



ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΠΟΛΥΜΕΣΩΝ

«ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΑΠΟΘΗΚΩΝ»



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Εισηγητής: Καλογεράκης Παναγιώτης

*Σπουδαστές: Βαφέας Χρήστος 907
Κακουράκης Αντώνης 957*

ΗΡΑΚΛΕΙΟ 2008

ΠΕΡΙΕΧΟΜΕΝΑ

Περιγραφή τυχαικής εργασίας	3
Γενικά-Εισαγωγικά	4
Περιγραφή των εργαλείων που χρησιμοποιήθηκαν για να αναπτυχθεί η εφαρμογή	9
1. Xampp v 2.4	9
2. PhpMyAdmin	10
3. Php Maker 5	12
4. Macromedia Dreamweaver MX 2004	13
5. Photoshop CS2	13
Περιγραφή της βάσης δεδομένων	14
1. Πίνακας warehouse	15
2. Πίνακας item	16
3. Πίνακας item_cat	17
4. Πίνακας item_subcat	18
5. Πίνακας unit_table	19
6. Πίνακας product_quantity	20
7. Πίνακας user	21
8. Πίνακας customer	22
9. Πίνακας supplier	23
10. Πίνακας cause_trans	24
11. Πίνακας product_trans	26
12. Πίνακας bounds_trans	27
Manual εφαρμογής	28
1. login	28
2. Αρχικό μενού πινάκων	29
3. Πίνακας warehouse	30
4. Πίνακας item	31
5. Πίνακας item_cat	34
6. Πίνακας item_subcat	35
7. Πίνακας unit_table	36
8. Πίνακας product_quantity	37
9. Πίνακας user	38
10. Πίνακας customer	39
11. Πίνακας supplier	40
12. Πίνακας product_trans	41
13. Πίνακας bounds_trans	44
14. Reports	45
Παράρτημα	49
Πηγές-Βιβλιογραφία	65

Περιγραφή Πτυχιακής Εργασίας:

Η παρούσα πτυχιακή αποσκοπεί στην ανάλυση και υλοποίηση λογισμικού που θα μπορούσε να χρησιμοποιηθεί από μία εμπορική εταιρία για την (απλοποιημένη) διαχείριση των αποθηκών της.

Η εφαρμογή θα πρέπει να είναι διαδικτυακή και να βασίζεται πάνω σε σχεσιακή βάση δεδομένων.

Η εταιρία θεωρούμε ότι έχει αποθήκες σε διάφορα γεωγραφικά σημεία.

Εμπορεύεται ένα μεγάλο σύνολο ειδών, τα οποία έχουν χωριστεί σε κατηγορίες και υποκατηγορίες. Όλες οι αποθήκες της εταιρίας μπορούν να διαχειριστούν το σύνολο των ειδών αυτής.

Σε κάθε αποθήκη γίνονται κινήσεις αγορών από προμηθευτές, καθώς και πωλήσεων σε πελάτες. Το σύστημα θα διαχειρίζεται τις κινήσεις αυτές μόνο ποσοτικά και θα τηρεί το τρέχον απόθεμα ανά είδος κάθε αποθήκης της εταιρίας.

Θα υποστηρίζει επίσης την δυνατότητα παραγγελιών προϊόντων.

Το σύστημα θα παρέχει την δυνατότητα αναζήτησης του τρέχοντος αποθέματος τόσο σε επίπεδο αποθήκης όσο και σε επίπεδο εταιρίας.

Θα παρέχεται επίσης η δυνατότητα εσωτερικής διακίνησης μεταξύ αποθηκών. Εάν μια αποθήκη θέλει συγκεκριμένη ποσότητα από ένα και αυτή υπάρχει σε κάποια άλλη αποθήκη, μπορεί να την πάρει από την αποθήκη αυτή.

Χρήστες του συστήματος θα είναι:

1. Ο αποθηκάριος της κάθε αποθήκης ο οποίος θα κάνει κινήσεις στις αποθήκες και θα μπορεί να αναζητάει για συγκεκριμένο είδος ή κατηγορία ειδών το ελεύθερο απόθεμα της κάθε αποθήκης της εταιρίας.
2. Τα μέλη της διοίκησης της εταιρίας, που μέσω στατιστικών αναφορών και αναζητήσεων θα μπορεί να παρακολουθεί το τρέχον απόθεμα ανά αποθήκη ή και συνολικά στην εταιρία.
3. Οι administrators που θα έχουν απεριόριστα δικαιώματα, ώστε ανά πάσα στιγμή να μπορούν να διορθώνουν τυχόν λάθη, να αλλάζουν όπου και όταν χρειάζεται τμήματα της εφαρμογής και το σημαντικότερο, να ορίζουν τα δικαιώματα των νέων χρηστών ώστε να μπορούν κι εκείνοι να έχουν πρόσβαση στη βάση.

Επιλέξαμε να δημιουργήσουμε την KA.VA Ltd. , μια εικονική εταιρία εισαγωγής-εξαγωγής και προμήθειας ποτών.

Γενικά-Εισαγωγικά

ΤΙ ΕΙΝΑΙ ΤΟ INTERNET

Το Internet είναι ένα πλέγμα από εκατομμύρια διασυνδεδεμένους υπολογιστές που εκτείνεται σχεδόν σε κάθε γωνιά του πλανήτη και παρέχει τις υπηρεσίες του σε εκατομμύρια χρήστες.

Αποτελεί ένα “Παγκόσμιο Ηλεκτρονικό Χωριό”, οι “κάτοικοι” του οποίου, ανεξάρτητα από υπηκοότητα, ηλικία, θρήσκευμα και χρώμα, μοιράζονται πληροφορίες και ανταλλάσσουν ελεύθερα απόψεις πέρα από γεωγραφικά και κοινωνικά σύνορα. Σύμφωνα με τις σχετικές εκτιμήσεις, αυτός ο παγκόσμιος ιστός υπολογιστών και χρηστών αριθμεί σήμερα πάνω από δέκα εκατομμύρια υπολογιστές και εκατό εκατομμύρια χρήστες, ενώ επεκτείνεται διαρκώς με εκθετικούς ρυθμούς. Αναμένεται ότι το 2000 το Internet θα εξυπηρετεί περισσότερους από ένα δισεκατομμύριο χρήστες

ΤΙ ΕΙΝΑΙ Η PHP

Η PHP είναι μια γλώσσα script από την πλευρά του διακομιστή , σχεδιασμένη ειδικά για το Web .Μέσα σε μια HTML σελίδα μπορείτε να ενσωματώσετε PHP κώδικα, που θα εκτελείται κάθε φορά που θα επισκέπτεστε τη σελίδα. Ο PHP κώδικας μεταφράζεται στο Web διακομιστή και δημιουργεί HTML ή άλλη έξοδο που θα δει ο επισκέπτης. Η PHP δημιουργήθηκε το 1994 και ήταν αρχικά η δουλειά ενός ατόμου, του **Rasmus Lerdorf**. Υιοθετήθηκε και από άλλα ταλαντούχα άτομα και έχει περάσει από τρεις βασικές εκδόσεις. Τον Ιανουάριο του 2001 ήταν σε χρήση σχεδόν σε πέντε εκατομμύρια τομείς παγκόσμια και αυτός ο αριθμός μεγαλώνει γρήγορα.

Η PHP είναι ένα προϊόν ανοιχτού κώδικα. Θα έχετε πρόσβαση στον κώδικα προέλευσης. Μπορείτε να τον χρησιμοποιήσετε , να τον αλλάξετε και να τον αναδιανείμετε , χωρίς χρέωση.

Η PHP αρχικά σήμαινε *Personal Home Page* (προσωπική αρχική σελίδα), αλλά άλλαξε σύμφωνα με την σύμβαση GNU και τώρα σημαίνει *PHP Hypertext Preprocessor* (προεπεξεργαστής κειμένου PHP).

Η τρέχουσα βασική έκδοση της PHP είναι η 5. Αυτή η έκδοση έχει μερικές μεγάλες βελτιώσεις στη γλώσσα.

ΤΙ ΕΙΝΑΙ Η MySQL

Η MySQL είναι ένα πολύ γρήγορο και δυνατό , σύστημα *διαχείρισης βάσεων δεδομένων*. Μια βάση δεδομένων σας επιτρέπει να αποθηκεύετε , να αναζητάτε , να ταξινομείτε και να ανακαλείτε τα δεδομένα αποτελεσματικά. Ο MySQL διακομιστής ελέγχει την πρόσβαση στα δεδομένα σας , για να μπορούν να δουλεύουν πολλοί χρήστες ταυτόχρονα, για να παρέχει γρήγορη πρόσβαση και να διασφαλίζει ότι μόνο πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση. Συνεπώς η MySQL είναι ένας πολυνηματικός διακομιστής πολλαπλών χρηστών. Χρησιμοποιεί την SQL (Structured Query Language) την τυπική γλώσσα ερωτημάτων για βάσεις δεδομένων, παγκόσμια . Η MySQL είναι διαθέσιμη από το 1996 αλλά η ιστορία της ξεκινά από το 1979 .

ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΗΣ PHP

Κάποιοι από τους βασικούς ανταγωνιστές της PHP είναι ο Perl, Microsoft Active Server Pages (ASP) , Java Server Pages (JSP) και Allaire Cold Fusion .

Σε σύγκριση με αυτά τα προϊόντα, η PHP έχει πολλά πλεονεκτήματα όπως :

- Υψηλή απόδοση
- Διασυνδέσεις με πολλά διαφορετικά συστήματα βάσεων δεδομένων
- Ενσωματωμένες βιβλιοθήκες για πολλές συνηθισμένες Web διαδικασίες
- Χαμηλό κόστος
- Ευκολία μάθησης και χρήσης
- Μεταφερσιμότητα
- Διαθεσιμότητα του κώδικα προέλευσης

Ακολουθεί μια πιο λεπτομερή περιγραφή αυτών των πλεονεκτημάτων:

✓ Απόδοση

Η PHP είναι πολύ αποτελεσματική. Με ένα φθηνό διακομιστή μπορείτε να εξυπηρετήσετε εκατομμύρια επισκέψεων καθημερινά. Οι δοκιμές που δημοσιεύθηκαν από την Zend Technologies (<http://www.zend.com>), δείχνουν ότι η PHP ξεπερνά τους ανταγωνιστές της.

✓ Ολοκλήρωση με Βάσεις δεδομένων

Η PHP έχει εγγενείς συνδέσεις για πολλά συστήματα βάσεων δεδομένων. Εκτός από την MySQL , μπορείτε να συνδεθείτε κατευθείαν με τις βάσεις δεδομένων PostgreSQL , mSQL , Oracle , dbm , filePro , Informix , InterBase , Sybase , κ.α. Χρησιμοποιώντας το *Open Database Connectivity Standard (ODBC)* μπορείτε να συνδεθείτε σε οποιαδήποτε βάση δεδομένων παρέχει ένα πρόγραμμα οδήγησης ODBC . Αυτό περιλαμβάνει και τα προϊόντα της Microsoft products , μεταξύ άλλων .

✓ **Ενσωματωμένες Βιβλιοθήκες**

Επειδή η PHP σχεδιάστηκε για να χρησιμοποιείται στο Web , έχει πολλές ενσωματωμένες βιβλιοθήκες , που εκτελούν πολλές χρήσιμες λειτουργίες σχετικές με το Web . Μπορείτε να δημιουργήσετε εικόνες GIF δυναμικά , να συνδεθείτε με άλλες υπηρεσίες δικτύων , να στείλετε ηλεκτρονικό ταχυδρομείο , να δουλέψετε με cookies και να δημιουργήσετε PDF έγγραφα : όλα αυτά με λίγες γραμμές κώδικα.

✓ **Κόστος**

Η PHP είναι δωρεάν . Μπορείτε να κατεβάσετε την τελευταία έκδοση από το <http://www.php.net> , χωρίς χρέωση.

✓ **Εκμάθηση της PHP**

Η σύνταξη της PHP βασίζεται σε άλλες γλώσσες προγραμματισμού , βασικά στην C και στην Perl .

✓ **Μεταφερσιμότητα**

Η PHP είναι διαθέσιμη για πολλά λειτουργικά συστήματα . Μπορείτε να γράψετε κώδικα PHP για δωρεάν συστήματα τύπου Unix , όπως LINUX και FreeBSD , για εμπορικές εκδόσεις του UNIX , όπως το Solaris και το IRIX ή για διαφορετικές εκδόσεις των Microsoft Windows.

Ο κώδικας σας συνήθως θα δουλεύει χωρίς αλλαγές στα συστήματα που τρέχουν την PHP.

✓ **Κώδικας προέλευσης**

Έχετε πρόσβαση στον κώδικα προέλευσης της PHP. Αντίθετα με εμπορικά , κλειστά προγράμματα , αν υπάρχει κάτι που θέλετε να αλλάξετε ή να προσθέσετε στη γλώσσα, μπορείτε να το κάνετε.

Δεν χρειάζεται να περιμένετε τον κατασκευαστή να εμφανίσει διορθώσεις. Δεν θα ανησυχείτε αν ο κατασκευαστής θα σταματήσει να υπάρχει ή αν θα σταματήσει να υποστηρίζει το προϊόν.



ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΗΣ MySQL

Μερικοί από τους κύριους ανταγωνιστές της MySQL είναι οι PostgreSQL , Microsoft SQL και Oracle.

Η MySQL έχει πολλά πλεονεκτήματα , όπως χαμηλό κόστος , εύκολη διαμόρφωση και μάθηση και ο κώδικας προέλευσης είναι διαθέσιμος.

✓ Απόδοση

Η MySQL είναι χωρίς αμφιβολία γρήγορη. Μπορείτε να δείτε την σελίδα δοκιμών <http://web.mysql.com/benchmark.html> . Πολλές από αυτές τις δοκιμές δείχνουν ότι η MySQL είναι αρκετά πιο γρήγορη από τον ανταγωνισμό.

✓ Χαμηλό κόστος

Η MySQL είναι διαθέσιμη δωρεάν , με άδεια ανοικτού κώδικα (Open Source) ή με χαμηλό κόστος , αν πάρετε εμπορική άδεια, αν απαιτείται από την εφαρμογή σας.

✓ Ευκολία Χρήσης

Οι περισσότερες μοντέρνες βάσεις δεδομένων χρησιμοποιούν SQL. Αν έχετε χρησιμοποιήσει ένα άλλο σύστημα διαχείρισης βάσεων δεδομένων δεν θα έχετε πρόβλημα να προσαρμοστείτε σε αυτό.

✓ Μεταφερσιμότητα

Η MySQL μπορεί να χρησιμοποιηθεί σε πολλά διαφορετικά συστήματα Unix όπως επίσης και στα Microsoft Windows .

✓ Κώδικας Προέλευσης

Όπως και με την PHP , μπορείτε να πάρετε και να τροποποιήσετε τον κώδικα προέλευσης της MySQL.

✓ Νέα έκδοση

Η νέα έκδοση MySQL 5 έχει έρθει με νέες εντυπωσιακές λειτουργίες. Είναι πλέον ικανή να υποστηρίξει πολύ μεγάλα projects με υψηλή αξιοπιστία.



ΤΙ ΕΙΝΑΙ Ο APACHE WEB SERVER

Ο Apache Web Server είναι ένας πολύ δημοφιλής διακομιστής διαδικτύου που διανέμεται ελεύθερα στο διαδίκτυο. Αναπτύχθηκε και συντηρείται από μια ομάδα εθελοντών που ήθελαν να υλοποιήσουν έναν εύρωστο κώδικα για διακομιστή δικτύου, που να μην είναι εμπορικός αλλά να υποστηρίζει πολλά χαρακτηριστικά.

Ο Apache όπως έχει αποδειχτεί είναι ο πιο γρήγορος, σταθερός, ασφαλής και υποστηρίζει τα περισσότερα χαρακτηριστικά από οποιονδήποτε άλλο διακομιστή δικτύου.

Ο Apache είναι εγκατεστημένος στο 80% των διακομιστών παγκοσμίως (πάνω από 6 εκατομμύρια διακομιστές). Πάνω του είναι εγκατεστημένα εκατομμύρια sites που δέχονται

εκατομμύρια hits καθημερινά χωρίς να παρουσιάζεται κανένα απολύτως πρόβλημα.

Σήμερα ο Apache θεωρείται από τους πιο σταθερούς διακομιστές δικτύου που κυκλοφορούν και θα πρέπει να τονίσουμε ότι αρκετοί εμπορικοί διακομιστές διαδικτύου, όπως ο HTTP Server της IBM, χρησιμοποιούν τον πυρήνα του Apache.

Χαρακτηριστικά του Apache 2

1. **Unix treading:** Υποστήριξη συστημάτων Unix με νήματα POSIX, όπου ο Apache μπορεί να “τρέχει” πολλές διεργασίες ταυτόχρονα.
2. **Υποστήριξη πολλαπλών πρωτοκόλλων:** Υποστηρίζει πιο γρήγορα και πιο σταθερά λειτουργικά όπως BeOS, OS/2 και Windows.
3. **Φίλτρα:** Υποστήριξη φίλτρων που διανέμονται από και προς τους διακομιστές.
4. **Λάθη:** Τα μηνύματα λαθών μπορούν να εμφανίζονται σε διάφορες γλώσσες.
5. **Απλοποιημένη παραμετροποίηση:** Έχουν απλοποιηθεί κάποια directives που ως τώρα ήταν κάπως μπερδεμένα.
6. **Υποστήριξη unicode:** Ο Apache 2 σε Windows NT χρησιμοποιεί μόνο utf-8 κωδικοποίηση.
7. **Κανονικές εκφράσεις:** Υποστήριξη της βιβλιοθήκης PCRE δηλαδή όλες οι κανονικές εκφράσεις που υποστηρίζει η Perl 5.

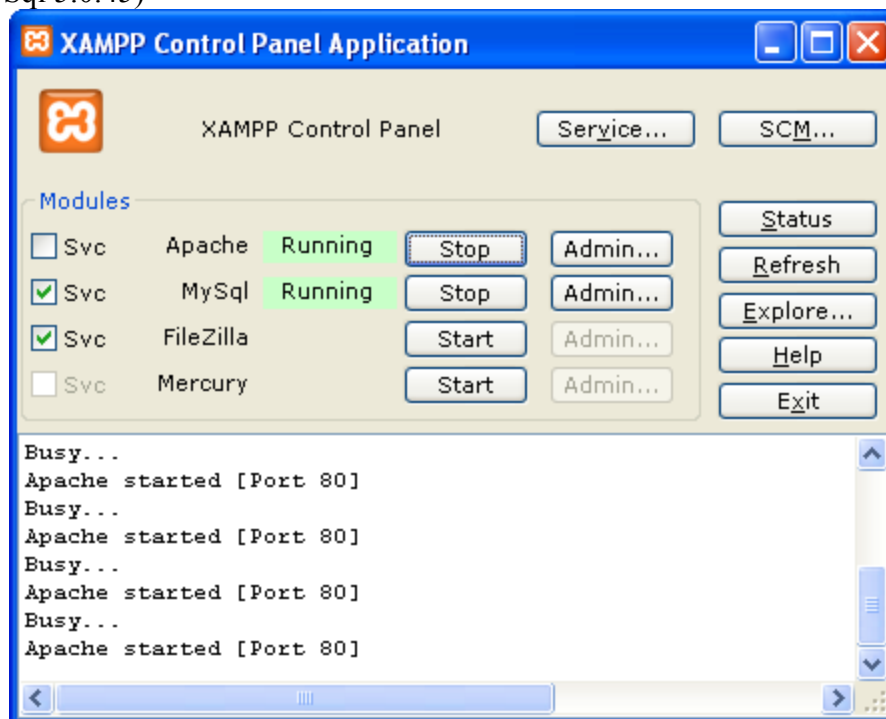


Περιγραφή Των Εργαλείων Που Χρησιμοποιήθηκαν Για Την Ανάπτυξη Της Εφαρμογής:

XAMPP 1.6.3.a

Το xampp είναι ένα ολοκληρωμένο πακέτο Server που περιλαμβάνει apache, php, mysql, perl, filezilla ftp server, mercury email server, υποστήριξη SSL και φυσικά όλα αυτά με αυτοματοποιημένη εγκατάσταση και ρύθμιση. Το μόνο που έχει να κάνει ο Διαχειριστής είναι να ορίσει τα subdomains στο αρχείο conf του apache.

Στην εφαρμογή μας χρησιμοποιήσαμε την έκδοση Xampp 1.6.3.a (PHP 5.2.3, Apache 2.2.4, MySql 5.0.45)



PhpMyAdmin 2.10.3

The screenshot shows the PhpMyAdmin 2.10.3 interface in a browser window. The browser address bar shows 'http://localhost/phpmyadmin/index.php'. The interface title is 'localhost / localhost / logisticsbase | phpMyAdmin 2.10.3 - Gran Paradiso'. The main content area displays a table list for the 'logisticsbase' database. The table list has columns: Πίνακας, Ενέργεια, Εγγραφές, Τύπος, Collation, Μέγεθος, and Επιβάρυνση. The tables listed are: cause_trans, customer, item, item_cat, item_subcat, product_quantity, product_trans, supplier, unit_table, user, and warehouse. A summary row at the bottom indicates 11 tables, 204 records, and 32.6 KB of data.

Πίνακας	Ενέργεια	Εγγραφές	Τύπος	Collation	Μέγεθος	Επιβάρυνση
cause_trans		5	MYISAM	greek_general_ci	2,1 KB	-
customer		6	MYISAM	greek_general_ci	2,8 KB	-
item		39	MYISAM	greek_general_ci	4,7 KB	-
item_cat		4	MYISAM	greek_general_ci	3,1 KB	-
item_subcat		10	MYISAM	greek_general_ci	2,4 KB	-
product_quantity		117	MYISAM	greek_general_ci	5,3 KB	-
product_trans		2	MYISAM	greek_general_ci	2,0 KB	-
supplier		6	MYISAM	greek_general_ci	2,4 KB	-
unit_table		6	MYISAM	greek_general_ci	2,1 KB	-
user		6	MYISAM	greek_general_ci	3,3 KB	-
warehouse		3	MYISAM	greek_general_ci	2,2 KB	-
11 Πίνακες/Πίνακες	Σύνολο	204	MYISAM	greek_general_ci	32,6 KB	0 Bytes

Below the table list, there is a form for creating a new table. The form has a label 'Δημιουργία νέου πίνακα στη βάση logisticsbase', a text input for 'Όνομα:', a text input for 'Number of fields:', and an 'Εκτέλεση' button. There is also a link 'Επιλογή όλων / Απεπιλογή όλων' and a dropdown menu 'Με τους επιλεγμένους:'.

Γενικά

Το PhpMyAdmin είναι ένα εργαλείο γραμμένο σε Php το οποίο διαχειρίζεται την MySQL

στο δίκτυο. Μπορεί να χειρίζεται πλήρως βάσεις δεδομένων, πίνακες, πεδία πινάκων αλλά και ολόκληρο τον MySQL Server. Υποστηρίζει 47 γλώσσες μεταξύ των οποίων και τα Ελληνικά και είναι λογισμικό ανοιχτού κώδικα.

Δυνατότητες του PhpMyAdmin:

Το PhpMyAdmin μπορεί να:

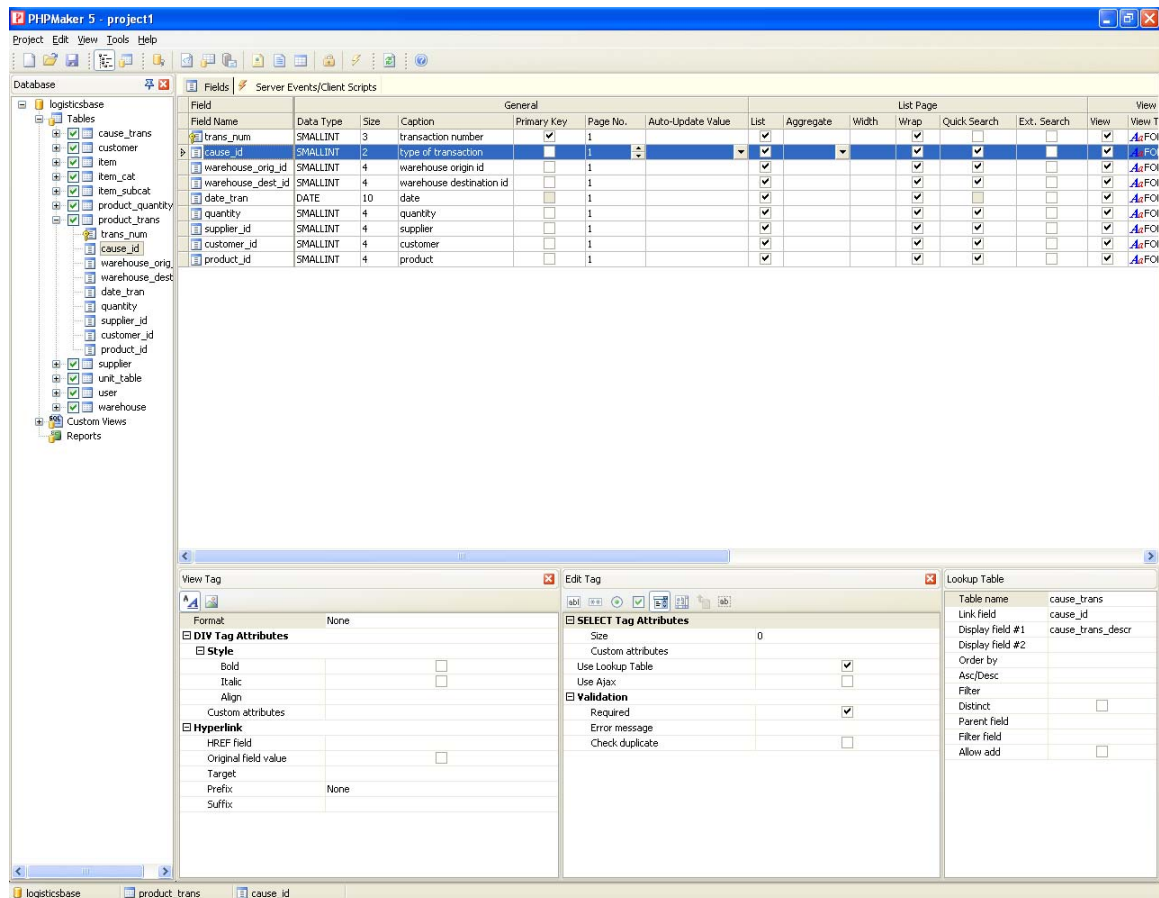
- Δημιουργεί και να διαγράφει βάσεις δεδομένων
- Δημιουργεί, τροποποιεί, διαγράφει, αντιγράφει και μετονομάζει πίνακες
- Κάνει συντήρηση της βάσης
- Προσθέτει, διαγράφει και τροποποιεί πεδία πινάκων
- Εκτελεί Sql ερωτήματα, ακόμα και ομαδικά (batch)
- Διαχειρίζεται κλειδιά σε πεδία
- “Φορτώνει” αρχεία κειμένου σε πίνακες
- Δημιουργεί και διαβάζει πίνακες (που προέρχονται από dump βάσης)
- Εξάγει δεδομένα σε μορφή CVS, Latex, XML
- Διαχειρίζεται πολλούς διακομιστές
- Διαχειρίζεται τους χρήστες MySQL και τα δικαιώματά τους
- Ελέγχει την αναφορική ακεραιότητα των δεδομένων των MyISAM πινάκων
- Δημιουργεί PDF γραφικών του layout της βάσης δεδομένων
- Εκτελεί αναζητήσεις σε όλη τη βάση ή μέρος αυτής
- Υποστηρίζει πίνακες InnoDB και ξένα κλειδιά
- Υποστηρίζει MySQLi, μια βελτιωμένη επέκταση της MySQL



Php Maker 5

Το PhpMaker είναι ένα ισχυρό εργαλείο(για windows) που με αυτοματοποιημένες ως επί το πλείστον λειτουργίες μπορεί να παράγει ολοκληρωμένο κώδικα php σε μικρό χρονικό διάστημα διαβάζοντας μια βάση δεδομένων MySql.

Χρησιμοποιώντας το PhpMaker, μπορούμε να δημιουργήσουμε άμεσα ιστοσελίδες που επιτρέπουν σε χρήστες να εκτελούν λειτουργίες όπως view, edit, search, add, delete ενώ είναι online. Το PhpMaker είναι σχεδιασμένο για να προσφέρει υψηλή ευελιξία, πολυάριθμες επιλογές που με την επιλογή τους μας επιτρέπουν να παράγουμε php applications σύμφωνα με τις ανάγκες μας. Οι κώδικες που παράγονται είναι καθαροί, ευθείς και εύκολοι στη διαχείριση. Χρησιμοποιώντας το PhpMaker μπορούμε να εξοικονομήσουμε πολύ χρόνο αφού δε θα χρειαστεί σε πολλές περιπτώσεις να ανατρέξουμε στη βάση δεδομένων μας για αλλαγές, θα βρούμε πολλές λειτουργίες αυτοματοποιημένες και πολλές φορές ένα απλό κλικ θα μας γλιτώσει από μερικές γραμμές κώδικα που θα έπρεπε να γράψουμε εναλλακτικά.



Macromedia Dreamweaver MX 2004

Ο Dreamweaver MX 2004 αποτελεί το κορυφαίο πρόγραμμα δημιουργίας website και διαδικτυακών εφαρμογών, προσφέρει δυνατότητες δημιουργίας προηγμένων γραφικών, οπτικά εργαλεία περιγράμματος, χαρακτηριστικά ανάπτυξης εφαρμογών και υποστήριξη επεξεργασίας κώδικα μέσω ενός περιβάλλοντος τεχνολογίας ανεξαρτήτου πλατφόρμας. Με το εργαλείο αυτό δημιουργήσαμε τη homepage της σελίδας μας καθώς και ρυθμίσαμε τα μεγέθη των διαφόρων πεδίων ώστε να πετύχουμε ένα πιο ομοιόμορφο οπτικό αποτέλεσμα.



Adobe Photoshop CS2

Το Adobe Photoshop CS2 είναι το επαγγελματικό πρότυπο στην επεξεργασία εικόνας, παρέχοντας περισσότερα από όσα θα μπορούσατε να επιθυμείτε. Με ριζοσπαστικά δημιουργικά εργαλεία που σας βοηθούν να επιτύχετε εξαιρετικά αποτελέσματα, το Photoshop ενσωματώνεται με πρωτοφανή προσαρμοστικότητα στον τρόπο που δουλεύετε. Εμείς χρησιμοποιήσαμε το Photoshop για να δημιουργήσουμε το λογότυπο της εικονικής μας εταιρίας (KA.VA Ltd) καθώς και για να βελτιώσουμε το παρουσιαστικό των τριών στατικών φωτογραφιών καθώς και τις μίας κινούμενης εικόνας που εισάγαμε στην homepage μας.



Περιγραφή της βάσης δεδομένων

Γενικά

Η πτυχιακή εργασία βασίζεται πάνω σε php και MySql. Περιλαμβάνει μια βάση δεδομένων με το όνομα "logisticsbase" η οποία περιέχει πίνακες δεδομένων. Μέσα σε κάθε πίνακα δηλώνουμε τον τύπο κάθε πεδίου (π.χ. Char, int, date) και το πρωτεύον του κλειδί (primary key), το οποίο μπορεί να αποτελείται από ένα πεδίο ή και από συνδυασμό περισσότερων πεδίων με την προϋπόθεση να είναι μοναδικό (unique). Οι πίνακες μέσα στη βάση συνδέονται μεταξύ τους με τα πρωτεύοντα κλειδιά, έτσι ώστε εάν ένας πίνακας αλλάξει δεδομένα, να ενημερώνονται και οι υπόλοιποι πίνακες που είναι συνδεδεμένοι με αυτόν. Επίσης, μέσω των ξένων κλειδιών (foreign keys), τα οποία αποτελούνται από πεδία-ορόσημα για την περάτωση του σκοπού μας, καταφέρνουμε να συνδέσουμε δυναμικά ή στατικά τους πίνακες, για να εμφανίζουμε τα πεδία ή τους πίνακες που επιθυμούμε κατά την περιήγηση μας στο site.

Συγκεκριμένα η βάση μας:

	Πίνακας
<input type="checkbox"/>	bounds_trans
<input type="checkbox"/>	cause_trans
<input type="checkbox"/>	customer
<input type="checkbox"/>	item
<input type="checkbox"/>	item_cat
<input type="checkbox"/>	item_subcat
<input type="checkbox"/>	product_quantity
<input type="checkbox"/>	product_trans
<input type="checkbox"/>	supplier
<input type="checkbox"/>	unit_table
<input type="checkbox"/>	user
<input type="checkbox"/>	warehouse

Και αναλυτικά ο κάθε πίνακας (τα πρωτεύοντα κλειδιά αποτελούνται από τα υπογραμμισμένα πεδία κάθε πίνακα):

- **warehouse**

Ξεκινάμε με τον πίνακα που αφορά τις αποθήκες. Από εδώ καταχωρούμε τα στοιχεία των αποθηκών που διαθέτει η εταιρία μας.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα
<input type="checkbox"/>	<u>warehouse_id</u>	smallint(4)		UNSIGNED	Όχι		auto_increment
<input type="checkbox"/>	<u>warehouse_name</u>	char(30)	greek_general_ci		Όχι		
<input type="checkbox"/>	<u>warehouse_location</u>	char(30)	greek_general_ci		Όχι		
<input type="checkbox"/>	<u>user_id</u>	smallint(4)		UNSIGNED	Όχι		

Αναλυτικά...

	A	B	C	D	E	F
1	Πίνακας	WAREHOUSE				
2	Περιγραφή	ΑΠΟΘΗΚΗ				
3						
4						
5	<i>Όνομα πεδίου</i>	<i>Τύπος</i>	<i>Συνομη περιγραφή</i>	<i>Αναλυτική Περιγραφή</i>	<i>Κλειδί</i>	<i>Αναφέρεται σε</i>
6	<u>warehouse_id</u>	Numeric(4)	ΚΩΔΙΚΟΣ ΑΠΟΘΗΚΗΣ	Ο ΚΩΔΙΚΟΣ ΤΗΣ ΑΠΟΘΗΚΗΣ Ο ΟΠΟΙΟΣ ΕΙΝΑΙ ΞΕΧΩΡΙΣΤΟΣ ΓΙΑ ΚΑΘΕ ΑΠΟΘΗΚΗ.	1	
7	<u>warehouse_name</u>	Char(30)	ΟΝΟΜΑ ΑΠΟΘΗΚΗΣ	ΤΟ ΟΝΟΜΑ ΤΗΣ ΑΠΟΘΗΚΗΣ		
8	<u>warehouse_location</u>	Char(30)	ΤΟΠΟΘΕΣΙΑ	Ο ΧΩΡΟΣ-ΓΕΩΓΡΑΦΙΚΗ ΘΕΣΗ ΣΤΗΝ ΟΠΟΙΑ ΒΡΙΣΚΕΤΑΙ Η ΚΑΘΕ ΑΠΟΘΗΚΗ		
9	<u>user_id</u>	Numeric(4)	ΑΠΟΘΗΚΑΡΙΟΣ	Ο ΑΠΟΘΗΚΑΡΙΟΣ Ο ΟΠΟΙΟΣ ΕΙΝΑΙ ΥΠΕΘΥΝΟΣ ΓΙΑ ΤΗ ΣΥΓΚΕΚΡΙΜΕΝΗ ΑΠΟΘΗΚΗ		ΧΡΗΣΤΕΣ
10	<u>quantity total</u>	Numeric(12,2)	ΠΟΣΟΤΗΤΑ ΕΙΔΟΥΣ	ΣΥΝΟΛΙΚΗ ΠΟΣΟΤΗΤΑ ΕΙΔΟΥΣ ΣΤΗ ΣΥΓΚΕΚΡΙΜΕΝΗ ΑΠΟΘΗΚΗ		

- **item**

Εδώ βλέπουμε τον πίνακα μέσω του οποίου εισάγουμε τα προϊόντα μας (όνομα, περιγραφή, κατηγορία και υποκατηγορία που ανήκουν, καθώς και τις μονάδες στις οποίες μετριέται το καθένα).

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα
<input type="checkbox"/>	<u>product_id</u>	smallint(4)		UNSIGNED	Όχι		auto_increment
<input type="checkbox"/>	<u>product_name</u>	char(30)	greek_general_ci		Όχι		
<input type="checkbox"/>	<u>product_description</u>	varchar(25)	greek_general_ci		Όχι		
<input type="checkbox"/>	<u>class_id</u>	smallint(2)		UNSIGNED	Όχι		
<input type="checkbox"/>	<u>subclass_id</u>	smallint(2)		UNSIGNED	Όχι		
<input type="checkbox"/>	<u>unit</u>	varchar(2)	greek_general_ci		Όχι		

Αναλυτικά...

	A	B	C	D	E	F
1	Πίνακας	ITEMS				
2	Περιγραφή	ΕΙΔΗ				
3						
4						
5	Όνομα πεδίου	Τύπος	Συνομη περιγραφή	Αναλυτική Περιγραφή	Κλειδί	Αναφέρεται σε
6	<u>product_id</u>	Numeric(4)	ΚΩΔΙΚΟΣ ΕΙΔΟΥΣ	ΚΩΔΙΚΟΣ ΕΙΔΟΥΣ Ο ΟΠΟΙΟΣ ΕΙΝΑΙ ΞΕΧΩΡΙΣΤΟΣ ΓΙΑ ΚΑΘΕ ΚΑΤΗΓΟΡΙΑ	1	
7	<u>product_name</u>	Char(30)	ΟΝΟΜΑ ΕΙΔΟΥΣ	ΓΕΝΙΚΗ ΟΝΟΜΑΣΙΑ ΤΟΥ ΕΙΔΟΥΣ-ΚΑΤΗΓΟΡΙΑΣ		
8	<u>product_description</u>	Varchar2(50)	ΠΕΡΙΓΡΑΦΗ ΕΙΔΟΥΣ	ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΣΥΓΚΕΚΡΙΜΕΝΟΥ ΕΙΔΟΥΣ-ΚΑΤΗΓΟΡΙΑΣ		
9	<u>class_id</u>	Numeric(2)	ΚΩΔΙΚΟΣ ΚΑΤΗΓΟΡΙΑΣ	ΚΩΔΙΚΟΣ ΚΑΤΗΓΟΡΙΑΣ Ο ΟΠΟΙΟΣ ΕΙΝΑΙ ΞΕΧΩΡΙΣΤΟΣ ΓΙΑ ΚΑΘΕ ΜΙΑ		ΚΑΤΗΓΟΡΙΕΣ ΕΙΔΩΝ
10	<u>subclass_id</u>	Numeric(2)	ΚΩΔΙΚΟΣ ΥΠΟΚΑΤΗΓΟΡΙΑΣ	ΚΩΔΙΚΟΣ ΥΠΟΚΑΤΗΓΟΡΙΑΣ Ο ΟΠΟΙΟΣ ΕΙΝΑΙ ΞΕΧΩΡΙΣΤΟΣ ΓΙΑ ΚΑΘΕ ΜΙΑ		class_id, subclass_id αναφέρεται σε ΥΠΟΚΑΤΗΓΟΡΙΕΣ ΕΙΔΩΝ
11	<u>unit</u>			Μονάδα μέτρησης		ΠΙΝΑΚΑ ΜΟΝΑΔΩΝ ΜΕΤΡΗΣΗΣ

- **item_cat**

Ένας λιτός πίνακας για να ομαδοποιούμε τις κατηγορίες των προϊόντων μας και όπως θα δούμε παρακάτω, απαραίτητος για τη σωστή προβολή τους.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα
<input type="checkbox"/>	<u>class_id</u>	smallint(2)		UNSIGNED	Όχι		auto_increment
<input type="checkbox"/>	<u>class_name</u>	char(30)	greek_general_ci		Όχι		
<input type="checkbox"/>	<u>class_description</u>	varchar(25)	greek_general_ci		Όχι		

Αναλυτικά...

	A	B	C	D	E	F
1	Πίνακας	ITEM_CAT				
2	Περιγραφή	ΚΑΤΗΓΟΡΙΕΣ ΕΙΔΩΝ				
3						
4	<i>Όνομα πεδίου</i>	<i>Τύπος</i>	<i>Συνομη περιγραφή</i>	<i>Αναλυτική Περιγραφή</i>	<i>Κλειδί</i>	<i>Αναφέρεται σε</i>
5	<u>class_id</u>	Numeric(2)	ΚΩΔΙΚΟΣ ΚΑΤΗΓΟΡΙΑΣ	ΚΩΔΙΚΟΣ ΚΑΤΗΓΟΡΙΑΣ Ο ΟΠΟΙΟΣ ΕΙΝΑΙ ΕΞΧΩΡΙΣΤΟΣ ΓΙΑ ΚΑΘΕ ΜΙΑ	1	
6	<u>class_name</u>	char(30)	ΟΝΟΜΑ ΚΑΤΗΓΟΡΙΑΣ	ΓΕΝΙΚΗ ΟΝΟΜΑΣΙΑ ΤΟΥ ΕΙΔΟΥΣ-ΚΑΤΗΓΟΡΙΑΣ		
7	<u>class_description</u>	Varchar2(50)	ΠΕΡΙΓΡΑΦΗ ΚΑΤΗΓΟΡΙΑΣ	ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΣΥΓΚΕΚΡΙΜΕΝΟΥ ΕΙΔΟΥΣ-ΚΑΤΗΓΟΡΙΑΣ		

- **item_subcat**

Λειτουργεί ακριβώς όπως ο item_cat αλλά ο συγκεκριμένος αφορά στις υποκατηγορίες.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα
<input type="checkbox"/>	<u>subclass_id</u>	smallint(2)		UNSIGNED	Όχι		auto_increment
<input type="checkbox"/>	<u>class_id</u>	smallint(2)		UNSIGNED	Όχι		
<input type="checkbox"/>	<u>subclass_name</u>	char(30)	greek_general_ci		Όχι		
<input type="checkbox"/>	<u>subclass_descr</u>	varchar(25)	greek_general_ci		Όχι		

Αναλυτικά...

	A	B	C	D	E	F
1	Πίνακας	SUBCAT_ITEMS				
2	Περιγραφή	ΥΠΟΚΑΤΗΓΟΡΙΕΣ ΕΙΔΩΝ				
3						
4						
5	<i>Όνομα πεδίου</i>	<i>Τύπος</i>	<i>Συνομη περιγραφή</i>	<i>Αναλυτική Περιγραφή</i>	<i>Κλειδί</i>	<i>Αναφέρεται σε</i>
6	<i>class_id</i>	Numeric(2)	ΚΩΔΙΚΟΣ ΚΑΤΗΓΟΡΙΑΣ	ΚΩΔΙΚΟΣ ΚΑΤΗΓΟΡΙΑΣ	1	ΚΑΤΗΓΟΡΙΕΣ ΕΙΔΩΝ
7	<i>subclass_id</i>	Numeric(2)	ΚΩΔΙΚΟΣ ΥΠΟΚΑΤΗΓΟΡΙΑΣ	ΚΩΔΙΚΟΣ ΥΠΟΚΑΤΗΓΟΡΙΑΣ Ο ΟΠΟΙΟΣ ΕΙΝΑΙ ΜΟΝΑΔΙΚΟΣ ΜΕΣΑ ΣΕ ΚΑΘΕ ΚΑΤΗΓΟΡΙΑ	2	
8	<i>subclass_name</i>	Char(30)	ΚΩΔΙΚΟΣ ΕΙΔΟΥΣ	ΟΝΟΜΑΣΙΑ ΤΗΣ ΥΠΟΚΑΤΗΓΟΡΙΑΣ		
9	<i>subclass_description</i>	Varchar(50)	ΟΝΟΜΑ ΕΙΔΟΥΣ	ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΣΥΓΚΕΚΡΙΜΕΝΗΣ ΥΠΟΚΑΤΗΓΟΡΙΑΣ ΚΑΤΗΓΟΡΙΑΣ		

- **unit_table**

Ο πιο απλός πίνακας της βάσης μας, ο οποίος απλά ονομάζει τους τύπους των μονάδων που αφορούν τα προϊόντα μας.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα
<input type="checkbox"/>	<u>unit_id</u>	varchar(25)	greek_general_ci		Όχι		
<input type="checkbox"/>	<u>unit_description</u>	varchar(25)	greek_general_ci		Όχι		

Αναλυτικά...

	A	B	C	D	E
1	Πίνακας	UNIT_ID			
2	Περιγραφή	μονάδες μετρησης προϊόντων			
3				Κλειδί	Αναφέρεται σε
4	unit_id	varchar2(5)	ΚΩΔΙΚΟΣ ΜΟΝΑΔΑΣ ΜΕΤΡΗΣΗΣ	1	
5	unit_dscr	varchar2(40)	ΠΕΡΙΓΡΑΦΗ ΜΟΝΑΔΑΣ ΜΕΤΡΗΣΗΣ		

- **product_quantity**

Ένας από τους πιο σημαντικούς πίνακες καθώς δίνει τη δυνατότητα να ελέγχουμε διαθέσιμες ποσότητες προϊόντος ανά αποθήκη, να δεσμεύουμε ποσότητα, αλλά και να υπολογίζεται αυτόματα η νέα συνολική-διαθέσιμη-δεσμευμένη ποσότητα μετά από κάθε κίνηση.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα
<input type="checkbox"/>	<u>product_id</u>	smallint(4)		UNSIGNED	Όχι		
<input type="checkbox"/>	<u>warehouse_id</u>	smallint(4)		UNSIGNED	Όχι		
<input type="checkbox"/>	<u>quantity_total</u>	smallint(12)		UNSIGNED	Όχι		
<input type="checkbox"/>	<u>quantity_bounded</u>	smallint(12)		UNSIGNED	Όχι		
<input type="checkbox"/>	<u>quantity_available</u>	smallint(12)		UNSIGNED	Όχι		

Αναλυτικά...

	A	B	C	D	E	F
1	Πίνακας	PRODUCT_QUANTITY				
2	Περιγραφή	ΑΠΟΘΕΜΑ				
3						
4						
5	Όνομα πεδίου	Τύπος	Συνομη περιγραφή	Αναλυτική Περιγραφή	Κλειδί	Αναφέρεται σε
6	product_id	Numeric(4)	ΚΩΔΙΚΟΣ ΕΙΔΟΥΣ	ΚΩΔΙΚΟΣ ΕΙΔΟΥΣ Ο ΟΠΟΙΟΣ ΕΙΝΑΙ ΜΟΝΑΔΙΚΟΣ ΓΙΑ ΚΑΘΕ ΠΡΟΪΟΝ	1	ΕΙΔΗ
7	warehouse_id	Numeric(4)	ΤΟΠΟΣ ΑΠΟΘΗΚΕΥΣΗΣ	ΑΠΟΘΗΚΗ ΣΤΗΝ ΟΠΟΙΑ ΒΡΙΣΚΕΤΑΙ Η ΠΟΣΟΤΗΤΑ ΤΟΥ ΠΡΟΪΟΝΤΟΣ ΣΤΗΝ ΟΠΟΙΑ ΑΝΑΦΕΡΟΜΑΣΤΕ	2	ΑΠΟΘΗΚΕΣ
8	quantity_total	Numeric(12,2)	ΠΟΣΟΤΗΤΑ ΕΙΔΟΥΣ	ΣΥΝΟΛΙΚΗ ΠΟΣΟΤΗΤΑ ΕΙΔΟΥΣ ΣΤΗ ΣΥΓΚΕΚΡΙΜΕΝΗ ΑΠΟΘΗΚΗ		
9	quantity_bounded	Numeric(12,2)	ΔΕΣΜΕΥΜΕΝΗ ΠΟΣΟΤΗΤΑ ΕΙΔΟΥΣ	ΔΕΣΜΕΥΜΕΝΗ ΠΟΣΟΤΗΤΑ ΕΙΔΟΥΣ ΣΤΗ ΣΥΓΚΕΚΡΙΜΕΝΗ ΑΠΟΘΗΚΗ		
10	quantity_available	Numeric(12,2)	ΔΙΑΘΕΣΙΜΗ ΠΟΣΟΤΗΤΑ	ΔΙΑΘΕΣΙΜΗ ΠΟΣΟΤΗΤΑ ΕΙΔΟΥΣ ΣΤΗ ΣΥΓΚΕΚΡΙΜΕΝΗ ΑΠΟΘΗΚΗ		

- **user**

Ο user είναι ένας πίνακας με πολύ ενδιαφέρον καθώς με τη βοήθεια αυτού δημιουργούμε τη σελίδα στην οποία γίνεται το login καθώς και η ανάθεση βαθμού στην εταιρία μας, μέσω της οποίας ορίζονται οι εξουσιοδοτήσεις. Επίσης μέσω αυτού του πίνακα γίνεται και το register.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα
<input type="checkbox"/>	<u>user_id</u>	smallint(4)		UNSIGNED	Όχι		auto_increment
<input type="checkbox"/>	<u>user_ident</u>	char(10)	greek_general_ci		Όχι		
<input type="checkbox"/>	<u>user_pass</u>	char(10)	greek_general_ci		Όχι		
<input type="checkbox"/>	<u>user_full_name</u>	char(30)	greek_general_ci		Όχι		
<input type="checkbox"/>	<u>attribute</u>	int(11)			Όχι		
<input type="checkbox"/>	<u>status</u>	enum('1', '0')	greek_general_ci		Όχι	1	

Αναλυτικά...

	A	B	C	D	E	F
1	Πίνακας	USER				
2	Περιγραφή	ΧΡΗΣΤΗΣ				
3						
4	<i>Όνομα πεδίου</i>	<i>Τύπος</i>	<i>Συνομη περιγραφή</i>	<i>Αναλυτική Περιγραφή</i>	<i>Κλειδί</i>	<i>Αναφέρεται σε</i>
5	<u>user_id</u>	Numeric(4)	ΚΩΔΙΚΟΣ ΧΡΗΣΤΗ	ΚΩΔΙΚΟΣ ΧΡΗΣΤΗ	1	
6	<u>user_ident</u>	Char(10)	USERNAME ΧΡΗΣΤΗ	ΤΟ USERNAME ΠΟΥ ΘΑ ΧΡΗΣΙΜΟΠΟΙΕΙ Ο ΧΡΗΣΤΗΣ ΓΙΑ ΝΑ ΚΑΝΕΙ LOGIN		
7	<u>user_pass</u>	Char(10)	PASSWORD ΧΡΗΣΤΗ	ΤΟ PASSWORD ΠΟΥ ΘΑ ΧΡΗΣΙΜΟΠΟΙΕΙ Ο ΧΡΗΣΤΗΣ ΓΙΑ ΝΑ ΚΑΝΕΙ LOGIN		
8	<u>user_full_name</u>	Char(30)	ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΧΡΗΣΤΗ	ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΧΡΗΣΤΗ		
9	<u>attribute</u>	Boolean(1)	ΙΔΙΟΤΗΤΑ	ΙΔΙΟΤΗΤΑ ΧΡΗΣΤΗ(ΑΝ ΕΙΝΑΙ ΔΙΟΙΚΗΤΙΚΟ ΣΤΕΛΕΧΟΣ Ή ΑΠΟΘΗΚΑΡΙΟΣ)		
10	<u>status</u>			ενεργός ή όχι		

- **customer**

Ο πίνακας αυτός εξυπηρετεί στην καταχώρηση πελατών της εταιρίας, εισάγοντας τα προσωπικά τους στοιχεία καθώς και το αν είναι ενεργοί ή όχι.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα
<input type="checkbox"/>	company	char(20)	greek_general_ci		Όχι		
<input type="checkbox"/>	<u>customer_id</u>	smallint(4)		UNSIGNED	Όχι		auto_increment
<input type="checkbox"/>	<u>customer_surname</u>	char(30)	greek_general_ci		Όχι		
<input type="checkbox"/>	<u>customer_name</u>	char(30)	greek_general_ci		Όχι		
<input type="checkbox"/>	<u>customer_address</u>	char(30)	greek_general_ci		Όχι		
<input type="checkbox"/>	location	char(20)	greek_general_ci		Όχι		
<input type="checkbox"/>	<u>customer_telephone</u>	char(10)	greek_general_ci		Όχι		
<input type="checkbox"/>	<u>status</u>	enum('1', '0')	greek_general_ci		Όχι		

Αναλυτικά...

	A	B	C	D	E	F
1	Πίνακας	CUSTOMER				
2	Περιγραφή	ΠΕΛΑΤΗΣ				
3						
4						
5	Όνομα πεδίου	Τύπος	Συνομη περιγραφή	Αναλυτική Περιγραφή	Κλειδί	Αναφέρεται σε
6	customer_id	Numeric(4)	ΚΩΔΙΚΟΣ ΠΕΛΑΤΗ	ΚΩΔΙΚΟΣ ΠΕΛΑΤΗ ΕΤΑΙΡΕΙΑΣ	1	
7	customer_full_name	Char(30)	ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΠΕΛΑΤΗ	ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΠΕΛΑΤΗ		
8	customer_name	Char(30)	ΟΝΟΜΑΤΕΠΩΝΥΜΟ / ΕΠΩΝΥΜΙΑ ΠΕΛΑΤΗ	ΟΝΟΜΑΤΕΠΩΝΥΜΟ / ΕΠΩΝΥΜΙΑ ΠΕΛΑΤΗ		
9	customer_address	Char(30)	ΔΙΕΥΘΥΝΣΗ	ΔΙΕΥΘΥΝΣΗ ΠΕΛΑΤΗ		
10	customer_telephone	Char(10)	ΣΤΑΘ. ΤΗΛΕΦΩΝΟ	ΣΤΑΘΕΡΟ ΤΗΛΕΦΩΝΟ ΕΠΙΧΕΙΡΗΣΗΣ ΠΕΛΑΤΗ		
11	status	Boolean(1)	STATUS	ΚΑΤΑΣΤΑΣΗ ΠΕΛΑΤΗ,ΑΝ ΕΙΝΑΙ ΕΝΕΡΓΟΣ Ή ΟΧΙ		

- **supplier**

Η λειτουργία του είναι ίδια ακριβώς με εκείνη του customer. Αυτός ο πίνακας όμως αφορά τους προμηθευτές.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα
<input type="checkbox"/>	company	char(20)	greek_general_ci		Όχι		
<input type="checkbox"/>	supplier_id	smallint(4)		UNSIGNED	Όχι		auto_increment
<input type="checkbox"/>	supplier_surname	varchar(20)	greek_general_ci		Όχι		
<input type="checkbox"/>	supplier_name	char(30)	greek_general_ci		Όχι		
<input type="checkbox"/>	supplier_address	char(30)	greek_general_ci		Όχι		
<input type="checkbox"/>	location	char(20)	greek_general_ci		Όχι		
<input type="checkbox"/>	supplier_telephone	char(10)	greek_general_ci		Όχι		
<input type="checkbox"/>	status	enum('1', '0')	greek_general_ci		Όχι	1	

Αναλυτικά...

	A	B	C	D	E	F
1	Πίνακας	SUPPLIER				
2	Περιγραφή	ΠΡΟΜΗΘΕΥΤΗΣ				
3						
4						
5	Όνομα πεδίου	Τύπος	Συνοπτη περιγραφή	Αναλυτική Περιγραφή	Κλειδί	Αναφέρεται σε
6	supplier_id	Numeric(4)	ΚΩΔΙΚΟΣ ΠΡΟΜΗΘΕΥΤΗ	ΚΩΔΙΚΟΣ ΠΡΟΜΗΘΕΥΤΗ ΕΤΑΙΡΕΙΑΣ	1	
7	supplier name	Char(30)	ΟΝΟΜΑΤΕΠΩΝΥΜΟ / ΕΠΩΝΥΜΙΑ ΠΡΟΜΗΘΕΥΤΗ	ΟΝΟΜΑΤΕΠΩΝΥΜΟ / ΕΠΩΝΥΜΙΑ ΠΡΟΜΗΘΕΥΤΗ		
8	supplier_address	Char(30)	ΔΙΕΥΘΥΝΣΗ	ΔΙΕΥΘΥΝΣΗ ΠΡΟΜΗΘΕΥΤΗ		
9	supplier telephone	Char(10)	ΣΤΑΘ. ΤΗΛΕΦΩΝΟ	ΣΤΑΘΕΡΟ ΤΗΛΕΦΩΝΟ ΕΠΙΧΕΙΡΗΣΗΣ ΠΡΟΜΗΘΕΥΤΗ		
10	status	Boolean(1)	STATUS	ΚΑΤΑΣΤΑΣΗ ΠΡΟΜΗΘΕΥΤΗ,ΑΝ ΕΙΝΑΙ ΕΝΕΡΓΟΣ Ή ΟΧΙ		

- **cause_trans**

Μέσω του πίνακα αυτού μπορούμε αλλάζοντας τους συνδυασμούς των τεσσάρων boolean πεδίων να εμφανίζουμε μόνο τα πεδία που θα χρειαστούμε ανάλογα με την εκάστοτε κίνηση κατά τη διάρκεια καταχώρησης μιας νέας κίνησης στον πίνακα product_trans που θα δούμε παρακάτω.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα
<input type="checkbox"/>	cause_id	smallint(2)		UNSIGNED	Όχι		
<input type="checkbox"/>	check_exist_orig	enum('1', '0')	greek_general_ci		Όχι		
<input type="checkbox"/>	profit_loss_orig	enum('1', '0')	greek_general_ci		Όχι		
<input type="checkbox"/>	check_exist_dest	enum('1', '0')	greek_general_ci		Όχι		
<input type="checkbox"/>	profit_loss_dest	enum('1', '0')	greek_general_ci		Όχι		
<input type="checkbox"/>	cause_trans_descr	varchar(40)	geostd8_general_ci		Όχι		

Αναλυτικά...

	A	B	C	D	E	F
1	Πίνακας	CAUSE_TRANS	Πίνακας	CAUSE_TRANS		
2	Περιγραφή	ΑΙΤΙΕΣ ΚΙΝΗΣΗΣ	Περιγραφή	ΑΙΤΙΕΣ ΚΙΝΗΣΕΙΣ		
3						
4						
5	Όνομα πεδίου	Τύπος	Συνομη περιγραφή	Αναλυτική Περιγραφή	Κλειδί	Αναφέρεται σε
6	cause_id	Numeric(2)	ΚΩΔΙΚΟΣ ΚΙΝΗΣΗΣ	ΤΟ ΕΙΔΟΣ ΤΗΝ ΚΙΝΗΣΗΣ ΑΝ ΕΙΝΑΙ ΑΓΟΡΑ, ΠΩΛΗΣΗ, ΕΠΙΣΤΡΟΦΗ Ή ΕΣΩΤΕΡΙΚΗ ΜΕΤΑΚΙΝΗΣΗ		1 ΚΙΝΗΣΕΙΣ
7	check_exist_orig	Boolean(1)	ΕΛΕΓΧΟΣ ΣΥΜΜΕΤΟΧΗΣ ΤΗΣ ΑΠΟΘΗΚΗΣ ΠΡΟΕΛΕΥΣΗΣ ΣΤΗ ΚΙΝΗΣΗ	ΑΝ ΤΟ ΑΠΟΤΕΛΕΣΜΑ ΕΙΝΑΙ ΝΑΙ (ΔΗΛΑΔΗ 1) ΤΟΤΕ ΕΧΟΥΜΕ ΑΠΟΘΗΚΗ ΠΡΟΕΛΕΥΣΗΣ, ΑΝ ΟΧΙ ΔΕΝ ΕΧΟΥΜΕ		
8	profit_loss_orig	Boolean(1)	ΚΕΡΔΟΣ-ΖΗΜΙΑ	ΑΥΞΗΣΗ Ή ΜΕΙΩΣΗ ΠΡΟΪΟΝΤΟΣ ΓΙΑ ΤΗΝ ΑΠΟΘΗΚΗ ΠΡΟΕΛΕΥΣΗΣ		
9	check_exist_dest	Boolean(1)	ΕΛΕΓΧΟΣ ΣΥΜΜΕΤΟΧΗΣ ΤΗΣ ΑΠΟΘΗΚΗΣ ΠΡΟΟΡΙΣΜΟΥ ΣΤΗ ΚΙΝΗΣΗ	ΑΝ ΤΟ ΑΠΟΤΕΛΕΣΜΑ ΕΙΝΑΙ ΝΑΙ (ΔΗΛΑΔΗ 1) ΤΟΤΕ ΕΧΟΥΜΕ ΑΠΟΘΗΚΗ ΠΡΟΟΡΙΣΜΟΥ, ΑΝ ΟΧΙ ΔΕΝ ΕΧΟΥΜΕ		
10	profit_loss_dest	Boolean(1)	ΚΕΡΔΟΣ-ΖΗΜΙΑ	ΑΥΞΗΣΗ Ή ΜΕΙΩΣΗ ΠΡΟΪΟΝΤΟΣ ΓΙΑ ΤΗΝ ΑΠΟΘΗΚΗ ΠΡΟΟΡΙΣΜΟΥ		
11		Varchar(40)	περιγραφή αιτίας κίνησης			

οι booleans του οποίου λειτουργούν ως εξής:

cd	dscr	fig_orgn	fig_orgn_add	fig_dstn	fig_dstn_add	
ΚΩΔΙΚΟΣ	ΠΕΡΙΓΡΑΦΗ	ΑΠΟΘΗΚΗ ΠΡΟΕΛΕΥΣΗΣ	ΣΥΝ / ΠΛΗΝ ΑΠΟΘΕΜΑ	ΑΠΟΘΗΚΗ ΠΡΟΟΡΙΣΜΟ Υ	ΣΥΝ / ΠΛΗΝ ΑΠΟΘΕΜΑ	
1	ΑΓΟΡΑ από προμηθ	ΟΧΙ	<null>	ΝΑΙ	ΣΥΝ	αυξάνει το απόθεμα της αποθήκης προορισμού
2	ΠΩΛΗΣΗ σε πελάτη	ΝΑΙ	ΠΛΗΝ	ΟΧΙ	<null>	μειώνει το απόθεμα της αποθήκης προέλευσης
3	ΜΕΤΑΦΟΡΑ εσωτερική	ΝΑΙ	ΠΛΗΝ	ΝΑΙ	ΣΥΝ	αφαιρούμε από την προέλευση και προσθέτουμε στον προορισμό
4	ΕΠΙΣΤΡΟΦΗ σε προμηθευτή	ΟΧΙ	<null>	ΝΑΙ	ΠΛΗΝ	αντίθετη της αγοράς: μειώνει το απόθεμα προορισμού
5	ΕΠΙΣΤΡΟΦΗ από πελάτη	ΝΑΙ	ΣΥΝ	ΟΧΙ	<null>	αντίθετη της πώλησης: αυξάνει το απόθεμα προέλευσης
6	ΑΠΟΓΡΑΦΗ	ΟΧΙ	<null>	ΝΑΙ	ΣΥΝ	αυξάνει το απόθεμα της αποθήκης προορισμού
7	ΚΑΤΑΣΤΡΟΦΗ	ΝΑΙ	ΠΛΗΝ	ΟΧΙ	<null>	μειώνει το απόθεμα της αποθήκης προέλευσης

Άρα οι Booleans θα μπουν ως εξής σύμφωνα με τον παραπάνω πίνακα:

ΑΓΟΡΑ ΑΠΟ ΠΡΟΜΗΘΕΥΤΗ → 0, 0, 1, 1

ΠΩΛΗΣΗ ΣΕ ΠΕΛΑΤΗ → 1, 1, 0, 0

ΜΕΤΑΦΟΡΑ ΕΣΩΤΕΡΙΚΗ → 1, 1, 1, 1

ΕΠΙΣΤΡΟΦΗ ΣΕ ΠΡΟΜΗΘΕΥΤΗ → 0, 0, 1, 1

ΕΠΙΣΤΡΟΦΗ ΣΕ ΠΕΛΑΤΗ → 1, 1, 0, 0

- **product_trans**

Ο πιο πολύπλοκος αλλά και χρήσιμος πίνακας της βάσης μας. Μέσω αυτού επιλέγουμε την κίνηση που επιθυμούμε να κάνουμε (σε συνεργασία πάντα με τον cause_trans), αλλά και τις αποθήκες, τους προμηθευτές, αλλά και τους πελάτες που θα αφορά η κίνηση αυτή.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα
<input type="checkbox"/>	trans_num	smallint(3)		UNSIGNED	Όχι		auto_increment
<input type="checkbox"/>	cause_id	smallint(2)			Όχι		
<input type="checkbox"/>	warehouse_orig_id	smallint(4)		UNSIGNED	Ναι	NULL	
<input type="checkbox"/>	warehouse_dest_id	smallint(4)		UNSIGNED	Ναι	NULL	
<input type="checkbox"/>	date_tran	date			Όχι		
<input type="checkbox"/>	quantity	smallint(4)		UNSIGNED	Όχι		
<input type="checkbox"/>	supplier_id	smallint(4)			Ναι	NULL	
<input type="checkbox"/>	customer_id	smallint(4)			Ναι	NULL	
<input type="checkbox"/>	product_id	smallint(4)		UNSIGNED	Όχι		

Αναλυτικά...

	A	B	C	D	E	F
1	Πίνακας	PRODUCT_TRANS				
2	Περιγραφή	ΚΙΝΗΣΕΙΣ				
3						
4						
5	<i>Όνομα πεδίου</i>	<i>Τύπος</i>	<i>Συνομη περιγραφή</i>	<i>Αναλυτική Περιγραφή</i>	<i>Κλειδί</i>	<i>Αναφέρεται σε</i>
6	cause_id	Numeric(2)	ΑΙΤΙΑ ΚΙΝΗΣΗΣ	ΤΟ ΕΙΔΟΣ ΤΗΝ ΚΙΝΗΣΗΣ_ΑΝ ΕΙΝΑΙ	1	
7	warehouse_orig_id	Numeric(4)	ΑΠΟΘΗΚΗ ΠΡΟΕΛΕΥΣΗΣ	Ο κωδικός αποθήκης από την οποία κινήθηκε το είδος.		ΑΠΟΘΗΚΗ
8	warehouse_dest_id	Numeric(4)	ΑΠΟΘΗΚΗ ΠΡΟΟΡΙΣΜΟΥ	Ο κωδικός αποθήκης στην οποία κινήθηκε το είδος.		ΑΠΟΘΗΚΗ
9	date_tran	Date(10)	ΗΜΕΡΟΜΗΝΙΑ ΚΙΝΗΣΗΣ	ΗΜΕΡΟΜΗΝΙΑ ΠΡΑΓΜΑΤΟΠΟΙΗΣΗΣ ΚΙΝΗΣΗΣ		
10	quantity	Numeric(4)	ΠΟΣΟΤΗΤΑ	ΠΟΣΟΤΗΤΑ κίνησης είδους (στην μονάδα μέτρησης του είδους)		
11	supplier_id	Numeric(4)	ΚΩΔΙΚΟΣ ΠΡΟΜΗΘΕΥΤΗ	ΚΩΔΙΚΟΣ ΠΡΟΜΗΘΕΥΤΗ ΕΤΑΙΡΕΙΑΣ, σε περίπτωση που η κίνηση έχει να κάνει με προμηθευτή. Σε αντίθετη περίπτωση η τιμή του πεδίου θα είναι NULL.		ΠΡΟΜΗΘΕΥΤΗΣ
12	customer_id	Numeric(4)	ΚΩΔΙΚΟΣ ΠΕΛΑΤΗ	ΚΩΔΙΚΟΣ ΠΕΛΑΤΗ, σε περίπτωση που η κίνηση έχει να κάνει με πελάτη, σε αντίθετη περίπτωση null		ΠΕΛΑΤΗΣ
13	product_id	Numeric(4)	ΚΩΔΙΚΟΣ ΕΙΔΟΥΣ	Κωδικός είδους που κινήθηκε.		ΕΙΔΗ

- **bounds_trans**

Τέλος, έχουμε και τον πίνακα δεσμεύσεων. Ουσιαστικά πρόκειται για ένα πίνακα παραγγελιών, μέσω του οποίου η εφαρμογή θα κρατάει τις παραγγελίες των πελατών που εκκρεμούν.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα
<input type="checkbox"/>	warehouse_id	smallint(4)		UNSIGNED	Όχι		
<input type="checkbox"/>	product_id	smallint(4)		UNSIGNED	Όχι		
<input type="checkbox"/>	customer_id	smallint(4)		UNSIGNED	Όχι		
<input type="checkbox"/>	quantity_bounded	smallint(12)		UNSIGNED	Όχι		

Αναλυτικά...

	A	B	C	D	E	F
1	Πίνακας	BOUNDS_TRANS				
2	Περιγραφή	Δεσμεύσεις				
3						
4						
5	Όνομα πεδίου	Τύπος	Συνομη περιγραφή	Αναλυτική Περιγραφή	Κλειδί	Αναφέρεται σε
6	warehouse_id	Numeric(4)	ΚΩΔΙΚΟΣ ΑΠΟΘΗΚΗΣ	ΚΩΔΙΚΟΣ ΑΠΟΘΗΚΗΣ Ο ΟΠΟΙΟΣ ΕΙΝΑΙ ΜΟΝΑΔΙΚΟΣ ΓΙΑ ΚΑΘΕ ΑΠΟΘΗΚΗ	1	WAREHOUSE
7	product_id	Numeric(4)	ΚΩΔΙΚΟΣ ΠΡΟΪΟΝΤΟΣ	ΚΩΔΙΚΟΣ ΠΡΟΪΟΝΤΟΣ Ο ΟΠΟΙΟΣ ΕΙΝΑΙ ΜΟΝΑΔΙΚΟΣ ΓΙΑ ΚΑΘΕ ΠΡΟΪΟΝ		ITEM
8	customer_id	Numeric(4)	ΚΩΔΙΚΟΣ ΠΕΛΑΤΗ	ΚΩΔΙΚΟΣ ΑΠΟΘΗΚΗΣ Ο ΟΠΟΙΟΣ ΕΙΝΑΙ ΜΟΝΑΔΙΚΟΣ ΓΙΑ ΚΑΘΕ ΠΕΛΑΤΗ		CUSTOMER
9	quantity_bounded	Numeric(12)	ΔΕΣΜΕΥΜΕΝΗ ΠΟΣΟΤΗΤΑ	ΠΟΣΟΤΗΤΑ ΤΟΥ ΠΡΟΪΟΝΤΟΣ ΠΟΥ ΕΧΕΙ ΔΕΣΜΕΥΤΕΙ ΑΠΟ ΕΝΑΝ Ή ΠΕΡΙΣΣΟΤΕΡΟΥΣ ΠΕΛΑΤΕΣ		

Manual Εφαρμογής

Ήρθε η ώρα λοιπόν, αφού είδαμε τους πίνακες μας, να δούμε το πώς υλοποιούνται στην πράξη. Αυτό θα γίνει αναλύοντας την κάθε σελίδα της εφαρμογής μας χωριστά.

- **Login**



KA.VA LTD
Login Page

User Name

Password

Auto login until I logout explicitly
 Save my user name
 Always ask for my user name and password

[Register](#)

Login

Κάπως έτσι ξεκινάνε όλα, η KA.VA Ltd υποδέχεται το χρήστη με μία φόρμα login, μέσω της οποίας μπορεί να εισέλθει ανάλογα με την ιδιότητα του (administrator, αποθηκάριος, μέλος διοίκησης) ή αν δεν ανήκει ακόμα στην εταιρία, μπορεί απλώς να κάνει register, ώστε να εξουσιοδοτηθεί ανάλογα από κάποιον administrator.

Εμείς θα κάνουμε login ως administrator για να συνεχίσουμε την ανάλυση της εφαρμογής έχοντας πρόσβαση σε όλα τα πιθανά πεδία.

Η πρώτη σελίδα λοιπόν που βλέπουμε είναι η παρακάτω:

KA.VA LTD

TABLE: cause trans

[Add](#)

cause id	check exist orig	profit loss orig	check exist dest	profit loss dest	fig supplier	fig customer	cause trans descr (*)				
1			EXIST	PROFIT	EXIST	NOT EXIST	Buy from supplier	View	Edit	Copy	Delete
2	EXIST	LOSS			NOT EXIST	EXIST	Sell to customer	View	Edit	Copy	Delete
3	EXIST	LOSS	EXIST	PROFIT	NOT EXIST	NOT EXIST	Internal transfer	View	Edit	Copy	Delete
4			EXIST	LOSS	EXIST	NOT EXIST	Return to supplier	View	Edit	Copy	Delete
5	EXIST	PROFIT			NOT EXIST	EXIST	Return from customer	View	Edit	Copy	Delete
6			EXIST	PROFIT	NOT EXIST	NOT EXIST	Inventory	View	Edit	Copy	Delete
8	EXIST	LOSS			NOT EXIST	NOT EXIST	Destruction	View	Edit	Copy	Delete

[Add](#)

Page 1 of 1
Records 1 to 7 of 7

[Logout](#)

Πάνω πάνω διακρίνεται το λογότυπο της εταιρίας μας, αριστερά είναι το sitemap, μια στήλη με links, μέσω των οποίων οδηγούμαστε όπου επιθυμούμε μέσα στην εφαρμογή ή κάνουμε logout. Ακολουθούν αναλυτικά, όλα όσα μπορούμε να δούμε στο κέντρο της σελίδας μας, ανάλογα με τον πίνακα που θα επιλέξουμε. Να σημειωθεί πως στα δεξιά κάθε εγγραφής υπάρχουν οι επιλογές “View”, “Edit”, “Copy”, “Delete” που ανάλογα με τα δικαιώματα του, ο κάθε χρήστης μπορεί να διαχειριστεί.

Ας ξεκινήσουμε λοιπόν...


- **Αποθήκες (warehouse)**

Ακολουθώντας μία λογική σειρά για την ευκολότερη κατανόηση της εφαρμογής, θα ξεκινήσουμε με τον πίνακα των αποθηκών. Από εδώ βλέπουμε το όνομα και την τοποθεσία της κάθε αποθήκης μας, καθώς και το ονοματεπώνυμο του αποθηκάρου που τη διαχειρίζεται.

[Add](#)

warehouse id	warehouse name (*)	warehouse location (*)	keeper's name
1	KA.VA Salonika	Thessaloniki	Tasos Pesmetzoglou
2	KA.VA Athina	Athens	Maria Panagiotopoulou
3	KA.VA Crete	Heraklion	Konstantakakis Michalis

[Add](#)

Page  1 of 1
Records 1 to 3 of 3

Add warehouse

[Go Back](#)

warehouse name *	<input type="text"/>
warehouse location *	<input type="text"/>
keeper's name *	<input type="text" value="Christos Vafeas"/> Aleksandros Konstantinou Konstantakakis Michalis

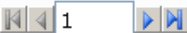
- **Προϊόντα (item)**

Δεύτερος, έρχεται ο πίνακας με τα προϊόντα μας, οι καταχωρήσεις γίνονται ως εξής

[Add](#)

product id	product name (*)	product description (*)	category	subcategory	unit
1	Johnnie Walker Red	Bottle of Red J.W	Whiskey	0,7	Box of 25 bottles
2	Haig	Bottle of Haig	Whiskey	0,7	Box of 25 bottles
3	Cutty Sark	Bottle of Cutty Sark	Whiskey	0,7	Box of 25 bottles
4	Dewar's	Bottle of Dewar's	Whiskey	0,7	Box of 25 bottles
5	Famous	Bottle of Famous Grouse	Whiskey	0,7	Box of 25 bottles
6	Johnnie Walker Red	Bottle of Johnnie W. Red	Whiskey	0,35	Box of 15 bottles
7	Cutty Sark	Bottle of Cutty Sark	Whiskey	0,35	Box of 15 bottles
8	Dewar's	Bottle of Dewar's	Whiskey	0,35	Box of 15 bottles
9	Johnnie Walker Black	Bottle of Johnnie Black	Whiskey	S	Box of 25 bottles
10	Dimple 15 years old	Bottle of Dimple	Whiskey	S	Box of 30 bottles
11	Glenfidich	Bottle of Glenfidich	Whiskey	S	Box of 25 bottles
12	Cardhu	Bottle of Cardhu	Whiskey	S	Box of 25 bottles
13	Absolut	Bottle of Absolut	Vodka	0,7	Box of 25 bottles
14	Stolichnaya	Bottle of Stolischnaya	Vodka	0,7	Box of 25 bottles
15	Grey Goose	Bottle of Grey Goose	Vodka	0,7	Box of 25 bottles
16	Smirnoff	Bottle of Smirnoff	Vodka	0,7	Box of 25 bottles
17	Sherkova	Bottle of Sherkova	Vodka	0,7	Box of 25 bottles
18	Absolut	Bottle of Absolut	Vodka	Litre	Box of 25 bottles
19	Stolichnaya	Bottle of Stolichnaya	Vodka	Litre	Box of 25 bottles
20	Grey Goose	Bottle of Grey Goose	Vodka	Litre	Box of 15 bottles

[Add](#)

Page  1 of 2
Records 1 to 20 of 39

Οι πληροφορίες που παίρνουμε, είναι το όνομα του προϊόντος, η περιγραφή του, η κατηγορία και η υποκατηγορία στην οποία ανήκει καθώς και η μονάδα μέτρησης του κάθε προϊόντος.

Στον πίνακα αυτόν, εκτός από το “Add” του, ενδιαφέρον έχει να δούμε και το search του, το οποίο είναι δυναμικό και βρίσκεται πάνω ακριβώς από το απλό search του κάθε πίνακα. Στο συγκεκριμένο snapshot, έχουμε παραμορφώσει τα πεδία, ανοίγοντας τα τελείως ώστε να διακρίνονται όλες οι πιθανές επιλογές.

TABLE: item

category = Please Select
R&J
Tequila
Vodka

subcategory = Please Select
Tetra
Bottles 1,5L
Cans

unit = Box of 25 cans
Box of 30 Tetra-Paks
Box of 30 bottles
Box of 15 bottles
Box of 20 bottles
Box of 25 bottles

[Show all](#)

Exact phrase All words Any word

Ο πίνακας subcategory αλλάζει δυναμικά, δίνοντας μας μόνο τις τιμές που μπορούμε να επιλέξουμε ανάλογα με την επιλογή μας στο category, ενώ ο πίνακας unit είναι στατικός.

Add item

[Go Back](#)

product name *	<input type="text"/>
product description *	<input type="text"/>
category *	<div>Please Select ▲ R&J Tequila Vodka ▼</div>
subcategory *	<div>Please Select ▲ Litre 0,7 ▼</div>
unit *	<div>Please Select ▲ Box of 25 cans Box of 30 Tetra-Paks Box of 30 bottles Box of 15 bottles Box of 20 bottles ▼</div>

Ακριβώς με τον ίδιο τρόπο που τα πεδία είναι δυναμικά στο search, είναι κι εδώ.

- **Κατηγορίες Προϊόνων (*item_cat*)**



Σε αυτόν τον βοηθητικό πίνακα, έχουμε εισάγει όλες τις διαθέσιμες κατηγορίες προϊόντων, καθώς και την περιγραφή τους, ώστε να μας διευκολύνουν κάνοντας πολύ πιο απλό και κατατοπιστικό τον πίνακα *item*.

TABLE: item_cat

[Add](#)

<u>class id</u>	<u>category</u>	<u>description</u>				
1	Whiskey	Our bottles of Whiskey	View	Edit	Copy	Delete
2	Vodka	Our bottles of Vodka	View	Edit	Copy	Delete
3	Tequila	Our bottles of tequila	View	Edit	Copy	Delete
4	R&J	Refreshments and juices	View	Edit	Copy	Delete

[Add](#)

Page  1  of 1
Records 1 to 4 of 4

Add item_cat

Add to TABLE: item_cat

[Go Back](#)

category *	<input type="text"/>
description *	<input type="text"/>

- **Υποκατηγορίες Προϊόντων (item_subcat)**

Όπως και στον item_cat, καταχωρούμε την υποκατηγορία και την περιγραφή της υποκατηγορίας για να συγκροτήσουμε αυτόν τον επίσης βοηθητικό πίνακα. Αυτή τη φορά όμως, έχοντας κάνει "look up" στον πρώτο, παράλληλα με την υποκατηγορία, βλέπουμε και την κατηγορία του εκάστοτε προϊόντος.

[Add](#)

subclass id	category	subcategory	description				
1	Whiskey	0,7	Bottles of 700ml	View	Edit	Copy	Delete
2	Whiskey	0,35	Bottles of 0,35ml	View	Edit	Copy	Delete
3	Whiskey	S	Special editions	View	Edit	Copy	Delete
4	Vodka	0,7	Bottles of 700ml	View	Edit	Copy	Delete
5	Vodka	Litre	Bottles of 1L	View	Edit	Copy	Delete
6	Tequila	Yellow	Bottles of yellow tequila	View	Edit	Copy	Delete
7	Tequila	White	Bottles of white tequila	View	Edit	Copy	Delete
8	R&J	Cans	Canned refreshments	View	Edit	Copy	Delete
9	R&J	Bottles 1,5L	Bottled refreshments 1,5L	View	Edit	Copy	Delete
10	R&J	Tetra	Juices in Tetra-Pak	View	Edit	Copy	Delete

[Add](#)

Page of 1
Records 1 to 10 of 10

Add item_subcat

[Go Back](#)

category *	<input type="text" value="Please select"/> <div style="border: 1px solid #ccc; padding: 2px;"> ▲ ▼ ☰ R&J Tequila Vodka Whiskey </div>
subcategory *	<input type="text"/>
description *	<input type="text"/>

Από εδώ καταχωρούμε τις πιθανές υποκατηγορίες των προϊόντων. Η κατηγορία επιλέγεται μέσω αναδιπλούμενης λίστας που περιέχει τις δυνατές επιλογές μας. Όπως έχουμε κάνει στις περισσότερες αναδιπλούμενες λίστες, έτσι κι εδώ, έχουμε τραβήξει τα όρια της για να βλέπουμε τις πιθανές επιλογές. Στην εφαρμογή όμως είναι όλες μαζεμένες δεσμεύοντας μόλις μια γραμμή πριν αναδιπλωθούν όπως θα δούμε και παρακάτω.


- **Μονάδες Μέτρησης (unit_table)**

Ο πίνακας αυτός, είναι καθαρά βοηθητικός και εδώ απλά καταχωρούνται οι διάφορες πιθανές μονάδες μέτρησης οι οποίες στη συνέχεια θα αντιστοιχηθούν στα προϊόντα.

[Add](#)

unit_id	unit_description				
1	Box of 25 bottles	View	Edit	Copy	Delete
2	Box of 20 bottles	View	Edit	Copy	Delete
3	Box of 15 bottles	View	Edit	Copy	Delete
4	Box of 30 bottles	View	Edit	Copy	Delete
5	Box of 30 Tetra-Paks	View	Edit	Copy	Delete
6	Box of 25 cans	View	Edit	Copy	Delete

[Add](#)

Page  1 of 1
Records 1 to 6 of 6

Add unit_table

[Go Back](#)

unit id *	<input type="text"/>
unit description *	<input type="text"/>

- **Απόθεμα (Product_quantity)**

Ο πίνακας με τις ποσότητες των προϊόντων, αποτελεί έναν από τα πιο εύχρηστα εργαλεία της εφαρμογής μας. Μας επιτρέπει να βλέπουμε, να δεσμεύουμε, και να ελέγχουμε τα αποθέματα του εκάστοτε προϊόντος στην κάθε αποθήκη. Λέγοντας «δεσμεύουμε», εννοούμε πως καταχωρούμε τις παραγγελίες των πελατών οι οποίες δεν εκτελούνται άμεσα (π.χ επειδή δεν υπάρχει εκείνη τη στιγμή η συγκεκριμένη ποσότητα προϊόντος στη συγκεκριμένη αποθήκη). Μετά την περάτωση της κίνησης ή σε περίπτωση ακύρωσης της παραγγελίας διαγράφουμε τη δέσμευση. Παρακάτω, έχουμε μία εικόνα με τυχαίες τιμές για το κάθε προϊόν:

product name	warehouse location	quantity bounded	quantity total
Olmeca / Yellow	Heraklion	0	21
Dimple 15 years old / S	Athens	0	34
Dewar's / 0,7	Thessaloniki	0	45
Johnnie Walker Red / 0,7	Heraklion	0	15
Coca Cola / 1,5L	Heraklion	0	17
Sierra / Yellow	Athens	10	20
Stolichnaya / 0,7	Thessaloniki	0	50
Johnnie Walker Red / 0,7	Thessaloniki	17	50

Ας δούμε λοιπόν το προϊόν που έχουμε highlighted. Είναι η κίτρινη Sierra. Η συνολική λοιπόν ποσότητα του συγκεκριμένου προϊόντος στην αποθήκη είναι 20 κούτες των 25 μπουκαλιών (το συγκεκριμένο προϊόν μετριέται σε κούτες 25 μπουκαλιών όπως είδαμε προηγουμένως στον πίνακα unit_tabe). Όπως βλέπουμε στο πεδίο “quantity bounded”, έχουμε δεσμεύσει 10 κούτες. Μέσω extended search θα μπορούμε να ψάχνουμε με κάθε δυνατό τρόπο την πληροφορία που μας ενδιαφέρει. Ο πίνακας, ενημερώνεται αυτόματα, αμέσως μετά από κάθε κίνηση που καταχωρείται στον product_trans και αμέσως μετά από κάθε δέσμευση που γίνεται στον bounds_trans.

Add product_quantity

Εδώ δεν υπάρχει η επιλογή add αφού δε μπορούμε να εισάγουμε αυθαίρετα, ό,τι προϊόν θέλουμε. Το «Απόθεμα» απλώς μας δίνει συγκεντρωτικές πληροφορίες. Οι προσθήκες προϊόντων στις αποθήκες μας γίνονται μόνω μέσω αγορών από

προμηθευτές ή μετά από απογραφή (χρησιμοποιώντας τον πίνακα κινήσεων), ενώ οι δεσμεύσεις γίνονται μόνο μέσω του πίνακα δεσμεύσεων.

- **Χρήστες (user)**

Φτάσαμε λοιπόν στον πίνακα καταχώρησης των χρηστών της εφαρμογής μας. Από εδώ βλέπουμε τα μέλη της επιχείρησης, την ιδιότητα τους (και αντίστοιχα τα δικαιώματά τους στην εφαρμογή μας), τα username και τα passwords τους, όπως επίσης και το αν είναι ενεργοί ή όχι. Είναι ένας πίνακας, ο οποίος είναι ορατός μόνο από τους administrators. Τέλος, μέσω αυτού του πίνακα, οι τελευταίοι ορίζουν την αρμοδιότητα του κάθε χρήστη αφού κάνει register.

Αναλυτικά οι ιδιότητες:

- Αν κάποιος χρήστης κάνει register θα δει μια λευκή οθόνη με το μήνυμα “ You do not have the right permission to view the page” και η μοναδική του επιλογή θα είναι: “Back to Login Page”. Θα πρέπει να περιμένει λοιπόν μέχρι να του δώσει αρμοδιότητα (keeper, company, administrator) κάποιος administrator).
- Αν οι administrators ορίσουν κάποιον ως μέλος διοίκησης (company), μπορεί να δει όλους τους πίνακες και τα πεδία τους, εκτός από τον users, αλλά δε μπορούμε να κάνουμε edit, copy ή delete σε κανέναν από αυτούς.
- Αν οι administrators ορίσουν κάποιον ως keeper, έχει τη δυνατότητα να δει, να επεξεργαστεί ή και να διαγράψει όλα τα πεδία των πινάκων πλην εκείνα του πίνακα users.
- Οι ίδιοι οι administrators μπορούν να ελέγξουν τα πάντα, έχοντας πρόσβαση σε όλα τα πεδία των πινάκων και δικαιοδοσία να τα αλλάξουν αλλά και να τα διαγράψουν.

Add

user id	user ident (*)	user pass (*)	user full name (*)	attribute	status			
7	ant-mi	caleb	antonis kakourakis	Administrator	active	View	Edit	Copy
5	t_pes	1985	Tasos Pesmetzoglou	Keeper	active	View	Edit	Copy
6	maria	mar	Maria Panagiotopoulou	Keeper	active	View	Edit	Copy
8	vaf	25167	Christos Vafeas	Administrator	active	View	Edit	Copy
9	catboom	12345	Aleksandros Konstantinou	Keeper	inactive	View	Edit	Copy
10	ofi4	ofara	Konstantakakis Michalis	Keeper	active	View	Edit	Copy
15	stella	stan	stella stefanaki	Company	active	View	Edit	Copy
16	asd	asd	asdasd	Default	active	View	Edit	Copy

Add user

[Go Back](#)

user ident *	<input type="text"/>
user pass *	<input type="password"/>
user full name *	<input type="text"/>
attribute *	<input type="text" value="Please Select"/>
status *	<input type="text" value="Please Select"/>

- Please Select
- Administrator
- Default
- Keeper

- **Πελάτες (customer)**

[Add](#)

customer id	company (*)	customer surname (*)	customer name (*)	customer address (*)	location (*)	customer telephone (*)	status
1	Vivalco	Papadopoulos	Ioannis	Kremou 19	Mosxato, Athens	2109421256	active
2	Popota	Jefferson	Jonas	Academy Avenue 243	London (England)	4721082161	active
3	Naosia	Nikolaou	Panagiotis	Lappa 67	Naoussa	6945677291	active
4	Piot-Comp	Konstantopoulos	Ilias	Thiseos 54	Kallithea, Athens	2103285920	inactive
5	Canava Roussos	Roussos	Georgios	Fira	Santorini	2286032204	active
6	Cretan Alcohol	Xaniotakis	Manolis	Kalokairinou 34	Heraklion	2810837219	inactive

[Add](#)

Page of 1
Records 1 to 6 of 6

Θα ξεκινήσουμε με τον πίνακα των πελατών μας. Εδώ βλέπουμε, όλα τα ονόματα των πελατών μας, εν ενεργεία ή μη, καθώς και την επωνυμία της εταιρίας τους, τη διεύθυνση, την περιοχή που εδρεύει η εταιρία τους καθώς και το τηλέφωνο

επικοινωνίας. Όπως θα δούμε και παρακάτω, πάνω και κάτω από κάθε πίνακα, υπάρχει η επιλογή “add” που μας επιτρέπει να δημιουργήσουμε μια νέα εγγραφή στον εκάστοτε πίνακα. Επίσης, κάτω βλέπουμε ένα εργαλείο, ώστε να μπορούμε να κινούμαστε ανάμεσα στις σελίδες που μας ενδιαφέρουν.

Add customer

Ας δούμε λοιπόν τη φόρμα εγγραφής που παίρνουμε πατώντας “Add” στον πίνακα των πελατών.

Add to TABLE: customer

[Go Back](#)

company *	<input type="text"/>
customer surname *	<input type="text"/>
customer name *	<input type="text"/>
customer address *	<input type="text"/>
location *	<input type="text"/>
customer telephone *	<input type="text"/>
status *	<input type="radio"/> active <input type="radio"/> inactive

- *Προμηθευτές (supplier)*

Ένας ακόμα βοηθητικός πίνακας. Στον πίνακα των προμηθευτών, καταχωρούμε όλους τους προμηθευτές με τους οποίους η εταιρία έχει συνεργαστεί.

[Add](#)

supplier id	company (*)	supplier surname (*)	supplier name (*)	supplier address (*)	location (*)	supplier telephone (*)	status
2	Volcano	Sigalas	Artemis	Firostefani	Santorini	6937826427	active
3	Atlas	Ioannidis	Konstantinos	Stratigou Kaliermou 87	Thessaloniki	6977281974	active
4	Kosmopot	Mavros	Antonis	Eriantis 39	Kastoria	6934871909	active
5	Cabron Buenas	Lopez	Martino	escudas 342	Madrid (Spain)	5782920132	active
6	Alcoola Italiano	Totti	Lorenzo	toscania 39	Milano (Italy)	2380598232	active
7	O Pontios	Triantafyllidis	Kosmas	pagkaiou 100	Kalamaria	6992819162	inactive

[Add](#)

Page of 1
Records 1 to 6 of 6

Αναλυτικά από αριστερά προς τα δεξιά του πίνακα, βλέπουμε τον κωδικό του προμηθευτή, το όνομα της εταιρίας του, το ονοματεπώνυμο του, η διεύθυνση και η

τοποθεσία της εταιρίας, το τηλέφωνο επικοινωνίας, καθώς και το αν είναι ενεργός προς την εταιρία μας ή όχι.

Add supplier

[Go Back](#)

company *	<input type="text"/>
supplier surname *	<input type="text"/>
supplier name *	<input type="text"/>
supplier address *	<input type="text"/>
location *	<input type="text"/>
supplier telephone *	<input type="text"/>
status *	<input checked="" type="radio"/> active <input type="radio"/> inactive

- ***Κινήσεις (product_trans)***

Η πεμπουσία της εφαρμογής μας. Ο κεντρικός πίνακας που ελέγχει τις κινήσεις ο οποίος περιλαμβάνει πολύ χρήσιμο extended search. Συγκεκριμένα, μπορούμε να ζητήσουμε να εμφανιστούν μόνο οι κινήσεις που μας ενδιαφέρουν, ανάλογα με τον τύπο τους, την αποθήκη προέλευσης ή προορισμού που αφορούν, τον προμηθευτή ή/και τον πελάτη.

Add

transaction number	type of transaction	warehouse origin id	warehouse destination id	date	quantity (*)	supplier	customer	product
109	Buy from supplier		KA.VA Crete	09/01/2008	21	Volcano		Olmecca / Yellow
108	Buy from supplier		KA.VA Athina	09/01/2008	34	Cabron Buenas		Dimple 15 years old / S
107	Buy from supplier		KA.VA Salonika	09/01/2008	45	Alcoola Italiano		Dewar's / 0,7
106	Destruction	KA.VA Salonika		08/01/2008	12			Johnnie Walker Red / 0,7
105	Return from customer	KA.VA Salonika		08/01/2008	2		Vivalco	Johnnie Walker Red / 0,7
104	Return to supplier		KA.VA Crete	08/01/2008	3	Kosmopot		Coca Cola / 1,5L
103	Internal transfer	KA.VA Salonika	KA.VA Crete	08/01/2008	15			Johnnie Walker Red / 0,7
102	Sell to customer	KA.VA Salonika		08/01/2008	25		Vivalco	Johnnie Walker Red / 0,7
101	Buy from supplier		KA.VA Crete	08/01/2008	20	Kosmopot		Coca Cola / 1,5L
100	Inventory		KA.VA Athina	08/01/2008	20			Sierra / Yellow

Βλέπουμε λοιπόν στα παραδείγματα κινήσεων από τα αριστερά προς τα δεξιά τον αριθμό της κίνησης, τον τύπο της κίνησης, την αποθήκη προέλευσης, την αποθήκη προορισμού, την ημερομηνία που πραγματοποιήθηκε η κίνηση, την ποσότητα που κινήθηκε, το όνομα της εταιρίας του προμηθευτή, το όνομα της εταιρίας του πελάτη καθώς και το προϊόν που θα κινηθεί. Εδώ πρέπει να επισημάνουμε πως έχουμε προκαθορίσει ότι η εταιρία μας δε θα πραγματοποιεί συναλλαγές με ιδιώτες παρά μόνο με άλλες εταιρίες.

Add product_trans

[Go Back](#)

type of transaction *	Sell to customer ▾
warehouse origin id *	Please Select ▾
warehouse destination id *	Please Select ▾
date *	<input type="text"/>
quantity *	<input type="text"/>
supplier	Please Select ▾
customer	Please Select ▾
product *	Please Select ▾

Add

Ο πιο δυναμικός πίνακας της εφαρμογής μας. Λειτουργεί ως εξής:

Στο “type of transaction” επιλέγουμε μία από τις πιθανές κινήσεις όπως βλέπουμε στο παρακάτω snapshot

[Go Back](#)

type of transaction *	Please Select ▾
warehouse origin id	Please Select
warehouse destination id	Buy from supplier
date *	Sell to customer
quantity *	Internal transfer
supplier	Return to supplier
customer	Return from customer
product *	Inventory
	Destruction
	Please Select ▾
	Please Select ▾

Add

Έστω ότι επιλέγουμε “Sell to customer” λοιπόν. Αυτόματα, τα πεδία που δε χρησιμοποιούνται (warehouse destination id, supplier) παύουν να είναι διαθέσιμα και συμπληρώνονται μόνο τα υπόλοιπα. Το ποια πεδία θα είναι διαθέσιμα ανάλογα με την εκάστοτε κίνηση ορίζεται από τον «κρυφό» μας πίνακα, cause_trans. Έστω λοιπόν ότι θέλουμε να πουλήσουμε 23 Haig 700ml σε ένα πελάτη. Αρχικά γίνεται έλεγχος αν υπάρχει η επιθυμητή ποσότητα διαθέσιμη στην αποθήκη μας και κατόπιν θα αφαιρεθεί από αυτή. Από την άλλη, αν δεν υπάρχει η ποσότητα που θέλουμε, μας βγάζει μήνυμα λάθους. Ο έλεγχος για τον αν υπάρχει μία ποσότητα ή όχι, γίνεται στον πίνακα product_quantity από τον οποίο ενημερώνεται ο product_trans.

- **Δεσμεύσεις (bounds_trans)**

Οι Δεσμεύσεις, είναι ο πίνακας που μας δείχνει τις παραγγελίες τις οποίες έχουν κάνει οι πελάτες της KA.VA Ltd. Με το που κάνουμε add μια δέσμευση, ενημερώνεται αυτόματα το αντίστοιχο πεδίο του πίνακα Απόθεμα (product_quantity).

[Add](#)

warehouse id	product id	customer id	quantity bounded
KA.VA Athina	Sierra / Yellow	Cretan Alcohol	10
KA.VA Salonika	Johnnie Walker Red / 0,7	Naoosia	17

Αναλυτικά...

[Go Back](#)

warehouse id *	Please Select ▾
product id *	Please Select ▾
customer id *	Please Select ▾
quantity bounded *	<input type="text"/>
<input type="button" value="Add"/>	

- **Reports**

Τελευταία αλλά πολύ σημαντικά είναι τα Reports τα οποία παράγει η εφαρμογή μας. Τα Reports, είναι συγκεντρωτικές αναφορές σε φιλική προς εκτύπωση μορφή, οι οποίες μας ομαδοποιούν συγκεκριμένες πληροφορίες σχετικά με τις αποθήκες και τα προϊόντα μας ανά πάσα στιγμή. Η εφαρμογή μας περιλαμβάνει τα εξής 4 reports (παρατίθεται η πρώτη μόνο σελίδα από κάθε report):

1. Συνολική Ποσότητα Ανά Προϊόν

KA.VA LTD	
Report: Συνολική Ποσότητα/Προϊόν	
product name	Coca Cola / 1,5L
	SUM(`quantity total `)
Summary for product name: Coca Cola / 1,5L (1 Detail Records)	
Sum	17
product name	Dewar's / 0,7
	SUM(`quantity total `)
Summary for product name: Dewar's / 0,7 (1 Detail Records)	
Sum	45
product name	Dimple 15 years old / S
	SUM(`quantity total `)
Summary for product name: Dimple 15 years old / S (1 Detail Records)	
Sum	34

2. Συνολική Ποσότητα Ανά Αποθήκη

KA.VA LTD

Report: Σ.Ποσ./Αποθήκη

warehouse name	KA.VA Athina
	SUM(`quantity total `)

Summary for warehouse name: KA.VA Athina (1 Detail Records)

Sum	54
------------	----

warehouse name	KA.VA Crete
	SUM(`quantity total `)

Summary for warehouse name: KA.VA Crete (1 Detail Records)

Sum	53
------------	----

warehouse name	KA.VA Salonika
	SUM(`quantity total `)

Summary for warehouse name: KA.VA Salonika (1 Detail Records)

Sum	145
------------	-----

3. Συνολική Ποσότητα Ανά Κατηγορία Και Αποθήκη

KA.VA LTD

Report: Σ.Ποσ./Κατ.+Αποθήκη

warehouse name	KA.VA Athina
category	Tequila
	SUM(`quantity total `)
Summary for category: Tequila (1 Detail Records)	
Sum	20
category	Whiskey
	SUM(`quantity total `)
Summary for category: Whiskey (1 Detail Records)	
Sum	34
Summary for warehouse name: KA.VA Athina (2 Detail Records)	
Sum	54
warehouse name	KA.VA Crete
category	R&J
	SUM(`quantity total `)
Summary for category: R&J (1 Detail Records)	
Sum	17

4. Δεσμευμένη Ποσότητα Ανά Κατηγορία Και Αποθήκη

KA.VA LTD

Report: Δεσμ.Ποσ./Κατ+Αποθήκη

warehouse name	KA.VA Athina
category	Tequila
	quantity bounded
Summary for category: Tequila (1 Detail Records)	
Sum	10
category	Whiskey
	quantity bounded
Summary for category: Whiskey (1 Detail Records)	
Sum	0
Summary for warehouse name: KA.VA Athina (2 Detail Records)	
Sum	10
warehouse name	KA.VA Crete
category	R&J
	quantity bounded

ΠΑΡΑΡΤΗΜΑ

Why PHP and MySQL?

This first chapter is an introduction to PHP, MySQL, and the interaction of the two. In it, we'll try to address some of the most common questions about these tools, such as "What are they?" and "How do they compare to similar technologies?" Most of the chapter is taken up with an enumeration of the many, many reasons to choose PHP, MySQL, or the two in tandem. If you're a techie looking for some ammunition to lob at your PHB ("Pointy-Haired Boss" for those who don't know the Dilbert cartoons) or a manager asking yourself what is this P-whatever thing your geeks keep whining to get, this chapter will provide some preliminary answers.

What Is PHP?

PHP is the Web development language written by and for Web developers. PHP stands for *PHP: Hypertext Preprocessor*. The product was originally named *Personal Home Page Tools*, and many people still think that's what the acronym stands for. But as it expanded in scope, a new and more appropriate (albeit GNU-ishly recursive) name was selected by community vote. PHP is currently in its fifth major rewrite, called PHP5 or just plain PHP.

PHP is a server-side scripting language, which can be embedded in HTML or used as a standalone binary (although the former use is much more common). Proprietary products in this niche are Microsoft's Active Server Pages, Macromedia's ColdFusion, and Sun's Java Server Pages. Some tech journalists used to call PHP "the open source ASP" because its functionality is similar to that of the Microsoft product—although this formulation was misleading, as PHP was developed before ASP. Over the past few years, however, PHP and server-side Java have gained momentum, while ASP has lost mindshare, so this comparison no longer seems appropriate.

We'll explore server-side scripting more thoroughly in Chapter 2, but for the moment you can think of it as a collection of super-HTML tags or small programs that run inside your Web pages—except on the server side, before they get sent to the browser. For example, you can use PHP to add common headers and footers to all the pages on a site or to store form-submitted data in a database.

Strictly speaking, PHP has little to do with layout, events, on the fly DOM manipulation, or really anything about what a Web page looks and sounds like. In fact, most of what PHP does is invisible to the end user. Someone looking at a PHP page will not necessarily be able to tell that it was not written purely in HTML, because usually the result of PHP *is* HTML.

PHP is an official module of Apache HTTP Server, the market-leading free Web server that runs about 67 percent of the World Wide Web (according to the widely quoted Netcraft Web server survey). This means that the PHP scripting engine can be built into the Web server itself, leading to faster processing, more efficient memory allocation, and greatly simplified maintenance. Like Apache Server, PHP is fully cross-platform, meaning it runs native on several flavors of Unix, as well as on Windows and now on Mac OS X. All projects under the aegis of the Apache Software Foundation—including PHP—are open source software.

What Is MySQL?

MySQL (pronounced My Ess Q El) is an open source, SQL Relational Database Management System (RDBMS) that is free for many uses (more detail on that later). Early in its history, MySQL occasionally faced opposition due to its lack of support for some core SQL constructs such as subselects and foreign keys. Ultimately, however, MySQL found a broad, enthusiastic user base for its liberal licensing terms, perky performance, and ease of use. Its acceptance was aided in part by the wide variety of other technologies such as PHP, Java, Perl, Python, and the like that have encouraged its use through stable, well-documented modules and extensions. MySQL has not failed to reward the loyalty of these users with the addition of both subselects and foreign keys as of the 4.1 series.

Databases in general are useful, arguably the most consistently useful family of software products—the “killer product” of modern computing. Like many competing products, both free and commercial, MySQL isn’t a database until you give it some structure and form. You might think of this as the difference between a database and an RDBMS (that is, RDBMS plus user requirements equals a database).

There’s lots more to say about MySQL, but then again, there’s lots more space in which to say it.

The History of PHP

Rasmus Lerdorf—software engineer, Apache team member, and international man of mystery—is the creator and original driving force behind PHP. The first part of PHP was developed for his personal use in late 1994. This was a CGI wrapper that helped him keep track of people who looked at his personal site. The next year, he put together a package called the *Personal Home Page Tools* (a.k.a. the *PHP Construction Kit*) in response to demand from users who had stumbled into his work by chance or word of mouth. Version 2 was soon released under the title PHP/FI and included the *Form Interpreter*, a tool for parsing SQL queries.

By the middle of 1997, PHP was being used on approximately 50,000 sites worldwide. It was clearly becoming too big for any single person to handle, even someone as focused and energetic as Rasmus. A small core development team now runs the project on the open source “benevolent junta” model, with contributions from developers and users around the world.

Zeev Suraski and Andi Gutmans, the two Israeli programmers who developed the PHP3 and PHP4 parsers, have also generalized and extended their work under the rubric of Zend.com (*Zeev, Andi, Zend, get it?*).

The fourth quarter of 1998 initiated a period of explosive growth for PHP, as all open source technologies enjoyed massive publicity. In October 1998, according to the best guess, just over 100,000 unique domains used PHP in some way. Just over a year later, PHP broke the one-million domain mark. When we wrote the first edition of this book in the first half of 2000, the number had increased to about two million domains. As we write this, approximately 15 million public Web servers (in the software sense, not the hardware sense) have PHP installed on them.

Public PHP deployments run the gamut from mass-market sites such as Excite Webmail and the Indianapolis 500 Web site, which serve up millions of pageviews per day, through “massniche” sites such as Sourceforge.net and Epinions.com, which tend to have higher functionality needs and hundreds of thousands of users, to e-commerce and brochureware sites such as The Bookstore at Harvard.com and Sade.com (Web home of the British singer), which must be visually attractive and easy to update. There are also PHP-enabled parts of sites, such as the forums on the Internet Movie Database (imdb.com); and a large installed base of nonpublic PHP deployments, such as LDAP directories (MCI WorldCom built one with over 100,000 entries) and trouble-ticket tracking systems.

In its newest incarnation, PHP5 strives to deliver something many users have been clamouring for over the past few years: much improved object-oriented programming (OOP) functionality.

PHP has long nodded to the object programming model with functions that allow object programmers to pull out results and information in a way familiar to them. These efforts still fell short of the ideal for many programmers, however, and efforts to force PHP to build in fully object-oriented systems often yielded unintended results and hurt performance. PHP5’s newly rebuilt object model brings PHP more in line with other object-oriented languages such as Java and C++, offering support for features such as overloading, interfaces, private member variables and methods, and other standard OOP constructions.

With the crash of the dot-com bubble, PHP is poised to be used on more sites than ever. Demand for Web-delivered functionality has decreased very little, and emerging technological standards continue to pop up all the time, but available funding for hardware, licenses, and especially headcount has drastically decreased. In the post-crash Web world, PHP’s shallow learning curve, quick implementation of new functionality, and low cost of deployment are hard arguments to beat.

The History of MySQL

Depending on how much detail you want, the history of MySQL can be traced as far back as 1979, when MySQL's creator, Monty Widenius, worked for a Swedish IT and data consulting firm, TcX. While at TcX, Monty authored UNIREG, a terminal interface builder that connected to raw ISAM data stores. In the intervening 15 years, UNIREG served its makers rather well through a series of translations and extensions to accommodate increasingly large data sets.

In 1994, when TcX began working on Web data applications, chinks in the UNIREG armor, primarily having to do with application overhead, began to appear. This sent Monty and his colleagues off to look for other tools. One they inspected rather closely was Hughes mSQL, a light and zippy database application developed by David Hughes. mSQL possessed the distinct advantages of being inexpensive and somewhat entrenched in the market, as well as featuring a fairly well-developed client API. The 1.0 series of mSQL release lacked indexing, however, a feature crucial to performance with large data stores. Although the 2.0 series of mSQL would see the addition of this feature, the particular implementation used was not compatible with UNIREG's B+-based features. At this point, MySQL, at least conceptually, was born.

Monty and TcX decided to start with the substantial work already done on UNIREG while developing a new API that was substantially similar to that used by mSQL, with the exception of the more effective UNIREG indexing scheme. By early 1995, TcX had a 1.0 version of this new product ready. They gave it the moniker MySQL and later that year released it under a combination open source and commercial licensing scheme that allowed continued development of the product while providing a revenue stream for MySQL AB, the company that evolved from TcX.

Over the past ten years, MySQL has truly developed into a world class product. MySQL now competes with even the most feature-rich commercial database applications such as Oracle and Informix. Additions in the 4.x series have included much-requested features such as transactions and foreign key support. All this has made MySQL the world's most used open source database.

Reasons to Love PHP and MySQL

There are ever so many reasons to love PHP and MySQL. Let us count a few.

Cost

PHP costs you nothing. Zip, zilch, nada, not one red cent. Nothing up front, nothing over the lifetime of the application, nothing when it's over. Did we mention that the Apache/PHP/MySQL combo runs great on cheap, low-end hardware that you couldn't even *think* about for IIS/ASP/SQL Server?

MySQL is a slightly different animal in its licensing terms. Before you groan at the concept of actually using commercial software, consider that although MySQL is open-source licensed for many uses, it is not and has never been primarily community-developed software. MySQL AB is a commercial entity with necessarily commercial interests. Unlike typical open source projects, where developers often have regular full-

time (and paying) day jobs in addition to their freely given open source efforts, the MySQL developers derive their primary income from the project. There are still many circumstances in which MySQL can be used for free (basically anything nonredistributive, which covers most PHP-based projects), but if you make money developing solutions that use MySQL, consider buying a license or a support contract. It's still infinitely more reasonable than just about any software license you will ever pay for.

For purposes of comparison, Table 1-1 shows some current retail figures for similar products in the United States. All prices quoted are for a single-processor public Web server with the most common matching database and development tool; \$0 means a no-cost alternative is a common real-world choice.

Table 1-1: Comparative Out-of-Pocket Costs

ASP/SQL ColdFusion

Item Server MX/SQL Server JSP/Oracle PHP/MySQL

Development tool	\$0–2499	\$599	\$0–~2000	\$0–249
Server	\$999	\$2298	\$0–~35,000	\$0
RDBMS	\$4999	\$4999	\$15,000	\$0–220

Open source software: don't fear the cheaper

But as the bard so pithily observed, we are living in a material world—where we've internalized maxims such as, “You get what you pay for,” “There's no such thing as a free lunch,” and “Things that sound too good to be true usually are.” You (or your boss) may, therefore, have some lingering doubts about the quality and viability of no-cost software. It probably doesn't help that until recently software that didn't cost money—formerly called *freeware*, *shareware*, or *free software*—was generally thought to fall into one of three categories:

- Programs filling small, uncommercial niches
- Programs performing grungy, low-level jobs
- Programs for people with bizarre socio-political issues

It's time to update some stereotypes once and for all. We are clearly in the middle of a sea change in the business of software. Much (if not most) major consumer software is distributed without cost today; e-mail clients, Web browsers, games, and even full-service office suites are all being given away as fast as their makers can whip up Web versions or set up FTP servers. Consumer software is increasingly seen as a loss-leader, the flower that attracts the pollinating honeybee—in other words, a way to sell more server hardware, operating systems, connectivity, advertising, optional widgets, or stock shares. The full retail price of a piece of software, therefore, is no longer a reliable gauge of its quality or the eccentricity-level of its user.

On the server side, open source products have come on even stronger. Not only do they compete with the best commercial stuff; in many cases there's a feeling that they far exceed the competition. Don't take our word for it! Ask IBM, any hardware manufacturer, NASA,

Amazon.com, Rockpointe Broadcasting, Ernie Ball Corporation, the Queen of England, or the Mexican school system. If your boss still needs to be convinced, further ammunition is available at www.opensource.org and www.fsf.org.

The PHP license

The freeness of open source and Free software is guaranteed by a gaggle of licensing schemes, most famously the *GPL* (*Gnu General Public License*) or *copyleft*. PHP used to be released under both the GPL and its own license, with each user free to choose between them. This has recently changed. The program as a whole is now released under its own extremely laissez-faire PHP license on the model of the BSD license, whereas Zend as a standalone product is released under the *Q Public License* (this clause applies *only* if you unbundle Zend from PHP and try to sell it).

You can read the fine print about the relevant licenses at these Web sites:

www.php.net/license/

www.mysql.com/doc/en/GPL_license.html

www.troll.no/qpl/annotated.html

Most people get PHP or MySQL via download, but you may have paid for it as part of a Linux distribution, a technical book, or some other product. In that case, you may now be silently disputing our assertion that PHP costs nothing. Here's the twist: Although you can't require a fee for most open source software, you *can* charge for delivering that software in a more convenient format—such as by putting it on a disk and shipping the disk to the customer. You can also charge anything the market will bear for being willing to perform certain services or accept certain risks that the development team may not wish to undertake. For instance, you are allowed to charge money for guaranteeing that every copy of the software you distribute will be virus-free or of reasonable quality, taking on the risk of being sued if a bunch of customers get bad CD-ROMs that contain hard-drive-erasing viruses.

Usually, open source software users can freely choose the precisely optimal cost-benefit equation for each particular situation: no cost and no warranties, or expensive but well supported, or something in between. No organized attempt has been made yet to sell service and support for PHP (although presumably that will be one of the value-adds of Zend). MySQL AB does sell support as part some of its licensing packages for the MySQL product. Other open source products, such as Linux, have companies such as Red Hat standing by to answer your questions, but the commercialization process is still in the early stages for PHP.

Ease of Use

PHP is easy to learn, compared to the other ways to achieve similar functionality. Unlike Java Server Pages or C-based CGI, PHP doesn't require you to gain a deep understanding of a major programming language before you can make a trivial database or remote-server call.

Unlike Perl, which has been semijokingly called a “write-only language,” PHP has a syntax that is quite easy to parse and human-friendly. And unlike ASP.NET, PHP is stable and ready to solve your problems today.

Many of the most useful specific functions (such as those for opening a connection to an

Oracle database or fetching e-mail from an IMAP server) are predefined for you. A lot of complete scripts are waiting out there for you to look at as you're learning PHP. In fact, it's entirely possible to use PHP just by modifying freely available scripts rather than starting from scratch—you'll still need to understand the basic principles, but you can avoid many frustrating and time-consuming minor mistakes.

We must mention one caveat: *Easy* means different things to different people, and for some Web developers it has come to connote a graphical, drag-and-drop, What You See Is What You Get development environment. To become truly proficient at PHP, you need to be comfortable editing HTML by hand. You can use WYSIWYG editors to design sites, format pages, and insert client-side features before you add PHP functionality to the source code. There are even ways, which we'll detail in Chapter 3, to add PHP functions to your favorite editing environment.

It's not realistic, however, to think you can take full advantage of PHP's capabilities without ever looking at source code.

Most advanced PHP users (including most of the development team members) are diehard hand-coders. They tend to share certain gut-level, subcultural assumptions—for instance, that hand-written code is beautiful and clean and maximally browser-compatible and therefore the only way to go—that they do not hesitate to express in vigorous terms. The PHP community offers help and trades tips mostly by e-mail, and if you want to participate, you have to be able to parse plain-text source code with facility. Some WYSIWYG users occasionally ask list members to diagnose their problems by looking at their Web pages instead of their source code, but this rarely ends well.

That said, let us reiterate that PHP really is easy to learn and write, especially for those with a little bit of experience in a C-syntaxed programming language. It's just a little more involved than HTML but probably simpler than JavaScript and definitely less conceptually complex than JSP or ASP.NET.

If you have no relational database experience or are coming from an environment such as Microsoft Access, MySQL's command line interface and lack of implicit structure may at first seem a little daunting. Again, the word *easy* is relative. However, MySQL's increasingly faithful adherence to the ANSI SQL-92 standard and a comprehensive suite of external client programs, coupled with graphical administration tools such as PHPMyAdmin and the new MySQL Control Center, will get even neophyte users up and running quickly compared to other databases. None of these will substitute for learning a little theory and employing good design practices, but that subject is for another chapter.

HTML-embeddedness

PHP is embedded within HTML. In other words, PHP pages are ordinary HTML pages that escape into PHP mode only when necessary. Here is an example:

```
<HEAD>
<TITLE>Example.com greeting</TITLE>
</HEAD>
<BODY>
<P>Hello,
<?php
// We have now escaped into PHP mode.
```

```

// Instead of static variables, the next three lines
// could easily be database calls or even cookies;
// or they could have been passed from a form.
$firstname = 'Joyce';
$lastname = 'Park';
$title = 'Ms.';
echo "$title $lastname";
// OK, we are going back to HTML now.
?>
. We know who you are! Your first name is <?php echo
$firstname; ?>.</P>
<P>You are visiting our site at <?php echo date('Y-m-d H: -- i:s');
?></P>
<P>Here is a link to your account management page: <A
HREF="http://www.example.com/accounts/<?php echo
"$firstname$lastname"; ?>"/><?php echo $firstname; ?>'s account
management page</A></P>
</BODY>
</HTML>

```

When a client requests this page, the Web server *preprocesses* it. This means it goes through the page from top to bottom, looking for sections of PHP, which it will try to resolve. For one thing, the parser will suck up all assigned variables (marked by dollar signs) and try to plug them into later PHP commands (in this case, the `echo` function). If everything goes smoothly, the preprocessor will eventually return a normal HTML page to the client's browser, as shown in Figure 1-1.



Figure 1-1: A result of preprocessed PHP

If you peek at the source code from the client browser (select Source or Page Source from the

View menu, or right-click if you're using the AOL browser), it will look like this:

```
<HEAD>
<TITLE>Example.com greeting</TITLE>
</HEAD>
<BODY>
<P>Hello,
Ms. Park
. We know who you are! Your first name is Joyce.</P>
<P>You are visiting our site at 2002-04-21 19-34-24</P>
<P>Here is a link to your account management page: <A
HREF="http://www.example.com/accounts/JoycePark">Joyce's account
management page</A></P>
</BODY>
</HTML>
```

This code is exactly the same as if you were to write the HTML by hand. So simple!

The HTML-embeddedness of PHP has many helpful consequences:

- PHP can quickly be added to code produced by WYSIWYG editors.

- PHP lends itself to a division of labor between designers and scripters.

- Every line of HTML does not need to be rewritten in a programming language.

- PHP can reduce labor costs and increase efficiency due to its shallow learning curve and ease of use.

Perhaps the sweetest thing of all about embedded scripting languages is that they don't need to be compiled into binary code before they can be tested or used—just write and run. PHP is interpreted (as are many newish computer languages), although the Zend Engine does some behind-the-scenes precompiling into an intermediate form for greater speed with complex scripts.

But what if you happen to want compilation? This can be desirable if you wish to distribute nonreversible binaries so others can use the code without being able to look at the source.

The Zend team now offers a precompiler, *Zend Encoder*, which will deliver the code in a nonreversible intermediate representation, as well as substantially speed up large complex PHP scripts.

Cross-platform compatibility

PHP and MySQL run native on every popular flavor of Unix (including Mac OS X) and Windows. A huge percentage of the world's HTTP servers run on one of these two classes of operating systems.

PHP is compatible with the three leading Web servers: Apache HTTP Server for Unix and Windows, Microsoft Internet Information Server, and Netscape Enterprise Server (a.k.a. iPlanet Server). It also works with several lesser-known servers, including Alex Belits' fhttpd, Microsoft's Personal Web Server, AOLServer, and Omnicentrix's

Omniserver application server. Specific Web-server compatibility with MySQL is not required, since PHP will handle all the dirty work for you.

Table 1-2 shows a brief matrix of the possible OS/Web-server combinations.

Table 1-2: Operating Systems and Web Servers for PHP **Variables UNIX Windows**

Flavors AIX, A/UX, BSDI, Digital UNIX/Tru64, Windows 95/98/ME
FreeBSD, HP-UX, IRIX, Linux, Mac OS X, Windows NT/2000/XP/2003
NetBSD, OpenBSD, SCO UnixWare, Solaris, SunOS, Ultrix, Xenix, and more
Web servers Apache, fhttpd, Netscape IIS, PWS, Netscape, Apache, Omni
Now that PHP runs on Macintosh, PHP is almost totally cross-platform. You can develop on almost any client OS using your favorite tools and then upload your PHP scripts to a server on almost any OS. We'll discuss the development process in more detail in Chapter 3.

Not tag-based

PHP is a real programming language. ColdFusion, by contrast, is a bunch of predefined tags, like HTML. In PHP, you can define functions to your heart's content just by typing a name and a definition. In ColdFusion, you have to use tags developed by other people or go through the Custom Tag Extension development process.

As a witty PHP community member once said, "ColdFusion makes easy things easy, and medium-hard things impossible." And as every programmer will agree, once you experience the power of curly brackets and loops, you never go back to tags.

Stability

The word *stable* means two different things in this context:

The server doesn't need to be rebooted often.

The software doesn't change radically and incompatibly from release to release.

To our advantage, both of these connotations apply to both MySQL and PHP.

Apache Server is generally considered the most stable of major Web servers, with a reputation for enviable uptime percentages. Although it is not the fastest nor the easiest to administer, once you get it set up, Apache HTTP Server seemingly never crashes. It also doesn't require server reboots every time a setting is changed (at least on the Unix side). PHP inherits this reliability; plus, its own implementation is solid yet lightweight. In a two-and-a-half-month head-to-head test conducted by the Network Computing labs in October 1999, Apache Server with PHP handily beat both IIS/Visual Studio and Netscape Enterprise Server/Java for stability of environment.

PHP and MySQL are also both stable in the sense of feature stability. Their respective development teams have thus far enjoyed a clear vision of their project and refused to be distracted by every new fad and ill-thought-out user demand that comes along. Much of

the effort goes into incremental performance improvements, communicating with more major databases, or adding better session support. In the case of MySQL, the addition of reasonable and expected new features has hit a rapid clip. For both PHP and MySQL, such improvements have rarely come at the expense of compatibility. Applications written in PHP3 will function with little or no revision for PHP4 and 5. And because of the standards-based SQL support, MySQL 3.x databases are easily moved to more current versions (and most likely always will be).

Speed

PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.

PHP5 is much faster for almost every use than CGI scripts. There is an unfortunate grain of truth to the joke that CGI stands for “Can’t Go Instantly.” Although many CGI scripts are written in C, one of the lowest-level and therefore speediest of the major programming languages, they are hindered by the fact that each request must spawn an entirely new process after being handed off from the http daemon. The time and resources necessary for this handoff and spawning are considerable, and there can be limits to the number of concurrent processes that can be running at any one time. Other CGI scripting languages such as Perl and Tcl can be quite slow. Most Web sites have moved away from use of CGI for performance and security reasons.

Although it takes a slight performance hit by being interpreted rather than compiled, this is far outweighed by the benefits PHP derives from its status as a Web server module. When compiled this way, PHP becomes part of the http daemon itself. Because there is no transfer to and from a separate application server (as there is with ColdFusion, for instance) requests can be filled with maximum efficiency.

Although no extensive formal benchmarks have compared the two, much anecdotal evidence and many small benchmarks suggest that PHP is at least as fast as ASP and readily outperforms ColdFusion or JSP in most applications.

Open source licensing

We’ve already dealt with the cost advantages of open source software in the “Cost” section of this chapter. The other major consequence of these licenses is that the complete source code for the software must be included in any distribution.

In fact, the Unix version of PHP is released *only* as source code; so far, the development team has staunchly resisted countless pleas to distribute official binaries for any of the Unixes. At first, new users (particularly those also new to Unix) tend to feel that source code is about as useful as a third leg, and most vastly prefer a nice convenient rpm. But there are both pragmatic and idealistic reasons for including folders full of pesky .c and .h files. The most immediate pragmatic advantage is that you can compile your PHP installation with only the stuff you really need for any given situation. This approach has performance and security advantages. For instance, you can put in hooks to the database(s) of your choice.

You can recompile as often as you want: maybe when an Apache security release comes out, or when you wish to support a new database application. By compiling a custom application specifically suited to your system, or any given snapshot of your system, performance and stability are increased over their already respectable baseline.

What sets open source software apart from its competitors is not just price but control. Plenty of consumer software is now given away under various conditions. Careful scrutiny of the relevant licenses, however, will generally reveal limits as to how the software can be used.

Maybe you can run it at home but not at the office. Perhaps you can load it on your laptop, but you're in violation if you use it for business purposes. Or, most commonly, you can use it for anything you want but forget about looking at the code—much less changing it. There are even community licenses that force you to donate your improvements to the codebase but charge you for use of the product at the end!

Don't even *think* about coming back with a riposte that involves violating a software license—we're covering our ears; we're *not* listening! Especially with the explosion in no-cost software, there's just no good reason to break the law. Besides, it's bad karma for software developers. What goes around, comes around, don't ya know?

For all their *openness*, the licenses for MySQL and PHP are quite different. You should not assume that you understand the MySQL terms simply because you have read the PHP license. They have many similarities to be sure but also some radically different provisions, especially when it comes to when you should pay. Table 1-3 shows examples of the various source and fee positions in today's software marketplace.

Table 1-3: Source/Fee Spectrum

Fee Structure Closed Source Controlled Source Open Source

Fee for all uses Macromedia ColdFusion — —

Fee for some uses Corel WordPerfect Sun Java MySQL

No fee for any use Microsoft IE Sun StarOffice GPLed software

Caution

Genuinely open source software like PHP cannot seek to limit the purposes for which it is used, the people allowed to use it, or a host of other factors. The most critical of these rights is the one allowing users to make and distribute any modifications along with the original software. In the most extreme case, where one or more developers decide to release a separate, complete version of a piece of software, this practice is referred to as *code forking*. If somewhere down the road you develop irreconcilable differences with the PHP development team, you can take every bit of code they've labored over for all these years and use it as the basis of your own product. You couldn't call it PHP, and you'd have to include stuff in your documentation that gave due credit to the authors—the rationale is that source code distributions make it next to impossible for any single person or group to hijack a program to the detriment of the community as a whole, because every user always has the power to take the source and walk.

Users new to the open source model should be aware that this right is also enjoyed by the developers. At any time, Rasmus, Zend, and company can choose to defect from the community and put all their future efforts into a commercial or competing product based

on PHP. Of course, the codebase up to this point would still be available to anyone who wanted to pick up the baton, and for a product as large as PHP that could be a considerable number of volunteer developers.

This leads to one other oft-forgotten advantage of open source software: You can be pretty sure the software will be around in a few years, no matter what. In these days of products with the life spans of morning glories, it's hard to pick a tool with staying power. Fans of OS/2, Amiga, NeXT, Newton, Firefly, Netscape, BeOS, Napster, and a host of other once-hot technologies know the pain of abandonment when a company goes belly-up, decides to stop supporting a technology, or is sold to a buyer with a new agenda. The open source model reduces the chances of an ugly emergency port in a couple of years and thus makes long-term planning more realistic.

Many extensions

PHP makes it easy to communicate with other programs and protocols. The PHP development team seems committed to providing maximum flexibility to the largest number of users.

Database connectivity is especially strong, with native-driver support for about 15 of the most popular databases plus ODBC. In addition, PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time. PHP5 extends this support even further, offering a fully incorporated GD graphics library and revamped XML support with DOM and simpleXML.

Most things that PHP does not support are ultimately attributable to closed-source shops on the other end. For instance, Microsoft has not thus far been eager to cooperate with open source projects like PHP. Potential users who complain about lack of native Mac OS 9 or .NET support on the PHP mailing list are simply misinformed about where the fault lies.

Fast feature development

Users of proprietary Web development technologies can sometimes be frustrated by the glacial speed at which new features are added to the official product standard to support emerging technologies. With PHP, this is not a problem. All it takes is one developer, a C compiler, and a dream to add important new functionality. This is not to say that the PHP team will accept every random contribution into the official distribution without community buy-in, but independent developers can and do distribute their own extensions which may be later folded into the main PHP package in more or less unitary form. For instance, Dan Libby's elegant xmlrpc-epi extension was adopted as part of the PHP distribution in version 4.1, a few months after it was first released as an independent package.

PHP development is also constant and ongoing. Although there are clearly major inflection points, such as the transition between PHP4 and PHP5, these tend to be most important deep in the guts of the parser—people were actually working on major extensions throughout the transition period without critical problems. Furthermore, the PHP group subscribes to the open source philosophy of “release early, release often,”

which gives developers many opportunities to follow along with changes and report bugs. Compare this release scheme to the .NET transition, which has left developers with almost a year in which Microsoft is not really improving IIS but has not yet released a prime-time version of .NET server.

It hasn't always been the case that MySQL added new features in a timely fashion. It would probably be fair to say that a significant chunk of PostgreSQL users are former MySQL users frustrated by the lack of transaction support, for example. However, the 4.0 and 4.1 versions have remedied this and other inequities. Transactions are in the software today, while subselects and foreign keys are experimental but coming along nicely.

Popularity

PHP is fast becoming one of the most popular choices for so-called two-tier development (Web plus data). Figure 1-2 charts growth since 1999.

Figure 1-2: Netcraft survey of PHP use

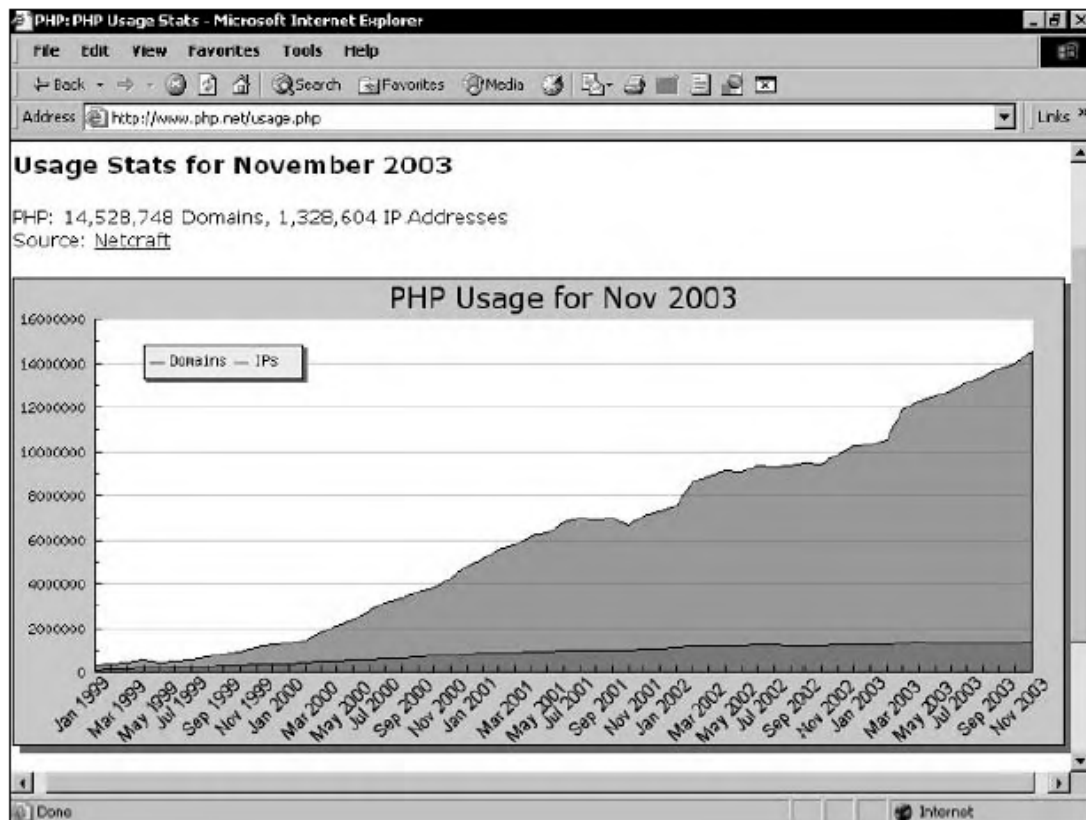


Figure 1-2: Netcraft survey of PHP use

Although it's not evident from this graphic, the period October 1998 through October 1999 showed 800 percent growth in the number of domains. As Web sites become even more ubiquitous, and as more of them go beyond simple static HTML pages, PHP is expected to gain ground quickly in absolute numbers of users.

Although it's somewhat more difficult to get firm figures, it seems that PHP is also in a strong position relative to similar products. According to a 2002 Zend report, Microsoft

Active Server Pages technology appears to be utilized on about 24 percent of Web servers, whereas ColdFusion is implemented on approximately 4 percent of surveyed domains. PHP is used on over 24 percent of all Web servers, as measured by a larger and more accurate sample, and is now said to be the most popular server-side scripting language on the Web. Active Server Pages and ColdFusion used to be highly visible because they tended to be disproportionately selected by large e-commerce sites. However, the realities of the Web finally caught up with us—and it is the flashy e-commerce sites that were disproportionately thinned by the dot-bomb crash. It is now becoming clearer that most Web sites are informational rather than direct revenue centers and, therefore, do not repay high development expenses in an immediate way. PHP enjoys substantial advantages over its competitors in this development category, which has turned out to be the majority of the Internet.

Not proprietary

The history of the personal computer industry to date has largely been a chronicle of proprietary standards: attempts to establish them, clashes between them, their benefits and drawbacks for the consumer, and how they are eventually replaced with new standards.

But in the past few years the Internet has demonstrated the great convenience of voluntary, standards-based, platform-independent compatibility. E-mail, for example, works so well because it enjoys a clear, firm standard to which every program on every platform must conform. New developments that break with the standard (for example, HTML-based e-mail stationery) are generally regarded as deviations, and their users find themselves having to bear the burdens of early adoption.

Furthermore, customers (especially the big-fish businesses with large systems) are fed up with spending vast sums to conform to a proprietary standard—only to have the market uptake not turn out as promised. Much of the current momentum toward XML and Web services is driven by years of customer disappointment with Java RMI, CORBA, COM, and even older proprietary methods and data formats.

Right now, software developers are in a period of experimentation and flux concerning proprietary versus open standards. Companies want to be sure they can maintain profitability while adopting open standards. There have been some major legal conflicts related to proprietary standards, which are still being resolved. These could eventually result in mandated changes to the codebase itself or even affect the futures of the companies involved. In the face of all this uncertainty, a growing number of businesses are attracted to solutions that they know will not have these problems in the foreseeable future.

PHP is in a position of maximum flexibility because it is, so to speak, *antiproprietary*. It is not tied to any one server operating system, unlike Active Server Pages. It is not tied to any proprietary cross-platform standard or middleware, as Java Server Pages or ColdFusion are. It is not tied to any one browser or implementation of a programming language or database. PHP isn't even doctrinaire about working only with other open source software. This independent but cooperative pragmatism should help PHP ride out the stormy seas that seem to lie ahead.

Strong user communities

PHP is developed and supported in a collaborative fashion by a worldwide community of users. Some animals (such as the core developers) are more equal than others—but that's hard to argue with, because they put in the most work, had the best ideas, and have managed to maintain civil relationships with the greatest number of other users.

The main advantage for most new users is technical support without charge, without boundaries, and without the runaround. People on the mailing list are available 24/7/365 to answer your questions, help debug your code, and listen to your gripes. The support is human and real. PHP community members might tell you to read the manual, take your question over to the appropriate database mailing list, or just stop your whining—but they'll never tell you to wipe your C drive and then charge you for the privilege. Often, they'll look at your code and tell you what you're doing wrong or even help you design an application from the ground up.

As you become more comfortable with PHP, you may wish to contribute. Bug tracking, offering advice to others on the mailing lists, posting scripts to public repositories, editing documentation, and, of course, writing C code are all ways you can give back to the community.

MySQL, while open-source licensed for nonredistributive uses, is somewhat less community driven in terms of its development. Nevertheless, it benefits from a growing community of users who are actively listened to by the development team. Rarely has a software project responded so vigorously to community demand. And the community of users can be extremely responsive to other users who need help. It's a point of pride with a lot of SQL gurus that they can write the complicated queries that get you the results you are looking for but had struggled with for days. In many cases, they'll help you for nothing more than the enduring, if small, fame that comes with the archived presence of their name on Google Groups. Try comparing *that* with \$100 per incident support.

Summary

PHP and MySQL, individually or together, aren't the panacea for every Web development problem, but they present a lot of advantages. PHP is built by Web developers for Web developer sand supported by a large and enthusiastic community. MySQL is a powerful standards compliant RDBMS that comes in at an extremely competitive price point, even more so if you qualify for free use. Both technologies are clear-cut cases of the community banding together to address its own needs.

Πηγές-Βιβλιογραφία

Βιβλιογραφία

- Julie C. Melony «PHP, MySql και Apache» Εκδόσεις Μ.Γκιούρδας

Πηγές

CRM

- http://www.ast.gr/downloads/MSCRM3.0_Service_DataGR.pdf
- <http://www.interworks.gr/landingPage/landing.asp?source=google>
- <http://www.salesmanager.gr/default.asp?pid=14&la=1>
- http://www.goline.gr/training/enot3/kef8/math1/3_8_1_6.htm?PHPSESSID=f7de0000eab97d3d4835d7fe250b859d
- http://en.wikipedia.org/wiki/Customer_relationship_management

PHP

- <http://www.php.net/>
- <http://myphp.gr/manual/el>
- <http://www.freestuff.gr/forums/viewtopic.php?t=18943>
- <http://gr.php.net/history/>
- <http://www.techteam.gr/wiki/PHP>
- <http://en.wikipedia.org/wiki/Php>
- <http://www.freestuff.gr/tutorials/f/27/PHP+%D0%F1%EF%E3%F1%E1%EC%EC%E1%F4%E9%F3%EC%FC%F2>
- <http://www.freestuff.gr/forums/viewtopic.php?t=21531>
- <http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-Php-Analytical.html>

SQL

- <http://dev.mysql.com/doc/refman/5.1/en/history.html>
- <http://www.mysql.com/why-mysql/>
- <http://www.clearlearning.com/v4/install/mysql.html>
- <http://en.wikipedia.org/wiki/Mysql>

APACHE

- http://en.wikipedia.org/wiki/Apache_HTTP_Server
- <http://httpd.apache.org/>

PhpMyAdmin

- <http://en.wikipedia.org/wiki/PhpMyAdmin>

