

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΑΤΡΩΝ



**ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ
ΤΜΗΜΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ ΣΧΕΔΙΑΣΜΟΥ & ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΣΧΕΔΙΑΣΜΟΣ & ΥΛΟΠΟΙΗΣΗ ΕΝΟΣ
ΗΛΕΚΤΡΟΝΙΚΟΥ ΔΥΝΑΜΙΚΟΥ
ΤΟΥΡΙΣΤΙΚΟΥ ΟΔΗΓΟΥ**

**ΓΙΑΝΝΟΠΟΥΛΟΣ ΣΠΥΡΙΔΩΝ Α.Μ. 1831
ΤΡΟΥΠΚΟΣ ΔΗΜΗΤΡΙΟΣ Α.Μ.1786
ΓΚΕΚΑΣ ΝΙΚΟΛΑΟΣ Α.Μ.1911**



**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ
ΠΑΠΑΪΩΑΝΝΟΥ ΒΑΙΟΣ**

ΠΑΤΡΑ 2012

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ	5
ΚΕΦΑΛΑΙΟ 1	9
ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ	9
1.1 ΕΙΣΑΓΩΓΗ	9
1.2 ΤΑ ΔΕΔΟΜΕΝΑ ΚΑΙ ΟΙ ΠΛΗΡΟΦΟΡΙΕΣ	10
1.3 ΟΙ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ	10
1.4 ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ	12
1.4.1 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΧΡΗΣΗΣ ΤΩΝ ΣΔΒΔ	13
1.5 ΜΟΝΤΕΛΑ ΔΕΔΟΜΕΝΩΝ	14
ΚΕΦΑΛΑΙΟ 2	16
INTERNET WEB SERVERS	16
2.1 ΙΣΤΟΡΙΑ	16
2.2 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ TCP/IP	17
2.3 ΜΟΝΤΕΛΟ OSI	18
2.3.1 APPLICATION LAYER:	19
2.3.2 PRESENTATION LAYER:	19
2.3.3 SESSION LAYER:	20
2.3.4 TRANSPORT LAYER:	20
2.3.5 NETWORK LAYER:	20
2.3.6 DATA LINK LAYER:	21
2.3.7 PHYSICAL LAYER:	21
2.4 ΑΡΧΙΤΕΚΤΟΝΙΚΗ TCP/IP	21
2.5 ΔΙΑΔΙΚΤΥΩΣΗ	24
2.6 IP ΔΙΕΥΘΥΝΣΗ	24
2.7 ΥΠΟΔΙΚΤΥΑ (SUBNETS)	26
2.8 ΔΡΟΜΟΛΟΓΗΣΗ (ROUTING)	27
2.9 CLIENTS ΚΑΙ SERVERS	29
2.10 APACHE	30
2.10.1 ΛΕΙΤΟΥΡΓΙΑ	31
ΚΕΦΑΛΑΙΟ 3	32
MYSQL (STRUCTURED QUERY LANGUAGE)	32
3.1 ΕΙΣΑΓΩΓΗ	32
3.2 Η MYSQL ΩΣ ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ	33

3.3 ΟΙ ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ ΤΗΣ MYSQL.....	33
3.3.1 ΟΙ ΑΡΙΘΜΗΤΙΚΟΙ ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ.....	34
3.3.3 ΟΙ ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ.....	36
3.4 ΟΙ ΕΝΤΟΛΕΣ ΤΗΣ MYSQL.....	37
3.4.1 Η ΕΝΤΟΛΗ ΔΗΜΙΟΥΡΓΙΑΣ ΠΙΝΑΚΩΝ.....	37
3.4.2 Η ΕΝΤΟΛΗ SELECT.....	39
3.4.3 Η ΕΝΤΟΛΗ UPDATE.....	41
3.4.4 Η ΕΝΤΟΛΗ JOIN.....	41
3.4.5 Η ΕΝΤΟΛΗ REPLACE.....	43
3.4.6 Η ΕΝΤΟΛΗ DELETE.....	43
ΚΕΦΑΛΑΙΟ 4.....	44
PHP.....	44
4.1 ΕΙΣΑΓΩΓΗ.....	44
4.2 ΣΥΝΤΟΜΟ ΙΣΤΟΡΙΚΟ ΤΗΣ PHP.....	44
4.3 ΟΙ ΕΝΤΟΛΕΣ ΤΗΣ PHP.....	45
4.3.1 ΜΕΤΑΒΛΗΤΕΣ.....	45
4.3.2 ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ ΤΗΣ PHP.....	46
4.3.3 ΤΕΛΕΣΤΕΣ ΚΑΙ ΕΚΦΡΑΣΕΙΣ ΤΗΣ PHP.....	46
4.3.4 ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΤΩΝ ΤΕΛΕΣΤΩΝ.....	48
4.3.5 ΣΤΑΘΕΡΕΣ.....	49
ΚΕΦΑΛΑΙΟ 5.....	51
ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ.....	51
5.1 ΤΙ ΕΙΝΑΙ ΕΝΑ ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ.....	51
5.2 ΤΙ ΕΙΝΑΙ ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ WEB.....	51
5.3 ΚΑΤΗΓΟΡΙΕΣ ΣΥΣΤΗΜΑΤΩΝ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ WEB.....	53
5.3.1 ΕΠΕΞΕΡΓΑΣΙΑ OFFLINE.....	53
5.3.2 ΕΠΕΞΕΡΓΑΣΙΑ ONLINE.....	54
5.3.3 ΥΒΡΙΔΙΚΑ ΣΥΣΤΗΜΑΤΑ.....	54
5.4 ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ ΠΟΥ ΥΠΑΡΧΟΥΝ.....	55
5.4.1 WORDPRESS.....	55
5.4.2 JOOMLA.....	57
5.4.3 DRUPAL.....	58
5.4.4 DOTNETNUKE (DNN).....	61
5.4.5 MOODLE.....	62
5.4.6 EZPUBLISH.....	64

ΚΕΦΑΛΑΙΟ 6	65
ΕΙΣΑΓΩΓΗ ΣΤΟ DREAMWEAVER MX	65
6.1 ΕΙΣΑΓΩΓΗ	65
6.2 ΤΟ DREAMWEAVER MX ΚΑΙ Η PHP	67
6.3 ΞΕΚΙΝΗΜΑ ΜΕ ΤΟ DREAMWEAVER MX	68
6.4 Ο ΒΑΣΙΚΟΣ ΟΔΗΓΟΣ ΓΙΑ ΤΟΝ ΟΡΙΣΜΟ ΤΟΥ SITE	68
6.5 Ο ΠΡΟΗΓΜΕΝΟΣ ΟΡΙΣΜΟΣ SITE	70
6.6 ΔΗΜΙΟΥΡΓΙΑ ΤΗΣ ΣΥΝΔΕΣΗΣ ΜΕ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ	71
6.7 Ο ΠΙΝΑΚΑΣ GUESTBOOK	72
6.8 Η ΙΣΤΟΣΕΛΙΔΑ ADD_USER_RECORD.PHP	72
6.9 Η ΣΕΛΙΔΑ ΜΑΣ	76
ΣΥΜΠΕΡΑΣΜΑΤΑ	80
ΒΙΒΛΙΟΓΡΑΦΙΑ	81

ΕΙΣΑΓΩΓΗ

Το Διαδίκτυο θα μπορούσε εύκολα να διεκδικήσει τον τίτλο ενός από τα θαύματα του σύγχρονου κόσμου. Πρόκειται για ένα μέσο που δίνει σε πραγματικό χρόνο πρόσβαση σε πληροφορία, γνώση αλλά και ψυχαγωγία. Οι ταχύτητες συνεχώς αυξάνονται κάνοντας την πληροφορία ακόμη πιο προσβάσιμη, ενώ και τα δίκτυα που χρησιμοποιούνται εξελίσσονται με γοργούς ρυθμούς ώστε η πληροφορία αυτή να είναι διαθέσιμη σε μεγαλύτερα τμήματα του πληθυσμού. Το πάτημα και μόνο του κουμπιού λειτουργίας ενός υπολογιστή συνδεδεμένου στο Διαδίκτυο, τον φέρνει σε άμεση σχεδόν επαφή με άλλους υπολογιστές, servers, υπηρεσίες, που θα ζητήσουν να λάβουν πληροφορίες για τα προγράμματα, το λειτουργικό ή θα θελήσουν να προμηθεύσουν στον υπολογιστή τις τελευταίες διαθέσιμες ενημερώσεις, μέσα από αυτοματοποιημένες διαδικασίες που ο χρήστης μπορεί να μην καταλάβει καν.

Στην γενική του έννοια, διαδίκτυο είναι ένα δίκτυο ηλεκτρονικών υπολογιστών που (δια)συνδέει άλλα δίκτυα. Ο αντίστοιχος αγγλικός όρος internet προκύπτει από τη σύνθεση λέξεων inter-network.

Η χρησιμοποίηση των ηλεκτρονικών υπολογιστών στις επιχειρήσεις έχει αυξηθεί σημαντικά σε σχέση με δέκα χρόνια πιο πριν.

Οι επιχειρηματίες χρησιμοποιούν υπολογιστές για να υποστηρίξουν τις λογιστικές λειτουργίες του καταστήματος, όπως για να υπολογίζουν το κέρδος αλλά και την ζημία, να τυπώνουν την μισθοδοσία, να διαχειρίζονται την αποθήκη τους, να ελέγχουν την διάρθρωση της βάρδιας, την είσοδο και την έξοδο του προσωπικού, την λήψη και την εκτέλεση των παραγγελιών καθώς και την χρέωση πιστωτικών καρτών.

Όλοι όσοι ενσωματώνουν τους υπολογιστές στη λειτουργία τους φροντίζουν να προσλαμβάνουν είτε μέσα στην επιχείρηση, είτε ως εξωτερικοί συνεργάτες εξειδικευμένο προσωπικό για να τους υποστηρίξει σε τυχόν προβλήματα ή αναβαθμίσεις.

Ο σημερινός καταναλωτής είναι πλέον πολύ απαιτητικός και ξοδεύει αρκετό χρόνο ψάχνοντας απαντήσεις σε οτιδήποτε τον απασχολεί στο διαδίκτυο, οι επιχειρηματίες φτιάχνουνε διαδραστικά δυναμικά site ή καλύτερα ιστότοπους, δίνοντας τη δυνατότητα στους καταναλωτές να δούνε αναλυτικά το μενού και τις τιμές του σημείου, να δούνε τους χώρους αυτού, να μάθουν τις ειδικές προσφορές της επιχείρησης, να γραφτούν για την αποστολή email από την επιχείρηση, να μάθουνε διάφορα νέα και άλλα.

Κι όλα αυτά, μπορούν πια μόνοι τους οι επιχειρηματίες ή οι συνεργάτες τους να τα αλλάξουν στον ιστότοπο όταν προκύπτει η ανάγκη, να ανανεώσουν τις πληροφορίες που περιλαμβάνει το site ή να προσθέσουν καινούρια στοιχεία που θα κάνουν το σημείο ακόμα πιο ελκυστικό.

Είναι πολλοί επίσης οι επιχειρηματίες που χρησιμοποιούν το internet για να βρουν πληροφορίες για διάφορα προϊόντα πρώτες ύλες, εργαλεία, σκεύη, μηχανήματα, κτλ, για νέα του κλάδου, για τον καιρό ή για έρευνα σε νέες ή εναλλακτικές μεθόδους παραγωγής. Επιπλέον είναι πολλοί αυτοί που προσπαθούν να προσελκύσουν μέσω του ιστότοπου που αντιπροσωπεύει την επιχείρησή τους υποψήφιους εργαζόμενους προτρέποντας τους να συμπληρώσουν online αιτήσεις για εργασία.

Τα ηλεκτρονικά καταστήματα συγκαταλέγονται στις εφαρμογές της κατηγορίας επιχείρηση προς καταναλωτή και αποτελούν σημεία προώθησης και πώλησης αγαθών μέσω του Internet. Ένα Ηλεκτρονικό Κατάστημα μπορεί να αναπαραστήσει και να εμπλουτίσει με νέες δυνατότητες τη λειτουργικότητα ενός καταστήματος. Σε αρκετές περιπτώσεις, τα ηλεκτρονικά καταστήματα λειτουργούν ως υποκαταστήματα των υφισταμένων (φυσικών) καταστημάτων, ενώ σε άλλες περιπτώσεις αποτελούν τα μοναδικά σημεία πώλησης των προϊόντων μιας επιχείρησης, στόχος ενός ηλεκτρονικού καταστήματος δεν είναι μόνο η προβολή των προϊόντων, αλλά και η εφαρμογή τεχνικών που προσδίδουν πρόσθετη αξία στην ηλεκτρονική παρουσία του εμπόρου και ωθούν τον πελάτη να προτιμήσει την ηλεκτρονική αγορά έναντι της παραδοσιακής. Οι τεχνικές αυτές αφορούν την αποτελεσματική ανταπόκριση στις ανάγκες του καταναλωτή μέσα από την παροχή υποστήριξης του πελάτη (πριν και μετά την πώληση). Επιπλέον, θα πρέπει να ληφθεί υπόψη και ο σχεδιασμός, η παραγωγή και η προώθηση προϊόντων, βάσει των ιδιαίτερων απαιτήσεων και αναγκών τον κάθε πελάτη, μεμονωμένα, καθώς και η δυνατότητα εξοικονόμησης κόστους και χρόνου.

Από την πλευρά του εμπόρου, εξασφαλίζεται η ηλεκτρονική του παρουσία σε παγκόσμιο επίπεδο καθώς και η δημιουργία νέων καναλιών προώθησης και πώλησης των προϊόντων. Επιπλέον, προκύπτουν νέα, κανάλια επικοινωνίας με τους πελάτες αλλά και με άλλες επιχειρήσεις, καθιστώντας ακόμα πιο ισχυρή και άμεση την επαφή της επιχείρησης μαζί τους. Αυτοματοποιημένες διαδικασίες, που αφορούν την διαχείριση αποθεμάτων και την διανομή (μεταφορά και παράδοση) των προϊόντων, μπορούν να υιοθετηθούν και να συντελέσουν στην εξοικονόμηση κόστους για την επιχείρηση, και κατ' επέκταση για τον ίδιο τον πελάτη.

Σε ένα μεγάλο ποσοστό τα υπάρχοντα ηλεκτρονικά καταστήματα στο Internet, διαθέτουν χαρακτηριστικά που αφορούν την προβολή και προώθηση των προϊόντων, την δυνατότητα άμεσης παραγγελίας και αγοράς αυτών είτε με αντικαταβολή είτε μέσω πιστωτικής κάρτας. Επίσης υιοθετούνται μέθοδοι για την αναγνώριση της ταυτότητας του πελάτη και σκιαγράφηση του προφίλ, όχι μόνο όσον αφορά τα προσωπικά του στοιχεία αλλά και τις αγοραστικές του συνήθειες. Με αυτό τον τρόπο κάθε φορά που ο πελάτης αυτός επισκέπτεται το κατάστημα, μπορεί να

δει κάποιες σελίδες διαμορφωμένες σύμφωνα με τις απαιτήσεις και την αγοραστική του συμπεριφορά κατά την τελευταία του επίσκεψη. Επιπλέον νέα προϊόντα και προσφορές που εμπίπτουν στις αγοραστικές του συνήθειες προτείνονται στους συχνότερους επισκέπτες.

ΚΕΦΑΛΑΙΟ 1

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

1.1 ΕΙΣΑΓΩΓΗ

Στις αρχές της δεκαετίας του 1960, ένας νέος όρος εμφανίστηκε στην βιβλιογραφία αλλά και στην πρακτική των ηλεκτρονικών υπολογιστών. Ο όρος ήταν «βάσεις δεδομένων» και αρχικά χρησιμοποιούταν από τους εργαζόμενους σε στρατιωτικές υπηρεσίες πληροφορικής, για να δηλώσουν συλλογές δεδομένων υπό τη διαχείριση μεγάλων υπολογιστικών συστημάτων διαμοιραζόμενου χρόνου. Η έννοια αυτή ήρθε να αντικαταστήσει τον όρο «ολοκληρωμένη επεξεργασία δεδομένων» (integrated data processing) που ήταν σε χρήση από την προηγούμενη δεκαετία. Σύντομα, ο όρος βάσεις δεδομένων διαδόθηκε σε όλο το εύρος των επιχειρήσεων και αρχικά υποδήλωνε τη συγκεντρωτική διαχείριση των δεδομένων ανεξάρτητων εφαρμογών.

Την εποχή εκείνη, υπήρχε στενή εξάρτηση της χρήσης μιας βάσης δεδομένων με τη δομή των δεδομένων και τις λεπτομέρειες της φυσικής τους αποθήκευσης. Ένα χρήστης έπρεπε να κατέχει σε σημαντικό βαθμό τις λεπτομέρειες αυτές. Μετά από λίγα χρόνια (1970) ο E.F. Codd στην σημαντικότερη εργασία του που θεμελιώνει τις σχεσιακές βάσεις δεδομένων σημείωνε: «Οι μελλοντικοί χρήστες των μεγάλων τραπεζών δεδομένων (data banks) πρέπει να προστατευτούν από το να πρέπει να ξέρουν τον τρόπο που τα δεδομένα είναι οργανωμένα στην μηχανή (εσωτερική αναπαράσταση)... Οι δραστηριότητες των χρηστών στα τερματικά και στα προγράμματα εφαρμογών πρέπει να παραμείνουν ανεπηρέαστες όταν η εσωτερική αναπαράσταση των δεδομένων αλλάξει αλλά και ακόμη όταν αλλάξουν κάποιες θεωρήσεις της εξωτερικής αναπαράστασής τους...».

Το σχεσιακό μοντέλο που εισήγαγε ο Codd, επρόκειτο να αλλάξει άμεσα και δραματικά τον τρόπο διαχείρισης και τη χρησιμότητα των βάσεων δεδομένων και να επαληθεύσει άμεσα τα λεγόμενά του. Οι σύγχρονες βάσεις δεδομένων συνεισέφεραν τα μέγιστα στην αποτελεσματική και χωρίς κόπο διαχείριση των δεδομένων μέσα σε κάθε επάγγελμα και έδωσαν το δικό τους στίγμα στην κοινωνία της πληροφορίας.

Στο εισαγωγικό αυτό κεφάλαιο παρουσιάζονται οι βασικές έννοιες και αρχές που χαρακτηρίζουν το αντικείμενο των βάσεων δεδομένων. Στην ενότητα 2, παρουσιάζονται οι βασικές έννοιες των δεδομένων και της πληροφορίας. Στην ενότητα 3, δίνεται ο ορισμός των βάσεων δεδομένων και επιχειρείται μια σύγκριση με την προ των βάσεων εποχή, από πλευράς διαχείρισης των δεδομένων. Στην ενότητα 4, επεξηγείται η έννοια και τα χαρακτηριστικά των συστημάτων διαχείρισης βάσεων

δεδομένων. Στην ενότητα 5, επεξηγείται η έννοια των μοντέλων δεδομένων και περιγράφονται τα κυριότερα μοντέλα που σχετίζονται με τις βάσεις δεδομένων.

1.2 ΤΑ ΔΕΔΟΜΕΝΑ ΚΑΙ ΟΙ ΠΛΗΡΟΦΟΡΙΕΣ

Στους υπολογιστές, οι έννοιες δεδομένα (data) και πληροφορία (information) πολλές φορές συγχέονται και χρησιμοποιούνται για τον ίδιο σκοπό. Αυτό είναι λάθος γιατί οι δυο έννοιες είναι διαφορετικές μεταξύ τους.

Τα δεδομένα είναι στοιχεία τυποποιημένα σε καθορισμένη μορφή και είναι κατάλληλα για επεξεργασία από ανθρώπους ή μηχανές. Τα δεδομένα που πρόκειται να αποθηκευτούν σε Η/Υ πρέπει να μετατραπούν σε κάποιον από τους γνωστούς κώδικες αναπαράστασης των υπολογιστών (π.χ. ASCII ή δυαδικός κώδικας). Με βάση τα παραπάνω, το σύνολο των λέξεων {Νίκος, Γιώργος, Παντελής, Κώστας} και το σύνολο των αριθμών {7, 3, 10, 9} είναι δεδομένα.

Η επεξεργασία των δεδομένων δημιουργεί την πληροφορία. Για την εξαγωγή της πληροφορίας είναι απαραίτητη η συγκεκριμένη γνώση των δεδομένων, π.χ το πεδίο αναφοράς τους. Για παράδειγμα, αν γνωρίζουμε πως το σύνολο δεδομένων {Νίκος, Γιώργος, Παντελής, Κώστας} αναπαριστά ονόματα σπουδαστών και το σύνολο {7, 3, 10, 9} τους αντίστοιχους βαθμούς τους σε ένα μάθημα, τότε η επεξεργασία τους μπορεί να δώσει πληροφορίες της μορφής:

«Ο Παντελής είναι ο καλύτερος όλων στο μάθημα»

«Ο Κώστας βαθμολογήθηκε με άριστα»

«Ο μέσος όρος των σπουδαστών στο μάθημα είναι 7,25».

Τα δεδομένα και οι πληροφορίες είναι δομικά στοιχεία για τα συστήματα των βάσεων δεδομένων που θα επεξηγήσουμε στη συνέχεια.

1.3 ΟΙ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

Ένα σημαντικό γνώρισμα των δεδομένων είναι πως μπορούν να κωδικοποιηθούν και να αποθηκευτούν στους Η/Υ.

Βάση Δεδομένων (ΒΔ) είναι μια διαμοιραζόμενη συλλογή από λογικά σχετιζόμενα δεδομένα μαζί με την περιγραφή τους, που είναι σχεδιασμένα να ικανοποιούν τις πληροφοριακές ανάγκες ενός οργανισμού. Οι βάσεις δεδομένων επομένως, προσφέρουν την οργάνωση και αποθήκευση των δεδομένων στον Η/Υ, ώστε να είναι δυνατή η επεξεργασία τους και η εξαγωγή της επιθυμητής πληροφορίας. Η τεχνολογία των βάσεων δεδομένων βρίσκει σημαντικότερες εφαρμογές σε όλες τις περιοχές που χρησιμοποιούνται οι υπολογιστές όπως στις

επιχειρήσεις, στην εκπαίδευση, στη διοίκηση, στην οικονομία, στην ιατρική και στα νομικά.

Εν συντομία μια βάση δεδομένων έχει τα εξής χαρακτηριστικά: Π Αναπαριστά μια άποψη του πραγματικού κόσμου του σχετιζόμενου με μια εφαρμογή, που πολλές φορές καλούμε μικρόκοσμο ή πεδίο αναφοράς. Για παράδειγμα μια βάση δεδομένων που υποστηρίζει το λογιστήριο μιας εταιρείας αναφέρεται στο μικρόκοσμο (ή πεδίο αναφοράς) του λογιστηρίου. Κάθε αλλαγή που επέρχεται στο μικρόκοσμο πρέπει να "περνάει" και στη ΒΔ. Περιέχει δεδομένα που είναι σχετικά μεταξύ τους. Μια τυχαία συνάθροιση διαφόρων δεδομένων δε μπορεί να θεωρηθεί ως ΒΔ. Π

- Σχεδιάζεται και διατηρεί δεδομένα για κάποιο συγκεκριμένο σκοπό. Επομένως, προορίζεται για μια συγκεκριμένη ομάδα χρηστών και για κάποιες εφαρμογές, για τις οποίες οι χρήστες αυτοί ενδιαφέρονται.
- Περιέχει δεδομένα που περιγράφουν το μικρόκοσμο είτε κατά την τρέχουσα χρονική στιγμή είτε στην εξέλιξη της ιστορίας του. Στην πράξη, μια σύγχρονη επιχειρησιακή ΒΔ είναι μια μεγάλη οργανωμένη αποθήκη δεδομένων, η οποία έχει δημιουργηθεί και τίθεται υπό διαχείριση με ενιαίο τρόπο και μπορεί να χρησιμοποιείται συγχρόνως, από πολλούς χρήστες και τμήματα του οργανισμού.

Ενδιαφέρον παρουσιάζει η σύγκριση με τα παλαιότερα αντίστοιχα συστήματα λογισμικού. Η παλαιότερη προσέγγιση για τη διαχείριση των δεδομένων βασιζόταν στα συστήματα αρχείων (file based systems). Τα συστήματα αυτά υποστηρίζονταν από διάφορα προγράμματα εφαρμογών, το καθένα από τα οποία διαχειριζόταν τα δικά του δεδομένα και αρχεία. Τα προγράμματα εφαρμογών ήταν υλοποιημένα με μια παραδοσιακή γλώσσα προγραμματισμού (π.χ. PASCAL, COBOL). Τα δεδομένα οργανώνονταν σε ανεξάρτητα αρχεία, τα οποία δεν είχαν ενιαίο τρόπο ορισμού και οργάνωσης και δεν υπήρχε κεντρικός τρόπος διαχείρισης. Κατά συνέπεια, τα συστήματα αυτά πολλές φορές υπέφεραν από σοβαρούς περιορισμούς και μειονεκτήματα όπως:

- Διαχωρισμό και απομόνωση δεδομένων, στην περίπτωση που τα δεδομένα βρίσκονταν σε ξεχωριστά αρχεία, για τα οποία δεν είχε οριστεί επαρκής μηχανισμός αλληλεπίδρασης και διαμοιρασμού.
- Επανάληψη δεδομένων, όταν τα ίδια δεδομένα αποθηκεύονταν ανεξέλεγκτα σε διαφορετικές θέσεις της επιχείρησης (π.χ. σε διαφορετικά τμήματα μιας επιχείρησης). Το φαινόμενο αυτό είναι γνωστό ως πλεονασμός δεδομένων. Το κύριο αίτιο ήταν η μη ορθή διαχείριση αλλά και η απουσία ενός επαρκούς μηχανισμού διαμοιρασμού.
- Αλληλεξάρτηση δεδομένων και ανελαστικότητα στην τροποποίηση της δομής τους, που ήταν το αποτέλεσμα κυρίως του τρόπου υλοποίησης του συστήματος. Συγκεκριμένα, η φυσική δομή και αποθήκευση των αρχείων ορίζονταν στα προγράμματα εφαρμογών. Οποιαδήποτε τροποποίηση στη

δομή των δεδομένων ήταν εξαιρετικά δύσκολη διαδικασία και θα μπορούσε να επιφέρει αλυσιδωτά προβλήματα στο σύστημα.

- Ασύμβατες μορφές αρχείων, που κυρίως ήταν το αποτέλεσμα της χρήσης διαφορετικών γλωσσών υλοποίησης των προγραμμάτων εφαρμογών.
- Δυσκολία στην επέκταση και στην αναβάθμιση των συστημάτων, γεγονός που οφειλόταν στην απουσία τεκμηρίωσης αλλά και στην απουσία τυποποίησης και κοινών κανόνων υλοποίησης. Τα συστήματα εξαρτούνταν σε μεγάλο βαθμό από τον προγραμματιστή εφαρμογών και το προσωπικό του ταλέντο, που εκ των πραγμάτων έπαιξε μεγάλο ρόλο στην υλοποίησή τους.
- Άναρχη αύξηση των αρχείων και των προγραμμάτων, που όμως δεν συνοδεύονταν από αντίστοιχη εξέλιξη στη διαχείριση, στην τεκμηρίωση και στη συντήρησή τους, στην ασφάλεια προσπέλασης και στην προστασία από αστοχία του υλικού (hardware) και λογισμικού (software).

Με το πέρασμα του χρόνου, τα συστήματα που βασίζονταν στα αρχεία παρουσίασαν ιδιαίτερα έντονα και μη αντιμετωπίσιμα προβλήματα και έτσι παρουσιάστηκε επιτακτικά η ανάγκη να δημιουργηθεί μια νέα γενιά συστημάτων διαχείρισης δεδομένων. Το πέρασμα από τα παραδοσιακά συστήματα αρχείων στις σύγχρονες βάσεις δεδομένων, διευκόλυνε η ευρεία διάδοση των Συστημάτων Διαχείρισης Βάσεων Δεδομένων.

1.4 ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ) είναι το λογισμικό που επιτρέπει στους χρήστες να δημιουργήσουν, να συντηρήσουν και να προσπελάσουν με οργανωμένο τρόπο μια βάση δεδομένων. Ένα ΣΔΒΔ παρέχει τις εξής δυνατότητες:

- Την περιγραφή της βάσης δεδομένων, π.χ. τον προσδιορισμό των τύπων, των δομών και των περιορισμών των δεδομένων της βάσης.
- Την κατασκευή της βάσης δεδομένων, δηλ. τη διαδικασία αποθήκευσης των δεδομένων σε ένα αποθηκευτικό μέσο (π.χ δίσκο) που ελέγχεται από το ΣΔΒΔ.
- Το χειρισμό της βάσης δεδομένων, δηλ. την υποβολή ερωτημάτων (queries) προς τη βάση για ανάκτηση πληροφοριών ή τη διαγραφή και τροποποίηση των παλαιών δεδομένων και την εισαγωγή νέων δεδομένων στη βάση.

Επομένως είναι ξεκάθαρη η διαφορά μεταξύ ενός ΣΔΒΔ και μιας βάσης δεδομένων. Το ΣΔΒΔ είναι λογισμικό που διευκολύνει τους χρήστες να δημιουργήσουν και να χρησιμοποιήσουν βάσεις δεδομένων. Αντίθετα, η βάση δεδομένων δημιουργείται μέσω ενός ΣΔΒΔ και χειρίζεται δεδομένα που αποθηκεύονται στο υλικό του Η/Υ.

Αρκετοί συγγραφείς αναφέρονται επίσης στην έννοια των Συστημάτων Βάσεων Δεδομένων. Ένα Σύστημα Βάσεων Δεδομένων αποτελείται από ένα ΣΔΒΔ, τις βάσεις δεδομένων που έχουν δημιουργηθεί και τις οποίες διαχειρίζεται το ΣΔΒΔ και το φυσικό μέσο αποθήκευσης που περιέχει τα δεδομένα και τους ορισμούς τους.

1.4.1 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΧΡΗΣΗΣ ΤΩΝ ΣΔΒΔ

Ένα ΣΔΒΔ παρέχει στο χρήστη σημαντικότερες διευκολύνσεις στη δημιουργία, συντήρηση και χρήση μιας βάσης δεδομένων. Όπως έχουμε ήδη αναφέρει, τα ΣΔΒΔ έχουν συνεισφέρει στο μέγιστο βαθμό στην εξέλιξη των σύγχρονων βάσεων δεδομένων και παρουσιάζουν γενικευμένη και συχνότατη χρήση. Γενικά, τα ΣΔΒΔ προσφέρουν τα παρακάτω πλεονεκτήματα χρήσης:

> Ολοκληρωμένο σύστημα διεπαφής και επικοινωνίας

Μια βάση δεδομένων πρέπει αρχικά να σχεδιαστεί και μετά να υλοποιηθεί στον υπολογιστή. Ο σχεδιασμός και η υλοποίηση περιλαμβάνουν τον ορισμό του τύπου των δεδομένων, την εισαγωγή δεδομένων κ.λ.π. Τα ΣΔΒΔ προσφέρουν ένα ενιαίο και φιλικό περιβάλλον επικοινωνίας και διεπαφής (user interface), με το οποίο ο χρήστης μπορεί να ολοκληρώσει με ευκολία και συντομία το σχεδιασμό και την υλοποίηση. Το σύστημα διεπαφής υποστηρίζει επίσης τη δημιουργία των ερωτημάτων, των φορμών και των αναφορών και κατά κανόνα είναι γραφικό και ιδιαίτερα αποτελεσματικό.

> Έλεγχος πρόσβασης

Επειδή οι περισσότερες βάσεις δεδομένων χρησιμοποιούνται από πολλούς χρήστες με διαφορετικές γνώσεις, ανάγκες και δικαιώματα πρόσβασης, είναι απαραίτητη η δυνατότητα ορισμού διαφορετικών κατηγοριών χρηστών. Η κάθε κατηγορία μπορεί να έχει δικαίωμα να εκτελεί διαφορετικές λειτουργίες πάνω στη βάση δεδομένων, να έχει προσπέλαση σε διαφορετικά δεδομένα κ.λ.π. Επίσης το ΣΔΒΔ φροντίζει για την τήρηση των κανόνων ασφάλειας και ελέγχει την πρόσβαση στις διάφορες βάσεις δεδομένων.

> Προστασία από βλάβες υλικού

Πολλές φορές κατά την διάρκεια της λειτουργίας της βάσης δεδομένων συμβαίνουν διακοπές της λειτουργίας του Η/Υ που οφείλονται σε αστοχίες του υλικού και του λογισμικού των υπολογιστών. Το αποτέλεσμα είναι συχνά καταστροφικό και έχει ως συνέπεια την απώλεια των δεδομένων. Τα ΣΔΒΔ προσφέρουν τη δυνατότητα τήρησης αντιγράφων ασφαλείας και ανάνηψης ή επαναφοράς (backup and recovery) των δεδομένων μετά από βλάβες υλικού

ή λογισμικού. Ο μηχανισμός επαναφοράς εξασφαλίζει πως αν το σύστημα αποτύχει π.χ. στο μέσον μιας διαδικασίας τροποποίησης

δεδομένων, η βάση δεδομένων θα επανέλθει αυτόματα στο σημείο που βρισκόταν πριν αρχίσει η τροποποίηση.

> Έλεγχος ακεραιότητας δεδομένων

Η ακεραιότητα των δεδομένων εξασφαλίζεται με την τήρηση συγκεκριμένων κανόνων. Οι κανόνες ακεραιότητας μπορεί να είναι γενικοί και να ισχύουν σε κάθε περίπτωση, ή να εξαρτώνται και να ορίζονται διαφορετικά σε κάθε βάση δεδομένων. Στην δεύτερη περίπτωση είναι γνωστοί ως κανόνες ορθότητας. Οι κανόνες ορθότητας σε μια βάση δεδομένων σχετίζονται συνήθως με την οργάνωση των δεδομένων ή τις τιμές που μπορούν να λάβουν τα δεδομένα. Για παράδειγμα, σε μια βάση δεδομένων που περιέχει τους βαθμούς των σπουδαστών, μπορούμε να ορίσουμε πως οι τιμές των δεδομένων των βαθμών θα είναι υποχρεωτικά από 0 έως 10. Η εισαγωγή οποιαδήποτε άλλης τιμής (δηλ. μικρότερης του 0 ή μεγαλύτερης του 10) θα προκαλούσε αυτόματα ένα μήνυμα λάθος από το ΣΔΒΔ.

> Έλεγχος πλεονασμών, συμβατότητα δεδομένων

Τα ΣΔΒΔ προσφέρουν μηχανισμούς ελέγχου και αποφυγής του πλεονασμού των δεδομένων. Στην περίπτωση που ηθελημένα τα ίδια δεδομένα αποθηκεύονται πάνω από μια φορά στη βάση δεδομένων, μπορεί να προκύψει ένα σημαντικότερο πρόβλημα: η ασυμβατότητα των δεδομένων μετά από τροποποιήσεις. Η ασυμβατότητα αυτής της μορφής εμφανίζεται με διαφορετικές τιμές για τα ίδια δεδομένα και οφείλεται σε μη επαρκή μηχανισμό ενημέρωσης. Για παράδειγμα, υποθέστε πως ο αριθμός μητρώου ενός σπουδαστή είναι αποθηκευμένος σε τρία διαφορετικά σημεία (π.χ. σε τρεις πίνακες). Στην περίπτωση αλλαγής του αριθμού μητρώου του συγκεκριμένου σπουδαστή, θα πρέπει να τροποποιηθούν αντίστοιχα και οι τρεις πίνακες. Τα ΣΔΒΔ προσφέρουν τον έλεγχο των πλεονασμών και διατηρούν με αυτόματους μηχανισμούς τη συμβατότητα των δεδομένων.

1.5 ΜΟΝΤΕΛΑ ΔΕΔΟΜΕΝΩΝ

Ένα μοντέλο δεδομένων είναι μια συλλογή από έννοιες που περιγράφουν τον τρόπο με τον οποίο τα δεδομένα αναπαρίστανται, προσπελαύνονται και αποθηκεύονται. Το μοντέλο μιας βάσης δεδομένων περιγράφει τη δομή της βάσης (δηλαδή τους τύπους, τις συσχετίσεις και τους περιορισμούς για τα δεδομένα) και τον τρόπο με τον οποίο μπορεί να χρησιμοποιηθεί η βάση (δηλαδή τις βασικές πράξεις ανάκτησης και ενημέρωσης δεδομένων).

Τα μοντέλα δεδομένων κατηγοριοποιούνται με βάση τη μορφή των εννοιών που περιέχουν, για την περιγραφή των δεδομένων. Τα υψηλού επιπέδου ή εννοιολογικά μοντέλα περιέχουν ένα επίπεδο αφαίρεσης (abstraction) που προσεγγίζει τον τρόπο που αντιλαμβάνεται τα δεδομένα

ο απλός άνθρωπος. Τα χαμηλού επιπέδου ή φυσικά μοντέλα δεδομένων περιγράφουν τον τρόπο αποθήκευσης των δεδομένων στον υπολογιστή. Τέλος, τα λογικά ή παραστατικά μοντέλα δεδομένων περιέχουν έννοιες μεσαίου επιπέδου αφαίρεσης, έτσι ώστε να μη απέχουν εξαιρετικά πολύ ούτε από τον τρόπο που σκέφτεται ο χρήστης αλλά ούτε και από τον τρόπο που αποθηκεύονται στον υπολογιστή

Ένα ιδιαίτερα γνωστό εννοιολογικό μοντέλο είναι το μοντέλο οντοτήτων συσχετίσεων, το οποίο θα παρουσιάσουμε στο τέταρτο κεφάλαιο. Κυρίαρχες έννοιες του μοντέλου αυτού είναι η οντότητα, η συσχέτιση και το γνώρισμα που θα εξηγηθούν αναλυτικά στο συγκεκριμένο κεφάλαιο. Το μοντέλο οντοτήτων-συσχετίσεων χρησιμοποιείται πολύ συχνά ως πρώτο βήμα για την ανάλυση - σχεδίαση μιας βάσης δεδομένων.

Τα λογικά μοντέλα είναι γνωστά και ως μοντέλα υλοποίησης και υποστηρίζονται από τα εμπορικά ΣΔΒΔ. Είναι τα μοντέλα που χρησιμοποιούνται στην πράξη για να περιγράψουν τις βάσεις δεδομένων και ιεραρχικά είναι ένα επίπεδο χαμηλότερα από τα κανονικά μοντέλα. Περιλαμβάνουν μεταξύ των άλλων το ιεραρχικό, το δικτυωτό και το σχεσιακό μοντέλο που θα αναφερθούν εν συντομία στη συνέχεια.

ΚΕΦΑΛΑΙΟ 2

INTERNET WEB SERVERS

2.1 ΙΣΤΟΡΙΑ

Το *internet*, το μεγαλύτερο παγκόσμιο δίκτυο ξεκίνησε ως ένα ερευνητικό πρόγραμμα του Υπουργείου Άμυνας των ΗΠΑ (*DARPA – Defense Advanced Research Project Agency*) για να συνδέσει δίκτυα δεδομένων μεταξύ απομακρυσμένων κόμβων. Στόχος ήταν η δημιουργία μίας υποδομής που θα επέτρεπε σε διαφορετικά δίκτυα να επικοινωνήσουν μεταξύ τους χωρίς η μερική κατάρρευση κάποιου υποδικτύου ή μέρους του δικτύου, να συντελέσει στην ολική κατάρρευση.

Ο τρόπος για την επίτευξη αυτού του στόχου ήταν η δημιουργία μίας αποκεντρωμένης δομής η οποία, σε αντίθεση με την περίπτωση ενός κεντρικού σημείου ελέγχου, δεν είναι ευάλωτη σ' ένα πλήγμα, καθώς η απουσία κάποιου κόμβου της, δε συνεπάγεται με την κατάρρευση όλου του δικτύου.

Μπορούμε πολύ εύκολα να κατανοήσουμε το έργο και τους στόχους του, αν αναλογιστούμε τις τεχνολογικές εξελίξεις και κοινωνικοπολιτικές συνθήκες της εποχής: βρισκόμαστε στα μέσα του ψυχρού πολέμου, όπου κάθε πλευρά αναζητά το στρατηγικό πλεονέκτημα έναντι της άλλης. Οι υπολογιστές ήδη χρησιμοποιούνται ευρέως σε στρατιωτικές εφαρμογές και προκύπτει η ανάγκη διαφορετικά συστήματα να μπορούν να επικοινωνούν μεταξύ τους ακόμα και στο ενδεχόμενο σημαντικό μέρος της υποδομής να καταστραφεί λόγω ενός πυρηνικού πολέμου. Αυτό το έργο λοιπόν που θα δημιουργούσε μία απρόσβλητη υποδομή ονομάστηκε «*Internetting project*» και το αποτέλεσμά του ήταν το *ARPANET*, ο πρόγονος του σημερινού *internet*.

Το 1969 συνδέθηκαν μεταξύ τους οι τέσσερις πρώτοι κόμβοι του *ARPANET*, και συγκεκριμένα τα Πανεπιστήμια των ΗΠΑ: *University of California Los Angeles*, *Stanford Research Institute*, *University of California Santa Barbara* και *University of Utah*. Ως κόμβοι χρησιμοποιήθηκαν mini-computers με κεντρική μνήμη 12 Kbytes και τηλεπικοινωνιακές γραμμές εύρους ζώνης (*bandwidth*) 50 Kbps. Ενδεικτικά αναφέρουμε ότι σήμερα οι προσωπικοί υπολογιστές έχουν περίπου 10.000 φορές περισσότερη μνήμη και ο κορμός (*backbone*) του *Internet*, δηλαδή οι επικοινωνιακοί σύνδεσμοι υψηλού εύρους ζώνης και διαθεσιμότητας που συνδέουν τα κεντρικά σημεία μεταγωγής μεταξύ τους και με το υπόλοιπο υποδίκτυο, έχει ένα τυπικό εύρος ζώνης περίπου 40-155 Kbps.

Το πειραματικό *ARPANET* ήταν τόσο επιτυχημένο ώστε πολλοί οργανισμοί που ήταν συνδεδεμένοι σε αυτό άρχισαν να το χρησιμοποιούν για επικοινωνία σε καθημερινό επίπεδο. Το 1975 το *ARPANET* μετατράπηκε από ένα ερευνητικό δίκτυο σε ένα πλήρως λειτουργικό ενώ την ευθύνη της διαχείρισής του ανέλαβε το *Defense Communications Agency (DCA)*. Ωστόσο η ανάπτυξη του *ARPANET* δεν σταμάτησε λόγω της μετατροπής από ένα ερευνητικό πρόγραμμα σε ένα λειτουργικό δίκτυο. Δεν πρέπει να ξεχνάμε ότι η ανάπτυξη του TCP/IP έγινε αφού το *ARPANET* σταμάτησε να είναι ένα πειραματικό δίκτυο.

Το 1983 ήταν μία σημαντική χρονιά για το *Internet* αφού τότε έγιναν τα πρώτα βήματα για τη δημιουργία του TCP/IP, το οποίο αρχικά ξεκίνησε ως ένα πρόγραμμα τυποποίησης του Αμερικανικού Στρατού, βάσει του οποίου όλες οι συνδεδεμένες συσκευές στο δίκτυο έπρεπε να κάνουν χρήση αυτού του πρωτοκόλλου. Για αυτόν το λόγο η *DARPA* χρηματοδότησε τους *Bolt, Beranek* και *Newman (BBN)* να ενσωματώσουν το παραπάνω πρωτόκολλο στο *Berkeley (BSD) Unix*.

Από τη στιγμή που πραγματοποιήθηκε η τυποποίηση του TCP/IP (*Transmission Control Protocol / Internet Protocol*), ο όρος *Internet* άρχισε να χρησιμοποιείται όλο και συχνότερα. Την ίδια χρονιά πραγματοποιήθηκε και ο διαχωρισμός του παλιού *ARPANET*, στο *MILNET* το διαβαθμισμένο δηλαδή μέρος του δικτύου το οποίο ελέγχεται από τη *Defense Data Network (DDN)* και δεσμεύει 68 από τους συνολικά 113 κόμβους και σ' ένα νέο αλλά μικρότερο *ARPANET*, για πανεπιστημιακή χρήση. Το 1990 το *ARPANET* παύει να υπάρχει, έχοντας παραχωρήσει τη θέση του σε αυτό που ονομάστηκε *Internet* (διαδίκτυο).

Το *Internet* απομακρύνθηκε πάρα πολύ από το σκοπό για τον οποίο αρχικά δημιουργήθηκε. Η *DDN* αποτελεί ακόμα και σήμερα ένα σημαντικό κομμάτι του δικτύου και υπηρεσίες όπως η *DCA* ασκούν ακόμα έλεγχο πάνω του. Όμως όλα τα επιπλέον εθνικά δίκτυα που συνδέθηκαν με αυτό άλλαξαν πάρα πολύ τον αρχικό χαρακτήρα του και το κατέστησαν πραγματικά διεθνές.

Ένα χαρακτηριστικό σημείο που δείχνει την επιτυχία του δικτύου είναι η σύγχυση που έχει επικρατήσει με τη λέξη *internet*. Με τον όρο αυτό περιγράφουμε το σύνολο πολλών διαφορετικών δικτύων τα οποία προκειμένου να επικοινωνήσουν μεταξύ τους χρησιμοποιούν κάποιο κοινό πρωτόκολλο. Αντίθετα τα δίκτυα που περιγράφονται με τον όρο *Internet* (κεφαλαίο I) είναι και πάλι συνδεδεμένα μεταξύ τους δίκτυα, που για να επικοινωνήσουν μεταξύ τους όμως χρησιμοποιούν ένα συγκεκριμένο πρωτόκολλο, το IP (*Internet Protocol*).

2.2 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ TCP/IP

Η ραγδαία αυτή εξάπλωση του TCP/IP στο *Internet* δεν οφείλονταν στο γεγονός ότι στο πρώιμο δίκτυο επιβλήθηκε από το Στρατό, αλλά

κυρίως λόγω του ότι αποτέλεσε τη συγκεκριμένη χρονική στιγμή αξιόπιστη λύση για την επίτευξη της διασύνδεσης, αλλά και λόγω του ότι ενσωμάτωνε ορισμένα ιδιαίτερα χαρακτηριστικά:

- Γνωρίσματα ανοιχτού κώδικα αφού το συγκεκριμένο πρωτόκολλο ήταν ανεξάρτητο και από το είδος του υπολογιστή στον οποίο λειτουργούσε αλλά και από το λειτουργικό του σύστημα.
- Ήταν επίσης ανεξάρτητο από το φυσικό μέσο επικοινωνίας, αφού μπορούσε να λειτουργήσει πάνω από *Ethernet*, *Token Ring*, *Dial-up line*, *X.25* και πολλά άλλα.
- Είχε ένα κοινό σύστημα διευθυνσιοδότησης
- Δομή και λειτουργία ώστε να μπορέσει να εξυπηρετήσει πάρα πολλές διαφορετικές μεταξύ τους υπηρεσίες.

2.3 ΜΟΝΤΕΛΟ OSI

Προκειμένου να κατανοήσουμε τη διασύνδεση των υπολογιστών στο δίκτυο αλλά και οποιασδήποτε άλλης συσκευής θα πρέπει για την περιγραφή να χρησιμοποιήσουμε κοινούς όρους. Ένα τέτοιο κοινό πλαίσιο με όρους που αναφέρονται στη διασύνδεση συσκευών για μεταφορά δεδομένων είναι το μοντέλο **OSI** (*Open System Interconnect*), το οποίο περιγράφει τους όρους που χρησιμοποιούμε όταν αναφερόμαστε σε δίκτυα υπολογιστών αλλά και καθορίζει τους κανόνες της μεταξύ τους επικοινωνίας καθώς το φυσικό μέσο σύνδεσης. Το μοντέλο αυτό έχει τόσο ευρέως χρησιμοποιηθεί που πολλές φορές η αναφορά σε αυτό είναι αναπόφευκτη.

7	Application Layer Περιλαμβάνει εφαρμογές που η λειτουργία τους απαιτεί δίκτυο.
6	Presentation Layer Τυποποιεί το πώς απεικονίζονται τα δεδομένα στις εφαρμογές.
5	Session Layer Διαχειρίζεται τους κανόνες επικοινωνίας μεταξύ των εφαρμογών.
4	Transport Layer Πραγματοποιεί τον εντοπισμό και τη διόρθωση των λαθών κατά την επικοινωνία των δύο άκρων.
3	Network Layer Πραγματοποιεί τη σύνδεση του δικτύου με τις

	εφαρμογές.
2	Data Link Layer Εξασφαλίζει την αξιόπιστη μεταφορά δεδομένων μέσα από το φυσικό μέσο.
1	Physical Layer Καθορισμός των φυσικών χαρακτηριστικών του μέσου που χρησιμοποιείται για τη διασύνδεση στο δίκτυο.

Πίνακας 1.1 Τα 7 επίπεδα του μοντέλου OSI

Το μοντέλο *OSI* αποτελείται από 7 επίπεδα (*Layers*). Σε καθένα από αυτά περιγράφονται όλες οι λειτουργίες που απαιτούνται για την αποστολή ή λήψη δεδομένων μεταξύ των εφαρμογών και του δικτύου. Κάθε επίπεδο δεν περιγράφει ένα μόνο πρωτόκολλο αλλά αντίθετα περιγράφει τις λειτουργίες εκείνες που μπορούν να γίνουν από πολλά πρωτόκολλα. Επομένως σε κάθε επίπεδο υπάρχουν περισσότερα του ενός πρωτόκολλα τα οποία είναι απαραίτητα προκειμένου να λειτουργήσουν οι υπηρεσίες που περιγράφονται από το μοντέλο *OSI* στο συγκεκριμένο επίπεδο. Για παράδειγμα το *FTP (File Transfer Protocol)* και το *SMTP (Simple Mail Transfer Protocol)* είναι διαφορετικά πρωτόκολλα του *Application Layer*. Με το μοντέλο αυτό εξασφαλίζουμε την «απομόνωση» των επιπέδων μεταξύ τους. Αποτέλεσμα του παραπάνω είναι νέες υπηρεσίες και πρωτόκολλα που δημιουργούνται με την πάροδο του χρόνου δεν απαιτούν αλλαγή στη *hardware* υποδομή μας και αλλαγή στη *hardware* υποδομή δεν οδηγεί σε επαναπρογραμματισμό των εφαρμογών των ανώτερων επιπέδων.

Ακολουθεί μία σύντομη περιγραφή του κάθε επιπέδου και των λειτουργιών που επιτελούνται σε αυτό:

2.3.1 APPLICATION LAYER:

Πρόκειται για το επίπεδο στο οποίο εκτελούνται οι εφαρμογές και επικοινωνούν με τους χρήστες. Ειδικά για το *TCP/IP*, όπως αυτό περιγράφεται από το μοντέλο *OSI* οτιδήποτε εκτελείται πάνω από το *Transport Layer* ουσιαστικά ανήκει σ' αυτό το επίπεδο, αφού τα υπόλοιπα σχεδόν σε κάθε περίπτωση «ενσωματώνονται» με το *Transport Layer*.

2.3.2 PRESENTATION LAYER:

Στην περίπτωση που οι εφαρμογές συνεργάζονται μεταξύ τους ανταλλάσσοντας δεδομένα στο επίπεδο αυτό εκτελούνται όλες οι

απαραίτητες εκείνες ρουτίνες οι οποίες κάνουν συμβατή αυτή τη συναλλαγή. Στο TCP/IP ένα μεγάλο μέρος αυτού του επιπέδου εκτελείται στο Application Layer.

2.3.3 SESSION LAYER:

Όπως ακριβώς και με το προηγούμενο επίπεδο το *Session Layer* δεν αποτελεί ξεχωριστό επίπεδο στο TCP/IP. Με τον όρο *session* εννοούμε ένα κανάλι επικοινωνίας μεταξύ δύο υπολογιστών για μία όμως συγκεκριμένη εφαρμογή. Ειδικά για το TCP/IP ο όρος αυτός έχει αντικατασταθεί από τους *socket* και *port*, για τους οποίους όμως θα μιλήσουμε αναλυτικότερα στο δεύτερο κεφάλαιο (παράγραφοι 2.4.4 και 2.4.5) όταν θα αναφερθούμε με περισσότερες λεπτομέρειες στα *embedded web servers* συστήματα.

2.3.4 TRANSPORT LAYER:

Το μεγαλύτερο μέρος των πρωτοκόλλων του TCP/IP υλοποιείται σ' αυτό το επίπεδο. Το επίπεδο αυτό στο μοντέλο OSI εξασφαλίζει πως τα δεδομένα παρελήφθησαν ακριβώς όπως εστάλησαν. Σ' αυτό το σημείο θα αναφερθούμε ακριβώς στο όνομα του πρωτοκόλλου του *Internet*, δηλαδή στο TCP/IP. Όλα τα πρωτόκολλα του δικτύου περιγράφονται από το όνομά τους φυσικά, προσθέτοντας το όνομα του μέρους εκείνου του πρωτοκόλλου το οποίο ευθύνεται για την παραπάνω λειτουργία, δηλαδή την αξιοπιστία στη μετάδοση των δεδομένων. Με τον όρο αξιοπιστία δεν αναφερόμαστε στην ασφάλεια (κρυπτογράφηση κ.α.) κατά την αποστολή των δεδομένων αλλά στην εξασφάλιση ότι ο αποδέκτης θα πάρει τα δεδομένα από τον αποστολέα χωρίς αυτά να έχουν αλλοιωθεί, επανεκπέμποντάς τα αν χρειαστεί. Το μέρος του IP που αναλαμβάνει αυτή τη λειτουργία είναι το TCP (*Transmission Control Protocol*) και αποτελεί το ένα κομμάτι του *Transport Layer*. Το άλλο κομμάτι του επιπέδου αυτού είναι το UDP (*User Datagram Protocol*) το οποίο δεν πραγματοποιεί ελέγχους για αξιόπιστη μεταφορά δεδομένων από άκρο σ' άκρο.

2.3.5 NETWORK LAYER:

Το επίπεδο αυτό διαχειρίζεται τις συνδέσεις διαμέσου του δικτύου και απομονώνει τα πρωτόκολλα των ανώτερων επιπέδων. Στο επίπεδο αυτό καθορίζονται οι διευθύνσεις των διαφόρων άκρων του δικτύου καθώς και όλη η διαδικασία που πρέπει ν' ακολουθηθεί προκειμένου να εντοπιστεί κάποιο άκρο του δικτύου.

2.3.6 DATA LINK LAYER:

Το επίπεδο αυτό διαχειρίζεται τις συνδέσεις διαμέσου του δικτύου σε *hardware* επίπεδο όμως, και παρέχει τους κανόνες βάσει των οποίων το IP επικοινωνεί με το *hardware* του μηχανήματος.

2.3.7 PHYSICAL LAYER:

Εδώ καθορίζονται τα χαρακτηριστικά του *hardware* που απαιτείται προκειμένου να επιτευχθεί η σύνδεση. Χαρακτηριστικά όπως: επίπεδα τάσης, ο αριθμός και θέση του κάθε ακροδέκτη στις κάρτες καθορίζονται σ' αυτό το επίπεδο. Ακόμη στάνταρτ όπως η RS232C και το V35, σύνδεση δικτύου με το IEEE 802.3 περιγράφονται λεπτομερώς από το επίπεδο αυτό. Το TCP/IP δεν καθορίζει δικούς του κανόνες για το *hardware* και τη μετάδοση μέσα στο φυσικό μέσο. Αντίθετα αξιοποιεί τα ήδη υπάρχοντα. Αυτός άλλωστε είναι και ο λόγος δημιουργίας και ύπαρξης του μοντέλου OSI.

2.4 ΑΡΧΙΤΕΚΤΟΝΙΚΗ TCP/IP

Παρά το γεγονός ότι το μοντέλο OSI αποτελεί ένα είδος τυποποίησης σ' ότι αφορά τα δίκτυα και τη λειτουργία τους, ωστόσο δεν υπάρχει ακόμη κάποια συμφωνία ότι το TCP θα ακολουθεί τη δομή που καθορίζεται από το OSI.

Στο μοντέλο OSI τα δεδομένα περνούν διαμέσου των επτά επιπέδων για να καταλήξουν στο πρώτο και από εκεί να σταλούν προς το δίκτυο, ή παραλαμβάνονται στο πρώτο επίπεδο και από εκεί ανεβαίνουν την ιεραρχία για να καταλήξουν στο έβδομο.

Οι περισσότερες περιγραφές του TCP/IP αναφέρονται σε μία αρχιτεκτονική τεσσάρων επιπέδων:

4	Application Layer Περιλαμβάνει εφαρμογές που η λειτουργία τους απαιτεί δίκτυο.
3	Transport Layer Παρέχει υπηρεσίες για τη μετάδοση δεδομένων από σε άκρο .
2	Internet Layer Καθορίζει τα δεδομένα και διαχειρίζεται τη δρομολόγησή τους διαμέσου του δικτύου.

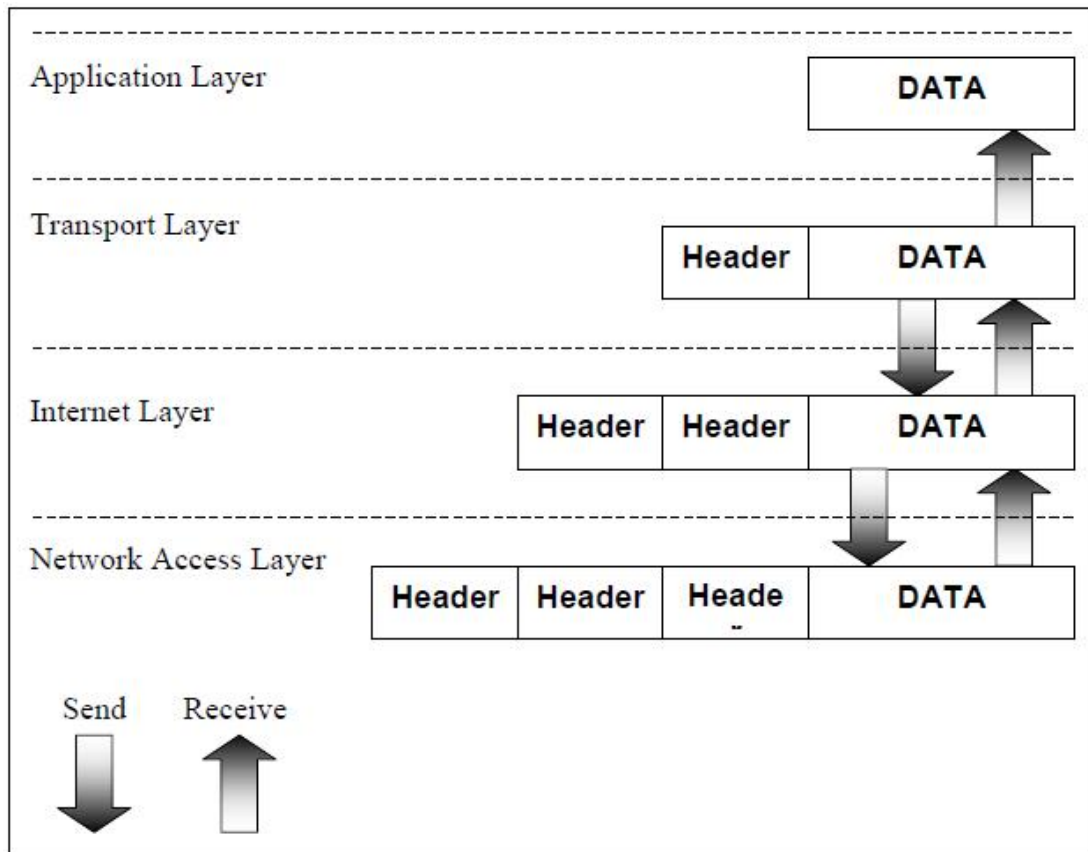
1	Network Access Layer Καθορισμός των φυσικών χαρακτηριστικών του φυσικού μέσου που χρησιμοποιείται για τη διασύνδεση στο δίκτυο.
----------	-------------------------------------------------------------------------------------------------------------------------------------------

Πίνακας 1.2 Τα 4 επίπεδα του TCP/IP

Η ίδια λογική ακολουθείται και στο μοντέλο των τεσσάρων επιπέδων θεωρώντας βέβαια ότι τα επτά επίπεδα του μοντέλου OSI συγχωνεύονται σε τέσσερα χάριν απλουστεύσεως. Όταν λοιπόν δεδομένα πρόκειται να σταλούν κατεβαίνοντας από το ένα επίπεδο στο αμέσως κατώτερο, προστίθενται σ' αυτά ορισμένες πληροφορίες που αφορούν το συγκεκριμένο *layer*. Οι επιπρόσθετες αυτές πληροφορίες ονομάζονται *header* (επικεφαλίδα). Καθώς τα δεδομένα με το *header* περνούν από το επόμενο *layer* ένας ακόμα *header* προστίθεται κ.ο.κ. Κάθε επίπεδο δηλαδή προσθέτει στα δεδομένα το δικό του *header* το οποίο περιέχει πληροφορίες οι οποίες έχουν να κάνουν με τη δρομολόγηση κυρίως των δεδομένων. Η διαδικασία αυτή ονομάζεται *encapsulation* (ενθυλάκωση).

Κατά τη λήψη δεδομένων αυτά ακολουθούν την αντίθετη πορεία όσον αφορά τα επίπεδα, ενώ κάθε επίπεδο αφαιρεί και επεξεργάζεται το *header* που το αφορά παραδίδοντας τα δεδομένα στο αμέσως ανώτερο του επίπεδο με ένα *header* λιγότερο. Τελικά τα δεδομένα καταλήγουν χωρίς καμία επικεφαλίδα στο *application layer* με τελικό αποδέκτη το χρήστη του μηχανήματος.

Η παραπάνω διαδικασία φαίνεται στο διάγραμμα:

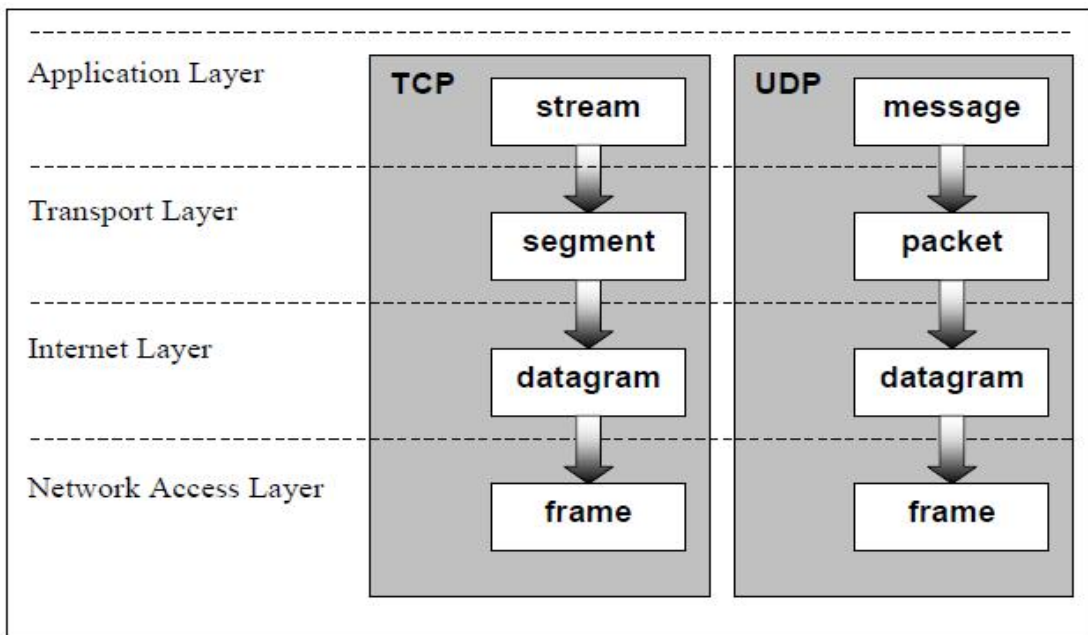


Διάγραμμα 1.1 Encapsulation

Κάθε επίπεδο έχει τη δίκη του δομή για το *header* του και αποθηκεύει σ' αυτό πληροφορίες έτσι ώστε να εξασφαλιστεί η σωστή μεταφορά των δεδομένων.

Προκειμένου τα δεδομένα να μεταφερθούν διαμέσου των επιπέδων είναι απαραίτητο να διασπαστούν σε μικρότερα τμήματα (ή όπως πολλές φορές χρησιμοποιείται η έκφραση «πακέτα»). Αυτή η διάσπαση γίνεται αρχικά από το ανώτερο επίπεδο, το *application layer*, και ονομάζονται *streams* εφόσον αποτελούν τμήματα του TCP ή *messages* αν είναι τμήματα του UDP.

Όπως αναφέραμε προηγουμένως το *Transport Layer* χωρίζεται σε δύο μέρη, το αξιόπιστο (*reliable*), το TCP και το μη αξιόπιστο (*unreliable*), το UDP. Όπως φαίνεται και στο διάγραμμα 1.2 τα τμήματα του TCP σ' αυτό το επίπεδο είναι τα *segments* ενώ του UDP τα *packets*. Στα άλλα δύο επίπεδα στα τμήματα των δεδομένων είτε αυτά είναι του TCP είτε του UDP, επειδή η πληροφορία που αποθηκεύεται στο *header* δεν έχει σχέση με την αξιόπιστη ή μη μετάδοση των δεδομένων αλλά με τη διεύθυνση του αποστολέα ή του παραλήπτη δεν έχουμε διαφορετικά *headers* με αποτέλεσμα στα κομμάτια των δεδομένων να προστίθενται κοινές πληροφορίες και το όνομά τους να είναι το ίδιο.



Διάγραμμα 1.2 Δομές Δεδομένων

2.5 ΔΙΑΔΙΚΤΥΩΣΗ

Έχοντας περιγράψει τα πρωτόκολλα, τα οποία εξασφαλίζουν το δίκτυο, πρέπει να καθοριστούν οι κανόνες για τη σύνδεση δύο άκρων στο δίκτυο. Όπως αναφέραμε προηγουμένως κάθε συσκευή, η οποία είναι συνδεδεμένη στο δίκτυο, ορίζεται μοναδικά σ' αυτό με μία διεύθυνση (*IP* διεύθυνση) ενώ η μεταφορά δεδομένων ανάμεσα σε δύο άκρα ακολουθεί ορισμένους κανόνες δρομολόγησης (*routing*).

Προκειμένου λοιπόν να επικοινωνήσουν δύο συνδεδεμένες στο δίκτυο συσκευές απαιτούνται τρεις προϋποθέσεις:

- **Διευθυνσιοδότηση:** όπως αναφέραμε κάθε συσκευή που συνδέεται στο δίκτυο έχει μία μοναδική διεύθυνση. Ωστόσο υπάρχουν συσκευές οι οποίες μπορούν να έχουν περισσότερες από μία μοναδικές διευθύνσεις, οι δρομολογητές (*routers*) για τους οποίους όμως θα μιλήσουμε αμέσως μετά σ' αυτό το κεφάλαιο.
- **Δρομολόγηση:** οι πύλες (*gateways*) αναλαμβάνουν να μεταφέρουν τα δεδομένα στο σωστό «δίκτυο».
- **Πολύπλεξη:** στα δεδομένα περιέχονται πληροφορίες για το πρωτόκολλο και το *port* έτσι ώστε όταν τελικά τα δεδομένα φτάσουν στον παραλήπτη-συσκευή, αυτά να παραληφθούν από την κατάλληλη εφαρμογή.

2.6 IP ΔΙΕΥΘΥΝΣΗ

Οι διευθύνσεις αυτές του *Internet* είναι αριθμοί των 32 *bits*, ενώ η διεύθυνση είναι της μορφής a.b.c.d, όπου κάθε a, b, c, d είναι ακέραιοι

αριθμοί από 0 έως 255 (8 *bits*). Ουσιαστικά κάθε διεύθυνση χωρίζεται σε δύο μέρη, το *network* μέρος και το *host* μέρος. Επιπλέον ο συνολικός αριθμός των διευθύνσεων χωρίζεται σε τέσσερις κλάσεις (A, B, C και D).

Κλάση A

Σ' αυτήν ανήκουν οι IP διευθύνσεις που το πρώτο τους *bit* είναι το 0. Οι διευθύνσεις αυτές έχουν 8 *bits network* μέρος και 24 *bits host* μέρος. Πρόκειται δηλαδή για τις διευθύνσεις από 0.0.0.0 έως 127.255.255.255. Ωστόσο όλες αυτές οι διευθύνσεις δεν είναι διαθέσιμες στο δίκτυο. Έτσι κάθε διεύθυνση της μορφής x.0.0.0, δηλαδή το *network* μέρος ή όπως συχνά αναφερόμαστε σε αυτό με τον όρο «καλώδιο» δεν μπορεί να ανήκει σε συσκευή διότι αποτελεί τη διεύθυνση του φυσικού μέσου. Επίσης στις διευθύνσεις της μορφής x.x.x.255 αποστέλλονται τα *broadcasts* από τις συσκευές οπότε και αυτές είναι δεσμευμένες. Με τον όρο *broadcast* εννοούμε τα δεδομένα τα οποία αποστέλλονται προς όλους τους *hosts* οι οποίοι έχουν το ίδιο *network* μέρος. Τέλος οι διευθύνσεις 0.x.x.x και 127.x.x.x είναι επίσης δεσμευμένες αφού αυτή με το *network* μέρος 0 ονομάζεται *default route* και αυτή με το *network* μέρος 127 λέγεται *loopback address* και χρησιμοποιείται για εσωτερική χρήση σε κάθε *host*. Επομένως σ' αυτήν την κλάση έχουμε συνολικά 126 *networks* με εκατομμύρια *hosts* το καθένα.

Κλάση B

Σ' αυτήν ανήκουν οι IP διευθύνσεις που τα δύο πρώτα τους *bits* είναι τα 1 και 0. Οι διευθύνσεις αυτές έχουν 16 *bits network* μέρος και 16 *bits host*. Πρόκειται δηλαδή για τις διευθύνσεις από 128.0.0.0 έως 191.255.255.255. Όπως και προηγουμένως όλες αυτές οι διευθύνσεις δεν είναι διαθέσιμες στο δίκτυο. Έτσι κάθε διεύθυνση της μορφής x.x.0.0, δηλαδή το *network* μέρος δεν μπορεί να ανήκει σε συσκευή διότι αποτελεί τη διεύθυνση του φυσικού μέσου. Επίσης στις διευθύνσεις της μορφής x.x.255.255 αποτελούν τις *broadcasts* διευθύνσεις. Επομένως σ' αυτήν την κλάση έχουμε αρκετές χιλιάδες *networks* με αρκετούς χιλιάδες *hosts* το καθένα.

Κλάση C

Σ' αυτήν ανήκουν οι IP διευθύνσεις που τα τρία πρώτα τους *bits* είναι τα 1, 1 και 0. Οι διευθύνσεις αυτές έχουν 24 *bits network* μέρος και 8 *bits host*. Πρόκειται δηλαδή για τις διευθύνσεις από 192.0.0.0 έως 223.255.255.255. Όπως και προηγουμένως όλες αυτές οι διευθύνσεις δεν είναι διαθέσιμες στο δίκτυο. Έτσι κάθε διεύθυνση της μορφής x.x.x.0, δηλαδή το *network* μέρος δεν μπορεί να ανήκει σε συσκευή διότι αποτελεί τη διεύθυνση του φυσικού μέσου. Επίσης στις διευθύνσεις της

μορφής x.255.255.255 αποτελούν τις *broadcasts* διευθύνσεις. Δηλαδή σ' αυτήν την κλάση έχουμε εκατομμύρια *networks* με αρκετούς 254 *hosts* το καθένα.

Κλάση D

Σ' αυτήν ανήκουν οι IP διευθύνσεις που τα τρία πρώτα τους *bits* είναι τα 1, 1 και 1. Σ' αυτές τις διευθύνσεις δεν υπάρχει ο διαχωρισμός σε *network* και *host* μέρος. Πρόκειται για *multicast* διευθύνσεις που χρησιμοποιούνται για πειραματικούς και ερευνητικούς σκοπούς. Οι *multicast* αυτές διευθύνσεις χρησιμοποιούνται για να αναγνωρίζονται ομάδες υπολογιστών που χρησιμοποιούν ένα κοινό πρωτόκολλο, όπως ακριβώς οι υπόλοιπες διευθύνσεις χρησιμοποιούνται για να καθορίζουν ομάδες υπολογιστών που βρίσκονται στο ίδιο *network*.

Τέλος υπάρχουν ορισμένα *networks* τα οποία δεν είναι διαθέσιμα στο *Internet* αλλά μπορούν να χρησιμοποιηθούν μόνο από εσωτερικά δίκτυα για ερευνητικούς ή εκπαιδευτικούς σκοπούς ή ακόμη γιατί είναι απαραίτητη η διευθυνσιοδότηση ενός εσωτερικού δικτύου TCP/IP, το οποίο για τη λειτουργία του απαιτεί όπως είναι φυσικό IP διευθύνσεις όμως δεν χρειάζεται τη σύνδεση με τον εξωτερικό κόσμο. Γι' αυτό το λόγο έχουν δεσμευτεί τρία *networks* (ένα από κάθε κλάση), τα 10.x.x.x, 178.x.x.x και το 192.168.x.x.

2.7 ΥΠΟΔΙΚΤΥΑ (SUBNETS)

Όπως αναφέραμε προηγουμένως σε κάθε IP διεύθυνση έχουμε δύο μέρη: το *network* και το *host*. Ο αριθμός των *bits* του κάθε μέρους εξαρτάται από τη κλάση της διεύθυνσης. Με τον όρο *subnet* λοιπόν αναφερόμαστε στη διαδικασία εκείνη κατά την οποία αλλάζουμε τον αριθμό των *bits* των δύο μερών σε τελικό σκοπό να μειώσουμε τον αριθμό των *hosts* σε ένα δίκτυο.

Η παραπάνω διαδικασία είναι επιβεβλημένη προκειμένου να γίνει ευκολότερη η διαχείριση ενός *network*. Αν ένας οργανισμός έχει για παράδειγμα στο δίκτυό της ένα *network* κλάσης C, αυτό σημαίνει πως θα έχει συνολικά 254 *hosts*. Με τα *subnets* μπορούμε να χωρίσουμε αυτό το δίκτυο σε άλλα μικρότερα (με λιγότερους *hosts*) και επομένως να έχουμε μία αποκέντρωση στη διαχείριση όλων αυτών των υποδικτύων.

Επιπλέον όπως θα δούμε και παρακάτω στη δρομολόγηση, οι δρομολογητές (*routers*) μπορούν να συνδέσουν διαφορετικά μεταξύ τους δίκτυα αρκεί το καθένα από αυτά να ανήκει σε διαφορετικό *network*. Με τον τρόπο λοιπόν αυτό μπορούμε από μια *network* διεύθυνση να δημιουργήσουμε πολλές άλλες αποδίδοντας κάθε μία από αυτές σε ένα φυσικό δίκτυο.

Ο χωρισμός του *network* σε *subnets* πραγματοποιείται με τη χρήση μίας μάσκας, της *subnet mask*, οποία είναι επίσης ένας αριθμός με 32 *bits* μήκος για τα οποία ισχύει: αν το *bit* στη μάσκα είναι 1 τότε το αντίστοιχο *bit* της διεύθυνσης ανήκει στο *network* μέρος, ενώ αν το *bit* στη μάσκα είναι 0 τότε το αντίστοιχο *bit* της διεύθυνσης ανήκει στο *host* μέρος.

Για παράδειγμα η μάσκα για ένα οποιοδήποτε δίκτυο κλάσης B είναι η 255.255.0.0. Για έναν οργανισμό ή μία επιχείρηση τα δίκτυα κλάσης C είναι τα περισσότερο προτιμητέα αφού ο αριθμός των 254 *hosts* που παρέχουν είναι ικανοποιητικός. Ωστόσο ο αριθμός τους είναι σχετικά μικρός για να ικανοποιήσουν τη ζήτηση. Από την άλλη υπάρχει ένας μεγάλος αριθμός κλάσης B δικτύων τα οποία όμως διαθέτουν χιλιάδες *hosts* πράγμα που τα καθιστά δύσχρηστα. Με τα *subnets* μπορούμε να επεκτείνουμε το *network* μέρος ενός δικτύου κλάσης B χρησιμοποιώντας τη μάσκα 255.255.255.0. Τα δύο πρώτα bytes καθορίζουν το *network* μέρος της κλάσης B, το τρίτο byte το *subnet* ενώ το τέταρτο byte τον *host*.

Στον παρακάτω πίνακα μπορούμε να δούμε παραδείγματα από τη συσχέτιση ενός *network* με μία *subnet mask*.

IP διεύθυνση	Subnet Mask	Αποτέλεσμα
128.66.12.1	255.255.255.0	Host 1 στο δίκτυο 128.66.12.0
130.97.16.132	255.255.255.192	Host 4 στο δίκτυο 130.97.16.128
195.178.16.66	255.255.255.192	Host 2 στο δίκτυο 195.178.16.64
132.90.132.5	255.255.240.0	Host 1029 στο δίκτυο 132.90.128.0
18.20.16.91	255.255.0.0	Host 4187 στο δίκτυο 18.20.0.0

Η όλη αυτή διαδικασία του διαχωρισμού ενός *network* σε επιμέρους *subnets* έχει να κάνει μόνο με θέματα διαχείρισης και όχι λειτουργίας του δικτύου και επομένως δεν είναι υποχρεωτική προκειμένου να λειτουργήσει ένα δίκτυο.

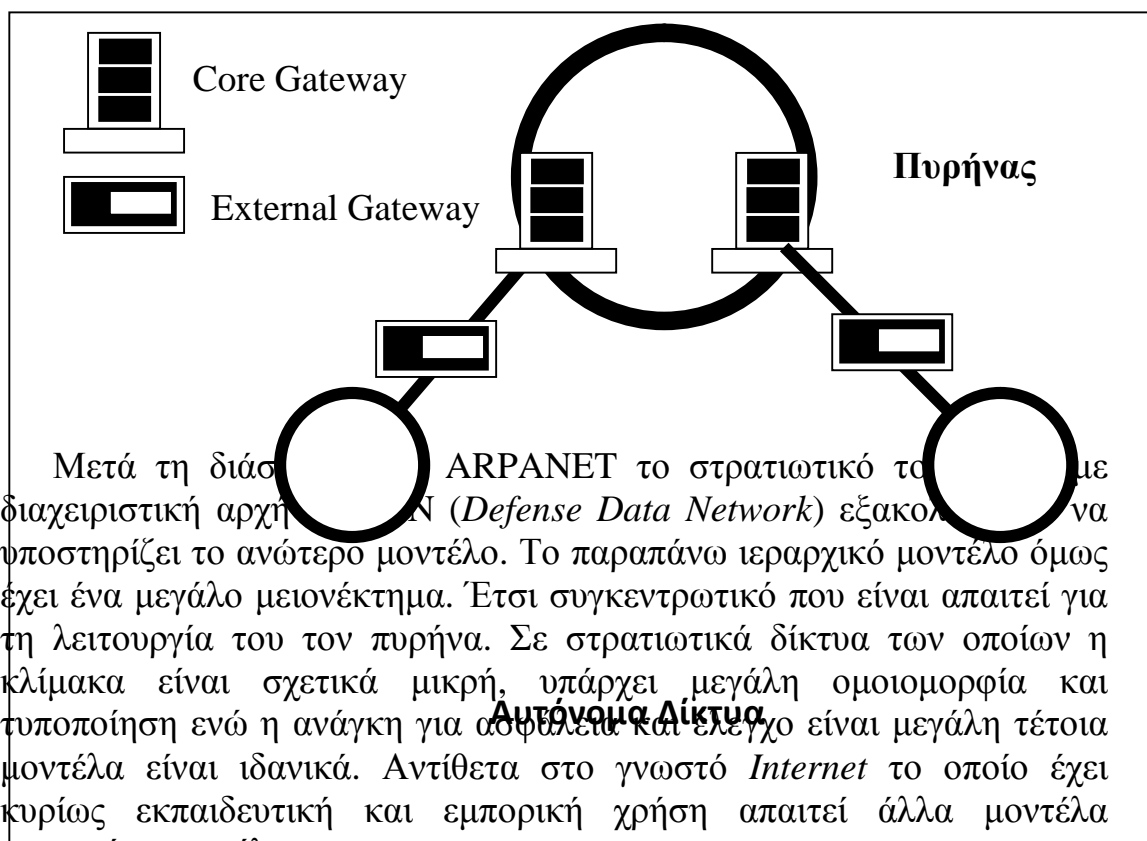
2.8 ΔΡΟΜΟΛΟΓΗΣΗ (ROUTING)

Έχοντας καθορίσει τη διεύθυνση του κάθε *host* καθώς και το *network* στο οποίο ανήκει έχουμε καθορίσει μοναδικά τον *host* στο δίκτυο. Ακολούθως πρέπει να περιγράψουμε τους κανόνες της δρομολόγησης, δηλαδή της μεταφοράς των δεδομένων μεταξύ των *hosts*. Διάφορα μοντέλα αναπτύχθηκαν από την αρχή κιόλας του δικτύου.

Το πρώιμο ARPANET εφάρμοσε ένα κεντρικό μοντέλο βάσει του οποίου το ίδιο αποτελούσε τη ραχοκοκαλιά (*backbone*) του δικτύου. Αυτό το καθαρά κεντρικό σύστημα ήταν ο πυρήνας, ο οποίος συνδεόταν με το υπόλοιπο δίκτυο με τα *core gateways*. Όλα τα υποδίκτυα που ήταν συνδεδεμένα σε αυτό «ανακοίνωναν» πληροφορίες για την ταυτότητά

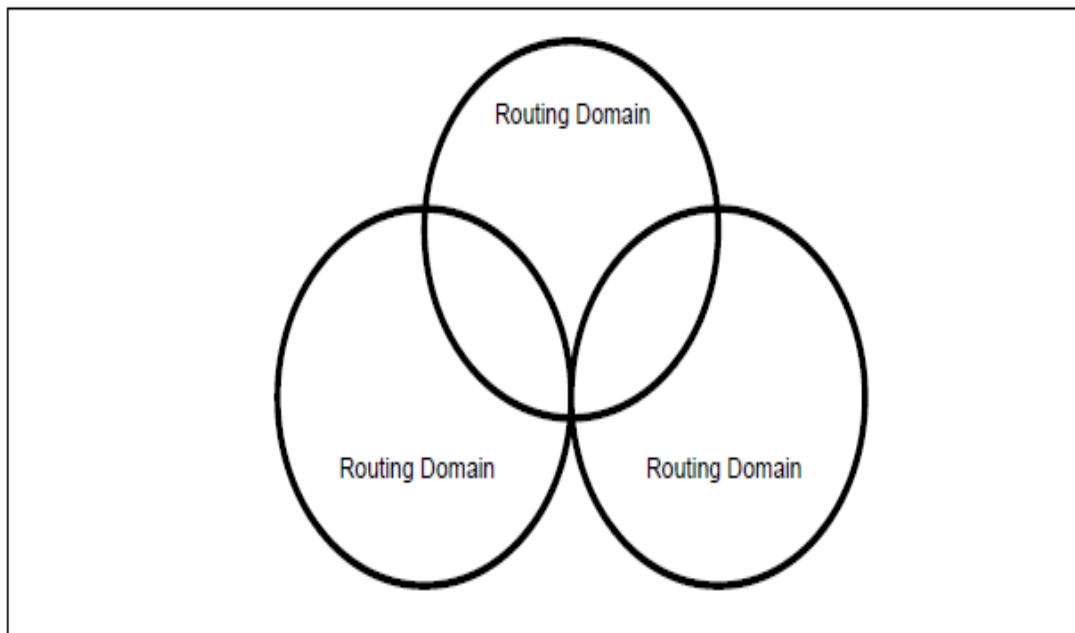
τους στον πυρήνα. Το πρωτόκολλο που ανέλαβε αυτή τη διαδικασία ήταν το GGP (*Gateway to Gateway Protocol*). Το κάθε επιμέρους δίκτυο δεν είχε καμία σχέση με αυτό το πρωτόκολλο παρά μόνο συνδεόταν με κάποιο *core gateway*.

Έξω από αυτόν τον πυρήνα υπήρχαν τα επιμέρους αυτόνομα δίκτυα. Κάθε τέτοιο αυτόνομο δίκτυο δεν αποτελούνταν από ένα μόνο δίκτυο αλλά ήταν ένα σύνολο από πολλά άλλα, είχε δικά του πρωτόκολλα για τη μεταφορά δεδομένων και πληροφοριών και ανακοίνωνε την ταυτότητά του στα άλλα αυτόνομα δίκτυα. Αυτές οι πληροφορίες ανέλαβε να μεταφέρει το EGP (*Exterior Gateway Protocol*).



Στο νέο αυτό μοντέλο που δημιουργήθηκε δεν υπάρχει πυρήνας αλλά αντί αυτού ισοδύναμα δίκτυα, τα *routing domains*. Κανένα από τα δίκτυα αυτά δεν χρειάζεται κάποιο άλλο για να λειτουργήσει. Το μόνο πράγμα που γίνεται είναι πως όταν κάποιο δίκτυο (*routing domain*) συνδεθεί με το υπόλοιπο δίκτυο τότε αυτό ανακοινώνει στα άλλα την ταυτότητα του (δηλαδή το *network* του) και λαμβάνει πληροφορίες για τα γειτονικά του δίκτυα, αλλά και οποιοδήποτε είναι ήδη συνδεδεμένο στο δίκτυο και πραγματοποιηθεί μία αλλαγή στο εσωτερικό του την ανακοινώνει στα υπόλοιπα. Το μοντέλο αυτό είναι πλήρως αποκεντρωμένο, εντελώς επεκτάσιμο με μοναδικό περιορισμό το συνολικό εύρος των IP διευθύνσεων ενώ η λειτουργία του δεν επηρεάζεται από κάποιο άλλο

ανώτερο στην ιεραρχία δίκτυο. Το πρωτόκολλο που αναλαμβάνει να μεταφέρει αυτές τις πληροφορίες μεταξύ των συνδεδεμένων δικτύων είναι το BGP (*Border Gateway Protocol*).



2.9 CLIENTS KAI SERVERS

Το *Internet* παρέχει τη σύνδεση ανάμεσα σε δύο κατηγορίες υπολογιστών: τους *servers* οι οποίοι παρέχουν στο δίκτυο τα ηλεκτρονικά έγγραφα και τους *clients* οι οποίοι κάνουν τις αιτήσεις προς τους *servers*. Οι διαδικασίες που λαμβάνουν χώρα σ' έναν *server* από την αρχή της αίτησης χαρακτηρίζονται με τον όρο *server-side*, ενώ οι διαδικασίες που πραγματοποιούνται στον *client* χαρακτηρίζονται με τον όρο *client-side*. Αυτό δε σημαίνει πως όταν ένας υπολογιστής χαρακτηριστεί ως *server* για μία υπηρεσία δεν μπορεί να είναι ταυτόχρονα *client* για μία άλλη.

Σ' αυτό το σημείο θα κάνουμε και την πρώτη μας αναφορά για την εργασία μας. Οι συσκευές που θα προγραμματίσουμε και θα χρησιμοποιήσουμε δεν είναι τίποτε άλλο από *web-servers*, δηλαδή ηλεκτρονικές συσκευές (θα μπορούσαμε να τους χαρακτηρίσουμε υπολογιστές με την ευρεία έννοια του όρου που όμως έχουν εξειδικευμένες δυνατότητες) οι οποίες μοιράζουν στο *Internet* HTML έγγραφα. Στο αμέσως επόμενο κεφάλαιο θα αναφερθούμε αναλυτικότερα για τους *web-servers*, στον προγραμματισμό τους και στον τρόπο λειτουργίας τους. Μόνο που στην περίπτωσή μας οι αιτήσεις μας προς τον *web-server* δεν γίνονται μόνο προκειμένου να «κατεβάσουμε» από αυτόν κάποια HTML σελίδα αλλά ουσιαστικά πρόκειται για εντολές προς το *hardware* που τον υποστηρίζει.

Το πρωτόκολλο που εκτελείται σ' έναν *web-server* είναι το HTTP (*Hypertext Transfer Protocol*) στην πόρτα 80 πρόκειται δηλαδή για μία

τυποποιημένη λειτουργία όπως παρατηρούμε από τον αριθμό της πόρτας (μικρότερη του 255).

2.10 APACHE

Είναι αδύνατον να μιλήσει για *Web Servers* χωρίς ν' αναφερθεί στον *apache*. Αποτελεί τον πλέον δημοφιλή *Web Server* του *Internet* και μάλιστα με μεγάλη διαφορά από τους προηγούμενους. Αποτελεί τη φυσική συνέχεια του πρώτου δημοφιλούς *web server* του NCSA HTTPd. Το 1994 ο *Rob McCool* (ο κατασκευαστής του NCSA HTTPd) έφυγε από τη NCSA και το πρόγραμμα πέρασε στην αφάνεια. Ουσιαστικά δεν υπήρχε κάποιος οργανισμός, ο οποίος να συλλέγει διορθώσεις πάνω στην εφαρμογή, να θέτει νέους στόχους και δυνατότητες και διανέμει μία λειτουργική εκτελέσιμη εφαρμογή στο δίκτυο.

Ωστόσο ο πηγαίος κώδικας υπήρχε διαθέσιμος στο *Internet*. Το Φεβρουάριο του 1995 οι *Brian Behlendorf* και *Cliff Skolnick* άρχισαν να μαζεύουν από το δίκτυο ιδέες και διορθώσεις πάνω στον παλιό *web server*. Έτσι γεννήθηκε ο NCSA HTTPd 1.3 από μία ομάδα εθελοντών προγραμματιστών οι οποίοι μάζευαν *patches* (διορθώσεις) από άλλους χρήστες στο *Internet*. Από τον τρόπο κατασκευής του ο νέος *web server* που προέκυψε ονομάστηκε *Apache (a patchy) server*, και η πρώτη του έκδοση 0.6.2, διατέθηκε για πρώτη φορά τον Απρίλιο του 1995.

Τα οκτώ πρώτα μέλη του *Apache Group* ήταν οι *Behlendorf, Skolnick, Roy T Fielding, Rob Hartill, David Robinson, Randy Terbush, Robbert S Thau* και *Andrew Wilson*.

Μετά την πρώτη έκδοση ακολούθησε από τον *Thau* η έκδοση 0.8.8 τον Αύγουστο του 1995, η οποία προγραμματίστηκε με εντελώς νέα αρχιτεκτονική, η οποία ουσιαστικά αποτελεί τη βάση για τις σημερινές εκδόσεις. Όπως θα δούμε στο 6^ο κεφάλαιο (*Embedded Web Servers*) απλές εκδόσεις του πυρήνα του *Apache* μαζί με μικροεπεξεργαστές συνεργάζονται προκειμένου να δημιουργήσουν ένα γραφικό περιβάλλον ελέγχου του μικροεπεξεργαστή βασισμένο πάνω στο TCP/IP.

Αν και οι *web servers* ξεκίνησαν με μοναδικό σκοπό την απεικόνιση ιστοσελίδων στο *Internet*, ο *Apache* ξεχώρισε γρήγορα λόγω ορισμένων ιδιαίτερων χαρακτηριστικών του.

- Κάθε εφαρμογή που «βγαίνει» στο *Internet* πρέπει να φροντίζει τόσο για την ασφάλεια των δεδομένων που διαθέτει όσο και για την ασφάλεια των δεδομένων του υπολογιστή στον οποίο εκτελείται. Ο *Apache* ανήκει στην κατηγορία των *daemons* του UNIX πράγμα που σημαίνει ότι αν κάποιος αναλάβει τον έλεγχο του (δηλαδή όπως λέμε στο UNIX αποκτήσει το *user id* του) μπορεί να πραγματοποιήσει οποιαδήποτε ενέργεια στα αρχεία του *web server*. Γι' αυτό το λόγο ο *apache* εκτελείται με το *user id* του χρήστη *nobody* (ελάχιστα δικαιώματα σ' ένα σύστημα UNIX). Επιπλέον υιοθέτησε από πολύ νωρίς

το πρότυπο AAA (*Authentication – Authorization – Access*) βάσει του οποίου υπάρχει η δυνατότητα να προστατευτεί ένας δικτυακός τόπος με ευαίσθητα δεδομένα με ελεγχόμενη πρόσβαση.

- Το σύστημα καταγραφής των ενεργειών και αιτήσεων που λαμβάνουν χώρα κατά τη λειτουργία του *server* είναι εξαιρετικά λεπτομερές και αποτελεσματικό.
- CGI (*Common Gateway Interface*) *scripts*. Με τον όρο αυτό αναφερόμαστε στη δυνατότητα του *Apache* να καλεί εκτελέσιμα αρχεία του συστήματος τα οποία παράγουν δυναμικά HTML κώδικα τον οποίο διαχειρίζεται ο *web server*. Τα CGI *scripts* αποτελούν μία *server side* εφαρμογή.
- Τέλος αναφέρουμε την σπονδυλωτή κατασκευή του *web server*. Εξαιτίας αυτής υπάρχει η δυνατότητα χωρίς να τροποποιούμε τον *server* και την αρχική του εγκατάσταση, να προσθέτουμε και να αφαιρούμε (εκτέλεση ή παύση λειτουργίας) κομμάτια εφαρμογών, είτε για λόγους αναβάθμισης είτε για λόγους επέκτασης των δυνατοτήτων του.

2.10.1 ΛΕΙΤΟΥΡΓΙΑ

Η λειτουργία ενός *web server* γίνεται με τον ακόλουθο τρόπο. Ένας χρήστης (*client*) αιτείται κάποια σελίδα HTML από ένα *web server*. Το αίτημα συμπληρώνεται με τα κατάλληλα *headers*, στα οποία περιέχονται οι IP διευθύνσεις του αποστολέα και του παραλήπτη, ο αριθμός του HTTP πρωτοκόλλου καθώς και τα *ports*. Το *port* του αποστολέα είναι ένας 16 *bits* αριθμός μεγαλύτερος του 1024, ενώ το *port* του παραλήπτη (δηλαδή του *web server*) είναι το 80.

Ο *web server* παίρνοντας την αίτηση, την επεξεργάζεται και στέλνει απάντηση στην IP διεύθυνση του αποστολέα, στο *port* που του είχε αρχικά στείλει το αίτημα. Η απάντηση του *web server* δεν μπορεί να είναι τίποτε άλλο από HTML κώδικα. Ο αποστολέας λαμβάνοντας από τον *web server* μία HTML σελίδα, την παρουσιάζει με τον *browser* του. Ο κώδικας αυτός μπορεί να προέρχεται είτε από κάποιο αποθηκευμένο HTML αρχείο στο δίσκο του *web server*, είτε να παραχθεί δυναμικά από το *web server* τη στιγμή της αίτησης ανάλογα με ορισμένες παραμέτρους. Οι παράμετροι αυτές είτε συνοδεύουν την αίτηση οπότε μιλάμε για εντολές προς τον *web server*, είτε εξαρτώνται από τοπικές μεταβλητές του *server* οπότε μιλάμε για ανακοίνωση εσωτερικών δεδομένων.

Φυσικά μπορεί να υπάρχει και μία ενδιάμεση κατάσταση, κατά την οποία οι παράμετροι δίνονται από τον *client* και σκοπό έχουν να καθορίσουν την εσωτερική λειτουργία του *server*. Αυτή ακριβώς η υλοποίηση βρίσκεται πιο κοντά σ' αυτό που ονομάζουμε *embedded web server* και θα δούμε αναλυτικότερα στο 6^ο κεφάλαιο.

Οι δυναμικές αυτές σελίδες μπορούν παράγονται από το *web server* με δύο τρόπους.

α) Ο *web server* μετά την αίτηση του *client* εκτελεί μία σειρά από εξωτερικές βιβλιοθήκες οι οποίες έχουν εγκατασταθεί ως αναπόσπαστο κομμάτι του και όταν η εκτέλεση ολοκληρωθεί ο παραχθείς HTML κώδικας αποστέλλεται στον *browser* του *client* και παρουσιάζεται.

β) Ο *web server* προωθεί την αίτηση του *client* σε μία συνεργαζόμενη με αυτόν εφαρμογή. Ο *apache* ειδικά έχει τη δυνατότητα σπονδυλωτής εγκατάστασης, χάρη στην οποία, οποιαδήποτε εξωτερική εφαρμογή (αρκεί φυσικά αυτή να υποστηρίζει κάτι τέτοιο) συνεργάζεται σαν CGI (*Common Gateway Interface*) *module* με τον *apache*. Το πλεονέκτημα από αυτή τη μέθοδο είναι μετά την προώθηση του αιτήματος ο *web server* μπορεί να ασχοληθεί με την εξυπηρέτηση οποιασδήποτε άλλης αίτησης από άλλον ή από τον ίδιο *client*, ενώ όταν η εξωτερική εφαρμογή ολοκληρώσει την εκτέλεση της αίτησης αποστέλλει τον HTML κώδικα στον *apache* ο οποίος με τη σειρά του το προωθεί στον αντίστοιχο *client*. Το μειονέκτημα αυτής της μεθόδου είναι ότι στην περίπτωση που ο HTML κώδικας παράγεται από τον *web server* τότε καθ' όλη τη διάρκεια αυτής της διαδικασίας ο *web server* έχει τον έλεγχο της εκτέλεσης, ενώ στην αντίθετη περίπτωση ο έλεγχος για ένα χρονικό διάστημα παραδίδεται σε ένα εκτελέσιμο με όλους τους κινδύνους που εγκυμονεί σ' ένα σύστημα UNIX μία τέτοια κατάσταση.

ΚΕΦΑΛΑΙΟ 3

MYSQL (STRUCTURED QUERY LANGUAGE).

3.1 ΕΙΣΑΓΩΓΗ

Η MySQL είναι ένα **σύστημα διαχείρισης βάσεων δεδομένων** (DBMS - Data Base Management System). Μια βάση δεδομένων είναι μια δομημένη συλλογή στοιχείων. Μπορεί να είναι οτιδήποτε, από έναν απλό κατάλογο αγορών μέχρι και μία συλλογή εικόνων ή απέραντα ποσά στοιχείων σε ένα εταιρικό δίκτυο. Για να έχουμε την ικανότητα να προσθέσουμε, να έχουμε πρόσβαση και να μπορούμε να διαχειριζόμαστε στοιχεία που αποθηκεύονται σε μια βάση δεδομένων σε έναν υπολογιστή, χρειάζεται ένα σύστημα διαχείρισης βάσεων δεδομένων, καθώς και ένας κεντρικός υπολογιστής στο διαδίκτυο στον οποίο θα εγκατασταθεί το κατάλληλο λογισμικό έτσι ώστε να χρησιμοποιηθεί ως *web server*, δηλαδή ένας υπολογιστής ο οποίος θα αναλάβει την αποθήκευση της βάσης δεδομένων. Στην περίπτωση της παρούσας εργασίας ως *web server* χρησιμοποιήθηκε το πρόγραμμα Apache. Επίσης για τη δημιουργία και επεξεργασία της βάσης δεδομένων, χρησιμοποιήθηκε το σύστημα διαχείρισης βάσεων δεδομένων MySQL. Σημειώνεται ότι τόσο ο Apache, όσο και η MySQL επιλέχθηκαν διότι

είναι ελεύθερα, λογισμικά, τα οποία μπορεί να αποκτήσει κάποιος δωρεάν.

Δεδομένου ότι οι "υπολογιστές είναι πολύ καλοί στο~ χειρισμό πληροφοριών σε μεγάλες ποσότητες, τα συστήματα διαχείρισης βάσεων δεδομένων διαδραματίζουν έναν κεντρικό ρόλο στον κόσμο των υπολογιστών, ως αυτόνομες δραστηριότητες, ή ως μέρος άλλων εφαρμογών.

3.2 Η MYSQL ΩΣ ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Ένα σύστημα δημιουργίας και διαχείρισης βάσεων δεδομένων (όπως είναι η MySQL) αποθηκεύει τα στοιχεία σε χωριστούς πίνακες και δεν τα τοποθετεί απλά όλα μαζί σε ένα μεγάλο αποθηκευτικό χώρο. Αυτό αυξάνει την ταχύτητα και την ευελιξία της βάσης. Οι πίνακες συνδέονται με καθορισμένες σχέσεις δίνοντας έτσι τη δυνατότητα να συνδυαστούν στοιχεία από διάφορους πίνακες όταν αυτό απαιτείται. Το τμήμα SQL στο "MySQL" σημαίνει 'Δομημένη γλώσσα διατύπωσης ερωτήσεων' ή 'Structured Query Language' και είναι η πιο κοινή τυποποιημένη γλώσσα που χρησιμοποιείται για πρόσβαση στις βάσεις δεδομένων.

Η MySQL όπως όλες οι γλώσσες προγραμματισμού διακρίνεται από κάποιους τύπους δεδομένων και από κάποιες βασικές εντολές για τη δημιουργία και διαχείριση βάσεων δεδομένων. Στο συγκεκριμένο κεφάλαιο θα αναφέρουμε και θα αναλύσουμε τους τύπους δεδομένων της MySQL καθώς και τις βασικές εντολές της και σε γενικό επίπεδο αλλά και όσον αφορά τη συγκεκριμένη πτυχιακή εργασία.

3.3 ΟΙ ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ ΤΗΣ MYSQL.

Η MySQL χρησιμοποιεί πολλούς και διαφορετικούς τύπους δεδομένων, οι οποίοι ταξινομούνται σε τρεις βασικές κατηγορίες:

1. Οι αριθμητικοί τύποι δεδομένων
2. Οι τύποι ημερομηνίας και ώρας και
3. Οι αλφαριθμητικοί τύποι δεδομένων

3.3.1 ΟΙ ΑΡΙΘΜΗΤΙΚΟΙ ΤΥΠΟΙ ΑΕΔΟΜΕΝΩΝ

Σε αυτήν την ενότητα θα παρουσιαστούν οι αριθμητικοί τύποι δεδομένων. Κατά την παρουσίασή τους θα χρησιμοποιηθούν οι όροι **προσημασμένος** και **μη-προσημασμένος**. Θεωρήσαμε λοιπόν σωστό να εξηγήσουμε τις δύο αυτές έννοιες. Προσημασμένος είναι ένας θετικός ή αρνητικός ακέραιος αριθμός (δηλαδή φέρει πρόσημο), ενώ μη-προσημασμένος είναι ένας μη αρνητικός ακέραιος αριθμός. Αφού επεξηγήσαμε τις δύο αυτές έννοιες, τώρα θα παρουσιάσουμε τους αριθμητικούς τύπους δεδομένων οι οποίοι είναι οι εξής οχτώ:

- **INT** (INTF.GRR) — Είναι, ένας ακέραιος κανονικού μεγέθους ο οποίος μπορεί να είναι είτε προσημασμένος, είτε μη-προσημασμένος. Εάν είναι προσημασμένος το υποστηριζόμενο πεδίο τιμών είναι από -2.147.483.648 έως +2.147.483.648. Εάν είναι μη-προσημασμένος το υποστηριζόμενο πεδίο τιμών είναι από 0 έως 4.294.967.295. Το μέγιστο μήκος του είναι 11 ψηφία.

- **TINYINT** — Είναι ένας πολύ μικρός ακέραιος, ο οποίος όπως και ο INT μπορεί να είναι είτε προσημασμένος είτε μη-προσημασμένος, απλά αλλάζει το υποστηριζόμενο πεδίο τιμών, το οποίο για τον TINYINT είναι πολύ μικρό. Δηλαδή, εάν είναι προσημασμένος το υποστηριζόμενο πεδίο τιμών είναι από -128 έως +127 και εάν είναι μη-προσημασμένος το υποστηριζόμενο πεδίο τιμών είναι από 0 έως 255. Το μέγιστο μήκος του είναι 3 ψηφία.

- **SMALLINT** — Είναι ένας μικρός ακέραιος για τον οποίο ισχύουν ότι - και για τους INT και TINYINT. Η μόνη διαφορά είναι το υποστηριζόμενο πεδίο τιμών το οποίο εάν είναι προσημασμένος είναι από -32.768 έως +32.767 και εάν είναι μη-προσημασμένος από 0 έως 65535 και ότι το μέγιστο μήκος του αριθμού είναι 5 ψηφία.

Στην ίδια ομάδα ανήκουν και οι:

- **MEDIUMINT** (με 9 ψηφία) και
- **BIGINT** (με 11 ψηφία)
- **FLOAT (M,D)** — Είναι ένας αριθμός κινητής υποδιαστολής, ο οποίος δεν μπορεί να είναι μη-προσημασμένος. Με το (M) ορίζεται ο συνολικός αριθμός των ψηφίων (συμπεριλαμβανομένων και των δεκαδικών) και με το (D) ο αριθμός των δεκαδικών ψηφίων. Εάν δεν ορίσουμε τα δεδομένα αυτά, τότε αυτόματα παίρνουν τις τιμές 10,2. Η ακρίβεια μιας τιμής τύπου FLOAT μπορεί να φτάσει τα 24 δεκαδικά ψηφία.

- **DOUBLE (M,D)** — Είναι ένας αριθμός κινητής υποδιαστολής διπλής ακρίβειας, ο οποίος δεν μπορεί να είναι μη-προσημασμένος. Το (M) και το (D) είναι το πλήθος των συνολικών και το πλήθος των δεκαδικών ψηφίων, αντίστοιχα. Σε περίπτωση

που δεν ορίσουμε τα (M,D) τότε αυτόματα θα πάρουν τις τιμές (16,4). Η ακρίβεια μιας τιμής τύπου DOUBLE μπορεί να φτάσει τα 53 δεκαδικά ψηφία. Ο όρος REAL είναι συνώνυμος του όρου DOUBLE.

· **DECIMAL (M,D)** — Είναι ένας μη-συμπιεσμένος αριθμός κινητής υποδιαστολής, ο οποίος δε μπορεί να είναι μη-προσημασμένος. Μη-συμπιεσμένος αριθμός τύπου DECIMAL σημαίνει ότι κάθε δεκαδικό ψηφίο αντιστοιχεί σε ένα *byte*. Στον αριθμό τύπου DECIMAL τα (M,D) είναι υποχρεωτικό να οριστούν. Το NUMERIC είναι ένα συνώνυμο για τον τύπο DECIMAL.

3.3.2 ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΤΙΜΕΣ ΗΜΕΡΟΜΗΝΙΑΣ ΚΑΙ ΩΡΑΣ

Η MySQL υποστηρίζει 5 τύπους δεδομένων για τις τιμές ημερομηνίας και ώρας οι οποίοι παρέχουν ένα μεγάλο βαθμό ευελιξίας στο χρήστη που θα εισάγει δεδομένα τέτοιας μορφής. Η ευελιξία αυτή οφείλεται στο ότι η MySQL ελέγχει μόνο την εγκυρότητα δύο στοιχείων:

1. το ότι η τιμή για το μήνα ανήκει στο πεδίο τιμών 0-12 και
2. η τιμή για τον ημέρα ανήκει στο πεδίο τιμών 0-31.

Οι πέντε τύποι δεδομένων ημερομηνίας και ώρας που υποστηρίζει η MySQL είναι οι εξής:

· **DATE** — Είναι μία τιμή ημερομηνίας με τη μορφή **EEEE-MM-HH**, και δέχεται τιμές μεταξύ 1000-01-01 και 9999-12-31. Για παράδειγμα η ημερομηνία 04 Αυγούστου του 2000 θα αποθηκευτεί ως 2000-08-04.

· **DATETIME** — Είναι ένας συνδυασμός τιμών ημερομηνίας και ώρας που παίρνει τη μορφή **EEEE-MM-HH ΩΩ:ΑΑ:ΑΑ** και τις τιμές μεταξύ 1000-01-01 00:00:00 και 9999-12-31 23:59:59. Για παράδειγμα, η ώρα 9:30 το βράδυ στις 04 Αυγούστου του 2000 αποθηκεύεται ως 2000-08-04 21:30:00.

· **TIMESTAMP** — Είναι μια χρονική ένδειξη ανάμεσα στα μεσάνυχτα τις 1^{ης} Ιανουαρίου 1970 και στο έτος 2037. Με την **TIMESTAMP** έχουμε τη δυνατότητα να ορίζουμε πολλαπλά μήκη στο πεδίο τιμών της, το οποίο συσχετίζεται άμεσα με την οποιαδήποτε τιμή αποθηκεύεται μέσα σε αυτό. Το προεπιλεγμένο μήκος για τον τύπο δεδομένων της **TIMESTAMP** είναι 14 ψηφία και αποθηκεύει τιμές της μορφής **EEEEMMHHΩΩΑΑΑΑ**. Η μορφή αυτή δείχνει παρόμοια με την μορφή του τύπου της **TIMESTAMP**, χωρίς τις παύλες ανάμεσα στους αριθμούς. Για παράδειγμα έστω ότι η ώρα είναι 9:30 το βράδυ στις 04 Αυγούστου του 2000, αυτά τα δεδομένα θα αποθηκευτούν ως 20000804213000. Αλλά μήκη του πεδίου τιμών της **TIMESTAMP**

"εκτός από το προεπιλεγμένο είναι αυτό με τα 12 ψηφία που έχει μορφή **EEMMHHΩΩΛΛΑΑ**, αυτό με τα 8 ψηφία και μορφή **EEEEMMHH** και τέλος αυτό με τα 6 ψηφία και την μορφή EEMMHH.

- **TIME** — Είναι ο τύπος δεδομένων που αποθηκεύουν την ώρα σε μορφή **ΩΩ:ΑΑ:ΔΔ**.

- **YEAR(M)** — Είναι ένας τύπος δεδομένων που αποθηκεύει τα έτη σε διψήφια ή τετραψήφια μορφή. Εάν η παράμετρος του μεγέθους, δηλαδή το (M), οριστεί δύο (πχ YEAR(2)) τότε το έτος μπορεί να είναι από 1970 μέχρι και 2069 (70 έως 69), εάν όμως το (M) οριστεί τέσσερα (YEAR(4)), τότε το έτος μπορεί να είναι από 1901 έως 2155. Το 4 είναι το προεπιλεγμένο μέγεθος του τύπου YEAR(M).

Οι πιο συχνά χρησιμοποιημένοι τύποι σχετιζόμενοι με ημερομηνία και ώρα είναι συνήθως οι τύποι δεδομένων DATETIME και DATE.

Η τρίτη και τελευταία μορφή τύπων δεδομένων της MySQL είναι οι αλφαριθμητικοί τύποι δεδομένων, που είναι ίσως οι σημαντικότεροι τύποι της MySQL.

3.3.3 ΟΙ ΑΛΦΑΡΙΘΜΗΤΙΚΟΙ ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ

Οι αλφαριθμητικοί τύποι δεδομένων είναι περισσότερο σημαντικοί από τους αριθμητικούς τύπους και τους τύπους ημερομηνίας και ώρας, λόγω του ότι τα περισσότερα δεδομένα σε μια βάση αποθηκεύονται με τη μορφή αλφαριθμητικών (string). Οι πιο κοινοί τύποι δεδομένων για αλφαριθμητικά που υποστηρίζει η MySQL είναι οι εξής :

- **CHAR(M)** — Είναι ένα αλφαριθμητικό δεδομένο σταθερού μεγέθους, από 1 έως 255 χαρακτήρες. Έστω ότι έχουμε το CHAR(5)_f αυτό σημαίνει ότι σε έναν αριθμό" θα του προστίθενται κενά διαστήματα στα δεξιά μέχρι να φτάσει στο προσδιοριζόμενο μήκος. Ο ορισμός του μήκους δεν είναι απαραίτητος, αλλά το προεπιλεγμένο μήκος είναι ένας χαρακτήρας.

- **VARCHAR(M)** — Είναι ένα αλφαριθμητικό μεταβλητού μεγέθους και μήκος από 1 έως 255 χαρακτήρες. Ο ορισμός της παραμέτρου του μήκους είναι απαραίτητος όταν δημιουργούμε ένα πεδίο VARCHAR.

- **BLOB** ή **TEXT** — Εδώ αναφερόμαστε σε δύο τύπους αλφαριθμητικών δεδομένων, που και οι δύο έχουν μέγεθος έως 65.535 χαρακτήρες. Ο όρος BLOB είναι ακρωνύμιο του Binary-Large Objects, που μεταφράζεται ως μεγάλα αντικείμενα δυαδικής μορφής. Ο τύπος BLOB χρησιμοποιείται για την αποθήκευση μεγάλων ποσοτήτων δυαδικών δεδομένων, όπως για παράδειγμα εικόνες ή και άλλους τύπους αρχείων. Όσον αφορά τον τύπο TEXT και αυτός χρησιμοποιείται για να αποθηκεύονται μεγάλες ποσότητες δεδομένων. Η μόνη διαφορά μεταξύ τους είναι ότι στα δεδομένα που

αποθηκεύονται με την μορφή BLOB γίνεται διαχωρισμός ανάμεσα στους πεζούς και στους κεφαλαίους χαρακτήρες, σε αντίθεση με τα δεδομένα που αποθηκεύονται σε μορφή TEXT. Κανένας από τους δύο τύπους TEXT και BLOB, δεν απαιτεί τον ορισμό της παραμέτρου του μεγέθους. Υπάρχουν βέβαια και έτοιμοι τύποι TEXT και BLOB με καθορισμένο μήκος όπως είναι ο TINYBLOB ή TINYTEXT με μέγεθος μέχρι 255 χαρακτήρες, ο MEDIUMBLOB ή MEDIUMTEXT με μέγιστο μήκος **16.777.215** χαρακτήρες και τέλος ο **LONGBLOB** ή **LONGTEXT** με **4.294.967.295** χαρακτήρες.

· **ENUM** — Είναι μία απαρίθμηση, δηλαδή μια λίστα. Όταν ορίζουμε έναν τύπο ENUM ουσιαστικά δημιουργείται μια λίστα από στοιχεία, από τα οποία πρέπει να επιλεγθεί μια τιμή, η οποία μπορεί να είναι και η *NULL* (NULL=0/απουσία τιμής). Για παράδειγμα, εάν θέλαμε να ορίσουμε το πεδίο μας να περιέχει μία από τις τιμές 1, 2, 3, τότε θα έπρεπε να ορίσουμε NULL('1' , '2' , '3'), έτσι μόνο αυτές οι τιμές να μπορούν να εισαχθούν στο πεδίο. Ένας τύπος ENUM μπορεί να περιέχει μέχρι και **65.535** διαφορετικές τιμές. Ο τύπος ENUM ορίζεται ως δείκτης προς μία τιμή, όπως όλοι οι τύποι απαρίθμησης, σε αντίθεση με τον τύπο SET που αποθηκεύεται σαν μια πλήρης τιμή, παρά το γεγονός ότι και ο τύπος SET ορίζεται και αυτός ως λίστα.

3.4 ΟΙ ΕΝΤΟΛΕΣ ΤΗΣ MYSQL

Αφού χωρίσαμε τους τύπους δεδομένων της MySQL σε τρεις βασικές κατηγορίες και αναλύσαμε τι περιέχει η κάθε μία, τώρα πρέπει να αναφερθούμε και στο πιο σημαντικό κομμάτι του προγραμματισμού, που είναι οι εντολές μιας γλώσσας και συγκεκριμένα της MySQL. Υπάρχουν πολλές εντολές που θα μπορούσε κανείς να χρησιμοποιήσει για να δημιουργήσει, να διαχειριστεί και να επεξεργαστεί βάσεις δεδομένων. Εμείς στη συγκεκριμένη πτυχιακή εργασία θα παρουσιάσουμε τις βασικές εντολές της MySQL.

3.4.1 Η ΕΝΤΟΛΗ ΔΗΜΙΟΥΡΓΙΑΣ ΠΙΝΑΚΩΝ

Η εντολή CREATE είναι αυτή που χρησιμοποιείται για τη δημιουργία πινάκων. Για να δημιουργήσουμε όμως έναν πίνακα, θα πρέπει πρώτα να δημιουργήσουμε μία βάση δεδομένων. Αυτό μπορεί να γίνει χρησιμοποιώντας πάλι την εντολή create. Έστω για παράδειγμα, ότι επιθυμούμε να δημιουργήσουμε έναν πίνακα με διάφορα δεδομένα, θα πρέπει πρώτα να πληκτρολογήσουμε την εντολή: **create database ονομα_βασης_δεδομένων;** και μετά η MySQL θα μας εμφανίσει το μήνυμα: *Query OK, 1 row affected << 0.00 sec>>*.

Στη συνέχεια πληκτρολογώντας: `use ονομα_βασης_θεδομένων;` η MySQL θα μας εμφανίσει το μήνυμα: **Database changed**. Μετά και από αυτή την πολύ σημαντική εγγραφή είμαστε πλέον έτοιμοι να δημιουργήσουμε τον πίνακα.

Η βασική μορφή της εντολής για τη δημιουργία πινάκων είναι: **create table όνομα_πίνακα {όνομα_ πεδίου τύπος_πεδίου,...}**; Το όνομα του πίνακα είναι προσωπική επιλογή του καθενός, αλλά για μεγαλύτερη ευκολία καλό θα ήταν να χρησιμοποιηθεί κάποιο όνομα που να φανερώνει τη χρήση-ή το ρόλο του πίνακα. Το ίδιο ισχύει και για τα ονόματα των πεδίων του πίνακα, δηλαδή για το όνομα της κάθε στήλης, θα πρέπει τα ονόματα να είναι περιγραφικά. Για παράδειγμα, εάν θέλουμε να φτιάξουμε μία βάση δεδομένων για ένα ταξιδιωτικό γραφείο, όπως στη συγκεκριμένη εργασία, καλό θα ήταν τον πίνακα να τον ονομάσουμε **Tours** και κάποιο από τα πεδία, που θα δείχνει τους προορισμούς, να το ονομάσουμε **destinations**. Δημιουργείται, λοιπόν, ένας πίνακας με το όνομα **Tours**, που περιείχε πέντε πεδία, το **id**, το **destinations**, το **departure**, το **duration** και το **price**.

Το πεδίο με τον κωδικό **id** ορίζεται ως **προτεύων κλειδί** (primary key) και είναι ένα δεδομένο που απαριθμεί τις γραμμές των υπόλοιπων δεδομένων, για αυτό γράφουμε και το **auto_increment** (αυτόματη αύξηση τιμής), έτσι ώστε η MySQL να δίνει σε κάθε νέο δεδομένο τον επόμενο διαθέσιμο αριθμό. Ο MySQL server ανταποκρίνεται με το μήνυμα Query OK κάθε φορά που εκτελείται με επιτυχία μια εντολή. Σε κάθε άλλη περίπτωση εμφανίζεται ένα μήνυμα σφάλματος.

Το προτεύων κλειδί, δηλαδή το **id**, είναι ένα πεδίο που προσθέτουμε στους πίνακες που δημιουργούμε οπου-σε κάθε σειρά να εισάγετε ένας αριθμός μοναδικός ώστε να ξεχωρίζουν αντίστοιχα μοναδικά οι εγγραφές του πίνακα. Οι εγγραφές είναι τα στοιχεία που εισάγουμε στον πίνακα και στην συγκεκριμένη περίπτωση είναι κάποια ταξίδια τα οποία είναι προσφορές που κάνει το ταξιδιωτικό γραφείο που έχουμε δημιουργήσει στους πελάτες του και αφορούν δέκα διαφορετικούς προορισμούς. Σε μία γραμμή του πίνακα θα απεικονίζονται ο αριθμός **id** της εγγραφής, ο προορισμός, η ημερομηνία αναχώρησης, η διάρκεια του ταξιδιού και η τιμή του.

Η Εντολή Εισαγωγής Δεδομένων (INSERT)

Αφού δημιουργήσαμε τον πίνακά μας με την εντολή **create** πρέπει στη συνέχεια να του προσθέσουμε κάποια δεδομένα, δηλαδή εγγραφές. Καλό βέβαια θα ήταν να αναφέρουμε ότι γράφοντας στην MySQL `show fields front Tours;`, εμφανίζεται ένας πίνακας που μας δείχνει πόσα πεδία και με τι είδους δεδομένα έχουμε δημιουργήσει. Η εντολή που θα χρησιμοποιήσουμε για να εισάγουμε δεδομένα στον πίνακά μας είναι η **INSERT** και η γενική σύνταξη της είναι η:

insert into όνομα _πίνακα(λίστα πεδίων) values (τιμές πεδίων); Όταν εισάγουμε ένα αλφαριθμητικό δεδομένο πρέπει πάντα να το γράφουμε μέσα σε εισαγωγικά. Εάν το αλφαριθμητικό αυτό περιέχει και από μόνο του εισαγωγικά τότε θα πρέπει εκτός από τα εισαγωγικά που θα του βάλουμε, να χρησιμοποιήσουμε και τον χαρακτήρα \ πριν από κάθε εισαγωγικό που περιέχει το αλφαριθμητικό. Για παράδειγμα, εάν θέλουμε να καταχωρήσουμε το αλφαριθμητικό Mike said "Hello ", θα πρέπει να το εισάγουμε γράφοντας:

```
insert into table_name (column_name) values ('Mike said \"Hello\~');
```

Όταν χρησιμοποιούμε την εντολή insert δύο είναι τα βασικά της μέρη, οι στήλες και οι τιμές- Απαραίτητη για να χρησιμοποιήσουμε την εντολή insert είναι η λίστα των τιμών, ενώ εάν παραλείψουμε τη λίστα των στηλών δεν δημιουργείται πρόβλημα. Το μόνο που-χρειάζεται να προσέξουμε είναι ότι πρέπει-να δηλώσουμε τα δεδομένα με τη σειρά που έχουμε εισάγει και τα πεδία. Δηλαδή, μπορούμε να εισάγουμε κάποια δεδομένα είτε γράφοντας:

```
insert into Tours (id, destinations, departure, duration, price) values (1  
England'2005-01-10'5', 1500');
```

είτε πάλι γράφοντας αποκλειστικά και μόνο τα δεδομένα που θέλουμε να εισάγουμε, δηλαδή:

```
insert into Tours values ('1' England'2005-01-10', '5', 1500');
```

Και στις δύο περιπτώσεις το μήνυμα που θα μας εμφανίσει η MySQL θα πρέπει να είναι το ίδιο: Query OK, 1 row affected (0.01 sec) .

Όπως είδαμε προηγουμένως, δημιουργήθηκε μία βάση δεδομένων με το όνομα Dimva_Travel_Agency, αφού ασχολούμαστε με ταξιδιωτικό γραφείο, και έναν πίνακα με το όνομα Tours, ο οποίος αποτελείται από πέντε πεδία. Τα πέντε αυτά πεδία είναι το id που όπως έχω αναφέρει είναι το πρωτεύων κλειδί, είναι το πεδίο destination, που περιέχει τους προορισμούς των

ταξιδιών-προσφορών, το πεδίο departure, που περιέχει τις ημερομηνίες αναχώρησης των ταξιδιών, το πεδίο duration το οποίο μας εμφανίζει την διάρκεια παραμονής στον -κάθε προορισμό και τέλος το πεδίο price το οποίο περιέχει-τις τιμές των προσφορών- αυτών.

Χρησιμοποιώντας την εντολή insert προσθέσαμε δεδομένα στον πίνακα αυτό που αφορούσαν σε δέκα ταξιδιωτικές προσφορές του ταξιδιωτικού γραφείου.

3.4.2 Η ΕΝΤΟΛΗ SELECT

Η εντολή SELECT είναι μια εντολή της MySQL η οποία χρησιμοποιείται για την **ανάκτηση** δεδομένων από ένα πίνακα. Η εντολή select μπορεί να γίνει από πολύ απλή μέχρι αρκετά πολύπλοκη. Η γενική

της σύνταξη, και η πιο απλουστευμένη από τις υπόλοιπες, είναι η: **select εκφράσεις_και_πεδία from όνομα_πίνακα;**

Δεν είναι αναγκαίο κάθε φορά που επιθυμούμε να δούμε τα δεδομένα του πίνακα να γράφουμε την εντολή ολόκληρη, ένα πολύ χρήσιμο και βολικό σύμβολο είναι το * (αστερίσκος) το οποίο -αντιπροσωπεύει τα πάντα.- Οπότε-εάν θέλουμε-να διαλέξουμε τα πάντα από-έναν- πίνακα αρκεί να γράψουμε την εντολή με την εξής μορφή: **select * from Tours;** Η εντολή select έχει επίσης τη δυνατότητα να ταξινομεί τα αποτελέσματα που μας εμφανίζει, που μας επιστρέφει στην οθόνη.

Υπάρχουν και κάποια όρισμα τα οποία βοηθούν την εντολή select στο να εμφανίσει με καλύτερο τρόπο τα δεδομένα ενός πίνακα.

- Το όρισμα ORDER-BY-μας βοηθάει να ταξινομήσουμε-τις εγγραφές – μας – σε – κατηγορίες, δηλαδή να εμφανίζονται κατά destinations ή price ή και τις υπόλοιπες στήλες.

Όπως μπορούμε να διακρίνουμε αυτή η μορφή της εντολή select μας εμφανίζει τους προορισμούς κατά αλφαβητική σειρά και τις τιμές από τη μικρότερη προς τη μεγαλύτερη.

- Υπάρχει ακόμη το όρισμα WHERE το οποίο ακολουθούμενο από μία συνθήκη μπορεί να συγκεκριμενοποιήσει περισσότερο τις πληροφορίες που θα εμφανίζονται στην οθόνη της MySQL, δηλαδή ενώ ως τώρα μάθαμε να κατηγοριοποιούμε κατά στήλη, τώρα η φράση αυτή θα μας βοηθήσει να εμφανίσουμε μόνο όποιες γραμμές επιθυμούμε.

Στο όρισμα where μπορούν να χρησιμοποιηθούν κάποια σύμβολα και κάποιοι συνδυασμοί τους τα οποία ονομάζονται τελεστές και μας βοηθούν να συγκεκριμενοποιήσουμε το where ακόμα πιο πολύ. Συγκεκριμένα υπάρχουν δύο κατηγορίες τελεστών, οι οποίες είναι οι τελεστές **σύγκρισης** και οι **λογικοί** τελεστές.

Υπάρχει επίσης ο τελεστής **between** ο οποίος είναι χρήσιμος για συγκρίσεις ακεραίων και εμφανίζει τα αποτελέσματα μεταξύ μιας ελάχιστης και μιας μέγιστης τιμής.

Όσον αφορά τους λογικούς τελεστές, οι βασικοί είναι ο AND (σύζευξη - ΚΑΙ) και ο OR _ (διάζευξη - Ή). Η διαφορά των δύο αυτών τελεστών είναι ότι όταν χρησιμοποιούμε τον τελεστή AND πρέπει οι συγκρίσεις της πρότασης where να αποτιμηθούν σαν αληθείς, ενώ με τον τελεστή OR απαιτείται να είναι αληθής τουλάχιστον μία από τις συγκρίσεις της πρότασης where. Χρησιμοποιώντας την πρόταση where μπορούμε επίσης να συγκρίνουμε αλφαριθμητικά δεδομένα. Πολύ χρήσιμος τελεστής για την ιδιότητα αυτή της where είναι ο **like** ο οποίος χρησιμοποιεί και δύο σύμβολα μπαλαντέρ που είναι το %-και το _ . Το _ σημαίνει ότι αυτό που ψάχνουμε ταιριάζει με έναν ακριβώς χαρακτήρα μέσα στον πίνακα με τα δεδομένα μας, ενώ το % σημαίνει ότι ταιριάζει με πολλαπλούς χαρακτήρες από τα δεδομένα μας.

3.4.3 Η ΕΝΤΟΛΗ UPDATE

Η εντολή UPDATE είναι μια εντολή της MySQL η οποία χρησιμοποιείται για την τροποποίηση του περιεχομένου ενός ή και περισσότερων πεδίων στα δεδομένα ενός πίνακα.

Από τις πιο απλές συντάξεις της update είναι, εάν για παράδειγμα έχουμε δημιουργήσει τον πίνακα και θέλουμε να αλλάξουμε ΟΛΕΣ τις ημερομηνίες αναχώρησης, η:

```
update Tours set departure = '2005-01-11
```

Με την εντολή update όμως μπορούμε να κάνουμε και **επιλεγμένη** τροποποίηση των δεδομένων απλά χρησιμοποιώντας, όπως και στην εντολή select, το όρισμα where. Οπότε η εντολή update, εάν θέλουμε για παράδειγμα να αλλάξουμε έναν προορισμό μπορεί να πάρει τη μορφή:

```
update Tours set destinations = 'United Kingdom' where destinations = 'England';
```

3.4.4 Η ΕΝΤΟΛΗ JOIN

Υπάρχουν πολλοί τρόποι ενώσεων στη MySQL, όλοι εκ των οποίων αναφέρονται στη σειρά με την οποία ενώνονται/συσχετίζονται διαφορετικοί πίνακες μαζί και τη σειρά με την οποία εμφανίζονται τα αποτελέσματα. Ο τύπος της ένωσης ο οποίος μάθαμε, αλλά δεν χρησιμοποιήσαμε στη συγκεκριμένη εργασία για τον λόγο του ότι δεν απαιτήθηκε κατά την υλοποίηση, ενώνει δύο πίνακες μαζί. Για παράδειγμα, έστω ότι είχαμε δημιουργήσει έναν πίνακα με ονόματα φρούτων και ένα με χρώματα, οι οποίοι και οι δύο αποτελούνται από δύο πεδία, τα χρώματα ή τα φρούτα αντίστοιχα και με τον αύξοντα αριθμό των εισαγωγών (id), όπως φαίνονται και στο σχέδιο που ακολουθεί.

id	Fruitname
1	apple
2	orange
3	grape
4	banana

Πίνακας
Φρούτων

id	Colorname
1	red
2	orange
3	purple
4	yellow

Πίνακας
Χρωμάτων

Η ένωση που θα χρησιμοποιήσουμε λέγεται εσωτερική ένωση (inner join) και η γενική σύνταξή της είναι αυτή που απεικονίζεται στο παρακάτω σχέδιο.

```
mysql select fruitname, colorname from fruit inner join color ~> on fruit.id = color.id;
```

Όπως μπορούμε να διακρίνουμε και από το σχέδιο, με την εντολή αυτή, ενώσαμε, το πεδίο με τα ονόματα των φρούτων, με το πεδίο με τα χρώματα, σύμφωνα με τον αύξοντα αριθμό τους. Εκτός από την inner join υπάρχουν και οι left join, right join, cross join, straight join και natural join.

3.4.5 Η ΕΝΤΟΛΗ REPLACE

Η εντολή **REPLACE** είναι άλλη μία εντολή τροποποίησης πινάκων της MySQL, η οποία όμως είναι μια συνδυασμένη χρήση της εντολής **delete** και της εντολής **insert**. Δηλαδή, χρησιμοποιώντας την εντολή **replace** διαγράφουμε μια εγγραφή και την αντικαθιστούμε εισάγοντας μια άλλη εγγραφή. Η σύνταξη της **replace** είναι της μορφής:
replace into όνομα_πίνακα values (τιμές στηλών);

3.4.6 Η ΕΝΤΟΛΗ DELETE

Η τελευταία εντολή της MySQL που θα αναφέρουμε είναι η εντολή **DELETE**, της οποίας η γενική σύνταξη και ποιο απλοποιημένη είναι η:
delete from πίνακα;

Η εντολή με τη μορφή αυτή διαγράφει **όλες** τις εγγραφές του πίνακα που δημιουργήσαμε. Μπορούμε όμως να χρησιμοποιήσουμε και το όρισμα *where*, έτσι ώστε να συγκεκριμενοποιήσουμε τη διαγραφή που θέλουμε να εκτελέσουμε στα δεδομένα του πίνακά μας. Για παράδειγμα, η ***delete from όνομα_πίνακα where όνομα_στήλης = 'εγγραφή'*** διαγράφει τα αποτελέσματα που εμείς ορίσαμε. Γράφοντας την εντολή με αυτή τη μορφή σε περιβάλλον MySQL, διαγράφεται από τον πίνακά μας η εγγραφή, δηλαδή στην συγκεκριμένη εργασία το ταξίδι, που η τιμή του είναι 700 €

ΚΕΦΑΛΑΙΟ 4 PHP

4.1 ΕΙΣΑΓΩΓΗ

Η PHP, της οποίας τα αρχικά σήμερα αντιπροσωπεύουν το "PHP: Hypertext Preprocessor" είναι μια ευρέως χρησιμοποιούμενη, ανοιχτού κώδικα, γενικού σκοπού server-side scripting γλώσσα προγραμματισμού, η οποία είναι ειδικά κατάλληλη για ανάπτυξη εφαρμογών για το Web και μπορεί να ενσωματωθεί στην HTML. Η PHP επιτρέπει στους χρήστες της να δημιουργήσουν **δυναμικές** ιστοσελίδες, δηλαδή ιστοσελίδες οι οποίες επιτρέπουν πρόσβαση σε βάσεις δεδομένων. Οι σελίδες οι οποίες έχουν στο περιεχόμενο τους κώδικα PHP συμπεριφέρονται όπως και οι HTML σελίδες. Στο συγκεκριμένο κεφάλαιο θα μιλήσουμε γενικά για την PHP, τις μεταβλητές της και τις εντολές της, και στο επόμενο κεφάλαιο θα αναλύσουμε τον κώδικα της συγκεκριμένης πτυχιακής εργασίας βήμα προς βήμα, έτσι ώστε να γίνει πιο κατανοητός.

4.2 ΣΥΝΤΟΜΟ ΙΣΤΟΡΙΚΟ ΤΗΣ PHP.

Η PHP προέρχεται από ένα παλαιότερο προϊόν, ονομασμένο ως PHP/FI. Η PHP/FI δημιουργήθηκε από τον Rasmus Lerdorf το 1994. Αρχικά τη χρησιμοποιούσε για τη δημιουργία απλών scripts τύπου Perl ώστε να μπορεί να δημιουργήσει μια απλή σελίδα στο διαδίκτυο. Η Perl είναι μια γενικής χρήσης γλώσσα προγραμματισμού η οποία αρχικά δημιουργήθηκε για την επεξεργασία κειμένου και τώρα πλέον έχει φτάσει στο σημείο να χρησιμοποιείται για διάφορες εργασίες όπως η διαχείριση συστημάτων, η δημιουργία ιστοσελίδων, ο προγραμματισμός δικτύων και πολλές ακόμα εργασίες. Αυτά τα scripts που δημιούργησε ο Rasmus, τα ονόμασε PHP Tools δηλαδή "Personal Home Page Tools".

Λόγω της ανάγκης για περισσότερη λειτουργικότητα, ο Rasmus έγραψε μια πολύ μεγαλύτερη εφαρμογή σε γλώσσα προγραμματισμού C, που ήταν σε θέση να επικοινωνήσει με βάσεις δεδομένων, και επέτρεψε με αυτόν τον τρόπο στους χρήστες να αναπτύξουν απλές δυναμικές εφαρμογές (απλές σελίδες) στο διαδίκτυο. Ο Rasmus αφού βελτίωσε τον κώδικά του όσο μπορούσε, μετά επέλεξε να το δημοσιεύσει, έτσι ώστε όλοι να μπορούν να το βλέπουν και οποιοσδήποτε να μπορεί να το χρησιμοποιήσει, καθώς επίσης και να μπορεί να διορθώνει κάποια λάθη (bugs), με αποτέλεσμα δηλαδή να βελτιωθεί ο κώδικας.

Το PHP/FI ως όρος ερμηνεύεται Personal Home Page/Forms. Interpreter, και περιλαμβάνει μερικές από τις βασικές λειτουργίες της PHP όπως τη γνωρίζουμε με τη σημερινή της μορφή. Η PHP/FI

αποτελούνταν από μεταβλητές τύπου Perl, μεταβλητές αυτόματης ερμηνείας φορμών καθώς και σύνταξη τέτοιας μορφής που της επέτρεπε να ενσωματωθεί σε HTML κώδικα.

Μέχρι το 1997, η PHP/FI 2.0, που ήταν η δεύτερη έκδοση της εφαρμογής PHP/FI, υπολογίζεται ότι είχε αρκετές χιλιάδες χρήστες σε όλο τον κόσμο, με περίπου 50.000 σελίδες του διαδικτύου να ισχυρίζονται πως την έχουν εγκαταστήσει, και αυτό το νούμερο αντιστοιχεί περίπου στο 1% των σελίδων του διαδικτύου. Παρά το γεγονός ότι υπήρξαν διάφοροι άνθρωποι που συνέβαλλαν στη βελτίωση του προγράμματος με μερικά μικρά κομμάτια κώδικα, το πρόγραμμα παρέμενε ακόμα στο μεγαλύτερο μέρος του δουλειά ενός ανθρώπου. Στην συνέχεια δημιουργήθηκε η PHP/FI 3.0 και εν τέλει η γνωστή PHP όπως είναι με τη σημερινή της μορφή.

4.3 ΟΙ ΕΝΤΟΛΕΣ ΤΗΣ PHP.

4.3.1 ΜΕΤΑΒΛΗΤΕΣ

Η PHP όπως και όλες οι γλώσσες προγραμματισμού αποτελείται από μεταβλητές (variables). Μια μεταβλητή είναι μια ειδική θέση την οποία μπορούμε να ορίσουμε εμείς για την αποθήκευση μιας τιμής . Οι μεταβλητές έχουν ζωτική σημασία για τον προγραμματισμό, γιατί χωρίς αυτές θα ήμασταν υποχρεωμένοι να ενσωματώνουμε στον κώδικά μας (script) όλες τις τιμές που χρειαζόμαστε. Οι μεταβλητές μας δίνουν τη δυνατότητα να δημιουργούμε πρότυπα, ορισμούς και συναρτήσεις για την εκτέλεση διάφορων ενεργειών, όπως για παράδειγμα την πρόσθεση, την αφαίρεση, τον πολλαπλασιασμό και τη διαίρεση δύο ή και περισσότερων αριθμών, καθώς και διάφορες άλλες ενέργειες χωρίς να ανησυχούμε για τις τιμές που περιέχουν οι μεταβλητές. Οι τιμές αντιτίθεται στις μεταβλητές μέσω των επιλογών που κάνει ο χρήστης της ιστοσελίδας και προέρχεται από μια ή και περισσότερες βάσεις δεδομένων. Μια μεταβλητή σύμφωνα με την PHP αποτελείται από ένα όνομα της επιλογής μας και μπροστά από το όνομα αυτό μπαίνει σαν πρόθεμα το σήμα του δολαρίου (\$). Για παράδειγμα, ας υποθέσουμε ότι θέλουμε να ορίσουμε μια μεταβλητή για κάποιους προορισμούς κάποιων ταξιδιών. Η μεταβλητή αυτή μπορεί να έχει τη μορφή *Sdestinations*. Τα ονόματα των μεταβλητών - μπορούν να περιλαμβάνουν γράμματα, αριθμούς και το χαρακτήρα της κάτω παύλας (_). Οι μεταβλητές δε μπορούν να έχουν διαστήματα, για αυτό το λόγο χρησιμοποιείται και η κάτω παύλα. Επίσης σε μια μεταβλητή πρέπει το όνομα να ξεκινάει από ένα γράμμα ή από το χαρακτήρα της κάτω παύλας, άρα μια λίγο πιο

ολοκληρωμένη μεταβλητή στο προηγούμενο παράδειγμα θα έπαιρνε τη μορφή `$destination_names_1`.

Για να τερματιστεί μια εντολή της PHP χρησιμοποιείται ο χαρακτήρας του ελληνικού ερωτηματικού (;) και για αυτό το λόγο δε μπορεί να συμπεριληφθεί μέσα στο όνομα μιας μεταβλητής.

Μια μεταβλητή έχει δυο μέρη, το όνομα της μεταβλητής και τις τιμές που παίρνει, τα οποία ενώνονται με διάφορους *τελεστές* όπως είναι το ίσον (=) για να ορίσουν τη σχέση μιας μεταβλητής με την τιμή της. Ας υποθέσουμε λοιπόν, ότι έχουμε μια μεταβλητή με το όνομα *X* η οποία ισούται με 5, για να την προσθέσουμε στον κώδικα μας, αρκεί να γράψουμε `$X=5;`. Για να εμφανίσουμε στην οθόνη μας την τιμή της μεταβλητής, αρκεί να γράψουμε την εντολή `print $X;` και στην οθόνη μας θα εμφανιστεί μόνο ο αριθμός 5.

Οι μεταβλητές επομένως, αποτελούνται από το όνομα της μεταβλητής το οποίο και αναλύσαμε, τους τύπους δεδομένων που είναι οι τιμές που δέχονται οι μεταβλητές και τους τελεστές τους ποιους θα αναλύσουμε σε επόμενη ενότητα.

4.3.2 ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ ΤΗΣ PHP

Διαφορετικοί τύποι δεδομένων δεσμεύουν διαφορετικά ποσά μνήμης και αντιμετωπίζονται ανάλογα με το πώς θα τους ορίσουμε σε κάθε script. Μερικές γλώσσες προγραμματισμού απαιτούν από τον προγραμματιστή να δηλώνει εξαρχής τους τύπους δεδομένων που θα αποθηκεύει μια μεταβλητή, σε αντίθεση με την PHP η οποία είναι λίγο πιο ελαστική και εξακριβώνει τους τύπους δεδομένων όταν αυτοί ανατίθενται σε μια μεταβλητή. Βέβαια δε θα πρέπει να κάνουμε κατάχρηση της ιδιότητας αυτής, διότι σε ένα μεγάλο script αν αλλάξουμε τους τύπους δεδομένων που δέχεται μια μεταβλητή θα δημιουργηθούν σφάλματα στον κώδικα.

4.3.3 ΤΕΛΕΣΤΕΣ ΚΑΙ ΕΚΦΡΑΣΕΙΣ ΤΗΣ PHP

Ένας *τελεστής* (operator) είναι ένα σύμβολο ή μια σειρά συμβόλων τα οποία όταν χρησιμοποιούνται σε συνδυασμό με τιμές, εκτελούν μια ενέργεια και συνήθως παράγουν μια νέα τιμή. Ο τελεστής μπορεί να τοποθετηθεί ανάμεσα σε δυο τιμές και ανάμεσα από μεταβλητή και τιμή. Όπως μπορούμε να διακρίνουμε, οι τιμές έχουν άμεση σχέση με τους τελεστές και ονομάζονται *τελεστέοι*. Στην πράξη τώρα, αν για παράδειγμα, θέλουμε να προσθέσουμε το 3 και το 2, οι τιμές (τελεστέοι) θα είναι αντίστοιχα το 3 και το 2 ενώ ο τελεστής θα είναι το σύμβολο της πρόσθεσης (+) και μέσω αυτής της εντολής θα πάρουμε

μια νέα τιμή που στην προκειμένη περίπτωση είναι το 5. Ο συνδυασμός των τελεστών με έναν τελεστή για την παραγωγή ενός αποτελέσματος ονομάζεται **έκφραση** (expression). Οι πιο χρησιμοποιούμενοι τελεστές είναι;

1. Τελεστής εκχώρησης τιμής

Ο τελεστής αυτός αντιπροσωπεύεται από το σύμβολο ίσον (=) και παίρνει την τιμή του δεξιού τελεστέου του και τον εκχωρεί στον αριστερό. Για παράδειγμα, έστω ότι έχουμε την έκφραση $\$destination$ - "England";. Αυτή η έκφραση σημαίνει ότι η μεταβλητή $Sdestination$ περιέχει το αλφαριθμητικό *England*.

2. Αριθμητικοί τελεστές

Οι αριθμητικοί τελεστές εκτελούν αριθμητικές πράξεις και είναι οι εξής:

+	Πρόσθεση	Π.χ. 10+4	14
-	Αφαίρεση	Π.χ. 10-4	6
/	Διαίρεση	Π.χ. 10/4	2,5
*	Πολλαπλασιασμός	Π.χ. 10*4	40
%	Ακέραιο Υπόλοιπο	Π.χ. 10%4	2

Η πιο ιδιόμορφη περίπτωση είναι αυτή του τελεστή ακέραιου υπόλοιπου (modulus) ο οποίος επιστρέφει το υπόλοιπο της διαίρεσης του αριστερού τελεστέου με του δεξιού.

3. Τελεστής συνένωσης

Ο τελεστής **συνένωσης** (concatenation) αντιπροσωπεύεται από το σύμβολο της τέλειας (.) και αντιμετωπίζει, και τους δυο τελεστέους σαν αλφαριθμητικά, προσαρτώντας το δεξιό στο τέλος του αριστερού. Για παράδειγμα, έστω ότι θέλουμε να εμφανιστεί στην οθόνη μας η έκφραση *HELLO WORLD*, αρκεί να γράψουμε την εντολή "*HELLO WORLD* ";

4. Σύνθετοι τελεστές εκχώρησης τιμής

Η PHP υποστηρίζει μερικούς σύνθετους τελεστές οι οποίοι μεταβάλλουν τον αριστερό τελεστέο τους και επιστρέφουν ένα αποτέλεσμα. Οι περισσότερες γλώσσες προγραμματισμού έχουν μόνο έναν τελεστή που είναι το ίσον (=), η PHP όμως κάνει την εξαίρεση. Οι σύνθετοι τελεστές απαρτίζονται από τους αριθμητικούς τελεστές και τον τελεστή συνένωσης που στο τέλος έχουν τον τελεστή ίσον (=). Ο πίνακας που ακολουθεί κάνει μια πιο λεπτομερή προσέγγιση των σύνθετων τελεστών.

+=	π.χ. \$X += 4	Ισοδυναμεί με \$X = \$X + 4 →	8
-=	π.χ. \$X -= 4	Ισοδυναμεί με \$X = \$X - 4 →	0
/=	π.χ. \$X /= 4	Ισοδυναμεί με \$X = \$X / 4 →	-1
=	" π.χ. \$X 4	Ισοδυναμεί με \$X = \$X * 4 →	16
%=	π.χ. \$X %= 4	Ισοδυναμεί με \$X = \$X % 4 →	0
=	π.χ. \$X .= "test"	Ισοδυναμεί με \$X = \$X . "test" →	"testtest"

5. Τελεστές σύγκρισης

Οι τελεστές σύγκρισης όπως υποδηλώνει και το όνομα τους συγκρίνουν δυο τιμές και επιστρέφουν σαν αποτέλεσμα τύπους δεδομένων της μορφής Boolean, δηλαδή **true** αν η σύγκριση είναι αληθής, ή **false** σε κάθε άλλη περίπτωση.

6. Λογικοί τελεστές

Οι **λογικοί** τελεστές ελέγχουν συνδυασμούς λογικών τιμών και επιστρέφουν και αυτοί όπως και οι τελεστές σύγκρισης αποτέλεσμα τύπου Boolean

4.3.4 ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΤΩΝ ΤΕΛΕΣΤΩΝ

Κατά τη χρήση ενός τελεστή η PHP, αυτός διαβάζει την έκφραση από αριστερά προς τα δεξιά. Ωστόσο για εκφράσεις που περιέχουν πάνω από έναν τελεστή τα πράγματα δυσκολεύουν. Αν δημιουργήσουμε μια έκφραση του τύπου $3+2$ τότε είναι ξεκάθαρη η πράξη που πρέπει να εκτελεστεί. Όταν όμως έχουμε μια έκφραση της μορφής $3+2*5$ δημιουργείται πρόβλημα γιατί αν η PHP εκτελέσει την πράξη αυτή από τα αριστερά προς τα δεξιά θα βγει το αποτέλεσμα 25 που είναι λάθος. Για το λόγο αυτό η PHP χρησιμοποιεί τους τελεστές κατά προτεραιότητα, αν και θα ήταν πολύ πιο λειτουργικό να

χρησιμοποιούσαμε παρενθέσεις για την αποφυγή σφαλμάτων σε πιο δύσκολες εκφράσεις.

4.3.5 ΣΤΑΘΕΡΕΣ

Οι μεταβλητές, όπως προαναφέραμε, είναι ένας τρόπος αποθήκευσης δεδομένων. Οι μεταβλητές έχουν όμως την ευελιξία ότι μπορούμε να αλλάξουμε τις τιμές και τον τύπο δεδομένων τους. Εάν λοιπόν θέλαμε να εργαστούμε με μια τιμή για όλη τη διάρκεια ενός script τότε μπορούμε να χρησιμοποιούμε μια **σταθερά** (constant). Για τη δημιουργία μιας σταθεράς πρέπει να χρησιμοποιούμε την συνάρτηση **define**(). Στην παρένθεση τοποθετούμε το όνομα και την τιμή της σταθεράς, τα οποία θα χωρίζονται μεταξύ τους μετένα κόμμα. Σε αντίθεση με τη μεταβλητή, δε χρειάζεται το πρόθεμα του δολαρίου (\$), συνεπώς αν θέλουμε να ορίσουμε μια σταθερά με το όνομα *destination name* και την τιμή *England* αρκεί να γράψουμε:

```
define ('DESTINATION_NAME', 'ENGLAND');
```

Η τιμή της σταθεράς μπορεί να είναι ένας αριθμός, ένα αλφαριθμητικό, ή μια λογική (Boolean) τιμή, και το όνομα της πρέπει να αναγράφεται με όλους τους χαρακτήρες κεφαλαίους.

Ας δούμε λοιπόν πως μπορούμε να ορίσουμε μια σταθερά μέσω ενός script.

```
1      <html>
2      <head>
3      <title>Defining a Constant </title>
4      </head>
5      <body>
6  <?php
7          define ("Destination_Name", "England");
8          print, "Welcome to" .Destination_Name;
9          ?>
10     </body>
11     </html>
```

Ορισμός μιας σταθεράς

Στη γραμμή 8 του πιο πάνω σχήματος, χρησιμοποιήσαμε τον τελεστή συνένωσης για να προσαρτήσουμε την τιμή της σταθεράς στο αλφαριθμητικό *Welcome to* και αυτό γίνεται για το λόγο ότι ο μηχανισμός εκτέλεσης της PHP δεν έχει καμία λειτουργία που να κάνει διαχωρισμό μεταξύ μιας σταθεράς και ενός αλφαριθμητικού που περικλείεται σε εισαγωγικά. Εάν αποθηκεύσουμε αυτό το script με το όνομα *define.php* στο φάκελο έγγραφων του Web server μας, στη συγκεκριμένη περίπτωση στο φάκελο **htdocs** του Apache, και τον

προσπελάσουμε μέσω του browser, στην οθόνη μας θα εμφανιστεί το κείμενο: *Welcome to England*.

Σημειώνουμε, ότι κάθε script σε PHP αρχίζει και τελειώνει με τα `<?php` και `?>` αντίστοιχα (γραμμές 6 και 9).

ΚΕΦΑΛΑΙΟ 5

ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ

5.1 ΤΙ ΕΙΝΑΙ ΕΝΑ ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ

CMS ή Σύστημα Διαχείρισης Περιεχομένου (Content Management System) ονομάζουμε ένα λογισμικό που μας βοηθά να ελέγχουμε και να διαχειριζόμαστε έναν ιστότοπο δημόσιας ή περιορισμένης πρόσβασης. Περιεχόμενο είναι οτιδήποτε αποφασίζουμε να «ανεβάσουμε» στον ιστότοπό μας: κείμενα, φωτογραφίες, μουσική, βίντεο, έγγραφα ή οποιουδήποτε άλλου είδους αρχείο. Ένα ιδανικά σχεδιασμένο CMS μας επιτρέπει να διαχειριζόμαστε τον ιστότοπό μας χωρίς να διαθέτουμε περίπλοκες τεχνικές γνώσεις ή δεξιότητες.

Ένα CMS μπορεί να υποστηρίζει τα ακόλουθα χαρακτηριστικά γνωρίσματα:

- αναγνώριση όλων των βασικών χρηστών και των ρόλων τους στη διαχείριση περιεχομένου δυνατότητα ανάθεσης ρόλων και ευθυνών σε διαφορετικούς τύπους και κατηγορίες περιεχομένου
- καθορισμός των ενεργειών ροής εργασιών για συνεργατική δημιουργία, συχνά σε συνδυασμό με μηνύματα γεγονότων, έτσι ώστε οι διαχειριστές να ενημερώνονται για αλλαγές στο περιεχόμενο (π.χ., ένας δημιουργός περιεχομένου υποβάλλει μια ιστορία, η οποία δημοσιεύεται, μόνο αφού αναθεωρηθεί από το συντάκτη αντιγράφων και εγκριθεί από τον αρχισυντάκτη.
- δυνατότητα παρακολούθησης και διαχείρισης πολλαπλών εκδόσεων του ίδιου περιεχομένου
- δυνατότητα σύλληψης περιεχομένου (π.χ. σάρωση)
- δυνατότητα έκδοσης του περιεχομένου σε αποθήκη (repository), ώστε να υποστηριχθεί η πρόσβαση στο περιεχόμενο (όλο και περισσότερο, η αποθήκη θα αποτελεί έμφυτο μέρος του συστήματος, ενσωματώνοντας επιχειρηματικές αναζητήσεις και ανακτήσεις.)
- χωρισμός της σημασιολογίας του περιεχομένου από τη δομή του (π.χ., το CMS μπορεί αυτόματα να καθορίσει το χρώμα ή τη γραμματοσειρά του κειμένου).

5.2 ΤΙ ΕΙΝΑΙ ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ WEB

Ένα Σύστημα Διαχείρισης Περιεχομένου Web (WCMS ή Web CMS) είναι ένα λογισμικό Σύστημα Διαχείρισης Περιεχομένου (CMS), το οποίο υλοποιείται συνήθως ως μία Web εφαρμογή για τη δημιουργία και τη διαχείριση περιεχομένου HTML. Χρησιμοποιείται για τη διαχείριση και τον έλεγχο μεγάλων και δυναμικών συλλογών Web υλικού (έγγραφα

HTML και οι σχετικές εικόνες τους). Ένα WCMS διευκολύνει τη δημιουργία, τον έλεγχο, τη μορφοποίηση, καθώς και πολλές σημαντικές λειτουργίες συντήρησης περιεχομένου Web.

Συνήθως το λογισμικό παρέχει εργαλεία δημιουργίας, τα οποία είναι σχεδιασμένα να επιτρέψουν σε χρήστες με ελάχιστη ή καθόλου γνώση γλωσσών προγραμματισμού ή γλωσσών δημιουργίας ιστοσελίδων (markup languages) να δημιουργήσουν και να διαχειριστούν περιεχόμενο με σχετική ευκολία.

Τα περισσότερα τέτοια συστήματα χρησιμοποιούν μια βάση δεδομένων για να αποθηκεύσουν περιεχόμενο, μεταδεδομένα (metadata), ή/και αντικείμενα που μπορεί να απαιτούνται από το σύστημα. Το περιεχόμενο είναι συχνά, αλλά όχι πάντα, αποθηκευμένο σε μορφή XML, για να διευκολύνει την επαναχρησιμοποίηση και να επιτρέψει τις εύκαμπτες επιλογές παρουσίασης.

Ένα στρώμα παρουσίασης (presentation layer) παρουσιάζει το περιεχόμενο στους επισκέπτες με βάση ένα σύνολο προτύπων (templates). Τα πρότυπα αυτά είναι μερικές φορές αρχεία XSLT.

Η διαχείριση γίνεται βασικά μέσω browser (browser – based interface), αλλά μερικά συστήματα απαιτούν τη χρήση ενός συστήματος πελάτη. Σε αντίθεση με τα παραδοσιακά συστήματα δημιουργίας ιστοσελίδων όπως το Microsoft Frontpage ή το Adobe Dreamweaver, ένα WCMS επιτρέπει σε χρήστες με ελάχιστη ή καθόλου τεχνική κατάρτιση να κάνουν αλλαγές σε έναν υπάρχοντα ιστοχώρο. Ένα WCMS απαιτεί συνήθως έναν πεπειραμένο προγραμματιστή για να οργανώσει και προσθέτει διάφορα χαρακτηριστικά, αλλά είναι πρώτιστα ένα εργαλείο συντήρησης ιστοχώρου για μη τεχνικούς διαχειριστές.

Ένα WCMS υποστηρίζει τα ακόλουθα κύρια χαρακτηριστικά:

- **Αυτοματοποιημένα πρότυπα**

Δημιουργία τυποποιημένων προτύπων (συνήθως HTML και XML) που μπορούν να εφαρμοστούν αυτόματα σε νέο και υπάρχον περιεχόμενο, επιτρέποντας την αλλαγή της εμφάνισης όλου του περιεχομένου από ένα κεντρικό σημείο.

- **Εύκολα διαμορφώσιμο περιεχόμενο**

Όταν το περιεχόμενο ενός site διαχωρίζεται από την οπτική παρουσίασή του, η διαχείρισή μπορεί να γίνει ευκολότερα και γρηγορότερα. Τα περισσότερα λογισμικά WCMS περιλαμβάνει εργαλεία διαμόρφωσης (τα λεγόμενα WYSIWYG) τα οποία επιτρέπουν σε μη τεχνικά χρήστες να δημιουργήσουν και να εκδώσουν περιεχόμενο.

- **Κλιμακωτές ομάδες χαρακτηριστικών**

Τα περισσότερα λογισμικά WCMS περιλαμβάνουν plug-ins ή modules που μπορούν να εγκατασταθούν εύκολα και να επεκτείνουν τις λειτουργικές ικανότητες ενός site.

- **Ενημερώσεις προτύπων Web**

Τα λογισμικά WCMS λαμβάνουν συνήθως σε τακτά χρονικά διαστήματα ενημερώσεις που περιλαμβάνουν νέα σύνολα χαρακτηριστικών και κρατούν το σύστημα συνεπές στα τρέχοντα πρότυπα Web.

- **Διαχείριση της ροής εργασιών**

Η ροή εργασιών είναι η διαδικασία δημιουργίας κύκλων διαδοχικών και παράλληλων διεργασιών που πρέπει να ολοκληρωθούν μέσα σε ένα CMS. π.χ., ένας δημιουργός περιεχομένου υποβάλλει μια ιστορία, η οποία δημοσιεύεται, μόνο αφού αναθεωρηθεί από το συντάκτη αντιγράφων και εγκριθεί από τον αρχισυντάκτη.

- **Διαχείριση εγγράφων**

Ένα λογισμικό CMS μπορεί να παρέχει τα μέσα για τη διαχείριση του κύκλου ζωής ενός εγγράφου από τον αρχικό χρόνο δημιουργίας, τις αναθεωρήσεις, τη δημοσίευση, την αρχειοθέτηση και την καταστροφή του εγγράφου.

- **Εικονικό περιεχόμενο**

Ένα λογισμικό CMS μπορεί να παρέχει τα μέσα σε κάθε χρήστη να εργαστεί μέσα σε ένα εικονικό αντίγραφο του συνόλου του ιστοχώρου, του συνόλου εγγράφων ή/και της βάσης κώδικα. Αυτό επιτρέπει στις διάφορες αλλαγές που πραγματοποιούνται σε πολλαπλούς αλληλοεξαρτώμενους πόρους του συστήματος να ελέγχονται πριν από την εκτέλεσή τους.

5.3 ΚΑΤΗΓΟΡΙΕΣ ΣΥΣΤΗΜΑΤΩΝ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ WEB

Υπάρχουν τρεις σημαντικοί τύποι συστημάτων WCMS: επεξεργασίας offline (offline processing), επεξεργασίας online (online processing) και υβριδικά συστήματα (hybrid systems). Οι όροι αυτοί περιγράφουν το πότε και το πώς εφαρμόζονται τα πρότυπα παρουσίασης (presentation templates) για να διαμορφώσουν μια ιστοσελίδα.

5.3.1 ΕΠΕΞΕΡΓΑΣΙΑ OFFLINE

Αυτά τα συστήματα προεπεξεργάζονται όλο το περιεχόμενο, εφαρμόζοντας πρότυπα πριν από τη δημοσίευση για να παράγουν ιστοσελίδες. Τα συστήματα Sagar Vignette και Bricolage είναι παραδείγματα αυτού του τύπου συστήματος. Δεδομένου ότι τα συστήματα προεπεξεργασίας δεν απαιτούν έναν κεντρικό υπολογιστή για να εφαρμόσουν τα πρότυπα, μπορούν επίσης να χρησιμοποιηθούν και ως καθαρά εργαλεία σχεδίασης. Το σύστημα Adobe Contribute αποτελεί ένα παράδειγμα αυτής της προσέγγισης.

5.3.2 ΕΠΕΞΕΡΓΑΣΙΑ ONLINE

Αυτά τα συστήματα εφαρμόζουν τα πρότυπα κατ' απαίτηση (on-demand). Ο κώδικας HTML παράγεται όταν ένας χρήστης επισκέπτεται τη σελίδα, ή "τραβιούνται" από μια μνήμη cache. Μερικοί από τα πιο γνωστά συστήματα ανοικτού κώδικα (open source) σε αυτή την κατηγορία είναι τα Mambo, Joomla! , Drupal, WordPress, Zikula και Plone.

5.3.3 ΥΒΡΙΔΙΚΑ ΣΥΣΤΗΜΑΤΑ

Τα συστήματα αυτά συνδυάζουν τις offline και online προσεγγίσεις. Μερικά από αυτά τα συστήματα παράγουν εκτελέσιμο κώδικα (π.χ. JSP, PHP, Perl) παρά στατικό HTML, έτσι ώστε το ίδιο το CMS να μη χρειάζεται να τοποθετηθεί σε κάθε Web server. Άλλα τέτοια συστήματα, όπως το Blosxom, είναι σε θέση να λειτουργούν είτε σε offline είτε σε online λειτουργία.

Τα Συστήματα Διαχείρισης Περιεχομένου άρχισαν να αναπτύσσονται επίσημα ως εμπορικά προϊόντα λογισμικού το 1995 με δύο αρχικά προϊόντα, το TeamSite της Interwoven και το Vignette Content Management της Vignette. Με την εξέλιξη του Internet, η σημασία των συστημάτων αυτών ως κομμάτι της υποδομής IT ενός οργανισμού άρχισε να γίνεται ολοένα και μεγαλύτερη. Έτσι, προμηθευτές από παρόμοιες αγορές, συμπεριλαμβανομένων και των Documentum και FileNet, εταιρειών με μακρά παράδοση στην ανάπτυξη λογισμικού Διαχείρισης Εγγράφων (Document Management) άρχισαν να αναπτύσσουν τα δικά τους WCMS. Άλλοι προμηθευτές, όπως οι Stellent και RedDot Solutions, άρχισαν να κάνουν την εμφάνισή τους. Μέχρι το 2002, τα τμήματα IT είχαν αρχίσει να αναζητούν ένα μοναδικό προμηθευτή που θα μπορούσε να διαχειριστεί όλο το μη δομημένο περιεχόμενο (έγγραφα, ιστοσελίδες, οπτικοακουστικά μέσα κτλ.) και η Διαχείριση Περιεχομένου Web (WCM) έγινε υποσύνολο μιας νέας υπερκατηγορίας, της Επιχειρησιακής Διαχείρισης Περιεχομένου (Enterprise Content Management – ECM).

5.4 ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ ΠΟΥ ΥΠΑΡΧΟΥΝ

Αρκετοί είναι οι υπέρμαχοι διαφόρων cms. Παρακάτω αναλύω κάποια από τα διασημότερα cms που έχω χρησιμοποιήσει σε διάφορα projects μου. Δεν είμαι της άποψης ότι ένα cms είναι για όλα. Αρκετές φορές βλέπουμε ιστοσελίδες φτιαγμένες με wordpress (πρόγραμμα κατασκευής και διαχείρισης ιστοσελίδων), οι οποίες θα ήταν ευκολότερο να γίνουν με Joomla και το αντίστροφο ή με κάποιο άλλο cms.

Υπάρχει μία τάση από αρκετούς κατασκευαστές να κατηγορούν τα έτοιμα cms, κυρίως το Joomla ότι δεν είναι αρκετά ευέλικτο. Όπως και τα custom cms, έτσι και το Joomla έχει φτιαχτεί για να εξυπηρετήσει κάποιες συγκεκριμένες ανάγκες. Δεν υπάρχουν γενικής χρήσης cms που να εξυπηρετούν όλες τις ανάγκες 100%. Το ζήτημα είναι να υπάρχει αρκετή εκπαίδευση πάνω σε διάφορα cms ώστε να γνωρίζει ο χρήστης ή ο developer πότε και ποιο cms να χρησιμοποιήσει.

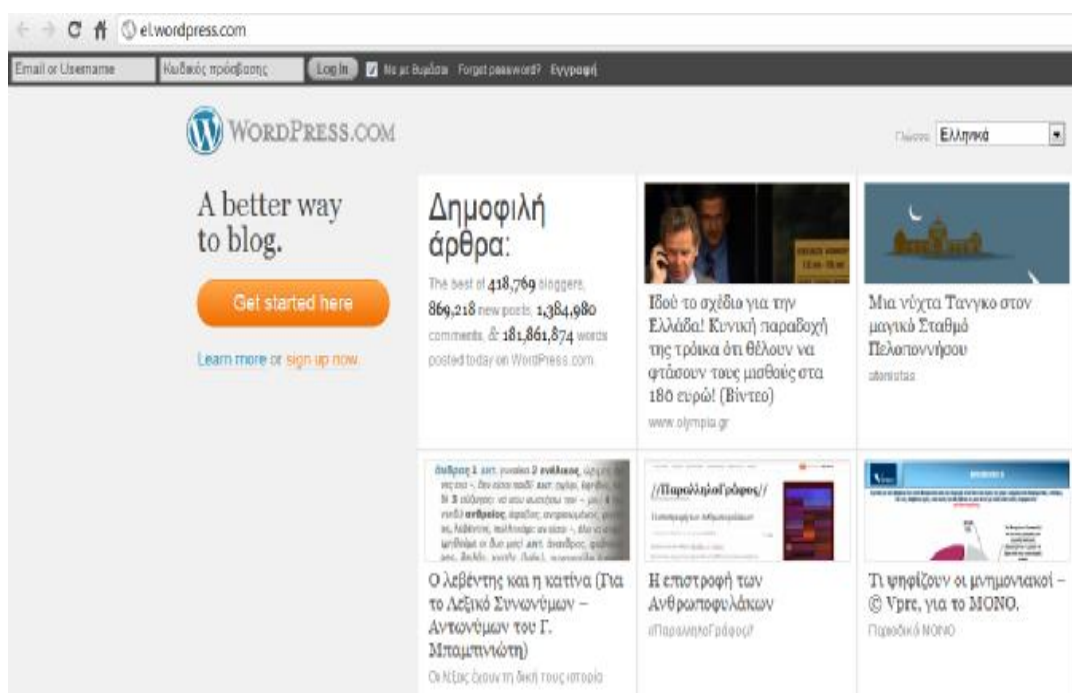
Ανάλογα λοιπόν με τις ανάγκες του project προτιμώ να επιλέγω cms που να κάνουν τη δουλειά μου πιο γρήγορα, από το να ήξερα ένα και μόνο cms, με το οποίο να προσπαθούσα να κάνω τα πάντα. Παρακάτω παραθέτω κάποια cms που τα γνωρίζουμε όλοι μας, καθώς επίσης και πότε θεωρώ ότι θα έπρεπε να χρησιμοποιηθούν. Να τονίσω ότι η παρακάτω παράθεση αποτελεί καθαρά προσωπική μου άποψη.

5.4.1 WORDPRESS

Το wordpress έχει αρχίσει εδώ και καιρό να κερδίζει αρκετό έδαφος και ολοένα και περισσότερες εταιρείες το χρησιμοποιούν ως cms. Χρησιμοποιείται αρκετές φορές, καθότι τα ελληνικά δεδομένα δεν έχουν υψηλές απαιτήσεις.

Το Wordpress από την κατασκευή του είναι μια καθαρή *blogging* πλατφόρμα, με πολύ καλό σύστημα διαχείρισης των σχολίων (*comments*) με την οποία μπορείτε να ξεκινήσετε τη δημοσίευση και ανάρτηση των άρθρων σας στο ιστολόγιό σας (*blog*). Τα άρθρα έχουν χρονολογική ταξινόμηση. Απευθύνεται συνήθως σε αρχάριους, οι οποίοι δεν έχουν καμία γνώση σχεδιασμού και κατασκευής ιστοσελίδων καθώς και HTML, CSS, PHP και MYSQL, αλλά τους ενδιαφέρει μόνο η διαχείριση του περιεχομένου, δηλαδή η ανάρτηση άρθρων. Από την αρχική εγκατάσταση του δεν διαθέτει κάποιο *menu* πλοήγησης, αν και αυτό λύνετε με κάποιο

άλλο *template* ή *plugin*. Το Wordpress μπορεί να επεκταθεί με τη χρήση *plugin* και την διαμόρφωσή του με *template*στα οποία εύκολα μπορείτε να κατεβάσετε από το Wordpress.org. Είναι από την κατασκευή του φιλικό προς τις μηχανές αναζήτησης και υπάρχουν επίσης διάφορα *plugins* που το κάνουν ακόμα πιο *SEO friendly*.



Εικόνα 1: Αρχική οθόνη wordpress

Εκτός όμως από την ευχρηστία και φιλικότητα που έχει αυτό το σύστημα διαχείρισης, προτιμούμε το wordpress για του παρακάτω βασικούς παράγοντες:

- Η κατασκευή ιστοσελίδων με wordpress είναι ιδιαίτερα εύελικτο. Έχει επίσης ένα σύνολο από πρόσθετα, τα οποία είναι ξεχωριστά από το κύριο λογισμικό. Κάποια από αυτά είναι δωρεάν, ενώ κάποια άλλα όχι. Μπορείτε λοιπόν αν θελήσετε να προσθέσουμε ένα Gallery, να το προσθέσουμε μέσα σε μερικά λεπτά. Έτσι γλιτώνετε το κόστος ανάπτυξης, αλλά και πολύτιμο χρόνο. Αφού το εγκαταστήσουμε, μπορείτε άμεσα να αρχίσετε να προσθέτετε περιεχόμενο.
- Το Wordpress θεωρείτε το πιο φιλικό σύστημα διαχείρισης προς τον χρήστη. Δεν είναι τυχαίο ότι το Wordpress θεωρείτε από πολλούς ως το πιο εύκολο σύστημα διαχείρισης που υπάρχει αυτή τη στιγμή στην αγορά. Έτσι, δεν θα χρειαστείτε αρκετές ώρες για την εκμάθηση του συστήματος.
- Φιλικό προς τις μηχανές αναζήτησης. Το Wordpress θεωρείται από πολλούς ως ένα από τα φιλικότερα συστήματα για τις μηχανές αναζήτησης. Μπορείτε έτσι να επιτύχετε εύκολα υψηλότερη κατάταξη στο Google και αποτελεσματικότερη προώθηση ιστοσελίδων. Επίσης,

υπάρχουν στην αγορά αρκετά διαθέσιμα πρόσθετα, τα οποία μπορούν να το κάνουν ακόμη πιο φιλικό και να αποκτήσετε το μέγιστο των δυνατοτήτων σας.

- **Πρόσθετα.** Τα πρόσθετα είναι αυτά που παρέχουν την έξτρα ευελιξία στο wordpress. Μπορείτε να έχετε κάποιες απαιτήσεις και το σύστημα διαχείρισης να μην ικανοποιεί τις ανάγκες σας. Με το wordpress, μπορούμε να περάσουμε τα κατάλληλα πρόσθετα και να αποκτήσετε αυτό ακριβώς που επιθυμείτε.
- **Είναι έγκυρο από το W3C.** Το Wordpress είναι ένα σύστημα διαχείριση περιεχομένου, το οποίο είναι σύμφωνο με τους κανόνες του W3C. Έτσι είναι σύμφωνο με όλα τα standards που θέτει το Google και οι άλλες μηχανές αναζήτησης.
- **Μεγάλος αριθμός από εταιρείες ανάπτυξης ιστοσελίδων.** Το wordpress αποτελείται από μία μεγάλη κοινότητα εταιρειών κατασκευής ιστοσελίδων, για αυτό και μπορείτε να βρείτε εύκολα αμέτρητα templates, ή να μισθώστε μία εταιρεία κατασκευής ιστοσελίδων, ώστε να κατασκευάσει το δικό σας.
- **Είναι λογισμικό ανοικτού κώδικα.** Το Wordpress είναι ένα σύστημα διαχείρισης περιεχομένου, το οποίο μπορεί εύκολα να τροποποιηθεί χωρίς περιορισμούς ανάλογα με τις ανάγκες σας, χωρίς να χρειάζεστε άδεια.

Το WordPress μπορεί να χρησιμοποιηθεί για:

- Προσωπικά - μικρά blogs ενώ με τα κατάλληλα plugins αποτελεί καλή λύση για μέτριοι μεγέθους portals ή blogs.
- Μικρά φόρουμ συζητήσεων
- Ιστοσελίδες εταιρικές με μικρές ανάγκες

Δεν συνίσταται για:

- Μεγάλα portals ή blogs
- Μεγάλα φόρουμ συζητήσεων
- Ιστοσελίδες εταιρικές με μέτριες ή υψηλές ανάγκες

5.4.2 JOOMLA

Το πιο διάσημο cms που υπάρχει αυτή τη στιγμή. Έχει τη μεγαλύτερη κοινότητα υποστήριξης και αρκετά extensions για να το κάνουν πιο ευέλικτο. Το χρησιμοποιώ αρκετές φορές όπως και το wordpress, καθότι με καλύπτει στα περισσότερα projects.



Εικόνα 2: Εικόνα από Joomla

Το Joomla μπορεί να χρησιμοποιηθεί για:

- Προσωπικές ιστοσελίδες
- |
- ιστοσελίδες για εταιρείες με μικρές ή μέτριες ανάγκες Directories
- Μικρά φόρουμ συζητήσεων
- Έχει αξιοπρεπές shopping cart (virtuemart), το οποίο είναι δωρεάν.

Δεν συνίσταται για:

- Μεγάλα portals ή blogs
- Μεγάλα φόρουμ συζητήσεων.
- Ιστοσελίδες για εταιρείες με υψηλές ανάγκες.

Σε γενικές γραμμές είναι καλύτερο το wordpress για προσωπικές ιστοσελίδες ή portals.

5.4.3 DRUPAL

Το Drupal θεωρείται από πολλούς, ένα από τα κορυφαία συστήματα διαχείρισης περιεχομένου. Το Drupal είναι στην αγορά εδώ και αρκετά χρόνια και είναι γνωστό και την άψογη αρχιτεκτονική του, την κοινότητα που το στηρίζει και την ποιότητα του κώδικα του. Το Drupal

αποτελεί την καλύτερη λύση για αυξημένες ανάγκες διαχείρισης. Σε γενικές γραμμές αποτελεί την καλύτερη επιλογή για projects μεσαίου ή υψηλού επιπέδου. Σε καμία περίπτωση δεν αναφέρω ότι δεν μπορεί να χρησιμοποιηθεί και σε μικρότερα projects, απλά δεν αποτελεί την πιο γρήγορη λύση.



Εικόνα 3: Εικόνα από Drupal

Το Drupal μπορεί να χρησιμοποιηθεί για:

- Μεγάλα portals
- Φόρουμ συζητήσεων
- Ιστοσελίδες για εταιρείες με αυξημένες ανάγκες Intranet applications
Προσωπικές ιστοσελίδες ή blogs Βιβλιοθήκες
- Έχει αξιοπρεπές shopping cart (ubercart), αλλά πρέπει να το αγοράσετε.
Δεν συνιστάται για κατασκευή ιστοσελίδων με ανάγκες όπως π.χ. NASA, national geographic κλπ

ΜΕΡΙΚΕΣ ΑΠΟ ΤΙΣ ΔΥΝΑΤΟΤΗΤΕΣ ΤΟΥ DRUPAL CMS

Διευθύνσεις σελίδων (URLS) φιλικές προς τις μηχανές αναζήτησης

- Το drupal αυτόματα (ή καθοριζόμενο από τον χρήστη), έχει την δυνατότητα να δημιουργεί για κάθε σελίδα περιεχομένου διευθύνσεις φιλικές προς τις μηχανές αναζήτησης έτσι ώστε η ιστοσελίδα σας να εμφανίζεται ψηλότερα στα αποτελέσματα των αναζητήσεων.

Για παράδειγμα η διεύθυνση μπορεί να έχει την μορφή www.ds-creative.gr/services/web_design.html που είναι σαφώς πιο εύκολο να διαβαστεί όχι μόνο από τις μηχανές αναζήτησης αλλά και από τους επισκέπτες της ιστοσελίδας σας.

Άλλα συστήματα διαχείρισης περιεχομένου εμφανίζουν συνήθως παράξενες διευθύνσεις της μορφής:

www.ds-creative.gr/services.php?actid=9&catid=3&cid=7& κτλ

Διαχείριση της ιστοσελίδας σας από όπου και αν βρίσκεστε

- Το μόνο που χρειάζεστε για να διαχειριστείτε την ιστοσελίδα σας είναι ένας υπολογιστής με σύνδεση στο internet και ο κωδικός πρόσβασης στο σύστημα. Έτσι μπορείτε να προσθέσετε κείμενο, φωτογραφίες και ότι άλλο θέλετε από όπου και αν βρίσκετε.

Το drupal είναι ένα από τα ασφαλέστερα CMS

- Η κοινότητα ανάπτυξης του Drupal φροντίζει με συνεχείς ανανεώσεις και ελέγχους ασφαλείας να είναι το drupal προστατευμένο από κακόβουλες "επιθέσεις".

Εύκολη και γρήγορη επεξεργασία περιεχομένου

- Με το Drupal έχετε την δυνατότητα να επεξεργαστείτε το περιεχόμενο της ιστοσελίδας σας έχοντας απλά λίγες γνώσεις επεξεργαστή κειμένου (όπως το Word).

Υποστήριξη πολύγλωσσικών ιστοσελίδων

- Το Drupal Cms σας δίνει την δυνατότητα να προσθέσετε μεταφράσεις των σελίδων της ιστοσελίδας σας σε πολλές διαφορετικές γλώσσες με απλό και γρήγορο τρόπο.

Απεριόριστες κατηγορίες και υπό – σελίδες

- Αν ο όγκος του περιεχομένου σας είναι μεγάλος, το drupal σας δίνει την δυνατότητα να το χωρίσετε σε πολλές διαφορετικές κατηγορίες και υπό σελίδες.

Διαχείριση χρηστών

- Το Drupal μπορεί να χρησιμοποιηθεί και για την ανάπτυξη μιας online κοινότητας στην οποία τα μέλη θα έχουν την δυνατότητα να εγγραφούν και να αλληλεπιδράσουν με τα άλλα μέλη της ιστοσελίδας.

Μερικές από τις extra δυνατότητες του Drupal

- Ανακοινώσεις και άρθρα.
- Σύστημα μαζικής αποστολής διαφημιστικών ή ενημερωτικών email (Newsletter).
- Σύστημα Blog.
- Προσαρμοζόμενες φόρμες επικοινωνίας (απεριόριστος αριθμός) με προστασία από spam.
- Σύστημα αναζήτησης περιεχομένου.
- Σχόλια χρηστών και ψηφοφορίες.

- Multimedia Gallery με αυτόματη δημιουργία μικρογραφιών και προσαρμογή του μεγέθους της φωτογραφίας.
- Προσαρμογή συστήματος Forum.

5.4.4 DOTNETNUKE (DNN)

Είναι το πιο διάσημο cms στην πλατφόρμα της .NET. Προσωπικά το επιλέγω ως εναλλακτική περίπτωση που χρειάζεται ένα site να τρέχει σε windows server. Έχει αρκετά μεγάλη κοινότητα υποστήριξης και μπορεί να χρησιμοποιηθεί σε αντίστοιχα μεγάλα projects όπως το Drupal. Υπάρχει επίσης η δυνατότητα εκτός από τη δωρεάν έκδοση να αγοράσετε κάποια από τα άλλα πακέτα. Οι διαφορές στο ίδιο το σύστημα είναι ουσιαστικά ανύπαρκτες όποιο πακέτο και να διαλέξετε, η μόνη διαφορά είναι στην υποστήριξη που υπάρχει για το κάθε πακέτο.



Εικόνα 4: Εικόνα από Dotnetnuke

Ιδιότητες του DotNetNuke (Features)

Ευπροσάρμοστο – DotNetNuke είναι ανοιχτού κώδικα εφαρμογή ιδανικό πλαίσιο για δημιουργία, ανάπτυξη και διαχείριση διαδραστικού Ιστού, για ενδοδικτυακές ή extranet περιοχές. (Dotnetnuke, 2009)

Φιλική χρήση – DotNetNuke είναι σχεδιασμένο για να κάνει εύκολη τη διαχείριση στους διαχειριστές, στους συντάκτες περιεχομένου, στους υπεύθυνους για την ανάπτυξη, και τους σχεδιαστές για να διαχειριστούν όλες τις πτυχές των προτερημάτων Ιστού τους. Wizards, περιεχόμενο – ευαίσθητη βοήθεια, και ένα καλά ερευνημένο ενδιάμεσο με τον χρήστη παρέχει μια ανώτερη εμπειρία χρηστών.

Δυναμικό – το DotNetNuke μπορεί να υποστηρίξει πολλαπλούς ιστοχώρους από μια ενιαία εφαρμογή εγκατάστασης. Διαιρώντας τις

διοικητικές επιλογές μεταξύ του επιπέδου οικοδεσποτών και των μεμονωμένων περιοχών, το DotNetNuke παρέχει στον διαχειριστή τη δυνατότητα να διαχειριστεί έναν απεριόριστο αριθμό ιστοχώρων – κάθε ένα με τη μοναδική εμφάνιση και το περιεχόμενό του.

Χαρακτηριστικό γνώρισμα, πλούσιο σε – το DotNetNuke είναι «φορτωμένο» με ένα σύνολο ενσωματωμένων χαρακτηριστικών γνωρισμάτων που παρέχει την εξαιρετική.

Φιλοξενία ιστοτόπων, σχεδίαση ιστού, διαχείριση περιεχομένου, ασφάλεια και επιλογή μέλους είναι όλα εύκολα για τη διαχείριση και προσαρμόζονται εύκολα μέσω απλής μηχανής αναζήτησης – βασισμένο σε εργαλεία.

Υποστηρίζει - το DotNetNuke είναι ρυθμισμένο να υποστηρίζεται από την εταιρία DotNetNuke Corporation, ένα νομικό πρόσωπο που παρέχει την παραγωγή Συμφωνίες Επιπέδων Εξυπηρέτησης όπως και άλλες επαγγελματικές υπηρεσίες για την πλατφόρμα. Το DotNetNuke υποστηρίζεται επίσης από Ομάδα πυρήνων των εθελοντικών υπεύθυνων για την ανάπτυξη και αφέρωση διεθνούς κοινότητας. Η βοήθεια είναι πάντα προσιτή, μέσω των ομάδων χρηστών και των σε απευθείας σύνδεση φόρουμ. Τέλος κάτι που κάνει προσιτή τη βοήθεια είναι οι πύλες των πόρων και ένα δίκτυο επιχειρήσεων οι οποίοι ειδικεύονται στο DNN®.

5.4.5 MOODLE

Το όνομα Moodle προέρχεται από το ακρόνυμο των λέξεων Modular Object- Oriented Dynamic Learning Environment και είναι ένα πακέτο λογισμικού για την δημιουργία διαδικτυακών μαθημάτων. Δημιουργήθηκε το 1999 από τον Αυστραλό Martin Dougiamas ως τμήμα της διδακτορικής του διατριβής. Οι δυνατότητες του δεν περιορίζονται στην εκπαίδευση από απόσταση αλλά μπορεί να λειτουργήσει συμπληρωματικά και στην κλασική διδασκαλία. Μέσα από την πλατφόρμα Moodle ο εκπαιδευτικός μπορεί να παρουσιάσει το μάθημα του με τέτοιο τρόπο ώστε να προκαλεί ενδιαφέρον στον εκπαιδευόμενο χρησιμοποιώντας διάφορα μέσα. Για παράδειγμα, διάθεση υλικού του μαθήματος σε διαφορετικές μορφές, προσθήκη δραστηριοτήτων, σύγχρονη και ασύγχρονη επικοινωνία μεταξύ μελών, ανάθεση εργασιών, αξιολόγηση της επίδοσης.

Το Moodle παρέχεται δωρεάν σαν λογισμικό Open Source (κάτω από την άδεια GNU-Public Lisence). Αυτό πρακτικά σημαίνει ότι το Moodle έχει πνευματικά δικαιώματα αλλά μπορεί να χρησιμοποιηθεί, να γίνουν αντιγραφές του, και τροποποιήσεις του υπό τον όρο να παρέχεται ο πηγαίος κώδικας και σε άλλους, να μην αλλάξει ή αφαιρεθεί η επίσημη άδεια και να εφοδιάζετε με αυτήν οποιαδήποτε παράγωγη δουλειά. Το

Moodle μπορεί να τρέξει σε οποιοδήποτε σύστημα υπολογιστή υποστηρίζει τη γλώσσα PHP, και διαφορετικούς τύπους βάσεων δεδομένων αλλά κυρίως χρησιμοποιεί την MySQL. Είναι μια εφαρμογή Web/Client, δηλαδή βρίσκεται εγκατεστημένο σε κάποιο Server και οι χρήστες έχουν πρόσβαση σ' αυτό μέσω ενός φυλλομετρητή, όπως είναι ο Internet Explorer, ο Mozilla Firefox κ.α. Επομένως από πλευράς χρήστη (μαθητή, καθηγητή και διαχειριστή) απαιτούνται μόνο η ύπαρξη μιας σύνδεσης στο Internet και το κατάλληλο λογισμικό περιήγησης.

Το Moodle ενσωματώνει λειτουργίες οι οποίες το καθιστούν ένα πολυσύνθετο, ευέλικτο και αρκετά ασφαλές εργαλείο στην ασύγχρονη εκπαίδευση από απόσταση. Είναι διαδεδομένο σε όλο τον κόσμο. Σήμερα υπάρχουν 19685 εγκαταστάσεις σε 171 χώρες και το λογισμικό του moodle είναι διαθέσιμο σε 75 γλώσσες. Μεταξύ των οργανισμών που το χρησιμοποιούν είναι το MIT, το Yale και άλλα πανεπιστήμια στην Αμερική και στην Ευρώπη. Στην Ελλάδα η πλατφόρμα έχει εγκατασταθεί σε περισσότερους από 45 φορείς εκπαίδευσης και κατάρτισης, μεταξύ των οποίων το Εθνικό Μετσόβειο Πολυτεχνείο και τα Πανεπιστήμια Μακεδονίας και Θεσσαλίας.

Βασικά Χαρακτηριστικά:

1. οργάνωση του εκπαιδευτικού υλικού ανάλογα με τις απαιτήσεις που υφίστανται σε κάθε περίπτωση (π.χ. ανά εβδομάδα ή ανά θεματική ενότητα)
2. Υποστήριξη μιας μεγάλης ποικιλίας δραστηριοτήτων διαφορετικού τύπου (Forums, Journals, Quizzes, Resources, Choices, Surveys και Assignments).
3. Αυτόματη εγγραφή των φοιτητών μέσα από το Διαδίκτυο οι οποίοι στη συνέχεια εφ' όσον έχουν τα κατάλληλα δικαιώματα μπορούν να εγγραφούν στα μαθήματα της αρεσκείας τους χωρίς την παρέμβαση του διαχειριστή του μαθήματος.
4. Παροχή υψηλού επιπέδου ασφαλείας.
5. Αυτόματη βαθμολόγηση των διαγωνισμάτων με απευθείας ενημέρωση του φοιτητή.
6. Δυνατότητα δημιουργίας του προσωπικού προφίλ για τους εγγεγραμμένους φοιτητές.
7. Δυνατότητα ηλεκτρονικής υποβολής των εργασιών των φοιτητών στο σύστημα. Για αυτές τις εργασίες υπάρχει η δυνατότητα καθορισμού προθεσμίας υποβολής (Deadline)
8. Δυνατότητα καταγραφής και ελέγχου των διάφορων τύπων δραστηριοτήτων των φοιτητών από το διαχειριστή του συστήματος.
9. Υποστήριξη 75 και πλέον διαφορετικών φυσικών γλωσσών, μεταξύ των οποίων περιλαμβάνεται και η Ελληνική γλώσσα.



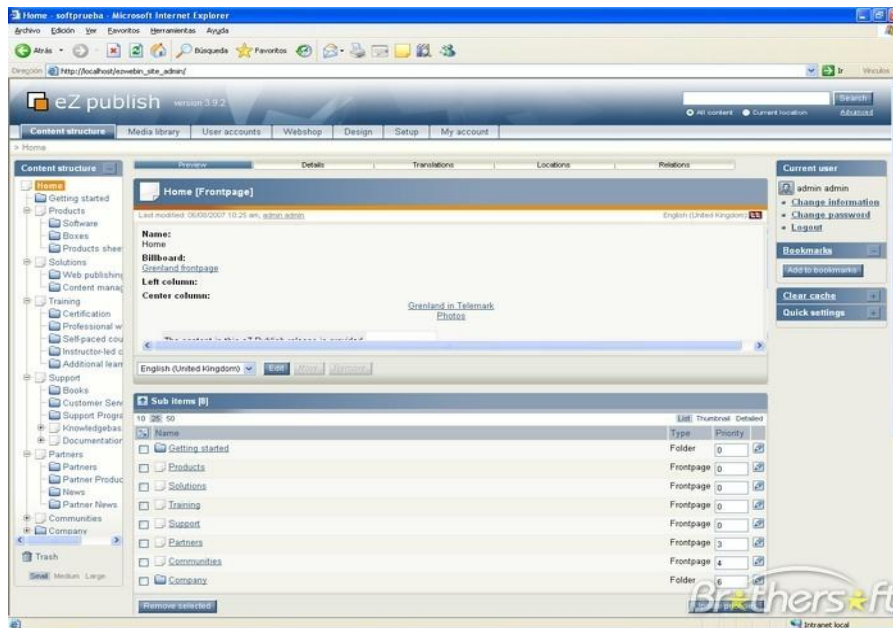
Εικόνα 5: Εικόνα από Moodle

5.4.6 EZPUBLISH

Από τις σημαντικότερες και πιο δημοφιλείς λύσεις είναι αυτή του eZpublish, το οποίο πέρα από enterprise CMS αποτελεί και μία πλατφόρμα (framework) για την ανάπτυξη εφαρμογών. Υπολογίζεται ότι έχει εγκατασταθεί σε περισσότερα από 150.000 sites σε περισσότερες από 16 χώρες. Η ανοιχτή αρχιτεκτονική του δίνει τη δυνατότητα στους χρήστες του να το χρησιμοποιούν για τη δημιουργία sites, intranets και document management. Το eZpublish παρέχεται τόσο με εμπορική όσο και με την κλασική άδεια GPL.

Το παραπάνω cms αποτελεί την ιδανική λύση για ευέλικτα projects μεγάλης έκτασης. Το χρησιμοποιούν οι MySQL (<http://www.mysql.com>) Zend (<http://www.zend.com>), ακόμη και η NASA, το National Geographic.

Η εφαρμογή eZ Publish (www.ez.no) είναι ένα βασισμένο σε PHP σύστημα διαχείρισης περιεχομένων κατάλληλο για την ανάπτυξη μικρών, προσωπικών σελίδων Web και μεγάλων, εμπορικών τόπων Web. Συμπεριλαμβάνει υποστήριξη για την δημιουργία και επεξεργασία περιεχομένων, την παρακολούθηση αλλαγών, την δόμηση προσαρμοσμένων ροών εργασίας δημοσιοποίησης και την δημιουργία ενός καταστήματος Web με ενοποιημένες συναρτήσεις ηλεκτρονικού εμπορίου (e – commerce).



Εικόνα 6: Εικόνα από eZpublish

ΚΕΦΑΛΑΙΟ 6 ΕΙΣΑΓΩΓΗ ΣΤΟ DREAMWEAVER MX

6.1 ΕΙΣΑΓΩΓΗ

Το Dreamweaver MX έχει ενσωματώσει τα χαρακτηριστικά και τις δυνατότητες των προγραμμάτων (πακέτων) Dreamweaver, UltraDev και HomeSite σ' ένα μόνο εργαλείο. Το Dreamweaver MX διαθέτει επίσης ενισχυμένη υποστήριξη για συστήματα server, όπως είναι τα PHP, ASP.NET και ColdFusion MX. Για πρώτη φορά το Dreamweaver MX μάς δίνει τη δυνατότητα να επιλέξουμε το στυλ και τη διάταξη (layout) του User Interface (διασύνδεση χρήστη). Όταν ξεκινάμε το Dreamweaver MX για πρώτη φορά, θα εμφανισθεί ένα πλαίσιο διαλόγου όπου θα μπορούμε να καθορίσουμε το στυλ του User Interface που θέλουμε να χρησιμοποιήσουμε.

Θα πρέπει να επιλέξουμε το πλήκτρο επιλογής *Dreamweaver MX Workspace* αλλά ακόμη και αν επιλέξουμε το πλήκτρο επιλογής *Dreamweaver 4 Workspace*, θα μπορούμε αργότερα να αλλάξουμε το layout από τα Preferences. Καθώς τα panels παίζουν έναν πολύ σημαντικό ρόλο στον τρόπο που δουλεύει το Dreamweaver MX, θα δούμε τα πιο συχνά χρησιμοποιούμενα απ' αυτά.

Το *Insert panel* αποτελείται από πολλές καρτέλες, όπου η κάθε καρτέλα περιέχει ένα διαφορετικό σύνολο από εργαλεία. Μερικές απ' αυτές τις καρτέλες αλλάζουν δυναμικά ανάλογα με το είδος του εγγράφου με το οποίο εργαζόμαστε. Για παράδειγμα, αν δημιουργήσουμε μια PHP σελίδα θα εμφανισθεί μια καρτέλα με τον τίτλο PHP, η οποία θα περιέχει κάποια εργαλεία χρήσιμα όταν δουλεύουμε σε πρωτογενή κώδικα

(source code). Παρόμοια, αν δουλεύουμε με μια σελίδα ColdFusion, θα εμφανισθούν κάποιες άλλες καρτέλες και η καρτέλα της PHP θα εξαφανισθεί καθώς δεν θα έχει νόημα η παρουσία της. Αυτό το είδος των αλλαγών αποκαλείται *context sensitivity* (ευαισθησία στο κείμενο) και είναι ειδικά σχεδιασμένο ώστε να μας βοηθάει να βρίσκουμε τα εργαλεία που χρειαζόμαστε, όταν τα χρειαζόμαστε.

Από το *Design panel* έχουμε εύκολη πρόσβαση στα στυλ CSS (Cascading Style Sheet) και στα στυλ της HTML που χρησιμοποιούμε. Επίσης, αν και δεν αποτελεί αυστηρά ένα στοιχείο του design, υπάρχει και η καρτέλα *Behaviors* από την οποία μπορούμε να έχουμε εύκολη πρόσβαση σε μια σειρά από ενσωματωμένες *συμπεριφορές* (*behaviors*) της JavaScript.

Οι συμπεριφορές αυτές είναι μικρά κομμάτια κώδικα με τα οποία μπορούμε να προσθέσουμε ενδιαφέροντα και ζωντανά εφέ στις ιστοσελίδες μας, όπως από το να εμφανίζουμε παράθυρα φυλλομετρητή όταν γίνεται κλικ σε κάποιον σύνδεσμο (link) έως και να ελέγχουμε διάφορα animations χρησιμοποιώντας το timeline του Dreamweaver.

Το *Code panel* περιέχει χρήσιμα εργαλεία και επιλογές. Από εδώ μπορούμε να δούμε ολόκληρη τη δομή της σελίδας μας χρησιμοποιώντας τον Tag Inspector, ο οποίος εμφανίζει τη σελίδα με τη μορφή ενός δένδρου και μπορούμε να επέμβουμε απευθείας στα διάφορα tags χωρίς να είμαστε αναγκασμένοι να πάμε σε κατάσταση πρωτογενούς κώδικα (Code view). Με την καρτέλα *Snippets* μπορούμε εύκολα να αποθηκεύσουμε και να διατηρήσουμε κομμάτια κώδικα που ίσως χρειασθεί να ξαναχρησιμοποιήσουμε σ' άλλες σελίδες ή και σ' άλλα web sites. Από έτοιμα meta tags έως και συναρτήσεις της JavaScript που υπολογίζουν την περιφέρεια ενός κύκλου, υπάρχει πάντα κάτι για τον καθένα. Η καρτέλα *Reference* περιέχει σύντομους οδηγούς αναφοράς γι' όλες τις υποστηριζόμενες γλώσσες server.

Το *Application panel* είναι αυτό στο οποίο θα ξοδέψουμε τον περισσότερο χρόνο μας όταν θα δημιουργήσουμε τις αλληλεπιδράσεις στην πλευρά του server (server-side interactions) του δικτυακού μας τόπου (web site). Αυτό το panel περιέχει τέσσερις καρτέλες (tabs): Databases, Bindings, Server Behaviors και Components. Από την καρτέλα *Databases* μπορούμε να δούμε τη δομή (structure) και τη διάταξη (layout) όποιας βάσης δεδομένων έχουμε δημιουργήσει στα web sites μας. Από εδώ μπορούμε επίσης να προσθέσουμε καινούργιες συνδέσεις με βάσεις δεδομένων και να τις κάνουμε άμεσα διαθέσιμες.

Όταν ανακτούμε δυναμικά δεδομένα για την ιστοσελίδα μας, θα πρέπει να δημιουργήσουμε αυτό που είναι γνωστό ως *Data Binding*, το οποίο παρέχει τον ουσιαστικό σύνδεσμο ανάμεσα στο περιβάλλον ανάπτυξης του Dreamweaver και στα δεδομένα μας. Από την καρτέλα αυτή μπορούμε να δημιουργήσουμε πολλά διαφορετικά Data Bindings για να

ανακτούμε πληροφορίες από βάσεις δεδομένων, πεδία φόρμας, cookies και πολλά άλλα.

Στο Application panel έχουμε πρόσβαση και σε επιπλέον συμπεριφορές (behaviors), οι οποίες όμως δεν προσθέτουν κώδικα JavaScript στις ιστοσελίδες μας. Εισάγουν server-side code για την τελική αλληλεπίδραση (interaction) ανάμεσα στο web site και τον χρήστη (επισκέπτη). Αυτή η καρτέλα (Server Behaviors) είναι context-sensitive, πράγμα που σημαίνει ότι αλλάζει ανάλογα με το server model που χρησιμοποιούμε, όπως για παράδειγμα PHP ή CFML κοκ.

Η καρτέλα *Components* δεν χρησιμοποιείται όταν εργαζόμαστε με PHP ιστοσελίδες αλλά προορίζεται ειδικά για άλλα server models, όπως είναι το ASP.

Από το *Files panel* και την καρτέλα Site έχουμε πλήρη πρόσβαση στο τοπικό (local) και στο απομακρυσμένο (remote) σύστημα αρχείων. Μπορούμε να μεταφέρουμε αρχεία και φακέλους (καταλόγους) προς και από τον κατάλογο εργασίας μας χωρίς την ανάγκη ύπαρξης ενός ξεχωριστού προγράμματος FTP (File Transfer Protocol).

Η καρτέλα *Assets* καταγράφει όλα τα εξωτερικά αρχεία, συνδέσμους (links) και χρώματα που χρησιμοποιούνται όταν δημιουργούμε το site μας, τα οποία είναι διαθέσιμα για εύκολη αναφορά, τροποποίηση και επαναχρησιμοποίηση.

Από το *Answers panel* μπορούμε να έχουμε άμεση πρόσβαση σε tutorials και άλλα χρήσιμα βοηθήματα. Αν κάνουμε κλικ στο πλήκτρο Update μπορούμε να συνδεθούμε με το site της Macromedia και να κατεβάσουμε ότι καινούργια στοιχεία υπάρχουν.

Από το panel *Properties* μπορούμε να ελέγχουμε απευθείας ειδικά στοιχεία των HTML tags και των σελίδων μας χωρίς να χρειασθεί να πάμε σε πηγαίο κώδικα. Αυτό το panel αλλάζει (προσαρμόζεται) συνέχεια ανάλογα με το στοιχείο που είναι επιλεγμένο στη σελίδα.

6.2 TO DREAMWEAVER MX ΚΑΙ Η PHP

Αν και η PHP υποστηρίζει πάρα πολλά συστήματα διαχείρισης βάσεων δεδομένων (DMBS, Database Management Systems), η πιο δημοφιλής βάση δεδομένων και αυτή που θα χρησιμοποιήσουμε είναι η MySQL. Αφού έχουμε σιγουρευτεί ότι ο web server στον οποίο θα στήσουμε την εφαρμογή μας υποστηρίζει την PHP, θα πρέπει να κάνουμε μια σύντομη και απλή δοκιμή πριν συνεχίσουμε.

Δημιουργούμε ένα αρχείο με το Σημειωματάριο (Notepad) των Windows, το οποίο αποθηκεύουμε με το όνομα test.php και με το εξής περιεχόμενο:
<?php phpinfo(); ?>

Το φορτώνουμε (upload) στον web server και το καλούμε από τον φυλλομετρητή μας. Αν όλα δουλεύουν κανονικά, θα πρέπει να δούμε μια

μεγάλη ιστοσελίδα μ' ένα πλήθος από τεχνικά χαρακτηριστικά για την εγκατεστημένη PHP.

6.3 ΞΕΚΙΝΗΜΑ ΜΕ ΤΟ DREAMWEAVER MX

Τώρα που έχουμε έναν web server που εκτελεί την PHP και έναν MySQL database server, είναι καιρός να ασχοληθούμε με το κυρίως θέμα μας που είναι ο σχεδιασμός PHP ιστοσελίδων με το Dreamweaver MX. Θα ξεκινήσουμε δίνοντας στο Dreamweaver MX όλες τις απαραίτητες πληροφορίες που θα χρειασθεί για να μπορέσει να συνεργασθεί με το περιβάλλον ανάπτυξης, δημιουργώντας έναν ορισμό του δικτυακού τύπου (Site Definition).

Αφού θα έχει ορισθεί το site, θα δούμε πώς μπορούμε να δημιουργήσουμε μια σύνδεση με τη βάση δεδομένων (database connection) μέσα στο Dreamweaver, ώστε να μπορούμε να έχουμε πρόσβαση στη βάση δεδομένων της MySQL που έχουμε δημιουργήσει νωρίτερα. Ο ορισμός ενός δικτυακού τύπου (site definition) είναι το βασικό σημείο αρχής κατά την ανάπτυξη ενός οποιουδήποτε web site. Με τις πληροφορίες που θα παρέχονται εδώ, το Dreamweaver θα είναι σε θέση να εντοπίζει τη δομή των φακέλων (directory structure) του site, να παρακολουθεί όλους τους εσωτερικούς και εξωτερικούς υπερσυνδέσμους (hyperlinks) και να διατηρεί μια λίστα όλων των εικόνων, των χρωμάτων και των εξωτερικών αρχείων.

Το Dreamweaver χρησιμοποιεί δύο μεθόδους για τον ορισμό ενός site. Υπάρχει η βασική μέθοδος, η οποία απασχολεί έναν βήμα-βήμα οδηγό (step-by-step Wizard) για να μας καθοδηγήσει στην όλη διαδικασία και υπάρχει μια προηγμένη διαδικασία ορισμού, με την οποία έχουμε πλήρη έλεγχο για κάθε θέμα που αφορά τον ορισμό του site. Το καλύτερο είναι να δουλέψει κάποιος αρχικά με τον οδηγό και μετά να κάνει τις αναγκαίες τροποποιήσεις με την προηγμένη διαδικασία ορισμού.

6.4 Ο ΒΑΣΙΚΟΣ ΟΔΗΓΟΣ ΓΙΑ ΤΟΝ ΟΡΙΣΜΟ ΤΟΥ SITE

Υπάρχουν τέσσερις διαφορετικοί τρόποι για να ορίσουμε το site μας με τον οδηγό ορισμού :

- Δημιουργία και δοκιμή τοπικά.
- Δημιουργία τοπικά και φόρτωση (upload) για δοκιμή στον απομακρυσμένο server.
- Δημιουργία απευθείας μέσω δικτύου (network).
- Δημιουργία απευθείας μέσω FTP.

Θα δούμε την κάθε περίπτωση ξεχωριστά. Τα δύο πρώτα βήματα στον ορισμό ενός site με τον βασικό οδηγό ορισμού είναι τα ίδια και στις

τέσσερις περιπτώσεις. Το πρώτο βήμα είναι η ονομασία του site και προς τον σκοπό αυτό πηγαίνουμε στο μενού *Site* και επιλέγουμε *New Site...* Θα ξεκινήσει ο οδηγός ορισμού του site (*Site Definition Wizard*) και στο πρώτο πλαίσιο διαλόγου θα πρέπει να δώσουμε ένα όνομα στο site μας, όπως *GuestBook*, το οποίο όνομα είναι καθαρά τυπικό για την αναγνώριση του site από εμάς και δεν πρόκειται να χρησιμοποιηθεί πουθενά.

Αφού έχουμε καθορίσει ένα όνομα για το site θα πρέπει να ορίσουμε την τεχνολογία server που θα χρησιμοποιήσει το site στο δεύτερο βήμα του οδηγού. Πρώτα απ' όλα θα πρέπει να αποφασίσουμε αν το site μας θα χρησιμοποιεί δυναμικές τεχνολογίες, όπως είναι η ASP, JSP, CFML ή PHP, ή απλά θα επιστρέφει στατικές HTML σελίδες. Επιλέγουμε το κατάλληλο πλήκτρο επιλογής (*Yes, I want to use a server technology*) και από την πτυσσόμενη λίστα που θα εμφανισθεί αμέσως από κάτω (*Which server technology?*), επιλέγουμε *PHP MySQL*.

Μπορούμε τώρα να προχωρήσουμε για να δώσουμε στο Dreamweaver περισσότερες πληροφορίες σχετικά με τα αρχεία που θα δουλέψουμε. Εδώ θα καθορίσουμε πού θα αποθηκευθούν τα πηγαία αρχεία που θα δημιουργήσουμε και επίσης αν ο web server θα είναι ένα τοπικό μηχάνημα ή ένα απομακρυσμένο μηχάνημα σ' ένα τοπικό δίκτυο ή ένα απομακρυσμένο μηχάνημα που χρειάζεται μια ενεργή σύνδεση στο Internet.

Τα επόμενα βήματα του οδηγού από εδώ και πέρα θα διαφέρουν ανάλογα με την επιλογή που θα κάνουμε σ' αυτό το βήμα. Η πρώτη επιλογή είναι η *Edit and test locally (my testing server is on this computer)*, την οποία θα πρέπει να επιλέξουμε αν όλα δουλεύουν από το ίδιο μηχάνημα. Αν έχουμε εγκαταστήσει την PHP και την MySQL στο τοπικό μας μηχάνημα, τότε αυτή είναι η καλύτερη επιλογή που μπορούμε να κάνουμε. Αν, όμως, έχουμε εγκαταστήσει την PHP και την MySQL σ' ένα άλλο μηχάνημα στο οποίο έχουμε πρόσβαση είτε μέσω του Internet είτε μέσω μιας σύνδεσης δικτύου, τότε θα πρέπει να επιλέξουμε τη δεύτερη επιλογή *Edit locally, then upload to remote testing server*.

Αν επιλέξουμε το πρώτο πλήκτρο επιλογής, η μόνη πληροφορία που θα χρειασθεί ο οδηγός θα είναι ο φάκελος όπου θα αποθηκεύσουμε όλα τα αρχεία τα σχετικά με το web site. Μπορούμε να επιλέξουμε κάποιον φάκελο κάνοντας κλικ στο εικονίδιο φακέλου ή να επιλέξουμε τον φάκελο που έχει ήδη αποφασίσει να δημιουργήσει για εμάς το Dreamweaver και ο οποίος έχει το όνομα του site που δώσαμε στο πρώτο βήμα του οδηγού.

Στο επόμενο βήμα θα πρέπει να δώσουμε την web address όπου θα βλέπουμε τις ιστοσελίδες μας. Εφόσον όλα γίνονται τοπικά, αυτή η διεύθυνση μπορεί να είναι μια από τις <http://localhost> ή <http://127.0.0.1>, που είναι ακριβώς ίδιες και δείχνουν προς την ίδια κατεύθυνση.

Μπορούμε μετά να κάνουμε κλικ στο πλήκτρο *Test URL* ώστε να διαπιστώσουμε αν όλα δουλεύουν κανονικά. Στο επόμενο βήμα θα πρέπει να επιλέξουμε *Yes* αν θέλουμε να μεταφερθούν τα αρχεία μας σ' έναν απομακρυσμένο server ή *No*, αν θέλουμε να αποθηκευθούν τα αρχεία μας τοπικά, που είναι και το πιθανότερο.

Η δεύτερη επιλογή *Edit locally, then upload to remote testing server* είναι η προτιμότερη στην περίπτωση που σχεδιάζουμε τις σελίδες μας και τις δημοσιεύουμε (upload) σ' έναν server που βρίσκεται σ' ένα διαφορετικό μηχάνημα, είτε σ' ένα τοπικό δίκτυο ή στο Internet. Και μ' αυτήν την επιλογή, η μόνη πληροφορία που θα χρειασθεί ο οδηγός θα είναι ο φάκελος όπου θα αποθηκεύσουμε τοπικά όλα τα αρχεία τα σχετικά με το web site.

Στο επόμενο βήμα του οδηγού θα πρέπει να απαντήσουμε στην ερώτηση *How do you connect to your testing server?* επιλέγοντας μία από τέσσερις επιλογές (*I'll set this up later, FTP, Local/Network, RDS*). Η επιλογή *RDS* (*Remote Development Service*) είναι για χρήση μ' έναν *ColdFusion server*. Αν επιλέξουμε *FTP*, που είναι το πιθανότερο, θα εμφανισθούν τέσσερα πλαίσια κειμένου όπου θα πρέπει να καταχωρήσουμε το hostname ή την *FTP address* του server, το όνομα του φακέλου στον server όπου θα αποθηκευθεί το web site και τέλος το login (user name) και το password για να μπορέσει να γίνει η *FTP* σύνδεση.

Έχουμε τη δυνατότητα να αποθηκεύσουμε τους κωδικούς για την *FTP* σύνδεση αν επιλέξουμε το πλαίσιο ελέγχου *Save* και επίσης να κάνουμε κλικ στο πλήκτρο *Test Connection* ώστε να διαπιστώσουμε αν η σύνδεση λειτουργεί κανονικά. Η τρίτη επιλογή είναι η *Edit directly on remote testing server using local network* και μπορούμε να την επιλέξουμε όταν κανένα από τα αρχεία με τα οποία δουλεύουμε δεν θα αποθηκευθούν στο τοπικό μηχάνημα. Θα πρέπει να υπάρχει ένα δεύτερο μηχάνημα που να ενεργεί ως web server σ' ένα τοπικό δίκτυο (*LAN*) και θέλουμε όλα τα αρχεία να αποθηκεύονται σ' αυτό το μηχάνημα.

Η τέταρτη επιλογή είναι η *Edit directly on remote testing server using FTP or RDS* και μπορούμε να την επιλέξουμε όταν θέλουμε όλα τα αρχεία να αποθηκεύονται απευθείας μέσω του πρωτοκόλλου *FTP* (ή του *RDS*) σ' έναν απομακρυσμένο server και όχι στο τοπικό μηχάνημα.

6.5 Ο ΠΡΟΗΓΓΜΕΝΟΣ ΟΡΙΣΜΟΣ SITE

Αν και ο οδηγός ορισμού site είναι πολύ απλός και φιλικός προς τον χρήστη, υπάρχουν πολλές λεπτομέρειες που δεν καλύπτει και έτσι είναι απαραίτητο να δούμε και τον προηγμένο ορισμό site. Επιλέγουμε την καρτέλα *Advanced* του πλαισίου διαλόγου *Site Definition* και βλέπουμε να εμφανίζονται επτά κατηγορίες στην αριστερή πλευρά. Η κατηγορία *Local Info* περιέχει στοιχεία που μας είναι ήδη γνωστά, όπως είναι το

όνομα του site (Site Name) και ο τοπικός φάκελος αποθήκευσης των αρχείων (Local Root Folder). Καινούργιο στοιχείο είναι ο φάκελος αποθήκευσης των εικόνων (Default Images Fodler), όπου μπορούμε να επιλέξουμε να αποθηκεύονται όλες οι εικόνες που χρησιμοποιούνται στο site.

Η κατηγορία *Remote Info* έχει να κάνει με τις ρυθμίσεις του απομακρυσμένου server, όπως είναι ο τρόπος πρόσβασης (π.χ. FTP), η διεύθυνση του FTP Host, το Host Directory και το Login και το Password για να μπορέσουμε να συνδεθούμε. Καινούργιο στοιχείο είναι το πλαίσιο ελέγχου *Automatically upload files to server on save*, όπου αν το επιλέξουμε, τα αρχεία μας θα στέλνονται (upload) στον web server κάθε φορά που θα τα αποθηκεύουμε τοπικά. Αν συνδεόμαστε με τον απομακρυσμένο server μέσω FTP, θα πρέπει να υπάρχει μια ενεργή σύνδεση με το Internet.

Η κατηγορία *Testing Server* περιέχει επιλογές που τις έχουμε ήδη καλύψει σχετικά με την δοκιμή λειτουργίας του server. Στην κατηγορία *Cloaking* μπορούμε να επιλέξουμε συγκεκριμένους τύπους αρχείων αλλά και φακέλους που δεν θέλουμε να γίνονται αυτόματα upload όταν κάνουμε upload όλο το site.

Στην κατηγορία *Design Notes* μπορούμε να επιλέξουμε αν θα υπάρχει η δυνατότητα για σημειώσεις (σχόλια) κατά την ανάπτυξη του site καθώς επίσης και αν θέλουμε να μοιραζόμαστε αυτές τις σημειώσεις με άλλους προγραμματιστές του site. Κάνοντας κλικ στο πλήκτρο *Clean Up...* μπορούμε να διαγράψουμε όλες τις σημειώσεις ενός site.

Στην κατηγορία *Site Map Layout* μπορούμε να δούμε διαγραμματικά τη δομή του web site. Μπορούμε να δούμε τη σχέση ανάμεσα στις διάφορες σελίδες καθώς και να βρούμε πολύ εύκολα αν υπάρχουν σπασμένα ή ορφανά links, δηλαδή σελίδες που δεν συνδέονται με καμία άλλη σελίδα.

6.6 ΔΗΜΙΟΥΡΓΙΑ ΤΗΣ ΣΥΝΔΕΣΗΣ ΜΕ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Για να ξεκινήσουμε τη δημιουργία μιας σύνδεσης με τη βάση δεδομένων (database connection), θα πρέπει να πάμε στο *Application panel* και να επιλέξουμε την καρτέλα *Databases*. Προσοχή : αν δίπλα στο πλήκτρο – υπάρχει το κείμενο *Document Type : HTML*, τότε θα πρέπει να κλείσουμε το τρέχον έγγραφο και να δημιουργήσουμε ένα καινούργιο αρχείο PHP.

Αυτό γίνεται καθώς το Dreamweaver αλλάζει (προσαρμόζει) τον τρόπο εργασίας του ανάλογα με τις επιλογές μας και κάνει διαθέσιμα μόνο τα εργαλεία που είναι απαραίτητα κάθε φορά. Έτσι, αν το έγγραφό μας είναι ένα απλό HTML αρχείο, το Dreamweaver δεν θα μας επιτρέψει πρόσβαση στις δυναμικές του συμπεριφορές, όπως είναι η σύνδεση με τη βάση δεδομένων.

Αφού ανοίξουμε μια κανούργια δυναμική ιστοσελίδα με PHP κωδικοποίηση, πηγαίνουμε πάλι στο Application panel και κάνουμε κλικ στο πλήκτρο + και μετά στο *MySQL Connection*. Θα εμφανισθεί το πλαίσιο διαλόγου *MySQL Connection* με τα εξής πέντε πεδία κειμένου : Connection Name, MySQL Server, User Name, Password και Database. Υπάρχει μια άτυπη σύμβαση για την ονομασία των συνδέσεων με τις βάσεις δεδομένων, όπου χρησιμοποιούμε το πρόθεμα (prefix) *conn*. Στην περίπτωση μας μπορούμε να ονομάσουμε τη σύνδεση που δημιουργούμε ως *connGuestBook*.

Δίνουμε μετά το όνομα (hostname) για τον MySQL Server, που είναι συνήθως ίδιο με το όνομα του web server, και το username και password, που είναι όμως ειδικά για σύνδεση με τον MySQL Server. Το hostname για τον MySQL Server είναι το *localhost* για όσους χρησιμοποιούν τις υπηρεσίες του Πανελληνίου Σχολικού Δικτύου (EDUnet).

Αφού έχουμε δώσει όλες τις παραπάνω πληροφορίες, κάνουμε κλικ στο πλήκτρο *Select...* για να επιλέξουμε τη βάση δεδομένων με την οποία θα δουλέψουμε ή μπορούμε και να γράψουμε το όνομά της απευθείας στο πλαίσιο κειμένου *Database*. Θα πρέπει μετά να κάνουμε κλικ στο πλήκτρο *Test* για να γίνει δοκιμή της σύνδεσης. Αν όλα δουλέψουν κανονικά, το Dreamweaver MX θα ενημερώσει αυτόματα την καρτέλα Databases του Application panel, όπου θα μπορούμε να δούμε όλους τους πίνακες (tables) που υπάρχουν στη βάση δεδομένων που έχουμε επιλέξει.

6.7 Ο ΠΙΝΑΚΑΣ GUESTBOOK

Ο μοναδικός πίνακας της βάσης δεδομένων μας περιέχει τα στοιχεία που ζητάμε να καταχωρήσει ένας χρήστης. Η εντολή δημιουργίας του πίνακα στην MySQL είναι η εξής :

```
CREATE TABLE guestbook (  
ID tinyint(11) NOT NULL auto_increment,  
firstName varchar(15) NOT NULL default ' ',  
lastName varchar(20) NOT NULL default ' ',  
city varchar(20) NOT NULL default ' ',  
email varchar(20) NOT NULL default ' ',  
profession varchar(20) NOT NULL default ' ',  
PRIMARY KEY (ID)  
) TYPE=MyISAM COMMENT='GuestBook Table';
```

6.8 Η ΙΣΤΟΣΕΛΙΔΑ ADD_USER_RECORD.PHP

Είναι η μοναδική ιστοσελίδα που θα δημιουργήσουμε. Η πρώτη μας δουλειά είναι να τοποθετήσουμε μια φόρμα στην ιστοσελίδα επιλέγοντας *Form* από το μενού *Insert*. Της δίνουμε το όνομα *userform* και

παρατηρούμε το κόκκινο περίγραμμα που εμφάνισε αυτόματα το Dreamweaver MX για να περιβάλλει τα στοιχεία της φόρμας.

Μετά δημιουργούμε έναν πίνακα (table) μέσα στη φόρμα με 2 στήλες και τόσες σειρές όσα είναι τα πεδία του πίνακα guestbook. Στην αριστερή στήλη γράφουμε περιγραφές για τα πεδία, τα οποία τοποθετούμε αντίστοιχα στη δεξιά στήλη του πίνακα. Η εντολή δημιουργίας πεδίων κειμένου για τη φόρμα είναι η *Text Field* του υπομενού *Form Objects* του μενού *Insert*.

Αφού έχουμε εισάγει ένα πεδίο κειμένου, κάνουμε κλικ πάνω του για να δούμε τις ιδιότητές του στην καρτέλα *Properties*. Αλλάζουμε το όνομά του ώστε να είναι ακριβώς ίδιο με το όνομα του αντίστοιχου πεδίου του πίνακα guestbook και κάνουμε ακόμη ρυθμίσεις και στις ιδιότητες *Char Width*, *Max Chars*, ενώ στην ενότητα *Type* θα πρέπει να είναι επιλεγμένο το πλήκτρο επιλογής *Single line*.

Για να δημιουργήσουμε ένα πεδίο με λίστα επιλογής, επιλέγουμε *List/Menu* από το υπομενού *Form Objects* του μενού *Insert*. Θα δημιουργηθεί μια κενή λίστα επιλογής. Στις ιδιότητές της της δίνουμε το όνομα *title* και κάνουμε κλικ

στο πλήκτρο *List Values...* ώστε να καταχωρήσουμε στο πλαίσιο διαλόγου *List Values* τα στοιχεία της λίστας με τις τιμές τους.

Τέλος, θα πρέπει να δημιουργήσουμε το πλήκτρο υποβολής (*submit*) της φόρμας. Επιλέγουμε *Button* από το υπομενού *Form Objects* του μενού *Insert*.

Στο πεδίο *Label* δίνουμε στο πλήκτρο την ονομασία (ετικέτα) *Υποβολή των Στοιχείων*, ενώ στην ενότητα *Action* θα πρέπει να είναι επιλεγμένο το πλήκτρο επιλογής *Submit form*.

Για να εμφανίσουμε επεξηγηματικά μηνύματα στον χρήστη όταν αυτός βρεθεί σ' ένα πεδίο της φόρμας, επιλέγουμε το πεδίο που θέλουμε και πάμε στην καρτέλα *Behaviors* του πάνελ *Design*. Κάνουμε κλικ στο πλήκτρο + και επιλέγουμε *Set Text of Status Bar* από το υπομενού *Set Text*. Στο πλαίσιο διαλόγου που θα εμφανισθεί και στο πεδίο κειμένου *Message* γράφουμε το μήνυμα (κείμενο) που θέλουμε να εμφανίζεται στη γραμμή κατάστασης (*status bar*) όταν οι δρομέας βρεθεί μέσα στο πεδίο κειμένου της φόρμας. Αφού κάνουμε κλικ στο OK, αλλάζουμε το συμβάν στη λίστα *Events* σε *onFocus* ώστε να λαμβάνει χώρα όταν ο χρήστης βρεθεί στο πεδίο είτε με το πλήκτρο *tab* ή αν έχει κάνει κλικ με το ποντίκι.

Θα δούμε τώρα πώς μπορούμε να κάνουμε έναν έλεγχο (*validation*) των στοιχείων που θα καταχωρήσει ο χρήστης. Επιλέγουμε το πλήκτρο υποβολής (*submit*) και πάμε στην καρτέλα *Behaviors* του πάνελ *Design*. Κάνουμε κλικ στο πλήκτρο + και επιλέγουμε *Validate Form*. Στο πλαίσιο διαλόγου *Validate Form* θα εμφανισθούν όλα τα πεδία της φόρμας στην περιοχή *Named Fields* και μπορούμε να τα επιλέξουμε ένα-ένα.

Μπορούμε να ορίσουμε ένα πεδίο να είναι υποχρεωτικό, επιλέγοντας το πλαίσιο ελέγχου *Required* και από την περιοχή *Accept* μπορούμε να επιλέξουμε ένα από τα πλήκτρα επιλογής *Anything*, *Number*, *Email Address* ή *Number from ... to ...*

Θα δούμε τώρα πώς θα καταχωρηθούν τα στοιχεία της φόρμας σε μια εγγραφή του πίνακα *guestbook* της βάσης δεδομένων όταν θα γίνει υποβολή της φόρμας. Θα πρέπει πρώτα να κάνουμε ξανά έλεγχο για το αν είναι ενεργή η σύνδεση με τη βάση δεδομένων (*database connection*) στην καρτέλα *Databases* του πάνελ *Application*. Αν η σύνδεση με τη βάση δεδομένων λειτουργεί, επιλέγουμε την καρτέλα *Server Behaviors* του πάνελ *Application*, κάνουμε κλικ στο πλήκτρο + και επιλέγουμε *Insert Record* από το μενού. Θα εμφανισθεί το πλαίσιο διαλόγου *Insert Record*.

Στη λίστα *Submit Values From* θα εμφανισθούν όλες οι φόρμες που υπάρχουν στην τρέχουσα ιστοσελίδα και επιλέγουμε την φόρμα από την οποία θα γίνει η υποβολή των στοιχείων (*userform*). Στη λίστα *Connection* επιλέγουμε τη σύνδεση με τη βάση δεδομένων (*connGuestBook*) και στη λίστα *Insert Table* επιλέγουμε τον αντίστοιχο πίνακα *guestbook* της βάσης δεδομένων.

Στην περιοχή κειμένου *Columns* θα εμφανισθούν όλα τα πεδία της φόρμας της ιστοσελίδας με τα αντίστοιχα πεδία του πίνακα *guestbook* της βάσης δεδομένων με τα οποία θα γίνει η αντιστοίχιση. Αν έχουμε δώσει τα ίδια ονόματα, το Dreamweaver MX θα έχει ήδη κάνει την αντιστοίχιση, ενώ αν έχουμε κάνει κάποιο λάθος στην ονομασία των πεδίων ή απλά δεν θέλουμε να είναι ίδιο το όνομα κάποιου πεδίου στη φόρμα και στη βάση δεδομένων, μπορούμε να επιλέξουμε το πεδίο της φόρμας από τη λίστα *Value* και το αντίστοιχο πεδίο του πίνακα της βάσης δεδομένων από τη λίστα *Submit As*. Τέλος, θα πρέπει να ορίσουμε την ιστοσελίδα στην οποία θα κατευθύνουμε τον χρήστη αφού θα έχουν εισαχθεί τα δεδομένα της φόρμας. Κάνουμε κλικ στο πλήκτρο *Browse...* και μετά στο πλήκτρο *Parameters* στο επόμενο πλαίσιο διαλόγου που θα εμφανισθεί. Στο πλαίσιο διαλόγου *Parameters* μπορούμε να ορίσουμε τις παραμέτρους που θα προστεθούν στο URL της σελίδας στην οποία θα κατευθύνουμε τον χρήστη και είναι ένας τρόπος επικοινωνίας (μεταφοράς πληροφορίας) από σελίδα σε σελίδα.

Κάνουμε κλικ στο πλήκτρο + και γράφουμε *email* για το όνομα (*name*) της παραμέτρου και *\$.HTTP_POST_VARS['email']* για την τιμή (*value*) της παραμέτρου. Κάνουμε κλικ στο OK και επιλέγουμε την ιστοσελίδα *guestbook_details.php*. Για παράδειγμα, αν ένας χρήστης με e-mail a@a.gr υπέβαλε στοιχεία στην φόρμα, τότε θα κατευθυνθεί στην ιστοσελίδα:

guestbook_details.php?email=a@a.gr

Μπορούμε τώρα να δημοσιεύσουμε (upload) την ιστοσελίδα στον server, να την καλέσουμε από έναν φυλλομετρητή και να καταχωρήσουμε κάποια δοκιμαστικά στοιχεία. Θα πρέπει μετά να πάμε στο κατάλληλο πρόγραμμα (phpmyadmin) ώστε να δούμε αν τα δεδομένα καταχωρήθηκαν σωστά και στα σωστά πεδία. Αν υπάρχει κάποιο λάθος, μπορούμε να κάνουμε διορθώσεις στο Insert Record behavior.

6.9 Η ΣΕΛΙΔΑ ΜΑΣ

ΑΡΧΙΚΗ ΣΕΛΙΔΑ

Αρχική Τουριστικά Γραφεία Ιστορία Διαμονή Χάρτης Επικοινωνία



Τουριστικός Οδηγός Πάτρας

Αρχική

Πάτρα, το τρίτο μεγαλύτερο αστικό χώρο και την πρωτεύουσα του νομού Αχαΐας, που βρίσκεται στο βόρειο τμήμα της Πελοποννήσου, 215 χιλιόμετρα δυτικά της Αθήνας. Η πόλη είναι χτισμένη στους πρόποδες του βουνού Παναχαϊκό, με θέα στον Πατραϊκό κόλπο.

Η μητροπολιτική περιοχή Πάτρας είναι ένα αστικό κέντρο του 222.460 κάτοικοι. [2] Ο οικισμός πυρήνας έχει μια ιστορία που εκτείνεται από τέσσερις χιλιάδες. Κατά τη ρωμαϊκή εποχή είχε εξελιχθεί σε κοσμοπολίτικο κέντρο της ανατολικής Μεσογείου, ενώ σύμφωνα με τη χριστιανική παράδοση, ήταν επίσης ο τόπος του Αγίου Ανδρέα μαρτύριου. Μεταγωγλιτισμένο Πύλη Ελλάδα προς τη Δύση, η Πάτρα είναι ένα εμπορικό κέντρο, ενώ το λιμάνι της απασχολημένος είναι ένα κομβικό σημείο για το εμπόριο και την επικοινωνία με την Ιταλία και την υπόλοιπη Δυτική Ευρώπη.

Η πόλη έχει δύο δημόσια πανεπιστήμια και ένα Τεχνολογικό Ινστιτούτο και φιλοξενεί έναν μεγάλο πληθυσμό φοιτητών και την καθιστά σημαντικό επιστημονικό κέντρο με έναν τομέα αριστείας στην τεχνολογική εκπαίδευση. Η Αντιρρίου γέφυρα Ρίου συνδέει το ανατολικότερο προάστιο της Πάτρας, Ρίο, με την πόλη του Αντιρρίου, που συνδέει το Πελοπόννησο με την ηπειρωτική Ελλάδα. Κάθε άνοιξη, η πόλη φιλοξενεί ένα από τα φαντασμαγορικά καρναβάλια της Ευρώπης μεγαλύτερο και πιο πολύ; αξιοσημείωτα χαρακτηριστικά του Πατρινού Καρναβαλιού περιλαμβάνουν τους μαμμούθ μεγέθους σατιρικά όργανα και εξωφρενικές μπάλες και τις παρελάσεις, που απολαμβάνουν οι εκατοντάδες χιλιάδες επισκέπτες σε ένα ευχάριστο μεσογειακό κλίμα, με σχετικά δροσερό αλλά υγρό καλοκαίρια και μάλλον ήπιους χειμώνες.

Η Πάτρα είναι επίσης γνωστή για την υποστήριξη ιθαγενών πολιτιστική σκηνή δραστηριοποιείται κυρίως στις τέχνες του θεάματος και της σύγχρονης αστικής λογοτεχνίας? Ήταν Πολιτιστική Πρωτεύουσα της Ευρώπης 2006.

Χρήσιμα links

1. [Δήμος Πατρέων](#)
2. [Καρναβάλι Πάτρας](#)
3. [Delivery Πάτρας](#)
4. [Πανεπιστήμιο Πάτρας](#)
5. [ΤΕΙ Πάτρας](#)



Αρχική Τουριστικά Γραφεία Ιστορία Διαμονή Χάρτης Επικοινωνία
Patras Guide © 2010

Πιο πάνω βλέπουμε την αρχική σελίδα του τόπου που δημιουργήσαμε. Στο πάνω μέρος της Σελίδας υπάρχει η βασική γραμμή διαχείρισης από την οποία μπορούμε να δούμε τι περιλαμβάνει η σελίδα μας και πιο συγκεκριμένα τι υπηρεσίες και πληροφορίες μπορεί να προσφέρει.

Επιπλέον στη μέση της σελίδας βλέπουμε την εκάστοτε επιλογή που έχουμε κάνει από τη γραμμή διαχείρισης για να αντλήσουμε πληροφορίες για αυτά που μας ενδιαφέρουν από τη σελίδα. Στην αριστερή πλευρά μπορούμε να δούμε χρήσιμους συνδέσμους που παρέχει η σελίδα μας προκειμένου οι ενδιαφερόμενοι να έχουν άμεση πρόσβαση, και κάτω από αυτούς τους συνδέσμους, διάφορες εικόνες που χαρακτηρίζουν την πόλη της Πάτρας. Τέλος στο κάτω μέρος βλέπουμε εκ νέου την γραμμή διαχείρισης, για μεγαλύτερη ευκολία του χρήστη, καθώς και τον κατοχυρωμένο τίτλο της σελίδας μας.

nikosssdadasdasdasdasdasd asjndaasnnnnn

Menu Διαχειριστή

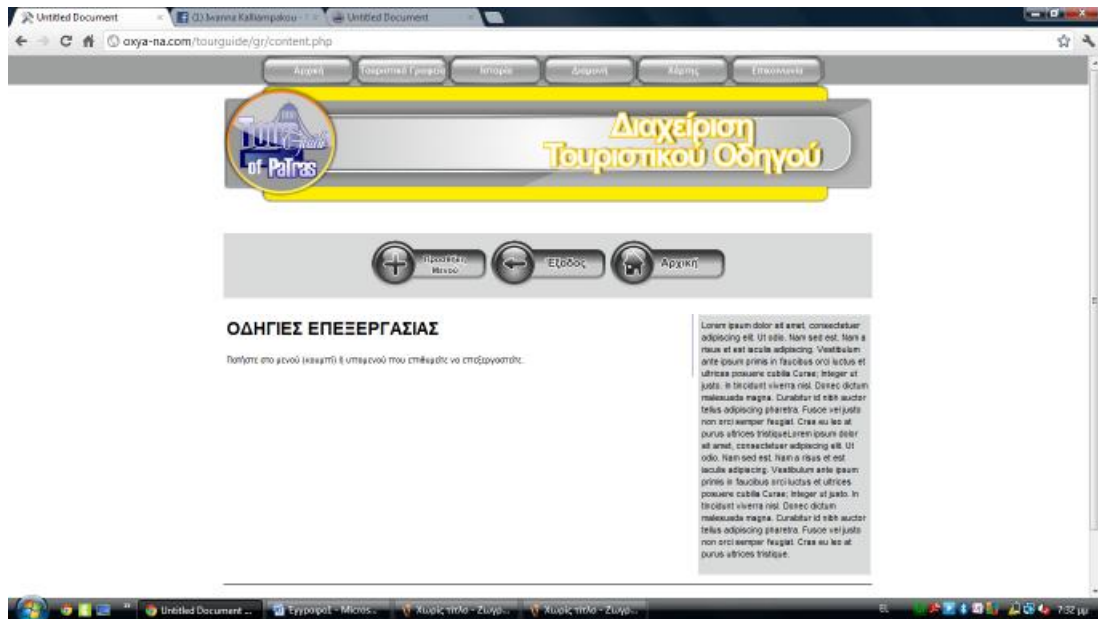
Welcome to the staff area, admin.

- Διαχείριση Ιστοσελίδας Ελληνικών
- Διαχείριση Ιστοσελίδας Αγγλικών
- Προσθήκη Διαχειριστή
- Αποσύνδεση

We only ask you to not remove "Design by Template World" and the link <http://www.templateworld.com> from the footer of the template. These templates are licensed under a Creative Commons Attribution 2.5 License. This means that you are free to modify the design to suit your tastes in any way you like, but you must include the provided link back to Template World.

Patras Guide © 2010

Στην επόμενη εικόνα βλέπουμε την σελίδα διαχείρισης του τόπου που έχουμε δημιουργήσει, και τις δυνατότητες επεξεργασίας που μας παρέχει το πρόγραμμα. Αυτές είναι η διαχείριση στα ελληνικά ή στα αγγλικά, η προσθήκη νέων διαχειριστών και η αποσύνδεση από την σελίδα. Επιλέγοντας τη διαχείριση στα ελληνικά φτάνουμε στην παραπάνω εικόνα.



Εδώ έχουμε τη δυνατότητα προσθέσουμε κάποιο νέο μενού στην σελίδα μας, ενώ μπορούμε επιλέγοντας ένα εικονίδιο από το ήδη υπάρχον μενού μπορούμε να προβούμε στην επεξεργασία που επιθυμούμε.



Add Subject

Subject name:

Position: ▼

Visible: No Yes

Content:

Επιλέγοντας την προσθήκη μενού φτάνουμε σε αυτή την εικόνα. Μπορούμε να προσθέσουμε τον τίτλο του θέματος που θέλουμε να δημιουργήσουμε, να επιλέξουμε την θέση που θα το τοποθετήσουμε στην γραμμή διαχείρισης, αν θα είναι φανερό στην σελίδα μας καθώς και να προσθέσουμε το κείμενο που επιθυμούμε. Έχουμε επίσης την επιλογή να το τοποθετήσουμε σαν κύριο μενού ή σαν υπομενού.

Εάν τέλος επιλέξουμε κάποιο από τα μενού ή υπομενού που ήδη υπάρχει στη σελίδα μας μπορούμε να προβούμε σε όποιου είδους επεξεργασία επιθυμούμε.



Edit Subject: Αρχική

Subject name:

Position:

Visible: No Yes

Φόρμα επικοινωνίας: No Yes

Για να ανεβάσετε φωτογραφίες πατήστε [εδώ](#)

Content:

HTML κώδικας

Στυλ Μορφή ... Γραμματ... M...

Στοιχίση Δεξιά

Πάτρα, το τρίτο μεγαλύτερο αστικό χώρο και την πρωτεύουσα του νομού Αχαΐας, που βρίσκεται στο βόρειο τμήμα της Πελοποννήσου, 215 χιλιόμετρα δυτικά της Αθήνας. Η πόλη είναι χτισμένη στους πρόποδες του βουνού Παναχαϊκό, με θέα στον Πατραϊκό κόλπο.

Η μητροπολιτική περιοχή Πάτρας είναι ένα αστικό κέντρο του 222.460 κάτοικοι. [2] Ο οικισμός πυρρήνας έχει μια ιστορία που εκτείνεται από τέσσερις χιλιάδες. Κατά τη ρωμαϊκή εποχή είχε εξελιχθεί σε κοσμοπολίτικο κέντρο της ανατολικής Μεσογείου, ενώ, σύμφωνα με τη χριστιανική παράδοση, ήταν επίσης ο τόπος του Αγίου Ανδρέα μαρτύριο. Μεταγωγτισμένο Πύλη Ελλάδα προς τη Δύση, η Πάτρα είναι ένα εμπορικό κέντρο, ενώ το λιμάνι της απασχολημένος είναι ένα κομβικό σημείο για το εμπόριο και την επικοινωνία με την Ιταλία και την υπόλοιπη Δυτική Ευρώπη.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην αρχή της εργασίας μας ξεκινήσαμε με την αναφορά μας στις βάσεις δεδομένων και την εκμάθηση του τι περιλαμβάνει μια ολοκληρωμένη βάση δεδομένων. Συνεχίζοντας την βιβλιογραφική μας ανασκόπηση, αναπτύξαμε τους κυριότερους Internet Web Servers οι οποίοι μας βοηθούν στην κατασκευή μιας ιστοσελίδας.

Στη συνέχεια ασχοληθήκαμε με την εκμάθηση της MySql και συνεχίσαμε με τη δημιουργία μίας βάσης δεδομένων MySQL. Μετά από την ολοκλήρωση αυτής της γλώσσας, συνεχίσαμε με τη γλώσσα Php. Μια γλώσσα ιδιαίτερα δύσκολη και απαιτητική η οποία θα ενσωματώνεται στην ήδη υπάρχουσα HTML σελίδα και θα τη συνδέει με τη βάση δεδομένων μας, από όπου και θα αντλεί πληροφορίες, ανάλογα με τις επιθυμίες του χρήστη.

Τέλος με την επιλογή του CMS Dreamweaver ολοκληρώσαμε την κατασκευή ενός τουριστικού οδηγού για την πόλη της Πάτρας. Το πρόγραμμα αυτό, το οποίο είναι ένας HTML editor που αναπτύχθηκε από την Macromedia, επιλέχτηκε για κυρίως για τα πλεονεκτήματα που παρουσιάζει κατά την κατασκευή μιας σελίδας. Κατ' αρχάς είναι της λογικής WYSIWYG (What You See Is What You Get), δηλ. αυτό που θα δούμε στο Web όταν θα εκδώσουμε (publish) μια σελίδα είναι ακριβώς αυτό που βλέπουμε στον editor κατά την κατασκευή της σελίδας. Ένα επιπλέον βασικό στοιχείο είναι πως για την λειτουργία του δεν απαιτείται καθόλου γνώση της γλώσσας HTML. Επιπλέον το συγκεκριμένο πρόγραμμα επιτρέπει διόρθωση του πηγαίου HTML κώδικα, ενώ έχει και ενσωματωμένα εργαλεία για εισαγωγή JavaScript, υποστηρίζει layers, style sheets και παρέχει δυνατότητες διαχείρισης ενός site.(απευθείας μεταφορά των Web pages στους servers, κλπ.).

ΒΙΒΛΙΟΓΡΑΦΙΑ

Hughes S., (2002), PHP οδηγός προγραμματισμού, Μτφ. Σαμαράς Γ., Εκδ. Γκιούρδας Μ., Αθήνα

Καράκος Α., (2007), Διαδίκτυο, παγκόσμιος ιστός και τεχνικές προγραμματισμού, Εκδ. Γκιούρδας Β

Λιακέας Γ. 2003 «Εισαγωγή στην Java 2» Εκδόσεις Κλειδάριθμος.

Μανωλόπουλος Ι. και Παπαδόπουλος Α.Ν., Συστήματα Βάσεων Δεδομένων: Θεωρία και Πρακτική Εφαρμογή, Α έκδοση, Εκδόσεις Νέων Τεχνολογιών, Αθήνα 2006 (βασικό σύγγραμμα)

Melony J., (2004) «PHP, MySql και Apache» Εκδόσεις Μ. Γκιούρδας

Melonie, J., (2008) Μάθετε PHP, MySQL και Apache Όλα σε ένα Εκδ. Γκιούρδας Μ., Αθήνα

Melonie J., (2003), Μάθετε PHP, MySQL και Apache σε 24 ώρες, Μτφ. Γκαγκάτσιου Ε., Εκδ. Γκιούρδας Μ., Αθήνα

Ullman, L., (2005) Εισαγωγή στην PHP για τον παγκόσμιο ιστό Με εικόνες, Εκδ. Κλειδάριθμος, Αθήνα

Ullman, L., (2009), Εισαγωγή στις PHP 6 & MYSQL 5 Με εικόνες: Μάθετε PHP 6 και MYSQL 5 γρήγορα και εύκολα, Εκδ. Κλειδάριθμος, Αθήνα

Vaswani, V., 2010 Οδηγός της PHP, Μτφ. Κουτρομπά Χρ., Εκδ. Γκιούρδας Μ., Αθήνα

Welling, L., 2005 Ανάπτυξη Web Εφαρμογών με PHP και MySQL, Εκδ. Γκιούρδας Μ., Αθήνα

Bruse B., (2007), Μάθετε το macromedia dreamweaver 8 σε 24 ώρες Απόδοση: Γιάννης Β. Σαμαράς, Εκδ. Γκιούρδας Μ., Αθήνα

West R, Muck T., (2005), Πλήρης οδηγός του Dreamweaver MX 2004 Απόδοση: Αγαμέμνων Μήλιος, Εκδ. Γκιούρδας Μ., Αθήνα