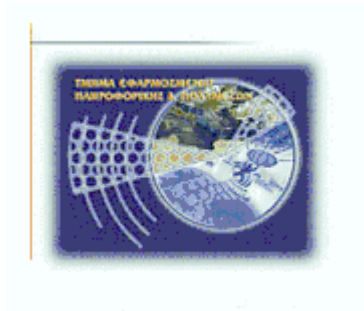




Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Πτυχιακή εργασία

**Τίτλος: ΔΗΜΙΟΥΡΓΙΑ ONLINE ΣΥΣΤΗΜΑΤΟΣ ΓΙΑ
ΑΛΥΣΙΔΑ SUPERMARKET**

Κεφαλογιάννης Εμμανουήλ (ΑΜ: 1551)

Ξανθουδάκης Αριστοτέλης (ΑΜ: 1667)

Επιβλέπων καθηγητής : Παπαδάκης Νικόλαος

**Επιτροπή Αξιολόγησης : Κορνάρος Γεώργιος
Παναγιωτάκης Σπυρίδων**

Ημερομηνία παρουσίασης: 20/09/2011

Ευχαριστίες:

Μέσω αυτού του βήματος επιθυμούμε να ευχαριστήσουμε όλους όσους διαδραμάτισαν κάποιο ρόλο σε αυτή την πτυχιακή εργασία:

- Στον καθηγητή κ. Παπαδάκη Νικόλαο για την εμπιστοσύνη που μας επέδειξε με την ανάθεση της πτυχιακής εργασίας και για την πολύτιμη βοήθειά του και τη συνεχή καθοδήγηση.
- Στις οικογένειές μας που στηρίζουν πάντα τις επιλογές μας και μας δίνουν δύναμη να προχωρήσουμε.

Abstract:

At the following project we created a web-based page which connects to a database and is used to manage a chain of super-markets. The main technologies used to achieve this goal are HTML, PHP and JavaScript for the development of our website and SQL for database management. Our site connects directly with suppliers and warehouses for ease in handling our products.

Σύνοψη:

Στην παρακάτω πτυχιακή εργασία δημιουργήσαμε μια σελίδα web η οποία συνδέεται με μια βάση δεδομένων και χρησιμοποιείται για την διαχείριση μιας αλυσίδας super-market. Οι κύριες τεχνολογίες που χρησιμοποιήσαμε για την επίτευξη του στόχου αυτού είναι η HTML, η PHP και η JavaScript για την ανάπτυξη της ιστοσελίδας μας και η SQL για την διαχείριση της βάσης δεδομένων. Η σελίδα μας συνδέεται άμεσα με τους προμηθευτές και τις αποθήκες για μεγαλύτερη ευκολία στην διαχείριση των προϊόντων μας.

Πίνακας Περιεχομένων

1. ΕΙΣΑΓΩΓΗ:	9
1.1. Περίληψη.....	9
1.2. Κίνητρο για την Διεξαγωγή της Εργασίας.....	9
1.3. Σκοπός και Στόχοι Εργασίας.....	9
1.4. Δομή Εργασίας.....	10
2. BACKGROUND	11
2.1. Τεχνολογίες διαδικτύου.....	11
2.1.1. <i>HTML (HyperText Markup Language)</i>	11
2.1.2. <i>PHP (Hypertext Preprocessor)</i>	14
2.1.3. <i>Perl</i>	17
2.1.4. <i>Python</i>	18
2.1.5. <i>JavaScript</i>	20
2.1.6. <i>JSP (JavaServer Pages)</i>	22
2.1.7. <i>Java Servlet</i>	25
2.2. Τεχνολογίες Βάσεων Δεδομένων.....	28
2.2.1. <i>MySQL</i>	28
2.2.2. <i>Oracle Database</i>	29
2.2.3. <i>PostgreSQL</i>	30
2.3. Επιλογή τεχνολογίας.....	33
3. ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ	34
3.1. Εισαγωγικά.....	34
3.1.1. <i>Βασικές Έννοιες</i>	34
3.1.2. <i>Ιστορική Εξέλιξη των Βάσεων Δεδομένων:</i>	35
3.1.3. <i>Αρχιτεκτονική τριών επιπέδων:</i>	35
3.1.4. <i>Μορφές Βάσεων Δεδομένων:</i>	36
3.1.5. <i>Στοιχεία Βάσεων Δεδομένων:</i>	38
3.1.6. <i>Συνηθισμένα προβλήματα ή δυσκολίες που συναντώνται στις ΒΔ:</i>	39
3.2. Ανάλυση των Απαιτήσεων.....	40
3.2.1. <i>Περιγραφή Συστήματος</i>	40
3.2.2. <i>Λειτουργικές απαιτήσεις (operational / functional specifications).</i>	41
3.2.3. <i>Απαιτήσεις λογισμικού.</i>	42
3.3. Entity – Relationship Model.....	42
3.3.1. <i>Ορισμός E-R model:</i>	42
3.3.2. <i>Βασικά Δομικά Στοιχεία του E – R μοντέλου:</i>	42
3.3.3. <i>Από e-r μοντέλο σε πίνακα (κανονικοποίηση)</i>	45
3.4. SQL Κώδικας.....	47
3.4.1. <i>Εξαγωγή SQL κώδικα από MySQL Workbench</i>	47
4. ΛΕΙΤΟΥΡΓΙΕΣ ΣΕ PHP	53
4.1. Λειτουργία login (είσοδος στην ιστοσελίδα).....	53
4.2. Λειτουργία connect (σύνδεση στη Β.Δ.).....	55
4.3. Λειτουργία Backup (δημιουργία αντιγράφου ασφαλείας).....	56
4.4. Λειτουργία restore (Επαναφορά αντιγράφου ασφαλείας).....	57
4.5. Αναζήτηση Στοιχείου.....	58
4.6. Εισαγωγή στοιχείου.....	59
4.7. Διαγραφή Στοιχείου.....	62
4.8. Επεξεργασία Στοιχείου.....	64
4.9. <i>Αριθμοί Τηλεφώνων.</i>	67
4.10. <i>Εισαγωγή στοιχείου από συσχέτιση μεταξύ πινάκων.</i>	69

4.11. Ειδικές Λειτουργίες.....	72
4.11.1. Σενάριο: Επαρκές Απόθεμα Αποθήκης για μεταφορά Προϊόντων σε Υποκατάστημα.....	73
4.11.2. Σενάριο: Μη επαρκές Απόθεμα Αποθήκης για μεταφορά προϊόντος σε Υποκατάστημα.....	76
4.11.3. Σενάριο: Απόθεμα Προϊόντος σε Αποθήκη κάτω του ελάχιστου ορίου.	79
4.11.4. Αποθέματα Υποκαταστήματος και Αποθήκης κάτω του ελάχιστου.....	82
5. ΕΠΙΛΟΓΟΣ	85
5.1. Αποτελέσματα	85
5.2. Συμπεράσματα.....	86
5.3. Βιβλιογραφία & χρήσιμες ιστοσελίδες	86
ΠΑΡΑΡΤΗΜΑ Α.....	87
Κώδικας για την τεχνολογία Ajax.....	87
Κώδικας στην σελίδα εμφάνισης «Προϊόν»	88
Κώδικας για αναζήτηση προϊόντος στην σελίδα που εκτελείται στο background	93
Εισαγωγή αποθέματος σε Υποκατάστημα	94
Εισαγωγή αποθέματος σε Αποθήκη.....	96
Αναζήτηση σε απόθεμα αποθήκης.....	99
Αναζήτηση σε απόθεμα Υποκαταστήματος.....	107
ΠΑΡΑΡΤΗΜΑ Β	115
Εργαλεία που χρησιμοποιήθηκαν	115
<i>Mysql Server 5.5</i>	115
<i>WampServer v.2</i>	122
<i>MySQL Workbench</i>	127
<i>Favicon</i>	129
Τεχνικές διαδικτύου που χρησιμοποιήθηκαν	130
<i>CSS (Cascading Style Sheets)</i>	130
<i>Ajax (asynchronous JavaScript and XML)</i>	130
Extra εργαλεία.....	131
<i>Adobe Dreamweaver</i>	131
<i>Xampp</i>	132
ΠΑΡΟΥΣΙΑΣΗ.....	133

Πίνακας Εικόνων

<i>Εικόνα 1</i>	35
<i>Εικόνα 2</i>	36
<i>Εικόνα 3</i>	37
<i>Εικόνα 4</i>	37
<i>Εικόνα 5</i>	38
<i>Εικόνα 6</i>	41
<i>Εικόνα 7</i>	43
<i>Εικόνα 8</i>	44
<i>Εικόνα 9</i>	44
<i>Εικόνα 10</i>	45
<i>Εικόνα 11</i>	48
<i>Εικόνα 12</i>	53
<i>Εικόνα 13</i>	54
<i>Εικόνα 14</i>	55
<i>Εικόνα 15</i>	56
<i>Εικόνα 16</i>	57
<i>Εικόνα 17</i>	58
<i>Εικόνα 18</i>	59
<i>Εικόνα 19</i>	61
<i>Εικόνα 20</i>	61
<i>Εικόνα 21</i>	62
<i>Εικόνα 22</i>	62
<i>Εικόνα 23</i>	64
<i>Εικόνα 24</i>	64
<i>Εικόνα 25</i>	65
<i>Εικόνα 26</i>	66
<i>Εικόνα 27</i>	67
<i>Εικόνα 28</i>	67
<i>Εικόνα 29</i>	69
<i>Εικόνα 30</i>	70
<i>Εικόνα 31</i>	70
<i>Εικόνα 32</i>	71
<i>Εικόνα 33</i>	71
<i>Εικόνα 34</i>	72
<i>Εικόνα 35</i>	72
<i>Εικόνα 36</i>	73
<i>Εικόνα 37</i>	74
<i>Εικόνα 38</i>	74
<i>Εικόνα 39</i>	75
<i>Εικόνα 40</i>	75
<i>Εικόνα 41</i>	76
<i>Εικόνα 42</i>	77
<i>Εικόνα 43</i>	78
<i>Εικόνα 44</i>	78
<i>Εικόνα 45</i>	79
<i>Εικόνα 46</i>	80
<i>Εικόνα 47</i>	80
<i>Εικόνα 48</i>	81

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

<i>Εικόνα 49</i>	81
<i>Εικόνα 50</i>	82
<i>Εικόνα 51</i>	83
<i>Εικόνα 52</i>	83
<i>Εικόνα 53</i>	84
<i>Εικόνα 54</i>	84

1. ΕΙΣΑΓΩΓΗ:

1.1. Περίληψη

Η πτυχιακή μας αποσκοπεί στην μελέτη, σχεδίαση και ανάπτυξη μιας ιστοσελίδας για την ευκολότερη και γρηγορότερη διαχείριση των προϊόντων μιας αλυσίδας super-market. Επίσης μελετήθηκαν υπάρχουσες τεχνολογίες ανάπτυξης ιστοσελίδων για την επίτευξη του στόχου αυτού. Πιο συγκεκριμένα, χρησιμοποιήσαμε την HTML, για την βασική ανάπτυξη της σελίδας, την PHP, για την σύνδεση στην βάση δεδομένων μας όπως επίσης και την σύνδεση των διαφορετικών σελίδων μεταξύ τους και την JavaScript για διάφορες επιπρόσθετες λειτουργίες. Επίσης, χρησιμοποιήθηκε το πρόγραμμα διαχείρισης βάσης δεδομένων (MySQL), με το οποίο καταφέραμε να διαχειριστούμε τα δεδομένα μας με ευκολία.

Έτσι, το σύστημα που αναπτύχθηκε, δίνει τη δυνατότητα στον χρήστη του να παραγγέλνει προϊόντα, να ενημερώνετε για το απόθεμα του καταστήματος όπως επίσης και της αποθήκης, να τα διαγράφει και να τα τροποποιεί και άλλα, να κάνει αναζήτηση ακόμα και με την χρήση φίλτρων κ.α.

1.2. Κίνητρο για την Διεξαγωγή της Εργασίας

Ο βασικότερος λόγος που επιλέξαμε αυτή τη πτυχιακή είναι το προσωπικό ενδιαφέρον που είχαμε για το συγκεκριμένο θέμα και η απόκτηση εμπειρίας στον τομέα αυτόν. Επίσης, γνωρίζοντας ότι στην αγορά εργασίας υπάρχει ακόμα ζήτηση για εξειδίκευση σε διαχείριση βάσεων δεδομένων καθώς και ότι πλέον είναι απαραίτητη η ικανότητα ανάπτυξης ιστοσελίδων, θα μας βοηθήσει για την μετέπειτα επαγγελματική μας αποκατάσταση.

Η ανάγκη για την απλοποίηση της διαχείρισης μας γεννήθηκε από τον μεγάλο αριθμό διαφορετικών προϊόντων που υπάρχουν σε μια αλυσίδα super-market ενώ η ανάγκη για την απλοποίηση της παραγγελίας τους ξεκίνησε λόγω της δυσκολίας διεκπεραίωσης παραγγελιών από πολλούς και διαφορετικούς προμηθευτές.

1.3. Σκοπός και Στόχοι Εργασίας

Σκοπός της πτυχιακής εργασίας μας είναι η εξοικείωση με τις βάσεις δεδομένων και ανάπτυξη ιστοσελίδων. Επίσης, η επέκταση και εμπάθυνση προηγούμενων γνώσεων σε γλώσσες σχεδίασης και προγραμματισμού (π.χ. html, php, javascript, jsp κλπ), γλώσσες ανάπτυξης βάσεων δεδομένων (π.χ. MySQL), σύγχρονες τεχνολογίες διαδικτύου (π.χ. AJAX) και βοηθητικά εργαλεία (π.χ. MySQL Server, phpMyAdmin, notepad++, Command Prompt-DOS των Windows).

Σημαντικοί στόχοι για την εκπόνηση της εργασίας:

- Σχεδίαση του σχεσιακού μοντέλου οντοτήτων (E-R Model)
- Υλοποίηση μοντέλου στην MySQL
- Σύνδεση βάσης με την ιστοσελίδα μας
- Υλοποίηση του τεχνικού μέρους της πτυχιακής εργασίας
- Έλεγχος λειτουργίας και διόρθωση σφαλμάτων
- Συγγραφή αναφοράς εργασίας
- Υποβολή αίτησης αξιολόγησης εργασίας
- Προετοιμασία παρουσίασης αναφοράς
- Παρουσίαση αναφοράς

1.4. Δομή Εργασίας

Η πτυχιακή μας εργασία χωρίζεται σε 4 κεφάλαια μετά την εισαγωγή. Το δεύτερο κεφάλαιο επικεντρώνετε στην θεωρητική ανάλυση των σύγχρονων τεχνολογιών διαδικτύου που σχετίζεται με δυναμικές ιστοσελίδες και τους λόγους για τους οποίους επιλέχτηκαν οι τεχνολογίες αυτές. Στο επόμενο κεφάλαιο αναλύσαμε τον σχεδιασμό και την υλοποίηση μιας σχεσιακής βάσης δεδομένων με τη χρήση των εργαλείων MySQL Server και MySQL Workbench. Στο προτελευταίο κεφάλαιο αναπτύξαμε τις λειτουργίες της δυναμικής ιστοσελίδας μας σε PHP (select, update, delete κλπ). Στο τελευταίο κεφάλαιο βρίσκουμε τον επίλογο της πτυχιακής μας εργασίας αναπαράγοντας εν συντομία τα παραπάνω.

2. BACKGROUND

2.1. Τεχνολογίες διαδικτύου

2.1.1. HTML (HyperText Markup Language)

Τα αρχικά HTML προέρχονται από τις λέξεις HyperText Markup Language. Η html δεν είναι μια γλώσσα προγραμματισμού. Είναι μια γλώσσα σήμανσης (markup language), δηλαδή ένας ειδικός τρόπος γραφής κειμένου. Ο καθένας μπορεί να δημιουργήσει ένα αρχείο HTML χρησιμοποιώντας απλώς έναν επεξεργαστή κειμένου. Αποτελεί υποσύνολο της γλώσσας SGML (Standard Generalized Markup Language) που επινοήθηκε από την IBM προκειμένου να λυθεί το πρόβλημα της μη τυποποιημένης εμφάνισης κειμένων στα διάφορα υπολογιστικά συστήματα. Ο browser αναγνωρίζει αυτόν τον τρόπο γραφής και εκτελεί τις εντολές που περιέχονται σε αυτόν.

Αξίζει να σημειωθεί ότι η html είναι η πρώτη και πιο διαδεδομένη γλώσσα περιγραφής της δομής μιας ιστοσελίδας. Η html χρησιμοποιεί τις ειδικές ετικέτες (τα tags) να δώσει τις απαραίτητες οδηγίες στον browser. Τα tags είναι εντολές που συνήθως ορίζουν την αρχή ή το τέλος μιας λειτουργίας. Τα tags βρίσκονται πάντα μεταξύ των συμβόλων < και >. Π.χ. <BODY> Οι οδηγίες είναι case insensitive, δεν επηρεάζονται από το αν έχουν γραφτεί με πεζά (μικρά) ή κεφαλαία. Ένα αρχείο HTML πρέπει να έχει κατάληξη htm ή html.

Κανόνες

Για να μπορούν οι browser να ερμηνεύουν σχεδόν απόλυτα σωστά την html έχουν θεσπιστεί κάποιοι κανόνες. Αυτοί οι κανόνες είναι γνωστοί ως προδιαγραφές. Επομένως σχεδόν κάθε είδος υπολογιστή μπορεί να δείξει το ίδιο καλά μια ιστοσελίδα. Οι πρώτες προδιαγραφές ήταν η html 2.0. Πρόβλημα προέκυψε όταν η Microsoft και η Netscape πρόσθεσαν στην html τέτοιες δυνατότητες που στην αρχή τουλάχιστον ήταν συμβατές μόνο με συγκεκριμένους browser. Ακόμη και σήμερα υπάρχουν διαφορές στην απεικόνιση κάποιας σελίδας από διαφορετικούς browsers. Ιδιαίτερο είναι το πρόβλημα όταν η ιστοσελίδα, εκτός από "καθαρή" HTML περιλαμβάνει και εφαρμογές Javascript.

Η HTML σήμερα

Σήμερα πολλοί είναι εκείνοι που δημιουργούν μια ιστοσελίδα σε κάποιο πρόγραμμα που επιτρέπει την δημιουργία χωρίς την συγγραφή κώδικα. Η κοινή άποψη πάνω στο θέμα όμως είναι ότι κάτι τέτοιο είναι αρνητικό επειδή ο δημιουργός δεν έχει τον απόλυτο έλεγχο του κώδικα με αποτέλεσμα πολλές φορές να υπάρχει οπτικό χάος στην προσπάθεια των browser να εμφανίσουν την ιστοσελίδα. Για το σκοπό αυτό έχει δημιουργηθεί ειδικό λογισμικό, που επιτρέπει το "στήσιμο" της σελίδας οπτικά, χωρίς τη συγγραφή κώδικα, δίνει όμως τη δυνατότητα παρέμβασης ΚΑΙ στον κώδικα. Χαρακτηριστικό παράδειγμα το λογισμικό Dreamweaver της Adobe και το FrontPage της Microsoft.

Εμφάνιση των αρχείων HTML

Οι σελίδες που γράφονται σε HTML είναι απλά αρχεία κειμένου και δε περιέχουν πληροφορίες συγκεκριμένες για ένα λειτουργικό σύστημα ή ένα πρόγραμμα, μπορούν να διαβάζονται από οποιοδήποτε συντάκτη υποστηρίζει απλό κείμενο. Τα tags της HTML είναι τα πράγματα που φαίνονται μέσα στα σύμβολα <>. Τα tags υποδεικνύουν χαρακτηριστικά ή στοιχεία μιας σελίδας. Τα περισσότερα tags της HTML δείχνουν περίπου έτσι:
<όνομα tag> επηρεαζόμενο κείμενο </όνομα tag>.

Το ίδιο όνομα του tag περικλείεται μέσα σε σύμβολα "μεγαλύτερο από" και "μικρότερο από" (<>). Τα tags της HTML έχουν γενικά ένα tag αρχής και ένα tag τέλους. Τα δυο αυτά συντροφικά tags περικλείουν το κείμενο που επηρεάζουν. Το tag αρχής ενεργοποιεί μια λειτουργία ή ένα χαρακτηριστικό (όπως ο χαρακτηρισμός μιας επικεφαλίδας, η έντονη γραφή), ενώ το tag τέλους την απενεργοποιεί. Τα tags τέλους έχουν το ίδιο όνομα με τα tags αρχής, με το πρόθεμα τον χαρακτήρα "/". Δεν έχουν μορφή ζεύγους όλα τα tags της HTML. Ορισμένα tags είναι "μονομελή", ενώ άλλα

περιέχουν επιπλέον πληροφορίες και κείμενο μέσα στα σύμβολα "<>". Όλα τα tags της HTML δεν κάνουν διάκριση μεταξύ κεφαλαίων και πεζών.

Σύνταξη και εντολές

Είναι μια περιγραφική γλώσσα μορφοποίησης δεδομένων και κλήσης άλλων αρχείων. Τα βασικά δομικά στοιχεία της html είναι οι ετικέτες(tags) που αναγνωρίζονται και μεταφράζονται σε εντολές από τους browsers και έτσι παρουσιάζεται η ιστοσελίδα.

Οι ετικέτες περικλείονται από τα σύμβολα <> και ομαδοποιούνται σε ζεύγη για άνοιγμα και κλείσιμο π.χ. <h3>...</h3>

Κάθε σελίδα HTML περικλείεται από τα tags <html>...</html> και χωρίζεται σε δύο τμήματα το <head>...</head> και στο <body>...</body>

Στο **τμήμα <head>** περιέχονται :

μεταπληροφορίες (metatags) για την ιστοσελίδα οι οποίες μπορούν να αξιοποιηθούν από τις μηχανές αναζήτησης βελτιώνοντας σημαντικά την κατάταξη της.

Ο τίτλος της ιστοσελίδας ανάμεσα στα tags <title> ...</title> ο οποίος εμφανίζεται στο πάνω μέρος του browser.

Ορίζεται η χρήση του CSS και JavaScript.

Στην **περιοχή body** ορίζεται το κύριο μέρος της ιστοσελίδας και μπορεί να περιέχει Ετικέτες μορφοποίησης

<p> ορίζει έναρξη παραγράφου.

<h1><h2>...<h6> Το κείμενο ανάμεσα σε αυτές τις ετικέτες έχει την μορφή επικεφαλίδας της οποίας το μέγεθος μειώνεται όσο μεταβαίνουμε από την <h1> στην <h6>.

<hr> εμφανίζει μια οριζόντια γραμμή στην σελίδα μας.

<i> εμφανίζει το κείμενο ανάμεσα στο εν λόγω tag με πλάγιους χαρακτήρες.

 εμφανίζει το κείμενο με έντονη μορφή.

<u> εμφανίζει το κείμενο υπογραμμισμένο.

 αλλαγή γραμμής.

Λίστες Ταξινομημένες ή αταξινομητες με τα tags και αντίστοιχα π.χ.

 Red Green Blue

Πλαίσια (frames) ορίζονται με το tag <frame>

Εικόνες φορτώνονται με το tag img όπου ορίζουμε το όνομα και την διαδρομή του αρχείου εικόνας(src), τις διαστάσεις της(with-height) και την επεξήγηση(title) π.χ.

Υπερσυνδέσμους σε άλλες σελίδες με το tag <A>... όπου ορίζουμε το όνομα και την διαδρομή της σελίδας(HREF), την επεξήγηση(title) π.χ.

HERE

Πίνακες δεδομένων ορίζονται με τα tags <table>...</table> Το <table> έχει διάφορες ιδιότητες όπως width border, caption π.χ.

<table width="100%" border="1" <caption> Πελάτες</caption>>

Οι γραμμές του πίνακα ορίζονται με το tag <tr>...</tr> ενώ οι στήλες με το tag <th>...</th>

για επικεφαλίδες στηλών και <td>...</td> για δεδομένα στηλών. Π.χ

<table><tr><th>Name</th><th>address</th><th>phone</th></tr>

<tr><th rowspan="2">J.papas</th><th>Herakleion</th></tr>

<tr><th>123-444</th></tr></table>

Η rowspan ορίζει σε πόσες γραμμές απλώνεται ένα κελί παρόμοια είναι και το colspan

Φόρμες εισαγωγής δεδομένων από τον client και αποστολής τους στον server. Ορίζονται με τα tags <form>...</form> Η <form> έχει 2 ιδιότητες την **Action** που ορίζει σε ποια σελίδα θα στείλουμε τα δεδομένα και την **Method** που ορίζει τη μέθοδο του πρωτοκόλλου HTTP θα

χρησιμοποιήσουμε για την αποστολή των δεδομένων **GET** ή **POST** π.χ. <form action="page1.php" method="get">

Με την μέθοδο POST τα δεδομένα στέλνονται μέσα στα HTTP μηνύματα που στέλνει ο browser και δεν τα βλέπει ο χρήστης ενώ με την μέθοδο GET τα δεδομένα προστίθενται στο url π.χ.

<http://www.teicrete.gr/page1.php?a1=12>

Μία φόρμα μπορεί να περιέχει τα παρακάτω στοιχεία:

Στοιχείο **Text** για την εισαγωγή κειμένου π.χ.

```
<input type="text" id="c1" name="code" value="0"> Πριν από το στοιχείο text μπορούμε να εμφανίσουμε μία ετικέτα για να γνωρίζει ο χρήστης τι είναι αυτό που εισάγει π.χ. <label for="c1">κωδικός:</label>
```

Στοιχείο **Textarea** για την εισαγωγή κειμένου περισσότερων από μία γραμμή π.χ.

```
<textarea name="area1">περιγραφή</textarea>
```

Στοιχείο για την εισαγωγή συνθηματικού(**password**) π.χ.

```
<input type="password" name="myPasswd">
```

Στοιχείο **checkbox** για να τσεκάρει ο χρήστης μία ή περισσότερες επιλογές π.χ.

```
<input type="checkbox" name="lang1" value="English"> English <br>
```

```
<input type="checkbox" name="lang2" value="German"> German <br>
```

```
<input type="checkbox" name="lang3" value="France"> France <br>
```

Στοιχείο **radio button** για να τσεκάρει ο χρήστης μία επιλογή π.χ.

```
<input type="radio" name="thesex" value="male"> male <br>
```

```
<input type="radio" name="thesex" value="female">female <br>
```

Στοιχείο **select** ορίζει μία pop up list για επιλέξει ο χρήστης μία επιλογή π.χ.

```
<select name="combo">
```

```
<option>Ag.Nikolaos</option>
```

```
<option>heraklion</option>
```

```
<option>rethymno</option>
```

```
<option>hania</option>
```

```
</select>
```

Στοιχείο **button** ορίζουμε κουμπιά διαφόρων τύπων όπως submit για αποστολή της φόρμας στον server , reset για επαναφορά της φόρμας π.χ.

```
<input type="submit" name="button1" value="Αποστολή">
```

```
<input type="reset" name="reset" value="reset">
```

Πλεονεκτήματα

- Εύκολο στη χρήση.
- Χαλαρή σύνταξη (αν και, επειδή είναι πάρα πολύ ευέλικτο δεν θα συμφωνεί με τα πρότυπα)
- Υποστηρίζεται σχεδόν σε κάθε πρόγραμμα περιήγησης, αν όχι όλους.
- Χρησιμοποιείται ευρέως. Ιδρύθηκε σχεδόν σε κάθε ιστοσελίδα, αν όχι όλες.
- Πολύ παρόμοια με την σύνταξη του XML, το οποίο χρησιμοποιείται όλο και περισσότερο για την αποθήκευση δεδομένων.
- Δωρεάν - Δεν χρειάζεται να αγοράσετε κανένα λογισμικό.
- Εύκολο στη μάθηση και δημιουργία κώδικα ακόμα και για αρχάριους προγραμματιστές.

Μειονεκτήματα

- Δεν μπορεί να παράγει δυναμική έξοδο από μόνη της, δεδομένου ότι είναι μια στατική γλώσσα.
- Μερικές φορές, η δομή των HTML εγγράφων είναι δύσκολο να κατανοηθούν.
- Θα πρέπει να ενημερώνεται για ξεπερασμένες ετικέτες, και να φροντίσει να μην τις χρησιμοποιεί.
- Εμφανίζονται απαξιωμένες ετικέτες επειδή μια άλλη γλώσσα που λειτουργεί με την HTML έχει αντικαταστήσει το πρωτότυπο έργο της ετικέτας. Έτσι η άλλη γλώσσα πρέπει να μαθευτεί (τις περισσότερες φορές, είναι η CSS).
- Οι λειτουργίες ασφαλείας που προσφέρει η HTML είναι περιορισμένες.

2.1.2. PHP (Hypertext Preprocessor)

Είναι γλώσσα προγραμματισμού για την δημιουργία δυναμικών ιστοσελίδων. Είναι **Server-side scripting** δηλαδή το πρόγραμμα εκτελείται στον web-server. Η εκτέλεση του προγράμματος στον web server γίνεται με την βοήθεια του **μεταγλωττιστή PHP(parser)** που πρέπει να έχει εγκατασταθεί στον web server. Η PHP είναι HTML ενσωματωμένη scripting γλώσσα. Δηλαδή κώδικας PHP συμπεριλαμβάνεται μέσα σε μία σελίδα HTML. Κατά την μεταγλώττιση η PHP αφήνει την σελίδα HTML όπως είναι μέχρι να συναντήσει ένα από τα ειδικά tags έναρξης php “<?” και τότε αρχίζει να μεταγλωττίζει το κείμενο ως κώδικα PHP μέχρι να συναντήσει το επόμενο PHP tag κλεισίματος “>” δηλαδή οτιδήποτε βρίσκεται έξω από τα tags της PHP μένει όπως είναι. Το αποτέλεσμα της μεταγλώττισης μορφοποιείται σε HTML και όλη ιστοσελίδα αποστέλλεται στον browser.

Από τα παραπάνω συμπεραίνεται ότι η PHP είναι **διερμηνέας(interpreter)** δηλαδή δεν παράγει εκτελέσιμο αρχείο. Είναι **command line scripting** δηλαδή ένα Php script μπορεί να τρέξει χωρίς Browser και Server. Χρειαζόμαστε μόνο τον μεταγλωττιστή. Υποστηρίζει πολλά πρωτόκολλα για επικοινωνία με άλλες υπηρεσίες όπως IMAP,SNMP,NNTP,HTTP,POP3 καθώς επίσης και τα socket. Υποστηρίζεται από τους περισσότερους web servers.

Επεκτάσεις αρχείων και διακομιστές

Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη επέκταση (π.χ. *.php, *.php4, *.phtml κ.ά.). Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης .html δεν θα λειτουργήσει και θα εμφανίσει στον browser τον κώδικα χωρίς καμία επεξεργασία, εκτός αν έχει γίνει η κατάλληλη ρύθμιση στα MIME types του server. Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο server να είναι ρυθμισμένος για να επεξεργάζεται κώδικα PHP. Ο διακομιστής Apache, που χρησιμοποιείται σήμερα ευρέως σε συστήματα με τα λειτουργικά συστήματα GNU/Linux και Microsoft Windows, υποστηρίζει εξ ορισμού την εκτέλεση κώδικα PHP.

Ιστορία της PHP

Η ιστορία της PHP ξεκινά από το 1994, όταν ένας φοιτητής, ο Rasmus Lerdorf δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα απλό script με όνομα php.cgi, για προσωπική χρήση. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασισμένη αυτή τη φορά στη γλώσσα C και αριθμώντας περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. Έτσι η PHP έφθασε στην έκδοση 3.0 η οποία θύμιζε περισσότερο τη σημερινή μορφή της.

Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το 1998 η έκδοση 4 της PHP, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5, ενώ αυτή τη στιγμή έχουν ήδη διατεθεί και οι πρώτες δοκιμαστικές εκδόσεις της επερχόμενης PHP 6, για οποιονδήποτε προγραμματιστή θέλει να τη χρησιμοποιήσει. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP.

Σύνταξη και Εντολές της PHP

Ο κώδικας αρχίζει με <? και τελειώνει με ?>.

Κάθε εντολή τελειώνει με Ελληνικό ερωτηματικό (;)

Τα σχόλια αρχίζουν με //

Οι μεταβλητές ορίζονται με την απόδοση τιμής. Το όνομα τους Ξεκινά με \$ και ακολουθεί γράμμα ή κάτω παύλα και στην συνέχεια οτιδήποτε εκτός από χαρακτήρες +,-,&,* . Είναι **case-sensitive** π.χ. \$total και \$Total είναι διαφορετικές μεταβλητές.

Υπάρχουν οι γνωστοί τύποι μεταβλητών όπως συμβολοσειρές (strings), ακέραιοι (integers), αριθμοί κινητής υποδιαστολής (floats), λογικές τιμές (booleans), πίνακες (arrays) και αντικείμενα (objects) .

Δομές ελέγχου

If (συνθήκη){εντολές}else{εντολές}
Switch(μεταβλητή) case τιμή1: εντολές break;case τιμή2: εντολές; break; ... default: εντολές;

Δομές επανάληψης

while(συνθήκη){εντολές}
Do{ εντολές}while(συνθήκη)
for(Αρχική τιμή; Τελική _____ τιμή

Ορισμοί Πινάκων

\$Pin1= new array(); //ορίζει ένα κενό πίνακα
\$colors = array('red', 'blue', 'green', 'yellow');
\$Pin2[]= "ΤΕΓ"; \$Pin2[]= "Ηράκλειο"; //Με αυτό τον τρόπο ο τελεστής [] ορίζει τον πίνακα \$Pin2 και του δίνει και περιεχόμενο στις θέσεις \$Pin2[0] και \$Pin2[1] τις τιμές "ΤΕΓ" και "Ηράκλειο" αντίστοιχα.

Η εμφάνιση των χρωμάτων που περιέχονται στον πίνακα \$colors που ορίσαμε παραπάνω γίνεται με τον βρόγχο που ακολουθεί

```
foreach ($colors as $color) { echo " $color\n";}
```

Πρόσβαση σε MYSQL βάση

Για να συνδεθούμε με την βάση χρησιμοποιούμε την συνάρτηση

mysql_connect(hostname , username ,password)

Hostname είναι η διεύθυνση του υπολογιστή που τρέχει ο mysql server

Η παραπάνω συνάρτηση σε περίπτωση επιτυχίας επιστρέφει έναν αναγνωριστή συνδέσμου τον οποίο πρέπει να αποθηκεύουμε σε μία μεταβλητή για να τον χρησιμοποιήσουμε σε επόμενες εντολές π.χ. \$conn=mysql_connect(hostname , username ,password)

Σε περίπτωση αποτυχίας η συνάρτηση mysql_connect επιστρέφει false.

mysql_select_db(όνομα βάσης, αναγνωριστής συνδέσμου) Χρησιμοποιείται για την επιλογή βάσης δεδομένων π.χ. mysql_select_db("test1", \$conn)

mysql_create_db(όνομα βάσης, αναγνωριστής συνδέσμου) Χρησιμοποιείται για την δημιουργία βάσης δεδομένων π.χ. mysql_create_db("dimopras", \$conn)

mysql_drop_db(όνομα βάσης, αναγνωριστής συνδέσμου) Χρησιμοποιείται για την διαγραφή βάσης δεδομένων π.χ. mysql_drop_db("dimopras", \$conn)

mysql_query(ερώτημα, αναγνωριστής συνδέσμου) Χρησιμοποιείται για την εκτέλεση ερωτημάτων.

Το αποτέλεσμα του ερωτήματος το αποθηκεύουμε σε μεταβλητή ειδικά αν έχουμε ερώτημα ανάκλησης δεδομένων από πίνακα π.χ.

```
$query1="select * from users";
```

```
$result=mysql_query($query1,$conn);
```

Για να επεξεργαστούμε το περιεχόμενο του αποτελέσματος σε γραμμές χρησιμοποιούμε τη συνάρτηση **mysql_fetch_array()** η οποία κάθε φορά επιστρέφει μία γραμμή από το αποτέλεσμα για παράδειγμα έστω ότι θέλω να πάρω τα στοιχεία των χρηστών που είναι καταχωρημένοι στον πίνακα users. Τα στοιχεία των χρηστών είναι ο κωδικός το όνομα και το τηλέφωνο δηλαδή τα πεδία code, fullname και phone αντίστοιχα.

```
$sql="select code,fullname,phone from users";
```

```
$result=mysql_query($sql,$conn);
```

```
while($row = mysql_fetch_array($result))
```

```
{ $code=$row['code'];
```

```
$fullname=$row['fullname'];
```

```
$phone=$row['phone'];
```

```
}
```

Αν κάτι δεν πάει καλά κατά την εκτέλεση των εντολών μπορούμε να χρησιμοποιήσουμε την συνάρτηση **die()** για να εμφανίσουμε κάποιο μήνυμα δικό μας ή το αντίστοιχο μήνυμα της mysql με την χρήση της συνάρτησης **mysql_error()** ή και τα δύο όπως στο παράδειγμα που ακολουθεί:

```
$conn=mysql_connect("127.0.0.1","user1","niki") or die("μή εφικτή η σύνδεση" .mysql_error ());
```

Πλεονεκτήματα:

- Είναι εύκολα προσβάσιμη.
- Είναι διαθέσιμη δωρεάν.
- Είναι διαθέσιμη με βοηθητικά έγγραφα σε πολλές γλώσσες.
- Υπάρχουν πολλά γκρουπ, forums και ομάδες υποστήριξης.
- Υπάρχει πλούτος πληροφοριών στο διαδίκτυο.
- Είναι γρήγορη η ανάπτυξη κώδικα.
- Ένα βασικό script στην PHP μπορεί να δημιουργηθεί χωρίς την απαραίτητη κατανόηση των εντολέων του προγραμματισμού, την κατάρτιση, καθώς και άλλες σημαντικές έννοιες του προγραμματισμού.
- Η PHP είναι χαλαρά δακτυλογραφημένη, γεγονός που καθιστά τα βασικά scripts πολύ πιο γρήγορα στην ανάπτυξη και με λιγότερη έμφαση στο σχεδιασμό.
- Προγραμματιστές της Java, Perl, BASIC, και άλλες δημοφιλείς γλώσσες μπορούν να βρουν πολλές ομοιότητες για την διευκόλυνση μετάβασης στην PHP.
- Η PHP είναι ευέλικτη. Χρησιμοποιεί OOP ή όχι. Χρησιμοποιεί την μετατροπή-ες ονομάτων ή όχι.
- Τρέχει σε πολλά διαφορετικά λειτουργικά συστήματα.
- Μπορεί να βελτιστοποιηθεί, ακόμη και αν είναι "καταρτισμένη" για τις επιδόσεις πιο κοντά σε εκείνη των περισσότερων γλωσσών.
- Είναι ανοιχτού κώδικα, άμεσα διαθέσιμο (μπορεί ήδη να το χρησιμοποιείτε σήμερα) και διπλής άδειας. Αν κάνετε μη κερδοσκοπική εργασία ή χωρίς τη χορήγηση αδειών, δεν υπάρχει κόστος.
- Πολύ εύκολο να καταλάβει κανείς τη σύνταξη, μερικά πραγματικά εντυπωσιακά χαρακτηριστικά γνωρίσματα (δείκτες και άλλα...)
- Έχει πολύ εύκολα διεπαφή με Apache / MySQL.
- Λειτουργεί στον Server (δεν χρειάζεται να μπλέκει με εγκατάσταση στον client - μόνο με θέματα rendering, εάν χρησιμοποιείτε CSS).
- Υπάρχουν άφθονες πηγές καλού πηγαίου κώδικα εκεί έξω για να χρησιμοποιηθεί και/ή να μάθουν από αυτόν, καθώς και όπως πολλές χρήσιμες βιβλιοθήκες για την εργασία με αρχεία PDF, γραφικά, κλπ.
- Πολλά καλά βιβλία και βοήθεια στο διαδίκτυο (php.net κ.α.).
- Πλατφόρμα αγνωστικιστής, μπορεί να τρέξει σε Windows, Linux ή Mac servers. Επίσης εξαιρετικά επεκτάσιμη.
- Πολλοί παροχείς φιλοξενίας ιστοσελίδων το έχουν έτοιμο προς χρήση, καμία ειδική διαμόρφωση (εκτός και αν έχετε ειδικές ανάγκες ασφαλείας).
- Πολύ εύκολη η πρόσβαση σε άλλα web-based εργαλεία μέσω της PHP (π.χ. Google maps, κλπ.)

Μειονεκτήματα:

- Η PHP τείνει να εκτελεί πιο αργά από την assembly, την C, και άλλες γλώσσες.
- Είναι χαλαρά δακτυλογραφημένη. Για τους προγραμματιστές όλων των επιπέδων, αυτό αφήνει περιθώρια για την απροσδόκητη συμπεριφορά, λόγω λάθους του προγραμματιστή που πολλές άλλες γλώσσες δεν θα μπορούσε να επιτρέψει (Φυσικά, λίγες, εάν όχι καμία γλώσσα μπορεί να προστατεύσει από σφάλμα του προγραμματιστή!).
- Υπάρχουν πολλοί τρόποι για να κάνει ένα πράγμα, και πολλές περιπτώσεις όπου μια λειτουργία έχει διφορούμενο χειρισμό λόγω υποστήριξης της κληρονομιάς ή την ιστορία ανάπτυξης.
- Αν θέλετε να κάνετε κάτι παραπάνω από απλές HTML / CSS σελίδες για τον πελάτη σας θα πρέπει να προσθέσετε επίσης JavaScript, Java ή άλλη client-side γλώσσα (ισχύει και για Perl και σε μερικές άλλες γλώσσες).
- Ο τρόπος με τον οποίο λειτουργούν τα προγράμματα περιήγησης κάνουν την διαχείριση των δεδομένων και την κωδικοποίηση των προγραμμάτων περισσότερο ενδιαφέροντα (τεχνικά

κάθε σελίδα είναι μια νέα λειτουργία για το σύστημα, έτσι ώστε να πρέπει να διαχειριστείτε τις μεταβλητές σας που μπαίνουν και βγαίνουν έξω και ανάμεσα στις σελίδες. Δεν είναι δύσκολο, απλά διαφορετικό).

- Κάθε προγραμματισμός Web (ανεξάρτητα από τη γλώσσα) είναι ανοικτός σε κενά ασφαλείας λόγω των μη εφαρμοσμένων ή άγνωστων ευπάθειών (απαιτεί λίγο περισσότερο προσοχή).

2.1.3. Perl

Η Perl είναι μία πολύ δημοφιλής αντικειμενοστραφής γλώσσα προγραμματισμού. Συνήθως ένα πρόγραμμα σε Perl εκτελείται χρησιμοποιώντας άμεσα ή έμμεσα το διερμηνέα της γλώσσας. Αυτό που διακρίνει την Perl από πολλές άλλες γλώσσες προγραμματισμού είναι το γεγονός ότι είναι διαθέσιμη για σχεδόν όλα τα λειτουργικά συστήματα. Η γλώσσα σχεδιάστηκε από τον Larry Wall και ο πηγαίος κώδικάς της διατίθεται βάση της αδειας ανοικτού κώδικα GPL. Η πρώτη έκδοση της γλώσσας εμφανίστηκε το 1987 ενώ στις αρχές του 2010 η πιο πρόσφατη έκδοσή της είναι η 5.10.1.

Ιστορία

Η γλώσσα προγραμματισμού Perl σχεδιάστηκε επειδή ο Larry Wall ήθελε μία γλώσσα η οποία να συνδυάζει μόνο τα καλά στοιχεία του sed, της C, της awk και του κελύφους Bourne του λειτουργικού συστήματος Unix. Η πρώτη έκδοση της γλώσσας ανακοινώθηκε στο alt.comp.sources του Usenet στις 18 Δεκεμβρίου του 1987. Μετά την ταχύτερη διάδοση της γλώσσας ακολούθησαν οι εκδόσεις 2 (1988) και 3 (1989). Η τέταρτη έκδοση του 1991 δεν είχε σημαντικές διαφορές από την 3η αλλά δημιουργήθηκε ως έκδοση αναφοράς για το πρώτο βιβλίο με θέμα τη γλώσσα. Η πέμπτη έκδοση ξεκίνησε να γράφεται το 1993 με την σταθερή έκδοση της να εκδίδεται τον Οκτώβριο του 1994. Η έκδοση αυτή έφερε σημαντικές αλλαγές και προσέθεσε σημαντικές δυνατότητες ενώ ταυτόχρονα αύξησε τον αριθμό των υποστηριζόμενων λειτουργικών συστημάτων. Η έκδοση 5 συνεχίζει να αναπτύσσεται ακόμα και βρίσκεται στην έκδοση 5.14.1. Η Perl 6 είναι η επόμενη έκδοση της γλώσσας η οποία θα επιφέρει αρκετές αλλαγές τόσο στο συντακτικό όσο και στις δυνατότητες της γλώσσας.

Χαρακτηριστικά

Ως γλώσσα προγραμματισμού η Perl έχει ένα δυναμικό σύστημα τύπων, δηλαδή μία μεταβλητή αποκτά τύπο μόνο μετά από την ανάθεση μιας τιμής σε αυτή. Η Perl μπορεί να διαχειριστεί αριθμούς (δεκαδικούς και ακραίους) και συμβολοσειρές. Επιπλέον, μπορούμε να έχουμε πίνακες απλούς (arrays) και συσχετιστικούς (associative arrays ή απλά hash tables), οι οποίοι είναι μηχανισμοί οργάνωσης δεδομένων. Μάλιστα με τους συσχετιστικούς πίνακες μπορούμε να δημιουργήσουμε δυναμικές δομές όπως δένδρα, στοίβες, ουρές κ.λπ. Υπάρχει πρόνοια για τον διαχωρισμό μεταβλητών στις οποίες αναθέτουμε απλές τιμές ή σύνθετες τιμές.

Όταν ορίζουμε μία διαδικασία ή μία συνάρτηση στην Perl δεν σημειώνουμε τυπικές παραμέτρους αλλά απλά θεωρούμε πως η διαδικασία ή η συνάρτηση θα έχουν ως όρισμα έναν πίνακα. Άρα για μια συνάρτηση ή μία διαδικασία δεν υπάρχει προκαθορισμένος αριθμός ορισμάτων. Επίσης οι συναρτήσεις και οι διαδικασίες στην Perl μπορούν να οριστούν αναδρομικά.

Ένα άλλο βασικό χαρακτηριστικό της γλώσσας είναι η εξαιρετική υποστήριξη των κανονικών εκφράσεων, δηλαδή παραστάσεων που μπορούν να περιγράψουν την γενική δομή μιας συμβολοσειράς και μπορούν να χρησιμοποιηθούν στη διαχείριση συμβολοσειρών. Η πιο χαρακτηριστική χρήση των κανονικών εκφράσεων είναι στην λεκτική ανάλυση γλωσσών προγραμματισμού, δηλαδή στο κομμάτι εκείνο ενός μεταγλωττιστή που «σπάει» την είσοδο σε λεκτικές μονάδες (tokens).

Όπως αναφέρθηκε η Perl είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού. Στην Perl οι κλάσεις είναι πακέτα με ειδικές συναρτήσεις τα οποία επιστρέφουν την τιμή 1. Ο τρόπος δημιουργίας και χρήσης κλάσεων δεν είναι ιδιαίτερα κομψός ενώ παράλληλα δεν υποστηρίζει πολλές από τις δυνατότητες που υποστηρίζουν κατεξοχήν αντικειμενοστραφείς γλώσσες όπως η Java και η Scala.

Επίσης η Perl παρέχει εργαλεία για ταυτόχρονη επεξεργασία δεδομένων μέσω νημάτων ή κάνοντας χρήση της κλήσης συστήματος fork. Επιπλέον παρέχει υποδομές για δικτυακό

προγραμματισμό. Επίσης παρέχει υποδομές για την εύκολη διασύνδεση με γνωστές βάσεις δεδομένων, ενώ τα τελευταία χρόνια έγινε ιδιαίτερη δημοφιλής λόγω της χρήσης της στη βιοπληροφορική. Τέλος, η Perl έγινε πολύ γνωστή λόγω της χρήσης της για τη δημιουργία των λεγομένων CGI scripts.

Χρήση

Η Perl χρησιμοποιείται από ένα ευρύ φάσμα χρηστών και σε πολλές εφαρμογές. Για παράδειγμα χρησιμοποιείται στη βιοπληροφορική, στη διαχείριση συστημάτων, στη διαχείριση ιστοτόπων κ.λπ.

Λειτουργικά συστήματα

Η Perl είναι προεγκατεστημένη στη συντριπτική πλειοψηφία των λειτουργικών συστημάτων τύπου Unix, ενώ υπάρχουν έτοιμες διανομές για τα περισσότερα λειτουργικά συστήματα. Πιο συγκεκριμένα η Perl μπορεί να χρησιμοποιηθεί στα παρακάτω λειτουργικά συστήματα:

- Linux
- OpenSolaris και Solaris
- Microsoft Windows (όλες οι εκδόσεις συμπεριλαμβανομένης της πλατφόρμας Pocket PC)
- Mac OS και Mac OS X
- FreeBSD, OpenBSD και NetBSD
- QNX

Πλεονεκτήματα:

- Είναι ελεύθερο λογισμικό.
- Τρέχει σε όλες τις πλατφόρμες και μπορεί να μεταφερθεί μεταξύ UNIX και Windows χωρίς προβλήματα.
- Παρέχει τα χαρακτηριστικά γνωρίσματα που απαιτούνται για τα μεγάλα projects, όπως Modularization, αντικειμενοστραφή τεχνικές και αυθαίρετα δεδομένα.

Μειονεκτήματα:

- Συμπεριλαμβάνει πάρα πολλές ειδικές κατασκευές.
- Είναι δύσκολο να δημιουργηθούν data structures.
- Έχει έλλειψη λειτουργικών υπογραφών το οποίο καθιστά δύσκολη τη λίστα των ονομάτων και τους τύπους των παραμέτρων.

Υποδείξεις:

Ένα από τα χαρακτηριστικά της Perl είναι ότι υπάρχουν περισσότεροι από ένας τρόπος για να κάνετε σχεδόν τα πάντα, γεγονός που καθιστά τη γλώσσα κάπως δύσκολο να την μάθουν αν κάποιος έχουν εκπαιδευτεί σε πιο περιοριστικό σύστημα. Μπορείτε να βρείτε βιβλία για τα Perl στο διαδίκτυο, καθώς και άλλες ηλεκτρονικές πηγές, αν έχετε οποιοδήποτε ερωτήσεις. Η Perl πολλές φορές μπορεί να υποφέρει από θέματα μειωμένης απόδοσης επειδή το πρόγραμμα πρέπει να συντάξει κάποιες πηγές για να μπορέσει να τρέξει μια εργασία.

2.1.4. Python



Η Python είναι μια γλώσσα προγραμματισμού η οποία δημιουργήθηκε από τον Ολλανδό Γκούντο βαν Ρόσσουμ (Guido van Rossum) το 1990. Ο κύριος στόχος της είναι η αναγνωσιμότητα

του κώδικά της και η ευκολία χρήσης της. Διακρίνεται λόγω του ότι έχει πολλές βιβλιοθήκες που διευκολύνουν ιδιαίτερα αρκετές συνηθισμένες εργασίες και για την ταχύτητα εκμάθησής της.

Η Python αναπτύσσεται ως ανοιχτό λογισμικό (free software) και η διαχείρισή της γίνεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation. Ο κώδικας διανέμεται με την άδεια Python Software Foundation License η οποία είναι συμβατή με την GPL. Το όνομα της γλώσσας προέρχεται από την ομάδα άγγλων κωμικών Μόντυ Πάιθον.

Ιστορικό

Αρχικά, η Python ήταν γλώσσα σεναρίων που χρησιμοποιούνταν στο λειτουργικό σύστημα Amoeba, ικανή και για κλήσεις συστήματος.

Η πιο πρόσφατη έκδοση της σειράς 2.x της Python είναι η 2.6. Από τις 3 Δεκεμβρίου 2008 έχει κυκλοφορήσει η έκδοση 3.0 (γνωστή και ως py3k ή python 3000). Τρέχουσα έκδοση είναι η 3.1. Πολλά από τα καινούργια χαρακτηριστικά αυτής της έκδοσης έχουν μεταφερθεί στην έκδοση 2.6 που είναι προς τα πίσω συμβατή.

Η python 3 είναι ιστορικά η πρώτη γλώσσα προγραμματισμού που σπάει την προς τα πίσω συμβατότητα με προηγούμενες εκδόσεις ώστε να διορθωθούν κάποια λάθη που υπήρχαν σε προγενέστερες εκδόσεις και να καταστεί ακόμα πιο σαφής ο απλός τρόπος με τον οποίο μπορούν να γίνουν κάποια πράγματα.

Δομή και σύνταξη

Η γλώσσα χρησιμοποιεί μεταγλωττιστή (compiler) για την δημιουργία του εκτελέσιμου κώδικα και σχετίζεται με τις γλώσσες προγραμματισμού Tcl, Perl, Scheme, Java και Ruby, καθώς και με την ABC η οποία υπήρξε η αρχική πηγή έμπνευσης για τη δημιουργία της.

Ένα από τα πιο απλά προγράμματα στην γλώσσα Python είναι η εμφάνιση ενός γραπτού αποτελέσματος (π.χ. Γεια σου, κόσμε!):

```
>>>print("Γεια σου, κόσμε!")
```

Γεια σου, κόσμε!

Ένα ιδιαίτερο χαρακτηριστικό της γλώσσας είναι η χρήση κενών διαστημάτων (whitespace) για τον διαχωρισμό των συντακτικών δομών που προγράμματος, σε αντίθεση με την πρακτική σε άλλες γλώσσες όπου για τον ίδιο σκοπό χρησιμοποιούνται ειδικά σύμβολα (πχ αγκύλες). Αυτό, σε συνδυασμό με το ότι χρησιμοποιεί πλήρεις αγγλικές λέξεις στη θέση συμβόλων, καθιστούν τον κώδικα της Python ευανάγνωστο από όσους έχουν βασική γνώση των αγγλικών.

Για παράδειγμα, ο παρακάτω γεννήτορας (generator) generate_primes παράγει πρώτους αριθμούς:

```
from itertools import count
def generate_primes(stop_at=None):
    primes = []
    for n in count(2):
        if stop_at is not None and n > stop_at:
            return
        composite = False
        for p in primes:
            if not n % p:
                composite = True
                break
        elif p**2 > n:
            break
        if not composite:
            primes.append(n)
            yield n
```

Και για να καλέσουμε τη συνάρτηση, υπολογίζοντας τους πρώτους αριθμούς από το 1 στο 100:

```
for i in generate_primes():
    if i > 100: break
    print(i)
```

Πλεονεκτήματα:

- Επεκτασιμότητα
- Καθαρός Κώδικας
- Portability
- Αντικειμενοστραφής προγραμματισμός
- Δωρεάν
- Scripting language for Web applications
- Scientific computing (βιβλιοθήκες)

Μειονεκτήματα:

- Γλώσσα Ερμηνευσης
- Δύσκολη μετάφραση σε άλλη γλώσσα προγραμματισμού
- Δεν είναι κατάλληλη για λειτουργίες που έχουν να κάνουν με μνήμη

2.1.5. JavaScript

Η γλώσσα JavaScript είναι μια γλώσσα σεναρίων (scripting language) που βοηθά τον προγραμματιστή στην κατασκευή διαδραστικών και λειτουργικών εγγράφων στο Internet. Τα σενάρια γραμμένα σε JavaScript ενσωματώνονται στον κώδικα HTML και εκτελούνται από τον web browser του χρήστη (client-side processing). Για να μπορέσει ένας browser να εκτελέσει ένα σενάριο γραμμένο σε JavaScript θα πρέπει να διαθέτει έναν διερμηνευτή JavaScript (interpreter). Οι γνωστότεροι browsers Firefox, Internet Explorer, Opera ενσωματώνουν διερμηνευτή της JavaScript.

Μεταξύ των δυνατοτήτων που δίνει η Javascript είναι:

- Έλεγχος στο περιεχόμενο και στην εμφάνιση των ιστοσελίδων
- Έλεγχος της συμπεριφοράς και των λειτουργιών του browser
- Αλληλεπίδραση με φόρμες HTML
- Αλληλεπίδραση με το χρήστη με τη βοήθεια γεγονότων
- Ανάγνωση ή καταγραφή του H/Y του χρήστη μέσω cookies
- Δυνατότητα για εναλλαγή εικόνων

Από άποψη περιορισμών, η Javascript δεν επιτρέπει προσπέλαση ή χειρισμό αρχείων στον H/Y του χρήστη, για λόγους ασφαλείας. Επίσης, τα σενάρια που εκτελούνται στο web browser του χρήστη δεν μπορεί να ενημερώσει αρχεία που βρίσκονται στο web server. Πολλοί πιστεύουν ότι το JavaScript είναι Java λόγω της ομοιότητας του ονόματος. Αυτό, όμως, δεν είναι αλήθεια. Πιστεύουμε ότι θα έπαιρνε πολύ να καταλάβουμε όλες τις διαφορές - γι' αυτό απλώς απομνημονεύουμε ότι το Javascript δεν είναι Java.

Η Javascript δεν θα πρέπει να συγχέεται με τη Java, που είναι διαφορετική γλώσσα προγραμματισμού και με διαφορετικές εφαρμογές. Η χρήση της λέξης "Java" στο όνομα της γλώσσας έχει περισσότερη σχέση με το προφίλ του προϊόντος που έπρεπε να έχει και λιγότερο με κάποια πιθανή συμβατότητα ή άλλη στενή σχέση με τη Java. Ρόλο σε αυτήν τη σύγχυση έπαιξε και ότι η Java και η Javascript έχουν δεχτεί σημαντικές επιρροές από τη γλώσσα C, ειδικά στο συντακτικό, ενώ είναι και οι δύο αντικειμενοστραφείς γλώσσες. Τονίζεται ότι ο σωστός τρόπος γραφής της είναι "Javascript" και όχι "Java script" σαν δύο λέξεις, όπως λανθασμένα γράφεται ορισμένες φορές.

Τι χρειάζεται ώστε να τρέξουν scripts γραμμένα σε JavaScript; Χρειαζόμαστε ένα browser που υποστηρίζει JavaScript - παραδείγματος χάριν το Netscape Navigator (από την έκδοση 2.0) ή το Microsoft Internet Explorer (MSIE εν συντομία - από την έκδοση 3.0). Εφόσον αυτοί οι δύο browsers είναι πολύ διαδεδομένοι, πολλοί χρήστες έχουν την δυνατότητα να τρέξουν scripts γραμμένα σε JavaScript.

Αυτό είναι βεβαίως ένας σημαντικός λόγος για να διαλέξουμε JavaScript για να εμπλουτίσουμε τις web- σελίδες μας. Βεβαίως χρειάζεται πρώτα να κατανοήσουμε βασικές λειτουργίες της HTML .

Ένα παράδειγμα κώδικα JavaScript το βλέπουμε παρακάτω :

```
<html>
<body>
<h1>My First Web Page</h1>
<script type="text/javascript">
document. write("<p>" + Date() + "</p>");
</script>
</body>
</html>
```

Ο παραπάνω κώδικας θα μας εμφανίσει στην οθόνη το παρακάτω : My First Web Page

Σε browsers που δεν υποστηρίζουν την JavaScript, θα εμφανιστεί το JavaScript, όπως το περιεχόμενο της σελίδας. Για να τους εμποδίσουμε να το κάνουν αυτό και ως μέρος της JavaScript προτύπου, η ετικέτα σχολίου HTML που θα πρέπει να χρησιμοποιηθεί για να "κρύψει" το JavaScript. Απλά προσθέτουμε ένα σχόλιο HTML ετικέτα (στο τέλος του σχολίου) μετά την τελευταία δήλωση της JavaScript, όπως αυτός <!-- πριν από την πρώτη δήλωση JavaScript, και ένα!>, όπως το παρακάτω παράδειγμα:

```
<html>
<body>
<script type="text/javascript">
<!--
document.getElementById("demo").innerHTML=Date();
//-->
</script>
</body>
</html>
```

Μοντέλο εκτέλεσης

Η αρχική έκδοση της Javascript βασίστηκε στη σύνταξη στη γλώσσα προγραμματισμού C, αν και έχει εξελιχθεί, ενσωματώνοντας πια χαρακτηριστικά από νεότερες γλώσσες.

Αρχικά χρησιμοποιήθηκε για προγραμματισμό από την πλευρά του πελάτη (client), που ήταν ο φυλλομετρητής (browser) του χρήστη, και χαρακτηρίστηκε σαν client-side γλώσσα προγραμματισμού. Αυτό σημαίνει ότι η επεξεργασία του κώδικα Javascript και η παραγωγή του τελικού περιεχομένου HTML δεν πραγματοποιείται στο διακομιστή, αλλά στο πρόγραμμα περιήγησης των επισκεπτών, ενώ μπορεί να ενσωματωθεί σε στατικές σελίδες HTML. Αντίθετα, άλλες γλώσσες όπως η PHP εκτελούνται στο διακομιστή (server-side γλώσσες προγραμματισμού).

Παρά την ευρεία χρήση της Javascript για συγγραφή προγραμμάτων σε περιβάλλον φυλλομετρητή, αξίζει να σημειωθεί ότι από την αρχή χρησιμοποιήθηκε και για τη συγγραφή κώδικα από την πλευρά του διακομιστή, από την ίδια τη Netscape στο προϊόν LiveWire, με μικρή επιτυχία. Η χρήση της Javascript στο διακομιστή εμφανίζεται πάλι σήμερα, με τη διάδοση του Node.js, ενός μοντέλου προγραμματισμού βασισμένο στα γεγονότα (events).

Δείγμα κώδικα Javascript

Ο κώδικας Javascript μιας σελίδας περικλείεται από τις ετικέτες της HTML <script type="text/javascript"> και </script>.

Για παράδειγμα, ο ακόλουθος κώδικας Javascript εμφανίζει ένα πλαίσιο διαλόγου με το κείμενο "Γεια σου, κόσμε!":

```
<script type="text/javascript">
alert("Γεια σου, κόσμε!");
</script>
```

Αν ο κώδικας Javascript περιέχει περισσότερες από μία εντολές, αυτές θα πρέπει να διαχωριστούν μεταξύ τους με το χαρακτήρα του ελληνικού ερωτηματικού ';' (δηλαδή της λατινικής άνω τελείας). Η χρήση του χαρακτήρα αυτού για την τελευταία εντολή δεν είναι απαραίτητη. Η διαχώριση των εντολών στους νεότερους φυλλομετρητές (browsers) δεν είναι απαραίτητη.

Μια άλλη βασική εντολή, η `window.prompt("μήνυμα προς το χρήστη")`, ζητάει από το χρήστη να συμπληρώσει ένα κομμάτι μιας αίτησης απευθείας ώστε τα δεδομένα να χρησιμοποιηθούν σαν κείμενο:

```
<script>
var FIRSTvariable = window.prompt("PLEASE FILL IN YOUR NAME")
alert("Your name is " + FIRSTvariable + ".")</script>
```

Πλεονεκτήματα:

Η JavaScript είναι μια εξαιρετική λύση για τις εφαρμογές, όταν χρησιμοποιούνται φόρμες εισόδου από την πλευρά του client. Αυτό σημαίνει ότι εάν ένας χρήστης ξεχάσει να εισάγει το όνομα του σε μια φόρμα, μια λειτουργία JavaScript επικύρωσης μπορεί να εμφανίσει αναδυόμενο ένα μήνυμα για να τον ενημερώσει σχετικά με την παράλειψη. Αυτή είναι μια πολύ καλύτερη λύση από το να έχει η πλευρά του server μια ρουτίνα επικύρωσης για να χειριστεί το σφάλμα, επειδή ο server δεν έχει να κάνει οποιαδήποτε πρόσθετη επεξεργασία. Επίσης, μια ρουτίνα asp ή php θα μπορούσε να γραφτεί για να επιτευχθεί το ίδιο έργο, αλλά η JavaScript δεν θα επέτρεπε στη φόρμα υποβληθεί εάν δεν είχε συμπληρωθεί σωστά από την αρχή, μια πολύ πιο ισχυρή λύση!

Ένας άλλος τομέας όπου η JavaScript υπερέχει είναι η δημιουργία δυναμικών εφέ, όπως rollover εικόνες και scripted slideshows, όπου η χρήση της έχει γίνει κοινή. Επειδή η JavaScript τρέχει μέσα στο πρόγραμμα περιήγησης του πελάτη μπορεί να χρησιμοποιηθεί για να αλλάξει την εμφάνιση της οθόνης των χρηστών αφού έχει ολοκληρωθεί η αποστολή της σελίδας από τον server. Αυτό του επιτρέπει να δημιουργήσει μερικά πολύ εντυπωσιακά δυναμικά αποτελέσματα εικόνας.

Μειονεκτήματα:

Ένα από τα σημαντικότερα μειονεκτήματα για τη χρήση JavaScript είναι ότι τείνει να φορτώνει τις ιστοσελίδες. Ο κώδικας JavaScript μπορεί γρήγορα να προσθέσει εκατοντάδες γραμμές κώδικα, αν το χρησιμοποιούν για να κάνουν οτιδήποτε, ακόμη και εξ αποστάσεως, ενδιαφέρον. Το πρόβλημα του μεγάλου κώδικα JavaScript είναι εύκολο να λυθεί με την αποθήκευση του κώδικα σε ξεχωριστά αρχεία πηγών κώδικα JavaScript που έχουν μια επέκταση .Js . Αυτό καθαρίζει τον κώδικα της σελίδας, επειδή ο κώδικας JavaScript αποθηκεύεται χωριστά στην ίδια σελίδα HTML, αφήνοντας μια πολύ καθαρότερη και πιο εύχρηστη ιστοσελίδα.

Λόγω της τάσης της JavaScript να φορτώνει τις ιστοσελίδες μπορεί να είναι πολύ επιζήμια για την μηχανή αναζήτησης της ιστοσελίδας σας. Αυτό συμβαίνει γιατί όταν μια μηχανή αναζήτησης φθάνει στο site σας ψάχνει για ποιοτικό περιεχόμενο και λέξεις-κλειδιά για να καθορίσει το περιεχόμενο της σελίδα σας. Το τελευταίο πράγμα που θέλει να δει είναι εκατοντάδες γραμμές κώδικα JavaScript. Και πάλι, το πρόβλημα αυτό λύνεται εύκολα με τακτοποιημένη αποθήκευση κώδικα JavaScript στα script αρχεία με επέκταση .JS και σύνδεση με το αρχείο δέσμης ενεργειών σε έγγραφο HTML σας.

2.1.6. JSP (JavaServer Pages)

Οι JavaServer Pages (JSP) είναι μια τεχνολογία Java που βοηθά τους προγραμματιστές λογισμικού να εξυπηρετούν τις δυναμικά δημιουργούμενες ιστοσελίδες που βασίζονται σε HTML, XML, ή άλλους τύπους εγγράφων. Κυκλοφόρησε το 1999, ως απάντηση της Sun στην ASP και PHP, η JSP σχεδιάστηκε για να αντιμετωπίσει την αντίληψη ότι το περιβάλλον προγραμματισμού Java δεν παρέχει στους προγραμματιστές αρκετή υποστήριξη για τον Ιστό.

Για να αναπτύξετε και να εκτελέσετε ένα συμβατό web server με Servlet απαιτείται container. Οι Java Servlet και JavaServer Pages (JSP) προδιαγραφές από την Sun Microsystems και την JCP πρέπει αφενός να καλυφθούν από το container.

Επισκόπηση

Αρχιτεκτονικά, η JSP μπορεί να θεωρηθεί ως μια υψηλού επιπέδου αφαίρεση των Java servlets. Οι σελίδες JSP φορτώνονται στον server και λειτουργούν από ένα ειδικά δομημένο εγκατεστημένο πακέτο Java Server που λέγεται Java EE Web Application, συνήθως σκευαζόμενο ως .war ή .ear αρχείο.

Η JSP επιτρέπει κώδικα Java και ορισμένες προκαθορισμένες δράσεις που θα περιπλέκονται με στατικό περιεχόμενο σήμανσης web, με τη τελική σελίδα να δημιουργείται και να εκτελείται στον server για να παραδώσει ένα HTML ή XML έγγραφο. Η δημιουργημένες σελίδες και τα οι εξαρτώμενες βιβλιοθήκες Java χρησιμοποιούν Java bytecode αντί για ένα κλασσικό λογισμικό, και συνεπώς πρέπει να εκτελεστεί μέσα σε εικονική μηχανή Java(JVM) που ενσωματώνει το λειτουργικό σύστημα υποδοχής να παρέχει ένα αφηρημένο ανεξάρτητα από πλατφόρμα περιβάλλον.

Η JSP σύνταξη είναι ένα ρευστό μείγμα από δύο βασικές μορφές περιεχομένου: τα στοιχεία scriptlet και markup. Το markup είναι συνήθως πρότυπο HTML ή XML, ενώ τα στοιχεία scriptlet είναι οριοθετημένα blocks του κώδικα της Java που μπορεί να είναι αναμειγμένα με το markup. Όταν ζητηθεί η σελίδα, εκτελείται ο κώδικας Java και προστίθεται το αποτέλεσμα, σε συνδυασμό με τον περιβάλλοντα markup για να δημιουργήσει την τελευταία σελίδα. Οι σελίδες JSP πρέπει να καταρτίζονται σε κλάσεις Java bytecode πριν να μπορούν να εκτελούνται, αλλά η σύνταξη είναι απαραίτητη μόνο όταν έχει συμβεί μια αλλαγή στο πηγαίο JSP αρχείο.

Ο κώδικας Java δεν απαιτείται να είναι πλήρης (αυτόνομα) εντός ενός scriptlet μπλοκ, αλλά μπορεί να μην συμπίπτει με τη σήμανση του περιεχομένου που παρέχει τη σελίδα ολόκληρη συντακτικά σωστή (για παράδειγμα, οποιοδήποτε Java if / for / while, οι οποία χρησιμοποιείται σε ένα στοιχείο scriptlet και πρέπει να κλείσει σωστά σε ένα μεταγενέστερο στοιχείο για την σελίδα για την επιτυχή μεταγλώττιση). Αυτό το σύστημα του διασπασμένου εσωτερικού τομέα κώδικα ονομάζεται step over scripting επειδή μπορεί να τυλιχτεί γύρω από τη στατική σήμανσης με την ενίσχυση από πάνω του. Η σήμανση η οποία εμπίπτει μέσα σε μια διάσπαση μπλοκ του κώδικα υπόκειται στον εν λόγω κώδικα, έτσι η σήμανσης μέσα σε ένα if μπλοκ, θα εμφανίζονται μόνο στην έξοδο, όταν η συνθήκη if αποτιμάται σε true. Επίσης η σήμανση μέσα σε μια κατασκευή βρόχου μπορεί να εμφανιστεί πολλές φορές στην έξοδο, ανάλογα με πόσες φορές λειτουργεί το σώμα βρόχου.

Η σύνταξη JSP προσθέτει επιπλέον ετικέτες τύπου XML, που ονομάζονται JSP actions, να επικαλεσθεί ενσωματωμένη λειτουργικότητα. Επιπλέον, η τεχνολογία επιτρέπει τη δημιουργία των tag JSP βιβλιοθηκών που λειτουργούν ως επεκτάσεις στο πρότυπο ετικέτας HTML ή XML. Οι JVM βιβλιοθήκες tag παρέχουν μια πλατφόρμα ανεξαρτήτου τρόπου για την αύξηση των δυνατοτήτων ενός web server. Σημειώστε ότι δεν είναι όλοι οι εμπορικοί διακομιστές Java συμβατοί με προδιαγραφές Java EE.

Από την έκδοση 1.2 των προδιαγραφών JSP, οι JavaServerPages έχουν αναπτυχθεί στο πλαίσιο της διαδικασίας κοινοτικής Java. Το JSR 53 καθορίζει το JSP 1.2 και το JSP Servlet 2.3 και το JSR 152 ορίζει τις JSP 2.0 προδιαγραφές. Από το Μάιο του 2006, η JSP 2.1 έχει κυκλοφορήσει κάτω από JSR 245 ως μέρος της Java EE 5. Από τις 10 Δεκέμβρη του 2009 η JSP 2.2 έχει κυκλοφορήσει ως ένα δελτίο συντήρησης JSR 245.

Πλεονεκτήματα:

Η JSP έχει διάφορα πλεονεκτήματα πέρα από τις πολλές εναλλακτικές λύσεις της. Παρακάτω παρουσιάζονται μερικά από αυτά και τα οποία είναι:

- Είναι ευκολότερη και ταχύτερη η ανάπτυξη τους, κυρίως για μικρά Projects.
- Με τους servlets είναι δυσκολότερη η παραγωγή στατικού HTML κώδικα, χρειάζονται πολλές `out.println` εντολές, π.χ.
`out.println("<body>");`
`out.println("<h2>Hello " + user + "</h2>");`
- Δίνει έμφαση στην μορφή και παρουσίαση της σελίδας (HTML) και αφήνουν την προγραμματιστική λογική σε Java κώδικα.
- Αν έχουμε έτοιμο JSP κώδικα, μπορεί εύκολα να ενσωματωθεί σε HTML σελίδες από web authors που δε γνωρίζουν προγραμματισμό.
- Σε σύγκριση με τις Active Server Pages (ASP):

Η ASP είναι μια ανταγωνιστική τεχνολογία προερχόμενη από τη Microsoft. Τα πλεονεκτήματα της JSP είναι διπλά. Κατ' αρχάς, το δυναμικό μέρος γράφεται σε Java, κι όχι σε VBScript ή μια άλλη ASP-specific γλώσσα, κι έτσι αυτό την κάνει να είναι ισχυρότερη και καλύτερη που ταιριάζει στις σύνθετες εφαρμογές που απαιτούν τα επαναχρησιμοποιήσιμα συστατικά. Δεύτερον, η JSP είναι φορητή σε άλλα λειτουργικά συστήματα και Servers δικτύου στα οποία δεν είμαστε κλειδωμένοι στα Windows NT/2000 και IIS. Θα μπορούσαμε να προβάλλουμε το ίδιο επιχείρημα κατά τη σύγκριση της JSP με ColdFusion με JSP που μπορούμε να χρησιμοποιήσουμε την Java και δεν είμαστε δεμένοι σε ένα κατάλληλο προϊόν Server.

- Σε σύγκριση με την PHP:
Η PHP είναι ελεύθερη, ανοικτού-κώδικα HTML-embedded scripting γλώσσα που είναι κάπως παρόμοια τόσο με την ASP, όσο επίσης, και με την JSP. Το πλεονέκτημα της JSP είναι ότι το δυναμικό μέρος γράφεται σε Java, για την οποία πιθανώς ήδη είμαστε ενήμεροι, το οποίο έχει ήδη ένα εκτενές API για τη δικτύωση, την πρόσβαση των βάσεων δεδομένων, τα διανεμημένα αντικείμενα, και τους ομοίους, ενώ η PHP απαιτεί μια εξ' ολοκλήρου νέα γλώσσα.
- Σε σύγκριση με τα καθαρά Servlets:
Η JSP δεν παρέχει οποιεσδήποτε ικανότητες που δεν θα μπορούσαν να ολοκληρωθούν σε γενικές γραμμές με ένα servlet. Στην πραγματικότητα, τα έγγραφα JSP είναι αυτόματα μεταφρασμένα στα servlets πίσω από τις σκηνές. Αλλά είναι καταλληλότερη να γράψει, αλλά και για να τροποποιήσει κανονικό HTML από το να έχει τεράστιο πλήθος δηλώσεων println που παράγουν το HTML. Επιπλέον, με το χωρισμό της παρουσίασης από το περιεχόμενο, μπορούμε να βάλουμε διαφορετικούς ανθρώπους σε διαφορετικούς στόχους: οι δικοί μας εμπειρογνώμονες σχεδίου ιστοσελίδας, μπορούν να χτίσουν το HTML χρησιμοποιώντας τα εξοικειωμένα εργαλεία και να αφήσουν τις θέσεις για τους προγραμματιστές μας των servlet για να παρεμβάλουν το δυναμικό περιεχόμενο.
- Σε σύγκριση με τα Server-Side Includes (SSI):
Τα SSI είναι μια ευρέως υποστηριγμένη τεχνολογία για την είσοδο των εξωτερικά καθορισμένων κομματιών στατικής ιστοσελίδας. Η JSP είναι καλύτερη επειδή έχουμε ένα πλουσιότερο σύνολο εργαλείων για το χτίσιμο αυτού του εξωτερικού κομματιού κι έχουμε περισσότερες επιλογές σχετικά με το στάδιο της απάντησης HTTP στην οποία το κομμάτι εισέρχεται πραγματικά. Εκτός αυτού, τα SSI προορίζονται πραγματικά μόνο για τους απλούς συνυπολογισμούς, κι όχι για τα "πραγματικά" προγράμματα τα οποία χρησιμοποιούν form data, κάνουν τις συνδέσεις των βάσεων δεδομένων, και τους ομοίους.
- Σε σύγκριση με την JavaScript:
Η JavaScript, που είναι απολύτως ευδιάκριτη από τη γλώσσα προγραμματισμού της Java, χρησιμοποιείται κανονικά για να παράγει το HTML δυναμικά στον client, που χτίζει τα μέρη ιστοσελίδας δεδομένου ότι ο browser «φορτώνει» το έγγραφο. Αυτή είναι μια χρήσιμη ικανότητα αλλά μόνο στην περίπτωση όταν χειρίζεται τις καταστάσεις όπου οι δυναμικές πληροφορίες είναι βασισμένες στο περιβάλλον του client. Με εξαίρεση τα Cookies, το στοιχείο αιτήματος HTTP δεν είναι διαθέσιμο στις client-side ρουτίνες JavaScript. Κι, από τις ρουτίνες ελλείψεων JavaScript για τον προγραμματισμό δικτύων, ο κώδικας JavaScript στον client δεν μπορεί να προσπελάσει server-side πόρους όπως τις βάσεις δεδομένων, τους καταλόγους, τις πληροφορίες τιμολόγησης, και τους ομοίους. Η JavaScript μπορεί επίσης, να χρησιμοποιηθεί στον Server, κι ειδικότερα στους Servers Netscape κι ως scripting γλώσσα για IIS. Η Java είναι με μεγάλη διαφορά πιο ισχυρή, εύκαμπτη, αξιόπιστη, και φορητή.
- Σε σύγκριση με την στατική HTML:
Το κανονικό HTML, φυσικά, δεν μπορεί να περιέχει τις δυναμικές πληροφορίες, κι έτσι οι στατικές σελίδες HTML δεν μπορούν να βασιστούν στις πηγές στοιχείων εισόδου ή στα serverside δεδομένα πηγών. Η JSP είναι τόσο εύκολη και κατάλληλη που είναι αρκετά λογικό να αυξηθούν οι σελίδες HTML που ωφελούνται μόνο ελαφρώς από την εισαγωγή των δυναμικών στοιχείων. Προηγουμένως, η δυσκολία της χρησιμοποίησης των δυναμικών στοιχείων απέκλεισε τη χρήση της σε όλες εκτός από τις πολυτιμότερες περιπτώσεις.

Μειονεκτήματα:

- Εύρεση και διόρθωση λαθών (debugging) είναι πολύπλοκη.
- Μεγάλες ποσότητες ενσωματωμένου κώδικα σε scriplets μπορεί να οδηγήσουν σε σελίδες που είναι δύσκολο να συντηρηθούν
- Η απόδοση JSPs είναι κατώτερη των servlets (το compilation 200 JSPs θα δημιουργήσει 200 servlets).
- Αν χρησιμοποιούσαμε τεχνολογία Servlets θα μπορούσαμε να ενσωματώσουμε περισσότερη λειτουργικότητα σε λιγότερους servlets.

2.1.7. Java Servlet

Ένα servlet είναι μια κλάση προγραμματισμού Java που χρησιμοποιείται για να επεκτείνουν τις δυνατότητες των servers που φιλοξενούν τις εφαρμογές προσβάσιμο μέσω ενός μοντέλου προγραμματισμού αίτημα-απάντηση. Αν και τα servlets μπορούν να ανταποκριθούν σε κάθε τύπο αιτήματος, χρησιμοποιούνται συνήθως για την επέκταση των εφαρμογών που φιλοξενούνται από τους Web Servers. Έτσι, μπορεί να θεωρηθεί ως μια βοηθητική εφαρμογή Java που τρέχει σε ένα διακομιστή αντί σε ένα πρόγραμμα περιήγησης.

Ένα Servlet είναι μια κλάση Java σε Java EE που συμμορφώνεται με το Java Servlet API, ένα πρωτόκολλο με το οποίο μια κλάση Java μπορεί να ανταποκριθεί στα αιτήματα. Δεν είναι συνδεδεμένη με ένα συγκεκριμένο πρωτόκολλο client-server, αλλά πιο συχνά χρησιμοποιείται με το πρωτόκολλο HTTP. Ως εκ τούτου, η λέξη "Servlet" χρησιμοποιείται συχνά κατά την έννοια του "Servlet HTTP". Έτσι, ένας προγραμματιστής μπορεί να χρησιμοποιήσει ένα servlet για να προσθέσετε δυναμικό περιεχόμενο σε ένα διακομιστή Web που χρησιμοποιούν την πλατφόρμα Java. Το περιεχόμενο που δημιουργείται συνήθως είναι HTML, αλλά μπορεί να είναι και άλλα δεδομένα, όπως XML. Οι Servlets πρόκειται για το αντίστοιχο Java για να μη δυναμικές τεχνολογίες περιεχομένου στον Παγκόσμιο Ιστό, όπως CGI και ASP.NET. Οι Servlets μπορούν να διατηρήσουν σε μεταβλητές την κατάσταση περιόδου λειτουργίας σε πολλές συναλλαγές με τη χρήση HTTP cookies, ή URL επανεγγραφή.

Για να αναπτύξετε και να την εκτελέσετε, μπορεί να χρησιμοποιηθεί ο Apache Tomcat. Είναι ένα servlet ανοιχτού κώδικα που έχει αναπτυχθεί από την Apache Software Foundation (ASF). Ο Tomcat εφαρμόζει τις Java Servlet και τις JavaServer Pages (JSP) προδιαγραφές από την Sun Microsystems, και παρέχει μια "καθαρή Java" HTTP περιβάλλον του web server για κώδικα Java για να τρέξει.

Το Servlet API, που περιέχονται στο πακέτο Java στην ιεραρχία javax.servlet, καθορίζει τις αναμενόμενες αλληλεπιδράσεις ενός δοχείου Web και ενός servlet. Ένα δοχείο Web είναι ουσιαστικά το συστατικό ενός διακομιστή Web που αλληλεπιδρά με τα servlets. Το δοχείο Web είναι υπεύθυνο για τη διαχείριση του κύκλου ζωής των servlets, χαρτογραφώντας ένα URL σε ένα συγκεκριμένο Servlet και την εξασφάλιση ότι ο αιτών URL έχει τα σωστά δικαιώματα πρόσβασης.

Ένα Servlet είναι ένα αντικείμενο που λαμβάνει μια αίτηση και παράγει μια απάντηση βασιζόμενη σε αυτό το αίτημα. Το βασικό πακέτο servlet καθορίζει Java αντικείμενα για να αντιπροσωπεύουν αιτήματα Servlet και απαντήσεις, καθώς και αντικείμενα για να αντανakλά τις παραμέτρους διαμόρφωσης του servlet και το περιβάλλον εκτέλεσης. Το πακέτο javax.servlet.http καθορίζει HTTP υποκλάσεις των γενικών στοιχείων servlet, συμπεριλαμβανομένων των αντικειμένων διαχείρισης συνεδρίας που παρακολουθούν πολλαπλά αιτήματα και τις απαντήσεις μεταξύ του Web server και ενός πελάτη. Τα Servlets μπορούν να συσκευαστούν σε ένα αρχείο WAR ως μια εφαρμογή Web.

Οι Servlets μπορούν να παραχθούν αυτόματα από JavaServer Pages (JSP) από τον compiler JavaServer Pages. Η διαφορά μεταξύ Servlets και JSP είναι ότι οι Servlets ενσωματώνουν συνήθως μέσα σε HTML κώδικα Java, ενώ JSPs ενσωματώνουν Java κώδικα σε HTML. Ενώ η άμεση χρήση των Servlets να παράγει HTML (όπως φαίνεται στο παρακάτω παράδειγμα) είναι σχετικά σπάνια στις μέρες μας, του ψηλότερο επίπεδο MVC web πλαισίου σε Java EE (JSF) εξακολουθεί να χρησιμοποιεί ρητά την Servlet τεχνολογία για το χαμηλό επίπεδο χειρισμού αίτηση /απάντηση μέσω της FacesServlet. Μια κάπως μεγαλύτερης ηλικίας χρήση είναι η χρήση servlets, σε συνδυασμό με JSPs

σε ένα σχέδιο που ονομάζεται "Μοντέλο 2", η οποία είναι μια γεύση από το μοντέλο model-view-controller.

Ιστορία

Η πλήρης προδιαγραφή servlet δημιουργήθηκε από την Sun Microsystems, με την έκδοση 1.0 που ολοκληρώθηκε τον Ιούνιο του 1997. Από την έκδοση 2.3, οι προδιαγραφές servlet αναπτύχθηκαν στο πλαίσιο της διαδικασίας κοινοτικής Java.JSR 53 το οποίο ορίζεται τόσο από τις Servlet 2.3 και JavaServer Page 1.2 προδιαγραφές. Το JSR154 διευκρινίζει τις Servlet 2.4 και 2.5 προδιαγραφές. Από το Μάρτιο του 26, 2010, η τρέχουσα έκδοση των προδιαγραφών servlet είναι 3.0.

Στο blog του στο java.net, ο Sun βετεράνος και ο GlassFish οδηγούν τον Jim Driscoll με λεπτομέρειες από την ιστορία της servlet τεχνολογίας. Ο James Gosling σκέφτηκε πρώτος του servlets κατά τις πρώτες ημέρες της Java, αλλά η ιδέα δεν είχε γίνει προϊόν μέχρι να αποσταλεί το Web Java προϊόν Server από την Sun. Αυτό ήταν πριν από αυτό που είναι τώρα η πλατφόρμα Java Platform, η έκδοση Enterprise έγινε σε μια προδιαγραφή.

Πλεονεκτήματα σε σχέση με το CGI

Τα πλεονεκτήματα από τη χρήση servlets είναι οι γρήγορες επιδόσεις τους και η ευκολία χρήσης σε συνδυασμό με περισσότερη δύναμη πέρα από την παραδοσιακή CGI (Common Gateway Interface). Τα παραδοσιακά CGI scripts γραμμένα σε Java έχουν μια σειρά από μειονεκτήματα, όταν πρόκειται για την απόδοση:

Όταν γίνεται ένα αίτημα HTTP, μια νέα διαδικασία δημιουργείται για κάθε κλήση του CGI script. Αυτή η επιβάρυνση της δημιουργίας διαδικασίας μπορεί να είναι ένα σύστημα υψηλής έντασης, ειδικά όταν το script έχει αρκετά γρήγορες εργασίες. Έτσι, η δημιουργία της διαδικασίας θα χρειαστεί περισσότερο χρόνο από ό, τι η εκτέλεση του CGI κώδικα. Τα Java servlets το έλυσαν αυτό, καθώς το servlet δεν είναι μια ξεχωριστή διαδικασία. Κάθε αίτηση από ένα servlet γίνεται από ένα ξεχωριστό νήμα Java στο πλαίσιο της διαδικασίας διακομιστή Web, παραλείποντας ξεχωριστή διακλάδωση από το δαίμονα HTTP.

Ταυτόχρονα CGI αιτήματα έχουν ως αποτέλεσμα το CGI script να αντιγραφεί και να φορτωθεί στη μνήμη όσες φορές υπάρχουν αιτήματα. Ωστόσο, με τα servlets, υπάρχουν την ίδια ποσότητα threads όσο και αιτήματα, αλλά θα υπάρξει μόνο ένα αντίγραφο του servlet class που θα δημιουργείται στη μνήμη που μένει εκεί, επίσης, μεταξύ των αιτημάτων. Μόνο ένα instance απαντά σε όλα τα αιτήματα ταυτόχρονα. Αυτό μειώνει τη χρήση της μνήμης και καθιστά τη διαχείριση των επίμονων δεδομένων εύκολη.

Ένα servlet μπορεί να διευθύνεται από ένα servlet engine σε ένα περιοριστικό περιβάλλον, που ονομάζεται sandbox. Αυτό είναι παρόμοιο με ένα applet που τρέχει στο sandbox του προγράμματος περιήγησης στο Web. Αυτό κάνει μια περιοριστική χρήση των επιβλαβείς servlets δυνατή.

Ο κύκλος ζωής ενός servlet

1. Το δοχείο ζητά τον no-arg constructor.
2. Το δοχείο Web καλεί την init () μέθοδο. Η μέθοδος αυτή αρχικοποιεί το servlet και πρέπει να κληθεί πριν από τη ζωή ενός servlet. Η init () μέθοδος καλείται μόνο μία φορά.
3. Μετά την προετοιμασία, το servlet μπορεί να εξυπηρετήσει αιτήματα πελατών. Κάθε αίτηση εξυπηρετείται σε δικό του ξεχωριστό thread. Το δοχείο Web κλήσεις την service () μέθοδο του servlet για κάθε αίτηση. Η μέθοδος αυτή προσδιορίζει το είδος του αιτήματος που γίνεται και το μετατρέπει σε μια κατάλληλη μέθοδο για να χειριστεί το αίτημα. Ο δημιουργός του servlet πρέπει να παρέχει μια υλοποίηση για τις μεθόδους αυτές. Αν δημιουργείτε μια αίτηση για μια μέθοδο που δεν τίθεται σε εφαρμογή από το servlet, καλείται η μέθοδος της γονικής κλάσης, συνήθως με αποτέλεσμα ένα σφάλμα που επιστρέφεται στον αιτούντα.
4. Τέλος, το δοχείο Web ζητά την μέθοδο destroy() που λαμβάνει από το servlet της υπηρεσίας. Η destroy() ,όπως και η init () καλείται μόνο μία φορά στον κύκλο ζωής ενός servlet.

Χρήση

Τα Servlets χρησιμοποιούνται πιο συχνά για να:

- επεξεργασθούν ή αποθηκεύσουν τα δεδομένα που υποβλήθηκαν από μια φόρμα HTML
- παρέχουν δυναμικό περιεχόμενο, όπως τα αποτελέσματα ενός ερωτήματος βάσης δεδομένων
- διαχειρίζονται πληροφορίες κατάστασης που δεν υπάρχουν στο ανιθαγενείς πρωτόκολλο TTP, όπως η συμπλήρωση των άρθρων στο καλάθι αγορών του κατάλληλου πελάτη.

Πλεονεκτήματα:

Με τα servlets μπορούμε να αναπτύξουμε πολύ ισχυρές web εφαρμογές που έχουν τα εξής πλεονεκτήματα:

- Βασίζονται και αναπτύσσονται στην Java και όχι σε μία script γλώσσα. Έτσι οι προγραμματιστές των servlets έχουν πρόσβαση στο Java API και τις μεγάλες δυνατότητες που δίνει. Άλλες τεχνολογίες, όπως το PHP, βασίζονται σε script γλώσσες που έχουν αναπτυχθεί ειδικά γι' αυτό το σκοπό. Με την δυνατότητα να χρησιμοποιήσουμε Java επωφελεύσατε εφόσον γνωρίζουμε ήδη τη Java.
- Είναι πολύ εύκολο να προσθέσουμε state σε εφαρμογές διαδικτύου. Το HTTP είναι ένα stateless πρωτόκολλο. Αυτό σημαίνει πως σε κάθε get που κάνει ο χρήστης ενός browser για μία ιστοσελίδα, για τον web server είναι σαν να ξεκινά από την αρχή. Με τα servlets υπάρχει η έννοια της συνεδρίας (session) με την οποία μπορούμε να κρατήσουμε πληροφορίες που σχετίζονται με την συνεδρία του χρήστη με το σύστημα που μπορούν να διατηρούνται ανάμεσα από διαδοχικά get. Αυτό είναι απαραίτητο σε πολλές εφαρμογές. Κλασικό παράδειγμα είναι το shopping cart σε μία εφαρμογή πώλησης προϊόντων μέσω Internet, στο οποίο ο χρήστης τοποθετεί προϊόντα καθώς πηγαίνει από σελίδα σε σελίδα. Όπως είναι αυτονόητο, το shopping cart θα πρέπει να συγκρατεί τα περιεχόμενά του ανάμεσα από διαδοχικές αιτήσεις για νέες σελίδες.
- Μπορούμε να έχουμε πρόσβαση σε βάσεις δεδομένων μέσω του JDBC API της Java, πράγμα που είναι απαραίτητο για όλες σχεδόν τις διαδικτυακές εφαρμογές των επιχειρήσεων.
- Έχουμε ανεξαρτησία από Λειτουργικό Σύστημα, μια και τα Servlets εκτελούνται από την Java Virtual Machine. Μία εξίσου ισχυρή τεχνολογία με τα Servlets/JSP που είναι το ASP.NET της Microsoft δεν διαθέτει αυτή τη δυνατότητα. Προς το παρόν ο μόνος web server που μπορεί να εκτελεί ASP.NET εφαρμογές είναι ο Internet Information Server της ίδιας εταιρείας που τρέχει σε Win2000 Professional, Win2000 Server και σε WinXP Professional. Άλλωστε το ίδιο .NET που είναι το Framework κομμάτι του οποίου αποτελεί το ASP.NET μπορεί και εγκαθίσταται μόνο στα προαναφερθέντα Λειτουργικά συστήματα της Microsoft.
- Έχουμε καλύτερη επίδοση (performance) από άλλες τεχνολογίες, όπως για παράδειγμα το CGI (Common Gateway Interface), διότι το μοντέλο διεργασιών που χρησιμοποιείται για τα servlets είναι αισθητά καλύτερο έναντι αυτών των τεχνολογιών. Το αν τα Servlets/JSP και ευρύτερα η πλατφόρμα J2EE έχουν καλύτερη επίδοση από το ASP.NET και το .NET framework ή το αντίθετο, όπως και το ποια είναι καλύτερη πλατφόρμα γενικότερα, είναι ένα θέμα ανοιχτής διαμάχης αυτό το διάστημα με ανακοινώσεις στο Internet από όλες τις ενδιαφερόμενες πλευρές.

Μειονεκτήματα:

- Χρησιμοποιεί πολύτιμη επεξεργαστική ισχύ του server.
- Κλιμάκωση (scalability)

2.2. Τεχνολογίες Βάσεων Δεδομένων

2.2.1. MySQL

Η MySQL είναι μια σχεσιακή βάση δεδομένων που μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Έλαβε το όνομά της από την κόρη του Μόντυ Βιντένιους, τη Μάι. Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων.

Ο κωδικός του εγχειρήματος είναι διαθέσιμος μέσω της GNU General Public License, καθώς και μέσω ορισμένων ιδιοκτητών συμφωνιών. Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, η οποία σήμερα ανήκει στην Oracle.

Σύνταξη και εντολές

Η Mysql είναι ένα δυνατό και γρήγορο σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Είναι ο περισσότερο διαδεδομένος database server και υποστηρίζεται από τα περισσότερα λειτουργικά συστήματα. Είναι λογισμικό ανοικτού κώδικα (Open Source software) έτσι διατίθεται δωρεάν στην διεύθυνση <http://dev.mysql.com/downloads/mysql> με άδεια ανοικτού κώδικα ή με χαμηλό κόστος για εμπορική άδεια. Ο database server είναι ένας πολυνηματικός πολλαπλών χρήσεων διακομιστής δηλαδή επιτρέπει να δουλεύουν πολλοί χρήστες ταυτόχρονα. Ελέγχει την πρόσβαση στα δεδομένα και επιτρέπει μόνο σε πιστοποιημένους χρήστες να έχουν πρόσβαση σε αυτά. Χρησιμοποιεί την **SQL (Structured Query Language)** γλώσσα ερωτημάτων για βάσεις Δεδομένων. Όπως όλα τα ΣΔΒΔ περιλαμβάνει εντολές για

Διαχείριση χρηστών

Δημιουργία : *CREATE USER username@hostname IDENTIFIED BY password;*

Διαγραφή χρήστη : *DROP USER ονομα_χρήστη@hostname;*

Διαχείριση βάσης δεδομένων

Δημιουργία βάσης δεδομένων : *CREATE DATABASE όνομα_βάσης;*

Διαγραφή βάσης δεδομένων : *DROP DATABASE όνομα_βάσης;*

Εμφάνιση όλων των βάσεων που έχει δικαιώματα ο χρήστης : *SHOW DATABASES;*

Σύνδεση με τη βάση : *Use όνομα_βάσης;*

Διαχείριση Δικαιωμάτων σε χρήστη

Εκχώρηση δικαιωμάτων σε χρήστη : *GRANT [privilege] ON [db][.table] TO user@host [IDENTIFIED BY password] [WITH GRANT OPTION];*

Ανάκληση : *REVOKE [privilege] ON [db][.table][.column] FROM user@host;*

Διαχείριση Πινάκων

Δημιουργία Πίνακα : *CREATE TABLE όνομ_πίνακα(όνομα_πεδίου τύπος πεδίου [NOT NULL] [DEFAULT value] , πεδίο τύπος πεδίου...,PRIMARY KEY(πεδία),[UNIQUE(πεδία)],[FOREIGN KEY(πεδίο σύνδεσης) REFERENCES όνομ_πίνακα(πεδίο σύνδεσης)] [ON DELETE RESTRICT | CASCADE | SET NULL | NO ACTION][ON UPDATE RESTRICT | CASCADE | SET NULL | NO ACTION]);*

Μεταβολή Δομής Πίνακα : *ALTER TABLE όνομα_πίνακα [ADD column πεδίο [first|after πεδίο]] [ADD constraint περιορισμός] [ADD primary key/index/unique(πεδίο1, ...)] [MODIFY όνομα_πεδίου τύπος πεδίου ...] [DROP πεδίο, περιορισμός, κλειδί κ.λ.π]*

Διαγραφή πίνακα: *DROP TABLE όνομα_πίνακα*

Αλλαγή ονόματος πίνακα: *rename table όνομα_πίνακα to νέο_όνομα*

Εμφάνιση πινάκων : *Show tables;* ή *show tables from όνομα_βδ;*

Εμφάνιση στηλών πίνακα : *Describe όνομα_πίν.;* ή *show columns from όνομα_πίν.;*

Διαχείριση ευρετηρίων

Δημιουργία ευρετηρίου : *CREATE [UNIQUE] INDEX όνομα_ευρετηρίου ON όνομα_πίνακα (πεδίο [ASC|DESC][, πεδίο [ASC|DESC], ...])*

Διαγραφή ευρετηρίου : *DROP INDEX όνομα_ευρετηρίου ON όνομα_πίνακα*

Εμφάνιση ευρετηρίων : *show index from όνομα_πίνακα;*

Διαχείριση Εγγραφών Πίνακα

Εισαγωγή εγγραφών: *INSERT INTO όνομα_πίνακα [(πεδίο1,πεδίο2...)]*

values(τιμή_πεδίου1,τιμή_πεδίου2...),(τιμή_πεδίου1,τιμή_πεδίου2...),...

Εισαγωγή εγγραφών από text αρχείο: Load data local infile 'διαδρομή/ονομα_αρχείου.txt' into table ονομα_πίνακα fields terminated by ',';

Εισαγωγή εγγραφών από πίνακα σε πίνακα : INSERT INTO ονομα_Πίνακα SELECT πεδίο1 ή τιμή1, πεδίο2 ή τιμή2,... FROM ονομα_Πίνακα WHERE συνθήκη;

Ερώτημα διαγραφής εγγραφών : DELETE FROM ονομα_Πίνακα WHERE συνθήκη;

Ερώτημα τροποποίησης εγγραφών : UPDATE όνομα_πίνακα SET πεδίο1=τιμή1 [,πεδίο2 = τιμή2,...] [WHERE συνθήκη]

Επιλογή εγγραφών από ένα πίνακα : SELECT [distinct] ονομα_πεδίου,... FROM όνομα_πίνακα [where συνθήκη] [order by ονομα_πεδίου [ASC|DESC], ...]

Όταν χρησιμοποιούμε παραπάνω από ένα πίνακες πρέπει να τους συνδέουμε με INNER ή LEFT ή RIGHT JOIN αλλιώς δημιουργείται καρτεσιανό γινόμενο

Π.χ. select grades.am, lastname, firstname, code, semester, grade from students left join grades on students.am=grades.am;

Πλεονεκτήματα:

- Είναι ένα πολύ γρήγορο και δυνατό σύστημα διαχείρισης βάσεων δεδομένων
- Ο MySQL διακομιστής ελέγχει την πρόσβαση στα δεδομένα, για να μπορούν να δουλεύουν πολλοί χρήστες ταυτόχρονα
- Μπορούν να υπάρχουν ταυτόχρονα περισσότερες από μια συνδέσεις με τη βάση χωρίς να υπάρχουν πολλαπλά αντίγραφα της, όπως συμβαίνει με άλλα συστήματα βάσεων δεδομένων
- Η απόδοσή της είναι καλύτερη σε μεγαλύτερο όγκο βάσεων δεδομένων
- Παρέχει ευκολίες στο backup
- Είναι ιδιαίτερα βελτιωμένη σε ταχύτητα για την ανάκτηση δεδομένων
- Είναι συμβατή και μεταφέρσιμη σε διάφορες πλατφόρμες και για διάφορα εργαλεία ανάπτυξης
- Είναι πιο κατάλληλη για χρήση στο Internet
- Είναι οικονομική
- Η MySQL είναι λογισμικό ανοιχτού κώδικα.

Μειονεκτήματα:

- Η MySQL δεν υποστηρίζει ένα πολύ μεγάλο μέγεθος της βάσης δεδομένων πολύ αποτελεσματικά.
- Η MySQL δεν υποστηρίζει ROLE, COMMIT, και Stored διαδικασίες στις εκδόσεις της κάτω από 5.
- Οι συναλλαγές δεν αντιμετωπίζονται πολύ αποτελεσματικά.

2.2.2. Oracle Database

Το Oracle Database είναι λογισμικό διαχείρισης σχεσιακών βάσεων δεδομένων. Είναι δημιουργία και ιδιοκτησία της εταιρείας Oracle Corporation.

Το Oracle αποτελείται από ένα σύστημα καταναμημένων διεργασιών. Υποστηρίζεται από διάφορων τύπων servers, όπως π.χ. ο Apache. Το σύστημα διαχείρισης της βάσης μπορεί να φορτωθεί σε υπολογιστές που χρησιμοποιούν λειτουργικά συστήματα Linux, Windows, Solaris, κ.ά.

Η Oracle SQL (Structured Query Language) είναι μια γλώσσα υπολογιστή που έχει σχεδιαστεί για να διαχειριστεί τα δεδομένα που αποθηκεύονται σε βάσεις δεδομένων RDBMS (σχεσιακά συστήματα διαχείρισης βάσεων δεδομένων). Το RDBMS είναι ένα σύστημα πινάκων που αποθηκεύει δεδομένα και αντιπροσωπεύει τις σχέσεις μεταξύ τους. Η SQL έχει διάφορα δομικά στοιχεία, τα clauses, τα expressions, τα queries, τα statements και τα predicates. Η Oracle SQL χρησιμοποιείται συνήθως για τη διαχείριση των δεδομένων που αποθηκεύονται σε βάσεις δεδομένων Oracle (ονομάζεται επίσης και Oracle RDBMS ή απλά Oracle).

Πλεονεκτήματα:

- Μικρή εφαρμογή, ελαφριά, και εύκολη στη διαχείρισή.
- Οικεία σε πολλούς προγραμματιστές.
- Τώρα παρέχεται παγκόσμιας κλάσης υποστήριξη.
- Θα ενισχυθεί από την κοινότητα της Oracle που συνεργάζεται με την κοινότητα MySQL.
- Μπορεί να αναπτυχθεί μαζί με την Oracle Database σε μια επιχείρηση εξυπηρετώντας διαφορετικές εφαρμογές και λειτουργίες των επιχειρήσεων.
- Συγκεντρωτικό σύστημα Διαχείρισης και Ελέγχου: Οι Oracle SQL δηλώσεις επιτρέπουν στα δεδομένα να ελέγχονται από ένα κεντρικό αποθετήριο πίνακα. Ένας διαχειριστής της βάσης δεδομένων (DBA) είναι υπεύθυνος για τη δημιουργία των χρηστών, να εκχωρεί δικαιώματα, να προσθέτει αρχεία, να διαγράφει περιττές πληροφορίες, να τροποποιεί τα υπάρχοντα δεδομένα και ερωτήματα. Αυτά τα κεντρικά αποθηκευμένα δεδομένα είναι κοινά και προσβάσιμα από διάφορες εφαρμογές. Αυτό εξαλείφει την εισαγωγή δεδομένων και πλεονασμό δεδομένων αποθήκευσης.
- Τυποποίηση: Ένα σημαντικό πλεονέκτημα της Oracle SQL είναι η τυποποίηση και τη συνέπειά του στις διάφορες εφαρμογές. Η SQL για πρώτη φορά τυποποιήθηκε από το ANSI (American Standards Institute) το 1986, και περαιτέρω επικυρώθηκε το 1987 από τον Διεθνή Οργανισμό Τυποποίησης (ISO), ο οποίος παραμένει ο οργανισμός τυποποίησης της.

Μειονεκτήματα:

- Αδυναμία Εφαρμογής Αναδρομικής Επεξεργασίας: Σύμφωνα με το "SQL for Dummies" ένα από τα σημαντικότερα μειονεκτήματα της SQL είναι η αδυναμία της να εκτελέσει αναδρομική επεξεργασία. Η αναδρομική επεξεργασία είναι ένα είδος λειτουργίας υπολογιστή (ή προγράμματος), στην οποία ένα από τα βήματα του προγράμματος (ή διαδικασία) επαναλαμβάνει ολόκληρο το πρόγραμμα (ή διαδικασία). Η SQL στερείται looping δομές που είναι κοινές σε άλλες υψηλού επιπέδου γλώσσες προγραμματισμού, όπως τους «for» και «what» βρόχους. Δεν μπορεί να επαναλάβει τις δράσεις και δεν υπάρχει τρόπος να ορίσετε επαναλαμβανόμενες looping κατασκευές στην SQL.
- Ασυμβατότητα και Πολυπλοκότητα: Ένα από τα σημαντικότερα μειονεκτήματα της Oracle SQL είναι η ασυνέπεια και η ασυμβατότητα των δεδομένων στους τομείς του χρόνου και της ημερομηνίας σύνταξης, στην αλληλουχία συμβολοσειρών και ευαισθησία περίπτωσης. Η γλώσσα είναι πολύπλοκη, με μια προσέγγιση του στυλ «λέξη-κλειδί» με δομή παρόμοια με της COBOL (common business-oriented language), με λιγότερους συντακτικούς και τους γραμματικούς κανόνες
- Περιορισμένη λειτουργικότητα: Η SQL είναι τύπου domain ή γλώσσα ειδικού σκοπού, και η χρήση του περιορίζεται σε ένα συγκεκριμένο τομέα του προγράμματος. Οι δηλώσεις SQL λειτουργούν σε πίνακες και σύνολα δεδομένων, προσωπικές βάσεις δεδομένων και λογιστικά φύλλα. Η SQL είναι μια δηλωτική συγκεκριμένη γλώσσα τομέα που περιορίζεται στην πινακοποίηση δεδομένων.

2.2.3. PostgreSQL

Η PostgreSQL αποτελεί μια ανοιχτού κώδικα σχεσιακή βάση δεδομένων με πολλές δυνατότητες. Η ανάπτυξη της ήδη διαρκεί πάνω από 20 χρόνια και βασίζεται σε μια αποδεδειγμένα καλή αρχιτεκτονική η οποία έχει δημιουργήσει μια ισχυρή αντίληψη των χρηστών της γύρω από την αξιοπιστία, την ακεραιότητα δεδομένων και την ορθή λειτουργία. Η PostgreSQL τρέχει σε όλα τα βασικά λειτουργικά συστήματα, περιλαμβάνοντας Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), και Windows. Είναι ACID συμβατή (ACID compliant), έχει ολοκληρωμένη υποστήριξη για foreign keys, joins, views, triggers, και stored procedures (σε διάφορες γλώσσες προγραμματισμού). Συμπεριλαμβάνει τα περισσότερα SQL92 και SQL99 data types, συμπεριλαμβανομένων INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, και TIMESTAMP. επίσης υποστηρίζει αποθήκευση binary large objects, όπως εικόνες,

ήχοι ή video. Διαθέτει native programming interfaces για C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, κ.α. καθώς και τεκμηρίωση.

Η PostgreSQL υλοποιεί εξεζητημένα χαρακτηριστικά όπως Multi-Version Concurrency Control (MVCC), point in time recovery, tablespaces, asynchronous replication, nested transactions (savepoints), online/hot backups, a sophisticated query planner/optimizer, write ahead logging for fault tolerance. Υποστηρίζει διεθνή σετ χαρακτήρων, κωδικοποίηση χαρακτήρων σε πολλά byte, Unicode καθώς και δυνατότητα ταξινόμησης δεδομένων ανεξάρτητα από το locale. Η PostgreSQL μπορεί να διαχειριστεί εύκολα μεγάλους αριθμούς ταυτόχρονων χρηστών καθώς και μεγάλο όγκο δεδομένων. Υπάρχουν ενεργές εγκαταστάσεις σε περιβάλλοντα παραγωγής που διαχειρίζονται πάνω από 4 terabytes δεδομένων.

Η PostgreSQL απολαμβάνει αναγνώριση από τους χρήστες της και την βιομηχανία πληροφορικής, συμπεριλαμβανομένων των Linux New Media Award for Best Database System, και έχει υπάρξει 3 φορές νικήτρια στο Linux Journal Editors' Choice Award for best DBMS.

Γενικά χαρακτηριστικά και συμβατότητα με προδιαγραφές

Η PostgreSQL είναι συνεπής με τις προδιαγραφές . Η υλοποίησή της είναι απολύτως σύμφωνη με τις προδιαγραφές ANSI-SQL 92/99. Έχει ολοκληρωμένη υποστήριξη για subqueries (συμπεριλαμβανομένων subselects μέσα από το FROM), read-committed και serializable transaction isolation levels. Η PostgreSQL αποτελεί ένα πλήρες σχεσιακό σύστημα που υποστηρίζει πολλαπλά σχήματα ανά database, ο κατάλογος (πληροφορίες σχετικά με τους πίνακες, στήλες, views, ...) είναι διαθέσιμος διαμέσου του Information Schema όπως ορίζεται στο SQL standard.

Στα Data integrity χαρακτηριστικά συμπεριλαμβάνονται: primary keys, foreign keys με υποστήριξη restricting και cascading updates/deletes, check constraints, unique constraints, και not null constraints.

Η PostgreSQL έχει αρκετά προηγμένα χαρακτηριστικά όπως: auto-increment columns μέσω sequences, LIMIT/OFFSET που επιτρέπουν την επιστροφή partial result sets. Όσον αφορά τα indexes υποστηρίζει compound, unique, partial, και functional indexes τα οποία μπορούν να χρησιμοποιήσουν οποιονδήποτε από τους B-tree, R-tree, hash, ή GiST αλγόριθμους.

GiST (Generalized Search Tree) indexing αποτελεί ένα προηγμένο σύστημα το οποίο συνδυάζει ένα μεγάλο εύρος από διαφορετικούς αλγόριθμους ταξινόμησης και αναζήτησης όπως B-tree, B+-tree, R-tree, partial sum trees, ranked B+-trees και αρκετούς ακόμα. Επίσης διαθέτει interface το οποίο επιτρέπει τόσο την δημιουργία custom data types όσο και επεκτάσιμους τρόπους ερωτήσεων (query) για την αναζήτησή τους. το GiST προσφέρει την ευελιξία στον προσδιορισμό του τι και με πιο τρόπο να το αποθηκεύσεις, και την δυνατότητα να ορίσεις νέους τρόπους αναζήτησης.

Πάνω στο GiST και την PostgreSQL έχουν θεμελιωθεί πολλά projects όπως τα OpenFTS και PostGIS. Το OpenFTS (Open Source Full Text Search engine) παρέχει online indexing και relevance ranking για αναζήτηση. Το PostGIS αποτελεί ένα project το οποίο προσθέτει υποστήριξη για geographic objects στην PostgreSQL, επιτρέποντας την χρήση της σαν spatial database για geographic information systems (GIS), περίπου όπως τα ESRI's SDE ή τα Oracle's Spatial extension.

Αλλα προηγμένα χαρακτηριστικά της PostgreSQL είναι: table inheritance, rules systems και database events. Το Table inheritance (κληρονομικότητα πινάκων) πρόσθετη μια αντικειμενοστραφής διάσταση στην δημιουργία πινάκων, επιτρέποντας στους σχεδιαστές database να δημιουργούν νέους πίνακες από άλλους πίνακες χρησιμοποιώντας τους ως βάση. Ακόμα καλύτερα Η PostgreSQL υποστηρίζει και μόνη και πολλαπλή κληρονομικότητα με τον δικό της τρόπο.

Το rules system, επίσης καλείται the query rewrite system, επιτρέπει στον σχεδιαστή βάσεων να δημιουργήσει κανόνες που ορίζουν συγκεκριμένες λειτουργίες για έναν πίνακα ή view, και να μετατρέψει δυναμικά την ώρα που εκτελούνται λειτουργίες σε άλλες εναλλακτικές.

Το events system αποτελεί ένα interprocess communication system στο οποίο μηνύματα και events μπορούν να μεταδοθούν μεταξύ πελατών (clients) χρησιμοποιώντας τις LISTEN και NOTIFY εντολές, επιτρέποντας από την απλή peer to peer επικοινωνία ως ένα εξελιγμένο συντονισμό βασισμένο σε database events. Εφόσον τα notifications μπορεί να προέρχονται από triggers και stored procedures, PostgreSQL clients μπορούν να επιβλέπουν λειτουργίες όπως: updates, inserts ή deletes πινάκων όταν αυτά γίνονται.

Η PostgreSQL τρέχει stored procedures σε πολλές γλώσσες προγραμματισμού συμπεριλαμβανομένων Java, Perl, Python, Ruby, Tcl, C/C++, και της PL/pgSQL η οποία είναι παρόμοια με την PL/SQL της Oracle. Στην βασική βιβλιοθήκη συναρτήσεων της PostgreSQL συμπεριλαμβάνονται εκατοντάδες built-in συναρτήσεις οι οποίες καλύπτουν απο βασικές μαθηματικές συναρτήσεις και διαχείριση Συμβολοσειρών ως κρυπτογραφία και Oracle compatibility. Triggers και stored procedures μπορούν να γράφουν σε c και να φορτωθούν μέσα στην database ως βιβλιοθήκη, επιτρέποντας μεγάλη ευελιξία στην επέκταση των δυνατοτήτων της βάσης. Παρομοίως η PostgreSQL περιλαμβάνει framework που επιτρέπει τον ορισμό και την δημιουργία custom data types καθώς και βοηθητικές συναρτήσεις και τελεστές (operators) που θα περιγράψουν την λειτουργία τους. Σαν αποτέλεσμα ένα πλήθος από εξελιγμένα data types έχουν δημιουργηθεί από γεωμετρικά και spatial δεδομένα ως διεύθυνσης δικτύων και ISBN/ISSN (International Standard Book Number/International Standard Serial Number) data types, τα οποία μπορούν κατ επιλογή να προστεθούν στο σύστημα.

Η PostgreSQL όπως διαθέτει πολλές procedure languages έτσι διαθέτει και πολλά library interfaces , επιτρέποντας πολλές γλώσσες προγραμματισμού είτε compiled είτε interpreted να επικοινωνούν με την PostgreSQL. Υπάρχουν interfaces για Java (JDBC), ODBC, Perl, Python, Ruby, C, C++, PHP, Lisp, Scheme, Qt, κ.α.

Τέλος ο πηγαίος κώδικας της PostgreSQL είναι διαθέσιμος κάτω από την πιο ελεύθερη open source άδεια: το BSD license. Αυτή η άδεια δίνει την δυνατότητα χρήσης, μετατροπής και διανομής της PostgreSQL σε οποιαδήποτε μορφή, ανοιχτού ή κλειστού κώδικα. Η PostgreSQL δεν είναι μόνο μια δυνατή βάση δεδομένων ικανή να τρέχει μέσα σε επιχειρήσεις, είναι μια πλατφόρμα ανάπτυξης πάνω στην οποία δύναται να γίνει ανάπτυξη in-house, web ή εμπορικών εφαρμογών τα οποία χρειάζονται RDBMS.

Πλεονεκτήματα:

Η PostgreSQL προσφέρει πολλά πλεονεκτήματα για την εταιρεία ή την επιχείρησή σας σε σχέση με άλλα συστήματα βάσεων δεδομένων:

- Ανοσία σε υπερ-ανάπτυξη:

Η υπερ-ανάπτυξη είναι αυτό που μερικοί προμηθευτές ιδιόκτητου λογισμικού βάσεων δεδομένων αναφέρουν ως το # 1 πρόβλημα συμμόρφωσης της άδειας. Με την PostgreSQL, κανείς δεν μπορεί να σας μηνύσει για το σπάσιμο των συμφωνιών αδειών εκμετάλλευσης, καθώς δεν υπάρχει σχετικό κόστος αδειοδότησης για το λογισμικό.

Αυτό έχει πολλά πρόσθετα πλεονεκτήματα:

- Περισσότερα κερδοφόρα επιχειρηματικά μοντέλα με ευρείας κλίμακας ανάπτυξη.
- Δεν υπάρχει δυνατότητα να ελεγχθούν ως προς τη συμμόρφωση άδειας σε οποιοδήποτε στάδιο.
- Η ευελιξία να κάνει έρευνά και υλοποιήσεις χωρίς να χρειάζεται να συμπεριλάβει πρόσθετες δαπάνες αδειοδότησης.

- Καλύτερη υποστήριξη από τους ιδιόκτητους πωλητές:

Εκτός από τις ισχυρές προσφορές υποστήριξης μας, έχουμε μια ζωντανή κοινότητα απο επαγγελματίες της PostgreSQL και οπαδούς που το προσωπικό σας μπορεί να βασιστεί και να συμβάλει.

- Σημαντική εξοικονόμηση στο κόστος προσωπικού:

Το λογισμικό μας έχει σχεδιαστεί και δημιουργηθεί για να έχει πολύ χαμηλότερες απαιτήσεις συντήρησης και ρύθμισης από τις μεγαλύτερες ιδιόκτητες βάσεις δεδομένων, αλλά εξακολουθεί να διατηρεί όλα τα χαρακτηριστικά, τη σταθερότητα και τις επιδόσεις. Εκτός από αυτό, τα προγράμματα κατάρτισης μας είναι γενικά γνωστά ως πολύ πιο αποδοτικό οικονομικά, εύχρηστο, πρακτικό και στον πραγματικό κόσμο από τους κορυφαίους προμηθευτές ιδιόκτητου λογισμικού βάσεων δεδομένων.

- Ο θρυλικός αξιοπιστία και τη σταθερότητα:

Σε αντίθεση με πολλές αποκλειστικές βάσεις δεδομένων, είναι εξαιρετικά κοινό για τις εταιρείες να αναφέρουν ότι η PostgreSQL δεν έχει ποτέ δημιουργήσει προβλήματα για αρκετά χρόνια από τη λειτουργία υψηλής δραστηριότητας. Ούτε καν μια φορά. Απλά λειτουργεί.

- Επεκτάσιμο:
Ο πηγαίος κώδικας είναι διαθέσιμος σε όλους χωρίς καμία επιβάρυνση. Εάν το προσωπικό σας να έχει ανάγκη να προσαρμόσει ή να επεκτείνει την PostgreSQL με οποιονδήποτε τρόπο, τότε είναι σε θέση να το πράξει με ελάχιστη προσπάθεια, και χωρίς να επισυνάπτεται το κόστος. Αυτό συμπληρώνεται από την κοινότητα των επαγγελματιών PostgreSQL και οπαδών σε όλο τον κόσμο που χρησιμοποιούν ενεργά την PostgreSQL σε καθημερινή βάση.
- Cross πλατφόρμα:
Η PostgreSQL είναι διαθέσιμη για σχεδόν κάθε είδος Unix (34 πλατφόρμες με την τελευταία σταθερή έκδοση), και για τα Windows είναι διαθέσιμη μέσω του πλαισίου Cygwin. Για τα Windows είναι επίσης διαθέσιμη με την έκδοση 8.0 και άνω.
- Σχεδιασμένος για περιβάλλοντα υψηλού όγκου:
Χρησιμοποιεί μια πολλαπλή στρατηγική για την αποθήκευση δεδομένων, η οποία ονομάζεται MVCC, για να κάνει την PostgreSQL πολύ καλή σε απόκριση σε περιβάλλοντα υψηλού όγκου. Ο κορυφαίος προμηθευτής ιδιόκτητου λογισμικού βάσεων δεδομένων χρησιμοποιεί αυτή την τεχνολογία για τους ίδιους λόγους.
- GUI σχεδίαση βάσης δεδομένων και εργαλεία διαχείρισης:
Υπάρχουν πολλά υψηλής ποιότητας εργαλεία GUI για την PostgreSQL από τους προγραμματιστές ανοιχτού κώδικα της και τους φορείς του εμπορίου.

Μειονεκτήματα:

- Δεν υπάρχουν προεπιλεγμένες παραμέτροι στο postgresql
- Καμία υποστήριξη ενσωματωμένη στην postgresql για τη δημιουργία web-based διαδικασίες (όπως στην Oracle)
- Προβλήματα συμβατότητας με τα Windows

2.3. Επιλογή τεχνολογίας

Με βάση τα παραπάνω καταλήξαμε στο συμπέρασμα ότι οι ιδανικότερες τεχνολογίες για την υλοποίηση της πτυχιακής είναι η HTML, η PHP, η JavaScript και την σχεσιακή βάση MySQL. Το σημαντικότερο πλεονέκτημα των παραπάνω είναι ότι είναι η ευκολότερη κατανόηση και η ανάπτυξη κώδικα με την χρήση τους καθώς και η μεγάλη κοινότητα υποστηρίξις τους που υπάρχει στο διαδίκτυο με την οποία μπορούμε να αντλήσουμε πληροφορίες για την επίλυση των προβλημάτων.

Ειδικότερα, η HTML μας δίνει την δυνατότητα καλύτερης δημιουργίας και προβολής των περιεχομένων της σελίδας μας. Η PHP χρησιμοποιήθηκε κυρίως για την διασύνδεση της δυναμικής ιστοσελίδας μας με την βάση δεδομένων και την διαχείριση των περιεχομένων αυτής. Η JavaScript μας έδωσε την δυνατότητα να υλοποιήσουμε πρόσθετα χαρακτηριστικά όπως μηνύματα στον χρήστη μέσω MessageBox και την ενσωμάτωση συμπληρωματικού περιεχομένου στην σελίδα μας.

Η MySQL επιλέχτηκε διότι είναι ιδιαίτερα γρήγορη στην ανάκτηση δεδομένων και επίσης είναι ανοιχτού κώδικα. Η MySQL επίσης διαθέτει μια σειρά εύχρηστων εργαλείων για την ευκολότερη διαχείρισή της.

3. ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

3.1. Εισαγωγικά

3.1.1. Βασικές Έννοιες

- **Τι είναι η Βάση Δεδομένων (ΒΔ) ή Database (DB);**

Ως βάση δεδομένων ορίζουμε μια συλλογή από συστηματικά οργανωμένα (formatted) και σχετιζόμενα μεταξύ τους δεδομένα. Ένας τηλεφωνικός κατάλογος, για παράδειγμα, θεωρείται βάση δεδομένων, καθώς αποθηκεύει και οργανώνει σχετιζόμενα τμήματα πληροφορίας, όπως είναι το όνομα και ο αριθμός τηλεφώνου. Ωστόσο, στον κόσμο των υπολογιστών, με τον όρο βάση δεδομένων αναφερόμαστε σε μια συλλογή σχετιζόμενων δεδομένων - τμημάτων πληροφορίας – ηλεκτρονικά αποθηκευμένων.

- **Τι είναι το Σύστημα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ) ή Database Management System (DBMS);**

ΣΔΒΔ καλείται ένα πρόγραμμα λογισμικού υπολογιστών που σχεδιάζεται για να διαχειρίζονται όλες οι βάσεις δεδομένων που υπάρχουν σε σκληρό δίσκο ή ένα δίκτυο συστημάτων. Εξάλλου, μπορεί να οριστεί ως ένα λογισμικό όπου μέσω προγράμματα υπολογιστών που ελέγχουν τη δημιουργία, τη συντήρηση, και τη χρήση μιας βάσης δεδομένων. Επιτρέπει σε οργανισμούς την να ανάπτυξη εύχρηστων βάσεων δεδομένων για τις διάφορες εφαρμογές από τους διαχειριστές των.

- **Τι είναι το Σύστημα Βάσης Δεδομένων (ΣΒΔ) ή Database System(DBS);**

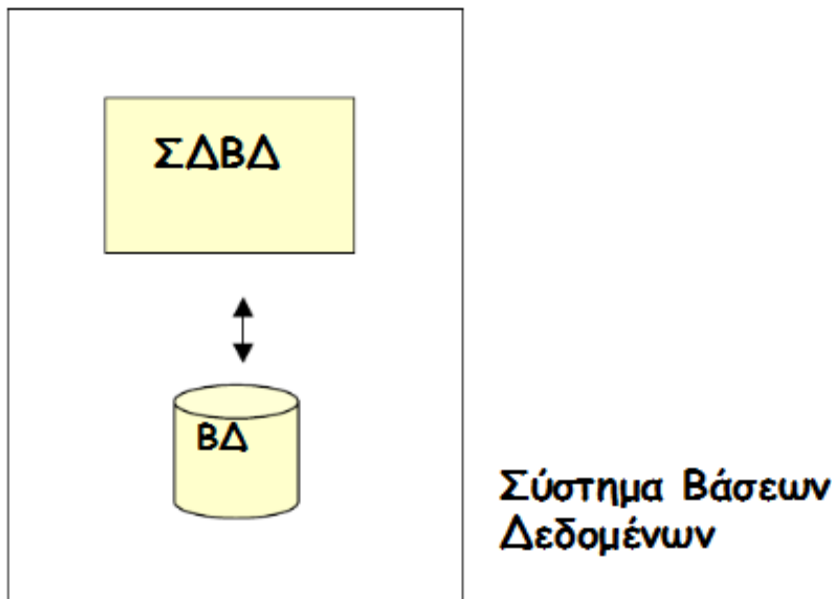
ΣΒΔ καλείται ο συνδυασμός και η σύνδεση της βάσης με το λογισμικό.

- **Τι είναι ο DataBase Administrator (DBA) ή διαχειριστής μιας Βάσης δεδομένων (ΔΒΔ) και ποιος ο ρόλος του;**

Ο διαχειριστής μιας Βάσης δεδομένων (DBA, DataBase Administrator) είναι αυτός που έχει την ευθύνη για τον σωστό, αποδοτικό και αξιόπιστο τρόπο δημιουργίας και λειτουργίας μια βάσης δεδομένων. Οι αρμοδιότητές του είναι οι εξής :

- Η απόφαση για το είδος των πληροφοριών που πρέπει να αποθηκευτούν.
- Η απόφαση για τον τρόπο αποθήκευσης και πρόσβασης στις πληροφορίες αυτές.
- Η συνεργασία με τους τελικούς χρήστες.
- Η απόφαση για τον τρόπο εξασφάλισης των πληροφοριών.
- Η απόφαση για το κάθε πότε θα γίνονται αντίγραφα ασφαλείας (backup) των αρχείων.
- Η παρακολούθηση της σωστής λειτουργίας της βάσης δεδομένων και η απαιτούμενη προσαρμογή της.

Εικόνα 1



3.1.2. Ιστορική Εξέλιξη των Βάσεων Δεδομένων:

- Δεκαετία του 1950

Κάρτες και ταινίες(σειριακή επεξεργασία): Εξελίξεις σε συσκευές μαζικής αποθήκευσης τυχαίας πρόσβασης και αύξηση υπολογιστικής ισχύος θέτουν τις προϋποθέσεις για την ανάπτυξη συστημάτων διαχείρισης δεδομένων σε αντικατάσταση των συστημάτων διαχείρισης αρχείων.

- Αρχή του 1960

Πρώτες βάσεις δεδομένων, πρώτα συστήματα διαχείρισης βάσεων δεδομένων (ΣΔΒΔ), στηριγμένα στο δικτυωτό μοντέλο -network data model- του Charles Bachman (1st Turing Award, 1973)

- Τέλη του 1960

Βάσεις δεδομένων και συστήματα διαχείρισης βάσεων δεδομένων (ΣΔΒΔ), στηριγμένα στο ιεραρχικό μοντέλο δεδομένων - Information Management System (IMS της IBM)-1970 Ορισμός του σχεσιακού μοντέλου από τον Codd της IBM. Edgar Codd (IBM, San Jose) relational data model (Turing Award, 1981) Ερευνητικά Προγράμματα : System R, INGRES-. Γλώσσες: SEQUEL, QBE, QUEL

- Αρχές του 1980

SQL (μέρος του System R) αντικατέστησε την QUEL στο σύστημα INGRES. Η έννοια του transaction management (Jim Gray, Turing Award, 1999) .Τάσεις: αντικειμενοστραφή, αρχιτεκτονική πελάτη – εξυπηρετή, κατακεντρωμένες βάσεις δεδομένων.

- Δεκαετία του 1980

ΒΔ σε προσωπικούς υπολογιστές, σχεσιακά συστήματα σε pc. : DBase II (Ashton Tate) εξελίχθηκε μέχρι τις μέρες μας ως Paradox, Microsoft ACCESS. Σχεσιακά συστήματα με μεγάλη διάδοση : Oracle, SQL Server, Sybase, Informix, DB2.

- Δεκαετία του 1990 μέχρι σήμερα

αντικειμενοστραφή συστήματα ΒΔ σύνδεση ΒΔ στο διαδίκτυο, ευρύτατη διάδοση τεχνολογίας που επιτρέπει την επικοινωνία των χρηστών με ΒΔ μέσω διαδικτύου (HTML, XML, ASP κλπ).

3.1.3. Αρχιτεκτονική τριών επιπέδων:

Ένα σύστημα βάσεων δεδομένων παρέχει στους χρήστες μια αφαιρετική άποψη. Το σύστημα κρύβει (από τους χρήστες) ορισμένες λεπτομέρειες για το πώς τα δεδομένα αποθηκεύονται και διατηρούνται. Γι' αυτό υπάρχουν τρία επίπεδα αφαίρεσης:

1. Φυσικό επίπεδο:

Είναι το χαμηλότερο επίπεδο αφαίρεσης και περιγράφει πώς αποθηκεύονται τα δεδομένα.

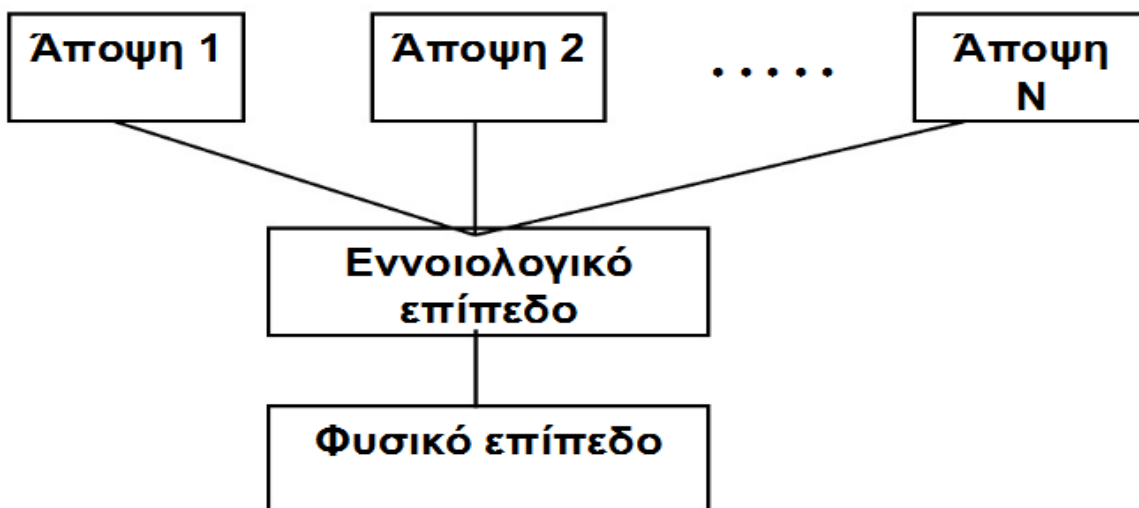
2. Εννοιολογικό επίπεδο:

Το αμέσως υψηλότερο επίπεδο αφαίρεσης. Περιγράφει τι είδους δεδομένα αποθηκεύονται και τις συσχετίσεις μεταξύ αυτών των δεδομένων.

3. Επίπεδο άποψης:

Το πιο υψηλό επίπεδο και περιγράφει τμήματα της βάσης δεδομένων για κάθε συγκεκριμένη ομάδα χρηστών, μπορεί να υπάρχουν πολλές διαφορετικές απόψεις, π.χ. Οι ταμίες σε μία τράπεζα παίρνουν μία άποψη των απολογισμών πελατών, αλλά όχι των δεδομένων μισθοδοτικών καταστάσεων.

Εικόνα 2



3.1.4. Μορφές Βάσεων Δεδομένων:

• **Οι Αντικειμενοστραφείς Βάσεις Δεδομένων:**

Μια βάση δεδομένων αντικειμένου (Object database) (επίσης αναφέρεται και ως αντικειμενοστραφές σύστημα διαχείρισης βάσεων δεδομένων) είναι ένα σύστημα διαχείρισης βάσεων δεδομένων στο οποίο οι πληροφορίες αναπαρίστανται με τη μορφή αντικειμένων όπως χρησιμοποιούνται στον αντικειμενοστραφή προγραμματισμό. Οι βάσεις δεδομένων αντικειμένου είναι ένας τομέας μέσα στην ευρύτερη αγορά συστημάτων διαχείρισης βάσεων δεδομένων (DBMS) όπου κυριαρχούν τα μοντέλα σχεσιακής διαχείρισης βάσης δεδομένων.

• **Οι Ημι - δομημένες Βάσεις Δεδομένων:**

Στο ημι-δομημένο μοντέλο (semi-structured) δεν υπάρχει κανένας διαχωρισμός μεταξύ των δεδομένων και του σχήματος, και το ποσό δομής χρησιμοποιούμενο αλλά εξαρτάται από το εκάστοτε σκοπό.

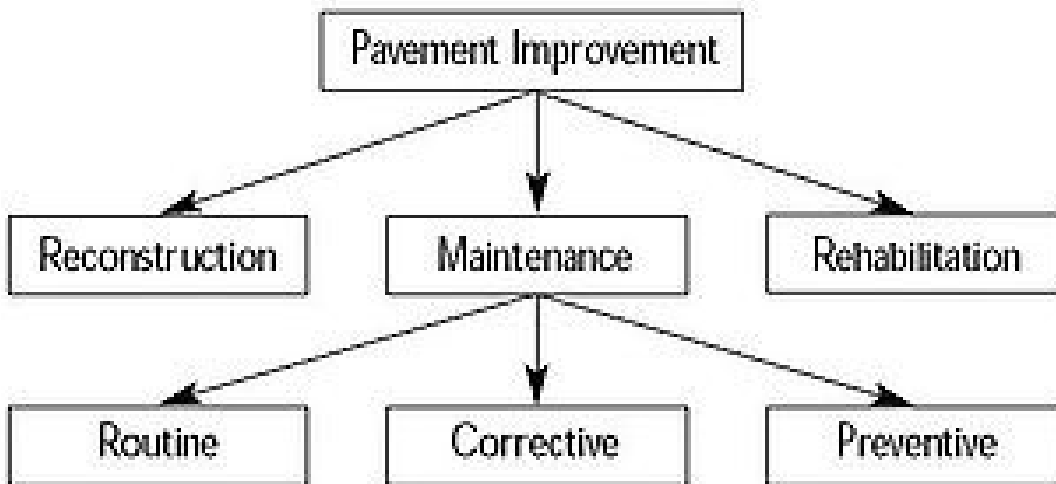
• **Οι Ιεραρχικές Βάσεις Δεδομένων:**

Στις Ιεραρχικές (Hierarchical) βάσεις δεδομένων τα δεδομένα αναπαρίστανται με δενδρικής μορφής δομές δεδομένων και συνδέονται μεταξύ τους με συνδέσμους (links). Η κάθε εγγραφή μπορεί να

συνδέεται προς τα πάνω μόνο με μία άλλη εγγραφή (γονέας), ενώ μπορεί να έχει έως δύο εγγραφές που να εξαρτώνται απ' αυτήν (παιδιά). Υπάρχει μία μόνο εγγραφή ρίζα (root), απ' την οποία εξαρτώνται όλες οι άλλες εγγραφές της βάσης δεδομένων. Έχουν το μειονέκτημα ότι είναι πολύπλοκες στην επεξεργασία των εγγραφών τους (προσθήκη, διαγραφή, τροποποίηση).

Εικόνα 3

Hierarchical Model

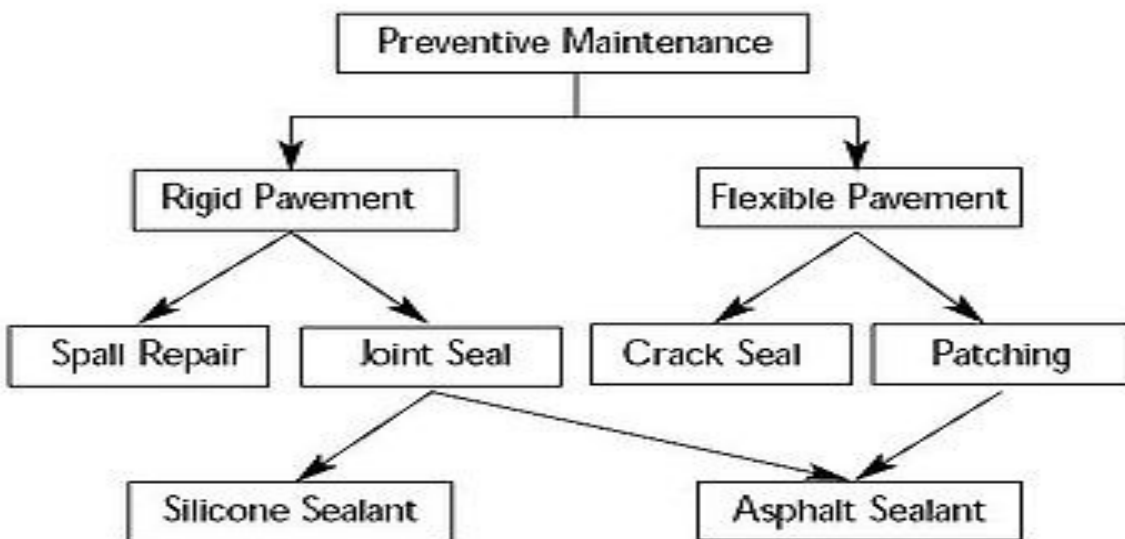


- **Οι Δικτυωτές Βάσεις Δεδομένων:**

Στις Δικτυωτές (Network) βάσεις δεδομένων τα δεδομένα αναπαρίστανται με δενδρικής μορφής δομές δεδομένων και συνδέονται μεταξύ τους με συνδέσμους (links), όπως ακριβώς και στις ιεραρχικές βάσεις δεδομένων, με την διαφορά ότι μια εγγραφή μπορεί να συνδέεται προς τα πάνω με περισσότερες από μία πατρικές εγγραφές (parent records). Είναι λογικά πιο δύσχρηστες αλλά και πιο γρήγορες από τις ιεραρχικές βάσεις δεδομένων.

Εικόνα 4

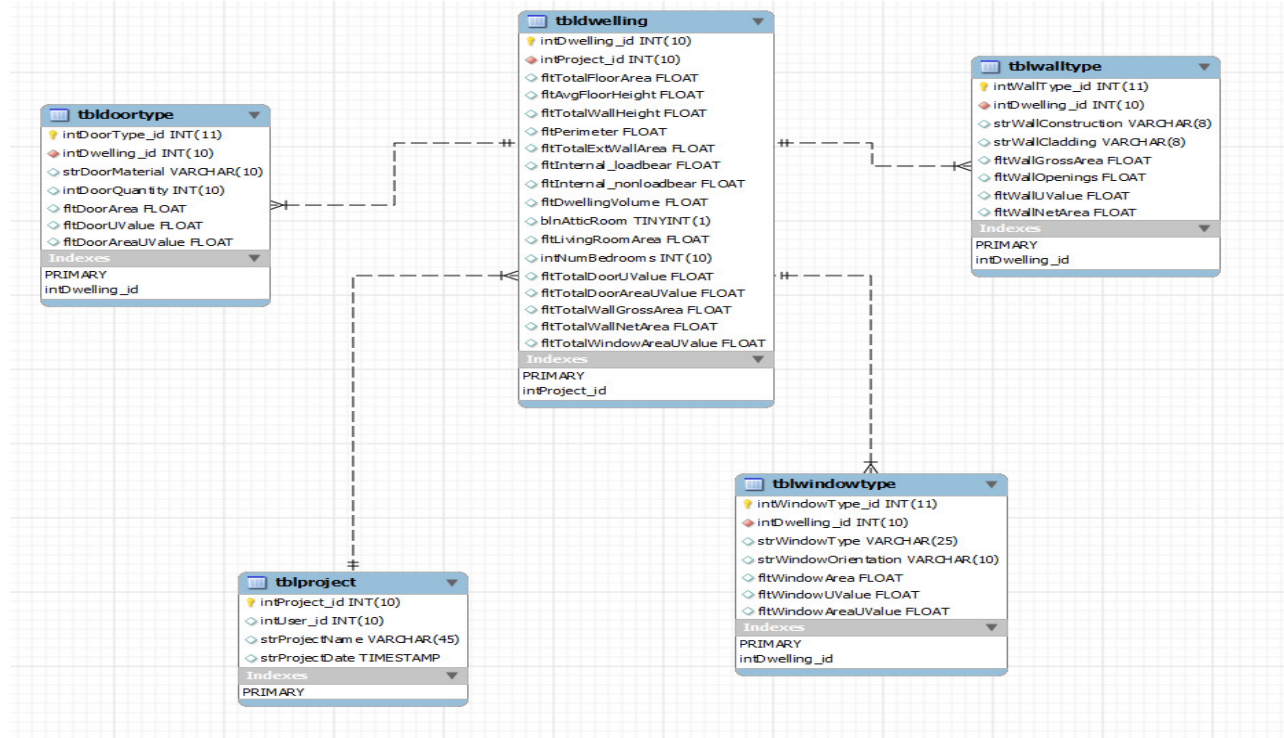
Network Model



- **Οι Σχεσιακές Βάσεις Δεδομένων:**

Στις Σχεσιακές (Relational) βάσεις δεδομένων, τα δεδομένα συνδέονται μεταξύ τους με σχέσεις (relations), οι οποίες προκύπτουν από τα κοινά πεδία που υπάρχουν σε διαφορετικά αρχεία. Τα αρχεία αποκαλούνται πίνακες (tables), οι εγγραφές γραμμές (rows) και τα πεδία στήλες (columns). Η ύπαρξη μιας κοινής τιμής στα πεδία δύο αρχείων καθορίζει και μια σχέση μεταξύ των γραμμών διαφορετικών πινάκων. Οι σχεσιακές βάσεις δεδομένων έχουν το πλεονέκτημα ότι είναι λογικά κατανοητές και πολύ ευέλικτες και δεκτικές σε αλλαγές.

Εικόνα 5



3.1.5. Στοιχεία Βάσεων Δεδομένων:

Τα δεδομένα μιας βάσης δεδομένων αποθηκεύονται (και οργανώνονται) στις εξής στοιχειώδεις μορφές:

- **Πεδίο (Field)**, είναι το μικρότερο κομμάτι δεδομένων στο οποίο μπορούμε να αναφερθούμε και περιέχει ένα μόνο χαρακτηριστικό ή ιδιότητα ενός στοιχείου της βάσης δεδομένων.
- **Εγγραφή (Record)**, είναι ένα σύνολο από διαφορετικά πεδία που περιέχει όλες τις πληροφορίες για ένα στοιχείο της βάσης δεδομένων.
- **Αρχείο (File)**, είναι ένα σύνολο από πολλά παρόμοια στοιχεία (εγγραφές) της βάσης δεδομένων.
- **Πρωτεύον Κλειδί (Primary Key)**, είναι ένα πεδίο ή συνδυασμός πεδίων που χαρακτηρίζει μοναδικά μια εγγραφή. Αρχικό κλειδί ενός σχεσιακού πίνακα προσδιορίζει μεμονωμένα κάθε εγγραφή του πίνακα. Μπορεί είτε να είναι μια κανονική ιδιότητα που είναι εγγυημένη για να είναι μοναδική (όπως ο αριθμός κοινωνικής ασφάλισης σε έναν πίνακα χωρίς περισσότερα από ένα αρχεία ανά άτομο) ή μπορεί να παραχθεί από το DBMS (σύστημα διαχείρισης βάσεων

δεδομένων). Τα αρχικά κλειδιά μπορούν να αποτελούν τόσο απλές ιδιότητες όσο και πολλαπλές σε συνδυασμό.

- **Ξένο Κλειδί (Foreign Key)**, είναι ένα πεδίο που έχει το ίδιο σύνολο τιμών με το πρωτεύον κλειδί ενός άλλου αρχείου. Αναλυτικότερα ξένο κλειδί είναι μια ιδιότητα σε έναν σχεσιακό πίνακα που ταιριάζει με ένα υποψήφιο κλειδί ενός άλλου πίνακα. Το ξένο κλειδί προσδιορίζει μια στήλη ή ένα σύνολο στηλών σε έναν πίνακα που αναφέρεται σε μια στήλη ή ένα σύνολο στηλών σε έναν άλλο πίνακα.

3.1.6. Συνηθισμένα προβλήματα ή δυσκολίες που συναντώνται στις ΒΔ:

1. Ασυνέπειας και πλεονασμού δεδομένων

Ο πλεονασμός στοιχείων εμφανίζεται στα συστήματα βάσεων δεδομένων που έχουν έναν πεδίο που επαναλαμβάνεται σε δύο ή περισσότερους πίνακες. Για παράδειγμα, η περίπτωση που πελάτης έχει καταχωρηθεί δύο φορές και συνδέεται με κάθε προϊόν που αγοράζεται επιφέρει πλεονασμός στα δεδομένων, γνωστό ως πηγή ασυνέπειας, δεδομένου ότι ο πελάτης να εμφανιστεί με τις διαφορετικές τιμές για τις δεδομένες ιδιότητες.

2. Ακεραιότητα δεδομένων

Οι περιορισμοί ακεραιότητας χρησιμοποιούνται για να εξασφαλίσουν την ακρίβεια και τη συνέπεια των στοιχείων σε μια σχεσιακή βάση δεδομένων. Η ακεραιότητα στοιχείων αντιμετωπίζεται σε μια σχεσιακή βάση δεδομένων μέσω της έννοιας της αναφερόμενης ακεραιότητας. Υπάρχουν πολλοί τύποι περιορισμών ακεραιότητας που διαδραματίζουν έναν ρόλο στην αναφερόμενη ακεραιότητα. Χωρίζονται στις εξής κατηγορίες σύμφωνα με τον Codd.

2.1. Ακεραιότητα οντοτήτων (Entity integrity)

Ο περιορισμός ακεραιότητας οντοτήτων δηλώνει ότι καμία τιμή βασικού κλειδιού δεν μπορεί να είναι μηδενική. Αυτό είναι επειδή η τιμή βασικού κλειδιού χρησιμοποιείται για να προσδιορίσει μεμονωμένα πλειάδες σε μια σχέση. Η κατοχή της μηδενικής αξίας για το αρχικό κλειδί υπονοεί ότι δεν μπορούμε να προσδιορίσουμε σίγουρα κάποιες πλειάδες. Αυτό επίσης εξασφαλίζει ότι δεν μπορούν να υπάρξουν οποιεσδήποτε διπλές καταχωρήσεις στην αρχική βασική στήλη (column).

2.2. Αναφερόμενη ακεραιότητα (Referential Integrity)

Ο αναφερόμενος περιορισμός ακεραιότητας αφορά δύο συσχετίσεις και χρησιμοποιείται για να διατηρηθεί η συνέπεια μεταξύ πλειάδων στις δύο σχέσεις. Πιο απλά, ο αναφερόμενος περιορισμός ακεραιότητας δηλώνει ότι πλειάδα σε μια συσχέτιση που αναφέρεται σε μια άλλη συσχέτιση πρέπει να αναφερθεί σε (reference to) πλειάδα σε αυτή τη άλλη συγκεκριμένη συσχέτιση.

2.3. Ακεραιότητα περιοχών (Domain Integrity)

Η ακεραιότητα περιοχών δηλώνει ότι κάθε στοιχείο από μια συσχέτιση πρέπει να ανταποκρίνεται στον τύπο και τους περιορισμούς των αντίστοιχων ιδιοτήτων του. Οι περιορισμοί θα μπορούσαν να είναι η πεδίο τιμών που το στοιχείο μπορεί να έχει, η προκαθορισμένη τιμή εάν καμία δεν παρέχεται, τότε η τιμή του στοιχείο μπορεί να είναι μηδενική (NULL).

2.4. Καθορισμένη από το χρήστη ακεραιότητα (User Defined Integrity)

π.χ.: Ηλικία >= 18 && Ηλικία <= 60

3. Ασφαλείας

Οι διαρροές στα συστήματα βάσεων δεδομένων περιλαμβάνουν, παραδείγματος χάριν:

- Αναρμόδια ή απρομελέτητη δραστηριότητα ή κακή χρήση από τους εξουσιοδοτημένους χρήστες βάσεων δεδομένων ή και hacking.
- Μολύνσεις Malware που προκαλούν τα γεγονότα όπως η αναρμόδια πρόσβαση και διαρροή.
- Υπερφορτώσεις, περιορισμοί απόδοσης με συνέπεια την ανικανότητα των εξουσιοδοτημένων χρηστών να χρησιμοποιήσουν τις βάσεις δεδομένων όπως θα έπρεπε .
- Φυσική ζημία στους κεντρικούς υπολογιστές βάσεων δεδομένων.
- Σχεδιαστικά σφάλματα και bugs κατά των προγραμματισμό των ΒΔ

- Απώλεια δεδομένων που προκαλούνται από την είσοδο των άκυρων καταχωρήσεων ή λανθασμένων εντολών, π.χ. τα λάθη στις διαδικασίες διαχείρισης βάσεων δεδομένων κ.λπ.

4. **Ανεξαρτησίας δεδομένων (δηλαδή πολλαπλά αρχεία και formats)**

Η ανεξαρτησία στοιχείων είναι μια μορφή διαχείρισης βάσεων δεδομένων που κρατά τα στοιχεία χωρισμένα από όλα τα προγράμματα που χρησιμοποιούν τα στοιχεία. Σαν ακρογωνιαίο λίθο για την ιδέα ενός ΣΔΒΔ (Σύστημα Διαχείρισης Βάσεων Δεδομένων) η ανεξαρτησία δεδομένων εξασφαλίζει ότι τα στοιχεία δεν μπορούν να επαναπροσδιοριστούν ή να αναδιοργανωθούν από οποιαδήποτε από τα προγράμματα τα οποία τα χρησιμοποιούν. Κατά αυτόν τον τρόπο, το στοιχείο παραμένει προσπελάσιμο, αλλά είναι επίσης σταθερό και δεν μπορεί να αλλοιωθεί από τις εφαρμογές χρησιμοποιώντας το.

5. **Προσπέλασης δεδομένων**

6. **Ενημέρωσης πολλών αρχείων**

Συνεπώς, στόχοι ώστε να επιτύχουμε μια ορθα σχεδιασμένη βάση δεδομένων πρέπει να είναι οι εξής:

1. Ο περιορισμός της πολλαπλής αποθήκευσης των ίδιων στοιχείων (redundancy).
2. Ο καταμερισμός (sharing) των ίδιων στοιχείων σ' όλους τους χρήστες.
3. Η ομοιομορφία (uniformity) στον χειρισμό και την αναπαράσταση των δεδομένων.
4. Η επιβολή κανόνων ασφαλείας (security).
5. Η διατήρηση της ακεραιότητας (integrity) και της αξιοπιστίας (reliability) των δεδομένων.
6. Η ανεξαρτησία των δεδομένων (data independence) και των προγραμμάτων από τον φυσικό τρόπο αποθήκευσης των δεδομένων.

Και τα κατά σειρά **βήματα σχεδιασμού της:**

Ο σχεδιασμός για τη δημιουργία μίας βάσης δεδομένων περνάει από τρία βασικά στάδια , Αρχικά :

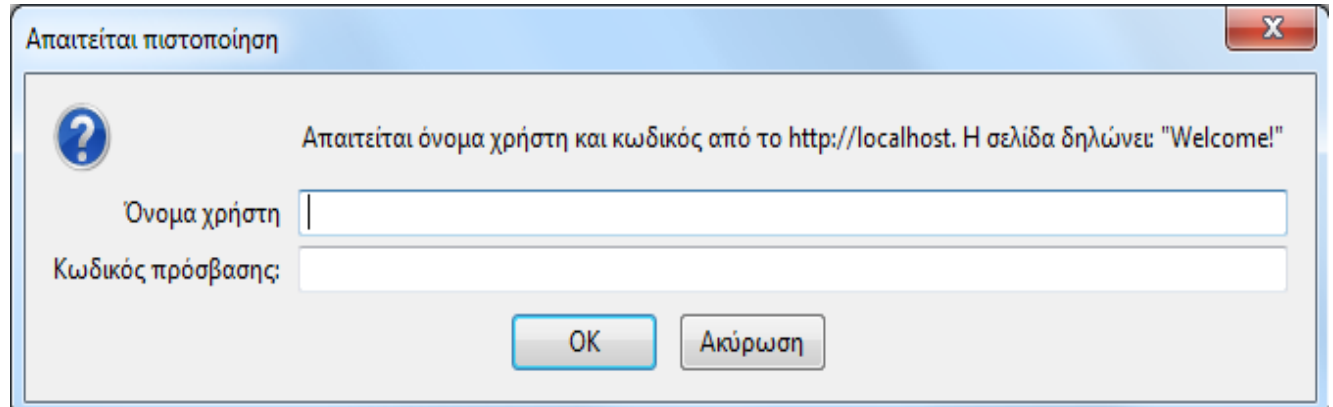
1. **Ανάλυση των Απαιτήσεων:**
Δηλαδή τι τύπου δεδομένα θέλουμε να αποθηκευτούν, ποιες διεργασίες θα πρέπει να γίνονται πάνω σε αυτά ή συχνότητα λειτουργιών που χρησιμοποιούνται. Κατόπιν,
2. **Εννοιολογικός σχεδιασμός:**
Ακριβής και υψηλού επιπέδου περιγραφή των δεδομένων που πρόκειται να αποθηκεύονται στη ΒΔ, ταυτόχρονη εμφάνιση των περιορισμών και σχέσεων μεταξύ δεδομένων (χρήση e-r μοντέλου). Τελικά,
3. **Λογικός Σχεδιασμός:**
Επιλογή ενός εργαλείου σχεδίασης βάσεων δεδομένων για την υλοποίηση του σχεδιασμού, μετατροπή του εννοιολογικού σχεδιασμού σε ένα σχήμα στο μοντέλο δεδομένων του επιλεγμένου ΣΔΒΔ (σ.σ. ΣΔΒΔ : Σύστημα Διαχείρισης Βάσης Δεδομένων - Database Management System - είναι ένα πακέτο λογισμικού για τη συστηματική αποθήκευση -storage-, αναζήτηση -retrieval- και συντήρηση -maintenance- δεδομένων).

3.2. **Ανάλυση των Απαιτήσεων**

3.2.1. Περιγραφή Συστήματος

Το online πληροφοριακό σύστημα που κατασκευάστηκε υποστηρίζει τη διαχείριση - και μόνο - μίας αλυσίδας supermarket. Δηλαδή δεν απαιτούνται διαφορετικές “όψεις” (π.χ. Ο απλός χρήστης/αγοραστής σε ένα ηλεκτρονικό μαγαζί) αλλά προσφέρεται για τον απόλυτο έλεγχο της μεγάλης κάθετα ολοκληρωμένης επιχείρησης από την παραγγελία των προϊόντων έως και τη μισθοδοσία των υπαλλήλων. Για να δημιουργηθεί η βάση δεδομένων που θα συνδέεται με την δυναμική ιστοσελίδα θα χρειαστεί μία μεθοδολογία. Θα την δούμε στις επόμενες παραγράφους του κεφαλαίου αυτού, από την ανάλυση των απαιτήσεων της εφαρμογής μέχρι και τον κώδικα.

Εικόνα 6



Σχόλιο: Έτσι ο πηγαίος κώδικας της ιστοσελίδας δεν είναι προσπελάσιμος και αποφεύγονται *Sql injection* σαν αυτά ('or' '=' ή 'or' 'I'='I').

Καταγραφή Απαιτήσεων

Απαίτηση από το λογισμικό είναι μία λειτουργία που αυτό θα πρέπει να επιτελεί ή μία συνθήκη που θα πρέπει να ικανοποιεί όταν ολοκληρωθεί η κατασκευή του εν λόγω λογισμικού.

Οι απαιτήσεις ενός λογισμικού αλλιώς περιγράφονται από τον πελάτη (τα λεγόμενα specs) και αλλιώς από τον κατασκευαστή διότι βλέπουν το λογισμικό από διαφορετική σκοπιά. Όμως ο προσδιορισμός και η καταγραφή των απαιτήσεων είναι ιδιαίτερα σημαντική διότι αποτελεί τη βάση από την οποία θα αποφασιστούν οι λειτουργίες τις οφείλει να εκτελεί το λογισμικό που θα κατασκευασθεί.

Λειτουργικές προδιαγραφές, εντέλει, είναι το σχεδιάγραμμα για το πώς θέλει ο πελάτης ένα συγκεκριμένο έργο ή web εφαρμογή για την εμφάνιση και την εργασία.

3.2.2. Λειτουργικές απαιτήσεις (operational / functional specifications).

1. Αποθήκευση πληροφορίας:

- 1.1. Για προϊόν: Όνομα, είδος, κατασκευαστής και τιμή του.
- 1.2. Για αποθήκες: Όνομα, διεύθυνση.
- 1.3. Για υποκαταστήματα: Όνομα, διεύθυνση.
- 1.4. Για υπαλλήλους: Ονοματεπώνυμο, διεύθυνση, τηλέφωνο, μισθό, αριθμό αστυνομικής ταυτότητας.
- 1.5. Για προμηθευτές: Ονοματεπώνυμο, διεύθυνση.
- 1.6. Για παραγγελίες: Κωδικό, ποσότητα.

2. Δυνατότητα διαγραφής και επεξεργασίας για όλες τις παραπάνω κατηγορίες.

3. Τρόπος σύνδεσης μιας κατηγορίας με μία άλλη.

- 3.1. Ένα υποκατάστημα εφοδιάζεται από μόνο μια αποθήκη, αλλά μια αποθήκη εφοδιάζει πολλά υποκαταστήματα.
- 3.2. Κάθε υπάλληλος σε ένα υποκατάστημα αλλά σε ένα υποκατάστημα μπορούν να δουλεύουν πολλοί υπάλληλοι.
- 3.3. Κάθε προμηθευτής παρέχει πολλά προϊόντα και κάθε προϊόν παρέχεται από πολλούς προμηθευτές.
- 3.4. Κάθε προμηθευτής συνδέεται με πολλές αποθήκες και ισχύει και το αντίστροφο.

4. Διαδικασίες που υποστηρίζονται.

- 4.1. Για κάθε προϊόν θέλουμε να αποθηκεύουμε τα αποθέματα που υπάρχουν για κάθε προϊόν. Για κάθε προϊόν σε κάθε υποκατάστημα θέλουμε να έχουμε ένα ελάχιστο απόθεμα για κάθε προϊόν.
- 4.2. Μεταφορά προϊόντων από την αποθήκη προς κάποιο υποκατάστημα με αυτόματη μείωση και αύξηση των αντίστοιχων αποθεμάτων.
- 4.3. Αυτόματη παραγγελία θα ενεργοποιείται όταν σε κάποιο υποκατάστημα τα αποθέματα κάποιου προϊόντος πέσουν κάτω απ' το ελάχιστο και θα γίνεται εφοδιασμός έως τα αποθέματα να γίνουν τα διπλάσια του ελάχιστου.
- 4.4. Επίσης όταν τα αποθέματα μίας αποθήκης πέσουν κάτω απ' το ελάχιστο τότε θα γίνεται παραγγελία προς τον προμηθευτή με την χαμηλότερη τιμή έως τα αποθέματα αυτό του προϊόντος σε αυτή την αποθήκη να γίνουν τα διπλάσια του ελάχιστου.
- 4.5. Δυνατότητα δημιουργίας αντιγράφων ασφαλείας.

3.2.3. Απαιτήσεις λογισμικού.

Το λογισμικό θα πρέπει να έχει την δυνατότητα διαχείρισης μέσω του Διαδικτύου (over-the-web), καθώς το συγκεκριμένο project όπως ζητήθηκε έχει την απαίτηση για ευχρηστία χωρίς την εφαρμογή έτοιμων πλατφόρμων (π.χ. drupal , cscart κ.α.) έτσι κρίνονται ιδιαίτερα σημαντικές οι κοινές προτάσεις της διασύνδεσης χρήστη μηχανήματος (GUI – graphical user interface).

Σαφώς σημαντικό λόγω τις φύσης της εργασίας είναι ταχύτητα εκτέλεσης των εντολών του χρήστη και επιτυγχάνεται τόσο με την αποφυγή πολλών και περιττών flash εφαρμογών και ανανεώσεων της σελίδας. Γι' αυτό και η εκτεταμένη χρήση της τεχνολογίας διαδικτύου AJAX όπου αναφέρθηκε και στο προηγούμενο κεφάλαιο.

3.3. Entity – Relationship Model

3.3.1. Ορισμός E-R model:

Στην τεχνολογία λογισμικού, το μοντέλο οντότητα-συσχετίσεων (Relationship – Entity Model) είναι μια αφηρημένη και εννοιολογική αντιπροσώπευση των δεδομένων.

Η δημιουργία ενός μοντέλου οντοτήτων -συσχετίσεων είναι μια μέθοδος διαμόρφωσης βάσεων δεδομένων, που χρησιμοποιείται για να παραγάγει έναν τύπο εννοιολογικού σχήματος ή το σημασιολογικό πρότυπο στοιχείων ενός συστήματος, συχνά μια σχεσιακή βάση δεδομένων, και οι απαιτήσεις της σε μια τύπου από επάνω προς τα κάτω σχεδίαση (top to bottom design). Τα διαγράμματα που δημιουργούνται με αυτήν την διαδικασία καλούνται τα διαγράμματα οντότητα-σχέσης, διαγράμματα του ER ή συντομότερα ERDs.

Όταν μια σχεσιακή βάση δεδομένων πρόκειται να σχεδιαστεί, ένα διάγραμμα οντοτήτων-συσχετίσεων σχεδιάζεται σε ένα αρχικό στάδιο και αναπτύσσεται καθώς οι απαιτήσεις της βάσης δεδομένων και της επεξεργασίας της γίνονται καλύτερα κατανοητές. Οι λεπτομέρειες για το πώς να σχεδιαστούν τα διαγράμματα ποικίλλουν ελαφρώς από μια μέθοδο σε άλλη, αλλά όλες έχουν τα ίδια βασικά στοιχεία. Τους τύπους, τις ιδιότητες και τις σχέσεις οντοτήτων , αυτές οι τρεις κατηγορίες θεωρούνται επαρκείς για να διαμορφώσουν τα ουσιαστικά στατικά μέρη των αναγκών επεξεργασίας πληροφοριών οποιασδήποτε οργάνωσης σε μία οποιαδήποτε ΒΔ.

Στην πρώτη φάση της σχεδίασης πληροφοριακών συστημάτων χρησιμοποιούνται η σχεδίαση σε διαγράμματα, κατά τη διάρκεια της οποίας, γίνεται ανάλυση των απαιτήσεων από τον σχεδιαστή προκειμένου να περιγράψουν οι απαιτήσεις πληροφοριών ή το είδος αυτών που θα αποθηκευτεί σε μια βάση δεδομένων.

3.3.2. Βασικά Δομικά Στοιχεία του E – R μοντέλου:

Μια οντότητα μπορεί να οριστεί ως ένα πράγμα που αναγνωρίζεται ως ικανό μιας ανεξάρτητης ύπαρξης και που μπορεί να προσδιοριστεί μεμονωμένα. Οντότητα μπορεί να είναι ένα

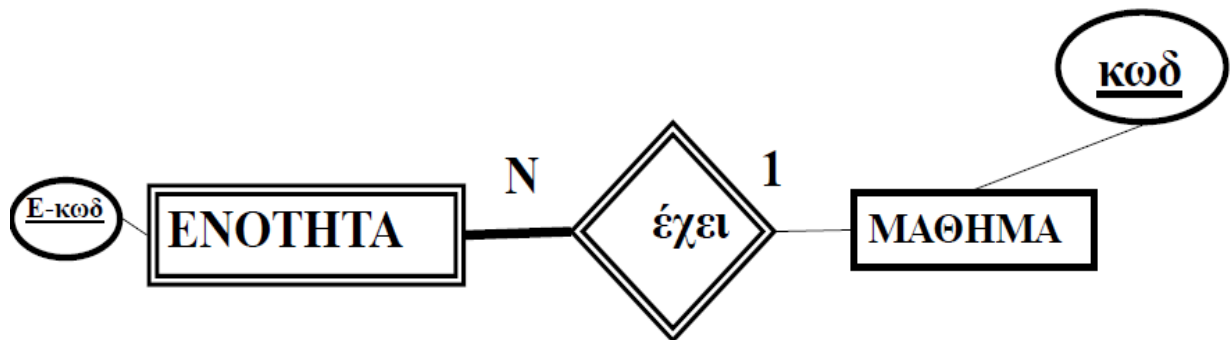
φυσικό αντικείμενο όπως ένα σπίτι ή ένα αυτοκίνητο, ένα γεγονός όπως μια πώληση σπιτιών ή μια υπηρεσία αυτοκινήτων, ή μια έννοια όπως μια συναλλαγή ή μια διαταγή πελατών.

Μια σχέση συλλαμβάνει πώς δύο ή περισσότερες οντότητες συσχετίζονται με τη μια άλλη. Οι σχέσεις μπορούν να θεωρηθούν ως ρήματα, που συνδέουν δύο ή περισσότερα ουσιαστικά. Παραδείγματα: ένα υποκατάστημα προμηθεύεται από ένα προμηθευτή .

Ιδιότητες μπορούν και οι δύο να έχουν και οι οντότητες και οι σχέσεις . Παραδείγματα: μια οντότητα υπαλλήλων να έχει την ιδιότητα του μισθού τους , ενώ η σχέση μπορεί να έχει μια ιδιότητα ημερομηνίας πληρωμής. Κάθε οντότητα (εκτός αν είναι μια αδύνατη οντότητα) πρέπει να έχει κατά ελάχιστο μοναδική ιδιότητα με την οποία και αναγνωρίζεται κάθε της στιγμιότυπο, το οποίο καλείται αρχικό κλειδί της οντότητας.

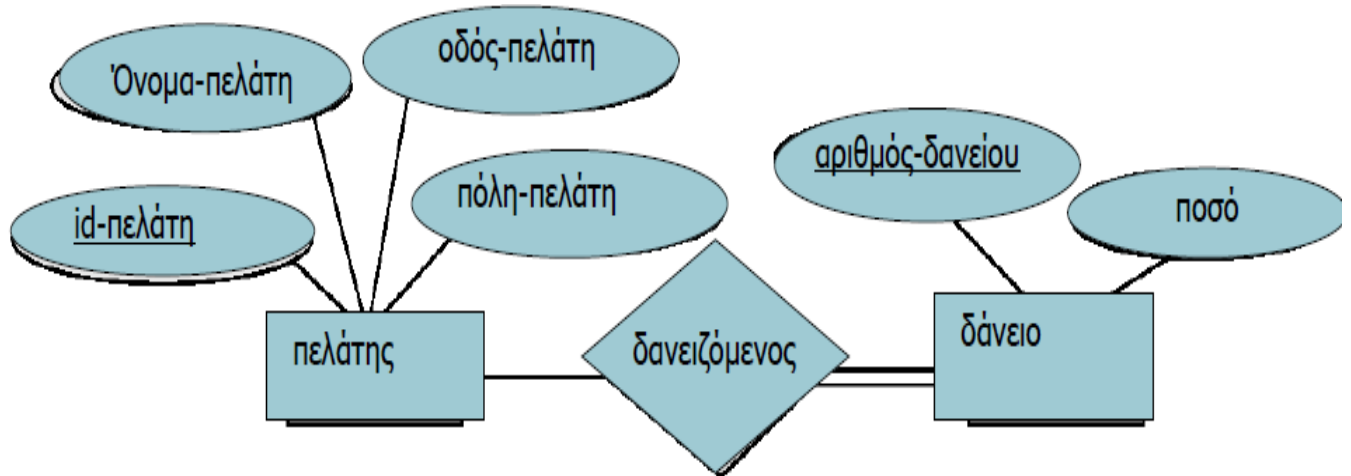
1. Γραμμές: για την σύνδεση των οντοτήτων με τα γνωρίσματα και με τις συσχετίσεις .
2. Ορθογώνια: που αναπαριστούν τα σύνολα των οντοτήτων
3. Ρόμβοι: που αναπαριστούν τις συσχετίσεις (ή σχέσεις) .
4. Ελλείψεις: που αναπαριστούν τα γνωρίσματα (ή ιδιότητες).
 - 4.1. Διπλές Ελλείψεις: Πλειότιμα γνωρίσματα.
 - 4.2. Διακεκομμένες Ελλείψεις: παραγόμενα γνωρίσματα.
5. Υπογράμμιση: γνωρίσματα που είναι πρωτεύοντα κλειδιά.
6. Ασθενείς οντότητες: (σχεδιάζεται με διπλό παραλληλόγραμμο) όταν μία οντότητα χρειάζεται να “δανειστεί” το κλειδί μιας άλλης.

Εικόνα 7



7. Συμμετοχή ενός συνόλου οντοτήτων σε ένα σύνολο συσχετίσεων:
 - 7.1. Ολική συμμετοχή: (σχεδιάζεται με διπλή γραμμή) κάθε οντότητα του συνόλου οντοτήτων συμμετέχει σε τουλάχιστον μια συσχέτιση στο σύνολο των συσχετίσεων. Π.χ. Κάθε δάνειο πρέπει να σχετίζεται με ένα πελάτη μέσω της συσχέτισης δανειζόμενος.
 - 7.2. Μερική συμμετοχή: ορισμένες οντότητες μπορεί να μην συμμετέχουν σε κάθε συσχέτιση του συνόλου συσχετίσεων. Π.χ. Μερική η συμμετοχή του πελάτη στο δανειζόμενος. Σχηματικά,

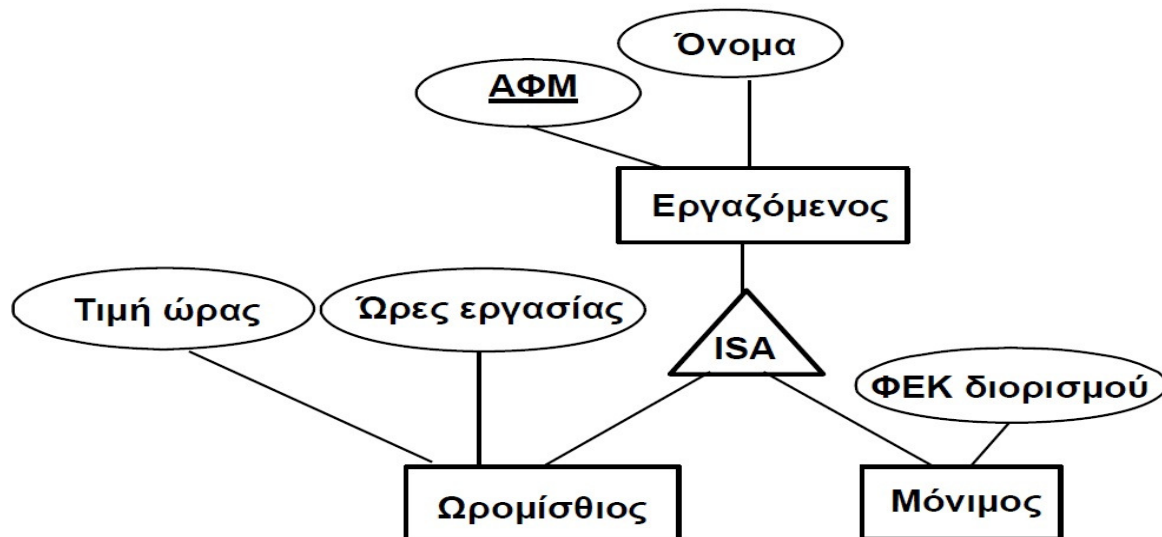
Εικόνα 8



8. Με την έννοια γενίκευση (generalization) εννοούμε τον εντοπισμό ενός συνόλου οντοτήτων (κλάση) που έχουν κοινά χαρακτηριστικά με πιο γενικευμένα αντικείμενα (υπέρκλαση). Η εξειδίκευση (specialization) είναι το ακριβώς αντίθετο της γενίκευσης, δηλαδή ο εντοπισμός υποσυνόλων ενός τύπου οντοτήτων με κοινά χαρακτηριστικά, τα οποία τα διαφοροποιούν από τα υπόλοιπα μέλη του.

Η συσχέτιση μεταξύ κάθε υποκλάσης και υπερκλάσης ονομάζεται ISA συσχέτιση.

Εικόνα 9



9. Η πληθικότητα (cardinality) Ο λόγος πληθικότητας

Η πληθικότητα (cardinality), περιγράφει τον αριθμό στιγμιότυπων ενός τύπου οντοτήτων που μπορούν να αντιστοιχίζονται με μία οντότητα ενός άλλου τύπου σε μια συσχέτιση.

Ο λόγος πληθικότητας ή πληθικός λόγος (cardinality ratio), είναι ο λόγος των πληθικοτήτων μιας συσχέτισης.

Μπορούμε να έχουμε συσχετίσεις με λόγο πληθικότητας:

1-1 (ένα-προς-ένα)

Αντιστοιχίζεται μια οντότητα ενός τύπου με το πολύ ή ακριβώς μια οντότητα ενός άλλου τύπου.

1-N (ένα-προς-πολλά)

Αντιστοιχίζεται μια οντότητα ενός τύπου με κανένα, ένα ή πολλά στιγμιότυπα ενός άλλου τύπου.

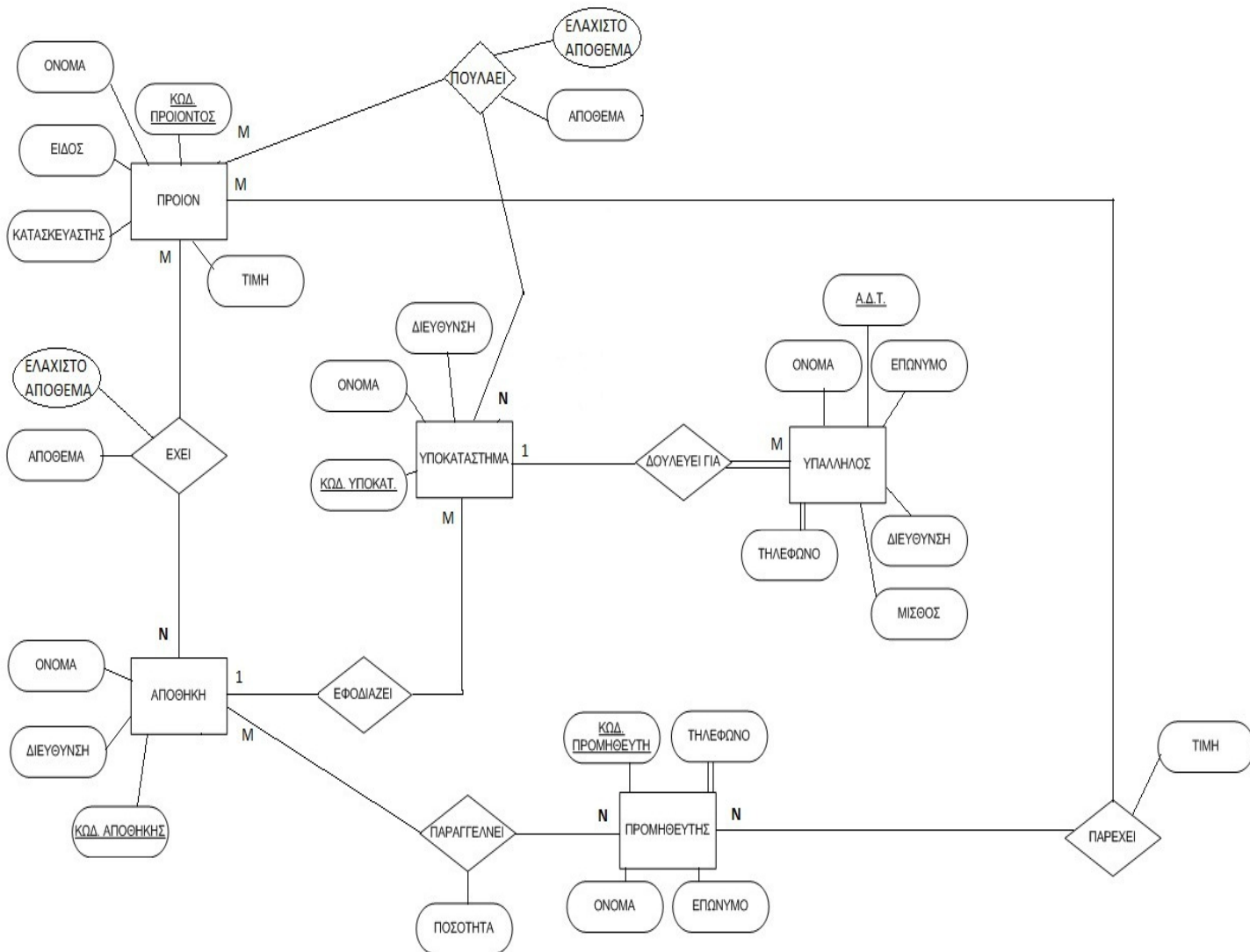
M-N (πολλά-προς-πολλά)

Αντιστοιχίζεται κάθε στιγμιότυπο του ενός τύπου με ένα, κανένα ή πολλά στιγμιότυπα του άλλου τύπου.

3.3.3. Από e-r μοντέλο σε πίνακα (κανονικοποίηση)

Μέσω του Microsoft Paint των Windows 7 παρουσιάζεται το μοντέλο οντοτήτων συσχετίσεων που αφορά την εργασία το οποίο έγινε σε χαρτί κατά την αρχική μας σχεδίαση. Σε αυτό το σχήμα καταλήξαμε προκειμένου να είναι ορθές οι πληθικότητες όπως δοθήκαν από τις απαιτήσεις και να είναι εύκολη η πραγματοποίηση των απαραίτητων ειδικών διεργασιών.

Εικόνα 10



Το σχεσιακό μοντέλο αναπαράστασης των δεδομένων μιας εφαρμογής (relational data model) καθιερώθηκε από τον Codd το 1970 και αποτέλεσε ένα από τα πιο απλά και ευέλικτα μοντέλα αυτού του είδους. Σε αυτό το μοντέλο, τα δεδομένα μιας εφαρμογής αναπαρίστανται ως ένα σύνολο από σχέσεις (relations) οι οποίες μπορεί να είναι πίνακες αρχεία. Στις πιο πολλές περιπτώσεις υιοθετείται η χρήση πινάκων (tables) που περιέχουν ένα πλήθος γραμμών (rows) και στηλών (columns). Η κάθε μια από αυτές τις γραμμές – οι οποίες στην ορολογία του μοντέλου ονομάζονται και πλειάδες (tuples) – περιέχει ένα σύνολο απλών πεδίων (attributes) τα οποία συσχετίζονται μεταξύ

τους. Επειδή όπως θα δούμε στις επόμενες παραγράφους οι πίνακες χρησιμοποιούνται για την αναπαράσταση των τύπων οντοτήτων καθώς και των τύπων συσχετίσεων που υφίστανται ανάμεσά τους, μπορούμε να θεωρήσουμε κάθε μια από τις γραμμές ενός πίνακα σαν ένα στιγμιότυπο οντότητας ή συσχέτισης ανάλογα με το αντικείμενο στο οποίο αναφέρεται.

Σχεδιασμός των τελικών σχέσεων (πινάκων) της βάσης δεδομένων με βάση το Διάγραμμα Οντοτήτων Συσχετίσεων Στο παρακάτω κείμενο περιγράφουμε 7 απλά βήματα με τα οποία μπορεί κάποιος από ένα καλοσχεδιασμένο Διάγραμμα Οντοτήτων Συσχετίσεων (ΔΟΣ) να οδηγηθεί σε ένα ικανοποιητικό σχεδιασμό της βάσης δεδομένων, δηλ. στον προσδιορισμό των σχέσεων (πινάκων) με τα κατηγορήματα και τα κλειδιά τους. Η μέθοδος που δίνουμε στη συνέχεια εφαρμόζεται εύκολα και δεν απαιτεί ιδιαίτερη πείρα (είναι καθαρά αλγοριθμική), αρκεί να έχει σχεδιαστεί προηγουμένως ένα ορθό και πλήρες ΔΟΣ. Σημειώστε πως ο παρακάτω σχεδιασμός δεν είναι πάντα βέλτιστος, κάτι που επιτυγχάνεται με την εφαρμογή των κανόνων κανονικοποίησης που αποτελούν αντικείμενο πιο προχωρημένης μελέτης των βάσεων δεδομένων.

ΒΗΜΑ 1

Για κάθε κανονικό τύπο οντοτήτων E στο ΔΟΣ δημιουργείτε μια σχέση R που να περιέχει όλα τα κατηγορήματα του E . Επιλέξτε ως πρωτεύον κλειδί της R ένα από τα κατηγορήματα κλειδιά του E . Αν το κλειδί του E είναι σύνθετο, το σύνολο των κατηγορημάτων του θα αποτελεί το πρωτεύον κλειδί της R .

ΒΗΜΑ 2

Για κάθε μη ισχυρό (ασθενή) τύπο οντοτήτων W στο ΔΟΣ με ιδιοκτήτη τον τύπο οντοτήτων E δημιουργείτε μια σχέση R στην οποία συμπεριλάβετε όλα τα κατηγορήματα του W ως κατηγορήματα της R . Επιπλέον συμπεριλάβετε ως κατηγορήματα της R τα κατηγορήματα των κλειδιών των σχέσεων που δημιουργήσατε για τους τύπους οντοτήτων – ιδιοκτήτες. Το κλειδί της R είναι ο συνδυασμός του κλειδιού (κλειδιών) του ιδιοκτήτη (των ιδιοκτητών) και αν υπάρχει, του μερικού κλειδιού του μη ισχυρού (ασθενούς) τύπου W .

ΒΗΜΑ 3

Για κάθε 1:1 δυαδική συσχέτιση R στο ΔΟΣ βρείτε τις σχέσεις S και T που αντιστοιχούν στους τύπους οντοτήτων που συμμετέχουν στην R . Επιλέξτε μια από τις σχέσεις (έστω την S) και συμπεριλάβετε στην S , ως νέο γνώρισμα (ξένο κλειδί), το πρωτεύον κλειδί της T . Συμπεριλάβετε επίσης ως κατηγορήματα της S όλα τα κατηγορήματα της συσχέτισης R (αν υπάρχουν).

ΒΗΜΑ 4

Για κάθε 1:N δυαδική συσχέτιση R στο ΔΟΣ βρείτε τις σχέσεις S και T που αντιστοιχούν στους τύπους οντοτήτων που συμμετέχουν στην R . Έστω T ο συμμετέχων τύπος οντοτήτων από την πλευρά N . Συμπεριλάβετε ως νέο γνώρισμα (ξένο κλειδί) στη σχέση T το πρωτεύον κλειδί της σχέσης S . Συμπεριλάβετε επίσης ως κατηγορήματα της T όλα τα κατηγορήματα της συσχέτισης R (αν υπάρχουν).

ΒΗΜΑ 5

Για κάθε $M:N$ δυαδική συσχέτιση R στο ΔΟΣ δημιουργείτε μια νέα σχέση S για να παρασταθεί η R . Συμπεριλάβετε ως κατηγορήματα στην S τα πρωτεύοντα κλειδιά των σχέσεων που παριστάνουν τους συμμετέχοντες τύπους οντοτήτων. Ο συνδυασμός τους θα αποτελεί το πρωτεύον κλειδί της S . Συμπεριλάβετε επίσης ως κατηγορήματα στην S όλα τα κατηγορήματα της συσχέτισης R (αν υπάρχουν).

Σημείωση: Οι συσχετίσεις 1:1 και 1:N μπορούν να απεικονιστούν με τον ίδιο τρόπο με τις $M:N$ (δηλ. με την δημιουργία νέας σχέσεις). Αυτή η προσέγγιση μπορεί να είναι καλύτερη όταν η συσχέτιση έχει λίγα στιγμιότυπα προκειμένου να αποφεύγονται τιμές NULL στα ξένα κλειδιά.

Το επόμενο βήμα αντιμετωπίζει τα πλειότιμα κατηγορήματα, δηλ. κατηγορήματα για τα οποία επιτρέπεται να δοθούν περισσότερες από μία τιμές (από ένα συγκεκριμένο πεδίο τιμών) για μια συγκεκριμένη εγγραφή. Υπενθυμίζουμε πως οντότητες που αντιστοιχίζονται σε σχέσεις με πλειότιμα κατηγορήματα (ΒΗΜΑ 1) δεν είναι νόμιμες.

ΒΗΜΑ 6

Για κάθε πλειότιμο γνώρισμα A κατασκευάστε μια νέα σχέση R η οποία θα περιλαμβάνει ένα γνώρισμα που θα αντιστοιχεί στο A καθώς και το γνώρισμα του πρωτεύοντος κλειδιού K της σχέσης που παριστάνει τον τύπο οντοτήτων ή τον τύπο συσχέτισης που έχει το R ως γνώρισμα. Πρωτεύον κλειδί της R είναι ο συνδυασμός του A και K .

ΒΗΜΑ 7

Για κάθε n-αδικό τύπο συσχέτισης R με $n > 2$, κατασκευάστε μια νέα σχέση S για την παράσταση του R. Συμπεριλάβετε ως κατηγορήματα της S τα πρωτεύοντα κλειδιά των σχέσεων που παριστάνουν τους συμμετέχοντες τύπους οντοτήτων. Συμπεριλάβετε επίσης ως κατηγορήματα της S όλα τα κατηγορήματα της συσχέτισης R (αν υπάρχουν). Το πρωτεύον κλειδί της S είναι συνήθως ένας συνδυασμός των (ξένων) κλειδίων που συμμετέχουν στην S.

Έτσι, ακολουθώντας αυτά τα βήματα καταλήγουμε στους εξής πίνακες:

Αποθήκη (Κώδ. Αποθήκης, Όνομα, Οδός, Αριθμός, Περιοχή, Πόλη)

Προϊόν (Κώδ. Προϊόντος, Όνομα, Τιμή, Είδος, Κατασκευαστής)

Υποκατάστημα (Κώδ. Υποκαταστήματος, Όνομα, Οδός, Αριθμός, Περιοχή, Πόλη, Κωδ. Αποθήκης)

Προμηθευτής (Κώδ. Προμηθευτή, Επώνυμο, Όνομα, Οδός, Αριθμός, Περιοχή, Πόλη)

Τηλ. Προμηθευτή (Κώδ. Προμηθευτή, Τηλέφωνο)

Υπάλληλος (Α.Δ.Τ., Επώνυμο, Όνομα, Οδός, Αριθμός, Περιοχή, Πόλη, Κώδ. Υποκαταστήματος)

Τηλ. Υπαλλήλου (Α.Δ.Τ., Τηλέφωνο)

Παρέχει(Κώδ. Προμηθευτή, Κώδ. Προϊόντος, Τιμή Πώλησης)

Παραγγέλνει(Κώδ. Αποθήκης, Κώδ. Προμηθευτή)

Απόθεμα Αποθηκών (Κώδ. Αποθήκης, Κώδ. Προϊόντος, Απόθεμα, Ελάχιστο Απόθεμα)

Απόθεμα Υποκαταστημάτων (Κώδ. Υποκαταστήματος, Κώδ. Προϊόντος, Απόθεμα, Ελάχιστο Απόθεμα)

3.4. SQL Κώδικας

3.4.1. Εξαγωγή SQL κώδικα από MySQL Workbench

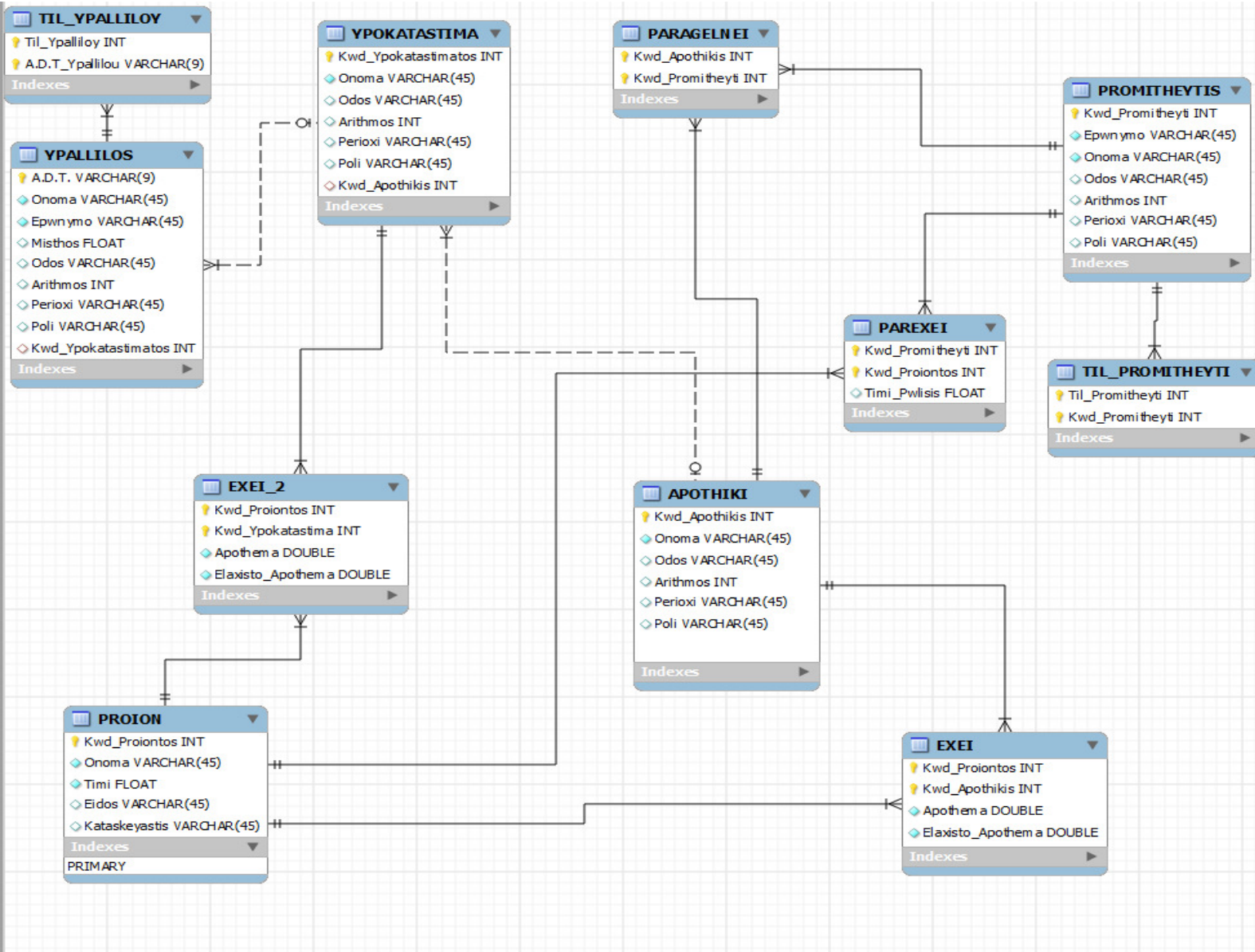
Σε αυτό το κομμάτι της εργασίας θα παρουσιαστεί και θα επεξηγηθεί ο κώδικας SQL που θα χρησιμοποιήσουμε για την δημιουργία της Β.Δ., δηλαδή την μετατροπή των πινάκων της παραγράφου 3.2 σε βάση δεδομένων για υπολογιστή ή δίκτυο.

Πρέπει πρώτα όμως να καταλάβουμε τι κάνουν κάποιες λέξεις (ή φράσεις) κλειδιά που φαίνονται στον παρακάτω κώδικα MySQL. Την εντολή Drop Table χρησιμοποιείται για ασφάλεια και προστατεύει την βάση από επιλοκές με άλλο πίνακα που πιθανόν να είχε το ίδιο όνομα. Στην ουσία διαγράφει όλα τα tables με τέτοιο όνομα. Για την δημιουργία ενός πίνακα ακολουθείτε αυτό το πρότυπο → CREATE TABLE `όνομα της βάσης`. `όνομα του πίνακα`. Κατόπιν, δηλώνουμε της στήλες του πίνακα (γνωρίσματα) π.χ. αν είναι αλφαριθμητική varchar, αν είναι ακέραιος int κ.α. και το πρωτεύον κλειδί.

Σε περίπτωση, που υπάρχει ξένο κλειδί γίνονται τρία πράγματα. Αρχικά, η εντολή foreign key(`όνομα πεδίου`) για την ονομασία του. Στη συνέχεια, δείχνουμε σε ποιο κλειδί αλλού πίνακα αναφέρεται αυτό → REFERENCES `όνομα της βάσης`. `όνομα του πίνακα που αναφέρεται` (`όνομα του πρωτεύον κλειδιού`). Τελικά, επιλέγουμε ανάμεσα απ' το Cascade ή No Action για το τι θα συμβεί όταν σε ανανέωση ή διαγραφή στον πίνακα της αναφοράς (No Action: καμία αλλαγή δεν επιτρέπεται, Cascade: επιτρέπονται αλλαγές).

Με το εργαλείο MySQL Workbench 5.2 CE, το οποίο είναι και freeware όπως αναφέραμε και σε προηγούμενο κεφάλαιο, συνδυάσαμε το e-r table με το αποτέλεσμα της κανονικοποίησης και το αποτέλεσμα ήταν αυτό της εικόνας 11.

Εικόνα 11



```

-- Table `mydb`.`PROION`
-----
DROP TABLE IF EXISTS `mydb`.`PROION` ;
CREATE TABLE IF NOT EXISTS `mydb`.`PROION` (
  `Kwd_Proiontos` INT UNSIGNED NOT NULL ,
  `Onoma` VARCHAR(45) NOT NULL ,
  `Timi` FLOAT UNSIGNED NOT NULL ,
  `Eidos` VARCHAR(45) NULL DEFAULT NULL ,
  `Kataskeyastis` VARCHAR(45) NULL DEFAULT NULL ,
  PRIMARY KEY (`Kwd_Proiontos`) )
ENGINE = InnoDB;

-----
-- Table `mydb`.`APOTHIKI`
-----
DROP TABLE IF EXISTS `mydb`.`APOTHIKI` ;
CREATE TABLE IF NOT EXISTS `mydb`.`APOTHIKI` (
  `Kwd_Apothikis` INT UNSIGNED NOT NULL ,

```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
`Onoma` VARCHAR(45) NOT NULL ,
`Odos` VARCHAR(45) NULL DEFAULT NULL ,
`Arithmos` INT UNSIGNED NULL DEFAULT NULL ,
`Perioxi` VARCHAR(45) NULL DEFAULT NULL ,
`Poli` VARCHAR(45) NULL DEFAULT NULL ,
PRIMARY KEY (`Kwd_Apothikis`) )
ENGINE = InnoDB;

-----
-- Table `mydb`.`PROMITHEYTIS`
-----
DROP TABLE IF EXISTS `mydb`.`PROMITHEYTIS` ;
CREATE TABLE IF NOT EXISTS `mydb`.`PROMITHEYTIS` (
  `Kwd_Promitheyti` INT UNSIGNED NOT NULL ,
  `Epwnymo` VARCHAR(45) NOT NULL ,
  `Onoma` VARCHAR(45) NOT NULL ,
  `Odos` VARCHAR(45) NULL DEFAULT NULL ,
  `Arithmos` INT UNSIGNED NULL DEFAULT NULL ,
  `Perioxi` VARCHAR(45) NULL DEFAULT NULL ,
  `Poli` VARCHAR(45) NULL DEFAULT NULL ,
  PRIMARY KEY (`Kwd_Promitheyti`) )
ENGINE = InnoDB;

-----
-- Table `mydb`.`YPOKATASTIMA`
-----
DROP TABLE IF EXISTS `mydb`.`YPOKATASTIMA` ;
CREATE TABLE IF NOT EXISTS `mydb`.`YPOKATASTIMA` (
  `Kwd_Ypokatastimatos` INT UNSIGNED NOT NULL ,
  `Onoma` VARCHAR(45) NOT NULL ,
  `Odos` VARCHAR(45) NULL DEFAULT NULL ,
  `Arithmos` INT UNSIGNED NULL DEFAULT NULL ,
  `Perioxi` VARCHAR(45) NULL DEFAULT NULL ,
  `Poli` VARCHAR(45) NULL DEFAULT NULL ,
  `Kwd_Apothikis` INT UNSIGNED NULL DEFAULT NULL ,
  PRIMARY KEY (`Kwd_Ypokatastimatos`) ,
  INDEX `Kwd_Apothikis` (`Kwd_Apothikis` ASC) ,
  CONSTRAINT `Kwd_Apothikis`
    FOREIGN KEY (`Kwd_Apothikis`)
    REFERENCES `mydb`.`APOTHIKI` (`Kwd_Apothikis`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;

-----
-- Table `mydb`.`YPALLILOS`
-----
DROP TABLE IF EXISTS `mydb`.`YPALLILOS` ;
CREATE TABLE IF NOT EXISTS `mydb`.`YPALLILOS` (
  `A.D.T.` VARCHAR(9) NOT NULL ,
  `Epwnymo` VARCHAR(45) NOT NULL ,
  `Onoma` VARCHAR(45) NOT NULL ,
  `Misthos` FLOAT UNSIGNED NULL DEFAULT NULL ,
  `Odos` VARCHAR(45) NULL DEFAULT NULL ,
  `Arithmos` INT UNSIGNED NULL DEFAULT NULL ,
  `Perioxi` VARCHAR(45) NULL DEFAULT NULL ,
```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
`Poli` VARCHAR(45) NULL DEFAULT NULL ,
`Kwd_Ypokatastimatos` INT UNSIGNED NULL DEFAULT NULL ,
PRIMARY KEY (`A.D.T.`) ,
INDEX `Kwd_Ypokatastimatos` (`Kwd_Ypokatastimatos` ASC) ,
CONSTRAINT `Kwd_Ypokatastimatos`
  FOREIGN KEY (`Kwd_Ypokatastimatos` )
  REFERENCES `mydb`.`YPOKATASTIMA` (`Kwd_Ypokatastimatos` )
  ON DELETE CASCADE
  ON UPDATE CASCADE)
ENGINE = InnoDB;

-----
-- Table `mydb`.`APOTHEMA_APOTHIKWN`
-----
DROP TABLE IF EXISTS `mydb`.`APOTHEMA_APOTHIKWN` ;
CREATE TABLE IF NOT EXISTS `mydb`.`APOTHEMA_APOTHIKWN` (
  `Kwd_Proiontos` INT UNSIGNED NOT NULL ,
  `Kwd_Apothikis` INT UNSIGNED NOT NULL ,
  `Apothema` DOUBLE NULL ,
  PRIMARY KEY (`Kwd_Proiontos`, `Kwd_Apothikis`) ,
  INDEX `Kwd_Proion` (`Kwd_Proiontos` ASC) ,
  INDEX `Kwd_Apothiki` (`Kwd_Apothikis` ASC) ,
  CONSTRAINT `Kwd_Proion`
    FOREIGN KEY (`Kwd_Proiontos` )
    REFERENCES `mydb`.`PROION` (`Kwd_Proiontos` )
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `Kwd_Apothiki`
    FOREIGN KEY (`Kwd_Apothikis` )
    REFERENCES `mydb`.`APOTHIKI` (`Kwd_Apothikis` )
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;

-----
-- Table `mydb`.`APOTHEMA_YPOKATASTIMATWN`
-----
CREATE TABLE IF NOT EXISTS `mydb`.`APOTHEMA_YPOKATASTIMATWN` (
  `Kwd_Proiontos` INT UNSIGNED NOT NULL ,
  `Kwd_Ypokatastimatos` INT UNSIGNED NOT NULL ,
  `Apothema` DOUBLE NULL ,
  PRIMARY KEY (`Kwd_Proiontos`, `Kwd_Ypokatastimatos`) ,
  INDEX `Kwd_Proion` (`Kwd_Proiontos` ASC) ,
  INDEX `Kwd_Ypokatastimata` (`Kwd_Ypokatastimatos` ASC) ,
  CONSTRAINT `Kwd_Proion0`
    FOREIGN KEY (`Kwd_Proiontos` )
    REFERENCES `mydb`.`PROION` (`Kwd_Proiontos` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `Kwd_Ypokatastimatos0`
    FOREIGN KEY (`Kwd_Ypokatastimatos` )
    REFERENCES `mydb`.`YPOKATASTIMA` (`Kwd_Ypokatastimatos` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
-----  
-- Table `mydb`.`PAREXEI`  
-----
```

```
DROP TABLE IF EXISTS `mydb`.`PAREXEI` ;  
CREATE TABLE IF NOT EXISTS `mydb`.`PAREXEI` (  
  `Kwd_Promitheyti` INT UNSIGNED NOT NULL ,  
  `Kwd_Proiontos` INT UNSIGNED NOT NULL ,  
  `Timi_Pwlisis` FLOAT UNSIGNED NULL DEFAULT NULL ,  
  PRIMARY KEY (`Kwd_Promitheyti`, `Kwd_Proiontos`),  
  INDEX `Kvd_Promitheyti` (`Kwd_Promitheyti` ASC) ,  
  INDEX `Kwd_Proiontos` (`Kwd_Proiontos` ASC) ,  
  CONSTRAINT `Kwd_Promitheyti`  
    FOREIGN KEY (`Kwd_Promitheyti` )  
    REFERENCES `mydb`.`PROMITHEYTIS` (`Kwd_Promitheyti` )  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
  CONSTRAINT `Kwd_Proiontos`  
    FOREIGN KEY (`Kwd_Proiontos` )  
    REFERENCES `mydb`.`PROION` (`Kwd_Proiontos` )  
    ON DELETE CASCADE  
    ON UPDATE CASCADE)  
ENGINE = InnoDB;
```

```
-----  
-- Table `mydb`.`TIL_YPALLILOY`  
-----
```

```
DROP TABLE IF EXISTS `mydb`.`TIL_YPALLILOY` ;  
CREATE TABLE IF NOT EXISTS `mydb`.`TIL_YPALLILOY` (  
  `A.D.T_Ypalliloy` VARCHAR(9) NOT NULL ,  
  `Til_Ypalliloy` DOUBLE UNSIGNED NOT NULL ,  
  PRIMARY KEY (`Til_Ypalliloy`, `A.D.T_Ypalliloy`),  
  INDEX `A.D.T_Ypalliloy` (`A.D.T_Ypalliloy` ASC) ,  
  UNIQUE INDEX `Til_Ypalliloy_UNIQUE` (`Til_Ypalliloy` ASC) ,  
  CONSTRAINT `A.D.T_Ypalliloy`  
    FOREIGN KEY (`A.D.T_Ypalliloy` )  
    REFERENCES `mydb`.`YPALLILOS` (`A.D.T.` )  
    ON DELETE CASCADE  
    ON UPDATE CASCADE)  
ENGINE = InnoDB;
```

```
-----  
-- Table `mydb`.`PARAGELNEI`  
-----
```

```
DROP TABLE IF EXISTS `mydb`.`PARAGELNEI` ;  
CREATE TABLE IF NOT EXISTS `mydb`.`PARAGELNEI` (  
  `Kwd_Apothikis` INT UNSIGNED NOT NULL ,  
  `Kwd_Promitheyti` INT UNSIGNED NOT NULL ,  
  `Posotita` FLOAT UNSIGNED NULL DEFAULT NULL ,  
  PRIMARY KEY (`Kwd_Apothikis`, `Kwd_Promitheyti`),  
  INDEX `Kwd_Apothik` (`Kwd_Apothikis` ASC) ,  
  INDEX `Kwd_Promitheyti` (`Kwd_Promitheyti` ASC) ,  
  CONSTRAINT `Kwd_Apothik`  
    FOREIGN KEY (`Kwd_Apothikis` )  
    REFERENCES `mydb`.`APOTHIKI` (`Kwd_Apothikis` )  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,
```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
CONSTRAINT `Kwd_Promitheyti`
  FOREIGN KEY (`Kwd_Promitheyti`)
  REFERENCES `mydb`.`PROMITHEYTIS` (`Kwd_Promitheyti`)
  ON DELETE CASCADE
  ON UPDATE CASCADE)
ENGINE = InnoDB;

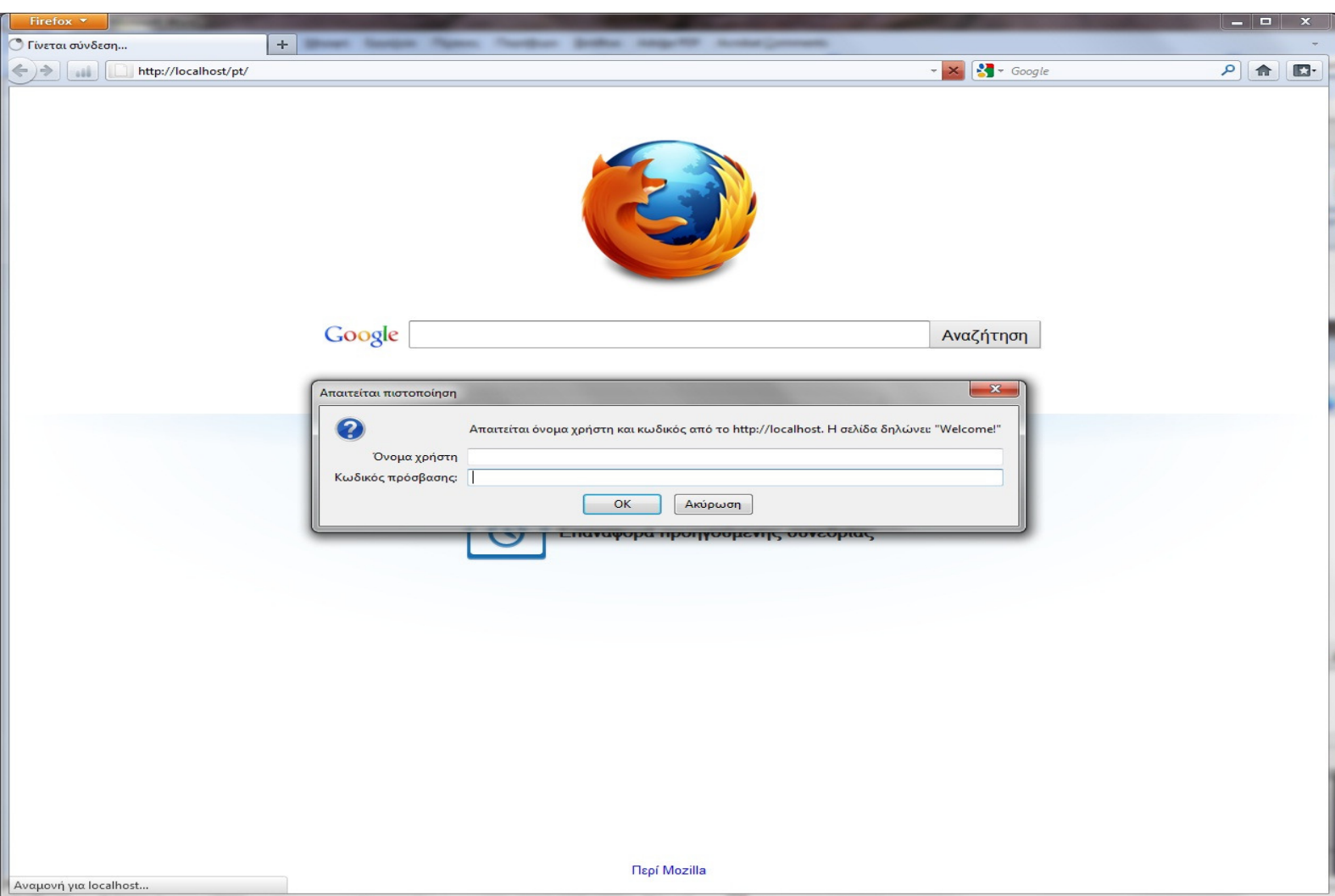
-- -----
-- Table `mydb`.`TIL_PROMITHEYTI`
-- -----
DROP TABLE IF EXISTS `mydb`.`TIL_PROMITHEYTI` ;
CREATE TABLE IF NOT EXISTS `mydb`.`TIL_PROMITHEYTI` (
  `Kwd_Promitheyti` INT UNSIGNED NOT NULL ,
  `Til_Promitheyti` DOUBLE UNSIGNED NOT NULL ,
  PRIMARY KEY (`Til_Promitheyti`, `Kwd_Promitheyti`) ,
  INDEX `Kwd_Promitheyti` (`Kwd_Promitheyti` ASC) ,
  UNIQUE INDEX `Til_Promitheyti_UNIQUE` (`Til_Promitheyti` ASC) ,
  CONSTRAINT `Kwd_Promitheyti`
    FOREIGN KEY (`Kwd_Promitheyti`)
    REFERENCES `mydb`.`PROMITHEYTIS` (`Kwd_Promitheyti`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;
```


4. ΛΕΙΤΟΥΡΓΙΕΣ ΣΕ PHP

Στις επόμενα υποκεφάλαια θα παρουσιαστούν οι βασικές λειτουργίες που εκτελούνται στην δυναμική μας ιστοσελίδα. Από την είσοδο του χρήστη, την σύνδεση με την βάση δεδομένων του πληροφοριακού συστήματος, ως και τις ειδικότερες όπως ο μηχανισμός των αυτόματων παραγγελιών. Θα δείξουμε τον κώδικα php αυτών των λειτουργιών και θα εξηγήσουμε τα βασικότερα του κομμάτια. Στην ανάλυση αυτή προσθέσαμε εικόνες, από την ιστοσελίδα που δημιουργήσαμε, για κάθε μία εκ των εφαρμογών που θα δείχνουμε.

4.1. Λειτουργία login (είσοδος στην ιστοσελίδα)

Εικόνα 12



Κώδικας PHP για την είσοδο στο site

```
<?php
//Login Popup Box
$users = array('admin' => 'admin'); //1
function httpauth() { //2
    header("WWW-Authenticate: Basic realm="Welcome!"); //3
    header("HTTP/1.0 401 Unauthorized"); //4
```

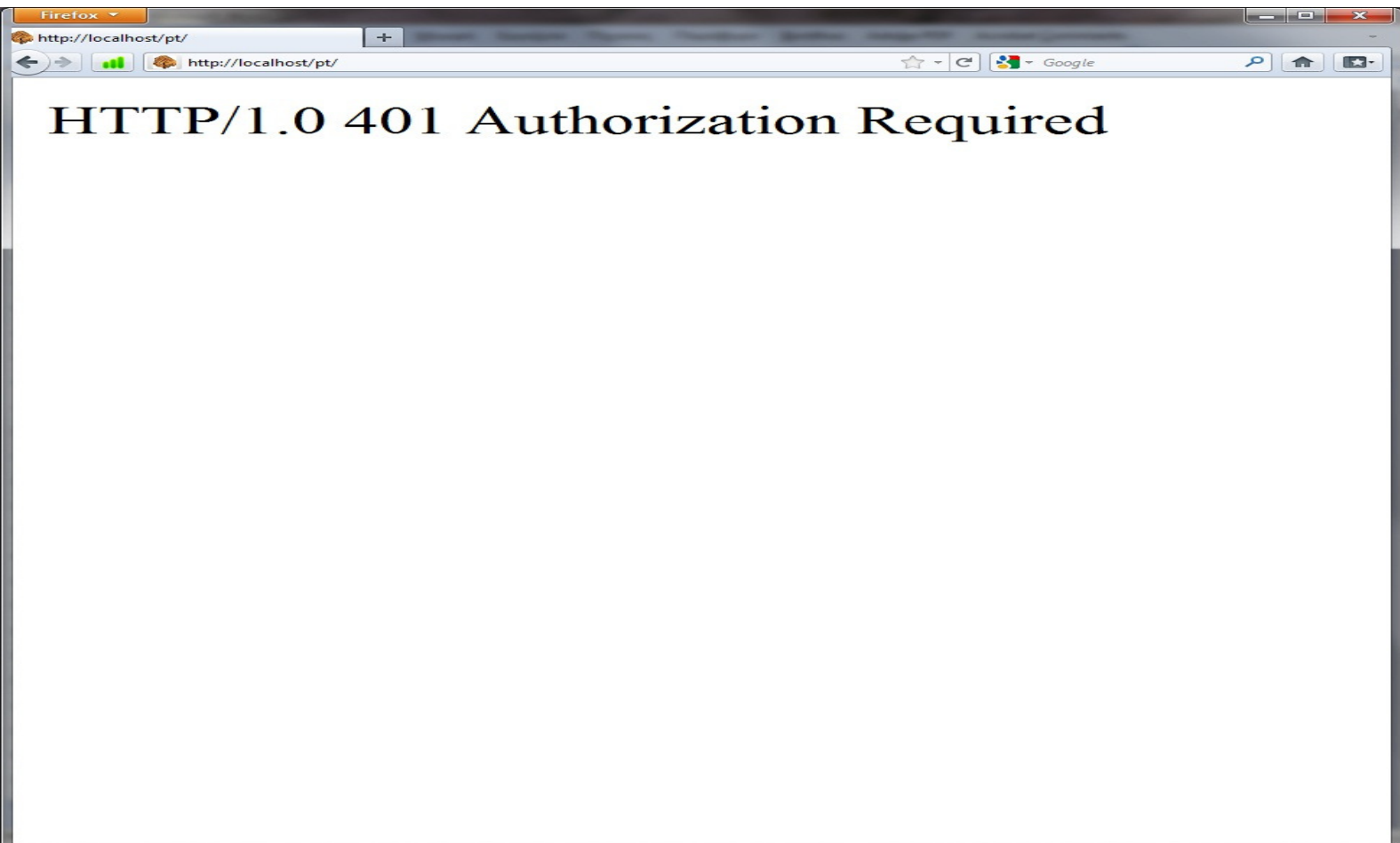
```
    echo 'HTTP/1.0 401 Authorization Required';           //5
    exit;                                               //6
}
while(!array_key_exists($_SERVER['PHP_AUTH_USER'], $users) || //7
$_SERVER['PHP_AUTH_PW'] != $users[$_SERVER['PHP_AUTH_USER']]){ //8
    httpauth();                                       //9
}
?>
```

Επεξήγηση του κώδικα PHP

Στην πρώτη γραμμή δημιουργούμε ένα πίνακα που καταχωρείται σε μια μεταβλητή και έχει ένα στοιχείο και τον δείκτη του στοιχείου. Κατόπιν, εισάγουμε σε μία συνάρτηση το μήνυμα που είδαμε παραπάνω που εμφανίζεται όταν υπάρξει προσπάθεια για την είσοδο στην ιστοσελίδα με την εντολή header.

Τελικά, μέσω ενός βρόγχου ελέγχεται η αυθεντικότητα του χρήστη με το στοιχείο του πίνακα να είναι το username και ο δείκτης του στοιχείου το password. Όταν λαμβάνει χώρα λάθος στην ταυτοποίηση της αυθεντικότητας ή πατηθεί το κουμπί cancel ενημερώνεται ο χρήστης ότι συνέβει το συγκεκριμένο HTTP σφάλμα.

Εικόνα 13



Το μήνυμα σε περιπτώσεις όπου αποτυγχάνει η επιβεβαίωση της αυθεντικότητας του χρήστη κατά την προσπάθεια εισόδου σε οποιαδήποτε από τις ιστοσελίδες διαχείρισης της εργασίας που δημιουργήσαμε.

4.2. Λειτουργία connect (σύνδεση στη Β.Δ.)

Κώδικας PHP για την αναζήτηση στοιχείων

```
//MySQL Connection Elements
$myServer = "localhost"; //1
$myUser = "root"; //2
$myPass = "1234"; //3
$myDB = "mydb"; //4

//connection to the database
$dbhandle = mysql_connect($myServer, $myUser, $myPass) //5
or die("Couldn't connect to SQL Server on $myServer"); //6

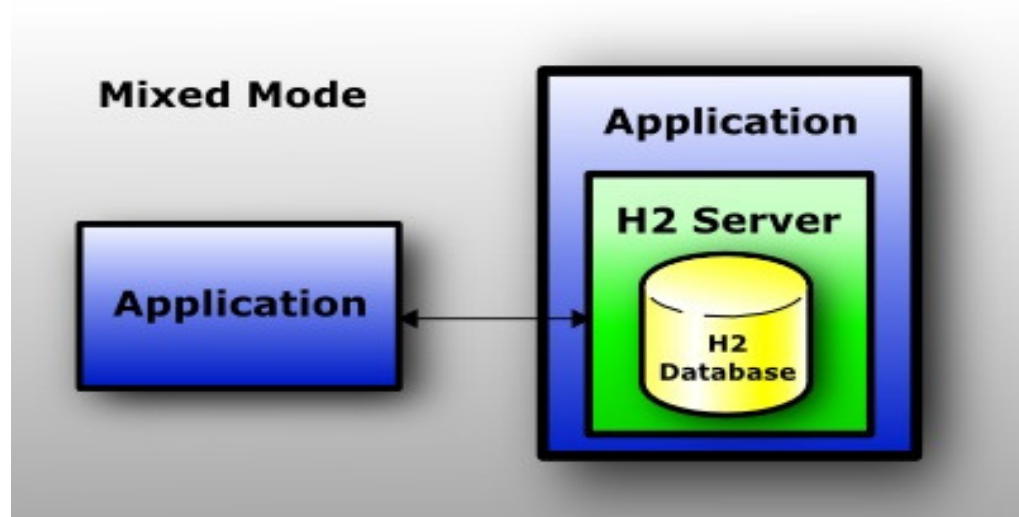
//select a database to work with
$selected = mysql_select_db($myDB, $dbhandle) or die("Couldn't open database $myDB"); //7
```

Επεξήγηση του κώδικα PHP

Στις πρώτες τα τέσσερις γραμμές τοποθετούμε σε μεταβλητές τα απαραίτητα στοιχεία για να γίνει σύνδεση σε μία βάση δεδομένων. Δηλαδή τον υπολογιστή (ή server) στον οποίο υπάρχει - localhost- το όνομα της -root- και το password.

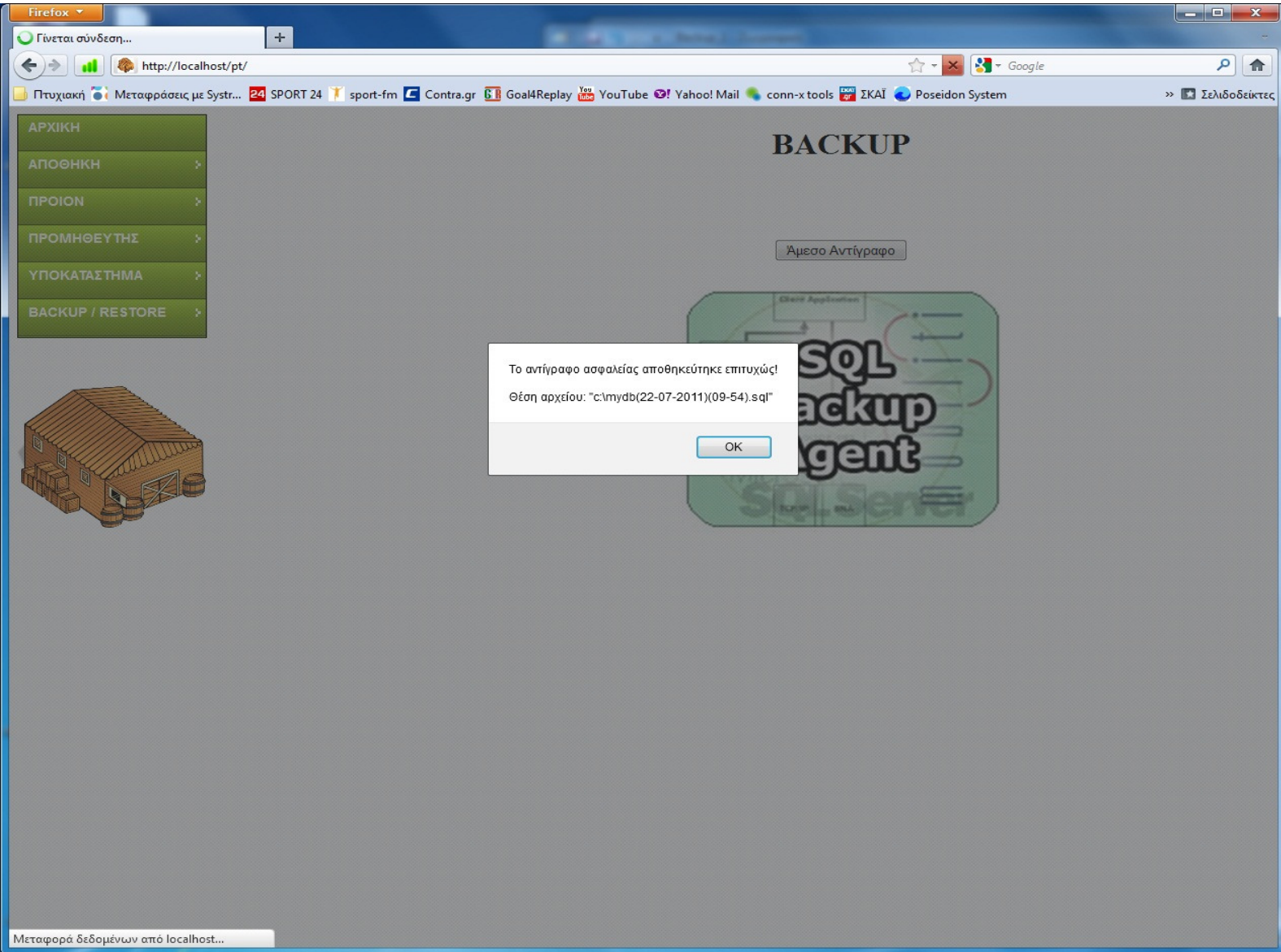
Στη συνέχεια εκτελείται η εντολή `mysql_connect` με τις μεταβλητές που είδαμε παραπάνω και η επιστροφή της εκχωρείται σε μια νέα μεταβλητή, αλλιώς έχουμε αποτυχία σύνδεσης στην `mysql` και εμφανίζεται κατάλληλο μήνυμα. Στην γραμμή 7 μέσω της εντολής `mysql_select_db` επιλέγεται η βάση δεδομένων που διαχειριζόμαστε ή διαφορετικά ενημερώνεται ο χρήστης μέσω μηνύματος.

Εικόνα 14



4.3. Λειτουργία Backup (δημιουργία αντιγράφου ασφαλείας)

Εικόνα 15



Μήνυμα: "Το αντίγραφο ασφαλείας...Θέση αρχείου: [διαδρομή αρχείου](ημερομηνία και ώρα).sql"

Κώδικας PHP για την δημιουργία αντιγράφου ασφαλείας

```
<?php
$backupFile = "c:\\\\" . $myDB . "(" . date("d-m-Y") . ")" . "(" . (date("H-i")) . ")" . ".sql";
$backupcmd = "mysqldump --user=root --password=1234 --databases mydb > $backupFile";
exec($backupcmd);
print "<script type='text/javascript'>";
print "alert('Το αντίγραφο ασφαλείας αποθηκεύτηκε επιτυχώς!\n\nΘέση αρχείου: \"$backupFile\")";
print "</script>";
?>
```

Ο κώδικας υπάρχει και είναι ελεύθερος για χρήση από <http://www.php-mysql-tutorial.com/wikis/mysql-tutorials/using-php-to-backup-mysql-databases.aspx>

4.4. Λειτουργία restore (Επαναφορά αντιγράφου ασφαλείας)

Εικόνα 16



Μήνυμα: “Επαναφορά Βάσης...Θέση αρχείου: [διαδρομή αρχείου](ημερομηνία,ώρα).sql”

Κώδικας PHP για την επαναφορά αντιγράφου ασφαλείας


```
<?php
$x=$_REQUEST["restorepath"];
$restoreFile = "c:\\\\" . $x;
$restorecmd = "mysql --verbose --user=root --password=1234 < $restoreFile";
exec($restorecmd);
print "<script type=\\\"text/javascript\\\">";
print "alert('Η επαναφορά της βάσης έγινε επιτυχώς!!\\n\\nΘέση αρχείου επαναφοράς:
\\\"$restoreFile\\\"')";
print "</script>";
?>
```

Ο κώδικας υπάρχει και είναι ελεύθερος για χρήση από <http://www.php-mysql-tutorial.com/wikis/mysql-tutorials/using-php-to-backup-mysql-databases.aspx>

4.5. Αναζήτηση Στοιχείου

Εικόνα 17

- ΑΡΧΙΚΗ
- ΑΠΟΘΗΚΗ >
- ΠΡΟΙΟΝ >
- ΠΡΟΜΗΘΕΥΤΗΣ >
- ΥΠΟΚΑΤΑΣΤΗΜΑ >
- BACKUP / RESTORE >



ΥΠΟΚΑΤΑΣΤΗΜΑΤΑ

Κωδ. Υποκαταστήματος
 Όνομα
 Οδός
 Περιοχή
 Πόλη
 Κωδ. Αποθήκης

Search: ←

Κωδ. Υποκαστήματος:	Όνομα:	Οδός:	Αριθμός:	Περιοχή:	Πόλη:	Κωδ. Αποθήκης:	Εργασία:
136	K5	-	2	Korfalonas	Kissamos	2	<input type="button" value="Edit"/> <input type="button" value="X"/>
647	AA2	BB2	16	Korfalonas	-	3	<input type="button" value="Edit"/> <input type="button" value="X"/>

Σχόλιο: Εκτελείται η αναζήτηση με επιλεγμένο το radiobutton “Περιοχή” και στο textbox τα γράμματα “Korf”. Τα αποτελέσματα της αναζήτησης εμφανίζονται χωρίς την ανανέωση ολόκληρης της σελίδας λόγω της χρήσης της τεχνικής Διαδικτύου Ajax

Κώδικας PHP για την αναζήτηση στοιχείων

```

$sql="SELECT * FROM ypokatastima WHERE $radiobtn like '%$stermtxt%' "; //1
$result = mysql_query($sql); //2
while($row = mysql_fetch_array($result)) //3
{
    $i+=1;
    echo '<td>'. "<a href=http://localhost/pt/exei_2.php?code_yp=$row[0]>$row[0]</a>". '</td>'; //4
    print "<input style=\"display:none\" type=\"text\" name=\"kwd[$i]\" value=$row[0] />"; //5
    echo '<td>' . $row[1] . '</td>'; //6
    print "<input style=\"display:none\" type=\"text\" name=\"onoma[$i]\" value=$row[1] />"; //7
    echo '<td>' . $row[2] . '</td>'; //8
    print "<input style=\"display:none\" type=\"text\" name=\"odos[$i]\" value=$row[2] />"; //9
    echo '<td>' . $row[3] . '</td>'; //10
    print "<input style=\"display:none\" type=\"text\" name=\"arithmos[$i]\" value=$row[3] />"; //11
    echo '<td>' . $row[4] . '</td>'; //13
    print "<input style=\"display:none\" type=\"text\" name=\"perioxi[$i]\" value=$row[4] />"; //14
    echo '<td>' . $row[5] . '</td>'; //15
    print "<input style=\"display:none\" type=\"text\" name=\"poli[$i]\" value=$row[5] />"; //16
}
    
```

```

echo '<td>' . $row[6] . '</td>'; //17
print "<input style=\"display:none\" type=\"text\" name=\"kwd_apoth[$i]\" value=$row[6] />"; //18
echo '<td>'; //19
print "<input name=\"E$i\" type=\"submit\" value=\"Edit\" //20
style=\"color:#FFFFFF; background-color:#09C; font:bold\" />"; //21
print "<input name=\"X$i\" type=\"submit\" value=\"X\" //22
style=\"color:#FFFFFF; background-color:#FF0000; font:bold\" />"; //23
echo '</td>';
echo '<tr>';
}

```

Επεξήγηση του κώδικα PHP

Αρχικά, δημιουργείται το ερώτημα (query) και εισάγεται σε μία μεταβλητή. Στη συνέχεια εκτελείται το ερώτημα προς την Β.Δ. ενώ και αυτό εισάγεται σε μία άλλη μεταβλητή προκειμένου να γίνει ένας βρόγχος όπου επιστρέφει σε κάθε του επανάληψη μια πλειάδα (γραμμές 1, 2, 3).

Στις επόμενες γραμμές (4 έως 19) πραγματοποιείται η εμφάνιση σε μορφή πίνακα όλων των πλειάδων που “υπακούν” στη συνθήκη που ορίζεται και η εισαγωγή τους σε κρυφά textboxes για την μετέπειτα χρησιμοποίησή τους. Μάλιστα, στις τελευταίες γραμμές τοποθετούνται τα κουμπία-εργαλεία τα οποία είναι η διαγραφή και η επεξεργασία.

Η σύνταξη μιας τέτοιας εντολής όπου επιστρέφει κάποιες (ή όλες) στήλες της Β.Δ. που ανταποκρίνονται σε κάποια συνθήκη είναι: SELECT [όνομα στήλης] FROM [όνομα πίνακα] WHERE [κάποια συνθήκη]. Το “*” στην περίπτωση μας επιστρέφει ολόκληρη την γραμμή του πίνακα, όχι μία μόνο στήλη.

4.6. Εισαγωγή στοιχείου

Εικόνα 18

The screenshot displays the 'ΑΠΟΘΗΚΕΣ' (Warehouses) section of the application. On the left, a green sidebar contains navigation links: ΑΡΧΙΚΗ, ΑΠΟΘΗΚΗ, ΠΡΟΪΟΝ, ΠΡΟΜΗΘΕΥΤΗΣ, ΥΠΟΚΑΤΑΣΤΗΜΑ, and BACKUP / RESTORE. The main content area features a table with columns for 'Κωδ. Αποθήκης', 'Όνομα', 'Οδός', 'Αριθμός', 'Περιοχή', and 'Πόλη'. Below the table, there is a search bar and two buttons: 'Insert' and 'Show All'. A red arrow points to the 'Insert' button. An inset window titled 'Insert: ΑΠΟΘΗΚΗΣ - Mozilla Firefox' shows a form with the same fields as the table, each with an asterisk indicating it is a required field. The 'Αριθμός' field has a note '(0 για άγνωστο)'. An 'Insert!' button is at the bottom of the form. A small illustration of a warehouse is located in the bottom left corner of the page.

Στο καινούριο παράθυρο που ανοίγει όταν επιλέξουμε το κουμπί *Insert*, μπορεί να γίνει καταχώριση μίας νέας εγγραφής.

Κώδικας PHP για την εισαγωγή νέων στοιχείων

```

if ( $kwd_ap == Null || $sonoma == Null){ //1
    print "<script type=\"text/javascript\">"; //2
    print "alert('Δεν έχετε συμπληρώσει τα υποχρεωτικά πεδία.')"; //3
    print "</script>"; //4
else //5
{
    if ($odos == Null ){ //6
        $odos = "-" ;} //7
    if ($arithmos == Null ){ //8
        $arithmos = '0' ;} //9
    if ($perioxi == Null ){ //10
        $perioxi = "-" ;} //11
    if ($poli == Null ){ //13
        $poli = "-" ;} //14

    if (!is_numeric($kwd_ap) || $kwd_ap < 0) //15
    { print "<script type=\"text/javascript\">"; //16
      print "alert('Στο πεδίο \"Κωδ. Αποθήκης\" εισάγετε μονο θετικούς ακεραίους.' )"; //17
      print "</script>"; //18
    } //19
    else if (!is_numeric($arithmos) || $arithmos < 0) { //20
        print "<script type=\"text/javascript\">"; //21
        print "alert('Στο πεδίο \"Αριθμός\" εισάγετε μονο θετικούς αριθμούς.' )"; //22
        print "</script>"; //23
    } //24
else if (mysql_query("INSERT INTO `ΑΠΟΘΗΚΗΓ` //25
(`Kwd_Apothikis`,`Onoma`,`Odos`,`Arithmos`,`Perioxi`,`Poli`) //26
VALUES ('$kwd_ap', '$sonoma', '$odos', '$arithmos', '$perioxi', '$poli')")) //27
{ //28
    print "<script type=\"text/javascript\">"; //29
    print "alert('Η αποθήκη με Κωδικό: $kwd_ap καταχωρήθηκε επιτυχώς!')"; //30
    print "</script>"; //31
    } //32
    else { //33
        echo "mysql_error()"; //34
    }
}

```

Επεξήγηση του κώδικα PHP:

Αρχικά, γίνεται ο βασικός έλεγχος ώστε να εισαχθούν τιμές στα υποχρεωτικά πεδία, τα οποία στην περίπτωση αυτή είναι ο κωδικός αποθήκης και το όνομα της. Αν μείνουν κενά τα πεδία αυτά τότε εμφανίζεται το κατάλληλο μήνυμα (γραμμές 1 έως 4).

Αν στα μη υποχρεωτικά πεδία δεν έχει εισαχθεί κάτι τότε αντί να έχουμε κενά κελία στη βάση μας καταχωρείται ο χαρακτήρας '-' για στήλες που δέχονται αλφαριθμητικά και το '0' (μηδέν) σε αυτές που δέχονται αριθμούς (γραμμές 5 έως 14). Μάλιστα, ελέγχονται και εμφανίζονται κατάλληλα μηνύματα στις περιπτώσεις που εισάγει ο χρήστης αλφαριθμητικούς χαρακτήρες σε πεδία που είναι για αριθμούς (γραμμές 15 έως 24).

Εντέλει, αν σε όλους τους ελέγχους δε υπάρχει σφάλμα φτάνουμε στην καταχώριση του νέου στοιχείου (γραμμές 25 έως 27), με την εμφάνιση μηνύματος αποθήκευσης (γραμμές 28 έως 32) ή την εμφάνιση ποιού MySql σφάλματος συνέβη (γραμμές 32 έως 34). Στις παρακάτω εικόνες

παρουσιάζεται η εισαγωγή νέου στοιχείου. Στις γραμμές 25, 26, 27 βρίσκεται και η εντολή mysql. Δηλαδή, INSERT INTO [όνομα πίνακα] (στήλες πίνακα) VALUES [τιμές που θέλουμε σε συμφωνία με τις στήλες].

Εικόνα 19

Insert ΑΠΟΘΗΚΗΣ - Mozilla Firefox

http://localhost/pt/bin/apothiki_insert.php

Κωδ. Αποθήκης : 18 *

Όνομα : HardwareStore *

Οδός :

Αριθμός : (0 για άγνωστο)

Περιοχή : Estavromenos

Πόλη : Heraklio

Insert!

* Υποχρεωτικό Πεδίο

Εικόνα 20

Insert ΑΠΟΘΗΚΗΣ - Mozilla Firefox

http://localhost/pt/bin/apothiki_insert.php

Κωδ. Αποθήκης : *

Όνομα :

Οδός :

Αριθμός : (στο)

Περιοχή :

Πόλη :

Η αποθήκη με Κωδικό: 18 καταχωρήθηκε επιτυχώς!

OK

Insert!

Μεταφορά δεδομένων από localho...

Μήνυμα επιτυχημένης καταχώρησης

Επισήμανση: Το παραπάνω ισχύει και για προμηθευτές, προϊόντα, υπαλλήλους, υποκαταστήματα.

Εικόνα 21

ΑΠΟΘΗΚΕΣ

Κωδ. Αποθήκης
 Όνομα
 Οδός
 Περιοχή
 Πόλη

Search:

Κωδ. Αποθήκης:	Όνομα:	Οδός:	Αριθμός:	Περιοχή:	Πόλη:	Εργαλεία:
<u>18</u>	HardwareStore	-	0	Estavromenos	Heraklio	<input type="button" value="Edit"/> <input type="button" value="X"/>

Επιβεβαίωση της καταχώρησης στη Βάση Δεδομένων.

4.7. Διαγραφή Στοιχείου

Εικόνα 22

- ΑΡΧΙΚΗ
- ΑΠΟΘΗΚΗ >
- ΠΡΟΙΟΝ >
- ΠΡΟΜΗΘΕΥΤΗΣ >
- ΥΠΟΚΑΤΑΣΤΗΜΑ >
- BACKUP / RESTORE >



ΑΠΟΘΗΚΕΣ

Κωδ. Αποθήκης
 Όνομα
 Οδός
 Περιοχή
 Πόλη

Search:

Κωδ. Αποθήκης:	Όνομα:	Οδός:	Αριθμός:	Περιοχή:	Πόλη:	Εργαλεία:
<u>1</u>	A1	B1	1	C1	D1	<input type="button" value="Edit"/> <input type="button" value="X"/>
<u>2</u>	A2	B2	2	C2	D2	<input type="button" value="Edit"/> <input type="button" value="X"/>
<u>3</u>	A3	B3	7	C3	D3	<input type="button" value="Edit"/> <input type="button" value="X"/>
<u>4</u>	asd	-	0	-	-	<input type="button" value="Edit"/> <input type="button" value="X"/>
<u>5</u>	asdasd	-	0	-	-	<input type="button" value="Edit"/> <input type="button" value="X"/>
<u>7</u>	ddd	-	0	-	-	<input type="button" value="Edit"/> <input type="button" value="X"/>

Στα εργαλεία το κόκκινο κουμπί "X" δίνει τη δυνατότητα διαγραφής μίας καταχώρησης.

Κώδικας PHP για την διαγραφή στοιχείων

```

<?php

$result = mysql_query("SELECT * FROM $table"); //1
$num_rows = mysql_num_rows($result); //2

for ($k=1 ;$k<=$num_rows; $k++) //3
{
if (isset($_POST["X$k"])) //4
{
if (!mysql_query("DELETE FROM $table WHERE `key` = '$kwd[$k]'")) //5
{
echo '<script language="JavaScript" type="text/javascript">alert("Η διαγραφή //6
ΔΕΝ ολοκληρώθηκε!\n\n Error Details: "+" .mysql_error()."</script>'; //7
// echo mysql_error(); //8
}
else {print "<script type='text/javascript'>; //9
print "alert('Η διαγραφή ολοκληρώθηκε!'); //10
print "</script>"; //11
}
}
}
?>

```

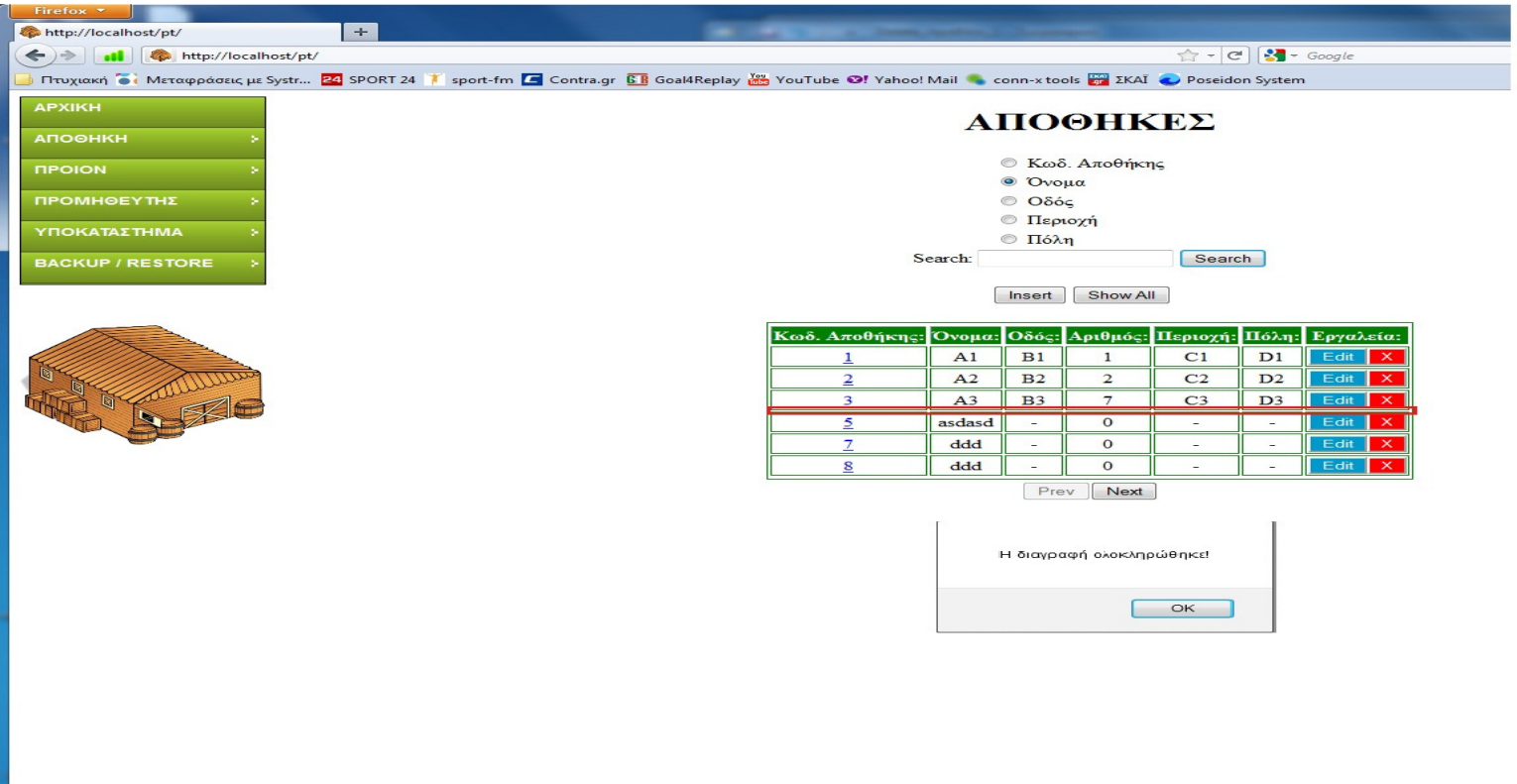
Επεξήγηση του κώδικα PHP:

Κάνουμε ένα query ώστε να μετρηθούν πόσες γραμμές υπάρχουν στο εκάστοτε πίνακα της Β.Δ. μας (γραμμές 1 και 2). Αφού πάρουμε το αποτέλεσμα τότε δημιουργούμε τόσα delete κουμπιά με -τη χρήση μιας for- όσα είναι δηλαδή η καταχωρήσεις στο πίνακα αυτόν (γραμμές 4 και 5).

Έχουμε κατασκευάσει τα κατάλληλα μηνύματα για την αποτυχημένη και την επιτυχημένη διαγραφή. Όταν για κάποιά αιτία αποτύχει μια προσπάθεια για διαγραφή ενός στοιχείου τότε εντοπίζεται και εμφανίζεται το σφάλμα Sql (γραμμές 6 έως 8), ενώ σε επιτυχημένη διαγραφή γίνεται η ενημέρωση στον χρήστη (γραμμές 9 έως 11).

Στην γραμμή 5 φαίνεται με ποιόν τρόπο διαγράφεται μια πλειάδα. Δηλαδή, DELETE FROM [όνομα πίνακα] WHERE [κάποια συνθήκη].

Εικόνα 23

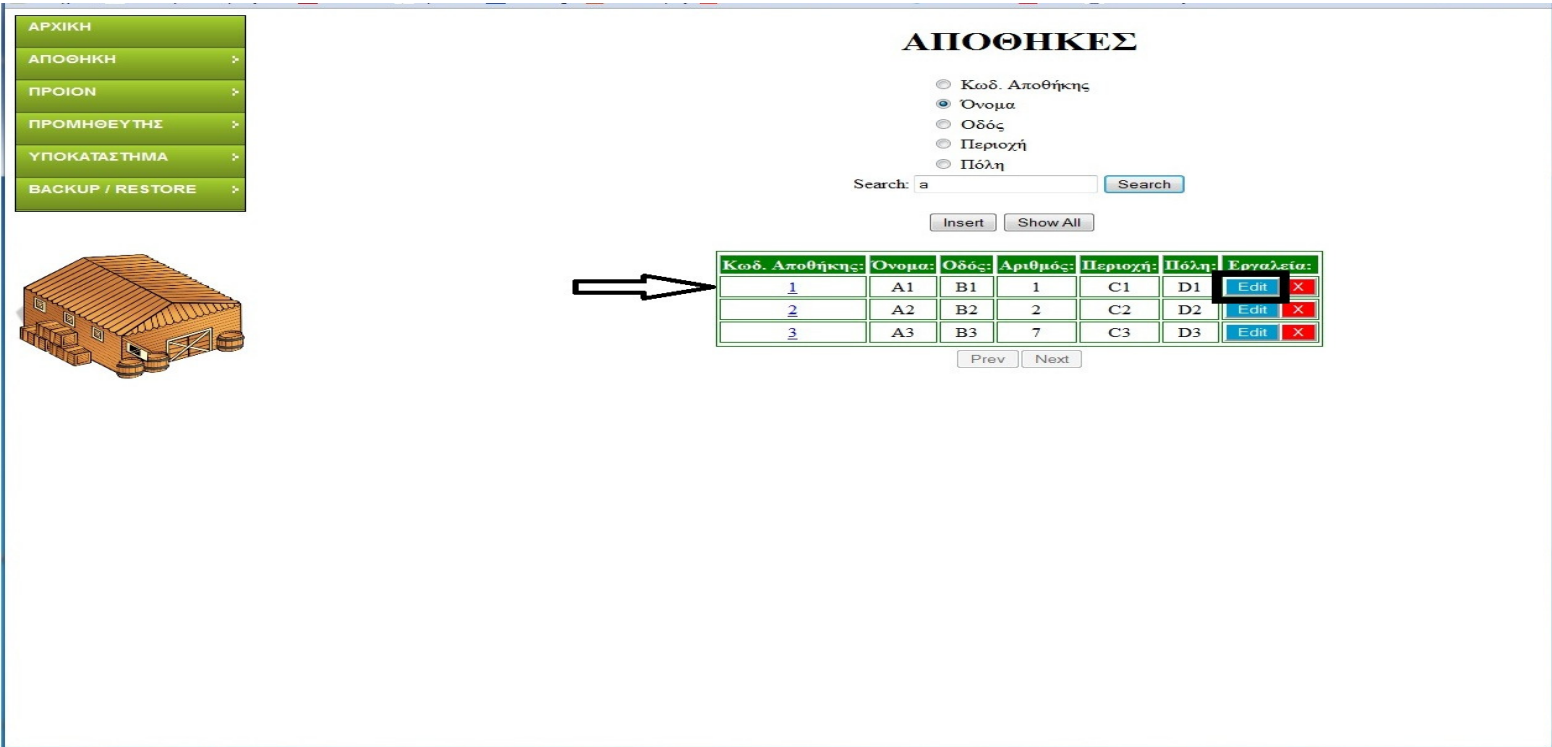


Στην περίπτωση αυτή έγινε επιτυχημένη διαγραφή της αποθήκης με τον κωδικό 4.

Επισήμανση: Το παραπάνω ισχύει και για προμηθευτές, προϊόντα, υπαλλήλους, υποκαταστήματα.

4.8. Επεξεργασία Στοιχείου

Εικόνα 24



Στα εργαλεία το γαλάζιο κουμπί "Edit" δίνει τη δυνατότητα διαγραφής μίας καταχώρησης.

Εικόνα 25

Τώρα μπορούμε να κάνουμε πολλαπλή επεξεργασία σε μια πλειάδα γράφοντας στα *textboxes* –όπου επιτρέπεται και ολοκληρώνοντας πατώντας το κουμπί “OK”. Στην περίπτωση αυτή αλλάζουμε την περιοχή από C1 σε Heraklio.

Κώδικας PHP για την επεξεργασία στοιχείων

```

If($onoma==NULL && $odos==NULL && $arithmos==NULL && $perioxi==NULL) //1
{
    print "<script type='text/javascript'>"; //2
    print "alert('Δεν έχετε συμπληρώσει κανένα πεδίο!');"; //3
    print "</script>"; //4
    $i=0; //5
}
else
    if ($arithmos!=NULL) //6
    {
        if ($arithmos < 0 || !is_numeric($arithmos)) //7
        {
            print "<script type='text/javascript'>"; //8
            print "alert('Παρακαλώ δώστε στον αριθμό ένα θετικό ακέραιο!');"; //9
            print "</script>"; //10
            $i=0;
        }
    }
else mysql_query("update apothiki set arithmos='\"$arithmos\"' where kwd_apothikis=$kwd_ap"); //11
if ($onoma!=NULL) //12
{ mysql_query("update apothiki set onoma='\"$onoma\"' where kwd_apothikis=$kwd_ap"); } //13
if ($odos!=NULL) //14
{ mysql_query("update apothiki set odos='\"$odos\"' where kwd_apothikis=$kwd_ap"); } //15
if ($perioxi!=NULL) //16
{ mysql_query("update apothiki set perioxi='\"$perioxi\"' where kwd_apothikis=$kwd_ap"); } //17
if ($poli!=NULL) //18
{ mysql_query("update apothiki set poli='\"$poli\"' where kwd_apothikis=$kwd_ap"); } //19
}
    
```


Επεξήγηση του κώδικα PHP

Στις γραμμές 1, 2, 3 γίνεται ο έλεγχος για το αν όλα τα πεδία γίνουν κενά να μην ολοκληρώνεται η ανανέωση και να ενημερώνεται ο χρήστης με το κατάλληλο μήνυμα. Επίσης, ελέγχεται η τιμή για το πεδίο αριθμός να είναι αριθμητική και μεγαλύτερη του μηδενός (6-10). Στις επόμενες 9 γραμμές εκτελούνται όλες οι ανανεώσεις για κάθε ένα από τα πεδία ξεχωριστά.

Στην περίπτωση κατά την οποία είτε λόγω λάθους του χρήστη είτε ηθελημένα μείνει κενό ένα ή μερικά πεδία τότε αντί το κελί να μείνει άδειο στη θέση του καταλαμβάνει ο χαρακτήρας της παύλας '-'. Αυτό προτιμάτε κατά κοινή ομολογία για την μεγαλύτερη διευκόλυνση σε μετέπειτα λειτουργίες όπως η αναζήτηση κενών κελιών θα γίνεται με την '-' και έτσι μένει ελεύθερη η αναζήτηση δίχως να γράφει κάτι ο χρήστης το οποίο επιστρέφει ως αποτέλεσμα όλες τις πλειάδες ανεξαρτήτως. -σ.σ. Ο κώδικας για το παραπάνω κομμάτι βρίσκεται στο παράρτημα-.

Σε αυτά τα σημεία βρίσκεται και η εντολή mysql με τον PHP κώδικα για την ανανέωση. UPDATE [όνομα πίνακα] SET (βάζουμε τιμή σε στήλη) WHERE [κάποια συνθήκη].

Εικόνα 26

The screenshot shows a web browser window displaying a PHP application interface. The browser's address bar shows 'http://localhost/pt/'. The page title is 'ΑΠΟΘΗΚΕΣ'. On the left, there is a sidebar with green buttons for navigation: 'ΑΡΧΙΚΗ', 'ΑΠΟΘΗΚΗ', 'ΠΡΟΙΟΝ', 'ΠΡΟΜΗΘΕΥΤΗΣ', 'ΥΠΟΚΑΤΑΣΤΗΜΑ', and 'BACKUP / RESTORE'. The main content area features a search form with radio buttons for 'Κωδ. Αποθήκης', 'Όνομα', 'Οδός', 'Περιοχή', and 'Πόλη'. Below the search form are buttons for 'Insert' and 'Show All'. A table with the following columns is displayed: 'Κωδ. Αποθήκης', 'Όνομα', 'Οδός', 'Αριθμός', 'Περιοχή', 'Πόλη', and 'Εργασία:'. The table contains three rows of data. The first row is highlighted, and the 'Περιοχή' cell is highlighted in red, indicating a change from 'C1' to 'Heraklio'. Below the table are 'Prev' and 'Next' buttons.

Κωδ. Αποθήκης	Όνομα	Οδός	Αριθμός	Περιοχή	Πόλη	Εργασία:
<u>1</u>	A1	B1	1	Heraklio	D1	Edit X
<u>2</u>	A2	B2	2	C2	D2	Edit X
<u>3</u>	A3	B3	7	C3	D3	Edit X

Επιβεβαίωση της αλλαγής σε περιοχή από 'C1' σε 'Heraklio'

Επισήμανση: Το παραπάνω ισχύει και για προμηθευτές, προϊόντα, υπαλλήλους, υποκαταστήματα.

4.9. Αριθμοί Τηλεφώνων.

Εικόνα27

ΤΗΛΕΦΩΝΑ ΥΠΑΛΛΗΛΩΝ

Α.Δ.Τ.: Τηλέφωνο:

Επέλεξε:

Α.Δ.Τ. Τηλέφωνο

Search:

Α.Δ.Τ. Υπάλληλου:	Επώνυμο	Όνομα	Τηλέφωνο:	Εργασία:
E-789456	A3	B3	1234567890	<input type="button" value="Edit"/> <input type="button" value="X"/>
F-123456	A1	B1	1234567891	<input type="button" value="Edit"/> <input type="button" value="X"/>
F-123456	A1	B1	1234567892	<input type="button" value="Edit"/> <input type="button" value="X"/>
F-123456	A1	B1	1234567893	<input type="button" value="Edit"/> <input type="button" value="X"/>
E-789456	A3	B3	1234567894	<input type="button" value="Edit"/> <input type="button" value="X"/>
E-789456	A3	B3	1234567895	<input type="button" value="Edit"/> <input type="button" value="X"/>

Εικόνα 28

ΤΗΛΕΦΩΝΑ ΥΠΑΛΛΗΛΩΝ

Α.Δ.Τ.: Τηλέφωνο:

F-123456 2822

Α.Δ.Τ. Τηλέφωνο

Search:

Λάθος στο πεδίο τηλεφώνου!

Από τις παραπάνω εικόνες επιβεβαιώνεται ότι υπάρχει η δυνατότητα για ένα υπάλληλο να διαθέτει πολλούς αριθμούς τηλεφώνου. Με την χρήση ενός drop down menu (κόκκινο πλαίσιο στην

πρώτη εικόνα) ο διαχειριστής μπορεί να επιλέξει άμεσα από τους υπαλλήλους δίχως να πρέπει να γράφει κάθε φορά τον αριθμό της αστυνομικής ταυτότητας. Τα ίδια ισχύουν και για τα τηλέφωνα προμηθευτών.

Κώδικας PHP για την καταχώρηση τηλεφώνων

```
<?php
if ($adt=='0' || $til==NULL) //1
{
    print "<script type=\"text/javascript\">"; //2
    print "alert(Κενό κάποιο υποχρεωτικό πεδίο!);"; //3
    print "</script>"; //4
}
else if (!is_numeric($til) || $til < 0 || strlen($til) != 10) //5
{
    print "<script type=\"text/javascript\">"; //6
    print "alert('Λάθος στο πεδίο τηλεφώνου!');"; //7
    print "</script>"; //8
}
else if (mysql_query("insert into til_ypalliloy(`A.D.T_Ypalliloy`, `til_ypalliloy`) values ( '$adt' , '$til'"))) //9
{
    print "<script type=\"text/javascript\">"; //10
    print "alert('Η καταχώρηση έγινε επιτυχώς!');"; //11
    print "</script>"; //12
}
else //13
{
    print "<script type=\"text/javascript\">"; //14
    print "alert('echo mysql_error()');"; //15
    print "</script>"; //16
}
?>
```

Επεξήγηση του κώδικα PHP

Στις πρώτες τέσσερις γραμμές γίνεται έλεγχος για το αν κάποιο από το drop down menu όπου επιλέγεται ο κωδικός των υπαλλήλων (ή προμηθευτών) ή από το πεδίο τηλεφώνου είναι κενό. Αν ισχύει τότε στον χρήστη εμφανίζεται το μήνυμα: «Κενό κάποιο υποχρεωτικό πεδίο». Στις επόμενες τέσσερις γραμμές εξετάζεται η περίπτωση στο πεδίο τηλεφώνου να μην είναι «καθαρά» αριθμητικό ή να μην έχει ακριβώς δέκα ψηφία, όταν συμβαίνει τέτοιο σφάλμα ανοίγει ένα alert παράθυρο με τη φράση: «Λάθος στο πεδίο Τηλεφώνου».

Από την ένατη έως την δωδέκατη σειρά εκτελείται η εντολή εισαγωγής ενός νέου στοιχείου στον πίνακα της βάσης Τηλέφωνα _ Υπαλλήλων και κατόπιν μέσω ενός alert γίνεται ενημέρωση για την επιτυχία της. Από την 13 – 16 καλύπτονται με μηνύματα όλες οι άλλες πιθανότητες αποτυχίας μιας προσπάθειας εισαγωγής και εμφανίζονται αναλυτικά στον διαχειριστή. Υπενθυμίζουμε πως το τηλέφωνο πήρε αυτή την σχεσιακή μορφή επειδή είναι πλειότιμο γνώρισμα.

Επισήμανση: Το παραπάνω ισχύει και για τα τηλέφωνα προμηθευτών.

4.10. Εισαγωγή στοιχείου από συσχέτιση μεταξύ πινάκων.

Σε αντίθεση με την απλή εισαγωγή στοιχείου στην οποία έχουμε ένα πρωτεύον κλειδί, όπως είδαμε παραπάνω, όταν χρειαστεί να περαστεί μία καταχώρηση στην οποία υπάρχουν δύο πρωτεύοντα κλειδιά απαιτήθηκαν μερικές διαφοροποιήσεις κατά τον τρόπο υλοποίησης της λειτουργίας σε PHP. Για να επιτευχθεί η βέλτιστη δυνατή διασύνδεση του λογισμικού με τον χρήστη –σ.σ. σε συμφωνία με τις κοινές αρχές του Graphical User Interface- επιλέχτηκε το pop up window που ανοίγεται για τις περιπτώσεις τέτοιου τύπου να έχει δυναμικότερο χαρακτήρα σε σύγκριση με τη σελίδα της απλής εισαγωγή στοιχείων.

Αρχικά, όπως βλέπουμε στη εικόνα 31 το νέο παράθυρο που προβάλλει υπάρχει μονάχα η επιλογή του υποκαταστήματος και αφότου ο χρήστης διαλέξει ένα απ' αυτά τότε αυτόματα του παρουσιάζεται ένα καινούργιο drop down menu -εικόνα 32-. Αν θυμηθούμε από τις απαιτήσεις το κάθε υποκατάστημα έχει μία αποθήκη που το εφοδιάζει συνεπώς το προϊόντα που δύναται να υπάρχουν σε αυτό είναι οπωσδήποτε αυστηρά ορισμένα. Με τον τρόπο αυτό ελαχιστοποιούνται λάθη που πιθανόν να συμβούν από ανθρώπινο παράγοντα, όπως επίσης γίνεται ταχύτερα η εργασία καθώς ο διαχειριστής δεν πρέπει να ελέγχει αν είναι δυνατές οι ενέργειες που πραγματοποιοει.

Εικόνα 29

The screenshot shows a web browser window with the URL 'http://localhost/pt/'. The page title is 'ΥΠΟΚΑΤΑΣΤΗΜΑΤΑ'. On the left is a green sidebar with menu items: ΑΡΧΙΚΗ, ΑΠΟΘΗΚΗ, ΠΡΟΙΟΝ, ΠΡΟΜΗΘΕΥΤΗΣ, ΥΠΟΚΑΤΑΣΤΗΜΑ, and BACKUP / RESTORE. The main content area has a search section with radio buttons for 'Κωδ. Υποκαταστήματος', 'Όνομα', 'Οδός', 'Περιοχή', 'Πόλη', and 'Κωδ. Αποθήκης'. Below the search is a 'Search' button and 'Insert'/'Show All' buttons. A table displays the following data:

Κωδ. Υποκαταστήματος	Όνομα	Οδός	Αριθμός	Περιοχή	Πόλη	Κωδ. Αποθήκης	Εργασία
136	AA1	BB1	54	CC15	DD1	2	Edit X
647	AA2	BB2	16	CC2	DD2	3	Edit X

At the bottom of the table are 'Prev' and 'Next' buttons. A small illustration of a warehouse is located in the bottom left corner of the page.

Το υποκατάστημα 'AA1' με id: '136' εφοδιάζεται μονό από την αποθήκη με id: '2'.

Εικόνα 30

Κωδ. Προϊόντος	Όνομα Προϊόντος	Κωδ. Αποθήκης	Όνομα Αποθήκης	Απόθεμα	Ελάχιστο Απόθεμα	Εργασία
1	A1	2	A2	6	3	Edit X
4	A1	2	A2	50	2	Edit X
5	B1	2	A2	2500	100	Edit X

 At the bottom of the table are 'Prev' and 'Next' buttons. On the left side of the page, there is a small illustration of a wooden warehouse building."/>

Στην αποθήκη με id: '2' υπάρχουν μόνο τα προϊόντα με id:[1,4,5]

Εικόνα 31

Αρχική μορφή του παραθύρου pop up μετά την επιλογή για την εισαγωγή στοιχείου.

Εικόνα 32

Insert Υποκαταστήματος - Mozilla Firefox
http://localhost/pt/bin/apothema_insert_2.php

Κωδ. Υποκαταστήματος:	Κωδ. Προϊόντος:
136	Επέλεξε:
Απόθεμα: *	Επέλεξε:
<input type="text"/>	1
	4
	5
	Απόθεμα: *
<input type="text"/>	<input type="text"/>

Insert! *Υποχρεωτικό Πεδίο

Επιβεβαίωση ότι για το υποκατάστημα '136' μπορούν να υπάρχουν μόνο τα προϊόντα με id:[1,4,5]

Εικόνα 33

Insert Υποκαταστήματος - Mozilla Firefox
http://localhost/pt/bin/apothema_insert_2.php

Κωδ. Υποκαταστήματος:	Κωδ. Προϊόντος:
136	5
Απόθεμα: *	Ελάχιστο Απόθεμα: *
1500	250

Insert! *Υποχρεωτικό Πεδίο

Εισαγωγή δεδομένων εγγραφής.

Εικόνα 34

Επιβεβαίωση της παραπάνω εγγραφής.

Κώδικας PHP στο Παράρτημα Α

Επισήμανση: Το παραπάνω ισχύει και για τα αποθέματα αποθηκών.

4.11. Ειδικές Λειτουργίες

Εικόνα 35

Αρχικά, γίνεται μια αναζήτηση και επιλέγεται για αυτή την καταχώρηση η επιλογή Edit (στα εργαλεία). Αν μειωθεί το απόθεμα του προϊόντος κάτω από το ελάχιστο που έχουμε ορίσει για κάθε ένα τότε μπορούν να συμβούν τα παρακάτω σενάρια.

4.11.1. Σενάριο: Επαρκές Απόθεμα Αποθήκης για μεταφορά Προϊόντων σε Υποκατάστημα.

Υπάρχει επαρκές πλήθος τεμαχίων συγκεκριμένου προϊόντος στην Αποθήκη κάποιου υποκαταστήματος για την άμεση μεταφορά αυτών από Αποθήκη στο Υποκατάστημα. Στην παρακάτω περίπτωση συμβαίνουν κατά σειρά τα εξής. Αρχικά, το υποκατάστημα με κωδικό: '136' και με απόθεμα του προϊόντος με κωδικό '1' που είναι 8 τεμάχια, ζητάει απ' την αποθήκη που το εφοδιάζει τέσσερα επιπλέον τεμάχια. Η αποθήκη με id '2' που είναι υπεύθυνη για το υποκατάστημα 136 ελέγχει αν έχει για το προϊόν '1' τόσα κομμάτια για να το εφοδιάσει. Εφόσον πράγματι έχει, τότε τα αποστέλλει και ενημερώνεται η αύξηση που γίνεται στο υποκατάστημα και η αντίστοιχη μείωση στην αποθήκη. Ακολουθεί το παράδειγμα με εικόνες για του λόγου το αληθές.

Συμπερασματικά, μπορούμε να θεωρήσουμε αυτόν τον αυτοματισμό ως μια ηλεκτρονική αντικατάσταση της συναλλαγής μεταξύ του αποθηκάριου και του υπεύθυνου των αποθεμάτων στα καταστήματα. Μόλις παρουσιάζεται έλλειψη σε κάποιο προϊόν αναλαμβάνει ο μηχανισμός αυτός ώστε να καλυφθούν απευθείας οι ανάγκες όπου αυτές συμβαίνουν.

Εικόνα 36

The screenshot shows a web browser window displaying a management interface. The main heading is "ΑΠΟΘΕΜΑ ΥΠΟΚΑΤΑΣΤΗΜΑΤΩΝ". There are search filters for "Κωδ. Προϊόντος" (A1) and "Κωδ. Υποκαταστήματος" (136). A search button is present. Below the search area is a table with the following data:

Κωδ. Προϊόντος	Όνομα Προϊόντος	Κωδ. Υποκαταστήματος	Όνομα Υποκαταστήματος	Απόθεμα	Ελάχιστο Απόθεμα	Εργασία
1	A1	136	AA1	8	3	Edit X

A red arrow points to the first row of the table. The interface also includes a sidebar with menu items like "ΑΡΧΙΚΗ", "ΑΠΟΘΗΚΗ", "ΠΡΟΙΟΝ", "ΠΡΟΜΗΘΕΥΤΗΣ", "ΥΠΟΚΑΤΑΣΤΗΜΑ", and "BACKUP / RESTORE".

Κατάσταση υποκαταστήματος με κωδικό '136' πριν τη μεταφορά, για το προϊόν '1' με απόθεμα '8' τεμάχια και ελάχιστο απόθεμα '3'.

Εικόνα 37

ΑΠΟΘΕΜΑ ΑΠΟΘΗΚΩΝ

Κωδ. Προϊόντος Κωδ. Αποθήκης Απόθεμα: = > <

 Όνομα Προϊόντος Όνομα Αποθήκης

Κωδ. Προϊόντος	Όνομα Προϊόντος	Κωδ. Αποθήκης	Όνομα Αποθήκης	Απόθεμα	Ελάχιστο Απόθεμα	Εργαλεία
1	A1	1	A1	0	0	Edit X
1	A1	2	A2	17	3	Edit X
1	A1	5	asdasd	15	4	Edit X
2	A2	1	A1	227	0	Edit X
2	A2	10	ddd	123	0	Edit X
3	A3	7	ddd	0	0	Edit X

Κατάσταση αποθήκης με κωδικό '2' πριν τη μεταφορά, για το προϊόν '1' με αποθεματικό '17' τεμάχια και ελάχιστο απόθεμα '3'.

Εικόνα 38

ΑΠΟΘΕΜΑ ΥΠΟΚΑΤΑΣΤΗΜΑΤΩΝ

Κωδ. Προϊόντος Κωδ. Υποκαταστήματος Απόθεμα: = > <

 Όνομα Προϊόντος Όνομα Υποκαταστήματος

Μεταφορά Προϊόντος: Προϊόν: "A1", Ποσότητα: "4", Από Αποθήκη: "A2".

Μήνυμα ενημέρωσης: Μεταφορά Προϊόντος [όνομα προϊόντος], Ποσότητα [πλήθος τεμαχίων] Από Αποθήκη [όνομα αποθήκης].

Εικόνα 39

ΑΠΟΘΕΜΑ ΑΠΟΘΗΚΩΝ

Κωδ. Προϊόντος
 Κωδ. Αποθήκης
 Απόθεμα:
 Όνομα Προϊόντος
 Όνομα Αποθήκης

Search Insert Show All

Κωδ. Προϊόντος	Όνομα Προϊόντος	Κωδ. Αποθήκης	Όνομα Αποθήκης	Απόθεμα	Ελάχιστο Απόθεμα	Εργασία
1	A1	1	A1	0	0	Edit X
1	A1	2	A2	13	3	Edit X
1	A1	5	asdasd	15	4	Edit X
2	A2	1	A1	227	0	Edit X
2	A2	10	ddd	123	0	Edit X
3	A3	7	ddd	0	0	Edit X

Prev Next

Κατάσταση αποθήκης μετά τη μεταφορά απόθεμα προϊόντος 'A1' έγινε 13 (δλδ. $17 - 4 = 13$).

Εικόνα 40

ΑΠΟΘΕΜΑ ΥΠΟΚΑΤΑΣΤΗΜΑΤΩΝ

Κωδ. Προϊόντος
 Κωδ. Υποκαταστήματος
 Απόθεμα:
 Όνομα Προϊόντος
 Όνομα Υποκαταστήματος

Search Insert Show All

Κωδ. Προϊόντος	Όνομα Προϊόντος	Κωδ. Υποκαταστήματος	Όνομα Υποκαταστήματος	Απόθεμα	Ελάχιστο Απόθεμα	Εργασία
1	A1	136	K5	12	3	Edit X

Prev Next

Κατάσταση υποκαταστήματος μετά τη μεταφορά απόθεμα προϊόντος 'A1' έγινε 12 (δλδ. $8 + 4 = 12$).

Κώδικας PHP στο Παράρτημα Α

4.11.2. Σενάριο: Μη επαρκές Απόθεμα Αποθήκης για μεταφορά προϊόντος σε Υποκατάστημα

Ένα δεν υπάρχει επαρκές πλήθος τεμαχίων για κάποιο προϊόν ώστε να γίνει άμεση μεταφορά από αποθήκη σε υποκατάστημα προκειμένου να καλυφθεί η απαίτηση που παρουσιάστηκε. Στο παράδειγμα που θα δείτε παρακάτω, όταν φτάσει η απαίτηση του υποκαταστήματος '136' στην αποθήκη '2' που το εφοδιάζει για ένα πλήθος προϊόντων με id '1' και η αποθήκη δεν δύναται να την εξυπηρετήσει άμεσα. Τότε, γίνεται αναζήτηση ανάμεσα στους προμηθευτές της αποθήκης που παρέχουν αυτό το προϊόν ώστε να βρεθεί η καλύτερη προσφορά και επιλέγεται αυτή (συγκεκριμένα για το παράδειγμά μας ο προμηθευτής με id: '2', όνομα: 'A2').

Πρέπει να εξετάσουμε τι θα συμβεί σε αποθήκη και σε υποκατάστημα όσο αφορά το αποθεματικό για το εν λόγω προϊόν. Με το αυτόματο σύστημα παραγγελίας αγοράζονται τόσα τεμάχια ώστε να εξυπηρετηθούν η ανάγκες του υποκαταστήματος συμφωνά με την απαίτηση του, αλλά και το απόθεμα στην αποθήκη να είναι πλέον τουλάχιστον το διπλάσιο από το ελάχιστο απόθεμα που έχει ορίσει ο διαχειριστής. Εδώ ελάχιστο απόθεμα τρία άρα το απόθεμα πρέπει να γίνει τουλάχιστον 6, ενώ το υποκατάστημα θα έχει δώδεκα.

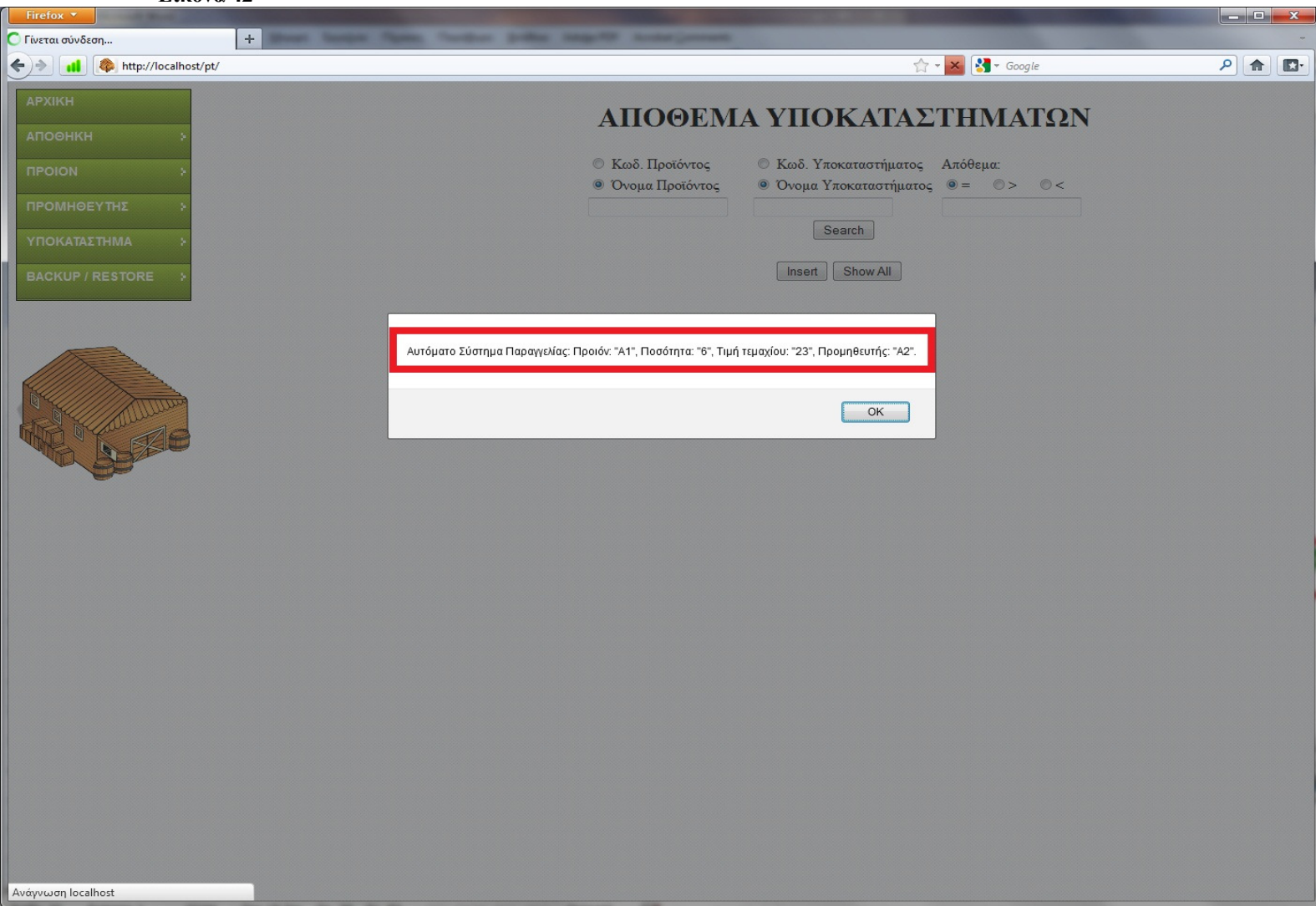
Ο αυτοματισμός αυτός είναι μία εξομοίωση ενός marketing manager σε μία επιχείρηση αλλά λειτουργεί ορθά μόνο αν οι διαχειριστές της βάσης δεδομένων ανανεώνουν έγκαιρα τις τιμές που προσφέρονται από τους προμηθευτές. Η εμφάνιση είναι: Αυτόματο Σύστημα Παραγγελίας, Προϊόν [όνομα προϊόντος], Ποσότητα [# τεμαχίων], Τιμή τεμαχίου [σε μονάδα νομίσματος], Προμηθευτής [όνομα προμηθευτή].

Εικόνα 41

Κωδ. Προϊόντος	Όνομα Προϊόντος	Κωδ. Αποθήκης	Όνομα Αποθήκης	Απόθεμα	Ελάχιστο Απόθεμα	Εργασία:
1	A1	2	A2	4	3	Edit X

Κατάσταση αποθήκης πριν την απαίτηση του υποκαταστήματος. - Απόθεμα '4', Ελάχιστο Απόθεμα '3'.

Εικόνα 42



Μήνυμα: Αυτόματο Σύστημα Παραγγελίας, Προϊόν 'A1', Ποσότητα '6', Τιμή τεμαχίου '23', Προμηθευτής 'A2'. Όπως βλέπουμε στην ενημέρωση τα τεμάχια που αγοράστηκαν από τον προμηθευτή είναι έξι καθώς πρέπει να πάρει το υποκατάστημα τα τέσσερα που ζήτησε κατευθείαν από την αποθήκη, αλλά και η αποθήκη πρέπει να έχει απόθεμα προϊόντος διπλάσιο του ελάχιστου.

Εικόνα 43

ΑΠΟΘΕΜΑ ΑΠΟΘΗΚΩΝ

Κωδ. Προϊόντος
 Κωδ. Αποθήκης
 Απόθεμα:
 =
 >
 <

Όνομα Προϊόντος
 Όνομα Αποθήκης

Κωδ. Προϊόντος	Όνομα Προϊόντος	Κωδ. Αποθήκης	Όνομα Αποθήκης	Απόθεμα	Ελάχιστο Απόθεμα	Εργασία:
1	A1	2	A2	6	3	Edit X

Κατάσταση αποθήκης μετά την ολοκλήρωση του αυτοματισμού – Απόθεμα '6'.

Εικόνα 44

ΑΠΟΘΕΜΑ ΥΠΟΚΑΤΑΣΤΗΜΑΤΩΝ

Κωδ. Προϊόντος
 Κωδ. Υποκαταστήματος
 Απόθεμα:
 =
 >
 <

Όνομα Προϊόντος
 Όνομα Υποκαταστήματος

Κωδ. Προϊόντος	Όνομα Προϊόντος	Κωδ. Υποκαταστήματος	Όνομα Υποκαταστήματος	Απόθεμα	Ελάχιστο Απόθεμα	Εργασία:
1	A1	136	AA1	12	3	Edit X

Κατάσταση υποκαταστήματος μετά την ολοκλήρωση του αυτοματισμού – Απόθεμα '12'.

Εικόνα 45

ΠΡΟΙΟΝΤΑ ΠΡΟΜΗΘΕΥΤΩΝ

Κωδ. Προϊόντος
 Κωδ. Προμηθευτή
 Τιμή:
 Ονομα Προϊόντος
 Επώνυμο Προμηθευτή
 =
 >
 <

Search Insert Show All

Κωδ. Προϊόντος	Όνομα Προϊόντος	Κωδ. Προμηθευτή	Επώνυμο Προμηθευτή	Τιμή	Εργαλεία
1	A1	1	B2	32	Edit X
7	oo	1	B2	32	Edit X
1	A1	2	A2	23	Edit X
3	A3	2	A2	23	Edit X
6	B2	2	A2	22	Edit X
7	oo	2	A2	25	Edit X

Prev Next

Επιβεβαίωση ότι πράγματι αγοράστηκαν τα προϊόντα από τον προμηθευτή με την καλύτερη προσφορά. Για το ίδιο προϊόν ο προμηθευτής 'A2' προσφέρει τιμή 23 ενώ ο 'B2' 32.

Κώδικας PHP στο Παράρτημα Α

4.11.3. Σενάριο: Απόθεμα Προϊόντος σε Αποθήκη κάτω του ελάχιστου ορίου.

Στο σενάριο αυτό θα δούμε πως λειτουργεί ο αυτοματισμός -σ.σ. μονάχα στην πλευρά της αποθήκης- όταν σε κάποιο προϊόν της ο αριθμός των τεμαχίων πάρει τιμές κάτω του ελάχιστου αποθέματος όπως έχει οριστεί απ' τους διαχειριστές του συστήματος. Για να τον δούμε πρακτικά τον αυτοματισμό θα αναλύσουμε τι ακριβώς συμβαίνει στις παρακάτω εικόνες. Στην αποθήκη 'A2' επιλέγουμε από τα εργαλεία το κουμπί της επεξεργασίας (edit) και χαμηλώνουμε το απόθεμα του προϊόντος 'A1' μέχρι την τιμή δύο ακριβώς δηλαδή ένα τεμάχιο λιγότερο του ελάχιστο αποθέματος. Στην οθόνη τώρα αναδύεται η ειδοποίηση ότι έχει καλεστεί άμεσα η διαδικασία αυτόματης παραγγελίας και έχει αναζητήσει τον προμηθευτή με την χαμηλότερη τιμή για το ίδιο προϊόν, απ' τον οποίο και παραγγέλλει τόσα τεμάχια ώστε στην αποθήκη πλέον να υπάρχουν τα διπλάσια τεμάχια για το συγκεκριμένο προϊόν στο απόθεμα.

Με τον παρόν μηχανισμό με την αμεσότητα που εκτελείται επιτρέπει την κάλυψη απαιτήσεων που παρουσιάζονται στα αποθέματα που έχει κάθε αποθήκη για οποιαδήποτε προϊόν ταχύτατα. Ο αυτοματισμός διαλέγει την βέλτιστη προσφορά και αγοράζει τόσο τεμάχια όσα χρειάζονται σύμφωνα με το σύστημα.

Εικόνα 46

ΑΠΟΘΕΜΑ ΑΠΟΘΗΚΩΝ

Κωδ. Προϊόντος Κωδ. Αποθήκης Απόθεμα: = > <

Ονομα Προϊόντος Ονομα Αποθήκης

Search Insert Show All

Κωδ. Προϊόντος	Όνομα Προϊόντος	Κωδ. Αποθήκης	Όνομα Αποθήκης	Απόθεμα	Ελάχιστο Απόθεμα	Εργασία
1	A1	1	A1	20	5	Edit X
1	A1	2	A2	17	3	Edit X
1	A1	5	asdasd	15	4	Edit X
2	A2	1	A1	227	2	Edit X
2	A2	10	ddd	123	2	Edit X
3	A3	7	ddd	15000	50	Edit X

Prev Next

Έστω ότι επιλέγεται η Αποθήκη 'A1' με το προϊόν 'A2' και μετά μέσω τις επιλογής edit βάζουμε το απόθεμα της 2 (δύο) που είναι κάτω απ' το ελάχιστο απόθεμα.

Εικόνα 47

ΑΠΟΘΕΜΑ ΑΠΟΘΗΚΩΝ

Κωδ. Προϊόντος Κωδ. Αποθήκης Απόθεμα: = > <

Ονομα Προϊόντος Ονομα Αποθήκης

Search Insert Show All

Αυτόματο Σύστημα Παραγγελίας: Προϊόν: "A1", Τιμή τεμαχίου: "23", Προμηθευτής: "A2".

OK

Τώρα εκτελείται η δυνατότητα αυτόματης παραγγελίας και βλέπουμε ότι η αποθήκη 'A1' επιλέγει για το Προϊόν 'A2' τον προμηθευτή με την καλύτερη προσφορά. Μήνυμα: "Αυτόματο Σύστημα Παραγγελίας: Προϊόν A1, Τιμή τεμαχίου 23, Προμηθευτής A2".

Εικόνα 48

ΠΡΟΙΟΝΤΑ ΠΡΟΜΗΘΕΥΤΩΝ

Κωδ. Προϊόντος Κωδ. Προμηθευτή Τιμή:

 Όνομα Προϊόντος Επώνυμο Προμηθευτή

Search Insert Show All

Κωδ. Προϊόντος	Όνομα Προϊόντος	Κωδ. Προμηθευτή	Επώνυμο Προμηθευτή	Τιμή	Εργαλεία
1	A1	1	B2	32	Edit X
7	oo	1	B2	25	Edit X
1	A1	2	A2	23	Edit X
3	A3	2	A2	23	Edit X
6	B2	2	A2	22	Edit X
7	oo	2	A2	25	Edit X

Prev Next

Για επιβεβαίωση πως δουλεύει σωστά ο μηχανισμός. Ο προμηθευτής A2 προσφέρει τιμή 23 ενώ ο B2 τιμή 32.

Εικόνα 49

ΑΠΟΘΕΜΑ ΑΠΟΘΗΚΩΝ

Κωδ. Προϊόντος Κωδ. Αποθήκης Απόθεμα:

 Όνομα Προϊόντος Όνομα Αποθήκης

Search Insert Show All

Κωδ. Προϊόντος	Όνομα Προϊόντος	Κωδ. Αποθήκης	Όνομα Αποθήκης	Απόθεμα	Ελάχιστο Απόθεμα	Εργαλεία
1	A1	1	A1	20	5	Edit X
1	A1	2	A2	6	3	Edit X
1	A1	5	asdasd	15	4	Edit X
2	A2	1	A1	227	2	Edit X
2	A2	10	ddd	123	2	Edit X
3	A3	7	ddd	15000	50	Edit X

Prev Next

Και επίσης η αποθήκη έχει πλέον απόθεμα γι' αυτό το προϊόν το διπλάσιο του ελάχιστο όπως ζητήθηκε στις απαιτήσεις (specs).

Κώδικας PHP στο Παράρτημα Α

4.11.4. Αποθέματα Υποκαταστήματος και Αποθήκης κάτω του ελάχιστου.

Σε αυτό το σενάριο θα δειχθεί τι ακριβώς συμβαίνει όταν τα αποθέματα για ένα προϊόν τόσο στο υποκατάστημα όσο και στην αποθήκη φτάσουν να βρίσκονται κάτω απ' το ελάχιστο επιτρεπτό όριο. Στο παράδειγμα που ακολουθεί στην υποκατάστημα υπάρχουν ακριβώς πάνω από ένα τεμάχια για το προϊόν με κωδικό '6' απ' το ελάχιστο απόθεμα και το ίδιο ισχύει και για την αποθήκη.

Στην συνέχεια ένα τεμάχιο του προϊόντος αυτού στο υποκατάστημα πωλείται συνεπώς ο αριθμός των τεμαχίων γίνεται μικρότερος του ορίου που έχει θέσει ο διαχειριστής. Όπως έχουμε δει και στις προηγούμενες περιπτώσεις αυτό έχει ως συνεπεία την κλήση, από την μεριά του υποκαταστήματος, της διεργασίας μεταφοράς του προϊόντος –με id: '6', όνομα: 'B2'- από την αποθήκη που το εφοδιάζει -με id: '3', όνομα: 'A3'-. Στην περίπτωση αυτή όμως επίσης και το απόθεμα στην αποθήκη αυτού το προϊόντος καταλήγει σε τιμή ξανά κατά ένα τεμάχιο μικρότερη του ελάχιστου αποδεκτού αποθέματος. Επομένως θα πρέπει τώρα να καλεστεί από την μεριά της αποθήκης η διεργασία της αυτοματοποιημένης βέλτιστης παραγγελίας με ποσότητα με την οποία θα πρέπει να καλύπτει το διπλάσιο απόθεμα και στην ίδια αλλά και στο υποκατάστημα – με id : '647' όνομα: 'AA2' -.

Το σενάριο αυτό είναι μια επέκταση των σεναρίων 1.1, 1.2. Είναι μια αλυσιδωτή αντίδραση κατά την οποία, αρχικά λαμβάνει χώρα η προσπάθεια για μεταφορά προϊόντων στο υποκατάστημα από την αποθήκη. Μόλις αυτή αποτυγχάνει το σύστημα προχωράει στην αποθήκη και παραγγέλνει από τον προμηθευτή, που επιλέγει με τον τρόπο που αναφέραμε παραπάνω, ακριβές αριθμό τεμαχίων ώστε να καλύψει τις ανάγκες και των δύο. Αφού στην αποθήκη μείνουν τόσα τεμάχια, όσα δηλαδή απαιτεί το σύστημα τα υπόλοιπα προωθούνται στο υποκατάστημα και καταλήγουμε στην εξ ολοκλήρου εξυπηρέτηση των απαιτήσεων που εμφανίστηκαν.

Εικόνα 50

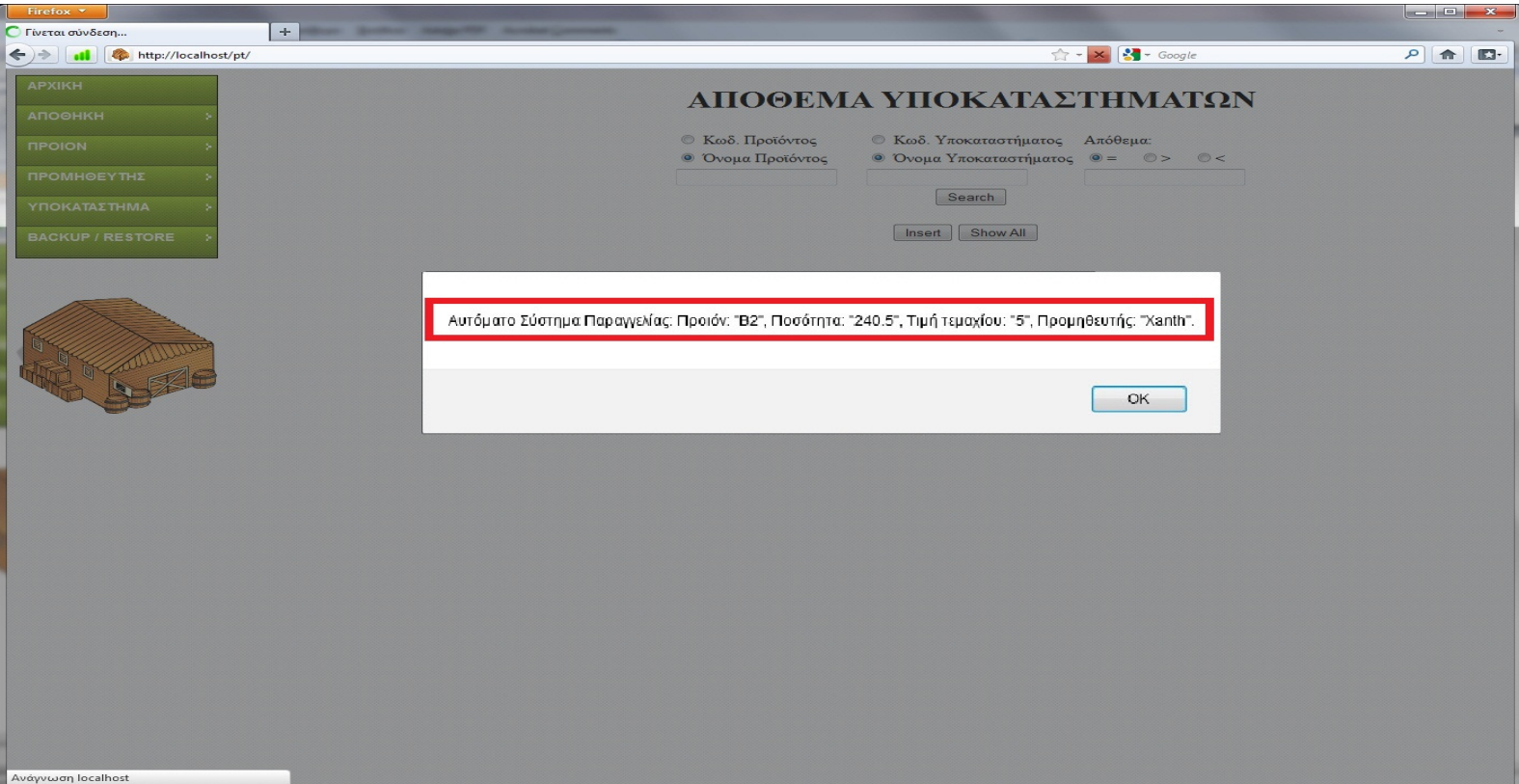
The screenshot shows a web browser window with the URL 'http://localhost/pt/'. The page title is 'ΑΠΟΘΕΜΑ ΑΠΟΘΗΚΩΝ'. On the left, there is a green sidebar with navigation links: ΑΡΧΙΚΗ, ΑΠΟΘΗΚΗ, ΠΡΟΙΟΝ, ΠΡΟΜΗΘΕΥΤΗΣ, ΥΠΟΚΑΤΑΣΤΗΜΑ, and BACKUP / RESTORE. The main content area has search filters for 'Κωδ. Προϊόντος' (value: 6), 'Όνομα Προϊόντος', 'Κωδ. Αποθήκης', and 'Όνομα Αποθήκης'. Below the filters is a 'Search' button and 'Insert' and 'Show All' buttons. A table displays the following data:

Κωδ. Προϊόντος	Όνομα Προϊόντος	Κωδ. Αποθήκης	Όνομα Αποθήκης	Απόθεμα	Ελάχιστο Απόθεμα	Εργασία
6	B2	3	A3	101	100.5	Edit X

The 'Απόθεμα' (101) and 'Ελάχιστο Απόθεμα' (100.5) cells in the table are highlighted with a red box. Below the table are 'Prev' and 'Next' buttons. In the bottom left corner, there is a small illustration of a wooden warehouse with a red roof and several barrels.

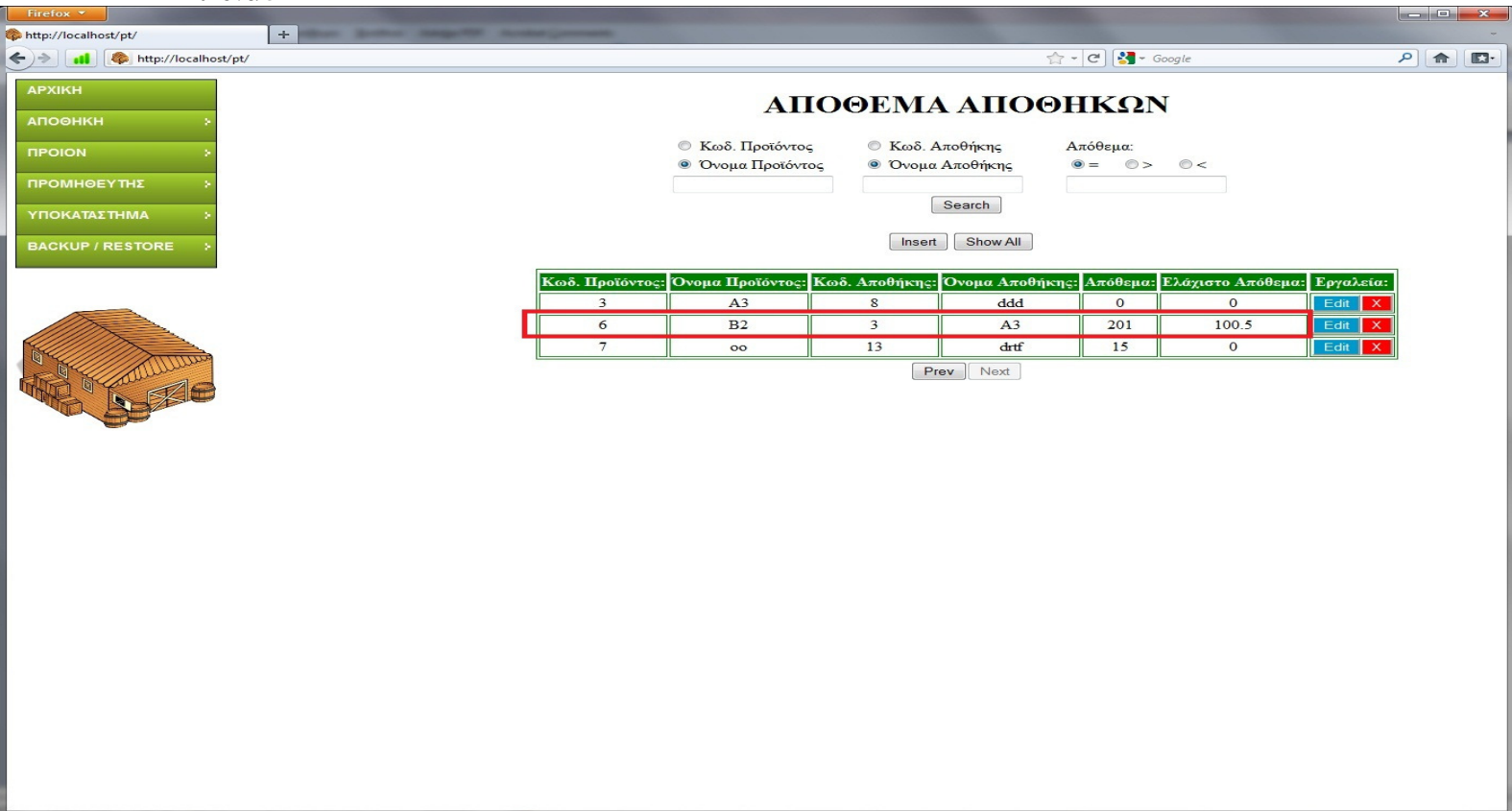
Απόθεμα στην αποθήκη 'A3' για το προϊόν 'B2' είναι 101 και το ελάχιστο απόθεμα 100,5.

Εικόνα 51



Αυτόματο Σύστημα: Προϊόν: 'B2', Ποσότητα: '240,5' Τιμή τεμαχίου: '5' Προμηθευτής: 'Xanthu'.

Εικόνα 52



Απόθεμα διπλάσιο του ελάχιστου όπως φαίνεται στην τονισμένη με κόκκινο περίγραμμα αποθήκη.

Εικόνα 53

ΑΠΟΘΕΜΑ ΥΠΟΚΑΤΑΣΤΗΜΑΤΩΝ

Κωδ. Προϊόντος
 Κωδ. Υποκαταστήματος
 Απόθεμα:
 Όνομα Προϊόντος
 Όνομα Υποκαταστήματος
 = > <

Search Insert Show All

Κωδ. Προϊόντος	Όνομα Προϊόντος	Κωδ. Υποκαταστήματος	Όνομα Υποκαταστήματος	Απόθεμα	Ελάχιστο Απόθεμα	Εργαλεία
1	A1	136	AA1	12	3	Edit X
6	B2	647	AA2	281	140.5	Edit X

Prev Next

Απόθεμα διπλάσιο του ελάχιστου όπως φαίνεται στην τονισμένο με κόκκινο περίγραμμα υποκατάστημα.

Εικόνα 54

ΠΡΟΙΟΝΤΑ ΠΡΟΜΗΘΕΥΤΩΝ

Κωδ. Προϊόντος
 Κωδ. Προμηθευτή
 Τιμή:
 Όνομα Προϊόντος
 Επώνυμο Προμηθευτή
 = > <

6 Search Insert Show All

a

Κωδ. Προϊόντος	Όνομα Προϊόντος	Κωδ. Προμηθευτή	Επώνυμο Προμηθευτή	Τιμή	Εργαλεία
6	B2	1	Xanth	5	Edit X
6	B2	2	A2	5.1	Edit X

Prev Next

Επιβεβαίωση για την βέλτιστη επιλογή προσφοράς για το προϊόν '6' στους προμηθευτές. Ο 'Xanthu' το παρέχει με τιμή πέντε ενώ ο 'A2' με τιμή πέντε και ένα δέκατο.

Κώδικας PHP στο Παράρτημα Α

5. ΕΠΙΛΟΓΟΣ

5.1. Αποτελέσματα

Στο λογισμικό που κατασκευάσαμε είχε σκοπό την εξοικείωση με τις βασικές τεχνολογίες Διαδικτύου, την ανάλυση λογισμικού και την εφαρμογή τους με βάσεις δεδομένων. Καταρχάς, δημιουργήσαμε τη βάση δεδομένων που συνδέεται με την σελίδα σύμφωνα με της απαιτήσεις όπως ακριβώς θα συναντούσαμε στην αγορά εργασίας. Επίσης, έγινε προσπάθεια βελτιστοποίησης στην Β.Δ. ακολουθώντας τα πρότυπα που υπάρχουν και αποφεύγοντας τα σφάλματα στα οποία συνηθίζεται να υποπίπτει ο προγραμματιστής όπως αναφέρθηκαν στο κεφάλαιο 2. Εξ άλλου αυτή είναι διαχωρισμένη σε δύο πλαίσια, τα λεγόμενα frames. Σε αυτό που βρίσκεται αριστερά είναι το menu πλοήγησης ενώ στο δεξιό εμφανίζεται η καθαυτή σελίδα.

Σαφώς από την φύση της ίδιας της εργασίας έπρεπε να δοθεί προσοχή στην ασφάλεια. Στις μέρες μας το SQL poisoning είναι πολύ διαδεδομένο συνεπώς οποιαδήποτε over-the-web εφαρμογή οφείλει να προστατεύεται μέσω κάποιων τεχνικών ή εργαλείων. Πολύ περισσότερο δε όταν αφορά ένα διαχειριστικό σύστημα ολοκληρωμένης κάθετα επιχείρησης με ιδιαίτερα «ευαίσθητα» δεδομένα που μπορεί να είναι προσωπικά στοιχεία υπαλλήλων, προμηθευτών ακόμα και στοιχεία των συναλλαγών μεταξύ των εμπλεκόμενων. Γι' αυτό στην δική μας εργασία απορρίφθηκε ο συνηθισμένος τρόπος εισαγωγής του χρήστη, δηλαδή μια αρχική σελίδα που ζητάει username, password και εκτελεί κάποιες sql εντολές για την αυθεντικότητα του χρήστη. Αλλά την επιβεβαίωση του χρήστη να την αναλαμβάνει ο apache server στον οποίο έχουμε καταχωρήσει μονάχα ένα σωστό συνδυασμό username, password. Αυτό διότι η σελίδα είναι εξ ολοκλήρου διαχειριστική οπότε δεν κρίθηκαν απαραίτητες επιπλέον όψεις.

Εκτός των παραπάνω, όπως ζητήθηκε από τις απαιτήσεις του λογισμικού ήταν προφανές πως η ταχύτητα και η ευχρηστία της ιστοσελίδας ήθελε επιπρόσθετο ενδιαφέρον ώστε να επιτευχθεί ένα επιτρεπτά ορθό αποτέλεσμα. Είδαμε στο προηγούμενο κεφάλαιο πως για την ευχρηστία εξαρχής ακολουθήθηκαν οι κανόνες που έχουν οριστεί στο Graphical User Interface και για την ταχύτητα αποφευχθήκαν οι περιττές flash εφαρμογές και έγινε εκτενείς χρήση της τεχνολογίας διαδικτύου Ajax. Τα flash συνήθως χρειάζονται αρκετό όγκο στην γραμμή (Bandwidth) ενώ για να εκτελεστούν και με τον Ajax γλιτώνουμε την ανανέωση ολόκληρης της ιστοσελίδας. Μάλιστα, χρησιμοποιήθηκε ένα ξεχωριστό χαρακτηριστικό (feature) της MySql για την καλύτερη χρηστικότητα της εφαρμογής τη σελιδοποίηση (pagination). Το feature αυτό το βλέπουμε όταν μετά από αναζήτηση κάποιου στοιχείου εμφανίζονται το πολύ έξι αποτελέσματα σε κάθε «σελίδα», με την πλοήγηση σε αυτές να γίνεται μέσω των κουμπιών next, prev τα οποία βρίσκονται κάτω από τα αποτελέσματα.

Η πιο πολύπλοκη λειτουργία του λογισμικού μας ήταν ο αυτοματισμός για την απευθείας παρακολούθηση των αποθεματικών. Αυτός αφορούσε τέσσερις οντότητες, τους προμηθευτές, τα προϊόντα, τις αποθήκες και τα υποκαταστήματα. Στην ουσία η υπηρεσία αυτή προσφέρει την βέλτιστη πάντα επιλογή για τις διαδικασίες που διεξάγονται ανάμεσα σε όλες τις παραπάνω οντότητες. Στο προηγούμενο κεφάλαιο δείχτηκε πως η εφαρμογή διαλέγει, όταν είναι απαραίτητο, είτε την μεταφορά προϊόντων από την αποθήκη σε υποκατάστημα. Είτε την παραγγελία στον προμηθευτή με την καλύτερη προσφορά σε περιπτώσεις έλλειψης κάποιου προϊόντος σε αποθήκη ή υποκατάστημα. Δηλαδή πρόκειται για έναν ηλεκτρονικό επιβλέπον διευθυντή που ενεργεί ανάλογα για κάθε περίπτωση με γνώμονα το συμφέρον της επιχείρησης.

Τέλος, μία ακόμα απαραίτητη λειτουργία για μια ολοκληρωμένη εργασία (όπως είδαμε στο κεφάλαιο 3) είναι η δημιουργία αντιγράφων ασφαλείας (backups). Αυτά αποθηκεύονται σε σκληρό δίσκο, ενώ υπάρχει και η δυνατότητα να επαναφέρει όποιο αντίγραφο επιλέξει σε οποιαδήποτε στιγμή το θελήσει ο administrator μέσα από την ιστοσελίδα. Η συνολική εφαρμογή μπορεί να θεωρηθεί ως ένα αποτέλεσμα ερασιτεχνικό δημιουργημένο εξ ολοκλήρου από κώδικα php, html, javascript και mysql.

5.2. Συμπεράσματα

Το βασικότερο συμπέρασμα που μπορεί να εξαχθεί είναι πως Η PHP, η Mysql και ο Apache Server συνεργάζονται άψογα μεταξύ τους. Μάλιστα στις μέρες μας είναι από τα πλέον χρησιμοποιούμενα εργαλεία για την ανάπτυξη διαδικτυακών εφαρμογών με υψηλά κριτήρια αξιοπιστίας και απόδοσης. Η ανάπτυξη μιας πτυχιακής βοηθά τον σπουδαστή να εμπεδώσει θεωρητικά και τεχνικά θέματα τα οποία διδάχθηκε κατά την διάρκεια των σπουδών του, δημιουργώντας τις προϋποθέσεις να τα χρησιμοποιήσει στην μετέπειτα σταδιοδρομία του.

Σαφώς και η συνολική εργασία μας δεν είναι εργαλείο επαγγελματικό αλλά μπορεί να θεωρηθεί ως μια παραλλαγή έτοιμων εφαρμογών όπως το cs-cart κ.α. Επίσης, γράφοντας εμείς οι ίδιοι την εφαρμογή από την αρχή μέχρι το τέλος βελτιώσαμε και αναπτύξαμε γνώσεις τόσο για τον χειρισμό διαφόρων εφαρμογών όσο και ξένου πηγαίου κώδικα.

Συνοψίζοντας μπορούμε με βεβαιότητα να πούμε ότι καλύψαμε σε ένα καλό επίπεδο την λειτουργία της ιστοσελίδας, την ευχρηστία της. Ακόμα ότι η βάση δεδομένων που κατασκευάσαμε, κανονικοποιήθηκε (έγινε στη μορφή πινάκων) με τρόπο κατά τον οποίο ακολουθήθηκαν οι κοινοί, παγκόσμιοι κανόνες. Τέλος, μελετήθηκαν ως ένα μερικό βάθος οι προτάσεις του GUI (Graphical User Interface) για την διεπαφή του χρήστη με το πληροφοριακό μας σύστημα. Προσπαθήσαμε να τις εφαρμόσουμε σε όσο μεγαλύτερο μέρος της ιστοσελίδα ήταν δυνατό.

5.3. Βιβλιογραφία & χρήσιμες ιστοσελίδες

1. “The Relational Model for Database Management”, Version 2, E. F. Codd
2. “An Introduction to Database Systems (8th Edition)”, C. J. Date
3. “Θεμελιώδης Αρχές Συστημάτων Βάσεων Δεδομένων”, 3η Έκδοση Αναθεωρημένη R. Elmasri-S.B. Navathe
4. “PHP Bible”, 2nd Edition, Tim Converse, Joyce Park
5. “Servlets και Σελίδες Διακομιστή Java”, Marty Hall, Larry Brown
6. “PHP 5 Power Programming”, Andi Gutmans, Stig Bakken, Derick Rethan
7. PHP and MySQL Web Development Luke Welling, Laura Thomson
8. <https://eclass.teicrete.gr/courses/TP204/> Παπαδάκης Νίκος
9. <https://eclass.teicrete.gr/courses/TP132/> Ακουμιανάκης Δημοσθένης
10. <http://www.w3schools.com/>
11. <http://www.php.net/>
12. <http://www.mysql.com/>
13. <http://el.wikipedia.org/wiki/Wiki> και <http://en.wikipedia.org/wiki/Wiki>
14. <http://www.careerride.com/MySQL-disadvantages.aspx>
15. http://www.ehow.com/list_6309384_advantages-disadvantages-oracle-sql.html
16. <http://www.internetnow.gr/en/node/102>
17. <http://postgresql.gr/node/3>
18. <http://forums.devshed.com/postgresql-help-21/advantages-and-disadvantages-of-postgresql-106805.html>
19. <http://www.neural.uom.gr/Documents/DataBases/chapter3.pdf>

6. ΠΑΡΑΡΤΗΜΑ Α

6.1. Κώδικας για την τεχνολογία Ajax

```

<script type="text/javascript">
function showAP(start, str_ap, str_pr, str_stock, rdio)
{
if (window.XMLHttpRequest)
  { // code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }
else
  { // code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.onreadystatechange=function()
  {
  if (xmlhttp.readyState==4 && xmlhttp.status==200)
  {

document.getElementById("txtHint_ap").innerHTML=xmlhttp.responseText
;
  var x=xmlhttp.responseText;
  }
  }

chosen_pr = ""
chosen_ap = ""
chosen_stock = ""

len_pr = document.form1.radio1.length
len_ap = document.form1.radio2.length
len_stock = document.form1.radio3.length

for (i = 0; i <len_pr; i++) {
if (document.form1.radio1[i].checked) {
chosen_pr = document.form1.radio1[i].value
}
}
for (i = 0; i <len_ap; i++) {
if (document.form1.radio2[i].checked) {
chosen_ap = document.form1.radio2[i].value
}
}
for (i = 0; i <len_stock; i++) {
if (document.form1.radio3[i].checked) {
chosen_stock = document.form1.radio3[i].value
}
}

if (str_pr=="")
{

```

```

str_pr=1;
chosen_pr=1;
}
if (str_ap=="")
{
str_ap=1;
chosen_ap=1;
}
if (str_stock=="")
{
str_stock=1;
chosen_stock=1;
apothema=1;
} else apothema="apothema"

if (rdio)
{
    chosen=rdio;
}

```

6.2. Κώδικας στην σελίδα εμφάνισης «Προϊόν»

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-7" />
<title>&Pi;&Rho;&Omicron;&Iota;&Omicron;&Nu;</title>

<?php //search
include("bin/ajax_search.php");
?>
xmlhttp.open("GET","bin/ajax_proion.php?start="+start+"&termtxt="+st
r+"&radiobtn="+chosen,true);
xmlhttp.send();
}

</script>

    <link rel="stylesheet" href="/pt/menu/table_style.css"
type="text/css" />
<link rel="stylesheet" href="/pt/menu/table_style_blue.css"
type="text/css" />
</head>

<body>
<?php //login
include("bin/login.php");
?>
<?php //connect
include("bin/connect.php");
?>
<H1 align="center">ΠΡΟΙΟΝ&Tau;&Alpha;</H1>

```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
<form id="form1" name="form1" method="post" align="center">
  <div align="center">
    <table width="151" border="0">
      <tr>
        <td width="145"><input type="radio" name="radio"
value="kwd_proiontos" />
Κωδ. Προϊόντος
        </td>
      </tr>
      <tr>
        <td><input type="radio" name="radio" value="onoma"
checked="checked"/>
Όνομα
        </td>
      </tr>
      <tr>
        <td><input type="radio" name="radio" value="timi" />
Τιμή
        </td>
      </tr>
      <tr>
        <td><input type="radio" name="radio" value="eidos" />
Είδος
        </td>
      </tr>
      <tr>
        <td><input type="radio" name="radio" value="kataskeyastis"
/>
Κατασκευαστή
        </td>
      </tr>
    </table>
  </div>
  <div align="center">Search:
  <input type="text" name="term"/>
  <input name="name_ap" type="button" onclick="showAP(0,
term.value)" value="Search"/>
  </div>
</form>
<br />
<div align="center">
<table width="200" border="0">
  <tr>
    <td width="80">
      <form action="bin/proion_insert.php" method="POST"
target="ap_ins" onsubmit="window.open('', 'ap_ins', 'top=80,
left=200,
width=450,height=250,status=yes,resizable=no,scrollbars=no')">
        <div align="right">
          <input type="submit" name="insert" value="Insert" />
        </div>
      </form>
    </td>
    <td width="109">
```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
<form action="bin/show.php" method="POST" target="ap_show"
onsubmit="window.open('', 'ap_show', 'top=80, left=200,
width=600,height=350,status=yes,resizable=yes,scrollbars=yes')">
    <input style="display:none" type="text" name="table"
value="proion" />
    <input type="submit" name="Show_All" value="Show All" />
</form>
</td>
</tr>
</table>
<form name="form" method="post">
<div id="txtHint_ap"></div>
</form>
</div>
<?php //delete
$table=proion;
$key=kwd_proiontos;
$kwd=$_REQUEST["kwd"];

include("bin/delete.php");
?>
</form>
<?php //edit
$onoma=$_REQUEST["onoma"];
$timi=$_REQUEST["timi"];
$eidος=$_REQUEST["eidος"];
$kataskeyastis=$_REQUEST["kataskeyastis"];

for ($k=1 ;$k<=$num_rows; $k++)
{
    if (isset($_POST["E$k"]))
    {
        echo '<br />';
        echo "<table border='1' class =\"edit\"
align='center'><tr>";
        echo "<colgroup span='5' align='left'></colgroup>";
        echo "<style= \"color:#0000FF;\"></colgroup>";
        echo '<th>Κωδ. Αποθήκης:</th>';
        echo '<th>Όνομα:</th>';
        echo '<th>Τιμή:</th>';
        echo '<th>Είδος:</th>';
        echo '<th>Κατασκευαστής:</th>';
        echo '<th></th>';
        echo '<tr>';

        print "<form action= \"$_SERVER[PHP_SELF]\" form
method= \"post\">";
        echo '<td>' . $kwd[$k] . '</td>';
        print "<input style= \"display:none\" type= \"text\"
name= \"kwdd\" value= $kwd[$k] size= \"15\">";
        echo '<td>';
        print "<input style= \"display\" type= \"text\"
name= \"onomaa\" value= $onoma[$k] size= \"15\">";
        echo '</td>';
        echo '<td>';
```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
        print "<input style=\"display\" type=\"text\"
name=\"timii\" value=$timi[$k] size=\"25\">";
        echo '</td>';
        echo '<td>';
        print "<input style=\"display\" type=\"text\"
name=\"eidoss\" value=$eidoss[$k] size=\"8\">";
        echo '</td>';
        echo '<td>';
        print "<input style=\"display\" type=\"text\"
name=\"kataskeyastiss\" value=$kataskeyastis[$k] size=\"15\">";
        echo '</td>';
        echo '<td align="center">';
        print "<input type=\"submit\" name=\"updatebtn\"
value=\"OK\">";
        echo '</td>';
    }

    if (isset($_POST["updatebtn"]))
    {
        $kwdd=$_REQUEST["kwdd"];
        $onomaa=$_REQUEST["onomaa"];
        $timii=$_REQUEST["timii"];
        $eidoss=$_REQUEST["eidoss"];
        $kataskeyastiss=$_REQUEST["kataskeyastiss"];
        if ($eidoss == Null ){
            $eidoss = "-" ;}
        if ($timii == Null ){
            $timii = '0' ;}
        if ($kataskeyastiss == Null ){
            $kataskeyastiss = "-" ;}
        if ($onomaa == Null )
        {
            print "<script type=\"text/javascript\">";
            print "alert('Δεν έχετε συμπληρώσει όνομα!')";
            print "</script>";
        }
        else if (!is_numeric($timii) || $timii < 0)
        {
            print "<script type=\"text/javascript\">";
            print "alert('Λάθος στο πεδίο αριθμού!')";
            print "</script>";
        }

        else if (!mysql_query("update proion set
`onoma`=`$onomaa`, `eidoss`=`$eidoss`, `timi`=`$timii`,
`kataskeyastis`=`$kataskeyastiss` WHERE `kwd_proiontos` = '$kwdd'"))
        {
            echo mysql_error();
        }
        else
        {
            print "<script type=\"text/javascript\">";
            print "alert('Η ανανέωση ολοκληρώθηκε!')";
            print "</script>";
        }
    }
}
```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
        mysql_close($dbhandle);
        exit;
    }
}

if (isset($_POST["next"]))
{
    $trm=$_REQUEST["termt"];
    $rdio=$_REQUEST["rdio"];
    $eu=$_REQUEST["eu"];
    $nume=$_REQUEST["nume"];
    $eu+=6;
    if ($trm==1)$trm="";
    print "<script type=\"text/javascript\">";
    print "    document.form1.term.value = \"$trm\";";
    print "</script>";
    if ($nume>$eu)
    {
        print "<script type=\"text/javascript\">";
        print "showAP($eu, '$trm', '$rdio');";
        print "</script>";
    }
}

if (isset($_POST["prev"]))
{
    $trm=$_REQUEST["termt"];
    $rdio=$_REQUEST["rdio"];
    $eu=$_REQUEST["eu"];
    $nume=$_REQUEST["nume"];
    $eu-=6;

    if ($trm==1)$trm="";
    print "<script type=\"text/javascript\">";
    print "    document.form1.term.value = \"$trm\";";
    print "</script>";
    if ($nume>$eu)
    {
        print "<script type=\"text/javascript\">";
        print "showAP($eu, '$trm', '$rdio');";
        print "</script>";
    }
}

print "</form>";
mysql_close($dbhandle);
?>
</body>
</html>
```


6.3. Κώδικας για αναζήτηση προϊόντος στην σελίδα που εκτελείται στο *background*

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>ajax_apothiki</title>
<link rel="stylesheet" href="/pt/menu/table_style.css"
type="text/css" />
</head>
<body>
<?php include("login.php"); ?>
<?php include("connect.php"); ?>
<?php
$termtxt=$_GET["termtxt"];
$radiobtn=$_GET["radiobtn"];
$start=$_GET['start'];
if(!$start > 0) { // This variable is set
to zero for the first page
$start = 0;
}
$eu = ($start -0);
$limit = 6; // No of records to be
shown per page.
$this1 = $eu + $limit;
$back = $eu - $limit;
$next = $eu + $limit;
$query2="SELECT * FROM proion WHERE $radiobtn like '%$termtxt%'";
$result2=mysql_query($query2);
echo mysql_error();
$nume=mysql_num_rows($result2);
print "<input style=\"display:none\" type=\"text\" name=\"eu\"
value=$eu>";
print "<input style=\"display:none\" type=\"text\" name=\"nume\"
value=$nume>";
$sql="SELECT * FROM proion WHERE $radiobtn like '%$termtxt%' limit
$eu, $limit";
$result = mysql_query($sql);
echo '<br />';
echo "<table border='1' class =\"asd\" align='center'><tr>";
echo "<colgroup span='5' align='left'></colgroup>";
echo "<style=\"color:#0000FF;\"></colgroup>";
echo '<th>Κωδ. Προϊόντος:</th>';
echo '<th>Όνομα:</th>';
echo '<th>Τιμή:</th>';
echo '<th>Είδος:</th>';
echo '<th>Κατασκευαστής:</th>';
echo '<th>Εργαλεία:</th>';
echo '<tr>';
print "<form action=\"$_SERVER[PHP_SELF]\" form
method=\"post\">";
while($row = mysql_fetch_array($result))
{

```

```

        $i+=1;
        echo '<td>'.<a
href=http://localhost/pt/exei.php?code_pr=$row[0]>$row[0]</a>'.</td
>';
        print "<input style=\"display:none\" type=\"text\"
name=\"kwd[$i]\" value=$row[0] />";
        echo '<td>' . $row[1] . '</td>';
        print "<input style=\"display:none\" type=\"text\"
name=\"onoma[$i]\" value=$row[1] />";
        echo '<td>' . $row[2] . '</td>';
        print "<input style=\"display:none\" type=\"text\"
name=\"timi[$i]\" value=$row[2] />";
        echo '<td>' . $row[3] . '</td>';
        print "<input style=\"display:none\" type=\"text\"
name=\"eidosa[$i]\" value=$row[3] />";
        echo '<td>' . $row[4] . '</td>';
        print "<input style=\"display:none\" type=\"text\"
name=\"kataskeyastis[$i]\" value=$row[4] />";
        echo '<td>';
        print "<input name=\"E$i\" type=\"submit\" value=\"Edit\"
style=\"color:#FFFFFF; background-color:#09C; font:bold\"/>";
        print "<input name=\"X$i\" type=\"submit\" value=\"X\"
style=\"color:#FFFFFF; background-color:#FF0000; font:bold\"/>";
        echo '</td>';
        echo '<tr>';
    }
    $y="disabled";
    if ($nume-7<$eu) $x="disabled";
    if ($eu!=0) $y="enabled";
    print "<input style=\"display:none\" type=\"text\"
name=\"termt\" value=$termtxt />";
    print "<input style=\"display:none\" type=\"text\"
name=\"rdio\" value=$radiobtn />";
    echo "<table border='0' align='center'><tr>";
    echo '<td>';
    print "<input type=\"submit\" name=\"prev\" value=\"Prev\"
$y>";
    print "<input type=\"submit\" name=\"next\" value=\"Next\"
$x>";
    echo '</td>';
    print "</form>" ;
    echo "</table>";
mysql_close($dbhandle);
?>
</body>
</html>

```

6.4. Εισαγωγή αποθέματος σε Υποκατάστημα

```

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-7" />
<title>Insert Υποκαταστήματος</title>
</head>
<body>
<script type="text/javascript">

```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
function showAP(th)
{
if (window.XMLHttpRequest)
    { // code for IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp=new XMLHttpRequest();
    }
else
    { // code for IE6, IE5
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
xmlhttp.onreadystatechange=function()
    {
    if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {
document.getElementById("txtHint_ap").innerHTML+xmlhttp.responseText
;
        var x+xmlhttp.responseText;
        }
    }
a=th.options[th.selectedIndex].value;
xmlhttp.open("GET","ajax_apothema_insert.php?a="+a,true);
xmlhttp.send();
}
</script>
<?php include("login.php"); ?>
<?php include("connect.php"); ?>
<?php
print "<form name=\"dsa\" action=\"$_SERVER[PHP_SELF]\" form
method=\"post\">" ;
?>
<table width="103%" border="0">
    <tr>
        <td width="33%">Κωδ. Υποκαταστήματος:
        <?php
print "<select name=\"ypokatastima\" onChange=\"showAP(this)\">";
$table=ypokatastima;
$result = mysql_query("SELECT kwd_ypokatastimatos FROM {$table}");
if (!$result) {
    die("Query to show fields from table failed");
}
$fields_num = mysql_num_fields($result);
print "<option
value=\"0\">&Epsilon;&pi;&é&lambda;&epsilon;&xi;&epsilon;:</option>";
while($row = mysql_fetch_row($result))
{
    foreach($row as $cell)
        print "<option value=\"\$cell\">$cell</option>";
}
mysql_free_result($result);
print "</select>";
?>
</td>
<td width="67%">
<div id="txtHint_ap"></div>
</td>
</tr>
```

```

<tr>
<td>
Απόθεμα: *<br />
<input type="text" name="apothema" />
</td>
<td>
Ελάχιστο Απόθεμα: *<br />
<input type="text" name="min_apothema" />
</td>
</tr>
</table>
<?php
print "<br><input type=\"submit\" value=\"Insert!\"
name=\"submit\">" ;
$opt1=$_POST["ypokatastima"];
$opt2=$_POST["proion"];
$apothema=$_POST["apothema"];
$min_ap=$_POST["min_apothema"];

if(isset($_POST["submit"]))

{
    if ($opt1==0 || $opt2==0 || !is_numeric($apothema) ||
!is_numeric($min_ap) || $apothema < 1 || $min_ap < 1 ||
$apothema==NULL || $min_ap==NULL || $apothema<$min_ap)
    {
        print "<script type=\"text/javascript\">";
        print "alert('Λάθος Επιλογή!')";
        print "</script>";
    }
    else if (mysql_query("insert into exei_2 (kwd_ypokatastimatos,
kwd_proiontos, apothema, elaxisto_apothema) values ( $opt1 , $opt2,
$apothema, $min_ap)"))
    {
        print "<script type=\"text/javascript\">";
        print "alert('Η καταχώρηση έγινε επιτυχώς!')";
        print "</script>";
    }
    else
    {
        echo mysql_error();
    }
}
mysql_close($dbhandle);
?>
*&Upsilon;&pi;&omicron;&chi;&rho;&epsilon;&omega;&tau;&iota;&kappa;&omicron;
&Pi;&epsilon;&delta;&iota;
</body>

```

6.5. Εισαγωγή αποθέματος σε Αποθήκη

```
<head>
```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-7" />
<title>Insert Προϊόντος</title>
</head>
<body>
<?php
include("login.php");
?>
<?php
include("connect.php");
?>
<?php
print "<form action=\"$_SERVER[PHP_SELF]\" form method=\"post\">" ;
?>

<table width="68%" border="0">
  <tr>
    <td width="41%">&Kappa;&omega;&delta;.
&Pi;&rho;&omicron;ϊό&nu;&tau;&omicron;&sigmaf;;
    <?php
$code=$_GET["code"];
$table=proion;
print "<select name=\"proion\">";

$result = mysql_query("SELECT kwd_proiontos FROM {$table}");
if (!$result) {
    die("Query to show fields from table failed");
}

$fields_num = mysql_num_fields($result);
print "<option
value=\"0\">&Epsilon;&pi;&é&lambda;&epsilon;&xi;&epsilon;:</option>";
while($row = mysql_fetch_row($result))
{
    foreach($row as $cell2)
        print "<option value=\"{$cell2}\">{$cell2}</option>";
}
mysql_free_result($result);
print "</select>";
?></td>
    <td width="59%">&Kappa;&omega;&delta;.
&Alpha;&pi;&omicron;&theta;ή&kappa;&eta;&sigmaf;;
    <?php
print "<select name=\"apothiki\">";
$table=apothiki;

$result = mysql_query("SELECT kwd_apothikis FROM {$table}");
if (!$result) {
    die("Query to show fields from table failed");
}
$fields_num = mysql_num_fields($result);

print "<option
value=\"0\">&Epsilon;&pi;&é&lambda;&epsilon;&xi;&epsilon;:</option>";
while($row = mysql_fetch_row($result))
{
```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
        foreach($row as $cell)
            print "<option value=\"\$cell\">$cell</option>";
    }
mysql_free_result($result);
print "</select>";
?>
</td>
</tr>
<tr>
<td>
Απόθεμα: *<br />
<input type="text" name="apothema" />
</td>
<td>
Ελάχιστο Απόθεμα: *<br />
<input type="text" name="min_apothema" />
</td>
</tr>
</table>
<?php
print "<br><input type=\"submit\" value=\"Insert!\"
name=\"submit\">" ;
$opt1=$_POST["apothiki"];
$opt2=$_POST["proion"];
$apothema=$_POST["apothema"];
$min_ap=$_POST["min_apothema"];

if(isset($_POST["submit"]))
{
    if ($opt1==0 || $opt2==0 || !is_numeric($apothema) ||
!is_numeric($min_ap) || $apothema < 1 || $min_ap < 1 ||
$apothema==NULL || $min_ap==NULL || $apothema<$min_ap)
    {
        print "<script type=\"text/javascript\">";
        print "alert('Λάθος Επιλογή!')";
        print "</script>";
    }
    else if (mysql_query("insert into exei (kwd_apothikis,
kwd_proiontos, apothema, elaxisto_apothema) values ( $opt1 , $opt2,
$apothema, $min_ap)"))
    {
        print "<script type=\"text/javascript\">";
        print "alert('Η καταχώρηση έγινε επιτυχώς!')";
        print "</script>";
    }
    else
    {
        echo mysql_error();
    }
}
mysql_close($dbhandle);
?>
*
```

Υποχρεωτικο
Πεδι

</body>

6.6. Αναζήτηση σε απόθεμα αποθήκης

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>ajax_apothiki</title>
<link rel="stylesheet" href="/pt/menu/table_style.css"
type="text/css" />
</head>
<body>
<?php
include("login.php");
?>
<?php
include("connect.php");
?>
<?php
$termtxt_ap=$_GET["termtxt_ap"];
$radiobtn_ap=$_GET["radiobtn_ap"];
$termtxt_pr=$_GET["termtxt_pr"];
$radiobtn_pr=$_GET["radiobtn_pr"];
$termtxt_stock=$_GET["termtxt_stock"];
$radiobtn_stock=$_GET["radiobtn_stock"];
$stock=$_GET["stock"];
$start=$_GET['start'];
print "<input style=\"display:none\" type=\"text\" name=\"term_pr\"
value=\"$termtxt_pr />";
print "<input style=\"display:none\" type=\"text\" name=\"rdio_pr\"
value=\"$radiobtn_pr />";
print "<input style=\"display:none\" type=\"text\" name=\"term_ap\"
value=\"$termtxt_ap />";
print "<input style=\"display:none\" type=\"text\" name=\"rdio_ap\"
value=\"$radiobtn_ap />";
print "<input style=\"display:none\" type=\"text\"
name=\"term_stock\" value=\"$termtxt_stock />";
print "<input style=\"display:none\" type=\"text\"
name=\"rdio_stock\" value=\"$radiobtn_stock />";
if(!($start > 0)) { // This variable is set to zero for the
first page
$start = 0;
}
$eu = ($start -0);
$limit = 6; // No of records to be shown per page.
$thisl = $eu + $limit;
$back = $eu - $limit;
$next = $eu + $limit;
function a($radiobtn_ap, $termtxt_ap, $radiobtn_pr, $termtxt_pr,
$stock, $radiobtn_stock, $termtxt_stock, $eu, $limit)
{
```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
echo '<tr>';
global $nume;
////////// WE have to find out the number of records in
our table. We will use this to break the pages////////
$query2="SELECT * FROM exei WHERE $radiobtn_ap like
'$stermtxt_ap' and $radiobtn_pr like '$stermtxt_pr' and $stock
$radiobtn_stock $stermtxt_stock";
$result2=mysql_query($query2);
echo mysql_error();
$nume=mysql_num_rows($result2);

print "<input style=\"display:none\" type=\"text\" name=\"eu\"
value=$eu>";
print "<input style=\"display:none\" type=\"text\" name=\"nume\"
value=$nume>";
$sql="SELECT * FROM exei WHERE $radiobtn_ap like '$stermtxt_ap' and
$radiobtn_pr like '$stermtxt_pr' and $stock $radiobtn_stock
$stermtxt_stock limit $eu, $limit";
$result = mysql_query($sql);
while($row_exei = mysql_fetch_array($result))
{
    $i+=1;
echo '<td>' . $row_exei[0] . '</td>';
print "<input style=\"display:none\" type=\"text\"
name=\"kwd_pr[$i]\" value=$row_exei[0] />";
$sql2="SELECT onoma FROM proion WHERE kwd_proiontos=$row_exei[0]";
$result2 = mysql_query($sql2);
$row_on_pr = mysql_fetch_array($result2);
echo '<td>' . $row_on_pr[0] . '</td>';
print "<input style=\"display:none\" type=\"text\"
name=\"onoma_pr[$i]\" value=$row_on_pr[0] />";
echo '<td>' . $row_exei[1] . '</td>';
print "<input style=\"display:none\" type=\"text\"
name=\"kwd_ap[$i]\" value=$row_exei[1] />";
$sql2="SELECT onoma FROM apothiki WHERE kwd_apothikis=$row_exei[1]";
$result2 = mysql_query($sql2);
$row_on_ap = mysql_fetch_array($result2);
echo '<td>' . $row_on_ap[0] . '</td>';
print "<input style=\"display:none\" type=\"text\"
name=\"onoma_ap[$i]\" value=$row_on_ap[0] />";
echo '<td>' . $row_exei[2] . '</td>';
print "<input style=\"display:none\" type=\"text\"
name=\"apothema[$i]\" value=$row_exei[2] />";
echo '<td>' . $row_exei[3] . '</td>';
print "<input style=\"display:none\" type=\"text\" name=\"min[$i]\"
value=$row_exei[3] />";
echo '<td>';
print "<input name=\"E$i\" type=\"submit\" value=\"Edit\"
style=\"color:#FFFFFF; background-color:#09C; font:bold\"/>";
print "<input name=\"X$i\" type=\"submit\" value=\"X\"
style=\"color:#FFFFFF; background-color:#FF0000; font:bold\"/>";
echo '</td>';
echo '</tr>';
}
}
echo '<br />';
```


Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
echo "<table border='1' class =\"asd\" align='center'><tr>";
echo "<colgroup span='5' align='left'></colgroup>";
echo "<style=\\\"color:#0000FF;\\\"></colgroup>";
echo '<th>Κωδ. Προϊόντος:</th>';
echo '<th>Όνομα Προϊόντος:</th>';
echo '<th>Κωδ. Αποθήκης:</th>';
echo '<th>Όνομα Αποθήκης:</th>';
echo '<th>Απόθεμα:</th>';
echo '<th>Ελάχιστο Απόθεμα:</th>';
echo '<th>Εργαλεία:</th>';
print "<form action=\\\"$_SERVER[PHP_SELF]\\\" form method=\\\"post\\\">";
if ($radiobtn_pr!=1 && $radiobtn_ap==1 && $stock==1) //search only
proion
{
    if ($radiobtn_pr=='onoma')
    {
        $radiobtn_pr="kwd_proiontos";
        $sql_pr="SELECT kwd_proiontos FROM proion WHERE onoma like
'$stermtxt_pr'";
        $result_pr = mysql_query($sql_pr);
        while ($row_pr = mysql_fetch_array($result_pr))
        {
            a(1, 1, $radiobtn_pr, $row_pr[0], 1, "=", 1, $start, $limit);
        }
    } else a(1, 1, $radiobtn_pr, $stermtxt_pr, 1, "=", 1, $start,
$limit);
}
else if ($radiobtn_pr==1 && $radiobtn_ap!=1 && $stock==1) //search
only apothiki
{
    if ($radiobtn_ap=='onoma')
    {
        $radiobtn_ap="kwd_apothikis";
        $sql_ap="SELECT kwd_apothikis FROM apothiki WHERE onoma like
'$stermtxt_ap'";
        $result_ap = mysql_query($sql_ap);
        while ($row_ap = mysql_fetch_array($result_ap))
        {
            a($radiobtn_ap, $row_ap[0], 1, 1, 1, "=", 1, $start, $limit);
        }
    } else a($radiobtn_ap, $stermtxt_ap, 1, 1, 1, "=", 1, $start,
$limit);
}
else if ($radiobtn_pr==1 && $radiobtn_ap==1 && $stock!=1) //search
only apothema
{
    $stock=apothema;
    if($radiobtn_stock=="eq")
    {
        a(1, 1, 1, 1, $stock, "=", $stermtxt_stock, $start, $limit);
    }
    if($radiobtn_stock=="gt")
    {
        a(1, 1, 1, 1, $stock, ">", $stermtxt_stock, $start, $limit);
    }
}
```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
        if($radiobtn_stock=="lt")
        {
a(1, 1, 1, 1, $stock, "<", $termtxt_stock, $start, $limit);
        }
    }
else if ($radiobtn_pr!=1 && $radiobtn_ap!=1 && $stock==1) //search
proion kai apothiki
{
    if ($radiobtn_ap=='onoma' && $radiobtn_pr=='onoma')
    {
        $radiobtn_ap="kwd_apothikis";
        $radiobtn_pr="kwd_proiontos";
        $sql_pr="SELECT kwd_proiontos FROM proion WHERE onoma like
        '$termtxt_pr'";
        $result_pr = mysql_query($sql_pr);
        while ($row_pr = mysql_fetch_array($result_pr))
        {
        $sql_ap="SELECT kwd_apothikis FROM apothiki WHERE onoma like
        '$termtxt_ap'";
            $result_ap = mysql_query($sql_ap);
        while ($row_ap = mysql_fetch_array($result_ap))
        {
a($radiobtn_ap, $row_ap[0], $radiobtn_pr, $row_pr[0], 1, "=", 1,
        $start, $limit);
            }
        }
    }
    else if ($radiobtn_ap=='onoma' &&
    $radiobtn_pr=='kwd_proiontos')
    {
        $radiobtn_ap="kwd_apothikis";
        $sql_ap="SELECT kwd_apothikis FROM apothiki WHERE onoma like
        '$termtxt_ap'";
        $result_ap = mysql_query($sql_ap);
        while ($row_ap = mysql_fetch_array($result_ap))
        {
a($radiobtn_ap, $row_ap[0], $radiobtn_pr, $termtxt_pr, 1, "=", 1,
        $start, $limit);
            }
        }
    else if ($radiobtn_ap=='kwd_apothikis' &&
    $radiobtn_pr=='onoma')
    {
        $radiobtn_pr="kwd_proiontos";
        $sql_pr="SELECT kwd_proiontos FROM proion WHERE onoma like
        '$termtxt_pr'";
        $result_pr = mysql_query($sql_pr);
        while ($row_pr = mysql_fetch_array($result_pr))
        {
a($radiobtn_ap, $termtxt_ap, $radiobtn_pr, $row_pr[0], 1, "=", 1,
        $start, $limit);
            }
        } else a($radiobtn_ap, $termtxt_ap, $radiobtn_pr, $termtxt_pr,
        1, "=", 1, $start, $limit);
    }
}
```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
else if ($radiobtn_pr!=1 && $radiobtn_ap==1 && $stock!=1) //search
proion kai apothema
{
    if ($radiobtn_pr=='onoma')
    {
        $stock=apothema;
        $radiobtn_pr="kwd_proiontos";
        $sql_pr="SELECT kwd_proiontos FROM proion WHERE onoma like
'$stermtxt_pr'";
        $result_pr = mysql_query($sql_pr);
        while ($row_pr = mysql_fetch_array($result_pr))
        {
            if($radiobtn_stock=="eq")
            {
a(1, 1, $radiobtn_pr, $row_pr[0], $stock, "=", $stermtxt_stock,
$start, $limit);
            }

            if($radiobtn_stock=="gt")
            {
a(1, 1, $radiobtn_pr, $row_pr[0], $stock, ">", $stermtxt_stock,
$start, $limit);
            }

            if($radiobtn_stock=="lt")
            {
a(1, 1, $radiobtn_pr, $row_pr[0], $stock, "<", $stermtxt_stock,
$start, $limit);
            }
        }
    }
    else
    {
        if($radiobtn_stock=="eq")
        {
a(1, 1, $radiobtn_pr, $stermtxt_pr, $stock, "=", $stermtxt_stock,
$start, $limit);
        }

        if($radiobtn_stock=="gt")
        {
a(1, 1, $radiobtn_pr, $stermtxt_pr, $stock, ">", $stermtxt_stock,
$start, $limit);
        }

        if($radiobtn_stock=="lt")
        {
a(1, 1, $radiobtn_pr, $stermtxt_pr, $stock, "<", $stermtxt_stock,
$start, $limit);
        }
    }
}
else if ($radiobtn_pr==1 && $radiobtn_ap!=1 && $stock!=1) //search
apothiki kai apothema
{
    if ($radiobtn_ap=='onoma')
```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
{
    $stock=apothema;
    $radiobtn_ap="kwd_apothikis";
    $sql_ap="SELECT kwd_apothikis FROM apothiki WHERE onoma like
    '$stermtxt_ap'";
    $result_ap = mysql_query($sql_ap);
    while ($row_ap = mysql_fetch_array($result_ap))
    {
        if($radiobtn_stock=="eq")
        {
            a($radiobtn_ap, $row_ap[0], 1, 1, $stock, "=", $stermtxt_stock,
            $start, $limit);
        }
        if($radiobtn_stock=="gt")
        {
            a($radiobtn_ap, $row_ap[0], 1, 1, $stock, ">", $stermtxt_stock,
            $start, $limit);
        }
        if($radiobtn_stock=="lt")
        {
            a($radiobtn_ap, $row_ap[0], 1, 1, $stock, "<", $stermtxt_stock,
            $start, $limit);
        }
    }
}
else
{
    if($radiobtn_stock=="eq")
    {
        a($radiobtn_ap, $stermtxt_ap, 1, 1, $stock, "=", $stermtxt_stock,
        $start, $limit);
    }
    if($radiobtn_stock=="gt")
    {
        a($radiobtn_ap, $stermtxt_ap, 1, 1, $stock, ">", $stermtxt_stock,
        $start, $limit);
    }
    if($radiobtn_stock=="lt")
    {
        a($radiobtn_ap, $stermtxt_ap, 1, 1, $stock, "<", $stermtxt_stock,
        $start, $limit);
    }
}
}
else if ($radiobtn_pr!=1 && $radiobtn_ap!=1 && $stock!=1) //search
me ola
{
    if ($radiobtn_ap=='onoma' && $radiobtn_pr=='onoma')
    {
        $radiobtn_ap="kwd_apothikis";
        $radiobtn_pr="kwd_proiontos";
        $sql_pr="SELECT kwd_proiontos FROM proion WHERE onoma like
        '$stermtxt_pr'";
        $result_pr = mysql_query($sql_pr);
        while ($row_pr = mysql_fetch_array($result_pr))
```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
{
$sql_ap="SELECT kwd_apothikis FROM apothiki WHERE onoma like
'$stermtxt_ap'";
        $result_ap = mysql_query($sql_ap);
while ($row_ap = mysql_fetch_array($result_ap))
    {
        if($radiobtn_stock=="eq")
        {
a($radiobtn_ap, $row_ap[0], $radiobtn_pr, $row_pr[0], $stock, "=",
$stermtxt_stock, $start, $limit);
        }

        if($radiobtn_stock=="gt")
        {
a($radiobtn_ap, $row_ap[0], $radiobtn_pr, $row_pr[0], $stock, ">",
$stermtxt_stock, $start, $limit);
        }

        if($radiobtn_stock=="lt")
        {
a($radiobtn_ap, $row_ap[0], $radiobtn_pr, $row_pr[0], $stock, "<",
$stermtxt_stock, $start, $limit);
        }
    }
}
else if ($radiobtn_ap=='onoma' &&
$radiobtn_pr=='kwd_proiontos')
{
        $radiobtn_ap="kwd_apothikis";
$sql_ap="SELECT kwd_apothikis FROM apothiki WHERE onoma like
'$stermtxt_ap'";
        $result_ap = mysql_query($sql_ap);
while ($row_ap = mysql_fetch_array($result_ap))
    {
        if($radiobtn_stock=="eq")
        {
a($radiobtn_ap, $row_ap[0], $radiobtn_pr, $stermtxt_pr, $stock, "=",
$stermtxt_stock, $start, $limit);
        }

        if($radiobtn_stock=="gt")
        {
a($radiobtn_ap, $row_ap[0], $radiobtn_pr, $stermtxt_pr, $stock, ">",
$stermtxt_stock, $start, $limit);
        }

        if($radiobtn_stock=="lt")
        {
a($radiobtn_ap, $row_ap[0], $radiobtn_pr, $stermtxt_pr, $stock, "<",
$stermtxt_stock, $start, $limit);
        }
    }
}
else if ($radiobtn_ap=='kwd_apothikis' &&
$radiobtn_pr=='onoma')
{
```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
$radiobtn_pr="kwd_proiontos";
$sql_pr="SELECT kwd_proiontos FROM proion WHERE onoma like
'$stermtxt_pr'";
$result_pr = mysql_query($sql_pr);
while ($row_pr = mysql_fetch_array($result_pr))
{
    if($radiobtn_stock=="eq")
    {
a($radiobtn_ap, $stermtxt_ap, $radiobtn_pr, $row_pr[0], $stock, "=",
$stermtxt_stock, $start, $limit);
    }

    if($radiobtn_stock=="gt")
    {
a($radiobtn_ap, $stermtxt_ap, $radiobtn_pr, $row_pr[0], $stock, ">",
$stermtxt_stock, $start, $limit);
    }

    if($radiobtn_stock=="lt")
    {
a($radiobtn_ap, $stermtxt_ap, $radiobtn_pr, $row_pr[0], $stock, "<",
$stermtxt_stock, $start, $limit);
    }
}
else
{
    if($radiobtn_stock=="eq")
    {
a($radiobtn_ap, $stermtxt_ap, $radiobtn_pr, $stermtxt_pr, $stock, "=",
$stermtxt_stock, $start, $limit);
    }
    if($radiobtn_stock=="gt")
    {
a($radiobtn_ap, $stermtxt_ap, $radiobtn_pr, $stermtxt_pr, $stock, ">",
$stermtxt_stock, $start, $limit);
    }
    if($radiobtn_stock=="lt")
    {
a($radiobtn_ap, $stermtxt_ap, $radiobtn_pr, $stermtxt_pr, $stock, "<",
$stermtxt_stock, $start, $limit);
    }
}
}
else a(1, 1, 1, 1, 1, "=", 1, $eu, $limit); //search me keno

$y="disabled";
if ($nume-7<$eu) $x="disabled";
if ($eu!=0) $y="enabled";
echo "<table border='0' align='center'><tr>";
echo '<td>';
print "<input type=\"submit\" name=\"prev\" value=\"Prev\"
$y>";
print "<input type=\"submit\" name=\"next\" value=\"Next\"
$x>";
echo '</td>';
```

```
print "</form>" ;
echo "</table>";
mysql_close($dbhandle);
?>
</body>
</html>
```

6.7. Αναζήτηση σε απόθεμα Υποκαταστήματος

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>ajax_ypokatastima</title>
<link rel="stylesheet" href="/pt/menu/table_style.css"
type="text/css" />
</head>
<body>
<?php include("login.php"); ?>
<?php include("connect.php"); ?>
<?php
$termtxt_ap=$_GET["termtxt_ap"];
$radiobtn_ap=$_GET["radiobtn_ap"];
$termtxt_pr=$_GET["termtxt_pr"];
$radiobtn_pr=$_GET["radiobtn_pr"];
$termtxt_stock=$_GET["termtxt_stock"];
$radiobtn_stock=$_GET["radiobtn_stock"];
$stock=$_GET["stock"];
$start=$_GET['start'];
print "<input style=\"display:none\" type=\"text\"
name=\"term_pr\" value=\"$termtxt_pr />";
print "<input style=\"display:none\" type=\"text\"
name=\"rdio_pr\" value=\"$radiobtn_pr />";
print "<input style=\"display:none\" type=\"text\"
name=\"term_ap\" value=\"$termtxt_ap />";
print "<input style=\"display:none\" type=\"text\"
name=\"rdio_ap\" value=\"$radiobtn_ap />";
print "<input style=\"display:none\" type=\"text\"
name=\"term_stock\" value=\"$termtxt_stock />";
print "<input style=\"display:none\" type=\"text\"
name=\"rdio_stock\" value=\"$radiobtn_stock />";
if(!($start > 0)) {
$start = 0;
}
$eu = ($start -0);
$limit = 6;
$thisl = $eu + $limit;
$back = $eu - $limit;
$next = $eu + $limit;
/*echo "rdio_ap=".$radiobtn_ap;
echo " rdio_pr=".$radiobtn_pr;
echo " term_ap=".$termtxt_ap;
```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
echo " term_pr=".$stermtxt_pr;
echo " stock=".$stock;
echo " rdio_stock=".$radiobtn_stock;
echo " term_stock=".$stermtxt_stock;*/

function a($radiobtn_ap, $stermtxt_ap, $radiobtn_pr, $stermtxt_pr,
$stock, $radiobtn_stock, $stermtxt_stock, $eu, $limit)
{
    echo '<tr>';
    global $nume;
    $query2="SELECT * FROM exei_2 WHERE $radiobtn_ap like '$stermtxt_ap'
and $radiobtn_pr like '$stermtxt_pr' and $stock $radiobtn_stock
$stermtxt_stock";
    $result2=mysql_query($query2);
    echo mysql_error();
    $nume=mysql_num_rows($result2);
    print "<input style=\"display:none\" type=\"text\" name=\"eu\"
value=$eu>";
    print "<input style=\"display:none\" type=\"text\" name=\"nume\"
value=$nume>";
    $sql="SELECT * FROM exei_2 WHERE $radiobtn_ap like '$stermtxt_ap' and
$radiobtn_pr like '$stermtxt_pr' and $stock $radiobtn_stock
$stermtxt_stock limit $eu, $limit";
    $result = mysql_query($sql);
    while($row_exei = mysql_fetch_array($result))
    {
        $i+=1;
        echo '<td>' .$row_exei[0] .'</td>';
        print "<input style=\"display:none\" type=\"text\"
name=\"kwd_pr[$i]\" value=$row_exei[0] />";
        $sql2="SELECT onoma FROM proion WHERE kwd_proiontos=$row_exei[0]";
        $result2 = mysql_query($sql2);
        $row_on_pr = mysql_fetch_array($result2);
        echo '<td>' .$row_on_pr[0] .'</td>';
        print "<input style=\"display:none\" type=\"text\"
name=\"onoma_pr[$i]\" value=$row_on_pr[0] />";
        echo '<td>' .$row_exei[1] .'</td>';
        print "<input style=\"display:none\" type=\"text\"
name=\"kwd_ap[$i]\" value=$row_exei[1] />";
        $sql2="SELECT onoma FROM ypokatastima WHERE
kwd_ypokatastimatos=$row_exei[1]";
        $result2 = mysql_query($sql2);
        $row_on_ap = mysql_fetch_array($result2);
        echo '<td>' .$row_on_ap[0] .'</td>';
        print "<input style=\"display:none\" type=\"text\"
name=\"onoma_ap[$i]\" value=$row_on_ap[0] />";

        echo '<td>' .$row_exei[2] .'</td>';
        print "<input style=\"display:none\" type=\"text\"
name=\"apothema[$i]\" value=$row_exei[2] />";
        echo '<td>' .$row_exei[3] .'</td>';
        print "<input style=\"display:none\" type=\"text\" name=\"min[$i]\"
value=$row_exei[3] />";
        echo '<td>';
        print "<input name=\"E$i\" type=\"submit\" value=\"Edit\"
style=\"color:#FFFFFF; background-color:#09C; font:bold\"/>";
    }
}
```


Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
print "<input name=\"X$i\" type=\"submit\" value=\"X\"
style=\"color:#FFFFFF; background-color:#FF0000; font:bold\"/>";
echo '</td>';
echo '</tr>';
}
}
echo '<br />';
echo "<table border='1' class =\"asd\" align='center'><tr>";
echo "<colgroup span='5' align='left'></colgroup>";
echo "<style=\"color:#0000FF;\"></colgroup>";
echo '<th>Κωδ. Προϊόντος:</th>';
echo '<th>Όνομα Προϊόντος:</th>';
echo '<th>Κωδ. Υποκαταστήματος:</th>';
echo '<th>Όνομα Υποκαταστήματος:</th>';
echo '<th>Απόθεμα:</th>';
echo '<th>Ελάχιστο Απόθεμα:</th>';
echo '<th>Εργαλεία:</th>';
print "<form action=\"$_SERVER[PHP_SELF]\" form method=\"post\">";
if ($radiobtn_pr!=1 && $radiobtn_ap==1 && $stock==1) //search only
proion
{
    if ($radiobtn_pr=='onoma')
    {
        $radiobtn_pr="kwd_proiontos";
        $sql_pr="SELECT kwd_proiontos FROM proion WHERE onoma like
'$stermtxt_pr'";
        $result_pr = mysql_query($sql_pr);
        while ($row_pr = mysql_fetch_array($result_pr))
        {
            a(1, 1, $radiobtn_pr, $row_pr[0], 1, "=", 1, $start, $limit);
        }
        } else a(1, 1, $radiobtn_pr, $stermtxt_pr, 1, "=", 1, $start,
$limit);
    }
else if ($radiobtn_pr==1 && $radiobtn_ap!=1 && $stock==1) //search
only ypokatastima
{
    if ($radiobtn_ap=='onoma')
    {
        $radiobtn_ap="kwd_ypokatastimatos";
        $sql_ap="SELECT kwd_ypokatastimatos FROM ypokatastima WHERE onoma
like '$stermtxt_ap'";
        $result_ap = mysql_query($sql_ap);
        while ($row_ap = mysql_fetch_array($result_ap))
        {
            a($radiobtn_ap, $row_ap[0], 1, 1, 1, "=", 1, $start, $limit);
        }
        } else a($radiobtn_ap, $stermtxt_ap, 1, 1, 1, "=", 1, $start,
$limit);
    }
else if ($radiobtn_pr==1 && $radiobtn_ap==1 && $stock!=1) //search
only apothema
{
    $stock=apothema;
    if($radiobtn_stock=="eq")
    {
```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
a(1, 1, 1, 1, $stock, "=", $termtxt_stock, $start, $limit);
}
if($radiobtn_stock=="gt")
{
a(1, 1, 1, 1, $stock, ">", $termtxt_stock, $start, $limit);
}

if($radiobtn_stock=="lt")
{
a(1, 1, 1, 1, $stock, "<", $termtxt_stock, $start, $limit);
}
}
else if ($radiobtn_pr!=1 && $radiobtn_ap!=1 && $stock==1) //search
proion kai ypokatastima
{
if ($radiobtn_ap=='onoma' && $radiobtn_pr=='onoma')
{
$radiobtn_ap="kwd_ypokatastimatos";
$radiobtn_pr="kwd_proiontos";
$sql_pr="SELECT kwd_proiontos FROM proion WHERE
onoma like '$termtxt_pr'";
$result_pr = mysql_query($sql_pr);
while ($row_pr = mysql_fetch_array($result_pr))
{
$sql_ap="SELECT kwd_ypokatastimatos FROM ypokatastima WHERE onoma
like '$termtxt_ap'";
$result_ap = mysql_query($sql_ap);
while ($row_ap = mysql_fetch_array($result_ap))
{
a($radiobtn_ap, $row_ap[0], $radiobtn_pr, $row_pr[0], 1, "=", 1,
$start, $limit);
}
}
}
else if ($radiobtn_ap=='onoma' &&
$radiobtn_pr=='kwd_proiontos')
{
$radiobtn_ap="kwd_ypokatastimatos";
$sql_ap="SELECT kwd_ypokatastimatos FROM ypokatastima WHERE onoma
like '$termtxt_ap'";
$result_ap = mysql_query($sql_ap);
while ($row_ap = mysql_fetch_array($result_ap))
{
a($radiobtn_ap, $row_ap[0], $radiobtn_pr, $termtxt_pr, 1, "=", 1,
$start, $limit);
}
}
else if ($radiobtn_ap=='kwd_ypokatastimatos' &&
$radiobtn_pr=='onoma')
{
$radiobtn_pr="kwd_proiontos";
$sql_pr="SELECT kwd_proiontos FROM proion WHERE onoma like
'$termtxt_pr'";
$result_pr = mysql_query($sql_pr);
while ($row_pr = mysql_fetch_array($result_pr))
{
```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
a($radiobtn_ap, $termtxt_ap, $radiobtn_pr, $row_pr[0], 1, "=", 1,
$start, $limit);
    }
    } else a($radiobtn_ap, $termtxt_ap, $radiobtn_pr, $termtxt_pr,
1, "=", 1, $start, $limit);

}
else if ($radiobtn_pr!=1 && $radiobtn_ap==1 && $stock!=1) //search
proion kai apothema
{
    if ($radiobtn_pr=='onoma')
    {
        $stock=apothema;
        $radiobtn_pr="kwd_proiontos";
        $sql_pr="SELECT kwd_proiontos FROM proion WHERE onoma like
'$termtxt_pr'";
        $result_pr = mysql_query($sql_pr);
        while ($row_pr = mysql_fetch_array($result_pr))
        {
            if($radiobtn_stock=="eq")
            {
a(1, 1, $radiobtn_pr, $row_pr[0], $stock, "=", $termtxt_stock,
$start, $limit);
            }
            if($radiobtn_stock=="gt")
            {
a(1, 1, $radiobtn_pr, $row_pr[0], $stock, ">", $termtxt_stock,
$start, $limit);
            }
            if($radiobtn_stock=="lt")
            {
a(1, 1, $radiobtn_pr, $row_pr[0], $stock, "<", $termtxt_stock,
$start, $limit);
            }
        }
    }
    else
    {
        if($radiobtn_stock=="eq")
        {
a(1, 1, $radiobtn_pr, $termtxt_pr, $stock, "=", $termtxt_stock,
$start, $limit);
        }
        if($radiobtn_stock=="gt")
        {
a(1, 1, $radiobtn_pr, $termtxt_pr, $stock, ">", $termtxt_stock,
$start, $limit);
        }
        if($radiobtn_stock=="lt")
        {
a(1, 1, $radiobtn_pr, $termtxt_pr, $stock, "<", $termtxt_stock,
$start, $limit);
        }
    }
}
}
```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
else if ($radiobtn_pr==1 && $radiobtn_ap!=1 && $stock!=1) //search
ypokatastima kai apothema
{
    if ($radiobtn_ap=='onoma')
    {
        $stock=apothema;
        $radiobtn_ap="kwd_ypokatastimatos";
        $sql_ap="SELECT kwd_ypokatastimatos FROM ypokatastima WHERE onoma
like '$stermtxt_ap'";
        $result_ap = mysql_query($sql_ap);
        while ($row_ap = mysql_fetch_array($result_ap))
        {
            if($radiobtn_stock=="eq")
            {
                a($radiobtn_ap, $row_ap[0], 1, 1, $stock, "=", $stermtxt_stock,
                $start, $limit);
            }
            if($radiobtn_stock=="gt")
            {
                a($radiobtn_ap, $row_ap[0], 1, 1, $stock, ">", $stermtxt_stock,
                $start, $limit);
            }
            if($radiobtn_stock=="lt")
            {
                a($radiobtn_ap, $row_ap[0], 1, 1, $stock, "<", $stermtxt_stock,
                $start, $limit);
            }
        }
    }
    else
    {
        if($radiobtn_stock=="eq")
        {
            a($radiobtn_ap, $stermtxt_ap, 1, 1, $stock, "=", $stermtxt_stock,
            $start, $limit);
        }
        if($radiobtn_stock=="gt")
        {
            a($radiobtn_ap, $stermtxt_ap, 1, 1, $stock, ">", $stermtxt_stock,
            $start, $limit);
        }
        if($radiobtn_stock=="lt")
        {
            a($radiobtn_ap, $stermtxt_ap, 1, 1, $stock, "<", $stermtxt_stock,
            $start, $limit);
        }
    }
}
else if ($radiobtn_pr!=1 && $radiobtn_ap!=1 && $stock!=1) {
    if ($radiobtn_ap=='onoma' && $radiobtn_pr=='onoma')
    {
        $radiobtn_ap="kwd_ypokatastimatos";
        $radiobtn_pr="kwd_proiontos";
        $sql_pr="SELECT kwd_proiontos FROM proion WHERE
onoma like '$stermtxt_pr'";
```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
        $result_pr = mysql_query($sql_pr);
        while ($row_pr = mysql_fetch_array($result_pr))
        {
$mysql_ap="SELECT kwd_ypokatastimatos FROM ypokatastima WHERE onoma
like '$stermtxt_ap'";
$result_ap = mysql_query($sql_ap);
while ($row_ap = mysql_fetch_array($result_ap))
        {
                if($radiobtn_stock=="eq")
                {
a($radiobtn_ap, $row_ap[0], $radiobtn_pr, $row_pr[0], $stock, "=",
$stermtxt_stock, $start, $limit);
                }
                if($radiobtn_stock=="gt")
                {
a($radiobtn_ap, $row_ap[0], $radiobtn_pr, $row_pr[0], $stock, ">",
$stermtxt_stock, $start, $limit);
                }
                if($radiobtn_stock=="lt")
                {
a($radiobtn_ap, $row_ap[0], $radiobtn_pr, $row_pr[0], $stock, "<",
$stermtxt_stock, $start, $limit);
                }
        }
    }
    else if ($radiobtn_ap=='onoma' &&
$radiobtn_pr=='kwd_proiontos')
    {
        $radiobtn_ap="kwd_ypokatastimatos";
$mysql_ap="SELECT kwd_ypokatastimatos FROM ypokatastima WHERE onoma
like '$stermtxt_ap'";
$result_ap = mysql_query($sql_ap);
while ($row_ap = mysql_fetch_array($result_ap))
        {
                if($radiobtn_stock=="eq")
                {
a($radiobtn_ap, $row_ap[0], $radiobtn_pr, $stermtxt_pr, $stock, "=",
$stermtxt_stock, $start, $limit);
                }
                if($radiobtn_stock=="gt")
                {
a($radiobtn_ap, $row_ap[0], $radiobtn_pr, $stermtxt_pr, $stock, ">",
$stermtxt_stock, $start, $limit);
                }
                if($radiobtn_stock=="lt")
                {
a($radiobtn_ap, $row_ap[0], $radiobtn_pr, $stermtxt_pr, $stock, "<",
$stermtxt_stock, $start, $limit);
                }
        }
    }
    else if ($radiobtn_ap=='kwd_ypokatastimatos' &&
$radiobtn_pr=='onoma')
    {
        $radiobtn_pr="kwd_proiontos";
    }
}
```

Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

```
$sql_pr="SELECT kwd_proiontos FROM proion WHERE onoma like
'$stermtxt_pr'";
$result_pr = mysql_query($sql_pr);
while ($row_pr = mysql_fetch_array($result_pr))
{
    if($radiobtn_stock=="eq")
    {
a($radiobtn_ap, $stermtxt_ap, $radiobtn_pr, $row_pr[0], $stock, "=",
$stermtxt_stock, $start, $limit);
    }

    if($radiobtn_stock=="gt")
    {
a($radiobtn_ap, $stermtxt_ap, $radiobtn_pr, $row_pr[0], $stock, ">",
$stermtxt_stock, $start, $limit);
    }
    if($radiobtn_stock=="lt")
    {
a($radiobtn_ap, $stermtxt_ap, $radiobtn_pr, $row_pr[0], $stock, "<",
$stermtxt_stock, $start, $limit);
    }
}
else
{
    if($radiobtn_stock=="eq")
    {
a($radiobtn_ap, $stermtxt_ap, $radiobtn_pr, $stermtxt_pr, $stock, "=",
$stermtxt_stock, $start, $limit);
    }
    if($radiobtn_stock=="gt")
    {
a($radiobtn_ap, $stermtxt_ap, $radiobtn_pr, $stermtxt_pr, $stock, ">",
$stermtxt_stock, $start, $limit);
    }
    if($radiobtn_stock=="lt")
    { a($radiobtn_ap, $stermtxt_ap, $radiobtn_pr,
$stermtxt_pr, $stock, "<", $stermtxt_stock, $start, $limit); }
}}
else a(1, 1, 1, 1, 1, "=", 1, $eu, $limit); //search me keno
$y="disabled";
if ($nume-7<$eu) $x="disabled";
if ($eu!=0) $y="enabled";
echo "<table border='0' align='center'><tr>";
echo '<td>';
print "<input type=\"submit\" name=\"prev\" value=\"Prev\"
$y>";
print "<input type=\"submit\" name=\"next\" value=\"Next\"
$x>";
echo '</td>';
print "</form>" ;
echo "</table>";
mysql_close($dbhandle);
?>
</body>
</html>
```

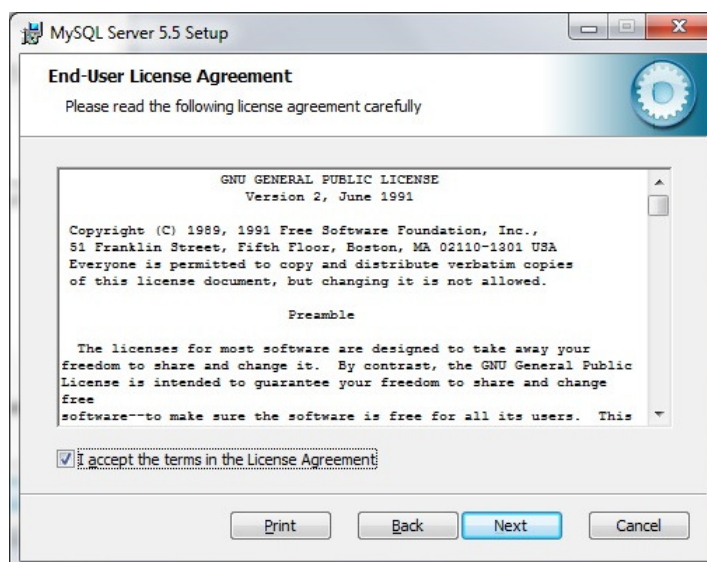
7. ΠΑΡΑΡΤΗΜΑ Β

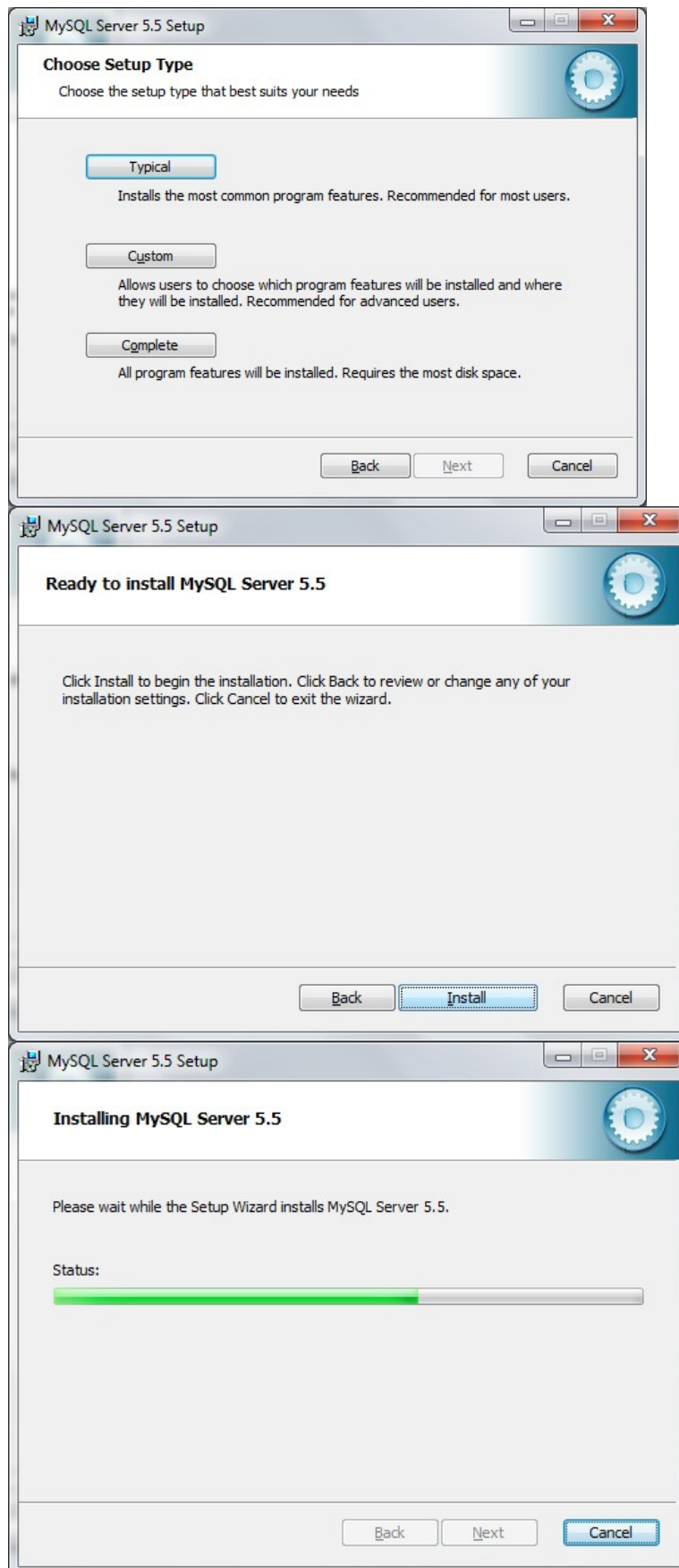
7.1. Εργαλεία που χρησιμοποιήθηκαν

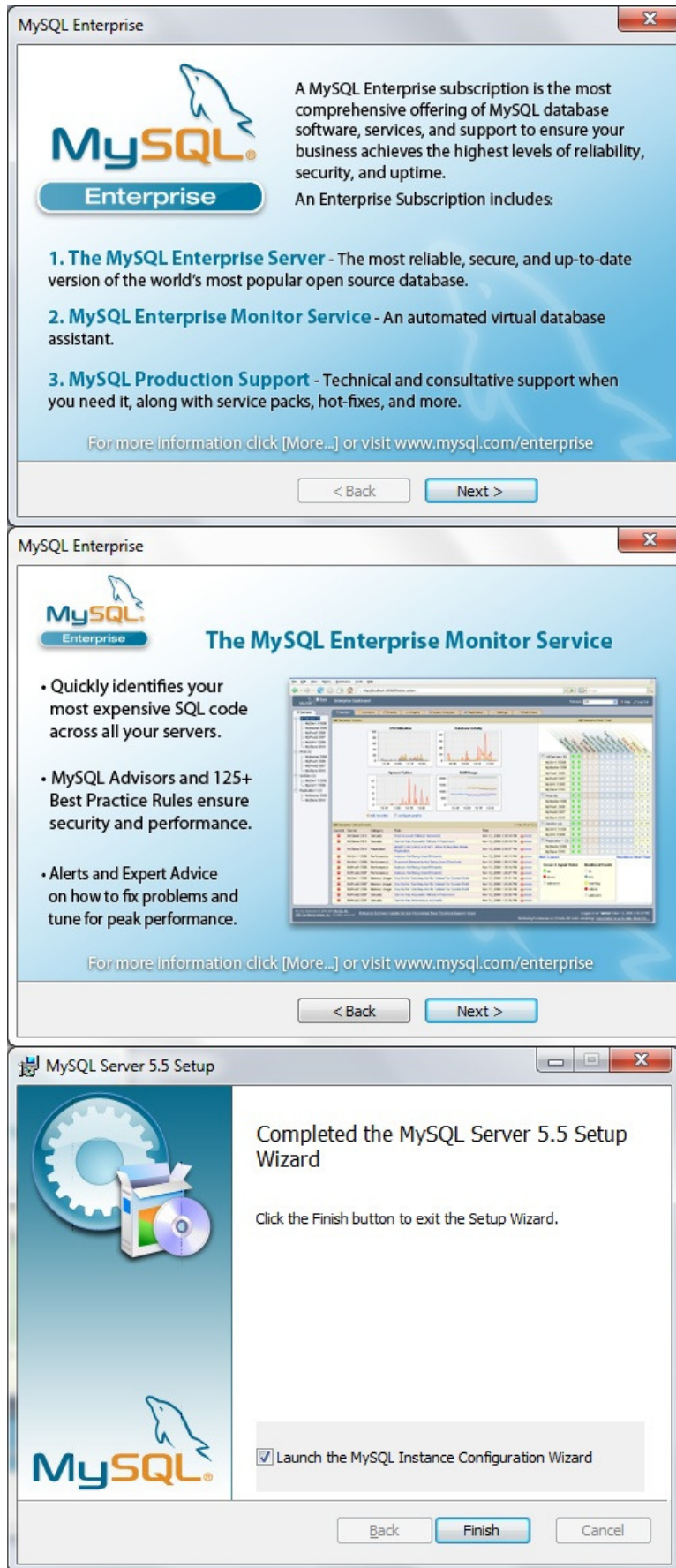
7.1.1. Mysql Server 5.5

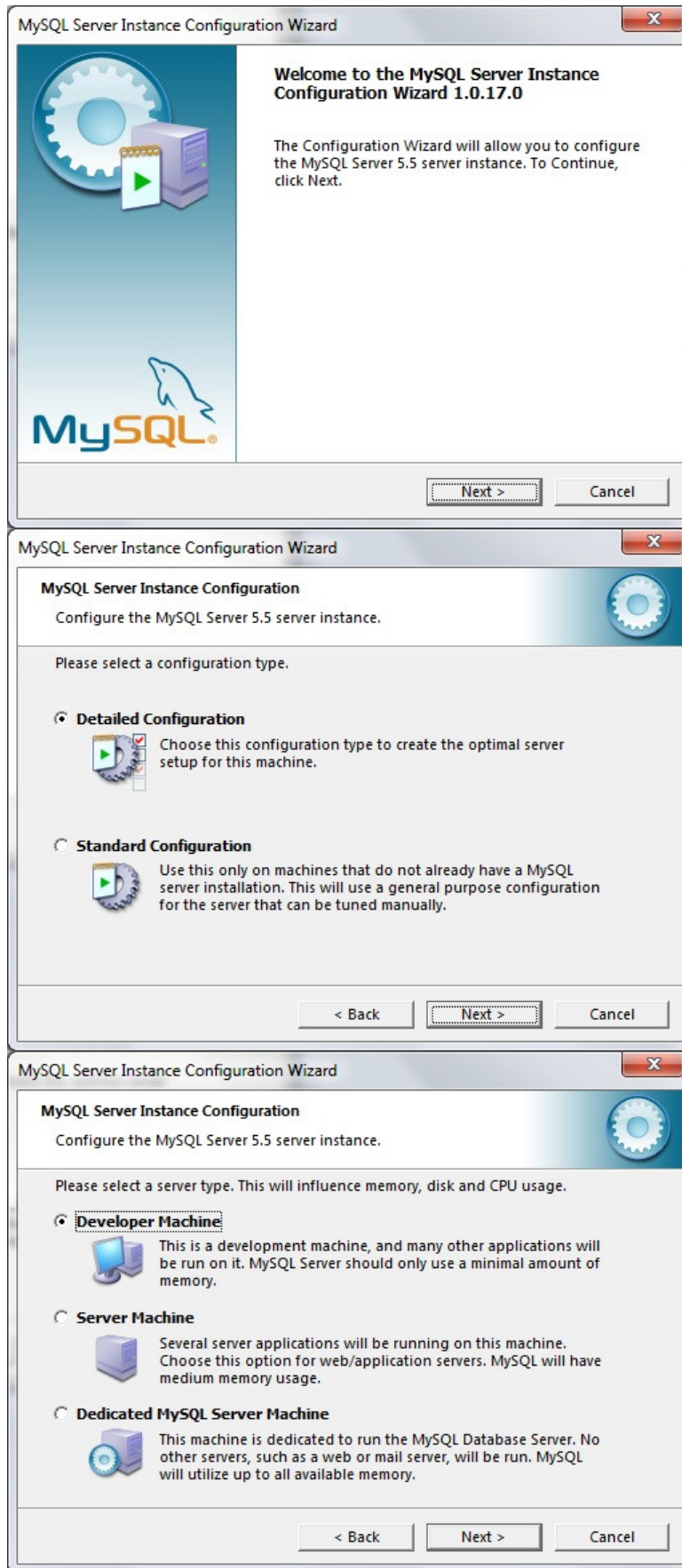
Εγκατάσταση:

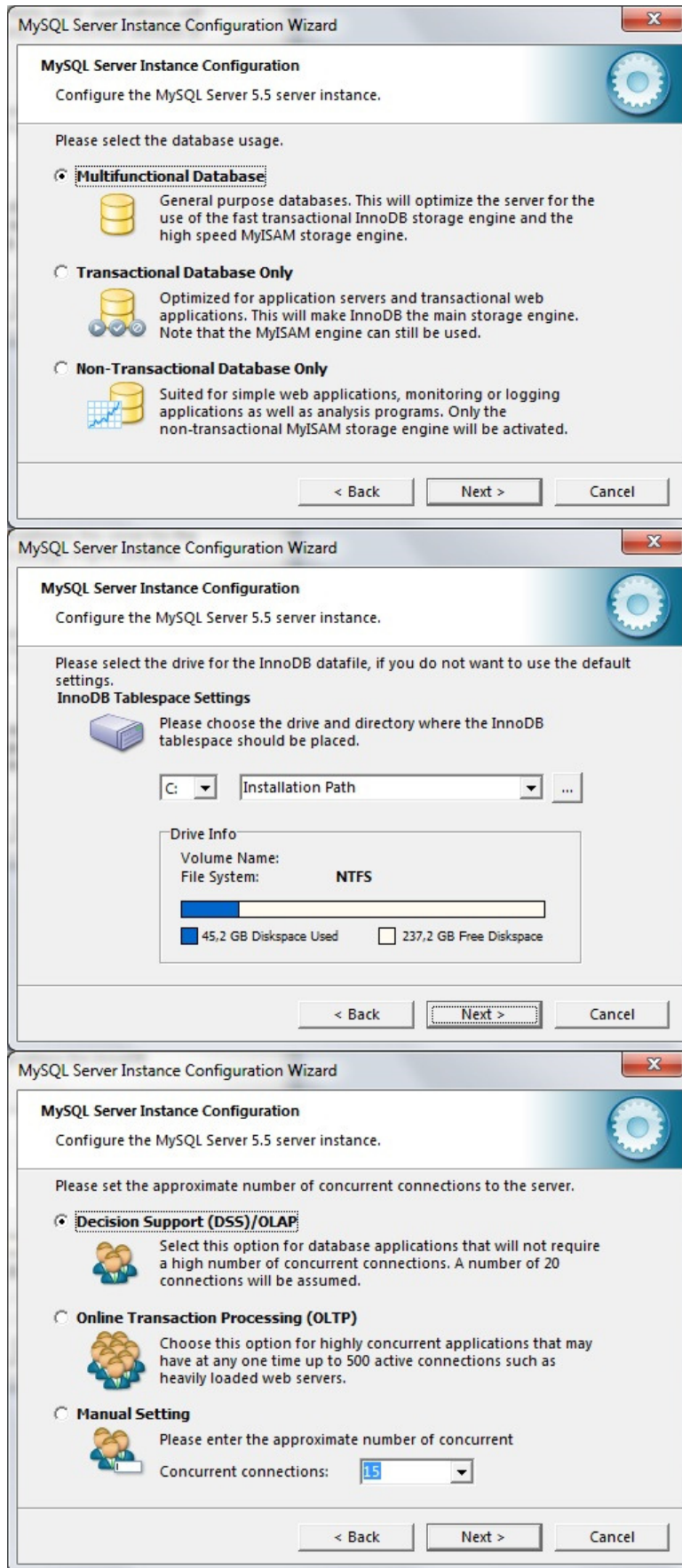
Κατεβάζουμε το αρχείο από την σελίδα <http://www.mysql.com/downloads/mysql/> και ακολουθούμε τις παρακάτω οδηγίες εγκατάστασης:

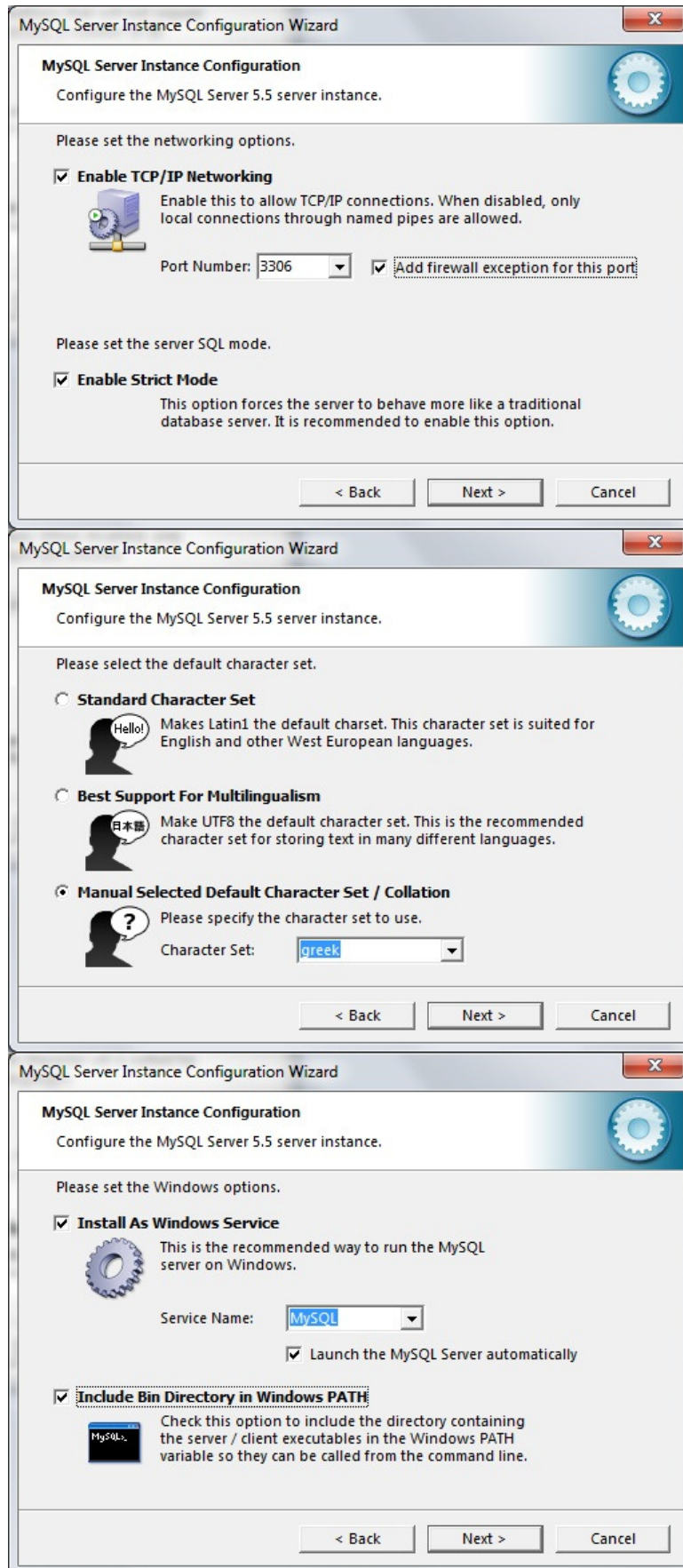


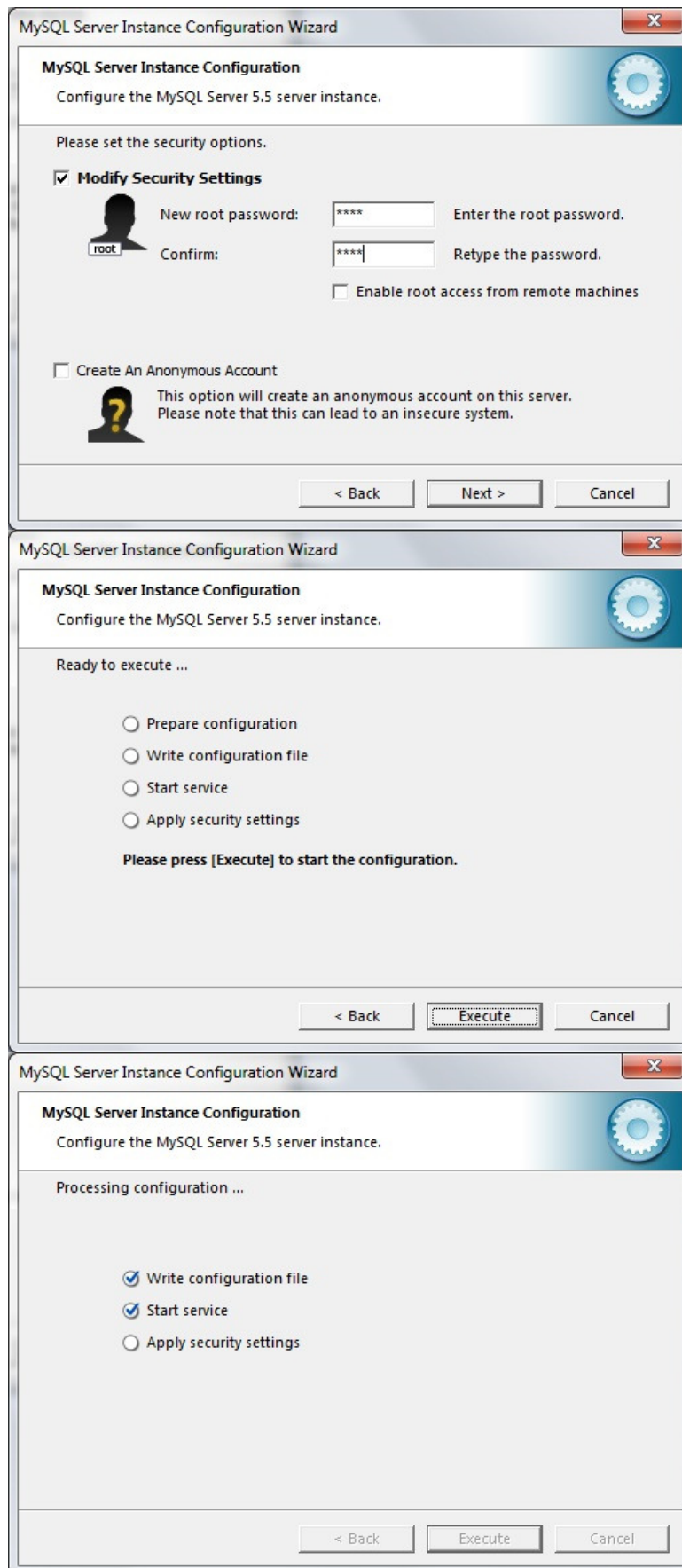


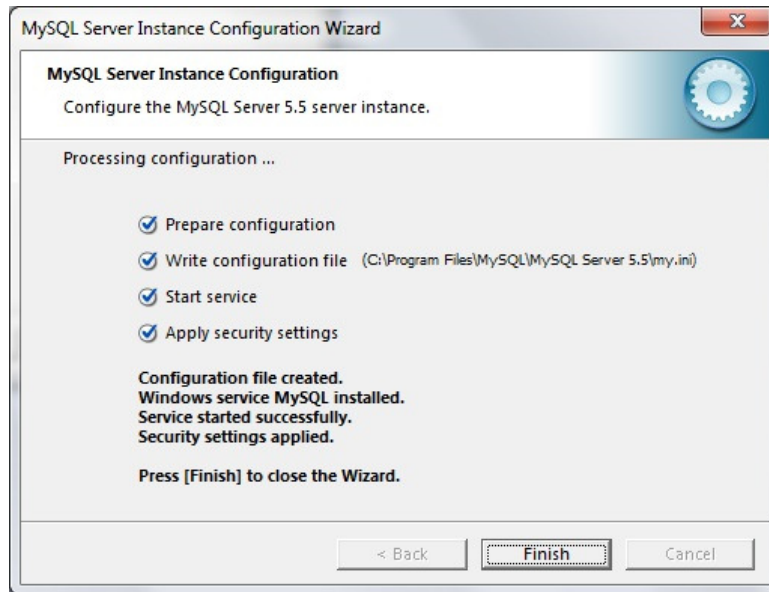








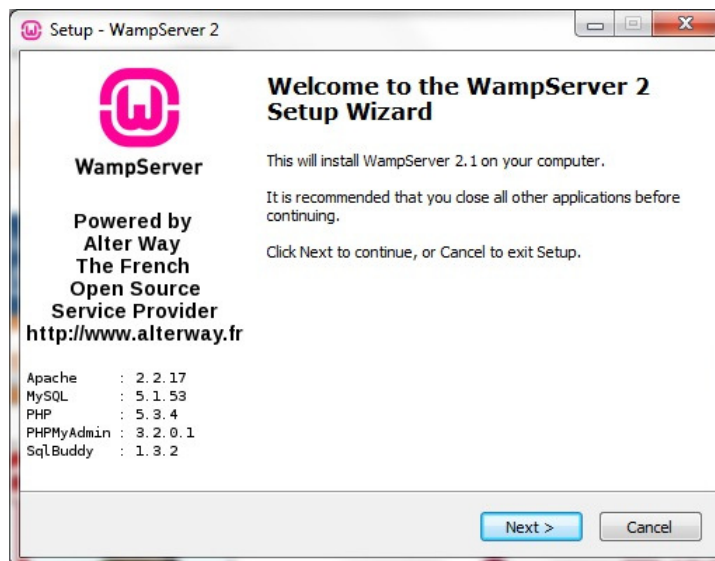


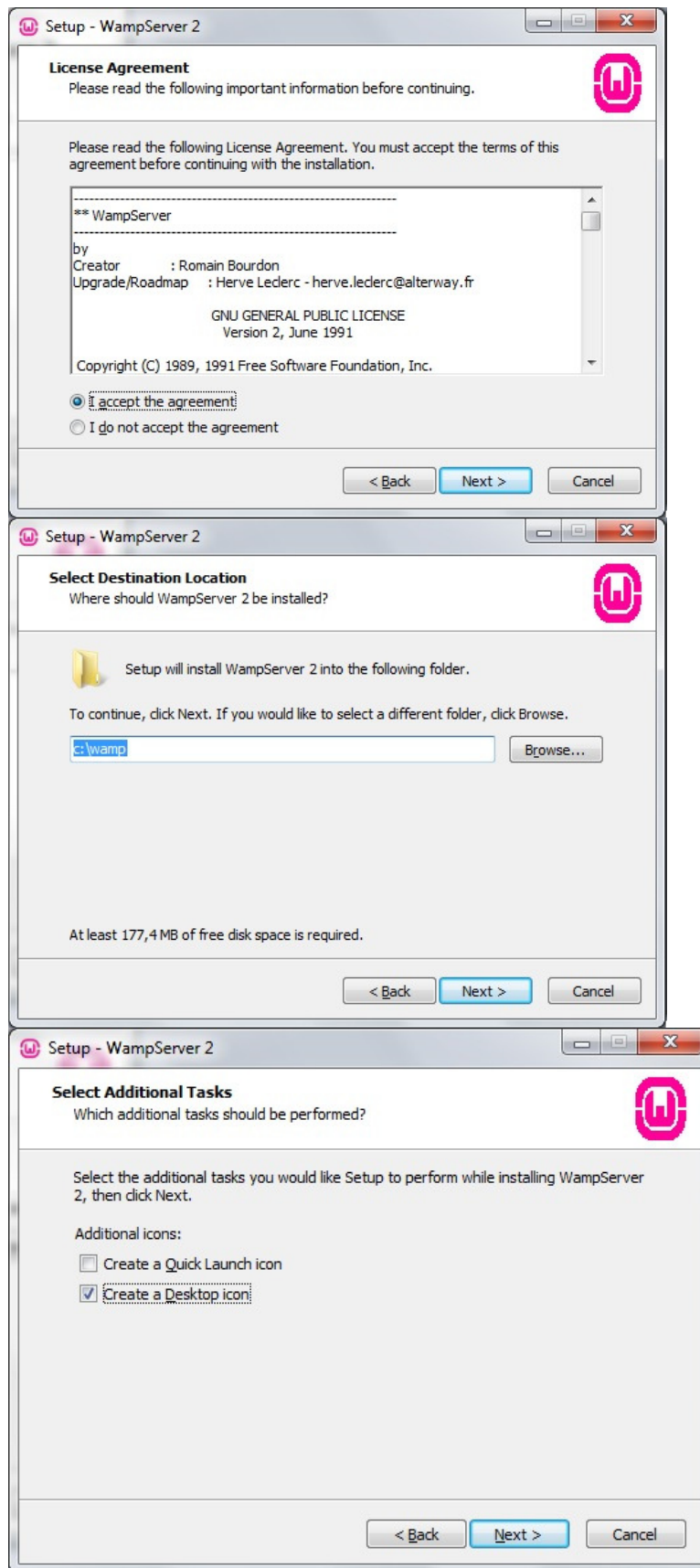


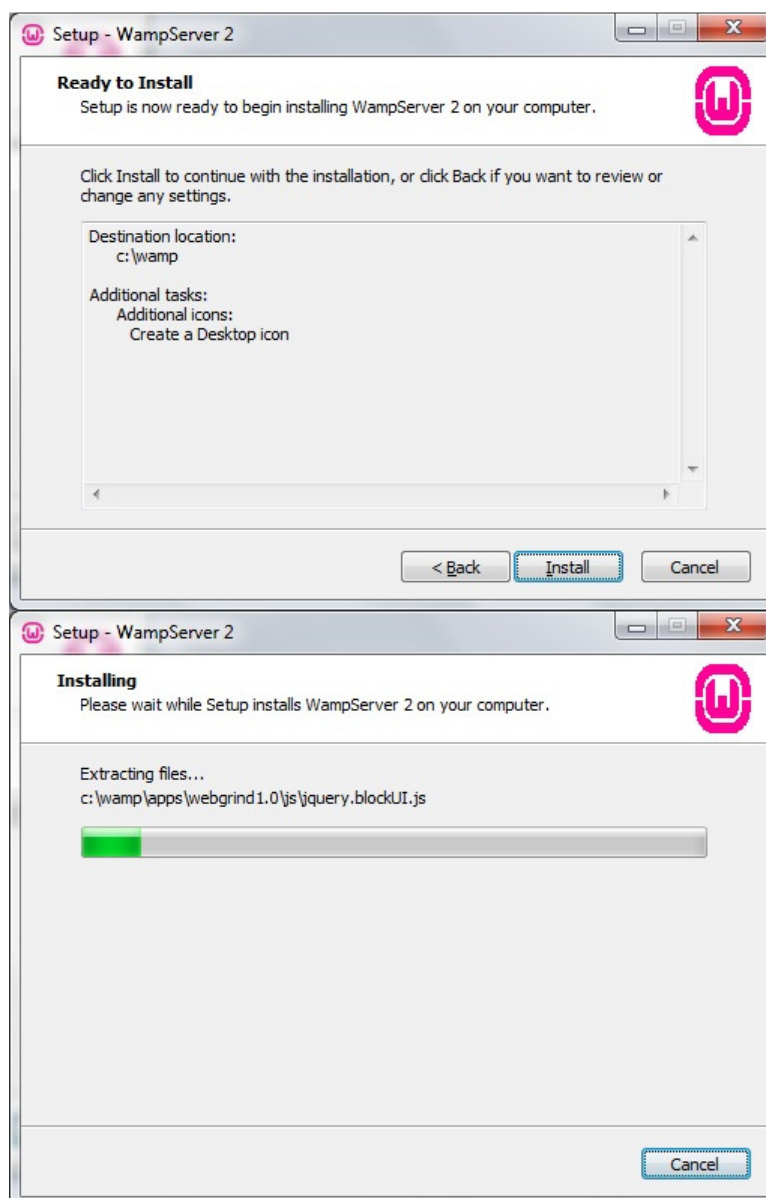
7.1.2. WampServer v.2

Εγκατάσταση:

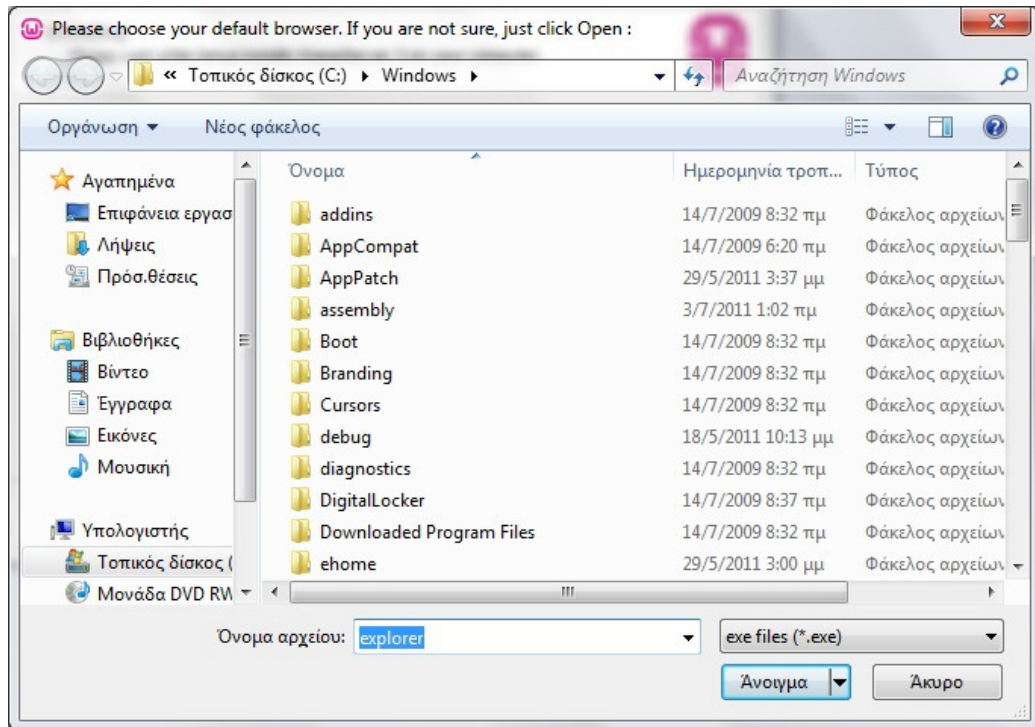
Κατεβάζουμε το αρχείο από την σελίδα <http://www.wampserver.com/en/download.php> και ακολουθούμε τις παρακάτω οδηγίες εγκατάστασης:



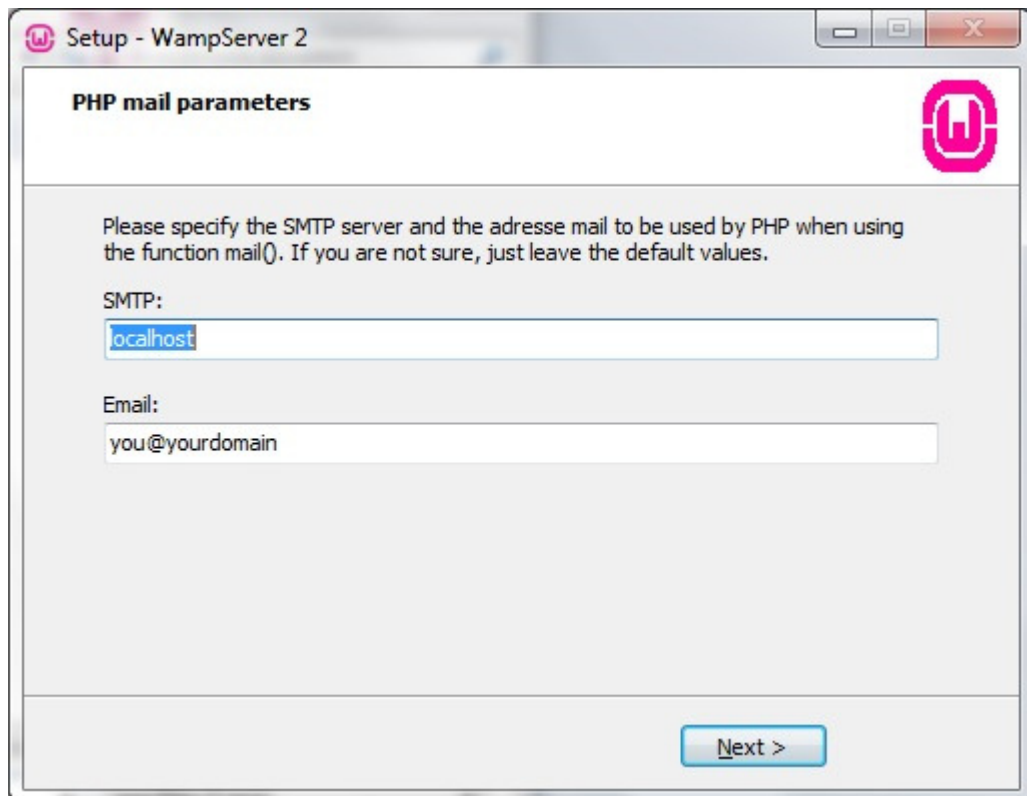


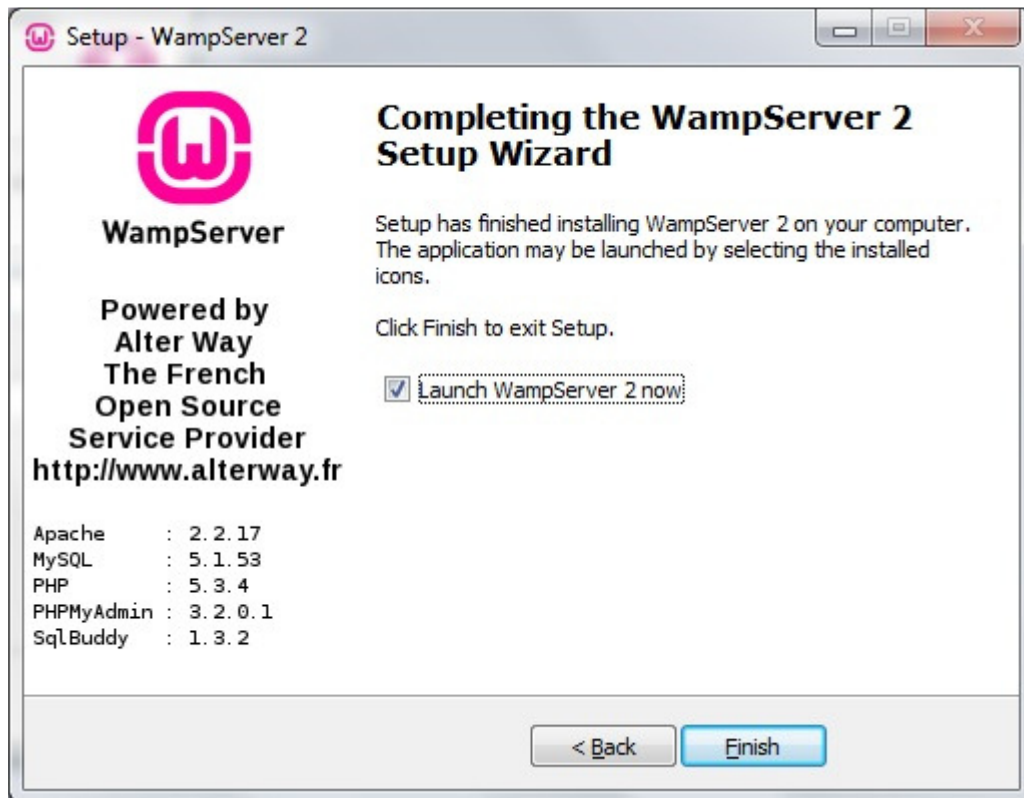


Online Σύστημα Διαχείρισης Αλυσίδας Supermarket

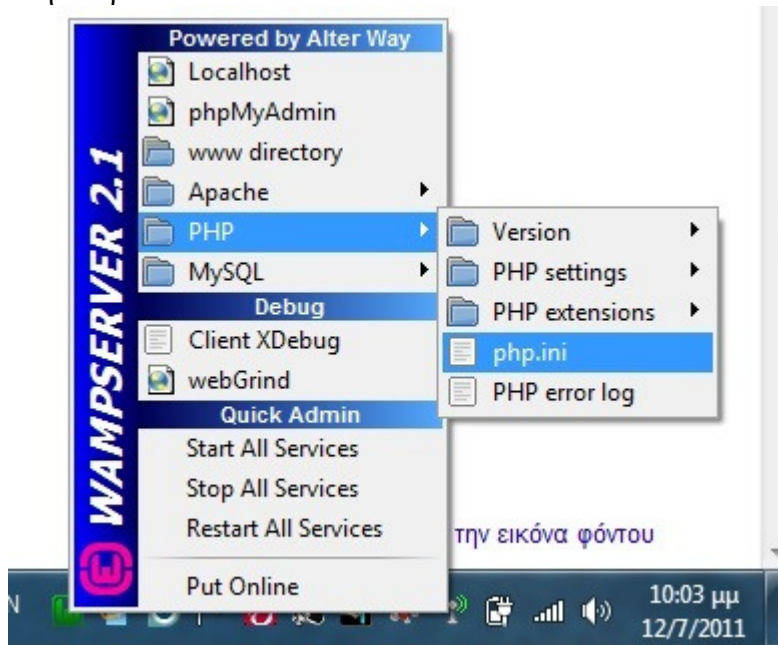


Όταν εμφανιστεί το παραπάνω παράθυρο πατήστε «Άνοιγμα» χωρίς να κάνετε κάποια αλλαγή!

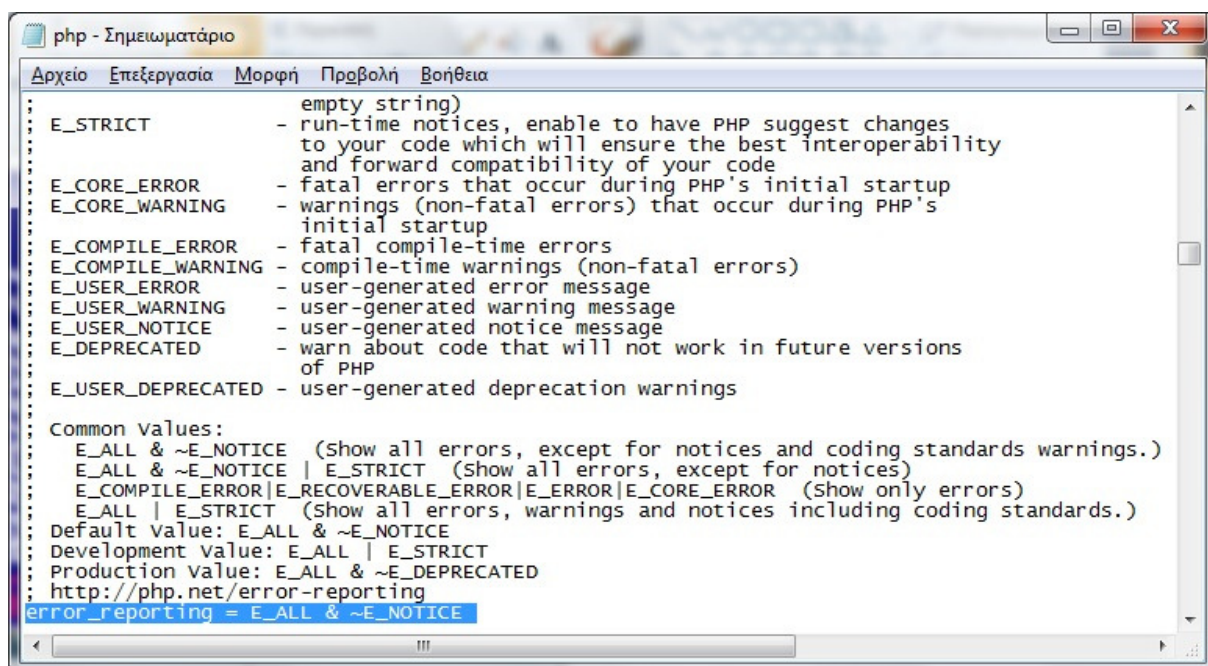




Στην συνέχεια τροποποιήστε το αρχείο php.ini κάνοντας δεξί κλικ στο εικονίδιο του wamp που βρίσκεται στο tray και επιλέγοντας το PHP->php.ini όπως φαίνεται και στην παρακάτω εικόνα:



Βρείτε την γραμμή του “error_reporting” και βάλτε μετά το = την τιμή “E_ALL & ~E_NOTICE” για να αποφύγουμε την εμφάνιση των ενοχλητικών warnings στις σελίδες μας.

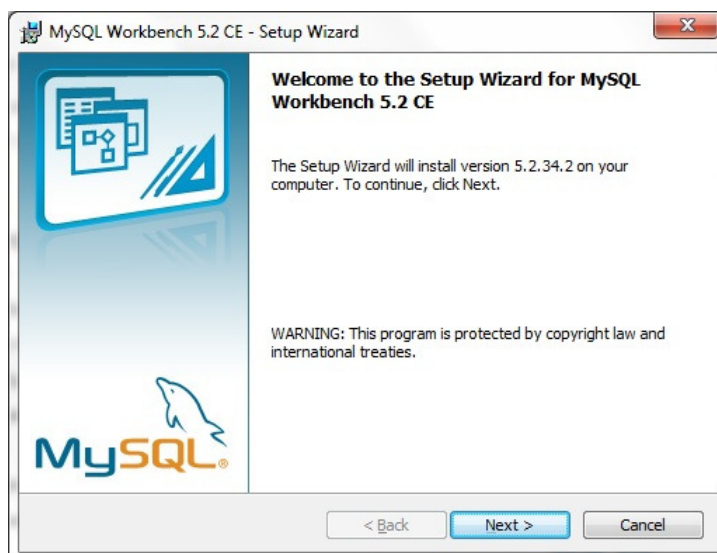


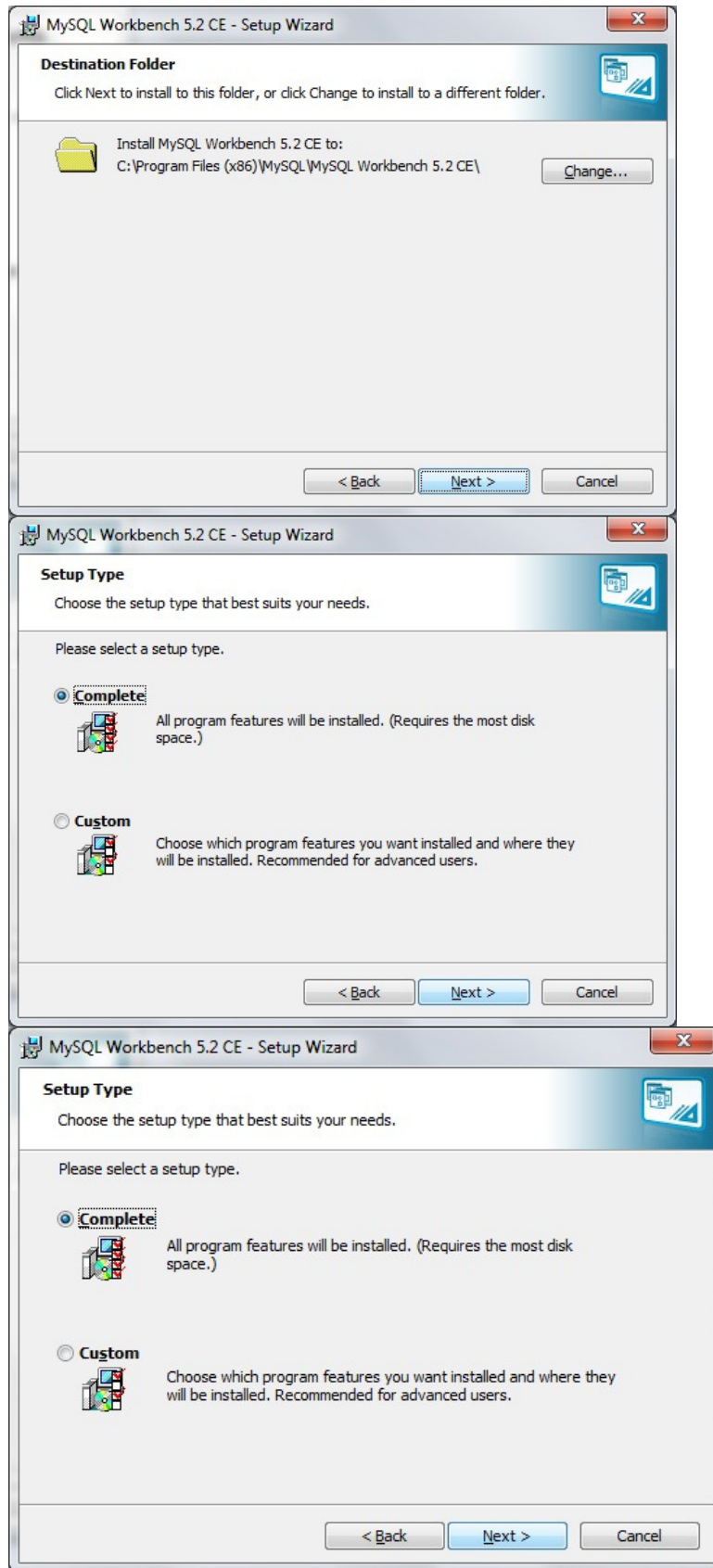
```
php - Σημειωματάριο
Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια
:
: empty string)
: E_STRICT - run-time notices, enable to have PHP suggest changes
: to your code which will ensure the best interoperability
: and forward compatibility of your code
: E_CORE_ERROR - fatal errors that occur during PHP's initial startup
: E_CORE_WARNING - warnings (non-fatal errors) that occur during PHP's
: initial startup
: E_COMPILE_ERROR - fatal compile-time errors
: E_COMPILE_WARNING - compile-time warnings (non-fatal errors)
: E_USER_ERROR - user-generated error message
: E_USER_WARNING - user-generated warning message
: E_USER_NOTICE - user-generated notice message
: E_DEPRECATED - warn about code that will not work in future versions
: of PHP
: E_USER_DEPRECATED - user-generated deprecation warnings
:
: Common values:
: E_ALL & ~E_NOTICE (Show all errors, except for notices and coding standards warnings.)
: E_ALL & ~E_NOTICE | E_STRICT (Show all errors, except for notices)
: E_COMPILE_ERROR|E_RECOVERABLE_ERROR|E_ERROR|E_CORE_ERROR (Show only errors)
: E_ALL | E_STRICT (Show all errors, warnings and notices including coding standards.)
: Default value: E_ALL & ~E_NOTICE
: Development value: E_ALL | E_STRICT
: Production value: E_ALL & ~E_DEPRECATED
: http://php.net/error-reporting
error_reporting = E_ALL & ~E_NOTICE
```

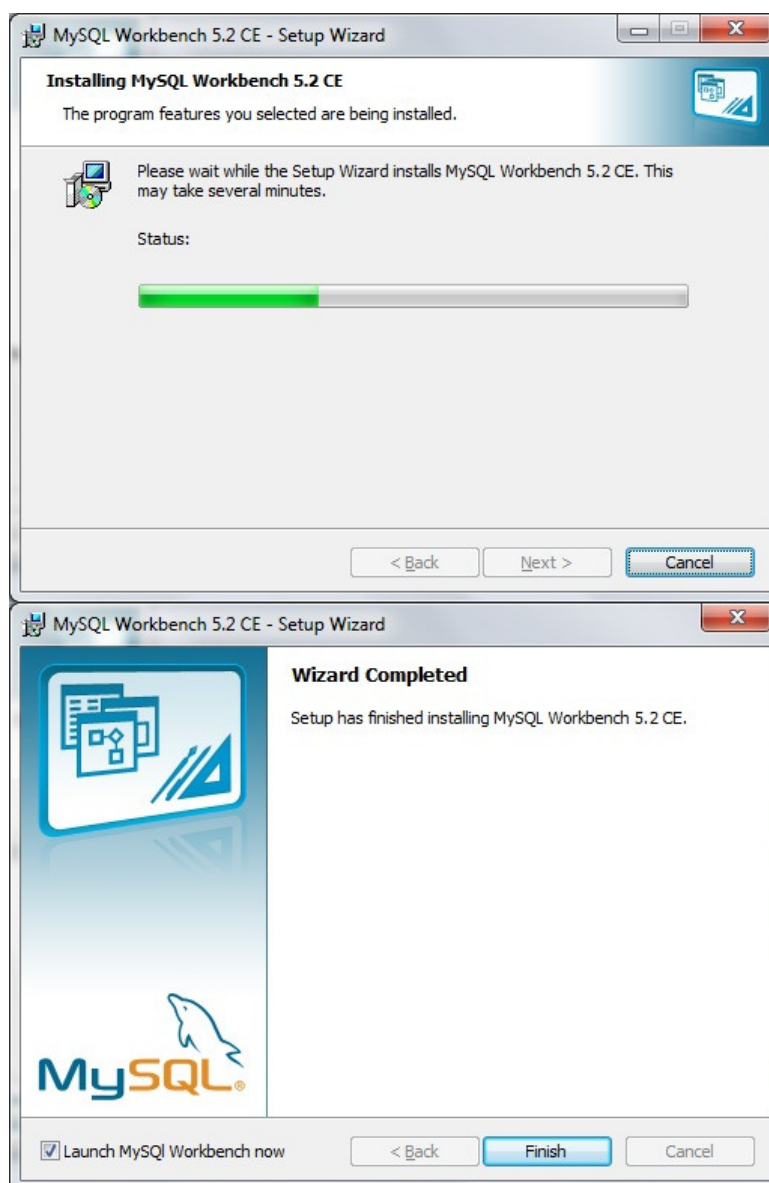
7.1.3. MySQL Workbench

Εγκατάσταση:

Κατεβάζουμε το αρχείο από την σελίδα <http://www.mysql.com/downloads/workbench/> και ακολουθούμε τις παρακάτω οδηγίες εγκατάστασης:







7.1.4. Favicon

Μια favicon (συντομογραφία του favorites icon), γνωστή επίσης ως ένα εικονίδιο συντόμευσης, Web εικονίδιο του site, εικονίδιο URL, ή εικονίδιο σελιδοδείκτη, είναι ένα αρχείο που περιέχει μία ή περισσότερες μικρές εικόνες, συνήθως 16 × 16 pixels, που συνδέεται με μια συγκεκριμένη τοποθεσία Web ή ιστοσελίδα. Ένας σχεδιαστής ιστοσελίδων μπορεί να δημιουργήσει μια τέτοια εικόνα και να την εγκαταστήσει σε μια ιστοσελίδα με διάφορα μέσα, και τα γραφικά προγράμματα περιήγησης στο Web θα κάνουν στη συνέχεια χρήση του. Τα προγράμματα περιήγησης που παρέχουν υποστήριξη favicon συνήθως το εμφανίζουν στη γραμμή διευθύνσεων του περιηγητή και δίπλα στο όνομα της σελίδας σε μια λίστα με τους σελιδοδείκτες. Τα προγράμματα περιήγησης που υποστηρίζουν έγγραφα με καρτέλες διαπαφής δείχνουν συνήθως τα favicon μιας σελίδας δίπλα στον τίτλο της σελίδας στην καρτέλα, και site-specific προγράμματα περιήγησης χρησιμοποιούν το favicon ως εικονίδιο στην επιφάνεια εργασίας.

Ιστορία

Τον Μάρτιο του 1999, η Microsoft κυκλοφόρησε τον Internet Explorer 5, στον οποίο υποστηρίζονται τα favicons για πρώτη φορά. Αρχικά, το favicon ήταν ένα αρχείο με όνομα favicon.ico τοποθετημένο στον root κατάλογο ενός δικτυακού τόπου (π.χ. <http://en.wikipedia.org/favicon.ico>).

Χρησιμοποιήθηκε σε αγαπημένα του Internet Explorer (bookmarks) και δίπλα από το URL στη γραμμή διεύθυνσης αν η σελίδα αυτή είχε γίνει σελιδοδείκτης. Σαν αποτέλεσμα ήταν ότι ο αριθμός των επισκεπτών που έχουν σελιδοδείκτη της σελίδα να μπορεί να εκτιμηθεί από τα αιτήματα των favicon. Η τεχνική αυτή έχει καταστεί άνευ αντικειμένου δεδομένου ότι όλα τα σύγχρονα προγράμματα περιήγησης υποστηρίζουν το favicon χωρίς bookmarking.

7.2. Τεχνικές διαδικτύου που χρησιμοποιήθηκαν

7.2.1. CSS (Cascading Style Sheets)

Η CSS (Cascading Style Sheets-Διαδοχικά Φύλλα Στυλ) ή (αλληλουχία φύλλων στυλ) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στυλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την html. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η χρήση της CSS κρίνεται ως απαραίτητη.

Η αλληλουχία εφαρμογής των φύλλων στυλ:

Για ένα έγγραφο πχ xhtml θα υπάρχουν παραπάνω από ένα φύλλα στυλ τα οποία περιέχουν δηλώσεις για την εμφάνιση ενός συγκεκριμένου στοιχείου. Το Φύλλο στυλ που εφαρμόζεται σε ένα έγγραφο μπορεί να προέρχεται από :

- το συγγραφέα μιας ιστοσελίδας
- το χρήστη του πλοηγού
- τον ίδιο τον πλοηγό, αν έχει το δικό του προκαθορισμένο φύλλο στυλ .

Συνεπώς για ένα xhtml στοιχείο θα υπάρχουν παραπάνω από μια δηλώσεις που πιθανόν να είναι συγκρουόμενες.

7.2.2. Ajax (asynchronous JavaScript and XML)

Ajax δεν είναι μια τεχνολογία, αλλά μια ομάδα τεχνολογιών. Ajax χρησιμοποιεί ένα συνδυασμό HTML και CSS για τη σήμανση και την ενημέρωση στυλ. Το DOM είναι προσβάσιμες με JavaScript για δυναμική εμφάνιση, και να επιτρέψει στο χρήστη να αλληλεπιδράσει με τις πληροφορίες που παρουσιάζονται. JavaScript και το αντικείμενο XMLHttpRequest παρέχει μια μέθοδο για την ανταλλαγή δεδομένων ασύγχρονα μεταξύ του browser και του server να αποφύγει την πλήρη επαναφορτώσεις σελίδων.

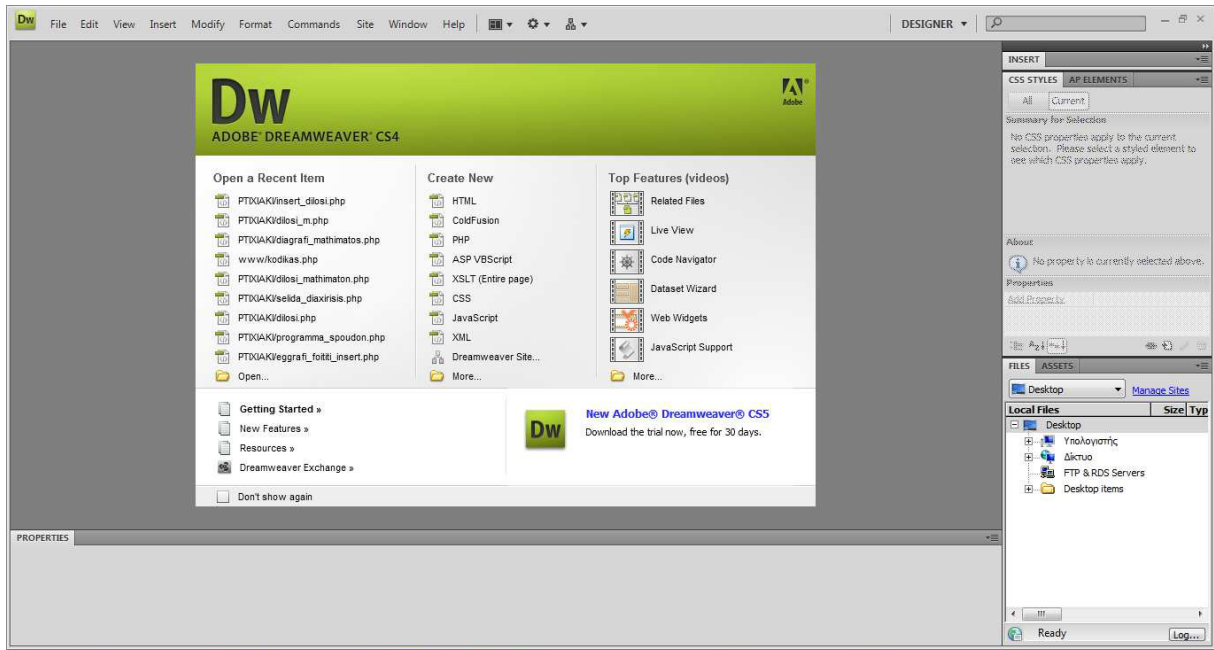
Ο όρος Ajax έχει έρθει να αντιπροσωπεύει μια ευρεία ομάδα των τεχνολογιών του Παγκοσμίου Ιστού που μπορούν να χρησιμοποιηθούν για την υλοποίηση μιας εφαρμογής web που επικοινωνεί με ένα διακομιστή στο παρασκήνιο, χωρίς να παρεμβαίνει με την τρέχουσα κατάσταση της σελίδας. Στο άρθρο που επινόησε τον όρο Ajax, Jesse James Garrett εξήγησε ότι έχουν ενσωματωθεί οι ακόλουθες τεχνολογίες:

- HTML (ή XHTML) και CSS για την παρουσίασή
- Το Μοντέλο Αντικειμένου Εγγράφου (DOM) για τηνδυναμική προβολή της και την αλληλεπίδραση με τα δεδομένα
- XML για την ανταλλαγή δεδομένων, και XSLT για τη χειραγώγηση του
- Το αντικείμενο XMLHttpRequest για ασύγχρονη επικοινωνία
- JavaScript για να φέρει αυτές τις τεχνολογίες μαζί

7.3. Extra εργαλεία

7.3.1. Adobe Dreamweaver

Το Adobe DreamWeaver χρησιμοποιείται ευρέως στην κατασκευή ιστοσελίδων με την χρήση κώδικα Html και XHTML. Ο χρήστης μπορεί να δημιουργήσει αρχεία τύπου php, html κ.α. Μπορεί επίσης να δημιουργεί ένα περιβάλλον είτε με τη χρήση κώδικα είτε γραφικά.



Εικόνα: Το λογισμικό Adobe DreamWeaver

Στην παρακάτω εικόνα φαίνεται αρχικά το περιβάλλον δημιουργίας ιστοσελίδας το οποίο είναι χωρισμένο σε δύο τομείς, έναν γραφικό και έναν για κώδικα. Στην Html μπορούμε να βάλουμε και php κώδικα αλλά επίσης μπορούμε να κάνουμε και το αντίθετο. Το Dreamweaver παρέχει αρκετά εργαλεία για τη δημιουργία και ανάπτυξη μίας δυναμικής ιστοσελίδας. Μερικά από αυτά είναι οι έτοιμες βιβλιοθήκες όπου μπορούμε να βρούμε έτοιμα κουμπιά, λίστες, φόρμες και τα συναφή που χρειάζεται μια φόρμα και πολλά άλλα. Μπορούμε να μορφοποιήσουμε την εφαρμογή μας με την χρήση css και να αποθηκεύσουμε την μορφοποίηση αυτή για μελλοντική χρήση. Όμως το Dreamweaver δεν μας υποδεικνύει τυχόν λάθη στο κώδικα που παράγουμε. Μπορούμε μόνο να τα εντοπίσουμε στον browser όπου ελέγχουμε την εφαρμογή μας αν λειτουργεί και τσεκάρουμε το αποτέλεσμα μας.

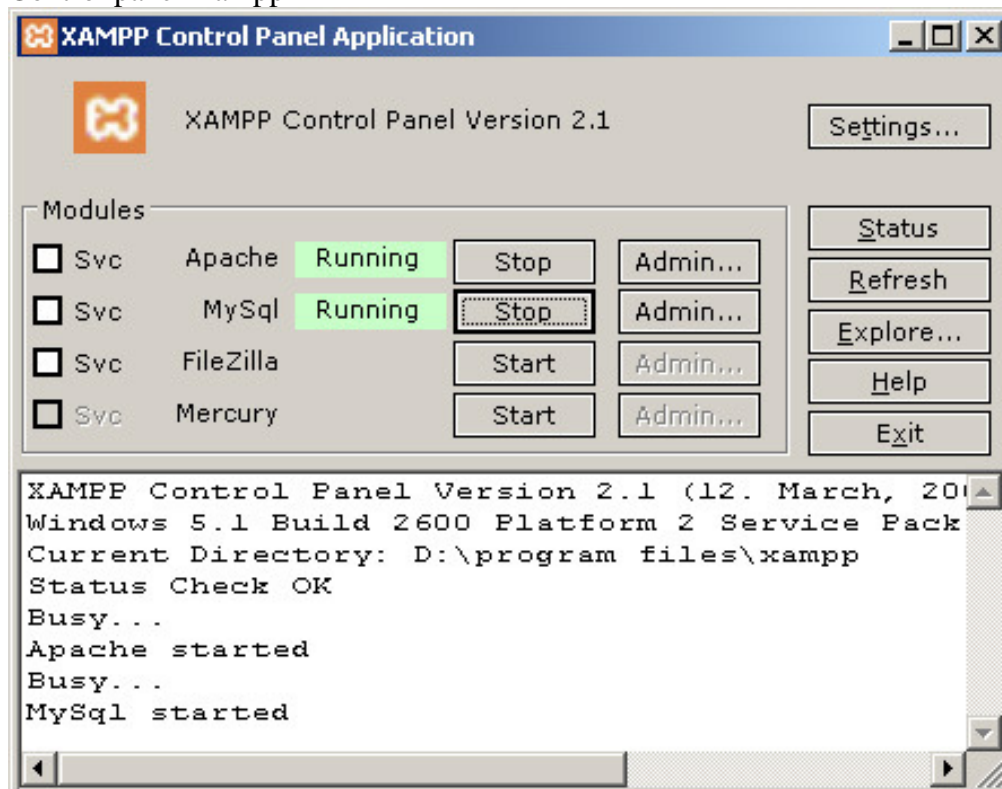
7.3.2. Xampp

Η χρήση του είναι ελεύθερη και η πρόσβαση γίνεται μέσα από την διεύθυνση:
<http://www.apachefriends.org/en/xampp.html>.

Εγκατάσταση XAMPP



Control panel Xampp



ΠΑΡΟΥΣΙΑΣΗ



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης
Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Τίτλος: ΔΗΜΙΟΥΡΓΙΑ ONLINE ΣΥΣΤΗΜΑΤΟΣ ΓΙΑ ΑΛΥΣΙΔΑ SUPERMARKET

Κεφαλογιάννης Εμμανουήλ (ΑΜ: 1551)
Ξανθουδάκης Αριστοτέλης (ΑΜ: 1667)

Επιβλέπων καθηγητής : Παπαδάκης Νικόλαος

ΕΙΣΑΓΩΓΗ

- Δημιουργία ιστοσελίδας και βάσης δεδομένων SQL για διαχείριση αλυσίδας Super-Market
- MySQL, HTML, PHP, APACHE

ΥΛΟΠΟΙΗΣΗ

- **Σχεδίαση Βάσης Δεδομένων**
 - Μοντέλο Οντοτήτων-Συσχετίσεων
 - Σχεσιακό Μοντέλο
 - Κώδικας MySQL
- **Δημιουργία Ιστοσελίδας**
 - Χρήση τεχνολογιών
 - Ασφάλεια
 - Σύνδεση με βάση δεδομένων

ΕΡΓΑΛΕΙΑ

WampServer 2.1a



MySQL Server 5.5



Notepad++ 5.9.2



Command Prompt (Windows 7)



Mozilla Firefox 4.0.1

ΤΕΧΝΟΛΟΓΙΕΣ 1

❖ HTML 4.0.1

❖ Apache 2.2.17

❖ Php 5.3.3

❖ Mysql 5.5.8

- Open Source Software
- Άψογη συνεργασία μεταξύ τους
- Δημοφιλείς
- Υποστήριξη από Windows, Linux



ΤΕΧΝΟΛΟΓΙΕΣ 2

● JavaScript

- Asynchronous JavaScript and XML (AJAX)
- MessageBox

● Cascading Style Sheets (CSS)

- Χαρακτηριστικά
- Χρώματα

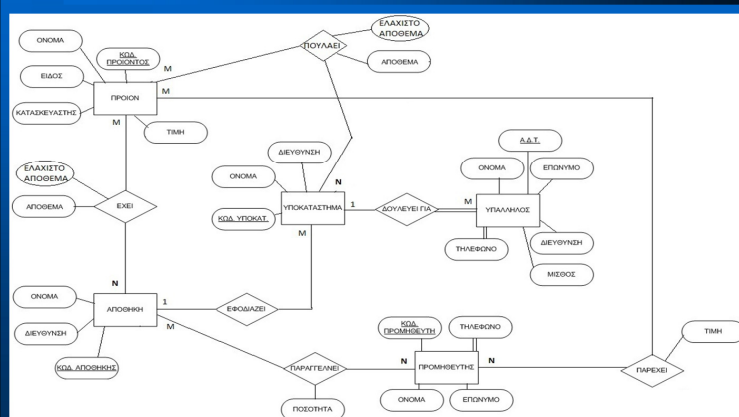
APACHE SERVER

- Εκτελεί κώδικα PHP
- Δυναμική σύνδεση
- Έλεγχος σφαλμάτων
- Συμβατότητα με πολλές Β.Δ.

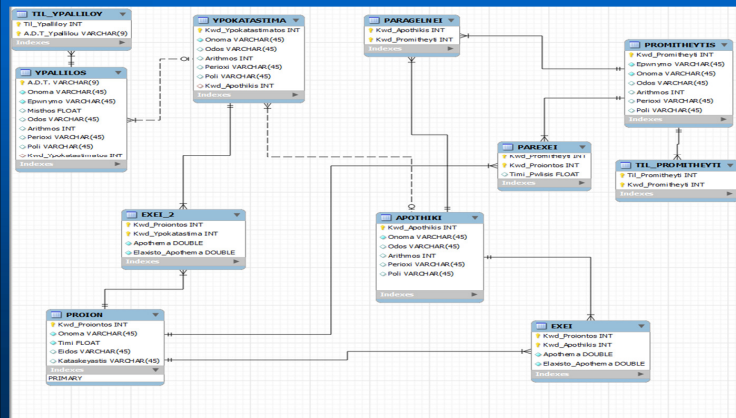
PHP (Hypertext Preprocessor)

- **Server-side scripting**
 - Εκτελείται στον web-server
- **Interpreter**
 - Μεταγλώττιση σε HTML
- **Command line scripting**
 - Απαιτείται μόνο ο μεταγλωττιστής

DATABASE



DATABASE



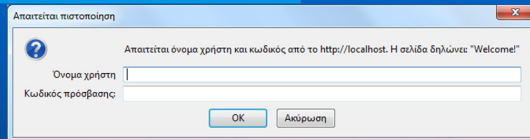
SQL ΚΩΔΙΚΑΣ

- **CREATE DATABASE** [όνομα βάσης]
- **CREATE TABLE** [όνομα πίνακα] ([όνομα πεδίου] τύπος πεδίου, **PRIMARY KEY** (πεδίο), **FOREIGN KEY** (πεδίο) **REFERENCES** [όνομα πίνακα] (πεδίο))
- **CASCADE || NO ACTION**

ΔΟΜΗ

1. Επιβεβαίωση αυθεντικότητας
2. Σύνδεση στην βάση δεδομένων
3. Λειτουργίες
4. Κλείσιμο σύνδεσης

ΑΥΘΕΝΤΙΚΟΤΗΤΑ



- Αποφυγή SQL Injection
- Ένας χρήστης
- Μη προσπελάσιμος πηγαίος κώδικας

Σύνδεση στην SQL

1. Στοιχεία σύνδεσης MySQL
 - Server
 - Username
 - Password
2. Σύνδεση στη MySQL
 - Επιλογή βάσης

ΛΕΙΤΟΥΡΓΙΕΣ 1

- **Αναζήτηση**
 - `SELECT column_name(s) FROM table_name`
- **Προσθήκη**
 - `INSERT INTO table_name VALUES (value1, value2,...)`
- **Επεξεργασία**
 - `UPDATE table_name SET column1=value, column2=value2,... WHERE some_column=some_value`
- **Διαγραφή**
 - `DELETE FROM table_name WHERE some_column=some_value`

ΛΕΙΤΟΥΡΓΙΕΣ 2

- **Αντίγραφα Ασφαλείας**
 - One-click Backup
 - Restore
- **Μηχανισμός αυτόματης παραγγελίας**
 - Αποθήκης
 - Υποκαταστήματος
 - Προμηθευτή

ΚΛΕΙΣΙΜΟ ΣΥΝΔΕΣΗΣ

- Αποδέσμευση μνήμης με `mysql_free_result($result)`;
- Αποσύνδεση από την βάση με `mysql_close($dbhandle)`;

ΑΠΟΤΕΛΕΣΜΑΤΑ

- **Ταχύτητα**
 - Εκτενείς χρήση AJAX
 - Μη χρήση FLASH applets
 - Μη χρήση JAVA applets
- **Ευχρηστία**
 - Κανόνες GUI
 - Pagation
 - Ελαχιστοποίηση ενεργειών
 - Συντομεύσεις