

<b>1. ΕΙΣΑΓΩΓΗ</b>	<b>2</b>
<b>2. ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ</b>	<b>3</b>
<b>2.1 Σχεσιακό Μοντέλο (Relational Model)</b>	<b>4</b>
2.1.1 SQL (Structured Query Language)	5
2.1.2 Πρωτεύων Κλειδί (Primary Key)	6
2.1.3 Σχεσιακοί Κανόνες (Relational Rules)	6
2.1.4 Κανονικοποίηση (Normalization)	7
<b>2.2 Ιεραρχικό Μοντέλο (Hierarchical Model)</b>	<b>9</b>
<b>2.3 Δικτυακό Μοντέλο (Network Model)</b>	<b>13</b>
<b>2.4 Βάσεις Δεδομένων: Κύριες Λειτουργίες - Λόγοι για να τις χρησιμοποιήσουμε</b>	<b>18</b>
<b>3. ACCESS: ΣΥΝΟΠΤΙΚΗ ΠΕΡΙΓΡΑΦΗ</b>	<b>20</b>
<b>3.1 Αντικείμενα Βάσεων Δεδομένων (Database Objects)</b>	<b>22</b>
3.1.1 Πίνακες	23
3.1.2 Ερωτήματα	23
3.1.3 Φόρμες	24
3.1.4 Αναφορές	25
3.1.5 Μακροεντολές - προγραμματισμός χωρίς προγραμματισμό	25
3.1.6 Access Basic	25
<b>4. Ανάλυση και σχεδίαση ΤΟΥ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ</b>	<b>27</b>
<b>4.1 ΑΝΑΛΥΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΤΜΗΜΑΤΟΣ "ΞΕΝΟΔΟΧΕΙΑ"</b>	<b>29</b>
4.1.1 Πίνακες Ξενοδοχείων	29
4.1.2 Δημιουργία Σχέσεων	37
4.1.3 Ερωτήματα	42
4.1.4 Φόρμες/Μακροεντολές	49
4.1.5 Αναφορές	59
<b>4.2 ΑΛΛΑ ΤΜΗΜΑΤΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ</b>	<b>61</b>
4.2.1 "Ενοικιαζόμενα Δωμάτια" (Rooms)	61
4.2.2 "Γραφεία Ενοικιάσεως Αυτοκινήτων" (Cars)	61
4.2.3 "Γραφεία Ενοικιάσεως Μοτοσυκλετών" (Moto)	62
4.2.4 "Ταξιδιωτικά Γραφεία" (Travel)	62
4.2.5 "Διάφορα Άλλα" (More)	62
<b>5. ΕΦΑΡΜΟΓΗ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ</b>	<b>64</b>
<b>5.1 Κύρια Φόρμα της Εφαρμογής</b>	<b>64</b>
<b>5.2 Κύρια Φόρμα των Ξενοδοχείων</b>	<b>67</b>
<b>5.3 Επιλογές</b>	<b>69</b>
<b>5.4 Εμφάνιση Στοιχείων Ξενοδοχείων</b>	<b>73</b>
<b>5.5 Εμφάνιση Στοιχείων Ξενοδοχείων:Εμφάνιση</b>	<b>76</b>
<b>5.6 Εμφάνιση Στοιχείων Ξενοδοχείων:Εισαγωγή</b>	<b>76</b>
<b>5.7 Υπεύθυνοι - Ιδιοκτήτες</b>	<b>80</b>
<b>6. Συμπεράσματα</b>	<b>81</b>
<b>7. ΒΙΒΛΙΟΓΡΑΦΙΑ</b>	<b>84</b>

## 1. ΕΙΣΑΓΩΓΗ

Το πεδίο των βάσεων δεδομένων έχει υποστεί δραματικές αλλαγές στα σαράντα περίπου χρόνια της ύπαρξής του και συνεχίζει να αλλάζει γρήγορα ακόμη και σήμερα. Η σχεσιακή προσέγγιση αναπτύχθηκε μόλις πριν από 25 χρόνια και τα προϊόντα σχεσιακών βάσεων δεδομένων είναι διαθέσιμα στο εμπόριο εδώ και 15 περίπου χρόνια, ενώ οι εξελίξεις είναι τόσο γρήγορες ώστε μόλις ανακοινώνονται νέα πρότυπα και νέα προϊόντα, σχεδόν αμέσως κάποια άλλα παίρνουν τη θέση τους, υποσχόμενα βελτιωμένα χαρακτηριστικά και δυνατότητες.

Σκοπός της διπλωματικής εργασίας είναι η δημιουργία μιας εφαρμογής για την Τουριστική Αστυνομία Χανίων, με τη βοήθεια του προγράμματος διαχείρισης σχεσιακών βάσεων δεδομένων Microsoft Access 95. Πρόκειται για μία βάση δεδομένων, η οποία καλύπτει τις ακόλουθες τουριστικές επιχειρήσεις που λειτουργούν στο νομό Χανίων :

- **"Ξενοδοχεία"**, στα οποία περιλαμβάνονται και άλλα τέσσερα είδη επιχειρήσεων που έχουν άδεια λειτουργίας Ξενοδοχείου, και είναι τα "Διαμερίσματα", τα "Ενοικιαζόμενα δωμάτια", τα "Οικοτροφεία" και οι "Ξενώνες".
- **"Ενοικιαζόμενα δωμάτια"**
- **"Γραφεία ενοικιάσεως αυτοκινήτων"**
- **"Γραφεία ενοικιάσεως μοτοσυκλετών"**
- **"Ταξιδιωτικά γραφεία"**, και
- **"Διάφορα άλλα"**, στα οποία περιλαμβάνονται τα εξής είδη επιχειρήσεων : "Bar", "Disco", "Club", "Restaurant", "Καφετέρια", "Κατάστημα", "Ζαχαροπλαστείο", "Ταβέρνα" και "Μικροπωλητές - Άγρα".

Για κάθε είδος, υπάρχει η δυνατότητα φιλτραρίσματος των δεδομένων μέσω συγκεκριμένων επιλογών που γίνονται από τον χρήστη καθώς και η εκτύπωση των αποτελεσμάτων που προκύπτουν, ενώ ιδιαίτερη φροντίδα έχει δοθεί στην όσο το δυνατόν πιο εύκολη καταχώρηση δεδομένων, με τη χρήση, όπου αυτό είναι δυνατό, των ανάλογων πλαισίων καταλόγου.

## 2. ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

*Βάση δεδομένων* είναι μία συλλογή από εγγραφές και αρχεία (συλλογή πληροφοριών) σχετικά με ένα ορισμένο θέμα ή επιχειρησιακή εφαρμογή. Οι βάσεις δεδομένων οργανώνουν τις πληροφορίες αυτές με λογική δομή έτσι ώστε να διευκολύνεται η προσπέλαση και η αναζήτησή τους[10].

Ένα σύστημα διαχείρισης βάσεων δεδομένων (database management system ή DBMS), έχει σαν σκοπό την οργάνωση και διαχείριση των αρχείων μιας επιχείρησης με τη βοήθεια Η/Υ. Για παράδειγμα, μια εταιρία χονδρικής πώλησης, φυσιολογικά θα χρησιμοποιούσε ένα DBMS για να κρατάει αρχεία σχετικά με τις πωλήσεις της, ενώ ένα Πανεπιστήμιο θα χρησιμοποιούσε ένα DBMS για να κρατάει αρχεία σχετικά με τους φοιτητές του (βαθμολογίες, δίδακτρα κ.λπ.). Επίσης, όλες οι αεροπορικές εταιρίες χρησιμοποιούν DBMS για να διαχειρίζονται στοιχεία όπως πτήσεις και κρατήσεις θέσεων.

Ένα σύστημα διαχείρισης βάσεων δεδομένων (DBMS) είναι δυνατόν να ασχολείται με περισσότερες από μία βάσεις δεδομένων κάθε στιγμή. Για παράδειγμα, ένα Πανεπιστήμιο μπορεί να διατηρεί μια βάση για τους εγγεγραμμένους φοιτητές και μια δεύτερη για τα βιβλία που υπάρχουν στη βιβλιοθήκη. Τότε, διαφορετικοί χρήστες μπορούν να έχουν πρόσβαση σ'αυτές τις δύο βάσεις δεδομένων μέσω του ίδιου συστήματος διαχείρισης βάσεων δεδομένων.

Έχουν αναπτυχθεί διάφορες προσεγγίσεις για τη δόμηση πρόσβασης στις πληροφορίες μιας βάσης δεδομένων. Ιστορικά, δύο απ' αυτές παρείχαν ό,τι αργότερα αναγνωρίστηκε σαν "διακριτά μοντέλα δεδομένων για τη δόμηση πληροφοριών" :

- Το "Ιεραρχικό μοντέλο" (Hierarchical model) στα τέλη της δεκαετίας του '60,
- Το "Δικτυακό μοντέλο" (Network model) στις αρχές της δεκαετίας του '70 [9].

Υπάρχει φυσικά και το "Σχεσιακό μοντέλο" (Relational model) το οποίο χρησιμοποιείται περισσότερο από κάθε άλλο και ο λόγος που συμβαίνει κάτι τέτοιο φαίνεται τόσο από την μελέτη του, όσο και από τη σύγκριση που γίνεται στη συνέχεια μεταξύ αυτού και των άλλων δύο μοντέλων.

## 2.1 Σχεσιακό Μοντέλο (*Relational Model*)

Το μοντέλο δεδομένων που χρησιμοποιείται περισσότερο από κάθε άλλο για προϊόντα συστημάτων βάσεων δεδομένων, είναι το σχεσιακό μοντέλο, το οποίο παρέχει μεγάλη ευελιξία και επιτρέπει σε μη-προγραμματιστές να δημιουργούν γενικά ερωτήματα γρήγορα και εύκολα, με τα οποία μπορούν να ανακτούν από μια βάση δεδομένων τις πληροφορίες που επιθυμούν. Ένα σύστημα διαχείρισης βάσεων δεδομένων (DBMS) που κάνει χρήση του σχεσιακού μοντέλου, είναι γνωστό σαν σχεσιακό DBMS ή RDBMS, αν και συχνά λέμε απλά DBMS ή σύστημα βάσεων δεδομένων και εννοούμε RDBMS.

Στο σχεσιακό μοντέλο τα δεδομένα αποθηκεύονται χωριστά σε πίνακες και κατόπιν συνδέονται ή σχετίζονται με κοινά πεδία του κάθε πίνακα. Η σύνδεση πινάκων μας βοηθά να δημιουργήσουμε ένα εξεζητημένο σύστημα βάσης δεδομένων και διευκολύνει στη διαχείριση των πληροφοριών. Κάθε σύνδεση ανάμεσα σε δύο πίνακες ονομάζεται σχέση (relationship). Μπορούμε να έχουμε μία αμφιμονοσήμαντη σχέση (ένα-προς-ένα), στην οποία μια εγγραφή ενός πίνακα σχετίζεται με μια εγγραφή σε έναν άλλο πίνακα μέσω ενός κοινού πεδίου. Υπάρχουν και άλλα είδη σχέσεων, όπως μονοσήμαντη (ένα-προς-πολλά) και πολυσήμαντη (πολλά-προς-πολλά), που μπορούν να χρησιμοποιηθούν σε πιο εξελιγμένες εφαρμογές [11].

Το σχεσιακό μοντέλο, το οποίο αναπτύχθηκε σαν αποτέλεσμα των αναγνωρισμένων αδυναμιών (ατελειών) των ιεραρχικών και δικτυακών DBMS, παρουσιάστηκε από τον Codd το 1970 σε ένα άρθρο του που εκδόθηκε στο περιοδικό "Communications of the ACM"[1]. Ένα μεγάλο μέρος από την πρακτική αξία του σχεσιακού μοντέλου βρίσκεται στην ισχυρή θεωρητική βάση του. Ο Codd, πίστεψε πως ένα μοντέλο με τεκμηριωμένη θεωρητική βάση θα έλυνε τα περισσότερα από τα πρακτικά προβλήματα που ενδεχομένως θα μπορούσαν να προκύψουν.

Σε ένα άλλο άρθρο του[2], ο Codd αναπτύσσει την υπόθεση της υιοθέτησης του σχεσιακού μοντέλου αντί των άλλων μοντέλων βάσεων δεδομένων και υπάρχουν τρεις λόγοι στους οποίους στηρίζεται αυτό το επιχείρημα. Πρώτον, τα άλλα μοντέλα αναγκάζουν τον προγραμματιστή εφαρμογών να κωδικοποιεί σε ένα χαμηλό και λεπτομερέςτατο επίπεδο δόμησης (at a low level of structural detail). Σαν αποτέλεσμα, τα προγράμματα εφαρμογών είναι πολύ πιο πολύπλοκα και απαιτούν

περισσότερο χρόνο τόσο για το γράψιμο, όσο και για την εκσφαλμάτωσή τους. Δεύτερον, δεν παρέχονται οι δυνατότητες για την ταυτόχρονη επεξεργασία πολλών εγγραφών. Στο σχεσιακό μοντέλο όπου κάτι τέτοιο είναι εφικτό, τα ερωτήματα μπορούν να εκφραστούν πολύ πιο περιληπτικά (και φυσικά πολύ πιο γρήγορα). Τρίτον, το σχεσιακό είναι το μοναδικό μοντέλο, το οποίο μέσω μιας γλώσσας ερωτημάτων όπως η SQL (Structured Query Language), αναγνωρίζει την ανάγκη των προγραμματιστών να δημιουργούν ερωτήματα εύκολα και κυρίως γρήγορα [13].

### 2.1.1 SQL (Structured Query Language)

Η SQL δημιουργήθηκε στα εργαστήρια ερευνών της IBM και χρησιμοποιείται ευρέως ως γλώσσα σχεσιακών βάσεων δεδομένων. Τόσο το Αμερικανικό Ινστιτούτο Τυποποίησης (American National Standards Institute - ANSI) όσο και ο Διεθνής Οργανισμός Τυποποίησης (International Organization for Standardization - ISO) έχουν επιλέξει την SQL ως την πρότυπη γλώσσα για αυτά τα συστήματα μια και οι δυνατότητές της είναι ουσιώδεις για την διαχείριση των δεδομένων σ' ένα κόσμο που εξαρτάται ολοένα και περισσότερο από την τεχνολογία των σχεσιακών βάσεων δεδομένων.

Πρόκειται για μία πλήρη γλώσσα βάσεων δεδομένων. Χρησιμοποιείται για τον σχεδιασμό μιας σχεσιακής βάσης δεδομένων, τη δημιουργία απόψεων (views) και τον καθορισμό ερωτημάτων. Επιπλέον, επιτρέπει την εισαγωγή, ανανέωση και διαγραφή εγγραφών. Κατά την ορολογία των βάσεων δεδομένων, πρόκειται τόσο για μια γλώσσα ορισμού δεδομένων (data definition language - DDL) όσο και μια γλώσσα διαχείρισης δεδομένων (data manipulation language - DML). Ωστόσο, δεν είναι μια πλήρης γλώσσα προγραμματισμού όπως η COBOL και η C. Εφόσον όμως οι εντολές της SQL μπορούν να ενσωματωθούν σε άλλες γλώσσες προγραμματισμού, χρησιμοποιείται συχνά σε συνδυασμό μ' αυτές τις γλώσσες για τη δημιουργία προγραμμάτων εφαρμογών. Οι ενσωματωμένες εντολές της SQL χειρίζονται την επεξεργασία των βάσεων δεδομένων, ενώ οι εντολές της γλώσσας προγραμματισμού εκτελούν τις απαραίτητες ενέργειες για την ολοκλήρωση της εφαρμογής [13].

### 2.1.2 Πρωτεύων Κλειδί (Primary Key)

Ένας πίνακας, όπως θα δούμε σε επόμενο κεφάλαιο, είναι ένα σύνολο σχετιζόμενων πληροφοριών (περιέχει πληροφορίες για ένα συγκεκριμένο θέμα, όπως οι ιδιοκτήτες). Περιέχει πεδία (fields) στα οποία αποθηκεύονται τα διαφορετικά είδη πληροφοριών (όπως είναι το όνομα ή η διεύθυνση ενός πελάτη), και εγγραφές (records) που συλλέγουν όλες τις πληροφορίες για μια συγκεκριμένη περίπτωση του θέματος, όπως όλες οι πληροφορίες για έναν ιδιοκτήτη που ονομάζεται Κώστας Γεωργίου.

Σε κάθε πίνακα μιας βάσης δεδομένων δημιουργούμε ένα πρωτεύων κλειδί, το οποίο είναι ένα πεδίο που περιέχει μία μοναδική τιμή για κάθε εγγραφή του πίνακα. Έτσι, δεν υπάρχουν δύο εγγραφές που να έχουν την ίδια τιμή για τα πρωτεύοντα κλειδιά τους, διότι η ανάθεση ενός πρωτεύοντος κλειδιού μας διασφαλίζει ότι δεν θα καταχωρήσουμε την ίδια πληροφορία για το πεδίο του περισσότερες από μία φορές σε έναν πίνακα, μια και η διπλή καταχώρηση δεν θα γίνει αποδεκτή. Ειδικά στην Access, ένα από τα πιο δημοφιλή προγράμματα διαχείρισης σχεσιακών βάσεων δεδομένων (υπάρχουν πολλά άλλα, όπως τα dBaseIV, Paradox, Oracle, Ingres, FoxPro και Btrieve, στα οποία η Access μπορεί να έχει άμεση πρόσβαση και να κάνει ενημερώσεις και επεξεργασία αρχείων), δημιουργείται αυτόματα ένα ευρετήριο για τα πρωτεύοντα κλειδιά, κι έτσι μπορεί εύκολα και γρήγορα να γίνει αναζήτηση πληροφοριών και ταξινόμηση πινάκων, βασισμένη στα πεδία τους [12].

### 2.1.3 Σχεσιακοί Κανόνες (Relational Rules)

Στο σχεσιακό μοντέλο υπάρχουν κάποιοι κανόνες που ορίζουν το πώς θα πρέπει να γίνεται η δόμηση πινάκων και περιορίζουν κάποιες πιθανές λειτουργίες ανάκτησης δεδομένων. Σκοπός αυτών των κανόνων είναι να υποδείξουν στις ποικίλες προσφορές εμπορικών προϊόντων κάποιες περιοχές τυποποίησης, έτσι ώστε ο σχεδιασμός βάσεων δεδομένων να είναι ίδιος για όλα τα προϊόντα. Είναι οι ακόλουθοι :

♦ *1ος Κανόνας: "Πρώτη Κανονική Μορφή"*

Στον ορισμό πινάκων, το σχεσιακό μοντέλο δεν επιτρέπει να υπάρχουν πεδία οποιασδήποτε εγγραφής με παραπάνω από μία τιμές (ή αλλιώς επαναλαμβανόμενα πεδία). Αποτελεί το πρώτο βήμα της Κανονικοποίησης που θα δούμε στη συνέχεια.

- ◆ *2ος Κανόνας: "Πρόσβαση σε κάποια εγγραφή μόνο μέσω των περιεχομένων της"*  
Μπορούμε να ανακτήσουμε δεδομένα από κάποια εγγραφή μέσω των τιμών κάποιων πεδίων της και όχι μέσω της σειράς με την οποία περάστηκε στη βάση δεδομένων. Η σειρά αυτή είναι άνευ σημασίας. Για παράδειγμα, δεν μπορούμε να ζητήσουμε από ένα ερώτημα να μας εμφανίσει την τρίτη εγγραφή του πίνακα Ιδιοκτήτες. Πρέπει να ρωτήσουμε για συγκεκριμένες τιμές, όπως Όνομα, Επώνυμο ή αύξων αριθμός.
  
- ◆ *3ος Κανόνας: "Μοναδικές εγγραφές"*  
Ο τρίτος κανόνας, λέει ότι δύο εγγραφές σ' έναν πίνακα δεν πρέπει να έχουν τις ίδιες τιμές για όλα τα πεδία τους. Αυτό συμβαίνει, επειδή κάθε εγγραφή πρέπει να ξεχωρίζει από όλες τις άλλες για να μπορεί να ανακτηθεί με βάση τις τιμές των πεδίων της από κάποιο ερώτημα [9].

#### 2.1.4 Κανονικοποίηση (Normalization)

Η κανονικοποίηση είναι μία βήμα-προς-βήμα αντιστρέψιμη μέθοδος για τη μετατροπή μιας (μη-κανονικοποιημένης) σχέσης, σε άλλες σχέσεις με απλούστερες δομές. Εφόσον μάλιστα πρόκειται για αντιστρέψιμη μέθοδο, δεν υπάρχει κίνδυνος να χαθούν κάποιες πληροφορίες κατά τη διάρκεια της μετατροπής, ενώ τα οφέλη που αποκομίζουμε από τη χρήση της μεθόδου αυτής, είναι η αντιμετώπιση του προβλήματος της "σπατάλης χώρου" στη βάση δεδομένων μας και φυσικά η δημιουργία πιο απλών και αποδοτικών σχέσεων. Οι τέσσερις κανόνες της κανονικοποίησης, είναι οι εξής:

*1ος Κανόνας: "Μοναδικότητα Πεδίων"*

Σύμφωνα με τον πρώτο κανόνα, "κάθε πεδίο κάποιου πίνακα πρέπει να περιέχει ένα μοναδικό είδος πληροφοριών". Πρόκειται για τον πρώτο από τους σχεσιακούς κανόνες που διατυπώθηκαν στην προηγούμενη παράγραφο.

*2ος Κανόνας: "Πρωτεύοντα Κλειδιά"*

Σ' ένα σωστό σχεδιασμό σχεσιακής βάσης δεδομένων, κάθε εγγραφή οποιουδήποτε πίνακα πρέπει να είναι μοναδική. Αυτό σημαίνει πως δεν πρέπει να υπάρχουν δύο γραμμές σ' έναν πίνακα που μπορεί να είναι ίδιες. Για παράδειγμα, δεν έχει νόημα να κρατάμε δύο εγγραφές που περιγράφουν τον ίδιο Ιδιοκτήτη. Επομένως, "κάθε πίνακας

πρέπει να έχει ένα μοναδικό αναγνωριστικό, ή πρωτεύον κλειδί, που να αποτελείται από ένα ή περισσότερα πεδία του".

**3ος Κανόνας: "Λειτουργική Εξάρτηση"**

Εφόσον έχουμε πρωτεύον κλειδί σε κάθε πίνακα, μπορούμε να κάνουμε έναν έλεγχο για να βεβαιωθούμε ότι έχουμε συμπεριλάβει όλες τις σχετικές πληροφορίες με το θέμα του πίνακα. Για να χρησιμοποιήσουμε την ορολογία της θεωρίας σχεδιασμού σχεσιακών βάσεων δεδομένων, θα πρέπει να ελέγξουμε αν κάθε πεδίο είναι *εξαρτημένο λειτουργικά* (functionally dependent) από το πρωτεύον κλειδί. Επομένως, "για κάθε μοναδική τιμή πρωτεύοντος κλειδιού πρέπει να υπάρχει μία και μόνο μία τιμή σε οποιαδήποτε από τις στήλες δεδομένων, η οποία θα είναι σχετική με το θέμα του πίνακα".

**4ος Κανόνας: "Ανεξαρτησία Πεδίων"**

Ο τελευταίος κανόνας, ελέγχει για πιθανά προβλήματα όταν κάνουμε αλλαγές στα δεδομένα των πινάκων. Σύμφωνα με αυτόν, "πρέπει να μπορούμε να κάνουμε αλλαγές σε κάποιο πεδίο (που δεν περιλαμβάνεται στο πρωτεύον κλειδί) χωρίς να επηρεάζεται κάποιο άλλο" [12].

Στην εικόνα που ακολουθεί φαίνεται ένα απλό παράδειγμα κανονικοποίησης. Στον πρώτο πίνακα (στον οποίο δεν έχει ορισθεί πρωτεύον κλειδί), υπάρχουν τα πεδία "ΙδιοκτήτηςID" και "Όνοματεπώνυμο Ιδιοκτήτη". Στο δεύτερο πεδίο περιλαμβάνεται τόσο το "Επώνυμο" όσο και το "Όνομα" του ιδιοκτήτη, ενώ οι δύο τελευταίες εγγραφές είναι ακριβώς ίδιες (κάτι που φυσικά δεν έχει νόημα να συμβαίνει). Στον δεύτερο (κανονικοποιημένο) πίνακα υπάρχουν ξεχωριστά πεδία για το "Επώνυμο" και το "Όνομα" και έχει οριστεί πρωτεύον κλειδί (το "ΙδιοκτήτηςID") το οποίο διασφαλίζει πως κάθε εγγραφή του πίνακα είναι μοναδική.

<u>ΙδιοκτήτηςID</u>	<u>Όνοματεπώνυμο Ιδιοκτήτη</u>	
	<i>Επώνυμο</i>	<i>Όνομα</i>
1	Ιωάννου	Κώστας
2	Πέτρου	Νίκος
3	Γεωργίου	Ανδρέας
3	Γεωργίου	Ανδρέας

*Μη κανονικοποιημένη μορφή*



<u>*ΙδιοκτήτηςID</u>	<u>Επώνυμο</u>	<u>Όνομα</u>
1	Ιωάννου	Κώστας
2	Πέτρου	Νίκος
3	Γεωργίου	Ανδρέας

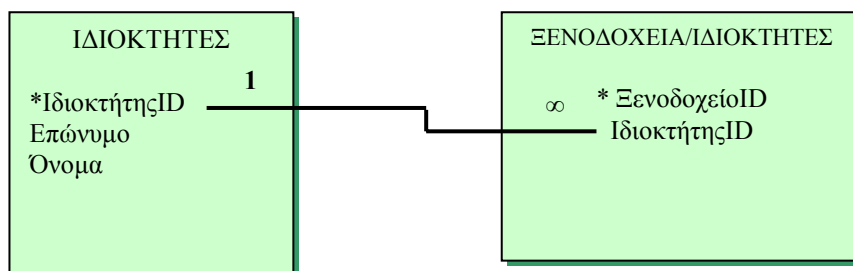
*Κανονικοποιημένη μορφή*

**Εικόνα 2-1:** Παράδειγμα Κανονικοποίησης

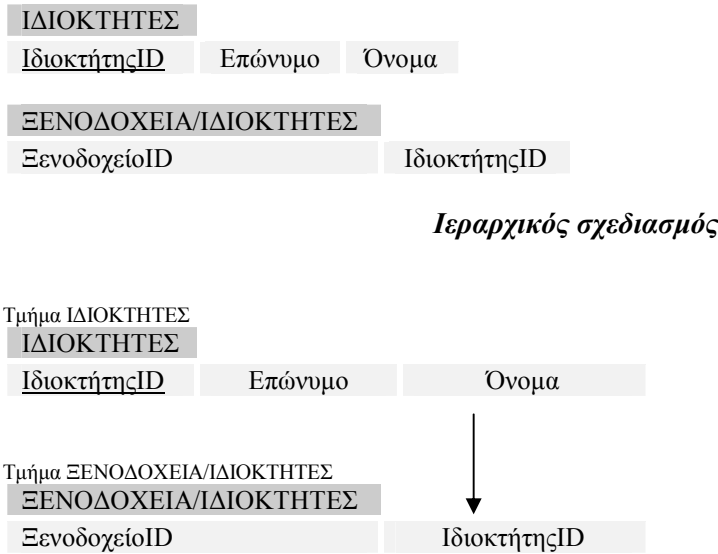
### 2.2 *Ιεραρχικό Μοντέλο (Hierarchical Model)*

Σε αντίθεση με το σχεσιακό μοντέλο, το ιεραρχικό δεν ξεκίνησε σαν ερευνητικό έργο σε κάποιο εργαστήριο. Οφείλει την ύπαρξή του στην ανάγκη των αμερικανικών διαστημικών προγραμμάτων για ταχύτερη πρόσβαση σε μεγάλες ποσότητες συσσωρευμένων δεδομένων. Συγκεκριμένα, οι μηχανικοί έπρεπε να μπορούν να διαχειρίζονται δεδομένα τα οποία περιέγραφαν εκατομμύρια τμήματα συσχετισμένα ιεραρχικά μεταξύ τους, κι έτσι το μοντέλο αυτό προέκυψε από την ανάγκη εξεύρεσης λύσης σε ένα εξαιρετικά σημαντικό πρόβλημα διαχείρισης δεδομένων.

Η βασική δομή του ιεραρχικού μοντέλου είναι μία ιεραρχία σχέσεων σε μορφή δέντρου. Η μορφή αυτή προκύπτει από την "ένα-προς-πολλά" σχέση (που ονομάζεται γονέας-παιδί σ' αυτήν την περίπτωση). Έτσι, ένας γονέας μπορεί να έχει πολλά παιδιά, αλλά ένα παιδί μπορεί να έχει μόνο έναν γονέα. Η δομή αυτή λοιπόν όταν σχεδιαστεί μοιάζει με δέντρο, αν και τις περισσότερες φορές η ρίζα είναι στην κορυφή. Το σχήμα που ακολουθεί (Εικόνα 2-2) δείχνει τον σχεσιακό και ιεραρχικό σχεδιασμό του ίδιου μοντέλου δεδομένων.



*Σχεσιακός σχεδιασμός*



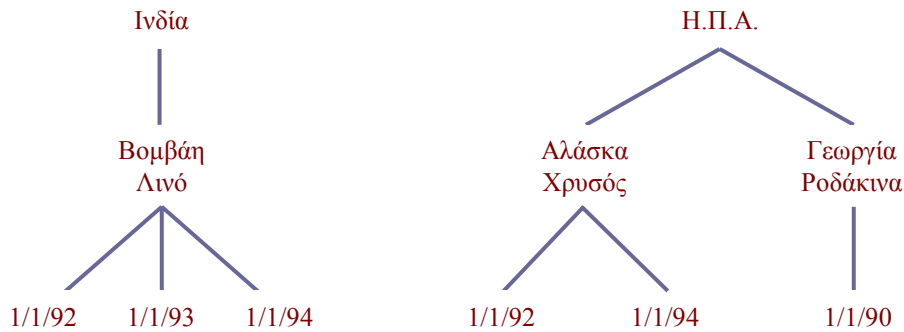
**Εικόνα 2-2:** Σύγκριση σχεσιακού και ιεραρχικού σχεδιασμού μιας βάσης δεδομένων.

Η κυριότερη διαφορά που παρατηρείται στον σχεδιασμό των δύο μοντέλων είναι η χρησιμοποίηση κοινών πεδίων στο σχεσιακό και δεικτών στο ιεραρχικό. Έτσι, στο σχεσιακό σχεδιασμό η "ένα-προς-πολλά" σχέση ανάμεσα στους πίνακες "Ιδιοκτήτες" και "Ξενοδοχεία-Ιδιοκτήτες" αναπαρίσταται με τη χρησιμοποίηση του κοινού πεδίου "ΙδιοκτήτηςID", ενώ στον ιεραρχικό, η ίδια σχέση αναπαρίσταται με έναν δείκτη από το τμήμα "Ιδιοκτήτες" προς το τμήμα " Ξενοδοχεία-Ιδιοκτήτες".

Το κάτω μέρος του σχήματος δείχνει κάποιες από τις αρχές του ιεραρχικού μοντέλου. Φαίνεται λοιπόν ότι τα πεδία είναι ομαδοποιημένα σε τμήματα ή εγγραφές. Τα έξι πεδία που περιγράφουν έναν ιδιοκτήτη αποτελούν το τμήμα "Ιδιοκτήτες". Οι κόμβοι ενός δέντρου είναι τμήματα. Κάθε τμήμα έχει έναν μόνο πατέρα (για παράδειγμα, το τμήμα "Ιδιοκτήτες" είναι ο γονέας του τμήματος "Ξενοδοχεία-Ιδιοκτήτες"). Επίσης, κάθε τμήμα έχει ένα πεδίο "διαδοχής" το οποίο καθορίζει τη σειρά με την οποία θα διατηρηθεί κάθε εγγραφή σ' αυτό. Για παράδειγμα, το πεδίο διαδοχής του τμήματος "Ιδιοκτήτες" είναι το "ΙδιοκτήτηςID", το οποίο όπως θα δούμε στη σχεδίαση της εφαρμογής, για κάθε εγγραφή έχει σαν τιμή έναν αύξοντα αριθμό. Έτσι, η πρώτη εγγραφή είναι αυτή που έχει την τιμή "1" (στο πεδίο "ΙδιοκτήτηςID"), ακολουθεί αυτή με την τιμή "2" κ.ο.κ..

Μελετώντας μερικά παραδείγματα δέντρων, φαίνονται κάποιες επιπλέον αρχές του ιεραρχικού μοντέλου. Στην Εικόνα 2-3 φαίνεται ο (ιεραρχικός) σχεδιασμός μιας

βάσης δεδομένων που περιλαμβάνει τα αγαθά των περιοχών κάποιων κρατών καθώς και τις ημερομηνίες εξαγωγής τους στο εξωτερικό.



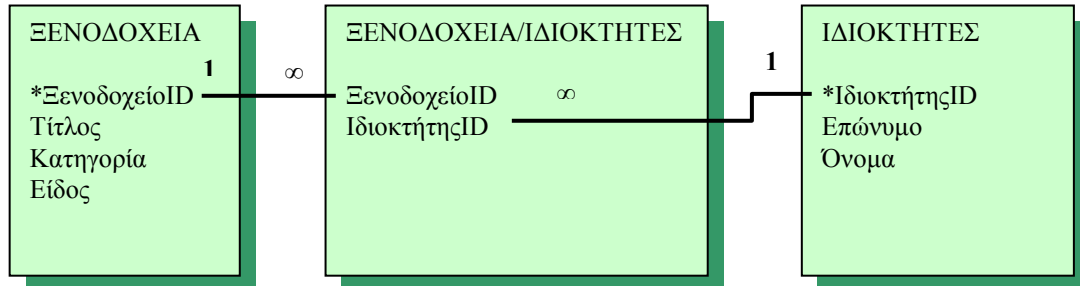
**Εικόνα 2-3:** Παραδείγματα δέντρων

Κάθε κράτος είναι η ρίζα ενός ξεχωριστού δέντρου. Έτσι, έχουμε χωριστά δέντρα για την Ινδία και τις Η.Π.Α., ενώ κάθε άλλο κράτος που τυχόν υπάρχει στο τμήμα της ρίζας, αναπαρίσταται από ένα ακόμα δέντρο. Επομένως, κάθε ιεραρχική βάση δεδομένων μπορεί να θεωρηθεί σαν μία ομάδα δέντρων. Στην Εικόνα 2-3, επεξηγείται επίσης η αρχή του "δίδυμου" (twin). Όλα τα τμήματα που έχουν τον ίδιο πατέρα ονομάζονται "δίδυμα τμήματα". Έτσι, στο τμήμα "Η.Π.Α.", τα τμήματα "Αλάσκα-Χρυσός" και "Γεωργία-Ροδάκινα" είναι δίδυμα.

Σε σύγκριση με μια σχεσιακή βάση δεδομένων, κατά το σχεδιασμό μιας ιεραρχικής βάσης δεδομένων θα πρέπει να ληφθεί υπόψη και η φυσική τους καταχώρηση (π.χ. σε ό,τι αφορά την επιλογή της μεθόδου που θα χρησιμοποιηθεί για την αναπαράσταση δέντρων), εφόσον έχει μεγάλη σημασία ο τρόπος με τον οποίο γίνεται η πρόσβαση στα δεδομένα.

Ένα άλλο πρόβλημα του ιεραρχικού μοντέλου είναι η αδυναμία του στην άμεση μοντελοποίηση των σχέσεων "πολλά-προς-πολλά", εφόσον υπάρχει ο περιορισμός ότι ένα "παιδί" μπορεί να έχει μόνο έναν "γονέα". Ο μόνος τρόπος για να αντιπαρέλθουμε αυτήν την αδυναμία είναι με τη δημιουργία δύο δέντρων. Επομένως, ένα μοντέλο δεδομένων το οποίο απαιτεί σχέση "πολλά-προς-πολλά" μεταξύ για παράδειγμα των τμημάτων "Ξενοδοχεία" και "Ιδιοκτήτες" μετατρέπεται, με τη βοήθεια ενός νέου τμήματος (π.χ. "Ξενοδοχεία/Ιδιοκτήτες") σε δύο δέντρα ή δύο σχέσεις "ένα-προς-πολλά" μεταξύ των τμημάτων "Ξενοδοχεία" - "Ξενοδοχεία/Ιδιοκτήτες" και "Ιδιοκτήτες" - "Ξενοδοχεία/Ιδιοκτήτες". Η δημιουργία όμως δύο χωριστών δέντρων

έχει ως αποτέλεσμα την καταχώρηση των δεδομένων για το "Ξενοδοχεία/Ιδιοκτήτες" δύο φορές, μια φορά για κάθε δέντρο, πράγμα που σημαίνει μεγαλύτερο χώρο αποθήκευσης δεδομένων (Εικόνα 2-4).



**Σχεσιακό Μοντέλο**



**Ιεραρχικό Μοντέλο**

**Εικόνα 2-4:** Χειρισμός σχέσης "πολλά-προς-πολλά" από το σχεσιακό και το ιεραρχικό μοντέλο.

Έχοντας πλέον μία ιδέα των κύριων χαρακτηριστικών του ιεραρχικού μοντέλου, θα ήταν χρήσιμο να γίνει μία σύγκριση αυτού με το σχεσιακό (Πίνακας 2.2-1). Η σύγκριση αυτή λαμβάνει υπόψη και τη "Γλώσσα Διαχείρισης Δεδομένων"

(DataManipulation Language ή DML) που κυρίως χρησιμοποιείται για κάθε μοντέλο: την SQL για το σχεσιακό, και την DL/I για το ιεραρχικό\* [13].

**Πίνακας 2.2-1:** Σύγκριση ιεραρχικού και σχεσιακού μοντέλου

<i>Ιεραρχικό Μοντέλο</i>	<i>Σχεσιακό Μοντέλο</i>
Πολύπλοκη θεμελιώδης δομή (δέντρο)	Απλή θεμελιώδης δομή (πίνακας)
Αναπαράσταση σχέσεων με χρήση δεικτών	Αναπαράσταση σχέσεων με χρήση κοινών πεδίων
Βέλτιστο για σχέσεις "ένα-προς-πολλά"	Χρησιμοποιείται για όλες τις σχέσεις
Χειρισμός σχέσεων "πολλά-προς-πολλά" με τη δημιουργία δύο δέντρων	Χειρισμός σχέσεων "πολλά-προς-πολλά" με δημιουργία ενός συμπληρωματικού (συνδετικού) πίνακα
Η DML (DL/I) είναι γλώσσα χαμηλού επιπέδου	Η DML (SQL) είναι γλώσσα υψηλού επιπέδου
Η DML (DL/I) είναι γλώσσα σειριακή επεξεργασίας εγγραφών (μια κάθε φορά)	Η DML (SQL) είναι γλώσσα ομαδικής επεξεργασίας εγγραφών (περισσότερες από μία κάθε φορά)
Ο προγραμματισμός είναι πολύ πιο δύσκολος επειδή ο προγραμματιστής πρέπει να προσπελάσει όλη τη δομή των δεδομένων	Ο προγραμματισμός είναι πιο εύκολος εφόσον η SQL πάντα επιστρέφει έναν μόνο πίνακα ο οποίος είναι εύκολα προσπελάσιμος
Η IBM είναι ο κύριος πωλητής	Πολλοί πωλητές
Δεν αποτελεί πρότυπο	Αποτελεί πρότυπο του American National Standards Institute (ANSI)

### 2.3 Δικτυακό Μοντέλο (Network Model)

Κάποιες σοβαρές αδυναμίες του ιεραρχικού μοντέλου (το γεγονός ότι δεν αναπαριστά αμέσως τις "πολλά-προς-πολλά" σχέσεις και δεν είναι βιομηχανικό πρότυπο) καλλιέργησαν το ενδιαφέρον για την ανάπτυξη καλύτερου μοντέλου δεδομένων και τεχνολογίας βάσεων δεδομένων. Έτσι, στα τέλη της δεκαετίας του '60, το Συνέδριο για τις Γλώσσες Συστημάτων Δεδομένων (Conference on Data Systems Languages ή CODASYL) ίδρυσε την Database Task Group (DBTG) που είχε σαν στόχο την έρευνα για ένα πρότυπο DBMS. Το 1971, η DBTG παρουσίασε το δικτυακό μοντέλο, το οποίο όμως δεν έγινε ποτέ αποδεκτό σαν πρότυπο από το ANSI για τους εξής λόγους:

1. Ήταν ένα εξαιρετικά πολύπλοκο μοντέλο χωρίς σαφές θεωρητικό υπόβαθρο.
2. Το σχεσιακό μοντέλο υπερείχε σε αξία μια και είχε πολλά πλεονεκτήματα σε σχέση με το δικτυακό, συμπεριλαμβανομένης και της προτυποποίησής του από το ANSI το 1986.

\* Η DML είναι η πιο κοινή μορφή διασύνδεσης του προγραμματιστή (και στην περίπτωση της SQL και του χρήστη) με το μοντέλο δεδομένων, κι αυτός είναι ο λόγος για τον οποίο σ'αυτήν εστιάζεται η σύγκριση.

3. Η δυνατότητα μεταφοράς των δικτυακών (όπως και των ιεραρχικών) βάσεων δεδομένων δεν ήταν πραγματοποιήσιμη, επειδή οι πωλητές εμπλούτισαν με επιπλέον στοιχεία (δυνατότητες) τις προσωπικές εκδόσεις τους με στόχο τη διαφοροποίησή τους από τους ανταγωνιστές.
4. Τέλος, η κυριαρχία της IBM στην αγορά έκανε πολύ δύσκολο στους ανταγωνιστές της να κάνουν πρόοδο σε βάρος του IMS\*.

Το δικτυακό μοντέλο είναι μία επέκταση του ιεραρχικού και επιτρέπει σε ένα παιδί να έχει κανέναν, ένα ή περισσότερους γονείς. Τα τρία δομικά στοιχεία του ορισμού δεδομένων είναι τα "αντικείμενα δεδομένων" (data-items), οι "εγγραφές ή αρχεία" (records) και τα "σύνολα" (sets). Αντικείμενο δεδομένων είναι ένα πεδίο ή χαρακτηριστικό το οποίο έχει όνομα και μορφοποίηση (για παράδειγμα, NATNAME, Character 20). Εγγραφή είναι μία συλλογή από αντικείμενα δεδομένων (για παράδειγμα, όλα τα χαρακτηριστικά που περιγράφουν κάποιο απόθεμα).

Ένα ιδιαίτερο χαρακτηριστικό του δικτυακού μοντέλου είναι το γεγονός ότι επιτρέπει επαναλήψεις. Για παράδειγμα, θα μπορούσε να χρησιμοποιηθεί μία μόνο εγγραφή για την αποθήκευση δεδομένων σχετικά με μια χώρα και τα αποθεματικά της, όπου τα δεδομένα για τα αποθεματικά μπορούν να επαναληφθούν πολλές φορές σε μία εγγραφή (Εικόνα 2-5).

Επαναλαμβανόμενη Ομάδα

NATCODE	NATNAME	EXCHRATE	SHRCODE	SHRFIRM	SHRPRICE	SHRQTY	SHRDIV
---------	---------	----------	---------	---------	----------	--------	--------

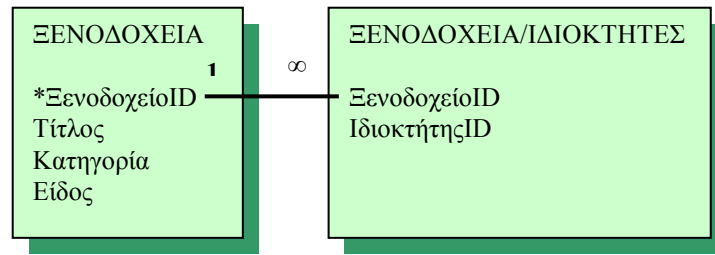
Εικόνα 2-5: Εγγραφή με επαναλαμβανόμενη ομάδα.

Παρόλο που αυτό επιτρέπεται, η χρήση επαναλαμβανόμενων ομάδων παραβιάζει τις αρχές καλής σχεδίασης βάσεων δεδομένων. Επομένως, τα δεδομένα NATION και STOCK θα έπρεπε να αποθηκευτούν σαν χωριστές οντότητες.

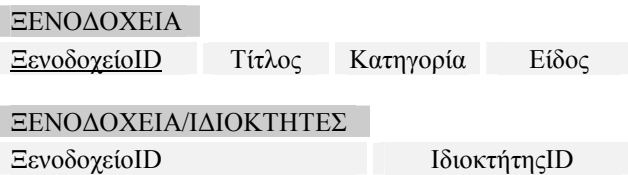
Το σύνολο, μία βασική έννοια του δικτυακού μοντέλου, είναι μία σχέση "ένα-προς-πολλά" μεταξύ εγγραφών. Έχει δύο μέρη: τον "ιδιοκτήτη" (owner) και τα "μέλη" (member). Το "ένα" της σχέσης, ο γονέας (parent), είναι ο ιδιοκτήτης του συνόλου, ενώ το "πολλά", περιέχει ένα μέλος του συνόλου. Συνεπώς, για παράδειγμα, τα

\* Η δημοφιλέστερη υλοποίηση του ιεραρχικού μοντέλου είναι το IMS (Information Management Systems) της IBM, το οποίο παρουσιάστηκε το 1968. Λόγω της κυριαρχίας της IBM στην αγορά, το IMS ήταν το πιο δημοφιλές DBMS για πάρα πολύ καιρό [7].

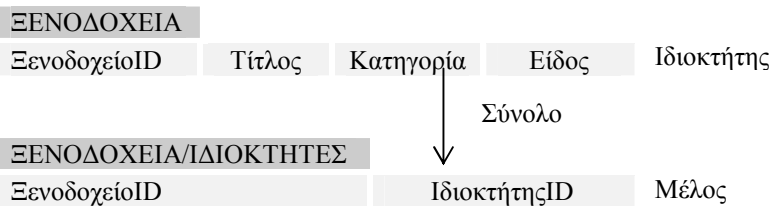
"Ξενοδοχεία" και "Ξενοδοχεία/Ιδιοκτήτες" αποτελούν ένα σύνολο (Εικόνα 2-6), όπου το "Ξενοδοχεία" είναι ο ιδιοκτήτης και το "Ξενοδοχεία/Ιδιοκτήτες" είναι το μέλος.



**Σχεσιακό Μοντέλο**



**Δικτυακό Μοντέλο**

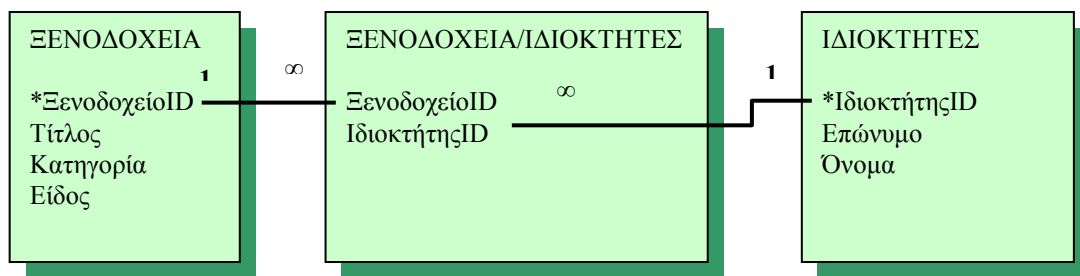


**Εικόνα 2-6:** Σύγκριση σχεσιακής και δικτυακής σχεδίασης μιας βάσης δεδομένων.

Υπάρχουν αρκετά κοινά στοιχεία μεταξύ των σχεσιακών και δικτυακών μοντέλων. Πρώτον, στο σχεσιακό μοντέλο όλες οι στήλες ενός πίνακα περιγράφουν ένα συγκεκριμένο τύπο δεδομένων (π.χ. "Κατηγορία") και στο δικτυακό όλες οι εγγραφές σε έναν ιδιοκτήτη περιγράφουν επίσης έναν τύπο δεδομένων. Δεύτερον, οι σχέσεις "ένα-προς-ένα" και "ένα-προς-πολλά" μπορούν εύκολα να αναπαρασταθούν και από τα δύο μοντέλα, παρόλο που διαφέρει η μέθοδος αναπαράστασης. Το σχεσιακό μοντέλο χρησιμοποιεί τα κοινά πεδία, και το Μέλος τει την έννοια του συνόλου και των δεικτών. Στην Εικόνα 2-6 φαίνονται η σχεσιακή και δικτυακή σχεδίαση για την ίδια βάση δεδομένων. Αξίζει να παρατηρήσουμε πώς αναπαρίσταται η σχέση "ένα-προς-πολλά". Στο σχεσιακό μοντέλο αυτό γίνεται με το κοινό πεδίο

"ΞενοδοχείοID", ενώ στο δικτυακό η σχέση αναπαρίσταται από ένα σύνολο, και εφαρμόζεται κάνοντας χρήση δεικτών.

Μια τρίτη ομοιότητα έχει να κάνει με την αναπαράσταση των σχέσεων "πολλά-προς-πολλά". Στο σχεσιακό μοντέλο δημιουργείται ένας συνδετικός πίνακας, ενώ στο δικτυακό μοντέλο δημιουργείται ένα μέλος το οποίο εμφανίζεται σε δύο σύνολα (Εικόνα 2-7). Στο σχεσιακό μοντέλο το συνδετικό στοιχείο είναι ο πίνακας "Ξενοδοχεία/Ιδιοκτήτες", ενώ στο δικτυακό, το "Ξενοδοχεία/Ιδιοκτήτες" γίνεται μέλος και εμφανίζεται στα δύο σύνολα που έχουν ως ιδιοκτήτες τα "Ξενοδοχεία" και "Ιδιοκτήτες".



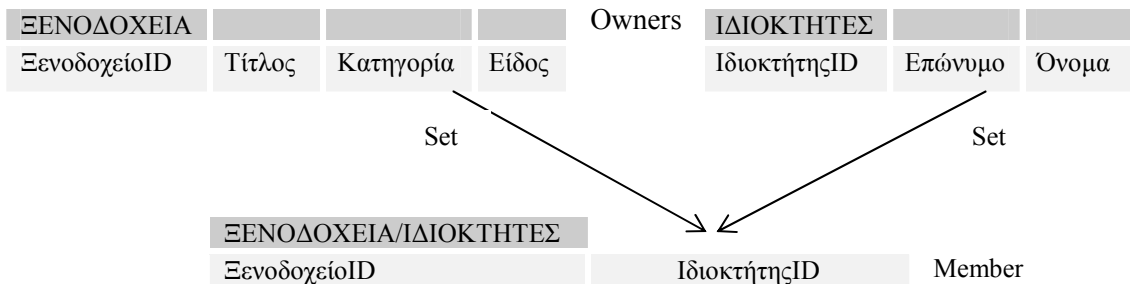
*Σχεσιακή Σχεδίαση*

ΞΕΝΟΔΟΧΕΙΑ			
<u>ΞενοδοχείοID</u>	Τίτλος	Κατηγορία	Είδος

ΙΔΙΟΚΤΗΤΕΣ		
<u>ΙδιοκτήτηςID</u>	Επώνυμο	Όνομα

ΞΕΝΟΔΟΧΕΙΑ/ΙΔΙΟΚΤΗΤΕΣ	
ΞενοδοχείοID	ΙδιοκτήτηςID

*Δικτυακή Σχεδίαση*



**Εικόνα 2-7:** Σχεσιακή και δικτυακή αναπαράσταση μιας σχέσης "πολλά-προς-πολλά".



Κλείνοντας την αναφορά στο δικτυακό μοντέλο, θα ήταν χρήσιμο να γίνει (όπως έγινε και στην περίπτωση του ιεραρχικού μοντέλου) μία σύγκριση ανάμεσα σ' αυτό και στο σχεσιακό (Πίνακας 2.3-1). Από τη σύγκριση αυτή σε συνδιασμό με όλα όσα έχουν αναφερθεί για το ιεραρχικό και δικτυακό μοντέλο, φαίνεται πως το σχεσιακό μοντέλο είναι σαφέστατα η καλύτερη επιλογή ενός προγραμματιστή συστημάτων διαχείρισης βάσεων δεδομένων [13].

**Πίνακας 2.3-1:** Σύγκριση του δικτυακού με το σχεσιακό μοντέλο

<i>Δικτυακό Μοντέλο</i>	<i>Σχεσιακό Μοντέλο</i>
Πολύπλοκη θεμελιώδης δομή (δίκτυο)	Απλή θεμελιώδης δομή (πίνακας)
Αναπαράσταση σχέσης "ένα-προς-πολλά" με τη χρήση συνόλων (sets)	Αναπαράσταση σχέσης "ένα-προς-πολλά" με χρήση κοινού πεδίου
Αναπαράσταση σχέσεων "πολλά-προς-πολλά" με τη χρήση ενός μέλους (member) που περιέχεται σε δύο σύνολα	Αναπαράσταση σχέσης "πολλά-προς-πολλά" με δημιουργία ενός συμπληρωματικού (συνδετικού) πίνακα
Υποστήριξη επαναλαμβανόμενων ομάδων	Μη υποστήριξη των επαναλαμβανόμενων πεδίων (ένα επαναλαμβανόμενο πεδίο αντιτίθεται στις αρχές του σχεσιακού μοντέλου)
Η DML είναι γλώσσα χαμηλού επιπέδου	Η DML (SQL) είναι γλώσσα υψηλού επιπέδου
Η DML είναι γλώσσα σειριακής επεξεργασίας εγγραφών	Η DML (SQL) είναι γλώσσα ομαδικής επεξεργασίας εγγραφών (περισσότερες από μία τη φορά)
Ο προγραμματισμός είναι πολύ δύσκολος επειδή ο προγραμματιστής πρέπει να προσπελάσει όλη τη δομή των δεδομένων	Ο προγραμματισμός είναι πιο εύκολος επειδή η SQL πάντα επιστρέφει έναν μόνο πίνακα ο οποίος είναι πιο εύκολα προσπελάσιμος
Πολλοί πωλητές	Πολλοί πωλητές
Μερικώς προτυποποιημένο	Πρότυπο του ANSI

## 2.4 Βάσεις Δεδομένων: Κύριες Λειτουργίες - Λόγοι για να τις χρησιμοποιήσουμε

Κλείνοντας τη σύντομη αναφορά μας στις σχεσιακές βάσεις δεδομένων, θα ήταν καλό να αναφέρουμε τις "κύριες λειτουργίες μιας βάσης δεδομένων", καθώς και τους "λόγους για να αρχίσουμε να χρησιμοποιούμε μία βάση δεδομένων". Στη συνέχεια, αφού γίνει μία σύντομη εισαγωγή στην "Microsoft Access", στην οποία φαίνονται καθαρά οι λόγοι για τους οποίους χρησιμοποιήθηκε, θα είμαστε πλέον σε θέση να προχωρήσουμε στην ανάλυση της εφαρμογής.

### A) "Κύριες Λειτουργίες μιας Βάσης Δεδομένων"

- **Ορισμός δεδομένων** (*Data definition*). Μπορούμε να ορίζουμε τι δεδομένα θα αποθηκεύονται στη βάση δεδομένων μας, τον τύπο τους (για παράδειγμα, αριθμοί ή χαρακτήρες), και τον τρόπο με τον οποίο θα συσχετίζονται μεταξύ τους. Σε μερικές περιπτώσεις, μπορούμε επίσης να ορίζουμε τον τρόπο μορφοποίησης των δεδομένων και τον τρόπο ελέγχου της εγκυρότητάς τους.
- **Χειρισμός δεδομένων** (*Data manipulation*). Μπορούμε να επεξεργαζόμαστε τα δεδομένα με πολλούς τρόπους. Μπορούμε να επιλέγουμε ποιά πεδία θέλουμε, και να φιλτράρουμε και να ταξινομούμε τα δεδομένα. Μπορούμε επίσης να ενώνουμε δεδομένα με άλλες συσχετισμένες πληροφορίες και να συνοψίζουμε τα δεδομένα (να παίρνουμε σύνολα και άλλα στατιστικά στοιχεία).
- **Έλεγχος δεδομένων** (*Data control*). Μπορούμε να ορίζουμε σε ποιόν θα επιτρέπεται η ανάγνωση, η ενημέρωση ή η εισαγωγή των δεδομένων. Σε πολλές περιπτώσεις, μπορούμε επίσης να ορίζουμε τον τρόπο με τον οποίο θα μοιράζονται και θα ενημερώνονται τα δεδομένα από πολλούς διαφορετικούς χρήστες ταυτόχρονα.

### B) "Λόγοι για να Αρχίσουμε να Χρησιμοποιούμε μία Βάση Δεδομένων"

- Έχουμε πολλά ξεχωριστά αρχεία ή πάρα πολλά δεδομένα σε ξεχωριστά αρχεία. Αυτό δυσκολεύει τη διαχείρισή τους. Επίσης, τα δεδομένα μπορεί να ξεπερνούν τα όρια του λογισμικού ή τη χωρητικότητα της μνήμης του συστήματος.
- Έχουμε πολλές διαφορετικές χρήσεις των δεδομένων - αναλυτικές κινήσεις (π.χ. τιμολόγια), ανάλυση συνοπτικών στοιχείων (π.χ. σύνοψη τριμηνιαίων πωλήσεων), και σενάρια ανάλυσης υποθέσεων ("what if"). Επομένως, χρειάζεται να μπορούμε

να εξετάζουμε τα δεδομένα με πολλούς διαφορετικούς τρόπους, και δυσκολευόμαστε να δημιουργούμε νέες "απόψεις" τους.

- Υπάρχει ανάγκη για μερισμό των δεδομένων. Για παράδειγμα, η εισαγωγή, η ενημέρωση και η ανάλυση των δεδομένων γίνεται από πολλά άτομα. Ενώ μόνο ένα άτομο μπορεί να ενημερώνει ένα λογιστικό φύλλο ή έγγραφο επεξεργαστή κειμένου, έναν πίνακα μιας βάσης δεδομένων μπορούν να τον μοιράζονται και να τον ενημερώνουν πολλά άτομα. Επίσης, η βάση δεδομένων εγγυάται ότι τα άτομα που διαβάζουν τα δεδομένα θα βλέπουν μόνο τις ενημερώσεις που έχουν ολοκληρωθεί.
- Τέλος, πρέπει να ελέγχουμε τα δεδομένα επειδή σ' αυτά έχουν πρόσβαση διαφορετικοί χρήστες. Έτσι, πρέπει να εξασφαλίζουμε την πρόσβαση σ' αυτά, να ελέγχουμε τις τιμές τους και να εγγυόμαστε τη σταθερότητά τους [12].

### 3. ACCESS: ΣΥΝΟΠΤΙΚΗ ΠΕΡΙΓΡΑΦΗ

Τα προγράμματα για βάσεις δεδομένων σε Η/Υ χρησιμοποιούνται για να διαχειρίζονται μεγάλες λίστες πληροφοριών, σχετικά με πελάτες, υπαλλήλους, γεγονότα και σχεδόν οτιδήποτε άλλο θελήσουμε. Μια απλή βάση δεδομένων αποθηκεύει το ίδιο είδος πληροφοριών που θα αποθηκεύαμε με το χέρι σε ένα σύστημα με κάρτες. Τα σχεσιακά προγράμματα βάσεων δεδομένων όπως η Microsoft Access, μας καθιστούν ικανούς να δημιουργούμε βάσεις δεδομένων που είναι πολύ πιο περίπλοκες από ένα αρχείο με κάρτες.

Μια βάση δεδομένων προσφέρει πολλά πλεονεκτήματα σε σχέση με ένα μη αυτόματο σύστημα αρχειοθέτησης. Αντί να ανατρέχουμε σε ένα σωρό από κάρτες ή χαρτιά για να βρούμε μία πληροφορία, μπορούμε γρήγορα και εύκολα να αναζητήσουμε και να βρούμε τις πληροφορίες που χρειαζόμαστε. Μπορούμε επίσης να ταξινομούμε τα δεδομένα μας, να τα φιλτράρουμε (για παράδειγμα, να εμφανίσουμε όλα τα ταξιδιωτικά γραφεία που βρίσκονται στον Πλατανιά ή στα Χανιά) και να δημιουργούμε όμορφες και κυρίως χρήσιμες αναφορές, χρησιμοποιώντας το πρόγραμμα για βάσεις δεδομένων που διαθέτουμε [11]. Η Microsoft Access για Windows '95 είναι ένα από τα πιο δημοφιλή από αυτά τα προγράμματα και ένα μόνο μέρος από τη γενική στρατηγική προϊόντων διαχείρισης δεδομένων της Microsoft Corporation. Πρόκειται για ένα πρόγραμμα με πολλές και ισχυρές δυνατότητες. Διαθέτει ένα δικό της σύστημα αποθήκευσης δεδομένων, και όπως όλες οι σχεσιακές βάσεις δεδομένων που σέβονται τον εαυτό τους, επιτρέπει την εύκολη σύνδεση των συσχετισμένων πληροφοριών, ενώ συνεργάζεται τέλεια με τις άλλες εφαρμογές του πακέτου Microsoft Office: το Microsoft Excel, το Microsoft Word, το Power Point και το Microsoft Mail. Διαθέτει επίσης ένα πολύ εξελιγμένο σύστημα ανάπτυξης εφαρμογών για το λειτουργικό σύστημα των Microsoft Windows, το οποίο κάνει εκτεταμένη χρήση των δεδομένων, όποια κι αν είναι η πηγή τους, έτσι ώστε να βοηθά το χρήστη να δημιουργεί γρήγορα και εύκολα εφαρμογές. Για την ακρίβεια, μπορούμε να δημιουργούμε απλές εφαρμογές, ορίζοντας (κυριολεκτικά ζωγραφίζοντας στην οθόνη) φόρμες και αναφορές που βασίζονται στα δεδομένα μας, και συνδέοντας αυτές μεταξύ τους με λίγες απλές μακροεντολές ή με μερικές εντολές της Microsoft Access Basic (δεν χρειάζεται η δημιουργία πολύπλοκου κώδικα με την κλασική προγραμματιστική έννοια).

Η Microsoft προώθησε αρχικά την Access με τις προδιαγραφές της Εικόνα 3-1, όπου φαίνεται ότι η Access μπορεί να χρησιμοποιηθεί σε όλα τα επίπεδα. Αρχίζοντας από το χαμηλότερο επίπεδο της ιεραρχίας και προχωρώντας προς τα πάνω, βλέπουμε πρώτα τα εργαλεία\*. Τα εργαλεία παρέχουν στον χρήστη τη δυνατότητα να δημιουργεί εύκολα φόρμες και αναφορές. Μπορούμε να επεξεργαστούμε απλά δεδομένα και να χρησιμοποιήσουμε παραστάσεις πιστοποίησης δεδομένων ή να παρουσιάσουμε αριθμούς με σύμβολα συναλλαγματικής μονάδας. Οι μακροεντολές επιτρέπουν την αυτοματοποίηση χωρίς προγραμματισμό, ενώ ο κώδικας ABC (Access Basic Code) επιτρέπει στο χρήστη να προγραμματίσει σύνθετες εργασίες. Τέλος, ένας προγραμματιστής C μπορεί να γράψει εφαρμογές που να συνδέονται με άλλα προγράμματα και πηγές δεδομένων.



**Εικόνα 3-1**

Οι βάσεις δεδομένων έχουν πολλές μορφές. Μερικές προέρχονται από άλλα προγράμματα διαχείρισης βάσεων δεδομένων, όπως για παράδειγμα τα dBase IV, Paradox, FoxPro, Btrieve, Oracle και Ingres και μερικές άλλες από επεξεργαστές κειμένου, υπολογιστές μεγάλων συστημάτων ή άγνωστες πηγές μέσα από το Internet. Το όνομα Access (πρόσβαση-προσπέλαση), έχει επιλεγεί για να υποδεικνύει ότι η Access παρέχει δυναμικά και χρήσιμα εργαλεία τα οποία βοηθούν το χρήστη να αποκτά πρόσβαση σε δεδομένα πολλών μορφών. Είναι επίσης δυνατόν να χρησιμοποιηθεί σαν διασύνδεση χρήστη (user interface) σε βάσεις δεδομένων μεγάλων εταιριών που υπάρχουν σε υπολογιστές μεγάλων συστημάτων [10].

Όλα αυτά λοιπόν, μαζί με τη δυνατότητα δημιουργίας των αντικειμένων που περιγράφονται στη συνέχεια και αναλύονται διεξοδικά σε επόμενα κεφάλαια, κάνουν την Access ένα ιδιαίτερα χρήσιμο και ευέλικτο πρόγραμμα διαχείρισης βάσεων δεδομένων.

---

\* Τα εργαλεία μπορεί να είναι κουμπιά διαταγής, πλαίσια κειμένου, πλαίσια καταλόγου κ.α.

### 3.1 Αντικείμενα Βάσεων Δεδομένων (Database Objects)

Τα αντικείμενα (objects) των βάσεων δεδομένων περιγράφουν τους διάφορους τρόπους εμφάνισης και διαχείρισης των δεδομένων σε μια βάση δεδομένων. Έτσι, με την Access μπορούμε να δημιουργήσουμε τα ακόλουθα αντικείμενα :

⇒ **Πίνακες** (Tables)

Χρησιμοποιούνται στην αποθήκευση δεδομένων.

⇒ **Ερωτήματα** (Queries)

Χρησιμοποιούνται στην υποβολή ερωτήσεων σε μια βάση δεδομένων. Η απάντηση συνήθως είναι μια ομάδα εγγραφών. Για παράδειγμα, μπορούμε να ρωτήσουμε ποιά Ξενοδοχεία έχουν τιμή δίκλινου δωματίου που κυμαίνεται από 12000δρχ έως 27000δρχ.

⇒ **Φόρμες** (Forms)

Χρησιμοποιούνται στην καταχώρηση και εμφάνιση των δεδομένων.

⇒ **Αναφορές** (Reports)

Χρησιμοποιούνται στην περιληπτική παρουσίαση και την εκτύπωση των δεδομένων.

⇒ **Μακροεντολές** (Macros)

Χρησιμοποιούνται στην αυτοματοποίηση των πιο συνηθισμένων εργασιών.

⇒ **Λειτουργικές Μονάδες** (Modules)

Χρησιμοποιούνται από τους πιο εξειδικευμένους χρήστες της Access για να γράφουν και να τροποποιούν τα προγράμματα της Access με τη βοήθεια της γλώσσας προγραμματισμού Visual Basic [11].

### 3.1.1 Πίνακες

Οι πίνακες είναι τα πρωταρχικά δυναμικά στοιχεία των περισσότερων βάσεων δεδομένων και χρησιμεύουν στην καταχώρηση, ανάκτηση και γενικά στην επεξεργασία των πληροφοριών που σχετίζονται με τη βάση δεδομένων μας.

Ένας πίνακας (table) είναι ένα σύνολο σχετιζόμενων πληροφοριών, όπως μία λίστα με τα ονόματα των γνωστών μας, τις διευθύνσεις και τους αριθμούς τηλεφώνου τους. Το κάθε σύνολο σχετιζόμενων στοιχείων στον πίνακα ονομάζεται εγγραφή (record). Για παράδειγμα, το όνομα πατέρα, το όνομα μητέρας, το έτος γέννησης, ο τόπος γέννησης και η διεύθυνση κατοικίας του Νικολακάκη Βασίλη, είναι μία εγγραφή. Σε έναν πίνακα της Access, οι εγγραφές μπορούν να θεωρηθούν σαν σειρές (rows) που έχουν αποθηκευτεί στον πίνακα.

Η κάθε ξεχωριστή πληροφορία που αποθηκεύεται σε μια εγγραφή ονομάζεται πεδίο (field). Έτσι, το Επίθετο και το Όνομα είναι παραδείγματα πεδίων. Σε έναν πίνακα της Access, το κάθε πεδίο μπορεί να θεωρηθεί σαν μία στήλη (column) του πίνακα. Για κάθε πεδίο που προσθέτουμε σε έναν πίνακα, επιλέγουμε τον τύπο δεδομένων του (αριθμός, σημειώσεις, κείμενο, ημερομηνία, αυτόματη αρίθμηση, Ναι/Όχι, λίστα αναζήτησης, χρηματικό ποσό), καθορίζοντας έτσι το είδος των δεδομένων που θα δέχεται σαν τιμές και ανάλογα με τον τύπο ορίζουμε τις ιδιότητές του, ενώ παράλληλα μπορούμε να δώσουμε και μία περιγραφή (description) του πεδίου αυτού [11].

### 3.1.2 Ερωτήματα

Ο βασικότερος λόγος για τον οποίο ξοδεύουμε το χρόνο μας καταχωρώντας όλα τα δεδομένα μας σε μία βάση δεδομένων, είναι για να μπορούμε εύκολα να βρίσκουμε και να δουλεύουμε με τις πληροφορίες. Στη βάση δεδομένων για την Τουριστική Αστυνομία Χανίων, θελησαμε κάποια στιγμή, για παράδειγμα, να εμφανίσουμε όλα τα Ταξιδιωτικά Γραφεία που υπάρχουν στα Χανιά. Για να γίνει κάτι τέτοιο θα πρέπει να δημιουργήσουμε ένα ερώτημα.

Ένα ερώτημα (query) θέτει ένα ερωτηματικό στη βάση δεδομένων, όπως το "Ποιές εγγραφές έχουν τα Χανιά στο πεδίο Πόλη;" και μετά ξεχωρίζει αυτές τις εγγραφές από τη βάση δεδομένων και τις τοποθετεί σε ένα υποσύνολο εγγραφών. Αν το ερώτημα απλώς επιλέγει ή ταξινομεί τα δεδομένα, το υποσύνολο έχει το ειδικό όνομα

"Δυναμικό Σύνολο" (dynaset). Μπορούμε μετά να δουλέψουμε μόνο με αυτές τις εγγραφές που έχουν επιλεγεί από το ερώτημα που θέσαμε.

Μπορούμε να δημιουργήσουμε ερωτήματα που εμφανίζουν όλες τις εγγραφές αλλά μόνο συγκεκριμένα πεδία, όπως για παράδειγμα τα πεδία Επίθετο και Κωδικός Ιδιοκτήτη. Μπορούμε επίσης να δημιουργήσουμε ένα ερώτημα που αναζητά συγκεκριμένες εγγραφές μέσα σε ένα πεδίο, όπως το παράδειγμα που δόθηκε παραπάνω με τα Χανιά, ή να δημιουργήσουμε περισσότερο περίπλοκα ερωτήματα, όπως γίνεται στα Ξενοδοχεία, όπου μπορούμε να ρωτήσουμε τη βάση δεδομένων να εμφανίσει όλα τα Ξενοδοχεία που ανήκουν στις κατηγορίες "Α" ή "Β", βρίσκονται στον Πλατανιά, έχουν πισίνα, pool bar, breakfast και parking, και η τιμή του δίκλινου δωματίου ανήκει στο διάστημα [ 15000δρχ , 35000δρχ ] [11].

### 3.1.3 Φόρμες

Όταν καταχωρούμε εγγραφές σε έναν πίνακα, κάθε εγγραφή εμφανίζεται σε μια γραμμή και μπορούμε να δούμε όλες τις εγγραφές. Αν ο πίνακας έχει πολλά πεδία, ίσως να μην μπορούμε να τον δούμε ολόκληρο στην οθόνη μας και να συναντάμε δυσκολίες όταν θέλουμε να δούμε όλα τα πεδία μιας συγκεκριμένης εγγραφής που μας ενδιαφέρει. Η Access μας προσφέρει μια εναλλακτική λύση με τη χρήση της φόρμας (form), για να βλέπουμε και να καταχωρούμε δεδομένα. Η χρήση μιας φόρμας προσφέρει τα εξής πλεονεκτήματα :

- Μπορούμε να επιλέξουμε τα πεδία που θέλουμε να συμπεριλάβουμε σε μία φόρμα και να τα τοποθετήσουμε με τη σειρά που επιθυμούμε.
- Μπορούμε να επιλέξουμε την προβολή μίας μόνο εγγραφής τη φορά, κάτι που μας διευκολύνει στην καλύτερη συγκέντρωσή μας πάνω στη συγκεκριμένη εγγραφή.
- Μπορούμε να εμπλουτίσουμε τη φόρμα με γραφικά ώστε να γίνεται πιο εμφανίσιμη.

Η Access παρέχει πολλά εργαλεία για τη δημιουργία μιας φόρμας. Έτσι, μπορούμε να δημιουργήσουμε μια Αυτόματη Φόρμα (AutoForm), να χρησιμοποιήσουμε τους



Οδηγούς Φορμών (Form Wizards) για να φτιάξουμε κάποιους άλλους συνηθισμένους τύπους φορμών (όπως για παράδειγμα μονόστηλη ή πινακοποιημένη), ή τέλος να κατασκευάσουμε μία φόρμα εξ' αρχής, όταν οι Οδηγοί Φόρμας δεν δημιουργούν ακριβώς τη φόρμα που θέλουμε [11].

### 3.1.4 Αναφορές

Σκοπός των πληροφοριών μιας βάσης δεδομένων δεν είναι μόνο η εμφάνισή τους στην οθόνη του υπολογιστή μας, αλλά και η δυνατότητα εκτύπωσής τους. Αυτό γίνεται με τη βοήθεια των αναφορών (reports) οι οποίες μπορούν να είναι τόσο απλές όσο ένα εκτυπωμένο αντίγραφο ενός πίνακα. Μπορούν επίσης να είναι περίπλοκα έγγραφα που ομαδοποιούν και αθροίζουν πληροφορίες.

Όπως και στις φόρμες, για τη δημιουργία μιας αναφοράς η Access παρέχει πολλές δυνατότητες. Έτσι, μπορούμε να δημιουργήσουμε μια Αυτόματη Αναφορά (AutoReport), να χρησιμοποιήσουμε τους Οδηγούς Αναφορών (Report Wizards) για να φτιάξουμε κάποιους άλλους συνηθισμένους τύπους αναφορών (όπως για παράδειγμα μονόστηλη ή ταχυδρομικές ετικέτες), ή να κατασκευάσουμε μια αναφορά εξ' αρχής, δίνοντάς της την εμφάνιση που θέλουμε [11].

### 3.1.5 Μακροεντολές - προγραμματισμός χωρίς προγραμματισμό

Στην Access μπορούμε να ορίζουμε μια μακροεντολή για να εκτελούμε οποιαδήποτε σχεδόν εργασία, την οποία θα έπρεπε διαφορετικά να ενεργοποιούμε με το πληκτρολόγιο ή το ποντίκι. Η μοναδική ισχύς των μακροεντολών της Access βρίσκεται στην ικανότητά τους να αυτοματοποιούν αποκρίσεις πολλών τύπων συμβάντων. Το συμβάν (event) μπορεί να είναι κάποια αλλαγή στα δεδομένα, το άνοιγμα ή το κλείσιμο μιας φόρμας ή αναφοράς, το πάτημα του πλήκτρου "Enter", ή ακόμα και η εναλλαγή από το ένα χειριστήριο στο άλλο (η ενεργοποίηση ενός νέου χειριστηρίου). Μέσα σε μια μακροεντολή μπορούμε να συμπεριλάβουμε πολλές ενέργειες και να ορίσουμε συνθήκες ελέγχου για να εκτελούνται κάποιες απ' αυτές, ανάλογα με κάποιες τιμές στις φόρμες ή στις αναφορές μας. Υπάρχουν σχεδόν 60 μακροεντολές για διαχείριση δεδομένων, δημιουργία διαλογικών πλαισίων, άνοιγμα φορμών και αναφορών, και, γενικώς, αυτοματποίηση οποιασδήποτε εργασίας [12].

### 3.1.6 Access Basic

Η Access είναι ένα σοβαρό περιβάλλον ανάπτυξης με μια πλήρη γλώσσα προγραμματισμού. Η Access Basic είναι ένα μοντέλο προγραμματισμού προσανατολι-

σμένου στη δραστηριότητα, και μία ισχυρή και δομημένη γλώσσα προγραμματισμού. Είναι τέλος πλήρως επεκτάσιμη και διαθέτει ρουτίνες κλήσης σε οποιαδήποτε δυναμική βιβλιοθήκη του λειτουργικού περιβάλλοντος Windows. Τέλος, ένα πλήρες περιβάλλον ανάπτυξης IDE (Integrated Development Environment) επιτρέπει τη σύνταξη και τη διόρθωση του κώδικα σε πολλά παράθυρα, αυτόματο έλεγχο και εκτέλεση βήμα-προς-βήμα [10].

#### 4. ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΤΟΥ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

Η Τουριστική Αστυνομία Χανίων είναι μία υπηρεσία η οποία εκτός των άλλων παρέχει πληροφορίες σχετικά με τις τουριστικές επιχειρήσεις που αναφέρθηκαν στην εισαγωγή και λειτουργούν στο νομό Χανίων. Οι πληροφορίες αυτές κρατούνται στη βιβλιοθήκη της υπηρεσίας, σε συγκεκριμένα αρχεία ανάλογα με το είδος της επιχείρησης. Το πρόβλημα που έπρεπε να λυθεί, δεν ήταν τόσο η εισαγωγή νέων στοιχείων, όσο το φιλτράρισμα των ήδη υπαρχόντων.

Σε ορισμένες περιπτώσεις, το πρόβλημα αυτό λυνόταν από την εμπειρία των εργαζομένων στην υπηρεσία. Αυτό γινόταν, όταν για παράδειγμα ένας επισκέπτης του νομού ενδιαφερόταν απλά για ένα φτηνό ξενοδοχείο που να βρίσκεται στην πόλη Καλύβες. Τότε, κάποιος εργαζόμενος που είχε γνώση των τιμών των ξενοδοχείων στην πόλη αυτή, έδινε στον επισκέπτη τις πληροφορίες που ήξερε χωρίς να χρειαστεί να γίνει κάποια διερεύνηση στα αρχεία της υπηρεσίας.

Υπήρχαν όμως και περιπτώσεις όπου ο επισκέπτης ήταν πολύ πιο απαιτητικός, υπό την έννοια ότι οι πληροφορίες για τις οποίες ενδιαφερόταν απαιτούσαν μία ιδιαίτερα χρονοβόρα διαδικασία διερεύνησης. Αυτό γινόταν, όταν για παράδειγμα ενδιαφερόταν για ένα ξενοδοχείο Α' ή Β' κατηγορίας που να βρίσκεται στα Χανιά, να έχει πισίνα, pool bar, parking και breakfast και η τιμή δίκλινου δωματίου να κυμαίνεται από 15.000 έως 25.000δρχ. Τότε, οι εργαζόμενοι ήταν υποχρεωμένοι να σπαταλήσουν πάρα πολύ χρόνο εξετάζοντας αναλυτικά τα αρχεία της υπηρεσίας προκειμένου να δώσουν στον επισκέπτη τις πληροφορίες που ήθελε.

Στόχος λοιπόν της διπλωματικής εργασίας είναι η ανάπτυξη μιας φιλικής προς το χρήστη εφαρμογής η οποία θα επιτυγχάνει τα ακόλουθα:

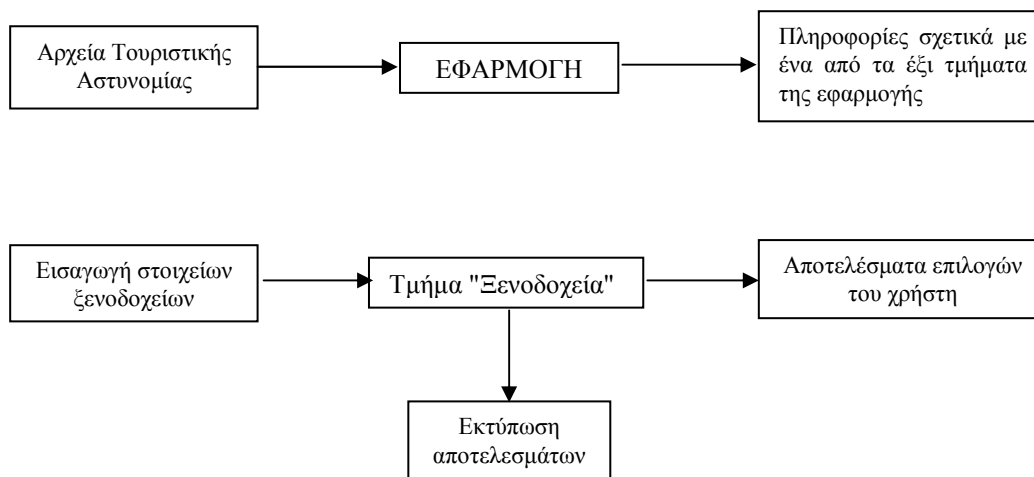
1. Εύκολη και γρήγορη καταχώρηση δεδομένων
2. Δυνατότητα φιλτραρίσματος στις τιμές των πεδίων "Επώνυμο" Ιδιοκτήτη/Υπεύθυνου, "ΑΦ\_Αρχείου" (Αριθμός Φακέλλου Αρχείου\*) και "Τίτλος" (ονομασία επιχείρησης) στο καθένα χωριστά

---

\* Επιθυμία της υπηρεσίας δεν ήταν η κατάργηση των αρχείων της αλλά η συνύπαρξη αυτής με την εφαρμογή.

3. Δυνατότητα φιλτραρίσματος στις τιμές των πεδίων "Πόλη", "Είδος", "Κατηγορία", "Τιμή" (για την τιμή, όπως θα δειχθεί στη συνέχεια, υπάρχει η δυνατότητα επιλογής "Ελάχιστης" και "Μέγιστης" τιμής), "Πισίνα", "Pool bar", "Parking", "Breakfast", και "Εστιατόριο", ταυτόχρονα
4. Εκτύπωση των αποτελεσμάτων που προκύπτουν από τα φίλτρα που εφαρμόζει ο χρήστης (με τις επιλογές που πραγματοποιεί)
5. Όμορφη και πρακτική εμφάνιση φορμών και αναφορών, και
6. Πρόβλεψη μελλοντικών αναγκών της υπηρεσίας.

Το τελευταίο επιτυγχάνεται, όπως θα δειχθεί στη συνέχεια, με την προσθήκη κάποιων πεδίων, όπως "Τιμή μονόκλινου δωματίου", "Αριθμός μονόκλινων δωματίων", "Εστιατόριο", για τα οποία η υπηρεσία δεν έχει στοιχεία στην παρούσα φάση, αλλά δεν αποκλείεται να έχει στο μέλλον. Όσο αφορά την επίτευξη ή όχι των άλλων πέντε στόχων, η απάντηση δίνεται στο κεφάλαιο "Λειτουργία της εφαρμογής".



**Εικόνα 4-4-1:** Σχεδίαση της εφαρμογής

#### **4.1 ΑΝΑΛΥΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΤΜΗΜΑΤΟΣ "ΞΕΝΟΔΟΧΕΙΑ"**

Το πρώτο από τα έξι τμήματα της εφαρμογής που κατασκευάστηκε είναι αυτό των "Ξενοδοχείων", το οποίο όπως έχουμε ήδη αναφέρει, εκτός των Ξενοδοχείων περιλαμβάνει και κάποια άλλα είδη επιχειρήσεων που έχουν άδεια λειτουργίας Ξενοδοχείου: τα "Διαμερίσματα", τα "Δωμάτια", τους "Ξενώνες" και τα "Οικοτροφεία". Και τα έξι τμήματα μοιάζουν αρκετά μεταξύ τους, μιας και σε όλα επεξεργαζόμαστε τα ίδια σχεδόν στοιχεία (διευθύνσεις, προσφερόμενες ανέσεις, τιμές, στοιχεία Ιδιοκτήτη, στοιχεία Υπευθύνου), τα Ξενοδοχεία όμως είναι το πιο περίπλοκο διότι περιλαμβάνει τα περισσότερα δεδομένα και με βάση αυτά γίνονται οι πιο πολλές επιλογές.

Το πρώτο βήμα για την κατασκευή των "Ξενοδοχείων", ήταν, όπως και για κάθε άλλη βάση, η δημιουργία των πινάκων.

##### **4.1.1 Πίνακες Ξενοδοχείων**

Τα δεδομένα των "Ξενοδοχείων" που αρχικά θέλαμε να επεξεργαστούμε, είναι τα εξής :

" Τίτλος, Είδος, Κατηγορία, Τιμή (δίκλινο δωμάτιο), Τηλέφωνο, FAX, Διεύθυνση Επιχείρησης, Αριθμός Φορολογικού Μητρώου, Αριθμός Φακέλου Αρχείου, Δωμάτια (δίκλινα), Ανέσεις (Πισίνα, Pool Bar, Parking, Breakfast, Εστιατόριο) και στοιχεία Ιδιοκτήτη/Υπευθύνου (Επίθετο, Όνομα, Όνομα Πατέρα, Όνομα Μητέρας, Έτος Γέννησης, Τόπος Γέννησης και Διεύθυνση Κατοικίας) " . Πρόκειται για τα στοιχεία που ενδιέφεραν την Τουριστική Αστυνομία.

Προκειμένου όμως να προβλέψουμε μελλοντικές ανάγκες της, προσθέσαμε τον αριθμό και τις τιμές μονόκλινων, τρίκλινων, τετράκλινων δωματίων, σουιτών και bangalows, καθώς και μία δεύτερη τηλεφωνική γραμμή. Επίσης, για την καλύτερη εισαγωγή και διαχείριση των δεδομένων αναλύσαμε όλες τις διευθύνσεις (Επιχείρησης, Ιδιοκτήτη, Υπευθύνου) σε "Πόλη", "Οδό" και "Αριθμό", ενώ τέλος προσθέσαμε και ένα πεδίο που ονομάσαμε "Photo", το οποίο είναι δυνατό να χρησιμεύσει μελλοντικά στην εμφάνιση φωτογραφιών των επιχειρήσεων, και ένα άλλο που ονομάσαμε "Video" για την προβολή video. Παρόμοιες αλλαγές έχουν γίνει φυσικά, όπου ήταν δυνατόν, και στα υπόλοιπα πέντε τμήματα της βάσης. Έτσι,

χωρίσαμε τελικά τα δεδομένα μας στους εξής πίνακες : "Ξενοδοχεία", "Στοιχεία Επιχείρησης", "Ανέσεις", "Ιδιοκτήτες" & "Υπεύθυνοι".

#### **4.1.1.1 "Ξενοδοχεία"**

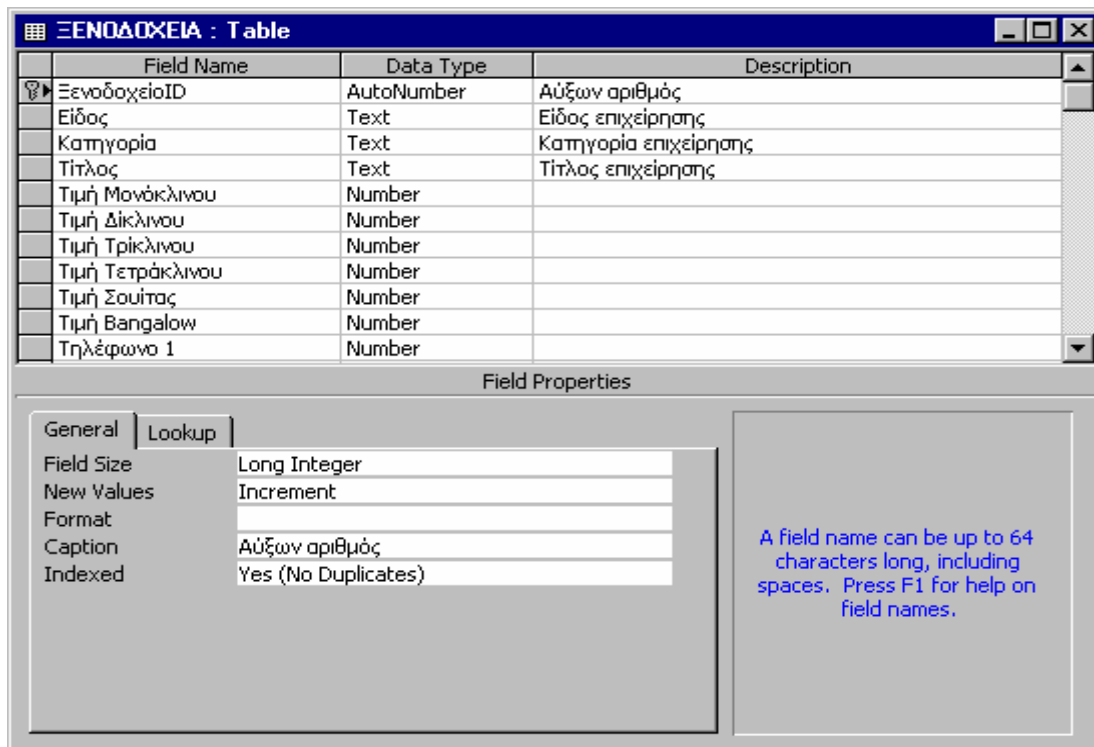
Τα πεδία που περιέχει ο πίνακας "Ξενοδοχεία" είναι ο Τίτλος, το Είδος, η Κατηγορία, τα Τηλέφωνο1, Τηλέφωνο2 και FAX, ο αριθμός και η τιμή μονόκλιων, δίκλιων, τρίκλιων, τετράκλιων δωματίων, σουιτών και bangalows, καθώς και τα πεδία "Photo" και "Video". Ο τύπος δεδομένων (data type) των πεδίων "Τίτλος", "Είδος" και "Κατηγορία" είναι Text (κείμενο), των πεδίων "Photo" και "Video" είναι OLE Object\* ενώ όλων των υπολοίπων είναι Number (αριθμός).

Σε κάθε πίνακα, όπως έχουμε ήδη αναφέρει, μπορούμε να δημιουργήσουμε ένα πρωτεύον κλειδί, ένα πεδίο δηλαδή το οποίο περιέχει μία μοναδική τιμή για κάθε εγγραφή του πίνακα. Έτσι, στον πίνακα που εξετάζουμε προσθέτουμε ένα νέο πεδίο στο οποίο δίνουμε το όνομα ΞενοδοχείοID και σαν τύπο δεδομένων του επιλέγουμε το AutoNumber (Αυτόματη Αρίθμηση). Στη συνέχεια, το δηλώνουμε στην Access σαν πρωτεύον κλειδί. Μ' αυτόν τον τρόπο, κάθε φορά που θα εισάγουμε μία νέα εγγραφή, η Access θα δίνει αυτόματα έναν αύξοντα αριθμό σ' αυτήν, εξασφαλίζοντας έτσι τη μοναδικότητά της, αφού ακόμα κι αν όλα τα άλλα πεδία της έχουν τις ίδιες τιμές με μια προηγούμενη εγγραφή, η τιμή του πεδίου ΞενοδοχείοID θα είναι διαφορετική.

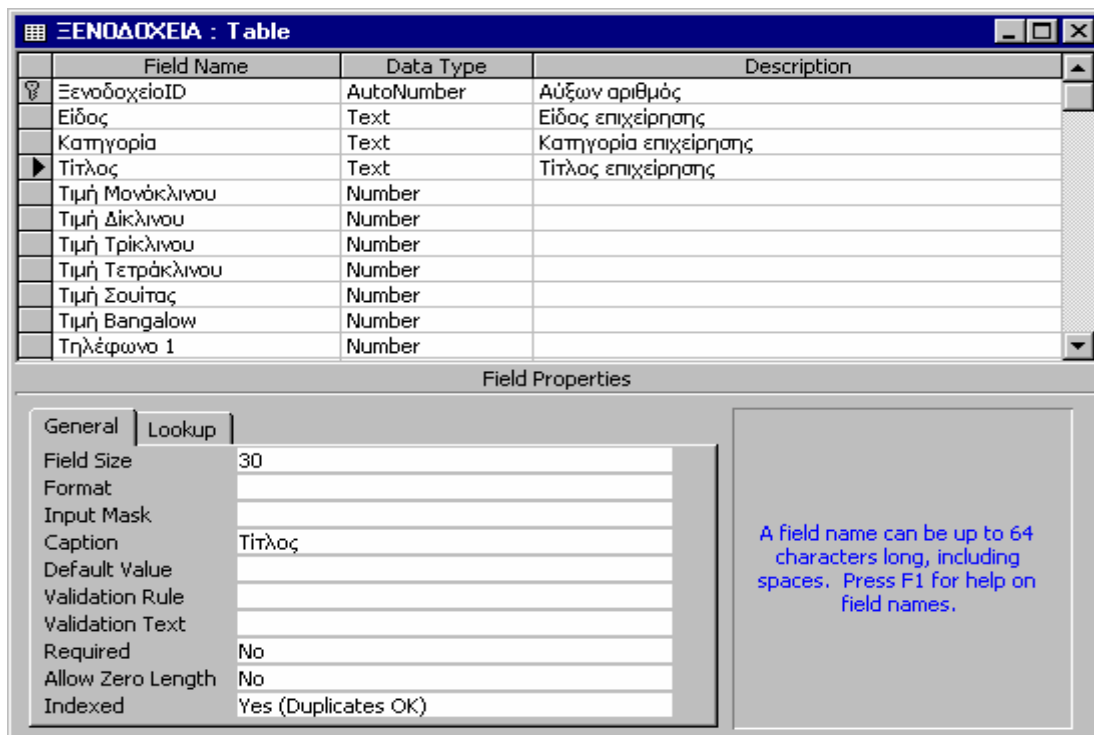
Η τελευταία ενέργεια που πρέπει να κάνουμε, είναι να δώσουμε στην ιδιότητα Index (ευρετήριο) κάποιων πεδίων την τιμή "YES" (NAI), διότι έτσι εξασφαλίζουμε πως η αναζήτηση και ταξινόμηση πληροφοριών στα συγκεκριμένα πεδία θα γίνεται πιο γρήγορα. Η Access, όπως έχουμε δει, δημιουργεί αυτόματα ένα ευρετήριο για τα πρωτεύοντα κλειδιά. Τα υπόλοιπα πεδία στα οποία πρέπει να γίνει κάτι ανάλογο, είναι ο "Τίτλος", το "Είδος", η "Κατηγορία" και η "Τιμή δίκλιου (δωματίου)", διότι όπως θα δούμε στη συνέχεια, θα χρειαστεί μέσω επιλογών που θα κάνει ο χρήστης, να γίνει φιλτράρισμα και αναζήτηση δεδομένων σε αυτά.

---

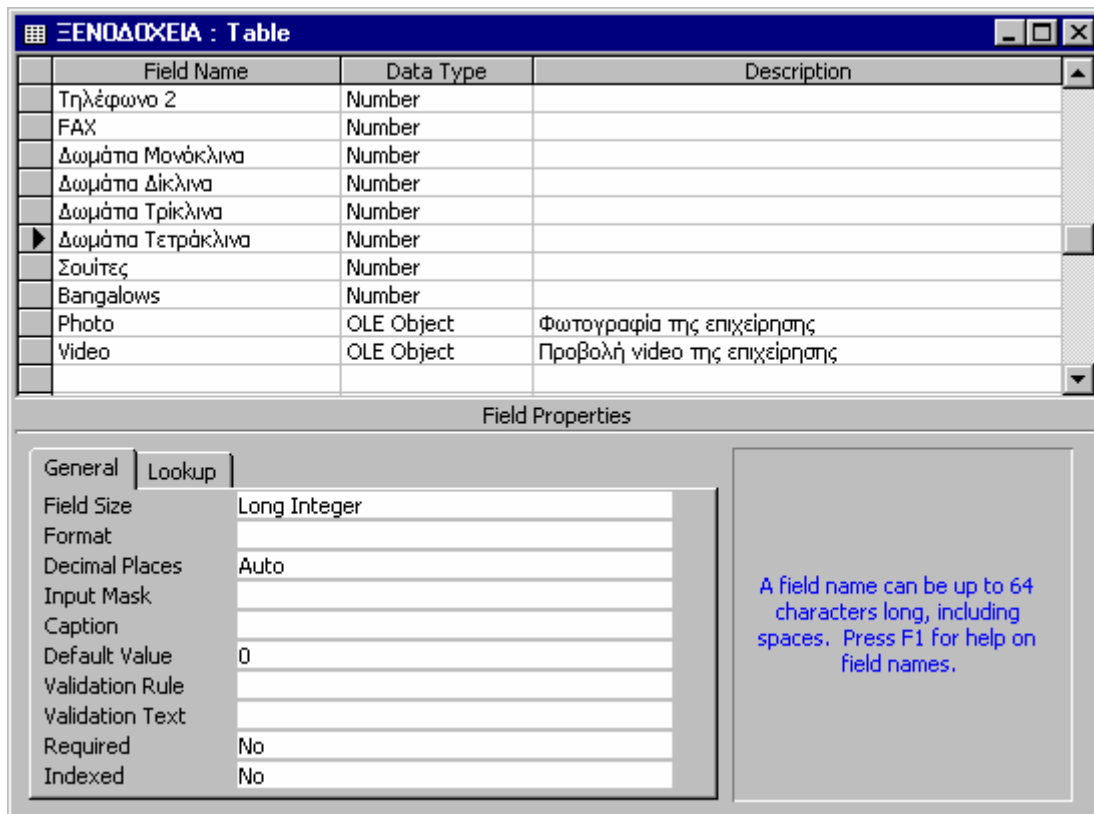
\* Η σύνδεση και ενσωμάτωση αντικειμένων (OLE) κάνει δυνατή τη συνεργασία μεταξύ εφαρμογών των Windows. Ένα αντικείμενο είναι μία μονάδα δεδομένων. Τα αντικείμενα μπορεί να είναι κείμενο, λογιστικά φύλλα, γραφικά, τμήματα οθόνης, ήχοι, και οτιδήποτε άλλο μπορεί να εμφανιστεί ή να ελεγχθεί από μια εφαρμογή.



**Εικόνα 4-2:** Τα πρώτα 11 πεδία του πίνακα "Ξενοδοχεία" με τους τύπους δεδομένων και τις περιγραφές τους (κάτω αριστερά φαίνονται οι ιδιότητες του τύπου AutoNumber πεδίου "ΞενοδοχείοID", το οποίο είναι και πρωτεύον κλειδί του πίνακα).



**Εικόνα 4-3:** Οι ιδιότητες του τύπου Text πεδίου "Τίτλος".



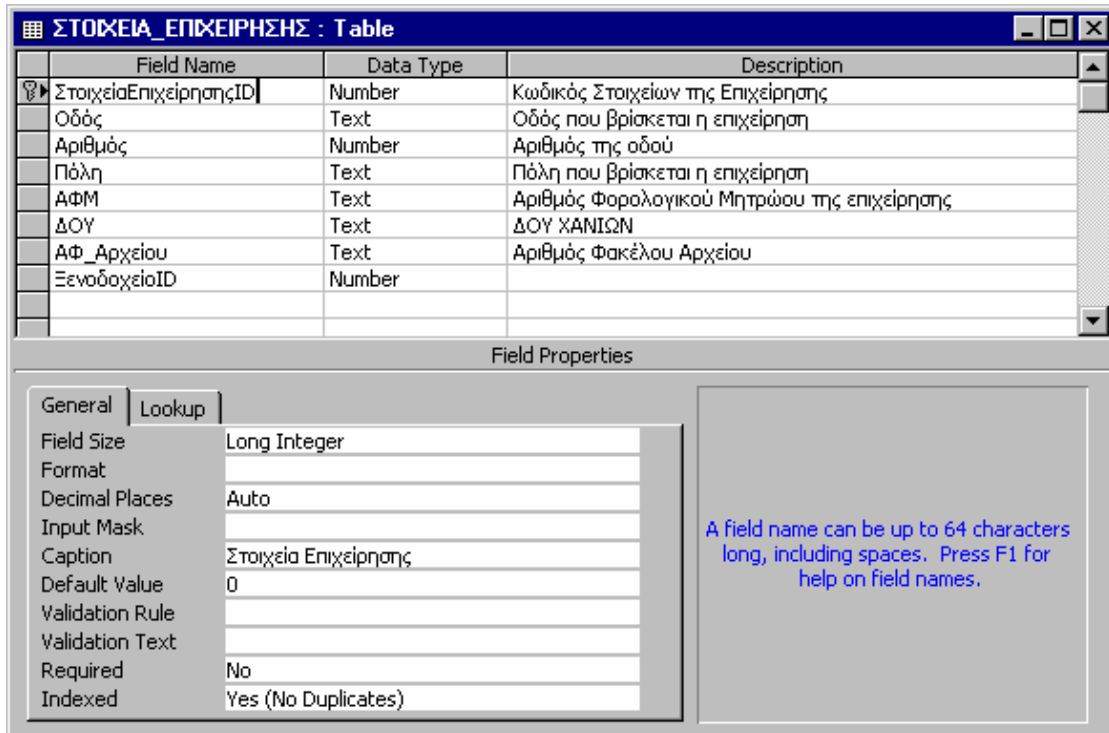
**Εικόνα 4-4:** Τα υπόλοιπα 10 πεδία του πίνακα "Ξενοδοχεία" (κάτω αριστερά φαίνονται οι ιδιότητες του τύπου Number πεδίου "Δωμάτια Τετράκλινα").

#### 4.1.1.2 "Στοιχεία Επιχείρησης"

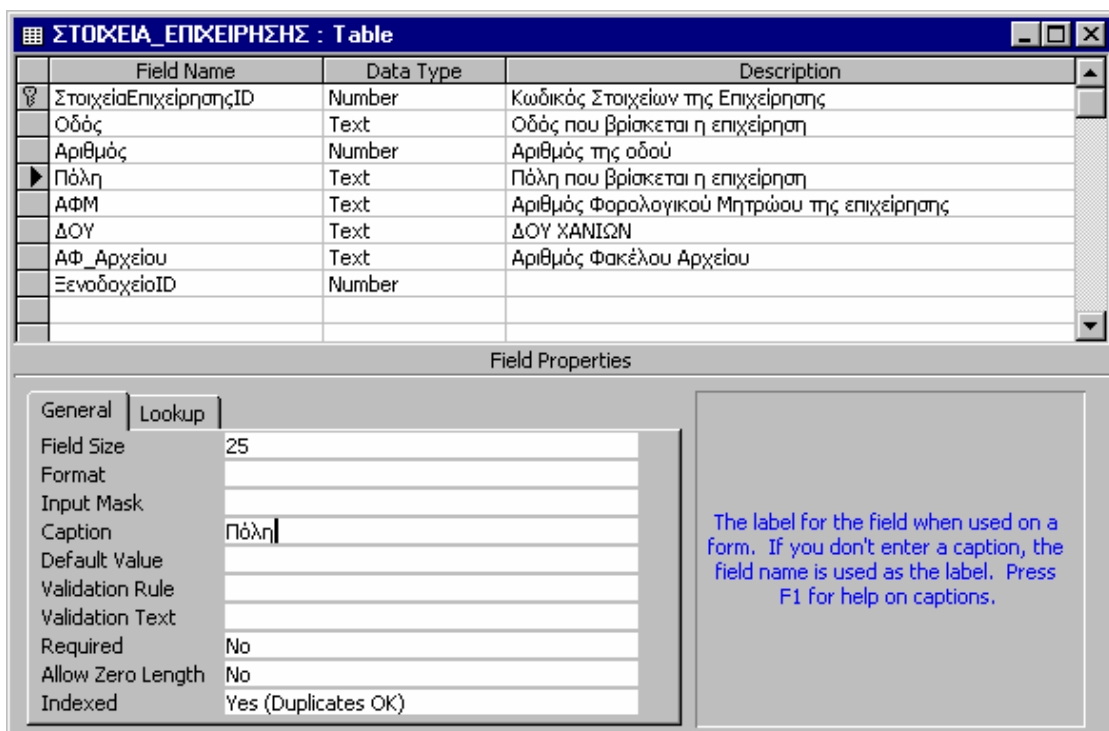
Ο πίνακας "Στοιχεία Επιχείρησης" περιέχει τα πεδία "Πόλη", "Οδός", "Αριθμός", "ΑΦΜ (Αριθμός Φορολογικού Μητρώου)" και "ΑΦ\_Αρχείου (Αριθμό Φακέλου Αρχείου)". Ο τύπος δεδομένων του πεδίου "Αριθμός" είναι φυσικά Number, ενώ όλων των άλλων είναι Text. Κάποιος θα μπορούσε να ρωτήσει γιατί τα πεδία "ΑΦΜ" και "ΑΦ\_Αρχείου" είναι τύπου Text και όχι Number, αφού περιέχουν αριθμητικές τιμές. Αυτό συμβαίνει διότι περιέχουν επίσης και χαρακτήρες όπως καθετούς και παύλες, τους οποίους δεν δέχονται τα πεδία τύπου Number.

Το πεδίο που προσθέτουμε για να παίξει το ρόλο του πρωτεύοντος κλειδιού είναι το "Στοιχεία ΕπιχείρησηςID". Σαν τύπο δεδομένων του επιλέγουμε το Number και συγκεκριμένα το Long Integer (Μεγάλος Ακέραιος), ενώ τα πεδία στα οποία δίνουμε την τιμή "NAI" στην ιδιότητα Index είναι η "Πόλη", η "Οδός" και ο "ΑΦ\_Αρχείου", για το λόγο που εξηγήθηκε στην προηγούμενη παράγραφο.





**Εικόνα 4-5:** Τα πεδία του πίνακα "Στοιχεία Επιχείρησης" (φαίνονται οι ιδιότητες του πρωτεύοντος κλειδιού "ΣτοιχείαΕπιχείρησηςID")

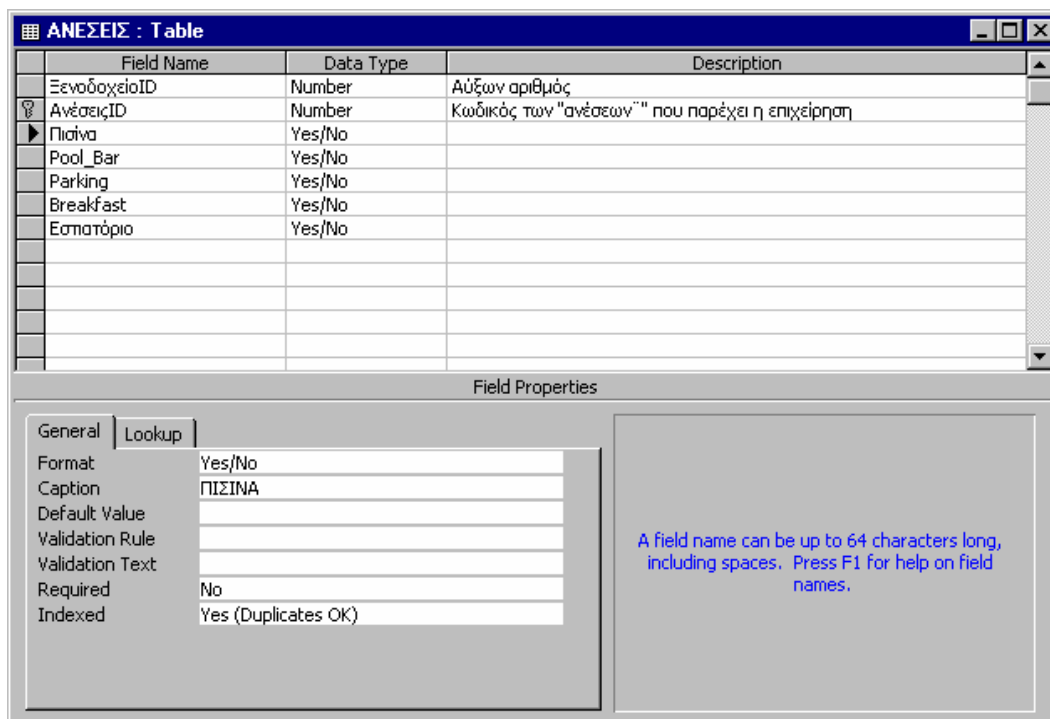


**Εικόνα 4-6:** Οι ιδιότητες του τύπου Text πεδίου "Πόλη"

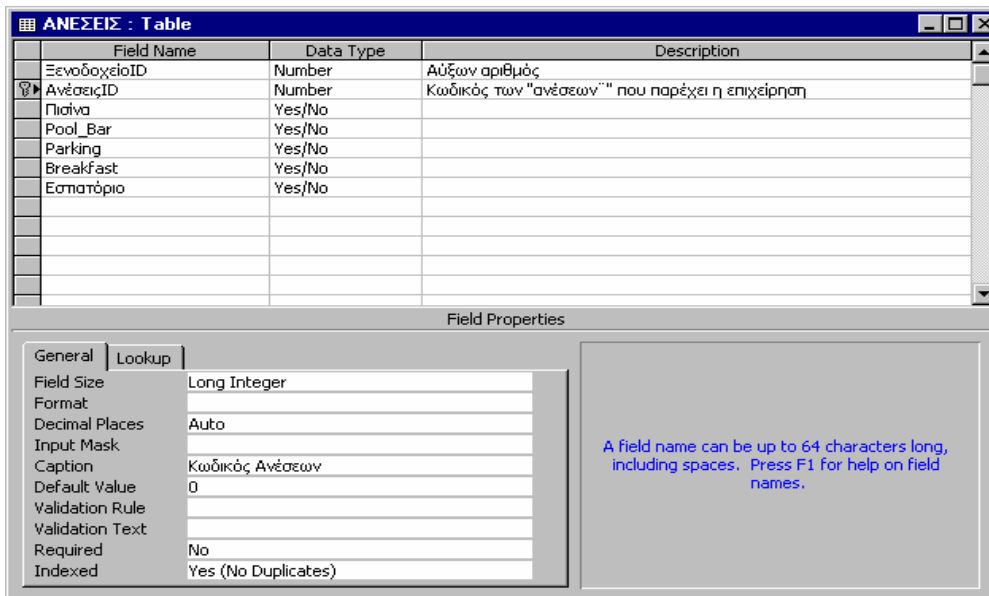
#### 4.1.1.3 "Ανέσεις"

Ο πίνακας αυτός περιλαμβάνει τις ανέσεις που μπορεί να έχει (ή να μην έχει) ένα Ξενοδοχείο και όπως θα δούμε αργότερα είναι ίδιος με τον αντίστοιχο της βάσης δεδομένων "Δωμάτια". Περιέχει τα πεδία "Πισίνα", "Pool bar", "Parking", "Breakfast" και "Εστιατόριο", τα οποία είναι τύπου "Yes/No" (Ναι/Όχι). Δέχονται δηλαδή μία από τις τιμές "Yes" ή "No". Έτσι, για παράδειγμα, όταν ένα ξενοδοχείο διαθέτει Πισίνα, Breakfast και Parking, τα αντίστοιχα πεδία θα πάρουν την τιμή "Yes", ενώ τα πεδία Pool bar και Εστιατόριο θα πάρουν φυσικά την τιμή "No". Το πρωτεύον κλειδί που προσθέτουμε είναι το πεδίο "ΑνέσειςID" (τύπου Long Integer-Number), ενώ όλα τα πεδία σ' αυτόν τον πίνακα έχουν την τιμή "NAI" στην ιδιότητα Index.

Τέλος, όσον αφορά τα πεδία "ΑνέσειςID" και "Στοιχεία ΕπιχείρησηςID", θα μπορούσαν να είναι τύπου AutoNumber για να παίρνουν αυτόματα, όπως είδαμε, έναν αύξοντα αριθμό σαν τιμή. Αυτό όμως, όπως θα δούμε στην ανάλυση Φορμών/Μακροεντολών, γίνεται με τη βοήθεια μακροεντολών οι οποίες ενημερώνουν αυτά τα δύο πεδία και παίζουν το ρόλο της αυτόματης αρίθμησης.



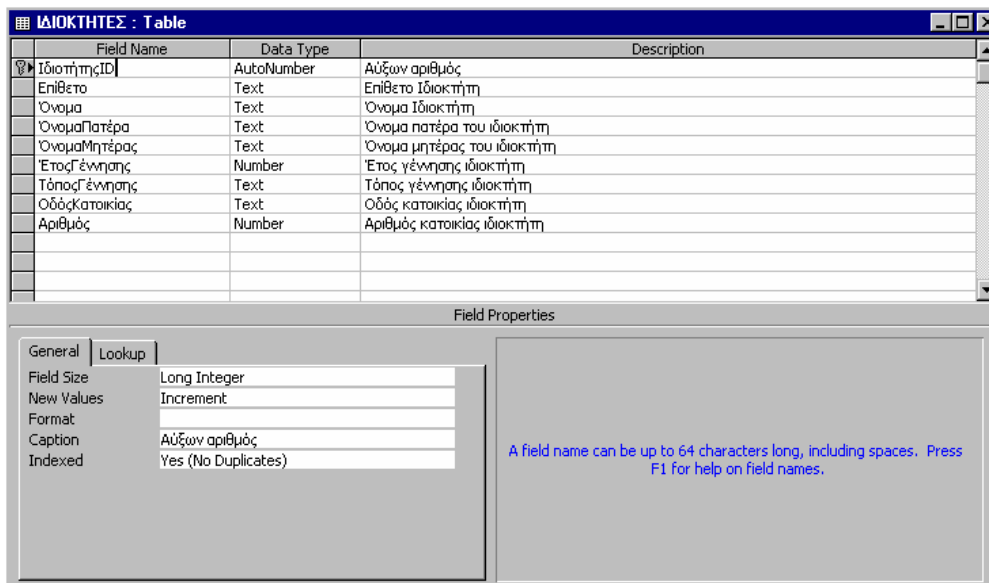
**Εικόνα 4-7:** Τα πεδία του πίνακα "Ανέσεις" (φαίνονται οι ιδιότητες του τύπου YES/NO πεδίου "Πισίνα")



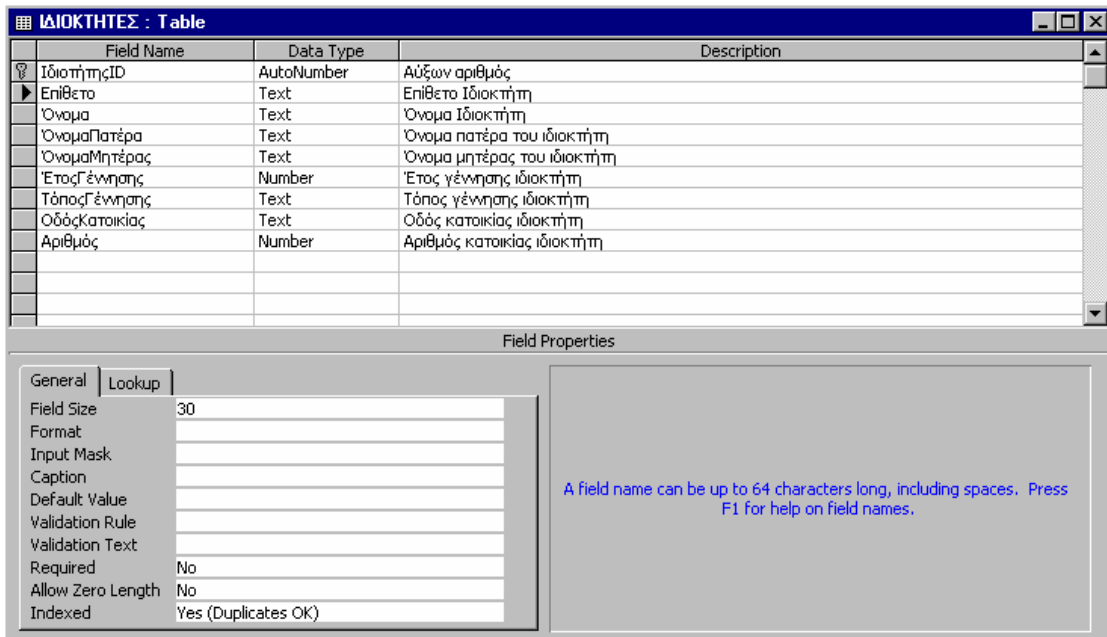
Εικόνα 4-8: Οι ιδιότητες του πρωτεύοντος κλειδιού "ΑνέσειςID".

#### 4.1.1.4 "Ιδιοκτήτες"

Ο πίνακας "Ιδιοκτήτες" περιλαμβάνει τα ακόλουθα πεδία που αναφέρονται στα προσωπικά στοιχεία του ιδιοκτήτη ενός ξενοδοχείου : "Επώνυμο, Όνομα, Όνομα πατέρα, Όνομα μητέρας, Έτος γέννησης, Τόπος γέννησης, Οδός (κατοικίας) και Αριθμός" . Ο τύπος δεδομένων των πεδίων "Έτος γέννησης" και "Αριθμός" είναι Number, ενώ όλων των άλλων είναι Text. Το πρωτεύον κλειδί που προσθέτουμε είναι το πεδίο "ΙδιοκτήτηςID" (τύπου AutoNumber), ενώ στο πεδίο "Επίθετο", που θα χρησιμεύσει όπως θα δούμε αργότερα στην επιλογή και αναζήτηση ιδιοκτήτη, δίνουμε στην ιδιότητα Index την τιμή "ΝΑΙ".



Εικόνα 4-9: Τα πεδία του πίνακα "Ιδιοκτήτες" (φαίνονται οι ιδιότητες του πρωτεύοντος κλειδιού "ΙδιοκτήτηςID")



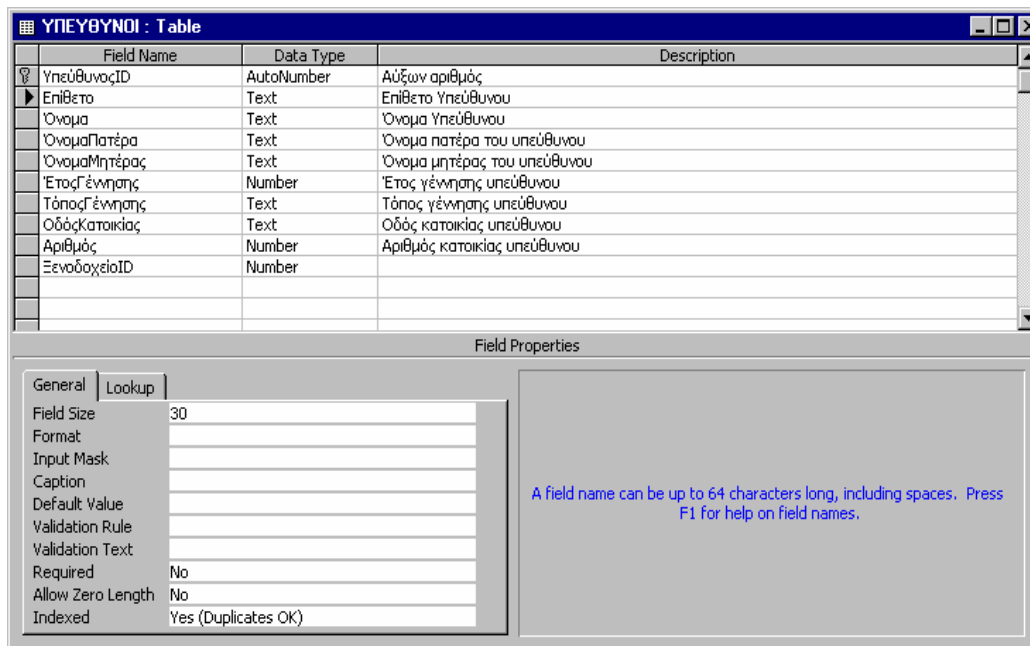
**Εικόνα 4-10:** Οι ιδιότητες του πεδίου "Επίθετο"

#### 4.1.1.5 "Υπεύθυνοι"

Ο πίνακας "Υπεύθυνοι" περιλαμβάνει πεδία που αναφέρονται στα προσωπικά στοιχεία του υπευθύνου ενός ξενοδοχείου :

"Επώνυμο, Όνομα, Όνομα πατέρα, Όνομα μητέρας, Έτος γέννησης, Τόπος γέννησης, Οδός και Αριθμός". Βλέπουμε πως τα πεδία του είναι ίδια μ' αυτά του πίνακα "Ιδιοκτήτες", μόνο που αναφέρονται φυσικά στα στοιχεία των υπευθύνων. Ο τύπος δεδομένων των πεδίων "Έτος γέννησης" και "Αριθμός" είναι Number, ενώ όλων των άλλων είναι Text.

Το πρωτεύον κλειδί που προσθέτουμε είναι το πεδίο "ΥπεύθυνοςID" (τύπου AutoNumber), ενώ προσθέτουμε και ένα ακόμη πεδίο, το "ΞενοδοχείοID" (τύπου Number), το οποίο όπως θα δούμε στη συνέχεια είναι το κοινό πεδίο που χρειάζεται για τη δημιουργία σχέσης μεταξύ των πινάκων "Ξενοδοχεία" και "Υπεύθυνοι". Τέλος, στο πεδίο "Επίθετο", όπως έγινε και στον πίνακα "Ιδιοκτήτες", δίνουμε στην ιδιότητα Index την τιμή "ΝΑΙ".



Field Name	Data Type	Description
ΥπεύθυνοςID	AutoNumber	Αύξων αριθμός
Επώνυμο	Text	Επώνυμο Υπεύθυνου
Όνομα	Text	Όνομα Υπεύθυνου
ΌνομαΠατέρα	Text	Όνομα πατέρα του υπεύθυνου
ΌνομαΜητέρας	Text	Όνομα μητέρας του υπεύθυνου
ΈτοςΓέννησης	Number	Έτος γέννησης υπεύθυνου
ΤόποςΓέννησης	Text	Τόπος γέννησης υπεύθυνου
ΟδόςΚατοικίας	Text	Οδός κατοικίας υπεύθυνου
Αριθμός	Number	Αριθμός κατοικίας υπεύθυνου
ΞενοδοχείοID	Number	

Field Properties	
General	Lookup
Field Size	30
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	No
Indexed	Yes (Duplicates OK)

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

**Εικόνα 4-11:** Τα πεδία του πίνακα "Υπεύθυνοι" (φαίνονται οι ιδιότητες του πρωτεύοντος κλειδιού "ΥπεύθυνοςID").

#### 4.1.2 Δημιουργία Σχέσεων

Το επόμενο βήμα είναι η "δημιουργία σχέσεων" μεταξύ των πινάκων που έχουμε κατασκευάσει.

Είδαμε ότι αντί να συγκεντρώσουμε όλες τις πληροφορίες που θέλουμε να διατηρήσουμε στη βάση δεδομένων μας σε έναν μεγάλο πίνακα, που θα είναι δύσκολο να διαχειριστούμε, μπορούμε να κρατήσουμε τα δεδομένα μας σε περισσότερους από έναν πίνακες, τους οποίους στη συνέχεια συνδέουμε ορίζοντας σχέσεις μεταξύ τους. Για να σχετίσουμε τους πίνακες, πρέπει αυτοί να έχουν τουλάχιστον ένα κοινό πεδίο. Ο τύπος δεδομένων για το κοινό πεδίο πρέπει να είναι ίδιος σε όλους τους πίνακες (δύο ή και περισσότερους) που σχετίζονται, με μία εξαίρεση: όταν το πρωτεύον κλειδί από τη μια μεριά της σχέσης είναι τύπου AutoNumber, μπορεί να σχετίζεται με ένα αριθμητικό πεδίο (δηλαδή τύπου Number) που έχει αποθηκευτεί σαν Long Integer (μεγάλος ακέραιος) από την άλλη μεριά της σχέσης. Τα πεδία τύπου AutoNumber στην πραγματικότητα αποθηκεύονται αυτόματα σαν Long Integers, οπότε ακόμη και αυτή η μοναδική εξαίρεση συμμορφώνεται με τον κανόνα "ίδιος τύπος δεδομένων". Από τη στιγμή που θα ορισθεί η σχέση, μπορούμε να επωφεληθούμε από αυτή βασίζοντας φόρμες και αναφορές σε ερωτήματα που χρησιμοποιούν δεδομένα από διάφορους πίνακες, και όχι μονάχα από έναν. Συνήθως, τα πεδία που συνδέουμε έχουν το ίδιο όνομα (το κοινό πεδίο που

προαναφέραμε), αλλά αυτό δεν είναι απαραίτητο για την ανάπτυξη μιας σχέσης μεταξύ τους. Οποσδήποτε όμως, οι τύποι δεδομένων των πεδίων που συνδέονται πρέπει να είναι ίδιοι (με την εξαίρεση φυσικά που είδαμε) [12].

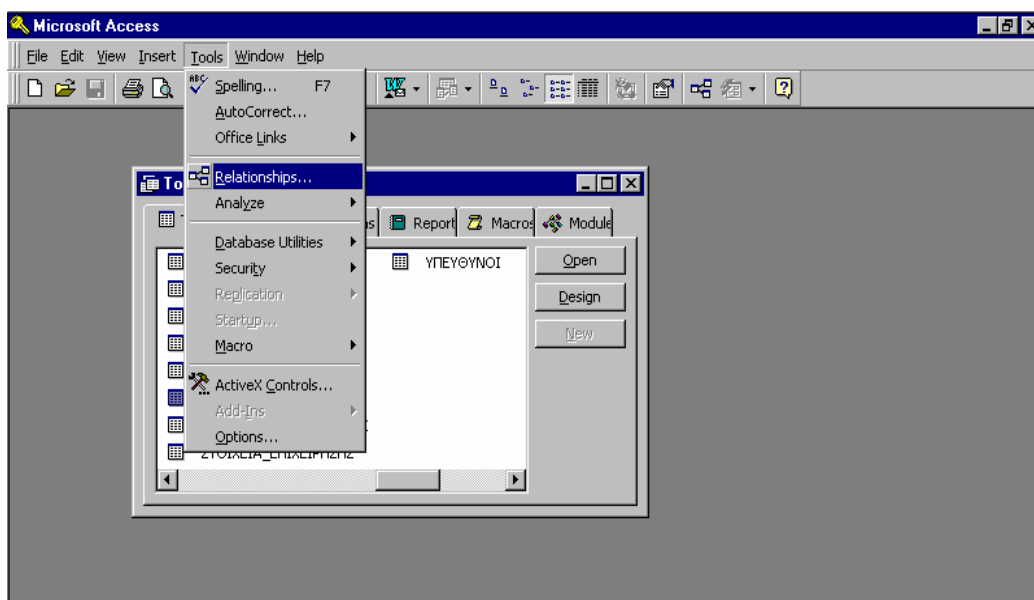
Εξετάζοντας τώρα τους πέντε πίνακες που έχουμε δημιουργήσει, παρατηρούμε ότι για μία εγγραφή του πίνακα "Ξενοδοχεία" αντιστοιχεί μία μόνο στους πίνακες "Στοιχεία Επιχείρησης" και "Ανέσεις". Για παράδειγμα, το Ξενοδοχείο "Αρκάδι" έχει συγκεκριμένες ανέσεις (π.χ. πισίνα, breakfast και εστιατόριο) και συγκεκριμένη διεύθυνση, ΑΦΜ και ΑΦ\_Αρχείου. Επομένως, ανάμεσα στους πίνακες "Ξενοδοχεία-Ανέσεις" και "Ξενοδοχεία-Στοιχεία Επιχείρησης" θα πρέπει να δημιουργήσουμε μία one-to-one (ένα-προς-ένα) σχέση. Επίσης, κάθε ξενοδοχείο έχει συνήθως έναν υπεύθυνο, αλλά θεωρητικά μπορεί να έχει και δύο ή περισσότερους. Γι' αυτό το λόγο η σχέση ανάμεσα στους πίνακες "Ξενοδοχεία" και "Υπεύθυνοι" θα πρέπει να είναι one-to-many (ένα-προς-πολλά).

Τέλος, όσον αφορά τους ιδιοκτήτες των ξενοδοχείων παρουσιάζεται η εξής ιδιομορφία: ένα ξενοδοχείο μπορεί να ανήκει σε πολλούς ιδιοκτήτες (one-to-many), αλλά επίσης, πολλά ξενοδοχεία είναι δυνατόν να ανήκουν σε έναν (many-to-one). Έτσι, ανάμεσα στους πίνακες "Ξενοδοχεία" και "Ιδιοκτήτες" θα πρέπει να δημιουργήσουμε μία many-to-many (πολλά-προς-πολλά) σχέση. Επειδή μία τέτοια σχέση δεν είναι δυνατόν να ορισθεί άμεσα, κάνουμε το εξής: κατασκευάζουμε έναν νέο (έκτο) πίνακα, τον οποίο ονομάζουμε "Ξενοδοχεία-Ιδιοκτήτες". Στον πίνακα αυτό, τοποθετούμε τα πεδία "ΞενοδοχείοID" και "ΙδιοκτήτηςID", τύπου Number-Long Integer. Πρόκειται φυσικά για τα πεδία που είναι πρωτεύοντα κλειδιά στους πίνακες "Ξενοδοχεία" και "Ιδιοκτήτες" αντίστοιχα. Τέτοιοι πίνακες ονομάζονται "Συνδυαστικοί" και σαν σκοπό έχουν τη σύνδεση δύο πινάκων και τη βοήθεια στη δημιουργία μιας σχέσης many-to-many μεταξύ τους. Έτσι, συνδέοντας με δύο one-to-many σχέσεις τους πίνακες "Ξενοδοχεία"- "Ξενοδοχεία-Ιδιοκτήτες" και "Ιδιοκτήτες"- "Ξενοδοχεία-Ιδιοκτήτες", έχουμε τελικά τη σχέση που θέλουμε.

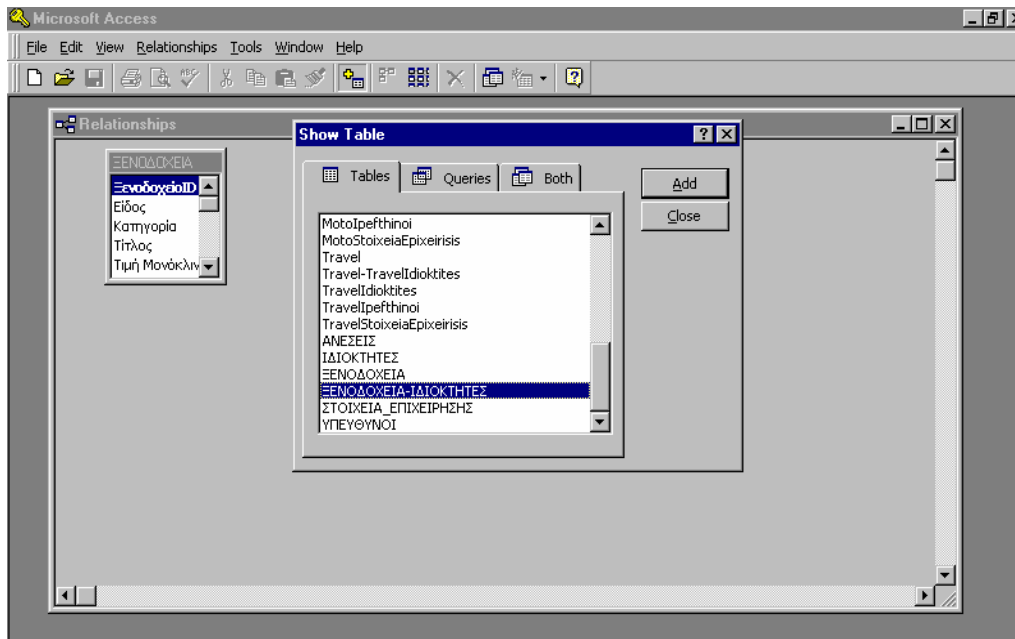
Για τη δημιουργία μίας σχέσης, η Access κάνει τα πράγματα πολύ απλά. Ενώ η βάση δεδομένων μας είναι ανοικτή, ανοίγουμε το μενού tools και επιλέγουμε την εντολή relationship. Εμφανίζεται μία λίστα με τους πίνακες που έχουμε κατασκευάσει και από αυτήν επιλέγουμε όποιους θέλουμε να συνδέσουμε. Όταν τα πεδία που σχετίζουμε είναι πρωτεύοντα κλειδιά, η σχέση που δημιουργείται είναι one-to-one, ενώ όταν μόνο το ένα είναι πρωτεύον κλειδί, τότε δημιουργείται μία one-to-many σχέση. Έτσι, για παράδειγμα, επιλέγουμε τους πίνακες "Ξενοδοχεία" και "Ξενοδοχεία-Ιδιοκτήτες" οι οποίοι έχουν το κοινό πεδίο ΞενοδοχείοID. Το πεδίο αυτό είναι πρωτεύον κλειδί μόνο στον πρώτο πίνακα (στον δεύτερο δεν έχουμε ορίσει πρωτεύον κλειδί), οπότε η σχέση που θα δημιουργηθεί θα είναι one-to-many. Στη συνέχεια επιλέγουμε το πεδίο "ΞενοδοχείοID" του πίνακα "Ξενοδοχεία", και αφού το

σύρουμε στο αντίστοιχο πεδίο του πίνακα "Ξενοδοχεία-Ιδιοκτήτες", εμφανίζεται ένα παράθυρο διαλόγου στο οποίο μπορούμε να επιβεβαιώσουμε ότι επιλέξαμε τα σωστά πεδία. Εκεί, επιλέγουμε τις εντολές "Enforce Referential Integrity" (Επιβολή-Ενίσχυση Αναφορικής Ακεραιότητας), για να είμαστε σίγουροι για την ακρίβεια των δεδομένων που περιέχονται στη βάση δεδομένων μας καθώς και για την τήρηση της επιλεγμένης σχέσης μεταξύ τους, και "Cascade Update Related Fields", "Cascade Delete Related Records", για την αλυσιδωτή ενημέρωση και διαγραφή συσχετισμένων πεδίων και εγγραφών αντίστοιχα. Τέλος, πατάμε το κουμπί "Create" (δημιουργία) και η σχέση μας είναι έτοιμη.

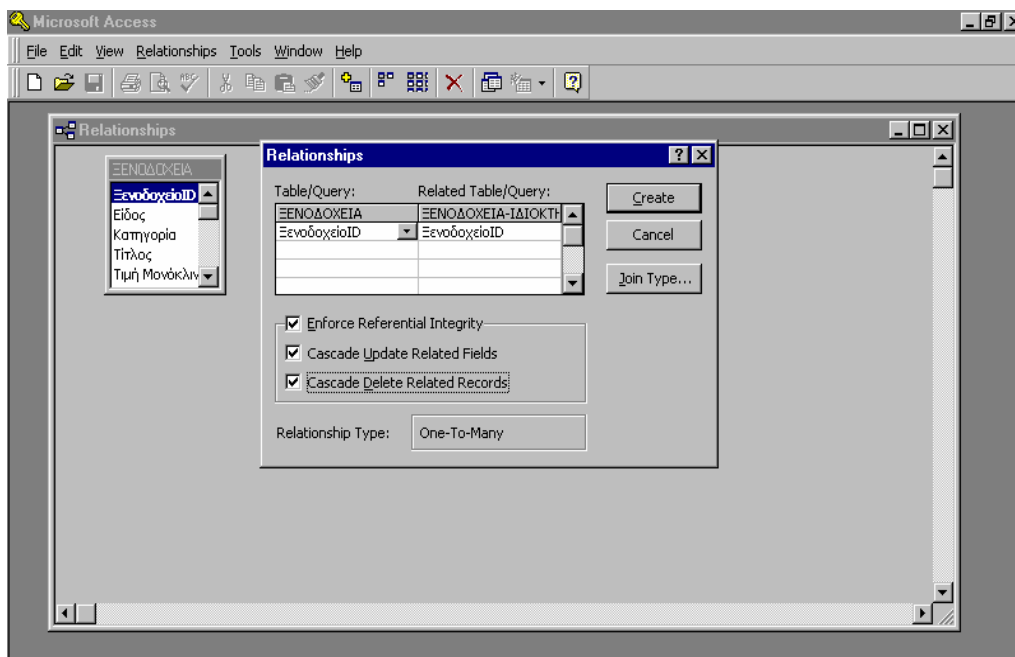
Στη συνέχεια, εργαζόμαστε ομοίως για τα υπόλοιπα ζευγάρια πινάκων που αναφέραμε και δημιουργούμε γρήγορα και εύκολα και τις υπόλοιπες σχέσεις μας. Έτσι, είμαστε πλέον σε θέση να προχωρήσουμε στο επόμενο στάδιο, που είναι η δημιουργία των "Ερωτημάτων".



**Εικόνα 4-13:** Από το μενού "Tools" επιλέγουμε την εντολή "Relationships"

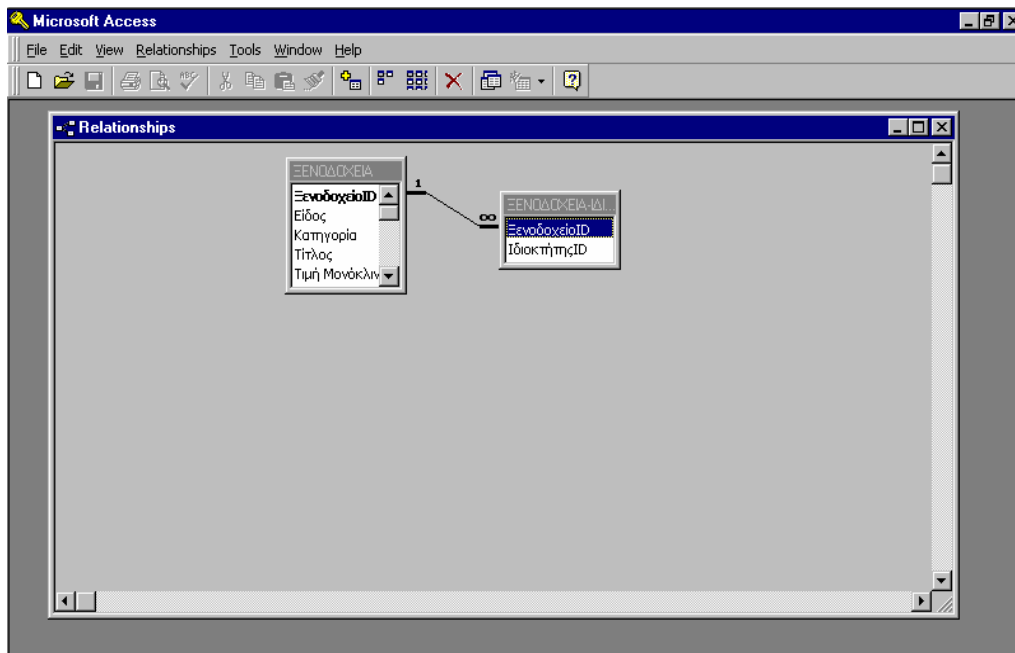


**Εικόνα 4-14:** Επιλέγουμε τους πίνακες που θέλουμε να συσχετίσουμε (π.χ. Ξενοδοχεία και Ξενοδοχεία-Ιδιοκτήτες)

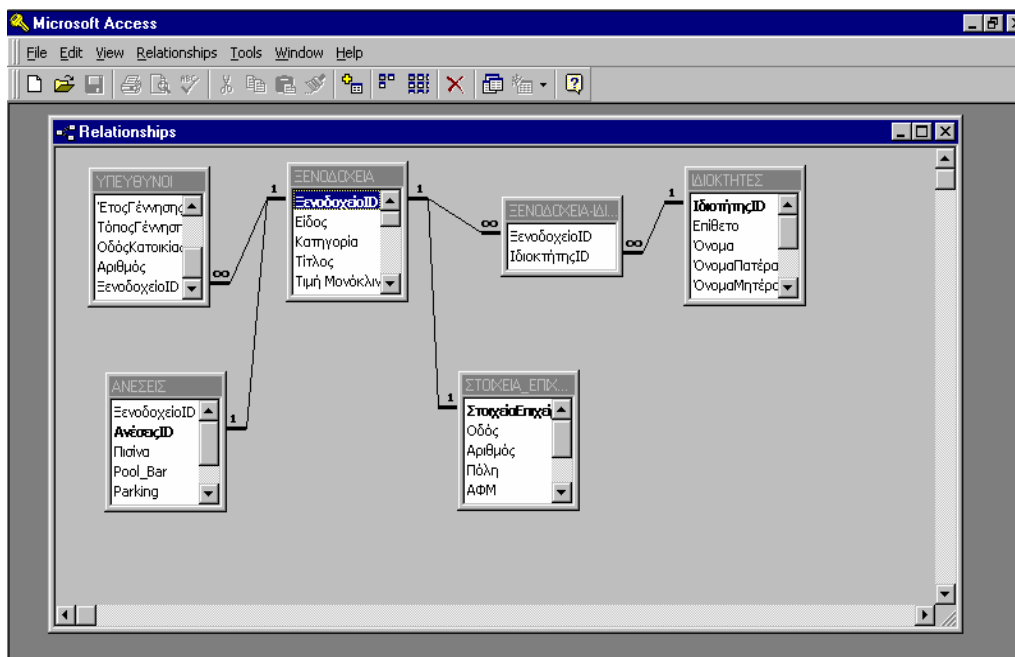


**Εικόνα 4-15:** Στο πλαίσιο διαλόγου που εμφανίζεται επιλέγουμε τις εντολές “Enforce Referential Integrity”, “Cascade Update Related Fields” και “Cascade Delete Related Records”





Εικόνα 4-16: Αφού πατήσουμε το κουμπι “Create”, η σχέση μας είναι έτοιμη.



Εικόνα 4-17: Οι σχέσεις του τμήματος “Ξενοδοχεία”

### 4.1.3 Ερωτήματα

Ένα από τα ισχυρότερα και σημαντικότερα γνωρίσματα της Access, είναι το παράθυρο Graphical Query by Example (GQBE) που επιτρέπει τη γραφική σύνδεση πινάκων. Μπορούν να συνδεθούν πίνακες ακόμα και διαφορετικού τύπου, όπως ένας της Access με έναν πίνακα dBASE. Οι συνδεδεμένοι πίνακες συμπεριφέρονται ως μία ενότητα και επιτρέπουν την επιβολή ερωτήσεων για ανάκτηση δεδομένων. Μπορούμε να επιλέγουμε πεδία, να ορίζουμε σειρά ταξινόμησης, να υπολογίζουμε παραστάσεις και (το κυριότερο) να ορίζουμε κριτήρια επιλογής εγγραφών. Στη συνέχεια, μπορούμε να παρουσιάζουμε τα αποτελέσματα του ερωτήματος σε ένα φύλλο δεδομένων, μία φόρμα ή μία αναφορά.

Επίσης, το ερώτημα είναι δυνατό να έχει διάφορες μορφές. Έτσι, μπορούμε να ορίσουμε ερωτήματα που υπολογίζουν σύνολα και παρουσιάζουν πολύστηλους πίνακες και παράγουν νέους πίνακες με τα αποτελέσματα. Τέλος, με ένα ερώτημα είναι δυνατό να γίνει αλλαγή στα δεδομένα ενός πίνακα, όπως και διαγραφή κάποιας εγγραφής ή προσάρτηση ενός πίνακα σε έναν άλλο [10]. Όλα αυτά γίνονται με τα *ερωτήματα ενεργείας* (action queries). Υπάρχουν φυσικά και τα *επιλεκτικά ερωτήματα* (select queries) τα οποία επιλέγουν πληροφορίες από τους πίνακες και τα ερωτήματα μιας βάσης δεδομένων (όλα τα ερωτήματα της εφαρμογής που αναλύονται στο επόμενο κεφάλαιο είναι select queries).

#### 4.1.3.1 Επιλογές

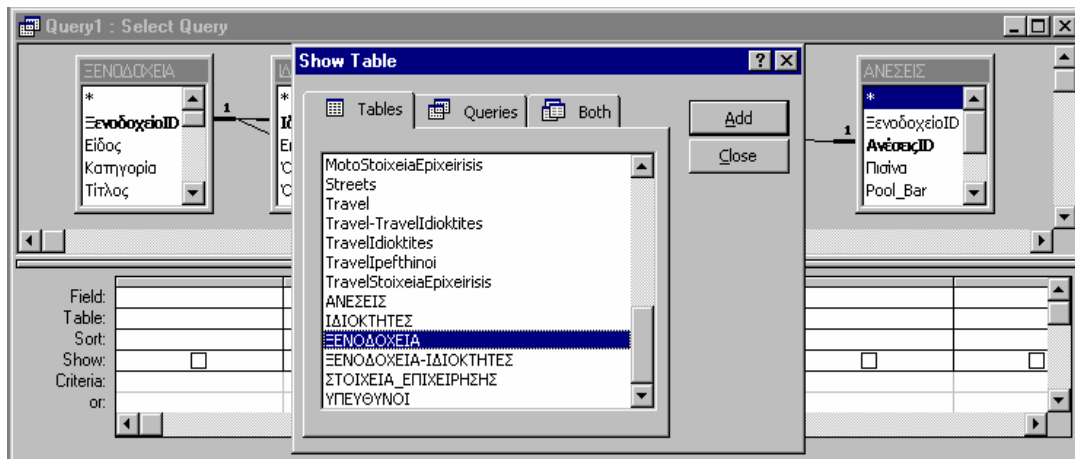
Οι υπεύθυνοι της Τουριστικής Αστυνομίας Χανίων, πέρα από την εισαγωγή των στοιχείων των τουριστικών επιχειρήσεων που λειτουργούν στο νομό Χανίων, θέλησαν να μπορούν να εφαρμόσουν και κάποια φίλτρα σ' αυτά, έτσι ώστε να είναι σε θέση να ξέρουν ποιές εγγραφές συμφωνούν με τα κριτήρια που θέτουν. Συγκεκριμένα, θέλησαν να κάνουν επιλογές με βάση τα εξής πεδία:

"Τίτλος", "Είδος", "Κατηγορία", "Τιμή" (Δίκλινου Δωματίου), "Επίθετο Ιδιοκτήτη/Υπευθύνου", "ΑΦ\_Αρχείου", "Οδός", "Πόλη", "Πισίνα", "Pool Bar", "Parking", "Breakfast", "Εστιατόριο".

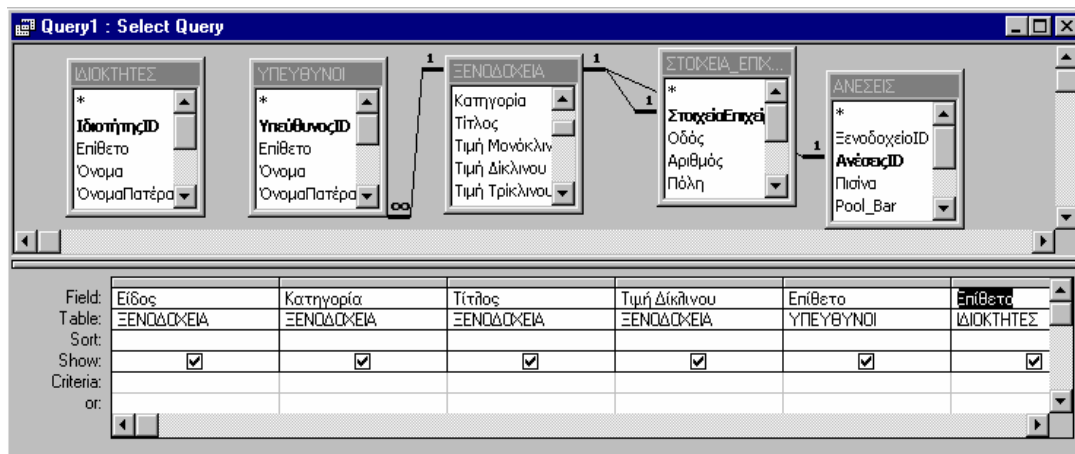
Έπρεπε λοιπόν να κατασκευάσουμε μια φόρμα στην οποία θα εμφανίζονται αυτά τα πεδία και οι τιμές τους για να μπορεί ο χρήστης να κάνει τις επιλογές που επιθυμεί. Όπως θα δούμε στη συνέχεια, κάθε φόρμα, όπως και κάθε αναφορά, βασίζεται σε έναν πίνακα ή ένα ερώτημα. Έτσι, επειδή τα πεδία που αναφέραμε δεν βρίσκονται σε έναν αλλά σε πέντε διαφορετικούς πίνακες, κατασκευάσαμε ένα επιλεκτικό ερώτημα,

το οποίο ονομάσαμε "Epilogos"\* , ενώ η αντίστοιχη φόρμα η οποία βασίζεται σε αυτό ονομάζεται "FrmEpilogos".

Το ερώτημα αυτό βασίζεται όπως είπαμε σε πέντε πίνακες και συγκεκριμένα περιέχει τα πεδία: "Τίτλος", "Είδος", "Κατηγορία" και "Τιμή" του πίνακα "Ξενοδοχεία", "Επίθετο", του πίνακα "Ιδιοκτήτες", "Επίθετο", του πίνακα "Υπεύθυνοι", "ΑΦ\_Αρχείου", "Οδός", "Πόλη", του πίνακα "Στοιχεία Επιχείρησης" και "Πισίνα", "Pool Bar", "Parking", "Breakfast" και "Εστιατόριο", του πίνακα "Ανέσεις".



**Εικόνα 4-18:** Η πρώτη ενέργεια για τη δημιουργία ενός ερωτήματος είναι η επιλογή των πινάκων που περιέχουν τα πεδία που μας ενδιαφέρουν.



**Εικόνα 4-19:** Στη συνέχεια, στη γραμμή "Field" (πεδία) τοποθετούμε τα πεδία που μας ενδιαφέρουν (στη γραμμή "Table" εισάγονται αυτόματα τα ονόματα των πινάκων από τους οποίους προέρχονται τα επιλεγμένα πεδία)

\* Σε ορισμένες περιπτώσεις είναι δυνατό να υπάρξει πρόβλημα στην εκτέλεση μιας μακροεντολής ή λειτουργικής μονάδας, με τη χρησιμοποίηση ελληνικών χαρακτήρων στο όνομα κάποιου αντικειμένου. Για καθαρά προληπτικούς λόγους λοιπόν, χρησιμοποιήθηκαν οι αντίστοιχοι αγγλικοί.

```

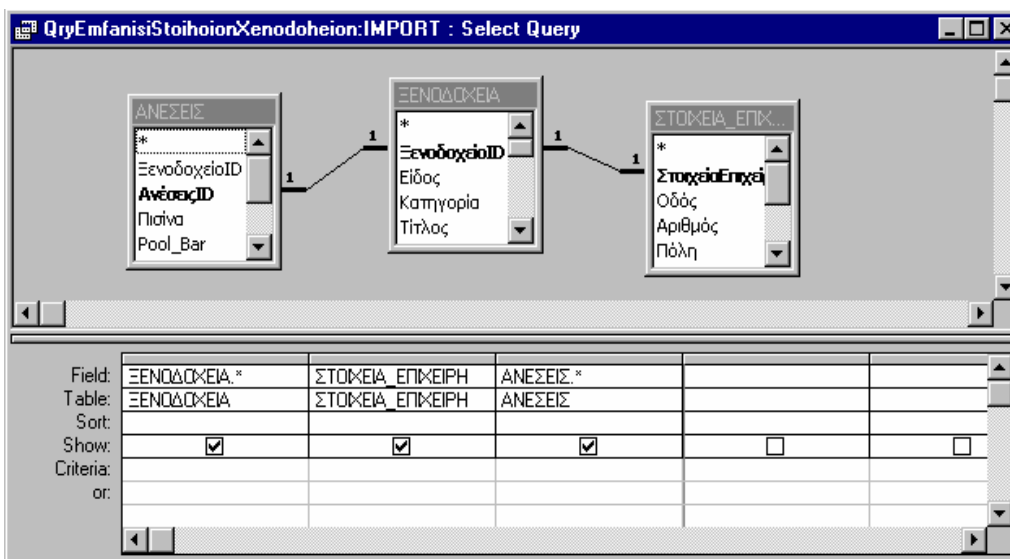
Query1 : Select Query
SELECT ΞΕΝΟΔΟΧΕΙΑ.Είδος, ΞΕΝΟΔΟΧΕΙΑ.Κατηγορία, ΞΕΝΟΔΟΧΕΙΑ.Τίτλος, ΞΕΝΟΔΟΧΕΙΑ.[Τιμή Δίκλιου],
ΥΠΕΥΘΥΝΟΙ.Επίθετο, ΙΔΙΟΚΤΗΤΕΣ.Επίθετο, ΑΝΕΣΕΙΣ.Πισίνα, ΑΝΕΣΕΙΣ.Pool_Bar, ΑΝΕΣΕΙΣ.Parking,
ΑΝΕΣΕΙΣ.Breakfast, ΑΝΕΣΕΙΣ.Εσπιατόριο, ΣΤΟΙΧΕΙΑ_ΕΠΙΧΕΙΡΗΣΗΣ.Οδός, ΣΤΟΙΧΕΙΑ_ΕΠΙΧΕΙΡΗΣΗΣ.Πόλη,
ΣΤΟΙΧΕΙΑ_ΕΠΙΧΕΙΡΗΣΗΣ.ΑΦ_Αρχείου
FROM ΙΔΙΟΚΤΗΤΕΣ, ((ΞΕΝΟΔΟΧΕΙΑ INNER JOIN ΥΠΕΥΘΥΝΟΙ ON ΞΕΝΟΔΟΧΕΙΑ.ΞενοδοχείοID =
ΥΠΕΥΘΥΝΟΙ.ΞενοδοχείοID) INNER JOIN [ΣΤΟΙΧΕΙΑ_ΕΠΙΧΕΙΡΗΣΗΣ] ON ΞΕΝΟΔΟΧΕΙΑ.ΞενοδοχείοID =
ΣΤΟΙΧΕΙΑ_ΕΠΙΧΕΙΡΗΣΗΣ.ΣτοιχείαΕπιχείρησηςID) INNER JOIN ΑΝΕΣΕΙΣ ON ΞΕΝΟΔΟΧΕΙΑ.ΞενοδοχείοID =
ΑΝΕΣΕΙΣ.ΑνέσειςID;

```

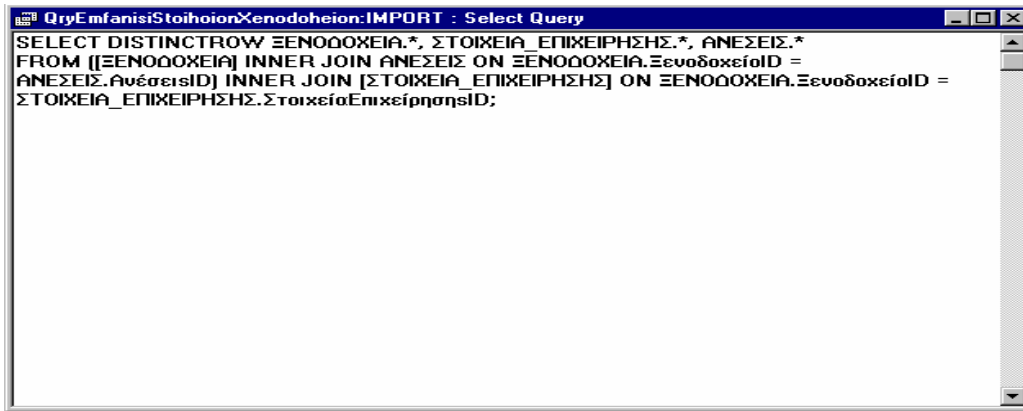
Εικόνα 4-20: Το ίδιο ερώτημα όπως φαίνεται σε μορφή “SQL” (βλ. Παράγραφο 2.1.1)

#### 4.1.3.2 Εμφάνιση Στοιχείων Ξενοδοχείων: Εισαγωγή

Για την εισαγωγή των στοιχείων στη Βάση Δεδομένων, θα μπορούσαμε για κάθε πίνακα να δημιουργήσουμε μια φόρμα που να βασίζεται σ' αυτόν και να εισάγουμε ξεχωριστά τις τιμές των πεδίων τους. Αντί γι' αυτό κάνουμε κάτι πιο πρακτικό: κατασκευάσαμε το ερώτημα "EmfanisiStoicheionXenodoxeion: Import" το οποίο βασίζεται στους πίνακες "Ξενοδοχεία", "Στοιχεία Επιχείρησης" και "Ανέσεις" και περιέχει όλα τα πεδία τους. Έτσι, σε μια μόνο φόρμα που χρησιμοποιείται για την εισαγωγή δεδομένων και βασίζεται σε αυτό το ερώτημα, εισάγουμε ταυτόχρονα τις τιμές όλων αυτών των πεδίων. Θα μπορούσαμε φυσικά να συμπεριλάβουμε και τα πεδία των πινάκων "Ιδιοκτήτες" και "Υπεύθυνοι", αλλά ο εμπλουτισμός μιας φόρμας, που περιέχει ήδη πάρα πολλά πεδία, με δεκατέσσερα επιπλέον, θα την έκανε κουραστική και όχι φιλική προς το χρήστη.



Εικόνα 4-21: Το ερώτημα “ΕμφάνισηΣτοιχείωνΞενοδοχείων:Εισαγωγή” περιέχει όλα τα πεδία των πινάκων “Ξενοδοχεία”, “ΣτοιχείαΕπιχείρησης” και “Ανέσεις”.



Εικόνα 4-22: Το ίδιο ερώτημα σε μορφή "SQL".

#### 4.1.3.3 Εμφάνιση Στοιχείων Ξενοδοχείων

Τα πεδία που χρησιμοποιούνται για τις επιλογές που κάνει ο χρήστης, χωρίζονται από τη φύση τους σε δύο κατηγορίες. Στην πρώτη ανήκουν τα πεδία "Είδος", "Κατηγορία", "Τιμή", "Πόλη", "Πισίνα", "Pool Bar", "Parking", "Breakfast" και "Εστιατόριο", στα οποία οι επιλογές γίνονται ταυτόχρονα. Αυτό συμβαίνει διότι ένας επισκέπτης, για παράδειγμα, δεν ζητάει πληροφορίες μόνο για το είδος, την πόλη και την ύπαρξη ή όχι πισίνας, αλλά ενδιαφέρεται για όλα τα παραπάνω πεδία. Στη δεύτερη κατηγορία ανήκουν τα πεδία "Τίτλος", "Επίθετο Ιδιοκτήτη/Υπευθύνου" και "ΑΦ\_Αρχείου", στα οποία κάθε επιλογή γίνεται ανεξάρτητα από όλες τις άλλες.

Όπως θα δούμε παρακάτω, όταν από τη φόρμα επιλογών (FrmEpiloges) κάνουμε φιλτράρισμα σε ένα από τα πεδία της δεύτερης κατηγορίας, για παράδειγμα επιλέγουμε τον τίτλο "Αρκάδι", εκτελείται η μακροεντολή "OpenForm" (Άνοιγμα Φόρμας), η οποία ανοίγει μια φόρμα παρόμοια με αυτήν που χρησιμοποιείται για την εισαγωγή δεδομένων και εμφανίζει την/τις εγγραφές που στο πεδίο "Τίτλος" του πίνακα "Ξενοδοχεία" έχουν την τιμή "Αρκάδι". Κάτι ανάλογο είναι δυνατό να γίνει και στην περίπτωση που θέλουμε να κάνουμε δύο ή και τρεις επιλογές μαζί, αλλά όχι και στα πεδία της πρώτης κατηγορίας όπου οι επιλογές είναι εννέα.

Τα ερωτήματα δίνουν τη λύση και σ' αυτό το πρόβλημα. Έτσι, κατασκευάζουμε ένα νέο ερώτημα, το οποίο ονομάζουμε "EmfanisiStoixeionXenodoxeion", που περιέχει τα ίδια πεδία με το ερώτημα "EmfanisiStoixeionXenodoxeion: Import". Η φόρμα που θα εμφανίζει τα δεδομένα μας όταν κάνουμε επιλογές στα πεδία της πρώτης κατηγορίας θα βασίζεται σ' αυτό το νέο ερώτημα το οποίο διαφέρει από το προηγούμενο ως προς τη γραμμή "κριτηρίων".

Η πρώτη ενέργεια που κάνουμε όταν δημιουργούμε ένα ερώτημα, είναι όπως είναι φυσικό η επιλογή των πινάκων που περιέχουν τα πεδία που μας ενδιαφέρουν. Αυτά τα

πεδία τα τοποθετούμε με τη σειρά που επιθυμούμε στη γραμμή "Fields" (Πεδία), ενώ στη συνέχεια, στις γραμμές "Sort" και "Show" μπορούμε να επιλέξουμε την ταξινόμηση και την εμφάνιση ή όχι κάποιων πεδίων στο σύνολο των εγγραφών. Η τέταρτη γραμμή είναι αυτή των "κριτηρίων" (Criteria), στην οποία μπορούμε να εισάγουμε τα κριτήρια που θέλουμε να χρησιμοποιήσουμε σαν φίλτρα. Εδώ ακριβώς βρίσκεται η διαφορά ανάμεσα στα δύο παρόμοια ερωτήματα.

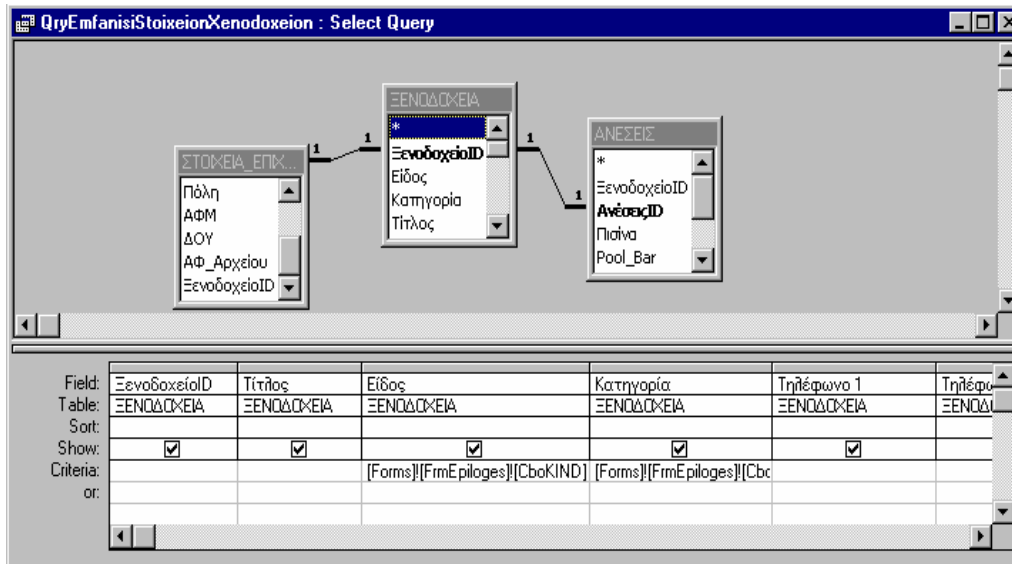
Η φόρμα "FrmEpilogues" περιέχει κάποια "χειριστήρια" όπως "Σύνθετα Πλαίσια Καταλόγου" (Combo Boxes) για τα πεδία "Είδος", "Κατηγορία", "Τιμή", "Πόλη" τα οποία είναι απλώς κατάλογοι (λίστες) που περιέχουν τις τιμές των πεδίων αυτών, και "Πλαίσια Ελέγχου" (Check Boxes) για τα πεδία "Πισίνα", "Pool Bar", "Parking", "Breakfast" και "Εστιατόριο" που χρησιμεύουν στην επιλογή ή όχι κάποιων απ' αυτών των ανέσεων. Τα χειριστήρια αυτά έχουν συγκεκριμένα ονόματα, και αυτά τα ονόματα τοποθετούμε στη γραμμή κριτηρίων. Έτσι, κάθε φορά που κάνουμε στα πεδία της πρώτης κατηγορίας κάποιες επιλογές, ανοίγει η φόρμα εμφάνισης δεδομένων η οποία βασίζεται στο ερώτημα μελετάμε. Οι τιμές που επιλέγουμε από τα χειριστήρια που αναφέραμε γίνονται φίλτρα στη γραμμή κριτηρίων και έτσι εκτελείται το ερώτημα, το οποίο ψάχνει να βρει ποιες εγγραφές συμφωνούν ταυτόχρονα με όλες τις επιλογές που έχουμε κάνει και αυτές οι εγγραφές εμφανίζονται τελικά στην οθόνη.

Τέλος, ένα σημείο που αξίζει να αναφερθεί είναι η ιδιαιτερότητα του πεδίου "Τιμή". Συγκεκριμένα, για την επιλογή της τιμής δεν χρησιμοποιήθηκε ένα, αλλά δύο πλαίσια καταλόγου: ένα για την ελάχιστη και ένα για τη μέγιστη κι αυτό γιατί συνήθως ένας επισκέπτης δεν ενδιαφέρεται για μια συγκεκριμένη τιμή (π.χ. 15500 δρχ.), αλλά για ένα εύρος τιμών (π.χ. 10000 - 25000 δρχ.). Έτσι, στη γραμμή κριτηρίων, για το πεδίο "Τιμή" δεν έχουμε τα ονόματα των αντίστοιχων χειριστηρίων (CboMIN και CboMAX), αλλά την έκφραση:

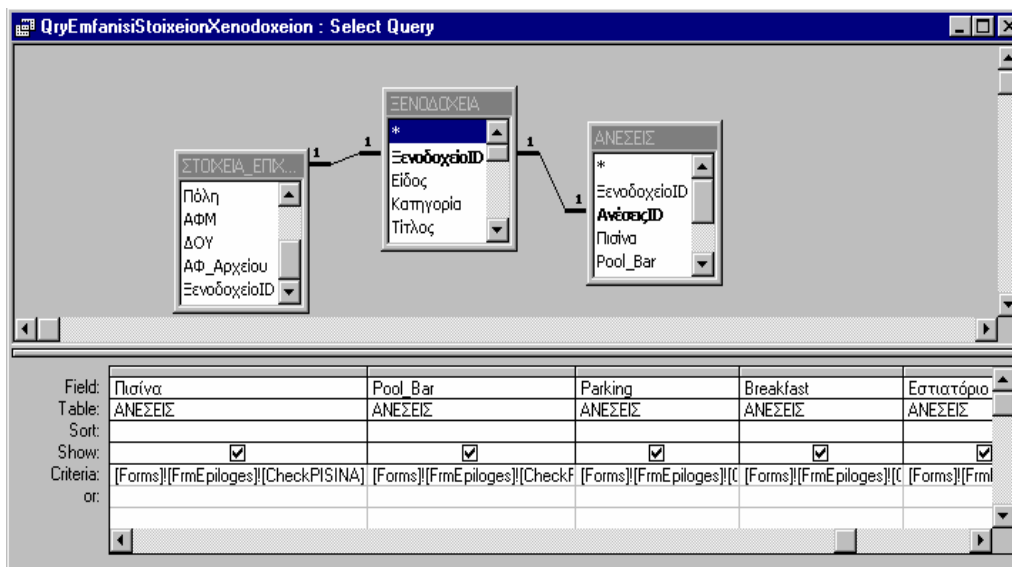
*"Between Forms![PrmEpilogues]![CboMIN] AND  
Forms![FrmEpilogues]![CboMAX]"\**

---

\* Υπάρχει η περίπτωση δύο χειριστήρια που ανήκουν σε διαφορετικές φόρμες (ή αναφορές) να έχουν το ίδιο όνομα. Για το λόγο αυτό στη δόμηση εκφράσεων σαν κι αυτή, μαζί με το όνομα του χειριστηρίου αναφέρουμε και το είδος του αντικειμένου (φόρμα, αναφορά κ.λπ.) στο οποίο περιέχεται, με το όνομά του.



Εικόνα 4-23: Στη γραμμή κριτηρίων μπορούμε να εισάγουμε τα κριτήρια που θέλουμε να χρησιμοποιήσουμε σαν φίλτρα

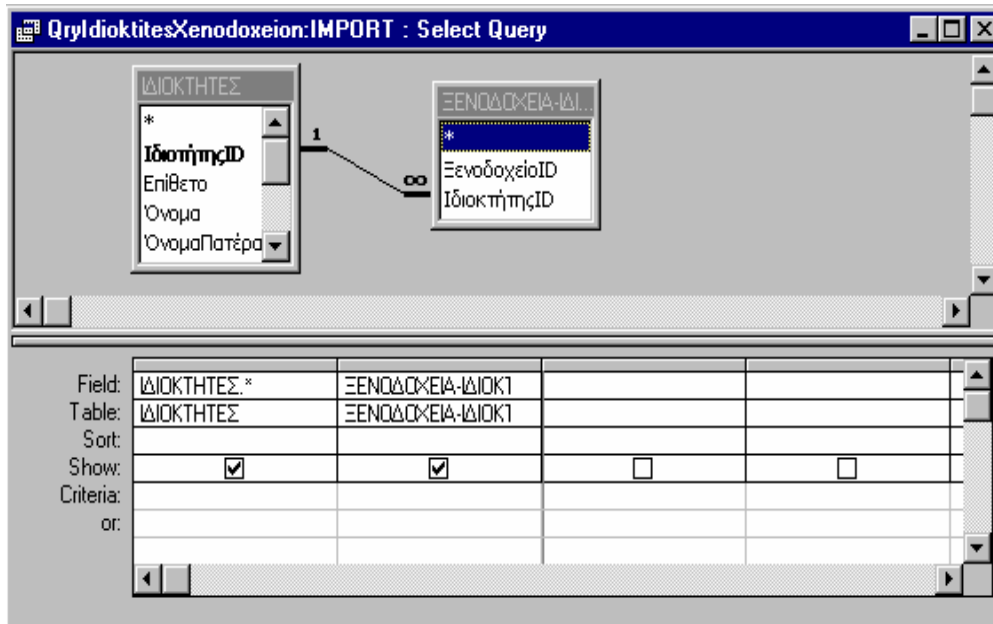


Εικόνα 4-24: Μερικά ακόμα κριτήρια που έχουμε θέσει στο ερώτημα "Εμφάνιση-ΣτοιχείωνΞενοδοχείων"

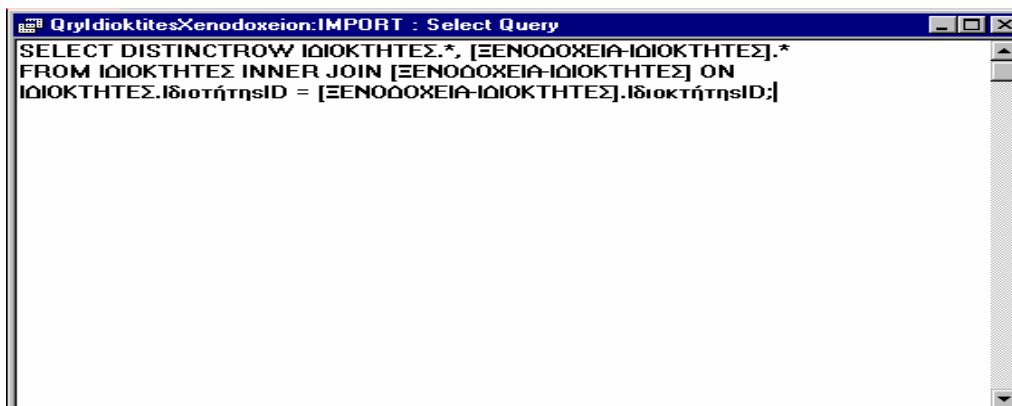
#### 4.1.3.4 "Ιδιοκτήτες Ξενοδοχείων: Εισαγωγή"

Για την εισαγωγή και εμφάνιση στοιχείων των υπευθύνων κατασκευάσαμε μια φόρμα βασισμένη στον πίνακα "Υπεύθυνοι". Θα μπορούσε να γίνει το ίδιο και για τους "Ιδιοκτήτες", αλλά σε μια τέτοια περίπτωση θα έπρεπε να γίνει κάτι αντίστοιχο και για τις τιμές του συνδετικού πίνακα "Ξενοδοχεία-Ιδιοκτήτες". Αντί αυτού, δημιουργήσαμε ένα τέταρτο ερώτημα το οποίο ονομάσαμε "IdioktitesXenodoxeion: Import", που περιέχει όλα τα πεδία των πινάκων "Ιδιοκτήτες" και "Ξενοδοχεία-Ιδιοκτήτες" και σ' αυτό το ερώτημα βασίζεται η φόρμα "FrmIdioktitesXenodoxeion:

Import" που χρησιμεύει, όπως θα δούμε στη συνέχεια, στην εισαγωγή και εμφάνιση στοιχείων των ιδιοκτητών.



**Εικόνα 4-25:** Το ερώτημα “ΙδιοκτήτεςΞενοδοχείων:Εισαγωγή” περιέχει όλα τα πεδία των πινάκων “Ιδιοκτήτες” και “Ξενοδοχεία-Ιδιοκτήτες”.



**Εικόνα 4-26:** Το ίδιο ερώτημα σε μορφή “SQL”

Έχοντας τελειώσει λοιπόν την ανάλυση των ερωτημάτων, μπορούμε να προχωρήσουμε στο επόμενο στάδιο της κατασκευής των “Ξενοδοχείων”, που δεν είναι άλλο από τη δημιουργία φορμών και μακροεντολών. Φυσικά, έχουμε ήδη πάρει μία γεύση από το επόμενο κεφάλαιο χάρη στη φόρμα “FrmEπιλογες”, κάτι που όμως ήταν απαραίτητο για την καλύτερη κατανόηση του μηχανισμού λειτουργίας του ερωτήματος “EmfanisiStoixeionXenodoxeion”.



#### 4.1.4 Φόρμες/Μακροεντολές

Οι φόρμες προσφέρουν έναν προσαρμοσμένο στις ανάγκες μας τρόπο για να εξετάζουμε και να διορθώνουμε τα δεδομένα μας, καθώς και να ελέγχουμε τη ροή της εφαρμογής μας. Μπορούμε να σχεδιάσουμε φόρμες για πολλούς διαφορετικούς σκοπούς [12]:

- Παρουσίαση και διόρθωση δεδομένων
- Έλεγχος της ροής της εφαρμογής
- Εισαγωγή δεδομένων
- Εμφάνιση μηνυμάτων
- Εκτύπωση πληροφοριών

Όλες οι φόρμες της εφαρμογής φαίνονται στο κεφάλαιο "Εφαρμογή Πληροφοριακού Συστήματος".

##### 4.1.4.1 "Εμφάνιση Στοιχείων Ξενοδοχείων: Εισαγωγή"

Η πρώτη φόρμα που κατασκευάσαμε είναι η "FrmEmfanisiStoixeionXenodoxeion: Import", η οποία χρησιμοποιείται για την εισαγωγή των δεδομένων και βασίζεται στο ερώτημα "Εμφάνιση Στοιχείων Ξενοδοχείων: Εισαγωγή", το οποίο όπως είδαμε περιέχει όλα τα πεδία των πινάκων "Ξενοδοχεία", "Στοιχεία Επιχείρησης" και "Ανέσεις". Η μορφή της, όπως και όλων των άλλων φορμών των "Ξενοδοχείων" φαίνεται στο κεφάλαιο "Λειτουργία της εφαρμογής".

Επιχειρώντας να κάνουμε την εισαγωγή των δεδομένων καθώς και την εμφάνιση της φόρμας όσο το δυνατό πιο φιλική προς το χρήστη, κάναμε τις εξής ενέργειες:

1. Όπως είδαμε στην ανάλυση των πινάκων, το πεδίο "ΞενοδοχείοID" του πίνακα "Ξενοδοχεία" είναι τύπου "AutoNumber" (Αυτόματη Αρίθμηση). Έτσι, το χειριστήριο "Text Box" (Πλαίσιο Κειμένου) που εμφανίζει τις τιμές του δεν ήταν απαραίτητο να εμφανίζεται στην οθόνη, μια και οι τιμές αυτές δεν εισάγονται από το χρήστη, αλλά αυτόματα από την Access. Για το λόγο αυτό δώσαμε στην ιδιότητα "Visible" (Ορατό) του χειριστηρίου αυτού την τιμή "No".
2. Εκτός από τις τιμές του πεδίου που αναφέραμε, στην οθόνη εμφανίζονταν αρχικά και αυτές των πεδίων "Στοιχεία ΕπιχείρησηςID", "ΞενοδοχείοID" του πίνακα "Στοιχεία Επιχείρησης" και "ΑνέσειςID", "ΞενοδοχείοID" του πίνακα "Ανέσεις". Για κάθε εγγραφή, οι τιμές αυτών των πέντε πεδίων (που είναι αύξοντες αριθμοί

της εγγραφής) είναι ίδιες εφόσον οι σχέσεις ανάμεσα στους πίνακες "Ξενοδοχεία", "Ανέσεις" και "Στοιχεία Επιχείρησης" είναι τύπου "ένα-προς-ένα". Για το λόγο αυτό, στην ιδιότητα συμβάντος "On Lost Focus" (Μετά την Απενεργοποίηση) του χειριστηρίου που αναφέραμε προηγουμένως έγινε χρήση της μακροεντολής "Set Value", η οποία δίνει τις τιμές που θέλουμε σε κάποια χειριστήρια. Συγκεκριμένα, για καθένα από τα τέσσερα πεδία που αναφέραμε, μια μακροεντολή "Set Value" δίνει στα χειριστήρια (Πλαίσια Κειμένου και εδώ) που εμφανίζουν τις τιμές τους, την αντίστοιχη τιμή του πεδίου "ΞενοδοχείοID" που μελετήσαμε στην αρχή, η οποία όπως είδαμε εισάγεται αυτόματα, ενώ στη συνέχεια δίνουμε στην ιδιότητα "Visible" των χειριστηρίων αυτών την τιμή "No". Έτσι, με τον τρόπο αυτό ο χρήστης δεν είναι αναγκασμένος να εισάγει για κάθε εγγραφή τον ίδιο αριθμό σε πέντε διαφορετικά πλαίσια κειμένου, ενώ ταυτόχρονα αυτά τα πλαίσια δεν εμφανίζονται στη φόρμα μια και οι πληροφορίες που δίνει ένας απλός αύξοντας αριθμός είναι μηδαμινές.

3. Τα δεδομένα που εισάγει ο χρήστης σε κάποια πεδία, ανήκουν σε ένα συγκεκριμένο σύνολο τιμών. Έτσι, το πεδίο "Είδος" μπορεί να πάρει τις τιμές "Ξενοδοχείο", "Διαμέρισμα", "Δωμάτιο", "Οικοτροφείο" ή "Ξενώνας", το πεδίο "Κατηγορία" τις τιμές "Α", "Β" ή "Γ", το πεδίο "Δ.Ο.Υ." τις τιμές "Α' Δ.Ο.Υ. Χανίων" ή "Β' Δ.Ο.Υ. Χανίων" και τα πεδία "Πόλη" και "Οδός" τις πόλεις και οδούς αντίστοιχα του νομού Χανίων. Για την εισαγωγή των τιμών αυτών των πεδίων έγινε χρήση των "Combo Box" (Σύνθετα Πλαίσια Καταλόγου), τα οποία είναι κατάλογοι (λίστες) που περιέχουν για κάθε πεδίο τις τιμές που αναφέραμε (μπορούν επίσης να περιέχουν τις τιμές κάποιου/ων πεδίων ενός πίνακα ή ερωτήματος). Έτσι, ο χρήστης δεν είναι αναγκασμένος να πληκτρολογεί για κάθε εγγραφή τις τιμές των πέντε αυτών πεδίων, αλλά απλά τις επιλέγει από τους αντίστοιχους καταλόγους, και έτσι η εισαγωγή δεδομένων γίνεται πιο εύκολα, και κυρίως πιο γρήγορα. Ειδικά για τα πεδία "Πόλη" και "Οδός", κατασκευάστηκαν οι δύο νέοι πίνακες "Cities" και "Streets" (οι οποίοι περιέχουν τις πόλεις και οδούς του νομού). Έτσι, τα "Combo Box" των πεδίων αυτών, εμφανίζουν τις τιμές των δύο αυτών πινάκων, κατά αλφαβητική σειρά μάλιστα. Αυτό επιτυγχάνεται με χρήση στην ιδιότητα "Row source" των "Combo Box" των εκφράσεων "Select [City] From [Cities] Order By [City]" και "Select [Street] From [Streets] Order By [Street]" αντίστοιχα, όπου "City" και "Street" είναι τα

πεδία των πινάκων που περιέχουν τις πόλεις και τις οδούς. Επίσης, στην ιδιότητα "Limit to List" έχει δοθεί η τιμή "Yes" κι έτσι ο χρήστης δεν μπορεί να εισάγει για παράδειγμα μία πόλη που δεν περιέχεται στη λίστα των πόλεων. Έτσι αποφεύγεται τυχόν λάθος, μια και μπορεί να θέλει (ο χρήστης) να εισάγει για μία εγγραφή την πόλη "Σούγια" και αντί να την επιλέξει από τον κατάλογο να πληκτρολογήσει την τιμή "Σούγα" (τυπογραφικό λάθος δηλαδή). Τότε, το μήνυμα λάθους που εμφανίζεται αποτρέπει το λάθος (μια και στη λίστα των πόλεων δεν περιέχεται η τιμή αυτή). Τέλος, έχει προβλεφθεί η περίπτωση της ενημέρωσης των πινάκων "Cities" και "Streets". Έτσι, δημιουργήθηκαν οι φόρμες "FrmCities" και "FrmStreets" που βασίζονται στους δύο αυτούς πίνακες. Στη συνέχεια, στις ιδιότητες "On Double Clic" των δύο "Combo Box" που αναφέρθηκαν παραπάνω έγινε χρήση των μακροεντολών "Open Form", για να ανοίγουν αυτές οι δύο φόρμες, και "Go to Record" (με την τιμή "New" στο όρισμα "Record"), για να πηγαίνουν σε νέα εγγραφή, ενώ χρησιμοποιήθηκε και η μακροεντολή "Requery" στις ιδιότητες "On Got Focus" (με την ενεργοποίηση) για την άμεση ενημέρωση των καταλόγων. Επομένως, όταν για παράδειγμα ο χρήστης θέλει για μία εγγραφή να εισάγει την πόλη "Καστέλι" η οποία δεν υπάρχει στον κατάλογο, κάνοντας "double clic" στο πλαίσιο καταλόγου μπαίνει στη φόρμα "FrmCities" όπου πληκτρολογεί τη νέα τιμή (την τιμή "Καστέλι" δηλαδή), ενημερώνοντας ουσιαστικά τον πίνακα "Cities". Στη συνέχεια, βγαίνοντας από την φόρμα αυτή επιστρέφει στη λίστα με τις πόλεις η οποία έχει ενημερωθεί και περιέχει πλέον την τιμή "Καστέλι". Οι ενέργειες αυτές έχουν γίνει φυσικά σε όλες τις φόρμες της εφαρμογής που περιλαμβάνουν εισαγωγή ή επιλογή πόλης ή οδού.

4. Για την καλύτερη εμφάνισή της φόρμας, δώσαμε στις ιδιότητες "Scroll Bars", "Record Relectors" και "Navigation Buttons" την τιμή "No", ενώ στην ιδιότητα "Min Max Buttons" την τιμή "Min Enabled". Για τη μετακίνηση του χρήστη από μια εγγραφή σε μια άλλη, δημιουργήθηκαν δύο "Command Buttons" (κουμπιά εντολών): το κουμπί "Επόμενη Εγγραφή", στην ιδιότητα "On Click" του οποίου έγινε χρήση της μακροεντολής "Go to Record" με την τιμή "Next" στο όρισμα "Record", και το κουμπί "Προηγούμενη Εγγραφή" στο οποίο χρησιμοποιήθηκε η ίδια μακροεντολή, αλλά με την τιμή "Previous" αυτή τη φορά στο όρισμα "Record". Επίσης, δημιουργήθηκε το κουμπί "Νέα Καταχώρηση" πατώντας το οποίο πηγαίνουμε σε νέα εγγραφή (με χρήση της "Go to Record" και σ' αυτή την

- πρίπτωση). Τέλος, για το κλείσιμο της φόρμας κατασκευάστηκε ένα ακόμα κουμπί διαταγής στην ιδιότητα "On Click" του οποίου έγινε χρήση της μακροεντολής "Close".
5. Στη φόρμα που μελετάμε, εισάγουμε τις τιμές των πεδίων των πινάκων "Ξενοδοχεία", "Ανέσεις" και "Στοιχεία Επιχείρησης". Επομένως, μία ακόμα ενέργεια που έπρεπε να γίνει, ήταν να βρεθεί ένας τρόπος ώστε να εισαχθούν και οι τιμές των πεδίων του πίνακα "Υπεύθυνοι" και του ερωτήματος "Ιδιοκτήτες Ξενοδοχείων: Εισαγωγή" (που βασίζεται όπως είδαμε στους πίνακες "Ιδιοκτήτες" και "Ξενοδοχεία-Ιδιοκτήτες"). Όπως θα δούμε στη συνέχεια, για την εισαγωγή και εμφάνιση των τιμών των πεδίων που ανήκουν στον πίνακα και το ερώτημα που αναφέραμε κατασκευάστηκαν οι φόρμες "FrmIpefthinoi" και "FrmIdioktitesXenodoxeion: Import". Έτσι, δημιουργήθηκαν δύο επιπλέον κουμπιά εντολής τα οποία ανοίγουν αυτές τις δύο φόρμες. Στο πρώτο, στην ιδιότητα συμβάντος "On Click" χρησιμοποιήθηκαν οι μακροεντολές "Open Form", με την τιμή "FrmIpefthinoi" στο όρισμα "Form Name", και "Go to Record" με την τιμή "New" στο όρισμα "Record". Μ' αυτό τον τρόπο, κάθε φορά που πατάμε αυτό το κουμπί εντολής, ανοίγει η φόρμα "FrmIpefthinoi" η οποία είναι έτοιμη για την εισαγωγή νέας εγγραφής. Ίδια ακριβώς είναι η κατασκευή, καθώς και ο τρόπος λειτουργίας του δεύτερου κουμπιού, με τη διαφορά φυσικά ότι στο όρισμα "Form Name" δίνουμε την τιμή "FrmIdioktitesXenodoxeion:Import".
  6. Κάθε φορά που ανοίγει η φόρμα που εξετάζεται πηγαίνει σε νέα εγγραφή, και το πεδίο "ΞενοδοχείοID" τύπου "AutoNumber" που περιέχει πρέπει να πάρει κάποια τιμή. Αν βγούμε από τη φόρμα χωρίς να εισάγουμε κάποια δεδομένα κάτι τέτοιο δεν γίνεται, οπότε είναι αναγκαίο να εμφανιστεί ένα μήνυμα που να επισημαίνει αυτό το λάθος. Έτσι, στην ιδιότητα "On error" της φόρμας χρησιμοποιήθηκε η συνάρτηση " Iff" για τη δόμηση της έκφρασης "Iff ([ΞενοδοχείοID]=Null; MsgBox ("Η παρούσα φόρμα χρησιμοποιείται για την εισαγωγή δεδομένων.Πρέπει λοιπόν, όταν μπαίνετε σ' αυτήν, αρχικά να εισάγετε τις τιμές των πεδίων που εμφανίζονται στην οθόνη και στη συνέχεια τα στοιχεία ιδιοκτήτη και υπεύθυνου (πατώντας τα αντίστοιχα κουμπιά)) ) ". Μ' αυτό τον τρόπο, κάθε φορά που κλείνει η φόρμα η συνάρτηση "Iff" ελέγχει αν το πεδίο "ΞενοδοχείοID" είναι κενό (Null). Αν ναι, εμφανίζει σε ένα πλαίσιο διαλόγου ή μηνύματος message box (MsgBox) το παραπάνω μήνυμα, το οποίο μάλιστα, πέρα από το

λάθος επισημαίνει και τον τρόπο λειτουργίας της φόρμας (κάτι που φυσικά γίνεται και στο κεφάλαιο "Λειτουργία της εφαρμογής").

#### 4.1.4.2 "Υπεύθυνοι"

Η φόρμα "FrmIpefthinoi" βασίζεται στον πίνακα "Υπεύθυνοι" και χρησιμοποιείται για την εισαγωγή και την εμφάνιση των στοιχείων των υπεύθυνων ενός ξενοδοχείου. Οι ρυθμίσεις στις ιδιότητες που αφορούν στην εμφάνιση της φόρμας είναι παρόμοιες μ' αυτές της προηγούμενης (FrmEmfanisiStoixeionXenodoxeion:Import), ενώ το ίδιο ισχύει και για τα κουμπιά ελέγχου "Επόμενη Εγγραφή" και "Προηγούμενη Εγγραφή" που χρησιμεύουν στη μετακίνηση από μια εγγραφή σε μια άλλη.

Κάθε ξενοδοχείο έχει έναν υπεύθυνο, επομένως τα πεδία "ΥπεύθυνοςID" και "ΞενοδοχείοID" του πίνακα "Υπεύθυνοι" έχουν την ίδια τιμή για κάθε εγγραφή. Επιπλέον, το πεδίο "ΥπεύθυνοςID" είναι τύπου "AutoNumber", οπότε οι τιμές του εισάγονται αυτόματα από την Access. Έτσι, δουλεύοντας όπως και στη φόρμα που εξετάσαμε προηγουμένως, στην ιδιότητα συμβάντος "On Lost Focus" του πλαισίου κειμένου που εμφανίζει τις τιμές του πεδίου "ΥπεύθυνοςID" χρησιμοποιούμε τη μακροεντολή "Set Value" η οποία δίνει αυτές τις τιμές στο πεδίο "ΞενοδοχείοID". Έτσι, οι τιμές στο πεδίο αυτό εισάγονται αυτόματα, και επειδή δεν είναι απαραίτητο να φαίνονται στην οθόνη δίνουμε στην ιδιότητα "Visible" του αντίστοιχου χειριστηρίου την τιμή "No".

Η τελευταία ενέργεια που έγινε, ήταν η δημιουργία ενός κουμπιού εντολής το οποίο ανοίγει την αναφορά "RptIpefthinoi", σε άποψη "Print Preview" (Προεπισκόπηση Εκτύπωσης)\*. Συνεπώς, στην ιδιότητα συμβάντος "On Click" του κουμπιού αυτού χρησιμοποιήθηκε η μακροεντολή "Open Report", στο όρισμα "Report Name" της οποίας δόθηκε η τιμή "RptIpefthinoi". Θέλουμε όμως να μην εμφανίζονται όλες οι εγγραφές, αλλά μόνο η τρέχουσα (που εισάγουμε ή εμφανίζουμε μετά από κάποια επιλογή). Η αναφορά "RptIpefthinoi" βασίζεται όπως και φόρμα που εξετάζουμε στον πίνακα "Υπεύθυνοι". Επομένως, στο όρισμα "Where Condition" της μακροεντολής "Open Report", στο οποίο δίνουμε σαν τιμή μια συνθήκη η οποία πρέπει να ισχύει (και απλά "φιλτράρει" τα δεδομένα), γράφουμε την εξής παράσταση:

---

\* Στη συνέχεια, για την εκτύπωση της αναφοράς, απλά πατάμε το κουμπί "Print" της "Μπάρας Εργαλείων" (Toolbar).

*"[Υπεύθυνοι]![ΥπεύθυνοςID]=Forms![FrmIpefthinoi]![TxtIpefthinosID]"\**

Μ' αυτό τον τρόπο, αν για παράδειγμα η τρέχουσα εγγραφή που εμφανίζεται στη φόρμα "FrmIpefthinoi" είναι αυτή με αύξοντα αριθμό "15" (TxtIpefthinosID=15), το πεδίο "ΥπεύθυνοςID" του πίνακα "Υπεύθυνοι" παίρνει την τιμή "15" και η αντίστοιχη εγγραφή (μόνο) εμφανίζεται στην αναφορά μας.

#### **4.1.4.3 "Ιδιοκτήτες Ξενοδοχείων: Εισαγωγή"**

Η φόρμα "FrmIdioktitesXenodoxeion:Import" βασίζεται στο ερώτημα "Ιδιοκτήτες Ξενοδοχείων: Εισαγωγή", και χρησιμοποιείται για την εισαγωγή ή για την εμφάνιση των στοιχείων των ιδιοκτητών ενός ξενοδοχείου, καθώς και των τιμών των πεδίων του συνδετικού πίνακα "Ξενοδοχεία - Ιδιοκτήτες". Οι ρυθμίσεις που αφορούν στην εμφάνιση της φόρμας και αναφέρθηκαν στις δύο προηγούμενες παραγράφους ισχύουν και εδώ. Παρόμοια είναι επίσης η αυτόματη εισαγωγή τιμών στο πεδίο "ΙδιοκτήτηςID" του πίνακα "Ξενοδοχεία - Ιδιοκτήτες" από τις αντίστοιχες του πεδίου "ΙδιοκτήτηςID" του πίνακα "Ιδιοκτήτες", όπως και το άνοιγμα της αναφοράς ""RptIdioktites" με τη χρήση της κατάλληλης συνθήκης που εξασφαλίζει ότι εμφανίζεται μόνο η τρέχουσα εγγραφή.

Ιδιαίτερο ενδιαφέρον παρουσιάζει η εισαγωγή των τιμών του πεδίου "ΞενοδοχείοID" (του πίνακα "Ξενοδοχεία - Ιδιοκτήτες"). Εδώ δεν χρησιμοποιήθηκε πλαίσιο κειμένου, στο οποίο ο χρήστης απλά θα πληκτρολογούσε έναν αύξοντα αριθμό, αλλά ένα σύνθετο πλαίσιο καταλόγου (Combo Box), το οποίο περιέχει δύο στήλες: στην πρώτη, τις τιμές του πεδίου "ΞενοδοχείοID" και στη δεύτερη αυτές του πεδίου "Τίτλος". Είναι δυνατό να ρυθμίσουμε τις ιδιότητες του χειριστηρίου αυτού, έτσι ώστε να φαίνεται στη φόρμα μόνο η δεύτερη στήλη, αλλά να παίρνει τις τιμές του από την πρώτη. Κάναμε αυτή τη ρύθμιση, κι έτσι ο χρήστης δεν είναι αναγκασμένος να πληκτρολογεί για κάθε εγγραφή έναν απλό αύξοντα αριθμό ξενοδοχείου, αλλά αντιθέτως επιλέγει από τον κατάλογο τον τίτλο του και σαν τιμή του πεδίου "ΞενοδοχείοID" δίνεται αυτόματα ο αντίστοιχος (αύξων) αριθμός. Για παράδειγμα, επιλέγοντας από τον κατάλογο την δέκατη τιμή, το πεδίο "ΞενοδοχείοID" παίρνει αυτόματα την τιμή δέκα. Στη φόρμα όμως φαίνεται ο αντίστοιχος τίτλος (της δέκατης εγγραφής), ο οποίος δίνει σαφώς περισσότερες πληροφορίες από έναν απλό αριθμό.

---

\* "TxtIpefthinosID" είναι το όνομα του χειριστηρίου (πλαίσιο κειμένου) που εμφανίζει τις

#### 4.1.4.4 "Επιλογές"

Η φόρμα "FrmEpilogos" βασίζεται στο ερώτημα "Epilogos", και χρησιμοποιείται στην εμφάνιση των πεδίων στα οποία οι υπεύθυνοι της Τουριστικής Αστυνομίας Χανίων θέλησαν να μπορούν να εφαρμόζουν φίλτρα (μέσω επιλογών που γίνονται στις τιμές τους). Για την εμφάνιση των φιλτραρισμένων δεδομένων χρησιμοποιούνται οι φόρμες "FrmEmfanisiStoixeionXenodoxeion" και "FrmEmfanisiStoixeionXenodoxeion:Emfanisi", οι οποίες βασίζονται στα ερωτήματα "EmfanisiStoixeionXenodoxeion" και "EmfanisiStoixeionXenodoxeion:Import" αντίστοιχα.

Στην ανάλυση του ερωτήματος "EmfanisiStoixeionXenodoxeion", μιλήσαμε για τις δύο κατηγορίες στις οποίες χωρίζονται από τη φύση τους τα πεδία στα οποία ο χρήστης κάνει τις επιλογές του. Για την εμφάνιση των τιμών των πεδίων της δεύτερης κατηγορίας (Τίτλος, Επίθετο Ιδιοκτήτη/Υπεύθυνου και ΑΦ\_Αρχείου) στα οποία κάθε επιλογή γίνεται ανεξάρτητα απ' όλες τις άλλες, χρησιμοποιήθηκαν σύνθετα πλαίσια καταλόγου, στην ιδιότητα "Row Source" των οποίων έγινε χρήση της έκφρασης "Select...From". Για παράδειγμα, στην ιδιότητα αυτή του καταλόγου που εμφανίζει τις τιμές του πεδίου "Τίτλος" δώσαμε την τιμή "Select [Τίτλος] From [Ξενοδοχεία];", όπου "Ξενοδοχεία" είναι ο πίνακας που περιέχει το πεδίο για το οποίο ενδιαφερόμαστε. Δίπλα σε κάθε πλαίσιο καταλόγου δημιουργήθηκε ένα κουμπί εντολής, το οποίο ανοίγει τη φόρμα "FrmEmfanisiStoixeionXenodoxeion:Emfanisi" επιτρέποντας να εμφανιστεί μόνο η τιμή που επιλέξαμε. Για παράδειγμα, δίπλα στον κατάλογο που εμφανίζει τις τιμές του πεδίου "ΑΦ\_Αρχείου" δημιουργήθηκε το αντίστοιχο κουμπί "ΑΦ\_Αρχείου", στην ιδιότητα συμβάντος "On Click" του οποίου έγινε χρήση της μακροεντολής "Open Form". Στη συνέχεια, στο όρισμα "Form Name" δόθηκε η τιμή "FrmEmfanisiStoixeionXenodoxeion:Emfanisi", ενώ στο όρισμα "Where Condition" η τιμή "[EmfanisiStoixeionXenodoxeion:Import]![ΑΦ\_Αρχείου]=Forms![FrmEmfanisiStoixeionXenodoxeion:Emfanisi]![CboArcheio]", όπου "CboArcheio" είναι το όνομα του σύνθετου πλαισίου καταλόγου. Έτσι, επιλέγοντας από τον κατάλογο ένα συγκεκριμένο ΑΦ\_Αρχείου και πατώντας το κουμπί εντολής, στη φόρμα που ανοίγει εμφανίζεται μόνο η εγγραφή/ές που συμφωνούν με την επιλογή μας.

Όσον αφορά τα πεδία της πρώτης κατηγορίας (Είδος, Κατηγορία, Τιμή, Πόλη, Πισίνα, Pool bar, Parking Breakfast και Εστιατόριο) στα οποία οι επιλογές γίνονται

---

τιμές του πεδίου "ΥπεύθυνοςID".

ταυτόχρονα, είδαμε στην ανάλυση του ερωτήματος "EmfanisiStoixeionXenodoxeion" πως τα ονόματα των αντίστοιχων χειριστηρίων που εμφανίζουν τις τιμές τους είναι κριτήρια στο ερώτημα αυτό. Έτσι, δημιουργήθηκε ένα νέο κουμπί εντολής το οποίο ονομάστηκε "Εύρεση", στην ιδιότητα συμβάντος "On Click" του οποίου έγινε χρήση της μακροεντολής "Open Form" με την τιμή "FrmEmfanisiStoixeionXenodoxeion" στο όρισμα "Form Name". Επομένως, κάθε φορά που επιλέγουμε τις τιμές που θέλουμε από τα παραπάνω πεδία, πατώντας το κουμπί "Εύρεση" ανοίγει η φόρμα αυτή, η οποία όπως είπαμε βασίζεται στο ερώτημα "EmfanisiStoixeionXenodoxeion". Οι τιμές αυτές γίνονται κριτήρια στο ερώτημα, κι έτσι τελικά εμφανίζονται μόνο οι εγγραφές που συμφωνούν με τις επιλογές μας.

Στα πεδία αυτά θα μπορούσε να συμπεριληφθεί κι αυτό της "Οδού", αλλά οι υπεύθυνοι της υπηρεσίας τουρισμού θέλησαν η επιλογή οδού να γίνεται μετά από την επιλογή στα πεδία της πρώτης κατηγορίας (συγκεκριμένα στη φόρμα "FrmEmfanisiStoixeionXenodoxeion") και ανάλογα με το μέθος της πόλης που επιλέχθηκε (για παράδειγμα, η επιλογή οδού για κάποιο μικρό χωριό δεν έχει νόημα). Έτσι, στη φόρμα "FrmFrmEmfanisiStoixeionXenodoxeion" περιέχεται ένα "ComboBox" παρόμοιο με αυτό της φόρμας που χρησιμοποιείται στην εισαγωγή δεδομένων, το οποίο λειτουργεί με το ίδιο ακριβώς τρόπο και εμφανίζει τις οδούς του νομού. Αφού επιλεγεί λοιπόν από τον κατάλογο κάποια οδός, πατώντας το κουμπί εντολής "Επιλογή Οδού" εκτελείται η μακροεντολή "ApplyFilter" (Εφαρμογή Φίλτρου) και εμφανίζονται στην οθόνη οι επιχειρήσεις που βρίσκονται στην επιλεγμένη οδό.

Τέλος, τα χειριστήρια που χρησιμοποιούνται για τα πεδία της πρώτης κατηγορίας είναι πάλι σύνθετα πλαίσια καταλόγου, εκτός από αυτά των πέντε ανέσεων για τα οποία χρησιμοποιήθηκαν "Check Boxes" (Πλαίσια Ελέγχου), τα οποία είτε είναι κενά, οπότε παίρνουν την τιμή "No", είτε είναι επιλεγμένα, οπότε περιέχουν το χαρακτήρα "x" και παίρνουν την τιμή "Yes". Αυτό προφανώς γίνεται, διότι όπως έχουμε δει στην ανάλυση του πίνακα "Ανέσεις" τα πεδία αυτά είναι τύπου "Yes/No".

#### **4.1.4.5 "ΕμφάνισηΣτοιχείωνΞενοδοχείων"**

Όπως είδαμε στην προηγούμενη παράγραφο, η μόνη διαφορά ανάμεσα στις φόρμες "FrmEmfanisiStoixeionXenodoxeion" και "FrmEmfanisiStoixeionXenodoxeion:Emfanisi" είναι πως βασίζονται σε διαφορετικά ερωτήματα. Επιπλέον, η εμφάνιση καθώς και ο τρόπος λειτουργίας τους είναι ίδιος μ' αυτόν της φόρμας



"FrmEmfanisiStoixeionXenodoxeion:Import" που αναλύθηκε στην παράγραφο (4.4.1), με τρεις μόνο εξαιρέσεις:

1. Σε αυτές τις δύο φόρμες, δεν θέλουμε να γίνεται εισαγωγή αλλά εμφάνιση δεδομένων. Έτσι, όταν ανοίγουν δεν πηγαίνουν σε νέα εγγραφή.
2. Το ίδιο ισχύει και για τα κουμπιά εντολής που ανοίγουν τις φόρμες "FrmIpefthinoi" και "FrmIdioktitesXenodoxeion:Import". Συνεπώς, στην ιδιότητα "On Click" αυτών των κουμπιών χρησιμοποιήθηκε μόνο η μακροεντολή "Open Form" (κι όχι και η "Go to Record"). Επιπλέον, στο όρισμα "Where Condition" δόθηκε η κατάλληλη συνθήκη προκειμένου να εμφανίζονται οι ιδιοκτήτες και υπεύθυνοι της τρέχουσας μόνο εγγραφής. Για παράδειγμα, για το κουμπί της φόρμας "FrmEmfanisiStoixeionXenodoxeion" που ανοίγει την "FrmIpefthinoi" δόθηκε η τιμή "[Υπεύθυνοι]![ΞενοδοχείοID]=Forms![ FrmEmfanisiStoixeionXenodoxeion]![TtxtXenodoxeioID]", όπου "TtxtXenodoxeioID" είναι προφανώς το όνομα του χειριστηρίου που εμφανίζει τις τιμές του πεδίου "ΞενοδοχείοID".
3. Τέλος, θέλουμε να μπορούμε να εκτυπώνουμε τα αποτελέσματα που προκύπτουν από τις επιλογές του χρήστη υπό τη μορφή μιας όμορφης αναφοράς. Έτσι, οι δύο αυτές φόρμες που εξετάζουμε έχουν ένα επιπλέον κουμπί εντολής, το οποίο με τη βοήθεια της μακροεντολής "Open Report" ανοίγουν τις αναφορές "RptEmfanisiStoixeionXenodoxeion" και "RptEmfanisiStoixeionXenodoxeion:Import", στις οποίες με τη χρήση πάλι της κατάλληλης συνθήκης εμφανίζεται η τρέχουσα εγγραφή.

#### **4.1.4.6 "Κύρια φόρμα Ξενοδοχείων"**

Η έβδομη και τελευταία φόρμα που δημιουργήθηκε στα "Ξενοδοχεία" είναι η "FrmXenodoxeiaMainForm", η οποία θα μπορούσε να χαρακτηριστεί και σαν "Πίνακας Ελέγχου" του τμήματος αυτού. Είναι η πρώτη φόρμα η οποία ανοίγει, όταν από την κεντρική φόρμα της εφαρμογής (την FrmMainForm που θα εξεταστεί στη συνέχεια) επιλέγουμε να μπούμε στα "Ξενοδοχεία".

Όπως συμβαίνει στις κύριες φόρμες όλων των τμημάτων της βάσης που κατασκευάστηκαν, περιέχει μία εικόνα του "Microsoft Clip Art" σχετική με το θέμα της και τρία κουμπιά εντολής. Στο πρώτο, το οποίο ονομάζεται "Εισαγωγή", έγινε χρήση των μακροεντολής "Open Form", στα όρισμα "Form Name" και "Mode" της οποίας δόθηκαν οι τιμές "FrmEmfanisiStoixeionXenodoxeion:Import" και "Add"

αντίστοιχα, και "Go to Record" με την τιμή "New" στο όρισμα "Record". Συνεπώς, το κουμπί αυτό απλά ανοίγει τη φόρμα που χρησιμοποιείται στην εισαγωγή δεδομένων και στέλνει το χρήστη σε νέα εγγραφή. Το δεύτερο, το οποίο ονομάζεται "Επιλογές", με τον ίδιο ακριβώς τρόπο ανοίγει τη φόρμα "FrmEpiloges", απ' όπου εφαρμόζουμε όπως έχουμε δει τα φίλτρα που επιθυμούμε. Τέλος, το τρίτο ονομάζεται "Επιστροφή" και κάνει χρήση της μακροεντολής "Close Form" προκειμένου να κλείνει την κύρια φόρμα των ξενοδοχείων και να επιστρέφει έτσι ο χρήστης στην κεντρική φόρμα της εφαρμογής.

#### **4.1.4.7 "Κεντρική φόρμα της εφαρμογής: Η τελευταία πινελιά"**

Όταν ολοκληρώθηκε η κατασκευή των έξι τμημάτων της εφαρμογής, η τελευταία ενέργεια που έπρεπε να γίνει ήταν η δημιουργία μιας κεντρικής φόρμας η οποία θα λειτουργούσε σαν "Κεντρικός Πίνακας Ελέγχου" και θα βοηθούσε το χρήστη να περνά εύκολα απ' το ένα τμήμα στο άλλο. Η φόρμα αυτή, η οποία είναι η πρώτη που βλέπουμε όταν ανοίγουμε την εφαρμογή, ονομάστηκε "FrmMainForm". Περιέχει τους απαραίτητους τίτλους, το σήμα του Πολυτεχνείου Κρήτης, ένα κουμπί εντολής στην ιδιότητα συμβάντος "On Click" του οποίου χρησιμοποιήθηκε η μακροεντολή "Quit" για την έξοδο από την εφαρμογή, έξι κουμπιά εντολής για την είσοδο στις κύριες φόρμες των έξι τμημάτων (με χρήση φυσικά της "Open Form") και τα αντίστοιχα "labels" για κάθε κουμπί.

Περιέχει επίσης έναν χάρτη της Κρήτης με τις πρωτεύουσες των νομών της και τις κυριότερες πόλεις του νομού Χανίων, τις οποίες προσθέσαμε προκειμένου να κάνουμε τη λειτουργία της εφαρμογής πιο απλή, και κυρίως πιο ευέλικτη. Αυτό γίνεται, διότι πολλές φορές ο χρήστης είναι δυνατό να θελήσει για κάποιο είδος επιχείρησης να κάνει μία μόνο επιλογή: αυτή της "Πόλης". Δηλαδή, να δει απλά τις επιχειρήσεις που υπάρχουν σε μία συγκεκριμένη πόλη. Έτσι, εμπλουτίσαμε το χάρτη με τα labels "Χανιά", "Πλατανιάς", "Ομαλός", "Κολυμβάρι", "Καστέλι", "Φαλάσαρνα", "Κάντανος", "Παλαιόχωρα", "Σούγια", "Σούδα", "Γεωργιούπολη", "Φραγκοκάστελλο" και "Καλύβες", τα οποία τοποθετήθηκαν στις αντίστοιχες τοποθεσίες. Στη συνέχεια, ακριβώς πάνω από κάθε label δημιουργήσαμε ένα κουμπί εντολής, στην ιδιότητα "On Click" του οποίου έγινε χρήση της μακροεντολής "Open Form" με την τιμή "FrmMainEpiloges" στο όρισμα "Form Name". Η φόρμα αυτή περιέχει έξι κουμπιά εντολής, ένα για κάθε είδος επιχείρησης, τα οποία ανοίγουν τις

αντίστοιχες φόρμες εμφάνισης στοιχείων, εμφανίζοντας τις εγγραφές που συμφωνούν με την πόλη και το είδος επιχείρησης που έχουν επιλεγεί ( με χρήση προφανώς της κατάλληλης συνθήκης στο όρισμα "Where Condition" της "Open Form", όπως ακριβώς έγινε και στις φόρμες που έχουν ήδη εξεταστεί). Κατόπιν, στην ιδιότητα "Control Tip Text" κάθε κουμπιού που τοποθετήσαμε στο χάρτη, περιγράψαμε το τι ακριβώς κάνει ( π.χ. για το κουμπί που βρίσκεται πάνω από το label "Χανιά" γράψαμε "Εμφάνιση των επιχειρήσεων που βρίσκονται στα Χανιά"), ενώ στην ιδιότητα "Transparent" δώσαμε την τιμή "Yes", κάνοντάς τα έτσι "διάφανα". Έτσι, στο χάρτη φαίνονται μόνο τα labels των πόλεων.

Όταν όμως ο χρήστης τοποθετήσει τον δείκτη του "mouse" πάνω σε μια πόλη, όπως για παράδειγμα τον "Πλατανιά", θα εμφανιστεί το μήνυμα "Εμφάνιση των επιχειρήσεων που βρίσκονται στον Πλατανιά". Εν συνεχεία, πατώντας στο label "Πλατανιάς" ( στο κουμπί εντολής δηλαδή το οποίο δεν φαίνεται στη φόρμα) ανοίγει η "FrmMainEpilogos". Εκεί, πατώντας για παράδειγμα το κουμπί εντολής που συνοδεύεται από το label "Ξενοδοχεία" (αφού φυσικά επιλέξουμε το "Είδος" που μας ενδιαφέρει) θα ανοίξει η γνωστή φόρμα "FrmEmfanisiStoixeionXenodoxeion: Emfanisi" στην οποία εμφανίζονται οι εγγραφές που στο πεδίο "Πόλη" έχουν την τιμή "Πλατανιάς". Επομένως, με το πάτημα δύο κουμπιών εντολής (ένα εκ των οποίων είναι "αόρατο"), επιλέγουμε την πόλη και την επιχείρηση που μας ενδιαφέρει και φιλτράρουμε τα δεδομένα εύκολα και γρήγορα.

#### 4.1.5 Αναφορές

Επειδή οι φόρμες έχουν σαν κύριο σκοπό την εμφάνιση στην οθόνη μεμονομένων εγγραφών ή μικρών ομάδων σχετικών μεταξύ τους εγγραφών με κάποιον ελκυστικό τρόπο, δεν αποτελούν τον καλύτερο τρόπο για να τυπώνουμε και να συνοψίζουμε μεγάλα σύνολα δεδομένων μιας βάσης δεδομένων. Ο καλύτερος τρόπος για τη δημιουργία ενός τυπωμένου αντίγραφου πληροφοριών που έχουν εξαχθεί ή υπολογισθεί από δεδομένα μιας βάσης, είναι οι αναφορές, οι οποίες υπερτερούν έναντι των άλλων μεθόδων εκτύπωσης δεδομένων, σε δύο βασικά σημεία:

1. Μπορούν να συγκρίνουν, να συνοψίσουν και να βρουν τα μερικά αθροίσματα για μεγάλα σύνολα δεδομένων.

2. Μπορούν να δημιουργηθούν έτσι ώστε να παράγουν κομμάτι τιμολόγια, δελτία παραγγελιών, ταχυδρομικές ετικέτες, υλικό για παρουσιάσεις και άλλα τυπωμένα έγγραφα που χρειάζονται για την αποτελεσματική λειτουργία κάποιων εμπορικών συναλλαγών.

Όπως και με τις φόρμες, μια αναφορά βασίζεται σε έναν πίνακα ή ένα ερώτημα, ενώ σε κάθε τμήμα της είναι δυνατό να ενσωματωθούν εικόνες, γραφήματα ή υποαναφορές.

#### **4.1.5.1 "Εμφάνιση Στοιχείων Ξενοδοχείων"**

Στην εξέταση των φορμών "FrmEmfanisiStoixeionXenodoxeion:Emfanisi" και "FrmEmfanisiStoixeionXenodoxeion", είδαμε πως περιέχουν από ένα κουμπί εντολής τα οποία ανοίγουν τις αναφορές "RptEmfanisiStoixeionXenodoxeion:Import" και "RptEmfanisiStoixeionXenodoxeion" αντίστοιχα. Η πρώτη, βασίζεται στο ερώτημα "RptEmfanisiStoixeionXenodoxeion:Import" και εμφανίζει την τρέχουσα εγγραφή της φόρμας με χρήση όπως είδαμε της κατάλληλης συνθήκης στο όρισμα "Where Condition" του κουμπιού εντολής. Η δεύτερη, βασίζεται στο ερώτημα "EmfanisiStoixeionXenodoxeion" και λειτουργεί με τον ίδιο ακριβώς τρόπο, ενώ και οι δύο δημιουργήθηκαν με χρήση του "Report Wizard".

#### **4.1.5.2 "Υπεύθυνοι" - "Ιδιοκτήτες"**

Οι αναφορές "RptIpefthinoi" και "RptIdioktites" βασίζονται στους πίνακες "Υπεύθυνοι" και "Ιδιοκτήτες", αντίστοιχα. Δημιουργήθηκαν με χρήση του "Report Wizard" και εμφανίζουν τις τρέχουσες εγγραφές των φορμών "FrmIpefthinoi" και "FrmIdioktitesXenodoxeion:Import". Ενδιαφέρον παρουσιάζει η αναφορά "RptIdioktites" η οποία, όπως και η αντίστοιχη φόρμα των ιδιοκτητών, θα μπορούσε να βασίζεται στο ερώτημα "IdioktitesXenodoxeion:Import". Θέλαμε όμως να εμφανίζονται μόνο τα στοιχεία κάποιου ιδιοκτήτη, και όχι να εισάγονται αυτά όπως και οι τιμές του πίνακα "Ξενοδοχεία - Ιδιοκτήτες". Επιπλέον, η εμφάνιση αυτών των τιμών δεν θα έδινε στο χρήστη κάποια ουσιαστική πληροφορία. Έτσι, επιλέχθηκε τελικά ο πίνακας "Ιδιοκτήτες" σαν πηγή προέλευσης της αναφοράς "RptIdioktites".

## 4.2 ΑΛΛΑ ΤΜΗΜΑΤΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Με την εξέταση των αναφορών ολοκληρώθηκε η ανάλυση του τμήματος "Ξενοδοχεία", το οποίο λόγω της πολυπλοκότητάς του και του μεγάλου αριθμού δεδομένων που περιέχει, κρίθηκε ότι πρόκειται για το πλέον αντιπροσωπευτικό σε σχέση με τα υπόλοιπα πέντε τμήματα, της παρούσας εφαρμογής.

Η λειτουργία καθώς και η κατασκευή των υπολοίπων τμημάτων γίνεται με τον ίδιο ακριβώς τρόπο με αυτόν των "Ξενοδοχείων", συνεπώς κρίθηκε σκόπιμο να παρουσιασθούν μόνο κάποιες στοιχειώδεις διαφορές, που αφορούν κυρίως στην ύπαρξη λιγότερων δεδομένων και συνεπώς λιγότερων επιλογών από το χρήστη, καθώς και στη διαφορετική ονομασία των αντικειμένων (πινάκων, ερωτημάτων, φορμών, αναφορών και μακροεντολών) της εκάστοτε βάσης\*. Συνεπώς, για καθένα θα αναφερθούν συνοπτικά τα σημεία στα οποία εντοπίζονται οι διαφορές αυτές.

### 4.2.1 "Ενοικιαζόμενα Δωμάτια" (Rooms)

Περιλαμβάνει τις επιχειρήσεις που λειτουργούν στο νομό Χανίων και έχουν άδεια λειτουργίας δωματίων. Πέρα από τη διαφορετική ονομασία των αντικειμένων που αναφέρθηκαν διαφέρει από τα "Ξενοδοχεία" σε ένα μόνο σημείο. Συγκεκριμένα, στον πίνακα "Rooms" ο οποίος είναι ο αντίστοιχος του πίνακα "Ξενοδοχεία" απουσιάζει το πεδίο "Κατηγορία", επομένως στη φόρμα "FrmRoomsEpiloges" η οποία είναι η αντίστοιχη της "FrmEpiloges" των ξενοδοχείων, έχουμε μια λιγότερη επιλογή (αυτή της κατηγορίας).

### 4.2.2 "Γραφεία Ενοικίασεως Αυτοκινήτων" (Cars)

Περιλαμβάνει τα γραφεία ενοικίασεως αυτοκινήτων που λειτουργούν στο νομό Χανίων. Διαφέρει από τα "Ξενοδοχεία" στο ότι απουσιάζουν τα πεδία "Είδος" και "Κατηγορία" του πίνακα "Rooms", καθώς και ο πίνακας "Ανέσεις". Αρχικά απουσίαζε και το πεδίο "Τιμή", αλλά προκειμένου να προβλεφθούν μελλοντικές ανάγκες της Τουριστικής Αστυνομίας Χανίων προστέθηκαν τα πεδία που αφορούν τις τιμές των αυτοκινήτων ανάλογα με τον κυβισμό τους, καθώς και ο αριθμός

αυτοκινήτων για κάθε κυβισμό: "Cars600cc", "Cars900cc", "Cars1000cc", "Jeep", "Timi600cc", "Timi900cc", "Timi100cc", και "TimiJeep". Έτσι, τα πεδία στα οποία ο χρήστης μπορεί να κάνει επιλογές στη φόρμα "FrmCarsEpiologes" μειώνονται στα εξής πέντε: "Τίτλος", "ΑΦ\_Αρχείου", "Πόλη", "Υπεύθυνος" και "Ιδιοκτήτες".

#### 4.2.3 "Γραφεία Ενοικιάσεως Μοτοσυκλετών" (Moto)

Περιλαμβάνει τα γραφεία ενοικιάσεως μοτοσυκλετών που λειτουργούν στο νομό Χανίων και είναι πανομοιότυπο με τα γραφεία ενοικιάσεως αυτοκινήτων. Όπως και στο προηγούμενο, αρχικά απουσίαζε το πεδίο "Τιμή", αλλά για το λόγο που αναφέραμε πριν, προστέθηκαν τα πεδία που αφορούν τις τιμές των μοτοσυκλετών ανάλογα με τον κυβισμό τους, καθώς και ο αριθμός μοτοσυκλετών για κάθε κυβισμό: "Moto50cc", "Moto70cc", "Moto90cc", "Moto100cc", "Moto125cc", "Timi50cc", "Timi70cc", "Timi90cc", "Timi100cc" και "Timi125cc". Έτσι, όπως και προηγουμένως, τα πεδία στα οποία ο χρήστης μπορεί να κάνει επιλογές στη φόρμα "FrmMotoEpiologes", είναι τα εξής: "Τίτλος", "ΑΦ\_Αρχείου", "Πόλη", "Υπεύθυνος" και "Ιδιοκτήτες".

#### 4.2.4 "Ταξιδιωτικά Γραφεία" (Travel)

Περιλαμβάνει τα ταξιδιωτικά γραφεία που λειτουργούν στο νομό Χανίων. Σε σχέση με τα "Ξενοδοχεία" απουσιάζουν τα πεδία "Είδος", "Κατηγορία" και "Τιμή" του πίνακα "Travel", καθώς και ο πίνακας "Ανέσεις". Έτσι, οι επιλογές που μπορεί να κάνει ο χρήστης στη φόρμα "FrmTravelEpiologes", περιορίζονται στα πεδία των δύο προηγούμενων τμημάτων (Cars και Moto).

#### 4.2.5 "Διάφορα Άλλα" (More)

Περιλαμβάνει τα εξής είδη επιχειρήσεων που λειτουργούν στο νομό Χανίων: "Bar", "Disco", "Club", "Καφετέρια", "Restaurant", "Κατάστημα", "Ζαχαροπλαστείο", "Ταβέρνα" και "Μικροπωλητές - Άγρα". Η μοναδική διαφορά από το τμήμα που εξετάσαμε προηγουμένως είναι η ύπαρξη του πεδίου "Είδος" στον πίνακα "More".

---

\* Για παράδειγμα, στη βάση δεδομένων "Δωμάτια" ο πίνακας που περιέχει τα στοιχεία των ιδιοκτητών ονομάζεται "RoomsIdioktites". Ο αντίστοιχος πίνακας της βάσης "Ταξιδιωτικά Γραφεία" που περιέχει τα ίδια πεδία (με διαφορετικές τιμές φυσικά) ονομάζεται "TravelIdioktites".

Έτσι, εκτός από τα πεδία "Τίτλος", "ΑΦ\_Αρχείου", "Πόλη", "Υπεύθυνος" και "Ιδιοκτήτες", ο χρήστης μπορεί να κάνει επιλογή στη φόρμα "FrmMoreEpiloges" και στο πεδίο "Είδος".

++

## 5. ΕΦΑΡΜΟΓΗ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

Ο τρόπος με τον οποίο λειτουργεί η εφαρμογή είναι εξαιρετικά απλός και φιλικός προς τον χρήστη, μια και η ροή της ελέγχεται πολύ εύκολα και με το πάτημα δύο ή τριών κουμπιών τόσο η εισαγωγή όσο και η επιλογή κάποιων δεδομένων γίνεται παιχνίδι. Επιπλέον, μένοντας για λίγα δευτερόλεπτα πάνω σε οποιοδήποτε από αυτά τα κουμπιά, εμφανίζεται ένα μήνυμα που εξηγεί τι ακριβώς κάνει (ποιά είναι δηλαδή η λειτουργία του), καθοδηγώντας έτσι τον χρήστη και αποτρέποντας τυχόν λάθη, ενώ κάτι ανάλογο γίνεται και στις φόρμες όπου γίνονται οι επιλογές, με "Labels" (Ετικέτες) οι οποίες εξηγούν ποιες γίνονται ταυτόχρονα και ποιες ανεξάρτητα απ' όλες τις άλλες (καθεμιά χωριστά).

Στην πραγματικότητα, η μοναδική διαφορά στην λειτουργία των έξι τμημάτων της εφαρμογής είναι η απουσία επιλογών που πραγματοποιούνται ταυτόχρονα στα τμήματα "Cars", "Moto", "Travel" και "More". Για το λόγο αυτό θα αναφερθούν μόνο τα "Ξενοδοχεία", μια και ο τρόπος λειτουργίας όλων των άλλων τμημάτων είναι πανομοιότυπος, ενώ όσα θεωρητικά αναφέρονται στη συνέχεια φαίνονται και οπτικά στο τέλος κάθε παραγράφου, όπου δείχνεται ο τρόπος με τον οποίο γίνεται εισαγωγή, αναζήτηση, εκτύπωση πληροφοριών και γενικά οι δυνατότητες της εφαρμογής.

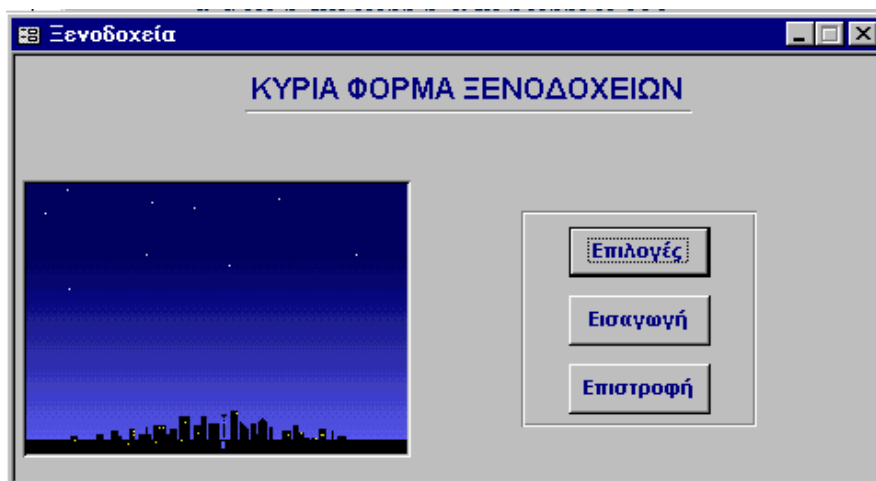
### 5.1 *Κύρια Φόρμα της Εφαρμογής*

Πρόκειται για την πρώτη φόρμα που εμφανίζεται στην οθόνη όταν ανοίγει η εφαρμογή (ο "Κεντρικός Πίνακας Ελέγχου"). Περιλαμβάνει τα κουμπιά εντολών που ανοίγουν τις κύριες φόρμες των έξι τμημάτων της εφαρμογής (ονομάζονται ανάλογα με το ποιο τμήμα ανοίγουν), καθώς και ένα έβδομο κουμπί ("Έξοδος") το οποίο χρησιμοποιείται για την έξοδο του χρήστη από την εφαρμογή. Έτσι, πατώντας το κουμπί "Ξενοδοχεία" ανοίγει η κύρια φόρμα των ξενοδοχείων, με το κουμπί "Δωμάτια" ανοίγει η κύρια φόρμα των δωματίων κ.ο.κ.

Περιέχει επίσης και έναν χάρτη της Κρήτης με τις κυριότερες πόλεις του νομού Χανίων, ο οποίος βοηθάει τον χρήστη να φιλτράρει τα δεδομένα του γρήγορα, εφόσον ενδιαφέρεται απλά για τις επιχειρήσεις μιας συγκεκριμένης πόλης. Για παράδειγμα, για την εμφάνιση των ταξιδιωτικών γραφείων που λειτουργούν στη Σούδα, μεταφερόμαστε στο χάρτη και πατάμε την ετικέτα "Σούδα" (στην πραγματικότητα, όπως έχει αναφερθεί στην παράγραφο 4.4.7 πατάμε ένα κουμπί



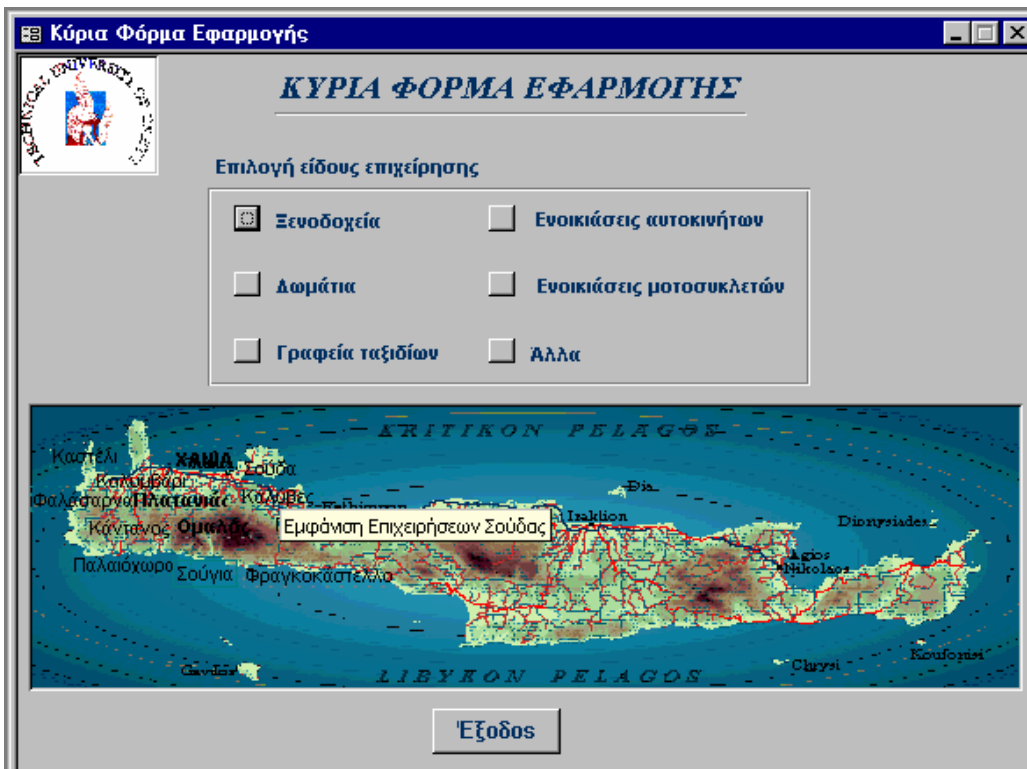
εντολής το οποίο βρίσκεται ακριβώς πάνω από την ετικέτα αλλά δεν φαίνεται στη φόρμα<sup>\*</sup>). Στην συνέχεια ανοίγει η φόρμα "Επιλογή Είδους" όπου πατάμε το κουμπί "Ταξιδιωτικά Γραφεία", και εμφανίζονται τελικά τα στοιχεία που μας ενδιαφέρουν.



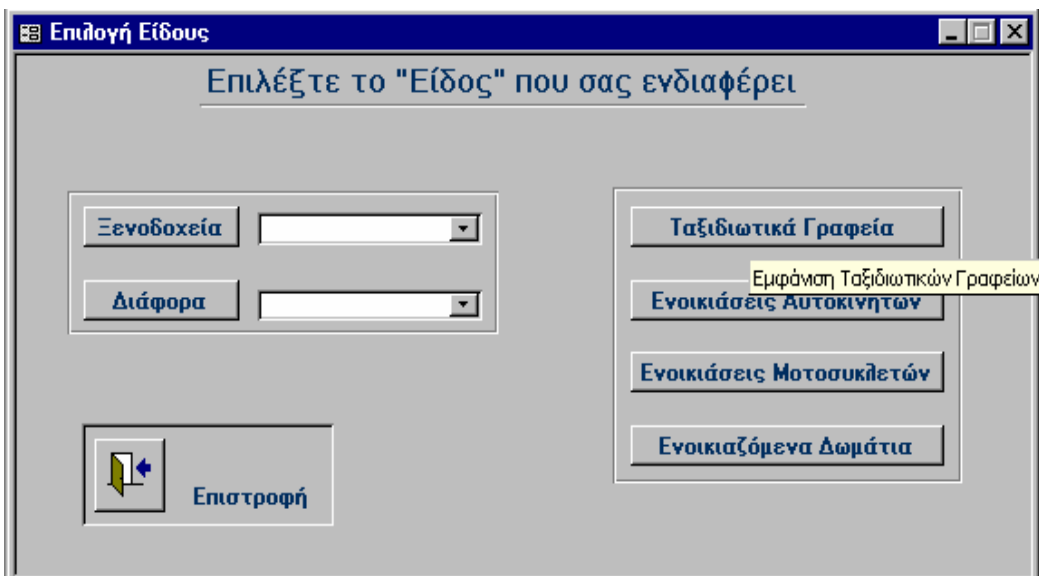
**Εικόνα 5-1:** Πατάμε το κουμπί "Ξενοδοχεία" για να ανοίξει η κύρια φόρμα ξενοδοχείων

**Εικόνα 5-2:** Αφού ανοίξει η κύρια φόρμα ξενοδοχείων μπορούμε να κάνουμε αναζήτηση ή εισαγωγή δεδομένων

<sup>\*</sup> Για το λόγο αυτό, μένοντας για λίγα δευτερόλεπτα πάνω από την ετικέτα "Σούδα" θα εμφανιστεί το μήνυμα "Εμφάνιση Επιχειρήσεων Σούδας", που είναι φυσικά η τιμή που έχει δοθεί στην ιδιότητα "Control Tip Text" του αντίστοιχου κουμπιού εντολής.



**Εικόνα 5-3:** Για να εμφανιστούν τα ταξιδιωτικά γραφεία της "Σούδας", πρώτα απ' όλα πηγαίνουμε στο χάρτη και πατάμε τη λέξη "Σούδα"



**Εικόνα 5-4:** Στη συνέχεια πατάμε το κουμπί "Ταξιδιωτικά Γραφεία" και εμφανίζονται έτσι τα δεδομένα μας (Εικόνα 5-5)

**Εμφάνιση Δεδομένων**

**ΤΑΞΙΔΙΩΤΙΚΑ ΓΡΑΦΕΙΑ - ΕΜΦΑΝΙΣΗ ΣΤΟΙΧΕΙΩΝ**

Τίτλος:

Α.Φ.Μητρώου:

ΔΟΥ:

Α.Φ.Αρχείου:

Οδός:

Αριθμός:

Πόλη:

Τηλέφωνο1:

Τηλέφωνο2:

FAX:

Επιλογή Οδού:

Εμφάνιση Ιδιοκτήτη

Εμφάνιση Υπεύθυνου

ΕΓΓΡΑΦΗ

Επόμενη

Προηγούμενη

Εκτύπωση

Επιστροφή

**Εικόνα 5-5:** Εμφάνιση των δεδομένων

## 5.2 Κύρια Φόρμα των Ξενοδοχείων

Πρόκειται για τον "Πίνακα Ελέγχου" των ξενοδοχείων και ανοίγει όταν από την προηγούμενη φόρμα επιλέγεται το κουμπί "Ξενοδοχεία". Περιλαμβάνει τρία κουμπιά εντολής:

1. "Επιλογές": Ανοίγει την ομόνυμη φόρμα απ' όπου ο χρήστης μπορεί να κάνει τις επιλογές που επιθυμεί.
2. "Εισαγωγή": Ανοίγει τη φόρμα που χρησιμοποιείται για την εισαγωγή δεδομένων.
3. "Επιστροφή": Πατώντας το κουμπί αυτό επιστρέφουμε στην κύρια φόρμα της εφαρμογής.

**Ξενοδοχεία**

**ΚΥΡΙΑ ΦΟΡΜΑ ΞΕΝΟΔΟΧΕΙΩΝ**

Επιλογές

Άνοιγμα φόρμας "Επιλογές"

Επιστροφή

**Εικόνα 5-6:** Πατώντας το κουμπί "Επιλογές" ανοίγει η ομόνυμη φόρμα απ' όπου εφαρμόζουμε τα φίλτρα που θέλουμε

**Εικόνα 5-7:** Φόρμα "Επιλογές"

**Εικόνα 5-8:** Πατώντας το κουμπί "Εισαγωγή" ανοίγει η φόρμα "Εμφάνιση Στοιχείων Ξενοδοχείων:Εισαγωγή" που χρησιμοποιείται για την εισαγωγή δεδμένων

Εικόνα 5-9: Φόρμα "Εμφάνιση Στοιχείων Ξενοδοχείων"

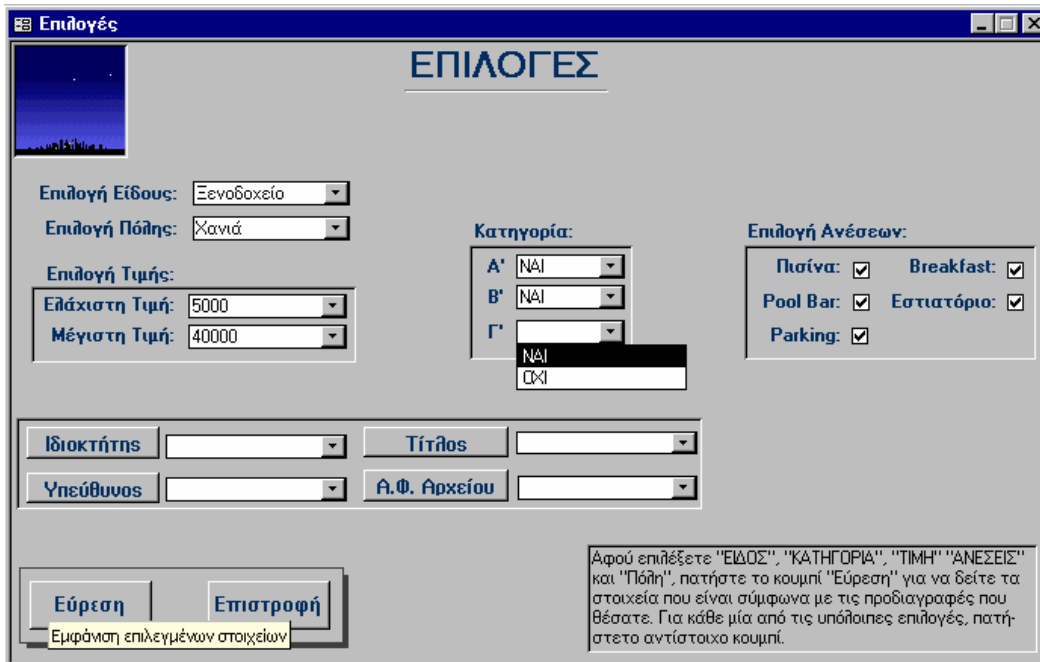
### 5.3 Επιλογές

Πρόκειται για την πιο ενδιαφέρουσα φόρμα της εφαρμογής. Εδώ, ο χρήστης εφαρμόζει τα φίλτρα που θέλει μέσω επιλογών που πραγματοποιεί στους καταλόγους και τα πλαίσια ελέγχου τα οποία εμφανίζουν τις τιμές κάποιων πεδίων. Έτσι, αφού επιλέξει "Είδος", "Πόλη", "Τιμή" (ελάχιστη και μέγιστη), "Κατηγορία" και "Ανέσεις"\*, πρέπει να πατήσει το κουμπί "Εύρεση" προκειμένου να εμφανιστούν οι εγγραφές που συμφωνούν με τις επιλογές του. Στη συνέχεια, στη φόρμα "Εμφάνιση Στοιχείων Ξενοδοχείων" όπου θα εμφανιστούν αυτές οι εγγραφές, μπορεί να επιλέξει αν θέλει κάποια συγκεκριμένη "Οδό" από τον αντίστοιχο κατάλογο και να πατήσει το κουμπί "Επιλογή Οδού" εφαρμόζοντας έτσι έτσι ένα επιπλέον φίλτρο.

Στην περίπτωση που επιλέξει "Ιδιοκτήτη", "Υπεύθυνο", "Τίτλο" ή "ΑΦ\_Αρχείου" (σ' αυτά τα πεδία κάθε επιλογή γίνεται ανεξάρτητα από τις υπόλοιπες) δεν έχει παρά να πατήσει το αντίστοιχο κουμπί εντολής. Για παράδειγμα, αφού επιλεγεί ο τίτλος "Ξενία", πατώντας το κουμπί "Τίτλος" ανοίγει η φόρμα "Εμφάνιση Στοιχείων Ξενοδοχείων:Εμφάνιση" όπου εμφανίζεται η εγγραφή/ές που στο πεδίο "Τίτλος" έχουν την τιμή "Ξενία".

Φυσικά, για την αποφυγή κάποιου λάθους που μπορεί να συμβεί λόγω του διαχωρισμού αυτού στις επιλογές, υπάρχει σχετική ετικέτα που διευκρινίζει πότε χρησιμοποιείται το κουμπί "Εύρεση" και πότε τα κουμπιά "Ιδιοκτήτης", "Υπεύθυνος",

"Τίτλος" και "ΑΦ\_Αρχείου", όπως υπάρχει και το κουμπί "Επιστροφή" πατώντας το οποίο επιστρέφουμε στην κύρια φόρμα των ξενοδοχείων.

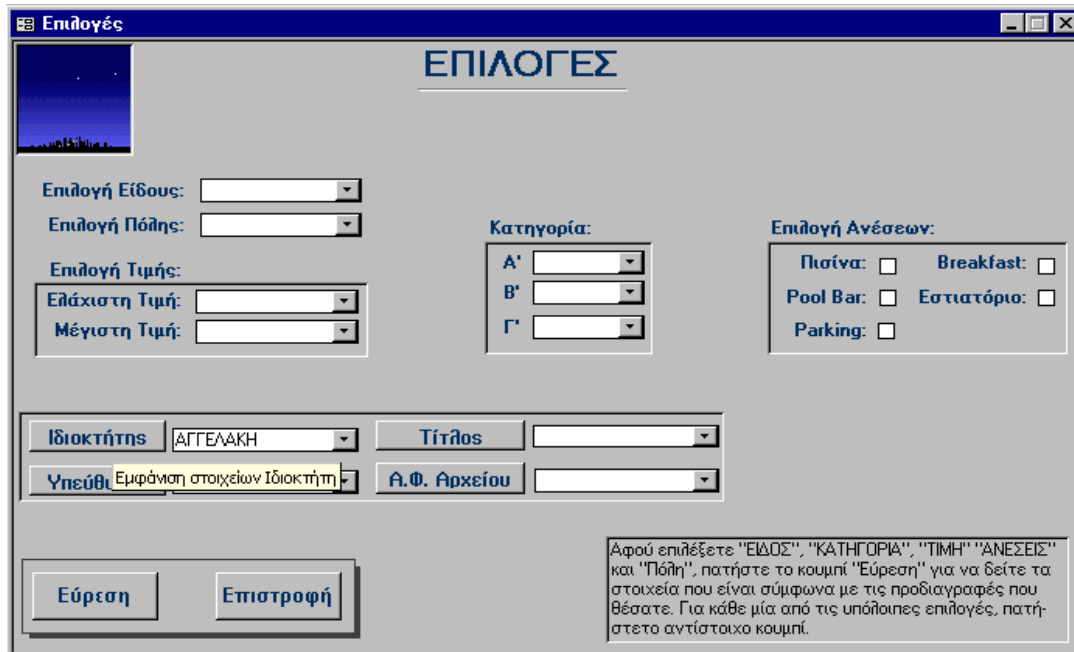


**Εικόνα 5-60:** Επιλέγουμε είδος, πόλη, τιμή, κατηγορία & ανέσεις και πατάμε το κουμπί "Εύρεση" για να ανοίξει η φόρμα "Εμφάνιση Στοιχείων Ξενοδοχείων" όπου εμφανίζονται τα δεδομένα που συμφωνούν με τις επιλογές μας.



**Εικόνα 5-11:** Φόρμα "Εμφάνιση Στοιχείων Ξενοδοχείων"

\* Πρόκειται για τα πεδία στα οποία όπως έχει αναφερθεί οι επιλογές γίνονται ταυτόχρονα.



**Επιλογές**

**ΕΠΙΛΟΓΕΣ**

Επιλογή Είδους:

Επιλογή Πόλης:

Επιλογή Τιμής:

Ελάχιστη Τιμή:

Μέγιστη Τιμή:

Κατηγορία:

Α'

Β'

Γ'

Επιλογή Ανέσεων:

Πισίνα:  Breakfast:

Pool Bar:  Εστιατόριο:

Parking:

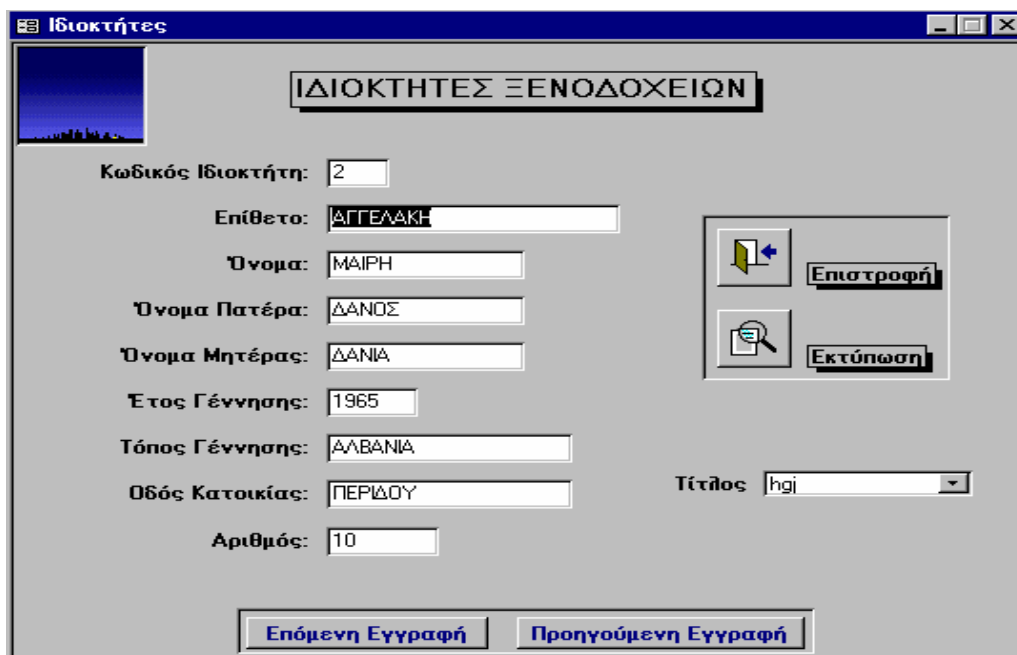
Ιδιοκτήτης:  Τίτλος:

Υπεύθι:  Α.Φ. Αρχείου:

Εύρεση  Επιστροφή

Αφού επιλέξετε "ΕΙΔΟΣ", "ΚΑΤΗΓΟΡΙΑ", "ΤΙΜΗ", "ΑΝΕΣΕΙΣ" και "Πόλη", πατήστε το κουμπί "Εύρεση" για να δείτε τα στοιχεία που είναι σύμφωνα με τις προδιαγραφές που θέσατε. Για κάθε μία από τις υπόλοιπες επιλογές, πατήστε το αντίστοιχο κουμπί.

**Εικόνα 5-12:** Επιλέγουμε ιδιοκτήτη και πατάμε το κουμπί "Ιδιοκτήτης" για να ανοίξει η φόρμα "Ιδιοκτήτες Ξενοδοχείων"



**Ιδιοκτήτες**

**ΙΔΙΟΚΤΗΤΕΣ ΞΕΝΟΔΟΧΕΙΩΝ**

Κωδικός Ιδιοκτήτη:

Επίθετο:

Όνομα:

Όνομα Πατέρα:

Όνομα Μητέρας:

Έτος Γέννησης:

Τόπος Γέννησης:

Οδός Κατοικίας:

Αριθμός:

Τίτλος:

Επιστροφή  Εκτύπωση

Επόμενη Εγγραφή  Προηγούμενη Εγγραφή

**Εικόνα 5-13:** Φόρμα "Ιδιοκτήτες Ξενοδοχείων"

ΕΠΙΛΟΓΕΣ

Επιλογή Είδους:

Επιλογή Πόλης:

Επιλογή Τιμής:

Ελάχιστη Τιμή:

Μέγιστη Τιμή:

Κατηγορία:

A'

B'

Γ'

Επιλογή Ανέσεων:

Πισίνα:  Breakfast:

Pool Bar:  Εστιατόριο:

Parking:

Ιδιοκτήτης:

Υπεύθυνος:

Τίτλος:

Α.Φ. Επιλογή Τίτλου Επιχείρησης:

Εύρεση  Επιστροφή

Αφού επιλέξετε "ΕΙΔΟΣ", "ΚΑΤΗΓΟΡΙΑ", "ΤΙΜΗ" "ΑΝΕΣΕΙΣ" και "Πόλη", πατήστε το κουμπί "Εύρεση" για να δείτε τα στοιχεία που είναι σύμφωνα με τις προδιαγραφές που θέσατε. Για κάθε μία από τις υπόλοιπες επιλογές, πατήστε το αντίστοιχο κουμπί.

**Εικόνα 5-14:** Επιλέγουμε τίτλο και πατάμε το κουμπί "Τίτλος" για να ανοίξει η φόρμα "Εμφάνιση Στοιχείων Ξενοδοχείων:Εμφάνιση"

ΕΜΦΑΝΙΣΗ ΣΤΟΙΧΕΙΩΝ ΞΕΝΟΔΟΧΕΙΩΝ

Τίτλος:  ΑΦΜ:

Είδος:  ΔΟΥ:

Κατηγορία:  ΑΦ\_Αρχείου:

Δωμάτια:	Τιμή
Μονόκλινα:	14 7000
Δίκλινα:	34 12000
Τρίκλινα:	33 25000
Τετράκλινα:	22 33000
Σουίτες:	11 50000
Bungalows:	6 75000

ΣΤΟΙΧΕΙΑ ΕΠΙΧΕΙΡΗΣΗΣ

Οδός:  Τηλέφωνο 1:

Αριθμός:  Τηλέφωνο 2:

Πόλη:  FAX:

ΑΝΕΣΕΙΣ

Πισίνα:  Breakfast:

Pool Bar:  Εστιατόριο:

Parking:

Εμφάνιση Ιδιοκτήτη  ΕΓΓΡΑΦΗ  Επόμενη

Εμφάνιση Υπεύθυνου  Προηγούμενη  Εκτύπωση  Επιστροφή

**Εικόνα 5-15:** Φόρμα "Εμφάνιση Στοιχείων Ξενοδοχείων:Εμφάνιση"



#### 5.4 Εμφάνιση Στοιχείων Ξενοδοχείων

Πρόκειται για τη φόρμα που ανοίγει αφού επιλεγθεί το "Είδος", η "Πόλη", η "Τιμή" (ελάχιστη και μέγιστη), η "Κατηγορία" και οι "Ανέσεις" και πατηθεί το κουμπί "Εύρεση". Εμφανίζει τις εγγραφές που συμφωνούν με τις επιλογές στα πεδία αυτά και όπως αναφέρθηκε προηγουμένως προσφέρει τη δυνατότητα μιας επιπλέον επιλογής στο πεδίο "Οδός". Έτσι, αφού επιλεγθεί από τον αντίστοιχο κατάλογο η επιθυμητή οδός, με το πάτημα του κουμπιού "Επιλογή Οδού" πραγματοποιείται ένα περαιτέρω φιλτράρισμα των δεδομένων. Για τη μετακίνηση του χρήστη από μια εγγραφή σε μια άλλη υπάρχουν τα κουμπιά "Επόμενη" και "Προηγούμενη", ενώ πατώντας τα κουμπιά "Υπεύθυνοι" και "Ιδιοκτήτες" εμφανίζονται τα στοιχεία του υπευθύνου και ιδιοκτήτη αντίστοιχα της τρέχουσας εγγραφής. Τέλος, με το κουμπί "Εκτύπωση" ανοίγει η αναφορά με τίτλο "Στοιχεία Επιχείρησης" σε άποψη προεπισκόπησης εκτύπωσης,\* ενώ με το κουμπί "Επιστροφή" επιστρέφουμε στη φόρμα "Επιλογές".

**Εικόνα 5-76:** Για να εμφανιστούν οι επιχειρήσεις που υπάρχουν σε μία συγκεκριμένη οδό (π.χ. στην

σφακίων) επιλέγουμε την οδό από τον κατάλογο που βρίσκεται δίπλα στο κουμπί "Επιλογή Οδού", το οποίο στη συνέχεια πατάμε

\* Στη συνέχεια, για την εκτύπωση των δεδομένων απλά πατάμε το κουμπί "Print" της μπάρας εργαλείων .

**Ξενοδοχεία-Εμφάνιση Δεδομένων**

**ΕΜΦΑΝΙΣΗ ΣΤΟΙΧΕΙΩΝ ΞΕΝΟΔΟΧΕΙΩΝ**

Τίτλος: γφ ΑΦΜ: γηγφ  
 Είδος: Ξενοδοχείο ΔΟΥ: Α' ΔΟΥ ΧΑΝΙΩΝ  
 Κατηγορία: Α ΑΦ\_Αρχείου: φηηφφφ

Δωμάτια	Τιμή
Μονόκλινα:	3 34545
Δίκλινα:	3 23333
Τρίκλινα:	4 43555
Τετράκλινα:	3 34555
Σουίτες:	3 34555
Bangalows:	3 35555

**ΣΤΟΙΧΕΙΑ ΕΠΙΧΕΙΡΗΣΗΣ**

Οδός: Σφακίων Τηλέφωνο1: 34444  
 Αριθμός: 13 Τηλέφωνο2: 33333  
 Πόλη: Χανιά FAX: 33333

Επίλογη Οδού: Σφακίων

**ΑΝΕΣΕΙΣ**

Παίνα:  Breakfast:   
 Pool Bar:  Εστιατόριο:   
 Parking:

Εμφάνιση Ιδιοκτήτη Επόμενη  
 Εμφάνιση Υπεύθυνου Εμφάνιση στοιχείων Ιδιοκτήτη Προηγούμενη  
 Εκτύπωση Επιστροφή

**Εικόνα 5-17:** Αφού εμφανιστούν οι επιχειρήσεων που βρίσκονται στην οδό σφακίων μπορούμε να δούμε τα στοιχεία του ιδιοκτήτη της τρέχουσας εγγραφής πατώντας το κουμπί "Εμφάνιση Ιδιοκτήτη"

**Ιδιοκτήτες**

**ΙΔΙΟΚΤΗΤΕΣ ΞΕΝΟΔΟΧΕΙΩΝ**

Κωδικός Ιδιοκτήτη: 23  
 Επίθετο: Γεωργίου  
 Όνομα: Κώστας  
 Όνομα Πατέρα: Νίκος  
 Όνομα Μητέρας: Άννα  
 Έτος Γέννησης: 1965  
 Τόπος Γέννησης: Αθήνα  
 Οδός Κατοικίας: Δημοκρατίας Τίτλος: Caravel  
 Αριθμός: 34

Επιστροφή Εκτύπωση

Επόμενη Εγγραφή Προηγούμενη Εγγραφή

**Εικόνα 5-18:** Εμφάνιση των στοιχείων ιδιοκτήτη στη φόρμα "Ιδιοκτήτες Ξενοδοχείων"

ΣΤΟΙΧΕΙΑ ΕΠΙΧΕΙΡΗΣΗΣΌνομα CaravelΕίδος ΞενοδοχείοΚατηγορία AΌδος ΔημοκρατίαςΑριθμός 18Πόλη ΧανιάΤηλέφωνο 1 44655Τηλέφωνο 2 44678ΦΑΧ 44667ΑΦΜ 3344344-34340444ΔΟΥ Α' ΔΟΥΧΑΝΙΩΑΦ Αρχείου 3242434-2424234/3424

<u>Τιμή Μονόκλινο</u>	7000	<u>Δωμάτια Μονόκλινο</u>	12
<u>Τιμή Δίκλινο</u>	10000	<u>Δωμάτια Δίκλινα</u>	15
<u>Τιμή Τρίκλινο</u>	15000	<u>Δωμάτια Τρίκλινα</u>	21
<u>Τιμή Τετράκλινο</u>	21500	<u>Δωμάτια Τετράκλινο</u>	10
<u>Τιμή Σουίτες</u>	28000	<u>Σουίτες</u>	8
<u>Τιμή Bangalow</u>	26000	<u>Bangalows</u>	10

ΠΙΣΙΝΑ YesPOOL BAR YesParking YesBreakfast YesΕστιατόριο Yes

**Εικόνα 5-19:** Πατώντας το κουμπί "Εκτύπωση" ανοίγει η αναφορά με τίτλο "Στοιχεία Επιχείρησης" σε άποψη προεπισκόπησης εκτύπωσης

### **5.5 Εμφάνιση Στοιχείων Ξενοδοχείων:Εμφάνιση**

Η φόρμα αυτή λειτουργεί ακριβώς όπως και η προηγούμενη. Η μοναδική διαφορά έγκειται στο ότι ανοίγει, όταν στη φόρμα "Επιλογές" επιλέγεται "Τίτλος" ή "ΑΦ\_Αρχείου", πεδία δηλαδή στα οποία η κάθε εφαρμογή φίλτρου γίνεται ανεξάρτητα από τις υπόλοιπες.

### **5.6 Εμφάνιση Στοιχείων Ξενοδοχείων:Εισαγωγή**

Κάθε φορά που πατάμε το κουμπί "Εισαγωγή" της κύριας φόρμας ξενοδοχείων ανοίγει η φόρμα με τίτλο "Εμφάνιση Στοιχείων Ξενοδοχείων:Εισαγωγή", η οποία είναι έτοιμη για την εισαγωγή νέων δεδομένων. Στην περίπτωση μάλιστα των πεδίων "Είδος", "Κατηγορία", "Πόλη", "Οδός" και "ΔΟΥ" τα οποία παίρνουν κάποιες συγκεκριμένες τιμές, εμφανίζονται κατάλογοι από τους οποίους ο χρήστης απλά επιλέγει τις τιμές αυτές, ενώ αν αντί για την επιλογή από τη λίστα πληκτρολογηθεί μία τιμή που δεν περιλαμβάνεται σ' αυτήν, εμφανίζεται ένα μήνυμα λάθους (αυτό γίνεται , αν για παράδειγμα ο χρήστης στον κατάλογο "Είδος" δώσει την τιμή "Ξανώνας" αντί για "Ξενώνας". Τότε, εμφανίζεται το μήνυμα λάθους και ο χρήστης πρέπει να επιλέξει τη σωστή τιμή από τον κατάλογο).

Επίσης, για την εισαγωγή μίας πόλης ή οδού που όντως υπάρχει στο νομό Χανίων αλλά δεν περιέχεται στον αντίστοιχο κατάλογο, υπάρχει η δυνατότητα ενημέρωσής του. Για παράδειγμα, αν ο χρήστης θελήσει να επιλέξει από τη λίστα των πόλεων τη "Σούγια" και η τιμή αυτή δεν περιέχεται, κάνοντας "double clic" (πατώντας δηλαδή δύο συνεχόμενες φορές) στο πλαίσιο του καταλόγου ανοίγει η φόρμα με τίτλο "Εισαγωγή Πόλης", που περιέχει τις πόλεις που έχουν καταχωρηθεί μέχρι εκείνη τη στιγμή, η οποία πηγαίνει αυτόματα σε νέα εγγραφή. Εκεί, εισάγεται το όνομα της πόλης ("Σούγια" στο παράδειγμα) και αφού κλείσει η φόρμα (πατώντας το κουμπί "Επιστροφή") ενημερώνεται αυτόματα ο κατάλογος και μπορεί πλέον ο χρήστης να επιλέξει απ' αυτόν την τιμή "Σούγια".

Αφού λοιπόν περαστούν οι τιμές των πεδίων που εμφανίζονται στην οθόνη, πατώντας τα κουμπιά "Υπεύθυνοι" και "Ιδιοκτήτες" ο χρήστης είναι σε θέση να περάσει τα στοιχεία του υπεύθυνου και του/των ιδιοκτητών της τρέχουσας εγγραφής. Ειδικά στην περίπτωση των ιδιοκτητών, αφού εισαχθούν τα στοιχεία κάποιου ιδιοκτήτη, πρέπει από τον κατάλογο που εμφανίζεται να επιλεγθεί ο τίτλος της επιχείρησης στην

οποία είναι ιδιοκτήτης. Για παράδειγμα, αφού εισαχθούν τα στοιχεία του ξενοδοχείου με τίτλο "Σαμαριά" και πατηθεί το κουμπί "Ιδιοκτήτες", μετά την εισαγωγή των στοιχείων του ιδιοκτήτη του ξενοδοχείου αυτού θα πρέπει από τον κατάλογο που αναφέρθηκε να επιλεγεί ο τίτλος "Σαμαριά" (ο οποίος προφανώς θα είναι τελευταίος στη λίστα). Αν τώρα υπάρχει και δεύτερος ή και τρίτος ιδιοκτήτης, ακολουθώντας την ίδια ακριβώς διαδικασία θα πρέπει για κάθε έναν να περάσουμε τα στοιχεία του και να επιλέξουμε από τον κατάλογο τον τίτλο "Σαμαριά". Αφού περαστούν και τα στοιχεία ιδιοκτήτη(-των) και υπεύθυνου, επιστρέφοντας στη φόρμα που εξετάζεται μπορούμε να προχωρήσουμε στην εισαγωγή της επόμενης εγγραφής, πατώντας απλά το κουμπί "Νέα Καταχώρηση" (οπότε πηγαίνουμε σε νέα εγγραφή). Δεν πρέπει να ξεχνάμε φυσικά πως η φόρμα αυτή χρησιμοποιείται αποκλειστικά για την εισαγωγή δεδομένων. Έτσι, αν ο χρήστης πατήσει στην κύρια φόρμα ξενοδοχείων το κουμπί "Εισαγωγή" θα πρέπει να εισάγει κάποια δεδομένα. Αν κάτι τέτοιο δεν γίνει και βγει αμέσως από τη παρούσα φόρμα, θα εμφανιστεί ένα μήνυμα λάθους το οποίο επισημαίνει τόσο το σκοπό της όσο και τον τρόπο σωστής λειτουργίας της.

Τα υπόλοιπα κουμπιά εντολής της παρούσας φόρμας αναφέρθηκαν στις δύο (παρόμοιες) προηγούμενες. Το τελευταίο σημείο που αξίζει να αναφερθεί, είναι πως στην περίπτωση που ο χρήστης διαγράψει μία εγγραφή (για παράδειγμα ένα οικοτροφείο που σταμάτησε η λειτουργία του) με τη βοήθεια του κουμπιού "Delete Record" της γραμμής εργαλείων, δεν θα πρέπει να ξεχάσει να πατήσει τα κουμπιά "Υπεύθυνοι" και "Ιδιοκτήτες" προκειμένου να διαγράψει τον υπεύθυνο όπως και τον ιδιοκτήτη της εγγραφής αυτής.

ΞενοδοχείοID: 33 ΑΦΜ: 342423432432  
 Τίτλος: Σαμαριά ΔΟΥ: Α' ΔΟΥ ΧΑΝΙΩΝ  
 Είδος: Ξενοδοχείο ΑΦ\_Αρχείου: 3232-34434/34  
 Κατηγορία: Α

Δωμάτια:	Τιμή
Μονόκλινα:	10 5000
Δίκλινα:	20 10000
Τρίκλινα:	10 15000
Τετράκλινα:	15 20000
Σουίτες:	5 30000
Bungalows:	10 35000

Οδός: [Dropdown]  
 Αριθμός: [Input] Τηλέφωνο 1: 22222  
 Πόλη: [Dropdown] Τηλέφωνο 2: 33333 FAX: 44444

Πισίνα:  Breakfast:   
 Pool Bar:  Εστιατόριο:   
 Parking:

Buttons: Νέα Καταχώρηση, Εισαγωγή Ιδιοκτήτη, Εισαγωγή Υπεύθυνου, ΕΓΓΡΑΦΗ, Επόμενη, Προηγούμενη, Επιστροφή

**Εικόνα 5-20:** Για να εισάγουμε μία οδό που δεν υπάρχει στον κατάλογο (π.χ. Παπαναστασίου), κάνουμε double clic στο λευκό πλαίσιο δίπλα στην ετικέτα "Οδός"

ΞενοδοχείοID: 33 ΑΦΜ: 342423432432  
 Τίτλος: Σαμαριά ΔΟΥ: Α' ΔΟΥ ΧΑΝΙΩΝ  
 Είδος: Ξενοδοχείο ΑΦ\_Αρχείου: 3232-34434/34  
 Κατηγορία: Α

Δωμάτια:	Τιμή
Μονόκλινα:	10 5000
Δίκλινα:	20 10000
Τρίκλινα:	10 15000
Τετράκλινα:	15 20000
Σουίτες:	5 30000
Bungalows:	10 35000

Οδός: [Dropdown] Αριθμός: 10 Τηλέφωνο 1: 22222 Τηλέφωνο 2: 33333 FAX: 44444  
 Πόλη: Χανιά

Πισίνα:  Breakfast:   
 Pool Bar:  Εστιατόριο:   
 Parking:

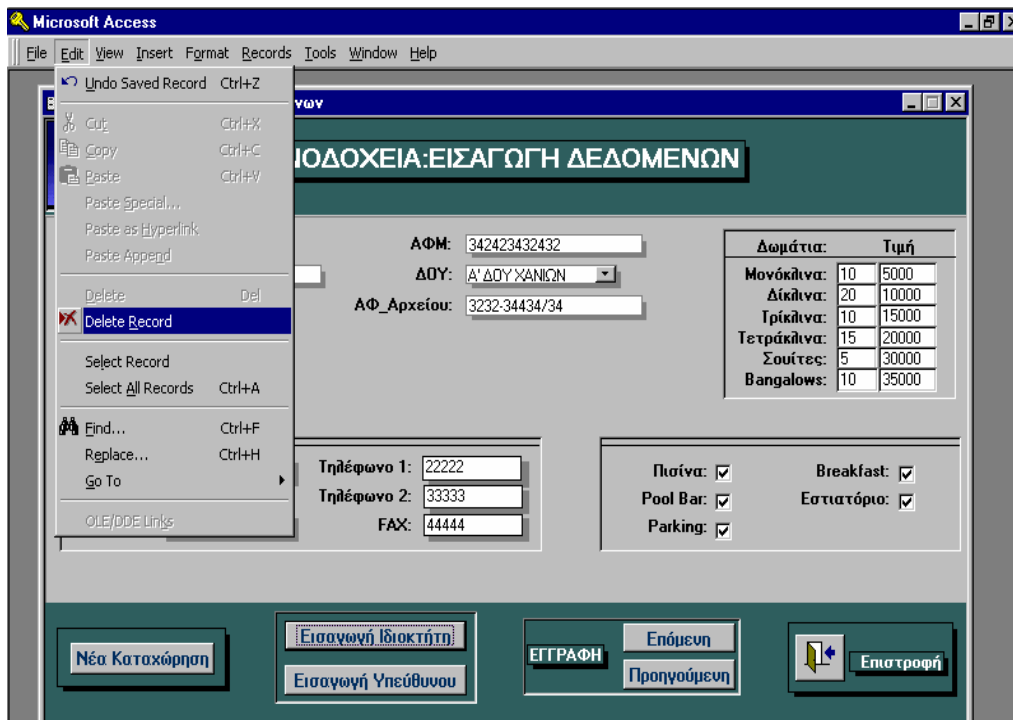
Dialog Box: Streets - Εισαγωγή Οδού  
 Όνομα Οδού: Παπαναστασίου  
 Button: Επιστροφή  
 Tooltip: Επιστροφή στη λίστα οδών

Buttons: Νέα Καταχώρηση, Εισαγωγή Ιδιοκτήτη, Εισαγωγή Υπεύθυνου, ΕΓΓΡΑΦΗ, Επόμενη, Προηγούμενη, Επιστροφή

**Εικόνα 5-28:** Στη φόρμα "Εισαγωγή Οδού" που ανοίγει πληκτρολογούμε την τιμή "Παπαναστασίου"

**Εικόνα 5-92:** Η λίστα οδών ενημερώνεται και μπορούμε πλέον να εισάγουμε την τιμή "Παπαναστασίου"

**Εικόνα 5-210:** Στη συνέχεια, πατώντας το κουμπί "Εισαγωγή Ιδιοκτήτη", εισάγουμε τα στοιχεία ιδιοκτήτη, επιλέγοντας από τη λίστα τίτλων την τιμή "Σαμαριά"



**Εικόνα 5-211:** Για να σβήσουμε μία εγγραφή, επιλέγουμε από τον κατάλογο "Edit" την εντολή "Delete Record" (το ίδιο φυσικά θα πρέπει να γίνει και με τον ιδιοκτήτη και υπεύθυνο της εγγραφής αυτής)

### 5.7 Υπεύθυνοι - Ιδιοκτήτες

Η φόρμα "Υπεύθυνοι" χρησιμοποιείται είτε για την εισαγωγή των στοιχείων κάποιου υπευθύνου (όταν από την προηγούμενη φόρμα πατηθεί το κουμπί "ΥΠΕΥΘΥΝΟΙ"), είτε για την εμφάνιση των στοιχείων του (όταν από τη φόρμα "Επιλογές" επιλεχθεί και πατηθεί το κουμπί "Υπεύθυνος). Περιέχει τα κουμπιά "Επόμενη" και "Προηγούμενη" (εγγραφή) που χρησιμοποιούνται για τη μετακίνηση του χρήστη από μια εγγραφή σε μια άλλη και φυσικά τα κουμπιά "Επιστροφή", με το οποία επιστρέφουμε στην προηγούμενη φόρμα (στις "Επιλογές" ή στην "Εμφάνιση Στοιχείων Ξενοδοχείων:Εισαγωγή"), και "Εκτύπωση" με το οποίο ανοίγει η αναφορά με τίτλο "Στοιχεία Υπεύθυνου" σε άποψη προεπισκόπησης εκτύπωσης (για την εκτύπωση των δεδομένων, πάλι πατάμε το κουμπί "Print" της μπάρας εργαλείων).

Τέλος, ακριβώς ίδιος είναι ο τρόπος λειτουργίας της φόρμας "Ιδιοκτήτες", με μοναδική διαφορά την ύπαρξη του καταλόγου με τις τιμές του πεδίου "Τίτλος", ο οποίος όπως αναφέρθηκε στην προηγούμενη παράγραφο χρησιμοποιείται στην εισαγωγή των στοιχείων των ιδιοκτητών.



## 6. ΣΥΜΠΕΡΑΣΜΑΤΑ

Αφού ολοκληρώθηκε η κατασκευή της εφαρμογής, έπρεπε να επιβεβαιωθεί η επίτευξη των στόχων που τέθηκαν στην "Ανάλυση του προβλήματος" και γενικά να διερευνηθούν οι δυνατότητές της. Για το λόγο αυτό, σε καθένα από τα τμήματά της πραγματοποιήθηκε ένας εξαιρετικά μεγάλος αριθμός εισαγωγής δεδομένων και εφαρμογής φίλτρων, τα οποία κάλυπταν όλες τις δυνατές περιπτώσεις που μπορούν να τύχουν (όπως για παράδειγμα, ένα ξενοδοχείο με πολλούς ιδιοκτήτες ή ένας ιδιοκτήτης με πολλά ξενοδοχεία). Έτσι, τα αποτελέσματα έδειξαν πως η εισαγωγή δεδομένων, ειδικά με τη ύπαρξη των πλαισίων καταλόγου με τα οποία ο χρήστης δεν γράφει καν τις τιμές που θέλει αλλά απλά τις επιλέγει από τον αντίστοιχο κατάλογο και εισάγονται αυτόματα, γίνεται ταχύτατα, ενώ τα μηνύματα που εμφανίζονται σε όλες τις φόρμες πάνω από τα κουμπιά εντολής και περιγράφουν τη λειτουργία τους εκμηδενίζουν την πιθανότητα λάθους. Όσο αφορά την εφαρμογή φίλτρων, είτε επιλεγεί απλά ένας ιδιοκτήτης, είτε γίνει μία σύνθετη επιλογή (πόλης, είδους, κατηγορίας, εύρους τιμής και ανέσεων ταυτόχρονα), εμφανίζονται στην οθόνη οι σωστές εγγραφές, εξαιρετικά γρήγορα και με το πάτημα δύο ή τριών κουμπιών. Αυτό το τελευταίο είναι εξαιρετικά σημαντικό, μια και αποτελεί ουσιαστικά το κύριο ζητούμενο: την εκτέλεση μιας εργασίας, που απαιτεί από πολλά λεπτά μέχρι και μερικές ώρες δουλειάς, , εύκολα και κυρίως γρήγορα (σε ελάχιστα δευτερόλεπτα, για την ακρίβεια). Φυσικά, υπάρχει και η άκρως λειτουργική κύρια φόρμα της εφαρμογής (ο πίνακας ελέγχου της), όπου με τις τροποποιήσεις που έγιναν στο χάρτη που περιλαμβάνει, πέρα από τη δημιουργία ενός όμορφου "εφέ" έγινε μία επιτυχημένη προσπάθεια "να γίνει το γρήγορο ακόμα πιο γρήγορο". Όσο αφορά την επίτευξη ή όχι του στόχου "όμορφη εμφάνιση των φορμών και αναφορών της εφαρμογής", η απάντηση φαίνεται στο κεφάλαιο "Λειτουργία της εφαρμογής", όπου επίσης φαίνεται η λειτουργικότητά τους, η φιλικότητα προς το χρήστη (εξαιρετικά εύκολος τρόπος χρήσης) καθώς και οι πολλές δυνατότητες της εφαρμογής.

Φυσικά, αν και οι απαιτήσεις της Τουριστικής Αστυνομίας Χανίων ικανοποιήθηκαν και οι δυνατότητες της εφαρμογής στην τελική της μορφή είναι πολύ περισσότερες απ' ότι στην πρώτη της έκδοση, πάντα υπάρχει η δυνατότητα βελτίωσης αν

μελλοντικά προκύψουν κάποιες νέες ανάγκες. Για παράδειγμα, είδαμε πως ο χρήστης μπορεί να κάνει τριών ειδών επιλογές:

1. Σύνθετη (ταυτόχρονη) επιλογή στις τιμές των πεδίων "Πόλη", "Είδος", "Κατηγορία", "Τιμή" (ελάχιστη και μέγιστη), "Πισίνα", "Pool bar", "Breakfast", "Parking" και "Εστιατόριο", στη φόρμα "Επιλογές".
2. Ξεχωριστή (μία κάθε φορά) επιλογή στις τιμές των πεδίων "Επίθετο" (Ιδιοκτήτη/Υπεύθυνου), "Τίτλος" και "ΑΦ\_Αρχείου", επίσης στη φόρμα "Επιλογές", και "Οδός", στη φόρμα "Εμφάνιση Στοιχείων Ξενοδοχείων".
3. Επιλογή "Πόλης" και "Είδους" με τη βοήθεια του χάρτη στην κύρια φόρμα της εφαρμογής.

Έτσι, μελλοντική βελτίωση της εφαρμογής θα μπορούσε να επιτευχθεί με τη δυνατότητα περισσότερο σύνθετων ή και πιο πολλών συνδιασμών επιλογών, όπως ταυτόχρονη επιλογή ανέσεων και τιμής, πόλης και κατηγορίας, ανέσεων τιμής και πόλης κλπ. Επίσης, όπως αναφέρθηκε στην ανάλυση του πίνακα "Ξενοδοχεία", με τη χρήση του πεδίου "Photo" (τύπου OLE Object) έχει προβλεφθεί η εμφάνιση φωτογραφίας των διαφόρων επιχειρήσεων. Πεδία αυτού του τύπου θα μπορούσαν να χρησιμοποιηθούν για την εμφάνιση επιπλέον φωτογραφιών (όπως δωματίων ή ανέσεων) ή ακόμα και για την προβολή video. Τέλος, μια πολύ καλή προσπάθεια βελτίωσης θα ήταν ο εμπλουτισμός των αναφορών που εμφανίζουν τα αποτελέσματα των επιλογών του χρήστη, με γραφήματα τα οποία θα παρουσιάζουν τον αριθμό επιχειρήσεων σαν συνάρτηση της τιμής ή της πόλης.

Τελειώνοντας, θα ήταν σοβαρή παράλειψη να μην αναφερθώ σε κάτι που κατά τη διάρκεια κατασκευής της εφαρμογής έκανε πολύ μεγάλη εντύπωση: οι μεγάλες δυνατότητες της Microsoft Access 95. Μ' αυτήν, η δημιουργία φορμών και αναφορών έγινε παιχνίδι (κυριολεκτικά "ζωγραφίζοντας" στην οθόνη), η εύκολη δημιουργία σχέσεων μεταξύ πινάκων και κατά συνέπεια η εύκολη δημιουργία ερωτημάτων συνέβαλαν στην εύκολη ανάπτυξη της εφαρμογής, ενώ ακόμα πιο μεγάλη ήταν η συμβολή των μακροεντολών τόσο στην ανάπτυξη όσο και τον έλεγχο ροής της εφαρμογής. Μάλιστα, δεν χρειάστηκε καθόλου η γλώσσα προγραμματισμού Microsoft Access Basic (κάτι που αρχικά θεωρούσα απαραίτητο να γίνει), κι αυτό χάρη στην πολύ μεγάλη ισχύ των μακροεντολών αλλά και της Access γενικότερα. Πέρα δηλαδή από τη θεωρία επιβεβαιώθηκε και στην πράξη ότι η επιλογή του

"εργαλείου" που ονομάζεται Microsoft Access για τη δημιουργία της εφαρμογής ήταν η καλύτερη δυνατή.

## 7. ΒΙΒΛΙΟΓΡΑΦΙΑ

1. E.F.Codd, "A Relational Model for Large Shared Data Banks," *Communications of the ACM* 13, Vol 6: 377-387 (1970).
2. E.F.Codd, "Relational Database: A Practical Foundation for Productivity," *Communications of the ACM* 25 Vol 2: 109-117 (1982).
3. E.F.Codd, *Normalised Data Base Structure: A Brief Tutorial*, IBM Research RJ935, 1971.
4. E.F.Codd, *Recent Investigations in Relational Data Base Systems*, IBM Research RJ1385, 1974.
5. C. J. Date, *An Introduction to Database Systems*, Addison Wesley Publishing Company, 1975.
6. S. M. Dean, *Fundamentals of Data Base Systems*, Macmillan Education Ltd, 1977.
7. George U. Hubbard, *IMS (DL/I) DATABASE ORGANIZATION AND PERFORMANCE*, Van Nostrand Reinhold Company Inc., 1986.
8. Roger Jennings, *USING ACCESS 2 for Windows*, Que Corporation, 1994.
9. Patrick O'Neil, *DATA BASE PRINCIPLES-PROGRAMMING-PERFORMANCE*, Morgan Kaufmann Publishers Inc, 1994.
10. Cary N. Prague and Michael R. Irwin, *H ΒΙΒΛΙΟΣ ΤΗΣ Microsoft ACCESS 2*, Εκδ. Μ. Γκιούρδας, 1995.
11. John Preston, Sally Preston, and Robert Ferrett, *ACCESS ΓΙΑ WINDOWS 95*, Β. Γκιούρδας Εκδοτική, 1995.
12. John L. Viescas, *Ο ΟΔΗΓΟΣ ΤΗΣ MICROSOFT ΓΙΑ ΤΗΝ ACCESS 2 FOR WINDOWS, ΚΛΕΙΔΑΡΙΘΜΟΣ*, 1994.
13. Richard T. Watson, *DATA MANAGEMENT - AN ORGANIZATIONAL PERSPECTIVE*, John Wiley & Sons Inc., 1996.

