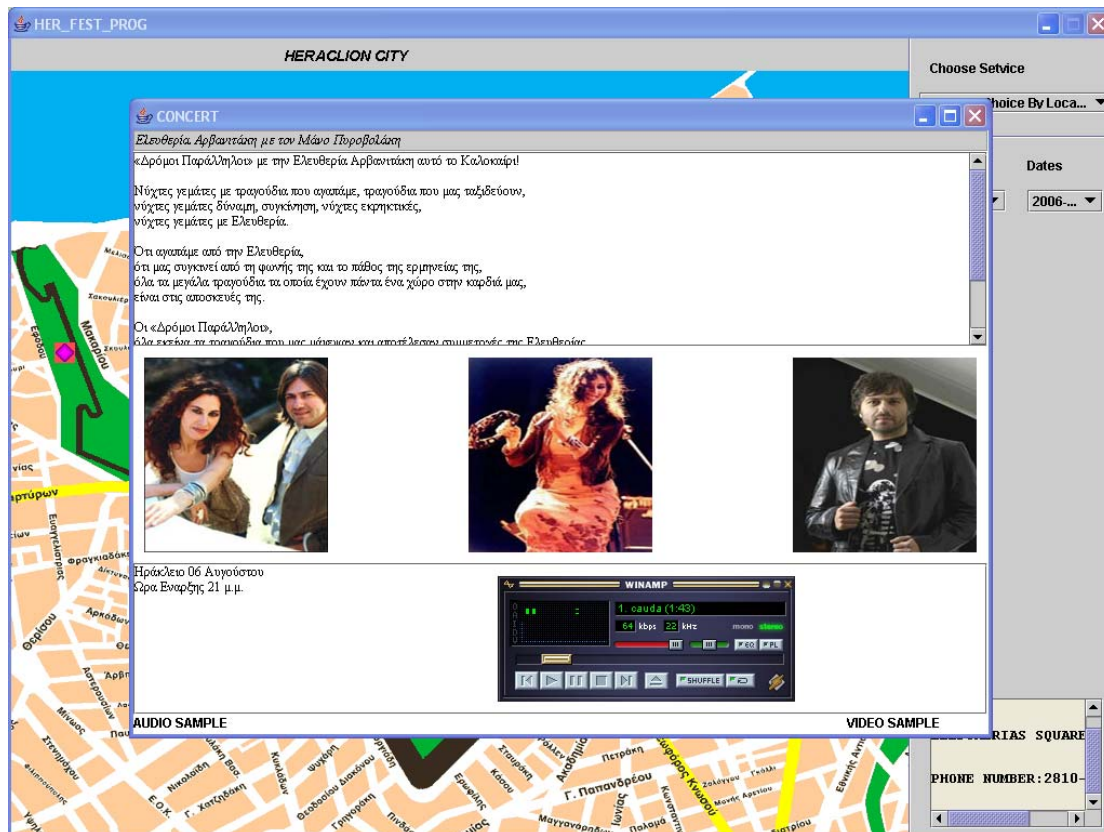


Εικόνα 48: Παράδειγμα παρουσίασης θεατρικής δραστηριότητας

Ας παραθέσουμε άλλο ένα παράδειγμα για να κατανοήσουμε καλύτερα τον τρόπο που λειτουργούν τα Templates. Στην πρώτο παράδειγμα επιλέγουμε Concert αντί για theater και από τις διαθέσιμες ημερομηνίες επιλέγουμε την 2006-08-06.Κάνοντας τέλος click στο highlighted σημείο εκτελούνται τα queries που αναφέραμε παραπάνω με τιμές πλέον:

```
T_code=CT
T_act=1
T_pla=1
T_da=2006/08/06
```

Έτσι λοιπόν ανοίγει το template για παρουσίαση συναυλιών (CT) παίρνει από την βάση δεδομένων τα αρχεία που χρειάζονται φιλτράρονται με βάση το filetype και τοποθετούνται πάνω στη φόρμα. Αξίζει να αναφέρουμε ότι σε αυτό το παράδειγμα υπάρχει οπτικό-ακουστικό υλικό οπότε αυτό μπορεί να ενεργοποιηθεί κάνοντας click στο αντίστοιχο button το οποίο θα ενεργοποιήσει το προεπιλεγμένο πρόγραμμα (Winamp στην δική μας περίπτωση) παρουσιάζοντας το ηχητικό δείγμα.



Εικόνα 49: Παράδειγμα παρουσίασης συναυλίας

Η μέθοδος που κάνει το φιλτράρισμα και την κατανομή είναι η εξής:

```
public void dst(ResultSet r) throws SQLException, IOException{
    ResultSetMetaData rmeta = r.getMetaData();
    int numColumns = rmeta.getColumnCount();
    int j=0;
    int point=0;
    int txtpoint=1;
    while (r.next())
    {
        for (int i=1;i<=numColumns;i++)
        {
            if (i<numColumns)
            {
                type[j]=r.getString(i);
            }
            else
            {
                Blob blob = r.getBlob(i);
                InputStream input =
                blob.getBinaryStream();
            }
        }
    }
}
```

```

ByteArrayOutputStream() ;
        ByteArrayOutputStream output = new
        byte[] rb = new byte[1024];
        int ch = 0;
        while ((ch=input.read(rb)) != -1) {
            output.write(rb, 0, ch);}

        byte[] blobData = output.toByteArray();
        if(type[j].equals("text"))
        {
            type2[j]=2;
        }
        else if (type[j].equals("image"))
        {
            type2[j]=1;
        }
        else if(type[j].equals("video"))
        {
            type2[j]=3;
        }
        else if(type[j].equals("audio"))
        {
            type2[j]=4;
        }
        switch(type2[j])
        {
            case 1:
                FileOutputStream fileOut = new
FileOutputStream("C:\\her_temp\\img"+point+".jpg");
                fileOut.write(blobData);
                fileOut.close();
                input.close();
                output.close();
                point++;
                break;
            case 2:
                FileOutputStream fileOuta = new
FileOutputStream("C:\\her_temp\\res"+j+".txt");
                fileOuta.write(blobData);
                fileOuta.close();
                DataInputStream df =null;
                FileInputStream f=null;
                String str="";
                byte []data=new byte[10240];
                f= new
FileInputStream("C:\\her_temp\\res"+j+".txt");
                df=new DataInputStream(f);
                df.read(data);
                switch(txtpoint)
                {
                    case 1:
                        str=new String(data);
                        str=str.trim();
                        jTextField1.setText(str);
                }
                break;
            case 2:
                str=new String(data);
                str=str.trim();
                jTextArea1.setText(str);

```

```

        break;
        case 3:
            str=new String(data);
            str=str.trim();
            jTextArea2.setText(str);
            break;
    }
    f.close();
    df.close();
    input.close();
    output.close();
    txtpoint++;
    break;

    case 3:
        FileOutputStream fileOutm = new
FileOutputStream("C:\\her_temp\\mov.mpeg");
        fileOutm.write(blobData);
        fileOutm.close();
        input.close();
        output.close();
        break;
    case 4:
        FileOutputStream fileOutau = new
FileOutputStream("C:\\her_temp\\aud.mp3");

        fileOutau.write(blobData);
        fileOutau.close();

        input.close();
        output.close();
        break;
    }
    input.close();
    output.close();
}
}
j++;
}
switch (point)
{
    case 0:
        break;
    case 1:
        jLabell1.setIcon(new
javax.swing.ImageIcon("C:\\her_temp\\img0.jpg"));
        break;
    case 2:
        jLabell1.setIcon(new
javax.swing.ImageIcon("C:\\her_temp\\img0.jpg"));
        jLabel2.setIcon(new
javax.swing.ImageIcon("C:\\her_temp\\img1.jpg"));
        break;
    case 3:
        jLabell1.setIcon(new
javax.swing.ImageIcon("C:\\her_temp\\img0.jpg"));
        jLabel2.setIcon(new
javax.swing.ImageIcon("C:\\her_temp\\img1.jpg"));

```



```
        jLabel15.setIcon(new  
javax.swing.ImageIcon("C:\\her_temp\\img2.jpg"));  
        break;  
    }  
}
```

Τέλος με την έξοδο από την φόρμα τα αρχεία που δημιουργήθηκαν καθώς και ο φάκελος Her\_temp διαγράφονται με τον εξής κώδικα:

```
private void exitForm(java.awt.event.WindowEvent evt) {  
    jLabel11.setIcon(null);  
    jLabel2.setIcon(null);  
    jLabel15.setIcon(null);  
  
    try{  
        String[] jk=Mydirc.list();  
        for(int i=0;i<jk.length;i++)  
        {  
  
            File temp = new File("C:\\her_temp\\"+jk[i]);  
            temp.delete();  
        }  
        Mydirc.delete();  
    }  
    catch (Exception e)  
    {  
        System.out.println(e);  
    }  
}
```

## 5 Συμπεράσματα

---

Ολοκληρώνοντας αυτή την εργασία και λαμβάνοντας υπ' όψιν όλες τις παραμέτρους της και το τρόπο λειτουργίας της διαπιστώνουμε πως πλέον η χρήση συστημάτων γραφικής αναπαράστασης επερωτήσεων δεν αποτελεί απλώς μια επιπλέον διευκόλυνση στη διάδραση συστήματος-χρήστη αλλά έχει γίνει μια απαραίτητη συνθήκη κυρίως στο χώρο των εμπορικών εφαρμογών τέτοιου τύπου. Στο πρόγραμμά μας το να κάνει ο χρήστης χωρικές δραστηριότητες όπως το κλικάρισμα πάνω σε εικονίδια, η επιλογή των περιεχομένων ενός ή περισσότερων combo boxes, η μετακίνηση μπαρών κύλισης τον βοηθάει στο να έχει καλύτερη αντίληψη και κατανόηση των ερωτήσεων που πρόκειται να θέσει στη βάση δεδομένων. Ουσιαστικά το σύστημα τον βοηθάει στο να αντιλαμβάνεται τι ακριβώς ζητάει και πως ακριβώς θα το βρει. Μετά την υποβολή των ερωτημάτων και την αναζήτηση των επιθυμητών αποτελεσμάτων στη βάση μας η παρουσίαση τους είναι επίσης τέτοια που να επιτρέπει την λήψη πολλών πληροφοριών σε μικρό χρονικό διάστημα με τρόπο εύκολο και κατανοήσιμο.

Όπως σε όλους τους έντυπους τουριστικούς οδηγούς, έτσι και σε ένα ηλεκτρονικό οδηγό ο χάρτης θεωρείται απαραίτητος. Καθοδηγεί τον χρήστη και τον βοηθάει στο να προσανατολιστεί σε σχέση με τις δραστηριότητες που λαμβάνουν χώρα κάθε φορά. Με βάση αυτό μεσ' τις δυνατότητες του προγράμματος θα ήταν και η ύπαρξη ενός δυναμικού χάρτη όπου θα πρόσφερε επιπλέον γεωπληροφοριακά δεδομένα και θα εμπλούτιζε σημαντικά την αξία της προσφερόμενης υπηρεσίας. Κάτι τέτοιο σε συνδυασμό με σύγχρονα συστήματα gis, gps αλλά και τη χρήση του σε φορητές συσκευές (pdas) θα μπορούσε να αποτελέσει μία ποιοτική λύση ενός σύγχρονου ηλεκτρονικού τουριστικού-ψυχαγωγικού οδηγού.

Όπως έχουμε επανειλημμένα αναφέρει σκοπός του προγράμματος μας, καθώς και όλων των προγραμμάτων γραφικής αναπαράστασης επερωτήσεων, είναι να επιτρέπει στον χρήστη να υποβάλει τα ερωτήματα του προς την βάση και να παίρνει τα αποτελέσματα που θέλει με τρόπο απλό και εύχρηστο. Δεν πρέπει δηλαδή να προϋποθέτουμε ότι ο χρήστης έχει προηγούμενη εμπειρία με παρόμοια συστήματα για να μπορεί να κάνει τις εργασίες που επιθυμεί. Επίσης είναι βασικό για το πρόγραμμα μας τα αποτελέσματα που παίρνει ο χρήστης να παρουσιάζονται με τρόπο σαφή και ευπαρουσίαστο. Οι παραπάνω στόχοι είχαν ως αποτέλεσμα μια εφαρμογή που επιτρέπει σε κάθε ενδιαφερόμενο την δυνατότητα να αναζητήσει δραστηριότητες επιλέγοντας απλά ημερομηνίες ή κάνοντας απλά click σε σημεία πάνω σε έναν χάρτη.

Το πρόγραμμα θα μπορούσε να εγκατασταθεί αυτόνομα έτσι ώστε να το χρησιμοποιούν οι χρήστες αλλά θα μπορούσε επίσης να ενταχθεί σε μια γενικότερη ομάδα προγραμμάτων που ασχολείται με θέματα ψυχαγωγίας ή τουρισμού. Ειδικά στην δεύτερη περίπτωση το πρόγραμμα μπορεί να δεχτεί μια σειρά από βελτιώσεις έτσι ώστε να συνυπάρχει αρμονικά με την υπόλοιπη e-community. Κάποιες από αυτές παραθέτονται παρακάτω:

- Προσθήκη administration έτσι ώστε η ανανέωση των δραστηριοτήτων να γίνεται από μη εξειδικευμένους χρήστες
- Προσθήκη επιλογής e-booking για την κράτηση θέσεων στις δραστηριότητες
- Εμπλουτισμός του χάρτη έτσι ώστε να περιέχει παραπάνω σημεία από τα σημεία διεξαγωγής των δραστηριοτήτων για να γίνεται στον χρήστη γνωστή η τοποθεσία διεξαγωγής της δραστηριότητας σε σχέση με την θέση στην οποία βρίσκεται ή άλλα γνωστά σημεία της πόλης.

## 6 Βιβλιογραφία

---

1. Journal of Visual Languages and Computing (2002) Iconic SQL: Practical issues in the querying databases through structured iconic expressions. A. Narayanan and D. Shaman. School of engineering and computer science. University of Exeter.
2. T. Catarci, M.F. Costabile, S. Levialdi, C. Batini Visual Query Systems for Databases: A Survey (1996)
3. *Journal of Visual Languages and Computing* (1997). Maps as a Metaphor in a Geographical Hypermedia System. JULIEN CAPORAL AND YANN VIE" MONT
4. 4. Spatial Query by Sketch (1997). Max j. Egenhofer. Department of spatial information science and engineering department of computer science and national center for geographical information and analysis. University of Maine.
5. Bringing Dynamic Queries to Mobile Devices: a Visual Preference-based Search Tool for Tourist Decision Support (2005). Stefano Burigat, Luca Chittaro, and Luca De Marco HCI Lab, Dept. of Math and Computer Science, University of Udine
6. Graphical Query Interfaces for Semi structured Data: The QURSED System (2005). MICHALIS PETROPOULOS, YANNIS PAPAKONSTANTINOU University of California, San Diego and VASILIS VASSALOS Athens University of Economics and Business, Athens, Greece
7. Αξιοποίηση Του Γεωγραφικού Συστήματος Πληροφοριών Για Την Υλοποίηση Υπηρεσιών Τουριστικής Πληροφόρησης Με Βάση Τον Εντοπισμό Θέσης. Κούβας Γιώργος, Κακαλέτρης Γιώργος, Βερνάρδος Γιώργος, Αναστασία Γαρμπή Πανεπιστήμιο Αθηνών, Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Exodus A.E., Αθήνα
8. Sketching Spatial Queries (2000). Andreas D. Blaser Swiss Federal Institute of Technology in Zurich A thesis submitted for the degree of doctor of philosophy